Содержание

Разделы и статьи в этом PDF.

Пользовательская документация

where to start.mdx

С чего начать ..... where-to-start.mdx

FAQ.mdx

FAQ ..... FAQ.mdx

glossary.mdx

Термины ..... glossary.mdx

archive.mdx

Архив документации ..... archive.mdx

library.mdx

Библиотека email-маркетолога ..... library.mdx

changelog

Обновления платформы ..... changelog/index.mdx

v2025.4.75 ..... changelog/v2025475.mdx

v2025.4.74 ..... changelog/v2025474.mdx

v2025.3.73 ..... changelog/v2025373.mdx

v2025.2.72 ..... changelog/v2025272.mdx

v2025.1.71 ..... changelog/v2025171.mdx

v2024.4.70 ..... changelog/v2024470.mdx

v2024.3.69 ..... changelog/v2024369.mdx

v2024.2.68.2 ..... changelog/v20242682.mdx

v2024.1.68 ..... changelog/v2024168.mdx

profiles and databases

Хранение и сбор данных ..... profiles-and-databases/index.mdx

Ресурсы подписок profiles-and-databases/66427.mdx

Работа с базами данных profiles-and-databases/66420.mdx

Профиль подписчика profiles-and-databases/9405665.mdx

Импорт профилей клиентов и обновление данных ..... profiles-and-databases/profiles-import-and-data-update.mdx

Импорт данных по расписанию profiles-and-databases/9404661.mdx

Автоматизация сбора данных о профиле profiles-and-databases/66433.mdx

Массовое обновление профилей клиентов profiles-and-databases/9406694.mdx

Double opt-in подписка profiles-and-databases/9404703.mdx

Стоп-списки profiles-and-databases/66534.mdx

Связи между профилями profiles-and-databases/20480122.mdx

Экспорт истории профилей profiles-and-databases/49710319.mdx

Экспорт профилей profiles-and-databases/76808636.mdx
Автоматическое создание статического сегмента при импорте profiles-and-databases/37847201.mdx
Как открыть CSV-файл ..... profiles-and-databases/csv-in-excel.mdx
Матчинг ..... profiles-and-databases/matching.mdx
Типы полей в базе данных ..... profiles-and-databases/types-of-db-fields.mdx
Глобальные контрольные группы ..... profiles-and-databases/gcg.mdx
Менеджер подписок ..... profiles-and-databases/subscription-manager.mdx

communication channels

Каналы коммуникации ..... communication-channels/index.mdx
SMS-канал ..... communication-channels/sms.mdx
Схема работы каналов коммуникации ..... communication-channels/scheme.mdx
Настройка собственного from-домена communication-channels/email/9404444.mdx
Настройка и использование постмастеров ..... communication-channels/email/postmasters.mdx
Быстрый старт ..... communication-channels/email/quick-start.mdx
Push-канал ..... communication-channels/push/index.mdx
Mobile Push ..... communication-channels/push/mobile-push/index.mdx
Настройка и подключение communication-channels/push/mobile-push/22708437.mdx
Интеграция приложения с Altcraft ..... communication-channels/push/mobile-push/mobile-push-app-integration/index.mdx
Провайдеры: структура push coo… communication-channels/push/mobile-push/mobile-push-app-integration/49709434.mdx
Обработка и добавление подписки .. communication-channels/push/mobile-push/mobile-push-app-integration/49709437.mdx
Регистрация событий communication-channels/push/mobile-push/mobile-push-app-integration/49709440.mdx
Web Push ..... communication-channels/push/web-push/index.mdx
Предварительные настройки ..... communication-channels/push/web-push/preliminary-settings.mdx
Подключение Web Push на сайт ..... communication-channels/push/web-push/setting-up.mdx
Передача данных в платформу ..... communication-channels/push/web-push/data-import.mdx
Методы Web Push SDK ..... communication-channels/push/web-push/methods.mdx
Apple Safari ..... communication-channels/push/web-push/providers/Apple-Safari_37847599.mdx
Firebase Cloud Messaging ..... communication-channels/push/web-push/providers/firebase-cloud-messaging.mdx
Mozilla Services ..... communication-channels/push/web-push/providers/Mozilla-Services_37847581.mdx
Перенос push-подписок из стороннего сервиса communication-channels/push/web-push/migration/37847852.mdx
Как перенести push-подписки, настроенные для S… communication-channels/push/web-push/migration/37847870.mdx
Миграция с OneSignal communication-channels/push/web-push/migration/37847889.mdx
Email ..... communication-channels/tutorials/email.mdx
SMS ..... communication-channels/tutorials/sms.mdx
Web Push ..... communication-channels/tutorials/web-push.mdx
Mobile Push ..... communication-channels/tutorials/mobile-push.mdx
WhatsApp ..... communication-channels/tutorials/whatsapp.mdx
Viber™ ..... communication-channels/tutorials/viber.mdx
Notify ..... communication-channels/tutorials/notify.mdx
Telegram Bot ..... communication-channels/tutorials/telegram-bot.mdx

Telegram Group ..... communication-channels/tutorials/telegram-group.mdx

segmentation

Сегментация ..... segmentation/index.mdx
Статические сегменты segmentation/9407279.mdx
Динамические сегменты ..... segmentation/dynamic-segments.mdx
Обновляемые сегменты ..... segmentation/updatable-segments.mdx
Сегментация профилей по внешним данным segmentation/66468.mdx
Сегментация профилей по внешним SQL-таблицам segmentation/8650757.mdx
Логические операторы "И" и "ИЛИ" ..... segmentation/operators_and_or.mdx
Рекомендации по работе с сегментами ..... segmentation/segmentation-recommendations.mdx
Условия сегментации ..... segmentation/new-segments/index.mdx
Сегментация по данным профиля ..... segmentation/new-segments/new-segments-from-profile.mdx
Сегментация по взаимодействиям с сущностями ..... segmentation/new-segments/new-segments-interactions.mdx
Сегментация по активности в каналах коммуникации ..... segmentation/new-segments/newsegemntation.mdx

message templates

Шаблоны сообщений ..... message-templates/index.mdx
Блочный редактор для email-шаблона ..... message-templates/block-editor.mdx
Фрагменты шаблонов message-templates/66494.mdx
Изображения в сообщениях message-templates/66507.mdx
Персонализация контента в сообщениях message-templates/66490.mdx
Формирование таблиц на основе элементов массива message-templates/37847460.mdx
Импорт и экспорт шаблона сообщения message-templates/56258129.mdx
Экспорт шаблона из Pixcraft ..... message-templates/export-from-pixcraft.mdx
Импорт шаблона из стороннего сервиса message-templates/76808233.mdx
Работа с шаблонами сообщений ..... message-templates/managing-templates/index.mdx
Работа в редакторе ..... message-templates/managing-templates/template-editor.mdx
Email-шаблон ..... message-templates/managing-templates/email.mdx
SMS-шаблон ..... message-templates/managing-templates/sms.mdx
Push-шаблон ..... message-templates/managing-templates/push.mdx
Telegram-шаблон ..... message-templates/managing-templates/telegram.mdx
WhatsApp-шаблон ..... message-templates/managing-templates/whatsapp.mdx
Viber™-шаблон ..... message-templates/managing-templates/viber.mdx
Notify-шаблон ..... message-templates/managing-templates/notify.mdx
Визуальный редактор для email-шаблона ..... message-templates/visual-editor/index.mdx
Интерфейс редактора message-templates/visual-editor/37848561.mdx
Добавление элементов message-templates/visual-editor/37848658.mdx
Элементы и их настройки message-templates/visual-editor/37848574.mdx
Пользовательские блоки message-templates/visual-editor/37849457.mdx
Стили элемента message-templates/visual-editor/37848592.mdx
Структура элементов message-templates/visual-editor/37849442.mdx

Переменные и функции Altcraft ..... message-templates/altcraft-variables-and-functions/index.mdx
Использование логических выражений в сообщениях message-templates/altcraft-variables-and-functions/66511.mdx
Использование циклов в сообщениях message-templates/altcraft-variables-and-functions/2196195.mdx
Использование переменных маркета в сообщениях message-templates/altcraft-variables-and-functions/56257541.mdx
Использование функционала JSONPath ..... message-templates/altcraft-variables-and-functions/jsonpath.mdx
Использование API-контента в сообщениях message-templates/dynamic-content/66503.mdx
Использование HTML-контента в сообщениях message-templates/dynamic-content/66496.mdx
Использование JSON-контента в сообщениях message-templates/dynamic-content/66498.mdx
Использование контента из SQL базы данных в сообщениях message-templates/dynamic-content/66501.mdx

campaigns

Рассылки ..... campaigns/index.mdx
Календарь рассылок campaigns/66540.mdx
Броадкаст рассылка campaigns/66526.mdx
Регулярная рассылка campaigns/9405823.mdx
Триггерная рассылка campaigns/66530.mdx
Мультивариантный тест (A/B/n) campaigns/9405875.mdx
Тестирование расылок campaigns/66542.mdx
Расписание рассылок campaigns/66538.mdx
Размещения campaigns/9406025.mdx

super campaigns

Кампании ..... super-campaigns/index.mdx
Работа с Кампаниями ..... super-campaigns/where-to-start.mdx
Локальные контрольные группы (ЛКГ) ..... super-campaigns/lcg.mdx
Расширение аудитории в кампании ..... super-campaigns/audience-expansion.mdx
Разметка аудитории в кампаниях ..... super-campaigns/audience-distribution.mdx

automation scenarios

Сценарии автоматизации ..... automation-scenarios/index.mdx
Работа со Сценариями автоматизации automation-scenarios/9406861.mdx
Узлы сценария automation-scenarios/9407058.mdx
Классические сценарии автоматизации маркетинга automation-scenarios/22708542.mdx
Приветственный сценарий: пошаговая настройка automation-scenarios/25985320.mdx
Автоматическое оповещение менеджера через сценарий automation-scenarios/28934313.mdx
Сценарий брошенной корзины automation-scenarios/56254697.mdx

market

Маркет ..... market/index.mdx
Настройки маркета market/49710292.mdx
Заказы market/49710295.mdx
Переменные маркета в шаблонах market/56252235.mdx
Руководство: как отправить письмо подтверждения заказа market/56257397.mdx

Продукты ..... market/products/index.mdx
Создание продукта вручную market/products/56249576.mdx
Импорт продукта из файла market/products/56249588.mdx
Импорт по расписанию ..... market/products/feed-import.mdx
Сегменты продуктов и SKU ..... market/products/segments.mdx
Подготовка YML-файла ..... market/products/yml.mdx

loyalty programs

Лояльность ..... loyalty-programs/index.mdx
Создание и настройка программы лояльности ..... loyalty-programs/loyalties.mdx
Интеграция лояльности с внешними системами ..... loyalty-programs/loyalty-market-integraton.mdx
Базовые кейсы использования программы лояльности ..... loyalty-programs/loyalty-cases.mdx
Промокоды loyalty-programs/9406715.mdx

weblayers

Формы ..... weblayers/forms/index.mdx
Создание формы weblayers/forms/37848193.mdx
Конструктор формы weblayers/forms/37848206.mdx
Оформление формы weblayers/forms/37848222.mdx
Действия при активации формы weblayers/forms/37848225.mdx
Аналитика данных weblayers/forms/37848282.mdx
Связывание данных канала и формы weblayers/forms/37849222.mdx
Условная постраничная логика в формах и опросах weblayers/forms/37849748.mdx
NPS-тестирование ..... weblayers/forms/nps.mdx
Пиксели ..... weblayers/pixels/index.mdx
Целевые действия клиентов и скоринг weblayers/pixels/9406180.mdx
Попапы ..... weblayers/popups/index.mdx
Создание и публикация попапа weblayers/popups/76809121.mdx
Настройка попапа в редакторе кода weblayers/popups/109903904.mdx
Управление попапами вручную через скрипт weblayers/popups/109903905.mdx
Аналитика попапов weblayers/popups/109903934.mdx
Руководство: попап для подписки на push ..... weblayers/popups/popup-to-push.mdx
Базовые кейсы размещения попапа через Менеджер тегов ..... weblayers/popups/popup-with-tagmanager.mdx
Кейс: Создание попапа с виджетом "Колесо фортуны" ..... weblayers/popups/wheel-of-fortune.mdx
Менеджер тегов ..... weblayers/tag-manager/index.mdx
Настройка и установка Менеджера тегов weblayers/tag-manager/95715878.mdx
Типы триггеров weblayers/tag-manager/95715843.mdx
Типы переменных weblayers/tag-manager/95715946.mdx
Связывание пикселя и Менеджера тегов weblayers/tag-manager/109903922.mdx

reports and analytics

Отчеты и аналитика ..... reports-and-analytics/index.mdx

Отчет по каналам ..... reports-and-analytics/channels-report.mdx

Отчёт по трафику ..... reports-and-analytics/traffic-report.mdx

Когортный отчёт ..... reports-and-analytics/cohorts-report.mdx

Время жизни ..... reports-and-analytics/lifetime-report.mdx

Воронка конверсии ..... reports-and-analytics/funnels-report.mdx

Цели ..... reports-and-analytics/goals-report.mdx

Прирост аудитории ..... reports-and-analytics/audience-growth-report.mdx

Карта кликов (Email) ..... reports-and-analytics/click-map-report.mdx

Отчет по программам лояльности ..... reports-and-analytics/loyalty.mdx

Отчёт о возвратах ..... reports-and-analytics/bounces.mdx

Отчёт о недоставках ..... reports-and-analytics/undeliveries.mdx

Отчет по глобальным контрольным группам ..... reports-and-analytics/gcg.mdx

Сводный отчёт ..... reports-and-analytics/summary-report/index.mdx

Все показатели сводного отчета reports-and-analytics/summary-report/33063594.mdx

integrations

Синхронизация статических сегментов integrations/9407286.mdx

Яндекс.Аудитории™ ..... integrations/yandex-audience.mdx

Аудитории Google Ads ..... integrations/google-audience.mdx

Facebook Ads Manager™ ..... integrations/facebook-ads-manager.mdx

Область видимости интеграции integrations/45514832.mdx

WhatsApp ..... integrations/whatsapp.mdx

Viber™ ..... integrations/viber.mdx

Tilda™ integrations/2195939.mdx

Yandex AppMetrica™ ..... integrations/yandex-app-metrica.mdx

Lpgenerator™ integrations/2195949.mdx

VK Реклама™ ..... integrations/vk.mdx

Передаваемые при синхронизации данные ..... integrations/transmitted-data.mdx

Notify ..... integrations/notify.mdx

Подключение Altcraft к Albato integrations/albato-integrations/76808402.mdx

Запуск приветственного сценария через Albato integrations/albato-integrations/768084024.mdx

Передача данных о событии integrations/albato-integrations/768084025.mdx

Отправка триггерной рассылки integrations/albato-integrations/768084026.mdx

Регистрация событий integrations/albato-integrations/768084027.mdx

Захват событий Altcraft integrations/action-hooks/37849833.mdx

Типы событий для захвата integrations/action-hooks/37849805.mdx

Структуры сообщений захвата событий integrations/action-hooks/37849819.mdx

Отправить JSON-запрос батчем integrations/action-hooks/37849807.mdx

Отправить сообщение в очередь RabbitMQ integrations/action-hooks/37849816.mdx

Отправить сообщение в exchange RabbitMQ integrations/action-hooks/37849810.mdx

Отправить сообщение в Kafka broker integrations/action-hooks/37849813.mdx

Предварительное тестирование события integrations/action-hooks/37849821.mdx

settings

Настройки аккаунта settings/9407614.mdx
Настройки атрибутов settings/attributes.mdx
Поисковые теги: создание и применение settings/89326254.mdx
Пользовательские ссылки settings/9407616.mdx
Виртуальные сендеры settings/9405938.mdx
Политики отправки settings/9405554.mdx
Журнал аудита settings/3113292.mdx
Пользователи и разграничение доступа settings/users-groups-access/index.mdx
Двухфакторная аутентификация (2FA) settings/users-groups-access/56255904.mdx
Подключения settings/connections/index.mdx
Подключение к Facebook Ads settings/connections/fb-connection.mdx
Подключение к Google Ads settings/connections/google-ads.mdx
Подключение к Яндекс.Аудиториям™ settings/connections/ya-audiences.mdx
Подключение к 360dialog settings/connections/360.mdx
Подключение к Edna settings/connections/edna.mdx
Подключение к Devino Telecom settings/connections/devino.mdx
Подключение к SMS Traffic settings/connections/smstraffic.mdx
Подключение к VK Рекламе™ settings/connections/vk-connection.mdx
Подключение к MTS OmniChannel settings/connections/mts.mdx
Подключение через OAuth2 settings/connections/oauth20.mdx
Подключение через Basic Authentication settings/connections/basic.mdx
Подключение через Token Authentication settings/connections/token.mdx
Подключение через Custom Authentication settings/connections/custom.mdx
Подключение к Notify settings/connections/notify.mdx
Подключение к Rapporto settings/connections/rapporto.mdx

api

API-запросы: с чего начать api/index.mdx
Импорт и обновление профиля api/api-import.mdx
Запуск триггерной рассылки api/api-trigger.mdx
Отправка профиля клиента в сценарий api/api-scenario.mdx

API / Developer Guide

Взаимодействие с API api-interaction.mdx
Матчинг 37848489.mdx
Список API-методов api-endpoints-list.mdx
Импорт и настройка коллекции API-методов в Postman api-collection.mdx
Импортировать профиль profiles/import.mdx

Обновить профиль ..... profiles/update.mdx
Добавить профиль в базу данных profiles/2195477.mdx
Получить информацию о профиле profiles/2195482.mdx
Импортировать профиль в RabbitMQ profiles/2195492.mdx
Массовое добавление профилей в базу данных profiles/2195496.mdx
Массовое обновление профилей profiles/2195501.mdx
Массовый импорт профилей profiles/2195504.mdx
Приостановить все профили в базе данных profiles/2195553.mdx
Восстановить все приостановленные профили в базе данных profiles/2195556.mdx
Удалить профиль profiles/28936545.mdx
Упрощенный импорт профиля profiles/3113182.mdx
Функциональное обновление полей базы profiles/3113720.mdx
Функциональное обновление полей подписки profiles/3113721.mdx
Выгрузка профилей в файл profiles/3115796.mdx
Получение данных по нескольким профилям profiles/37847989.mdx
Объединение нескольких профилей profiles/41484440.mdx
Отписать профиль от ресурса profiles/41484441.mdx
Разделение профиля profiles/41484445.mdx
Получить историю одного профиля profiles/action-history/2195546.mdx
Получить историю нескольких профилей profiles/action-history/2195548.mdx
Добавить связь profiles/profile-relations/9406148.mdx
Удалить связь profiles/profile-relations/9406152.mdx
Усилить/ослабить связь profiles/profile-relations/9406155.mdx
Перезаписать значения свойств связи profiles/profile-relations/9406157.mdx
Получить информацию о связях профиля profiles/profile-relations/9406159.mdx
Получить список связей профиля ..... profiles/profile-relations/get-relations-list.mdx
Добавить или редактировать подписку profiles/subscriptions/2195519.mdx
Получить все подписки профиля profiles/subscriptions/2195522.mdx
Получить все подписки нескольких профилей profiles/subscriptions/2195524.mdx
Получить информацию о подписке профиля profiles/subscriptions/2195527.mdx
Удалить подписку профиля profiles/subscriptions/2195529.mdx
Восстановить удаленную подписку профиля profiles/subscriptions/2195531.mdx

Приостановить все подписки profiles/subscriptions/2195533.mdx
Восстановить все приостановленные подписки profiles/subscriptions/2195539.mdx
Получить список баз данных databases/2195563.mdx
Получить информацию о базе данных databases/2195565.mdx
Получить информацию о полях базы данных databases/2195567.mdx
Очистка базы данных для тестирования databases/2195815.mdx
Получить статистику по базе данных ..... databases/get-counters.mdx
Обновить статистику по базе данных ..... databases/refresh-counters.mdx
Получить список ресурсов resources/2195587.mdx
Получить информацию о ресурсе resources/2195590.mdx
Получить информацию о полях подписки ресурса resources/2195591.mdx
Получить статистику по ресурсам ..... resources/get-counters.mdx
Обновить статистику по ресурсам ..... resources/refresh-counters.mdx
Добавить сегмент segments/2195594.mdx
Обновить сегмент segments/2195596.mdx
Получить информацию о сегменте segments/2195598.mdx
Получить список сегментов segments/2195600.mdx
Удалить сегмент segments/2195602.mdx
Получить статистику по сегментам ..... segments/get-counters.mdx
Обновить статистику по сегментам ..... segments/refresh-counters.mdx
Добавить или удалить профиль segments/9407390.mdx
Получить данные профилей статического или обновляемого сегмента segments/9407394.mdx
Добавить стоп-список suppression-lists/2195613.mdx
Переименовать стоп-список suppression-lists/2195615.mdx
Получить информацию о стоп-списке suppression-lists/2195617.mdx
Получить информацию о нескольких стоп-списков suppression-lists/2195619.mdx
Удалить стоп-список suppression-lists/2195621.mdx
Выгрузить данные из стоп-списка в файл suppression-lists/76809125.mdx
Добавление и удаление из стоп-списка ..... suppression-lists/suppression-list-actions/index.mdx
Проверить email-адрес в стоп-списке suppression-lists/suppression-list-actions/2195633.mdx
Добавить email-адрес в стоп-список suppression-lists/suppression-list-actions/2195625.mdx
Добавить один или несколько email-адресов в стоп-список suppression-lists/suppression-list-actions/2195627.mdx

Удалить email-адрес из стоп-списка suppression-lists/suppression-list-actions/2195637.mdx
Удалить все email-адреса из стоп-списка suppression-lists/suppression-list-actions/2195641.mdx
Проверить домен в стоп-списке suppression-lists/suppression-list-actions/2195635.mdx
Добавить домен в стоп-список suppression-lists/suppression-list-actions/2195629.mdx
Добавить один или несколько доменов в стоп-список suppression-lists/suppression-list-actions/2195631.mdx
Удалить домен из стоп-списка suppression-lists/suppression-list-actions/2195639.mdx
Удалить все домены из стоп-списка suppression-lists/suppression-list-actions/2195643.mdx
Проверить номер телефона в стоп-списке suppression-lists/suppression-list-actions/76809084.mdx
Добавить номер телефона в стоп-список suppression-lists/suppression-list-actions/76809087.mdx
Добавить один или несколько номеров в стоп-список suppression-lists/suppression-list-actions/76809091.mdx
Удалить номер из стоп-списка suppression-lists/suppression-list-actions/76809096.mdx
Удалить все номера из стоп-списка suppression-lists/suppression-list-actions/76809100.mdx
Получить список шаблонов templates/2195606.mdx
Получить информацию о шаблоне templates/2195608.mdx
Удалить шаблон templates/56255090.mdx
Добавить шаблон сообщения templates/56255095.mdx
Обновить шаблон сообщения templates/56255277.mdx
Channel object ..... templates/Channel-object_56255164.mdx
Получить список рассылок campaigns/2195648.mdx
Получить информацию о рассылке campaigns/2195650.mdx
Получить лог рассылки campaigns/2195652.mdx
Клонировать рассылку campaigns/2195654.mdx
Удалить рассылку campaigns/2195656.mdx
Активировать рассылку campaigns/2195660.mdx
Деактивировать рассылку campaigns/2195662.mdx
Получить статус рассылки campaigns/2195664.mdx
Получить список броадкаст рассылок campaigns/broadcast-campaigns/2195885.mdx
Получить информацию о броадкаст рассылке campaigns/broadcast-campaigns/2195887.mdx
Добавить броадкаст рассылку campaigns/broadcast-campaigns/2195891.mdx
Обновить броадкаст рассылку campaigns/broadcast-campaigns/2195895.mdx
Запустить броадкаст рассылку campaigns/broadcast-campaigns/3116196.mdx
Получить список регулярных рассылок campaigns/regular-campaigns/2195899.mdx

Получить информацию о регулярной рассылке campaigns/regular-campaigns/2195901.mdx
Добавить регулярную рассылку campaigns/regular-campaigns/2195903.mdx
Обновить регулярную рассылку campaigns/regular-campaigns/2195905.mdx
Запустить регулярную рассылку campaigns/regular-campaigns/3116201.mdx
Получить список триггерных рассылок campaigns/trigger-campaigns/2195915.mdx
Получить информацию о триггерной рассылке campaigns/trigger-campaigns/2195917.mdx
Добавить триггерную рассылку campaigns/trigger-campaigns/2195922.mdx
Обновить триггерную рассылку campaigns/trigger-campaigns/2195925.mdx
Запуск триггерной рассылки (API call) campaigns/trigger-campaigns/2196177.mdx
Импорт профиля + Отправка триггера campaigns/trigger-campaigns/22708362.mdx
Задание на массовую отправку триггера campaigns/trigger-campaigns/33063451.mdx
Задание на массовый импорт профилей + отправка триггера campaigns/trigger-campaigns/33063474.mdx
Массовая отправка триггера campaigns/trigger-campaigns/37847471.mdx
Массовый импорт профилей + отправка триггера campaigns/trigger-campaigns/37847473.mdx
Клонировать триггер рассылку campaigns/trigger-campaigns/56258725.mdx
Data array ..... campaigns/trigger-campaigns/Data-array_41484531.mdx
Получить информацию о кампании ..... super-campaigns/get-scampaigns-info.mdx
Получить список кампаний ..... super-campaigns/get-scampaigns-list.mdx
Активация кампании ..... super-campaigns/scampaigns-activation.mdx
Завершение кампании ..... super-campaigns/scampaigns-complete.mdx
Деактивация кампании ..... super-campaigns/scampaigns-deactivation.mdx
Получить статус кампании ..... super-campaigns/scampaigns-status.mdx
Отправить профиль клиента в сценарий automation-scenarios/2195668.mdx
Одновременный импорт и запуск профиля в сценарий automation-scenarios/28934227.mdx
Массовый импорт и запуск профилей в сценарий automation-scenarios/33063492.mdx
Задание на массовый импорт и запуск профилей в сценарий automation-scenarios/33063493.mdx
Получить список сценариев automation-scenarios/95715617.mdx
Активировать сценарий ..... automation-scenarios/activate-workflow.mdx
Деактивировать сценарий ..... automation-scenarios/deactivate-workflow.mdx
Импортировать промокоды promocodes/2195573.mdx
Получить информацию о промокоде promocodes/2195575.mdx
Активировать промокод promocodes/2195577.mdx

Обновить промокод promocodes/2195579.mdx
Привязать промокод к профилю promocodes/2195581.mdx
Отвязать промокод от профиля promocodes/2195583.mdx
Получить все промокоды promocodes/3113099.mdx
Получить уровень профиля в программе лояльности ..... loyalty/tiers.mdx
Экспорт транзакций баллов ..... loyalty/transactions-export.mdx
Сгораемые баллы за период ..... loyalty/expiring-points.mdx
Получение транзакций по счёту профиля ..... loyalty/transactions-list.mdx
Получение списка триггерных промоакций ..... loyalty/trigger-promotions.mdx
Начисление баллов участнику ..... loyalty/accrue-points.mdx
Списание баллов участника ..... loyalty/redeem-points.mdx
Подтверждение временной транзакции ..... loyalty/commit-transaction.mdx
Предварительный расчет заказа ..... loyalty/preorder.mdx
Подтверждение заказа ..... loyalty/preorder-commit.mdx
Отмена временной транзакции ..... loyalty/rollback-transaction.mdx
Отмена балльной транзакции ..... loyalty/cancel-transation.mdx
Получение баланса балльного счёта ..... loyalty/balance.mdx
Регистрация участника в программе лояльности ..... loyalty/register-member.mdx
Удаление участника из программы лояльности ..... loyalty/delete-member.mdx
Регистрация события достижения цели goals/2195843.mdx
Обработка и добавление подписки application-push-notifications/2196103.mdx
Добавить события с app push application-push-notifications/2196102.mdx
Структура заказа (order data object) market/market-objects/56249763.mdx
Product data object ..... market/market-objects/Product-data-object_56249759.mdx
Структура SKU (SKU data object) ..... market/market-objects/SKU-data-object_56249769.mdx
Категории (categories array) ..... market/market-objects/Categories-array_56249780.mdx
Custom fields array ..... market/market-objects/Custom-fields-array_56249775.mdx
Импорт заказа и статусов позиций market/orders/56249746.mdx
Получить список заказов market/orders/56249749.mdx
Удалить заказ market/orders/76808865.mdx
Получить статус заказа ..... market/orders/get-status.mdx
Изменение статуса позиции заказа ..... market/orders/update-order-line-status.mdx

Импорт продуктов, SKU и категорий market/products-and-sku/56249718.mdx
Получение списка продуктов market/products-and-sku/56249723.mdx
Получение списка SKU market/products-and-sku/56249724.mdx
Импорт SKU и категорий market/products-and-sku/56249733.mdx
Удалить продукты ..... market/products-and-sku/delete-product.mdx
Удалить SKU ..... market/products-and-sku/delete-sku.mdx
Получить сводный отчет analytic-reports/2195743.mdx
Получить отчет о возвратах analytic-reports/2195748.mdx
Получить отчет о недоставках analytic-reports/2195750.mdx
Получить список сендеров senders/2195754.mdx
Получить список виртуальных сендеров senders/virtual-senders/2195762.mdx
Получить информацию о виртуальном сендере senders/virtual-senders/2195765.mdx
Клонировать виртуальный сендер senders/virtual-senders/2195767.mdx
Добавить виртуальный сендер senders/virtual-senders/2195769.mdx
Обновить виртуальный сендер senders/virtual-senders/2195771.mdx
Удалить виртуальный сендер senders/virtual-senders/2195773.mdx
AKMTA object ..... objects/AKMTA-object_2195811.mdx
Content object ..... objects/Content-object_2195781.mdx
Email rule object ..... objects/Email-rule-object_2195791.mdx
File object ..... objects/File-object_2195839.mdx
Profile data object ..... objects/Profile-data-object_3116008.mdx
Sender object ..... objects/Sender-object_2195784.mdx
Sender types ..... objects/Sender-types_2195789.mdx
SMS rule object ..... objects/SMS-rule-object_2195796.mdx
Start schedule object ..... objects/Start-schedule-object_2195802.mdx
Subscription object ..... objects/Subscription-status_2195807.mdx
Trigger types ..... objects/Trigger-types_2195809.mdx
Добавить запрос сегментации ..... datatables/query_segments/add.mdx
Обновить запрос сегментации ..... datatables/query_segments/update.mdx
Получить информацию о запросе сегментации ..... datatables/query_segments/get.mdx
Получить список запросов на сегментацию ..... datatables/query_segments/list.mdx
Удалить запрос сегментации ..... datatables/query_segments/delete.mdx

Добавить запрос для шаблонов ..... datatables/query_templates/add.mdx
Обновить запрос для шаблонов ..... datatables/query_templates/update.mdx
Получить информацию о запросе для шаблонов ..... datatables/query_templates/get.mdx
Получить список запросов для шаблонов ..... datatables/query_templates/list.mdx
Удалить запрос для шаблонов ..... datatables/query_templates/delete.mdx
Загрузить файл miscellaneous/2195777.mdx
Получить веб-версию сообщения miscellaneous/2196089.mdx
Push провайдеры miscellaneous/37848490.mdx
Дедупликация запросов miscellaneous/41484312.mdx
Работа с API через RabbitMQ miscellaneous/44662785.mdx
Список гендерных идентификаций miscellaneous/9405775.mdx
Получить допустимые значения полей browsers, devices, tz, oses, languages ..... miscellaneous/dictionary.mdx
mSDK ..... sdk/mobile/index.mdx
Работа с ролевым и JWT-токеном ..... sdk/mobile/jwt-r-token.mdx
Быстрый старт ..... sdk/mobile/android/quick-start.mdx
Конфигурация SDK ..... sdk/mobile/android/setup.mdx
Функционал SDK ..... sdk/mobile/android/functionality.mdx
Публичный API SDK ..... sdk/mobile/android/api.mdx
Firebase Cloud Messaging ..... sdk/mobile/android/providers/fcm.mdx
Huawei Mobile Services ..... sdk/mobile/android/providers/hms.mdx
RuStore ..... sdk/mobile/android/providers/rustore.mdx
Быстрый старт ..... sdk/mobile/ios/quick-start.mdx
Конфигурация SDK ..... sdk/mobile/ios/setup.mdx
Функционал SDK ..... sdk/mobile/ios/functionality.mdx
Публичный API SDK ..... sdk/mobile/ios/api.mdx
Apple Push Notification Service ..... sdk/mobile/ios/providers/apns.mdx
Firebase Cloud Messaging ..... sdk/mobile/ios/providers/fcm.mdx
Huawei Mobile Services ..... sdk/mobile/ios/providers/hms.mdx
Быстрый старт ..... sdk/mobile/react-native/quick-start.mdx
Конфигурация SDK ..... sdk/mobile/react-native/setup.mdx
Функционал SDK ..... sdk/mobile/react-native/functionality.mdx
Публичный API SDK ..... sdk/mobile/react-native/api.mdx

С чего начать

Altcraft Platform — это омниканальная CDP-платформа, предназначенная для хранения данных о клиентах и управления коммуникациями через различные каналы: email, SMS, push-уведомления, мессенджеры и другие.

Платформа включает большое количество функций и инструментов, поэтому при первом входе может быть неочевидно, с чего именно начать работу и как связаны между собой различные модули интерфейса. Эта статья поможет разобраться в работе Altcraft Platform и подскажет, в какие разделы документации стоит переходить дальше.

Если вы только начинаете знакомство с платформой, рекомендуем также ознакомиться с разделами FAQ и Термины.

Если удобнее изучать платформу через практические примеры, в статье также приведены видео с разбором базовых операций: подготовки базы профилей и запуска первой email-рассылки.

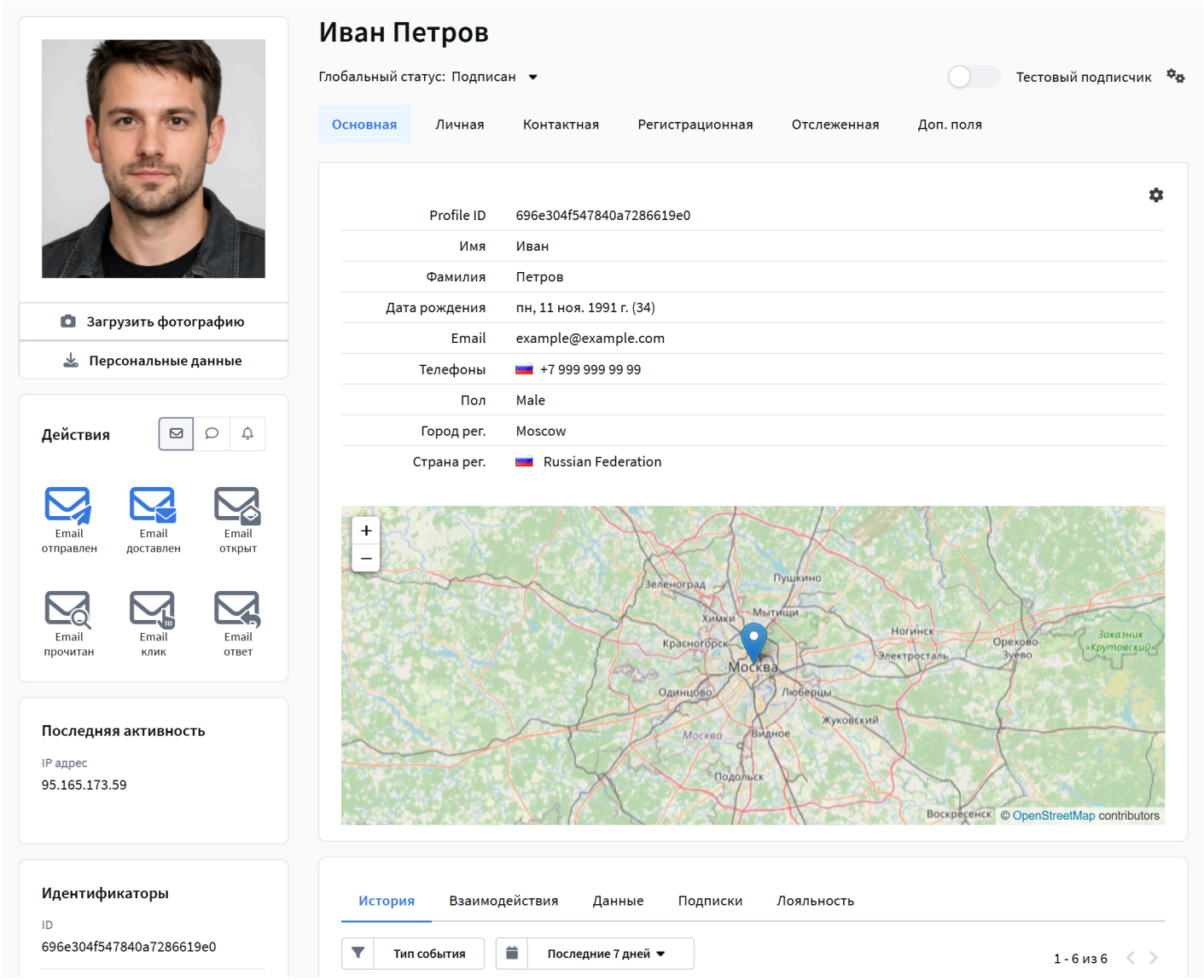
Базы профилей и профиль подписчика

В основе Altcraft Platform лежит работа с данными. Все коммуникации, автоматизация и аналитика строятся вокруг профилей клиентов и их действий. Все данные о клиентах в Altcraft хранятся в базах профилей.

База профилей определяет структуру данных: какие поля используются, какие типы данных допустимы, а также какие ресурсы (источники подписок) могут быть связаны с профилями. Как правило, одна база профилей соответствует одному продукту, сервису или бизнес-направлению. Более подробную информацию по настройке структуры данных вы можете узнать в этой статье.

Профиль подписчика — это единый цифровой профиль клиента. В нём объединяются контактные данные, персональная информация, история действий, подписки на каналы и участие в маркетинговых активностях. В документации могут использоваться термины “клиент”, “подписчик” или “пользователь”, однако во всех случаях речь идёт об одном и том же объекте — профиле подписчика.

Например, если профиль сначала содержит только email, затем к нему добавляется подписка на SMS, а позже фиксируется использование мобильного приложения, все эти данные продолжают храниться в рамках одного профиля подписчика.



Более подробную информацию о структуре профиля и его вкладках вы можете узнать в этой статье.

Ресурсы и подписки

Для управления коммуникациями в Altcraft используется ресурсы подписок.

Ресурс — это логический источник контента, в рамках которого клиент получает сообщения. Его можно представить как контекст или точку взаимодействия: сайт, мобильное приложение, сервис, бренд или отдельный проект компании:

Ресурсы позволяют разделять подписки и согласия клиентов. Один и тот же профиль может быть подписан на несколько ресурсов и иметь разные статусы подписки в каждом из них.

Подписка профиля связывает клиента с конкретным ресурсом и каналом связи. Именно подписка определяет, можно ли отправлять сообщения по конкретным контактным данным.

Более подробную информацию о настройке ресурсов и подписок вы можете узнать в этой статье.

Как создать базу профилей, ресурс и подписку

Ниже показан базовый процесс подготовки данных для коммуникаций: создание базы профилей, ресурса и добавление профилей с подпиской. Эти действия формируют основу для дальнейшей сегментации, рассылок и сценариев автоматизации.

Rutube video player

Данные и история действий

Altcraft Platform фиксирует все ключевые действия клиента и сохраняет их в истории профиля. Это могут быть отправки и открытия сообщений, клики по ссылкам, заполнение форм, события на сайте или в приложении, а также данные о заказах и промокодах:

История действий используется для сегментации, автоматизации и аналитики. Например, вы можете отправить сообщение только тем клиентам, которые открывали письма за последние 7 дней или совершили покупку.

Сегментация аудитории

Сегментация позволяет выделять группы профилей по заданным условиям:

В Altcraft используются статические, динамические и обновляемые сегменты. Они отличаются способом формирования аудитории и логикой обновления данных.

Статические сегменты применяются для работы с фиксированными списками профилей. Динамические сегменты пересчитываются в момент запуска рассылки и всегда используют актуальные данные. Обновляемые сегменты пересчитываются по заданному расписанию и сохраняют результат до следующего обновления.

Например, можно создать сегмент клиентов, которые совершили заказ за последнюю неделю, или сегмент активных подписчиков, открывавших письма в течение месяца.

Более подробную информацию о типах сегментов и условиях выборки вы можете узнать в этой статье.

Коммуникации

Коммуникации в Altcraft Platform — это отправка сообщений профилям через различные каналы связи. Платформа поддерживает email, SMS, Web Push, Mobile Push, мессенджеры и другие каналы, которые могут использоваться как по отдельности, так и в рамках омниканальных сценариев.

Каждый канал имеет свои особенности настройки и работы, но общая логика коммуникаций в платформе едина. Сообщения отправляются на основе данных профилей и всегда учитывают контактные данные клиента, его подписки и согласия.

В Altcraft существует несколько основных видов рассылок. Броадкаст-рассылки используются для единичных отправок, например новостей или акций. Регулярные рассылки подходят для повторяющихся коммуникаций по расписанию. Триггерные рассылки позволяют автоматически отправлять сообщения в ответ на действия клиента или наступление определённых событий.

Коммуникации могут запускаться как на всю базу профилей, так и на отдельные сегменты аудитории. Выбор аудитории для отправки обычно строится на основе сегментов, сформированных по данным профилей и истории их действий. Кроме того, рассылки часто используются как часть более сложных процессов — сценариев автоматизации или кампаний.

Как запустить первую email-рассылку

INFO

Для отправки email требуется настроенный “DKIM” для домена отправителя. Он не выдается автоматически. Для подключения обратитесь в чат технической поддержки.

Ниже показан пример полного процесса создания email-рассылки: от подготовки шаблона до запуска отправки. Видео демонстрирует, как работа с профилями, подписками и шаблонами объединяется в реальный процесс коммуникации.

Rutube video player

INFO: Обратите внимание:

В интерфейсе, показанном на видео, может использоваться термин “кампания” в значении рассылки.
В актуальной версии платформы терминология изменена:

Рассылки — это отдельный инструмент для отправки сообщений (Броадкаст, Регулярная, Триггерная)

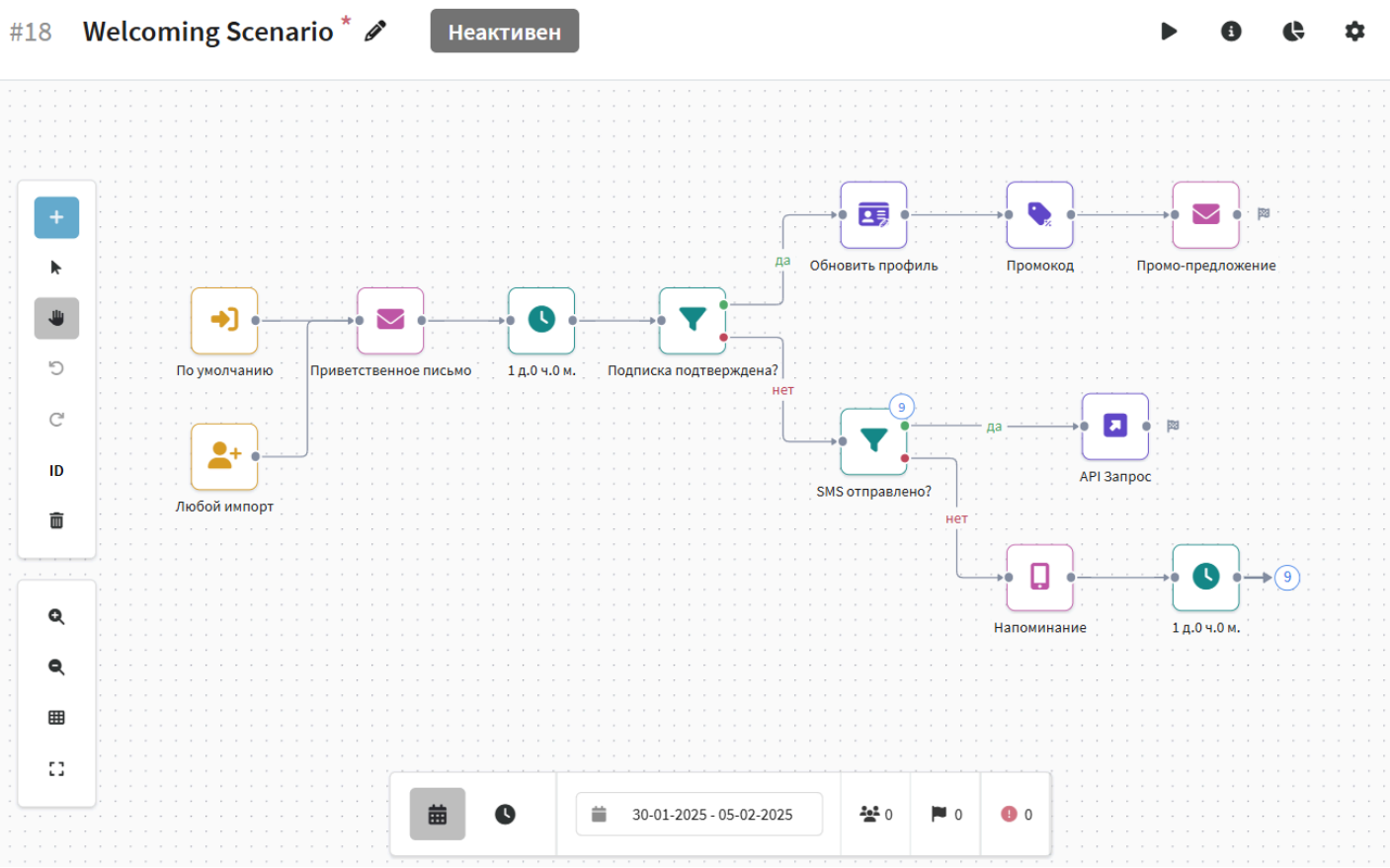
кампании — это другой инструмент, предназначенный для построения комплексных маркетинговых процессов и объединения коммуникаций, сценариев и логики взаимодействия с клиентами.

Для первого знакомства с отправкой сообщений рекомендуется начать с практических гайдов по каналам коммуникации. Например, со статьи Первая email-рассылка. Полный список обучающих материалов доступен в разделе Создание рассылок с нуля.

Сценарии автоматизации

Сценарии автоматизации позволяют выстраивать цепочки коммуникаций с клиентами на основе их действий, событий и условий.

Сценарии используются для онбординга, приветственных цепочек, реактивации, напоминаний и других типовых маркетинговых задач. Профили могут попадать в сценарии как из сегментов, так и напрямую из базы, если они соответствуют условиям входа.



Общий обзор инструмента доступен в разделе Сценарии автоматизации, а примеры типовых сценариев — в статье Классические сценарии автоматизации маркетинга.

Другие возможности платформы

Altcraft Platform не ограничивается хранением данных, сегментацией и отправкой сообщений. Платформа включает набор дополнительных модулей, которые позволяют выстраивать комплексную работу с клиентами на всех этапах их жизненного цикла.

Для управления маркетинговыми активностями применяются кампании. Подробнее — в разделе Кампании.

Работа с товарными и заказными данными описана в разделе Маркет.

Модуль лояльности позволяет создавать программы поощрения и использовать промокоды. Подробнее — в разделе Лояльность.

Для сбора данных и управления подписками используется веб-слой. Подробнее — в разделе Веб-слой.

Обзор аналитики представлен в разделе Отчёты и аналитика.

Интеграции описаны в разделе Интеграции, а технические возможности — в Документации разработчика и статье API-запросы: с чего начать.

FAQ

FAQ

Работа с подписками профиля

Люди, которые отписываются не по ссылке в письме, а в интерфейсе почтового сервиса, отмечаются как отписавшиеся?

Да, если почтовый сервис поддерживает специальный заголовок, который мы передаем в письмо.

Если при отправке сообщения не был выбран ресурс — отписка происходит глобально.

Если отправка сообщения была с указанием ресурса — отписка происходит по ресурсу.

Почему при отписке от рассылки без ресурса статус профиля меняется?

Если рассылка происходит без указания ресурса, используется глобальный статус профиля.

Восстановится ли пользователь из стоп-списка, если захочет вновь подписаться?

Нет, если рассылка была без ресурса или от того же ресурса. Воспользуйтесь API, чтобы восстановить его (см. страницу Стоп-списки).

В дальнейшем рекомендуем пользоваться подписками на разные ресурсы.

Что такое идентификатор ресурса?

Ресурс — это источник рассылок по одному или нескольким каналам. Например, один сайт — один ресурс. Идентификатор ресурса — число, которое отображается рядом с его названием. Используется в API запросах.

Какие различия между стоп-списком и отпиской?

Стоп-список и отписка это разные вещи. Стоп-список обычно используют для временного исключения из рассылок, например когда контент однозначно вызовет негатив группы подписчиков.

Как быть с подтверждением подписки для тех, кто увидел только текстовую версию?

У них тоже может быть ссылка. Имя для нее можно также задать как `<name="имя">`. Часть в угловых скобках не будет показана при отправке.

Какие сущности можно помечать тегом в платформе?

Базы данных, ресурсы, сегменты, каналы, шаблоны, импорты по расписанию, стоп-списки, рассылки, сценарии, промокоды, отчёты.

Почему не совпадает количество кликов по ссылке отписки и количество отписок?

Пользователь может отписаться не только по ссылке в письме, но и через интерфейс сервиса (например, по кнопке). Такие действия не всегда считаются кликами по ссылке. Поэтому общее количество кликов может не совпадать с числом отписок.

Работа с базами данных

Что должен сделать разработчик, когда необходимо настроить обмен данными (например, из форм подписок) с сайтом?

Разработчику необходимо предоставить API-токен и ссылку на стр. документации — API пользователя.

Можно ли загрузить базу без email-адресов? Например для СМС-рассылки.

Можно. Выберите совпадение по email-адресу из профиля. Email-адрес заполнится данными, типа “ID@localhost”.
Идентификатор профиля генерируется платформой.

Какие есть способы определить штат или страну подписчика?

Платформа умеет определять геолокацию по IP-адресу.

Как передавать подписки на ресурс при импорте профилей в базу?

В объект **data** при импорте подписчиков включите массив с подписками, например:

```
subscriptions": [
      {
        "resource_id": 46,
        "channel": "email",
        "email_data": {
          "email": "example_email@example.org",
        },
        "priority": 0,
        "status": "subscriber"
      }, ...
```

И упрощенный вариант:

```
subscriptions": [
      {
        "resource_id": 46,
        "channel": "email",
        "email": "example_email@example.org",
        "priority": 0,
        "status": "subscriber"
      },
```

Можно ли в одном API запросе на импорт профиля передать несколько профилей?

Для импорта нескольких подписчиков используйте Массовый импорт профилей.

Как очистить базу данных?

Используйте API запрос на очистку базы данных.

Возможна также очистка базы запросом к MongoDB.

Можно ли узнать устройства, ОС и браузеры, которые используются пользователями?

В профиле пользователя собирается информация в поле UserAgent.

Как сделать в профиле поле, обозначающее его интересы в разных категориях?

В базе данных можно создать поле типа “Список тегов”. Теги можно добавлять и убирать вручную, через API и через сценарии автоматизации.

Каким образом можно скопировать базу со всеми профилями и подписками?

При копировании базы данных копируется только её структура. Для копирования профилей нажмите на иконку “Скопировать профили клиентов”, после чего будет предложено выбрать сегмент.

Если я экспортирую клиентов из базы данных, я экспортирую email-адреса из профиля или из профиля и подписок?

Подписки можно экспортировать только в зашифрованном виде. По умолчанию будут экспортированы все подписки профиля. В меню экспорта можно настроить, какие именно подписки на какие ресурсы учитывать.

Почему показатель в отчете не совпадает с количеством профилей в сегменте?

Отчеты строятся по событиям, а не по профилям. Один профиль может сгенерировать несколько одинаковых событий — например, дважды кликнуть по ссылке или несколько раз открыть письмо. В таких случаях количество событий превышает количество уникальных профилей в сегменте.

Почему не пересчитывается статический сегмент?

Статический сегмент не обновляется автоматически — это фиксированный список профилей, сформированный по заданным условиям на момент создания. Если данные в базе изменились, статический сегмент нужно пересчитать вручную. Для этого нажмите кнопку **“Пересчитать”** в верхней части редактора сегмента.

При пересчёте будут использоваться только актуальные фильтры, а добавленные или удалённые вручную профили не сохранятся.

Работа с рассылками

Какой информационный характер несет очередь у рассылок?

Показывает текущий статус отправки сообщений для этой рассылки.

Как дослать рассылку тем, кто ее не получил?

Можно сделать сегмент и выбрать всех, кому не было доставлено сообщение.

Есть ли глобальный лимит по количеству рассылок одному пользователю в день, неделю? Можно ли игнорировать его для каких-то писем? Где устанавливается?

Есть, это называется “Политика отправки”. Может быть назначена на ресурс или целиком на аккаунт. В некоторых рассылках можно переназначить, если необходимо, например, её игнорировать.

Как отправлять рассылку постепенно (частями)?

Установите лимиты на отправку в рассылке.

Почему, когда я открыл(а) письмо и добавил(а) его в спам, жалоба не отразилась в отчете?

Возможно вы не настроили FBL/CFL. Обратитесь к вашему системному администратору.

Как выглядит API запрос на отправку триггера по нескольким подписчикам?

```
{
  "token": "e00c3da0a53e4e54a795da58701f4d70",
  "id": 5511,
  "custom_request": {
    "Custom_ID": {
      "$in": [
        "IpHicQquekzYkufl", "xzFKLXeaJkCYHuMb", "SRNOwHVZiWFItJhr"
      ]
    }
  }
}
```

Подробнее отправка триггеров описана в этом разделе.

Работа с шаблонами сообщений

Как пользоваться таблицами данных?

См. страницу Использование контента из SQL базы данных в сообщениях.

Работа со сценариями

Как профиль двигается по сценарию, если шаг с отправкой сообщения завершился ошибкой?

Если отправка сообщения завершилась ошибкой (например, по технической причине), профиль автоматически переходит на следующий шаг сценария.

Как профиль двигается по сценарию, если произошла ошибка в элементе API-вызова, обновления профиля или условия?

Если в шаге произошла ошибка, и в сценарии не предусмотрен отдельный выход для обработки ошибки, профиль будет исключён из сценария. Это делается для предотвращения некорректного выполнения.

HTTP-элемент поддерживает обработку ошибок — при неудачном запросе он может выполнить повторную попытку или направить профиль по альтернативному пути.

Что будет, если удалить шаг сценария, в котором находятся профили?

Если шаг сценария удаляется во время выполнения сценария и в нём находятся активные профили, то такие профили будут немедленно исключены из сценария.

Как посмотреть apicontent профиля?

Чтобы просмотреть apicontent профиля, наведите курсор на нужный элемент сценария. Над ним будет отображаться иконка с числом — это количество профилей, находящихся в этом элементе. Кликните по этой

иконке, чтобы открыть список профилей. В открывшемся окне можно увидеть данные каждого профиля, включая блок apicontent, если в сценарии использовались API-запросы.

Рекомендации по работе со сценариями

- Старайтесь избегать условия с отрицанием (например, “не равно”) — это усложняет логику и снижает наглядность сценария.
- Рекомендуется отсеивать большую часть профилей уже в стартовом элементе сценария, чтобы минимизировать лишнюю нагрузку на сценарий.
- Используйте статические сегменты.

Прочее

В интерфейсе недоступен полный функционал.

Обратитесь к администратору платформы для включения Smart настроек.

В какой кодировке приложение импортирует файлы?

UTF-8.

Где можно выгрузить отчет?

В основном меню выберите пункт “Задания”.

Можно ли при передаче события в платформу автоматически обновлять поля профилей?

Да, используйте сценарии (см. страницу Работа со Сценариями)

Почему в отчетах нет жалоб?

Для появления в отчетах жалоб необходимо настроить FBL. Обратитесь к вашему систему администратору.

> **NOTE**
>
> В платформе жалобы обрабатываются с автоматическим изменением подписки.

Что такое пиксель и какую информацию он передаёт?

Пиксель — это скрипт, отправляющий запрос на сервер отслеживания, когда подписчик совершает какое-либо действие на веб-сайте или в приложении.

Пиксель может передавать информацию, связанную с этим действием, например: адрес сайта, с которого перешёл подписчик, достигнутую цель и ценность совершённого действия. Данные записываются в коде

каждого пикселя и передаются в историю активности профиля. Достигнутые цели и ценность действий могут использоваться для сегментации, в том числе автоматически — в сценариях.

Также данные пикселей используются в аналитических отчётах “Пиксели” и “Цели”.

Есть ли возможность строить сегмент по параметру User-Agent?

Да, в сегментах он разбит на Браузер/Платформа.

Кнопки импорта и генерации промокодов не работают.

Сначала необходимо сохранить программу лояльности.

Название файлов во вложении в письмах имеет некорректное название

Решить проблему можно следующими способами:

1. Переустановить Outlook из пакета Standard.
2. Настроить клиент Outlook: использовать UTF-8 для параметра **Кодировка исходящих сообщений**. Для этого необходимо:
    - В приложении Microsoft Outlook выбрать **Файл** -> **Параметр** -> **Дополнительно**.
    - В поле “Кодировка исходящих сообщений” выбрать UTF-8
3. Не использовать русские символы в названии файла.
4. Воспользоваться другой версией Outlook.

Как отслеживаются открытия и прочтения email-писем?

В конец письма автоматически добавляются несколько пикселей размером 1×1.

При загрузке первого фиксируется открытие письма. При обращении ко второму пикселю, с небольшим задержанием (до 8 секунд), фиксируется прочтение.

Как отслеживается геопозиция профиля?

Геолокация профиля определяется по его IP-адресу в момент открытия письма или выполнения действия.

Термины

Термины email-маркетинга

API (Application Programming Interface, интерфейс программирования приложений) — это язык общения одной компьютерной программы с другой для совместного исполнения какой-нибудь общей задачи, когда одна программа выполняют запросы другой.

BCC (Blind Carbon Copy, скрытая копия) — скрытые получатели письма, чьи адреса не показываются другим получателям.

Caller ID — идентификатор абонента-отправителя SMS сообщений. Может содержать латинские буквы и цифры — без пробелов.

Double Opt-In (двухэтапная подписка) — пользователь указал свой email при подписке и подтвердил его, перейдя по специальной ссылке в письме подтверждения. Преимущество: в базу попадает более заинтересованная и активная аудитория. Недостаток: снижение конверсии, так как часть аудитории не подтверждает подписку.

DKIM (Domain Keys Identified Mail) — технология идентификации отправителя Email, которая добавляет в письмо цифровую подпись, связанную с from-доменом. Подпись автоматически проверяется на стороне получателя, после чего используется для уточнения репутации и помечается для пользователя.

OR (Open Rate) — это процент получателей, открывших письмо.

CTR (Click-Through Rate) — это процентное соотношение кликов по ссылке в письме к количеству доставленных писем.

CTOR (Click to Open Rate) — это процентное соотношение кликов по ссылке в письме к количеству открытий письма.

SB (Soft Bounce, мягкие возвраты) — возвращение письма отправителю из-за переполненности ящика или временной недоступности сервера получателя.

SPF (Sender Policy Framework) — технология, позволяющая проверить, не подделан ли домен отправителя. Для её работы необходимо добавить на отправляющий домен (сендер домен, фром домен) специальную TXT-запись, которая определяет политику разрешения отправки этого домена с различных хостов.

SQL (язык структурированных запросов, Structured Query Language) — специальный язык, используемый для работы с базами данных. С помощью SQL можно задавать структуру базы, организовывать выборку данных из неё по каким-то определённым критериям, удалять данные.

HB (Hard Bounce, жесткие возвраты) — это адреса, которые не принимают письма из-за недействительных email-ов или неожиданной ошибки во время отправки.

Pop-up — это элемент интерфейса в виде небольшого окно, которое появляется поверх основного содержимого веб-страницы.

SMS-шлюз — инфраструктурный компонент, обеспечивающий передачу SMS-сообщений от платформы Altcraft к операторам мобильной связи (или посредникам). Он обрабатывает маршрутизацию, форматирование, очереди отправки, отчёты о доставке и ошибки.

Захват событий — механизм фиксации и передачи (сбора) событий пользователя (например, просмотр страницы, добавление в корзину, клик по ссылке) в платформу Altcraft. Эти события используются для сегментации, персонализации, триггеров и аналитики.

Веб-версия письма — онлайн-страница, на которой отображается содержимое письма (HTML), аналогичного отправленному в email, но доступная через веб-браузер.

Внутренние термины Altcraft Platform

Профиль подписчика — объединяет информацию о получателях рассылок — клиентах, партнёрах, наблюдателях за рассылками

База профилей — определяет структуру основных и дополнительных полей данных профиля, а также возможность подписок на те или иные ресурсы.

Статический сегмент — фиксированный список профилей, сформированный с помощью определенных условий, или иными способами (вручную, с помощью сценария или API запроса).Статический сегмент изначально может быть рассчитан на выборке из базы данных, однако после сохранения он не будет пересчитываться автоматически.

Динамический сегмент — часть базы данных, которая включает профили, подходящие под определённые в сегменте условия.Такой сегмент не сохраняет список профилей, удовлетворивших условиям, вместо этого сохраняется сама структура запроса выборки.

Ресурс — средство для организации подписок клиентов на определённый контент по определённым каналам связи.

Подписка профиля — запись, объединяющая контактные данные по конкретному каналу связи и сведения о регистрации.

Рассылка — инструмент для организации отправок сообщений клиентам.

Броадкаст-рассылка — инструмент для организации массовых однократных рассылок контента подписчикам.

Регулярная рассылка — инструмент для организации массовых многократных рассылок контента подписчикам.

Триггерная рассылка — инструмент создания транзакционных сообщений в Altcraft Platform.

Мультивариантный тест — рассылка, в которой несколько (до 8) шаблонов сообщений отправляется части аудитории, эффективность этих вариантов сравнивается, и шаблон-победитель отправляется остальной части аудитории.

Журнал рассылок — инструмент для отслеживания всех этапов отправки рассылок и ошибок, возникших в процессе отправок.

Шаблон сообщения — контент, который Вы отправляете подписчикам в рамках одной или нескольких маркетинговых рассылок.

Сценарий — программы автоматизации маркетинговых процессов.

Отчёт по каналам — предоставляет данные по всем событиям, которые отслеживаются в рассылках: отправки, доставки, открытия, клики и др.

Отчёт по трафику — предоставляет данные о трафике профилей из таких источников как каналы, формы, пиксели и попапы.

Когортный отчет — отчет по группам подписчиков, привлечённым в определённый временной промежуток.

Время жизни (отчёт) — отчет, который показывает среднее время жизни подписчика в когорте.

Воронка конверсии — представляет собой последовательность шагов, по которым проходит группа ваших клиентов.

Цели (отчет) — отчёт о выполнении вашими клиентами целевых действий и о ценности их совершения.

Прирост аудитории (отчет) — отчёт для отслеживания динамики роста аудитории подписчиков на ваши ресурсы.

Карта кликов — отчёт по переходам ваших клиентов по ссылкам в отправленном Email сообщении.

Панель пользователя — интерфейс, в котором происходит работа пользователя с Altcraft Platform.

Панель администратора — служебный интерфейс, используемый администратором Altcraft Platform для настройки платформы

Тег — помечает сущности: базы данных, сегменты, шаблоны, импорты по расписанию, стоп-списки, рассылки, сценарии, формы, продукты, заказы и служит для навигации и улучшения поиска этих сущностей в платформе.

Подключение — механизм, позволяющий авторизоваться во внешних системах для расширения возможностей Altcraft Marketing функционалом этих систем. Обычно для создания такого подключения требуется аккаунт во внешней системе с правами доступа через API.

Политика отправки — настройки, ограничивающие количество отправляемых одному получателю сообщений. Политику отправки можно назначить для аккаунта, ресурса или рассылки.

Стоп-список — это список email адресов и доменов, а также телефонных номеров, которым вы не хотите отправлять сообщения. Стоп-лист фильтрует email-адреса и номера телефонов в момент отправки рассылки.

Трекинг домен — это домен для отслеживания действий подписчика. Он позволяет вам узнать, по каким ссылкам перешел получатель сообщения.

Пиксель — это HTML или JavaScript-код на сайте, позволяющий отслеживать поведение пользователей сайта.

База наблюдателей — база профилей, которые будут получать те же сообщения, что и профили, для контроля за ходом и результатом рассылки.

Маркет — функционал, позволяющий импортировать данные о товарах и заказах из различных маркетплейсов в платформу и использовать эти данные для коммуникации с клиентами.

Форма — это инструмент для сбора данных от ваших клиентов, с помощью которого можно реализовывать различные механизмы с использованием внешних данных, например, добавлять новые профили в вашу базу, управлять подписками на различные ресурсы и получать обратную связь от клиентов.

Менеджер тегов — это инструмент, который позволяет вам управлять HTML тегами и JavaScript кодом без необходимости привлекать разработчиков. С его помощью вы можете создавать, добавлять и обновлять фрагменты кода на сайте, чтобы отслеживать действия пользователей и передавать данные в платформу.

mSDK — это мобильный программный интерфейс (SDK) для интеграции push-уведомлений в мобильные приложения. Он помогает разработчикам легко подключить систему уведомлений Altcraft к своим Android- и iOS-приложениям через платформы Firebase, Huawei Mobile Service, RuStore и APNs.

Модуль Best Send Time — функционал, который автоматически рассчитывает оптимальное время отправки сообщения каждому пользователю на основе его поведения и исторических данных (активности, открытия, кликов). Позволяет повысить эффективность рассылок, минимизируя “время простоя” и увеличивая вовлеченность.

Менеджер подписок Altcraft — это страница, позволяющая подписчикам управлять своими данными профиля, а также подписками на ресурс: останавливать, возобновлять или приостанавливать их на определенный срок (14, 30, 60 или 90 дней). Если в вашей компании уже есть использующийся менеджер подписок, то его можно интегрировать в платформу.

Архив документации

С выходом новых версий Altcraft Platform представленная на сайте документация обновляется в соответствии с изменениями платформы. Для удобства пользователей, документация для неактуальных версий хранится в течение полугода после выхода следующей версии. Ниже находятся ссылки на пользовательскую документацию и API-справочник в формате PDF.

- **Версия 74** — PDF
- **Версия 73** — PDF
- **Версия 72** — PDF
- **Версия 71** — PDF

INFO: Важно

Вышеуказанные версии документации могут содержать фактические ошибки. Изменения в архивные версии **не вносятся**, они остаются в том же состоянии, в котором были зафиксированы.

Библиотека email-маркетолога

Опросили коллег, проанализировали 100+ инструментов для email-маркетинга и выбрали только лучшие. Разбили по разделам, чтобы вам было удобнее ориентироваться в нашем гайде. Здесь вы найдете всё, что может пригодиться в создании эффективной email-рассылки: от верстки мастер-шаблона до анализа результатов.

Сбор базы

Быстро и качественно собрать контактную базу можно при помощи формы подписки на сайте. Вот несколько хороших инструментов.

Инструмент	Описание
Hello Bar	Сервис для создания всплывающих окон на сайте. С помощью таких форм подписок можно собрать собственную базу email-адресов. Есть бесплатный тариф. Язык интерфейса — английский.
JumpOut	Российский сервис с продвинутым таргетированием поп-апов для сбора базы. Имеет множество интеграций с разными сервисами. Есть бесплатный тариф.
LPgenerator	Удобный инструмент для создания целевых страниц. Платформа имеет полный функционал по управлению лендингами в интернете.
Mailmunch	Англоязычный сервис создания поп-апов для сбора контактной базы. Гибкие настройки, хорошая аналитика и широкий функционал. Есть бесплатный тариф.
Sumo	Сервис для сборы email-адресов, установки форм и виджетов на сайте. В бесплатной версии доступно неограниченное количество адресов, кастомизация, таргетинг, аналитика и другие классные инструменты.
Tilda	Инструмент для создания целевых страниц: готовые дизайны, библиотека блоков, адаптивность и многое другое.

Проверка базы

Контактную базу необходимо постоянно чистить от дублирующихся, несуществующих, неправильно записанных адресов и спам ловушек. Существует множество сервисов, которые справятся с этим. Почти все из них платные, но у каждого есть пробный период или фиксированное количество бесплатных проверок. Мы собрали только самые лучшие

😉

Инструмент	Описание
BrightVerify	Надежный англоязычный сервис для проверки email-адресов на валидность.
Mailvalidator	Мощный инструмент для валидации email-адресов, который интегрируется с множеством маркетинговых платформ.
Mailfloss	Этот сервис автоматически удалит невалидные email-адреса в вашей базе. Достаточно просто подключить его к платформе.

Инструмент	Описание
QuickEmailVerification	Онлайн-платформа для валидации email-адресов. В день можно проверять до 100 контактов.
Zerobounce	Доступна бесплатная проверка 100 адресов.

Инструменты для работы с изображениями

Разбили на подразделы. Здесь вы найдете бесплатные фотостоки с классными изображениями, удобные программы для скриншотов и графические редакторы, которые сделают вашу работу чуточку проще.

Фотостоки:

- Unsplash
- Pixabay
- Pexels
- Depositphotos
- ByPeople
- IRA Design
- Illlustrations
- 365PSD

Работа со скриншотами:

- Licecap
- LightShot
- Monosnap.com

Графические редакторы/иллюстраторы

Векторные

- Figma
- Sketch
- Adobe Illustrator

Растровые

- Adobe Photoshop

Как графические редакторы, но проще. Плюс сервисы по работе с анимациями

- VistaCreate

- Canva
- Animaker
- Giphy
- Gifovina

Инструменты для работы с шаблонами писем

В этом разделе сделали подборку из нескольких сервисов для создания профессиональных и адаптивных шаблонов email-писем. Знания HTML-не требуются.

Инструмент	Описание
Altcraft Platform	В нашу платформу уже встроен онлайн-редактор сообщений. В нем можно готовить шаблоны для всех каналов коммуникаций: email, sms, mobile/web push.
Beefree	Тоже онлайн-редактор шаблонов писем. Язык интерфейса - английский.
Stripo.email	Онлайн-сервис, с помощью которого можно создавать профессиональные и адаптивные шаблоны писем. Есть интеграция со всеми популярными ESP.

Тестирование email-рассылок

Чтобы ваши подписчики увидели такое же красивое письмо, как и вы, его необходимо проверить перед отправкой. Вот несколько сервисов для этого.

Инструмент	Описание
Email on Acid	Здесь вы сможете посмотреть, как ваше письмо будет отображаться на 70-ти почтовых сервисах и устройствах. Бонус: можете пройдёт ли ваша рассылка спам-фильтры.
FreeFormatter	Сервис для проверки кода HTML, Json на наличие ошибок.
Litmus	Показывает, как на разных устройствах и почтовых сервисах будет выглядеть ваше письмо. Единственный его минус - сервис платный.
PutsMail	Валидация HTML-кода письма.

Проверка эффективности темы

Тестируйте темы писем и выясняйте их гипотетическую привлекательность для клиента при помощи сервисов Email Subject Line Tester и Subjectline.

Проверяем синтаксис записей DNS и SPF

Перед запуском рассылки проверяем синтаксис DNS и SPF записей.

Инструмент	Описание
Dnsquery.org	Проверит, доступны ли записи DNS почтовым провайдерам.
Spf-policy-tester	Здесь вы можете проверить синтаксис SPF-записи домена.

Инструменты для тестирования доставляемости

Инструмент	Описание
Mail-tester	Сервис прогоняет ваше письмо через алгоритм SpamAssasin, который проверяет на “спамность”. Проверяет синтаксис записей SPF, DKIM, ADSP, DMARC.
SendForensics	Сервис с подробным отчетом: оценивает доставляемость на разные почтовики, проверяет контент и верстку на наличие ошибок, репутацию домена в динамике, соответствие GDPR и многое другое. Правда, такой глубокий анализ доступен только в платной версии.
Glock Apps	Сервис проверяет содержимое письма на спам, а также показывает доставляемость письма на разные почтовые сервисы. Сервис полностью бесплатный.

Постмастеры

Постмастеры отслеживают результаты email-рассылок: сколько было доставлено, сколько попало в спам, на какое количество писем пожаловались и т.д. Регистрация постмастера обязательна, если вы планируете доставлять большие объемы писем: больше 10 тысяч в сутки.

ISP	Ссылка
AOL + Yahoo	https://postmaster.yahoo.com
Comcast	https://postmaster.comcast.net
Facebook*	https://www.facebook.com/postmaster/index/
Google	https://gmail.com/postmaster
Mail.ru	https://postmaster.mail.ru
Outlook	https://sendersupport.olc.protection.outlook.com/pm/postmaster.aspx

Отслеживание действий на сайте

Для персонализации своих email-рассылок вам важно знать онлайн-поведение пользователя и его предпочтения. Эти данные также нужны для правильной автоматизации маркетинга.

Инструмент	Описание
Altcraft Pixel	HTML или JavaScript-код, который отслеживает действия пользователя на сайте. С его помощью можно получить отчет по целевым действиям ваших клиентов.

Инструмент	Описание
Leadfeeder	С бесплатным тарифом вы можете отслеживать представителей разных компаний, которые заходят на ваш сайт.
Page Analytics	Простое расширение для Chrome, которое помогает увидеть взаимодействия пользователей с вашим сайтом.
SimilarWeb	Онлайн-сервис, который позволяет получать наиболее важные сведения о пользователях на сайте.

Опросы для рассылок

Узнавайте напрямую у подписчиков их отношение к бренду через опросы в теле письма. Собрали несколько подходящих для этого сервисов.

Инструмент	Описание
Altcraft Forms	В нашей платформе есть удобный инструмент для создания форм и опросов. Создайте форму в конструкторе, а затем разместите её на странице сайта или поделитесь ссылкой на форму с подписчиками.
Anketolog	https://anketolog.ru/
Google Forms	https://www.google.com/forms/about/
Microsoft Forms	https://forms.office.com/
SurveyMonkey	https://ru.surveymonkey.com/
Survio	https://www.survio.com/ru/
Typeform	https://www.typeform.com/

Для вдохновения

Выгорание может настигнуть каждого. Сохраняйте к себе в закладки эти сервисы, чтобы быть готовыми к нему.

Инструмент	Описание
Коллекция Stripo.email	Дизайны email-рассылок на любую тему.
Email-competitors	Коллекция рассылок. Будьте в тренде!
Pinterest	Здесь вы можете найти доски с email-рассылками на различные тематики или просто поискать вдохновение в красивых картинках.

Инструмент	Описание
Really Good Emails	Еще одна библиотека с красивыми и продающими email-письмами.

Полезные ресурсы для обучения

Ресурс	Описание
Altcraft блог	Уверены, что у нас в блоге вы найдете много полезных и интересных материалов по автоматизации маркетинга и не только.
Блог агентства EmailSoldiers	Интересно и без воды рассказывают об email и контент-маркетинге.
Блог агентства EMAILMATRIX	Нескучные и полезные статьи об email-маркетинге, дизайне и вёрстке писем, контент-маркетинге, аналитике и многом другом. А еще ребята в формате заметок делятся полезными сервисами, лайфхаками в маркетинге и подборками интересных книг.
Блог Texterra о маркетинге	Тут вы не только прокачаете свои знания в области email-маркетинга, но и расширите границы в других направлениях интернет-маркетинга. Удобная система фильтрации на сайте.
Дэн Кеннеди “Продающее письмо”	О том, как создавать продающее письмо, можно узнать в этой книге. Здесь упор идет на написание понятного и работающего текста. От идеи до шедевра рекламы вы вместе с автором пройдете 28 шагов.
Дмитрий Кот “E-mail-маркетинг. Исчерпывающее руководство”	Полезная книга для новичков в email-маркетинге.
Леонид Николаев “Доставка в Inbox - уровень Бог”	Если вы уже разбираетесь в том, какой путь проходит рассылка, то советуем прочитать эту книгу.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Обновления платформы

INFO: **Политика обновлений платформы**

Обновления распространяются среди пользователей с учетом их запросов. В первую очередь они становятся доступными тем, кто запрашивал новый функционал. На этом этапе проводится оценка работы платформы и сбор обратной связи. После этого обновление становится доступным для остальных.

После получения новой сборки рекомендуется выполнить обновление в течение 14 календарных дней. В обновления включены не только новые функции, но и важные исправления безопасности, а также устранение выявленных ошибок.

Исправления и обновления выпускаются только для актуальной версии платформы. Поддержка устаревших версий не осуществляется. Ошибки, найденные в версиях, отстающих более чем на два мажорных выпуска, не рассматриваются и не исправляются. Чтобы обеспечить стабильную и безопасную работу, важно своевременно обновляться.

TIP

Номер версии платформы указывает на дату выпуска и объем изменений. Например, версия **v2024.3.69.490** означает, что обновление было выпущено в **3-м квартале 2024 года**, это **69-я версия платформы**, и она включает **490 исправлений багов и улучшений**.

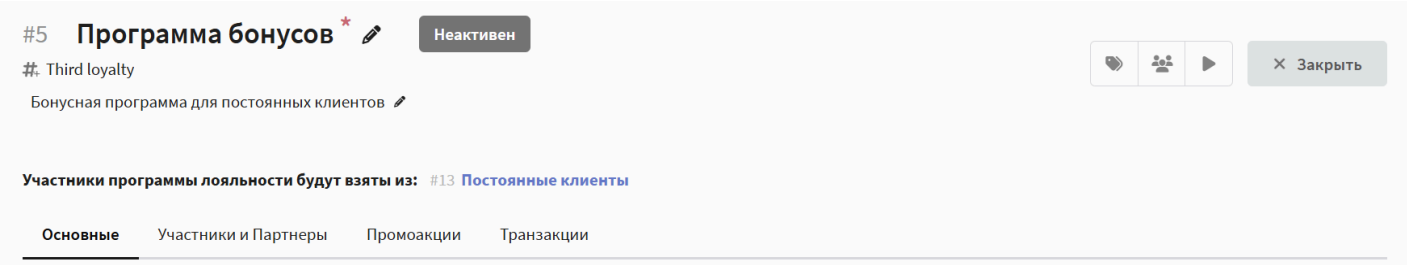
v2025.4.75

v2025.4.75.0 – Loyalty Programs, New segmentation, UI Update

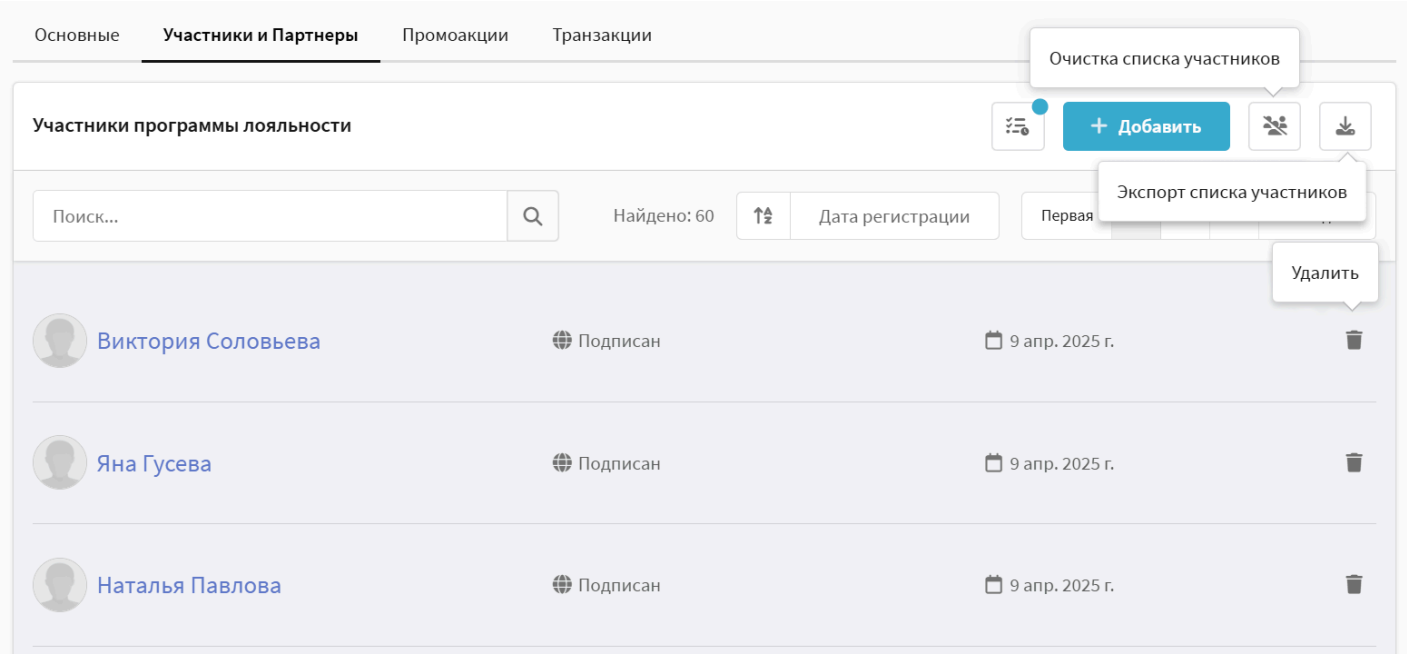
> **CAUTION**
>
> Для корректной работы платформы требуется провести **миграцию**. Для этого:
>
> 1. Замените файлы платформы на файлы из новой сборки;
> 2. Запустите `./akd upgrade` и согласитесь с выполнением миграции;
> 3. Полностью перезапустите платформу командой `./akd restart`.
>
> Также необходимо добавить следующие обязательные объекты в файл конфигурации `main.json`:
>
> - `SYSTEM_REDIS` — основная конфигурация Redis-хранилища
> - `LOYALTY_CACHE` — конфигурация Redis-хранилища, используемого для работы программ лояльности
> - `PROCWORKFLOW_DISTRIBUTED_CACHE_DB` — параметры распределённого кэш-хранилища, используемого процессом `procworkflow`
> - `CAMP_DUPLICATESDB` — конфигурация Redis для базы дубликатов
>
> Подробнее об этих параметрах конфигурации можно прочесть в документации для администраторов.

Программы лояльности

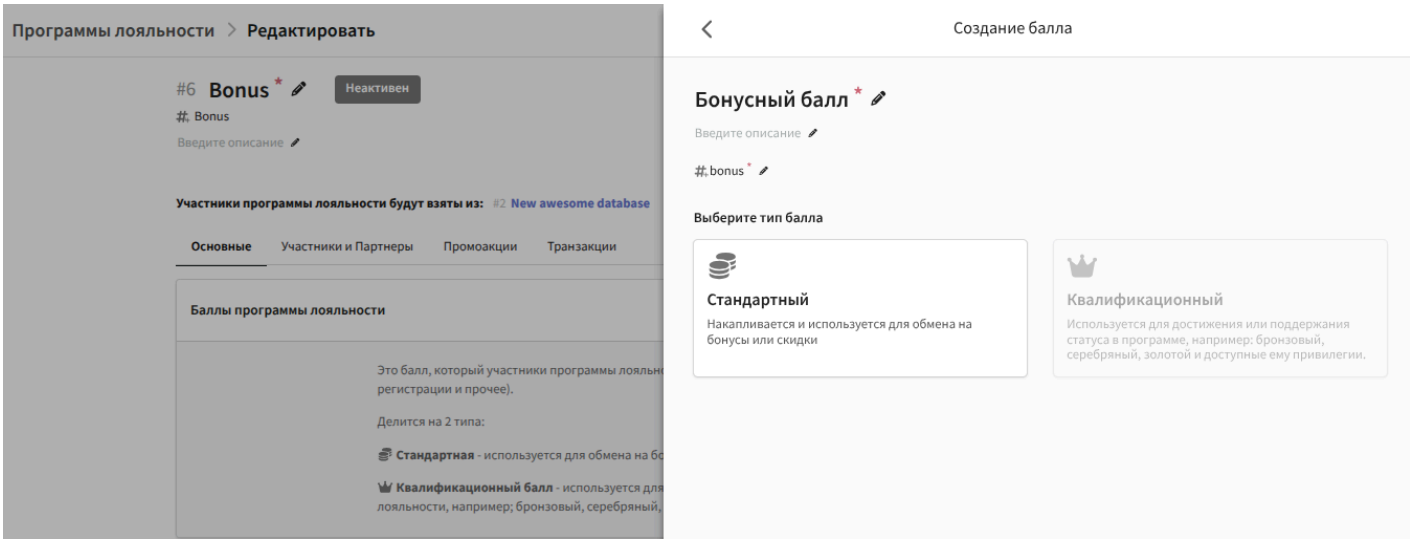
В этом обновлении в платформу Altcraft добавлен модуль “Программы лояльности”. **Программа лояльности** — это система поощрения клиентов за целевые действия: покупки, регистрацию, просмотр контента и другое. Клиенты получают баллы, которые могут потратить на скидки, фильмы, тарифы и услуги.



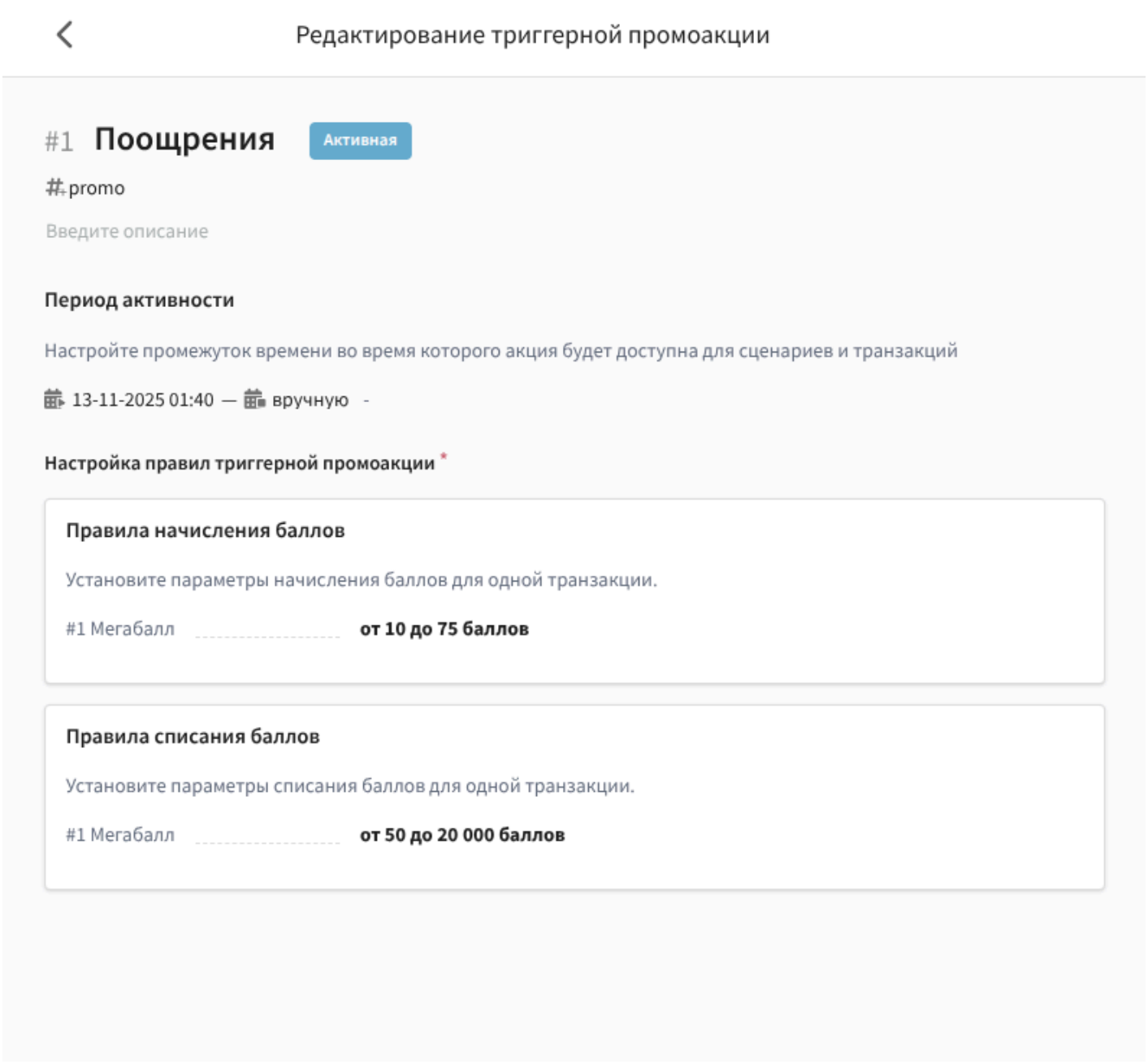
Участниками программы лояльности могут стать все профили в базе или же определенный сегмент. Также можно добавлять участников вручную или при помощи API-запросов:



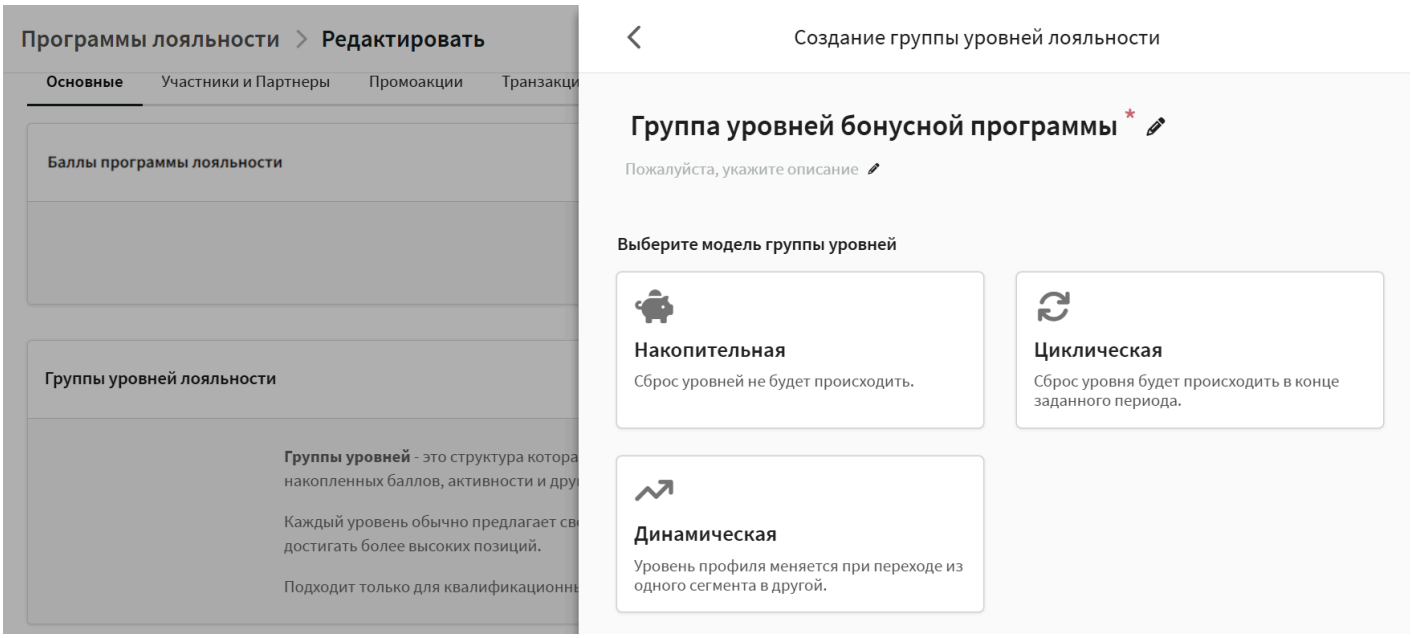
Участникам программ лояльности можно начислять **баллы**. Баллы делятся на стандартные и квалификационные. Стандартные используются для вознаграждений, квалификационные — для определения уровня участника. Каждому типу баллов можно задать эквивалент в реальной валюте:



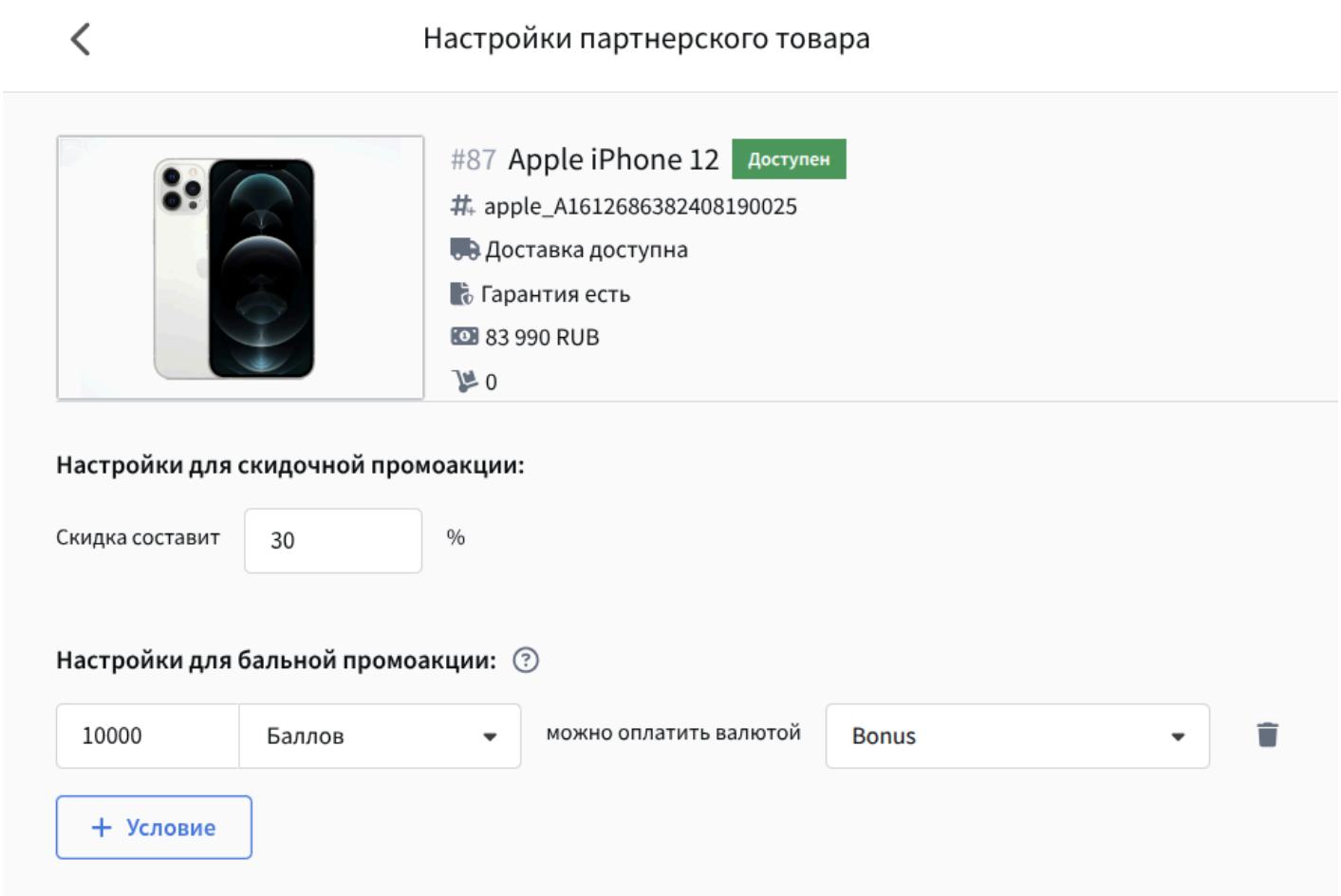
Управление начислениями и списаниями баллов осуществляется при помощи **триггерных промоакций**:



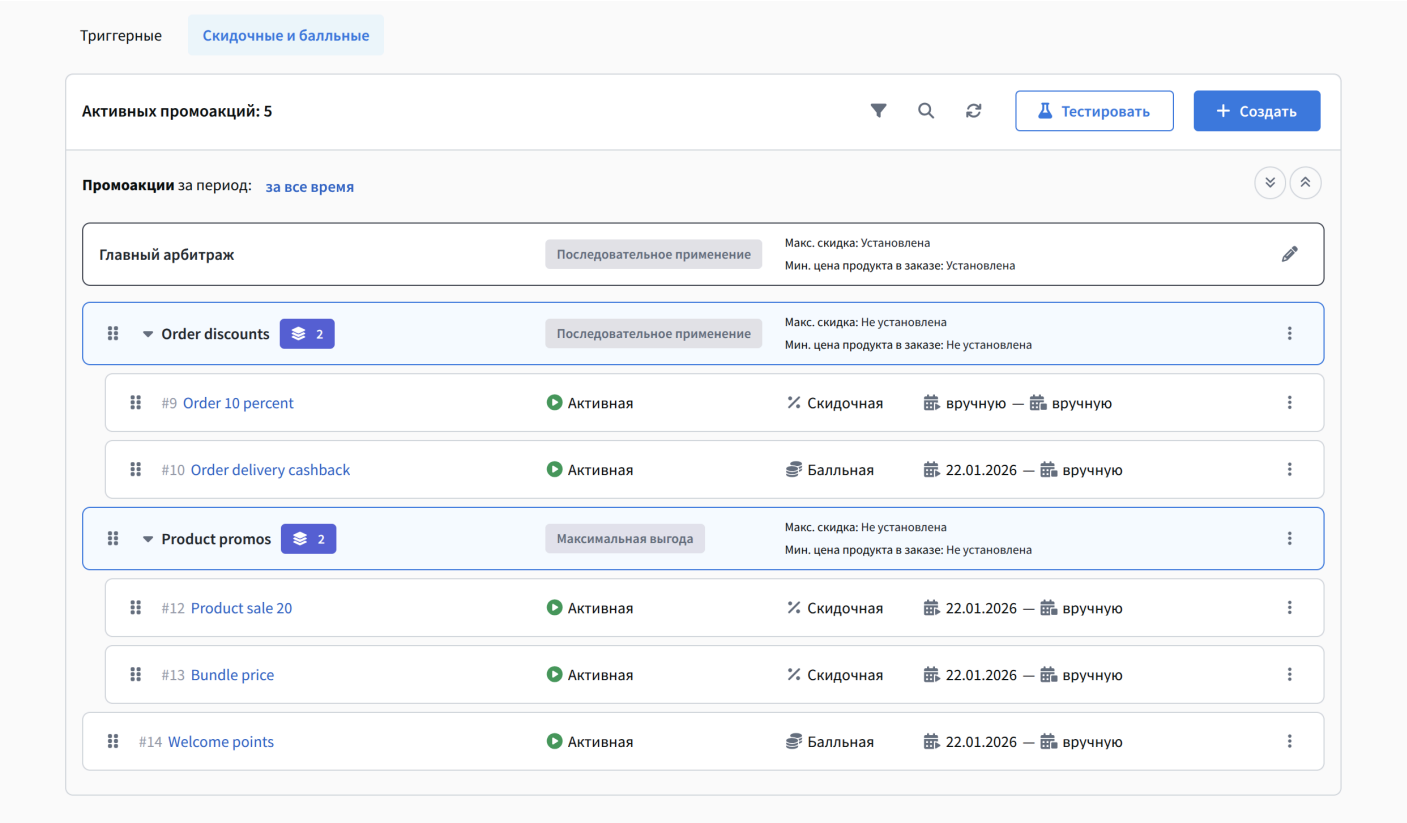
Для **уровней лояльности** настраиваются группы со своими правилами пересмотра статуса и сброса баллов. Для участников на каждом уровне можно настроить разные механики поощрения и стимуляции покупок:



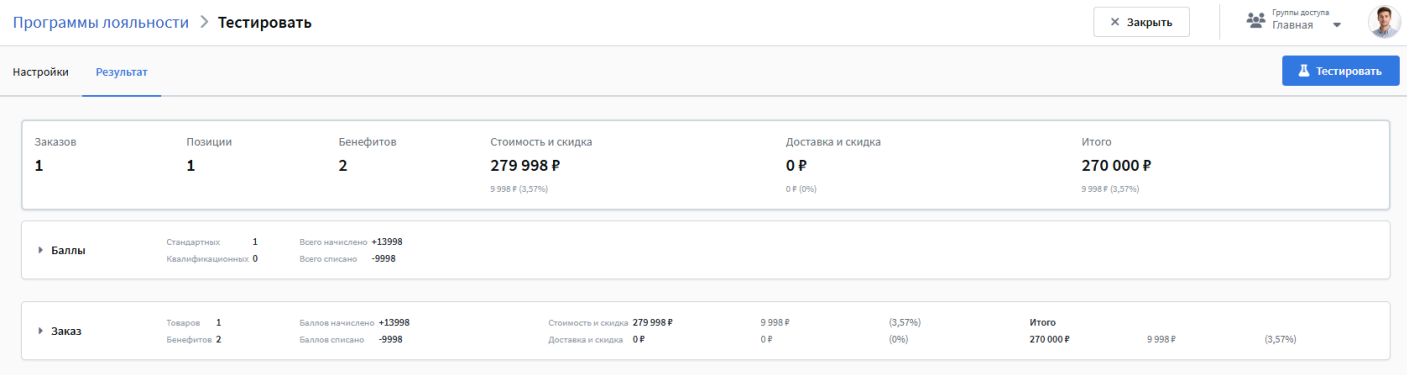
Пользователи могут устанавливать **партнеров** программы лояльности. Для партнерских товаров из Маркета настраиваются специальные предложения:



А **скидочные** и **балльные промоакции** отвечают за поощрения при заказе продуктов из Маркета. Пользователь может создать сразу несколько подобных акций и тонко настроить их взаимодействие при помощи **арбитража**:



После настройки баллов и промоакций вы можете проверить корректность их работы. Для этого существует функция **тестирования промоакций**. Вы можете провести предварительный расчет заказа как для действительного профиля в базе, так и для тестового участника, контекст которому вы укажете самостоятельно:



Тестирование позволит вам точно понять, как и в каком порядке для участников лояльности сработают бенефиты.

Подробнее о том, как создать и настроить программы лояльности читайте здесь.

Обновленная сегментация

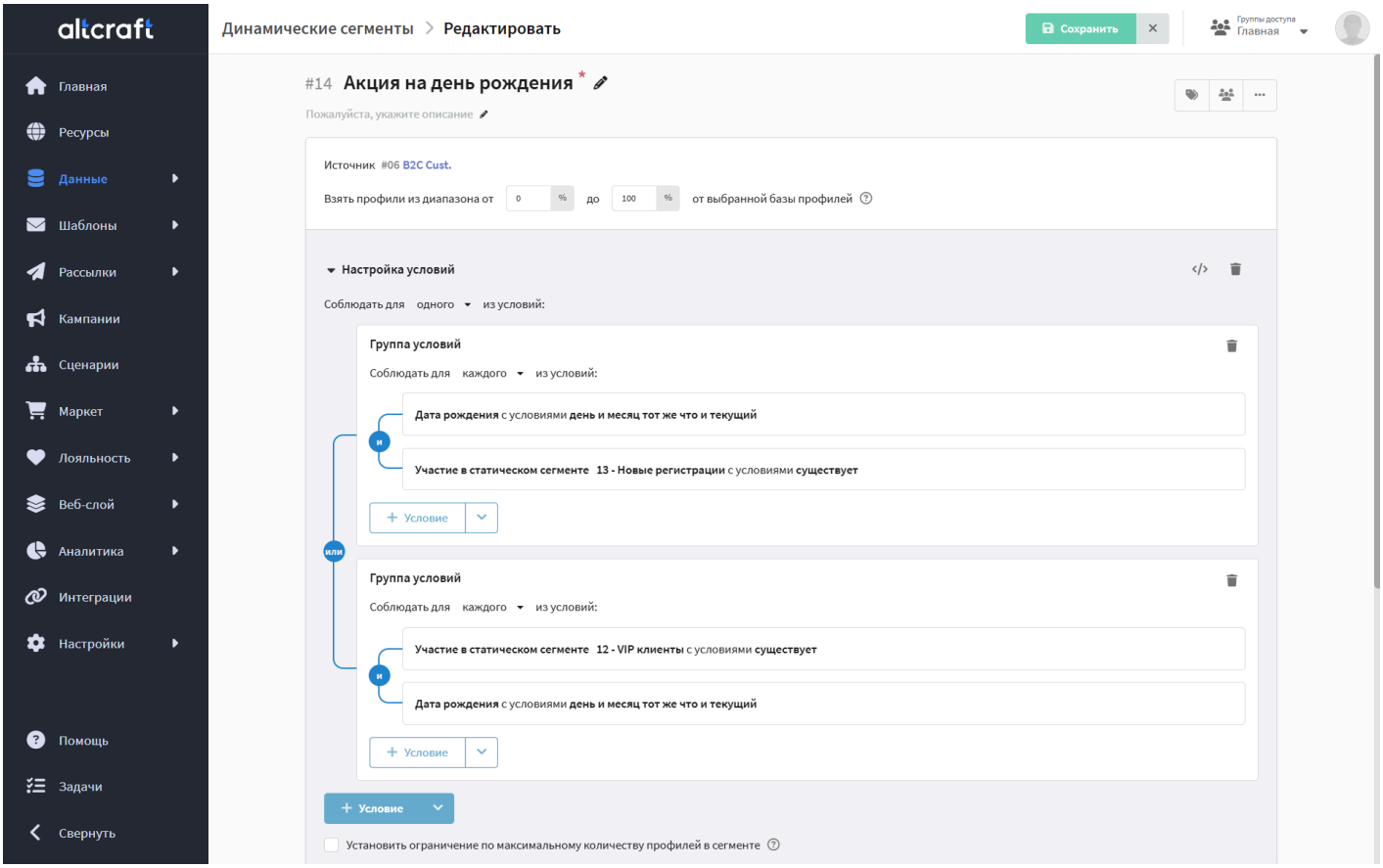
Крупные изменения внесены в механизм сегментации профилей в платформе. Главное нововведение — переработка типов сегмента:

- **Статический сегмент** — представляет собой сущность для хранения профилей, заполняемую вручную или по API. Сегмент не пересчитывается при использовании и автоматически не обновляется. Сегмент можно сохранить пустым, главное вписать имя и выбрать БД.
- **Динамический сегмент** — этот сегмент по сути является набором условий, а не списком профилей. Динамический сегмент просчитывается перед каждым новым использованием. Он также может показать

результаты выборки без сохранения сегмента.

- **Обновляемый сегмент** — такой сегмент нельзя наполнять вручную или по API. Сегмент наполняется по заданным условиям согласно установленному расписанию или при ручном пересчете.
- Быстрые сегменты были убраны из платформы. Вместо них применяется динамический сегмент, для этого при настройке сегмента нажмите кнопку “Рассчитать”.

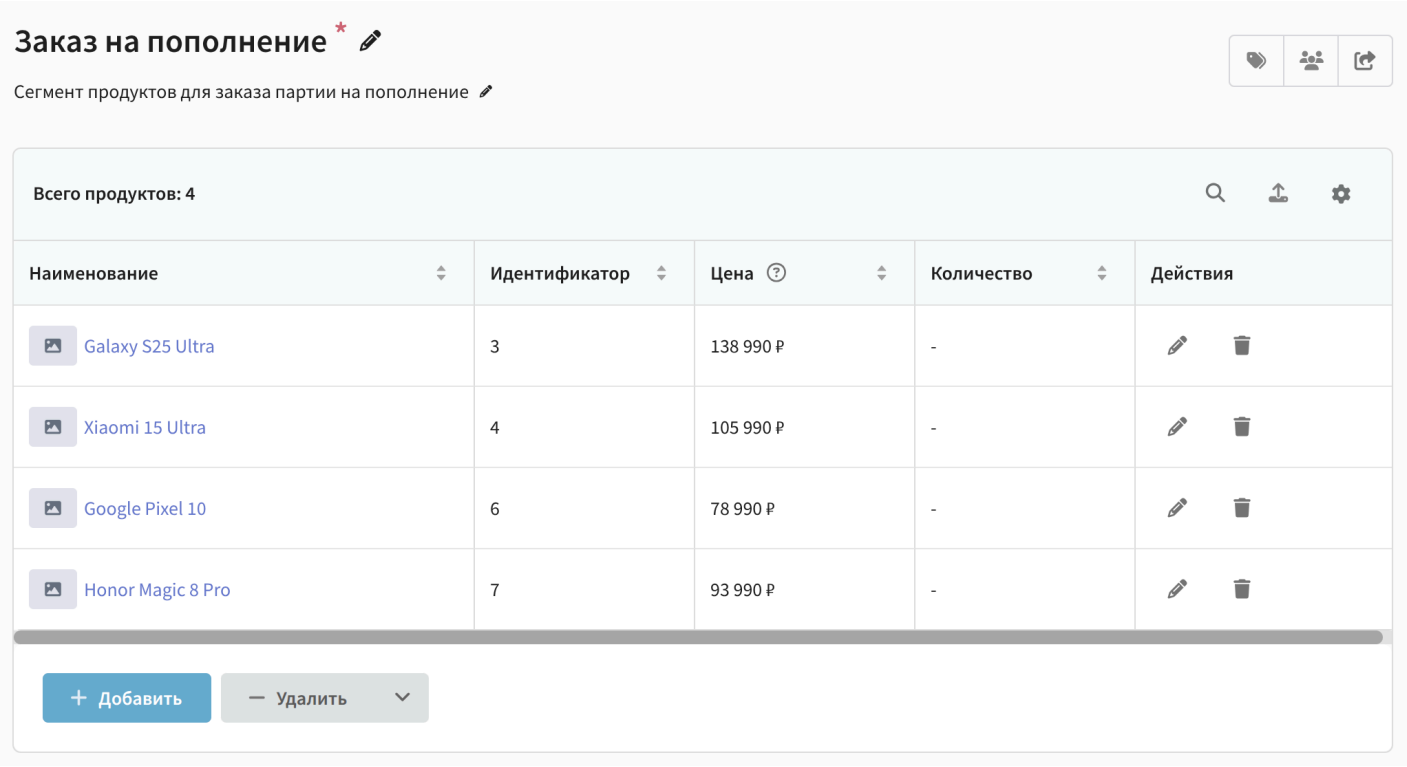
Также существенно переработан внешний вид меню настройки сегментов:



Интерфейс стал более современным и интуитивно понятным. Более подробная информация по сегментации доступна в этом разделе документации.

Продуктовые сегменты

Теперь в рамках Маркета вы можете объединять продукты в **продуктовые сегменты**. Это наборы продуктов и их модификаторов (SKU), объединенных в группу по определенному признаку:



Сегменты бывают двух видов: **статические** (объединенные через API или вручную при помощи условий отбора) и **обновляемые** (пересчитываемые автоматически по расписанию). Готовый сегмент можно выгрузить и использовать для анализа данных о ваших продажах.

Инструкция по настройке сегментов продуктов доступна в этой статье документации по Маркету.

Обновления интерфейса

Вместе с обновлением мы улучшили интерфейс:

- Обновлена цветовая палитра элементов платформы;
- Изменили внешний вид некоторых компонентов, таких как выбор диапазона дат;
- Улучшили отображение боковых вспомогательных меню (например, стратификации в ГКГ);
- Теги и атрибуты в списках теперь выглядят органичнее и не выбиваются из общего дизайна;
- Исправлены некоторые ошибки, связанные с отображением таблиц.

Цветовая палитра

До: После:

Диапазоны дат

До: После:

Отображение в списках

До: После:

Подобные небольшие улучшения повышают удобство пользования платформой, делают интерфейс более цельным и современным, снижая визуальный шум

Интеграция с Keycloak

В этом обновлении добавлена возможность настроить интеграцию платформы с **Keycloak**. Это дает администраторам платформы возможность настраивать систему доступа с помощью единой централизованной аутентификации, а также назначать группы и роли в соответствии с настройками Keycloak.

Создание коннектора и настройка доступов производится через Панель администратора.

Прочее

- Изменения в настройках виртуального сендера снова сохраняются корректно.
- Устранена ошибка, вследствие которой параметр `CLICKHOUSE_SYSTEM.CONN_TIMEOUT` действовал на общий таймаут подключения к хостам, а не на конкретную попытку.
- ГКГ более не вызывает ошибки при работе после ее повторного создания.
- Исправлена ошибка, из-за которой обязательное поле пользовательского канала могло становиться необязательным.
- Email-адрес теперь корректно приводится к нижнему регистру при создании LDAP-пользователя.
- Устранена ошибка, при которой регулярное выражение для массивов проверяло только первый объект массива и не давало пройти валидацию в JSON-фильтре.
- Превью шаблона для Telegram в сценариях теперь отображается корректно.
- Улучшено отображение ошибок и предупреждений в платформе.
- Письма-дубликаты с предупреждением о достижении лимита по импорту профилей больше не генерируются.
- Устранена ошибка, при которой некоторые сценарии не могли сохраняться в платформе.
- Оптимизирована скорость импорта профилей в платформу.
- Удаленные пользовательские события теперь не отображаются в списке захвата событий и не активируют триггеры.

References

v2025.4.75.0: PL-20429 PL-20573 PL-20313 PL-20551 PL-20498 PL-20474 PL-20485 PL-20095 PL-19769 PL-20192 PL-19817 PL-16969 PL-20344 PL-20141 PL-20057 PL-19157 PL-19942 PL-20372 PL-20306 PL-17956 PL-20270 PL-20273 PL-19106 PL-20266 PL-19173 PL-19937 PL-19087 PL-18576 PL-20170 PL-19912 PL-19289 PL-19935 PL-19938 PL-19762 PL-20032 PL-20060 PL-19785 PL-19941 PL-18104 PL-18357 PL-19455 PL-18995 PL-19781 PL-19639 PL-17787 PL-18173 PL-19851 PL-19548 PL-17393 PL-18637 PL-19101 PL-19064 PL-17906

---

v2025.4.75.1005 – Loyalty fixes, minor improvements

Улучшено

- Добавлена возможность ожидания определённой даты в узле “Ожидание события”.
- Добавлена валидация наличия шаблона при запуске рассылки, а также запрет на удаление шаблона, привязанного к рассылке.

Прочее

В этом релизе были исправлены ошибки в работе модуля программы лояльности, а также устранен ряд других ошибок.

- Создание рассылки не мастер-пользователем теперь не вызывает ошибок.
- Устранена ошибка, при которой некоторые сценарии не прикреплялись к кампаниям.
- Ограничение на повторное заполнение формы снова работает корректно

- Добавлена валидация и наследование параметров конфигурации, отвечающих за Redis-хранилище.
- Исправлены ошибки в сборе и отображении аналитики программ лояльности.
- События в журнале кампаний теперь отображаются корректно.
- Улучшена стабильность процесса `webcontrol`.
- Добавлена фильтрация по статусам в списках кампаний.
- Устранена ошибка, возникавшая при попытке сохранить кампанию.
- Платформа теперь отображает корректную информацию об ошибке при проблемах с подключением к Yandex.Audience.
- Переход к шаблону из превью в списке шаблонов теперь работает корректно.
- Устранена ошибка, из-за которой при экспорте данных формы не передавался ID профиля.
- Теперь время в узле сценария “Пауза” изменяется корректно.
- Устранена ошибка, не позволявшая вставить значение менее 100% в правилах виртуальных сендеров.
- Фильтрация по группам в списке программ лояльности снова работает корректно.
- Исправлены ошибки, возникавшие при открытии email-шаблона.
- Деактивация рассылки во время отправки снова работает корректно.
- Исправлены некоторые ошибки в работе визуального редактора.
- Превью AMP-версии email-шаблона теперь открывается и отображается корректно.
- Устранена ошибка, при которой в событиях жалоб не указывался провайдер.
- Дата в захвате событий и API снова передаётся корректно.
- Исправлена ошибка, из-за которой превью в пользовательских каналах всегда отображалось как HTML-сообщение.
- Переменная `subscription_fields` теперь корректно передается в тестовых отправках.
- Расчёт быстрого сегмента с запросом к внешней БД снова работает корректно.
- Добавлена валидация на привязку к несуществующей в платформе группе.

References

v2025.4.75.1005: PL-20912 PL-20916 PL-20899 PL-20826 PL-20881 PL-20624 PL-20847 PL-20791 PL-20794 PL-20752 PL-20756 PL-20753 PL-20767 PL-19819 PL-20760 PL-20205 PL-20757 PL-20744 PL-20754 PL-20749 PL-20718 PL-20644 PL-20724 PL-20730 PL-20732 PL-20725 PL-20041 PL-20469 PL-20705 PL-20632 PL-20704 PL-20552 PL-20702 PL-20635 PL-20683 PL-20693 PL-20684 PL-20190 PL-20618 PL-19500 PL-20380 PL-20674 PL-20099 PL-20669 PL-20651 PL-20671 PL-20648 PL-20668 PL-20666 PL-20620 PL-20665 PL-17606 PL-20659 PL-20650 PL-20069 PL-20653 PL-20649 PL-20647 PL-20637 PL-19347 PL-20233 PL-20148 PL-20625 PL-20606 PL-20314 PL-19211 PL-20523 PL-20572 PL-20534 PL-19602 PL-20500 PL-18808 PL-20376 PL-20169 PL-20377 PL-20073 PL-19950 PL-19940 PL-19822 PL-18473 PL-19411 PL-18360 PL-19802 PL-19688 PL-18742 PL-19370 PL-19619 PL-16670 PL-18771 PL-19338 PL-19174 PL-18741 PL-18918 PL-16688 PL-16687

---

v2025.4.75.2408 – React Native mSDK, product segment improvements

DANGER: Миграция

Если вы **уже обновлялись до версии 75,** то для корректной работы статических сегментов с этим релизом требуется выполнить **миграцию**. Для этого:

1. Остановите процессы `webcontrol`, `api` и `proctask`.
2. Запустите миграцию командой `./aktool segments run-separation-migration --config=путь-к-файлу-конфигурации`.

3. Запустите остановленные процессы.

mSDK Altcraft для React Native

В мобильном SDK Altcraft была добавлена поддержка кроссплатформенного фреймворка **React Native**. Его применение позволит использовать SDK одновременно для iOS- и Android-приложений.

Подробнее узнать о подключении и использовании mSDK Altcraft на React Native можно здесь. NPM-пакет доступен по этой ссылке.

Улучшено

- Для продуктовых сегментов добавлен фильтр по участию в сегментах, а также добавлена возможность импортировать условия из другого сегмента.
- При создании и редактировании шаблона и фрагмента шаблона в интерфейсе теперь отображается список доступных атрибутов.
- В API-запросы, передающие данные о рассылках, добавлена информация о привязке к кампании.
- В мобильном SDK добавлена возможность обновления полей профиля.
- Обновление аудитории сегмента теперь блокируется, если он используется в качестве основного сегмента кампании.
- Добавлено уведомление о необходимости завершить кампанию перед её удалением.
- Добавлена проверка подключения к Redis-хранилищу при запуске платформы.

Прочее

- Атрибуты теперь корректно отображаются в предпросмотре шаблона рассылки.
- JSON из полей Объект и Массив объектов в узле API-запроса снова подставляется корректно.
- Устранены ошибки с Viber-отправками через SMSTraffic, возникавшие из-за UTC-времени в параметре `start_date`.
- Улучшены сбор и отображение отчётности по программам лояльности.
- Фильтр по типу событий в карточке профиля снова работает корректно.
- В email о заполнениях формы больше не содержатся недопустимые спецсимволы.
- Исправлены некоторые ошибки, связанные с атрибутами.
- Лимит в пайпе `http_request` пользовательских каналов теперь работает корректно.
- Устранены ошибки при клонировании шаблона пользовательских каналов.
- Условие “существует” по строковым полям в быстром сегменте снова работает корректно.
- Улучшена стабильность процессов `webadmin`, `proctask` и `api`.
- Увеличена скорость разбора очереди `trk_pixel_def`.
- Ссылки в push-шаблонах в бета-редакторе теперь работают корректно.
- Устранены проблемы с подключением к платформе по LDAP.
- Исправлена ошибка, из-за которой превью шаблонов не сохранялись в FileDB.
- Обработка пользовательского поля с названием системного поля теперь проходит корректно.
- Пользовательские каналы снова могут использоваться в сценариях.
- Устранена ошибка, при которой выгрузка журнала аудита содержала время, указанное не в часовом поясе аккаунта.
- Теперь UTM-параметр `utm_medium` обновляется в карточке профиля при регистрации цели по API.
- Исправлена ошибка, из-за которой не регистрировались hard bounce из рассылок Telegram-бота.
- Политики отправки теперь применяются при массовой триггерной отправке профилей в сценарий.
- Теперь ограничение по количеству символов не будет учитываться при использовании динамических переменных с длинными логическими условиями.

- Устранена ошибка, при которой API-метод `/profiles/list` возвращал не все профили из базы.
- Исправлены ошибки отображения значения External object в шаблонах пользовательских каналов.

References

v2025.4.75.2408: PL-20956 PL-21090 PL-21117 PL-21116 PL-21118 PL-21115 PL-19263 PL-21087 PL-21089 PL-18153 PL-21062 PL-21046 PL-17906 PL-20943 PL-21038 PL-20987 PL-21023 PL-20499 PL-20999 PL-20641 PL-21008 PL-20971 PL-20986 PL-20636 PL-20542 PL-20865 PL-18467 PL-20802 PL-20846 PL-20780 PL-18407 PL-17935 PL-19747 PL-20682 PL-20719 PL-20652 PL-20312 PL-18757 PL-18575 PL-18663 PL-21027 PL-21009 PL-20990 PL-18111 PL-20914 PL-20980 PL-21025 PL-20968 PL-20970 PL-21002 PL-21005 PL-20984 PL-20860 PL-20833 PL-20966 PL-20931 PL-20953 PL-17662 PL-20896 PL-20875 PL-20862 PL-20541 PL-20764 PL-20738 PL-20553 PL-19277 PL-20428 PL-19278 PL-20201 PL-17754 PL-18981 PL-21018 PL-20992 PL-20815 PL-20967 PL-20963 PL-20456 PL-20962 PL-20934 PL-20893 PL-20825 PL-20540 PL-20588 PL-20236 PL-18822 PL-20901 PL-20525 PL-19672 PL-20751 PL-17954 PL-20980 PL-20965 PL-20891 PL-20954 PL-20958 PL-20951 PL-20908 PL-20955 PL-20927 PL-20935 PL-20922 PL-20297 PL-2087 PL-20879 PL-20886 PL-20866 PL-20830 PL-20804 PL-20831 PL-20761 PL-20814 PL-20789 PL-20720 PL-19344 PL-20808 PL-20714 PL-20803 PL-20707 PL-20790 PL-20786 PL-20759 PL-20433 PL-20539 PL-20688 PL-20709 PL-20685 PL-18299 PL-20712 PL-20623 PL-20408 PL-20164 PL-20677 PL-20160 PL-20594 PL-20622 PL-20056 PL-20247 PL-18498 PL-16893

---

v2025.4.75.2774 – Custom Authentication, block editor improvements

CAUTION: Запрет на отправку без использования ресурса

В ближайших обновлениях все активные рассылки, которые не используют ресурс, **перестанут работать**.

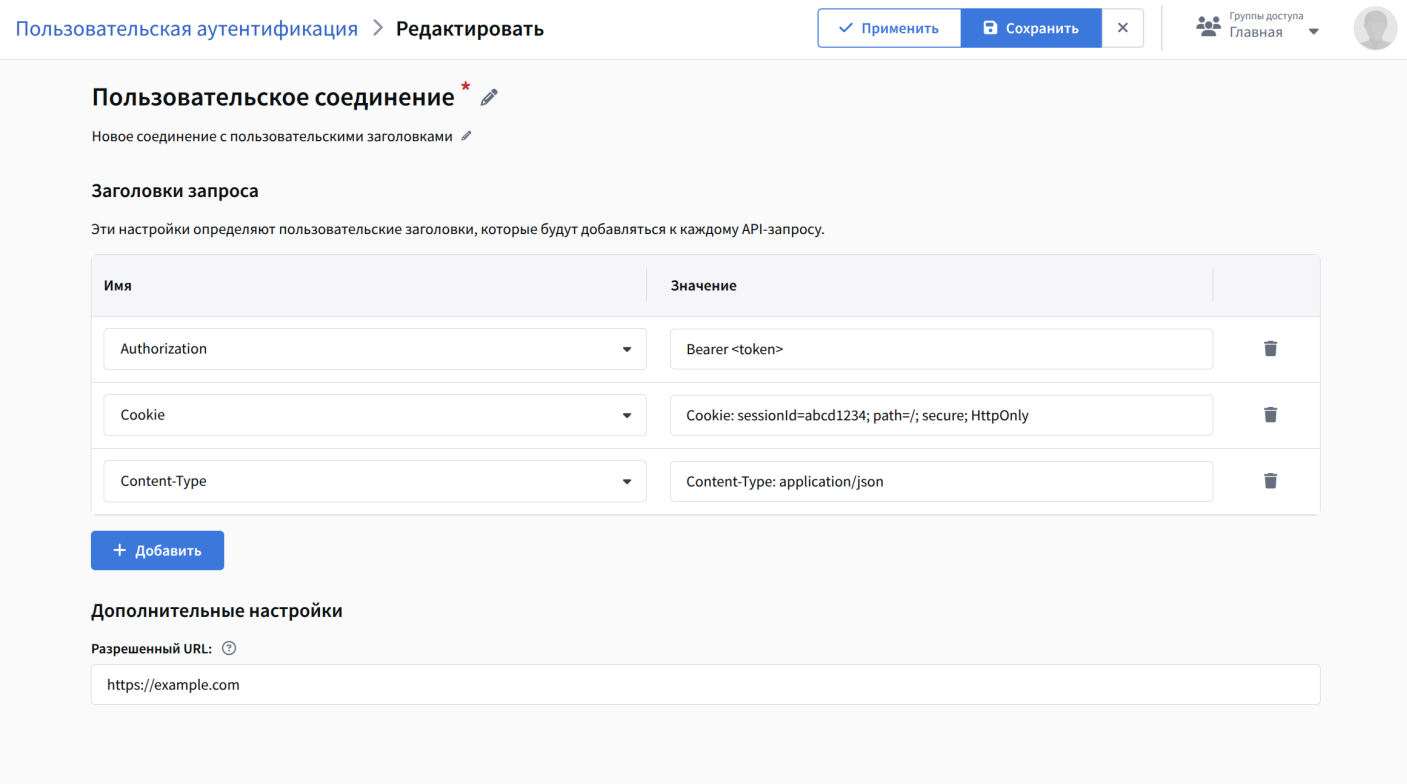
Чтобы рассылки могли корректно работать дальше, создайте новый или добавьте существующий **ресурс** и добавьте его в настройках рассылки.

Список таких рассылок можно получить с помощью команды `./aktool campaigns upgrade-blocker-list` или `./aktool campaigns upgrade-blocker-list --account-id 12345` (для конкретного аккаунта).

Добавлено

Добавлен новый вид коннекторов для внешних подключений — **Custom Authentication**. Такое подключение позволяет настроить пользовательские HTTP-заголовки для API-запросов, включая предустановленный заголовок Authorization. Это решение предоставляет гибкость в настройке и отправке API-запросов, а также поддерживает валидацию и обработку ошибок.

С помощью этого типа подключения вы можете связать платформу Altcraft с любым внешним сервисом, который требует пользовательских HTTP-заголовков для аутентификации или других целей, включая API-запросы с ключом в заголовке.



Подробнее о настройке данного типа подключения можно узнать здесь.

Улучшено

Улучшен блочный редактор для email-шаблонов:

- добавлена возможность вставлять ссылки и emoji в таблицы;
- обновлён UI сохранения и экспорта;
- исправлено проксирование пустых изображений;
- визуальные и UI улучшения некоторых элементов интерфейса.

Больше о блочном редакторе можно прочесть в этой статье.

Прочее

- Команда Altcraft продолжает работу над исправлением ошибок модуля Лояльности и улучшением его стабильности.
- Исправлена ошибка, при которой директория `data/segment` не очищалась автоматически.
- Пароли со спецсимволами к подключению PostgreSQL теперь корректно работают.
- Устранена ошибка, из-за которой невозможно было создать транзакцию с большим сроком сгорания баллов.
- Редактирование рассылок, сценариев и кампаний теперь невозможно, если им назначен обязательный атрибут без выбранного значения.
- Улучшена стабильность работы платформы при выполнении нескольких API-вызовов триггера.
- Исправлены проблемы с отображением аналитики и отчетов в платформе.
- Значения полей в шаблонах для `external object` теперь сохраняется корректно.
- Устранены некоторые ошибки, связанные с атрибутами и передачей их значений.
- Теперь кампания корректно деактивируется даже ессли момент деактивации совпадает с моментом формирования аудитории.

References

v2025.4.75.2774: PL-20989 PL-21186 PL-20844 PL-21107 PL-21228 PL-21163 PL-19018 PL-19865 PL-21208 PL-21194 PL-20755 PL-21189 PL-21051 PL-21145 PL-21154 PL-16968 PL-21074 PL-21130 PL-21111 PL-20834 PL-20870 PL-21106 PL-21093 PL-21048 PL-21100 PL-20807 PL-21102 PL-21099 PL-18812 PL-21064 PL-20938 PL-21000 PL-20071 PL-20932 PL-20066 PL-20973 PL-21015 PL-20972 PL-20824 PL-20828 PL-20774 PL-20858 PL-20809 PL-20703 PL-21205 PL-18625 PL-20952

v2025.4.74

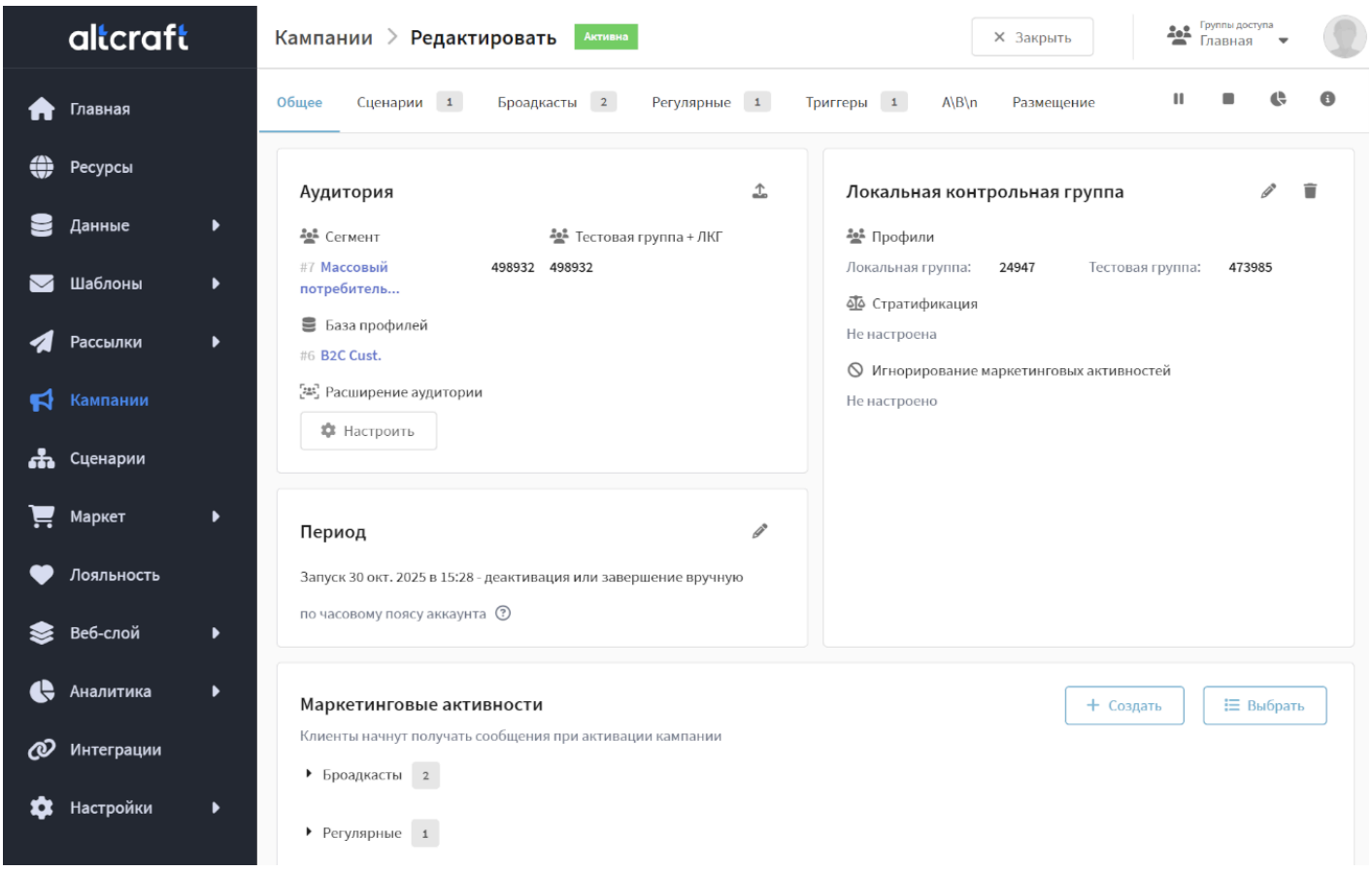
v2025.4.74.0 – Campaigns

> **CAUTION**
>
> Для корректной работы платформы требуется провести **миграцию**. Для этого:
>
> 1. Замените файлы платформы на файлы из новой сборки;
> 2. Запустите `./akd upgrade` и согласитесь с выполнением миграции;
> 3. Полностью перезапустите платформу командой `./akd restart`.

Кампании: комплексный маркетинговый инструмент

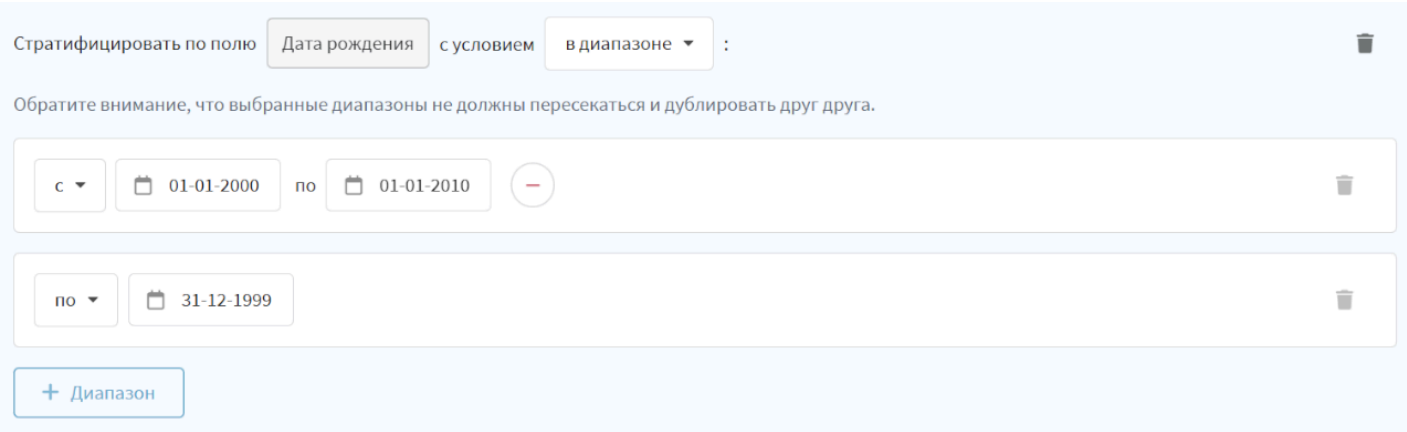
Раньше **“Кампаниями”** в платформе назывались все виды маркетинговых рассылок. В новой версии платформы они переименованы в “Рассылки”, а **“Кампании”** теперь представляют собой инструмент для комплексного управления маркетинговыми активностями в едином пространстве.

Внутри кампании можно настроить аудиторию, сегментировать базы данных, управлять профилями пользователей, контрольными группами и сценариями. Для кампаний можно как создавать новые, так и добавлять уже готовые маркетинговые активности: сценарии, броадкасты, регулярные, триггерные, мультивариантные и размещения.



Также для кампаний можно настроить **локальные контрольные группы**. Локальная контрольная группа позволяет выделить часть аудитории, которая не будет получать коммункаций в рамках кампании. В дальнейшем при сравнении основной и контрольной группы вы сможете замерить эффективность вашей

кампании. Для ЛКГ доступна опция стратификации — выделения групп (страт) на основе какого-либо признака. Стратификация позволяет создать контрольную группу, максимально приближенную к вашей аудитории.



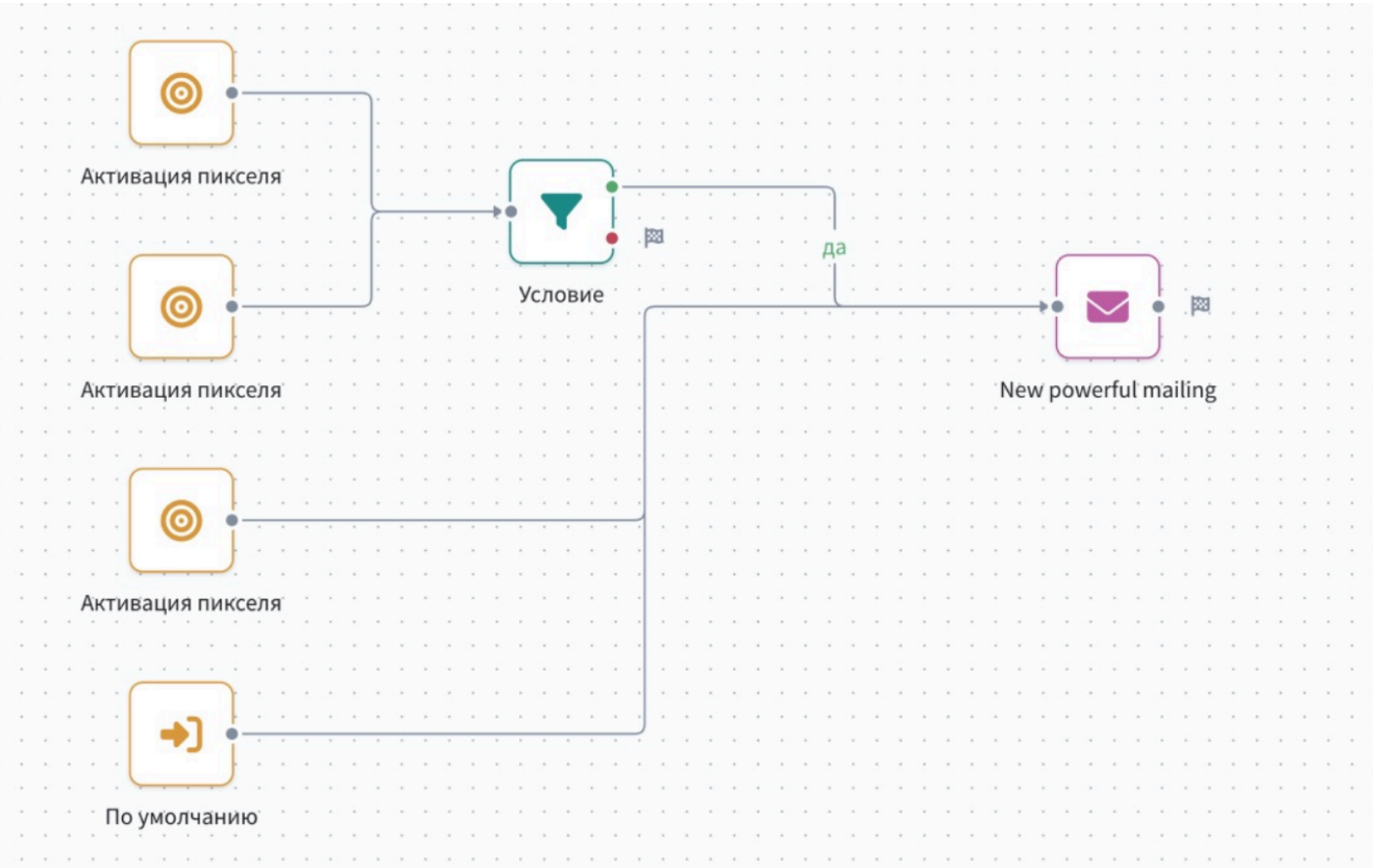
Аналитика по всем активностям внутри кампании также доступна в рамках единого отчета с множеством показателей:

Новые кампании позволяют консолидированно управлять разными активностями, отслеживать эффективность коммуникаций, а также проводить маркетинговые эксперименты.

Улучшено

Комбинированный захват профилей в сценарии

Раньше платформа позволяла добавить по одному входному триггеру каждого типа внутри одного сценария. Теперь же вы можете добавить сразу несколько триггеров одинакового типа, например, активация разных пикселей или обновление полей профиля:



Обратите внимание, максимальное количество стартовых узлов ограничено 10. Изменить это можно в парамтре конфигурации `WORKFLOW_MAX_START_NODES_COUNT`.

Прочее

- Улучшены некоторые UI/UX-компоненты платформы.
- Исправлены ошибки в работе некоторых API маркета.
- Улучшена стабильность процессов `proctask` и `procpixel`.
- Исправлены некоторые ошибки кампаний с пользовательскими каналами.
- Доставка сообщения в интеграции с SMS-провайдером Intellin теперь корректно регистрируется платформой.
- Устранены ошибки, при которых сегменты могли не расчитываться или расчитывались чрезвычайно долго.
- Цели со SMID из пользовательского канала снова регистрируются корректно.
- Теперь при запуске платформы с флагом `--config` пользовательский путь к файлу конфигурации передается всем сервисам платформы, а не только основному процессу.

References

v2025.4.74.0: PL-16253 PL-20320 PL-20303 PL-20304 PL-19139 PL-20225 PL-20226 PL-20194 PL-19904 PL-20183 PL-19144 PL-20157 PL-20114 PL-20092 PL-20082 PL-19212 PL-19729 PL-19606 PL-19775 PL-19709 PL-19870 PL-19734 PL-19919 PL-19981 PL-19850 PL-19855 PL-19818 PL-19671 PL-19661 PL-19633 PL-19328 PL-18624 PL-19600 PL-19567 PL-1785 PL-19053 PL-17465 PL-19441 PL-17851 PL-19266 PL-19257 PL-19254 PL-19227 PL-18933 PL-19082 PL-1875 PL-17774 PL-18922 PL-8756 PL-18527 PL-18857 PL-18572 PL-17696 PL-18570 PL-18788 PL-17775 PL-18232 PL-17740 PL-17008 PL-19207

v2025.3.73

CAUTION

Начиная с версии **v2025.3.74**, будут внесены следующие изменения в работу API:

- Устаревшее API версии 1.0 будет отключено. Рекомендуется заранее перейти на актуальную версию API.
- API-токен больше нельзя будет передавать в параметрах GET или в теле JSON. Вместо этого используйте заголовок `Authorization: Bearer <token>`.
- Параметр `format` больше не будет поддерживаться. Указывайте нужный формат через заголовок `Accept` (например, `Accept: application/json`).

Рекомендуем заранее внести все необходимые изменения, чтобы избежать перебоев в работе API.

Кроме того, **запуск кампаний без ресурса станет невозможен**. Пожалуйста, проверьте, что у всех кампаний есть привязанный ресурс.

v2025.3.73.1 – Attributes, GCG stratification, Scenario improvements

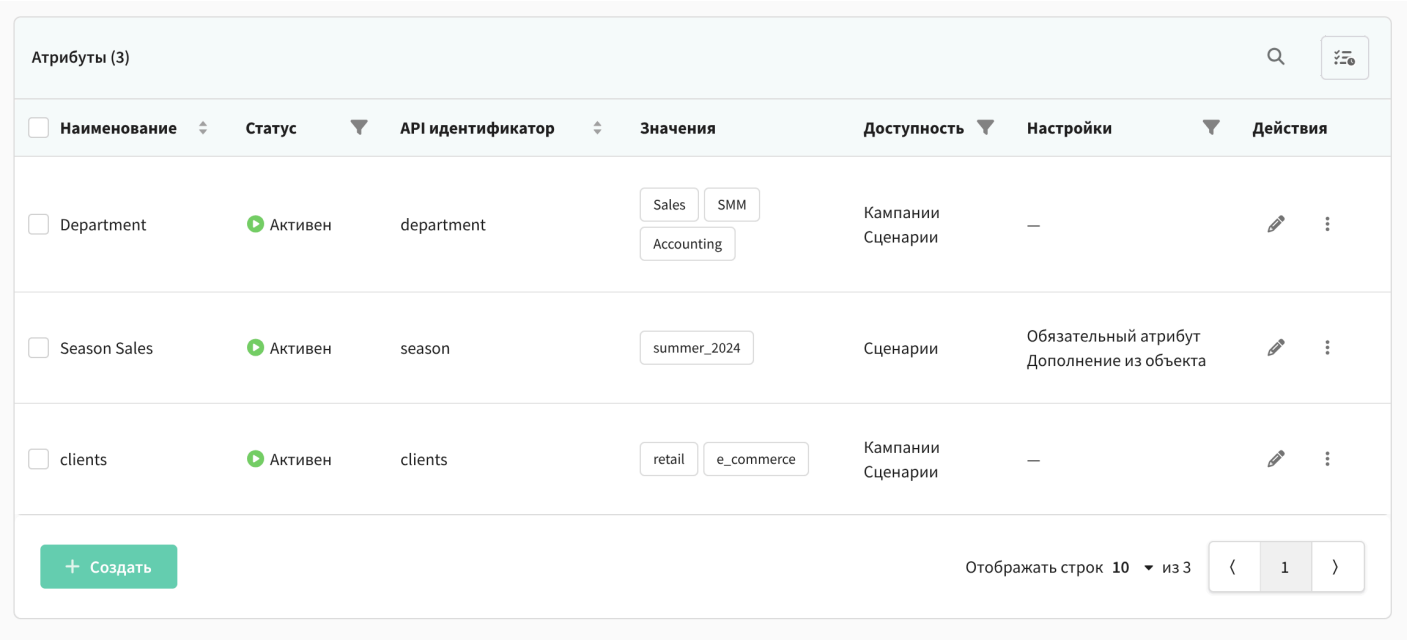
Добавлено

Атрибуты

Добавлены **атрибуты** для сущностей платформы. Атрибуты — специальные пользовательские поля, добавляемые к сущностям платформы. Они представляют собой определённый **параметр** с набором возможных **значений**.

С помощью атрибутов можно:

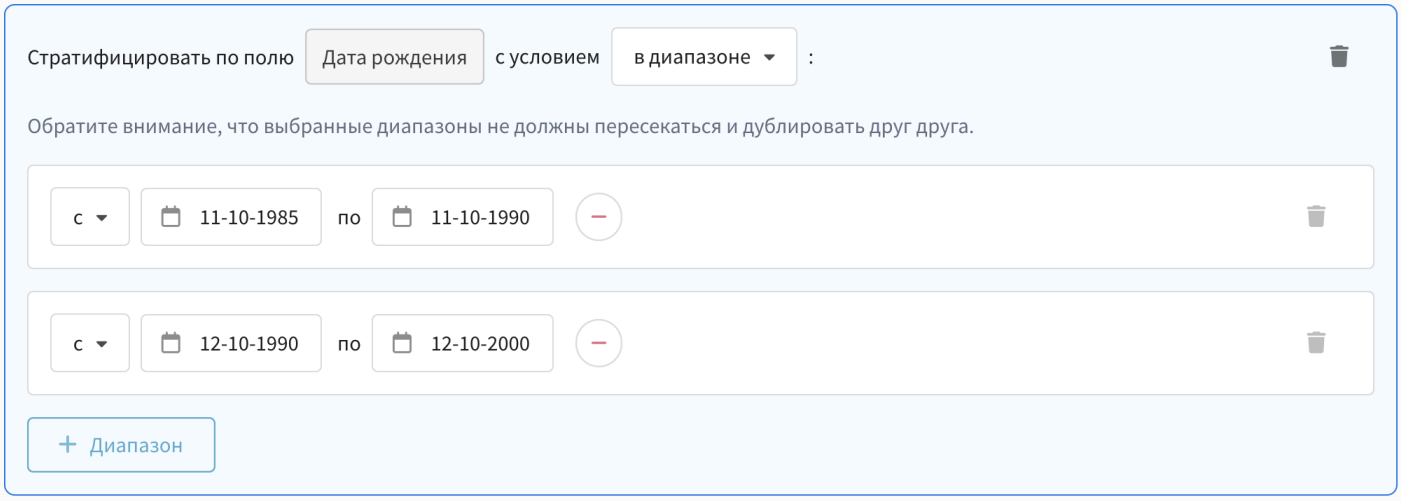
- искать и организовывать объекты;
- выбирать кампании и сценарии для сегментации;
- подставлять значения в контент и API-запросы.



Подробнее об атрибутах можно узнать в [этом разделе].

Стратификация в ГКГ

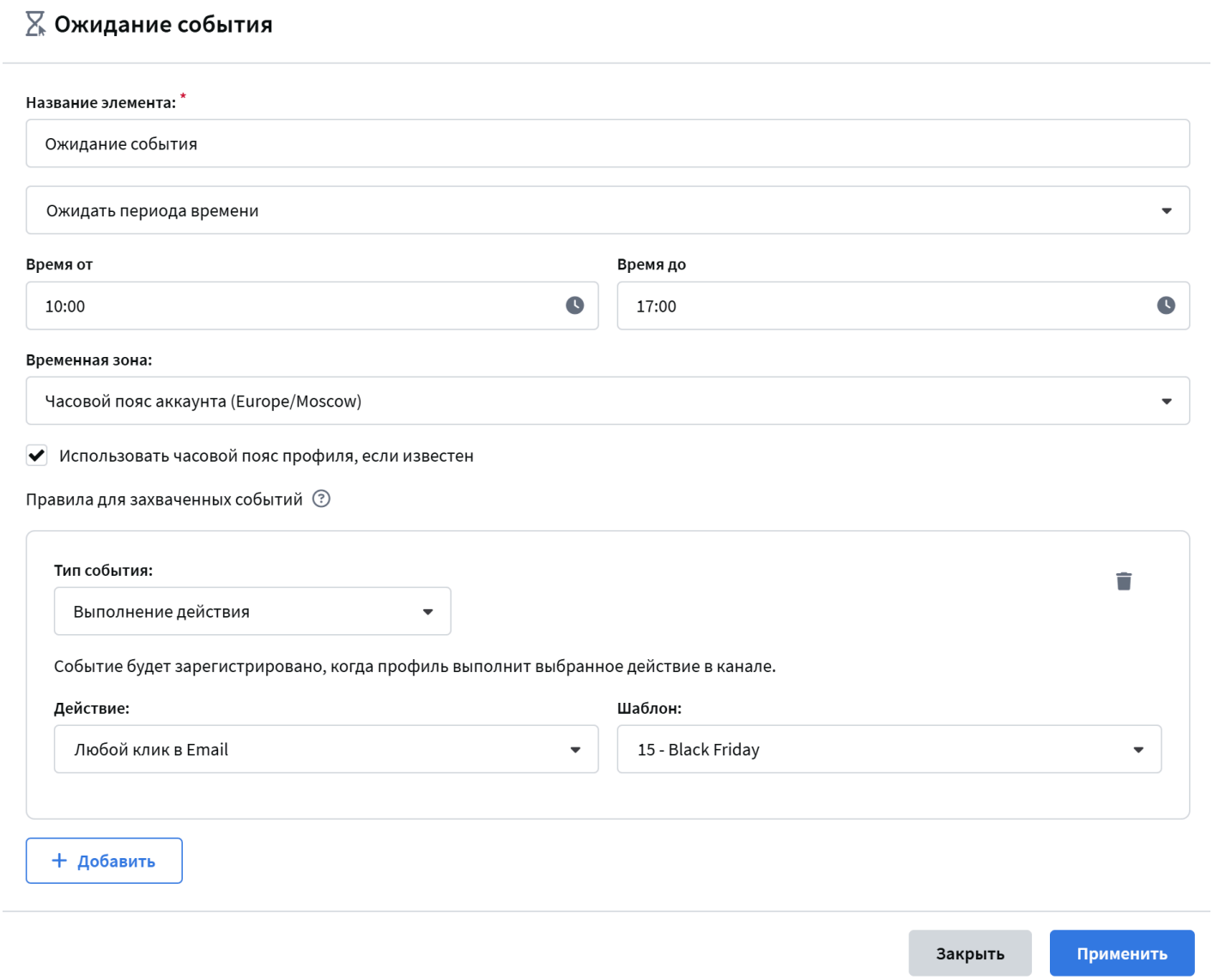
Теперь для глобальной контрольной группы можно настроить **стратификацию**. Стратификация — это метод случайного отбора профилей с предварительным разделением аудитории на подгруппы (страты) по заданным характеристикам. Аудитория разбивается на группы так, чтобы каждая из них отражала состав всей совокупности.



Из каждой группы случайным образом выбирается часть профилей, пропорциональная размеру этой группы. Это помогает сформировать контрольную группу с таким же распределением характеристик, как у тестовой, что делает результаты кампании более точными. Настройка стратификации описана в [соответствующей статье] документации.

Новый элемент сценария “Ожидание события”

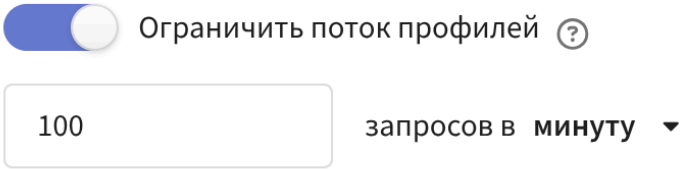
Добавлен новый элемент сценария — оператор **“Ожидание события”**. Он останавливает движение профиля до тех пор, пока им не будет выполнено определенное действие. Как только любое из указанных событий зарегистрируется платформой, профиль перейдет на следующий элемент по соответствующей ветке. Если же событие не наступает в течение указанного времени, то профиль идет по другой ветке.



С помощью элемента можно настроить реактивационные цепочки или, например, [сценарий брошенной корзины]. Прочесть больше об элементе можно [здесь].

Ограничитель профилей в элементе “Вызов API”

Элемент сценария “Вызов API” теперь может **ограничивать количество профилей**, проходящих за определенный период времени через этот узел. Это необходимо, чтобы не перегружать внешние сервисы, к которым мы обращаемся из сценария. Механизм работы ограничителя основан на алгоритме **“текущего ведра”**. Подробнее — в соответствующем разделе статьи [“Элементы сценариев”].



Улучшено

- Оптимизирована работа процессов `proctask` и `procintegras`.
- Обновлен CSS-инлайнер в шаблонах сообщений и кампаниях.

Прочее

- Устранены различные UI/UX-ошибки в платформе.
- Счетчики повторных попыток отправки в сендерах теперь корректно отображают очередь.
- Исправлены ошибки, возникавшие при попытке создать и сохранить мультивариантный тест.
- Отображение и фильтрация сендеров в Панели администратора теперь работает корректно.
- Исправлена ошибка, не позволявшая искать в списке профили, находившиеся в определенном элементе сценария. Также теперь JSON-контент, привязанный к этому профилю в элементе, открывается без ошибок.
- SQL-экспорт в сегменты снова работает корректно.
- Устранены ошибки, связанные с генерацией и отображением превью шаблонов.
- Добавлена возможность сохранить коннектор к внешней БД с SSL-подключением без указания сертификатов.
- Аналитика по целям с косыми чертами в названии снова отображается корректно.

References

v2025.3.73.1: PL-18644 PL-18219 PL-18646 PL-18645 PL-18715 PL-18218 PL-18220 PL-18217 PL-18452 PL-18224 PL-18228 PL-18344 PL-17534 PL-18216 PL-18802 PL-18149 PL-18309 PL-18133 PL-18184 PL-18082 PL-17559 PL-18772 PL-16592 PL-18913 PL-18978 PL-18338 PL-18867 PL-18899 PL-18885 PL-18537 PL-18072 PL-18864 PL-18861 PL-18875 PL-18825 PL-12690 PL-18240 PL-16060 PL-18944 PL-18878 PL-18873 PL-18722 PL-18832 PL-18881 PL-15383 PL-18351

---

v2025.3.73.1140 – Altcraft Mobile SDK v2

Добавлено

Altcraft mobile SDK v2

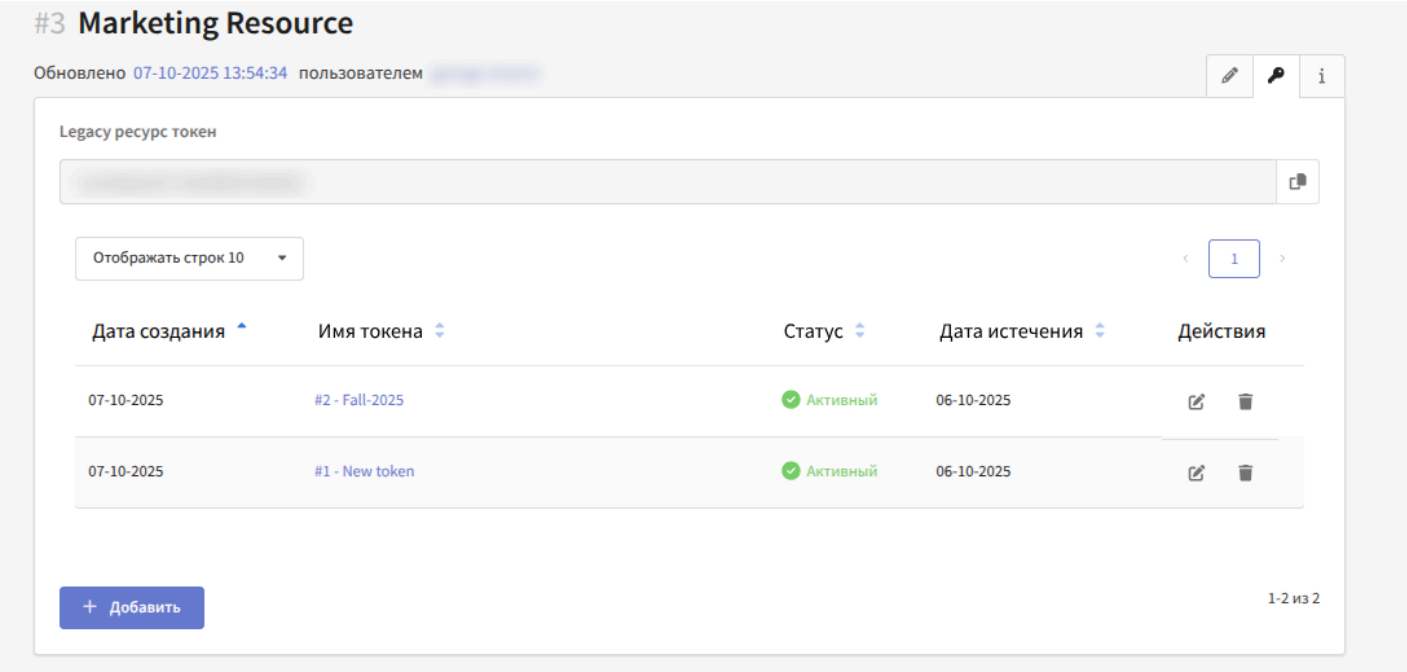
Мы улучшили наш мобильный SDK для работы push-уведомлений в приложениях. Основные улучшения:

- Поддержка нескольких видов авторизации: JWT-авторизация и авторизация с использованием ролевого токена Altcraft;
- Возможность показа push-уведомлений с контекстом, настраиваемым в платформе;
- Автоматическое обновление токена в случае его изменения;
- Автоматическая передача событий доставки и открытия;
- Новые функции, позволяющие работать с анонимными и зарегистрированными пользователями, а также с несколькими пользователями на одном устройстве;

Ознакомиться подробнее можно в разделе документации для разработчиков. Altcraft mSDK доступен на GitHub.

Остальное

- Добавлены переменные для использования данных пиксельных событий в сценариях и различном контенте. Подробнее об этих переменных читайте здесь.
- Добавлена вкладка с настройкой токенов для ресурса. На этой вкладке можно создать токен для ресурса, указать его срок действия и роли для него. Перейти на вкладку можно после создания и сохранения ресурса (вкладка расположена в правой верхней части экрана):



- Добавлены API-хендлеры для удаление продуктов и SKU из Маркета.
- Добавлена возможность создать в платформе интеграцию с Yandex.Appmetrica без создания пикселя. Такая опция может быть полезна, если нужен импорт профилей без использования целей.
- Добавлена проверка динамического JSON, добавляемого пользователем, на наличие объекта (массив не принимается).
- Добавлена возможность указать текущий день в условиях отбора и в сегментации.
- Добавлена проверка подключения к Redis-хранилищу при запуске платформы.

Улучшено

- Теперь история сообщений в Jivo-чате для клиентов привязана к логину пользователя — это поможет сотрудникам поддержки быстрее решать проблемы клиентов с платформой.
- Улучшена стабильность процессов платформы.
- Некоторые улучшения и исправления компонентов UI/UX.
- Теперь генератор коротких ссылок не создает ссылки с точкой на конце.
- В логах процесса `procintegras` теперь указывается информация, связанная с профилем и отправкой push-сообщения в Firebase.

Прочее

- Параметр `FIREBASE_PUSH_BATCH_WORKERS` теперь по умолчанию равен 200.
- Устранена ошибка, при которой в пользовательском целочисленном поле кампании можно было записать точку или запятую.
- Обработка статуса запроса при отправке SMS в интеграции с SMS Traffic снова работает корректно.
- Устранена ошибка, из-за которой сегменты с условием “Участие в сегменте” рассчитывались дольше обычного.
- Зарезервированные задачи снова переводятся на статус `registered`.
- Устранена ошибка, при которой логическое поле профиля не меняло свое значение после заполнения формы.
- Атрибуты теперь корректно подставляют значения при использовании в пользовательских каналах.
- Исправлены ошибки в работе аналитики по NPS-формам.
- Интеграция с EDNA снова работает корректно.
- Исправлены ошибки, связанные с миграцией данных при обновлении версии платформы.

- Регистрация push-событий Rustore снова происходит корректно.
- Функция `formatdate` в элементе сценария “Вызов API” теперь корректно работает с полями профиля.
- Устранена ошибка, из-за которой коллекции веб-версий удалялись слишком рано.
- Менеджер подписок снова корректно отображается у получателей сообщений.
- При обновлении шаблона через API дата и время изменения снова меняется в соответствии с действительным временем обновления.
- Теперь при создании сендера не отображаются уже занятые IP-адреса.
- Устранена ошибка, при которой нельзя было отправить профиль в сценарий с атрибутом, который игнорируется в ГКГ.
- Исправлены часто возникающие проблемы при работе в визуальном email-редакторе:
    - исправлено отображение нескольких шрифтов;
    - функции Transition, Perspective, Transform снова работают корректно;
    - устранены ошибки при установке и сохранении фона письма;
    - изображения в блоке со стилями загружаются корректно.
- Переход по ссылкам в письме снова работает корректно.
- Значения по умолчанию для параметров `SEGSTATIC_PROFILE_CHANGES_BATCH_SIZE` и `API_GET_PROFILES_BATCH_SIZE` изменены на 20000 и 10000 соответственно.
- Устранена ошибка, из-за которой API-запрос `templates/list` не возвращал поле `next_from_id`.
- Смена пароля при помощи утилиты `aktool` снова работает корректно.
- Исправлена ошибка, при которой ограничение количества стратификаций КГ из параметра конфигурации не отображалось в интерфейсе платформы.

References

v2025.3.73.1140: PL-19333 PL-19309 PL-19326 PL-18399 PL-19176 PL-19273 PL-18437 PL-17717 PL-18369 PL-18236 PL-17276 PL-19221 PL-19137 PL-18870 PL-18665 PL-17683 PL-17885 PL-17853 PL-17154 PL-17100 PL-19285 PL-18320 PL-19060 PL-18770 PL-19053 PL-19120 PL-19254 PL-19229 PL-18327 PL-19218 PL-18836 PL-15418 PL-19233 PL-19204 PL-17933 PL-19214 PL-19223 PL-18414 PL-19118 PL-19210 PL-18328 PL-19199 PL-18432 PL-18891 PL-19116 PL-19176 PL-19068 PL-18334 PL-10830 PL-19131 PL-19089 PL-19122 PL-19125 PL-18694 PL-19117 PL-17977 PL-18933 PL-17274 PL-18912 PL-18611 PL-17809 PL-17768 PL-18308 PL-13780 PL-18366 PL-19119 PL-18333 PL-19114 PL-19115 PL-18155 PL-18471 PL-18594 PL-18547 PL-16869 PL-19079 PL-19086 PL-19075 PL-19083 PL-19028 PL-19050 PL-17758 PL-19002 PL-19010 PL-18607 PL-18223 PL-18620 PL-16448 PL-17580 PL-19026 PL-19267 PL-19314 PL-19262 PL-16464 PL-18837 PL-19296 PL-19228 PL-19277 PL-18994 PL-19238 PL-18363 PL-19105 PL-18343 PL-19143 PL-18643 PL-18538

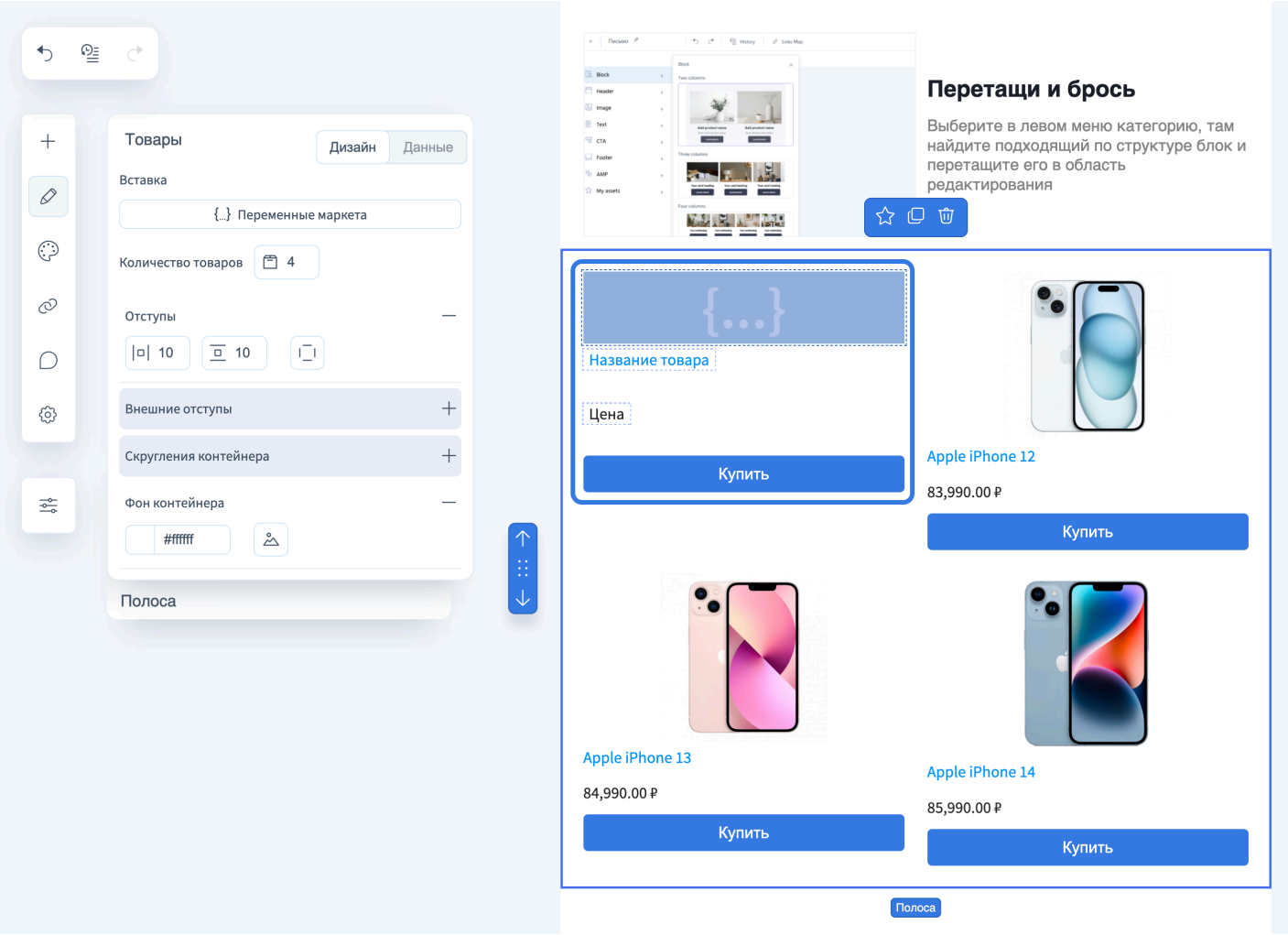
---

v2025.3.73.1942 – Updated email block editor, Faster Imports and Campaigns

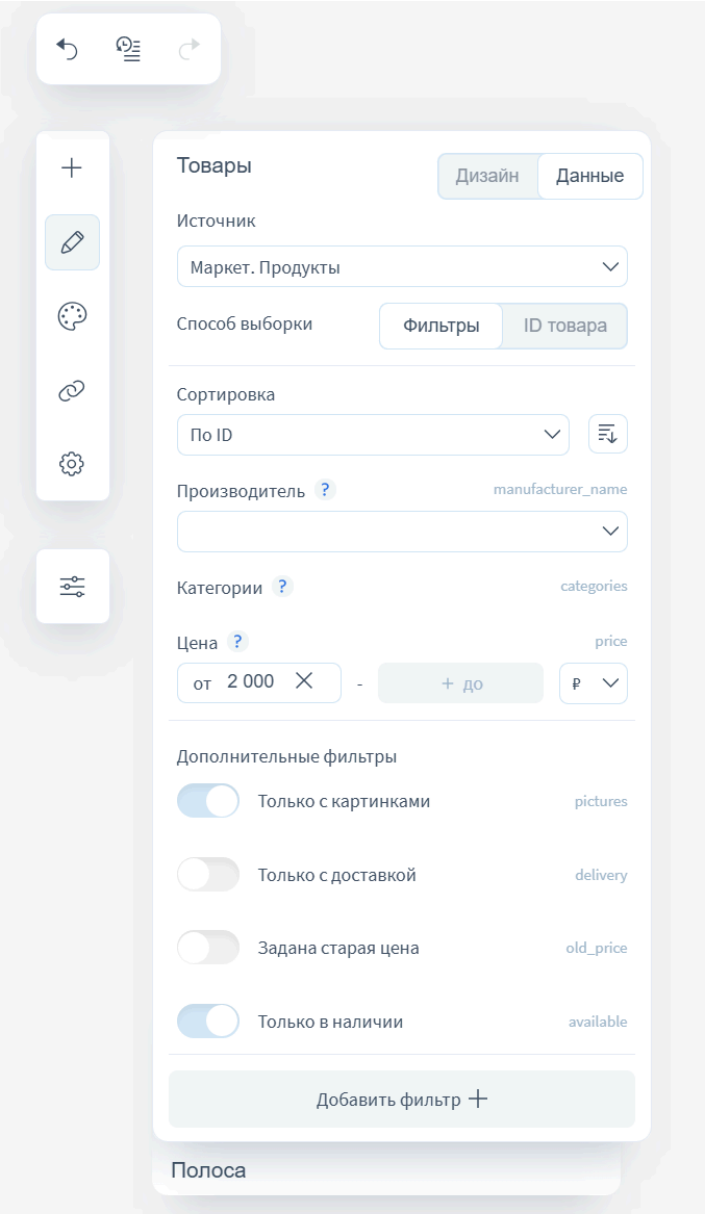
Добавлено

Обновленный блочный email-редактор

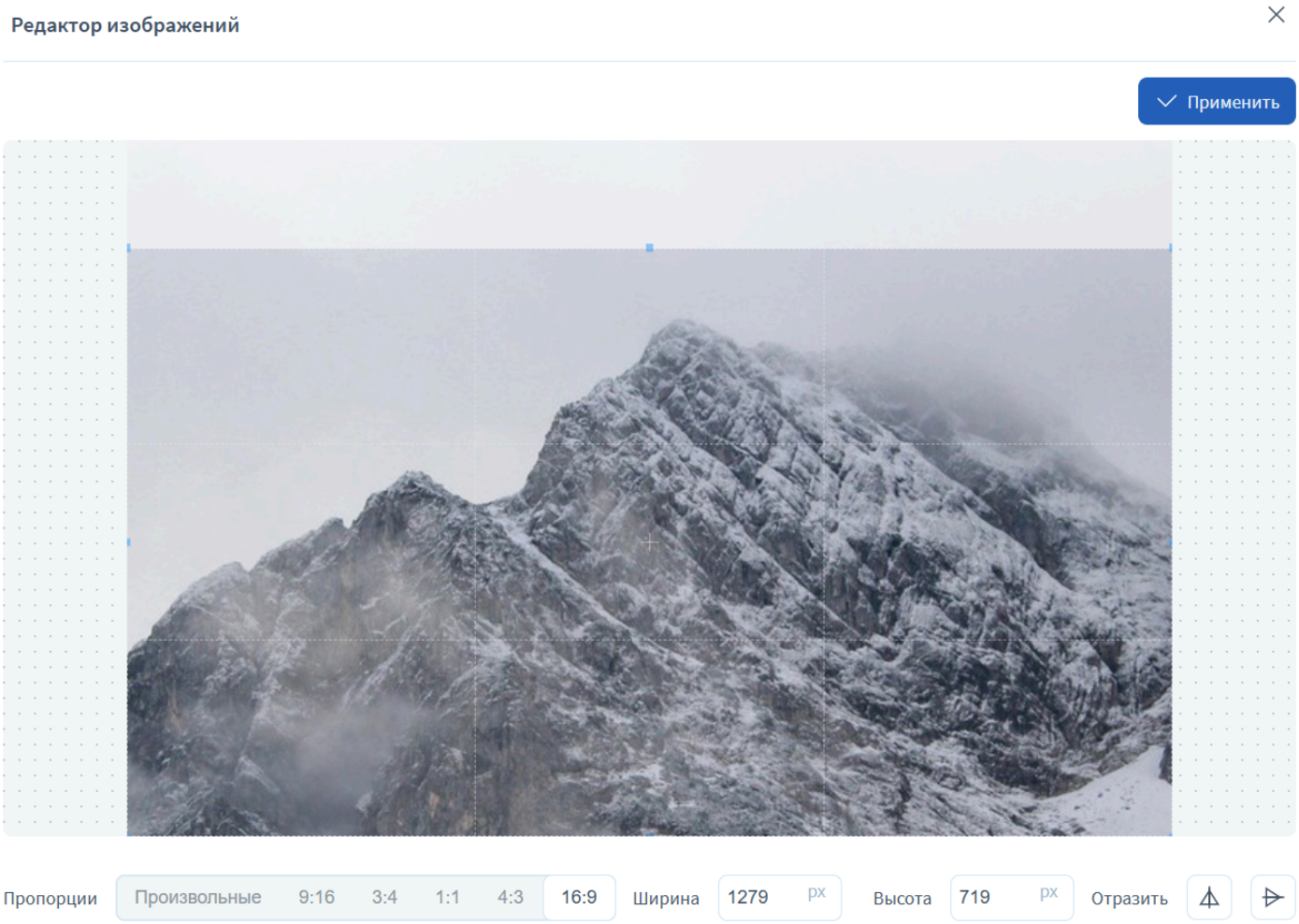
Блочный редактор писем стал более функциональным и удобным. Добавлен новый элемент “Товары”, который автоматически подтягивает продукты и SKU из Маркета Altcraft.



Теперь вы можете настраивать дизайн карточек товаров и фильтровать их прямо в интерфейсе редактора. Данные будут динамически обновляться при отправке письма. Если же нужна однократная рассылка, можно зафиксировать товары в карточке и превратить их в статичный контент.



Кроме того, в галерее блочного редактора появилась возможность редактировать изображения: обрезать, менять пропорции, поворачивать и отражать.



Остальное

- В SDK появилась возможность регистрации произвольных пользовательских событий.
- Расширен метод /api/v1.1/templates/get. Добавлены параметры для фильтрации шаблонов по каналам и для получения списка переменных, используемых в шаблоне.
- Если кампания запускается в рамках сценария, то методы /api/v1.1/campaigns/triggers/list и /api/v1.1/campaigns/triggers/get теперь возвращают id этого сценария (параметр `workflow_id`). В методе /api/v1.1/campaigns/triggers/list добавлена возможность фильтрации кампаний по `workflow_id`.
- Добавлена поддержка вывода кастомных полей подписки в шаблонах через функцию `subscription_field`.
- Добавлена функция `len` для сравнения длины массива с переданным числом. Например, чтобы подставить в шаблон только продукты, у которых есть изображения, вы можете использовать выражение: `products("!len(product.pictures, 0)" 10 "manufacturer_name" "desc")`.

Улучшено

- При экспорте собранных данных формы теперь в отчёт записывается ID профиля, что позволяет определить, какой профиль заполнил форму.
- Улучшен процесс записи событий в журнал аудита: теперь события обрабатываются через отдельную очередь `ak_audit_events` и сохраняются батчами.
- Значительно оптимизирован процесс проверки и обновления политик отправки. Ранее во время выполнения кампании политика каждого контакта (подписки) проверялась и обновлялась отдельно; в новой версии политики обрабатываются батчами, что в некоторых случаях повышает скорость выполнения кампании до 10 раз.

- Оптимизирован процесс импорта профилей за счёт пакетной обработки данных. В некоторых случаях это увеличивает скорость импорта через API, файлы и SQL-таблицы до двух раз. Для активации нового режима необходимо установить параметр `IMPORT_BATCH_OPTIMIZER_ENABLE` в значение `true`.

Прочее

- В разделе “Зависимые объекты” сегментов теперь корректно отображаются все связанные объекты, включая сценарии и триггерные кампании.
- API-контент корректно отображается при просмотре профиля через сценарий в элементе ожидания.
- Расчёт ГКГ теперь корректно выполняется для больших баз профилей.
- Исправлены некоторые уязвимости в системе безопасности платформы.
- Исправлена ошибка, из-за которой после объединения профилей цель сохранялась в истории, но такой профиль не отображался в результатах сегментации.
- Политика отправки теперь корректно работает, и профили, не подходящие под условия кампании, больше не включаются в отправку при повторных запусках.
- При использовании SQL-запросов шаблон теперь корректно рендерится даже при отсутствии у профиля используемых полей — вместо них подставляются значения по умолчанию.
- Исправлена регистрация целей через API для пользовательских каналов с использованием `send_message_id`.
- Пробелы в начале и конце названия цели теперь автоматически обрезаются.

References

v2025.3.73.1942: PL-19442, PL-19540, PL-19064, PL-17853, PL-19472, PL-19467, PL-19453, PL-18128, PL-19302, PL-19258, PL-19328, PL-16031, PL-19142, PL-19030, PL-17284, PL-17393, PL-19439, PL-18582, PL-18428, PL-17607, PL-19434, PL-19324, PL-18759, PL-17456, PL-17769, PL-19347, PL-19255, PL-17817, PL-18222, PL-19195, PL-18533, PL-19098, PL-18954, PL-19055, PL-18796, PL-18542, PL-18529, PL-18221, PL-17392, PL-18109, PL-19360, PL-19219, PL-19345, PL-19259, PL-19222, PL-19237, PL-17829, PL-18405, PL-18511, PL-19385, PL-18842

---

v2025.3.73.2276 – Platform Improvements, Fixed Event Context

В этом обновлении исправлены часто возникающие ошибки, улучшен функционал платформы и её стабильность.

Улучшено

Контекст событий

Исправлены ошибки в работе переменных, связанных с контекстом событий:

- Теперь объект `event` содержит количество событий, указанное в параметре `WORKFLOW_EVENT_CONTEXT_LENGTH`;
- Итератор по событиям внутри шаблона (вида `{for $index $event = event.all}`) снова работает корректно.

Подробнее об этих переменных и о том, как ими пользоваться, можно прочитать в этой статье.

Блокировка повторного срабатывания триггеров

Теперь опция пикселя “Блокировать повторное срабатывание триггеров” ограничивает конкретные **цели** пикселя, а не весь пиксель целиком. Например, если у вас настроены сценарии с триггером по одному и тому же пикселю, но по разным целям, все они будут работать корректно. Это позволяет использовать один пиксель для разных маркетинговых задач.

Прочее

- Устранена ошибка логирования, связанная с неправильной работой параметра `LOG_ROTATE_MAX_FILES`. Теперь параметр означает общее количество файлов ротации, кроме текущего лога, а параметр `LOG_ROTATE_ZIP_FILES` означает то, сколько из них надо заархивировать.
- Добавлен параметр конфигурации `PROC_TASK_RESERVED_TASK_IDLE_TIME_LIMIT_MIN`. Он отвечает за максимальное время простоя задачи в статусе `reserved`. Если задача находится в статусе `reserved` дольше этого времени, она вернется в статус `registered`. Это предотвращает ошибки в задачах при разрывах соединения.
- Статический сегмент больше не пересчитывается автоматически при его отправке в сценарий — в сценарий попадут только те профили, которые уже входят в него на момент отправки.
- Сценарии с кампаниями, использующими фильтр по дополнительным полям SKU, снова работают корректно.
- Спецсимволы теперь корректно отображаются в push-сообщениях, отправляемых через AppMetrica.
- Устранена ошибка, при которой в веб-версиях писем обрезались адреса ссылок.
- Экспорт писем, созданных в Визуальном редакторе, снова работает корректно.
- Устранена ошибка, при которой в некоторые push-шаблоны не вставлялись переменные Промокодов.
- URL-декодер в Панели администратора теперь может обрабатывать трекинговые ссылки из сообщений пользовательских каналов.
- Исправлена ошибка отправки сообщений, возникающая после изменения настроек пользовательского канала. Теперь шаблоны в таких каналах работают корректно и без пересохранения.
- Исправлена ошибка авторизации, возникавшая при использовании параметра `PASSWORDS_MAX_AGE`.
- Шаблоны сообщений с переменной `{resource_token}` теперь рендерятся корректно.
- Оптимизирована запись статистики сценариев, благодаря чему снижена нагрузка на RabbitMQ.
- Улучшена стабильность работы процессов.
- Исправлены некоторые UI/UX-ошибки.
- Добавлена проверка подключения к БД Redis при запуске платформы.
- Подключение к платформе по LDAP (как с SSL-, так и с TLS-соединением) снова работает корректно.

References

v2025.3.73.2276: PL-19691 PL-18213 PL-19277 PL-11085 PL-19553 PL-19022 PL-19574 PL-16848 PL-19242 PL-19462 PL-19332 PL-19630 PL-19596 PL-19432 PL-19318 PL-19618 PL-19608 PL-17716 PL-19473 PL-19554 PL-19247 PL-19270 PL-19216 PL-19101 PL-19401 PL-19099 PL-19600 PL-19564 PL-19555 PL-18362 PL-18568 PL-19155 PL-11262

v2025.3.73.2565 – Action Hook for past events

Добавлено

Захват исторических событий

Теперь при помощи захвата событий можно передать случившиеся события из прошлого. Для этого воспользуйтесь соответсвующей функцией при настройке захвата и укажите даты выгрузки. Набор

передаваемых полей ограничен, но все ключевые данные события будут переданы.

Подробнее узнать о захвате исторических событий можно здесь.

Улучшено

- Теперь при отправке сообщения в Telegram по несуществующему ID платформа будет регистрировать событие hard bounce и менять статус этой подписки.
- Обновление блочного редактора: улучшена стабильность рендеринга шаблонов, обновлен стандартный шаблон письма, а также устранена ошибка, при которой пользователям отправлялась невалидная версия AMP-письма.
- Для настройки доступна новая версия интеграции с провайдером 360Dialog для отправки сообщений в WhatsApp.
- Внесены некоторые улучшения интерфейса, связанные с таблицами и счетчиками. Также исправлены ошибки в
- В захват событий добавлены события, связанные с глобальными контрольными группам.

Прочее

- Добавлена валидация значений для времени ожидания и значения пикселя на странице кода пикселя.
- Улучшена стабильность процессов `procsmsev` и `proctask`.
- Устранена ошибка, возникающая при попытке подключить к платформе БД без таблиц.
- Добавлена поддержка Markdown для превью пользовательских каналов.
- Исправлена ошибка, из-за которой при активации сценария появлялось модальное окно с предупреждением, что в сценарии обрабатываются профили.
- Запрос к хендлеру `/cookiesaver/readiness` снова работает корректно.
- Устранены ошибки в работе интеграции с сендером SMS Traffic.
- Теперь пользователь с истекшим сроком действия пароль обязательно должен его поменять при следующем входе.
- При редактировании push-шаблона снова можно вставить и сохранить параметр в поле “Ссылка для перехода по клику (Только SSL)”.
- Улучшено логирование web push-отправок в браузеры Safari.
- Функция `format` больше не обрезает знак отрицания для отрицательных начинающихся с нуля дробных чисел.

References

v2025.3.73.2460: PL-19898 PL-19216 PL-19876 PL-19680 PL-18358 PL-19638 PL-19681 PL-19862 PL-19573 PL-19745 PL-19771 PL-19212 PL-19013 PL-19547 PL-19141 PL-13097 PL-19572 PL-19497 PL-19501 PL-19746 PL-17495 PL-17658 PL-19460 PL-19753 PL-19721 PL-19733 PL-19670 PL-19588 PL-17528 PL-19592 PL-19594 PL-19394 PL-19469 PL-19334 PL-18736 PL-17537 PL-16721

v2025.3.73.2888 – Tag Manager variables in HTML tags

Добавлено

Использование переменных Менеджера тегов в HTML-теге

Ранее переменные можно было использовать только в триггерах Менеджера тегов. Теперь их можно вставлять в HTML-тег контейнера в формате: `{{varName}}`. Это упрощает многократное использование тегов

в различных контекстах. Подробнее.

Прочее

- Триггер активации пикселя теперь корректно проверяет диапазон дат.
- Статистика NPS-опроса больше не сбрасывается при обновлении формы.
- Сценарии с запросами к внешним таблицам работают корректно.
- Многостраничные формы с подстановкой полей профиля снова успешно отправляются.
- Устранена ошибка, при которой apicontent профилей при массовом импорте в сценарий работал некорректно.
- Числовые значения из apicontent корректно подставляются в URL запроса в API-узле.
- При экспорте истории профиля данные цели, в том числе pixel_data, успешно выгружаются.
- Исправили проблему подключения к базе данных в Яндекс Облаке с использованием CA-сертификата.
- Оптимизирована работа процессов proctask и procintegras.
- Исправлены некоторые UI/UX-ошибки в интерфейсе.

References

v2025.3.73.2888: PL-16683, PL-17392, PL-18422, PL-18751, PL-18847, PL-18862, PL-18876, PL-19523, PL-19694, PL-19755, PL-19842, PL-19881, PL-19884, PL-19899, PL-19916, PL-19924, PL-19926, PL-19983, PL-20002, PL-20003, PL-20039, PL-20054, PL-20090, PL-20101, PL-20196, PL-20200, PL-20304, PL-20320

v2025.2.72

v2025.2.72.0 – Redis storage, processes optimization

Добавлено

Redis-хранилище

Теперь для корректного функционирования платформы требуется использование **Redis-совместимого хранилища**. Такое хранилище:

- позволяет постепенно отказаться от файловых персистентных хранилищ;
- повышает отказоустойчивость и управляемость платформы;
- обеспечивает более надежное и быстрое взаимодействие между сервисами.

Мы рекомендуем использовать **Kvrocks**. В документации по установке платформы приведено руководство по его подключению.

Поддержка кворумных очередей RabbitMQ и единый виртуальный хост

CAUTION

Перед установкой обновления необходимо создать новый виртуальный хост (по умолчанию `altcraft`) и убедиться, что текущий пользователь имеет на него полные права.

INFO

Если в старых виртуальных хостах остались сообщения, для их переноса можно использовать плагин Shovel. Ниже приведен пример переноса сообщений из очереди `trk_read` в виртуальном хосте `/` в виртуальный хост `altcraft`:

```
rabbitmqctl set_parameter shovel trk_read '{
  "src-protocol": "amqp091",
  "src-uri": "amqp://altcraft:abcdefghijklmnopqrstuvwxyzABCDEF@localhost:5672",
  "src-queue": "trk_read",
  "dest-protocol": "amqp091",
  "dest-uri":
"amqp://altcraft:abcdefghijklmnopqrstuvwxyzABCDEF@localhost:5672/altcraft",
  "dest-queue": "trk_read"
}'
```

Другие изменения

- Теперь пользователь может создать новый отчет на основе существующего.
- Добавлена возможность настроить пул обработчиков для RMQ Event Sender.

- Теперь время пользовательской сессии настраивается через параметр `PROCAUTH_SESSION_TIME_SEC`, а возможность продолжать использовать сессию при смене IP-адреса — через `PROCAUTH_SESSION_ALLOW_CHNET`.

Улучшено

- Улучшена производительность платформы.
- Улучшена стабильность процесса `procintegras`.
- Оптимизирован механизм обработки задач процессом `proctask`.

Прочее

- Исправлены некоторые UI/UX-ошибки в интерфейсе.
- Устранены ошибки, связанные с некорректным отображением показателей аналитики.
- Проверка наличия ссылки подтверждения в шаблонах снова работает корректно.
- Устранены ошибки, связанные с авторизацией в платформе.
- Элементы сценариев автоматизации стали работать стабильнее — устранены ошибки, возникавшие в их работе.

References

v2025.2.72.0: PL-18318 PL-18343 PL-18519 PL-18367 PL-18349 PL-18323 PL-18192 PL-18321 PL-18322 PL-18262 PL-18316 PL-18483 PL-18445 PL-18482 PL-18276 PL-18458 PL-18427 PL-18389 PL-18372 PL-18347 PL-18319 PL-18332 PL-18403 PL-18396 PL-18359 PL-18375 PL-18379 PL-18302 PL-18307 PL-18295 PL-18255 PL-17490 PL-17886 PL-18134 PL-17904 PL-16949 PL-16946 PL-17729 PL-17590 PL-16965 PL-17223 PL-15798 PL-17545 PL-17068 PL-17474 PL-17453 PL-17131 PL-17451 PL-17442 PL-17399 PL-16918 PL-17328 PL-17302 PL-16782 PL-17192 PL-17243 PL-17023 PL-16948 PL-17101 PL-17087 PL-17091 PL-11284 PL-16885 PL-17030 PL-17029 PL-16969 PL-17013 PL-6658 PL-16541 PL-16356 PL-16702 PL-3437 PL-16530

---

v2025.2.72.707 – Custom channel improvements, link shorter optimization

DANGER

Обратите внимание на важные изменения в работе сокращателя ссылок:

- Контент веб-версии сообщения теперь передается без модификаций. Это значит, что действия с ней будут отображаться в карточке профиля как **действия с контентом**. Ссылки на веб-версии по прежнему передаются с параметром `dummy=1`;
- Формат ссылок изменен с `^(/s/[A-Za-z0-9_-]{5,20})$` на `^(/s[A-Za-z0-9_\-!~\.]{10,12})$`.

Добавлено

- Добавлено динамическое отображение статуса процессов при запуске и остановке платформы с помощью `akd`.
- Теперь над элементами сценария отображается количество вошедших и вышедших профилей.
- В меню вставки переменных доступны переменные **имени** и **ID сценария**, а также **ID элемента** и его **тип**.
- Теперь у пользователей есть возможность **переопределять шаблоны формы** при помощи `custom`-файлов.

- Добавлен **журнал запусков** для триггерного элемента сценария “Захват по расписанию”.
- В карточке профиле теперь отображается информация об ошибках сценариев, если профиль находился в ГКГ.
- Для API-запросов на получения списка сценариев и информации о шаблоне добавлены параметры `updated_user` и `update_time`.

Улучшено

Улучшения пайплайнов пользовательских каналов

Значительно улучшена работа **пайплайнов пользовательских каналов**. Внесены следующие изменения:

- Исправлены ошибки, возникающие в ходе работе пайплайнов.
- Добавлен доступ к именам объектов.
- Теперь пайплайн может принимать дату событий в объекте EventObject.
- Добавлен уникальный идентификатор сообщения в пайплайне, совпадающий с ID в ClickHouse.

Другие улучшения

- Теперь в файле конфигурации можно переопределить **Firebase URL для каждого аккаунта платформы**.
- Интерфейс **поиска и фильтрации** по платформе был обновлен.
- Произведен **рефакторинг работы сокращателя ссылок**. Устранена дубликация ID ссылок и сокращена максимальная длина ссылки до 14 символов.
- Константа `maxAmountOfChanges` вынесена из `segstat_recalc` в отдельный параметр конфигурации — `SEGSTATIC_PROFILE_CHANGES_BATCH_SIZE`.
- Оптимизирована работа **ML-сервиса Best Send Time**. Связанные с сервисом параметры конфигурации также были изменены.
- Улучшена производительность платформы.
- Улучшена стабильность процессов `procworkflow`, `webcontrol` и `proctask`.

Прочее

- Статические сегменты, созданные через API, теперь корректно пересчитываются.
- Исправлен неверный расчет сегментов с одинаковым названием файлов.
- В базах профилей теперь корректно отображается дата последнего обновления.
- Триггер захвата подписок в сценарии теперь учитывает выбранный канал и ресурс.
- Исправлена JavaScript-ошибка на странице выбора типа кампании при создании кампании из календаря.
- Сессия в веб-интерфейсе больше не завершается слишком быстро.
- Функция `adddate` теперь корректно работает в элементе сценария “Вызов API”.
- JSON-фильтр теперь поддерживает проверку полей с символом подчеркивания в имени.
- Push-шаблон, импортированный через API, теперь можно открыть в бета-редакторе.
- Параметры `not_strict` и `subscription_filter` в сценариях теперь работают корректно.
- AKMTA теперь корректно обрабатывает DSN при использовании адреса из поля `From` в `MAIL FROM`.
- Исправлена ошибка 500 при запросе веб-версии через API для профилей в стоп-списке.
- Исправлена ошибка, возникающая при размещении пикселя через Менеджер тегов.
- Исправлен парсинг ответа по статусам от сендера SMSTraffic.
- При парсинге .xls файлов номера телефонов теперь корректно читаются.

- Значение `send_message_id` в custom JSON для Push теперь совпадает с `send_message_id` из хука захвата событий.

References

v2025.2.72.707: PL-18764 PL-18548 PL-18710 PL-18763 PL-18671 PL-18659 PL-17043 PL-16495 PL-17891 PL-18293 PL-18486 PL-18752 PL-18674 PL-18761 PL-18095 PL-18634 PL-18661 PL-18548 PL-18531 PL-17937 PL-18383 PL-18264 PL-18449 PL-17595 PL-17827 PL-18750 PL-13124 PL-18737 PL-18070 PL-17894 PL-17772 PL-18132 PL-18672 PL-18703 PL-18712 PL-18669 PL-18619 PL-18613 PL-18648 PL-18546 PL-13124 PL-18573 PL-15659 PL-18734 PL-18705 PL-18693 PL-18668 PL-17784 PL-18662 PL-18608 PL-17418 PL-16610 PL-18697 PL-18043 PL-18685 PL-18683 PL-18676 PL-18656 PL-18605 PL-18524 PL-18660 PL-18658 PL-18447 PL-18494 PL-18632 PL-15330 PL-18592 PL-17661 PL-17120 PL-18534 PL-18446 PL-18304 PL-18261 PL-17705 PL-18417 PL-15337 PL-18164

---

v2025.2.72.1475 – password recovery and billing info

CAUTION

В рамках обновления была изменена структура хранимых данных, поэтому для корректной работы платформы требуется провести **миграцию**. Для этого остановите платформу и выполните следующую команду:

```
./aktool migrate add-updatefeed --config=/opt/MarketingPlatform/config/main.json
```

Не запускайте миграцию несколько раз! Миграция проводится **однократно**. При возникновении ошибок обратитесь к команде технической поддержки.

Добавлено

- Добавлены метрики времени обработки каждого профиля на каждом этапе кампании.
- К файлу журнала при ротации добавлен номерной постфикс.
- Для контента добавлены переменные `{segid}` и `{segname}` — **ID** и **название сегмента**.
- Добавлен новый параметр `PROCAUTH_WELCOME_LINK_TTL_SEC` — время жизни ссылки завершения регистрации и установки пароля.
- Добавлены дополнительные логи к процессу обработки Reply в AKMTA.
- Добавлены новые индексы в MongoDB для наиболее частых запросов для улучшения стабильности платформы.
- Добавлена интеграция с SMS-отправщиком **7tech**.
- Добавлены логи к запросам в Notify для получения статуса интеграции.

Улучшено

- Убрана сортировка профилей при наполнении статического сегмента
- Расширена информация в уведомлении о заморозке и деактивации аккаунта.
- Обновлена страница для смены пароля, а также письмо-уведомление об успешной смене.
- Улучшена стабильность процесса `proctask`.
- Дополнительные поля товаров Маркета теперь импортируются с корректным типом данных.

Прочее

- Исправлены некоторые UI/UX-ошибки в интерфейсе.
- Флаг `--faster` команды `./akd restart` снова работает корректно.
- Устранена ошибка, при которой в элементе “JSON фильтр” не отображается условие, если в нем есть обращение к массиву.
- Изображения в шаблон Viber снова корректно вставляются через галерею.
- Устранены ложные срабатывания триггеров, связянных со связями профилей.
- Время запуска в расписании кампаний теперь расчитывается верно, даже при использовании часового пояса, отличного от часового пояса аккаунта.
- Исправлены ошибки, возникающие при попытках пересчитать сегмент профилей.
- Сценарии теперь корректно сохраняются при использовании элемента “Захват профилей при добавлении/изменении подписки” с фильтром по пользовательским каналам.
- Устранена JS-ошибка, возникающая при открытии элемента “Сегмент по расписанию”.
- Теперь AKMTA сохраняет письма, пришедшие с ящика постмастера.
- Для сендера Sendsay добавлен хедер `X-Letter-Id`.
- В карточке профиля снова отображаются события отписки по каналу Viber™*.
- Функция `uppercasefirst` теперь корректно работает с полями профилей.
- Исправлены ошибки, связанные с отправкой писем после заполнения форм.
- Превью Telegram-шаблона снова корректно работает.
- Исправлена ошибка, при которой в ГКГ отображались неверные данные.
- Устранены ошибки, связанные с ролями доступа и правами пользователей.

References

v2025.2.72.1475: PL-18764 PL-18548 PL-18710 PL-18763 PL-18671 PL-18659 PL-17043 PL-16495 PL-17891 PL-18293 PL-18486 PL-18752 PL-18674 PL-18761 PL-18095 PL-18634 PL-18661 PL-18548 PL-18531 PL-17937 PL-18383 PL-18264 PL-18449 PL-17595 PL-17827 PL-18750 PL-13124 PL-18737 PL-18070 PL-17894 PL-17772 PL-18132 PL-18672 PL-18703 PL-18712 PL-18669 PL-18619 PL-18613 PL-18648 PL-18546 PL-13124 PL-18573 PL-15659 PL-18734 PL-18705 PL-18693 PL-18668 PL-17784 PL-18662 PL-18608 PL-17418 PL-16610 PL-18697 PL-18043 PL-18685 PL-18683 PL-18676 PL-18656 PL-18605 PL-18524 PL-18660 PL-18658 PL-18447 PL-18494 PL-18632 PL-15330 PL-18592 PL-17661 PL-17120 PL-18534 PL-18446 PL-18304 PL-18261 PL-17705 PL-18417 PL-15337 PL-18164

---

v2025.2.72.1664 – interface improvements

CAUTION

В рамках обновления была изменена структура хранимых данных, поэтому для корректной работы платформы требуется провести **миграцию**. Для этого остановите платформу и выполните следующую команду:

```
./aktool migrate add-updatefeed --config=/opt/MarketingPlatform/config/main.json
```

Если вы уже проводили миграцию, не запускайте ее повторно! Миграция проводится **однократно**. При возникновении ошибок обратитесь к команде технической поддержки.

Улучшено

Исправления интерфейса

Исправлены некоторые UI-/UX-ошибки:

- Добавлена локализация вместо системных названий в некоторых местах;
- Улучшено отображение некоторых предупреждений и сообщений об ошибках;
- Добавлена ссылка на документацию для подключений Basic Authentication и Token Authentication;
- Обновлены шрифты и стили некоторых текстов в платформе;
- Устранено наложение некоторых элементов сценариев.

Прочее

- Отправки WhatsApp с кнопкой типа “PHONE” в EDNA снова работают корректно.
- Устранена ошибка, при которой не импортировались шаблоны с изображениями из архива
- Теперь значение поля при создании промокодов сохраняется корректно.
- Оптимизирована работа процесса `procintegras`.
- Исправлены недочеты в работе счетчиков очереди сендеров.
- Улучшена стабильность “Изменение значения поля”.
- Устранены некоторые проблемы, связанные с превью шаблонов.
- Добавлена возможность сохранить коннектор к внешней БД с SSL-подключением без указания сертификатов.
- Устранена ошибка, возникающая при попытке посмотреть аналитику по цели с косыми чертами в имени.
- Просмотр API-контента через элемент сценария снова работает корретно.
- Исправлена ошибка `empty query` в SQL-запросах на сегментацию.

References

v2025.2.72.1664: PL-18416 PL-19009 PL-18915 PL-18990 PL-18567 PL-18183 PL-18913 PL-18978 PL-18338 PL-18867 PL-18899 PL-18885 PL-18537 PL-18072 PL-18864 PL-18861 PL-18875 PL-18825 PL-12690 PL-18240 PL-16060 PL-18944 PL-18878 PL-18873 PL-18722 PL-18832 PL-18881 PL-15383 PL-18351

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*
***Viber является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

v2025.1.71

v2025.1.71.0 - Global control groups, administrative panel rework, NPS testing

CAUTION

Для корректной работы платформы после обновления вам необходимо провести **миграцию**, для этого:

1. Перед запуском `procevent` выполните миграцию данных для ClickHouse:

```
/opt/MarketingPlatform/ak/bin/aktool profile_history add_gcg_id_column --config
/opt/MarketingPlatform/config/main.json
/opt/MarketingPlatform/ak/bin/aktool history_storage add_gcg_id_column  --config
/opt/MarketingPlatform/config/main.json
/opt/MarketingPlatform/ak/bin/aktool history_storage add_profile_marks --config
/opt/MarketingPlatform/config/main.json
```

Запустить платформу можно будет только после выполнения миграций ClickHouse.

2. После миграции ClickHouse выполните `aktool` **из новой сборки** для миграции MongoDB:

```
./aktool --config config/main.json profiles migrate-gcg-flags
```

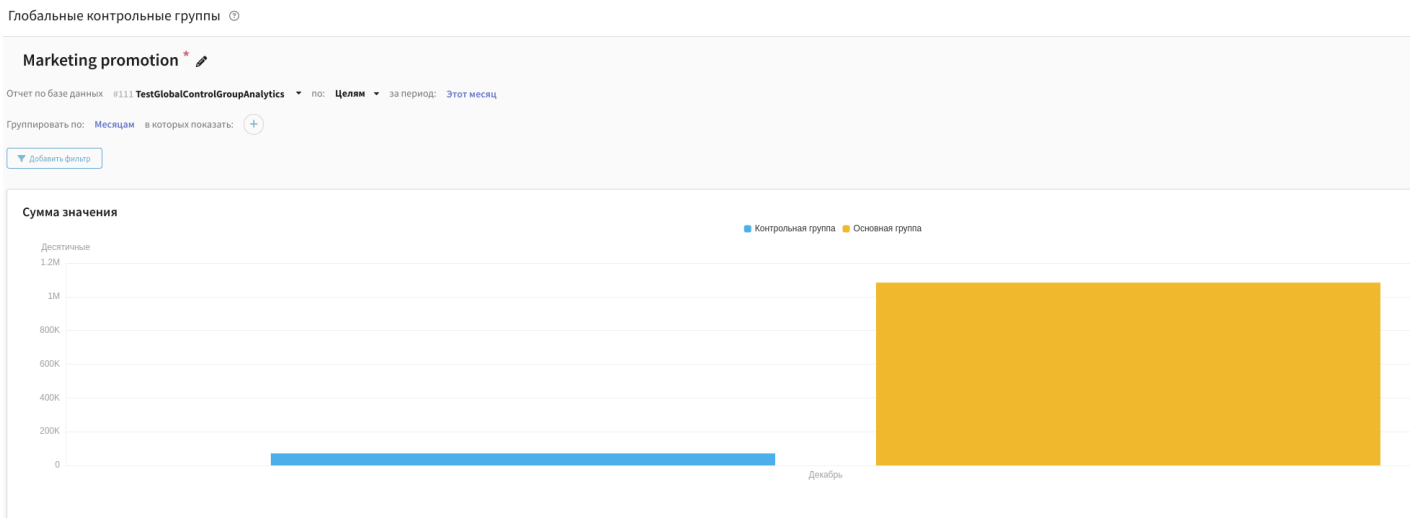
Обратите внимание, что миграция MongoDB может занять от одного до нескольких часов. Функционал глобальных контрольных групп будет доступен только после завершения миграции MongoDB.

3. Запустите `procevent`.

Основные изменения

Механизм выделения глобальных контрольных групп

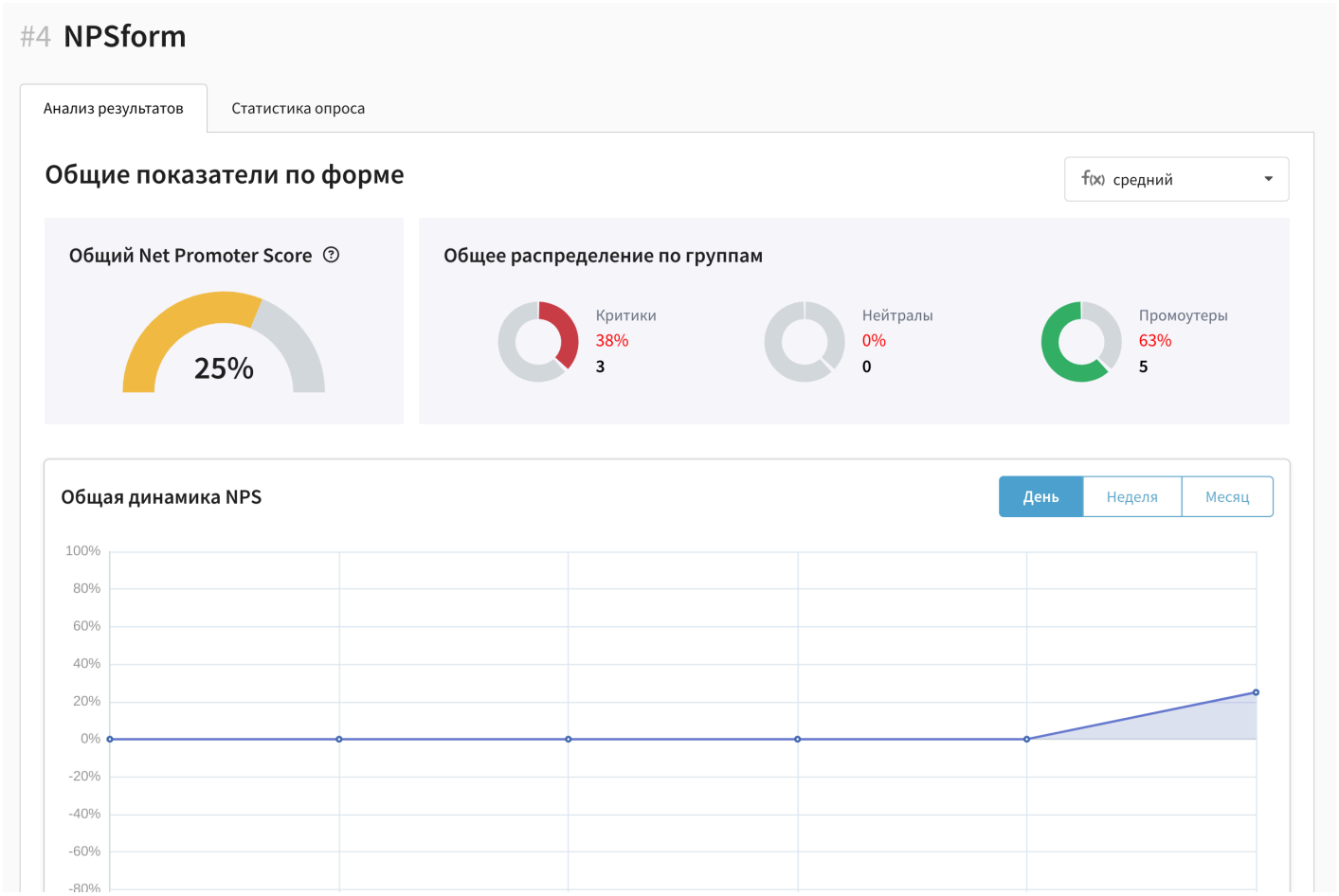
В платформе появилась функция **выделения глобальных контрольных групп**. **Глобальные контрольные группы** — это часть клиентской базы, которая временно не получает маркетинговые коммуникации с целью анализа эффективности маркетинговых стратегий. Спустя некоторое время пользователь может сравнить ключевые метрики основной части клиентов и контрольной группы. В том случае, если показатели групп отличаются незначительно, маркетолог может сделать вывод о неэффективности проводимых им кампаний:



Подробнее о глобальных контрольных группах можно прочитать здесь.

NPS-опросы в формах

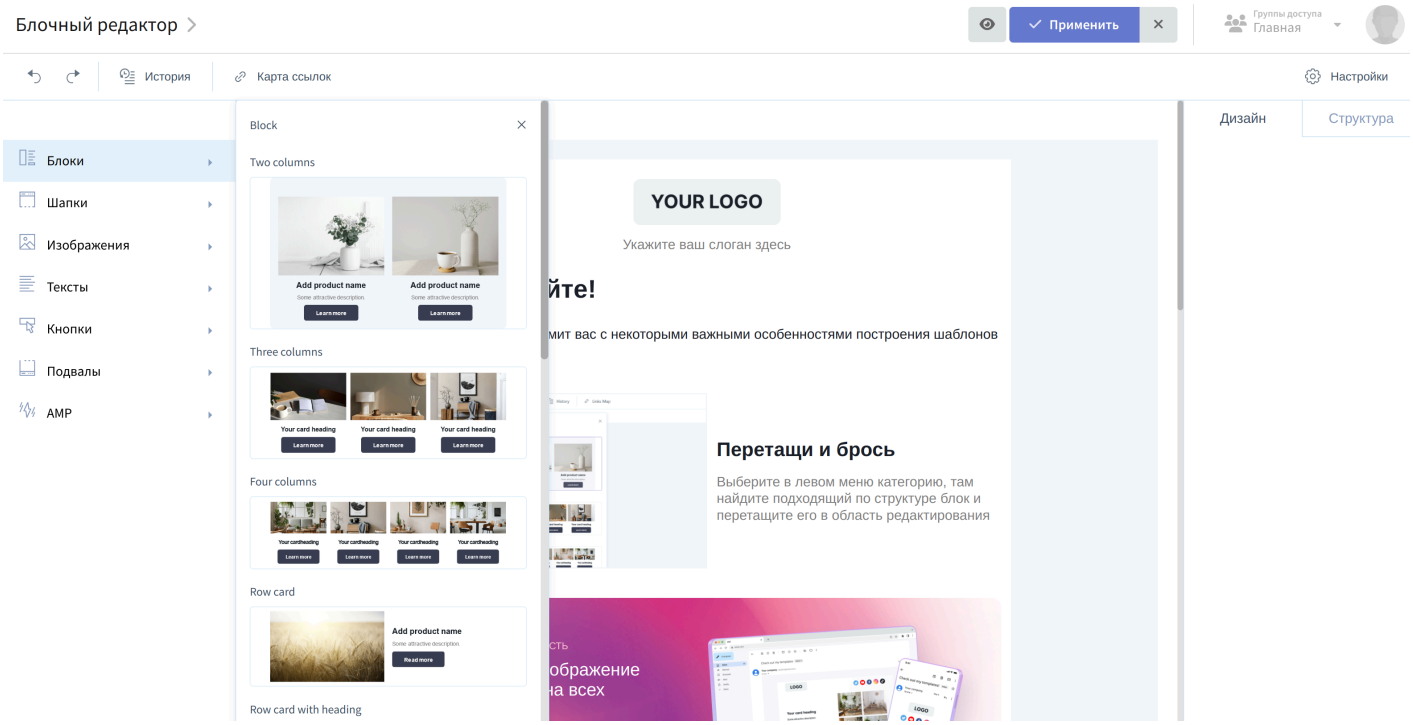
Теперь вы можете добавить **NPS-опрос** в веб-формы, созданные в платформе Altcraft. Такие опросы помогают вычислить индекс Net Promoter Score (индекс потребительской лояльности) — метрику, показывающую приверженность клиентов компании или товару. Этот показатель является одним из главных индексов измерения клиентской лояльности. NPS позволяет проанализировать сильные и слабые стороны продукта и скорректировать маркетинговую стратегию:



Чтобы узнать, как настраивать NPS-опросы в платформе, читайте статью или воспользуйте видеогайдом.

Блочный email-редактор

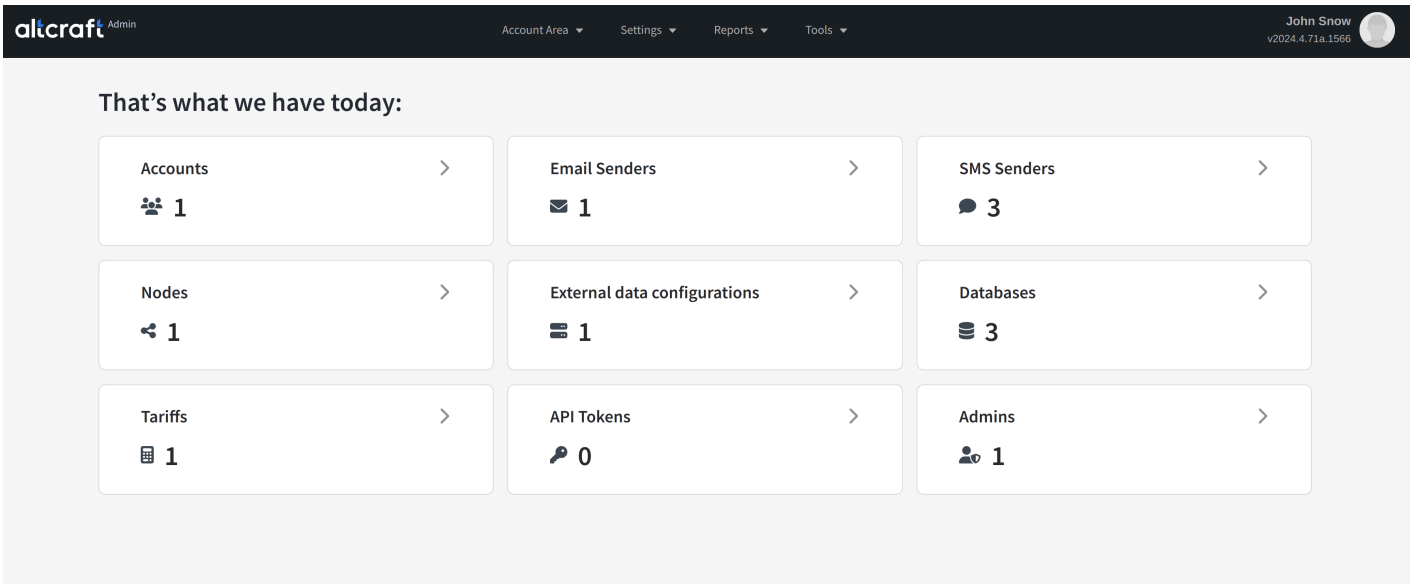
В редактор шаблонов для email-сообщений добавлен **блочный редактор**. Он позволяет быстро и просто создать шаблон электронного письма, который корректно отобразится на всех типах устройств. Также редактор поддерживает AMP-содержимое для ваших писем:



Больше о блочном редакторе можно прочитать здесь.

Переработана панель администратора

Панель администратора значительно переработана: изменен интерфейс, добавлен новый функционал, также полностью обновлен движок панели:



WARNING

Начиная с версии 72, платформа потребует использование Redis-совместимого хранилища, мы предлагаем использовать **Kvrocks**. Документация по Kvrocks доступна здесь.

Интеграция с SMS-провайдером Stream Telecom

Добавлена возможность настроить интеграцию с SMS-провайдером **Stream Telecom**. Для создания интеграции настройте сендер соответствующего типа в Панели администратора.

Другие изменения

- Доработаны массовые операции в платформе — активация, деактивация и удаление для таких элементов, как сценарии, шаблоны, пользователи и т.д.
- Для запросов API 1.1. добавлена поддержка передачи заголовков Request-ID и Token, также улучшена работа дедупликации запросов.
- Обновлен интерфейс подключений.
- Добавлено предупреждение при попытке удаления профилей, которые недавно участвовали в кампаниях.
- Обновлен интерфейс настроек расписания для некоторых разделов платформы.
- Нажатие кнопки “Сохранить” теперь не закрывает редактор и не перенаправляет пользователя в меню раздела.

Исправления

- Исправлена ошибка, при которой в платформе не сохранялся статус настроек машинного обучения.
- Теперь пользователь не может удалять сущности из корзины, если аккаунт заморожен.
- Устранена ошибка, из-за которой при кликах в SMS и push-сообщениях регистрировались лишние события доставки.
- Исправлены ошибки в работе переменных Менеджера тегов.
- Теперь сценарии, настроенные на захват при активации пикселя, работают корректно.
- Внесены некоторые изменения в Rich Push-редактор.
- Устранена ошибка, при которой во время импорта продуктов с пользовательскими полями происходила попытка повторно создать поле.
- Экранирование ссылки с параметрами внутри пикселя теперь работает корректно.
- Улучшена стабильность процессов `procworkflow`, `procintegras`, `proctask`, `proctrigger`, `procevent`, `akmtad`.
- Устранена ошибка, при которой Viber™-кампания с SMS Traffic отправляла сообщение только одному подписчику.
- Теперь информация об участии профиля в сценарии обновляется корректно.
- Исправлена проблема, из-за которой данные для входа не отображались в приветственном письме.
- Сообщения в очередях RabbitMQ снова обрабатываются корректно.
- Добавлена валидация для расширения файла при создании сегмента по условию “В файле в колонке”.
- Специальный символ в названии файла больше не приводит к ошибкам при запросе к таблице данных.
- SMS-шаблон теперь сохраняется даже с незаполненным полем “Лимит на количество SMS”.
- Устранены проблемы с запуском тестовой отправки для Telegram-каналов.
- Данные формы теперь не дублируются при передаче в платформу.
- Исправлена ошибка, из-за которой в push-сообщениях для Mozilla Firefox не работал переход по ссылке.
- Запуск Telegram-кампаний теперь использует меньше памяти.
- Улучшена стабильность модуля Аналитика: исправлены различные ошибки, связанные с выгрузкой отчетов и сбором данных.
- Функция `$dateNowFunc` в JSON-запросах снова работает корректно.
- Исправлена ошибка, при которой к нескольким объектам не добавлялся тег.
- Изображение с превью шаблона теперь выгружается корректно.
- Исправлены ошибки в работе API-запросов `market/sku/import`, `profiles/segstatic/list`.

- Добавлена валидация для удаления внешних баз данных.
- В редактора push-сообщений больше не пропадают Deep Link при сохранении шаблона.
- Теги в сценариях теперь отображаются корректно.
- Устранена ошибка, при которой кампания не открывалась при переходе из отчета или истории профиля.
- Замороженные в сценарии профили теперь не удаляются при изменении сценария.
- Исправлена ошибка, при которой AKMTA отправлял пустые сообщения.
- Условие ожидания дня недели снова корректно работает в сценариях.
- Улучшена валидация email-адресов в списке получателей скрытой копии письма.
- События получения и открытия Push-сообщений регистрируются корректно.
- Оптимизирован процесс генерации и отправки Push-сообщений.
- Улучшено логирование некоторых процессов.
- Улучшения интерфейса: исправлены проблемы с отображением некоторых элементов, устранены ошибки и опечатки в текстах платформы.

References

PL-17608 PL-17599 PL-17574 PL-17572 PL-17570 PL-17564 PL-17558 PL-17556 PL-17549 PL-17526 PL-17524 PL-17519 PL-17518 PL-17516 PL-17508 PL-17497 PL-17494 PL-17487 PL-17486 PL-17470 PL-17459 PL-17455 PL-17448 PL-17429 PL-17426 PL-17424 PL-17410 PL-17402 PL-17398 PL-17374 PL-17373 PL-17372 PL-17369 PL-17327 PL-17322 PL-17305 PL-17303 PL-17287 PL-17285 PL-17283 PL-17282 PL-17278 PL-17277 PL-17259 PL-17258 PL-17247 PL-17240 PL-17220 PL-17208 PL-17187 PL-17170 PL-17132 PL-17126 PL-17114 PL-17084 PL-17078 PL-17074 PL-16987 PL-16979 PL-16954 PL-16953 PL-16930 PL-16907 PL-16854 PL-16701 PL-16700 PL-16679 PL-16641 PL-16640 PL-16607 PL-16600 PL-16579 PL-16558 PL-16542 PL-16539 PL-16537 PL-16523 PL-16515 PL-16513 PL-16498 PL-16493 PL-16479 PL-16477 PL-16468 PL-16460 PL-16430 PL-16414 PL-16410 PL-16406 PL-16404 PL-16403 PL-16386 PL-16358 PL-16350 PL-16348 PL-16344 PL-16328 PL-16296 PL-16247 PL-16201 PL-16144 PL-16069 PL-16024 PL-16003 PL-15998 PL-15997 PL-15994 PL-15980 PL-15944 PL-15852 PL-15829 PL-15825 PL-15777 PL-15647 PL-15639 PL-15557 PL-15498 PL-15404 PL-15379 PL-15378 PL-15370 PL-15327 PL-15245 PL-15021 PL-14968 PL-14230 PL-13369 PL-12949 PL-12732 PL-12297 PL-17562 PL-17634 PL-17579 PL-17613 PL-17036 PL-17576 PL-13434 PL-17358 PL-17135 PL-16695

---

v2025.1.71.255 - Mandatory use of the resource in campaigns

CAUTION

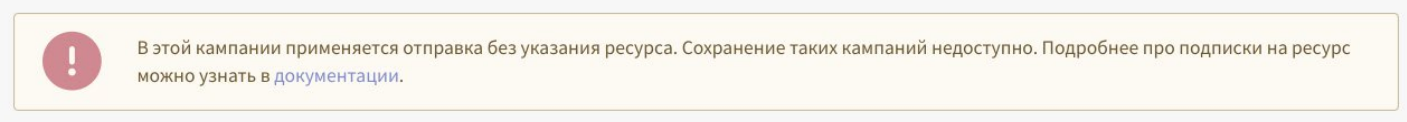
Начиная со следующего обновления, запуск кампаний без привязанного ресурса будет **невозможна!**

Если вы пользовались рассылками без использования ресурса, пожалуйста, привяжите к ним **ресурс**.

Основные изменения

Обязательное использование ресурса в кампаниях

Со следующего обновления все кампании в платформе будут использовать **ресурс**. В связи с этим, сохранение кампаний без указания ресурса теперь невозможно:



Другие изменения

- Улучшено логирование процесса обработки Reply в AKMTA.
- Добавлен параметр конфигурации, отвечающий за максимальный размер выгружаемых данных из колонки SQL-таблицы.

Исправления

- Исправлены проблемы с отображением некоторых элементов интерфейса.
- Улучшена работа элементов сценария “Захват профилей при активации пикселя” и “Обновление профиля”, а также кампаний, запускаемых из сценария.
- Устранены ошибки, возникающие при регистрации событий Viber™ в интеграции с SMS Traffic.
- Viber™-интеграция с Rapporto теперь работает корректно на всех аккаунтах.
- Оптимизирован процесс разбора очереди `procevent`.
- Исправлена ошибка, при которой удаление профилей было доступно пользователям без соответствующих прав.
- Счетчик профилей в сегменте теперь работает корректно при отправке через Telegram-бота.
- Устранена ошибка, вследствие которой данные, полученные из push-сообщений, не полностью передавались в платформу.
- Улучшена стабильность процессов `procworkflow` и `proctask`.

References

PL-17644 PL-17561 PL-17618 PL-17648 PL-17643 PL-17640 PL-16953 PL-17552 PL-17557 PL-17508 PL-17481 PL-17467 PL-17270 PL-17039 PL-17031 PL-16369 PL-15722 PL-17164 PL-17650 PL-17597 PL-17360 PL-17680 PL-15760 PL-17505 PL-16454

---

v2025.1.71.507 - Restructuring the policy counter repository

CAUTION

Для корректной работы платформы после обновления вам необходимо провести **миграцию**. До запуска новой версии сделайте следующие действия:

1. Остановите необходимые сервисы:

```
./akd stop {proctrigger,procworkflow,proctask}
```

2. Запустите `aktool` **из новой сборки** с помощью команды:

```
./aktool migrate policystore --config /opt/MarketingPlatform/config/main.json
```

3. Перезапустите сервисы платформы:

```
./akd restart
```

Справка по мигратору:

```
Migrate policystore db to accounts dbs collections policy_contact, policy_profile

Usage:
  aktool migrate policystore [flags]
  aktool migrate policystore [command]

Available Commands:
  ensure_policystore_indexes Ensures policystore migrate indexes created

Flags:
      --account int          account id
      --batch_interval int   batch interval in seconds (default 1)
      --batch_size int       batch size (default 10000)
      --batch_workers int    number of batch workers (default 8)
  -h, --help                 help for policystore
      --workers int          number of workers for accounts (default 8)

Global Flags:
      --clickhouse_database_name string   usage database for clickhouse
      --config string                     config path (default
"/opt/akd/config/main.json")
  -o, --output string                     file path to output in the result of the
command (default "stdout")
      --truncate                          whether to truncate the file specified by --
output

Use "aktool migrate policystore [command] --help" for more information about a command.
```

Изменения

- Проведена оптимизация структуры хранилища счетчиков политик отправки.
- Метрики по API-запросам вызова в сценарий теперь доступны в аналитике.
- Теперь специальные символы могут использоваться в названиях сущностей.
- Исправлены некоторые ошибки в функционале и интерфейса Панели администрирования.
- Улучшено логирование некоторых процессов платформы.

Исправления

- Устранена ошибка, при которой push-кампании генерировали большое количество несуществующих недоставок.
- Статусы в SMS-интеграции с Devino Telecom теперь обрабатываются корректно.
- Устранены ошибки, возникающие при тестировании кампаний пользовательских каналов.
- Аналитика сущности с HTML-тегом в названии снова работает корректно.
- Устранены ошибки, возникающие при регистрации событий в Viber-интеграции с SMS Traffic.
- Улучшена стабильность методов Web Push SDK.
- Исправлены некоторые ошибки интерфейса.

- Улучшена стабильность процесса `proctask`.

References

PL-17756 PL-17753 PL-17446 PL-17727 PL-17731 PL-17728 PL-17715 PL-17720 PL-17586 PL-17718 PL-17711 PL-17706 PL-17704 PL-17648 PL-17698 PL-17677 PL-17670 PL-17656 PL-17642 PL-17626 PL-17641 PL-17249 PL-17272 PL-17631 PL-17619 PL-17507 PL-17211 PL-17111 PL-16993 PL-16354

---

v2025.1.71.627 - RMQ buffer base mode

Изменения

- Добавлен режим буфферной базы для RMQ в `prochook`.
- Метрики по API-запросам вызова в сценарий теперь доступны в аналитике.
- Теперь специальные символы могут использоваться в названиях сущностей.
- Добавлена возможность конфигурации `global qos` для RMQ.
- Исправлены некоторые ошибки в функционале и интерфейсе Панели администрирования.
- Улучшено логирование некоторых процессов платформы.

Исправления

- Устранена ошибка, при которой push-кампании генерировали большое количество несуществующих недоставок.
- Статусы в SMS-интеграции с Devino Telecom теперь обрабатываются корректно.
- Устранены ошибки, возникающие при тестировании кампаний пользовательских каналов.
- Аналитика сущности с HTML-тегом в названии снова работает корректно.
- Устранены ошибки, возникающие при регистрации событий в Viber-интеграции с SMS Traffic.
- Улучшена стабильность методов Web Push SDK.
- Исправлена ошибка, связанная с некорректным отображением статуса неактивной кампании.
- Исправлены некоторые ошибки интерфейса.
- Улучшена стабильность процесса `proctask`.

References

PL-17756 PL-17753 PL-17446 PL-17727 PL-17731 PL-17728 PL-17715 PL-17720 PL-17586 PL-17718 PL-17711 PL-17706 PL-17704 PL-17648 PL-17698 PL-17677 PL-17670 PL-17656 PL-17642 PL-17626 PL-17641 PL-17249 PL-17272 PL-17631 PL-17619 PL-17507 PL-17211 PL-17111 PL-16993 PL-16354

---

v2025.1.71.873 - SSL support for CH

Основные изменения

Поддержка SSL для ClickHouse

Добавлена поддержка SSL для подключения к БД ClickHouse. Для её настройки активируйте соответствующий параметр в Панели администратора.

Другие изменения

- Глобальная контрольная группа теперь сохраняет группы, назначенные для исходной базы профилей.

Исправления

- События пользовательских каналов теперь корректно отображаются в истории профиля.
- Улучшена работа отправок в Telegram-ботах в сценариях.
- Тестирование кампании внутри сценария снова работает корректно.
- Исправлена ошибка, при которой поля подписки на пользовательский канал не сохранялись.
- Связи теперь корректно работают в шаблонах сообщений.
- Устранена ошибка, при которой параметр `send_message_id` принимал разные значения для одного и того же сообщения в захвате событий.
- Отбор по номеру телефона внутри сценариев снова работает корректно.
- Исправлена ошибка, вследствие которой параметры политики отправки могли произвольно меняться.
- Некоторые улучшения и исправления в работе Панели администратора
- Устранены ошибки, связанные с установкой формата для чисел с плавающей точкой.
- Улучшена работа истории профилей и некоторых логов.
- Устранено непреднамеренное изменение приоритета подписки при ее изменении через API.

References

PL-17067 PL-17351 PL-17361 PL-17888 PL-17876 PL-17814 PL-17801 PL-17536 PL-17748 PL-17532 PL-17798 PL-17723 PL-17433 PL-17824 PL-17719 PL-17846 PL-17555 PL-17810 PL-17774 PL-17757 PL-17755 PL-17587 PL-17447 PL-7853PL-17795 PL-17801 PL-17627 PL-17703 PL-17796 PL-17707 PL-17729 PL-17472 PL-17708

---

v2025.1.71.1059 - Scenario API improvements

Основные изменения

Улучшение API-запросов, связанных со сценариями

- Добавлены запросы на активацию и деактивацию сценария: `/workflows/activate` и `/workflows/deactivate`.
- Теперь API-запросы сценариев возвращают в ответе идентификатор кампании и шаблона, а также информацию о переходе в другой сценарий.

Другие изменения

- Максимальная длина названия для сценариев увеличена до 156 символов.
- Улучшено логирование некоторых процессов платформы.

Исправления

- Улучшена верстка некоторых элементов интерфейса.
- Устранена ошибка, при которой невозможно было провести тестовый запуск кампании с динамическим сегментом.
- Авторизация по LDAP снова работает корректно.
- Теперь фильтр по кампаниям находит те кампании, которые были созданы внутри сценария.
- Устранены ошибки, возникающие при попытке ресинхронизации сегментов Яндекс.Аудиторий.

- Исправлена ошибка, из-за которой время публикации и скрытия формы в настройках задавалось некорректно.
- Трекинг сервисы снова корректно прослушивают переданные через параметры адрес и порт.

References

PL-17896 PL-17938 PL-16077 PL-17925 PL-17832 PL-17831 PL-17823 PL-16045 PL-16896 PL-17773 PL-17765 PL-9599 PL-17930 PL-17482 PL-17899 PL-17538 PL-17291 PL-17762 PL-17668 PL-17664 PL-17636 PL-17928 PL-17881 PL-17868 PL-17834

---

v2025.1.71.1487 - API calls optimization, improved logging

Изменения

- Расширена поддержка типов данных в интеграциях с таблицами ClickHouse.
- Добавлена возможность указывать время запуска глобальных контрольных групп в файле конфигурации платформы.
- Теперь платформа собирает в логах информацию о времени обработки каждого профиля на всех этапах кампании.
- Добавлена кнопка остановки перерасчета сегмента.

Исправления

- Исправлена ошибка, при которой данные об удаленном поле профиля все равно возвращались через API.
- Больше не появляются дубликаты связей при их создании через API.
- Оптимизирован API-метод `/profiles/import_batch`.
- В элементе сценария “API-вызов” переменная с форматированной датой снова работает корректно.
- Исправлены ошибки, связанные с отображением шрифтов в попапах.
- Ссылки на веб-версии писем снова корректно работают в тестовых отправках.
- Исправлены некоторые ошибки интерфейса.
- Ошибки рендера теперь корректно отображаются в журнале кампаний.
- Устранена ошибка, при которой шаблоны и кампании не перемещались в корзину.
- Для элемента сценария “Обновление профиля” теперь можно установить нулевое значение.
- Тег `blockquote` корректно обрабатывается в Telegram-шаблонах.
- Исправлены некоторые ошибки, связанные с шаблонами пользовательских каналов.
- Улучшен сбор ошибок по отправкам.
- Функция `format` теперь корректно работает в циклах.
- Улучшена стабильность процесса `webcontrol`.

References

PL-17984 PL-18003 PL-17901 PL-17947 PL-15365 PL-13140 PL-17926 PL-17900 PL-17945 PL-17568 PL-17684 PL-17709 PL-17052 PL-17356 PL-15570 PL-17949 PL-17254 PL-17786 PL-17663 PL-17701 PL-17553 PL-18018 PL-17953 PL-17996 PL-17970 PL-16377 PL-17931 PL-17525 PL-17968 PL-17938 PL-16892 PL-17965 PL-17097 PL-17280 PL-17808 PL-17841 PL-17767 PL-17825 PL-17766 PL-17785 PL-11838 PL-17241 PL-17074 PL-17738 PL-17689 PL-17603 PL-15518 PL-8650 PL-16616

---

v2025.1.71.2447 - SberMobile, Intellin integrations

> **CAUTION**
>
> Для корректной работы платформы после обновления вам необходимо провести **миграцию**, для этого выполните команду `./aktool migrate tags`. Миграция проходит в 2 этапа:
>
> - миграция сущностей в MongoDB
> - миграция сущностей в PostgreSQL

Основные изменения

Новые интеграции: СберМобайл и Intellin

Добавлены следующие интеграции:

- интеграция с SMS-провайдером **Intellin**;
- интеграция с SMS-шлюзом от **СберМобайл**.

Улучшение сбора данных по отправкам и сценариям

- Добавлены **логи к запросам в Notify** для получения статуса интеграции. С их помощью вы сможете понять, почему ваша интеграция перестала быть активной.
- В заголовки AKMTA добавлен **USMID**. USMID — уникальный идентификатор сообщения, в котором содержится подробная информация об отправке. Зная его, вам будет проще анализировать причины возможных ошибок отправки.

Несколько ссылок в условии “Клик по ссылке”

Теперь пользователь может указывать списком ссылки для условия сегментации “Клик по ссылке”.

Другие изменения

> **CAUTION**
>
> Если вы указывали адрес в параметре RPC_HOST_CLI, то все клиенты (даже с других хостов) будут подключаться по этому адресу. Перед установкой обновления обязательно укажите для этого параметра публичный или “серый” адрес.

- RPC-клиент `procworkflow` теперь подключается по адресу, установленному в специальном параметре конфигурации.
- Добавлена возможность отправлять батчем веб-версии в хранилище S3.

Исправления

- Исправлены ошибки, связанные с двухфакторной аутентификацией в Панели администратора.
- Улучшена стабильность работы интеграции с Tilda.
- Исправлена ошибка, при которой при запуске сценария не пересчитывался сегмент по расписанию.
- Изменена логика отправки отчетов по заполнениям форм, повышена точность сбора данных.

- Функция `format` в циклах и в сочетании с `apicontent` теперь работает корректно.
- Улучшена регистрация кликов в Telegram-ботах.
- Статусы отправок, а также сообщения-дубликаты снова корректно регистрируются для SMS-рассылок.
- Устранены ситуации, при которых пользователь без соответствующей роли получал доступ к сущностям в платформе.
- Тестовые кампании с динамическим сегментом подписчиков снова запускаются корректно.
- Исправлены некоторые UI/UX ошибки в интерфейсе.
- Устранены ошибки, связанные с использованием условий в сегментах и сценариях.
- Пользовательские bounce-правила для AKMTA снова работают корректно.
- Улучшена стабильность процессов `webcontrol`, `procintegras`, `proctask`, `prochook` и `procworkflow`.

References

PL-18148 PL-18022 PL-18189 PL-18131 PL-18102 PL-17352 PL-17981 PL-17931 PL-17397 PL-17357 PL-15357 PL-15143 PL-18194 PL-17998 PL-18130 PL-15386 PL-18079 PL-18077 PL-18178 PL-18165 PL-18123 PL-18105 PL-18112 PL-17821 PL-17615 PL-18054 PL-17695 PL-18048 PL-18047 PL-15625 PL-17771 PL-17735 PL-17770 PL-17689 PL-17653 PL-15344 PL-14393 PL-12555 PL-18122 PL-18127 PL-18098 PL-17804 PL-18060 PL-18084 PL-17799 PL-18050 PL-17976 PL-17983 PL-17721 PL-17395 PL-16585 PL-15340 PL-18090 PL-17957 PL-17275 PL-17803 PL-18097 PL-17463 PL-18074 PL-17179 PL-15009 PL-17861 PL-17977 PL-17874 PL-17400 PL-17003 PL-16578 PL-16589 PL-16022 PL-17105 PL-17489 PL-17972 PL-17294 PL-17942 PL-17836 PL-17488 PL-17747 PL-17943 PL-17967 PL-18031 PL-17550 PL-17279 PL-17815 PL-17950 PL-15815 PL-17286 PL-17896 PL-17887 PL-8823 PL-16693

---

v2025.1.71.2873 - Hive connector support

CAUTION

Для корректной работы платформы после обновления вам необходимо провести **миграцию**. До запуска новой версии сделайте следующие действия:

1. Остановите необходимые сервисы:

```
./akd stop {proctrigger,procworkflow,proctask}
```

2. Запустите `aktool` **из новой сборки** с помощью команды:

```
./aktool migrate policystore --config /opt/MarketingPlatform/config/main.json
```

3. Перезапустите сервисы платформы:

```
./akd restart
```

Справка по мигратору:

```
Migrate policystore db to accounts dbs collections policy_contact, policy_profile

Usage:
  aktool migrate policystore [flags]
  aktool migrate policystore [command]

Available Commands:
  ensure_policystore_indexes Ensures policystore migrate indexes created

Flags:
      --account int          account id
      --batch_interval int   batch interval in seconds (default 1)
      --batch_size int       batch size (default 10000)
      --batch_workers int    number of batch workers (default 8)
  -h, --help                 help for policystore
      --workers int          number of workers for accounts (default 8)

Global Flags:
      --clickhouse_database_name string   usage database for clickhouse
      --config string                     config path (default
"/opt/akd/config/main.json")
  -o, --output string                     file path to output in the result of the
command (default "stdout")
      --truncate                          whether to truncate the file specified by --
output

Use "aktool migrate policystore [command] --help" for more information about a command.
```

Основные изменения

SQL-коннектор Hive

Добавлена поддержка **SQL-коннектора Hive/Hadoop** в качестве внешней базы данных. Подробнее о настройке коннектора можно прочесть здесь.

Другие изменения

- Из правил блокировки на сендере MTA удалено правило `"not\\savailable"`, поскольку оно приводило к ошибкам в работе email-кампаний.
- Обновлены API для новых версий Google Ads и Facebook*™ Ads Manager.
- Добавлена поддержка Rabbit версии 4+.
- Платформа Altcraft прошла сертификацию на совместимость и корректность работы с отечественной операционной системой “РЕД ОС 8”.
- Параметр `MTA_DEFAULT_TTL`, отвечающий за время ожидания доставки SMS и Email, теперь по умолчанию равен 3 суткам (259200 секунд).

Исправления

- Оптимизированы механизмы защиты платформы.
- Исправлены некоторые UI/UX ошибки в интерфейсе.
- Улучшена стабильность процессов `akd`, `akmta`, `aktool`, `procpiper`, `proctrigger`, `procintegras`, `proctask` и `procworkflow`.
- Оптимизирован разбор очереди `oxy_triggers`.
- Пользовательские bounce-правила в AKMTA теперь работают корректно.
- Оптимизирован механизм ротации логов.
- Веб-версии email-писем, отправленных через платформу, теперь сохраняютя в корректном формате.
- Улучшена стабильность работы 2FA.
- Исправлены некоторые ошибки, связанные с редактированием шаблонов сообщений в платформе.
- JSON-фильтр теперь корректно работает во вложенных объектах.
- Исправлена ошибка, при которой данные из формы ошибочно передавались дважды.
- Добавлена валидация для некоторых полей в Панели администратора.
- Исправлены некоторые ошибки, связанные с ролями доступа.
- Дополнительные поля профиля снова выгружаются при экспорте ПД.
- Устранена ошибка, при которой `apicontent` профилей при массовом импорте в сценарий работал некорректно.

References

PL-17550 PL-18489 PL-18393 PL-18384 PL-18106 PL-18136 PL-15177 PL-18457 PL-18247 PL-18386 PL-18382 PL-18215 PL-18104 PL-15824 PL-18424 PL-18401 PL-18312 PL-18238 PL-18244 PL-18139 PL-18415 PL-18238 PL-18350 PL-18402 PL-18145 PL-18177 PL-18331 PL-18339 PL-18287 PL-15410 PL-18294 PL-18289 PL-18135 PL-18235 PL-18314 PL-18290 PL-18198 PL-18145 PL-18175 PL-18013 PL-18281 PL-18154 PL-18229 PL-18150 PL-17878 PL-17807 PL-18283 PL-18268 PL-17604 PL-18038 PL-18169 PL-18073 PL-18168 PL-18263 PL-18265 PL-18255 PL-17816 PL-18053 PL-18161 PL-17546 PL-18196

---

v2025.1.71.3092 - Event backward compatibility

Изменения

- Добавлена обратная совместимость событий из пользовательских каналов устаревших версий платформы.
- Оптимизированы механизмы защиты.

Исправления

- Тестовые отправки в Telegram теперь работают корректно.
- Значение `0` в условии “в таблице данных” теперь отображается при передаче через параметры запроса.
- Устранена ошибка в редакторе форм, при которой учитывались некорректные правила доступа.
- Добавлен тайм-аут для невыполненных запросов в PostgreSQL.
- Пользовательский путь файла конфигурации для сервисов платформы снова указывается корректно.
- Улучшена читаемость API-логов.
- При создании объектов через API теперь корректно обрабатываются теги и дополнительные поля.
- Визуальный редактор снова корректно обновляет настройки пользовательского блока.
- При формировании контрольной группы из файла теперь правильно работают поля типа “Целое число”.
- Снижено потребление памяти процессом `campaign` при отправке сообщений в Telegram с вложениями.
- В шаблонах теперь корректно работает проверка наличия ссылки подтверждения.

- Исправлена работа опции “Повышать приоритет для новых подписок” в настройках ресурса.
- AKMTA теперь корректно переключается на резервные MX при недоступности основного.
- В шаблонах теперь корректно подставляется дата из профиля при использовании функции `formatdate`.
- Исправлены ошибки, связанные с обработкой событий сендера Amazon SNS.

References

PL-10119 PL-18521 PL-18315 PL-17401 PL-18125 PL-18000 PL-17857 PL-17818 PL-18266 PL-18426 PL-18544 PL-17224 PL-18478 PL-18269 PL-18193 PL-18384 PL-18323 PL-17622 PL-18527 PL-17548 PL-18041 PL-18425 PL-18368 PL-18237 PL-18151

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*
***Viber является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

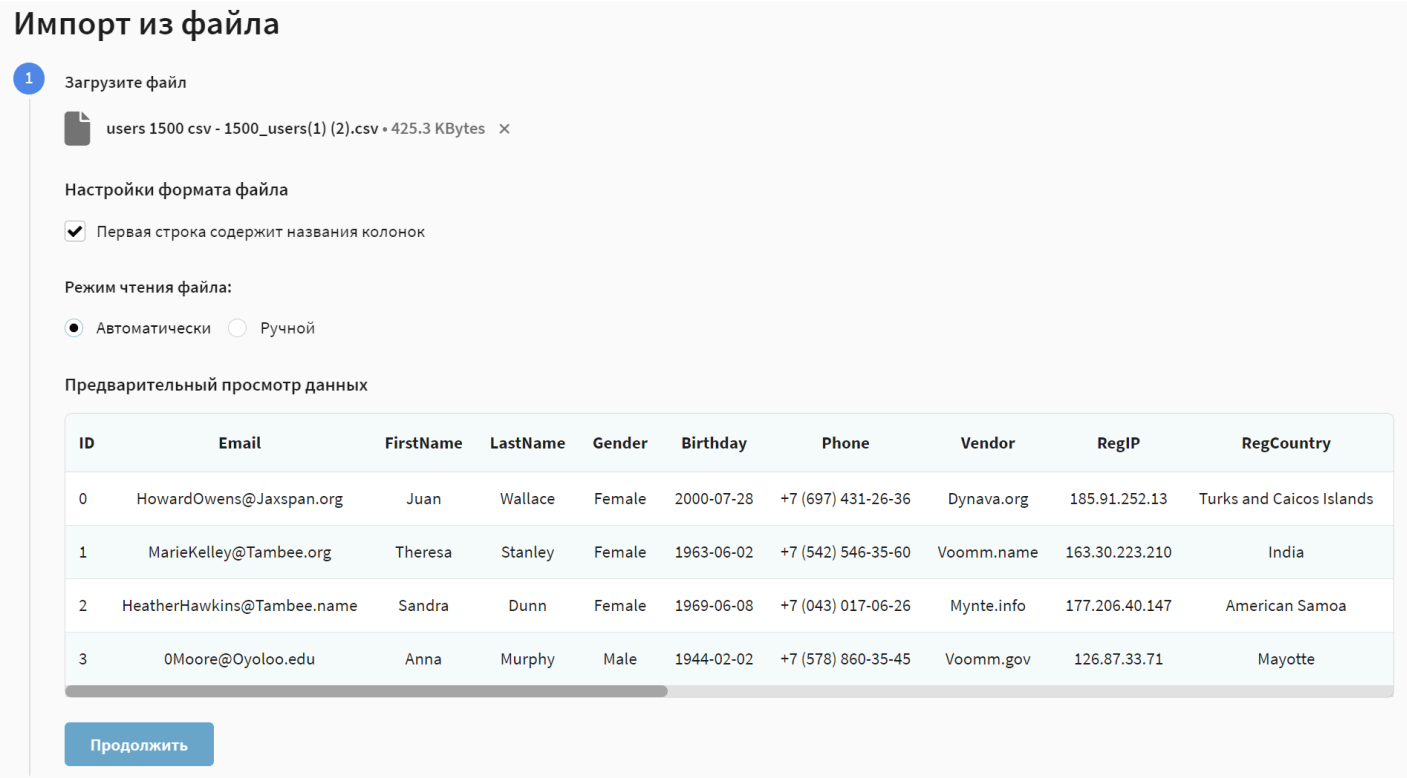
v2024.4.70

v2024.4.70.0 - New import menu, Rapporto integration, License info

Основные изменения

Улучшение импорта профилей клиентов

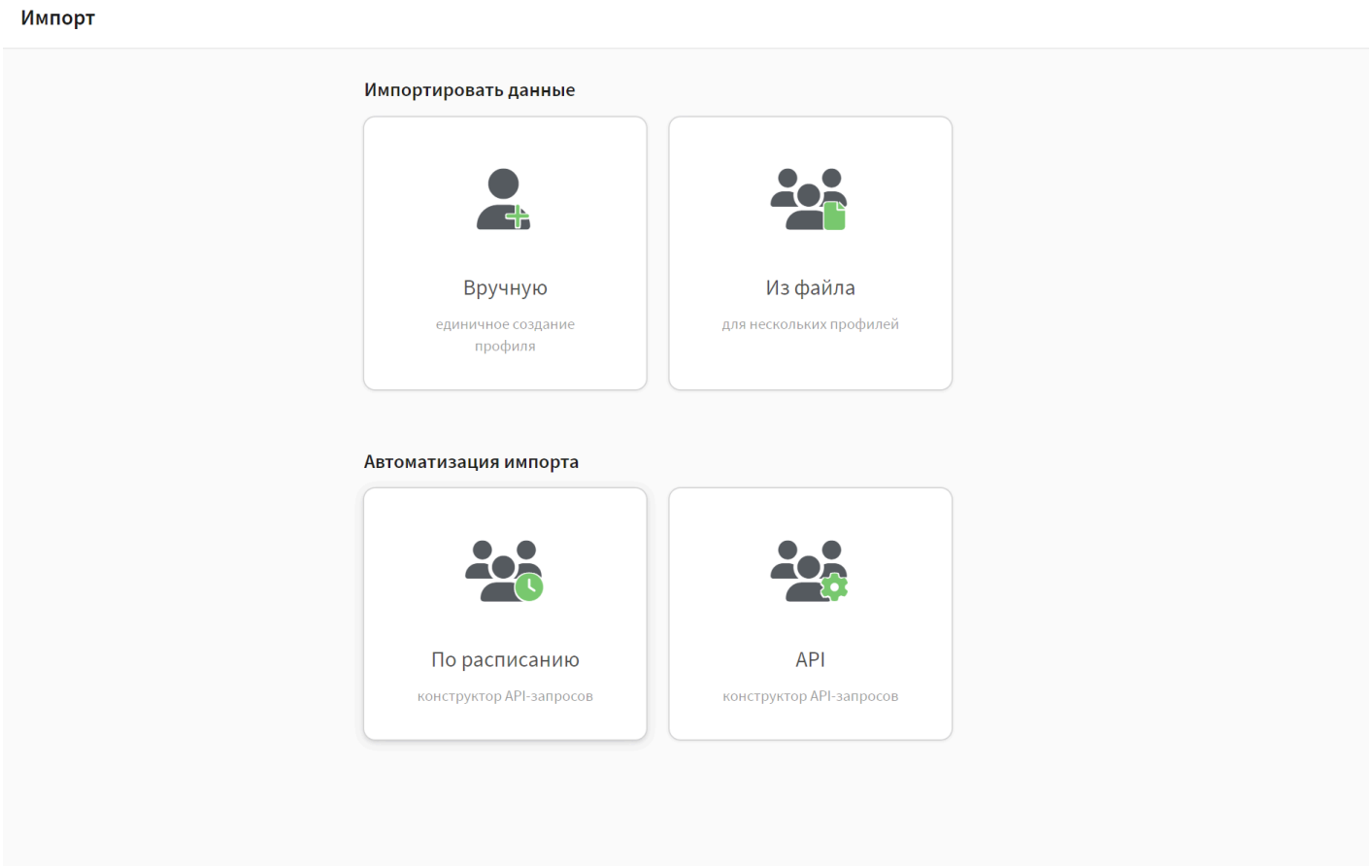
Полностью переработан импорт профилей из файла. Теперь при выборе этого пункта перед вами появится пошаговое руководство с подробным описанием всех этапов и настроек импорта:



ID	Email	FirstName	LastName	Gender	Birthday	Phone	Vendor	RegIP	RegCountry
0	HowardOwens@Jaxspan.org	Juan	Wallace	Female	2000-07-28	+7 (697) 431-26-36	Dynava.org	185.91.252.13	Turks and Caicos Islands
1	MarieKelley@Tambee.org	Theresa	Stanley	Female	1963-06-02	+7 (542) 546-35-60	Voomm.name	163.30.223.210	India
2	HeatherHawkins@Tambee.name	Sandra	Dunn	Female	1969-06-08	+7 (043) 017-06-26	Mynte.info	177.206.40.147	American Samoa
3	0Moore@Oyoloo.edu	Anna	Murphy	Male	1944-02-02	+7 (578) 860-35-45	Voomm.gov	126.87.33.71	Mayotte

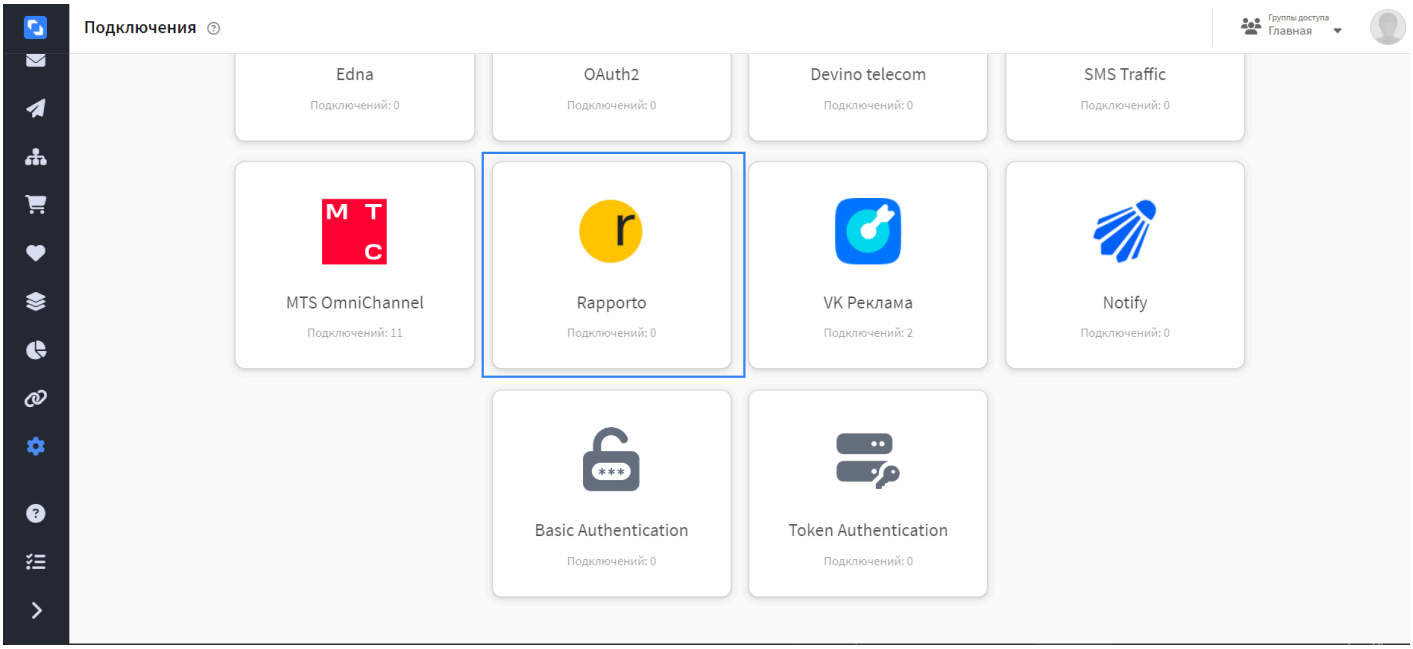
Появилась возможность выбирать кодировку файла, разделитель, выбирать формат даты и редактировать формат телефонных номеров прямо во время импорта.

Также обновлено меню импорта в базу профилей:



Интеграция с Rapporto

Добавлена поддержка интеграции с **Rapporto**. Теперь вы можете использовать эту платформу как шлюз для отправки сообщений в Viber™:



Прежде чем настраивать интеграцию, вам необходимо связать ваш аккаунт в Rapporto с Altcraft Platform через подключение. Подробнее о настройке интеграции вы можете прочитать здесь.

Информация о лицензии

В Панели администратора добавлен пункт **License** в разделе Tools. В этом пункте можно посмотреть подробную информацию о версии платформы, лицензии и ее сроке действия.

В поле **Notify about license changes to** вы можете указать адрес электронной почты, на который будут приходить уведомления о смене статуса лицензии или об истечении срока ее действия. Сообщения начнут отправляться за 2 недели до окончания срока действия лицензии, а также в случае изменения ее статуса.

Другие изменения

- В элемент сценария “JSON-Фильтр” добавлены новые операторы: `contains`, `in`, `notin` и `!~=`. Подробнее об их использовании читайте здесь.
- В элемент сценария “API-запрос” появилась возможность добавлять пользовательские заголовки.
- Теперь вы можете отправлять письма без текстовой версии сообщения.
- Обновлен справочник полов `genders`.

> **INFO**
>
> Если вам необходимо оставить старую конфигурацию справочника `genders` или внести в него какие-то изменения, создайте новый файл, назовите его “custom.genders.json” и внесите необходимые правки туда. Подробнее об этом можно прочитать здесь.

- Оптимизирована работа элемента сценария “Сегмент”.
- Переработан механизм прерывания циклов в сценариях, теперь он корректно срабатывает в сложных сценарных цепочках.

Исправления

- Исправлена ошибка, при которой экспортированный шаблон содержал не все исходные данные.
- Устранены ошибки, возникавшие при ресинхронизации с сервисом VK Реклама.
- Условия ожидания определенного дня недели теперь корректно работают в сценариях.
- Исправлена ошибка, связанная с выбором канала в триггерных кампаниях.
- Экспорт профилей с подпиской снова проходит корректно.
- Исправлена ошибка, при которой сегментация с отрицательными условиями по Push-доставкам некорректно отбирала профили.
- Условие сегментации “Находится в списке” теперь корректно отбирает значения, содержащие слэш.
- Элементы в сценариях теперь корректно графически соединяются между собой.
- Событие доставки для web push в Safari снова корректно регистрируется только после фактической отправки.
- Устранены ложные срабатывания триггера “Таймер” в Менеджере тегов.
- Устранены ошибки, возникавшие при настройке подключения к Facebook Ads.
- Улучшена работа аналитики по тестовым отправкам.
- Фильтр по кампаниям в отчете по каналам теперь корректно обнаруживает кампании, созданные внутри сценариев.
- Исправлена ошибка, при которой перемещение шаблонов и кампаний в корзину работало некорректно.
- Улучшена стабильность процессов `akmtad`, `procworkflow`, `proctrigger`.

References

PL-16980 PL-17004 PL-16990 PL-16934 PL-15925 PL-16939 PL-16170 PL-16565 PL-16649 PL-16306 PL-16100 PL-16450 PL-16458 PL-16457 PL-16459 PL-16180 PL-16357 PL-15777 PL-16326 PL-16323 PL-16312 PL-16317

PL-16226 PL-16023 PL-15844 PL-13617 PL-15843 PL-13634 PL-12949 PL-16857 PL-17055 PL-17050 PL-17041 PL-17036 PL-17022 PL-17001 PL-16978 PL-16983 PL-16890 PL-16891 PL-16866 PL-16945 PL-16875 PL-16856 PL-16790 PL-16545 PL-16707 PL-16299 PL-16310 PL-16244 PL-16045 PL-15518 PL-17000 PL-16754

---

v2024.4.70.963v2024.4.70.963 - Action Hook SMID, push suspend method

Основные изменения

Изменения в идентификаторах сообщения захвата событий

Изменена структура сообщения захвата событий. Теперь такие сообщения содержат поля:

- `send_message_id` — уникальный идентификатор (USMID);
- `full_message_id` — FSMID для события (при наличии);
- `email_message_id` — ESMID сообщения для email-канала.

Введение новых идентификаторов позволяет однозначно находить необходимые события и тем самым избегать сложностей при анализе ошибок. `send_message_id` по-прежнему является уникальным ID сообщения, но теперь выглядит значительно короче.

Новый метод приостановки push-подписки

Реализован новый метод для приостановки push-подписки пользователей на ресурс через web push SDK. Подробнее о SDK и его настройке можно узнать здесь.

Другие изменения

- Добавлена комбинация “Текст + Изображение” для шаблонов Viber™.
- Добавлена поддержка выбора полей аутентификации для SMS-шлюза MFMS EDNA.
- Теперь коллекции веб-версий с истекшим сроком жизни удаляются вместе с их содержимым.
- Для элемента сценариев “Вызов API” добавлен редактор пользовательских заголовков.

Исправления

- Теперь счетчик импортированных профилей в Панели администратора автоматически пересчитывается при удалении базы профилей.
- Прикрепление файла в разделе “Сегмент” теперь работает корректно.
- Исправлена ошибка, при которой платформа ошибочно регистрировала больше событий отправки и открытий для некоторых каналов и шлюзов.
- Картинки теперь корректно загружаются при импорте HTML-шаблона.
- Улучшена стабильность работы раздела “Аналитика”, исправлены некоторые ошибки сервера, связанные с аналитикой.
- Выбор победителя по кликам в мультивариантных тестах теперь работает корректно.
- Исправлена ошибка, вызывающая дубликацию подписок для профиля при нажатии на кнопку сохранения.
- Статусы по отправкам через WhatsApp*-канал теперь приходят корректно.
- Улучшена валидация при заполнении полей в некоторых разделах платформы.
- Импорт через файл снова корректно работает для клиентов облачной версии платформы.
- Устранена ошибка, возникающая при подключении коннектора PostgreSQL к базе данных на Яндекс Облаке.

- Импорт при помощи zip-архива снова работает корректно.
- Исправлена ошибка, связанная с достижением максимального лимита активных сессий в MongoDB.
- Теперь профили, попавшие в сценарий, разбираются обработчиками равномерно, не создавая задержек.
- Исправлена ошибка, при которой часть сообщения обрезалась при предпросмотре SMS-шаблона.
- Условие “Участие в сегменте” теперь не отображает последний созданный сегмент, если используемый сегмент был удален.
- Улучшена работа полноэкранного режима редактора шаблонов на Safari.
- События писем в интеграции с Sendsay теперь регистрируются корректно.
- Выбор дробного числа снова доступен в сегменте по числовым данным цели.
- Убраны PostgreSQL-запросы к Маркету в тех сценариях, где они не должны инициироваться.
- Шаблон с пользовательской структурой теперь корректно импортируется в платформу.
- Устранена ошибка, при которой платформа не запускалась при недоступности одного узла в кластере ClickHouse.
- Условие ожидания дня недели снова корректно работает в сценариях.
- Push-уведомления со специальными последовательностями Unicode-символов теперь корректно отправляются на провайдере iOs Firebase.
- Исправлен алгоритм автовыравнивания в сценариях.
- При отправке пустого SMS-шаблона в журнале кампании теперь появляется запись об ошибке.
- Тип события в Notify теперь корректно отображается при экспорте истории профиля.
- Исправлена ошибка, которая делала возомжным создание продукта без заполнения обязательных полей.
- Устранены ошибки, связанные с сохранением или удалением подписки на ресурс пользовательского канала.
- Отписка при отключении уведомлений в браузере Safari теперь работает корректно.
- Теперь для импорта из файла в Маркете указаны поддерживаемые форматы файлов.
- Устранена ошибка, при которой при копировании сегмента в его структуре сохранялись некорректные для каналов условия.
- Из журнала кампании убрана нерелевантная ошибка “Config server db posgres is missing for account”.
- Исправлены ошибки, связанные с сохранением Telegram-шаблонов.
- Улучшена стабильность процессов `procworkflow`, `webcontrol`.
- Исправлены некоторые ошибки в интерфейсе платформы.

References

PL-17403 PL-17338 PL-17301 PL-16970 PL-17313 PL-17190 PL-17167 PL-17116 PL-13523 PL-17152 PL-12931 PL-17113 PL-16889 PL-13032 PL-17069 PL-16689 PL-16988 PL-16645 PL-16785 PL-16811 PL-16729 PL-17290 PL-16126 PL-17333 PL-16090 PL-17325 PL-17314 PL-17289 PL-17261 PL-17129 PL-17202 PL-17238 PL-17226 PL-16615 PL-17217 PL-16840 PL-17148 PL-17201 PL-17207 PL-17199 PL-15413 PL-10841 PL-17171 PL-15854 PL-17151 PL-16967 PL-17117 PL-17124 PL-17130 PL-17142 PL-15860 PL-17093 PL-16528 PL-14496 PL-16082 PL-17036 PL-16803 PL-16802 PL-17034 PL-15434 PL-16789 PL-16440 PL-16821 PL-16819 PL-17196 PL-17081 PL-17129 PL-17058 PL-17112 PL-16830 PL-13026 PL-16823 PL-17185 PL-17086 PL-14429 PL-16866 PL-16887 PL-17073 PL-15837 PL-17007 PL-16816 PL-17044 PL-16825 PL-17035 PL-17019 PL-16859 PL-16941 PL-16992 PL-17012 PL-17011 PL-16202 PL-15451 PL-15679 PL-16627 PL-16966 PL-16732 PL-16867 PL-16844 PL-16753 PL-2915 PL-5732 PL-11319 PL-16381 PL-16313 PL-16237 PL-17053 PL-16847 PL-17020 PL-16521 PL-16527 PL-17063

---

v2024.4.70.1088 - Web services handlers, stability improvements

Основные изменения

API-запросы к хендлерам статусов веб-сервисов

Корректная балансировка веб-сервисов требует от пользователя постоянную и точную передачу их состояния. Для выполнения этого условия реализованы специальные **API-запросы к хендлерам статуса процессов**. Были добавлены запросы к хендлерам следующих процессов: `webcontrol`, `webadmin`, `webapi`, `trkaction`, `trkwebctrl`, `cookiesaver`.

Узнать больше об этих API-запросах можно по ссылке.

Другие изменения

- Добавлена валидация при загрузке видео-файла в шаблоне Telegram.
- История профилей теперь обновляется быстрее, обеспечивая корректную работу сценариев со сложной логикой.

Исправления

- Временные файлы платформы теперь корректно очищаются по истечению срока хранения.
- Исправлено некорректное отображение показателей отчета по приросту.
- Устранены проблемы с отображением некоторых элементов в интерфейсе.
- Исправлены ошибки, из-за которых журнал действий в Панели администрирования фиксировал не все события.
- Улучшена стабильность процессов `procworkflow` и `proctask`.

References

PL-17467 PL-17445 PL-17127 PL-17267 PL-17182 PL-17090 PL-16638 PL-16931 PL-17183 PL-17016 PL-16286 PL-10929 PL-17014 PL-16911

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*
***Viber является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

v2024.3.69

v2024.3.69.0 - Altcraft mSDK, Notify

Основные изменения

SDK мобильных пушей

Altcraft mSDK — это мобильный программный интерфейс (SDK) для интеграции push-уведомлений в мобильные приложения. Он помогает разработчикам легко подключить систему уведомлений Altcraft к своим Android и iOS приложениям через платформы Firebase и APNs.

Altcraft mSDK позволяет:

- Автоматически подписывать пользователей на push-уведомления.
- Поддерживать отправку уведомлений с текстом, изображениями и кнопками с действиями.
- Актуализировать и обновлять токены устройства на сервере для бесперебойной доставки уведомлений.
- Отслеживать доставку уведомлений и взаимодействие пользователей с ними.
- Автоматически повторять неудавшиеся запросы и работать в фоновом режиме.

> **TIP**
>
> На данный момент Altcraft mSDK предоставляется **по запросу** в виде архива.

Миграция шаблонов мобильных пушей

Обновлена структура шаблонов мобильных пушей в следующих файлах:

- `json_consts/templates/AndroidFirebase.json`
- `json_consts/templates/IOSAPNS.json`
- `json_consts/templates/IOSFirebase.json`

Если вы отправляете мобильные пуши через данных провайдеров, но при этом не переопределяли эти файлы ранее файлами с префиксами `custom.`, то вам **необходимо перенести содержимое исходных файлов перед обновлением.**

Для этого скопируйте содержимое каждого исходного файла в соответствующие файлы с префиксами `custom.`:

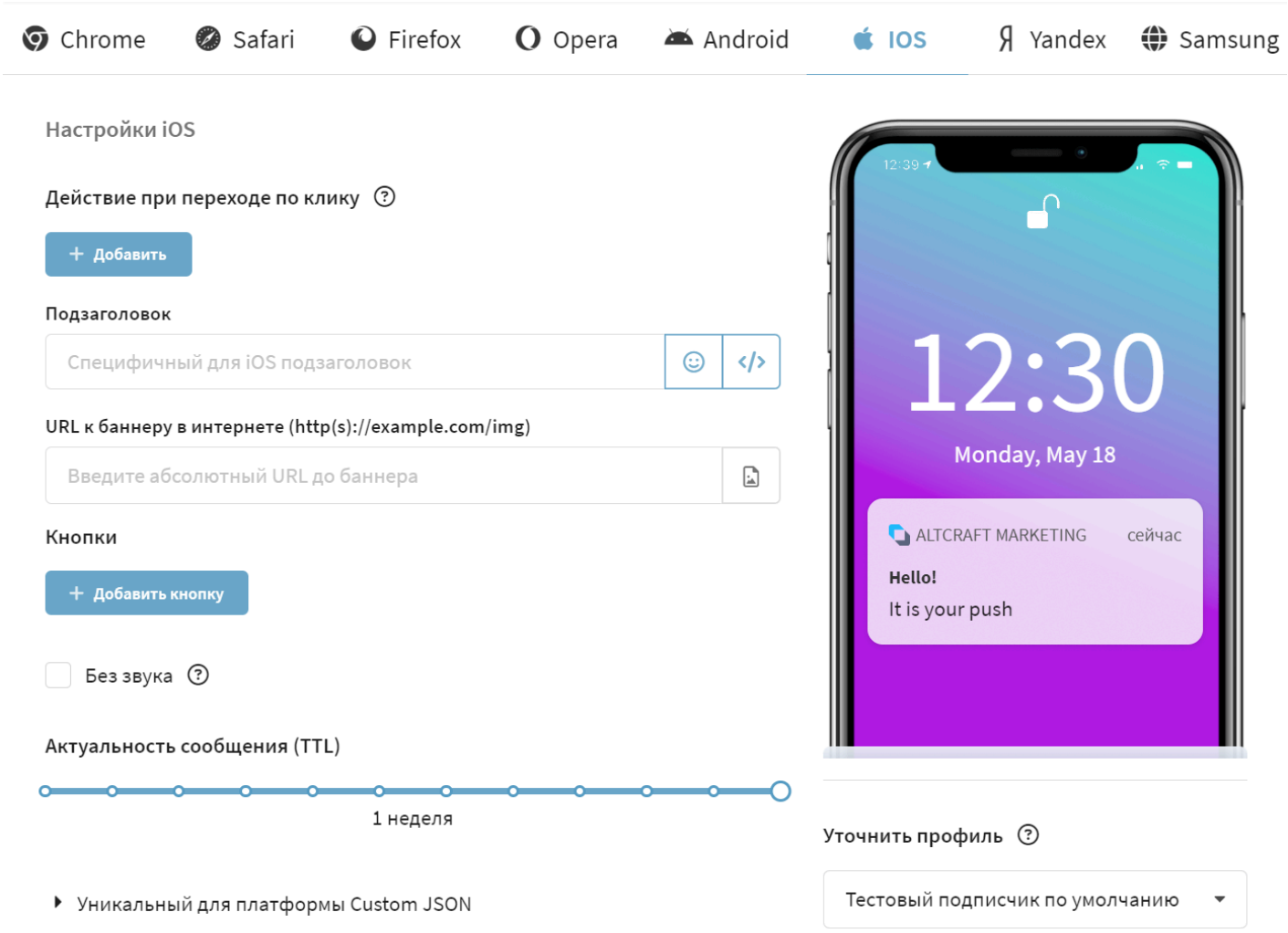
- `json_consts/templates/custom.AndroidFirebase.json`
- `json_consts/templates/custom.IOSAPNS.json`
- `json_consts/templates/custom.IOSFirebase.json`

Либо перенесите их в файлы, привязанные к аккаунту:

- `json_consts/templates/custom.<номер аккаунта>.AndroidFirebase.json`
- `json_consts/templates/custom.<номер аккаунта>.IOSAPNS.json`
- `json_consts/templates/custom.<номер аккаунта>.IOSFirebase.json`

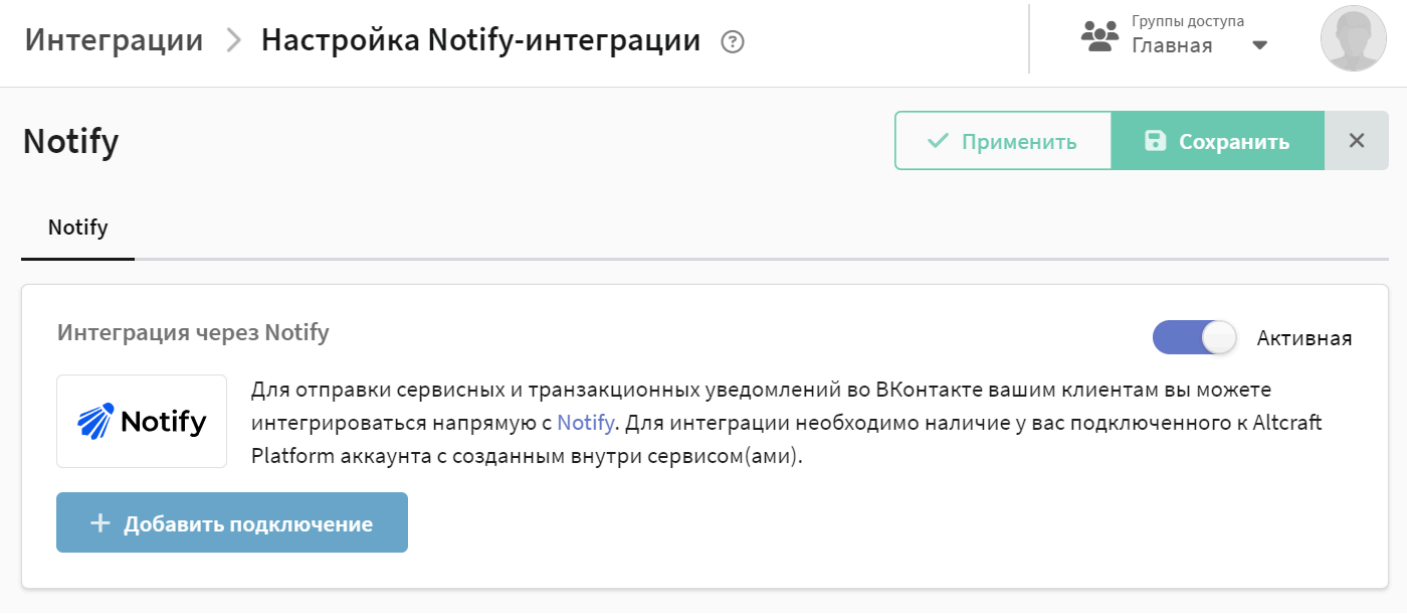
Обновление редактора мобильный пушей

Обновлён редактор мобильных пушей (**редактор β**). Удалены избыточные поля настройки, добавлены новые кнопки с возможностью выбора действия, реализована загрузка баннера из галереи платформы, добавлена возможность отключения звука. Для iOS также удалено поле “Критическое PUSH-уведомление”.



Интеграция Notify

В платформу добавлена интеграция с **Notify** для отправки транзакционных и сервисных сообщений в VK, Одноклассники и SMS. Реализована возможность настройки канала в ресурсе и создания пользовательских **шаблонов**.



Другие изменения

- Добавлена корректная поддержка XML в API 1.1 версии платформы. **Подробнее**.
- Доработаны сценарии. Добавлена возможность масштабирования на тачпаде. Изменена система установки связей между элементами сценария.
- Улучшены превью шаблонов для канала Viber™.
- Добавлены новые методы в API администратора для создания и удаления аккаунта.

Исправления

- Исправлена проблема зависания кампании на этапе рекконекта к RMQ, вызванная высоким потреблением памяти.
- Экспорт истории профилей теперь выполняется быстрее.
- Изменение API-токена в настройках web push канала для Safari успешно сохраняется.
- В аналитике по недоставкам корректно отображается статистика по браузерным пушам.
- В поле профиля с типом массив объектов теперь можно сохранить пустой массив вручную, указав значение `[]`, или через API, передав в поле `"[]"` или `""`
- Добавлено предупреждение при выборе просроченной даты запуска в расписании кампании.
- Добавлено предупреждение при сохранении сценария, если в узле “Сегмент по расписанию” не указаны условия запуска или выбран пустой сегмент, чтобы избежать случайного запуска на всю базу данных.
- В галереи теперь отображается протокол http(s) в URL изображений.
- Обновлён TypeScript для улучшения поддержки и совместимости.
- Устранены JS-исключения в узле обновления профиля и при работе с формами.
- Улучшена стабильность процесса `proctask` и `webcontrol`.
- Исправлены незначительные недочёты интерфейса платформы.

References

PL-16878, PL-16717, PL-16767, PL-16799, PL-16860, PL-16798, PL-16843, PL-16020, PL-16831, PL-16814, PL-16764, PL-16806, PL-16813, PL-16800, PL-16822, PL-16796, PL-16758, PL-16832, PL-16815, PL-16768, PL-16769, PL-16757, PL-16766, PL-16746, PL-16735, PL-16760, PL-16751, PL-16235, PL-16251, PL-16726, PL-15887, PL-15878, PL-15677, PL-16568, PL-16658, PL-16644, PL-15124, PL-16143, PL-12781, PL-16292, PL-16223, PL-6765, PL-15227, PL-16554, PL-16391, PL-16544, PL-16431, PL-16419, PL-14376, PL-16062, PL-16229, PL-16070, PL-

15875, PL-15840, PL-15830, PL-14796, PL-8846, PL-15522, PL-13267, PL-15429, PL-15372, PL-15367, PL-15377, PL-15374, PL-14911

---

v2024.3.69.490 - Batch import API improvment

Основные изменения

Улучшение массового импорта профилей через API

Оптимизирован API-метод для массового импорта профилей в базу данных (`/v1.1/profiles/import_batch`). Теперь его выполнение занимает значительно меньше времени, чем раньше.

Другие изменения

- Обновлена интеграция с Google Ads в связи с выходом новой версии Google Ads API.
- Добавлена валидация для поля “Лимит” в разделе Связи.
- Добавлена проверка на расширение загружаемого файла в шаблонах Telegram.

Улучшения интерфейса

- Исправлено отображение кавычек в тегах и группах.
- Скорректированы тексты многих ошибок в логах платформы.
- Исправлено отображение элементов на мобильной версии.
- Улучшено отображение выпадающих списков.
- Добавлена подсказка в интерфейсе для параметра “Копировать профили из базы данных”.
- Улучшена верстка в условиях активации триггеров.
- Устранена ошибка, при которой кнопки “Применить” и “Сохранить” некорректно отображались в разных разделах платформы.

Исправления

- Исправлена ошибка, при которой некоторые web push-уведомления не приходили пользователям.
- Вызовы метода Close драйвера MongoDB теперь работают корректно.
- Устранена ошибка, возникающая при переходе между страницами в истории действий профиля.
- В запросах импорта профилей теперь отображается корректный код ошибки.
- Исправлена ошибка, при которой невозможно было использовать циклический перебор для сущностей и полей Маркета.
- Режим авторизации SASL в настройке Actions Hooks теперь работает корректно.
- Устранена ошибка, при которой было невозможно сохранить ресурс для мобильных push без поля `appId` .
- Выбор режима отправки для push-уведомлений теперь правильно отображается для всех кампаний.
- Исправлена ошибка, при которой некоторые процессы не работали при запуске платформы с включенным SOCKETS.
- Сценарии, у которых в кампании указан ресурс с Firebase API Legacy, снова сохраняются корректно.
- Устранены ошибки, возникающие при сохранении воронки с фильтрацией по тегам.
- Очереди на сендере теперь удаляются корректно.
- Исправлена ошибка, при которой во время сохранения шаблона пропадал part в html-версии.
- Теперь профили, учавствующие в статическом сегменте, корректно попадают в быстрый сегмент.
- Устранена ошибка, при которой не срабатывал триггер по импорту профиля через push.

- Исправлена ошибка, при которой элемент сценария “JSON-фильтр” работал некорректно.
- Улучшена стабильность работы элемента “API-запрос” в сценариях.
- Коннектор PostgreSQL теперь корректно подключается к базе данных в Yandex Cloud.
- Исправлена ошибка, при которой оператор условия “Email подписка: домен (не) в списке” в сценариях искал профили по полному адресу, а не по домену.
- Быстрый сегмент теперь корректно ищет профили по заданным условиям.
- Исправлена ошибка, при которой в заказе создавалась новая позиция вместо изменения данных старой.
- Очистка временных файлов в `data/segment` теперь работает корректно.
- Исправлены ошибки, связанные с WhatsApp-интеграцией.
- Устранена перегрузка памяти в тех случаях, когда в шаблон добавлялся запрос к базе данных.
- Тело шаблона теперь не обрезается при отправке push-уведомлений через IOS ANPs.
- Исправлены ошибки, возникающие при отправке push-уведомлений со старой структурой.
- Параметр, позволяющий послать push-уведомление на несколько одинаковых подписок, теперь работает корректно.
- Профили с неверно указанным номером телефона теперь корректно отображаются при экспорте.
- Улучшена стабильность процесса `akmtad`.

References

PL-16976 PL-16955 PL-16940 PL-16928 PL-16752 PL-16886 PL-16877 PL-15784 PL-15086 PL-16194 PL-16624 PL-16605 PL-16936 PL-16914 PL-16929 PL-15817 PL-16922 PL-16897 PL-16905 PL-16910 PL-16196 PL-16881 PL-16840 PL-16876 PL-16741 PL-15717 PL-16177 PL-16761 PL-14418 PL-8897 PL-16275 PL-6070 PL-15583 PL-15438 PL-16925 PL-16916 PL-16919 PL-16909 PL-16207 PL-16747 PL-16784 PL-16809 PL-16865 PL-15765 PL-16810 PL-16181 PL-16728 PL-16839 PL-16750 PL-16711 PL-16475 PL-14771 PL-11860 PL-13762 PL-4626 PL-16099

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*
***Viber является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

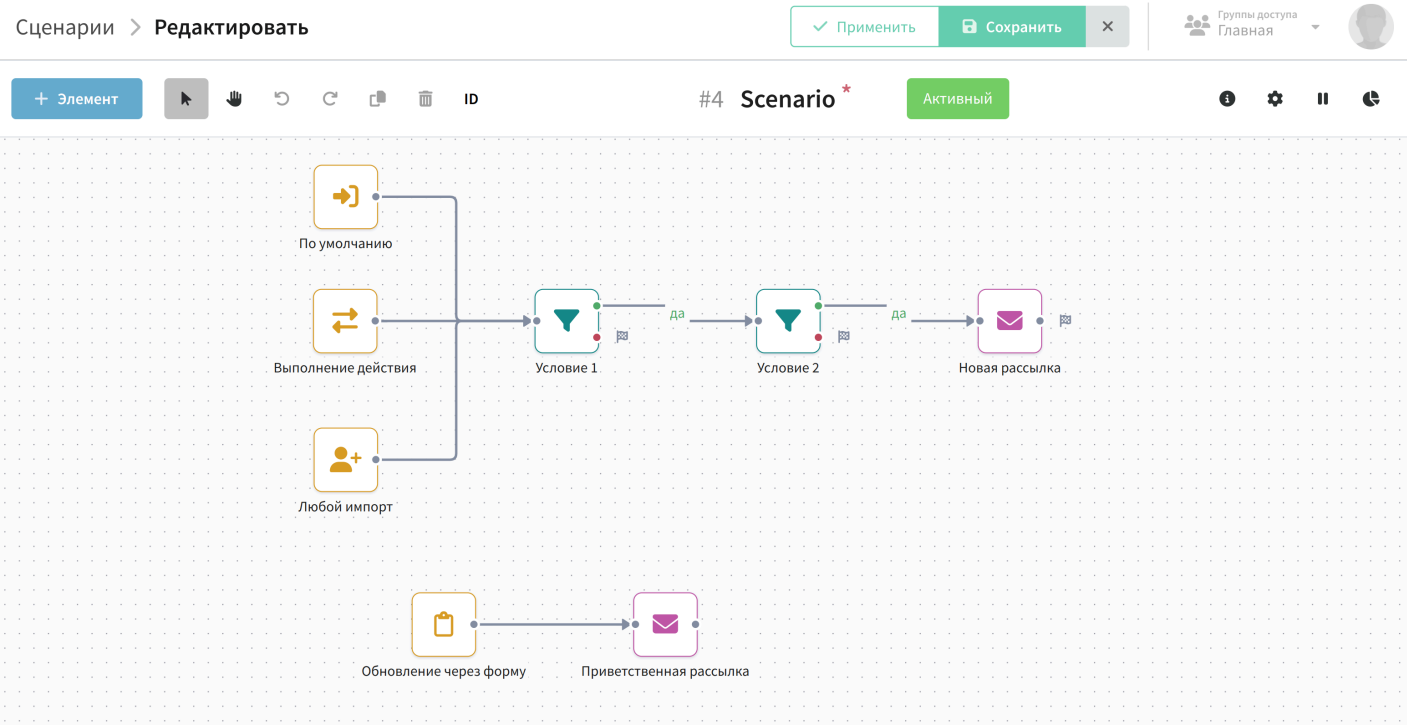
v2024.2.68.2

v2024.2.68.2.287 - Multi-capture scenario, Product feed, Pixel trigger improvements

Основные изменения

Комбинированный захват профилей в сценарий

Добавлена возможность настроить **захват профилей несколькими условиями** в рамках одного сценария. Такой захват позволяет пользователю учесть разные варианты входа, не создавая при этом множество новых сценариев. Для одного сценария можно добавить до 10 таких условий.



Также в сценарии доступна **статистика по захваченным профилям**, в том числе для каждого условия отдельно. Данная функция позволяет анализировать данные по захвату из различных бизнес-точек. Подробнее о настройке элементов сценария можно узнать [здесь](#).

Импорт фида продуктов по ссылке в Маркете

CAUTION

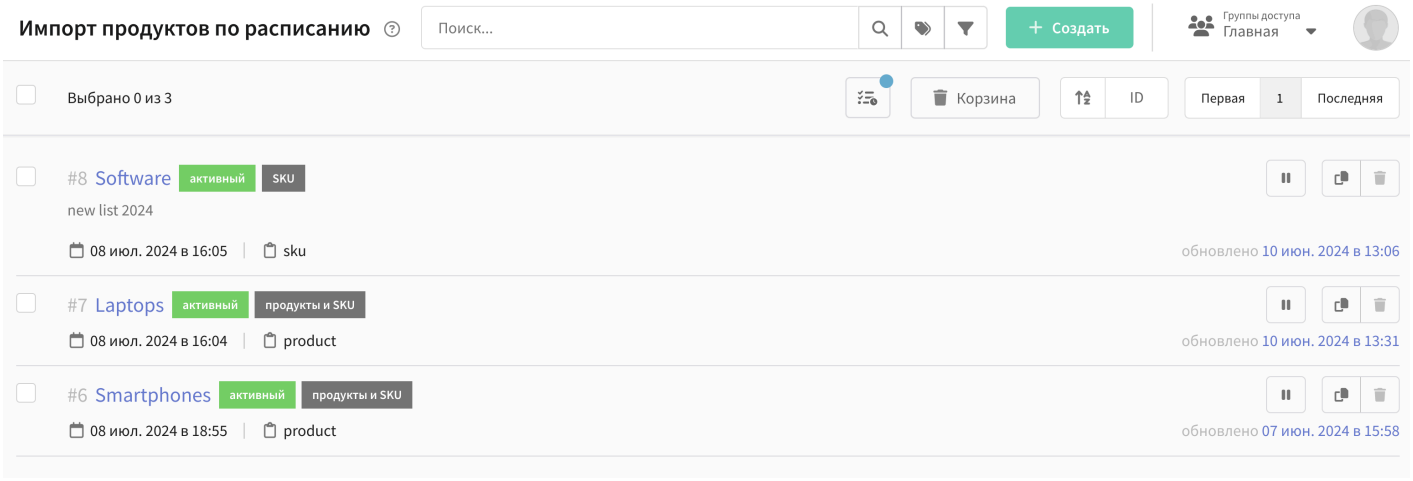
После добавления товарного фида для корректной работы модуля Маркет необходимо произвести **миграцию**. Для этого до запуска новой версии сделайте следующие действия:

1. Остановите сервис `webcontrol`
2. Запустите `aktool` **из новой сборки** с помощью команды:

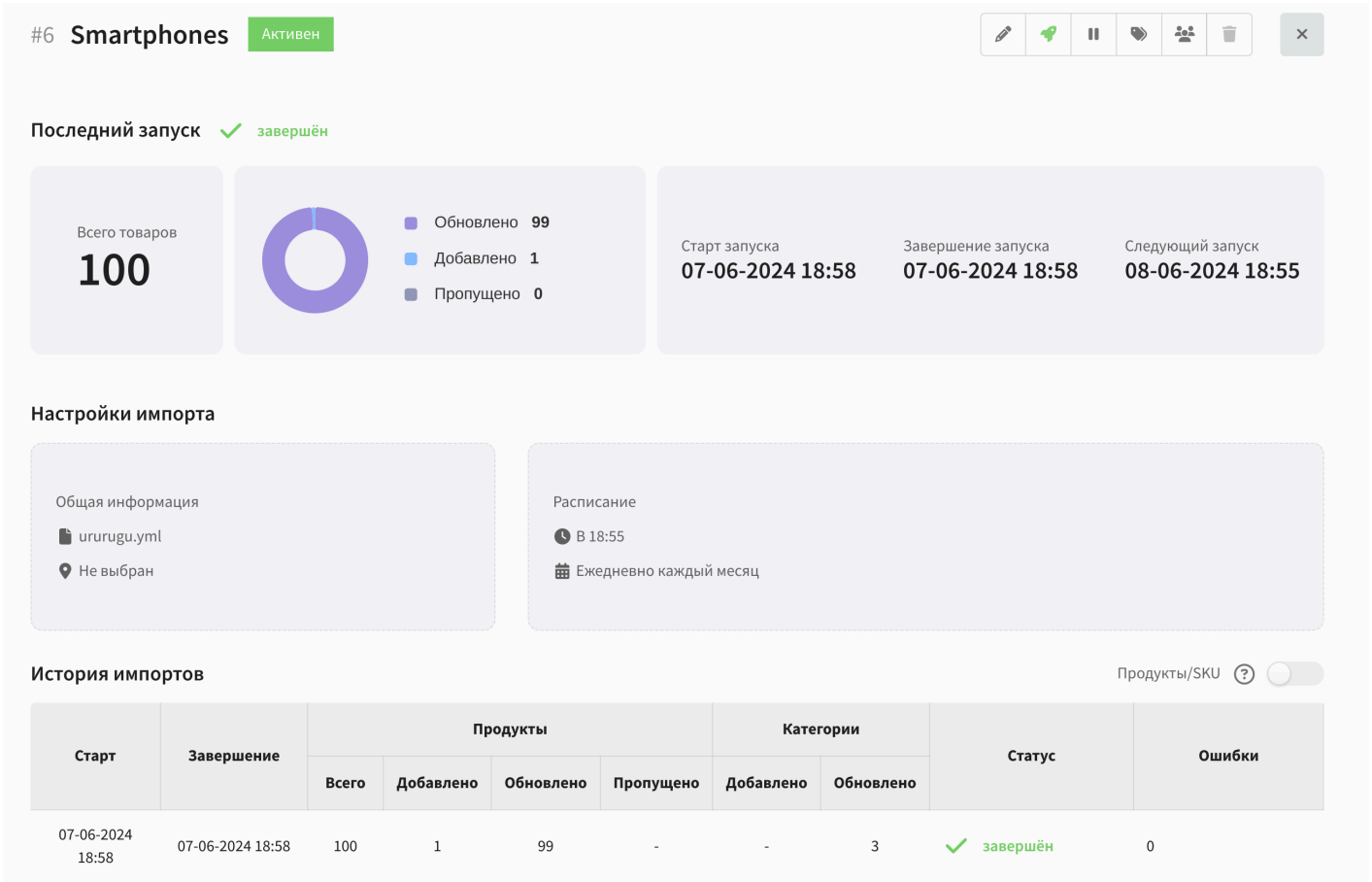
```
./aktool migrate importfeed --config /opt/akd/config/main.json
```

3. Запустите сервис `webcontrol`

Теперь вы можете настроить синхронизацию продуктов и SKU в Маркете с данными вашего товарного фида. Функция **“Импорт продуктов по расписанию”** позволяет указать ссылку на файл фида и время, в которое будет происходить импорт. Узнать больше о настройке синхронизации с товарным фидом можно **[в этой статье](#)**.

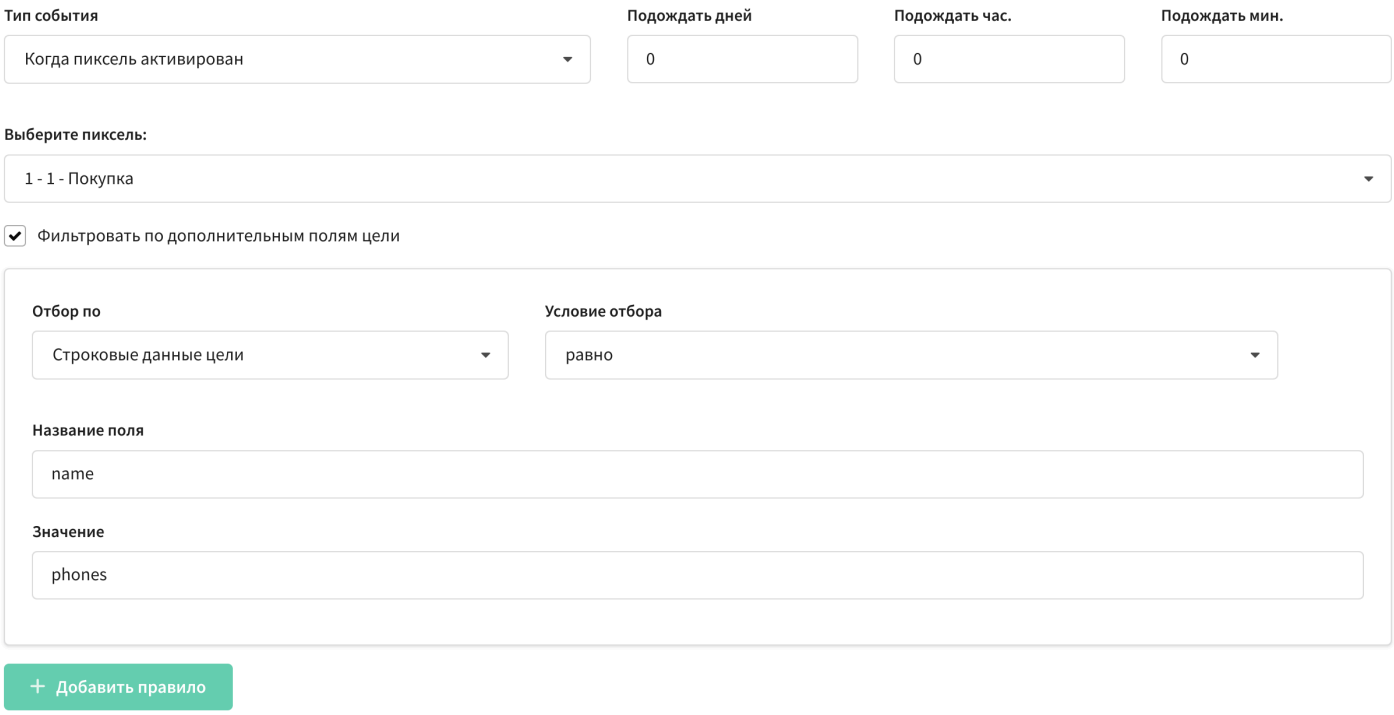


Для каждой синхронизации доступна подробная статистика импорта:



Фильтрация по дополнительным полям для триггера на активацию пикселя

Добавлена возможность указать правила отбора по дополнительным полям при настройке триггера на активацию пикселя. Это позволяет точнее настраивать кампанию под определенную группу клиентов или конкретные товары:



Другие изменения

- Добавлены новые виды подключения: **Basic Authentication** и **Token Authentication**.
- Обновлен пакет системного драйвера и коннектор ClickHouse.

Улучшения и исправления интерфейса

- Устранена ошибка во всплывающей подсказке в списке профилей.
- ID элемента выхода из сценария теперь отображается в карточке профиля.
- Исправлено управление ролями для мастер-пользователей.
- Ссылка на пиксель теперь корректно отображается в истории профиля.
- Маркете при импорте продуктов теперь отображается корректное количество продуктов.
- Формы теперь работают корректно в режиме предпросмотра.
- На панели вставки изображений в редакторе шаблонов отображается корректный URL.
- Информация о создателе и дате последнего изменения в интеграциях и подключениях отображается корректно.
- Из настроек ресурса для push-уведомлений удалено устаревшее Firebase API.
- Список по метрикам в аналитике попапов теперь отображается корректно.
- Улучшено отображение некоторых данных в разделе “Аналитика”.
- Улучшена локализация.
- Исправлены опечатки и некорректные ссылки.

Улучшение производительности

- Исправление некоторых проблем в AKMTA.
- Переделан механизм получения имени глобальной функции `acpopup`.
- Исправлено высокое потребление памяти в процессах `akmtad` и `procworkflow`.
- Исправлено зависание при отправке email-сообщений с некоторыми политиками отправки.

Исправления

- Исправлена ошибка, при которой невозможно было выбрать триггерное условие “Импорт через попап”.
- Профили снова корректно попадают в статистический сегмент по комплексному условию.
- Устранена ошибка “Not calculating statistics with errors” при получении отчета в Панели администратора.
- Аналитика по трафику с фильтром “По попапам” теперь отображается корректно.
- Сортировка по возрасту в Базах данных теперь работает корректно.
- Исправлена ошибка, возникающая при добавлении 18 и более символов в дополнительное поле с типом данных “Целое число” или “Число с точкой” в сущностях маркета.
- Теперь трекинг не приостанавливает работу при слишком длинном ключе шифрования.
- События добавления связи корректно отображаются в истории профиля.
- Дополнительные параметры теперь корректно добавляются в `custom_data` в Action Hooks.
- Исправлены различные проблемы с сегментацией.
- Статистика в сценариях теперь работает корректно.
- Добавлена обработка параметров запроса при загрузке скрипта для управления попапами.
- Устранена ошибка, при которой редактирование кампаний с политиками отправки было доступно только мастер-пользователям.
- Добавлена валидация для сохранения шаблонов для отправки через WhatsApp*.
- Экспорт истории профилей по пушам теперь работает корректно.
- Исправлены ошибки в работе модуля Маркет.
- Добавлена валидация для удаления внешних БД.
- Устранена ошибка, при которой невозможно было создать сегмент через API.
- Политики отправки в триггерной кампании снова работают корректно.
- Улучшена работа Баз наблюдателей при отправке кампании.
- Исправлена ошибка, возникающая при копировании удаленного заказа или создании заказа с именем, как у удаленного заказа.
- Добавлен TTL для ссылок в SMS-сообщениях.
- Исправлена ошибка, из-за которой LDAP-пользователи меняли тип авторизации после сохранения.
- Подключение 2FA для LDAP-пользователя снова работает корректно.
- Пустую строку теперь можно добавить в качестве разделителя в функции `format` в шаблонах.
- Кнопки “Сохранить” и “Применить” недоступны для замороженных аккаунтов.
- При изменении обязательных полей в профиле настройки вкладок больше не сбрасываются.
- Подключение к Devino Telecom с помощью логина и пароля работает корректно.
- Исправлены некоторые ошибки поиска продуктов.
- Картинки с именами менее 12 символов теперь корректно загружаются при импорте HTML шаблона.
- Устранены ошибки, возникающие при обновлении триггерной кампании.
- Стоп-список теперь пропускает домены, которые содержат более 6 символов после точки.
- Устранена паника при попытке сохранить удаленный объект в сегментах.
- Устранена паника при при добавлении шаблона с помощью API.
- Устранены паники в `cookie_saver`, `procintegras`, `xls.go`, `proctask` и `tariffcontroller`.

References

PL-15888 PL-12880 PL-15993 PL-15991 PL-15865 PL-14353 PL-15869 PL-15926 PL-15840 PL-15979 PL-10455 PL-15813 PL-15970 PL-14286 PL-13837 PL-15334 PL-16037 PL-15779 PL-15939 PL-15662 PL-16032 PL-13906 PL-15800 PL-16112 PL-16001 PL-16048 PL-10320 PL-14477 PL-15907 PL-14312 PL-16013 PL-14015 PL-15104 PL-15989 PL-15712 PL-15693 PL-15700

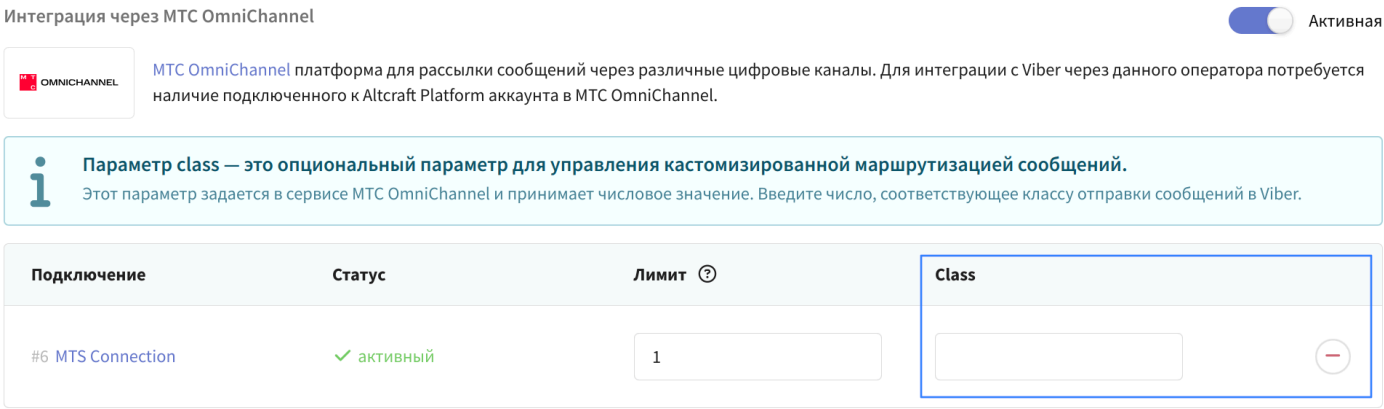
---

v2024.2.68.2.749 - Viber™ integration updates

Основные изменения

Улучшение интеграции канала Viber™ через MTC OmniChannel

В настройках интеграции канала с помощью шлюза MTC OmniChannel добавлен параметр `class`, который управляет маршрутизацией сообщений в сервисе:



Исправления

- Скорректированы тексты и ссылки в интерфейсе платформы, в том числе в мобильной версии.
- Улучшен HTML-парсинг при импорте шаблона сообщения.
- Исправлена ошибка, при которой было невозможно удалить email из профиля.
- Улучшена стабильность работы платформы.
- Исправлены ошибки JavaScript в разделах “Сегменты” и “Интеграции”.
- Устранена ошибка, возникавшая при открытии предпросмотра шаблона по умолчанию.
- Тестовая отправка формы теперь корректно сохраняет результат заполнения.
- Устранена ошибка, при которой в MongoDB генерировалось несколько запросов при сохранении сценариев.
- Удаление профилей по сегменту теперь проходит быстрее.
- Исправлены ошибки, возникавшие при создании ресурса.
- Фильтр по статусу подключения в списке подключений работает корректно.
- Исправлен статус у новых подключений к Яндекс.Аудитории.
- AKPush больше не создает дубликаты профилей при очистке хранилища и cookie.
- Исправлены ошибки при массовом добавлении групп в Базу данных.
- Поиск по шаблонам в отчёте “Карта кликов” работает корректно.
- Пользовательская ссылка отписки теперь отображается в “Карте кликов”.
- Исправлена ошибка, при которой невозможно было удалить регион и категорию в разделе “Продукты”.
- Очереди AKMTA в RabbitMQ теперь очищаются после удаления сендера в Панели администрирования.
- Счетчик сообщений в очереди AKTMA отображает корректные значения.
- Исправлены некоторые ошибки отправки в AKMTA.
- Регулярные выражения в запросах сегментации по структуре профиля теперь работают корректно.
- Обновлен драйвер MongoDB до актуальной версии.
- API-запрос на получение данных сводного отчета возвращает актуальные параметры.
- Улучшен контроль привилегий и прав пользователей для некоторых действий (например, для экспорта задач).
- Устранены ошибки с API HTTPS в некоторых библиотеках.
- Отписка от канала корректно отображается в истории действий профиля при попадании в стоп-список.

- Исправлена ошибка, при которой в триггерных кампаниях нельзя было указать не зарегистрированную до этого цель.
- Формы теперь корректно сохраняют данные после принудительной отправки на email.
- Устранены паники в akmtad.

References

PL-15737 PL-16402 PL-16405 PL-16407 PL-16174 PL-8775 PL-16439 PL-16467 PL-16461 PL-16058 PL-15091 PL-14838 PL-16151 PL-16059 PL-16021 PL-16445 PL-16394 PL-16420 PL-16423 PL-5732 PL-16401 PL-16121 PL-16175 PL-16278 PL-15695 PL-13728 PL-15668 PL-16101 PL-13916 PL-7600 PL-15192 PL-15466 PL-16395 PL-16187 PL-16409 PL-14836 PL-13393 PL-16269 PL-16220 PL-16208 PL-16110 PL-3435

---

v2024.2.68.2.950 - Improved bot filtering mechanism

Основные изменения

Улучшение фильтра ботов

Улучшен механизм фильтрации действий ботов в платформе. Настройки фильтрации содержатся в двух конфиг-файлах: `bots_ip.json` и `bots_ua.json`. С их помощью можно установить действия, которые будут применятся к обнаруженным действиям ботов. Узнать подробнее о работе этих файлов можно **здесь**.

Другие изменения

- Теперь можно настраивать в конфигурации платформы таймаут на соединение к базе данных с историей событий.
- Улучшена производительность некоторых процессов платформы.

Исправления

- Исправлена ошибка, при которой элемент сценария “API Запрос” передавал в платформу некорректные данные параметров запроса.

> **CAUTION**
>
> Для исправления данной ошибки пользователю необходимо произвести **миграцию**. Для этого до запуска новой версии сделайте следующие действия:
>
> 1. Остановите сервисы `webcontrol` и `procworkflow`
> 2. Запустите `aktool` **из новой сборки** с помощью команды:
>
> ```
> ./aktool workflow api_nodes_convert --config /root/MarketingPlatform/config/main.json
> ```
>
> 3. Запустите сервисы `webcontrol` и `procworkflow`
>
> Справка по мигратору:

```
Usage:
  aktool workflow api_nodes_convert --accs=1,2... --wfs=1,2... [flags]

Examples:
./aktool workflow api_nodes_convert --accs=1,2 --wfs=1,2

Flags:
  -a, --accs ints   Account IDs (comma-separated list, empty means all)
  -h, --help        help for api_nodes_convert
  -w, --wfs ints    Workflow IDs (comma-separated list, empty means all)

Global Flags:
      --config string                     config path (default
"/opt/akd/config/main.json")
  -o, --output string                     file path to output in the result of the
command (default "stdout")
```

- Исправлена ошибка, при которой знак пробела неправильно кодировался платформой в элементе сценария “Вызов API”.
- AMP-версия шаблона письма теперь формируется корректно.
- Исправлена ошибка, из-за которой в сценарий не попадали профили при активации пикселей.
- Устранена ошибка, при которой процессы `procpush` и `tariffcontroller` не прекращали работу без принудительной остановки.
- Теперь данные о предыдущих заполнениях формы попапа не удаляются после переименования полей этой формы.
- Исправлены ошибки, связанные с некорректным удалением сущностей из профилей.
- Сторонние переменные шаблонов писем теперь корректно конвертируются в платформе при импорте шаблонов через вставку кода.
- Исправлена ошибка, при которой определение местоположения при импорте профилей в некоторых случаях работало некорректно.
- Улучшено отображение некоторых элементов интерфейса в мобильной версии.
- Скорректировано описание некоторых параметров в платформе.
- Исправлены некоторые ошибки в интерфейсе платформы.
- Устранена паника в `tzdata`.

References

PL-16513 PL-16476 PL-16441 PL-15870 PL-16283 PL-16134 PL-16351 PL-16499 PL-16490 PL-15447 PL-15328 PL-16238 PL-15859 PL-16469 PL-15646 PL-14839 PL-16186 PL-16133 PL-16198 PL-16243 PL-16160 PL-16028

---

v2024.2.68.2.1131 - SSL support for MongoDB

Другие изменения

- Добавлена поддержка SSL для подключения к MongoDB.
- Добавлено условие сегментации “Домен не в списке” для поля “Email подписка”.

Улучшения интерфейса

- Теперь в карточке профиля указывается id элемента, из которого профиль вышел из сценария.
- Улучшен дизайн интерфейса редактора форм в платформе.
- Исправлены ошибки в тексте некоторых подсказок.

Исправления

- Исправлена ошибка, при которой профили не попадали в сценарии с сегментом по расписанию по условию “Участие в динамическом сегменте”.
- Теперь над элементами сценария корректно отображается информация о возникающих ошибках.
- Исправлена ошибка, при которой профили не попадали в сценарии, настроенные на захват при активации пикселя.
- Исправлена ошибка в запросе `products`, возникавшая при передаче идентификатора, содержащего одинарные кавычки.
- Теперь в элементе сценария “Обновить профиль” можно передать api-переменную для поля с типом “Число”.
- Устранена ошибка “invalid syntax” в GET-запросе на получение истории действий нескольких профилей.
- Исправлены некоторые уязвимости в интерфейсе.
- Теперь при сохранении профиля с невалидными данными в одном из полей платформа выведет ошибку и не сохранит профиль.
- Исправлена ошибка, при которой тег с запятой в названии определялся как два разных тега.
- Обновлены некоторые библиотеки во избежание потенциальных уязвимостей.
- Устранены паники в `proctask`, `webcontrol` и `procworkflow`.

References

PL-16567 PL-16563 PL-16513 PL-16557 PL-16276 PL-14369 PL-16535 PL-16526 PL-16105 PL-16544 PL-16536 PL-15864 PL-16517 PL-16532 PL-16496 PL-16103 PL-16373 PL-15154 PL-16471 PL-16449 PL-16307 PL-16236 PL-16137 PL-16487 PL-16422 PL-16210 PL-16082

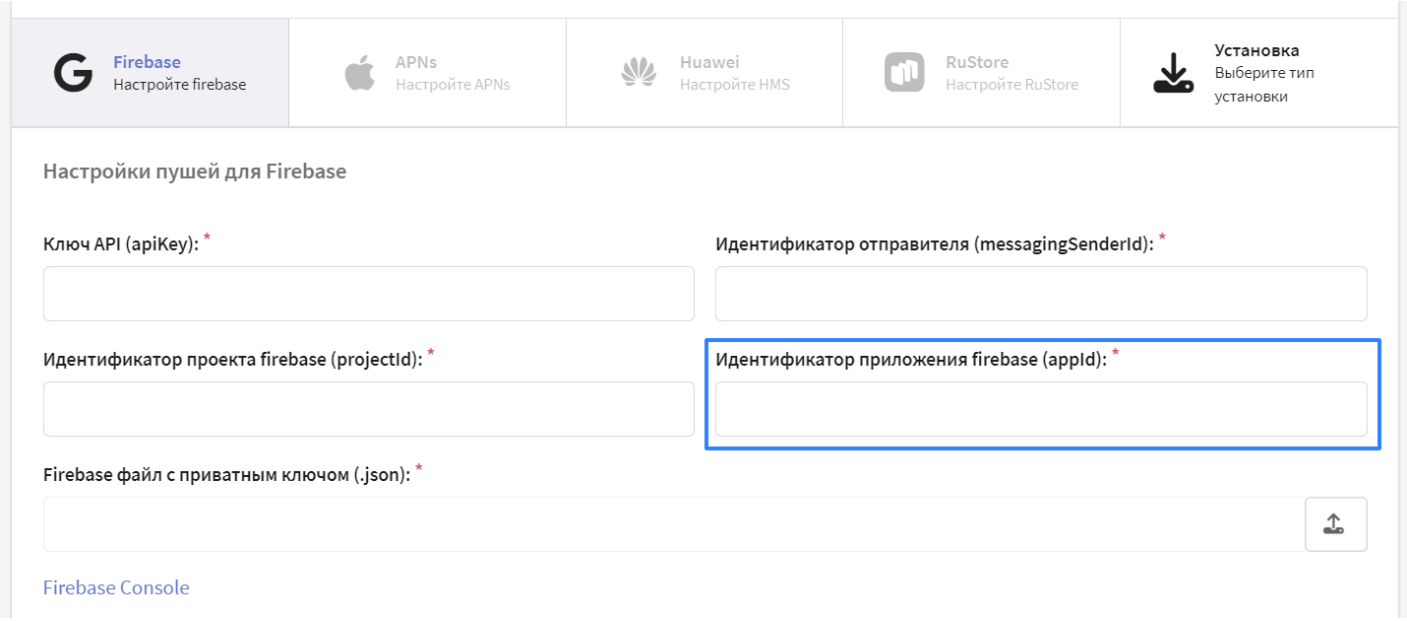
---

v2024.2.68.2.1378 - Firebase API updates

Восстановление работы web push-уведомлений

Сервис Google Firebase изменил некоторые API-хендлеры, в связи с этим для нормальной работы web push-уведомлений требуется провести **миграцию**:

- Во всех ресурсах, скрипты которых размещены на сайтах, необходимо заполнить поле **Application ID** в настройках Firebase:



Значение Application ID можно найти в настройках проекта в Firebase Console.

- Если на сайтах размещены статичные **ak-push.js** и **service-worker.js**, то после редактирования и сохранения ресурса необходимо заново загрузить эти скрипты на сайт.

Другие изменения

- Теперь изображения в редакторе шаблона вставляются с `https` в начале ссылки, а не с `http`, как было раннее.
- Связь между входным элементом сценария и другими элементами теперь добавляется корректно.
- Исправлена ошибка, при которой невозможно было сгенерировать Промокод без ограничения на количество.
- Максимальное количество символов для домена первого уровня в платформе увеличено до 128.
- Исправлены ошибки, возникавшие при взаимодействии с пустой базой профилей.
- Устранена ошибка, при которой не происходила отправка данных на email после заполнения формы пользователем.
- Теперь в настройках ресурса можно указать трекинг-домен с однобуквенным поддоменом.
- Исправлена ошибка, при которой сценарии с сегментом по расписанию не работали корректно после первого запуска.
- Ускорена загрузка страниц в пользовательском интерфейсе при использовании LDAP-авторизации.
- Устранена ошибка, при которой внесенные в Воронку изменения не сохранялись в платформе.
- Улучшена валидация параметров в некоторых API-запросах.
- Выпадающие списки корректно работают в последних версиях Google Chrome.
- Отчет по аналитике трафика теперь выгружается корректно.
- Исправлены ошибки интерфейса в некоторых разделах платформы.
- Скорректированы описания некоторых ошибок в платформе.

References

PL-16637 PL-16634 PL-15610 PL-16179 PL-16146 PL-16625 PL-16620 PL-16150 PL-16619 PL-14996 PL-16469 PL-16611 PL-16598 PL-14729 PL-16586 PL-16135 PL-16483 PL-16473 PL-16030 PL-14749

---

v2024.2.68.2.1788 - Order status API

Основные изменения

- Добавлен API-метод для проверки статуса заказа в Маркете. Подробнее.

Другие изменения

Улучшение интерфейса

- Исправлены ошибки верстки в разделе “Формы” на мобильных устройствах.
- Выпадающие списки в модальных окнах сценариев корректно закрываются при сбросе фокуса.
- Исправлены неработающие ползунки прокрутки на некоторых страницах платформы.

Исправления

- Устранены ошибки, связанные с достижением максимального лимита активных сессий в MongoDB.
- Запись о выходе профиля из сценария больше не перезаписывается при очистке сценария от всех профилей.
- Исправлены ошибки, возникающие при работе динамического сегмента с условием отбора по пользовательскому полю.
- Устранена дубликация web push-уведомлений в браузерах.
- Теперь связи корректно перезаписываются при редактировании из профиля.
- Исправлена ошибка, при которой время создания пользователя обнулялось при обновлении данных о нем.
- Теперь при клонировании сценария в нем не сохраняются сообщения об ошибках в исходном сценарии.
- Устранена ошибка, при которой недоступный ресурс отображался в списке без соответствующей пометки.
- Пользовательский `custom_json` в шаблонах push-уведомлений теперь работает корректно.
- Исправлена ошибка, при которой неверно отображался последний редактировавший сценарий пользователь.
- Событие отписки при запрете уведомлений через Safari теперь корректно регистрируется платформой.
- Исправлена ошибка, при которой в Панели администратора удаленный ISP был доступен для ноды.
- Улучшена стабильность процессов `webcontrol`, `akmta`.

References

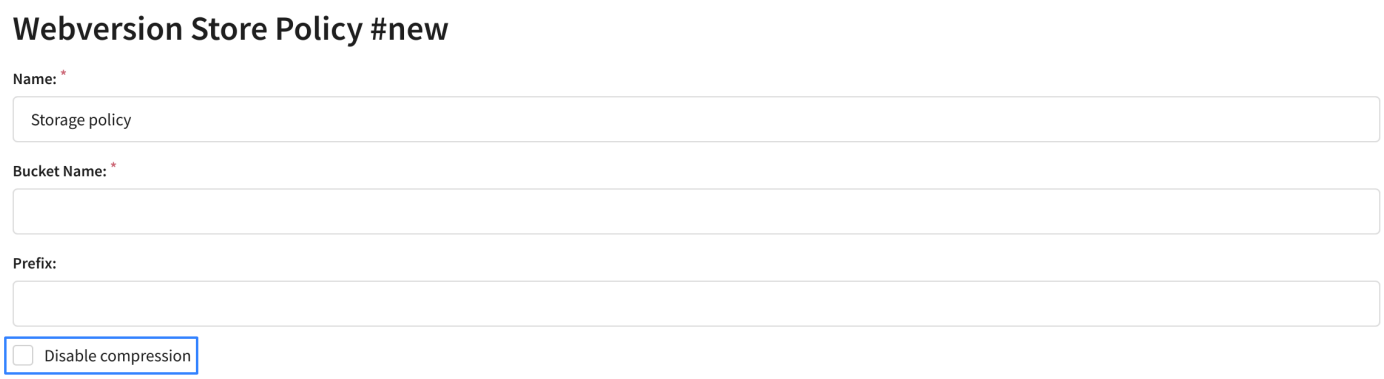
PL-16629 PL-16658 PL-16650 PL-16646 PL-16593 PL-16425 PL-16446 PL-16566 PL-15758 PL-16056 PL-14880 PL-16651 PL-16630 PL-16472 PL-16639 PL-16266 PL-16615 PL-16428 PL-16596 PL-16257 PL-16470

---

v2024.2.68.2.1957 - Web version compression mode

Основные изменения

- При создании политики хранения веб-версий в панели администратора добавлена опция для отключения сжатия:



- В журнале кампании теперь указывается время получения сегмента.

Другие изменения

Улучшение интерфейса

- Добавлено отображение времени следующего запуска кампании Мультивариантного тестирования.
- Устранена ошибка, при которой всплывающие окна подсказок не исчезали спустя некоторое время.

Исправления

- Исправлена ошибка, при которой в платформе не отображалась очередь кампании на сендерах.
- Устранена ошибка, при которой платформа не учитывала летнее и зимнее время для часовых поясов в некоторых регионах.
- Журнал кампании снова корректно работает при открытии из редактора кампаний.
- Устранены различные ошибки, связанные с обновлением профиля в платформе.
- Теперь профили с подпиской на push-уведомления Apple Safari Native сохраняются корректно.
- Исправлена ошибка, при которой можно было создать кампанию с шаблоном-черновиком.
- Веб-версии писем корректно сжимаются перед сохранением.
- Импорт в Маркет с помощью файла теперь работает корректно.
- Исправлена ошибка, при которой кэш результата в превью SQL-запросов шаблонов/сегментации не сбрасывался.

References

PL-16095 PL-16671 PL-16655 PL-16281 PL-10310 PL-16553 PL-16609 PL-11837 PL-16132 PL-15971 PL-15761 PL-16666 PL-16661 PL-16480 PL-16617 PL-16633 PL-8356 PL-16274 PL-16292 PL-16221 PL-13312

---

v2024.2.68.2.2206 - Sendsay, Fixes

Основные изменения

Интеграция с Sendsay

Добавлена возможность отправки email-писем через SMTP-релей Sendsay. Администратору платформы необходимо настроить релей в AKMTA-сендере. Для получения статусов доставки используются вебхуки.

Другие изменения

- Расширены API-запросы на получение истории профилей: `/api/v1.1/subscribers/history_get` и `/api/v1.1/subscribers/history_get_all`. Ответ возвращает дополнительные поля, позволяющие идентифицировать уникальное событие и получить по нему более детальную информацию.
- Удалены процессы `tariffcontroller` и `procnotify`, их функционал перенесен в `proctask`.
- Улучшен импорт в платформу email-шаблонов, выгруженных из сторонних сервисов. Теперь изображения могут храниться на одном уровне с HTML-файлом или в папке с одним из следующих названий: `images`, `image`, `img`, `imgs`, `pictures`, `pics`. Архив с таким email-шаблоном не должен содержать файлы `text.txt`, `subject.txt`, `template.json`, так как их наличие может вызывать ошибки при импорте. При этом **предыдущая структура** архива с шаблоном также продолжает работать без изменений.
- Показатель “Web-импорт” в **отчёте по приросту** теперь включает общее количество профилей, добавленных в базу как через формы, так и через попапы, размещённые на сайте.

CAUTION

Ранее “Web-импорт” учитывал только данные по импорту через форму. Чтобы обновить показатель и включить данные о приросте через попап, необходимо выполнить миграцию.

Миграция выполняется после обновления платформы до новой версии и не требует предварительной остановки каких-либо процессов:

1. Удалите представление и таблицу `engagement_report`:

```
./aktool stats delete engagement --config=/opt/akd/config/main.json
```

2. Инициализируйте таблицу в новом формате:

```
./aktool stats init engagement --config=/opt/akd/config/main.json
```

3. Запустите перерасчёт записей в таблице:

```
./aktool stats fill engagement --config=/opt/akd/config/main.json
```

При успешном выполнении в консоль будет выведено время заполнения таблицы. Пример: `complete fill stats table for 183.707488ms`.

Исправления

- Исправлена проблема с отправкой веб-пушей в браузер Safari через APNS.
- Восстановлена возможность выбора пула в настройках сендера.
- Исправлена проблема с добавлением дополнительных параметров в `custom_data` в хуках.
- Устранена ошибка, из-за которой при добавлении поля `_createtime` в `custom_data` не выполнялись ретраи хуков.
- Сегмент с отбором по полю с типом “Список” и условием отбора “Равно” работает корректно.

- Счётчик над нодой SMS-кампании в сценарии снова показывает количество профилей.
- Исправлена проблема с дублированием еженедельных и ежедневных отчетов для пользователей платформы.
- При тестировании регулярной кампании в журнале отображается корректное время получения сегмента.
- Триггерная кампания, настроенная на пользовательское событие попапа, теперь успешно сохраняется.
- Скорректирована работа параметра `next_from_id` в ответе на запрос `/api/v1.1/subscribers/history_get_all`
- Отсутствие поля `categories` в запросе `/api/v1.1/market/sku/import` больше не приводит к ошибкам.
- Подключён компонент `ampcarousel` в демонстрационной AMP-версии шаблона.
- Улучшена валидация в триггерном элементе сценария “Захват сегмента профилей по расписанию” и в подписках на Viber™ и WhatsApp*-каналы.
- Улучшена стабильность процесса `procactions`.
- Исправлены незначительные недочёты интерфейса платформы.

References

PL-15845 PL-16842 PL-15693 PL-16694 PL-16796 PL-16763 PL-16805 PL-15149 PL-16743 PL-16737 PL-16314 PL-15822 PL-16712 PL-16654 PL-16685 PL-16704 PL-11654 PL-15426 PL-16122 PL-16599 PL-16663 PL-16069 PL-16631 PL-16656 PL-16093 PL-13768 PL-16636 PL-16714 PL-16709 PL-16614 PL-16116 PL-16533 PL-16644 PL-16676 PL-14390 PL-16476 PL-16259 PL-16608 PL-15941

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*
***Viber является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

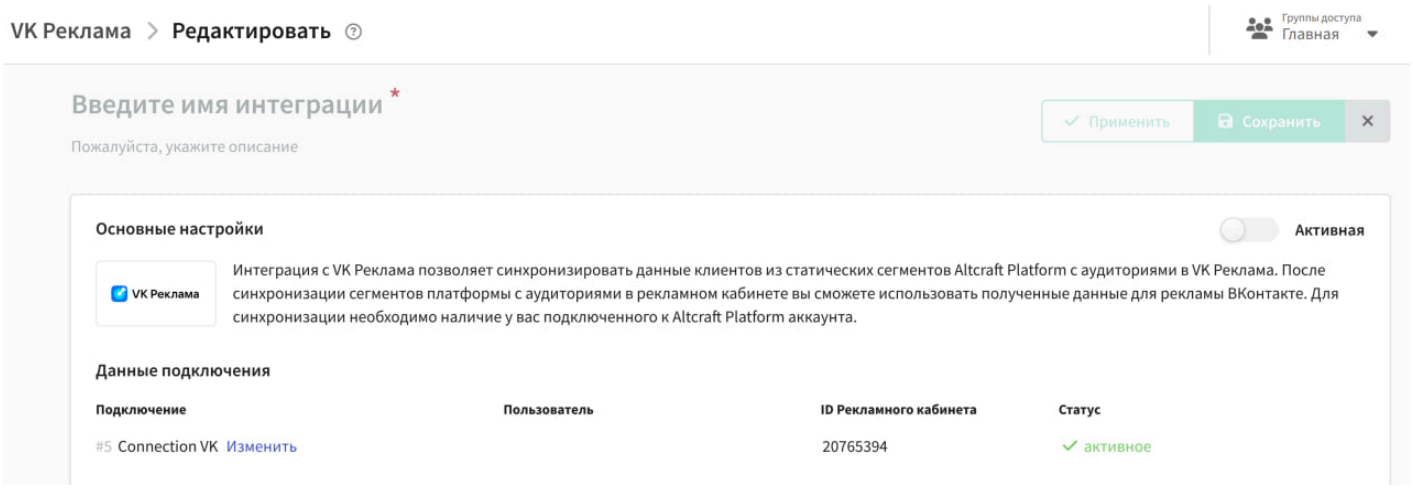
v2024.1.68

v2024.1.68.0 - VK Ads™, new AKMTA sender settings

Основные изменения

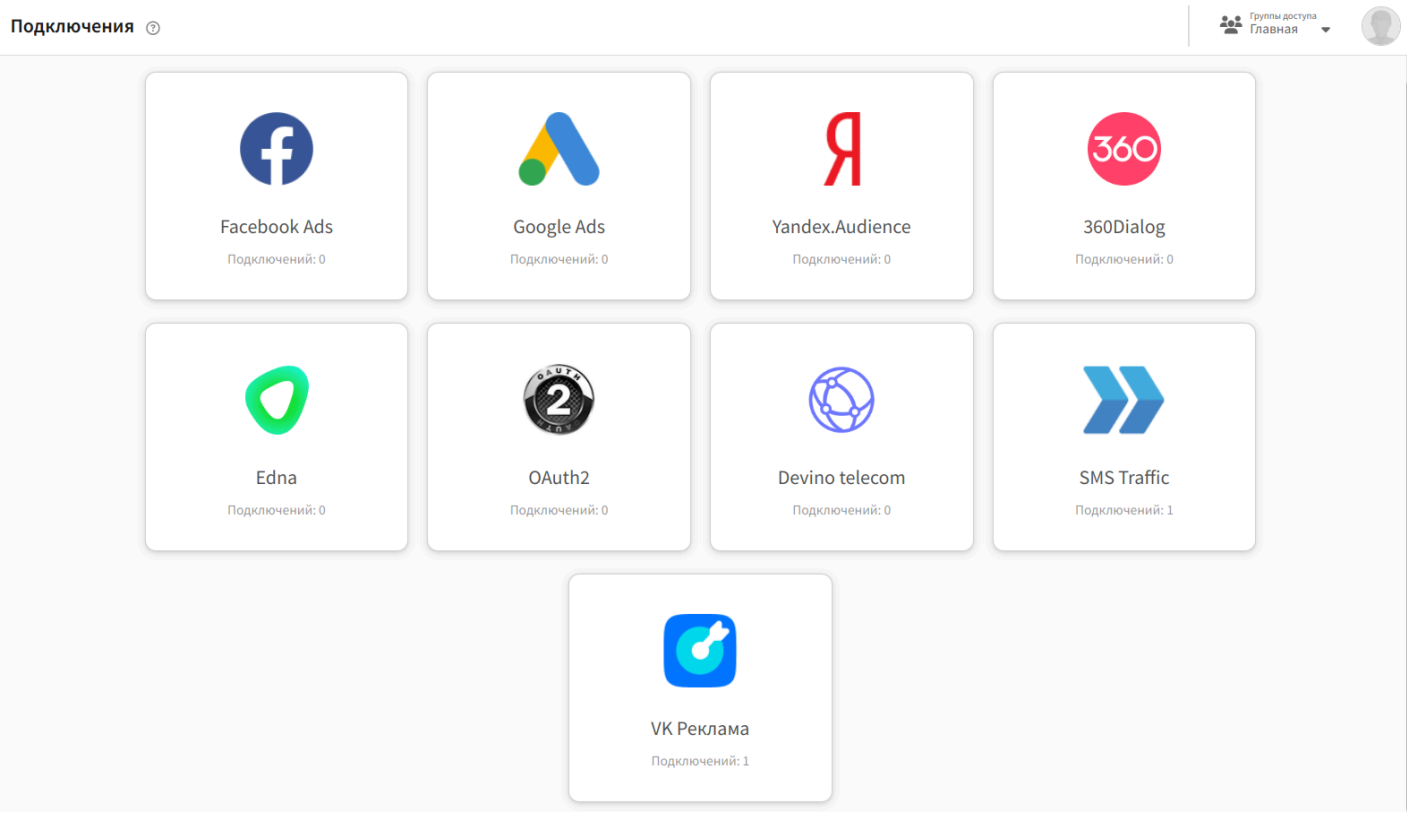
VK Реклама™

	Добавлена интеграция с **VK Рекламой™**. Новая функция позволяет переносить сегменты данных из Altcraft Platform в аудитории в ВК Рекламе™. Статьи о [создании подключения](/user-guide/settings/connections/vk-connection/) и [настройке интеграции](/user-guide/integrations/vk/) доступны на портале документации.



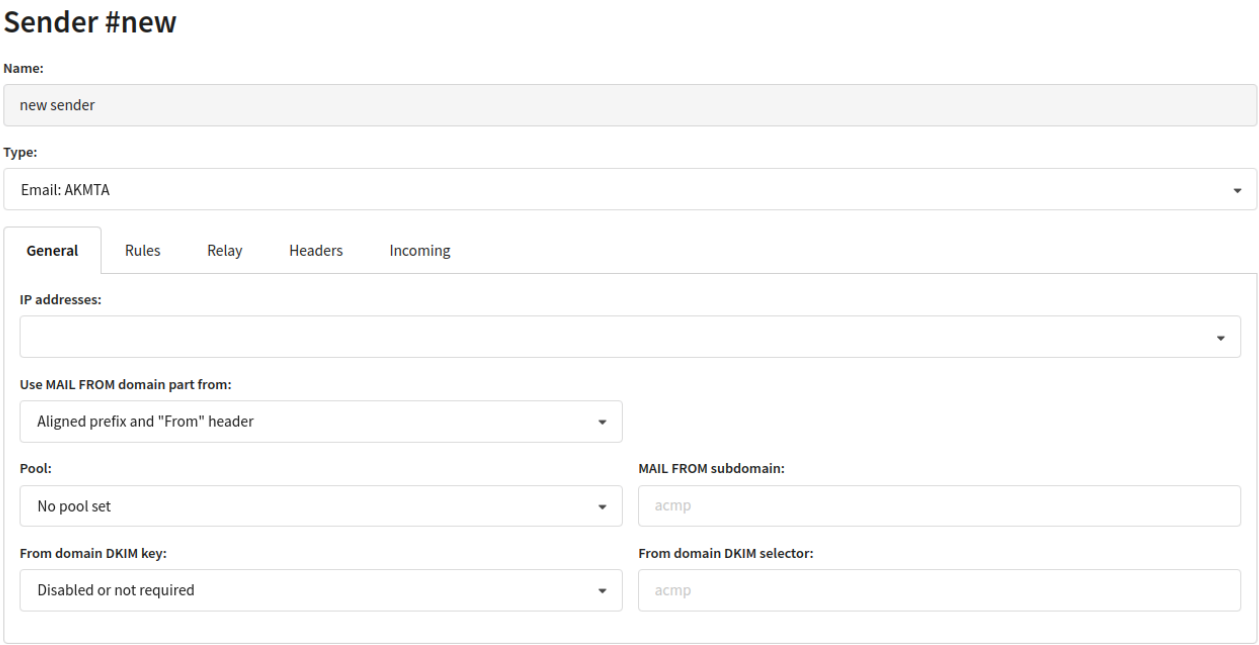
Обновлено меню подключений в панели управления

Удалены пункты меню Viber™ и WhatsApp*, добавлены отдельные кнопки для создания подключений с помощью **edna, 360dialog, Devino и SMSTraffic**.



Обновление панели администратора

- При создании AKMTA-сендера добавлен новый тип для доменной части MAIL FROM — **Aligned prefix and “From” header**. Такая настройка позволяет одновременно отслеживать статистику отправки и осуществлять выравнивание SPF. Для её использования необходимо создать и настроить пул.
- **Добавлены пулы**. Пул — набор ресурсов, необходимых для отправки писем от определенного домена. Подробную инструкцию по его созданию и настройке можно прочитать здесь.
- Настройки AKMTA в разделе “Управление сендерами” разделены на вкладки **General, Rules, Relay, Headers** и **Incoming**.



Другие изменения

- Улучшен поиск в списке пользователей: добавлен поиск по email и номеру телефона, устранены различные ошибки при поиске элементов.
- Фамилия пользователя теперь не видна специалистам поддержки в Jivo.
- Убрано ограничение на количество отображаемых столбцов в таблицах данных.
- Также в таблицах данных добавлена возможность посмотреть и скопировать значение столбца в тех случаях, когда оно не умещается полностью в поле.

Улучшения производительности

- Оптимизирован api-запрос на массовый импорт и запуск профилей в сценарий
- Исправлено чрезвычайно высокое потребление памяти у akmtad
- Устранены зависания визуального редактора при клике на элементы

Обновления и исправления интерфейса

- Исправлено отображение недоступных действий для замороженных аккаунтов.
- Обновлен интерфейс в модулях “Пиксели” и “Карта кликов”.
- В отчётах по каналам теперь снова доступна колонка “% Открытий”.
- Исправлены ошибки с отображением меню дополнительных полей.
- Теперь время создания/обновления подключения отображается корректно.
- Скроллбар во всплывающих меню теперь отображается корректно.
- Отчет журнала аудита теперь выгружается корректно.
- Скорректированы некоторые наименования и переводы.

Исправления

- В отчете по недоставкам теперь корректно отображается причина недоставки.
- Исправлена ошибка, при которой в поле pop-up редактора не вставлялись переменные.
- При массовом обновлении профилей push-подписки добавляются только на указанный провайдер.
- Исправлены различные ошибки в разделе “Лояльность”.
- Исправлена ситуация, когда в пустой элемент “Текст” в визуальном редакторе нельзя было добавить текст.
- Устранены ошибки, возникающие при удалении базы данных и связанных с ней сегментов.
- Улучшено логирование.
- Сохранение цели в триггерных компаниях теперь работает корректно.
- Специальные символы теперь корректно экранируются в LDAP.
- Устранена ошибка, при которой отправка больше 10 смс на шлюзе МТС не работала.
- Устранены ошибки, связанные с некорректным заполнением даты и времени в различных элементах.
- Устранены ошибки, связанные с аналитикой, такие как ложные клики от ботов, счёт hardbounce-ответов и другие.
- В журнал кампании добавлена запись об ошибке о превышении количества символов в SMS.
- В триггерных кампаниях с типом события “При добавлении/изменении подписки” можно выбрать статус “Не подтверждён”.
- Устранены повторяющиеся ошибки “context canceled” в логах procworkflow.
- Удаление сущности теперь корректно записывается как обновление.
- Устранены паники в procintegras, webcontrol, proctask и regen_preview_msgs.
- Устранены паники при опубликовании pop-up’а и при использовании 360dialog.

References

PL-15764 PL-15754 PL-15741 PL-15735 PL-15731 PL-15730 PL-15727 PL-15725 PL-15711 PL-15703 PL-15699 PL-15688 PL-15675 PL-15654 PL-15650 PL-15644 PL-15634 PL-15633 PL-15630 PL-15612 PL-15608 PL-15604 PL-15601 PL-15596 PL-15588 PL-15579 PL-15574 PL-15567 PL-15553 PL-15509 PL-15333 PL-15286 PL-15016 PL-14793 PL-14439 PL-14291 PL-13485 PL-11892 PL-15573 PL-15572 PL-15463 PL-15387 PL-15056 PL-1500 PL-14992 PL-14950 PL-14900 PL-14718 PL-14547 PL-13257 PL-13012 PL-12585 PL-12582 PL-12500 PL-12499 PL-15754 PL-15633 PL-15630 PL-15612 PL-15604 PL-15574 PL-15509 PL-11972

---

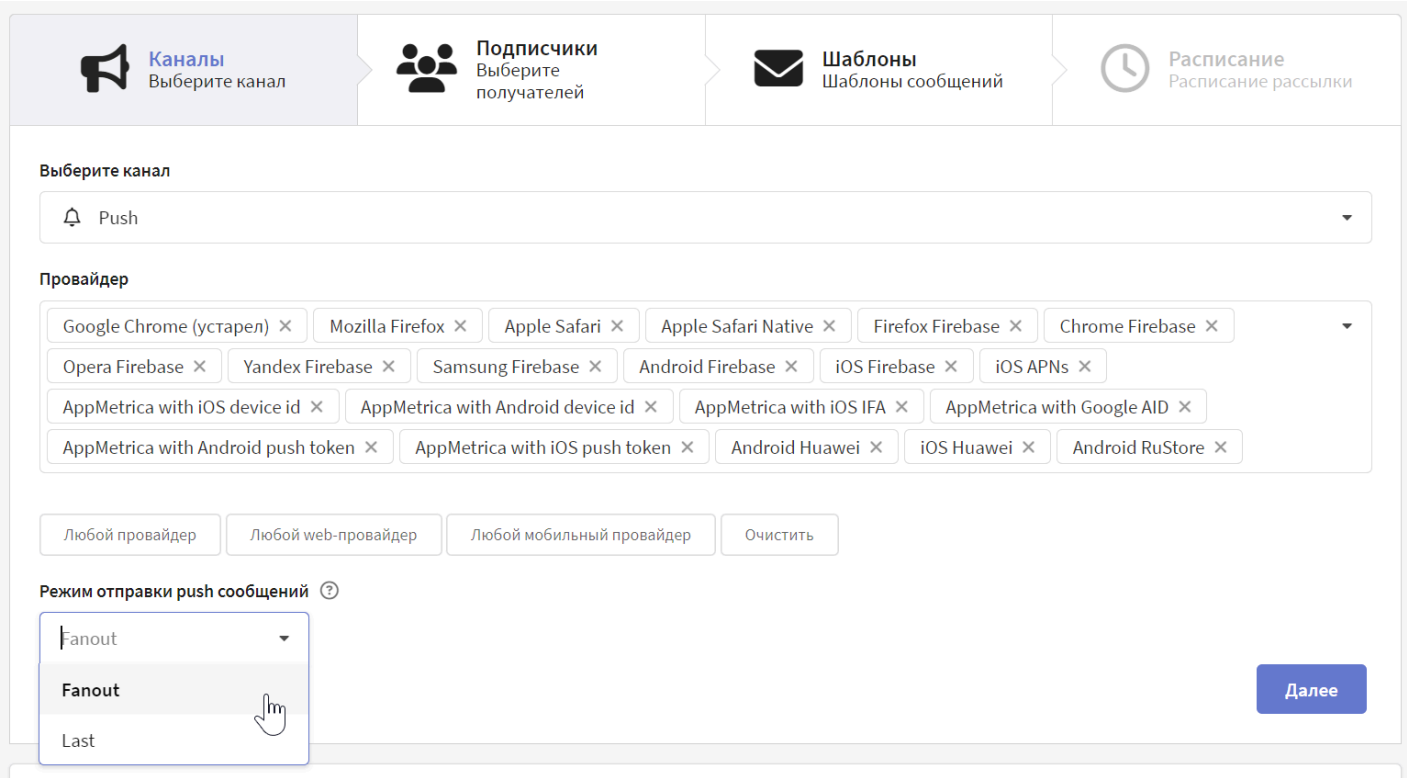
v2024.1.68.626 - New push sending mode, Fixes

Основные изменения

Новый режим отправки пушей

При настройке кампаний через push-канал теперь можно выбрать режим отправки push-сообщений:

- **fanout** (по умолчанию) — отправка на все подписки профилей, которые соответствуют заданным в кампании фильтрам.
- **last** — отправка на подписку с самой свежей датой создания.



Изменения в таблице статистики

Поскольку сводный отчет возвращал некорректное количество событий, была изменена структура таблицы `channel_report`. При обновлении требуется пересоздать таблицу:

```
./aktool stats delete channel_report
./aktool stats init channel_report
./aktool stats fill channel_report
```

CAUTION

Перед выполнением миграции требуется остановить сервис `procevent`.

Другие изменения

- Улучшена валидация ссылок в push-шаблоне. В журнал добавлена ошибка “Количество невалидных ссылок в сообщении”.
- В журнале кампании запись “Количество обработанных подписчиков” переименована в “Количество найденных подписчиков”.
- Добавлена возможность сохранять мастер-пользователя без указания ролей.
- Добавлено логирование в syslog отправки через relay.
- Улучшена валидация полей в шаблоне для канала Telegram Bot.
- Обновлена логика работы команды `./aktool stats`, которая используется для работы с хранилищем отчетов. Заполнение таблицы теперь происходит следующим образом: `./aktool stats fill table_name`. Кроме того, теперь можно удалять `channel_report` и `traffic_report` по отдельности.

Исправления

- Исправлена работа сегмента по целям с комплексным условием.
- В элементе сценария JSON фильтр теперь можно использовать оператор ИЛИ и выполнять арифметические действия.
- В настройках кампании при удалении сегмента, по которому происходила отправка, теперь отображается корректный сегмент.
- Исправлены недочеты поиска в ресурсах, в ролях и в маркете.
- В тестовых SMS-сообщения дата подставляется в указанном формате.
- В быстром сегменте при отборе профилей по номеру телефона пробелы между цифрами никак не влияют на результат.
- В визуальном редакторе изображения корректно показываются и в десктопной, и мобильной версиях.
- В визуальном редакторе отображается футер письма со ссылкой отписки.
- Ссылки в push-шаблонах после добавление UTM меток ведут на указанную страницу.
- Теперь невозможно сохранить шаблон email-письма с вложением больше 12МБ.
- В формах исправлена работа функции “Скрывать кнопку отправки”, которая ранее не давала отправить форму, если не были выбраны все варианты в обязательном элементе “Один вариант”.
- Экспорт истории профилей статического сегмента, созданного при импорте, выгружает данные без ошибок.
- В сценариях исправлена работа элемента “Условие” с включенной опцией “Ожидать условие в течение заданного времени”.
- В захвате событий для одного push-сообщения возвращается один и тот же `send_message_id` для всех типов событий.
- Устранена ошибка при удалении профиля через API.
- Исправлена ситуация, когда через API можно было создать push-подписку, не указав `subscription_id`.
- Скорректирован механизм 2FA для администраторов.
- Экспорт журнала аудита в панели администрирования больше не вызывает ошибки.
- Изменения в настройках сендера с типом SMS: Rapporto SMS успешно сохраняются.
- Исправлена ситуация, когда в логах AKMTA наблюдались дубликаты с ответами об отклонении писем.
- Оптимизирована производительность некоторых процессов платформы.
- Устранены некоторые недочеты интерфейса.

References

PL-15986 PL-15983 PL-15964 PL-15959 PL-15952 PL-15886 PL-15876 PL-15849 PL-15828 PL-15823 PL-15820 PL-15802 PL-15759 PL-15743 PL-15739 PL-15719 PL-15715 PL-15684 PL-15658 PL-15649 PL-15609 PL-15577 PL-15486 PL-15432 PL-14887 PL-14795 PL-14446 PL-14072 PL-10471 PL-10464 PL-10450 PL-10097 PL-9688 PL-7740 PL-15918 PL-15917 PL-15914 PL-15912 PL-15911 PL-15905 PL-15903 PL-15898 PL-15891 PL-15885 PL-15882 PL-15881 PL-15871 PL-15863 PL-15862 PL-15858 PL-15839 PL-15816 PL-15804 PL-15803 PL-15797 PL-15796 PL-15786 PL-15785 PL-15781 PL-15778 PL-15770 PL-15746 PL-15745 PL-15744 PL-15740 PL-15738 PL-15732 PL-15697 PL-15689 PL-15676 PL-15669 PL-15642 PL-15632 PL-15622 PL-15606 PL-15594 PL-15592 PL-15551 PL-15460 PL-15098 PL-15062 PL-15037 PL-14748 PL-14603 PL-13304 PL-13120 PL-12999 PL-11158 PL-10533

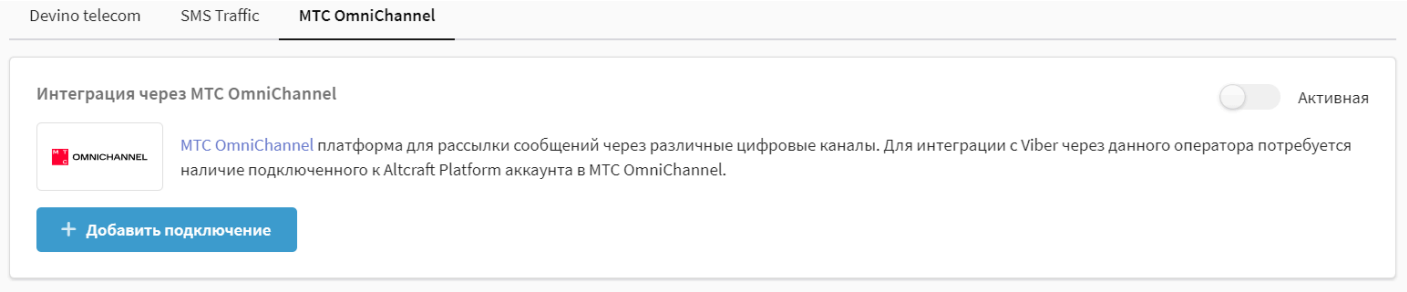
---

v2024.1.68.1354 - MTS OmniChannel integration, bug fixes

Основные изменения

Интеграция с MTC OmniChannel

Добавлена поддержка интеграции с **MTC OmniChannel**. Теперь вы можете использовать эту платформу как шлюз для отправки сообщений в Viber™.



Прежде чем настраивать интеграцию, вам необходимо связать ваш аккаунт в MTC OmniChannel с Altcraft Platform через **подключение**. Подробнее о настройке интеграции вы можете прочитать **здесь**.

Другие изменения

- Раздел “Пиксели” перенесен в пункт меню “Веб-слой”.
- Обновлены данные geoip за апрель 2024.
- Улучшена производительность платформы.
- Добавлена возможность убирать валидацию для скрытых полей в формах.

Исправления

- Устранена ошибка, связанная с дубликацией трекинг-ссылок.
- Теперь элементы сценария с условием сохраняются с корректным id.
- В замороженных аккаунтах снова действует ограничение прав на просмотр.
- Исправлена проблема, при которой ошибки валидации форм выводились в логах на error уровне.
- Проверка разрешенных доменов для перенаправления при открытии пикселя теперь работает корректно.
- Улучшена работа отчета по трафику при использовании группировки по целям пикселя.
- Исправлено некорректное поведение платформы во время удаления баз профилей в тех случаях, когда у роли пользователя нет прав на удаление.

- Улучшена работа режима критических пушей с APNS.
- Валидация при использовании сегментов со сбором данных по ссылке и при ручном создании профиля теперь работает корректно.
- История входов в платформу теперь отображает корректные данные.
- Исправлены ошибки в работе логов.
- Улучшена работа попапов при мобильном отображении.
- Интеграции в платформе теперь корректно сохраняются.
- Устранена ошибка, связанная с созданием подключений к Facebook*.
- Исправлены некоторые ошибки аналитики и отчетов.
- Для отправок на почтовые ящики Yahoo теперь используется корректный mx.
- Исправлены ошибки при работе интеграции Yandex.Appmetrica.
- События открытия и заполнения формы в истории профиля теперь отображаются корректно.
- Исправлена валидация полей в формах.
- Устранены ошибки при предпросмотре многостраничных форм.
- `click_url` в веб-версии триггерных пуш-рассылок теперь сохраняется корректно.
- Устранена паника `runtime error` при отправке кампании со скрытой копией.
- Устранена паника в `form_tst.go` .

References

PL-15888 PL-12880 PL-15993 PL-15991 PL-15865 PL-14353 PL-15869 PL-15926 PL-15840 PL-15979 PL-10455 PL-15813 PL-15970 PL-14286 PL-13837 PL-15334 PL-16037 PL-15779 PL-15939 PL-15662 PL-16032 PL-13906 PL-15800 PL-16112 PL-16001 PL-16048 PL-10320 PL-14477 PL-15907 PL-14312 PL-16013 PL-14015 PL-15104 PL-15989 PL-15712 PL-15693 PL-15700

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*
***Viber является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

Хранение и сбор данных

С чего начать?

Создайте базу данных

	Начните с создания базы данных, в которой будете хранить цифровые профили своих клиентов. Для базы определите структуру, используя стандартные поля данных и дополнительные — для реализации ваших бизнес-процессов. **Создание базы данных — первый шаг в работе с Altcraft.**

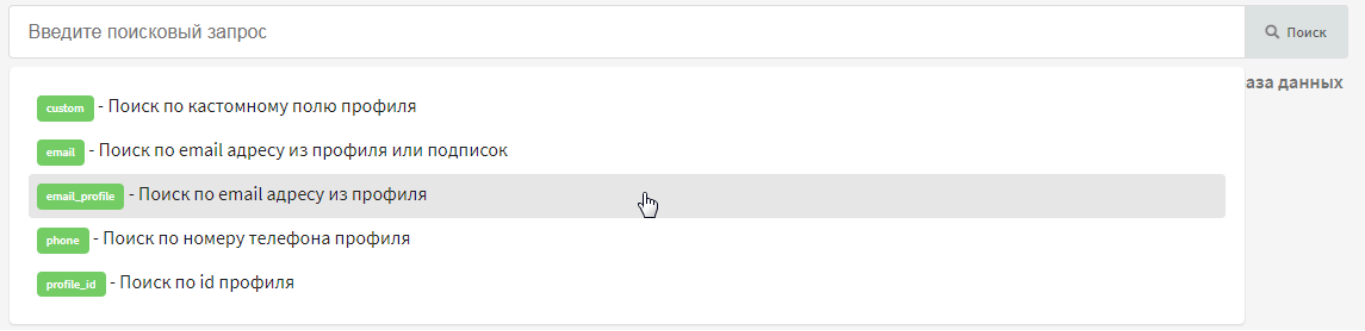
Создайте ресурс подписок

	Коммуникация с клиентами в Altcraft рассчитана на взаимодействие по нескольким каналам связи, в том числе с использованием разных контактных данных в рамках одного канала. Подписки профилей на тот или иной тип рассылок (рекламные, inbound, сервисные, партнёрские) сгруппированы в Ресурсы. Ресурс — издатель контента, для которого можно отдельно настраивать данные отправителя, трекинг, политики отправки, а также используемые каналы коммуникации. **Чтобы начать общение с клиентами, создайте как минимум один ресурс.**

INFO

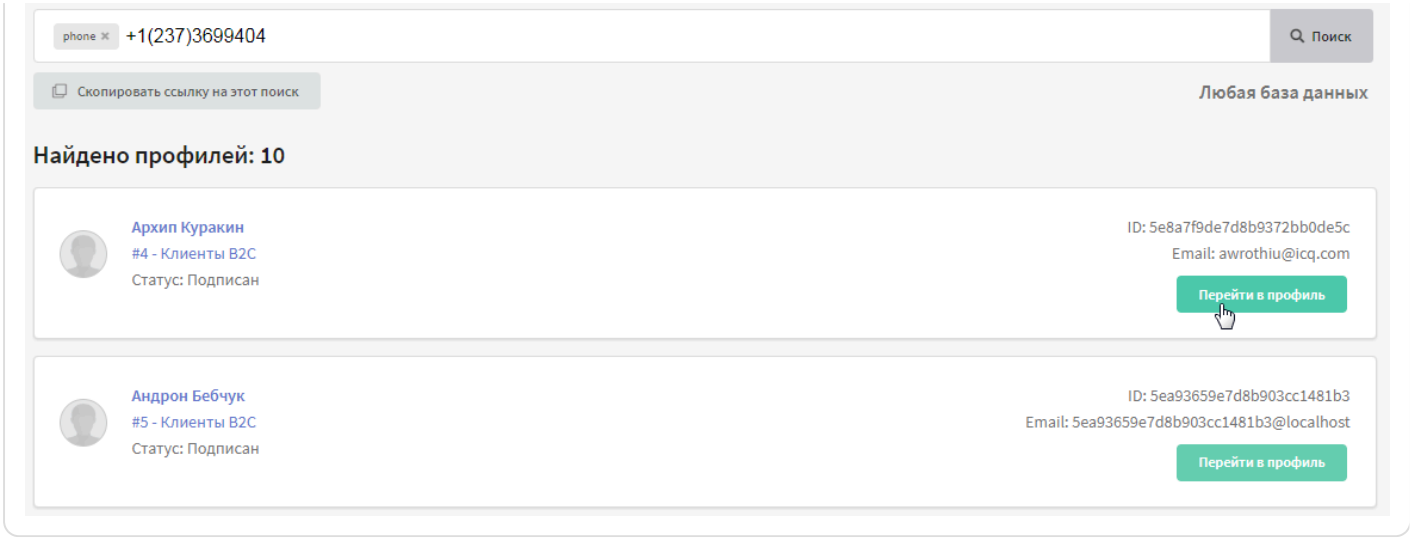
Чтобы быстро найти профиль нужного клиента, используйте **Поиск** в разделе **“Данные”**. Скопируйте или введите данные клиента — система автоматически определит режим поиска.

Для Email вы можете указать поиск по подпискам или только по данным профиля.



Для поиска профиля по дополнительному (кастомному) полю используйте следующий формат: `название_поля:значение`.

Вы можете открыть найденные профили и **скопировать ссылку** на поисковый запрос:


phone ×
+1(237)3699404
Поиск
Скопировать ссылку на этот поиск
Любая база данных
Найдено профилей: 10
Архип Куракин
#4 - Клиенты B2C
Статус: Подписан
ID: 5e8a7f9de7d8b9372bb0de5c
Email: awrothiu@icq.com
Перейти в профиль
Андрон Бебчук
#5 - Клиенты B2C
Статус: Подписан
ID: 5ea93659e7d8b903cc1481b3
Email: 5ea93659e7d8b903cc1481b3@localhost
Перейти в профиль

Ресурсы подписок

Ресурс — это средство для организации подписок клиентов на определённый контент по определённым каналам связи, иными словами — **издатель**. Подписки на разные ресурсы не связаны: прекращение подписки на один ресурс не влияет на другие подписки и глобальный статус профиля, а значит подписчик может в личном кабинете выбрать, какой именно контент получать.

В ресурсе может быть настроен один или несколько каналов связи для формирования подписок. Ресурсом может быть:

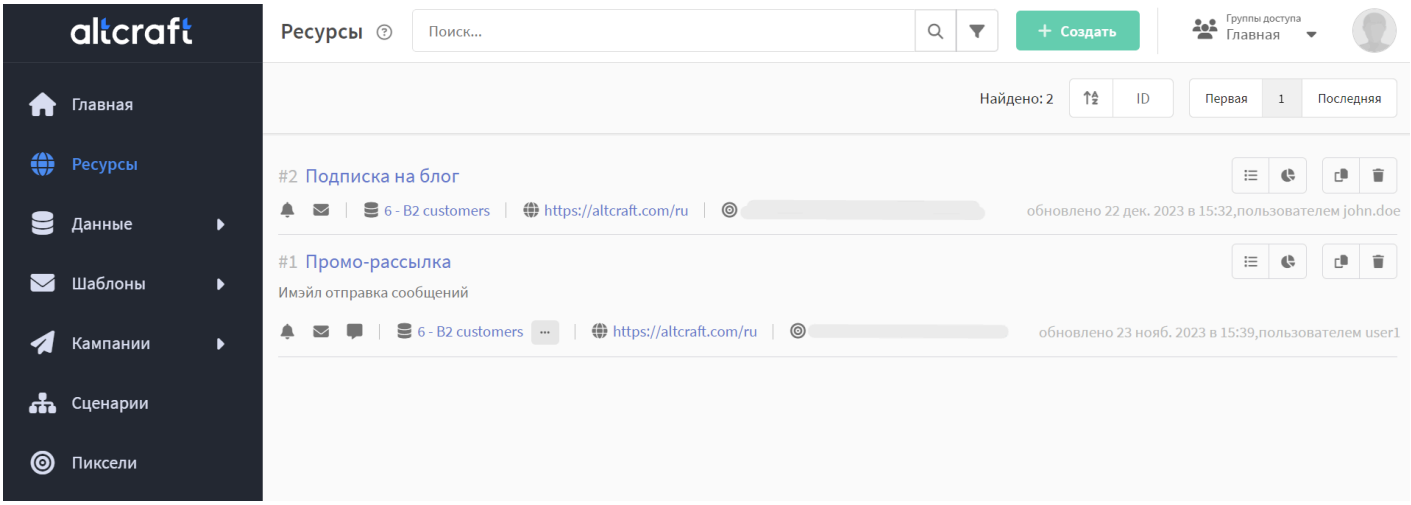
- сайт или отдельный лендинг,
- мобильное или десктопное приложение,
- форма регистрации на сайте партнёра (и сама партнёрская программа),
- определённый тип рассылок (промо-акции, новости, полезные советы),
- конкретная акция или продукт.

Например, если подписчика не устраивают рекламные рассылки, но интересуют информационные, то в случае использования двух ресурсов шанс сохранения потенциального клиента и заключения сделки существенно выше.

Разные ресурсы могут использовать разные домены трекинга и отправщика, имена отправителя. В “Отчётах по каналам” доступна общая группировка статистики по ресурсам, что позволяет проводить эффективный сквозной анализ разных направлений вашего маркетинга.

Управление ресурсами

Чтобы перейти к управлению ресурсами, в главном меню выберите раздел **Ресурсы**.



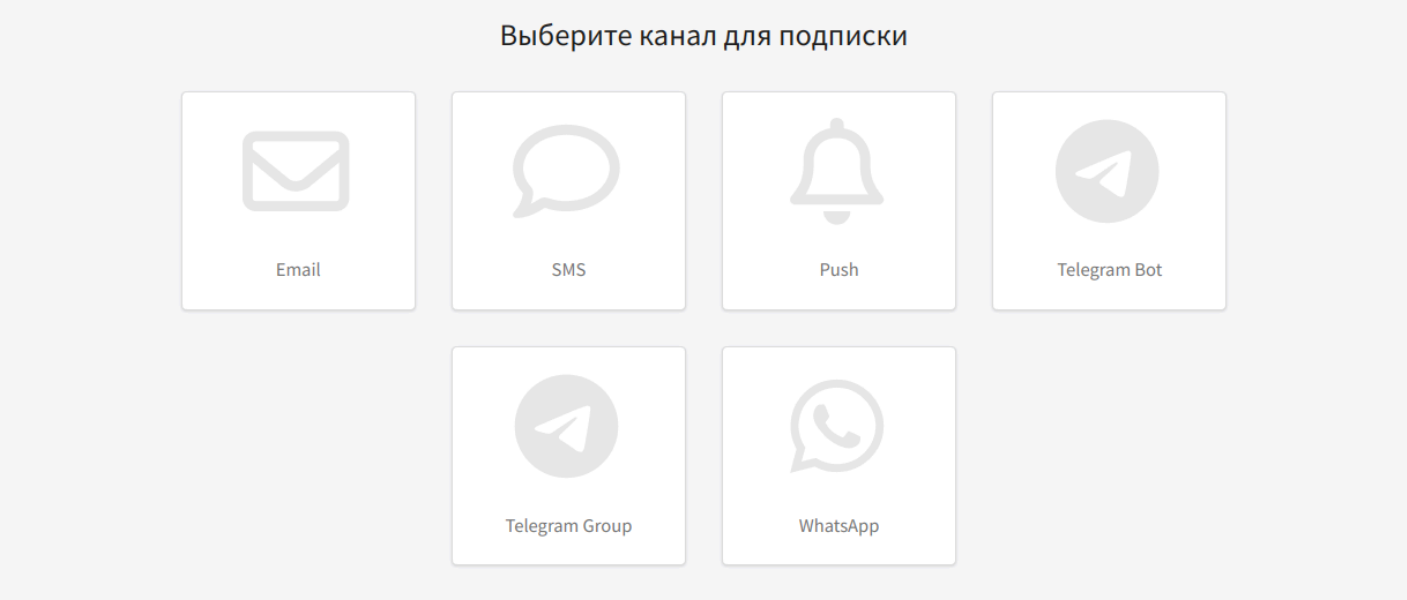
В открывшемся списке отображаются существующие ресурсы. Чтобы перейти к редактированию, нажмите на название ресурса. Из меню действий в правой части можно просмотреть подписчиков ресурса

, клонировать ресурс

или удалить ресурс

. Чтобы добавить новый ресурс, нажмите **+ Создать** в верхней части страницы.

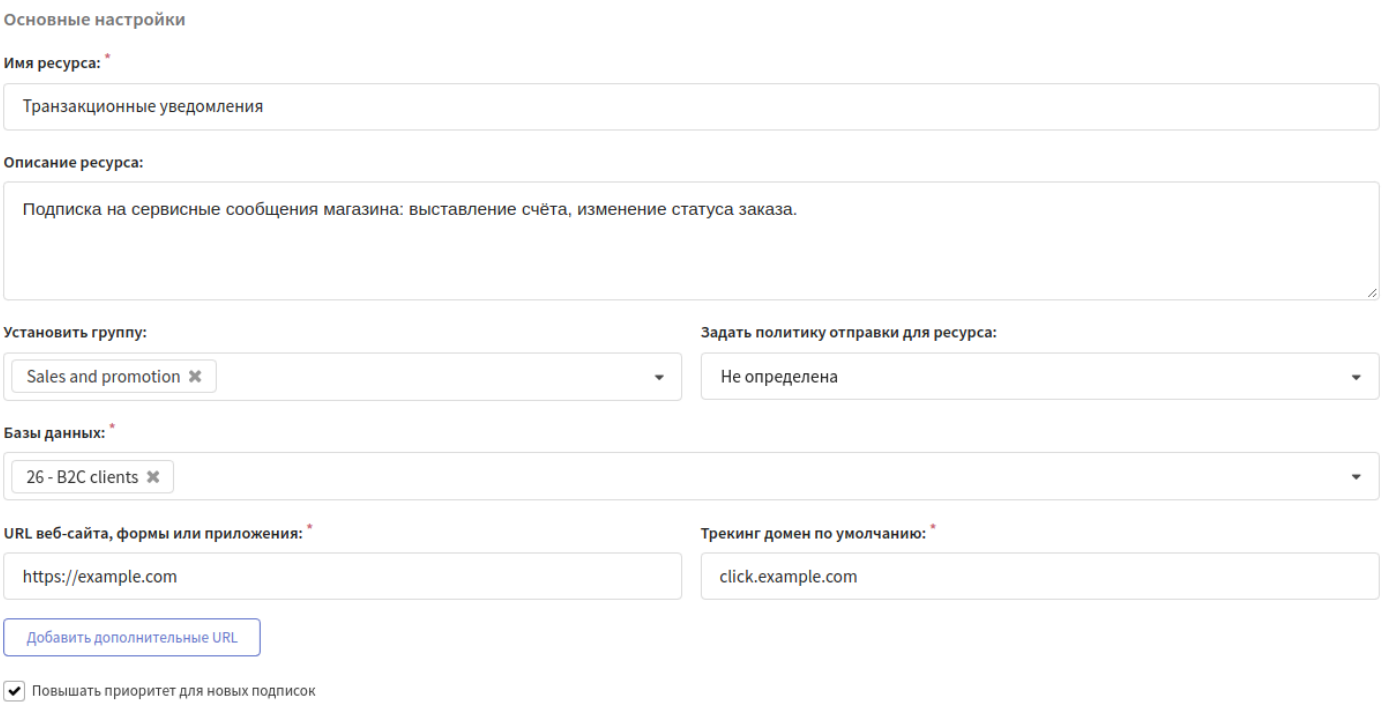
Система предложит выбрать основной канал связи, по которому клиенты будут получать сообщения:



Основные настройки

Для нового ресурса укажите название и опционально — описание. Для каждого ресурса настраивается:

- **Группа доступа** — поместите ресурс в группу объектов аккаунта.
- **Политика отправки для ресурса** — ограничить коммуникацию с одним подписчиком в рамках ресурса.
- **Базы данных** — выбрать, профили каких баз данных могут подписываться на ресурс.
- **URL вебсайта, формы или приложения** — URL источника подписчиков ресурса. Можно добавить дополнительные URL.
- **Трекинг-домен по умолчанию** — домен для отслеживания действий подписчика с отправляемыми сообщениями. В Smart настройках рассылки можно изменить эту настройку и добавить несколько трекинг-доменов.



TIP: Повышать приоритет подписки

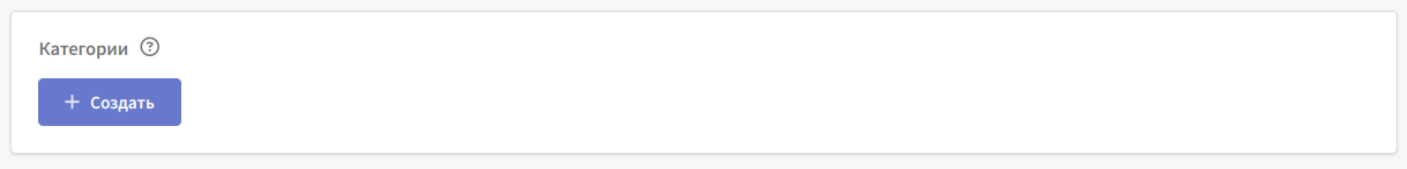
Пользователь может подписаться на один ресурс несколько раз, указав разные контактные данные (email или номер телефона). Если вы активируете опцию **“Повышать приоритет для новых подписок”**, то контактные данные, которые пользователь указал при последней подписке, станут приоритетными. Именно на этот email или номер будут отправляться сообщения в последующих рассылках.

Обратите внимание, опция “Повышать приоритет для новых подписок” работает только в случае, если при добавлении новой подписки явно не указан приоритет. Если приоритет новой подписки указывается явно (например, в API-запросе), то приоритеты существующих подписок не изменяются.

Категории ресурса

Категории позволяют детализировать тип контента, на который подписан профиль. Например, ресурс “Мобильное приложение” может содержать такие категории, как “Новости”, “Промо-предложения” и “Транзакции”. Если клиент отпишется от одной категории ресурса, он продолжит получать сообщения в рамках других категорий. В отличие от подписки на ресурс, подписка на категорию имеет только 2 статуса: “Подписан” и “Отписан”. Изменять статус вы можете вручную в профиле клиента или через API-запросы.

Чтобы добавить новую категорию в ресурс, нажмите кнопку **+ Создать**:



Укажите произвольное **имя категории** и **имя**, которое будет использоваться в **API-запросах**:

Дополнительно вы можете добавить категорию в менеджер подписок, чтобы у подписчика была возможность самостоятельно управлять подписками на категории. Если получение сообщений в рамках этой категории является обязательным, активируйте опцию **“Подписчик не может отписаться от категории в менеджере подписок”**.

Количество категорий не ограничено. Для удаление категории нажмите кнопку

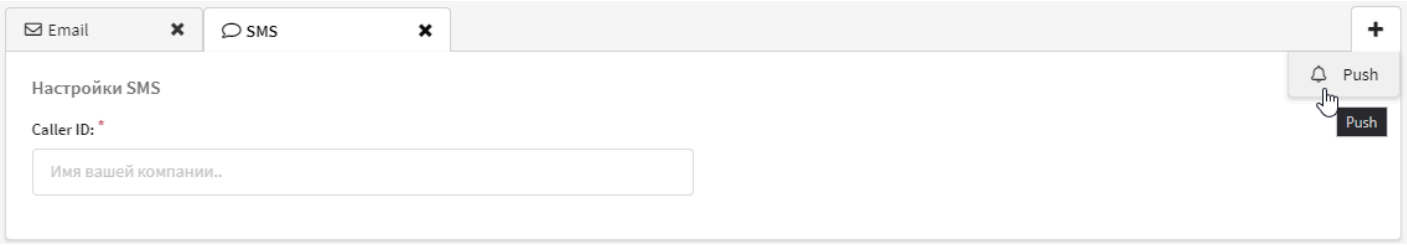
.

Обратите внимание, что после сохранения ресурса вы сможете отредактировать имя категории, но не API имя.

Настройки каналов связи

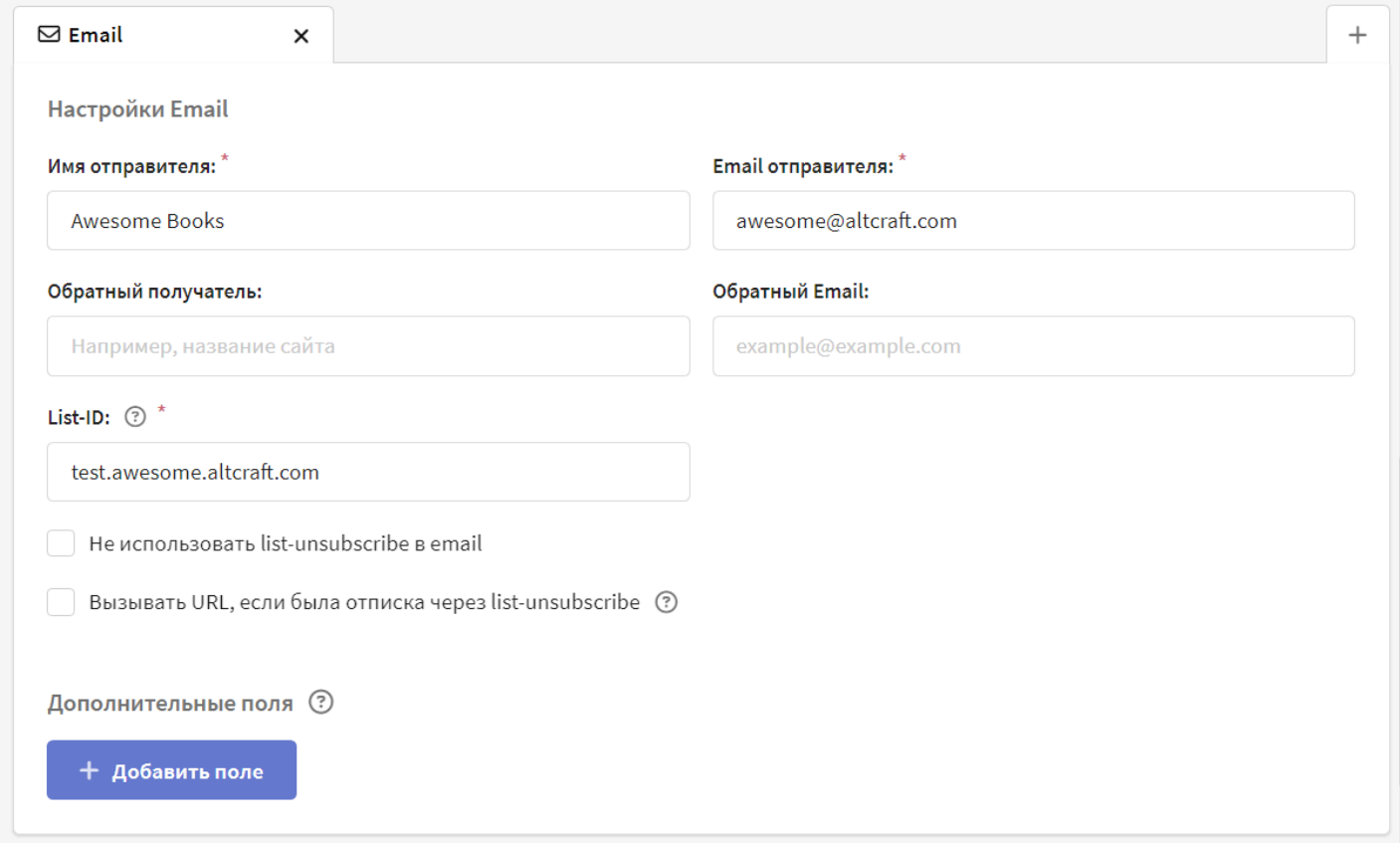
> **INFO**
>
> Для одного ресурса можно добавить **не более 8 каналов коммуникации!**

Каждый ресурс может формировать подписки на один или несколько каналов коммуникации. По умолчанию доступно 7 каналов: Email, SMS, Push, Telegram, WhatsApp*, Viber™ и Notify. Пользовательские каналы предварительно настраиваются администраторами платформы, и после этого становятся доступны для выбора. Чтобы добавить в ресурс новый канал, нажмите на **плюс** и выберите его из выпадающего меню:



Email канал

Для коммуникации с клиентами по электронной почте активируйте в ресурсе **Email канал**, введите **имя и адрес отправителя** (From name и From email), а также **имя и адрес для получения ответов** (Reply-to name и Reply-to email) — если рассылки по этому ресурсу предполагают ответы подписчиков.



Также необходимо заполнить поле **“List-ID”**. List-ID — это уникальный идентификатор списка рассылки, который передается в заголовке email-письма. Почтовые клиенты используют информацию из заголовка для фильтрации и сортировки входящих сообщений. Подробности стандарта List-ID можно найти по ссылке.

Введите List-Id в формате `[list-label].[domain]`. Примеры:

- list-header.nisto.com
- commonspace-users.list-id.within.com
- 0Jks9449.list-id.cmu.edu

List-ID указывается без символов `<` и `>`. Длина идентификатора не должна превышать 255 символов. Домен должен совпадать с доменом отправителя.

INFO

Рекомендуется использовать разные List-ID для каждого ресурса.

Настройки отписки в Email-канале

Платформа предоставляет два механизма отписки от Email-канала в ресурсе:

1. **Ссылка отписки в List-Unsubscribe** — передается в заголовке письма для использования почтовыми сервисами.
2. **Ссылка отписки в теле письма** — добавляется в шаблон с помощью переменной `{unsubscribe}`.

Для управления отпиской через **List-unsubscribe** доступны следующие настройки:

- **Не использовать list-unsubscribe в email** — ссылка отписки list-unsubscribe не добавляется в параметр заголовка.
- **Вызывать URL, если была отписка через list-unsubscribe** — отобразится обязательное поле **“Вызываемый URL”**. В нем следует указать URL, на который перейдет пользователь после отписки через ссылку отписки почтового сервиса. Если URL не указывать, то пользователь перейдет на системную страницу отписки.

CAUTION

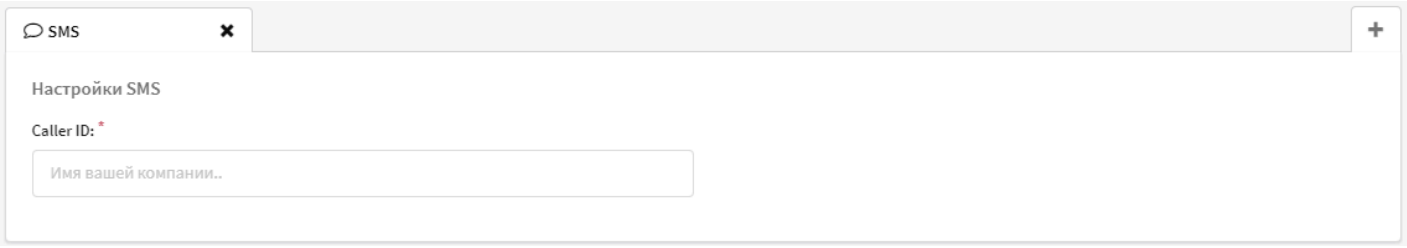
Опция отписки **“Не использовать list-unsubscribe в email”** доступна только для пользователей On-Premise решения. Для активации этой опции необходимо добавить параметр `"unconditional_subscriber"` в свойство `EXTRAS` в файл конфигурации `main.json`.

TIP

Для коммуникации с подписчиками через email-канал используется email-адреса в формате `john@example.com`. Email-адрес должен соответствовать стандартам RFC. Основные требования вы найдете по этой ссылке.

SMS-канал

Для коммуникации с клиентами по SMS активируйте в ресурсе **SMS-канал** и введите **имя отправителя** (Caller ID). Обычно его нужно согласовать с SMS провайдером, иначе он использует своё собственное имя.



TIP

Для коммуникации с подписчиками через sms-канал используются номера телефонов. Номер должен быть указан в международном формате (без пробелов, дефисов и других разделителей, кроме плюса в начале). Пример: `+79123456789`

Push-канал

Push-канал объединяет настройки push уведомлений для всех поддерживаемых платформ.

Веб-браузеры: Google Chrome, Mozilla Firefox, Apple safari и Opera browser.

Мобильные приложения: Android и iOS.

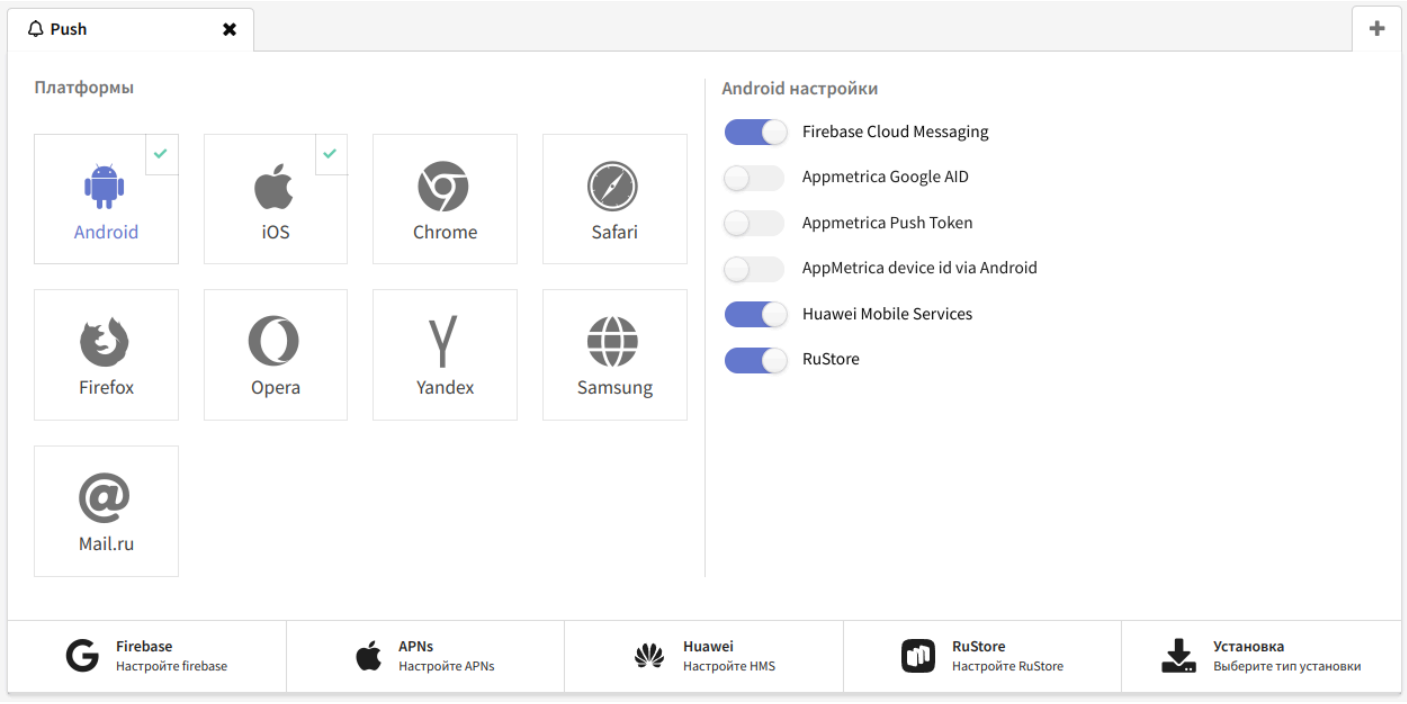
Для отправки уведомлений доступны следующие провайдеры:

- **Google Firebase Cloud Messaging** — все платформы за исключением Safari.
- **Apple Push Notification Service** — только Safari и iOS.
- **Mozilla Services** — только Mozilla Firefox.
- **Yandex AppMetrica** (доступна автоподписка при интеграции с AppMetrica) — только Android и iOS. Для отправки уведомлений AppMetrica используется Google Firebase SDK.
- **Huawei Mobile Services** — Android и iOS.
- **RuStore** — только Android.

Для работы с push уведомлениями активируйте в ресурсе **Push-канал** и выберите необходимые браузеры, платформы и провайдеры.

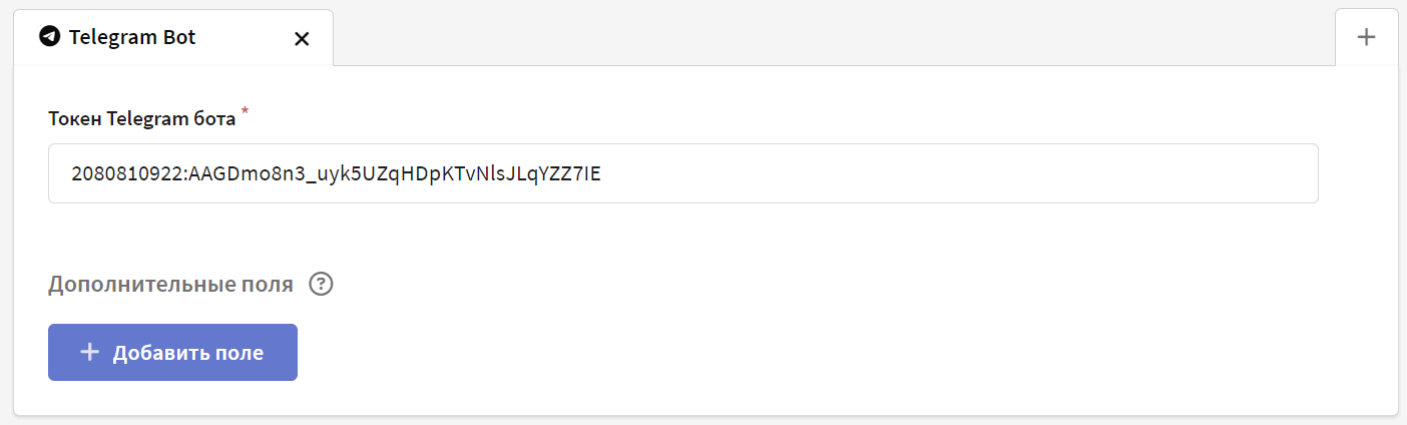
TIP

Настройка push уведомлений в зависимости от провайдера подробно описана в этой статье.



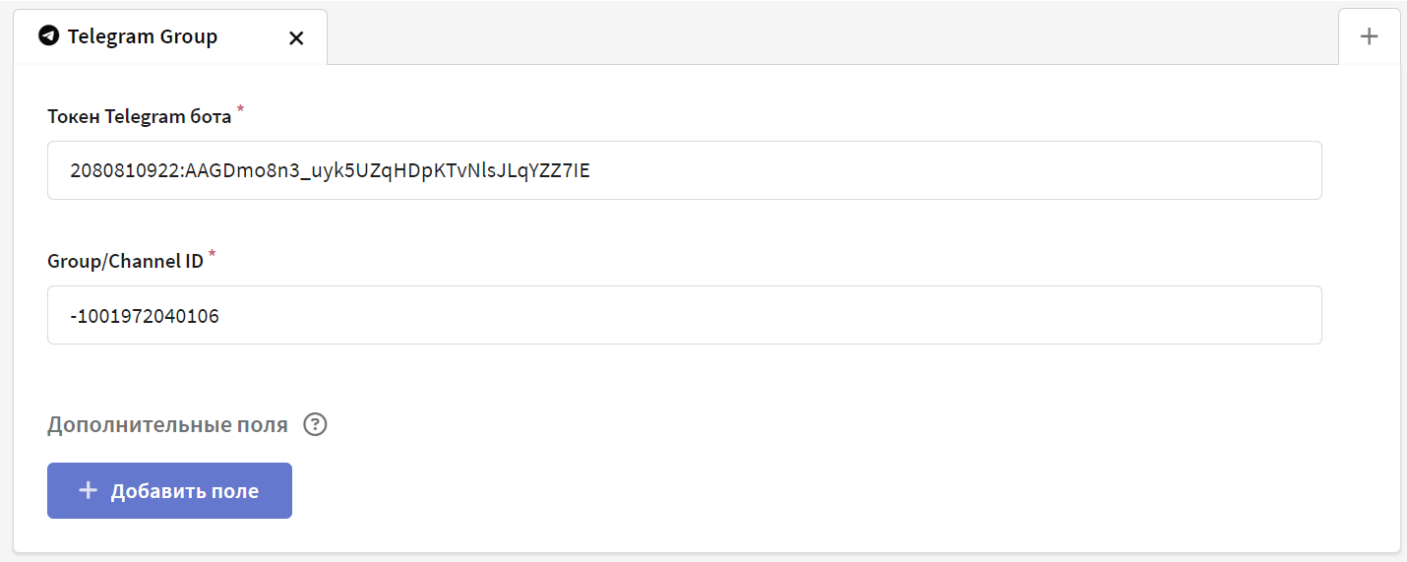
Telegram Bot канал

Используйте этот канал для отправки подписчикам сообщений через Telegram-бота. Необходимо указать токен бота. О том, как получить токен, читайте в этом гайде.



Telegram Group канал

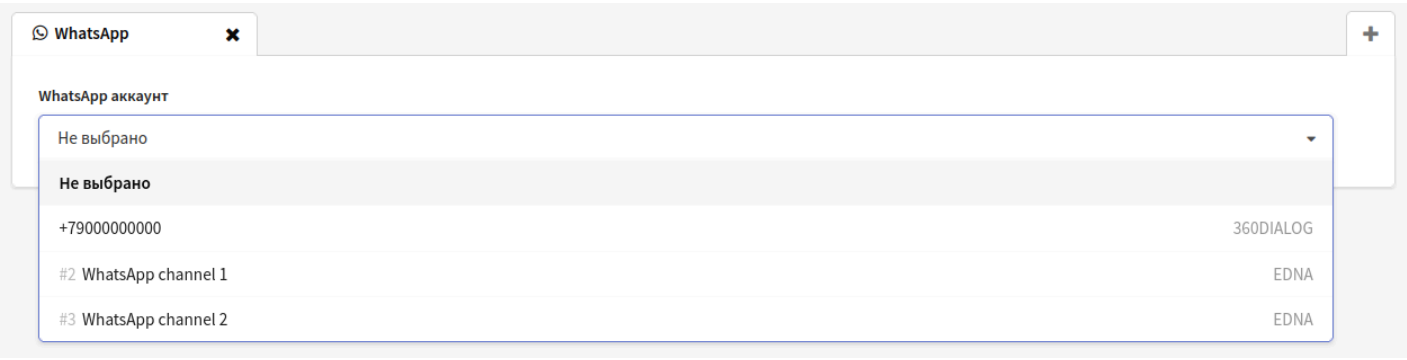
Сообщения в этот канал можно отправлять с помощью рассылок размещения. Укажите токен бота и идентификатор группы или канала. Больше информации о настройке канала вы найдете здесь.



WhatsApp-канал

Чтобы отправлять подписчикам сообщения в мессенджер, необходимо настроить интеграцию с WhatsApp* Business API. Интеграция осуществляется через одного из провайдеров: Edna или 360dialog.

После настройки интеграции выберите подключенный аккаунт (360dialog) или канал (Edna) в выпадающем списке:



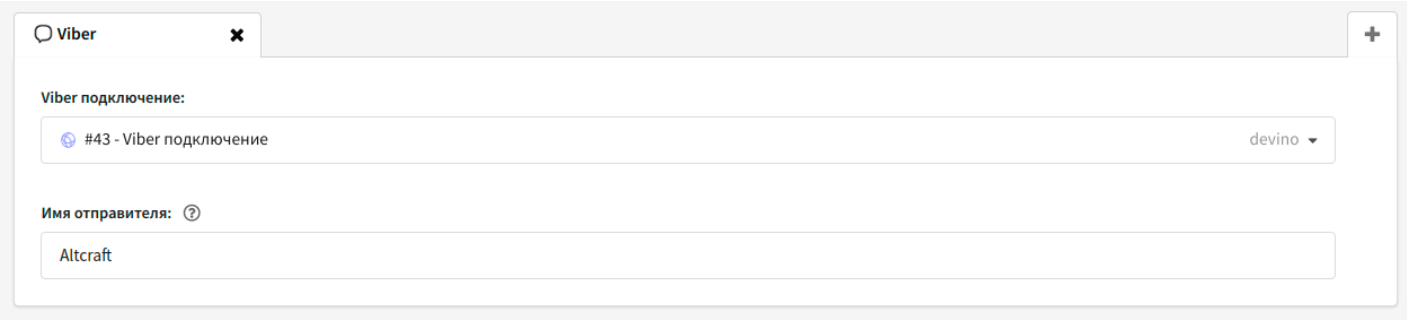
TIP

Для коммуникации с подписчиками через WhatsApp*-канал используются номера телефонов. Номер должен быть указан в международном формате (без пробелов, дефисов и других разделителей, кроме плюса в начале). Пример: `+79123456789`

Viber™-канал

Для отправки сообщений в Viber™ необходимо настроить интеграцию. Интеграция осуществляется через провайдеров Devino Telecom, SMS Traffic, MTS OmniChannel или Rapporto.

В настройках канала выберите созданное раннее Viber™-подключение и укажите **имя отправителя**. Это имя увидят подписчики при получении сообщения. Имя отправителя можно найти в личном кабинете сервиса-провайдера; оно должно быть с ним предварительно согласовано.



INFO

Для настройки коллбэков со статусами сообщений обратитесь к менеджеру компании или в техническую поддержку вашего провайдра: Devino Telecom, SMStraffic, MTC OmniChannel или Rapporto.

TIP

Для коммуникации с подписчиками через Viber™-канал используются номера телефонов. Номер должен быть указан в международном формате (без пробелов, дефисов и других разделителей, кроме плюса в начале). Пример: `+79123456789`

Notify-канал

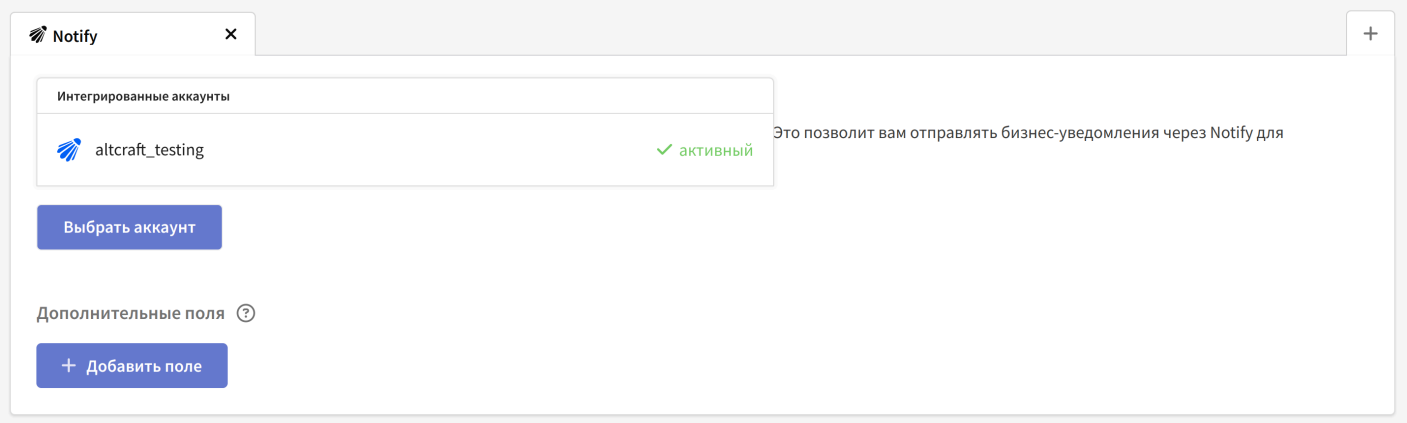
INFO

Согласно политике Notify, в данный момент новые доступы к личному кабинету сервиса **не предоставляются.**

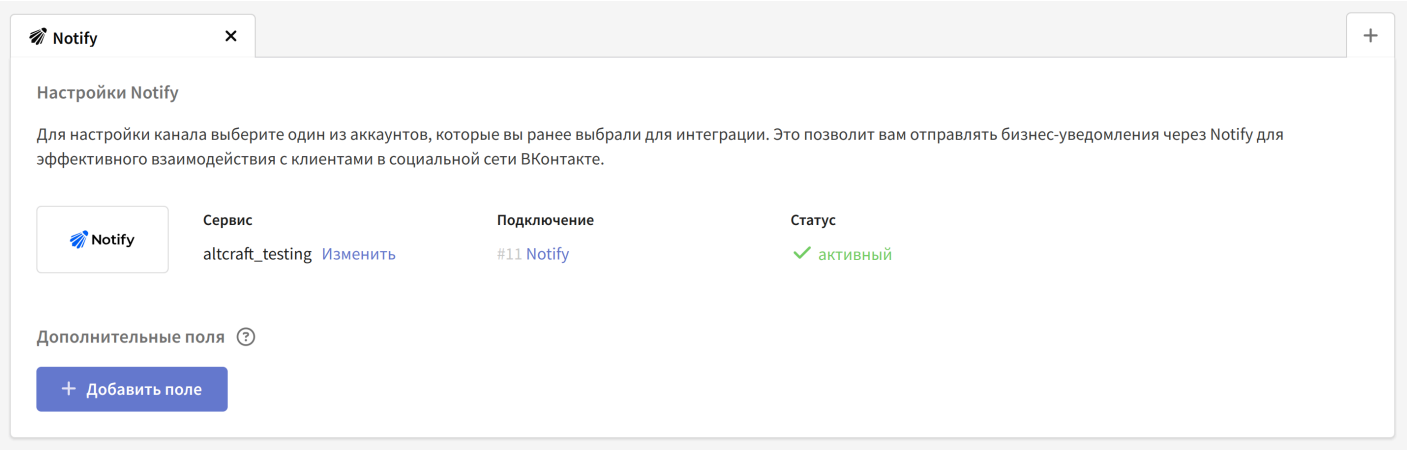
Поддержка текущих пользователей доступна через техподдержку MAX Business.

Сервис Notify позволяет рассылать бизнес-уведомления через VK и OK для эффективной коммуникации с клиентами. Для использования Notify в платформе необходимо предварительно настроить подключение и интеграцию сервиса.

В настройках канала выберите интегрированный раннее аккаунт Notify:



Проверьте, активен ли аккаунт, и сохраните ресурс:

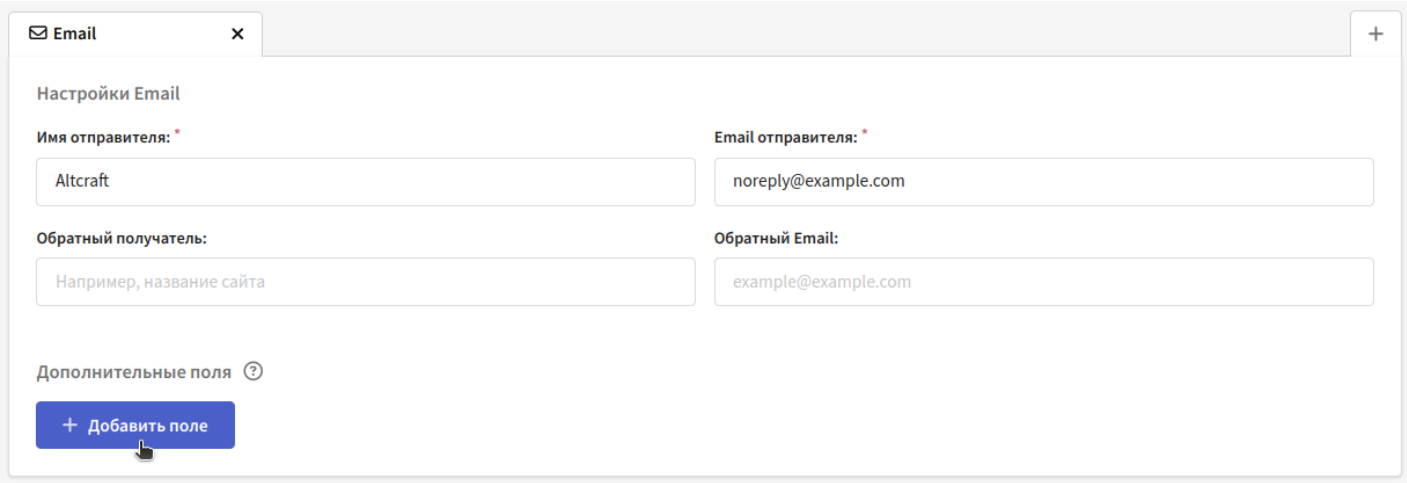


Дополнительные поля в каналах подписок

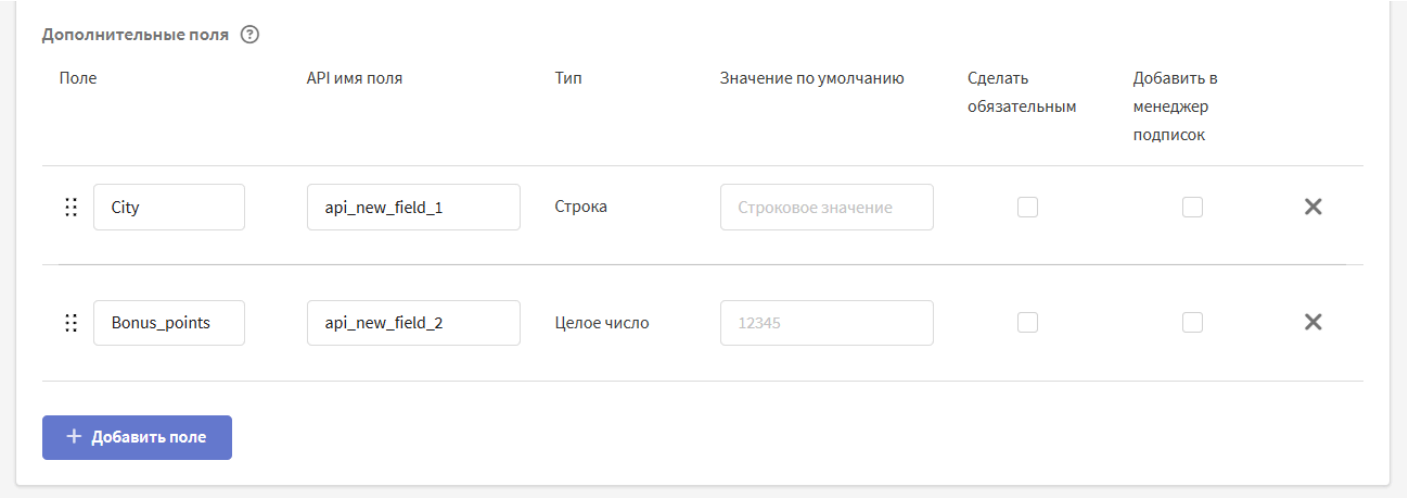
В любой канал ресурса можно добавить дополнительные поля. Они аналогичны дополнительным полям в базе профилей. Используйте их для хранения дополнительной информации по подписке на канал.

Чтобы создать поле, нажмите кнопку **“Добавить поле”** и выберите его тип:

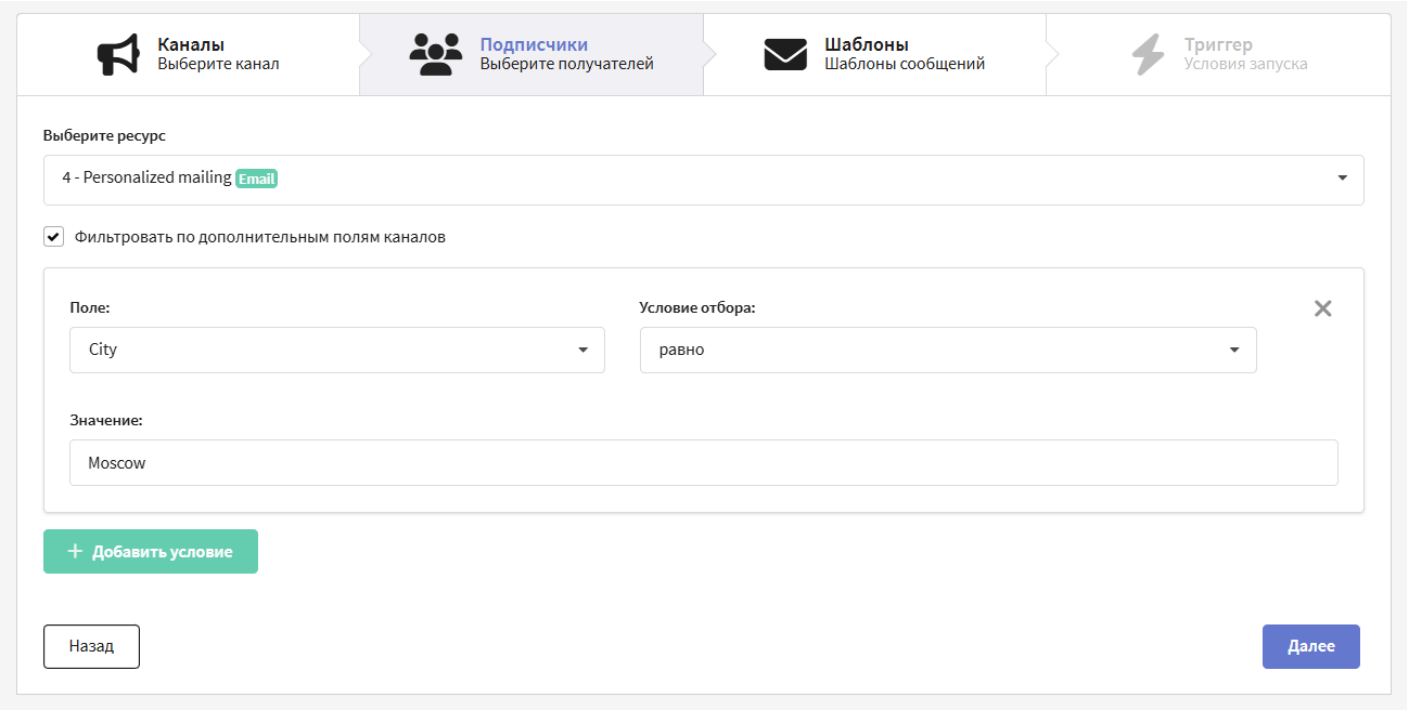
- **Целое число** — положительные и отрицательные целые числа.
- **Число с точкой** — положительные и отрицательные числа с плавающей точкой.
- **Строка** — текст на любом языке, включая цифры, специальные символы, пробелы. Используется стандарт Unicode.
- **Boolean** — поле, которое предоставляет возможность выбора из двух вариантов True или False (Да или Нет).



Далее укажите **имя поля** и **имя**, которое будет использоваться в **API-запросах**. В имени допускается использование любых символов. В API-имени можно использовать только латинские символы, цифры и символ подчеркивания; также оно должно начинаться с буквы.



После создания полей вы можете использовать их для настройки триггерной рассылки. Это позволит сделать её более персонализированной.



Дополнительные настройки полей:

- **Значение по умолчанию** — указанное значение будет установлено в поле подписки, если не было передано других данных для этого поля.
- **Сделать обязательным** — данные для этого поля необходимо будет указывать при добавлении подписки профилю.
- **Добавить в менеджер подписок** — поле появится на странице менеджера подписок.

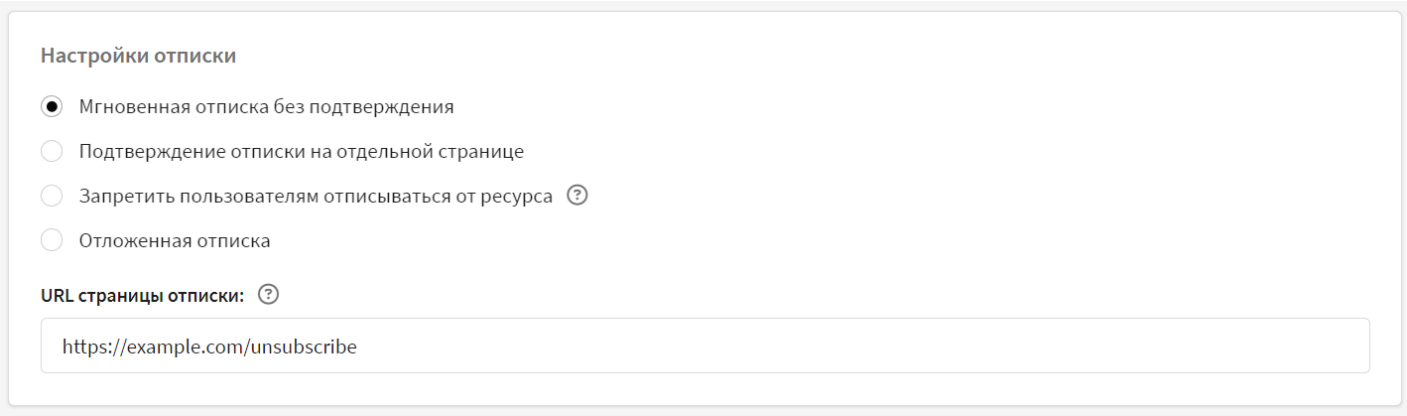
Чтобы удалить поле, нажмите крестик. Количество дополнительных полей не ограничено. После сохранения ресурса вы сможете редактировать имя поля и его дополнительные настройки, однако изменение API-имени будет невозможно.

Настройка отписки

Для каждого ресурса можно настроить отдельный механизм отписки. URL страницы отписки, указанный в ресурсе, переопределит системную страницу отписки. Однако если в шаблоне письма указана пользовательская ссылка на отписку, то она будет приоритетной. Подробнее о пользовательских ссылках.

Мгновенная отписка без подтверждения

При клике на ссылку отписки профиль переходит на системную страницу отписки и его статус сразу же меняется на “Отписан”. Данный тип отписки используется по умолчанию.

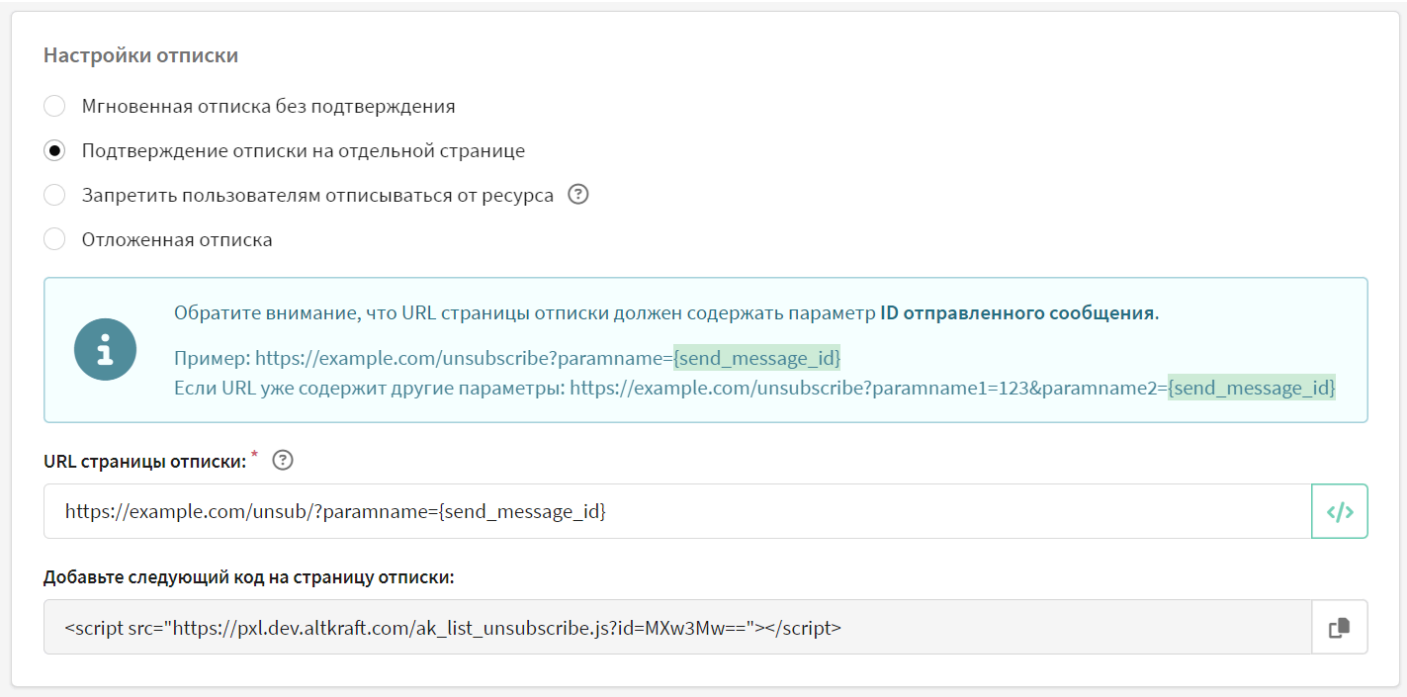


Вы можете перенаправлять профиль на собственную страницу отписки, указав другой URL-адрес в поле **“URL страницы отписки”**.

Подтверждение отписки на отдельной странице

> **CAUTION**
>
> Эта настройка отписки доступна только для пользователей On-Premise решения. Для активации этих настроек необходимо добавить параметр `"unconditional_subscriber"` в свойство `EXTRAS` в файл конфигурации `main.json`.

При клике на ссылку отписки в сообщении профиль перейдет на пользовательскую страницу отписки. Его статус изменится на “Отписан” только после этого, как он подтвердит отписку. Такая страница может использоваться для уточнения причины отписки.



Настройка полей:

Укажите **“URL страницы отписки”** — этот URL переопределит URL-адрес системной страницы отписки.

В URL страницы отписки добавьте параметр **ID отправленного сообщения** `{send_message_id}`. Это обязательный параметр, который позволит идентифицировать ресурс и профиль, для которого требуется осуществить отписку.

Выберите механизм отписки:

Механизм отписки с использованием Altcraft SDK

> **CAUTION**
>
> Этот механизм реализуется **на стороне клиента** (в браузере). Безопасен для размещения на странице отписки, доступной пользователю. Все операции выполняются через SDK, без передачи токенов или других чувствительных данных в открытом виде.

Вставьте предоставленный в ресурсе код на вашу страницу отписки для подключения Altcraft SDK. Далее инициализируйте класс `AkUnsub`. Пример:

```
<!-- Подключение Altcraft SDK -->
<script src="https://pxl.altkraft.com/ak_list_unsubscribe.js?id=MXw3Mw=="></script>

<!--  Инициализация AkUnsub класса -->
<script>
  const akUnsub = new AkUnsub();
</script>
```

> **INFO**
>
> При необходимости вы можете переопределить токен ресурса и пиксель домен в скрипте:
>
> ```
> new AkUnsub({
>   rtoken: "example_token",
>   pxl_domain: "your_pxl_domain",
> });
> ```

Класс AkUnsub имеет 2 публичных метода:

- `confirmUnsub` — метод для подтверждения отписки
- `cancelUnsub` — метод для отмены отписки. Используется только **для отложенной отписки**, подробнее о ней можно узнать в следующем пункте статьи.

Каждый из методов принимает следующие **аргументы:**

- **SMID** — идентификатор отправленного сообщения (обязательный параметр).
- **Массив категорий** ресурса, от которых происходит отписка (необязательный параметр).

Пример:

```
AkUnsub.confirmUnsub("SMID", ["cat_to_unsub"]);
AkUnsub.cancelUnsub("SMID", ["cat_to_unsub"]);
```

Для вызова методов на страницу отписки можно добавить кнопки. Пример:

```
<!-- Кнопка для подтверждения отписки -->
<button onclick="akUnsub.confirmUnsub('SMID',['category_1', 'category_2'])">
  Подтвердить отписку
</button>

<!-- Кнопка для отмены отписки -->
<button onclick="akUnsub.cancelUnsub('SMID',['category_1', 'category_2'])">Отменить
отписку</button>
```

Замените `'SMID'` и `['category1', 'category2']` на соответствующие значения.

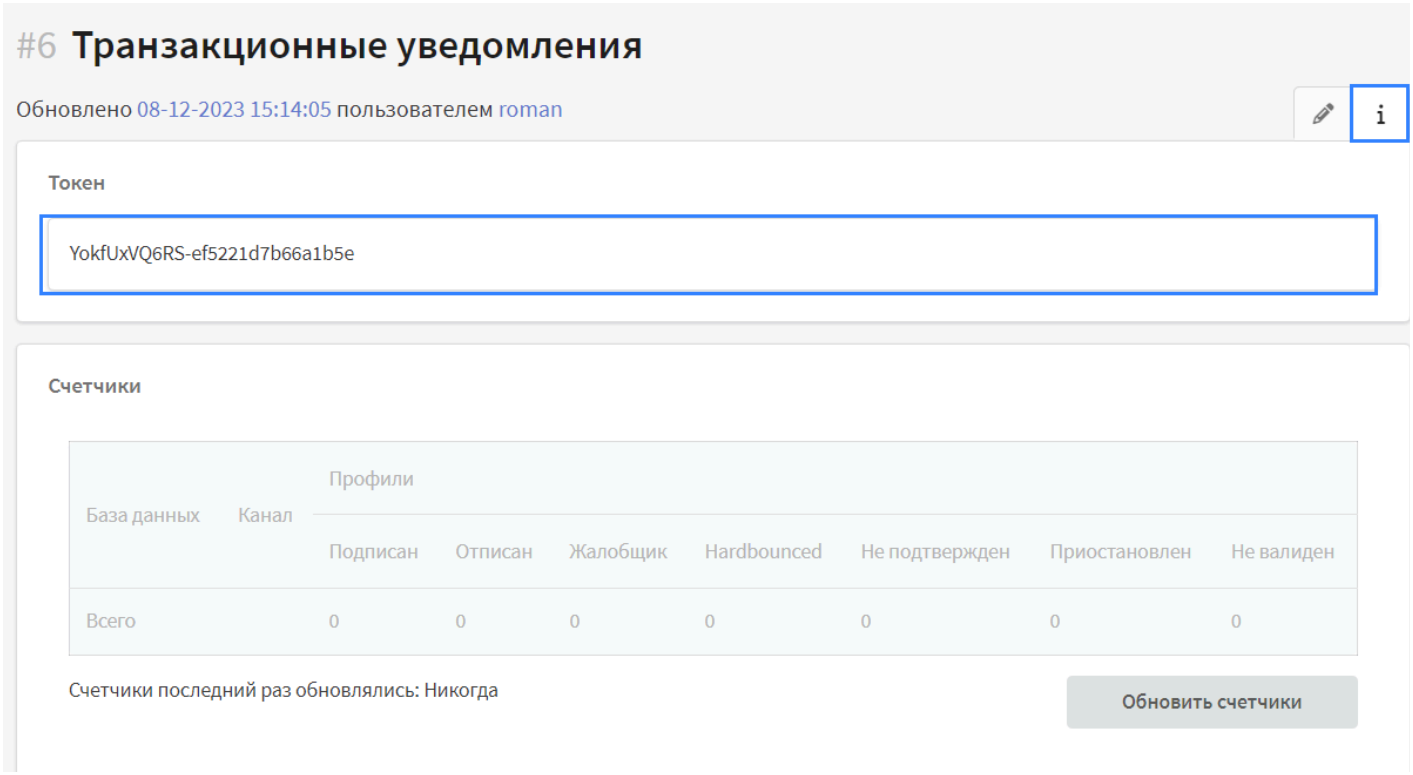
Механизм отписки без использования Altcraft SDK

CAUTION

Этот механизм реализуется **на стороне сервера**. Использует API-токен и токен ресурса, которые являются конфиденциальными. Их нельзя размещать в коде страницы или отправлять напрямую из браузера.

Вы можете реализовать механизм отписки на своей стороне. Для этого в **URL страницы отписки** необходимо добавить 2 параметра:

- **ID отправленного сообщения** `{send_message_id}`. Это обязательный параметр, позволяющий идентифицировать ресурс, от которого требуется осуществить отписку.
- **Токен ресурса** `{resource_token}`. Этот параметр требуется для дополнительной защиты и для реализации собственного механизма отписки без использования Altcraft SDK. Токен ресурса можно передавать не в URL страницы отписки, а в теле API-запроса. Токен можно получить в окне редактирования ресурса на вкладке “i” — детали:



Если профиль своими действиями подтвердил отписку на вашей странице, то необходимо передать эти данные в платформу с помощью API запроса `/api/v1.1/profiles/message_unsub`. Статус подписки профиля на ресурс изменится на “Отписан”. Если передавать параметр `categories`, то профиль будет отписан только от указанных категорий ресурса.

Пример запроса:

```
{
  "token": "abcdefghijklmnqrstuvwxyz", // API-токен
  "smid": "fGmXiWU9Y4NSIpCwl4QG6QXAAAQE", // ID отправленного сообщения
  "rtoken": "2fHLS9YR9daS-a90b371eecc2952f", // Токен ресурса
  "categories": ["cat_to_unsub"] // Категории ресурса, от которых отписать профиль
}
```

TIP

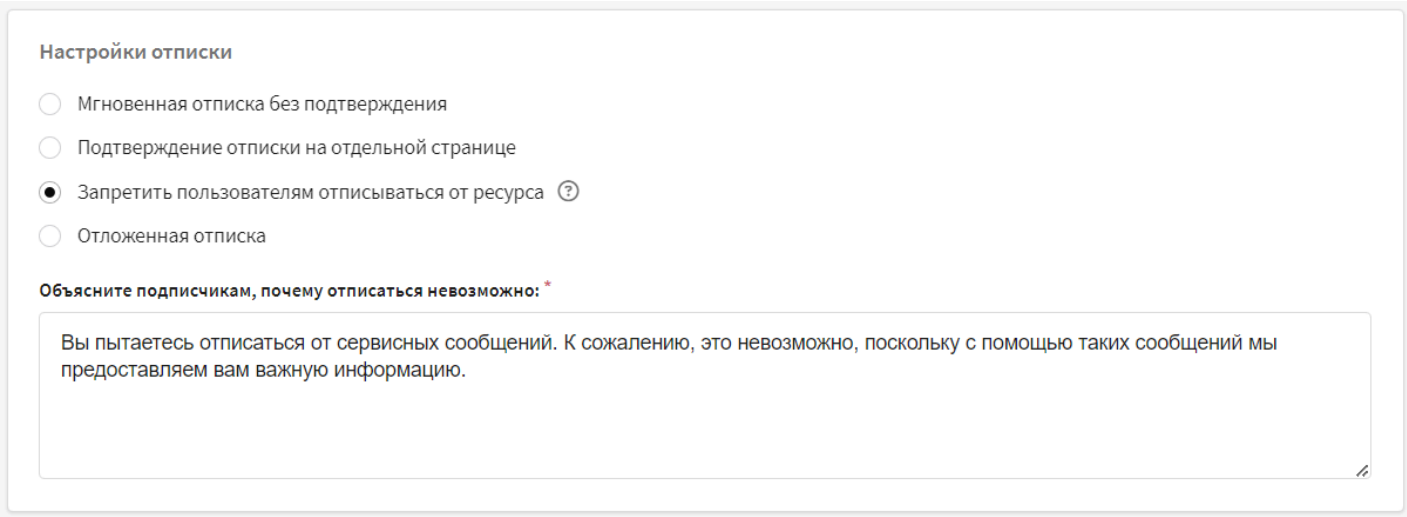
API запрос **/api/v1.1/profiles/message_unsub** описан в **документации**.

Запретить пользователям отписываться от ресурса

CAUTION

Эта настройка отписки доступна только для пользователей On-Premise решения. Для активации этих настроек необходимо добавить параметр `"unconditional_subscriber"` в свойство `EXTRAS` в файл конфигурации `main.json`.

Для определенных типов рассылок можно установить ограничение на отписку. При выборе данного параметра все ссылки отписки, указанные в шаблоне, будут вести на страницу с причиной, по которой нельзя отписаться.



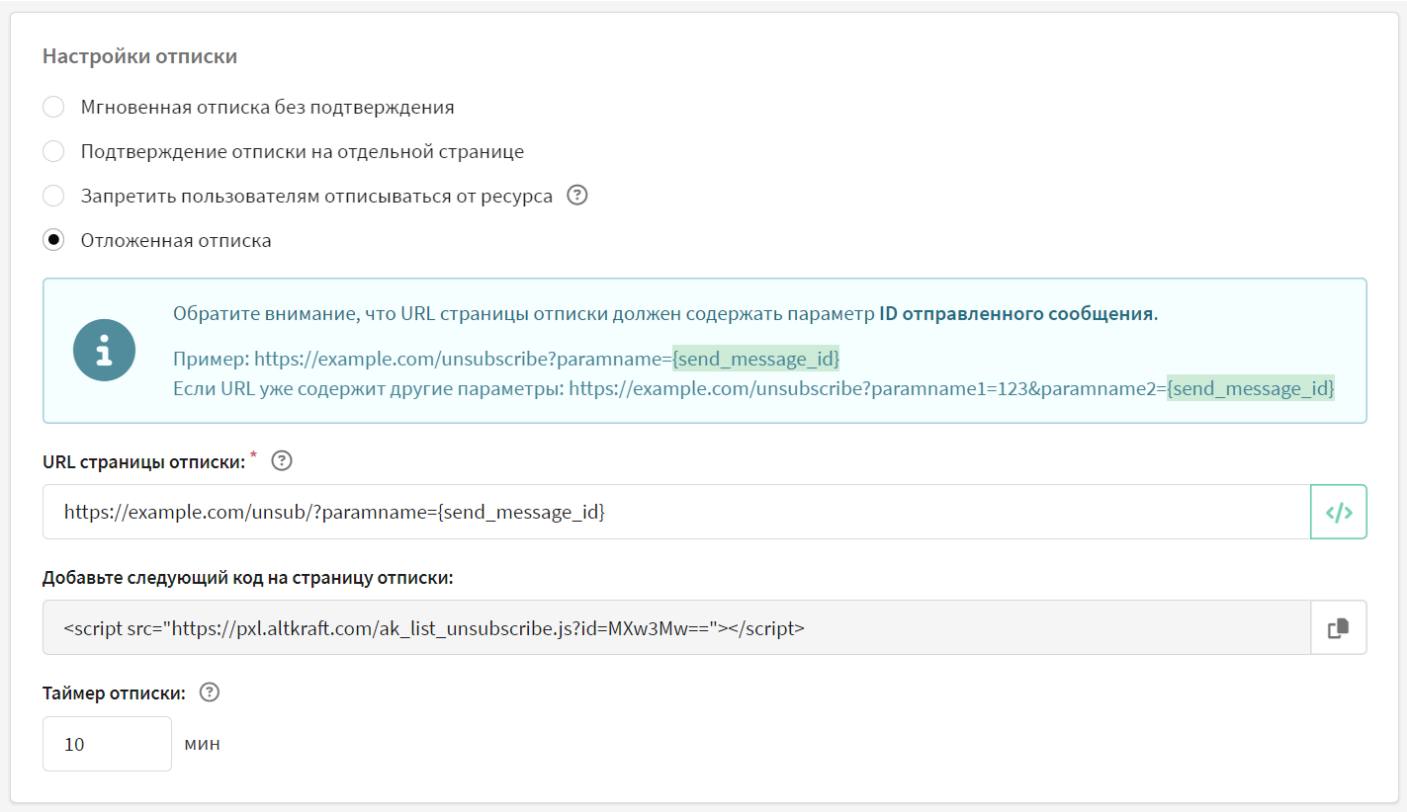
INFO

Отписка через ссылку list-unsubscribe в Email-письме также будет запрещена.

Отложенная отписка

При клике на ссылку отписки в сообщении профиль перейдет на пользовательскую страницу отписки. Его статус изменится на “Отписан”, если:

- профиль своими действиями подтвердит отписку;
- профиль не отменит отписку в течение времени, заданном в поле “Таймер” (максимальное значение таймера — 30 минут).



Настройка полей:

Укажите **“URL страницы отписки”** — этот URL переопределит URL-адрес системной страницы отписки.

В URL страницы отписки добавьте параметр **ID отправленного сообщения** `{send_message_id}`. Это обязательный параметр, который позволит идентифицировать ресурс и профиль, для которого требуется осуществить отписку.

Выберите механизм отписки:

Механизм отписки с использованием Altcraft SDK

> **CAUTION**
>
> Этот механизм реализуется **на стороне клиента** (в браузере). Безопасен для размещения на странице отписки, доступной пользователю. Все операции выполняются через SDK, без передачи токенов или других чувствительных данных в открытом виде.

Вставьте предоставленный в ресурсе код на вашу страницу отписки для подключения Altcraft SDK. Далее инициализируйте класс `AkUnsub`. Пример:

```
<!-- Подключение Altcraft SDK -->
<script src="https://pxl.altkraft.com/ak_list_unsubscribe.js?id=MXw3Mw=="></script>

<!--  Инициализация AkUnsub класса -->
<script>
  const akUnsub = new AkUnsub();
</script>
```

> **INFO**
>
> При необходимости вы можете переопределить токен ресурса и пиксель домен в скрипте:
>
> ```
> new AkUnsub({
>   rtoken: "example_token",
>   pxl_domain: "your_pxl_domain",
> });
> ```

Класс AkUnsub имеет 2 публичных метода:

- `confirmUnsub` — метод для подтверждения отписки
- `cancelUnsub` — метод для отмены отписки

Каждый из методов принимает следующие **аргументы:**

- **SMID** — идентификатор отправленного сообщения (обязательный параметр).
- **Массив категорий** ресурса, от которых происходит отписка (необязательный параметр).

Пример:

```
AkUnsub.confirmUnsub("SMID", ["cat_to_unsub"]);
AkUnsub.cancelUnsub("SMID", ["cat_to_unsub"]);
```

Для вызова методов на страницу отписки можно добавить кнопки. Пример:

```
<!-- Кнопка для подтверждения отписки -->
<button onclick="akUnsub.confirmUnsub('SMID',['category_1', 'category_2'])">
  Подтвердить отписку
</button>

<!-- Кнопка для отмены отписки -->
<button onclick="akUnsub.cancelUnsub('SMID',['category_1', 'category_2'])">Отменить
отписку</button>
```

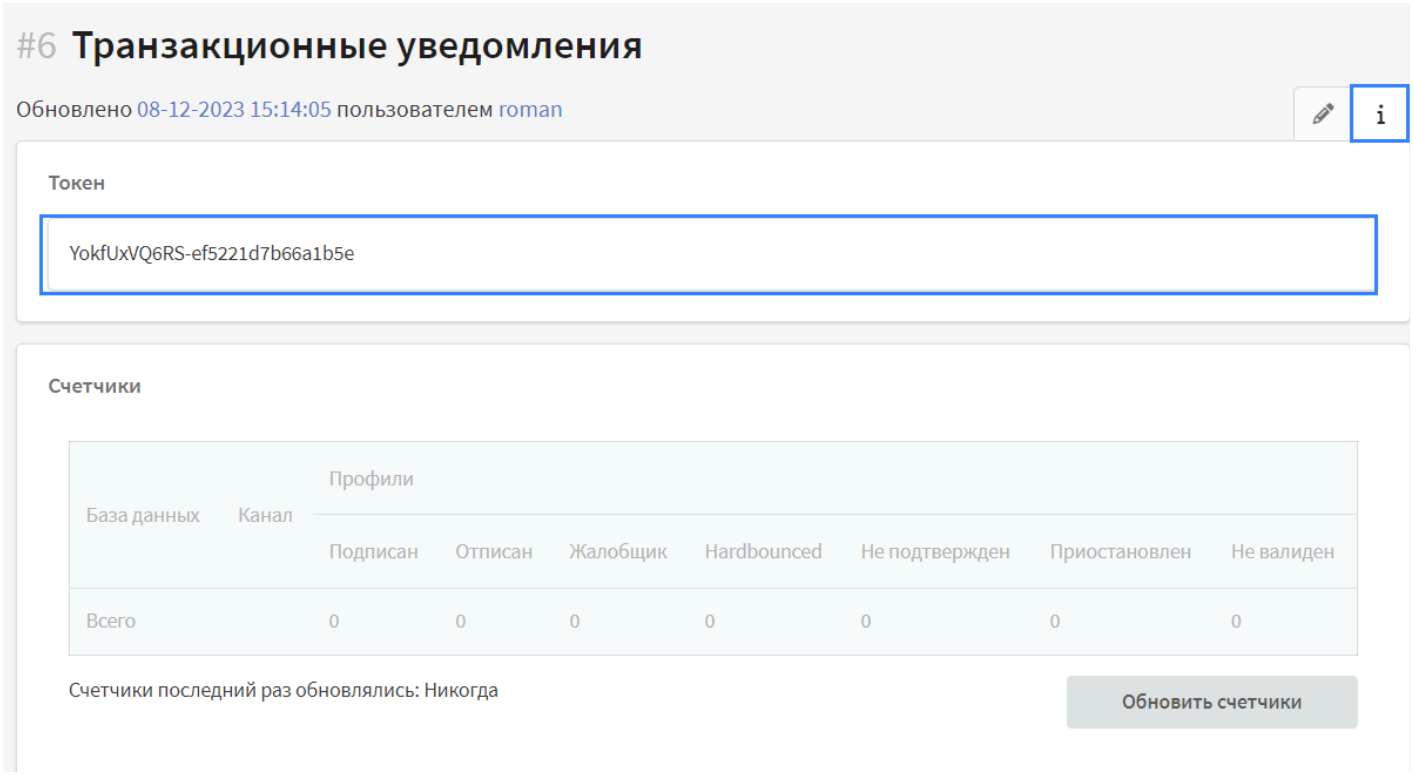
Замените `'SMID'` и `['category1', 'category2']` на соответствующие значения.

Механизм отписки без использования Altcraft SDK

> **CAUTION**
>
> Этот механизм реализуется **на стороне сервера**. Использует API-токен и токен ресурса, которые являются конфиденциальными. Их нельзя размещать в коде страницы или отправлять напрямую из браузера.

Вы можете реализовать механизм отписки на своей стороне. Для этого в **URL страницы отписки** необходимо добавить 2 параметра:

- **ID отправленного сообщения** `{send_message_id}`. Это обязательный параметр, позволяющий идентифицировать ресурс, от которого требуется осуществить отписку.
- **Токен ресурса** `{resource_token}`. Этот параметр требуется для дополнительной защиты и для реализации собственного механизма отписки без использования Altcraft SDK. Токен ресурса можно передавать не в URL страницы отписки, а в теле API-запроса. Токен можно получить в окне редактирования ресурса на вкладке “i” — детали:



Если профиль своими действиями подтвердил отписку на вашей странице, то необходимо передать эти данные в платформу с помощью API запроса `/api/v1.1/profiles/message_unsub`. Статус подписки профиля на ресурс изменится на “Отписан”. Если передавать параметр `categories`, то профиль будет отписан только от указанных категорий ресурса.

Пример запроса:

```
{
  "token": "abcdefghijklmnqrstuvwxyz", // API-токен
  "smid": "fGmXiWU9Y4NSIpCwl4QG6QXAAAQE", // ID отправленного сообщения
  "rtoken": "2fHLS9YR9daS-a90b371eecc2952f", // Токен ресурса
  "categories": ["cat_to_unsub"] // Категории ресурса, от которых отписать профиль
}
```

TIP

API запрос **/api/v1.1/profiles/message_unsub** описан в **документации**.

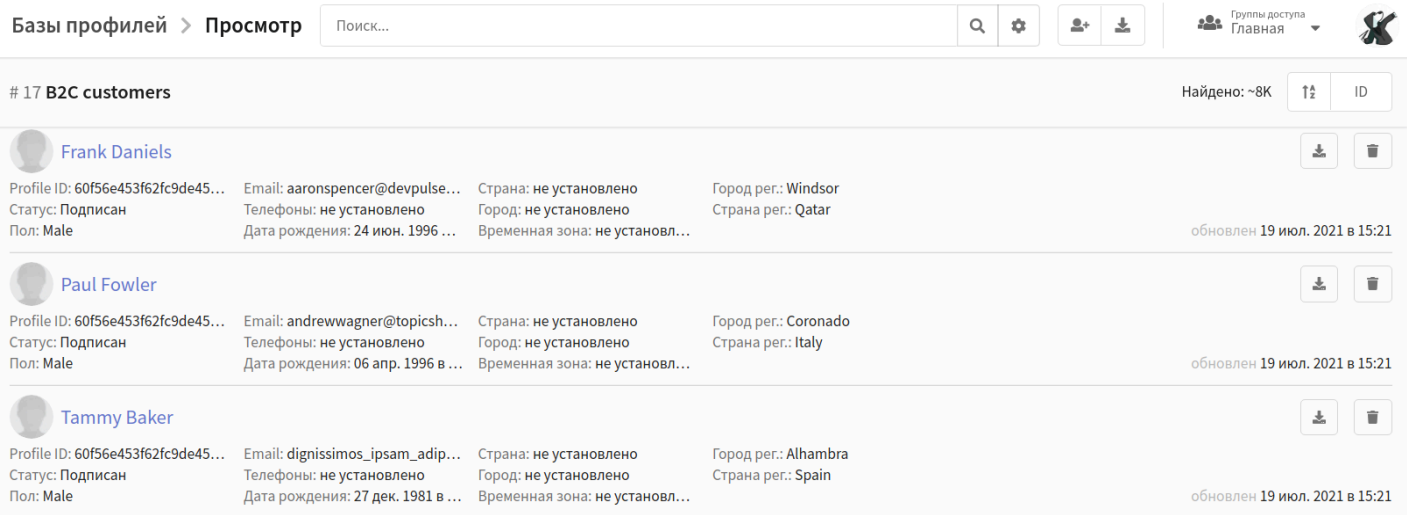
**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Работа с базами данных

В основе современного People-Based маркетинга лежат поведенческие, полученные в реальном времени, и личные данные потребителей.

База данных в Altcraft — это мощный и гибкий инструмент, который поможет вам накапливать и использовать информацию о профиле клиента.



Гибкость и полезность базы данных зависят от того, как она создана, заполнена и поддерживается. Если данные правильно подобраны и организованы, то появляется возможность выделять часть подписчиков на основании определённых критериев в отдельные целевые группы и рассылать каждой группе сообщения с таргетированным контентом.

Это называется сегментацией, и для этого у нас есть отдельный элемент — сегмент, в котором имеется большое число интуитивно понятных условий для выборки групп подписчиков.

Задумайтесь на минуту о своём маркетинге. Какие у вас цели, кто ваша аудитория и какая информация ей интересна.

Например:

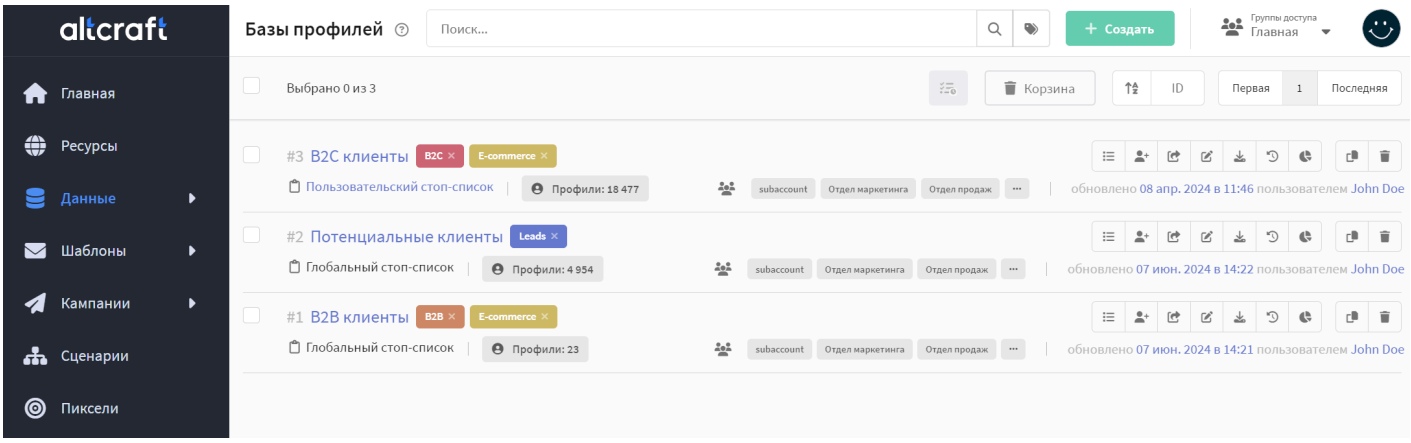
- Есть ли у вас сезонные покупатели, любят ли они определённые типы товаров?
- Требуется ли рассылать разный контент в зависимости от занятия подписчиков?
- Какие другие параметры влияют на вашу аудиторию?

Определите, какая информация о потребителях вам необходима и запишите её в блокнотик. Это пригодится при разработке дополнительных (настраиваемых) полей для хранения информации о клиентах в вашей базе.

Хорошая подготовка при создании базы данных позволит вам получить максимальную отдачу от неё, особенно в долгосрочной перспективе.

Управление базами профилей

Чтобы перейти к управлению базами данных, в основном меню выберите **Данные —> Базы профилей**. Для создания новой базы, нажмите **+ Создать** в верхней панели. Для редактирования существующей базы данных, нажмите на ее название.



В контекстном меню доступны следующие функции:

— открыть список профилей в базе

— импортировать профили клиентов (переход к разным вариантам импорта)

— скопировать профили в другую базу данных

— обновить профили (массовое обновление профилей в базе данных)

TIP

С помощью этого инструмента можно массово добавлять подписчикам определённые данные, подписывать профили на ресурсы, а также привязывать профилям клиентов промокоды или отвязывать их.

— экспортировать профили (выгрузить определенные данные о профилях клиентов в файл)

— экспортировать историю профилей (выгрузить данные о действиях профилей)

— клонировать базу (создать новую **пустую** базу данных с настройками выбранной; дополнительные поля для данных копируются в новую базу)

— удалить базу (полностью удалить базу данных с профилями)

TIP: Удаление базы

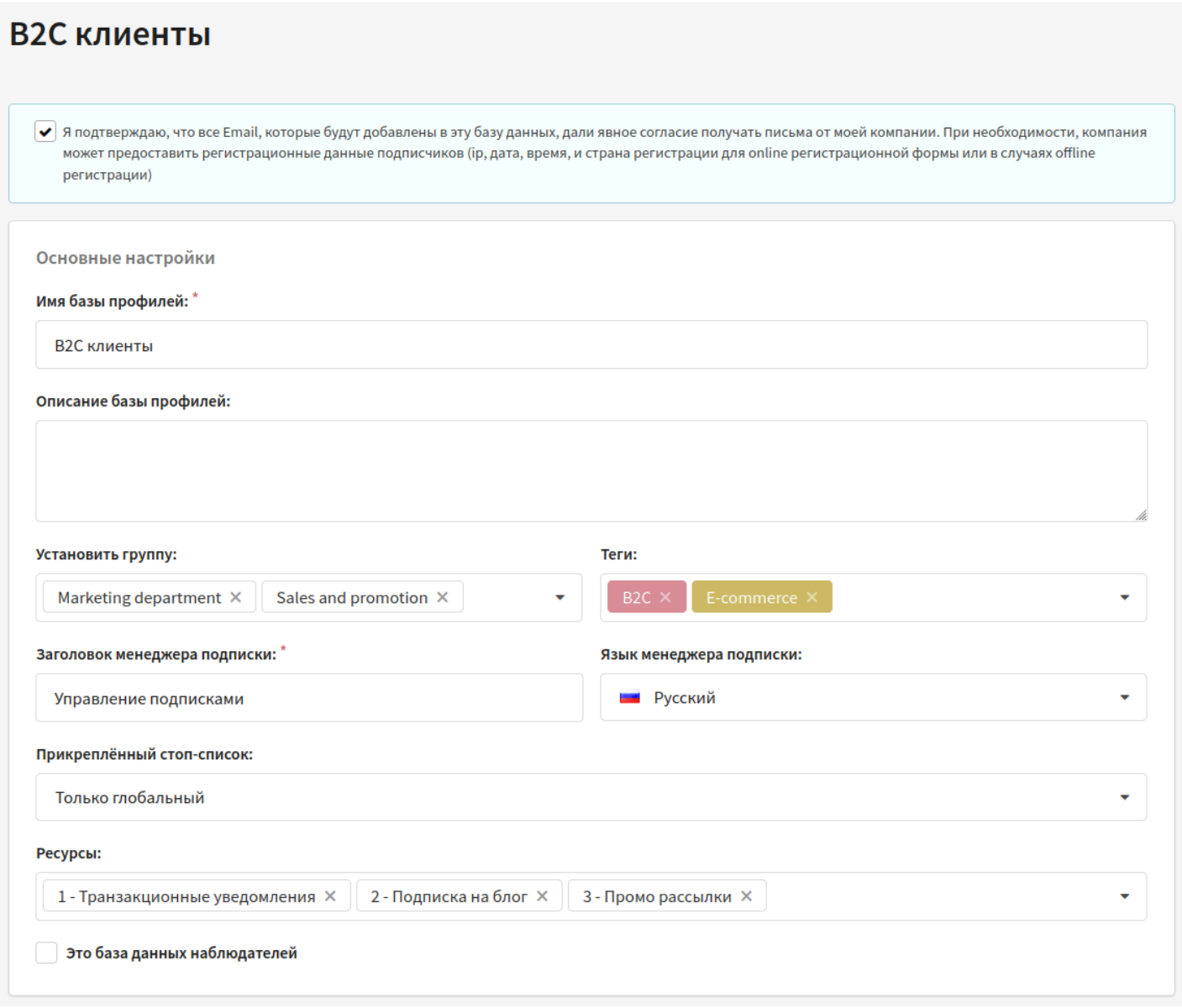
Удаление базы возможно только при условии, что в ней нет профилей, с которыми когда-либо происходило взаимодействие в рамках аккаунта. Если вы взаимодействовали с ними, например, отправляли рассылку, то необходимо предварительно удалить эти профили из базы.

Создание базы профилей

Основные настройки

Для новой базы данных укажите следующие настройки:

- **Имя базы профилей** — отображается во всех списках баз данных.
- **Описание** — произвольное описание назначения базы данных.
- **Установить группу** — назначить группы доступа к базе данных.
- **Теги** — используются для поиска баз данных по тегам.
- **Заголовок менеджера подписки** — отображается только в менеджере подписок.
- **Язык менеджера подписки** — определяет язык, на котором будет показана страница менеджера подписок для подписчиков этой базы данных.
- **Прикреплённый стоп-список** — блокирует отправку сообщений по email каналу на содержащиеся в нём адреса.
- **Ресурсы** — определяют возможность коммуникации с профилями базы по различным каналам в рамках конкретной тематики. Настройка ресурса описана в этой статье.
- **Это база данных наблюдателей** — опция определяет базу данных наблюдателей за рассылками. Базы данных с этой опцией можно выбирать в Smart настройках рассылки.



Стандартные поля

Стандартные поля профиля подготовлены для записи наиболее часто востребованной информации о клиенте.

Значение по умолчанию — используется, когда при импорте профиля не было передано данных для этого поля.

Сделать обязательным — данные для этого поля будут необходимы при импорте. (Если установить значение по умолчанию, то импорт без данных для этого поля будет возможен)

Добавить в менеджер подписок — поле появится на странице менеджера подписок.

Стандартные поля

Поле	Тип	Значение по умолчанию	Сделать обязательным	Добавить в менеджер подписок
Email	Строка		☑	☑
Телефоны	Список телефонов		☐	☑
Имя	Строка		☐	☐
Фамилия	Строка		☐	☐
Дата рождения	Дата	%Y-%m-%d	☐	☐
Пол	Список	Не выбрано	☐	☐
Дата рег.	Дата	%Y-%m-%d	☐	☐
IP адрес рег.	IP	127.0.0.1	☐	☐
Город рег.	Строка		☐	☐
Страна рег.	Строка	Не выбрано	☐	☐
Ссылка рег.	Строка		☐	☐

NOTE

Поля с гео-данными клиентов делятся на два типа:

- Первый ("рег." в названии поля) — информация, полученная при регистрации: на основе IP адреса или статических данных импорта.
- Второй — последние данные, полученные из истории действий клиента: также на основе IP адреса или переданных данных.

Дополнительные (пользовательские) поля

Дополнительные поля данных создаются пользователями системы Altcraft для хранения специфической для бизнеса информации о клиенте. Количество дополнительных полей не ограничено.

Такие поля могут содержать данные 10 типов для реализации различной бизнес-логики:

- **Число** — положительные и отрицательные целые числа.
- **Число с точкой** — положительные и отрицательные числа с плавающей точкой.
- **Строка** — текст на любом языке, включая цифры, специальные символы, пробелы. Используется стандарт Unicode.
- **Дата** — дата, записанная в формате RFC 3339, например, `2019-08-22T00:00:00.000Z`.
- **Boolean** — поле, которое предоставляет возможность выбора из двух вариантов `True` или `False` (Да или Нет). Используется для отражения наличия (или отсутствия) у клиента того или иного продукта или

другого признака, например, VIP статуса.

- **Список** — это поле похожее на **boolean**, но позволяющее выбирать из неограниченного числа вариантов, созданных самим пользователем, только один подходящий. Например, тарифный план клиента. Варианты записываются в строчку через запятую.
- **IP** — IP в формате IPv4. Например, `127.0.0.1`.
- **Теги** — поле, в котором может быть записано несколько значений одновременно. Это могут быть приобретённые клиентом продукты, услуги или другие значения, состав которых может меняться. Теги записываются в строчку через `,` `|` или `;`. Поэтому в самом теге эти символы не допустимы. Например: `Продукт-1, Продукт 2, Продукт_3`.
- **Объект** — поле для хранения неопределенных данных: набор флагов или сложные структуры данных.
- **Массив объектов** — поле для хранения подтаблицы, которая, например, может содержать список менеджеров, договоров и др.

INFO: Валидация в объектах и массивах

1. **Правила наименования.** В именах полей объектов допустимо использовать только латинские буквы (в любом регистре), цифры от 0 до 9, символ нижнего подчеркивания `_` и дефис `-`.

2. **Тип данных.** В полях “Массив объектов” или “Объект” можно хранить данные типа `int`, `float`, `string`, `boolean`.

Пример

```
[
    {
        "key1": 12.3,
        "key2": 12,
        "key3": true,
        "key4": "text"
    }
]
```

3. **Структура объектов.** В поле с типом “Массив объектов” или “Объект” можно записывать только одноранговые объекты или одноранговые объекты, вложенные в другие объекты. Если вы используете объекты с более сложной структурой, подключите SQL-базу и создайте запросы для выборки профилей. Также массивы должны иметь одну структуру в рамках указанного пользовательского поля.

Допустимые значение

```
[
        {"x":1},
        {"x":2}
]

//ИЛИ

{
  "key_1":
    {
      "sub_key_1":
        {"x":1,"y":2}
    },
  "key_2":
    {
      "sub_key_1":
        {"x":3,"y":5}
    }
}

//ИЛИ

[
    {"key": "value_1", "new_key": "new_value1"},
    {"key": "value_2", "new_key": "new_value2"}
]
```

Недопустимые значения:

```
[
        ["x","y"],
        ["z","w"]
]

//ИЛИ

{
  "key":
    [
      ["x","y"],
      ["z","w"]
    ]
}

//ИЛИ

[
    {"key1": "value_1", "new_key10": "new_value1"},
    {"key7": "value_2", "new_key300": "new_value2"}
]
```

Дополнительные поля

Поле	Тип	Значение по умолчанию	Сделать обязательным	Добавить в менеджер подписок	
Custom_Integer	Целое число	123	☐	☐	✕
Custom_Float	Число с точкой	123.45	☐	☐	✕
Custom_String	Строка	Test	☐	☐	✕
Custom_ISO_date	Дата	01-01-2026	☐	☐	✕
Custom_Boolean	Логический	Истина	☐	☐	✕
Custom_List	Список	Option_1	☐	☐	✕
Custom_IP	IP	127.0.0.1	☐	☐	✕
Custom_Tags	Теги	Tag 1, Tag 2, Tag 3	☐	☐	✕
Custom_Object	Объект	{"X":1,"Y":2,"Z":3}	☐	☐	✕
Custom_Array	Массив объектов	[{"key":"value_1"},{"key":"value_2"}]	☐	☐	✕

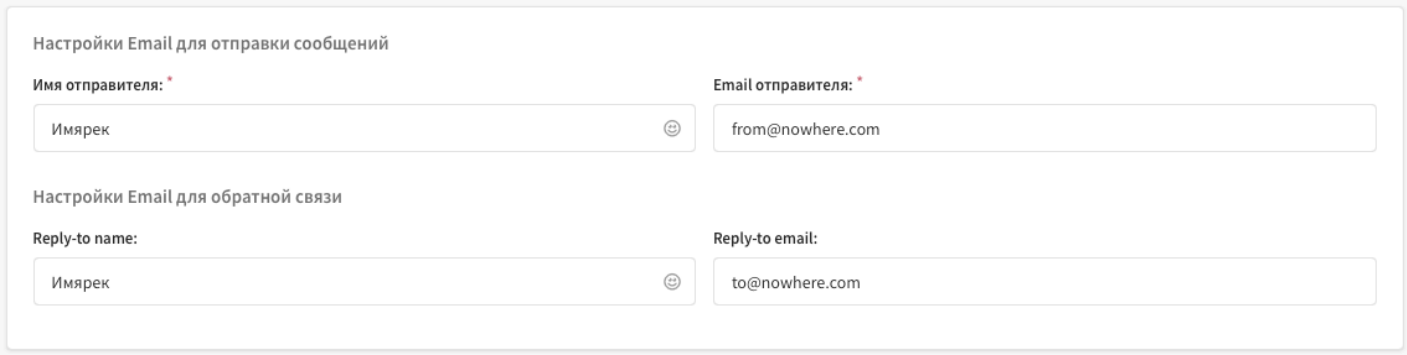
TIP

Подробную информацию о типах полей в базе и допустимых значениях вы найдете в **отдельной статье**.

Настройки Email

Для базы данных нужно указать значения по умолчанию для следующих полей:

- Имя отправителя
- Email отправителя
- Имя для ответов
- Email для ответов



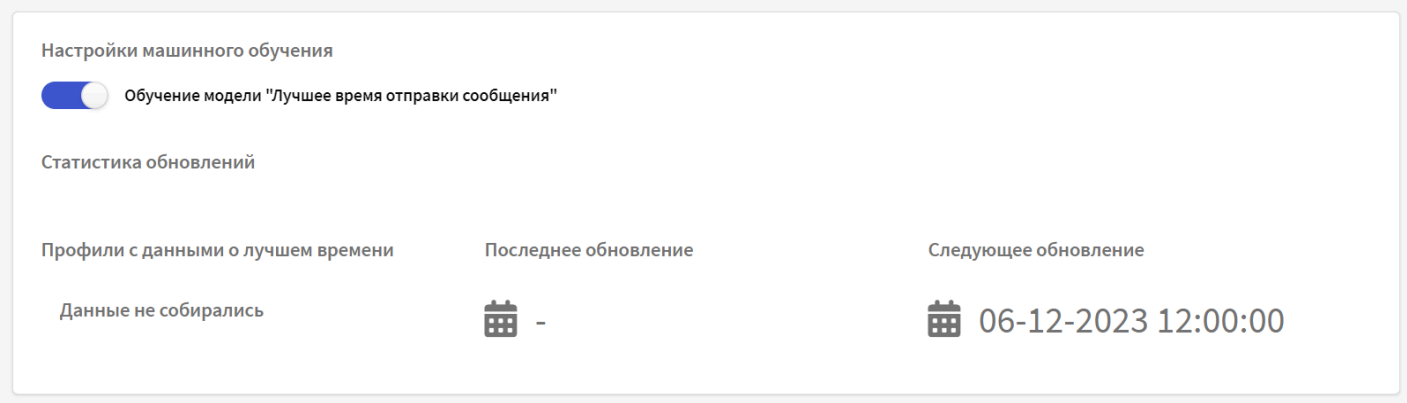
Включение машинного обучения

Включение машинного обучения в базе позволяет определить лучшее время для отправки сообщения профилю. Далее эти данные можно использовать для сегментации профилей и настройки сценария.

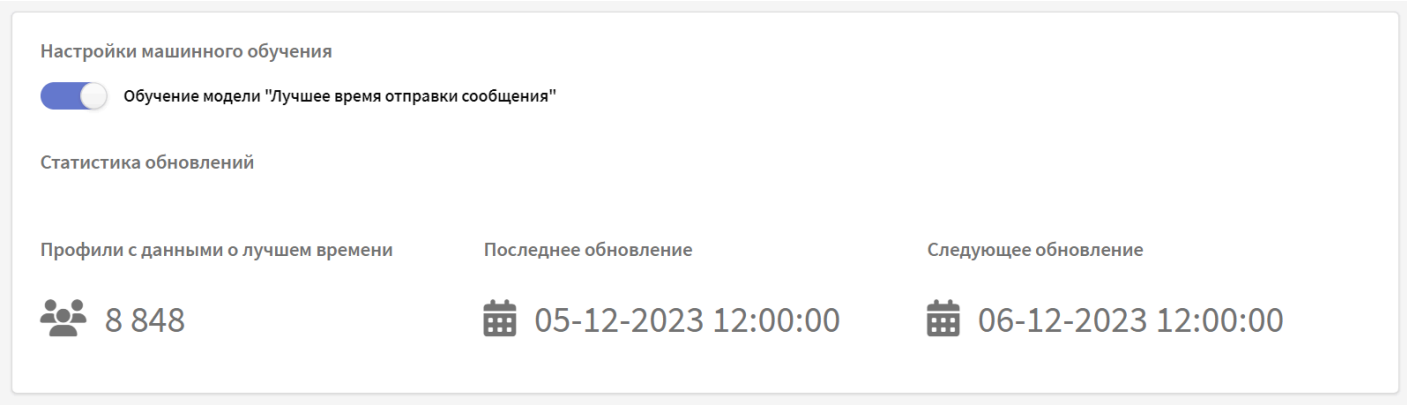
INFO

Для активации функционала требуется добавить дополнительные поля в конфигурационный файл платформы.

Активируйте обучение модели с помощью переключателя. После сохранения настроек вы увидите окно с информацией о том, когда запланирован пересчёт данных профилей в базе по машинному обучению.



Первые данные по лучшему времени отправки появятся после первого рассчёта. Помимо даты последнего и следующего обновления данных, здесь также отображается количество профилей в базе, для которых удалось рассчитать лучшее время отправки.

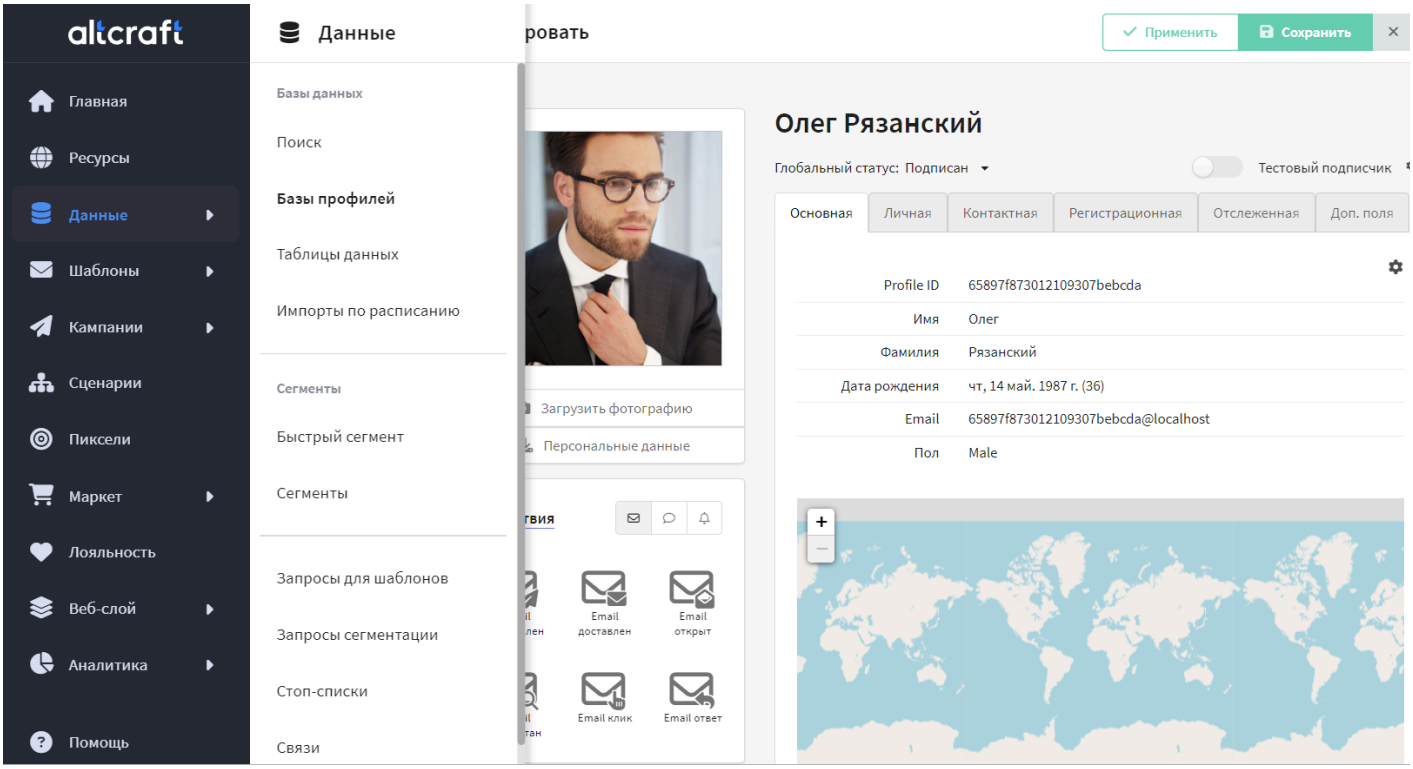


TIP

Расписание для обновления данных по машиному обучению устанавливается в файле конфигурации в поле `ML_REFRESH_TIME_HOURS` (по часовому поясу аккаунта). По умолчанию — в 00:00.

Профиль подписчика

Информация о получателях рассылок — клиентах, партнёрах, наблюдателях за рассылками — хранится в **профилях**, организованных в базы данных. База профилей определяет структуру основных и дополнительных **полей данных профиля**, а также возможность подписок на те или иные ресурсы.

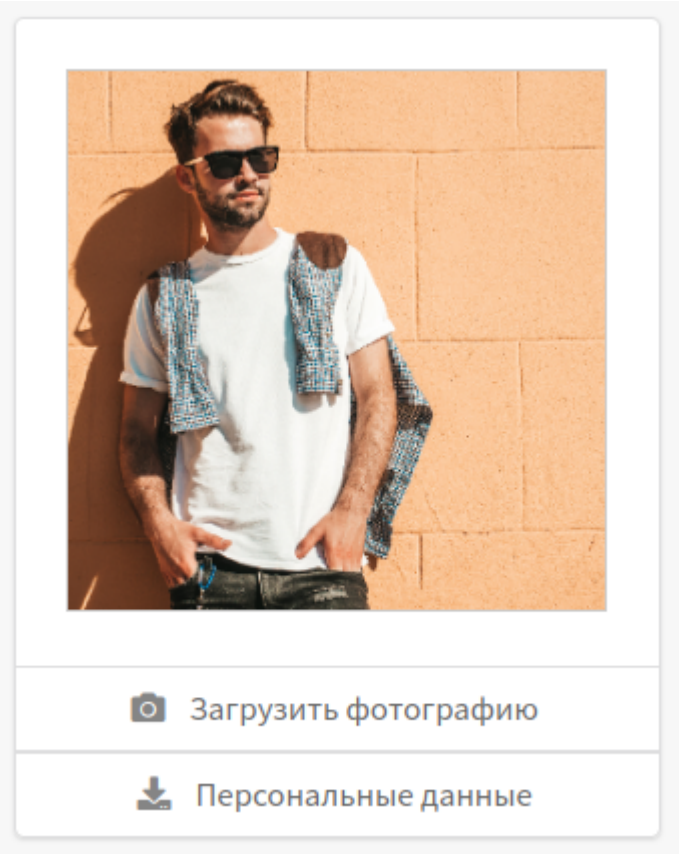


Профиль подписчика объединяет в себе несколько типов данных. На главной странице отображается та информация, которую вы записываете в базу данных вручную, из файла или автоматически в рамках интеграции. Вкладки в нижней части страницы содержат дополнительные сведения о клиенте:

- **История** — история событий, связанных с клиентом: отправки, открытия, достижение целей и др.
- **Взаимодействия** — участие профиля в статических сегментах и маркетинговых сценариях; можно вручную отправить профиль в сценарий, добавить в сегмент или запустиить триггерную рассылку.
- **Данные** — информация о промокодах, заказах и связях профиля, а также список форм, которые он заполнял.
- **Подписки** — подписки клиентов (контактные данные, информация о регистрации) на различные ресурсы.

Аватар и выгрузка данных

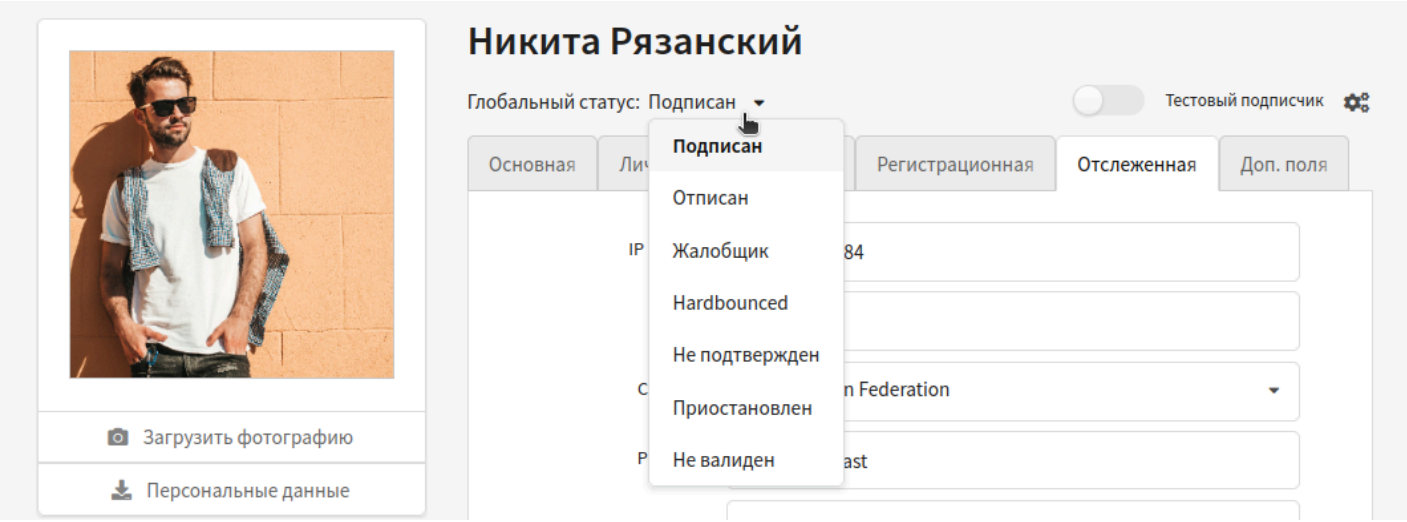
В левой верхней части страницы находится аватар вашего клиента. С помощью кнопок под изображением вы сможете вручную загрузить изображение или скачать данные клиента.



Глобальный статус профиля

Возможность коммуникации с клиентом по любым каналам связи определяет глобальный статус профиля. Статусы отдельных подписок зависят от глобального статуса. Так, если у профиля есть несколько активных подписок, но глобальный статус запрещает коммуникацию — отправки происходить не будут. Всего доступно 7 статусов подписки:

1. **Подписан** — доступен для коммуникации по всем каналам связи.
2. **Отписан** — отписался от получения сообщений по всем каналам.
3. **Жалобщик** — пожаловался на Спам. Коммуникация остановлена.
4. **Hard bounced** — провайдер полностью отказался принимать сообщения на этот адрес, скорее всего ящика не существует и попытки что-то отправить будут портить ваш рейтинг у провайдера. Невалидные адреса записываются в глобальный стоп-список платформы.
5. **Не подтверждён** — подписчик зарегистрировался, но не подтвердил согласие на получение рассылок. Используется при Double opt-in подписке. После подтверждения меняется на **Подписан**.
6. **Приостановлен** — подписка приостановлена на некоторое время.
7. **Не валиден** — исключен из маркетинговых рассылок по любым другим причинам.



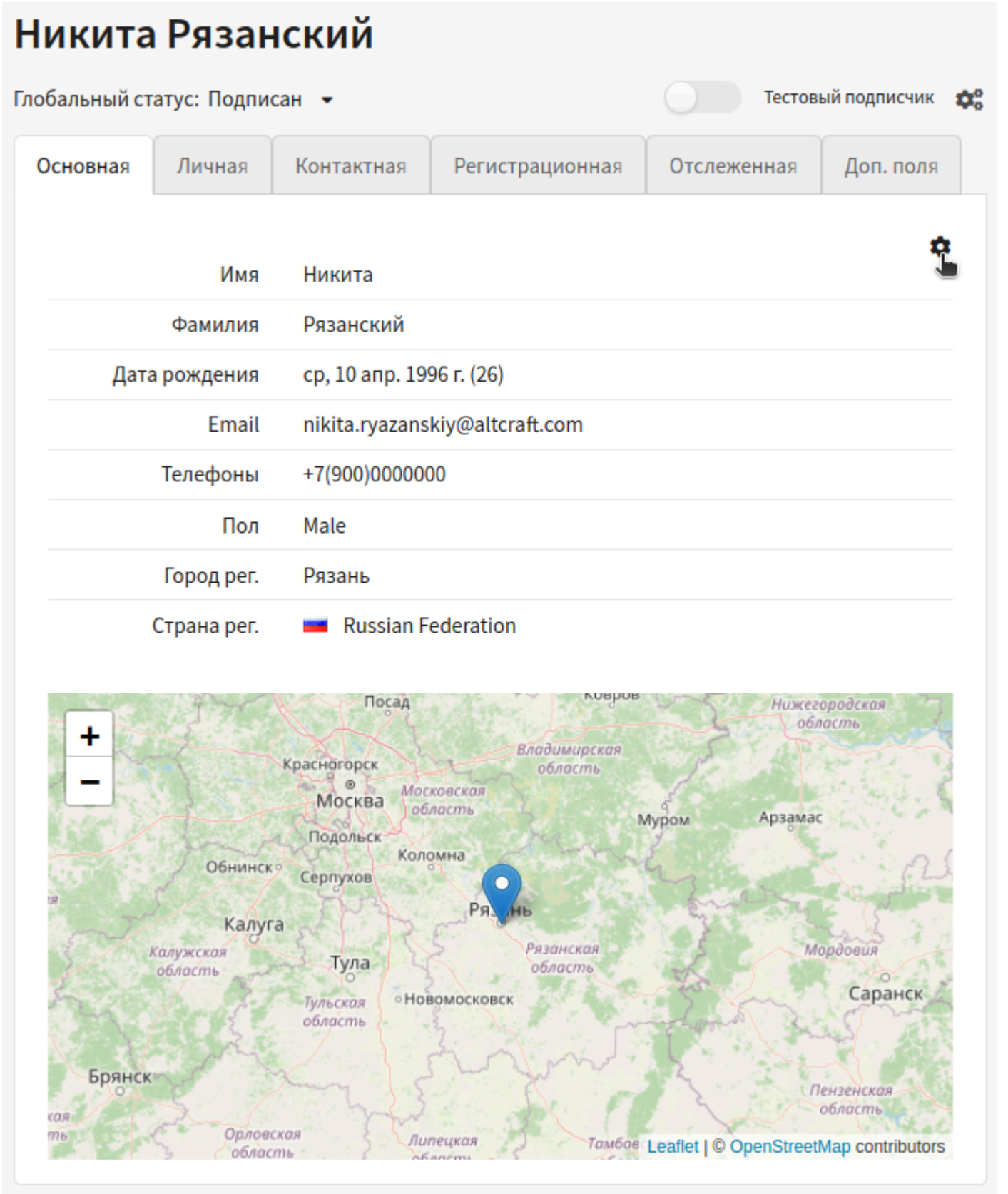
Данные профиля

Информация о клиенте разделена на несколько вкладок:

- **Основная** — базовая информация о подписчике, геолокация. Нажмите на "шестерёнку", чтобы отображать на основной вкладке любые необходимые поля.
- **Личная** — демографические данные клиента.
- **Контактная** — основные контактные данные клиента.
- **Регистрационная** — opt-in информация, подтверждающая законность коммуникации.
- **Отслеженная** — IP адрес и геоданные последнего отслеженного действия клиента.
- **Доп. поля** — пользовательские поля базы данных с бизнес-информацией.

CAUTION

Поля Email и Телефоны в Контактной информации служат для идентификации профиля в базе данных. В маркетинговых рассылках используются подписки профиля на ресурсы.



TIP

Данные о местоположении подписчика можно внести в профиль **вручную**. Для этого используются два типа **стандартных** полей:

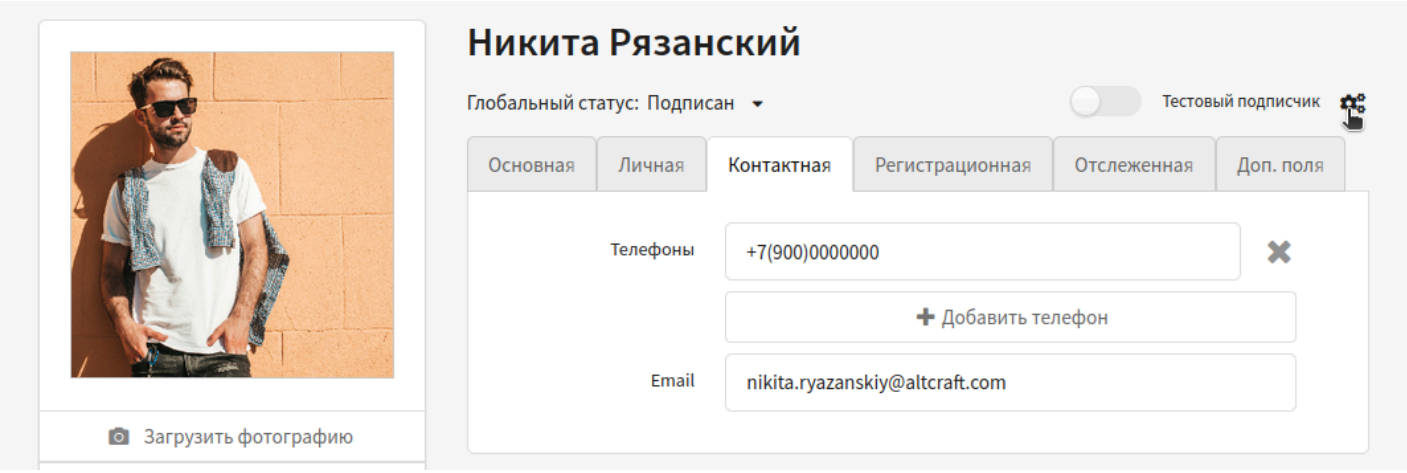
- **Последние известные данные:** Страна, Регион, Город, Индекс и Временная зона.
- **Данные регистрации:** Страна рег., Город рег.

Автоматически геолокация в Altcraft определяется по IP адресу подписчика. Это происходит, когда клиенты открывают письма, переходят по ссылкам или совершают действия на сайте. Помимо указанных выше полей по IP определяются географические координаты, которые отображаются на встроенном виджете карты.

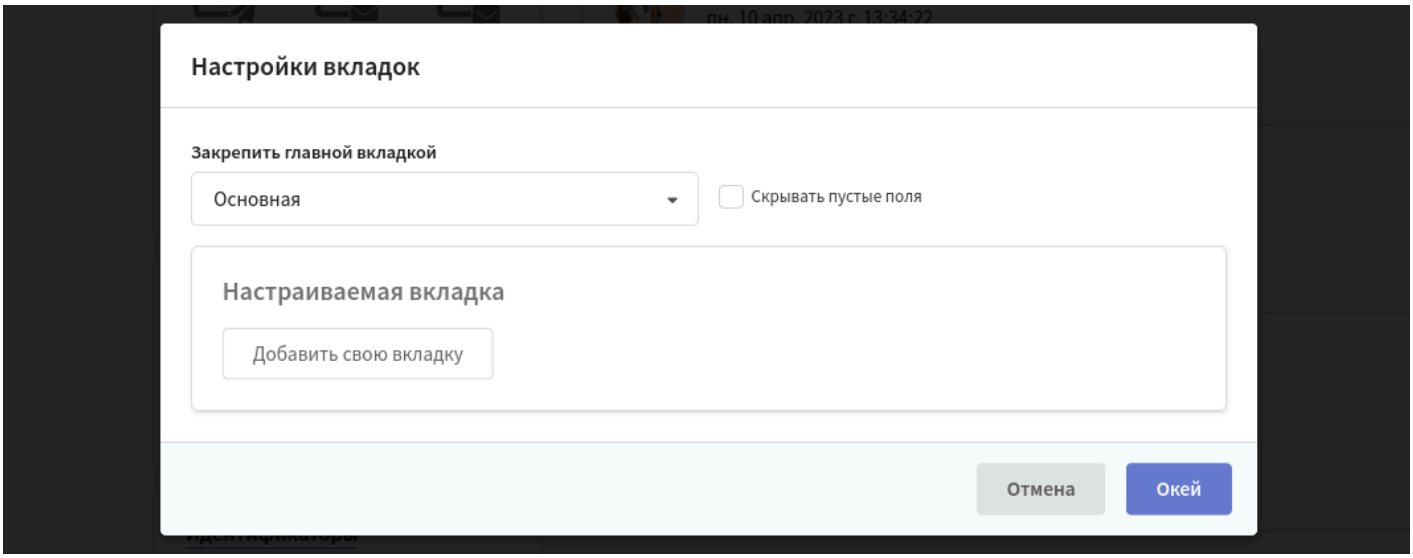
Также трекинг обновляет системное поле **User agent**.

При импорте подписчика доступно автоопределение локации. Для данных регистрации предусмотрено отдельное поле **IP адрес рег.** Для импорта из файла опцию автоопределения необходимо активировать вручную, для API она включена по умолчанию.

Когда вы открываете профиль, по умолчанию будет открыта **Основная** вкладка. Чтобы изменить эти настройки, нажмите на иконку с шестерёнками:



В открывшемся окне вы можете выбрать, какую вкладку закрепить, а также создать и настроить собственную вкладку с необходимой информацией:



INFO

Дополнительные поля настраиваются в структуре базы профилей. Такие поля могут содержать данные 10 типов для реализации различной бизнес-логики:

1. **Число** — положительные и отрицательные целые числа.
2. **Число с точкой** — положительные и отрицательные числа с плавающей точкой.

3. **Строка** — текст на любом языке, включая цифры, специальные символы, пробелы. Используется стандарт Unicode.
4. **Дата** — дата, записанная в формате RFC 3339, например: `2019-08-22T00:00:00.000Z`.
5. **Boolean** — поле, которое предоставляет возможность выбора из двух вариантов `True` или `False` (Да или Нет). Используется для отражения наличия (или отсутствия) у клиента того или иного продукта или другого признака, например — VIP статуса.
6. **Список** — это поле похожее на **boolean**, но позволяющее выбирать из неограниченного числа вариантов, созданных самим пользователем, только один подходящий. Например — тарифный план клиента. Варианты записываются в строчку через запятую.
7. **IP** — IP в формате IPv4. Например, 127.0.0.1
8. **Теги** — поле, в котором может быть записано несколько значений одновременно. Это могут быть приобретённые клиентом продукты, услуги или другие значения, состав которых может меняться. Теги записываются в строчку через `"," "|" или ";"`. Поэтому в самом теге эти символы не допустимы. Например: `Продукт-1, Продукт 2, Продукт_3`.
9. **Объект** — поле для хранения неопределенных данных: набор флагов или сложные структуры данных.
10. **Массив объектов** — поле для хранения подтаблицы, которая, например, может содержать список менеджеров, договоров и др.

Подробную информацию о типах полей в базе и допустимых значениях вы найдете в **отдельной статье**.

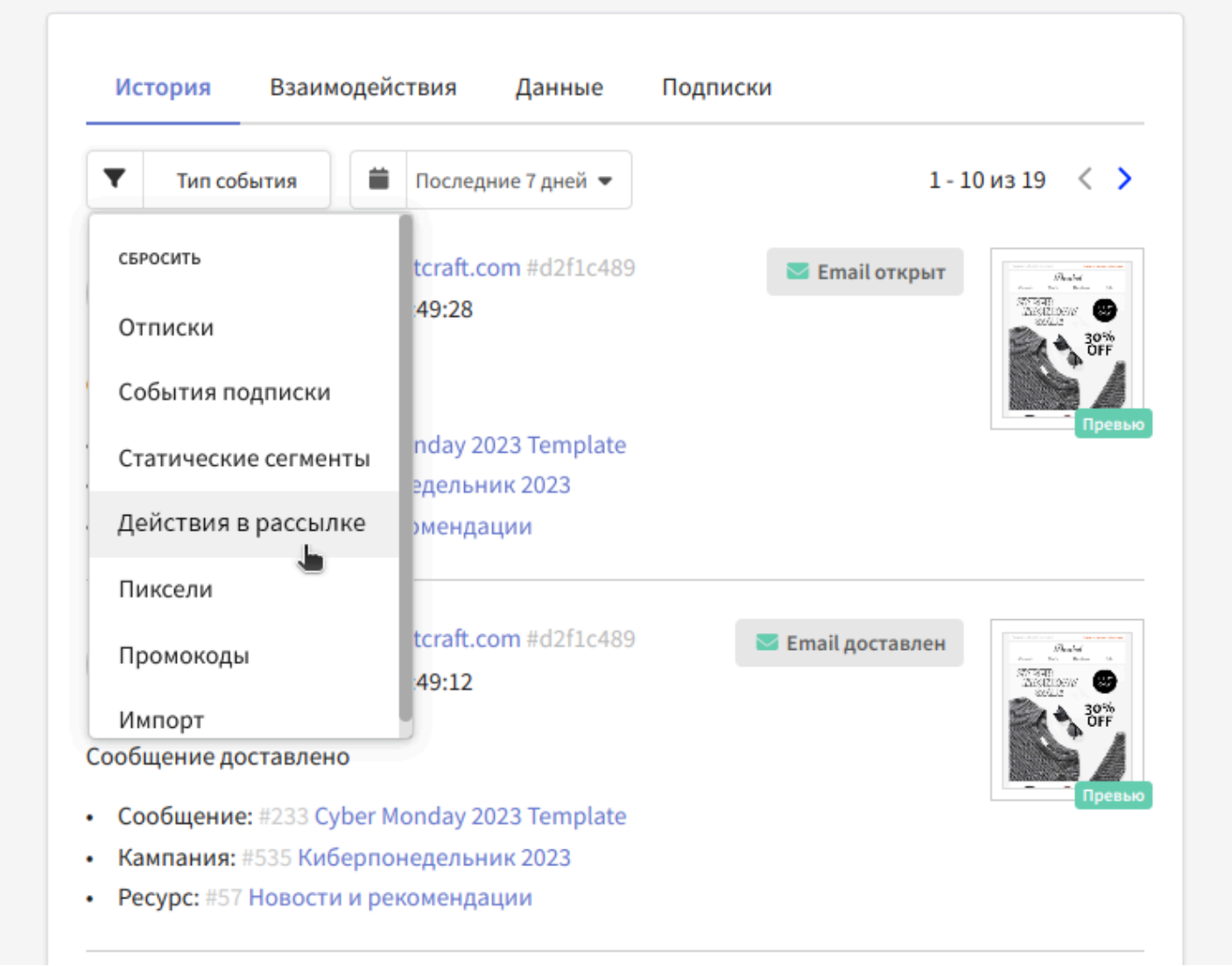
История активности

В истории отображаются все события, связанные с профилем клиента:

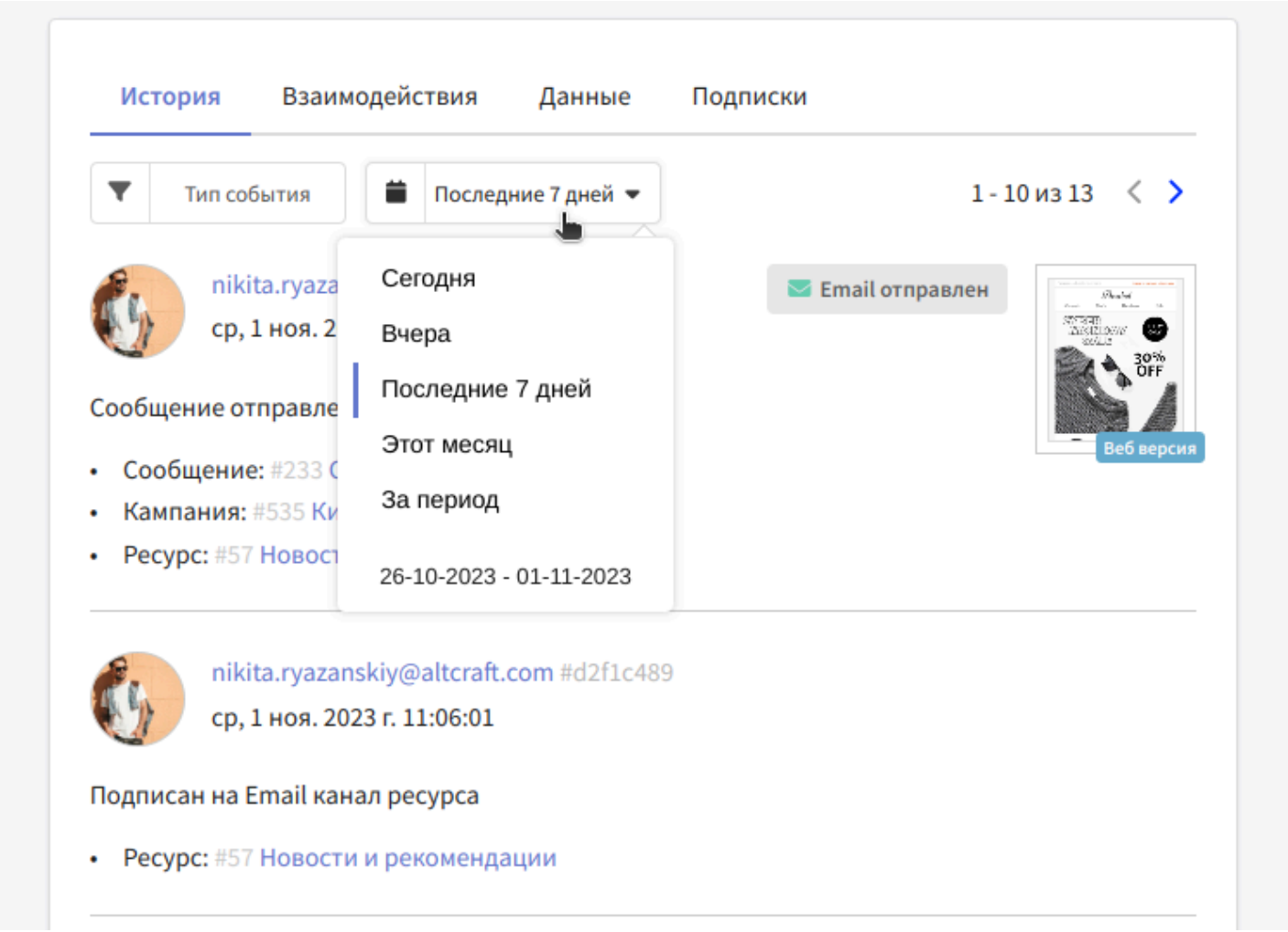
- **Импорт** — способ добавление профиля в базу: вручную, через API импорт, импорт из файла, push-импорт.
- **События подписки** — добавление подписки на определенный канал ресурса (подписан на Email/SMS/Push/Telegram/WhatsApp*/Viber/Notify/пользовательский канал), а также отписка от канала в ресурсе (изменение статуса подписки на "Отписан").
- **Действия в рассылке** — основные события, отслеживаемые в рассылках: отправки, доставки, открытия, прочтения, клики, переходы по ссылке подтверждения, а также изменения статуса подписки на "Hardbounce" и "Жалобщик".
- **Отписки** — событие отписки, которое фиксируется, когда профиль переходит по ссылке отписки в сообщении.
- **Пиксели** — выполнение действия на сайте или в мобильном приложении, в результате которого сработал пиксель отслеживания. Фиксируется достижение цели и ее стоимость, а также дополнительная информация о событии (например, геоданные, данные о промокодах, заказах и др.).
- **Промокоды** — промокод привязан, отвязан или активирован.
- **Сегмент** — профиль вошел или вышел из статического сегмента.
- **Лояльность** — любые действия в рамках программ лояльности, изменения уровня или бонусного баланса.

Чтобы посмотреть определенный тип события, используйте фильтр. При клике на

условия в фильтре сбросятся.



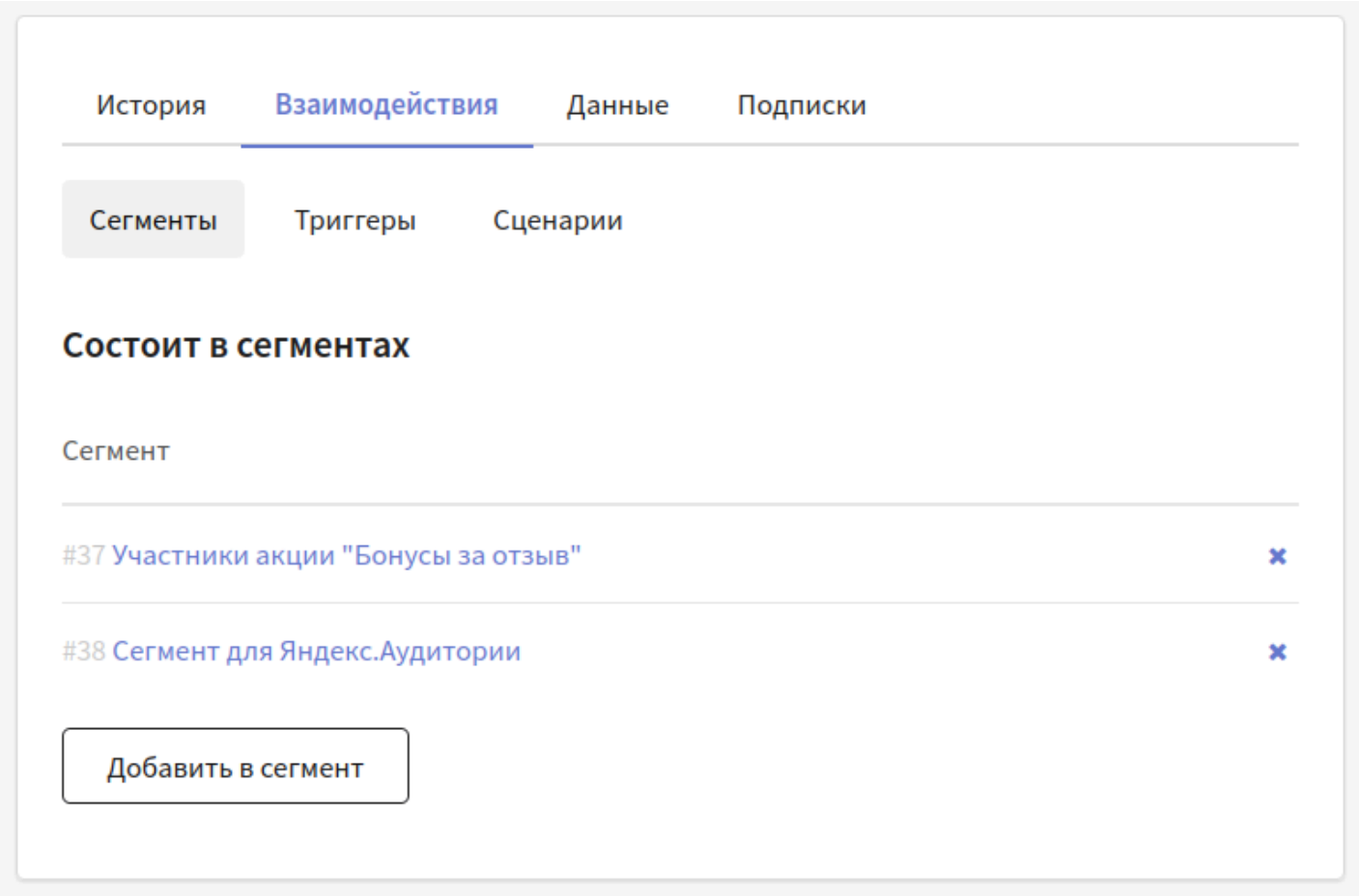
По умолчанию отображается история действий профиля за последние 7 дней. Вы можете выбрать другой период:



Взаимодействия

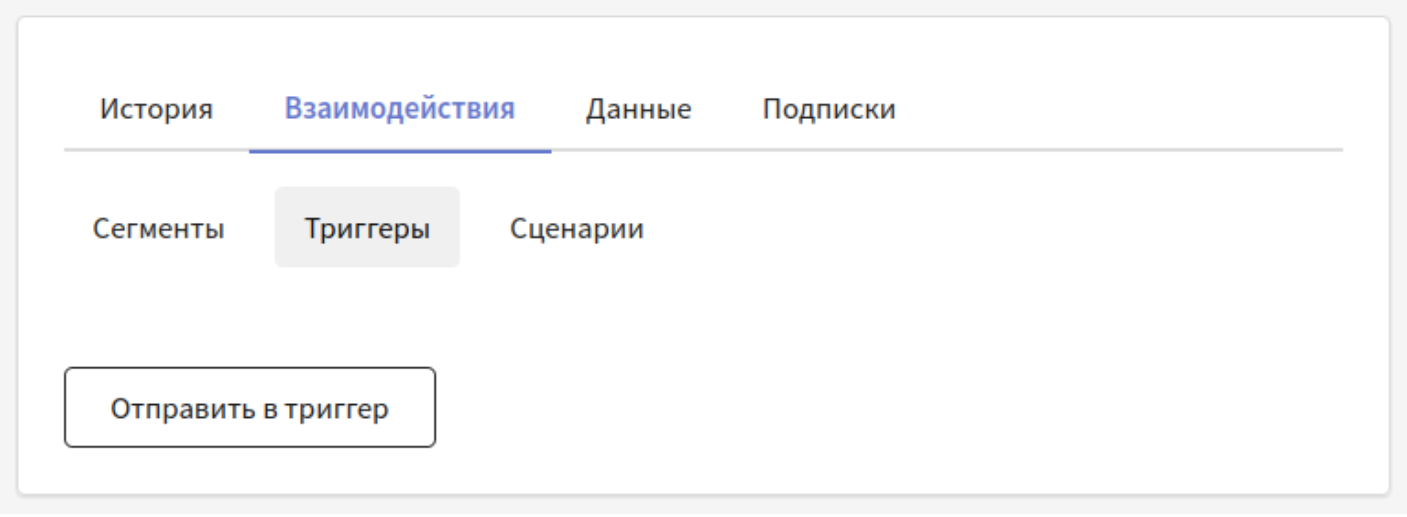
Сегменты

Статические сегменты используются, когда вам нужно работать с фиксированной группой клиентов. Изменения в статических сегментах можно экспортировать автоматически в дата-центр компании. Здесь вы можете добавить клиента в сегмент или удалить из него вручную.



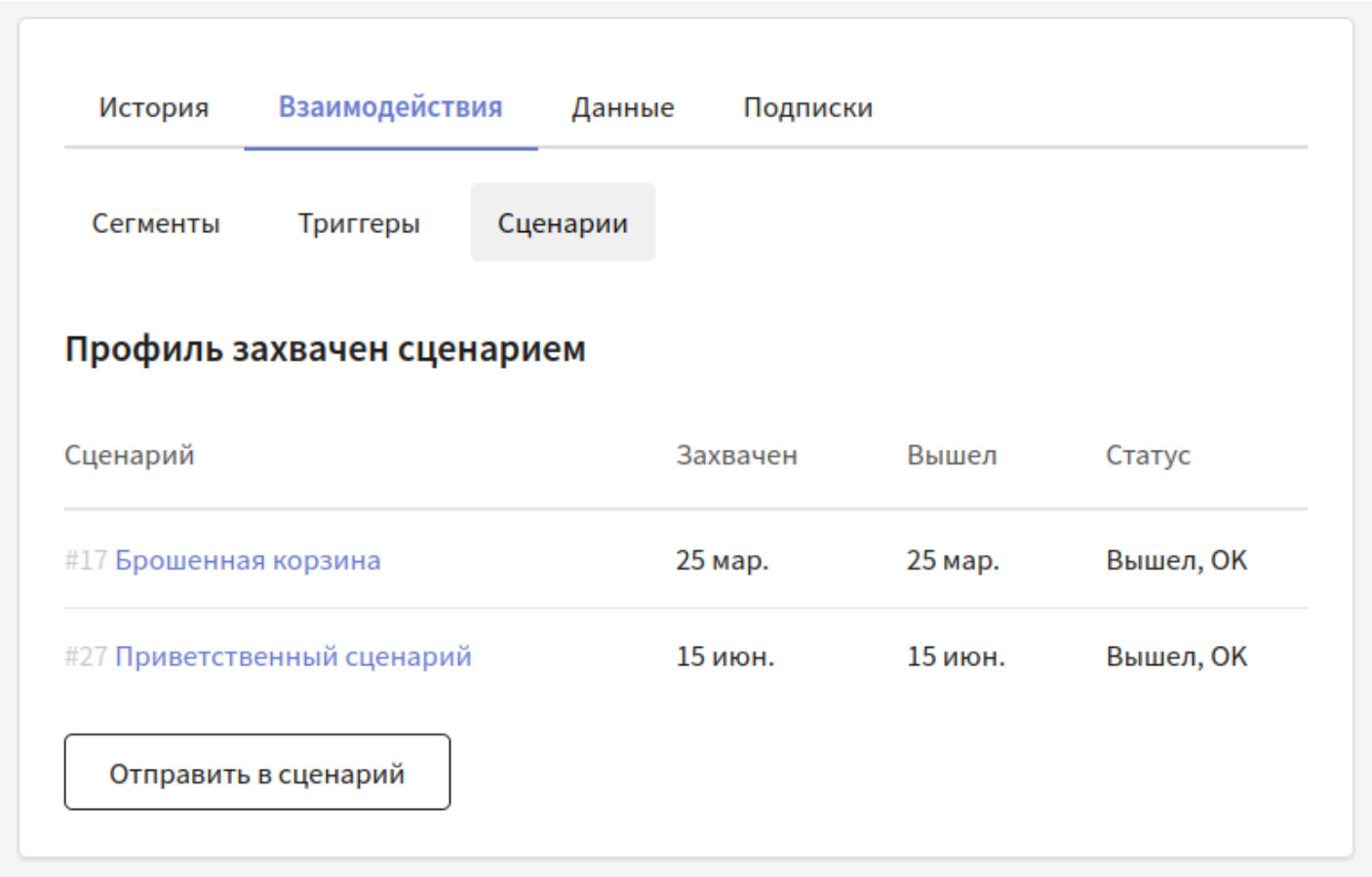
Триггеры

Триггер — это рассылка, которая запускается при наступлении определенного события (переход по ссылке, заполнение формы и др.). Отсюда можно вручную отправить профилю триггерную рассылку.



Сценарии

Сценарии — цепочки автоматизации вашего маркетинга. На этой вкладке отображаются сценарии, в которых участвовал клиент. Отсюда же можно вручную отправить профиль клиента в сценарий.



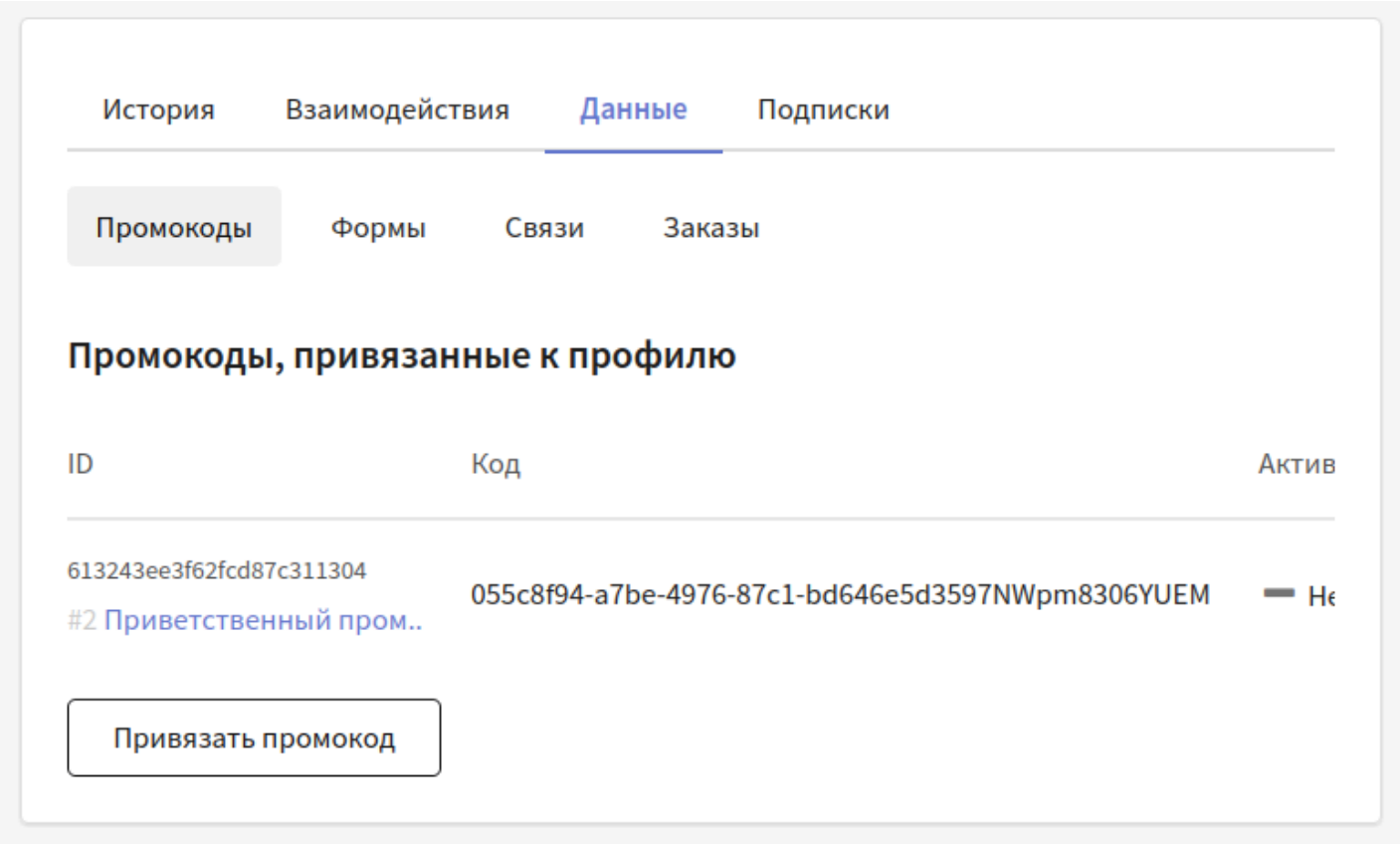
Кампании

Кампании — это инструмент для комплексного управления маркетинговыми активностями в едином пространстве. На этой вкладке отображаются кампании в которых учствует профиль. При необходимости отдельные профили можно вручную включить в кампанию, например для тестирования или проверки работы активностей.

Данные

Промокоды

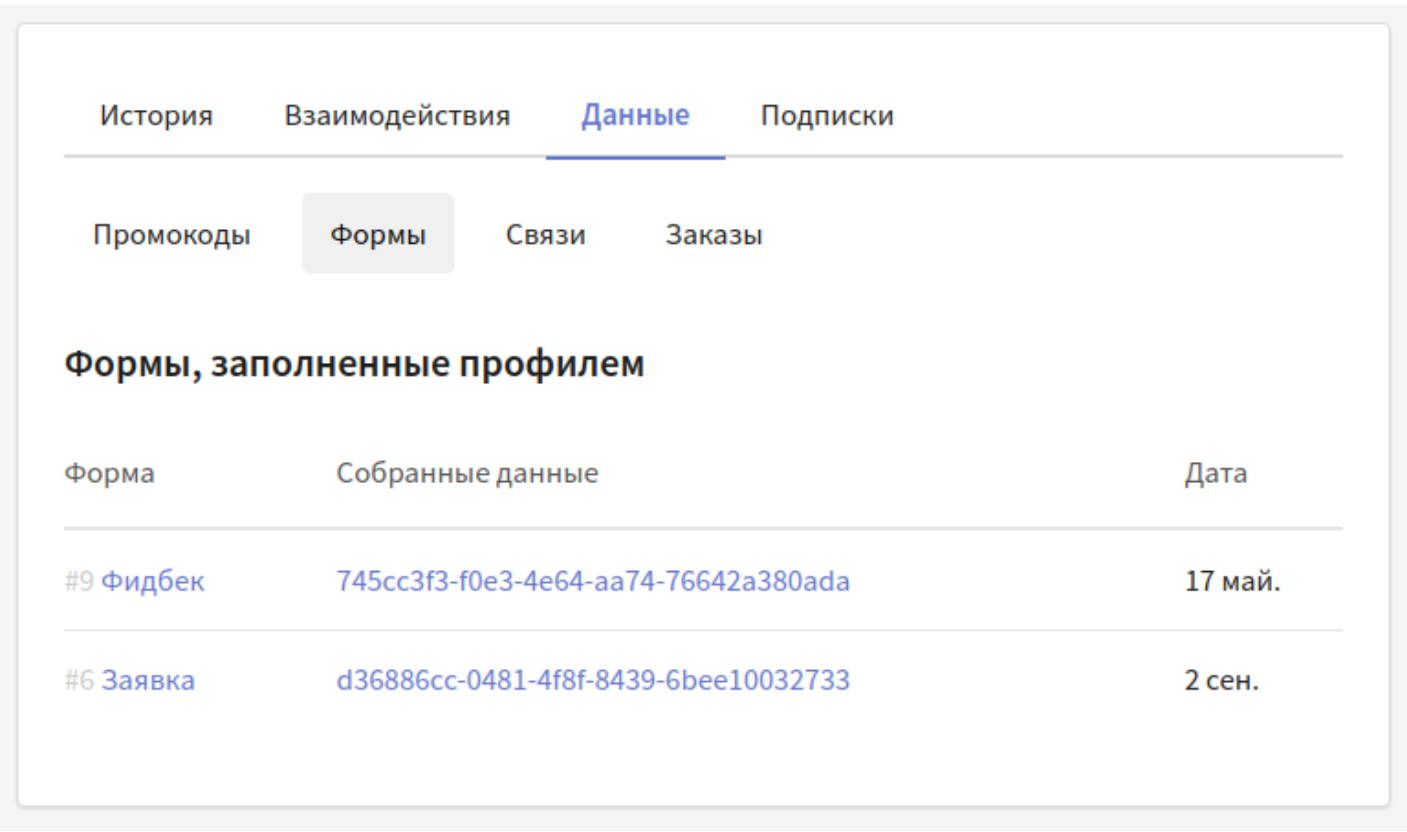
Промокоды используются для стимулирования сделок. В Altcraft Platform за них отвечает модуль “**Лояльность**”— промокоды можно генерировать в платформе или загружать из внешнего сервиса. Также отслеживается информация об активации и истечении срока действия промокода. На этой вкладке вы вручную можете назначить клиенту промокод.



Формы

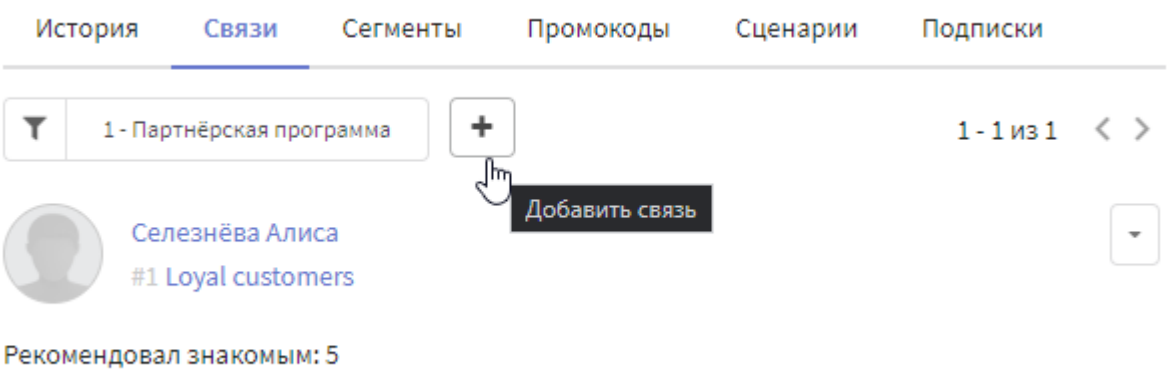
Форма — это инструмент для сбора данных от ваших клиентов. С её помощью можно реализовывать различные механизмы с использованием внешних данных, например, добавлять новые профили в базу, управлять подписками на различные ресурсы и получать обратную связь от клиентов.

На этой вкладке показаны все формы, заполненные профилем. При клике на собранные данные вы перейдете в новое окно, где можно посмотреть данные, которые оставил подписчик при заполнении.



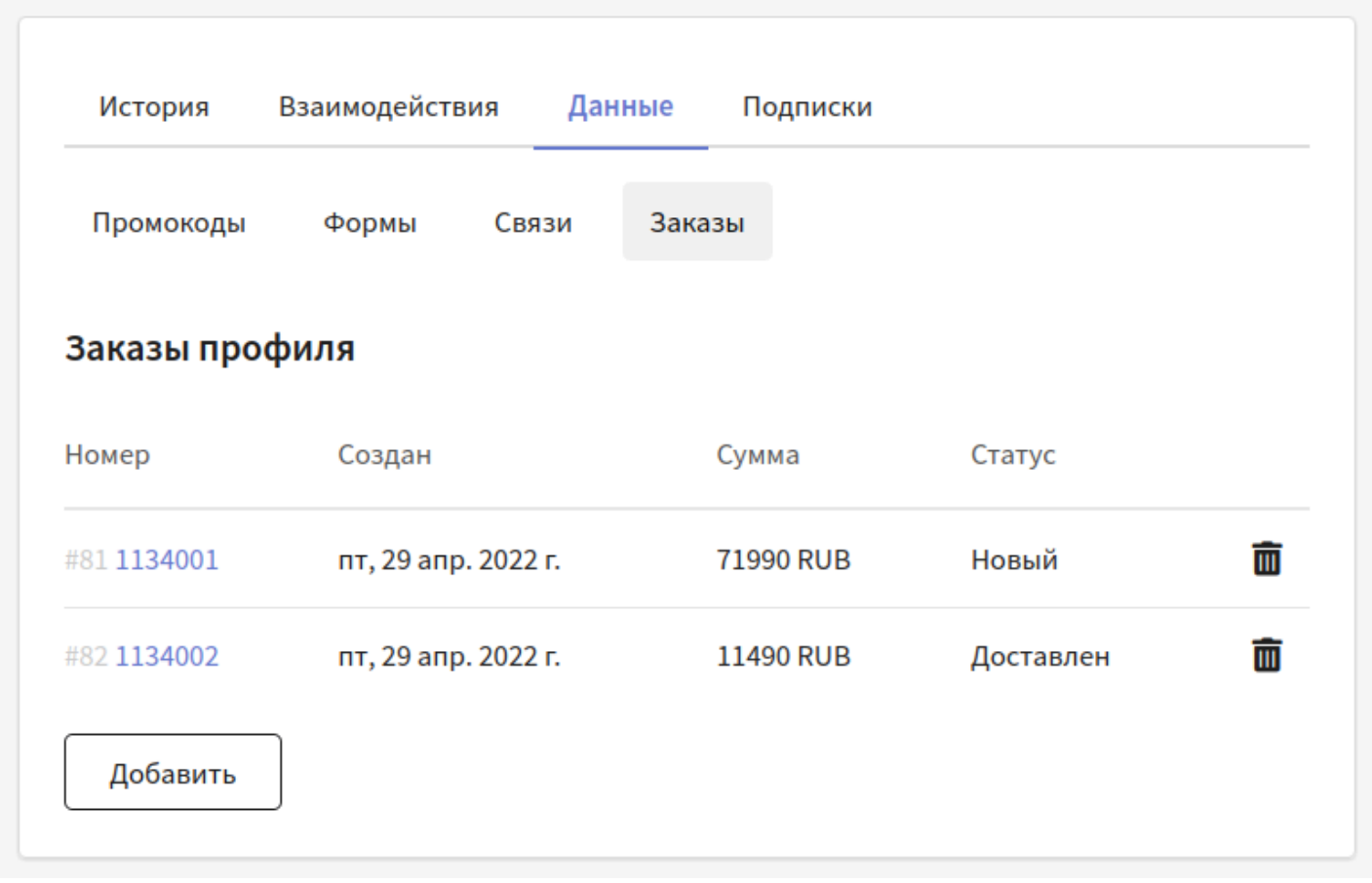
Связи

Связи могут использоваться для учёта взаимодействий между профилями. Это могут быть лайки, или подписчики в социальной сети, или рекомендации товара/сервиса по партнёрской программе. На этой вкладке вы можете выбрать связь и добавить профилю вручную с необходимыми свойствами.



Заказы

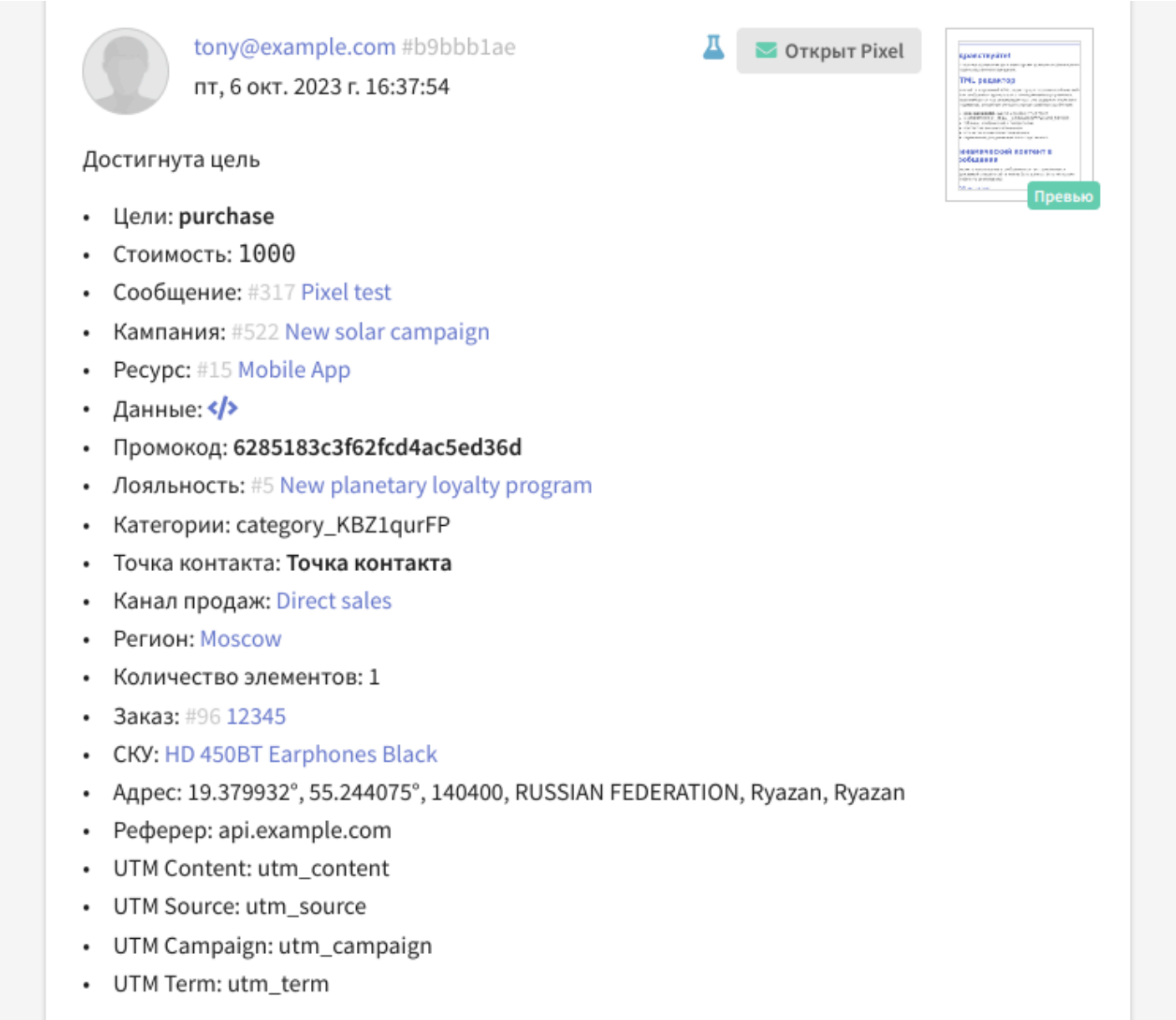
Заказы — это элемент модуля **“Маркет”**. Этот модуль позволяет загрузить данные о продуктах и заказах в платформу и использовать эти данные для коммуникации с подписчиками. Для активации этого модуля необходимо настроить соответствующие роли доступа.



Пиксели

Пиксель — это HTML или JavaScript-код, размещенный на веб-сайте или в приложении, который предназначен для отслеживания действий пользователей. Когда подписчик выполняет определенное действие на сайте или в приложении, на сервер отправляется запрос для регистрации этого действия.

Пиксель также может передавать информацию, связанную с этим действием, например, достигнутую цель, ценность совершённого действия, URL сайта, с которого перешёл подписчик, геоданные профиля, информацию об отправленных рассылках и другое. Эти данные записываются в историю профиля и используются для сегментации.



TIP

Для просмотра данных, переданных в объекте `pixel_data`, нажмите `</>` в пункте **“Данные”**.

CAUTION

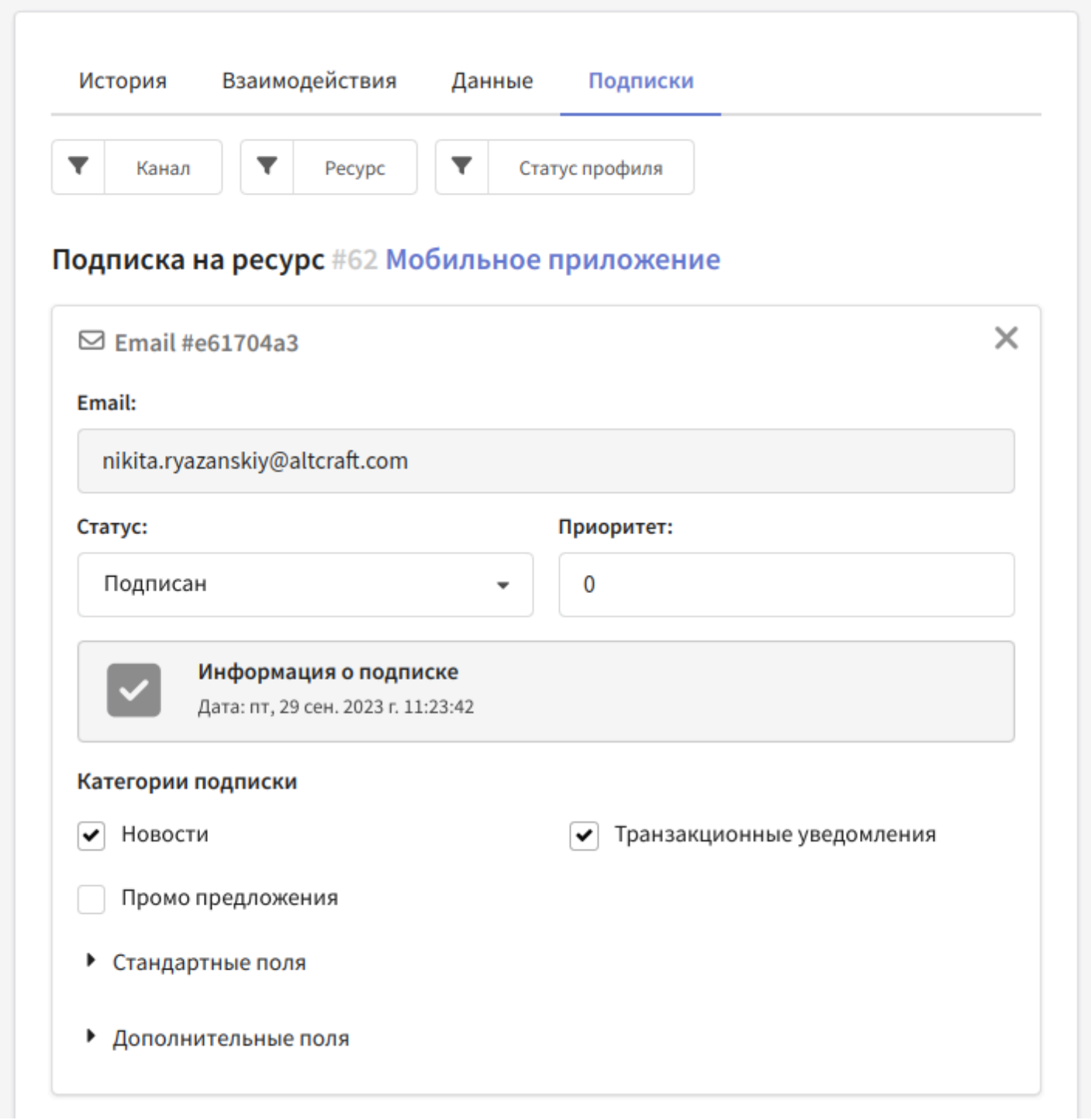
Цели могут регистрироваться для клиентов, чьих контактных и личных данных у вас ещё нет. Такие события называются **анонимными**. Друг от друга Altcraft отличает их по данным о соединении (cookie).

Когда клиент после этого регистрируется из того же браузера, его данные записываются в уже существующий профиль.

Подписки профиля

Подписка профиля — запись, объединяющая контактные данные по конкретному каналу связи и сведения о регистрации. В разных ресурсах могут использоваться одинаковые контактные данные. Так, например, отписавшись от одного ресурса, клиент продолжит получать желательный контент в рамках другого. У каждого профиля установлен лимит в 200 подписок, либо до 20 подписок на один ресурс в одном канале.

Здесь вы можете вручную изменить приоритет или статус подписки:



Статус

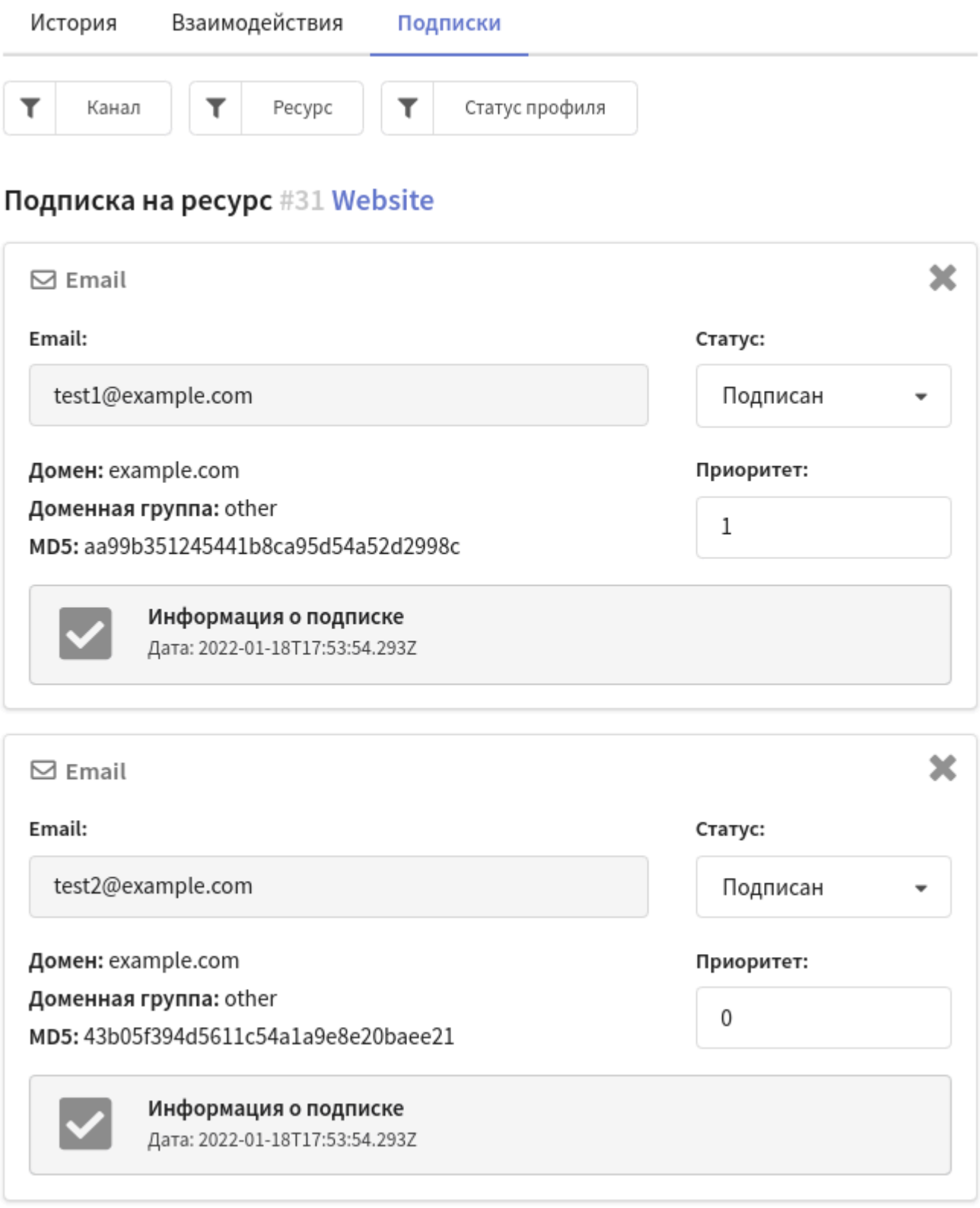
Статус отдельной подписки определяет возможность коммуникации с клиентом по конкретным контактным данным в рамках определённого ресурса. Здесь, как и в глобальном статусе профиля, есть семь вариантов:

1. **Подписан** — подписка активна, контактные данные будут использоваться для отправок по ресурсу.
2. **Отписан** — клиент отписался от рассылок на эти контактные данные в рамках данного ресурса.
3. **Complainer** — клиент пожаловался на Спам. Коммуникация остановлена.
4. **Hard bounced** — провайдер полность отказался принимать сообщения на этот адрес, скорее всего ящика не существует и попытки что-то отправить будут портить ваш рейтинг у провайдера. Невалидные адреса записываются в глобальный стоп-список платформы. Подписки, содержащие такие же данные, не будут использоваться в рассылках.
5. **Не подтверждён** — подписчик зарегистрировался, но не подтвердил согласие на получение рассылок. Используется при Double opt-in подписке. После подтверждения меняется на **Подписан**.
6. **Приостановлен** — подписка приостановлена на некоторое время.
7. **Невалиден** — подписка не будет использоваться по любым другим причинам.

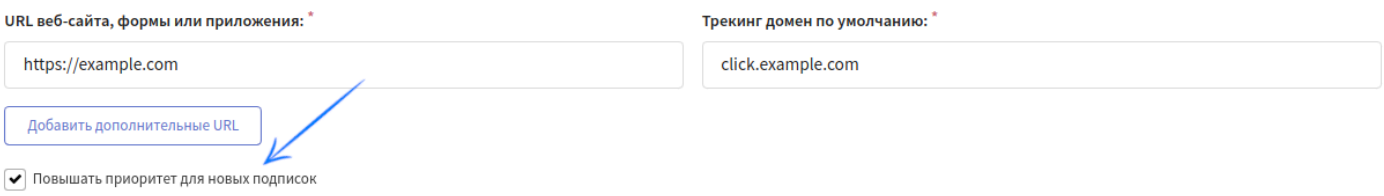
Приоритет

Пользователь может подписаться на один ресурс несколько раз, указав разные контактные данные (email или номер телефона). Чем больше значение в поле “Приоритет”, тем ниже приоритет подписки.

В примере ниже рассылки в рамках ресурса “Website” будут отправляться на адрес `test2@example.com`, так как подписка с этими контактными данными имеет наивысшей приоритет — 0.



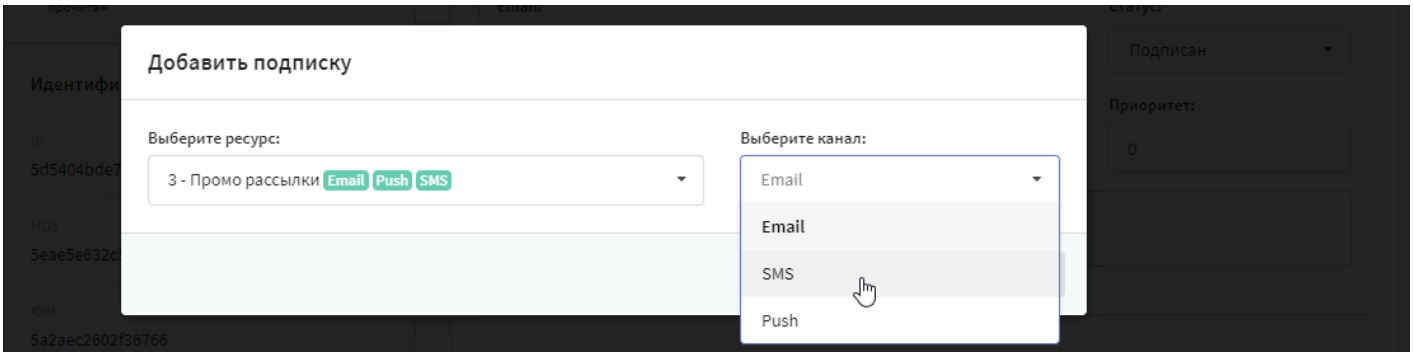
Автоматическое повышение приоритета можно настроить в ресурсе:



CAUTION

Обратите внимание, опция “Повышать приоритет для новых подписок” работает только в случае, если при добавлении новой подписки явно не указан приоритет. Если приоритет новой подписки указывается явно (например, в API-запросе), то приоритеты существующих подписок не изменяются.

Чтобы вручную подписать клиента, нажмите на кнопку **Добавить подписку**, выберите необходимые ресурс и канал связи. После этого введите контактные данные и сохраните изменения.



TIP

Процесс подписки (opt-in) бывает двух видов:

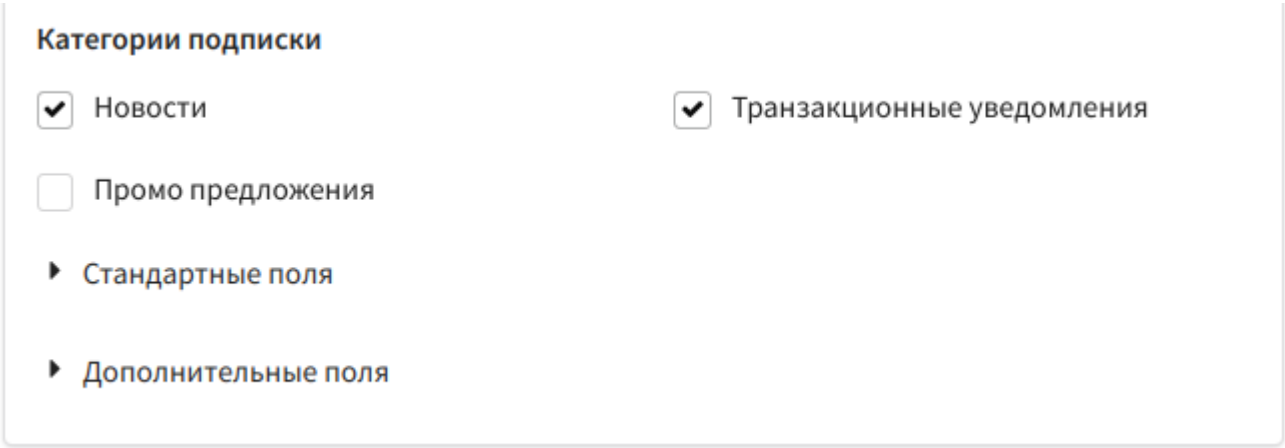
- **Single opt-in (SOI)** — без подтверждения согласия на получения контента. Достаточен факт получения издателем контактных данных (создание подписки на ресурс в Altcraft).
- **Double opt-in (DOI)** — факта создания подписки недостаточно. Клиентам обязательно надо отправить письмо с подтверждением согласия на получение сообщений. Такой процесс позволяет исключить из маркетинговых рассылок потенциально не заинтересованных клиентов, тем самым повышая их эффективность и улучшая репутацию отправителя.

По **этой ссылке** вы найдёте инструкцию по организации Double opt-in подписок в Altcraft.

Категории подписки

Подписка на ресурс может содержать несколько категорий. Категории позволяют персонализировать рассылки. Клиент будет получать сообщения только в рамках тех категорий, на которые подписан.

В отличие от подписки на ресурс, подписка на категорию имеет только 2 статуса: “Подписан” и “Отписан”. Чтобы подписать профиля на категорию, поставьте галочку в соответствующем поле. Также статусом подписки на категории можно управлять через API-запросы.



Поля подписки на канал

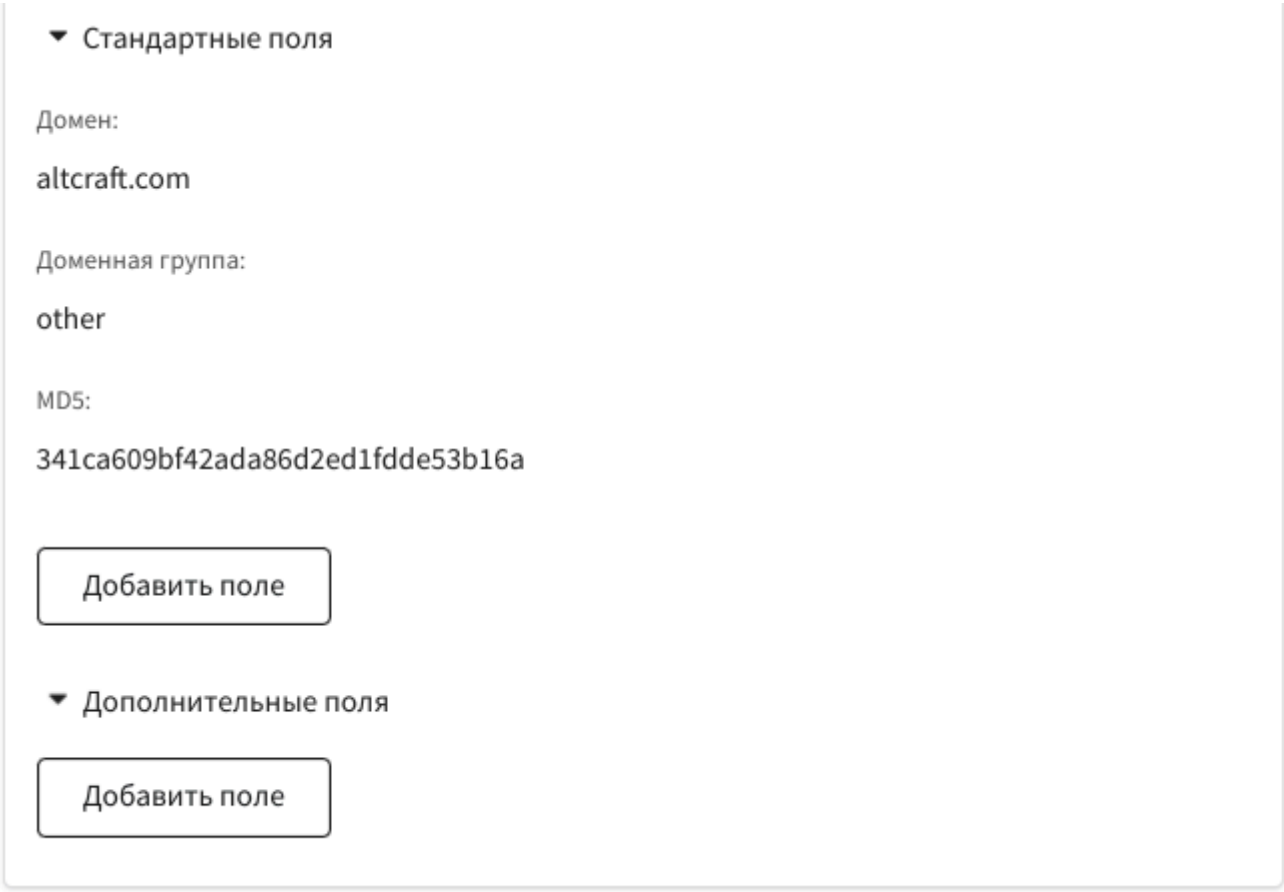
Данные о подписке профиля на определенный канал хранятся в **стандартных** и **дополнительных полях**.

Возможно добавление других стандартных полей, которые будут содержать информацию об устройстве и приложении профиля:

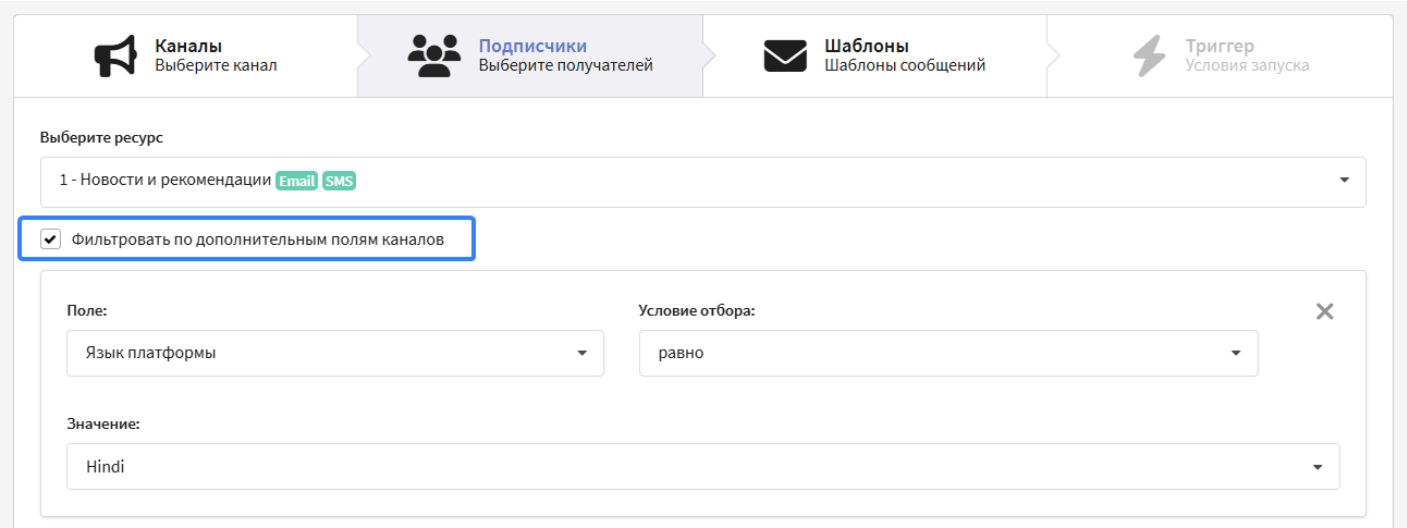
- Тип устройства (Desktop, Mobile, Other)
- Модель устройства
- Имя устройства
- Платформа (Windows, Linux, Mac, Opera, iOS, BlackBerry, Android)
- Версия платформы
- Часовой пояс платформы
- Язык платформы
- Реклама разрешена (Истина или Ложь)
- Рекламный идентификатор
- Идентификатор приложения
- Идентификатор установки приложения
- Версия приложения
- Браузер (Bots, Internet Explorer, Firefox, Safari, Facebook*, Chrome, Opera, Outlook, Thunderbird)
- Версия браузера

Чтобы добавить новое поле, нажмите кнопку **“Добавить поле”** и выберите поле в списке, а затем укажите его значение.

Для хранения дополнительной информации о подписке на канал используйте дополнительные поля. Они создаются в настройках ресурса.



Вы можете использовать их для фильтрации получателей триггерной рассылки. Это позволит сделать рассылку более персонализированной.



Лимит подписок

Для каждого профиля существует лимит подписок. В профиле может быть не больше 200 подписок, либо не больше 20 подписок на один ресурс в канале. Если в профиле достигнуто максимальное количество, то новая подписка вытеснит одну из старых.

В первую очередь вытесняются следующие категории подписок:

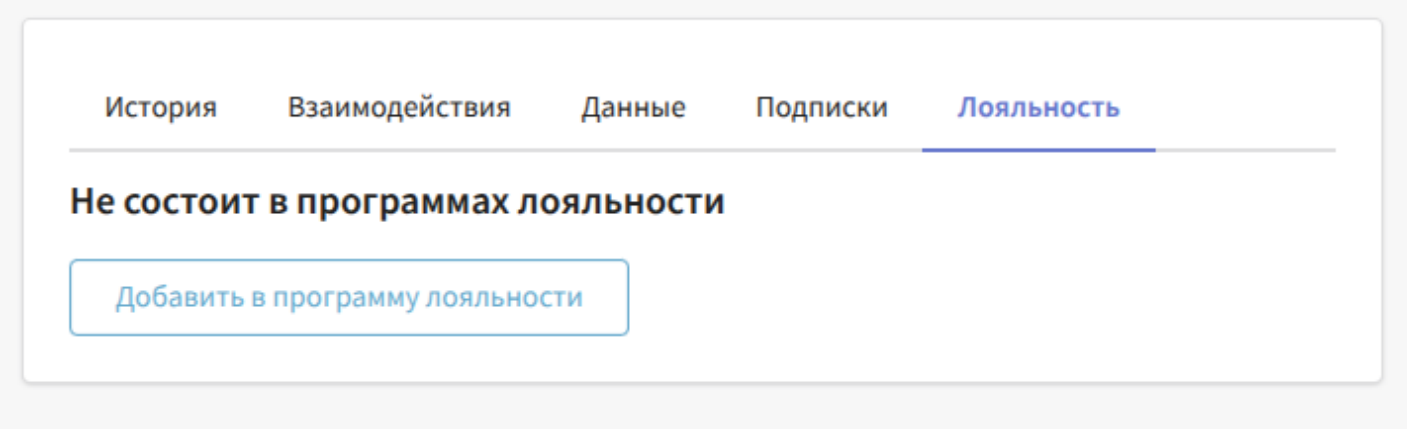
- **Все удалённые подписки**
- **Все подписки со статусами:**
    - Невалиден

    - Hard bounced
    - Complainer
    - Приостановлен
    - Не подтверждён
- **Подписки с наиболее низким приоритетом**
- **Наиболее старые подписки по дате создания**

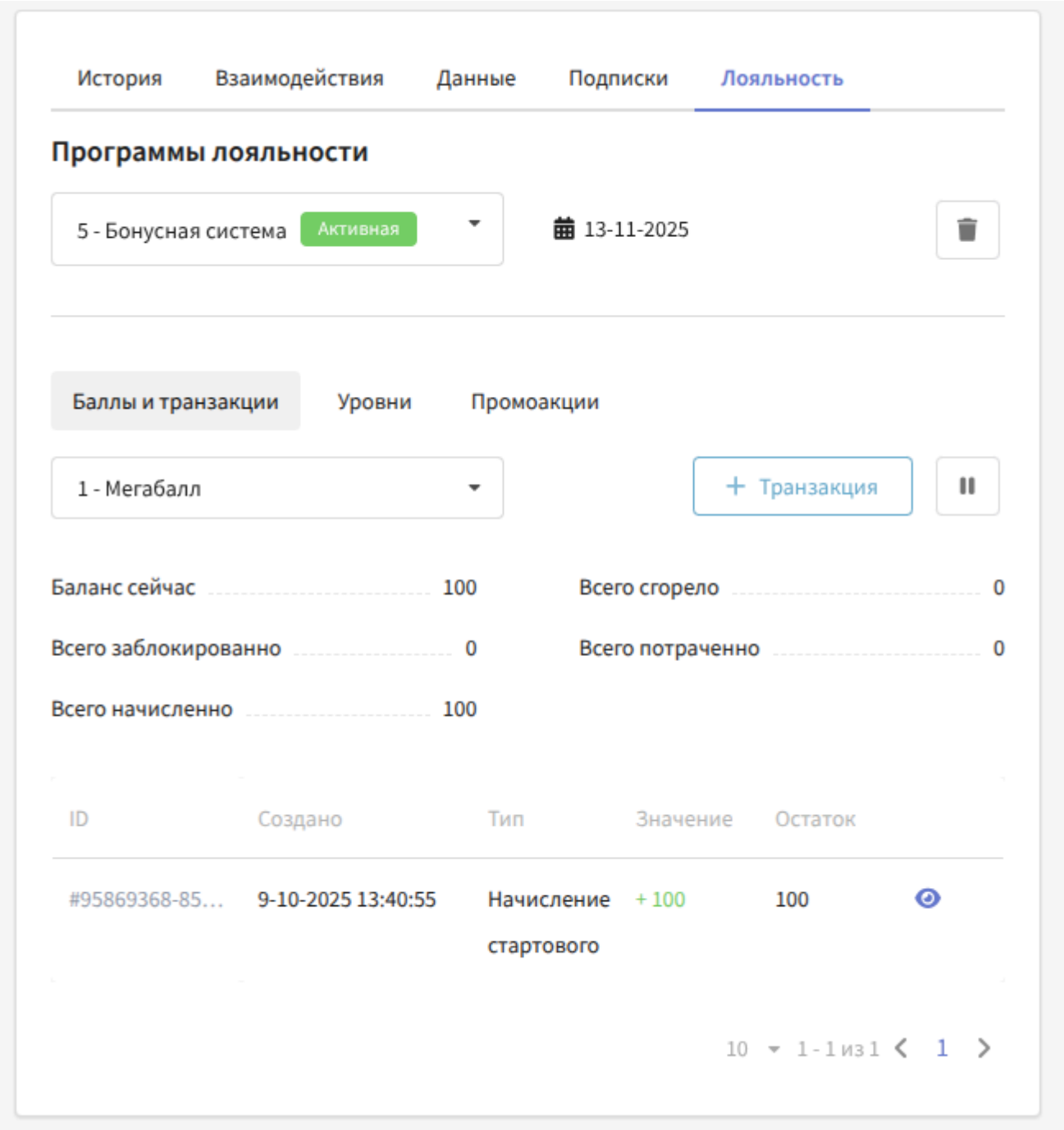
Если попытаться добавить вытесненную подписку, платформа посчитает её за новую и заново внесёт её в список подписок.

Программы лояльности

В этом разделе отображается информация об участии профиля в программах лояльности. Если профиль не состоит в программе, его можно зарегистрировать в ней:



Если же профиль уже зарегистрирован в одной из программ лояльности, то на этой вкладке отобразятся баллы и транзакции профиля, его уровень в ПЛ и участие в промоакция программы. Отсюда вы также можете провести транзакцию баллов для профиля:

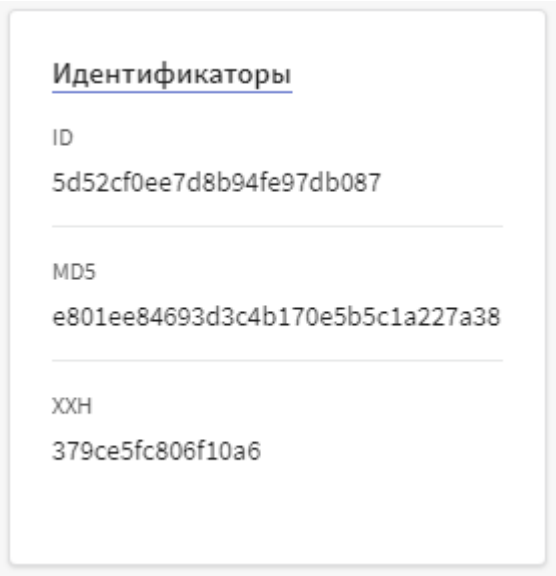


Идентификаторы

Каждый профиль подписчика в Altcraft получает уникальный для всех баз данных **идентификатор профиля (profile_id)**.

Также каждому профилю присваиваются уникальные **MD5** и **XXH** идентификаторы.

В рамках одной базы данных для профиля является уникальным стандартное поле **Email**. Однако оно не является обязательным для профиля. Если при импорте профиля не передать адрес электронной почты в стандартное поле, то будет создана системная запись в виде `profile_id@localhost`.

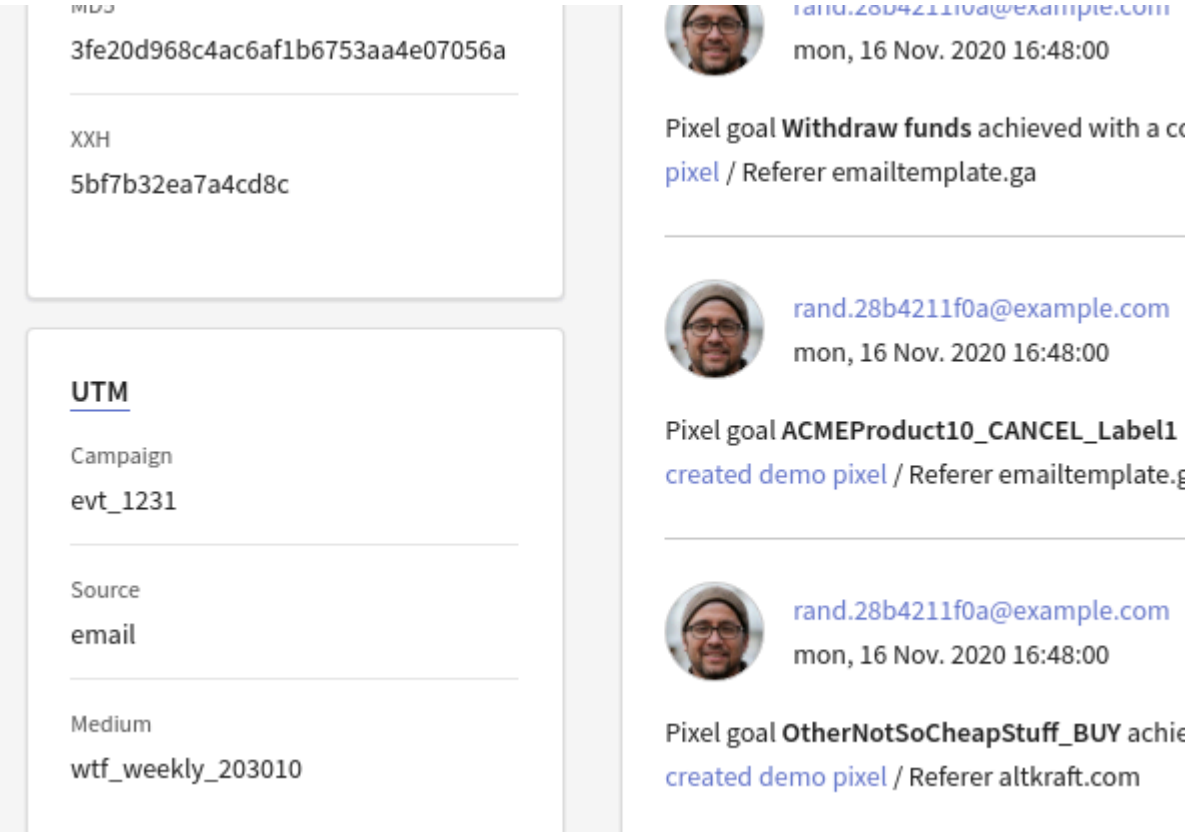


UTM

UTM-метки обновляются в профиле подписчика в следующих случаях:

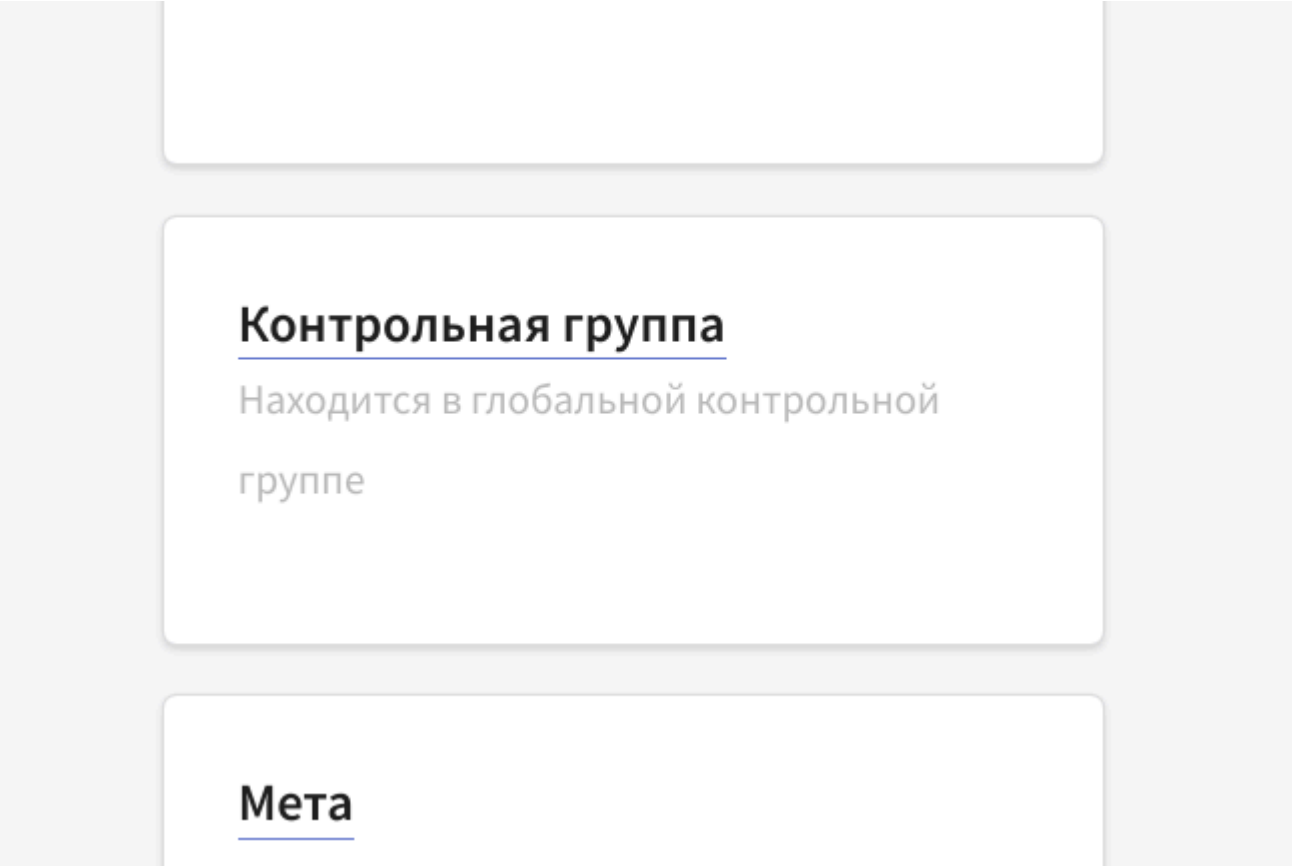
- профиль перешел по ссылке с utm-метками в шаблоне сообщения;
- профиль заполнил форму с настроенными utm-метками (в форме активирована опция “Обновлять данные UTM в профиле”).

Обратите внимание, что UTM-метки добавляются только в том случае, если при публикации формы мы активировали опцию **“Обновлять данные UTM в профиле”**.

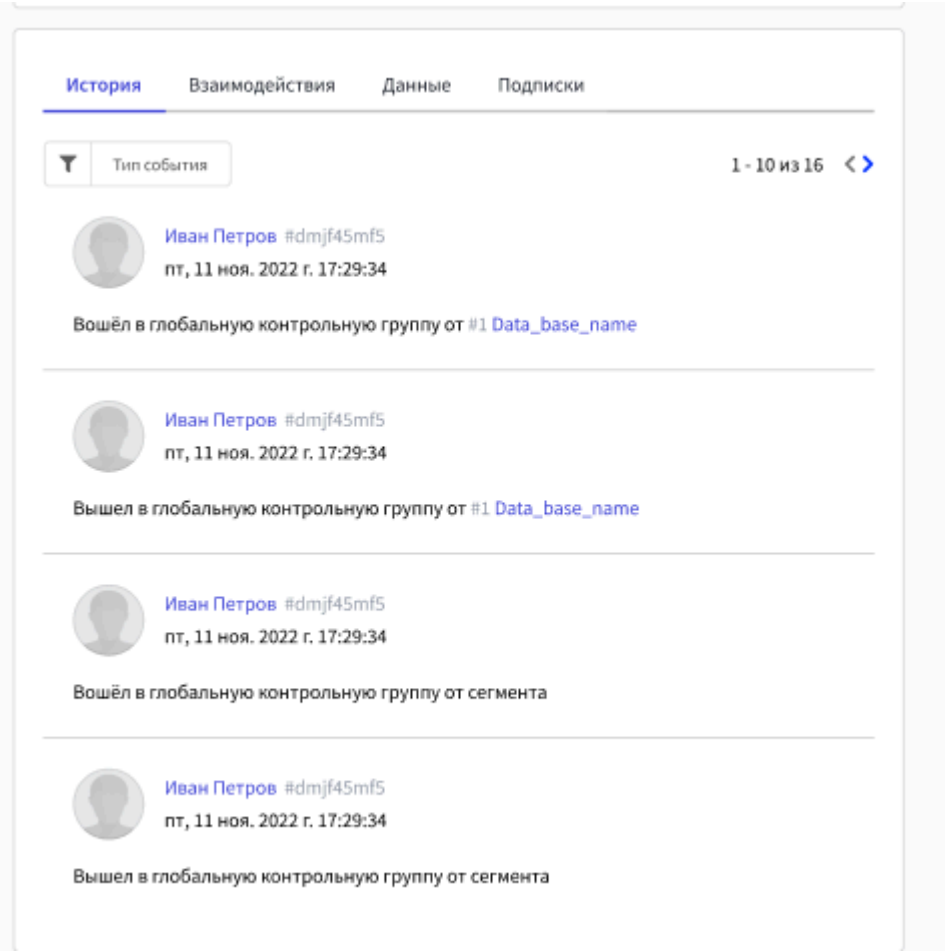


Участие в контрольной группе

Если профиль клиента находится в данный момент в глобальной контрольной группе, соответствующая пометка появится в его профиле:

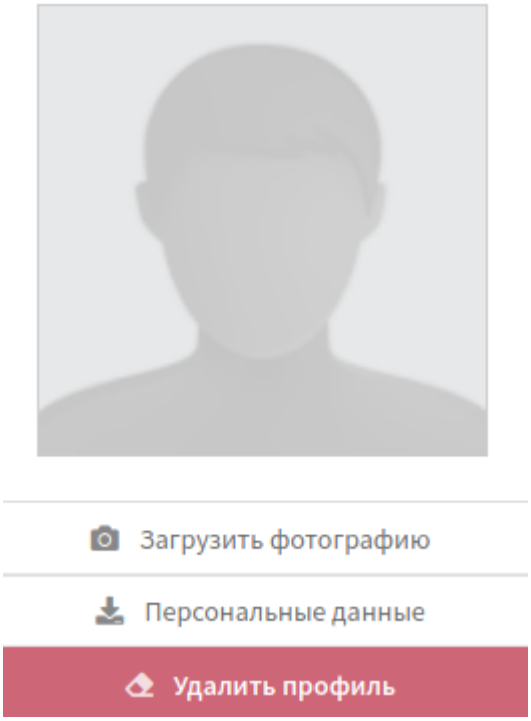


Также в истории действий будет отображено каждое попадание в ГКГ:

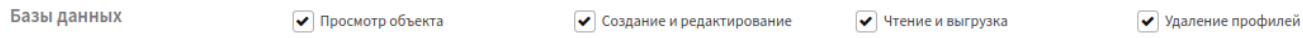


Удаление профиля

Если по каким-либо причинам вы хотите полностью удалить профиль из базы, вы можете это сделать с помощью кнопки “Удалить профиль”. При этом удалится не только профиль, но и его история и связи.



Чтобы пользователь мог удалять профили из базы, в роли, которая на него назначена, должна быть активирована соответствующая настройка. Подробнее про настройку ролей можно прочитать здесь.



Также профиль может быть удален с помощью API запроса подробнее здесь.

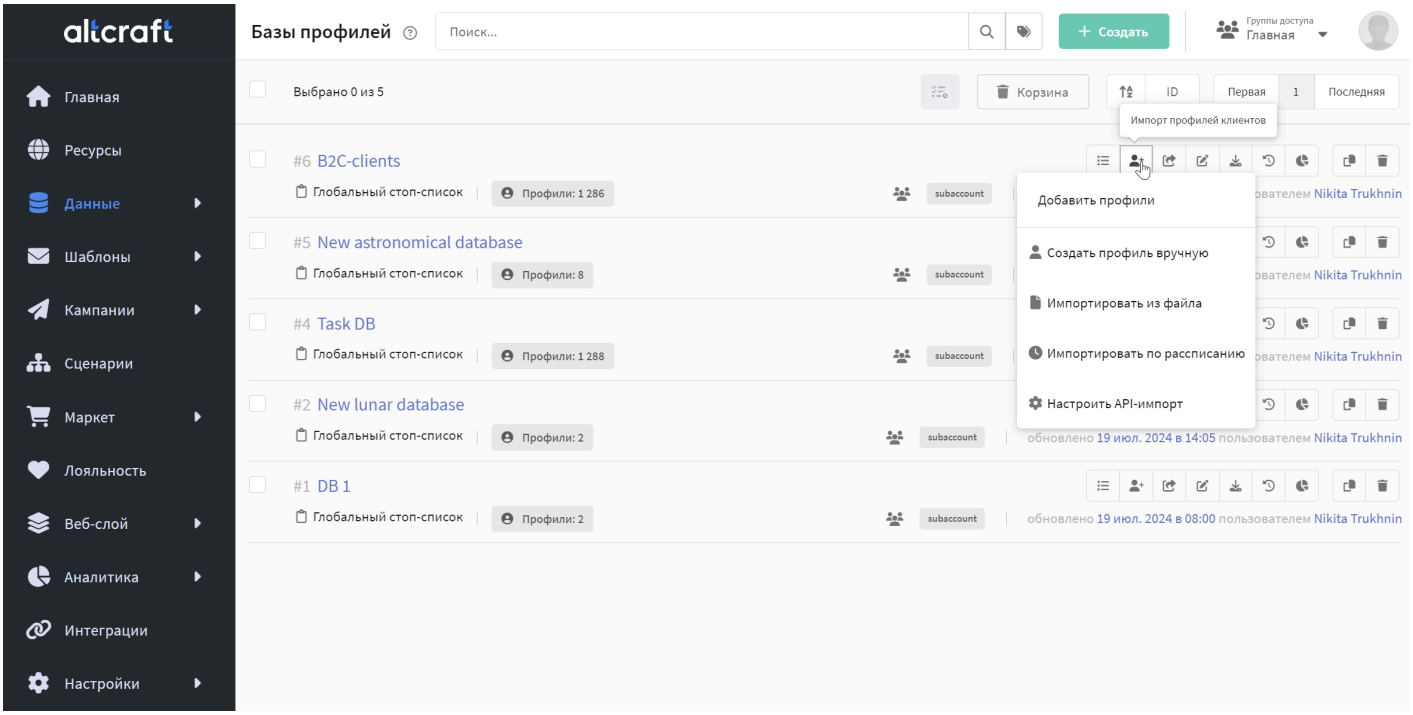
**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Импорт профилей клиентов и обновление данных

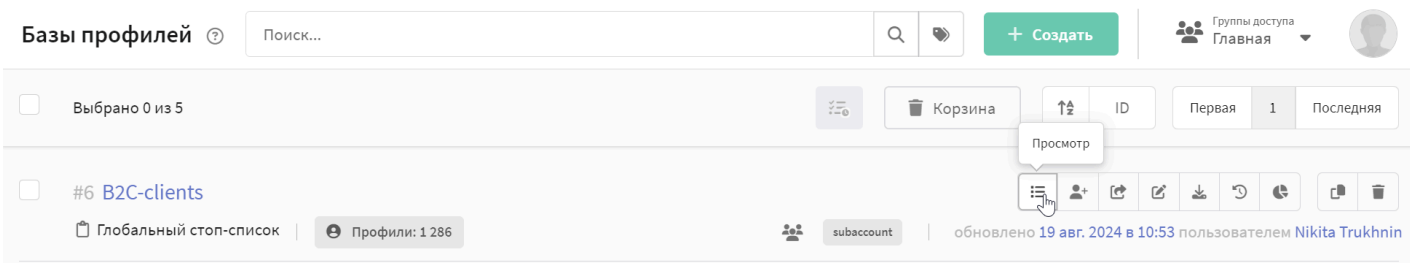
Начало работы

В платформе Altcraft перейти в окно импорта данных можно несколькими способами:

1. В разделе **Данные** выберите **Базы профилей**. Нажмите на “**Импорт профилей клиентов**” напротив нужной вам базы профилей и выберите подходящий способ загрузки данных:

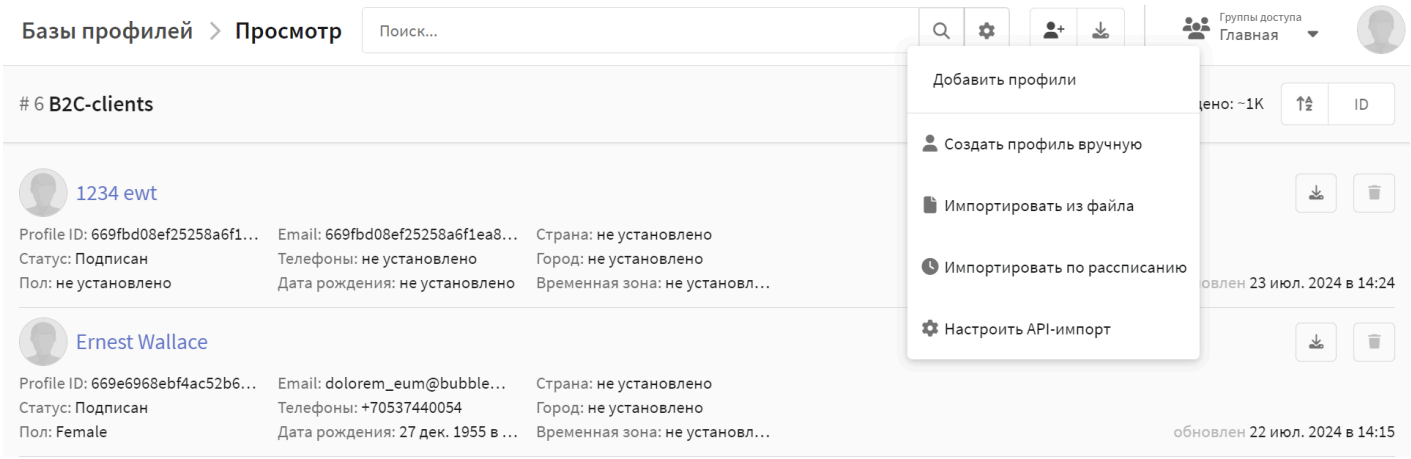


2. Выберите нужную вам **Базу профилей** и нажмите на **Просмотр**:



Нажмите на значок импорта

в верхней части экрана и выберите подходящий вам способ импорта:

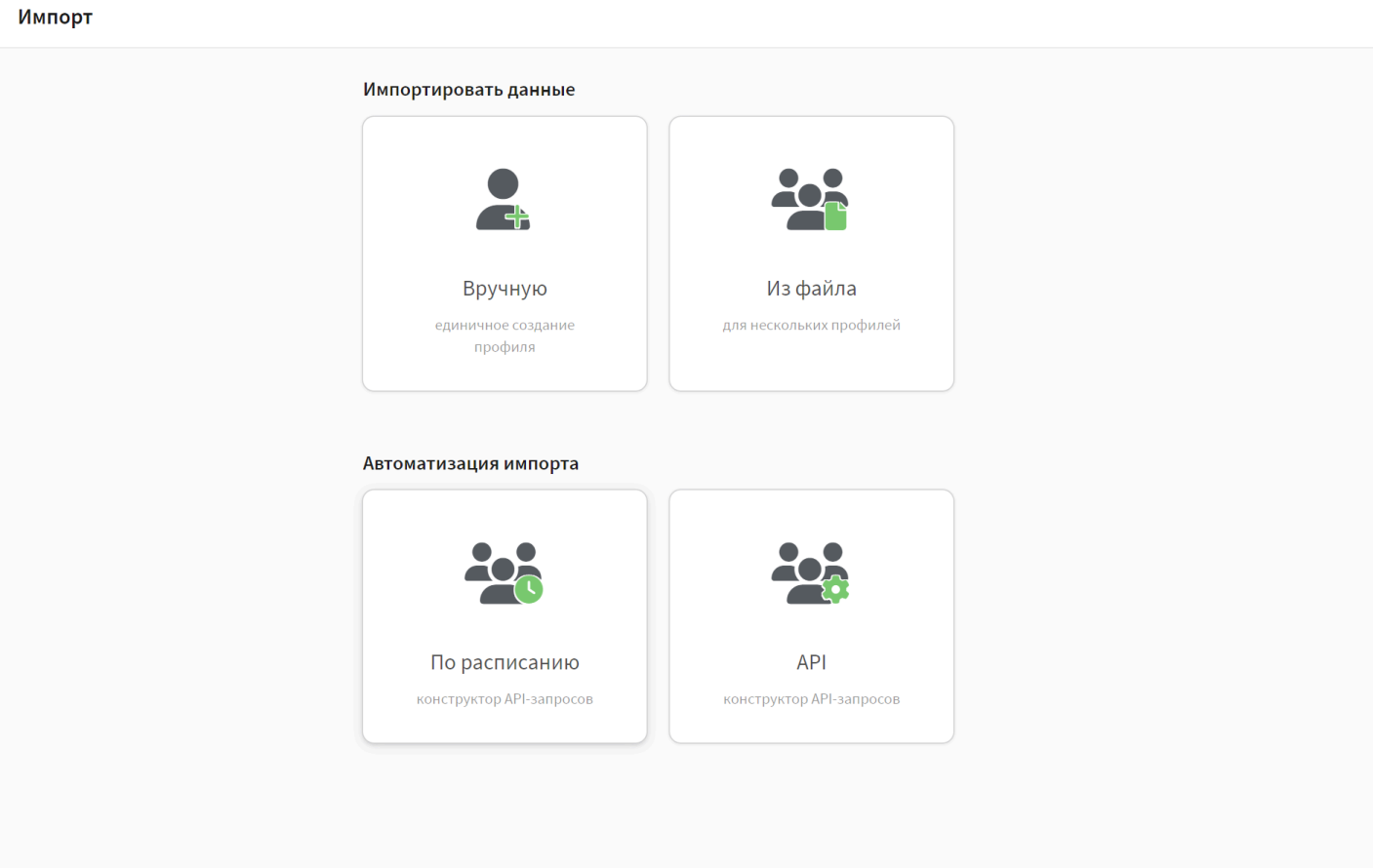


3. Из главного меню перейдите в раздел **Данные —> Импорт**. В открывшемся окне выберите базу для импорта.

Способы импорта

Импорт и обновление данных клиентов в базах профилей Altcraft можно организовать различными способами в зависимости от объема клиентской базы, источников новых подписчиков (например, напрямую или через CRM), а также технических возможностей вашей компании. Доступно 4 способа загрузки данных:

- **Вручную** — создание и редактирование профилей по одному.
- **Из файла** — загрузка данных клиентов из файла-таблицы (например, CSV, Excel или TXT).
- **Через API** — настраиваемый импорт одного или нескольких профилей с помощью **API**.
- **Расписание** — автоматический импорт данных клиентов из внешней SQL базы данных по заданному расписанию.



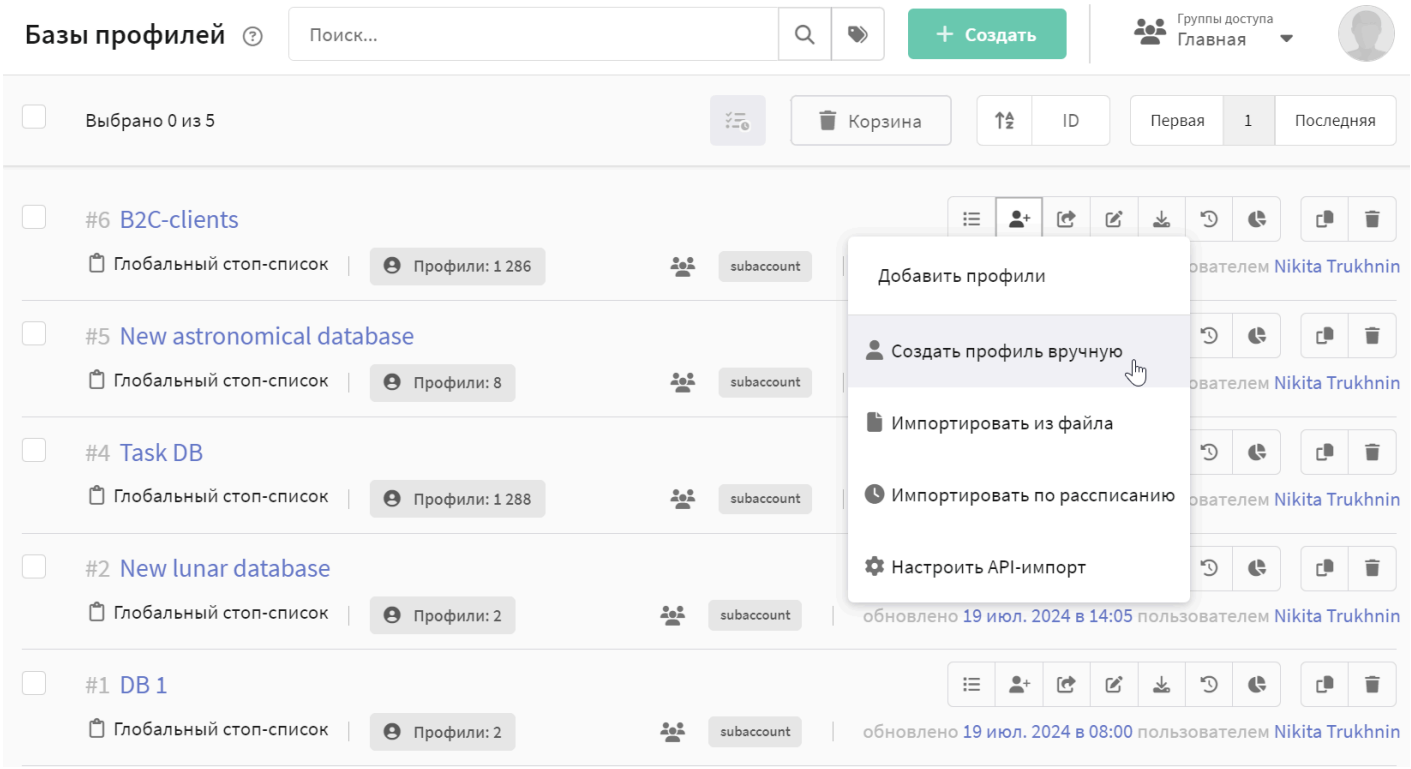
TIP

Помимо перечисленных способов получения клиентских данных, в Altcraft Platform есть готовые интеграции с внешними сервисами:

- **Tilda** — интеграция с формами сбора данных конструктора сайтов Tilda.
- **LPGenerator** — интеграция с формами сбора данных конструктора лэндингов LPGenerator.
- **Yandex AppMetrica** — интеграция с метрикой приложений от Яндекса.

Создание профиля клиента вручную

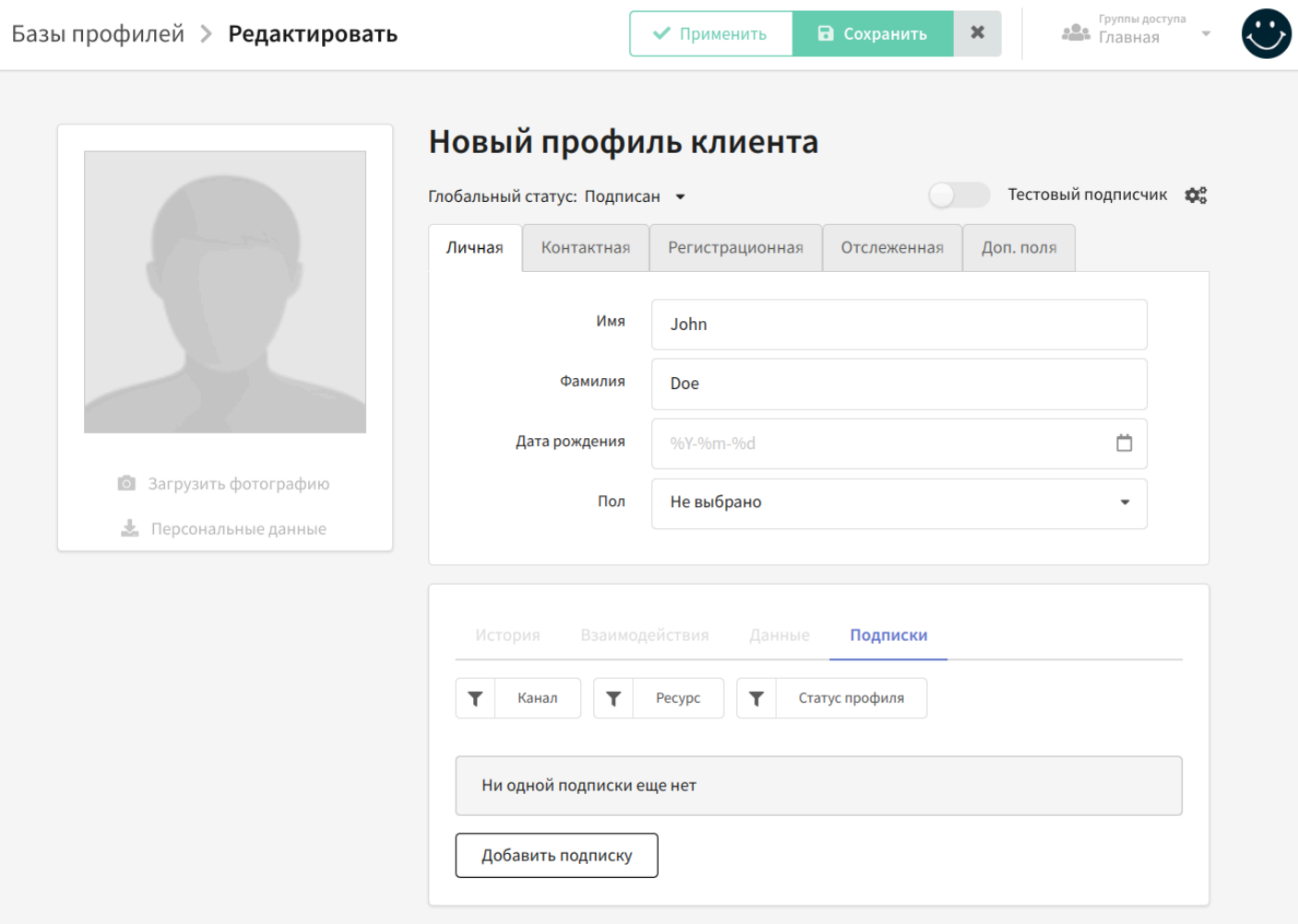
Выберите способ импорта — **“Создать профиль вручную”**:



Откроется **форма создания и редактирования профиля**. На этапе создания профиля вы можете добавить личные, контактные, регистрационные и отслеженные данные. Эти данные записываются в **стандартные поля базы**.

Значение **дополнительных (кастомных) полей** указываются на вкладке “Доп. поля”. Обязательные поля отмечены красной звёздочкой. Создать дополнительные поля, а также установить обязательные вы можете на **странице редактирования базы данных**.

Ниже на вкладке **Подписки** вы можете сразу при создании **подписать клиента** на тот или иной **ресурс**.

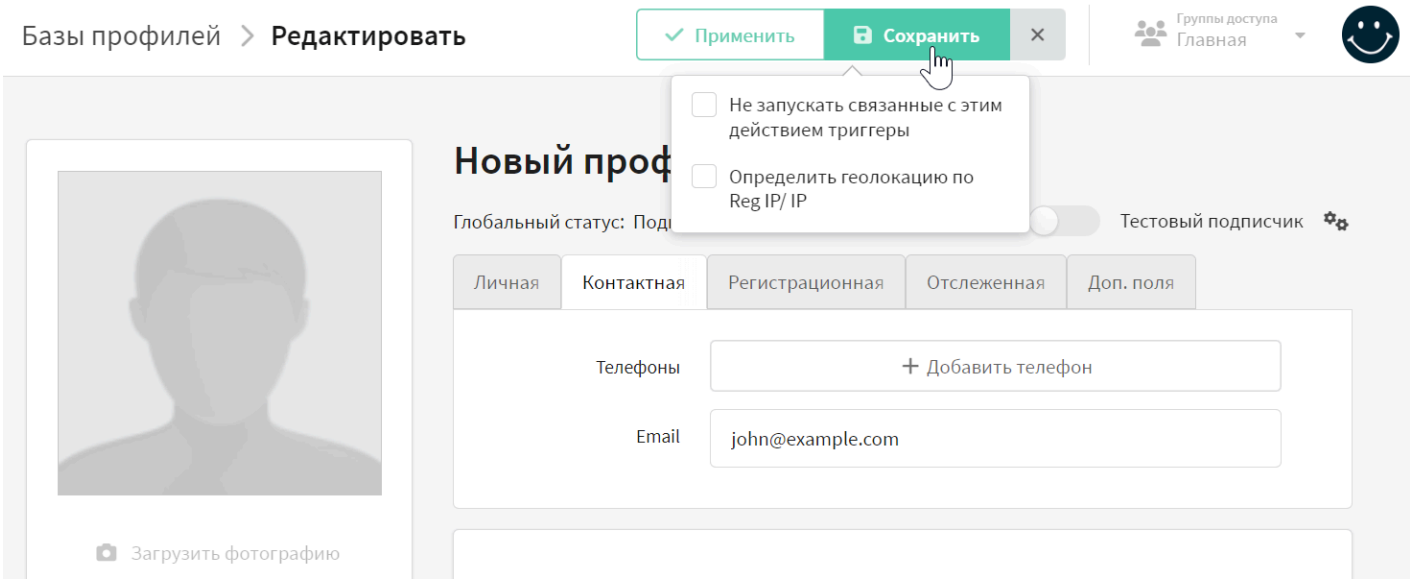


CAUTION

Поля **Email** и **Телефоны** на вкладке **Профиль** не используются для коммуникации с клиентом. В них вы можете записать базовые контакты, чтобы использовать их потом в подписках на ресурсы. Или можете оставить поля пустыми, а контактные данные записать напрямую в **подписки**.

Поле **Email** используется по умолчанию для идентификации профиля наряду с **profile_id**. Если при импорте профиля не передать адрес электронной почты в это поле, будет создана системная запись в виде `profile_id@localhost`.

При сохранении нового профиля можно отключить отправку для него триггерных оповещений, а также включить автоматическое определение геолокации клиента по данным в полях `Reg IP` (IP адрес регистрации) и `IP` (последний известный IP адрес):

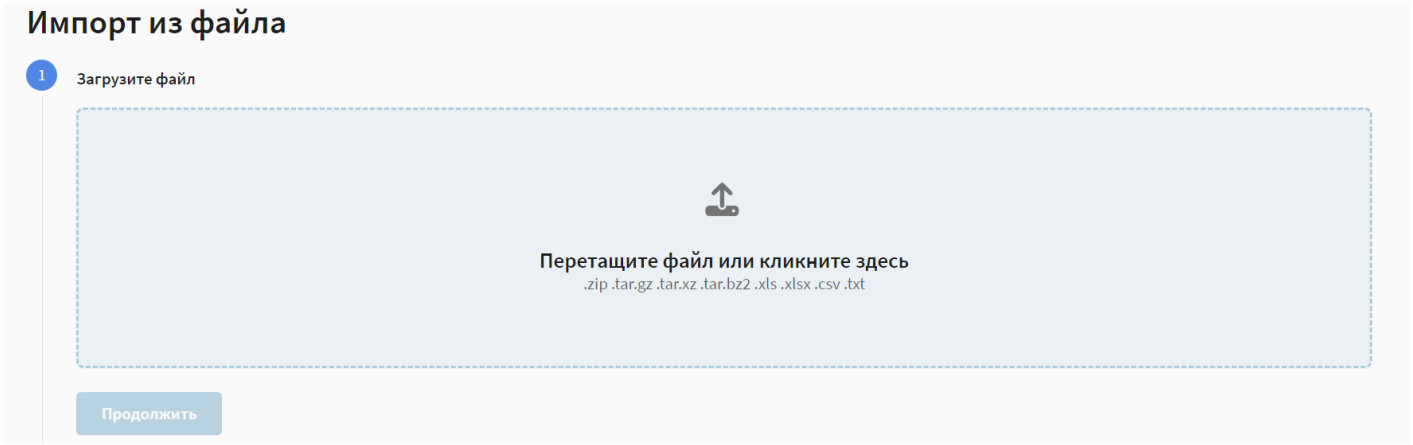


Импорт из файла

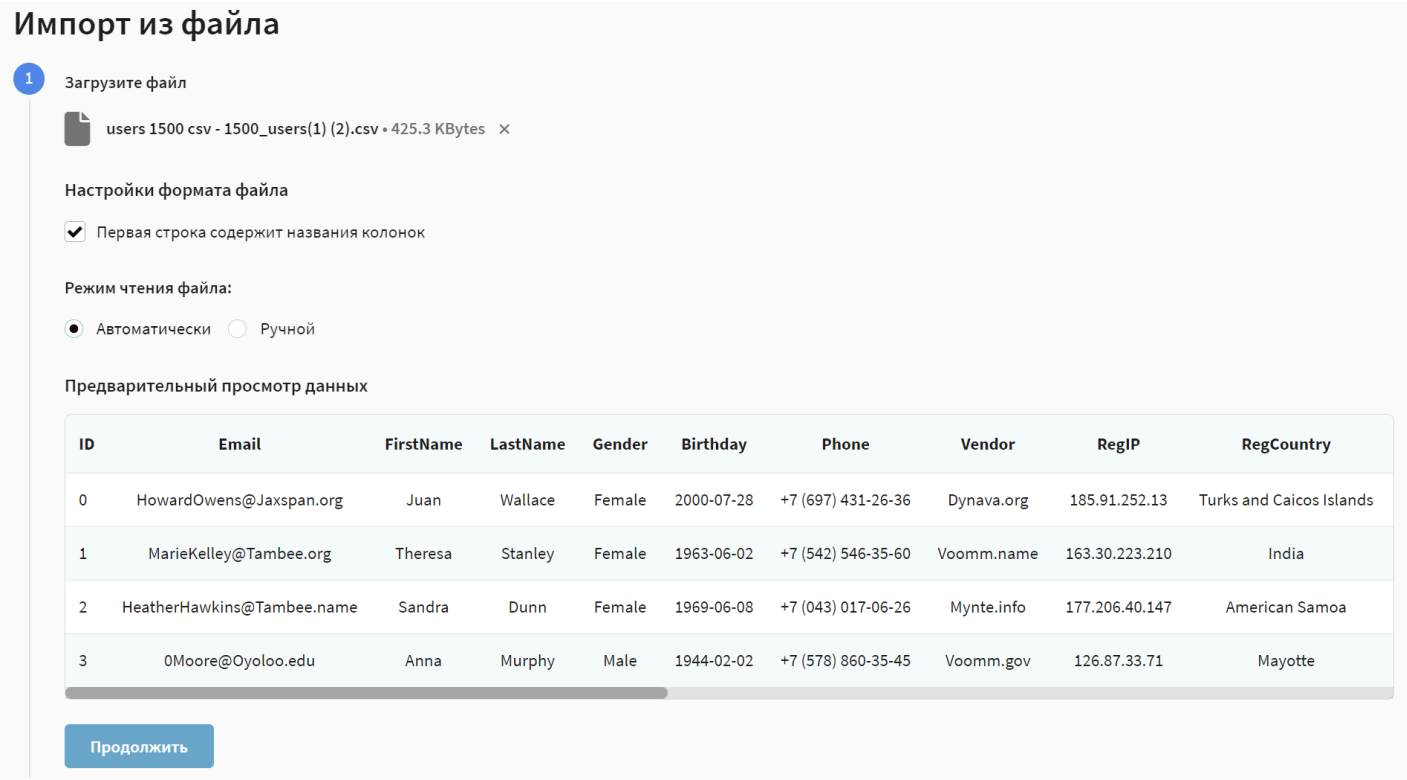
Для загрузки большого количества профилей, а также для обновления существующих данных используется импорт из файла. Это могут быть **.csv** таблицы и **.txt** файлы в CSV формате, **.xls** и **.xlsx** таблицы данных. Файл может быть помещён в архив **.zip** или **.tar.gz**.

Загрузка файла

Чтобы добавить новый файл, перетащите его мышью в область загрузки или кликните по ней и выберите файл. Чтобы **удалить файл**, нажмите на крестик напротив его названия.

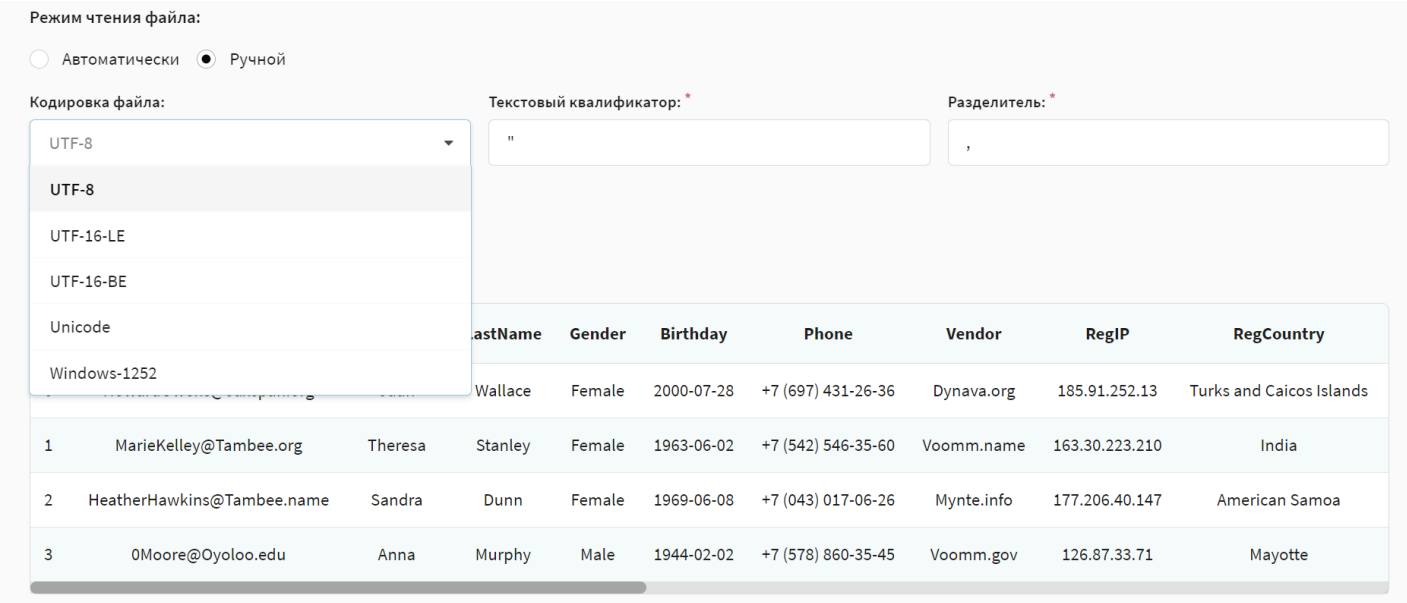


Поставьте галочку в окне “Первая строка содержит название колонок”, если первая строка в вашем файле с профилями служит для обозначения колонок.



Выберите **режим чтения файла**. Доступно два режима: **автоматический** и **ручной**.

В **автоматическом режиме** используется кодировка UTF-8. Если ваш файл имеет другую кодировку, то вы можете перейти в **ручной режим чтения** и выбрать подходящий вам вариант:



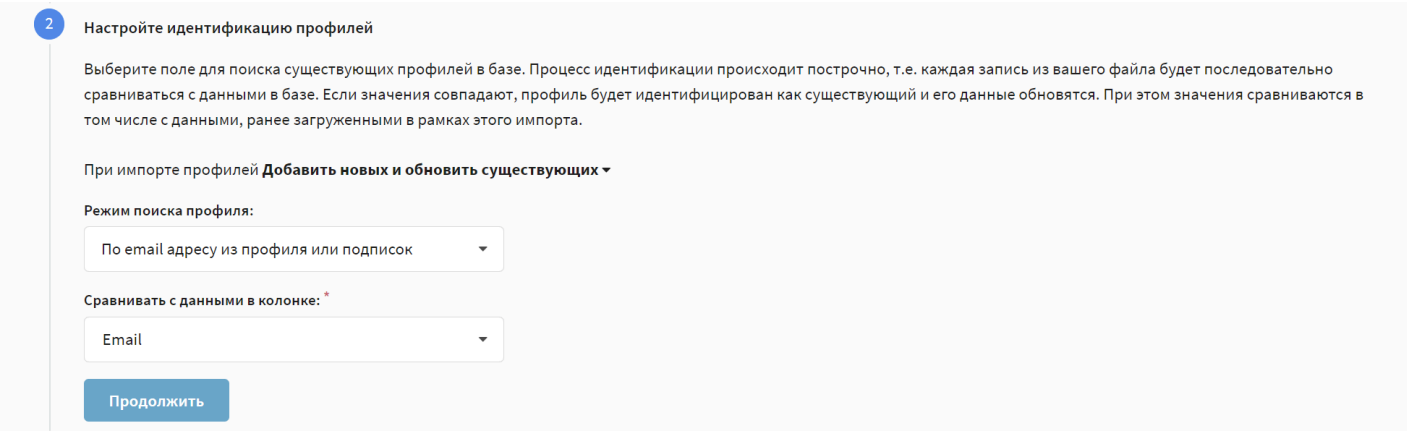
- **Текстовый квалификатор** — это символ, который используется для обозначения начала и конца текстовых данных внутри ячейки таблицы. Квалификаторы необходимы, когда данные внутри ячейки содержат символы, которые могут быть восприняты как разделители или другие специальные символы.
- **Разделитель** — это символ, который используется для разделения данных на отдельные поля (ячейки) в строке таблицы. Разделители применяются для указания границ между столбцами в таблице.

Если вы изменили настройки в ручном режиме, нажмите кнопку **“Предпросмотр”**, чтобы проверить, что платформа корректно прочитает файл.

Идентификация профилей

Выберите **способ импорта** — он будет зависеть от того, добавляете вы новые записи или обновляете уже существующие в базах профилей Altcraft:

- Добавить новых и обновить существующих
- Только добавить новых, не обновлять существующих
- Только обновлять существующих, не добавлять новых



Режим поиска профиля определяет, по каким данным из вашего файла Altcraft будет искать существующие профили. Это необходимо для обновления данных клиентов и избежания дублирующихся записей. Доступны следующие режимы:

- **По email адресу из профиля или подписок** — по полю профиля **Email** или по email адресам из подписок.
- **По email адресу из профиля** — только по полю профиля **Email**.
- **По телефону из полей данных или подписок** — по полю профиля **Телефоны** или по телефонам из SMS подписок.
- **По дополнительному полю** — по пользовательскому полю базы данных. Например, ID клиента в CRM компании.
- **По подписке** — по данным подписки.
- **По идентификатору профиля** — по уникальному ID профиля (только при обновлении данных).

TIP

Необходимо учитывать следующие особенности матчинга:

- Значение поля для поиска должно быть **уникально** для каждого профиля в базе, в ином случае платформа не сможет однозначно определить искомый профиль
- Для одной базы профилей следует использовать **только один** режим поиска. Это необходимо для того, чтобы избежать создания дублирующихся профилей в базе.
- Если вы собираетесь пользоваться матчингом по **дополнительному полю**, вам нужно обратиться к администратору платформы для создания **индекса** этого поля. Благодаря индексу поиск профиля в базе будет занимать значительно меньше времени.

Настройка полей

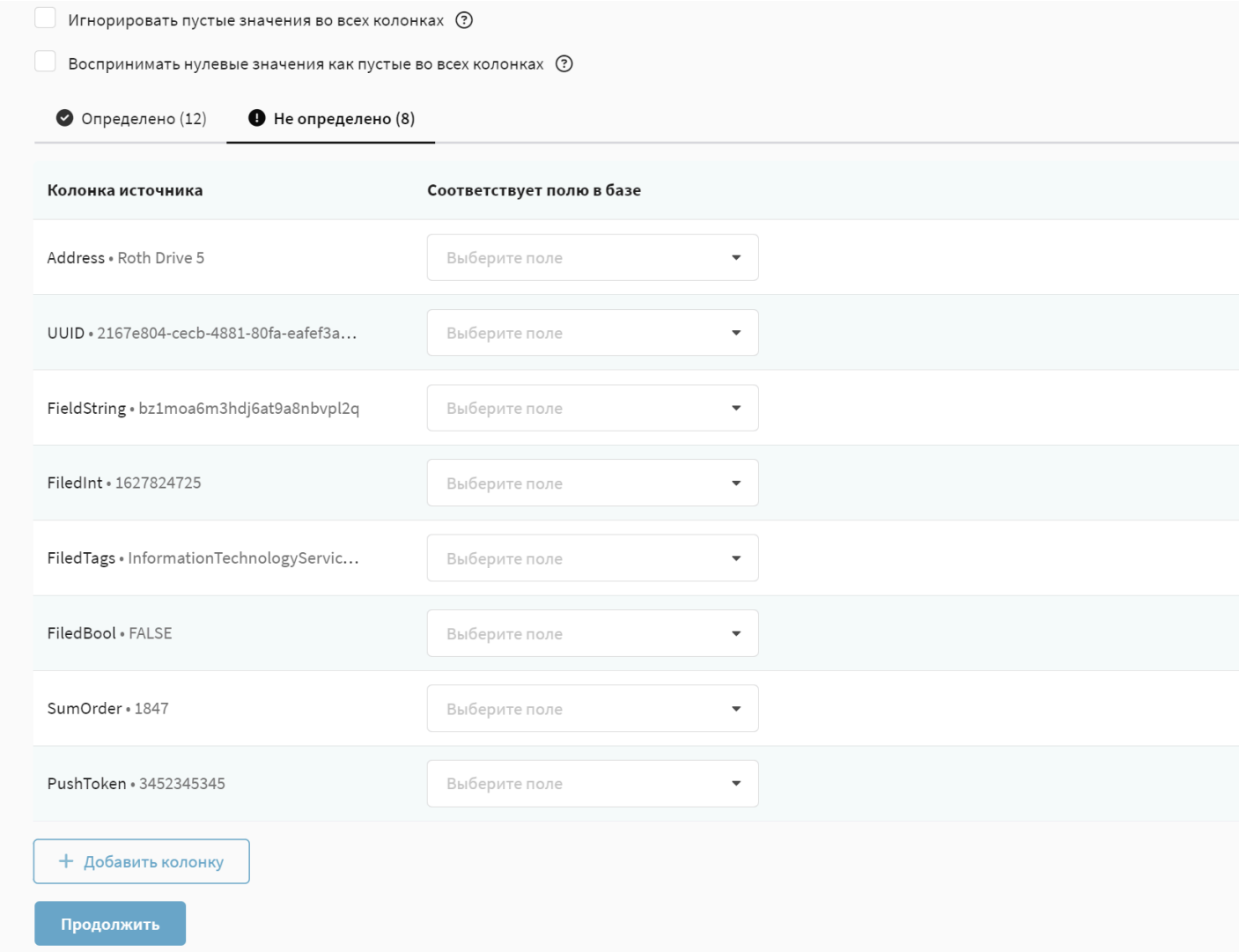
TIP

Для рассылок платформа использует контактные данные, указанные в подписках профиля на ресурсы. Вы можете также записать email-адрес и номер телефона клиента в основные поля профиля, но эти данные

будут использоваться только для поиска профиля в базе. Если идентификация клиентов в вашей базе настроена по данным CRM или другим уникальным идентификаторам, то **Email** и **Телефоны** в полях профиля можно не указывать.

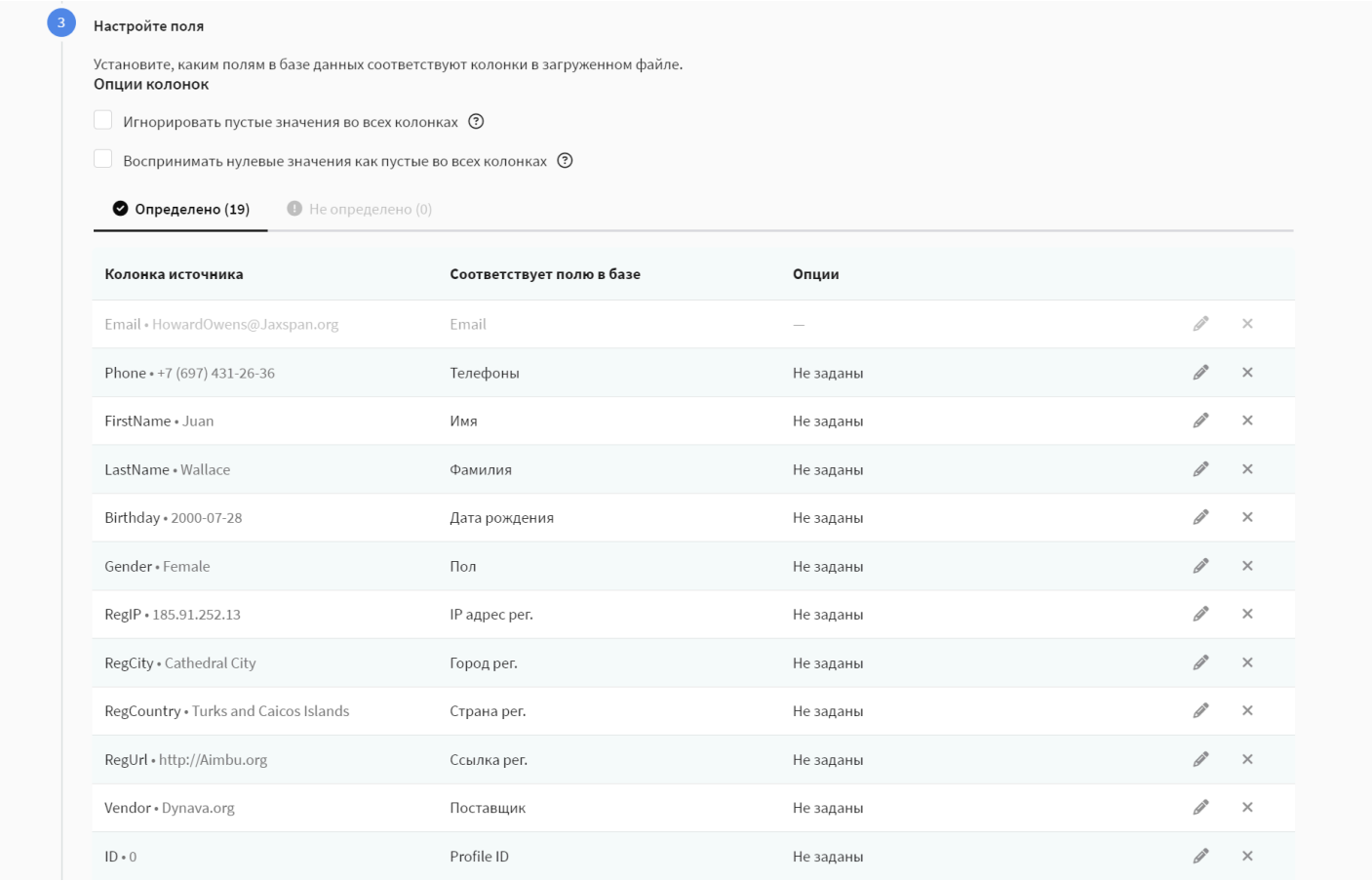
Перейдите к определению соответствия колонок в базе профилей Altcraft с колонками данных в вашем файле-таблице. Altcraft определит колонки автоматически, если их названия подойдут под системные, однако, если ваши колонки не соответствуют системным, вы можете настроить их вручную. Для этого перейдите в столбец **“Не определено”**:

- **Колонка источника** — выберите колонку вашего файла для получения данных. По каждой колонке будет отображаться первая запись — так вы можете точнее определить тип данных.
- **Соответствует полю в базе** — выберите поле профиля, в которое нужно записать данные клиента. Можно выбрать стандартные и пользовательские (дополнительные) поля **базы профилей**.



После того как вы выберите соответствие для неопределившейся колонки, она автоматически будут перенесена в столбец **“Определено”**. Если вы установили неправильное соответствие, то нажмите на крестик справа от нужного вам поля, тогда оно снова окажется в столбце **“Не определено”**.

Данные из неопределенных колонок не будут импортированы в базу.



Дополнительные настройки полей:

- **Игнорировать пустые значения во всех колонках.** Опция используется при обновлении данных профилей, которые уже есть в базе. Например, в поле профиля “vendor” уже записано значение “test.com”, но сейчас вы импортируете файл, где поле “vendor” у этого клиента пустое. Платформа перезапишет поле, подставив пустое значение. Чтобы избежать этого, включите опцию “Игнорировать пустые” для соответствующей колонки. Тогда значение в поле профиля “vendor” останется прежним.
- **Воспринимать нулевые значения как пустые во всех колонках.** Если в вашем файле-источнике неизвестные о клиенте данные записаны с помощью таких значений, как `null`, `nil`, `none` или `undefined`, платформа также может обработать эти значения как пустые. Для этого вам необходимо одновременно актировать поля “Воспринимать нулевые значения как пустые” и “Игнорировать пустые значения” рядом с той колонкой, где указано неизвестное значение. В таком случае платформа не обновит поле. При этом регистр null, nil, none и undefined неважен. Значения `"0001-01-01 00:00:00 +0000 UTC"` и `"0001-01-01T00:00:00Z"` тоже могут быть обработаны как пустые.

Опции

При настройке импорта данных из файла для каждого типа поля доступны определённые опции, которые определяют, как данные будут обрабатываться и сохраняться в системе:

Опция	Описание	Тип поля
Игнорировать пустые значения	Предотвращает перезапись данных профиля пустыми значениями при импорте, сохраняя существующую информацию, если в импортируемом файле соответствующие поля пусты.	Все

Опция	Описание	Тип поля
Считать нулевые значения пустыми	Платформа будет обрабатывать значения типа null, nil, none или undefined как пустые, если активировать данную опцию.	Все
Автоматическая коррекция регистра	Преобразовывает буквы в выбранный регистр. Например: Все буквы строчные, Каждое слово начинается с заглавной буквы или Все буквы заглавные.	Имя, Фамилия, Подписки (тип поля - Строка)
Добавить телефонный код ко всем значениям	Позволяет добавить международный код в начало всех импортируемых номеров (например, +7). Платформа обрабатывает телефоны только в международном формате, поэтому используйте эту опцию, если в вашем файле телефоны записаны в другом формате.	Телефон
Преобразовать локальный телефонный код	Позволяет заменить часть номера (например, преобразовать код 8 в +7)	Телефон
Выберите формат даты из файла	Позволяет выбрать формат даты, который представлен в импортируемом файле.	Дата
Значение по умолчанию	Предустановленное значение, которое автоматически записывается в поле, если в загружаемом файле для данного поля отсутствуют данные.	Дата
Установить соответствие значений колонок и полей	Если значения в вашей колонке не соответствуют системным значениям платформы, вы можете уточнить их соответствие.	Пол (тип поля - Список)
Увеличить значение поля на значение из колонки	Добавляет значение из импортируемого файла к уже существующему значению в указанном поле базы данных, вместо его замены.	Целое число, Число с точкой
Уменьшить значение поля на значение из колонки	Вычитает значение, указанное в импортируемом файле, из текущего значения в выбранном поле базы, вместо его замены.	Целое число, Число с точкой

Опция	Описание	Тип поля
Округлить число до указанного количества десятичных знаков	Автоматически сокращает числовое значение до заданного количества знаков после запятой во время загрузки данных, чтобы обеспечить точность или соответствие формату. Округление применяется к данным из файла, а не к итоговому значению, которое запишется в поле профиля. Это может привести к погрешностям при использовании опций “Уменьшить на значение из файла” и “Увеличить на значение из файла”. Например, если значение в файле составляет 123.124, то после округления до двух знаков оно будет 123.12. При вычитании или сложении этих данных с числом из профиля может возникнуть небольшая погрешность, например, при сложении 0.1 и 0.2 вместо ожидаемого 0.3 может получиться значение 0.3000000001.	Число с точкой
Разделитель тегов	Символ или набор символов, используемый для разделения нескольких тегов (меток) в одном поле файла, чтобы платформа могла правильно распознать и импортировать их как отдельные значения.	Теги
Добавить или обновить данные по полю объекта	Опция, которая либо создает новое значение для указанного поля объекта, если оно отсутствует, либо обновляет уже существующее значение в базе на основе данных из импортируемого файла. Обратите внимание, подобное обновление сработает корректно только если объекты в массиве имеют одинаковую структуру ([{“**key**”:“value_1”},{“**key**”:“value_2”}]).	Массив объектов
Удалить данные по полю объекта	Опция, которая удаляет объекты с одинаковым значением в указанном поле.	Массив объектов
Импортировать в свойство	Записывает значение из колонки файла в указанное свойство объекта.	Объект

Особенности импорта полей

При импорте данных в базу платформа ожидает определённые форматы значений для каждого типа поля. Все допустимые форматы описаны в **этой статье**. Убедитесь, что данные в вашем источнике соответствуют указанным требованиям, иначе платформа не сможет корректно обработать их.

Для полей типа **Объект** и **Массив объектов** платформа ожидает значение в виде строки, содержащей корректный JSON. Чтобы задать такое значение, необходимо сначала преобразовать JSON-объект в строку, а затем экранировать её по правилам CSV, описанным в **RFC 4180** (двойные кавычки " заменяются на “”).

Пример JSON-структуры:

```
[
    {
        "field1": 123,
        "field2": {"subfield": false},
        "field3": "abc"
    }
]
```

Ожидаемое значение: `"[{""field1"": 123, ""field2"": {""subfield"": false}, ""field3"": ""abc""}]"`

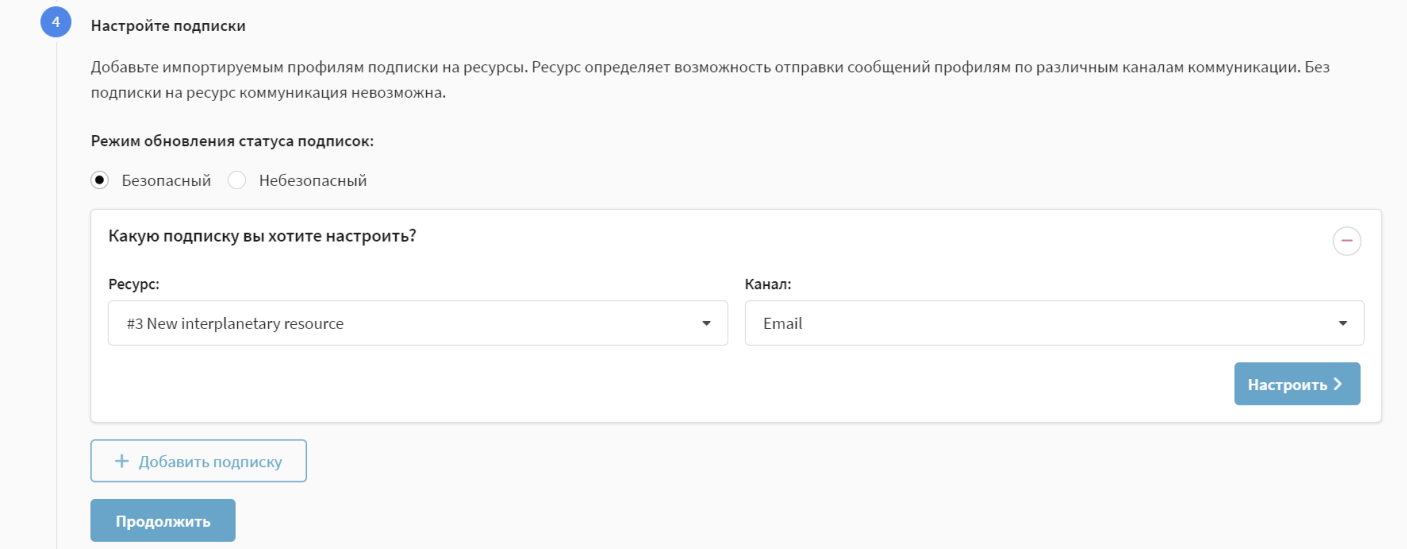
В поле типа **“Список”** можно импортировать сами значения или их индексы. Например, если вы создали в базе список с вариантами “тариф_1”, “тариф_2”, “тариф_3”, то в колонке источника можно указать эти же значения или использовать индексы: 1 — это “тариф_1”, 2 — это “тариф_2”, 3 — это “тариф_3”, 0 — “Не выбрано” (значение по умолчанию).

Обратите внимание, что значения, которые состоят только из цифр, невозможно импортировать напрямую — платформа воспримет их как индексы. Например, если в списке указаны варианты 100 и 200, то при импорте нужно передать соответствующие им индексы: 1 — это 100, 2 — это 200.

Настройка подписок

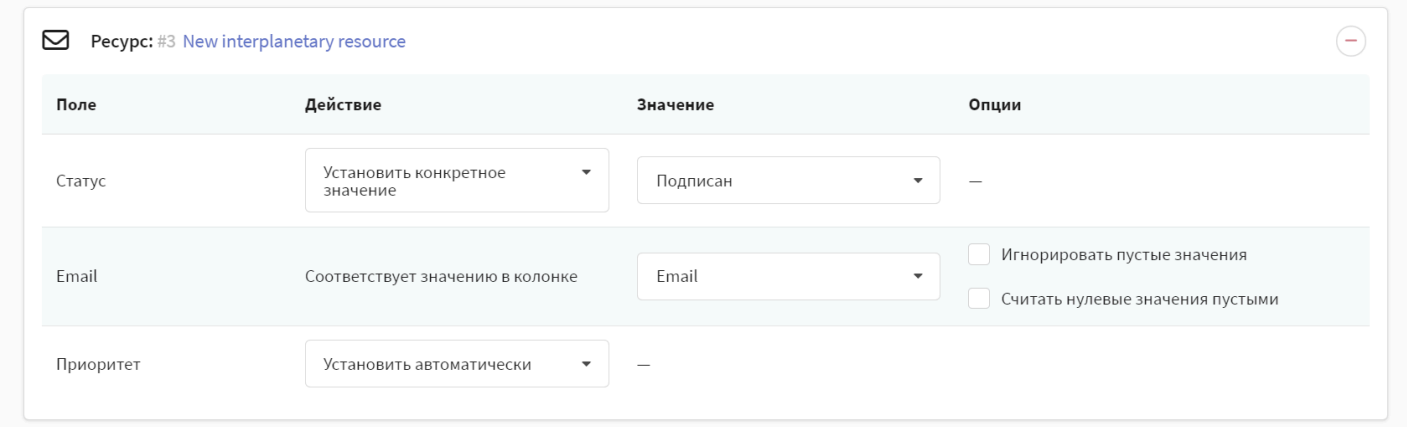
При импорте из файла подпишите своих клиентов на необходимые ресурсы. Ресурс определяет возможность отправки сообщений профилям по различным каналам коммуникации. Без подписки на ресурс коммуникация невозможна. В блоке **Настройте подписки** нажмите кнопку **+ Добавить подписку**.

Вам доступно два режима обновления статуса подписок: **безопасный** и **небезопасный**. При выборе **небезопасного** режима будут обновлены подписки профилей со статусами: Отписан, Жалобщик, Hardbounced, Не валиден. Если у профиля уже есть подписка на выбранный канал в ресурсе с одним из этих статусов, после импорта её статус изменится в соответствии с заданными настройками.

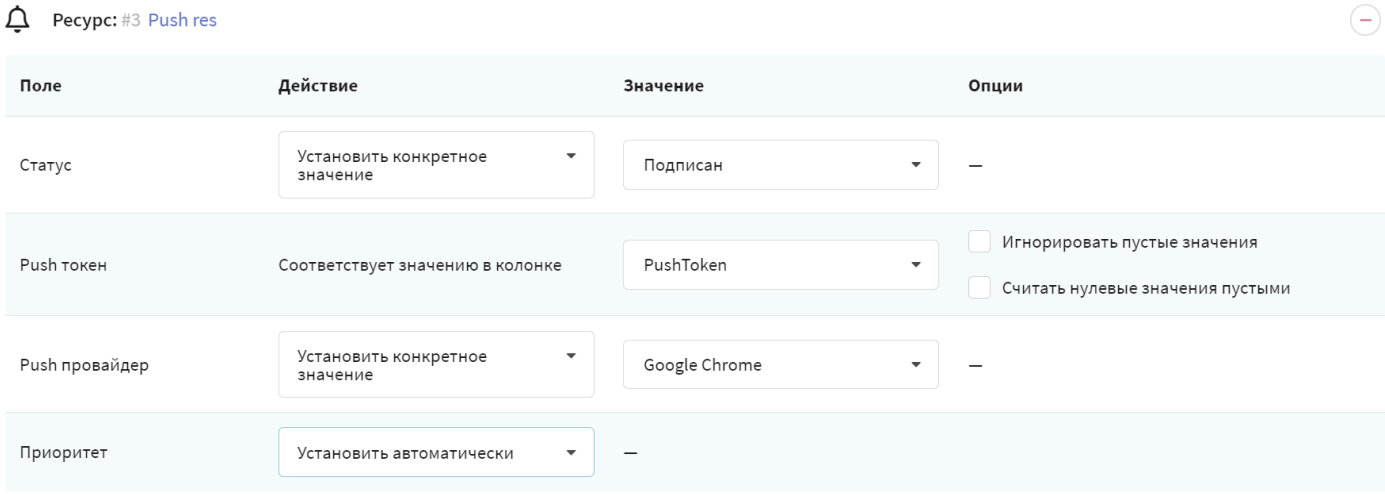


Для каждой подписки выберите **ресурс** и **канал связи**, после чего нажмите кнопку **“Настроить”**. Во втором окне выберите **статус и приоритет подписки**, а также колонку в файле, где указаны контактные данные для рассылок по выбранному каналу:

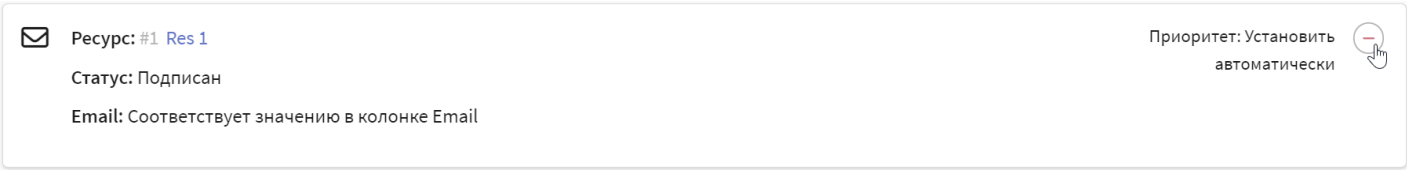
Каждый профиль имеет ограничение — не более 20 подписок на один ресурс.



Для подписок на **push-уведомления** дополнительно требуется указать **провайдер push-сервиса** и **push-токен**:

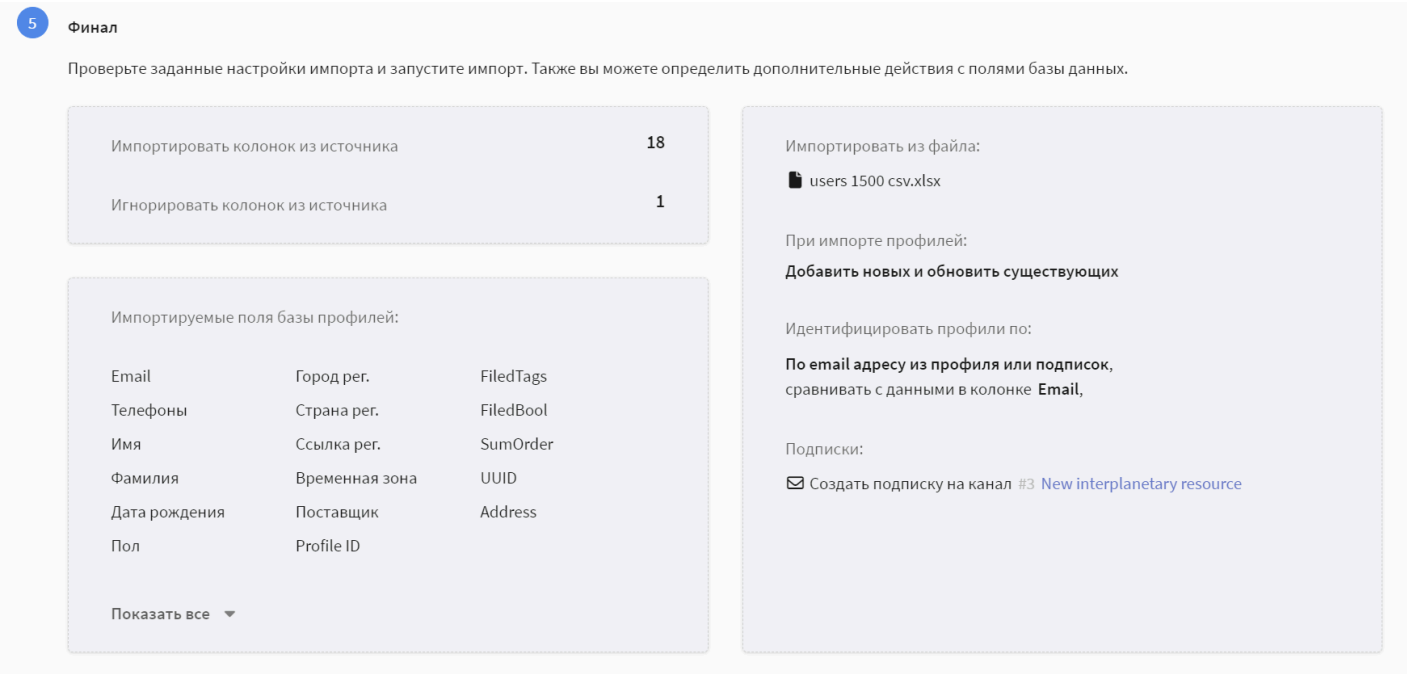


После настройки подписки, вы можете **отредактировать** её, кликнув по ней, или **удалить** нажатием на значок с красным минусом.

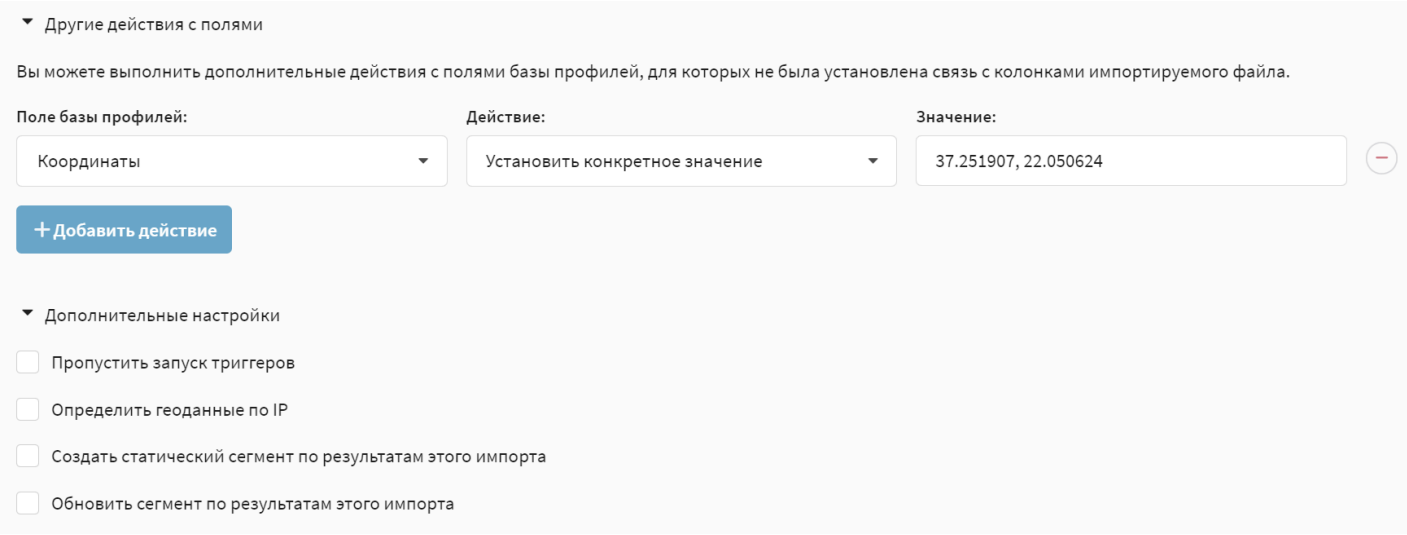


Финал

На финальном этапе импорта вы можете проверить ваши настройки и запустить импорт, а также определить дополнительные действия с полями базы данных. Вы можете вернуться на один или несколько шагов назад, но при этом сбросятся настройки текущего шага.



В блоке **Другие действия с полями** вы можете выполнить дополнительные действия с полями базы профилей, для которых не была установлена свярь: установить фиксированное значение для всех профилей или удалить текущее значение из поля.



Помимо этого в блоке **Дополнительные настройки** включаются добавочные опции:

- **Пропустить запуск триггеров** — не запускать сценарии и триггерные рассылки, настроенные на импорт в выбранную базу данных.
- **Определить геоданные по IP** — при передаче данных в поля **Reg IP** или **IP** в формате **IPv4** Altcraft автоматически определит геолокацию, а вы сможете выбрать такие поля профиля, как Координаты, Страна, Город — или Страна рег. и Город рег. для заполнения, также будет отмечено **положение профиля на виджете карты**.
- **Создать статический сегмент по результатам этого импорта** — при импорте профилей **платформа создаст статический сегмент**, в который войдут все добавленные и обновленные профили.
- **Обновить сегмент по результатам этого импорта** — при импорте профилей платформа автоматически пересчитает статический сегмент.

Передача статуса при импорте

При **настройке полей** и **подписок** можно передать глобальный статус профиля и статус его подписки, соответственно. Статус в колонке файла может быть указан как в текстовом, так и в числовом виде.

Возможные значения поля:

- 0, “subscribed” — “Подписан” (доступен для коммуникации по всем каналам связи)
- 1, “unsubscribed” — “Отписан” (отписался от получения сообщений по всем каналам)
- 2, “complained” — “Жалобщик” (пожаловался на спам, коммуникация остановлена)
- 3, “hardbounced” — “Hardbounced” (провайдер полность отказался принимать сообщения на этот адрес)
- 4, “unconfirmed” — “Не подтвержден” (подписчик зарегистрировался, но не подтвердил согласие на получение рассылок)
- 5, "suspended — “Приостановлен” (подписка приостановлена на некоторое время)
- 6, “invalid” — “Не валиден” (исключен из маркетинговых рассылок по любым другим причинам)

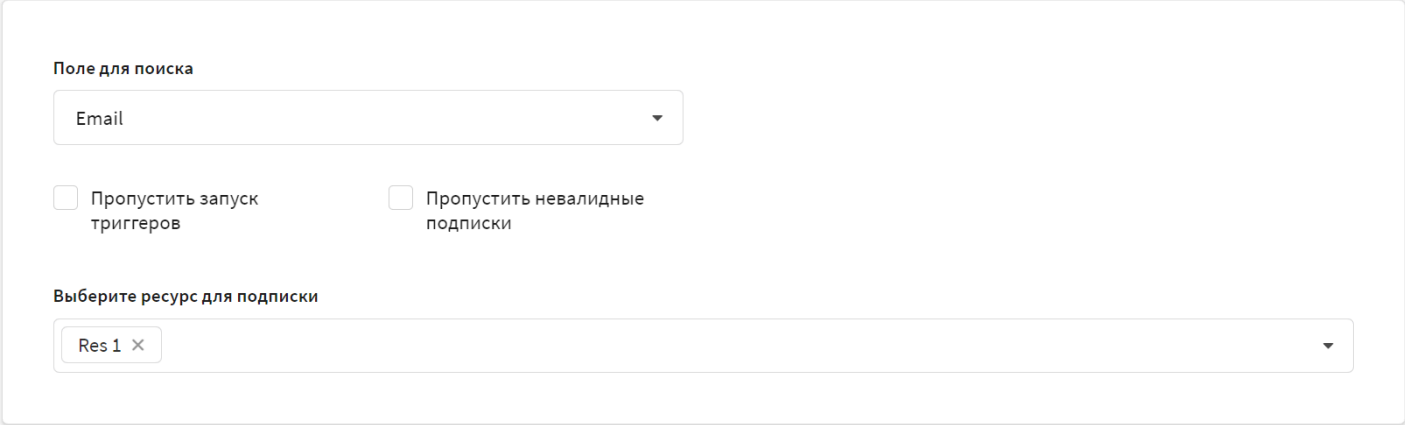
API импорт

API используется для интеграции с внешними системами — сайтами, приложениями, сервисами управления клиентскими данными (CRM), формами сбора данных. В конструкторе API-запроса вы можете задать определённые параметры для импорта данных и передать пример запроса вашим разработчикам.

В **“Поле для поиска”** выберите данные, по которым будет производиться поиск уже существующих профилей:

- **Email** — по полю профиля **Email**.
- **Телефон** — по полю профиля **Телефоны**.
- **По кастомному полю** — по пользовательскому полю базы данных. Например, ID клиента в CRM компании.
- **Идентификатор профиля** — по уникальному ID профиля (только при обновлении данных).

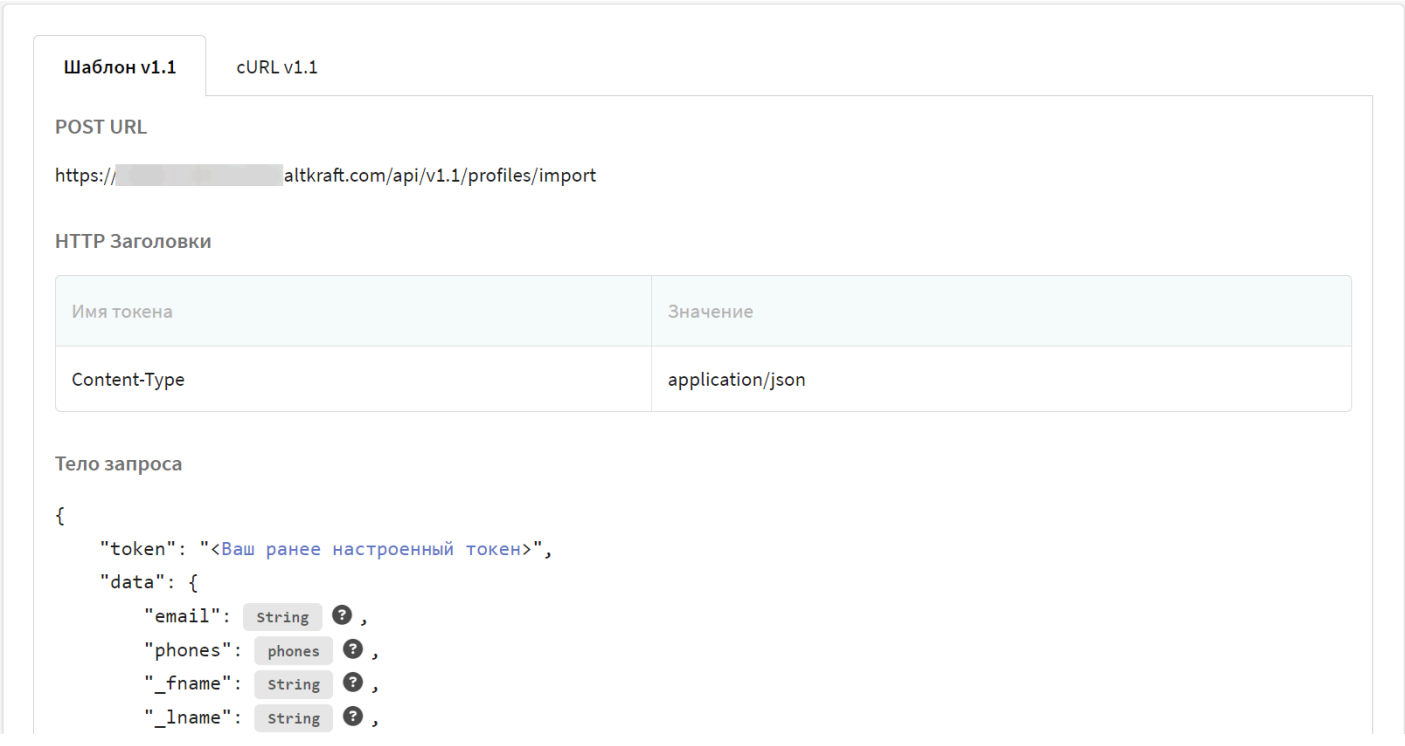
Также выберите **ресурс** для подписки.



Поставьте галочку в окне **Пропустить запуск триггеров**, если не хотите запускать сценарии и триггерные рассылки, настроенные на импорт в выбранную базу данных.

Поставьте галочку в окне **Пропустить невалидные подписки**, чтобы не добавлять подписку профиля, если в запросе присутствуют невалидные данные.

Ниже вы увидите **шаблон** API-запроса:



TIP

- Для импорта или обновления одного профиля используйте **этот запрос**.

- Другие запросы к API импорта описаны в разделе **Профили**.
- Управление подписками профиля описано в разделе **Подписки**.

Импорт из внешних SQL баз данных по расписанию

TIP

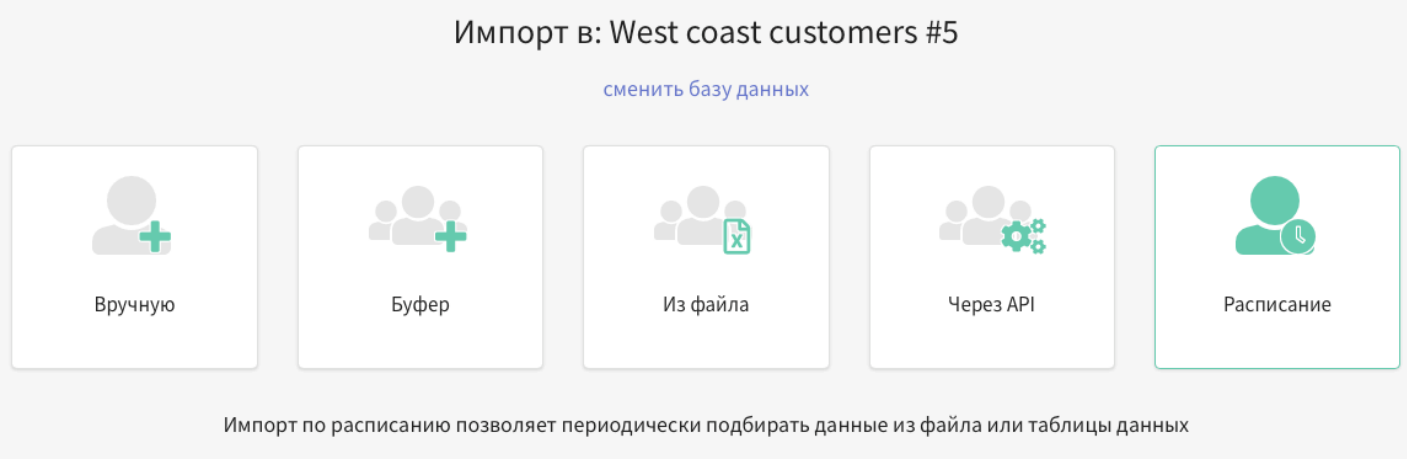
Altcraft может автоматически забирать данные для импорта новых профилей клиентов и обновления существующих из внешней таблицы данных ClickHouse, MySQL, MSSQL, PostgreSQL или Oracle RDBMS.

Инструкция по настройке импорта по расписанию вынесена в **отдельную статью**.

Импорт данных по расписанию

Импорт новых подписчиков в базы данных Altcraft, а также обновление данных существующих профилей можно автоматизировать с помощью **Импорта по расписанию**. Модуль импорта забирает данные из внешних таблиц данных. Для настройки автоматического забора данных требуется настроить в панели администратора **SQL-коннектор** и привязать его к аккаунту. В аккаунте, чтобы перейти к созданию импорта, перейдите в **Данные —> Импорты по расписанию**.

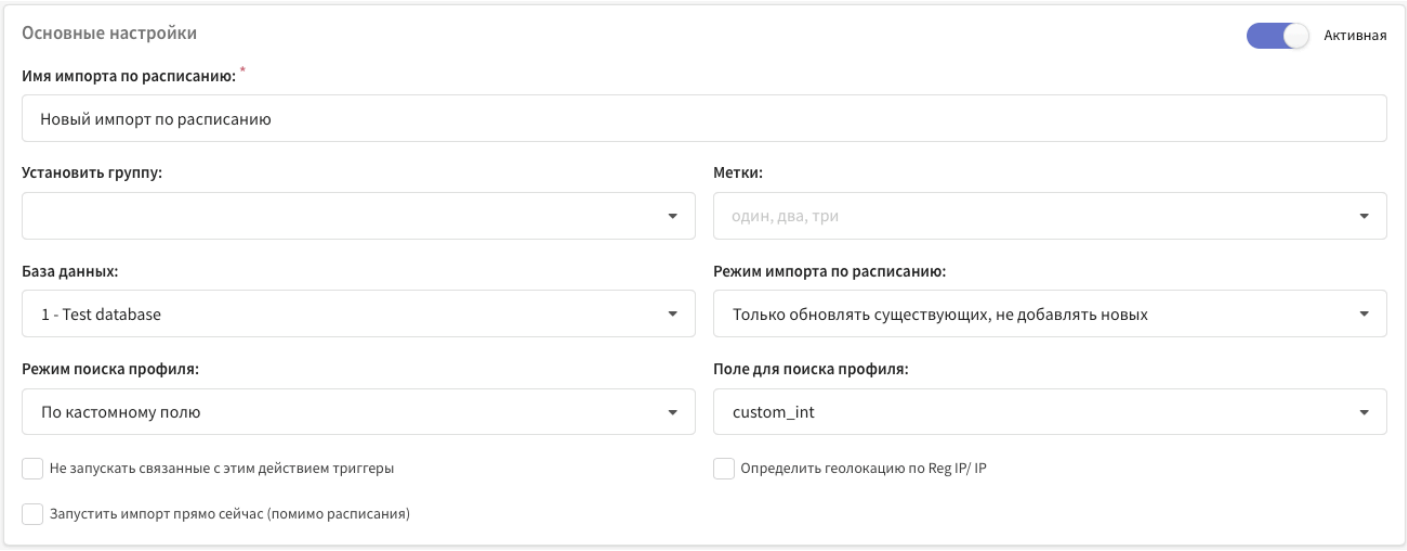
Также можно перейти в этот раздел из списка баз данных, выбрав в контекстном меню справа “Импорт профилей клиентов”:



Основные настройки

В основных настройках импорта вы можете задать имя импорта, назначить группы доступа и метки и установить, как импорт будет обрабатывать данные в аккаунте платформы:

- **База данных** — импорт будет добавлять или обновлять поля в выбранной базе профилей клиентов. От базы зависит набор доступных полей для хранения информации.
- **Режим импорта по расписанию** — импорт может работать в трёх режимах:
    1. Добавить новых и обновить существующих
    2. Только добавить новых, не обновлять существующих
    3. Только обновлять существующих, не добавлять новых
- **Режим поиска профиля** — доступен поиск по контактным данным профиля, данным подписки (по Email, SMS, Push и кастомным подпискам), по идентификатору профиля для обновления существующих профилей — и по кастомному полю базы данных. Также доступен мультиматчинг — по Email или телефону из профиля и подписок (`email_phone`), или по email или телефону из подписок (`email_phone_sub`)
- **Поле для поиска профиля** — при поиске профилей по кастомному полю базы данных его нужно выбрать в этом меню.



Для импорта по расписанию также доступно автоопределение геолокации по IP. Если у вас настроены триггерные рассылки на импорт или обновление профилей, можно ограничить их срабатывание для импорта по расписанию.

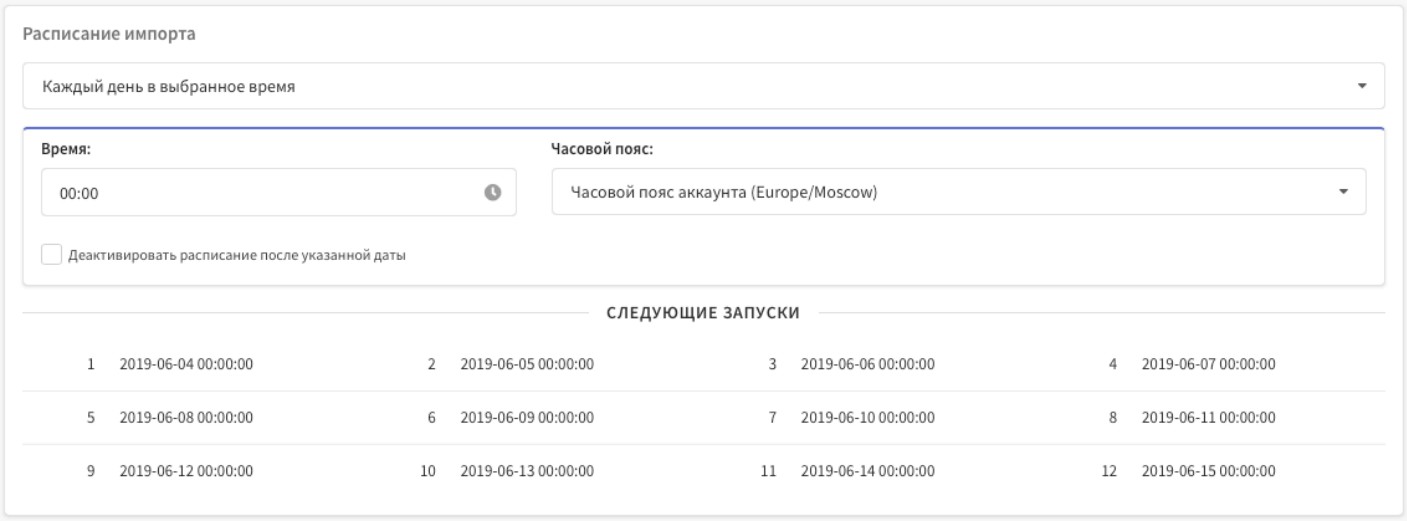
В основных настройках находится опция мгновенного запуска импорта помимо расписания. Если отмечен этот пункт, после нажатия на кнопку **Применить** или **Сохранить** импорт будет запущен.

Расписание импорта

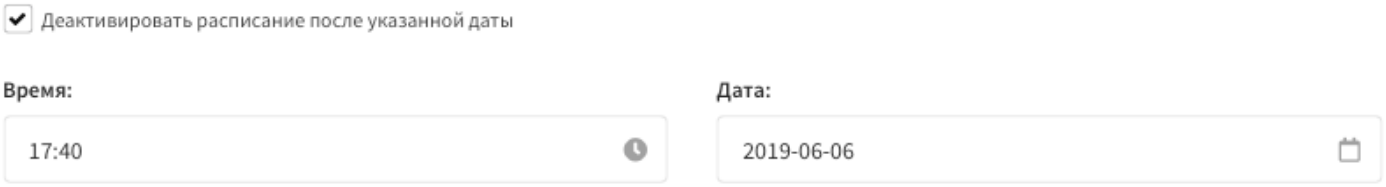
Для регулярного импорта можно гибко настроить расписание. Из выпадающего меню выберите план расписания и настройте в зависимости от назначения импорта. Есть разовые и регулярные планы расписания.

TIP

Расписание рассылок подробно описано в отдельной статье.

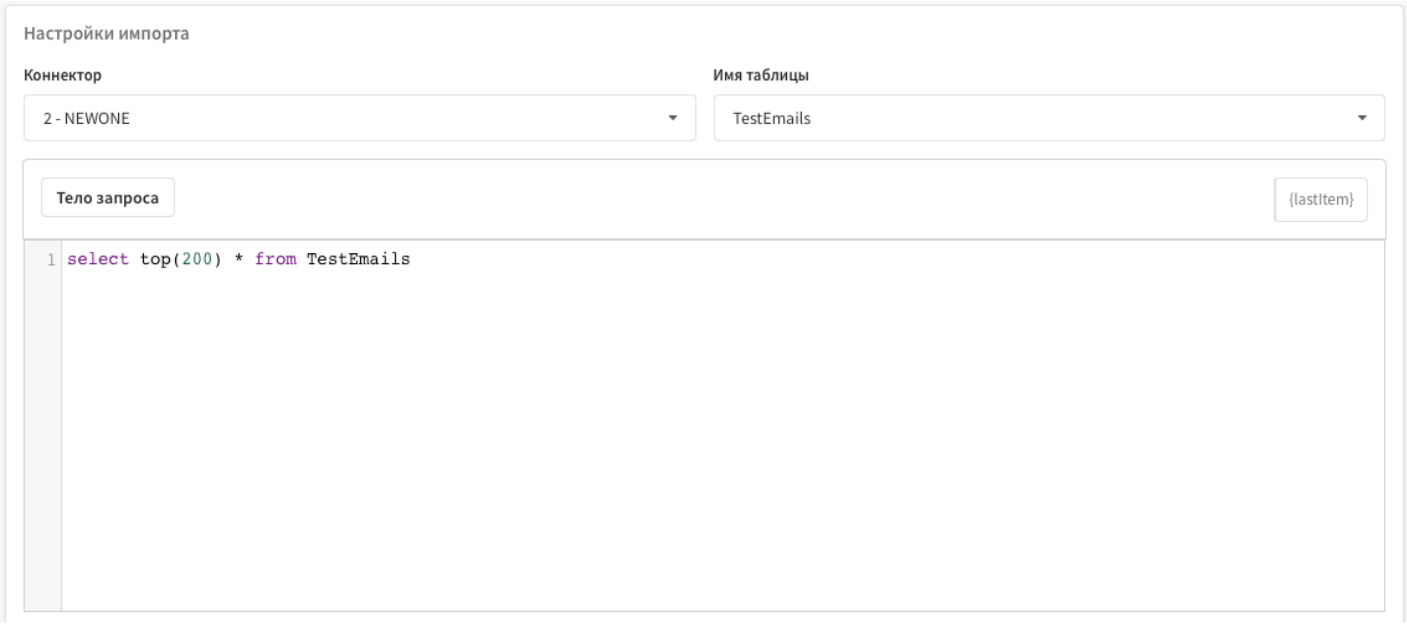


Для каждого плана расписания можно установить часовой пояс для расчёта. Также можно указать дату деактивации импорта:



Настройки импорта

В настройках импорта выберите необходимый коннектор к базе данных (настраивается в панели администратора) и таблицу в этой базе.

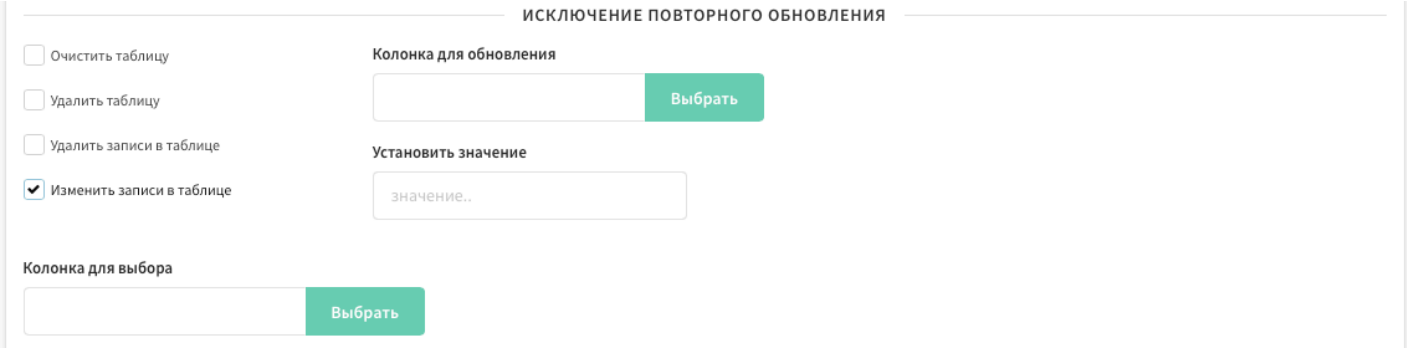


Составьте тело запроса в зависимости от используемой СУБД: MySQL, PostgreSQL, MSSQL, Oracle, Hive или ClickHouse. В запросах обязательно должен присутствовать оператор **SELECT**. Других ограничений не предусмотрено.

Исключение повторного обновления

Чтобы исключить повторное обновление данных, доступны следующие функции:

- **Очистить таблицу** — удаляет все записи в таблице данных
- **Удалить таблицу** — удаляет таблицу из базы данных
- **Удалить записи в таблице** — если значение указанной колонки в записи из таблицы и в профиле в базе данных совпадает, такая запись будет удалена.
- **Изменить записи в таблице** — меняет значения в выбранной колонке для импортированных записей. Идентифицирует записи по **Колонке для выбора** (уникальна для каждого профиля и не используется для поиска) и обновляет данные в **Колонке для обновления** на установленное значение.



Разделение импорта на части

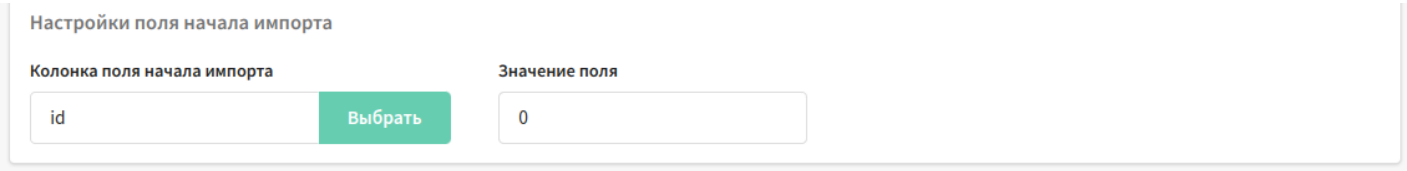
При импорте большого количества записей можно разделить процесс на части, чтобы распределить нагрузку. Для этого нужно добавить переменную **{lastItem}** в тело запроса для колонки с уникальным идентификатором. Идентификаторы при составлении запроса рекомендуется отсортировать.

NOTE

Пример запроса: `SELECT * from TestTable where id>{lastItem} LIMIT 10`

Далее нужно указать, с какой строки в таблице вы хотите запустить первый импорт. Для этого выберите колонку с идентификатором в **Колонка для начала импорта** и укажите **Значение поля**.

Пример настроек для запуска импорта с первой строки (если данные в колонке `id` отсортированы по возрастанию):

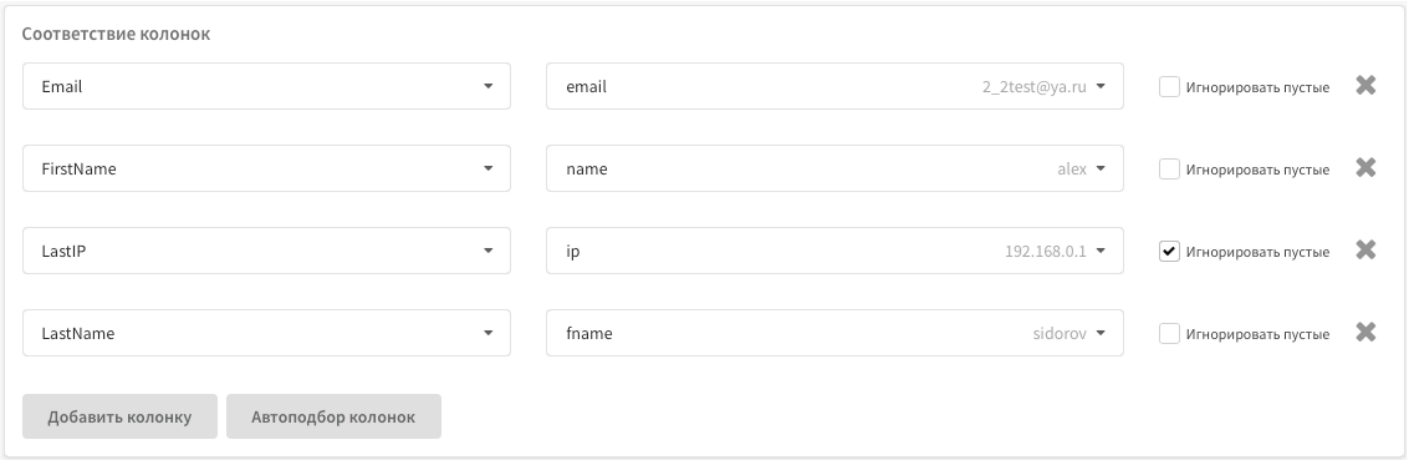


Когда вы запустите регулярный импорт, **Значение поля** будет автоматически обновляться — в него будет подставляться последний идентификатор после предыдущего импорта.

Соответствие колонок

В этом блоке нужно определить соответствие полей профилей в базе данных Altcraft и колонок внешней SQL базы. После нажатия на кнопку **Добавить колонку** слева из выпадающего меню выберите поле профиля Altcraft для обновления или записи данных.

При выборе колонки внешней SQL базы данных первое значение в ней будет отображаться справа серым текстом, как в примере ниже:



Особенности импорта полей

При импорте данных в базу платформа ожидает определённые форматы значений для каждого типа поля. Все допустимые форматы описаны в **этой статье**. Убедитесь, что данные в вашем источнике соответствуют указанным требованиям, иначе платформа не сможет корректно обработать их.

Для полей типа **Объект** и **Массив объектов** платформа ожидает значение в виде строки, содержащей корректный JSON. Чтобы задать такое значение, необходимо сначала преобразовать JSON-объект в строку, а затем экранировать её по правилам CSV, описанным в **RFC 4180** (двойные кавычки " заменяются на “”).

Пример: `"[{""field1"": 123, ""field2"": {""subfield"": false}, ""field3"": ""abc""}]"`

В поле типа **“Список”** можно импортировать сами значения или их индексы. Например, если вы создали в базе список с вариантами “тариф_1”, “тариф_2”, “тариф_3”, то в колонке источника можно указать эти же значения или использовать индексы: 1 — это “тариф_1”, 2 — это “тариф_2”, 3 — это “тариф_3”, 0 — “Не выбрано” (значение по умолчанию).

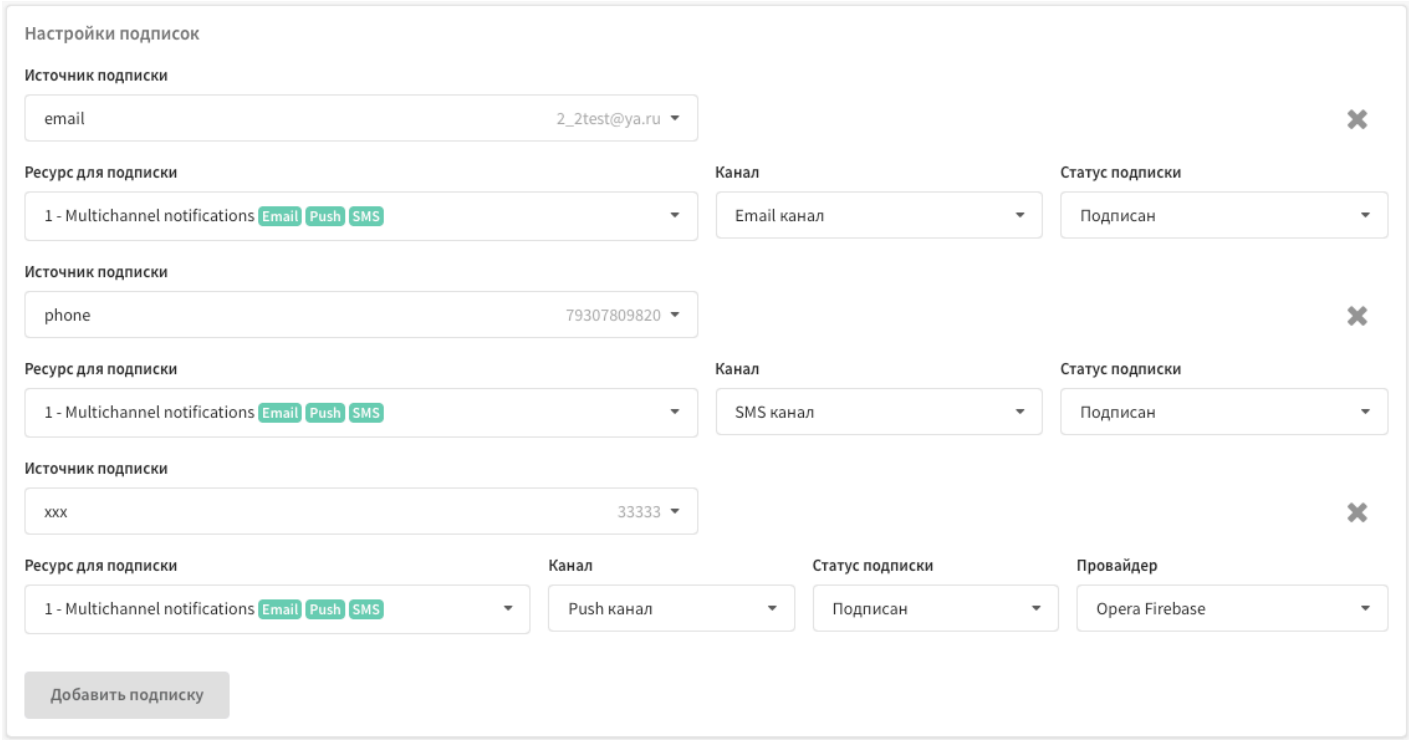
Обратите внимание, что значения, которые состоят только из цифр, невозможно импортировать напрямую — платформа воспримет их как индексы. Например, если в списке указаны варианты 100 и 200, то при импорте нужно передать соответствующие им индексы: 1 — это 100, 2 — это 200.

Настройки подписок

При импорте можно автоматически подписывать клиентов на те или иные Ресурсы по одному или нескольким каналам. Для каждой подписки соответственно нужно выбрать ресурс, канал коммуникации и выбрать колонку в таблице данных, из которой будут импортироваться контактные данные для подписок.

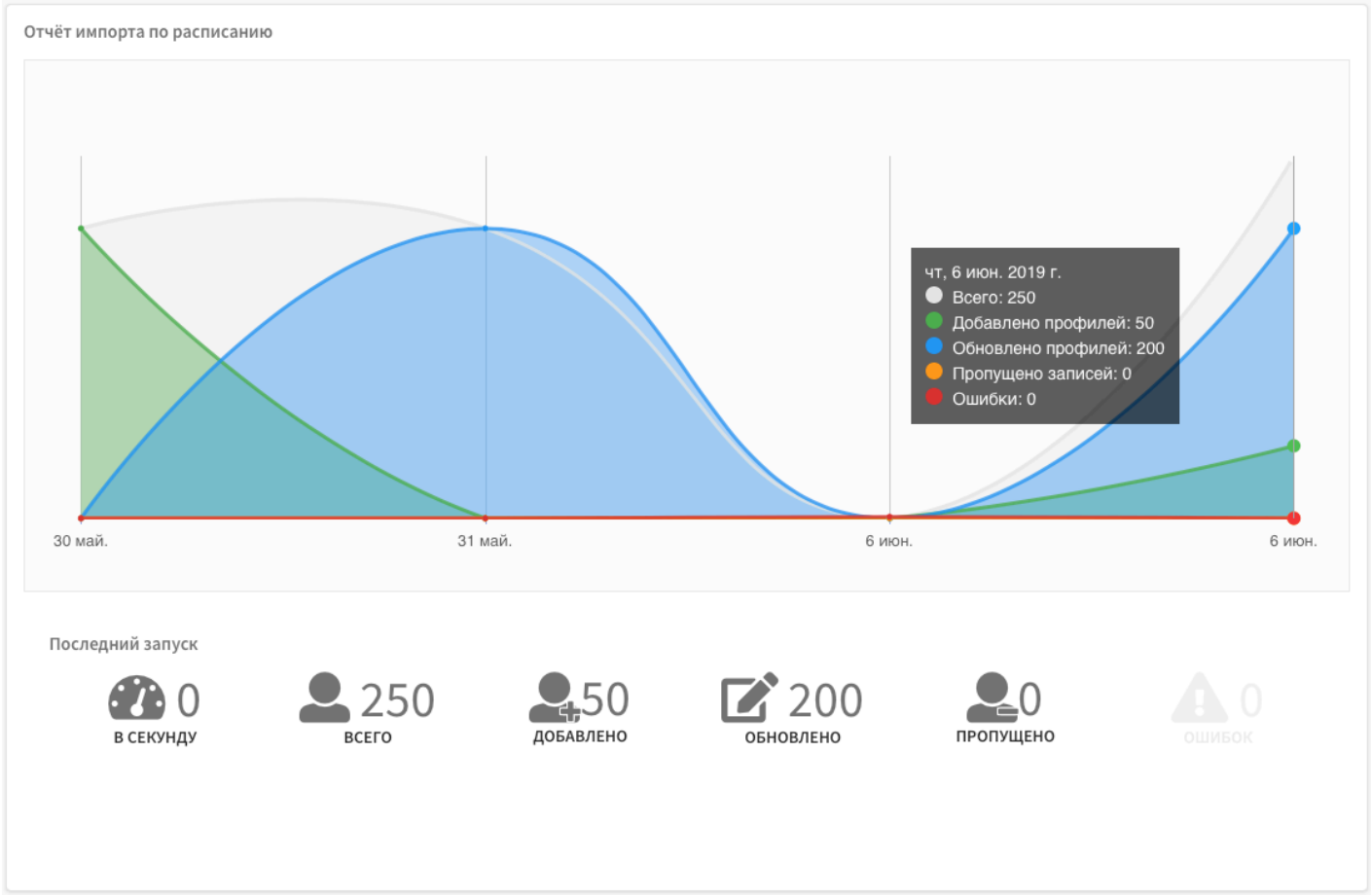
Для каждой подписки можно установить **статус**:

- при импорте можно передавать статус “Подписан” для Single opt-in, или “Не подтверждён” для **Double opt-in логики**
- при обновлении данных можно менять статус подписки в зависимости от действий пользователя, например приостановить, возобновить или отменить подписку



Отчёт импорта по расписанию

В этом блоке вы можете просмотреть информацию по последним 10 запускам. Если во время последнего импорта произошли ошибки, отчёт по ним можно выгрузить.



Автоматизация сбора данных о профиле

В системе ALTCRAFT существует несколько способов автоматически поддерживать актуальность информации в профиле:

Трекинг

Собирает информацию о действиях подписчиков в сообщениях с помощью отслеживания показов изображений, времени чтения сообщения и кликов по ссылкам.

Email-трекинг

Сбор данных о действиях подписчика в email-сообщении осуществляется с помощью пикселей и ссылок, которые вставляются в email письмо.

Трекинг ссылок и пикселей создаёт события:

- **Открытие** — получатель открыл сообщение.
- **Клик** — получатель открыл сообщение и кликнул по одной из ссылок.
- **Прочтение** — получатель открыл сообщение и не закрывал его как минимум 12 секунд.
- **Подтверждение** — получатель кликнул на ссылку подтверждающую подписку.
- **Отписка** — получатель отписался от ресурса.

Email-отправщик (sender) собирает события, которые ему возвращает email-провайдер:

- **Доставлено** — сообщение было доставлено получателю.
- **Не доставлено** — сообщение не было доставлено получателю.
- **Softbounce** — ошибка клиента отправки.
- **Hardbounce** — отправка на почтовй сервер невозможна. Статус подписчика автоматически изменен на **Hard-bounced**.
- **Complain** — клиент пожаловался на Спам. Коммуникация остановлена.

Общая статистика отображается в отчёте по каналам, а также каждый профиль имеет собственную историю событий.

Статистика по кликам в отдельном email сообщении также собирается в отчёте карта кликов, причём каждая версия этого сообщения имеет собственный отчёт.

Также данные email трекинга используются в отчётах когорты и время жизни.

SMS-трекинг

Ссылки в SMS-сообщении также отслеживаются трекингом, однако они намного короче за счёт встроенного шортера.

```
example.com/s/GrlX73
```

SMS-отправщик (sender) собирает события **Доставлено**, **Не доставлено** от провайдеров SMS-рассылок.

Web Push-трекинг

События с web push-сообщениями обрабатываются через службу Веб Push Mozilla или Google Firebase Cloud Messaging.

В результате действий пользователя в платформу возвращаются события **Push доставлен**, **Push открыт** и **Клик на ссылку в Push**.

App Push-трекинг

События в app push-сообщениях обрабатываются самим мобильным приложением.

Для событий **Push доставлен** и **Push открыт** в структуре app push-сообщения передаются две ссылки “ack” и “open”, на которые приложение должно послать GET запрос.

Опционально, для события **Клик на ссылку в Push** используется параметр “click_action”.

AK Pixel

Пиксели собирают информацию о действиях пользователей сайта. Если пользователь открывает email-сообщения, отправленные из Altcraft, и сайт в одном браузере, то данные будут сохраняться в его профиль.

API для редактирования профиля и подписок

С помощью API можно создавать и редактировать профили.

- Профили
- Подписки

Сценарии (цепочки)

Сценарий — настраиваемая последовательность действий системы в случае выбранного события.

Узел “Обновить данные” непосредственно изменяет данные в выбранных полях профиля.

Узел “Обратная связь” отправляет сообщение на заданный сервер, который может использовать Altcraft API или работать непосредственно с базой данных.

Захват событий (Action hooks)

Захват событий отправляет сообщения о событиях в другие системы.

Реагирует на большее количество событий, чем сценарий, а также использует не только HTTP, но и умеет отправлять сообщения в RabbitMQ.

Системные атрибуты ссылок в сообщениях

Атрибут ссылок `update-вставьте_имя_поля="запишите в него это значение"` изменяет данные в профиле подписчика при клике на такую ссылку:

```
<a href="http://www.example.org" name="link" update-clicked_link="true">Обновить данные
поля clicked_link!</a>
```

Атрибут **subscribe-to** при переходе по ссылке добавит клиенту **подписку** по тому же каналу на указанный ресурс:

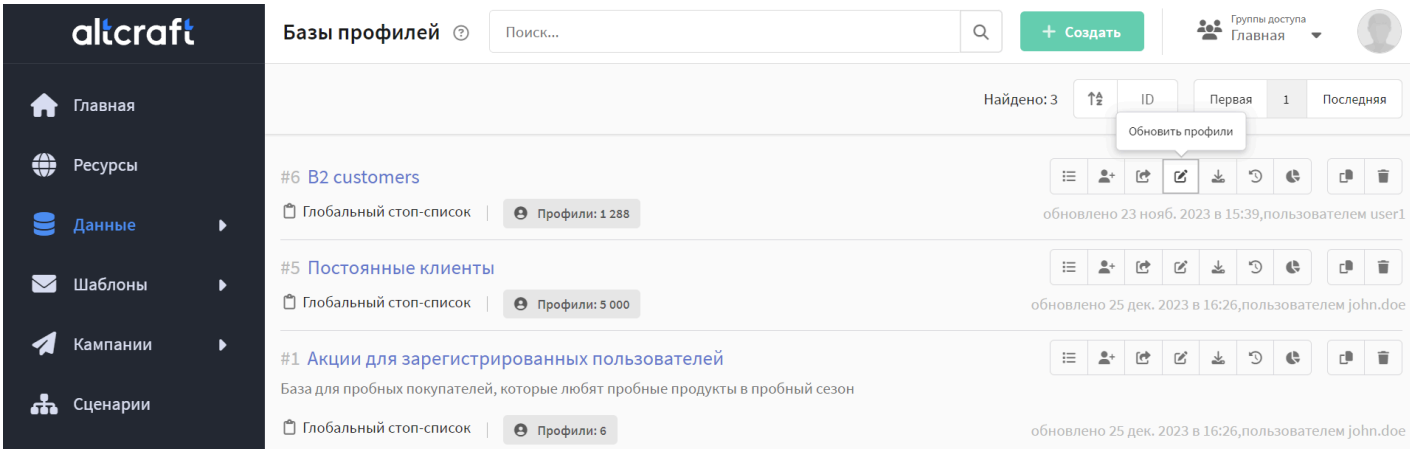
```
<a href="http://www.example.org" subscribe-to="1">Подписаться на ресурс #1</a>
```

Массовое обновление профилей клиентов

Массовое обновление профилей позволяет вручную обновить **Базу данных** или **Сегмент** подписчиков. При этом можно добавить условия выборки, чтобы уточнить запрос на обновление. Доступны следующие функции:

- Добавление и изменение значений полей профиля клиента
- Добавление и обновление подписок профилей
- Управление промокодами
- Удаление профилей
- Глобальная отписка профилей

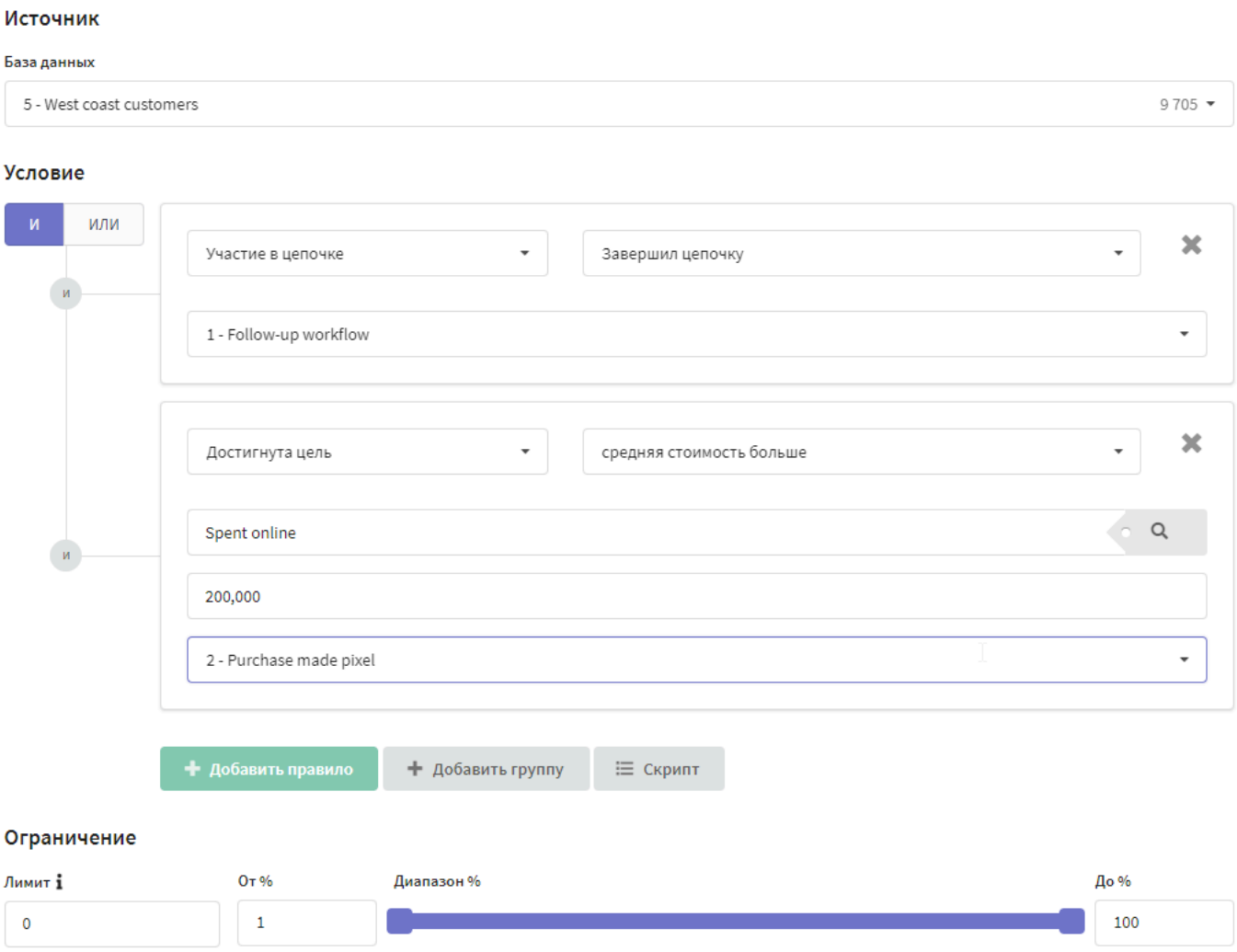
Чтобы открыть инструмент массового обновления, в главном меню выберите **Данные —> Базы профилей** или **Данные —> Сегменты**. В контекстном меню группы профилей выберите **Обновить профили:**



Выборка профилей клиентов

Выберите **базу**, где хранятся профили, которые необходимо обновить. Если вы хотите обновить только часть профилей в базе, то добавьте уточняющие **правила сегментации.**

Вы также можете добавить несколько правил и объединить их с помощью логического **И** (все условия должны выполняться) и **ИЛИ** (хотя бы одно условие должно выполняться).



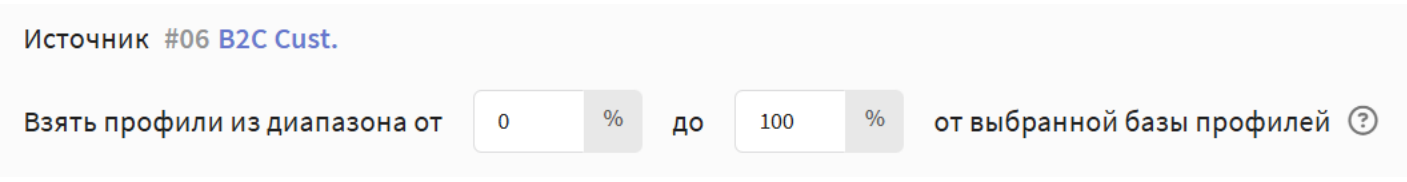
TIP

В [этом разделе] документации вы найдете информацию о сегментации профилей клиентов по личным данным, данным подписок и истории действий.

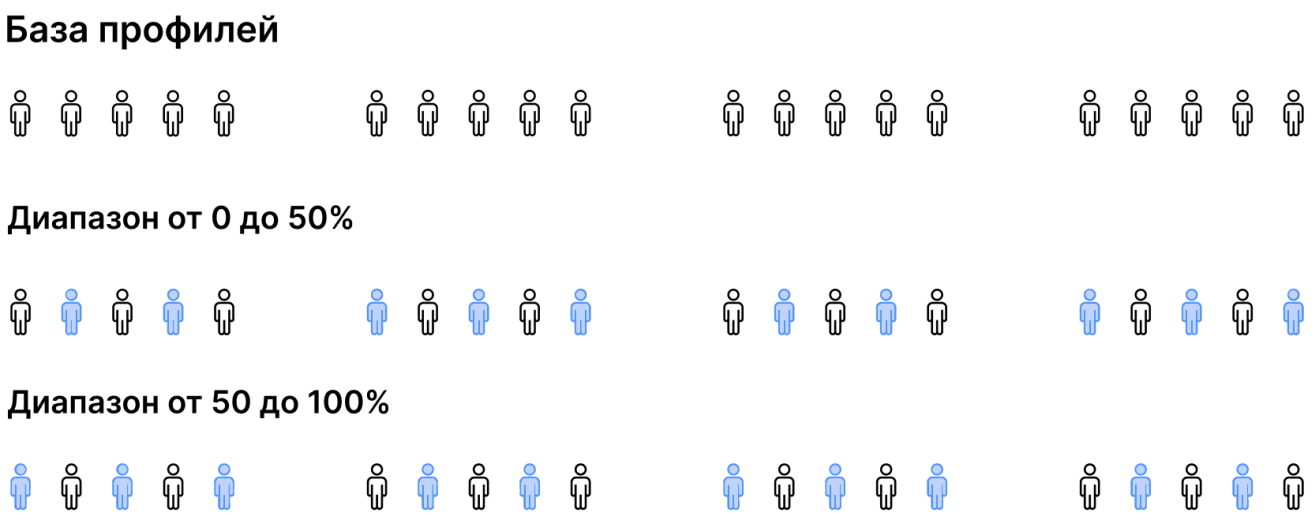
Вы также можете установить ограничение для получившейся выборки в виде **Лимита** или **Диапазона**:

- **Диапазон** определяет часть базы, к которой будут применены условия сегментации. По умолчанию диапазон равен 100%, т.е. если в базе 1000 профилей, то условия сегментации применятся ко всем профилям в базе. Если, например, вы установите диапазон от 10 до 50%, то условия будут применены к примерно 400 профилям, выбранным равномерно из всего объема базы.
- **Лимит** — максимальное количество профилей, которые могут быть включены в сегмент. Например, если 1000 профилей соответствуют условиям сегментации, а лимит — 400, то в сегмент войдут первые 400 профилей из выборки.

Ограничения, заданные с помощью **диапазона**, распространяются на всю базу. Ограничения, заданные с помощью **лимита**, накладываются на результат выборки сегмента.



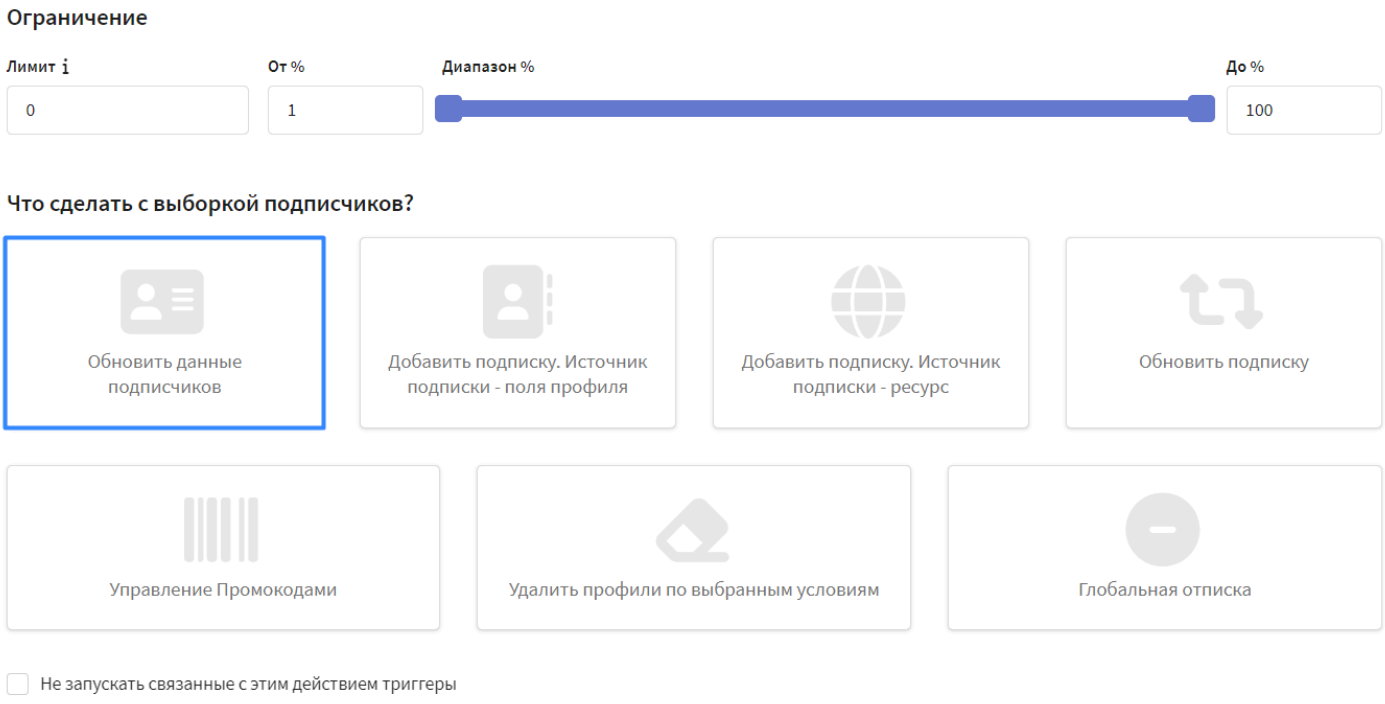
Платформа закрепляет диапазон за конкретными пользователями: один и тот же диапазон всегда отбирает **одних и тех же людей** из базы, даже если вы используете его в разных сегментах. Это означает, что выборка **не случайна** — каждому пользователю сопоставляется числовое значение, и попадает он в сегмент только при попадании этого значения в заданный диапазон. Благодаря этому можно создавать непересекающиеся выборки, назначая разные диапазоны (например, диапазон от 0 до 20 процентов и диапазон от 80 до 100 процентов никогда не пересекутся). При добавлении новых пользователей в базу они также могут попасть в заданный диапазон с той же вероятностью — например, с 1/5, если выбран диапазон в 20%.



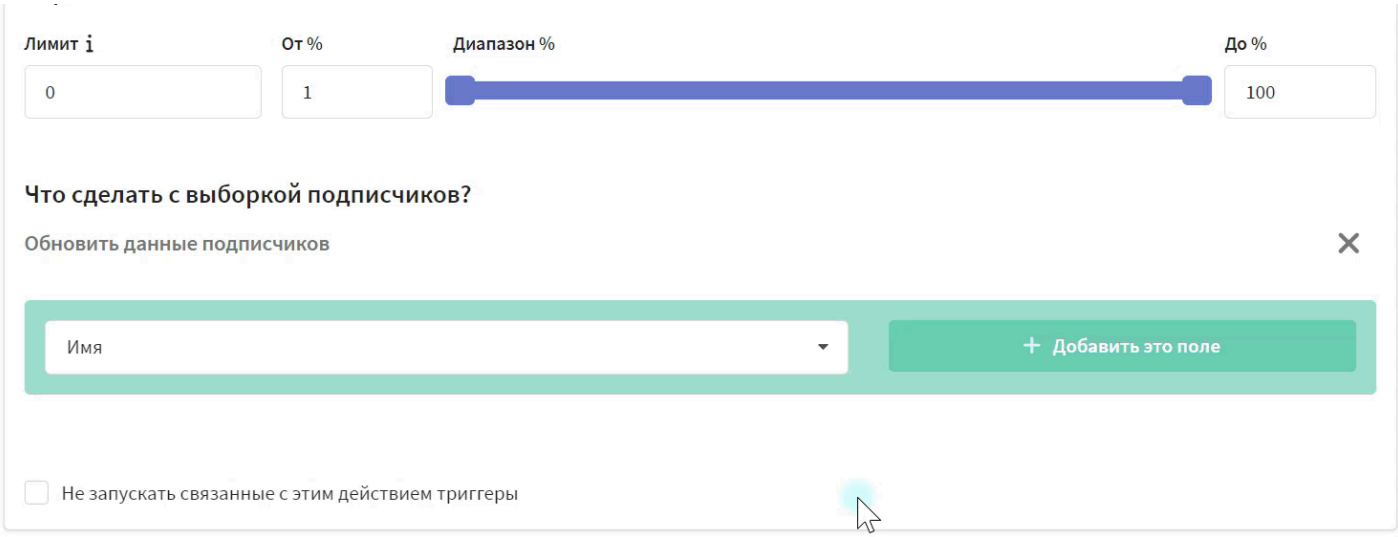
Обновление профилей клиентов

Обновление полей профиля клиента

Выберите опцию **“Обновить данные подписчиков”**:

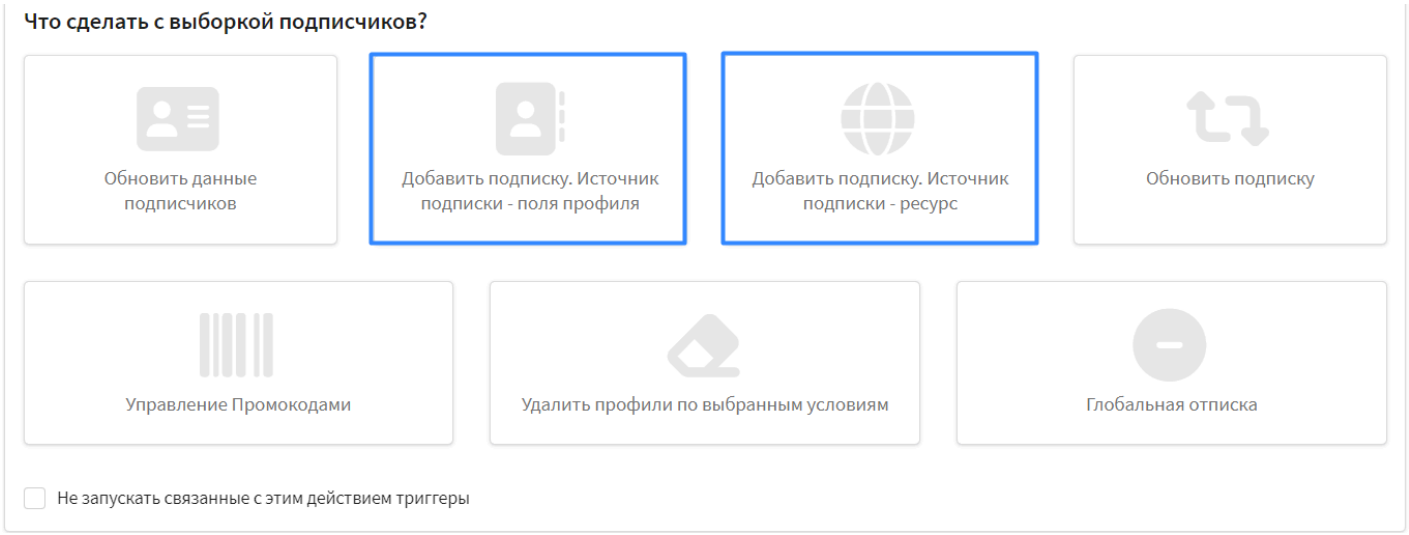


В выпадающем списке доступны все поля базы. Выберите поле, которое хотите обновить и нажмите **+ Добавить это поле.** Далее выберите вариант обновления: изменить или удалить значение. При изменении выберите новое значение в зависимости от типа поля:



Добавление подписки на новый ресурс

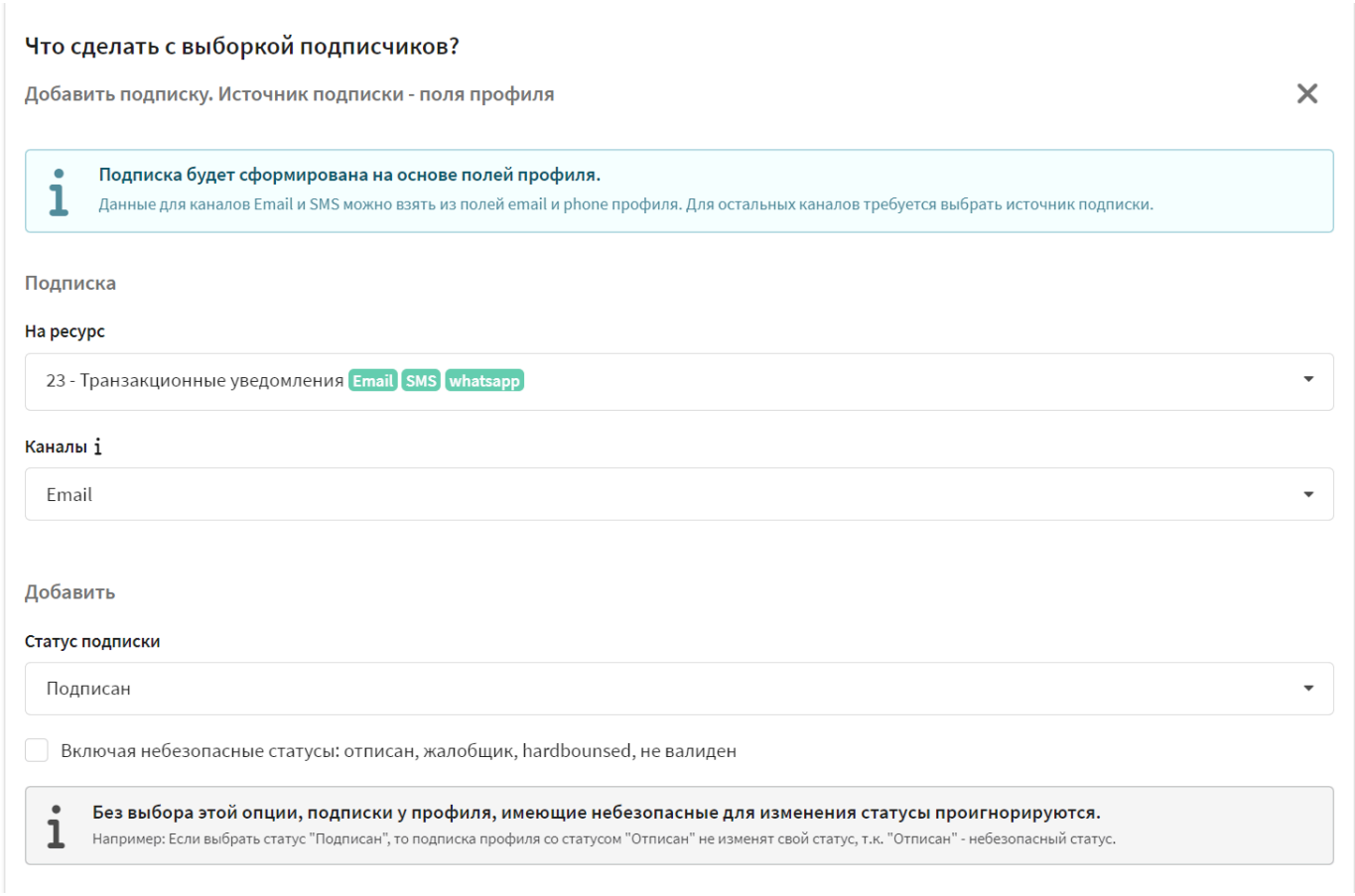
В зависимости от источника контактных данных доступно два способа добавления подписки:



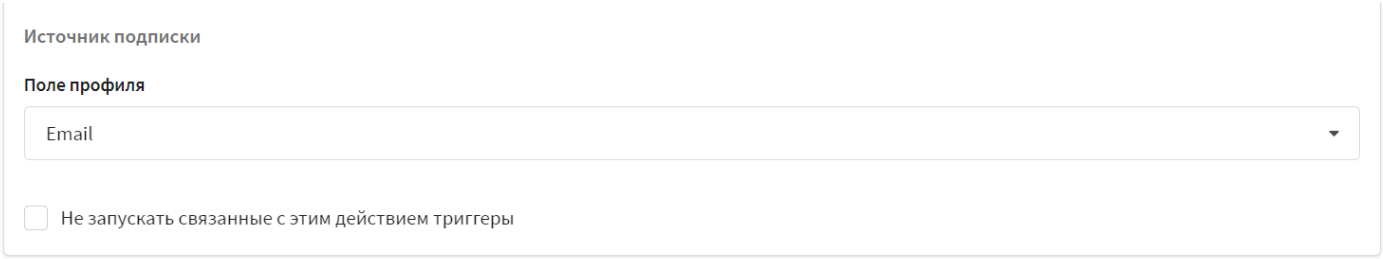
Источник подписки — поля профиля

Подписка создается на основе контактных данных (номер телефона, адрес электронной почты и др.), которые указаны в полях профиля.

Выберите **ресурс** и **канал**, на который хотите добавить подписку. Ресурс должен быть привязан к базе. Далее укажите **статус подписки**.



В качестве **источника подписки** выберите поле профиля. Для подписки на Email-канал используйте поле `Email`, для подписки на каналы SMS, WhatsApp*, Viber™ — поле `Phones`. Данные для других каналов коммуникаци вы можете хранить в **дополнительных полях базы**.



CAUTION: Небезопасные статусы

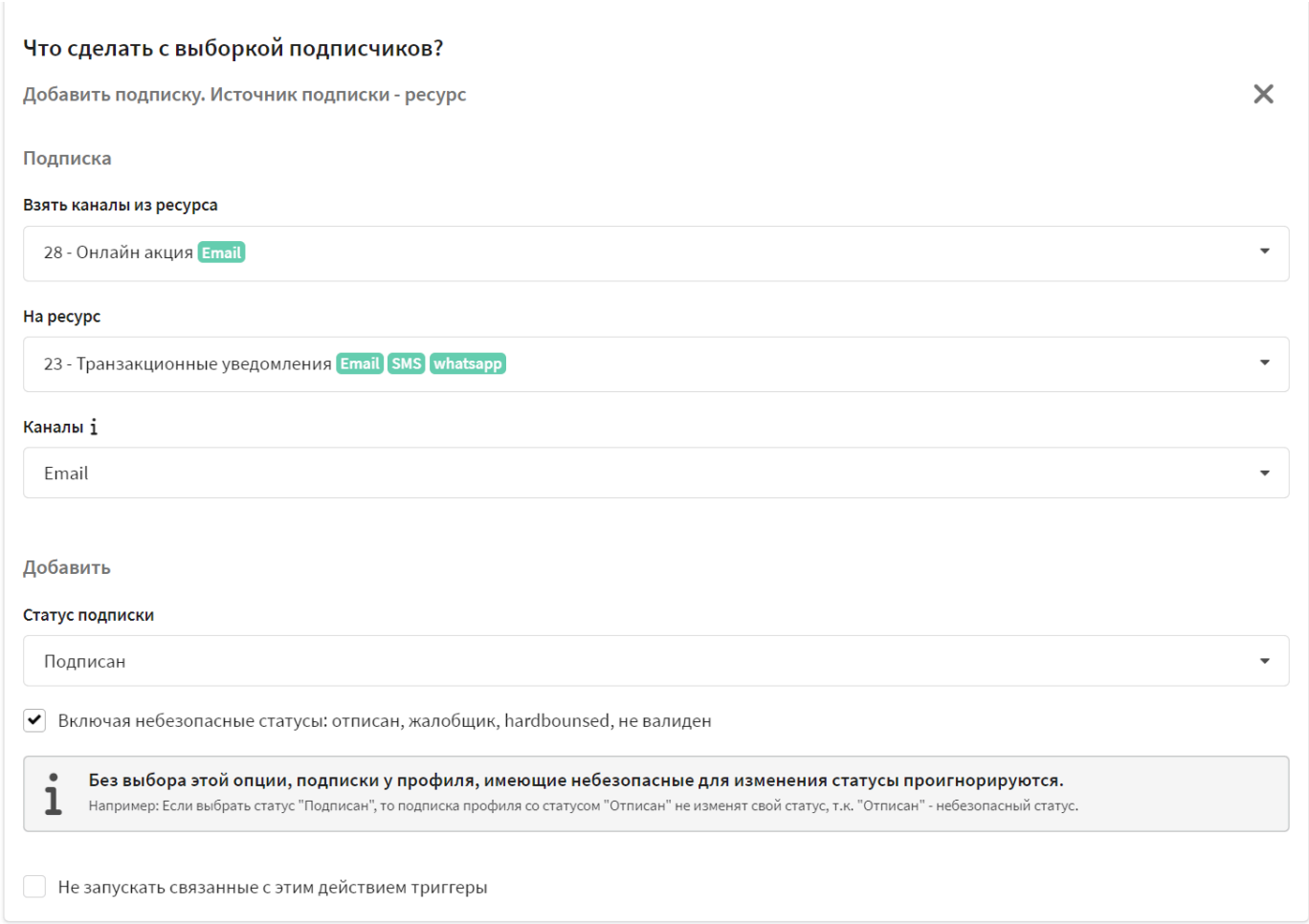
Если вы меняете статус подписки на “Подписан”, то по умолчанию подписки профилей с небезопасными статусами не обновляются. К таким статусам относятся: Отписан, Жалобщик, Hardbounced, Невалиден. Чтобы изменить это поведение, активируйте опцию **“Включая небезопасные статусы”**.

Источник подписки — ресурс

Подписка создается на основе контактных данных, которые используются в подписке на другой ресурс. Настройте поля:

- **Взять каналы из ресурса** — платформа использует для обновления контактные данные из подписки на выбранный ресурс. Если у профиля нет подписки на этот ресурс, платформа не добавит ему новую подписку.
- **На ресурс** — платформа добавит профилям подписку на выбранный ресурс.

- **Каналы** — платформа добавит профилям подписку на выбранный канал в ресурсе. Обратите внимание, что этот канал должен быть настроен в обоих ресурсах.
- **Статус подписки** — платформа добавит подписку с выбранным статусом.

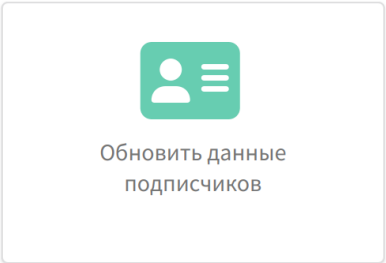


CAUTION: Небезопасные статусы

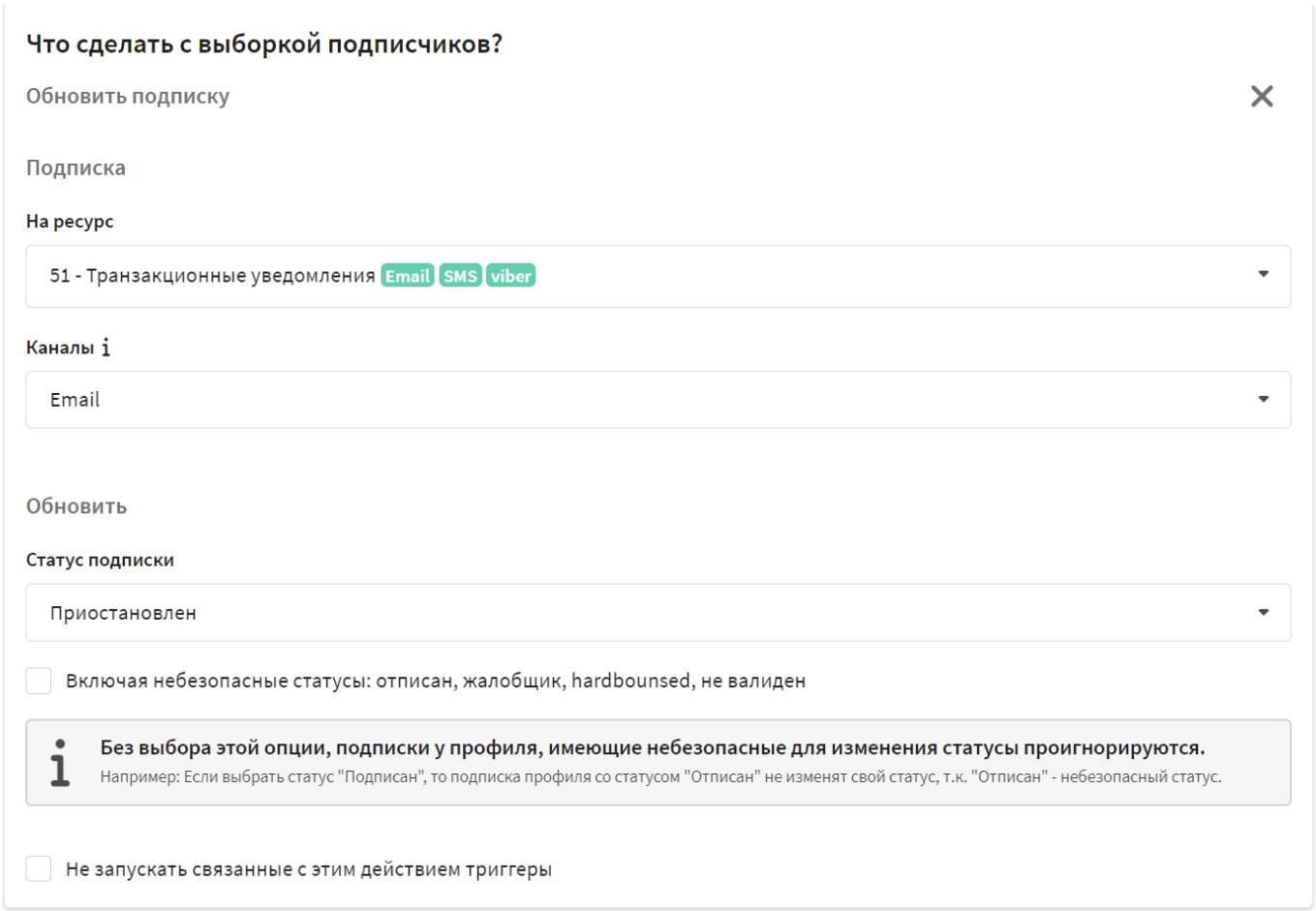
Если вы меняете статус подписки на “Подписан”, то по умолчанию подписки профилей с **небезопасными статусами** не обновляются. К таким статусам относятся: Отписан, Жалобщик, Hardbounced, Невалиден.

Также подписка не обновляется, если для обновления вы используете данные из подписки с небезопасным статусом. Чтобы изменить это поведение, активируйте опцию **“Включая небезопасные статусы”**.

Обновление подписки

	Меняет статус существующих подписок на ресурс. Удобно, если вы, например, хотите разово отписать всех неактивных клиентов.

Выберите ресурс и канал, а затем укажите, какой статус необходимо установить:

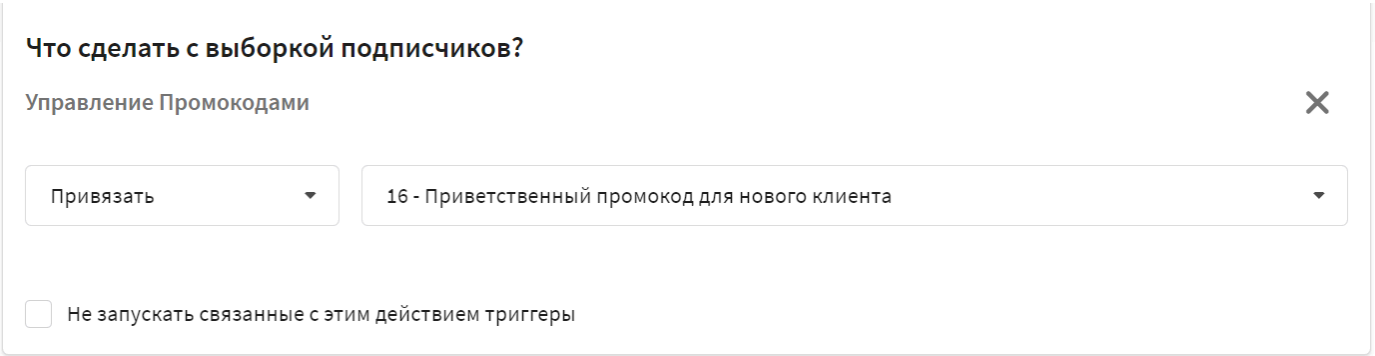


CAUTION: Небезопасные статусы

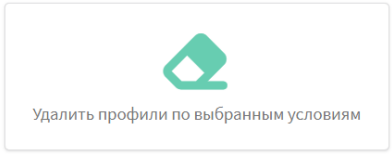
Если вы меняете статус подписки на “Подписан”, то по умолчанию подписки профилей с небезопасными статусами не обновляются. К таким статусам относятся: Отписан, Жалобщик, Hardbounced, Невалиден. Чтобы изменить это поведение, активируйте опцию **“Включая небезопасные статусы”**.

Управление промокодами

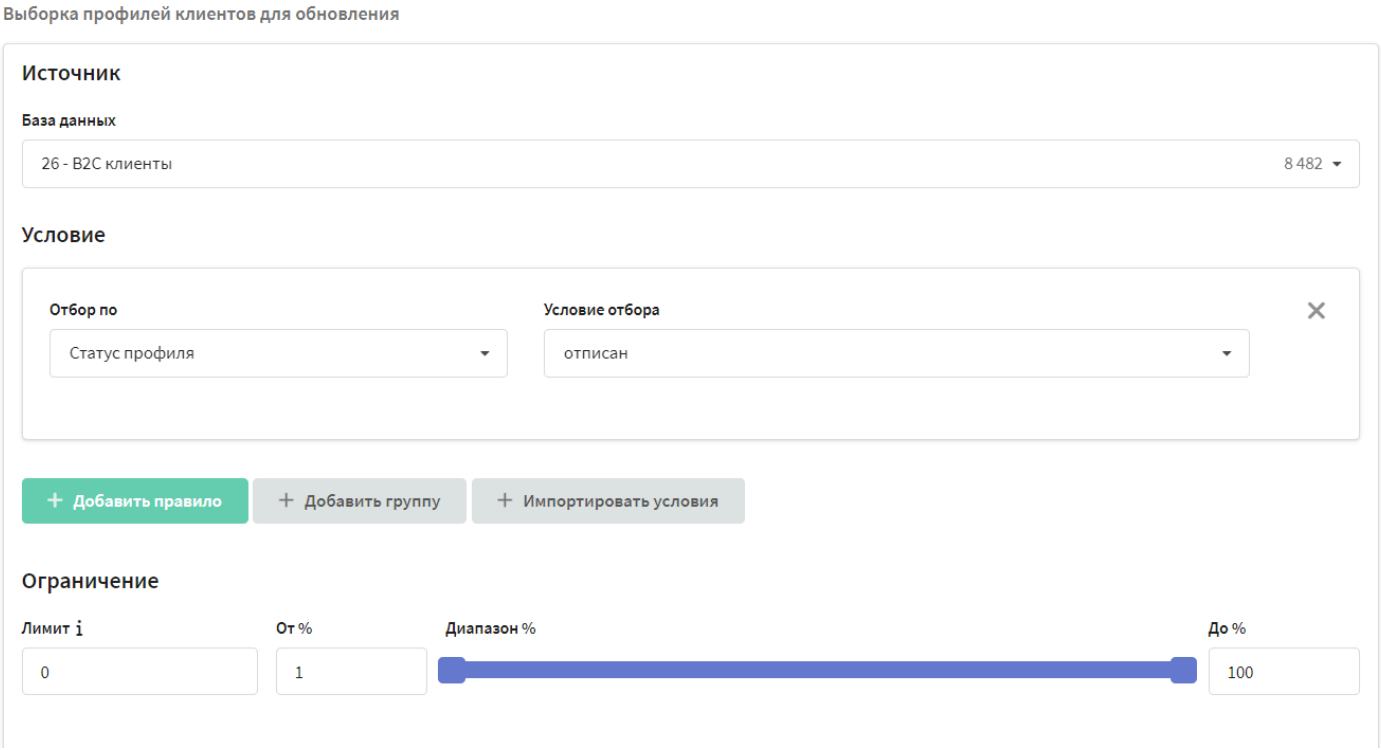
Данная опция позволяет **привязать** профилям или **отвязать** от них промокоды определенной **Программы лояльности**:



Удалить профили по выбранным условиям

	Удаляет из базы профили, соответствующие заданным условиям. При этом статистика действий удаленных профилей сохраняется в отчётах.

С помощью правил сегментации выберите профили, которые хотите удалить, а затем нажмите кнопку “Обновить” в правом верхнем углу:



Обратите внимание, платформа удаляет не все выбранные профили, а только те, с которыми не было взаимодействий в течение определённого периода.

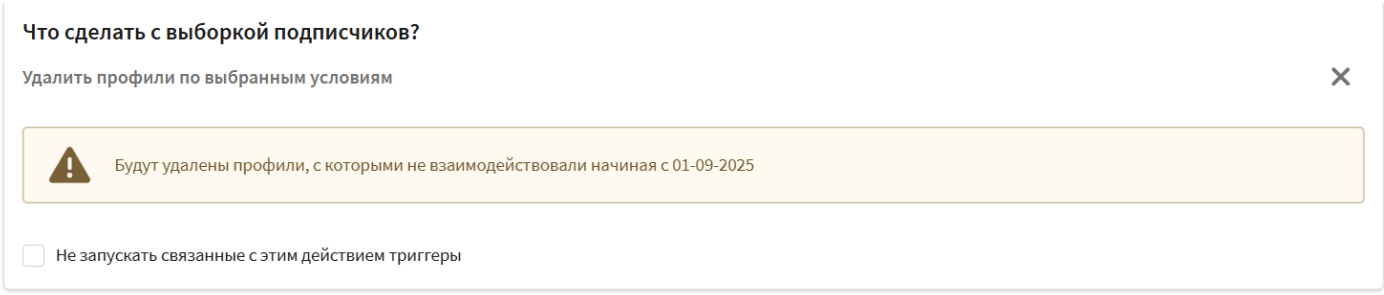
В **облачной версии** можно удалить профили, с которыми не было взаимодействий в предыдущем месяце. Например:

- Если сегодня 1 октября 2025 года, будут удалены профили, с которыми не было взаимодействий с 1 сентября 2025 года.
- Если сегодня 31 октября 2025 года, будут удалены те же профили.

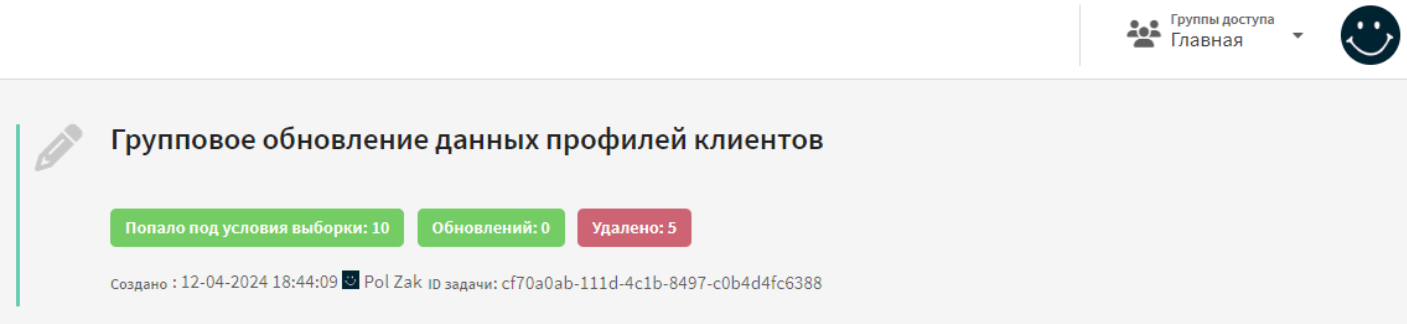
TIP

В **On-premises версии** период задаётся в файле конфигурации через параметр `PROFILES_REMOVAL_MIN_MONTHS`. Подробнее.

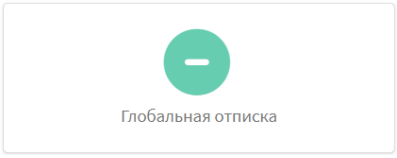
Взаимодействие предполагает любое действие профиля, записанное в его историю. Информация об актуальном периоде отобразится ниже:



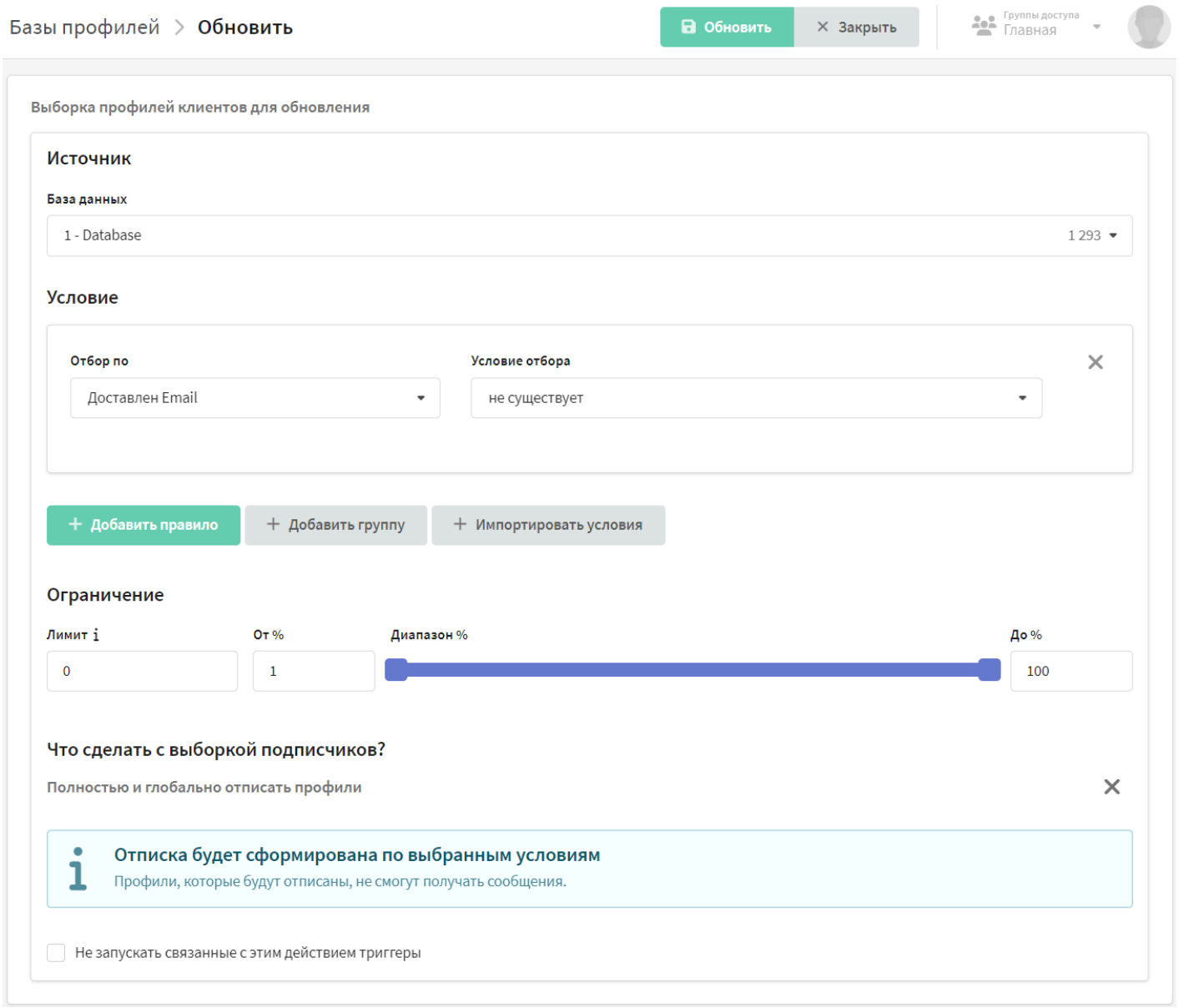
После запуска обновления вы можете перейти в **Задачи**, чтобы посмотреть, сколько профилей было удалено. Если профиль соответствовал условиям выборки, но не был удален, значит, с ним были зафиксированы взаимодействия в течение заданного периода.



Глобальная отписка

	Устанавливает всем профилям [глобальный статус](/user-guide/profiles-and-databases/9405665#Globalprofilestatus) **"Отписан"**. Это полностью останавливает коммуникацию с клиентом по любым каналам. Даже если у профиля есть активные подписки на ресурсы, рассылки не будут отправлять ему сообщения.

С помощью правил сегментации выберите профили, которые хотите отписать, а затем нажмите кнопку “Обновить” в правом верхнем углу:



В платформе могут быть настроены триггерные рассылки или сценарии, которые запускаются при добавлении или изменении подписки, при обновлении полей профиля или при назначении промокода. Если вы не хотите, чтобы эти триггеры сработали после текущего обновления, активируйте опцию **“Не запускать связанные с этим событием триггеры”**.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Double opt-in подписка

Double opt-in (DOI) — процесс оформления подписки на рассылку в два этапа:

1. Подписчик оставляет контактные данные — онлайн и офлайн, давая согласие на получение контента.
2. Подписчик подтверждает своё согласие, переходя по специальной ссылке

У такой логики есть несколько преимуществ перед Single opt-in (SOI) — когда учитывается только передача контактных данных:

- Улучшение репутации отправщика у провайдеров электронной почты
- Выполнение требований законодательства о рассылках
- Большая производительность маркетинговых рассылок
- Повышение лояльности, упрощение работы с постоянными клиентами

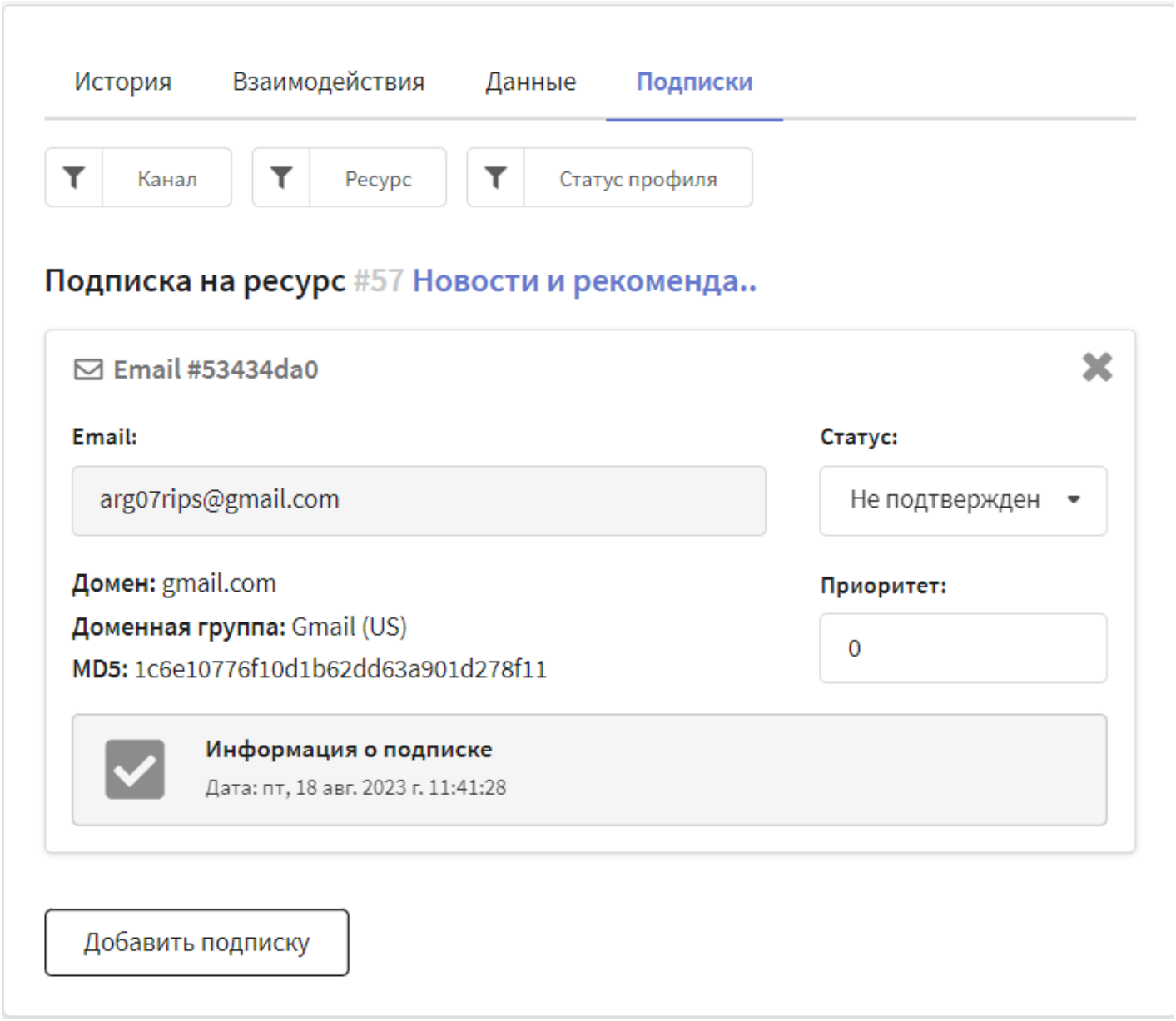
> **CAUTION**
>
> Double opt-in доступен только при использовании подписок на Ресурсы.

Этап 1: Подписка

При формировании подписки клиентов на определённый ресурс нужно установить статус подписки “Не подтверждён”.

Добавление подписки одному профилю

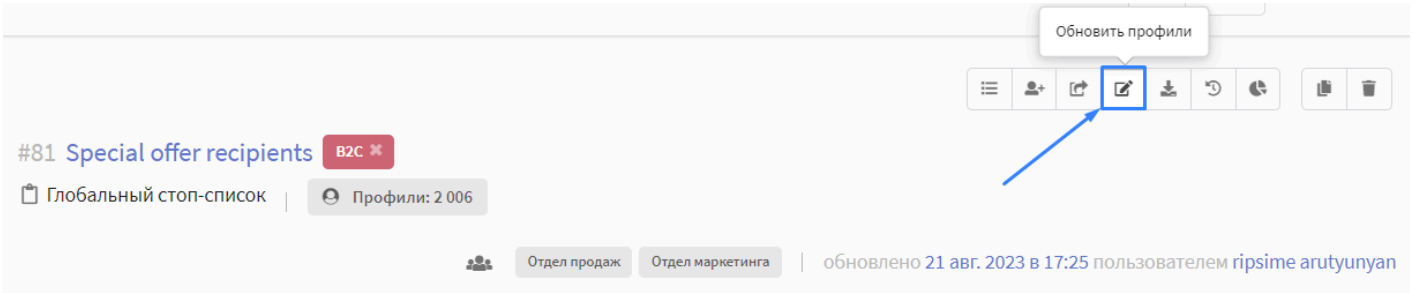
Откройте профиль подписчика и перейдите на вкладку **Подписки**. Выберите нужный ресурс и Email канал. Введите адрес почты для подписки или скопируйте из данных профиля. В меню **Статус** выберите “Не подтверждён” и сохраните изменения.



После сохранения система сохранит информацию о факте подписки — дату и время создания (UTC) и IP адрес подписчика. Эти данные подписчики могут потребовать при сомнениях в правомерности подписки.

Добавление подписки нескольким профилям

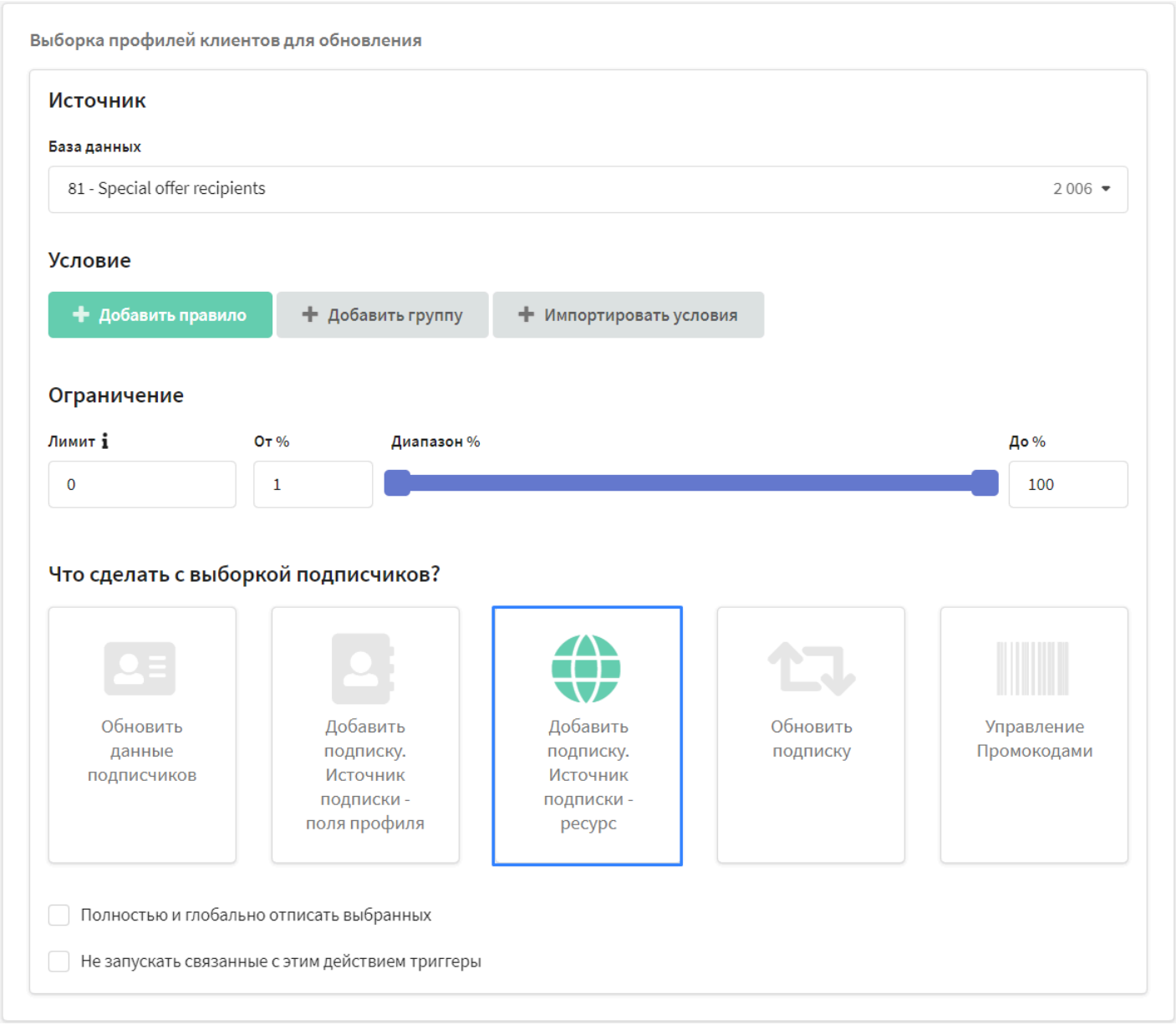
Для того, чтобы организовать Double opt-in для группы подписчиков, в контекстном меню базы данных или сегмента выберите пункт **Обновить профили клиентов**.



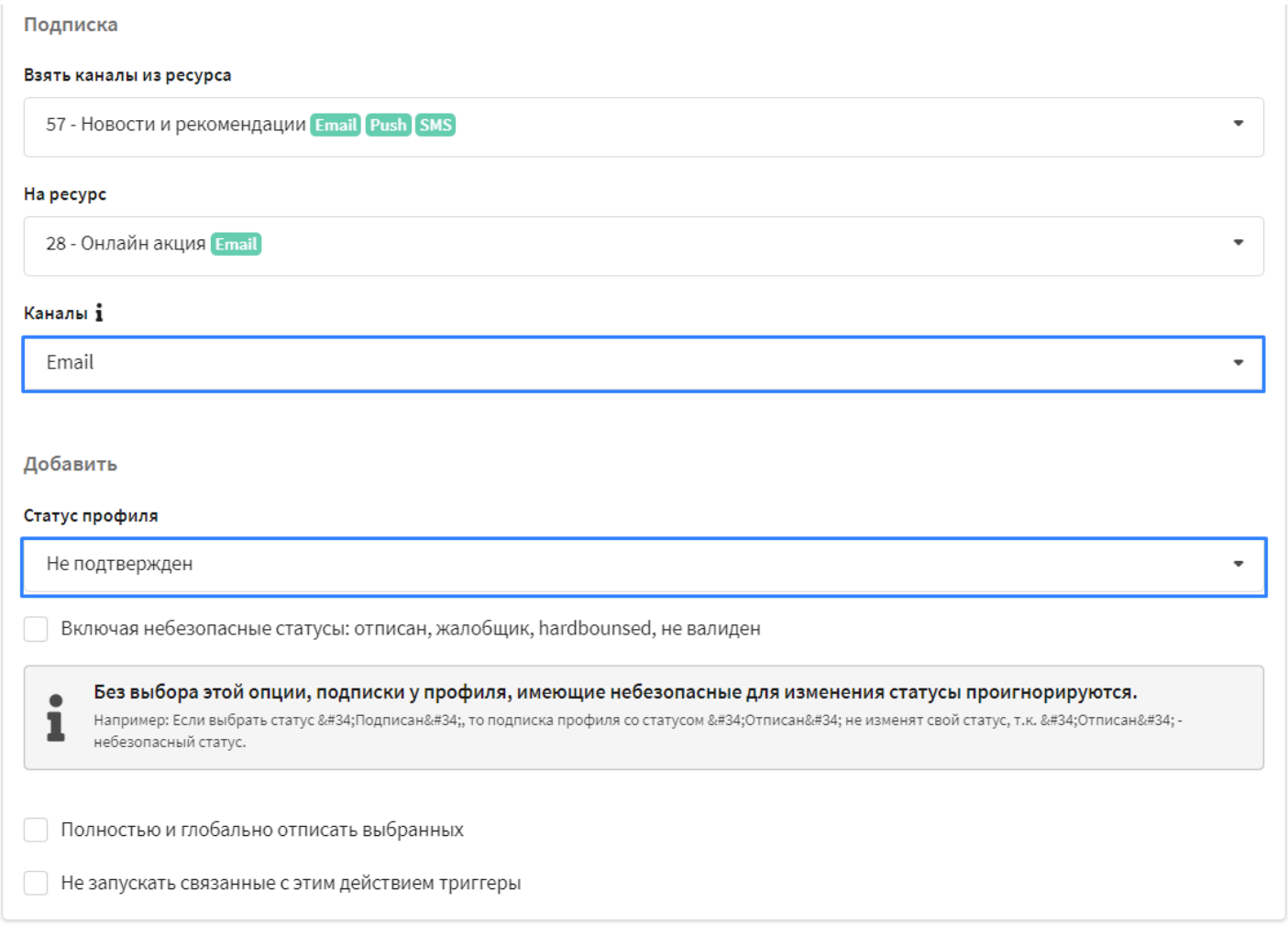
В открывшемся окне при необходимости скорректируйте аудиторию, которую хотите подписать. Для выборки используются правила и группы правил сегментации Altcaft MP. После этого можно перейти к разделу **Управление подпиской.**

CAUTION

Массовое обновление профилей настраивается отдельно для каждой базы данных, если в сегменте их несколько.



Выберите опцию **Добавить подписку**, укажите необходимый ресурс с Email каналом и установите статус подписки **Не подтверждён**. В качестве источника контактов можно выбрать данные профиля или контакты подписок на другой ресурс.



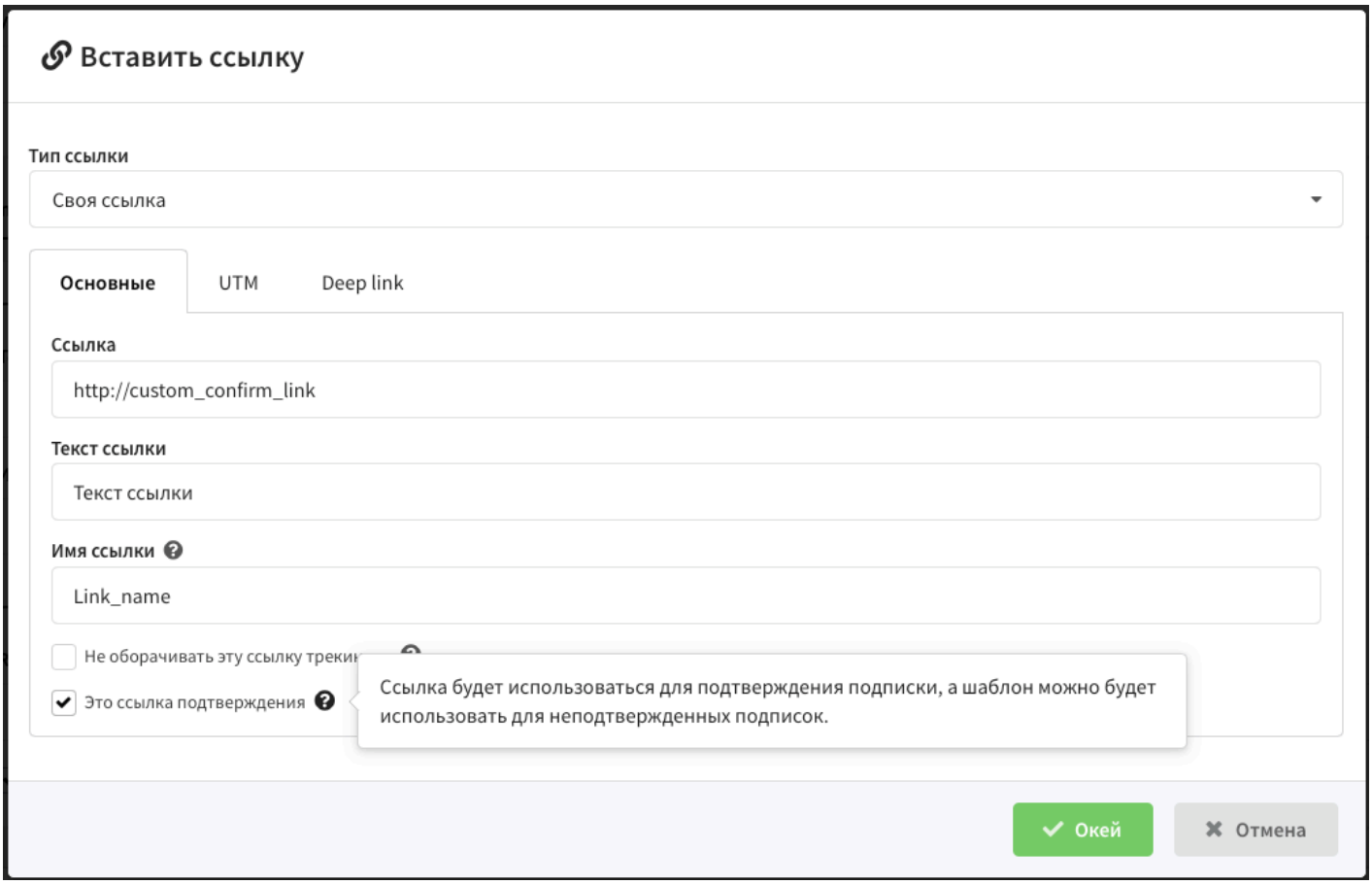
Когда все настройки введены, вы можете начать обновление профилей и просмотреть статус обновления в **Заданиях**.

NOTE

При импорте подписчиков и обновлении данных по API в подписке для DOI нужно передавать `"status":"unconfirmed"`

Этап 2: Подтверждение

Клиентам с неподтверждённой подпиской будут отправляться только сообщения, которые содержат ссылку подтверждения. Чтобы создать шаблон сообщения перейдите в раздел “Шаблоны”. Ссылка подтверждения — это любая ссылка в шаблоне с атрибутом confirm-link. Вы можете добавить атрибут вручную или создать ссылку в конструкторе. Для этого в редакторе сообщения нажмите “Вставить ссылку” и отметьте пункт “Это ссылка подтверждения”:



После этого шаблон сообщения со ссылкой подтверждения нужно добавить в рассылку, использующую тот же ресурс, на который были подписаны клиенты. Получатели сообщения при переходе по такой ссылке подтвердят своё согласие на получение рассылок и их статус подписок изменится с **Не подтверждён** на **Подписан**.

NOTE

Нажатие на ссылку подтверждения может выступать условием запуска триггерных рассылок и сценариев автоматизации. Для этого нужно в настройках триггера или сценария выбрать **При выполнении выбранного действия → Клик на Confirm ссылку**.

Стоп-списки

	Стоп-список — это список email адресов и доменов, а также телефонных номеров, которым вы НЕ хотите отправлять сообщения. Стоп-лист фильтрует email-адреса и номера телефонов **в момент отправки рассылки**. Стоп-списки используются, чтобы придерживаться нормативных требований закона CAN-SPAM.

В платформе используется три типа списков блокировки отправки:

- **Глобальный стоп-список Платформы** — действует для рассылок **всех аккаунтов** платформы. В глобальный список попадают все невалидные email адреса и телефонные номера, по которым от провайдеров пришёл **hard bounce**. Отдельно хранятся валидные адреса и номера, отправку на которые необходимо блокировать.
- **Глобальный стоп-список Аккаунта** — создаётся автоматически в каждом аккаунте платформы. Добавленные контакты будут блокироваться во всех рассылках **Аккаунта**.
- **Пользовательские стоп-списки** — создаются пользователями в аккаунтах платформы. Могут быть прикреплены к **Базам профилей**, **Шаблонам сообщений**.

Так как в рассылках может участвовать сегмент (в котором может быть несколько баз данных со стоп-списками) и несколько сообщений, то и учитывается несколько стоп-листов по определённым правилам.

Глобальный стоп-список Платформы

Глобальный стоп-список платформы действует во всех Аккаунтах платформы. Он состоит из двух частей:

- **Список невалидных email или номеров** — в этот список попадают email-адреса и телефонные номер со всех Аккаунтов платформы, по которым от провайдера пришёл **hard bounce** (ответ о том, что ящик или мобильный номер не существует).

CAUTION

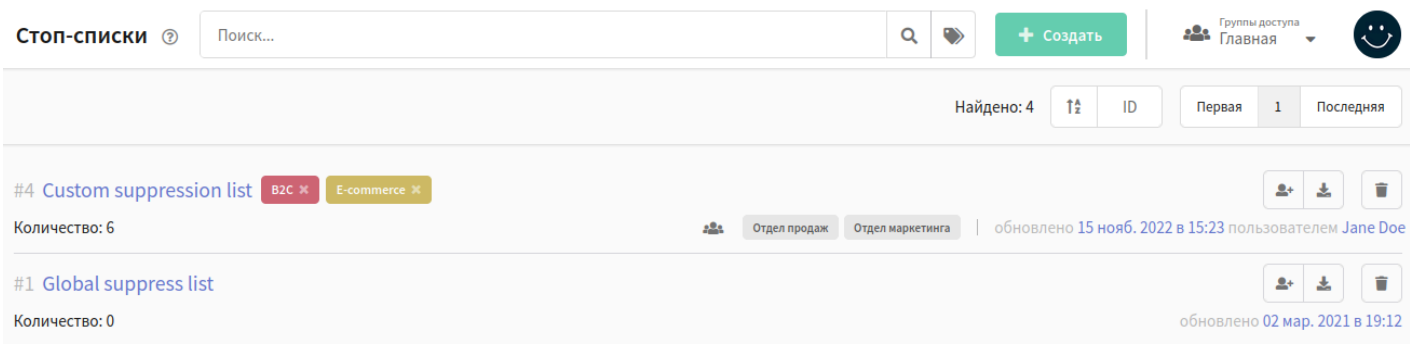
Обратите внимание, что при таком ответе **Статус подписки** клиента меняется на **hard bounce**. Такие подписки не будут использоваться в рассылках, так как отправка на несуществующие адреса ухудшает репутацию отправителя в почтовых сервисах.

Если изменить статус таких подписок на **Подписан**, они останутся в глобальном стоп-списке и не будут обрабатываться. Чтобы переподписать таких клиентов, нужно будет удалить их из глобального стоп-списка платформы.

- **Глобальный чёрный список email или номеров** — блокирует любые email-адреса, домены или телефонные номера для всех Аккаунтов платформы. Данные в него может добавить Администратор платформы, используя соответствующий функционал панели администратора Альткрафт.

Стоп-списки Аккаунта

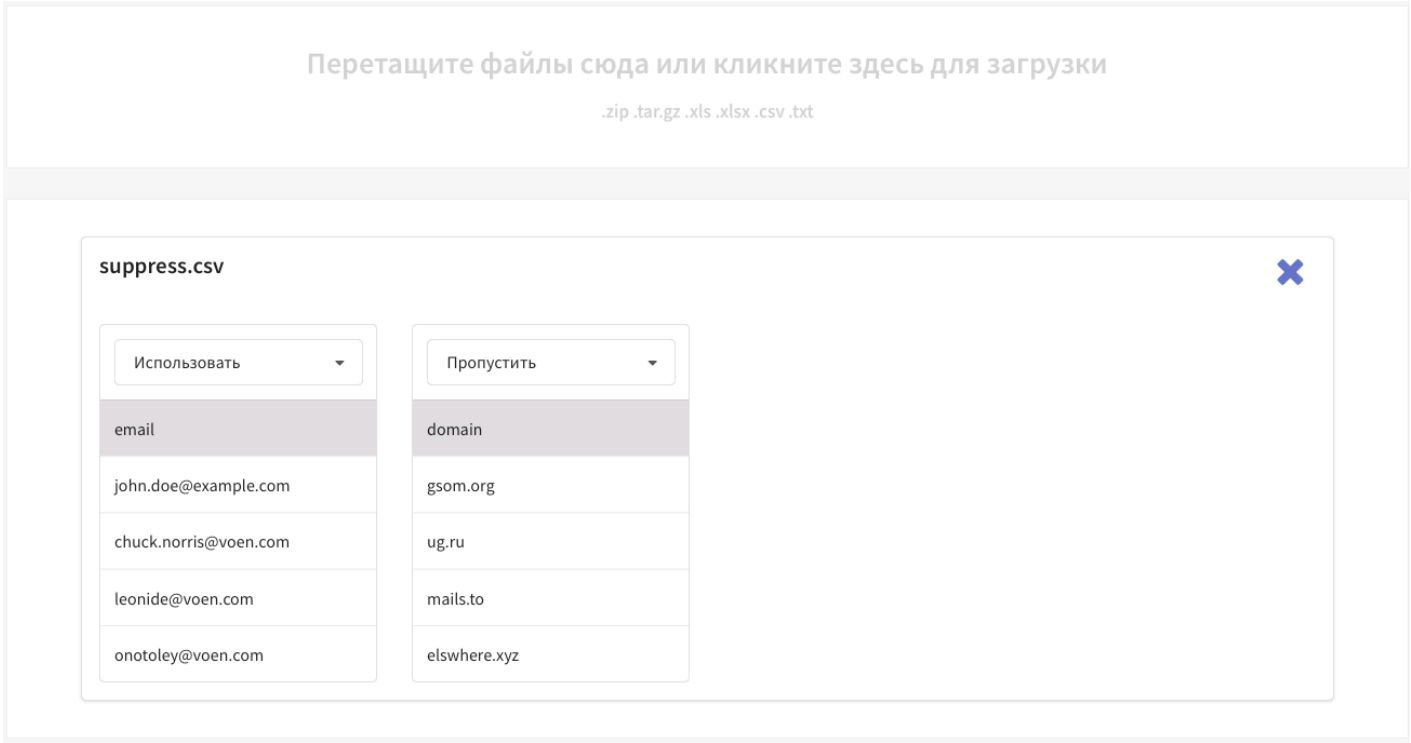
Чтобы перейти к управлению стоп-списками Аккаунта платформы, в главном меню выберите **Данные —> Стоп-списки**. В открывшемся меню отображаются существующие в Аккаунте стоп-списки. Панель быстрых действий позволяет сразу приступить к работе с необходимым стоп-списком:



Импорт из файла

Чтобы добавить в стоп-список контакты из файла, нажмите

в панели быстрых действий. В открывшуюся форму перетащите файл со списком контактов в формате CSV или XLS. Таких файлов может быть несколько, и в файлах могут быть несколько колонок. После загрузки файла вы можете выбрать, какую колонку использовать для импорта в стоп-список:



Импорт вручную

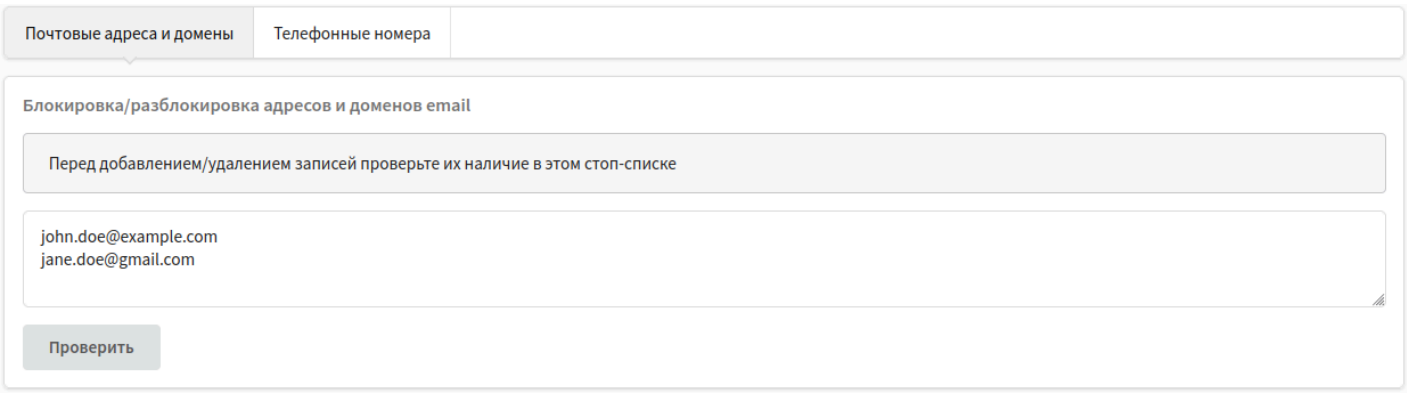
Чтобы добавить контакты клиентов вручную, откройте необходимый стоп-список. В стоп-список можно добавлять email-адрес, email-домены и телефонные номера клиентов. В зависимости от типа контакта, откройте одну из вкладок и в блоке **Блокировка/разблокировка** введите контактные данные, которые хотите добавить в список.

TIP

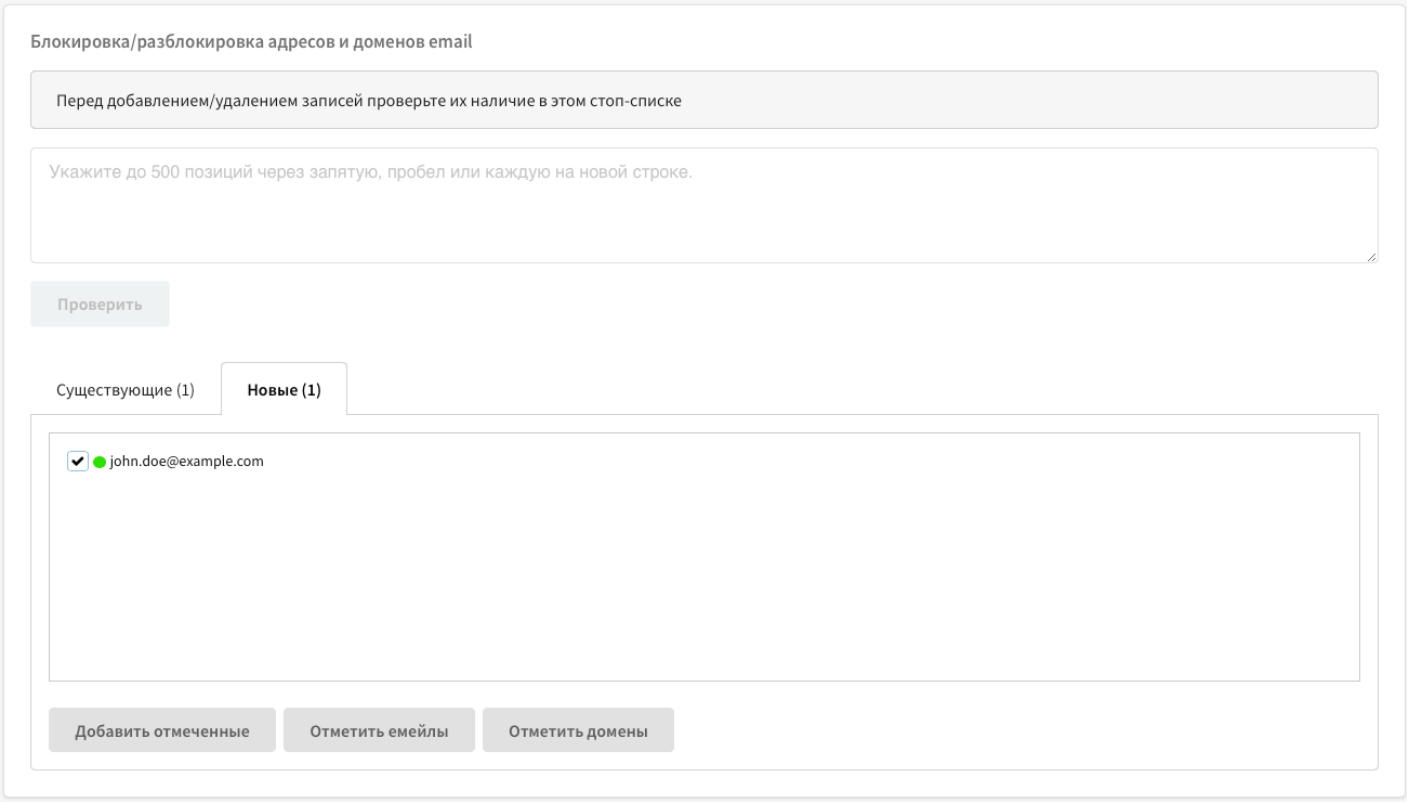
Несколько позиций указываются через запятую, пробел или каждая на новой строке. Можно указать до 500 email-адресов, доменом или номеров.

Номер телефона указывается в международном формате (например, +79123456789).

Затем нажмите **Проверить**:



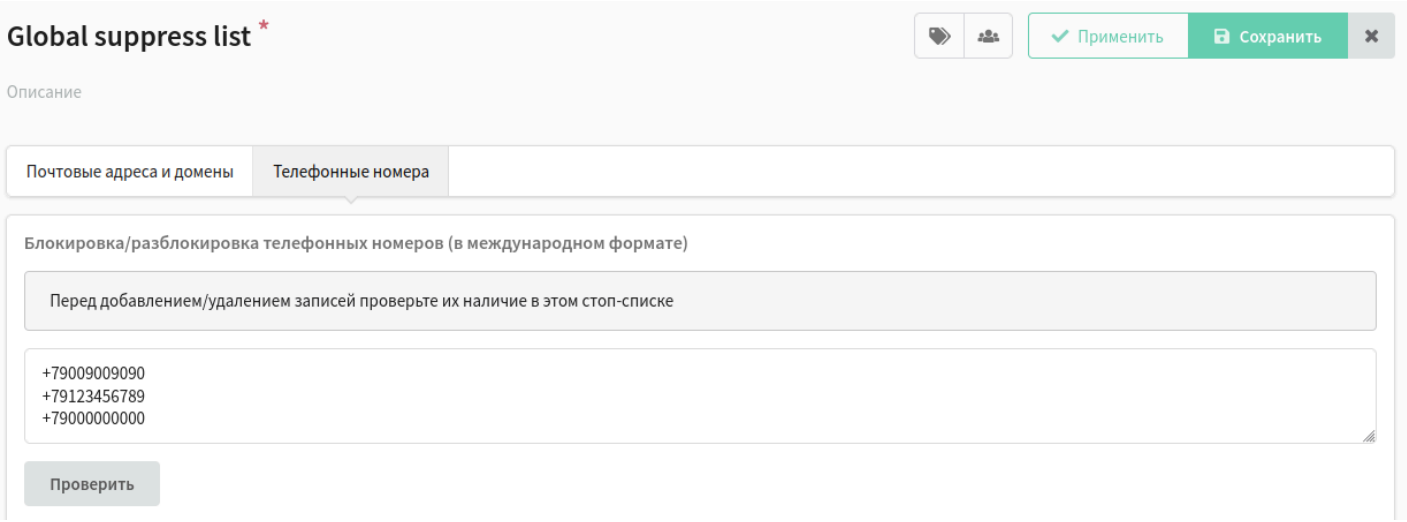
Откроется две вкладки: **Существующие** и **Новые**. Если домен, адрес или номер отсутствуют в стоп-списке (появится во вкладке **Новые**), вы можете выделить его и добавить. Если существует — его можно будет так же удалить из стоп-списка. Дополнительно на вкладках можно выделить все email адреса/домены или номера телефонов:



Глобальный стоп-список

В каждом аккаунте Altcraft автоматически создаётся один глобальный стоп-список, который работает всегда — для всех рассылок в рамках **Аккаунта**. Соответственно, его не нужно прикреплять вручную к **Базам профилей** и **Шаблонам сообщений**.

В глобальный стоп-список можно добавлять подписки вручную и импортировать из файла:



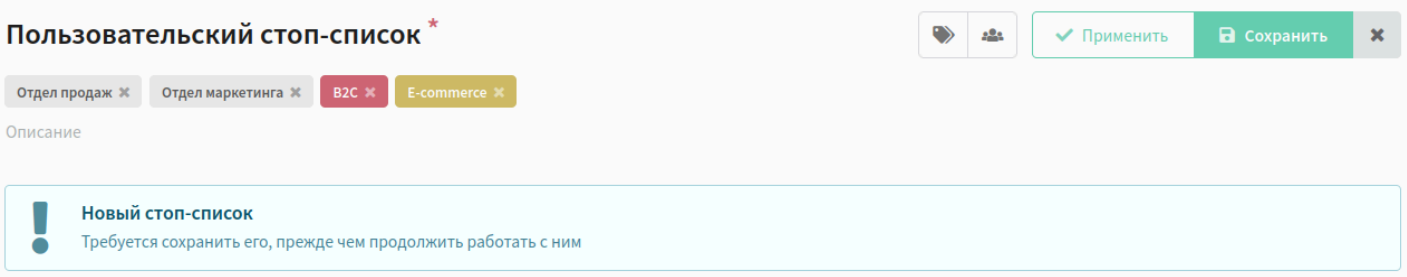
Для того чтобы подписки автоматически добавлялись в глобальный стоп-список, нужно использовать в сообщении ссылку — URL глобальной отписки переменной `{globalunsubscribe}`. Адрес подписчика, который перейдет по ссылке, сразу попадет в глобальный стоп-список и будут исключен из всех дальнейших рассылок аккаунта.

TIP

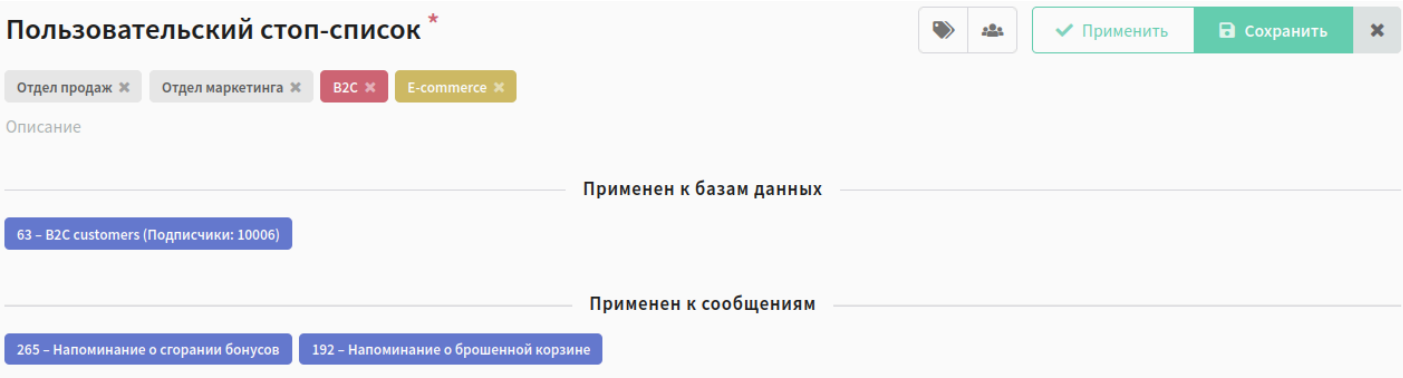
Обратите внимание, что коммуникации прекратятся **только для того email, на которой пришло письмо с ссылкой отписки**. Если у профиля есть подписка с другим email'ом, сообщения на него будут приходить как обычно.

Пользовательские стоп-списки

Чтобы добавить в аккаунт пользовательский стоп-список, в меню списков нажмите **+ Создать**. Для нового списка укажите **Имя**, введите **Описание**, при необходимости установите группу доступа и теги для быстрого поиска. Сохраните его, чтобы продолжить работу.

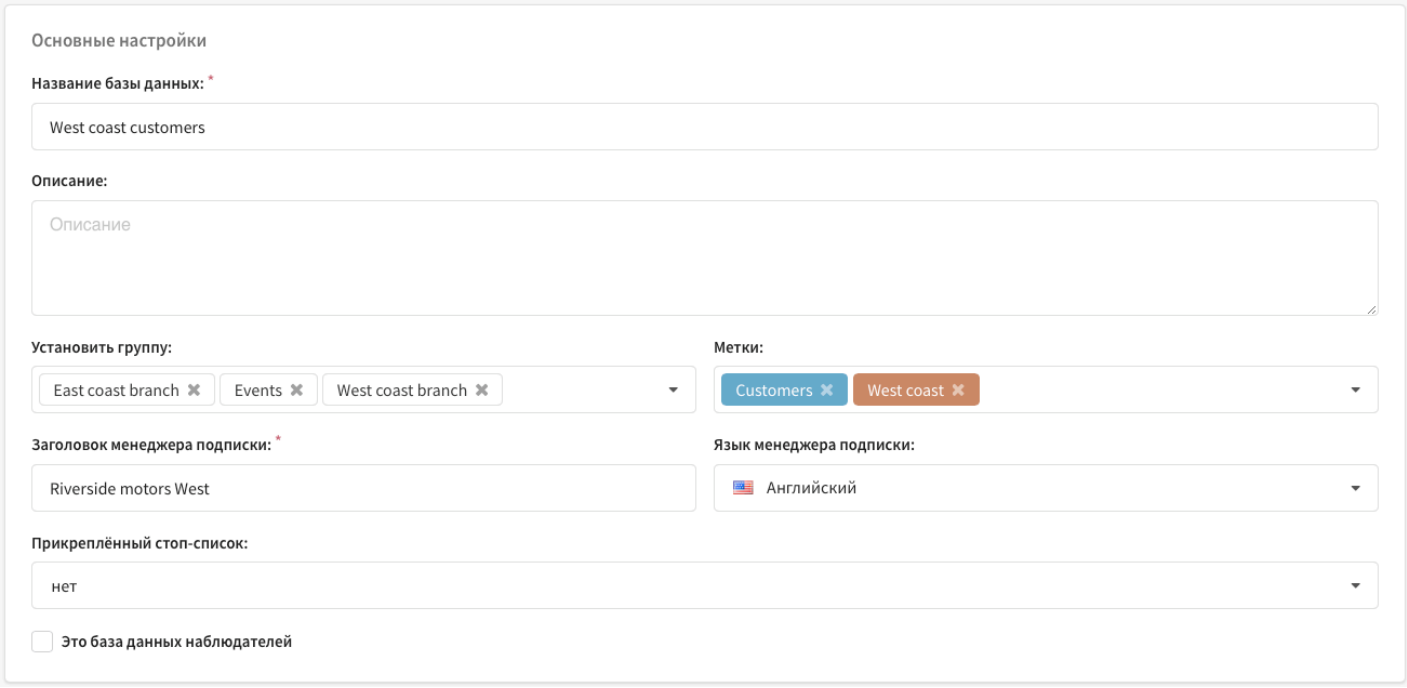


После этого вы можете добавлять контакты клиентов в созданный стоп-список вручную или импортом из файла, а также прикреплять список к базам профилей и шаблонам. У вас также есть возможность в любой момент посмотреть, к каким объектам платформы прикреплен список:



Стоп-список для Базы профилей

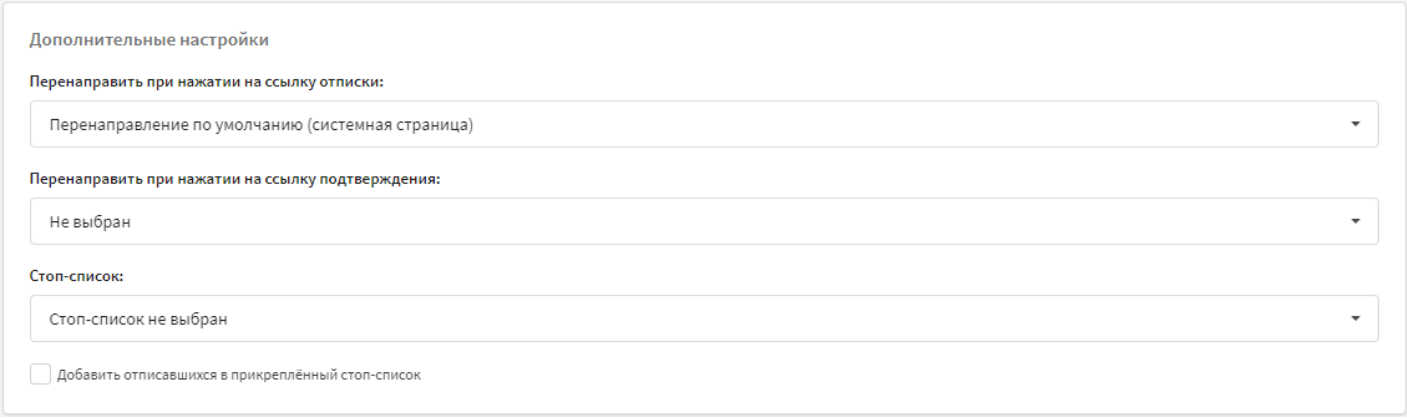
Пользовательские стоп-списки можно прикреплять к определённой **Базе профилей** Аккаунта. Такой список будет действовать только при отправке **Рассылок** аудитории, содержащей эту базу профилей. Для этого в форме редактирования Базы данных в **Основных настройках** выберите **Прикреплённый стоп-список** из выпадающего меню:



Стоп-список для Шаблона сообщения

Второй вариант использования пользовательских стоп-списков — прикрепление к конкретным **Шаблонам сообщений**. Это позволит контролировать, какой аудитории не следует рассылать определённый контент: профили, чьи контакты есть в прикрепленном стоп-списке, не будут получать рассылки с этим шаблоном.

Откройте необходимый **Шаблон**, перейдите на вкладку “Опции” и в блоке “**Дополнительные настройки**” выберите нужный стоп-список. Если вы хотите, чтобы контакты профилей, отписавшихся по ссылке в шаблоне, попали в стоп-список, активируйте опцию “**Добавить отписавшихся в прикрепленный стоп-список**”.



Правила взаимодействия стоп-списков

Во время рассылки могут использоваться одновременно несколько стоп-списков. Исключение подписчиков из рассылки происходит по следующему алгоритму:

1. Проверяются **глобальные стоп-списки** и из рассылки исключаются все подписки, которые там находятся.

2. Далее стоп-списки, прикреплённые к **Базам профилей**, исключают из рассылки подписчиков своих баз данных. Каждый только из той базы, к которой прикреплён.

NOTE

Если в двух базах данных одинаковые подписки, а стоп-список, который их содержит, прикреплён только к первой базе, то сообщения от рассылки, содержащей сегмент с этими двумя базами, уйдут полностью. Потому что стоп-список подействует только на ту базу данных, к которой прикреплён, то есть первую.

3. Потом список, прикреплённый к **Шаблону сообщения**, исключает из рассылки только те подписки, которым досталось бы это сообщение.

Связи между профилями

Связи между профилями позволяют реализовать на платформе стратегии взаимодействия между клиентами. Связь можно установить между профилями внутри одной базы данных или связать профили из разных баз. Существует два типа связи:

- **Прямая** — исходящая. Показывает отношение профиля к другим профилям. Например, связь клиента-реферала с его клиентами.
- **Обратная связь** — входящая. Показывает, как другие профили относятся к выбранному профилю. Например, клиенты одного агента или менеджера компании оценивают качество его работы.

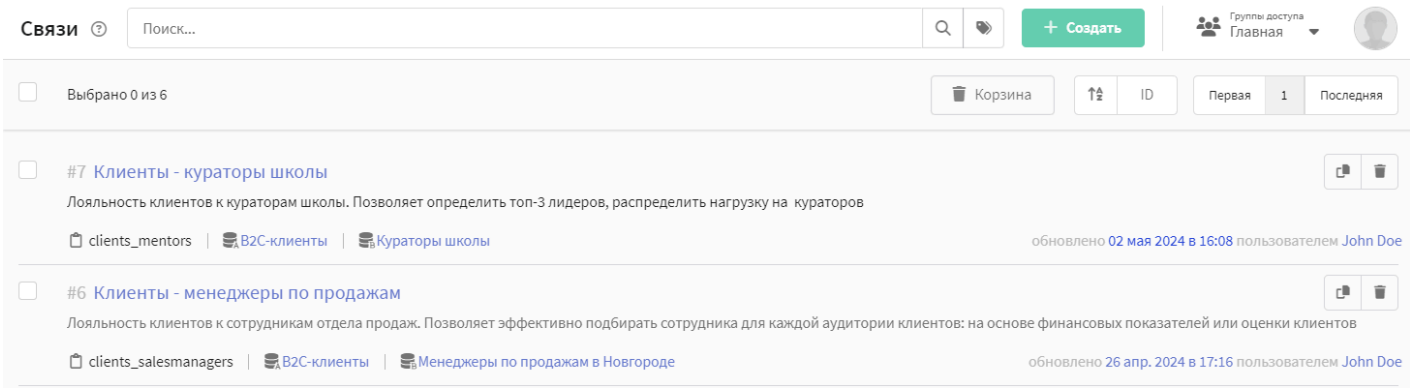
У каждой связи есть **свойства** — отношения между клиентами. Свойства могут быть в виде числовых или булевых (ложь — истина) значениях.

Объекты связей создаются в интерфейсе пользователя. Добавить, удалить, усилить или изменить связь вы можете как вручную, так и используя особые API запросы, описанные в этом разделе документации.

Создание и редактирование связи

В главном меню откройте раздел **Данные** и выберите пункт **Связи**. В списке отображаются существующие связи с информацией о создании и обновлении, а также о связанных базах данных. С помощью меню действий в правой части плитки вы можете клонировать или удалить связь.

Чтобы добавить новую связь между базами данных профилей, нажмите **Создать** в верхней части страницы. Чтобы отредактировать имеющуюся связь, нажмите на ее имя.



В основных настройках введите **название** и **описание** новой связи. Укажите также **Короткое имя** связи, оно будет использовано в редакторе шаблонов для подставления переменных, а также в API запросах управления связями.

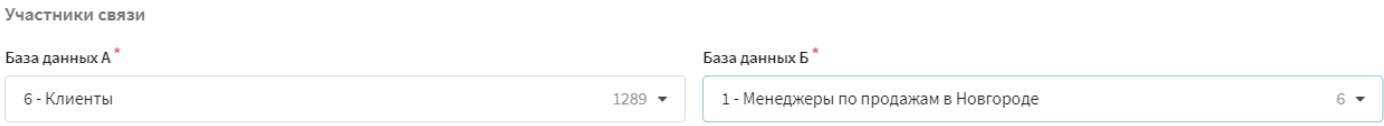
В основных настройках введите **название** и **короткое имя** новой связи. **Короткое имя** связи будет использовано в редакторе шаблонов для подставления переменных, а также в API запросах управления связями.

Ниже выберите базы данных, профили которых необходимо связать:

- **База данных А** — содержит профили, к которым направлена связь. Например, менеджеры компании или филиалы, работу которых оценили клиенты. Иными словами — **объекты**.
- **База данных Б** — содержит профили, инициирущие связь. Например, клиенты, вносящие средства или оставляющие оценки качества сервиса. Иными словами — **субъекты**.

TIP

Если вам нужно отразить взаимоотношения между профилями одной базы данных, выберите её и как **Базу данных А**, и как **Базу данных Б**.



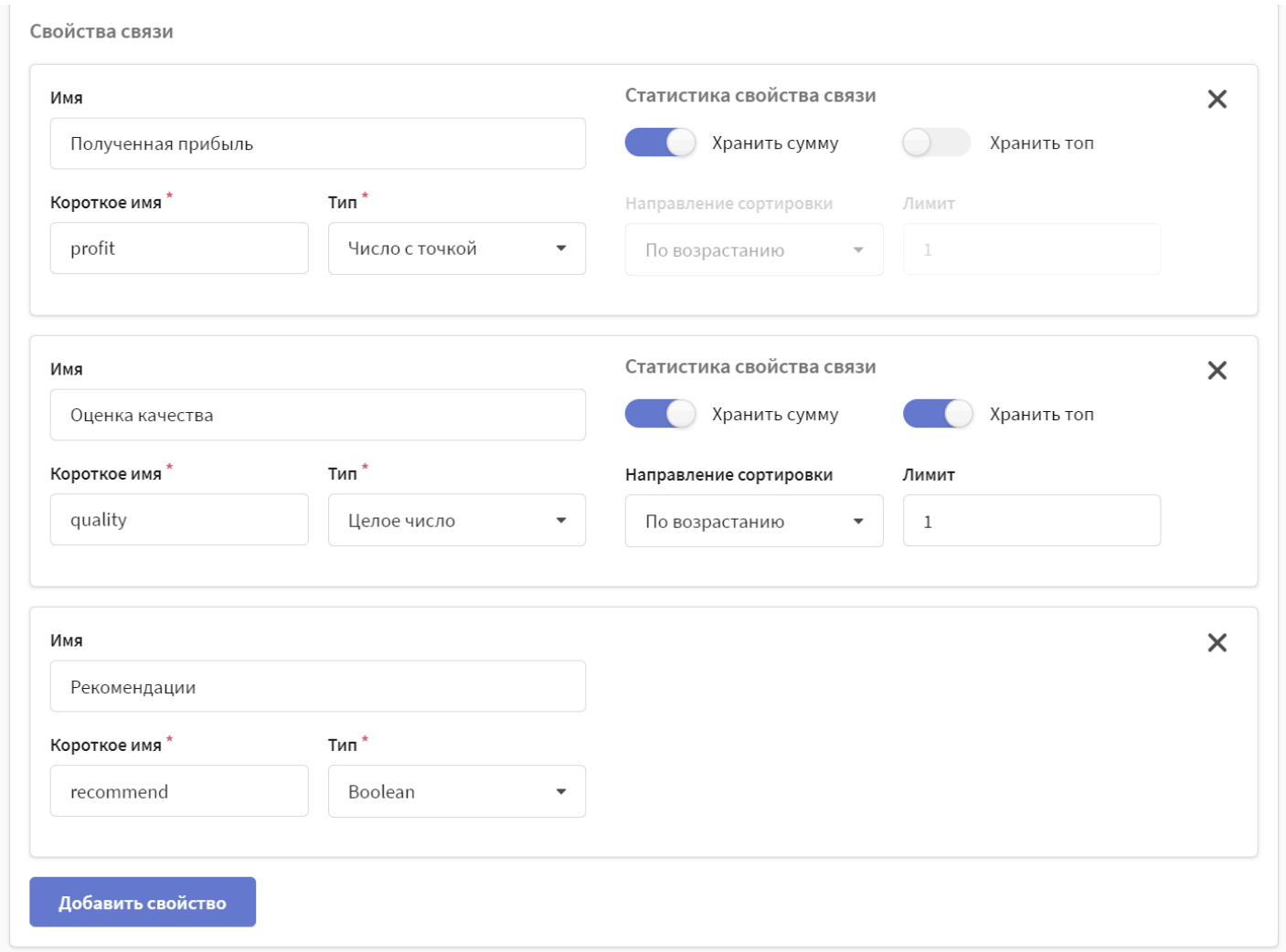
Свойства связи

Связи могут содержать **до 11 свойств**, отражающих метрики взаимоотношения профилей в связанных базах данных. Для каждого нового свойства введите **имя** — для отображения в списке и **короткое имя** — для шаблонов и API запросов.

Тип связи определяет данные, которые будут использоваться для взаимодействия:

- **Целое число** — любое целочисленное значение. Удобно для оценки качества или показателя лояльности.
- **Число с плавающей точкой** — число с дробной частью. Может использоваться для финансовых и иных точных показателей.
- **Boolean** — булевое значение: ложь или истина, да или нет. Используется для регистрации разового действия, например посещения страницы или покупки.

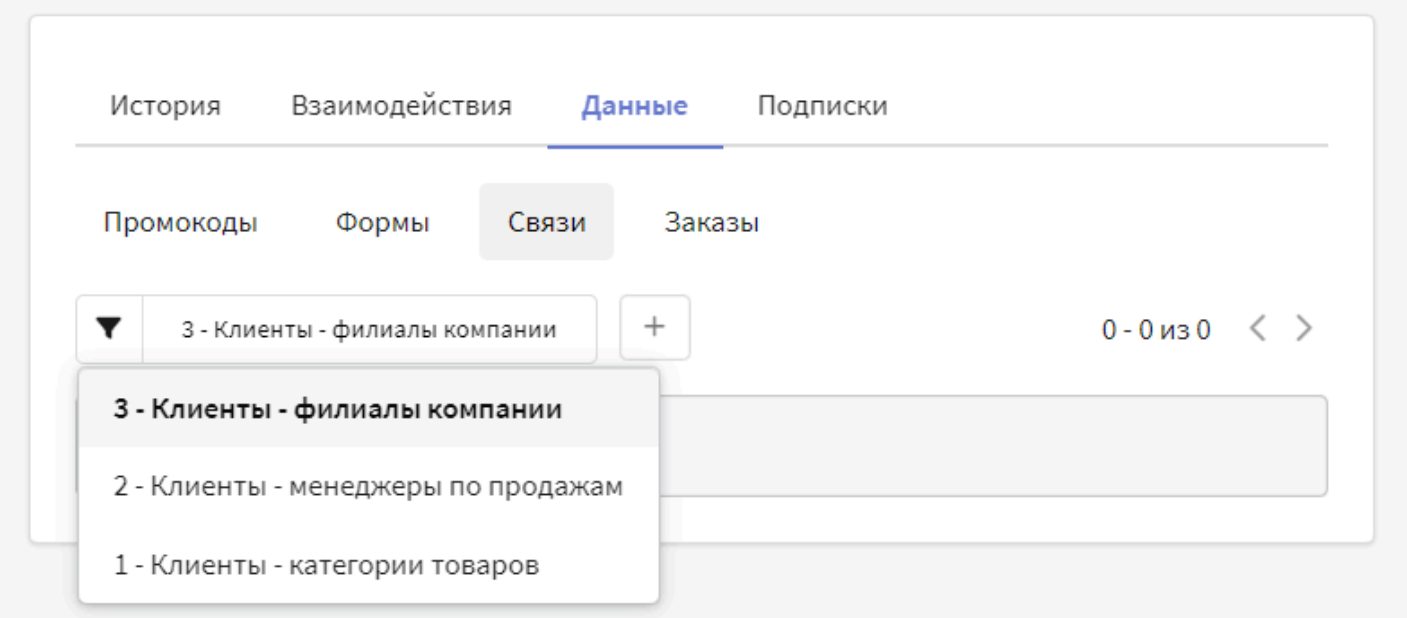
Для числовых типов в связи можно хранить статистику: **сумму** значений по добавленным связям, а также **топ** связей. **Направление сортировки** определяет, хранить в топе наибольшие (по убыванию) или наименьшие (по возрастанию) значения, а **лимит** (от 1 до 20) устанавливает количество связей для статистики:



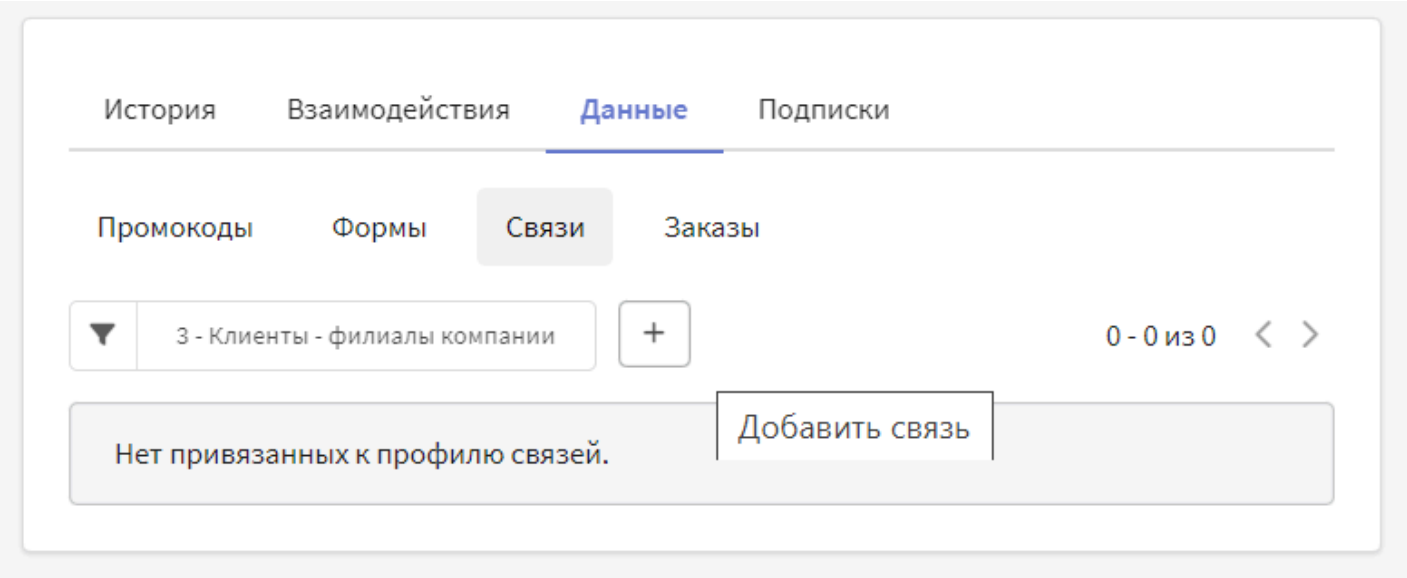
Добавление связи вручную

Откройте необходимый профиль подписчика в **Базе данных А**. На вкладке **Данные** перейдите в раздел **Связи**. Здесь отображаются доступные для профиля текущей базы данных связи.

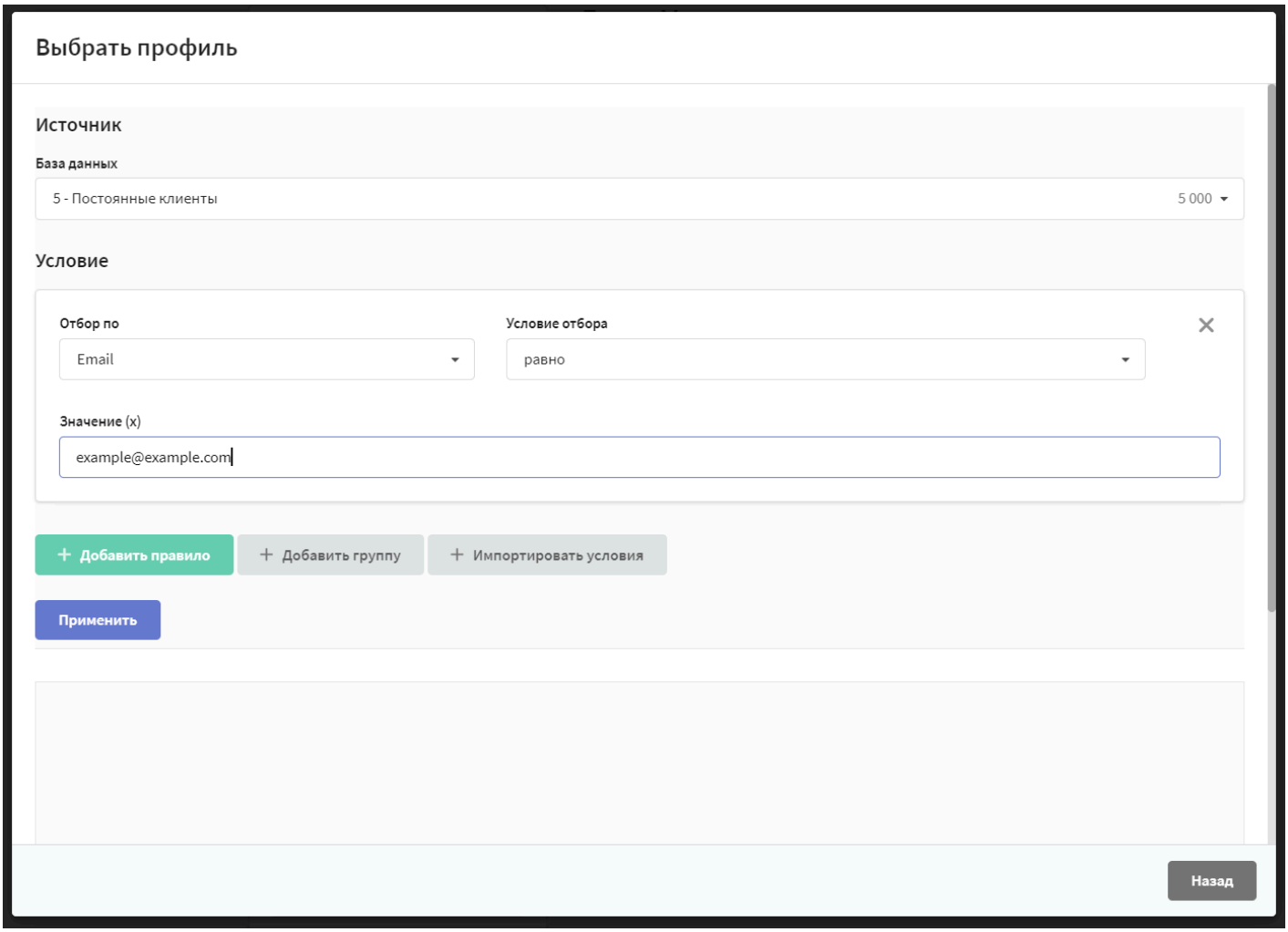
Нажмите на кнопку с фильтром, чтобы открыть выпадающее меню:



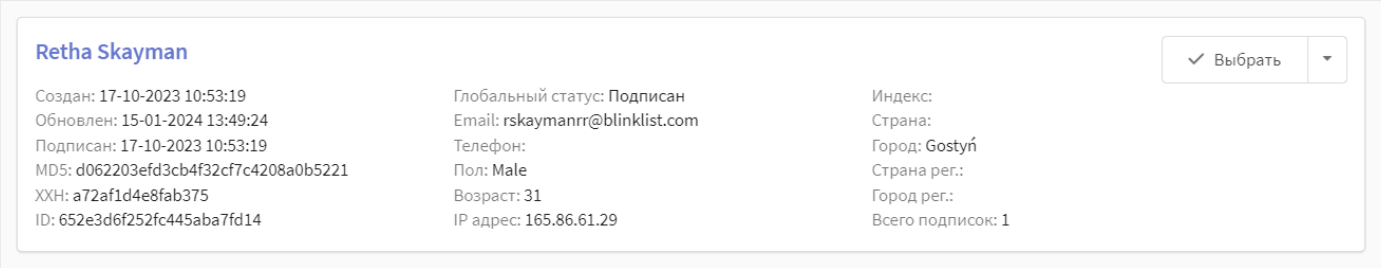
Выберите необходимую связь и добавьте нажатием на **+**.



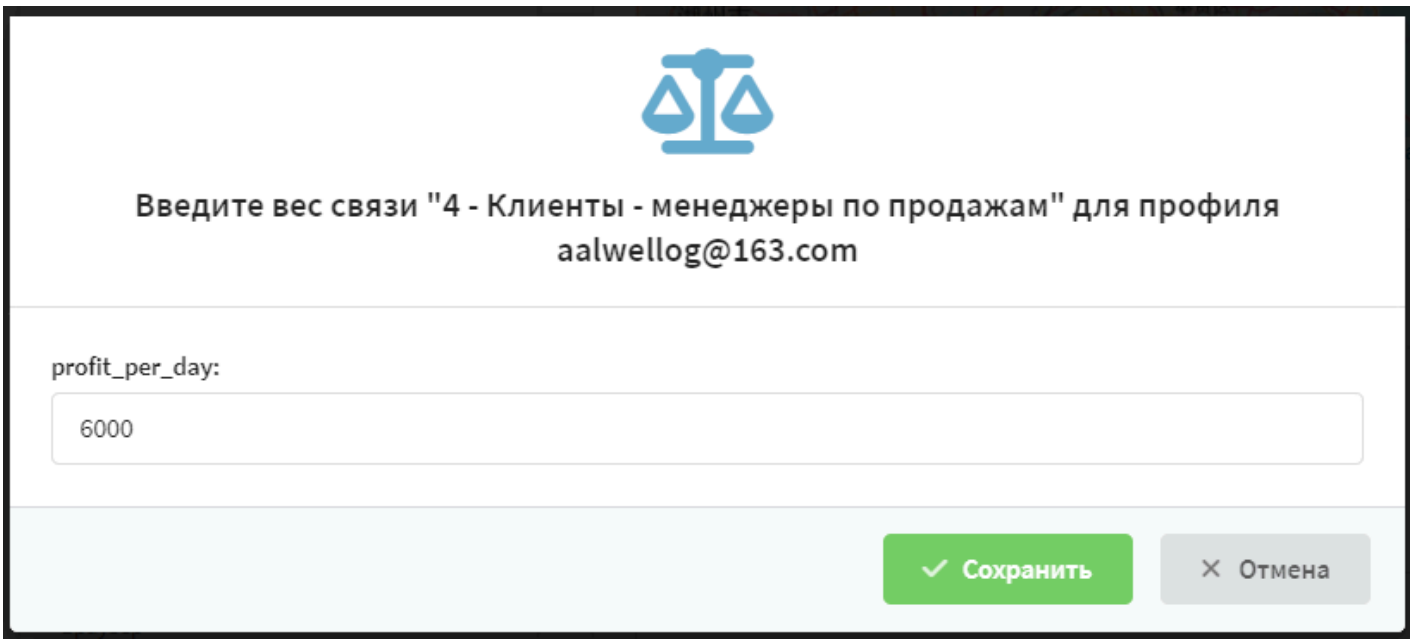
В открывшемся окне укажите правило для отбора профиля из **Базы данных Б**. Используются стандартные правила, как при поиске или сегментации профилей:



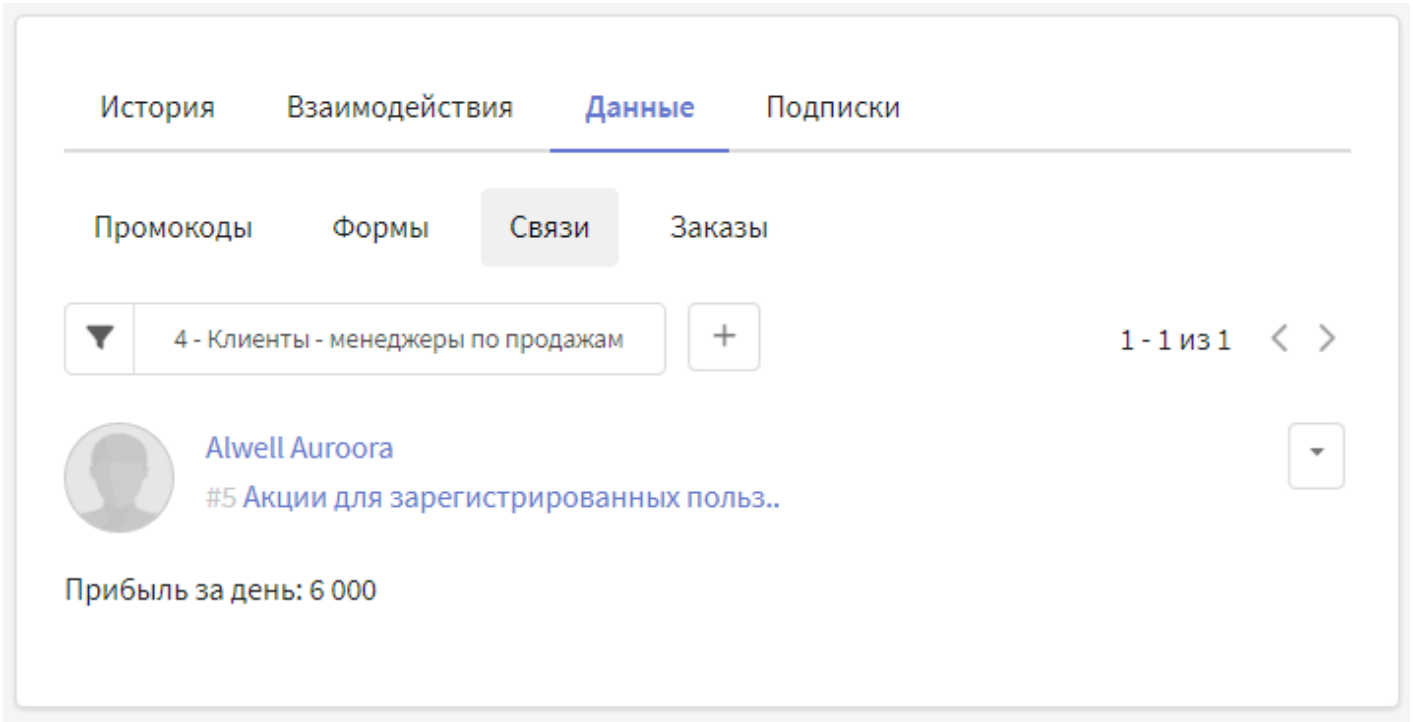
Нажмите **“Применить”** и **“Выбрать”**:



После этого укажите вес связи в открывшемся окне:



После добавления связи в профиле станет доступна краткая сводка и топ по свойствам связи, а также список связанных профилей из **Базы данных Б**. Вы можете изменить вес связи или удалить связь, воспользовавшись выпадающим меню в правой части записи:



Использование связей и их значений в платформе

1. Сегментация по связям

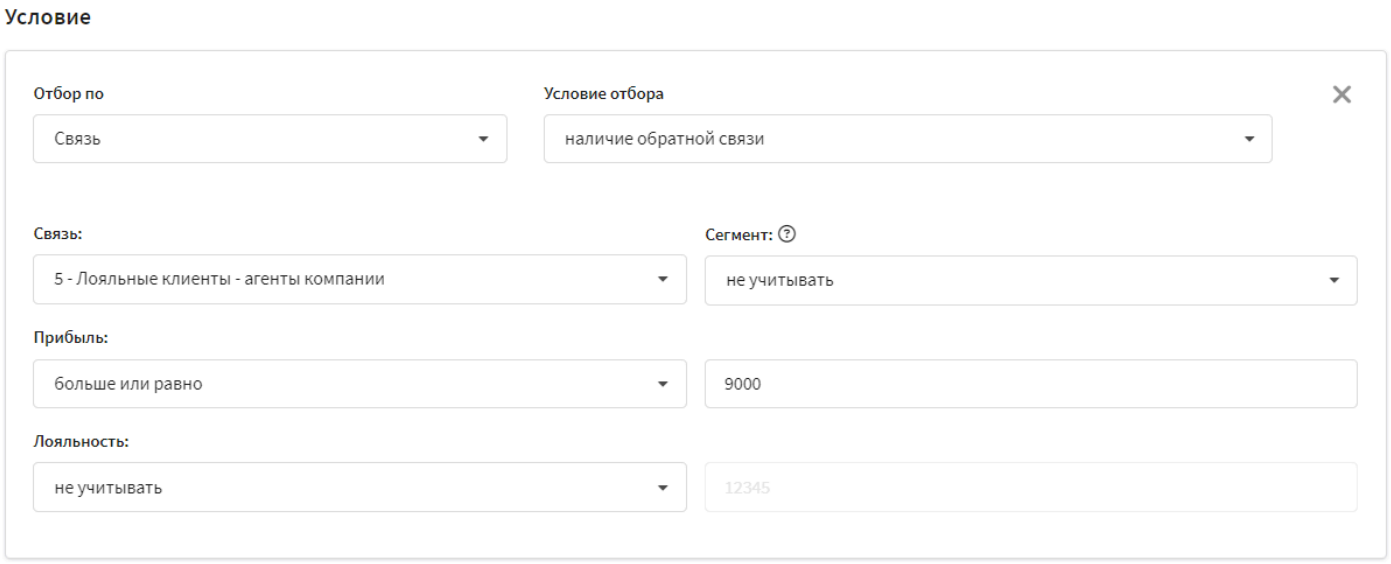
С их помощью можно сформировать статический или динамический сегмент. Благодаря этому из всех клиентов одной базы данных “отделяется” целевая аудитория по признаку отношения ее участников друг к другу.

Дополнительно можно добавить проверки определённых свойств связи:

- ложь или истина — для булевых свойств
- больше/меньше, больше/меньше или равно, равно/не равно — для числовых свойств

В правиле сегментации выберите наличие или отсутствие **прямой или обратной** связи, и добавьте необходимые параметры.

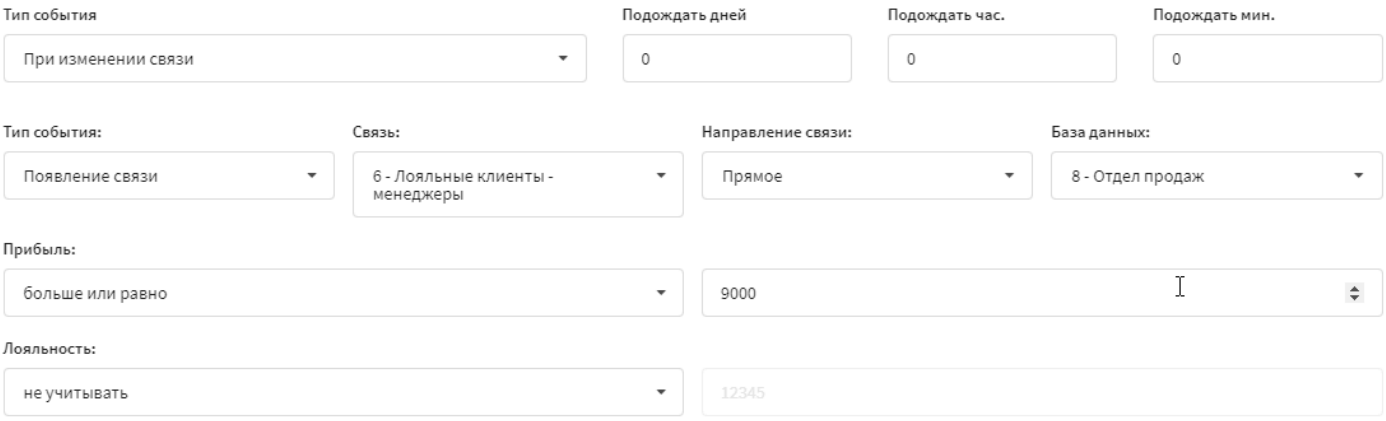
> **TIP**
>
> Например, с помощью такого правила вы можете создать сегмент агентов или менеджеров компании, которые заключили сделки больше, чем на 9000.



2. Подстановка динамического контента в шаблонах

Значения связей могут быть добавлены в шаблон сообщения с помощью специальных переменных.

Триггер может запускаться при изменении как **прямой**, так и **обратной** связи между профилями. При этом может учитываться **появление**, **изменение** или **удаление** связи, а также достижение свойством связи заданного в триггере значения:



TIP

Например, если у вас есть реферальная программа, вы можете настроить две разные последовательности:

- первую для клиента-реферала. В сценарии или триггере нужно выбрать запуск при появлении прямой связи (когда реферал передаёт другому клиенту токен участия в реферальной программе). Прямая связь здесь будет учитывать всех приведённых рефералом клиентов. В свойствах связи можно учитывать приносимую рефералом прибыль. Например, вы можете назначать разные бонусы при покупках на разную сумму.
- вторую — для приведённых клиентов. Как условие запуска тут будет выступать появление обратной связи. Использовать такой триггер или сценарий вы можете для предоставления клиентам бонусов и предложения участия в реферальной программе.

Свойства связи в шаблонах сообщений

В шаблоне сообщения можно вывести значения свойств связи: сумму, количество или топ записей. В **меню переменных** выберите прямую или обратную связь и необходимое значение для подстановки.

TIP

Например, для онлайн-центра обучения можно уведомлять методистов о том, сколько часов клиенты провели, обучаясь у тех или иных инструкторов, а для клиентов — предоставлять актуальное расписание предпочитаемых ими инструкторов.

Переменные связей также можно вставить вручную, используя следующий скрипт редактора:

`{relation.[короткое_имя_связи].[direct для прямых или reverse для обратных].[имя_свойства].[count – количество, total – сумма, top – топ связей]}`

Количество прямых связей со свойством profit:

```
{relation.managers_loyal.direct.profit.count}
```

Топ обратных связей со свойством `profit`. Используется цикл для вывода нескольких значений.

TIP

Цикл подставляет данные тех профилей, которые указаны в связи, а не из базы данных, по которой отправляется рассылка. Таким образом, если вы отправляете письмо своему агенту, в нём могут быть использованы личные данные клиентов, хранящиеся в отдельной базе данных.

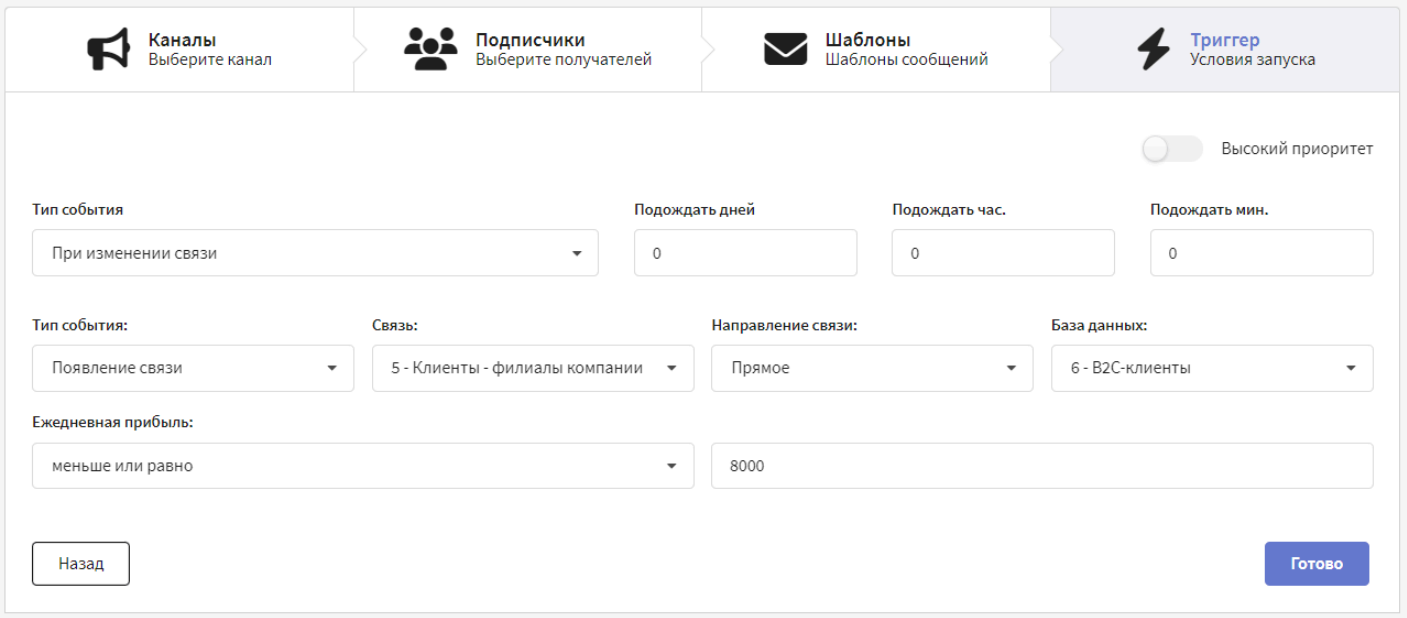
```
{for $index $item = relation.managers_loyal.reverse.profit.top}
{$item.lead._fname} {$item.lead._lname} ({$item.value} profit)
{else} There is no ‘profit’ for you :(
{end}
```

Сумма значения свойства profit для прямой связи:

```
{relation.managers_loyal.direct.profit.total}
```

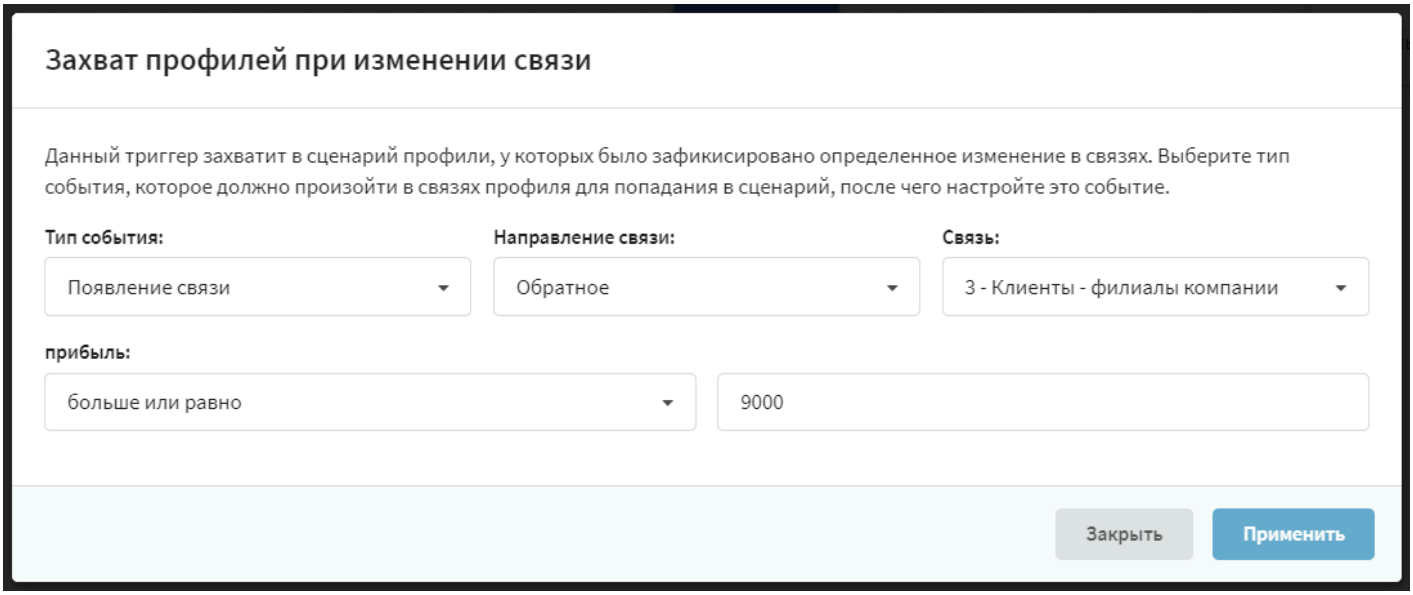
3. Условие для запуска триггерной рассылки или сценария

На появление или удаление связи может быть настроена триггерная рассылка:



Триггер может запускаться при изменении как **прямой**, так и **обратной** связи между профилями. При этом может учитываться **появление**, **изменение** или **удаление** связи, а также достижение свойством связи заданного в триггере значения.

Либо связи могут реализовываться в сценарий автоматизации:



TIP

Приведем в пример следующую ситуацию:

У вас есть реферальная программа, для нее вы можете настроить две разные последовательности:

- Первую для клиента-реферала: в сценарии или триггере нужно выбрать запуск при появлении прямой связи — когда реферал передаёт другому клиенту токен участия в реферальной программе. Прямая связь здесь будет учитывать всех приведённых рефералом клиентов. В свойствах связи можно учитывать приносимую рефералом прибыль. Например, вы можете назначать разные бонусы при покупках на разную сумму.
- Вторую для приведённых клиентов: как условие запуска тут будет выступать появление обратной связи. Использовать такой триггер или сценарий вы можете для предоставления клиентам бонусов и предложения участия в реферальной программе.

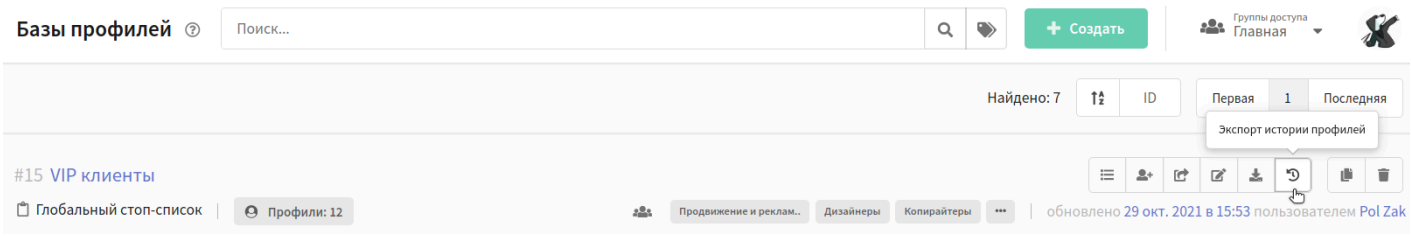
Экспорт истории профилей

Описание

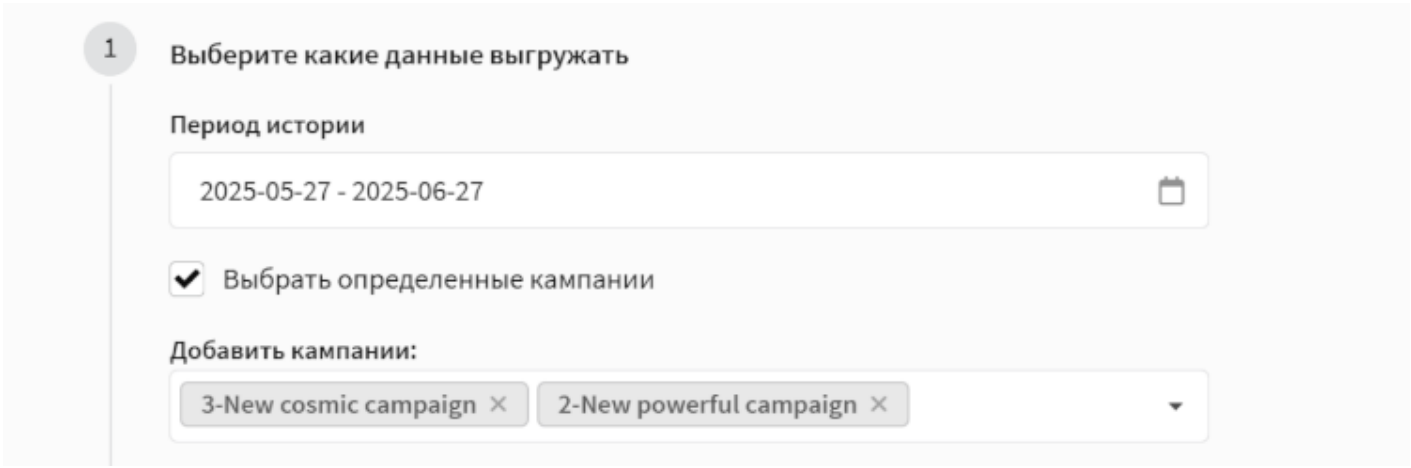
Экспорт истории профилей позволяет выгрузить из платформы данные о действиях профилей, которые находятся в определенной базе данных или в сегменте. Данные будут выгружены в csv-файл.

Как экспортировать историю профилей

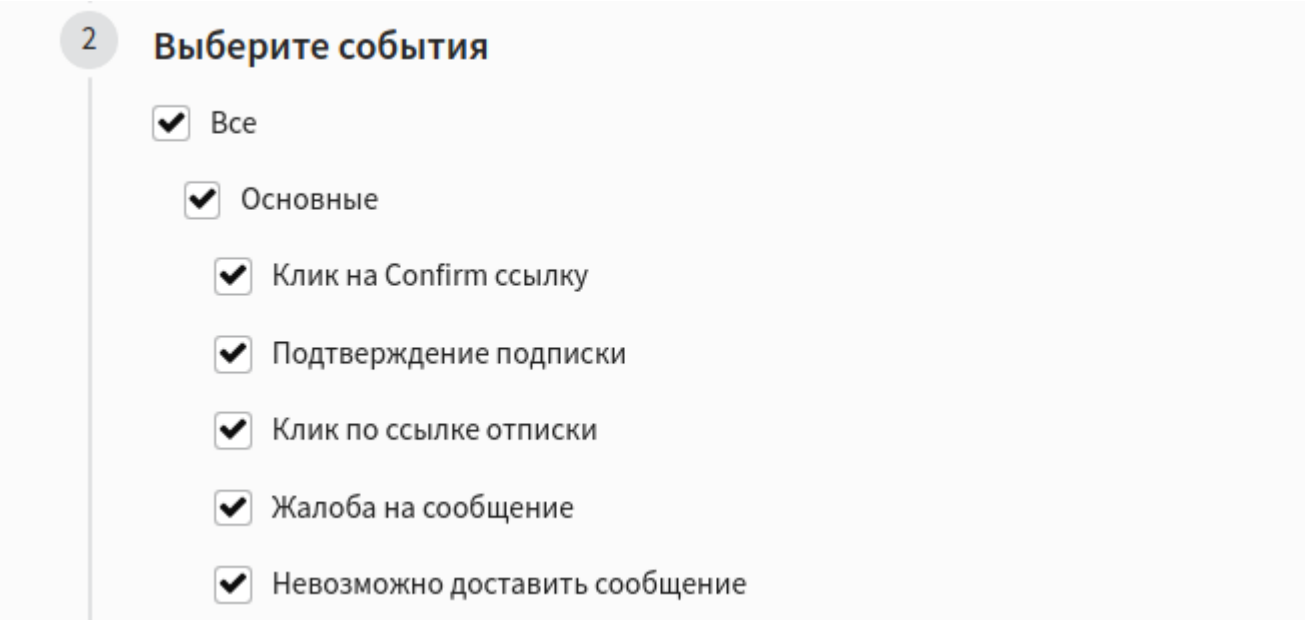
- Откройте список баз или сегментов, найдите нужную базу и кликните на иконку **“Экспорт истории профилей”**:



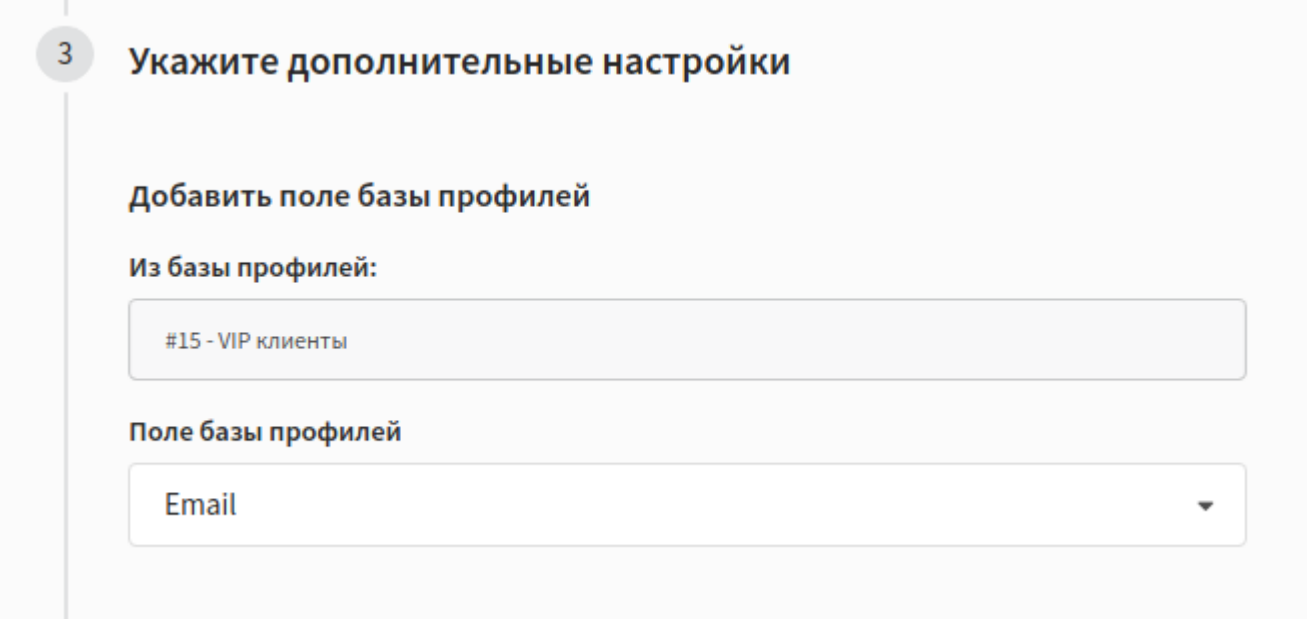
- Укажите **период**, данные за который необходимо выгрузить. При необходимости можно указать определенные рассылки:



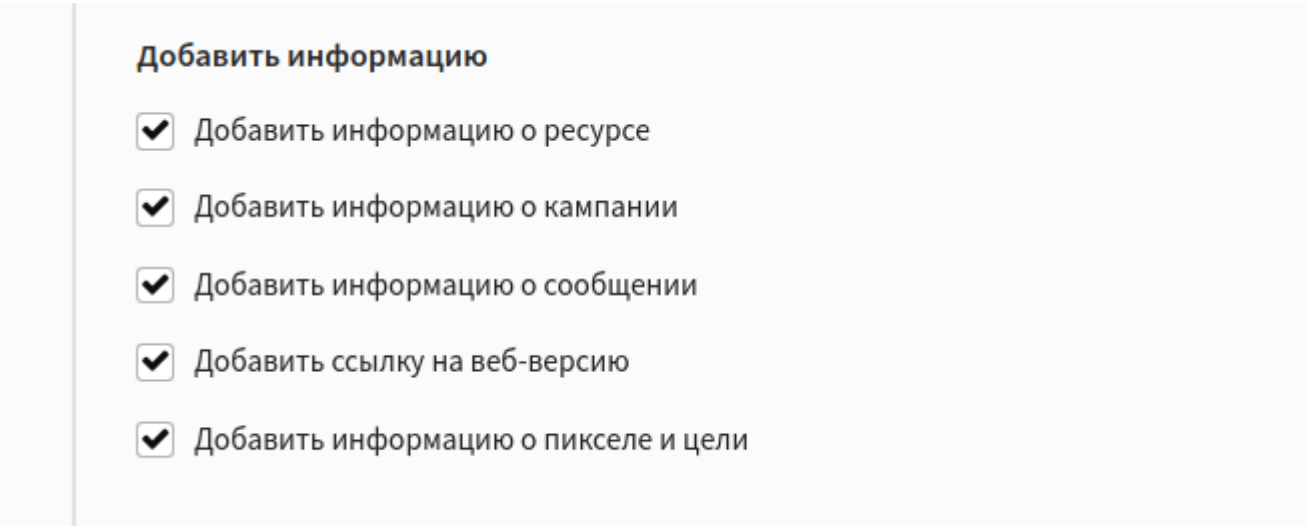
- Выберите **события** для экспорта:
    - Основные (подтверждение подписки, жалобы, отписка и др.)
    - Действия в различных каналах связи (доставки сообщений, открытия, клики и др.)
    - Пиксель и регистрация цели
    - Промокоды
    - Связи



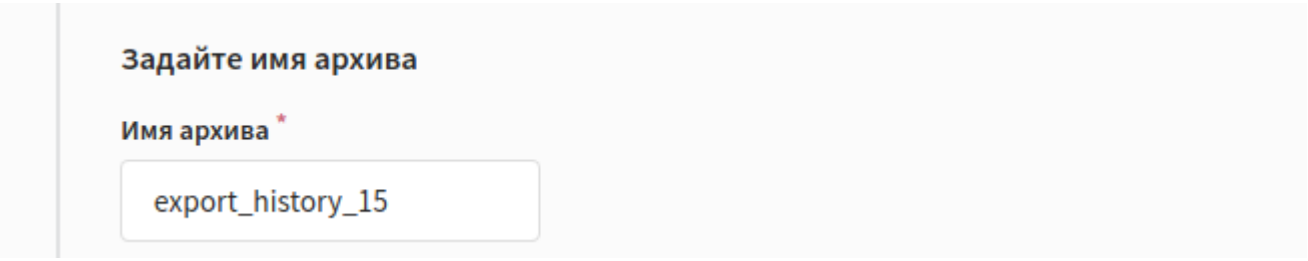
- Укажите **поле из базы профилей**, по которому хотите сгруппировать выгружаемые данные:



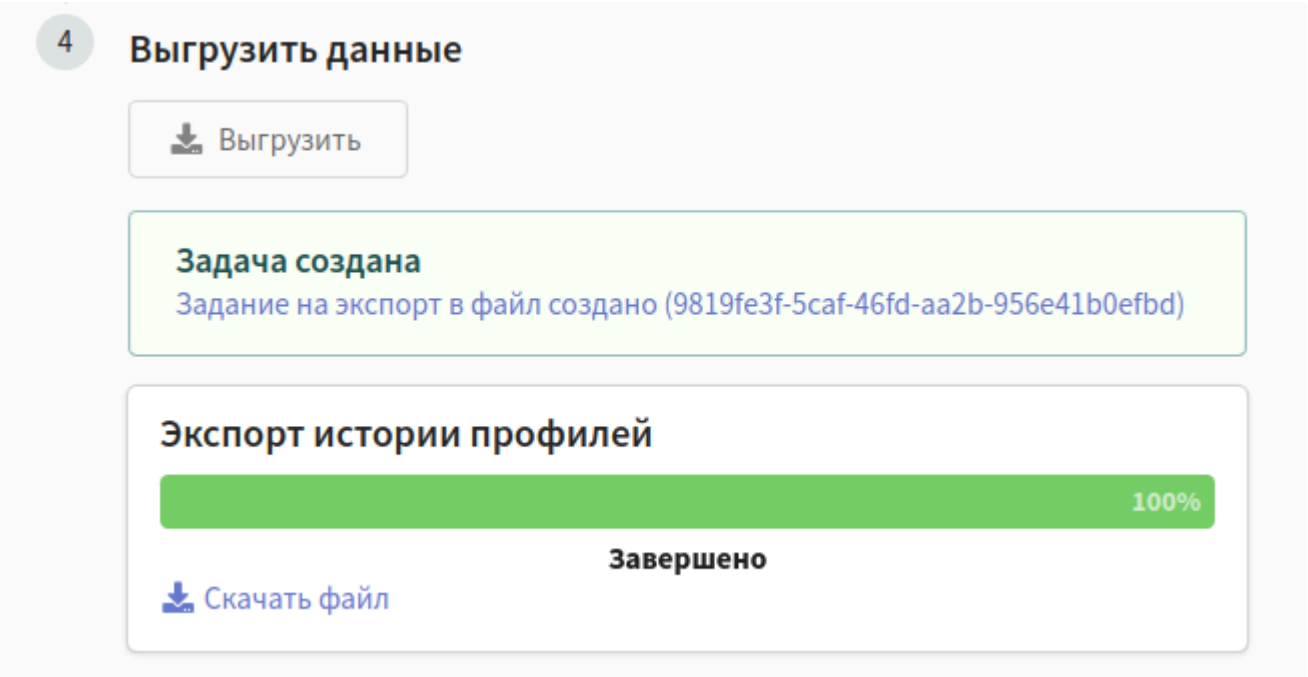
- Добавьте **дополнительную информацию** для экспорта:
    - ресурс — добавление в конечный файл ID и имя ресурса, на который подписался профиль
    - рассылка — ID и имя рассылки, в которой произошло событие;
    - сообщение — ID и имя использованного шаблона;
    - ссылка на веб-версию письма;
    - пиксели и цель — ID и имя пикселя, а также наименование и значение (value) цели, которую пиксель регистрирует.



- **Задайте** произвольное **имя** архива:



- Экспортируйте данные, нажав **“Выгрузить”**. Автоматически будет создана задача на экспорт. Ее прогресс отобразится тут же. Когда задача будет завершена, кликните **“Скачать файл”**. Архив в формате `.zip` будет скачан на ваше устройство.



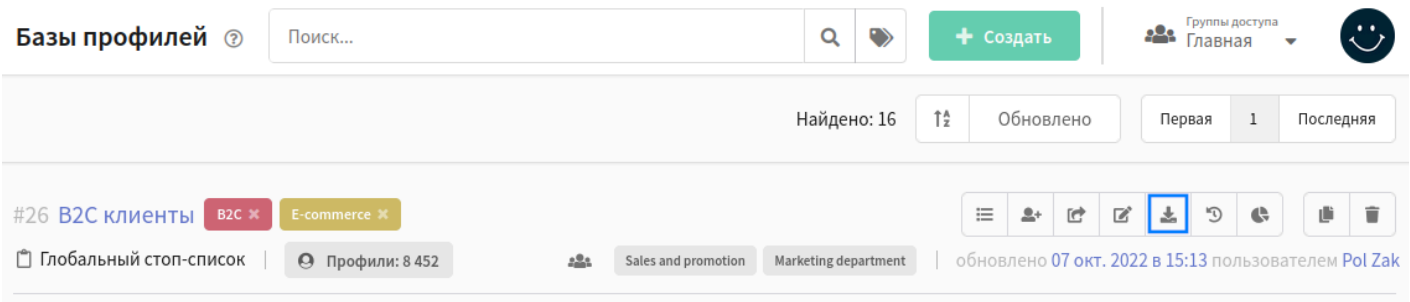
Экспорт профилей

Описание

Экспорт профилей позволяет выгрузить csv-файл с информацией обо всех профилях в базе данных или в сегменте. Вы можете самостоятельно выбрать данные, которые хотите получить.

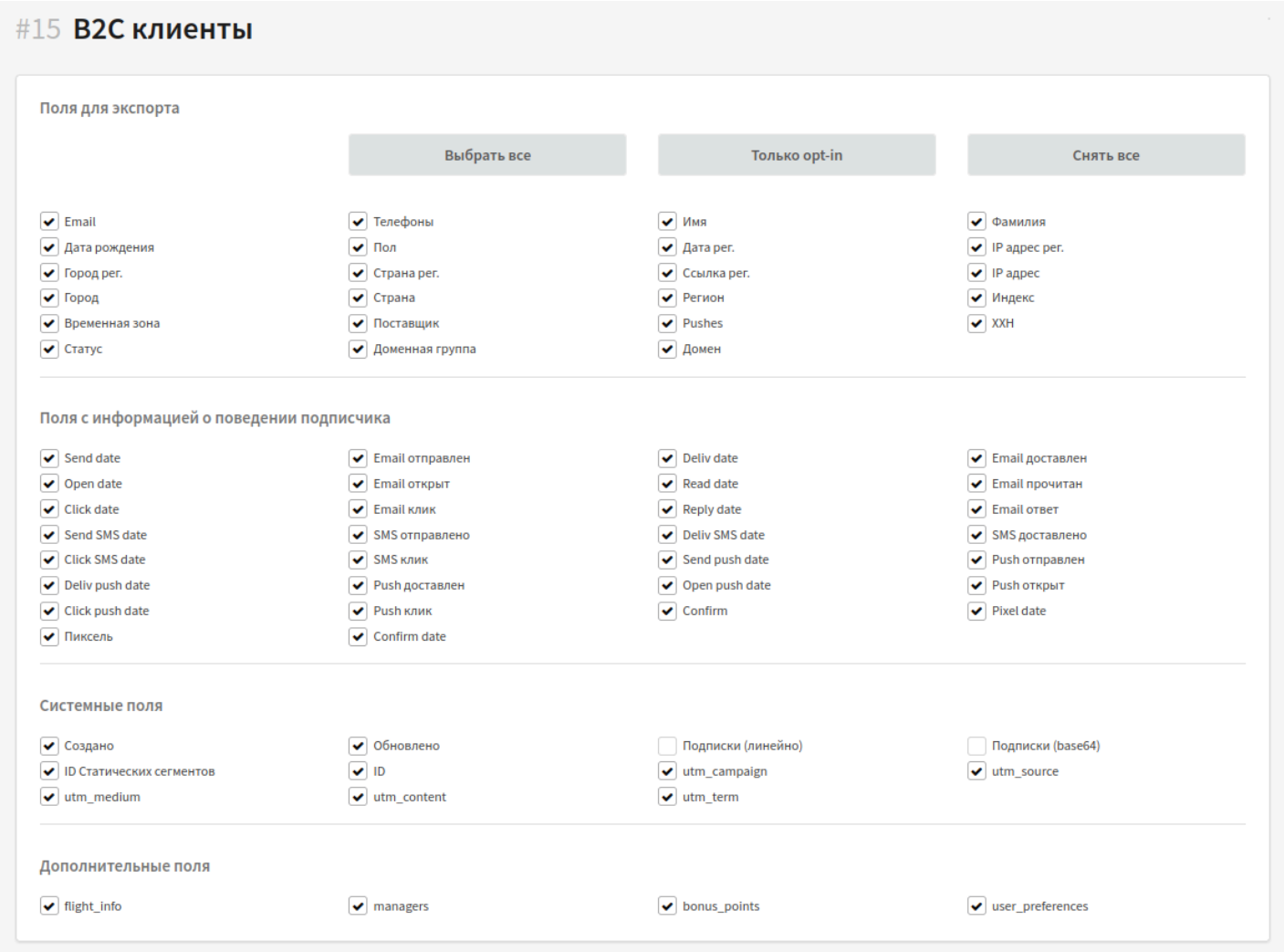
Чтобы начать экспорт, выберите базу или сегмент и нажмите кнопку

в панели управления:



Поля для экспорта

Отметьте поля профиля, которые необходимо выгрузить. Здесь вам могут помочь кнопки в верхней части страницы: **выбрать все поля** или снять выделение с полей, а также отметить только **opt-in данные**. К opt-in данным относится регистрационная информация: email, дата, ссылки и IP адрес при регистрации.



Для экспорта доступны следующие поля профиля:

- стандартные поля;
- поля с информацией о поведении подписчика;
- системные поля;
- дополнительные поля.

Стандартные поля

Поле для экспорта	Колонка в файле
Email	Email
Телефоны	Phones
Имя	FirstName
Фамилия	LastName
Дата рождения	Birthday
Пол	Gender
Дата рег.	RegDate
IP адрес рег.	LastCity
Город. рег	RegCity

Поле для экспорта	Колонка в файле
Страна рег.	RegCountry
Ссылка рег.	RegUrl
IP адрес	LastIP
Город	LastCity
Страна	LastCountry
Регион	LastRegion
Индекс	LastPostalCode
Временная зона	LastTZ
Поставщик	Vendor
XXH (xxhash идентификатор профиля)	XXH
Статус (Глобальный статус профиля)	Status
Доменная группа	DomainGroup
Домен	EmailDomain

Поля с информацией о поведении подписчика

Поле для экспорта	Колонка в файле	Описание
Send date Send SMS date Send push date	SendDate SendSMSDate SendPushDate	Дата и время первого отправленного сообщения (Email, SMS, Push)
Open date Open push date	OpenDate OpenPushDate	Дата и время последнего открытия сообщения (Email, Push)
Click date Click SMS date Click push date	ClickDate ClickSMSDate ClickPushDate	Дата и время последнего клика в сообщении (Email, SMS, Push)
Deliv date Deliv SMS date Deliv push date	DelivDate DelivSMSDate DelivPushDate	Дата и время последней доставки сообщения (Email, SMS, Push)
Read date	ReadDate	Дата и время последнего прочтения сообщения (Email)
Reply date	ReplyDate	Дата и время последнего ответа на сообщение (Email)
Confirm date	ConfirmDate	Дата и время перехода по ссылке подтверждения
Pixel date	PixelDate	Дата и время последней активации пикселя

Поле для экспорта	Колонка в файле	Описание
Email отправлен SMS отправлено Push отправлен	SendExists SendSMSExists SendPushExists	Было ли отправлено сообщение за весь период взаимодействия с профилем (Email, SMS, Push)
Email открыт Push открыт	OpenExists OpenPushExists	Было ли открыто сообщение за весь период взаимодействия с профилем (Email, Push)
Email клик SMS клик Push клик	ClickExists ClickSMSExists ClickPushExists	Был ли переход по ссылке в сообщении за весь период взаимодействия с профилем (Email, SMS, Push)
Email доставлен SMS доставлено Push доставлен	DelivExists DelivSMSExists DelivPushExists	Было ли доставлено сообщение за весь период взаимодействия с профилем (Email, SMS, Push)
Email прочитан	ReadExists	Было ли прочитано сообщение за весь период взаимодействия с профилем (Email)
Email ответ	ReplyExists	Был ли получен ответ на сообщение за весь период взаимодействия с профилем (Email)
Confirm	ConfirmExists	Был ли переход по ссылке подтверждения в сообщении за весь период взаимодействия с профилем
Pixel	PixelOpenExists	Был ли активирован любой пиксель за весь период взаимодействия с профилем

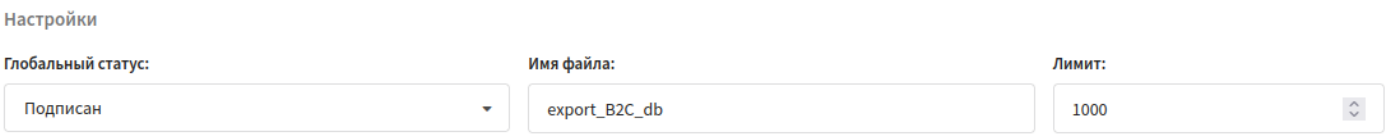
Системные поля

Поле для экспорта	Колонка в файле	Описание
Создано	CreateTime	Дата и время создания профиля
Обновлено	UpdateTime	Дата и время последнего обновления профиля

Поле для экспорта	Колонка в файле	Описание
Подписки (линейно)		Выгружаются данные по подписке: • subscription_resource_id (идентификатор ресурса) • subscription_channel (канал подписки) • subscription_status (статус) • subscription_priority (приоритет) • subscription_hash_id (хеш) • subscription_is_delete (была ли подписка удалена) • subscription_email (электронный адрес) • subscription_phone (номер телефона) • subscription_subscription_id (идентификатор push подписки) • subscription_provider (push провайдер) • subscription_date (дата подписки) • subscription_ip (IP-адрес) • subscription_url (URL-адрес подписки)
Подписки (base64)		Данные о подписках выгружаются в виде строки, закодированной в base64
utm_campaign utm_source utm_medium utm_content utm_term	UtmCampaign UtmSource UtmMedium UtmContent UtmTerm	UTM метки
ID статических сегментов	StaticSegments	Идентификатор статического сегмента, в котором состоит профиль
ID	Id	Идентификатор профиля (profile_id)

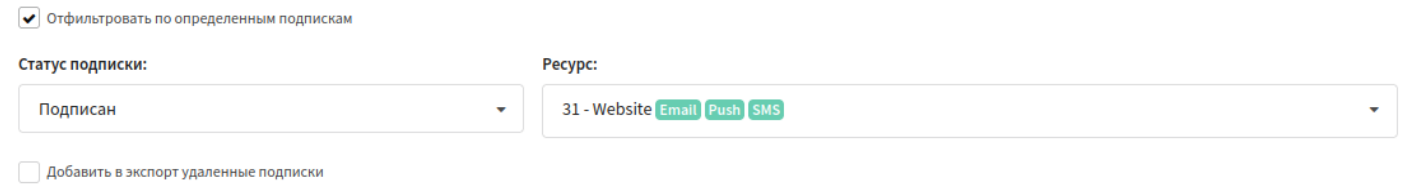
Настройки экспорта

В этом блоке задаются дополнительные настройки экспорта. Вы можете экспортировать данные о профилях только с определенным **глобальным статусом**, а также ограничить количество выгружаемых профилей, указав значение в поле **“Лимит”**:



Чтобы выгрузить информацию о профилях с подпиской на один из ресурсов, активируйте опцию “**Отфильтровать по определенным подпискам**”. Выберите ресурс и при необходимости статус подписки.

Если необходимо, чтобы в файле также отображались данные профилей, у которых раннее была подписка на выбранный ресурс, но теперь удалена — поставьте галочку в поле “**Добавить в экспорт удаленные подписки**”.



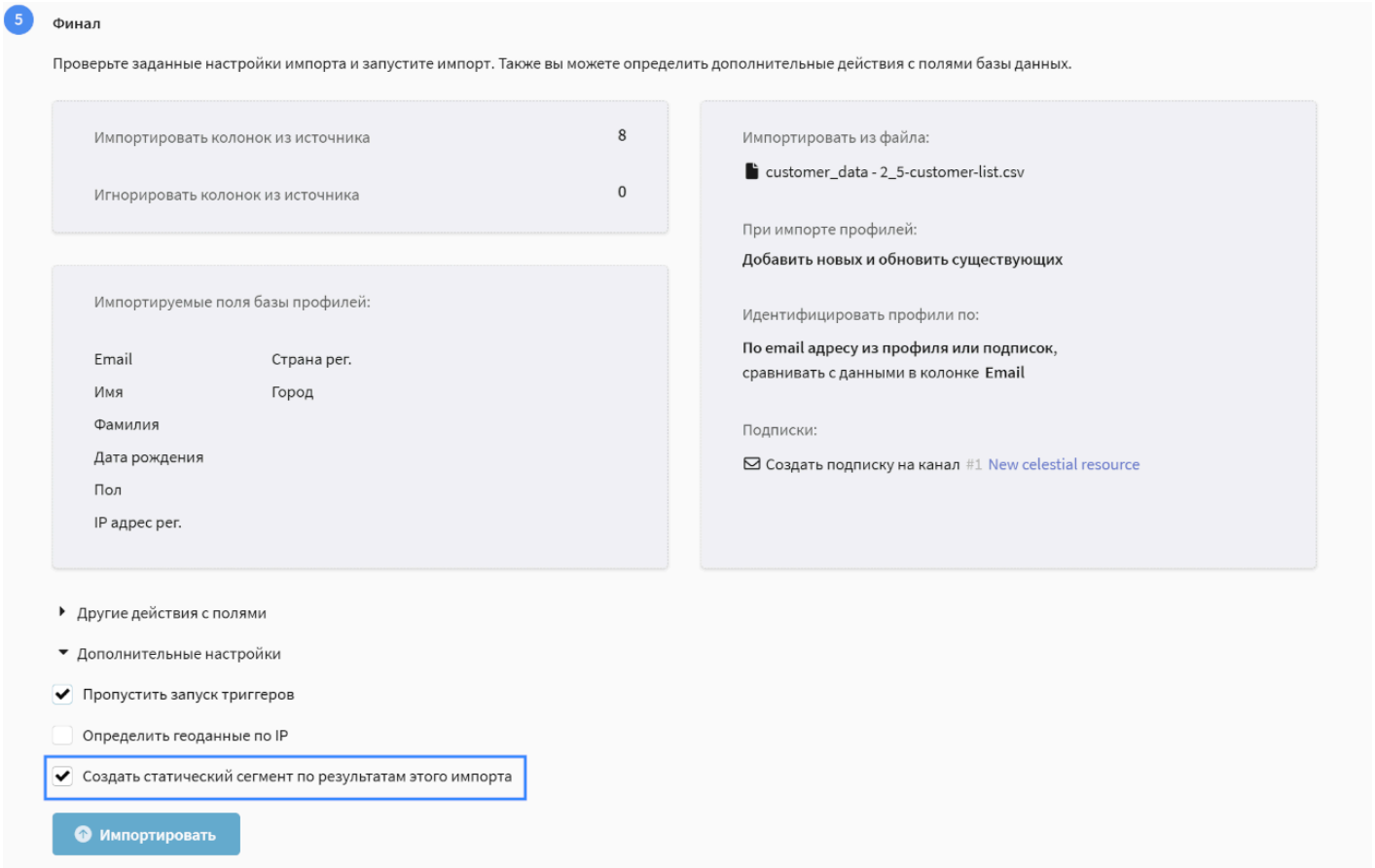
Когда вы выберите поля для выгрузки и настроите детали экспорта, нажмите на кнопку **“Экспортировать”**. Платформа запустит процесс. Перейдите в раздел **“Задания”**, чтобы скачать сгенерированный файл с данными профилей.

Автоматическое создание статического сегмента при импорте

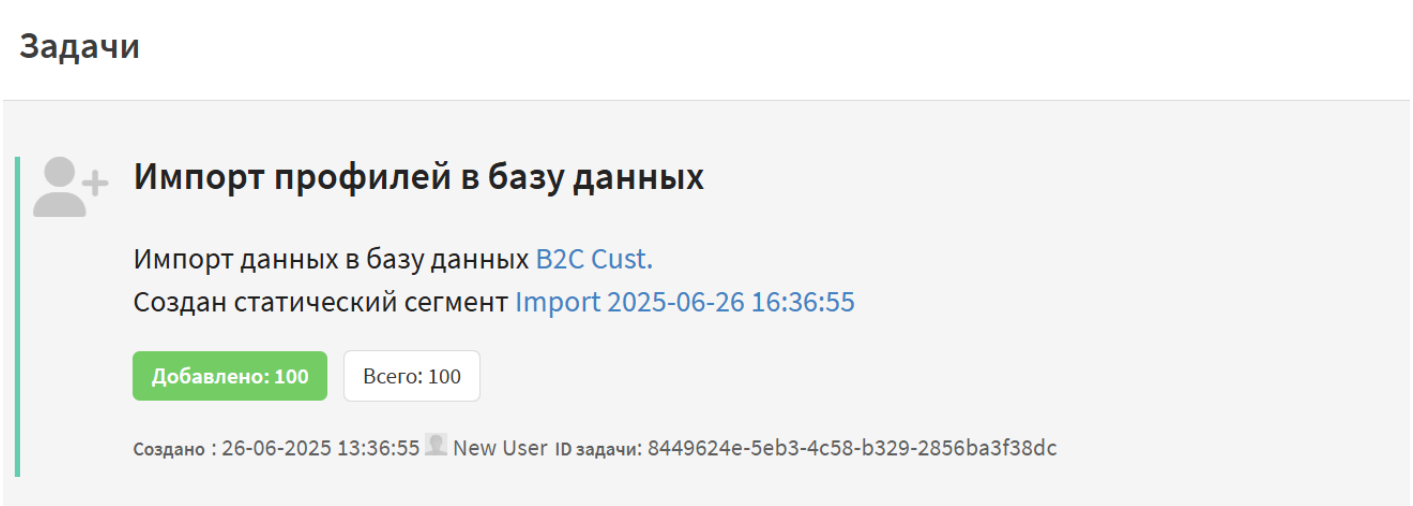
Если мы импортируем профили в базу с помощью файла, в деталях импорта можно включить опцию **“Создать статический сегмент по результатам импорта”**.

TIP

Это удобно, если регулярно в одну и ту же БД импортируются новые профили, для которых надо осуществить однократную рассылку. Создание статического сегмента позволит объединить данные.



Можно проверить процесс импорта и создания сегмента в **Заданиях**:



Таким образом, как только импорт завершится, будет создан новый статический сегмент, куда войдут только что импортированные профили. Название статического сегмента будет сгенерировано автоматически (например: `Import 2025-06-26 16:36:55`).

Как открыть CSV-файл

Что такое CSV

CSV (Comma-Separated Values) — это текстовый формат, предназначенный для представления табличных данных. Данные в таком файле разделены запятыми или другими разделителями, такими как точка с запятой или табуляция.

```
Email,Phones,FirstName,Gender,Birthday
irina@example.com,+79000001234,Ирина,Female,1995-07-21T01:00:00+04:00
artem@example.com,+79009009090,Артем,Male,2000-04-24T12:00:39+04:00
tanya@example.com,+79000000000,Татьяна,Female,1999-02-04T12:01:17+03:00
sergey@example.com,+79000000001,Сергей,Male,1994-06-13T12:02:38+04:00
```

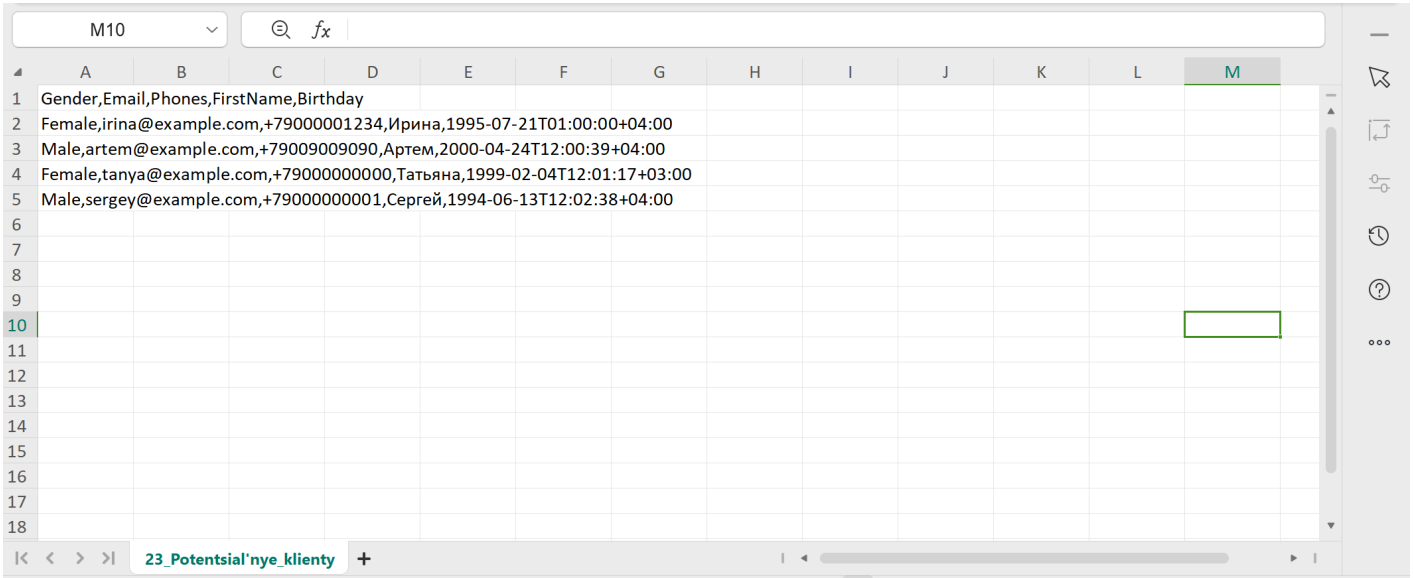
INFO

CSV-файлы могут быть сохранены в разных кодировках, включая UTF-8. Если файл содержит символы кириллицы или других нелатинских алфавитов, рекомендуется использовать UTF-8 для корректного отображения данных.

Использование CSV файлов в платформе

В платформе CSV-файлы используются в следующих случаях:

- импорт профилей в базу из файла
- экспорт профилей и их истории из базы
- выгрузка отчетов
- выгрузка результатов выполнения задач и т.д.



Для работы с CSV-файлами часто используют редакторы электронных таблиц, например, Excel или WPS Office. Однако при открытии файла иногда данные могут отображаться некорректно. Основные проблемы:

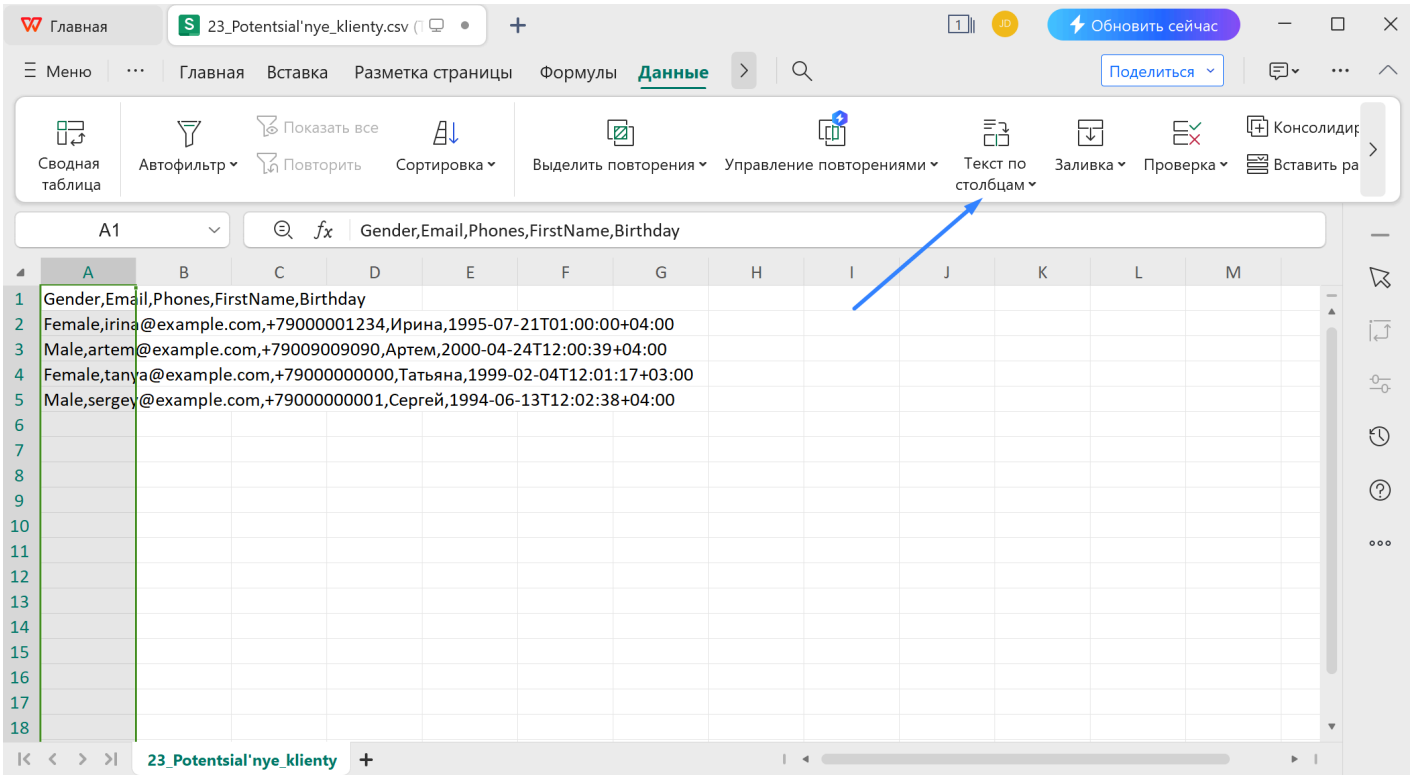
- Excel неправильно определяет используемый разделитель и помещает все данные в одну колонку
- Если файл сохранён в UTF-8, Excel может не распознать кириллицу и отобразить её в виде нечитаемых символов.

Ниже вы найдете инструкции, как правильно открыть CSV-файл. В зависимости от используемой программы интерфейс может незначительно отличаться.

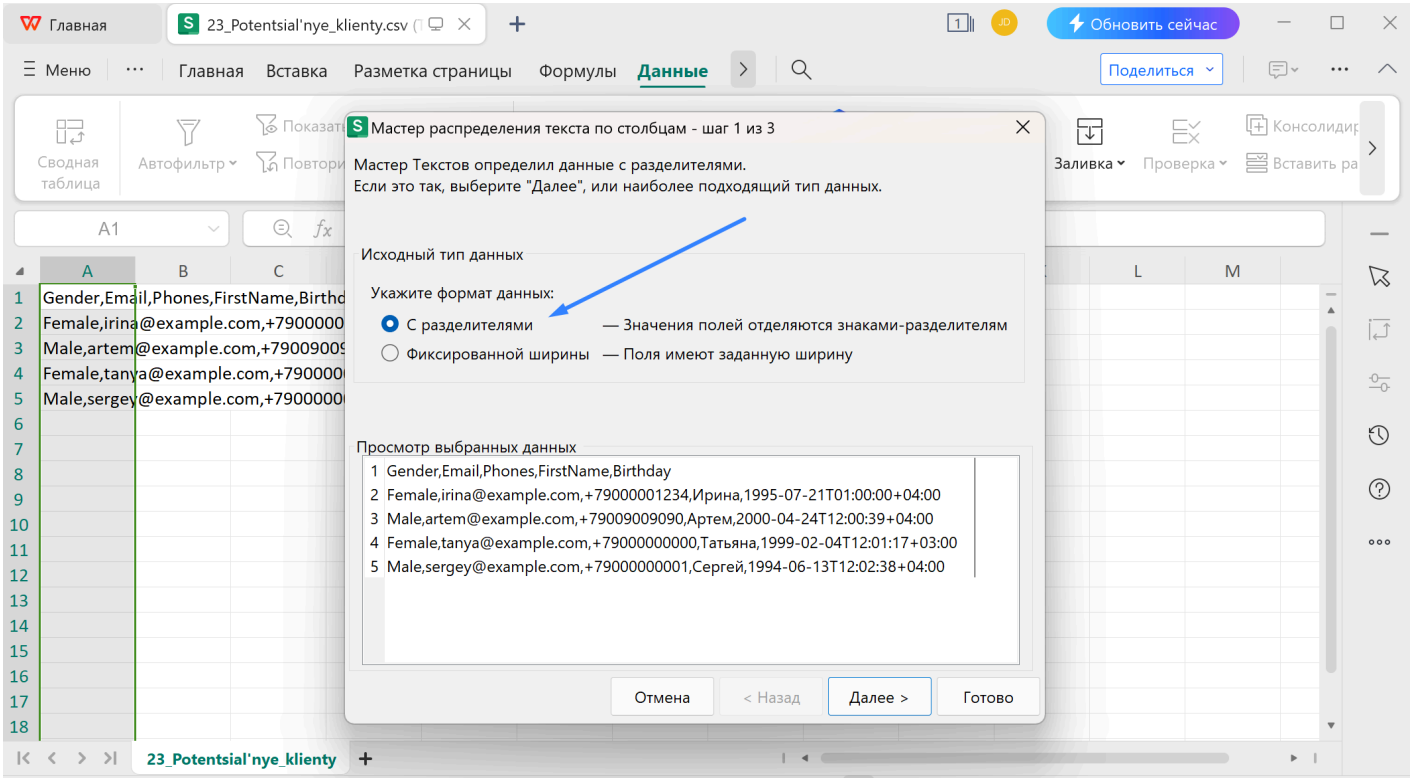
Инструкция по открытию CSV-файла

Ручное разделение данных CSV

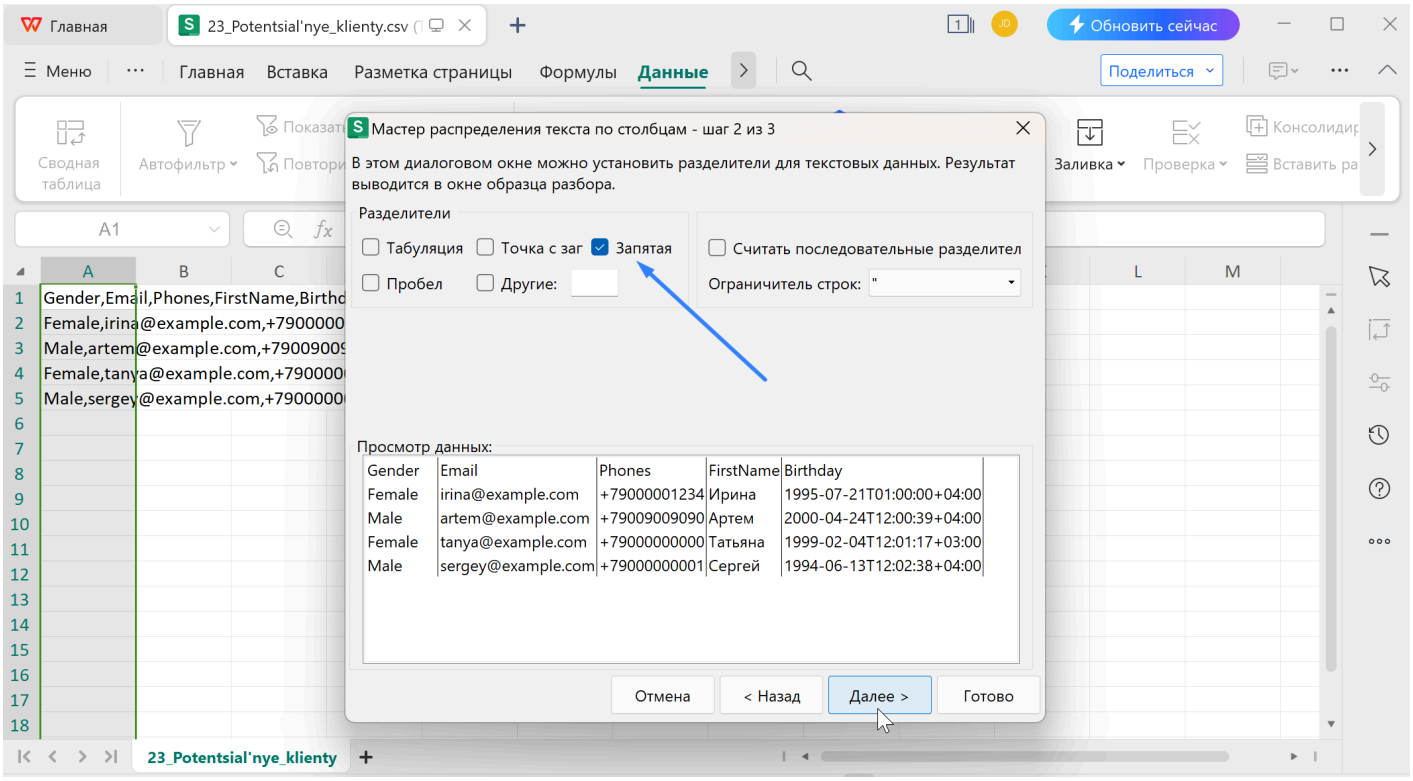
Откройте файл, выделите весь первый столбец и на верхней панели перейдите в раздел **“Данные”**. Нажмите **“Текст по столбцам”**:



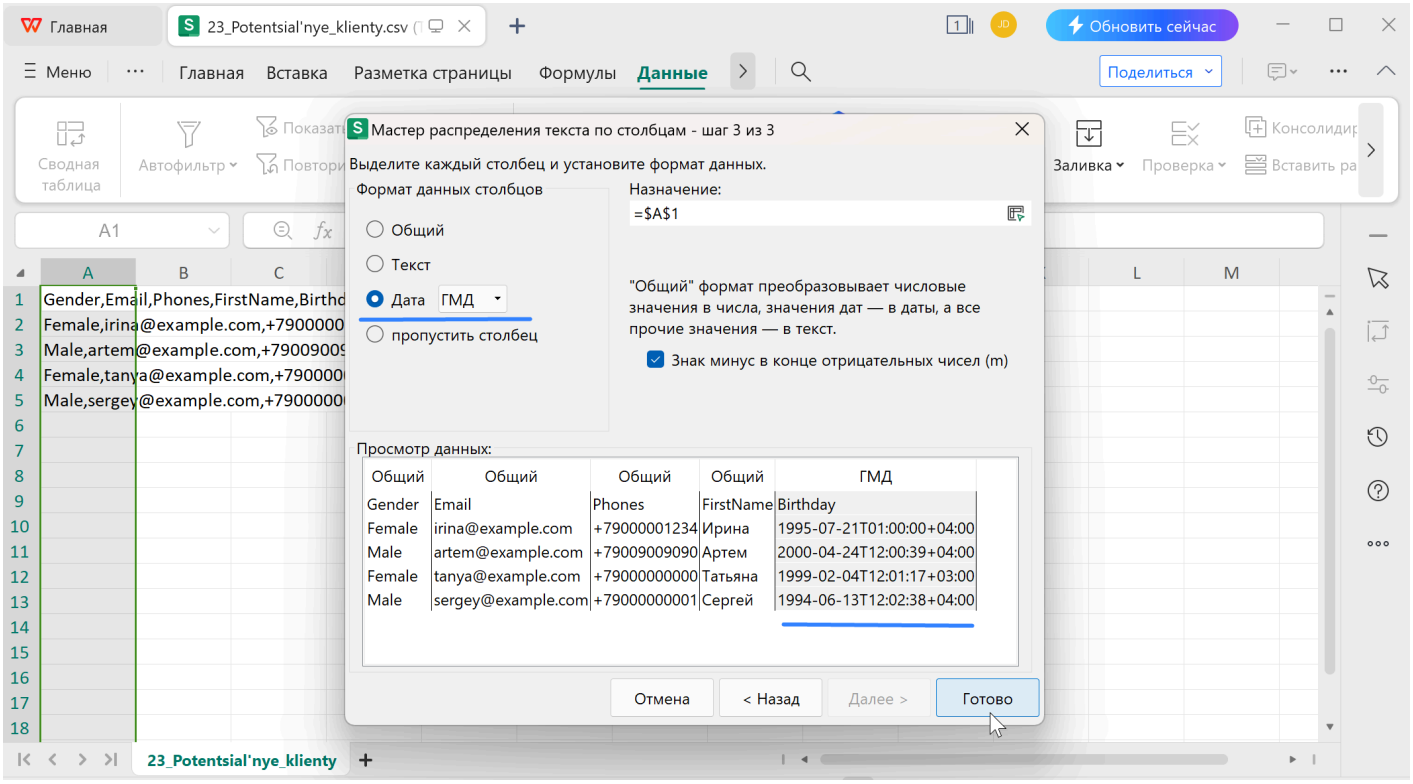
Программа откроет мастер распределния текста по столбцам, который поможет вам правильно интерпретировать данные. На первом шаге укажите формат исходных данных — **с разделителем**. Нажмите **“Далее”**:



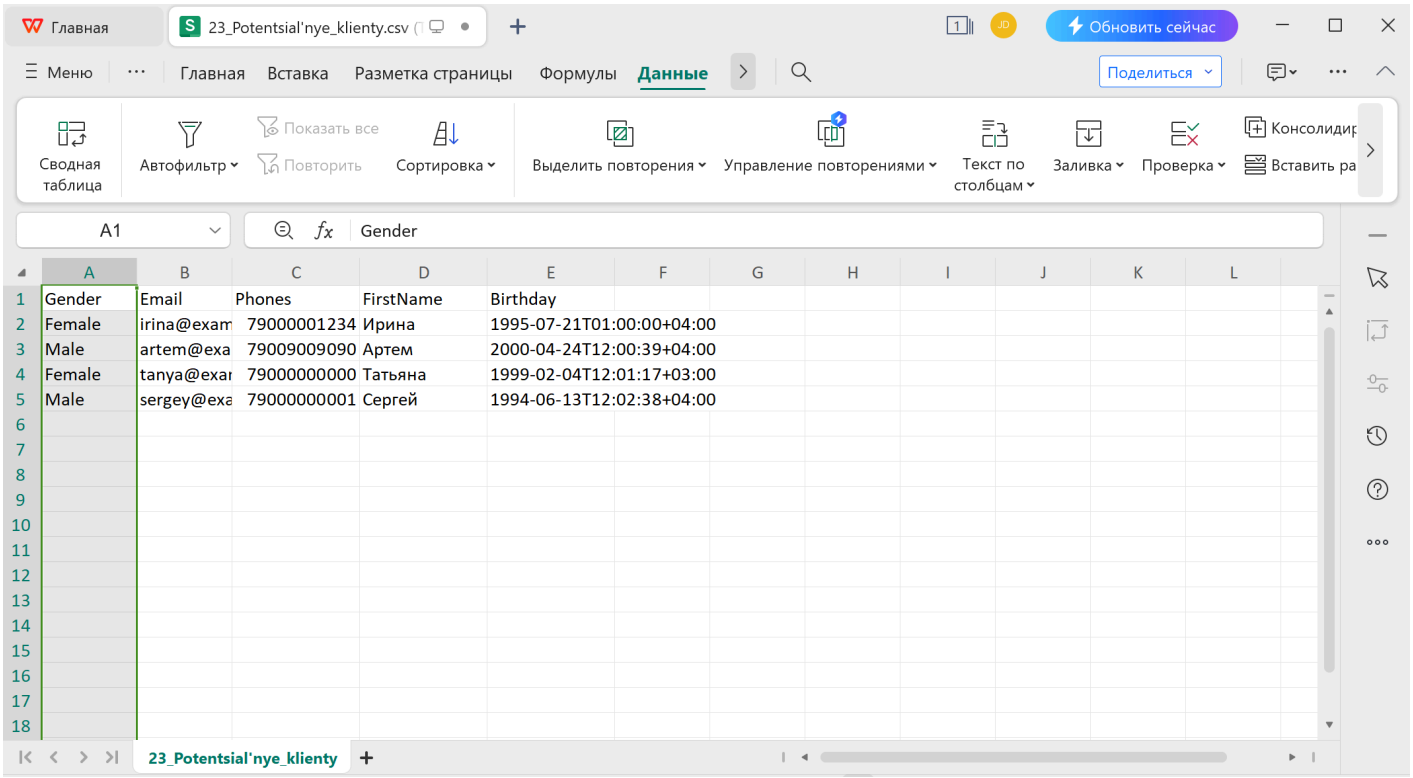
На втором шаге выберите **запятую** в качестве **разделителя столбцов**. Нажмите **“Далее”**:



На третьем шаге при необходимости вы можете определить формат данных в столбцах. В примере ниже для колонки Birthday устанавливается формат “Дата (год-месяц-день)”. Затем нажмите **“Готово”**:



Если все шаги были выполнены правильно, то данные в файле распределятся по соответствующим колонкам:



Кодировка UTF-8

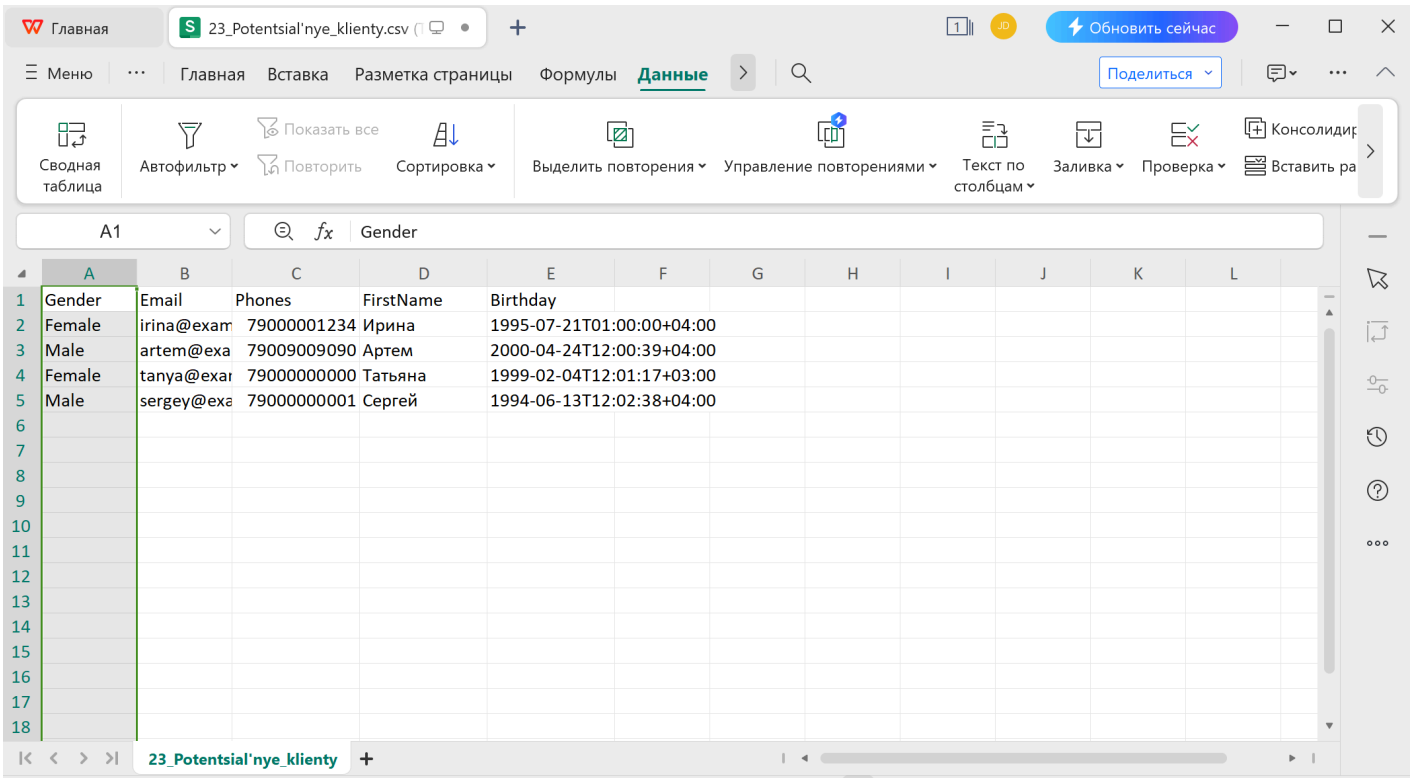
Если файл закодирован в UTF-8, то кириллица в нем можем отображаться некорректно. Ниже приведена инструкция, как это исправить.

Перейдите на вкладку **“Данные”** —> **“Импорт данных”** (или **“Изменить импорт данных”**, если файл уже открыт):

Выберите нужный CSV-файл и укажите кодировку **UTF-8:**

Выберите в качестве разделителя **запятую:**

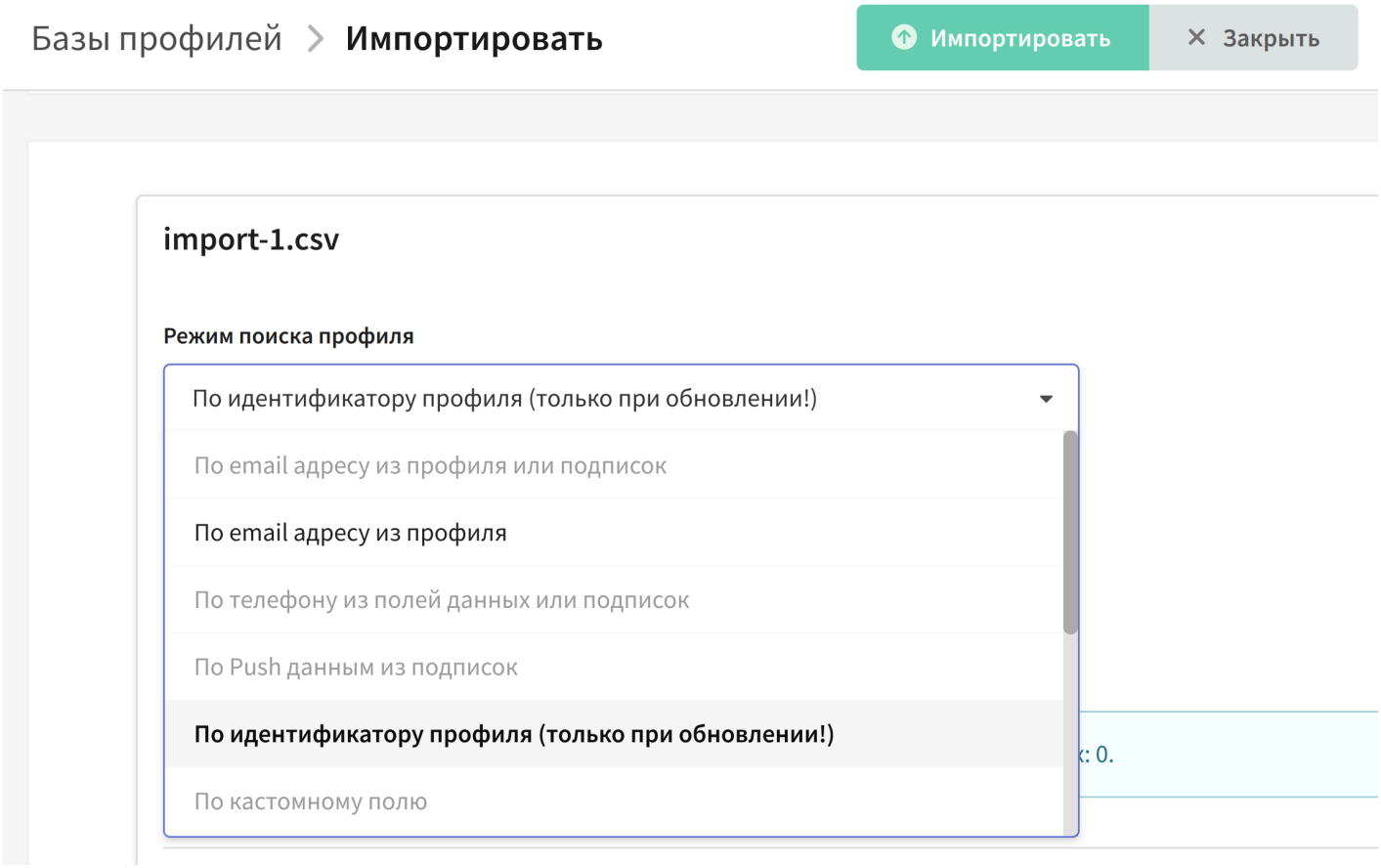
Нажмите **“Готово”,** и данные будут правильно распределены по столбцам:



Матчинг

Матчинг — это процесс поиска и идентификации существующих профилей клиентов в базе данных по определенным данным. Он используется для обновления информации о клиентах и предотвращения создания дублирующихся записей в базе.

При импорте профилей в базу данных (из файла, через API-запрос или из внешней SQL таблицы), вы указываете режим поиска профиля:



В процессе импорта платформа сравнивает эти данные с теми, что уже есть в базе. Если она находит профиль с такими же данными, то **обновляет его**. Если не находит — **создает новый профиль**.

Обратите внимание, данные, которые вы собираетесь использовать для поиска, должны быть в источнике. Например, если вы импортируете данные из файла и выбираете режим поиска по email, убедитесь, что у вас есть колонка с email-адресами в вашем файле.

Режимы поиска

Доступны следующие режимы поиска:

- **По email адресу из профиля или подписок** — по полю профиля **Email** или по email адресам из подписок.
- **По email адресу из профиля** — только по полю профиля **Email**.
- **По email адресу из подписок**
- **По телефону из полей данных или подписок** — по полю профиля **Телефоны** или по телефонам из SMS подписок.
- **По телефону из SMS подписок**

- **По Push данным из подписок** — по идентификатору Push подписки (токену устройства).
- **По идентификатору профиля** — по системному идентификатору **profile_id**.
- **По кастомному полю** — по пользовательскому полю базы данных. Например, ID клиента в CRM компании.
- **По кастомному полю из подписок**
- **Мультиматчинг** — по Email или телефону из профиля и подписок (email_phone), или по email или телефону из подписок (email_phone_sub).

Особенности работы матчинга

Есть несколько важных нюансов, связанных с матчингом:

- Поле, по которому осуществляется поиск, должно быть уникальным для каждого профиля. Значение этого поля не должно совпадать у разных профилей в одной базе данных.
- Не рекомендуется использовать разные типы матчинга в одной базе данных, чтобы избежать дублирования профилей.
- Если при импорте или обновлении профилей через API не передавать режим поиска в поле matching, платформа будет искать профиль по email-адресу по умолчанию.

INFO

Если вы собираетесь пользоваться матчингом по пользовательскому полю, вам нужно обратиться к администратору платформы для создания **индекса** этого поля. Благодаря индексу поиск профиля в базе будет занимать значительно меньше времени.

Типы полей в базе данных

Стандартные поля базы

В таблице ниже представлен список стандартных полей базы, их соответствия в API, типы данных, а также примеры значений, которые можно использовать.

Поле в базе	Имя поля в API	Тип поля в UI	Тип поля в API	Пример значения
Email	email	Строка/Email	string	"example@example.com"
Телефоны	phones	Список телефонов	array	["+79001234567"]
Имя	_fname	Строка	string	"John"
Фамилия	_lname	Строка	string	"Doe"
Дата рождения	_bdate	Дата	string	"2001-12-31" "2001-12-31T00:00+03:00" Время можно передавать только через API.
Пол	_sex	Список	string	"male" "female"
Дата рег.	_regdate	Дата	string	"2023-12-31" "2023-12-31T23:59:59+03:00"
IP адрес рег.	_regip	IP	string	"127.0.0.1"
Город рег.	_regcity	Строка	string	"Москва"
Страна рег.	_regcountry	Строка	string	"Russian Federation" "RU"
Ссылка рег.	_regurl	Строка	string	"https://example.com"
IP адрес	_ip	IP	string	"127.0.0.1"
Город	_city	Строка	string	"Москва"
Страна	_country	Строка	string	"Russian Federation" "RU"
Регион	_region	Строка	string	"Europe/Moscow"
Индекс	_postal_code	Строка	string	"190000"
Временная зона	_tz	Строка	string	"Europe/Moscow"
Поставщик	_vendor	Строка	string	"form #30"

Дополнительные поля базы

От типа поля зависит, какие операторы будут доступны в сегментах и в динамических переменных шаблона, поэтому рекомендуем внимательно выбирать тип дополнительного поля в базе.

Тип поля в UI	Тип поля в API	Пример значения
Целое число	int	100
Число с точкой	float	100.5
Строка	string	"string value"
Дата	string	"2024-05-27T12:00:00Z"
Boolean	boolean	true false
Список	string	"Вариант 1" Также можно передавать значение в виде числа, например, 1. Число соответствует порядковому номеру элемента в списке.
IP	string	"127.0.0.1"
Теги	array	["Тег 1", "Тег 2", "Тег 3"]
Объект	object	`{ "key_1": { "sub_key": "sub_value" } }`
Массив объектов	array of objects	`[ { "key_1": "value_1" }, { "key_2": "value_2" } ]`

INFO

При API-импорте или обновлении полей с типом “Теги” их значения будут **перезаписаны**, а не добавлены к уже существующим. Чтобы добавить новые значения, не удаляя старые, следует воспользоваться функциональным обновлением.

Допустимые форматы и ограничения

Платформа ожидает определённые форматы значений для каждого типа поля. Убедитесь, что импортируемые данные соответствуют указанным требованиям, иначе система не сможет обработать их корректно.

- **Email** — формат должен соответствовать стандартам RFC. Для валидации используется регулярное выражение:

```
^[-a-z0-9!#$%&'*+/=?^_{|}~]+(\.[-a-z0-9!#$%&'*+/=?^_{|}~]+)*@((?:localhost|([a-z0-9]([-a-z0-9]{0,61}[a-z0-9])?\.)*([a-z]{1,24})))$
```

- **Телефоны** — номер должен соответствовать международному формату (без пробелов, дефисов и других разделителей, кроме плюса в начале). Подробнее со стандартом можно ознакомиться здесь.
- **Даты** — записываются в формате RFC 3339 (ISO 8601) с указанием часового пояса.
- **Страна** — принимается в виде двухбуквенного кода ISO 3166-1 alpha-2 или полного названия (в любом регистре).
- **Пол** — допустимые значения описаны в этой статье.
- **Поставщик** — длина строки не больше 26 символов.
- **IP** — IPv4 формат.
- **Строка** — не рекомендуется хранить более 100 КБ текста в одном поле.
- **Объекты и массивы объектов** — можно записывать только одноранговые объекты или одноранговые объекты, вложенные в другие объекты. Если вы используете объекты с более сложной структурой, подключите SQL базу и создайте запросы для выборки профилей. В именах полей объектов допустимо использовать только латинские буквы (в любом регистре), цифры от 0 до 9, символ нижнего подчеркивания `_` и дефис `-`. Можно хранить данные типа `int`, `float`, `string`, `boolean`.
- Ограничение на объем данных **одного профиля в базе составляет 16 МБ**, включая все метаданные. Однако не рекомендуется хранить суммарно больше 2 МБ в дополнительных полях профиля.

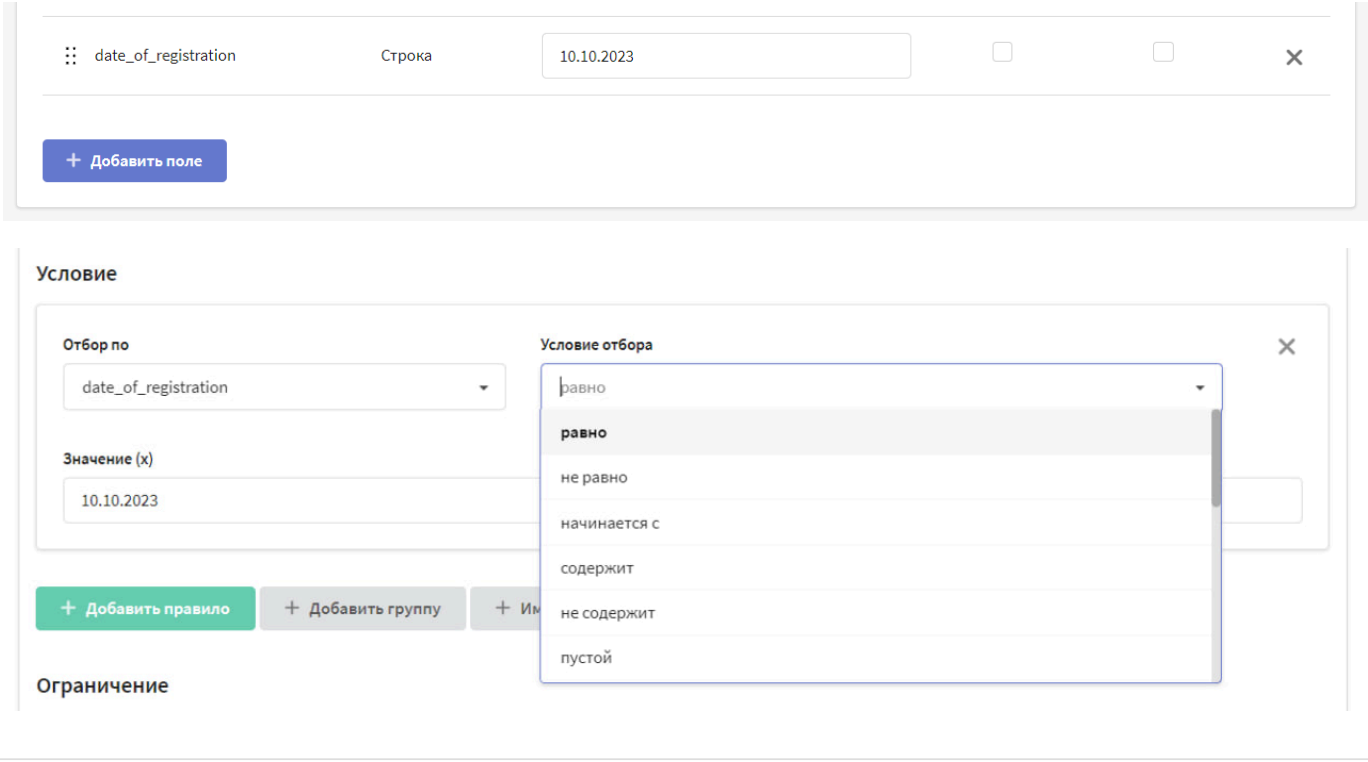
Соответствие полей базы и элементов формы

Если вы настраиваете импорт данных профиля из полей формы в поля базы, убедитесь, что вы используете корректный тип поля для каждого элемента формы.

Элемент формы	Поле в базе
Короткий текст	Доп. поле с типом "Строка"
Email	Стандартное поле "Email"
Номер телефона	Стандартное поле "Телефоны"
Длинный текст	Доп. поле с типом "Строка"
Число	Доп. поле с типом "Целое число"
Да/Нет	Доп. поле с типом "Boolean"
Один вариант	Доп. поле с типом "Строка"
Множественный выбор	Доп. поле с типом "Теги"
Выпадающий список	Доп. поле с типом "Теги"
Дата	Стандартное или доп. поле с типом "Дата"
Диапазон	Доп. поле с типом "Целое число"

NOTE

Хоть вы и можете записать все данные пользователя в строковые поля, это ограничит возможности для дальнейшей сегментации. Например, если дата будет записана в строковое поле, вы не сможете использовать операторы сравнения “дата меньше” или “дата больше” в сегментах или шаблонах.



Глобальные контрольные группы

Что такое глобальная контрольная группа

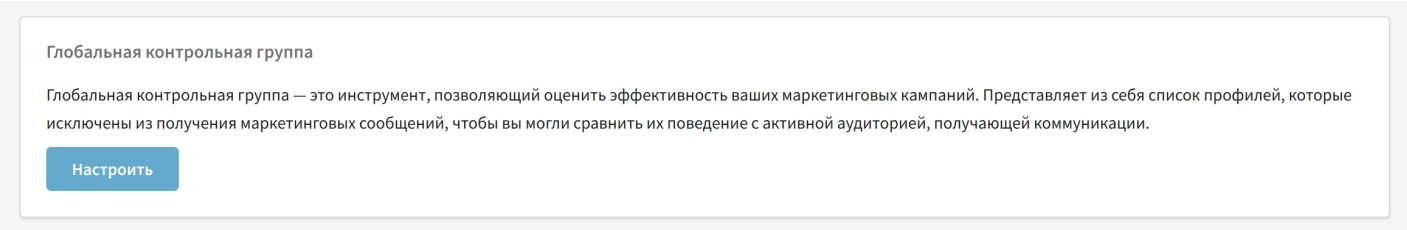
Глобальные контрольные группы (Global control groups) — часть клиентской базы, которая временно не получает маркетинговые коммуникации с целью анализа эффективности маркетинговых стратегий. Спустя некоторое время пользователь может сравнить ключевые метрики основной части клиентов и контрольной группы. В том случае, если показатели групп отличаются незначительно, маркетолог может сделать вывод о неэффективности проводимых им рассылок.

Наличие контрольных групп позволяет:

- Оценить эффективность маркетинговых стратегий;
- Оптимизировать затраты на маркетинг;
- Увеличить конверсию маркетинговых акций.

Настройка глобальных контрольных групп

Настроить формирование глобальной контрольной группы для базы профилей можно только после ее создания. Для этого перейдите к редактированию созданной базы и найдите соответствующий раздел внизу страницы:



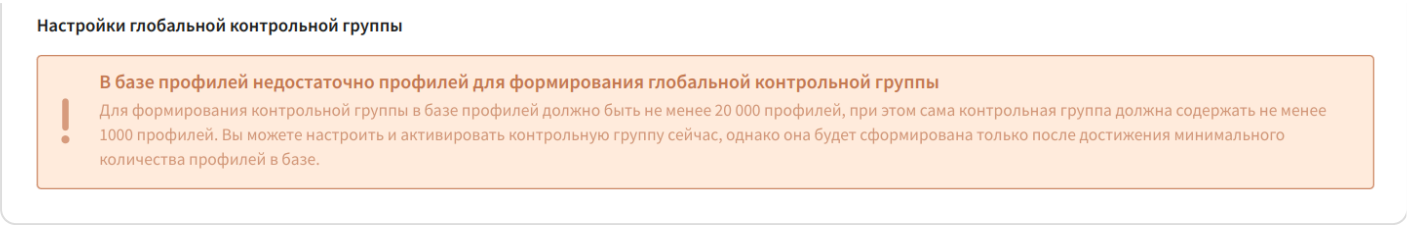
Сперва необходимо выбрать **источник аудитории** для ГКГ. Таким источником может являться либо **база профилей**, либо **сегмент**. В том случае, если вы выбрали сегмент в качестве источника для ГКГ, вам необходимо настроить запрос сегментации. Подробнее о сегментации и условиях отбора можно прочесть в этом разделе.

TIP

Вне зависимости от источника аудитории для глобальной контрольной группы существуют следующие требования:

- ГКГ должна включать от 1 000 до 10 миллионов профилей;
- ГКГ должна составлять не менее 5% профилей и не более 25% от всей базы профилей;
- Для формирования контрольной группы в базе профилей должно быть не менее 20 000 профилей.

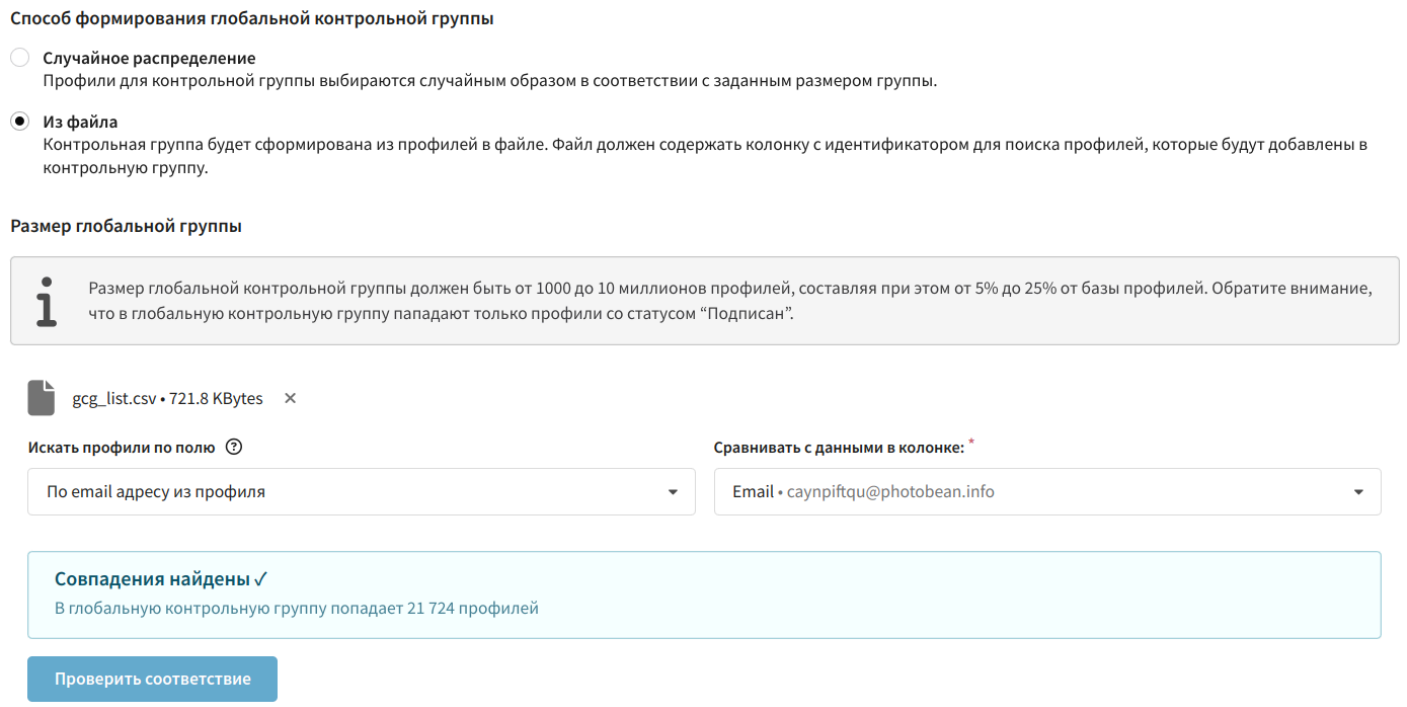
Вы можете настроить и активировать контрольную группу в любой момент, однако она будет сформирована только **после выполнения всех вышеуказанных критериев**. В случае каких-либо проблем, связанных с количеством профилей в ГКГ, платформа оповестит вас об этом:



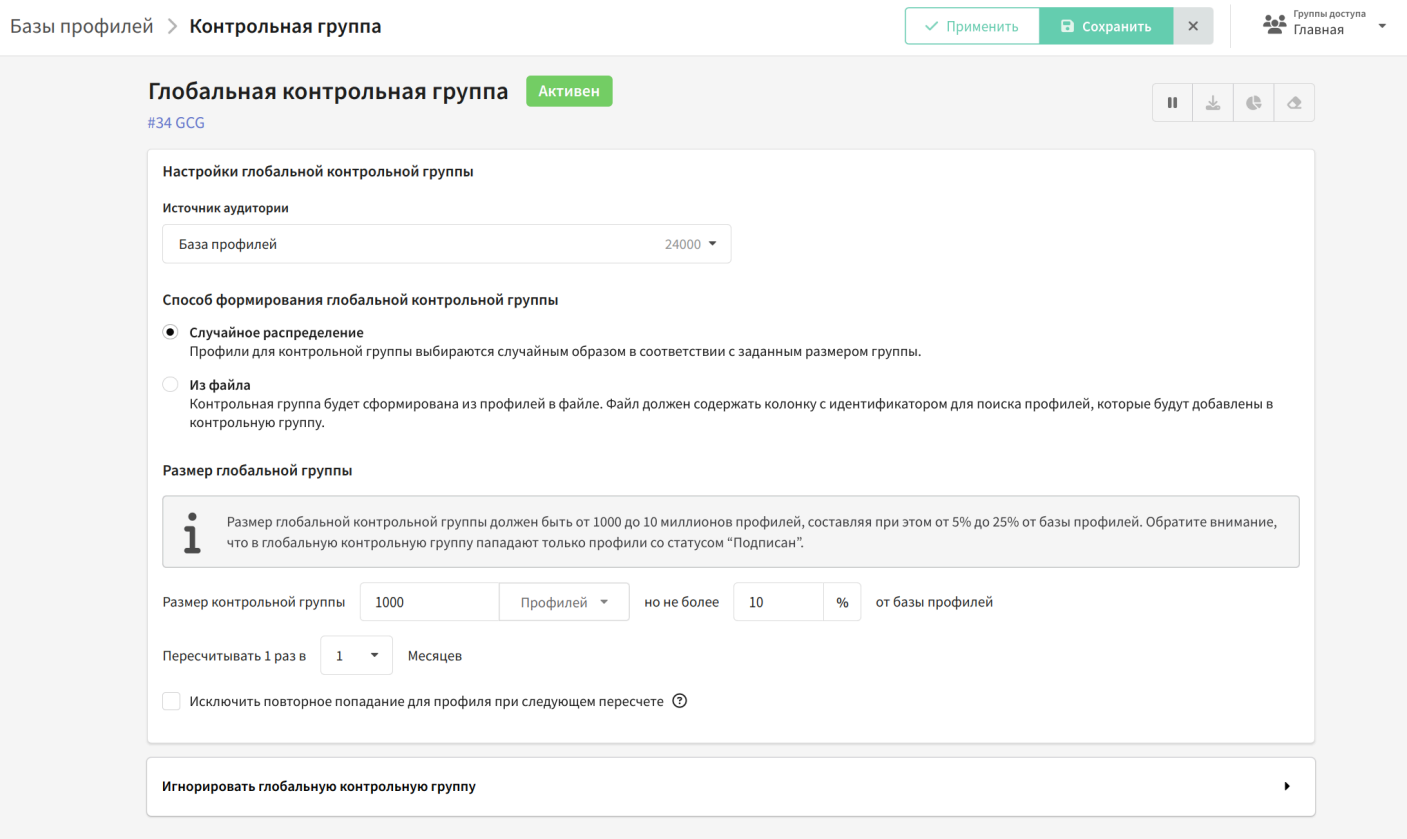
У глобальной контрольной группы на основе базы профилей существует два способа формирования:

- **Случайное распределение** — профили для контрольной группы выбираются случайным образом в соответствии с заданным размером группы.
- **Из файла** — контрольная группа будет сформирована из профилей в файле. Файл должен содержать колонку с идентификатором для поиска профилей для контрольной группы. Имейте в виду, что ГКГ будет сформирована из **всего файла**.

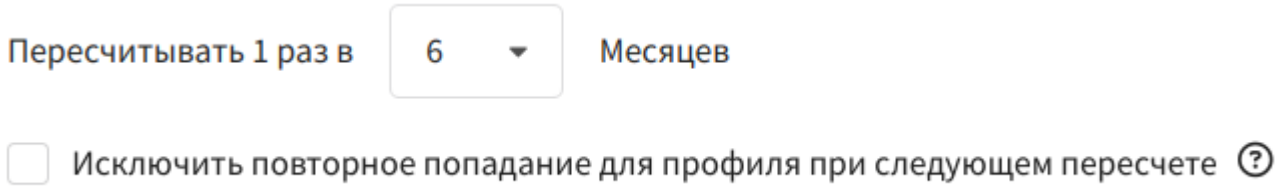
Если вы выбрали вариант формирования из файла, вам будет предложено загрузить файл со списком клиентов в формате .xls, .xlsx, .csv или .txt, а затем указать поле для поиска профилей в БД и колонку для сравнения из файла. После нажатия на кнопку **“Проверить соответствие”** платформа покажет сообщение с указанием количества профилей, которые войдут в контрольную группу:



Для случайного распределения можно указать конкретное количество профилей в группе, а также максимальное процентное отношение ГКГ ко всей базе профилей. Последний параметр нужен для того, чтобы ограничить размер ГКГ в случае, если количество профилей в базе значительно изменится:



Затем необходимо указать периодичность **пересчета** для глобальной контрольной группы, минимальное значение — раз в 12 месяцев, максимальное — раз в 1 месяц. Пересчет происходит в соответствии с указанным расписанием в 00:00 часов по часовому поясу аккаунта:

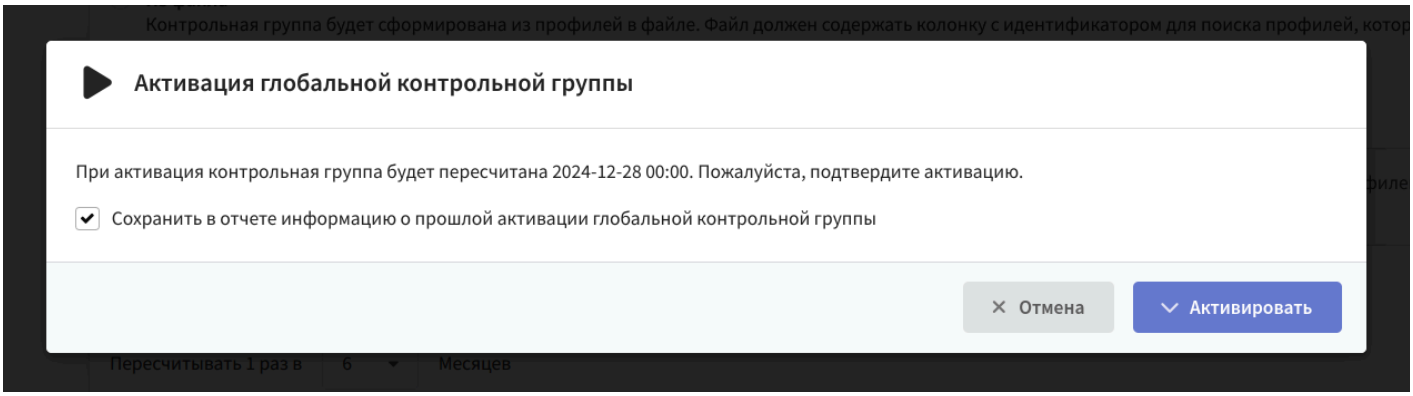


Параметр **“Исключить повторное попадание для профиля при следующем пересчете”** позволяет запретить профилям, участвующим в текущей контрольной группе, попадать туда при ближайшем пересчете. Однако эти профили смогут попасть в ГКГ при третьем и последующих пересчетах.

TIP

Если одни и те же профили постоянно попадают в контрольную группу, это может исказить результаты эксперимента и привести к ошибочным выводам. Параметр “Исключить повторное попадание для профиля при следующем пересчете” позволяет менять состав контрольной группы, что делает результаты более достоверными.

Для того чтобы глобальная контрольная группа начала работать, необходимо **активировать ее** и **сохранить**. Перед активацией платформа сообщит вам следующую дату пересчета контрольной группы и предложит сохранить информацию о прошлой активации:

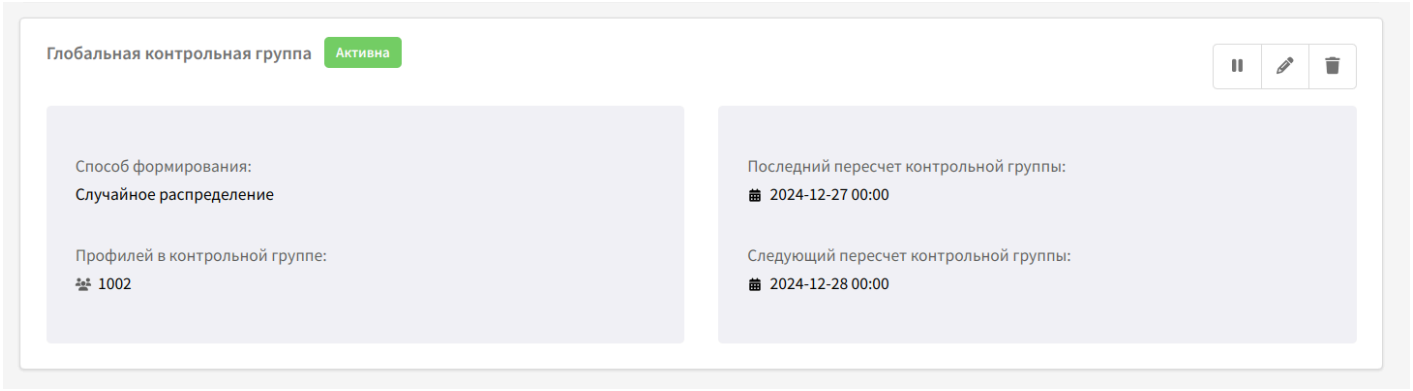


После сохранения и запуска будет доступно **меню быстрых действий** в верхней части экрана:

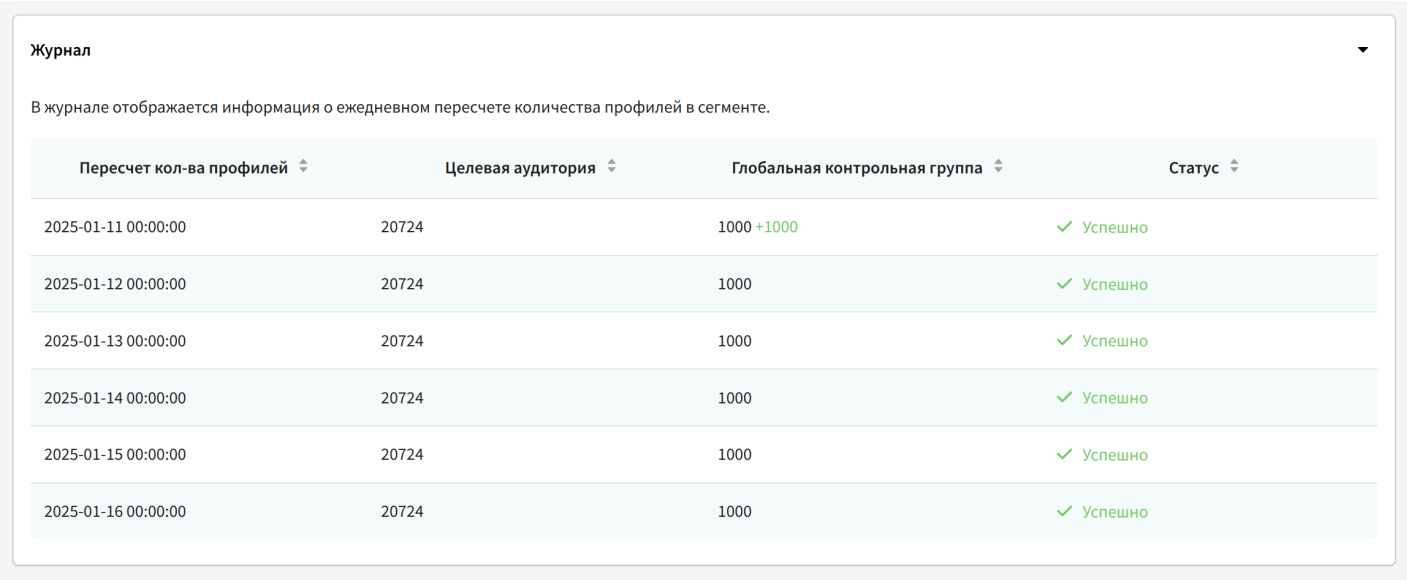
В меню вам доступны следующие кнопки:

- **Деактивировать** — останавливает работу ГКГ
- **Выгрузить отчет** — позволяет выгрузить список профилей, находящихся в ГКГ
- **Аналитика** — открывает соответствующую страницу аналитики
- **Сбросить настройки** — сбрасывает настройки ГКГ и очищает все заполненные поля.

Имейте в виду, что первые данные по участникам ГКГ и аналитика по ней будут доступны **после первого расчета**. После успешной настройки и запуска ГКГ на экране редактирования базы профилей отобразится **информация о контрольной группе, количестве профилей в ней**, а также **даты последнего и следующего пересчета**:

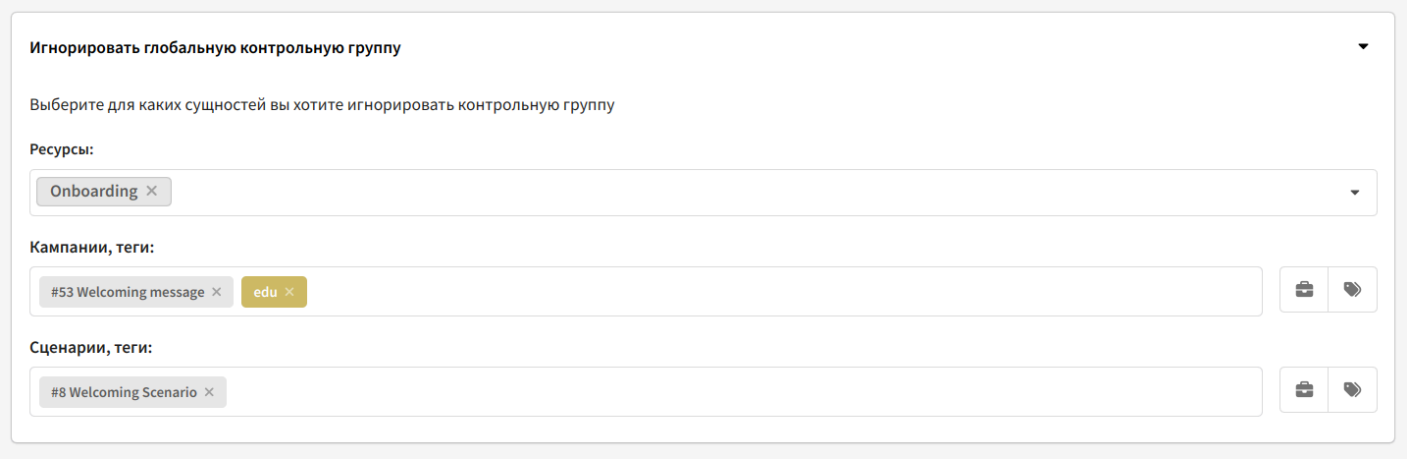


Для того чтобы посмотреть **журнал**, в котором содержится подробная информация о пересчетах ГКГ, необходимо перейти к редактированию контрольной группы и открыть соответствующую вкладку в нижней части экрана:



Исключения для глобальной контрольной группы

Вы можете настроить исключения из работы глобальной контрольной группы в разделе **“Игнорировать глобальную контрольную группу”**. В рамках выбранных ресурсов, рассылок и сценариев будут осуществлятся маркетинговые коммуникации даже для тех пользователей, которые находятся в глобальной контрольной группе:



Здесь можно указать:

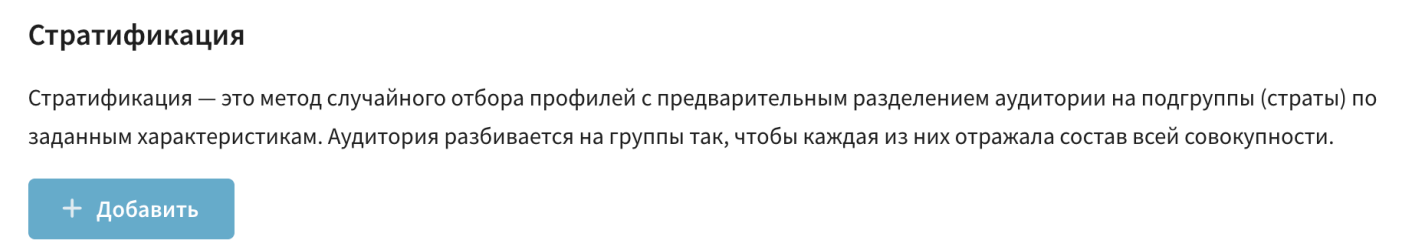
- конкретные рассылки и сценарии;
- сущности под определенными **тегами**;
- сущности с определенным **значением атрибута**.

Стратификация

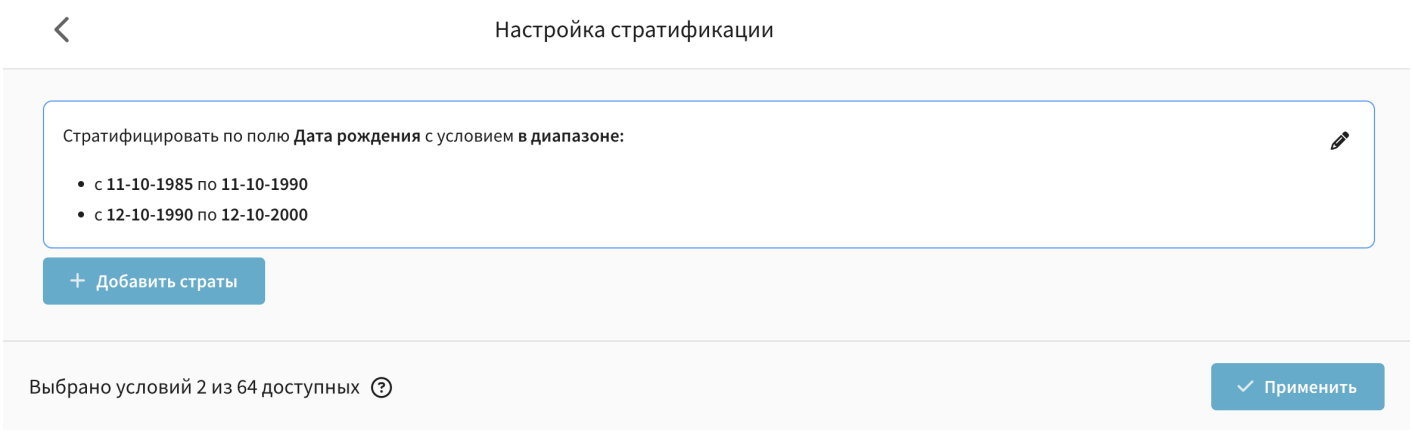
Стратификация — это метод случайного отбора профилей с предварительным разделением аудитории на подгруппы (страты) по заданным характеристикам. Аудитория разбивается на группы так, чтобы каждая из них отражала состав всей совокупности.

Из каждой группы случайным образом выбирается часть профилей, пропорциональная размеру этой группы. Это помогает сформировать контрольную группу с таким же распределением характеристик, как у тестовой, что делает результаты рассылки более точными. Максимальное количество страт зависит от числа выбранных характеристик и количества групп внутри них. Эти ограничения можно настроить в системе.

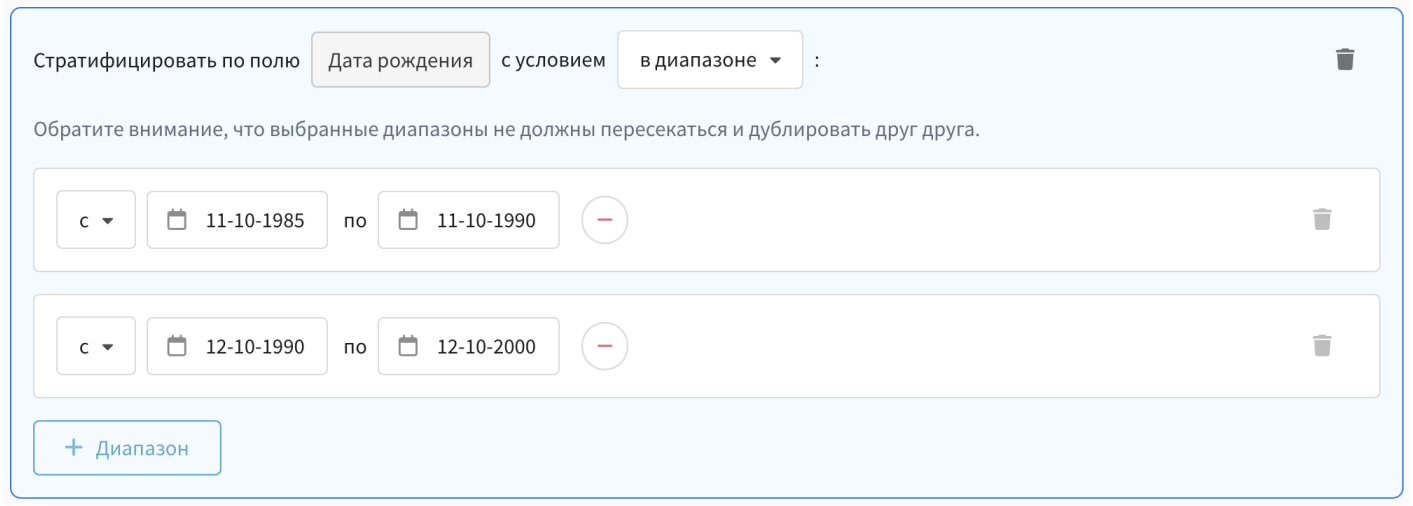
Добавьте стратификацию, нажав соответствующую кнопку:



Вы окажетесь в окне **настройки стратификации**, здесь вы сможете создавать и редактировать страты:

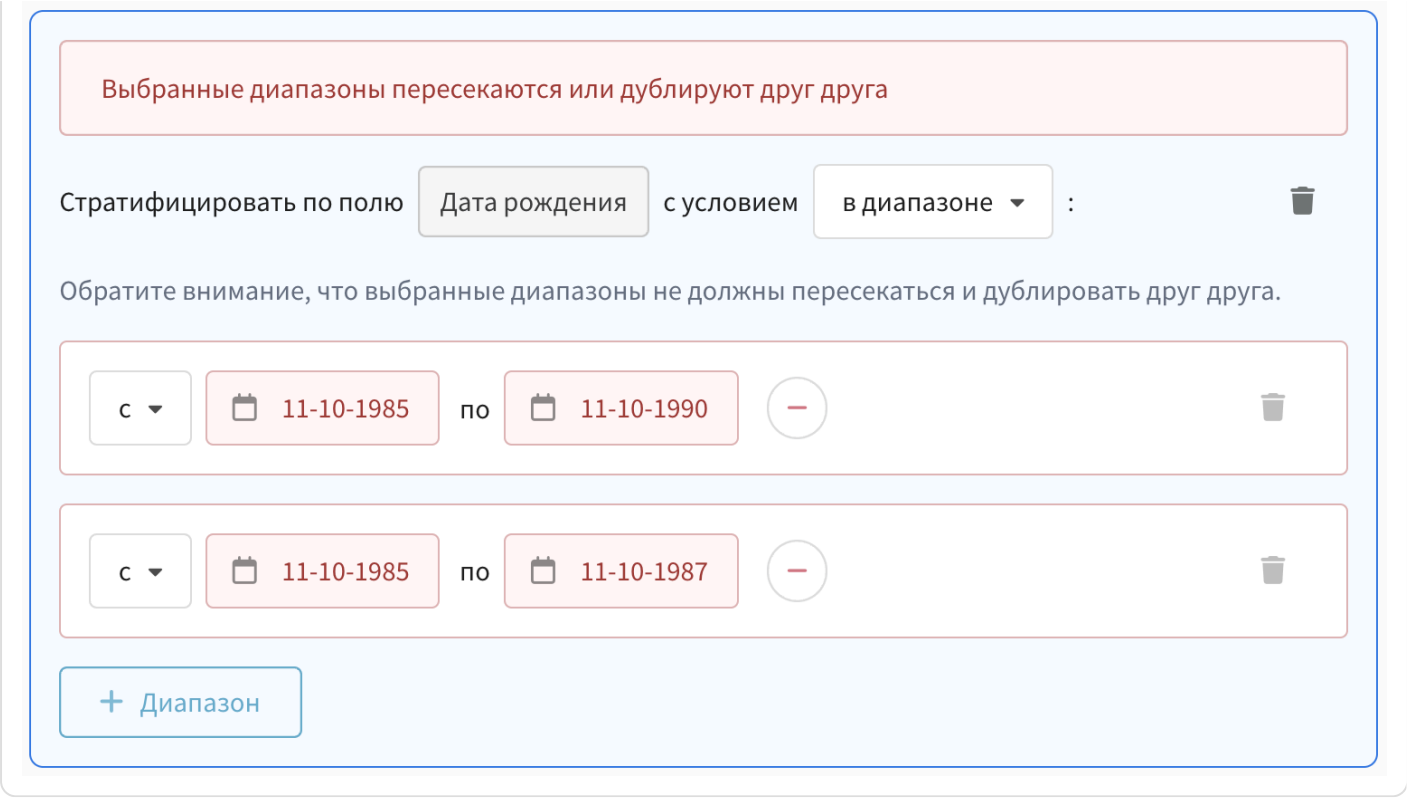


Задайте страты по одному или нескольким полям, определив условия для каждой группы: например, **диапазоны дат**, **числовые значения**, **группы элементов** из списка и т.д. Настройка стратификации осуществляется через пользовательский интерфейс, где можно последовательно добавлять страты и задавать параметры для каждой из них:



CAUTION

Обратите внимание, что значения в стратах не должны пересекаться или дублировать друг друга:



Поддерживаемые типы полей для стратификации:

- **boolean** — логическое значение (например, да/нет);
- **date** — дата (например, дата рождения, регистрации);
- **enum** — список фиксированных значений (например, регион, категория);
- **float** — дробные числовые значения;
- **integer** — целочисленные значения;
- **unicode** — строковые значения (например, текстовые метки или имена групп).

Для каждого из этих типов доступны соответствующие условия: **диапазоны**, **конкретные значения**, **группы значений** и др.

TIP

Общее число комбинаций условий в стратификации **не должно превышать 64**. Это значение рассчитывается **перемножением количества условий в каждой добавленной страте**.

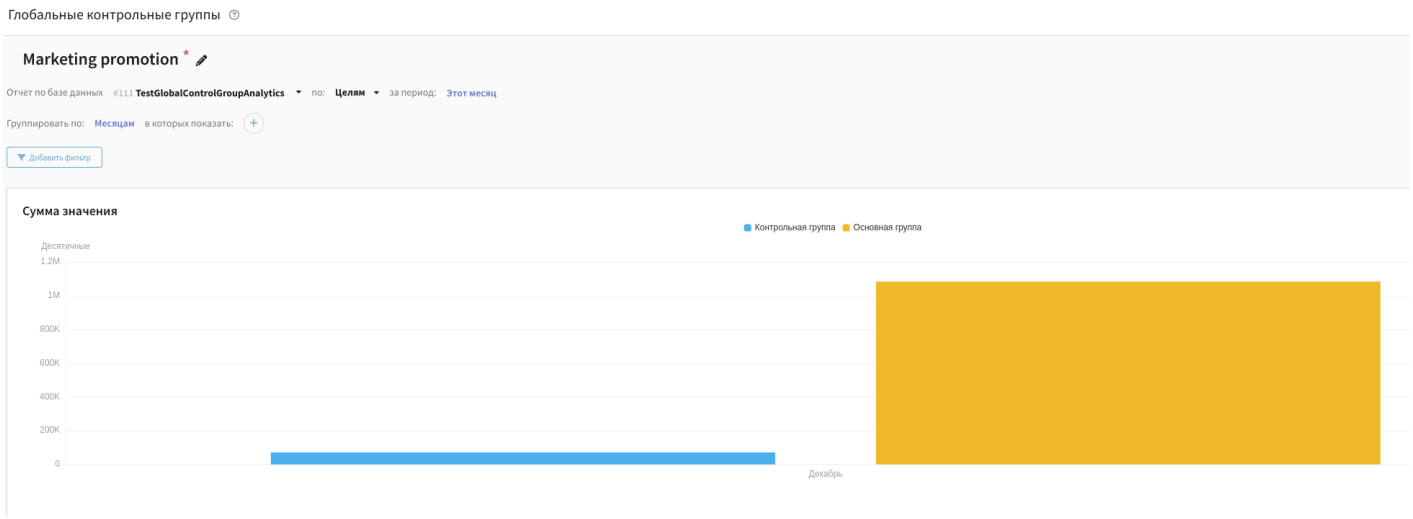
Например, если вы выбрали поле строкового типа и задали в нём 3 условия, а затем добавили числовое поле с 4 условиями, общее число комбинаций составит: 3 × 4 = 12 условий.

Если количество комбинаций превышает 64, система не позволит сохранить настройки — это необходимо для стабильной работы и корректной генерации выборки.

Статистика по ГКГ

Сравнение метрик по аудитории, получающей коммуникации, и по аудитории контрольной группы можно посмотреть в разделе **Аналитика — Глобальные контрольные группы**:

Перед вами появится график сравнения основной группы и контрольной. Вы можете выбрать **пиксель** и **цель** для сравнения показателей, полученных по данным пикселей отслеживания, а также канал продаж для показателей Маркета:



Подробнее об аналитике по глобальным контрольным группам можно прочитать здесь.

Менеджер подписок

Описание

Менеджер подписок Altcraft — это страница, позволяющая подписчикам управлять своими данными профиля, а также подписками на ресурс: **останавливать**, **возобновлять** или **приостанавливать** их на определенный срок (14, 30, 60 или 90 дней). Если в вашей компании уже используется свой менеджер подписок, то его можно интегрировать в платформу.

> **INFO**
>
> Встроенный менеджер подписок работает только с той подпиской, на которую было отправлено сообщение с ссылкой на менеджер.

Настройка менеджера подписок

Менеджер подписок автоматически активируется при создании **базы профилей**. В обязательном поле **“Заголовок менеджера подписок”** укажите название, которое будут видеть пользователи при входе в интерфейс. В соседнем поле выберите **язык интерфейса** менеджера.

> **INFO**
>
> Язык менеджера задается в настройках базы профилей и **не зависит** от региона пользователя или языка браузера. Если выбран русский язык, интерфейс всегда будет отображаться на русском.

При переходе по ссылке пользователь попадает на страницу встроенного менеджера подписок. Интерфейс менеджера по умолчанию отображает основные данные получателя: **Имя, Фамилию, Email-адрес** и **Номер телефона**:

Настройка отображаемых полей

В менеджере подписок доступна настройка видимости полей профиля. Помимо обязательных email и телефона, вы можете включить отображение стандартных или дополнительных полей — просто отметьте соответствующие опции в колонке **“Добавить в менеджер подписок”**.

> **CAUTION**
>
> Любые изменения, внесённые пользователем в отображаемые поля менеджера подписок, автоматически обновят соответствующие данные в его профиле. Убедитесь, что изменение этих полей не приведет к ошибкам.

После сохранения изменений отмеченные поля отобразятся в менеджере подписок:

Категории и дополнительные настройки

Категории в менеджере подписок

Категории помогают детализировать тип контента, на который подписан пользователь. Если пользователь отпишется от одной категории, он продолжит получать контент из других.

В отличие от подписки на ресурс, у категорий всего два статуса (**Подписан**/**Отписан**). Статус можно изменить вручную в профиле или через API-запросы. Для того чтобы подписчик мог управлять категорией самостоятельно, добавьте ее в менеджер подписок:

После сохранения изменений отмеченные категории отобразятся в менеджере подписок в разделе **“Категории”**:

Подробнее про **категории ресурса** — в отдельной статье.

Дополнительные поля

В каждый канал ресурса можно добавить дополнительные поля, аналогичные тем, что используются в базе профилей. Они помогают хранить дополнительную информацию о подписке.

Создание дополнительных полей

- Перейдите в **настройки ресурса.**
- Нажмите “Добавить поле” и выберите его тип:
    - **Целое число** — положительные и отрицательные целые числа. Например, можно указать лимит заказов, бонусные баллы и др.
    - **Число с точкой** — положительные и отрицательные числа с плавающей точкой. Примеры использования: “Средний чек”, “Коэффициент” и др.
    - **Строка** — текст на любом языке, включая цифры, специальные символы, пробелы. Используется стандарт Unicode. Примеры использования: “Комментарии к заказам”, “Интересы” и др.
    - **Логический** — поле, которое предоставляет возможность выбора из двух вариантов **True** или **False** (Да или Нет). В менеджере подписок отображается в виде ползунка. Примеры использования: “Уведомлять о начале распродажи”, “Получать эксклюзивные предложения” и др.
- Отметьте поля, которые нужно отобразить в менеджере подписок.

> **CAUTION**
>
> Получатели могут изменять значения дополнительных полей, добавленные в менеджер подписок. Убедитесь, что изменение этих полей не приведет к ошибкам.

В менеджере подписок выбранные поля отобразятся в разделе **“Подробности”**:

Подробнее про **дополнительные поля** можно прочитать в этой статье.

Интеграция менеджера подписок

Добавление ссылки в шаблон сообщения

Чтобы добавить ссылку на менеджер подписок:

- Откройте редактор шаблона.
- Выберите “Вставить ссылку”.
- Укажите тип “Ссылка на менеджер подписок”.

Использование альтернативного менеджера подписки

Для подключения стороннего менеджера подписок:

- Создайте пользовательскую ссылку отписки.
- В поле “Ссылка” укажите URL-адрес вашего менеджера подписок.

Теперь при использовании в шаблоне ссылки отписки получатель будет переходить на страницу менеджера подписок.

CAUTION

В этом случае переход по ссылке менеджера подписок **автоматически отпишет профиль** от ресурса. Чтобы этого избежать, необходимо настроить выбранный менеджер. Для этого обратитесь к администратору вашей платформы.

Каналы коммуникации

Связь с клиентами

В платформе Altcraft по умолчанию доступно 7 каналов связи: **Email, SMS, Push** (браузерные и мобильные), а также **Telegram, WhatsApp*, Viber™ и Notify.** В той или иной конфигурации платформы вам могут быть доступны пользовательские (кастомные) каналы, например, мессенджеры или баннеры. В этом разделе собрана информация по настройке и отладке каналов коммуникации с клиентами.

Клиенты, профили которых попадают в **Базу данных**, подписываются на получение контента от определённого **Ресурса** — по одному или нескольким каналам связи.

Каждая **Рассылка** в Altcraft Platform отправляется **по одному** из каналов связи. Для проведения омниканальных каскадных рассылок используются **Сценарии** — цепочки автоматизации маркетинга. В качестве контактных данных используются **подписки клиентов** на ресурс.

Контент для **всех каналов** связи хранится в **Шаблонах сообщений**. Рассылки по разным каналам связи используют один шаблон, что позволяет быстрее отслеживать события в омниканальных коммуникациях.

Email

	Электронная почта позволяет отправлять клиентам наибольшее количество контента, так как в письме могут использоваться различные элементы графического оформления, а также интерактивные элементы, повышающие уровень вовлечённости клиентов. Для проведения Email рассылок вам потребуется домен отправки, который необходимо указать в Ресурсах, использующих Email канал.

INFO

Организуйте свою первую Email рассылку с помощью **Пошагового руководства**!

INFO

В этой статье собраны **рекомендации по взаимодействию с почтовыми провайдерами** во время проведения массовых рассылок.

SMS

	Сервисы коротких сообщений помогают эффективнее всего организовать транзакционные уведомления, так как для получения SMS клиенту достаточно оставить вам свой номер телефона. Все сообщения так или иначе открываются, в отличие от Email. Отправку SMS осуществляют операторы сотовой связи. Они же регламентируют использование **CALLER ID**, который необходимо указать в Ресурсах, использующих SMS канал.

INFO

Организуйте свою первую SMS рассылку с помощью **Пошагового руководства**!

INFO

В **этой статье** собрана информация о **готовых интеграциях с SMS шлюзами**.

Push

	Push уведомления гораздо выгоднее SMS, так как их отправка фактически ничего не стоит. В то же время, с помощью мобильных и браузерных уведомлений можно существенно увеличить вовлечённость клиентов.

INFO

Для работы с Push каналом необходимо активировать его в **Панели администратора**.

INFO

Организуйте свою первую Web push-рассылку с помощью **Пошагового руководства**!

INFO

Руководство по организации рассылки в мобильные приложения вы найдёте **по этой ссылке**!

Telegram

	Для коммуникации с подписчиками в Telegram доступны 2 канала: **Telegram Bot** и **Telegram Group.** Telegram Bot позволит вам персонально отправлять сообщения подписчикам, которые вступили в чат с ботом. С помощью Telegram Group вы сможете через бота публиковать посты в Telegram-группе или канале.

INFO

Пошаговое руководство позволит вам организовать отправку персональных сообщений подписчику в Telegram.

Обратите внимание, при использовании канала Telegram Bot со стороны платформы происходит только отправка рассылок. Сбор и передача в платформу данных о пользователях, которые вступили в чат с Telegram ботом, осуществляется на стороне клиента.

INFO

Для коммуникации с подписчиками в Telegram группе или канале используйте **это руководство**.

WhatsApp

	WhatsApp* — это эффективный канал коммуникации, который позволяет выстроить диалог с клиентом в режиме реального времени. Через этот канал вы можете сообщать подписчикам о важных событиях, изменениях по статусам заказа и предстоящих мероприятий. Открываемость писем в мессенджере достигает 90%.

INFO

Организуйте свою первую WhatsApp* рассылку с помощью **Пошагового руководства**!

Viber™

	Viber — быстрый и удобный канал коммуникации. Отправляйте промо-рассылки, транзакционные и триггерные сообщения вашим подписчикам. Стоимость таких отправок дешевле, чем SMS.

INFO

Интеграция с Viber™ может быть реализована через провайдеров Devino Telecom, SMS Traffic, MTC OmniChannel или Rapporto. **Это руководство** поможет вам создать свою первую Viber™-рассылку.

Notify

	Notify — эффективный канал для бизнес-коммуникации с клиентом. Отправляйте транзакционные и сервисные сообщения через "Вконтакте" и "Одноклассники".

INFO

Это руководство поможет вам создать свою первую Notify-рассылку.

Ограничения в каналах коммуникации

Функция	Email	SMS	Push	WhatsApp™	Viber™	Notify™	Пользовательские (кастомные) каналы
Мультивариантное тестирование	✅	✅	✅	❌	❌	❌	❌
Сохранение веб-версий	✅	✅	✅	❌	❌	❌	❌
Подписка на канал из сценария	✅	✅	❌	❌	❌	❌	❌

Функция	Email	SMS	Push	WhatsApp™	Viber™	Notify™	Пользовательские (кастомные) каналы
Стоп-списки	✅	✅	❌	❌	❌	❌	❌
Карта кликов	✅	❌	❌	❌	❌	❌	❌

Подробнее о технических ограничениях кастомных каналов вы можете прочитать в документации администратора.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

** Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

SMS-канал

SMS: Готовые интеграции

Описание

Altcraft Marketing использует интеграции со шлюзами операторов сотовой связи для отправки SMS подписчикам. Если готового решения на данный момент нет, мы можем разработать его в рамках пилотного проекта. Подробнее вы можете узнать, написав на team@altcraft.com.

Готовые интеграции

Для **SMS** доступны следующие готовые интеграции:

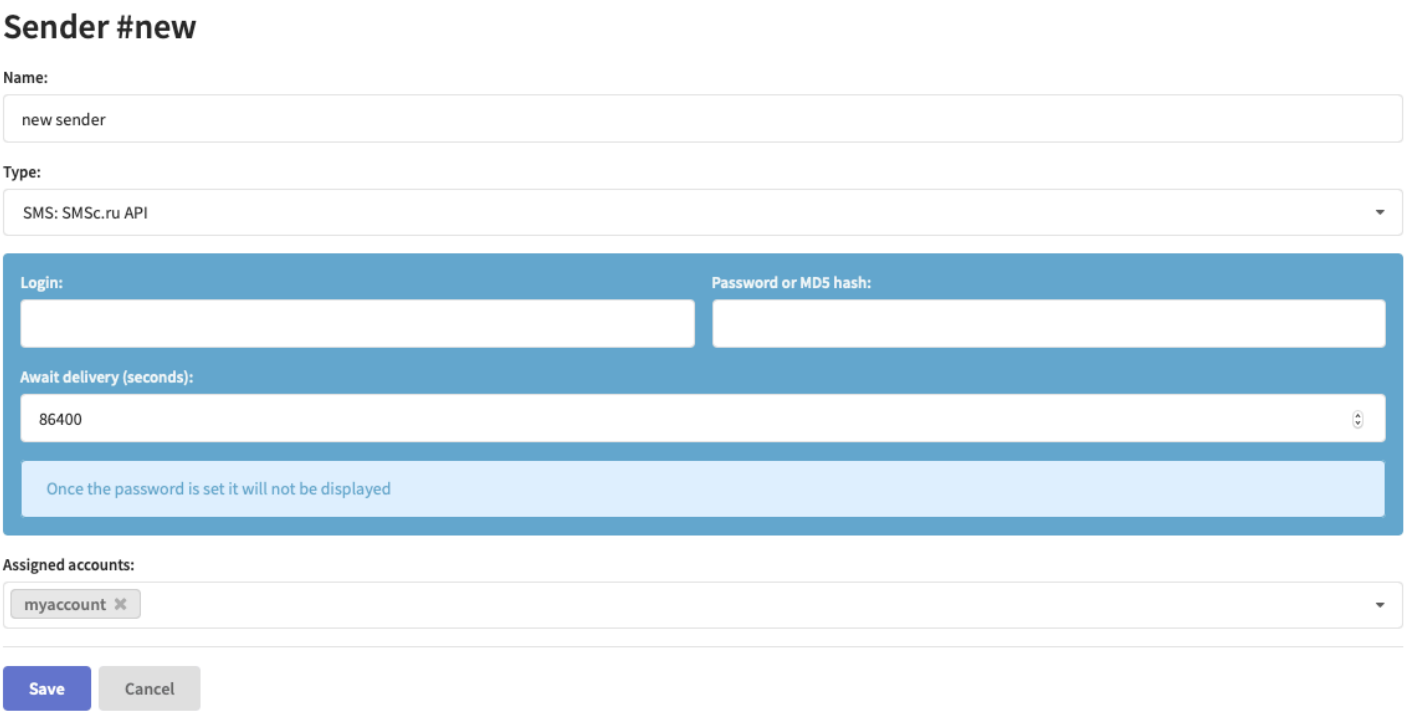
- **SMSc.ru** — интеграция с SMS-шлюзом от “СМС-Центр”.
  Ссылка на документацию: https://smsc.ru/api/
- **SMSimple.ru** — интеграция с SMS-шлюзом от “Юлабс”.
  Ссылка на документацию: http://www.smsimple.ru/api/
- **BulkSMS.com** — интеграция с SMS-шлюзом от “Celerity Systems”.
  Ссылка на документацию: https://www.bulksms.com/developer/json/v1/#
- **Zagruzka** — интеграция с SMS-платформой Zagruzka.
  Ссылка на документацию: https://docs.zagruzka.com/pages/viewpage.action?pageId=327775
- **Devino.Online SMS** — интеграция с API многофункционального шлюза Devino Telecom.
  Ссылка на документацию: https://docs.devino.online/0.2.0/ru
- **SMSTraffic** — интеграция с SMS-шлюзом от “СМС-Трафик”.
  Ссылка на документацию: https://lk.smstraffic.ru/techdocs
- **Infobip** — интеграция с SMS-шлюзом от “Infobip”.
  Ссылка на документацию: https://www.infobip.com/docs/api#channels/sms/send-sms-message
- **MTS** — интеграция с SMS-шлюзом от “МТС”.
  Ссылка на документацию: https://support.mts.ru/mts_marketolog/Rassilki-po-svoei-baze-PRO-i-API-k-nim/dokumentatsiya-rest-api
- **MFMS EDNA** — интеграция с SMS-шлюзом от “EDNA”.
  Ссылка на документацию: SMS-MFMS-PROTOCOL-HPG.pdf
- **Sky Media** — интеграция с SMS-шлюзом Sky Media. Ссылка на документацию: https://sky-ms.ru/api.html
- **Play Mobile** — интеграция с SMS-шлюзом Play Mobile (документация).
- **Rapporto** — интеграция с SMS-шлюзом от Rapporto (ex. Zagruzka). Ссылка на документацию: https://doc.rapporto.ru/api/http/http_main.html
- **VK Notify** — интеграция с SMS-шлюзом сервиса Notify.
- **СберМобайл** — интеграция с SMS-шлюзом от СберМобайл. Ссылка на документацию: SMS-SBER-PROTOCOL.pdf
- **Stream telecom** — интеграция с SMS-шлюзом от “Stream telecom”. Ссылка на документацию: https://stream-telecom.ru/solutions/integrations/
- **Intellin** — интеграция с SMS-шлюзом от Intellin.
- **Seven Tech** — интеграция с SMS-шлюзом от SEVENTECH. Ссылка на документацию: https://docs.seven.tech/

TIP

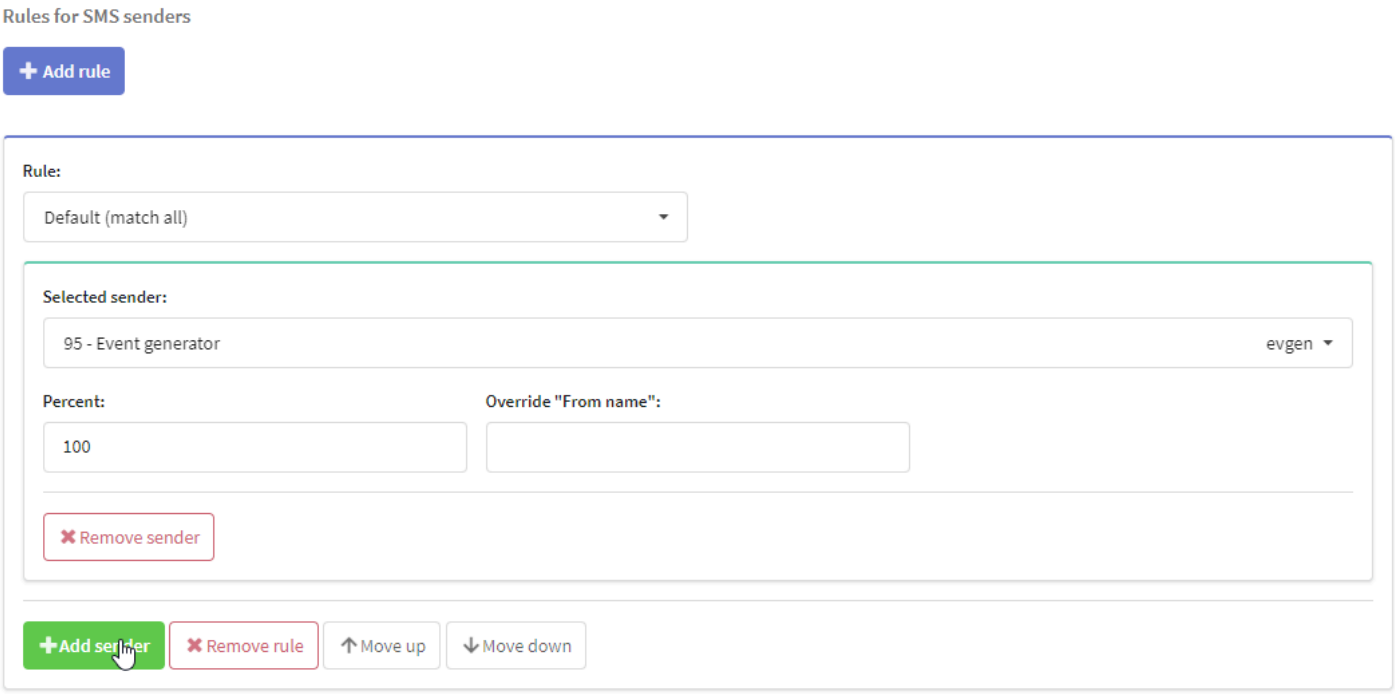
Для корректной работы некоторых интеграций нужно предоставить какую-либо информацию провайдеру (например, для получения статусов отправленных SMS-сообщений необходимо предоставить Rapporto URL для приема статусов доставки сообщений), либо указать специфичный для провайдера параметр. Для этого обратитесь к администраторам платформы.

Настройка отправки

Чтобы начать рассылать SMS, настройте сендер в панели администратора платформы, добавив учётную запись вашего провайдера:



После этого добавьте правила отправки SMS настроенного сендера в один из **виртуальных отправщиков**:



После согласования с провайдером CALLER ID укажите его в **Ресурсе** SMS подписок:

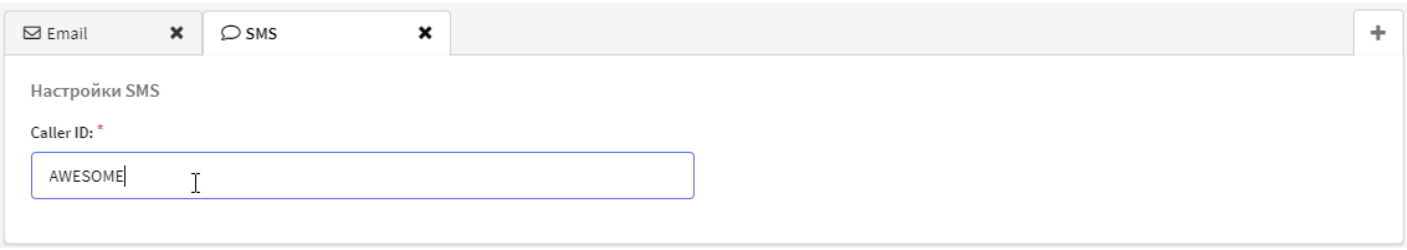


Схема работы каналов коммуникации

Общая структура

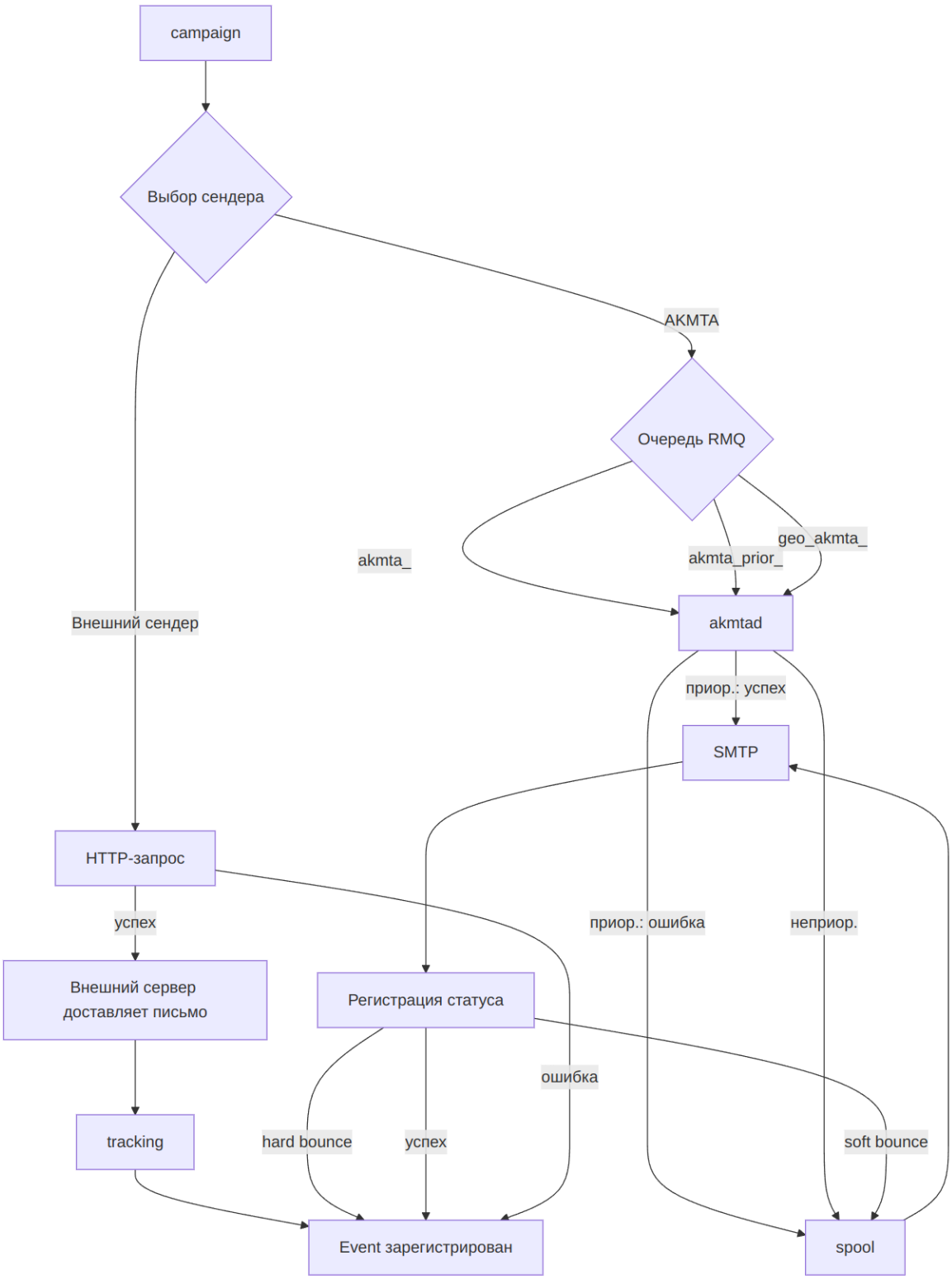
При запуске кампании запускается бинарный файл `campaign`, который:

- обрабатывает шаблон и бизнес-логику (стоп-списки, лимиты, подписки, фильтры и т.д.);
- формирует структуру сообщений для отправки;
- направляет готовые сообщения в очереди RabbitMQ;
- фиксирует ошибки в журнале кампании.

Далее каждое сообщение попадает в соответствующий сервис обработки в зависимости от канала.

Email

Для email-канала поддерживается несколько видов сендеров (отправщиков): AKMTA (сендер Altcraft) и внешние сендеры. В зависимости от используемого сендера, схема работы может отличаться.



Процесс отправки

AKMTA

Кампания отправляет сообщения в очереди:

- `akmta_<sender_id>` — обычная очередь;
- `akmta_prior_<sender_id>` — приоритетная очередь;

- `geo_akmta_<sender_id>` — очередь рассылки по часовому поясу. Она приоритетнее обычной очереди, но менее приоритетна `akmta_prior`. Используется для того, чтобы даже при высокой нагрузке на платформу клиент получал коммуникации в пределах выбранного часа в его часовом поясе.

Из очередей сообщения обрабатывает сервис **`akmtad`**. Дальнейшее поведение процесса `akmtad` зависит от приоритетности сообщения:

- Для **неприоритетных** сообщений — сохраняет сообщения на диск, периодически читает и отправляет их по стратегии отправки, заданной для конкретного `sender_id`;
- Для **приоритетных** — производит попытку немедленной отправки. Если установлены правила повторной отправки, то при неудаче сообщение будет сохранено на диск.

На диске сообщения попадают в очередь **`spool`**. Отправка оттуда происходит в соответствии с правилами сендера, платформа “читает” `spool` с периодичностью равной `MTA_EXCHANGE_SCAN_PERIOD`. Обратите внимание, что эти сообщения больше **не хранятся в RabbitMQ**.

Схема отправки выглядит следующим образом:

`campaign` —> Очередь RMQ (**`akmta`** / **`akmta_prior`**) —> **`akmtad`** —> (при неудаче — запись в очередь **`spool`** на диске) —> **`SMTP`**

Остальные сендеры

Отправка происходит от бинарного файла `campaign` через **внешний HTTP-запрос**. Если платформа не может отправить сообщение, то регистрируется **ошибка кампании**. Обратите внимание, что подобная ошибка не засчитывается как “Недоставка” и в отчет по недоставкам не вносится.

Регистрация статусов

AKMTA

Статусы доставки регистрируются **сразу** по:

- коду ответа SMTP-сервера;
- bounce-паттернам (анализ шаблонов возвратов).

Остальные сендеры (#other-status)

Статус события обрабатывают трекинг-сервисы платформы.

Повторная отправка

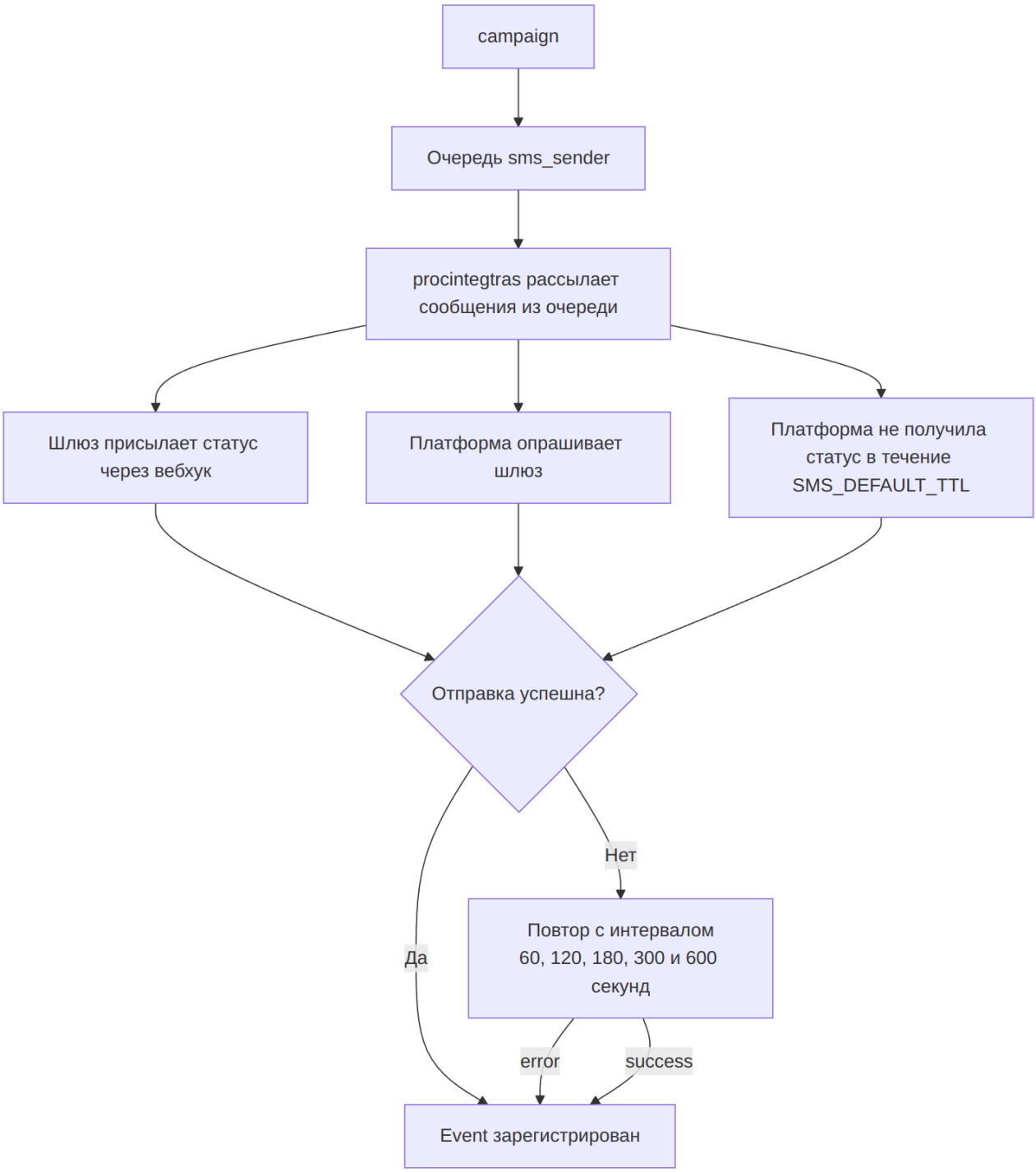
AKMTA

- Правила повторов задаются отдельно через Панель администратора. О том, как это сделать, можно прочесть в документации администратора;
- В зависимости от SMTP-кода регистрируются статусы **hard bounce** или **soft bounce**;
- Для soft и hard bounce правила повторной отправки могут отличаться.

Остальные сендеры

Для остальных сендеров повторные попытки отправки не предусмотрены. В случае ошибки при отправке http-запроса, платформа не производит никаких действий.

SMS



Процесс отправки

Кампания отправляет сообщения в одну единую очередь — `sms_sender`. Затем сервис `procintegras` рассылает сообщения из очереди. Некоторые шлюзы поддерживают отправку батчем.

Регистрация статусов

Регистрация статуса работает одним из следующих способов:

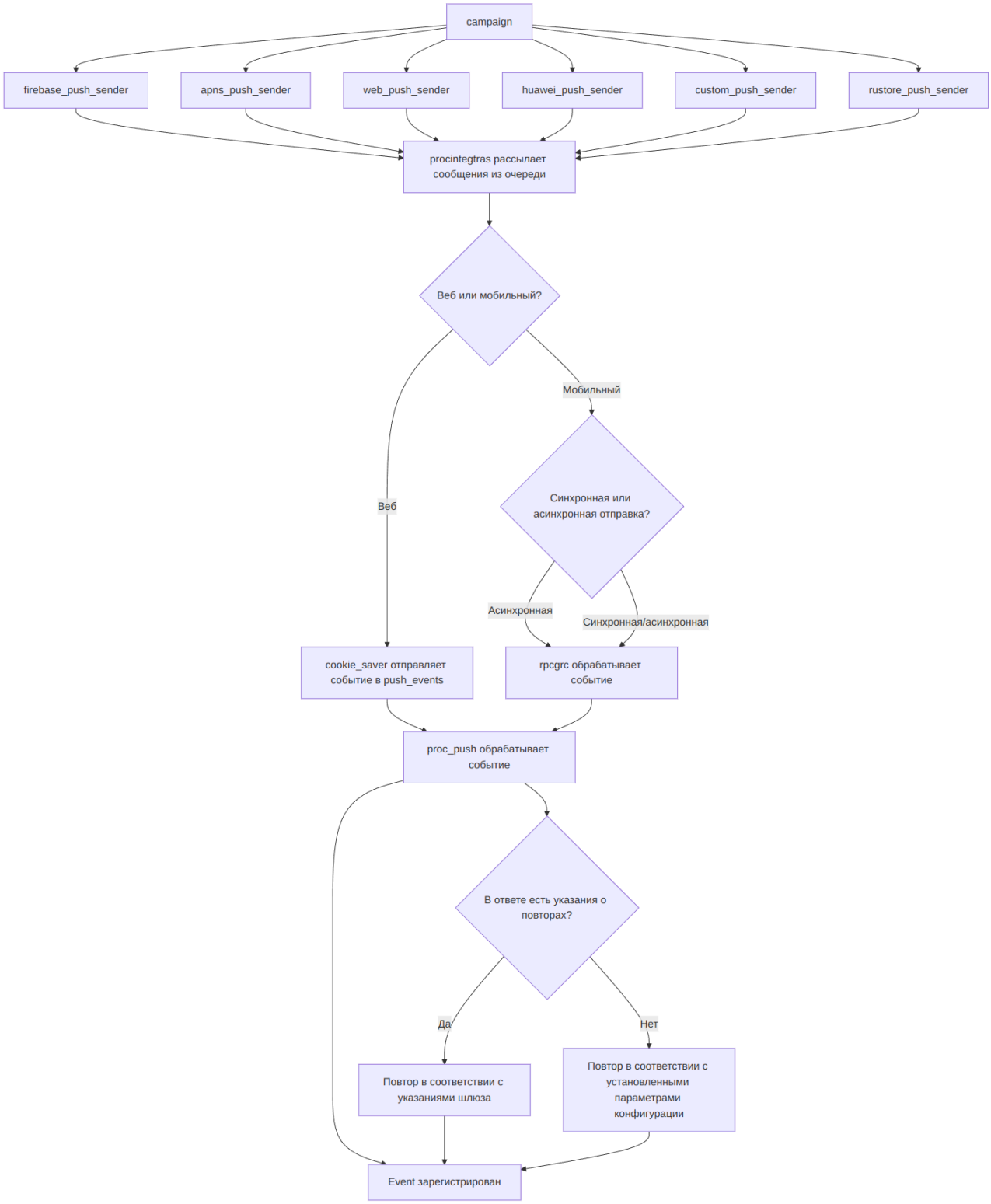
- Шлюз присылает статус сообщения через вебхук. Статус обрабатывается сервисом `procsmslisten`;
- Платформа “опрашивает” шлюз. Статус обрабатывается сервисом `procsmsev`.

Время жизни SMS-сообщения указывается в параметре конфигурации `SMS_DEFAULT_TTL` (по умолчанию — 1 сутки). Платформа регистрирует недоставку, если статус не получен в течение этого времени.

Повторная отправка

Повторные отправки SMS-сообщений происходят через 60, 120, 180, 300 и 600 секунд с момента последней неудачной попытки. Максимальное количество попыток — 5.

Push



Процесс отправки

Кампания отправляет сообщения в очередь. Очередь зависит от **провайдера**:

- **APNs** — `apns_push_sender`
- **Firebase** — `firebase_push_sender`
- **Huawei** — `huawei_push_sender`
- **RuStore** — `rustore_push_sender`
- Браузерные push **Firefox** и **Safari** — `web_push_sender`
- Прочие push-провайдеры — `custom_push_sender`

Затем сервис `procintegras` рассылает сообщения из очереди. Некоторые шлюзы поддерживают отправку батчем.

Регистрация статусов

За обработку статуса **браузерных** веб-пушей отвечает сервис `cookie_saver`, который отправляет событие в `push_events`. Дальнейшая обработка происходит в `procpush`.

Регистрация статусов для **мобильных** push происходит схожим образом, главное отличие в том, что обработка асинхронных отправок происходит в очереди `push_events`, а синхронные сразу обрабатываются сервисом `procrpc`.

Время жизни push-сообщения указывается в параметре конфигурации `MTA_DEFAULT_TTL` (по умолчанию — 3 суток). Недоставка регистрируется, если не получен статус в течение этого времени.

> **TIP: Синхронные и асинхронные запросы**
>
> **Синхронные запросы** — те запросы, при которых платформа регистрирует ответ **по результату** отправки, т.е. платформа **ждет ответа** на свой запрос.
> **Асинхронные запросы** — те запросы, при которых платформа регистрирует ответ по постановке отправки **в очередь**, т.е. платформа **не ждет ответа** на свой запрос.

Повторная отправка

Сендер смотрит на ответ шлюза и на возвращаемые заголовки и, в зависимости от ответа, устанавливает правило повтора. Если в ответе шлюза содержатся конкретные указания (например, шлюз “просит” попробовать заново через 5 минут), то повторная отправка будет выполнена в соответствии с ними. Если ответа нет, или в нем не содержатся точные указания, то срабатывают правила, установленные в конфигурации платформы:

- `PUSH_SENDER_RETRY_MAX_AMOUNT` — максимальное количество попыток повторной отправки push-сообщений. По умолчанию — 3;
- `PUSH_SENDER_RETRY_INTERVAL` — время между попытками повторной отправки (в секундах). По умолчанию — 1800;
- `PUSH_SENDER_RETRY_SCAN_INTERVAL` — периодичность сканирования базы повторных отправок на наличие новых записей (в секундах). По умолчанию — 60.

Подробнее об этих параметрах можно прочесть в документации администратора.

Каналы мессенджеров

Процесс отправки

Кампания отправляет сообщения в очередь. Далее сообщения в очереди RabbitMQ обрабатываются в зависимости от канала:

- **Telegram Group, Telegram Bot** — очередь `piper_messages`, обрабатывается в `procpiper`;
- **Viber, WhatsApp, Notify** — обрабатывается в `procintegras`.

Регистрация статусов

- **Telegram** — регистрируется в зависимости от http-ответа Telegram. Доставка регистрируется, если нет ошибок во время отправки в API Telegram. Соответственно, недоставка регистрируется, если платформа получила в ответ ошибку при отправке;
- **Viber, WhatsApp, Notify** — статусы попадают в платформу через вебхуки. Если во время отправки сообщения получена ошибка, то регистрируется событие недоставки. Статусы “прослушиваются” сервисом `trklisten`.

Повторная отправка

В данный момент повторные отправки для сообщений в мессенджерах **не предусмотрены.**

INFO: Ограничения по отправкам

Для Telegram действует ограничение в 28 запросов в секунду на один токен. Для Notify по умолчанию установлено ограничение 20 запросов в секунду, значение можно изменить в параметре конфигурации `VKNOTIFY_SEND_LIMIT_PER_SECOND`, максимально — 50 запросов в секунду.
Для Viber по умолчанию установлен максимум в 50 запросов в секунду, изменяется в параметре `VIBER_DEVINO_SEND_LIMIT_PER_SECOND`. Для WhatsApp по умолчанию установлено ограничение в 20 запросов в секунду (`WHATSAPP_SEND_LIMIT_PER_SECOND`); у различных провайдеров могут быть установлены свои ограничения по запросам.

Настройка собственного from-домена

Email: Настройка собственного from-домена

Настройка SPF

Sender Policy Framework (SPF) — технология, позволяющая проверить не подделан ли домен отправителя. Для её работы необходимо добавить на отправляющий домен (сендер домен, фром домен) специальную TXT-запись, которая определяет политику разрешения отправки этого домена с различных хостов.

Мы рекомендуем внести записи разных версий:

Домен	Тип записи	Содержимое
example.com.	TXT	`v=spf1 include:spf.akmta.net ~all`
example.com.	TXT	`spf2.0/pra include:spf.akmta.net ~all`

На домене spf.akmta.net содержится актуальный список IP адресов сендеров Altcraft.

Если ваш фром домен уже используется для рассылок, то уточните, какие SPF-записи уже существуют и добавьте к ним “include:spf.akmta.net”.

Настройка DKIM

Технология DomainKeys Identified Mail (DKIM) добавляет в письмо цифровую подпись, связанную с from-доменом. Подпись автоматически проверяется на стороне получателя, после чего используется для уточнения репутации и помечается для пользователя. Для подписи письма используется приватный ключ, который устанавливается на стороне отправщика и более никому не известен. Публичный ключ располагается в виде специальной TXT-записи на поддомене from-домена.

Необходимо внести следующие записи (используется ключ Altcraft, ak):

Домен	Тип записи	Пример содержимого
akru2022._domainkey.example.com.	TXT	`v=DKIM1; k=rsa; p=<ваш персональный ключ>`
_domainkey.example.com.	TXT	`o=-;`

Настройка DMARC

Технология DMARC позволяет почтовому серверу решить, что делать с почтой, если с DKIM и SPF-записями что-то не так.

Мы рекомендуем установить правила, запрещающие принимать сообщения, которые получены с чужих IP-адресов, либо не правильно подписаны.

Домен	Тип записи	Пример содержимого
_dmarc.example.com.	TXT	`v=DMARC1; p=reject; sp=reject; rua=mailto:report@example.com`

Настройка BIMI

Технология BIMI (Brand Indicators for Message Identification) используется совместно с SPF, DKIM и DMARC и используется для идентификации компании по брендовому изображению, помещаемому рядом с темой сообщения в инбоксе получателя. Почтовые провайдеры таким образом могут дополнительно валидировать отправителя.

> **NOTE**
>
> Аутентификацию писем BIMI поддерживают такие почтовые сервисы как Fastmail, Gmail, and Yahoo!

Чтобы использовать технологию, вам нужно указать путь до файла в формате **svg**. Изображение должно быть квадратным и не иметь дополнительных слоёв.

Внесите в настройках домена следующую запись:

Домен	Тип записи	Пример содержимого
default._bimi.example.com	TXT	`v=BIMI1; l=https://example.com/bimi/bimi.svg`

Вы можете также воспользоваться генератором записи на https://bimigroup.org/bimi-generator/

Настройка трекинга

На поддомене вашего фром домена необходимо внести следующую запись:

Домен	Тип записи	Пример содержимого
trk.example.com.	CNAME	`trk.aksend.net`

Дополнительно, вы можете использовать сколько угодно трекинг доменов, настройки будут аналогичные.

> **NOTE**
>
> Запись **trk.aksend.net** используется только для облачного решения Altcraft. Если вы используете On-premise решение, введите собственную запись.

Настройка CNAME для обработки bounces

Для облачных клиентов необходимо также настроить дополнительную CNAME-запись, направленную на домен `akmta.net`:

Пример записи:

```
ac-bounces.example.com IN CNAME akmta.net
```

Здесь `ac-bounces` представляет собой поддомен, а `example.com` — ваш from-домен. Этот поддомен будет использоваться в Return-Path, что позволит проходить SPF-аутентификацию и выполнять “выравнивание” для улучшения доставляемости писем.

Настройка и использование постмастеров

Email: Настройка и использование постмастеров

Описание

Постмастер (postmaster) — это сервис почтового провайдера, который отслеживает статистику email-рассылок: сколько писем было доставлено, сколько попало в спам, сбои при отправке, на какое количество писем пожаловались и т.д.

Регистрация постмастера **обязательна**, если вы планируете доставлять большие объемы писем, больше 10 тысяч в сутки. Постмастер также необходим при регистрации FBL.

> **TIP**
>
> FBL/CFL (FeedBack Loop) – это стандарт выдачи информации о жалобах на спам от провайдера услуг электронной почты отправителю писем. Крайне рекомендуется пользоваться этой технологией на всех ISP, где она доступна в том или ином виде. Узнать больше о FBL можно в этой статье.

Выбор постмастера зависит от того, какие почтовые домены чаще используются в вашей базе данных. Каждый постмастер выдает статистику только по своим доменам.

Наиболее популярные сервисы-постмастеры в России:

- Mail.ru Postmaster
- Gmail Postmaster Tools
- Outlook.com Postmaster
- Yahoo Complaint Feedback Loop Service

Как подключить Постмастер Mail.ru

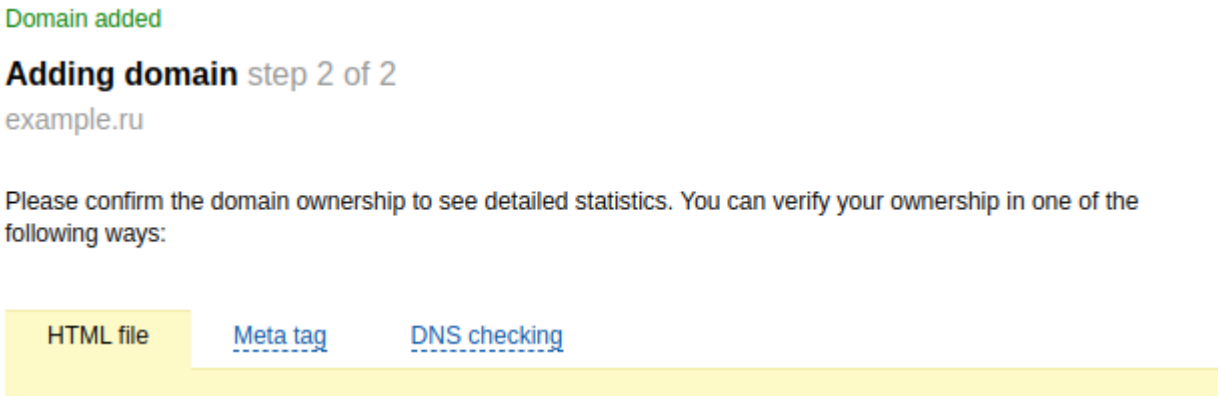
Шаг 1. Создайте аккаунт Mail.ru или авторизуйтесь с помощью одного из указанных способов.

Шаг 2. Перейдите на страницу добавления домена — https://postmaster.mail.ru/add/.

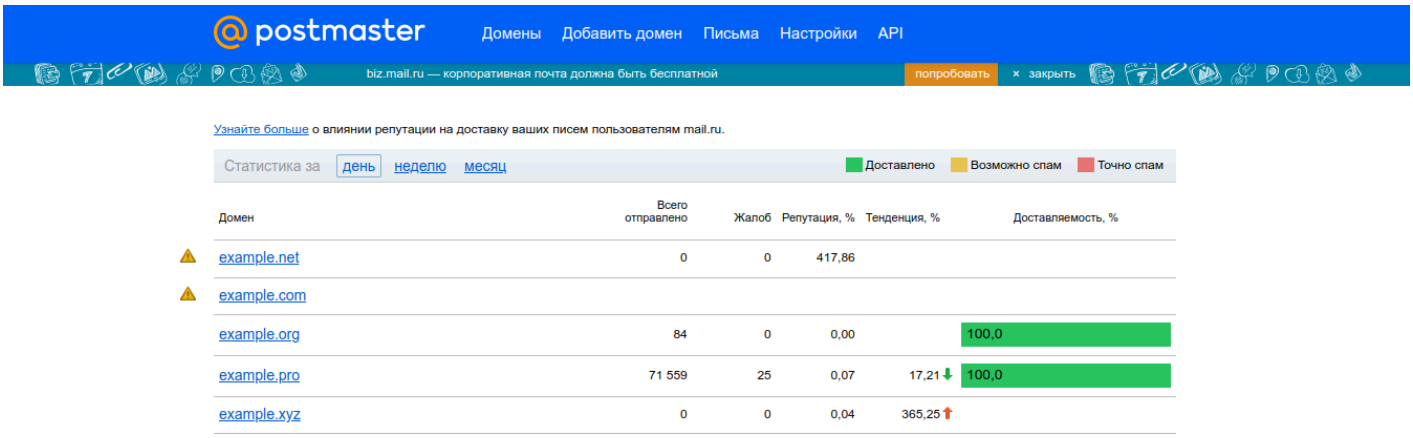
Шаг 3. Введите название домена, о котором вы хотите собирать статистику, и нажмите “Добавить домен”:

Шаг 4. Выберите один из способов подтверждения домена:

- **HTML-файл** — данный способ можно использовать только при наличии веб-сервера у домена. Выполните инструкции, описанные на вкладке, и нажмите “Подтвердить”
- **Meta-тег** — используется только при наличии собственного сайта. Добавьте тег с персональным кодом в раздел `head` главной страницы сайта и нажмите “Подтвердить”
- **DNS-проверка** — позволяет использовать постмастер даже при отсутствии собственного сайта. Значение DNS-записи необходимо разместить в DNS-записях домена, используя интерфейс хостинг-провайдера или регистратора доменного имени.



Если все инструкции выполнены верно, домен будет добавлен в постмастер. Описанным образом вы можете добавить неограниченное количество доменов:

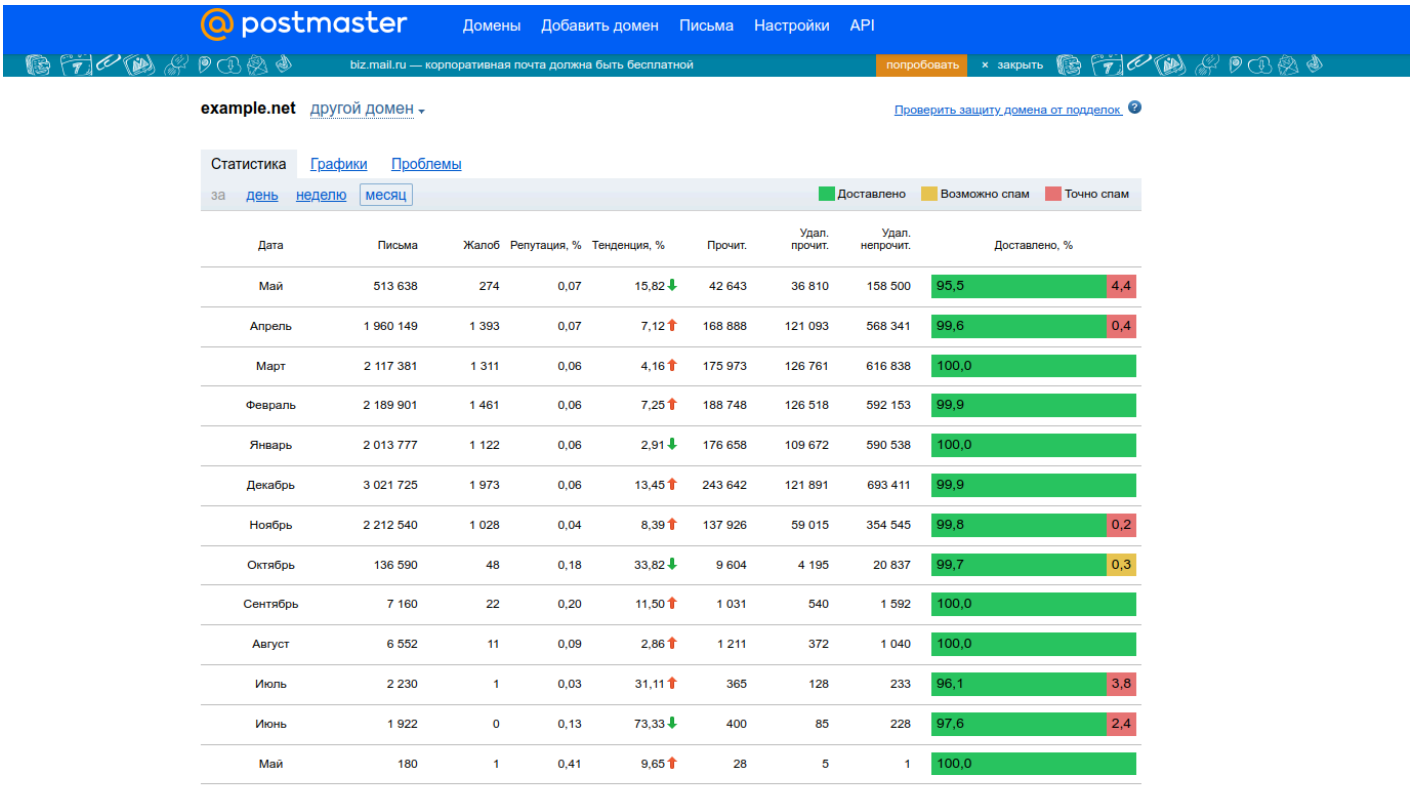


Настройка пересылки FBL

Постмастер Mail.ru поддерживает технологию **FeedBack Loop**. FeedBack Loop предоставляет доступ к статистике рассылок, группируя ее по дням, неделям и месяцам.

Данная статистика позволяет отслеживать следующие показатели:

- **Письма** — количество отправленных писем;
- **Жалоб** — количество нажатий на кнопку “Спам”;
- **Репутация, %** — средний процент жалоб за 30 дней. Она рассчитывается как отношение количества жалоб за 30 дней к общему количеству отправок за этот промежуток времени и выражается в процентах;
- **Тенденция, %** — изменение процента жалоб за 7 дней по отношению к проценту за 30 дней;
- **Прочит.** — количество открытых писем;
- **Удал. прочит.** — количество писем, удалённых после прочтения;
- **Удал. непрочит.** — количество писем, удалённых до прочтения;
- **Доставлено, %** — процентное соотношение следующих типов писем:
    - Доставлено — количество писем, доставленных в папку “Входящие”;
    - Возможно, спам — количество писем, доставленных в папку “Спам”;
    - Точно спам — количество писем, которые не были приняты серверами Mail.ru. В ответ на такие письма всегда приходит письмо с ошибкой “550 spam message discarded”.



Дата	Письма	Жалоб	Репутация, %	Тенденция, %	Прочит.	Удал. прочит.	Удал. непрочит.	Доставлено, %
Май	513 638	274	0,07	15,82 ↓	42 643	36 810	158 500	95,5 / 4,4
Апрель	1 960 149	1 393	0,07	7,12 ↑	168 888	121 093	568 341	99,6 / 0,4
Март	2 117 381	1 311	0,06	4,16 ↑	175 973	126 761	616 838	100,0
Февраль	2 189 901	1 461	0,06	7,25 ↑	188 748	126 518	592 153	99,9
Январь	2 013 777	1 122	0,06	2,91 ↓	176 658	109 672	590 538	100,0
Декабрь	3 021 725	1 973	0,06	13,45 ↑	243 642	121 891	693 411	99,9
Ноябрь	2 212 540	1 028	0,04	8,39 ↑	137 926	59 015	354 545	99,8 / 0,2
Октябрь	136 590	48	0,18	33,82 ↓	9 604	4 195	20 837	99,7 / 0,3
Сентябрь	7 160	22	0,20	11,50 ↑	1 031	540	1 592	100,0
Август	6 552	11	0,09	2,86 ↑	1 211	372	1 040	100,0
Июль	2 230	1	0,03	31,11 ↑	365	128	233	96,1 / 3,8
Июнь	1 922	0	0,13	73,33 ↓	400	85	228	97,6 / 2,4
Май	180	1	0,41	9,65 ↑	28	5	1	100,0

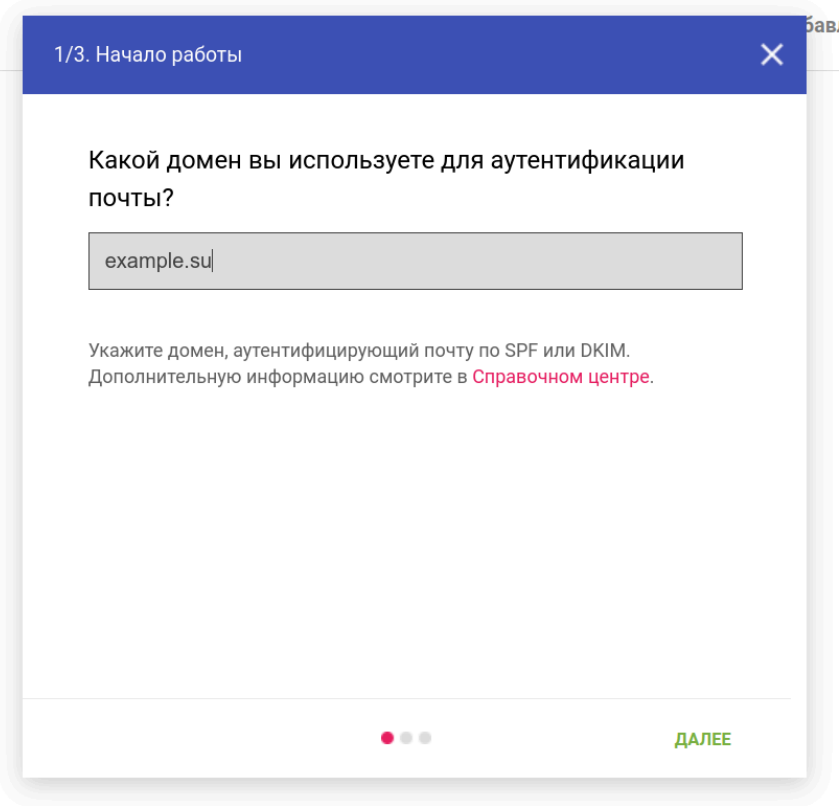
На странице настроек для всех доменов укажите адрес FBL. На него будет поступать информация о том, какие адресаты нажали кнопку “Спам” после получения письма. После каждой рассылки Постмастер Mail.ru будет отображать статистку по количеству отправленных писем и их состоянию: **“Доставлено”**, **“Возможно спам”**, **“Точно спам”**.

Как подключить Postmaster Tools by Gmail

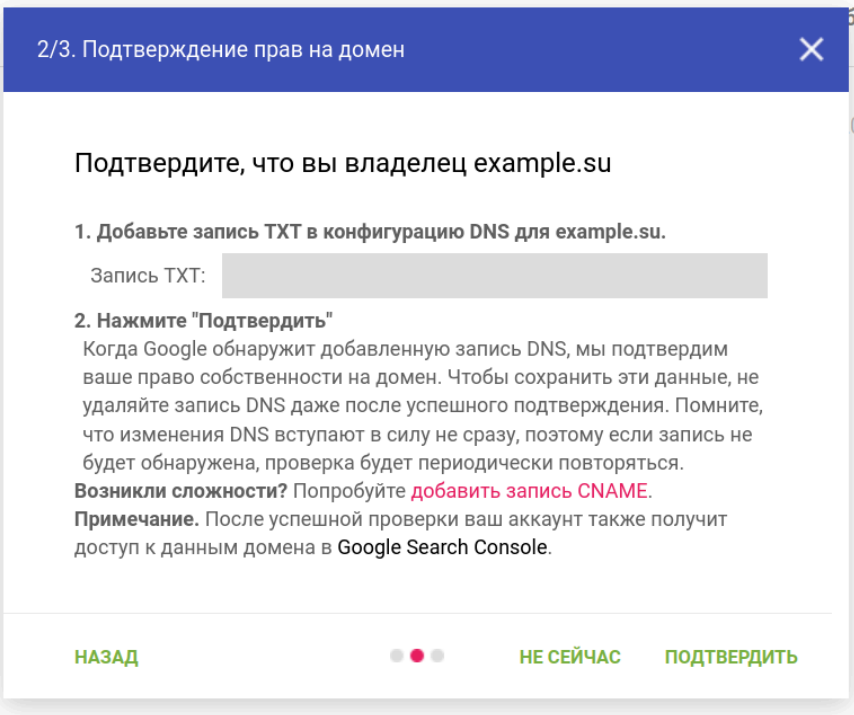
Шаг 1. Выполните вход в Postmaster Tools с помощью учетной записи Google.

Шаг 2. В правом нижнем углу нажмите кнопку “Добавить”.

Шаг 3. Введите свой домен аутентификации (DKIM или SPF) и нажмите “Далее”:



Шаг 4. Подтвердите право собственности на домен. Для этого получите доступ к записи DNS и добавьте **запись TXT**, предоставленную Google. После завершения вернитесь к Google Postmaster и выберите “Проверить”. Если запись не найдена, проверьте, правильно ли был добавлен TXT, и повторите попытку. Если у вас есть проблемы с добавлением TXT-записи, Google предлагает второй метод проверки, который включает в себя добавление записи “CNAME”:

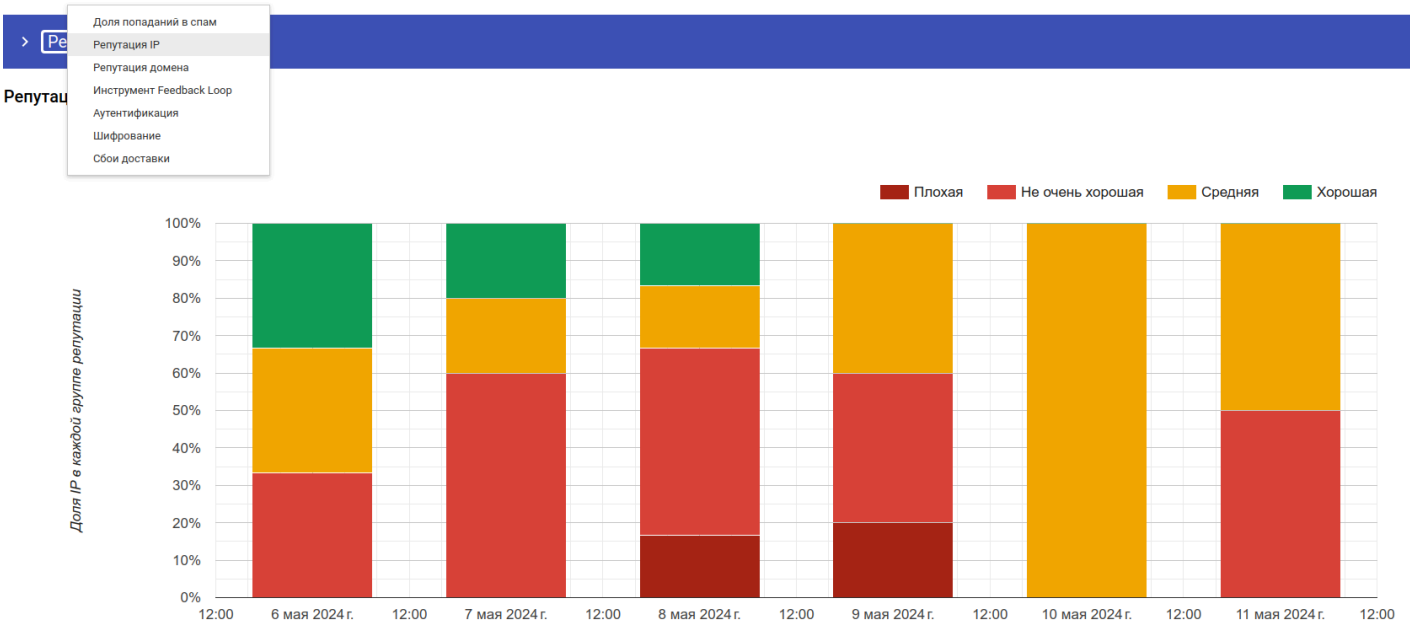


После подтверждения прав на домен, он отобразится в интерфейсе Postmaster Tools со статусом “Подтвержден”:

Домен	Статус	Дата добавления
example.net	Подтвержден	25 авг. 2017 г.
example.com	Подтвержден	25 авг. 2017 г.
example.org	Подтвержден	29 дек. 2020 г.
example.pro	Подтвержден	2 мар. 2023 г.
example.xyz	Подтвержден	14 мар. 2023 г.
example.io	Подтвержден	25 сент. 2023 г.
example.su	Подтвержден	26 февр. 2024 г.

Постмастер Gmail позволяет собирать информацию о ваших рассылках. Отслеживаются такие данные, как:

- доля попадания в спам;
- репутация IP и домена;
- информация об аутентификации (прохождении проверок DKIM, SPF и DMARC);
- информация о шифровании трафика (доля TLS);
- информация о сбоях доставки.



Все отчёты в постмастере Gmail формируются в виде графиков и таблиц с выборкой данных за промежуток от 7 до 120 дней. Информация в отчетах может отсутствовать в случае малого количества отправок на почтовые ящики Gmail. Данные начнут отображаться при наличии достаточного объема данных (от ста записей) о ежедневном трафике.

Настройка пересылки FBL

Для того чтобы настроить FBL, отправитель должен добавить новый заголовок **Feedback-ID**, состоящий из параметров, уникально определяющих рассылку. Такие параметры называются **идентификаторами**.

Идентификаторы, сообщения с которыми часто попадают в спам или испытывают проблемы с доставкой, отображаются на панели FBL в Postmaster Tools.

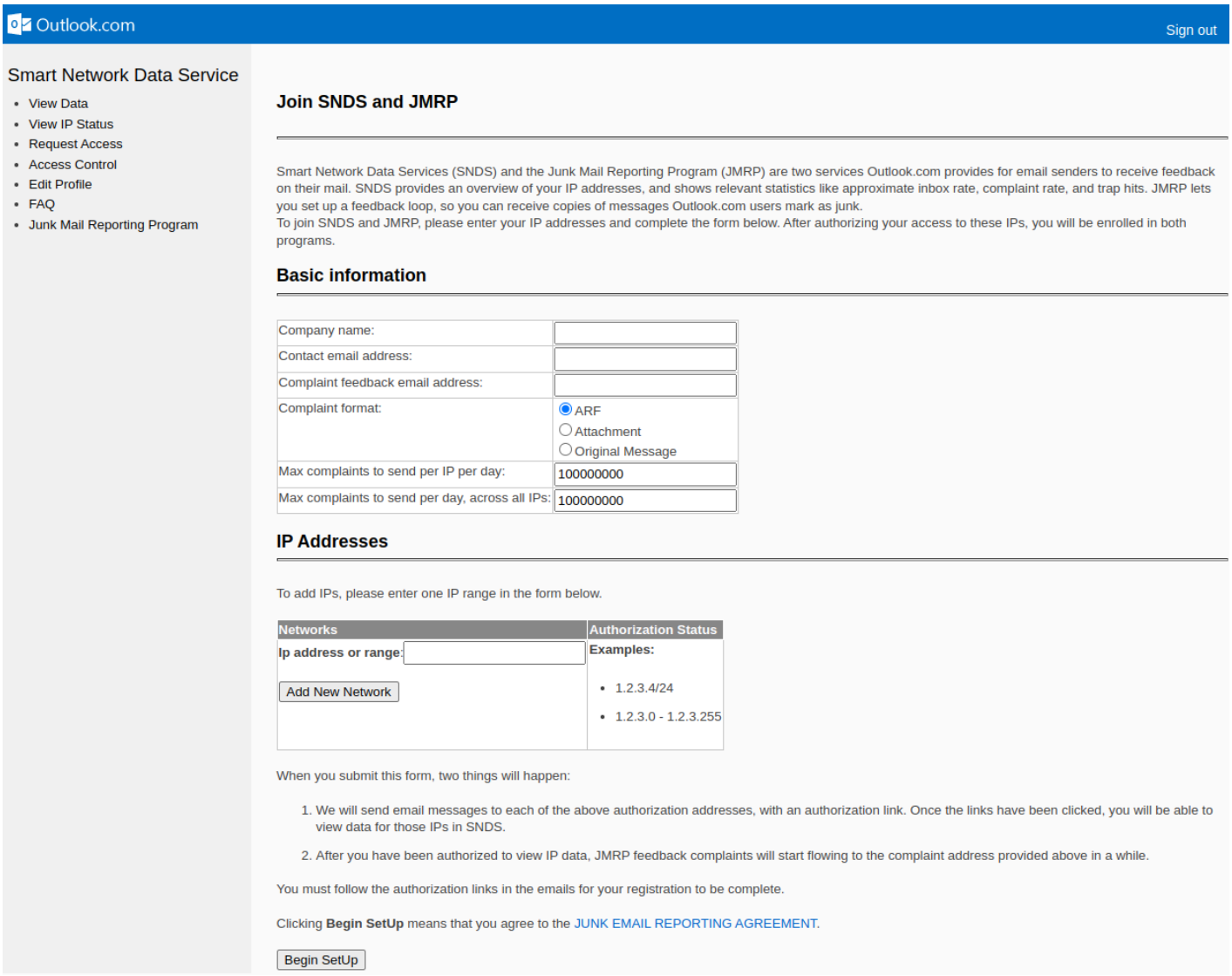
Формат заголовка — `Feedback-ID: a:b:c:SenderId`,

где `Feedback-ID:` – имя добавляемого заголовка, а `a, b, c:` – необязательные поля, с помощью которых отправитель может задать до трех идентификаторов (например, с названием компании или клиента). `SenderId` – выбранный отправителем обязательный уникальный идентификатор (5–15 символов). Он должен быть одинаковым во всех письмах.

Как подключить Outlook.com Postmaster

Шаг 1. Войдите в аккаунт Microsoft и перейдите к сервису.

Шаг 2. Заполните и отправьте форму на странице Junk Mail Reporting Program. Здесь необходимо ввести информацию о компании и все IP-адреса для рассылок:



Шаг 3. После ввода IP-адреса сервис предлагает выбрать почтовые ящики, на которые после отправки формы придёт письмо со ссылкой подтверждения. По этой ссылке будет необходимо перейти.

После выполнения всех требований на указанный почтовый ящик начнут приходить уведомления о жалобах на рассылки.

Службы Outlook.com Postmaster позволяют пользователю:

- получать отчеты о случаях доставки нежелательной почты, о которых сообщили пользователи Outlook.com
- распознать потенциальные проблемы, связанные с методами распространения и содержимым электронной почты
- улучшить репутацию отправителей путем удаления из списков нежелательных подписчиков
- предоставлять общие сведения об оценке пользователями получаемой электронной почты и надежности диапазона IP-адресов отправителей на основе показателей системы Outlook.com
- увидеть оценку вашей электронной почты спам-фильтрами
- отслеживать количество пользователей, жалующихся на вашу электронную почту

Как подключить Yahoo FBL

Yahoo предоставляет отправителям инструмент для отписки пользователей, пожаловавшихся на спам — **Yahoo Complaint Feedback Loop Service (YFBL)**.

Шаг 1. Для работы YFBL вам понадобится почтовый ящик вида **postmaster@(ваш_домен)**. Создайте его на сервисе, где настроена ваша доменная почта.

Шаг 2. Перейдите по ссылке и заполните обязательные поля:

- **Company name** — название организации;
- **Contact Name** — имя контактного лица;
- **Contact Email** — email контактного лица;
- **Reporting Email** — адрес для получения FBL-отчетов;
- **Selector** — укажите “*”;
- **Request type** — тип запроса: добавление, изменение или удаление. Укажите “Add”;
- **Domain** — домен, для которого настраиваете FBL. После заполнения обязательных полей нажмите кнопку “Get Verification Code”.

Шаг 3. Найдите в созданном почтовом ящике письмо от сервиса. Вставьте код подтверждения в соответствующее поле на странице сервиса YFBL и нажмите “Create Request”.

Обратите внимание: подтверждение настроек и включение отправки отчетов занимает до 48 часов. После успешной настройки сервис будет работать следующим образом: если получатель нажмет кнопку “Spam” в письме из вашей рассылки, то в ваш аккаунт придёт FBL-отчёт, а получатель будет автоматически отписан от рассылок.

Быстрый старт

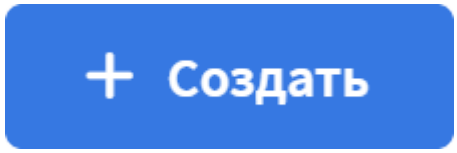
Как отправить первую email-рассылку

В этой статье показан базовый рабочий процесс в Altcraft Platform. Цель — отправить первое email-письмо и понять логику работы платформы.

Последовательность действий: **База профилей** — **Ресурс** — **Профиль** — **Шаблон** — **Рассылка** — **Отчеты**

Шаг 1. Создайте базу профилей

База профилей — это место хранения данных клиентов. В документации могут использоваться термины “клиент”, “подписчик”, “пользователь”, но во всех случаях имеется в виду **профиль**.

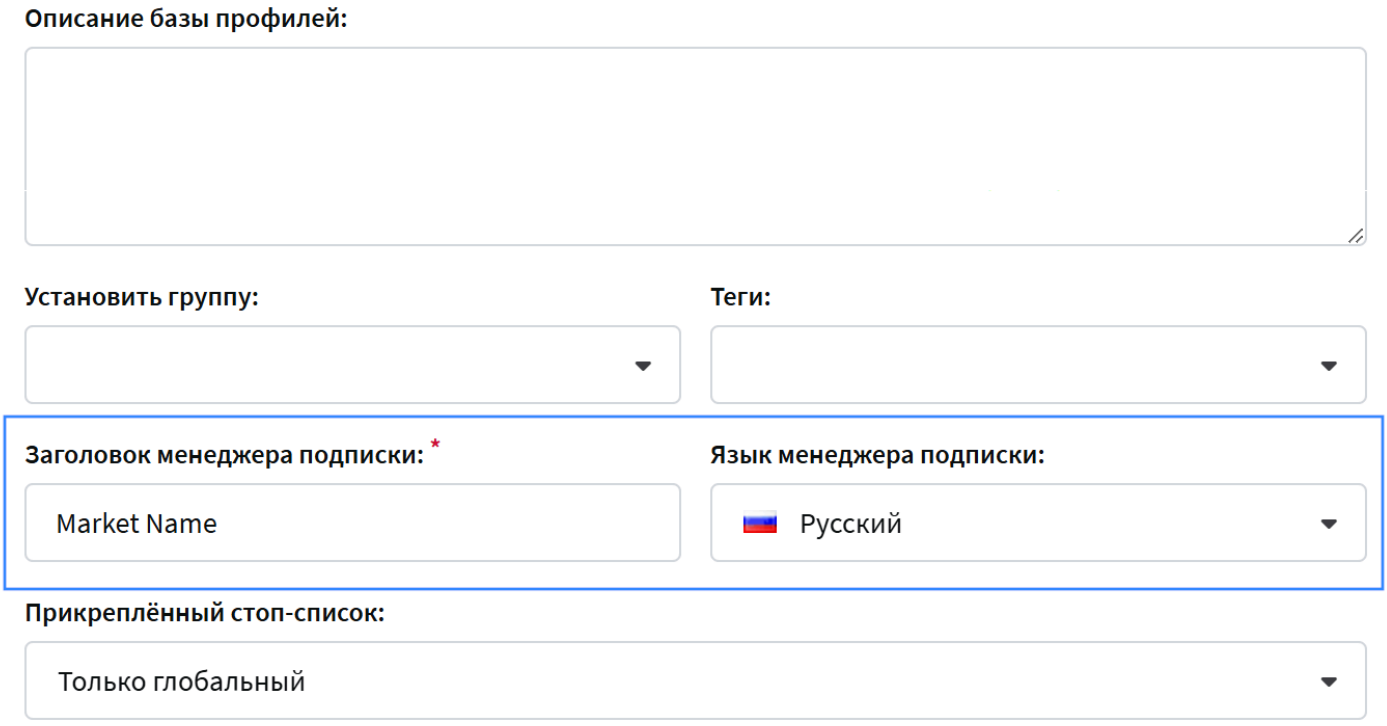
1. Перейдите в раздел **Данные — Базы профилей** и нажмите: 	

---

2. Укажите **Имя базы профилей**	

---

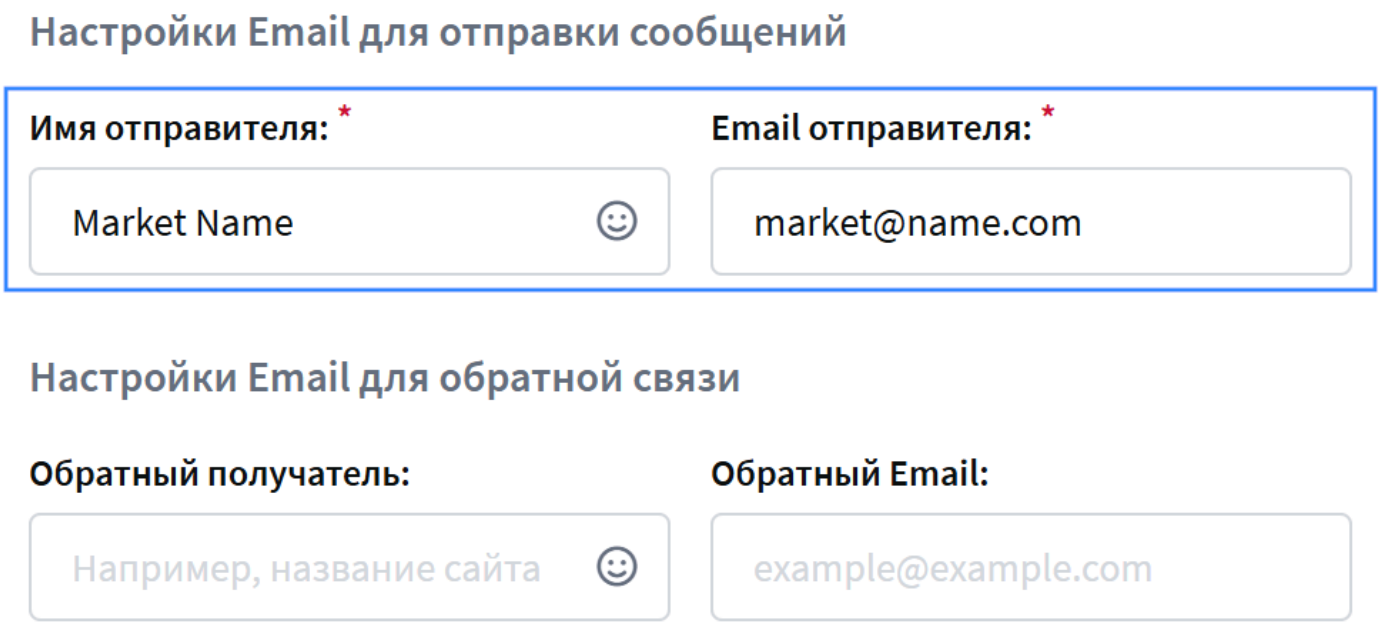
3. Задайте **Заголовок менеджера подписок**



Заголовок менеджера подписок — текст на странице управления подписками (например, название компании).

---

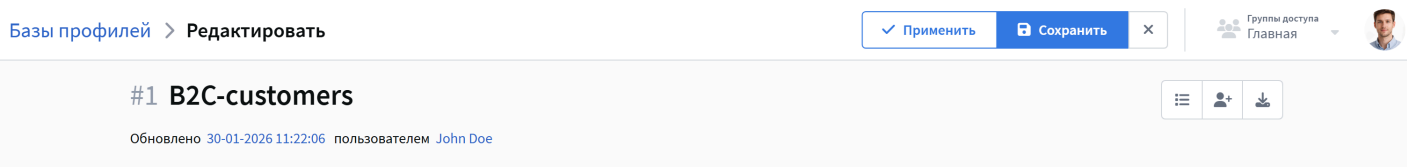
4. Укажите **Имя** и **Email отправителя**



Имя отправителя — имя, которое получатель видит в письмах
Email отправителя — адрес, с которого отправляются письма

---

5. Сохраните базу



Более подробную информацию по работе с базами профилей вы можете узнать в этой статье.

Шаг 2. Создайте ресурс

Ресурс — это источник коммуникаций: сайт, сервис, бренд или проект. Именно через ресурс оформляются подписки и отправляются сообщения. Без ресурса не получится создать рассылку.

Ресурс нельзя создать без привязки к базе профилей.

> **INFO**
>
> Для отправки email требуется настроенный “DKIM” для домена отправителя. Он не выдается автоматически. Для подключения обратитесь в чат технической поддержки.

1. Перейдите в раздел **Ресурсы**	

---

2. Выберите канал для подписки. **Для первой рассылки рекомендуем использовать Email**	

```
**3.** Укажите **Имя ресурса**

  ---
  **4.** Выберите созданную ранее **базу профилей** <br/><br/>

  ---

  **5.** Укажите **URL сайта** и **Tracking-домен**

  </td>
  <td width="55%px">
  </td>
</tr>
```

URL сайта / формы / приложения — адрес, где пользователи оставляют контакты. Можно указать основной сайт компании

Tracking-домен — технический домен для отслеживания переходов из писем. Если поле уже заполнено, оставьте значение без изменений

```
**6.** Укажите **Имя** и **Email отправителя**

  ---

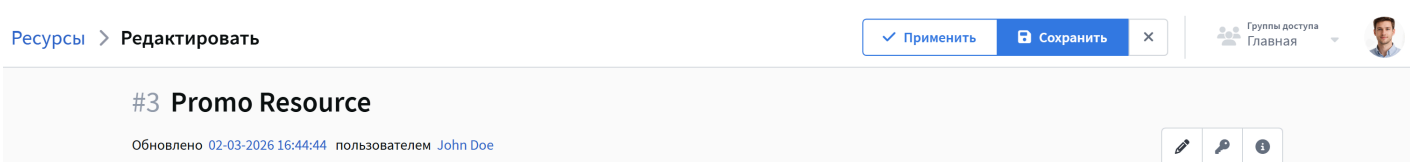
  **7.** Укажите **List ID**
  </td>
  <td width="65%px">
  </td>
</tr>
```

- **Имя отправителя** — имя, отображаемое в письмах - **Email отправителя** — адрес отправителя - **List ID** — идентификатор списка подписок внутри ресурса. Укажите произвольное понятное значение, можно использовать Email отправителя в формате `example.domain.com`

> **INFO**
>
> **Имя** и **Email отправителя** уже указывались в базе профилей. В настройках ресурса эти значения имеют приоритет.

---

8. Сохраните ресурс



Более подробную информацию по работе с ресурсами вы можете узнать в этой статье.

Шаг 3. Добавьте профиль

Создайте профиль, которому будет отправлено письмо. Для этого вернитесь в раздел **Данные** — **Базы профилей**.

1. Нажмите **Импорт профилей клиентов** — **Создать профиль вручную**

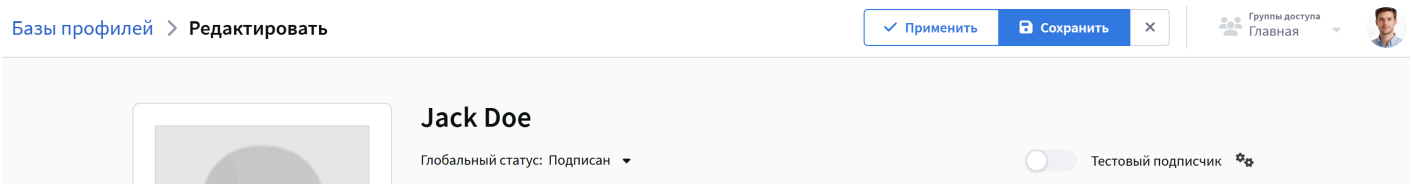
2. Укажите имя профиля и нажмите **Добавить подписку**	

3. Выберите созданный ресурс и канал Email	

4. Укажите email, на который будет отправлено сообщение	

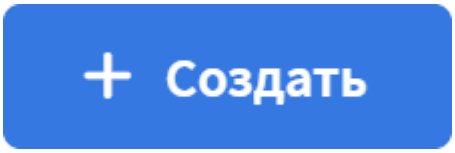
5. Сохраните профиль

Более подробную информацию о профилях вы можете узнать в этой статье.



Шаг 4. Создайте шаблон сообщения

Шаблон — это содержимое письма.

1. Перейдите в раздел **Шаблоны** и нажмите: 	

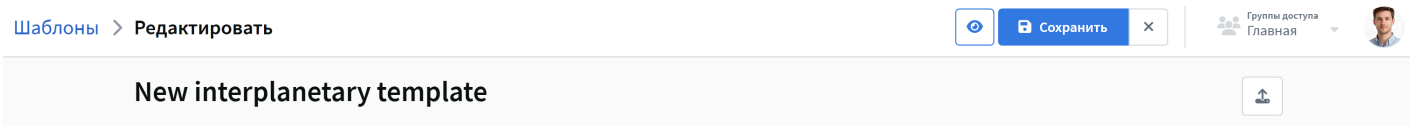
---

2. Выберите способ создания **Редактор** и канал **Email**

---

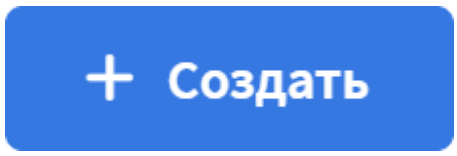
3. Откройте **Редактор кода**:

и **сохраните** шаблон:



В данном случае будет использован стандартный шаблон (заглушка) сообщения Альткрафт. Для настройки или импорта собственных шаблонов рекомендуем ознакомиться с разделом **Шаблоны сообщений**.

Шаг 5. Создайте рассылку

1. Перейдите в раздел **Рассылки — Броадкасты** и нажмите:  	

2. Введите **Имя рассылки** и выберите канал **Email**	

---

3. Настройте получателей

Перейдите на вкладку **Подписчики** и укажите:

- **Ресурс** — выберите созданный ранее ресурс
- **Источник подписчиков** — оставьте значение **База данных**
- **База профилей** — выберите базу, созданную в шаге 1

---

4. Выберите контент

На вкладке **Шаблоны** укажите созданный ранее **Шаблон сообщения**.

---

5. Настройте время отправки

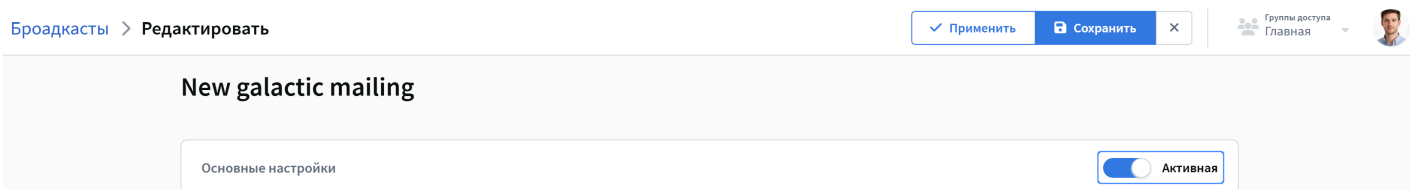
Перейдите на вкладку **Расписание**:

- Вы можете настроить конкретное время старта рассылки
- Чтобы отправить письмо сразу после сохранения, включите чекбокс **Запустить рассылку прямо сейчас (помимо расписания)**

---

6. Сохраните и запустите

Активируйте рассылку и нажмите кнопку **Сохранить**. После сохранения рассылка начнет отправку в соответствии с вашими настройками.



Письмо будет отправлено созданному тестовому профилю.

INFO

Более подробную информацию по отправке email-рассылок вы можете узнать в этой статье.

Шаг 6. Посмотрите результат

Проверить результат отправки можно несколькими способами: в блоке аналитики или в карточке профиля.

Способ 1. В блоке “Аналитика”

1. Выберите завершенную рассылку и кликните на кнопку

2. Изучите статистику

Способ 2. В профиле

1. Перейдите в раздел **Данные — Базы профилей**

1. Перейдите в раздел **Данные — Базы профилей**	

2. Откройте карточку созданного профиля и перейдите во вкладку **История** — там отображается вся информация о конкретном профиле:

INFO

Более подробную информацию об отчетах вы можете узнать в этой статье.

Поздравляем! Вы отправили свою первую email-рассылку в Altcraft Platform. Дальше предлагаем ознакомиться с тем как:

- **Создать HTML-шаблон** — заменить тестовый шаблон на свое письмо
- **Импортировать базу клиентов** — загрузить реальные контакты
- **Настроить сегменты** — отправлять письма только целевой аудитории
- **Создать триггерную рассылку** — автоматизировать коммуникацию

Push-канал

Push-уведомления

Push-уведомления — это мощный инструмент для мгновенного взаимодействия с аудиторией. Короткие персонализированные сообщения, отправляемые прямо на экраны смартфонов или компьютеров, позволяют эффективно напоминать об акциях, новых продуктах или важных событиях, даже если пользователь не активен в приложении или на сайте.

Mobile Push

Mobile Push-уведомления — это короткие сообщения, которые появляются на экране мобильного устройства, даже если приложение не активно. Благодаря возможности персонализации и таргетирования, такие уведомления помогают удерживать внимание пользователей, повышать вовлеченность и стимулировать повторное взаимодействие с приложением.

Web Push

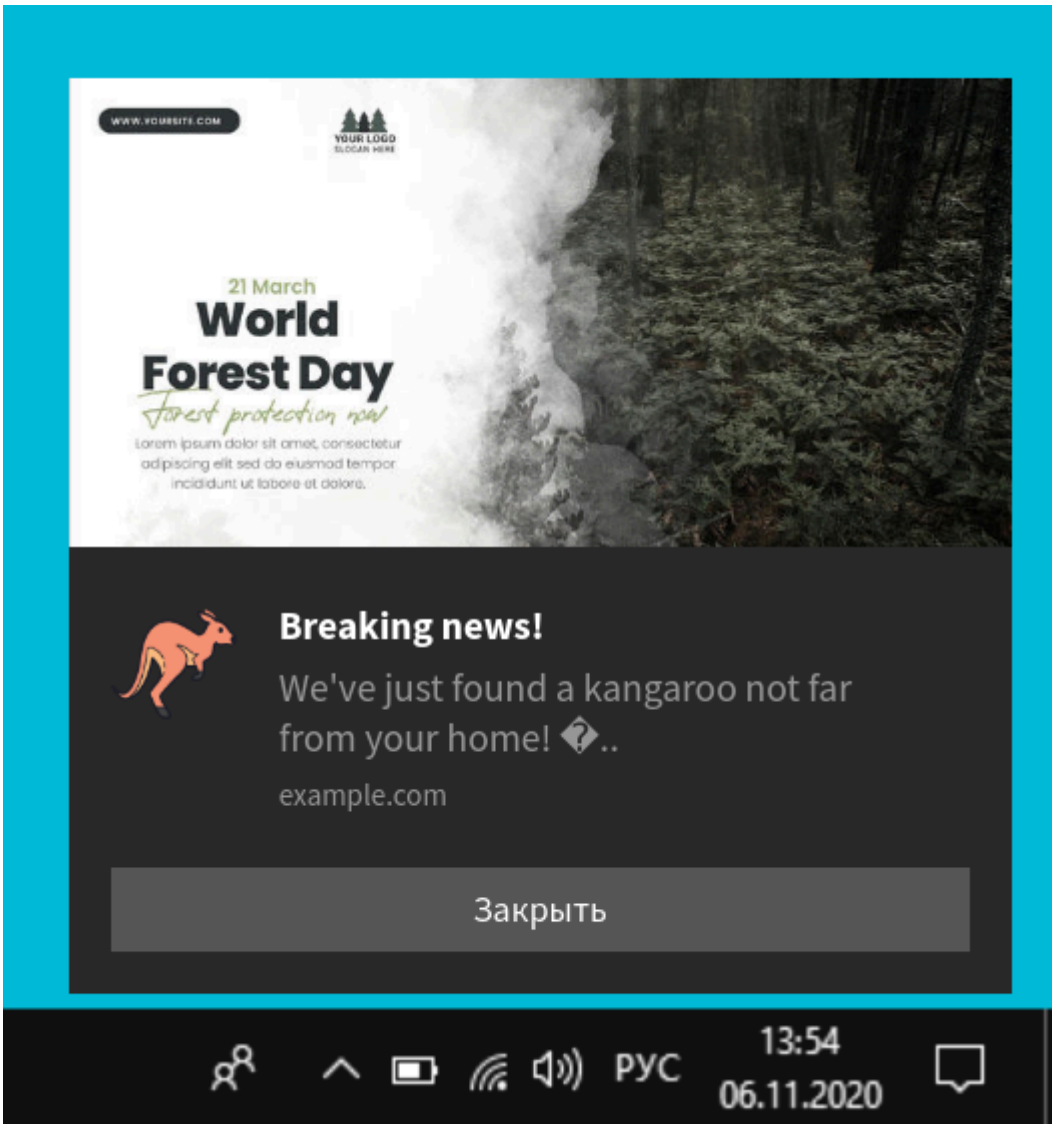
Web Push-уведомления — это сообщения, которые появляются на экране компьютера или мобильного устройства через веб-браузер, даже если сайт не открыт. Они работают через технологии браузеров (например, Chrome, Safari, Firefox) и не требуют установки приложения. Основное отличие от мобильных Push — Web Push не зависят от наличия приложения. Web Push уведомления могут приходить как в десктопные, так и в мобильные браузеры, что делает их универсальным инструментом для взаимодействия с аудиторией.

Платформа Altcraft также поддерживает **отправку push-уведомлений от PWA-приложений**.

Rich Push

Платформа Altcraft поддерживает создание **Rich Push-уведомлений.** Они предоставляют уникальные преимущества за счет использования визуально насыщенного контента. Это позволяет не просто информировать пользователя, но и вовлекать его в интерактивное взаимодействие, повышая конверсию и удерживая внимание. Вы можете добавить в Rich push-уведомление:

- картинку (баннер), видео или аудио;
- кнопки и внешние ссылки.



Чтобы создать шаблон для подобных уведомлений, воспользуйтесь **редактором для Rich Push**.

Mobile Push

Mobile Push: настройка и подключение

TIP

Настройка **браузерных push-уведомлений** описана в соответствующей статье.

	Mobile Push — это короткие сообщения, которые появляются на экране мобильного устройства, даже если приложение не активно. В отличие от Web Push, **Mobile Push** уведомления приходят не в браузер, а в мобильное приложение

Push-уведомление может содержать следующие данные:

- **Заголовок** — аналогичен теме Email сообщения;
- **Тело** — любой текст для сообщения;
- **Иконка** — изображение в уведомлении;
- **Ссылка для перехода** — перенаправление при нажатии на push-уведомление.

Для отправки таких уведомлений используются следующие провайдеры:

Сервис	Поддерживаемые ОС
Google Firebase Cloud Messaging	Android, iOS
Apple Push Notification Service	iOS
Yandex.AppMetrica	Android, iOS
Huawei Mobile Services	Android, iOS
RuStore	Android

Как настроить и подключить Mobile Push-уведомления

Настройка и подключение

Mobile push: настройка и подключение

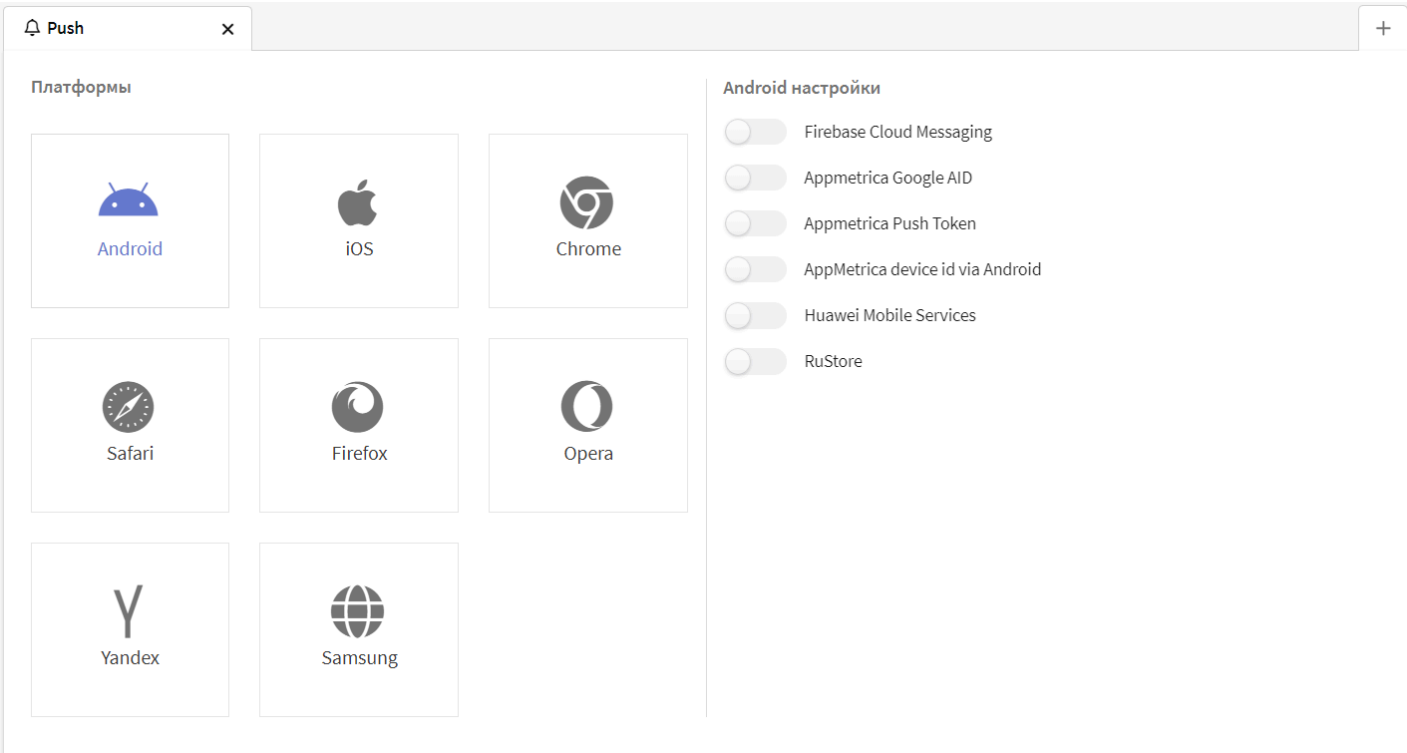
TIP

Настройка **браузерных Push-уведомлений** описана в соответствующей статье.

Мобильные push-уведомления отображаются по мере получения в центре уведомлений мобильного устройства — в “шторке” Android и iOS смартфонов, планшетов и других гаджетов. По умолчанию уведомления включены для устанавливаемых приложений, так что вы можете сразу с момента установки начать процесс вовлечения клиента.

Для мобильных приложений есть 5 вариантов настройки:

- **Google Firebase Cloud Messaging** — для Android и iOS приложений
- **Apple Push Notification Service** — только для iOS приложений
- **Yandex.AppMetrica** — для Android и iOS приложений
- **Huawei Mobile Services** — для Android и iOS приложений
- **RuStore** — только для Android приложений



CAUTION

Yandex.AppMetrica использует для отправки SDK Google Firebase. Для отправки уведомлений вам нужно будет установить его в приложение.

TIP

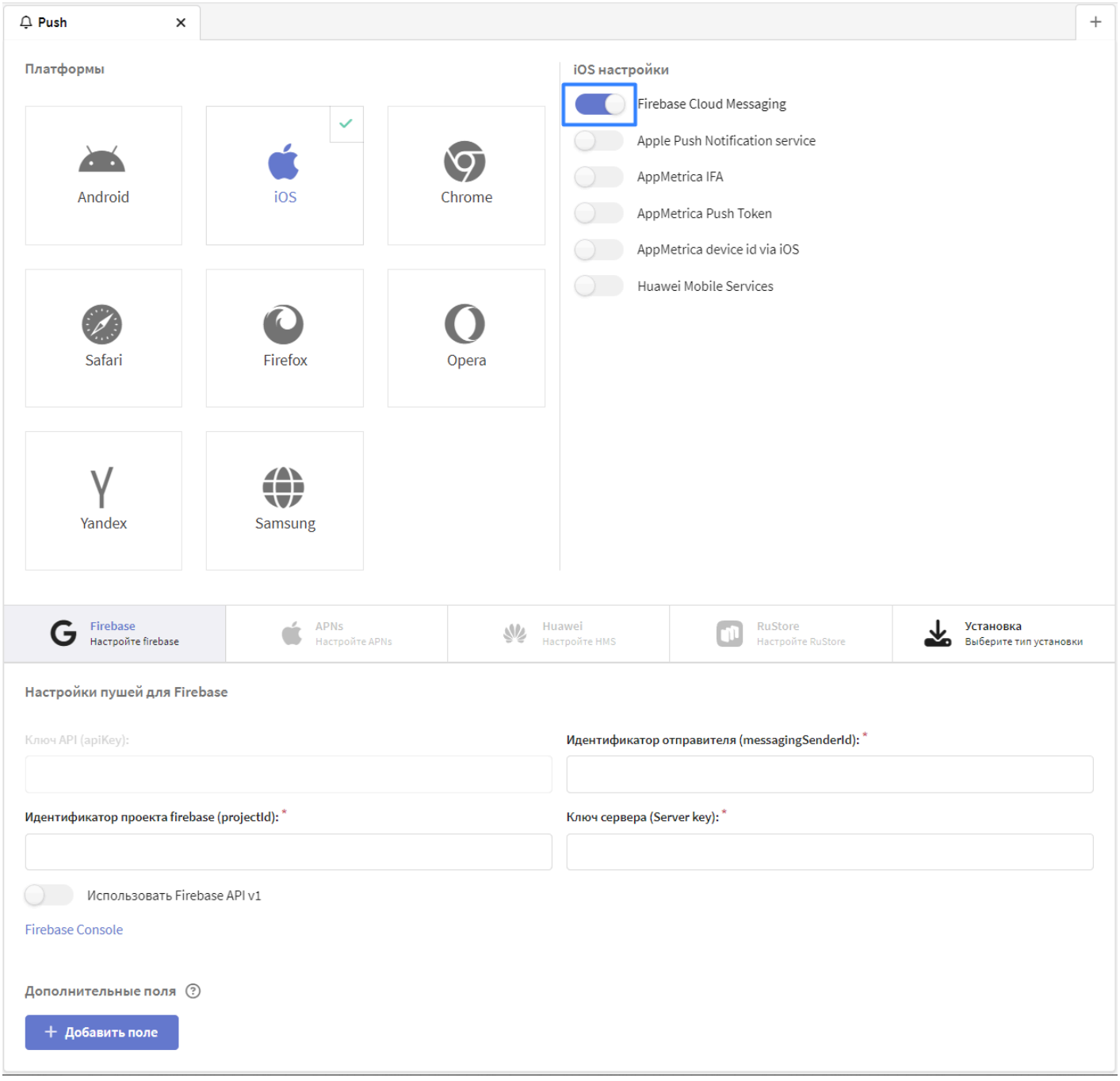
В Altcraft Marketing доступна интеграция с Yandex.AppMetrica для импорта профилей пользователей, регистрации их действий и связанной с ними ценности (стоимости).

Настройка App Push в мобильных приложениях Android & iOS

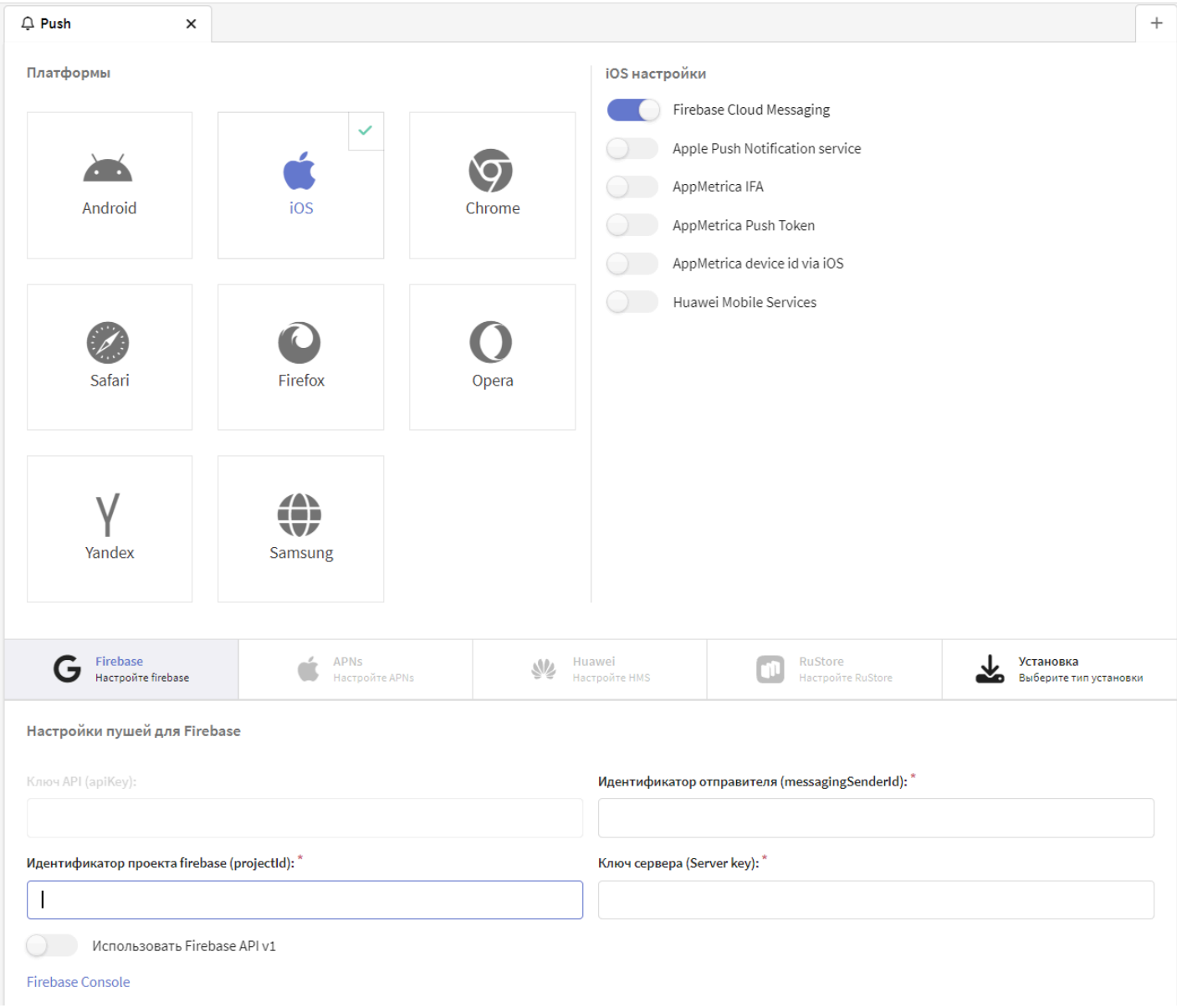
Firebase Cloud messaging

Firebase	Для отправки Push мобильных приложений на устройства с Android и iOS с помощью платформы **Google Firebase** необходимо предварительно настроить проект в Firebase Console.

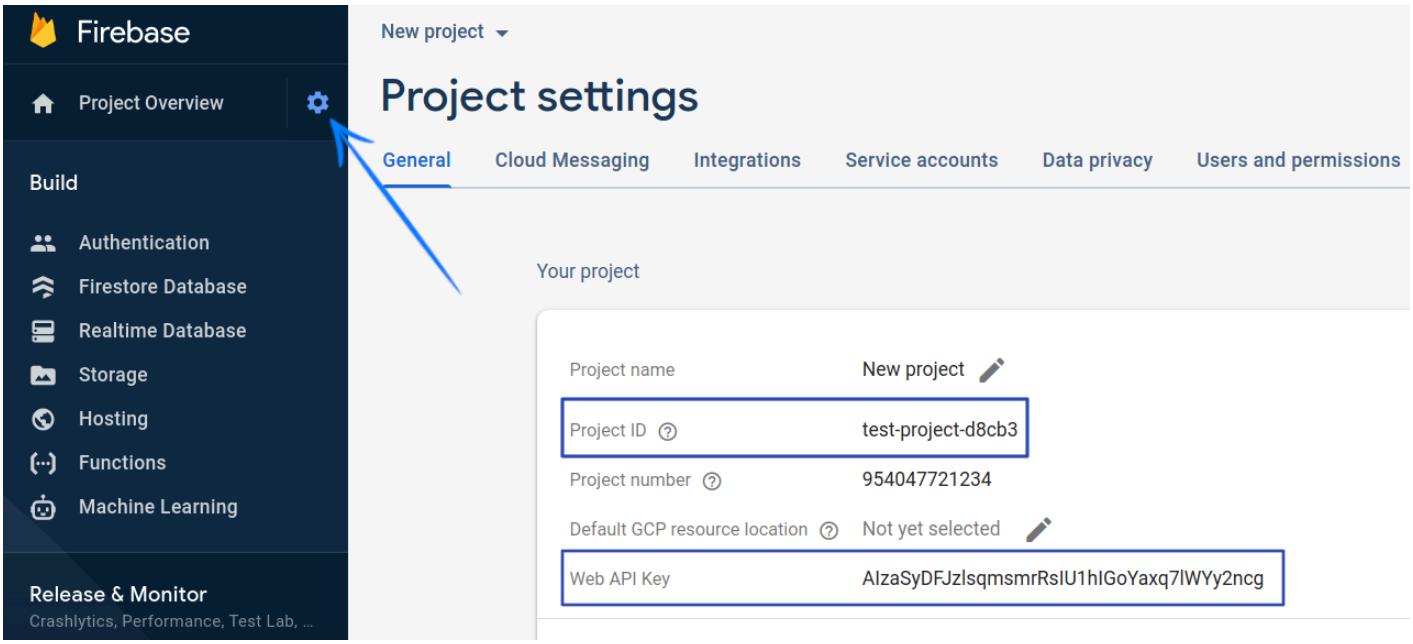
Создайте учётную запись в Google Firebase и зарегистрируйте свой сайт или приложение:



Откройте настройки ресурса и активируйте Firebase Cloud Messaging для ваших Android и iOS приложений. В открывшейся снизу вкладке введите настройки проекта Firebase:

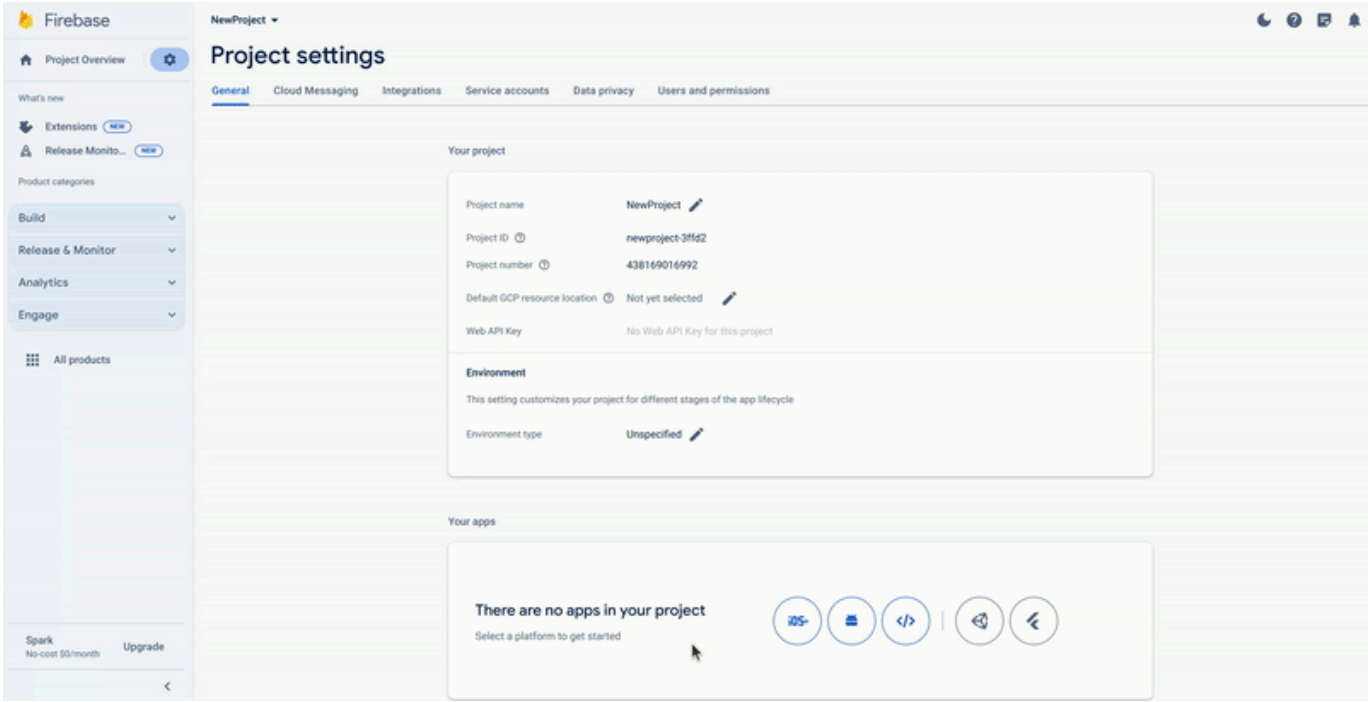


Ключ API для веб-приложения (Web API Key) и **Идентификатор проекта firebase** (Project ID) вы можете найти в настройках Firebase console в разделе общие (General):

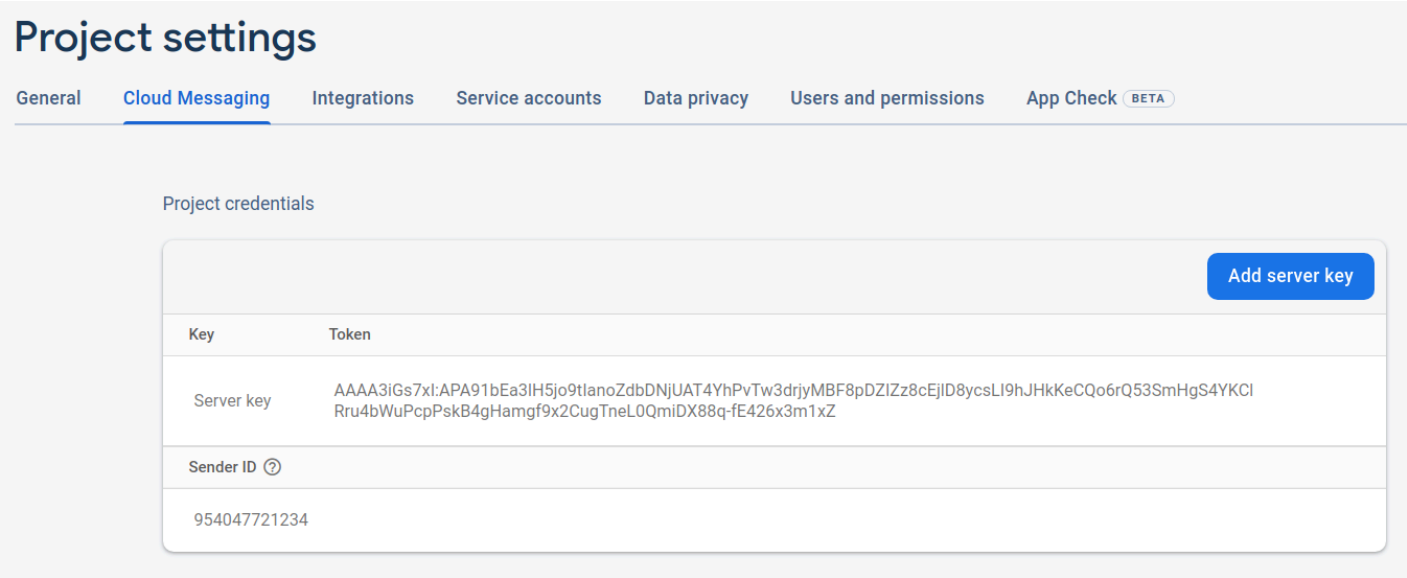


TIP: Web API Key

Если у вас не отображается Web API Key, перейдите в раздел **Build** и выберите пункт **Authentication** в выпадающем меню. Нажмите на кнопку **Get started**, которая появится на экране. Теперь вы можете вернуться в настройки, нажав на соответствующую иконку в меню слева.



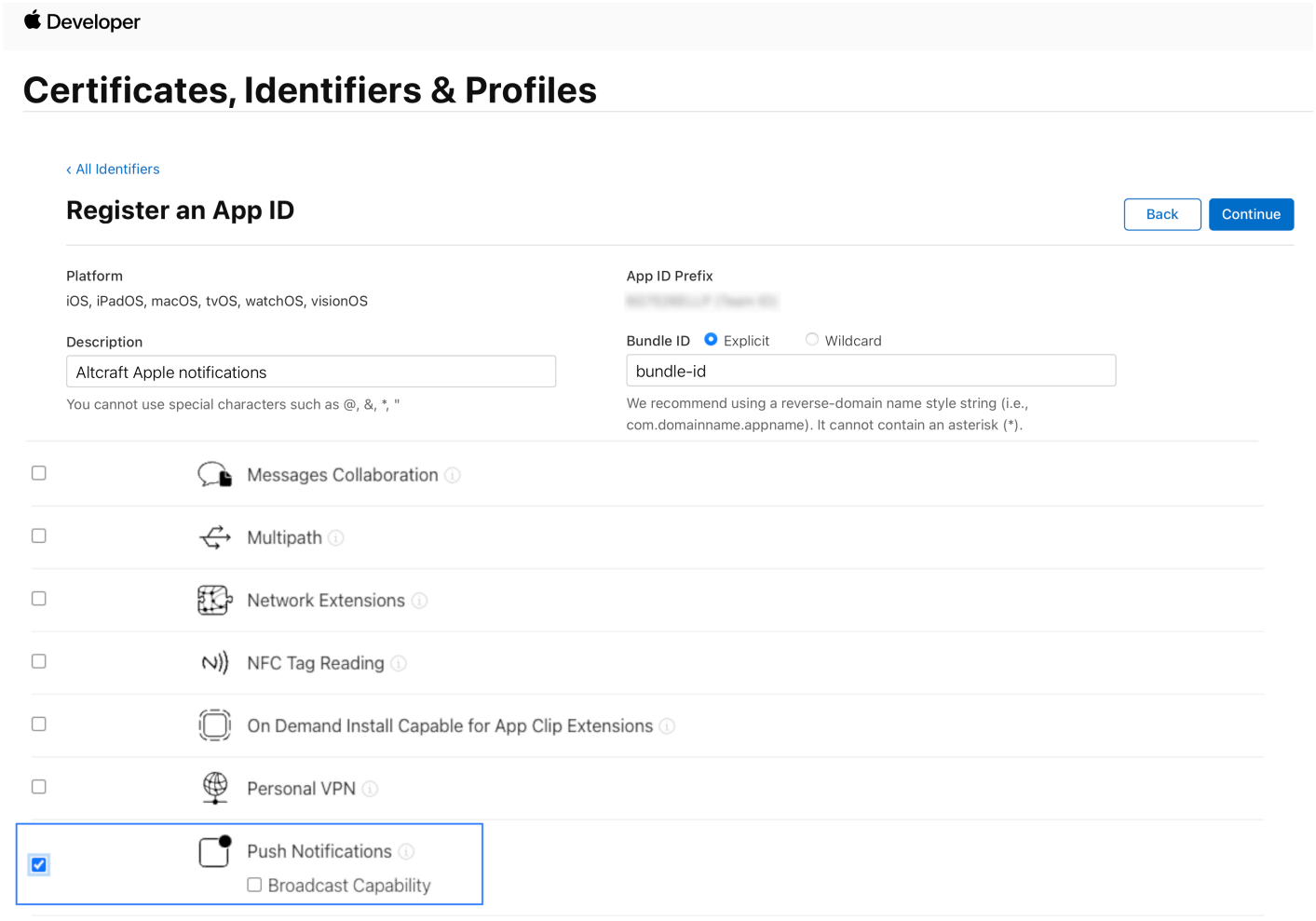
Ключ сервера (Server key) и **Идентификатор отправителя** (Sender ID) находится в настройках Firebase console в разделе Cloud Messaging:



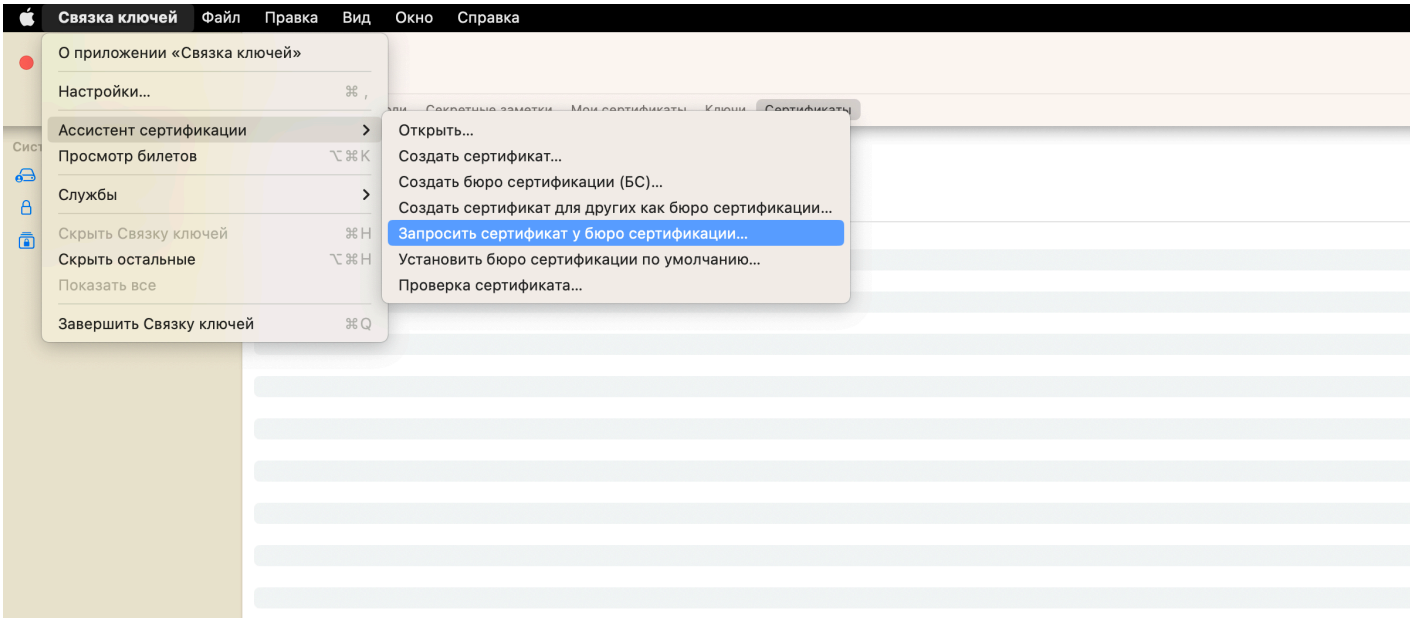
Apple Push Notification service

	Apple Push Notification service позволяет отправлять push-уведомления мобильных приложений на устройства с iOS. Для настройки сервиса необходим аккаунт Apple Developer и устройство на macOS.

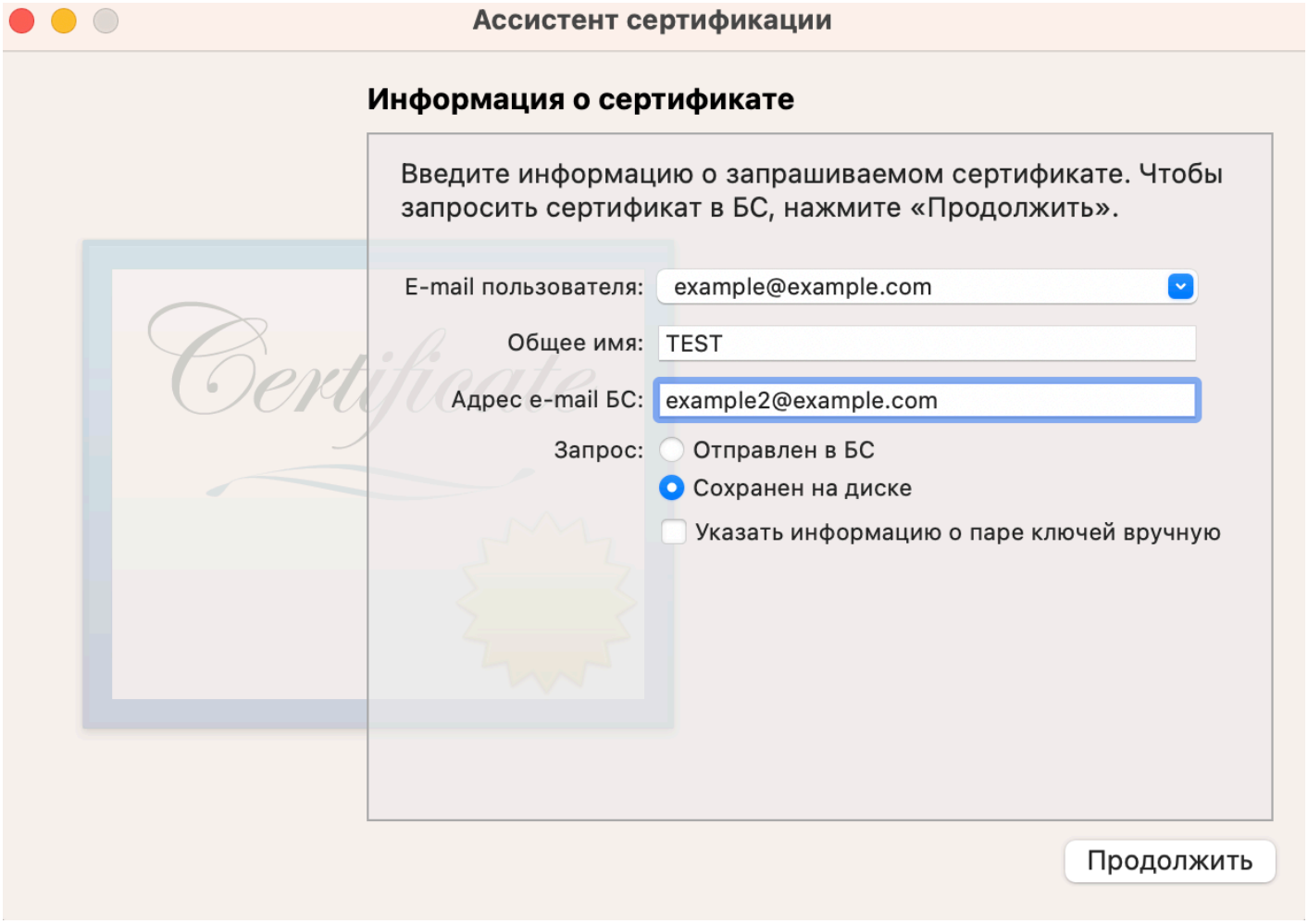
Перейдите на сайт Apple Developer в раздел **“Account”** — **“Certificates, Identifiers & Profiles”** — **“Identifiers”**. Создайте новый идентификатор. Выберите опцию **App IDs**, затем **App**. Введите в поле **Description** название, укажите **Bundle ID** приложения в соответствующем поле, выберите опцию **Explicit**, затем в списке ниже выберите пункт **Push Notifications** и зарегистрируйте ID:



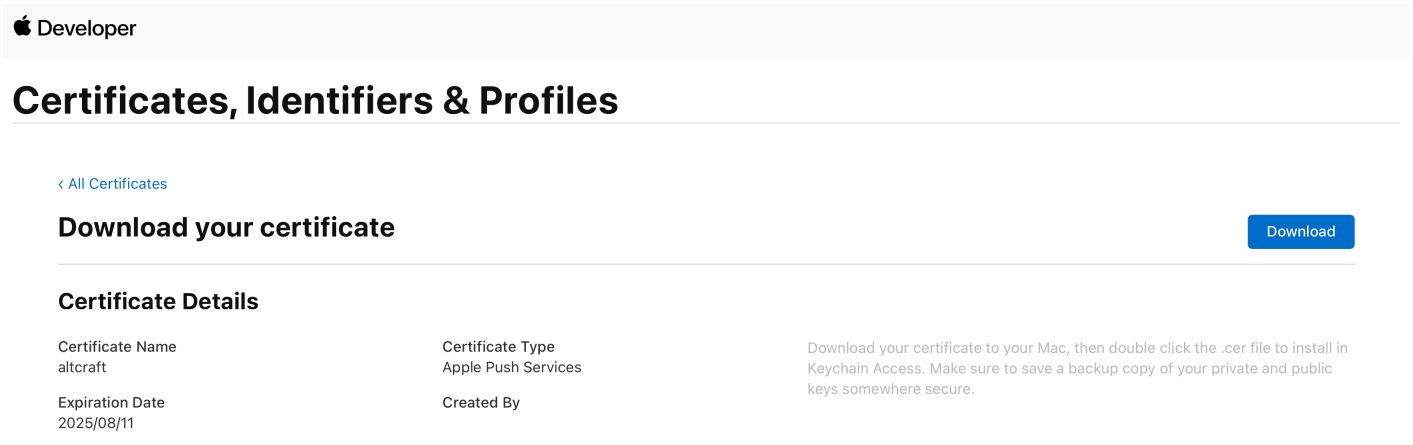
Откройте на устройстве приложение **“Связка ключей”**. Перейдите в раздел **“Связки ключей по умолчанию”** — **“Вход”**, в меню выберите **“Ассистент сертификации”** — **“Запросить сертификат у бюро сертификации”**:



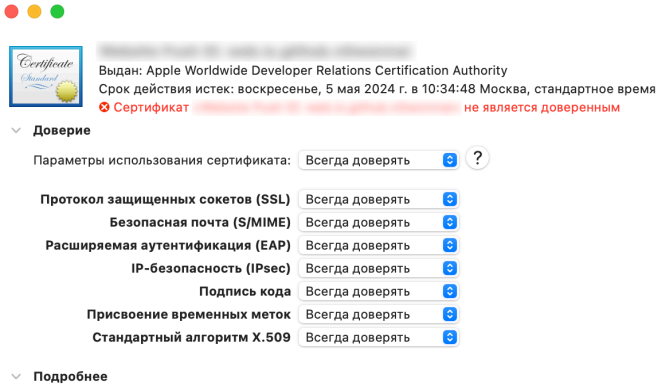
В открывшемся окне заполните необходимые поля и выберите пункт **“Сохранен на диске”**. Сохраните сертификат на устройстве:



Вернитесь в раздел **“Certificates, Identifiers & Profiles”** — **“Certificates”** на сайте. Создайте новый сертификат. В списке Services выберите **Apple Push Notification service SSL (Sandbox & Production)**. В следующем пункте найдите ваше приложение в выпадающем списке **AppID**. Загрузите созданный раннее сертификат в соответствующее поле. Получившийся сертификат Apple сохраните на устройстве:



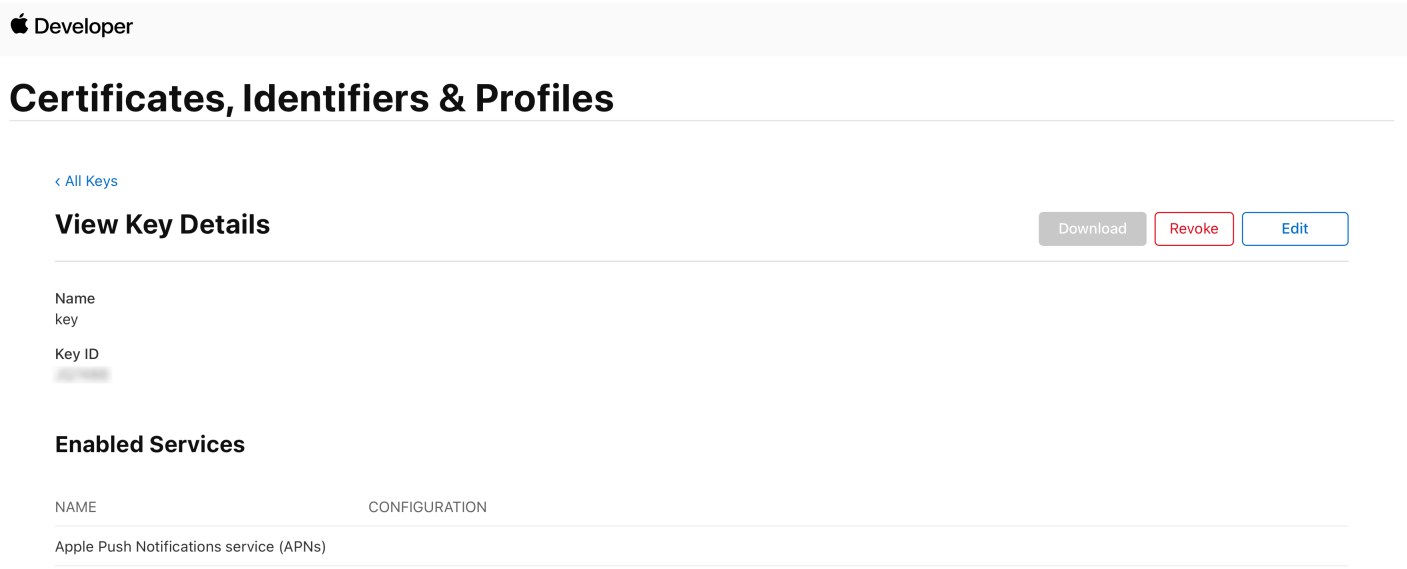
Дважды нажмите на него, чтобы он появился в приложении “Связка ключей”. Откройте его и выберите в параметрах использования сертификата опцию **“Всегда доверять”**:



Дальнейшие действия зависят от того, какой тип аутентификации вы будете использовать: аутентификация с помощью **токена** или с помощью **сертификата**.

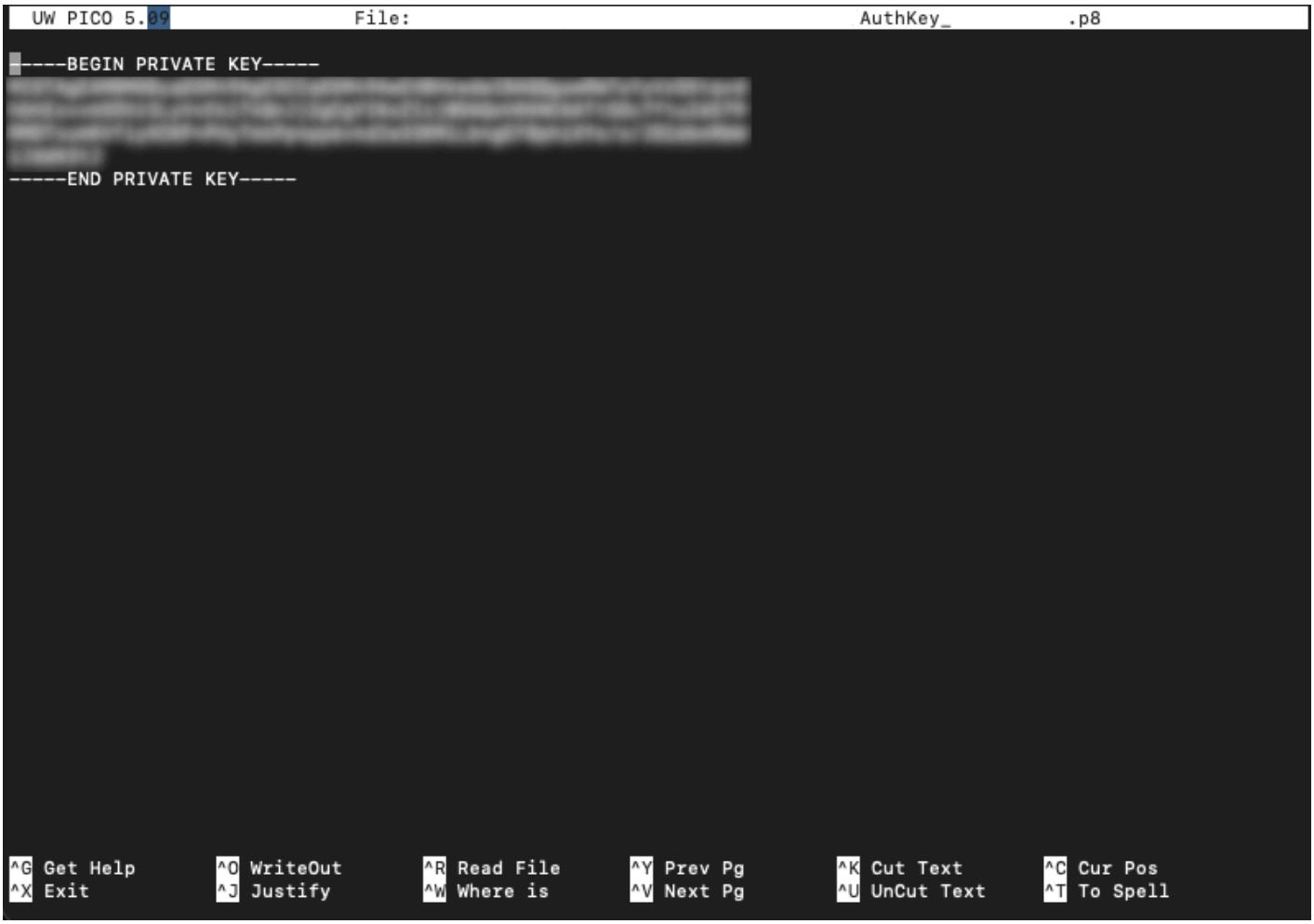
Аутентификация с токеном

На сайте Apple Developer перейдите в раздел **“Certificates, Identifiers & Profiles”** — **“Keys”**. Создайте новый ключ. Укажите название ключа и его тип — **APNs**, нажмите **“Continue”**, затем **“Register”**. Запишите **Key ID**, он будет необходим для конфигурации ресурса на платформе:

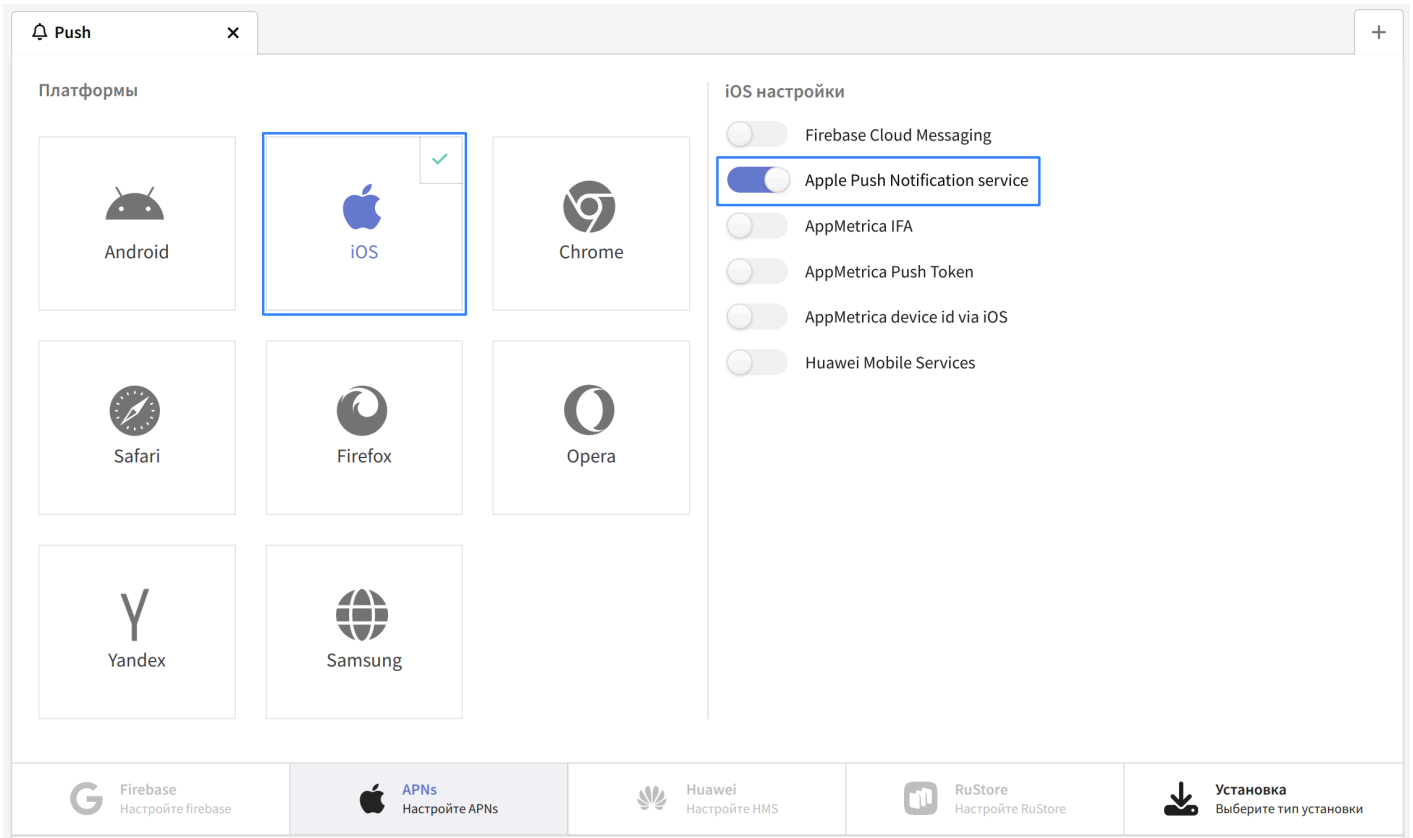


Скачайте ключ к себе на устройство. Его необходимо открыть с помощью любого текстового редактора. Внутри содержится **ключ подписи**. Его нужно будет указать в соответствующем поле при настройке ресурса.

> **CAUTION**
>
> Ключ можно скачать лишь **один раз**. Если вы не можете его скачать сейчас, нажмите **“Done”** и вернитесь, когда такая возможность появится.

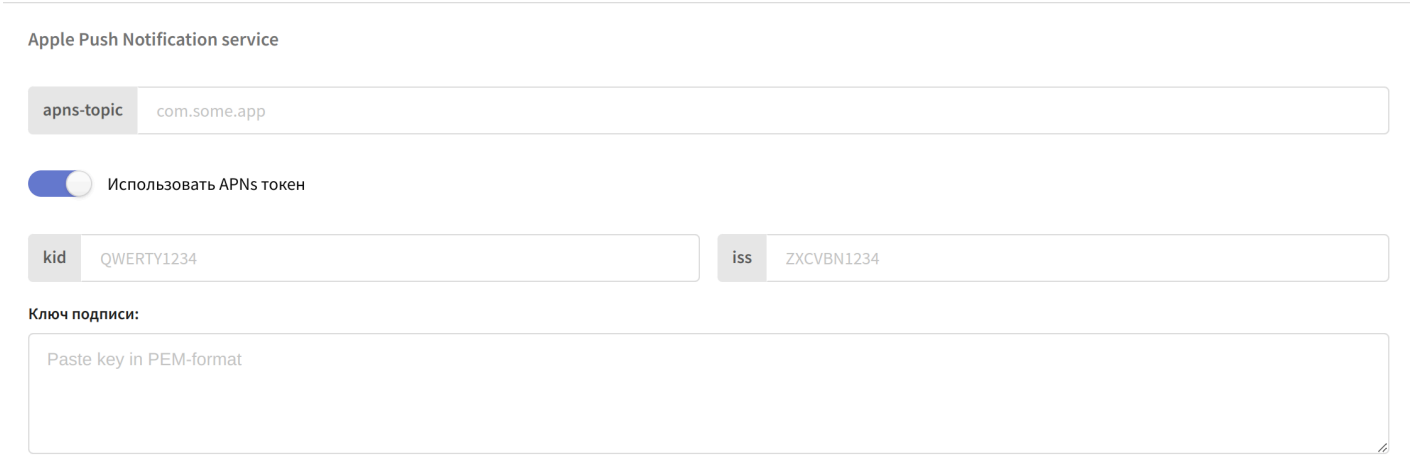


На платформе Altcraft создайте **ресурс** для push-канала, выберите платформу **iOS** и включите переключатель **Apple Push Notification service**:



Активируйте переключатель **“Использовать APNs токен”** в настройках ниже. Заполните следующие поля:

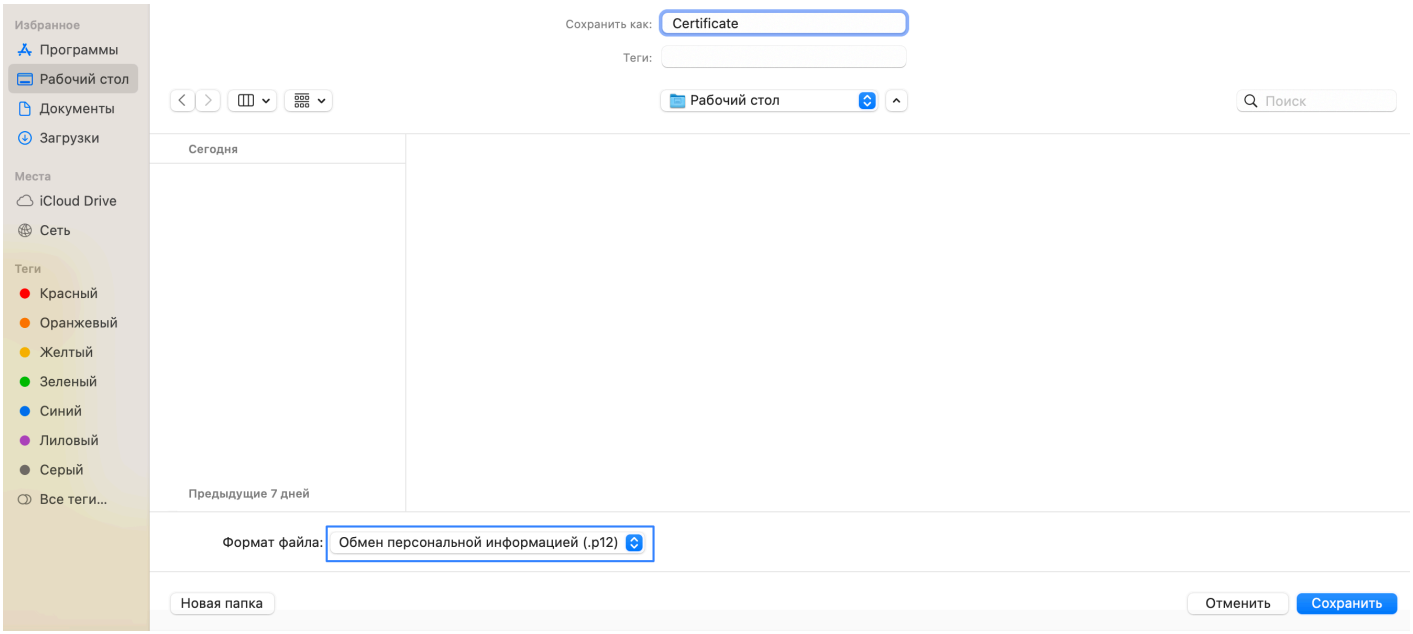
- **apns-topic** — Bundle ID приложения.
- **kid** — значение keyID ключа.
- **iss** — Team ID, обычно указан в аккаунте разработчика на сайте Apple Developer.
- **Ключ подписи** — содержимое файла-ключа, скачанного раннее.



Сохраните настроенный ресурс с помощью кнопок в верхней части страницы. Платформа готова к отправке уведомлений с помощью APNs.

Аутентификация с сертификатом

В приложении “Связка ключей” нажмите правой кнопкой мыши на созданный сертификат и экспортируйте его. Выберите формат .p12. При необходимости установите для сертификата пароль:

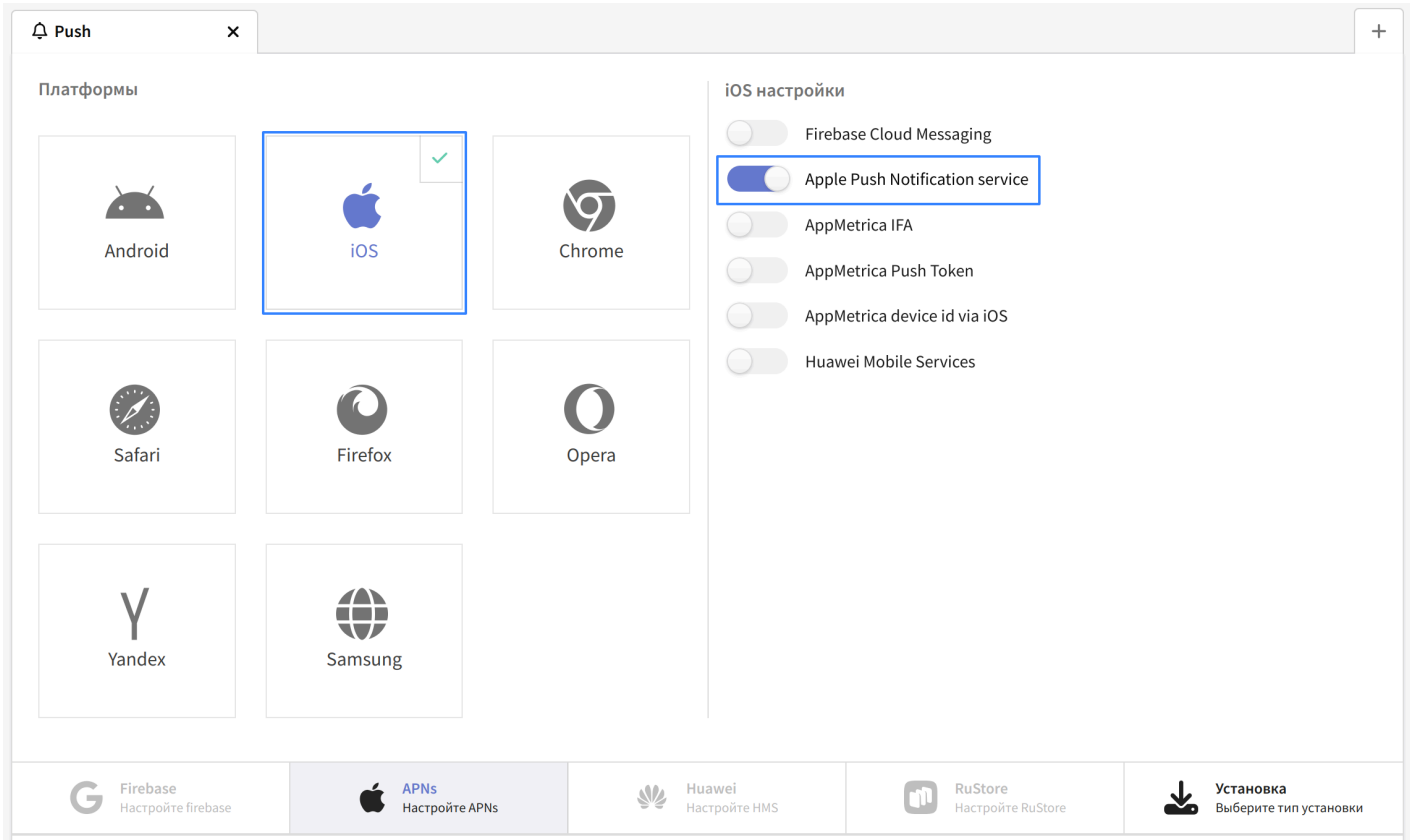


Используя **терминал** перейдите в папку, в которую вы экспортировали сертификат. Создайте **pem-ключ** с помощью команды `openssl pkcs12 -in filename.p12 -out filename.key.pem -nocerts -nodes`, где `filename.p12` — имя экспортированного файла, а `filename.key.pem` — название создаваемого pem-файла с ключом.

Не закрывая терминал и не переходя в другие папки, создайте **pem-сертификат** с помощью команды `openssl pkcs12 -in filename.p12 -out filename.crt.pem -clcerts -nokeys` по аналогии с командой выше (`filename.p12` — исходный файл, `filename.crt.pem` — конечный файл).

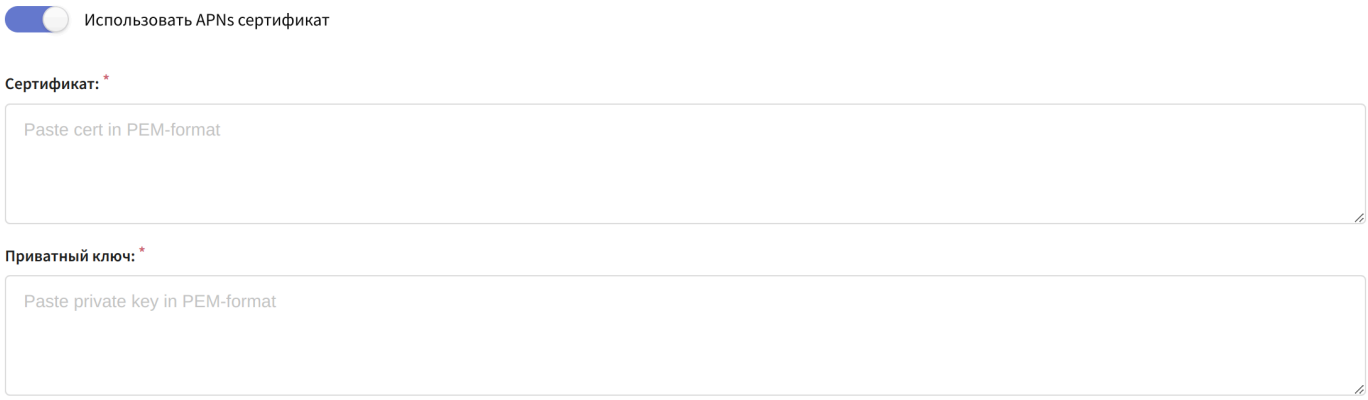
Откройте данные файлы с помощью **редактора nano** и скопируйте значение ключа и сертификата:

На платформе Altcraft создайте **ресурс** для push-канала, выберите платформу **iOS** и включите переключатель **Apple Push Notification service**:



Активируйте переключатель **“Использовать APNs сертификат”** в настройках ниже. Заполните следующие поля:

- **Сертификат** — значение созданного вами pem-сертификата.
- **Приватный ключ** — значение созданного вами pem-ключа.



Сохраните настроенный ресурс с помощью кнопок в верхней части страницы. Платформа готова к отправке уведомлений с помощью APNs.

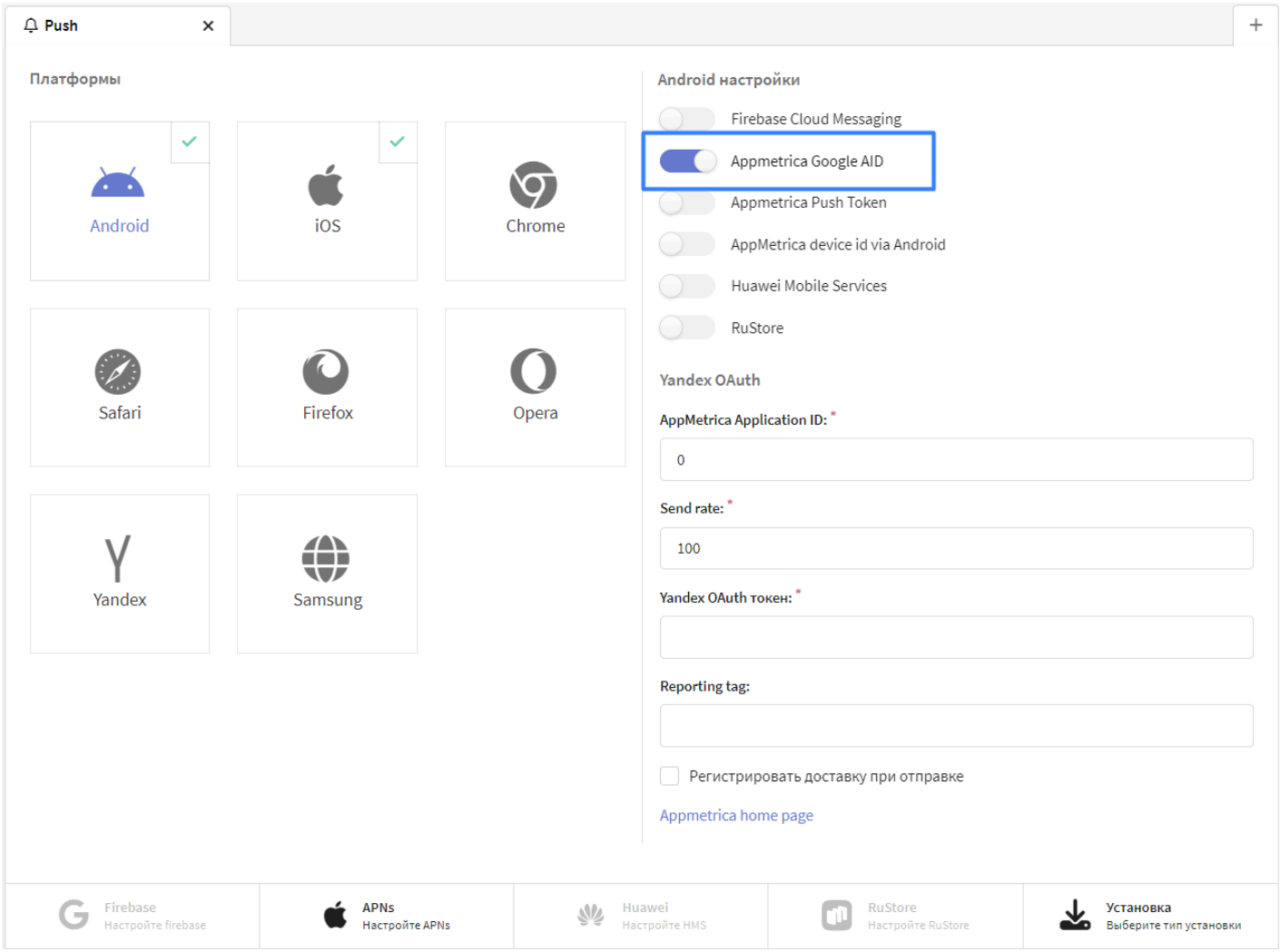
Yandex.AppMetrica push SDK

AppMetrica	Аналитическая платформа Yandex.AppMetrica может использоваться для отправки пуш-уведомлений на **Android** , так и на **iOS** устройства. Для этого Push SDK метрики должен быть установлен в приложение. Ссылка на документацию по подключению SDK — Документация Yandex AppMetrica.

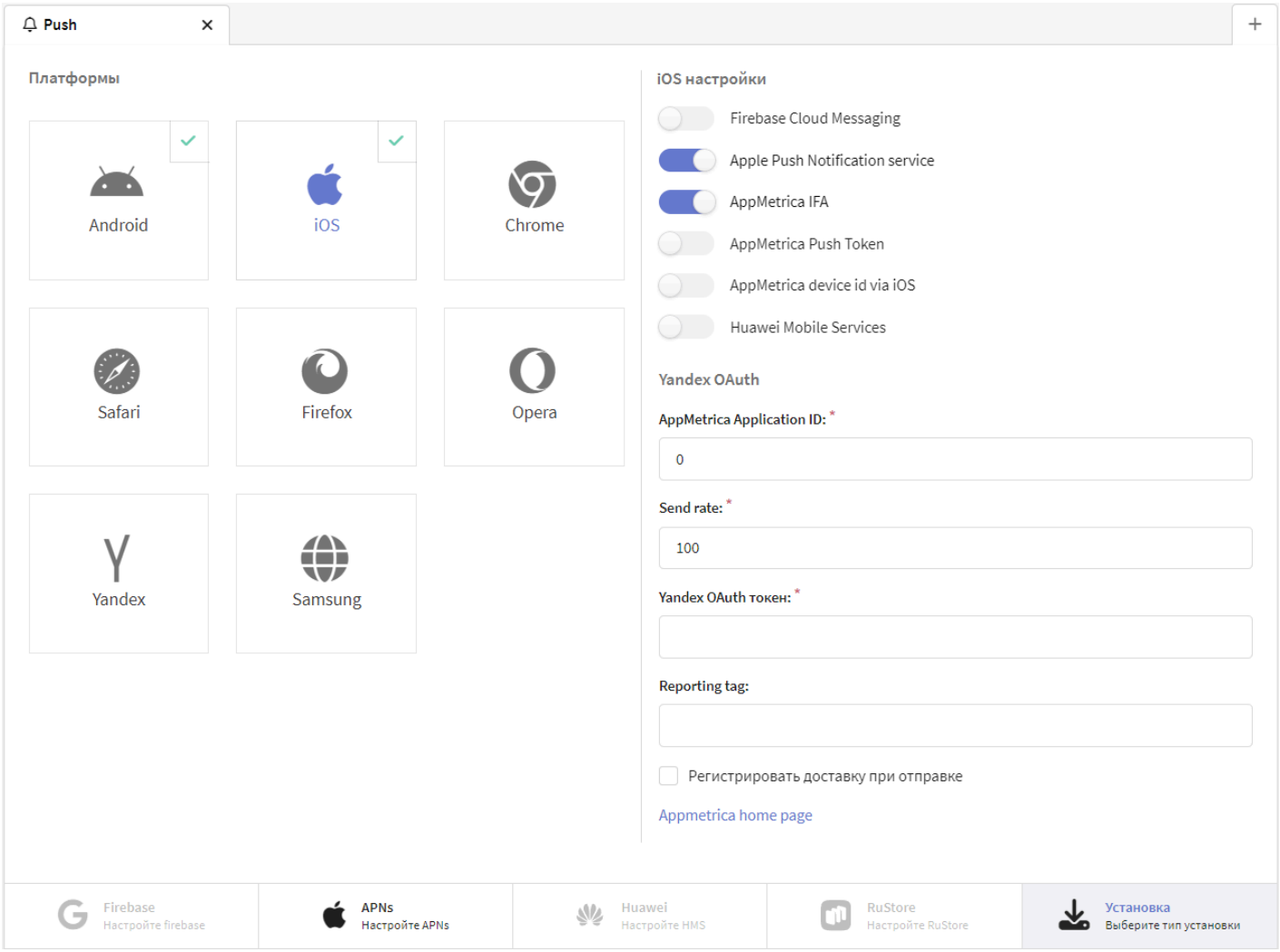
В качестве источника подписки может выступать один из следующих уникальных идентификаторов устройств:

- **AppMetrica device ID** — идентификатор устройства в AppMetrica (для обеих операционных систем).
- **iOS IFA** — рекламный идентификатор (только для Apple устройств).
- **Google AID** — рекламный идентификатор (только для Android).
- **Android Push token** и **iOS Push Token** — токены, назначаемые AppMetrica SDK.

В рамках одного ресурса можно настроить отправку приложений в обе операционные системы. На вкладке **Push** в ресурсе выберите мобильную платформу и внесите настройки в правой части вкладки:



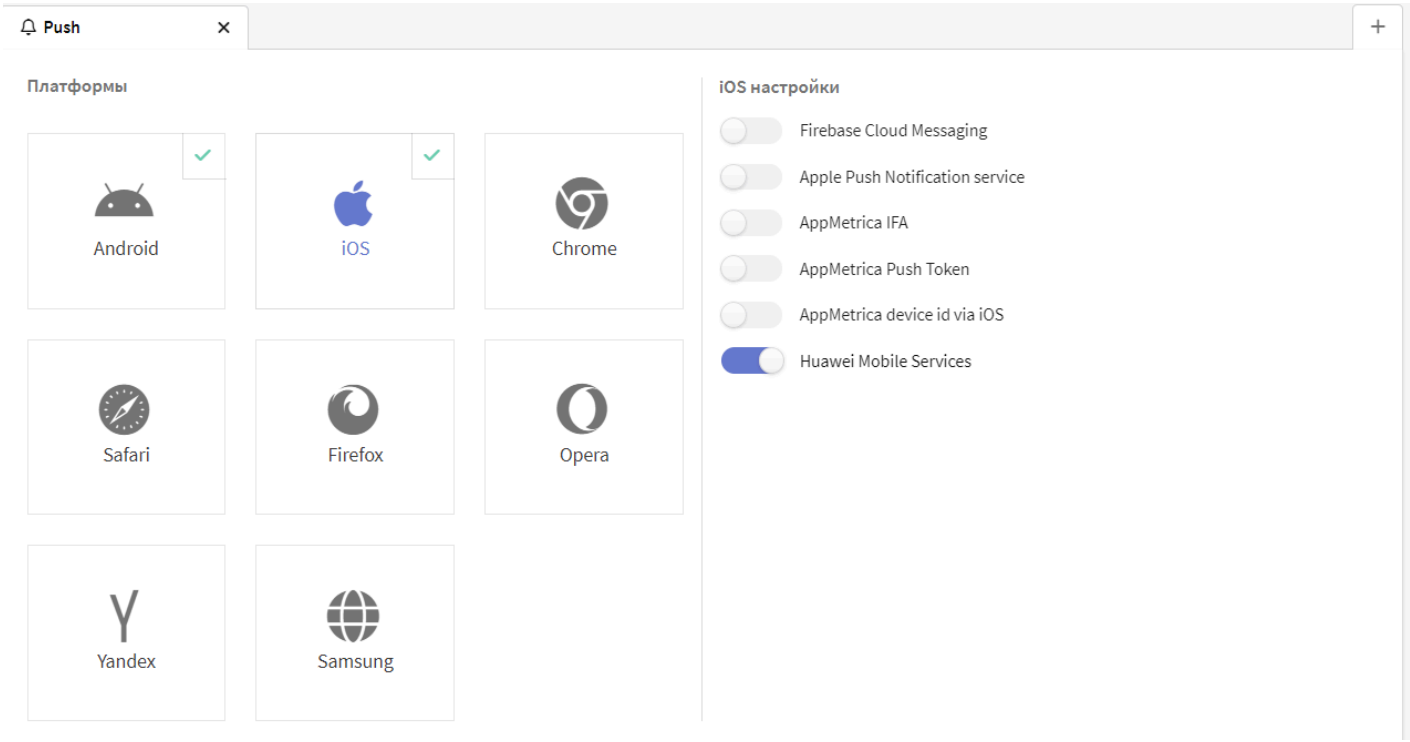
- Для доступа к AppMetrica используется **AppMetrica API-токен**. Для каждого приложения указывается уникальный **AppMetrica Application ID**. Получить эти данные можно в разделе **Настройки** приложения в AppMetrica.
- Параметр **Send rate** ограничивает максимальную скорость отправки уведомлений в секунду
- **Yandex OAuth токен** — токен авторизации в сервисах Яндекс.
- В поле **Reporting tag** введите произвольный тег, который будет использоваться в отчётах AppMetrica.



Huawei Mobile Services

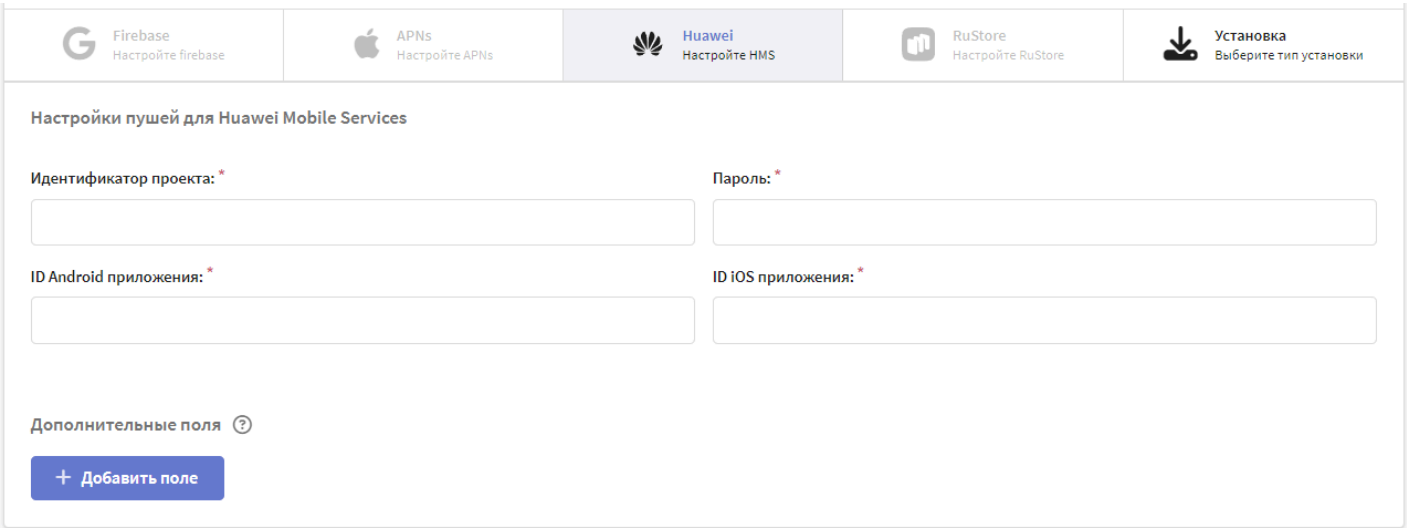
HUAWEI	Huawei Mobile Services позволяет отправлять push-уведомления из приложений, поддерживаемых Huawei, на Android и iOS устройства.

Откройте настройки ресурса и активируйте **Huawei Mobile Services** для ваших **Android** и **iOS** приложений:



На панели внизу выберите **Huawei**:

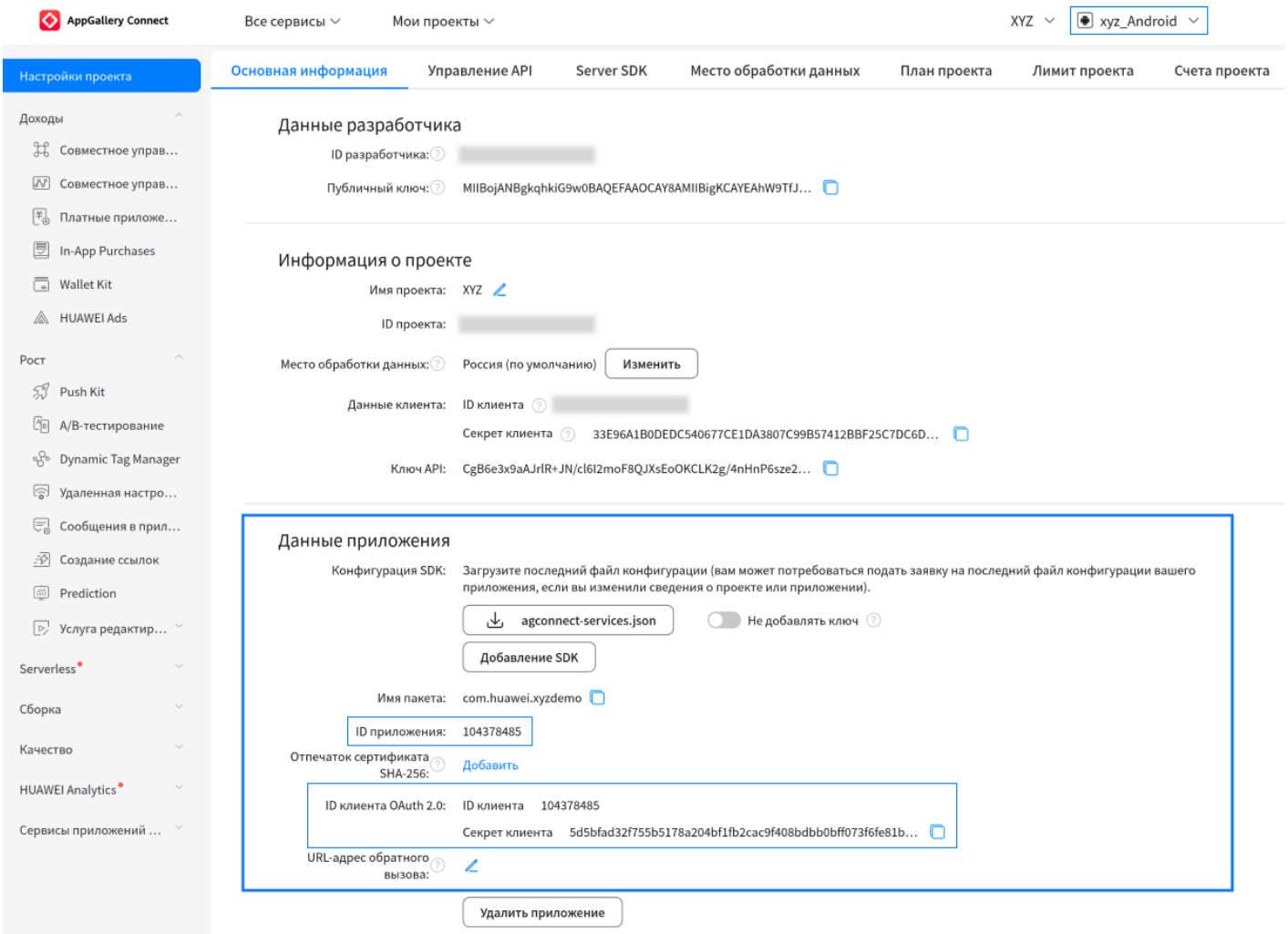
Введите данные для настройки:



Идентификатор проекта, пароль и ID Android или iOS приложений вы можете найти в настройках своего проекта. Для этого в аккаунте разработчика Huawei откройте консоль и перейдите на страницу AppGallery Connect —> **Мои проекты**. На панели сверху в выпадающем списке выберите приложение, для которого хотите настроить отправку push-уведомлений.

Используйте данные приложения:

- **App ID = ID клиента = Идентификатор проекта**
- **Секрет клиента = Пароль**
- **ID приложения = ID Android/iOS приложения**



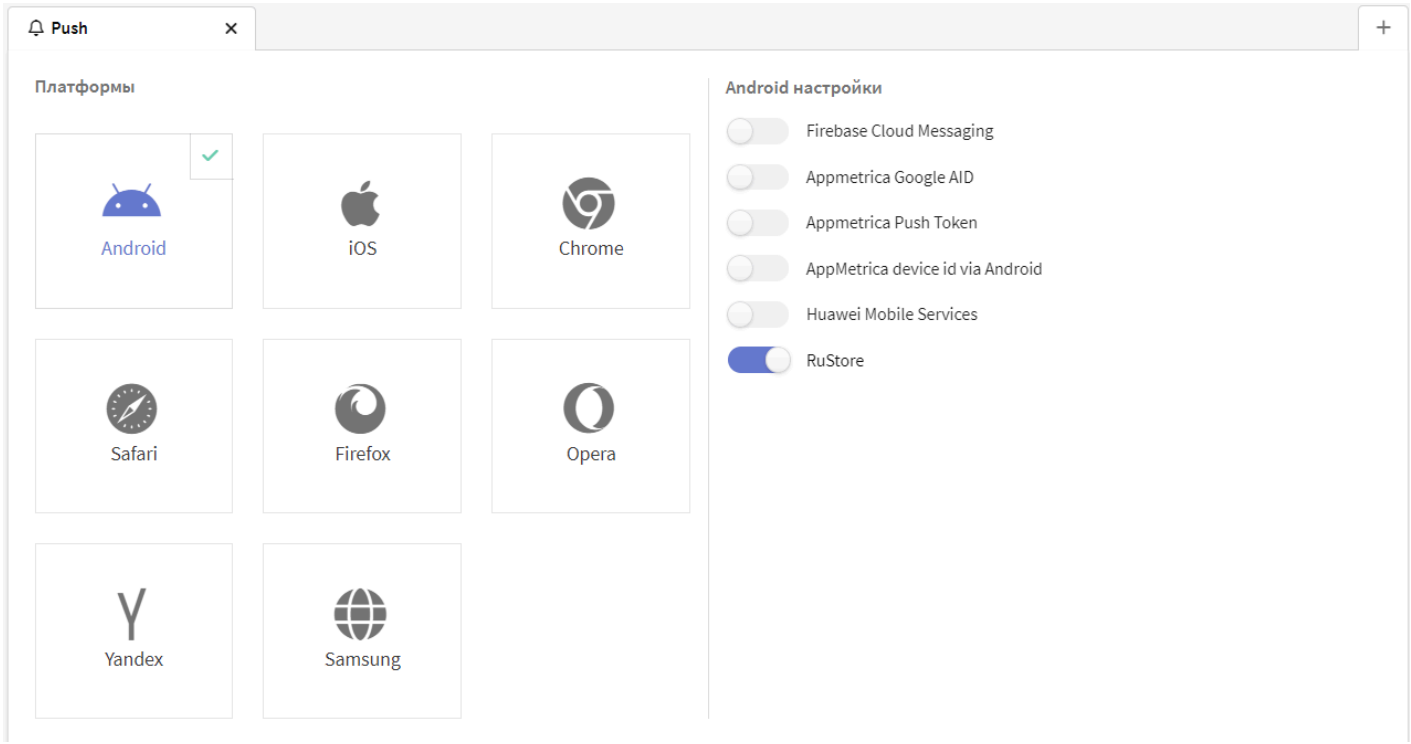
RuStore

RuStore	Сервис RuStore позволяет отправлять push-уведомления из приложений, размещенных в магазине RuStore. Предварительно необходимо создать проект push-уведомлений для вашего приложения.

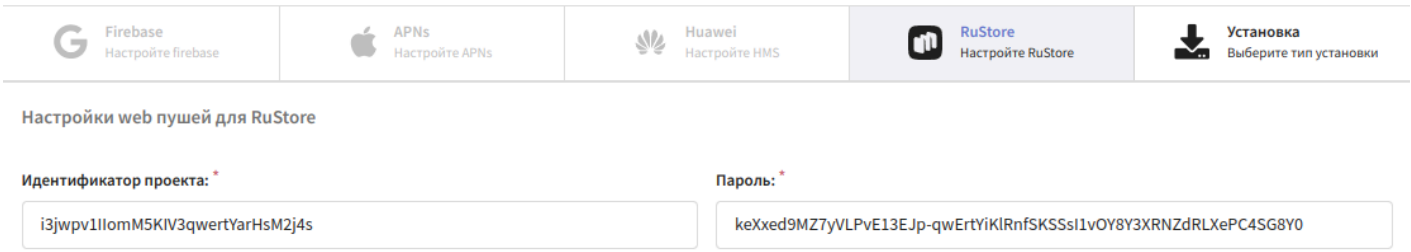
TIP

Для публикации приложения в магазине RuStore следуйте официальной документации.

Откройте настройки ресурса и активируйте RuStore для Android приложений:

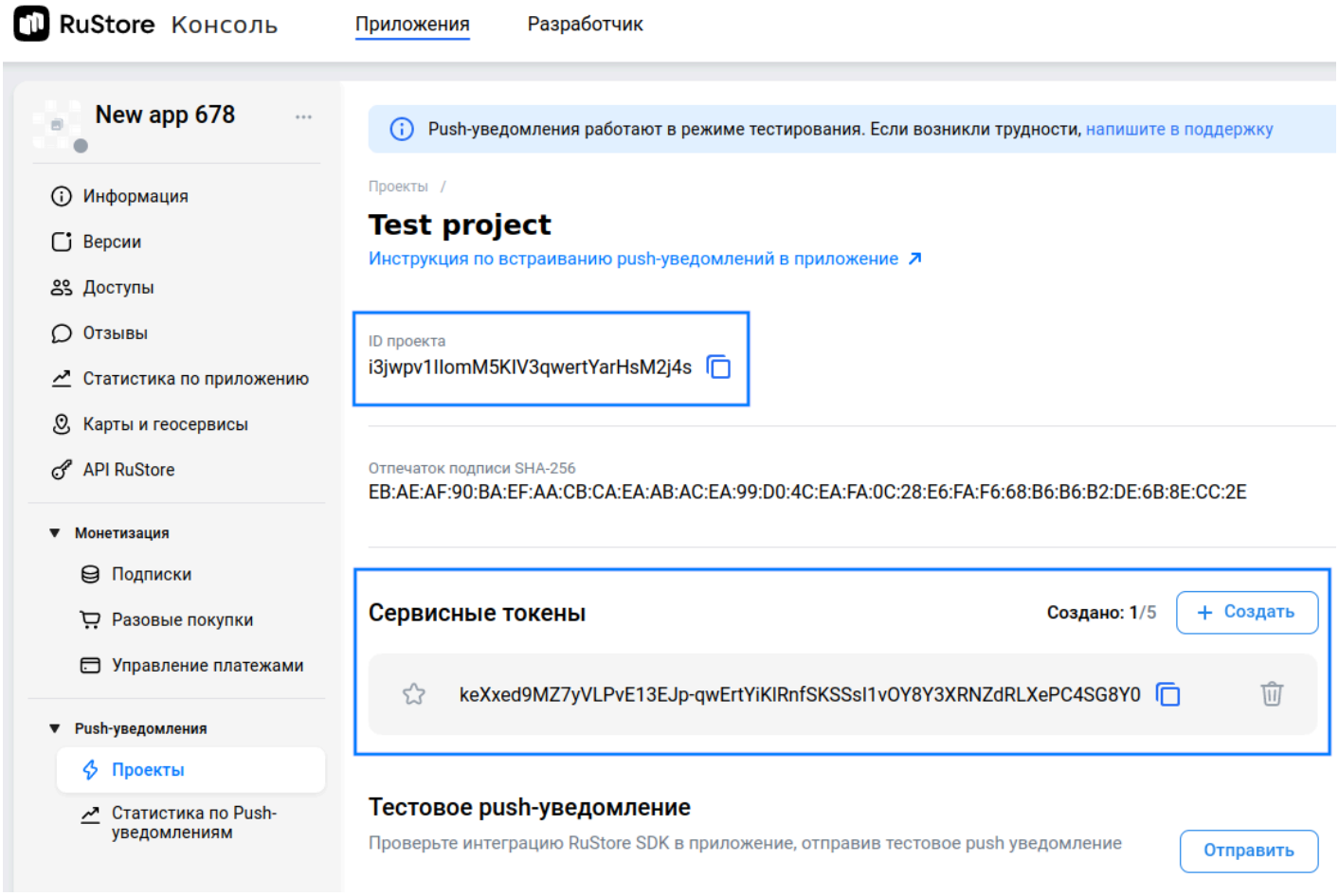


На панели внизу выберите RuStore и введите данные для настройки:



Данные для настройки ресурса можно получить в системе RuStore Консоль. Для этого на странице приложения перейдите в раздел “Push-уведомления” → “Проекты” и откройте соответствующий проект:

- Идентификатор проекта — это ID проекта;
- Пароль — это сервисный токен.



Добавление подписки на push-приложений

Подписки на уведомления из приложений присваиваются пользователям API запросом из приложения: Добавить подписку на push в базу данных.

TIP

Подписка на **Yandex.AppMetrica** может добавляться **автоматически** пользователям приложения с подключенным сервисом метрики. Подробнее в статье Интеграция с Yandex.AppMetrica.

Трекинг событий в app push

В push для приложений входят ссылки `ack` и `open`. GET запрос этих ссылок создаёт события push доставлен и push открыт. Они отобразятся в истории пользователя и отчёте по каналам.

```
{
    "hub_link": {
        "ack": "http://your.domian.com/api/v1.1/pushes/crypto_post_events?/2ab9162159...",
        "open": "http://your.domian.com/api/v1.1/pushes/crypto_post_events?/555fbe0b9..."
    }
}
```

Регистрация события в app push

С помощью вызова API можно добавлять произвольные события: Добавить события с app push.

Интеграция приложения с Altcraft

Mobile Push: интеграция приложения с Altcraft

Для интеграции приложения с Altcraft Platform разработчик должен самостоятельно реализовать:

- подключение нужного SDK push провайдера
- отправку новых push подписок в Altcraft
- отображение push сообщения
- регистрацию событий

Используйте следующие инструкции:

Шаг 1. Провайдеры: структура push сообщения

В этой инструкции описана структура push сообщений в Altcraft Platform для каждого провайдера.

Шаг 2. Обработка и добавление подписок

Узнайте, как импортировать подписки из мобильного приложения в Altcraft Platform.

Шаг 3. Регистрация событий

Читайте о том, как регистрировать события доставки и открытия push сообщения в Altcraft Platform.

Провайдеры: структура push сообщения

Altcraft Platform поддерживает следующие сервисы для отправки мобильных push сообщений:

- **Google Firebase Cloud Messaging** — для Android и iOS приложений
- **Apple Push Notification Service** — только для iOS приложений
- **Yandex.AppMetrica** — для Android и iOS приложений
- **Huawei Mobile Services** — для Android и iOS приложений
- **RuStore** — для Android приложений

> **NOTE**
>
> Требуется подключение соответствующих SDK в приложении.

> **TIP**
>
> **Yandex.AppMetrica** использует для отправки SDK Google Firebase. Для отправки уведомлений вам нужно будет установить его в приложение.
>
> В Altcraft Platform доступна интеграция с Yandex.AppMetrica для импорта профилей пользователей, регистрации их действий и связанной с ними ценности (стоимости).

Необходимо проверить, что приложение поддерживает структуру push-уведомлений Altcraft Platform. Если формат не поддерживается, Altcraft может скорректировать структуру сообщения со своей стороны.

Формат отправки push сообщения

Android Firebase (legacy)

> **NOTE**
>
> Возможно редактирование объекта JSON **(по запросу)**.
>
> Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
        "to": "SubscriptionID",
        "priority": "Priority",
        "time_to_live": 0-86400,
        "content_available": true,
        "mutable_content": true,
        "data": {
                "uid": "UID",
                "message": "Body",
                "mediaUrl": "Icon",
        "image": "Image",
                "launch_id": "LaunchID",
                "hub_link": {
                        "open": "URLOpenEvent",
                        "ack": "URLDelivEvent"
                }
        },
        "notification": {
                "title": "Title",
                "body": "Body",
                "icon": "Icon",
                "color": "ImageBackgroundColor",
                "click_action": "ClickURL",
                "sound": "Sound"
        }
}
```

TIP

`uid` — внутренний идентификатор отправки, уникальный для каждого сообщения. `launch_id` — внутренний идентификатор отправки, уникальный в рамках рассылки.

Android Firebase (api v1)

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
        "message": {
                "token": "SubscriptionID",
                "android": {
                        "priority": "Priority",
                        "ttl": TTL,
                        "data": {
                                "uid": "UID",
                                "launch_id": "LaunchID",
                                "hub_link": "{\"open\": \"URLOpenEvent\", \"ack\":
\"URLDelivEvent\"}"
                        },
                        "notification": {
                                "title": "Title",
                                "body": "Body",
                                "icon": "Icon",
                                "click_action": "ClickURL",
                                "sound": "Sound",
                                "color": "ImageBackgroundColor",
                                "light_settings": {
                                        "color": {
                                                "red": 0-1,
                                                "green": 0-1,
                                                "blue": 0-1,
                                                "alpha": 0/1
                                        },
                                        "light_on_duration": "LEDOnDuration",
                                        "light_off_duration": "LEDOffDuration"
                                },
                                "image": "Image",
                                "default_vibrate_timings": true/false,
                                "notification_count": 0-999
                        }
                }
        }
}
```

TIP

`uid` — внутренний идентификатор отправки, уникальный для каждого сообщения. `launch_id` — внутренний идентификатор отправки, уникальный в рамках рассылки.

iOS Firebase (legacy)

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
        "to": "SubscriptionID",
        "priority": "Priority",
        "time_to_live": 0-86400,
        "content_available": true,
        "mutable_content": true,
        "data": {
                "uid": "UID",
                "message": "Body",
                "mediaUrl": "Icon",
        "media": "Media",
                "launch_id": "LaunchID",
                "hub_link": {
                        "open": "URLOpenEvent",
                        "ack": "URLDelivEvent"
                }
        },
        "notification": {
                "title": "Title",
                "body": "Body",
                "icon": "Icon",
                "color": "ImageBackgroundColor",
                "click_action": "ClickURL",
                "sound": "Sound"
        }
}
```

TIP

`uid` — внутренний идентификатор отправки, уникальный для каждого сообщения. `launch_id` — внутренний идентификатор отправки, уникальный в рамках рассылки.

iOS Firebase (api v1)

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
        "message": {
                "token": "SubscriptionID",
                "apns": {
                        "headers": {
                                "apns-expiration": "TTL"
                        },
                        "payload": {
                                "aps": {
                                        "badge": 0-999,
                                        "sound": "Sound",
                                        "mutable-content": 1,
                                        "alert": {
                                                "title": "Title",
                                                "subtitle": "SubTitle",
                                                "body": "Body"
                                        }
                                },
                                "click-url": "ClickURL",
                                "uid": "UID",
                                "launch_id": "LaunchID",
                                "hub_link": "{\"open\": \"URLOpenEvent\", \"ack\":
\"URLDelivEvent\"}",
                                "media": "Media",
                "mediaUrl": "Icon"
                        },
                        "fcm_options": {
                                "image": "Media"
                        }
                }
        }
}
```

TIP

`uid` — внутренний идентификатор отправки, уникальный для каждого сообщения. `launch_id` — внутренний идентификатор отправки, уникальный в рамках рассылки.

iOS APNS

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
        "aps": {
                "badge": 0-999,
                "sound": "Sound",
                "mutable-content": 1,
                "alert": {
                        "title": "Title",
            "subtitle": "SubTitle",
                        "body": "Body"
                }
        },
        "click-url": "ClickURL",
        "icon": "Icon",
    "media": "Media",
        "hub_link": {
                "open": "URLOpenEvent",
                "ack": "URLDelivEvent"
        }
}
```

AppMetrica iOS

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
        "push_batch_request": {
                "group_id": 97737,
                "tag": "tag",
                "batch": [
                        {
                                "messages": {
                                        "iOS": {
                                                "silent": true/false,
                                                "content": {
                                                        "title": "Title",
                                                        "text": "Body",
                                                        "badge": 0-999,
                                                        "expiration": 0,
                                                        "data": "{\"hub_link\":
{\"ack\":\"URLDelivEvent\",\"open\":\"URLOpenEvent\"},\"icon\":\"Icon\",\"media\":\"Media\"
}"
                                                },
                                                "open_action": {
                                                        "url": "URL",
                                                        "deeplink": "Deeplink"
                                                }
                                        }
                                },
                                "devices": [
                                        {
                                                "id_values": ["SubscriptionID"],
                                                "id_type": "ios_push_token"
                                        }
                                ]
                        }
                ]
        }
}
```

AppMetrica Android

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
        "push_batch_request": {
                "group_id": 97737,
                "tag": "tag",
                "batch": [
                        {
                                "messages": {
                                        "android": {
                                                "silent": true/false,
                                                "content": {
                                                        "title": "Title",
                                                        "text": "Body",
                                                        "time_to_live": 0,
                                                        "image": "Icon",
                                                        "icon_background":
"IconBackground",
                                                        "banner": "Banner",
                                                        "led_color": "LEDColor",
                                                        "data": "{\"hub_link\":
{\"ack\":\"URLDelivEvent\",\"open\":\"URLOpenEvent\"}}"
                                                },
                                                "open_action": {
                                                        "url": "URL",
                                                        "deeplink": "Deeplink"
                                                }
                                        }
                                },
                                "devices": [
                                        {
                                                "id_values": ["SubscriptionID"],
                                                "id_type": "google_aid"
                                        }
                                ]
                        }
                ]
        }
}
```

HMS Android

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
        "message":{
                "android":{
                        "notification":{
                                "badges":{
                                        "set_num":2
                                },
                                "body":"Уважаемый пользователь, предлагаем Вам наш новый
продукт, productName!",
                                "click_action":{

"intent":"#Intent;compo=com.example/app2;S.W=U;end",
                                        "type":1
                                },
                                "color":"#cdc167",
                                "icon":"https://i.postimg.cc/2yVZyQMn/demo-Push-Icon.jpg",
                                "importance":"NORMAL",
                                "light_settings":{
                                        "color":{
                                                "alpha":1,
                                                "blue":0.70980394,
                                                "green":0.78039217,
                                                "red":0.5686275
                                        },
                                "light_off_duration":"1.5s",
                                "light_on_duration":"1.5s"
                                },
                                "sound":"tadaam.ogg",
                                "title":"Title of push message"
                        },
                        "params":{
                                "hub_link": "{\"open\": \"URLOpenEvent\", \"ack\":
\"URLDelivEvent\"}",
                                "launch_id":"2_6j_4LxYQQ8VECT",
                                "uid":"w4LxYQQsznE1_2_6j_i_o___4.2Pzr5NNonCwUWQwvJ_4u71jH"
                        },
                        "ttl":"604800s"
                },
                "token":[
                        "AGIwonWCgyDXGEG-airg4cUbVxeq0RFDfTuDM2hPWRrxfMociyvLNmn-
GWdd4I6ZDrXIee46IrWSjC-cPDauP3ciS74bVi8wJ7T3PwHhvEY38VdYSu5zeHRXQzBRDgTnxA"
                ]
        }
}
```

HMS iOS

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.


```
{
        "message":{
                "apns":{
                        "headers":{
                                "apns-expiration":"3600"
                        },
                        "hms_options":{
                                "target_user_type":1
                        },
                        "payload":{
                                "apple":"red",
                                "aps":{
                                        "alert":{
                                                "body":"Уважаемый пользователь, предлагаем
Вам наш новый продукт, productName!",
                                                "subtitle":"sub title",
                                                "title":"Title of push message"
                                        },
                                        "badge":0,
                                        "mutable_content":1,
                                        "sound":"default"
                                },
                                "click-url":"",
                                "hub_link": "{\"open\": \"URLOpenEvent\", \"ack\":
\"URLDelivEvent\"}",
                                "launch_id":"2_6j_4LxYKovmDvu",
                                "mediaUrl":"https://i.postimg.cc/2yVZyQMn/demo-Push-
Icon.jpg",
                                "uid":"w4LxYKq9nBMc_2_6j_i_o___4.2Pzr5NNonCwUWQwvJ_aDfVX"
                        }
                },
                "token":[
                        "IQAAAACy0huaAQwLF6BO92GYzCAnfqZgaxhD1f7wLWGmOxQ6bFKCQYQQYtTi2-
1hXHlsXAnCzLQdX3m4RTCCEI-KYu-XcaBJYovDZANg2eRTvPv7qA"
                ]
        }
}
```


RuStore Android

NOTE

Возможно редактирование объекта JSON **(по запросу)**.

Поддерживается добавление дополнительных данных в объект `data`. Это можно сделать для каждой отправки персонально в настройках сообщения Altcraft MP.

```
{
  "validate_only": false,
  "message": {
    "token":  "{{.SubscriptionID}}",
    "data": {
      "uid": "{{.UID}}",
      "message": "{{.Body}}",
      "mediaUrl": "{{.Icon}}",
      "image":   "{{.Image}}",
      "launch_id": "{{.LaunchID}}",
      "hub_link": "{\"open\": \"{{.URLOpenEvent}}\", \"ack\": \"{{.URLDelivEvent}}\"}"
    },
    "android": {
      "priority": "{{.Priority}}",
      "notification": {
        "title": "{{.Title}}",
        "body": "{{.Body}}",
        "icon": "{{.Icon}}",
        "color": "{{.ImageBackgroundColor}}",
        "image": "{{.Image}}",
        "click_action": "{{.ClickURL}}",
        "sound": "{{.Sound}}",
        "channel_id": "{{.ChannelID}}",
        {{if ne .LEDColorHex ""}}
        "light_settings": {
          "color": {
            "red": {{.LEDColorSRGBA.R}},
            "green": {{.LEDColorSRGBA.G}},
            "blue": {{.LEDColorSRGBA.B}},
            "alpha": {{.LEDColorSRGBA.A}}
          },
          "light_on_duration": "{{.LEDOnDuration}}",
          "light_off_duration": "{{.LEDOffDuration}}"
        },
        {{end}}
        "notification_count": {{.Badges}}
      },
      "ttl": {{.TTL}}
    }
  }
}
```

TIP

`uid` — внутренний идентификатор отправки, уникальный для каждого сообщения. `launch_id` — внутренний идентификатор отправки, уникальный в рамках рассылки.

Обработка и добавление подписки

При получении push токена приложение должно передать его в Altcraft Platform для формирования подписки. Для этого используется API запрос.

С помощью этого запроса можно:

- Добавить новые подписки.
- Поменять статус уже существующих подписок.

Режим поиска профиля можно задавать с помощью поля `matching` в запросе (см. спецификацию ниже). Если профиль не будет найден, то будет создан новый.

При этом, если в ресурсе указано несколько баз данных, то все они будут обрабатываться по отдельности. Запрос к такому ресурсу будет создавать и обновлять профили сразу в нескольких базах. Пример такого случая можно посмотреть ниже.

Если в ресурсе нет ни одной базы, то будет возвращена ошибка.

URL-адрес

Метод: POST

`https://<cookiesaver_host>/push_subscribe`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
id	string	“abcdefghijklmnqrstuvwxyz”	Нет	Идентификатор профиля
resource_token	string	“abcdefghijklmnqrstuvwxyz”	Да	Уникальный идентификатор ресурса
db_ids	[ ] int	[2,3]	Нет	Фильтр баз данных ресурса.
matching	string	“custom”	Нет	Режим поиска подписчика. Если поле не передано или пустое, то будет использован поиск профиля по id или, если id тоже не передано, по пуш подпискам. **Для push_subscribe метода это поле может принимать только следующие значения:** • custom • profile_id

Параметр	Тип	Пример	Обязательный	Описание
subscriptions	JSON array SubscriptionObject	`[{ “provider” : “android-firebase”, “subscription_id” : “%subscriptionid%” }]`	Да	Подписки на push, которые нужно добавить в профиль или обновить. “subscription_id” — уникален для каждого устройства.

SubscriptionObject

Параметр	Тип	Пример	Обязательный	Описание
provider	string	“android-firebase”	Да	Идентификатор провайдера в Altcraft MP. Возможны следующие значения: • android-firebase • ios-firebase • ios-apns • yandex-Appmetrica-device-id-ios • yandex-Appmetrica-device-id-android • yandex-Appmetrica-ios-ifa • yandex-Appmetrica-google-aid • yandex-Appmetrica-android-token • yandex-Appmetrica-ios-token • android-huawei • ios-huawei **Каждому типу токена в AppMetrica соответствует свой id провайдера.**
subscription_id	string	“abcdef-ghijklmn-qrstuvwxyz”	Да	Токен push подписки
unsubscribe	bool	false	Нет	Флаг отписки. Отменяет подписку, если true.

Возвращаемые параметры

Параметр	Тип	Описание
data	JSON array DataObject	Информация о созданных подписках. **В массиве может быть несколько объектов в следующих случаях:** • В запросе было передано несколько SubscriptionObject • В ресурсе Altcraft MP используется несколько баз данных. В этом случае для каждой базы из ресурса добавится новый профиль с подписками.
error	int	Код ошибки
error_text	string	Текст ошибки

DataObject

Параметр	Тип	Описание	
subscription_id	string	“abcdef-ghijklmn-qrstuvwxyz”	Токен push подписки
unsubscribe	bool	false	Флаг, была ли подписка добавлена или наоборот отменена
profile_id	string	“5fb62e815ae47a4aa04dfadd”	id профиля (переданный или вновь созданный), для которого добавлена подписка.
provider	string	“android-firebase”	Идентификатор провайдера в Altcraft MP. Возможны следующие значения: • android-firebase • ios-firebase • ios-apns • yandex-Appmetrica-device-id-ios • yandex-Appmetrica-device-id-android • yandex-Appmetrica-ios-ifa • yandex-Appmetrica-google-aid • yandex-Appmetrica-android-token • yandex-Appmetrica-ios-token • android-huawei • ios-huawei Каждому типу токена в AppMetrica соответствует свой id провайдера
db_id	int	2	id базы данных в Altcraft MP, в которой находится профиль
resource_id	int	1	id ресурса в Altcraft MP
subscription_hash_id	string	“92807e89”	Хеш подписки

Параметр	Тип	Описание	
status	string	“error”	Статус операции • ok • error
description	string	“Duplicate push subscription”	Описание ошибки

Примеры

Запрос с матчингом по системному id профиля

Запрос

```
{
  "id": "59db33d269d426509e9c4bf9",
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "data": [
    {
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false,
      "profile_id": "59db33d269d426509e9c4bf9",
      "provider": "android-firebase",
      "db_id": 17,
      "resource_id": 5,
      "subscription_hash_id": "adsfdfdffdf",
      "status": "ok",
      "description": ""
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```

Запрос с матчингом по дополнительному полю профиля

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "matching": "custom",
  "field_name": "CustomNumber1",
  "field_value": 123,
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "data": [
    {
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false,
      "profile_id": "59db33d269d426509e9c4bfc",
      "provider": "android-firebase",
      "db_id": 17,
      "resource_id": 5,
      "subscription_hash_id": "adsfdfdffdf",
      "status": "ok",
      "description": ""
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```

Запрос с матчингом по push-подпискам

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "data": [
    {
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false,
      "profile_id": "59db33d269d426509e9c4bfa",
      "provider": "android-firebase",
      "db_id": 17,
      "resource_id": 5,
      "subscription_hash_id": "adsfdfdffdf",
      "status": "ok",
      "description": ""
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```

Запрос к ресурсу с несколькими базами данных

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "matching": "custom",
  "field_name": "CustomNumber1",
  "field_value": 12345,
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "error": 200,
  "error_text": "Successful operation",
  "data": [
    {
      "db_id": 34,
      "description": "",
      "profile_id": "6156ed18eb7fec10a206e916",
      "provider": "android-firebase",
      "resource_id": 22,
      "status": "ok",
      "subscription_hash_id": "340ca49c",
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false
    },
    {
      "db_id": 2,
      "description": "matching error: unknown field",
      "resource_id": 22,
      "status": "error"
    }
  ]
}
```

Запрос с матчингом по дополнительному полю профиля в определенной базе

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "db_ids": [1],
  "matching": "custom",
  "field_name": "CustomNumber1",
  "field_value": 123,
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": [
    {
      "db_id": 1,
      "description": "",
      "profile_id": "637f1a284fda7c5c08be62f",
      "provider": "android-firebase",
      "resource_id": 31,
      "status": "ok",
      "subscription_hash_id": "f8af057de",
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false
    }
  ]
}
```

Запрос не обновил и не добавил ни одной подписки профиля

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "matching": "custom",
  "field_name": "CustomNumber1",
  "field_value": 12345,
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "error": 200,
  "error_text": "Successful operation",
  "data": [
    {
      "db_id": 34,
      "description": "",
      "profile_id": "6156ed18eb7fec10a206e916",
      "resource_id": 22,
      "status": "ok"
    }
  ]
}
```

Регистрация событий

Существует две схемы регистрации событий доставки и открытия push сообщения в Altcraft MP:

1. Напрямую от приложения.
2. Через промежуточный сервис.

Регистрация событий напрямую от приложения

По умолчанию при отправке push сообщения в полезных данных отправляется объект `hub_link`

```
{
     "hub_link" : {
         "ack" :
"http://your.domain.com/api/v1.1/pushes/crypto_post_events?/2ab9162159..." ,
         "open" :
"http://your.domain.com/api/v1.1/pushes/crypto_post_events?/555fbe0b9..."
     }
}
```

Приложение должно выполнить GET запрос по ссылкам `ack` и `open`, для регистрации доставки и открытия пуша соответственно.

Регистрация событий через промежуточный сервис

> **CAUTION**
>
> Этот способ требует наличия Extra UID в пользовательских данных push сообщения **(настраивается по запросу)**

Приложение сообщает промежуточному сервису факт события с передачей последнему Extra UID из пуша.

Шлюз должен передать через API это событие в Altcraft Platform.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/pushes/post_events`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен

Параметр	Тип	Пример	Обязательный	Описание
events	int	1 - доставлен 2 - открыт 3 - доставлен и открыт	Да	События
uid	string	“uid”	Да	Приходит вместе с push сообщением от Altcraft в Data payload.

Пример запроса

```
{
     "token" :  "abcdefghijklmnqrstuvwxyz" ,
     "events" : 1,
     "uid" :  "e|2_4S_4HFn8RDhhmA|w4HFn8RPDFUi_2_4S_S_3___6.2NvyN8Hb67wmzJshM_6BuFTt|ios-apns"
}
```

Пример ответа

```
{
     "error" : 0,
     "error_text" :  "Successful operation"
}
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Web Push

Web Push: настройка и подключение

TIP

Настройка **мобильных push-уведомлений** описана в соответствующей статье.

	Web Push — это небольшие всплывающие окна в браузере. Вы можете отправлять Web Push-уведомления в десктопные и мобильные браузеры ваших клиентов. Когда пользователь подписывается на уведомления, то при рассылке оно отображается на его устройстве в углу или верхней части экрана. Причем отображается оно, даже если окно браузера неактивно.

В отличие от Mobile Push, **Web Push** уведомления могут приходить как в десктопные, так и в мобильные браузеры. Такие уведомления приходят от веб-сайтов. **Mobile Push** уведомления приходят от **мобильных приложений**.

Push-уведомление может содержать следующие данные:

- **Заголовок** — аналогичен теме Email сообщения;
- **Тело** — любой текст для сообщения;
- **Иконка** — изображение в уведомлении;
- **Ссылка для перехода** — перенаправление при нажатии на push-уведомление.

Различные push-провайдеры поддерживают создание уведомлений для разных браузеров:

Сервис	Поддерживаемые браузеры
Google Firebase (бывш. Google Cloud Messaging)	Google Chrome, Mozilla Firefox, Opera, Яндекс.Браузер, Samsung Internet Browser
Apple Push Notification Service	Apple Safari (в т.ч. PWA-приложения)
Mozilla Services	Mozilla Firefox

INFO

Обратите внимание, что отправка уведомлений через браузер **Mozilla Firefox** возможна как напрямую — **Mozilla Services** (SDK встроен в браузер), так и через сервис **Google Firebase**.

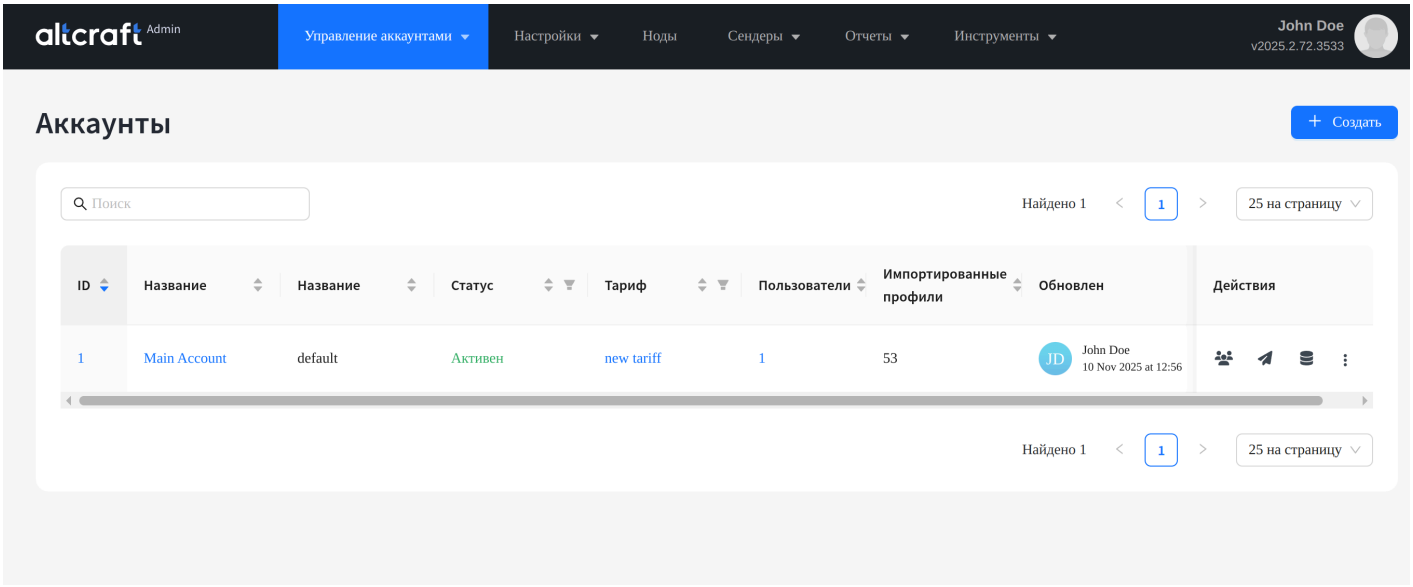
Как настроить и подключить Web Push-уведомления

Предварительные настройки

Технические настройки

Убедитесь, что ваш web-сайт работает **по протоколу HTTPS**.

Кроме того, в системе Altcraft Platform должны быть **настроены web-пиксели**. Зайдите в **Панель администратора** → **Управление аккаунтами**. Нажмите на ваш аккаунт:



Активируйте **push-уведомления** и **pixel** для **аккаунтов**, с которых планируется отправка:

Настройки ресурса в платформе

> **TIP**
>
> Для каждого сайта должен быть создан **отдельный** ресурс.

1. В платформе Altcraft создайте или выберите **Базы данных**, в которых будут храниться подписчики на push-уведомления.

2. Выберите **Ресурс** для Web Push-подписок или создайте новый.

3. Введите **Имя** и **Описание** ресурса.

4. В поле **URL веб-сайта, формы или приложения** укажите адрес сайта, на котором происходит подписка на push и с которого будут отправляться уведомления. В дальнейшем этот сайт вносится в файлы Toolkit, поэтому, если указать некорректный адрес, push-уведомления работать **не будут**.

5. В поле **Трекинг домен по умолчанию** укажите домен, на котором запущен трекинг Altcraft. Этот домен будет обрабатывать переходы ваших клиентов по ссылкам в сообщениях.

Основные настройки

Имя ресурса: *

Web Push Resource

Описание ресурса:

Установить группу:

Задать политику отправки для ресурса:

Не определена

Базы данных: *

5 - New powerful database ✖

URL веб-сайта, формы или приложения: *

https://example.org

Трекинг домен по умолчанию: *

trk.example.org

Добавить дополнительные URL

Повышать приоритет для новых подписок

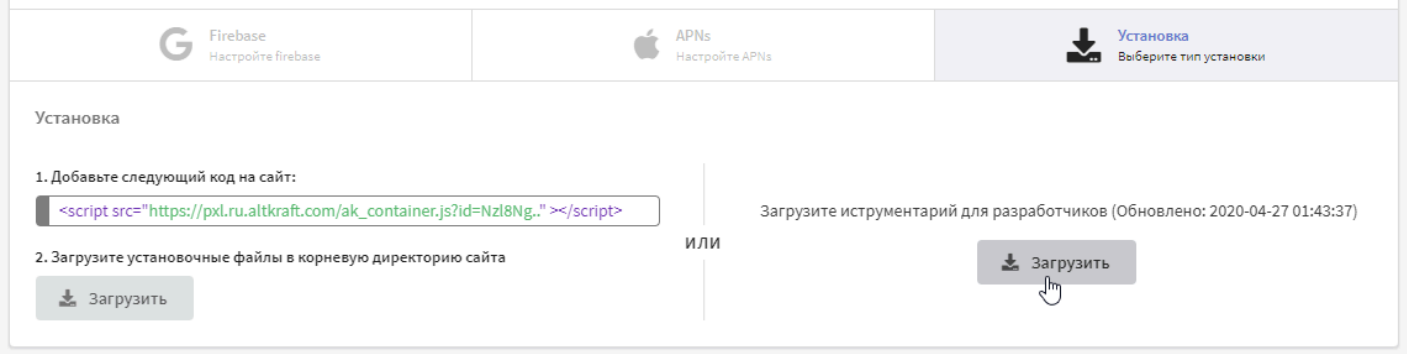
Подключение Web Push на сайт

Загрузка Toolkit

После того как нужные настройки ресурса заполнены, можно сохранить его и скачать **Toolkit**, части которого затем размещаются на вашем сайте. **Toolkit** — это набор файлов, предназначенных для работы Web Push-уведомлений на вашем сайте. Он помогает настроить подписку у пользователей и отправку уведомлений через браузеры.

> **INFO**
>
> Toolkit сам по себе не вызывает окно подписки, он лишь содержит необходимые файлы и скрипты для работы с уведомлениями. Чтобы пользователь увидел предложение подписаться, нужно добавить код на сайте, который вызывает соответствующую функцию `ak-push.js`.

Toolkit появится в разделе **Установка** вместо предупреждения “**Сохраните ресурс, для того чтобы скачать инструментарий**”:



Установка нужных файлов на сайт возможна двумя способами:

1. Добавление кода на сайт и загрузка установочных файлов в корневую директорию сайта;
2. Загрузка инструментария для разработчиков.

Способ 1: Добавление кода на сайт и загрузка установочных файлов в корневую директорию сайта

Алгоритм действий для данного варианта указан в левой части. Добавьте указанный код на сайт, затем сохраните и загрузите в корневую директорию сайта установочные файлы:

- **manifest.json** — данные о проекте для отправки push через Google Chrome;
- **service-worker.js** — service worker, который регистрируется в браузерах пользователей и принимает push сообщения.

Здесь файл **service-worker.js** является **динамическим**. Он автоматически обновляется каждый раз, когда вы вносите изменения в настройки ресурса.

```
importScripts('https://pxl.test.com/service-worker.js?id=MXw3');
```

Способ 2: Загрузка инструментария для разработчиков

Алгоритм действий для данного варианта указан в правой части. Нажмите на кнопку “**Загрузить**”, после чего будет загружен архив Toolkit. Распакуйте архив и разместите файлы Toolkit в корневой директории вашего сайта.

Toolkit состоит из четырёх файлов:

- **README.txt** — инструкция по установке Toolkit;
- **ak-push.js** — скрипт, который показывает пользователям предложение подписаться на push-оповещения;
- **manifest.json** — данные о проекте для отправки push через Google Chrome;
- **service-worker.js** — service worker, который регистрируется в браузерах пользователей и принимает push-сообщения.

Здесь файлы **service-worker.js** и **ak-push.js** является **статичным**. При использовании данного способа после изменения настроек ресурса необходимо заново загрузить Toolkit и обновить файлы на сайте.

TIP

При использовании инструментария для разработчиков можно **изменить путь до файла service-worker.js**. Для этого откройте файл **ak-push.js**, найдите свойство **swPath** и отредактируйте его, указав нужный путь до файла.

Добавление скриптов на сайт

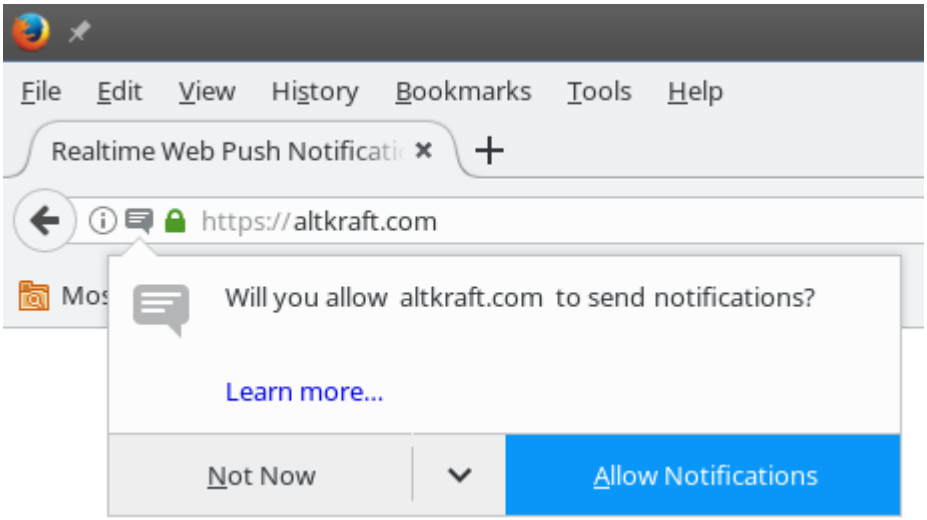
После того как Toolkit загружен на сайт, добавьте код, который будет показывать окно подписки для необходимого действия.

Ниже приведен пример кода для кнопки **“Подписаться”**. Этот код запускает библиотеку и инициализирует подписку:

```
<button id="init_sub">Подписаться на уведомления</button> // Кнопка подписки

// ..Основная часть кода..

<script>
        document.getElementById('init_sub').addEventListener('click', function() { // По
клику на кнопку..
        try {
                const akPush = new AKPush();
                akPush.initSubscription() // ..показать форму подписки
                }
        catch (e) {
                console.log(e); // ..или записать ошибку в консоль браузера
                }
        });
</script>
```



TIP

Через платформу Altcraft вы можете создать собственный **попап** с предложением подписки. Подробнее прочесть об этом можно в этой статье.

Код, описанный выше, работает по следующей логике:

- Пользователь нажимает на кнопку;
- Вызывается метод `initSubscription`;
- Всплывает системное окно браузера с предложением подписаться на уведомления этого сайта.

Пользователь может настроить свою логику и решить, каким путем посетитель сайта дойдет до системного окна подписки. Стили системного браузерного окна с запросом разрешений на push-уведомления изменить нельзя.

Как только все будет настроено, можно будет зайти на сайт и подписаться на push-уведомления. В браузере клиента при подтверждении устанавливается Service worker, отображающий сообщения даже при свернутом окне браузера.

Обратите внимание, что подписка на один и тот же ресурс будет происходить повторно **для каждого браузера (или учетной записи), которым пользуется клиент**. Идентификатором для подписки служит **токен браузера или учетной записи**. Таким образом, у одного профиля может быть несколько активных push-подписок, если он подписывался на уведомления с разных браузеров.

При подписке на уведомления скрипт может дополнительно передавать данные о пользователе или регистрировать информацию о событиях. Подробнее о настройке такого кода можно прочитать в этой статье.

Передача данных в платформу

Когда пользователь дает разрешение на получение уведомлений, платформа автоматически создаёт для него профиль в платформе. Помимо создания профиля также возможна передача других дополнительных данных. В данной статье мы расскажем, как настроить скрипт для сбора информации о пользователе и его действиях.

TIP

Настройка Web Push на платформе Altcraft предполагает использование **методов Web Push SDK**. Подробнее о методах SDK можно узнать из этой статьи. Инструкцию по установке SDK на ваш сайт можно прочесть здесь.

Импорт профилей клиентов при подписке на Web Push-уведомления

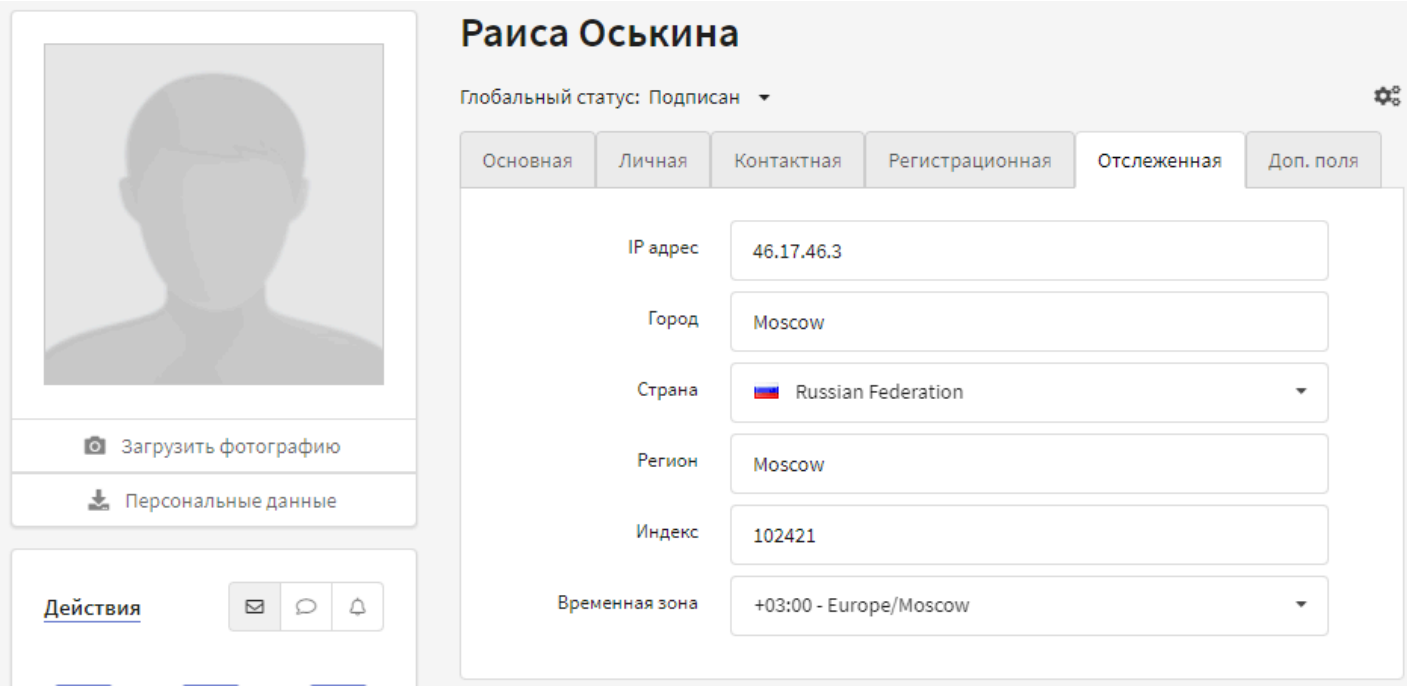
Когда пользователь подпишется на Web Push-уведомления, в **Базы данных** и в **Ресурс**, в котором был сгенерирован Toolkit, размещенный на сайте, будет добавлен **новый профиль**.

INFO

Если один из профилей базы данных ранее уже открывал вашу email-рассылку с браузера, на котором сейчас подписывается на push, то подписка будет добавлена ему. Новый профиль в этом случае не создаётся.

Когда подписчик кликает на push-уведомление (переход по ссылке), платформа Altcraft собирает данные о профиле:

- страна, регион, часовой пояс, город, почтовый индекс и IP адрес, с которого произошла подписка на push;
- браузер и операционная система подписчика.



В случае устаревания токена или повторной подписки пользователя на push, платформа автоматически обновит его подписку и выдаст новый токен. Таким образом в базе данных не будут создаваться дубликаты профилей.

NOTE

Мобильные браузеры также поддерживают Service worker и Web push-уведомления.

Передача данных клиента при импорте профиля через push-канал

В момент регистрации подписки на push-уведомления в профиль нового пользователя можно записать любые данные, известные на момент подписки. Например, имя или фамилия клиента, дата рождения, информация о просмотренных страницах и другая бизнес-информация. При этом можно проверить, нет ли среди существующих подписчиков клиента с такими же данными.

Пример с поиском по email в ресурсе и добавлением имени и фамилии пользователя:

```
const akPush = new AKPush();
akPush.initSubscription(
        {
                email: 'example@email.com'
        }, // Поиск профиля по email. Также будет записано в профиль, если совпадений не
найдётся.

        {
                _fname: 'Anatoly',
                _lname: 'Wasserman'
        } // запись в профиль подписчика имени и фамилии
)
```

Если к ресурсу привязано несколько баз данных, вы можете задать идентификатор той базы, в которую необходимо импортировать профиль. Пример с поиском по email в базе данных с `id=1`:

```
const akPush = new AKPush();
akPush.initSubscription(
        {
                db: 1
                email: 'example@email.com'
        }, // Поиск профиля по email. Также будет записано в профиль, если совпадений не
найдётся.

        {
                _fname: 'Anatoly',
                _lname: 'Wasserman'
        } // запись в профиль подписчика имени и фамилии
)
```

Пример с поиском по email в базах данных с `id=15` и `id=20`:

```
const akPush = new AKPush();
akPush.initSubscription(
        {
                db: [15,20]
                email: 'example@email.com'
        }, // Поиск профиля по email. Также будет записано в профиль, если совпадений не найдётся.

        {
                _fname: 'Anatoly',
                _lname: 'Wasserman'
        } // запись в профиль подписчика имени и фамилии
)
```

TIP

Список полей профиля приведён в этой статье.

Обновление данных профиля и приостановка подписок

CAUTION

akPush.updateSubscription вернет ошибку, если перед этим не был получен актуальный токен через **akPush.initSubscription**, или он был удален из кеша с помощью **akPush.removeToken**.

Пользователь, подписавшийся в браузере на пуши, может менять свое состояние. Например, авторизоваться в личном кабинете и получить другой матчинг. При этом текущую пуш подписку необходимо отключить на предыдущем профиле и включить на новом. Для этих целей необходимо использовать метод `akPush.updateSubscription`:

```
const akPush = new AKPush();
akPush.updateSubscription(
        {
                db: [15,20]
                email: 'example@email.com'
        },

        {
                _fname: 'Anatoly',
                _lname: 'Wasserman'
        }, // запись в профиль подписчика имени и фамилии
        false // флаг для inexclusive,  определяет, нужно ли приостанавливать все подписки на ресурс с текущим токеном браузера у остальных профилей (по умолчанию)
)
```

События создания и обновления Web Push-подписки

Пользователи платформы могут подписаться на события создания и обновления (изменения токена подписчика) push-подписки. При наступлении такого события данные о подписке можно отправить на сторонний сервер с помощью дополнительного кода.

Чтобы подписаться на события, добавьте следующий код на страницы сайта:

```
AKPush.events.on('subscribe', (data) => { // подписка на событие subscribe
   console.log(data);
});
AKPush.events.on('update_subscription', (data) => { // подписка на событие update_subscription
   console.log(data);
});
```

Данный код выведет в консоль токен подписчика, а также **информацию о браузере и push-провайдере**.

Методы Web Push SDK

TIP

Подробнее о настройке Web Push-рассылок на платформе, а также установке SDK на сайт можно прочитать в этом разделе.

В данной статье описаны методы Web Push SDK и возможные параметры для них. Для каждого метода также приведен пример использования.

initSubscription

Метод инициализирует подписку. Используется для записи данных клиента в профиль и выполнения поиска среди существующих профилей.

Параметры

Параметр	Тип	Обязательный	Описание
match	object	**Да**	Определяет [матчинг](/user-guide/profiles-and-databases/matching) профиля. Если не будет передана база профилей для поиска, платформа будет искать профиль в привязанных к ресурсу базах.
update	object	Нет	Содержит данные для внесения в новый или существующий профиль

Пример использования

```
const akPush = new AKPush();
AKPush.initSubscription(
        {
                db: 1
                email: 'example@email.com'
        }, // Поиск профиля по email. Также будет записано в профиль, если совпадений не найдётся.

        {
                _fname: 'Anatoly',
                _lname: 'Wasserman'
        } // запись в профиль подписчика имени и фамилии
)
```

updateSubscription

Метод для обновления данных профиля и изменения статуса push-подписок.

Параметры

Параметр	Тип	Обязательный	Описание
match	object	**Да**	Определяет [матчинг](/user-guide/profiles-and-databases/matching) профиля. Если не будет передана база профилей для поиска, платформа будет искать профиль в привязанных к ресурсу базах.
update	object	Нет	Содержит данные для внесения в новый или существующий профиль. Если необходимо передать параметр **inexclusive** без обновления данных профиля, нужно указать в **update** пустой объект `{}`.
inexclusive	boolean	Нет	Если флаг `inexclusive` не передан или равен `false`: 1. Если фильтр баз не указан, то в каждой базе ресурса будет только один профиль с одной подпиской на актуальный токен со статусом "Подписан". 2. Если фильтр указан, то в каждой из заданных баз будет только один профиль с одной подпиской на актуальный токен со статусом подписан. В остальных базах при этом будут приостановлены все совпадающие подписки на ресурс у всех профилей. Если флаг `inexclusive` равен `true`, то вызов эквивалентен вызову `AKPush.initSubscription`, т.е. произойдет обновление данных профиля и смена статуса подписки на "Подписан".

Пример использования

```
const akPush = new AKPush();
AKPush.updateSubscription(
        {
                db: [15,20]
                email: 'example@email.com'
        },

        {
                _fname: 'Anatoly',
                _lname: 'Wasserman'
        }, // запись в профиль подписчика имени и фамилии
        false // нужно ли приостанавливать все подписки на ресурс с текущим токеном браузера у остальных профилей (по умолчанию)
)
```

AKPush.events.on

Метод для подписки на события, связанные с push-уведомлениями.

Параметры

Параметр	Тип	Обязательный	Описание

subscribe update_subscription suspend_subscription	string	**Да**	Название произошедшего события

callback	function	**Да**	Функция, которая будет вызвана в момент, когда происходит событие. Для `subscribe` будут возвращены параметры:
			{
			browser: "<браузер подписчика>"
			provider: "<push-провайдер>"
			token: "<токен подписчика>"
			}
			Для `update_subscription` и `suspend_subscription` будут возвращены параметры:
			{
			browser: "<браузер подписчика>"
			provider: "<push-провайдер>"
			token: "<токен подписчика>"
			tokenCached: "<кэшированный токен подписчика>"
			}
			Параметр токена может вернуть `null` в качестве значенияи.

Пример использования

```
AKPush.events.on('subscribe', (data) => { // подписка на событие subscribe
   console.log(data);
});
AKPush.events.on('update_subscription', (data) => { // подписка на событие
update_subscription
   console.log(data);
});
```

suspendSubscription

Метод, приостанавливающий подписку пользователя на push-уведомления.

Параметры

Параметр	Тип	Обязательный	Описание
match	object	**Да**	Определяет [матчинг](/user-guide/profiles-and-databases/matching) профиля. Если не будет передана база профилей для поиска, платформа будет искать профиль в привязанных к ресурсу базах.
update	object	Нет	Содержит данные для внесения в новый или существующий профиль

Пример использования

```
const akPush = new AKPush();
AKPush.suspendSubscription(
        {
                db: 1
                email: 'example@email.com'
        }, // Поиск профиля по email.

        {
                _status: 'inactive'
        } // обновление статуса в профиле подписчика
)
```

AKPush.removeToken

Данный метод используется для удаления токена устройства из кэша. Используется, например, в тех ситуациях, когда на устройстве сменился пользователь. Для этого метода **не нужно задавать параметры**.

Пример использования

```
AKPush.removeToken()
```

Apple Safari

	Для настройки Web Push в браузер Safari необходимо иметь расширение Developer в Apple аккаунте.

CAUTION

Все действия выполняются на macOS X устройстве.

INFO: Отправка пушей в PWA приложения

В версиях iOS 16.4+ Apple добавила **поддержку push-уведомлений от PWA-приложений**.

PWA (от англ. “progressive web app”) — это технология, которая позволяет установить сайт на смартфон или на компьютер как приложение. Установка не требует загрузки через магазин приложений — пользователь просто заходит на сайт и добавляет иконку на рабочий стол своего устройства. Такое приложение обладает функционалом нативного мобильного приложения: оно может работать в автономном офлайн режиме, кэшировать данные и принимать пуши.

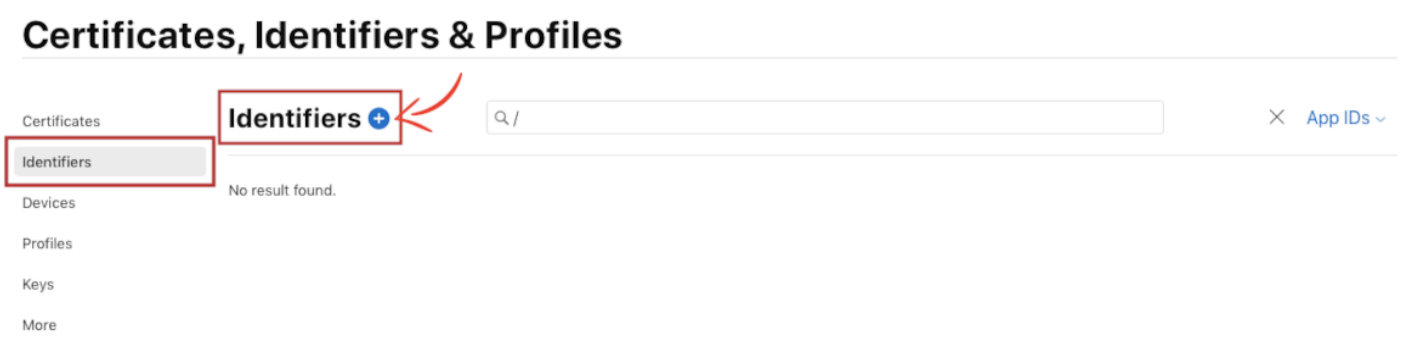
Чтобы отправлять push-уведомления пользователям PWA-приложений, необходимо **настроить провайдер APNS для Safari в ресурсе**. При сохранении настроек автоматически подключится провайдер PWA.

Обратите внимание, идентификатор подписки на PWA пуши привязан к конкретному ресурсу. Если вы перенесёте подписки из одного ресурса в другой, они не будут работать.

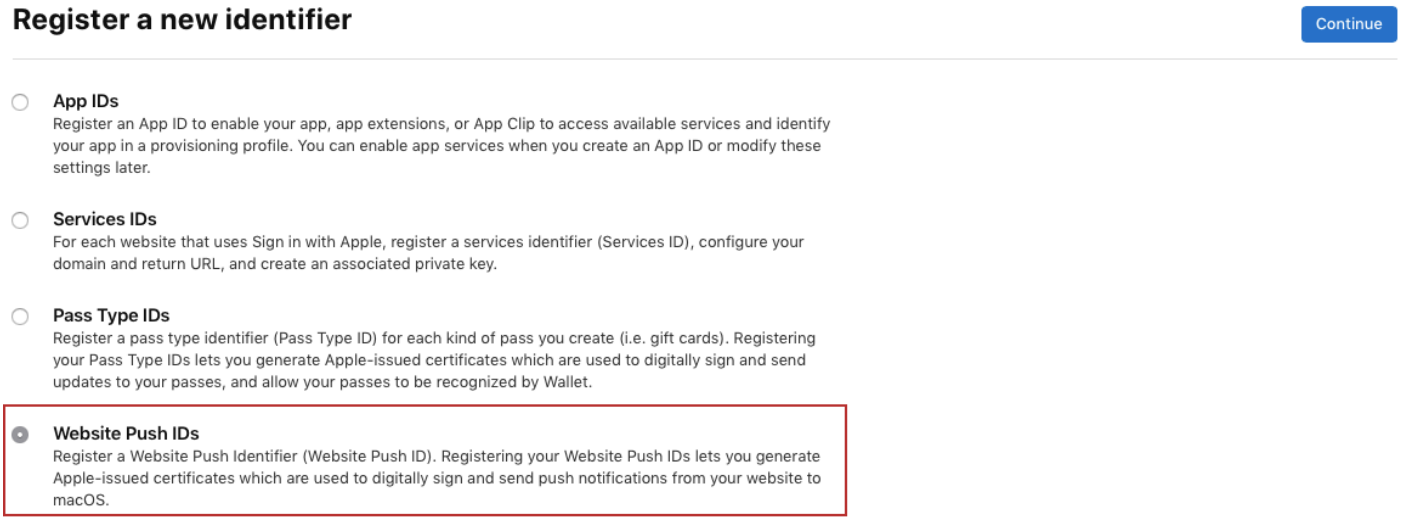
Шаг 1. Регистрация нового идентификатора

1. Перейдите по ссылке https://developer.apple.com/account/ios/certificate (**Certificates, Identifiers & Profiles**) и войдите в аккаунт разработчика Apple.

2. Слева в панели навигации выберите вкладку “**Identifiers**” и нажмите на значок **+** рядом с заголовком “Identifiers”:



3. В открывшемся окне выберите **Website Push IDs**, поскольку мы хотим отправлять пуш-уведомления на устройства macOS с веб-сайта:

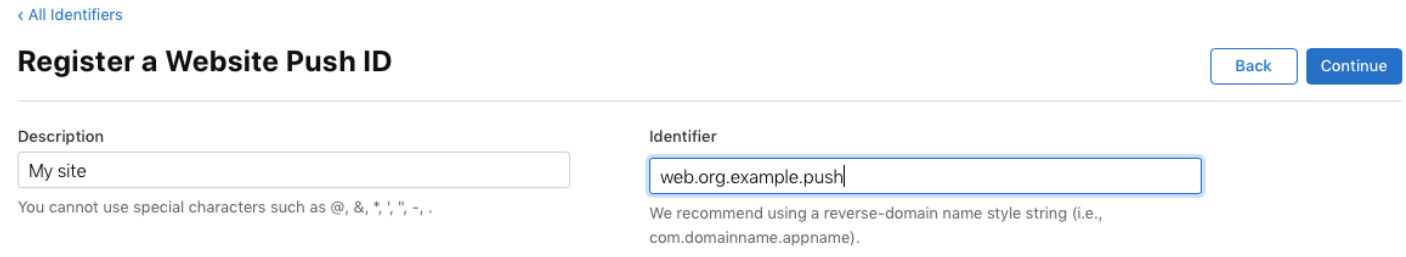


4. Введите **Description** (Описание) и **Identifier** (Идентификатор).

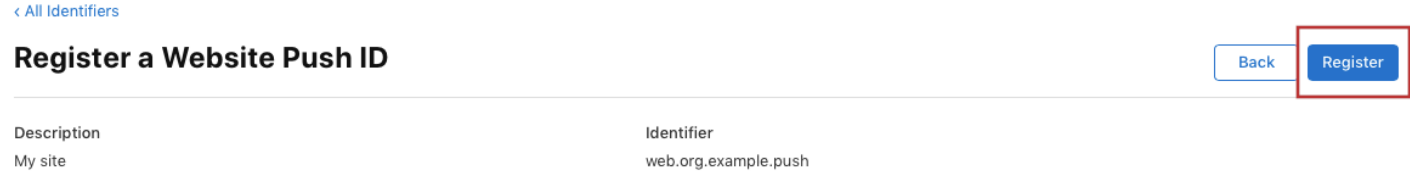
TIP

Идентификатор должен иметь формат `web.com.domainname.appname`.

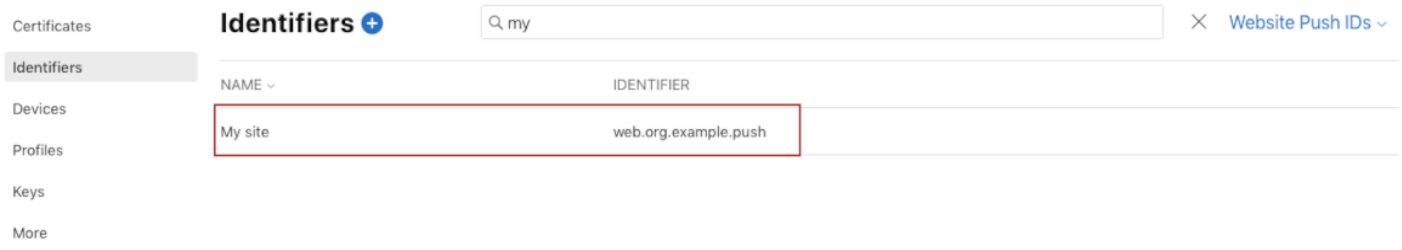
Важно! Первая часть всегда — **web**. Частей может быть сколько угодно, но в разумных пределах. Рекомендуется использовать стиль обратного домена, например, web.org.example.push.



5. Нажмите кнопки **Continue**, затем **Register**.

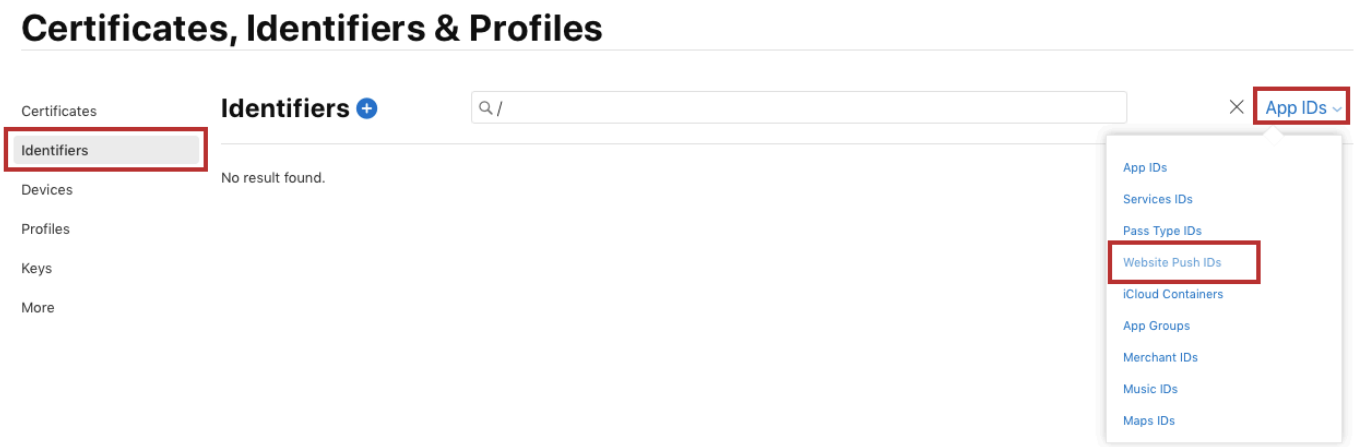


Итак, вы создали и зерегистрировали **Website Push ID**. Теперь вы можете в любой момент посмотреть информацию об этом идентификаторе на вкладке “**Identifiers**”:



TIP

Чтобы увидеть идентификаторы для рассылки push уведомлений именно с веб-сайтов, откройте выпадающий список в правом углу и выберите **Website Push IDs**.



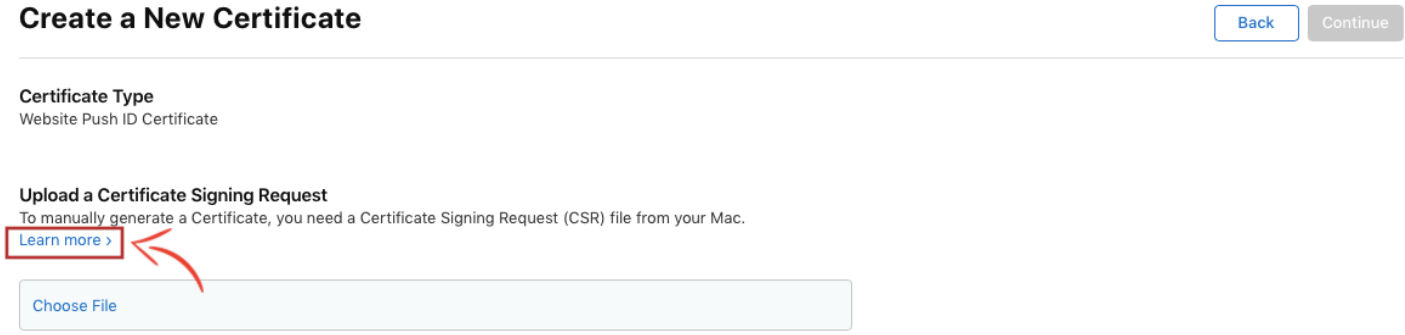
Шаг 2. Создание сертификата

1. Для каждого идентификатора требуется **Сертификат**. Найдите только что созданный **Web Push ID** в списке и кликните по нему.

2. Обратите внимание на поле “**Identifier**”. Информация из этого поля потребуется для настройки ресурса в платформе Altcraft, а именно — при заполнении поля **Website Push ID**.

3. Нажмите “**Create Certificate**”, чтобы создать сертификат:

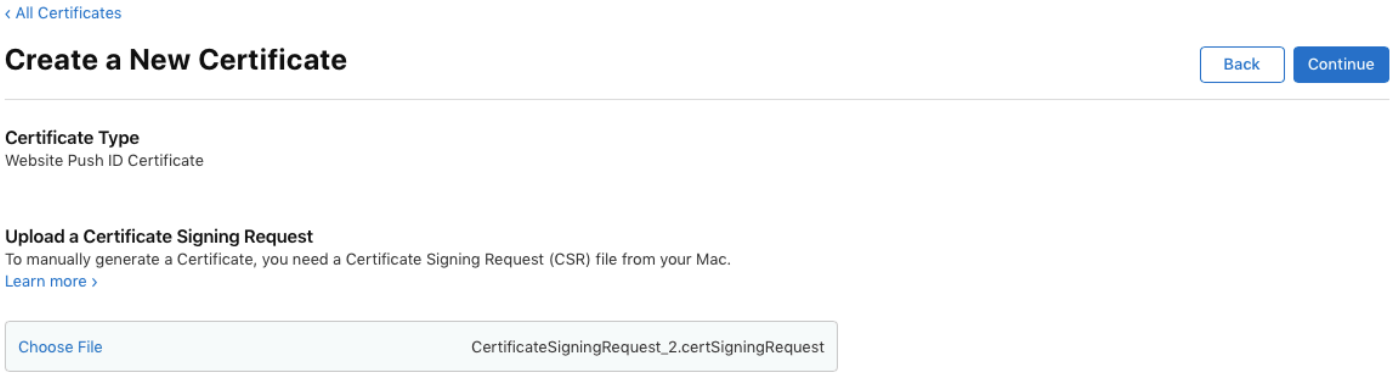
4. Теперь необходимо сгенерировать Сертификат. Для этого требуется **Certificate Signing Request** (CSR) (формат запроса — .**certSigningRequest**). Для создания запроса кликните **Learn more**:



5. Откроется новая страница. Следуйте инструкциям, приведённым на ней.

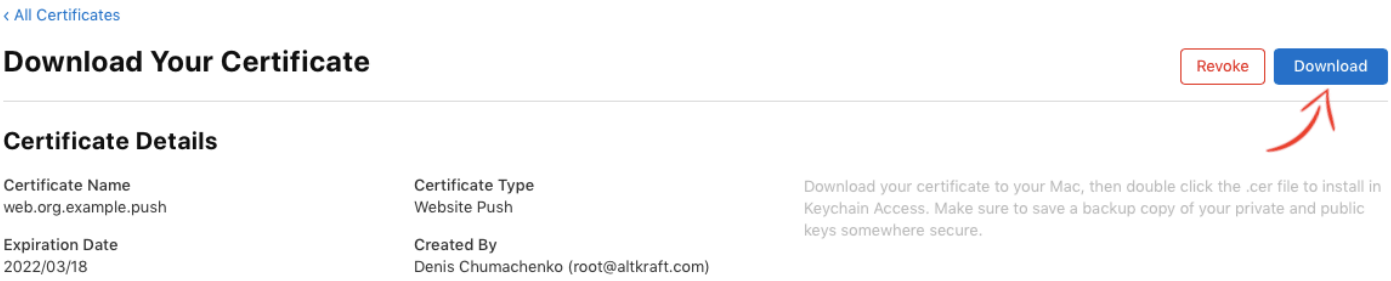
6. После создания **Certificate Signing Request** вернитесь на предыдущую страницу. Нажмите “**Choose File**” и выберите созданный **CSR.**

7. Нажмите “**Continue**”.

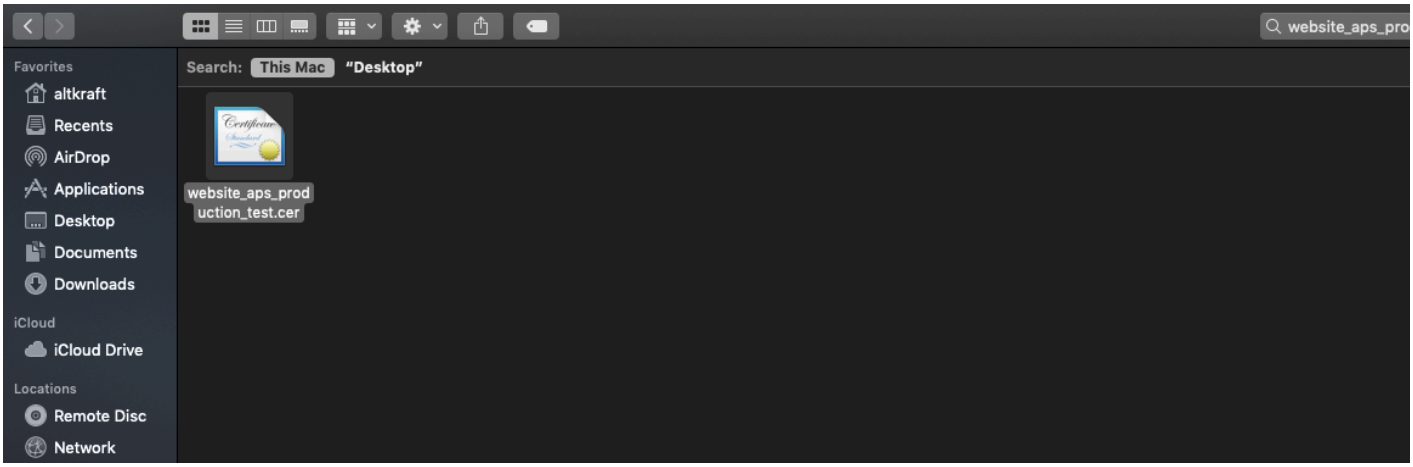


Шаг 3. Сохранение сертификата

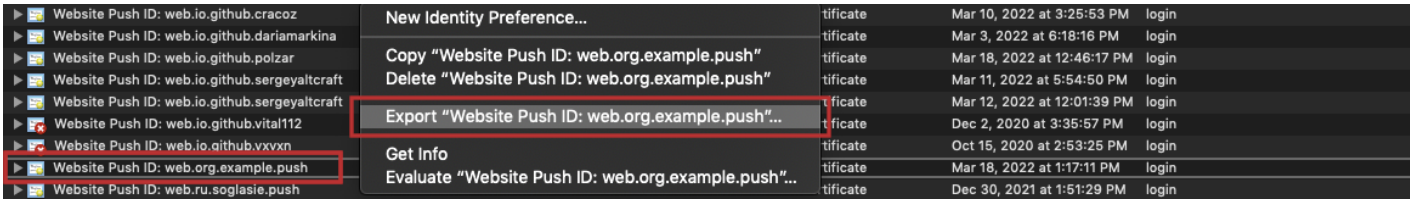
1. На следующей странице нажмите “**Download**”, чтобы скачать сертификат в формате **.cer** на ваш Mac:

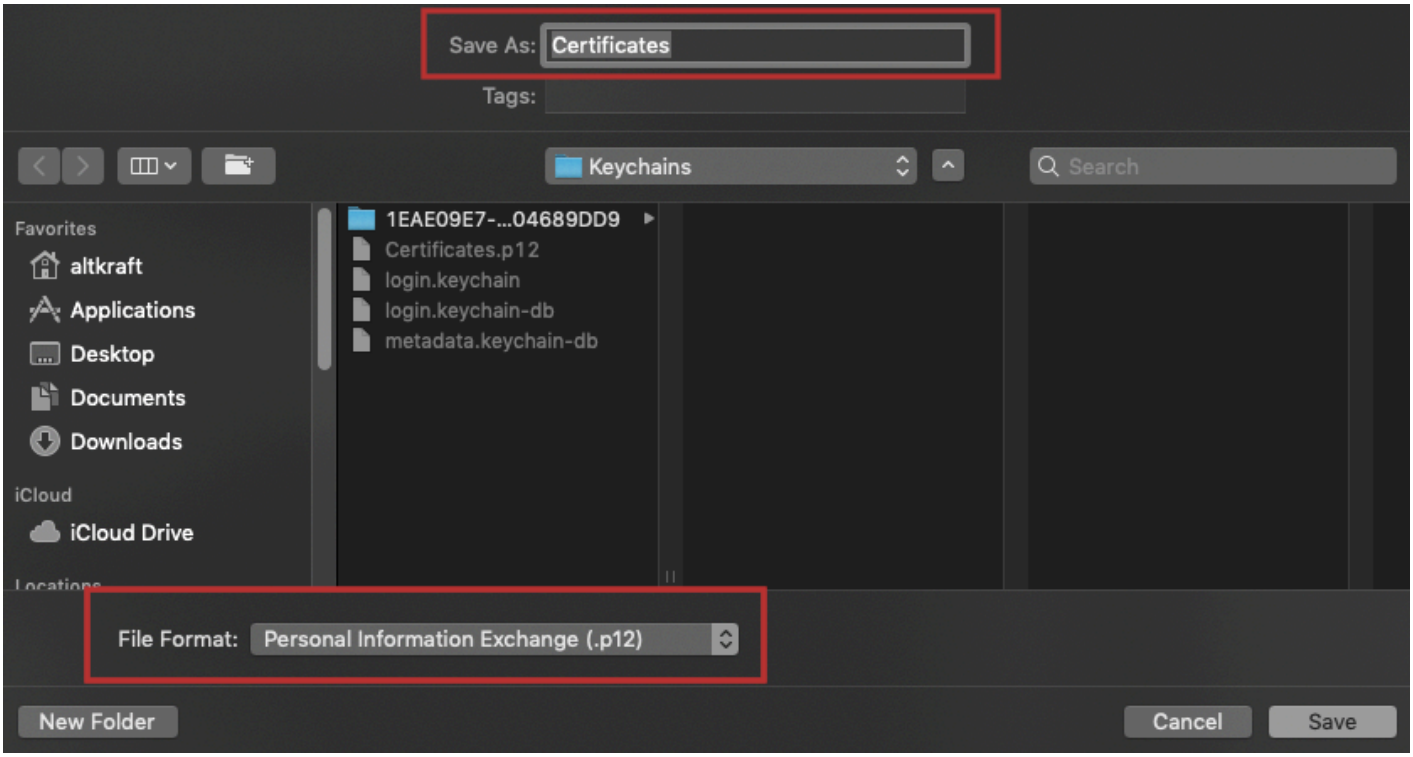


2. Найдите загруженный файл на вашем устройстве и дважды кликните по нему, чтобы установить его в **Keychain**:

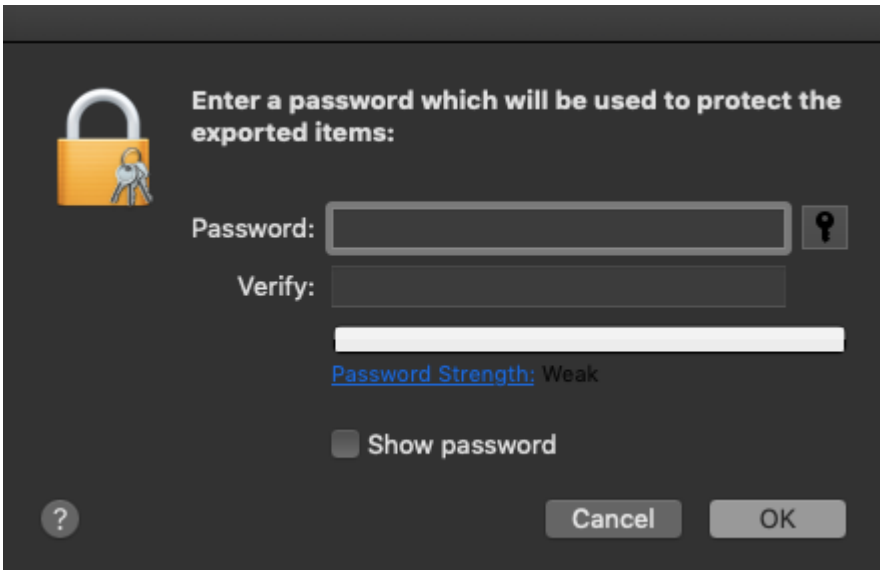


3. Найдите установленный сертификат в Keychain, кликните правой кнопкой мыши и экспортируйте его на диск в формате **.p12**:





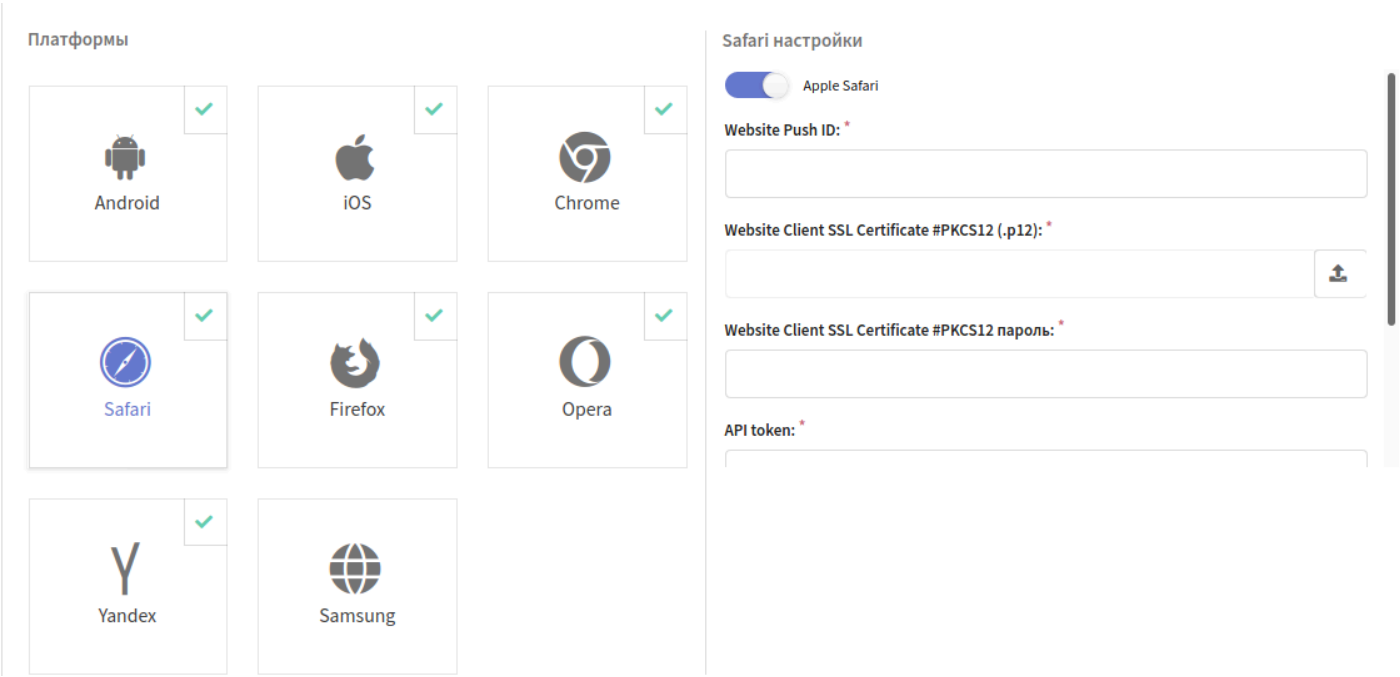
4. В процессе экспорта вам будет предложено ввести пароль. **Запомните этот пароль**, поскольку он потребуется для настройки push-рассылок в платформе Altcraft.



Шаг 4. Настройка ресурса в платформе Altcraft

1. Вернитесь в платформу Altcraft и откройте настройки ресурса, в рамках которого вы хотите настроить рассылку браузерных push-уведомлений. Перейдите в настройки push-канала.

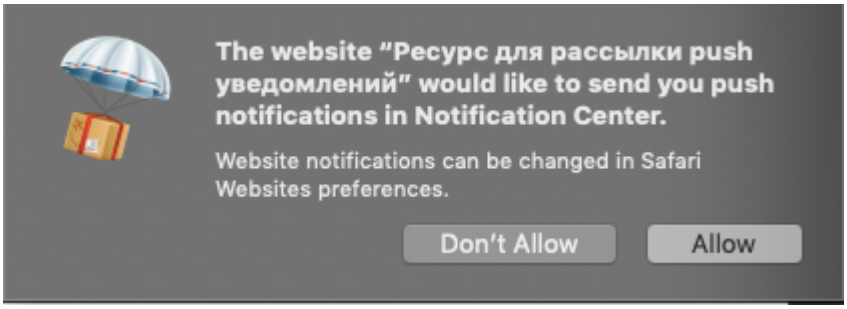
2. Выберите платформу **Safari** и активируете сервис **Apple Safari**:



3. Заполните следующие поля:

- **Website Push ID** — идентификатор в аккаунте разработчика Apple. Это значение берется из поля “Identifier” в вашем Apple Developer аккаунте.
- **Website Client SSL Certificate** — файл сертификата с расширением .p12. Этот сертификат экспортируется из Keychain Access на вашем Mac-устройстве.
- **Website Client SSL Certificate пароль** — пароль, использованный при экспорте сертификата .p12
- **API token** — токен платформы. Позволяет APNS отправлять запросы на подписку в платформу. Создать токен можно в разделе **Настройки —> Токены**. Если токен удалить, платформа перестанет принимать подписки от Safari для данного ресурса.

- **Иконка** — изображение в формате .png, которое будет показано в push-уведомлении. Иконка помогает пользователю идентифицировать источник уведомления. Рекомендуется использовать размер **256x256** пикселей или больше. Safari поддерживает только одно изображение для ресурса (веб-сайта).



TIP

Следующий шаг: **Подключение Web push на сайт**

Firebase Cloud Messaging

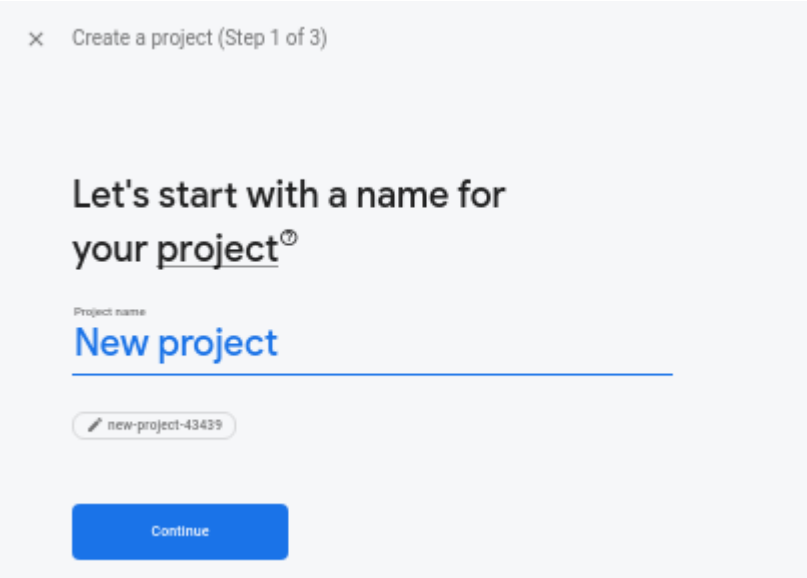
TIP

Видео-гайд по настройке Firebase-провайдера в push-канале ресурса доступен по **ссылке**.

Шаг 1. Создание проекта Firebase

	Для отправки Web Push в браузеры **Mozilla Firefox**, **Opera**, **Google Chrome, Яндекс.Браузер, Samsung Internet Browser** необходимо предварительно настроить проект в Firebase Console.

Перейдите по ссылке, создайте учётную запись в Google Firebase и зарегистрируйте свой сайт или приложение. Для этого нажмите на кнопку **Add project** (Добавить проект) и укажите название проекта:



TIP

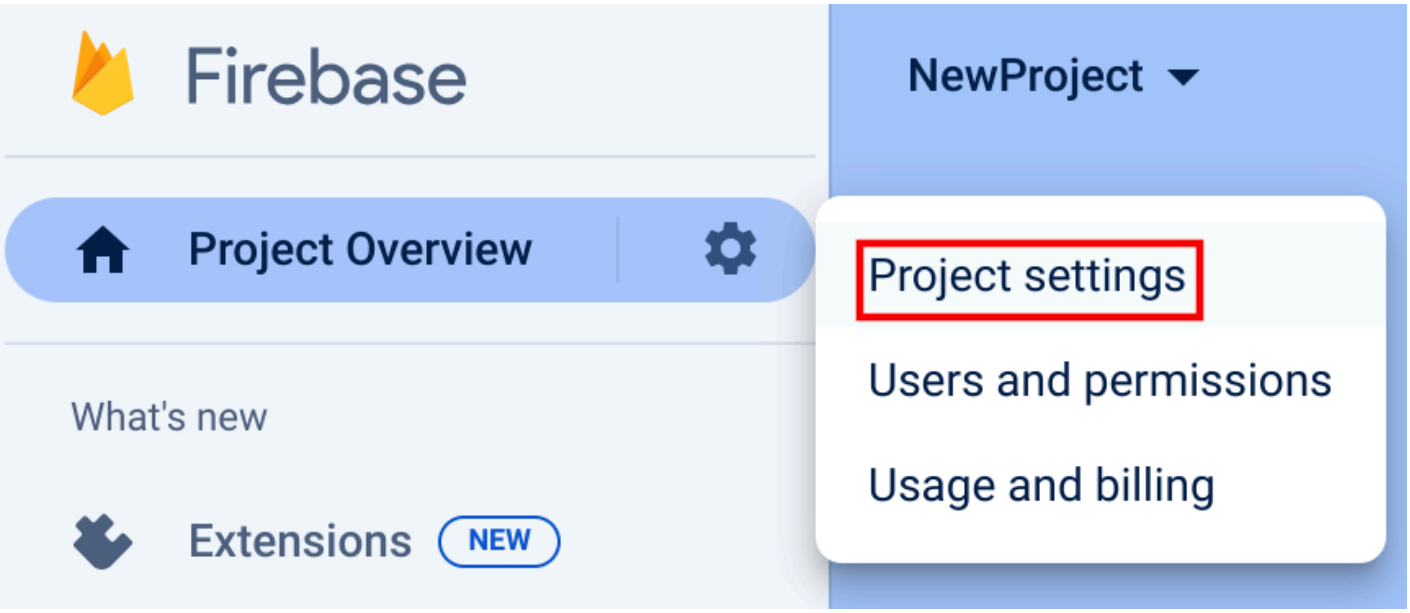
При создании проекта вы можете настроить Google Analytics, а также добавить Firebase в ваше веб-приложение, но это не является обязательными действиями.

Шаг 2. Получение данных проекта

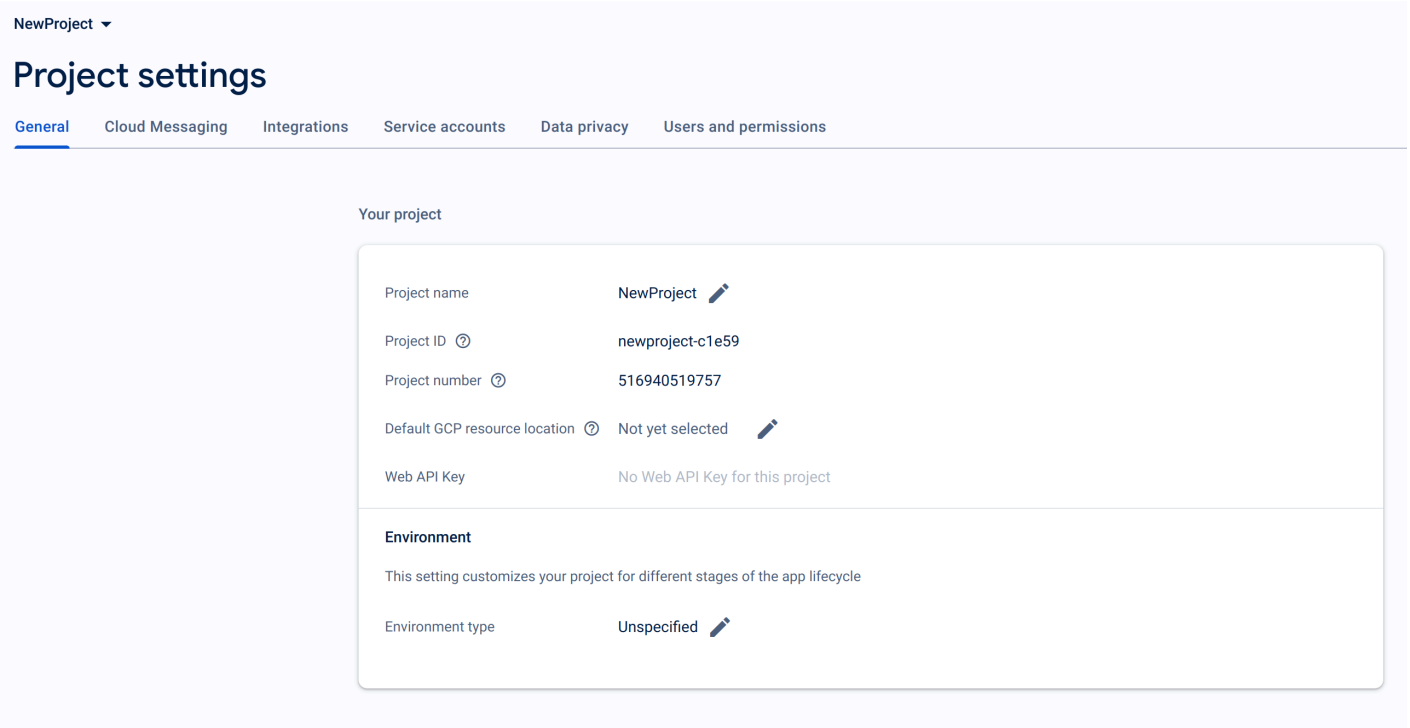
Для настройки Firebase Cloud Messaging в платформе Altcraft вам потребуется следующая информация:

- **Ключ API для веб-приложений**
- **Идентификатор проекта firebase**
- **Идентификатор отправителя**
- **Идентификатор приложения App ID**
- **Firebase файл с приватным ключом**

Данную информацию вы можете найти в интерфейсе Firebase Console. Для этого в навигационной панели слева нажмите на иконку “Настройки” и выберите **Project settings** (Настройки проекта):

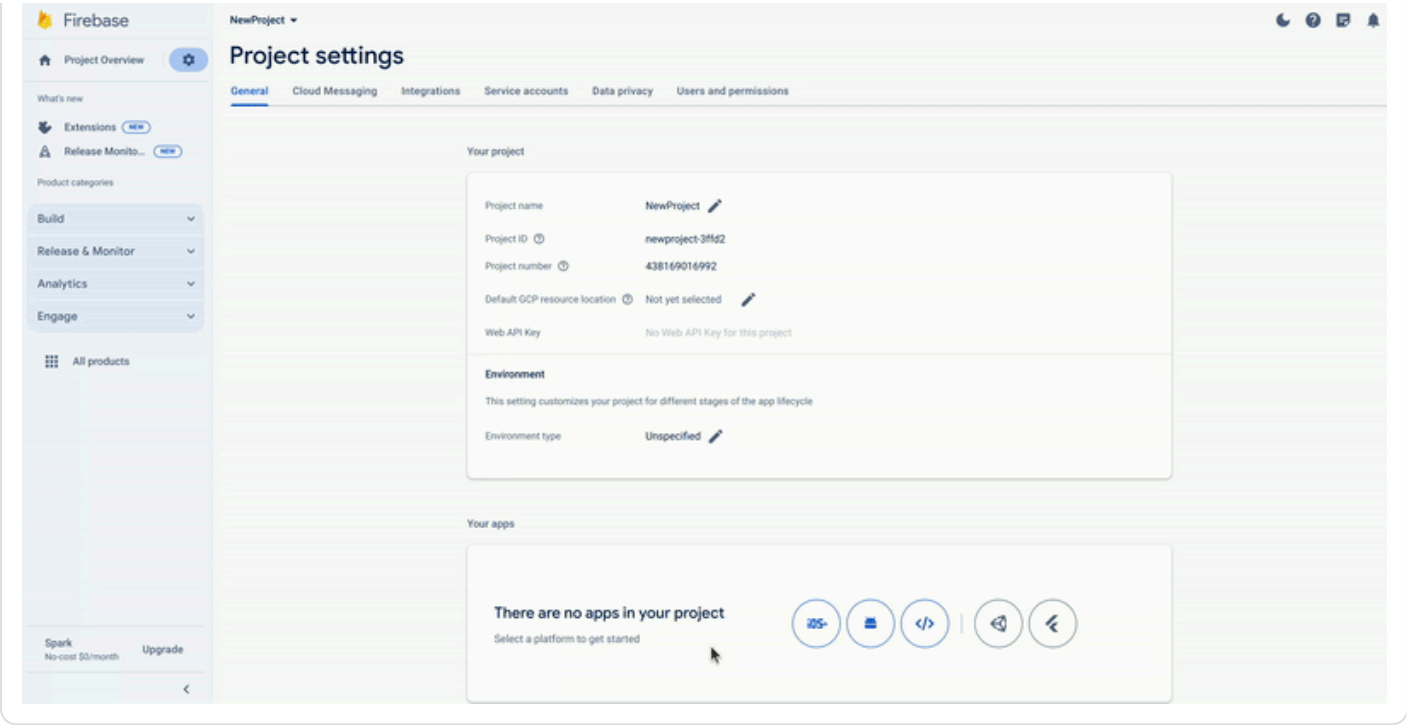


Идентификатор проекта firebase (Project ID) и **Ключ API для веб-приложения** (Web API key) находятся в разделе **Общие настройки** (General):

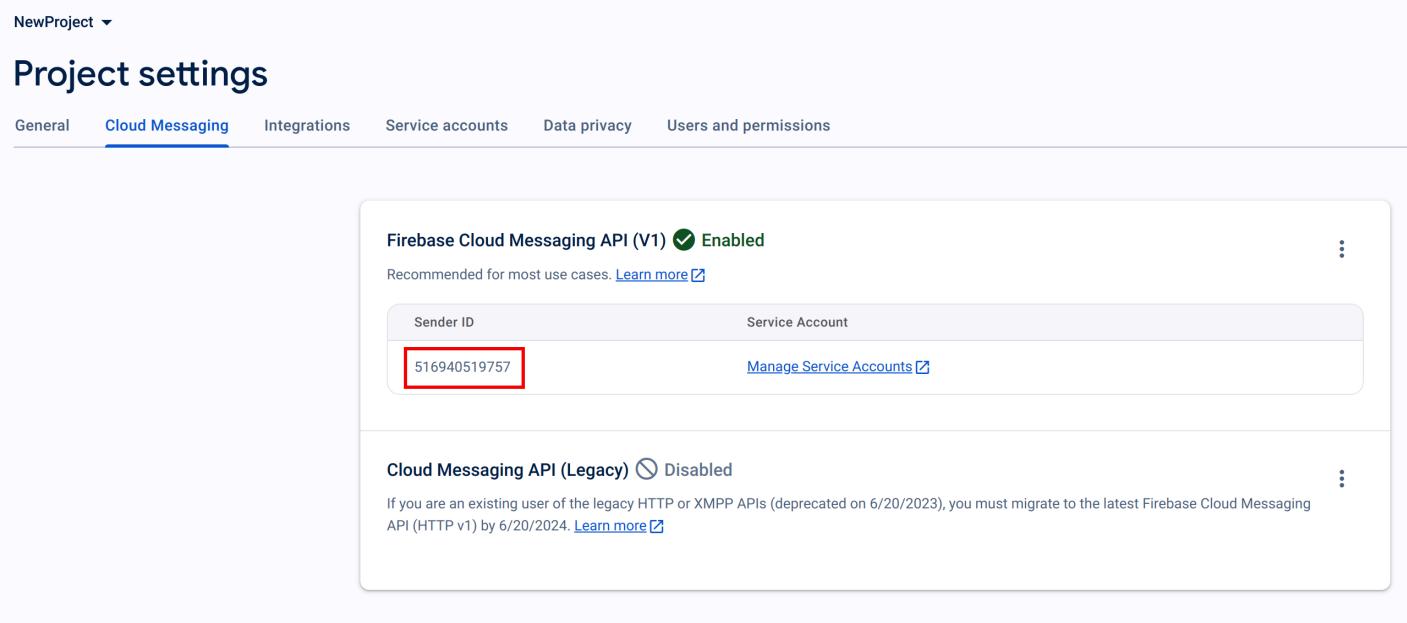


TIP

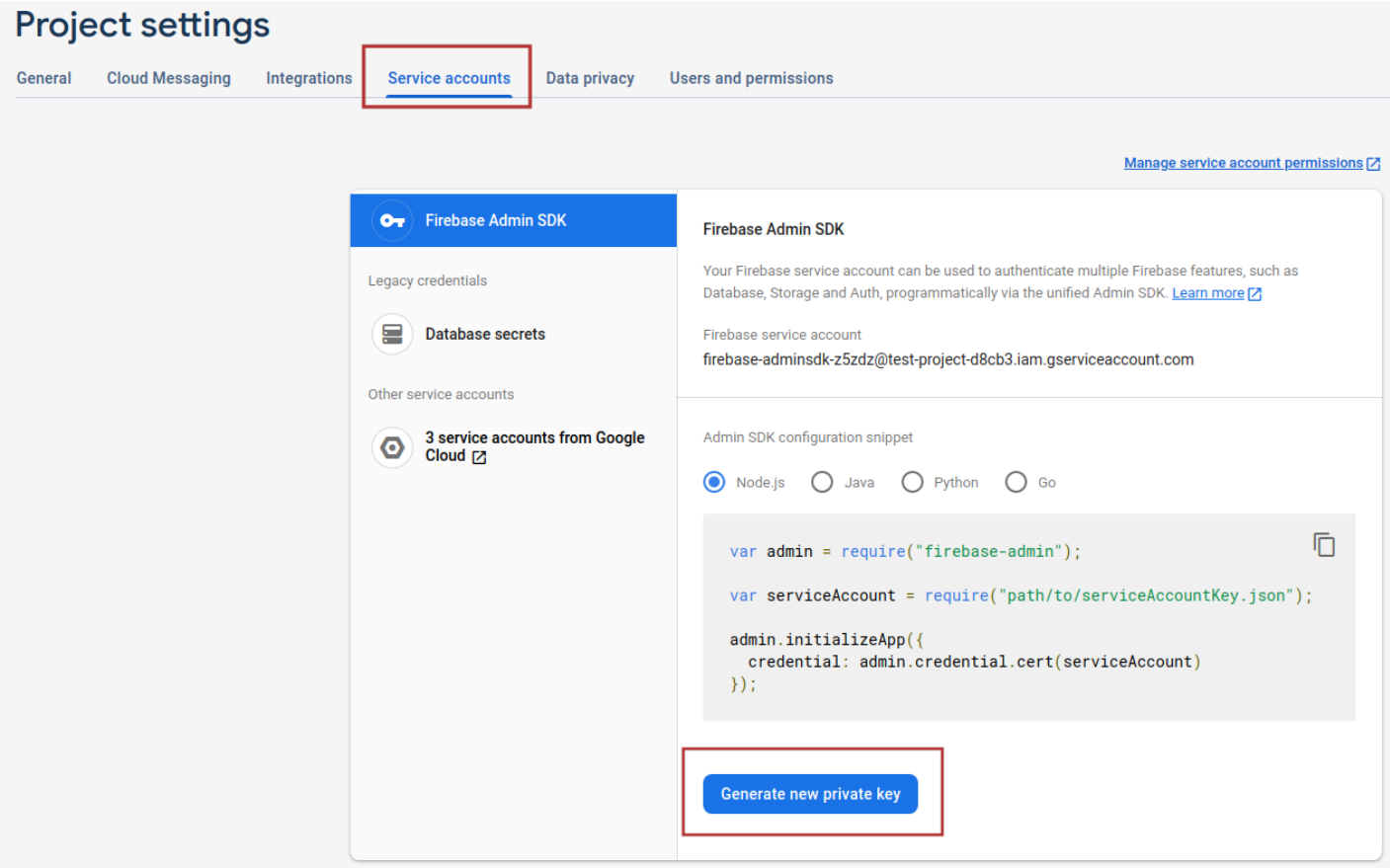
Если у вас не отображается Web API Key, перейдите в раздел **Build** и выберите пункт **Authentication** в выпадающем меню. Нажмите на кнопку **Get started**, которая появится на экране. Теперь вы можете вернуться в настройки, нажав на соответствующую иконку в меню слева.



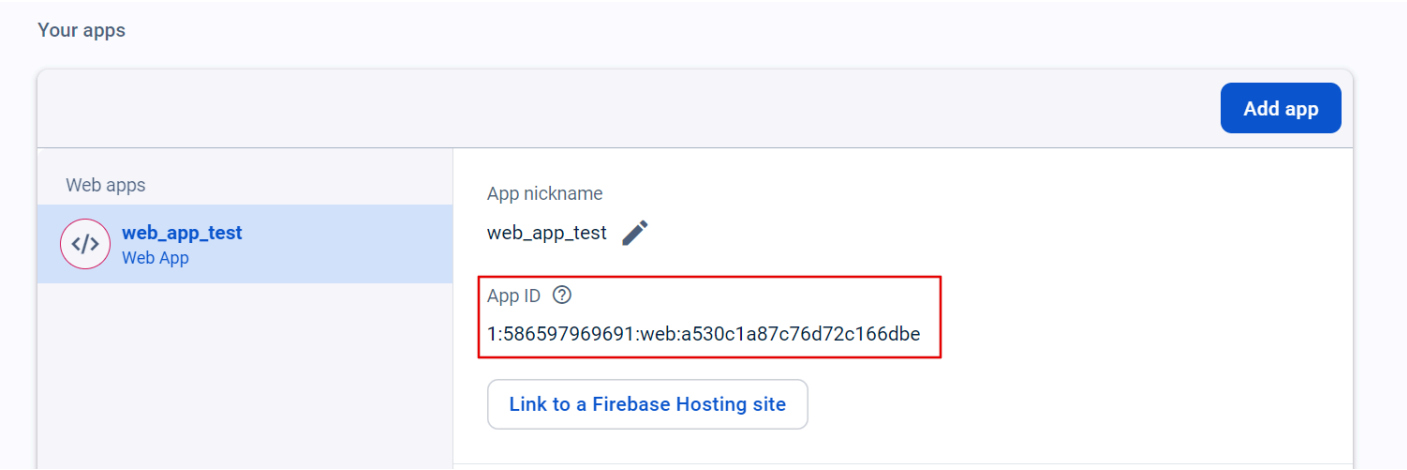
Идентификатор отправителя (Sender ID) находится в разделе **Cloud Messaging**:



Для настройки Firebase на платформе нам также понадобится **firebase-файл с приватным ключом**. Чтобы его скачать, перейдите на вкладку **Service accounts**. Кликните **Generate new private key**. Начнется скачивание файла.

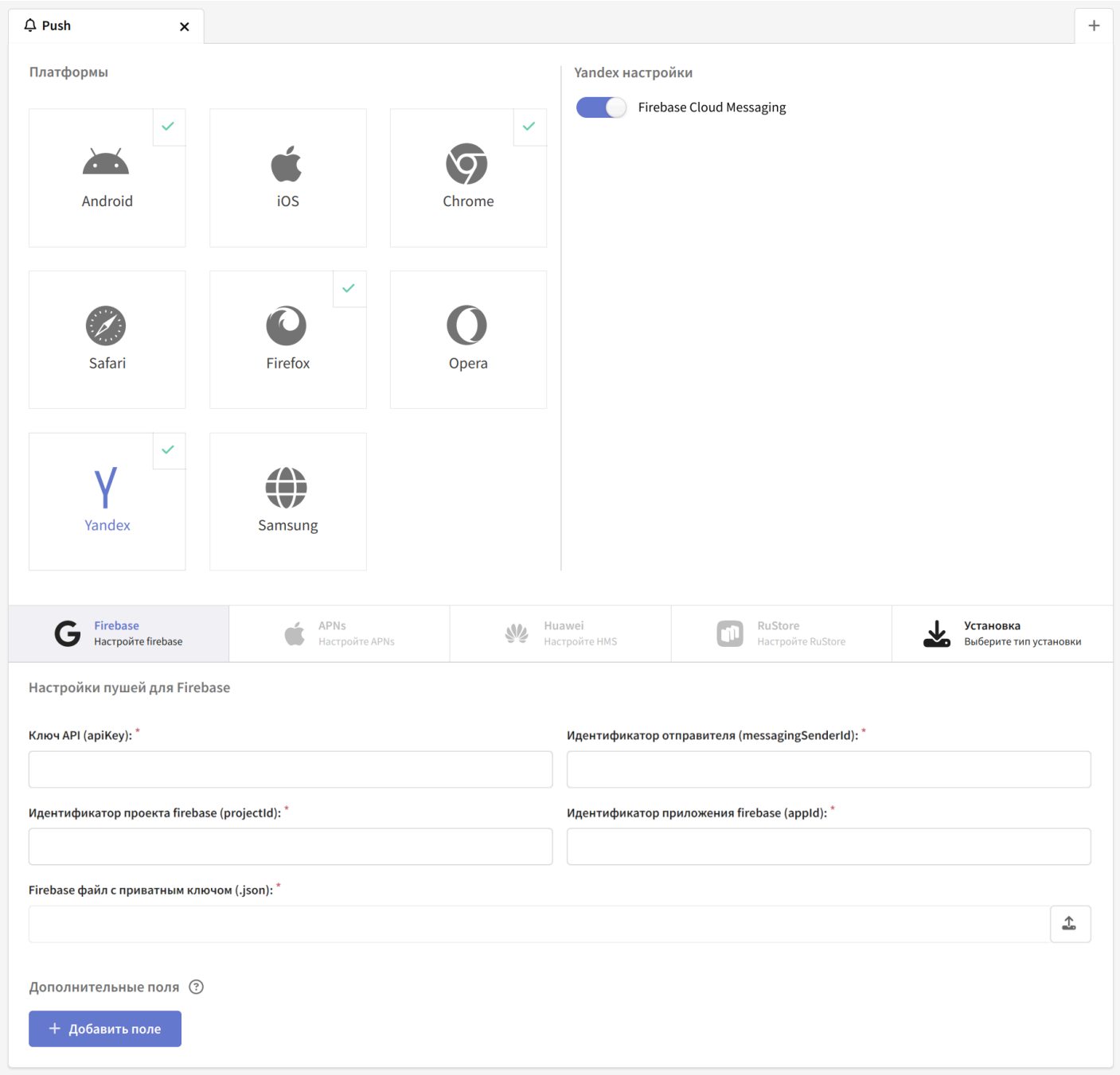


App ID можно найти в разделе Your Apps в Firebase Console:



Шаг 3. Настройка push-канала в ресурсе

Вернитесь в платформу Altcraft и откройте настройки ресурса. Настройте Firebase Cloud Messaging для поддерживаемых браузеров, используя данные из Firebase Console. Для поля **“Firebase файл с приватным ключом (.json)”** выберите скачанный файл из предыдущего шага:



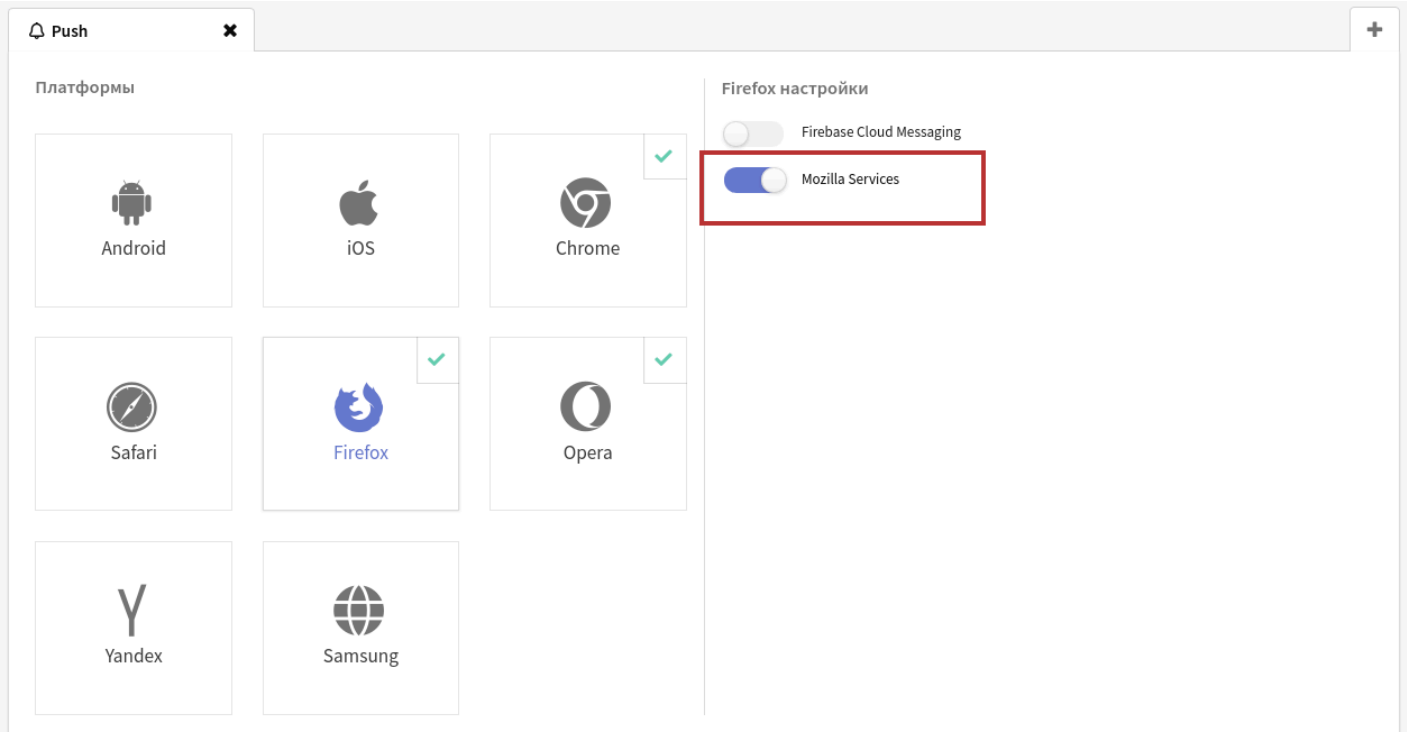
Сохраните ресурс или **примените** изменения, используя кнопки в правом верхнем углу экрана. Только после этого перейдите на вкладку **“Установка”**.

TIP

Следующий шаг: [Подключение Web push на сайт]

Mozilla Services

	Для отправки push сообщений в Mozilla Firefox достаточно активировать **"Mozilla services"** в блоке **"Firefox настройки"**.



TIP

Следующий шаг: Подключение Web push на сайт

Перенос push-подписок из стороннего сервиса

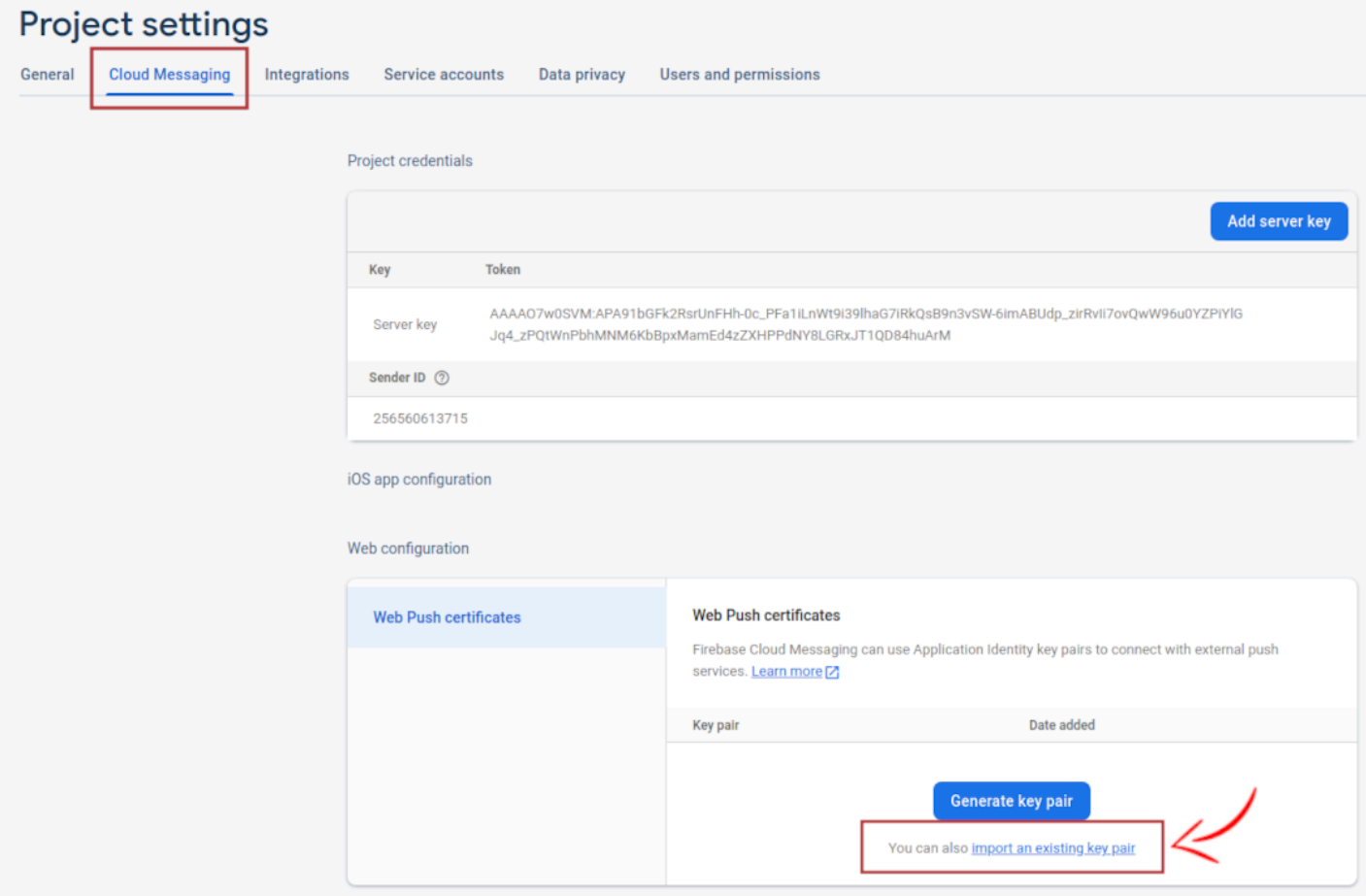
Шаг 1. Получение VAPID-ключей у текущего сервиса

Запросите VAPID-ключи у технической поддержки сервиса, с которого предполагается миграция. Необходимы как приватный, так и публичный ключи: **private key**, **public key**.

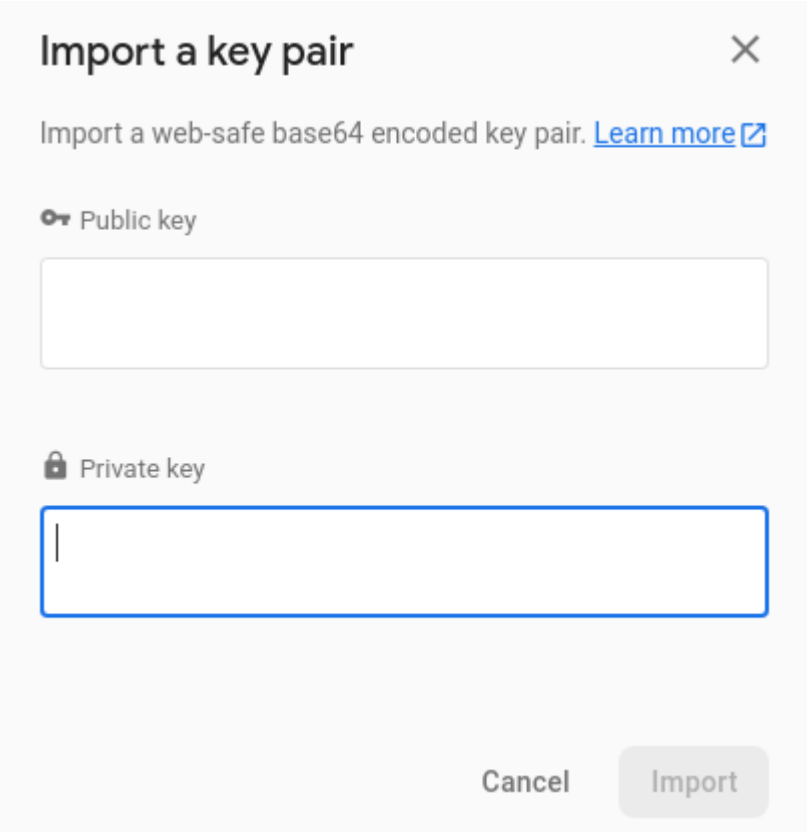
> **TIP**
>
> Обратите внимание, что **VAPID keys** и **Application Server keys** — это одно и то же.

Шаг 2. Импорт VAPID-ключей в проект Firebase

Откройте настройки проекта в Firebase и перейдите на вкладку **Cloud Messaging**. Внизу страницы вы увидите раздел **Web Configuration**. Нажмите **import an existing key pair**:



В появившемся окне введите **Public key** и **Private key**, которые ранее предоставил вам сервис.



Шаг 3. Настройка ресурса

О настройке ресурса для Web push уведомлений читайте в разделе Web push: настройка и подключение.

TIP

Обратите внимание, что настройка ресурса отличаются у разных браузеров.

Шаг 4. Создание CSV-файла с подписками

При подписке нового пользователя на push-уведомления создается объект `PushSubscription`. Именно в нем содержится вся информация, которая необходима для последующей отправки push-уведомлений этому пользователю. Объект хранится в JSON-формате:

```
{
"endpoint": "https://some.pushservice.com/something-unique",
"keys": {
        "p256dh":
"BIPUL12DLfytvTajnryr2PRdAgXS3HGKiLqndGcJGabyhHheJYlNGCeXl1dn18gSJ1WAkAPIxr4gK0_dQds4yiI=",
        "auth": "FPssNDTKnInHVndSTdbKFw=="
        }
}
```

Вам необходимо выгрузить эти данные из стороннего сервиса в CSV-файл. Для каждой подписки должны быть указаны поля `endpoint`, `p256dh`, `auth`.

Шаг 5. Подготовка CSV-файла к импорту

Если платформа Altcraft установлена в контур вашей компании, то передайте созданный CSV-файл с подписками администратору платформы.

Если платформа установлена в облаке, то передайте CSV-файл в службу поддержки Altcraft.

Администратор (или служба поддержки) с помощью утилиты должны привести данные к формату, совместимому с импортом в платформу. После обработки вы получите готовый к импорту CSV-файл.

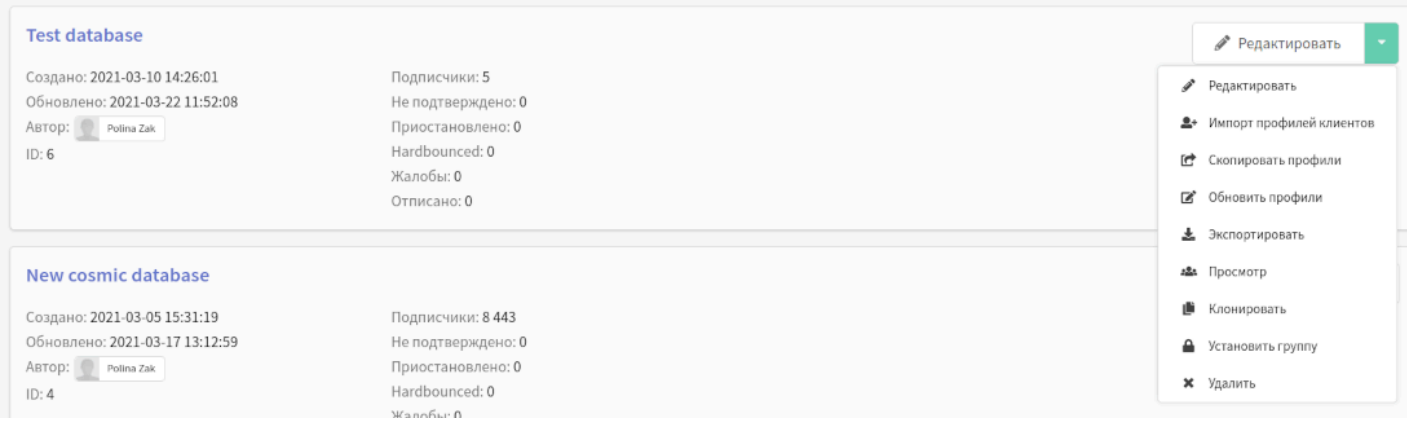
Шаг 6. Импорт профилей с подписками

Теперь вы можете импортировать профили в **Базы профилей** из CSV-файла, при этом **создавая подписку** на нужный провайдер.

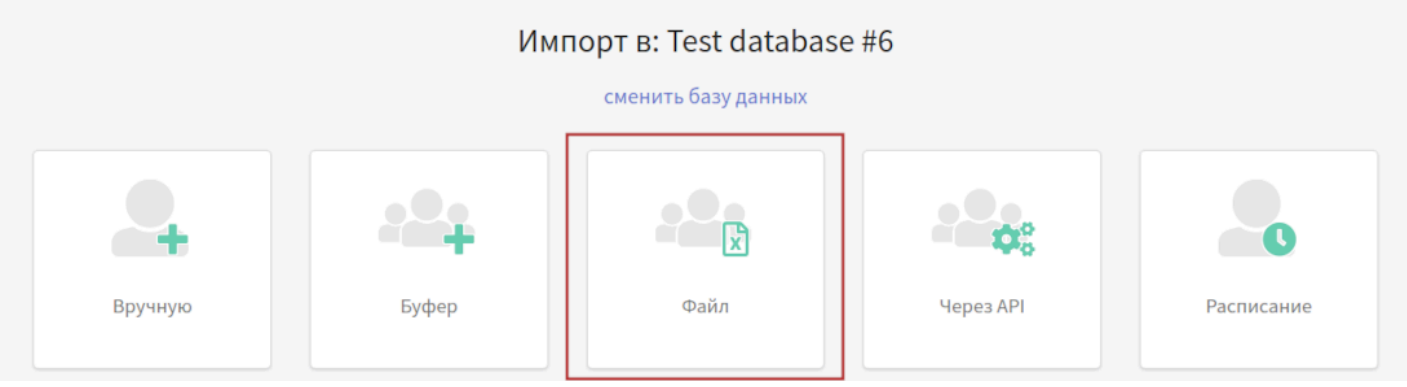
Кроме того, у вас есть возможность выполнить **обновление базы** с созданием подписки на этот же провайдер. Обратите внимание, в данном случае это обязательно должен быть один из Firebase-провайдеров.

> **TIP**
>
> Идентификатор провайдера позволяет узнать, к какому именно браузеру относится подписка и через какой сервис она создавалась. Примеры провайдеров: Chrome Firebase, Apple Safari, Mozilla Firefox.

Итак, создайте или выберите подходящую базу и в выпадающем списке кликните **Импорт профилей клиентов**.



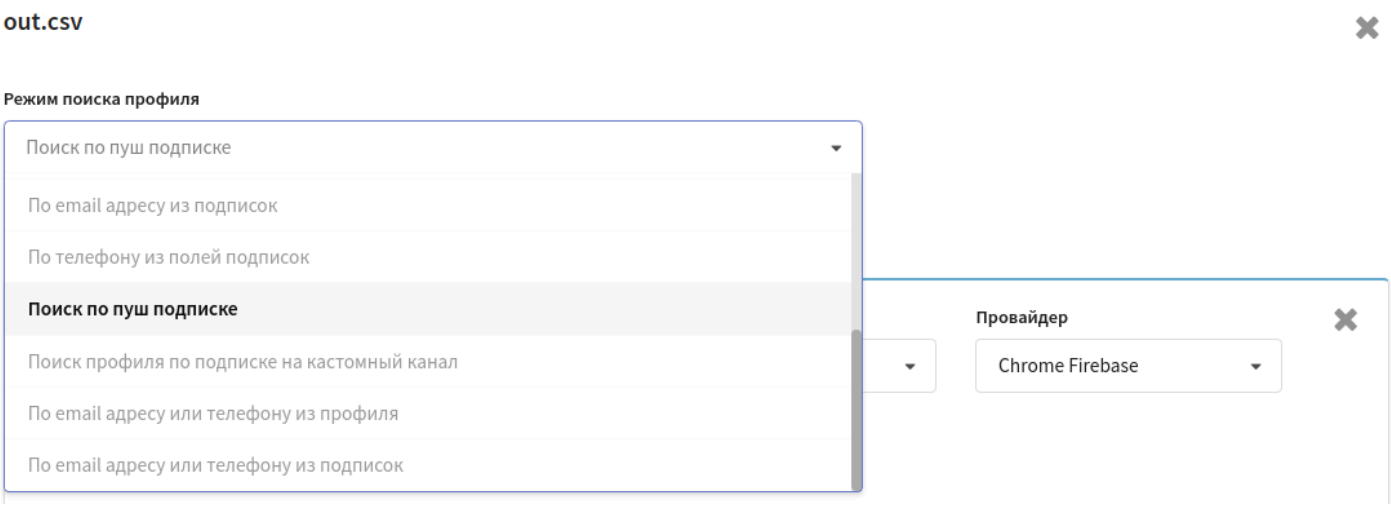
Выберите способ импорта — **Файл.** Загрузите необходимый CSV-файл с подписками.



Кликните **Режим поиска подписки** и в выпадающем списке выберите **Поиск по пуш подписке**.

CAUTION

Если в файле есть данные, по которым можно сопоставить существующих пользователей (например, email, phone), выберите соответствующий режим поиска профиля.

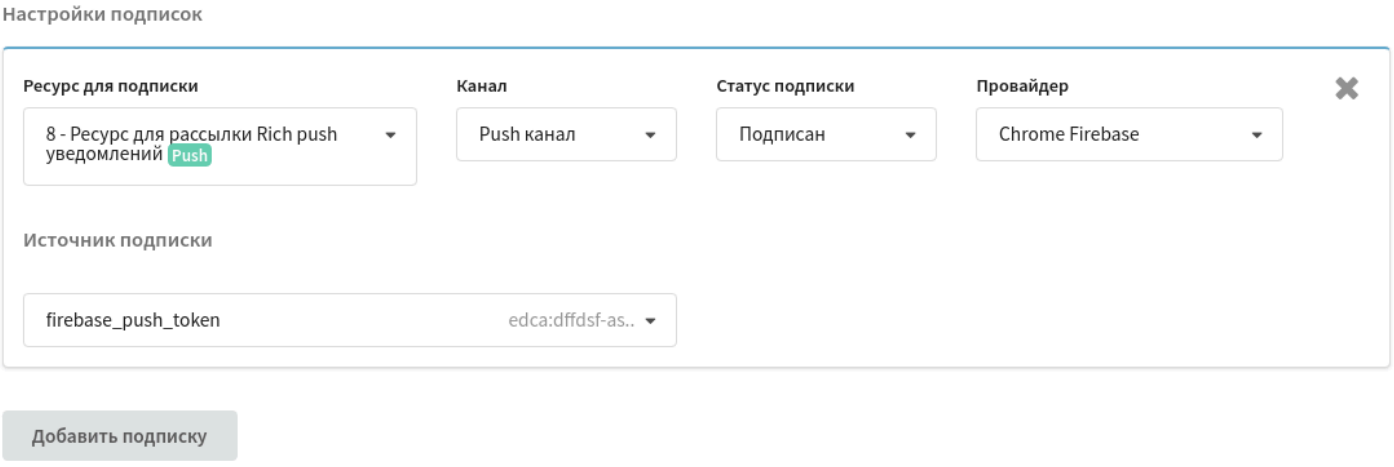


Нажмите на кнопку **Добавить подписку**. Затем настройте подписку:

- Канал — **Push канал**;
- Провайдер — если бразуер подписки неизвестен, то можно импортировать подписки с провайдером Chrome Firebase;
- Источник подписки — **firebase_push_token**.

CAUTION

Для создания подписки необходимо название поля, где хранится id той подписки, которую вы импортируете. В вашем CSV-файле такое поле будет называться `firebase_push_token`.



Теперь необходимо настроить **соответствие колонок**:

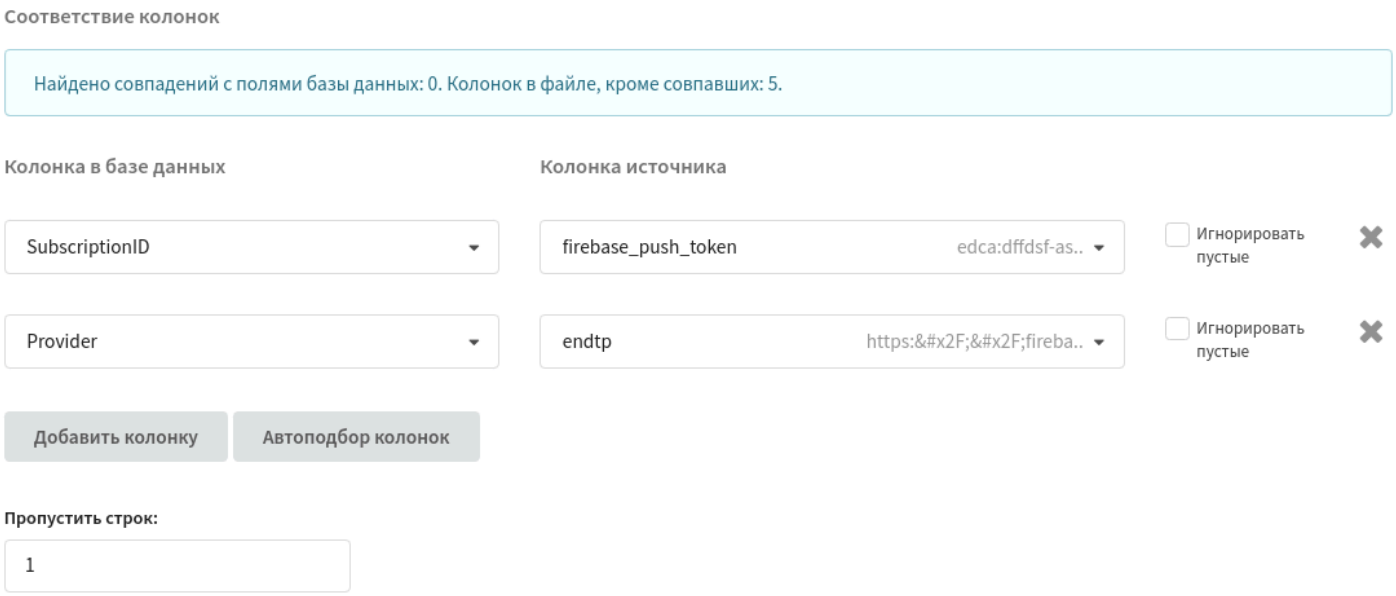
- **SubscriptionID** — **firebase_push_token**;
- **Provider** — поле идентификатора провайдера Altcraft.

TIP

Если в csv файле отсутствует поле идентификатора провайдера Altcraft, то укажите то же поле, что и в SubscriptionID. Таким образом для каждой подписки будет создан новый подписчик.

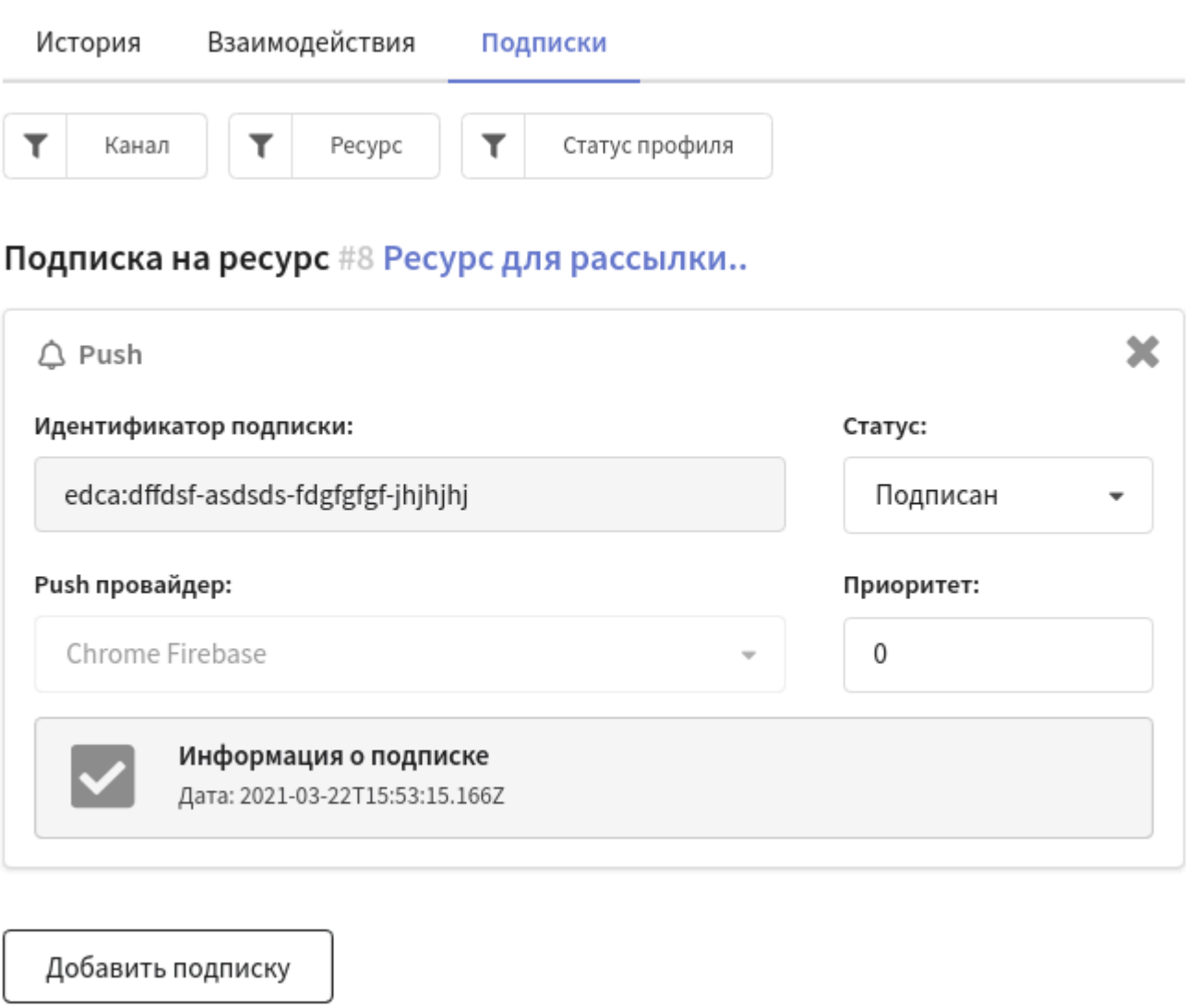
CAUTION

В поле **Пропустить строк** укажите значение “1”, если в файле есть заголовки. В противном случае, при импорте будут захватываться заголовки.



После этого нажмите **Импортировать**. Профили будут добавлены в базу, а подписки — привязаны к соответствующим провайдерам.

Как только профили будут успешно импортированы, вы можете посмотреть их в выбранной вами Базе профилей. Открыв вкладку **Подписки**, вы увидите новую подписку профиля на push-уведомления и информацию о ней.



TIP

Если вы хотите осуществить миграцию с сервиса OneSignal, изучите эту статью: Миграция с OneSignal.

Как перенести push-подписки, настроенные для Safari

CAUTION

Если вы хотите перенести push-подписки, которые настроены для браузера Safari, вам необходимо иметь расширение **Developer в Apple аккаунте**. Кроме того, перенос возможен только в случае, если они создавались с сертификатом и ключом из этого аккаунта.

Шаг 1. Настройка ресурса

О настройке ресурса для Web push уведомлений читайте в разделе Web push: настройка и подключение.

Шаг 2. Создание csv файла с подписками

Выгрузите идентификаторы Safari подписок из внешнего сервиса в формате **csv файла**.

Шаг 3. Импорт профилей

Теперь вы можете импортировать профили в **Базы профилей** из csv файла, при этом **создавая подписку** на нужный провайдер.

Кроме того, у вас есть возможность выполнить **обновление базы** с созданием подписки на этот же провайдер.

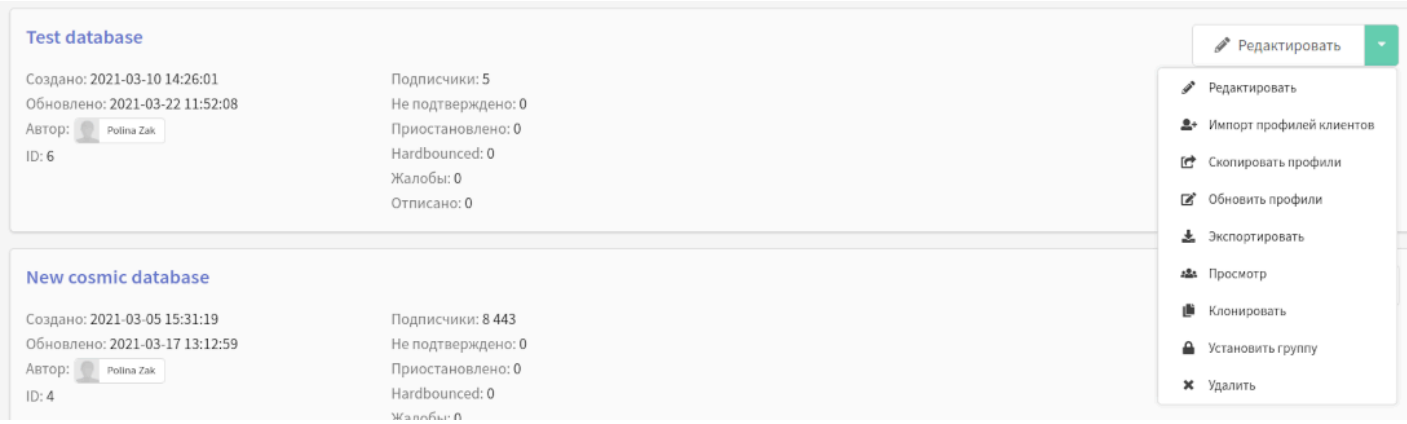
TIP

Идентификатор провайдера позволяет узнать, к какому именно браузеру относится подписка и через какой сервис она создавалась.

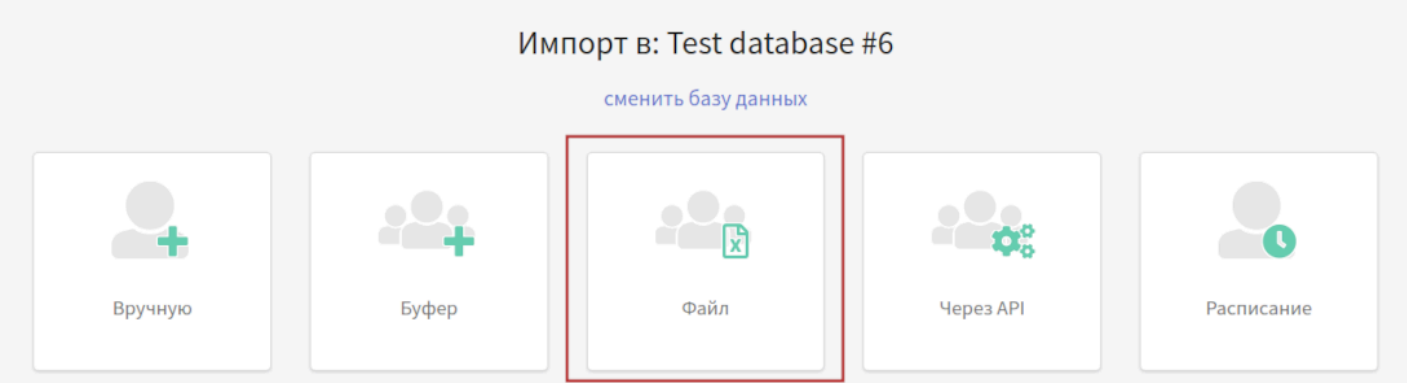
CAUTION

Обратите внимание, в данном случае это обязательно должен быть один из Firebase провайдеров.

Итак, создайте или выберите подходящую базу и в выпадающем списке кликните **Импорт профилей клиентов**.



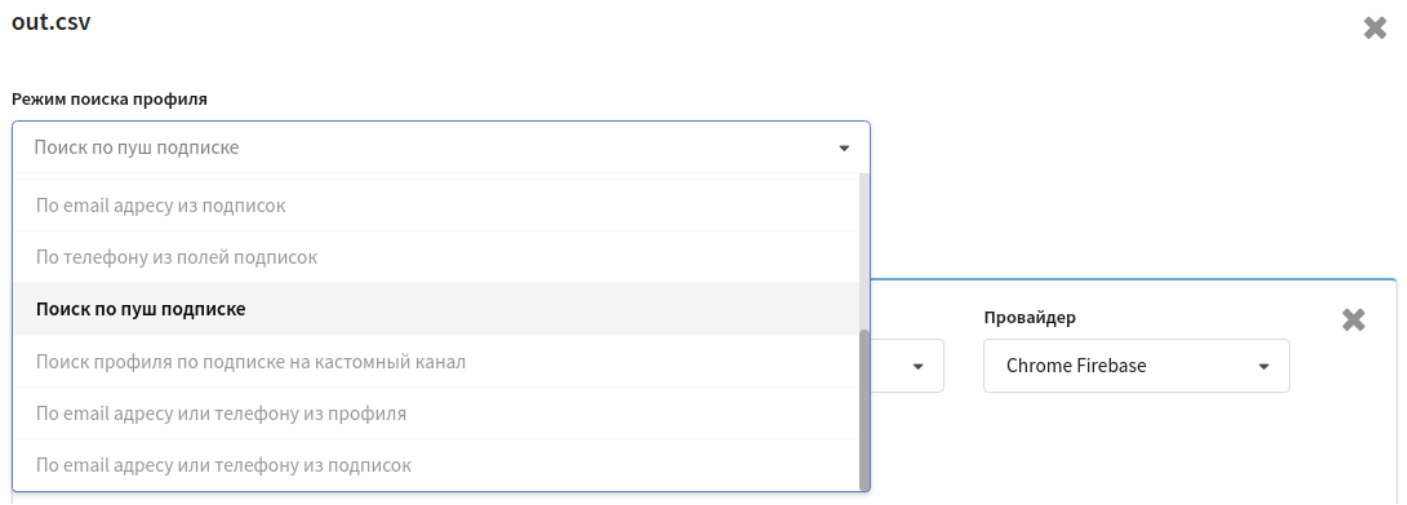
Выберите способ импорта — **Файл**. Загрузите необходимый csv файл с подписками.



Кликните **Режим поиска подписки** и в выпадающем списке выберите **Поиск по пуш подписке**.

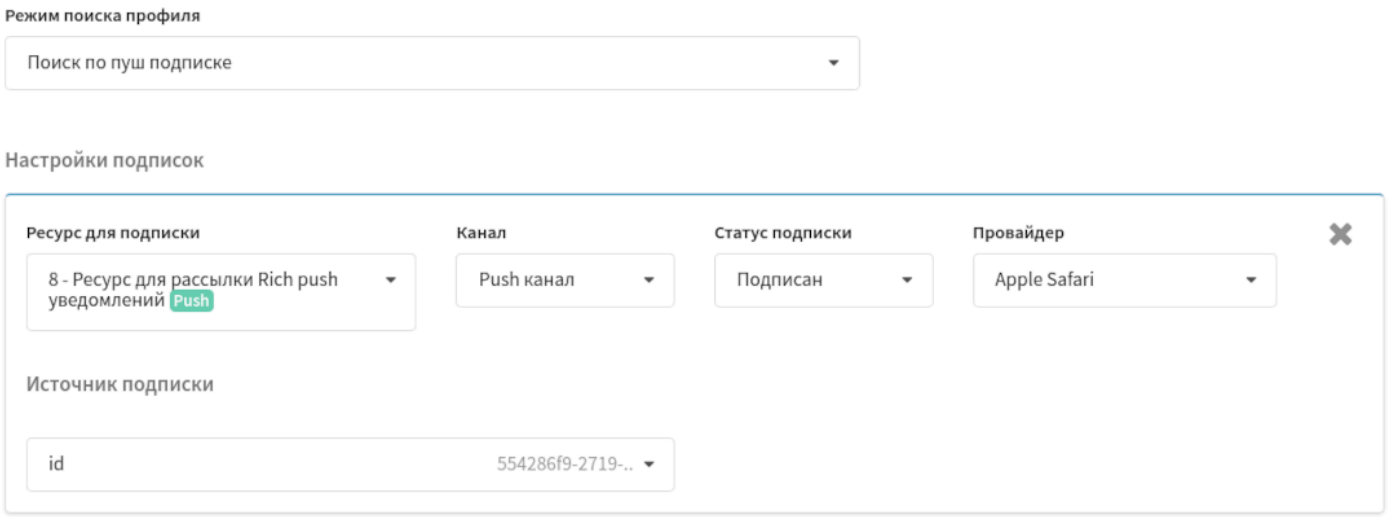
CAUTION

Если в вашем csv файле есть какие-то данные, которые можно использовать, чтобы настроить соответствия между профилями (“сматчить пользователей”), то при импорте выберите соответсвующий режим поиска профиля.



Нажмите на кнопку **Добавить подписку**. Затем настройте подписку:

- Канал — **Push канал**;
- Провайдер — **Apple Safari**;
- Источник подписки — здесь необходимо указать поле идентификатора подписки из файла.



Теперь необходимо настроить **соответствие колонок**:

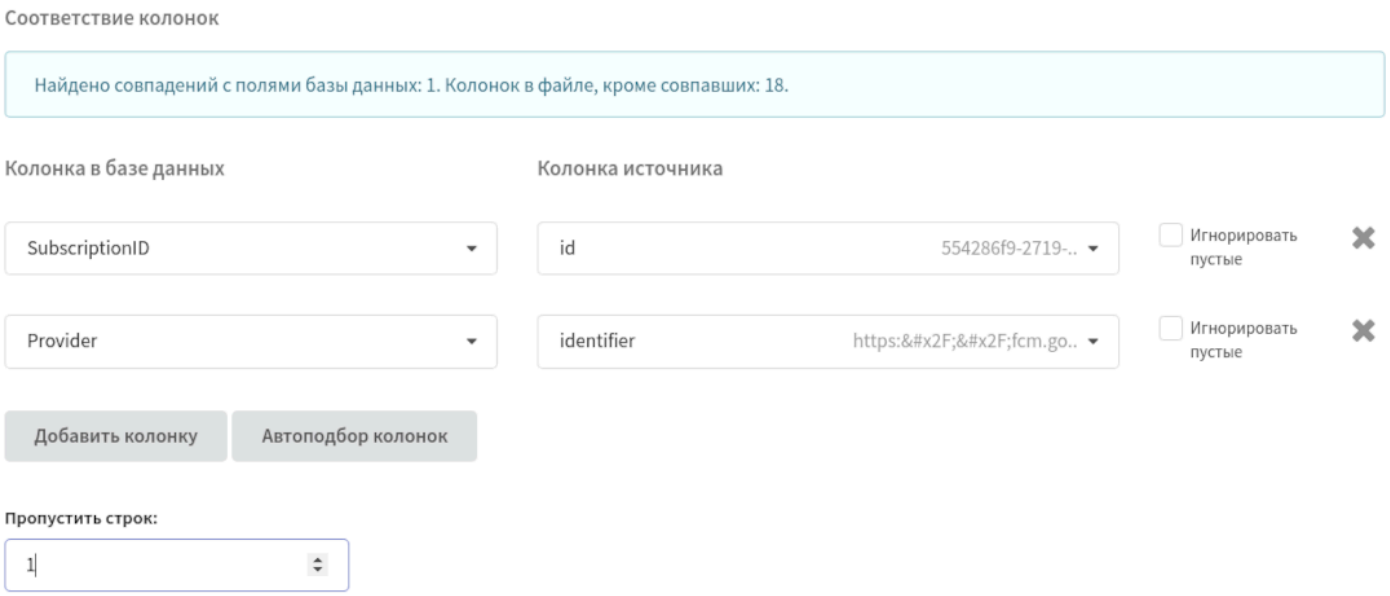
- **SubscriptionID**: поле идентификатора подписки из файла;
- **Provider**: поле идентификатора провайдера Altcraft;

TIP

Если в csv файле отсутствует поле идентификатора провайдера Altcraft, то укажите то же поле, что и в SubscriptionID. Таким образом для каждой подписки будет создан новый подписчик.

CAUTION

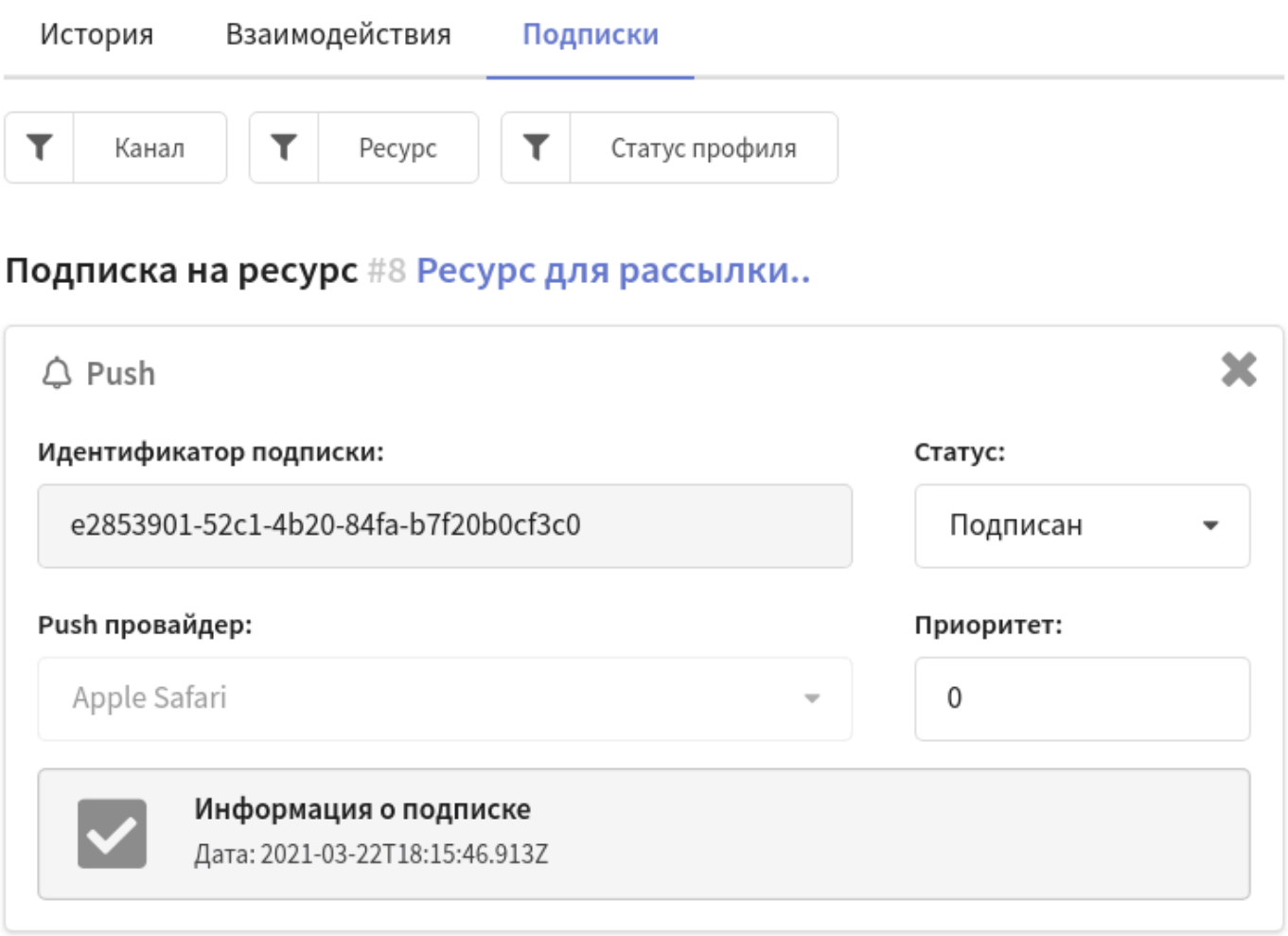
В поле "**Пропустить строк"** укажите значение “**1**” если в файле присутсвуют заголовки.



После того, как вы нажмете кнопку **Импортировать** в правом верхнем углу экрана, начнется процесс импорта новых профилей в Базу данных.

Как только профили будут успешно импортированы, вы можете посмотреть их в выбранной вами Базе профилей. Открыв вкладку **Подписки**, вы увидите новую подписку профиля на push уведомления и

информацию о ней.



Миграция с OneSignal

Шаг 1. Получение VAPID ключей с сервиса OneSignal

> **CAUTION**
>
> Пропустите этот шаг, если вы переносите подписки, настроенные для браузера Safari.

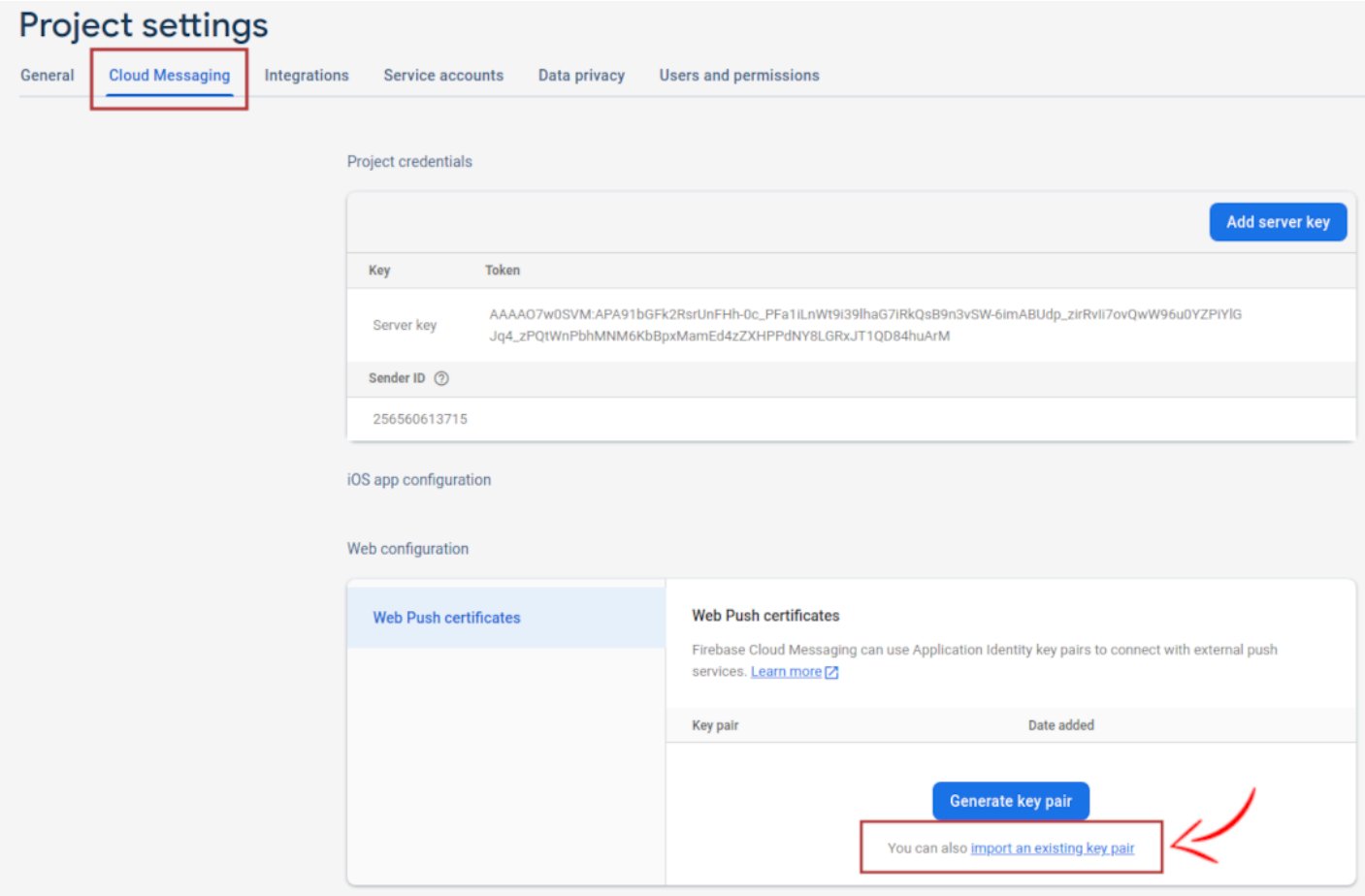
Запросите ключи у технической поддержки сервиса OneSignal. Необхо димы как приватный, так и публичный ключи: **private key**, **public key**.

> **TIP**
>
> Обратите внимание, что **VAPID keys** и **Application Server keys**— это одно и то же.

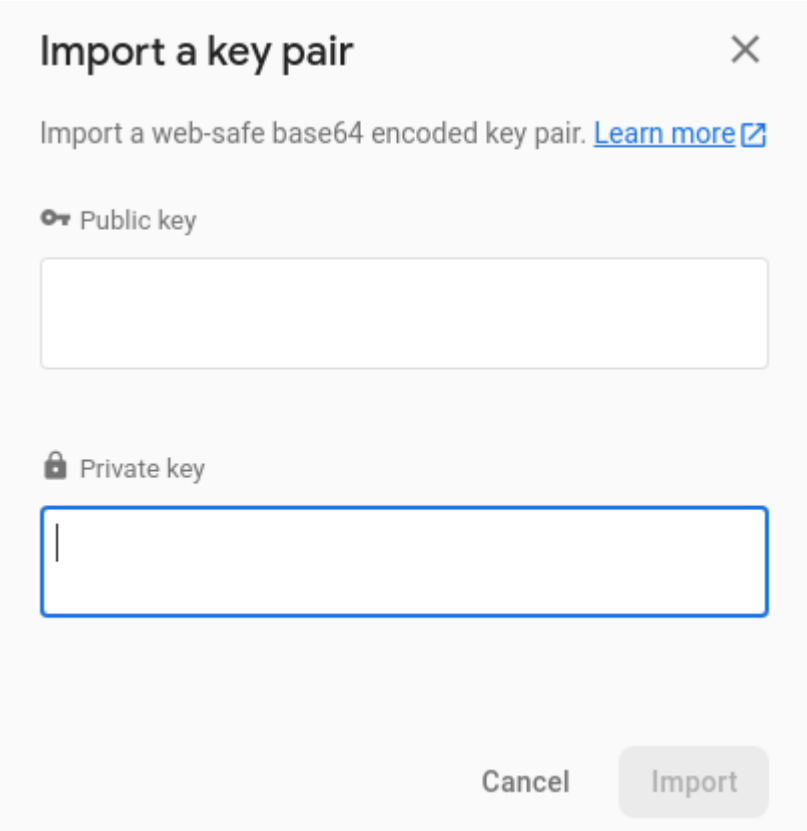
Шаг 2. Импорт в Firebase проект VAPID ключей

> **CAUTION**
>
> Пропустите этот шаг, если вы переносите подписки, настроенные для браузера Safari.

Зайдите в настройки проекта в Firebase и перейдите на вкладку **Cloud Messaging**. Внизу страницы вы увидите раздел **Web Configuration**. Здесь вы можете сгенерировать ключи самостоятельно, но нам это не нужно, поскольку вы уже получили VAPID ключи у своего сервиса. Кликните “**import an existing key pair**” в самом низу раздела.



В появившемся окне введите **Public key** и **Private key**, которые ранее предоставил вам OneSignal.



Шаг 3. Настройка ресурса

О настройке ресурса для Web push-уведомлений читайте в разделе Web push: настройка и подключение.

TIP

Обратите внимание, что настройка ресурса отличаются у разных браузеров.

Шаг 4. Экспорт подписок

Экспорт подписок предполагает создание CSV файла, который будет содержать все ваши текущие данные о пользователях.

Используйте утилиту **Curl**, чтобы сделать **POST запрос**. Запрос сгенерирует **URL**, по которому вы сможете **скачать** сжатый **CSV файл** (**.csv .gz**).

TIP

Создание файла может занять несколько минут в зависимости от количества ваших пользователей.

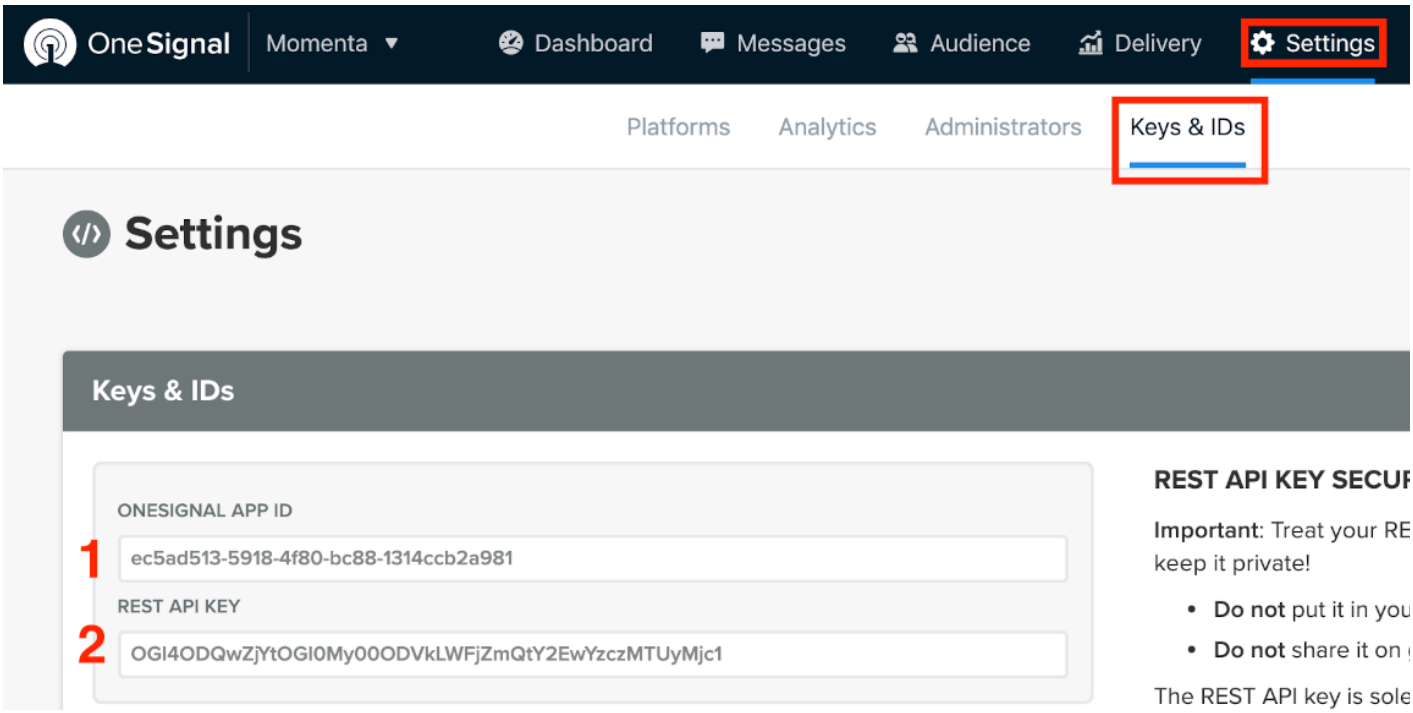
Вы можете проверить, завершен ли процесс создания файла, отправив GET-запрос, в котором в качестве параметра будет передано значение `csv_file_url`.

Если файл еще не готов, будет возвращена ошибка 403. В противном случае будет возвращен сам файл.

Ниже приведен пример запроса, который можно запустить в терминале.

```
curl -X POST -H "Authorization: Basic YOUR_REST_API_KEY" -H "Content-Type:
application/json" -d '{
    "extra_fields": ["web_auth", "web_p256"],
    "last_active_since": "1469392779", "segment_name": "Active Users"
}' "https://onesignal.com/api/v1/players/csv_export?app_id=YOUR_APP_ID"
```

Здесь необходимо указать ваши **REST_API_Key** и **APP_ID**. Эти данные вы можете найти на сервисе OneSignal. Зайдите в **Settings** и перейдите на вкладку **Keys & IDs**.



В теле запроса необходимо передать следующие параметры:

Поле	Значение
“extra_fields”	Дополнительные поля, которые вам необходимы. На данный момент доступны: *location, country, rooted, notification_types, ip, external_user_id, web_auth, web_p256.* В нашем случае обязательно надо указать **web_auth** и **web_p256** .
“last_active_since”	Экспортировать все устройства, у которых значение поля *last_active* больше указанной временной отметки timestamp.
“segment_name”	Экспортировать все устройства, которые относят к указанному сегменту. Вы можете использовать это поле, чтобы экспортировать только интересующие вас сегменты. Подробнее о сегментах в OneSignal: https://documentation.onesignal.com/docs/segmentation.

TIP

Более подробную информацию об экспорте подписок с сервиса OneSignal вы можете получить здесь: https://documentation.onesignal.com/reference/csv-export.

Шаг 5. Передача csv файла с полями администратору или службе поддержки Altcraft

CAUTION

Пропустите этот шаг, если вы переносите подписки, настроенные для браузера Safari.

Если плафторма Altcraft установлена в контур вашей компании, то передайте созданный csv файл с подписками администратору платформы.

Если платформа установлена на облаке, то передайте csv файл в службу поддержки Altcraft.

Шаг 6. Получение обработанного csv файла у администратора

CAUTION

Пропустите этот шаг, если вы переносите подписки, настроенные для браузера Safari.

Шаг 7. Импорт профилей

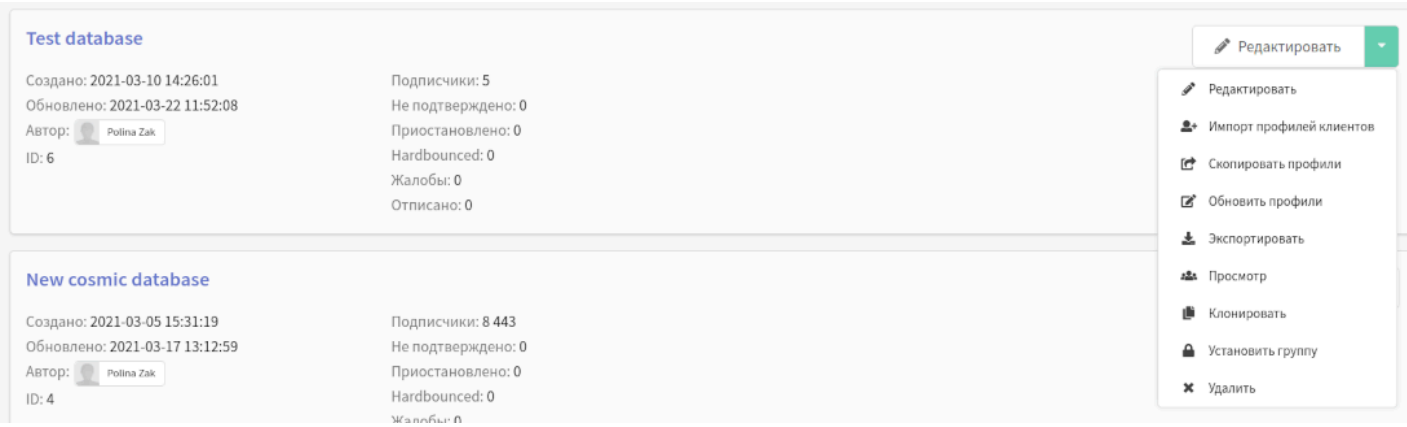
Теперь вы можете импортировать профили в **Базы профилей** из csv файла, при этом **создавая подписку** на нужный провайдер.

Кроме того, у вас есть возможность выполнить **обновление базы** с созданием подписки на этот же провайдер. Обратите внимание, в данном случае это обязательно должен быть один из Firebase провайдеров.

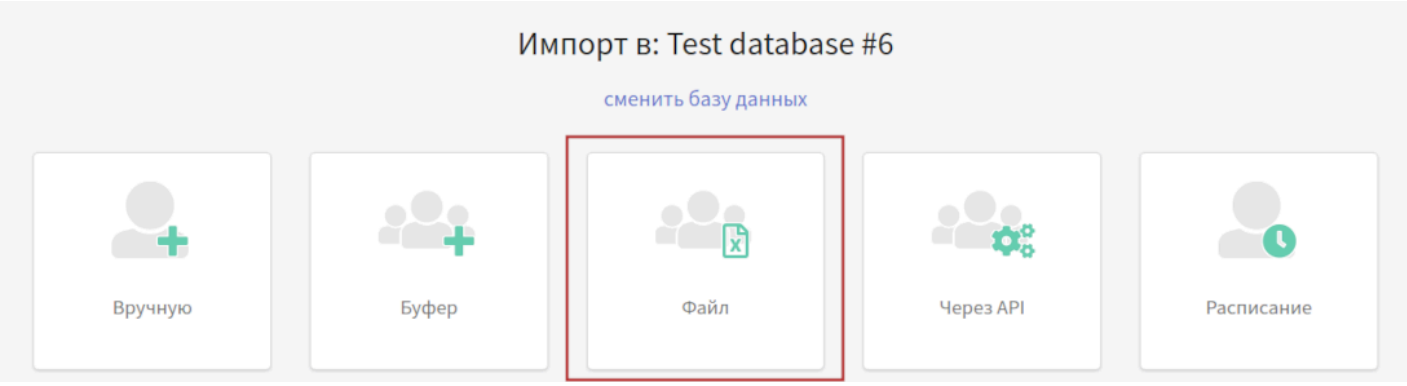
TIP

Идентификатор провайдера позволяет узнать, к какому именно браузеру относится подписка и через какой сервис она создавалась. Примеры провайдеров: Chrome Firebase, Apple Safari, Mozilla Firefox.

Итак, создайте или выберите подходящую базу и в выпадающем списке кликните **Импорт профилей клиентов**.



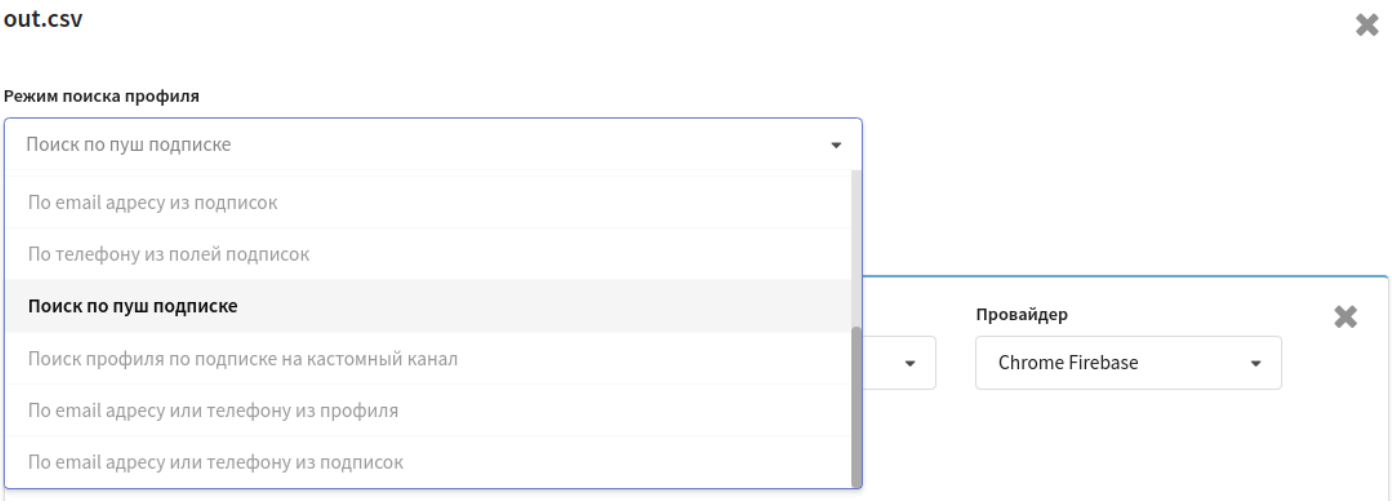
Выберите способ импорта — **Файл**. Загрузите необходимый csv файл с подписками.



Кликните **Режим поиска подписки** и в выпадающем списке выберите **Поиск по пуш подписке**.

CAUTION

Если в вашем csv файле есть какие-то данные, которые можно использовать, чтобы настроить соответствия между профилями (“сматчить пользователей”), то при импорте выберите соответствующий режим поиска профиля.

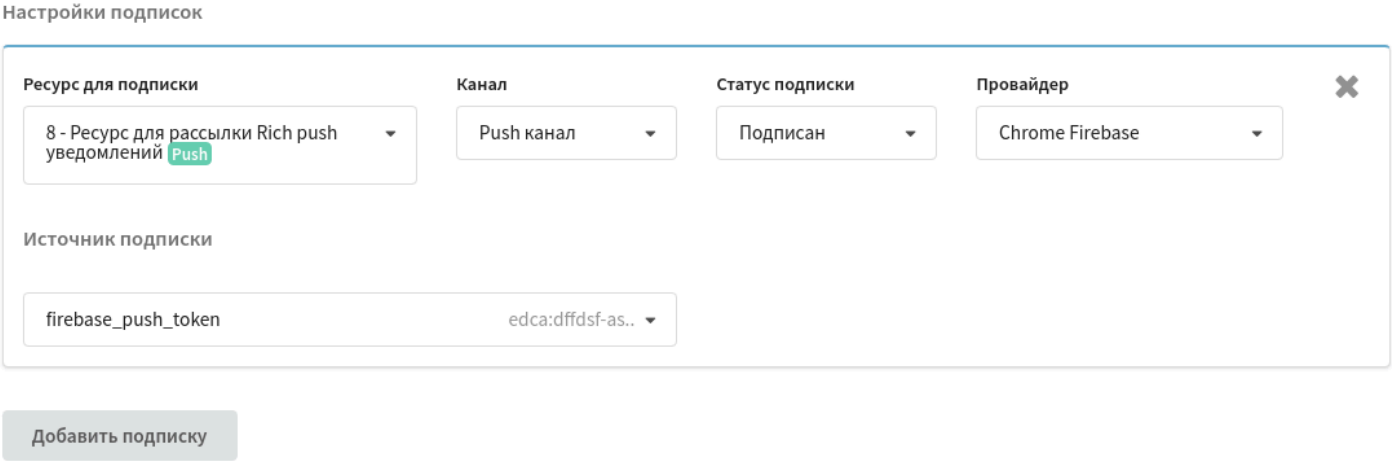


Нажмите на кнопку **Добавить подписку**. Затем настройте подписку:

- **Канал** — **Push канал**;
- **Провайдер** — браузер, через который осуществлена подписка. Если пользователь подписался через браузер **Safari**, то укажите **Apple Safari**.Если браузер подписки неизвестен, то можно импортировать подписки с провайдером **Chrome Firebase**.
- **Источник подписки** — поле идентификатора подписки из csv файла. Если пользователь подписался через браузер **Safari**, то вам необходимо определить данное поле самостоятельно. Для других бразуеров здесь необходимо указать **firebase_push_token**.

CAUTION

Для создания подписки необходимо название поля, где хранится id той подписки, которую вы импортируете. В нашем csv файле такое поле будет называться `firebase_push_token`.

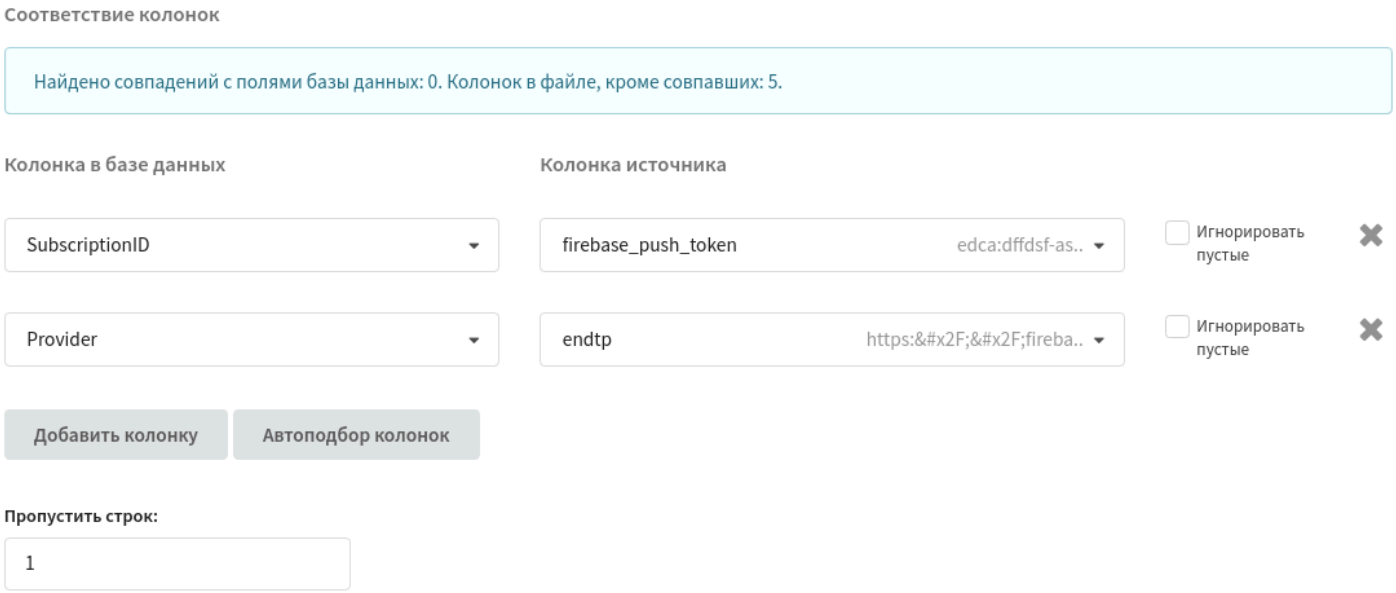


Теперь необходимо настроить соответствие колонок:

Поле	Для браузеров Safari	Для остальных браузеров
SubscriptionID	Поле идентификатора подписки из csv файла	firebase_push_token
Provider	Если в csv файле отсутствует поле идентификатора провайдера Altcraft, то укажите то же поле, что и в SubscriptionID. Таким образом для каждой подписки будет создан новый подписчик.	

CAUTION

В поле **Пропустить строк** укажите значение “**1**”. В противном случае, при импорте будут захватываться заголовки.



После того, как вы нажмете кнопку **Импортировать** в правом верхнем углу экрана, начнется процесс импорта новых профилей в Базу данных.

Как только профили будут успешно импортированы, вы можете посмотреть их в выбранной вами Базе профилей. Открыв вкладку **Подписки**, вы увидите новую подписку профиля на push уведомления и информацию о ней.

История Взаимодействия Подписки

Канал Ресурс Статус профиля

Подписка на ресурс #8 Ресурс для рассылки..

Push

Идентификатор подписки:

edca:dffdsf-asdsds-fdgfgfgf-jhjhjhj

Статус:

Подписан

Push провайдер:

Chrome Firebase

Приоритет:

0

Информация о подписке

Дата: 2021-03-22T15:53:15.166Z

Добавить подписку

Email

Первая Email-рассылка

Электронная почта остаётся одним из наиболее эффективным способом маркетинговой коммуникации с клиентами. Вы можете отправлять подписчикам больше контента, чем по другим каналам: изображения, ссылки на различные товары и услуги, интерактивные элементы, привлекающие внимание и повышающие вовлечённость клиентов.

Это руководство поможет вам организовать Email-рассылку с нуля: от настройки домена отправки до просмотра статистики рассылки.

INFO

Altcraft Platform отслеживает действия подписчиков в Email-сообщении, такие как открытие, прочтение, переход по ссылке.

Шаг 1: Настройте отправщик Email

Платформа Altcraft отправляет электронную почту по протоколу **SMTP**. Для этого нами был разработан собственный транспортный модуль **AKMTA** (Altcraft Mail Transfer Agent), который позволяет оптимизировать доставку писем — даже если аудитория рассылки составляет **десятки миллионов** получателей.

Домены отправки и отслеживания

Для **отправки** email-сообщений вам потребуются **домен отправки**. Домен будет использоваться в адресе отправителя (**from email**), который видят получатели в клиенте почты. С помощью технологий **SPF и DKIM** домен связывается с SMTP отправщиком, что не позволяет подделать ваши рассылки.

Для **отслеживания** действий получателей email вам также потребуется **трекинг-домен**. Это может быть поддомен вашего основного домена или отдельный домен. На трекинг-домен перенаправляются переходы по ссылкам в сообщении — так платформа отслеживает **клики**. Через него же в сообщения загружаются изображения для регистрации **открытий** письма.

Когда вы приобрели домен — на стороне предоставившего его **регистратора доменных имён** необходимо внести записи конфигурации, обеспечивающие эффективную доставку.

TIP

Настройки собственного домена описаны в **отдельной статье**. Если вы пользуетесь **облачным** аккаунтом Altcraft Marketing — передайте информацию службе поддержки на **team@altcraft.com**.

При установке платформы на ваше оборудование свяжитесь с нами и мы поможем настроить инфраструктуру отправки.

Установка транспортного модуля AKMTA

CAUTION

При использовании **облачного** сервиса мы предоставляем собственную инфраструктуру отправки со всеми необходимыми настройками. Единственное, что нужно для начала отправки — ваш настроенный домен (см. предыдущий пункт).

При установке платформы Altcraft Marketing на ваше оборудование (**on-Premise**) вы можете настроить один или несколько серверов отправки с транспортными модулями AKMTA. Для этого в Панели администратора платформы необходимо настроить:

- **Ноды** — сетевые узлы отправки. Модуль использует доступные вам **IP адреса** отправки.
- **Сендеры** — записи конфигурации, объединяющие параметры идентификации отправщика, скорости отправки и настройки поведения сендера при ошибках отправки.
- **Виртуальные сендеры** — правила отправки с нескольких сендеров. Могут быть настроены как в Панели администратора, так и в [Интерфейсе пользователя](/user-guide/settings/9405938).

Шаг 2: Подпишите клиентов на Email-канал

Базы профилей

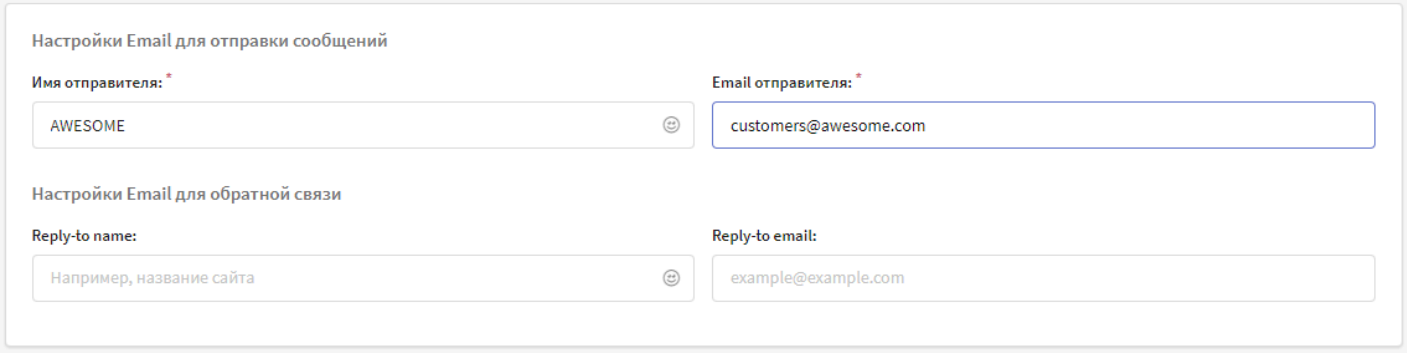
Рассылки Altcraft Marketing получают ваши подписчики. **Профили** подписчиков хранятся в **базах данных профилей**.

У каждого профиля клиента есть системное поле **Email**: оно используется для идентификации профилей и в тех случаях, когда у ваших клиентов только один адрес электронной почты.

TIP

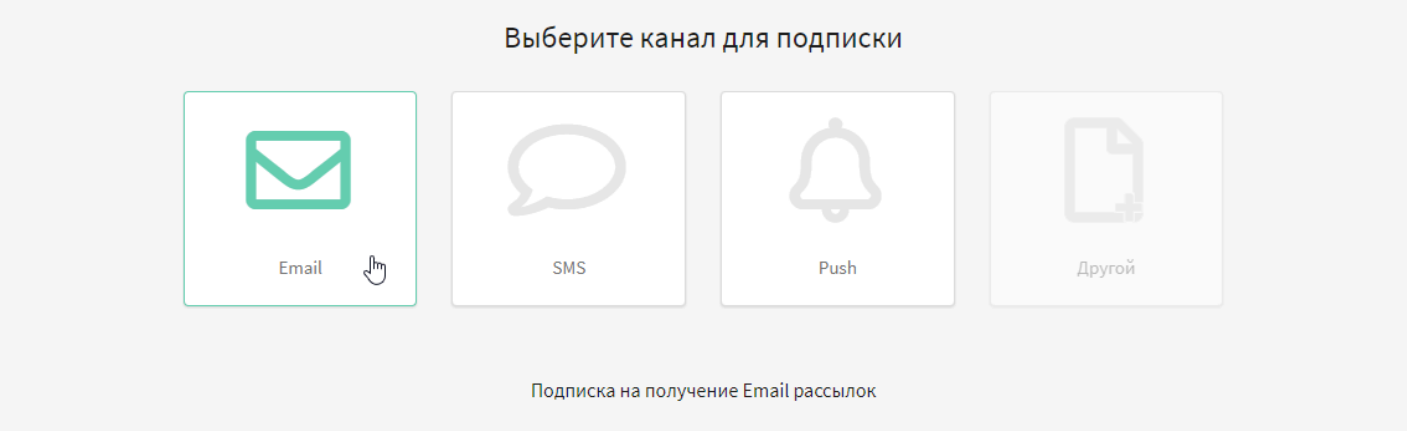
Если вы используете разные контактные данные одного и того же клиента — или отправляете контент разного содержания — настройте несколько **ресурсов подписок**.

При создании базы данных **Email отправителя** является обязательным параметром. От его имени будет производиться отправка без использования ресурсов. Укажите адрес отправки в соответствующем поле. Также укажите Имя отправителя, и при необходимости — данные для ответа на письмо (**Обратный получатель** и **Обратный email**):

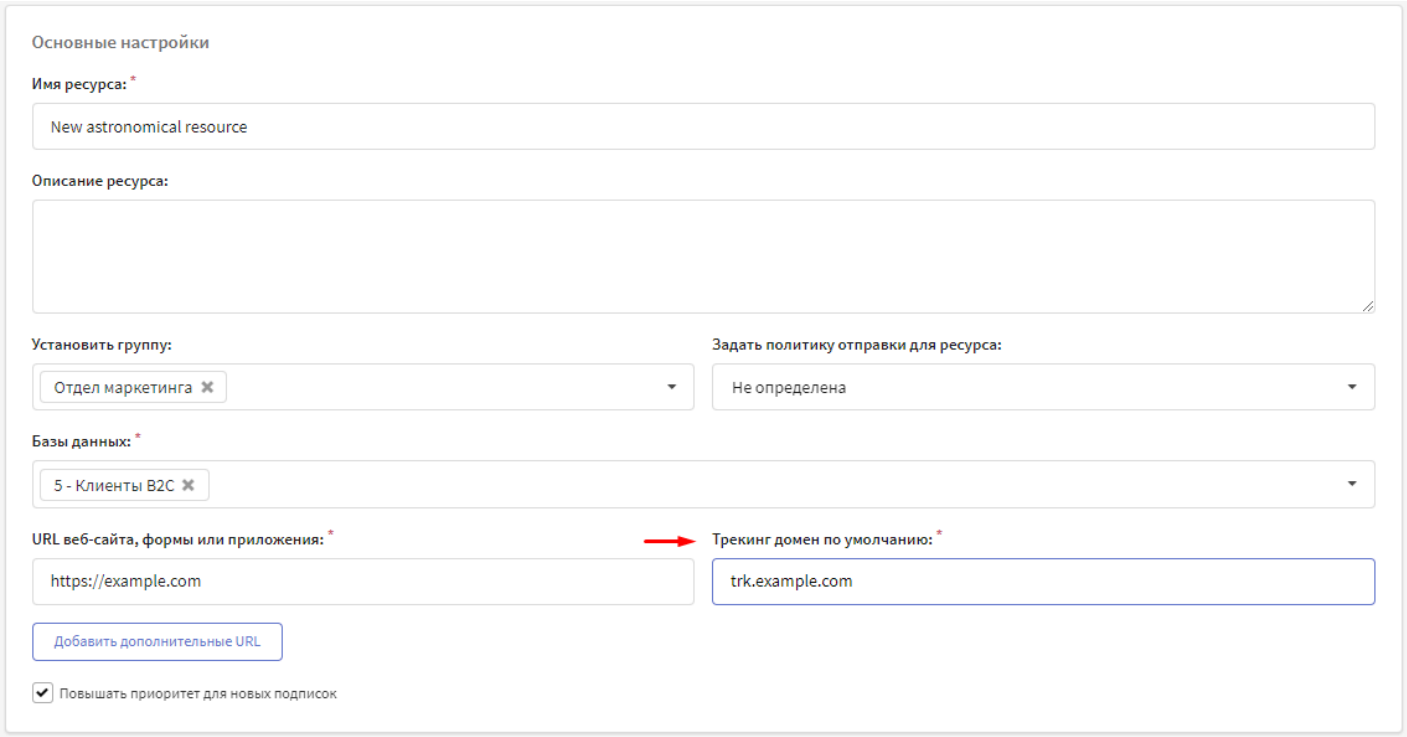


Ресурсы подписок

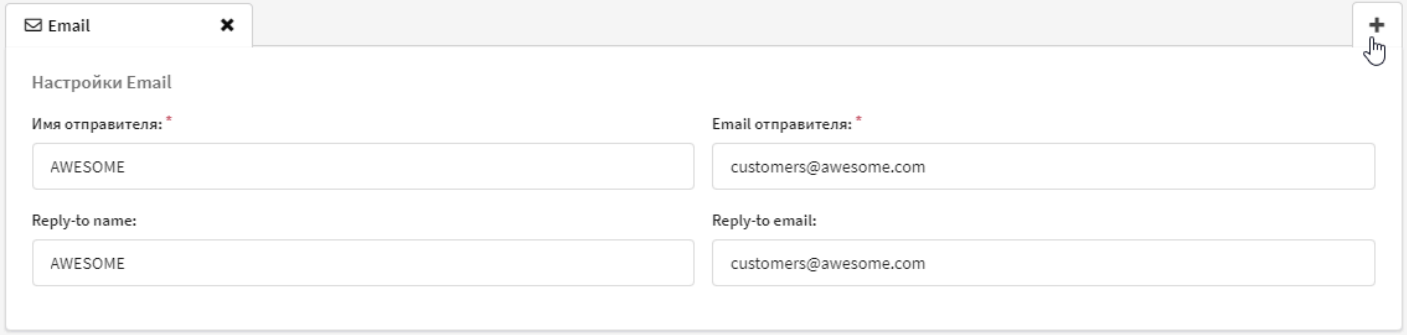
Каждый профиль клиента может быть подписан на один или несколько **ресурсов**. Ресурсом может быть отдельное направление маркетинга, сайт, приложение или тип контента. Выберите соответствующий пункт главного меню, создайте новый или выберите существующий ресурс. Добавьте Email-канал в существующий ресурс или выберите его в качестве основного канала для нового:



В основных настройках добавьте необходимые базы данных, при необходимости назначьте **политику отправки**. Укажите настроенный ранее **Трекинг домен по умолчанию**:



В настройках Email-канала ресурса укажите **Имя** и **Email отправителя**, а также — при необходимости — имя и адрес для получения ответов. Эти настройки **переопределят** аналогичные параметры **базы данных** при отправке.



Профили клиентов и подписка на ресурс

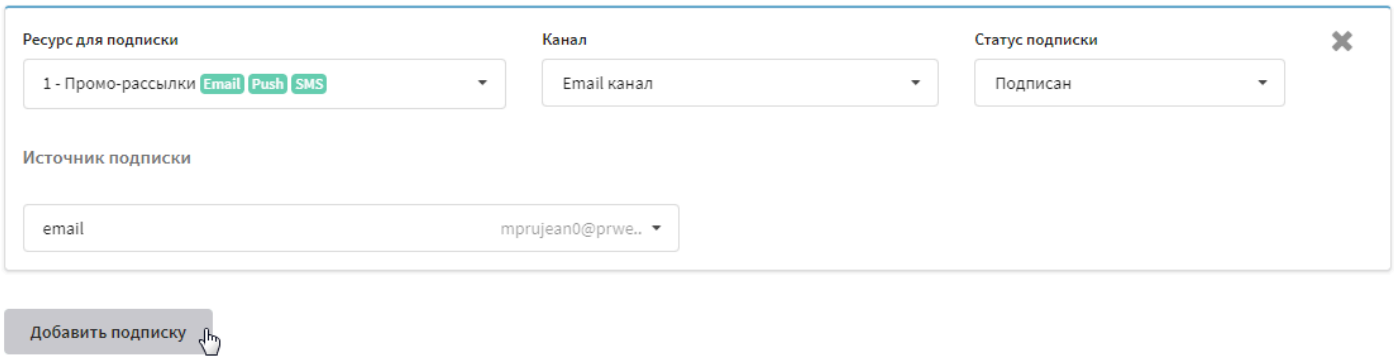
После настройки базы и ресурса подписок можно импортировать в платформу данные клиентов. Для каждого клиента в Altcraft Marketing создаётся единый цифровой профиль, объединяющий личные данные, историю действий и подписки на ваши ресурсы.

Вы можете загрузить данные клиентов вручную: по одному, списком, или из табличного файла. Различные способы импорта описаны в этой статье.

Также вы можете настроить автоматический импорт профилей клиентов:

- по расписанию из вашей SQL базы данных
- посредством интеграции с API платформы

При импорте данных подпишите клиентов на подготовленный ранее ресурс. Для этого выберите Email в качестве канала связи и укажите колонку с адресами подписчиков:



TIP

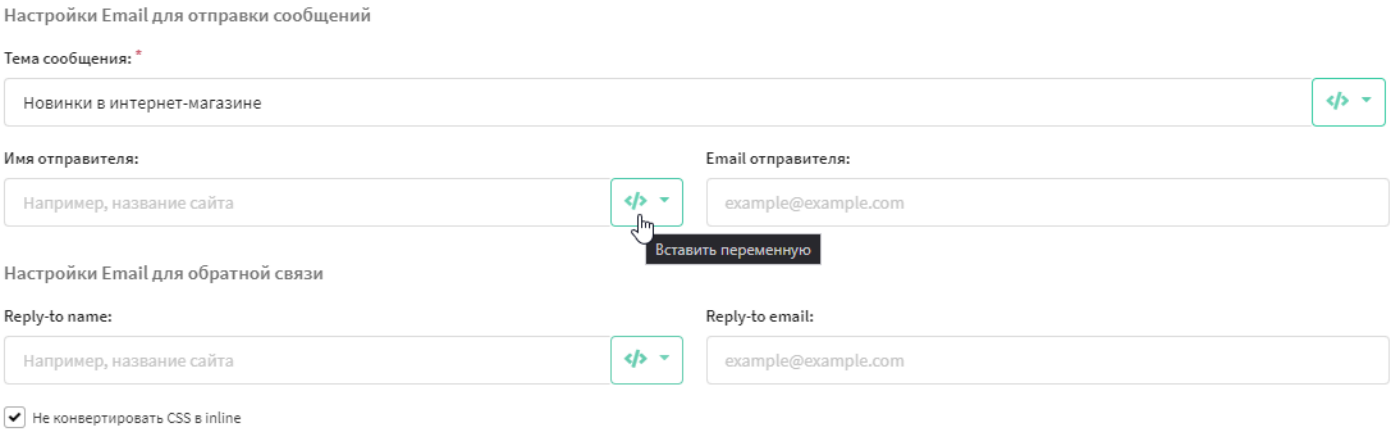
Если у вас уже есть база профилей, добавьте им подписку на новый ресурс. Есть два варианта:

- если у клиентов есть подписки на другой ресурс с актуальными email адресами — используйте массовое обновление профилей.
- если для подписки на ресурс, email-адреса необходимо загрузить в платформу — используйте импорт профилей в режиме “только обновление”.

Шаг 3: Подготовьте контент для отправки

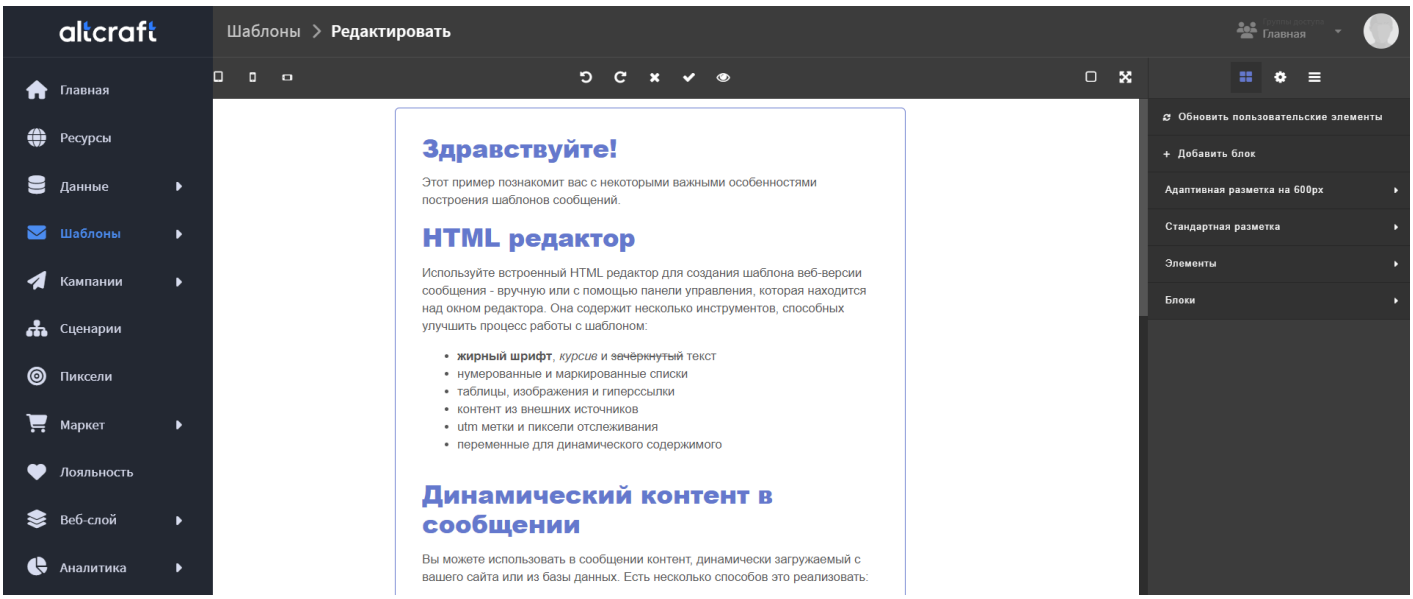
Контент для ваших Email-рассылок создаётся в **Шаблонах сообщений**. Шаблоны содержат контент для всех доступных каналов. При создании нового шаблона выберите Email-канал или добавьте его для существующего.

Для Email-версии шаблона можно задать **Тему** — она будет отображаться в переписке, а также задать **имя и адрес отправителя** — настройки в шаблоне **переопределят** настройки **как ресурса, так и базы данных**.



Шаблоны в Altcraft Marketing предлагают множество возможностей **персонализации** контента для конкретного получателя. Для этого используются различные типы переменных, логические выражения, циклы и функции, а также динамический контент из различных внешних источников.

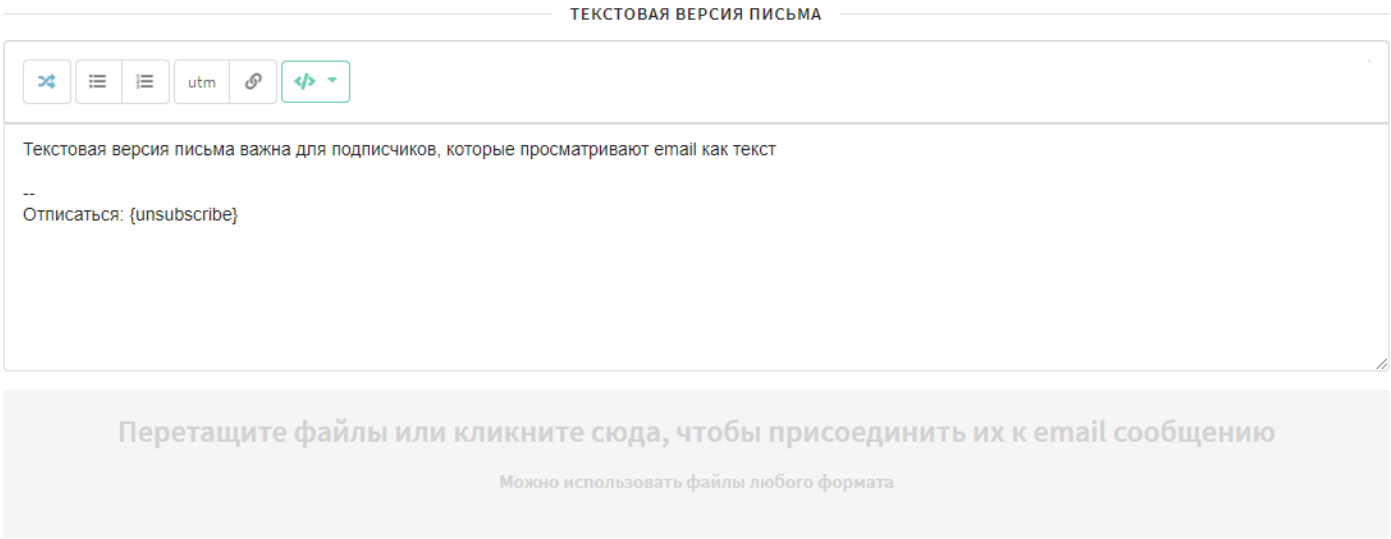
Для создания шаблонов Email-сообщений используйте **Редактор кода** или **Визуальный редактор**:



Часть ваших подписчиков может пользоваться клиентом почты, в котором недоступна загрузка изображений — например, по корпоративной политике. Для таких случаев в шаблоне есть **Текстовая версия**.

CAUTION

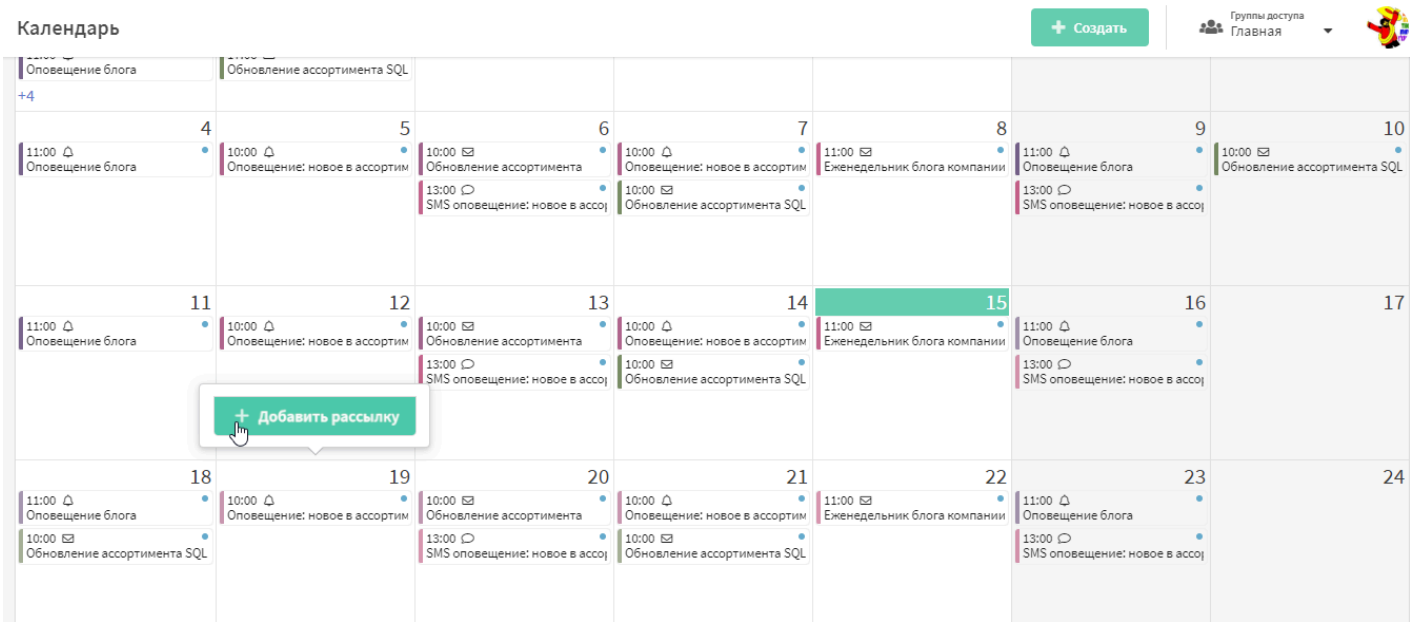
Текстовая версия отправляется вместе с HTML-версией письма. Если она будет не заполнена — или не будет соответствовать HTML — это может несколько негативно сказаться на репутации отправителя.
В случае же её наличия показатели доверия к рассылке серьёзно возрастают.



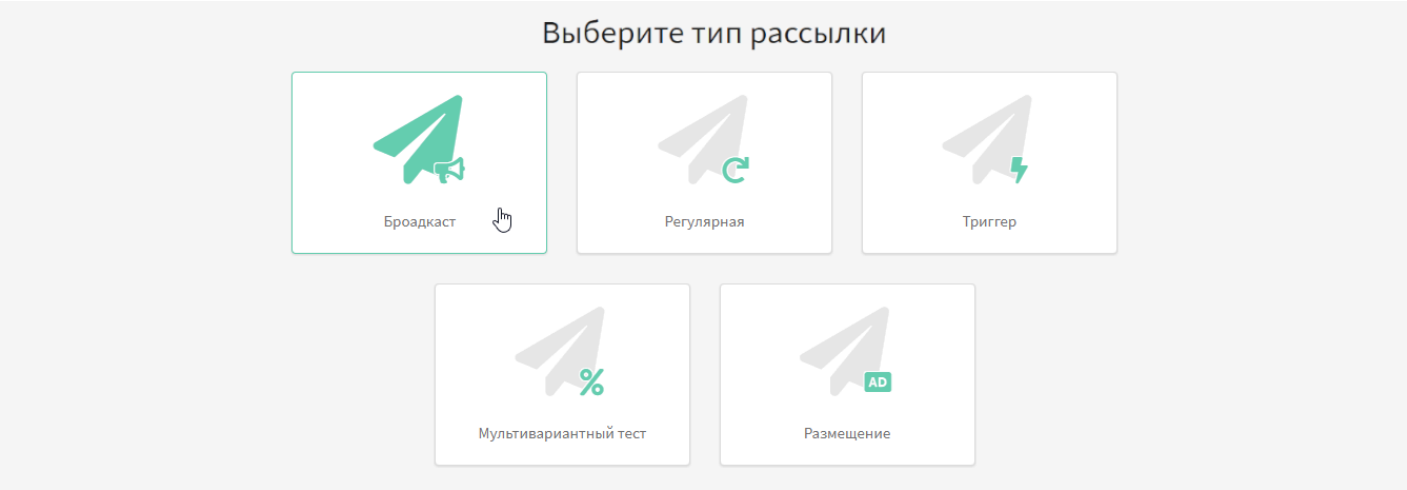
Ниже редактора текстовой версии находится форма, с помощью которой вы можете **прикрепить** к сообщению **файлы в любом формате**.

Шаг 4: Выберите тип рассылки

	Для отправки Email сообщений используется инструмент **Рассылка**. Если вам нужно произвести **разовую** рассылку по целой базе профилей или определённому сегменту аудитории — подойдёт броадкаст. Если же рассылка будет **периодической** — ваш выбор это регулярная рассылка. В главном меню выберите раздел **Рассылки** и откройте **Календарь**. В нём отображаются все рассылки, отправляемые по расписанию.



Определитесь с датой и создайте рассылку нужного типа:



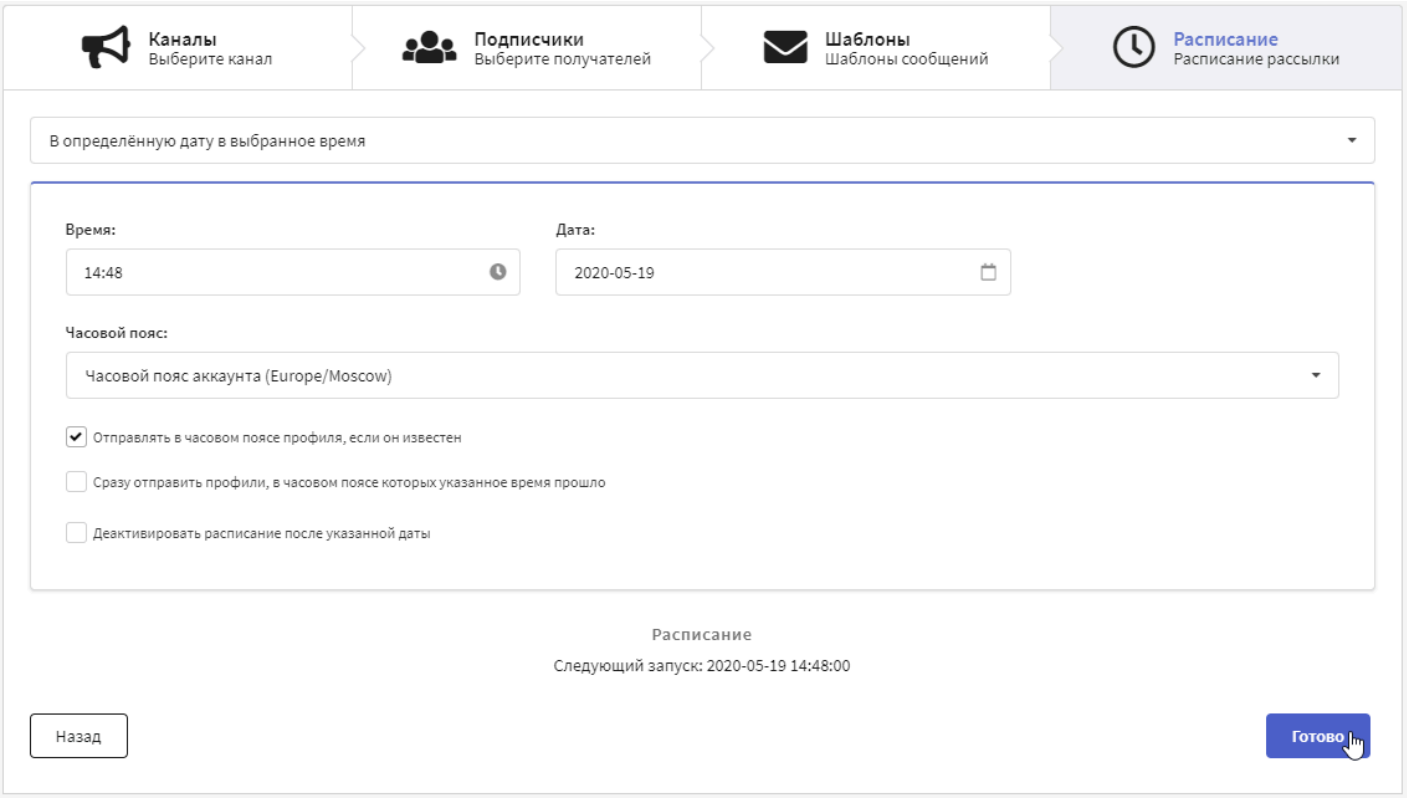
TIP

Подробную инструкцию по настройке **Броадкаста** вы найдёте на этой странице.

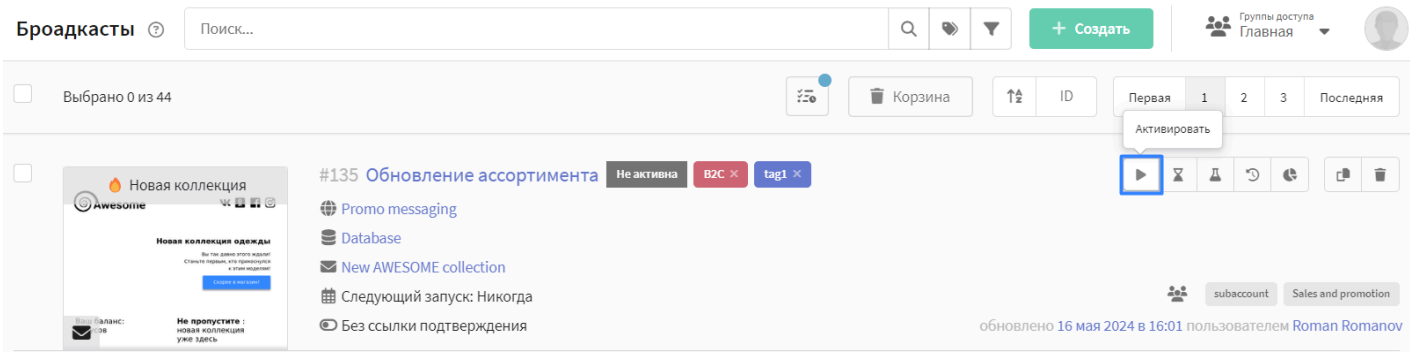
TIP

Подробную инструкцию по настройке **Регулярной рассылки** вы найдёте на этой странице.

Заполнив основные настройки рассылки, выберите **Email** как канал связи, настроенный ранее ресурс подписок и аудиторию получателей. Добавьте в рассылку шаблон сообщения и установите расписание. Например, для броадкаста подойдёт вариант “В определённую дату в выбранное время”:



Когда рассылка настроена, сохраните её. Она будет отображаться в списке рассылок соответствующего типа. В правой части каждой записи есть меню быстрых действий. С его помощью вы можете сделать **тестовую** отправку — и если всё успешно, **активировать** рассылку:



CAUTION

Активная рассылка начнёт отправлять письма, когда наступит время запуска по расписанию. Сообщения передаются на сендер-отправщик. Очередь сендера вы также можете открыть из меню быстрых действий.

Скорость отправки зависит от репутации домена отправителя.

Шаг 5: Оцените эффективность рассылки

Журнал рассылок

Журнал рассылок — это инструмент, предоставляющий краткую сводку о ходе выполнения рассылки. В нем фиксируются ключевые события, параметры рассылок и статистика, что позволяет выявить возможные проблемы.

Чтобы открыть журнал рассылки, нажмите на кнопку **“Журнал”** в меню быстрых действий:

Журнал рассылки содержит следующую информацию:

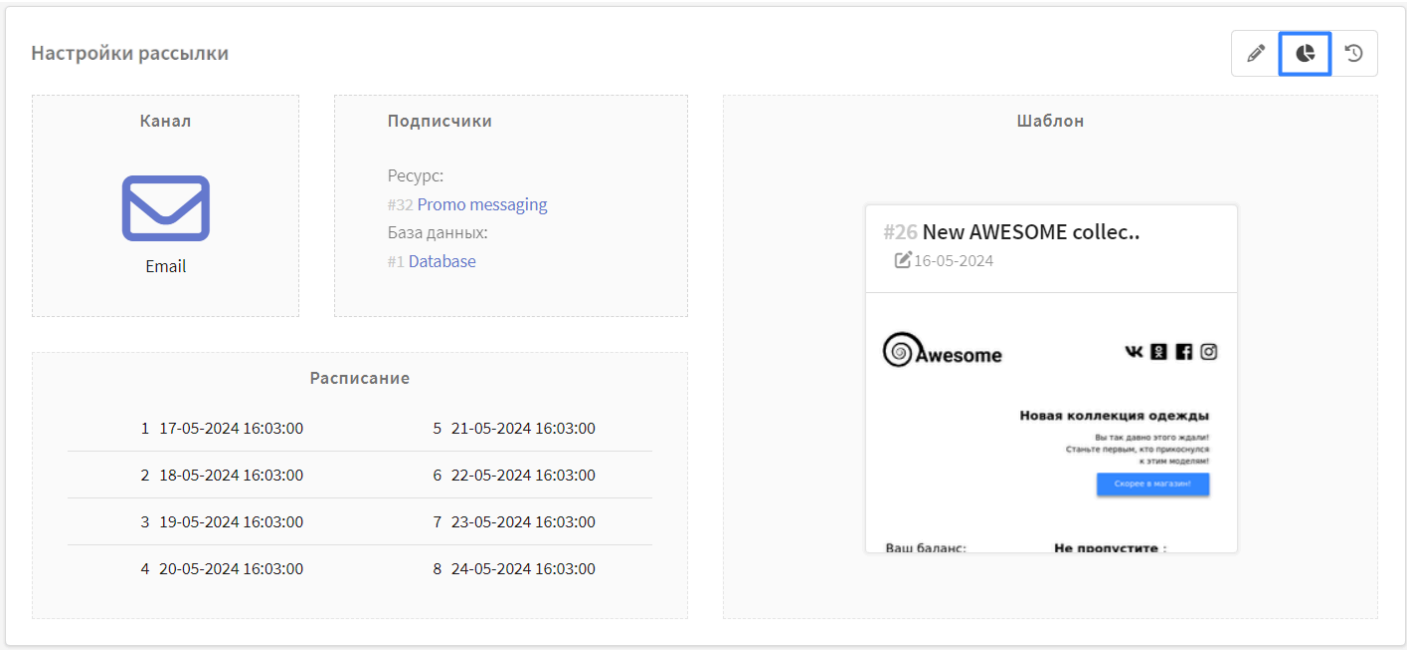
Тип	Описание
Запуск рассылки	Время начала выполнения email-рассылки
Лимит	Показывает установлен ли лимит на отправку
Источник подписчиков	Источник подписчиков: **List** — База данных **Segment** — Сегмент
ID сегмента / ID базы данных	Идентификатор сегмента / Идентификатор базы данных
Время получения сегмента	Время, затраченное на загрузку данных сегмента из базы (в мсек)
Количество профилей полученных из источника	Общее число профилей, найденных в указанном сегменте/базе данных
Количество найденных подписчиков	Число профилей в указанном сегменте/базе данных с подпиской на выбранный ресурс
Количество hard bounces	Письма, которые не могут быть доставлены (несуществующие адреса и т.д.)

Тип	Описание
Количество подписчиков, кому был отправлен email	Фактическое число успешных отправок
Рассылка завершена	Время окончания рассылки
Всего времени затрачено	Общее время выполнения всей рассылки от запуска до завершения

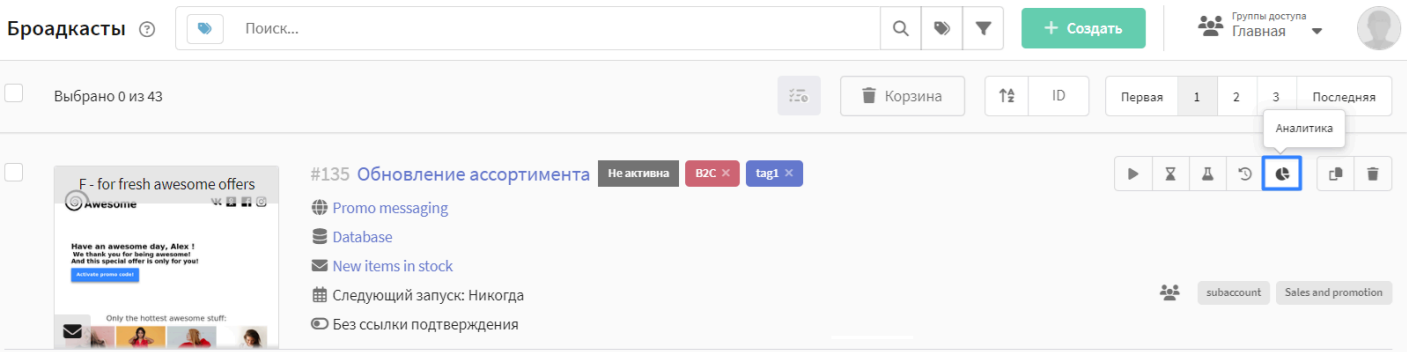
Отчеты и Аналитика

Для оценки результатов текущей рассылки вы можете просмотреть отчет по каналу. **Отчёт по каналам** предоставляет аналитику по одной или нескольким рассылкам, по выбранному каналу связи.

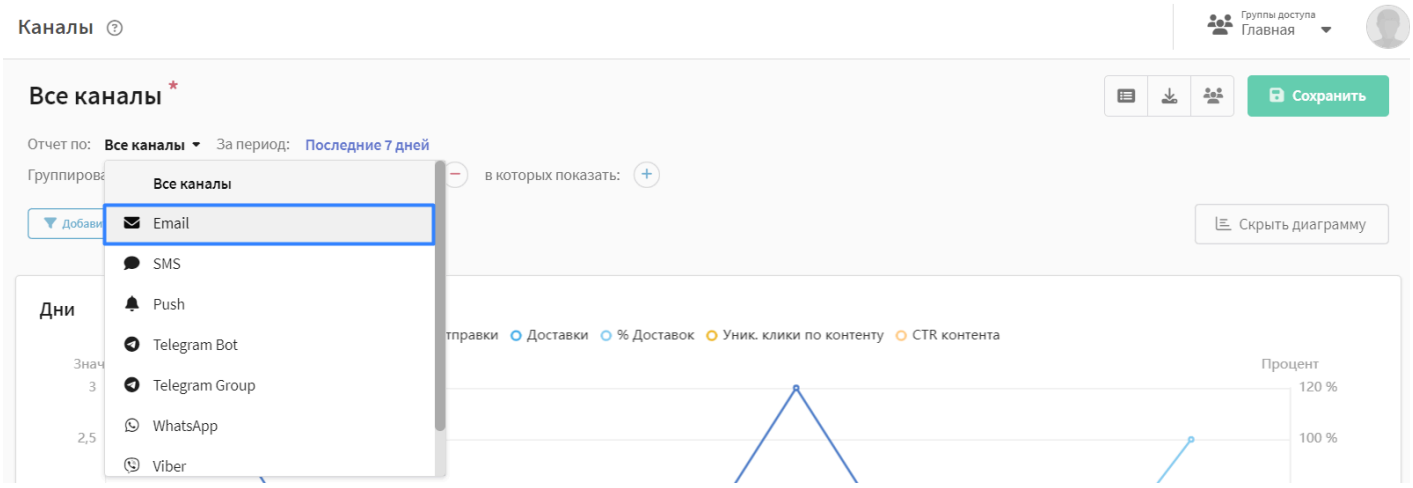
К отчёту по конкретной рассылке можно перейти по **ссылке в редакторе рассылок**:



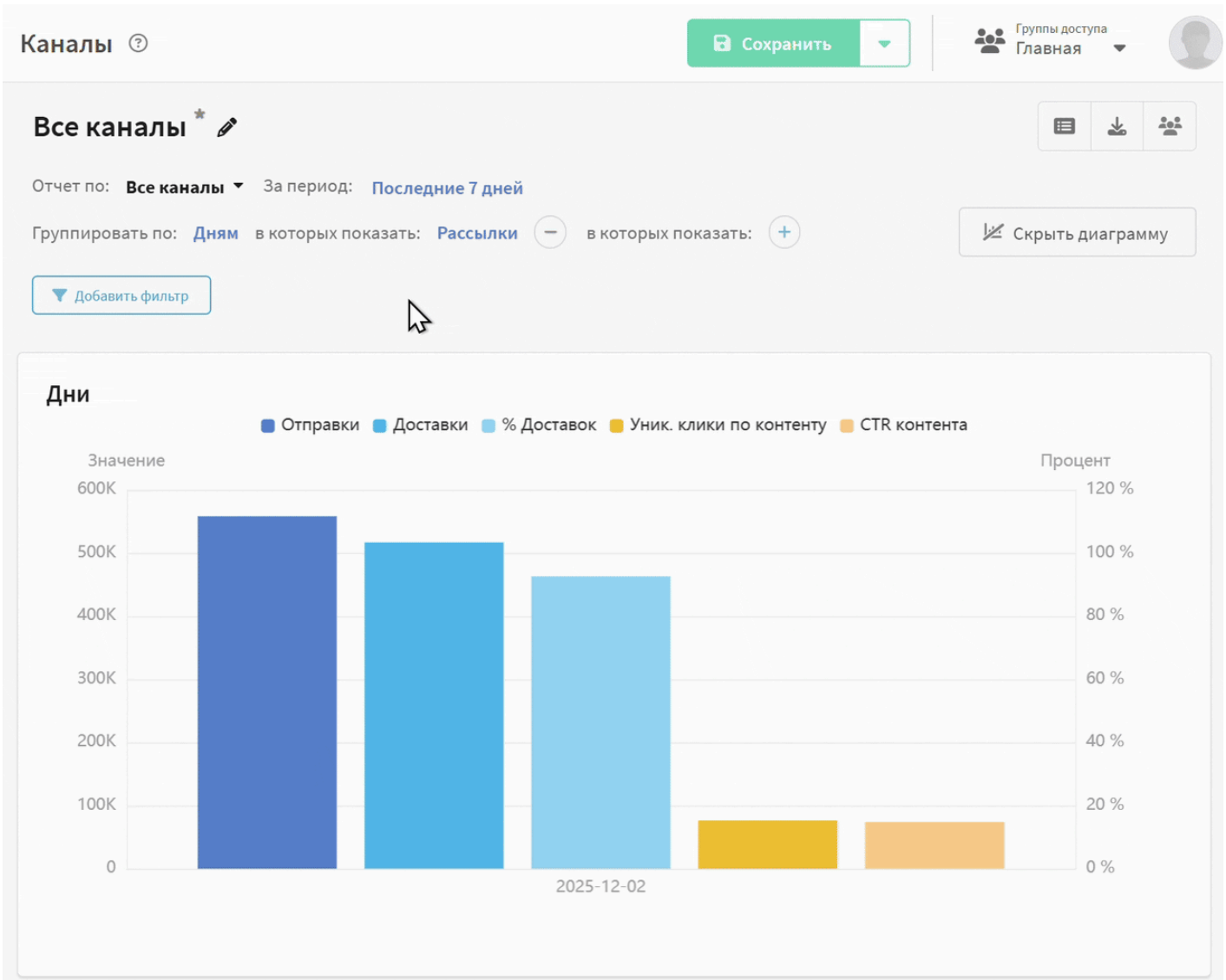
Или выберите кнопку **Аналитика** из панели быстрых действий:



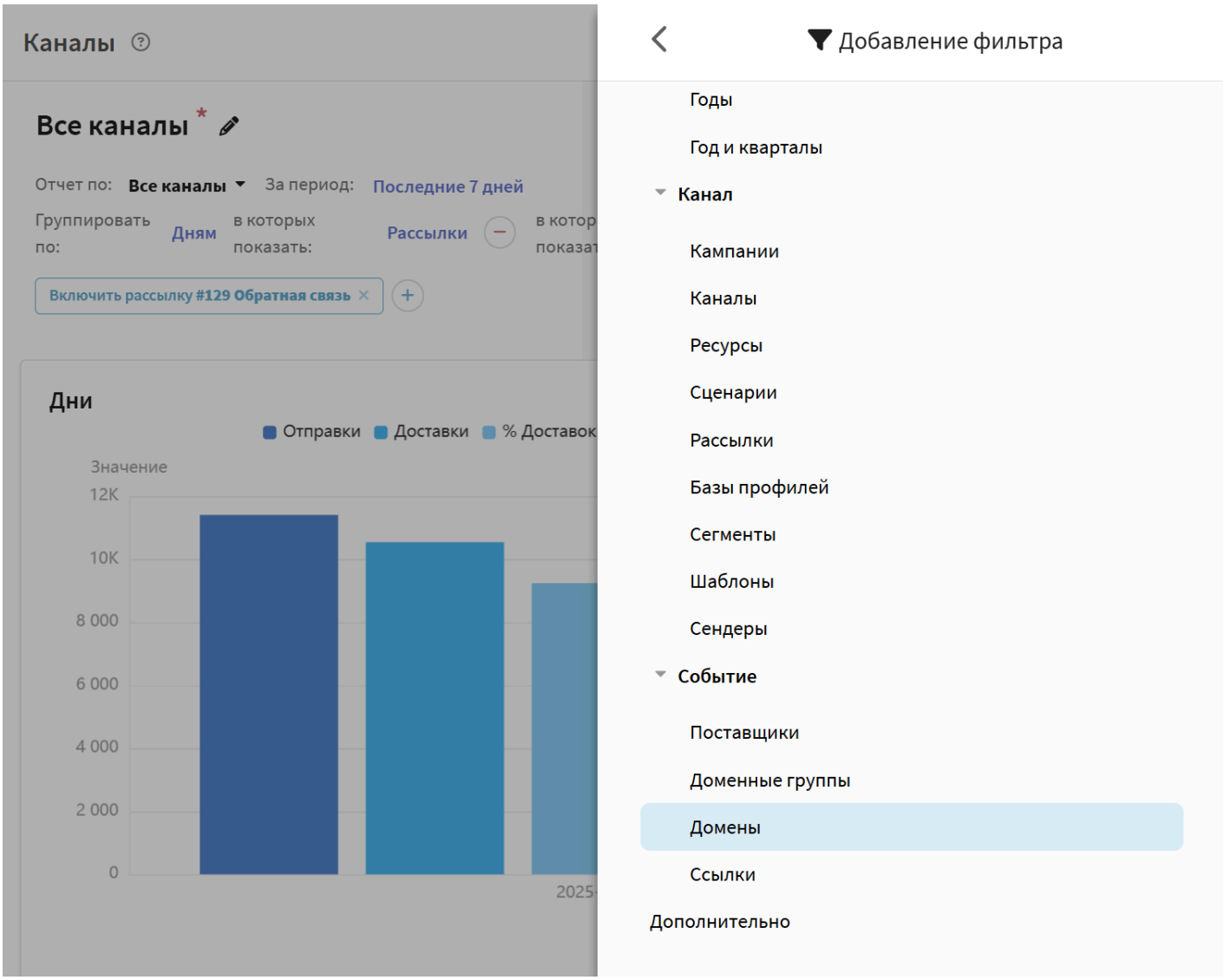
Есть еще третий способ получения статистики по рассылке. Для этого выберите раздел **“Аналитика”** по **“Каналам”** в главном меню платформы. По умолчанию отобразится отчет по **“Всем каналам”** за последние 7 дней. Для группировки по **Email** каналу выберите **“Отчет по”:**.



Следующим шагом **“Добавьте фильтр”** по **“Рассылкам”**. В отобразившемся списке выберите нужную рассылку.



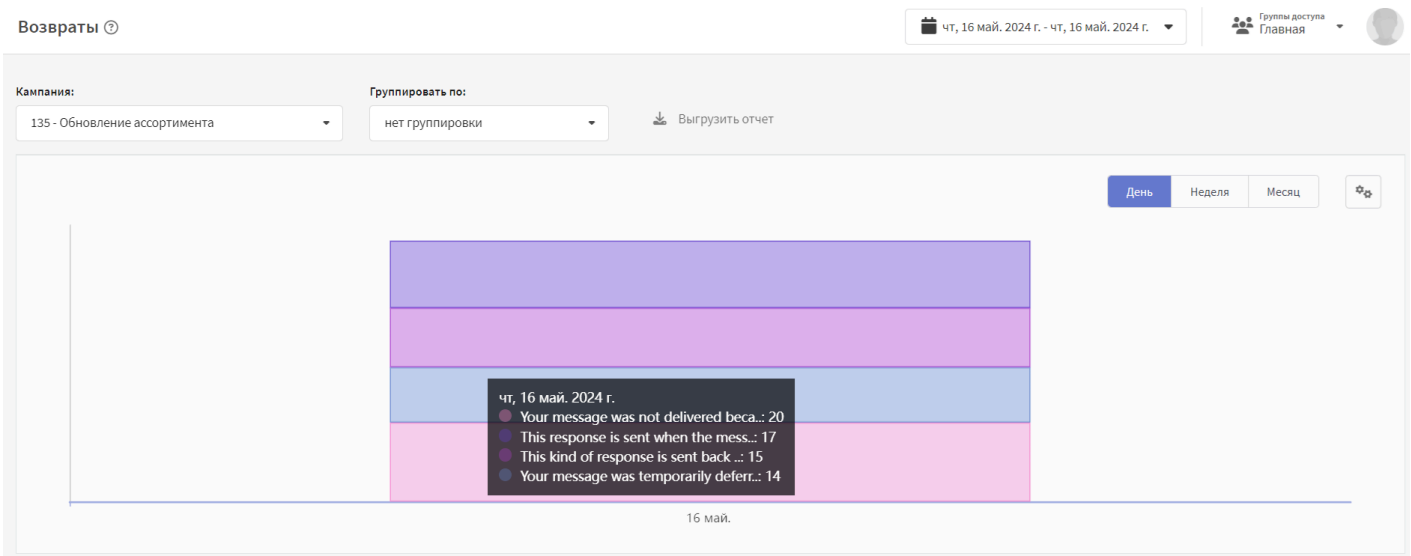
Чтобы проанализировать доставляемость сообщений на электронные почтовые ящики разных провайдеров, вы можете сгруппировать отчёт по **доменной группе** получателей:



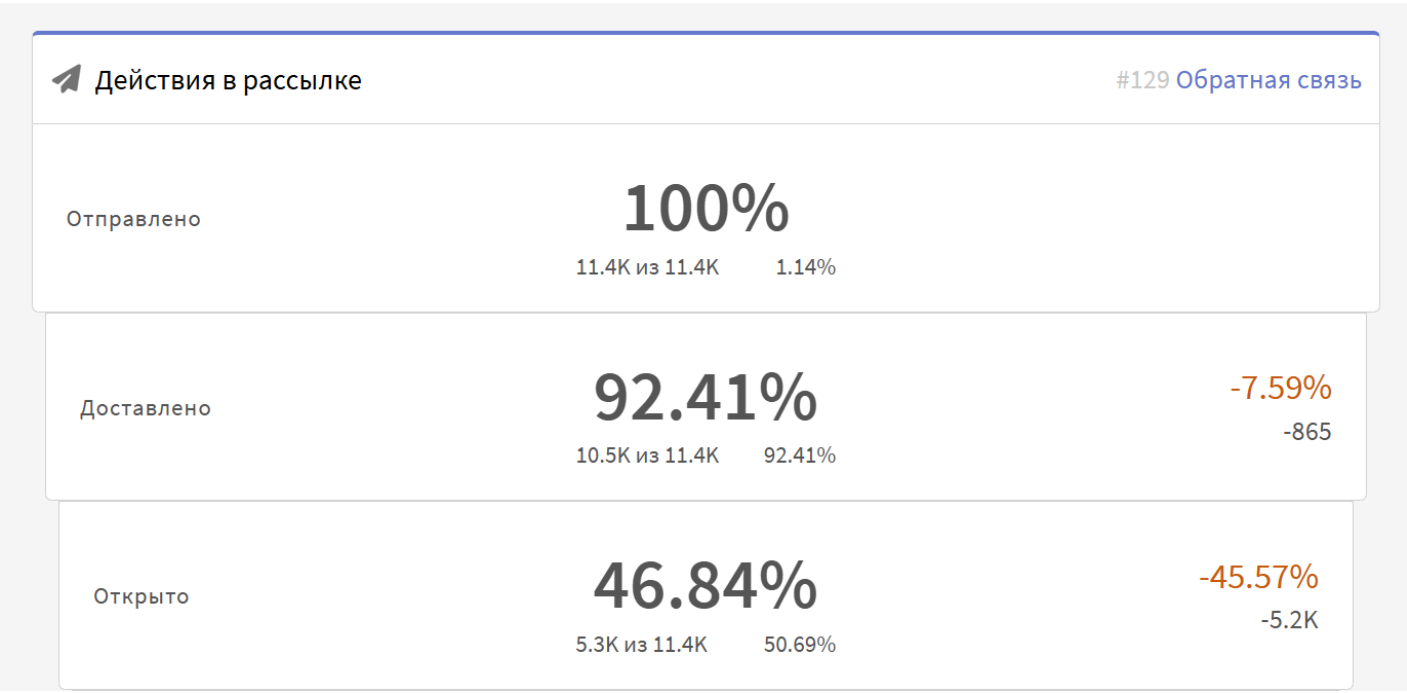
Информация по провайдерам появится в **таблице** под графиком:

Наименование	Отправки	Доставки	% Доставок	Открытия	Прочтения	% Прочтений	Клик по контенту	Уник. клики по контенту	Недоставки
Всего	766	703	91,78 %	351	266	37,84 %	1 057	151	63
2024-05-16	766	703	91,78 %	351	266	37,84 %	1 057	151	63

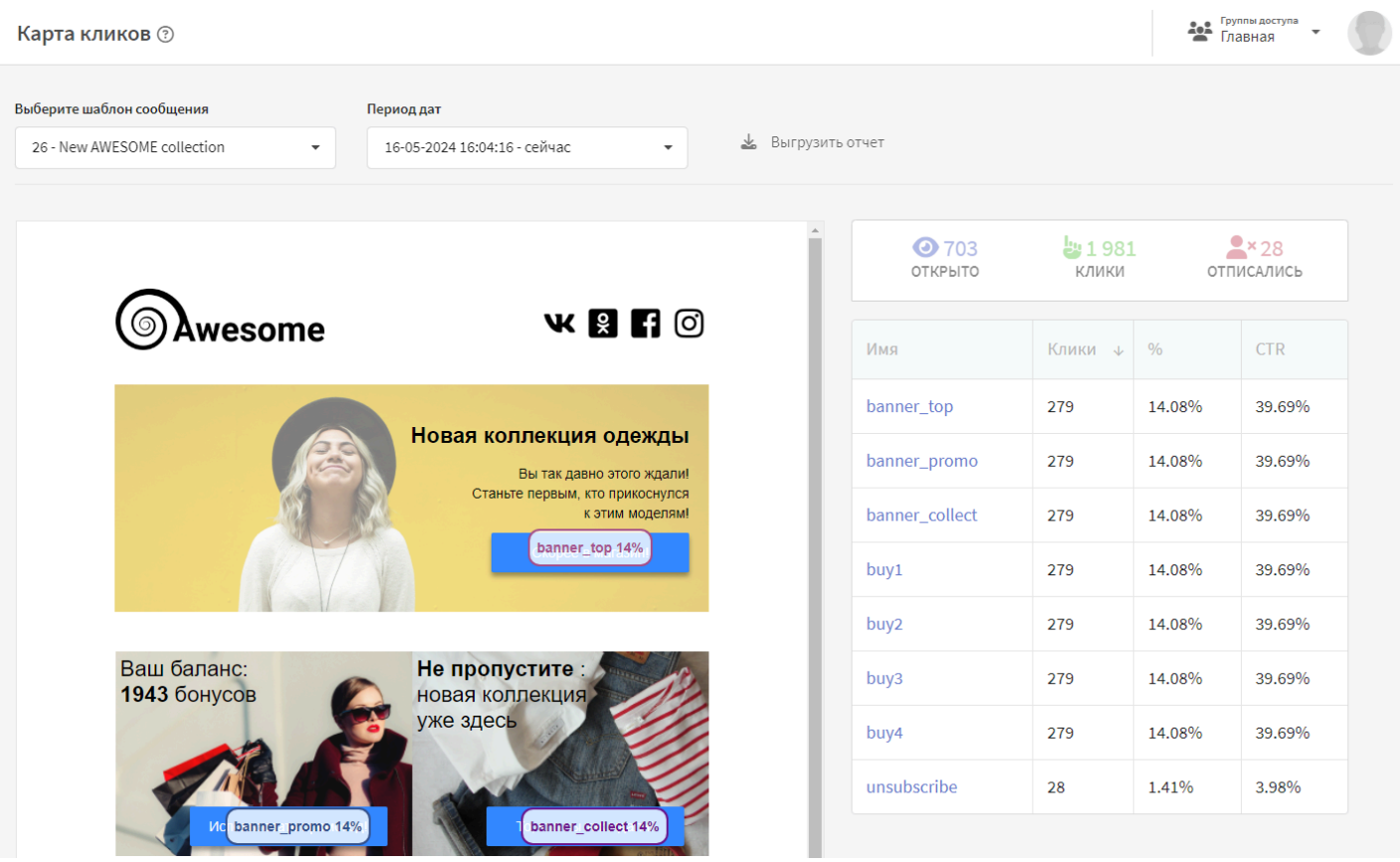
Для анализа проблем с доставляемостью Email используйте отчёты по **возвратам** и **недоставкам**:



Действия в Email-рассылке или шаблоне могут стать шагами в **интерактивной воронке конверсии**:



Для Email также доступна **карта кликов** — она составляется для отправленных хотя бы раз шаблонов с Email-версией. Когда вы вносите изменения в шаблон и отправляете снова — создаётся новая версия карты переходов, что позволяет вам оценить динамику поведения подписчиков:



SMS

Первая SMS-рассылка

SMS как канал связи с клиентами незаменим. Для него необязательно наличие интернет-соединения, а мобильный телефон сегодня есть у каждого. Процесс подписки также проще, чем у других каналов: достаточно номера телефона клиента, чтобы начать.

Это руководство поможет организовать SMS-рассылку в платформе Altcraft.

> **INFO**
>
> В SMS нет возможности добавить медиаконтент, однако можно добавить **ссылки** на ресурсы в сети Интернет. Altcraft Platform автоматически **укорачивает ссылки** в сообщении, а также **отслеживает клики** получателей.

Шаг 1: Подключите SMS шлюз

Платформа Altcraft Marketing использует интеграции с телеком-провайдерами (операторами мобильной связи).

> **TIP**
>
> Готовые интеграции перечислены в этой статье. Если вы используете другой шлюз отправки, напишите нам на team@altcraft.com и мы согласуем с вами сроки подключения нового шлюза.

Для подключения отправщика SMS потребуются **данные авторизации** телеком-провайдера, а также согласованный с ним **CALLER ID**.

> **CAUTION**
>
> Если вы пользуетесь **облачным решением**, передайте данные авторизации нашей службе поддержки для подключения интеграции.

Если вы пользуетесь Altcraft Marketing на собственном оборудовании по лицензии **On-Premise**, нужно будет настроить отправщик SMS в Панели администратора. Свяжитесь с **администраторами** вашей платформы, чтобы они, воспользовавшись инструкциями ниже, настроили:

- **Сендеры** — записи конфигурации, объединяющие параметры идентификации отправщика, скорости отправки и настройки поведения сендера при ошибках отправки.
- **Виртуальные сендеры** — правила отправки с нескольких сендеров. Могут быть настроены как в Панели администратора, так и в Интерфейсе пользователя.

> **TIP**

Или просто обратитесь в нашу службу поддержки и мы поможем вам настроить отправку SMS.

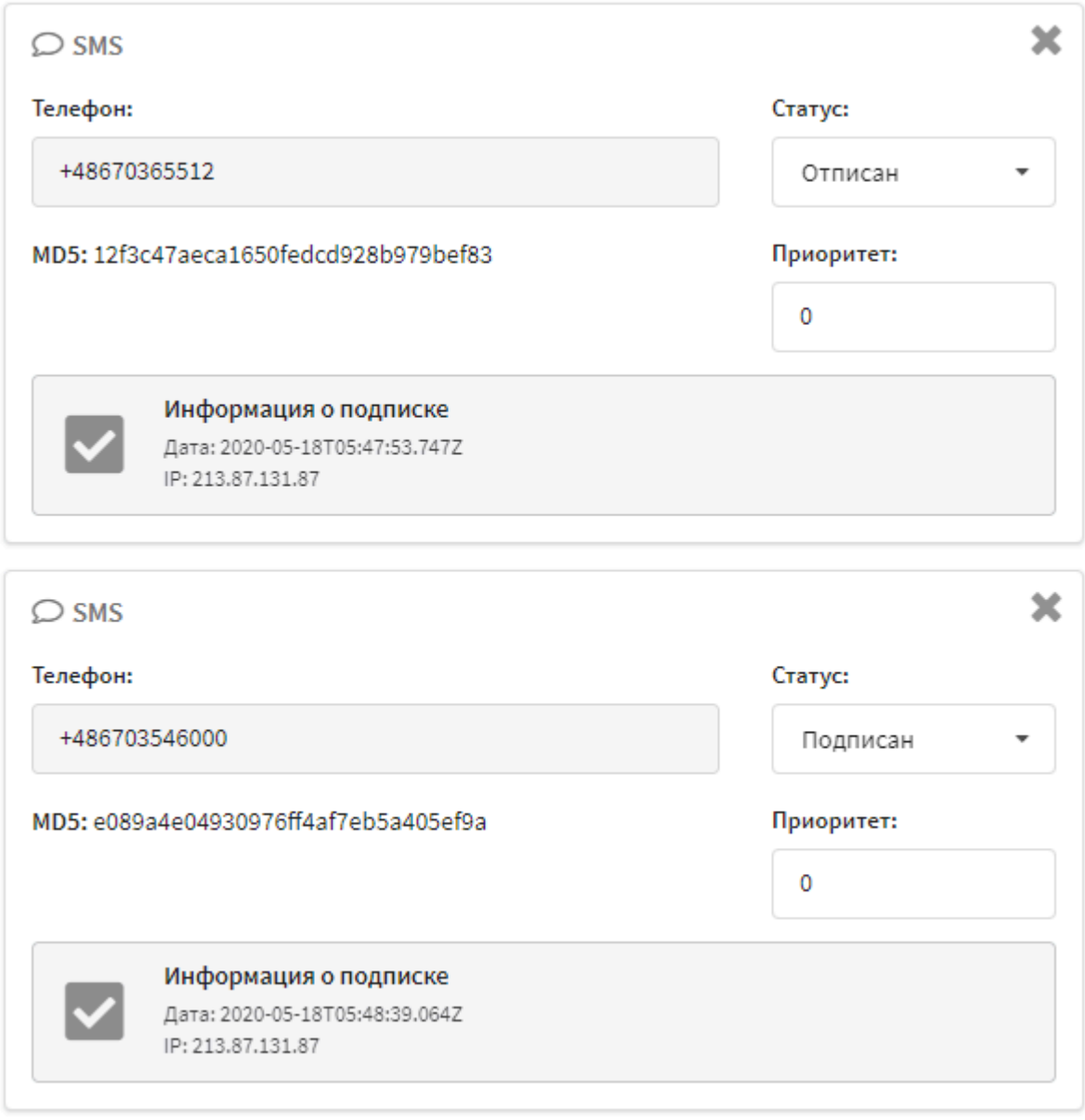
Шаг 2: Подпишите клиентов на SMS канал

Базы профилей

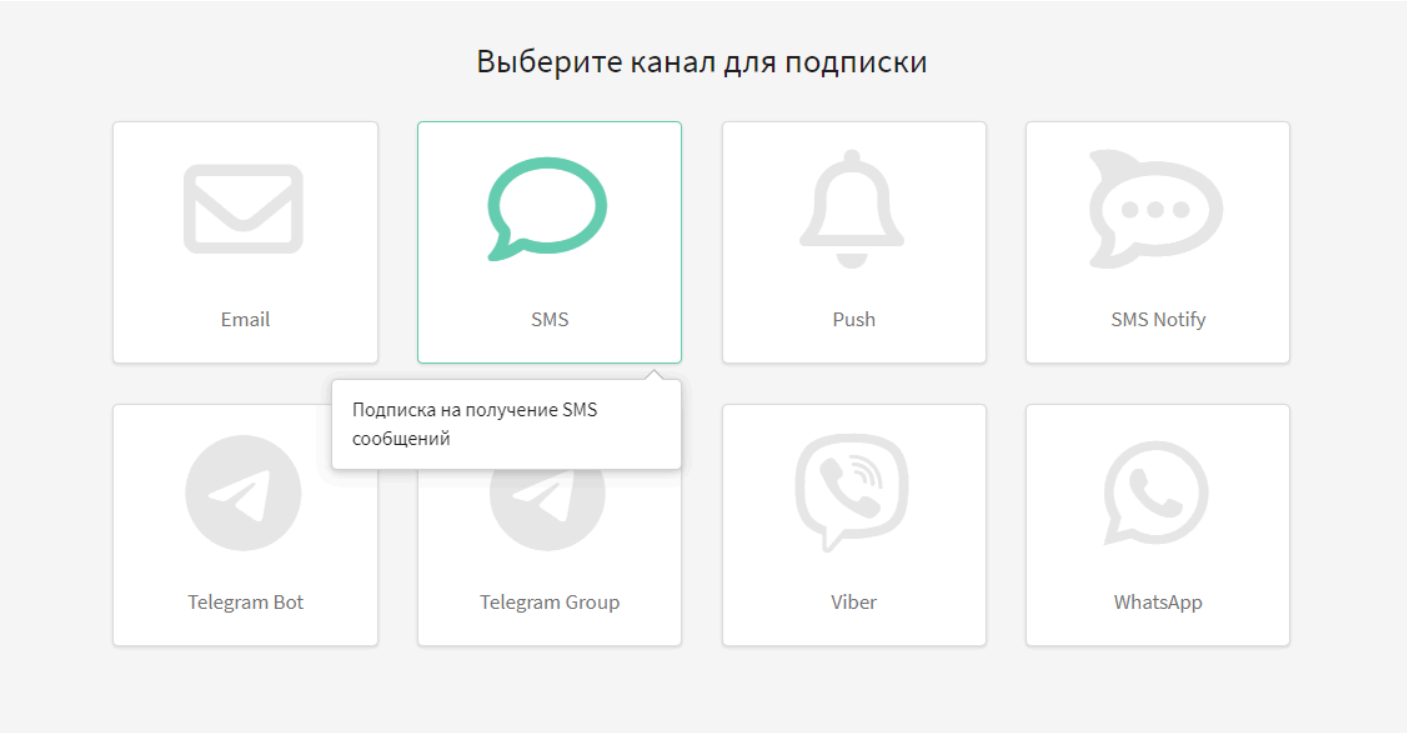
Рассылки Altcraft Marketing получают ваши подписчики. **Профили** подписчиков хранятся в **базах данных профилей**. У каждого профиля в базе данных есть стандартное поле **Телефоны**, в котором вы можете хранить номера телефонов своих клиентов. Однако, используется оно не для отправки, а для идентификации клиента в платформе.

Ресурсы подписок

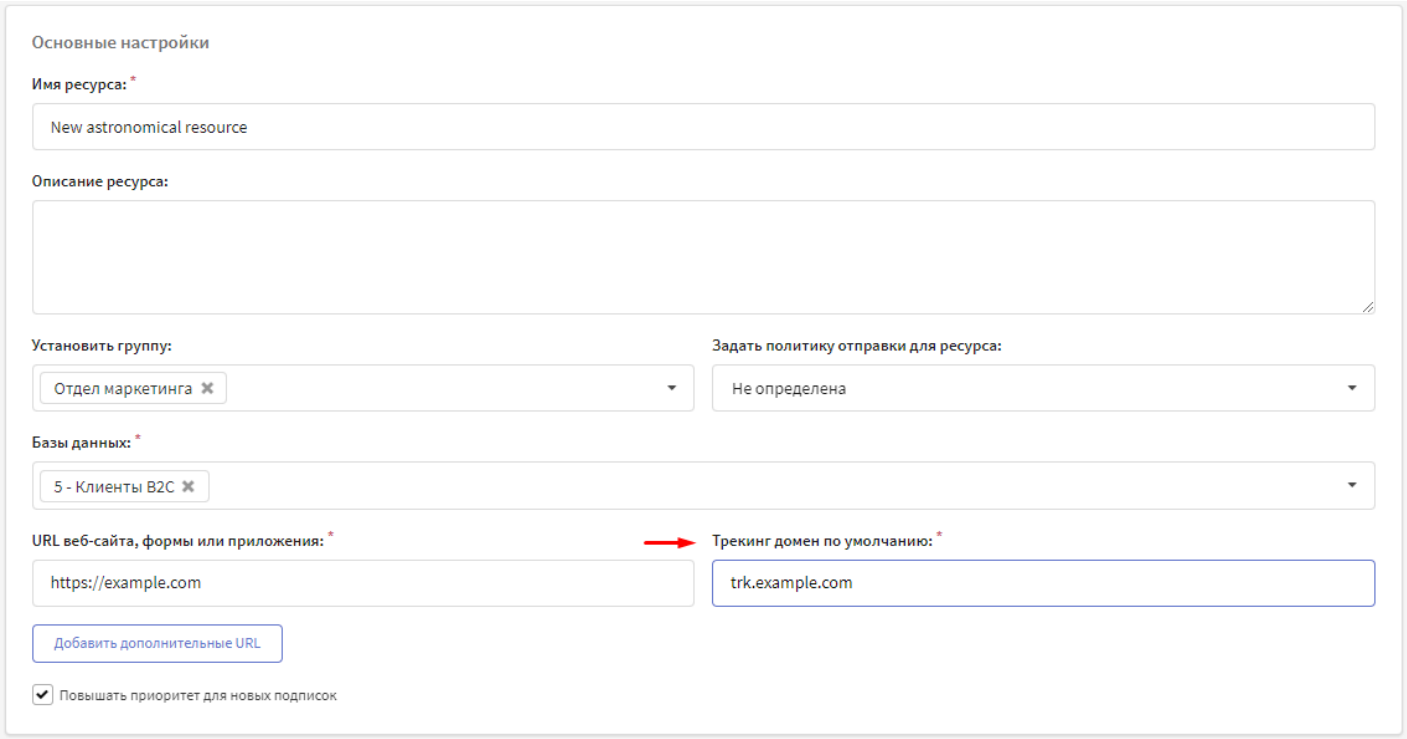
Рассылки при отправке SMS используют **подписки профиля** на **Ресурсы** — так вы можете оповещать клиента по нескольким номерам телефона. Ресурсом может быть отдельное направление маркетинга, сайт, приложение или тип контента.



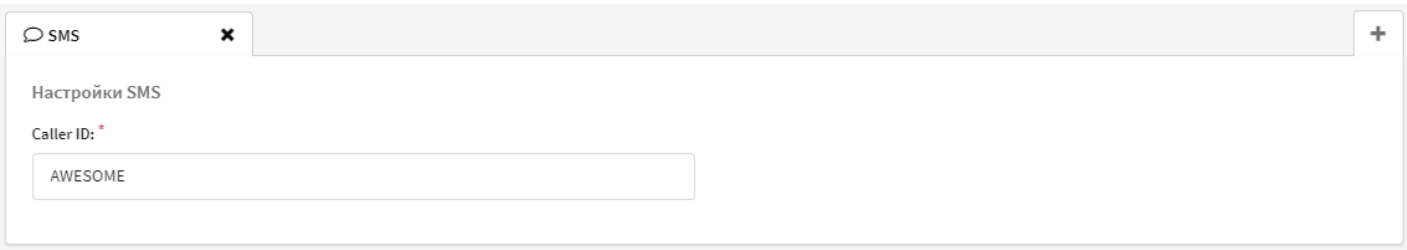
Выберите соответствующий пункт главного меню, создайте новый или выберите существующий **ресурс**. Добавьте **SMS канал** в существующий ресурс или выберите его в качестве основного канала для нового:



В основных настройках добавьте необходимые базы данных, при необходимости назначьте политику отправки. Укажите **Трекинг домен по умолчанию**. Он будет обрабатывать переходы ваших клиентов по ссылкам в SMS сообщениях:



В настройках SMS канала ресурса укажите **Caller ID**, согласованный с вашим провайдером телеком услуг:



Профили клиентов и подписка на ресурс

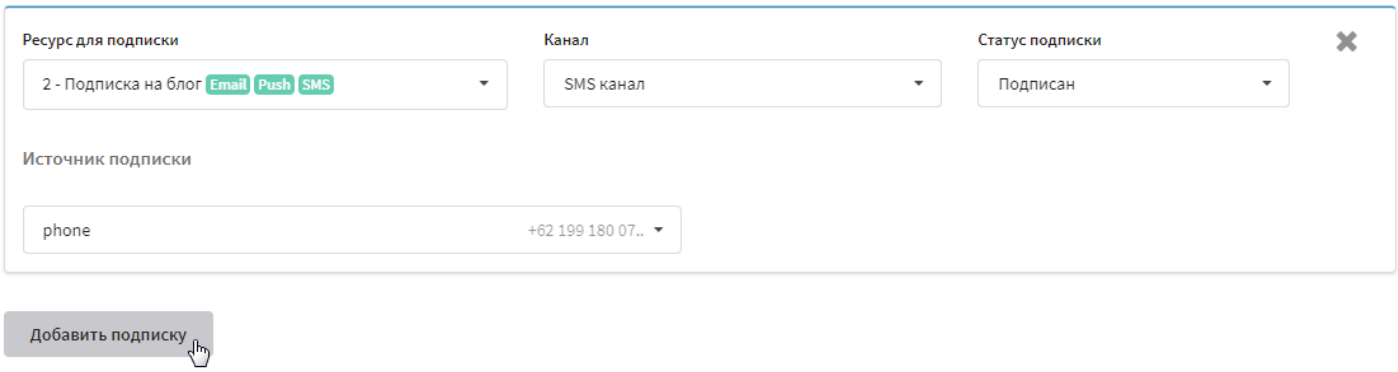
После того, как база профилей и ресурс подписок настроены, самое время импортировать в платформу данные клиентов. Для каждого клиента в Altcraft Marketing создаётся единый цифровой профиль, объединяющий личные данные, историю действий и подписки на ваши ресурсы.

Вы можете загрузить данные клиентов вручную: по одному или из табличного файла. Различные способы импорта описаны в этой статье.

Также вы можете настроить автоматический импорт профилей клиентов:

- по расписанию из вашей SQL базы данных
- посредством интеграции с API платформы

При импорте данных подпишите клиентов на подготовленный ранее ресурс. Для этого выберите **SMS** в качестве канала связи и укажите колонку с телефонами подписчиков:



TIP

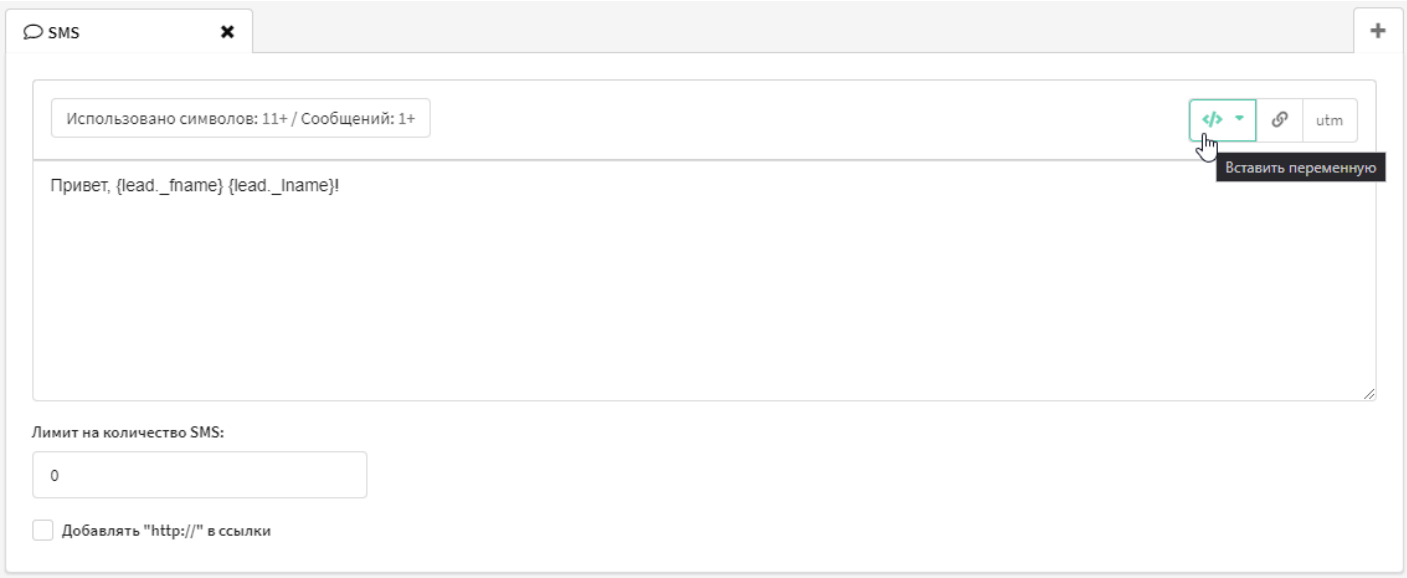
Если у вас уже есть база профилей, добавьте им подписку на новый ресурс. Есть два варианта:

- если у клиентов есть подписки на другой ресурс с актуальными номерами телефонов, используйте массовое обновление профилей.
- если для подписки на SMS, номера телефонов необходимо загрузить в платформу, используйте импорт профилей в режиме “только обновление”.

Шаг 3: Подготовьте контент для отправки

Контент для ваших SMS рассылок создаётся в **Шаблонах сообщений**. Шаблоны содержат контент для всех доступных каналов. При создании нового шаблона выберите SMS канал, или добавьте его для существующего.

SMS версия шаблона представляет собой простой текстовый редактор с возможностью добавлять данные конкретного клиента, а также динамический текстовый контент из внешних источников:



INFO: Обратите внимание

При использовании в шаблоне логических выражений или циклов платформа учитывает все символы форматирования, включая пробелы и переносы строк, как часть итогового сообщения. Каждый пробел или перенос строки добавляет к SMS один символ, что может повлиять на стоимость рассылки. Для экономии символов и более точного контроля над длиной сообщения записывайте логические конструкции в одну строку.

Например, вместо:

```
{if lead._city equal "Москва"}
Приглашаем на конференцию в Москве 30 июня!
{else}
Приглашаем на конференцию в вашем городе!
{end}
```

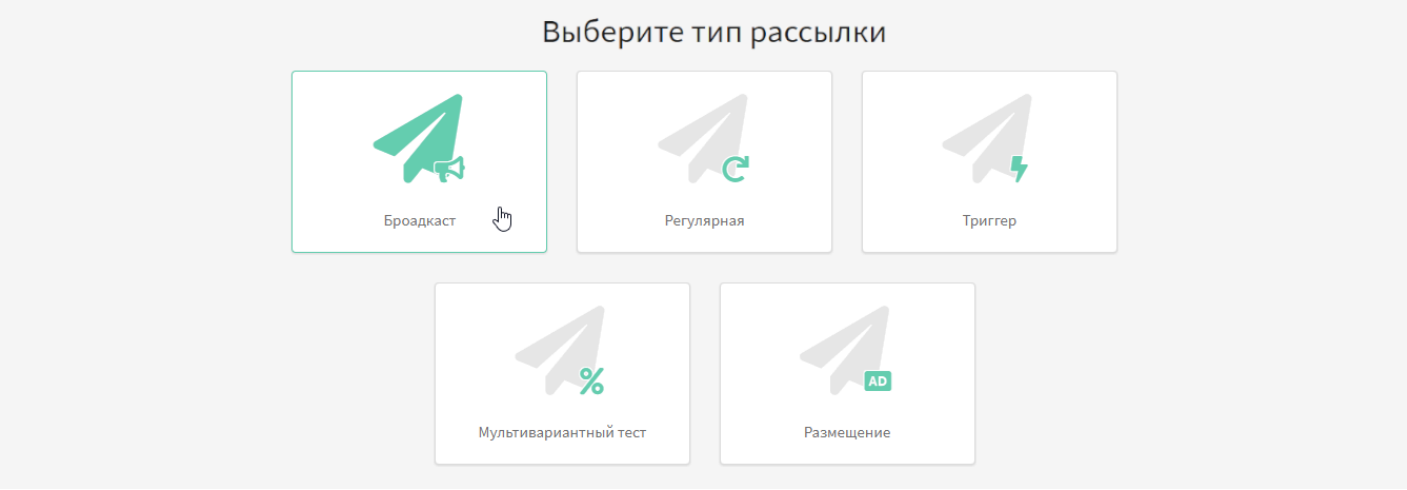
Используйте: ```text {if lead._city equal "Москва"} Приглашаем на конференцию в Москве 30 июня!{else} Приглашаем на конференцию в вашем городе!{end} ```

Такое форматирование не повлияет на логику работы, но сделает шаблон компактнее и поможет избежать неожиданного увеличения длины SMS.

Шаг 4: Выберите тип рассылки

	Для отправки SMS сообщений используется инструмент **Рассылка**. Если нужно произвести **разовую** рассылку по целой базе профилей или определённому сегменту аудитории — подойдёт броадкаст. Если же рассылка будет **периодической** — ваш выбор это регулярная рассылка. В главном меню выберите раздел **Рассылки** и откройте **Календарь**. В нём отображаются все рассылки, отправляемые по расписанию.

Определитесь с датой и создайте рассылку нужного типа:



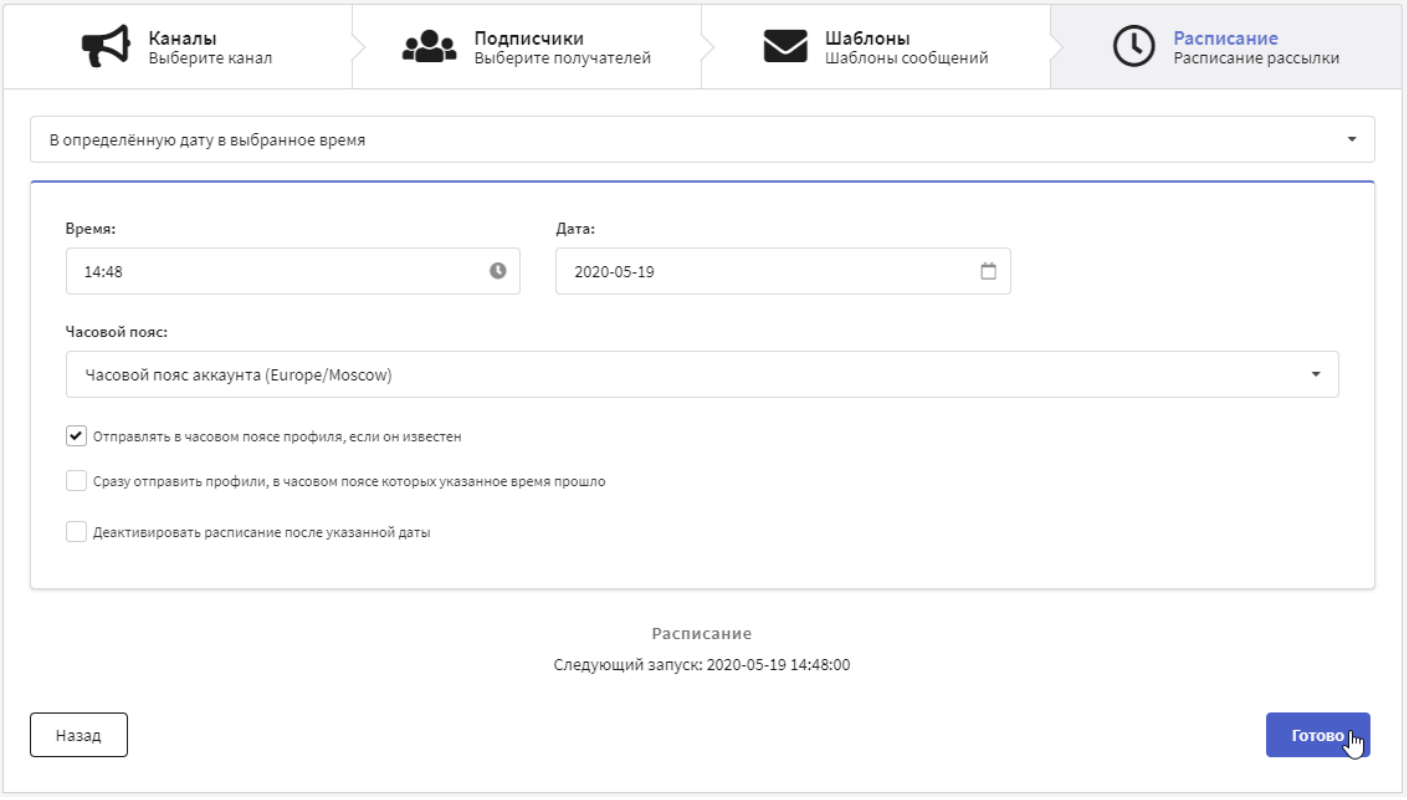
TIP

Подробную инструкцию по настройке **Броадкаста** вы найдёте на этой странице.

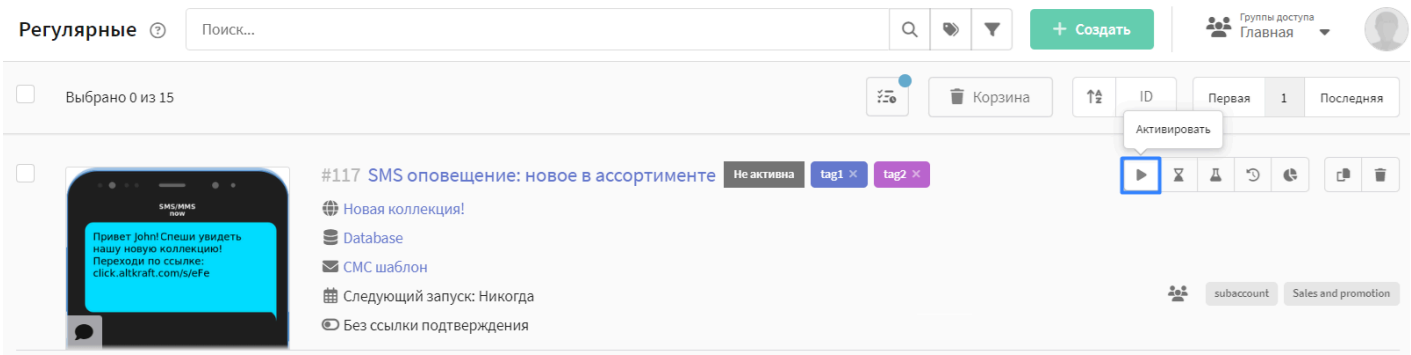
TIP

Подробную инструкцию по настройке **Регулярной рассылки** вы найдёте на этой странице.

Заполнив основные настройки рассылки, выберите **SMS** как канал связи, настроенный ранее ресурс подписок и аудиторию получателей. Добавьте в рассылку шаблон сообщения и установите расписание. Например, для броадкаста подойдёт вариант “В определённую дату в выбранное время”:



Когда рассылка настроена, сохраните её. Она будет отображаться в списке рассылок соответствующего типа. В правой части каждой записи есть меню быстрых действий. С его помощью вы можете сделать **тестовую** отправку — и если всё успешно, **активировать** рассылку:

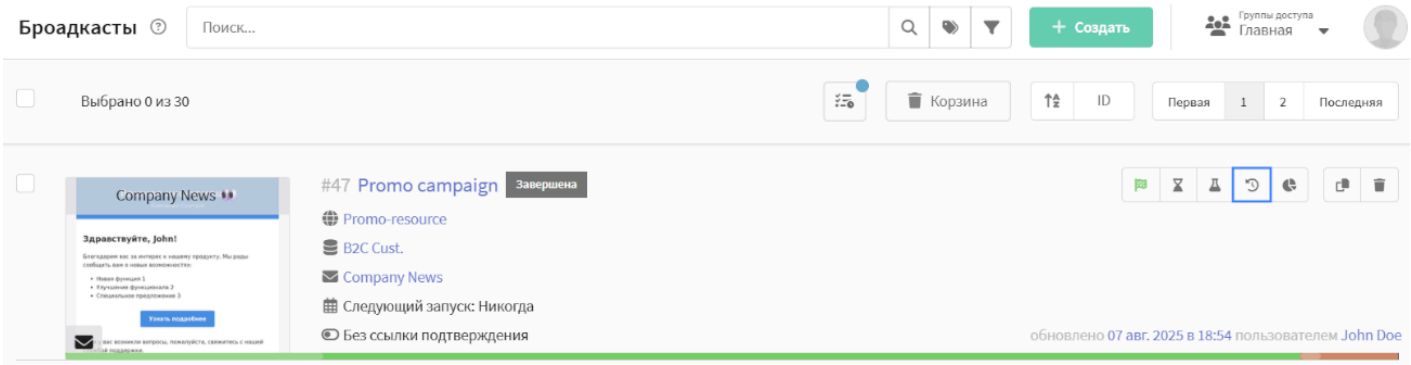


Шаг 5: Оцените эффективность рассылки

Журнал рассылки

Журнал рассылки — это инструмент, предоставляющий краткую сводку о ходе выполнения рассылки. В нем фиксируются ключевые события, параметры рассылки и статистика, что позволяет выявить возможные проблемы.

Чтобы открыть журнал рассылки, нажмите на кнопку **“Журнал”** в меню быстрых действий:



Журнал рассылки содержит следующую информацию:

Тип	Описание
Запуск рассылки	Время начала выполнения рассылки
Лимит	Показывает, установлен ли лимит на отправку сообщений
Источник подписчиков	Источник контактов: **List** — База данных **Segment** — Сегмент
ID сегмента / ID базы данных	Идентификатор сегмента или базы данных
Время получения сегмента	Время загрузки данных из источника (мс)
Количество профилей из источника	Общее число профилей в указанном источнике

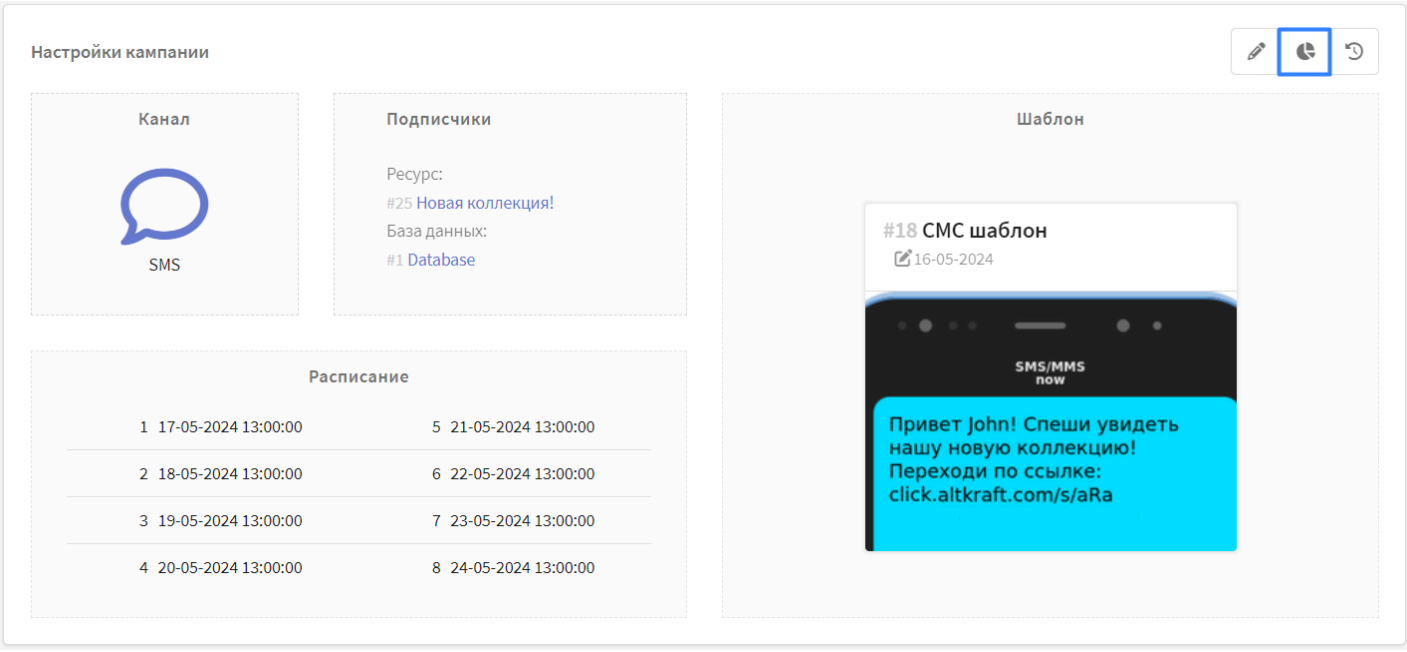
Тип	Описание
Количество найденных подписчиков	Число пользователей в указанном источнике с активной подпиской на выбранный ресурс
Количество дубликатов телефонов	Число повторяющихся номеров
Количество отправленных сообщений	Фактическое число успешных отправок
Рассылка завершена	Время окончания рассылки
Всего времени затрачено	Общая продолжительность рассылки от запуска до завершения (сек)

Отчеты и Аналитика

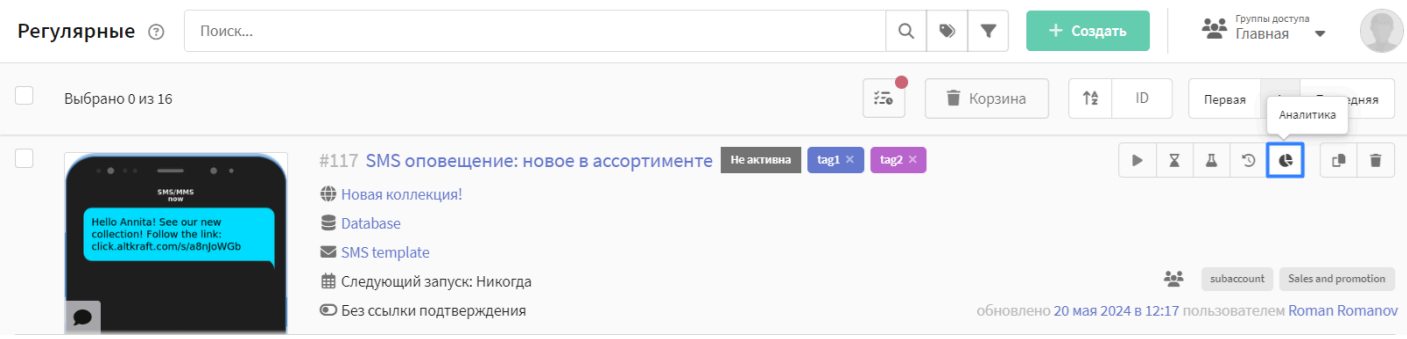
Для оценки результатов ваших рассылок воспользуйтесь отчетом по каналам. **Отчет по каналам** предоставляет аналитику по одной или нескольким рассылкам, по одному или нескольким каналам связи.

На платформе есть три способа получения аналитики по текущей рассылке:

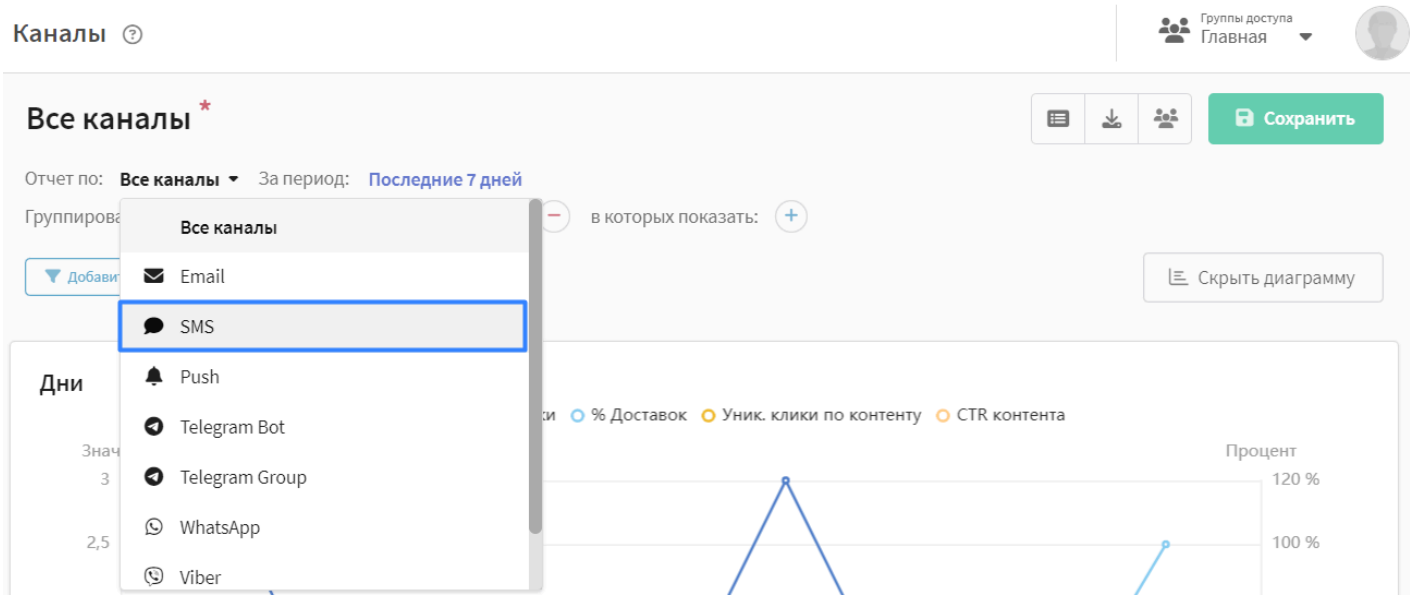
1. К отчёту по каналам конкретной рассылки можно перейти по ссылке в редакторе **рассылок**:



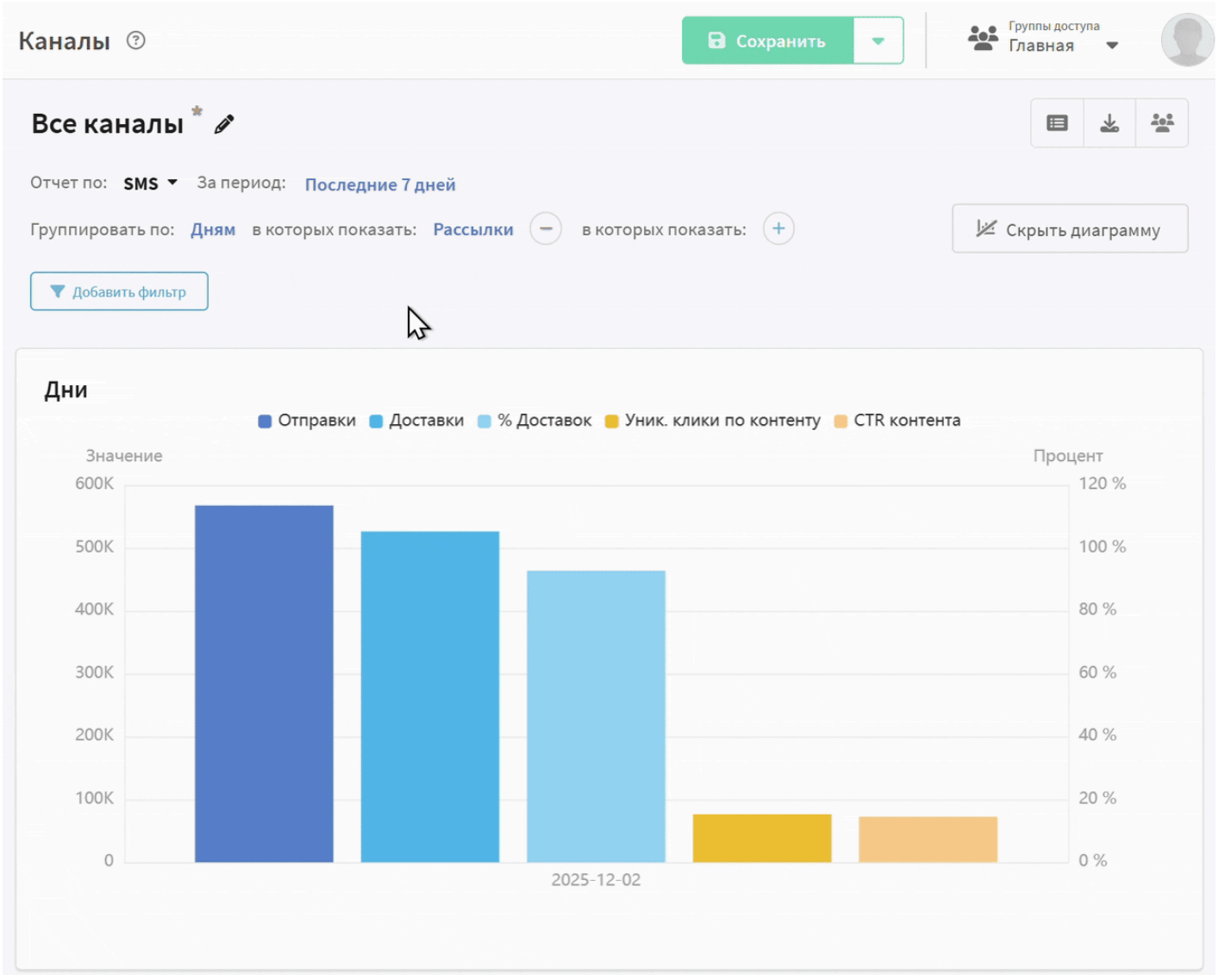
2. Также возможно получить статистику через меню быстрых действий:



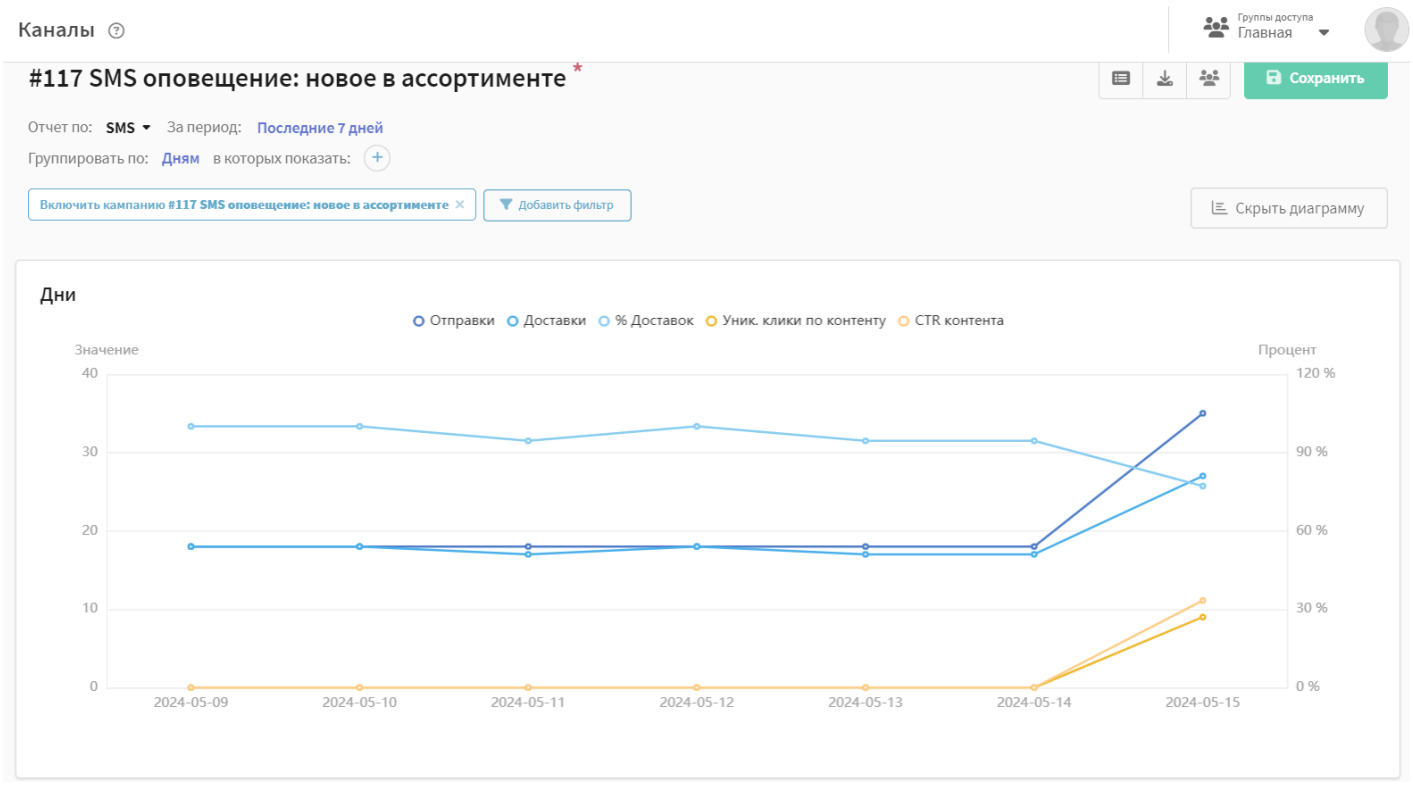
3. Выберите раздел **“Аналитика”** по **“Каналам”** в главном меню платформы. По умолчанию отобразится отчет по **“Всем каналам”**. Для группировки по SMS каналу выберите **“Отчет по”:**



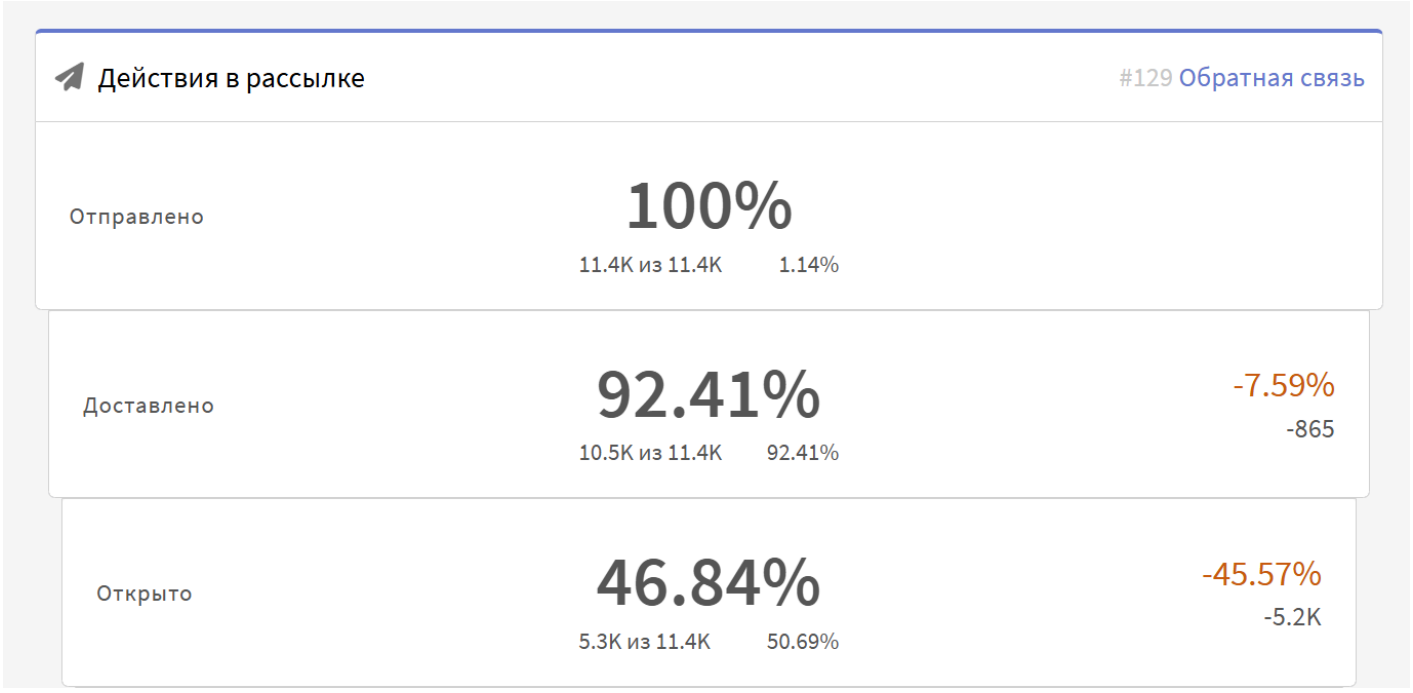
Следующим шагом **“Добавьте фильтр”** по **“Рассылкам”**. В отобразившемся списке выберите нужную рассылку.



В результате вы получите статистику по выбранной рассылке за последние 7 дней.



Действия в SMS-рассылке или шаблоне могут стать шагами в **интерактивной воронке конверсии**:



Web Push

Первая Web push-рассылка

Web push, или браузерные пуши — это уведомления, которые вы можете отправлять в десктопные и мобильные браузеры ваших клиентов. При подписке они отображаются по мере получения в центре уведомлений операционной системы — в углу или верхней части экрана — даже если окно браузера неактивно.

Это руководство поможет вам организовать рассылку **браузерных push-уведомлений** в платформе Altcraft Marketing.

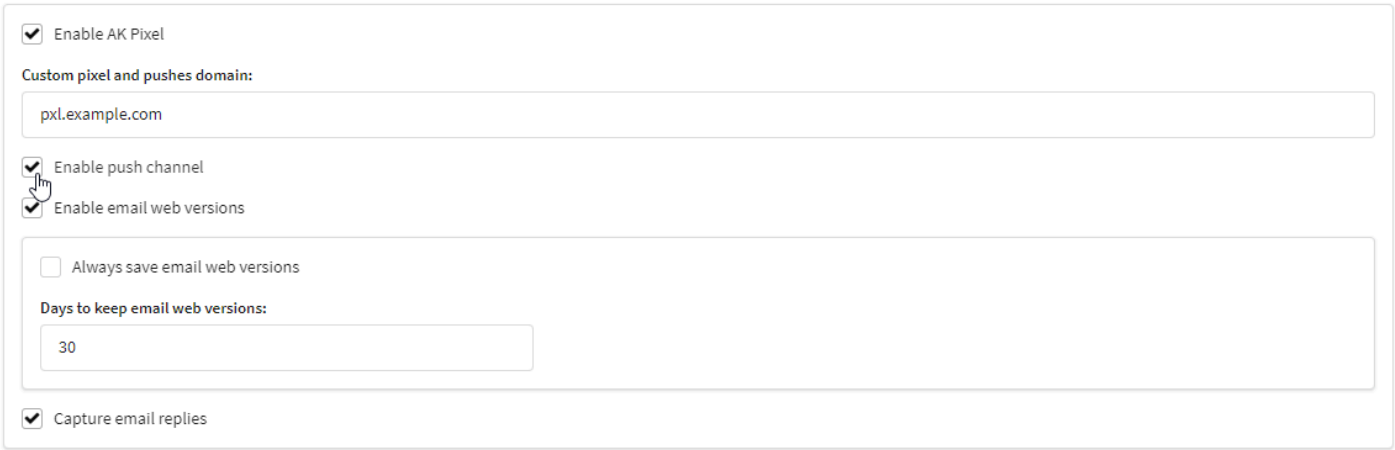
> **TIP**
>
> Push-уведомления гораздо выгоднее SMS, так как их отправка фактически ничего не стоит.

Шаг 1: Настройте домен для работы с Push

> **CAUTION**
>
> Если вы пользуетесь **облачным решением**, пропустите этот шаг и перейдите к настройкам подписки.
>
> При установке Altcraft Marketing на ваше оборудование **On-Premise** передайте инструкции из этого шага **администраторам** вашей платформы. Или свяжитесь с нашей службой поддержки!

Убедитесь, что **и ваш веб сайт, и платформа** работают **по протоколу HTTPS**. В системе Altcraft Marketing должны быть **настроены web-пиксели**.

Необходимо активировать **пиксели** и **push-уведомления** в Панели администратора для **аккаунтов**, с которых планируется отправка:

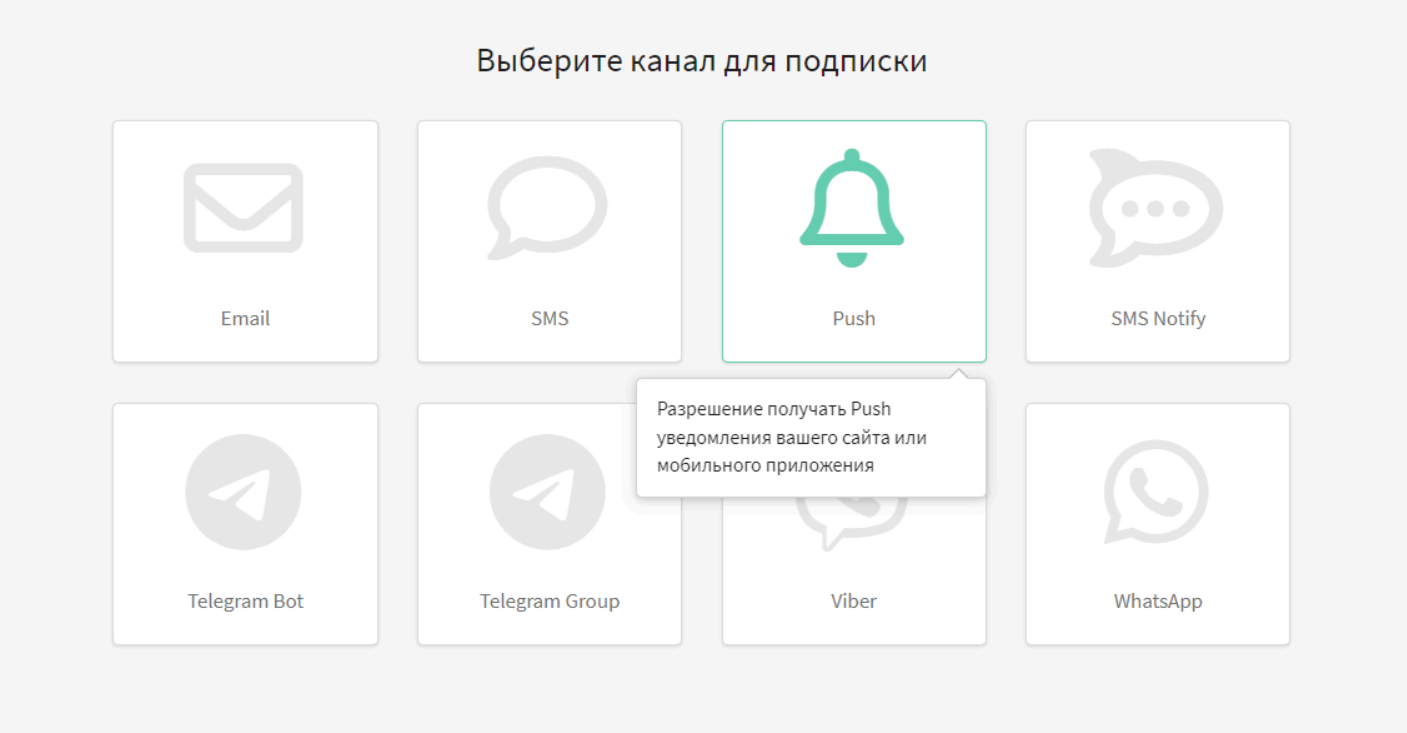


Шаг 2: Подпишите клиентов на Push-канал

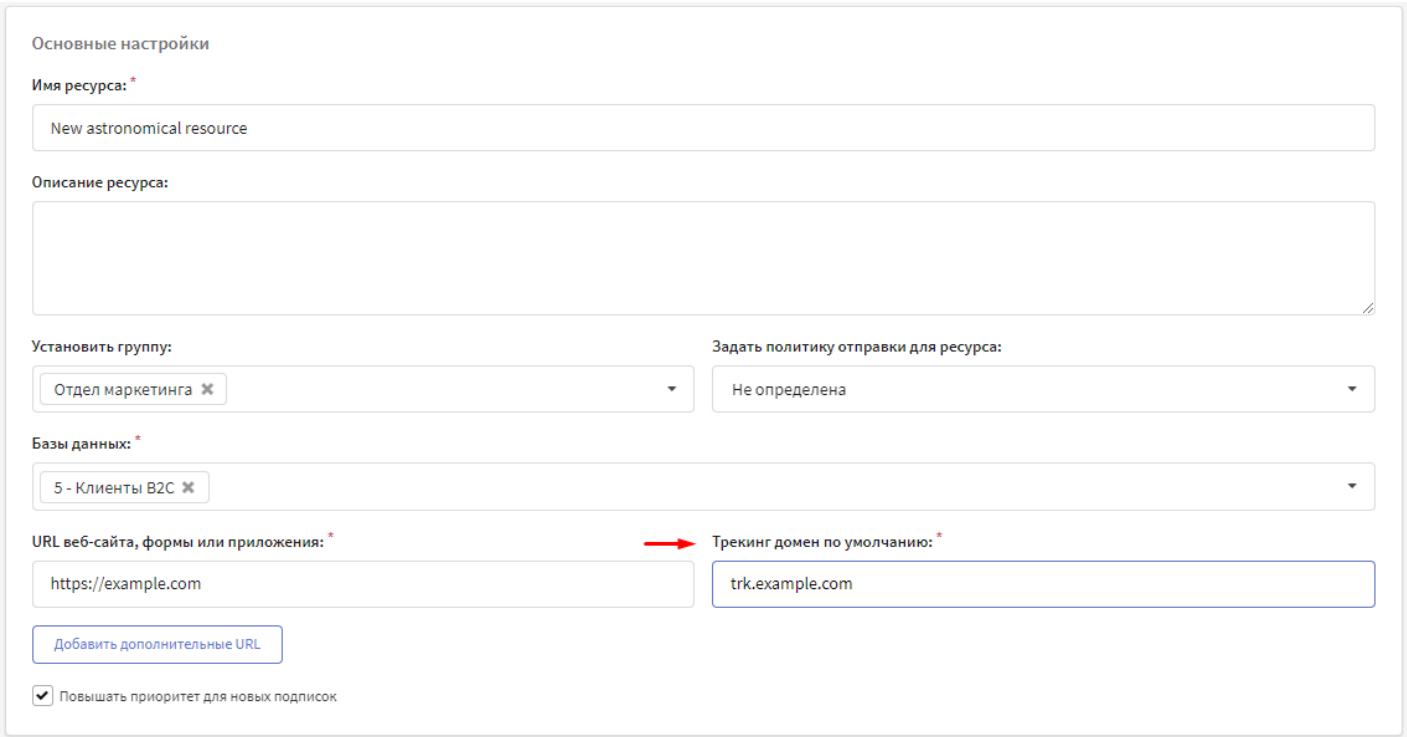
Рассылки Altcraft Marketing получают ваши подписчики. **Профили** подписчиков хранятся в **базах данных профилей**.

Рассылки при отправке push используют **подписки профиля** на **Ресурсы**: так вы можете отправлять уведомления на различные устройства клиента или в разные браузеры. Ресурсом может быть отдельное направление маркетинга, сайт, приложение или тип контента.

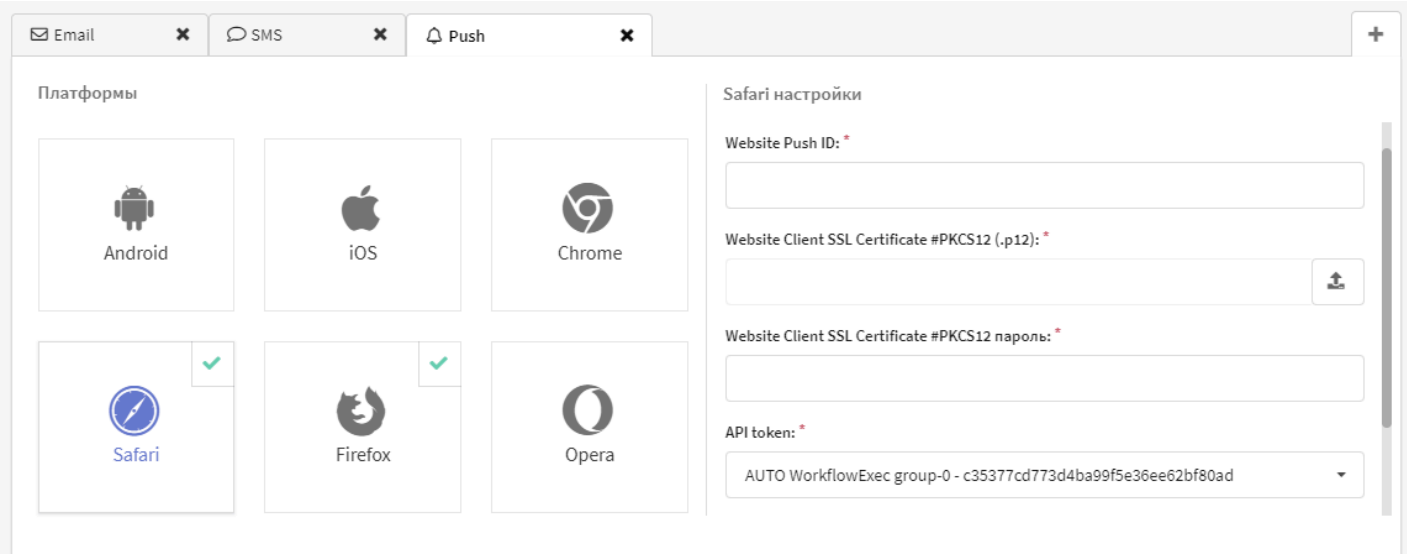
Выберите в главном меню раздел **“Ресурсы”**, создайте новый ресурс или выберите существующий. Добавьте **push-канал** в существующий ресурс:



В основных настройках добавьте необходимые базы данных, при необходимости назначьте политику отправки и укажите **Трекинг домен по умолчанию**. Он будет обрабатывать переходы ваших клиентов по ссылкам в сообщениях:



В настройках push-канала ресурса выберите **браузеры**, в которые собираетесь отправлять уведомления клиентам. После этого внесите соответствующие настройки:



Для отправки уведомлений в разные браузеры используются разные push-сервисы:

Сервис	**Какие браузеры используют данный сервис для отправки Web push уведомлений?**
Google Firebase	Google Chrome, Mozilla Firefox, Opera, Яндекс.Браузер, Samsung Internet Browser
Apple Push Notification Service	Apple Safari
Mozilla Services	Mozilla Firefox

Процесс настройки для различных платформ описан в **этом руководстве**.

Профили клиентов и подписка на ресурс

Для каждого клиента в Altcraft Marketing создаётся единый цифровой профиль, объединяющий личные данные, историю действий и подписки на ваши ресурсы. Для отправки уведомления конкретному получателю используется токен (идентификатор) устройства. В случае с Web push этот идентификатор связан с браузером, в котором клиент разрешил показ уведомлений.

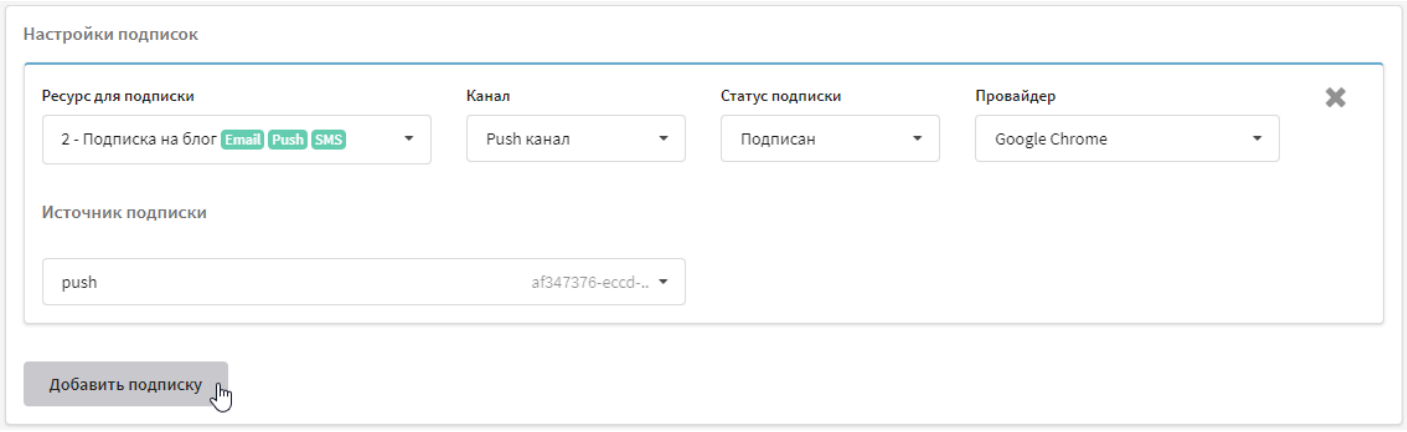
Существующая база подписчиков

Если у вас есть база токенов для ваших клиентов, вы можете загрузить эти данные в платформу и использовать для отправки рассылок: по одному клиенту, списком или из табличного файла. Различные способы импорта данных описаны в этой статье.

Также вы можете настроить автоматический импорт профилей клиентов:

- по расписанию из вашей базы данных SQL
- посредством интеграции с API платформы

При импорте данных подпишите клиентов на подготовленный ранее ресурс. Для этого в разделе **“Настройки подписки”** выберите Push в качестве канала связи, выберите провайдера уведомлений, в поле **“Источник подписки”** укажите колонку с токенами устройств:



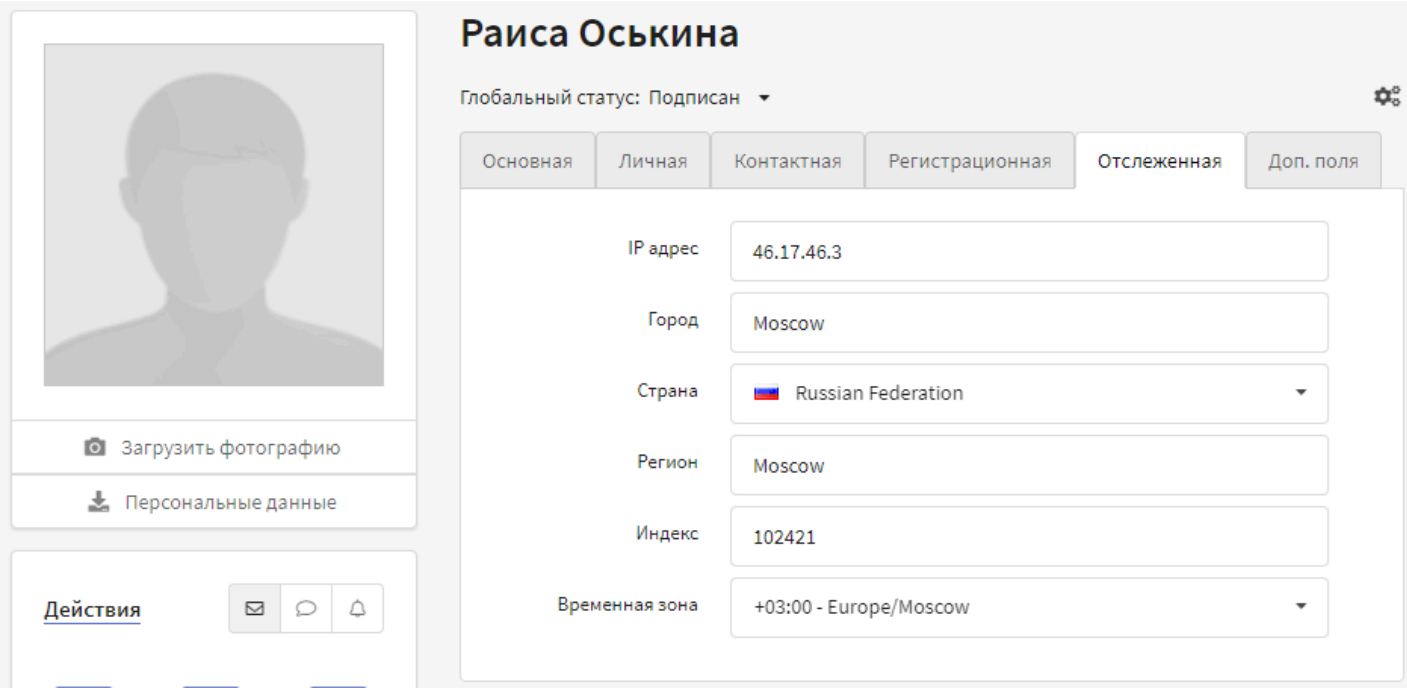
TIP

Если у вас уже есть база профилей, добавьте им подписку на новый ресурс. Есть два варианта:

- если у клиентов есть подписки на другой ресурс с актуальными токенами устройств — используйте массовое обновление профилей;
- если для подписки на Push-уведомления идентификаторы необходимо загрузить в платформу — используйте импорт профилей в режиме “только обновление”.

Новые push-подписчики сайта

Altcraft Marketing импортирует профили посетителей вашего сайта, когда те разрешают показ push-уведомлений. При этом в профиль записывается информация об устройстве подписчика, IP адрес и определённая по нему геолокация:

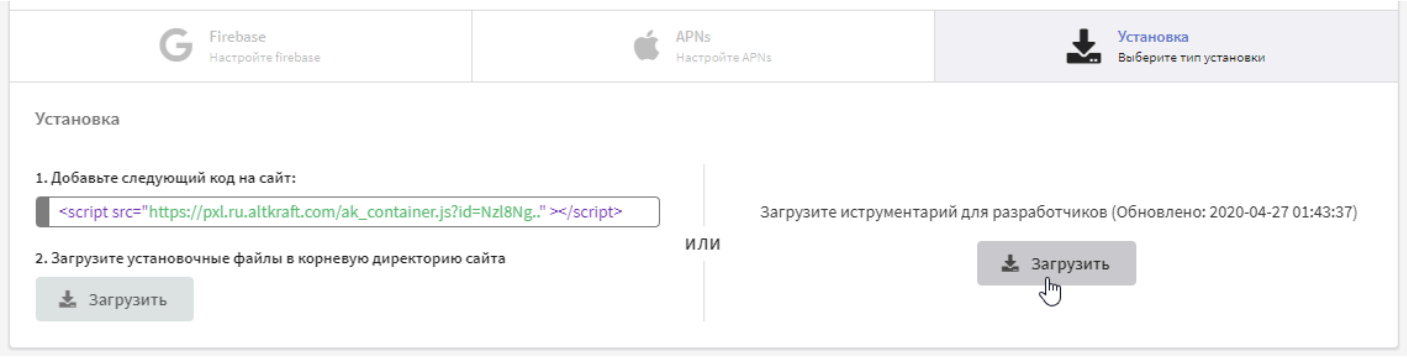


Для настройки импорта новых push-подписчиков перейдите к следующему шагу.

Шаг 3: Установите инструменты push на ваш сайт

Как только вы сохраните ресурс, у вас появится возможность загрузить инструментарий разработчика — **Web Push SDK.**

> **TIP**
>
> Скачайте и передайте SDK Toolkit разработчикам вашего сайта для размещения на небходимых страницах. Передайте им также ссылку на эту статью или на статью по конфигурации web push.



Установка нужных файлов на сайт возможна в двух вариантах:

1. Добавление кода на сайт и загрузка установочных файлов в корневую директорию сайта
2. Загрузка инструментария для разработчиков

Способ 1: Добавление кода на сайт и загрузка установочных файлов в корневую директорию сайта

Алгоритм действий для данного варианта указан в левой части. Добавьте указанный код на сайт, затем сохраните и загрузите в корневую директорию сайта установочные файлы:

- **manifest.json** — данные о проекте для отправки push через Google Chrome;
- **service-worker.js** — service worker, который регистрируется в браузерах пользователей и принимает push сообщения.

Здесь файл **service-worker.js** является **динамическим**. Он автоматически обновляется каждый раз, когда вы вносите изменения в настройки ресурса.

```
importScripts('https://pxl.test.com/service-worker.js?id=MXw3');
```

Способ 2: Загрузка инструментария для разработчиков

Алгоритм действий для данного варианта указан в правой части. Нажмите на кнопку “**Загрузить**”, после чего будет загружен архив Toolkit. Распакуйте архив и разместите файлы Toolkit в корневой директории вашего сайта.

Toolkit состоит из четырёх файлов:

- **README.txt** — инструкция по установке Toolkit;
- **ak-push.js** — скрипт, который показывает пользователям предложение подписаться на push-оповещения;

- **manifest.json** — данные о проекте для отправки push через Google Chrome;
- **service-worker.js** — service worker, который регистрируется в браузерах пользователей и принимает push-сообщения.

Здесь файлы **service-worker.js** и **ak-push.js** является **статичным**. При использовании данного способа после изменения настроек ресурса необходимо заново загрузить Toolkit и обновить файлы на сайте.

TIP

При использовании инструментария для разработчиков можно **изменить путь до файла service-worker.js**. Для этого откройте файл **ak-push.js**, найдите свойство **swPath** и отредактируйте его, указав нужный путь до файла.

Добавление скриптов на сайт

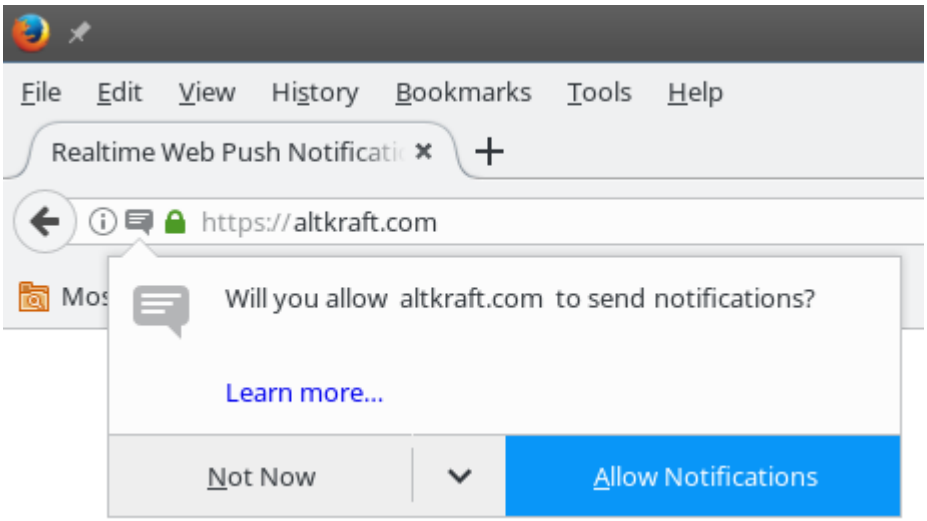
После того как Toolkit загружен на сайт, добавьте код, который будет показывать окно подписки для необходимого действия.

Ниже приведен пример кода для кнопки **“Подписаться”**. Этот код запускает библиотеку и инициализирует подписку:

```
<button id="init_sub">Подписаться на уведомления</button> // Кнопка подписки

// ..Основная часть кода..

<script>
        document.getElementById('init_sub').addEventListener('click', function() { // По
клику на кнопку..
        try {
                const akPush = new AKPush();
                akPush.initSubscription() // ..показать форму подписки
                }
        catch (e) {
                console.log(e); // ..или записать ошибку в консоль браузера
                }
        });
</script>
```



TIP

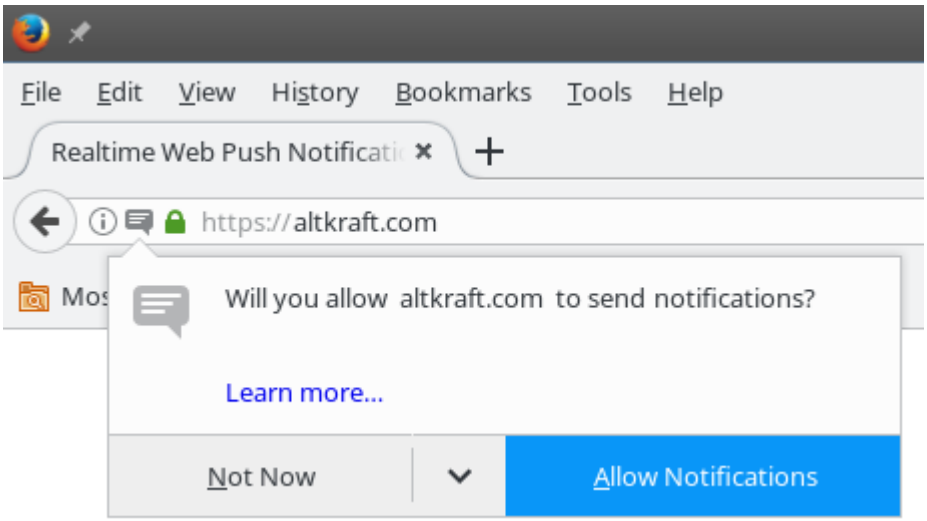
Через платформу Altcraft вы можете создать собственный **попап** с предложением подписки. Подробнее прочесть об этом можно в этой статье.

Код, описанный выше, работает по следующей логике:

- Пользователь нажимает на кнопку;
- Вызывается метод `initSubscription`;
- Всплывает системное окно браузера с предложением подписаться на уведомления этого сайта.

Пользователь может настроить свою логику и решить, каким путем посетитель сайта дойдет до системного окна подписки. Стили системного браузерного окна с запросом разрешений на push-уведомления изменить нельзя.

Как только все будет настроено, можно будет зайти на сайт и подписаться на push-уведомления. В браузере клиента при подтверждении устанавливается Service worker, отображающий сообщения даже при свернутом окне браузера:



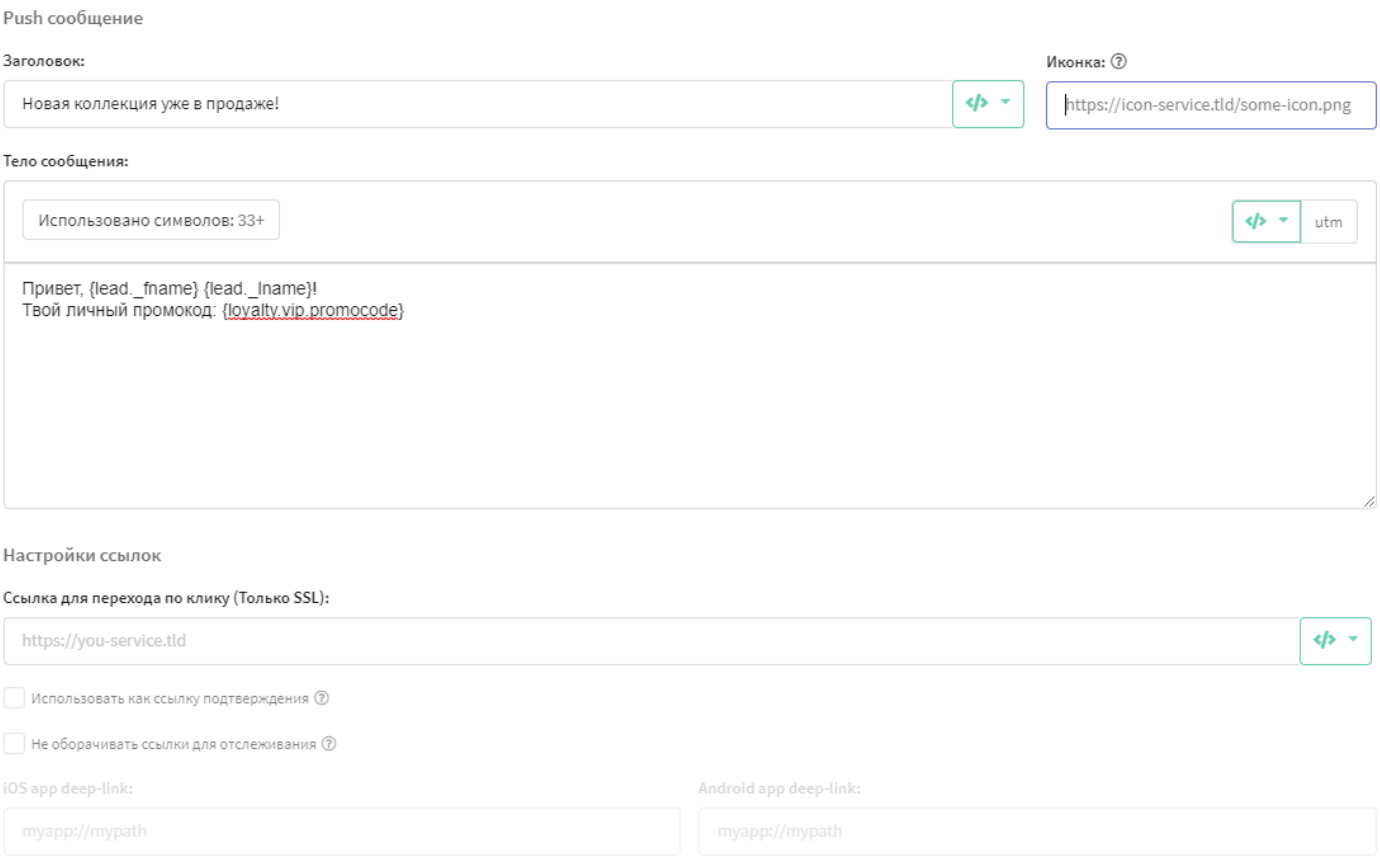
Шаг 4: Подготовьте контент для отправки

Контент для ваших push-рассылок создаётся в **Шаблонах сообщений**. Шаблоны содержат контент для всех доступных каналов. При создании нового шаблона выберите push-канал или добавьте его для уже существующего шаблона.

Push-версия шаблона содержит:

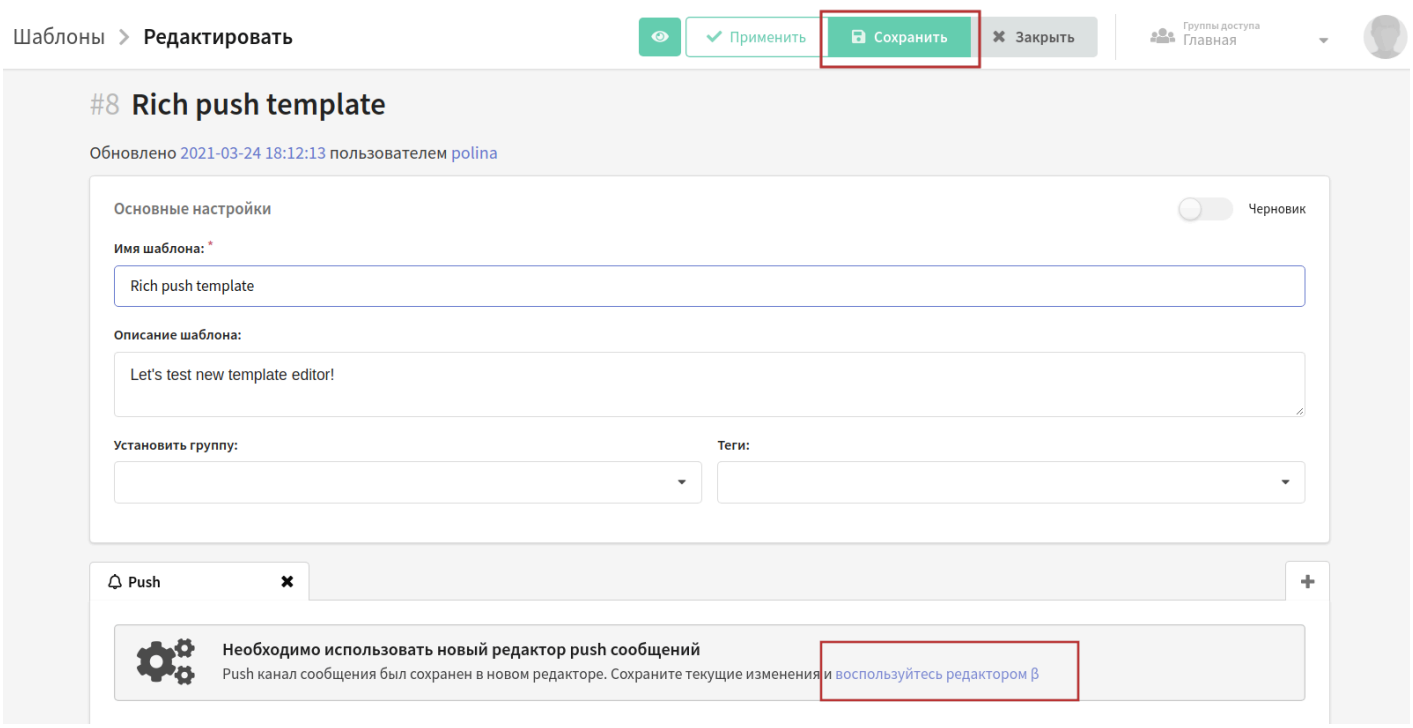
- **Заголовок** — аналогичен теме Email-письма.
- **Иконка** — любое изображение для отображения в уведомлении. Для **Safari** иконка настраивается одна для всех рассылок.
- **Тело сообщения** — основная текстовая часть сообщения.
- **Ссылка для перехода** — ссылка, по которой перейдёт подписчик при нажатии на push.

В заголовок, ссылку и тело сообщения вы можете добавить данные конкретного клиента, а также динамический текстовый контент из внешних источников:



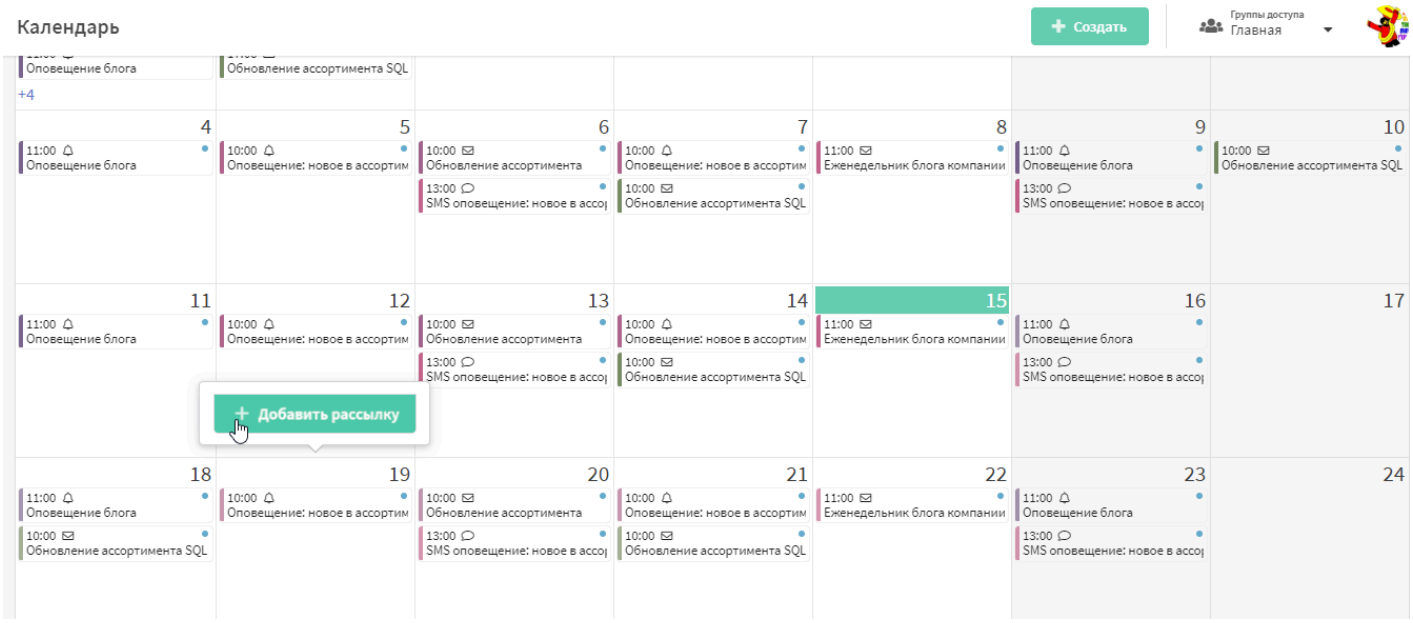
Если вы рассылаете Rich push-уведомления, то для создания шаблона вам стоит воспользоваться специальным **редактором**. В Rich push-уведомления можно добавлять не только текст и иконку, но и кнопки, изображения, банеры и даже аудио и видеофайлы. Более того, новый редактор позволяет отдельно настраивать уведомление для определенной ОС, а также добавлять адаптивные ссылки и deeplink.

Чтобы открыть новый редактор, зайдите в обычный редактор шаблонов и сделайте начальную версию push-сообщения. Как это сделать, мы описали выше. После этого нажмите кнопку **Сохранить**. Теперь вернитесь в созданный шаблон и кликните **“воспользуйтесь редактором β”**, чтобы перейти в новый редактор.

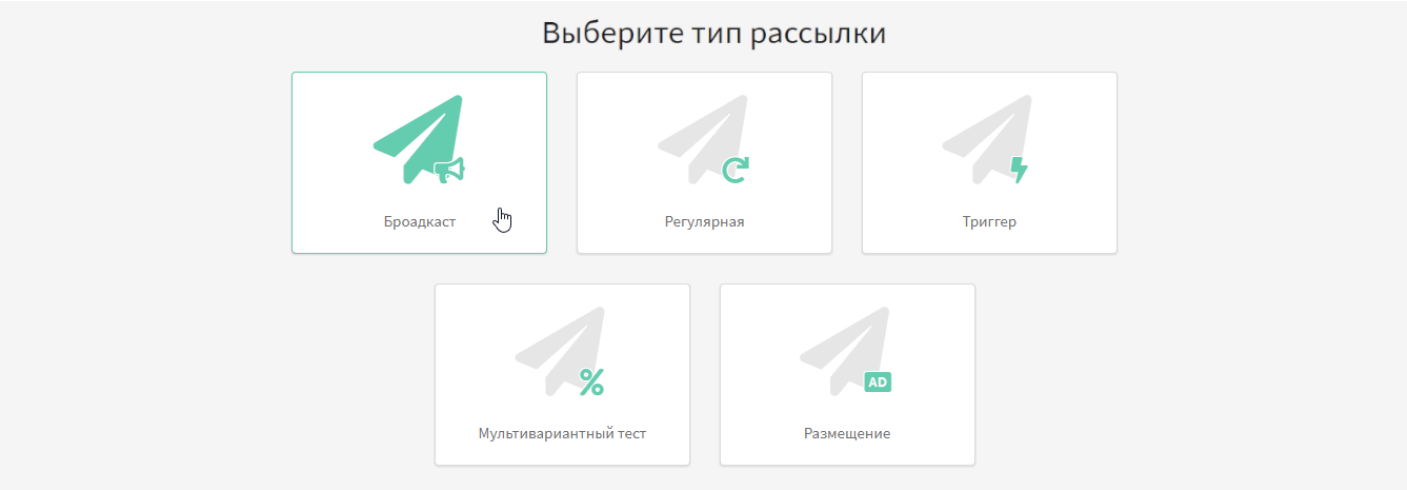


Шаг 5: Выберите тип рассылки

	Для отправки Push-сообщений используется инструмент **Рассылки**. Если вам нужно произвести **разовую** рассылку по целой базе профилей или определённому сегменту аудитории — подойдёт броадкаст. Если же рассылка будет **периодической** то вам следует использовать **регулярную рассылку**. В главном меню выберите раздел **Рассылки** и откройте **Календарь**. В нём отображаются все рассылки, отправляемые по расписанию.



Определитесь с датой и создайте рассылку нужного типа:



TIP

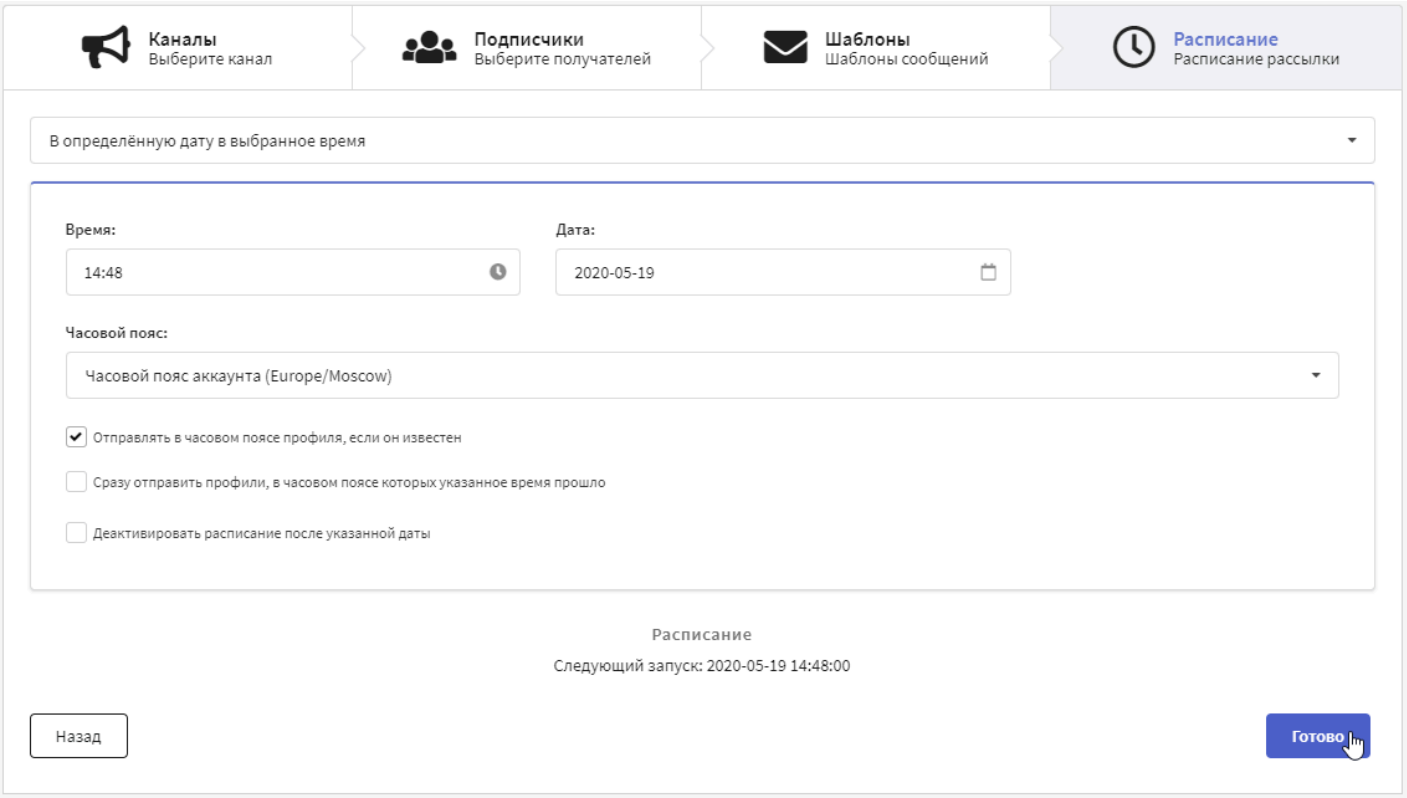
Подробную инструкцию по настройке **Броадкаста** вы найдёте на этой странице.

TIP

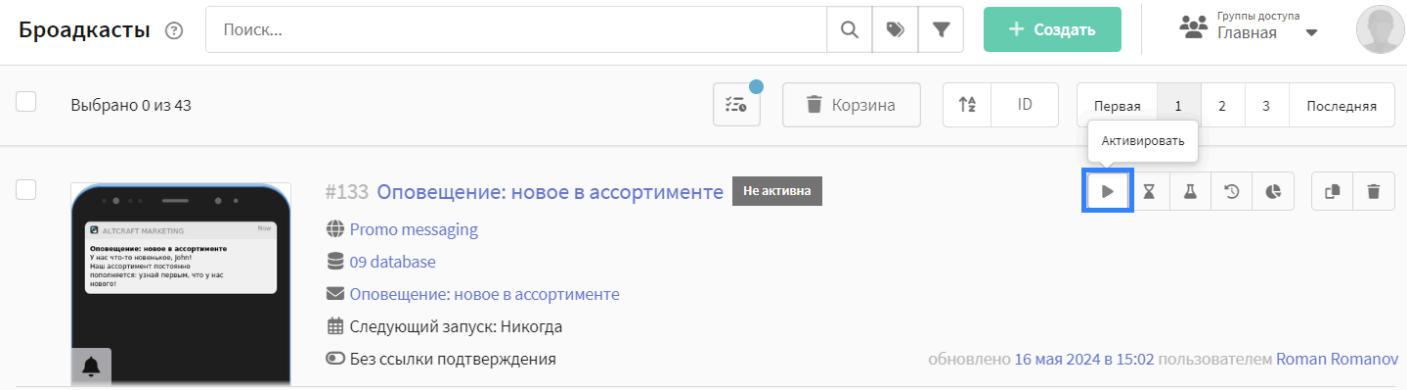
Подробную инструкцию по настройке **Регулярной рассылки** вы найдёте на этой странице.

Заполнив основные настройки рассылки, вам необходимо:

1. Выбрать **Push** как канал связи.
2. Выберите провайдеров для рассылки и режим отправки. Есть два вида отправки — **fanout**, рассылка на все подписки с учетом фильтров, или **last**, отправка на подписку с самой свежей датой создания. Если несколько подписок были созданы одновременно, то из них будет выбрана одна случайная. Если в рассылке заданы фильтры по провайдеру, дополнительным полям или категориям, то сначала накладываются фильтры, а потом уже ищется самая свежая подписка.
3. Выбрать настроенный ранее **ресурс** подписок и **аудиторию** получателей.
4. Добавить в рассылку **шаблон** сообщения.
5. Установить **расписание**. Например, для броадкаста подойдёт вариант “В определённую дату в выбранное время”:



Когда рассылка настроена, сохраните её. Она будет отображаться в списке рассылок соответствующего типа. В правой части каждой записи есть меню быстрых действий. С его помощью вы можете сделать **тестовую** отправку и, если всё успешно, **активировать** рассылку:

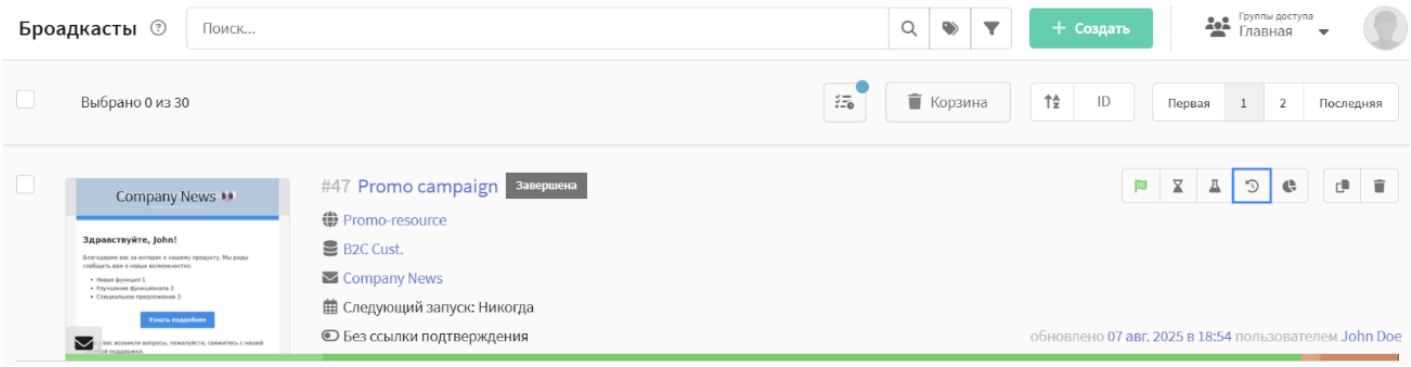


Шаг 6: Оцените эффективность рассылки

Журнал рассылки

Журнал рассылки — это инструмент, предоставляющий краткую сводку о ходе выполнения рассылки. В нем фиксируются ключевые события, параметры рассылки и статистика, что позволяет выявить возможные проблемы.

Чтобы открыть журнал рассылки, нажмите на кнопку **“Журнал”** в меню быстрых действий:



Журнал рассылки содержит следующую информацию:

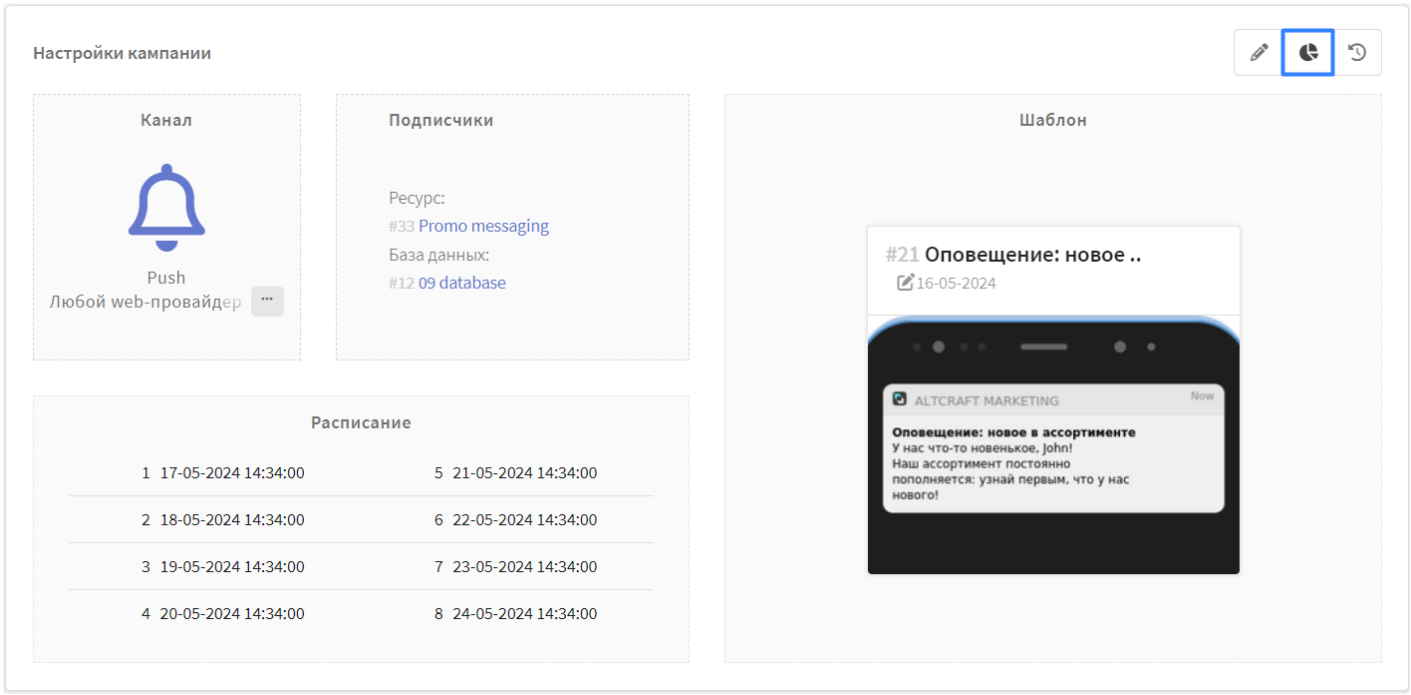
Тип	Описание
Запуск рассылки	Время начала выполнения Push-рассылки
Лимит	Показывает, установлен ли лимит на количество Push-уведомлений
Источник подписчиков	Источник контактов: **List** — База данных **Segment** — Сегмент
ID сегмента / ID базы данных	Идентификатор сегмента или базы данных
Время получения данных	Время загрузки данных из сегмента или базы данных(мс)
Количество профилей из источника	Общее число профилей в базе данных или сегменте
Количество найденных подписчиков	Число пользователей в выбранной базе данных или сегменте с активной подпиской на выбранный ресурс
Количество подписчиков без push-подписок	Профили без разрешения на Push-уведомления
Количество отправленных Push	Фактическое число успешных отправок
Профилей игнорировано настройками ресурса	Пользователи, исключённые по правилам рассылки
Рассылка завершена	Время окончания рассылки
Всего времени затрачено	Общая продолжительность рассылки

Отчеты и Аналитика

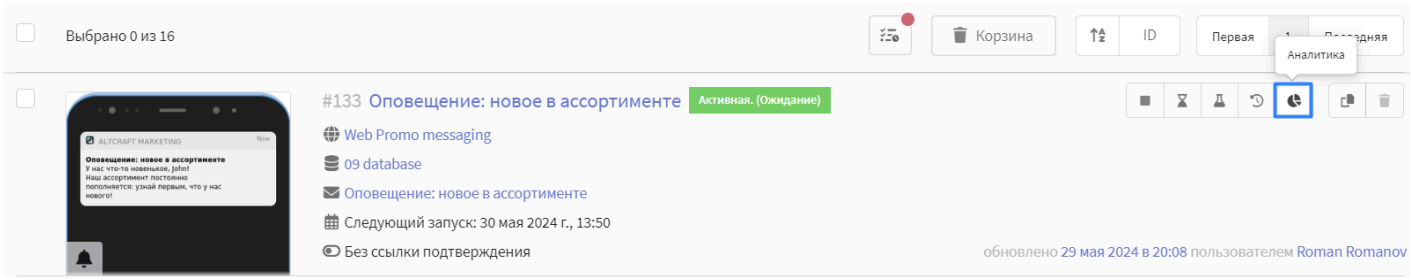
Для оценки результатов ваших рассылок воспользуйтесь отчетом по каналам.**Отчёт по каналам** предоставляет аналитику по одной или нескольким рассылкам, по одному или нескольким каналам связи.

На платформе есть три способа получения аналитики по текущей рассылке:

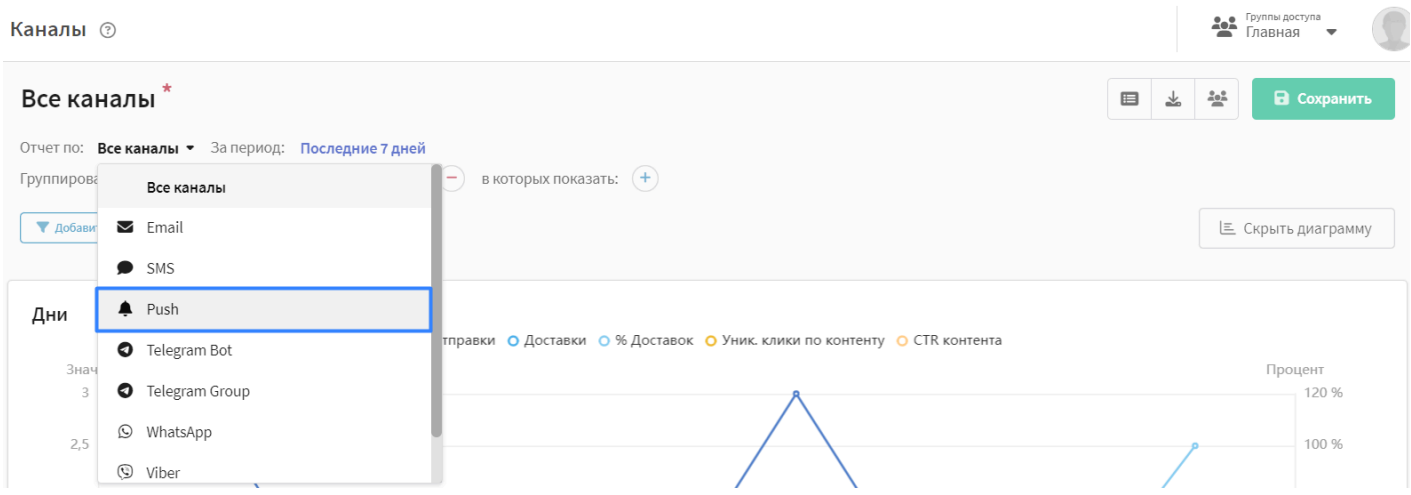
1. К отчёту по каналам конкретной рассылки можно перейти по ссылке в редакторе **рассылки**:



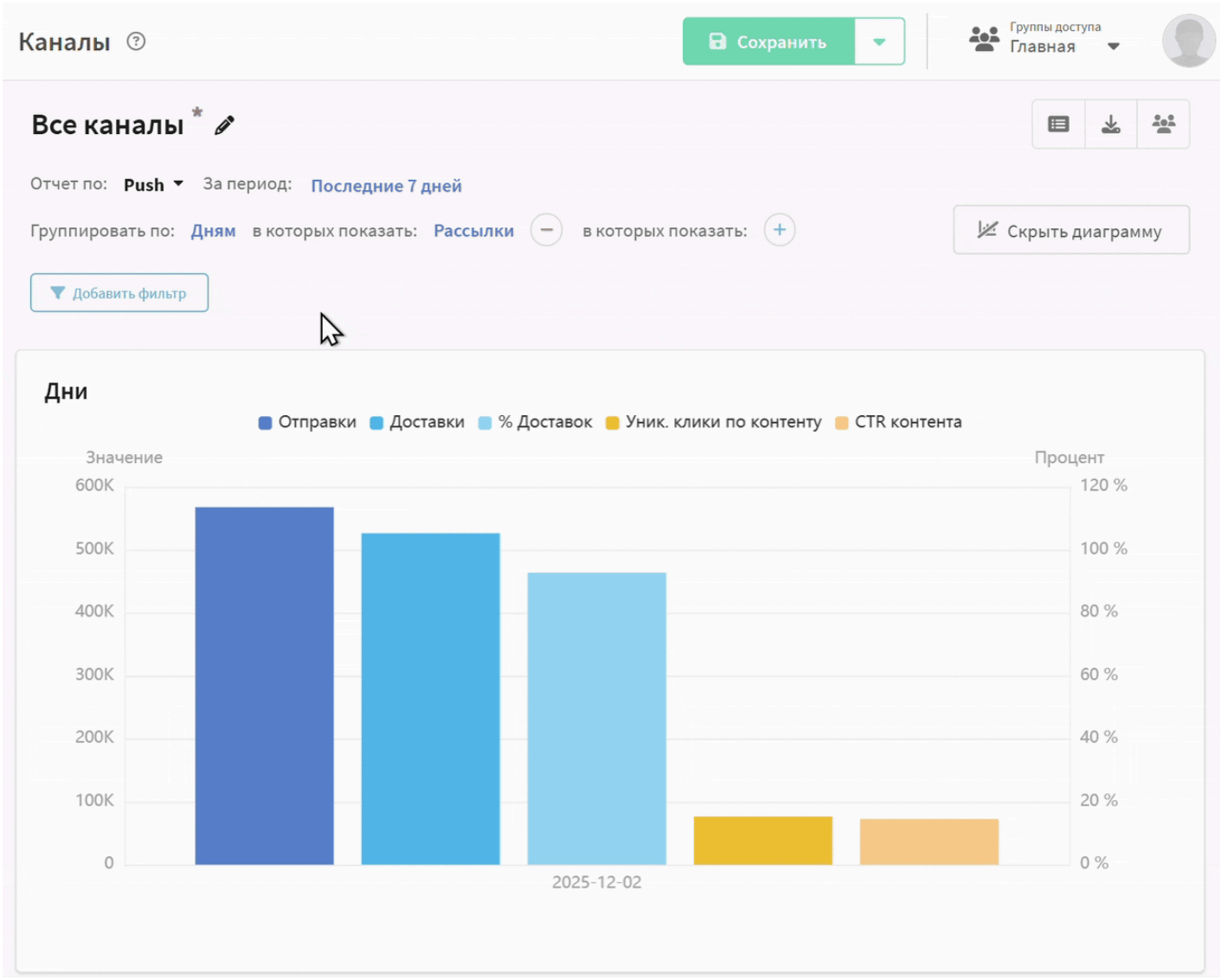
2. Вызов аналитики по текущей рассылке также возможен через меню быстрых действий:



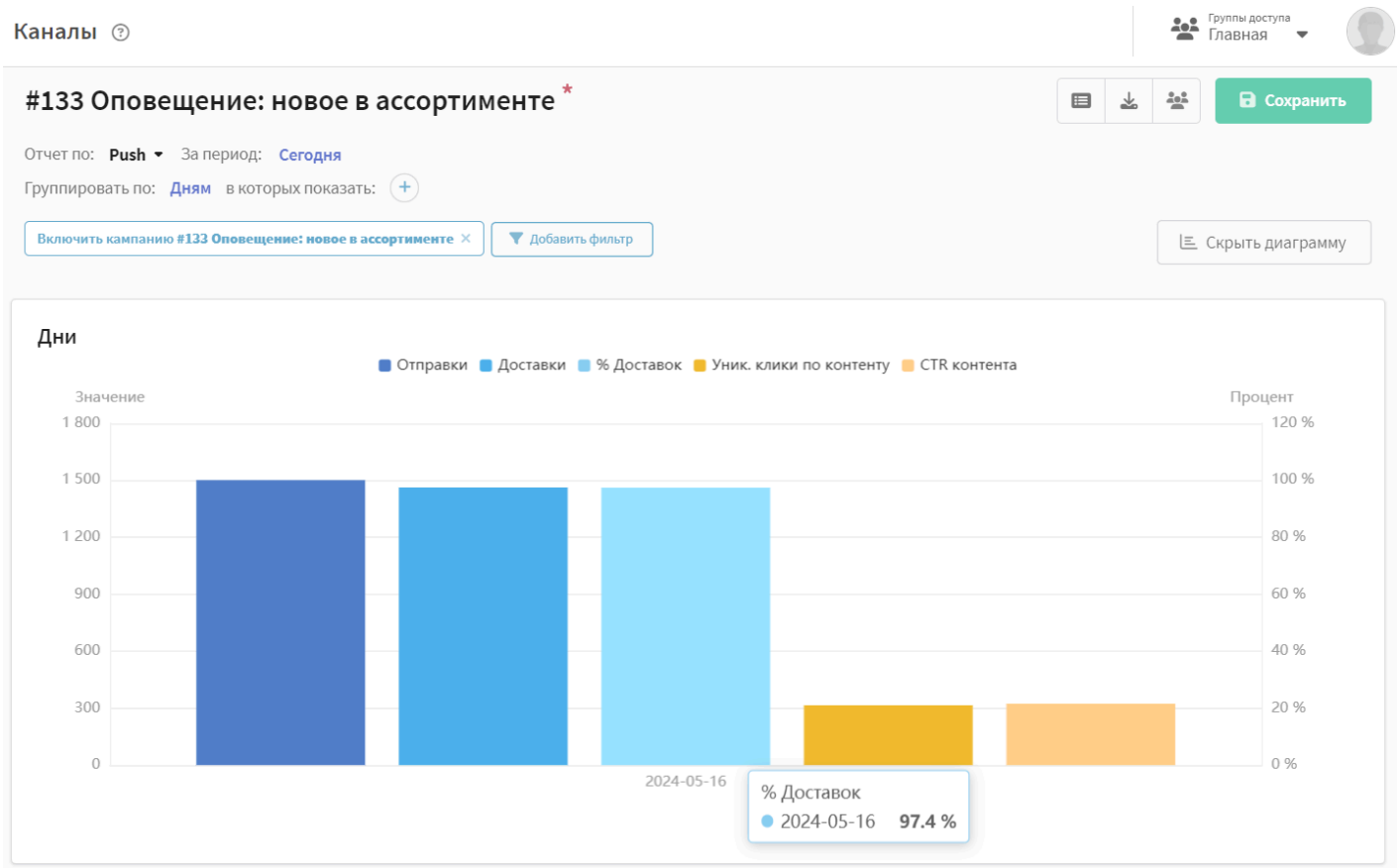
3. Выберите раздел **“Аналитика”** по **“Каналам”** в главном меню платформы. По умолчанию отобразится отчет по **“Всем каналам”** за последние 7 дней. Для группировки по **push-каналу** выберите **“Отчет по”:**



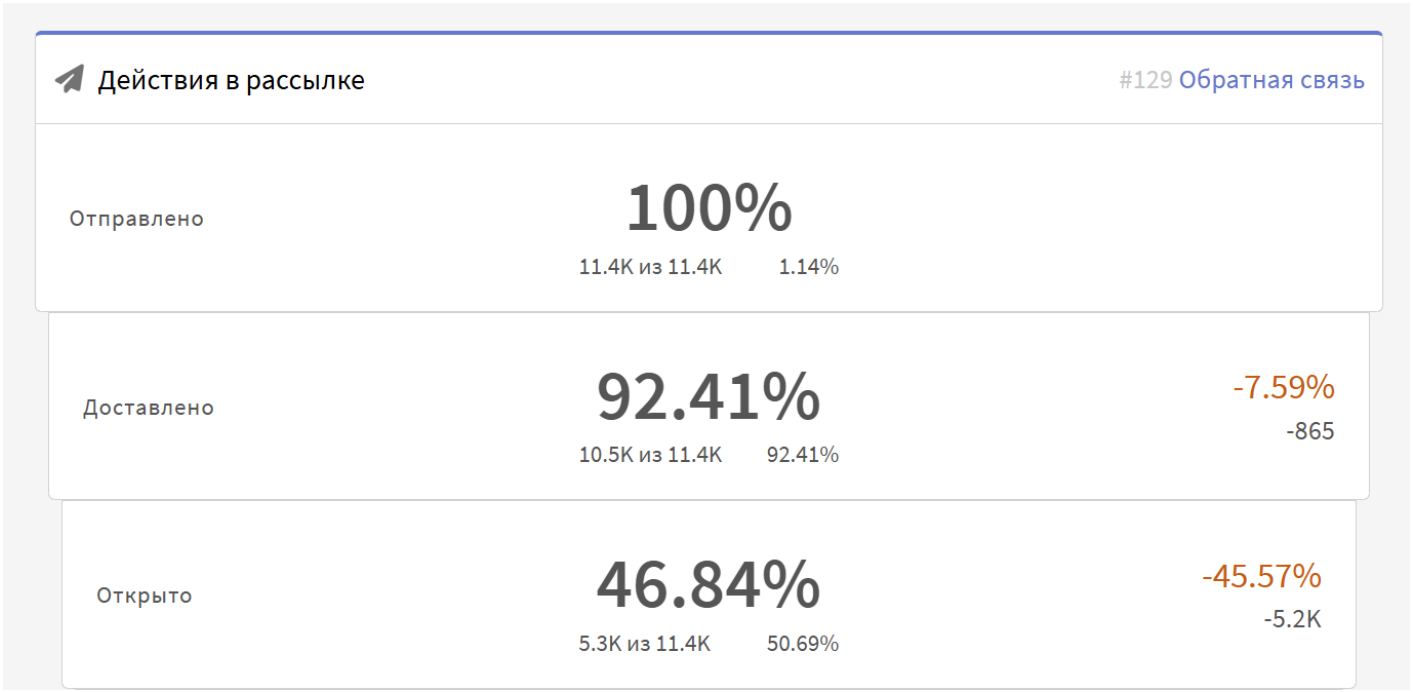
Следующим шагом **“Добавьте фильтр”** по **“Рассылкам”**. В отобразившемся списке выберите нужную рассылку.



В результате вы получите статистику по выбранной рассылке.



Действия в push-рассылке или шаблоне могут стать шагами в **интерактивной воронке конверсии**:



Mobile Push

Первая Mobile push-рассылка

Мобильные push-уведомления отображаются по мере получения в центре уведомлений мобильного устройства — в “шторке” Android и iOS смартфонов, планшетов и других гаджетов. По умолчанию уведомления включены для устанавливаемых приложений, так что вы можете сразу с момента установки начать процесс вовлечения клиента.

Это руководство поможет вам организовать рассылку мобильных push-уведомлений в платформе Altcraft.

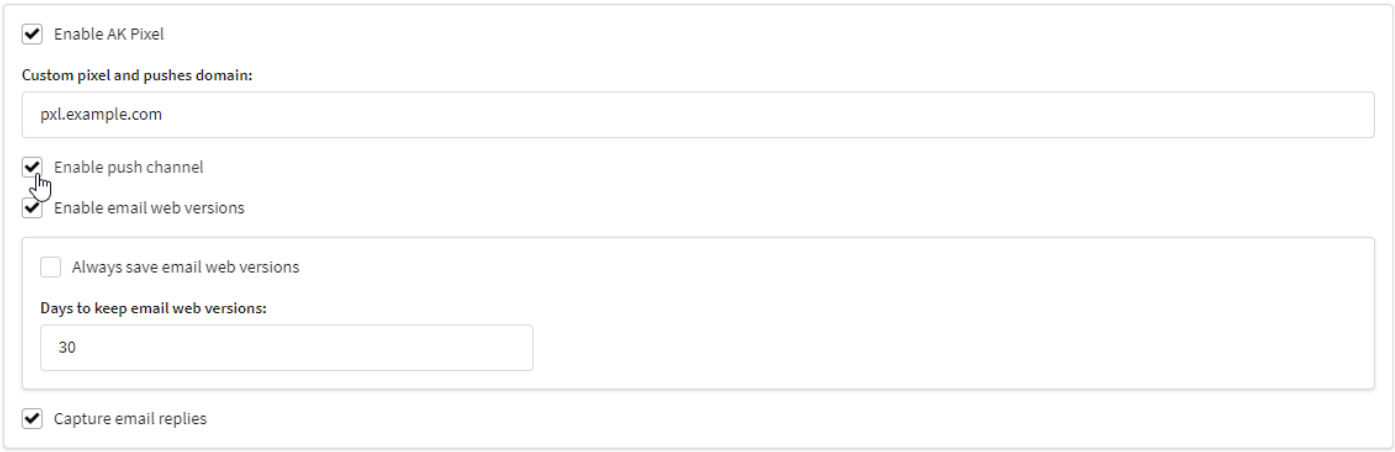
> **TIP**
>
> Push-уведомления гораздо выгоднее SMS, так как их отправка фактически ничего не стоит.

Шаг 1: Настройте домен для работы с Push

> **CAUTION**
>
> Если вы пользуетесь **облачным решением**, пропустите этот шаг и перейдите к настройкам подписки.
>
> При установке Altcraft Platform на ваше оборудование **On-Premise** передайте инструкции из этого шага **администраторам** вашей платформы. Или свяжитесь с нашей службой поддержки!

Убедитесь, что **и ваш веб-сайт, и платформа** работают **по протоколу HTTPS**. В платформе Altcraft должны быть **настроены web-пиксели**.

Необходимо активировать **push-уведомления** и **pixel** в Панели администратора для **аккаунтов**, с которых планируется отправка:



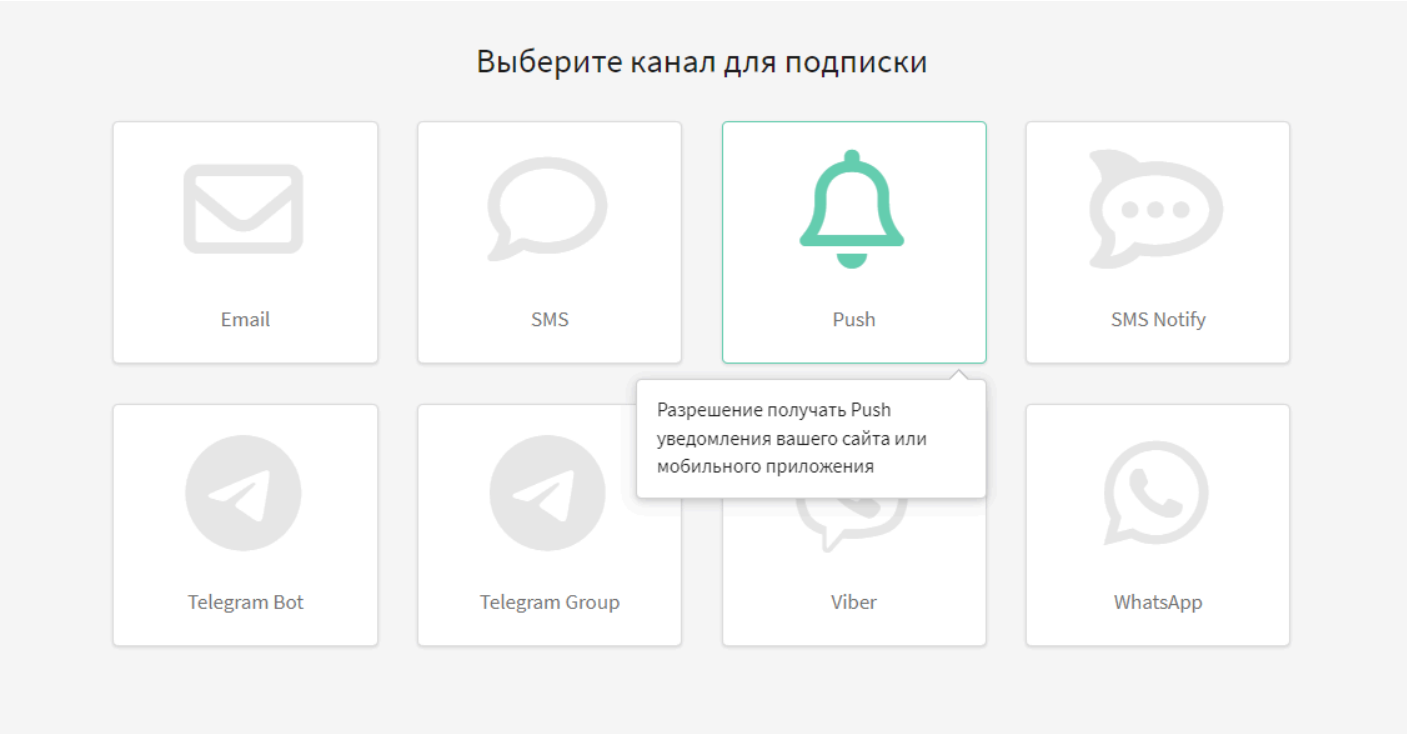
Шаг 2: Подпишите клиентов на Push канал

Базы профилей и ресурсы подписок

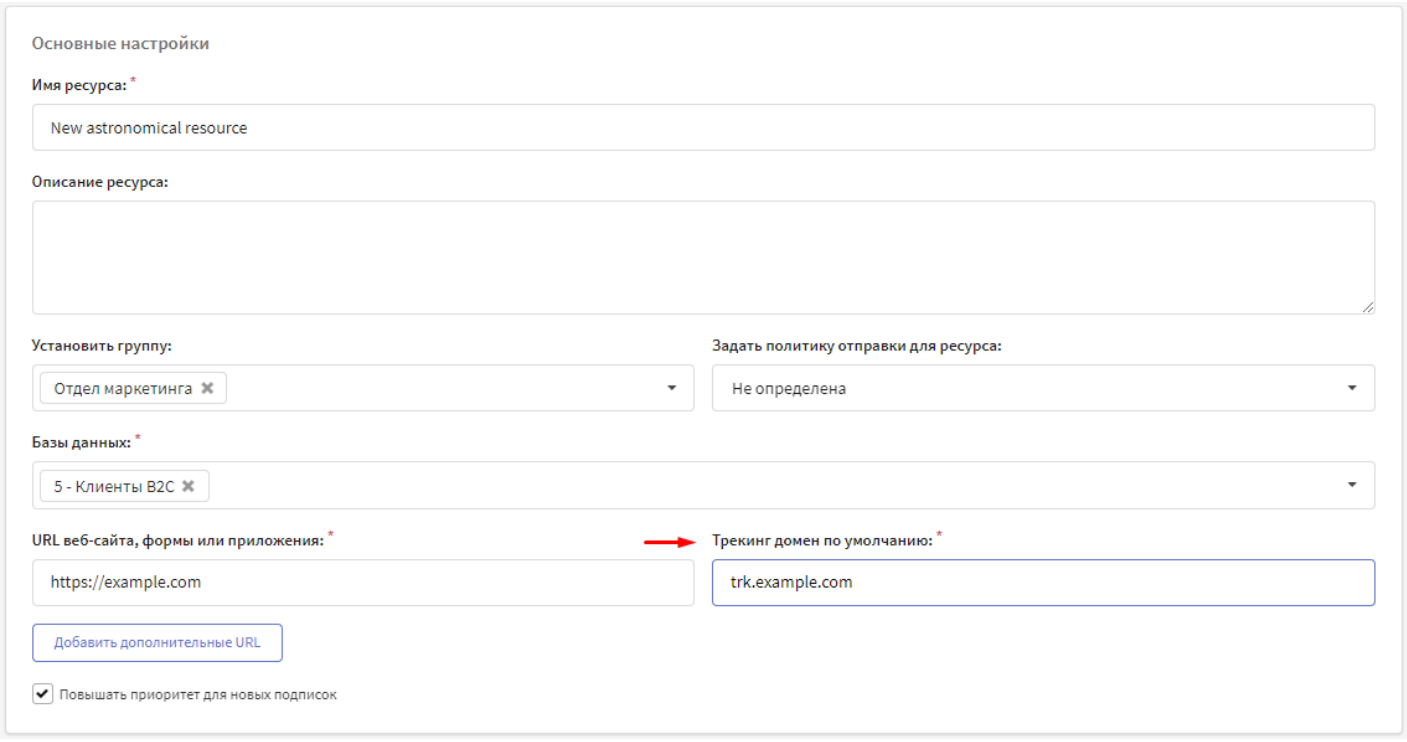
Рассылки Altcraft Platform получают ваши подписчики. **Профили** подписчиков хранятся в **базах данных профилей**.

Рассылки при отправке Push используют **подписки профиля** на **Ресурсы**: так вы можете отправлять уведомления на различные устройства клиента или в разные браузеры. Ресурсом может быть отдельное направление маркетинга, сайт, приложение или тип контента.

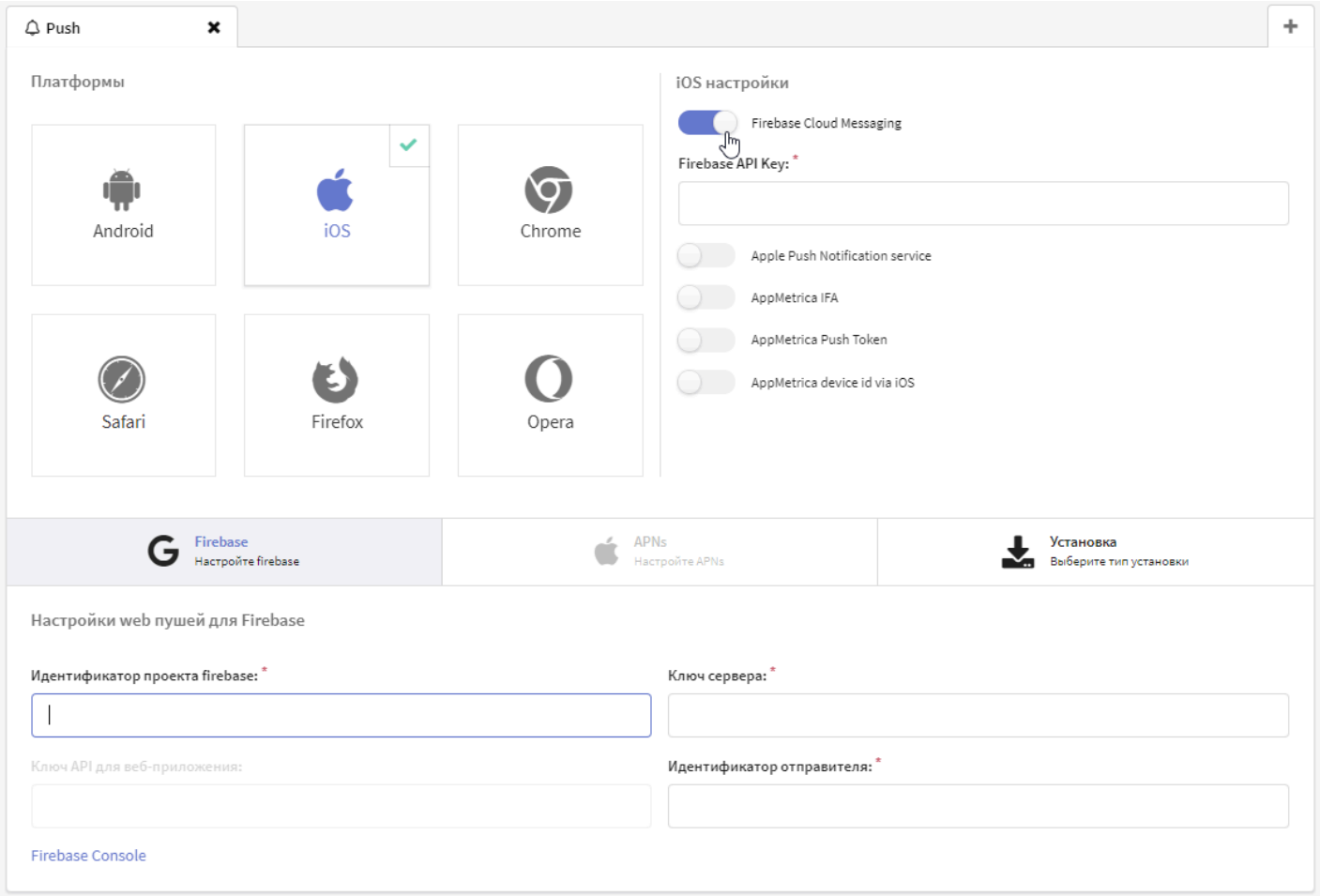
Выберите в главном меню раздел “Ресурсы”, создайте новый ресурс или выберите уже существующий. После этого добавьте **Push-канал** в существующий ресурс или выберите его в качестве основного канала для нового:



В основных настройках добавьте необходимые базы данных, при необходимости назначьте политику отправки. Укажите **Трекинг домен по умолчанию**: он будет обрабатывать переходы ваших клиентов по ссылкам в сообщениях:



В настройках Push-канала ресурса выберите **платформы**, на которые собираетесь отправлять уведомления клиентам, и внесите соответствующие настройки:



Для мобильных приложений есть 4 варианта настройки:

- **Google Firebase Cloud Messaging** — для Android и iOS приложений.
- **Apple Push Notification Service** — только для iOS приложений.
- **Yandex.AppMetrica** — для Android и iOS приложений.
- **Huawei Mobile Services** — для Android и iOS приложений.
- **RuStore** — только для Android приложений.

Процесс настройки для различных платформ описан в **этом руководстве**.

Профили клиентов и подписка на ресурс

Для каждого клиента в Altcraft Platform создаётся единый цифровой профиль, объединяющий личные данные, историю действий и подписки на ваши ресурсы. Для отправки уведомления конкретному получателю используется токен (идентификатор) устройства. В случае с mobile push этот идентификатор обычно связан с конкретной установкой приложения.

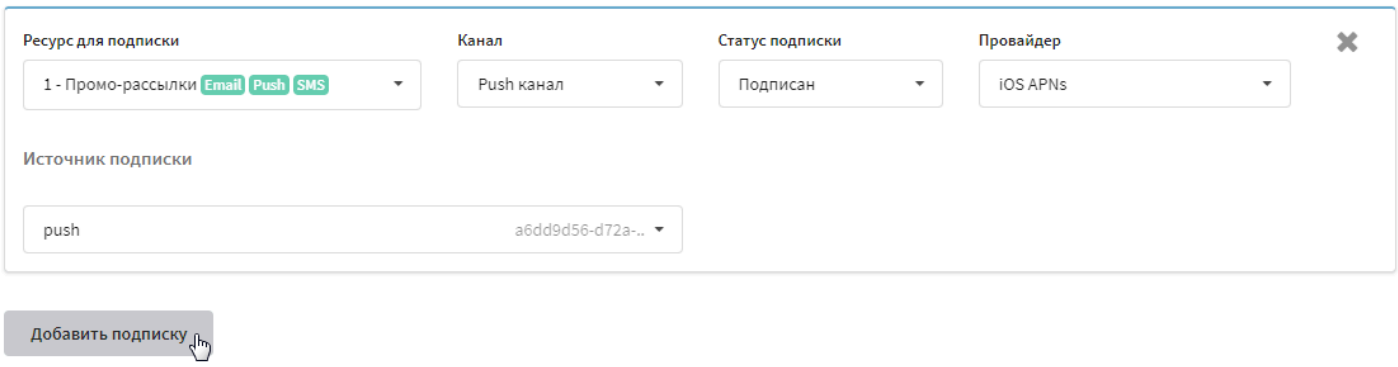
Существующая база подписчиков

Если у вас есть база токенов пользователей приложения, вы можете загрузить данные в платформу и использовать для отправки рассылок: по одному клиенту или из табличного файла. Различные способы импорта данных описаны в этой статье.

Также вы можете настроить автоматический импорт профилей клиентов:

- по расписанию из вашей SQL базы данных
- посредством интеграции с API платформы

При импорте данных подпишите клиентов на подготовленный ранее ресурс. Для этого в разделе **“Настройки подписки”** выберите Push в качестве канала связи, выберите провайдера уведомлений, в поле **“Источник подписки”** укажите колонку с токенами устройств:



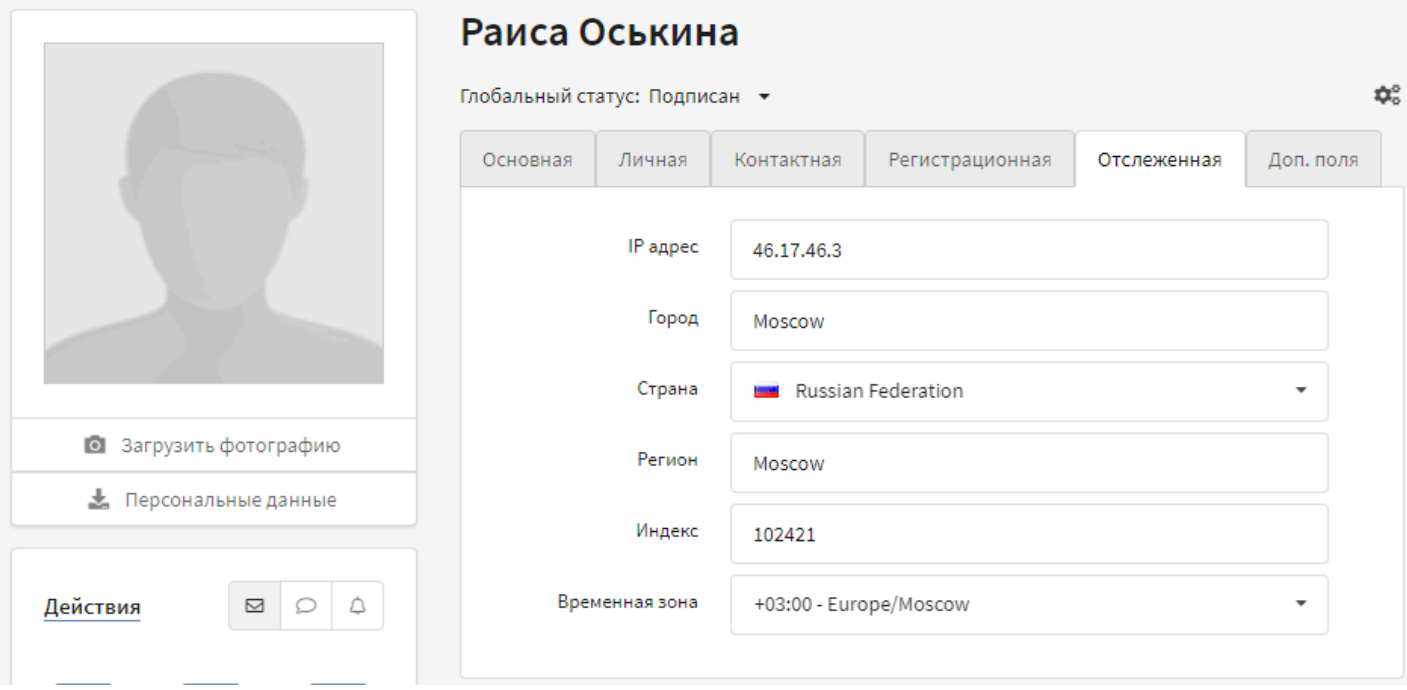
TIP

Если у вас уже есть база профилей, добавьте им подписку на новый ресурс. Есть два варианта:

- если у клиентов есть подписки на другой ресурс с актуальными токенами устройств — используйте массовое обновление профилей
- если для подписки на Push-идентификаторы необходимо загрузить в платформу — используйте импорт профилей в режиме" только обновление"

Новые push подписчики приложения

Altcraft Platform импортирует профили пользователей вашего приложения, когда те устанавливают приложение на мобильное устройство: по умолчанию показ обновлений включен как в Android, так и в iOS. При этом в профиль записывается информация об устройстве подписчика, IP адрес и определённая по нему геолокация:

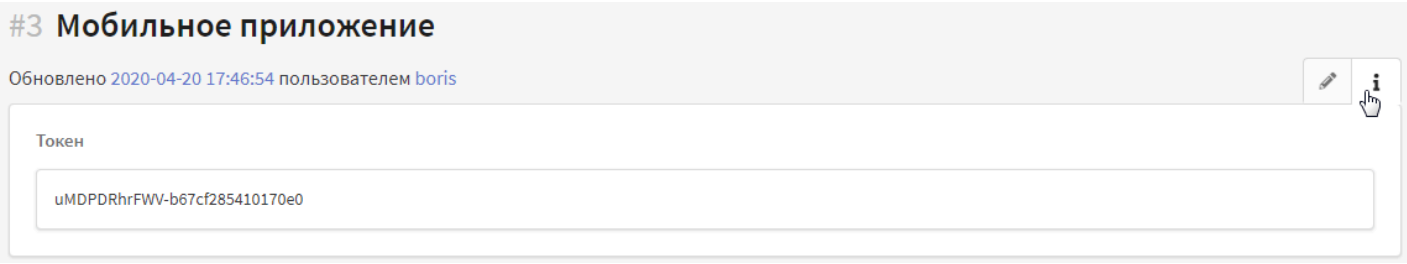


Для настройки импорта новых push-подписчиков перейдите к следующему шагу.

Шаг 3: Добавьте push SDK в своё приложение

Для проведения push-рассылок в ваше приложение должен быть установлен инструментарий для работы с push, а именно SDK той платформы отправки, которую вы планируете использовать.

Свяжитесь с **разработчиками** вашего приложения для согласования работы и передайте им токен настроенного ранее **ресурса** (в настройках ресурса, см. снимок ниже).



Подписки на уведомления из приложений присваиваются пользователям API-запросом из приложения: Добавить подписку на push в базу данных.

У Altcraft Platfrom есть собственный SDK, **Altcraft mSDK**. Он реализован на **Firebase** для платформы **Android** и на **Firebase** и **APNs** для платформы **iOS**. Для получения доступа к SDK вы можете связаться с нашей службой поддержки. Узнать подробнее о возможностях Altcraft mSDK можно здесь.

TIP

Подписка на **Yandex.AppMetrica** может добавляться **автоматически** пользователям приложения с подключенным сервисом метрики. Подробнее в статье Интеграция с Yandex.AppMetrica.

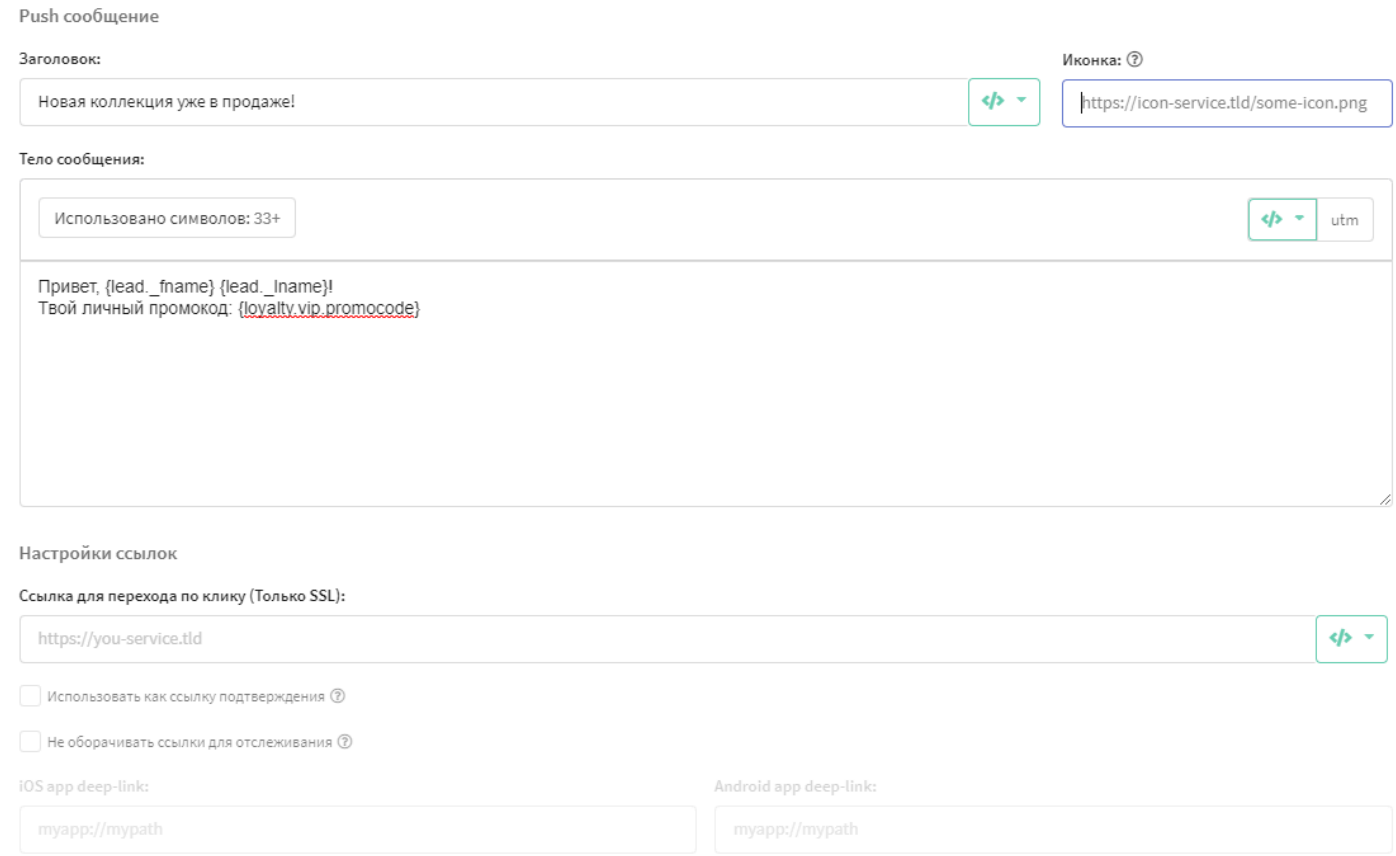
Шаг 4: Подготовьте контент для отправки

Контент для ваших Push-рассылок создаётся в **Шаблонах сообщений**. Шаблоны содержат контент для всех доступных каналов. При создании нового шаблона выберите Push-канал, или добавьте его для существующего.

Push-версия шаблона содержит:

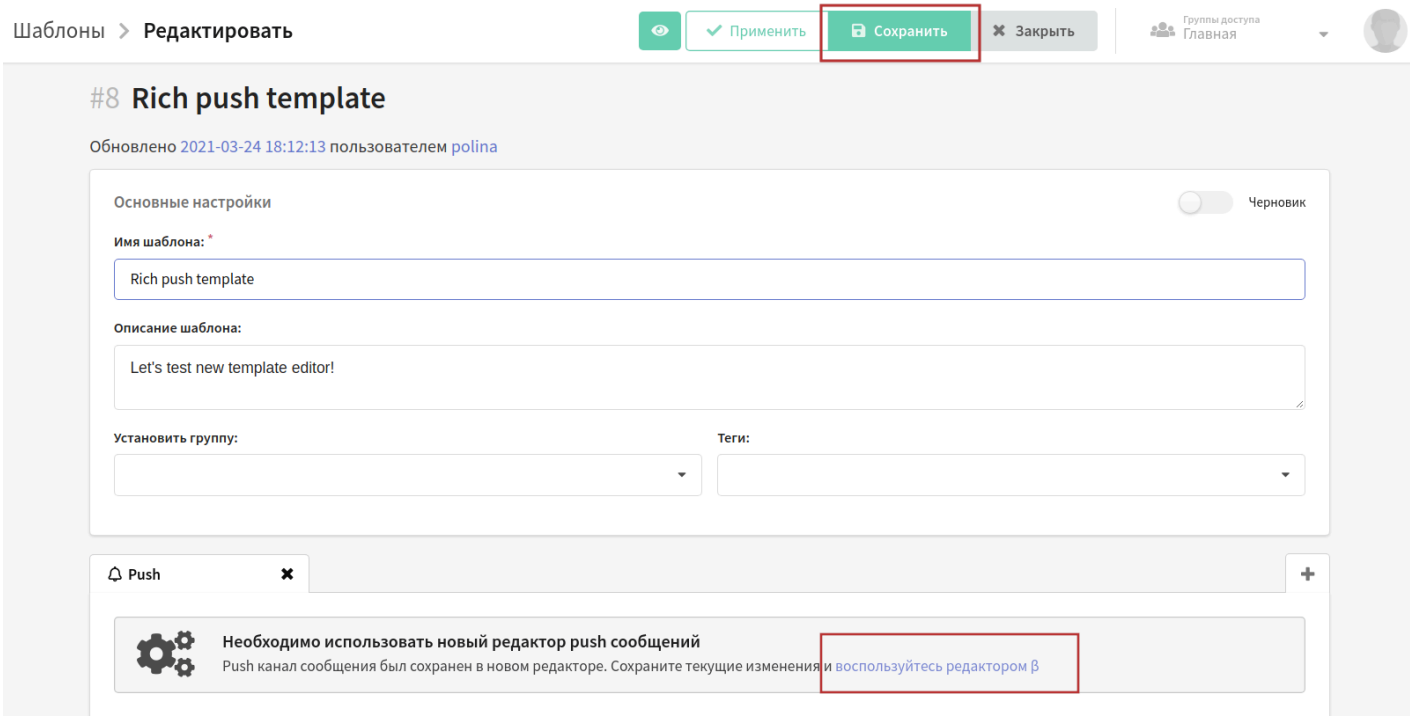
- **Заголовок** — аналогичен теме Email письма.
- **Иконка** — любое изображение для отображения в уведомлении.
- **Тело сообщения** — основная текстовая часть сообщения.
- **Ссылка для перехода** — ссылка, по которой перейдёт подписчик при нажатии на push. Отдельно настраиваются deep link для каждой платформы.

В заголовок, ссылку и тело сообщения вы можете добавить данные конкретного клиента, а также динамический текстовый контент из внешних источников:



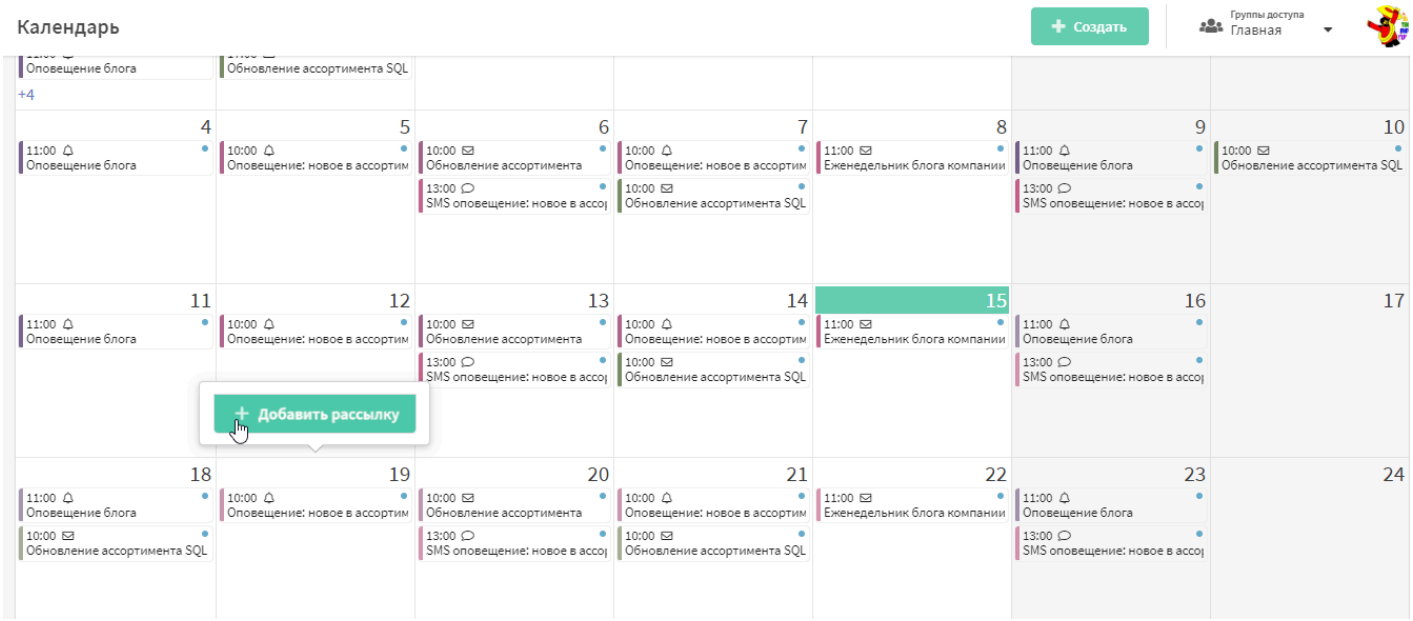
Если вы рассылаете Rich push уведомления, то для создания шаблона вам стоит воспользоваться специальным **редактором**. В Rich push уведомления можно добавлять не только текст и иконку, но и кнопки, изображения, банеры и даже аудио и видеофайлы. Более того, новый редактор позволяет отдельно настраивать уведомление для определенной ОС, а также добавлять адаптивные ссылки и deeplink.

Чтобы открыть новый редактор, зайдите в обычный редактор шаблонов и сделайте начальную версию push-сообщения. Как это сделать, мы описали выше. После этого нажмите кнопку **Сохранить**. Теперь вернитесь в созданный шаблон и кликните **“воспользуйтесь редактором β”**, чтобы перейти в новый редактор.

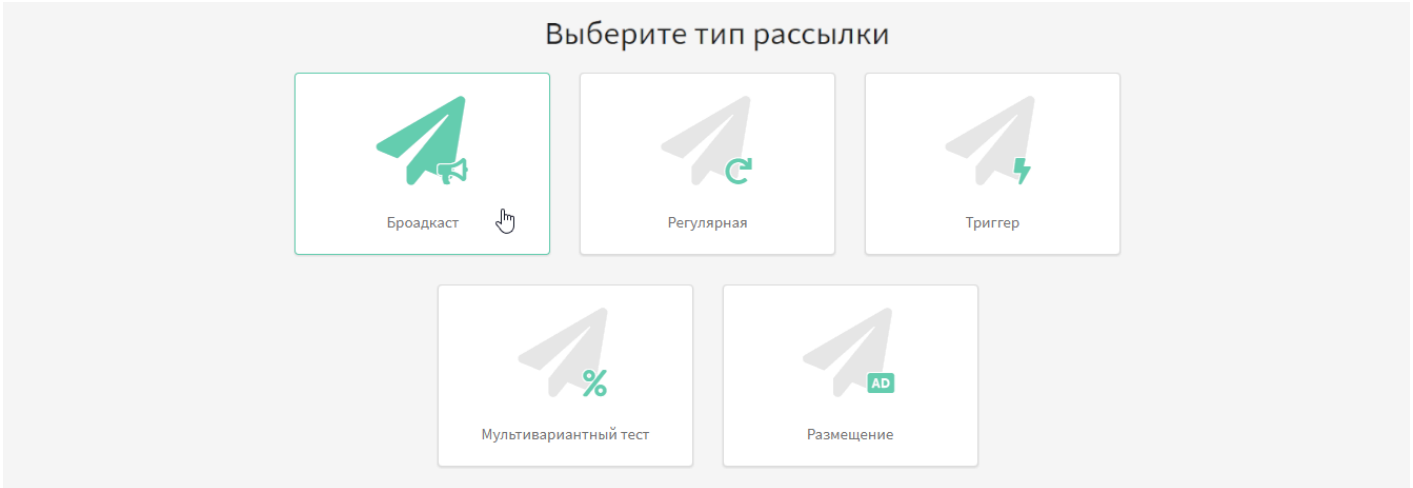


Шаг 5: Выберите тип рассылки

	Для отправки Push-сообщений используется инструмент **Рассылка**. Если вам нужно произвести **разовую** рассылку по целой базе профилей или определённому сегменту аудитории — подойдёт броадкаст. Если же рассылка будет **периодической** — ваш выбор это регулярная рассылка. В главном меню выберите раздел **Рассылки** и откройте **Календарь**. В нём отображаются все рассылки, отправляемые по расписанию.



Определитесь с датой и создайте рассылку нужного типа:



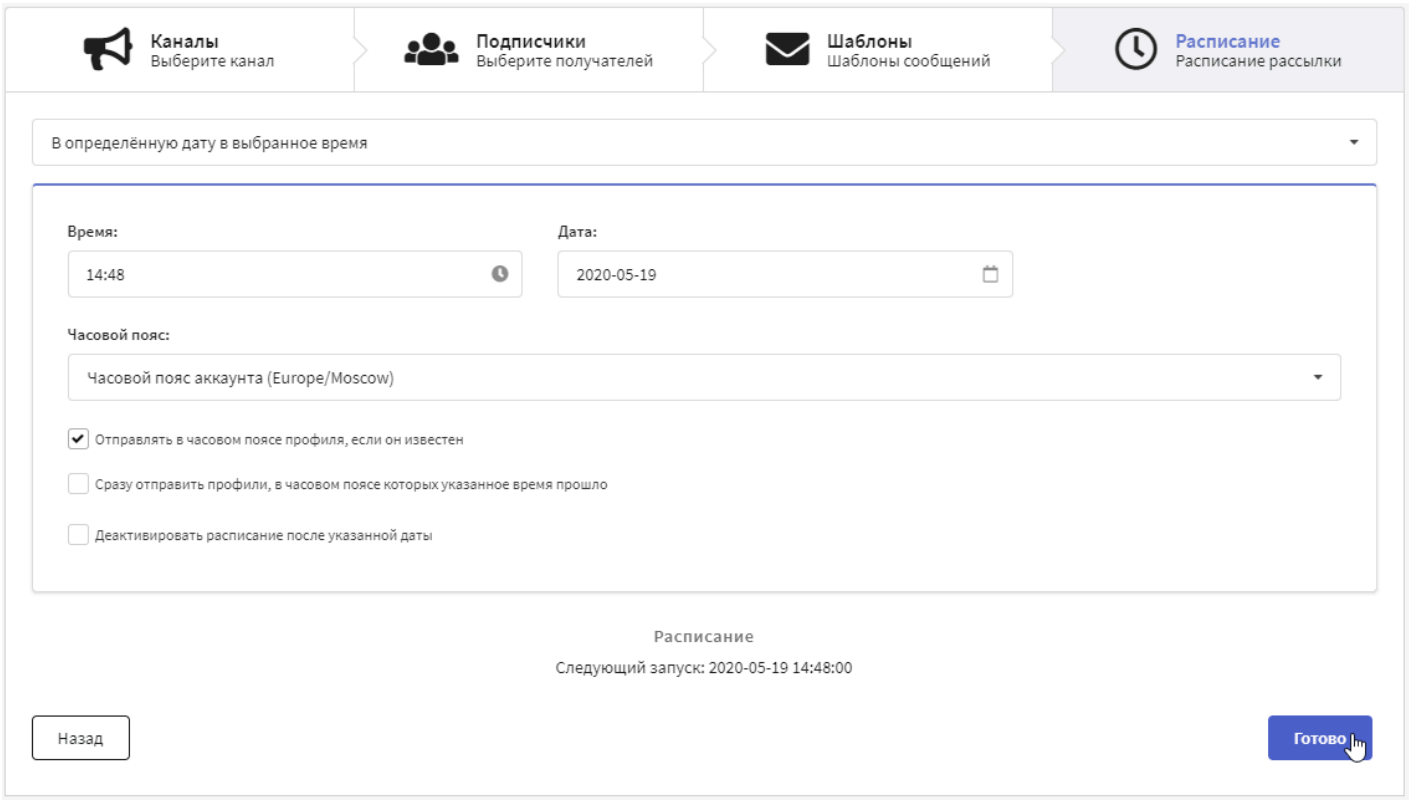
TIP

Подробную инструкцию по настройке **Броадкаста** вы найдёте на этой странице.

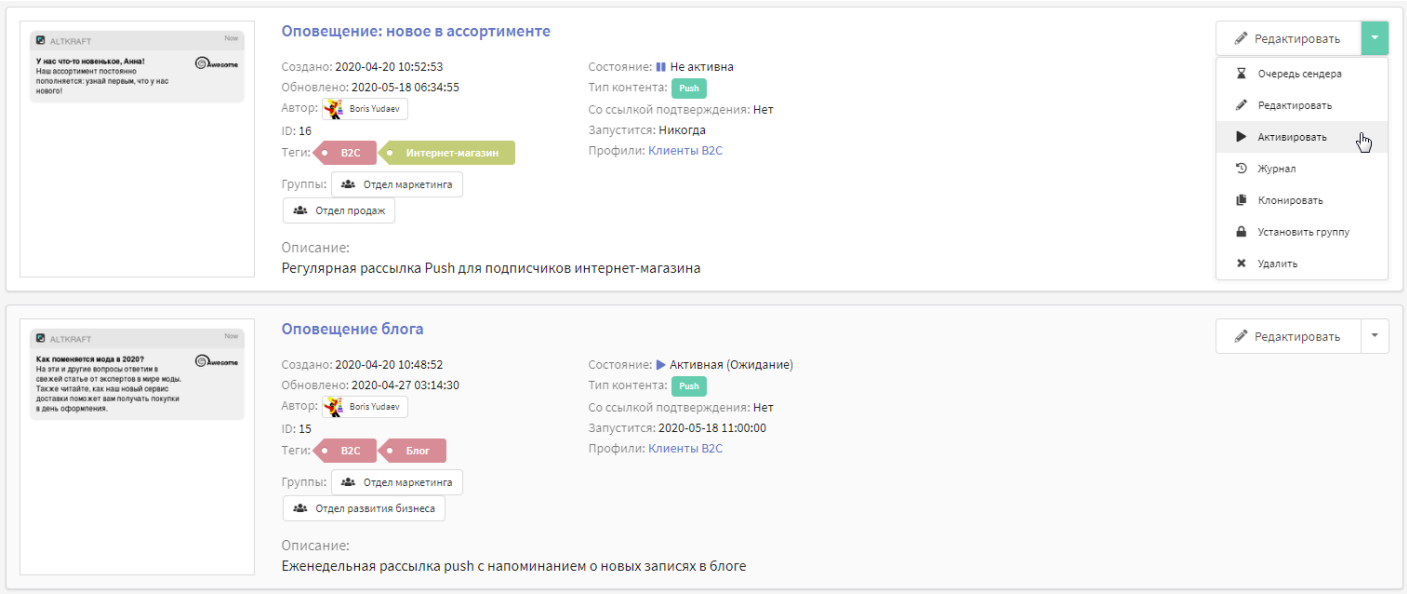
TIP

Подробную инструкцию по настройке **Регулярной рассылки** вы найдёте на этой странице.

1. Выбрать **Push** как канал связи
2. Выберите провайдеров для рассылки и режим отправки. Есть два вида отправки — **fanout**, рассылка на все подписки с учетом фильтров, или **last**, отправка на подписку с самой свежей датой создания. Если несколько подписок были созданы одновременно, то из них будет выбрана одна случайная. Если в рассылке заданы фильтры по провайдеру, дополнительным полям или категориям, то сначала накладываются фильтры, а потом уже ищется самая свежая подписка.
3. Выбрать настроенный ранее **ресурс** подписок и **аудиторию** получателей
4. Добавить в рассылку **шаблон** сообщения
5. Установить **расписание**. Например, для броадкаста подойдёт вариант “В определённую дату в выбранное время”:



Когда рассылка настроена, сохраните её. Она будет отображаться в списке рассылок соответствующего типа. В правой части каждой записи есть меню быстрых действий. С его помощью вы можете сделать **тестовую** отправку — и если всё успешно, **активировать** рассылку:

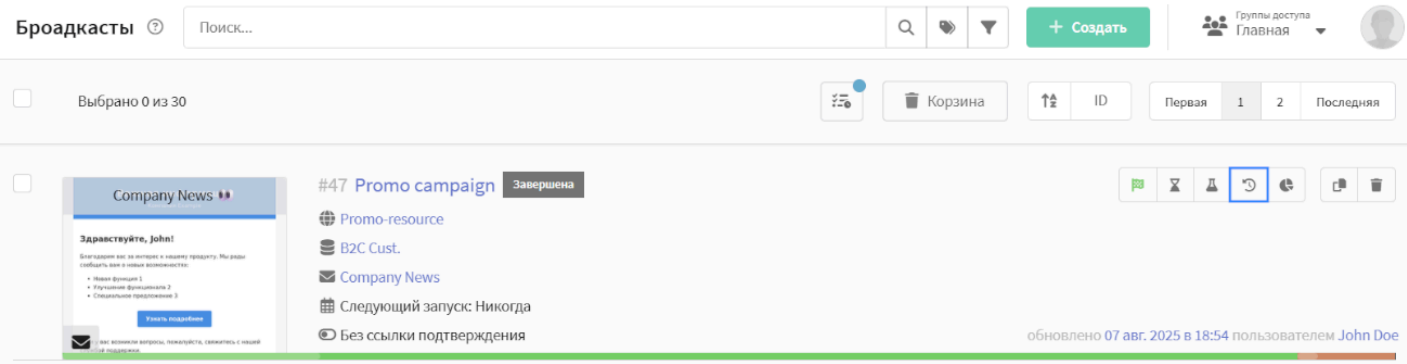


Шаг 6: Оцените эффективность рассылки

Журнал рассылок

Журнал рассылок — это инструмент, предоставляющий краткую сводку о ходе выполнения рассылки. В нем фиксируются ключевые события, параметры рассылок и статистика, что позволяет выявить возможные проблемы.

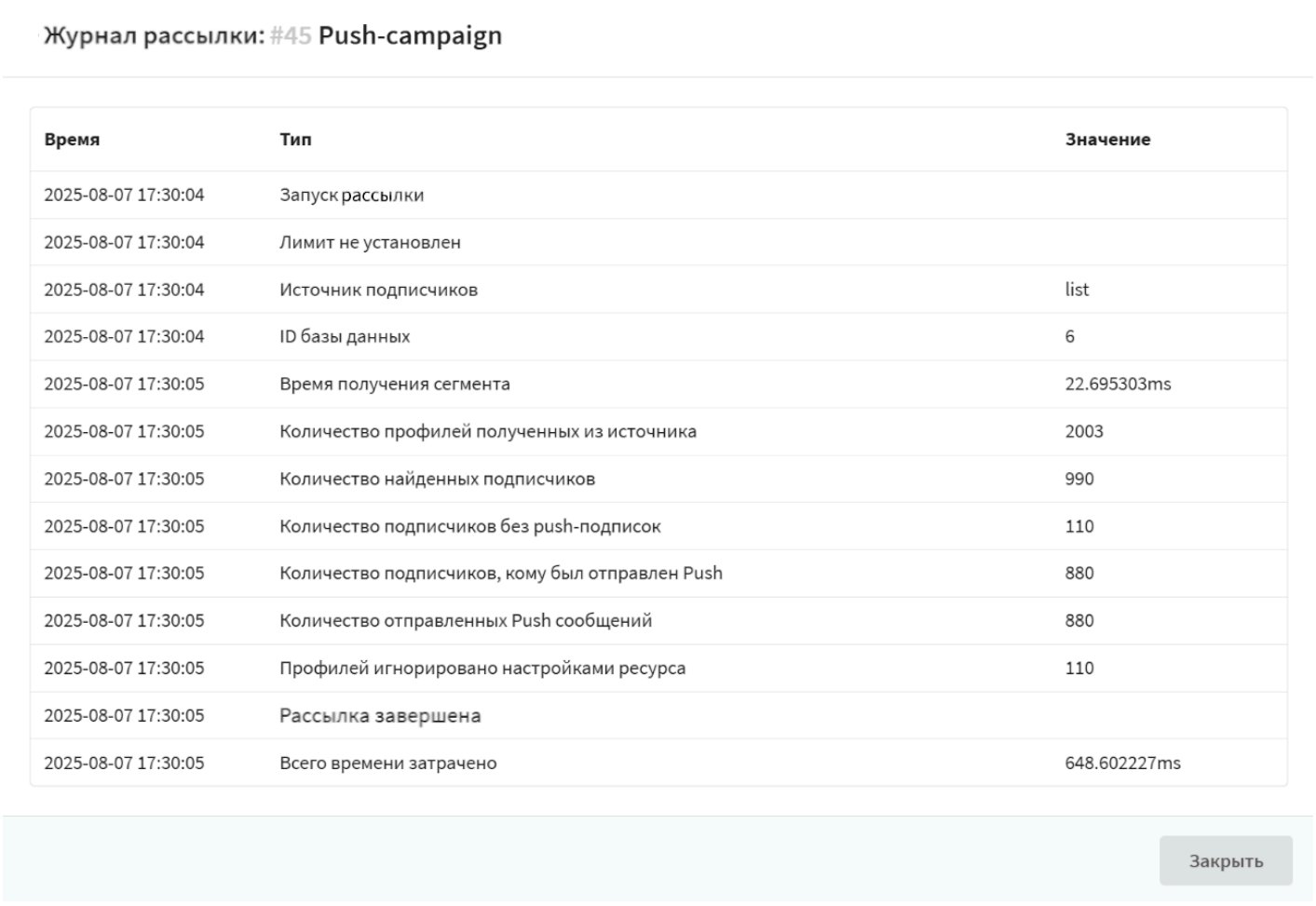
Чтобы открыть журнал рассылки, нажмите на кнопку **“Журнал”** в меню быстрых действий:



Журнал рассылки содержит следующую информацию:

Тип	Описание
Запуск рассылки	Время начала выполнения Push-рассылки
Лимит	Показывает, установлен ли лимит на количество Push-уведомлений
Источник подписчиков	Источник контактов: **List** — База данных **Segment** — Сегмент
ID сегмента / ID базы данных	Идентификатор сегмента или базы данных
Время получения данных	Время загрузки данных из сегмента или базы данных(мс)
Количество профилей из источника	Общее число профилей в базе данных или сегменте

Тип	Описание
Количество найденных подписчиков	Число пользователей в выбранной базе данных или сегменте с активной подпиской на выбранный ресурс
Количество подписчиков без push-подписок	Профили без разрешения на Push-уведомления
Количество отправленных Push	Фактическое число успешных отправок
Профилей игнорировано настройками ресурса	Пользователи, исключённые по правилам рассылки
Рассылка завершена	Время окончания рассылки
Всего времени затрачено	Общая продолжительность рассылки

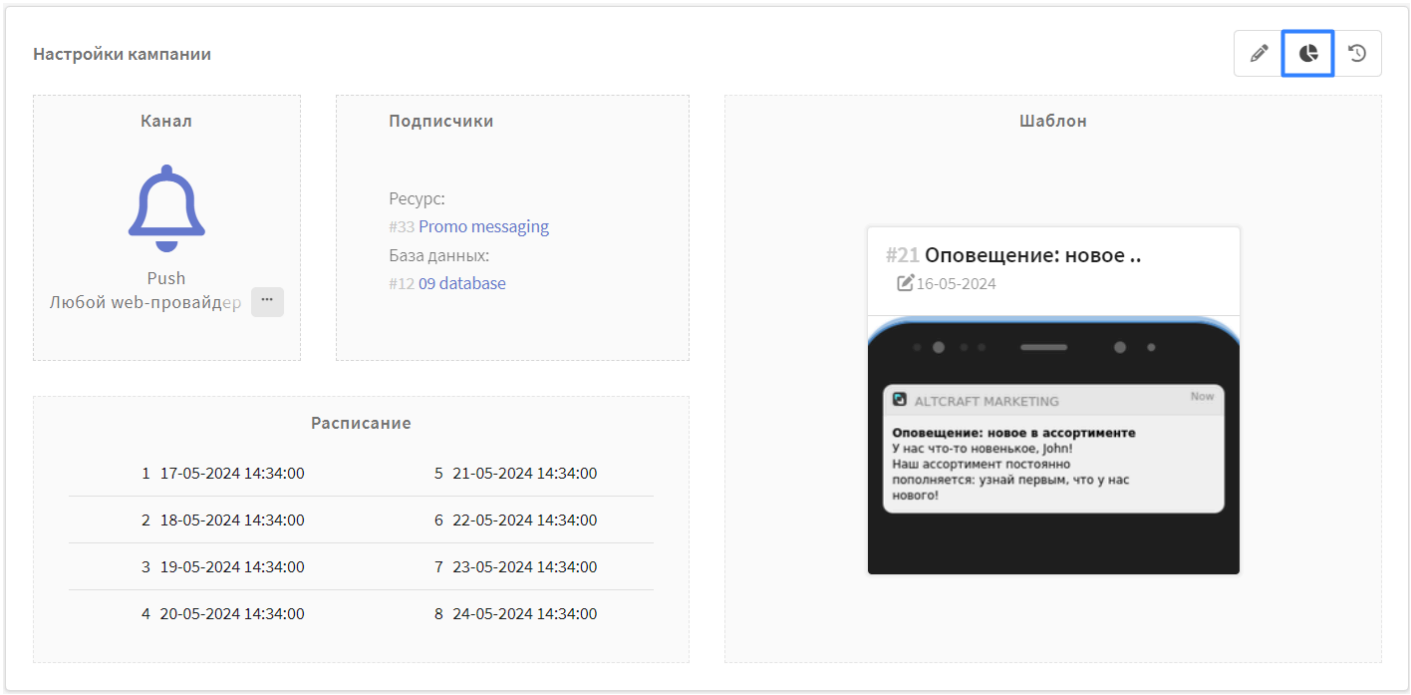


Отчеты и Аналитика

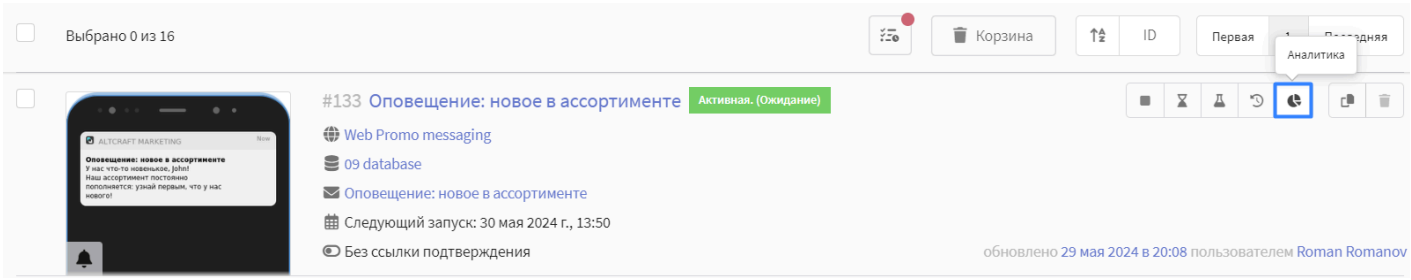
Для оценки результатов ваших рассылок воспользуйтесь отчетом по каналам.**Отчёт по каналам** предоставляет аналитику по одной или нескольким рассылкам, по одному или нескольким каналам связи.

На платформе есть три способа получения аналитики по текущей рассылке:

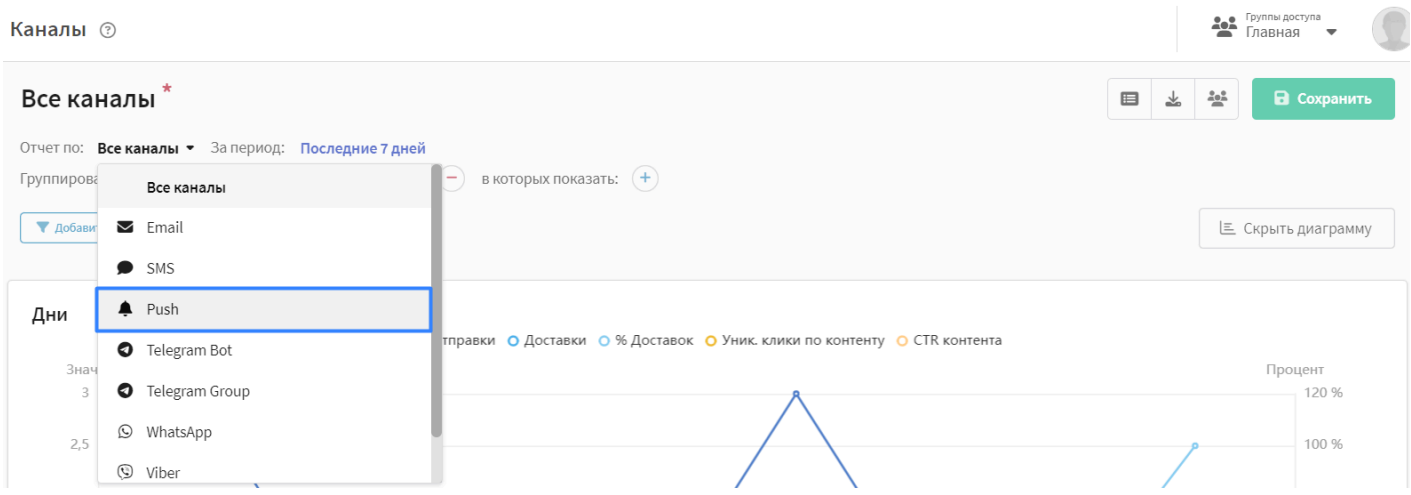
1. К отчёту по каналам конкретной рассылки можно перейти по ссылке в редакторе **рассылки**:



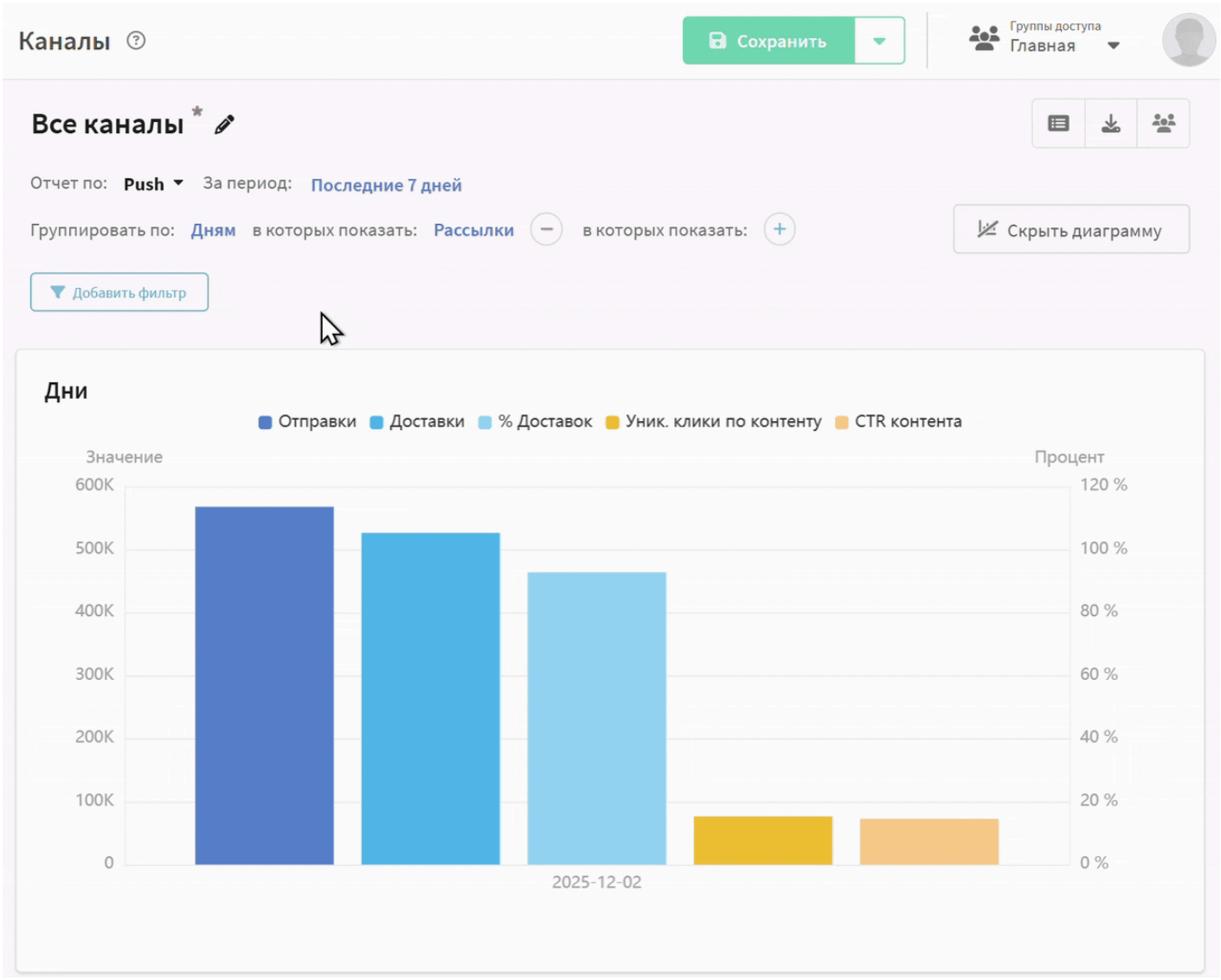
2. Вызов аналитики по текущей рассылке также возможен через меню быстрых действий:



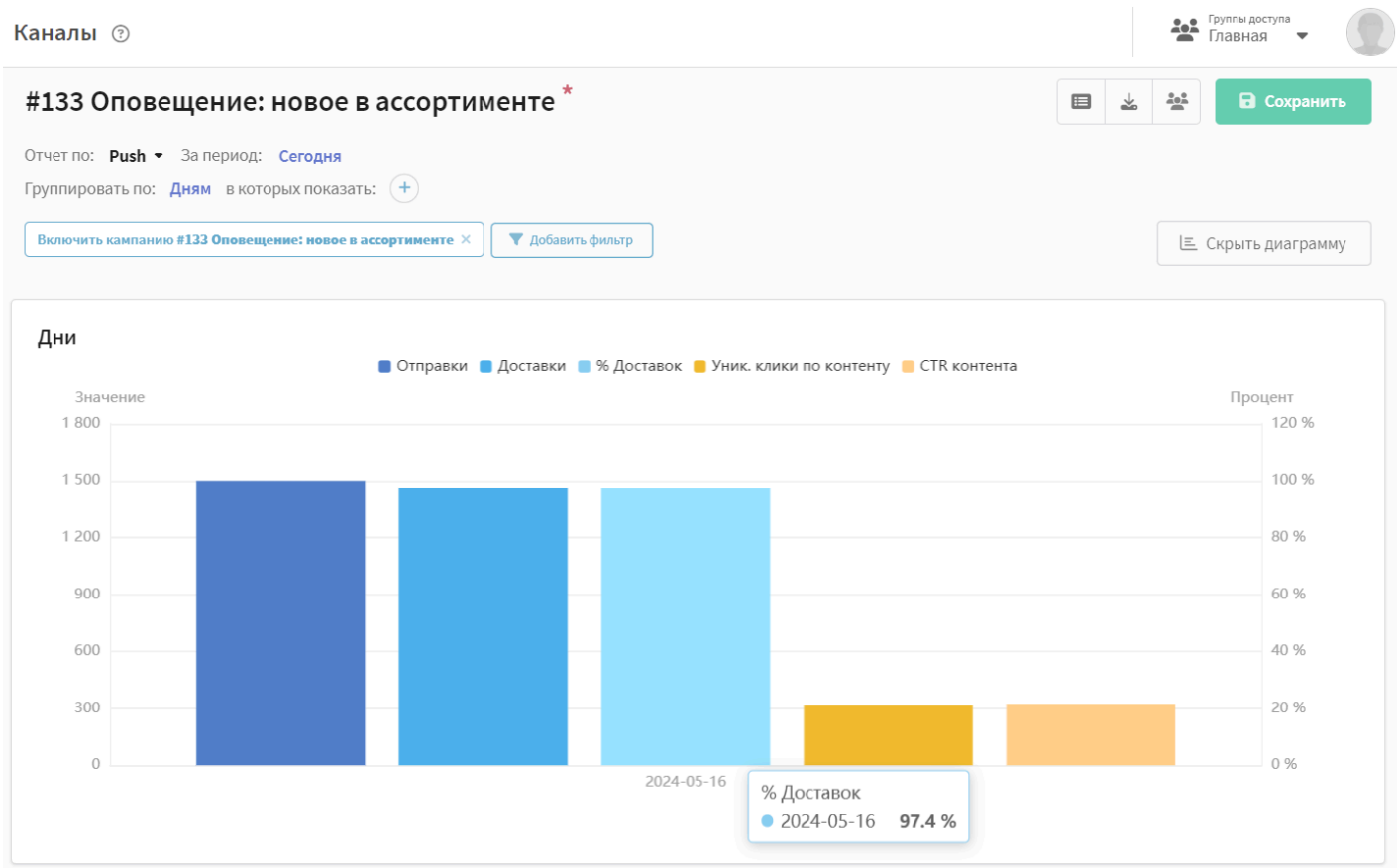
3. Выберите раздел **“Аналитика”** по **“Каналам”** в главном меню платформы. По умолчанию отобразится отчет по **“Всем каналам”** за последние 7 дней. Для группировки по **Push-каналу** выберите **“Отчет по”:**



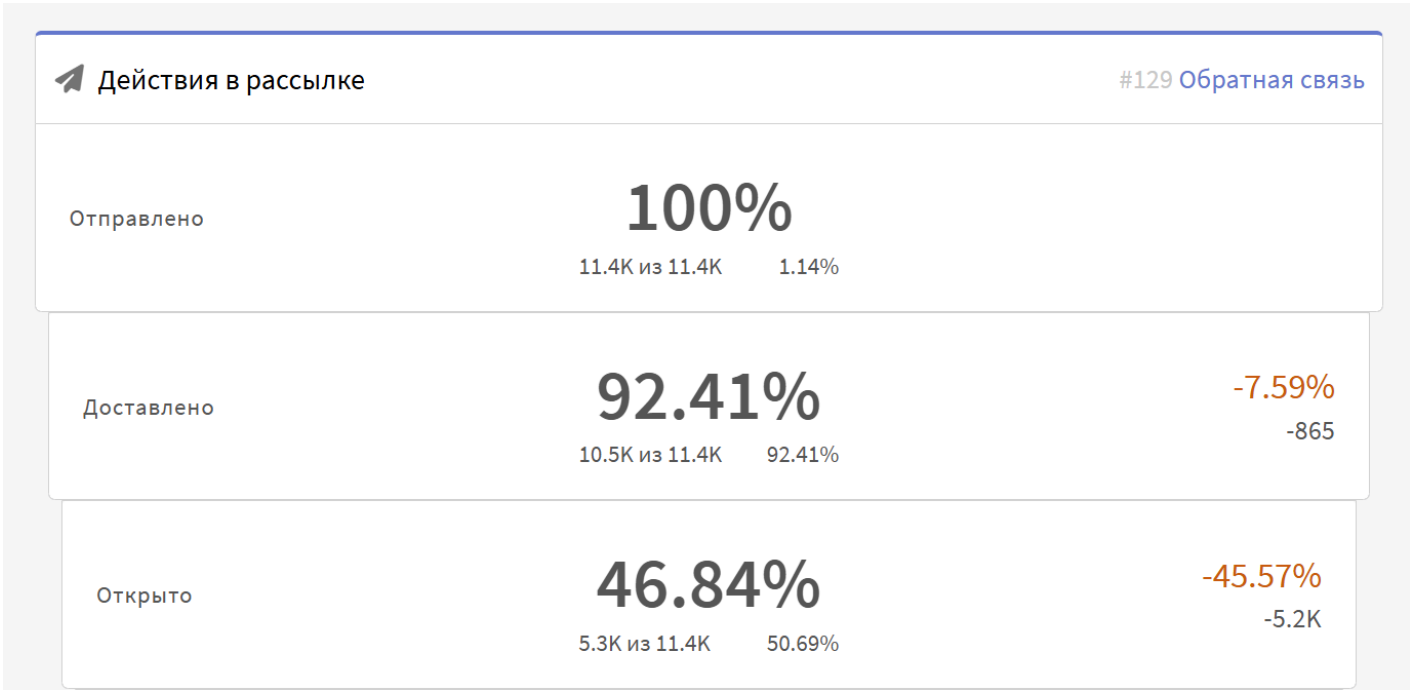
Следующим шагом **“Добавьте фильтр”** по **“Рассылкам”**. В отобразившемся списке выберите нужную рассылку.



В результате вы получите статистику по выбранной рассылке.



Действия в Push-рассылке или шаблоне могут стать шагами в **интерактивной воронке конверсии**:



WhatsApp

Первая WhatsApp-рассылка

WhatsApp* — это эффективный канал коммуникации, который позволяет выстроить диалог с клиентом в режиме реального времени. Через этот канал вы можете сообщать подписчикам о важных событиях, изменениях по статусам заказа и предстоящих мероприятий. Открываемость писем в мессенджере достигает 90%.

Данное руководство поможет вам запустить первую рассылку в WhatsApp*.

Шаг 1. Подключите WhatsApp Business API через провайдера

Чтобы отправлять массовые рассылки в WhatsApp*, необходимо подключить WhatsApp* Business API. В платформе Altcraft такое подключение возможно через официальных провайдеров: **Edna** или **360dialog**. Зарегистрируйте аккаунт у одного из провайдеров. Для этого вам потребуются номер телефона и **верифицированный аккаунт** в Facebook* Business Manager.

TIP

Если вы хотите зарегистрировать WABA аккаунт через Edna, оставьте заявку по этой ссылке.

360dialog собирает заявки через эту форму.

TIP

Вы можете использовать городской +7(495) или мобильный +7(900) номер. Для регистрации подойдет номер телефона, который не связан с приложением WhatsApp*, или номер, который не был активен в WhatsApp* в течение 6 месяцев.

CAUTION

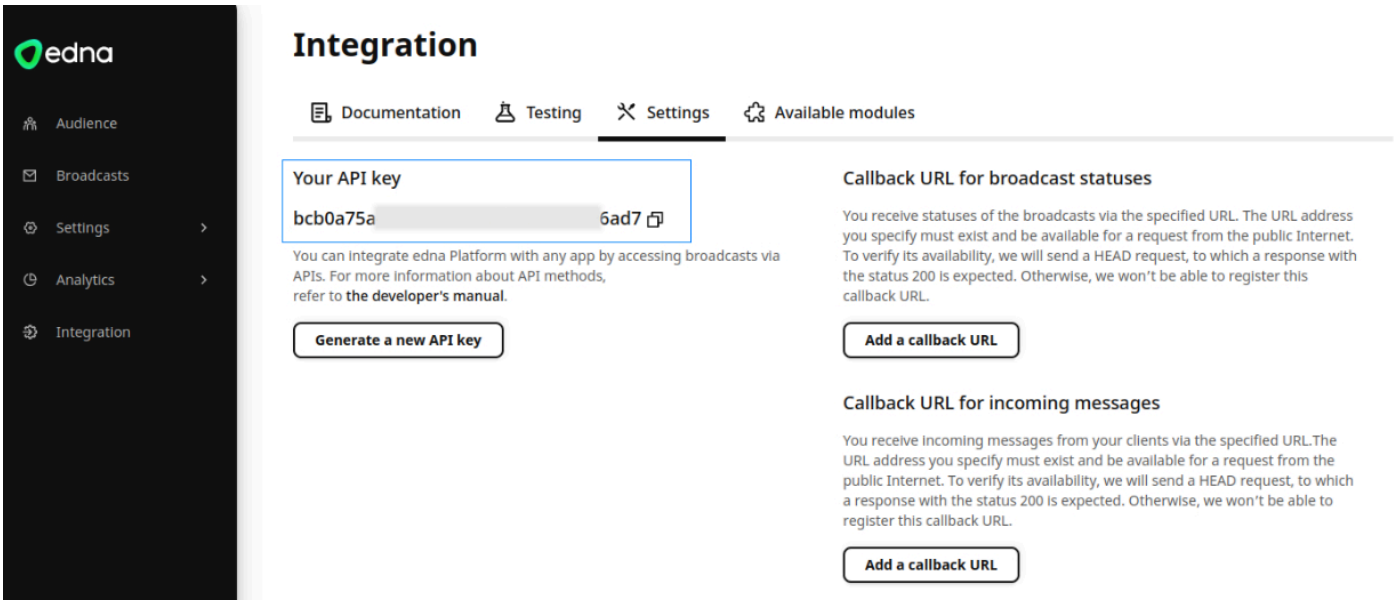
Интеграция с WhatsApp* Business API и последующая отправка сообщений не тарифицируются дополнительно на стороне Altcraft. Однако вам потребуется оплачивать услуги вашего провайдера (360dialog или Edna). Кроме того, стоимость отправляемых сообщений в WhatsApp* также оплачивается отдельно. Подробнее о тарифах на сообщения в WhatsApp* вы можете узнать здесь.

Шаг 2. Получите API ключ

API ключ можно получить в личном кабинете провайдера:

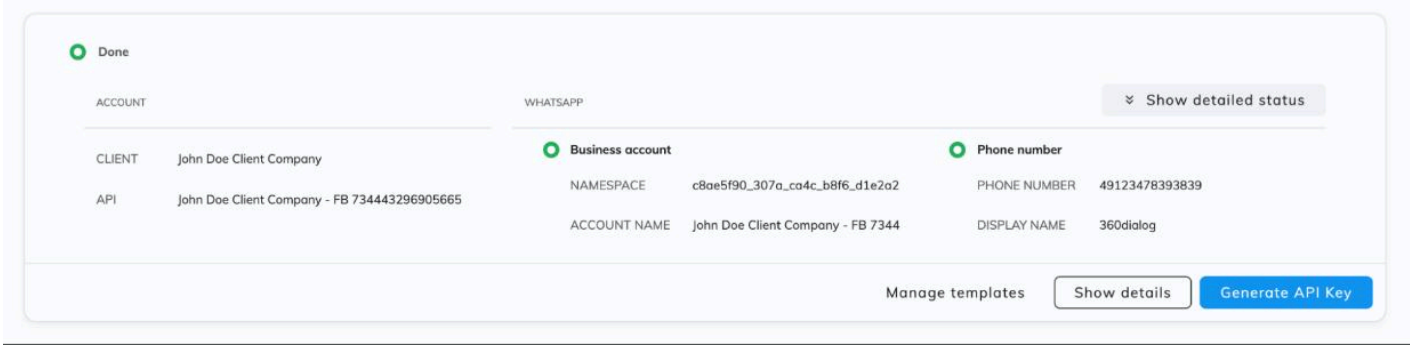
Edna

В личном кабинете Edna откройте раздел **Интеграции** и перейдите на вкладку **Настройки**:



360dialog

В 360 Client Hub перейдите в раздел *WhatsApp Accounts** и нажмите на кнопку **Generate API key**:

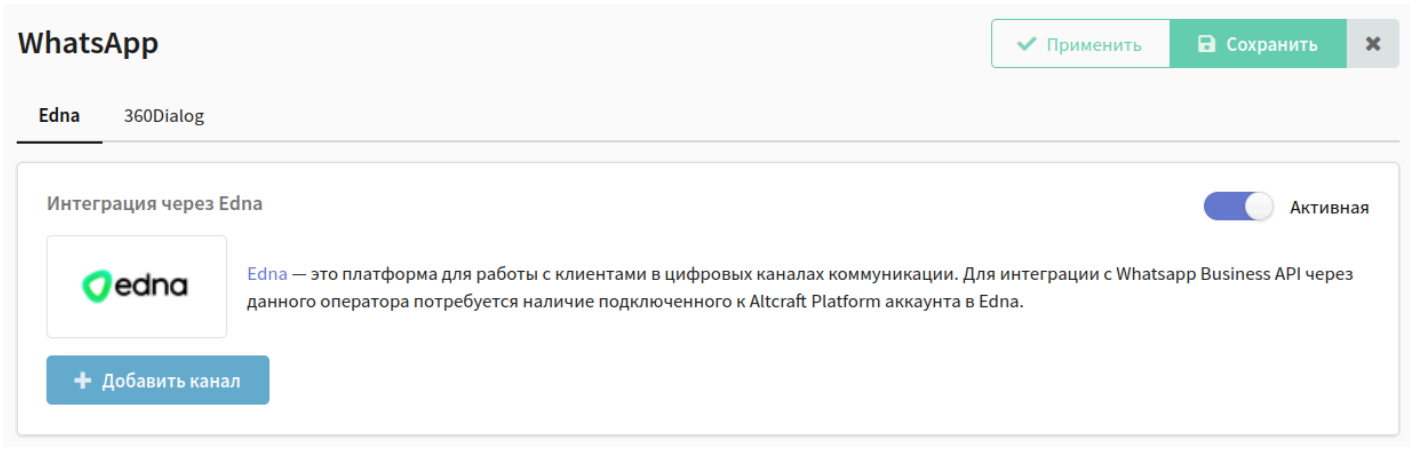


Шаг 3. Настройте интеграцию

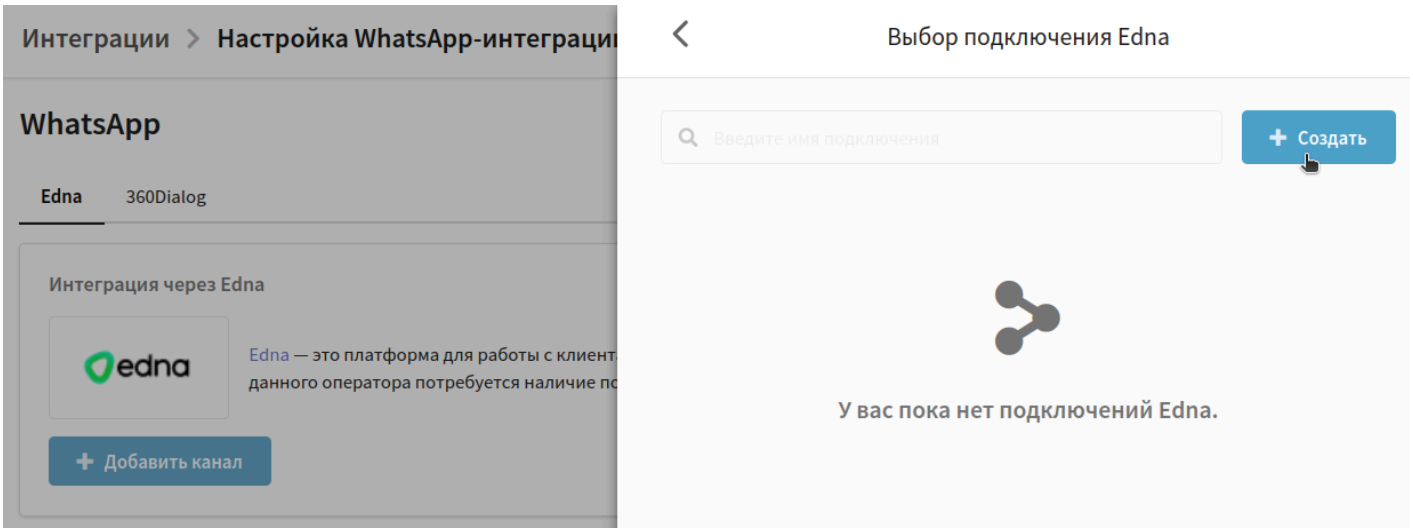
Вернитесь в платформу и откройте раздел **Интеграции**. Выберите тип интеграции — *WhatsApp.** Перейдите на вкладку вашего провайдера.

Edna

1. **Активируйте** интеграцию с помощью переключателя в правом верхнем углу. Нажмите **Добавить канал**:

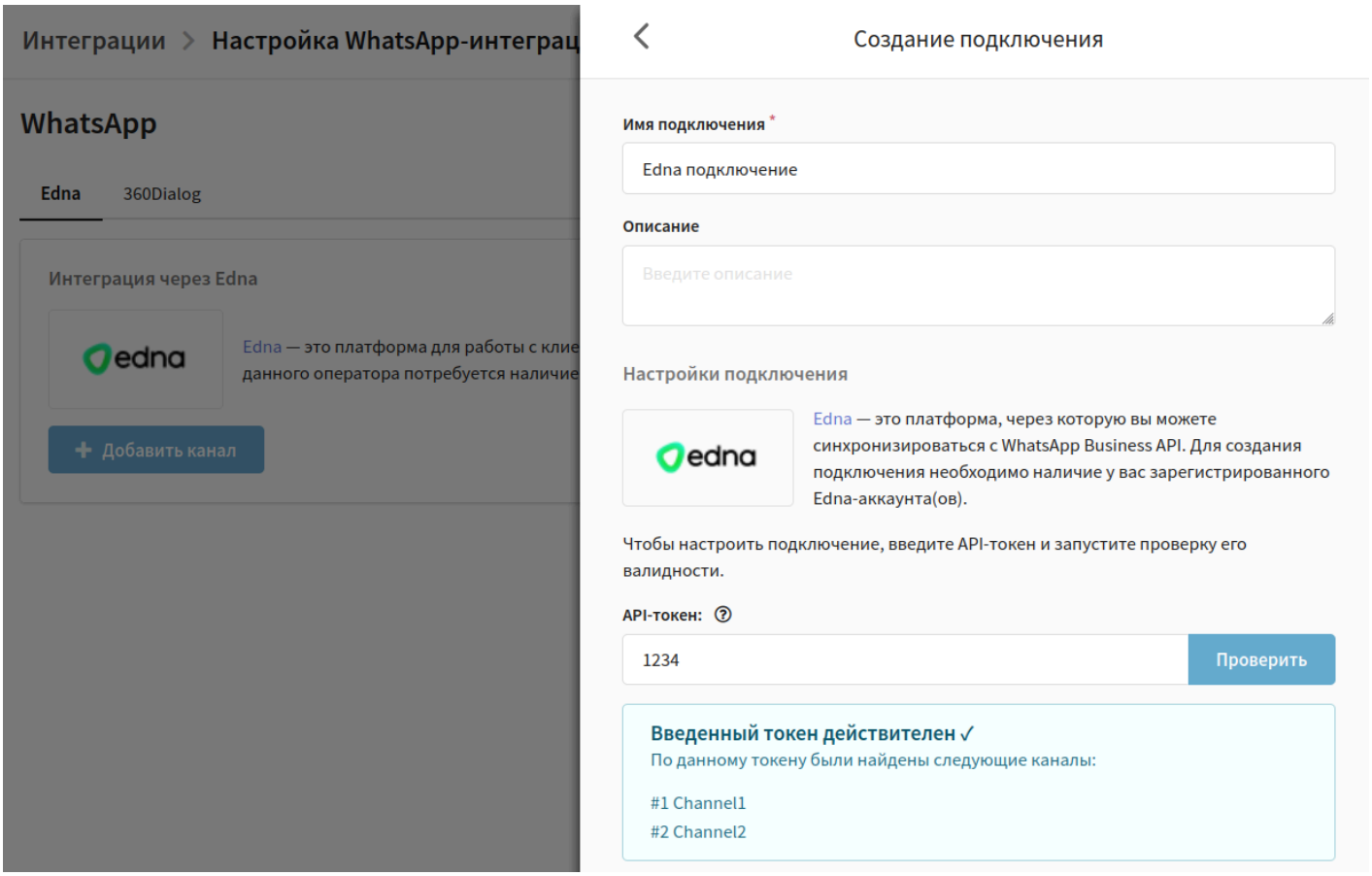


2. Для интеграции необходимо связать аккаунт провайдера с платформой Altcraft. Связывание аккаунтов происходит через подключение. Нажмите **Создать,** чтобы настроить новое подключение:

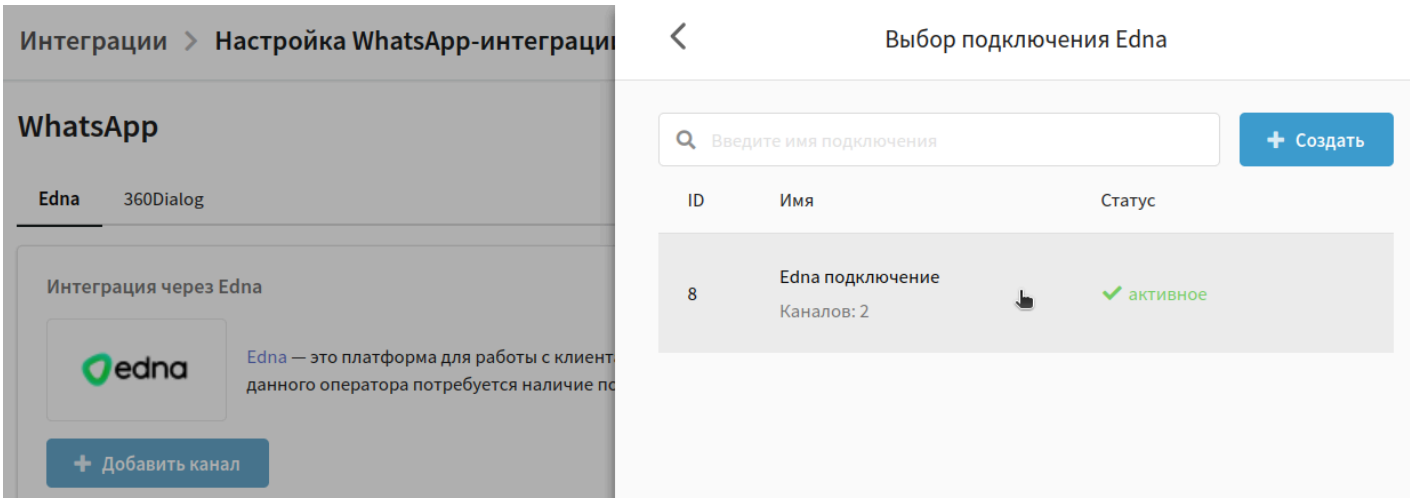


3. В новом окне задайте **имя** и **описание** подключения.

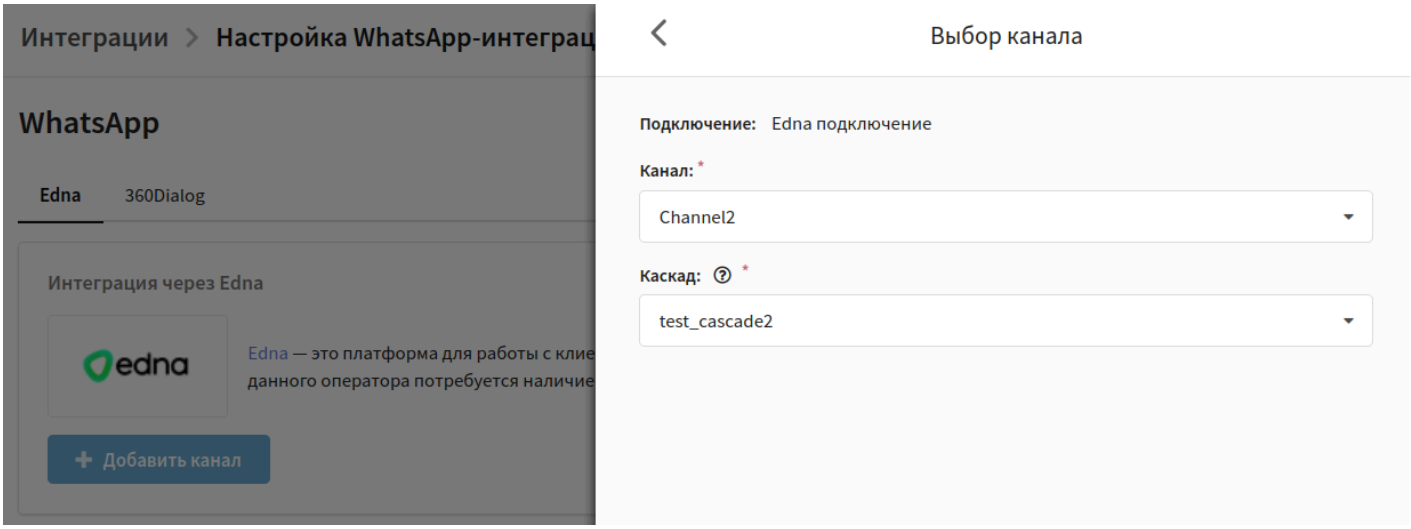
4. Введите **API-токен**, сгенерированный в личном кабинете провайдера, и запустите **проверку,** чтобы подтвердить его валидность. Если токен прошёл проверку, в окне отобразятся данные по доступным каналам в акккаунте:



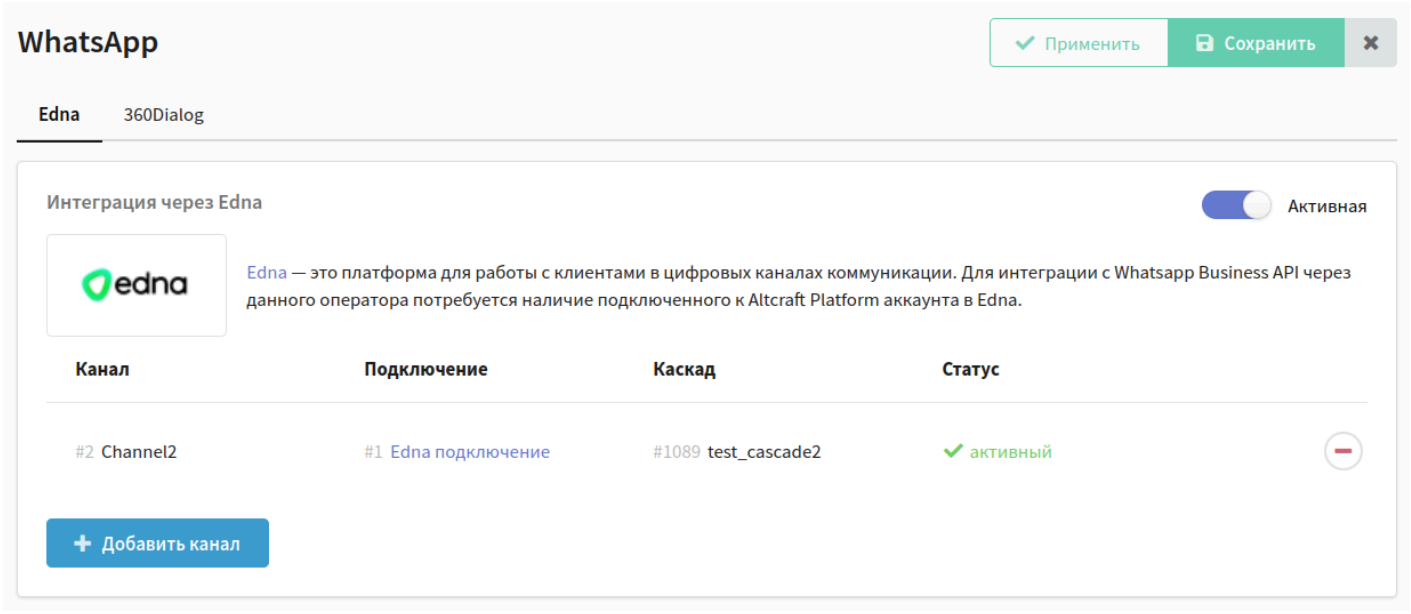
5. Нажмите кнопку **Создать** в правом нижнем углу, чтобы сохранить подключение. Вы вернётесь всписок подключений. Выберите только что созданное:



6. Выберите **канал** и **каскад**. Канал должен находиться в статусе “Работает исправно”, а каскад должен использовать только один канал. Затем нажмите **Добавить** в правом нижнем углу.



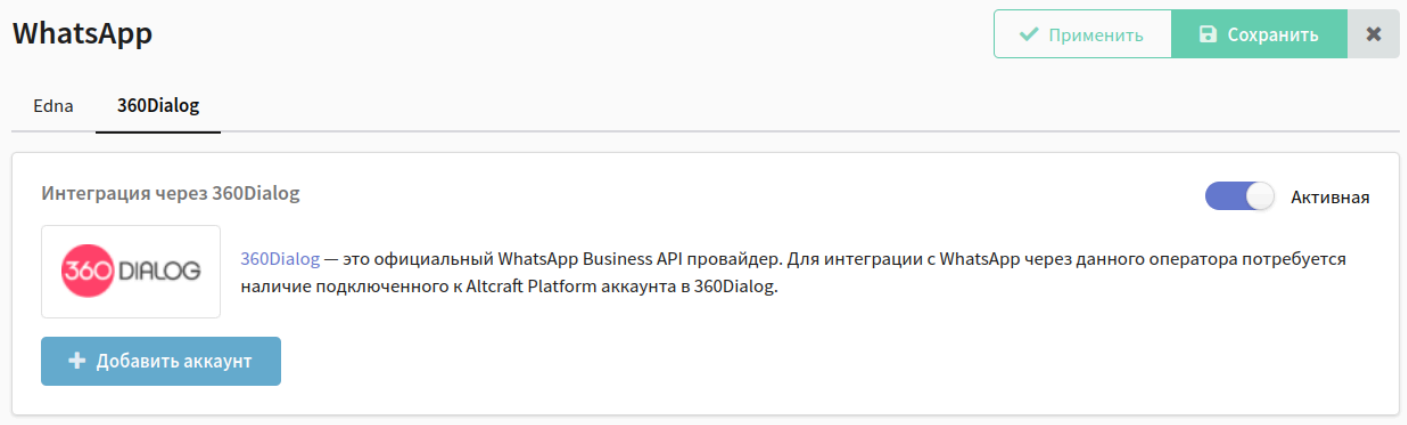
Данные по добавленному Edna-каналу отобразятся в интеграции. При необходимости вы можете добавить несколько каналов для Edna.



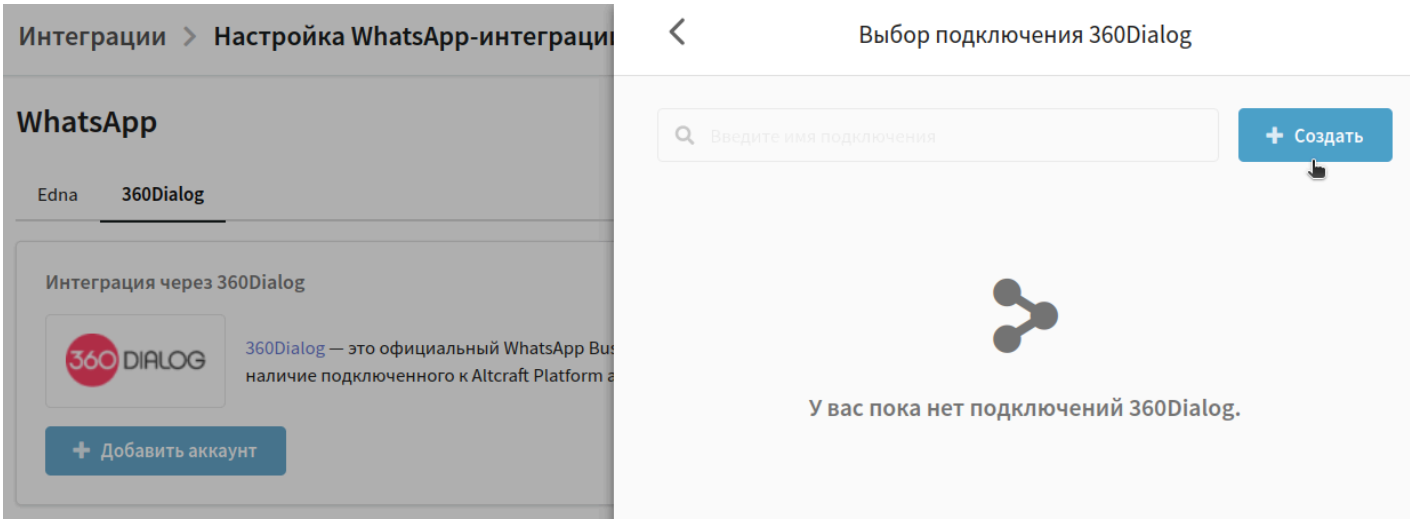
7. **Сохраните интеграцию** с помощью кнопок в правом верхнем углу.

360dialog

1. **Активируйте** интеграцию с помощью переключателя в правом верхнем углу. Нажмите **Добавить аккаунт**:

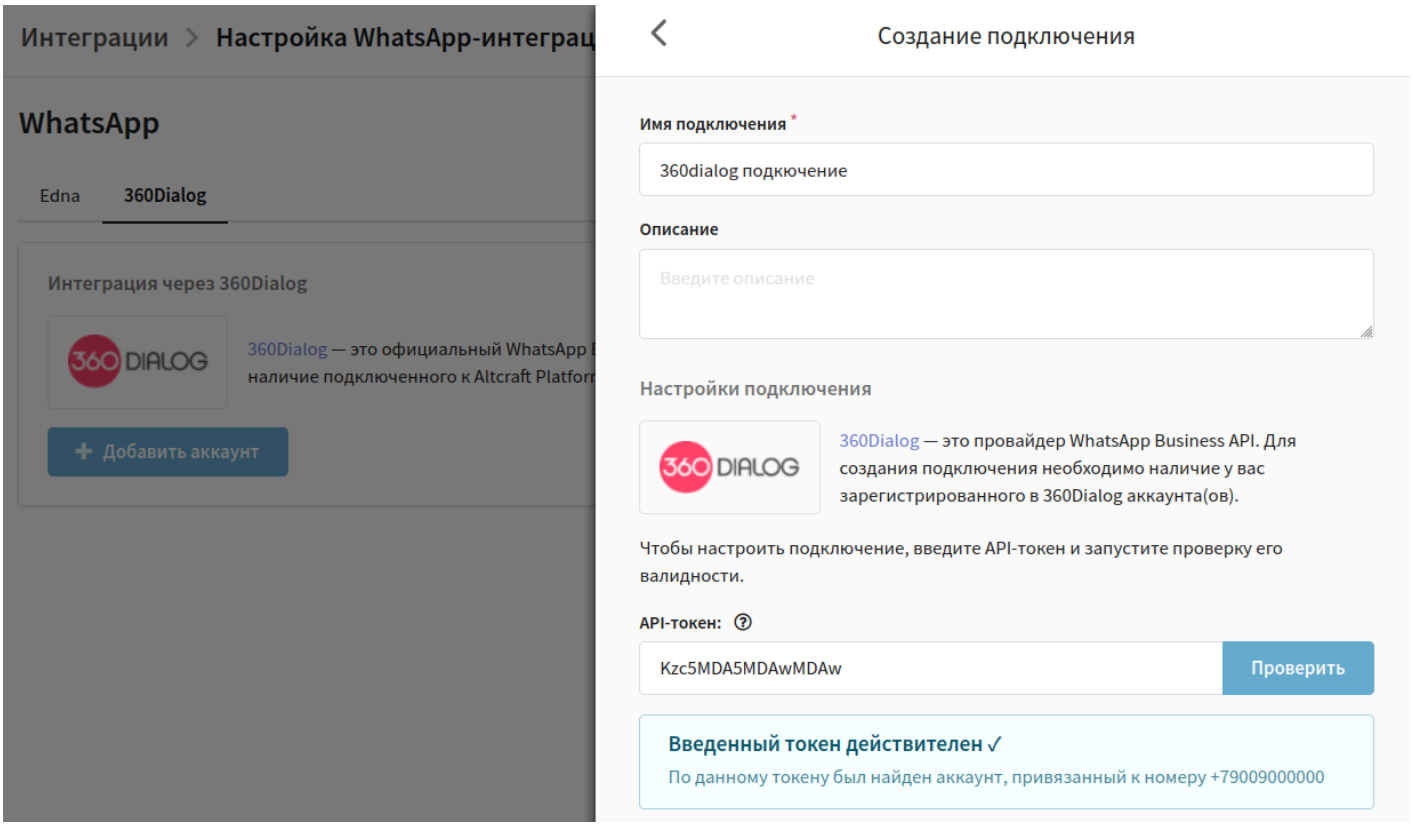


2. Для интеграции необходимо связать аккаунт провайдера с платформой Altcraft. Связывание аккаунтов происходит через подключение. Нажмите **Создать**, чтобы настроить новое подключение:

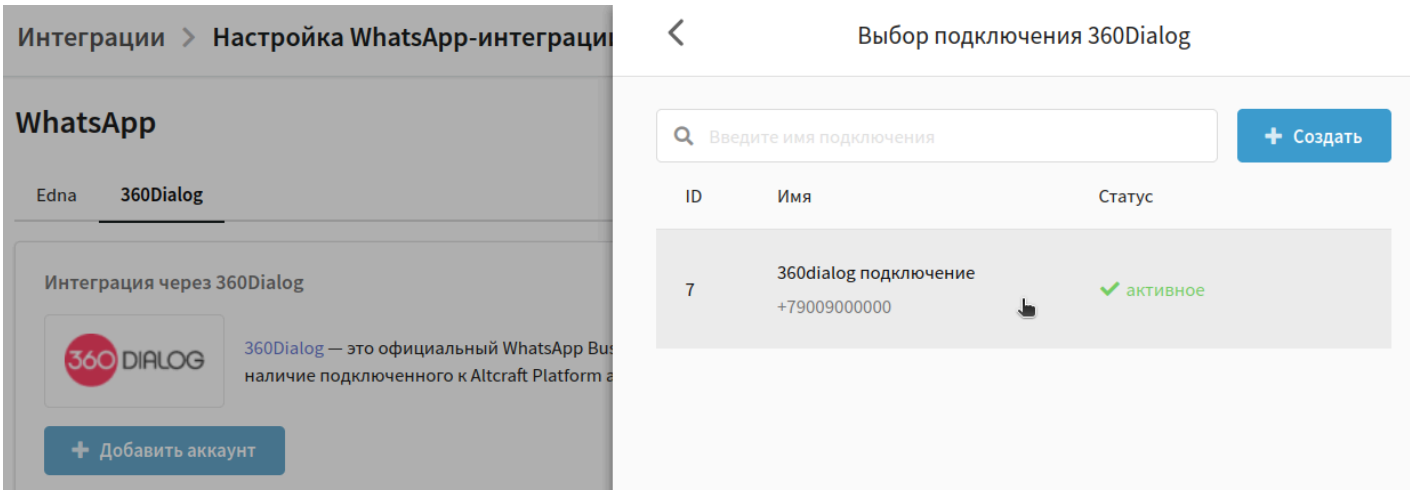


3. В новом окне задайте **имя** и **описание** подключения.

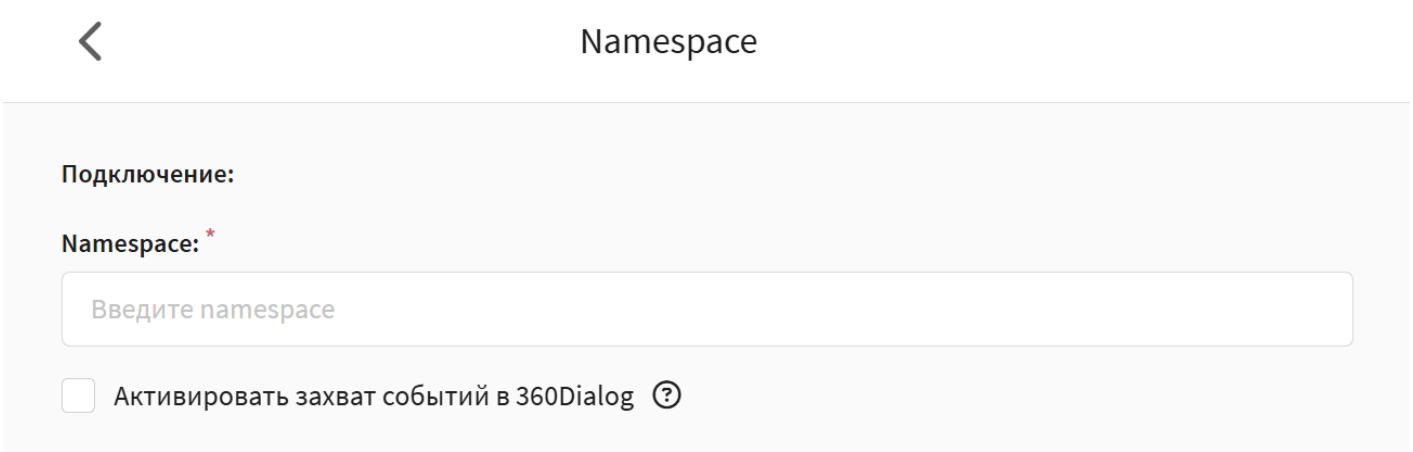
4. Введите **API-токен**, сгенерированный в личном кабинете провайдера, и запустите **проверку**, чтобы подтвердить его валидность. Если токен прошёл проверку, в окне отобразится номер телефона, привязанный к подключаемому аккаунту:



5. Нажмите кнопку **Создать** в правом нижнем углу, чтобы сохранить подключение. Вы вернётесь в список подключений. Выберите только что созданное:

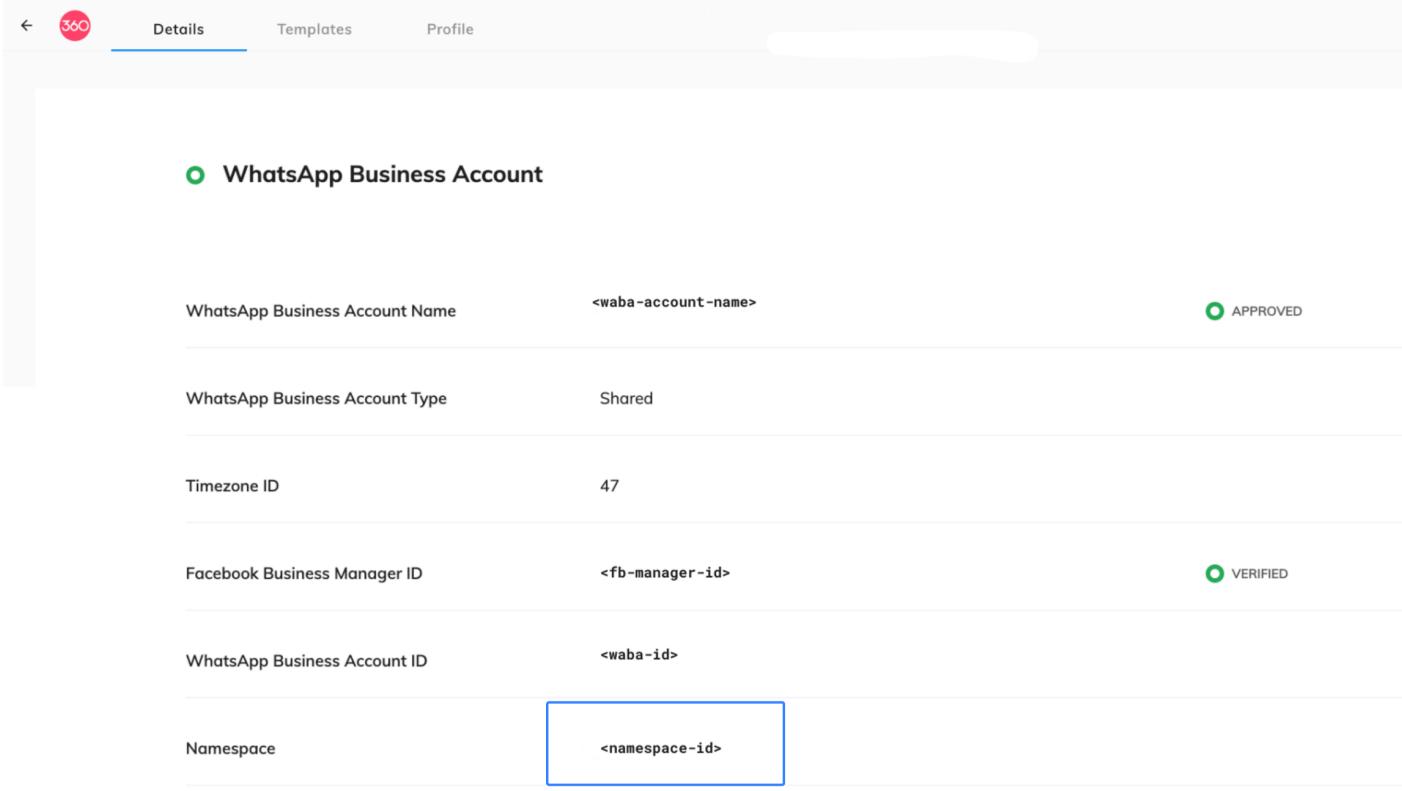


Затем заполните поле **“Namespace”**:

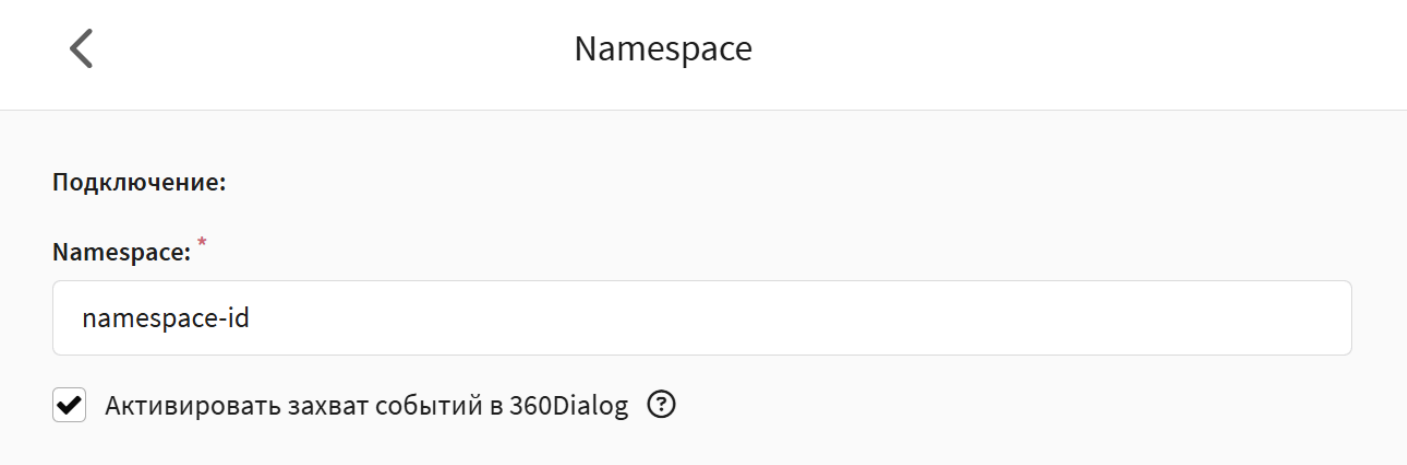


“Namespace” — это свободная строка. В каждом аккаунте провайдера 360dialog хранятся созданные для него шаблоны, привязанные через данный оператор. Namespace можно посмотреть в личном кабинете провайдера в настройках аккаунта.

Перейдите в настройки личного кабинета 360Dialog:



Скопируйте оттуда значение Namespace, вернитесь в платформу и вставьте его в нужное поле:



При желании вы можете активировать опцию **“Захват событий в 360Dialog”**.

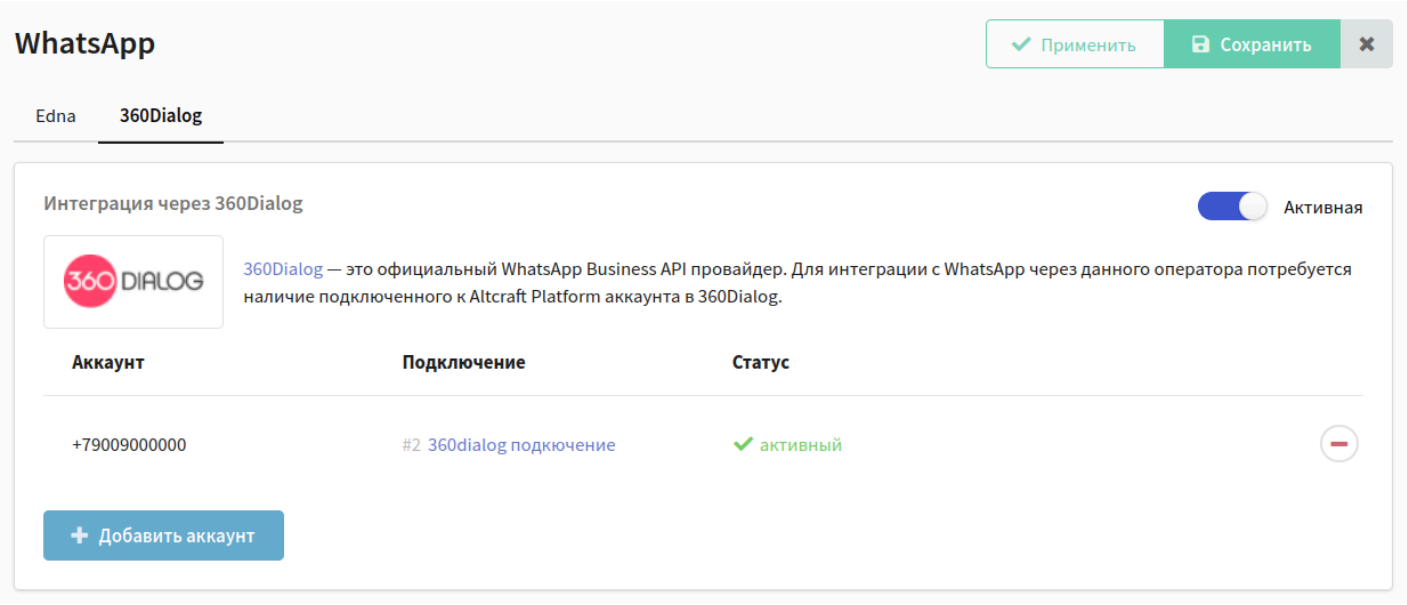
Параметр **“Активировать захват событий”** включает отправку данных о событиях WhatsApp в платформу Altcraft из 360Dialog.

Если он включен, после сохранения интеграции в 360Dialog автоматически добавится специальная ссылка (Callback URL), через которую Altcraft будет получать информацию о статусах сообщений, действиях пользователей и других событиях.

Если передача этих данных не нужна, отключите параметр.

Нажмите **“Добавить”** в правом нижнем углу.

После этого вы увидите данные по добавленному аккаунту. При необходимости вы можете добавить несколько аккаунтов для 360dialog.



7. **Сохраните интеграцию** с помощью кнопок в правом верхнем углу.

Шаг 4. Подпишите клиентов на WhatsApp канал

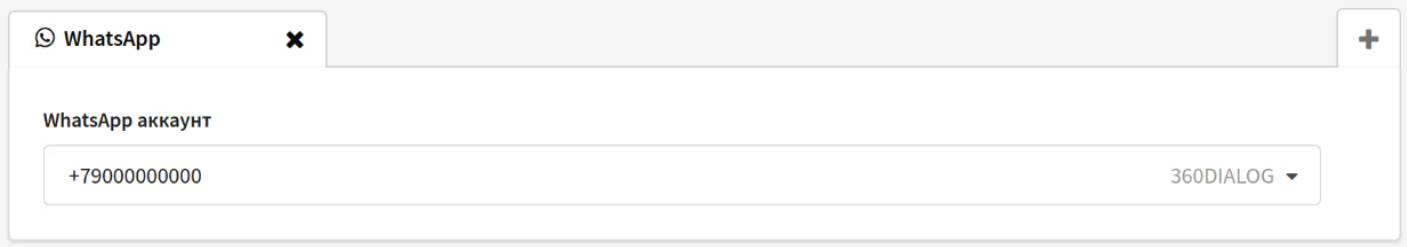
Для отправки сообщения пользователю в WhatsApp* требуется:

- Наличие номера телефона пользователя в базе данных;
- Наличие у профиля подписки на WhatsApp*-канал ресурса;
- Согласие пользователя на получение сообщений от вашей компании.

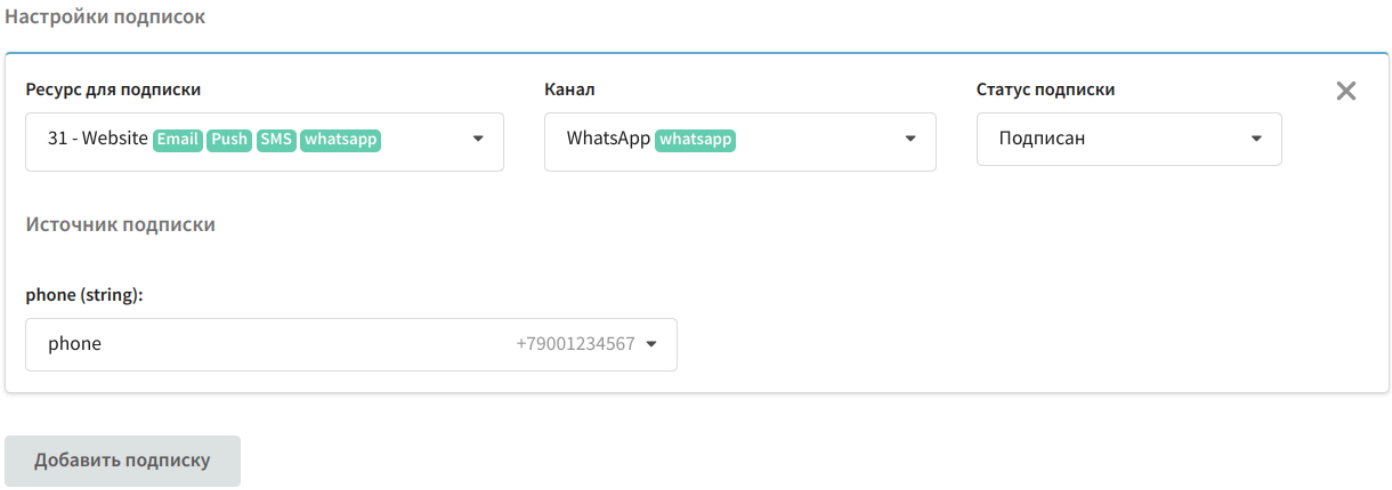
CAUTION

Номер телефона профиля, на который вы отправляете сообщения, должен быть записан в международном формате. Необходимо использовать следующий формат: +79001234567.

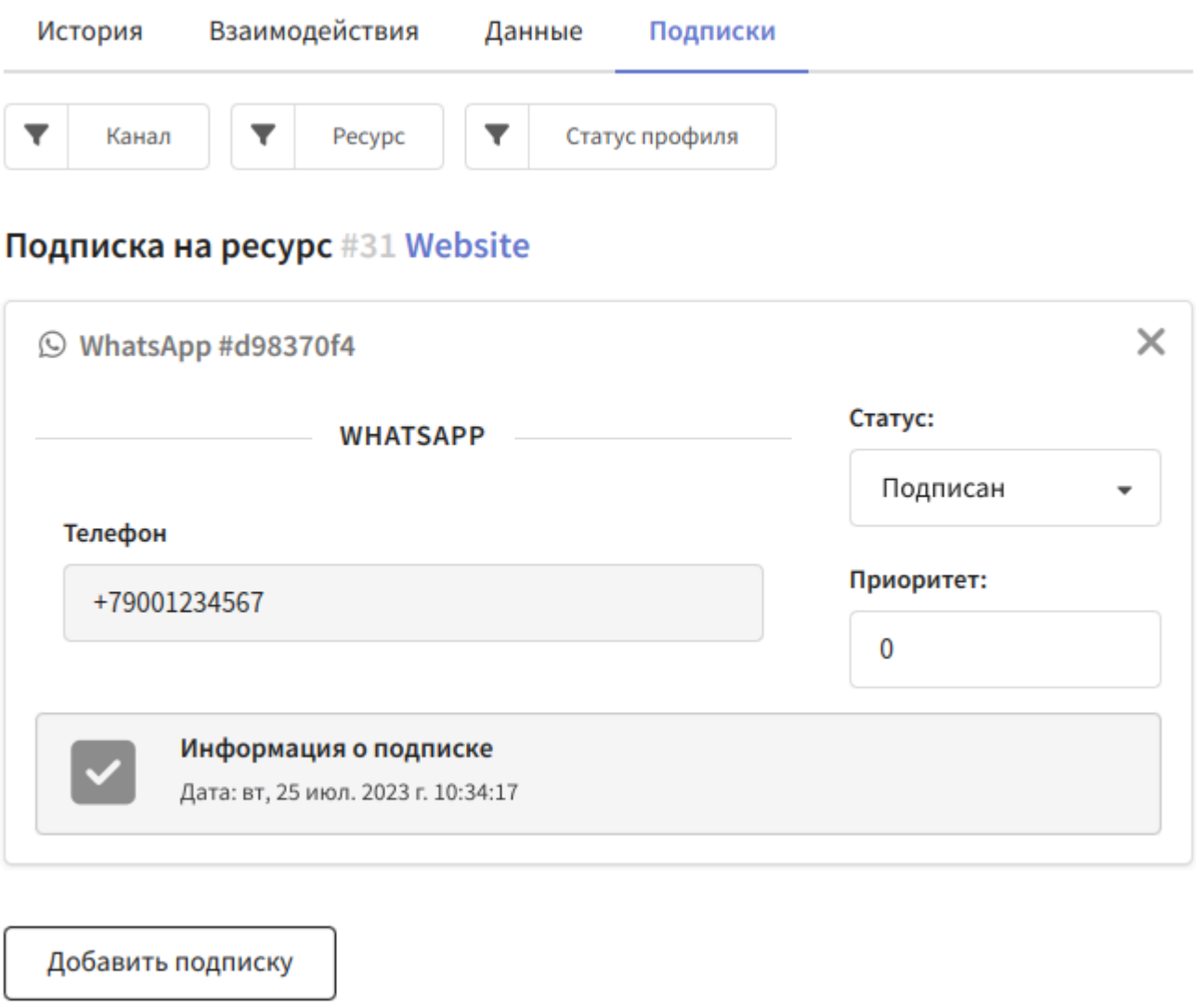
Сначала выберите или создайте базу профилей, в которой будут храниться данные подписчиков, в том числе номера телефонов. Затем создайте новый ресурс или отредактируйте существующий, добавив WhatsApp*-канал. Для настройки канала выберите подключенный ранее WhatsApp*-аккаунт:



Далее импортируйте данные подписчиков в платформу — например, через файл. При импорте добавьте им подписку на ресурс с WhatsApp*-каналом. В качестве **источника подписки** выберите колонку вашего файла, в которой записаны номера телефонов.



Для каждого клиента будет создан [цифровой профиль](), объединяющий личные данные, историю действий и подписки на ресурсы. Если при импорте клиенту была успешно добавлена подписка на WhatsApp*-канал, эти данные отобразятся в его истории:



Шаг 5. Создайте шаблон в личном кабинете провайдера

В WhatsApp* Business API доступны два типа сообщений:

- **Произвольное сообщение** — сообщение в свободном формате, отправляемое в рамках 24-часового окна;
- **Шаблонное сообщение** — сообщение, шаблон которого ранее был согласован с Facebook*; отправляется после закрытия 24-часового окна.

TIP

24-часовое диалоговое окно открывается, если клиент первым написал компании или ответил на ее сообщение. После этого компания в течение суток может отправить клиенту неограниченное количество произвольных сообщений.

Шаблонные сообщения используются, чтобы инициировать диалог с клиентом или продолжить общение спустя 24 часа после последнего входящего сообщения.

CAUTION

Платформа Altcraft позволяет отправлять только **шаблонные сообщения (HSM)**. Поддерживаются различные типы: текстовые, мультимедийные и сообщения с кнопками.

Шаблонное сообщение состоит из фиксированного текста и переменных. Переменные — это текст, который можно менять перед каждой рассылкой без согласования с Facebook*.

Шаблон создается в личном кабинете провайдера, а в платформе Altcraft вы только подставляете текст в переменные, настраиваете кнопки и добавляете медиа файлы.

TIP

Больше информации о создании шаблонов вы можете найти в документации вашего провайдера:

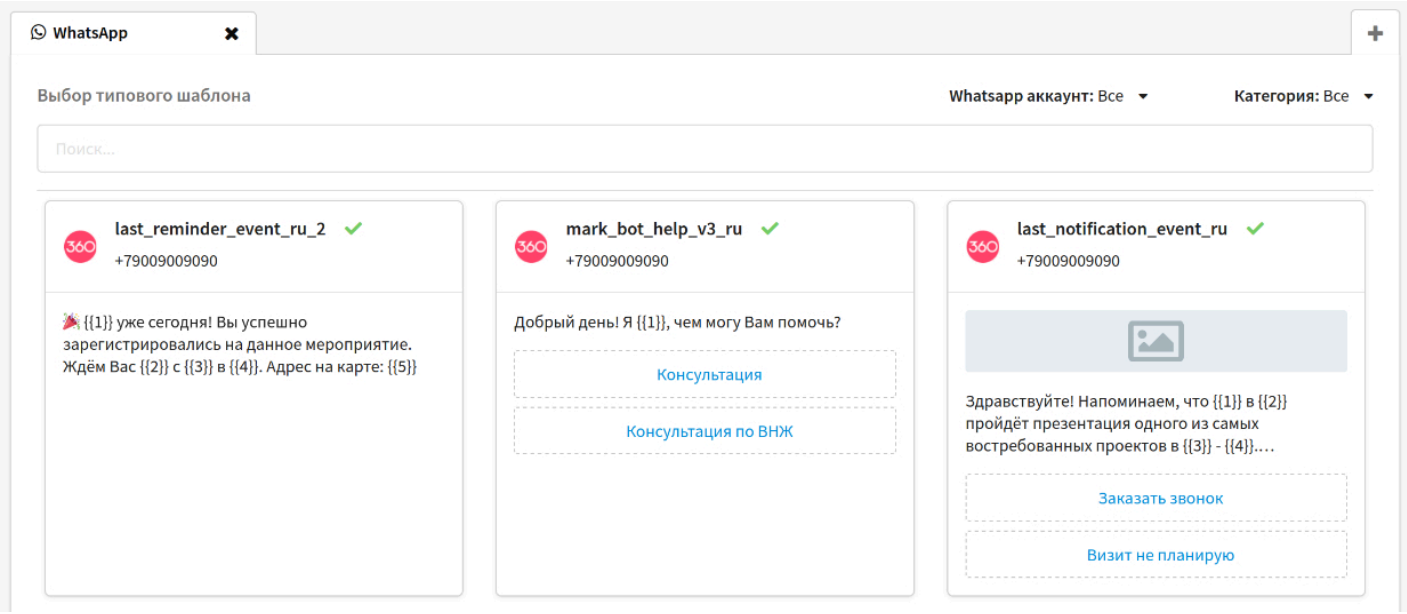
- **Edna:** https://pulse-alpha-six.vercel.app/docs/template/whatsapp/operator-template
- **360dialog:** https://docs.360dialog.com/docs/360-client-hub/template-management-ui

CAUTION

Шаблон, созданный в личном кабинете провайдера, должен пройти верификацию в Facebook*. В рассылках, запускаемых через Altcraft, можно использовать только шаблоны, которые успешно прошли проверку.

Шаг 6. Создайте сообщение на основе шаблона в Altcraft

В платформе Altcraft откройте раздел **Шаблоны** и создайте новый шаблон для WhatsApp*-канала. Все шаблоны, доступные в WhatsApp*-аккаунтах, подключенных через edna или 360dialog, автоматически доступны в платформе и отображаются в редакторе:



Выберите шаблон и создайте сообщение на его основе. Для этого заполните переменные, настройте кнопки и добавьте медиа файлы.

TIP

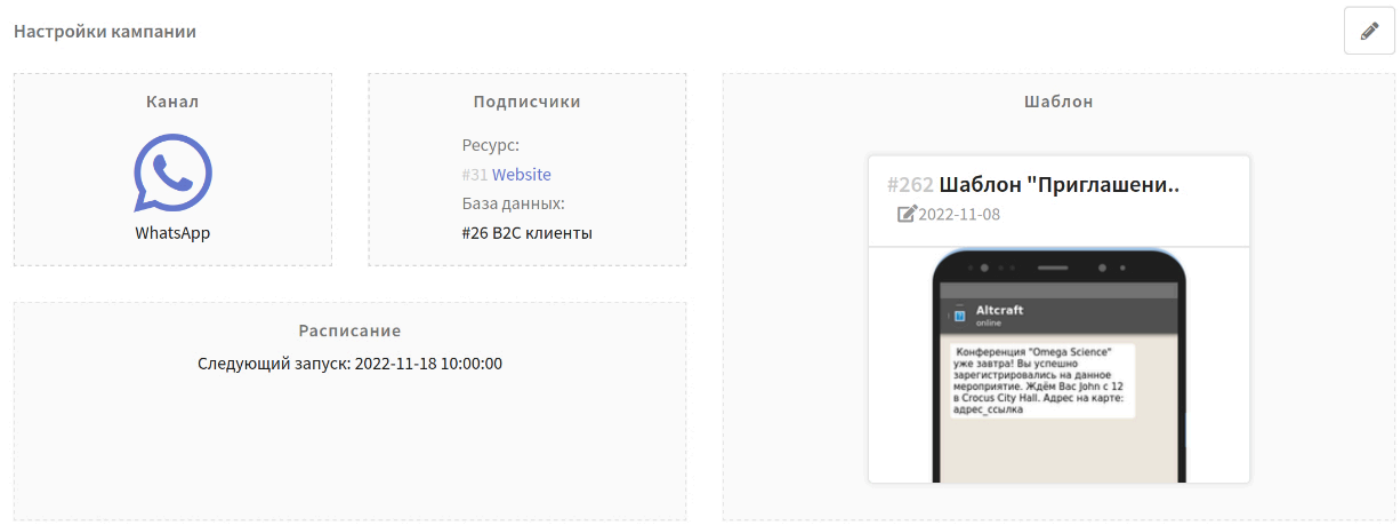
Создание шаблона для WhatsApp канала описывается в **этой статье**.

Шаг 7. Запустите рассылку и оцените эффективность рассылки

Перейдите в раздел **Рассылки** и выберите тип рассылки:

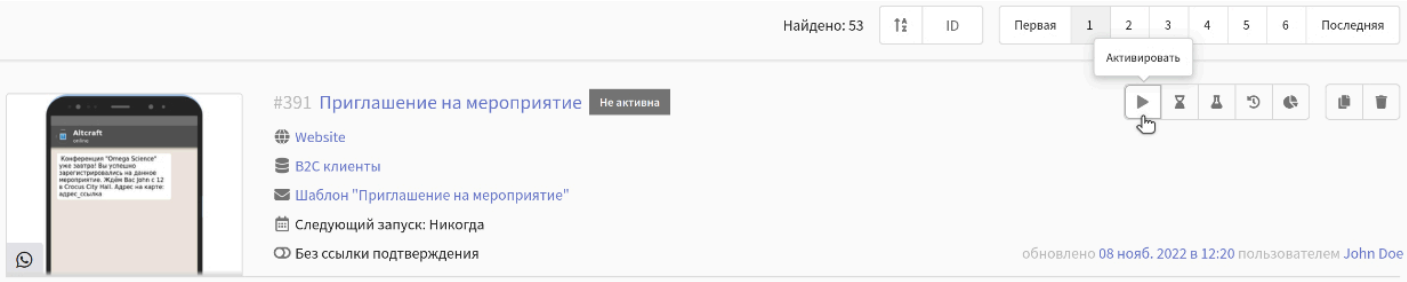
- Броадкаст — разовая рассылка сообщений.
- Регулярная рассылка — периодическая рассылка.

Настройте рассылку: выберите WhatsApp* как канал связи, настроенный ранее ресурс и аудиторию получателей (база или сегмент). Добавьте в рассылку шаблон сообщения и установите расписание:



Сохраните рассылку. Найдите её в списке рассылок выбранного типа и активируйте, кликнув

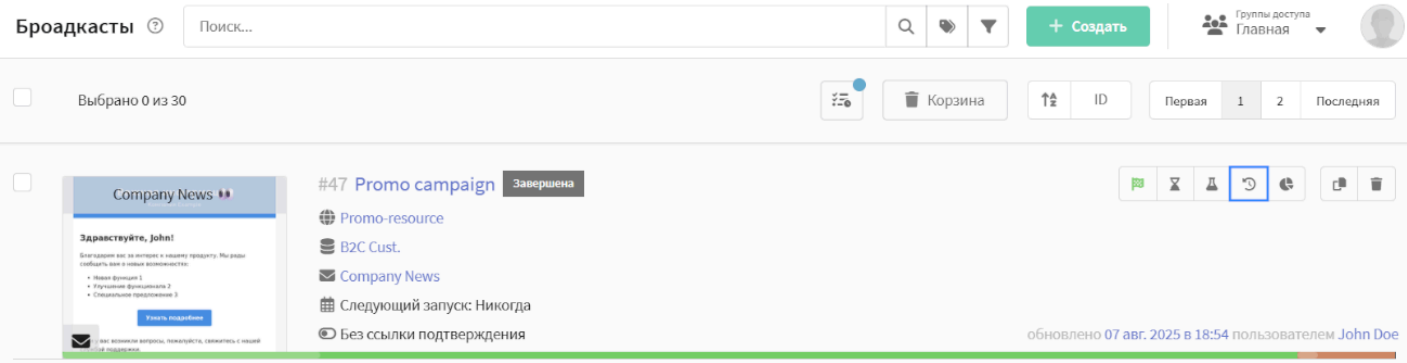
в панели быстрых действий. Активная рассылка начнёт отправлять сообщения, когда наступит время запуска по расписанию.



Журнал рассылки

Журнал рассылки — это инструмент, предоставляющий краткую сводку о ходе выполнения рассылки. В нем фиксируются ключевые события, параметры рассылки и статистика, что позволяет выявить возможные проблемы.

Чтобы открыть журнал рассылки, нажмите на кнопку **“Журнал”** в меню быстрых действий:



Журнал рассылки содержит следующую информацию:

Тип	Описание
Запуск рассылки	Время начала выполнения рассылки
Лимит	Показывает, установлен ли лимит на отправку сообщений
Источник подписчиков	**list** — База данных **segment** — Сегмент (если применимо)
ID базы данных / ID сегмента	Идентификатор используемой базы или сегмента
Время получения данных	Время загрузки контактов из источника (мс)
Количество профилей из источника	Общее число контактов в указанном источнике
Количество валидных подписчиков	Число пользователей с активной подпиской
Количество подписчиков без подписок	Пользователи без подписки на целевой канал (0 в примере)
Количество с некорректными полями	Ошибки в данных подписки (0 в примере)
Количество дубликатов	Повторяющиеся контакты (40 в примере)
Количество отправленных сообщений	Фактическое число успешных отправок
Рассылка завершена	Время окончания рассылки

Тип	Описание
Всего времени затрачено	Общая продолжительность рассылки

Для более точной оценки результатов вашей рассылки воспользуйтесь отчётом по каналам. Чтобы перейти в отчёт, нажмите на кнопку аналитики

в панели быстрых действий. Вы получите информацию о количестве отправленных, доставленных и недоставленных сообщений, а также о количестве новых подписок на WhatsApp*-канал. Для удобства группировки можно выбрать **пресет** показателей для WhatsApp*.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Viber™

Первая Viber™-рассылка

Данное руководство поможет вам запустить первую рассылку в **Viber™**. Интеграция Altcraft Platform с Viber™ осуществляется с помощью **Devino Telecom**, **SMS Traffic**, **MTC OmniChannel** или **Rapporto**. Если у вашей компании нет аккаунта ни в одном из этих сервисов, зарегистрируйте его.

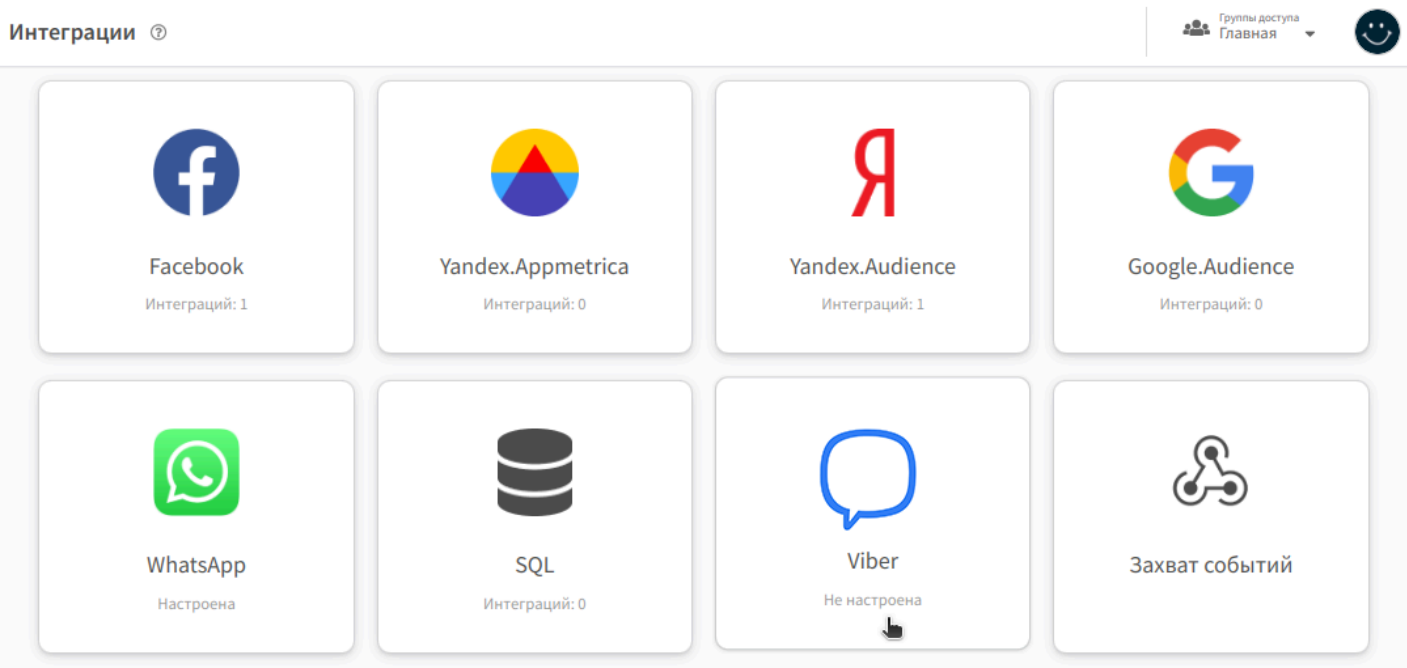
INFO

Стоимость сообщений определяется на стороне провайдера. Altcraft Platform не взимает дополнительной платы за отправку сообщений в Viber™.

Шаг 1. Настройте интеграцию

1. В главном меню откройте раздел **“Интеграции”.**

2. Выберите тип интеграции — **Viber™.**



3. Выберите провайдера — **Devino Telecom**, **SMS Traffic**, **MTC OmniChannel** или **Rapporto**.

Интеграция с Viber

Применить | Сохранить | ×

Devino telecom | SMS Traffic | MTS OmniChannel | Rapporto

Интеграция через Devino telecom — Активная

Devino Telecom — разработчик облачных платформ для общения бизнеса со своей аудиторией в цифровых каналах коммуникации. Для интеграции с Viber через данного оператора потребуется наличие подключенного к Altcraft Platform аккаунта в Devino Telecom.

+ Добавить подключение

Интеграция с Viber

Применить | Сохранить | ×

Devino telecom | SMS Traffic | MTS OmniChannel | Rapporto

Интеграция через SMS Traffic — Активная

SMS Traffic — разработчик облачных платформ для взаимодействия бизнеса со своей аудиторией в цифровых каналах коммуникации. Для интеграции с Viber через этого провайдера требуется наличие аккаунта в SMS Traffic, подключенного к Altcraft Platform.

+ Добавить подключение

Callback URL:

https://click. /fbp/v1/viber/1?acc=4h6NRD82Qgo%3D&integ_type=sms_traffic

Интеграция с Viber

Применить | Сохранить | ×

Devino telecom | SMS Traffic | MTS OmniChannel | Rapporto

Интеграция через MTS OmniChannel — Активная

MTC OmniChannel платформа для рассылки сообщений через различные цифровые каналы. Для интеграции с Viber через данного оператора потребуется наличие подключенного к Altcraft Platform аккаунта в MTC OmniChannel.

Параметр class — это опциональный параметр для управления кастомизированной маршрутизацией сообщений.

Этот параметр задается в сервисе MTC OmniChannel и принимает числовое значение. Введите число, соответствующее классу отправки сообщений в Viber.

Подключение	Статус	Лимит	Class
#11 Viber connection	✓ активный	100	

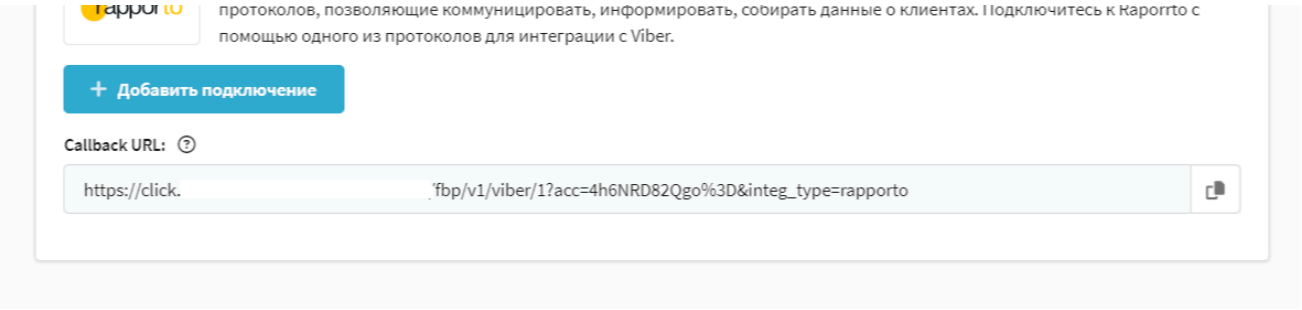
Интеграция с Viber

Применить | Сохранить | ×

Devino telecom | SMS Traffic | MTS OmniChannel | Rapporto

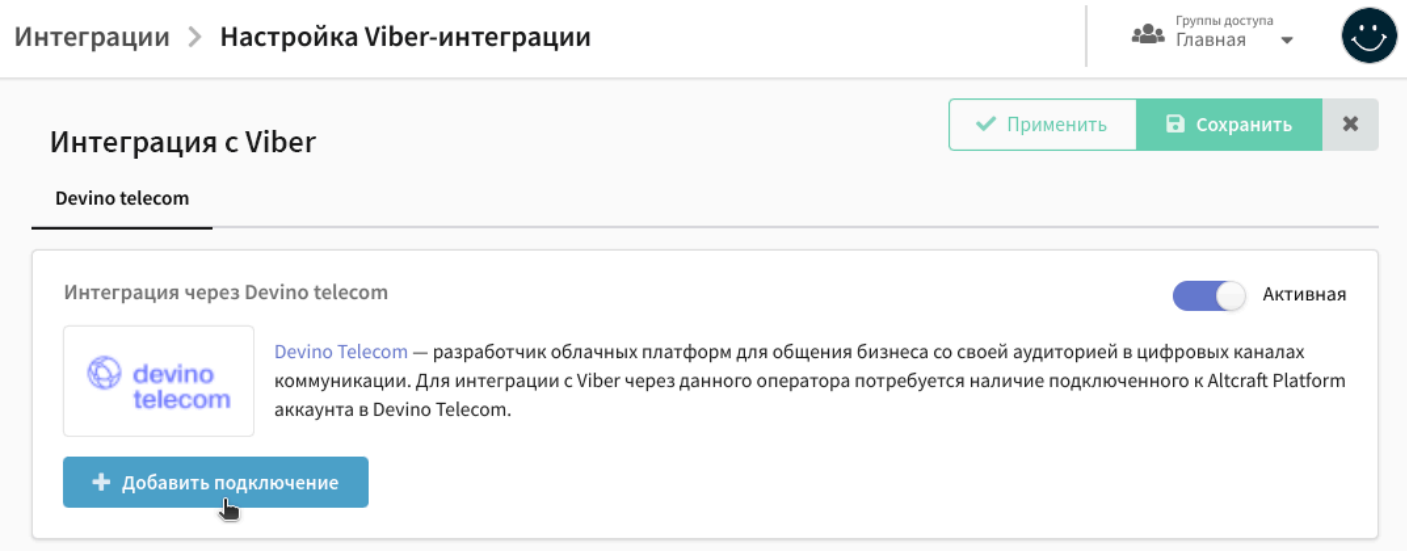
Интеграция через Rapporto — Активная

Rapporto — платформа для интеграции с мобильными каналами коммуникации. Для подключения доступны разные виды

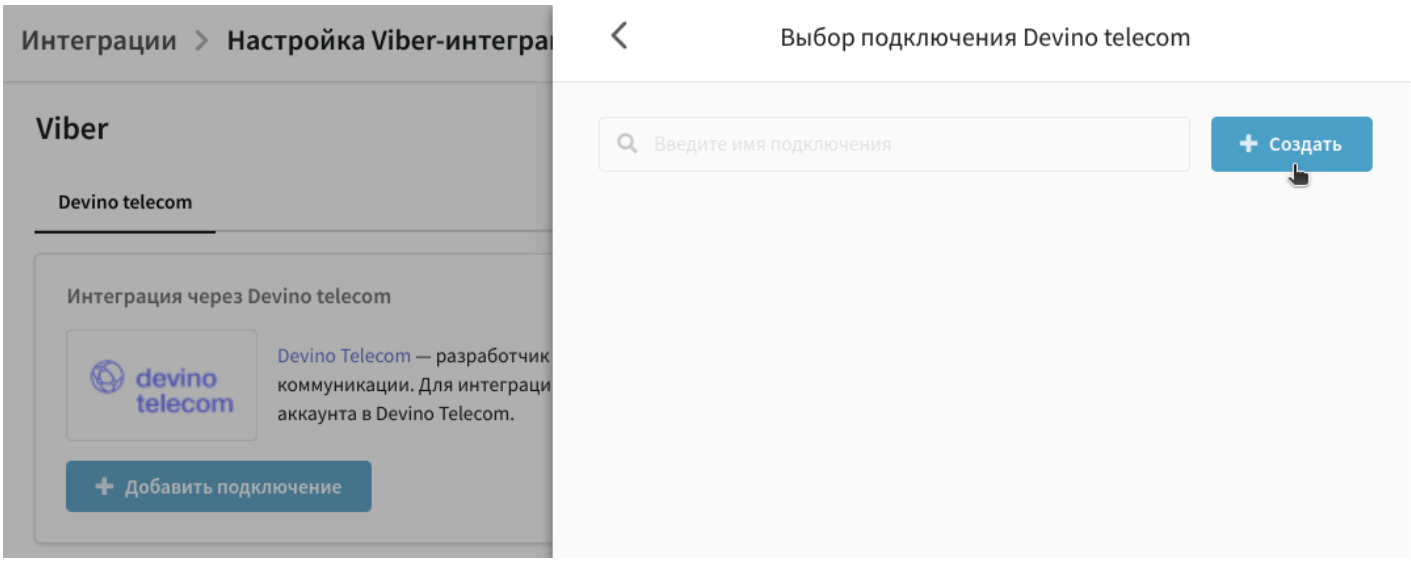


4. **Активируйте интеграцию** с помощью переключателя в правом верхнем углу.

5. Нажмите **“Добавить подключение”:**



6. Справа появится новое окно с доступными подключениями. Подключение позволяет связать аккаунт провайдера с платформой Altcraft. Нажмите **“Создать”**, чтобы настроить новое подключение:



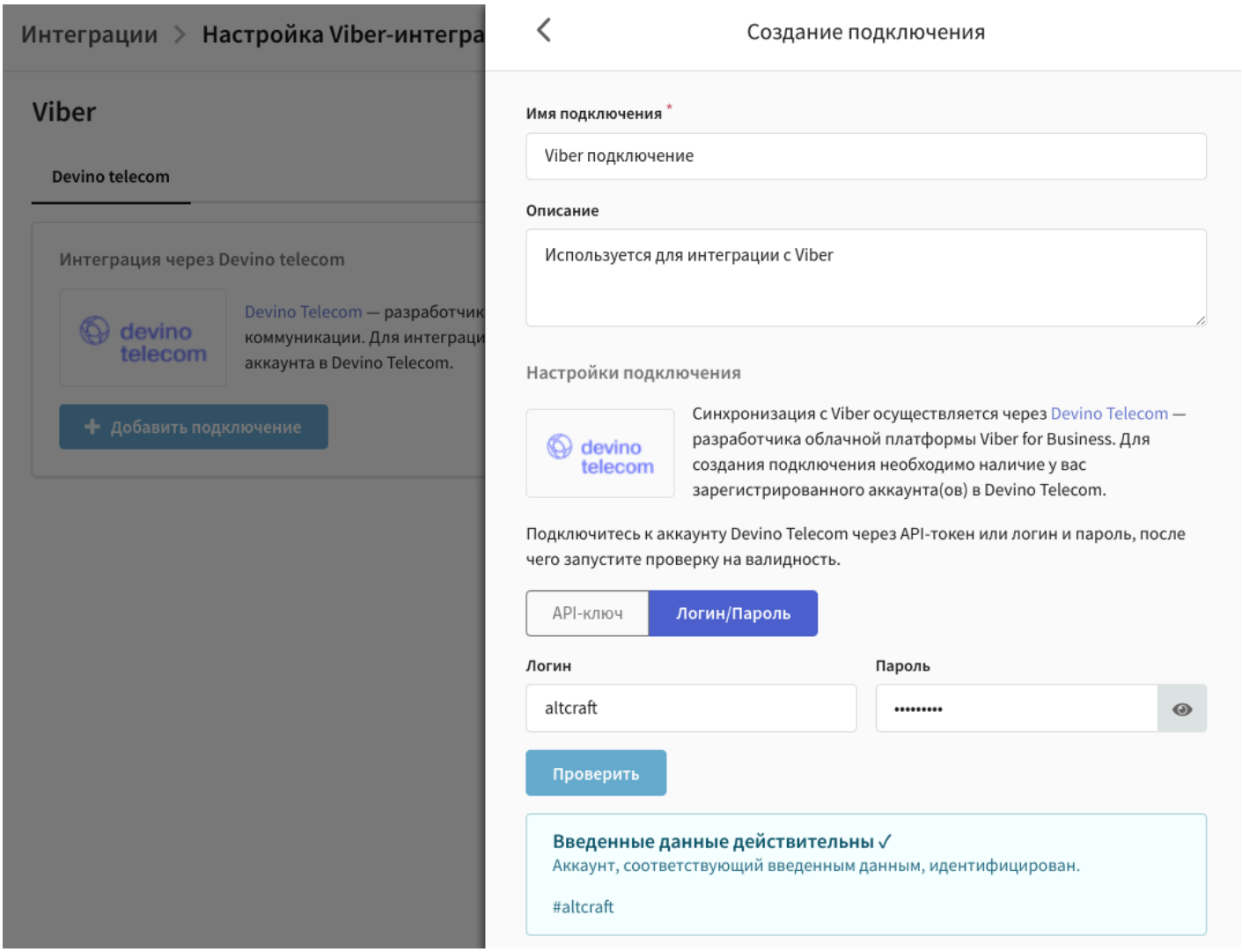
7. В новом окне задайте **имя** и **описание** подключения.

8. Авторизуйтесь в аккаунте провайдера.

TIP

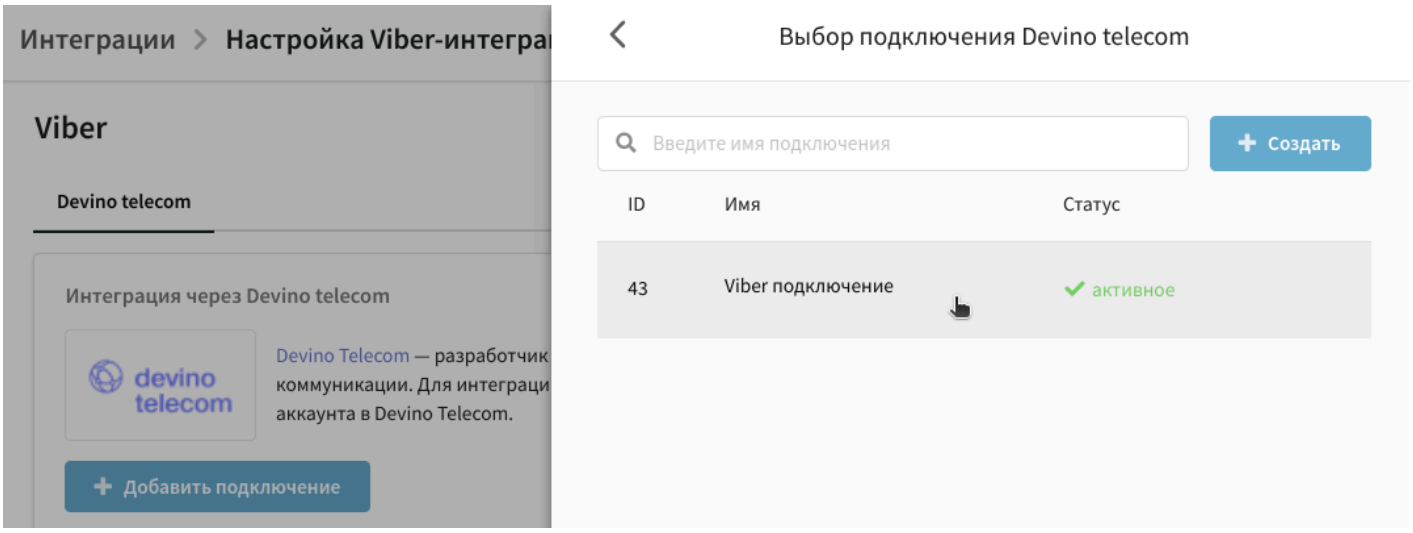
Для **Devino Telecom** существует два способа авторизации. Подробнее о создании подключения можно узнать в этой статье.

9. Введите данные для авторизации и запустите проверку валидности введенных данных с помощью кнопки **“Проверить”**. Если данные действительны, ниже появится информация об успешно идентифицированном аккаунте:

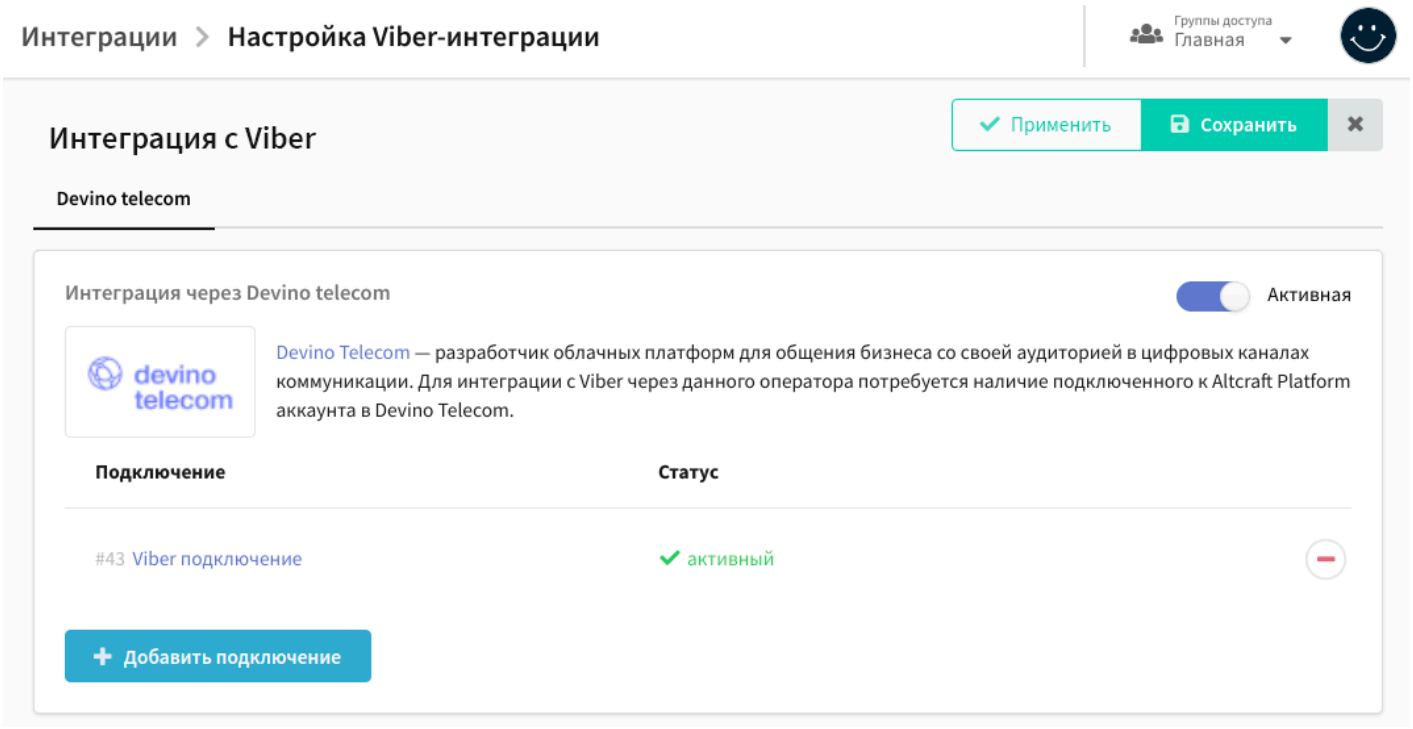


10. Нажмите кнопку **“Создать”** в правом нижнем углу, чтобы сохранить подключение.

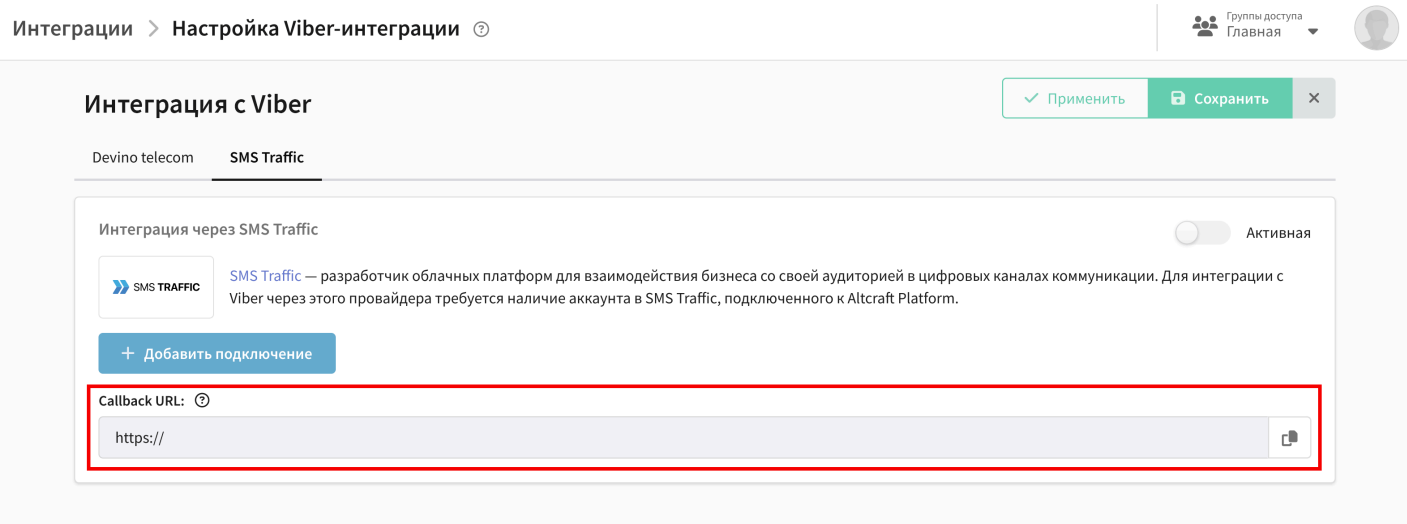
11. Вы вернётесь в список подключений. Выберите только что созданное:



12. Информация о добавленном подключении отобразится в интеграции. Убедитесь, что статус подключения — “активный”. В противном случае проверьте статус вашего аккаунта в личном кабинете провайдера. Имейте в виду, что аккаунт может быть заблокирован, если не внесена абонентская плата.



13. Если вы используете SMS Traffic в качестве оператора, то вам необходимо отправить в техническую поддержку ваш **Callback URL**, указанный в интеграции.



14. Сохраните интеграцию с помощью кнопок в правом верхнем углу.

Шаг 2. Подпишите клиентов на Viber™ канал

Для отправки сообщения пользователю в Viber™ требуется:

- Наличие номера телефона пользователя в базе данных;
- Наличие у профиля подписки на Viber™-канал ресурса.

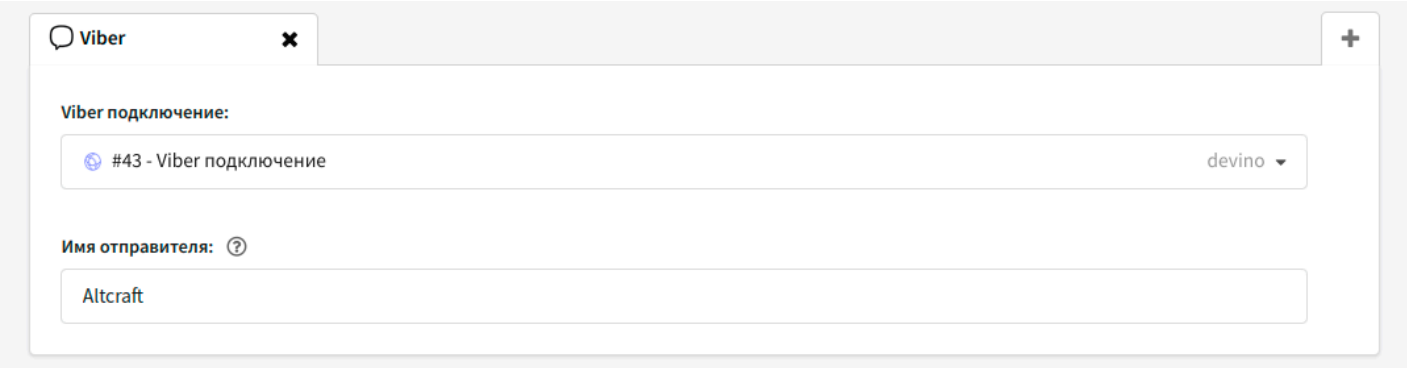
INFO

Номер телефона профиля, на который вы отправляете сообщения, должен быть записан в международном формате. Необходимо использовать следующий формат: `+79001234567`.

Чтобы подписать клиентов на Viber™-канал, выполните следующие действия:

1. Выберите или создайте базу профилей, в которой будут храниться данные подписчиков, в том числе номера телефонов.

2. Создайте новый ресурс или отредактируйте существующий, добавив Viber™-канал. Для настройки канала выберите созданное раннее **Viber™ подключение** и укажите имя отправителя. Это имя увидят подписичики при получении сообщения. **Имя отправителя** можно найти в личном кабинете провайдера; оно должно быть согласовано с вашим провайдером.

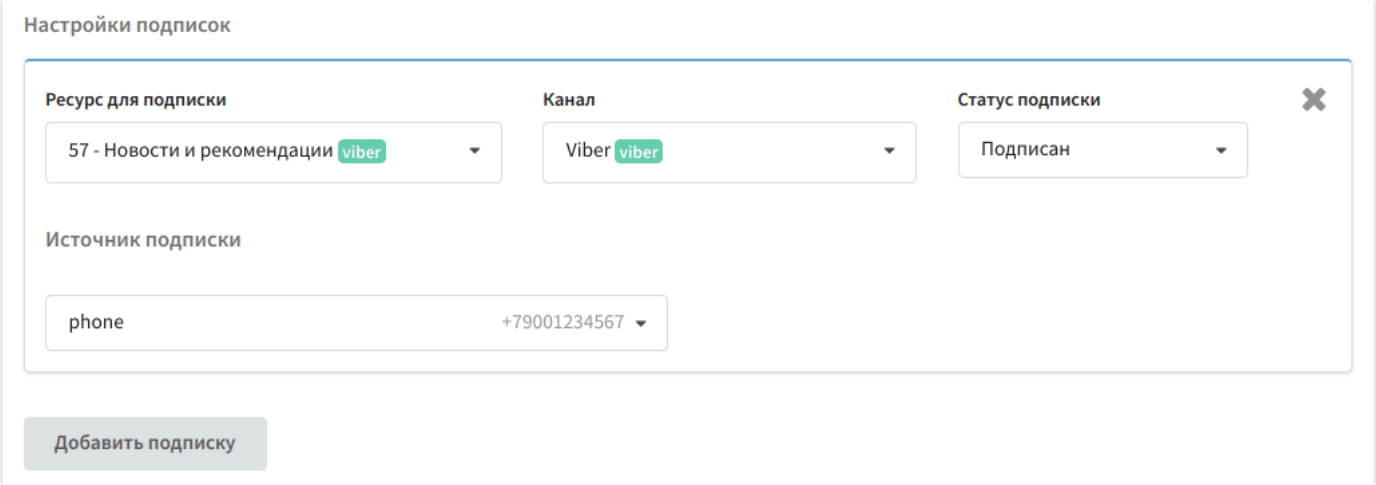


INFO

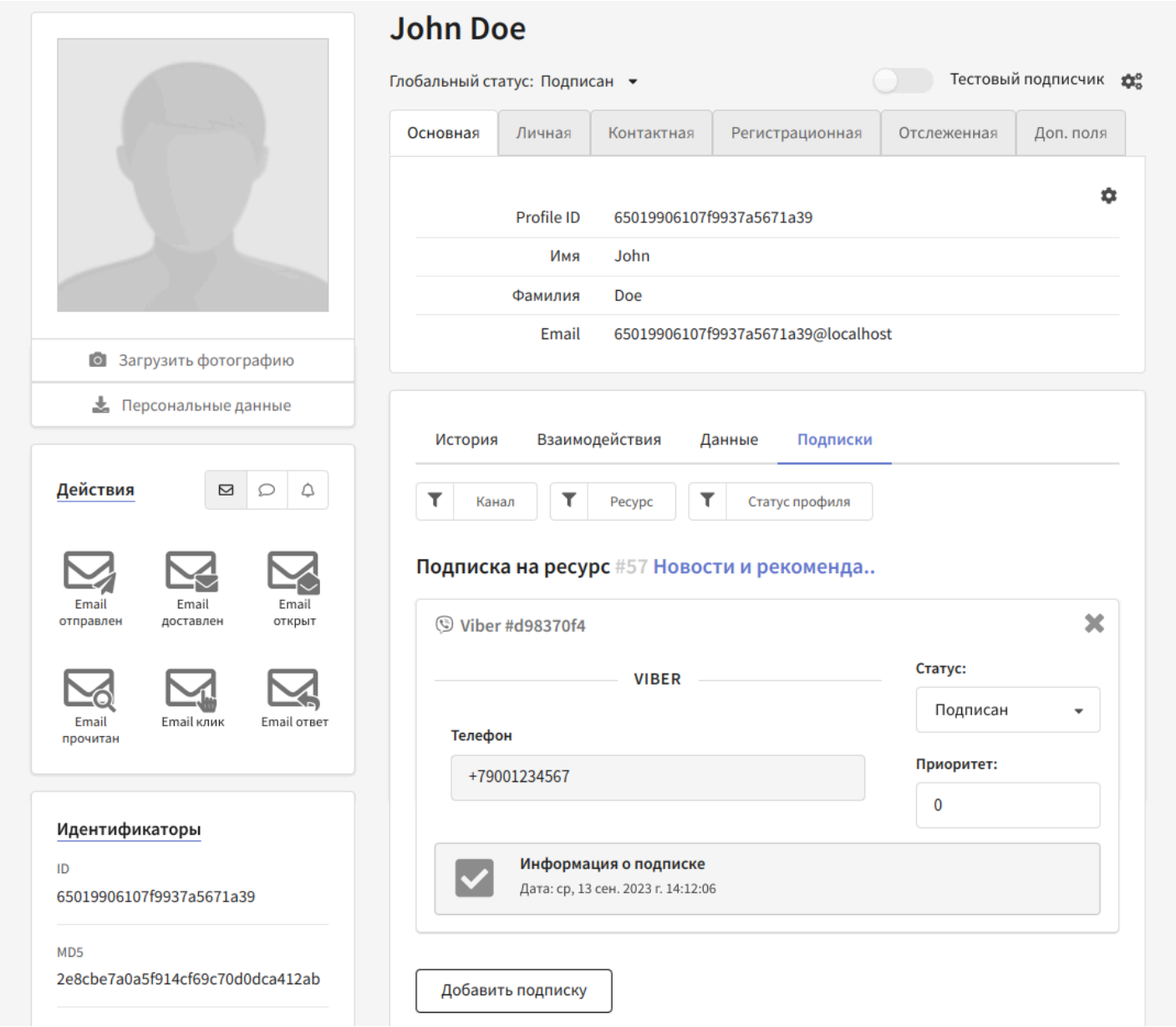
Для настройки коллбэков со статусами сообщений обратитесь к менеджеру компании провайдера или в **техническую поддержку**:

- Техническая поддержка Devino Telecom.
- Техническая поддержка SMS Traffic.
- Техническая поддержка MTC OmniChannel.
- Техническая поддержка Rapporto.

3. Импортируйте данные подписчиков в платформу — например, через файл. При импорте добавьте им подписку на Viber™ канал в ресурсе. В качестве **источника подписки** выберите колонку вашего файла, в которой записаны номера телефонов:



Для каждого клиента будет создан цифровой профиль, объединяющий личные данные, историю действий и подписки на ресурсы. Если при импорте клиенту была успешно добавлена подписка на Viber™-канал, эти данные отобразятся в его истории:



Шаг 3. Создайте шаблон сообщения

1. В главном меню откройте раздел **“Шаблоны”** и создайте новый шаблон с помощью кнопки **“Создать”** на верхней панели.

2. Выберите способ создания — **Редактор** и канал отправки — **Viber™**.

3. Заполните **основные настройки** шаблона.

4. Viber™-шаблон состоит из серии сообщений. Чтобы добавить первое сообщений в серию, нажмите кнопку **+ Сообщения**.

5. Выберите **тип сообщения**:

Тип сообщения	Devino	SMSTraffic	MTC OmniChannel	Rapporto
Текст	+	+	+	+
Текст + кнопка	+	+	+	+
Текст + изображение + кнопка	+	+	+	+

Тип сообщения	Devino	SMSTraffic	MTC OmniChannel	Rapporto
Изображение	+	+	+	-
Видео	+	+	-	-
Видео + текст	+	+	-	-
Видео + текст + кнопка	+	-	-	-
Файл	+	-	-	-

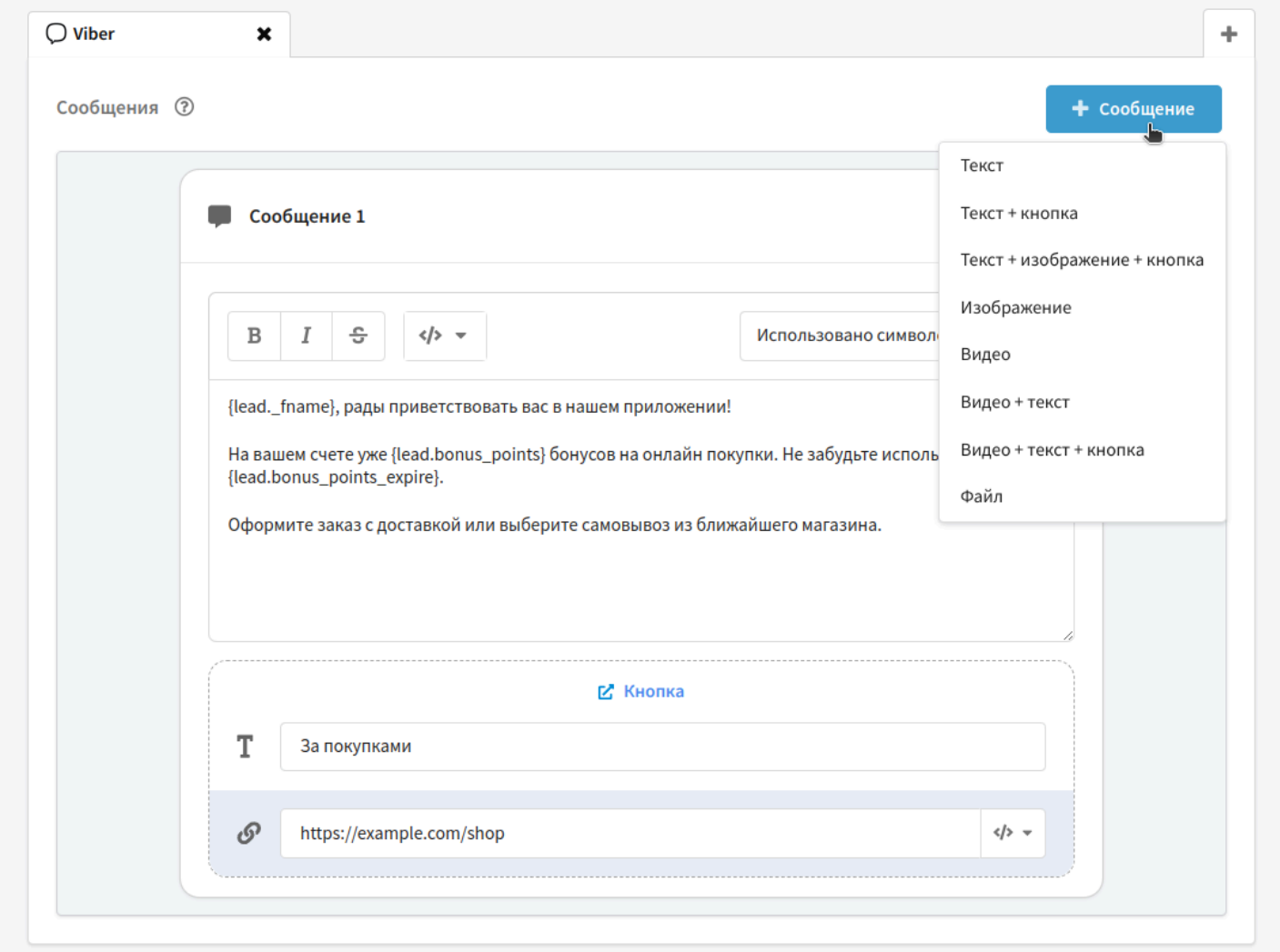
6. Заполните содержание. Для персонализации рассылки используйте в сообщениях переменные

</>

.

INFO

Максимальное количество сообщений в серии — 5. Если в шаблоне используется несколько сообщений, каждое из них будет отправляться с интервалом в 2 секунды.



7. Проверьте, как ваши получатели увидят сообщение. Для этого нажмите на кнопку **предпросмотра** на верхней панели.

8. **Сохраните** шаблон сообщения.

> **TIP**
>
> Больше информации о создании шаблона для Viber™-канала вы найдёте в этой статье.

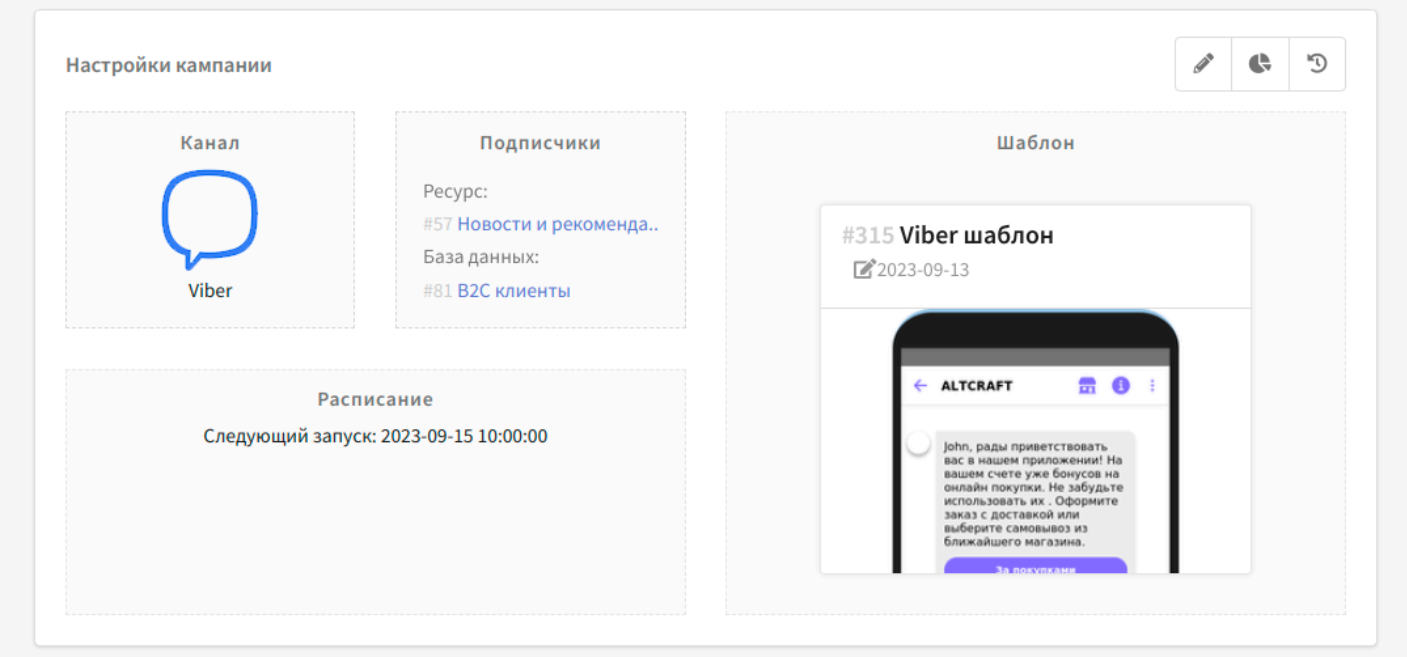
Шаг 4. Запустите рассылку и оцените эффективность рассылки

1. Перейдите в раздел “Рассылки” и выберите тип рассылки:

- Броадкаст — разовая рассылка сообщений.
- Регулярная рассылка — периодическая рассылка.

2. Настройте рассылку:

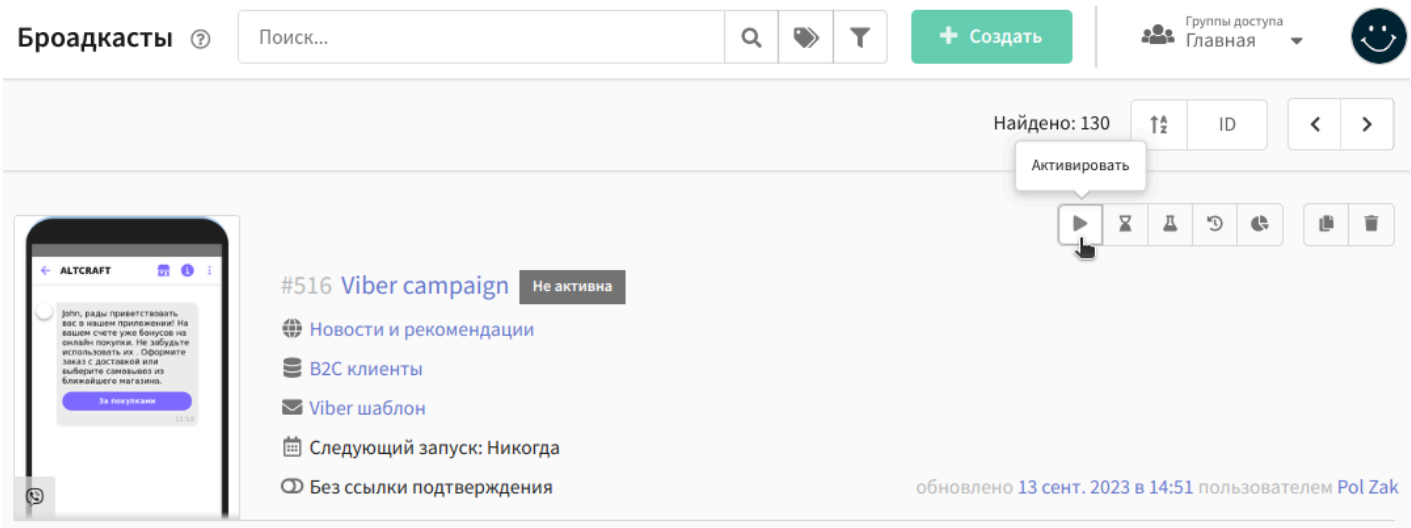
- выберите Viber™ как канал связи,
- выберете настроенный ранее ресурс и аудиторию получателей (база или сегмент),
- добавьте в рассылку шаблон сообщения,
- установите расписание.



3. Сохраните рассылку.

4. Найдите её в списке рассылок выбранного типа и активируйте, кликнув

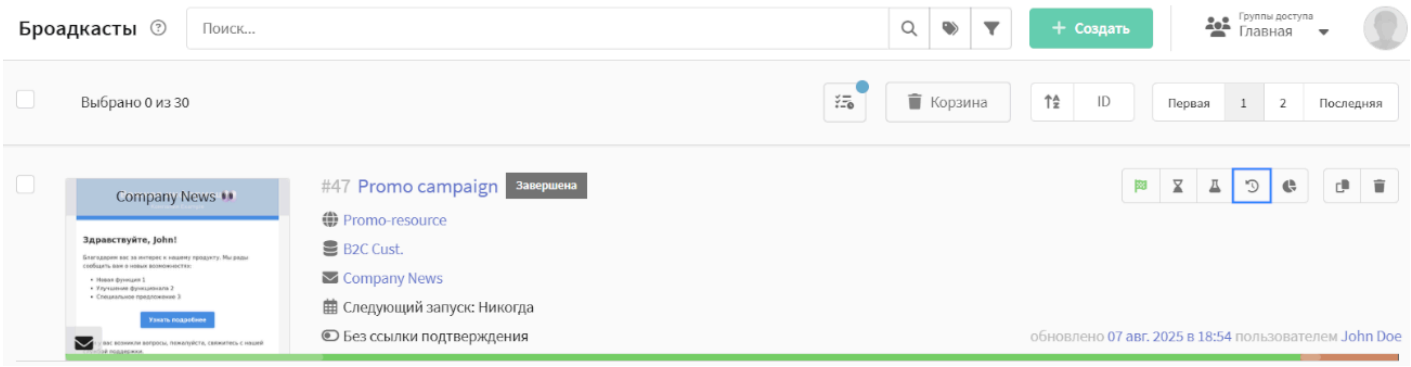
в панели быстрых действий. Активная рассылка начнёт отправлять сообщения, когда наступит время запуска по расписанию.



Журнал рассылок

Журнал рассылок — это инструмент, предоставляющий краткую сводку о ходе выполнения рассылки. В нем фиксируются ключевые события, параметры рассылок и статистика, что позволяет выявить возможные проблемы.

Чтобы открыть журнал рассылок, нажмите на кнопку **“Журнал”** в меню быстрых действий:



Журнал рассылок содержит следующую информацию:

Тип	Описание
Запуск рассылки	Время начала выполнения рассылки
Лимит	Показывает, установлен ли лимит на отправку сообщений
Источник подписчиков	**list** — База данных **segment** — Сегмент (если применимо)
ID базы данных / ID сегмента	Идентификатор используемой базы или сегмента
Время получения данных	Время загрузки контактов из источника (мс)
Количество профилей из источника	Общее число контактов в указанном источнике
Количество валидных подписчиков	Число пользователей с активной подпиской
Количество подписчиков без подписок	Пользователи без подписки на целевой канал (0 в примере)
Количество с некорректными полями	Ошибки в данных подписки (0 в примере)

Тип	Описание
Количество дубликатов	Повторяющиеся контакты (40 в примере)
Количество отправленных сообщений	Фактическое число успешных отправок
Рассылка завершена	Время окончания рассылки
Всего времени затрачено	Общая продолжительность рассылки

Для более точной оценки результатов вашей рассылки воспользуйтесь отчётом по каналам. Чтобы перейти в отчёт, нажмите на кнопку аналитики

в панели быстрых действий. Отчёт предоставит следующую информацию о рассылке:

- количество отправленных, доставленных и недоставленных сообщений;
- количество прочтений и переходов по ссылкам в сообщениях;
- количество отписок и подписок на Viber™-канал.

---

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Notify

Первая Notify-рассылка

INFO

Согласно политике Notify, в данный момент новые доступы к личному кабинету сервиса **не предоставляются.**

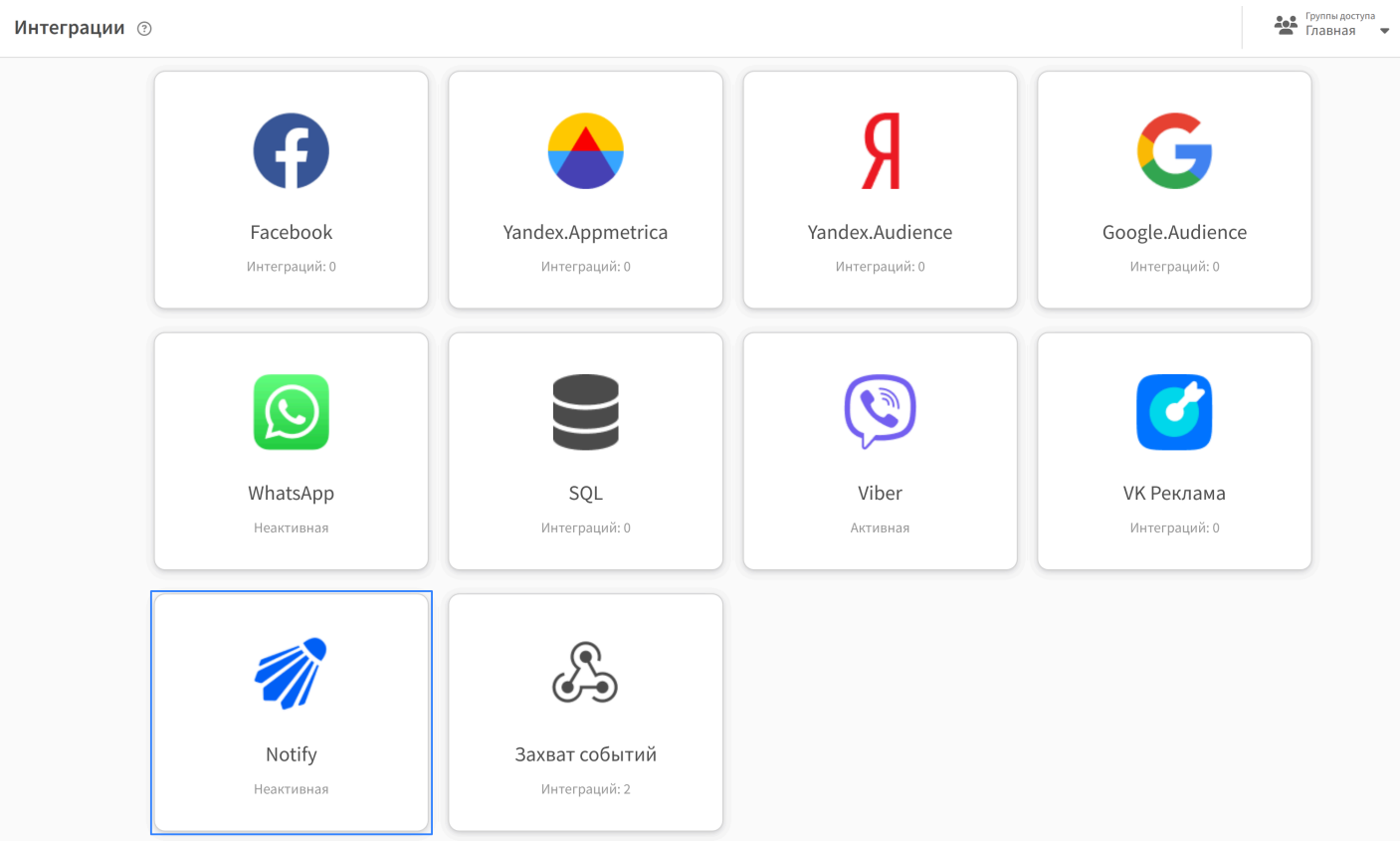
Поддержка текущих пользователей доступна через техподдержку MAX Business.

Notify — это сервис, предоставляющий компаниям возможность отправлять сервисные и транзакционные уведомления пользователям “ВКонтакте” и “Одноклассников”. Уведомления доставляются от официальной группы компании, даже если пользователь не подписан на неё, и могут быть отправлены в виде сервисных сообщений, на которые можно отвечать, или бизнес-уведомлений без возможности ответа.

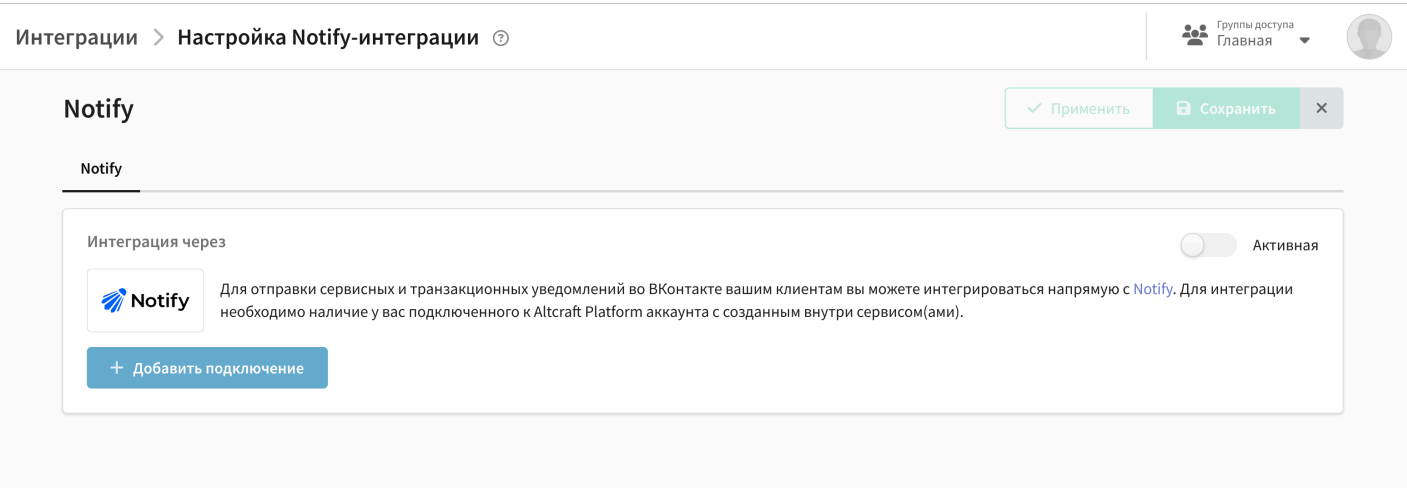
Сервис позволяет значительно сократить расходы на рассылки, обеспечивая высокую скорость доставки и высокий процент прочтения сообщений. Помимо бизнес-уведомлений пользователям “ВКонтакте” и “Одноклассников”, через Notify-канал также возможно отправлять напоминания по SMS.

Шаг 1. Настройте интеграцию

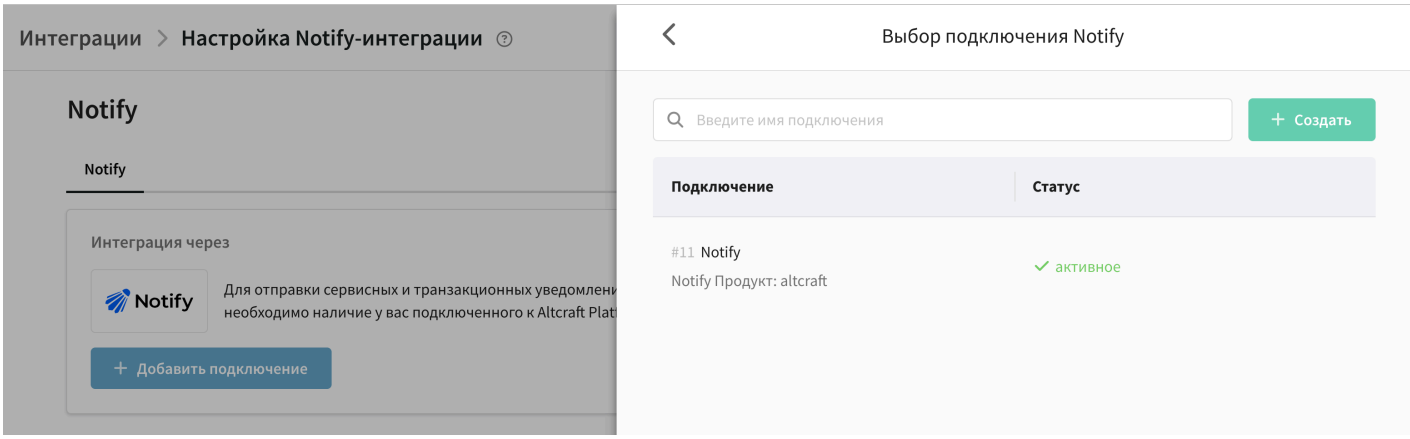
1. Откройте раздел **"Интеграции"** и выберите тип интеграции — **Notify**:



2. Нажмите **“Добавить подключение”**:



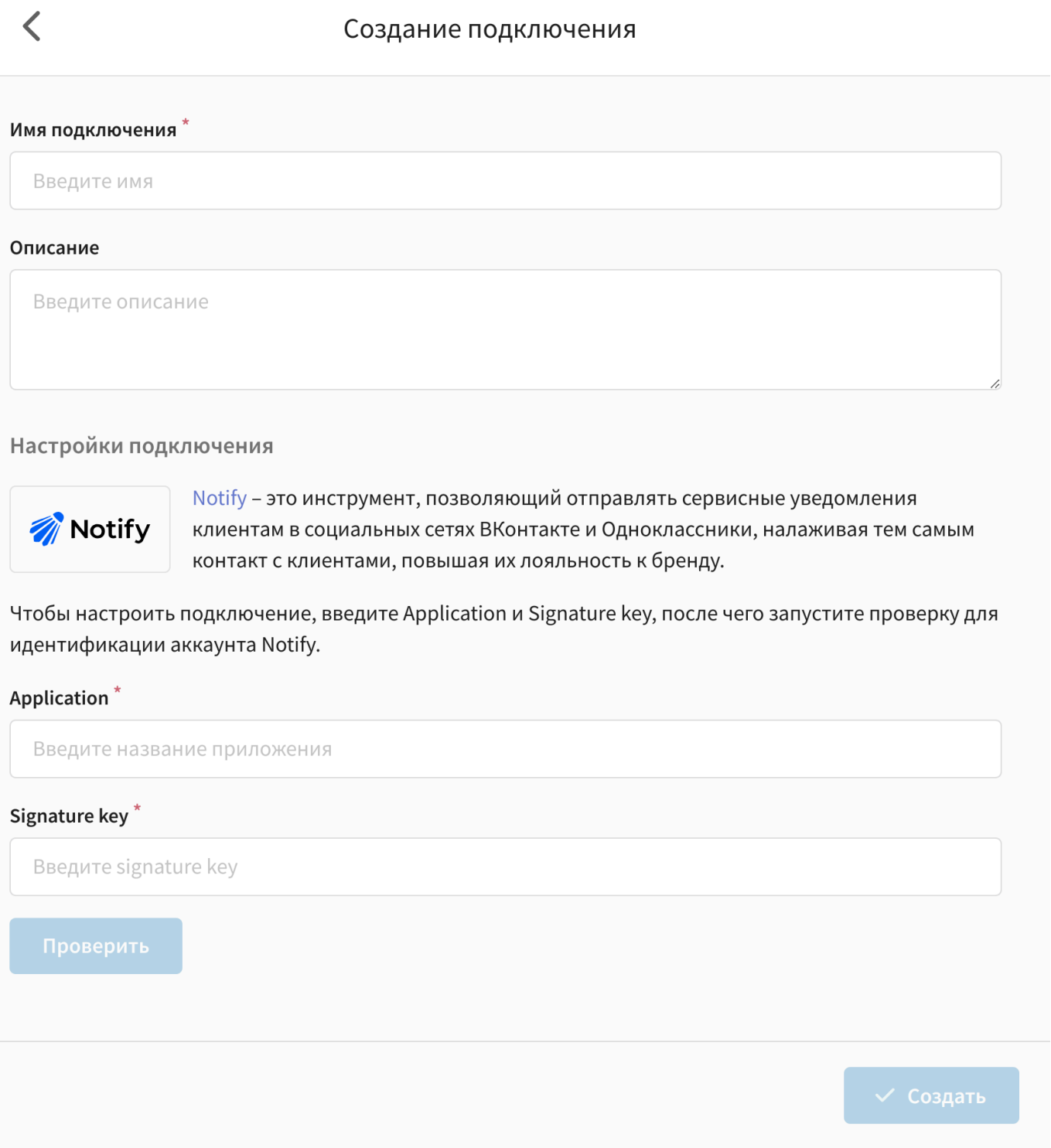
3. Выберите доступное **подключение** к Notify:



Нет доступных подключений

Если вы ещё не настраивали подключение к Notify, вы можете создать его непосредственно в этом окне.

Нажмите **“Создать”**. Введите имя и описание подключения. Далее укажите параметры **Application** и **Signature key** и нажмите кнопку **“Проверить”**, чтобы проверить правильность введенных данных. Если данные прошли проверку, нажмите кнопку **“Создать”** в правом нижнем углу:



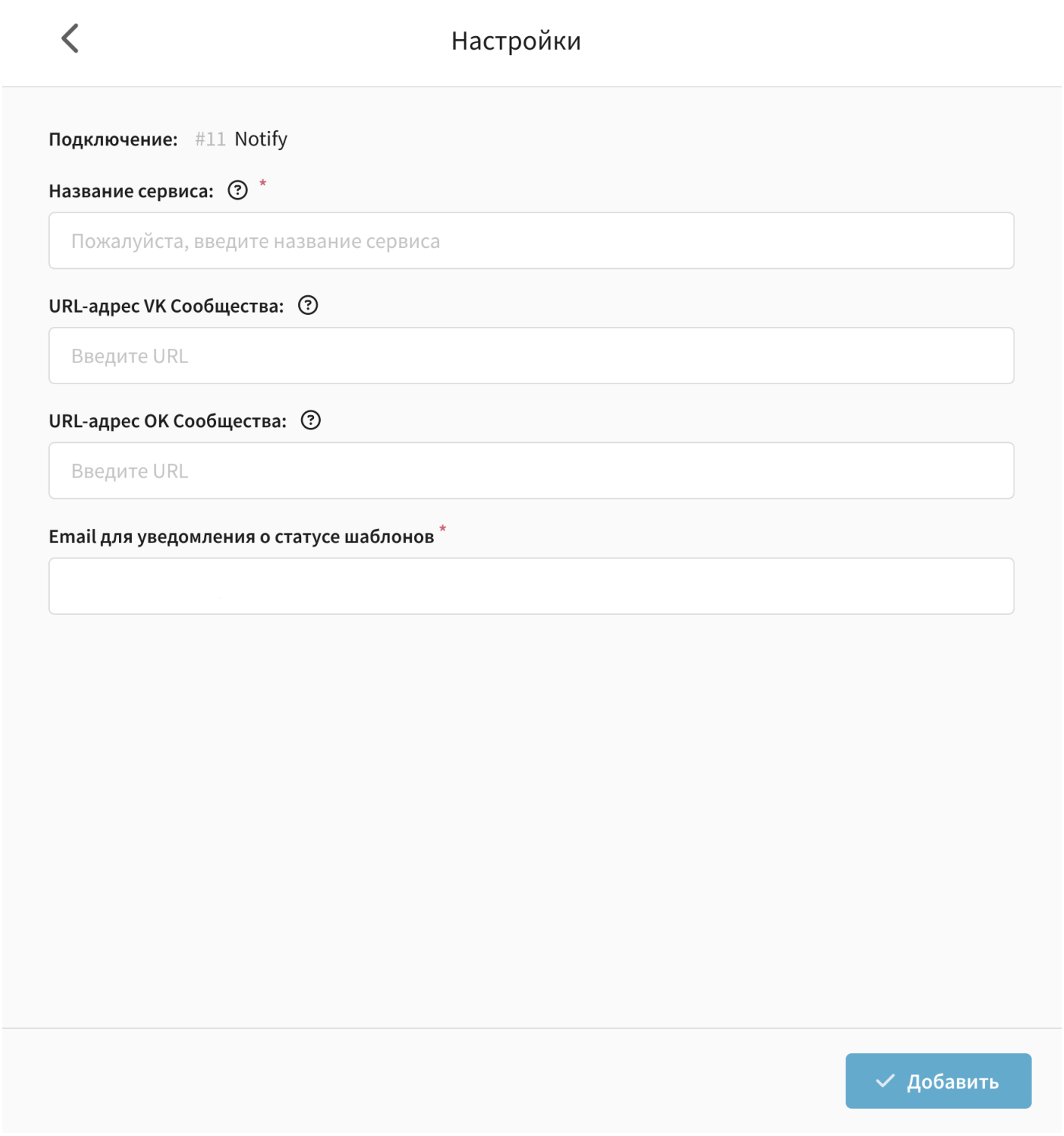
TIP

Подробнее о создании подключения можно прочитать **здесь**.

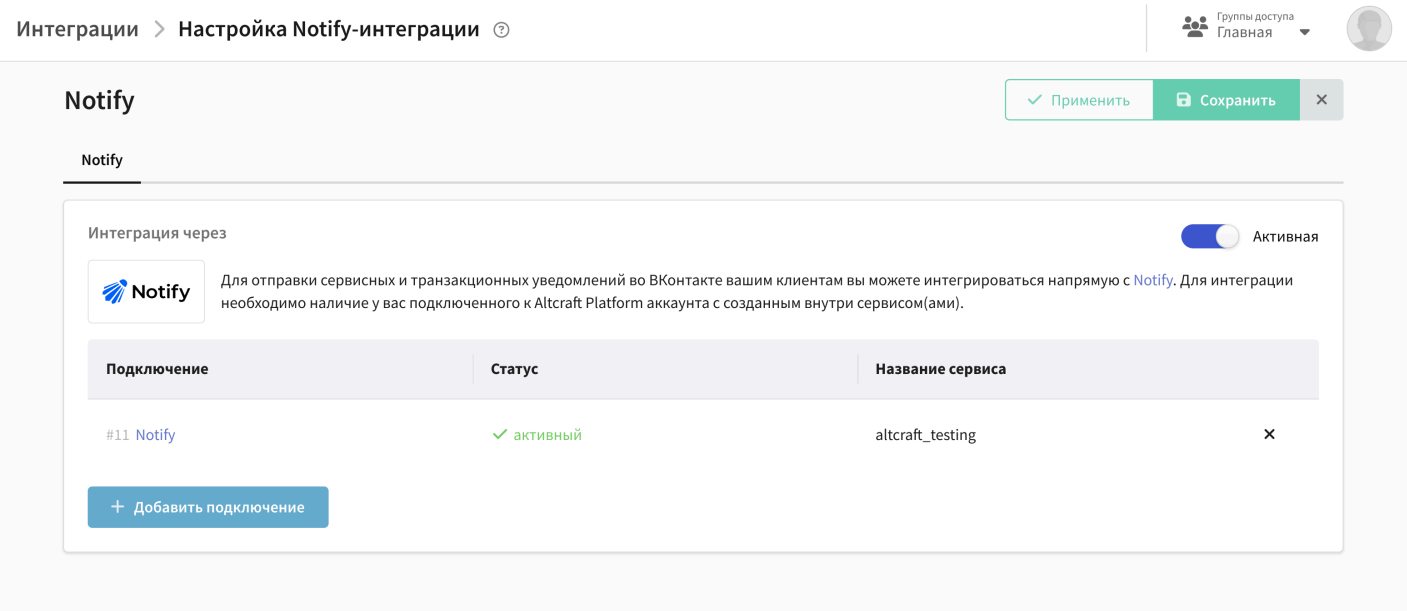
4. Далее укажите **параметры интеграции**, для корректной работы они должны **точно совпадать** с соответствующими параметрами из личного кабинета Notify:

- **Название сервиса** — название сервиса для рассылки, обязательное поле.
- **URL-адрес VK Сообщества** — ссылка на сообщество Вконтакте, от имени которого будет происходить рассылка.
- **URL-адрес OK Сообщества** — ссылка на сообщество в Одноклассники, от имени которого будет происходить рассылка.

- **Email для уведомления о статусе шаблонов** — электронный почтовый адрес, на который будут приходить отчеты о статусах шаблонов. Обязательное поле.



5. После заполнения полей нажмите кнопку **“Добавить”**. Если введенные данные некорректны, платформа попросит проверить их правильность. Если же данные верны, то параметры сохранятся.
6. **Активируйте** интеграцию с помощью переключателя и **сохраните** её с помощью кнопок в правом верхнем углу.



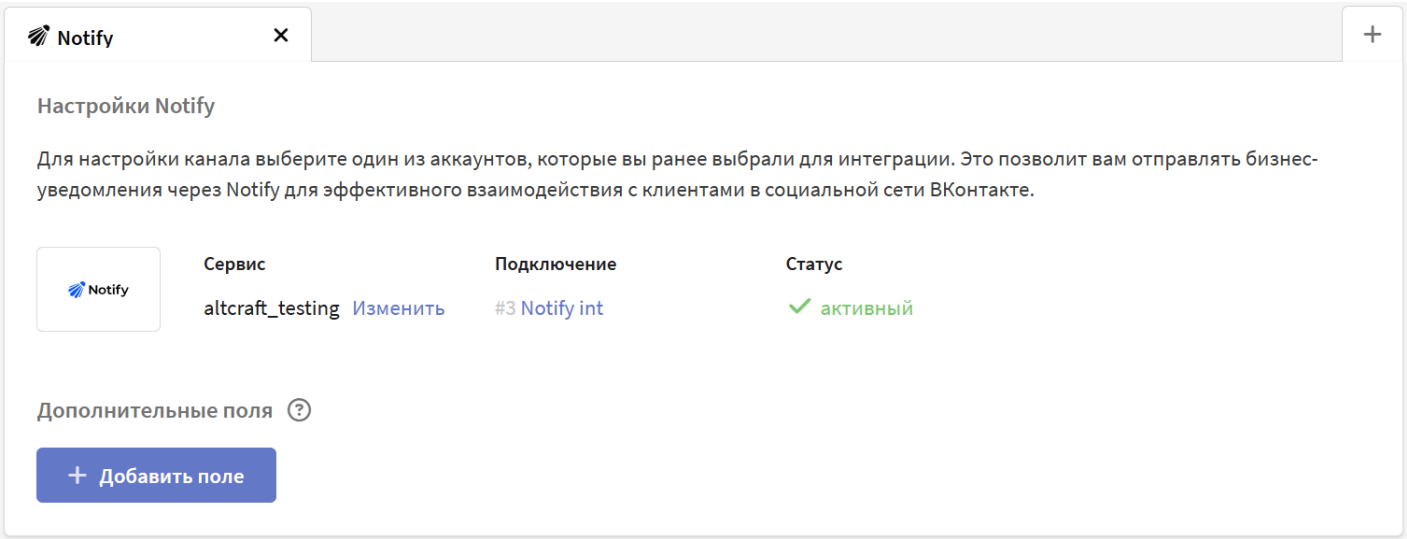
После настройки интеграции вы можете использовать её для добавления Notify-канала в **ресурс**.

Шаг 2. Подпишите клиентов на Notify канал

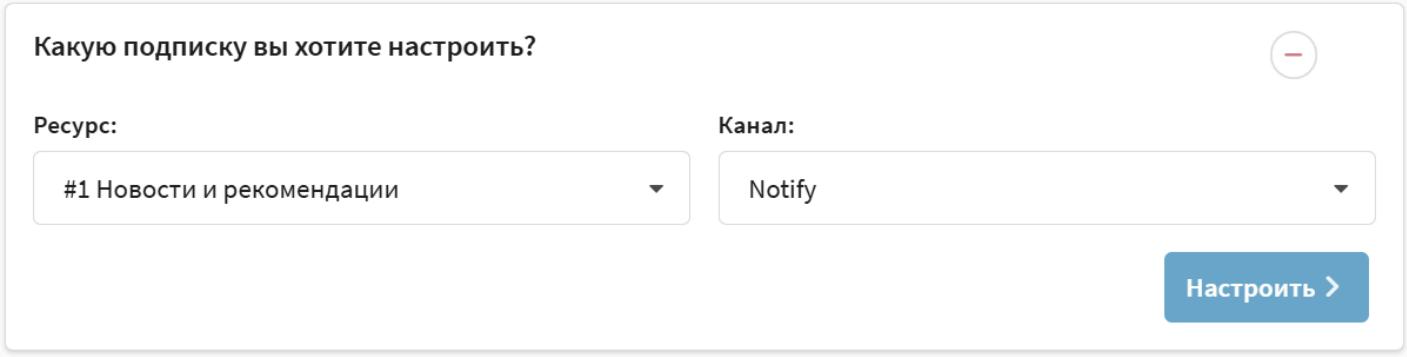
Чтобы подписать клиентов на Notify-канал, выполните следующие действия:

1. Выберите или создайте базу профилей, в которой будут храниться данные подписчиков.

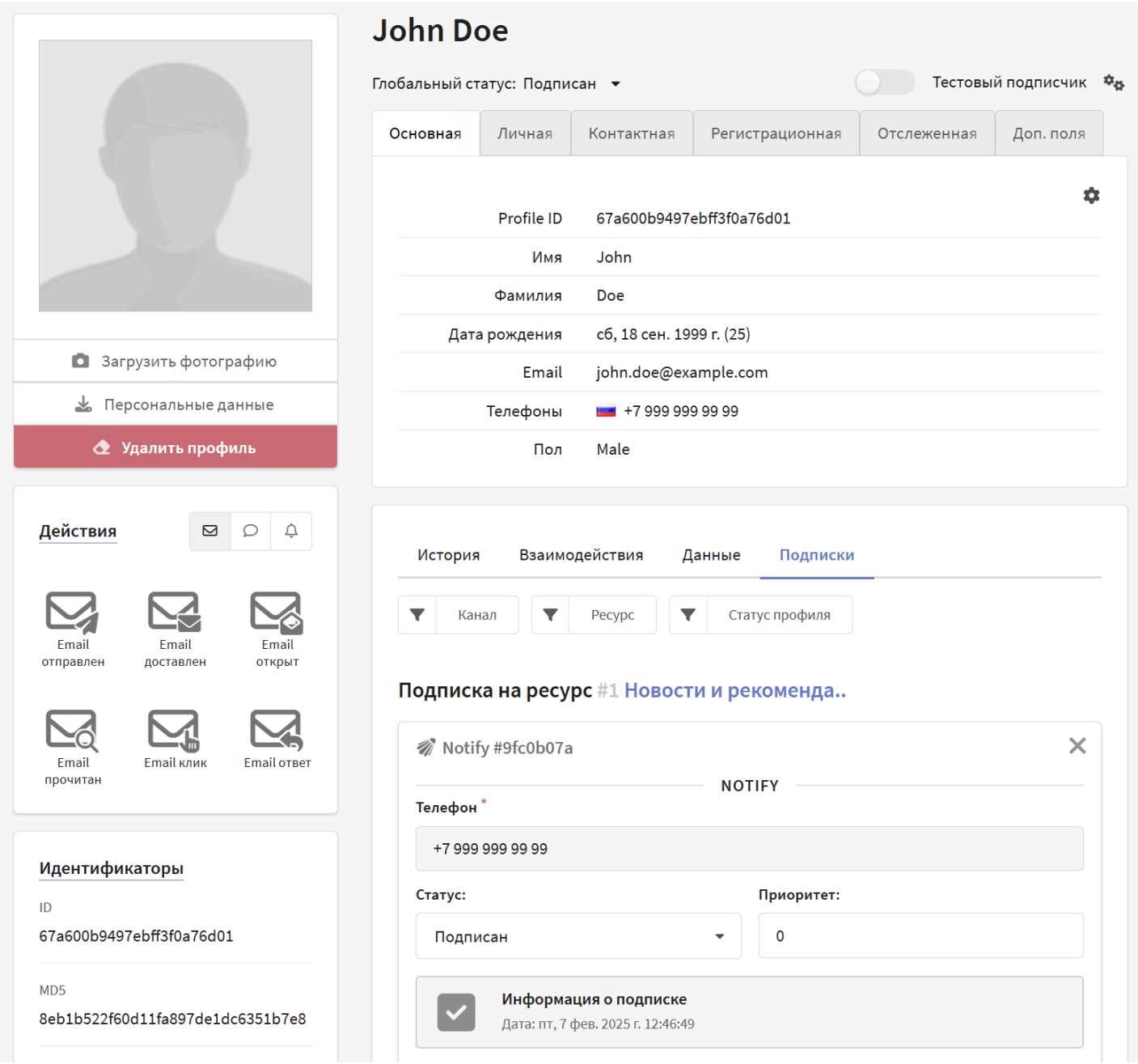
2. Создайте новый ресурс или отредактируйте существующий, добавив Notify-канал. Для настройки канала выберите один из аккаунтов, которые вы ранее выбрали для интеграции.



3. Импортируйте данные подписчиков в платформу — например, через файл. При импорте добавьте им подписку на Notify-канал в ресурсе. В данном случае номер телефона является источником подписки.



Для каждого клиента будет создан [цифровой профиль](), объединяющий личные данные, историю действий и подписки на ресурсы. Если при импорте клиенту была успешно добавлена подписка на Notify-канал, эти данные отобразятся в его истории:



Шаг 3. Создайте шаблон сообщения

INFO

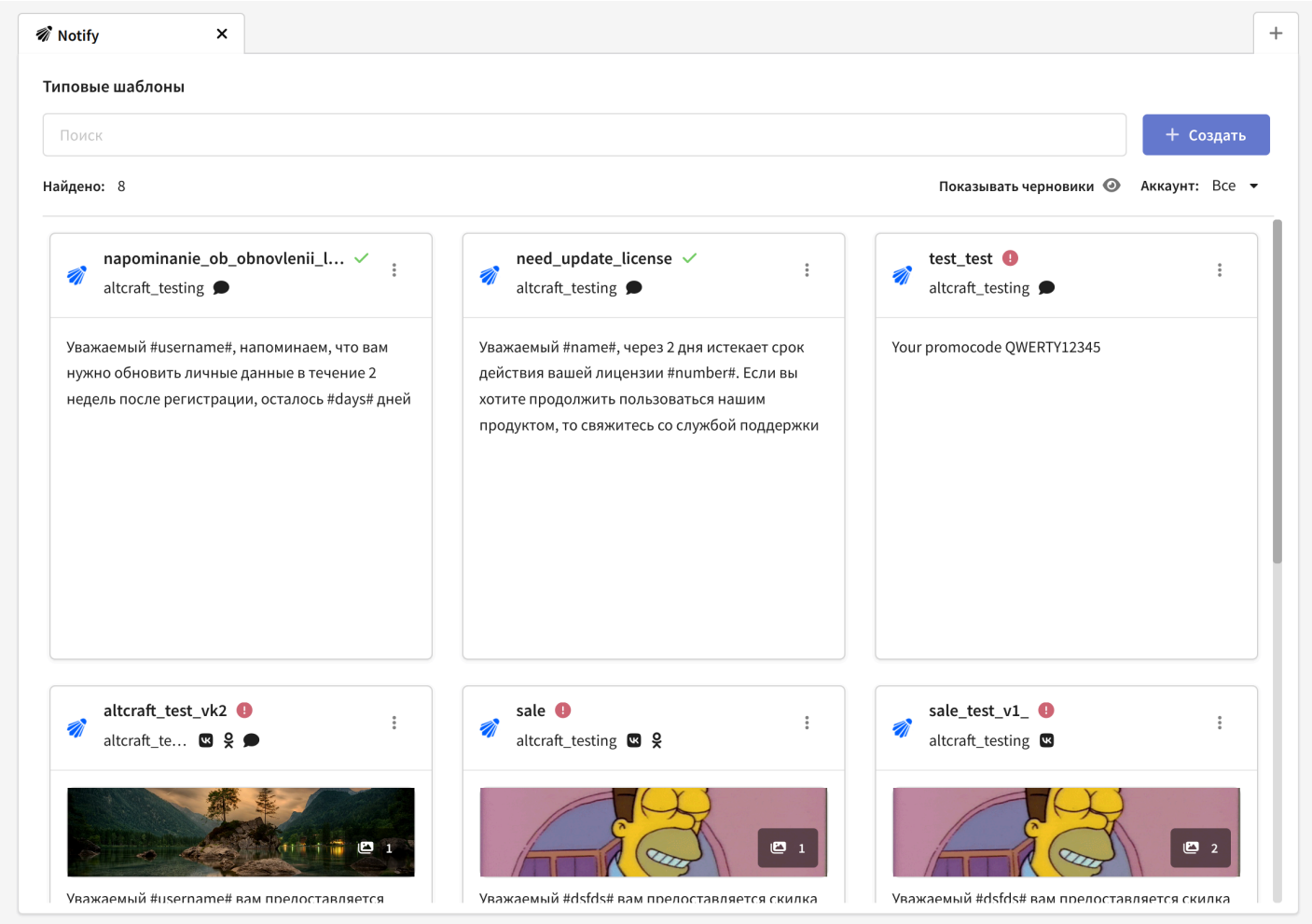
Для отправки сообщений могут использоваться только шаблоны, прошедшие модерацию в системе Notify.

1. В главном меню откройте раздел **“Шаблоны”** и создайте новый шаблон с помощью кнопки **“Создать”** на верхней панели.

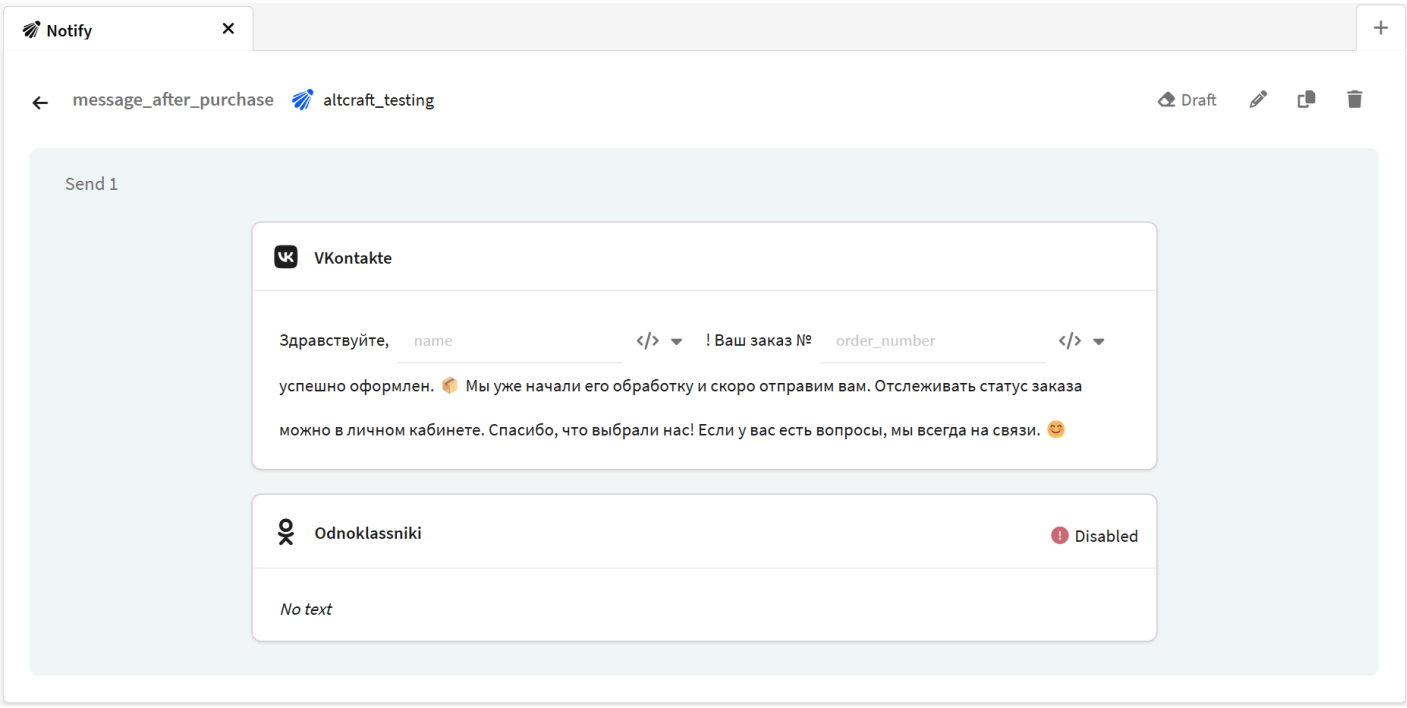
2. Выберите способ создания — **Редактор** и канал отправки — **Notify**.

3. Заполните **основные настройки** шаблона.

4. При открытии редактора сообщений для Notify вам будут показаны типовые шаблоны, созданные в вашем личном кабинете Notify. Выберите нужный вам шаблон.



5. Заполните **параметры** внутри шаблона. **Параметры** — текстовые блоки, которые меняются для каждого нового сообщения (имя, адрес, название мероприятия, дата и время, ссылка и т.д.). Вы можете вручную прописать значения параметров или добавить переменные Altcraft.



6. Проверьте, как ваши получатели увидят сообщение. Для этого нажмите на кнопку **предпросмотра** на верхней панели.

7. **Сохраните** шаблон сообщения.

TIP

Вы можете также создать типовой шаблон Notify, находясь в платформе. Для создания шаблона нажмите соответствующую кнопку. Откроется окно создания шаблона, в этом окне необходимо указать имя типового шаблона и настроенный в личном кабинете сервис. Больше информации о создании шаблона для Notify-канала вы найдёте в этой статье.

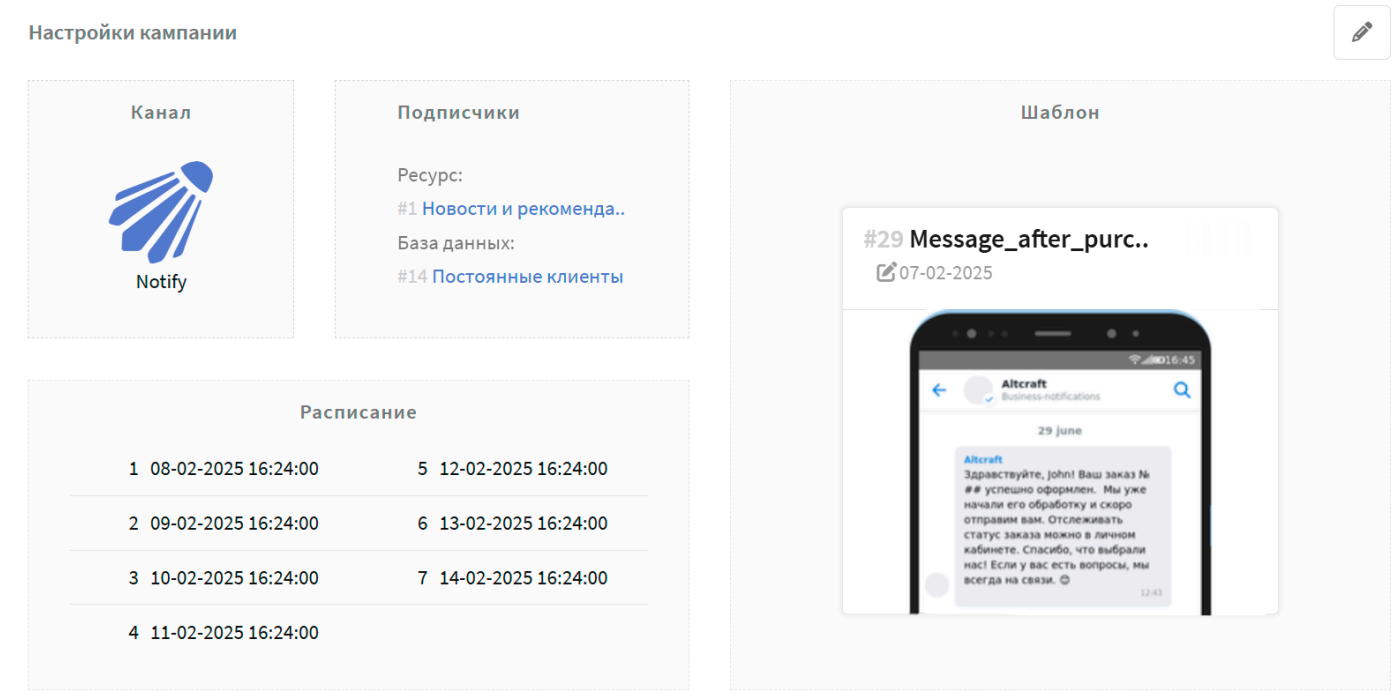
Шаг 4. Запустите рассылку и оцените эффективность рассылки

1. Перейдите в раздел “Рассылки” и выберите тип рассылки:

- Броадкаст — разовая рассылка сообщений.
- Регулярная рассылка — периодическая рассылка.

2. Настройте рассылку:

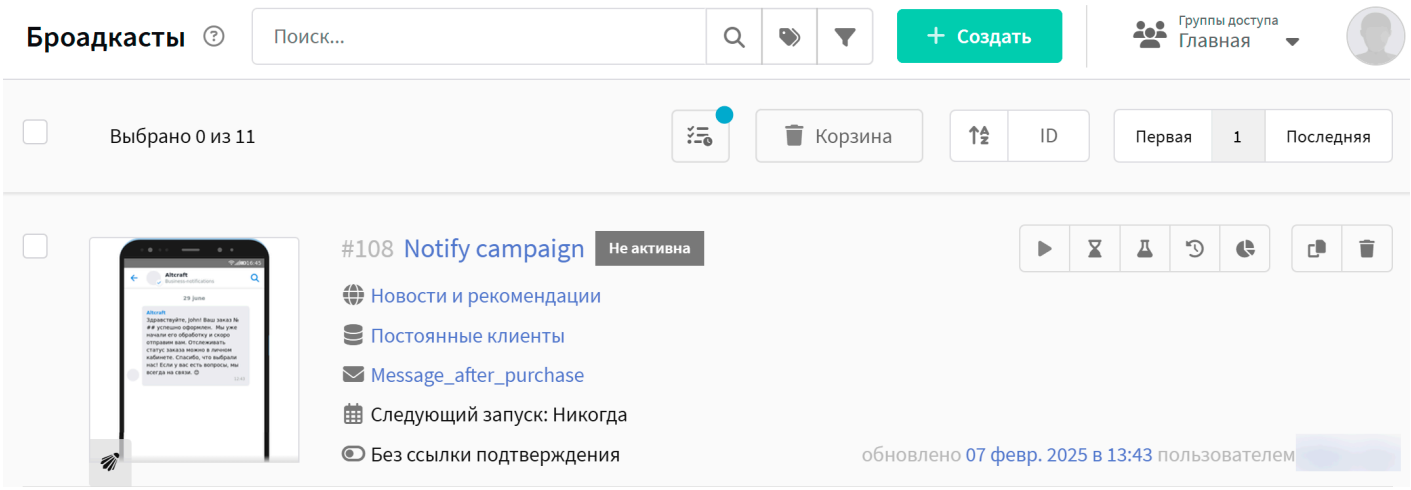
- выберите Notify как канал связи,
- выберете настроенный ранее ресурс и аудиторию получателей (база или сегмент),
- добавьте в рассылку шаблон сообщения,
- установите расписание.



3. Сохраните рассылку.

4. Найдите её в списке рассылок выбранного типа и активируйте, кликнув

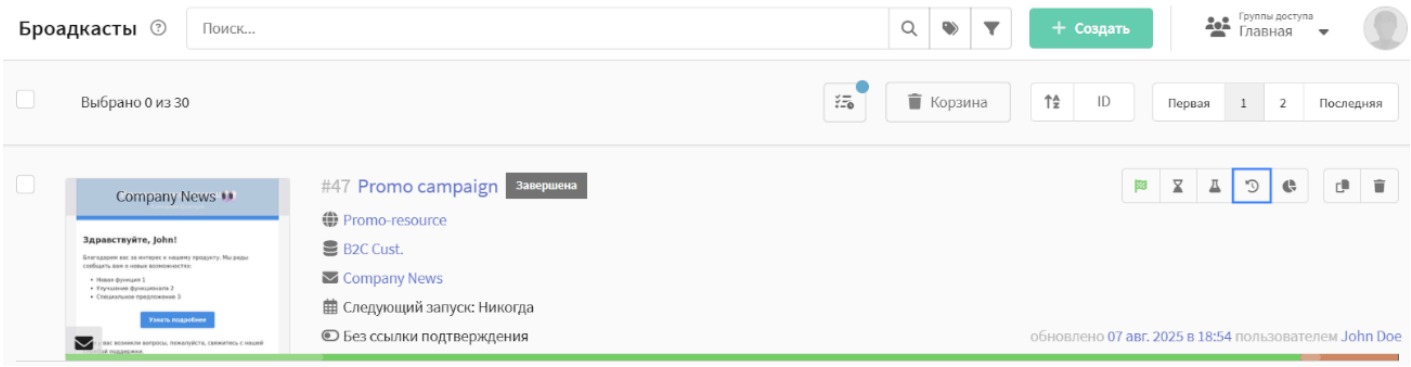
в панели быстрых действий. Активная рассылка начнёт отправлять сообщения, когда наступит время запуска по расписанию.



Журнал рассылки

Журнал рассылки — это инструмент, предоставляющий краткую сводку о ходе выполнения рассылки. В нем фиксируются ключевые события, параметры рассылки и статистика, что позволяет выявить возможные проблемы.

Чтобы открыть журнал рассылки, нажмите на кнопку **“Журнал”** в меню быстрых действий:



Журнал рассылки содержит следующую информацию:

Тип	Описание
Запуск рассылки	Время начала выполнения рассылки
Лимит	Показывает, установлен ли лимит на отправку сообщений
Источник подписчиков	**list** — База данных **segment** — Сегмент (если применимо)
ID базы данных / ID сегмента	Идентификатор используемой базы или сегмента
Время получения данных	Время загрузки контактов из источника (мс)
Количество профилей из источника	Общее число контактов в указанном источнике
Количество валидных подписчиков	Число пользователей с активной подпиской
Количество подписчиков без подписок	Пользователи без подписки на целевой канал (0 в примере)
Количество с некорректными полями	Ошибки в данных подписки (0 в примере)
Количество дубликатов	Повторяющиеся контакты (40 в примере)
Количество отправленных сообщений	Фактическое число успешных отправок
Рассылка завершена	Время окончания рассылки
Всего времени затрачено	Общая продолжительность рассылок

Для более точной оценки результатов вашей рассылки воспользуйтесь отчётом по каналам. Чтобы перейти в отчёт, нажмите на кнопку аналитики

в панели быстрых действий. Отчёт предоставит следующую информацию о рассылке:

- количество отправленных, доставленных и недоставленных сообщений;
- количество прочтений и переходов по ссылкам в сообщениях;
- количество отписок и подписок на Notify-канал.

Telegram Bot

Первая рассылка через Telegram Bot

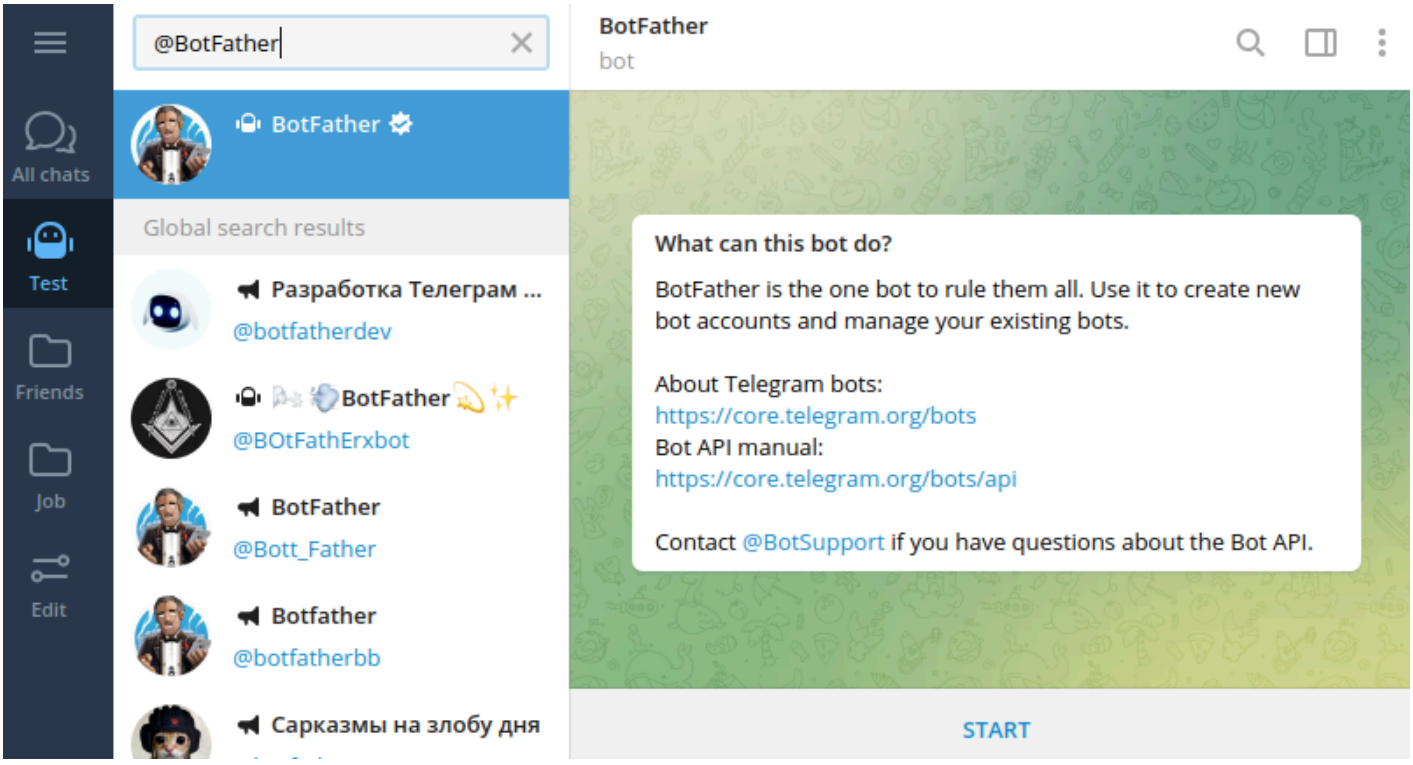
Telegram-бот — это директ-канал, позволяющий отправлять персональные сообщения подписчикам, которые вступили в чат с ботом.

Как происходит подписка? Пользователь стартует бота. Вы извлекаете его chat_id, через API передаёте в платформу и подписываете профиль на канал в ресурсе. После этого можно отправлять рассылки. Обратите внимание, платформа не получает chat_id автоматически. Вам потребуется настроить бота так, чтобы он самостоятельно собирал данные пользователя и передавал их в платформу.

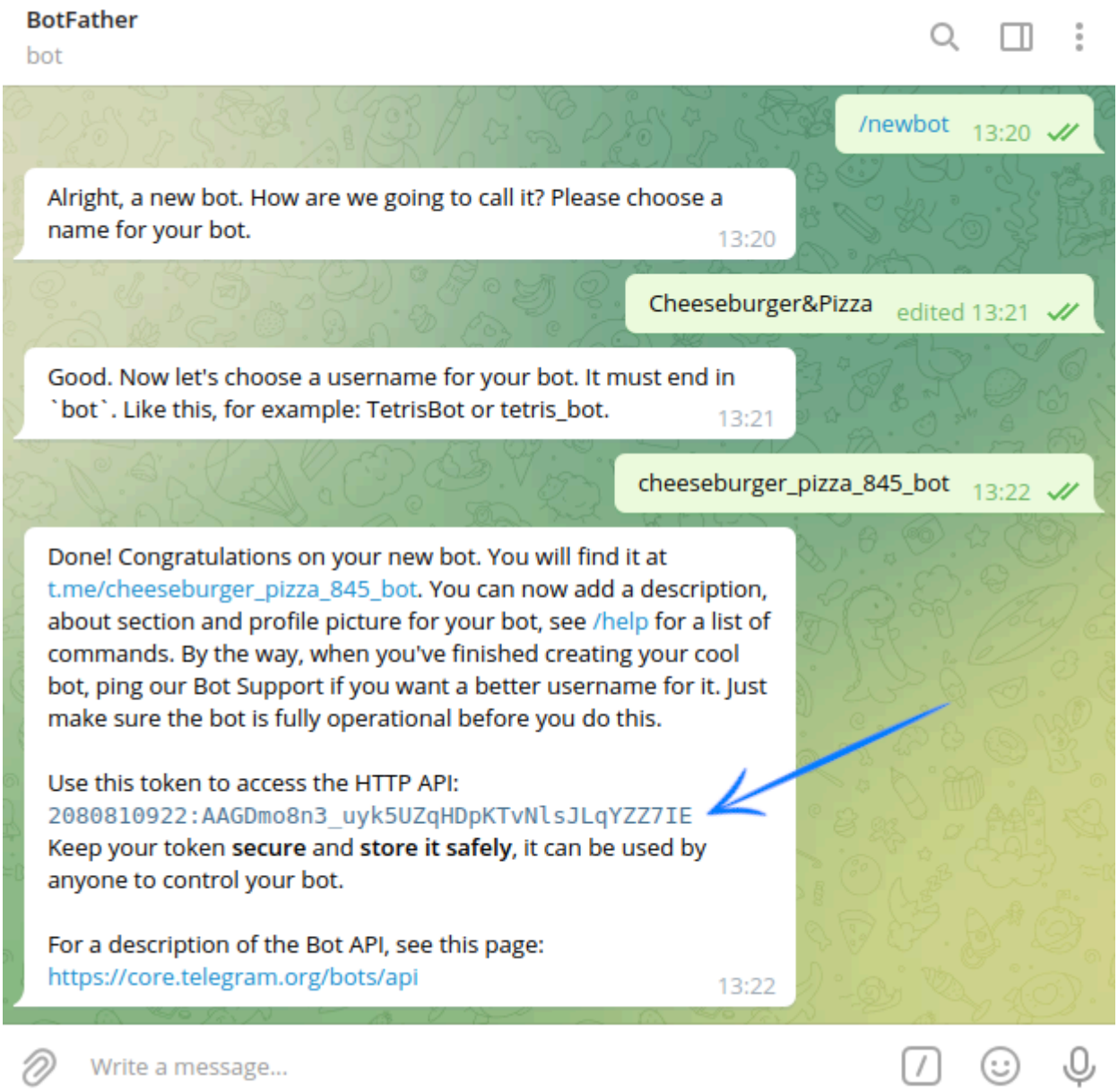
Ниже пошагово описано, как организовать рассылку с помощью Telegram-бота в платформе Altcraft.

Шаг 1. Создайте бот и получите токен

1. Зайдите в Telegram, введите **@BotFather** в окне поиска и выберете нужного бота:



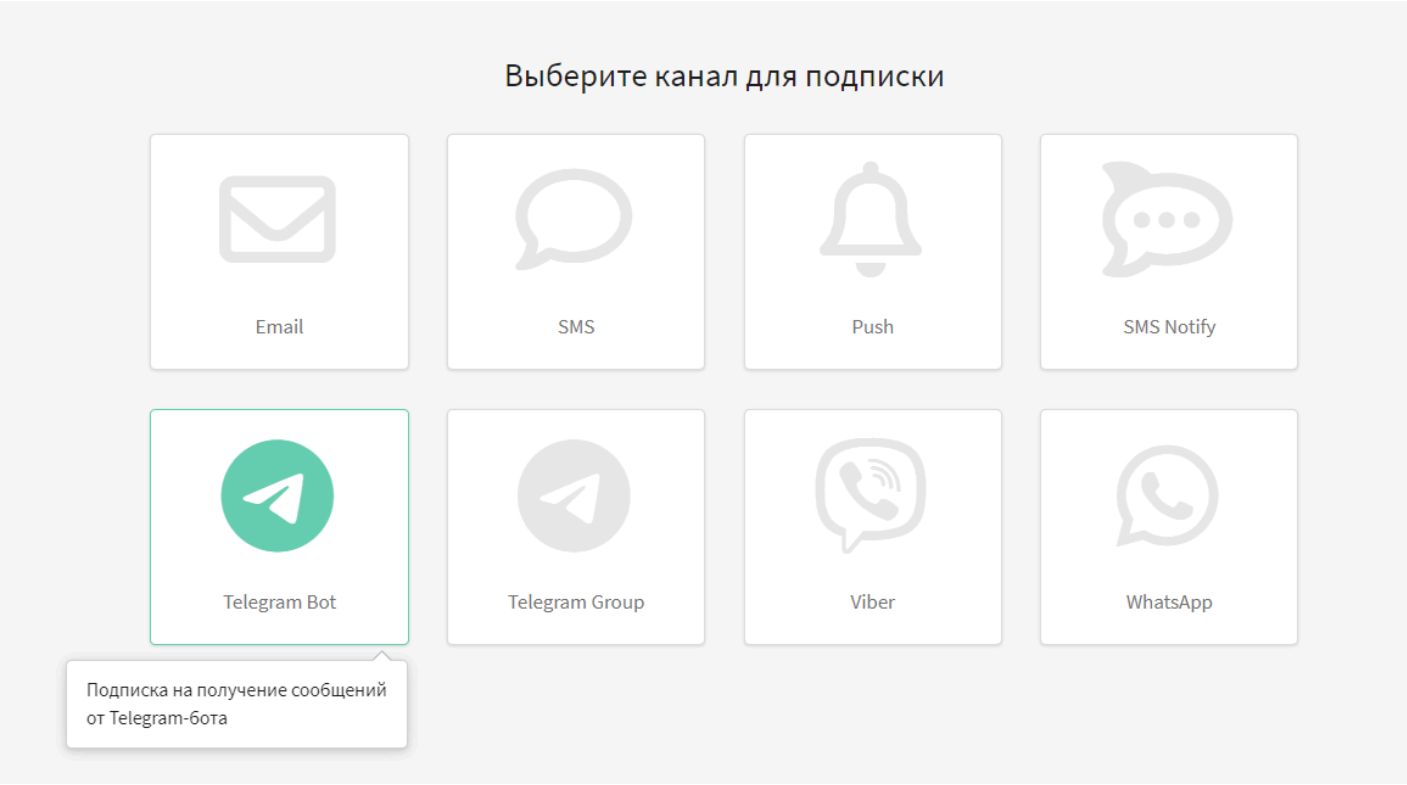
2. Начните диалог с ботом, нажав кнопку **START** в чате. Затем отправьте боту команду `/newbot`. Введите сначала имя, а затем username для нового бота. Обратите внимание, username должен оканчиваться на “**bot**” или “**Bot**”. Скопируйте полученный токен бота:



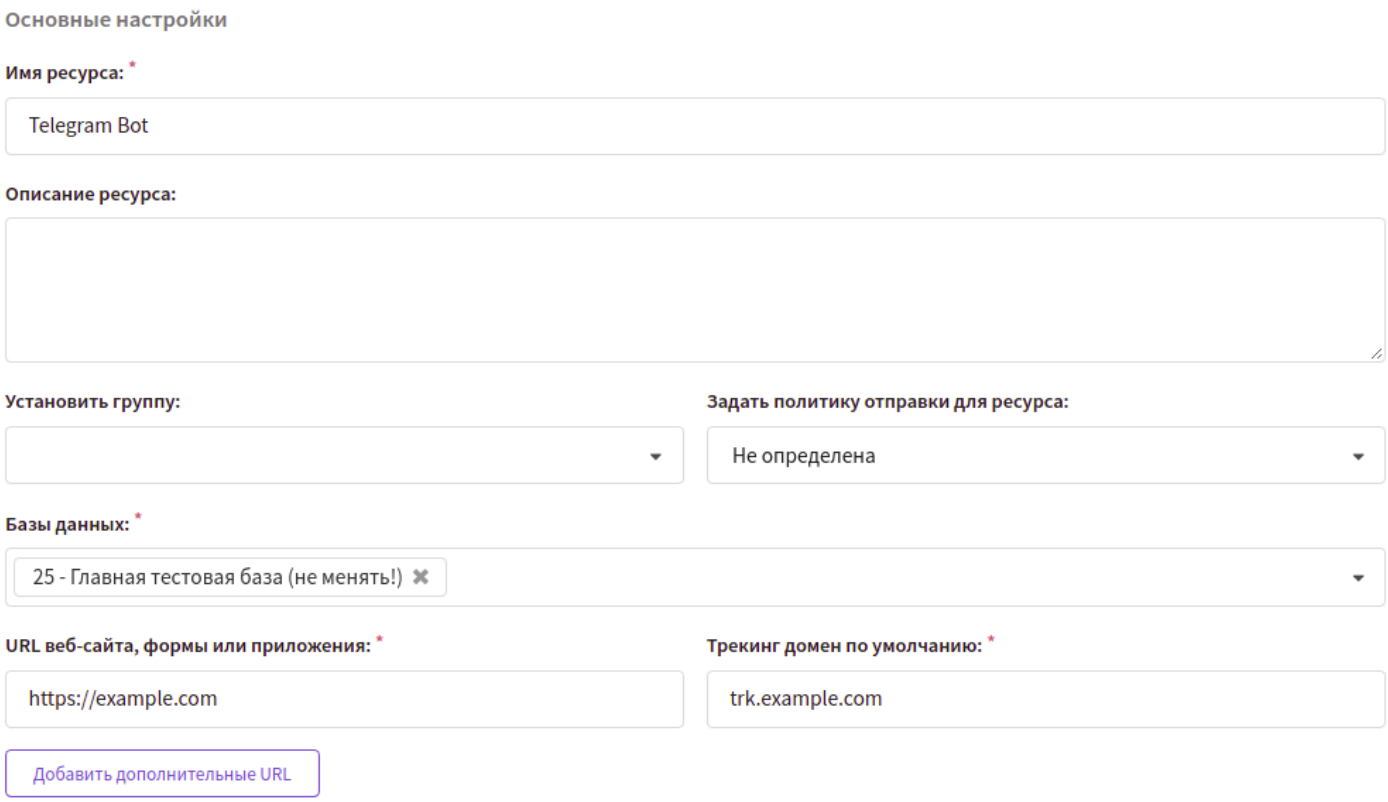
Шаг 2. Подпишите клиентов на Telegram Bot

Перейдите в раздел “**Данные**” —> “**База профилей**” и выберите базу, в которую будете передавать данные пользователей, подписанных на Telegram-бота.

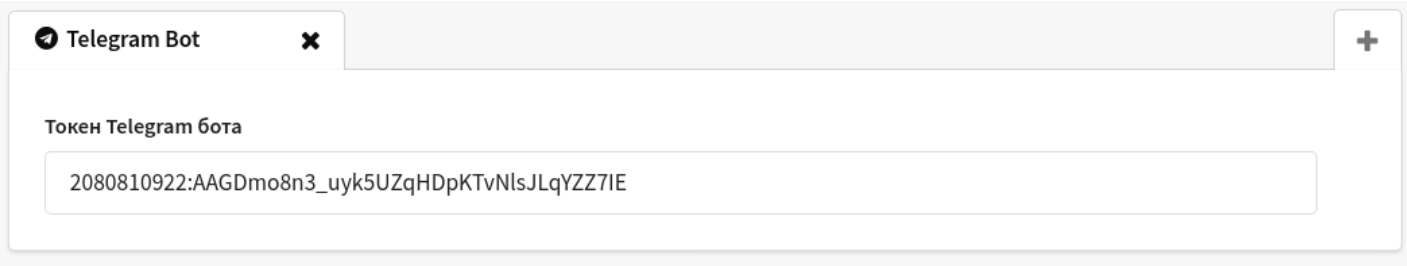
Затем в разделе “**Ресурсы**” создайте новый ресурс или выберите существующий. В качестве канала используйте **Telegram Bot**.



В основных настройках добавьте необходимые базы данных. Укажите **трекинг домен по умолчанию**, который будет обрабатывать переходы подписчиков по ссылкам в сообщениях Telegram бота:



Добавьте скопированный из Telegram токен бота и сохраните ресурс.



Теперь необходимо импортировать данные пользователей в базу, подписав их на соответствующий ресурс. В подиску необходимо передать chat_id пользователя.

NOTE

Обратите внимание, со стороны платформы происходит только отправка рассылок. Сбор и передача в платформу данных о пользователях, которые подписались на Telegram-бота, в том числе передача chat_id осуществляется на стороне клиента. Подробнее о том, как это сделать, читайте в документации Telegram.

Пример получения chat_id клиента с помощью метода getUpdates

Метод: POST

URL-адрес:

`https://api.telegram.org/bot<bot_token>/getUpdates`

Ответ:

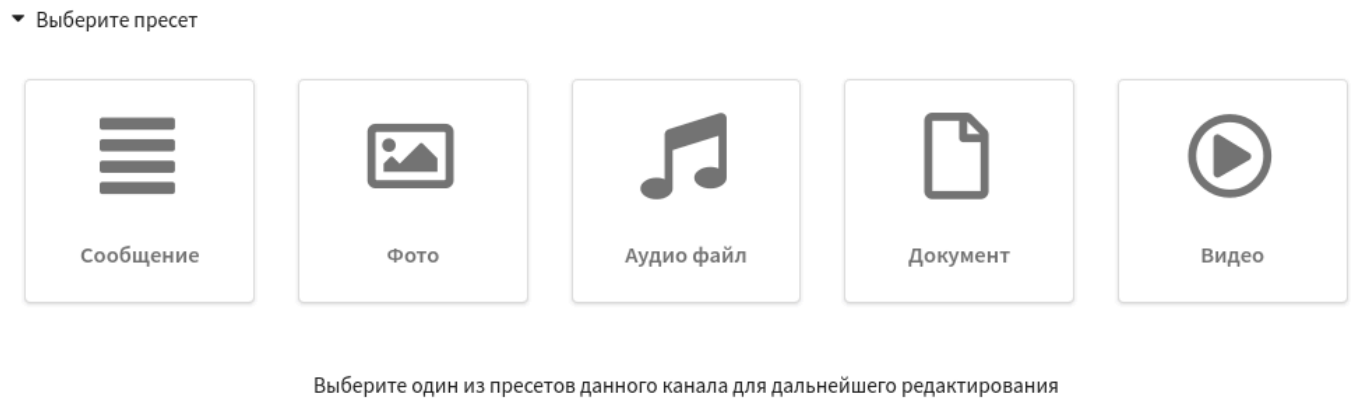
```
{
    "ok": true,
    "result": [
        {
            "update_id": 262756455,
            "message": {
                "message_id": 47,
                "from": {
                    "id": <chat_id>,
                    "is_bot": false,
                    "first_name": "John",
                    "last_name": "Doe",
                    "username": "john_doe",
                    "language_code": "en"
                },
                "chat": {
                    "id": <chat_id>,
                    "first_name": "John",
                    "last_name": "Doe",
                    "username": "john_doe",
                    "type": "private"
                },
                "date": 1652284613,
                "text": "Hello, world"
            }
        }
    ]
}
```

`<chat_id>` — это идентификатор чата с конкретным пользователем, который необходимо передать в платформу.

Шаг 3. Создайте шаблон

Перейдите в раздел “**Шаблоны**”. При создании нового шаблона выберите **Telegram Bot** канал или добавьте этот канал для уже существующего шаблона.

В **Telegram** версии шаблона необходимо выбрать тип контента:



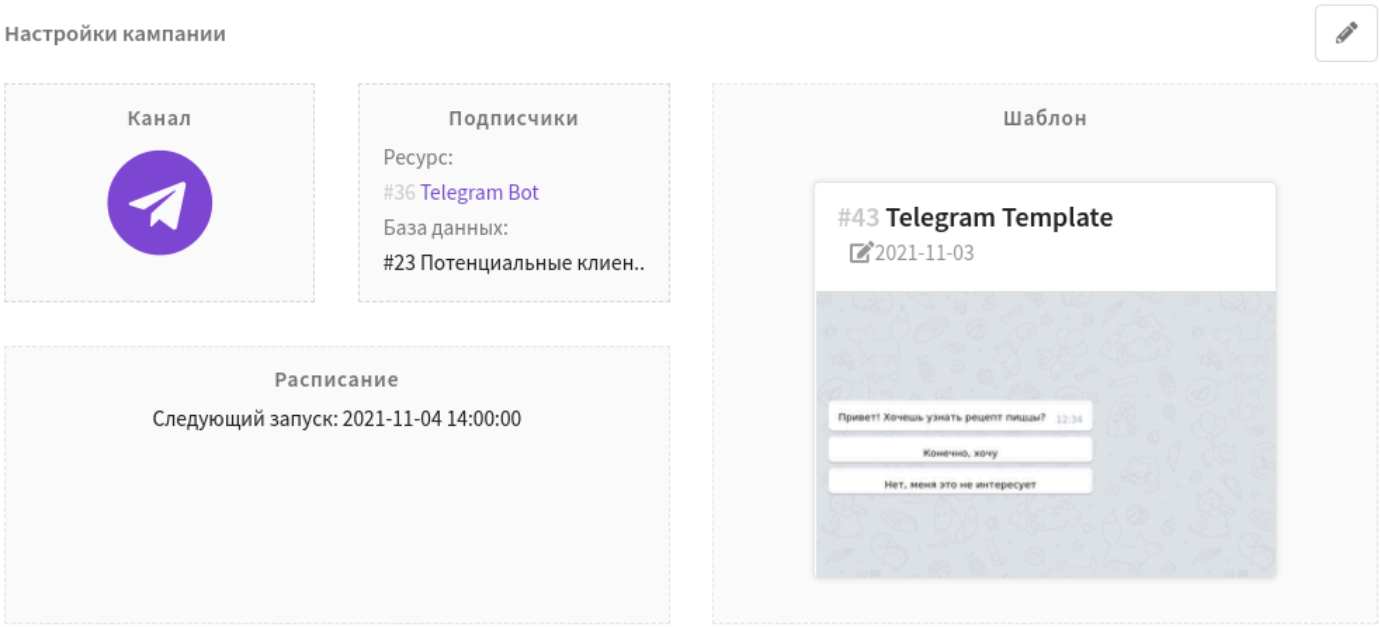
Есть возможность добавлять данные конкретного клиента, а также динамический текстовый контент из внешних источников.

Шаг 4. Настройте и запустите рассылку

Откройте раздел “**Рассылки**” и выберите подходящий способ рассылки: броадкаст, регулярная или триггерная рассылка.

Создайте новую рассылку. После заполнения основных настроек необходимо:

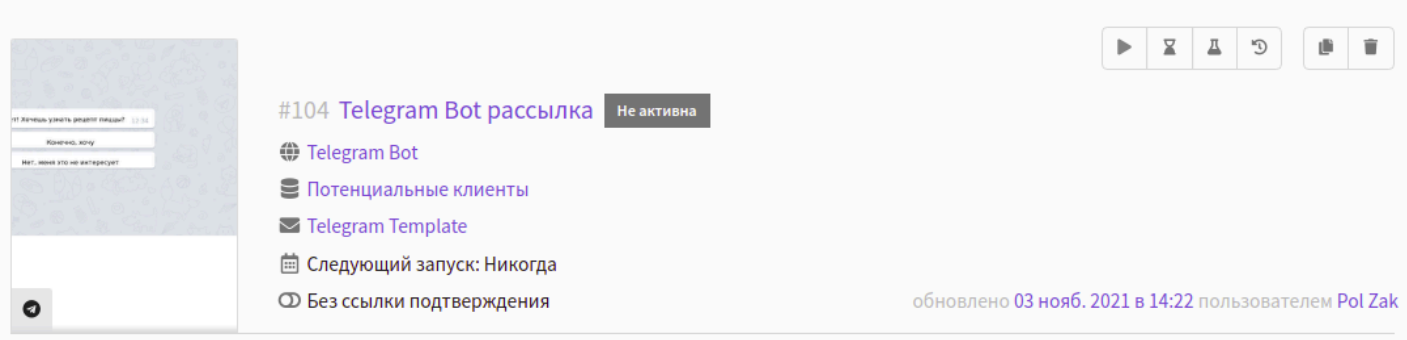
1. Выбрать **Telegram Bot** как канал связи
2. Выбрать настроенный ранее **ресурс** подписок и **аудиторию** получателей
3. Добавить в рассылку **шаблон** сообщения
4. Установить **расписание**



После завершения настройки рассылки сохраните её. Она будет отображаться в списке рассылок соответствующего типа. В правой части каждой записи есть меню быстрых действий. С его помощью вы можете сделать **тестовую** отправку

и, если всё успешно, **активировать** рассылку

:

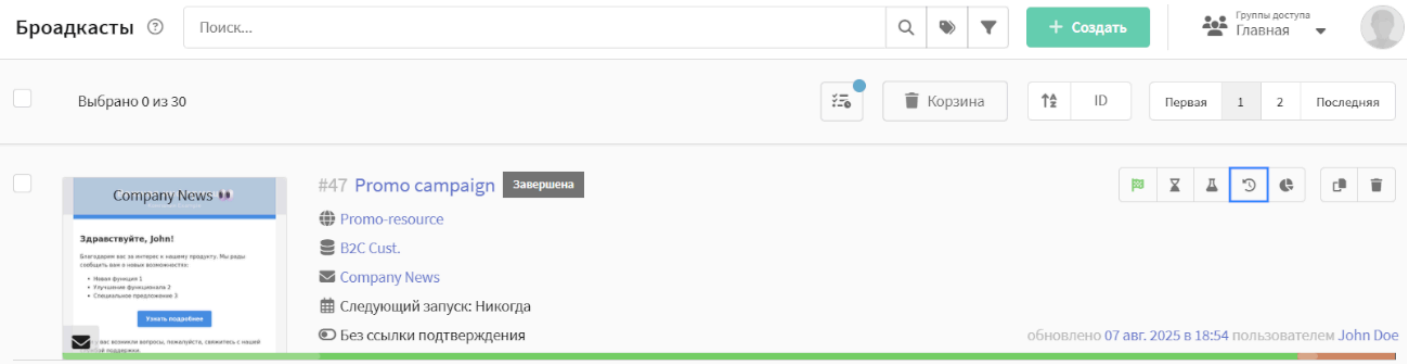


Шаг 5. Оцените эффективность рассылки

Журнал рассылок

Журнал рассылок — это инструмент, предоставляющий краткую сводку о ходе выполнения рассылки. В нем фиксируются ключевые события, параметры рассылок и статистика, что позволяет выявить возможные проблемы.

Чтобы открыть журнал рассылок, нажмите на кнопку **“Журнал”** в меню быстрых действий:



Журнал рассылок содержит следующую информацию:

Тип	Описание
Запуск рассылки	Время начала выполнения рассылки
Лимит	Показывает, установлен ли лимит на отправку сообщений
Источник подписчиков	**list** — База данных **segment** — Сегмент (если применимо)
ID базы данных / ID сегмента	Идентификатор используемой базы или сегмента
Время получения данных	Время загрузки контактов из источника (мс)
Количество профилей из источника	Общее число контактов в указанном источнике
Количество валидных подписчиков	Число пользователей с активной подпиской
Количество подписчиков без подписок	Пользователи без подписки на целевой канал (0 в примере)
Количество с некорректными полями	Ошибки в данных подписки (0 в примере)
Количество дубликатов	Повторяющиеся контакты (40 в примере)

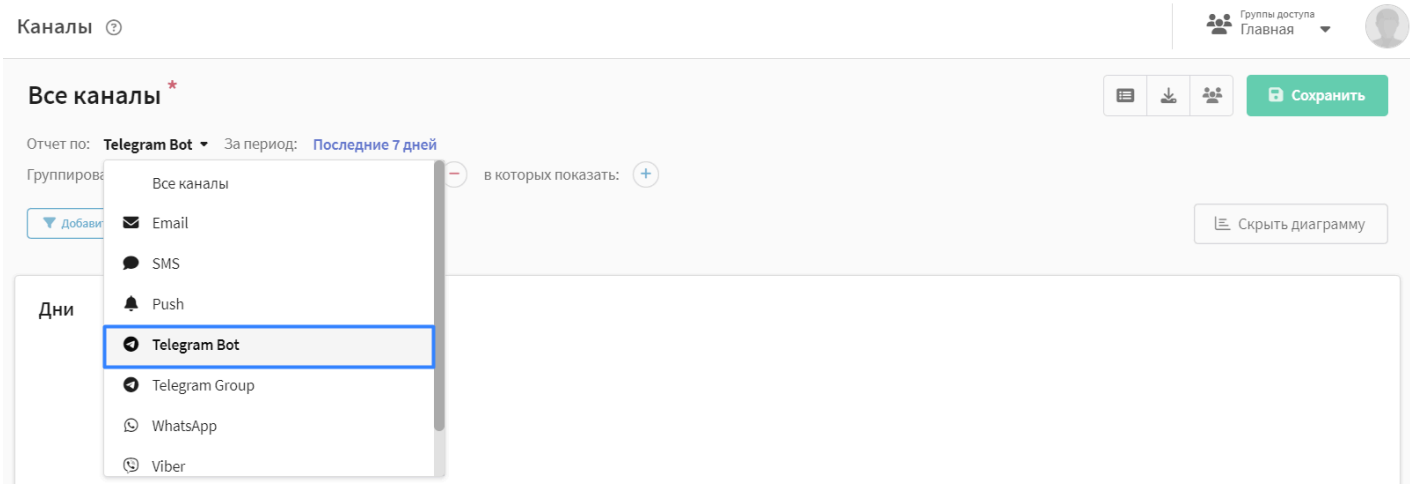
Тип	Описание
Количество отправленных сообщений	Фактическое число успешных отправок
Рассылка завершена	Время окончания рассылки
Всего времени затрачено	Общая продолжительность рассылки

Отчеты и Аналитика

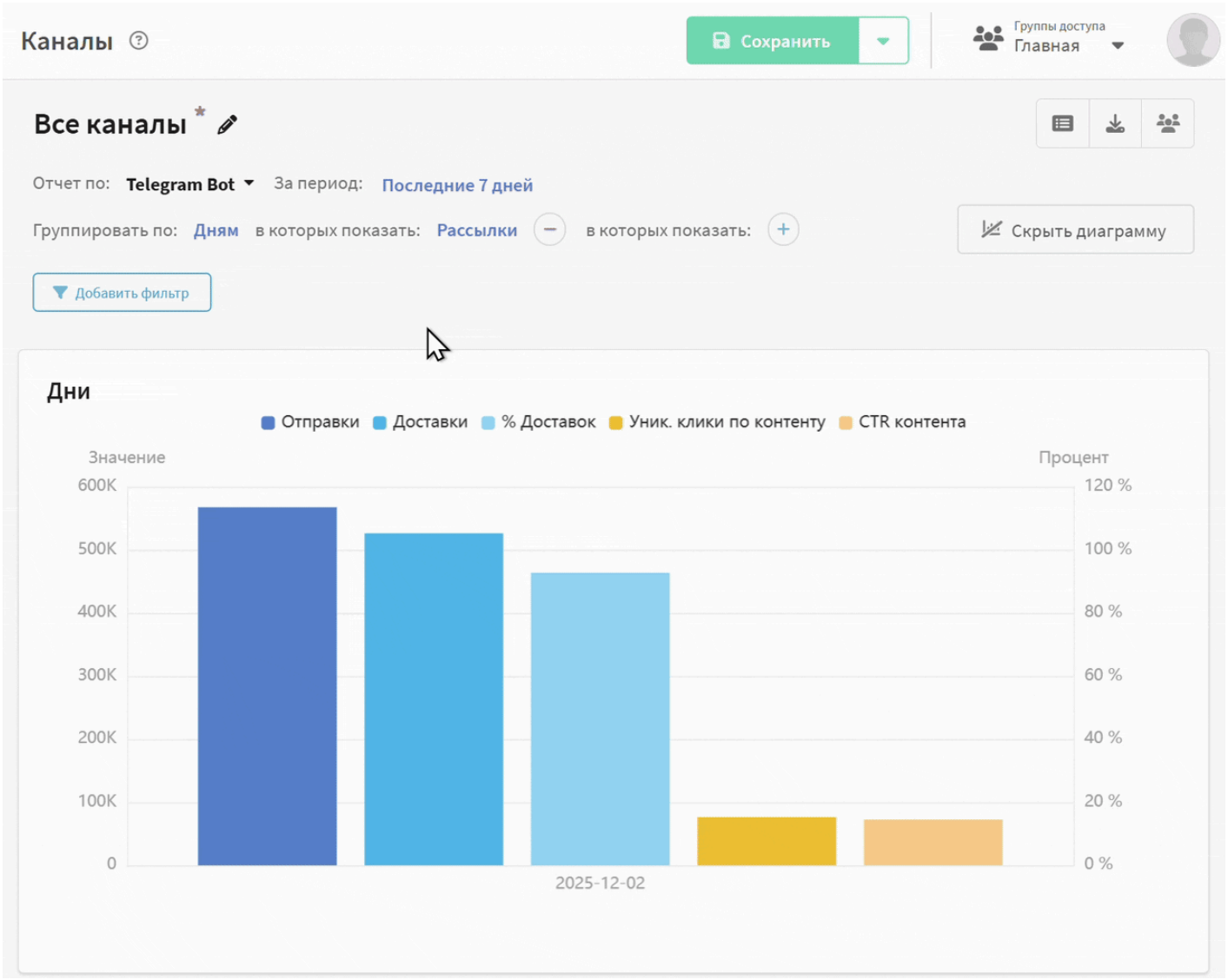
Для оценки результатов текущей рассылки вы можете просмотреть отчет по каналу.

Отчет по каналам предоставляет аналитику по одной или нескольким рассылкам.

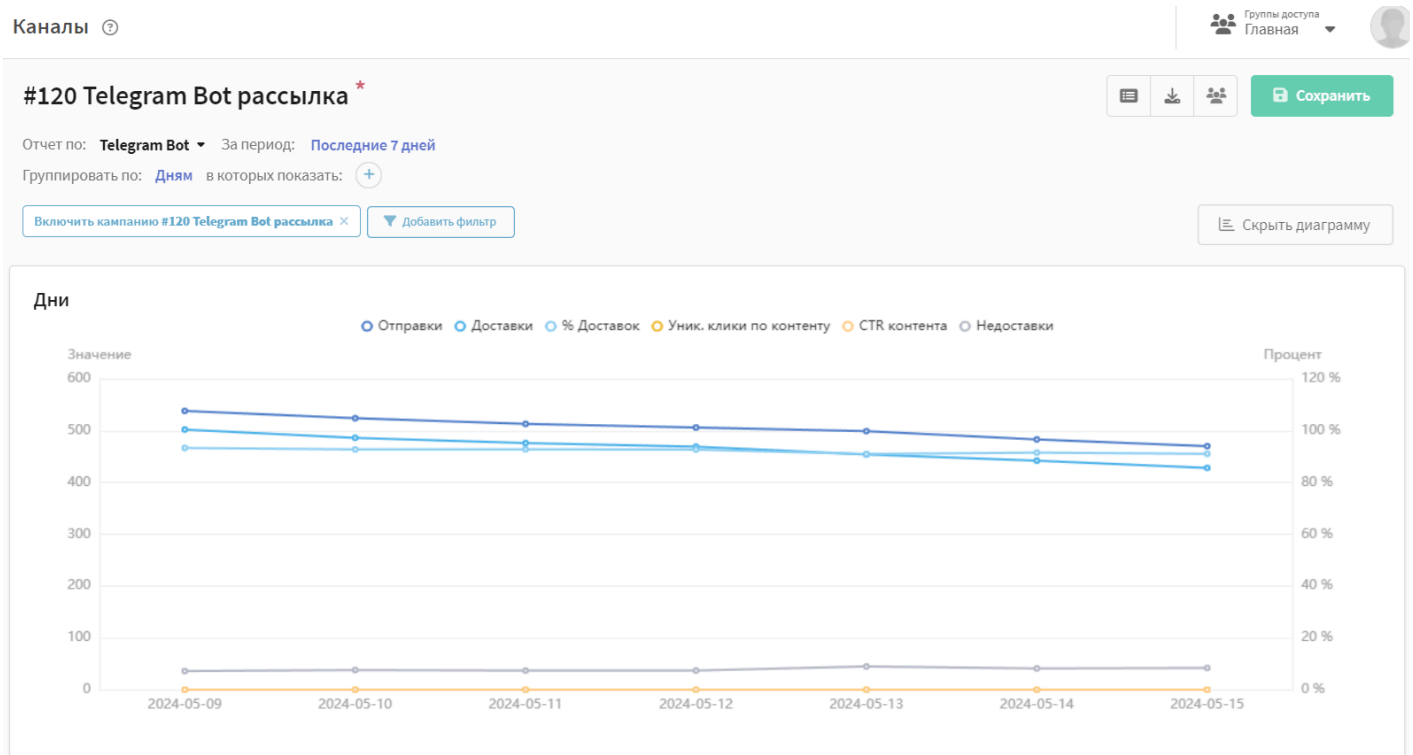
Для получения статистики используйте раздел “**Аналитика**” по **Каналам** в главном меню платформы. По умолчанию отобразится отчет по **Всем каналам** за последние 7 дней. Для группировки по **“Telegram Bot”** каналу выберите **“Отчет по”:** .



Следующим шагом **“Добавьте фильтр”** по **“Рассылкам”**. В отобразившемся списке выберите нужную рассылку.



В результате вы получите статистику по выбранной рассылке.



В таблице отобразятся данные о количестве отправленных, доставленных или недоставленных сообщений, а также информация о количестве уникальных кликов и открытий.

Наименование	Отправки	Доставки	% Доставок	Недоставки	% Недоставок	Ошибки генерации
Всего	3 533	3 257	92,19 %	276	7,81 %	5 863
2024-05-15	470	428	91,06 %	42	8,94 %	877
2024-05-14	483	442	91,51 %	41	8,49 %	861
2024-05-13	499	454	90,98 %	45	9,02 %	842
2024-05-12	506	469	92,69 %	37	7,31 %	835
2024-05-11	513	476	92,79 %	37	7,21 %	828
2024-05-10	524	486	92,75 %	38	7,25 %	817
2024-05-09	538	502	93,31 %	36	6,69 %	803

Telegram Group

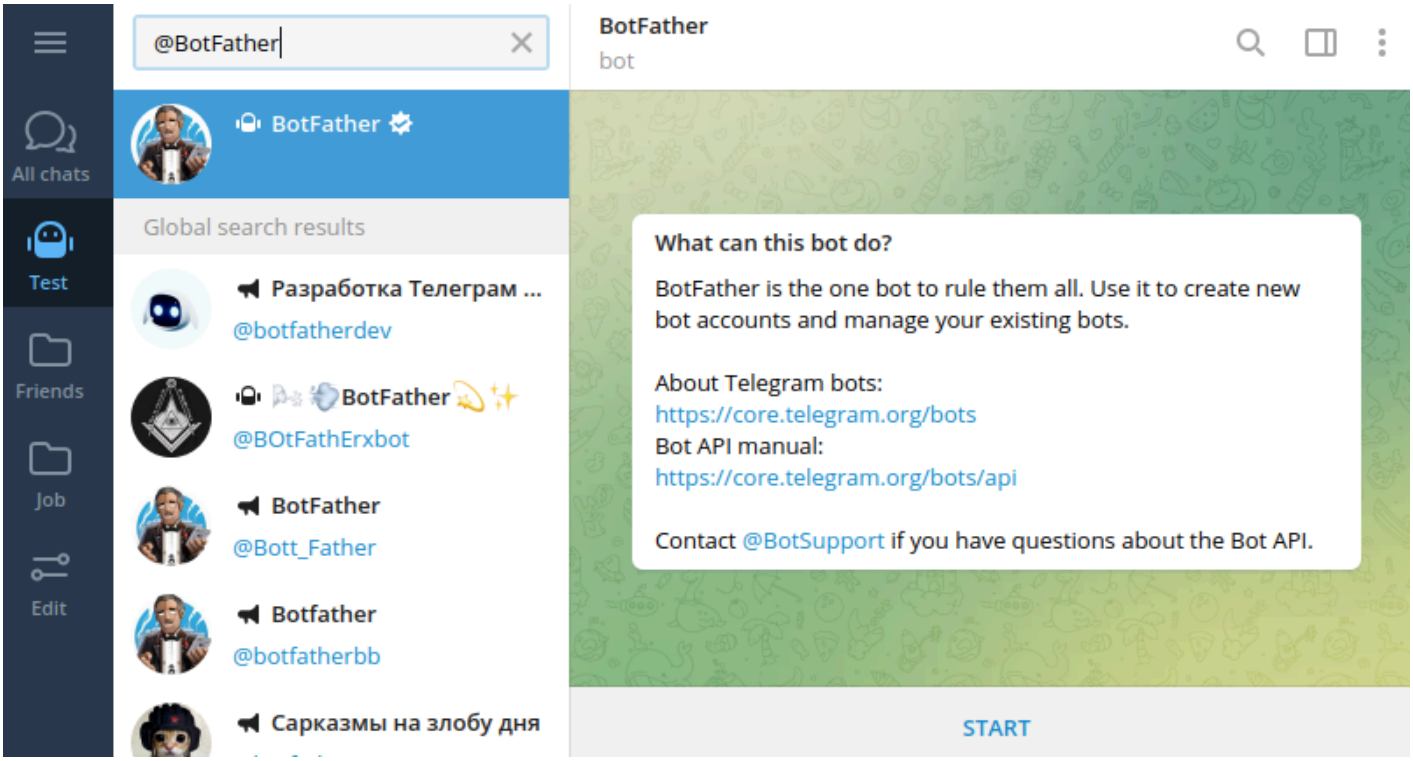
Первая рассылка в Telegram Group

Telegram-канал представляет собой ленту текстовых сообщений, которые дополняются медиафайлами и эмодзи. Также в канале доступны комментарии и опросы пользователей. Группы отличаются от каналов двусторонней коммуникацией: каждый пользователь может написать сообщение в общий чат.

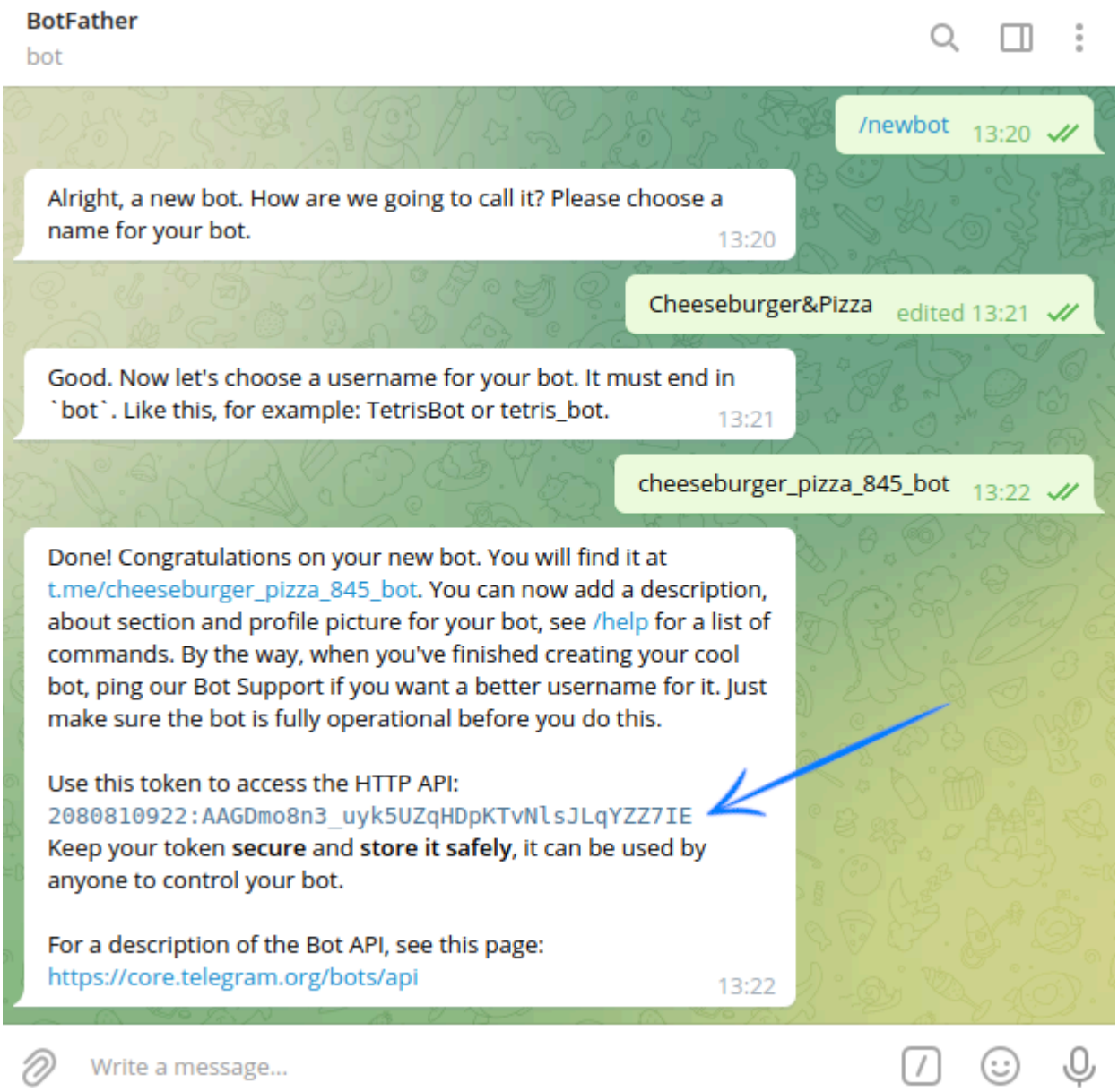
Канал Telegram Group в платформе Altcraft позволит вам настроить автоматическую публикацию постов в каналах и группах мессенджера с помощью размещения.

Шаг 1. Создайте бот и получите токен

1. Зайдите в Telegram, введите `@BotFather` в окне поиска и выберете нужного бота:



2. Начните диалог с ботом, нажав кнопку `START` в чате. Затем отправьте боту команду `/newbot`. Введите сначала имя, а затем username для нового бота. Обратите внимание, username должен оканчиваться на “**bot**” или “**Bot**”. Скопируйте полученный токен бота:



Шаг 2. Добавьте бота в группу или канал и получите chat_id

1. Создайте группу или канал. Более подробную информацию о том, как это сделать, вы найдете здесь.

2. Добавьте бота в группу или канал. Затем назначьте боту права администратора.

3. Отправьте тестовое сообщение в чат с помощью API-запроса. Сделать это можно через Postman или любой другой инструмент для тестирования API.

Метод: POST

URL-адрес:

```
https://api.telegram.org/bot[BOT_API_KEY]/sendMessage?chat_id=[CHANNEL_NAME]&text=[MESSAGE_TEXT]
```

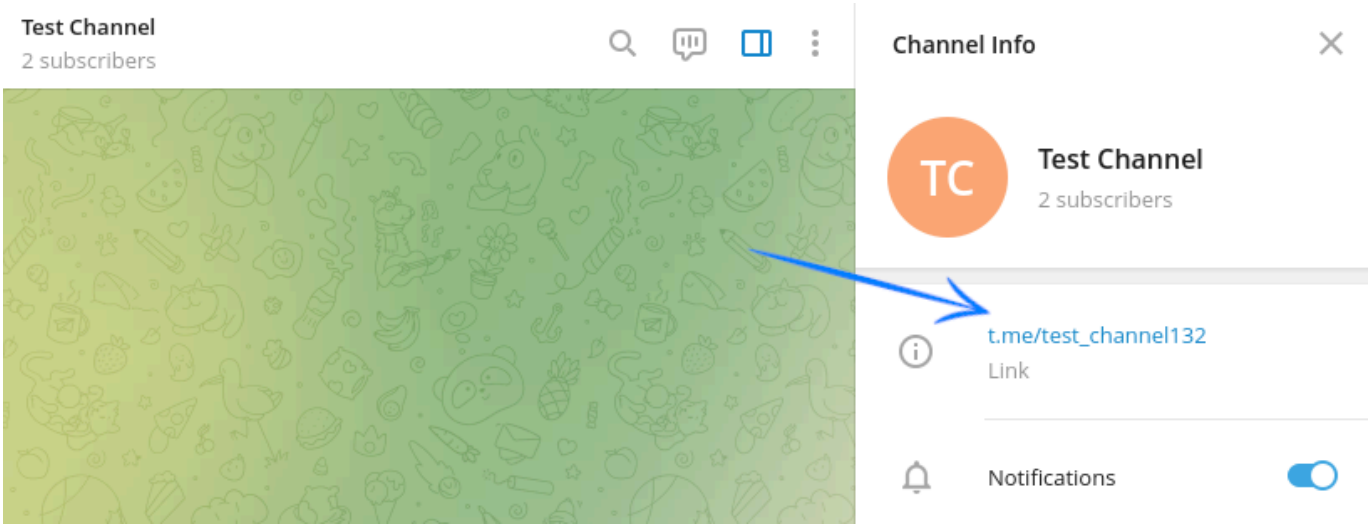
- `[BOT_API_KEY]` — токен бота, полученный в диалоге с `@BotFather`.
- `[CHANNEL_NAME]` — название канала или группы через `@`.
- `[MESSAGE_TEXT]` — текст сообщения.

Пример:

`https://api.telegram.org/bot2080910120:AAGfho8n3_uyk5trh4DpKTvNlsJLkadsZ7IE/sendMessage?chat_id=@new_test_group132&text=Привет!`

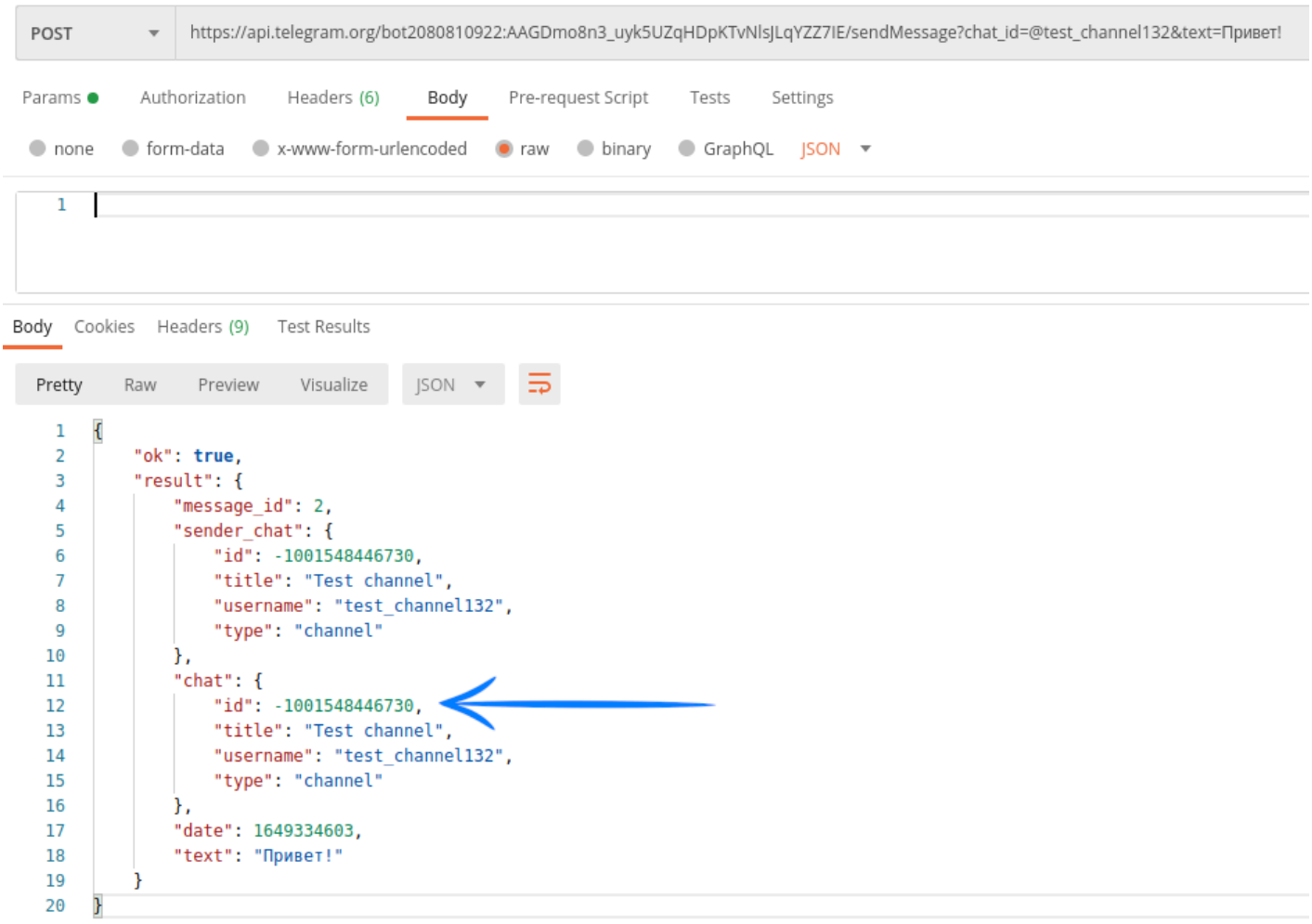
INFO: Где найти название канала или группы

Название заложено в ссылке на канал или группу. Например, если ссылка `t.me/test_channel132`, то названием канала станет часть после слеша — `test_channel132`.



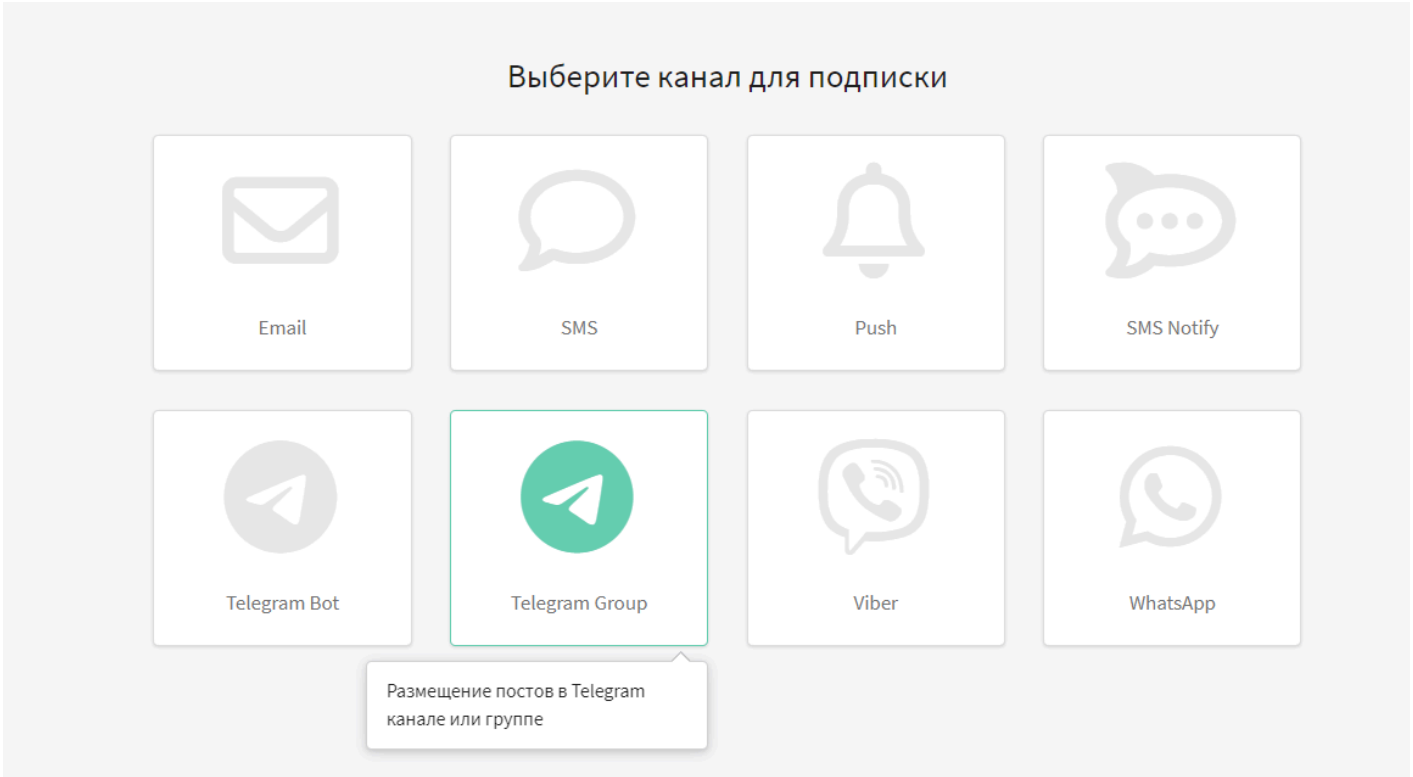
Ответ:

В теле ответа вы получите **`chat_id`**. Он должен начинаться с **`-100`** .



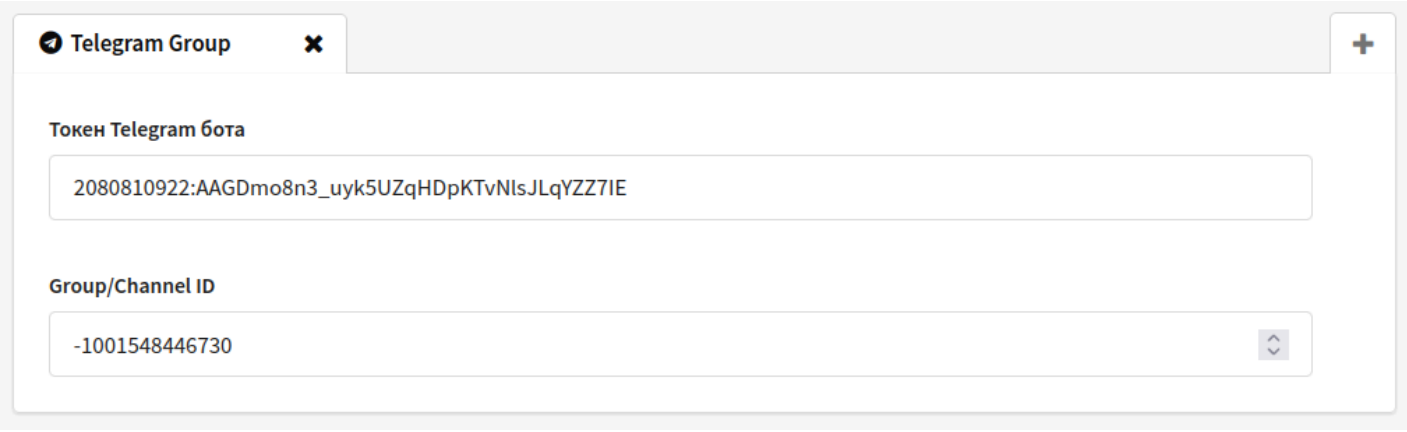
Шаг 3. Создайте ресурс

Перейдите в раздел “**Ресурсы**” и создайте новый ресурс или выберите существующий. В качестве канала используйте **Telegram Group**:



В основных настройках добавьте необходимые базы данных. Укажите **трекинг домен по умолчанию**, который будет обрабатывать переходы подписчиков по ссылкам в постах Telegram группы.

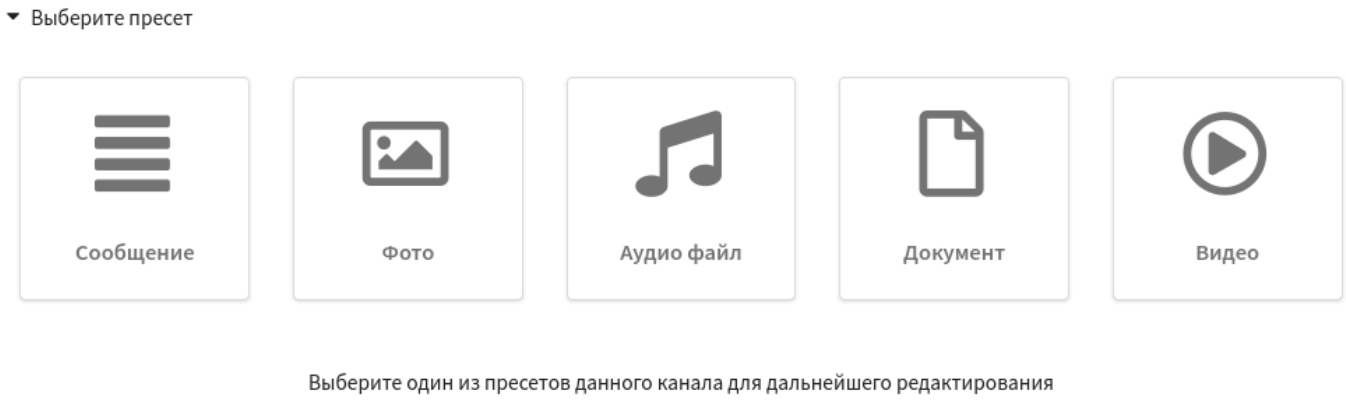
Добавьте полученные ранее **токен бота** и **ID группы или канала** (`chat_id`):



Шаг 4. Создайте шаблон

Перейдите в раздел “**Шаблоны**”. При создании нового шаблона выберите **Telegram Group** канал или добавьте этот канал для уже существующего шаблона.

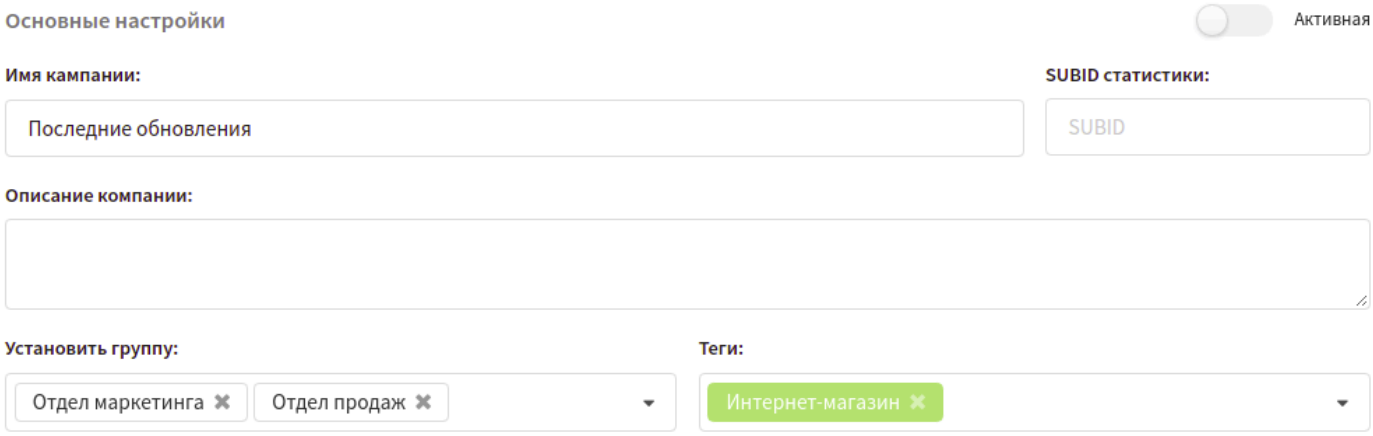
В **Telegram** версии шаблона необходимо выбрать тип контента:



Есть возможность добавлять данные конкретного клиента, а также динамический текстовый контент из внешних источников.

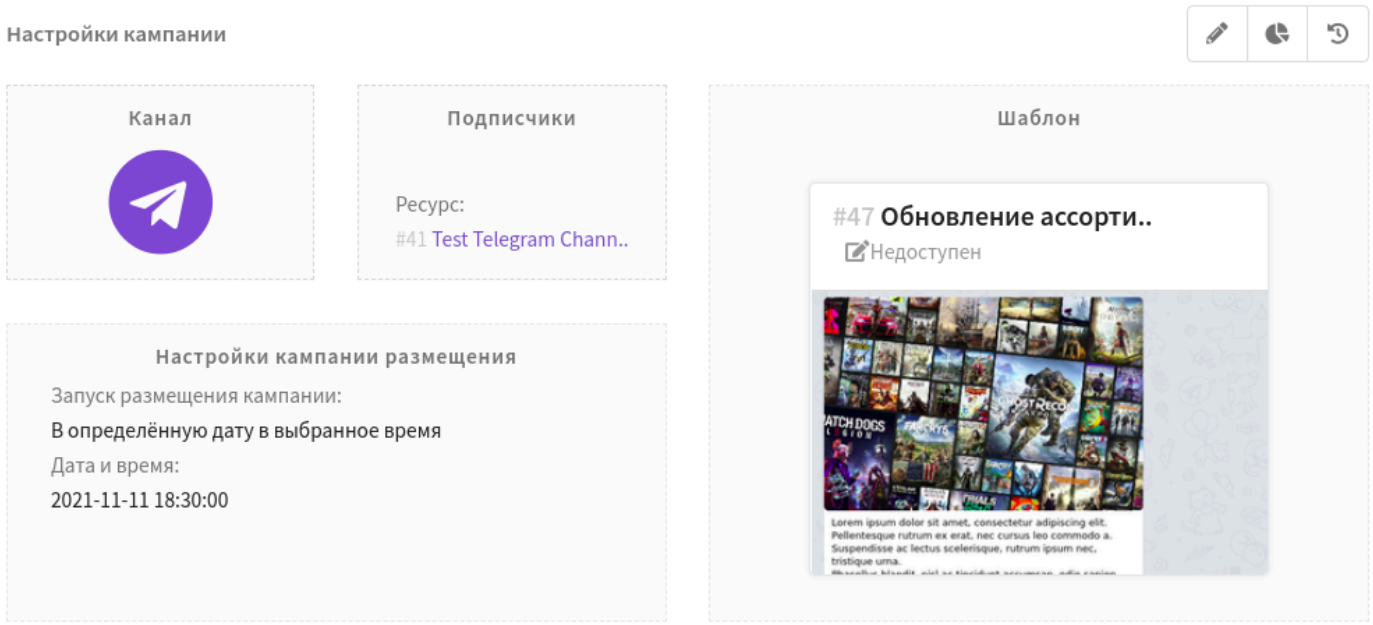
Шаг 5. Настройте и запустите размещение

Откройте раздел “**Рассылки**” и выберите “**Размещения**”. Создайте новую рассылку.



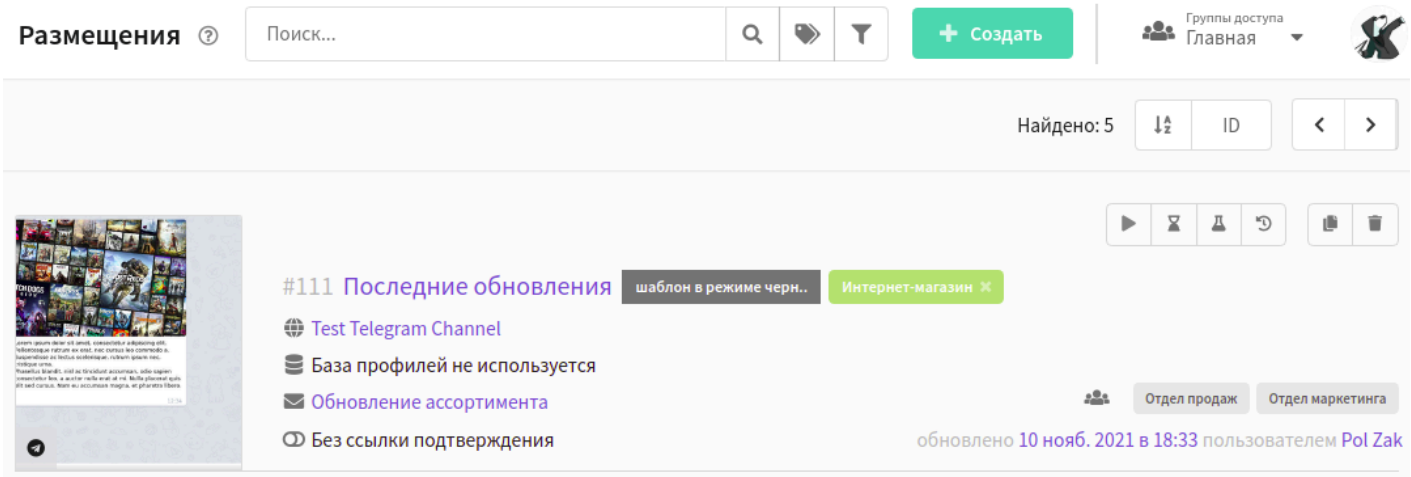
После заполнения основных настроек необходимо выбрать:

1. **Telegram Group** как канал связи;
2. Настроенный ранее **ресурс** подписок;
3. **Шаблон** сообщения;
4. **Дату и время** выкладывания поста в группу.



По окончанию настройки сохраните рассылку. Она будет отображаться в общем списке рассылок. В правой части каждой записи есть меню быстрых действий. С его помощью вы можете **активировать** размещение

:

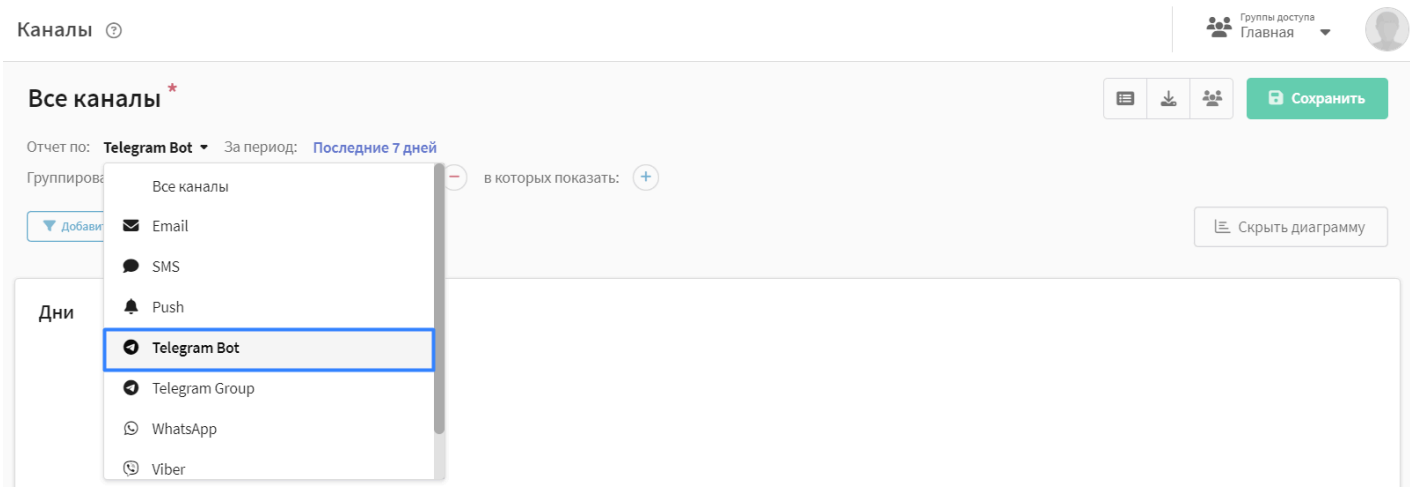


Шаг 6. Оцените эффективность рассылок

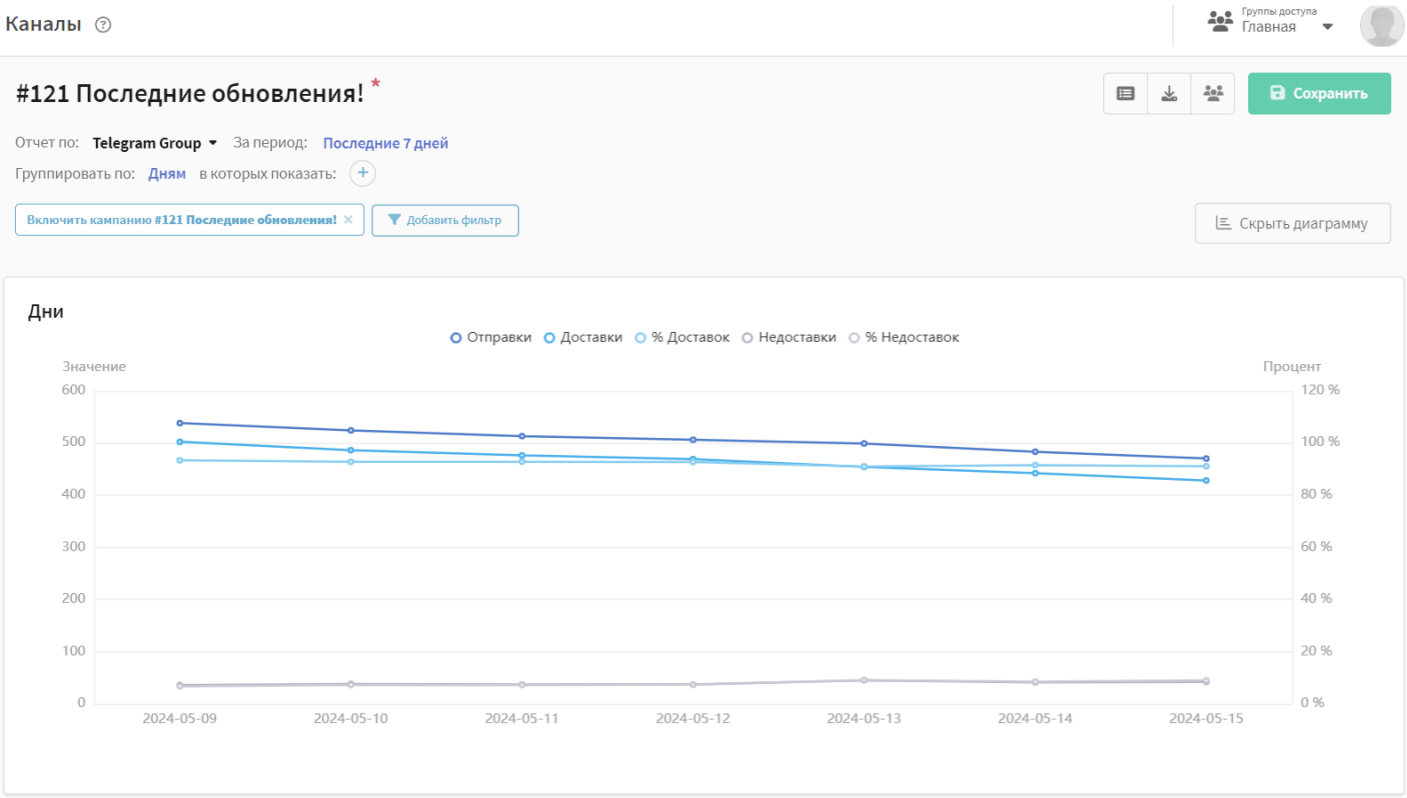
Для оценки результатов текущей рассылки вы можете просмотреть отчет по каналу.

Отчет по каналам предоставляет аналитику по одной или нескольким рассылкам.

Для получения статистики используйте раздел “**Аналитика**” по **“Каналам”** в главном меню платформы. По умолчанию отобразится отчет по **Всем каналам** за последние 7 дней. Для группировки по **“Telegram Group”** каналу выберите **“Отчет по”:** .



Следующим шагом **“Добавьте фильтр”** по **“Рассылкам”**. В отобразившемся списке выберите нужную рассылку. В результате вы получите статистику по выбранной рассылке.



В таблице отобразятся данные о количестве отправленных, доставленных или недоставленных сообщений, а также информация о количестве уникальных кликов и открытий.

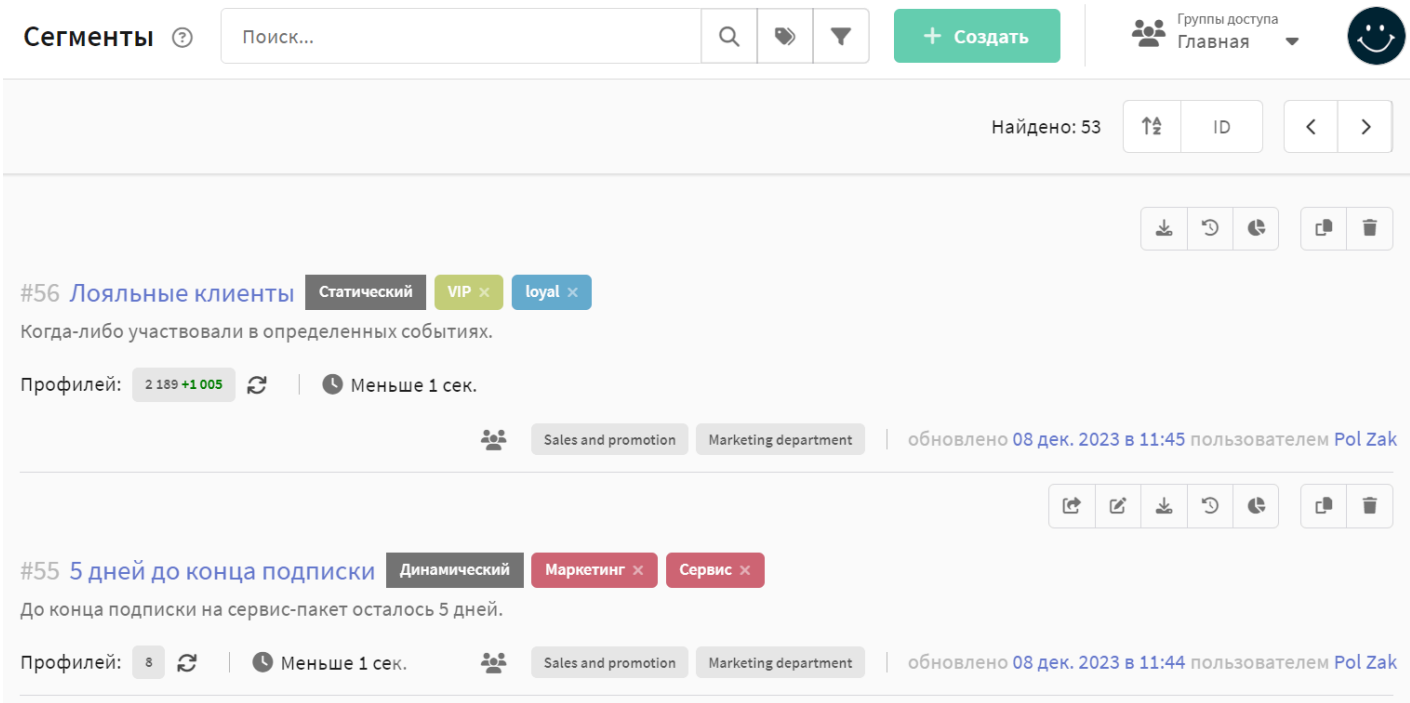
Наименование	Отправки	Доставки	% Доставок	Недоставки	% Недоставок	Ошибки генерации
Всего	3 533	3 257	92,19 %	276	7,81 %	5 863
2024-05-15	470	428	91,06 %	42	8,94 %	877
2024-05-14	483	442	91,51 %	41	8,49 %	861
2024-05-13	499	454	90,98 %	45	9,02 %	842
2024-05-12	506	469	92,69 %	37	7,31 %	835
2024-05-11	513	476	92,79 %	37	7,21 %	828
2024-05-10	524	486	92,75 %	38	7,25 %	817
2024-05-09	538	502	93,31 %	36	6,69 %	803

Сегментация

	Сегмент — это часть базы данных, которая включает профили, подходящие под определённые в сегменте условия. После того, как сегмент создан, его можно включать в кампанию или рассылку как источник подписчиков.

Сегментация профилей подписчиков позволяет таргетировать ваши маркетинговые рассылки и кампании. Выборка происходит на основе данных профилей клиентов, истории действий в сообщениях, на сайте или в приложении, а также на основе запросов к внешним источникам, таким как SQL-базы данных или API веб-сервисов.

Чтобы сегментировать аудиторию подписчиков, в главном меню откройте **Данные —> Сегменты**. В списке для каждого сегмента будет отображаться **тип** и краткая информация. С помощью контекстного меню вы можете совершить дополнительные действия, например, массово обновить профили сегмента.



Откройте существующий сегмент или добавьте новый запрос, нажав в верхней части страницы кнопку **+ Создать**.

Типы сегментов

Статический	**Статические сегменты** Статический сегмент изначально может быть рассчитан на выборке из базы данных, однако после сохранения он не будет пересчитываться автоматически. Вы сможете добавлять профили в статический сегмент вручную по одному или на основе другого сегмента, а также удалять профили из сегмента. Такие сегменты используются при работе с фиксированными группами клиентов.

 Динамический	**Динамические сегменты** Динамические сегменты используются в основном для таргетирования ваших маркетинговых рассылок. Такой сегмент не сохраняет список профилей, удовлетворивших условиям. Вместо этого сохраняется сама структура запроса выборки. Расчёт происходит непосредственно в момент запуска рассылки — и, значит, используются наиболее актуальные данные.
 Обновляемый	**Обновляемый сегмент** Обновляемый сегмент нужен для динамического обновления аудитории по расписанию. В него входят профили, которые соответствуют заданным условиям. Аудитория сохраняется до следующей проверки по расписанию. При необходимости расчет можно запустить вручную.

Статические сегменты

Статический сегмент — это фиксированный список профилей из выбранной базы данных. После наполнения его состав не изменяется автоматически при обновлении данных в базе.

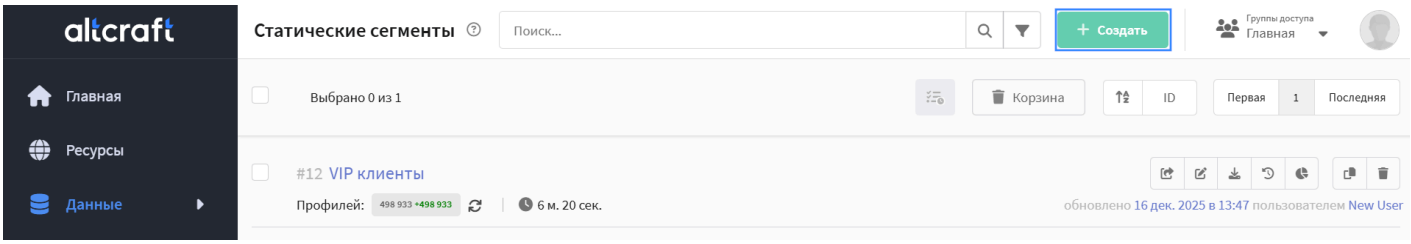
Этот тип сегмента подходит для задач, где нужна фиксированная аудитория например: рассылки по заранее утверждённому списку клиентов или для разовых выборок для отчётов или интеграций. Также на основе статического сегмента формируется аудитория [кампании](#).

	Чтобы начать работу, перейдите в раздел **Данные — Сегменты — Статические сегменты**.

Создание и общее управление сегментами

Создание сегмента

В списке сегментов нажмите **+ Создать:**



В открывшейся форме укажите **Имя** и **Описание** сегмента, а также выберите **Базу данных (БД)**, из которой будут отбираться профили. Затем нажмите **Создать:**

INFO: Важно

База данных задаётся только при создании и не может быть изменёна позже

Настройка и управление сегментом

После создания или выбора сегмента откроется его редактор. В правом верхнем углу редактора сегмента доступна панель действий для работы со всем текущим списком профилей:

- **Задать теги**: Назначить или изменить теги для категоризации сегмента.
- **Указать группы доступа**: Настроить, какие группы пользователей могут видеть и редактировать этот сегмент.
- **Скопировать профили в базу профилей**: Скопировать все профили из этого сегмента в другую выбранную базу профилей.
- **Скопировать профили в статический сегмент**: Скопировать все профили из этого сегмента в другой выбранный статический сегмент.
- **Массовое редактирование профилей**: Открыть интерфейс для изменения данных (например полей или подписок) у всех профилей в сегменте одновременно.
- **Добавить в сценарий**: Отправить участников сегмента в выбранный сценарий. **По умолчанию отправляются только те профили, которые ещё не участвуют в этом сценарии.** Чтобы отправить всех участников сегмента, включая уже участвующих, установите галочку в соответствующем чекбоксе.

INFO: Обратите внимание

Теги и группы доступа можно задать сразу после создания сегмента. Остальные действия будут доступны после первого пересчёта данных.

Управление аудиторией

Состав сегмента можно менять разными способами: вручную, через настройку условий, с помощью API или импорта. Обратите внимание, что условия действуют только один раз при их применении и не сохраняются для постоянного обновления сегмента. Для добавления или удаления профилей нажмите:

- **“Добавить”**: При нажатии открывается боковая панель для настройки условий, по которым профили будут **добавлены** в сегмент.
- **“Удалить”** (становится доступна после добавления профилей): При нажатии открывается аналогичная боковая панель для настройки условий, по которым профили будут **удалены** из сегмента.

Добавление профилей по условиям

При нажатии кнопки **“Добавить”** открывается редактор условий в боковой панели. Здесь доступны категории для настройки правил выборки из выбранной базы профилей:

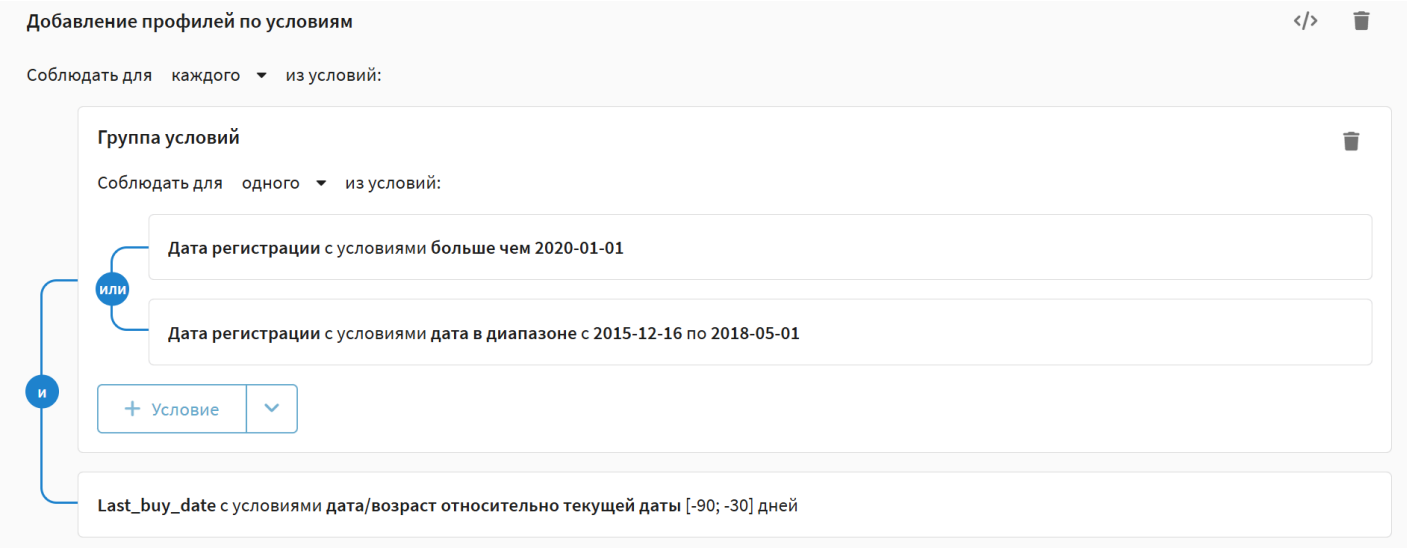
Правила строятся на основе фильтров, которые можно объединять в группы. Доступны следующие категории данных для фильтрации:

- **Данные профиля**, включая сырые запросы, структуру профиля, данные браузера и операционной системы, статусы, системные поля (UTM, подписки, идентификатор), основные поля (email, телефоны, имя, фамилия, дата рождения, пол, данные о регистрации и местоположении), а также дополнительные пользовательские поля.

- **Взаимодействия с сущностями**: участие в сценариях и сегментах (динамических или статических), поведение в попапах (открытие, заполнение, клики), работа с промокодами, связи, формы, заказы, достижение целей, участие в кампаниях (включая принадлежность к контрольной или тестовой группе).

- На основе **Поведения в рамках канала**: взаимодействие с Push-уведомлениями, SMS, Email, Telegram Bot, Telegram Group, WhatsApp, Viber и VK Notify.

Внутри группы условия могут работать по логике “И” (соблюдать для каждого условия) или “ИЛИ” (соблюдать для одного из условий):



Можно быстро загрузить готовые условия, нажав **“Импортировать условия”** и выбрав существующий сегмент — скопируются только внутренние правила фильтрации.

> **INFO: Обратите внимание**
>
> Статический сегмент не сохраняет условия сегментации, поэтому импорт условий доступен только из динамических или обновляемых сегментов.

После настройки правил нажмите **“Добавить”**. Система начнёт расчёт количества подходящих профилей. Как только расчёт завершится, обновится список профилей в сегменте.

Скрипт

Запрос сегментации можно также составлять вручную, с помощью встроенного редактора **JSON**. Для этого нажмите в запросе кнопку **Скрипт**:

Удаление профилей по условиям

Для того чтобы удалить профили из сегмента нажмите **“Удалить”**. Справа откроется аналогичный редактор условий. Он позволяет настроить фильтры для исключения профилей из текущего сегмента. В нём также доступны правила и группы правил, импорт условий и настройка ограничений аудитории сегмента.

> **NOTE**
>
> Функция **“Удалить профили, которые состоят в другом сегменте”** позволяет быстро исключить из текущего сегмента все пересекающиеся с другим сегментом профили.

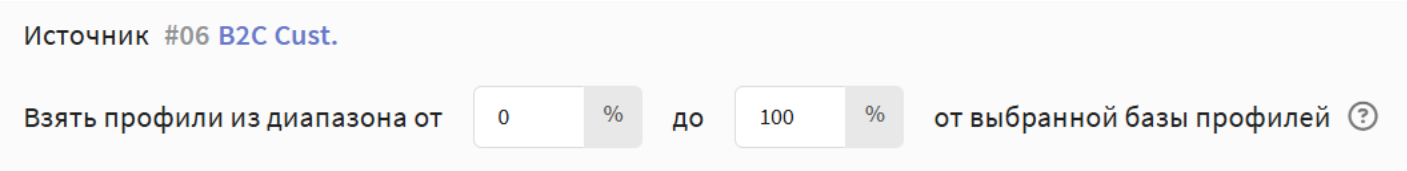
Ограничение аудитории

В статическом сегменте можно применять два типа ограничений, которые работают последовательно.

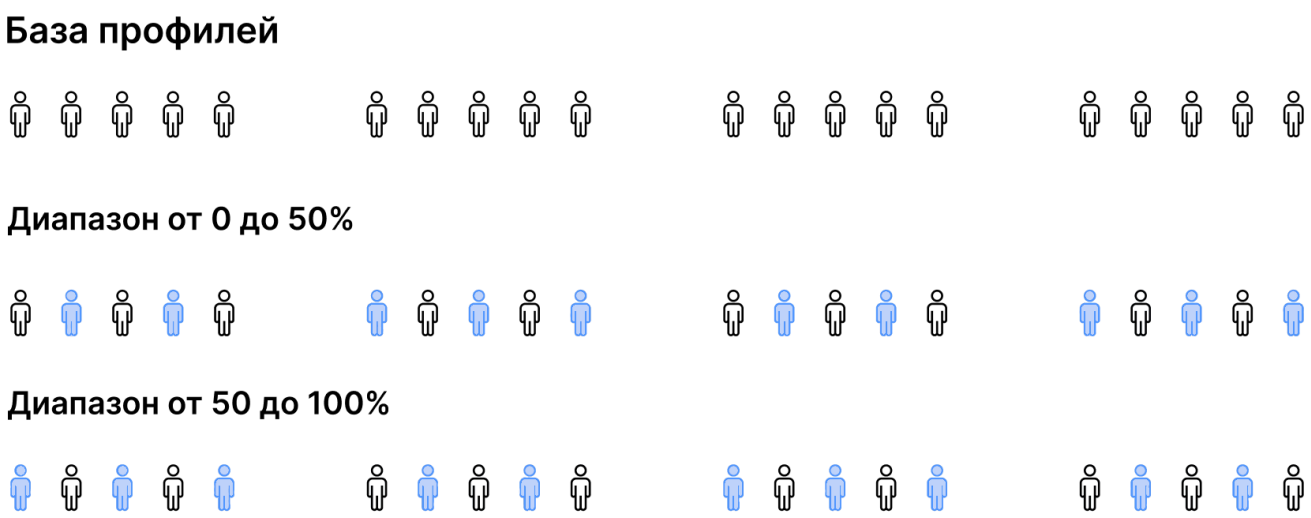
Диапазон определяет часть базы, к которой будут применены условия сегментации. По умолчанию диапазон равен 100%, т.е. если в базе 1000 профилей, то условия сегментации применятся ко всем профилям в базе.

Ограничения по максимальному количеству профилей — лимит профилей, которые могут быть включены в сегмент. Например, если 1000 профилей соответствуют условиям сегментации, а **максимальное количество** — 400, то в сегмент войдут первые 400 профилей из выборки.

Ограничения, заданные с помощью **диапазона**, распространяются на всю базу. Ограничения, заданные с помощью **Ограничения по максимальному количеству профилей**, накладываются на результат выборки сегмента.



Платформа закрепляет диапазон за конкретными пользователями: один и тот же диапазон всегда отбирает **одних и тех же людей** из базы, даже если вы используете его в разных сегментах. Это означает, что выборка **не случайна** — каждому пользователю сопоставляется числовое значение. Профиль входит в сегмент, если рассчитанное для него значение попадает в выбранный диапазон. Благодаря этому можно создавать непересекающиеся выборки, назначая разные диапазоны (например, диапазон от 0 до 20 процентов и диапазон от 80 до 100 процентов никогда не пересекутся). При добавлении новых пользователей в базу они также могут попасть в заданный диапазон с той же вероятностью — например, с 1/5, если выбран диапазон в 20%.



Импорт профилей из другого сегмента

Для импорта профилей из другого сегмента, нажмите на стрелку вниз ▾, она находится рядом с основной кнопкой **“Добавить”** и в контекстном меню выберите **Импортировать из сегмента**:

Эта опция позволяет быстро добавить в текущий сегмент **все профили** из другого существующего сегмента.

> **INFO: Важно**
>
> Можно выбрать только сегмент, созданный на основе той же базы профилей. После выбора сегмента профили будут добавлены, обновится список, статистика и история.

Ручное добавление и удаление профилей

Добавить или удалить из сегмента профиль можно вручную. Для этого в карточке профиля перейдите во вкладку **Взаимодействия** в блок **Сегменты**, выберите необходимый сегмент и отредактируйте:

Просмотр и анализ состава сегмента

Таблица профилей

После расчета в окне статического сегмента отображается таблица со всеми профилями, входящими в сегмент. Помимо данных профиля, в таблице отображаются системные поля, такие как “Тип источника” (например, импорт, условия) и “Источник добавления” (идентификатор сегмента или рассылки). Эти поля помогают отследить историю попадания каждого профиля в сегмент.

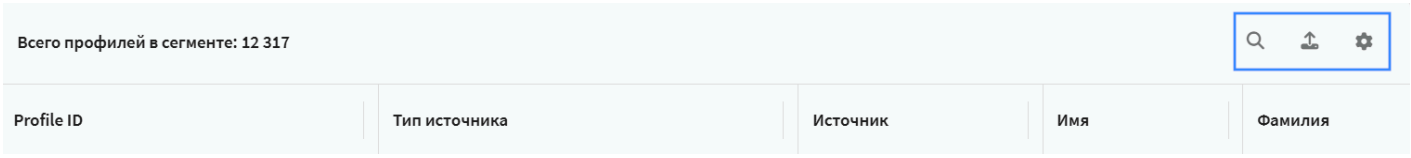
Доступны следующие действия для работы с этой таблицей:

- **Поиск**

  : найти конкретный профиль по имени, email или другим данным.
- **Экспортировать профили**

  : выгрузить отображаемый список во внешний файл.
- **Настройка отображаемых полей**

  : чтобы добавить или убрать колонки в таблице, перетащите нужное **Поле** из боковой колонки **“Добавить поле”** в область **“Отображаемые поля”** или обратно.



История обновлений

Под таблицей профилей расположен блок **“История обновлений”**. В нём наглядно представлена информация об изменении общего количества профилей в сегменте за выбранный период. Вы можете указать нужный период (дату начала и окончания) и сгруппировать данные по дням или неделям:

При наведении курсора на столбец графика появится всплывающая подсказка. В ней показано **суммарное изменение числа профилей** и детализация: по каким конкретно действиям и в каком количестве профили были добавлены в сегмент или удалены из него:

Подсказка группирует изменения по следующим типам действий:

Добавлено

- **Импорт из другого сегмента**: добавление всех профилей из другого сегмента.
- **Импорт из файла в базу**: автоматическое добавление профилей после импорта файла в Базу профилей, если была включена соответствующая опция.
- **По условиям сегмента**: добавление по фильтрам (операция “Добавить”).
- **Из карточки профиля**: ручное добавление через карточку профиля (вкладка **“Взаимодействия”**).

- **Из узла сценария**: добавление профилей через узел “Обновление статического сегмента” в сценарии.
- **По API**: добавление через API.

Удалено

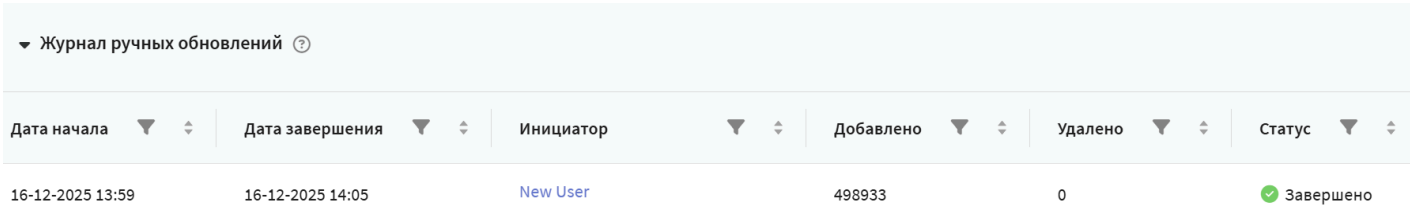
- **Импорт из другого сегмента**: удаление всех профилей, которые есть в другом сегменте (операция “Удалить профили, которые состоят в другом сегменте”).
- **По условиям сегмента**: удаление по фильтрам (операция “Удалить”).
- **Из карточки профиля**: ручное удаление через карточку профиля .
- **Из узла сценария**: удаление профилей узлом “Обновление статического сегмента” в сценарии.
- **Из базы профилей**: удаление профиля из самой Базы профилей (приводит к его автоматическому удалению из всех сегментов).
- **По API**: удаление через API.

Напротив каждого типа действия указано точное количество затронутых профилей.

Журнал ручных изменений

Ниже истории обновлений находится **“Журнал ручных изменений”**. Это детальный лог всех операций, повлиявших на состав сегмента. По каждой операции отображается:

- **Дата начала**: время запуска процесса сегментации (добавления/удаления по условиям).
- **Дата завершения**: время окончания процесса.
- **Инициатор**: имя пользователя, который инициировал операцию.
- **Добавлено**: количество профилей, добавленных в результате операции.
- **Удалено**: количество профилей, удалённых в результате операции.
- **Статус**: текущее состояние операции (например, “Завершено”, “В процессе”, “Ошибка”).



NOTE

В журнале не отображаются записи о добавлении профилей через API или из сценария.

Зависимые объекты

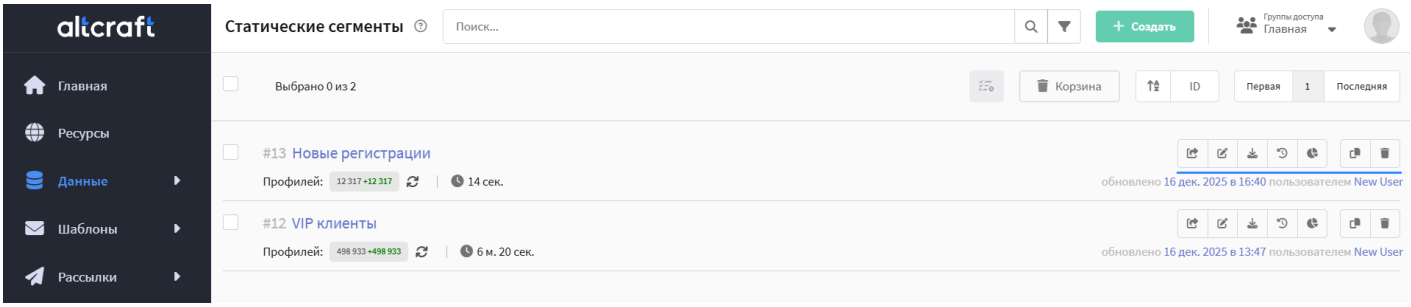
Ниже журнала ручных изменений может отображаться блок **“Зависимые объекты”**. Он появляется автоматически, если текущий статический сегмент используется в других сущностях платформы.

Этот блок показывает, где именно задействован ваш сегмент (например в рассылках или сценариях)

Управление сегментами

В строке с нужным статическим сегментом в общем списке доступны следующие действия для работы со статическими сегментами:

- **Копировать профили из сегмента**: Скопировать список профилей из текущего сегмента в другой выбранный сегмент.
- **Обновить профили**: Массово изменить данные профилей в сегменте (например, обновить поля, добавить подписку).
- **Экспортировать профили**: Выгрузить данные профилей, входящих в сегмент, во внешний файл.
- **Экспортировать историю профилей**: Выгрузить историю взаимодействий и изменений для профилей из сегмента.
- **Аналитика**: Перейти к аналитическим отчётам по данному сегменту.
- **Клонировать сегмент**: Создать точную копию сегмента (с профилями, или без них).
- **Удалить сегмент**: Переместить сегмент в корзину.



Динамические сегменты

Динамический сегмент — это сегмент, который отбирает пользователей непосредственно в момент каждого его использования.

Вы можете работать с таким сегментом, не сохраняя его, если вам нужна однократная быстрая выборка. Если вы решите сохранить сегмент, его можно будет использовать в рассылках, сценариях и т.д. При каждом запуске сегмент будет **заново** отбирать профили по указанным условиям.

	Чтобы начать работу, перейдите в раздел **Данные — Сегменты — Динамические сегменты**.

Создание динамического сегмента

В списке динамических сегментов нажмите **+ Создать**. В открывшейся форме укажите **Имя** сегмента и нажмите на блок **“Выбрать источник”**. В боковой панели выберите одну **Базу профилей** — это обязательный шаг, без выбора базы создание сегмента невозможно:

INFO

Обратите внимание, что после выбора источника его изменить будет невозможно.

Настройка диапазона и условий

Здесь же можно задать **Диапазон**, который определяет часть базы, к которой будут применены условия сегментации:

По умолчанию установлен диапазон от **0% до 100%** от всей выбранной базы профилей. Профили отбираются пропорционально — один и тот же диапазон всегда выбирает одних и тех же пользователей.

Основной блок — **“Настройка условий”**. Для добавления правила нажмите кнопку **“+ Условие”**. В выпадающем меню доступны опции **“Добавить группу условий”** или **“Импортировать условия”** из другого сегмента. После выбора откроется окно настройки условия:

Механика полностью идентична статическим сегментам: доступны все те же категории фильтров по данным профиля, взаимодействиям и поведению в каналах. Условия внутри группы можно комбинировать по логике **“соблюдать для каждого из условий”** (И) или **“соблюдать для одного из условий”** (ИЛИ).

В блоке “Настройка условий” также доступны кнопки **“Удаление”** (очищает заданные правила) и **“Скрипт”** (открывает редактор для ручного написания условий в формате JSON).

После настройки условий в этом же окне можно активировать чекбокс **“Установить ограничения по максимальному количеству профилей в сегменте”** и указать нужное число. Это ограничение применяется к результату выборки **после** отбора по диапазону и условиям:

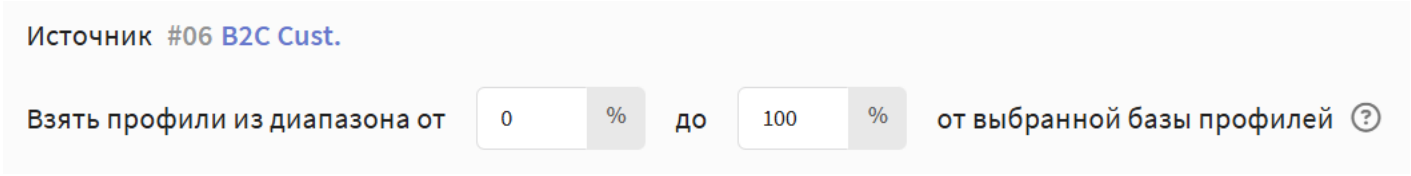
Подробнее о работе ограничения аудитории

В динамическом сегменте можно применять два типа ограничений, которые работают последовательно.

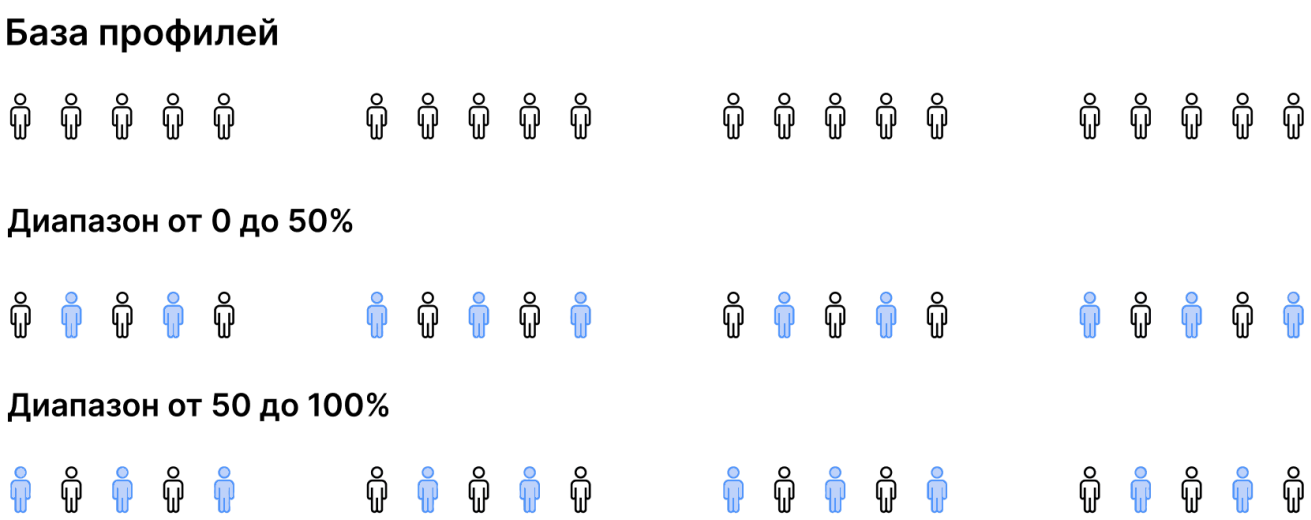
Диапазон определяет часть базы, к которой будут применены условия сегментации. По умолчанию диапазон равен 100%, т.е. если в базе 1000 профилей, то условия сегментации применятся ко всем профилям в базе.

Ограничения по максимальному количеству профилей — лимит профилей, которые могут быть включены в сегмент. Например, если 1000 профилей соответствуют условиям сегментации, а максимальное количество — 400, то в сегмент войдут первые 400 профилей из выборки.

Ограничения, заданные с помощью **диапазона**, распространяются на всю базу. Ограничения, заданные с помощью **Ограничения по максимальному количеству профилей**, накладываются на результат выборки сегмента.



Платформа закрепляет диапазон за конкретными пользователями: один и тот же диапазон всегда отбирает **одних и тех же людей** из базы, даже если вы используете его в разных сегментах. Это означает, что выборка **не случайна** — каждому пользователю сопоставляется числовое значение. Профиль входит в сегмент, если рассчитанное для него значение попадает в выбранный диапазон. Благодаря этому можно создавать непересекающиеся выборки, назначая разные диапазоны (например, диапазон от 0 до 20 процентов и диапазон от 80 до 100 процентов никогда не пересекутся). При добавлении новых пользователей в базу они также могут попасть в заданный диапазон с той же вероятностью — например, с 1/5, если выбран диапазон в 20%.



Работа с аудиторией

Быстрый сегмент

До нажатия кнопки “Сохранить” динамический сегмент можно использовать как быстрый сегмент — например, экспортировать профили или скопировать их в статический сегмент без необходимости сохранения самого сегмента.

После добавления хотя бы одного условия в правом верхнем углу, рядом с полями для тегов и групп доступа, становятся доступны дополнительные действия для работы с текущей выборкой профилей:

- Скопировать профили в базу профилей;
- Скопировать профили в статический сегмент;
- Экспортировать профили;
- Массовое редактирование профилей;
- Добавить профили в сценарий.

Предварительный расчёт

Также после настройки условий можно сразу оценить размер будущей аудитории. Нажмите кнопку **“Рассчитать”** в блоке расчёта профилей — система обработает базу данных и покажет, сколько профилей им соответствует:

Чтобы увидеть сами профили, нажмите **“Показать”** — откроется таблица с первыми 100 профилями из рассчитанной выборки:

После расчёта количество профилей можно вручную обновить, используя соответствующую кнопку рядом со счётчиком.

Заполните остальные поля в форме: при необходимости укажите **Описание**, **Теги** и **Группы доступа**. Затем нажмите **“Сохранить”**. После сохранения сегмент автоматически рассчитает свою аудиторию в первый раз.

INFO

Если при сохранении сегмента не задано ни одного условия, в сегмент автоматически попадут все профили из выбранной базы данных в рамках установленного диапазона.

Зависимые объекты

Если динамический сегмент используется в других сущностях платформы (например, в рассылках или сценариях), ниже журнала обновлений автоматически появится блок **“Зависимые объекты”**. Этот блок показывает, где именно задействован ваш сегмент:

Управление сегментами из общего списка

В строке с нужным динамическим сегментом в общем списке доступны следующие действия для работы с динамическими сегментами:

- **Копировать профили из сегмента**: Скопировать список профилей из текущего сегмента в другой выбранный сегмент.
- **Обновить профили**: Массово изменить данные профилей в сегменте (например, обновить поля, добавить подписку).
- **Экспортировать профили**: Выгрузить данные профилей, входящих в сегмент, во внешний файл.
- **Экспортировать историю профилей**: Выгрузить историю взаимодействий и изменений для профилей из сегмента.
- **Аналитика**: Перейти к аналитическим отчётам по данному сегменту.
- **Клонировать сегмент**: Клонировать условия сегмента, теги и группы доступа, и Базу профилей.
- **Удалить сегмент**: Переместить сегмент в корзину.

Обновляемые сегменты

Обновляемые сегменты пересчитываются автоматически в заданное время. Они сохраняют список профилей до следующего обновления, а при пересчете включают только тех, кто соответствует условиям на данный момент. Используйте их, если нужно регулярно обновлять аудиторию по заранее заданным критериям — например, для отправки напоминаний новым подписчикам или формирования списков активных клиентов.

INFO

Сегмент **нельзя наполнять вручную или через API**. Он наполняется **только** согласно расписанию или при ручном пересчёте.

Создание сегмента

	Чтобы начать работу, перейдите в раздел **Данные — Сегменты — Обновляемые сегменты**.

В списке обновляемых сегментов нажмите **+ Создать**. В открывшейся форме укажите **Имя** сегмента — это обязательное поле. Затем выберите источник сегмента (Базу данных). Без выбора базы создание сегмента невозможно:

Настройка условий

После выбора базы откроется блок **“Условия и расписание”**. Механика настройки условий полностью совпадает с динамическими и статическими сегментами. Вы можете задать Диапазон, который определяет часть базы для применения условий.

Для добавления правила нажмите кнопку **“+ Условие”**. В выпадающем меню доступны опции **“Добавить группу условий”** и **“Импортировать условия”**. После настройки условий внутри группы их можно комбинировать по логике **“соблюдать для каждого из условий”** (И) или **“соблюдать для одного из условий”** (ИЛИ):

В этом же блоке доступны кнопки **“Удаление”**, которая очищает заданные правила, и **“Скрипт”**, открывающая редактор для ручного написания условий.

После настройки условий можно задать ограничения, активировав чекбокс **“Установить ограничения по максимальному количеству профилей в сегменте”**, чтобы задать лимит. Это ограничение применяется к результату выборки **после** отбора по диапазону и условиям.

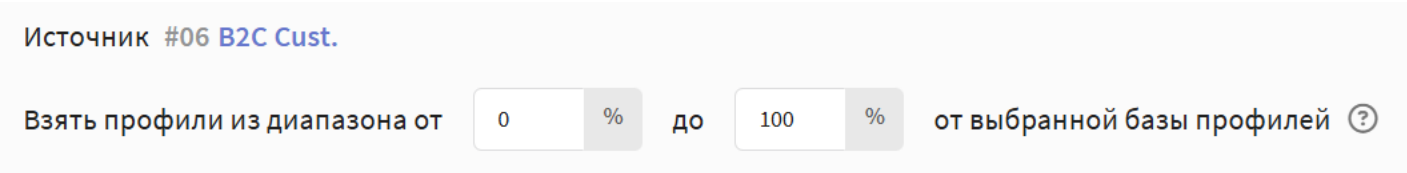
Подробнее о работе ограничения аудитории

В обновляемом сегменте можно применять два типа ограничений, которые работают последовательно.

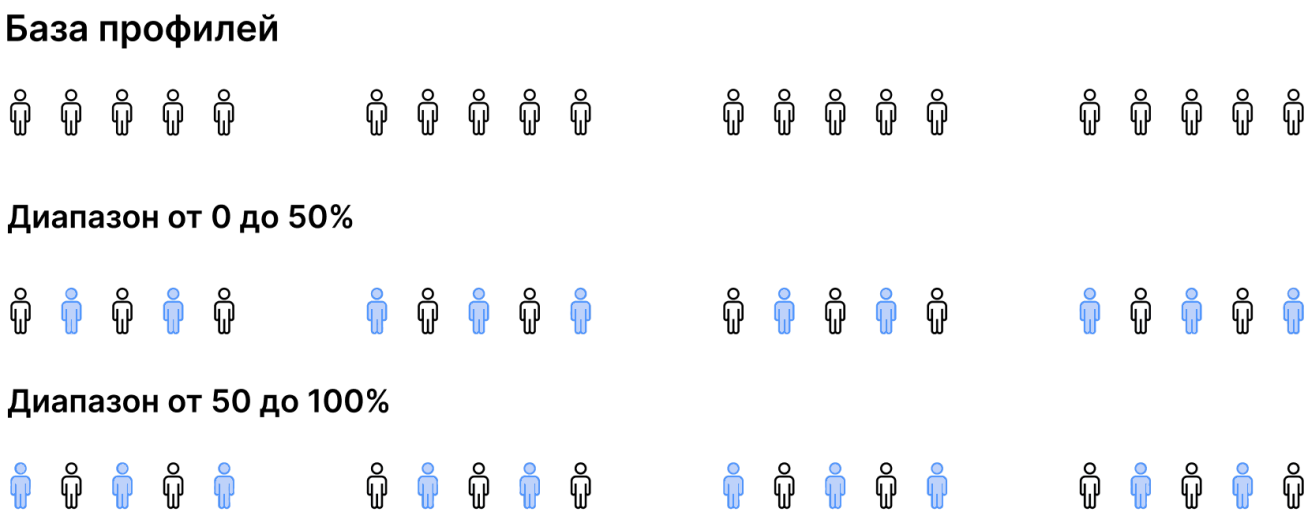
Диапазон определяет часть базы, к которой будут применены условия сегментации. По умолчанию диапазон равен 100%, т.е. если в базе 1000 профилей, то условия сегментации применятся ко всем профилям в базе.

Ограничения по максимальному количеству профилей — лимит профилей, которые могут быть включены в сегмент. Например, если 1000 профилей соответствуют условиям сегментации, а максимальное количество — 400, то в сегмент войдут первые 400 профилей из выборки.

Ограничения, заданные с помощью **диапазона**, распространяются на всю базу. Ограничения, заданные с помощью **Ограничения по максимальному количеству профилей**, накладываются на результат выборки сегмента.



Платформа закрепляет диапазон за конкретными пользователями: один и тот же диапазон всегда отбирает **одних и тех же людей** из базы, даже если вы используете его в разных сегментах. Это означает, что выборка **не случайна** — каждому пользователю сопоставляется числовое значение. Профиль входит в сегмент, если рассчитанное для него значение попадает в выбранный диапазон. Благодаря этому можно создавать непересекающиеся выборки, назначая разные диапазоны (например, диапазон от 0 до 20 процентов и диапазон от 80 до 100 процентов никогда не пересекутся). При добавлении новых пользователей в базу они также могут попасть в заданный диапазон с той же вероятностью — например, с 1/5, если выбран диапазон в 20%.



Настройка расписания

В блоке **“Условия и расписание”** настройте параметры автоматического обновления. Укажите точное **время** запуска (ЧЧ:ММ) и выберите **часовой пояс**. Затем задайте **периодичность** пересчёта из доступных вариантов: ежедневно, в определённую дату, по времени (дням недели, дням месяца и т. д.):

Опционально можно активировать функцию **“Деактивировать расписание после указанной даты”**, чтобы автоматические обновления прекратились в заданный день.

После настройки всех параметров нажмите **“Сохранить”**. Первый расчёт запустится в ближайшее запланированное время.

Работа с созданным сегментом

После создания и сохранения сегмента откроется его страница. Теперь в правом верхнем углу, рядом с полями для тегов и групп доступа, становятся доступны дополнительные действия для работы с текущей выборкой профилей:

- Скопировать профили в базу профилей;
- Скопировать профили в статический сегмент;
- Экспортировать профили;
- Массовое редактирование профилей;
- Добавить профили в сценарий.

> **INFO**
>
> Действия с профилями станут доступны после первого обновления сегмента

Управление автоматическим обновлением

В блоке **“Автоматическое обновление сегмента”** отображается текущее расписание. Отсюда можно **“Остановить”/“Запустить”** (в зависимости от текущего состояния), **Перенастроить расписание** или **Запустить ручной пересчёт** вне запланированного времени.

> **CAUTION**
>
> Ручной пересчёт и редактирование расписания обновлений недоступны в момент выполнения обновления сегмента (автоматического или ручного).

Ниже находится спойлер **“Показать условия обновления”**. При раскрытии отображаются все заданные условия, диапазон и лимит. Их можно **отредактировать**, **посмотреть скрипт** или **удалить**.

Анализ состава и истории

После первого обновления на странице сегмента станет доступна **Таблица профилей** с текущим составом:

Доступны следующие действия для работы с этой таблицей:

- **Поиск**

  : найти конкретный профиль по имени, email или другим данным.
- **Экспортировать профили**

  : выгрузить отображаемый список во внешний файл.
- **Настройка отображаемых полей**

  : чтобы добавить или убрать колонки в таблице, перетащите нужное **Поле** из боковой колонки **“Добавить поле”** в область **“Отображаемые поля”** или обратно.

История обновлений

Блок **“История обновлений”** — график, визуализирующий изменение количества профилей в сегменте за выбранный период. Период можно задать с помощью календаря или готовых вариантов (сегодня, последние 7 дней, этот месяц и т.д.), а также сгруппировать данные по дням, неделям, месяцам, кварталам или годам. При

наведении на точку на графике отображается детализация по профилям: добавлено или удалено автоматически (пересчёт был запущен по расписанию) и вручную (персчёт был запущен вручную):

Журнал

Блок **“Журнал”** — таблица с детальной информацией по каждому запуску пересчёта. Для каждой операции указывается дата и время старта и завершения, тип (Ручное / Автоматическое), инициатор (пользователь или “Система”), количество добавленных и удалённых профилей, а также статус (`В процессе`, `Завершено`, `Ошибка`).

Зависимые объекты

Если сегмент используется в других сущностях платформы (рассылки или сценарии), ниже журнала обновлений появится блок **“Зависимые объекты”**:

Управление сегментами из общего списка

В строке с нужным обновляемым сегментом в общем списке доступны следующие действия:

- **Копировать профили из сегмента**: Скопировать список профилей из текущего сегмента в выбранный статический сегмент.
- **Обновить профили**: Массово изменить данные профилей в сегменте (например, обновить поля, добавить подписку).
- **Экспортировать профили**: Выгрузить данные профилей, входящих в сегмент, во внешний файл.
- **Экспортировать историю профилей**: Выгрузить историю взаимодействий и изменений для профилей из сегмента.
- **Аналитика**: Перейти к аналитическим отчётам по данному сегменту.
- **Клонировать сегмент**: Клонировать условия сегмента, теги и группы доступа, базу профилей и расписание.
- **Удалить сегмент**: Переместить сегмент в корзину.

Сегментация профилей по внешним данным

Правила сегментации могут обращаться за данными для выборки профилей к внешнему источнику. В качестве такого хранилища данных могут выступать:

- API веб-сервиса
- Загруженный файл
- Область ввода данных внутри правила

Это позволяет автоматизировать сегментацию по многим полям, а также уменьшить количество однообразных условий в сегменте.

> **TIP**
>
> Сегментация профилей подписчиков по внешним SQL базам данных описана в отдельной статье.

HTTP запросы к API внешнего сервиса

При выборе в правиле сегментации оператора **(Не) В ответе на post url в поле** вы можете проводить выборку подписчиков на основе ответа от сервиса. Платформа принимает данные в форматах **JSON** и **XML**, а также записанные текстом с переносом строк (**PIPE**).

Пример правила сегментации для JSON

Если внешний сервис отдаёт данные в формате JavaScript Notation Object, будет ожидаться заголовок MIME типа

`"Content-Type": "application/json"`

и данные в следующей нотации:

```
{
  "segment": {
    "emails": [
      "john@example.org",
      "suzy@example.com",
      "dick@example.net",
      "andy@example.edu"
    ]
  }
}
```

В правиле сегментации необходимо будет указать URL-адрес эндпойнта API и путь до необходимого поля в JSON объекте:

Пример правила сегментации для XML

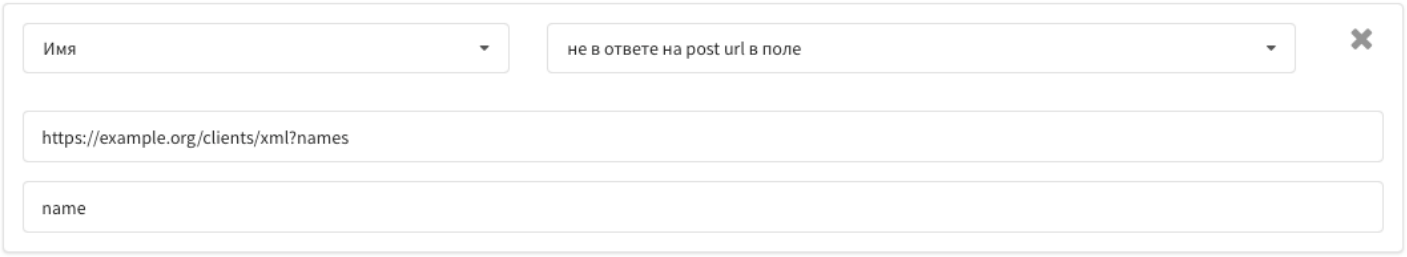
Если внешний сервис отдаёт данные в формате XML, будет ожидаться заголовок MIME типа

`"Content-Type": "application/xml; charset=UTF-8"`

и данные в следующей нотации:

```
<names>
  <name>Andy</name>
  <name>Dave</name>
  <name>John</name>
  <name>Suzy</name>
</names>
```

В правиле сегментации необходимо будет указать URL адрес эндпойнта API и путь до необходимого поля в XML объекте:



Пример правила сегментации для PIPE

PIPE формат представляет собой данные, разделённые переносом строки (записанные одним столбцом). Если внешний сервис отдаёт данные в формате PIPE, будет ожидаться заголовок MIME типа

`"Content-Type": "text/plain; charset=UTF-8"`

и данные в следующей нотации:

```
+78002000100
+78004000200
+78006000400
+78008000600
```

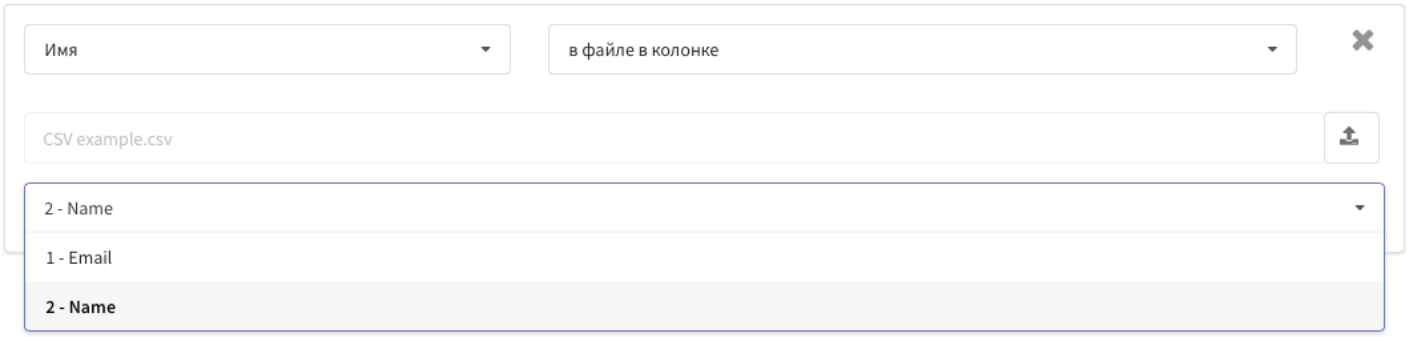
В правиле сегментации необходимо будет указать URL адрес эндпойнта API:

Загрузка файла

При выборе правила сегментации **(Не) В файле в колонке** появится форма для загрузки табличного файла в формате **CSV** (comma separated values):

```
Email,Name
john@example.org,John
suzy@example.com,Suzy
dave@example.net,Dave
andy@example.edu,Andy
```

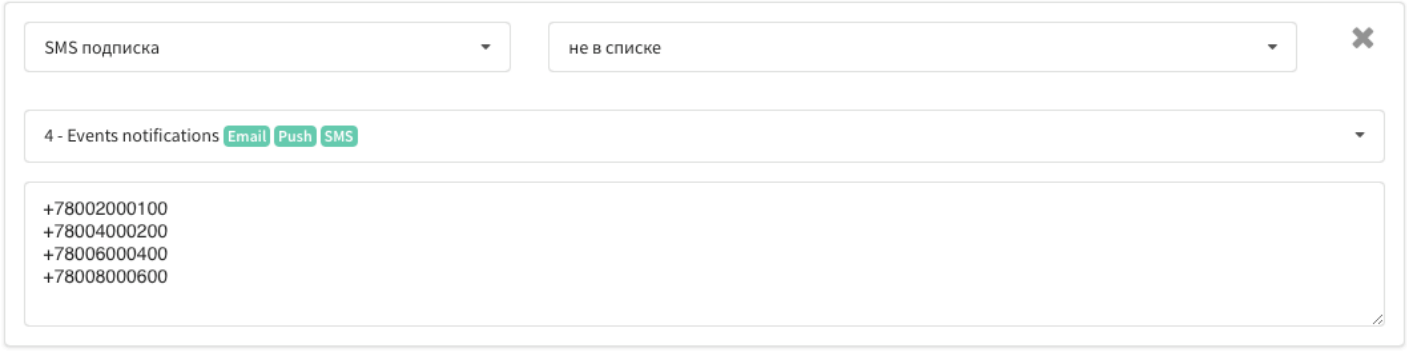
После загрузки файла вы сможете выбрать необходимую колонку из выпадающего меню:



Данные, вставленные из буфера

При выборе оператора сегментации **(Не) В списке** данные вставляются в **специальное поле** вручную в формате **PIPE**.

PIPE формат представляет собой данные, разделённые переносом строки (записанные одним столбцом).



Сегментация профилей по внешним SQL-таблицам

Возможность подключения внешних SQL баз данных позволяет более гибко подходить к вопросу сегментации профилей.

- Нет необходимости держать все данные в Altcraft MP
- Соответственно нет необходимости в регулярной синхронизации и данные всегда актуальны
- Можно использовать данные по сущностям, которые в Altcraft MP не предусмотрены
- Не зависит от платформы и системы которую вы используете для хранения данных, необходим лишь доступ для выполнения SQL запросов

Коннекторы для подключения к внешним базам создаются в панели администратора Altcraft.

TIP

Если вы пользуетесь **облачным решением** — передайте данные для подключения к базе данных нашей службе поддержки: team@altcraft.com. Там же можно уточнить по вопросам, связанным с возможностью подключения к другим внешним службам и БД.

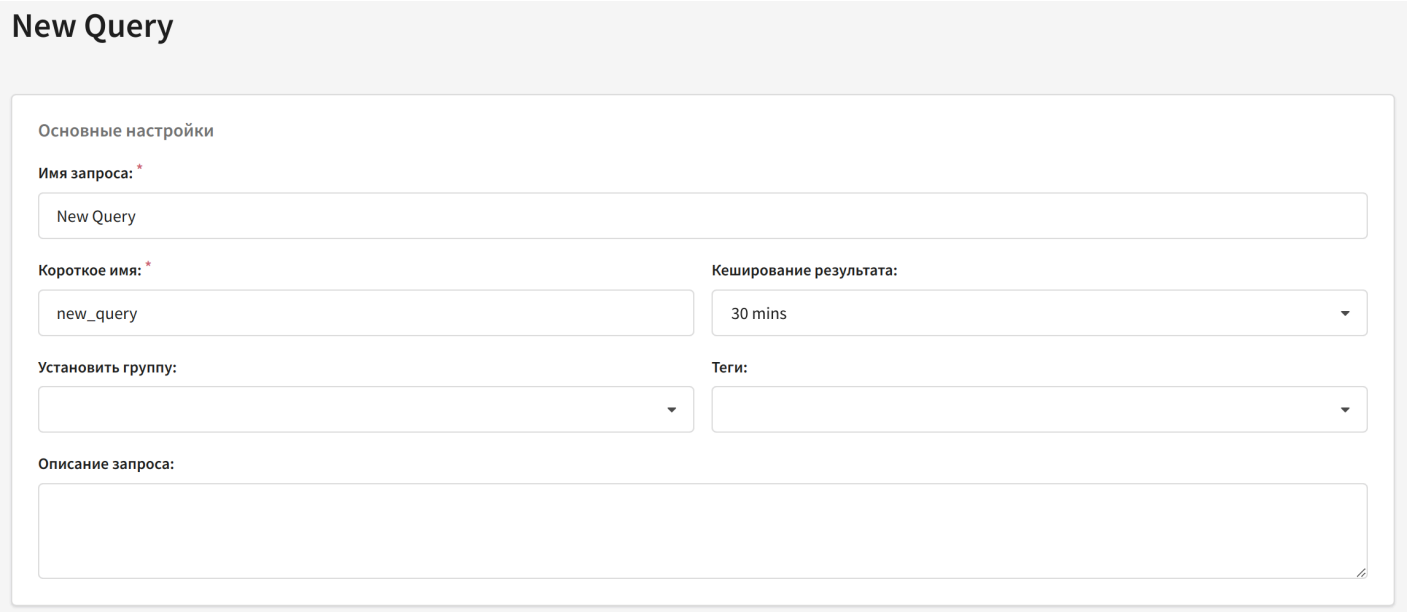
Создание запроса к базе данных

Запросы можно конфигурировать как в панели администратора, так и в панели пользователя системы. Разница в том, что в панели пользователя вы сможете сразу просматривать профили, которые были выбраны вашим запросом.

Редактирование запросов в панели пользователя можно запретить настройкой конфигурации в main.json если это необходимо. Для этого нужно будет установить параметр `"DISABLE_QUERY_EDITOR": true`.

	Запросы настраиваются в разделе **Данные —> Запросы сегментации**. Перейдите в раздел и нажмите кнопку **+ Создать**.

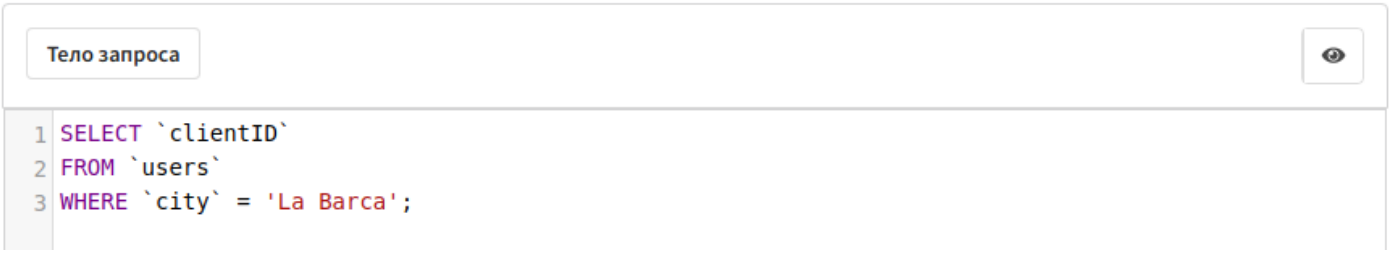
В основных настройках укажите **имя запроса**, **короткое имя** (уникально для каждого запроса, используется в переменных платформы и в API-запросах), срок **кэширования результата**, а также группу, теги и описание запроса:



INFO

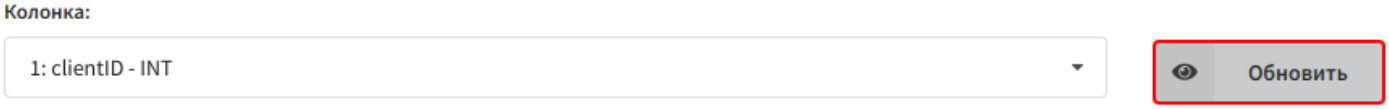
Обратите внимание:срок **кэширования резльтата** работает только при расчете сегментов. В сценариях автоматизации используется отдельный механизм кеширования с собственными настройками.

Чтобы запрос сегментации работал правильно, он должен выбирать колонку с данными, которые уже есть в Altcraft MP. Например, если в Altcraft есть идентификатор клиента в базе профилей, то необходимо сформировать такой запрос, чтобы на выходе получился именно этот идентификатор, как минимум в одной из колонок:



Эти запросы будут использоваться внутри сегмента. Можно построить неограниченное количество таких запросов для уточнения сегмента. В каждом таком запросе вы можете выбирать по одному или нескольким параметрам, комбинируя запрос, делая подзапросы, JOIN, UNION итд.

Обязательно выберите колонку, которая будет использоваться для поиска профилей, используйте кнопку обновить, чтобы получить новый список колонок по вашему запросу:

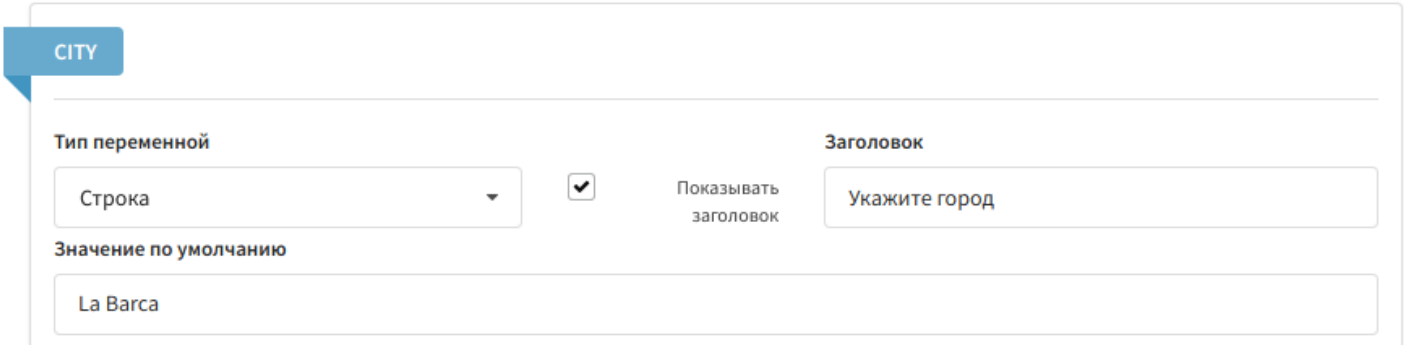


В запрос можно добавлять параметры, чтобы запрос изменялся уже при использовании внутри сегмента. Для этого используйте произвольное, но уникальное имя вашего параметра в фигурных скобках, например `{CITY}`:

Тело запроса

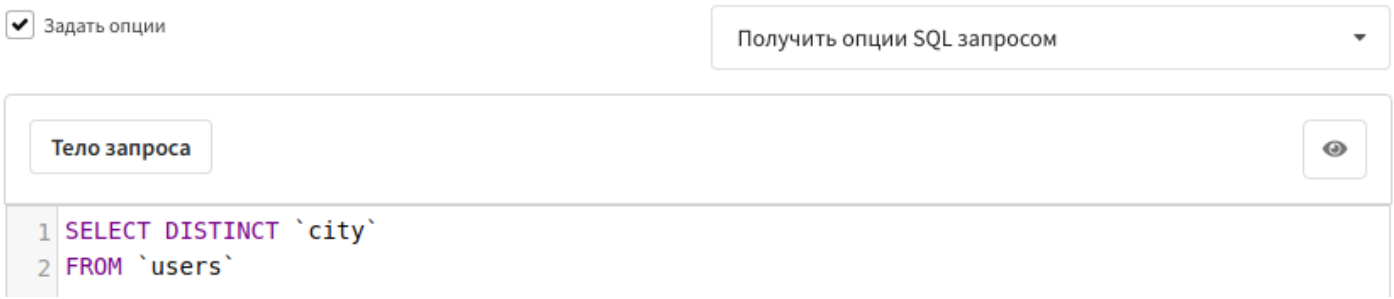
```
SELECT `clientID`
FROM `users`
WHERE `city` = '{CITY}';
```

Как только вы его укажете в запросе, появится форма с настроками параметра:

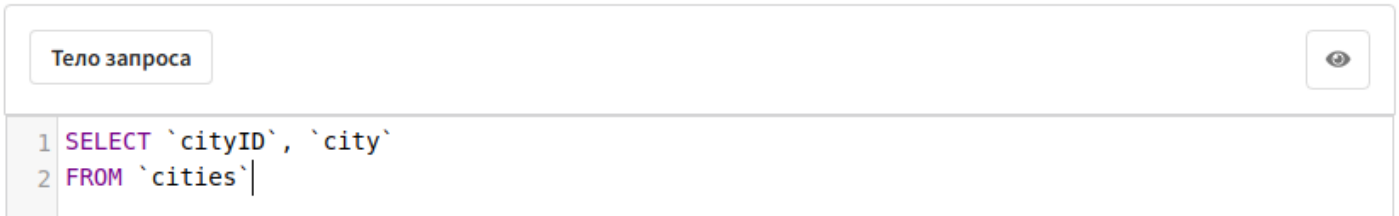


Здесь задается тип поля, его значение по умолчанию и заголовок, который будет отображен при создании сегмента.

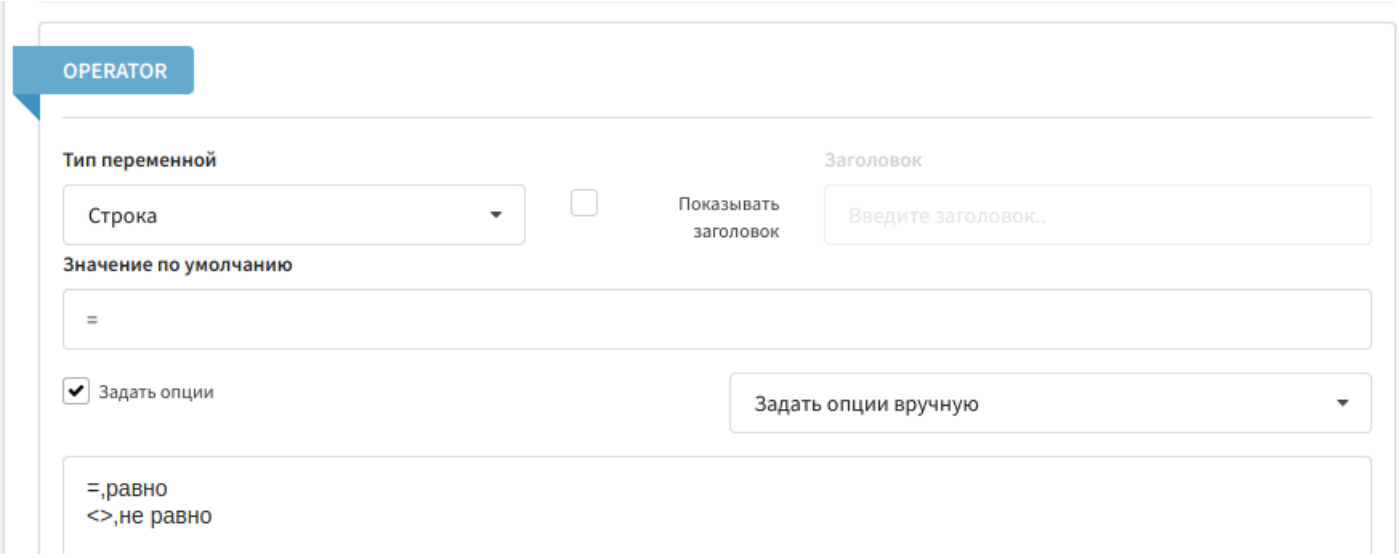
Если мы хотим, чтобы в сегменте поле выбиралось из списка, мы используем настройку **Задать опции**. При ее включении выбираем, будем ли мы перечислять список опций или тоже выберем его из SQL:



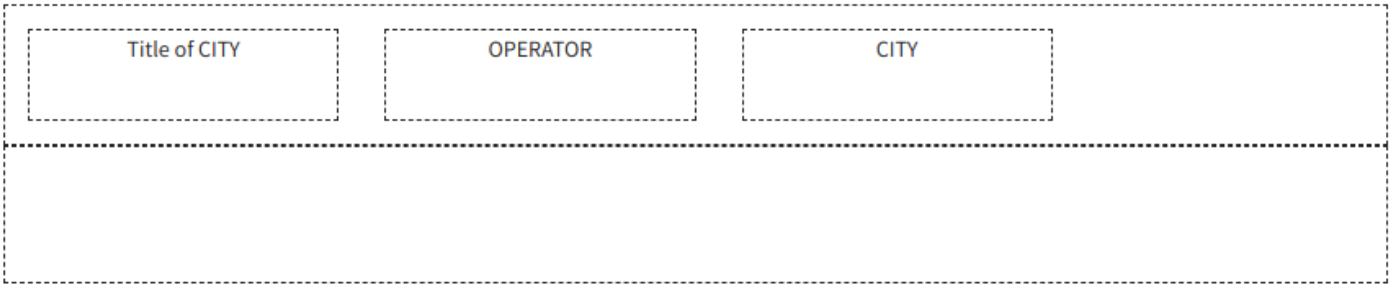
При этом, если применяются идентификаторы, то SQL запрос должен возвращать две колонки, первая используется как идентификатор, а вторая будет отображаться пользователю:



В случае с заданным списком идентификатор и опции можно перечислить через запятую:

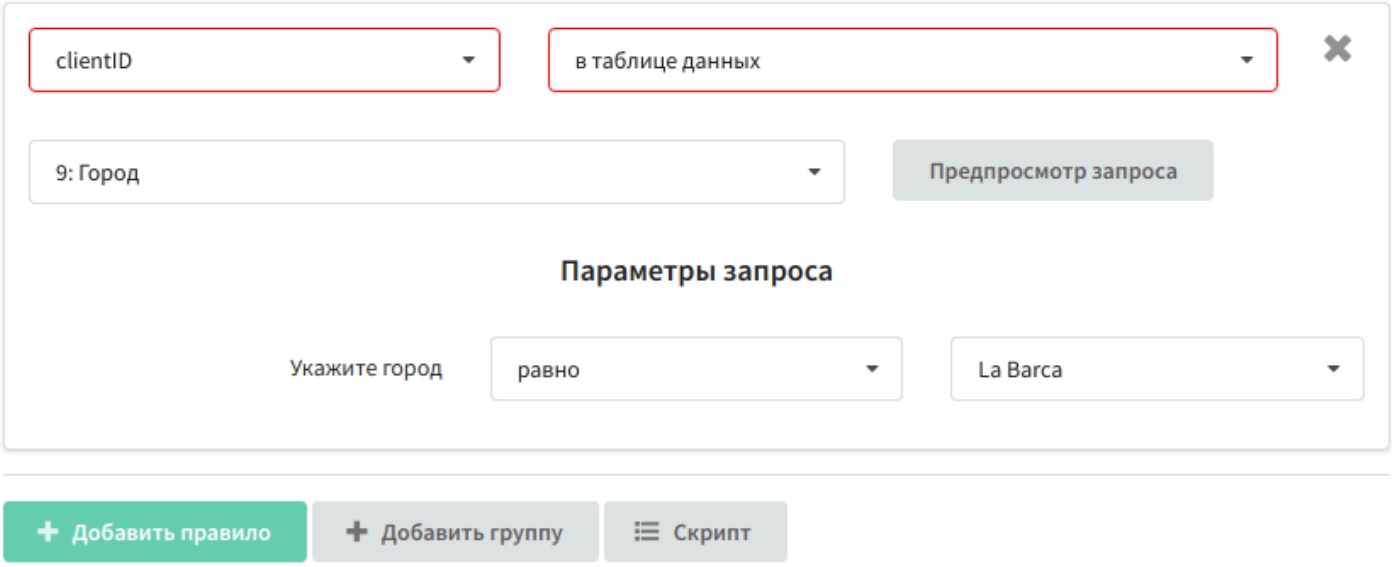


Для всех параметров конфигурируется порядок их отображения при построении сегмента. Порядок можно менять простым перетаскиванием элементов. Заголовки позиционируются отдельно, для более гибкого отображения.



Воспользуйтесь кнопкой предпросмотра выборки из базы данных, для того чтобы посмотреть как форма запроса будет выглядеть в сегменте и как будет выглядеть тестовая выборка:

Укажите базу данных и поле профиля для выборки, а также колонку если их несколько. В разделе **Параметры запроса** сформируется форма в том виде, как она потом будет выглядеть в сегменте. Нажав кнопку **Применить**, вы запустите запрос и он сразу покажет те профили, которые удовлетворили условию запроса:



После сохранения запроса его можно будет использовать в сегментах.

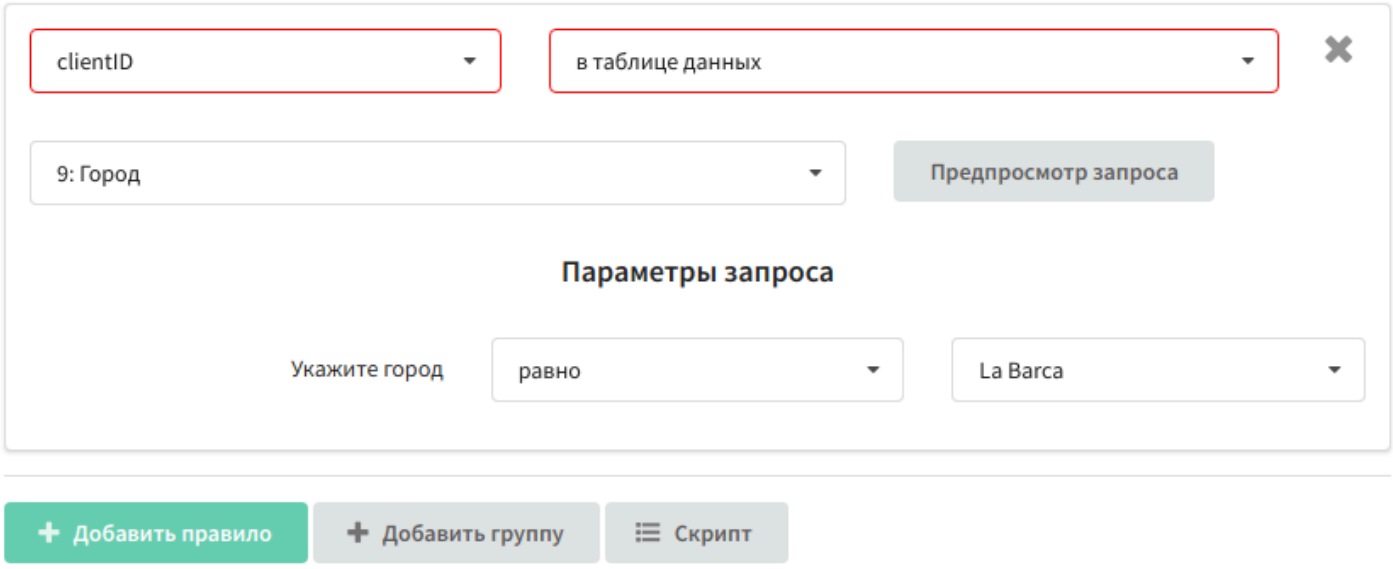
Использование запроса к внешней базе в сегменте

Перейдите в раздел **Данные —> Сегменты**. Создайте новый сегмент или откройте существующий для редактирования.

Добавьте новое правило и выберите поле-идентификатор или поле по которому осуществляется поиск профиля запросом. Затем выберите один из вариантов:

- **В таблице данных** — если хотите включить всех кто подойдет под данное условие.
- **Не в таблице данных** — если хотите включить всех кто не подойдет под данное условие.

Затем выберите ваш запрос к таблице и заполните форму параметров запроса:



Теперь можете сохранить сегмент.

Условия **в таблице данных \ не в таблице даных** можно неоднократно применять в сегменте, в том числе внутри групп, обьединяя их по *И*, если хотите чтобы каждое из них удовлетворяло условиям или по *ИЛИ*, если хотите чтобы любое из них удовлетворяло условиям.

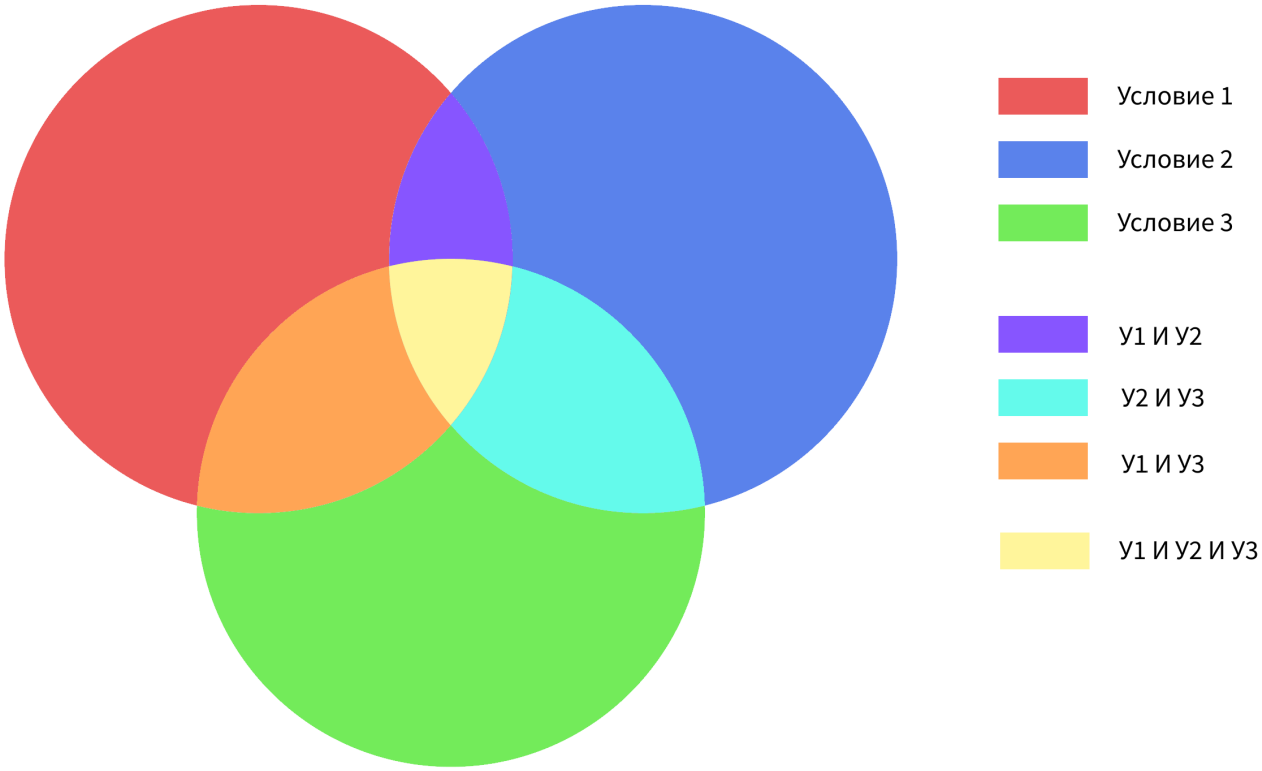
Конечно, в сегментах с запросами можно обьединять любые условия: по полям в Altcraft MP, по действиям в сообщениях, по достижению целей итд.

Логические операторы “И” и “ИЛИ”

Сегментация в платформе строится на основе условий. Зачастую, одного условия недостаточно для того, чтобы вычислить необходимую для той или иной рассылки аудиторию. В таких случаях необходимо указать два или более условия отбора или **группу правил**. Условия связываются между собой с помощью логических операторов формальной логики — **“И”** и **“ИЛИ”**. В этой статье мы на примерах объясним, как работают условия с этими операторами.

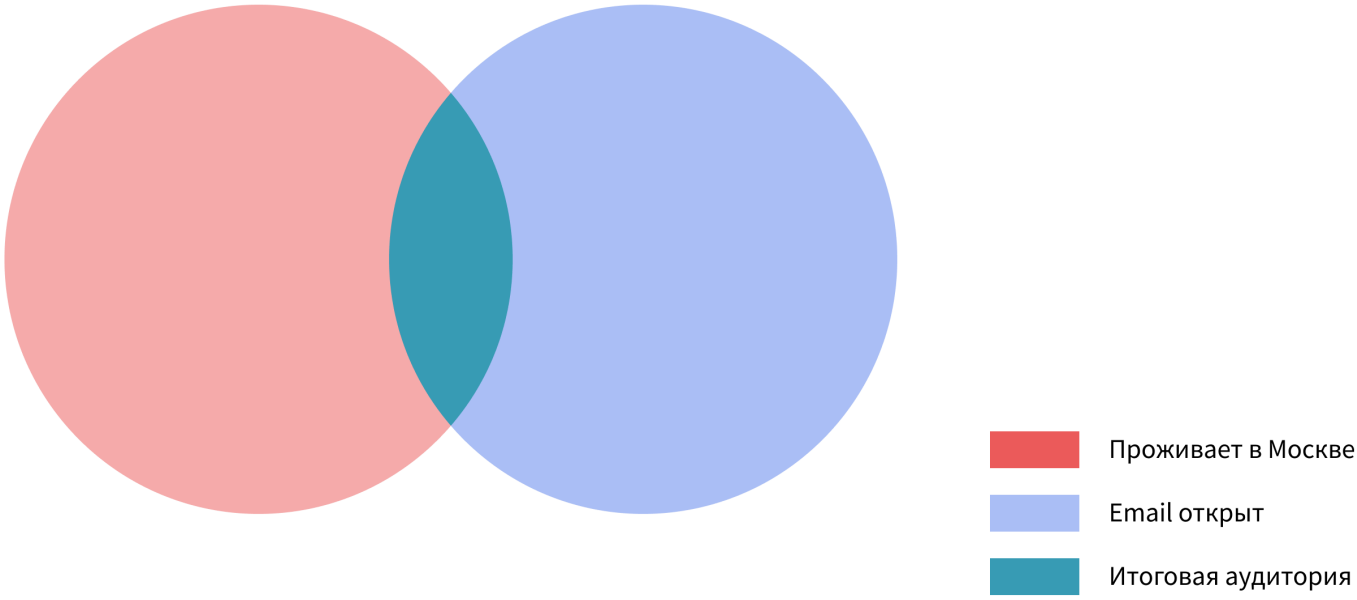
Оператор "И"

Оператор “И” будет выбирать тех подписчиков, для которых действительны **все** правила и группы правил в запросе. Этот оператор подразумевает под собой строгое соответствие условию у всех соединяемых элементов. Его можно представить схематично:

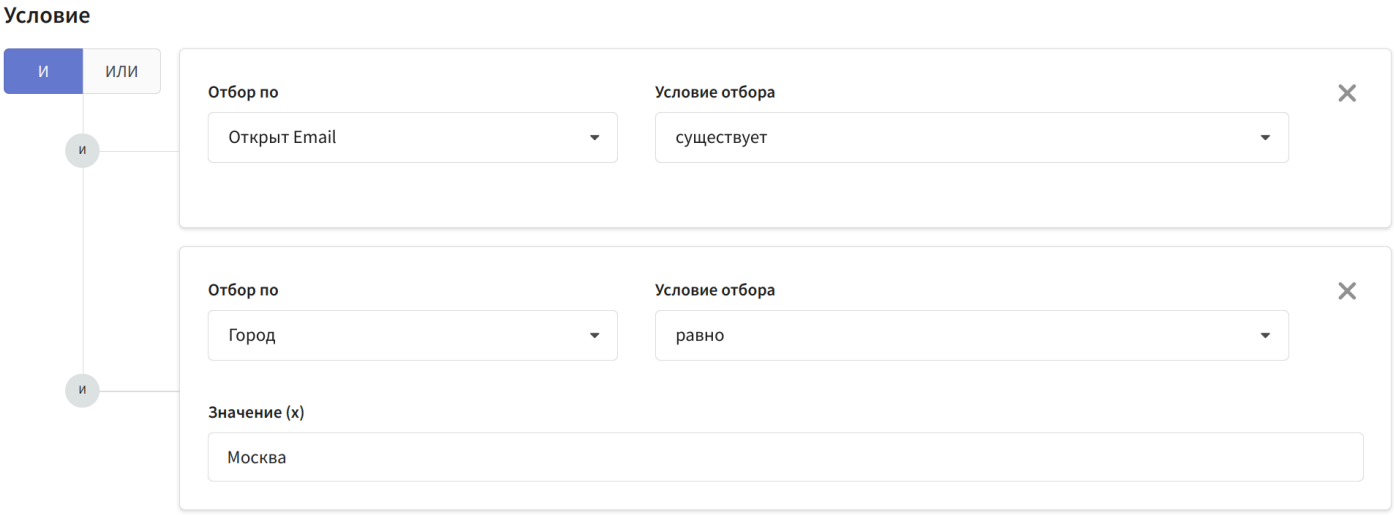


Пример использования

Смоделируем ситуацию. Для рассылки приглашения на промо-мероприятие нам необходимо отобрать **проживающих в Москве** подписчиков, которые **хотя бы раз открывали ваше Email-сообщение**:

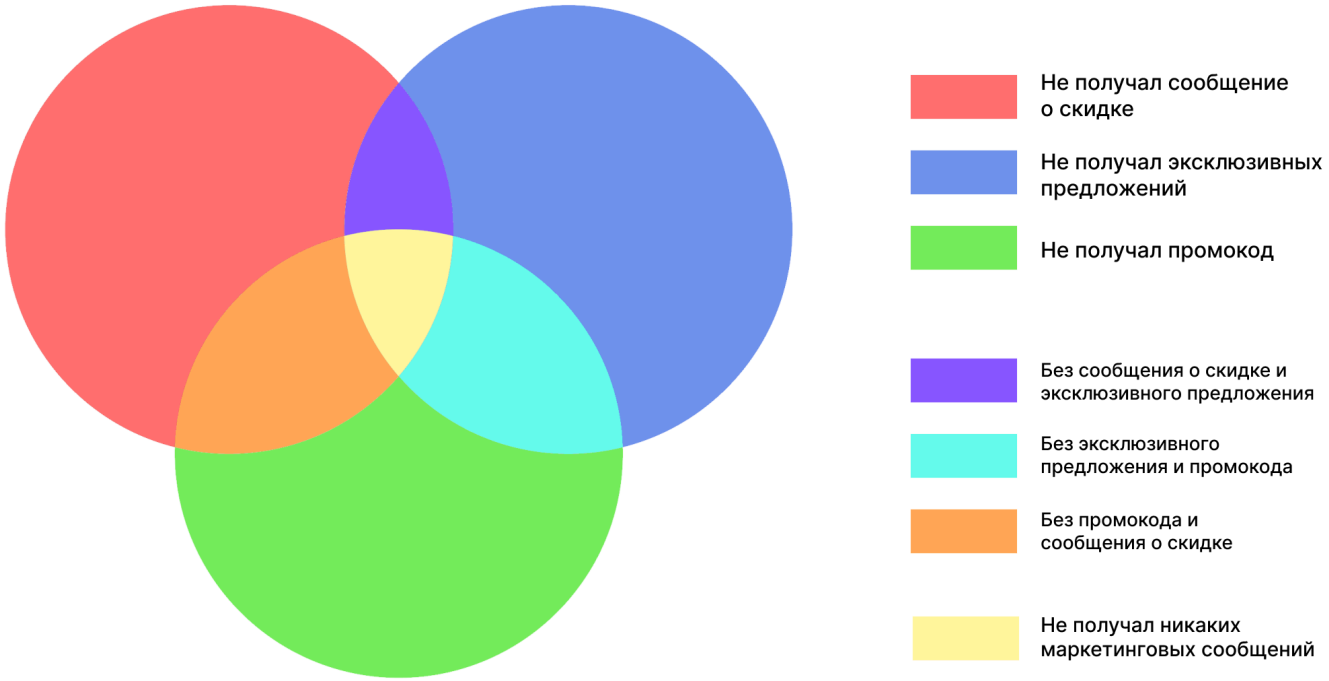


Условия будут выглядеть так:

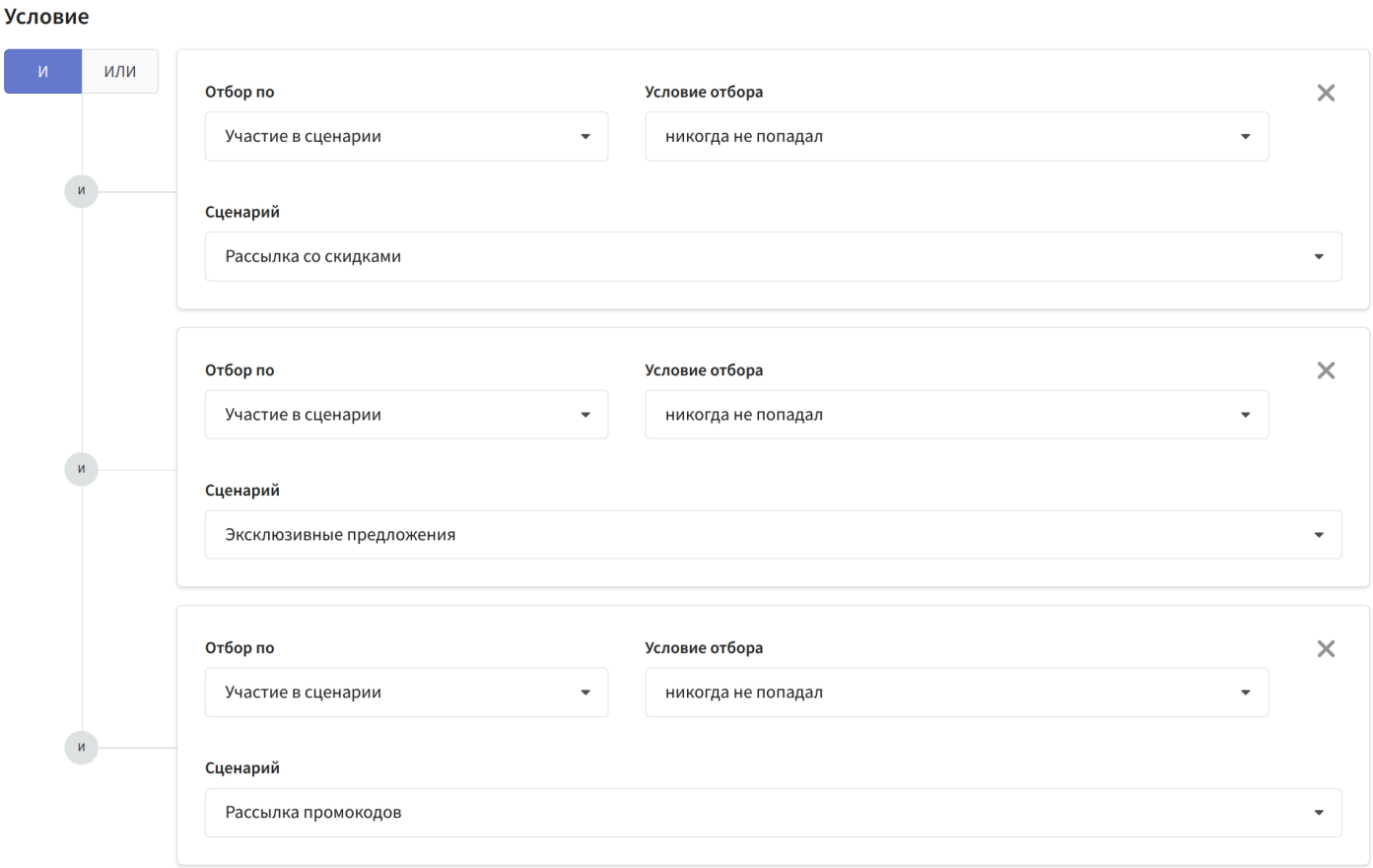


Так мы выберем тех клиентов, которые соответствуют обоим условиям.

Допустим, ранее вы создали несколько сценариев с различными предложениями. Теперь вам необходимо выделить тех, кто **еще не получал от вас сообщений**:



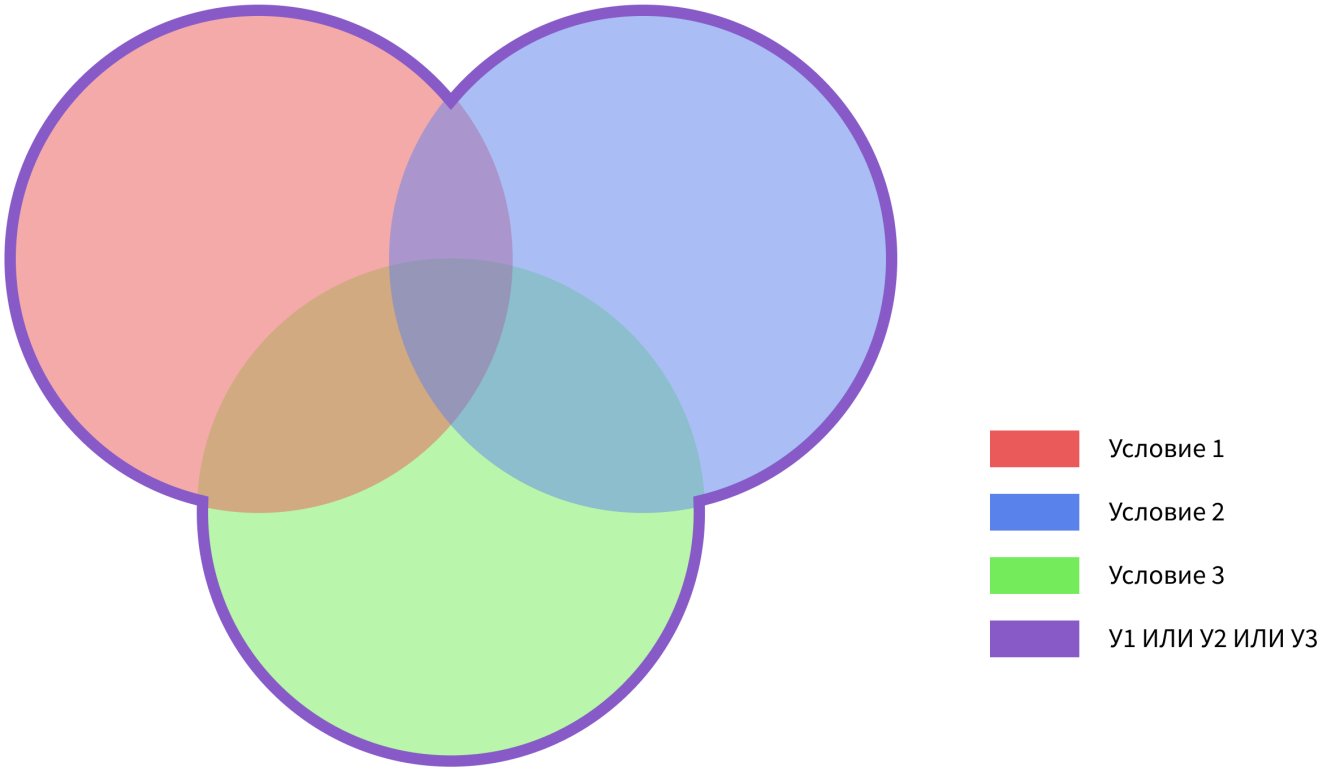
В этом случае условия также будут соединены оператором “И”:



В этот сегмент попадут те профили, которые не получали ни сообщений о скидках, ни писем об эксклюзивных предложениях, ни рассылок с промокодами.

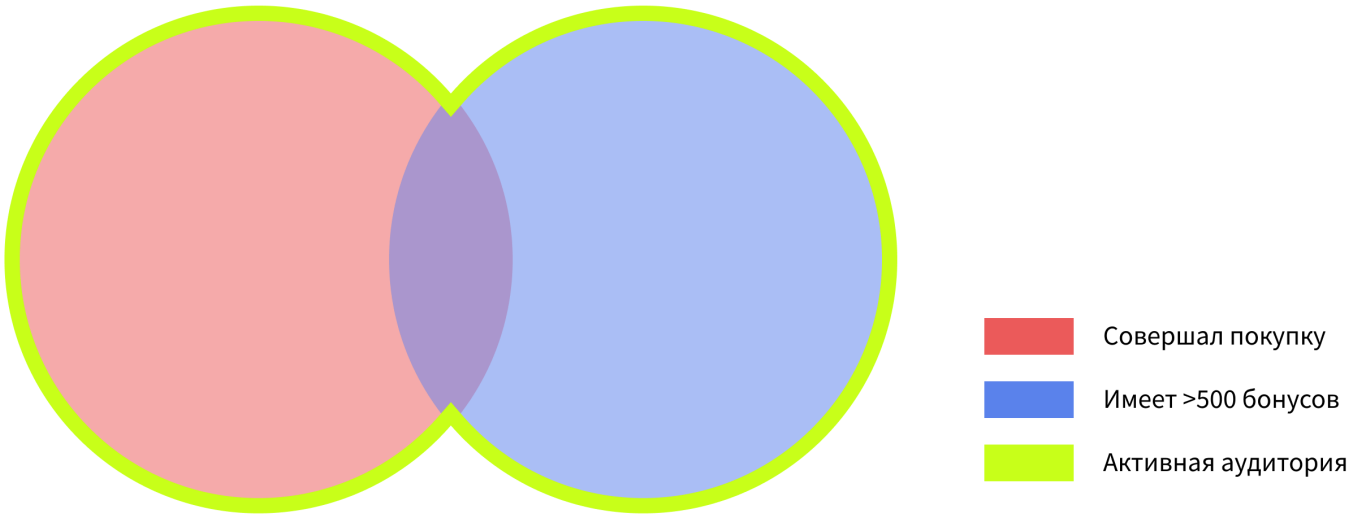
Оператор "ИЛИ"

Оператор “ИЛИ” выбирает те профили, для которых выполнимо **хотя бы одно** правило или группа правил. Если профиль удовлетворяет какому-то одному условию, то он попадет в такой сегмент. Схема его работы выглядит следующим образом:

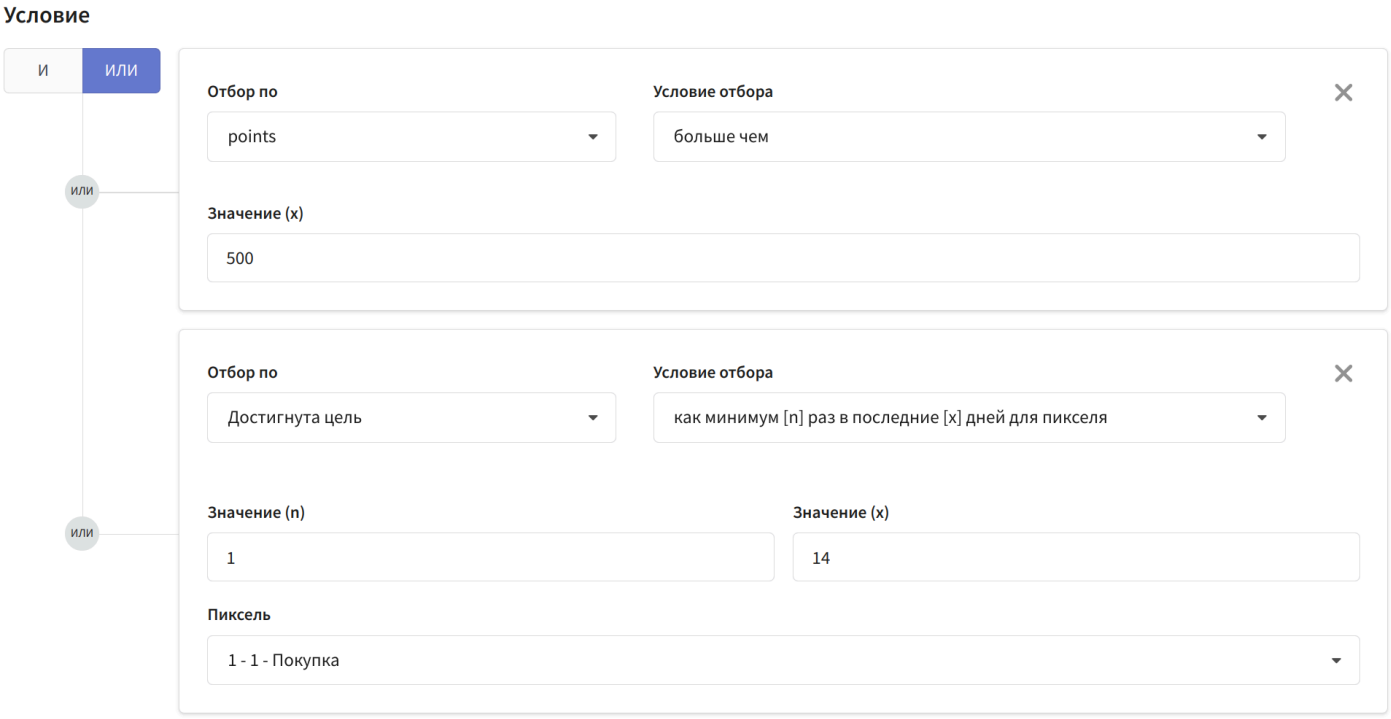


Пример использования

Предположим, что вам пришла идея о поощрении активных клиентов специальным промокодом. Активными будем считать тех, кто **либо совершил покупку в последние 2 недели, либо имеет более 500 бонусных баллов на своем счету**:

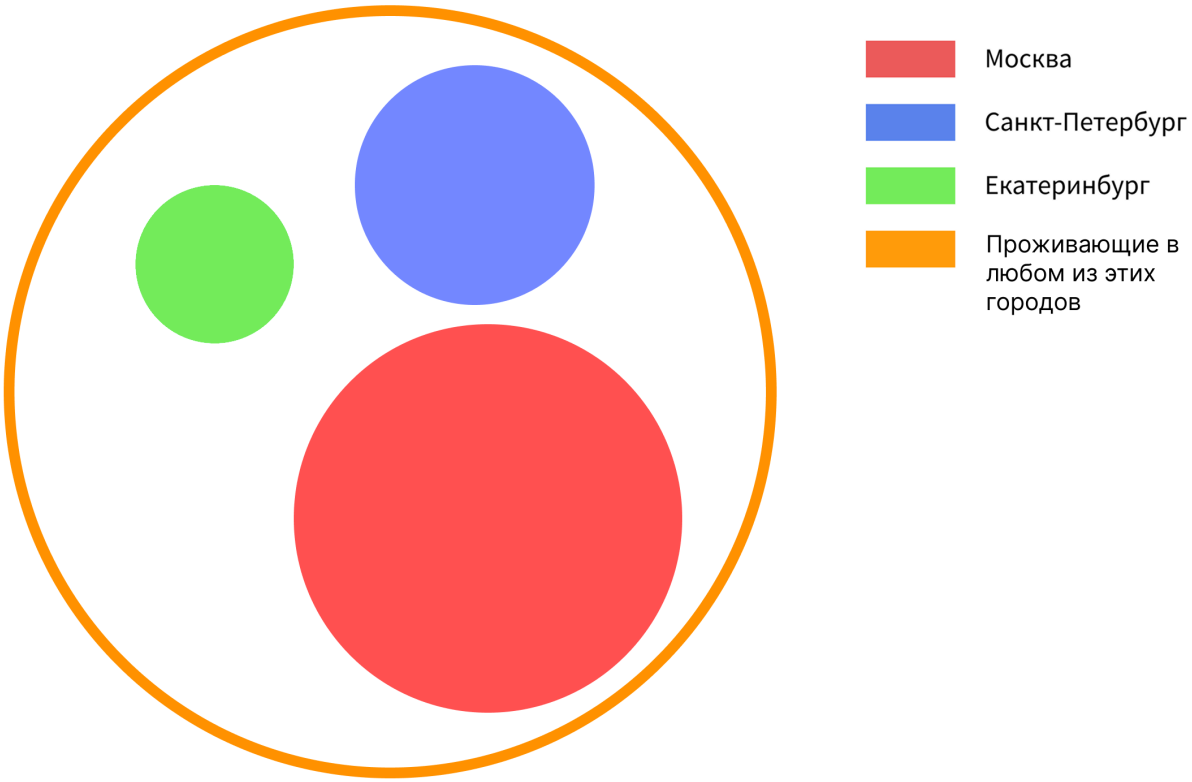


Составим условие:

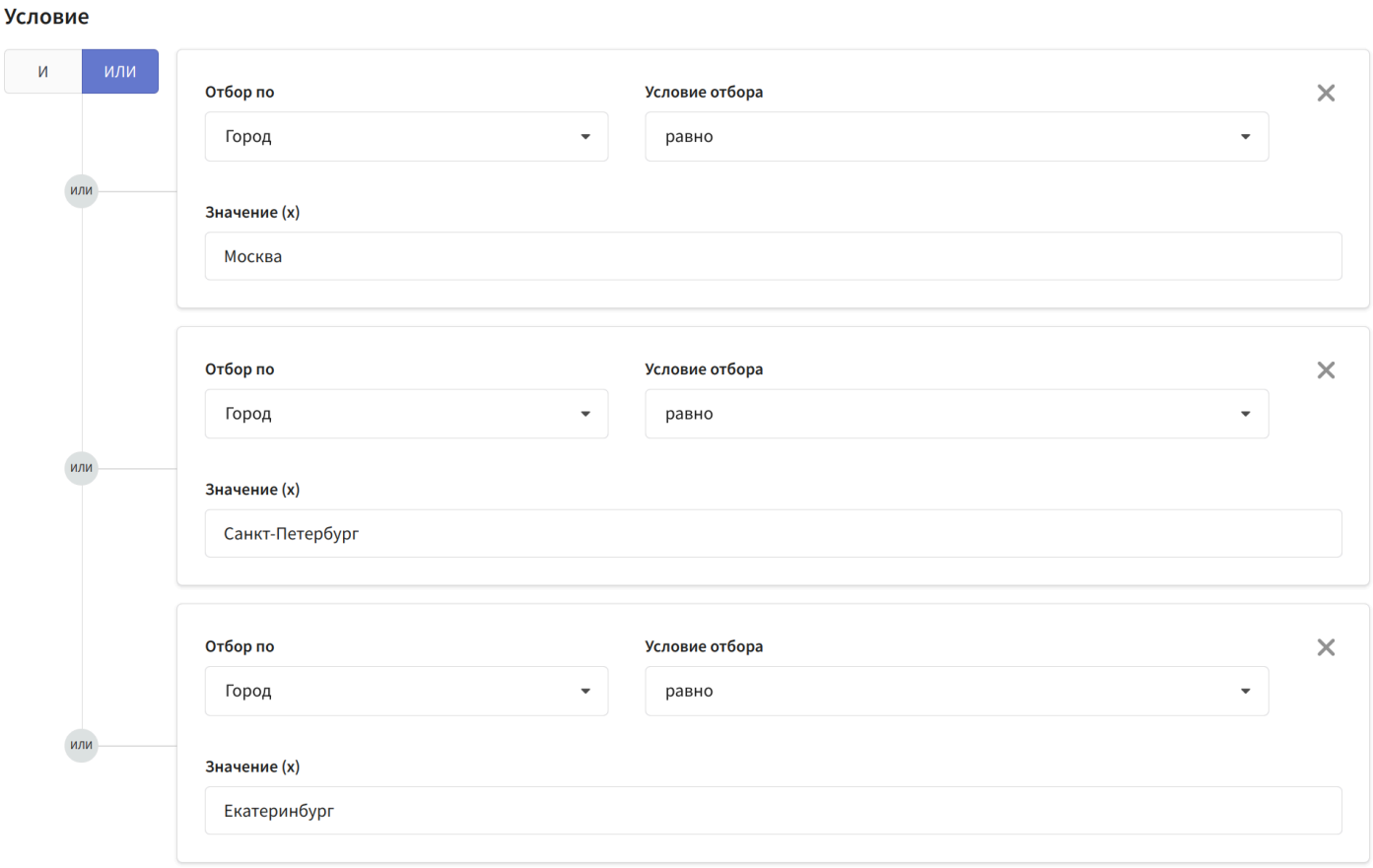


При выполнении любого из этих условий клиент получит промокод.

Если вам необходимо выбрать в качестве аудитории жителей **нескольких разных городов**, вы также будете использовать оператор “ИЛИ” при построении сегмента:



Условия будут выглядеть так:



INFO

Использование условия с оператором “ИЛИ” и отрицаниями может приводить к ошибкам. Обычно в таких условиях уже заложена логическая неточность.

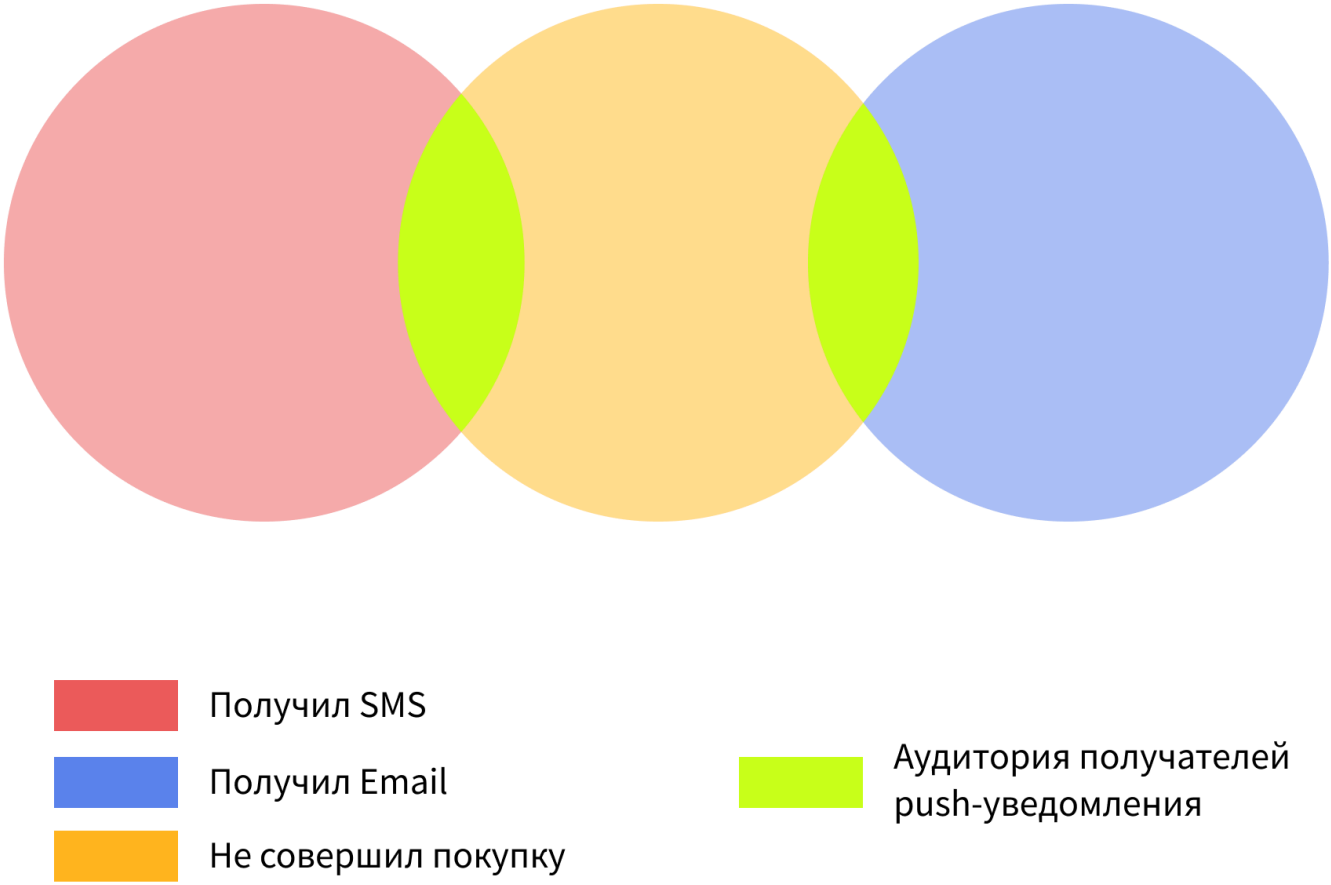
Не используйте такие конструкции — лучше заменить их эквивалентными условиями с “И”. Если всё же используете, обязательно **проверьте себя** с помощью наглядной логической схемы.

Группа правил

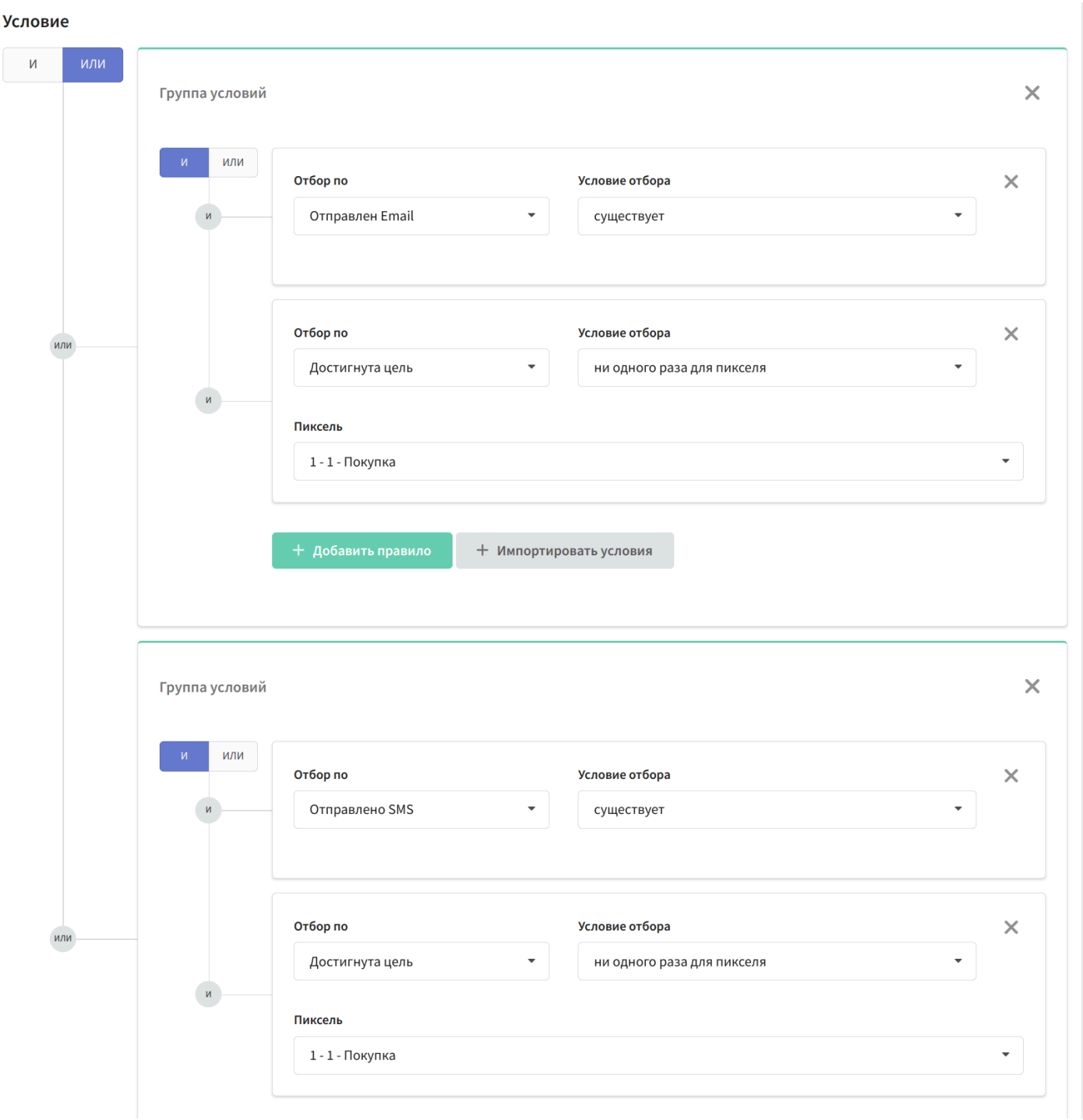
Вы можете связывать несколько правил одного уровня либо при помощи “ИЛИ”, либо при помощи “И”, но использовать их вместе не получится. Для построения сложных схем, необходимо использовать **группу правил**. Внутри группы правил можно использовать оператор, отличный от того, которым связаны условия другого уровня. Это позволяет построить более сложные по структуре аудитории для ваших маркетинговых рассылок.

Пример использования

Объясним на наглядном примере. Представим ситуацию, в которой вам нужно создать Viber™-рассылку для малоактивных покупателей — для тех, кто **получал и email-сообщение, и SMS-сообщение,** но покупку так и **не совершил**. Схематично аудитория получателей будет выглядеть так:



Для такого условия набор правил будет составлен следующим образом:



---

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Рекомендации по работе с сегментами

Сегменты — ключевой инструмент работы с данными в CDP-платформе. От их корректной настройки зависит качество выборок, аналитики и запуска маркетинговых рассылок. Однако на практике иногда возникают ситуации, когда сегмент не работает так, как ожидается: не пересчитывается, не обновляется или возвращает неожиданные результаты.

Чтобы упростить поиск и устранение таких проблем, в этой статье собраны **рекомендации по работе с сегментами**.

Использование условия с “НЕ” через “ИЛИ”

Часто при построении сегмента задаётся условие вида:

```
НЕ (имеет активный договор) ИЛИ НЕ (прошёл верификацию)
```

На первый взгляд может показаться, что такое условие исключит всех клиентов без договора и без верификации. На практике оно работает иначе: в сегмент попадут **все клиенты, кроме тех, у кого одновременно есть и активный договор, и пройдена верификация**.

Это значит, что условие исключает только тех, у кого **оба признака выполняются одновременно**, а не всех, кто обладает хотя бы одним из признаков.

Рекомендации:

- Если требуется отобрать только клиентов, у которых нет договора и нет верификации, используйте:

```
НЕ (имеет активный договор) И НЕ (прошёл верификацию)
```

- Для повышения читаемости всегда используйте скобки и избегайте длинных выражений с чередованием **НЕ** и **ИЛИ**.

Слишком большое количество условий с “НЕ”

Чрезмерное использование отрицаний в условиях сегментации усложняет логику и повышает риск ошибок.

```
НЕ (в сегменте A)
И НЕ (в сегменте B)
И НЕ (в сегменте C)
И НЕ (в сегменте D)
...
```

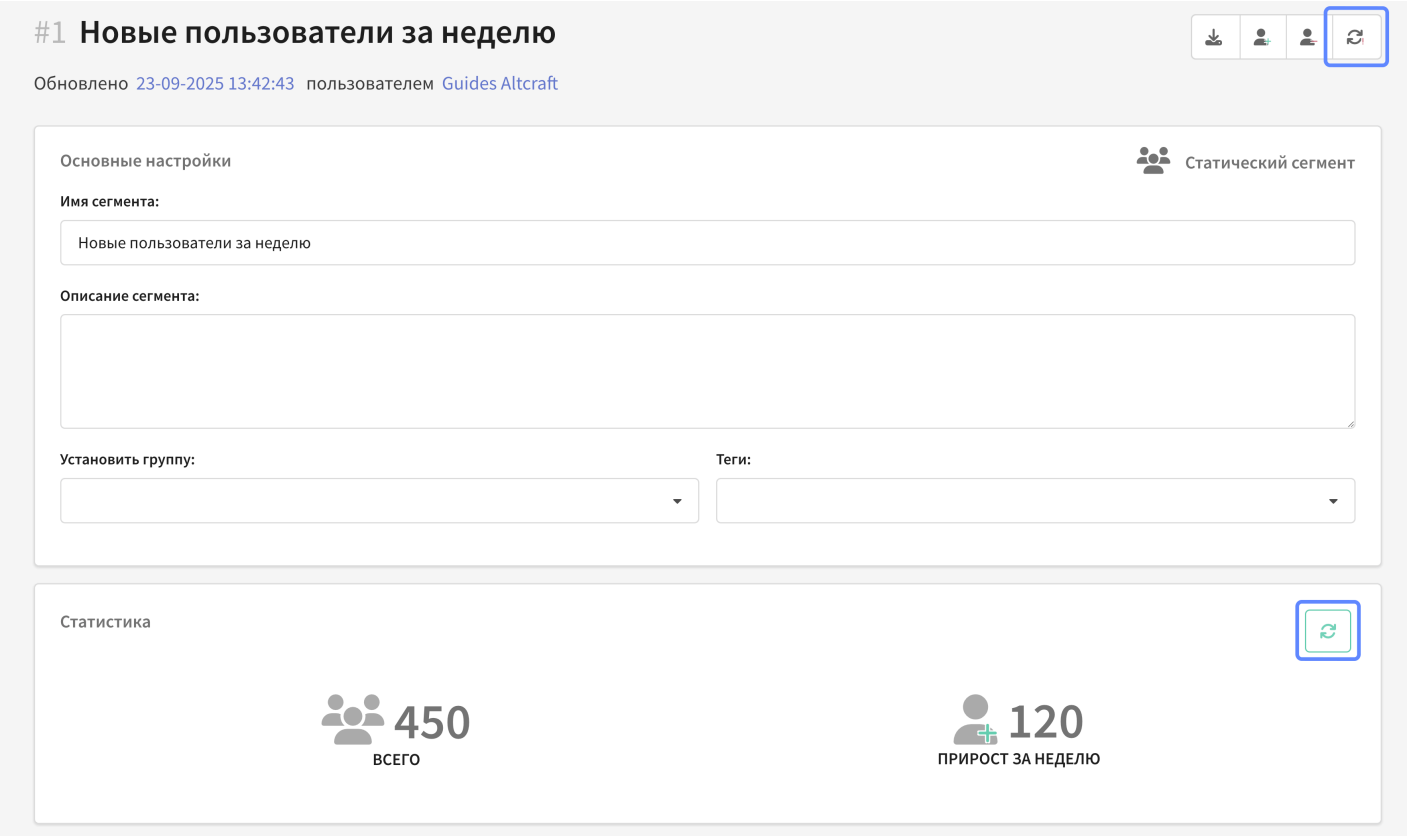
Такие конструкции затрудняют понимание и поддержку правил. Кроме того, производительность пересчёта сегментов снижается — каждое отрицание требует дополнительных проверок.

Рекомендации:

- Не используйте более **2–3 условий** с “НЕ” подряд без веской причины.
- По возможности заменяйте набор условий с “НЕ” на одно позитивное правило. Вместо того чтобы перечислять, где объект не должен находиться, лучше описать, где он должен находиться.
- Документируйте правила сегментации: указывайте, зачем каждое отрицание добавлено. Это позволит избежать ситуаций, когда одно из условий становится неактуальным, а логика остаётся слишком сложной.

Некорректный выбор действия при пересчёте сегмента

В интерфейсе сегментации доступны **две разные кнопки** с похожей иконкой “обновления”:



- **Кнопка в верхней панели** — запускает пересчёт сегмента. При её использовании обновляется состав сегмента, т.е. пересчитывается список участников на основе заданных условий.
- **Кнопка в блоке “Статистика”** — обновляет только отображаемые показатели (“Всего”, “Прирост за неделю”). Эта кнопка не запускает пересчёт сегмента.

Из-за одинаковой визуальной метки пользователи часто нажимают кнопку в блоке статистики, ожидая, что сегмент пересчитается.

Рекомендации:

- Для пересчёта сегмента используйте именно верхнюю кнопку на панели.
- Кнопку в блоке “Статистика” используйте только для обновления числовых показателей — она не влияет на состав сегмента.

Условия сегментации

В данном разделе собраны условия, по которым можно выделять сегменты профилей:

Сегментация по данным профиля

Данные профиля — это основная информация о подписчике, которую можно использовать для точной сегментации аудитории. Этот раздел объединяет всё: от имени и email до системных меток и статусов подписок.

Основные поля

К основным полям относятся стандартные данные профиля, такие как Email, Телефон, Имя, Фамилия, Дата рождения и другие. Эти поля могут быть разных типов, и для каждого типа доступны свои операторы сравнения.

Операторы

Для большинства полей доступны следующие операторы:

- **Равно / не равно** — точное соответствие поля подписчика правилу сегментации. Значение в правиле задаётся вручную или выбирается из списка.
- **Существует / не существует** — проверяет, активно ли поле для конкретного профиля. Существующее поле может быть пустым или содержать значение.
- **Пустой / не пустой** — проверяет, содержит ли существующее поле какое-либо значение.
- **В файле в колонке / не в файле в колонке** — правило ищет соответствие поля в колонке табличного файла (CSV, xls), который можно прикрепить к правилу.
- **В таблице данных / не в таблице данных** — правило ищет соответствие поля в запросе к внешней SQL базе данных.
- **В ответе на post url в поле / не в ответе на post url в поле** — правило ищет соответствие поля в ответе метода **post** запроса к внешнему API.
- **В списке / не в списке** — правило ищет соответствие поля в списке, который можно скопировать в область для ввода данных. Значения должны быть разделены переносом строки (pipe).

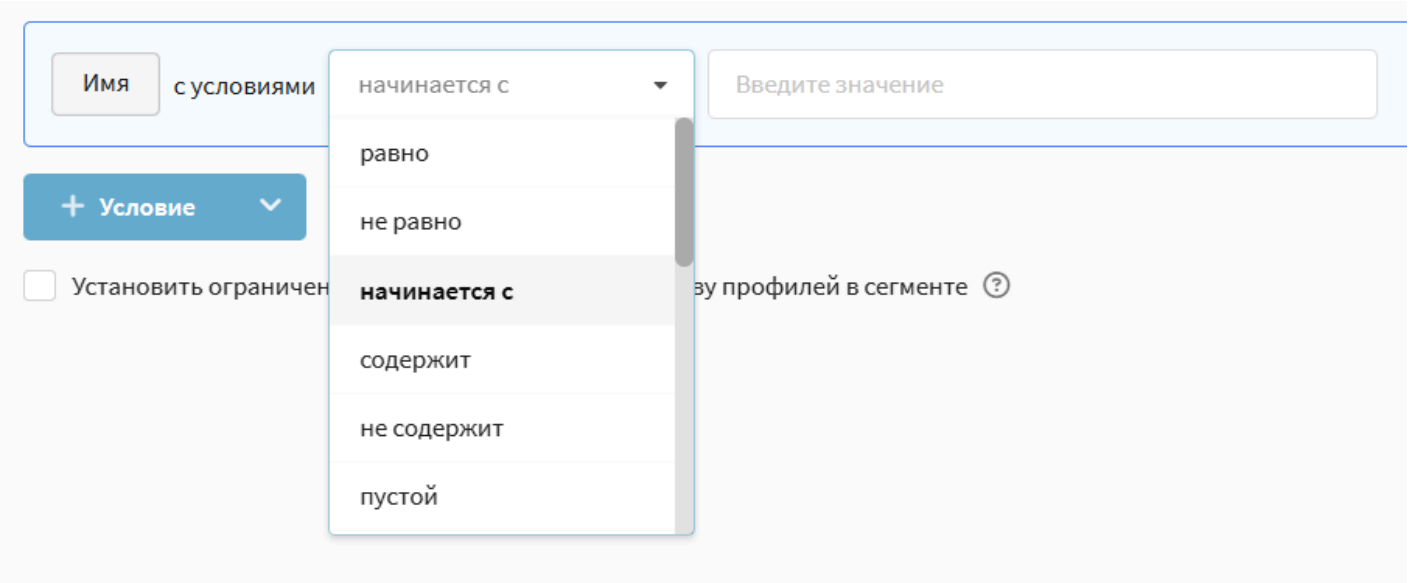
Типы полей и их операторы

Строковые поля

В строковых полях записываются любые значения: буквы, цифры, специальные символы. К ним относятся: `Имя`, `Фамилия`, `Ссылка регистрации`, `Город`, `Регион`, `Индекс` и `Поставщик`.

Для строковых полей доступен дополнительный набор операторов:

- **Начинается с** — поиск по началу значения поля.
- **Содержит / не содержит** — поиск по части значения поля (регистр не учитывается). Оператор “Не содержит” потребляет много системных ресурсов.



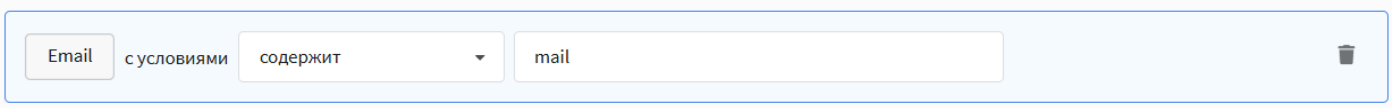
NOTE

Стандартное поле **Телефоны** имеет такой же набор правил сегментации, как строковые поля. Однако в этом поле могут храниться только номера телефонов в формате +71234567890.

Поля типа Email

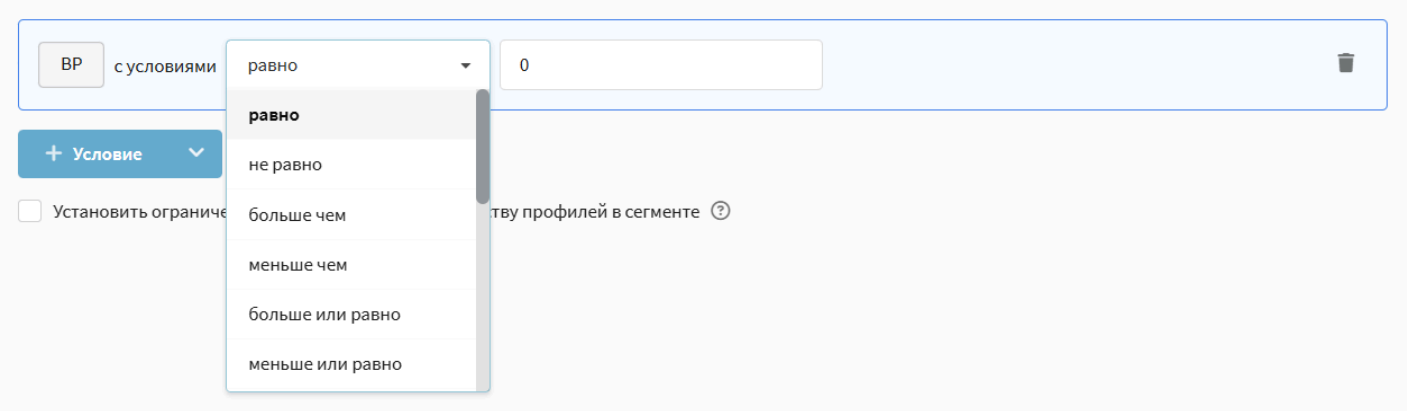
Поле **Email** используется для идентификации профиля и имеет особый набор операторов, касающихся доменов и доменных групп.

- **Доменная группа равна / не равна** — поиск email адресов выбранной доменной группы (mail.ru, yandex.ru).
- **Домен равен / не равен / содержит** — поиск по домену электронной почты подписчиков или его части (bk.ru, list.ru, ya.ru, домены организаций).



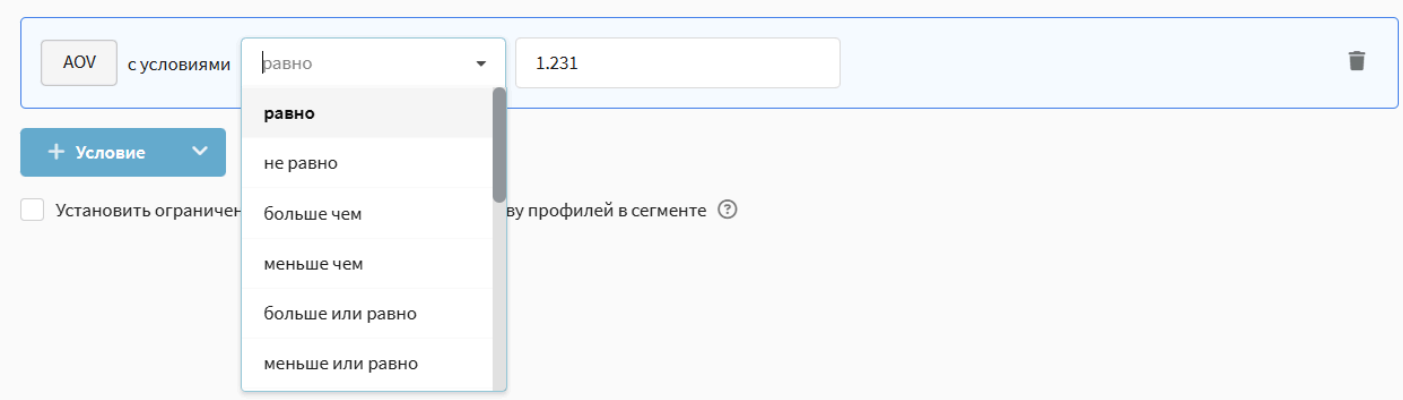
Целочисленные поля

Поля целочисленного типа могут содержать **положительные и отрицательные целые числа**. Доступны операторы: **Равно / не равно / больше чем / меньше чем**.



Поля с плавающей точкой

Поля с плавающей точкой могут содержать **положительные и отрицательные десятичные значения**. Доступны операторы: **Равно / не равно / больше чем / меньше чем**.

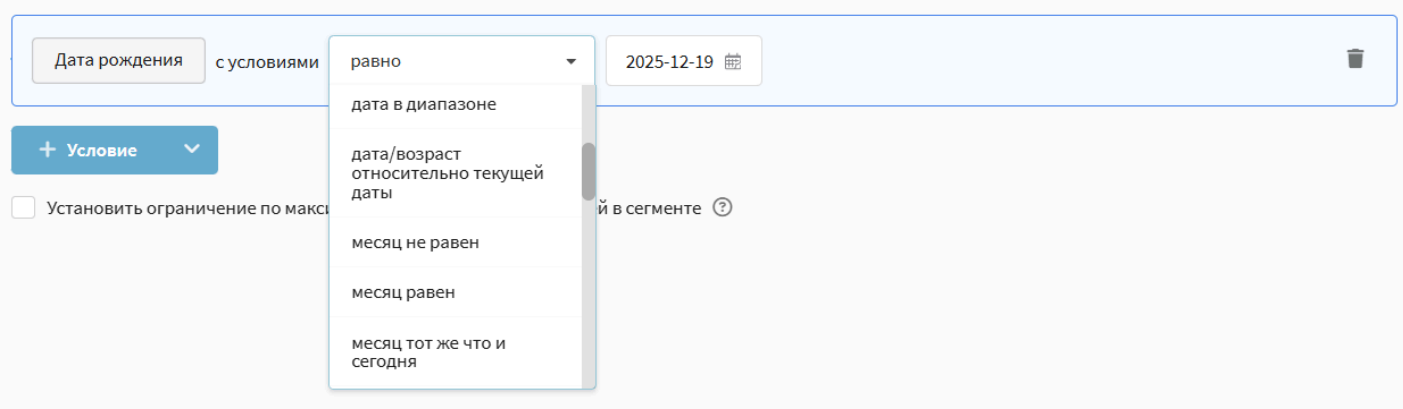


Дата

Поля типа **Дата** хранят дату в формате RFC 3339 (например, 2018-02-22T21:00:00Z). К ним относятся: `Дата регистрации` и `Дата рождения`.

Для полей с датой предусмотрен расширенный набор операторов сегментации:

- **Больше чем / меньше чем** — выборка по дате после или до указанной.
- **Дата в диапазоне** — выборка по дате, входящей в указанный период.
- **Дата/возраст относительно текущей даты** — выборка по моменту в прошлом или будущем.
- **Месяц равен / не равен / тот же что и сегодня** — выборка по месяцу.
- **День месяца равен / не равен / больше чем / меньше чем / такой же что и текущий** — выборка по дню месяца.
- **День и месяц равны / не равны / больше чем / меньше чем / тот же что и текущий** — выборка по дню и месяцу (без года).

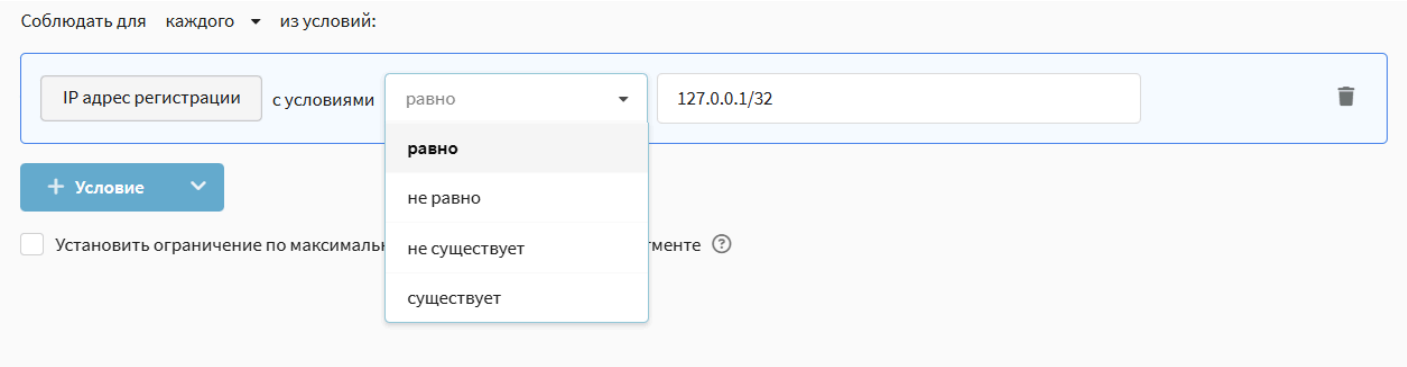


TIP

Динамическая сегментация по полю с датой отлично подходит для триггерных напоминаний (например, об окончании срока услуги). Для этого настройте ежедневную регулярную рассылку по такому сегменту.

IP адрес

В полях типа **IP адрес** хранятся адреса в формате IPv4. К ним относятся `IP адрес` и `IP адрес регистрации`. Для IP адресов доступны только операторы **равно / не равно** и **существует / не существует**.



NOTE

Для IP реализован поиск по маске сети:

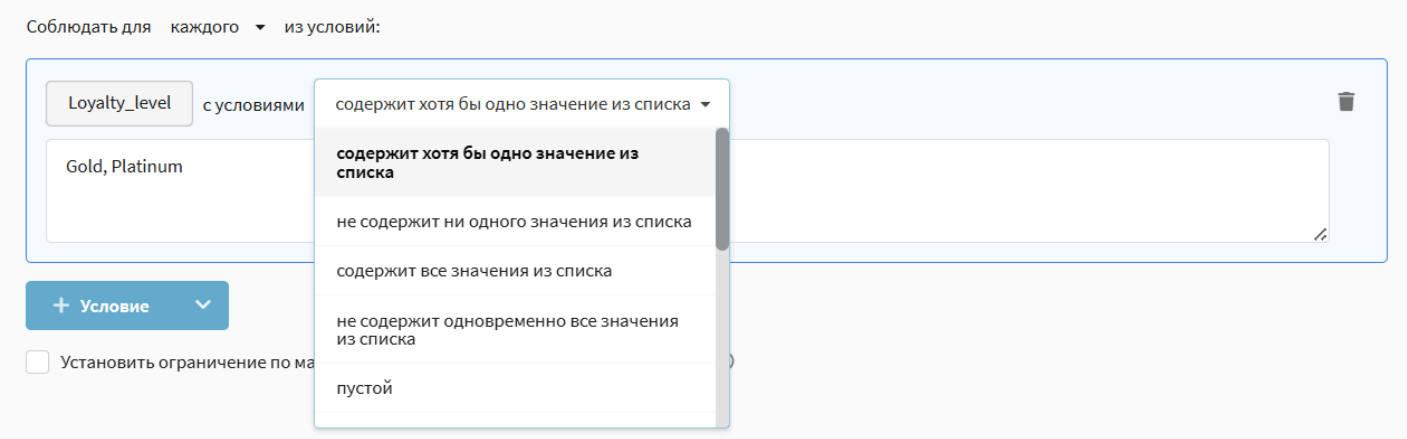
- **равно** — найти подписчиков, IP адреса которых входят в указанную подсеть (например, `127.0.0.1/25`).
- **не равно** — найти подписчиков, IP адреса которых не входят в указанную подсеть. Для поиска по диапазону объедините два правила условием **И**.

Теги

Поле **Теги** представляет собой **список**, разделённых запятыми. Это могут быть, например, приобретённые продукты.

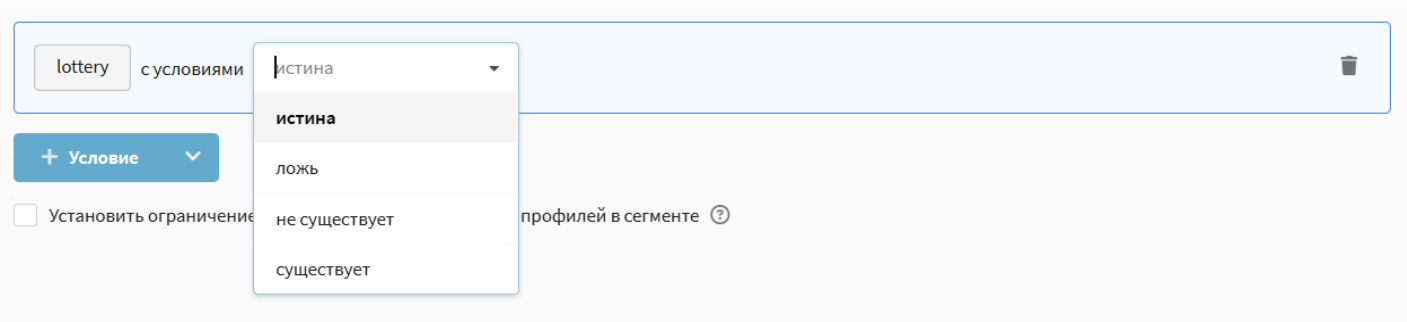
Операторы сегментации тегов:

- **Один из** — в поле есть как минимум одно из значений из правила.
- **Ни один из** — в поле нет ни одного из значений из правила.
- **Все из** — в поле есть все значения из правила.
- **Не все из** — в поле нет каких-либо значений из правила.



Boolean (логическое поле)

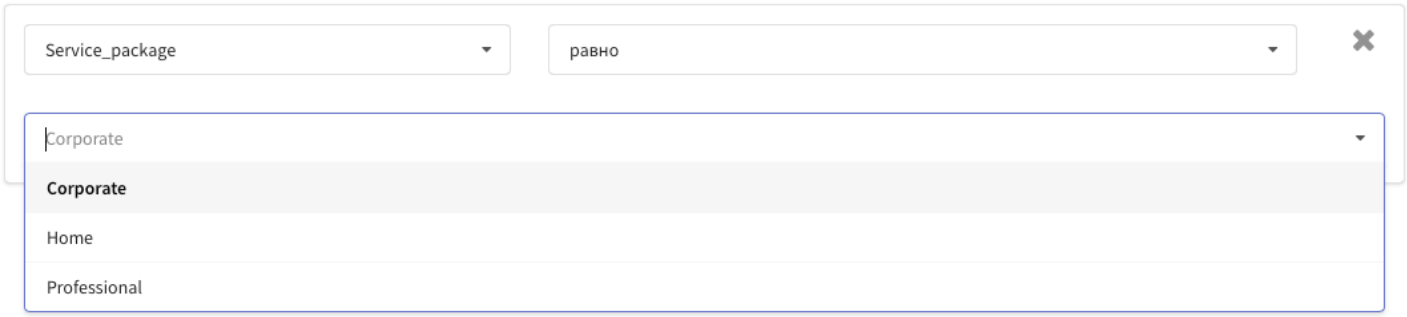
Поле, содержащее логическое ДА/НЕТ (true/false). Удобно для статусов или наличия характеристик. Операторы: **равно true / равно false**.



Список (enumeration)

Поле, содержащее одно из предопределённых значений. В стандартных полях такой тип имеет поле **Пол**.

Операторы могут только сравнивать значение с выбранным в списке или проверять существование поля.



Дополнительные поля

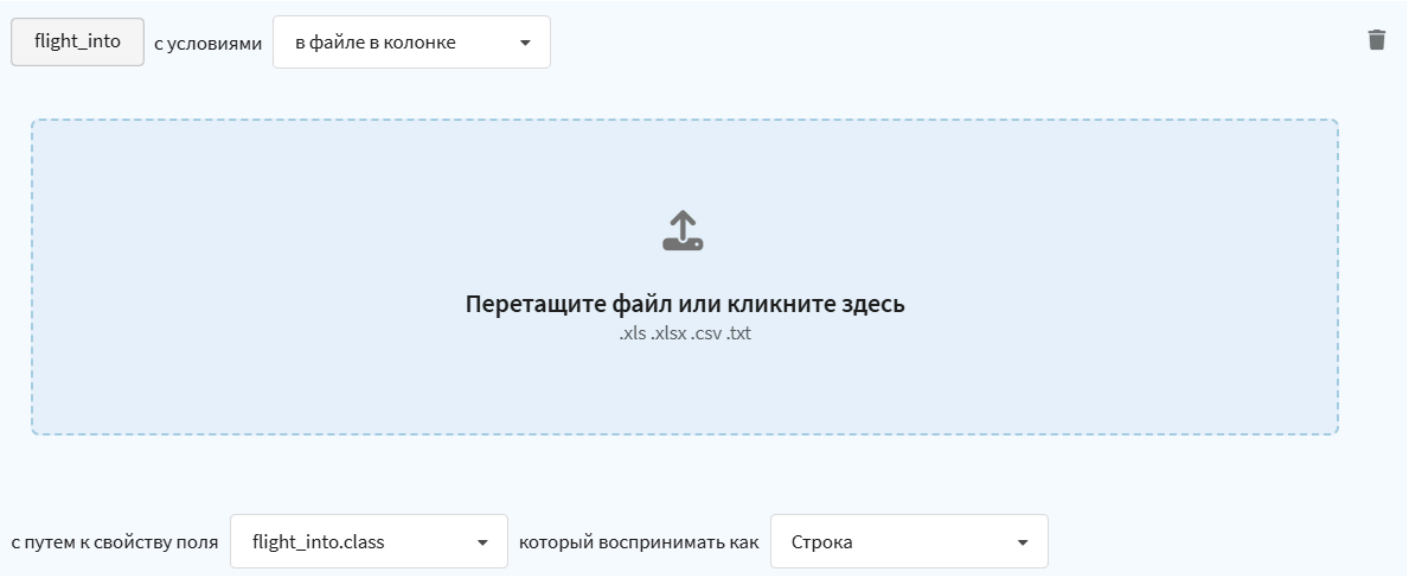
Вы можете создавать в структуре базы собственные поля для хранения специфичной информации. Для сегментации доступны те же операторы, что и для основных полей, в зависимости от типа поля.

Объект

Поле содержит произвольный объект для хранения сложных структур данных.

Доступные условия: **существует / не существует**, а также проверки в списках и внешних источниках.

При необходимости укажите **путь** к конкретному свойству поля. **Тип** свойства определится автоматически, но можно переопределить его, изменив значение в поле “**Воспринимать как**”.

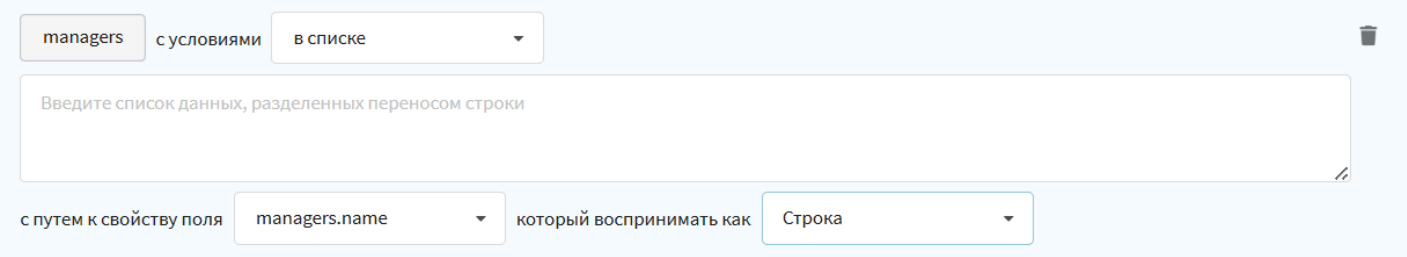


Массив объектов

Поле содержит массив произвольных объектов для хранения подтаблицы (например, перечень договоров).

Доступные условия: **существует / не существует**, а также проверки в списках и внешних источниках с уточнением “все”, “хотя бы один” или “не”.

Также можно указать **путь** к свойству и его **тип**.



Системные поля

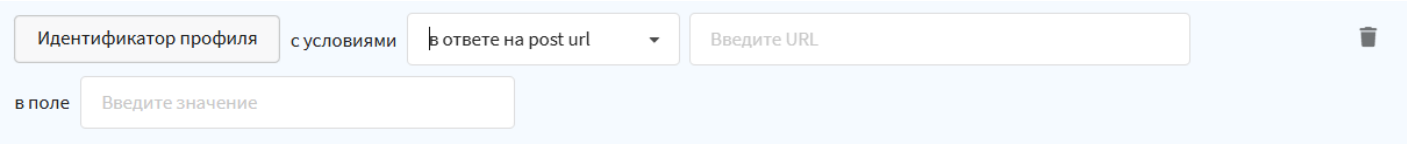
К системным данным, доступным для сегментации, относятся поля, заполняемые автоматически платформой.

Статус профиля

Содержит только [глобальный статус профиля], определяющий возможность коммуникации с клиентом. Выберите необходимый статус из выпадающего меню.

Идентификатор профиля

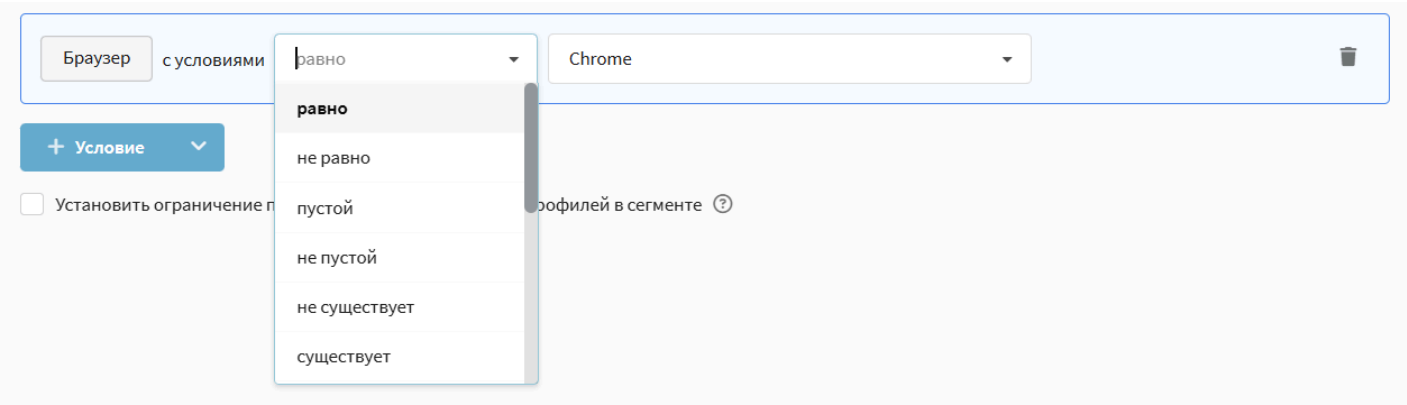
Уникальный идентификатор для каждого клиентского профиля. Доступны опции **равно/не равно**, а также проверки наличия/отсутствия идентификатора в списках и внешних источниках (файл, SQL, API).



Браузер и операционная система

Поля **Браузер** и **Операционная система** автоматически заполняются, когда клиент открывает письмо или выполняет действие на сайте (используется заголовок User agent).

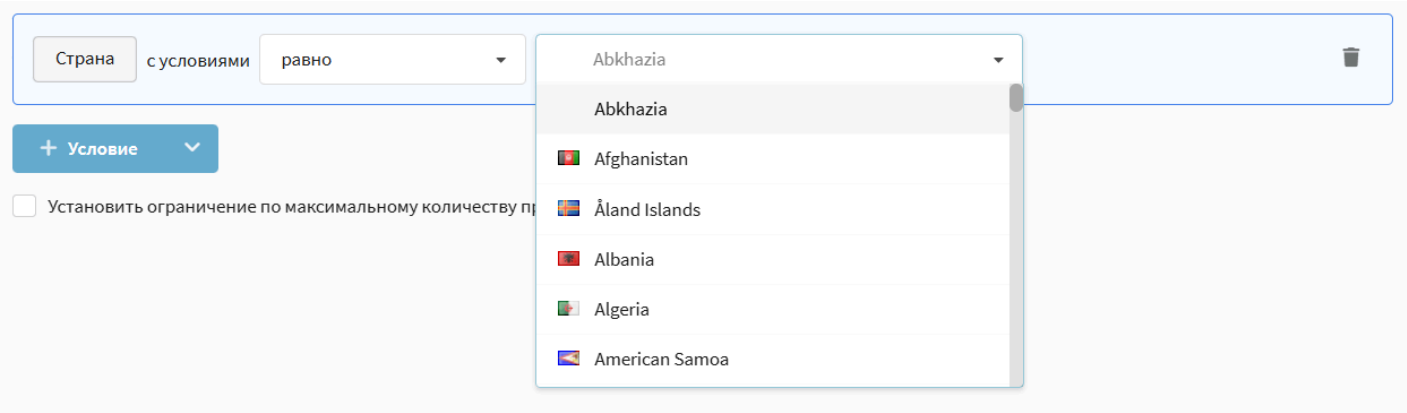
Доступны опции **равно/не равно**, **существует/не существует**, **пустой/не пустой**, а также проверки в списках и внешних источниках.



Данные геолокации

Данные местоположения (Страна, Регион, Город, Индекс, Временная зона) могут быть импортированы вручную или определены автоматически по IP адресу в момент открытия письма или перехода по ссылке.

Используйте стандартные операторы для строковых полей для сегментации по геолокации.

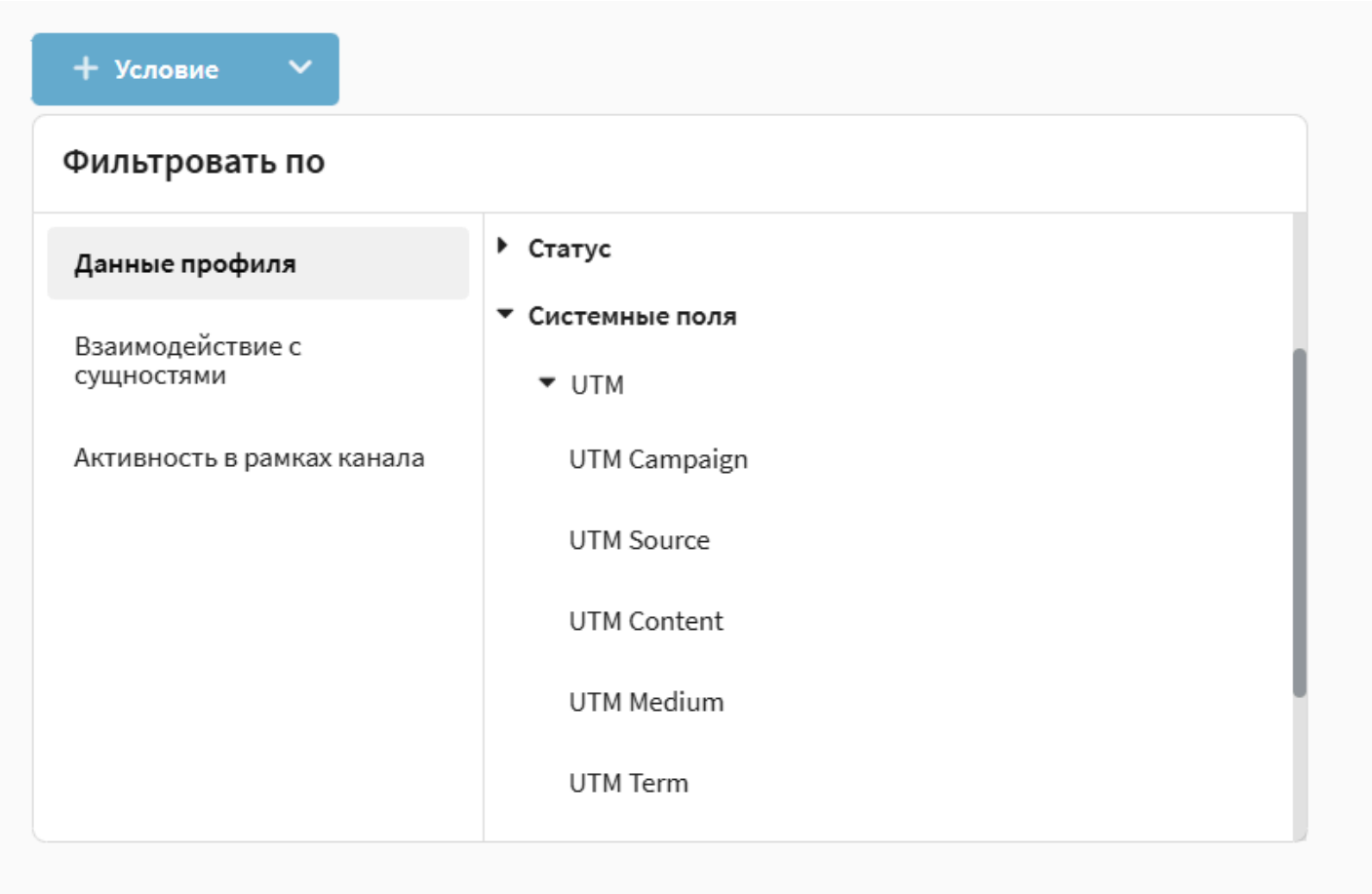


UTM-метки

UTM-метки используются для отслеживания источников трафика. Доступны следующие поля:

- **UTM Campaign** — название рассылки.
- **UTM Source** — источник трафика (рекламная система).
- **UTM Medium** — тип рассылки (контекстная, баннерная и т.д.).
- **UTM Content** — дополнительная информация для различия объявлений.
- **UTM Term** — ключевое слово.

Для UTM-полей доступны операторы: **Равно / не равно**, **Начинается с**, **Содержит / не содержит**, **Пустой / не пустой**, **Существует / не существует**, а также проверки в списках и внешних источниках.



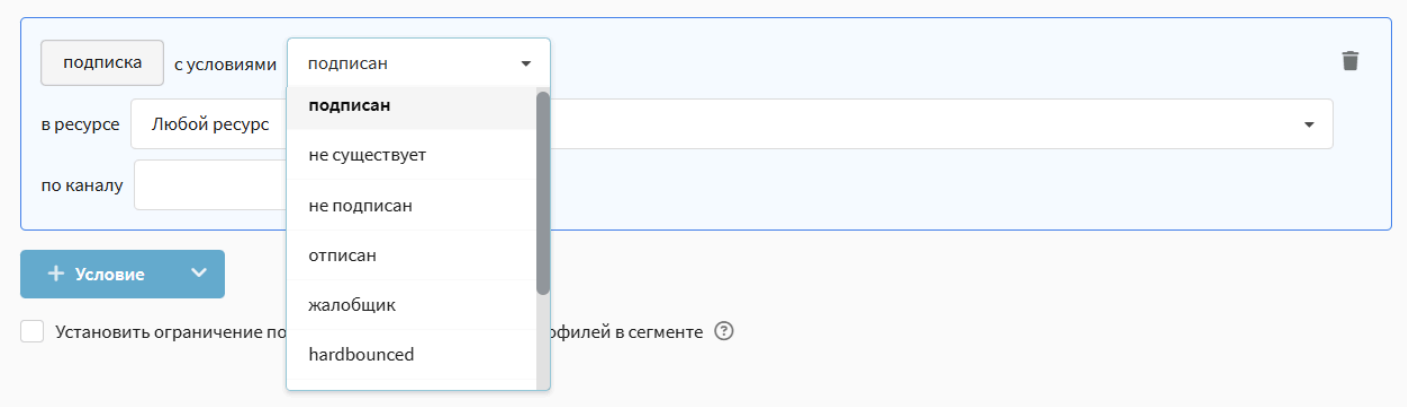
Подписки

Сегментация по данным подписок позволяет определить возможность коммуникации с клиентами по различным каналам связи и настроить омниканальные рассылки.

Статус подписки

Позволяет выбрать профили по наличию и статусу подписки на определённые ресурс и канал связи.

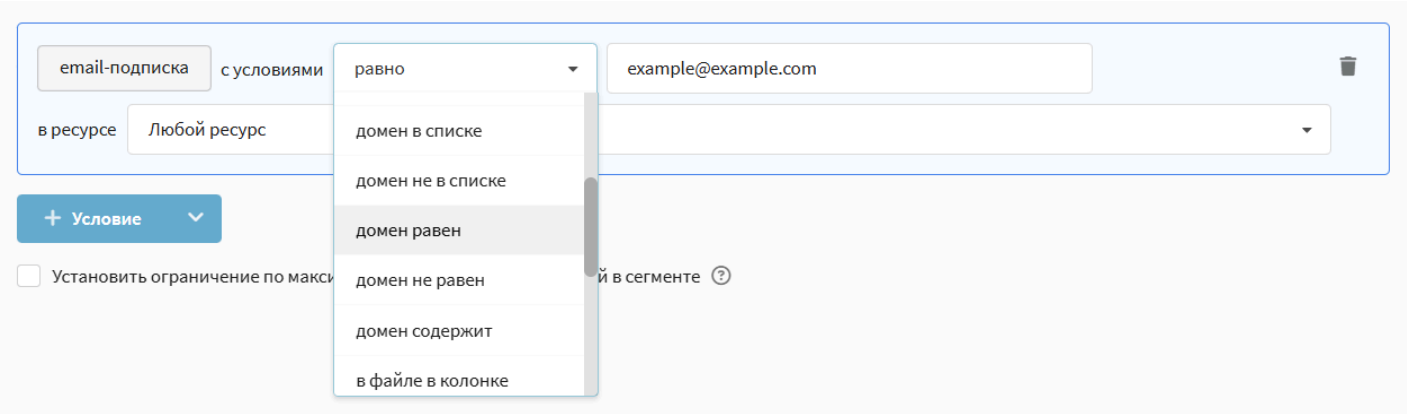
- **Не существует** — у профиля **отсутствует** подписка на выбранные ресурс и канал.
- **Подписан** — у профиля активные подписки (статус “Подписан”).
- **Не подписан** — подписка **существует**, но её статус не позволяет отправлять сообщения (“Отписан”, “Жалобщик”, “Hardbounced”, “Не подтвержден”, “Приостановлен”, “Не валиден”).
- **Конкретный статус** — выборка по одному из статусов: “Отписан”, “Жалобщик” и т.д.



Email-подписка

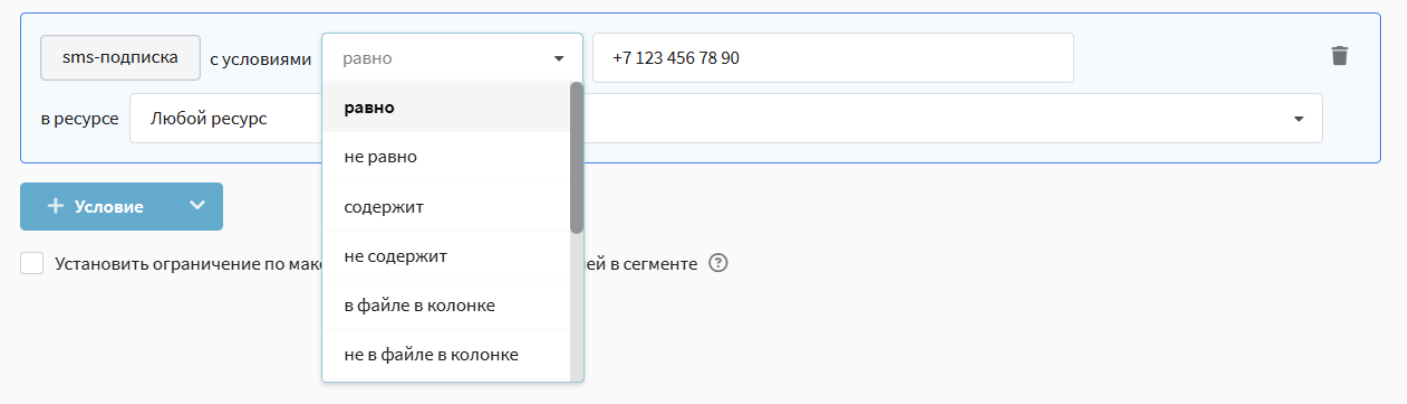
Позволяет выделить группы подписчиков по провайдеру электронной почты или найти конкретные email. Набор операторов аналогичен основному полю **Email**:

- **Равно / не равно** — точное соответствие адреса.
- **Содержит** — поиск по части email адреса.
- **Доменная группа равна / не равна** — поиск по доменной группе (mail.ru, yandex.ru).
- **Домен равен / не равен / содержит** — поиск по домену или его части.
- **Стандартные операторы** поиска данных в списках и внешних источниках.



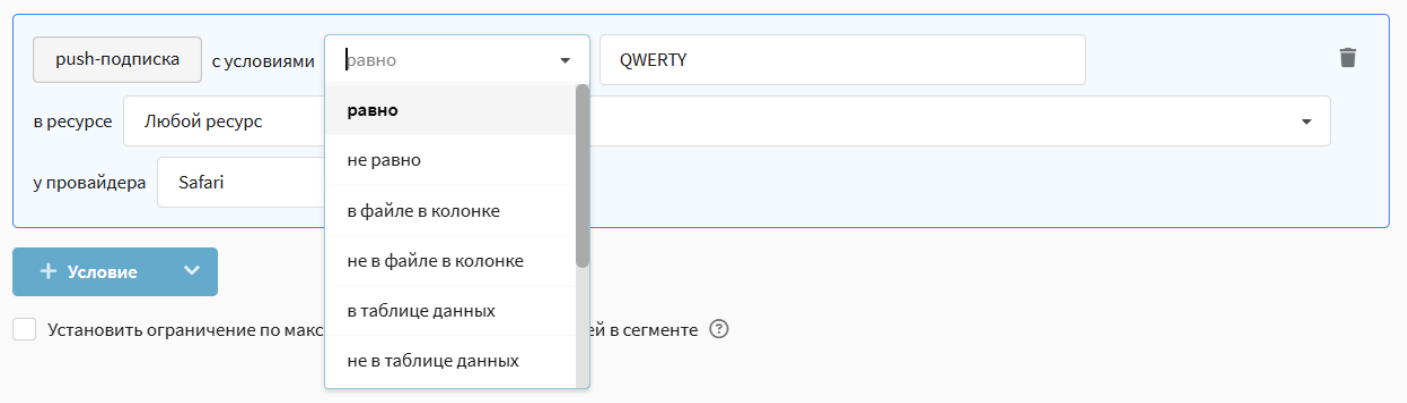
SMS-подписка

Сегментация по данным, содержащимся в SMS подписке. Доступны выборки по точному или частичному соответствию: **равно / не равно, содержит / не содержит**, а также проверки в списках и внешних источниках.



Push-подписка

Сегментация по данным, содержащимся в Push подписке. Доступны выборки только по полному соответствию **равно / не равно** и операторы поиска данных в списках и внешних источниках. Дополнительно можно уточнить провайдера сервиса push-уведомлений.



TIP

Подробную информацию о типах полей в базе и допустимых значениях вы найдете в **отдельной статье**.

Сегментация по взаимодействиям с сущностями

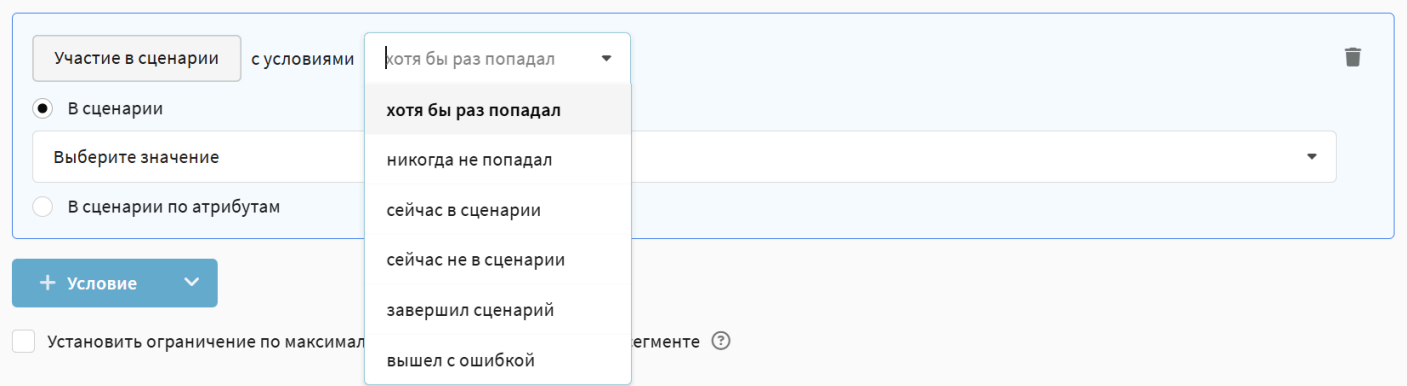
В этом разделе собраны правила для сегментации профилей на основе их взаимодействия с различными функциональными сущностями платформы Альткрафт, такими как сценарии автоматизации, формы, попапы, сегменты и программы лояльности.

Сценарии

Сегментация по участию в сценариях автоматизации. Позволяет выбрать профили, которые находятся в сценарии, завершили его или вышли с ошибкой.

Доступные операторы:

- **хотя бы раз был** — профиль когда-либо участвовал в выбранном сценарии.
- **никогда не был** — профиль никогда не участвовал в выбранном сценарии.
- **сейчас в сценарии** — профиль в данный момент активен в сценарии.
- **сейчас не в сценарии** — профиль в данный момент не активен в сценарии.
- **завершил сценарий** — профиль успешно завершил выполнение сценария.
- **вышел с ошибкой** — профиль вышел из сценария из-за ошибки.



Сегменты

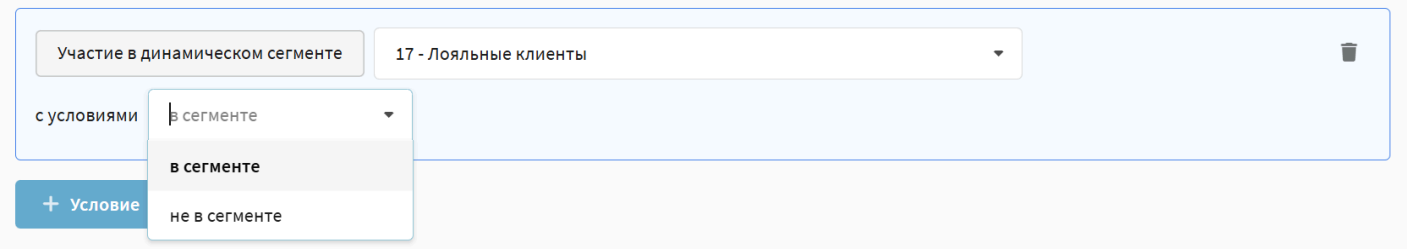
Выборка профилей на основе их участия в других сегментах (динамических или статических). Позволяет создавать сложные многоуровневые выборки.

Участие в динамическом сегменте

Правило проверяет, входит ли профиль в результат расчёта другого динамического сегмента.

Доступные операторы:

- **в сегменте** — профиль входит в выбранный динамический сегмент.
- **не в сегменте** — профиль не входит в выбранный динамический сегмент.



CAUTION

Старайтесь избегать оператора **Не в сегменте** для больших выборок, так как такое правило потребляет значительные системные ресурсы для сравнения массивов данных.

NOTE

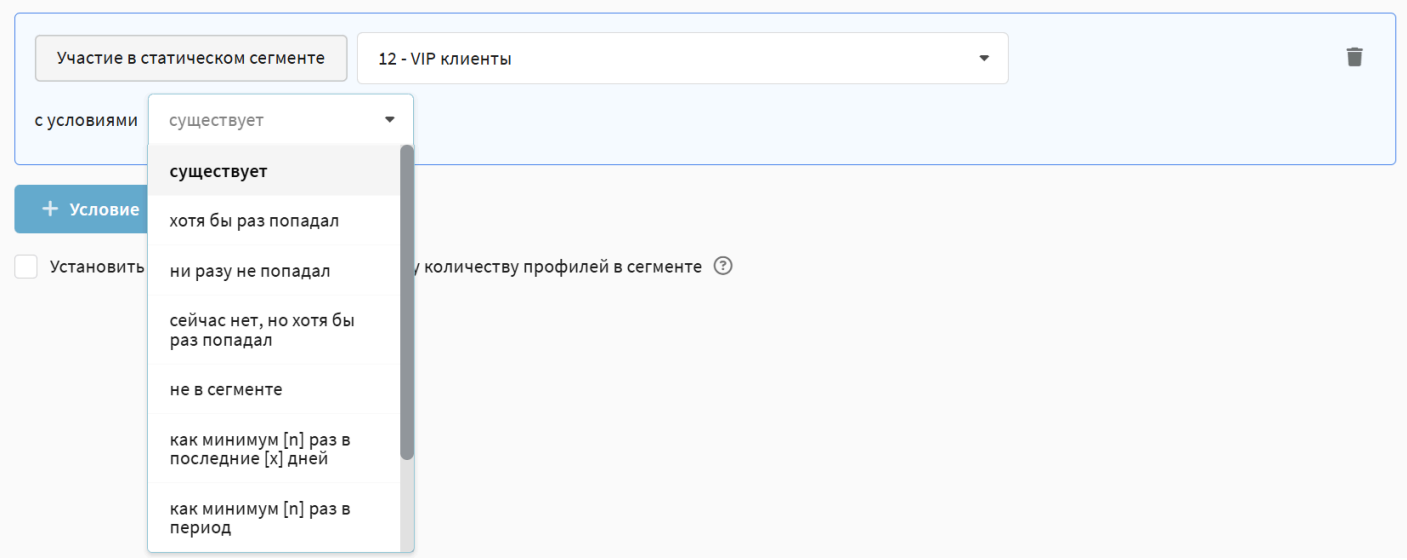
Динамический сегмент для выборки должен быть настроен на ту же базу данных, что и текущий сегмент.

Участие в статическом сегменте

Правило проверяет, находится ли профиль в статическом сегменте (контрольной группе) или когда-либо попадал в него.

Доступные операторы:

- **сейчас в сегменте** — профиль в данный момент находится в статическом сегменте.
- **хотя бы раз попадал** — профиль когда-либо был в этом сегменте.
- **ни разу не попадал** — профиль никогда не был в этом сегменте.
- **сейчас нет, но хотя бы раз попадал** — профиль сейчас не в сегменте, но ранее был в нём.
- **не в сегменте** — профиль в данный момент не входит в сегмент.
- **попадал как минимум [n] раз за [x] дней** — профиль попадал в сегмент указанное количество раз за последние X дней.
- **попадал как минимум [n] раз за период** — профиль попадал в сегмент указанное количество раз за выбранный период.
- **не попадал ни разу за [x] дней** — профиль не попадал в сегмент за последние X дней.
- **не попадал ни разу за период** — профиль не попадал в сегмент за выбранный период.

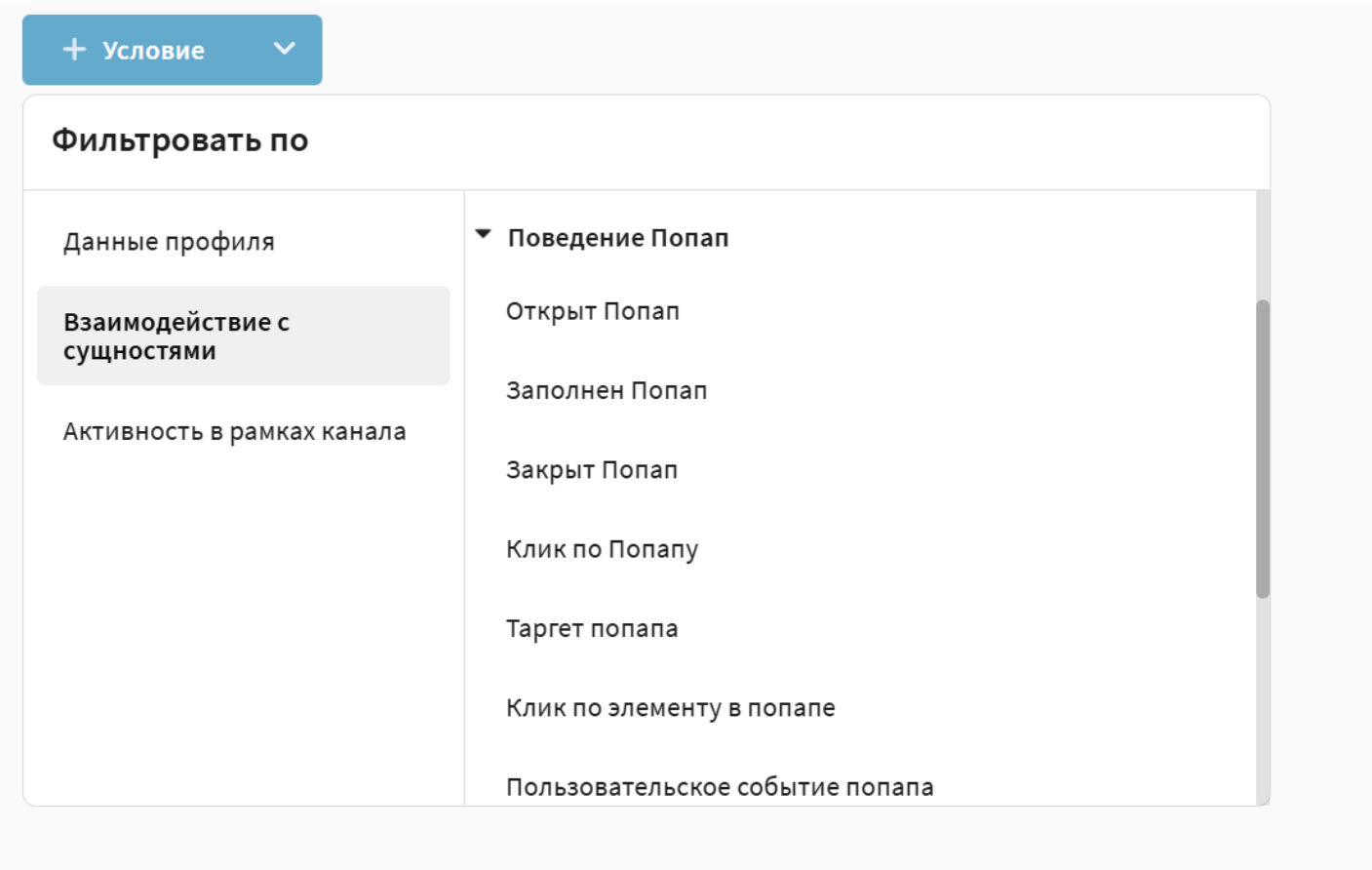


Поведение с попапами

Сегментация по взаимодействию пользователей с попапами на вашем сайте.

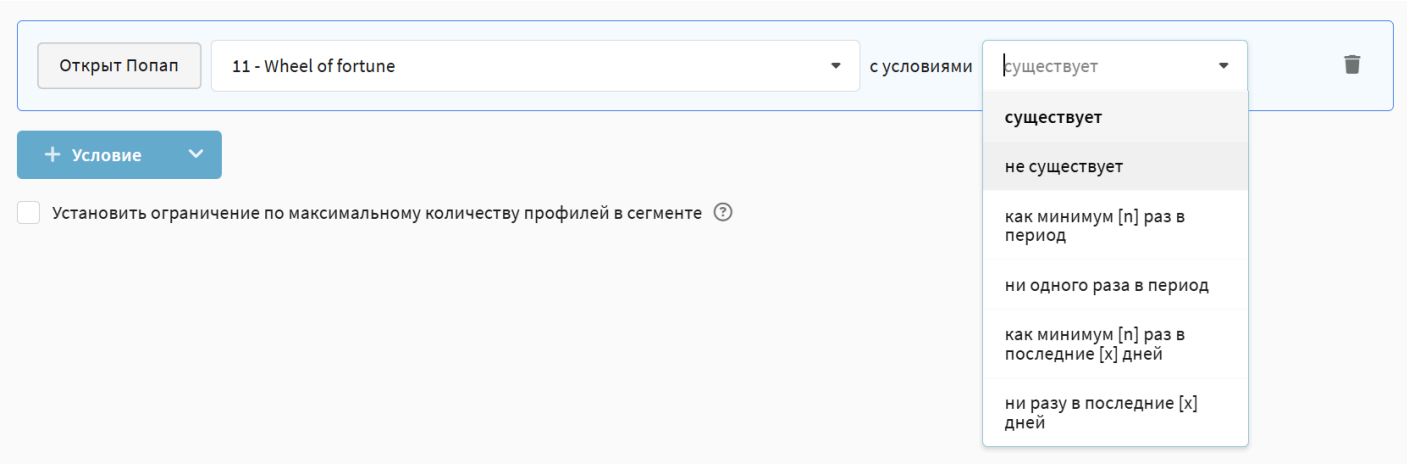
Доступные события для отбора:

- **Открыт Попап** — попап был показан пользователю.
- **Заполнен Попап** — форма в попапе была успешно заполнена и отправлена.
- **Закрыт Попап** — пользователь закрыл попап (клик по кнопке “Закрыть”).
- **Клик по Попапу** — клик по любому элементу попапа, кроме кнопки закрытия.
- **Таргет попапа** — условия для показа попапа были выполнены (даже если он не был показан из-за лимитов).
- **Клик по элементу в попапе** — клик по конкретному HTML-элементу с заданным атрибутом `name`.
- **Пользовательское событие попапа** — событие, настроенное через JavaScript код.



Доступные условия отбора:

- **Существует** — событие было зарегистрировано хотя бы один раз.
- **Не существует** — событие никогда не регистрировалось.
- **Как минимум [n] раз в период** — событие происходило не менее N раз за заданный период.
- **Ни одного раза в период** — событие не происходило за заданный период.
- **Был как минимум [n] раз за [x] дней** — событие происходило не менее N раз за последние X дней.
- **Ни разу в последние [x] дней** — событие не происходило за последние X дней.



Лояльность

Программы лояльности

Для сегментации профилей можно использовать данные по действиям в программах лояльности.

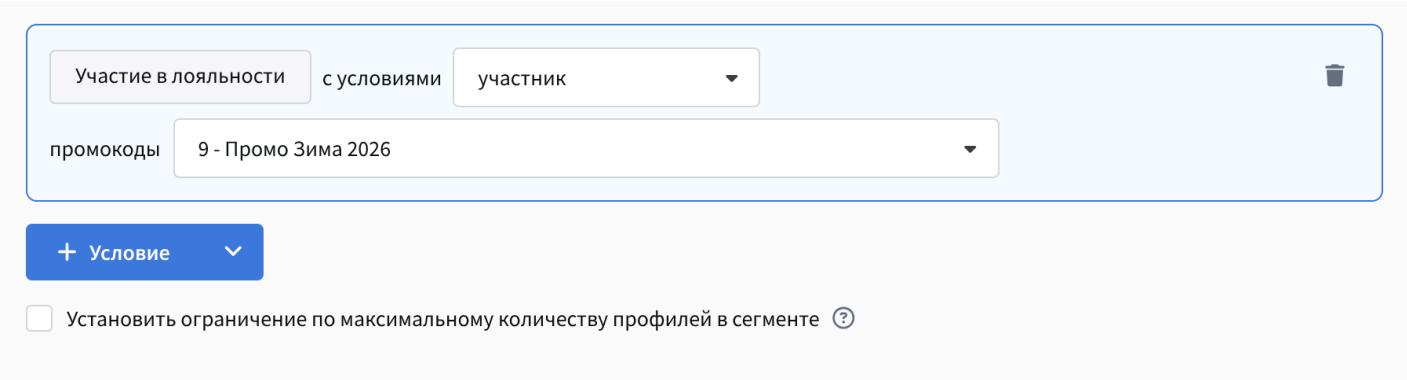
Доступные условия отбора:

- **Участие в программе лояльности** — является ли профиль участником выбранной программы лояльности.
- **Баланс баллов на счету участников** — количество баллов в программе лояльности (равно).
- **Начислены баллы в промоакциях**.
- **Сгорание баллов в промоакциях**.

Доступные операторы:

- **Участник / Не участник**.
- Баланс баллов **равен** / **не равен** / находится **в диапазоне баллов**.
- Начислена **сумма баллов от n до m в период / сумма баллов от n до m в ближайшие x дней**.
- Сгорела **сумма баллов от n до m в период / сумма баллов от n до m в ближайшие x дней**.

Для настройки этих условий может понадобиться выбрать программу лояльности, промоакцию, тип баланса (активный/итоговый), а также тип баллов.



TIP: Итоговый и активный баланс

У начисляемых баллов есть **время активации**. Если оно не указано, то начисленные баллы будут активны сразу. Если же время активации указано, до его наступления баллами нельзя будет воспользоваться.

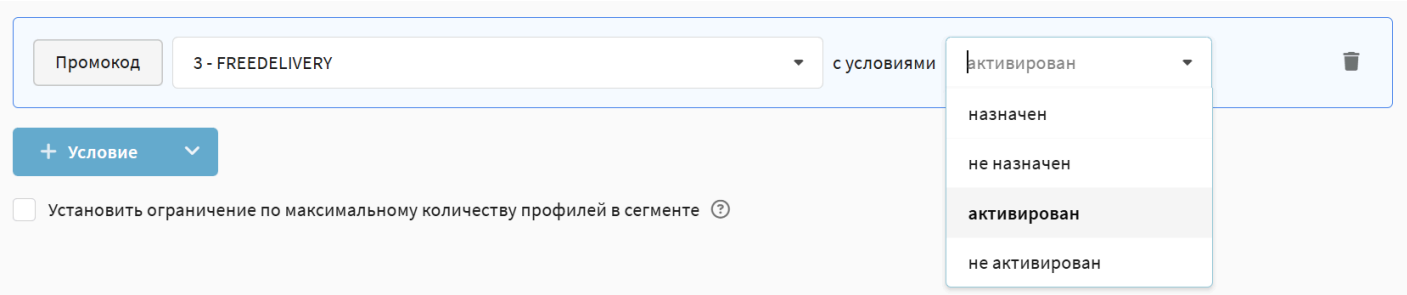
Активный баланс — это количество только тех баллов, которыми можно воспользоваться. А **итоговый баланс** — общий баланс баллов, в том числе и неактивных.

Промокод

Сегментация по взаимодействию с промокодами.

Доступные операторы:

- **Назначен / не назначен** — выборка подписчиков, которым промокод был назначен программой лояльности (вручную или автоматически).
- **Активирован / не активирован** — выборка по факту активации промокода (например, при совершении покупки).



Связи

Сегментация по наличию связей между профилями подписчиков.

Доступные операторы:

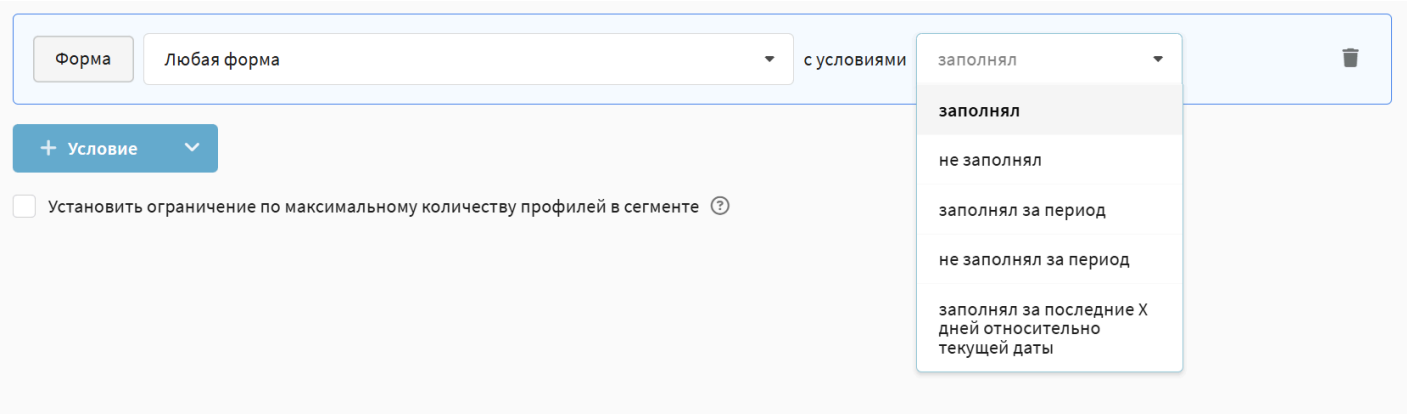
- **Наличие / отсутствие прямой связи** — выборка подписчиков, у которых есть или отсутствует прямая связь с другим профилем.
- **Наличие / отсутствие обратной связи** — выборка подписчиков, у которых есть или отсутствует обратная связь от другого профиля.

Формы

Сегментация по взаимодействию с формами, размещёнными на сайте.

Доступные условия отбора:

- **заполнял** — подписчик когда-либо заполнял указанную форму.
- **не заполнял** — подписчик никогда не заполнял указанную форму.
- **заполнял за выбранный период** — подписчик заполнял форму в указанные даты.
- **не заполнял за выбранный период** — подписчик не заполнял форму в указанные даты.
- **заполнял за последние [x] дней относительно текущей даты** — динамическая выборка по последним X дням.



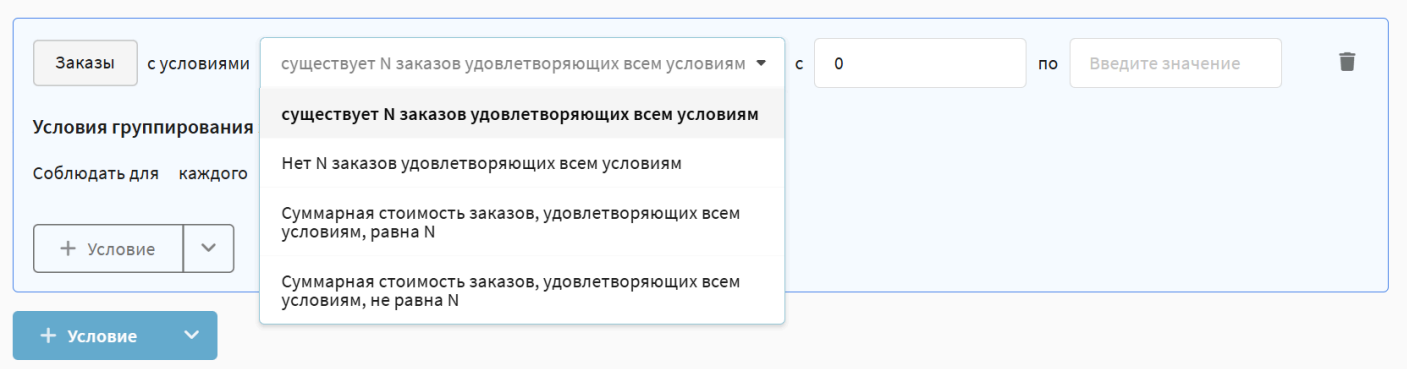
Заказы

Сегментация на основе заказов, зарегистрированных в [модуле “Маркет”](). Позволяет строить выборки по количеству, сумме, среднему чеку, составу заказа, а также по географии, каналам продаж и статусам.

Основные условия отбора

Правила этого типа позволяют выбрать профили, у которых есть (или отсутствуют) заказы, соответствующие заданным параметрам. Доступны следующие операторы:

- **Существует N заказов, удовлетворяющих всем условиям** — выборка профилей с указанным количеством заказов, попадающих под дополнительные условия.
- **Нет N заказов, удовлетворяющих всем условиям** — выборка профилей, у которых нет указанного количества подходящих заказов.
- **Суммарная стоимость заказов, удовлетворяющих всем условиям, равна N** — выборка по общей сумме заказов с учётом дополнительных условий.
- **Суммарная стоимость заказов, удовлетворяющих всем условиям, не равна N** — исключение по общей сумме.
- **Суммарная стоимость заказов, удовлетворяющих всем условиям, в диапазоне** — выборка профилей, у которых общая сумма заказов попадает в указанный числовой диапазон.
- **Суммарная стоимость заказов, удовлетворяющих всем условиям, не в диапазоне** — исключение по диапазону общей суммы.



Условия для анализа покупок

Для более глубокого анализа покупательского поведения доступны правила, основанные на средних значениях. Эти условия учитывают все заказы профиля (или только те, что подходят под дополнительные параметры) за определённый промежуток времени.

Доступные операторы включают проверки **за период** (конкретные даты) и **за последние X дней** (относительно текущей даты). Для каждого из этих временных промежутков можно задать условие “в диапазоне” или “не в диапазоне”:

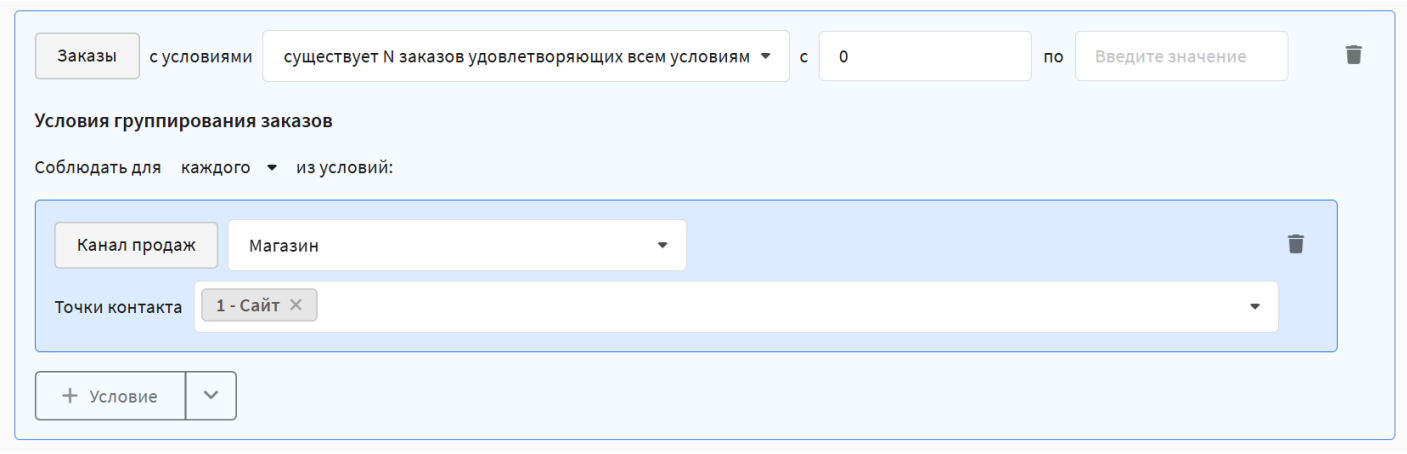
- **Средний чек** — средняя стоимость одного заказа (сумма всех заказов / количество заказов)
- **Средняя сумма покупок** — средняя стоимость одной позиции в заказах (сумма всех заказов / общее количество позиций во всех заказах)
- **Средний чек по товарам из сегмента A** — средняя стоимость заказа, в котором есть товары из указанного сегмента продуктов (сумма заказов, содержащих товары из сегмента A / количество таких заказов)
- **Средняя сумма покупок по товарам из сегмента A** — средняя стоимость одной позиции товара из указанного сегмента (сумма стоимости всех позиций товаров из сегмента A / количество этих позиций)

Условия группировки заказов

В любом из правил, описанных выше, можно задать дополнительные условия, чтобы уточнить, какие именно заказы учитывать. Доступны следующие параметры:

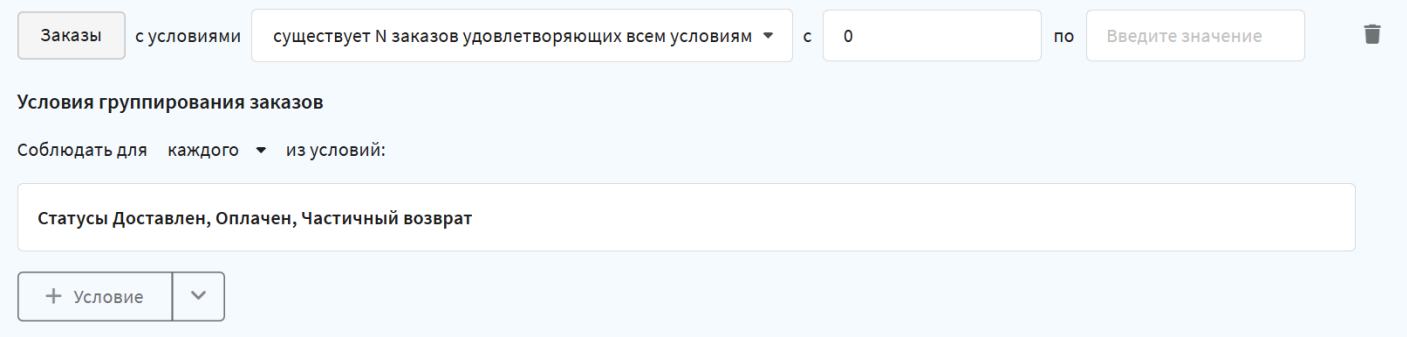
Канал продаж

Выбирает заказы, совершенные в одном канале продаж. Можно указать несколько точек контакта.



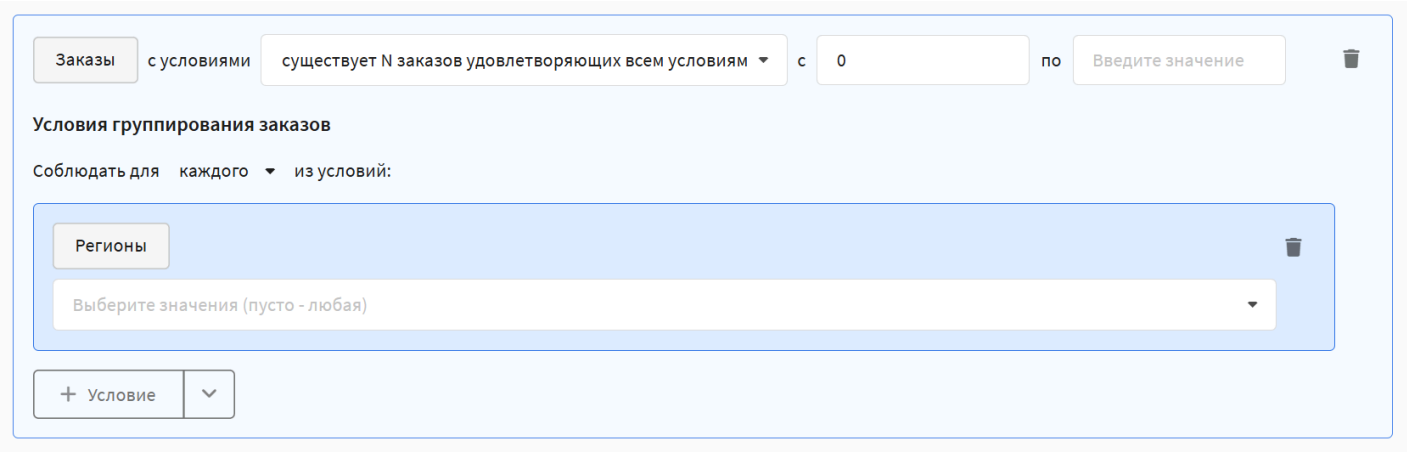
Статусы

Выбирает заказы с указанными статусами: *новый, доставлен, оплачен, отменен, частичный возврат*.



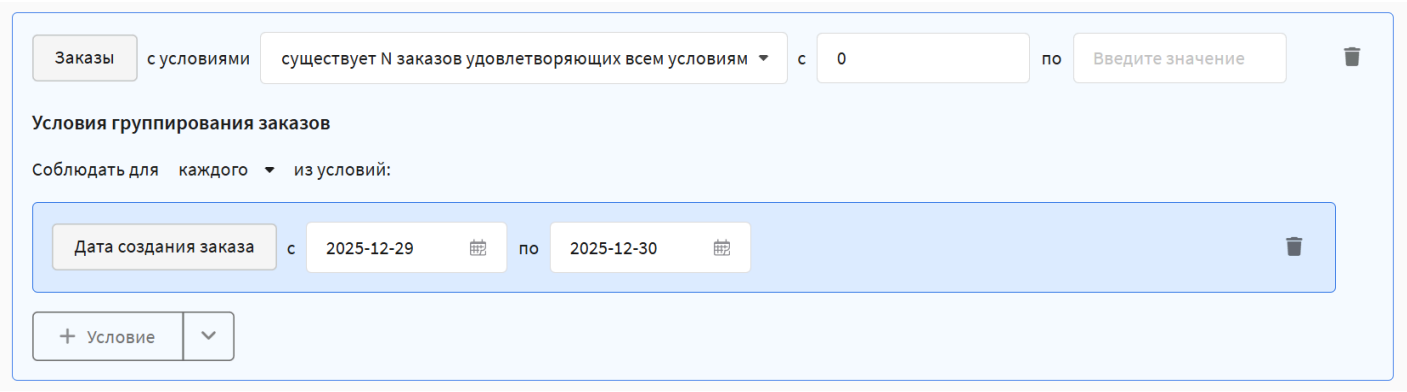
Регионы

Выбирает заказы, совершенные в определенных регионах.



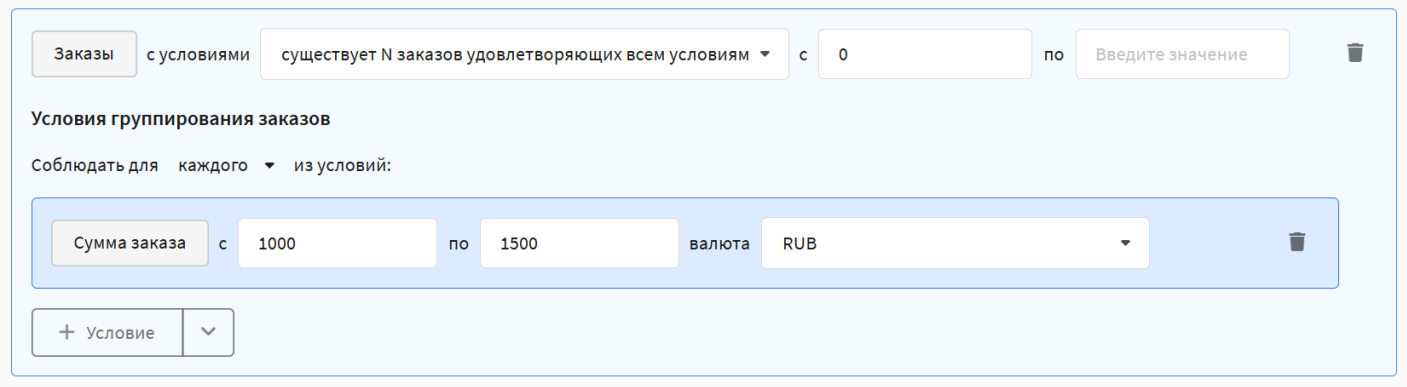
Дата создания заказа

Выбирает заказы, зарегистрированные в указанный период времени.



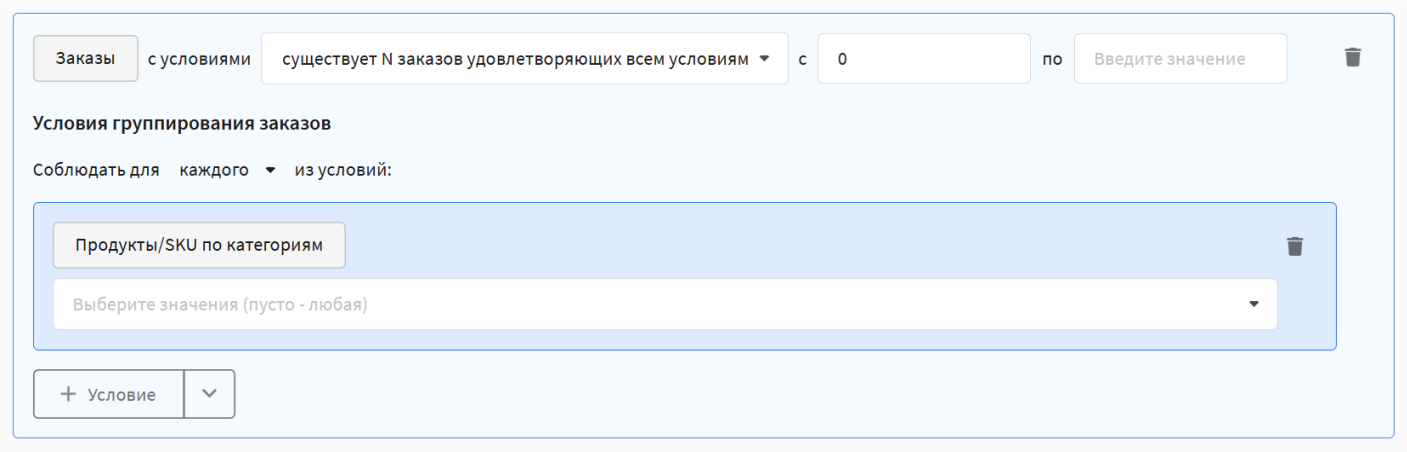
Сумма заказа

Выбирает заказы с определенной стоимостью. Можно указать валюту расчёта.



Продукты/SKU по категориям

Выбирает заказы, содержащие товары из определенных категорий.



Продукты по именам

Выбирает заказы, содержащие конкретные товары — по названию или артикулу (SKU).

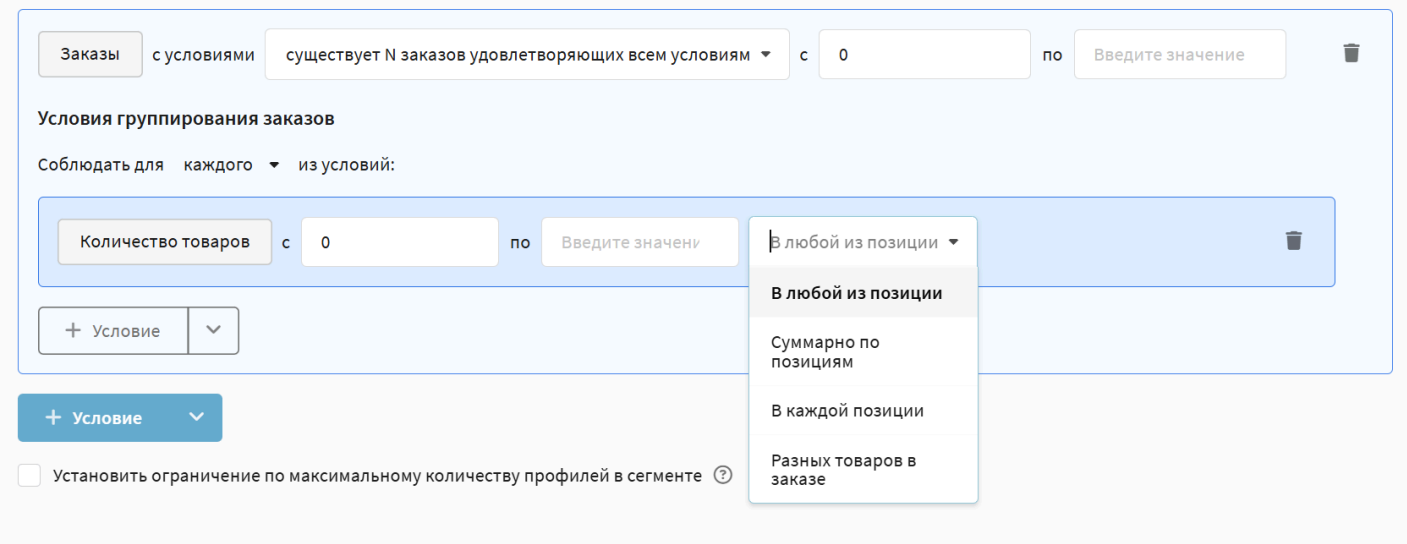
Сегменты продуктов

Выбирает заказы, содержащие товары, входящие в заданный сегмент продуктов.

Количество товаров

Выбирает заказы с определенным количеством товаров. Доступные способы подсчёта:

- **в любой из позиций** — в какой-либо позиции заказа есть указанное количество.
- **суммарно по позициям** — во всех позициях заказа вместе взятых есть указанное количество.
- **в каждой из позиций** — в каждой позиции заказа есть указанное количество.
- **разных товаров в заказе** — количество позиций в заказе (1 товар = 1 позиция).



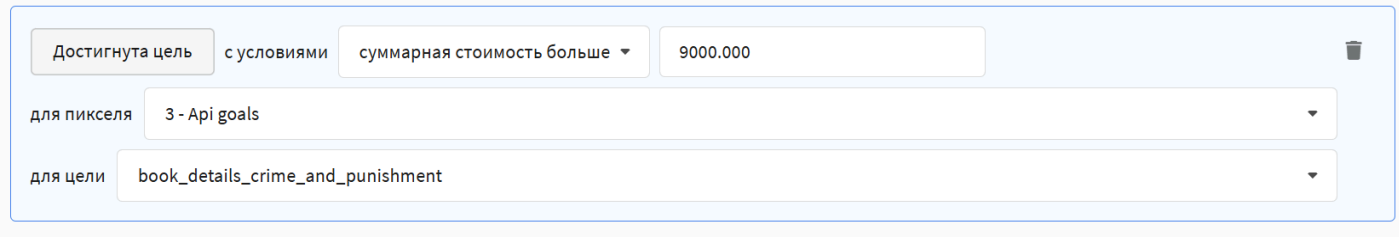
Поведение цели/пикселя

Достигнута цель

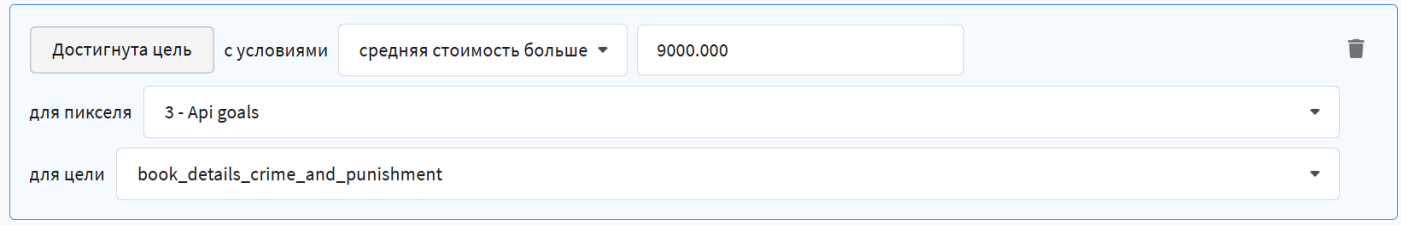
Сегментация по действиям подписчиков на внешних ресурсах (сайтах, в приложениях), отслеживаемым с помощью пикселей Альткрафт.

Доступные операторы:

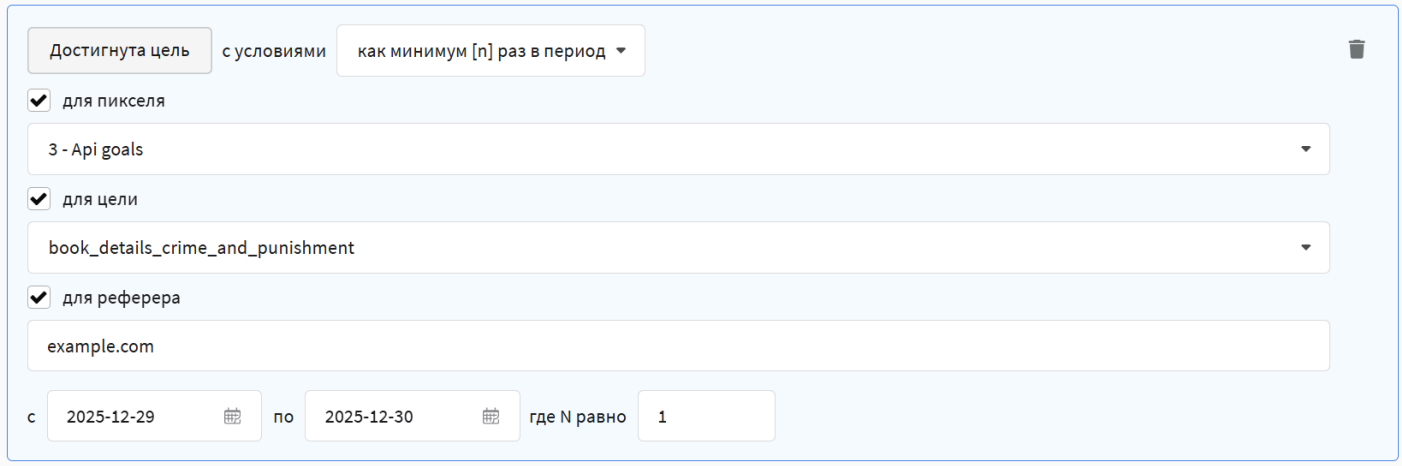
- **Существует / не существует** — проверяет наличие или отсутствие достижения целей.
- **Суммарная стоимость** — рассчитывает сумму стоимости достигнутых целей. Операторы: **больше / меньше / равна**. Можно указать расчёт за всё время, **за период** или **за последние [x] дней**.



- **Средняя стоимость** — рассчитывает среднее арифметическое стоимости достигнутых целей. Операторы: **больше / меньше / равна**. Можно указать расчёт за всё время, **за период** или **за последние [x] дней**.



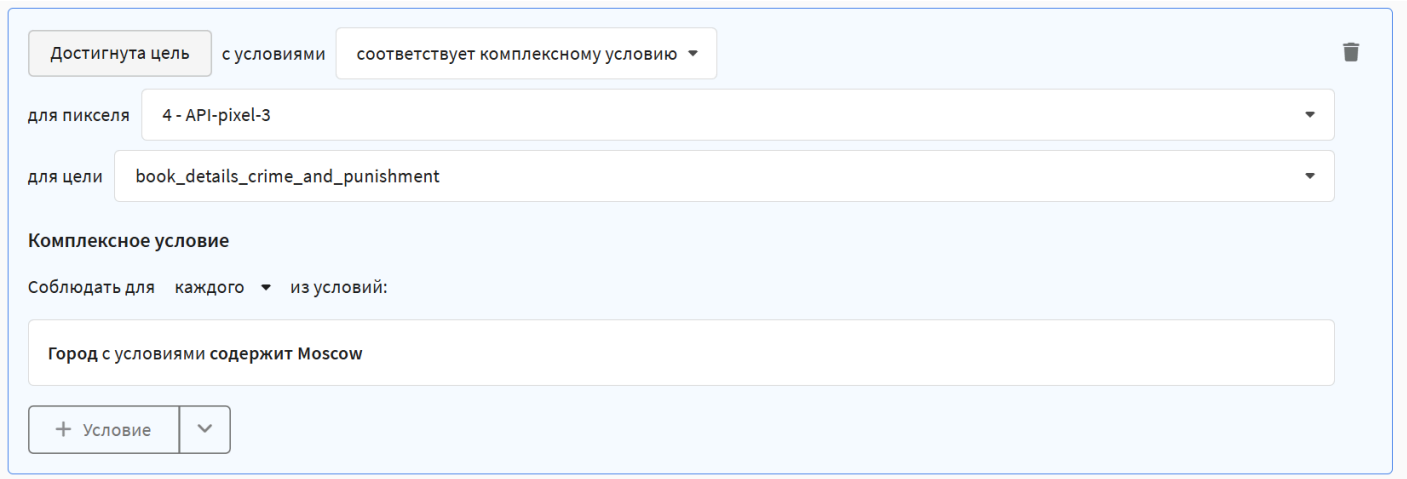
- **Как минимум [n] раз / Ни разу** — выборка по количеству и частоте совершаемых действий. Можно уточнить **цель**, **пиксель** или **реферера** (источник трафика).



- **Соответствует / Не соответствует комплексному условию** — расширенная выборка на основе дополнительных данных, переданных при регистрации цели через API.

Можно настроить правила отбора по:

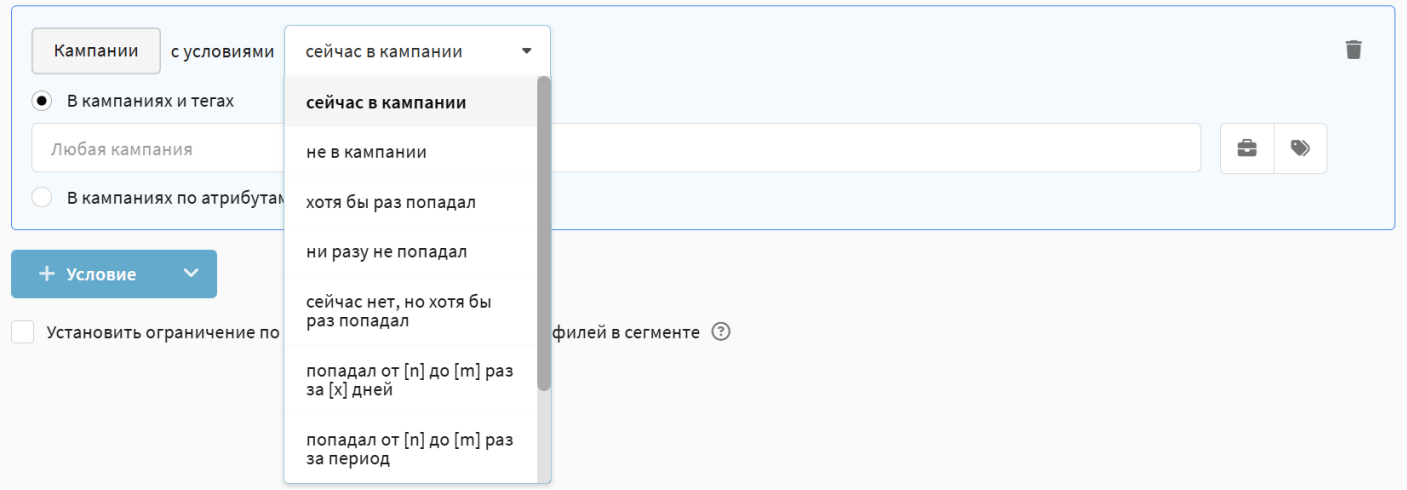
- Значению цели (стоимости)
- Данным из объекта `pixel_data` (строковые, числовые, дата, булевы)
- Устройству профиля и геоданным (браузер, ОС, страна, город и т.д.)
- Отправленным сообщениям и промокодам
- UTM-меткам
- URL Referer и Origin
- Данным заказов профиля



Кампании

Сегментация по участию в кампании

- **сейчас в кампании** — профиль в данный момент находится в глобальной контрольной группе.
- **не в кампании** — профиль в данный момент не находится в кампании.
- **хотя бы раз попадал** — профиль когда-либо был в кампании.
- **ни разу не попадал** — профиль никогда не был в кампании.
- **сейчас нет, но хотя бы раз попадал** — профиль сейчас не в кампании, но ранее был.
- **попадал как минимум [n] раз за [x] дней** — профиль попадал в кампании указанное количество раз за последние X дней.
- **попадал как минимум [n] раз за период** — профиль попадал в кампании указанное количество раз за выбранный период.
- **не попадал ни разу за [x] дней** — профиль не попадал в кампании за последние X дней.
- **не попадал ни разу за период** — профиль не попадал в кампании за выбранный период.



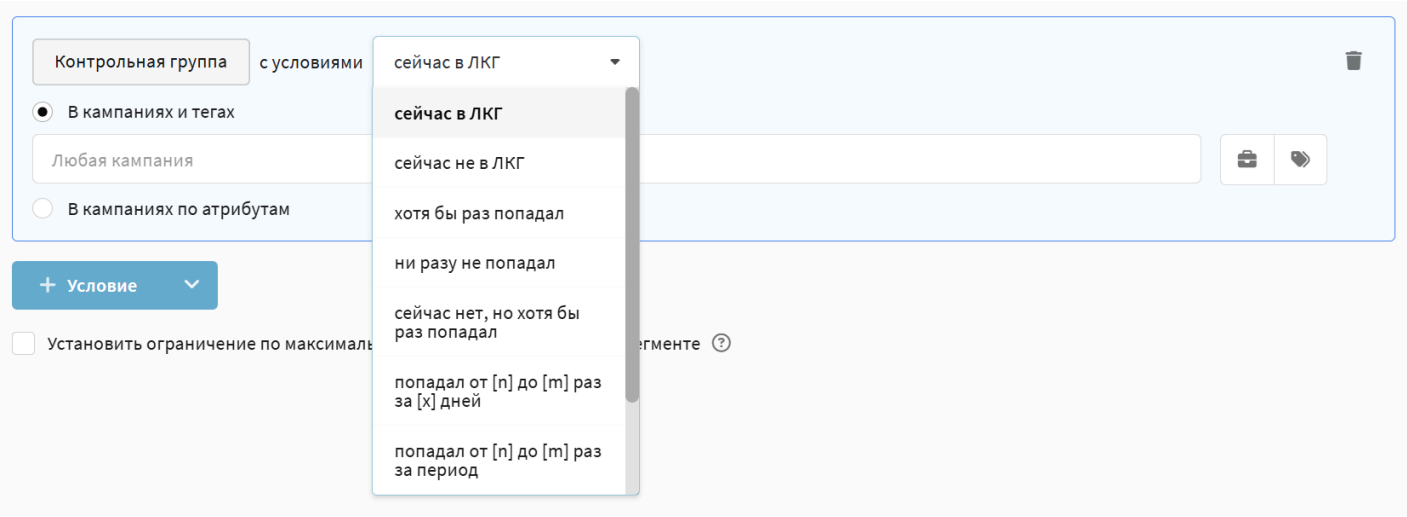
Контрольная группа / Тестовая группа

Сегментация по участию в тестовых или локальных контрольных группах.

Доступные операторы:

- **сейчас в ЛКГ** — профиль в данный момент находится в глобальной контрольной группе.
- **не в ЛКГ** — профиль в данный момент не находится в ЛКГ.
- **хотя бы раз попадал** — профиль когда-либо был в ЛКГ.

- **ни разу не попадал** — профиль никогда не был в ЛКГ.
- **сейчас нет, но хотя бы раз попадал** — профиль сейчас не в ЛКГ, но ранее был.
- **попадал как минимум [n] раз за [x] дней** — профиль попадал в ЛКГ указанное количество раз за последние X дней.
- **попадал как минимум [n] раз за период** — профиль попадал в ЛКГ указанное количество раз за выбранный период.
- **не попадал ни разу за [x] дней** — профиль не попадал в ЛКГ за последние X дней.
- **не попадал ни разу за период** — профиль не попадал в ЛКГ за выбранный период.



Сегментация по активности в каналах коммуникации

История действий подписчиков в рамках отправленных сообщений формируется на основе событий в различных каналах связи. Выборки по этой активности позволяют проводить сегментацию по вовлеченности (например, по открытиям и кликам) и эффективно настраивать ретаргетинговые кампании.

Для большинства правил сегментации по истории активности в сообщениях доступны следующие типы операторов выборки:

- **Существует/не существует** — выборка по наличию как минимум одного действия выбранного типа у подписчиков.
- **Ни одного раза** — выборка по отсутствию хотя бы одного действия выбранного типа. В зависимости от оператора можно дополнительно указать **Рассылку**, **Сообщение**, **период времени** или **количество дней с текущей даты**.
- **Как минимум [n] раз** — выборка по наличию указанного количества событий у подписчиков. В зависимости от оператора можно дополнительно указать **Рассылку**, **Сообщение**, **период времени** или **количество дней с текущей даты**.

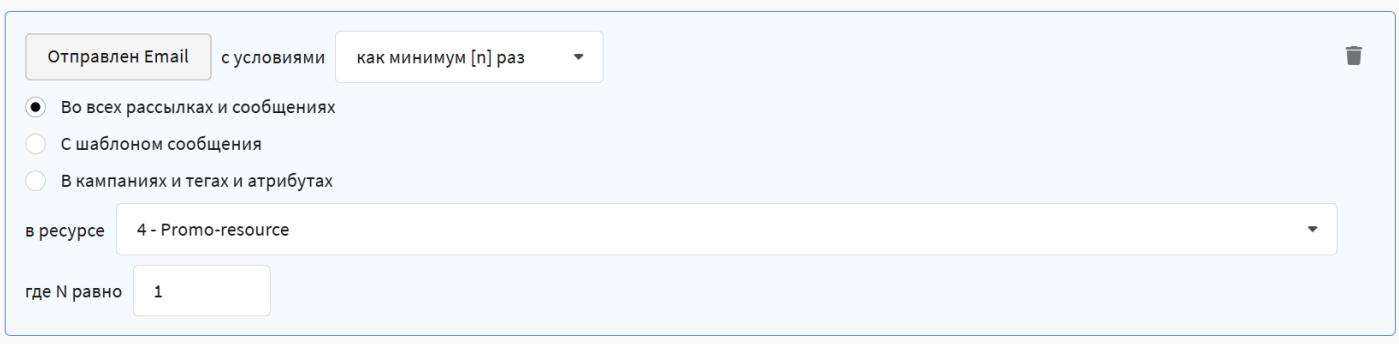
Общие события для всех каналов

Для большинства каналов связи доступны схожие типы событий. Ниже приведены общие группы событий с указанием каналов, где они применяются.

Отправка сообщения

Событие регистрируется, когда сообщение формируется и перенаправляется на отправляющий модуль (почтовый сервер, SMS-шлюз, платформу push или мессенджера).

- **Доступно в:** Email, SMS, Push, Telegram Bot, Telegram Group, WhatsApp, Viber, VK Notify.



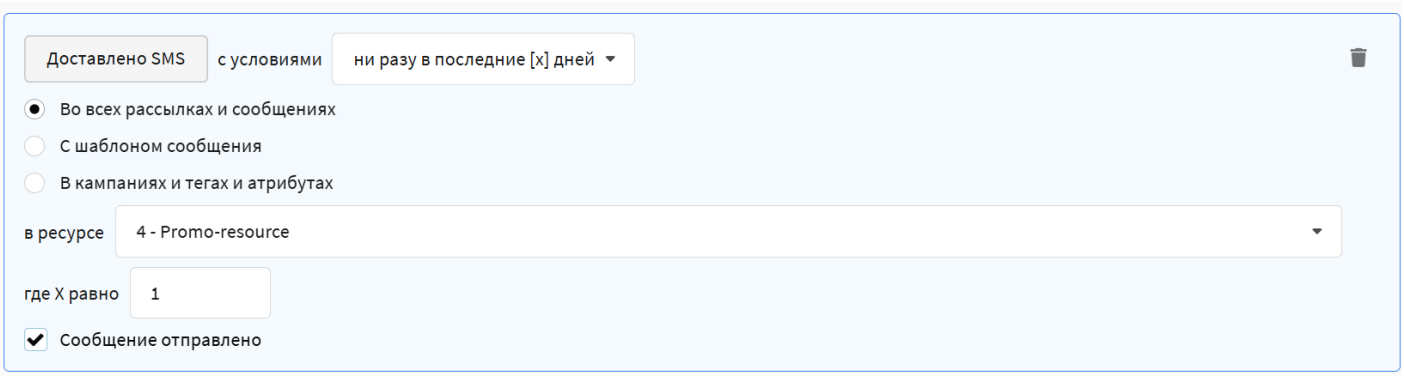
Доставка сообщения

Событие регистрируется, когда платформа получает подтверждение о принятии сообщения для доставки от провайдера, шлюза или оператора связи.

- **Доступно в:** Email, SMS, Push, Telegram Bot, Telegram Group, WhatsApp, Viber, VK Notify.

TIP

Позволяет определить, кому из подписчиков сообщение не было доставлено при временной недоступности канала.



Открытие / Прочтение сообщения

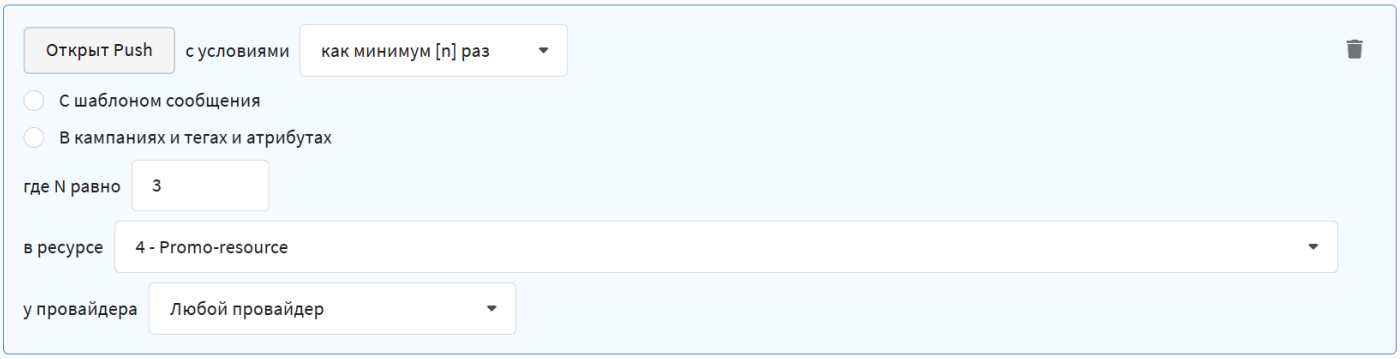
Событие регистрируется, когда пользователь открывает сообщение.

- **Открыт:** Email, SMS, Push.
- **Прочитан:** Email, WhatsApp, Viber, VK Notify.

Для Email открытие фиксируется с помощью пикселя в теле письма.

TIP

Позволяет получить точное количество уникальных открытий и рассчитать Open Rate (OR) для конкретной группы получателей.



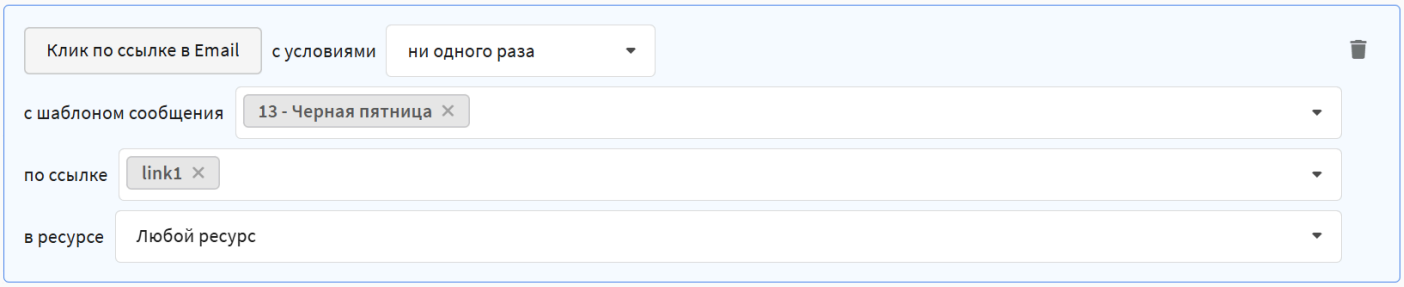
Клики в сообщениях

Различные события, связанные с кликами по ссылкам внутри сообщений.

- **Любой клик в сообщении:** Email, SMS, Push, VK Notify. Выбирает получателей, кликнувших по любой ссылке в сообщении.
- **Клик по ссылке в сообщении:** Email, SMS, Push, Viber, WhatsApp, Telegram Bot, VK Notify. Позволяет указать конкретные шаблоны и ссылки для выборки переходов.
- **Клик по ссылке отписки в сообщении:** Email, SMS, Push, VK Notify. Выбирает получателей, отписавшихся по специальной ссылке отписки.

TIP

Клик по ссылке отписки имеет отдельный тип события и не попадает в выборку по правилу “Клик по ссылке в [канале]”.



Уникальные клики в мессенджерах (Telegram, WhatsApp, Viber, VK Notify)

Для мессенджеров доступна детальная статистика по кликам:

- **Клик** — фиксируется каждый переход по ссылке.
- **Уникальный клик** — событие регистрируется, если получатель кликнул хотя бы по любой ссылке в сообщении (1 уникальный клик на сообщение).
- **Уникальный клик по ссылке** — уникальный клик по конкретной ссылке (фиксируется одно событие для каждой ссылки, даже при многократных кликах).

Недоставка и ошибки отправки

События, связанные с проблемами при отправке или доставке.

- **Недоставка:** Telegram Bot, Telegram Group, WhatsApp, Viber, VK Notify.
- **Нарушение политики отправки:** Telegram Bot, Telegram Group, WhatsApp, VK Notify.
- **Hard bounces** (постоянные ошибки доставки): Telegram Bot.

Управление подпиской в канале

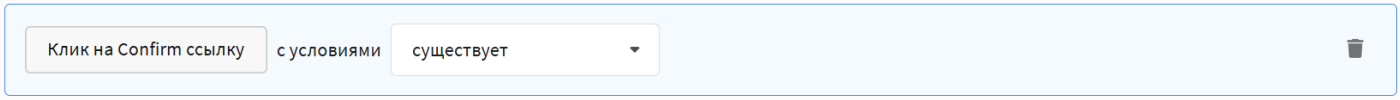
События, связанные с подпиской и отпиской от коммуникации в конкретном канале.

- **Подписался на канал:** Telegram Bot, WhatsApp, Viber, VK Notify.
- **Отписка от канала:** Telegram Bot, WhatsApp, Viber, VK Notify.
- **Подтверждение подписки:** Telegram Bot, Email (Confirm ссылка).

Специфичные события по каналам

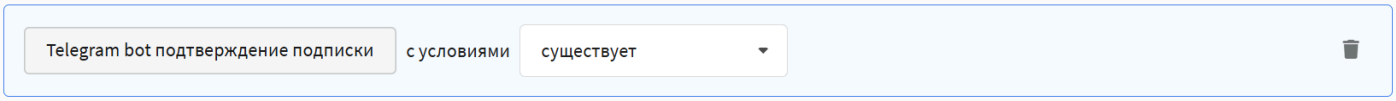
Email

- **Прочитан Email** — срабатывает пиксель `{read}` после 12 секунд открытого письма.
- **Клик на Confirm ссылку** — подтверждение подписки по Double opt-in.
- **Подтверждение подписки (Email)** — событие подтверждения подписки.



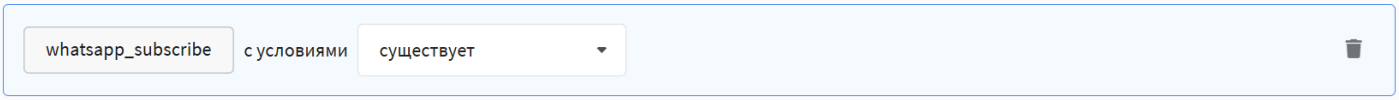
Telegram Bot

- **Telegram bot подтверждение подписки** — подтверждение подписки на бота.



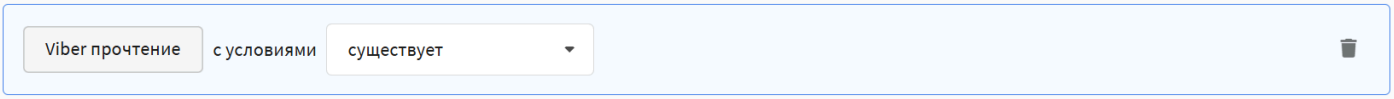
WhatsApp

- **WhatsApp прочтение** — событие прочтения сообщения.



Viber

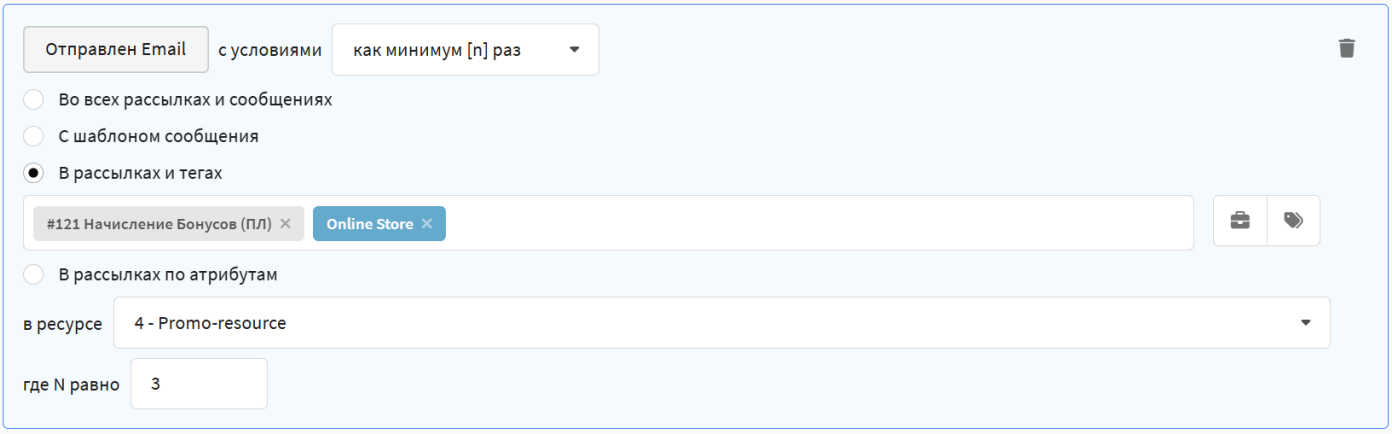
- **Viber прочтение** — событие прочтения сообщения.



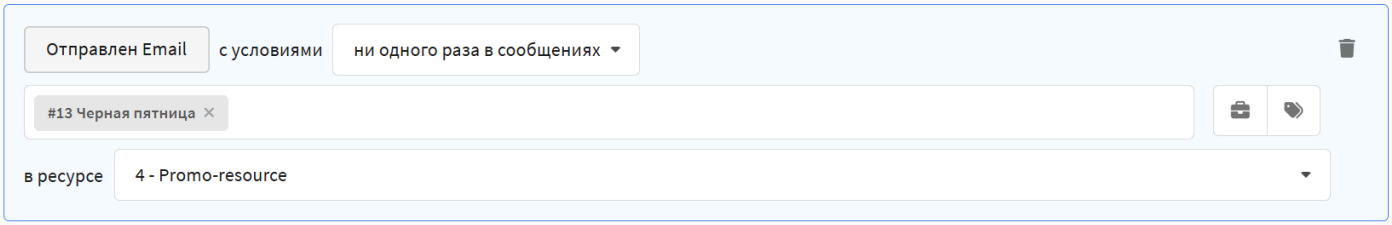
Проверка события в сообщении или рассылке

Для точной проверки, происходило ли событие в определённой рассылке или сообщении, используйте условия с указанием объектов:

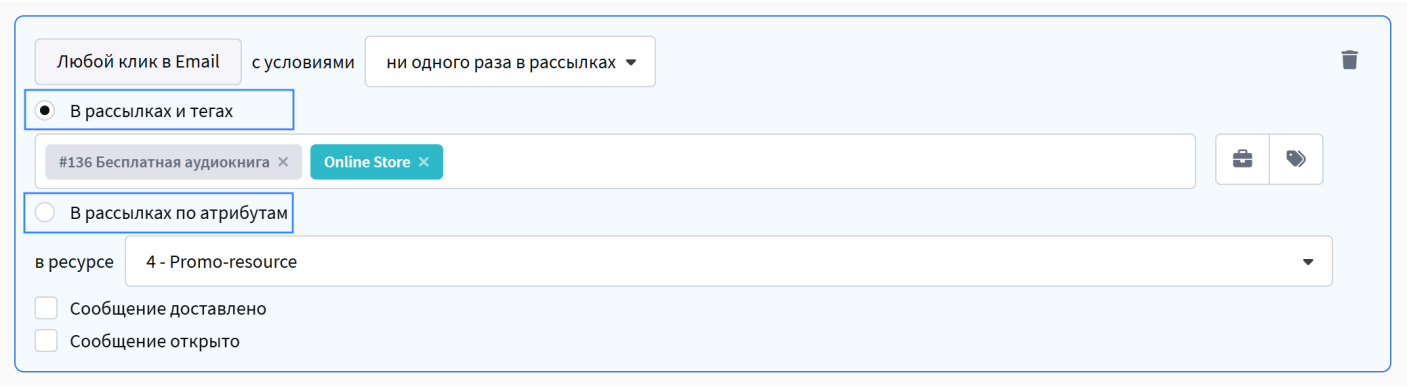
- **Как минимум [n] раз в сообщениях / в рассылках** — профиль попадёт в сегмент, если действие произошло в любом из выбранных шаблонов или рассылок.



- **Ни одного раза в сообщениях / рассылках** — профиль попадёт в сегмент, если действие не произошло в каждом из выбранных объектов.

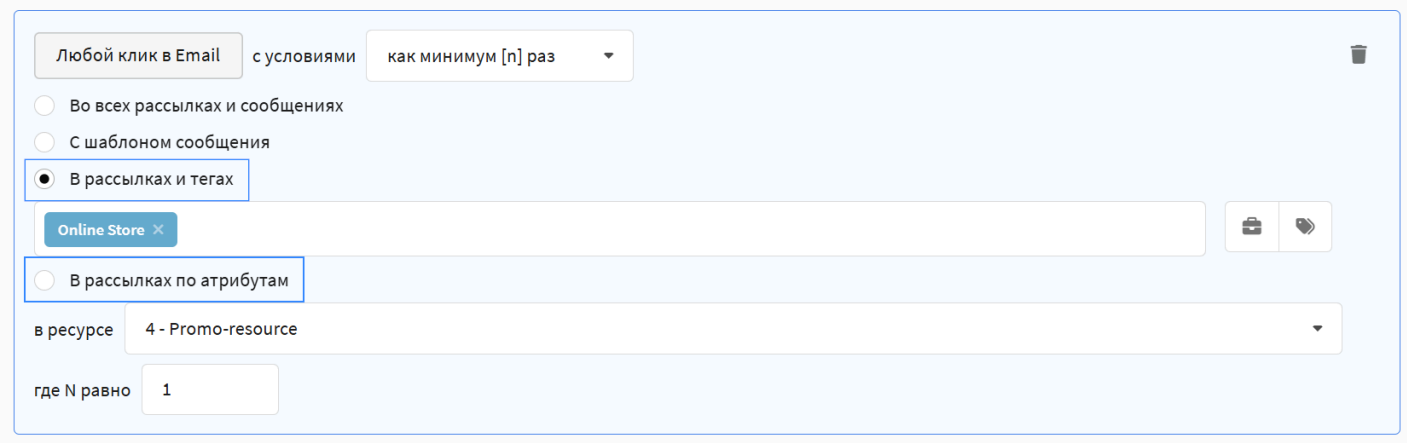


Для обратных условий можно добавить проверку на предшествующее событие (например, для клика — проверку на доставку или открытие).



Отбор кампаний, рассылок или шаблонов по тегам или атрибутам

Доступен отбор объектов для проверки не только поштучно, но и **по тегам** или **по атрибутам**.



** Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

** Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

Шаблоны сообщений

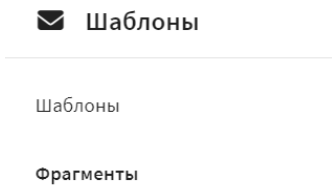
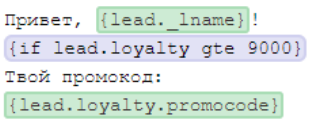
Описание

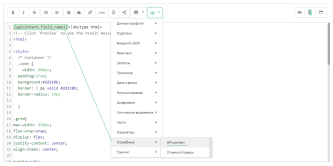
Шаблон сообщения — это форма создания контента для ваших маркетинговых рассылок. Внутри одного шаблона можно настроить контент для разных каналов связи.

Шаблоны в платформе поддерживают несколько видов динамического контента, например, запросы к внешним источникам, сайтам и SQL-базам данных, а также данные, передающиеся в API-запросах на запуск рассылок. Редактор сообщений позволяет добавить в шаблон **переменные персонализации, функции, логические выражения** и **циклы**. Также можно настроить показ уникального контента для каждой конкретной группы подписчиков.

К шаблону можно прикрепить файлы для отправки подписчикам и стоп-списки для блокировки email-адресов. Также шаблоны могут содержать специальные ссылки на подтверждение подписки или отписки от рассылки.

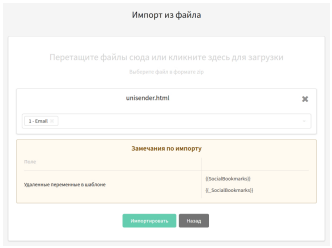
С чего начать?

	Работа с шаблонами сообщений Создание и редактирование шаблонов сообщений. Редакторы контента для Email, Push и SMS версий шаблона. Основные и дополнительные настройки. Персонализация и предпросмотр.
	**Фрагменты шаблонов** Фрагменты — части шаблонов сообщений, которые хранятся отдельно от шаблонов и могут служить для решения различных задач. Фрагмент может содержать простой текст или HTML код для email.
	**Переменные и функции шаблона** В шаблонах используются переменные значения, логические выражения, циклы и функции. В этом справочнике вы найдёте описание синтаксиса и примеры использования для персонализации сообщений.



Динамический контент в сообщениях

Вы можете вставлять в ваши письма динамический контент, например, HTML, API, JSON-контент, а также данных из внешних SQL-таблиц. Подробнее о каждом виде контента вы можете прочитать в этом разделе.



Импорт шаблона из стороннего сервиса

Если у вас уже есть привычный вам шаблон для сообщений, вы легко можете перенести его в платформу Altcraft и использовать в своих маркетинговых рассылках. Узнайте, как это сделать, перейдя в этот справочник.

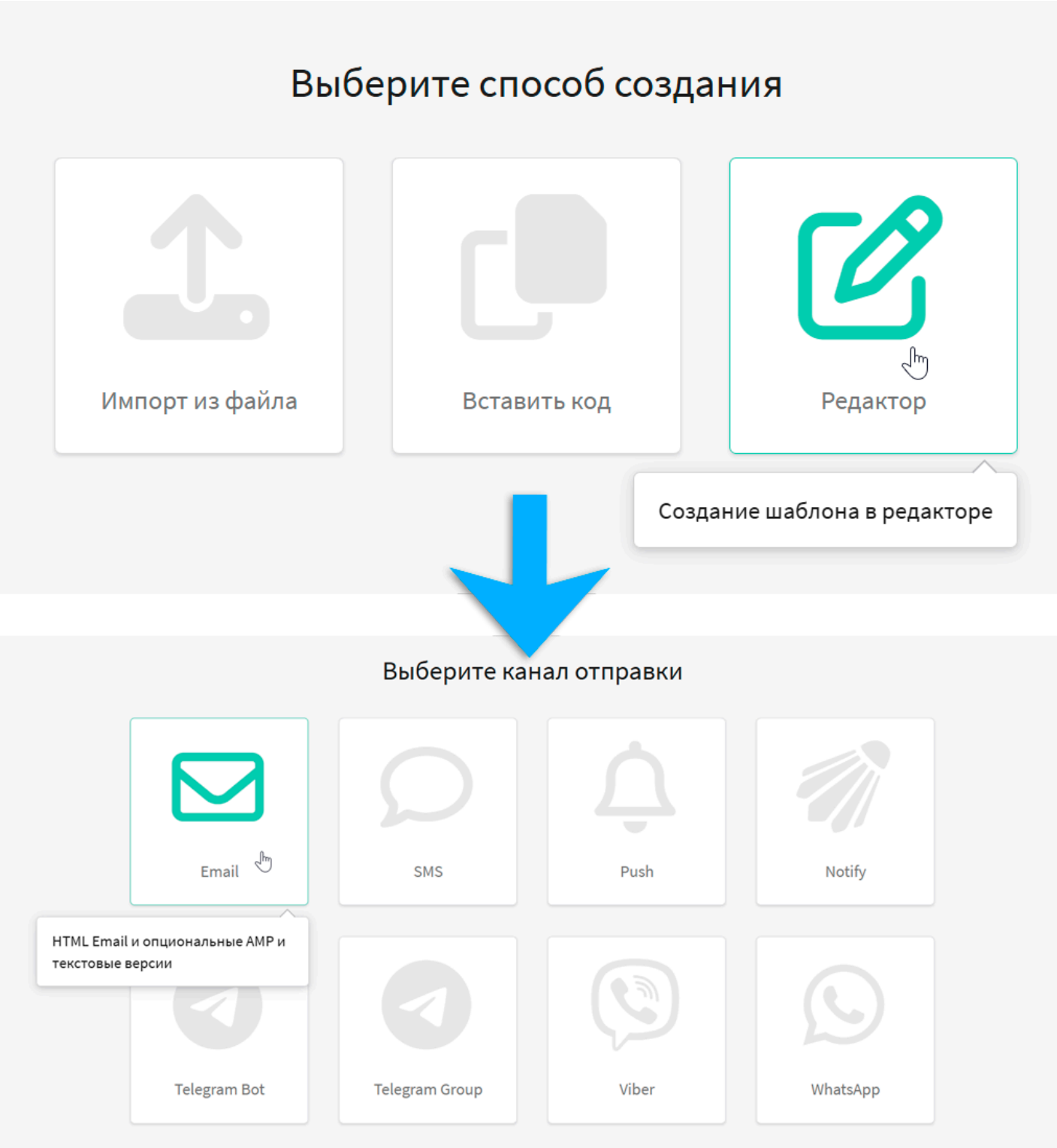
Блочный редактор для email-шаблона

Блочный редактор для email-шаблона — это инструмент для создания, редактирования и настройки маркетинговых email—шаблонов и писем без необходимости написания кода. Он предназначен для маркетологов и дизайнеров, которые хотят быстро собирать профессиональные письма с помощью удобного интерфейса, используя блоки, визуальные элементы и настройки дизайна.

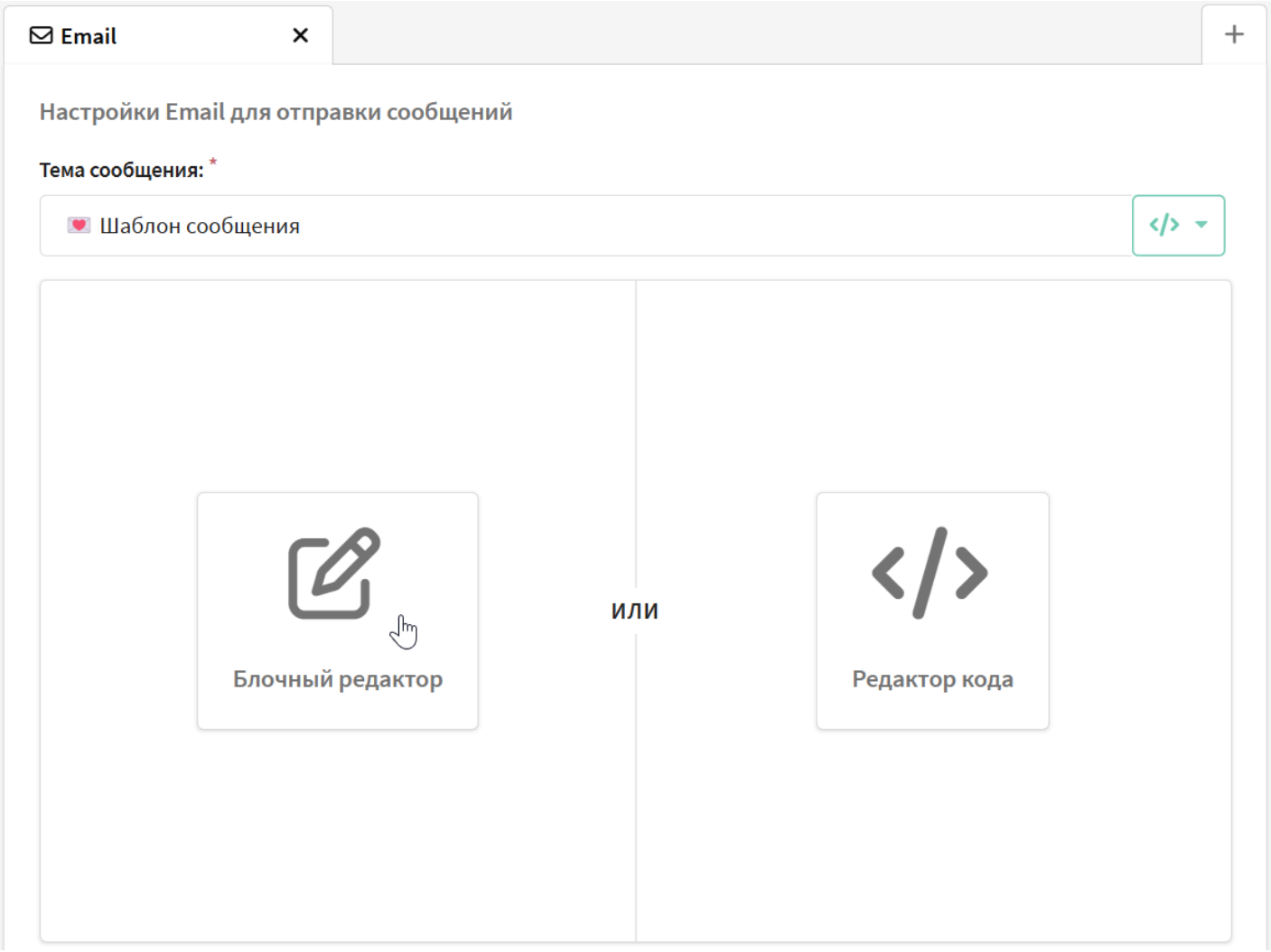
Подробные инструкции, примеры использования и советы по работе с редактором доступны в **документации**.

Начало работы в редакторе

В главном меню откройте раздел **Шаблоны** и создайте новый шаблон сообщения. Выберите **Редактор —> Email**.



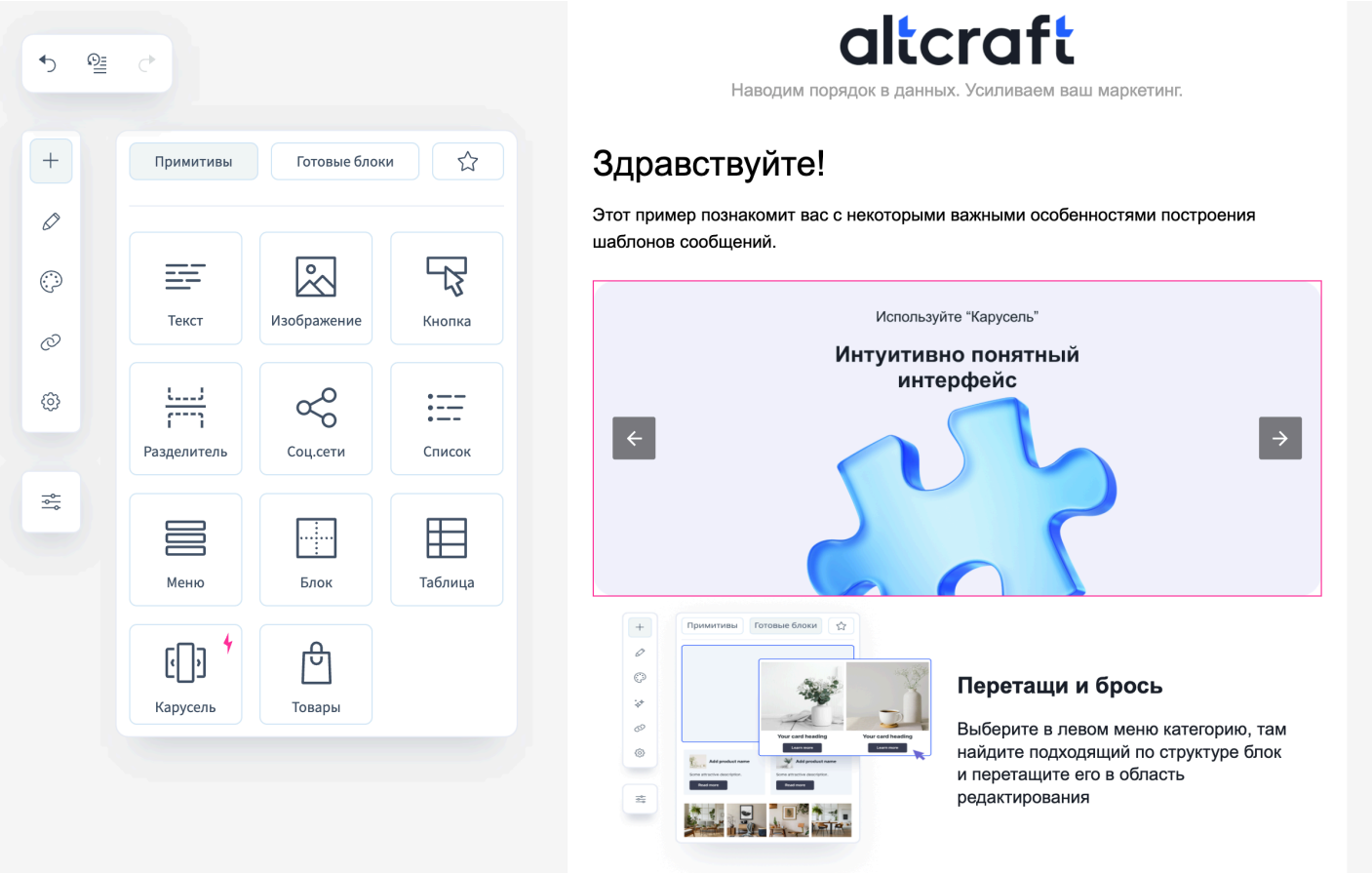
Перед вами появится окно настройки email-шаблона, в нём выберите **“Блочный редактор”**.



Редактирование шаблонов и интерфейс

После того как вы создали или открыли шаблон, вы попадаете в **редактор писем**. Это — основное пространство конструктора. Здесь вы можете добавить новые блоки, изменить свойства компонентов и настроить ссылки.

В верхней панели доступны кнопки **отмены** и **повтора** действий, а также **журнал изменений**.



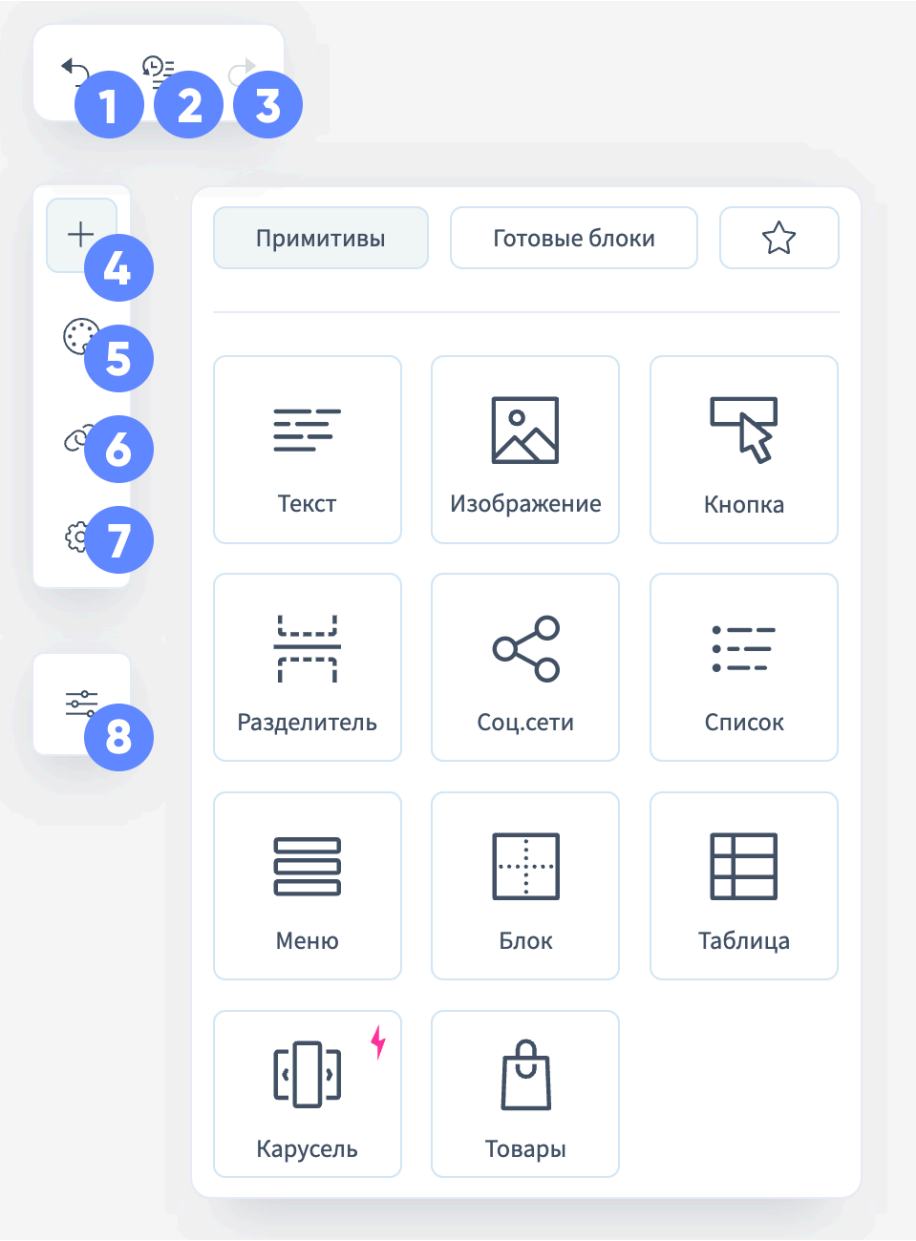
Интерфейс редактора состоит из двух основных частей: **рабочей панели** и **области редактирования**.

На **рабочей панели** размещаются инструменты и блоки, которые можно добавлять в шаблон.

В **области редактирования** отображается само письмо — здесь выполняется визуальное построение и настройка контента.

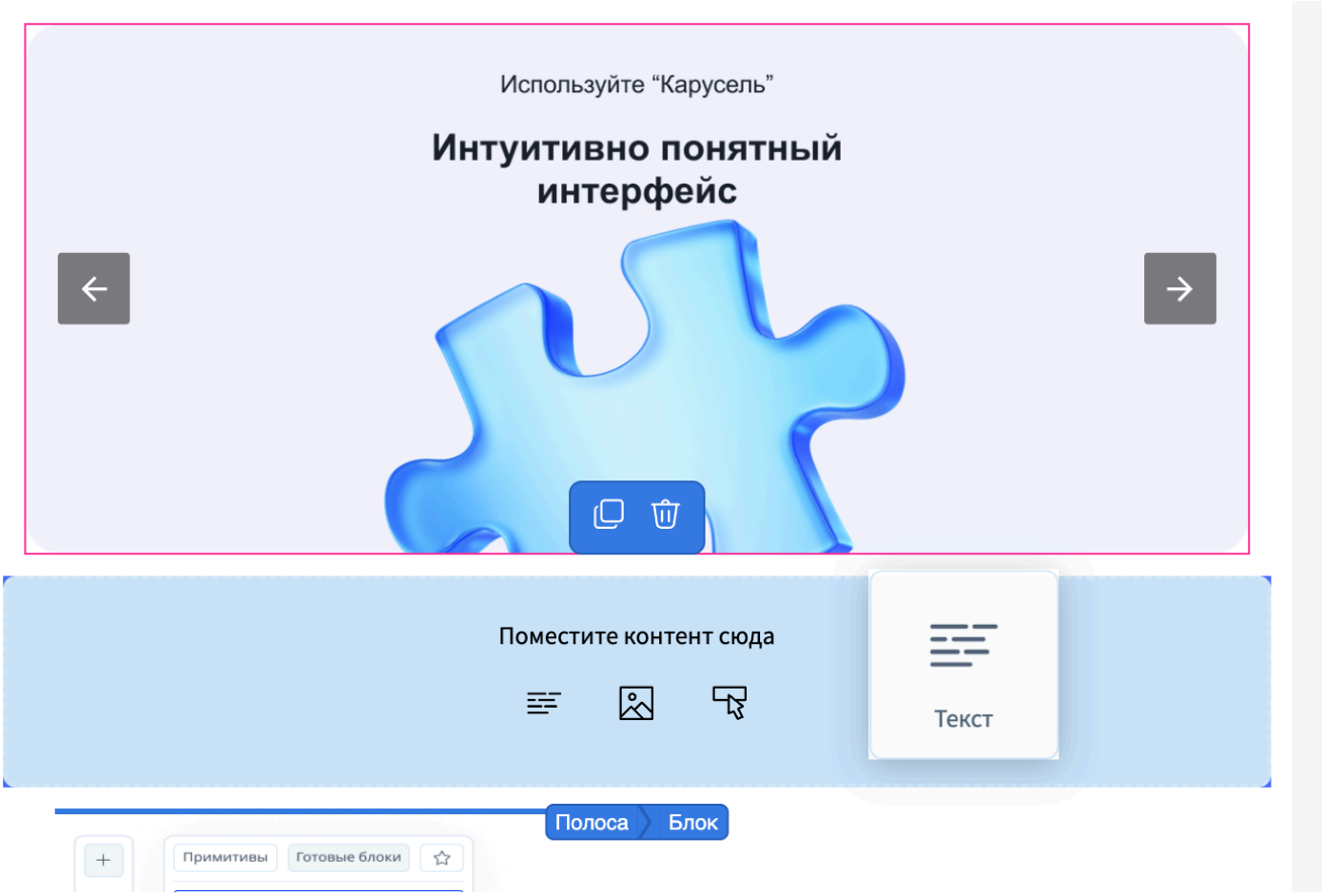
Рабочая панель

Рабочая панель содержит следующие элементы управления:



1. **Отменить** — отменяет последнее выполненное действие;
2. **История действий** — открывает список всех изменений, позволяет вернуться к любой предыдущей версии;
3. **Повторить** — повторяет ранее отменённое действие;
4. **Добавить** — открывает меню примитивов и готовых блоков для вставки в шаблон;
5. **Стили** — панель для настройки оформления шаблона;
6. **Карта ссылок** — отображает все ссылки, используемые в шаблоне, для проверки и редактирования;
7. **Настройки** — настройки темы и прехедера письма;
8. **Параметры** — общие параметры редактора (тема интерфейса и положение панели свойств).

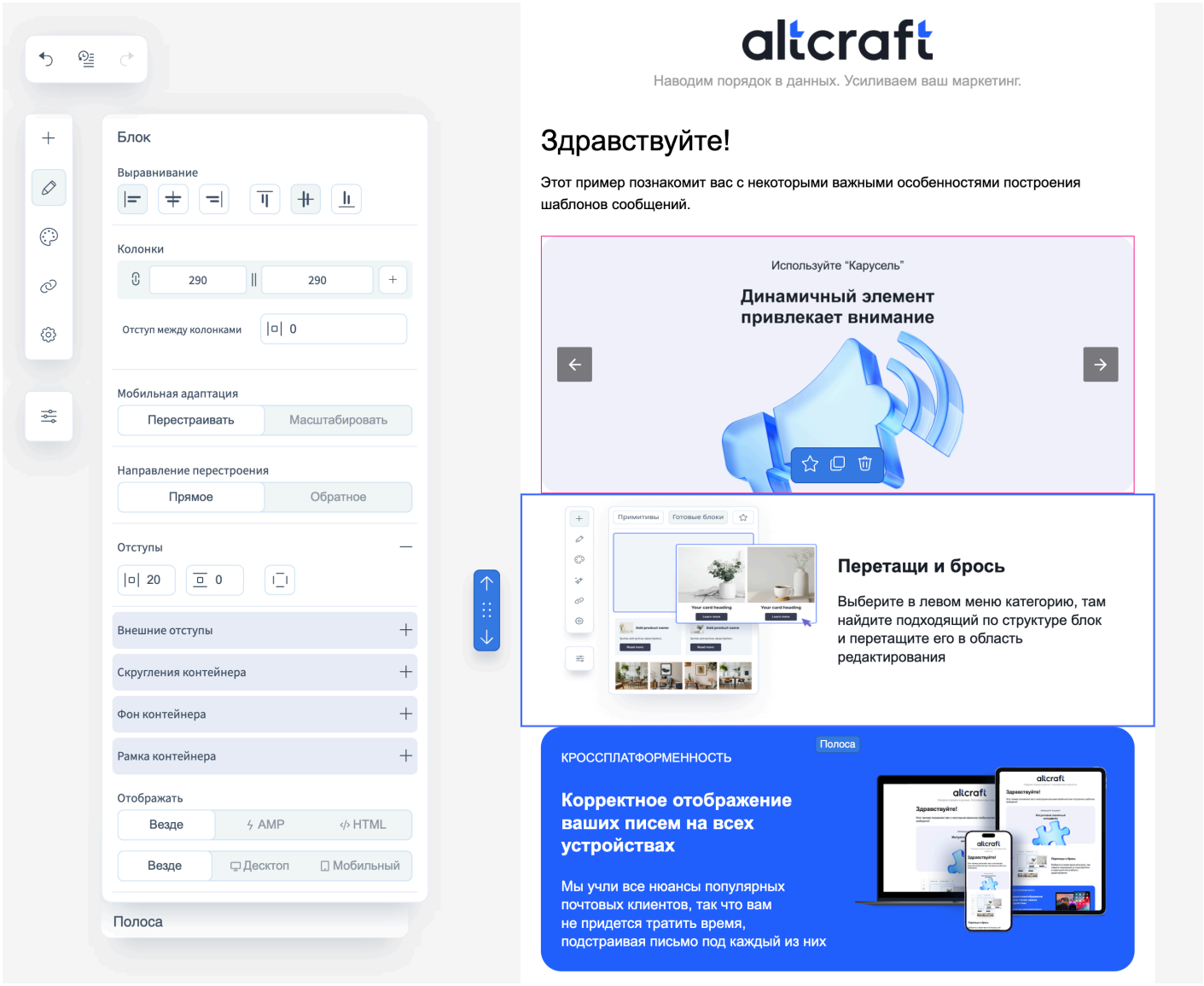
Область редактирования



Главная часть редактора, где происходит сборка письма. Сюда можно перетаскивать блоки из рабочей панели. Каждый элемент подсвечивается границами и может быть изменён напрямую.

Блоки и элементы

Блок

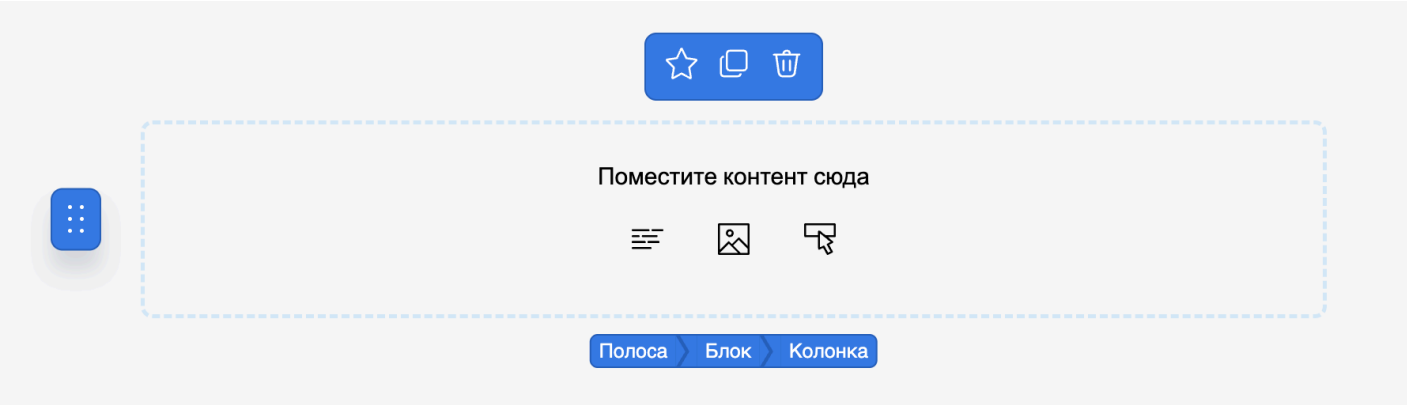


Для блока с контентом настраиваются следующие параметры:

- **Выравнивание** — выравнивание блока по правому, левому краю или по середине письма;
- **Колонки** — здесь настраивается количество колонок, их ширина, а также отступы между ними;
- **Мобильная адаптация** — свойство становится доступно, если в блоке больше одной колонки. Определяет то, как будут выглядеть колонки на мобильных устройствах: **уменьшаться в размере** или **перестраиваться вертикально**, сохраняя размер.
- **Направление перестроения** — порядок колонок, если выбран режим перестроения в прошлом параметре;
- **Отступы** — отступы **контента** блока от **его границ**;
- **Внешние отступы** — отступы **границ блока** от **границ письма**;
- **Скругления контейнера** — настройка скругления углов;
- **Фон контейнера** — здесь можно либо указать цвет фона, либо загрузить фоновое изображение;
- **Рамка контейнера** — цвет, толщина и стиль границ контейнера блока;

- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

Вы можете добавить контент в пустой блок, нажав соответствующие иконки внутри элемента:



Свойства колонок блоков также кастомизируются:

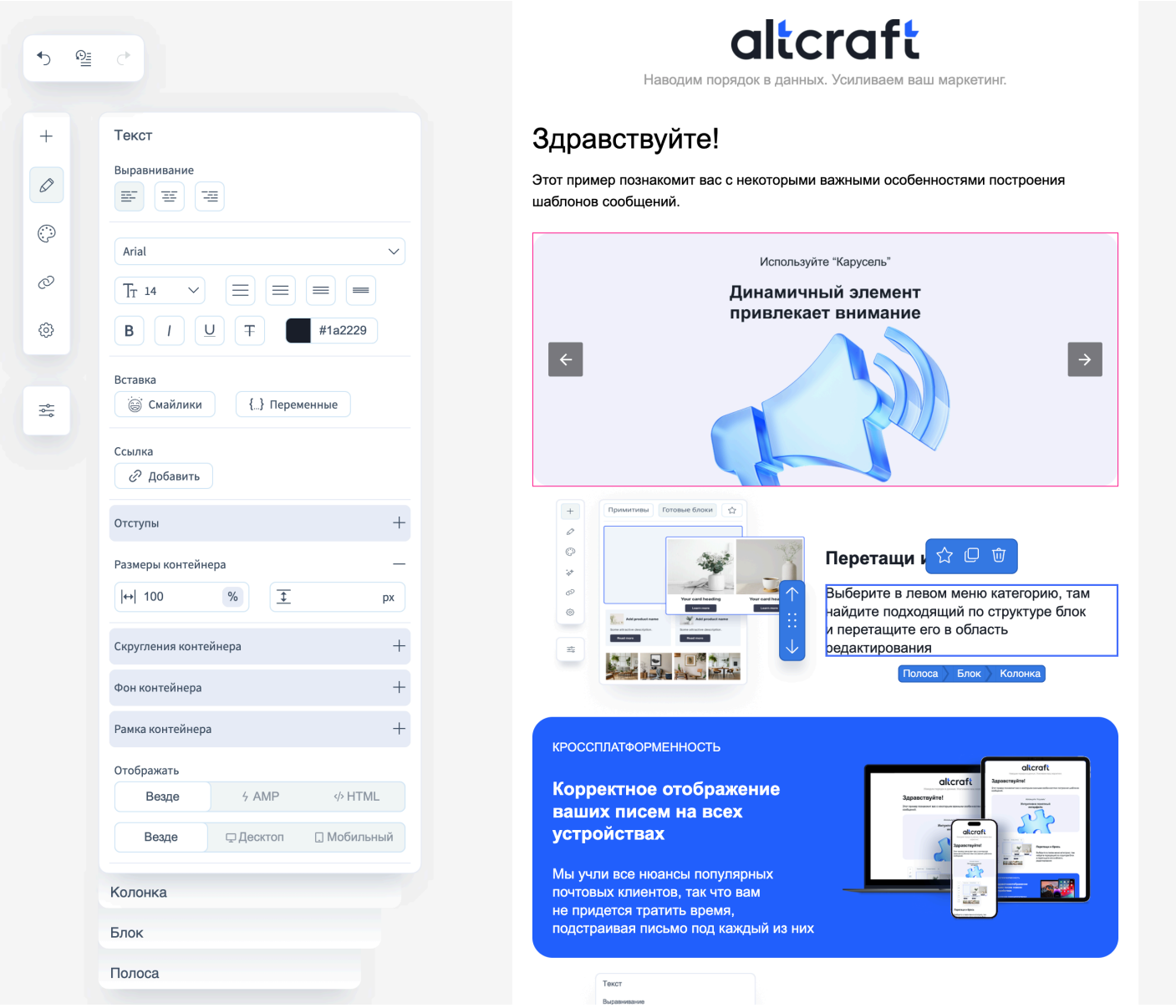
- **Выравнивание**;
- **Отступы**;
- **Фон колонки**;
- **Скругления колонки**;
- **Рамка колонки**;
- **Размеры колонки**.

Для полосы, содержащей блок, можно настроить параметры:

- **Выравнивание**;
- **Фон**;
- **Ширина контента**;
- **Скругление контента**;
- **Рамки контента**;
- **Отступы**.

Некоторые параметры полосы влияют на вид **всего письма**.

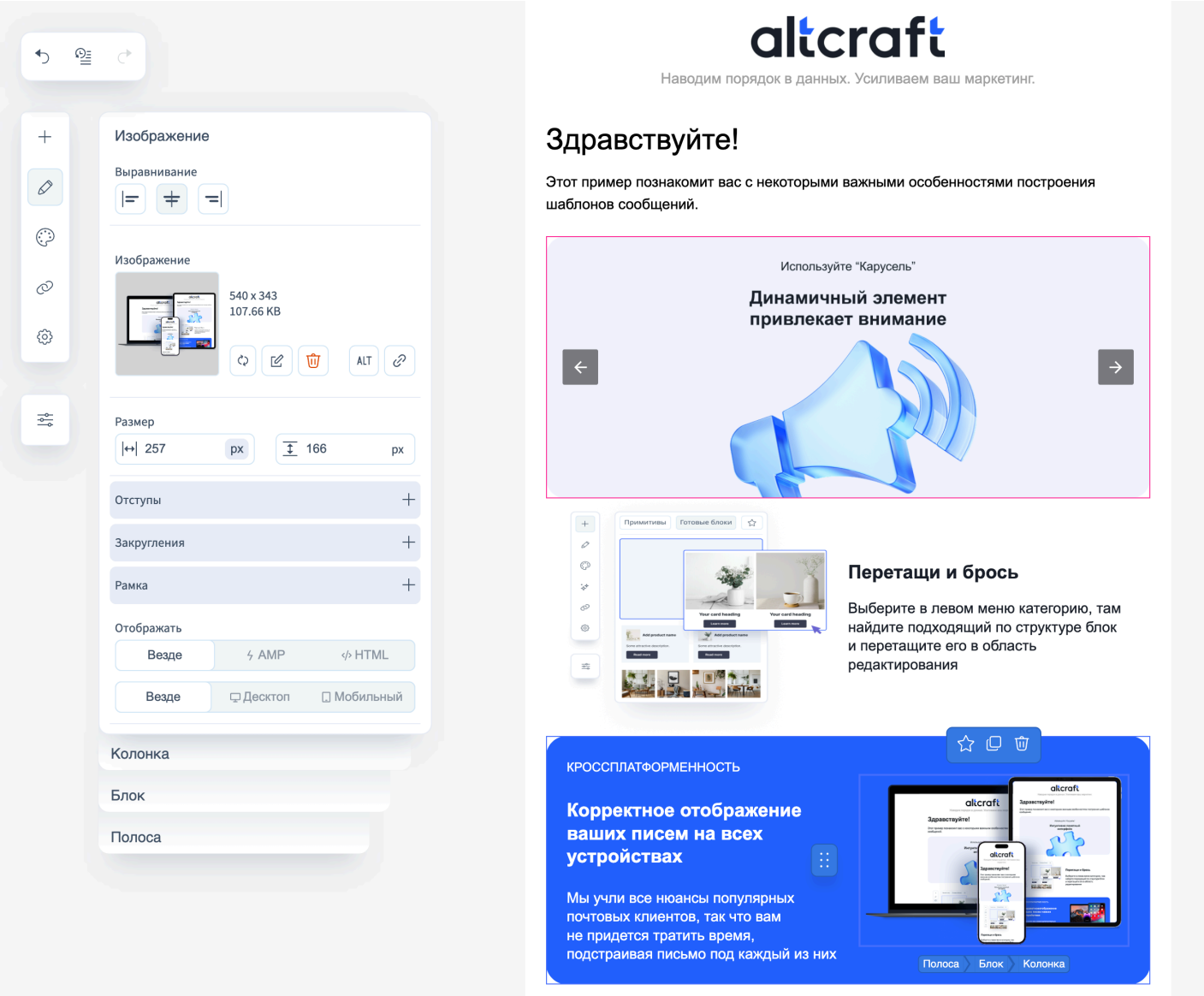
Текст



Для текстового блока настраиваются следующие свойства:

- **Выравнивание** текста внутри контейнера;
- **Шрифт**, его размеры, свойства и цвет;
- Вставка **смайликов** и **переменных**. Переменные указаны для популярных email-сервисов;
- Вставка **ссылок** в текст;
- **Отступы** внутри блока;
- **Размер, скругление, фон** и **рамка** контейнера.
- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

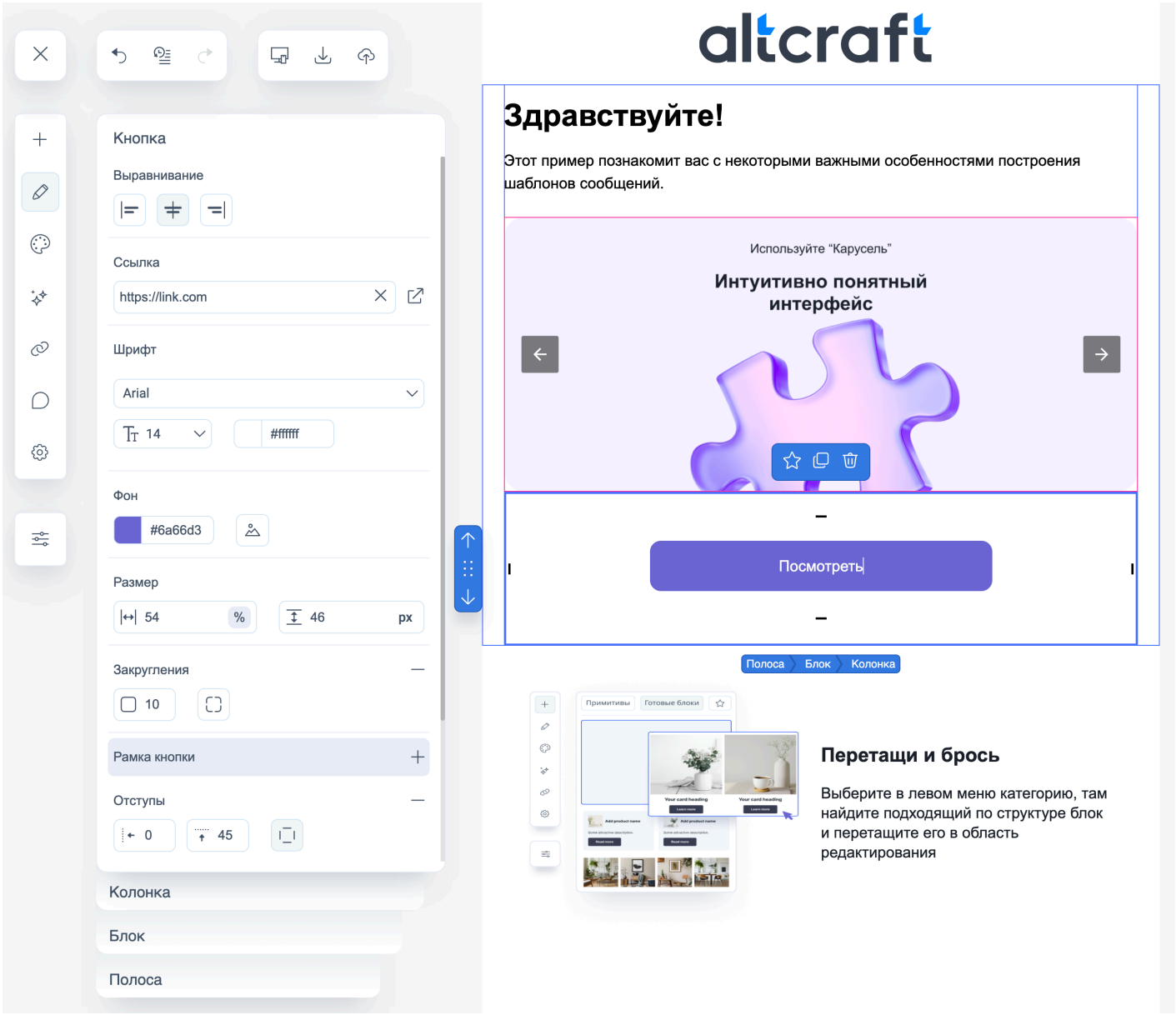
Изображение



Для блока с изображением настраиваются следующие свойства:

- **Выравнивание** изображения внутри контейнера;
- **Размер** изображения;
- **Отступы** внутри блока;
- **Закругление** контейнера и его **рамки**;
- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

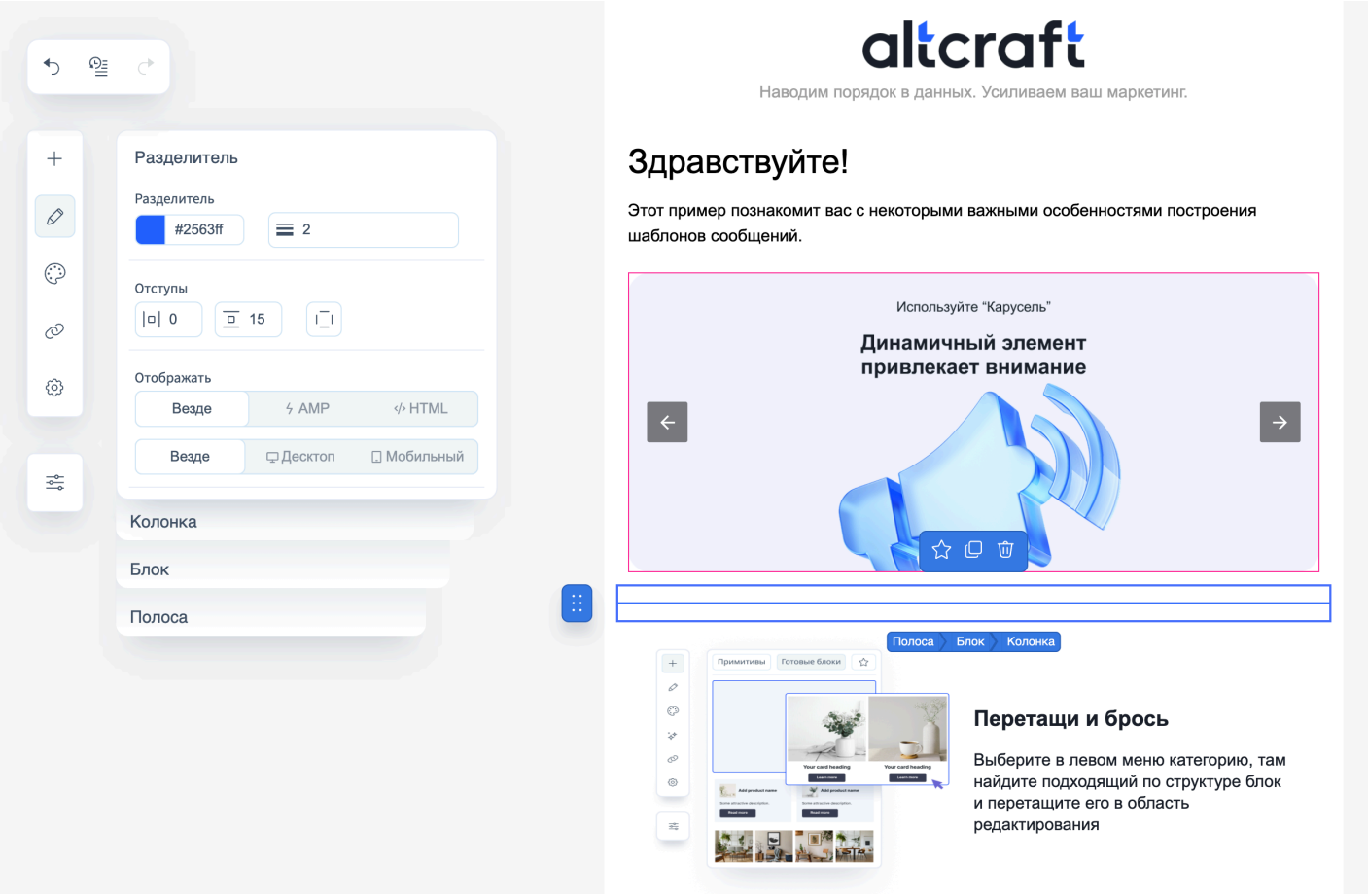
Кнопка



Для кнопок настраиваются следующие свойства:

- **Выравнивание** кнопки внутри контейнера;
- **Ссылка** для перехода по клику;
- **Шрифт** текста, его **размер** и **цвет**;
- **Фон** кнопки. Можно загрузить изображение;
- **Размер** кнопки, **стиль** и **закругление** ее рамок, а также **отступы** внутри контейнера;
- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

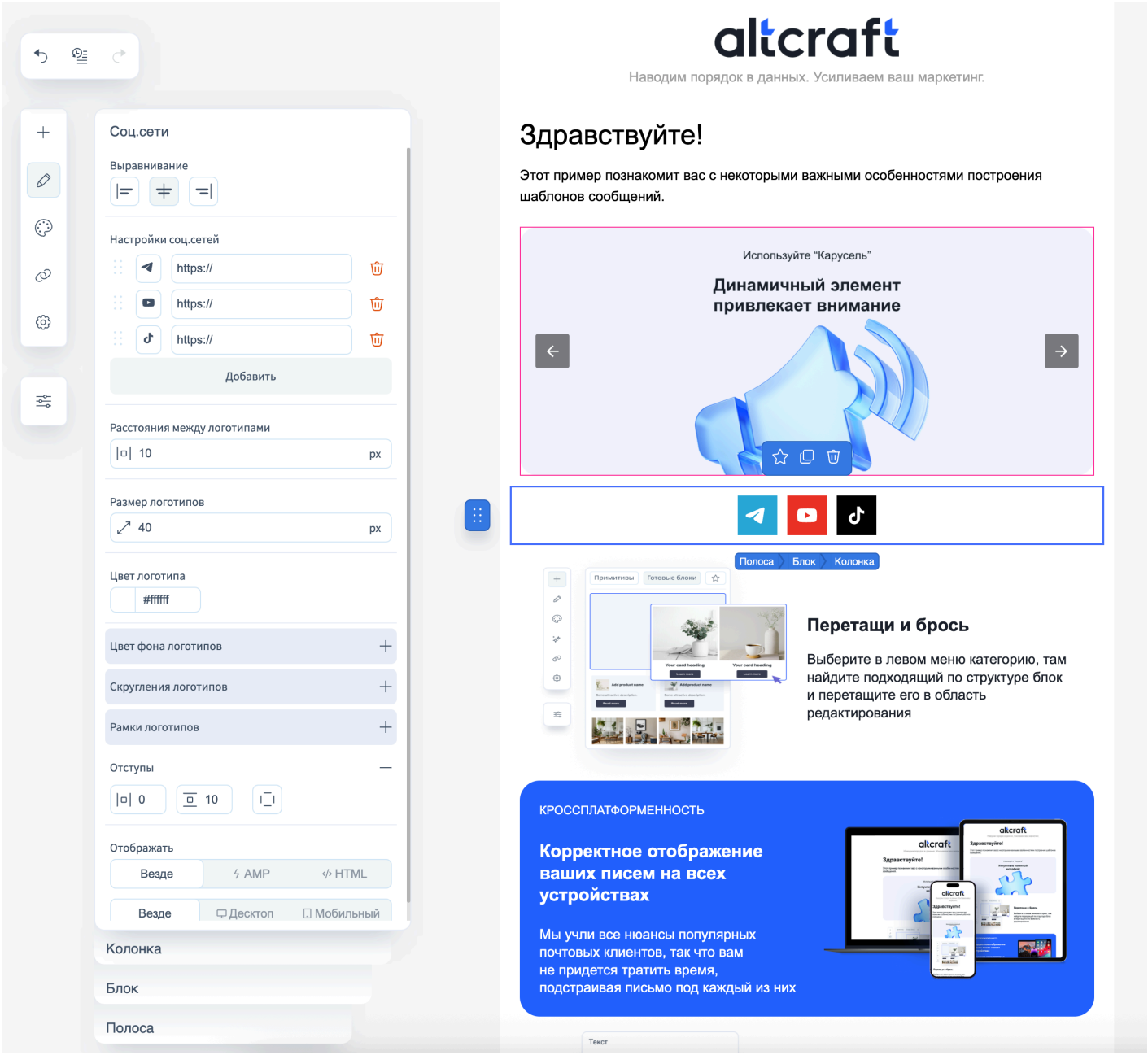
Разделитель



Разделитель представляет собой пустой блок с горизонтальной полосой внутри. Для него настраиваются:

- **Цвет** и **толщина** разделителя;
- **Отступы** внутри контейнера;
- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

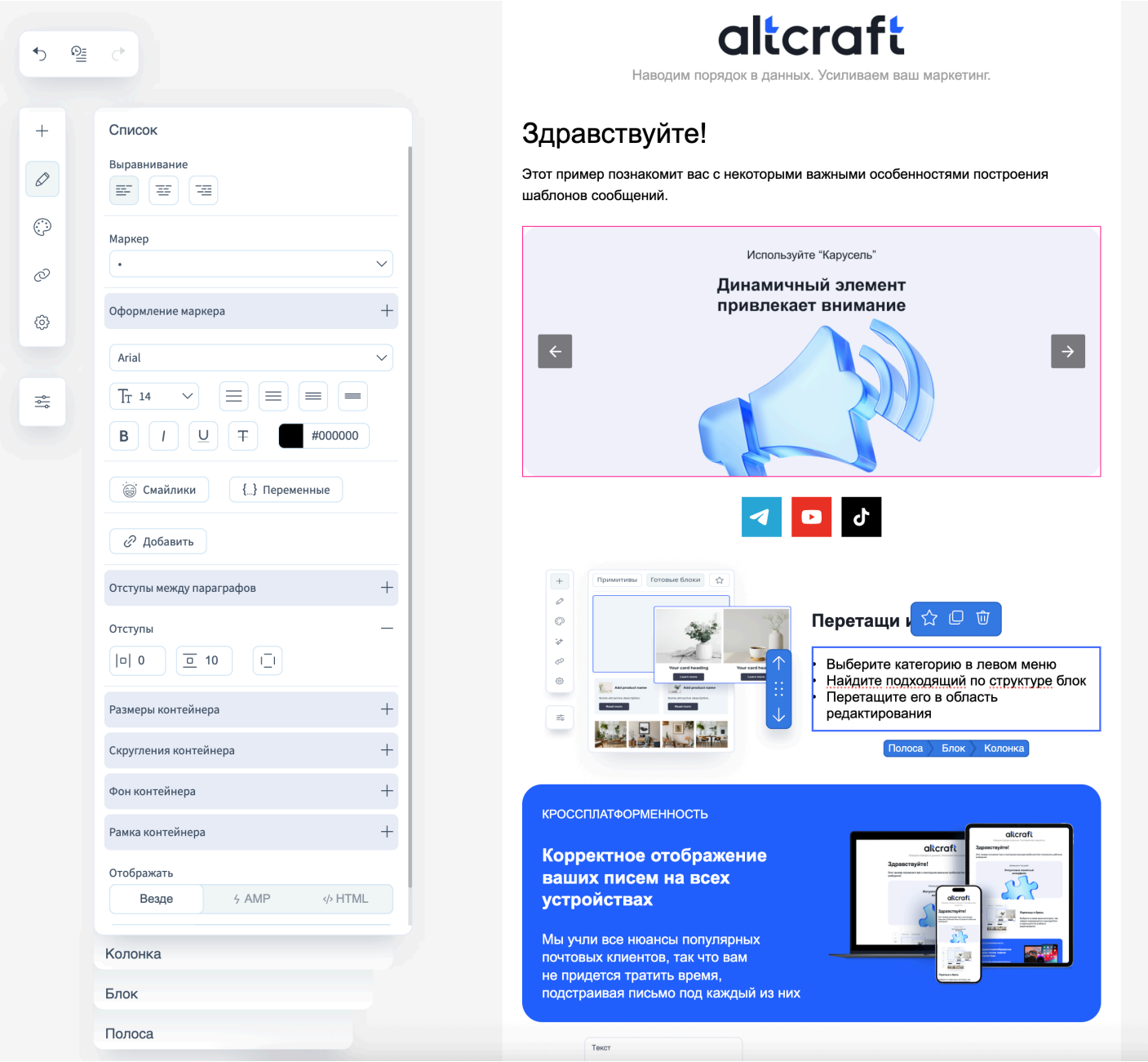
Социальные сети



Соц.сети — блок с кликабельными иконками основных социальных сетей. Для блока можно настроить:

- **Выравнивание** внутри контейнера;
- **Настройки соц.сетей** — здесь можно добавить или убрать иконки, изменить порядок их отображения и настроить ссылки для перехода по клику. Вы также можете перейти к любой иконке и настроить для нее отдельный цвет фона и логотипа;
- **Расстояние** между логотипами, **размер** логотипов;
- **Цвет логотипа**, а также **цвет фона**. Их можно также настроить отдельно для каждой соц.сети, для этого кликните на нужную иконку в блоке;
- **Отступы** внутри контейнера, **скругления** и **рамки** логотипов;
- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

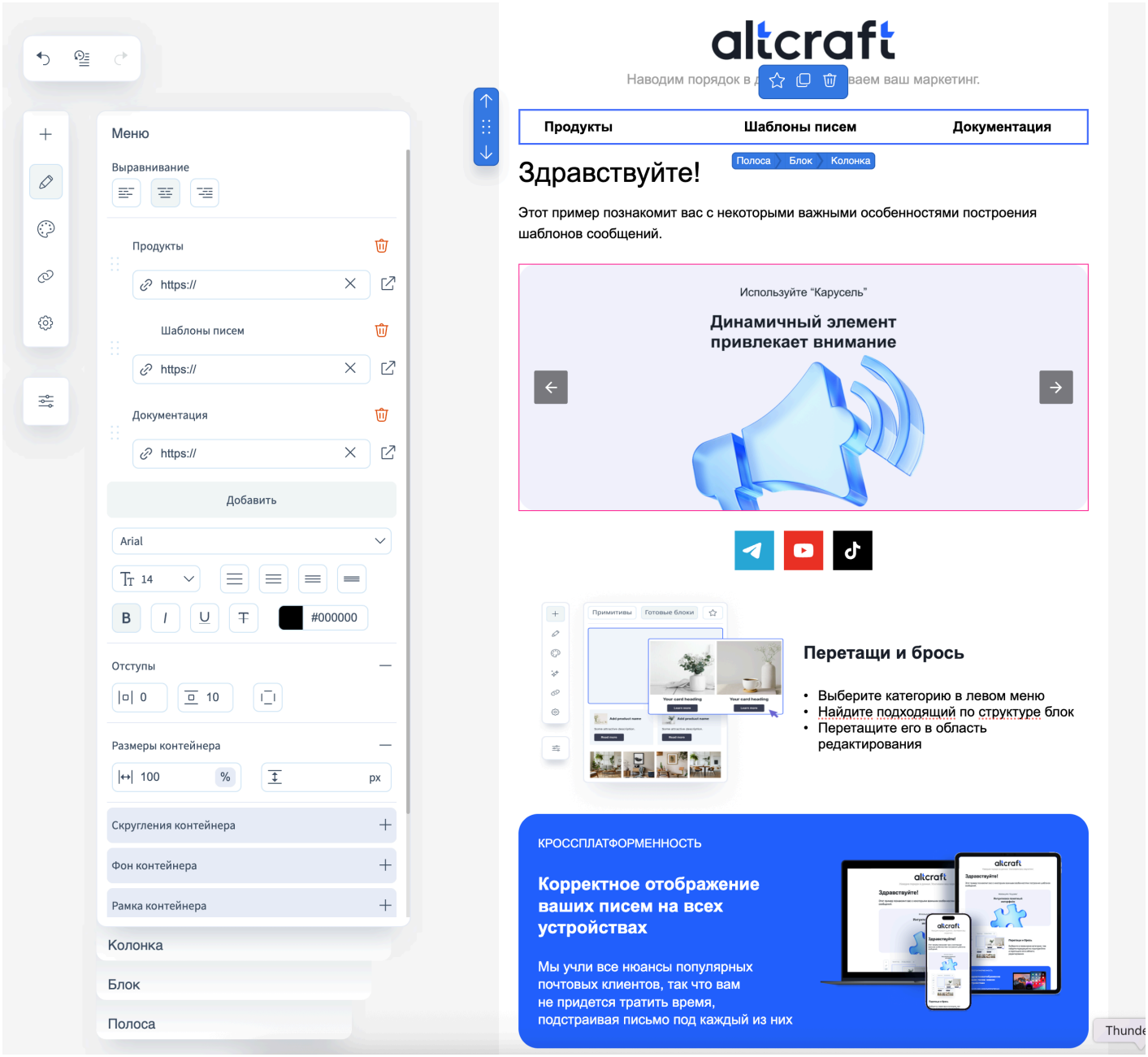
Список



Для списков настраиваются следующие свойства:

- **Выравнивание** внутри контейнера;
- Стиль **маркера**, а также **размер**, **цвет** и **шрифт** (если выбран цифровой маркер);
- **Размер** и **фон** контейнера, **стиль** и **скругление** его рамок, а также **отступы** внутри контейнера;
- Свойства **текста** — размер, цвет, шрифт, стиль, а также вставка переменных и смайликов. Здесь же можно вставить ссылки в текст;
- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

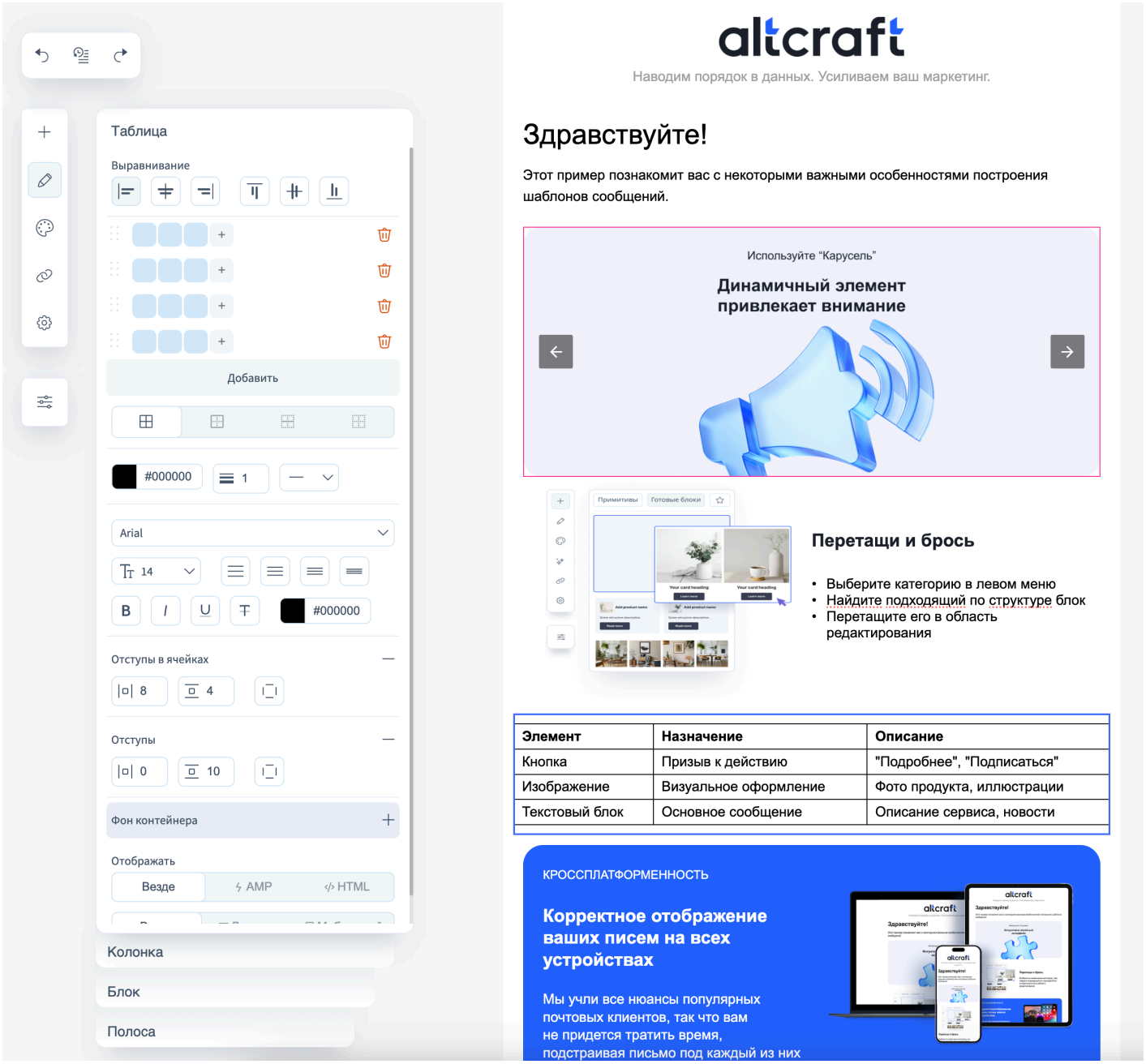
Меню



Для меню настраиваются следующие свойства:

- **Выравнивание** внутри контейнера;
- **Пункты** в меню и их порядок. Здесь можно не только указать текст, но и добавить ссылку для перехода;
- Свойства **текста** — размер, цвет, шрифт, стиль;
- **Размер** и **фон** контейнера, **стиль** и **скругление** его рамок, а также **отступы** внутри контейнера;
- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

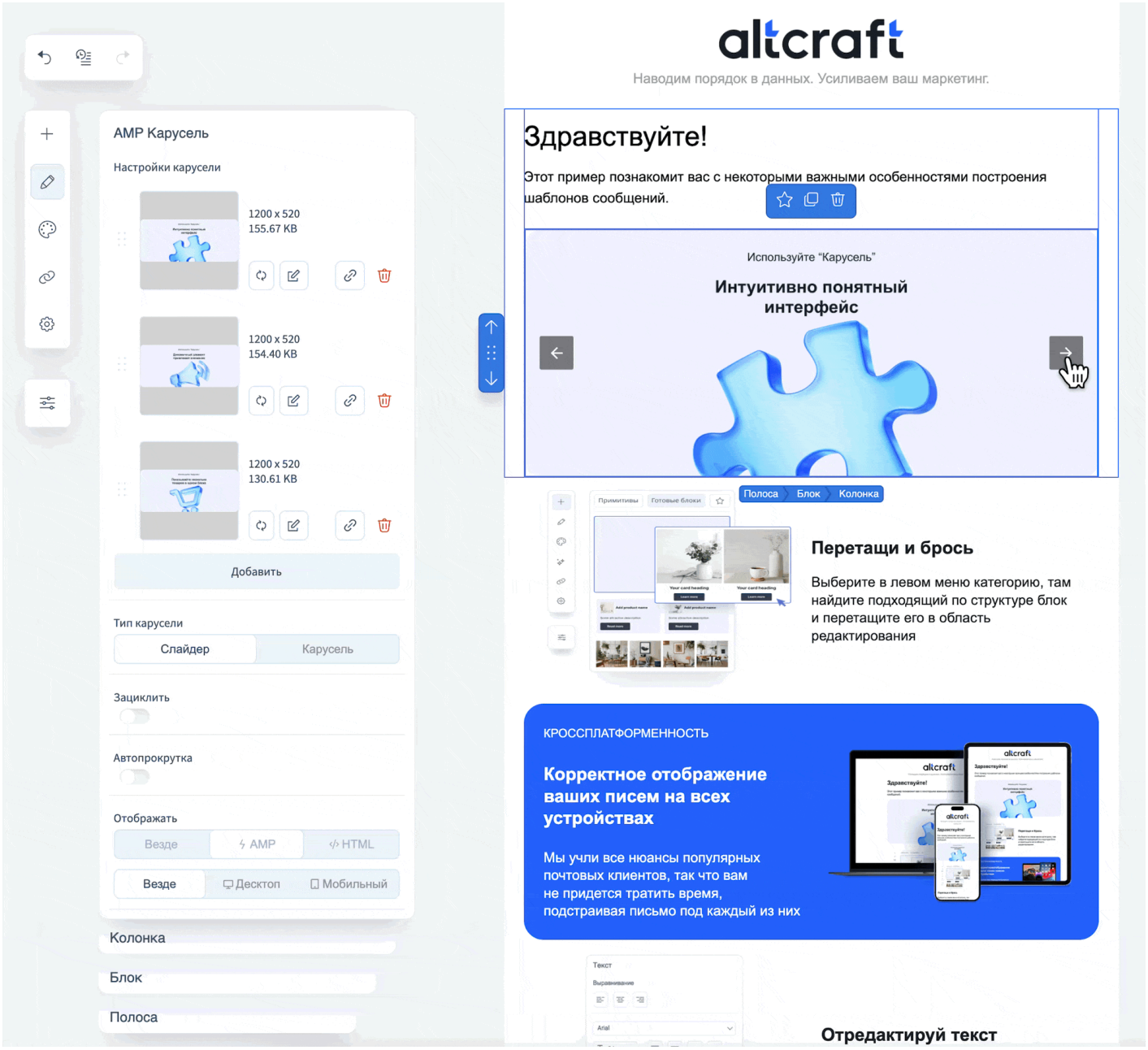
Таблица



Для таблиц настраиваются следующие свойства:

- **Выравнивание** текста внутри таблицы;
- **Размеры** таблицы, а также стили и толщина границ между ячейками;
- Свойства **текста** — размер, цвет, шрифт, стиль;
- Отступы **внутри ячеек**;
- Отступы **внутри контейнера**;
- **Фон** контейнера;
- **Отображение** — определяет отображение блока в разных версиях письма (HTML/AMP/везде, десктоп/мобильное).

Карусель

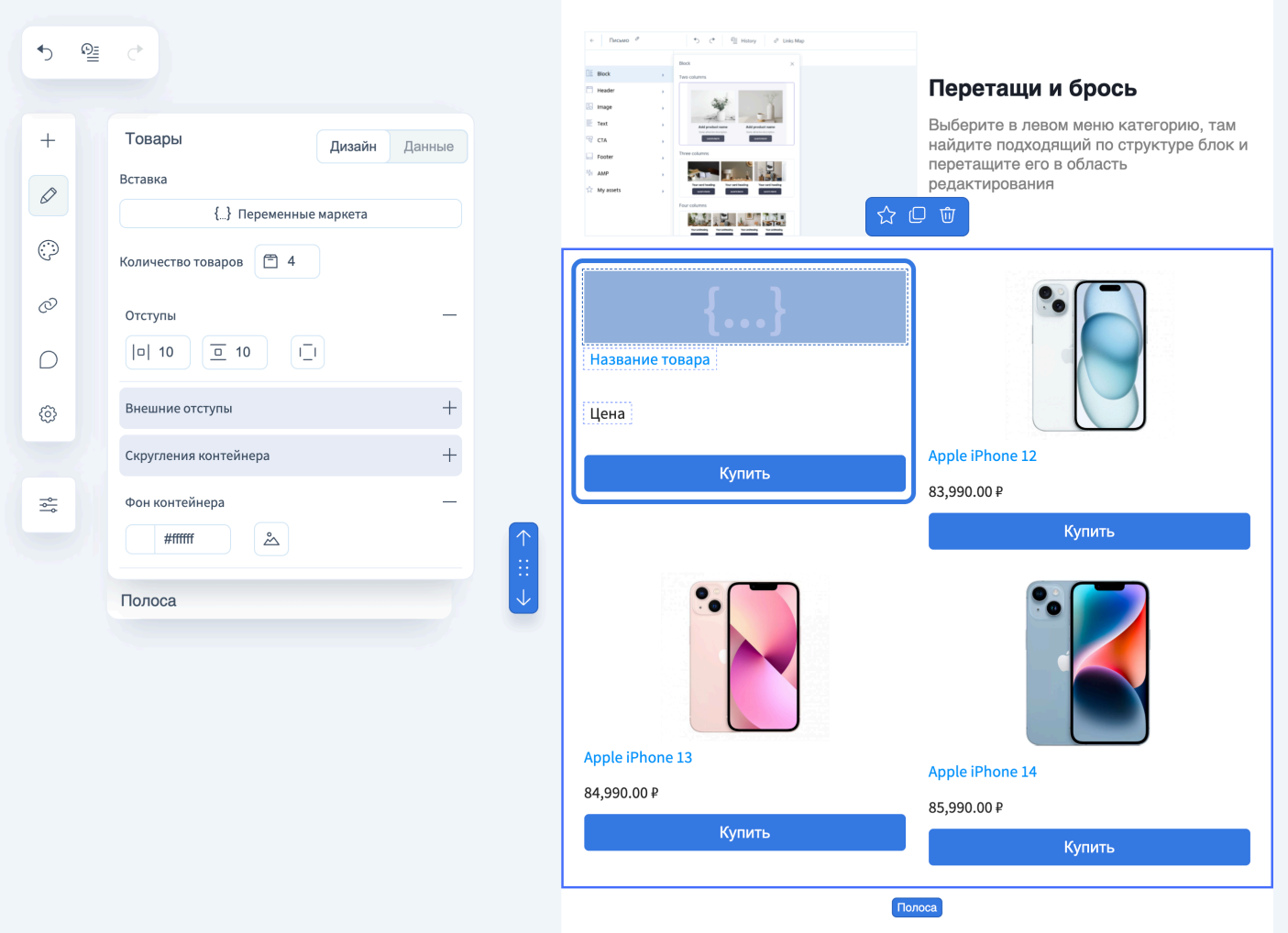


Карусель — AMP-компонент, представляющий из себя несколько прокручивающихся изображений. Для карусели доступны следующие настройки:

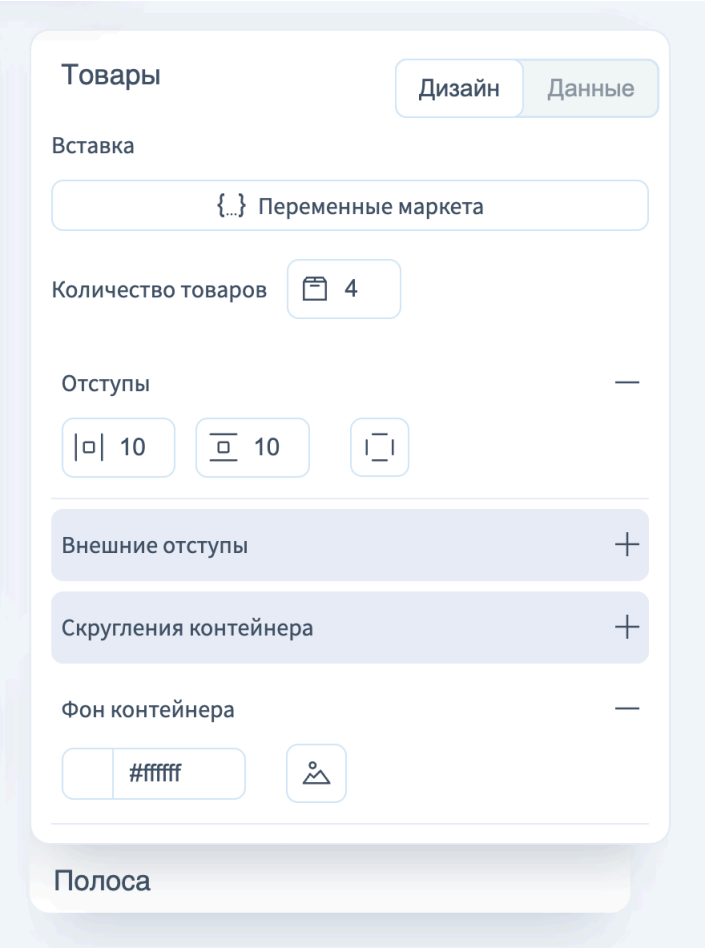
- **Изображения**, которые составляют карусель. Можно загрузить файл с устройства, выбрать из галереи Pixcraft, а также добавить ссылку для перехода по клику;
- **Тип** карусели. Доступна карусель-слайдер (отображается одна картинка) и классическая карусель (отображается несколько картинок).
- Для карусели-слайдера можно включить **зацикливание** карусели и автопрокрутку после указанного времени ожидания;
- В классической карусели настраивается **высота** ее изображений;
- **Отображение** — определяет отображение блока в разных версиях письма (десктоп/мобильное). Поскольку это AMP-компонент, он **не будет отображаться** в HTML-письмах.

Товары

Блок **Товары** используется для динамического формирования товарных блоков в рассылках — например, подборок популярных, акционных или рекомендованных товаров, синхронизированных с актуальными данными из CDP.



Дизайн

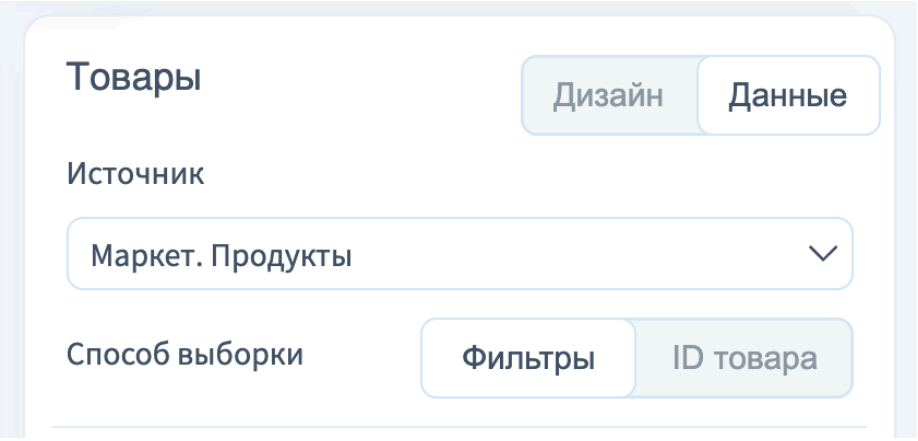


Вкладка **Дизайн** отвечает за визуальное оформление карточек товаров и содержит следующие свойства:

- **Переменные маркета** — определяют, какие элементы отображаются в карточке: название, цена, изображение, кнопка и т. д.
- **Количество товаров** — регулирует количество карточек, отображаемых в строке.
- **Отступы** — задают внутренние и внешние поля вокруг карточек.
- **Скругление контейнера** — позволяет задать радиус углов карточек.
- **Фон контейнера** — устанавливает цвет или изображение фона.

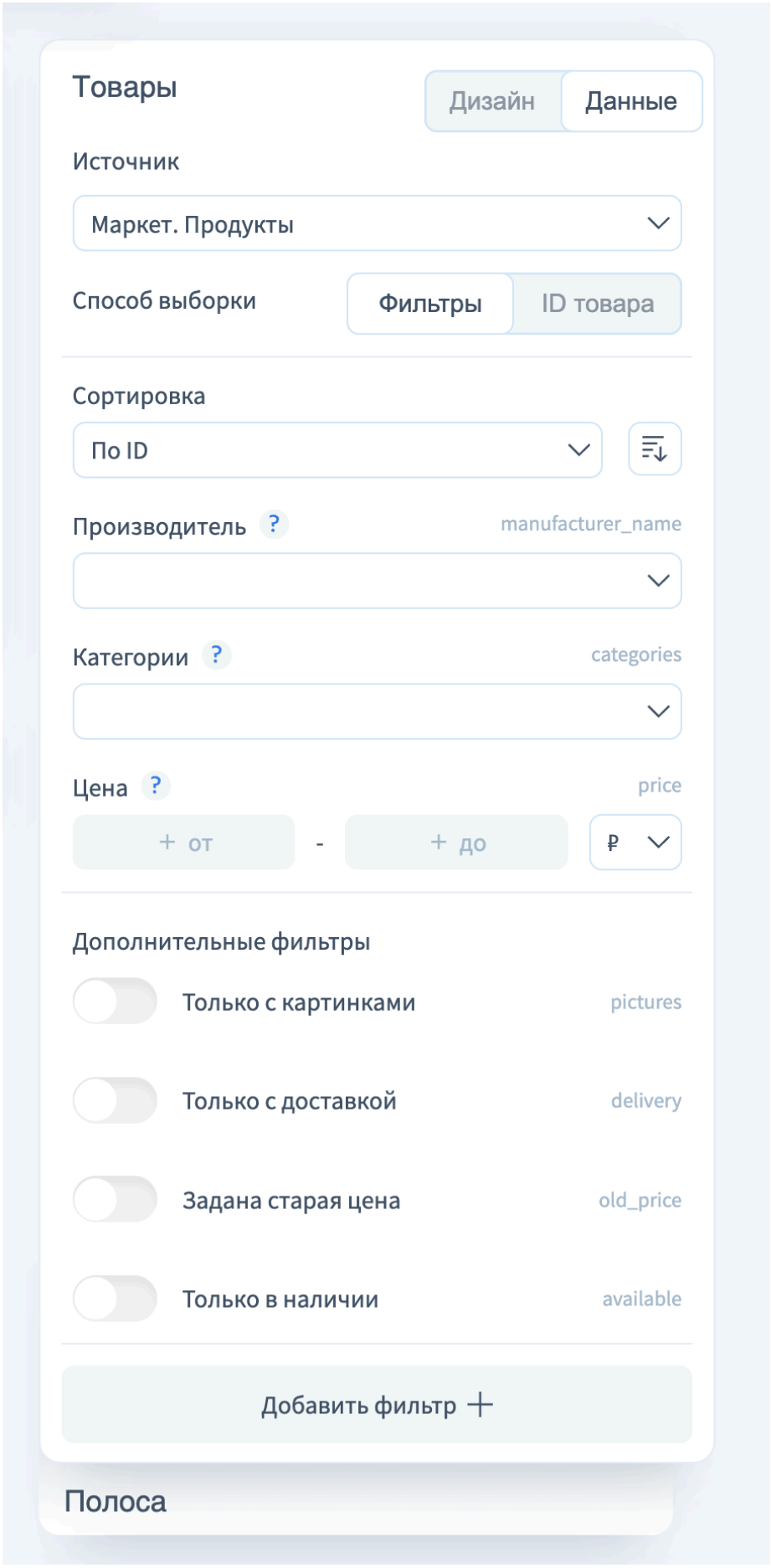
Все изменения, внесённые во вкладке, применяются сразу ко всем карточкам блока.

Данные

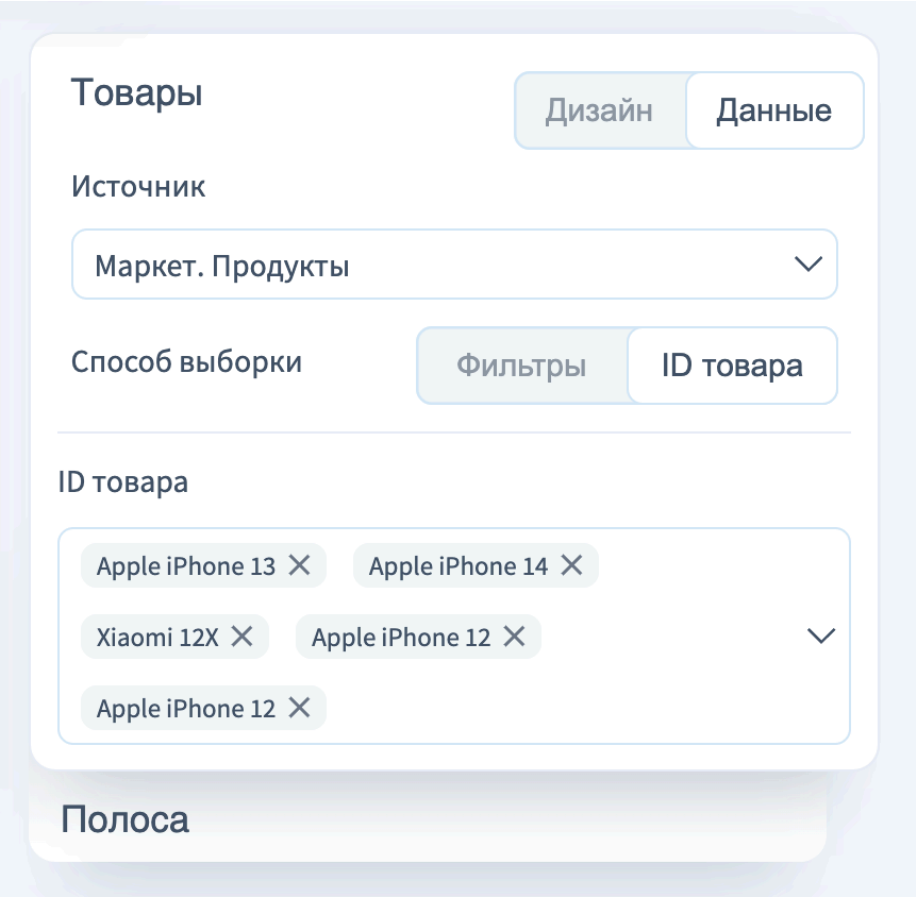


Вкладка **Данные** предназначен для настройки источников и параметров выборки товаров из Altcraft CDP, содержит следующие свойства:

- **Источник** — выбор источника, из которого загружаются данные о товарах.
- **Способ выборки** — определяет метод получения данных:
    - **Фильтры** — режим для динамической выборки товаров по заданным критериям (**сортировка**, **производитель**, **категории**, **цена**). Можно использовать дополнительные фильтры: **только с картинками**, **только с доставкой**, **задана старая цена**, **только в наличии**.

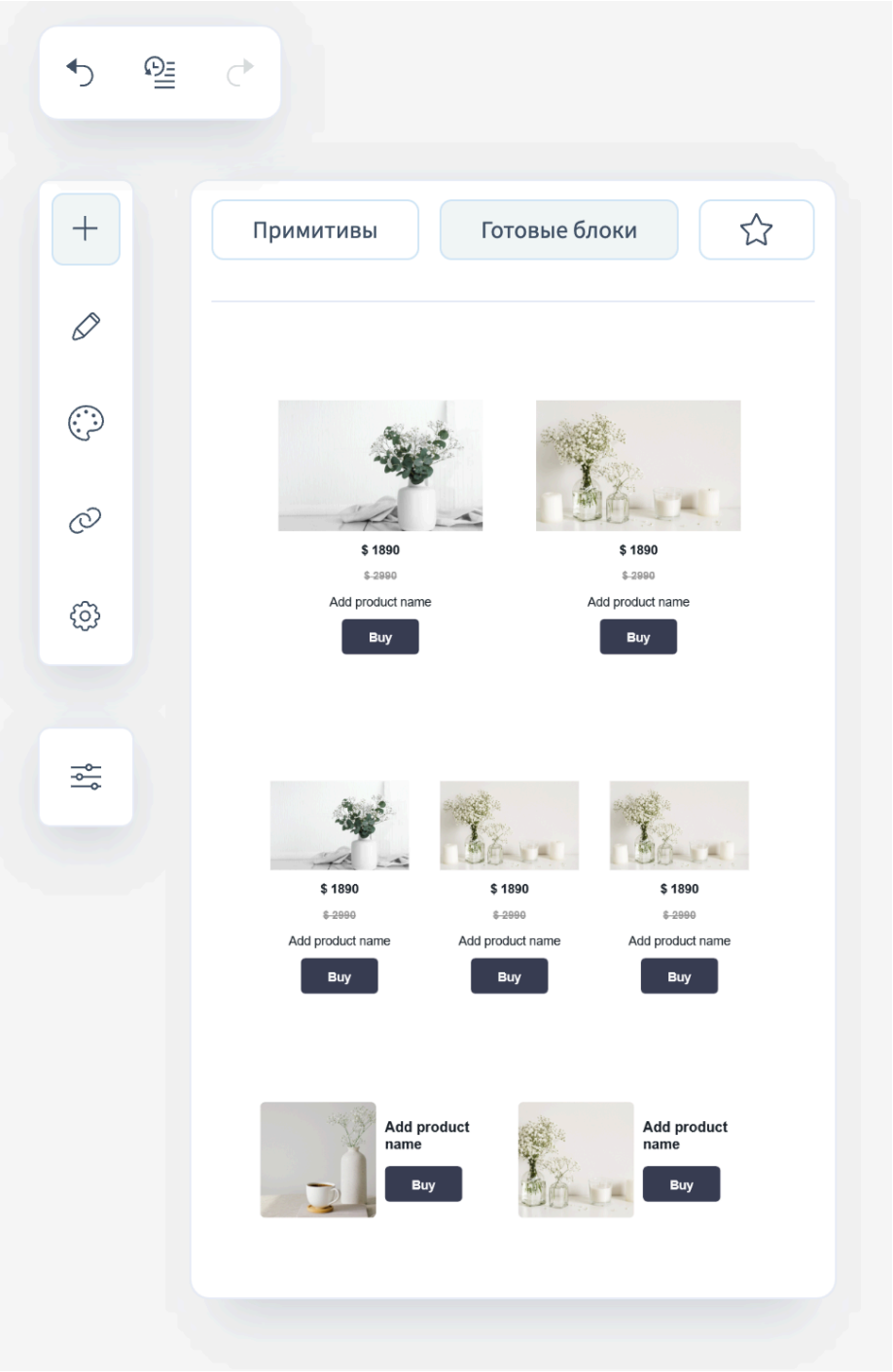

Товары
Дизайн
Данные
Источник
Маркет. Продукты
Способ выборки
Фильтры
ID товара
Сортировка
По ID
Производитель
manufacturer_name
Категории
categories
Цена
price
+ от
-
+ до
₽
Дополнительные фильтры
Только с картинками
pictures
Только с доставкой
delivery
Задана старая цена
old_price
Только в наличии
available
Добавить фильтр
Полоса

- **ID товара** — режим ручного выбора конкретных товаров по их идентификаторам. Подходит, если нужно отобразить фиксированные позиции, не зависящие от обновлений ассортимента.



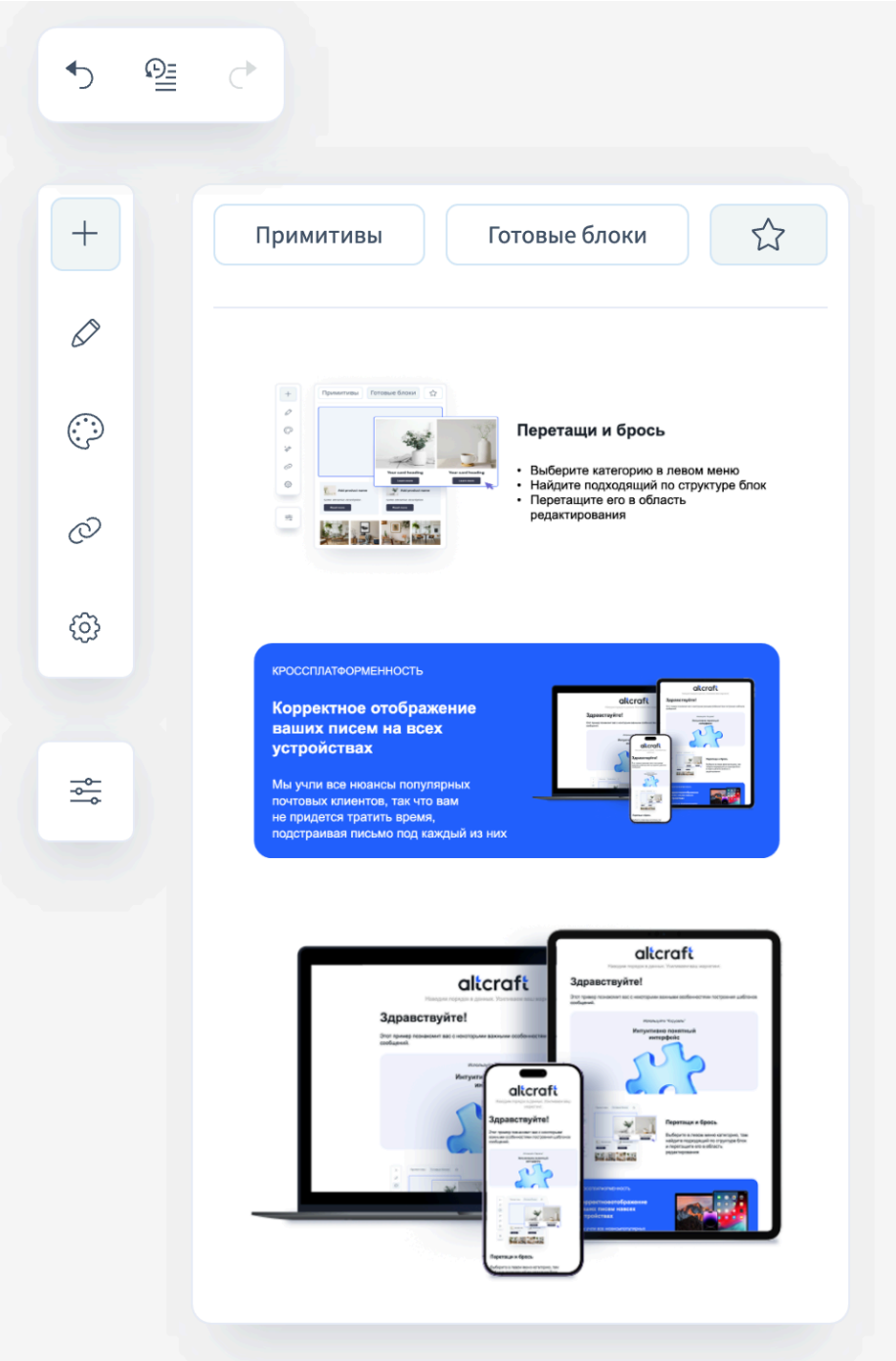
Готовые блоки

Набор заготовленных блоков, подобранных под самые разные цели. С их помощью вы сможете за одну минуту собрать базовый вариант письма, которое подойдет именно вам.



Избранные блоки

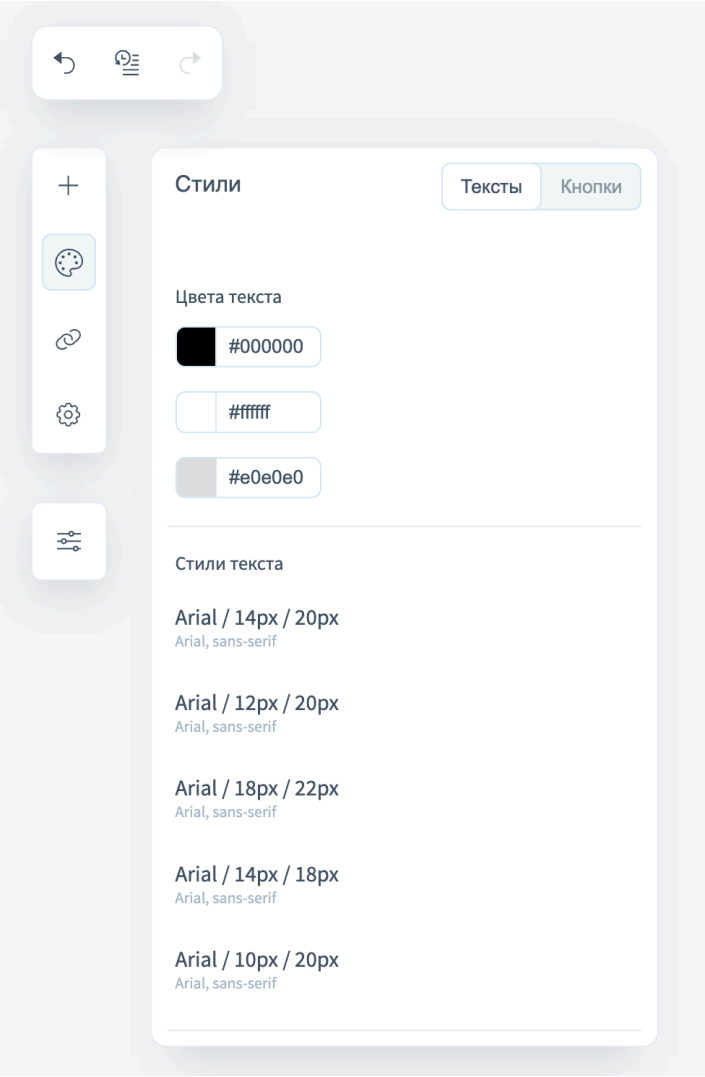
Если вы часто используете какой-либо компонент или хотите использовать одинаковые по стилю блоки в разных письмах, вы можете **сохранить его** для дальнейшего использования.



Чтобы сохранить блок, нажмите на значок “звёздочки”.

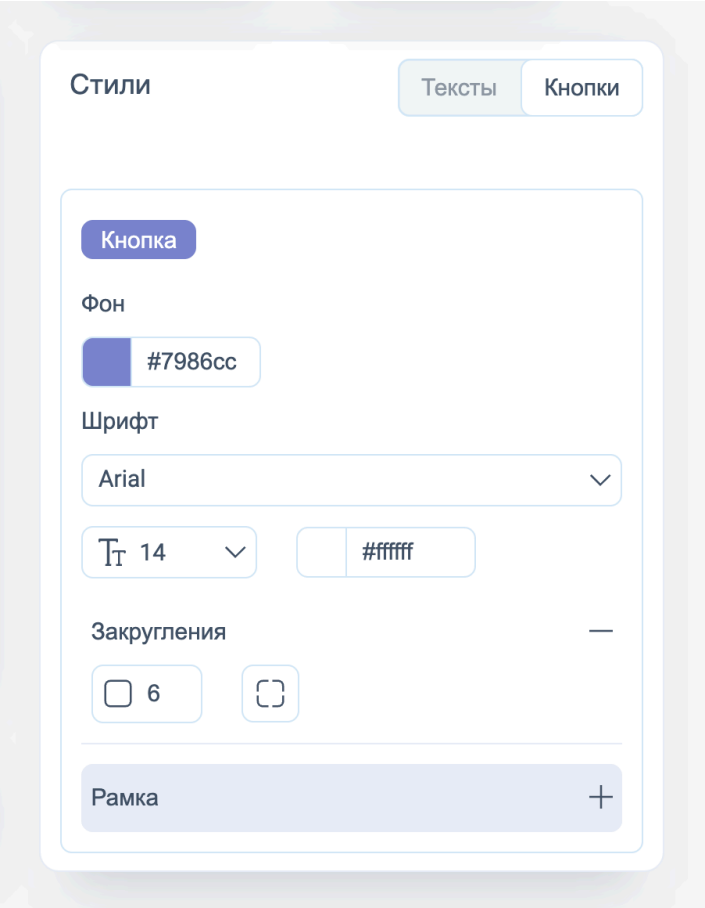
В дальнейшем он будет всегда доступен в редакторе на соответствующей вкладке (справа от раздела “Готовые блоки”).

Стили



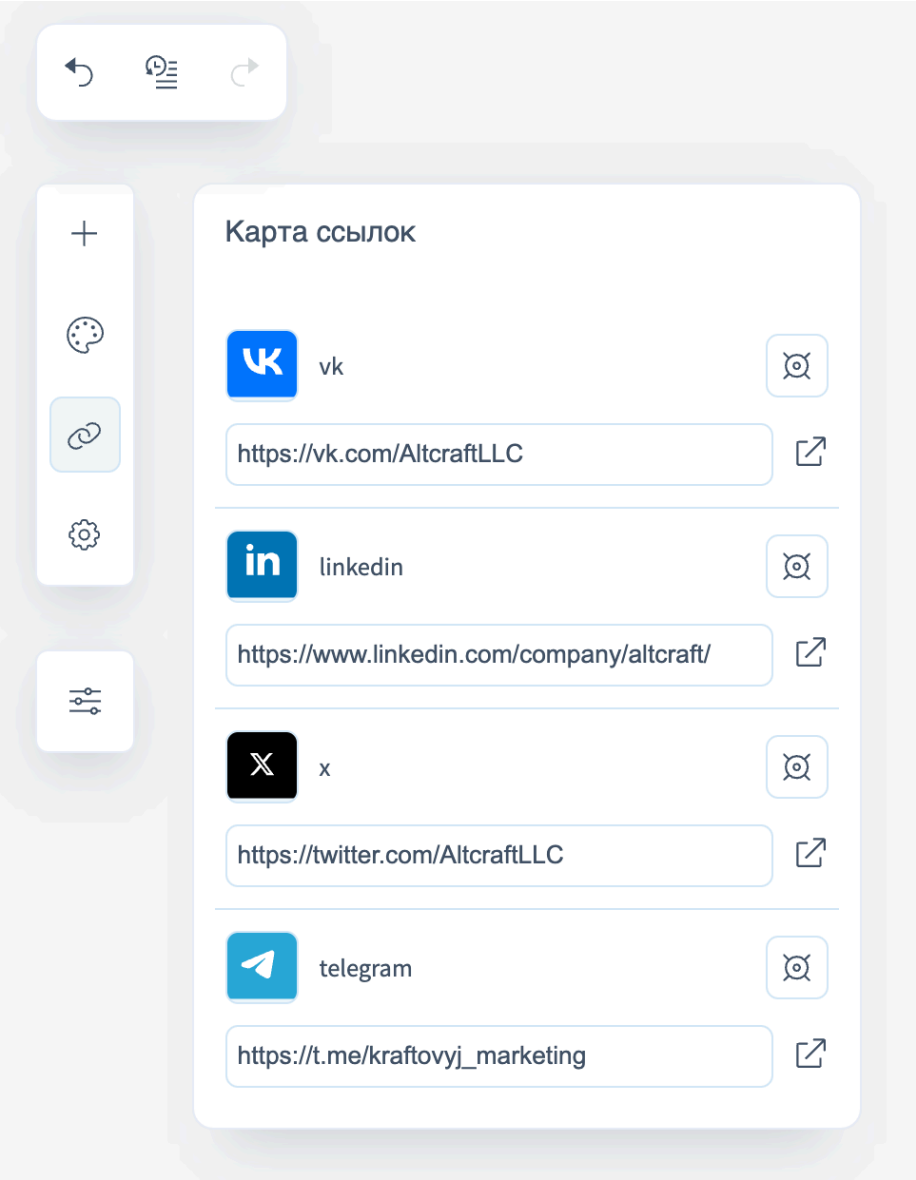
Сохранить единобразие визуальной составляющей и придерживаться одного дизайн-кода поможет раздел **“Стили”**. Здесь можно отслеживать используемые цвета и стили текста внутри **блоков** и **кнопок**.

В разделе “Кнопки” можно увидеть список всех кнопок и их стиль, из этого же меню можно изменять шрифт и цвет текста, а также визуальные свойства кнопок:



Рядом с каждым стилем и цветом расположена кнопка, которая перенаправит вас к элементу письма, в котором используется выбранный стиль.

Карта ссылок

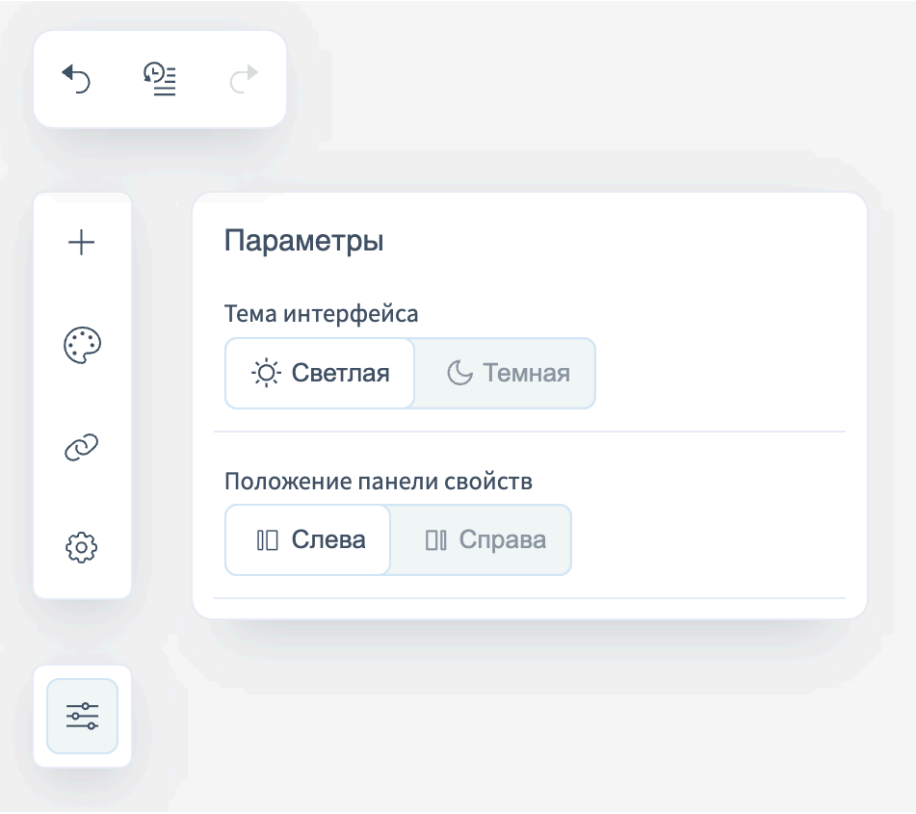


Этот раздел отображает все ссылки, используемые в текущем письме. Для каждой ссылки указано, в каком элементе она используется

Доступны следующие кнопки управления:

- **Переход по ссылке** — позволяет открыть ссылку в новом окне браузера и проверить, куда она реально ведёт.
- **Поиск элемента** — при нажатии подсвечивает элемент письма, в котором используется данная ссылка.
- **Удалить** — удаляет ссылку из списка и из соответствующего элемента на холсте.

Параметры

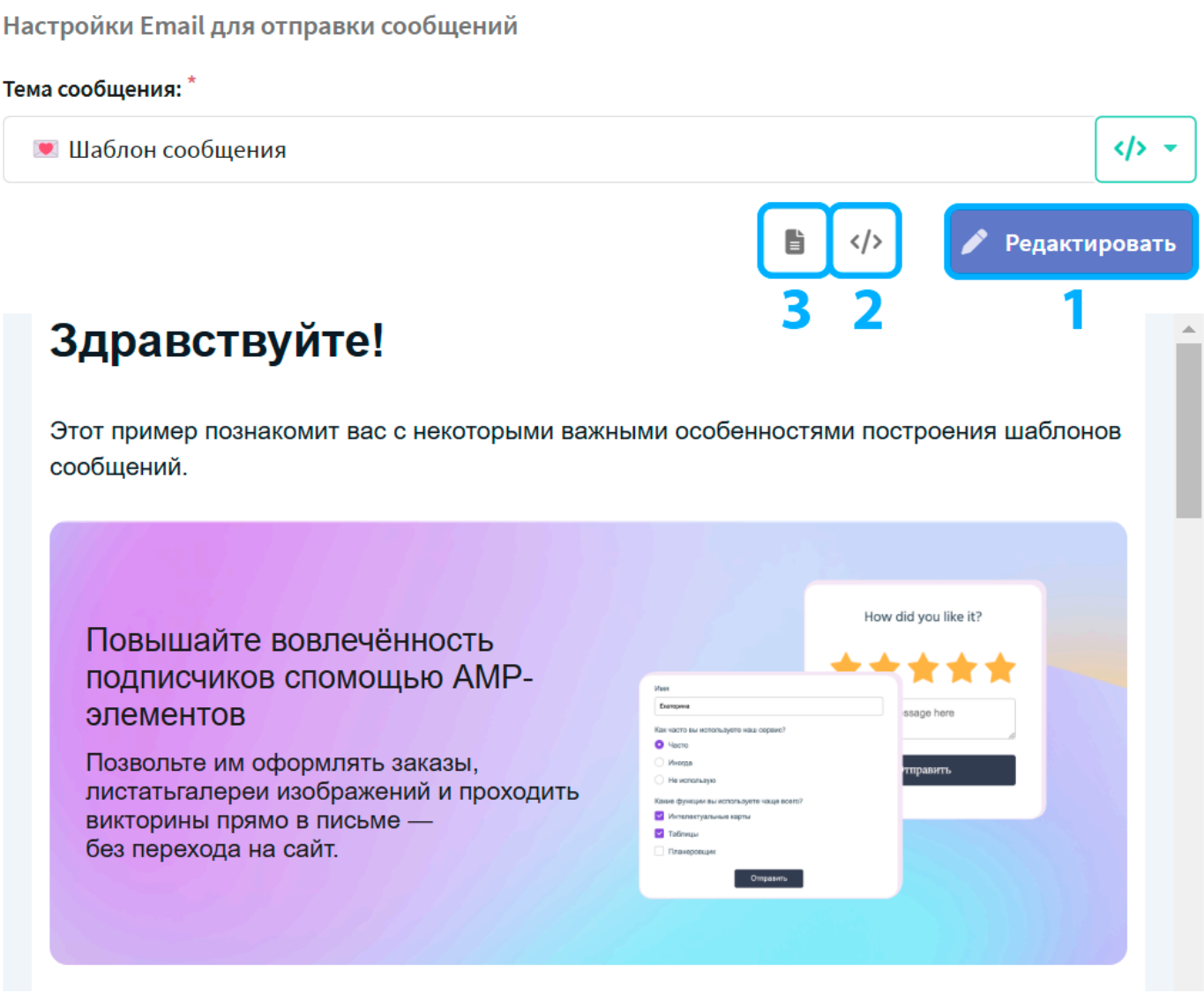


В этом разделе вы можете настроить внешний вид и расположение элементов редактора под себя.

- **Тема интерфейса** позволяет выбрать визуальное оформление редактора (светлая или тёмная тема).
- **Положение панели свойств** определяет, где будет располагаться панель свойств. Панель можно разместить **слева** или **справа** — в зависимости от ваших предпочтений.

Работа с шаблоном после создания

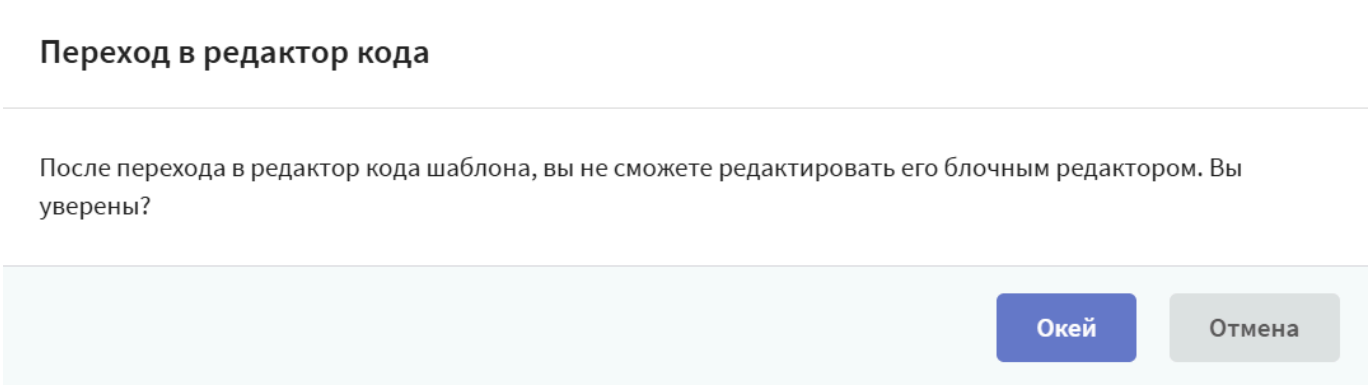
После создания шаблона и применения изменений, вам будет доступно несколько действий:



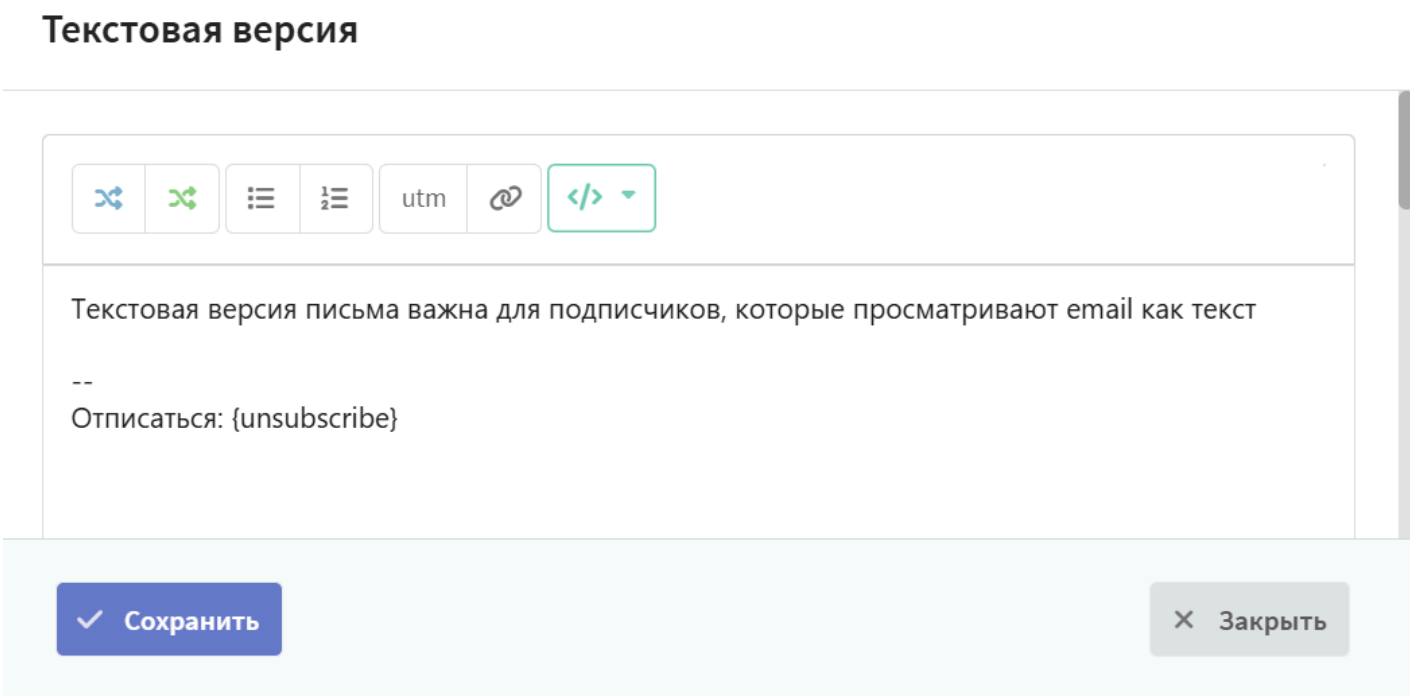
1. **Кнопка “Редактировать”** — позволяет вернуться в редактор и внести изменения в шаблон;
2. **Кнопка “Показать код шаблона”** — демонстрирует HTML-код созданного шаблона:

CAUTION

Вы можете перейти в редактор кода из появившегося окна, нажав кнопку “редактировать как код”, однако вы больше не сможете редактировать ваш шаблон в блочном редакторе.



3. Кнопка “**Редактировать текстовую версию письма**” — демонстрирует текстовую версию вашего шаблона, а также позволяет её редактировать.



Фрагменты шаблонов

	Фрагменты — это блоки сообщений, которые используются в нескольких сообщениях. Например, это может быть хедер или футер сообщения, блок со ссылками на социальные сети — любой контент, который вы будете отправлять в рамках нескольких рассылок.

TIP

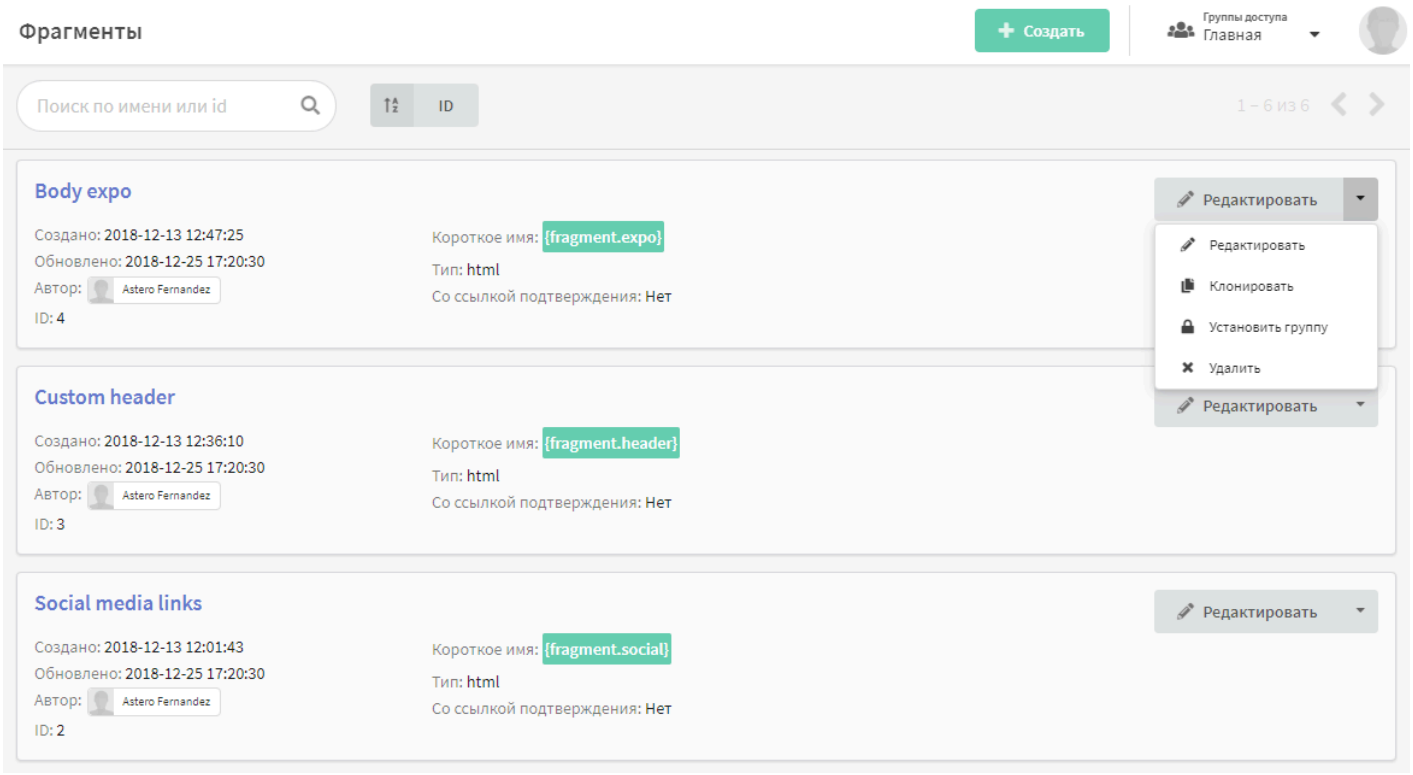
Фрагменты аналогичны **частям** шаблона, однако могут использоваться в нескольких шаблонах, в то время как части доступны только в шаблоне, в котором были созданы. Подробнее **части** описаны в настройках email-контента шаблона.

Например, во многих письмах могут быть ссылки на страницы в социальных сетях:

Со временем может накопиться множество таких сообщений, и добавление ссылки на ещё одну страницу во все эти сообщения займёт огромное количество времени, что-то будет забыто, где-то будут ошибки. Чтобы этого избежать, и созданы фрагменты.

Работа с фрагментами

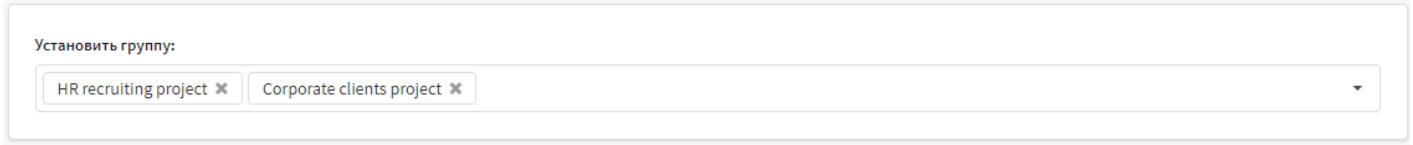
Чтобы начать работу с фрагментами, в главном меню выберите **Шаблоны —> Фрагменты**. Откроется список существующих фрагментов. С помощью контекстного меню вы можете редактировать, клонировать или удалить фрагмент, а также установить группу доступа к фрагменту. Если вам нужен новый фрагмент — нажмите **+ Создать** в верхней части страницы:



Основные настройки

В основных настройках фрагмента введите **имя** для отображения в списке фрагментов, а также **короткое имя**. Короткое имя используется при подстановке фрагмента в шаблон, вручную в виде `{fragment.<короткое имя>}` или с помощью меню переменных (`</>`). Также выберите **тип** фрагмента: HTML или текст (plain text):

В нижней части страницы находится блок, в котором вы можете установить группу доступа к фрагменту:



HTML фрагменты

HTML фрагменты подойдут для email сообщений, а также для других каналов, поддерживающих HTML. В редакторе поддерживаются HTLM и CSS, а также переменные персонализации. Для облегчения форматирования и работы с объектами доступны следующие функции:

Форматирование:

— **жирный текст**

— *курсив*

— ~~зачёркнутый текст~~

— маркированный список

— нумерованный список

— вставка таблицы

Служебные функции:

— добавление **пикселей** для отслеживания **открытия** и **прочтения** письма. Если вы не вставите пиксели в процессе редактирования сообщения, система предупредит вас об этом при сохранении.

— добавление **внешнего HTML-контента**. Вы также можете указать адрес источника переменной вручную `{[http://example.com]}`.

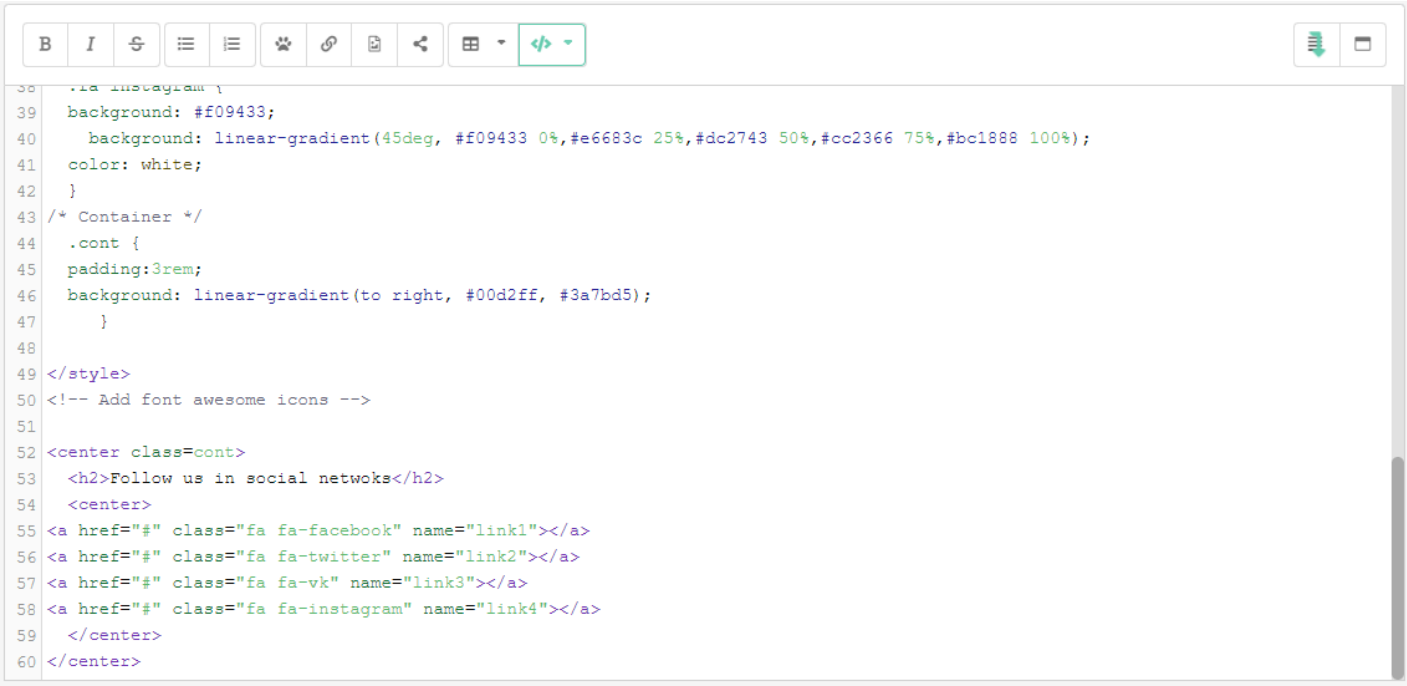
— добавление или редактирование **ссылок.**

— настройка **UTM меток.**

— добавление или редактирование **изображений.**

— добавление кнопок **Поделиться в социальных сетях.**

— меню **переменных и функций редактора.** С помощью этой кнопки вы можете добавить в сообщение данные профилей клиентов, промокоды, циклы и логические выражения, а также динамический контент и системные ссылки.



```
  .fa-instagram {
  background: #f09433;
    background: linear-gradient(45deg, #f09433 0%,#e6683c 25%,#dc2743 50%,#cc2366 75%,#bc1888 100%);
  color: white;
  }
/* Container */
  .cont {
  padding:3rem;
  background: linear-gradient(to right, #00d2ff, #3a7bd5);
     }

</style>
<!-- Add font awesome icons -->

<center class=cont>
  <h2>Follow us in social netwoks</h2>
  <center>
<a href="#" class="fa fa-facebook" name="link1"></a>
<a href="#" class="fa fa-twitter" name="link2"></a>
<a href="#" class="fa fa-vk" name="link3"></a>
<a href="#" class="fa fa-instagram" name="link4"></a>
  </center>
</center>
```

Текстовые фрагменты

Текстовые фрагменты могут использоваться во всех типах контента. В редакторе доступны следующие функции:

— маркированный список

— нумерованный список

— добавление или редактирование **ссылок.**

— меню **переменных и функций редактора**. С помощью этой кнопки вы можете добавить в сообщение данные профилей клиентов, промокоды, циклы и логические выражения, а также динамический контент и системные ссылки.

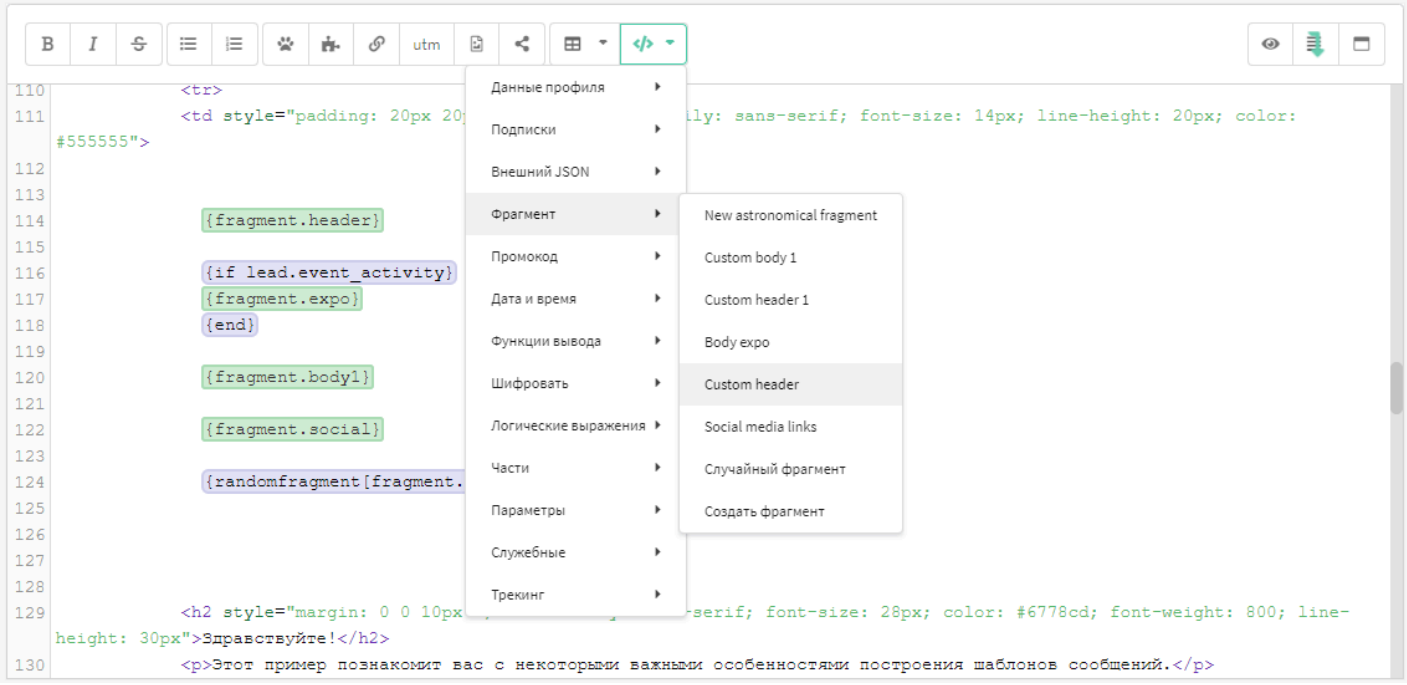
Использование фрагментов в шаблоне

Фрагменты добавляются в шаблон с помощью меню переменных редактора (`</>`). Или вручную с помощью `{fragment.<короткое имя>}`. Фрагменты могут отображаться в зависимости от определённых условий и данных подписчика.

TIP

Если вы хотите отображать в случайном порядке один из нескольких фрагментов, выберите вариант **Случайный фрагмент** и перечислите, какие фрагменты использовать:

```
{randomfragment[fragment.header_1 fragment.header_2 fragment.header_3]}
```



Изображения в сообщениях

Описание

Галерея позволяет хранить и организовывать изображения для использования в различных маркетинговых шаблонах:

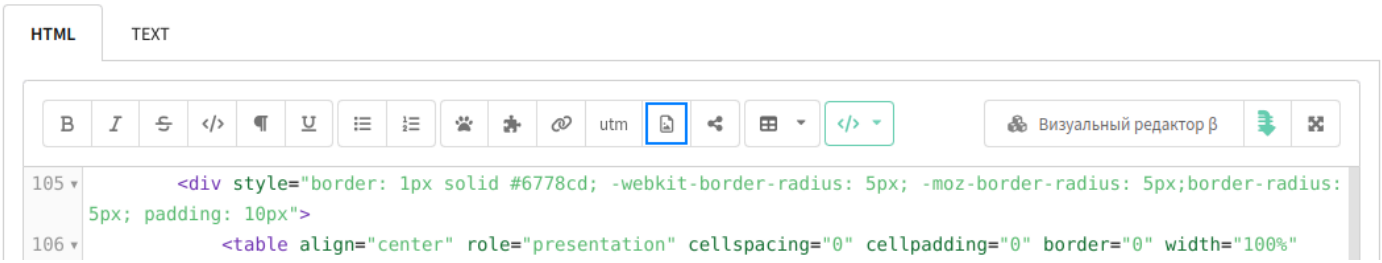
- HTML и AMP версии Email-шаблонов;
- Push-шаблоны (баннеры и иконки).

Изображения загружаются в сообщения и попапы с сервера платформы — через трекинг-домен.

Работа в галерее

Чтобы открыть галерею изображений, перейдите в редактор шаблона и нажмите на иконку

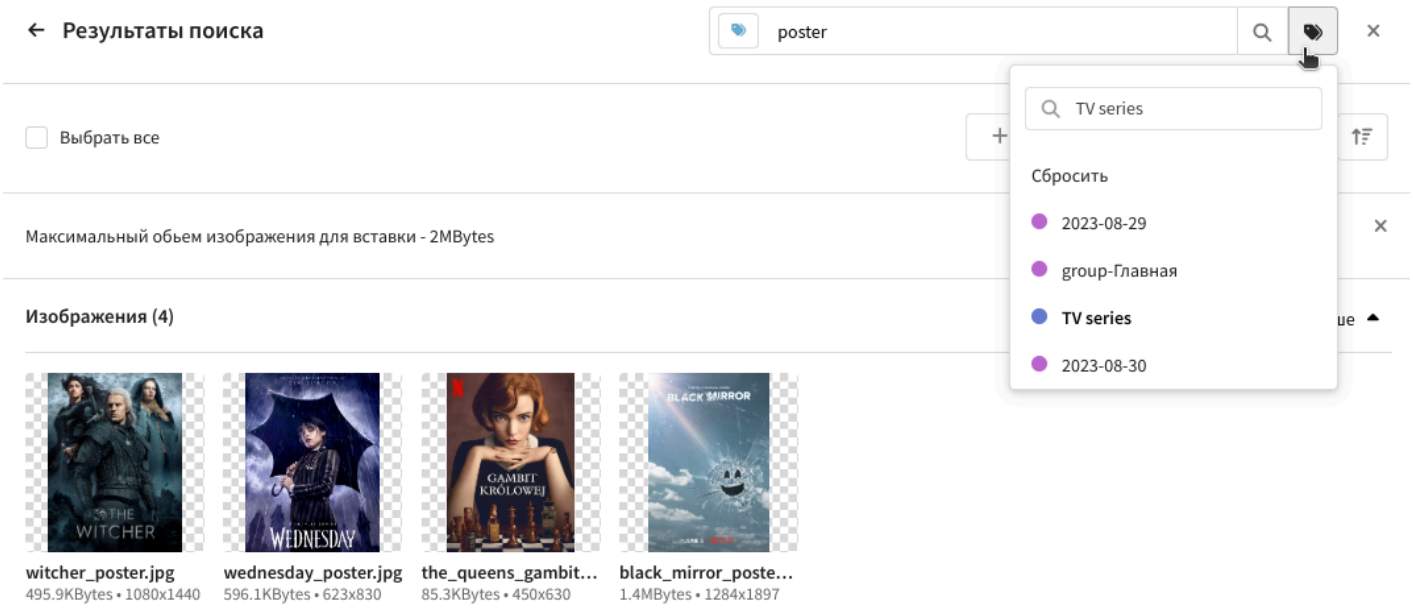
. Например, в **редакторе Email сообщения** перейти в галерею можно с помощью кнопки в панели инструментов:



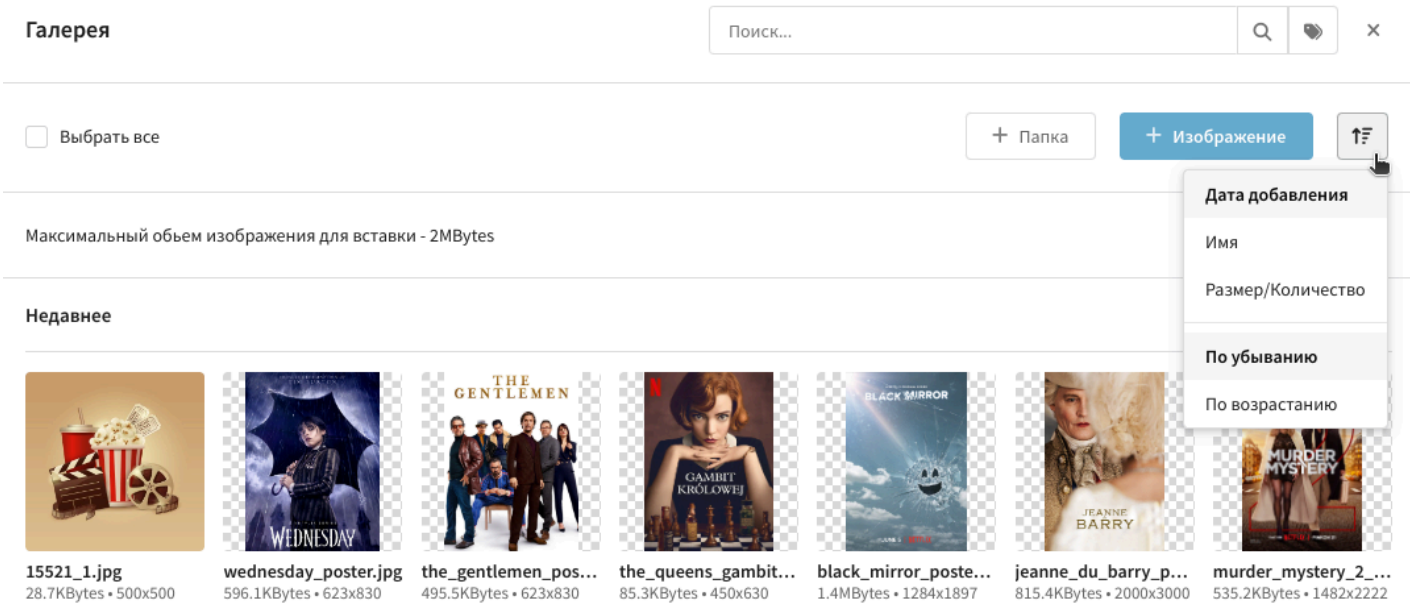
Галерея представляет собой модальное окно, визуально разделённое на 3 блока:

- Недавнее (все изображения, добавленные за последний месяц);
- Папки;
- Изображения.

Доступен **поиск по** названию файла/папки или по прикрепленным тегам. Чтобы выйти из режима поиска, нажмите ← **Результаты поиска**.



Также можно **отсортировать объекты** в галерее по дате добавления, по имени или по размеру:

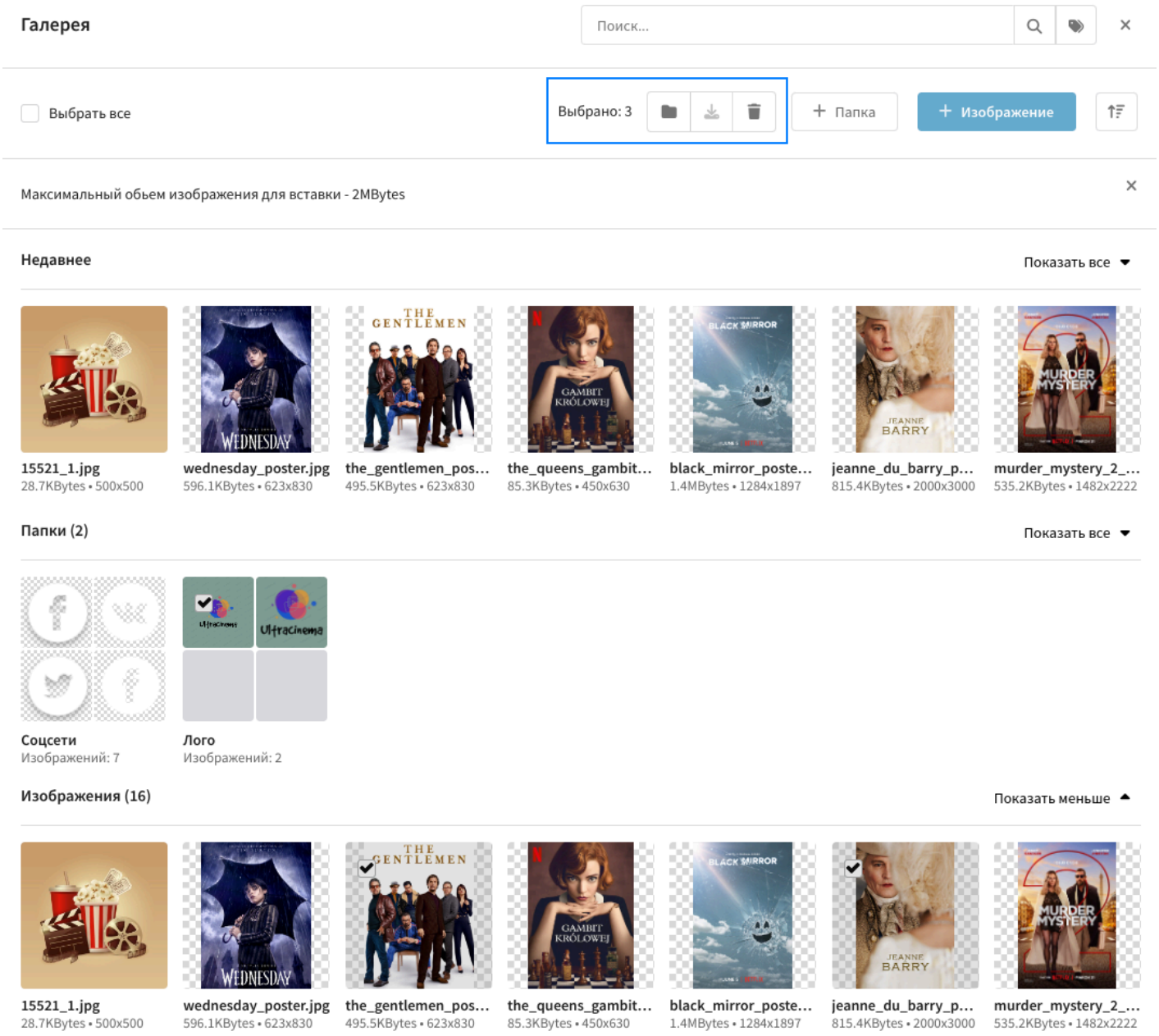


Вы можете выбрать один, несколько или сразу все объекты в галерее и применить одно из следующих действий:

— переместить объект в другую папку.

— скачать объект (только для изображений).

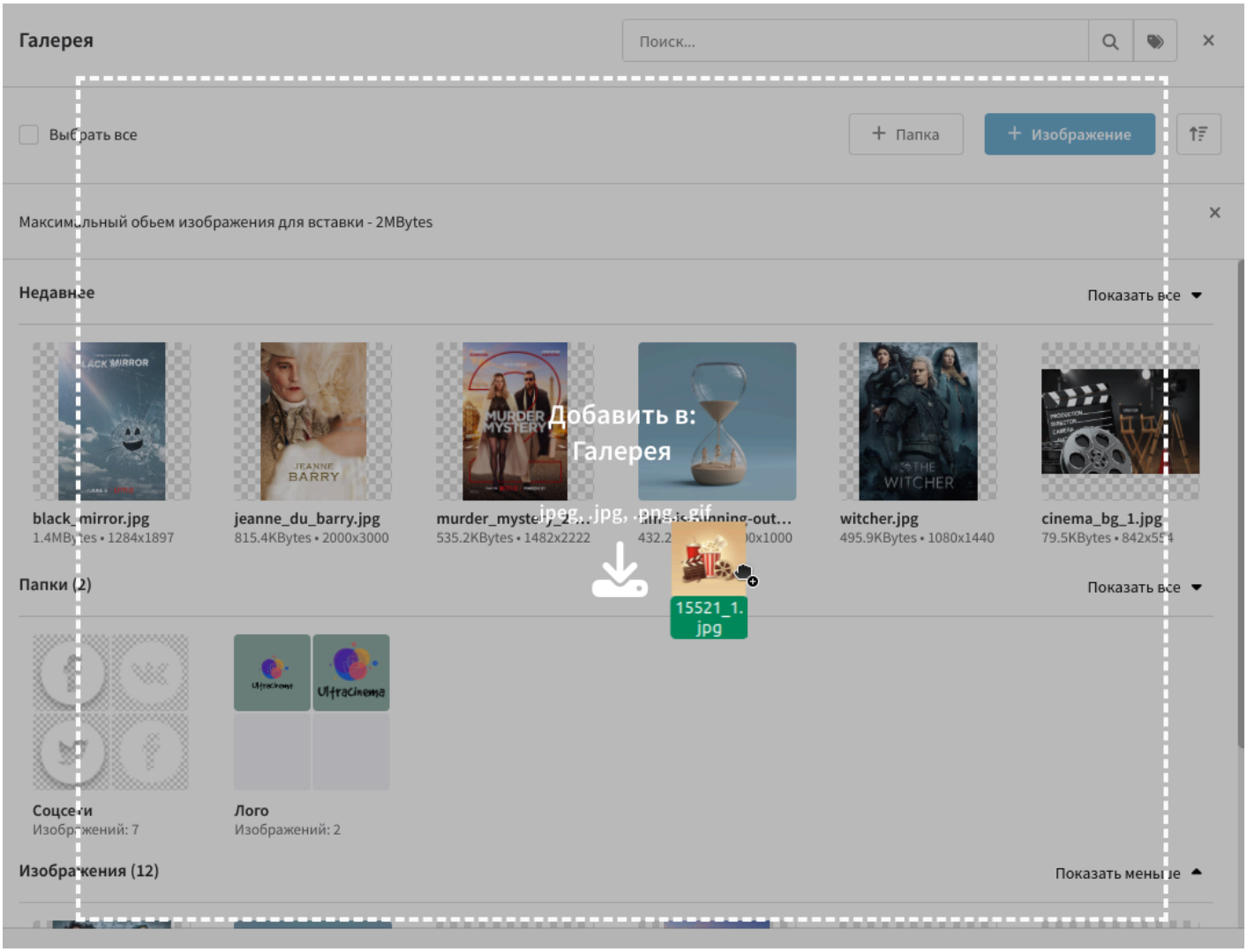
— удалить объект.



Работа с изображениями

Загрузка изображения

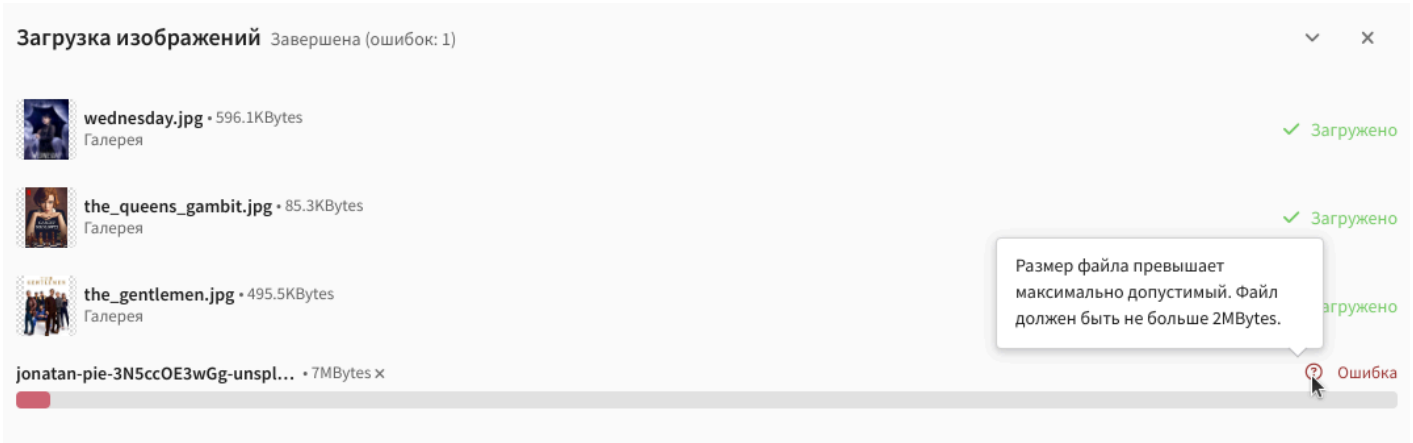
Чтобы добавить изображение в галерею, нажмите **“+ Изображение”** и выберите файл на своём устройстве. Второй способ — перетащить изображение в область галереи:



INFO

Можно загружать сразу несколько изображений. Размер одного изображения не должен превышать 2 МБ. Доступные форматы: jpeg, jpg, png, gif.

Статус загрузки отображается в нижнем блоке. Вы можете прервать загрузку, нажав на крестик. Если при добавлении изображения возникла ошибка, наведите курсор на знак вопроса, чтобы узнать причину:



Если вы закроете галерею, не дождавшись завершения импорта всех файлов, процесс импорта прервётся, однако все файлы, которые к текущему моменту были успешно загружены, останутся в галерее.

Действия с изображениями

Для работы с загруженными изображениями наведите курсор на изображение и нажмите

. Откроется список доступных действий:

— вставить в шаблон ссылку на изображение. Удобно, когда вам не нужно менять форматирование, а изменить атрибут src тега img.

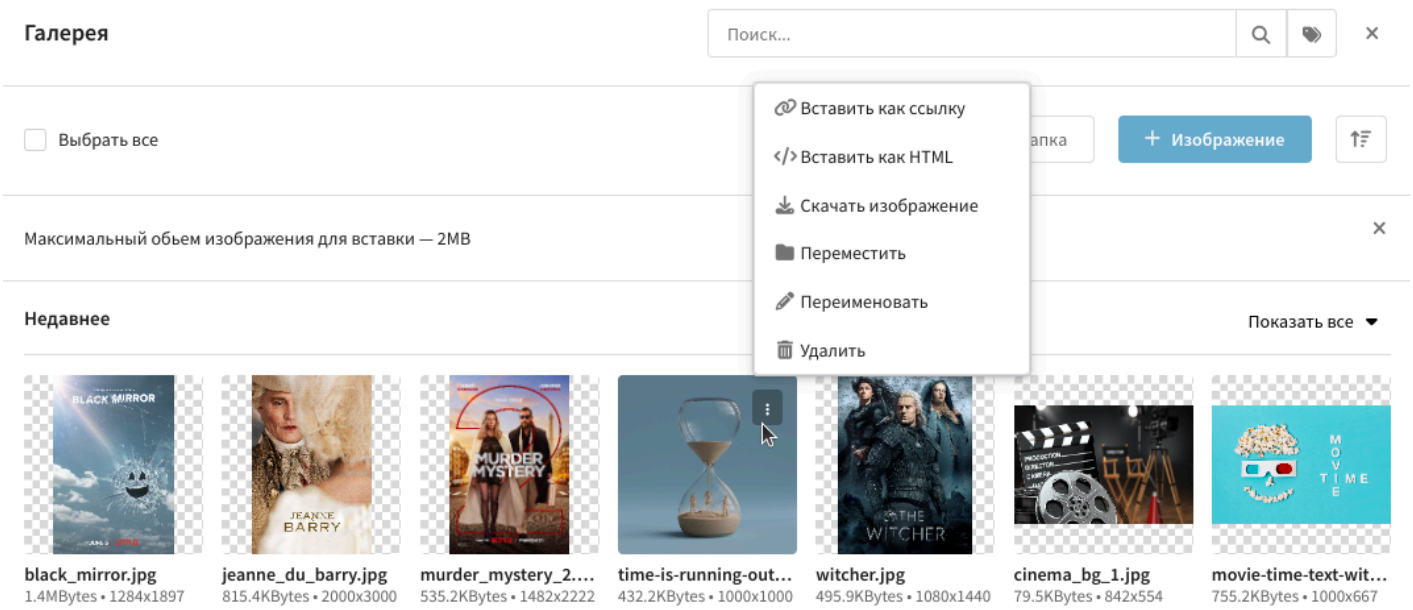
— вставить изображение в шаблон в формате HTML (новый тег img).

— скачать изображение на своё устройство. Изображения выгружаются в zip-архив.

— переместить изображение в другую папку.

— переименовать изображение. При изменении имени ссылка на изображение не меняется.

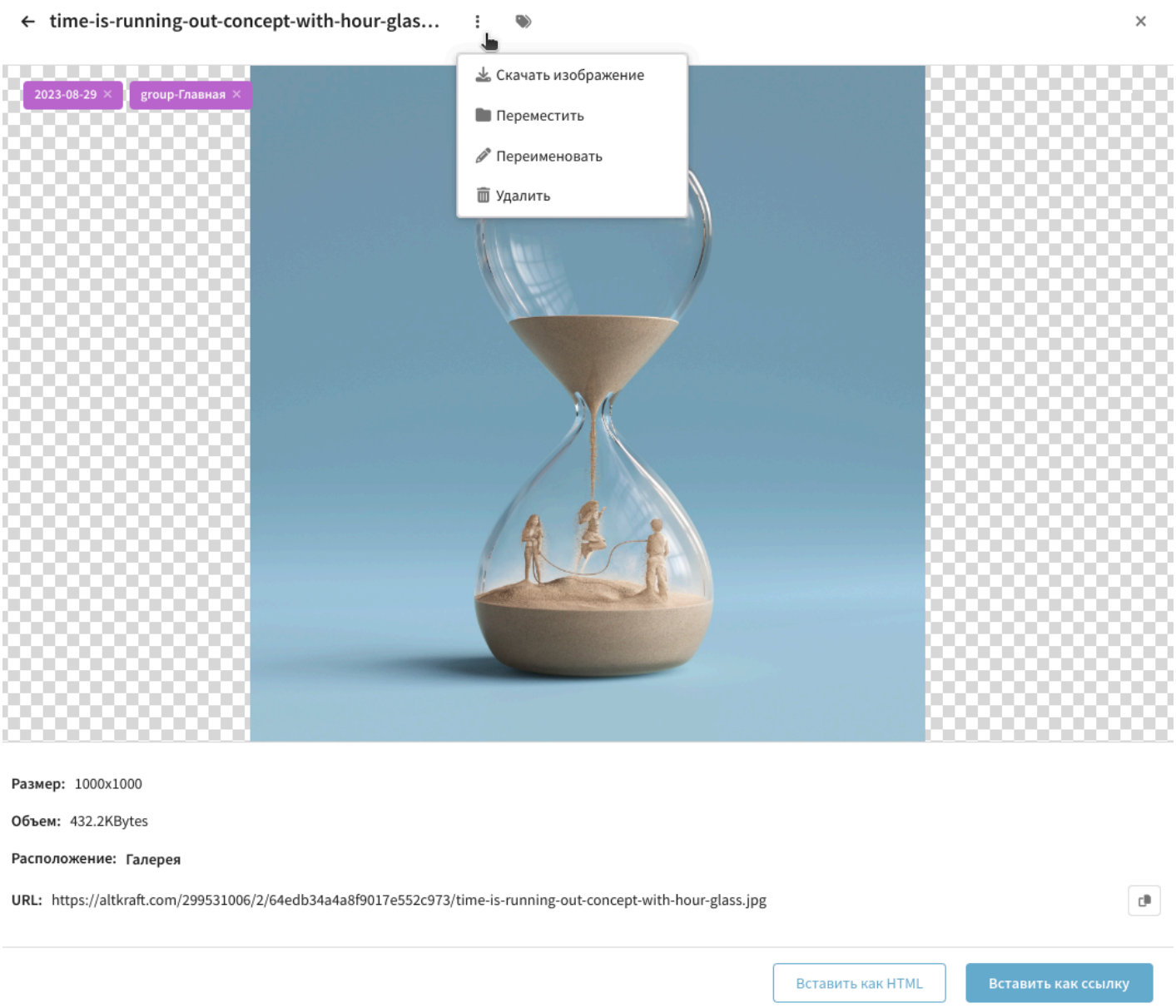
— удалить изображение из галереи.



Режим просмотра изображения

Если вы хотите получить больше информации по изображению, нажмите на него. Вы перейдете в режим просмотра изображения. Здесь доступны все действия, описанные выше. Чтобы открыть список действий, нажмите

:



Ниже есть возможность скопировать ссылку на изображение

. Такую ссылку можно использовать, например, при настройке иконки или баннера в push-шаблоне.

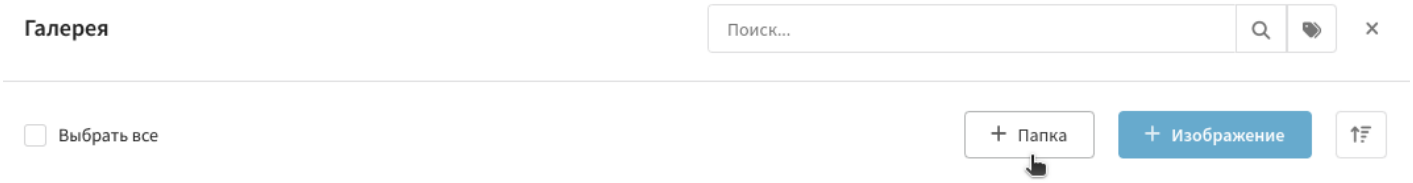
Кроме того, в этом режиме можно просмотреть теги, прикреплённые к изображению, а также добавить новые или убрать существующие. Некоторые теги создаются автоматически, например:

- `2023-08-29` (тег даты добавления изображения в галерею);
- `group-Главная` (название группы, в которой работал пользователь в момент добавления изображения).

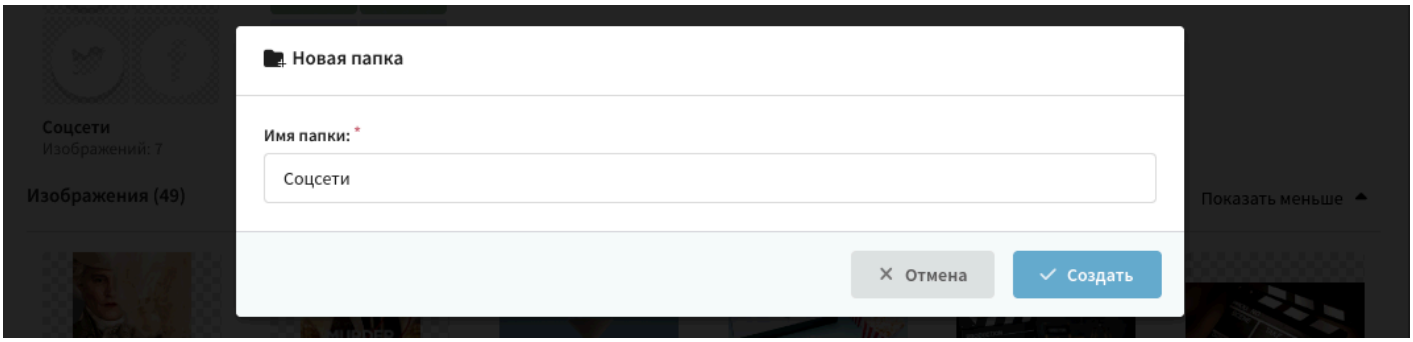
Чтобы выйти из режима просмотра изображения, нажмите ← **<имя_изображения>**.

Работа с папками

Для более удобной организации изображений в галерее используйте папки. Чтобы создать новую папку, нажмите **“+ Папка”:**



Введите название папки. В названии можно использовать любые символы, но не более 244.



Для работы с папкой наведите на нее курсор и нажмите

— откроется список доступных действий:

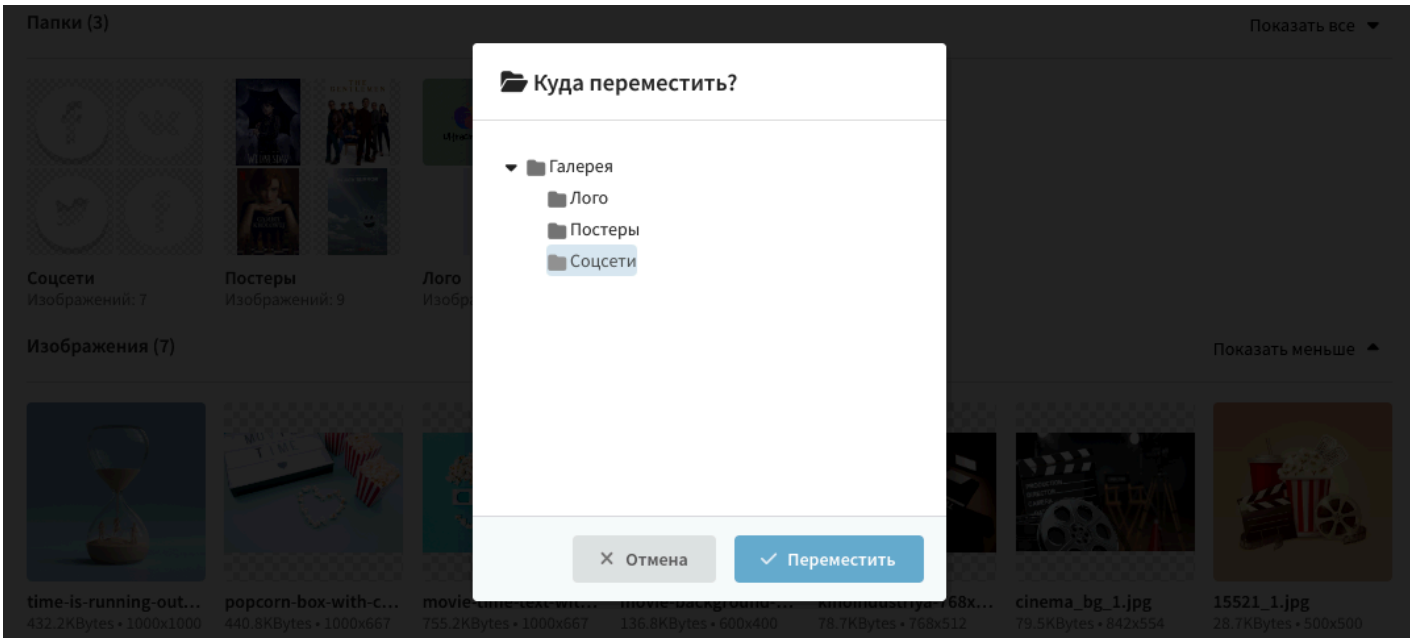
— переместить папку в другую папку. Нельзя перемещать родительскую папку в дочернюю.

— переименовать папку. Имя папки должно быть уникальным.

— удалить папку. Все объекты в папке также будут удалены из галереи.

Если вы хотите загрузить изображения в определенную папку, перейдите в нее и нажмите **“+ Изображения”** или перетащите файлы в область галереи.

Чтобы переместить уже добавленные изображения в папку, выделите изображения, в списке действий нажмите **“Переместить”** и выберите нужную папку:



Также доступно создание вложенных папок. Для этого перейдите в папку-родитель и нажмите **“+ Папка”.** Уровень вложенности не ограничен.

Чтобы вернуться в предыдущую папку, нажмите ← **<имя_папки>.**

Загрузка изображений из внешних источников

Зачастую в шаблоне используются ссылки на изображения из внешних источников. Например, если вы сверстали шаблон в стороннем сервисе, а затем перенесли в платформу Altcraft. В случае сбоев на внешнем сервисе такие изображения могут стать недоступны. Чтобы избежать этого, рекомендуется загрузить их в галерею платформы.

Нажмите кнопку

в панели инструментов редактора (кнопка появляется только после добавления в атрибут `src` URL-адреса на внешнее изображение):

После загрузки на сервер платформы значение атрибута `src` для всех внешних изображений изменится, например, `<img src="https://img.com/glass-jar.jpg">` заменится на `<img src="http://{trkdomain}/383848228/1/64eefca84a8f9017e5/glass-jar.jpg">`. Теперь эти изображения будут загружаться в сообщения через трекинг-домен.

Персонализация контента в сообщениях

Данные профилей клиентов могут использоваться для создания динамических и персонифицированных сообщений, не требующих постоянных изменений. Шаблон будет подставлять значения профиля из Базы данных, что позволяет персонализировать массовые рассылки.

TIP

Например, сообщение с ежедневным гороскопом:

```
Hello, Eddie!

Daily horoscope for Scorpio.

This is the greatest day to consider your options, study economic trends, and analyze what you want from investing.
```

Можно записать так, используя данные профиля и ссылку на сервер, генерирующий гороскопы. В системе ALTCRAFT это называется динамический HTML-контент.

```
Hello, {lead._fname}!

Daily horoscope for {lead.Sign}.

{[http://127.0.0.1/horoscopes?sign={lead.Sign}]}
```

Таким образом вы получаете одно письмо на все 12 знаков зодиака, которое не требует никаких затрат времени после создания.

Подстановка данных клиента в шаблон

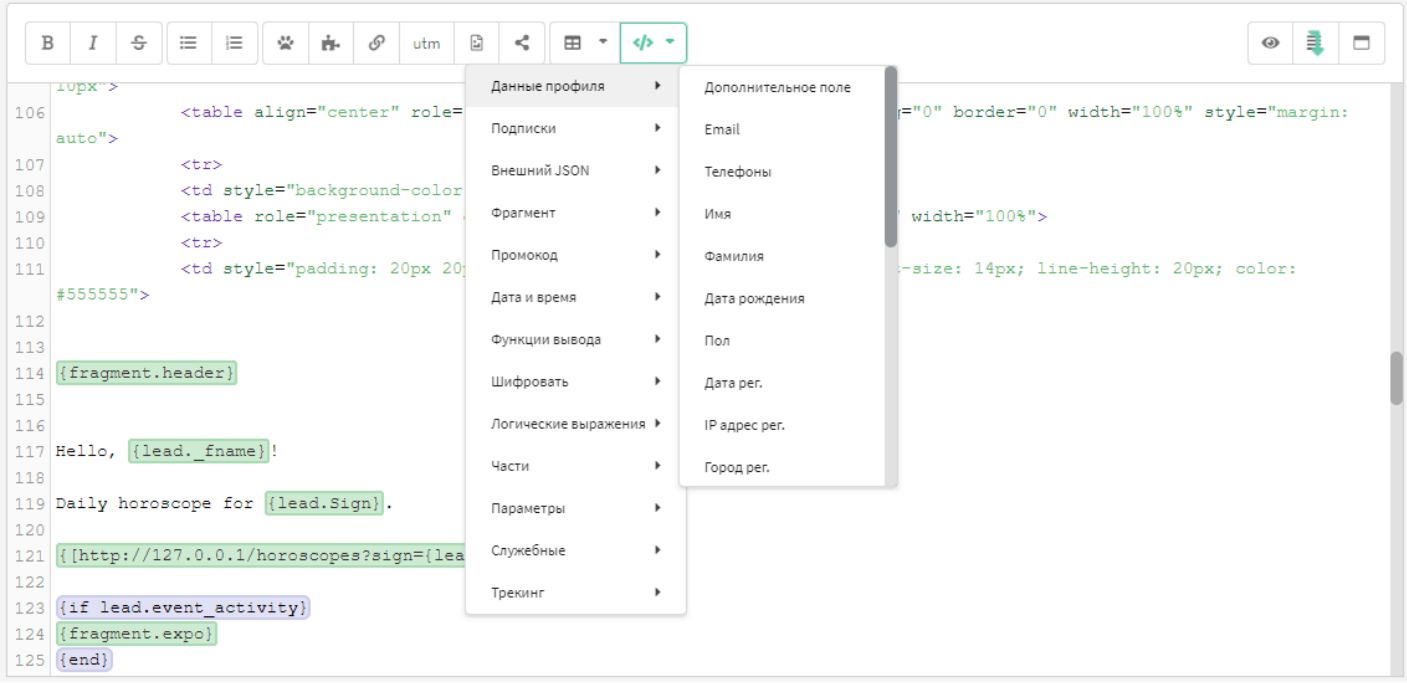
Чтобы добавить данные из стандартных полей, откройте **меню переменных** (`</>`) в редакторе контента сообщения, и выберите в списке один из следующих разделов:

- **Данные профиля** — данные клиента из стандартных полей Базы профилей.
- **Подписки** — данные подписок профиля на различные ресурсы.
- **Промокод** — текст промокода определённой программы лояльности.
- **Параметры** — дополнительные данные, такие как названия Базы данных, Рассылки, Шаблона, Ресурса.

Чтобы добавить данные из **дополнительных (кастомных)** полей профиля клиента, напишите в коде письма `{lead.Custom_Field_Name}`, где `Custom_Field_Name` — это имя дополнительного поля, написанное в точности, как в базе данных профиля. В нашем примере выше это `{lead.Sign}`.

Также вы можете добавлять данные из дополнительного поля типа “*Объект*”. Вложенность может быть любой. Например, `{lead.Custombject.fieldname}`.

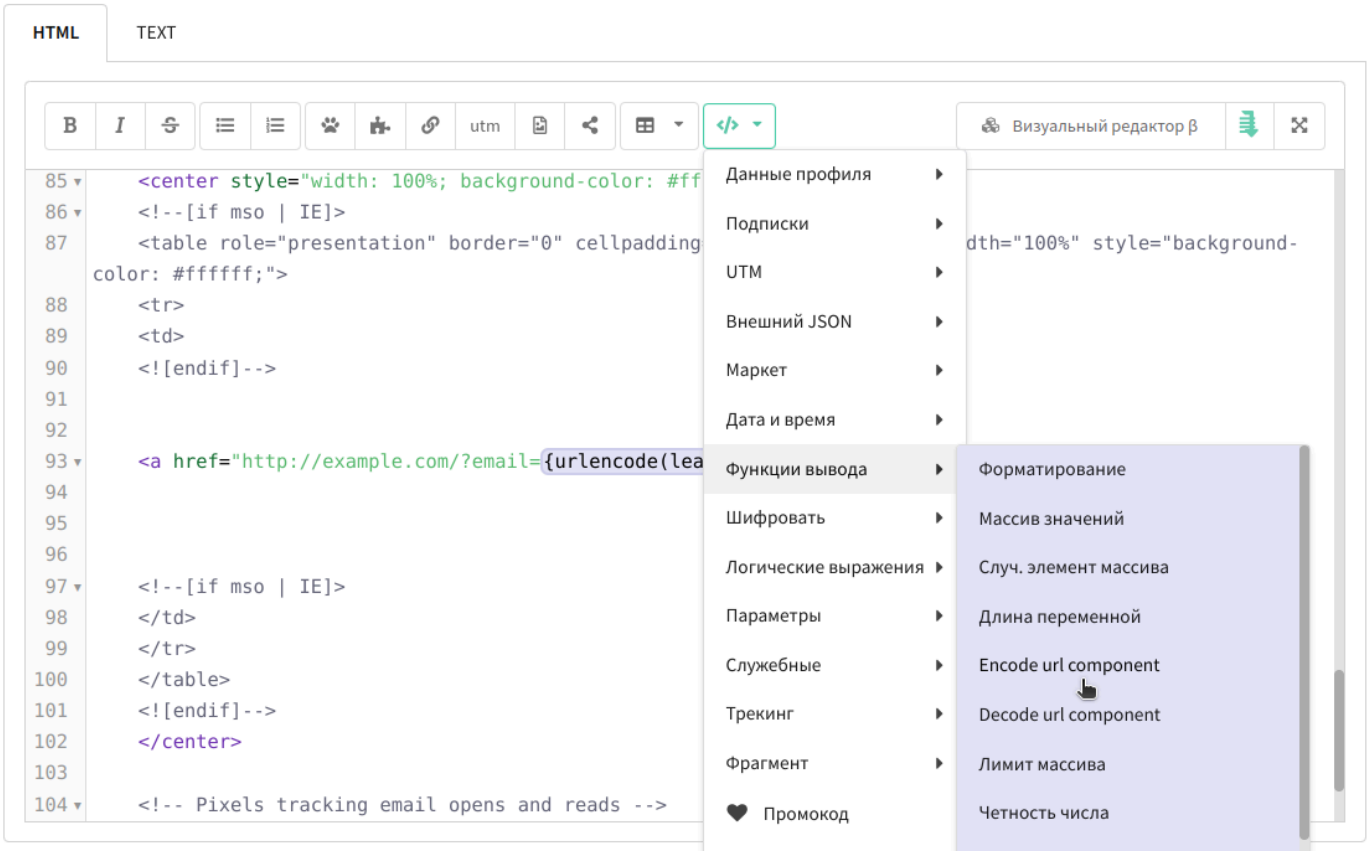
> **TIP**
>
> Список доступных **переменных** вы найдёте на [этой странице].



Использование переменных в URL параметрах

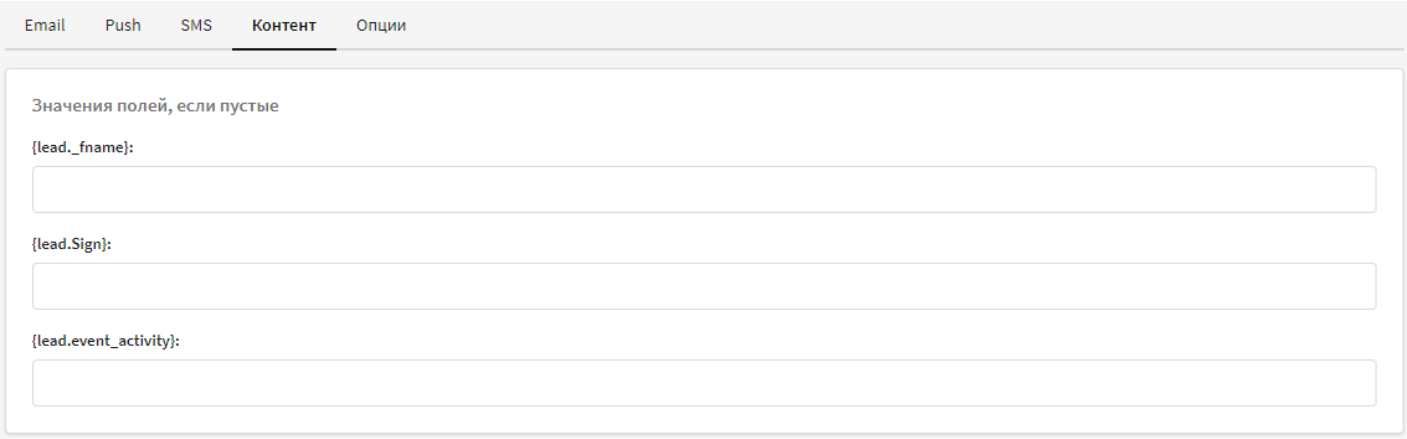
В URL адресе допустимы только латинские буквы, цифры и ограниченный набор знаков препинания. Однако, при использовании переменных, которые подтягивают в шаблон данные из полей профиля, в URL могут попасть недопустимые символы (например, пробелы или кириллица). В таком случае ссылка в шаблоне не будет работать.

Чтобы избежать подобной ситуации, **необходимо кодировать параметры в URL адресе**. Это можно сделать с помощью функции `{urlencode}`. Например, если вы подставляете переменную `lead.email` в ссылку, используйте следующую конструкцию: `https://example.com/?email={urlencode(lead.email)}`.



Значения по умолчанию

Во вкладке **Контент** шаблона можно установить значения по умолчанию для использованных в этом письме полей — на случай, если нужных данных у подписчика не окажется.



Функциональное изменение данных клиента

Если вы хотите, чтобы в зависимости от перехода по той или иной ссылке в профиль клиента записывалось какое-либо значение, добавьте к любой ссылке в сообщении атрибут `update-имя_поля="значение"`.

При переходе по такой ссылке значение поля профиля клиента в Базе данных поменяется на заданное. Например:

```
<a href="http://www.example.org" name="link" update-clicked_link="true">Обновить данные поля clicked_link!</a>
```

Также клиента можно прямо из письма подписать на другой ресурс по тому же каналу и контактным данным. Укажите следующий атрибут, подставив идентификатор нужного ресурса:

```
<a href="http://www.example.org" subscribe-to="1">Подписаться на ресурс #1</a>
```

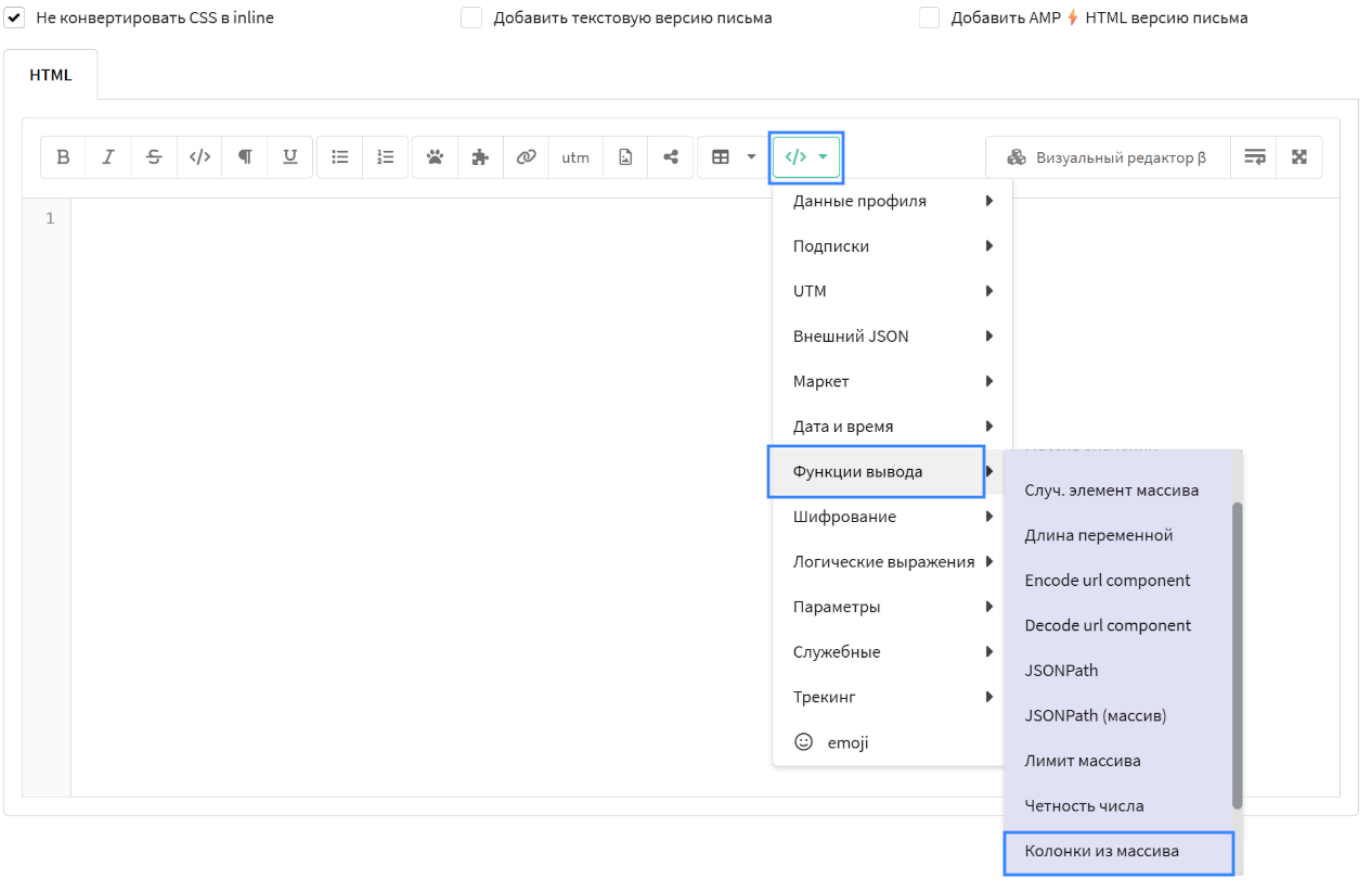
Формирование таблиц на основе элементов массива

Формирование таблиц на основе элементов массива — это одна из возможностей платформы, которой можно воспользоваться при передаче динамического контента в сообщении. Мы можем взять массив линейных объектов из базы данных, а затем распределить элементы этого массива в таблице так, как того требует ситуация.

TIP

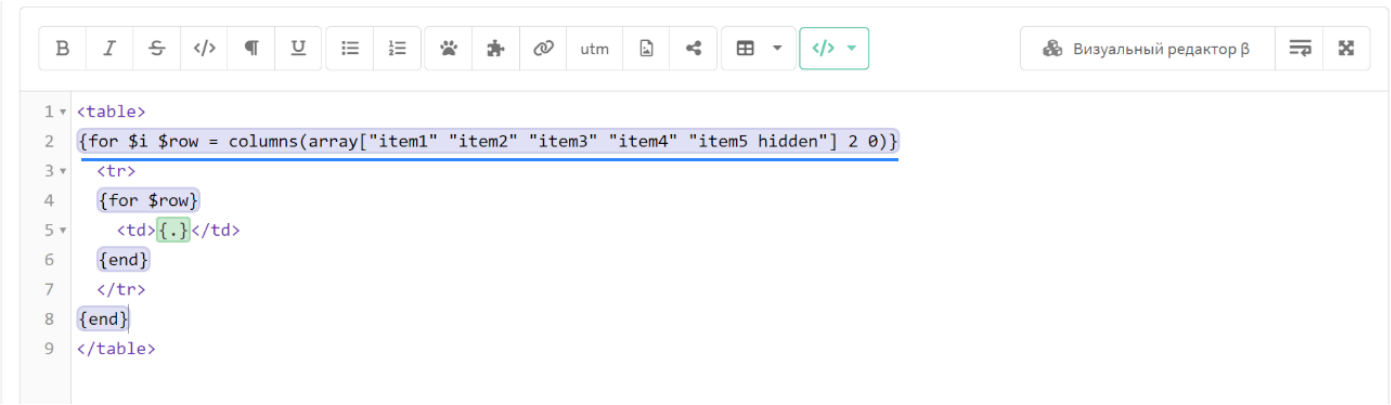
Данная возможность доступна при передаче контента любым способом: API-контент, JSON-контент, контент из SQL базы данных.

Чтобы создать таблицу на основе элементов массива, в окне **HTML версия письма** кликните: **Вставить переменную —> Функции вывода —> Колонки из массива.**



Шаблон кода появится в окне редактора. Обратите внимание на следующую строчку:

```
{for $i $row = columns(array["item1" "item2" "item3" "item4" "item5 hidden"] 2 0)}
```



Здесь `array` — это массив, из которого мы берем контент для таблицы. Каждый элемент массива (`"item1"`, `"item2"`, `"item3"` и т. д.) содержит контент для определенной ячейки таблицы.

Цифра “**2**” — это количество колонок, которые будут в таблице. Цифра “**0**” — это количество элементов, которые не будут добавлены в таблицу (отсчет начинается с первого элемента массива).

Таким образом, в сгенерированной таблице все элементы массива будут распределены в 2 колонки и ни один из них не будет пропущен.

Пример реализации

Давайте рассмотрим один из случаев применения данной опции.

Предположим, на сайте авиакомпании динамически меняется контент для каждого пользователя в зависимости от истории его полетов. Каждый пользователь видит на своем экране 5 предложений о рейсах, которые ему, возможно, будут интересны. Эти предложения показаны в интерфейсе как 5 блоков (`item`). В каждом из них есть информация о городах, датах и ценах за билет. Мы хотим не только отобразить эти блоки, но и вставить между ними отдельный блок, где предложим пользователю самому выбрать город полета (`offer`):

<table>
<tr><td></td><td>
<table>
<tr><td colspan="2">Item1</td></tr>
<tr><td colspan="2">Offer</td></tr>
<tr><td>Item2</td><td>Item3</td></tr>
<tr><td>Item4</td><td>Item5</td></tr>
</table>
</td></tr>
</table>

Наши действия:

1. Создаем массив, куда складываем информацию о всех пяти предложениях (`"item1"`, `"item2"`, `"item3"`, `"item4"`, `"item5"`).
2. Отдельно добавляем первый элемент массива (`item1`).
3. Добавляем блок с предложением (`offer`).
4. Добавляем оставшиеся элементы массива, распределяя их на 2 колонки. Первый элемент не будет добавлен, потому что мы указали, что хотим пропустить один элемент.

```
<table>
  <tr>
    <td colspan="2">{for limit(array["item1"] 1)}{.}{else}{cancel}{end}</td>
  </tr>
  {if lead.offer}
  <tr>
    <td colspan="2">{lead.offer}</td>
  </tr>
  {end}
{for $i $row = columns(array["item1" "item2" "item3" "item4" "item5" "item6 hidden"] 2 1)}
  <tr>
  {for $row}
    <td>{.}</td>
  {end}
  </tr>
{end}
</table>
```

Формирование таблиц на основе элементов массива — гибкий механизм, который позволяет в любое время легко изменить отображение вашего сообщения для пользователя.

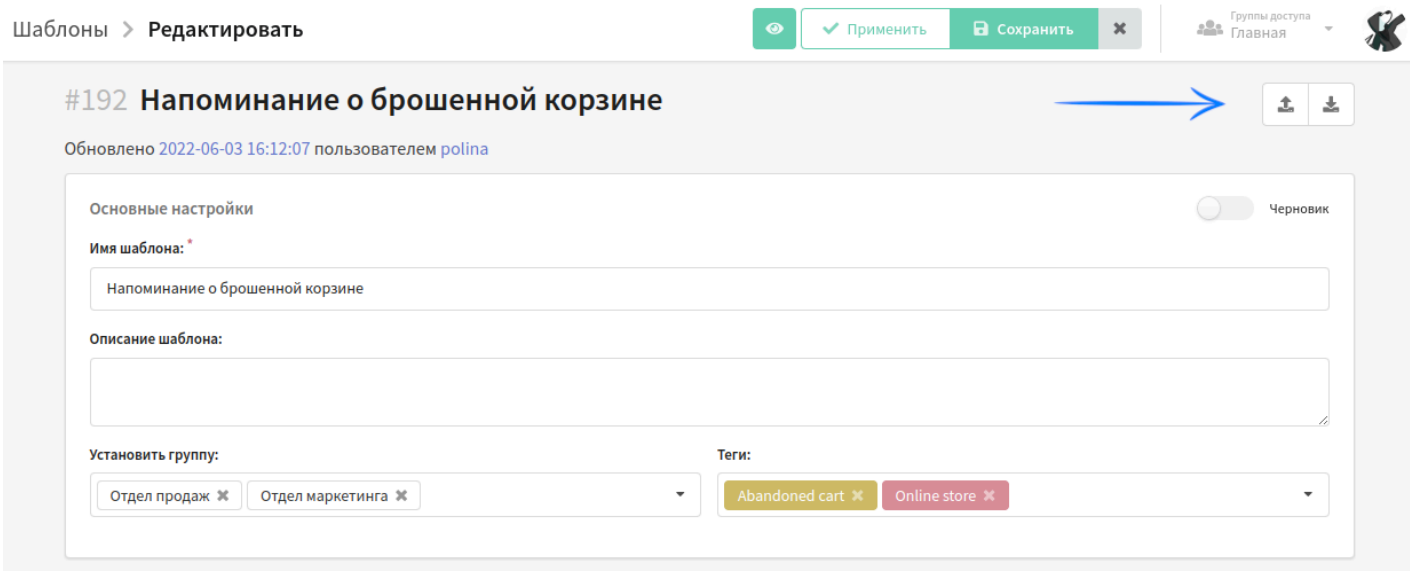
Импорт и экспорт шаблона сообщения

Данный функционал позволяет загрузить в платформу или выгрузить на своё устройство шаблон сообщения. Это может быть шаблон как для одного канала, так и для нескольких сразу. Для экспорта и импорта используется файл в формате `zip`.

Соответствующие кнопки доступны в окне редактирования шаблона в правом верхнем углу:

— импортировать шаблон

— экспортировать шаблон



Импорт шаблона

Чтобы импортировать шаблон в платформу, нажмите на иконку

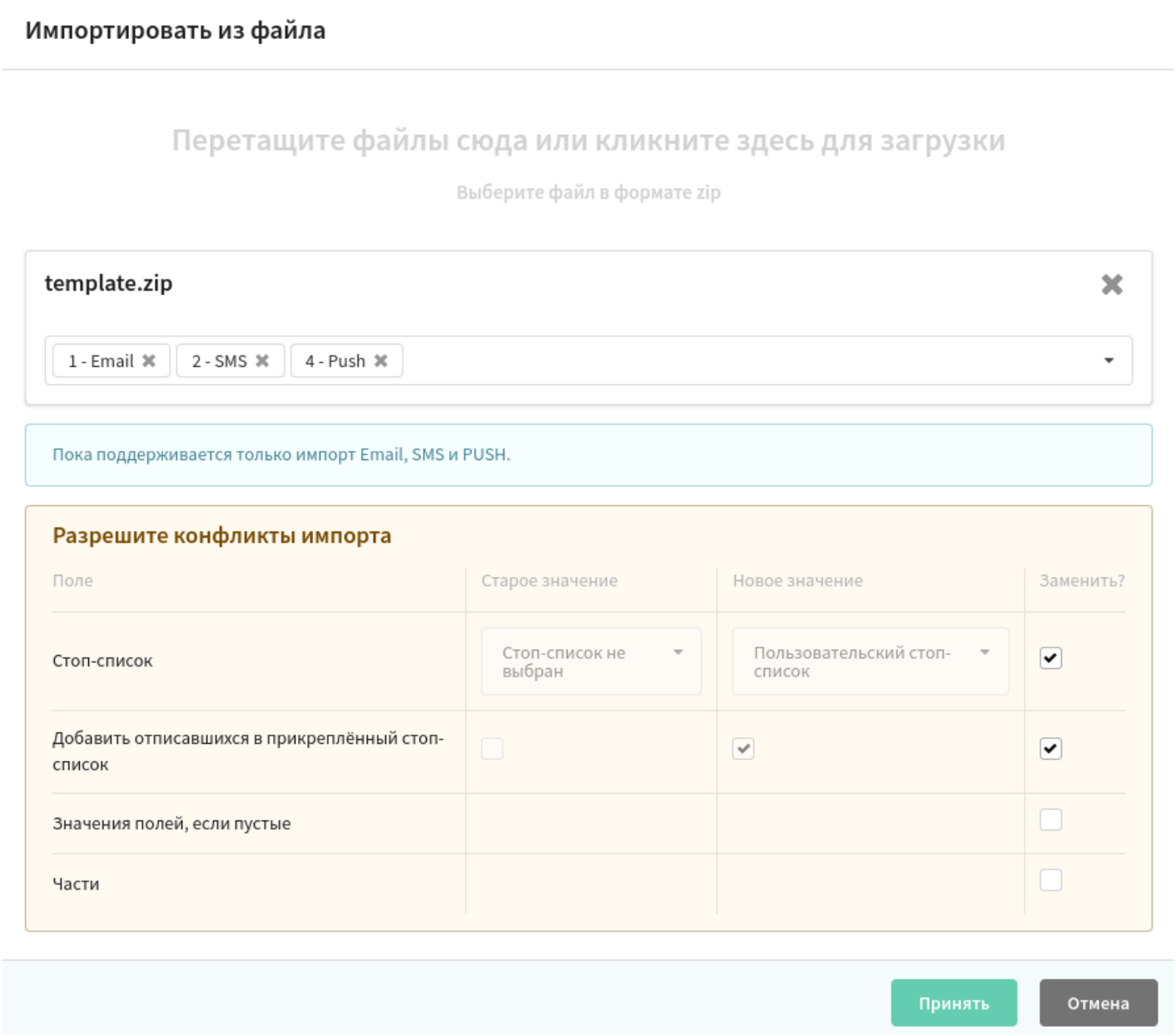
. Далее кликните на область загрузки и выберите файл в формате `zip` на своём устройстве.

> **TIP**
>
> О переносе шаблона из стороннего сервиса читайте в этой статье.

Настройте импорт:

1. Выберите **каналы**, для которых необходимо импортировать шаблон (Email, SMS, Push).
2. Разрешите **конфликты импорта**. Конфликты возникают, если импортируемые данные и данные, которые уже сохранены в платформе, отличаются. В таблице показаны старое и новое значение. Если вы хотите обновить данные, установите галочку в колонке “Заменить”.

Нажмите **Принять**, чтобы запустить процесс импорта.



Структура zip-файла

Для импорта используется файл в формате `zip`. Архив должен иметь определенную структуру:

```
<catalog_name>/
    <channel_name>/
        <fieldname>.<type>
        <attach>/
            <fileid>/
                <filename>
    images/
        <image_name>
    template.json
```

Описание структуры:

Элемент	Пример	Описание
<catalog_name>	template_newsletter	Корневая директория. Можно использовать произвольное название.
<channel_name>	email sms push	Директория содержит данные шаблона для определенного канала отправки. Название директории должно соответствовать каналу. Поддерживается импорт Email, SMS и Push шаблонов. Информацию о том, какие файлы можно импортировать для каждого канала, вы найдете здесь.
images	image_1.png	Директория для хранения изображений.
template.json	—	Файл содержит общие данные о шаблоне (ссылка на динамический JSON контент, дополнительные настройки, фрагменты и части). Более подробную информация вы найдете ниже.

Пример zip-файла

TIP

Начиная с версии 68, изображения могут храниться на одном уровне с HTML-файлом или в папке с одним из следующих названий: images, image, img, imgs, pictures, pics. Архив с таким email-шаблоном не должен содержать файлы text.txt, subject.txt, template.json, так как их наличие может вызывать ошибки при импорте.

Базовая структура для email-канала

```
template_123/
    email/
        html.html
        text.txt
        subject.txt
    images/
        image_1.jpeg
    template.json
```

Полная структура для всех каналов

```
template_123/
    email/
        html.html
        text.txt
        subject.txt
        amp.txt
        is_amp.txt
        is_not_inline.txt
        from_email.txt
        from_name.txt
        replyto_email.txt
        replyto_name.txt
        attach/
            023f-ad8c-4412-9bba-16ac/
                attach_file.zip
    sms/
        sms.txt
        add_sms_url_scheme.txt
    push/
        title.txt
        body.txt
        click_url.txt
        icon.txt
        is_rich_push.txt
    images/
        image_1.jpeg
        image_2.jpeg
    template.json
```

Пути к изображениям в HTML

Для корректного отображения картинок в импортируемом шаблоне используйте относительные пути, соответствующие структуре архива.

TIP

Убедитесь, что пути в атрибуте `src` соответствуют фактическому расположению файлов в архиве.

Рекомендуемая структура:

```
template_name
├── email
│   └── html.html
└── images
    └── banner.png
```

Пример корректного пути в HTML:

```
<img src="../images/banner.png" alt="Баннер">
```

INFO

Не создавайте папку `images` одновременно в корне и внутри папки канала — это может привести к конфликтам при импорте

Описание полей для каждого канала

Email

Файл	Пример	Описание
html.html	`<html>Hello, world! </html>`	HTML-версия письма
text.txt	Hello, world!	Текстовая версия письма
subject.txt	Лучшие новинки 2022	Тема письма
amp.txt	`<html>Hello, world! </html>`	AMP HTML версия письма
is_amp.txt	false	Есть ли в шаблоне amp-версия
is_not_inline.txt	true	Не конвертировать CSS в inline
from_email.txt	awesome@example.com	Email отправителя
from_name.txt	Awesome Books	Имя отправителя
replyto_email.txt	reply_awesome@example.com	Обратный Email
replyto_name.txt	Awesome Books	Обратный получатель
attach (директория)	023f-ad8c-4412-9bba-16ac/ attach_file.pdf	Директория для хранения прикрепленных файлов. Содержит дополнительные директории, в каждой из которых хранится отдельный файл. Дополнительным директориям можно давать произвольные имена.

SMS

Файл	Пример	Описание
sms.txt	Hello, world!	Текст сообщения
add_sms_url_scheme.txt	true	Добавлять “http://” в ссылки

Push

Файл	Пример	Описание
title.txt	Лучшие новинки 2022	Заголовок уведомления
body.txt	Hello, world!	Текст уведомления
click_url.txt	https://example.com/	Ссылка для перехода по клику
icon.txt	https://example.com/images/test.jpg	URL-адрес, по которому расположена иконка для уведомления
is_rich_push.txt	false	Использование rich-версии для push сообщения
<platform>.json	`<html>Hello, world! </html>`	Данные по шаблону для определенного браузера или ОС (при использовании rich push)

Файл template.json

С помощью файла `template.json` можно загрузить данные, которые отображаются в блоках “Опции” и “Контент”.

Пример template.json

```
{
  "json_url": "https://altkraft.com/example/RU_content.json",
  "suppress_id": 1,
  "is_unsub_to_supp": true,
  "field_overrides": {
    "lead": {
      "_fname": "клиент"
    }
  },
  "parts": []
}
```

Описание полей в template.json

Поле	Пример	Обязательный	Описание
json_url	“https://altkraft.com/example/content.json”	Нет	URL для динамического JSON-контента
suppress_id	1	Нет	Идентификатор стоп-списка
is_unsub_to_supp	true	Нет	Добавлять отписавшихся в прикрепленный стоп-список

Поле	Пример	Обязательный	Описание

field_overrides	{
	“lead”: {
	_fname": “клиент”
	}
	}
		Нет	Значения полей, если пустые

parts	[
	{
	“name”: “part_1”,
	“shortname”: “part_1”,
	“type”: “text”,
	“content”: “”
	},
	{
	“name”: “part_2”,
	“shortname”: “part_2”,
	“type”: “text”,
	“content”: “”
	}
	]
		Нет	Части шаблона

TIP

Вы можете экспортировать любой готовый шаблон из своего аккаунта, чтобы получить наглядный пример zip-архива для импорта.

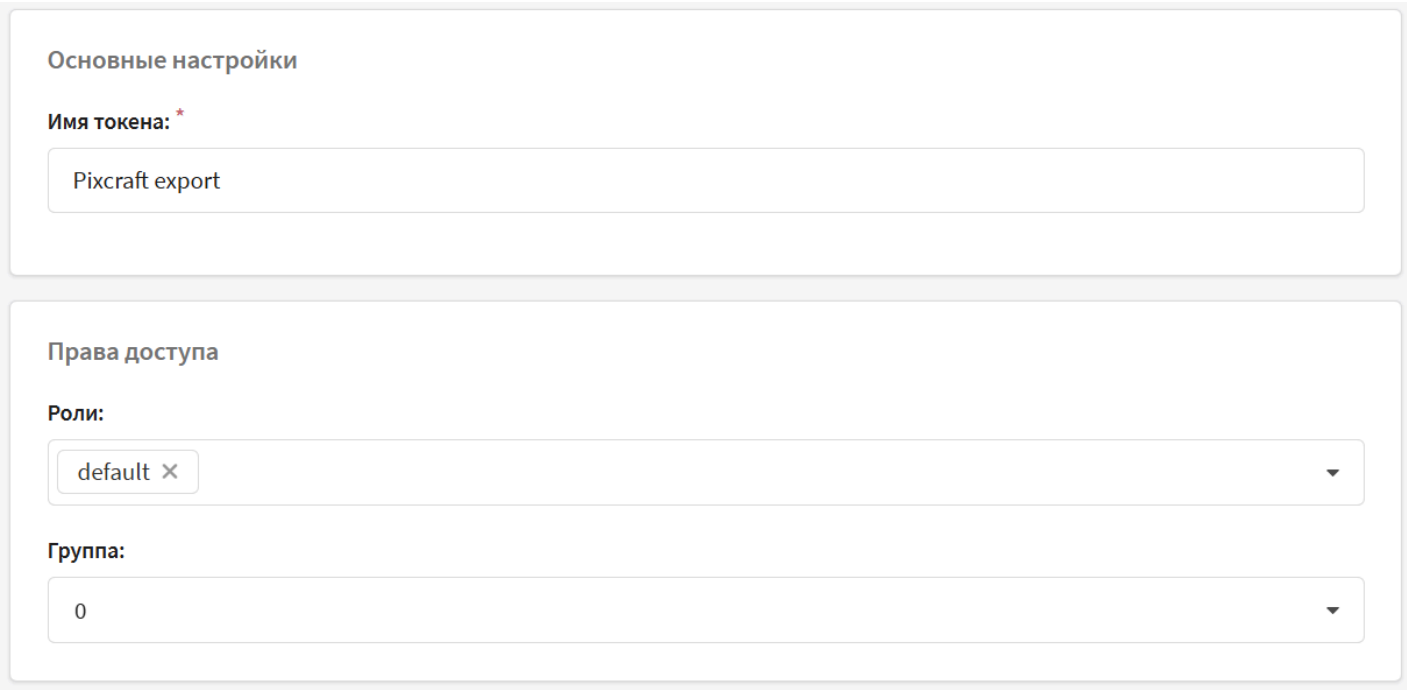
Экспорт шаблона из Pixcraft

Для экспорта из Pixcraft необходимо создать API-токен.

Перейдите в **Настройки/Токены**:

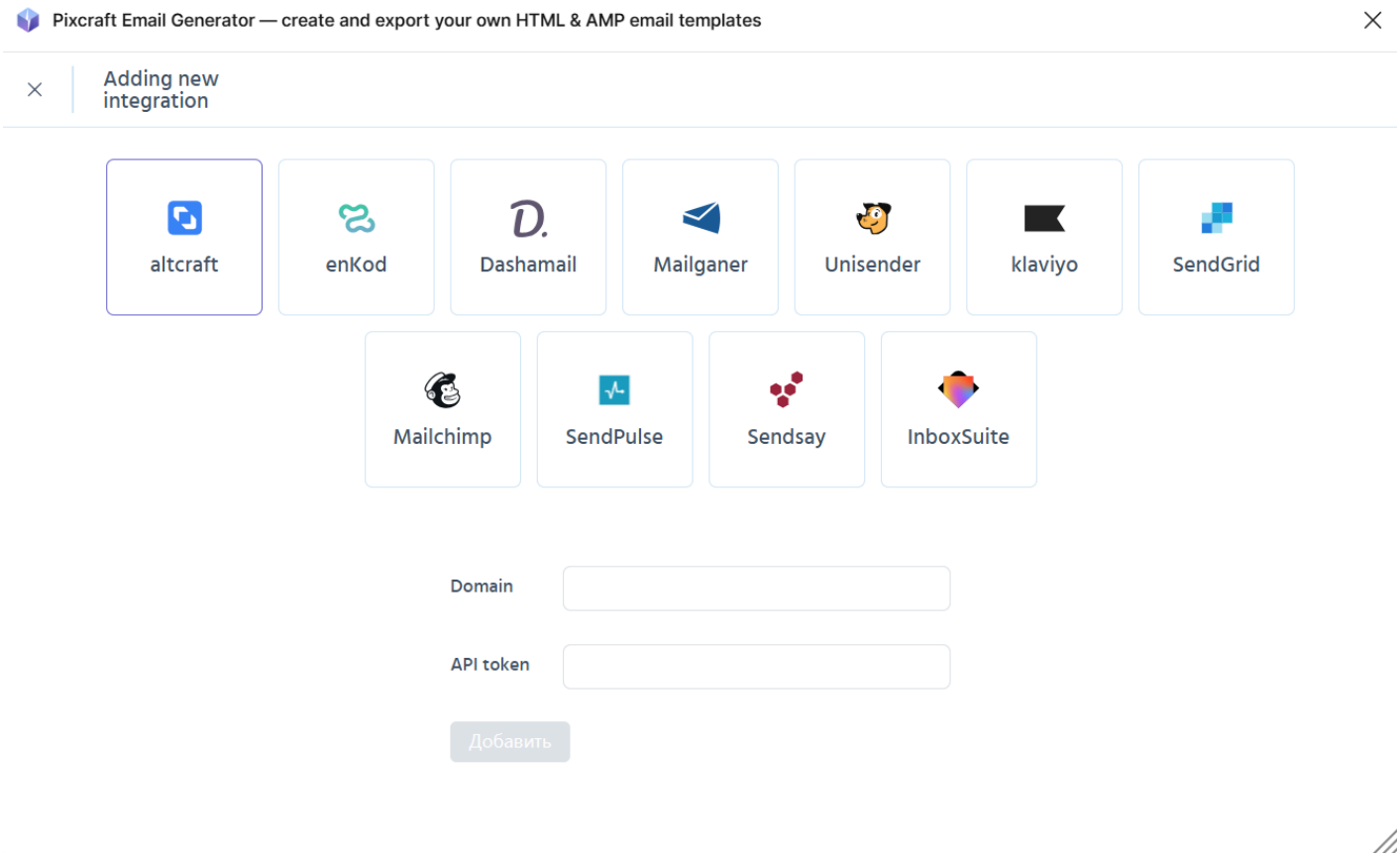
Нажмите **Создать**, укажите название токена, например, Pixcraft export.

Добавьте роль, у которой есть права на **создание и редактирование** шаблонов.



Нажмите **Сохранить**.

Скопируйте API-токен и укажите его при добавлении интеграции в Pixcraft. Для этого в Pixcraft(плагин для Figma) перейдите во вкладку **Экспорт в ESP** и выберите **Altcraft**:



После добавления интеграции выберите ее в списке и нажмите **Export**. Браузер откроет экспортированное письмо в интерфейсе Altcraft.

Импорт шаблона из стороннего сервиса

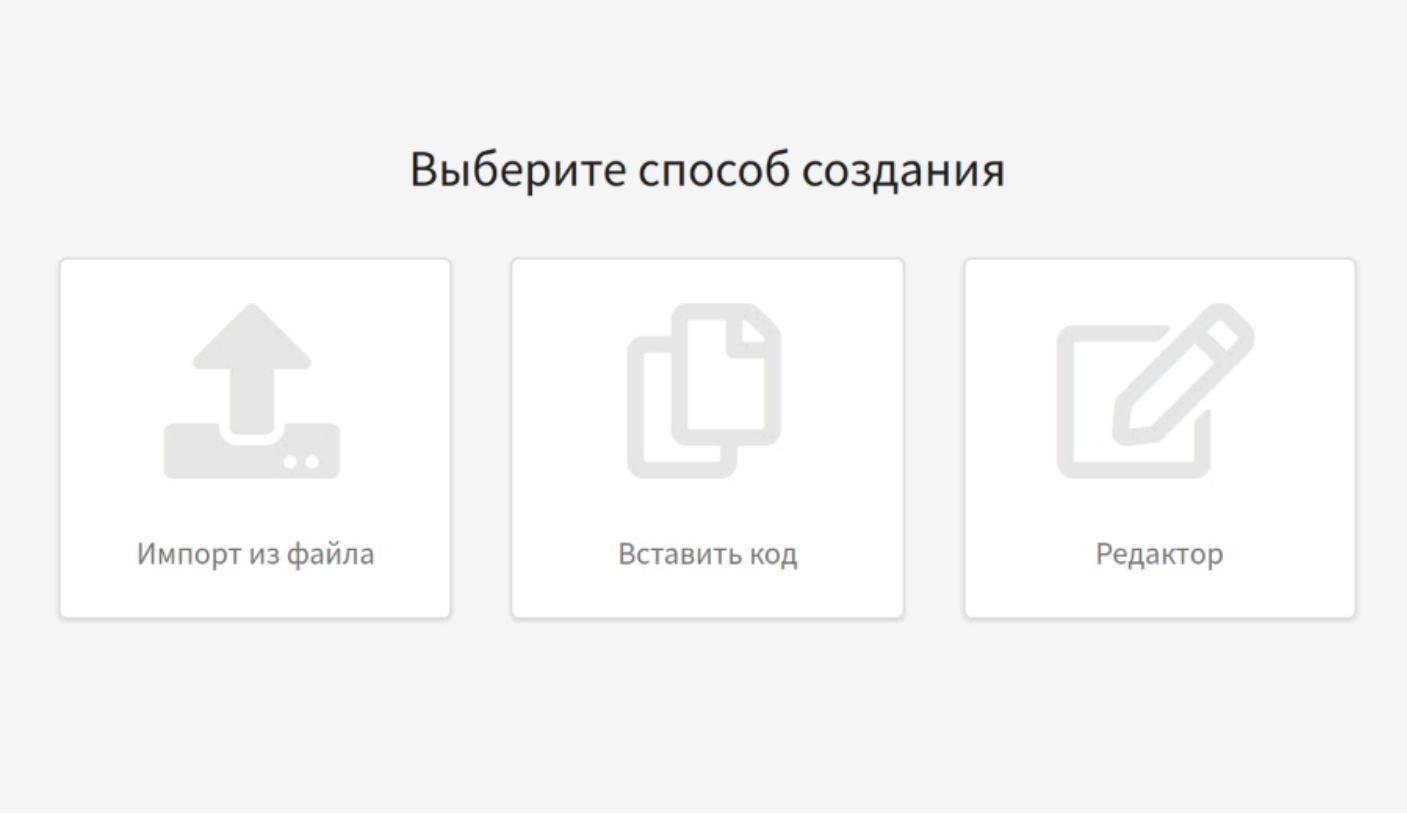
Если у вас есть готовые шаблоны сообщений в других сервисах, вы можете перенести их в Altcraft Platform. Переносить можно как Email, так и SMS шаблоны.

TIP

На данный момент доступен импорт из:

- Unisender
- Mailchimp
- Sendsay

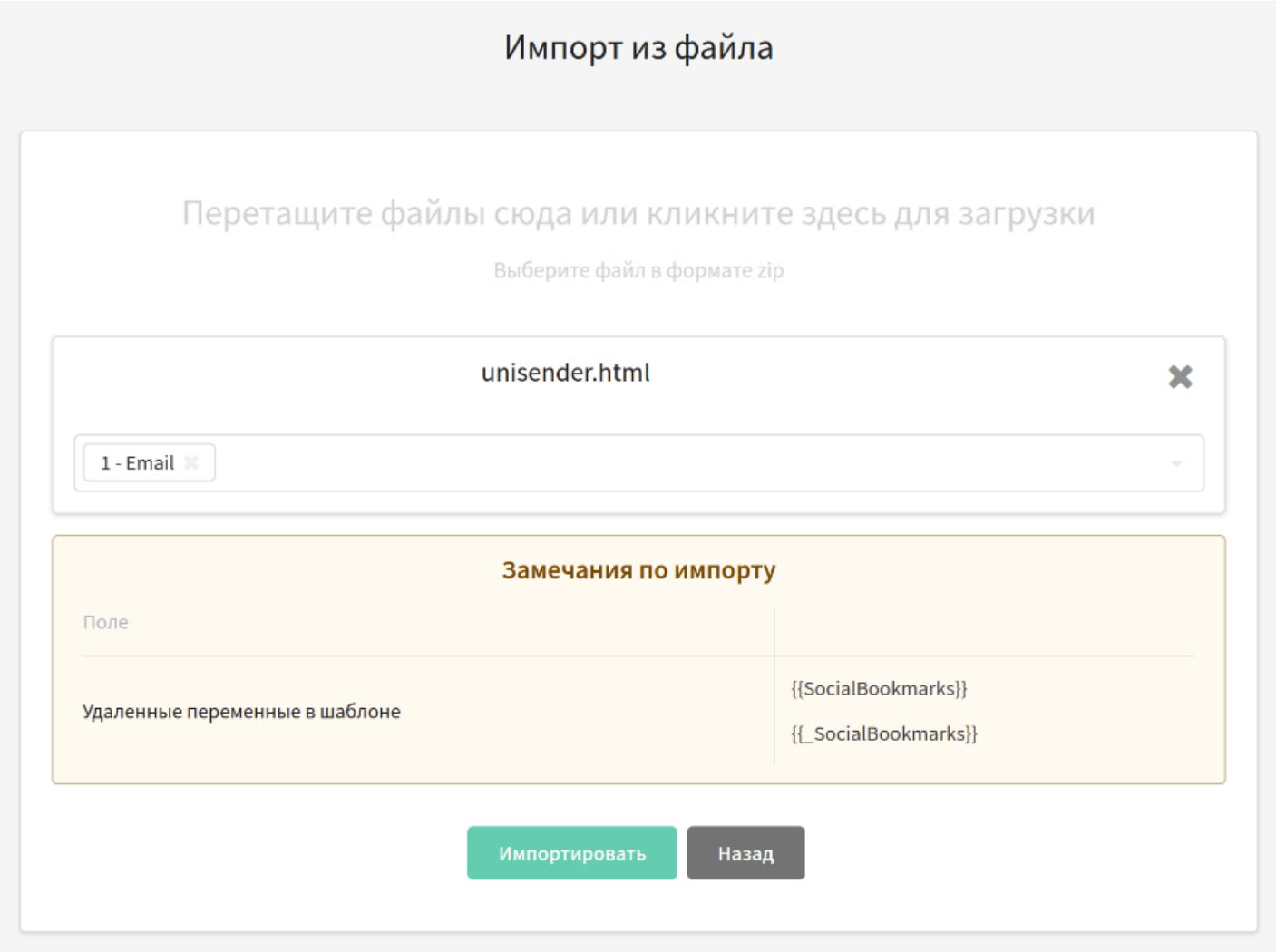
Чтобы приступить к импорту, перейдите в раздел **“Шаблоны”** и нажмите кнопку **“Создать”** на верхней панели. Далее выберите способ создания: импортировать шаблон из файла или вставить код.



Импорт из файла

Если вы хотите перенести в платформу Email-шаблон, вам необходимо предварительно экспортировать **html-файл** с готовым шаблоном из стороннего сервиса. Для переноса SMS-шаблона потребуется **файл в формате txt**.

В платформе кликните на область загрузки и выберите файл на своём устройстве. После загрузки внизу появится окно с **замечаниями по импорту**. Здесь будут перечислены все удалённые из шаблона переменные.



TIP

Каждый сервис использует свои переменные в шаблонах сообщений. В нашей платформе они выглядят так: `{lead.email}`.

При импорте шаблона из стороннего сервиса Altcraft автоматически заменит некоторые переменные. Если аналогичной переменной нет в платформе, она будет удалена из шаблона.

Информацию о том, какие переменные сохранятся, вы найдете ниже.

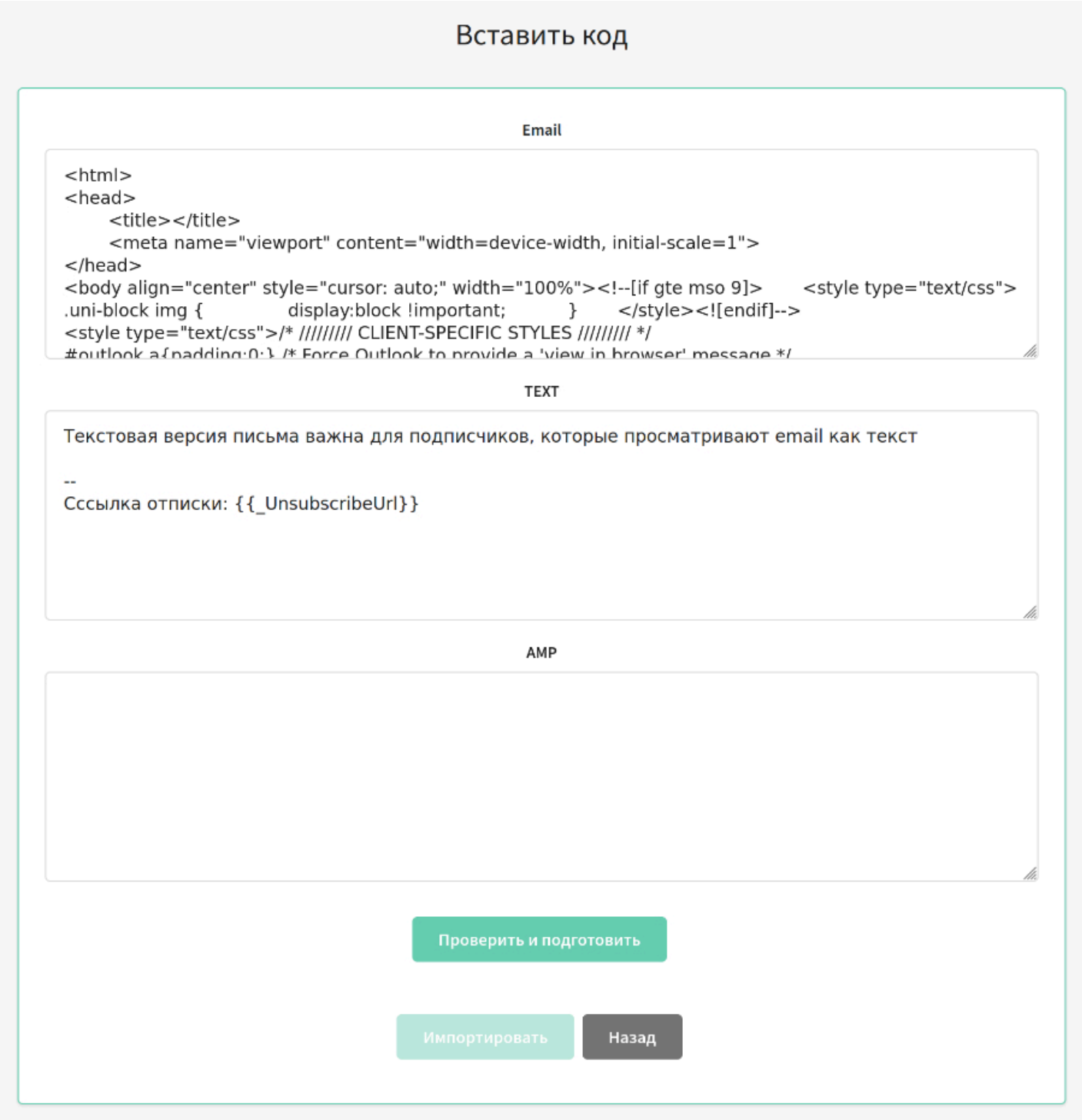
Вставить код

При создании шаблона через вставку кода первое, что вам предложат — выбрать канал (Email или SMS).

Если вы выбираете Email, вам потребуется скопировать html-код шаблона из редактора в стороннем сервисе. Далее добавьте скопированный код в текстовое окно **“Email”**. Текстовая версия письма сгенерируется автоматически.

При необходимости вы можете вручную добавить код для каждой версии письма:

- Email — html-версия письма.
- Text — текстовая версия письма.
- AMP — amp-версия письма.



Импорт шаблона для SMS канала проще — потребуется заполнить только одно текстовое окно.

После добавление кода запустите его проверку, нажав кнопку **“Проверить и подготовить”**. Внизу также отобразится таблица с замечаниями по импорту, где будут перечислены все удаленные переменные. Нажмите **“Импортировать”**.

Автозамена переменных в шаблоне

Altcraft Platform автоматически заменяет переменные стороннего сервиса собственными переменными:

Unisender

Описание переменной	Altcraft	Unisender
Имя	`{lead._fname}`	`{{Name}},` `{{_LocalPart}},` `{{LocalPart}}`
Email-адрес подписки	`{subscription.email}`	`{{_Email}},` `{{Email}}`
Телефон подписки	`{subscription.phone}`	`{{_Phone}}`
Имя отправителя email сообщения	`{from_name}`	`{{_ListTitle}},` `{{ListTitle}}`
Дополнительное поле профиля	`{lead.custom_field}`	`{{custom_field}}`
Ссылка подтверждения	`<a href="#" title="Подтвердите подписку" confirm-link name="link1">Подтвердите подписку</a>` При импорте текстовой версии письма ссылка подтверждения переносится в таком виде: `[Подтвердите подписку](# <confirm-link>)`. Необходимо заменить символ # на URL.	`{{ConfirmUrl}}`
Отписка от ресурса или глобальная отписка	`{unsubscribe}`	`{{_UnsubscribeUrl}},` `{{UnsubscribeUrl}}`
Ссылка на веб-версию письма	`{webversion}`	`{{_WebLetterUrl}},` `{{WebLetterUrl}}`
Полная дата	`{datenow}`	`{{SendDate}}`

Sendsay

Описание переменной	Altcraft	Sendsay
Имя	`{lead._fname}`	`[% anketa.base.firstName %]`
Фамилия	`{lead._lname}`	`[% anketa.base.lastName %]`
Email-адрес подписки	`{subscription.email}`	`[% anketa.base.email %]`
Домен email-подписки	`{subscription.email_domain}`	`[% anketa.member["confirm.host"] %]`
Пол	`{lead._sex}`	`[% anketa.base.gender %]`
Город	`{lead._city}`	`[% anketa.base.city %]`
Дата рождения	`{lead._bdate}`	`[% anketa.base.birthDate %]`
Дополнительное поле профиля	`{lead.custom_field}`	`[% anketa.base.custom_field %]`
Условие	`{if lead.custom_field}{else}{end}`	`[% IF anketa.base.custom_field %][% ELSE %][% END %]`
Длина переменной	`{length(lead.custom_field)}`	`[% size(anketa.base.custom_field) %]`

Итерация по массиву	`{for $index $i = items}`
	`  {$i}`
	`{end}`
		`[% FOREACH i IN items %]`
		`  [% i %]`
		`[% END %]`

Mailchimp

Описание переменной	Altcraft	Mailchimp
Имя	`{lead._fname}`	*|FNAME|*
Фамилия	`{lead._lname}`	*|LNAME|*
Email-адрес подписки	`{subscription.email}`	*|EMAIL|*
Телефон подписки	`{subscription.phone}`	*|PHONE|*
Дата регистрации	`{lead._regdate}`	*|OPTIN_DATE|*, *|OPTIN_DATETIME|*
Имя отправителя email сообщения	`{from_name}`	*|LIST:DESCRIPTION|*, *|LIST:NAME|*
Email отправителя	`{from_email}`	*|LIST:EMAIL|*, *|ABUSE_EMAIL|*

Описание переменной	Altcraft	Mailchimp
Дополнительное поле профиля	`{lead.custom_field}`	*|custom_field|*
Ссылка подтверждения	`<a href="#" title="Подтвердите подписку" confirm-link name="link1">Подтвердите подписку</a>` При импорте текстовой версии письма ссылка подтверждения переносится в таком виде: `[Подтвердите подписку](# <confirm-link>)`. Необходимо заменить символ # на URL.	*|LIST:SUBSCRIBE|*

Отписка от ресурса или глобальная отписка	`{unsubscribe}`	*|UNSUB|*,
		
		`{{unsub}}`,
		
		`{{unsub "http://example.com/unsub"}}`
		
		`{{unsub redirect_merge_var}}`

Ссылка на веб-версию письма	`{webversion}`	*|ARCHIVE|*
Полная дата	`{datenow}`	*|DATE:d/m/y|* `{{date "Y-m-d"}}`
Год полностью	`{yearlong}`	*|CURRENT_YEAR|*
ID ресурса	`{resid}`	*|LIST:UID|*
SUBID рассылки	`{subid}`	*|CAMPAIGN_UID|*
URL веб-сайта, формы или приложения из настроек ресурса	`{resurl}`	*|LIST:URL|*
Верхний регистр	`{uppercase("your text")}`	`{{upper "your text"}}`
Первый символ в верхнем регистре	`{uppercasefirst("your text is neat")}`	`{{title "your text is neat"}}`
Условие	`{if lead.custom_field}{else}{end}`	*|IF:custom_field|* …*|ELSE:|*… *|END:IF|*

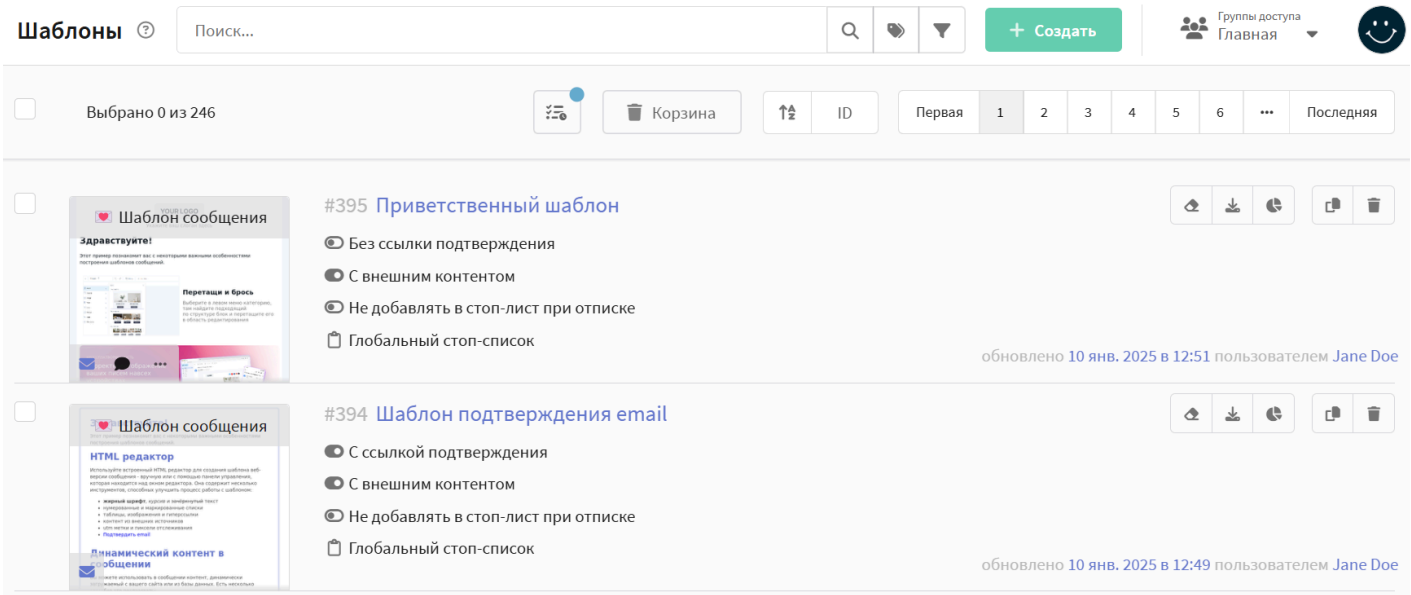
TIP

Если вы хотите расширить синтаксис переноса переменных, отправьте запрос на team@altcraft.com.

Работа с шаблонами сообщений

	Для того чтобы посмотреть существующие шаблоны и добавить новые, откройте раздел **Шаблоны** в главном меню, затем перейдите в одноименный пункт.

Перед вами откроется список шаблонов, доступных в аккаунте или в выделенной группе доступа. Чтобы найти шаблон в общем списке, введите в поисковую строку название шаблона или его ID. Также вы можете отфильтровать шаблоны по тегам, каналу отправки, отобразить только черновики или уже готовые шаблоны.



Для каждого шаблона собрана общая информация:

- **Наличие ссылки подтверждения** — шаблон с такой ссылкой необходим при использовании Double opt-in подписки.
- **Наличие внешнего контента** — использование в шаблоне ссылки на динамический JSON-контент.
- **Добавление в стоп-список при отписке** — если подписчик перейдет по ссылке отписки в шаблоне сообщения, он попадет в выбранный стоп-список.
- **Стоп-список** — рассылка, где используется шаблон, заблокирует отправку письма подписчику, чьи контакты есть в указанном стоп-списке.

Справа доступно меню быстрых действий:

— включить режим черновика;

— экспортировать шаблон;

— перейти в отчёт по шаблону;

— клонировать шаблон;

— удалить шаблон.

При нажатии на миниатюру шаблона откроется просмотр сообщения. Здесь вы можете проверить, как сообщение будет выглядеть в различных каналах и на различных устройствах.

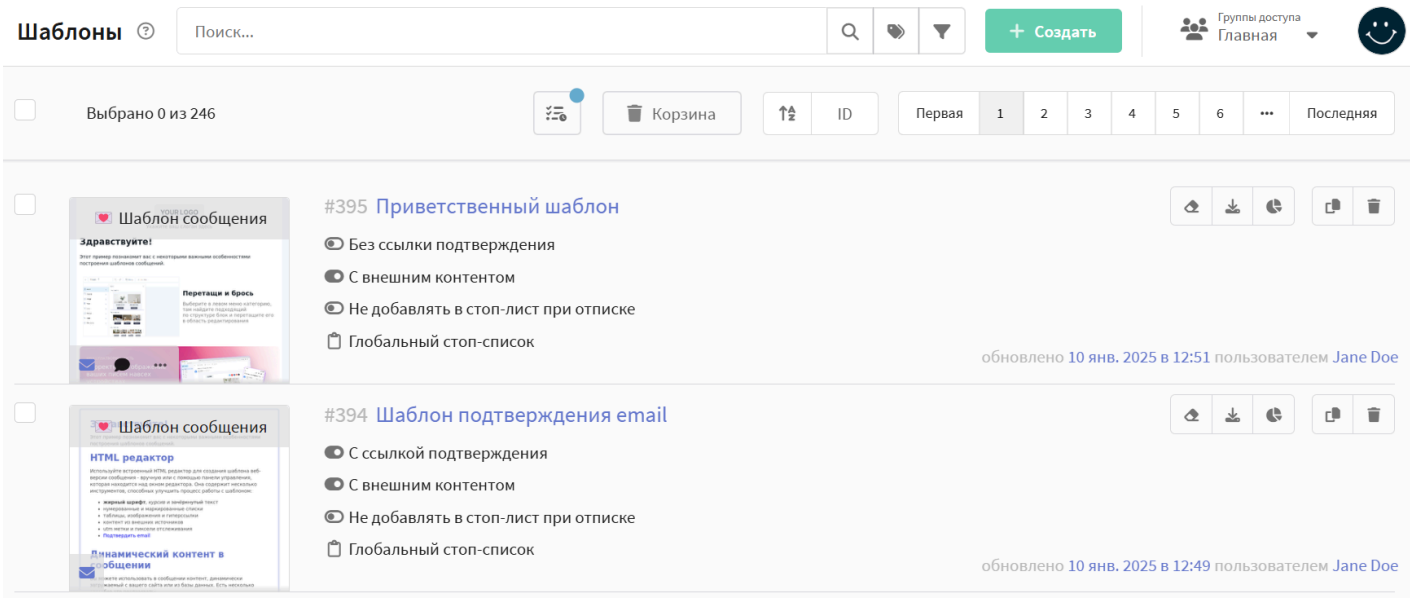
Чтобы добавить новый шаблон, в верхней части страницы нажмите кнопку **+ Создать**. Если необходимо отредактировать существующий шаблон, нажмите на его заголовок.

Подробнее о работе с редактором шаблонов и о шаблонах для каждого канала коммуникации читайте в следующих статьях:

Работа в редакторе

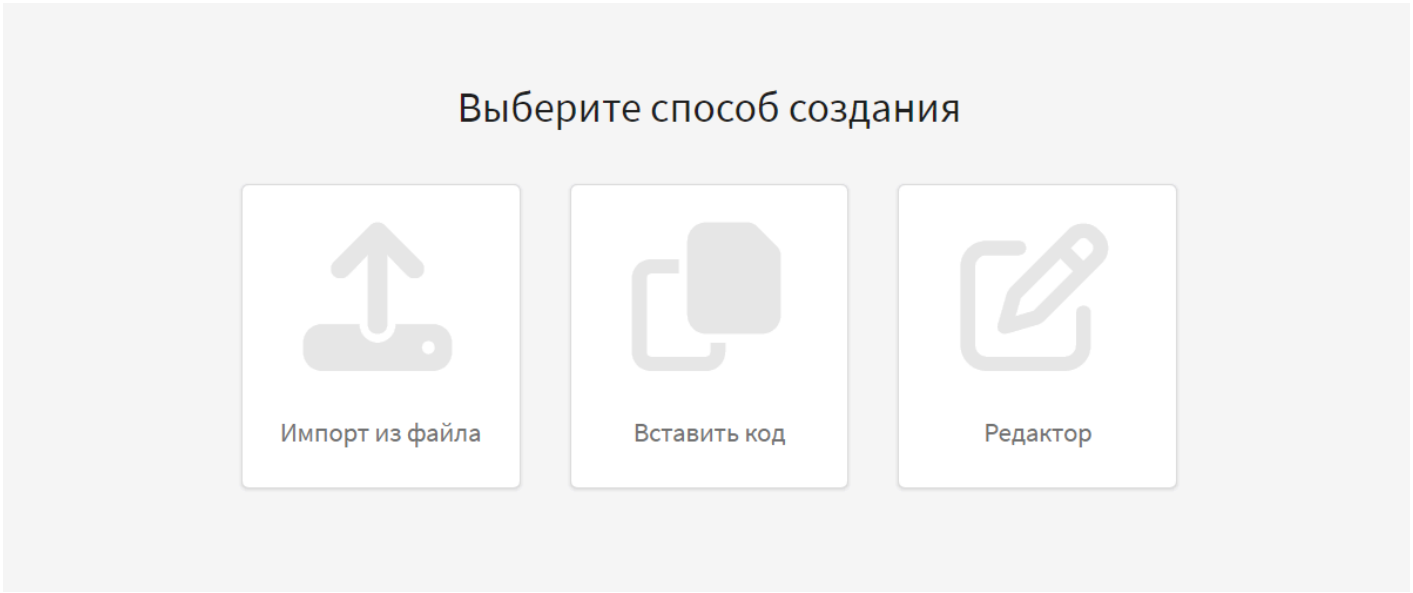
Создание шаблона

Чтобы добавить новый шаблон, в верхней части страницы нажмите кнопку **+ Создать**:



Далее выберите способ создания шаблона:

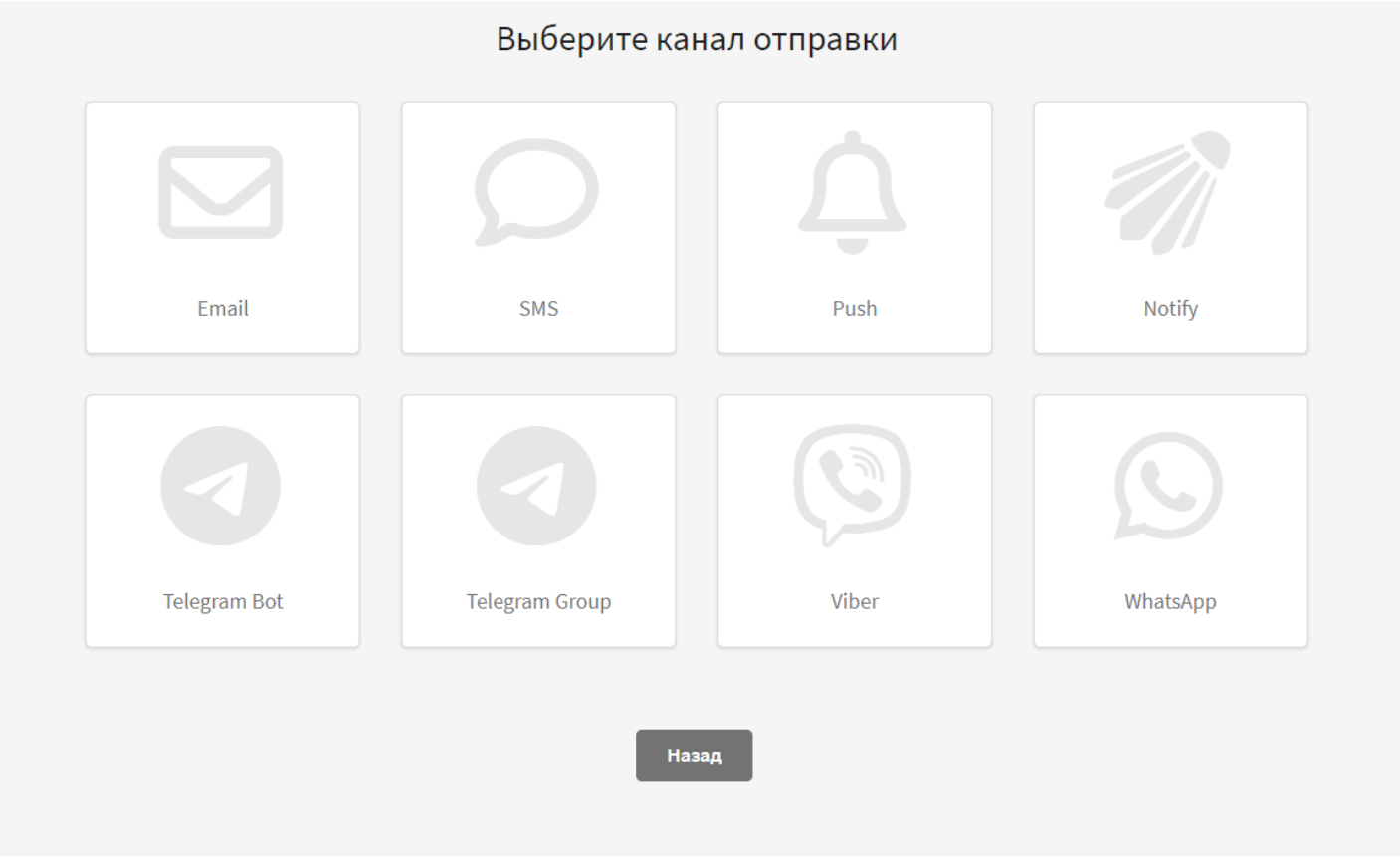
- **Импорт из файла** — загрузка готового шаблона в виде zip-архива или html-файла (Email-канал).
- **Вставить код** — добавление скопированного html-кода в текстовое окно (Email- и SMS-каналы).
- **Редактор** — создание шаблона в редакторе (все каналы).



TIP

Если вам необходимо перенести шаблон из стороннего сервиса, воспользуйтесь импортом из файла или вставкой кода.

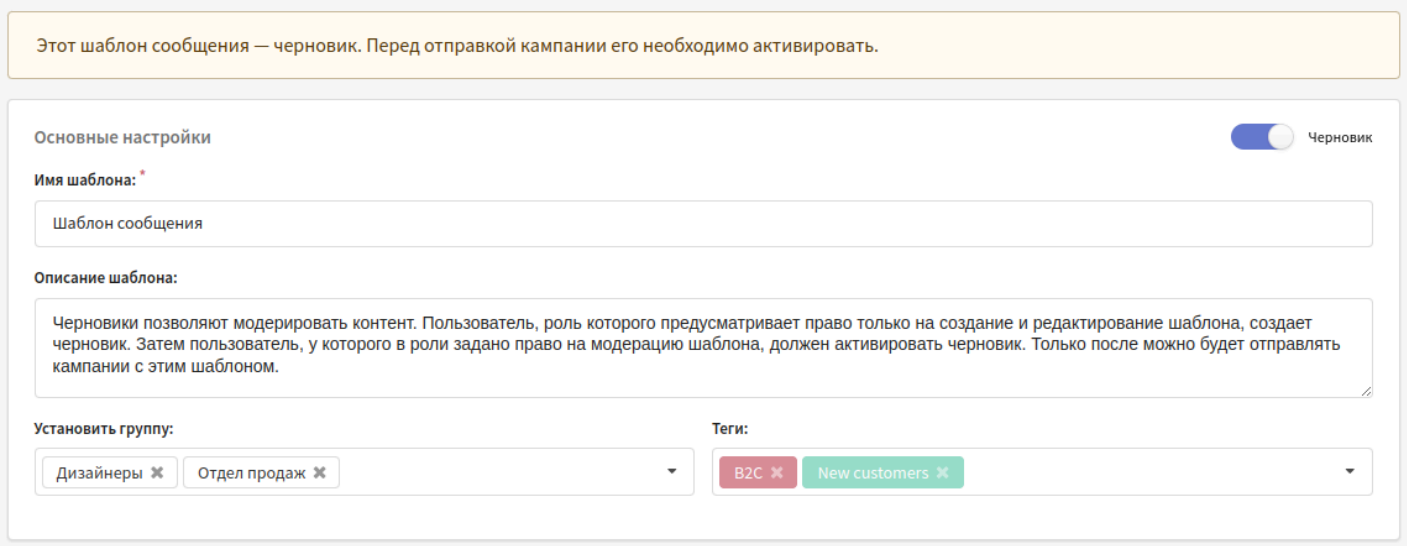
Выберите **канал отправки**. В дальнейшем этот канал можно будет изменить или добавить новый.



Основные настройки

В **основных настройках** шаблона укажите его название и описание, добавьте **теги** и при необходимости установите **группы доступа**.

Если необходимо организовать модерацию контента, в шаблонах предусмотрена функция создания **Черновиков**. Для этого нужно создать пользователя с ролью, предусматривающей только создание сообщений, но не их модерацию. Этот пользователь сможет наполнять шаблоны содержимым, но такие шаблоны нельзя будет сразу использовать в рассылках. Второй пользователь, с правами на модерацию, должен открыть шаблон и деактивировать опцию **Черновик** в **Основных настройках**.

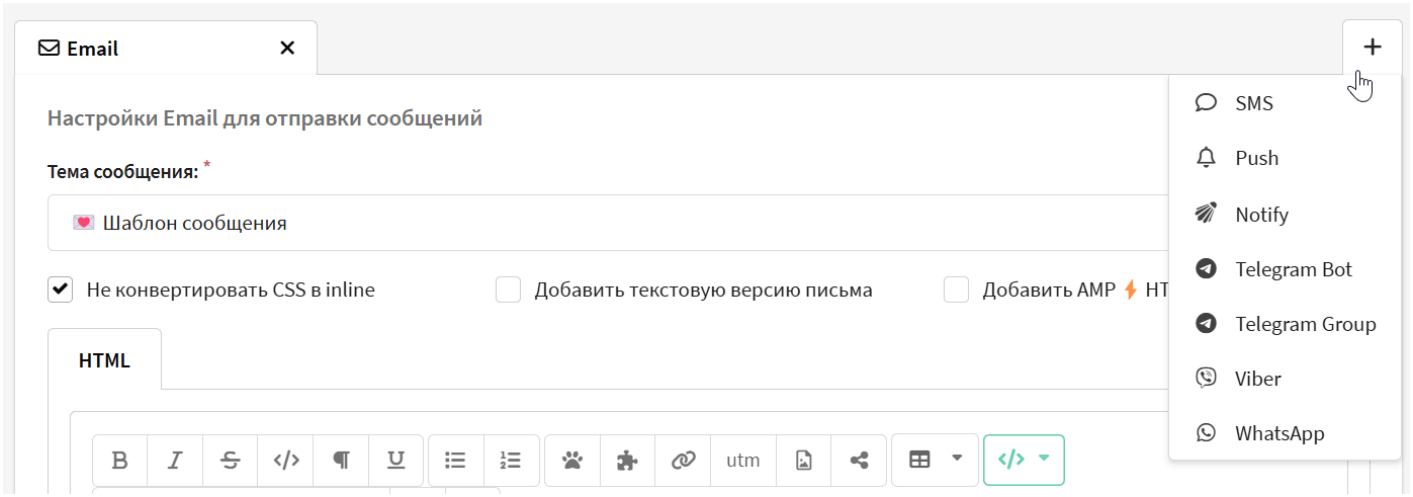


Редактор контента

Встроенный в платформу редактор помогает создавать красивые сообщения, быстро редактировать уже существующие шаблоны, а также упрощает персонализацию контента.

Один шаблон может объединять контент для отправки по всем доступным каналам связи. Канал отправки вы определяете на этапе создания шаблона, однако его можно изменить или добавить новый. Чтобы добавить в шаблон **контент** для определённого **канала отправки**, нажмите на кнопку

в правом верхнем углу:



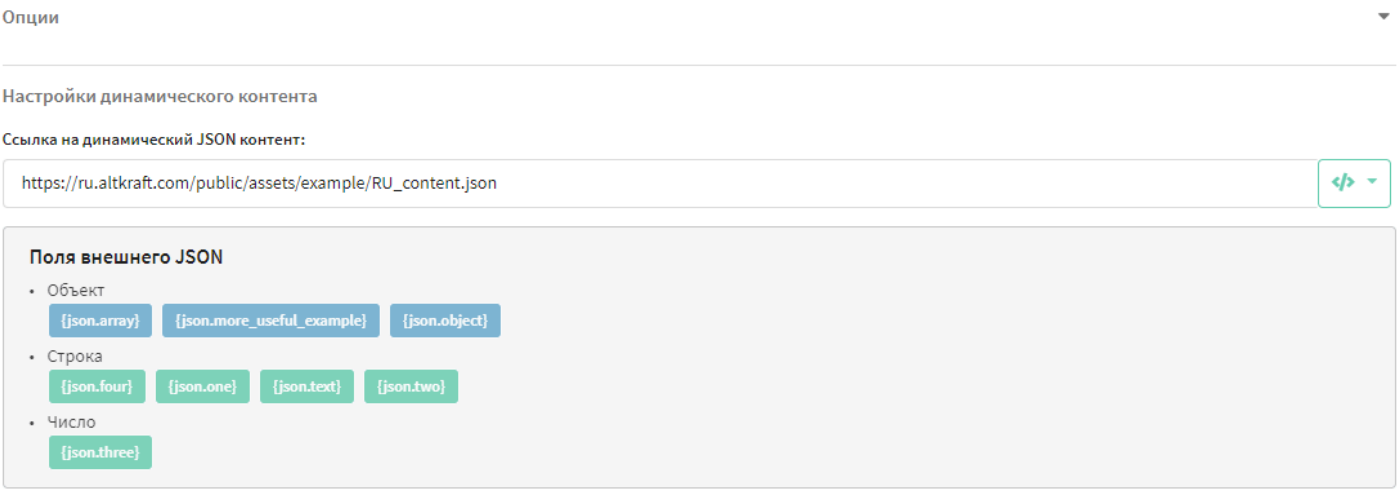
Работа с каждым типом контента описана в отдельных статьях:

- Email;
- SMS;
- Push;
- Telegram;
- WhatsApp;
- Viber™;
- Notify.

Опции сообщения

В разделе **Опции** доступны расширенные настройки шаблона.

В блоке **Настройки динамического контента** можно указать источник данных для JSON-контента в сообщениях. При наличии валидного JSON по введённому адресу откроется предпросмотр объектов первого уровня:



CAUTION

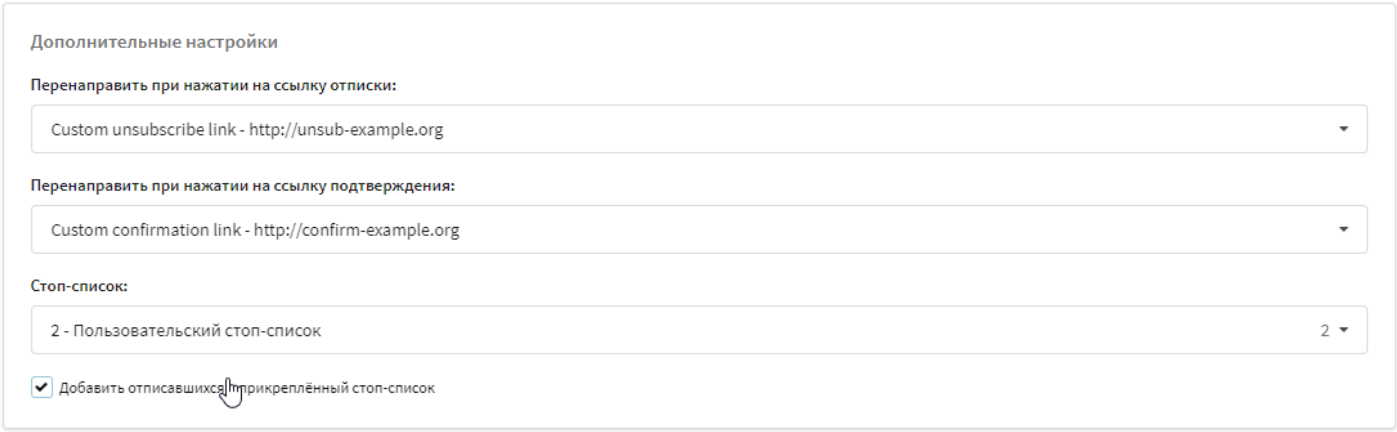
По умолчанию размер динанамического JSON **ограничен 1 МБ**. В **On-premises версии** этот лимит можно изменить в файле конфигурации через параметр `MAX_REMOTE_CONTENT_BYTES`.

В **Дополнительных настройках** вы можете выбрать для сообщения Пользовательские ссылки **отписки** и **подтверждения подписки**. Например, если вы хотите направлять клиентов в личный кабинет сайта или на страницу управления подписками.

TIP

Пользовательские ссылки отписки и подтверждения могут создавать **мастер-пользователи** аккаунта. В опциях шаблона возможно только выбрать предварительно созданные ссылки. Подробнее в разделе **Настройки**.

Также в дополнительных настройках можно выбрать пользовательский Стоп-список для использования с шаблоном. С помощью этой функции легко ограничить отправку клиентам, которым не подходит тип рассылаемого контента. Дополнительно можно включить опцию **Добавить отписавшихся в прикреплённый стоп-список:**

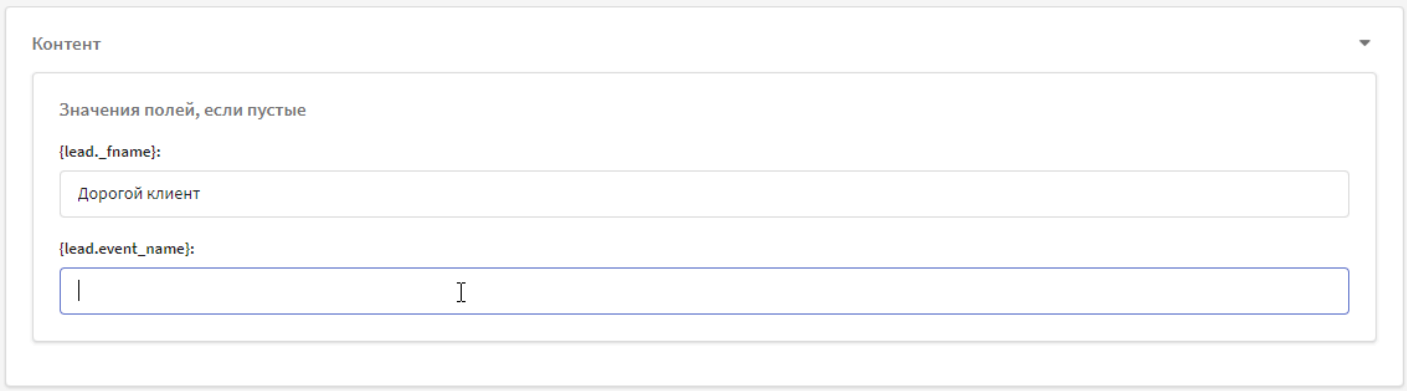


Значения полей по умолчанию и Части

Во вкладке **Контент** вы можете определить значения по умолчанию для **переменных**, которые вы используете в шаблоне. Так, если вы используете подстановку имени клиента, но у некоторых профилей поле **Имя** не заполнено, будет использоваться значение по умолчанию, например “дорогой Клиент”.

TIP

Вкладка **Контент** появляется в редакторе, когда вы используете в шаблоне переменные с данными профиля или **части** сообщения.

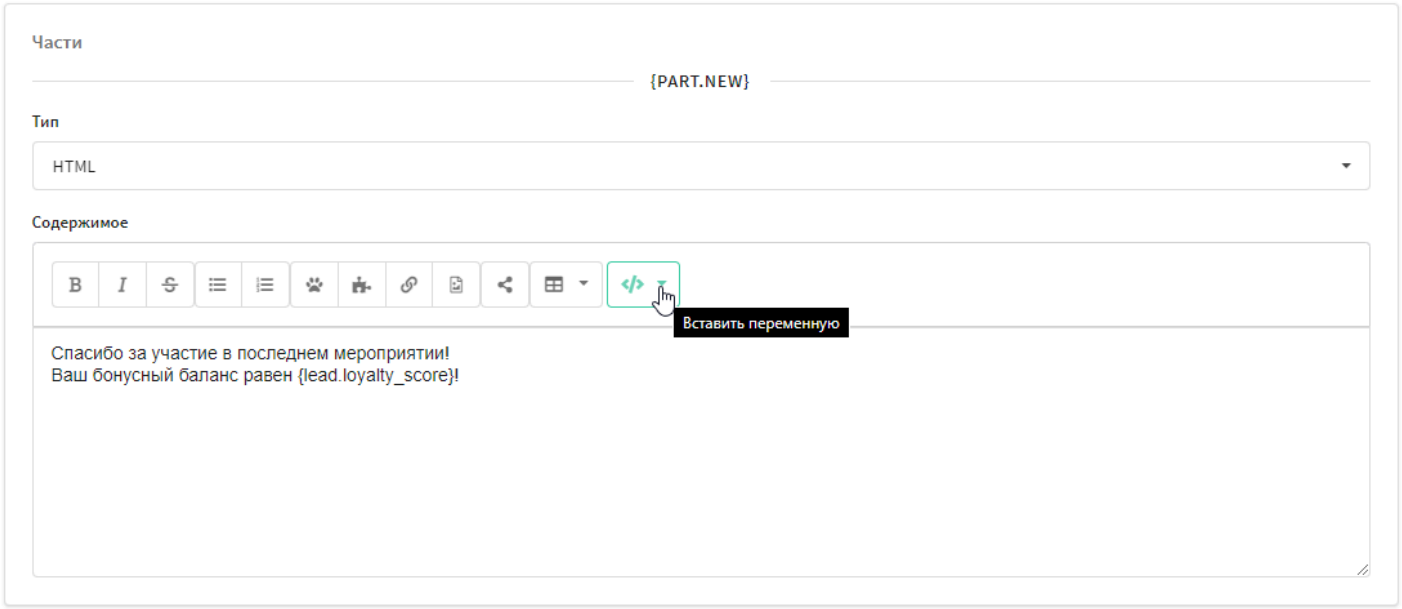


Части сообщения аналогичны **Фрагментам**, однако они могут использоваться только в том шаблоне, где были созданы. Фрагменты можно использовать в нескольких шаблонах.

TIP

Чтобы начать работу с частями, добавьте в код шаблона переменную `{part.example_part}`. После этого внизу страницы появится вкладка “Контент”, где вы сможете настроить части шаблона.

Для каждой части можно выбрать **тип контента:** HTML для email сообщений, или текст — для текстовой версии email, а также для SMS и Push контента. В зависимости от типа контента будет открыт соответствующий редактор контента со всеми необходимыми функциями:



TIP

Использовать части удобно вместе с логическими выражениями для более компактного отображения кода сообщения. Например, в зависимости от количества баллов подписчика (Кастомное числовое поле) будет подставлена одна или другая часть:

```
{if lead.score gte 9000}
{part.part_1}
{else}
{part.part_2}
{end}
```

Предпросмотр сообщения

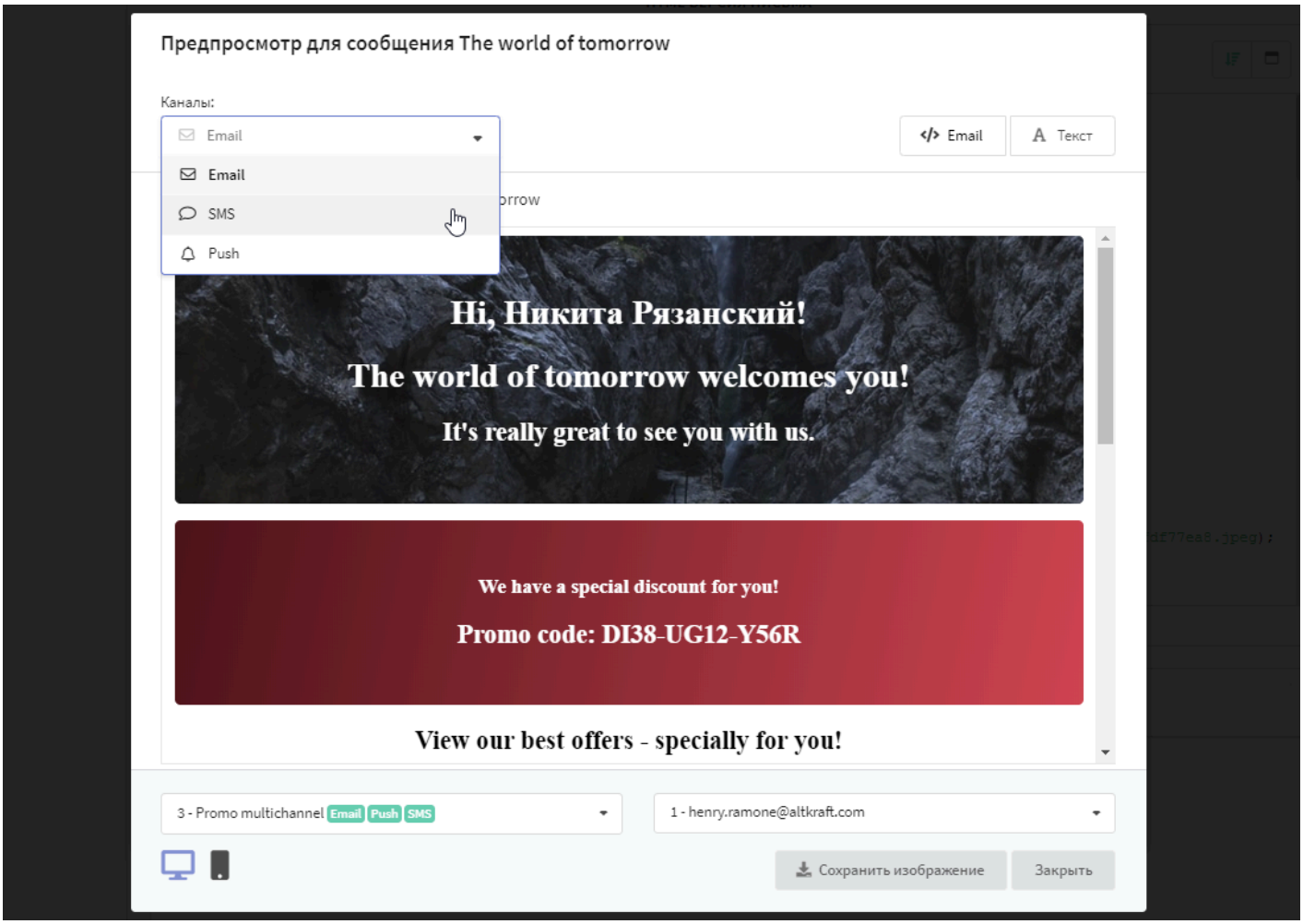
Предпросмотр сообщения позволяет не только проверить вёрстку HTML, но и увидеть результаты персонализации контента на основе данных подписчика. Нажмите на эту кнопку в верхней части страницы, чтобы активировать предпросмотр:

В окне предпросмотра вы можете выбрать необходимый **канал**, контент для которого хотите просмотреть. Для Email доступен предпросмотр как HTML, так и текстовой версии.

В нижней части окна находятся кнопки переключения десктопного/мобильного режимов отображения, а также настройки персонализации.

TIP

Выберите ресурс, от которого планируете отправить шаблон, и в выпадающем меню добавьте новый профиль. Поиск аналогичен поиску по базе данных — используются правила сегментации: вы сразу сможете выбрать профиль с необходимыми данными.



Для любого канала доступна выгрузка предпросмотра шаблона в формате png-изображения. Чтобы скачать изображение на своё устройство, нажмите кнопку **“Сохранить изображение”**.

Для корректной выгрузки предпросмотра email-шаблона следуйте правилам:

1. CSS-стили `text-decoration`, `text-decoration-line` и `text-decoration-color` устанавливаются непосредственно в родительский элемент текста. Например:

```
<div>
  <span style="text-decoration: line-through;">Текст</span>
</div>
```

2. Если вы используете шрифты, импортированные из внешнего источника (например, Google Fonts), то рекомендуется заменить их — задайте в свойстве `font-family` шрифты, установленные в систему. Например:

```
<!-- Корректное отображение в превью -->
<span style="font-family: Roboto, Arial, Tahoma, sans-serif">Текст</span>

<!-- Некорректное отображение в превью -->
<link
  href="https://fonts.googleapis.com/css2?
family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1
,900&display=swap"
  rel="stylesheet"
/>
```

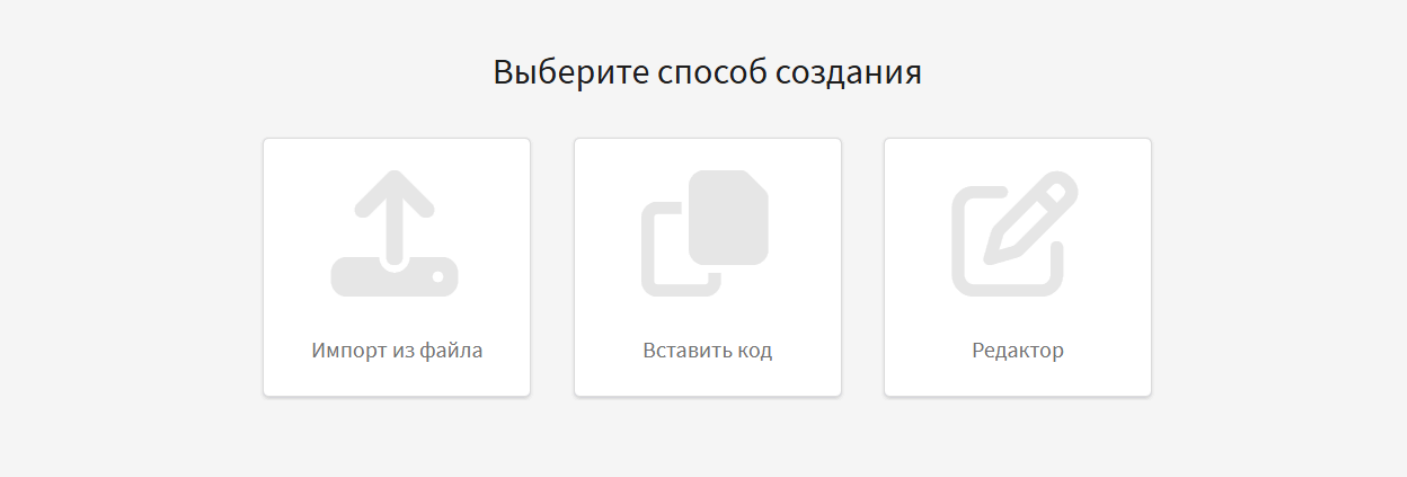
---

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Email-шаблон

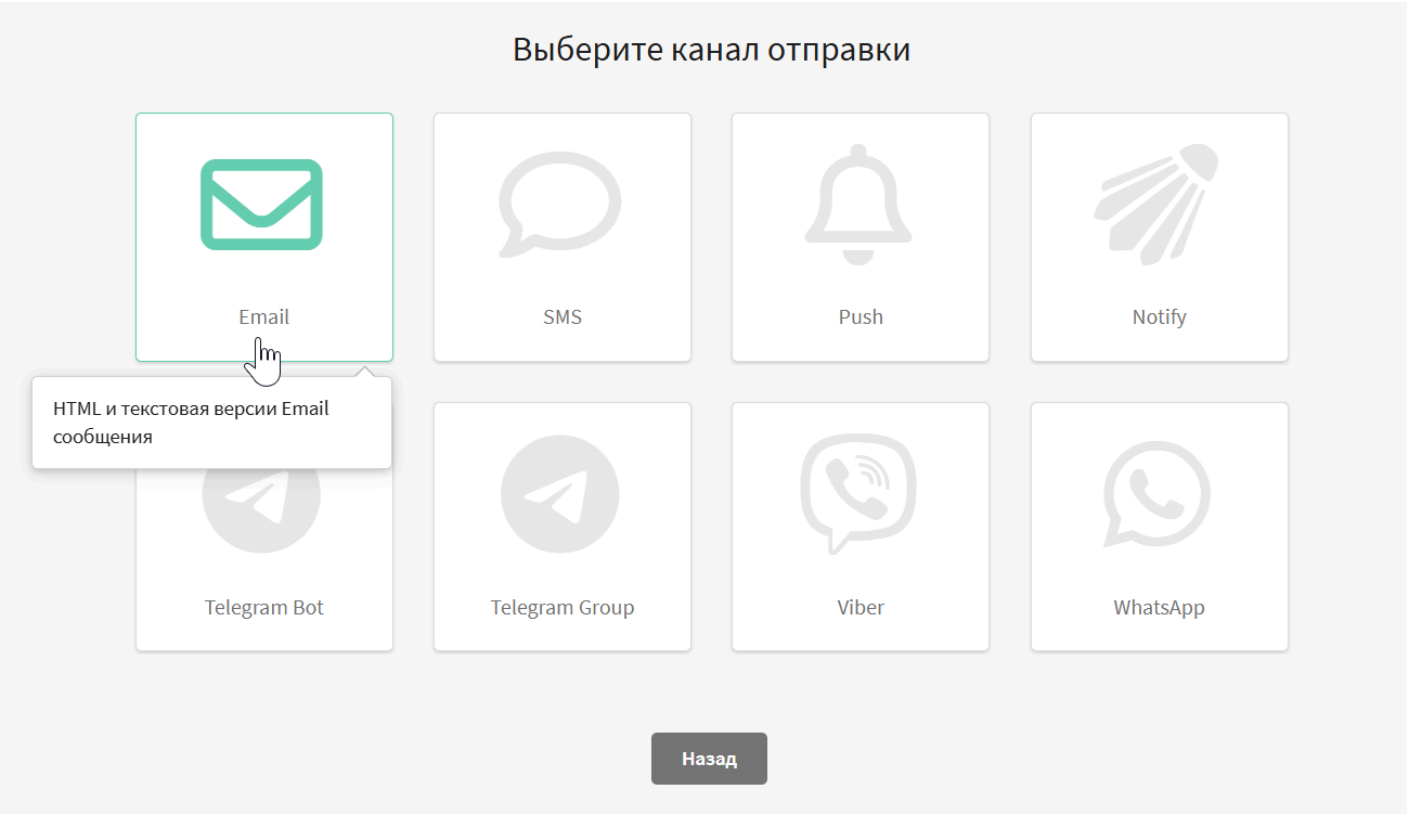
С чего начать

В главном меню выберите раздел **“Шаблоны”** и создайте новый шаблон сообщения. Далее выберите **способ создания**:

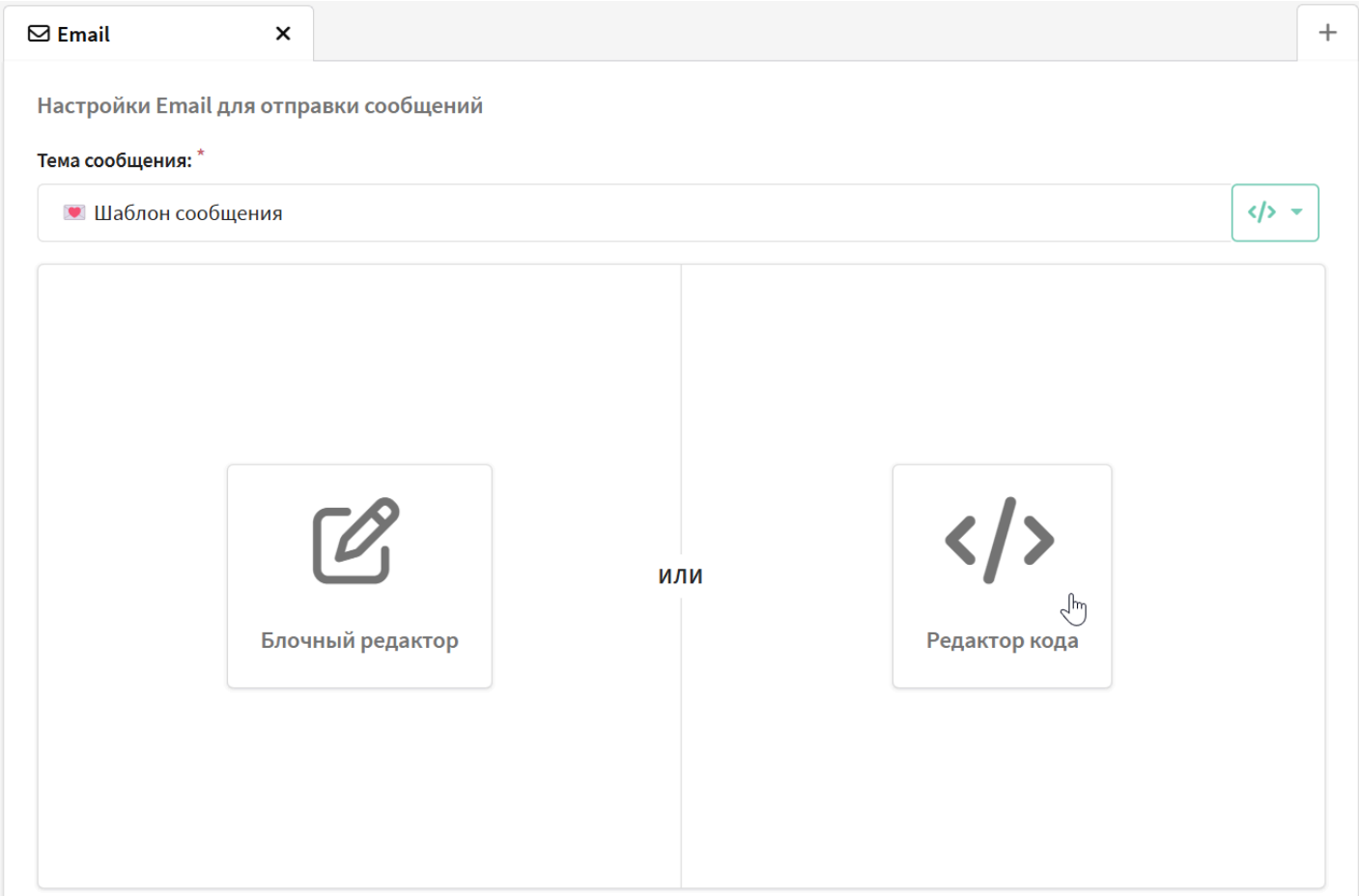


Если у вас уже есть готовый шаблон, вы можете импортировать его в платформу из файла или вставить HTML-код. Структура импортируемых шаблонов описана в **этой статье**. После загрузки шаблона вы автоматически перейдете в редактор.

Чтобы сразу начать работу в редакторе, выберите **Редактор —> Email**:

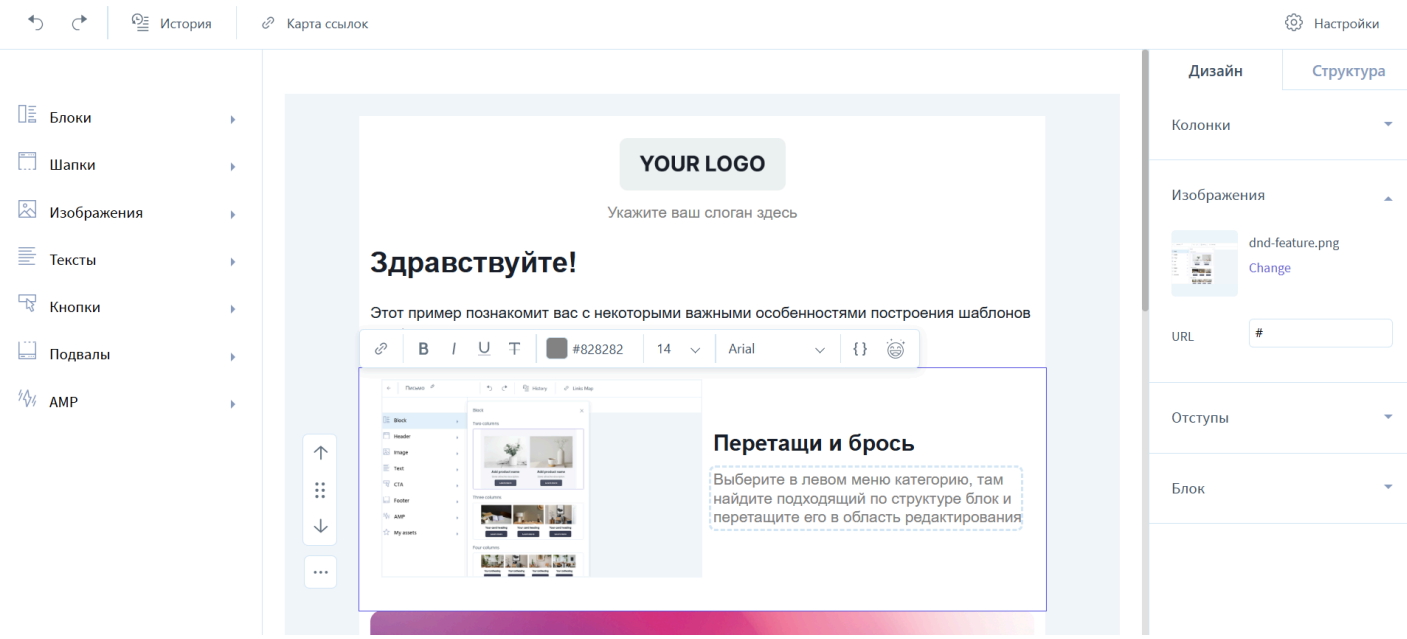


В окне настройки нового email-шаблона вам необходимо выбрать редактор, в котором продолжить работу.

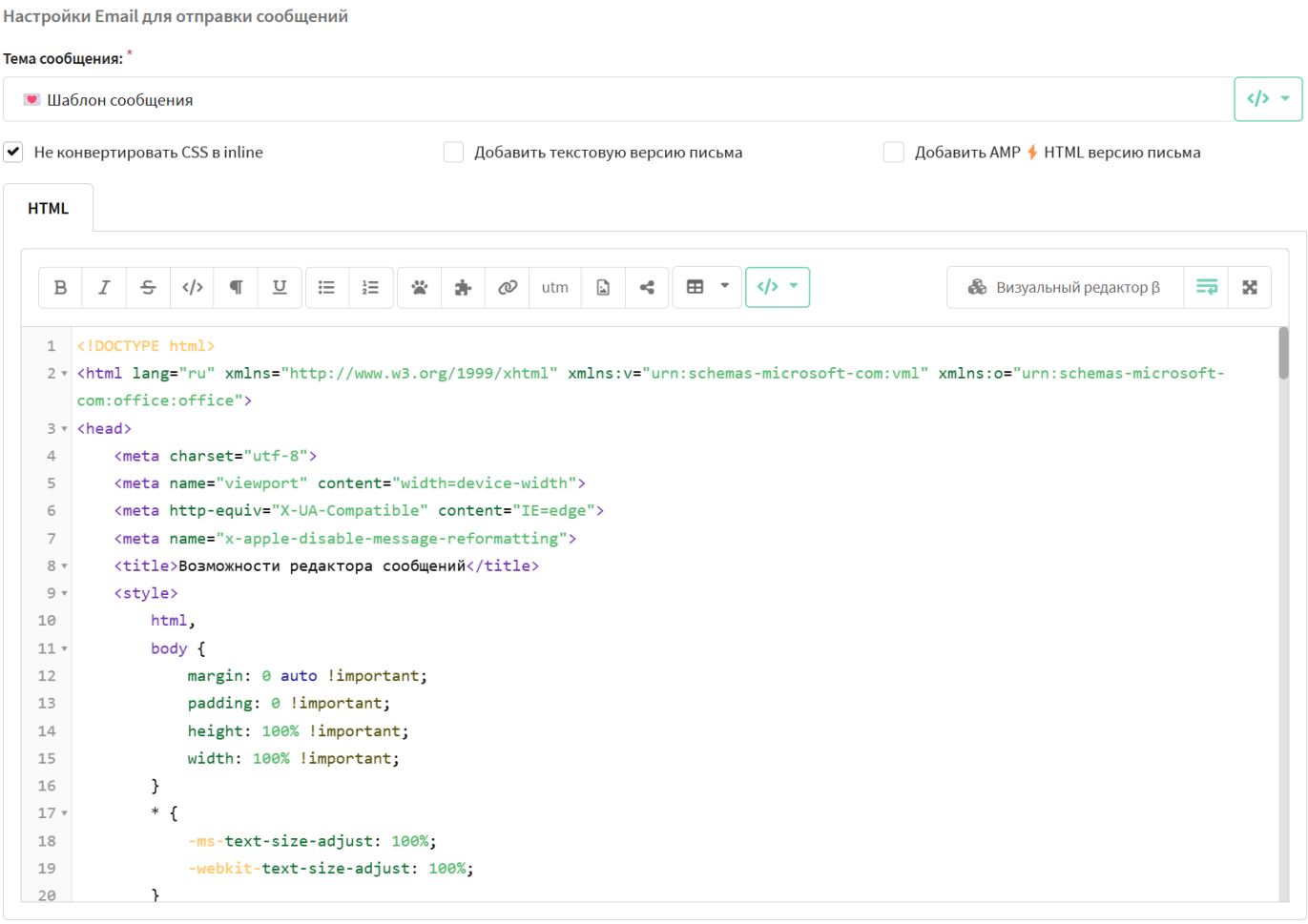


В платформу встроено три редактора:

1. Блочный редактор позволяет быстро создать шаблон с нуля, используя готовые блоки. Он подойдет тем, кто не знаком с HTML и CSS. Также в этом редакторе доступны AMP-элементы для создания интерактивных писем. Подробнее о возможностях блочного редактора читайте в **этой статье**.



2. Редактор кода предназначен для работы с HTML и CSS. Используйте его, чтобы внести изменения в уже готовый шаблон. Подробнее о работе в редакторе кода читайте **ниже**.

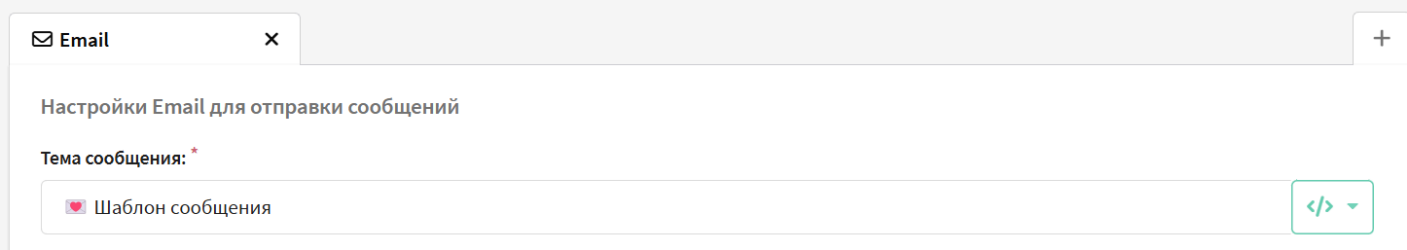


3. Визуальный реадактор подходит для редактирования готового email-шаблона. С его помощью вы можете быстро изменить цвет текста, добавить ссылку или заменить изображение без знания HTML и CSS. Чтобы перейти в визуальный редактор, сперва откройте редактор кода, после чего нажмите кнопку **“Визуальный редактор β”** в правом верхнем углу. Работа в визуальном редакторе описана в **этом разделе**.

> **CAUTION**
>
> Обратите внимание, вы можете перейти из блочного редактора в редактор кода, но обратный переход невозможен.

Настройки email-шаблона

В основных настройках контента для email-канала укажите **Тему письма**, которая будет отображаться в почтовом клиенте получателя. Вы можете персонализировать тему, воспользовавшись меню переменных `</>`, или вручную, используя системные переменные. Также в тему можно добавить эмодзи.

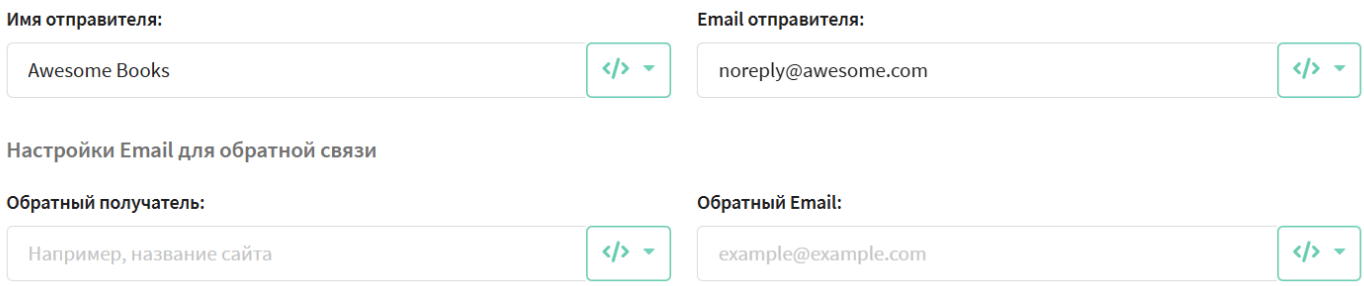


Ниже заполните следующие поля:

- **Имя отправителя** — имя, от которого письмо отправляется подписчику. Можно добавить персонализацию и эмодзи.
- **Email отправителя** — адрес, от которого письмо отправляется клиенту.

Для **настройки обратной связи** заполните следующие поля:

- **Обратный получатель** — имя для ответа на сообщение. Доступна персонализация: например, вы можете указать здесь имя личного менеджера клиента.
- **Обратный email** — электронный адрес для ответа на сообщение.

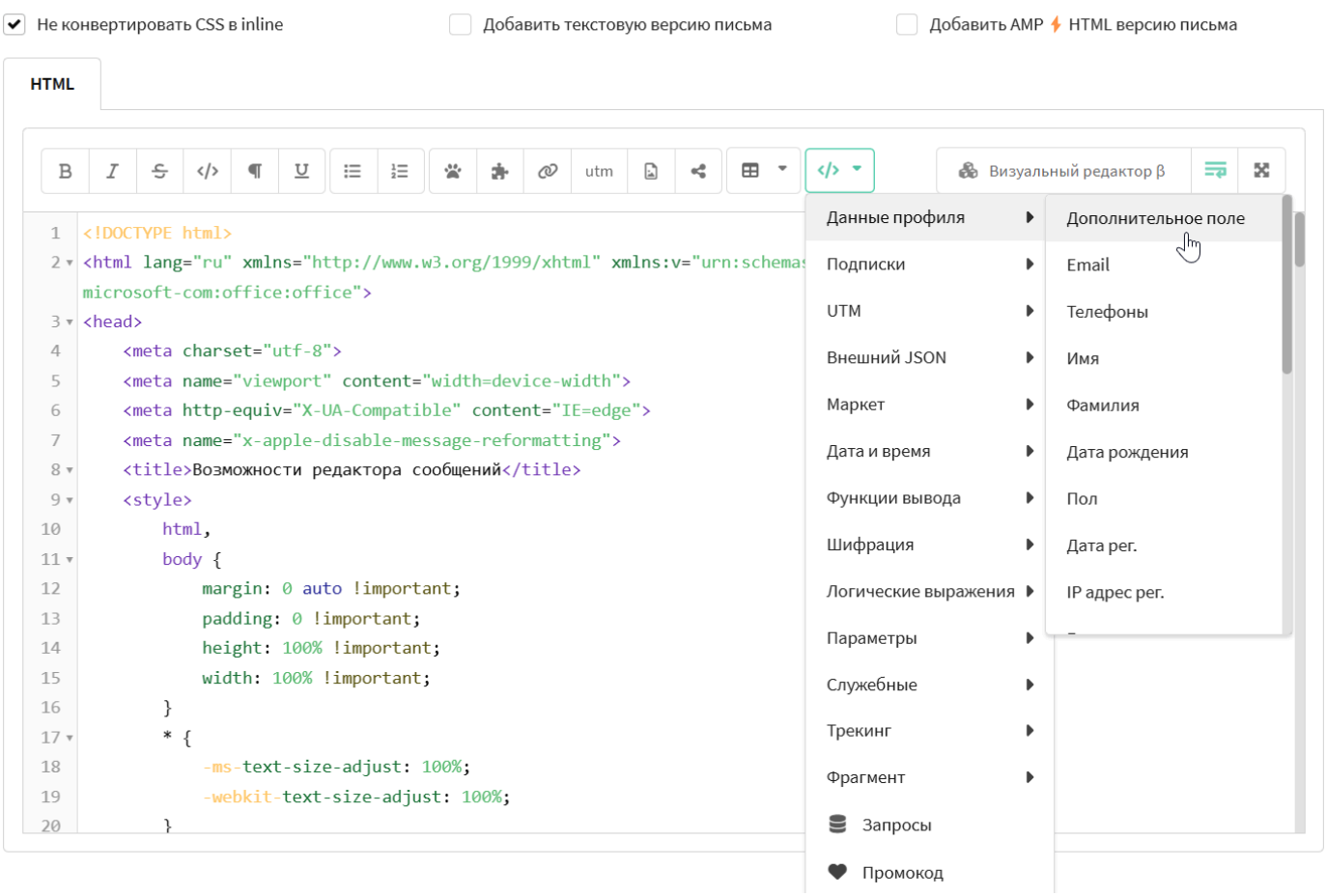


CAUTION

Поля **Имя отправителя** и **Email отправителя** переопределяют значения, настроенные для Email-канала в Ресурсе или — если ресурсы не используются — значения аналогичных полей в Базе профилей.

HTML-версия письма

Большинство подписчиков видят в своих почтовых ящиках HTML-версию сообщения. В редакторе кода поддерживаются HTML и CSS, а также переменные персонализации.



Для облегчения форматирования и работы с объектами доступны следующие функции:

Форматирование

B

— **жирный текст**

I

— *курсив*

~~S~~

— ~~зачёркнутый~~ текст

¶

— отформатированный текст

U

— подчеркнутый текст

— маркированный список

— нумерованный список

— вставка таблицы

Служебные функции

— добавление **пикселей** для отслеживания **открытия** и **прочтения** письма. Если вы не вставите пиксели в процессе редактирования сообщения, система предупредит вас об этом при сохранении.

— добавление **внешнего HTML-контента.** Вы также можете указать адрес источника переменной вручную `{[http://example.com]}`.

— добавление или редактирование ссылок.

— настройка **UTM меток**.

— добавление или редактирование **изображений**.

— добавление кнопок **Поделиться в социальных сетях.**

— меню **переменных и функций редактора**. С помощью этой кнопки вы можете добавить в сообщение данные профилей клиентов, промокоды, циклы и логические выражения, а также динамический контент и системные ссылки.

TIP

С помощью ряда кнопок в правой части окна вы можете включить режим **переноса строк**

или **полноэкранный режим**

, а также перейти в визуальный редактор.

TIP

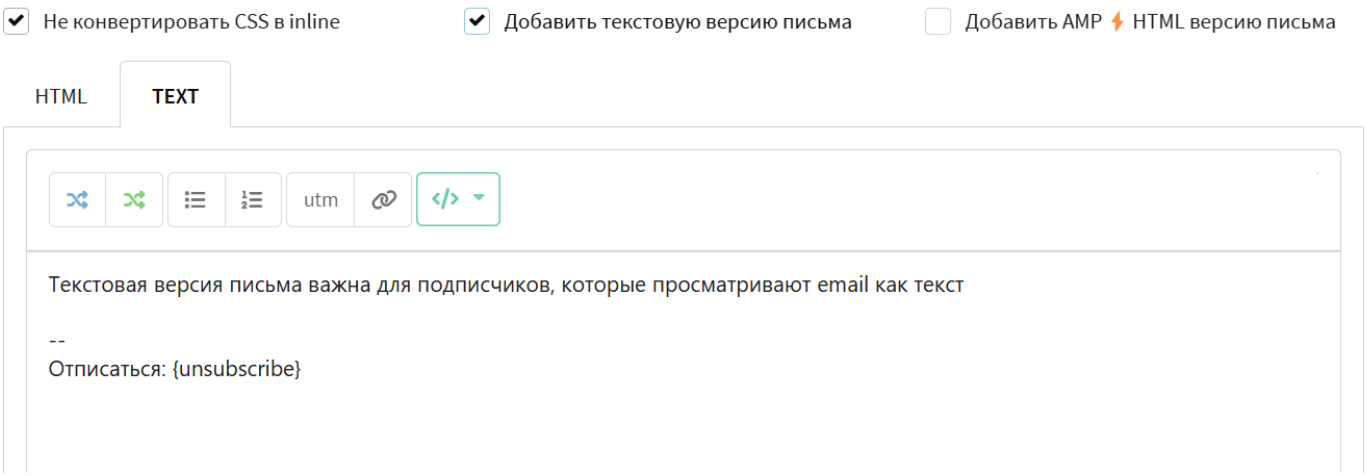
По умолчанию Altcraft не конвертирует стили, включённые в HTML-версию сообщения, в строчные теги **(Не конвертировать CSS в inline).** Однако, если эту опцию убрать, стили будут добавляться атрибутами в элементы сообщения:

```
<h1 style="font-size:36px;">Привет!</h1>
<h2 style="font-size:28px;">Хороший день, не правда ли?</h2>
```

Это обеспечивает большую совместимость с Outlook и прочими клиентами почты, которые не поддерживают стили, вынесенные в блок `<style>`.

Текстовая версия письма

Текстовая версия письма представляет собой текст без форматирования. Она отображается в почте вместо HTML, если email-провайдер подписчика или сам клиент блокирует HTML-версию. По умолчанию данная версия отключена. Чтобы включить её в письмо, отметьте галочкой **“Добавить текстовую версию письма”**.



Вы можете сгенерировать текстовую версию на основе HTML с помощью кнопок в парвом верхнем углу редактора:

— генерирует текстовую версию с гиперссылками.

— генерирует текстовую версию с использованием Markdown-ссылок, что может повысить совместимость с различными почтовыми системами.

Текстовая версия также поддерживает списки, UTM метки, ссылки и переменные.

CAUTION

Если в HTML-версии письма есть ссылка отписки или confirm-ссылка, она также должна присутствовать в текстовой версии письма. В противном случае, вы не сможете сохранить шаблон.

AMP-версия

Технология AMP позволяет добавлять JavaScript-компоненты в email-письма и делать их интерактивными. В AMP-письме пользователь может пройти опрос, купить товар или подтвердить регистрацию, не переходя на

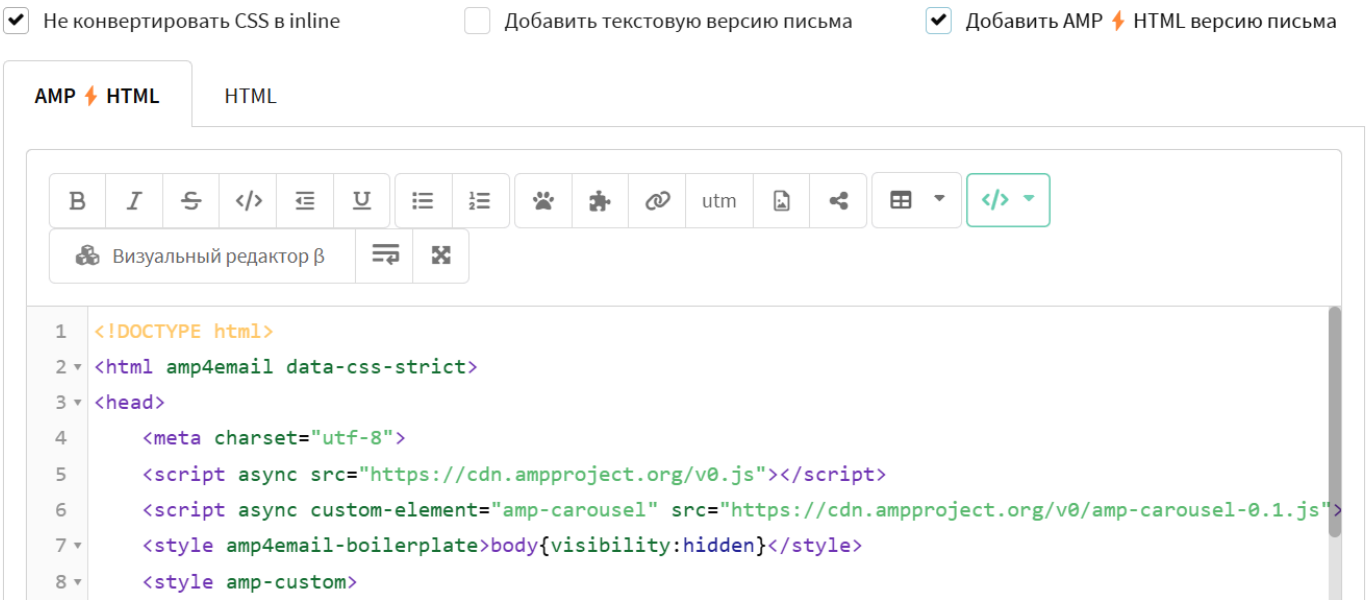
сайт.

Почтовые клиенты требуют специальной регистрации от отправителей AMP-писем. Руководство по добавлению вашего адреса отправителя в разрешённый список вы найдёте в официальной документации AMP.

TIP

Тем, кто только начинает осваивать amp-технологию, будет полезен этот гайд.

Для отправки AMP-писем необходимо установить флажок в поле **“Добавить AMP⚡HTML версию письма”**. Ниже появится дополнительная вкладка **“AMP⚡HTML”**, где вы сможете использовать AMP-компоненты.



TIP

На данный момент AMP-письма поддерживают следующие почтовые клиенты:

- Gmail;
- Yahoo;
- Mail.ru.

Если письмо придёт на почтовый клиент, который не поддерживает технологию AMP, получатель увидит HTML-версию.

Ссылки в AMP-версии

В amp-версии письма для наименования ссылок поддерживается только атрибут `data-name`. При сохранении шаблона платформа автоматически подставит атрибут во все ссылки.

```
<a href="https://example.com" title="Текст ссылки" data-name="link1">Текст ссылки</a>
```

Также обратите внимание, если вы добавляете amp-параметр в ссылку, не надо оборачивать ссылку трекингом, иначе она не будет работать. Чтобы отключить трекинг, добавьте атрибут `no-track`. Клики по такой ссылке отслеживаться не будут.

```
<a href="${{exampleUrl}}" title="Текст ссылки" data-name="link1" no-track>Текст ссылки</a>
```

Экранирование шаблонных выражений

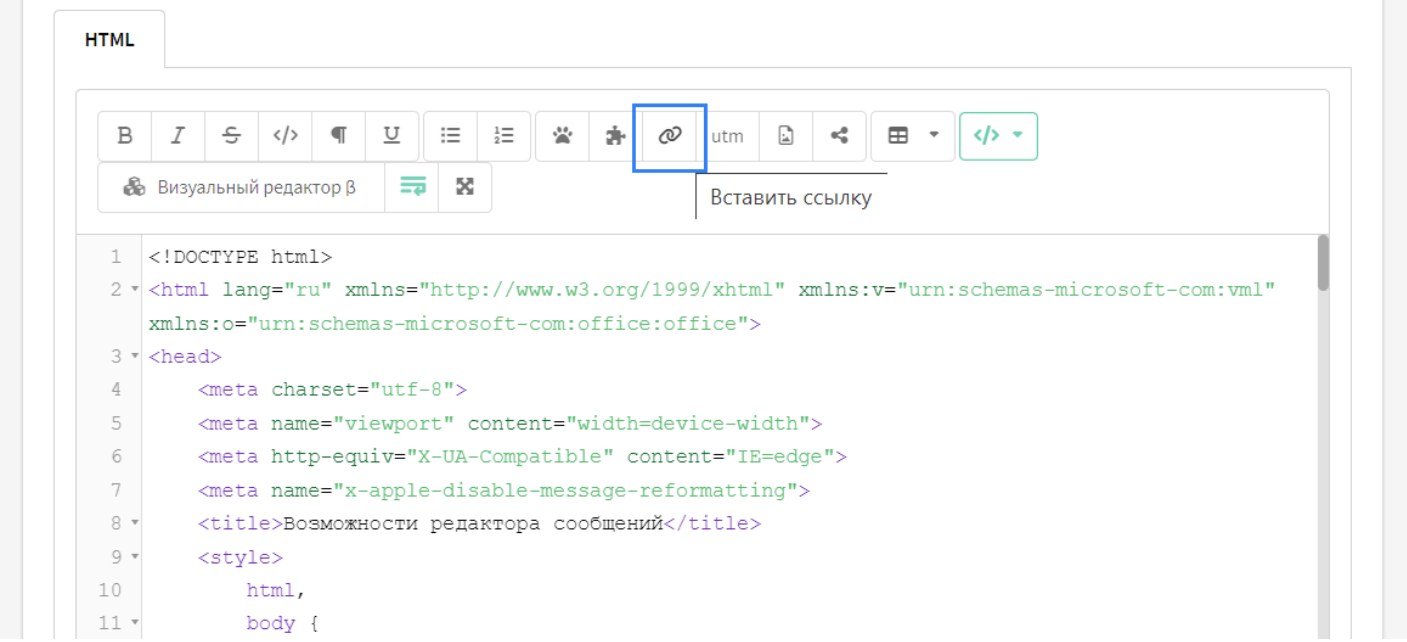
Синтаксис шаблонизации AMP совпадает с синтаксисом шаблонизации Altcraft. Чтобы избежать некорретной обработки кода, необходимо экранировать шаблонные выражения AMP с помощью символа **$**.

Например, если вы используете компонент `<amp-list>`, то замените переменную для подстановки элемента списка `{{title}}` на `${{title}}`:

```
<amp-list id="myList" src="https://foo.com/list.json">
  <template type="amp-mustache">
    <div>${{title}}</div>
  </template>
</amp-list>
```

Конструктор ссылок

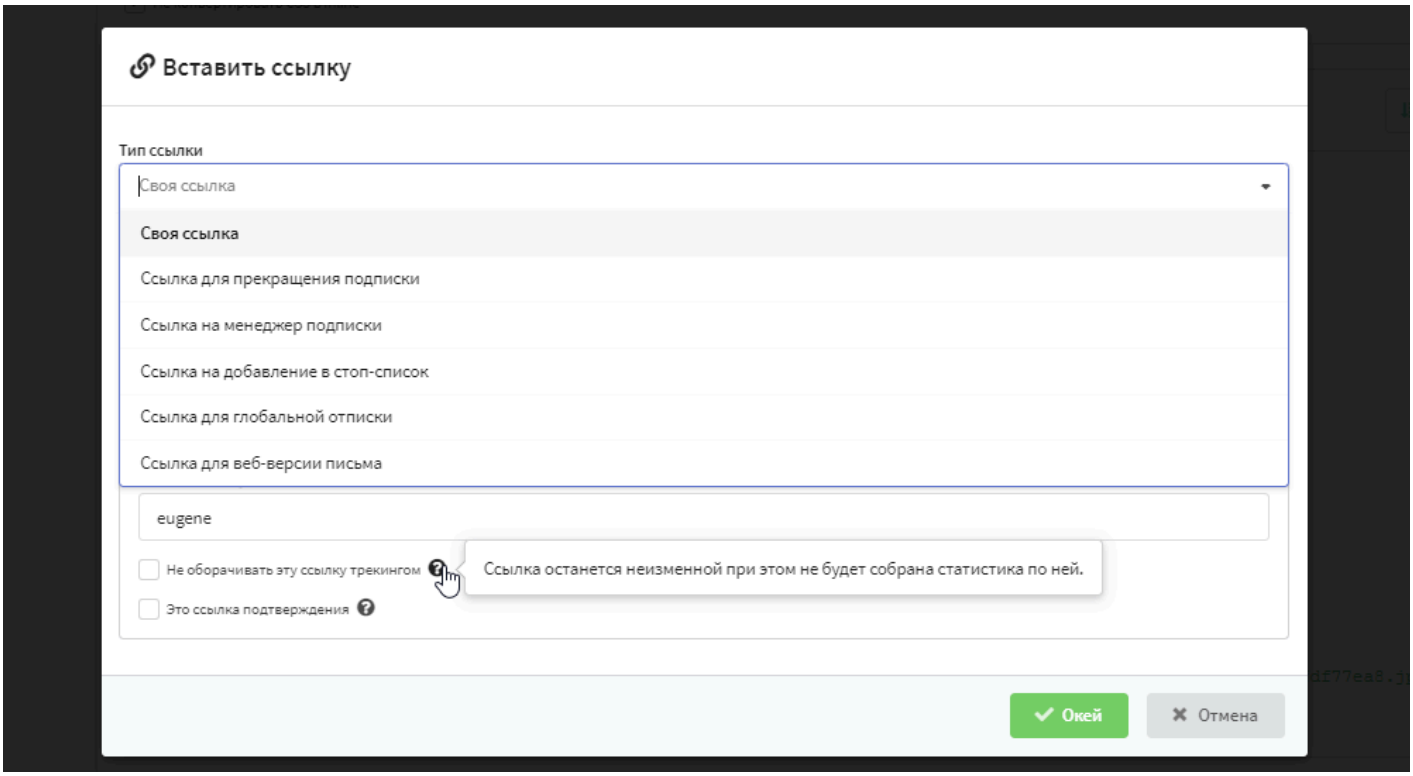
В редактор писем встроен удобный конструктор ссылок. Чтобы добавить ссылку, установите курсор в нужную строку и нажмите на кнопку **Вставить ссылку**. Если вы хотите изменить ссылку, которую уже добавили в шаблон, выделите её в редакторе и после этого откройте конструктор:



В конструкторе можно выбрать **тип ссылки** в зависимости от назначения:

- **Своя ссылка** — ссылка на любую веб-страницу, кроме пользовательских ссылок отписки и подтверждения.

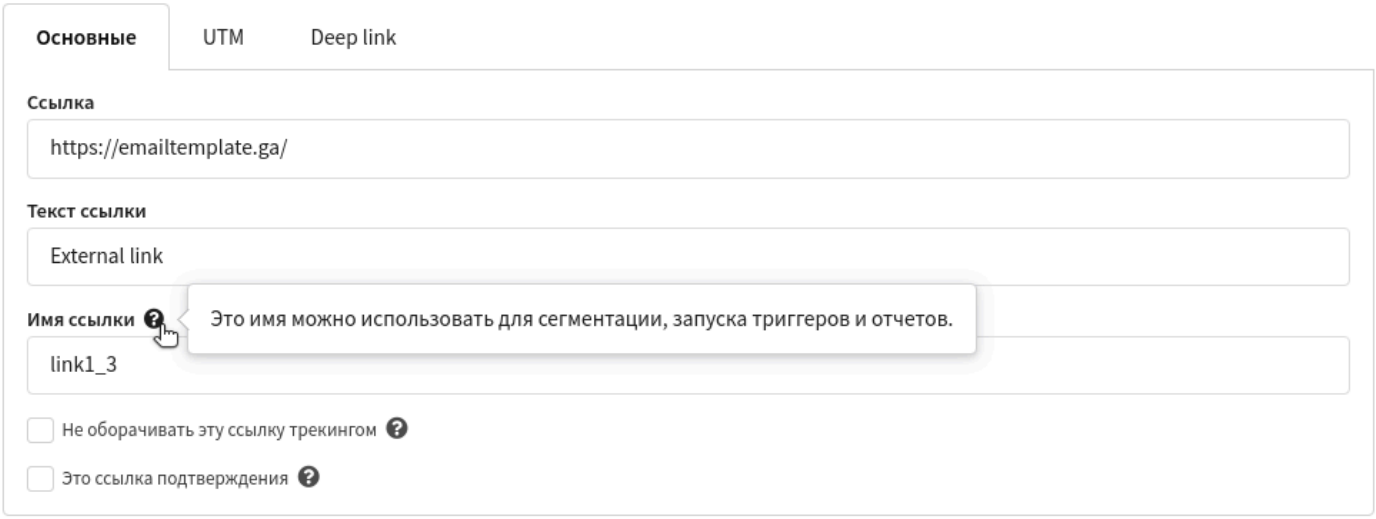
- **Ссылка прекращения подписки** — ссылка на системную или пользовательскую страницу прекращения подписки на ресурс. Подписка переходит в статус **Отписан**. Не влияет на глобальный статус профиля и другие подписки.
- **Ссылка на менеджер подписок** — ссылка на системную страницу управления подписками. Клиент сможет приостановить свои подписки, возобновить или отписаться от рассылок.
- **Ссылка на добавление в стоп-список** — ссылка на добавление в стоп-список, по которой клиент сможет заблокировать получение определённого контента.
- **Ссылка для глобальной отписки** — изменяет глобальный статус профиля клиента на **Отписан**, полностью прекращая возможность коммуникации.
- **Ссылка для веб-версии письма** — ссылка на точную копию письма, отправленного подписчику. Если у клиента возникнут проблемы с отображением письма — оно будет доступно по ссылке из любого браузера.



Основные настройки

В **основных настройках** введите URL для перехода — если вы добавляете свою ссылку. Для системных ссылок поле будет предзаполнено. Также введите **текст ссылки** для отображения в письме и **имя ссылки** — оно используется в отчётах, сегментации и при запуске триггеров и сценариев.

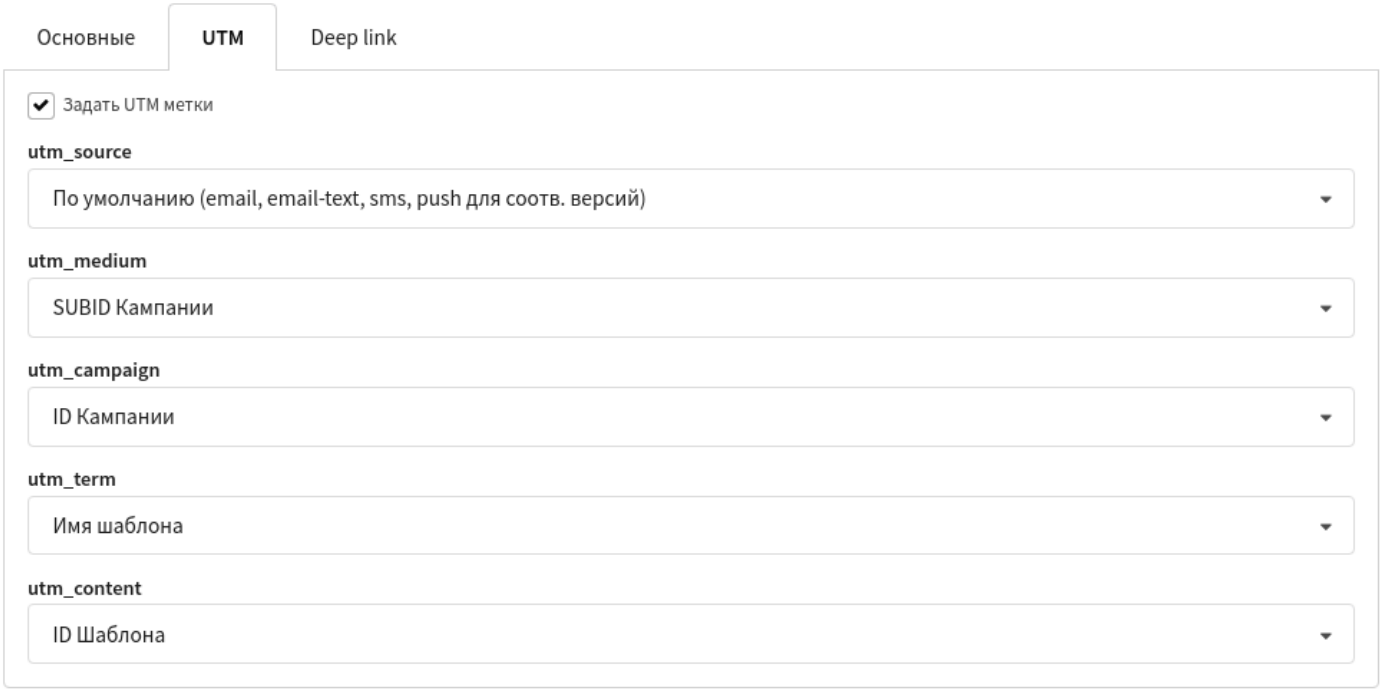
Дополнительно вы можете **отключить** перенаправление на **трекинг** при переходе по ссылке, а также сделать любую ссылку ссылкой подтверждения подписки.



UTM метки

Для каждой **отдельной** ссылки в сообщении вы можете переопределить **UTM метки**. Для этого необходимо перейти на вкладку **UTM** и активировать поле **Задать UTM метки**.

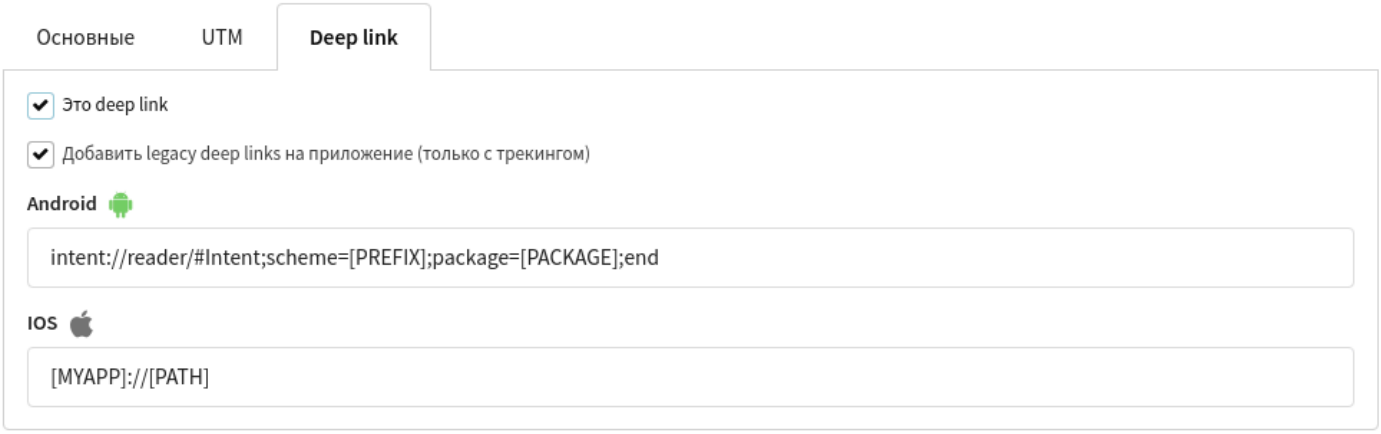
В качестве меток могут использоваться переменные, такие как название рассылки или базы данных.



Deep link

Если вы хотите использовать в email-письме ссылку для перехода в мобильное приложение, активируйте опцию **Это deep link** на соответствующей вкладке.

Чтобы указать различные ссылки для Android и iOS платформ, добавьте **legacy deep links**. Обратите внимание, что такие ссылки могут быть добавлены только с использованием **трекинга**.



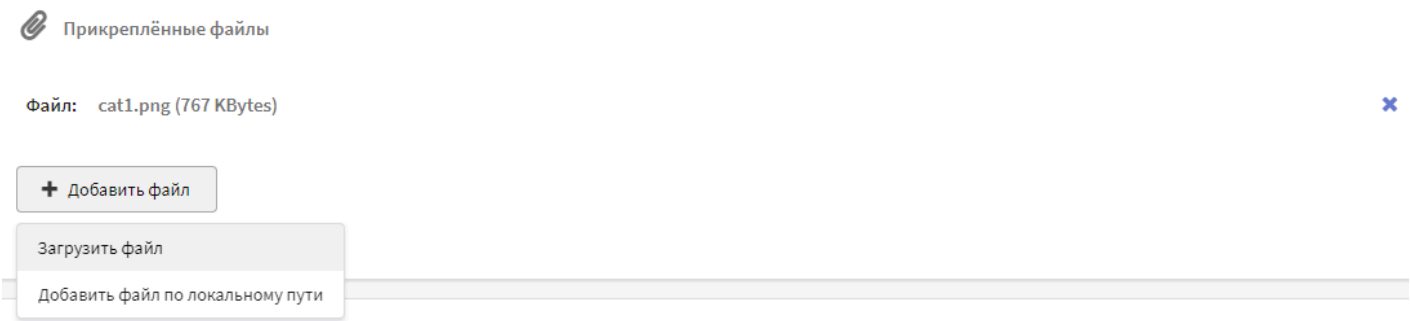
Вложения в Email

INFO

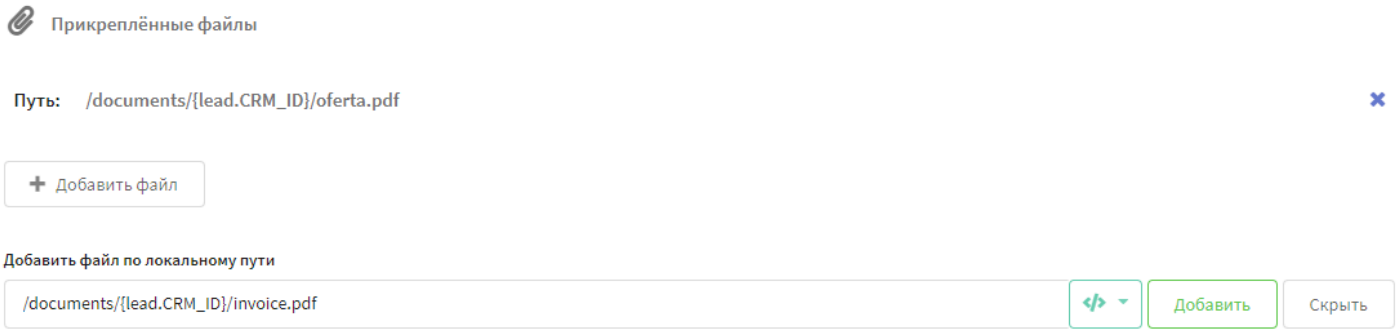
Вы можете вложить в письмо файл любого формата, но есть ограничение на максимальный размер файла — 12 МБ. Это значение можно изменить в параметре конфигурации `MTA_ATTACH_MAX_SIZE`.

Чтобы прикрепить к письму файл, воспользуйтесь формой в нижней части редактора. Есть два варианта:

1. Загрузить файл в платформу с компьютера:



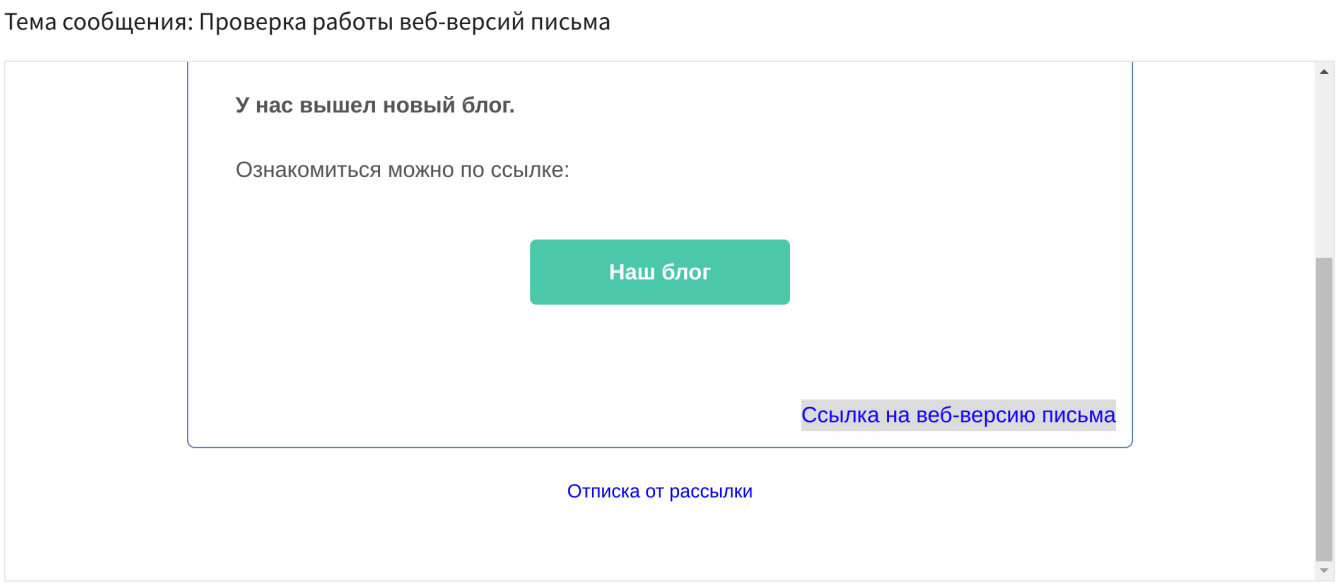
2. Выбрать директорию хранения **файла на сервере**, на котором установлена платформа. В пути директории вы можете использовать переменные профиля клиента. Введите расположение файла и нажмите **Добавить**:



Удалить загруженные файлы и сохранённые пути файлов вы можете, нажав на крестик в правой части формы.

Веб-версия письма

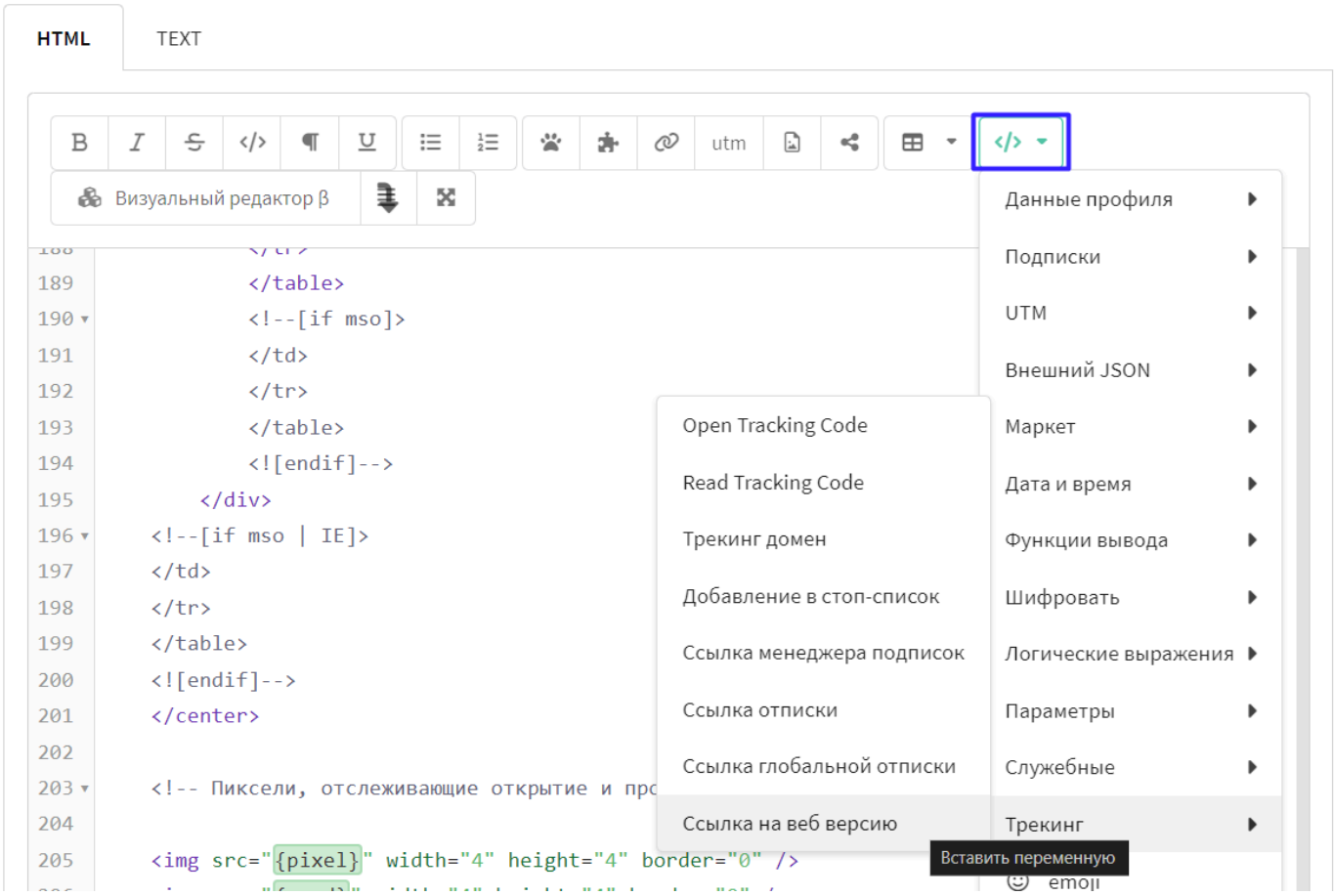
Веб-версия письма — это версия email-сообщения, которую пользователь может просмотреть как веб-страницу в браузере в случае его некорректного отображения в почтовых сервисах. Веб-версии увеличивают шансы того, что ваше сообщение будет просмотрено даже при неожиданных ошибках почтового клиента. Также наличие ссылки на веб-версию позволяет быстро и удобно поделиться письмом при необходимости.



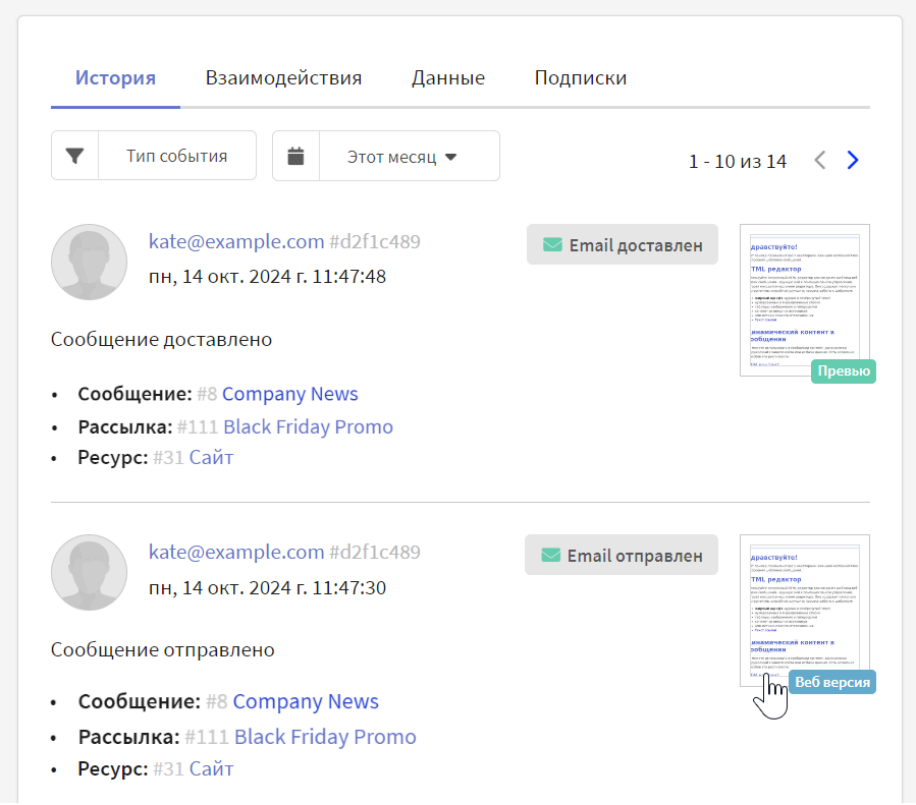
Добавить ссылку на веб-версию можно через конструктор ссылок или вручную:

```
<a href="{webversion}" title="Веб-версия письма" name="web_version"> Открыть письмо в браузере</a>
```

Также добавление ссылки возможно через меню переменных. Откройте его, кликнув **</>** на панели инструментов. Выберите **Трекинг — Ссылка на веб версию:**



Посмотреть веб-версию отправленного клиенту письма можно в [истории профиля](). По умолчанию веб-версии хранится 30 дней. Изменить период хранения может администратор платформы в настройках аккаунта.



TIP

Ссылка на веб-версию не открывается в режиме предпросмотра и при отправке тестового письма. Веб-версия формируется только для отправленных сообщений. Чтобы проверить ссылку, отправьте письмо тестовому подписчику через броадкаст.

INFO

По умолчанию сохранение веб-версий отключено в аккаунте. Для активации этой функции администратору платформы необходимо настроить политику веб-версий. После этого в истории профилей появятся ссылки для просмотра веб-версий отправленных сообщений.

Если в шаблоне отправленного сообщения отсутствует ссылка на веб-версию, веб-версия письма не будет сохранена. Тем не менее, администратор может изменить это поведение, активировав опцию **Always save email web versions** в настройках аккаунта.

SMS-шаблон

В этой статье описывается, как создать шаблон сообщения для отправки через SMS-канал.

Тело сообщения

Введите текст короткого сообщения в **поле ввода**. В теле сообщения могут использоваться ссылки, UTM-метки и переменные:

</>

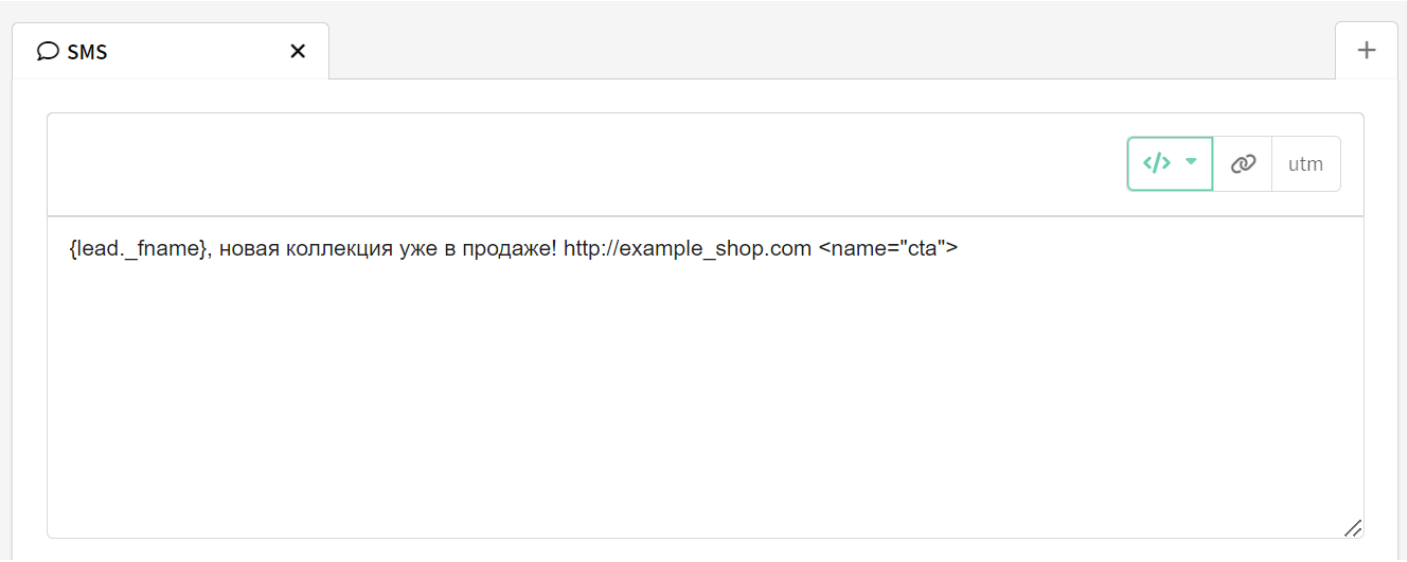
— меню переменных и функций редактора. С помощью этой кнопки вы можете добавить в сообщение данные профилей клиентов, промокоды, циклы и логические выражения, а также динамический контент и системные ссылки.

🔗

— редактор ссылок.

utm

— настройка UTM меток.

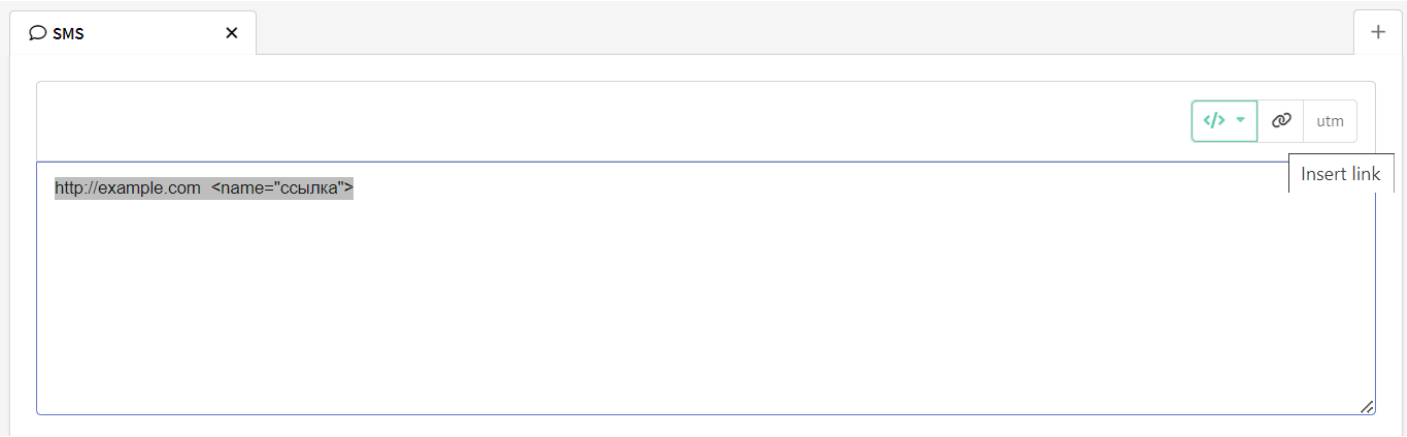


Редактор ссылок в SMS

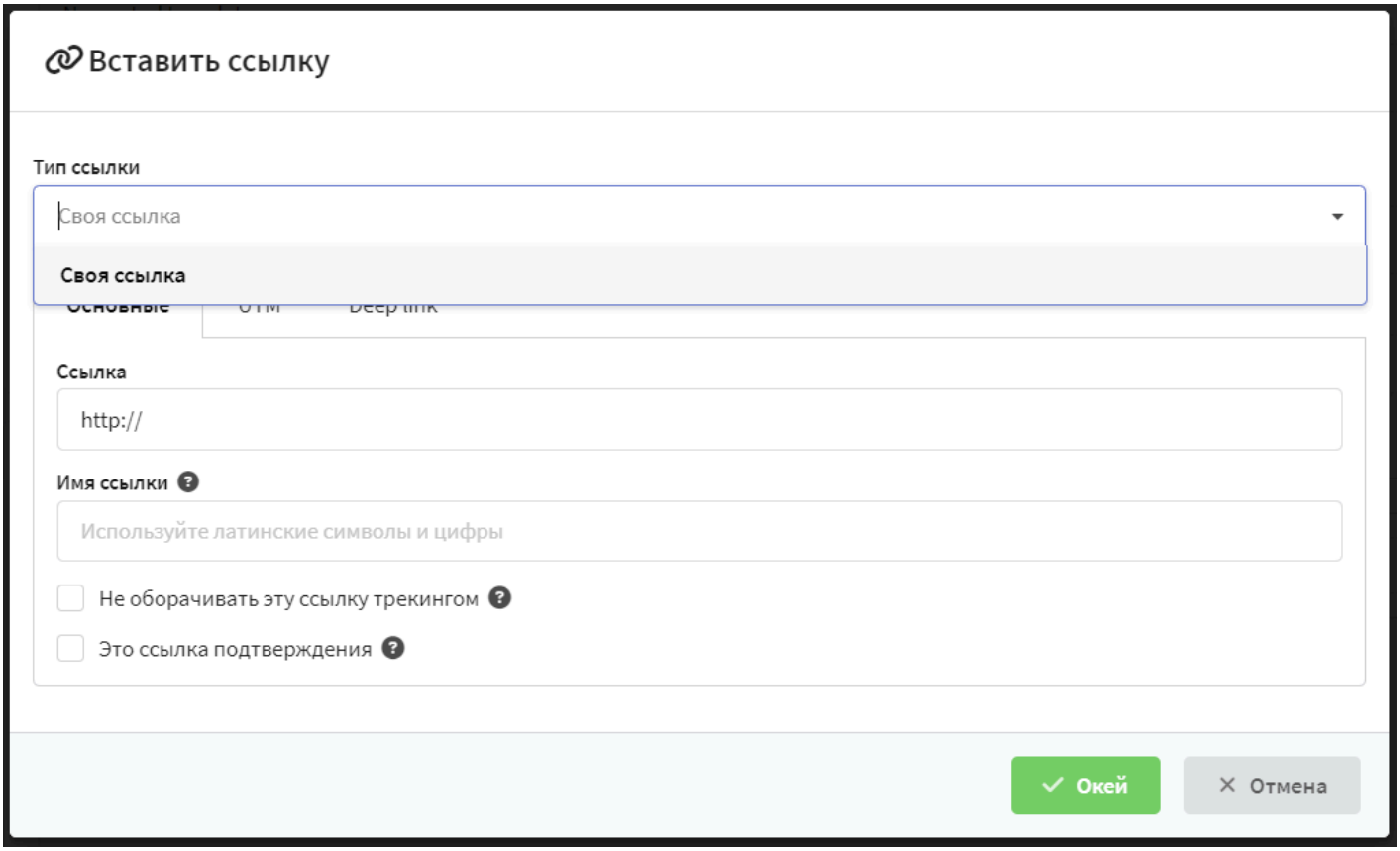
TIP

Ссылки, вставленные через платформу, автоматически сокращаются. Итоговая длина пути ссылки без учета трекинг-домена — 12-14 символов, включая `/`.

Редактор ссылок доступен по нажатию на кнопку **Вставить ссылку**. Если вы хотите редактировать ссылку, которую уже добавили в шаблон — выделите её в редакторе и после этого откройте форму:



Для SMS-сообщений доступен только один тип ссылок — **своя ссылка**. Он позволяет добавить ссылку на любую веб-страницу.

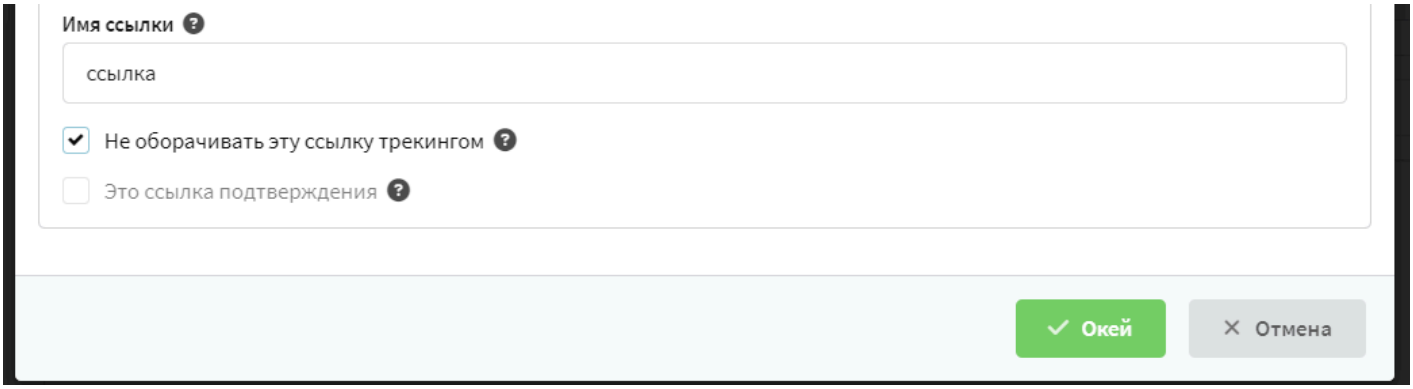


Основные настройки

В **основных настройках** введите URL для перехода. Также укажите **имя ссылки** — оно используется в отчётах, сегментации и при запуске триггеров и сценариев. Имя ссылки может состоять только из цифр и латинских букв.

> **CAUTION**
>
> Обратите внимание, когда вы отбираете профили по клику на ссылку в SMS, в услових отобра необходимо указать имя ссылки, а не URL-адрес.

Дополнительно вы можете **отключить** перенаправление на **трекинг** при переходе по ссылке, а также сделать любую ссылку ссылкой подтверждения подписки.



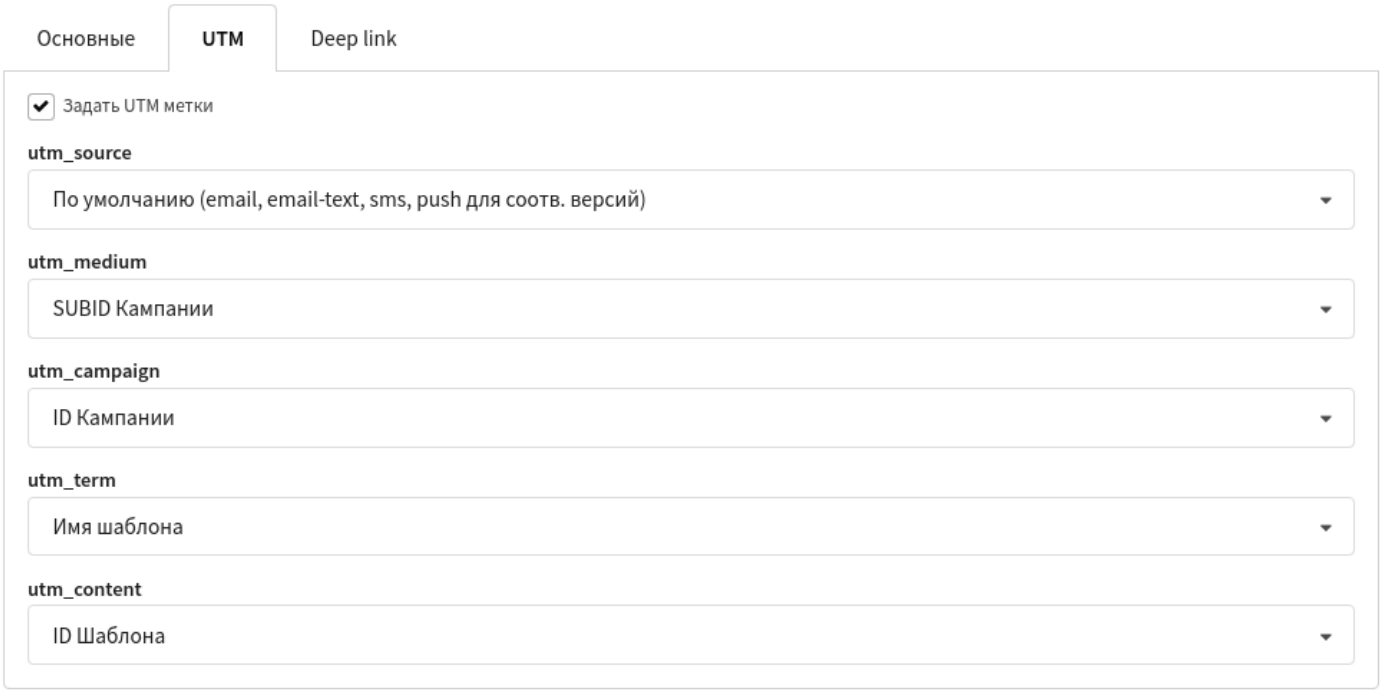
В SMS сообщениях угловые скобки используются для обозначения параметров (например, для параметра имени будет использоваться `<name="">`). Всё, что вы укажете в угловых скобках, не попадёт в итоговое сообщение. Примеры использования:

- `<no-track>` — отключает перенаправление на трекинг при переходе по ссылке.
- `<confirm-link>` — делает ссылку подтверждения.
- `<name="example_name">` — задает имя ссылки.

UTM-метки

Для каждой **отдельной** ссылки в сообщении вы можете переопределить **UTM-метки**. Для этого необходимо перейти на вкладку **UTM** и активировать поле **Задать UTM метки**.

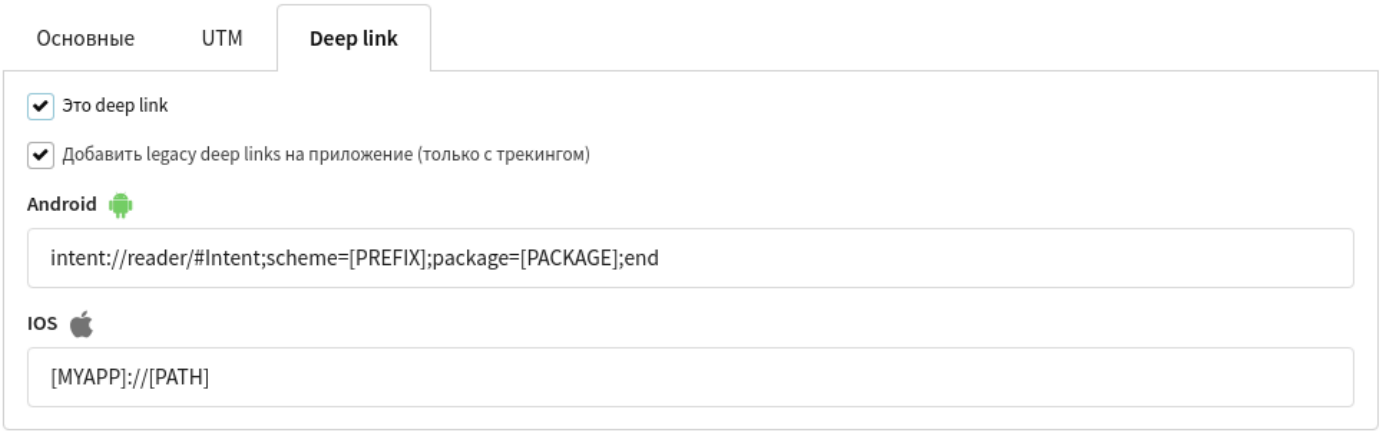
В качестве меток могут использоваться переменные, такие как название рассылки или базы данных.



Deep link

Если вы хотите использовать в SMS ссылку для перехода в мобильное приложение, активируйте опцию **Это deep link** на соответствующей вкладке.

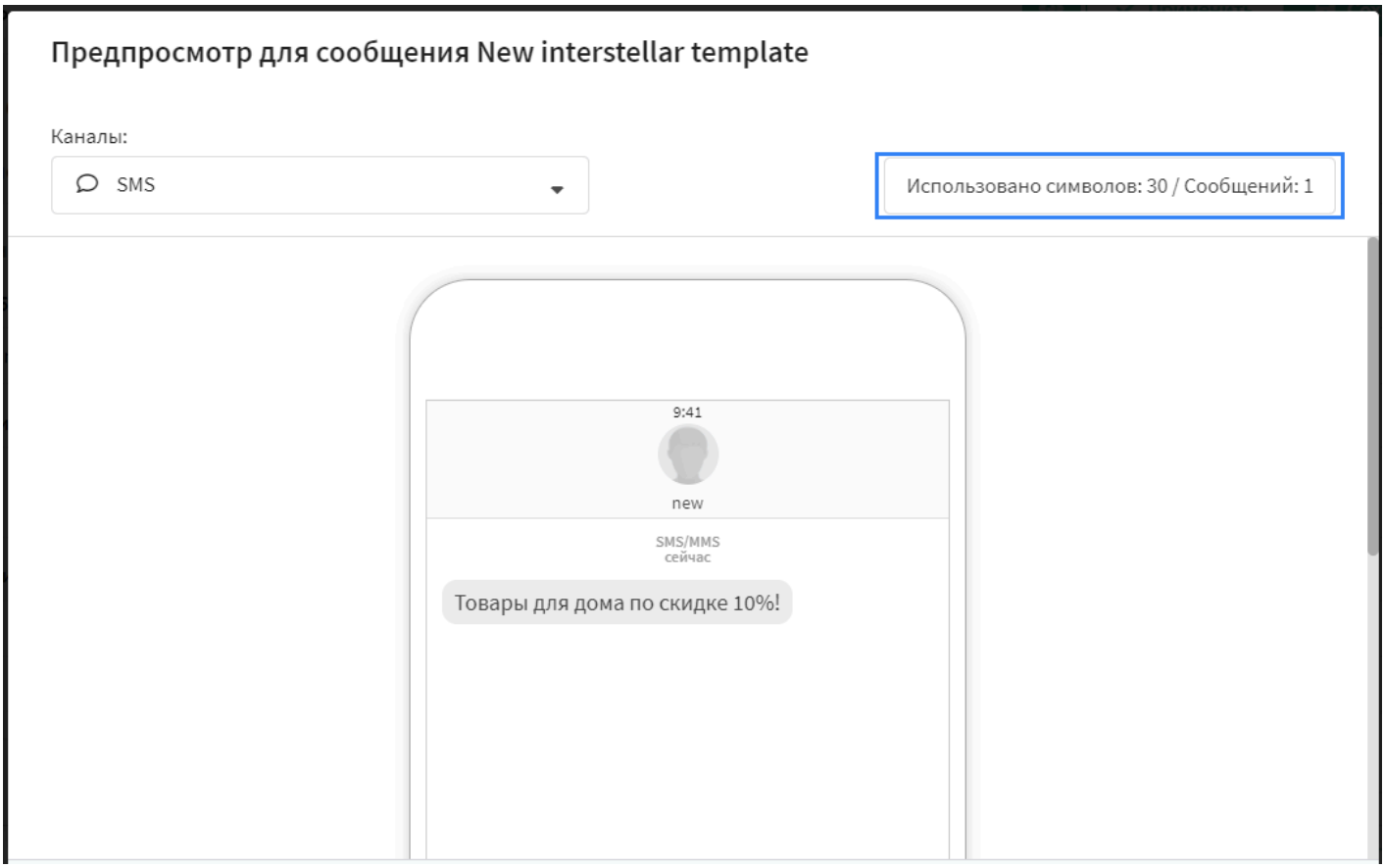
Чтобы указать различные ссылки для Android и iOS платформ, добавьте **legacy deep links**. Обратите внимание, что такие ссылки могут быть добавлены только с использованием **трекинга**.



Ограничение на количество символов

При отправке SMS-сообщений важно учитывать длину текста, так как количество символов влияет на стоимость отправки. Одно SMS-сообщение на кириллице может содержать **до 70 символов**, а на латинице — **до 160 символов**, включая пробелы и знаки препинания. Если вы превысите эти лимиты, ваше сообщение будет тарифицировано как два SMS. Наличие хотя бы одного кириллического символа делает всё сообщение кириллическим.

Для удобства редактор автоматически подсчитывает количество символов и сообщений. Чтобы увидеть счетчик, сохраните шаблон и перейдите в режим предпросмотра:



TIP

Обратите внимание, данные, которые подтягиваются из динамических переменных, могут превысить ожидаемое количество символов.

Кроме того, для сокращения ссылок в шаблон не добавляется протокол `http(s)://`, но вы можете включить автоподстановку. Это касается только трекинговых ссылок.

Количество отправляемых SMS можно ограничить, указав значение в поле “Лимит на количество SMS”. Если текст сообщения после разбиения превышает указанный лимит SMS-частей, сообщение не будет отправлено. Например, если пользователь установил лимит в 3 SMS, а длина текста составляет 250 символов, при рассылке на одна часть такого сообщения отправлено не будет.

INFO: Обратите внимание

При использовании в шаблоне логических выражений или циклов платформа учитывает все символы форматирования, включая пробелы и переносы строк, как часть итогового сообщения. Каждый пробел или перенос строки добавляет к SMS один символ, что может повлиять на стоимость рассылки. Для экономии символов и более точного контроля над длиной сообщения записывайте логические конструкции в одну строку.

Например, вместо:

```
{if lead._city equal "Москва"}
Приглашаем на конференцию в Москве 30 июня!
{else}
Приглашаем на конференцию в вашем городе!
{end}
```

Используйте: ```text {if lead._city equal "Москва"} Приглашаем на конференцию в Москве 30 июня!{else} Приглашаем на конференцию в вашем городе!{end} ```

Такое форматирование не повлияет на логику работы, но сделает шаблон компактнее и поможет избежать неожиданного увеличения длины SMS.

Push-шаблон

Платформа поддерживает два редактора для push-шаблонов: **стандартный редактор** и **специальный редактор для Rich push**. В Rich push редакторе доступно добавление баннеров и кнопок, а также есть возможность менять настройки пуша в зависимости от браузера или ОС.

Стандартный редактор

Для push-уведомлений настраиваются:

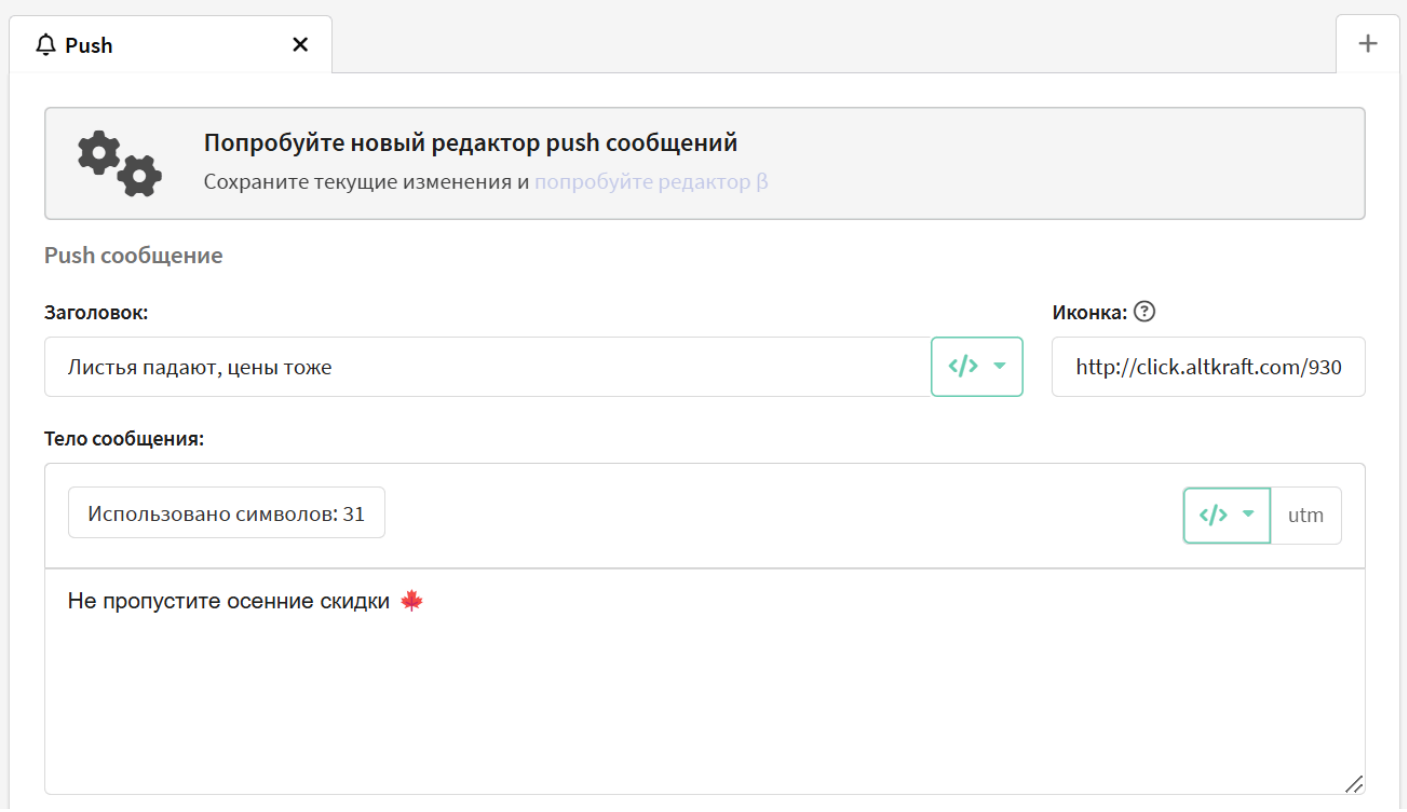
- **Заголовок** — рекомендуется использовать не более 25 символов, чтобы текст не обрезался на устройствах. Поддерживаются эмодзи. Заголовок можно персонализировать, используя меню переменных

  </>

  .
- **Иконка** — изображение для отображения в уведомлении. В браузере **Apple Safari** поддерживается только одно изображение для всех уведомлений, его настройка выполняется в настройке push-канала в ресурсе.
- **Тело сообщения** — основной текст уведомления. Поддерживаются эмодзи. Максимальная длина текста — 256 символов, рекомендуется использовать до 150 символов. В теле сообщения также можно использовать переменные для персонализации

  </>

  . Редактор автоматически подсчитывает количество символов:

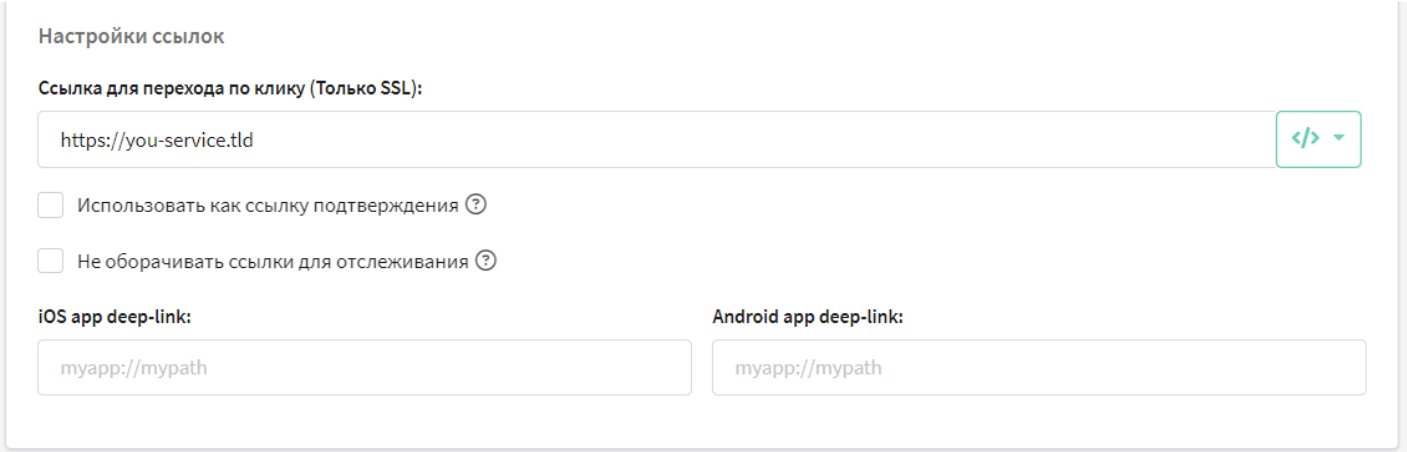


TIP

Ссылки, вставленные через платформу, автоматически сокращаются. Итоговая длина пути ссылки без учета трекинг-домена — 12-14 символов, включая `/`.

В блоке **Настройка ссылок** доступны:

- **Ссылка для перехода по клику** — URL-адрес, который открывается при клике на уведомление. Используется только защищенный протокол (SSL). Доступно меню переменных для вставки, например, промокодов, чтобы пользователь сразу перешел на страницу с предложением и рассчитанной скидкой.
- **iOS app deep-link** — ссылка для перехода в приложения на iOS.
- **Android app deep-link** — ссылка для перехода в приложения на Android.



Для добавления в ссылку **utm-меток** нажмите на кнопку

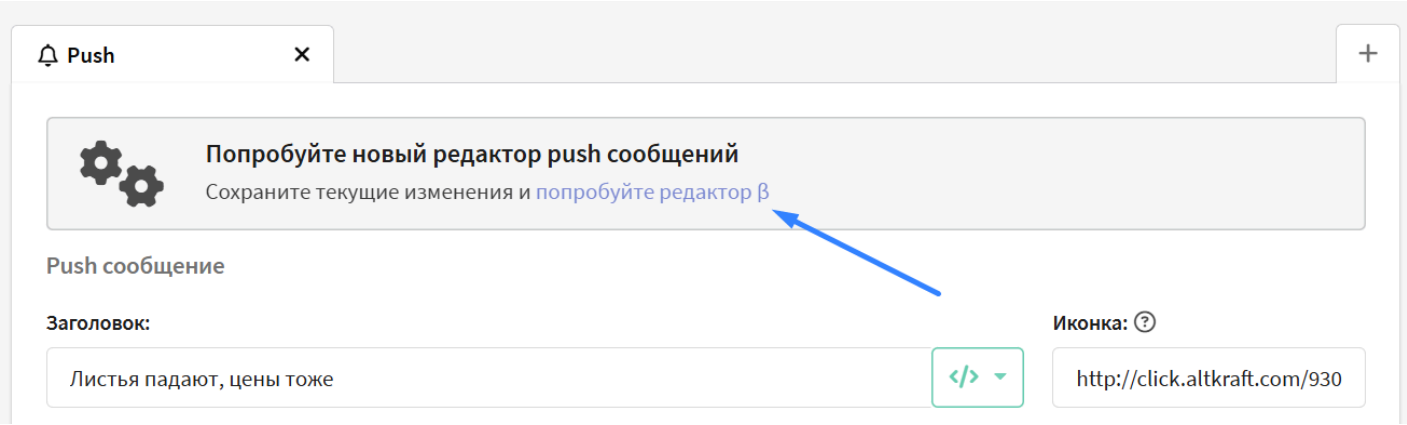
в правом верхнему углу редактора или задайте метки вручную.

Дополнительно ссылку в пуше можно **сделать ссылкой подтверждения** для реализации механизма двухэтапной подписки, а также **отключить трекинг**. Если вы решите не оборачивать ссылки трекингом для отслеживания, имейте в виду, что в этом случае платформа не сможет фиксировать клики по ссылке, и deeplink-ссылки станут недоступными.

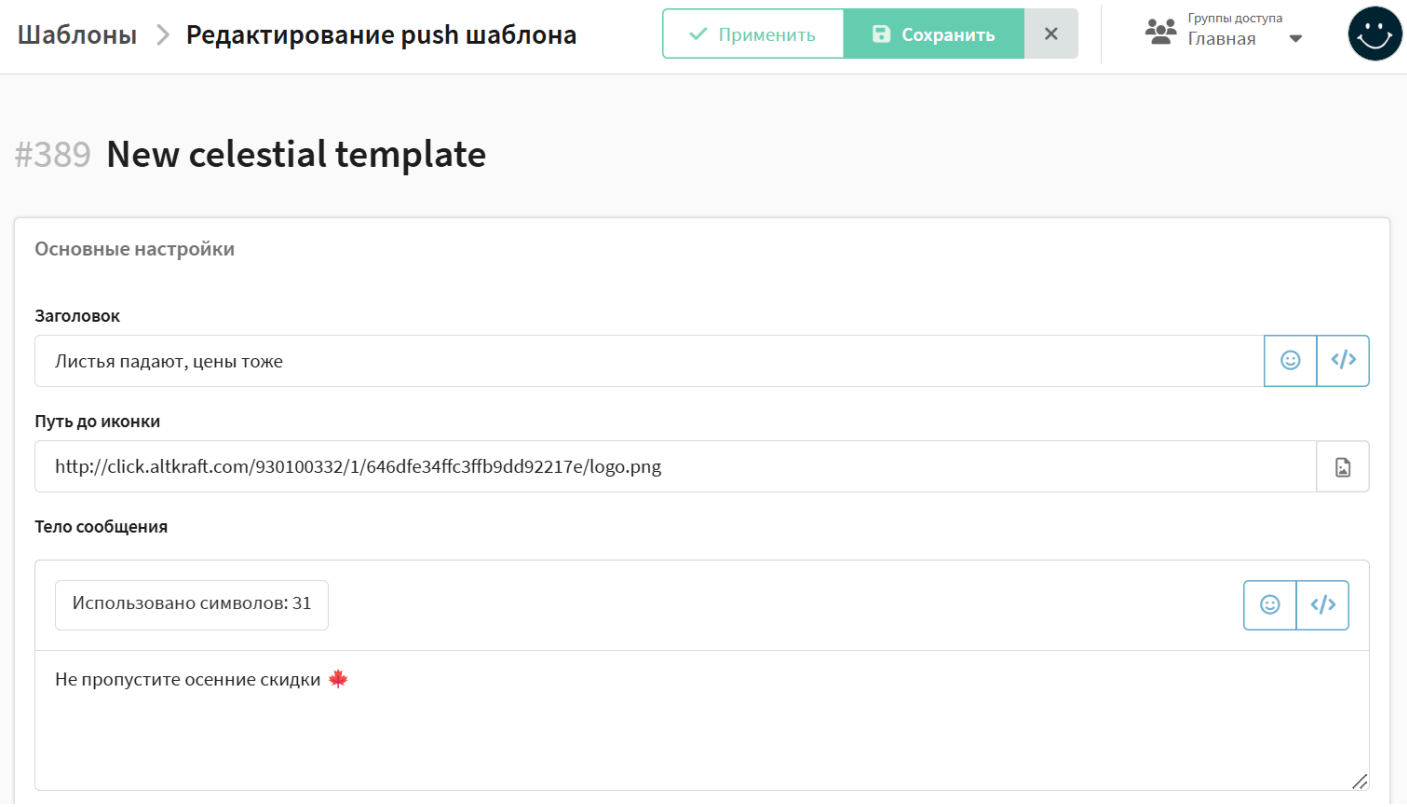
Rich Push редактор

Начало работы

Чтобы перейти в редактор для Rich push уведомлений, вам необходимо подготовить начальную версию сообщения в стандартном редакторе (можно без ссылки для перехода) и сохранить шаблон с помощью кнопки “Применить”. После этого в редакторе станет доступна кнопка **“попробуйте редактор β”**:



Заголовок, путь до иконки и тело сообщения будут заполнены данным, которые вы указали в стандартном push-редакторе. Вы можете отредактировать эти поля, если возникла такая необходимость.

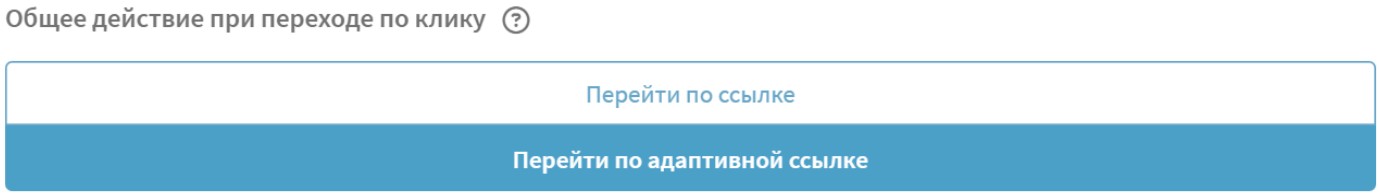


Ссылка для перехода

Действие при переходе по клику — это ссылка, по которой перейдет пользователь, когда кликнет на push-уведомление. В этом блоке вы устанавливаете общую ссылку. Пользователи будут переходить по этой ссылке, если не задана уникальная ссылка для различных браузеров и ОС. Уникальная ссылка задаётся ниже — в блоке “Расширенные настройки”.

В Rich push редакторе шаблонов можно настроить 2 типа действия:

- **Перейти по ссылке**
- **Перейти по адаптивной ссылке**

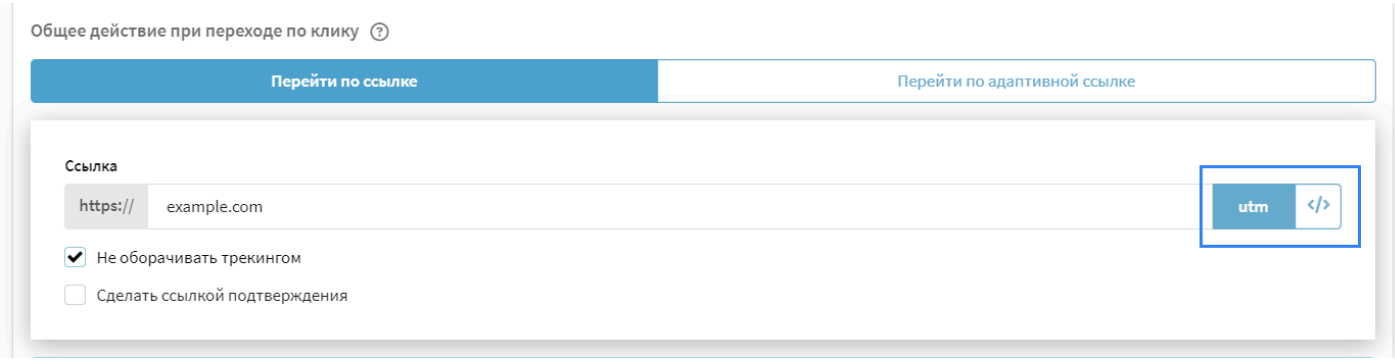


Перейти по ссылке

Укажите URL-адрес для перехода. Использовать можно только https ссылки, т.е. ссылка должна вести на сайт, который работает на безопасном SSL-соединении.

Дополнительные настройки:

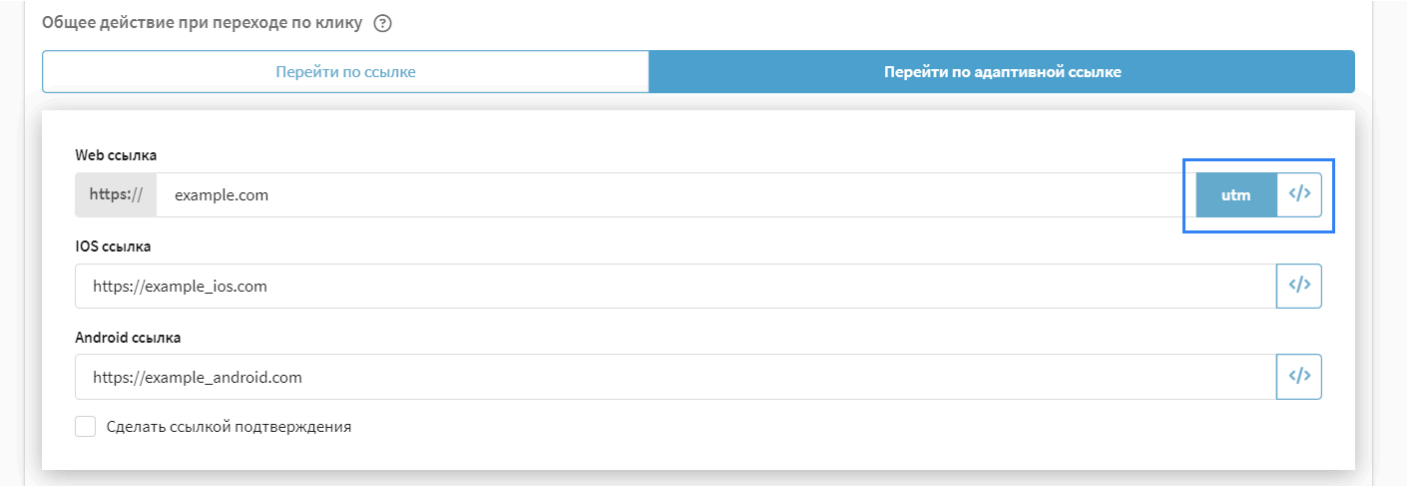
- **Персонализировать ссылку**, вставив переменные с помощью кнопки `</>`.
- **Добавить UTM метки** для отслеживания трафика подписчиков. Для этого нажмите “utm” и в открывшемся окне настройте метки, а затем нажмите “Задать utm”.
- **Отключить перенаправление на трекинг при переходе по ссылке.** Для этого активируйте поле “Не оборачивать трекингом”. Обратите внимание, что статистика кликов в этом случае собираться не будет.
- **Сделать ссылкой подтверждения.** Шаблон с такой ссылкой можно будет использовать для подтверждения подписки. При переходе по ссылке статус подписки профиля обновится: `Не подтвержден` —> `Подписан`.



Перейти по адаптивной ссылке

Адаптивная ссылка — ссылка для переадресации пользователя на разные URL, в зависимости от того, с какого устройства он откроет push-уведомление (десктоп, Android или iOS).

- **Web ссылка** — пользователь перейдет по этой ссылке, если откроет push уведомление с десктопного устройства
- **IOS ссылка** — пользователь перейдет по этой ссылке, если откроет push уведомление с IOS устройства.
- **Android ссылка** — пользователь перейдет по этой ссылке, если откроет push уведомление с Android устройства.



Дополнительные настройки:

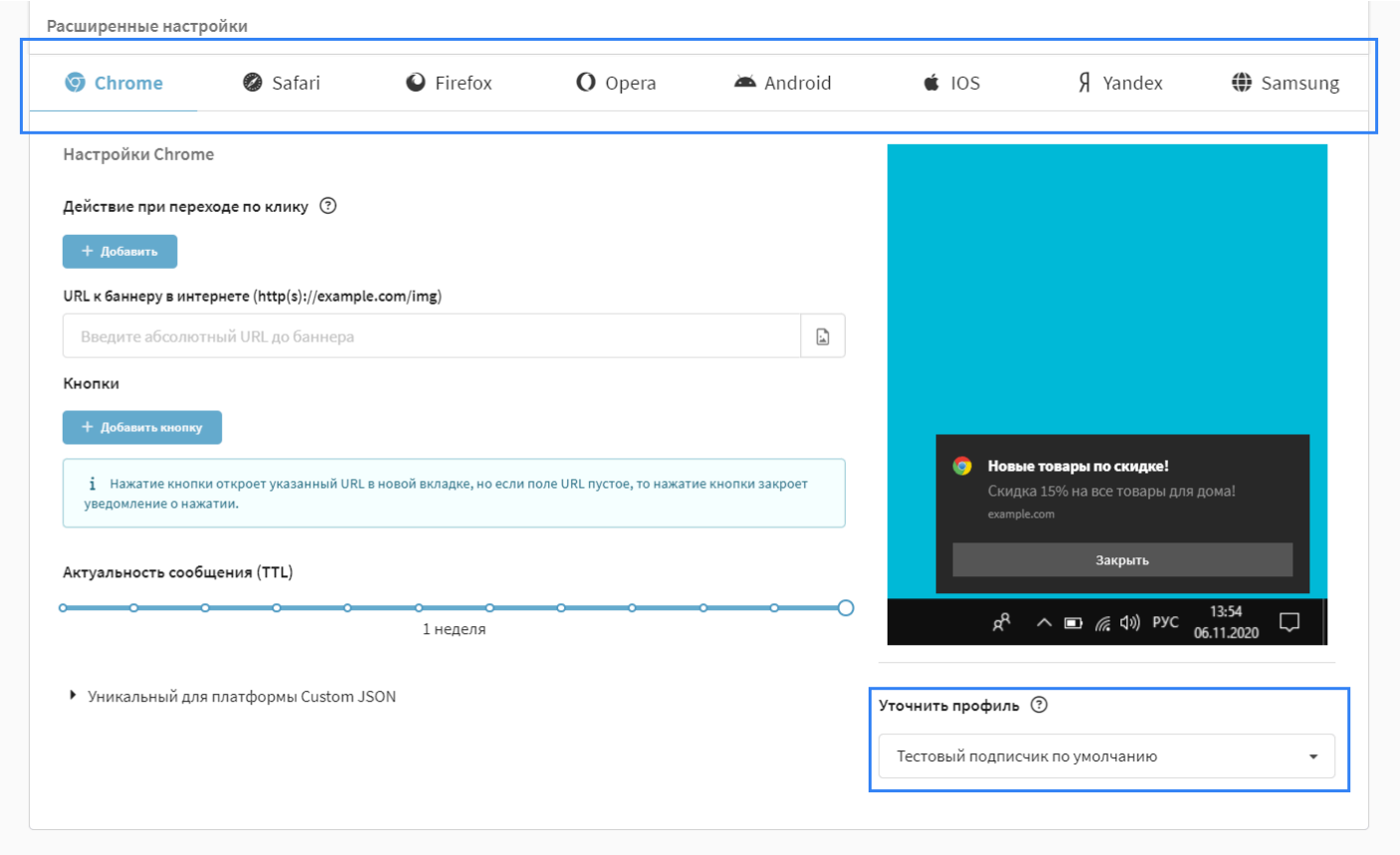
- **Добавить [переменные]** в ссылку с помощью кнопки `</>`.
- **Добавить UTM метки**. Для этого кликните кнопку “utm” и в открывшемся окне настройте метки, а затем нажмите Задать utm.
- **Сделать ссылкой подтверждения.** Шаблон с такой ссылкой можно будет использовать для подтверждения подписки. При переходе по ссылке статус подписки профиля обновится: `Не подтвержден` —> `Подписан`.

Расширенные настройки

Расширенная версия редактора push-уведомлений позволяет создавать push-сообщения **для разных вариантов ОС и браузеров**. Переключайтесь между ОС и браузерами в верхней панели и настраивайте push-сообщение для каждого из них.

Слева доступен **предпросмотр** push-сообщений в режиме реального времени.

Вы также можете **указать тестового подписчика**, чтобы посмотреть, как именно созданное push-сообщение будет отображаться на его устройстве.



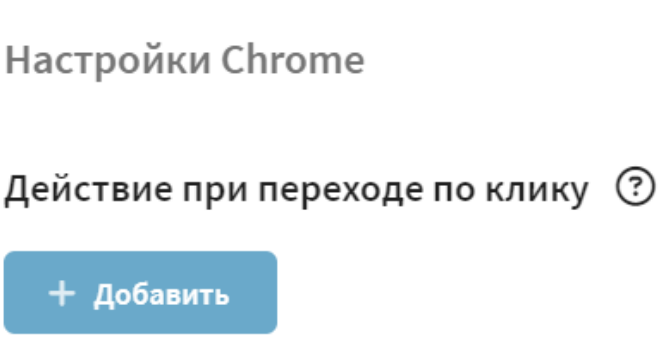
Индивидуальная настройка пуша доступна для 6 браузеров и 2 ОС. Большинство настроек совпадают.

Действие при переходе по ссылке

Для каждого браузера и ОС можно добавить **ссылку для перехода по клику**. Напоминаем, что именно по этой ссылки перейдет пользователь, когда кликнет на сообщение.

Как настроить ссылку для перехода по клику?

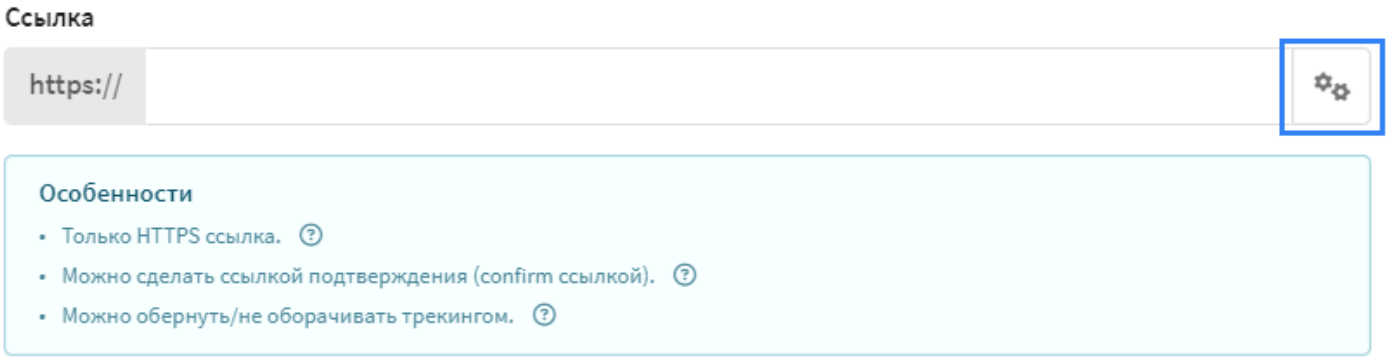
1. Нажмите **Добавить**:



2. Выберите **тип ссылки перехода**.

- **Обычная ссылка**
- **Адаптивная ссылка**
- **Deep-link ссылка**

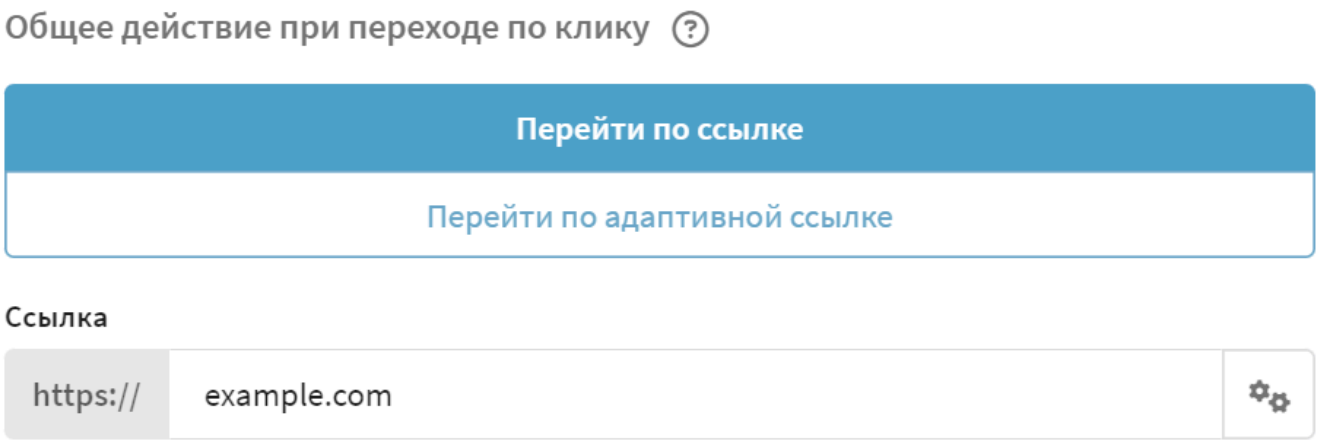
3. Введите адрес ссылки и настройте её.



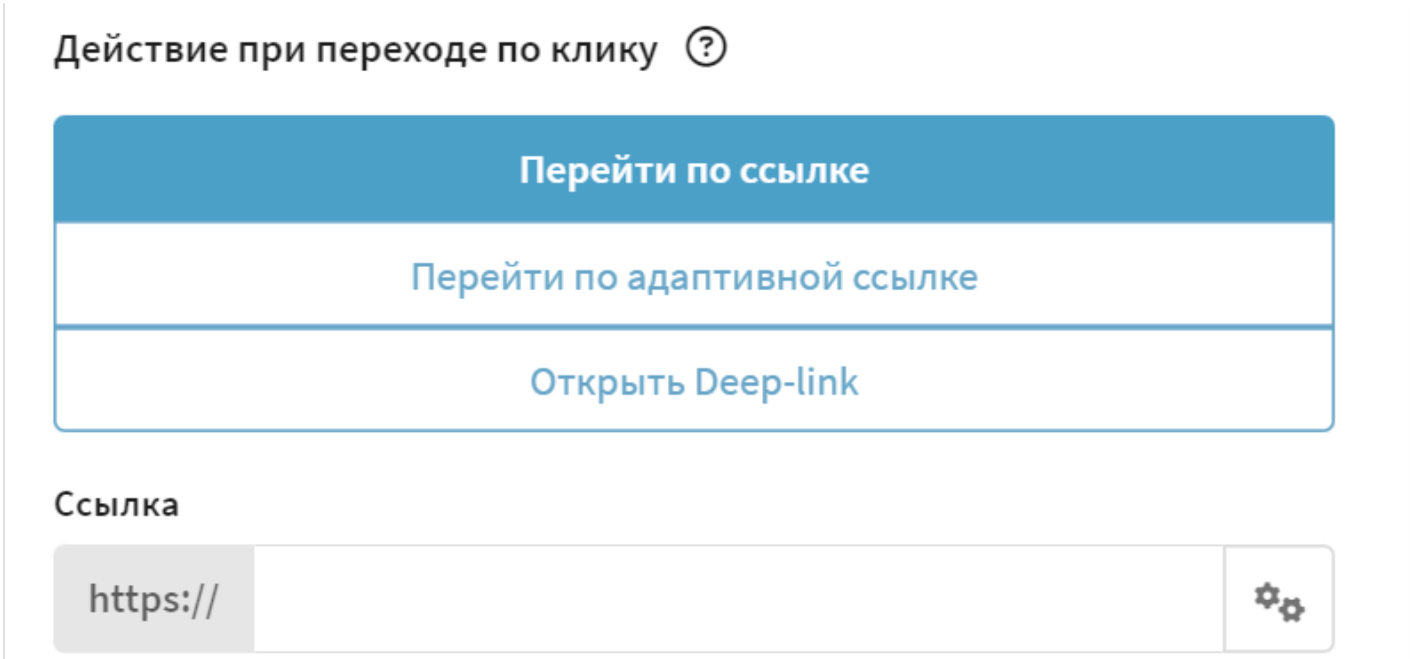
CAUTION

Обратите внимание, что ссылка, которую вы указываете для каждой платформы “перезатирает” **общую ссылку для перехода**.

Общая ссылка для перехода



Уникальная для платформы ссылка

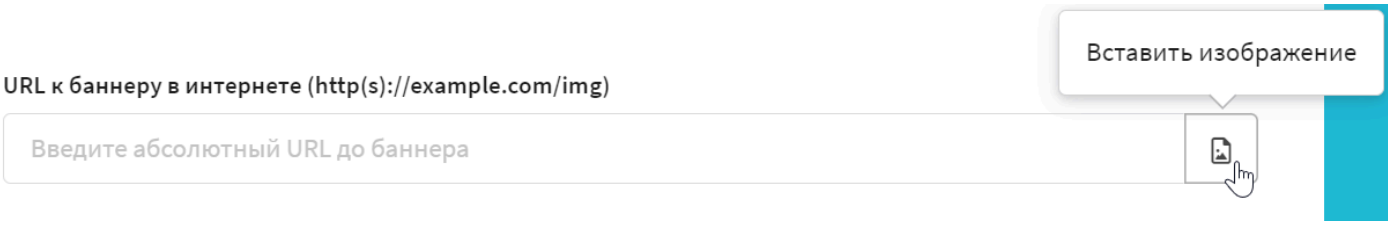


Если оставить уникальную для платформы ссылку пустой, то даже в этом случает она “перезатрет” общую ссылку для перехода.

Когда вы оставляете поле с **ссылкой для перехода по клику** пустой, то при клике по ней пользователь просто закроет push уведомление.

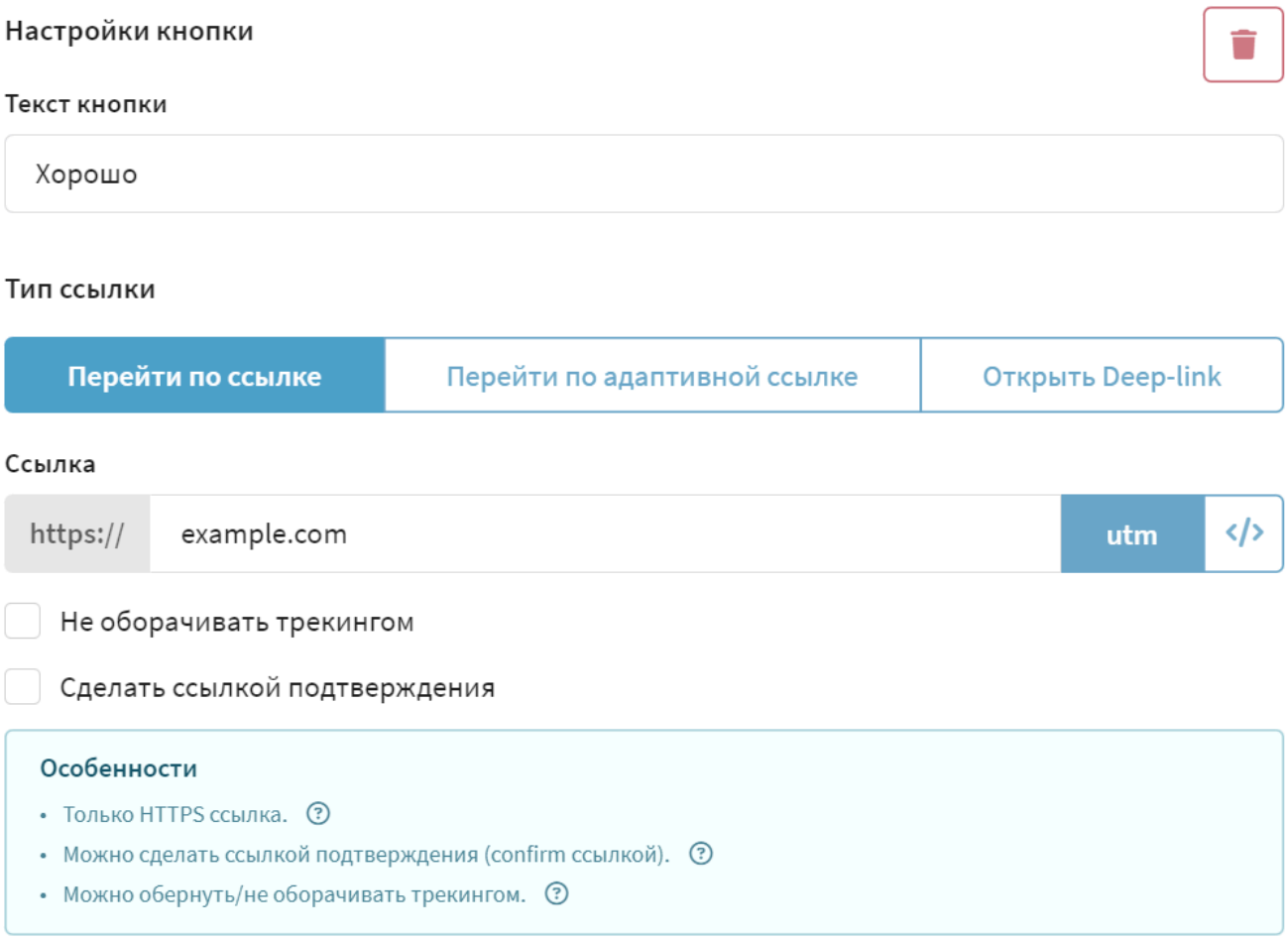
Баннер

Укажите URL-адрес изображения. Если URL корректный, баннерное изображение отобразится в предпросмотре. Кроме того, вы можете добавить изображение из галереи.



Кнопки

Нажмите **“Добавить кнопку”**. В открывшемся окне введите **текст** кнопки и выберите **действие**, которое совершит пользователь, когда нажмет на кнопку. О настройке ссылки для перехода читайте выше.

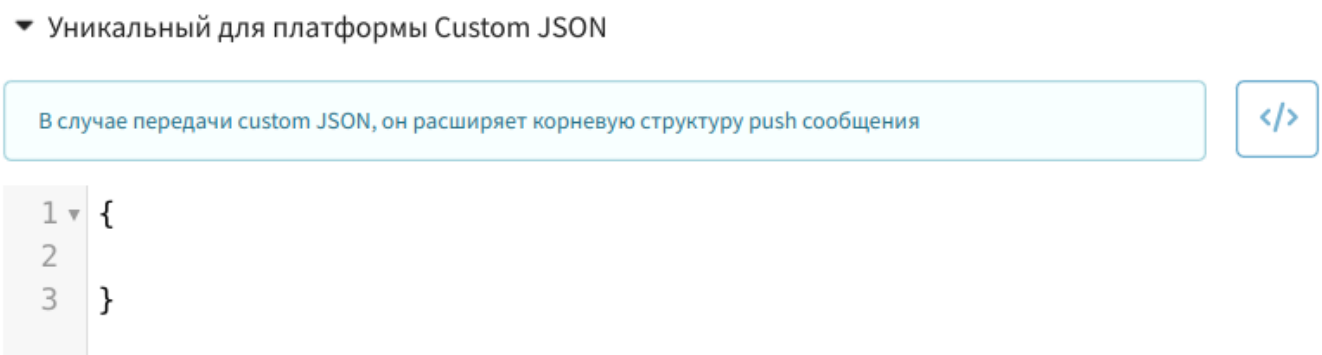


CAUTION

Если поле **“Ссылка”** оставить пустым, то при клике на кнопку пользователь закроет пуш.

Уникальный для платформы Custom JSON

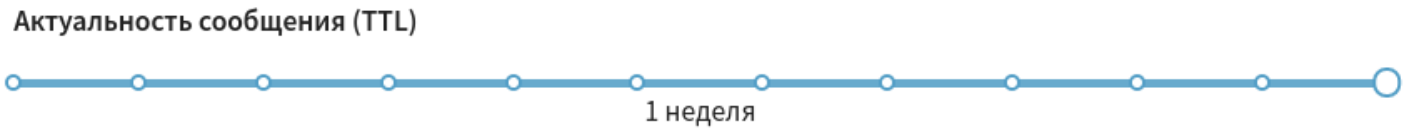
В сообщение вы можете передать дополнительные поля в формате JSON, чтобы расширить корневую структуру push-сообщения.



Актуальность сообщения (TTL)

TTL (“time to live”) – это период времени, в течение которого push-сервис мониторит статус устройства адресата на предмет его подключения к сети Интернет. Как только, в рамках установленного времени, получатель появится в сети, ему будет доставлено push-сообщение. Если же время закончилось (заданный TTL истёк), а адресат так и не появился в сети, система перестает отслеживать его статус, и, соответственно, push-уведомление считается не доставленным.

Передвиньте ползунок, чтобы установить время жизни push сообщения:

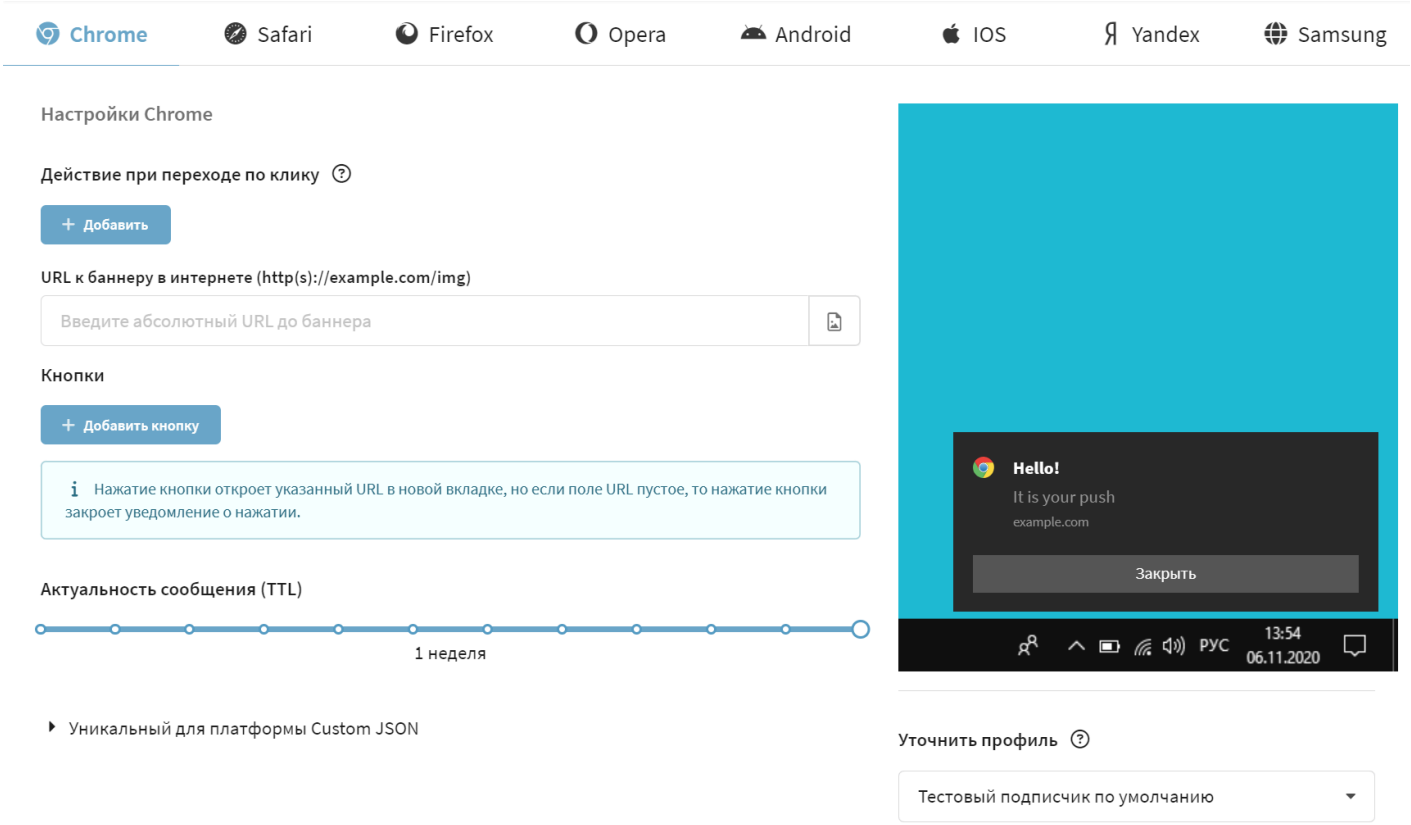


Настройка push для браузеров и ОС

Chrome

Для этого браузера доступны следующие настройки:

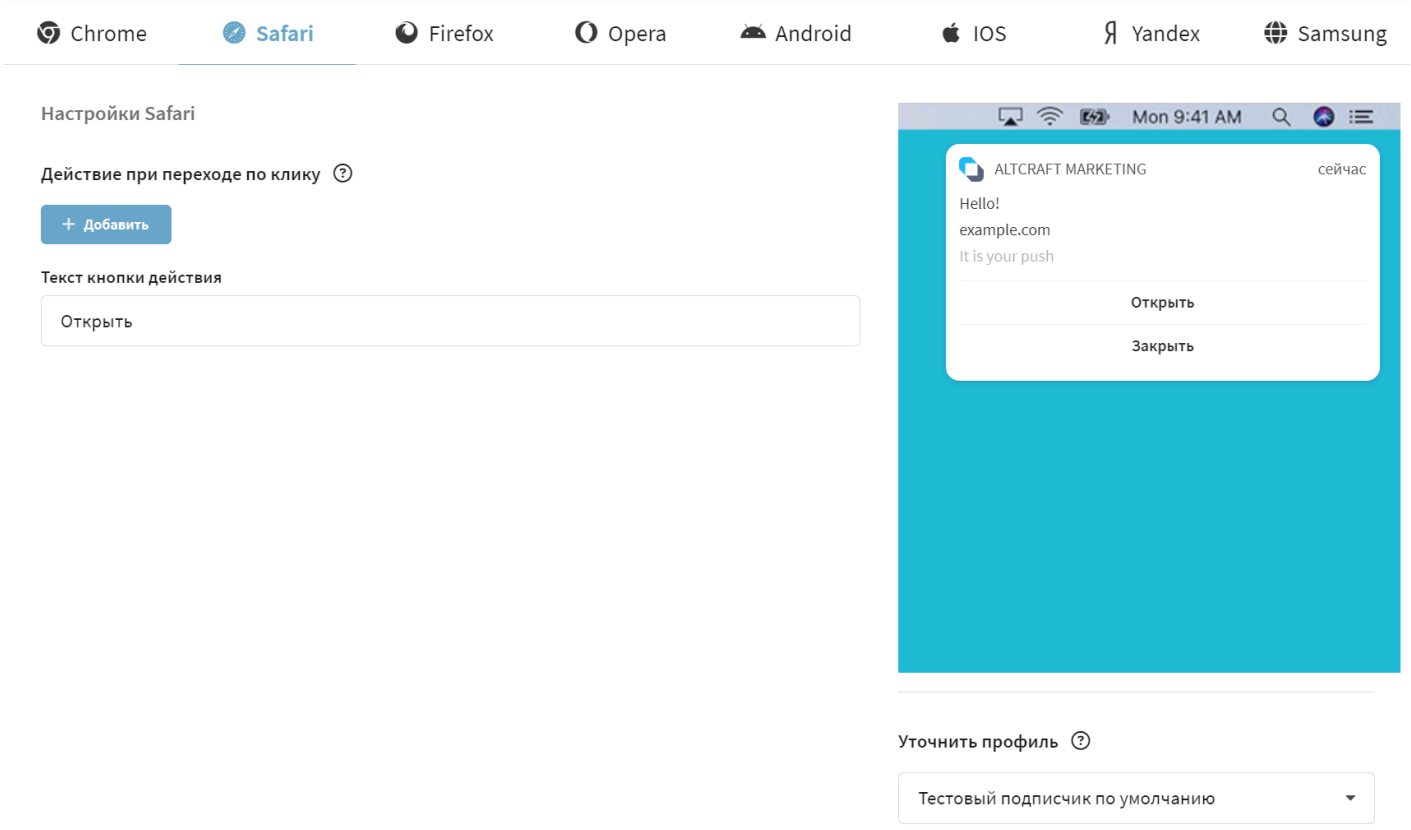
Поле	Значение
Действие при переходе по клику	Ссылка, по которой перейдет пользователь, когда кликнет по push сообщению
Баннер	Ссылка на изображение-банер
Кнопки	При нажатии на кнопку пользователь выполнит определённое действие
TTL сообщения	Время, в течение которого сервис отслеживает подключение устройства к сети для доставки сообщения, после чего уведомление удаляется
Уникальный для платформы Custom JSON	Дополнительные поля в формате JSON, чтобы расширить корневую структуру push-сообщения



Safari

Для этого браузера доступны следующие настройки:

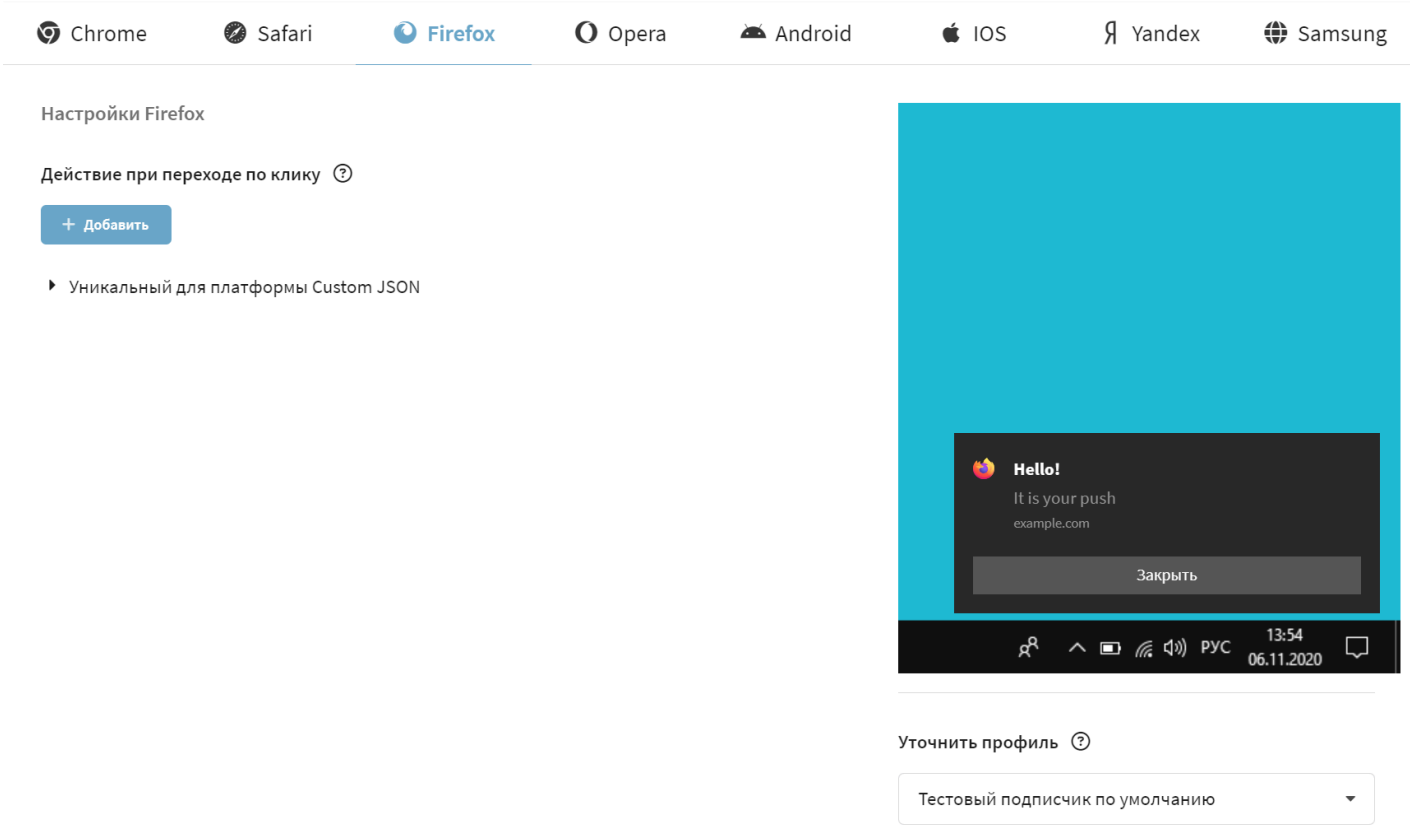
Поле	Значение
Действие при переходе по клику	Ссылка, по которой перейдет пользователь, когда кликнет по push сообщению
Текст кнопки действия	При нажатии на кнопку пользователь выполнит определённое действие



Firefox

Для этого браузера доступны следующие настройки:

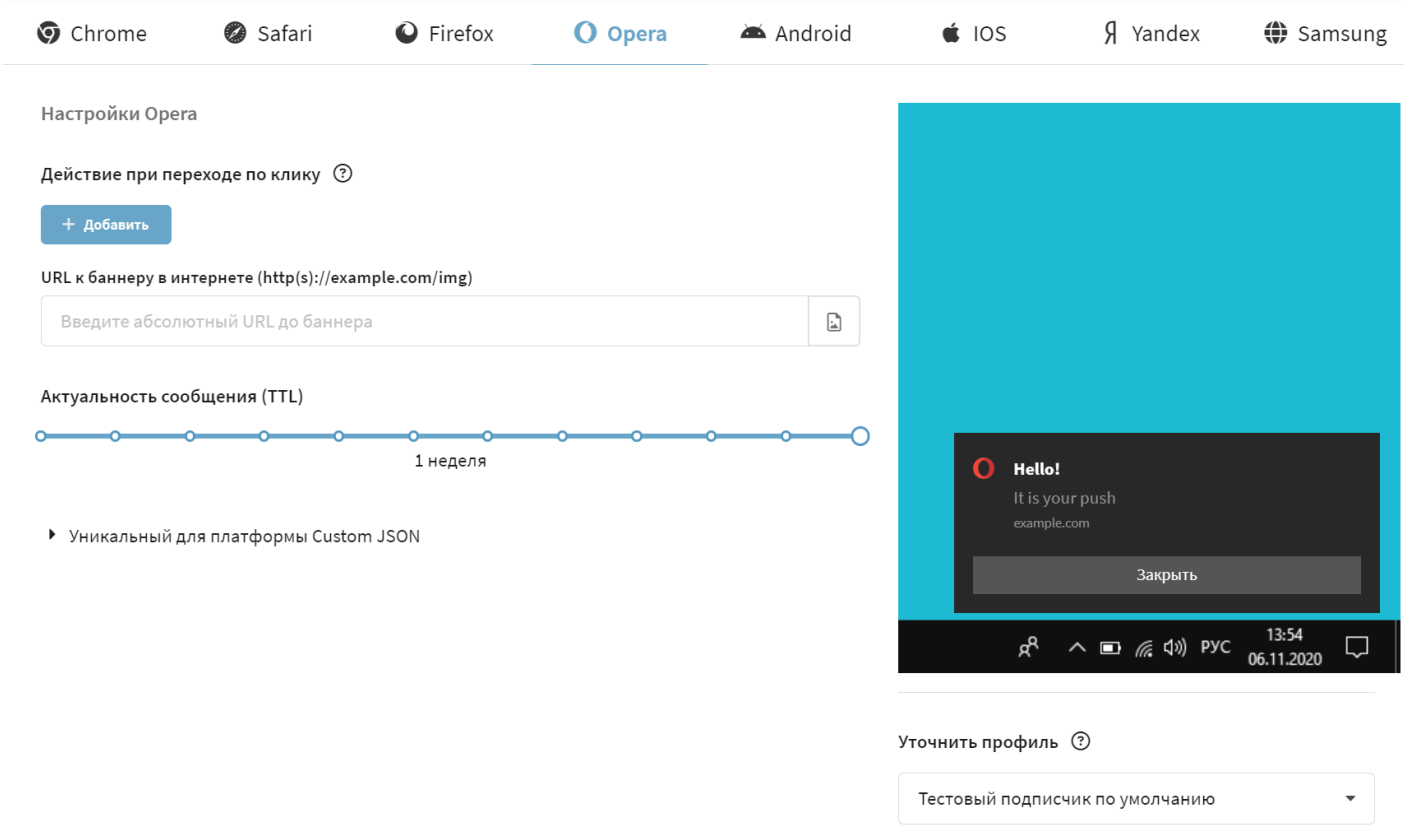
Поле	Значение
Действие при переходе по клику	Ссылка, по которой перейдет пользователь, когда кликнет по push сообщению
Уникальный для платформы Custom JSON	Дополнительные поля в формате JSON, чтобы расширить корневую структуру push-сообщения



Opera

Для этого браузера доступны следующие настройки:

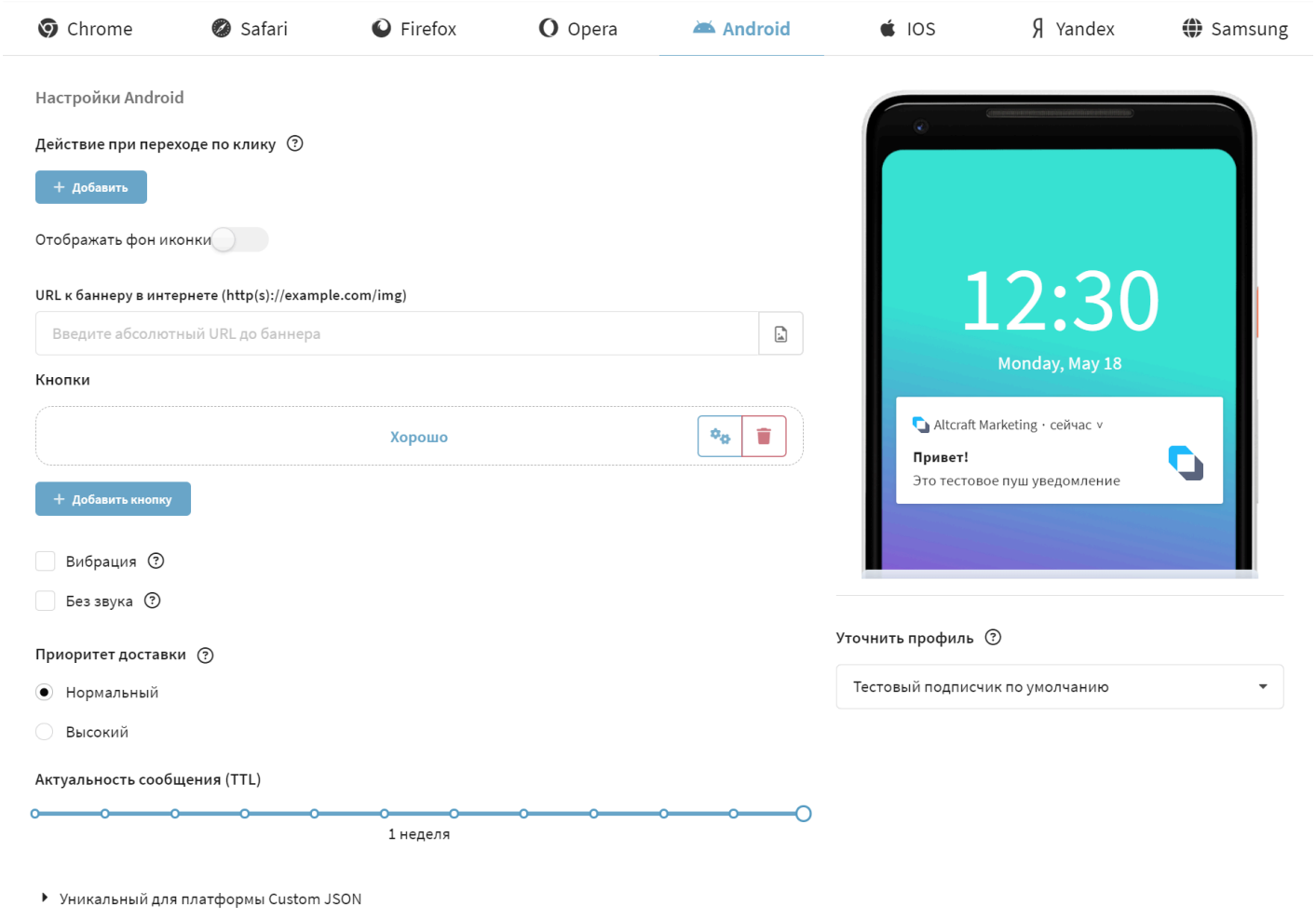
Поле	Значение
Действие при переходе по клику	Ссылка, по которой перейдет пользователь, когда кликнет по push сообщению
Баннер	Ссылка на изображение-банер
Кнопки	При нажатии на кнопку пользователь выполнит определённое действие
TTL сообщения	Время, в течение которого сервис отслеживает подключение устройства к сети для доставки сообщения, после чего уведомление удаляется
Уникальный для платформы Custom JSON	Дополнительные поля в формате JSON, чтобы расширить корневую структуру push-сообщения



Android

Для этой ОС доступны следующие настройки:

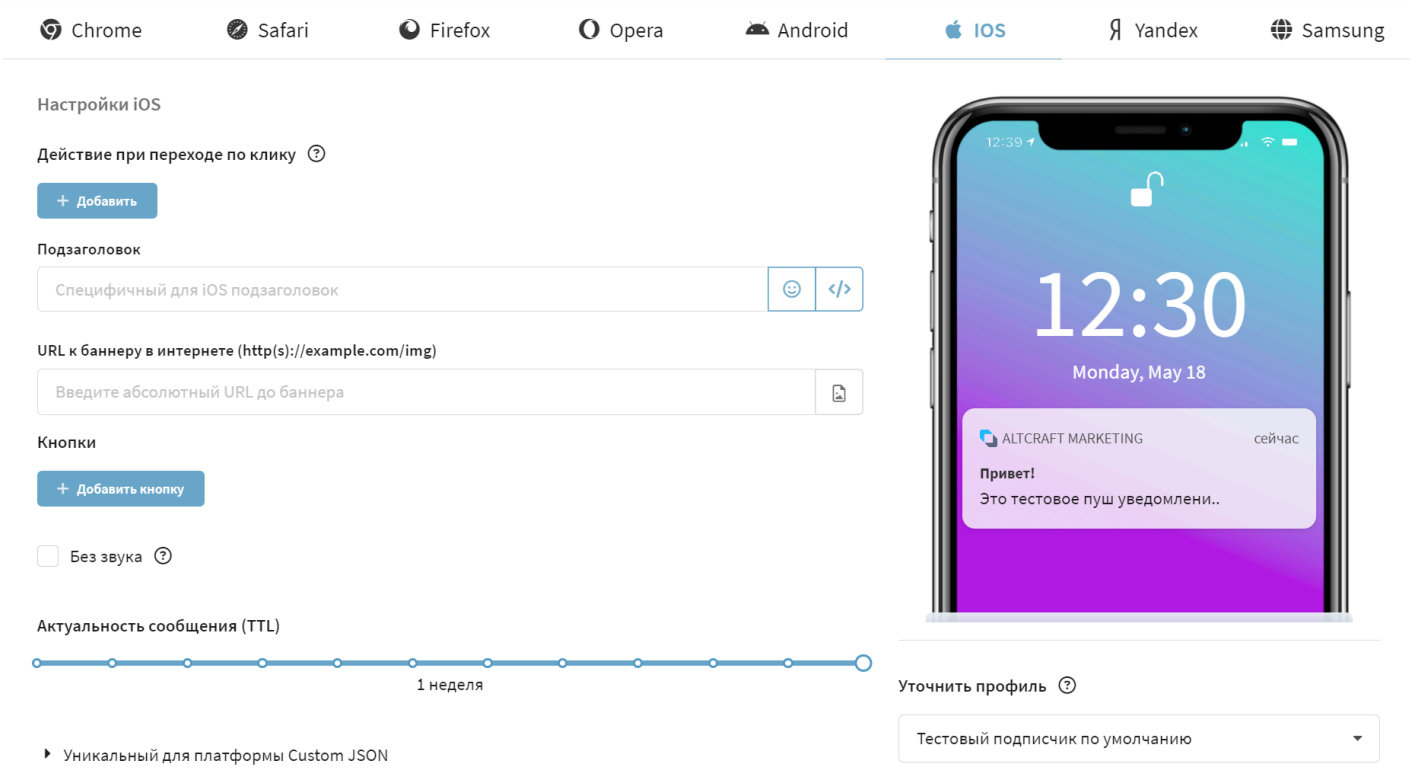
Поле	Значение
Действие при переходе по клику	Ссылка, по которой перейдет пользователь, когда кликнет по push сообщению
Фон иконки	Фон push-уведомления
Баннер	Ссылка на изображение-баннер
Кнопки	При нажатии на кнопку пользователь выполнит определённое действие
Вибрация	При получении push-уведомления устройство пользователя помимо звука, будет вопспроизводить вибрацию
Без звука	При доставке на устройство push-уведомление не будет воспроизводить звук и вибрацию
Приоритет доставки	**Нормальный:** доставка сообщений с таким приоритетом может быть отложена на неопределнное время для экономии заряда батареи. **Высокий:** push уведомление запускает приложение в фоновом режиме и сразу же получает доступ в Интернет.
TTL сообщения	Время, в течение которого данные остаются актуальными в сети, после чего они удаляются или обновляются.
Уникальный для платформы Custom JSON	Дополнительные поля в формате JSON, чтобы расширить корневую структуру push-сообщения



iOS

Для этой ОС доступны следующие настройки:

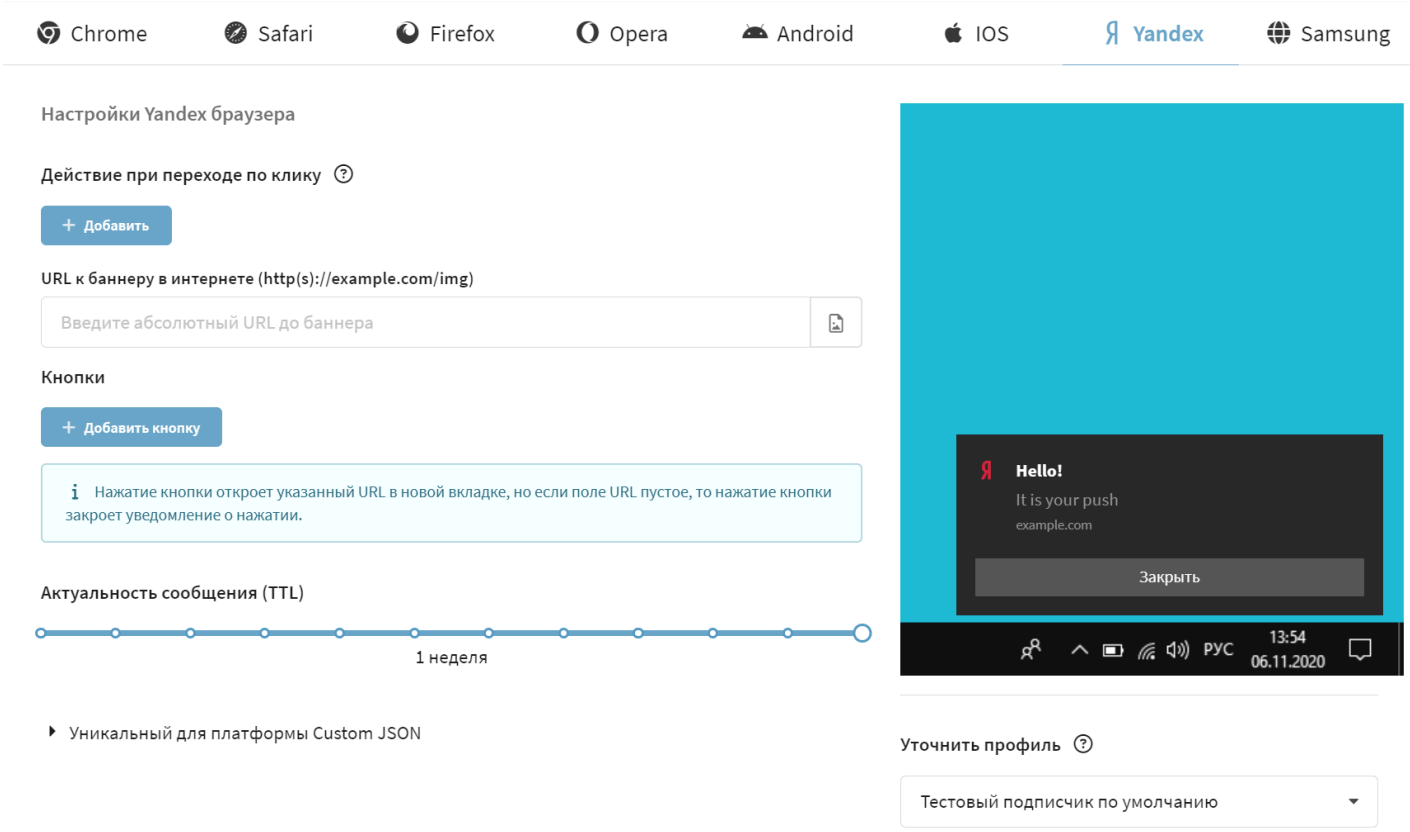
Поле	Значение
Действие при переходе по клику	Ссылка, по которой перейдет пользователь, когда кликнет по push сообщению
Подзаголовок	Специфичный для IOS подзаголовок
Баннер	Ссылка на изображение-банер
Кнопки	При нажатии на кнопку пользователь выполнит определённое действие
Без звука	При доставке на устройство push-уведомление не будет воспроизводить звук и вибрацию
TTL сообщения	Время, в течение которого сервис отслеживает подключение устройства к сети для доставки сообщения, после чего уведомление удаляется
Уникальный для платформы Custom JSON	Дополнительные поля в формате JSON, чтобы расширить корневую структуру push-сообщения



Yandex

Для этого браузера доступны следующие настройки:

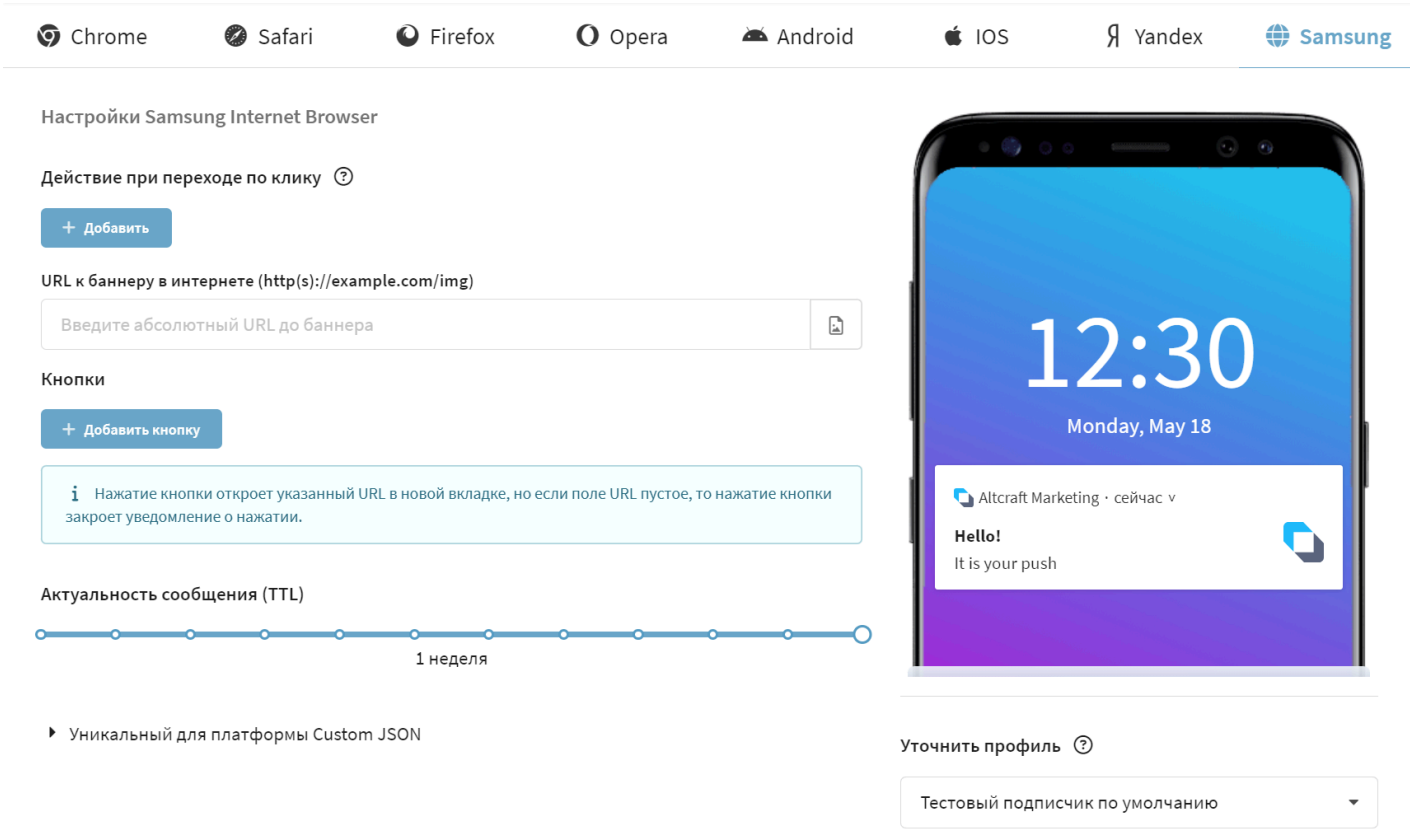
Поле	Значение
Действие при переходе по клику	Ссылка, по которой перейдет пользователь, когда кликнет по push сообщению
Баннер	Ссылка на изображение-банер
Кнопки	При нажатии на кнопку пользователь выполнит определённое действие
TTL сообщения	Время, в течение которого сервис отслеживает подключение устройства к сети для доставки сообщения, после чего уведомление удаляется
Уникальный для платформы Custom JSON	Дополнительные поля в формате JSON, чтобы расширить корневую структуру push-сообщения



Samsung

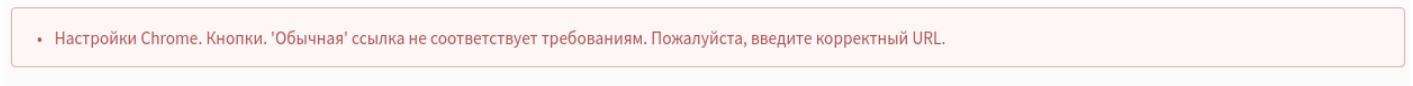
Для этого браузера доступны следующие настройки:

Поле	Значение
Действие при переходе по клику	Ссылка, по которой перейдет пользователь, когда кликнет по push сообщению
Баннер	Ссылка на изображение-банер
Кнопки (не более двух)	При нажатии на кнопку пользователь выполнит определённое действие
TTL сообщения	Время, в течение которого сервис отслеживает подключение устройства к сети для доставки сообщения, после чего уведомление удаляется
Уникальный для платформы Custom JSON	Дополнительные поля в формате JSON, чтобы расширить корневую структуру push-сообщения



Как только вы настроите все необходимые параметры своего шаблона, сохраните его, нажав на кнопку “**Сохранить**” в правом верхнему углу экрана:

Если вы допустили какую-то ошибку при заполнении полей шаблона, она отобразится в верхней части экрана. Исправьте указанную ошибку и сохраните шаблон еще раз.

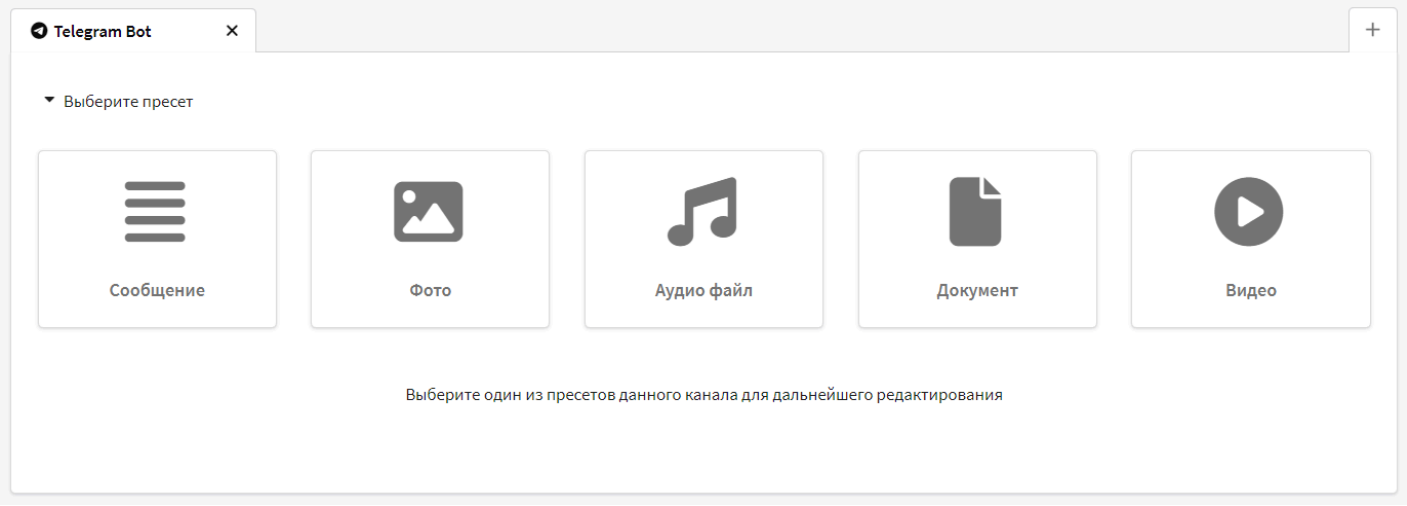


Теперь вы можете использовать этот шаблон для отправки Rich Push уведомлений. Подробнее о том, как это сделать, читайте здесь: [Web push: настройка и подключение](#).

Telegram-шаблон

В этой статье описывается, как создать шаблоны для отправки сообщений через такие каналы как **Telegram Group** и **Telegram Bot**.

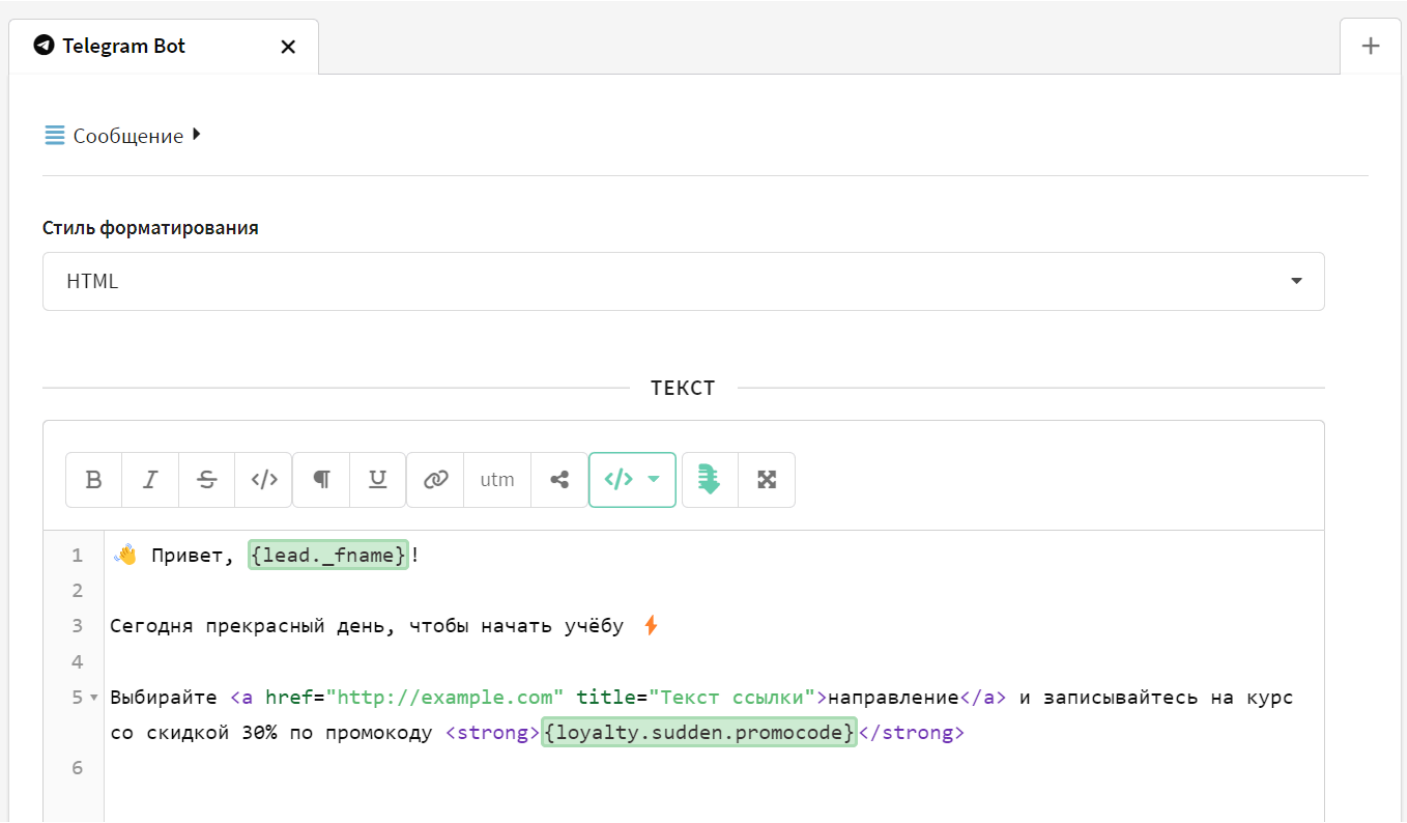
Сначала выберите тип контента, который хотите отправить подписчикам:



Сообщение

Выберите стиль форматирования. Рекомендуем выбрать **HTML**, поскольку остальные стили форматирования устарели. Затем введите текст сообщения.

> **TIP**
>
> Максимальная длина текстового сообщения в Telegram — 4096 символов. Обратите внимание, данные, которые подтягиваются из динамических переменных, могут превышать этот лимит. Для обеспечения надёжной доставки мы рекомендуем следить за длиной сообщения.



Панель инструментов в Telegram-редакторе

Для облегчения форматирования и работы с объектами доступны следующие функции:

B

— **жирный текст**.

I

— *курсив*.

S

— ~~зачёркнутый текст~~.

</>

— <блок кода>.

¶

— предварительно отформатированный текст.

U

— подчеркнутый текст.

🔗

— редактор ссылок (см. вкладку email). Ссылки, вставленные через платформу, автоматически сокращаются. Итоговая длина пути ссылки без учета трекинг-домена — 12-14 символов, включая `/`.

utm

— добавление UTM-меток.

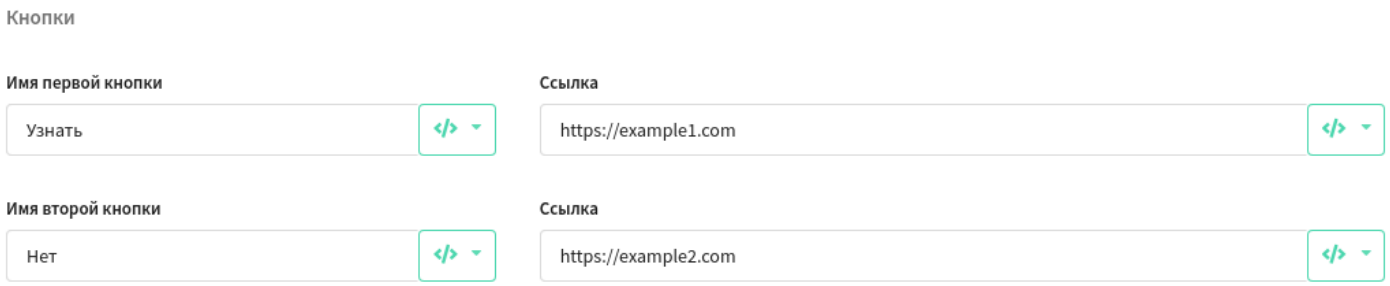
— добавление кнопок **“Поделиться в социальных сетях”**.

— меню переменных и функций редактора. С помощью этой кнопки вы можете добавить в сообщение данные профилей клиентов, промокоды, циклы и логические выражения, а также динамический контент и системные ссылки.

— перенос строк в редакторе.

— полноэкранный режим работы в редакторе.

Чтобы добавить кнопку в сообщение, введите имя и ссылку, по которой перейдет пользователь, кликнув на кнопку. Возможно добавить не более 2-х кнопок.

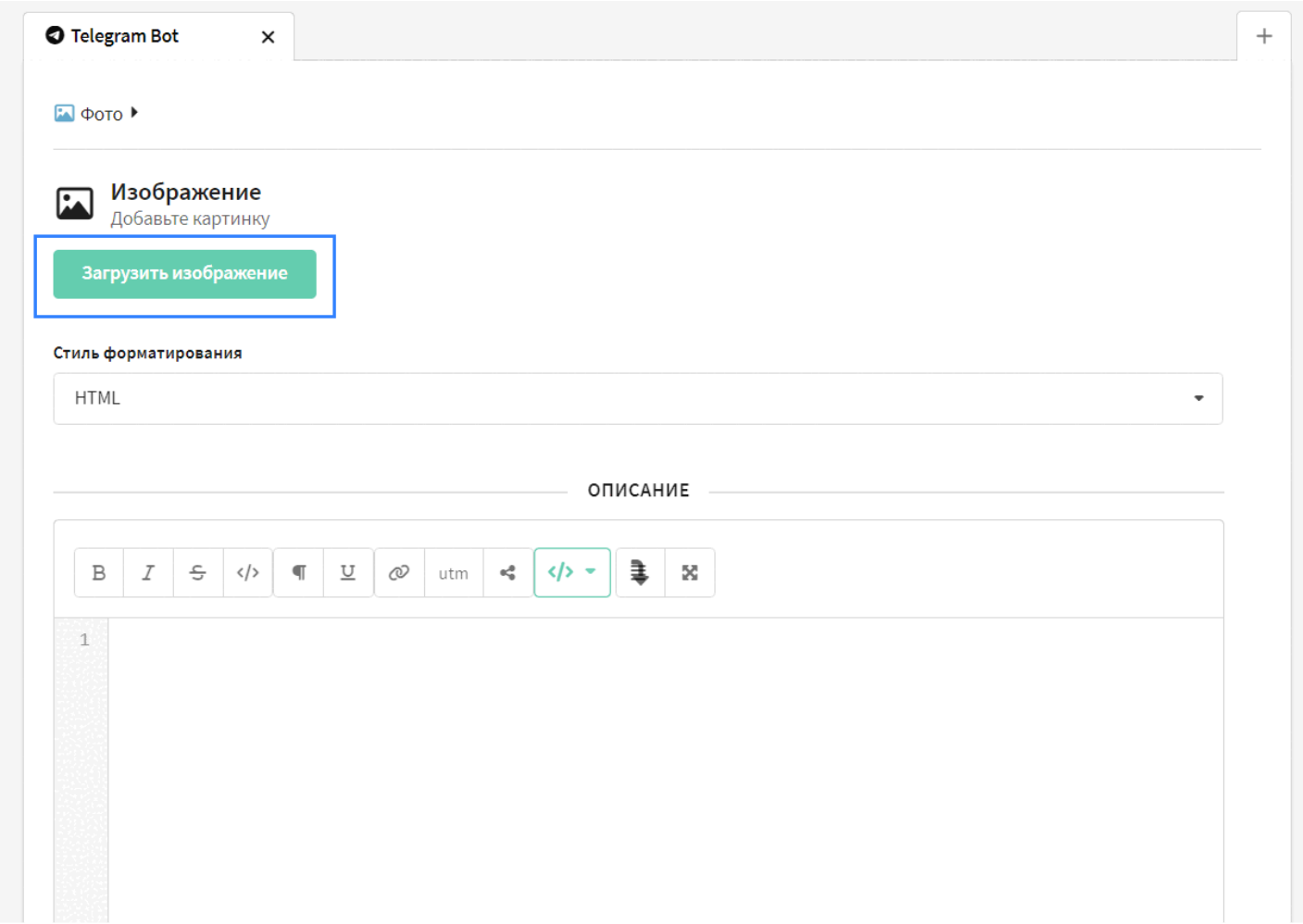


TIP

Ссылки в шаблонах для Telegram Bot и Telegram Group сокращаются.

Медиа файлы

Чтобы добавить изображение, нажмите на кнопку **“Загрузить изображение”** и выберите изображение в галерее.



Требования к изображениям:

- Форматы: PNG, JPG, JPEG, GIF.
- Максимальный размер: 10 МБ.
- Сумма ширины и высоты фотографии не должна превышать 10000.
- Соотношение ширины и высоты должно быть не более 20.

Добавить аудио, видео или документ можно двумя способами:

1. **Загрузить файл** с вашего устройства.
2. **Добавить файл по локальному пути.** Выберите директорию хранения файла на сервере, на котором установлена платформа. В пути директории вы можете использовать переменные профиля клиента. Введите расположение файла и нажмите “Добавить”.

Требования к аудио файлу:

- Форматы: MP3 и M4A.
- Максимальный размер: 5 МБ.

Требования к документу:

- Форматы: произвольный формат документа.
- Максимальный размер: 5 МБ.

Требования к видео:

- Форматы: MP4.

- Максимальный размер: 5 МБ.

TIP

К файлам любого типа также можно добавить описание **(до 1024 символов)**.

WhatsApp-шаблон

Описание

В данный момент через Altcraft Platform можно отправлять только **типовые шаблоны (HSM)**.

Типовый шаблон — это HSM-шаблон, который предварительно необходимо согласовать с Facebook. Используйте такой шаблон, если вы инициируете общение с пользователем. С 1 июля 2025 года каждый отправленный шаблон тарифицируется отдельно в зависимости от его категории.

Если клиент первым написал вам в WhatsApp, открывается 24-часовое окно, в течение которого можно бесплатно отправлять сервисные и нешаблонные сообщения.

Шаблоны сообщений для отправки в WhatsApp создаются в личном кабинете провайдера.

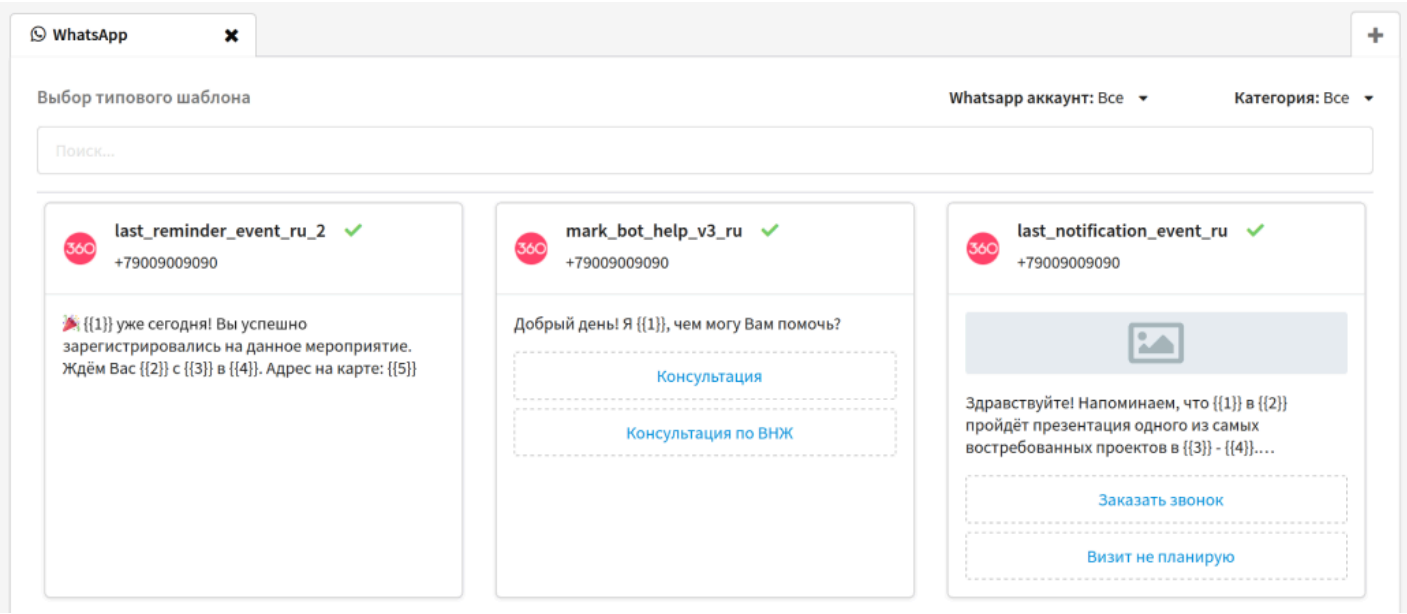
TIP

Больше информации о создании шаблонов вы можете найти в документации вашего провайдера:

- **Edna:** https://pulse-alpha-six.vercel.app/docs/template/whatsapp/operator-template
- **360dialog:** https://docs.360dialog.com/docs/360-client-hub/template-management-ui

Создание сообщения

Все шаблоны, доступные в WhatsApp-аккаунтах, подключенных через Edna или через 360dialog, автоматически доступны в Altcraft Platform и отображаются в редакторе. Для поиска нужного шаблона в платформе вы можете ввести запрос в поисковую строку или использовать фильтр по аккаунту/категории в правом верхнем углу.



CAUTION

В рассылках можно использовать только шаблоны, согласованные с Facebook.

Статус шаблона отображается справа от его названия:

— шаблон успешно прошел модерацию;

— шаблон отклонён;

— шаблон находится на модерации;

— шаблон не отправлен на модерацию.

Выберите шаблон, который хотите использовать в рассылке. Чтобы перейти к созданию сообщения на основе WhatsApp-шаблона, нажмите на него. В редакторе отобразится структура шаблона:

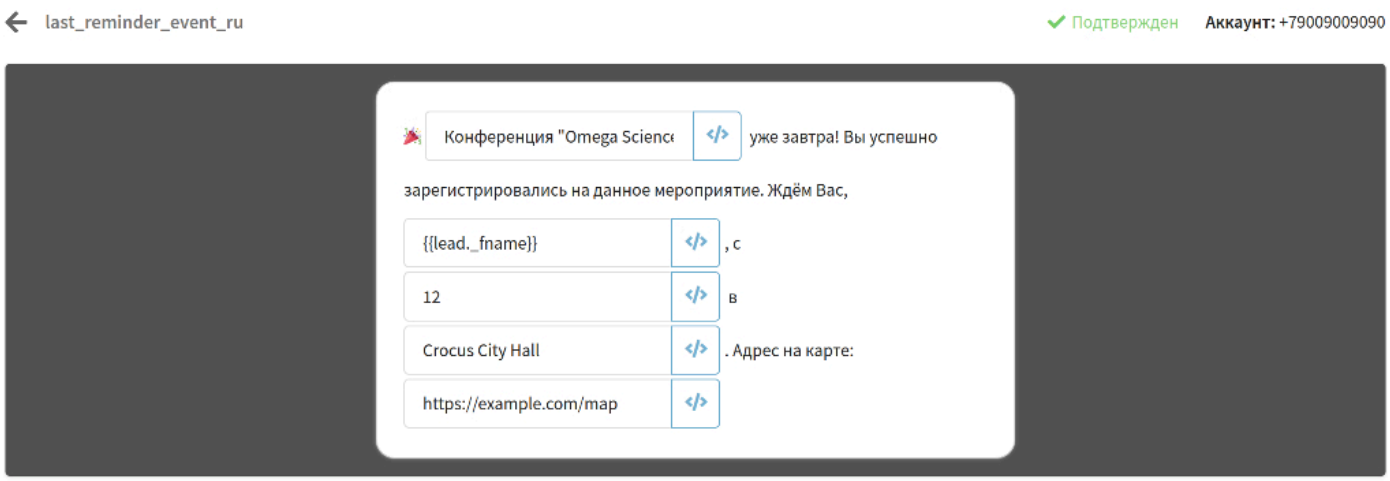
- Header — шапка шаблона с приветственным текстом или медифайлом;
- Body — основной текст сообщения с переменными (обязательный блок);
- Footer — дополнительный текст;
- Button — блок с кнопками для добавления интерактивности.

Вы можете изменить такие элементы шаблона, как **переменные**, **медиафайлы** и **кнопки**.

Переменные

Текстовый блок в WhatsApp состоит из фиксированного текста и переменных. **Переменные** – это текст шаблона, который меняется для каждого нового сообщения (имя, адрес, название мероприятия, дата и время, ссылка и т.д.).

Вы можете вручную прописать значения переменных или добавить переменные Altcraft. Чтобы открыть список переменных, нажмите `</>`. При отправке сообщения введённый текст автоматически подставится в шаблон.



TIP

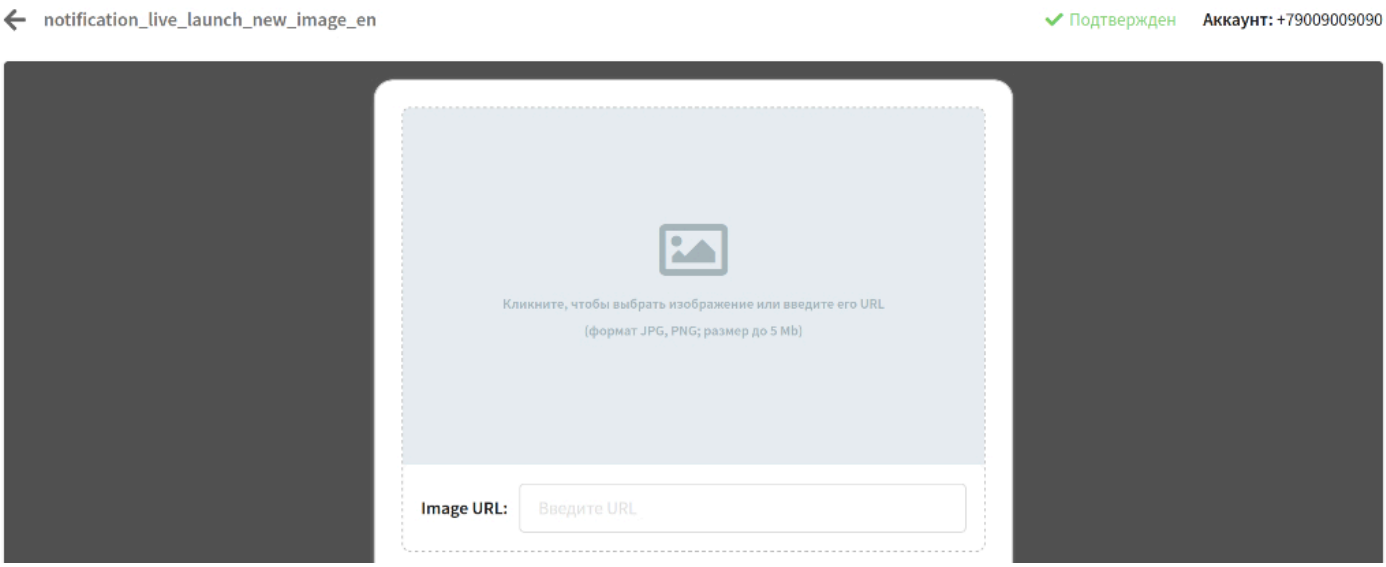
Вставленные ссылки автоматически сокращаются. Итоговая длина пути ссылки без учета трекинг-домена — 12-14 символов, включая `/`.

Медиафайлы

Доступные медиафайлы:

- изображение (JPG/PNG, до 5 МБ, рекомендуемый размер изображения 640x640.)
- файл (любой корректный MIME тип, до 16 МБ)
- видео (MP4, 3GP, 3GPP, до 100 МБ)

Чтобы вставить изображение, нажмите на область загрузки и выберите изображение в галерее платформы или введите URL:



Для добавления файла или видео необходимо ввести URL-адрес, по которому располагается файл.

Кнопки

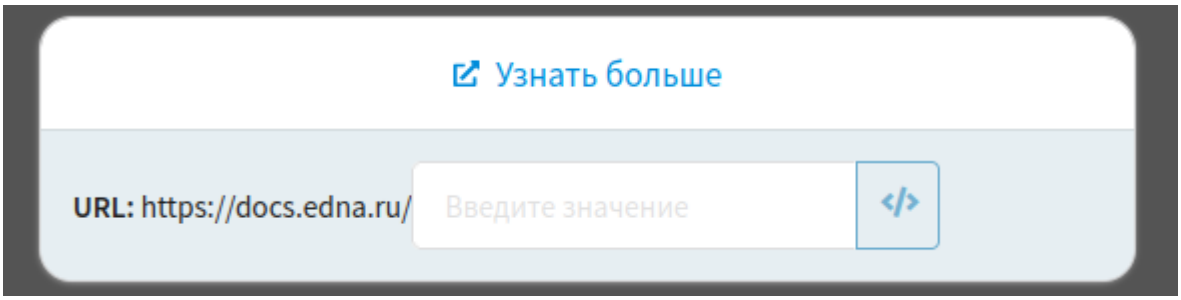
WhatsApp предлагает два типа **кнопок** для HSM-шаблонов:

- **Call-to-action** — кнопка призыва к действию;
- **Quick-reply** — кнопка быстрого ответа.

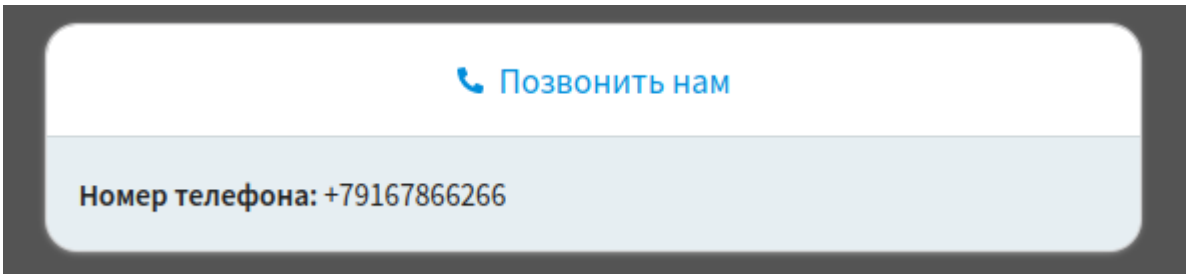
Кнопки Call-to-action

В качестве CTA выступают кнопка-ссылка (URL) и кнопка-звонок (Phone). Всего в шаблоне может быть 2 кнопки данного типа.

При клике на **кнопку-ссылку** пользователь переходит на заданную страницу. Если в WhatsApp-шаблоне задан параметризируемый URL, необходимо его заполнить.

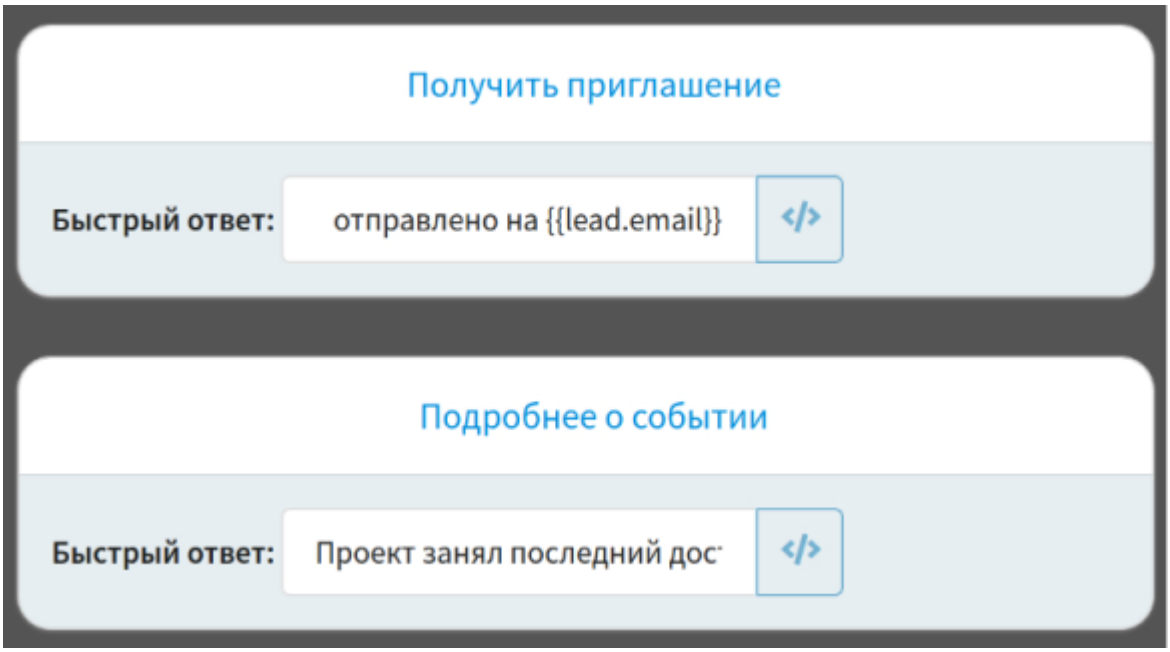


При клике на **кнопку-звонок** происходит набор номера. Номер телефона нельзя редактировать.



Кнопки Quick-reply

При клике на кнопку пользователь получает быстрый ответ, прописанный в шаблоне. Шаблон может содержать не более 3 кнопок данного типа.



TIP

Сохранение шаблона возможно только после заполнения всех переменных, кнопок, а также добавления медиафайла.

Предпросмотр сообщения

Если вы хотите проверить, как будет выглядеть сообщение, перейдите в режим предпросмотра шаблона, кликнув на кнопку

:

Предпросмотр для сообщения Напоминание о конференции

Каналы:

WhatsApp

Настроить API контент

Не использовать ресурсы

Тестовый профиль

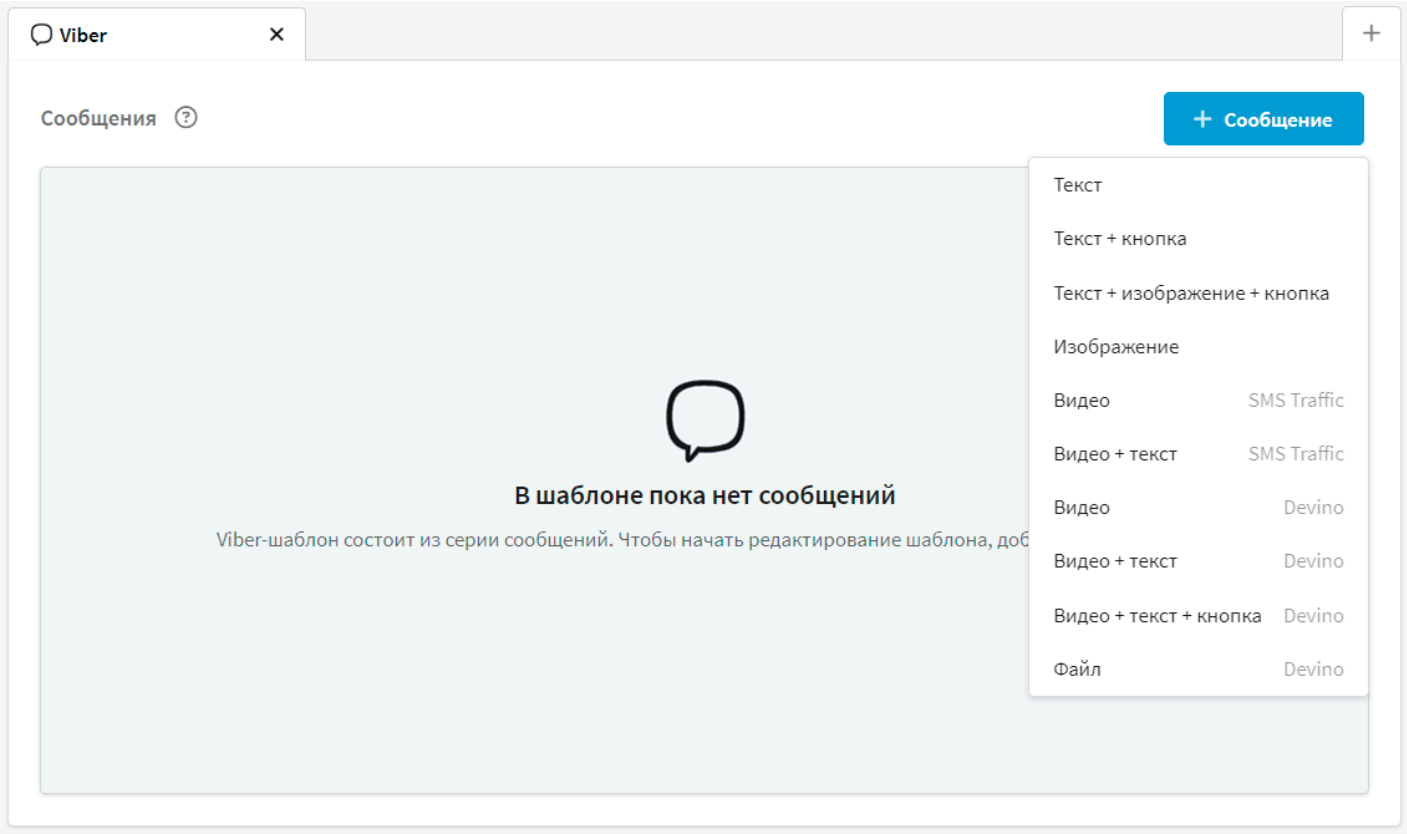
Сохранить изображение

Закрыть

Viber™-шаблон

Создание шаблона

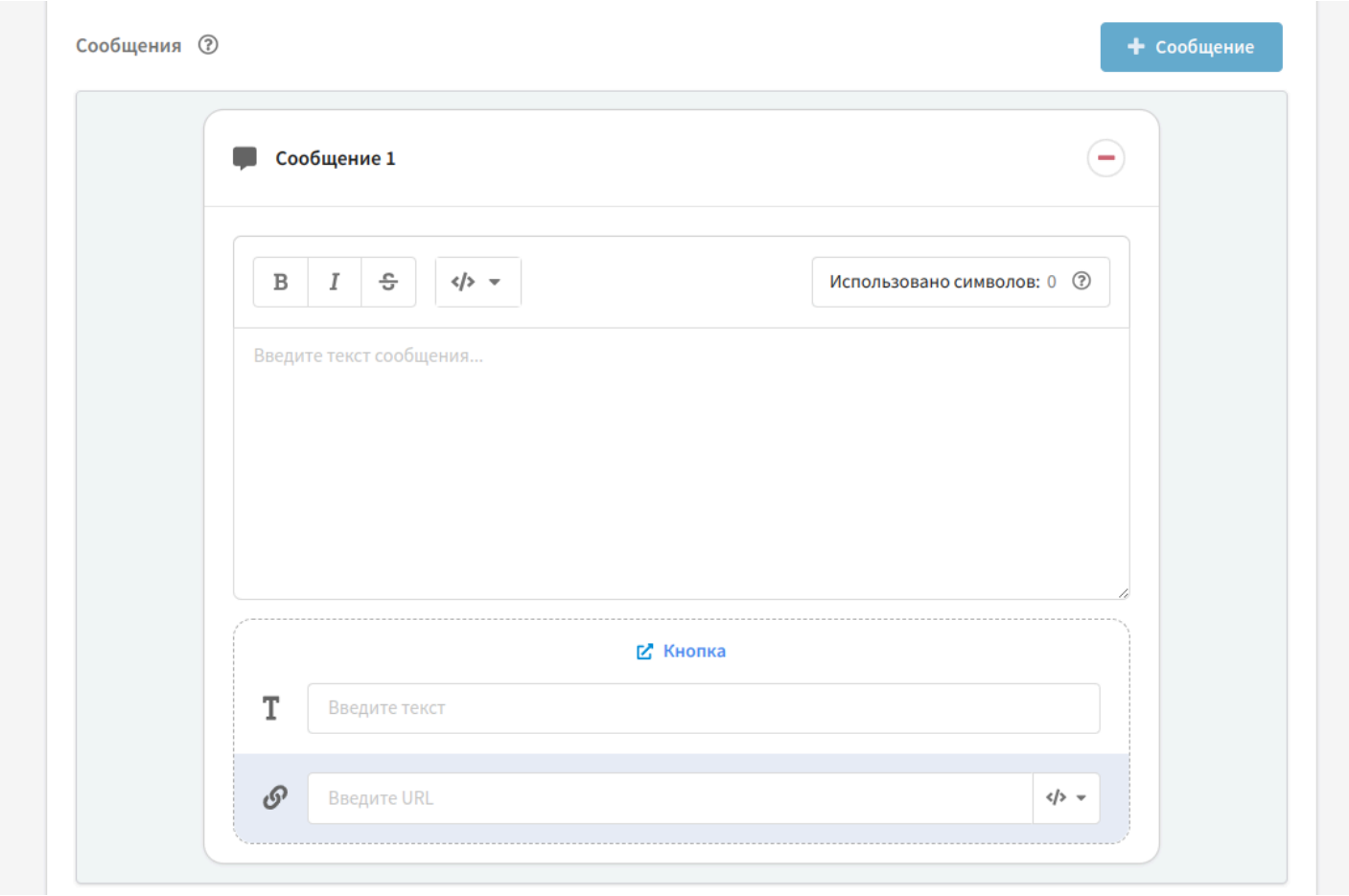
Viber™-шаблон состоит из серии сообщений. Чтобы начать редактирование шаблона, добавьте сообщение с помощью кнопки **+ Сообщение**:



Максимальное количество сообщений в серии — 5. Если в шаблоне используется несколько сообщений, каждое из них будет отправляться с интервалом в 2 секунды.

Чтобы убрать лишнее сообщение, нажмите

.



Типы сообщений

Доступны следующие типы сообщений:

Тип сообщения	Devino	SMSTraffic	MTC OmniChannel	Rapporto
Текст	+	+	+	+
Текст + кнопка	+	+	+	+
Текст + изображение + кнопка	+	+	+	+
Изображение	+	+	+	+
Видео	+	+	-	-
Видео + текст	+	+	-	-
Видео + текст + кнопка	+	-	-	-
Файл	+	-	-	-

Текст

Введите текст вашего сообщения в редакторе, используя формат Markdown. Для удобства форматирования на верхней панели доступны следующие функции:

— жирный текст

— курсив

— зачёркнутый текст

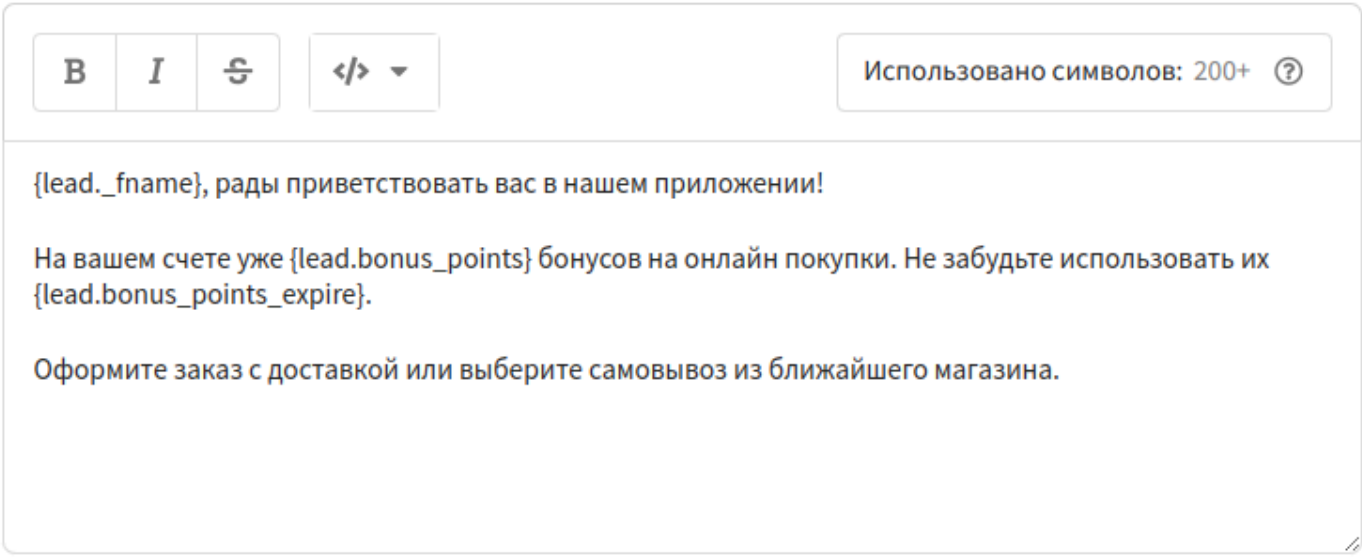
— список переменных и функций редактора для персонализации сообщения.

CAUTION

Обратите внимание, в режиме предпросмотра отображается Markdown разметка (например, `"текст"`), однако получатель сообщения увидит корректное форматирование (**“текст”**).

Также в блок “Текст” нельзя добавлять ссылки в формате Markdown.

Все ссылки, вставленные через платформу, автоматически сокращаются. Итоговая длина пути ссылки без учета трекинг-домена — 12-14 символов, включая `/`.



TIP

В Viber™-сообщения можно добавлять эмодзи. Для вставки эмодзи необходимо использовать его код. Например, чтобы добавить 🙂, введите `(smiley)`. Коды для всех доступных эмодзи вы найдёте по этой ссылке.

Обратите внимание, при подсчете количества символов в сообщении учитывается каждый символ в коде эмодзи.

TIP

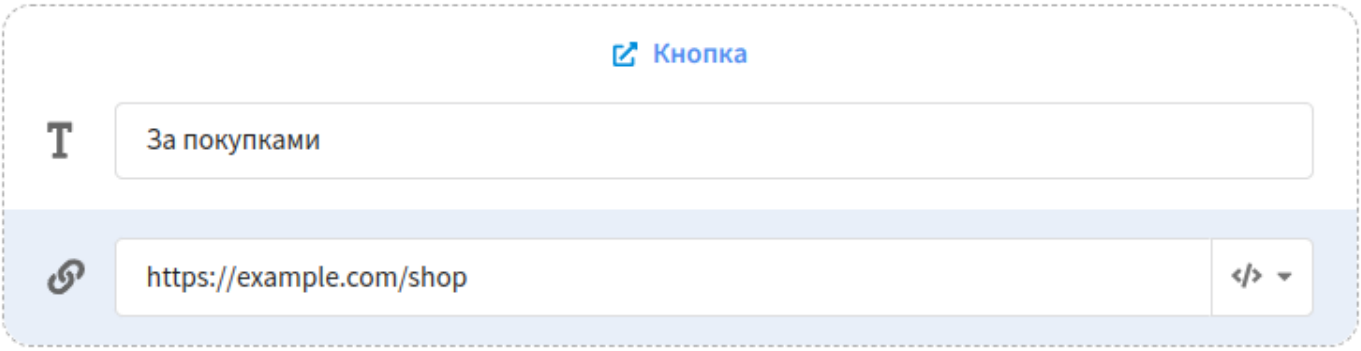
В режиме предпросмотра эмодзи не отображаются!

INFO

Справа находится счётчик символов. Максимальная длина одного сообщения — 1000 символов. Обратите внимание, что данные, которые подтягиваются из динамических переменных, могут превышать этот лимит. Для обеспечения надёжной доставки мы рекомендуем следить за длиной сообщения.

Кнопка

Укажите текст кнопки, который отобразится на кнопке, и ссылку для перехода по клику. В URL-адрес можно добавлять переменные.



Изображение

Поддерживаемые форматы — jpg, jpeg, png. Максимальный размер — 5 МБ.

Добавить изображения можно двумя способами:

- **Из галереи платформы.** Чтобы открыть галерею, кликните на область загрузки. Выберите нужное изображение или загрузите новое с вашего устройства.
- **Из внешнего источника.** Укажите ссылку на изображение.

Видео

Поддерживаемые форматы — mp4, 3gpp. Кодек — H.264. Максимальный размер — 5 МБ.

Для добавления видео укажите его URL-адрес. Также необходимо установить обложку для видео. Вы можете выбрать изображение из галереи платформы или вставить ссылку на изображение.

Файл

Поддерживаемые форматы — doc, docs, rtf, odt, odf, txt, pdf, eps, xls, xlsx, ods, csv. Максимальный размер файла — 200 МБ. Ограничение на количество символов в названии файла — 25 (с учетом расширения).

Для отправки файла укажите ссылку на него.

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Notify-шаблон

INFO

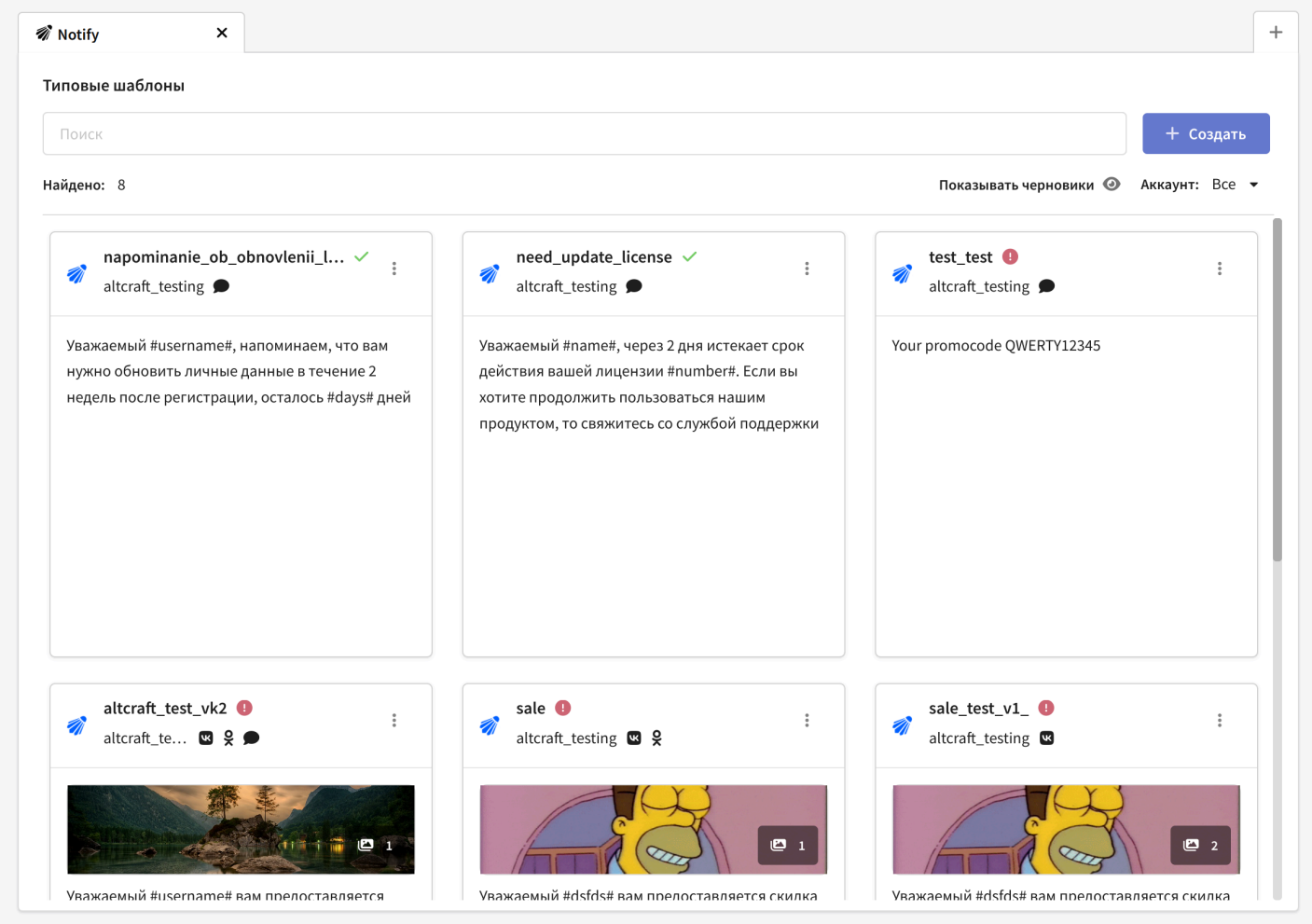
Согласно политике Notify, в данный момент новые доступы к личному кабинету сервиса **не предоставляются.**

Поддержка текущих пользователей доступна через техподдержку MAX Business.

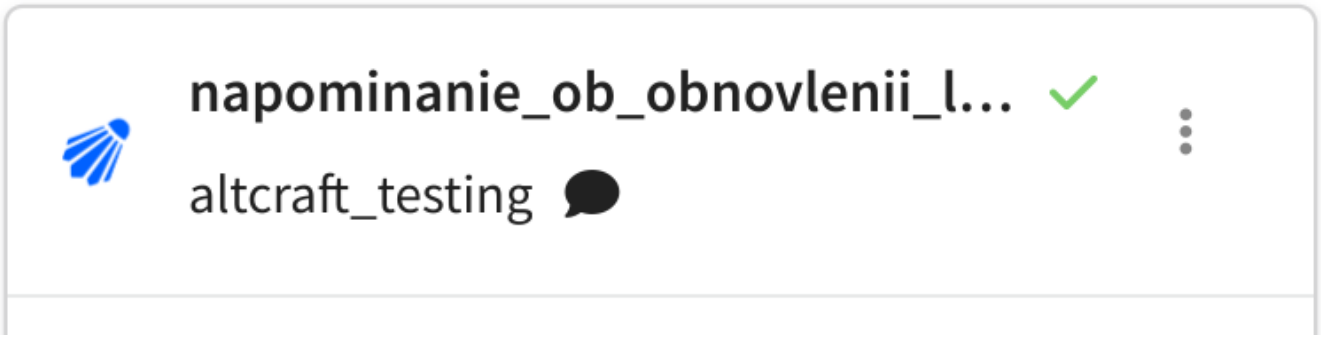
При помощи канала Notify вы можете делать рассылку бизнес-уведомлений через **“Вконтакте”**, **“Одноклассники”**, а также отправлять напоминания по **SMS**. Сообщения в социальных сетях и SMS-сообщения настраиваются в разных **отправках**, которые следуют друг за другом.

Существующие типовые шаблоны

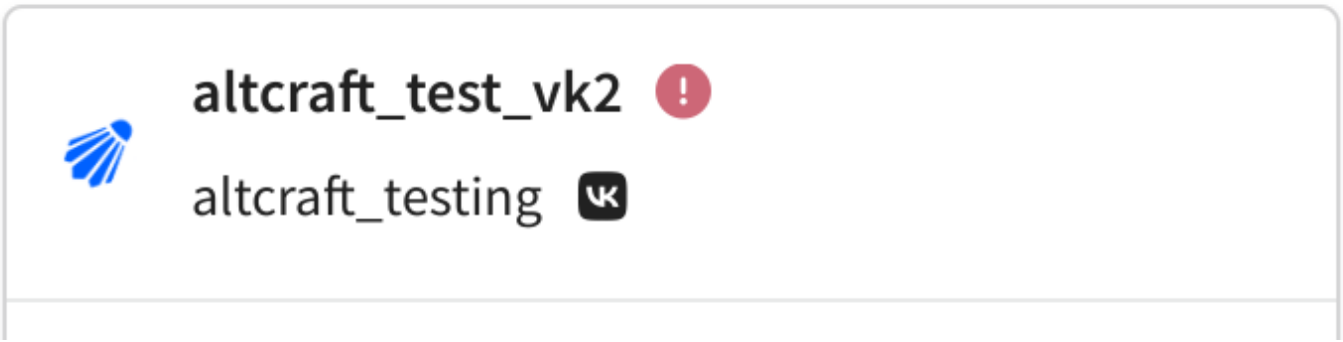
При открытии редактора сообщений для Notify вам будут показаны **типовые шаблоны**, созданные в вашем личном кабинете Notify:



Все шаблоны для отправок проходят **модерацию сервиса**. Одобренные сервисом шаблоны имеют пометку в виде галочки:

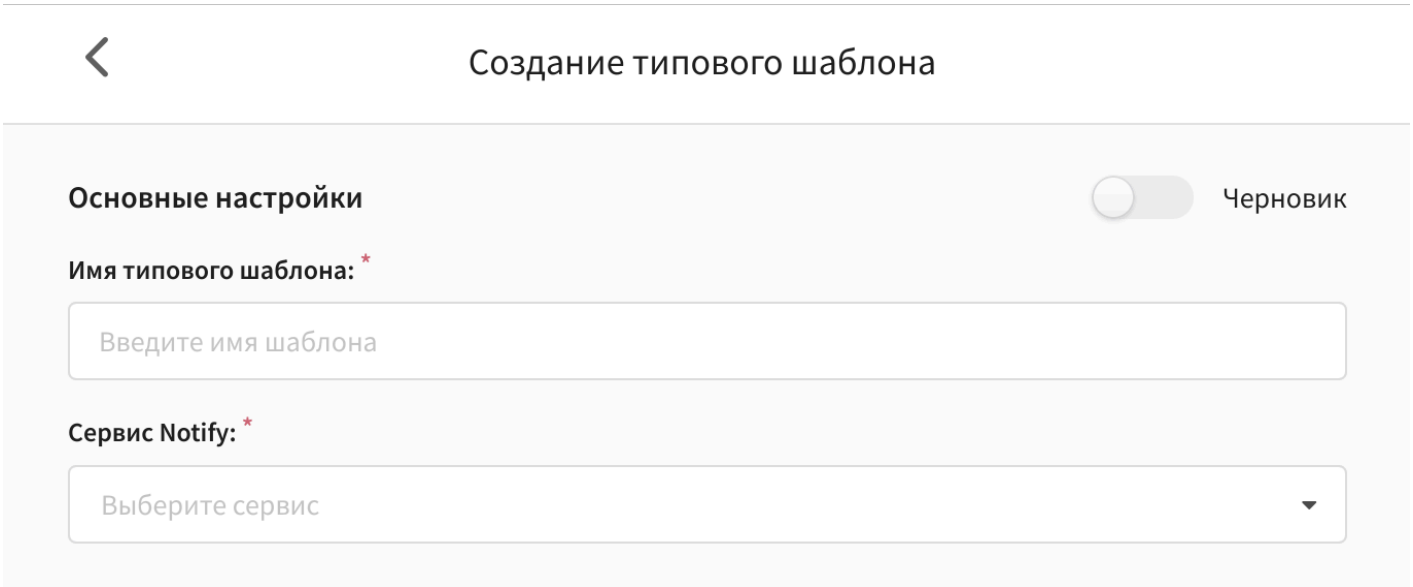


Шаблоны, которые были перенесены в архив или не прошли модерацию, будут содержать предупреждение. Их нельзя будет использовать, пока вы не отредактируете их и они не будут одобрены:



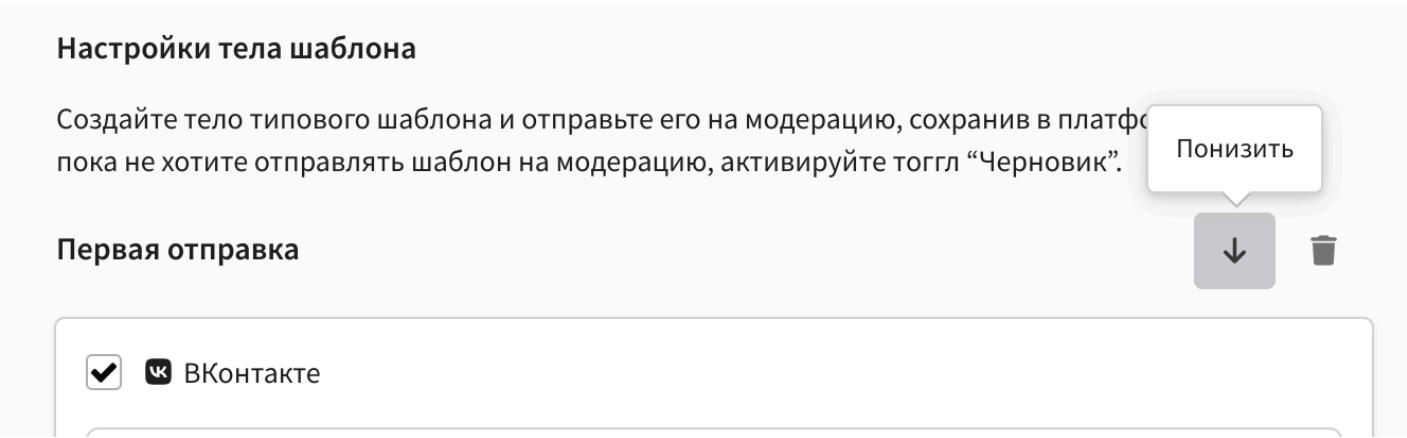
Создание типового шаблона

Вы можете также создать новый шаблон Notify, находясь в платформе. Для создания шаблона нажмите соответствующую кнопку. Откроется окно создания шаблона, в этом окне необходимо указать **имя** типового шаблона и настроенный в личном кабинете **сервис**:

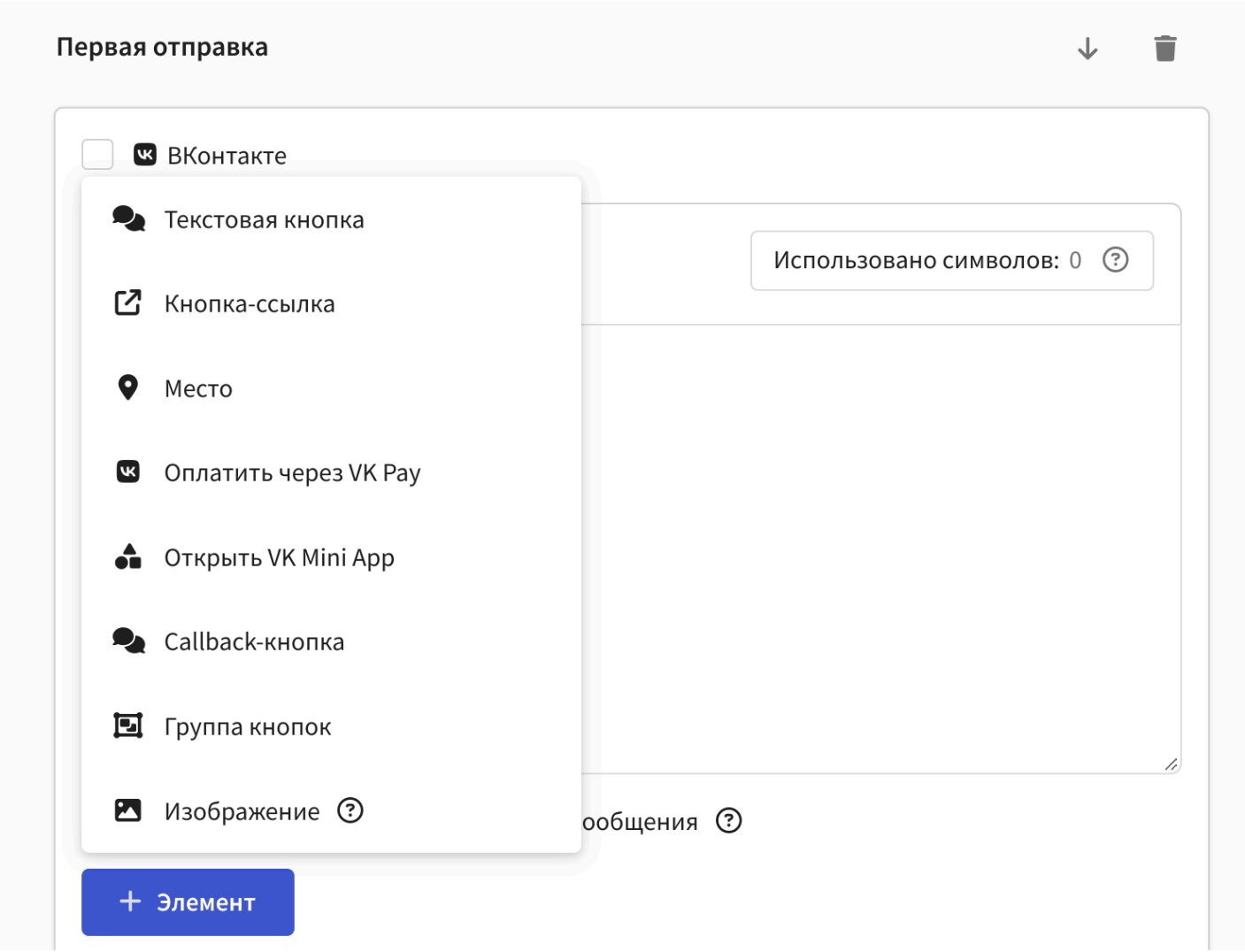


В шаблоне существует возможность задать **время ожидания доставки и прочтения сообщения**. Время ожидания доставки — время, в течение которого Notify будет пытаться повторно отправить сообщение, если произошла какая-либо ошибка. Когда это время истечет, будет запущена вторая отправка. Время ожидания прочтения — время, в течение которого Notify будет ожидать прочтения сообщения в “ВКонтакте” или “Одноклассниках”. Когда это время истечет, будет запущена отправка с SMS-сообщением.

Шаблон Notify состоит из **отправок**. Одна отправка — сообщение в “Вконтакте” или “Одноклассниках”, другая отправка — SMS на номер телефона пользователя. Вы можете добавить или удалить отправку, а также изменить порядок их отправок, нажав на соответствующую кнопку:



Тело сообщения определяет содержимое вашего уведомления. Сообщение в “Одноклассниках” может состоять **только из текста**. К сообщению “Вконтакте” можно прикрепить элемент:



- **Текстовая кнопка** — отправляет сообщение с текстом. Для текстовой кнопки вы можете выбрать цвет, а также приложить дополнительную информацию в формате JSON.
- **Кнопка-ссылка** — открывает указанную ссылку. В кнопке можно указать её текст, ссылку для перехода, а также дополнительную JSON-информацию.

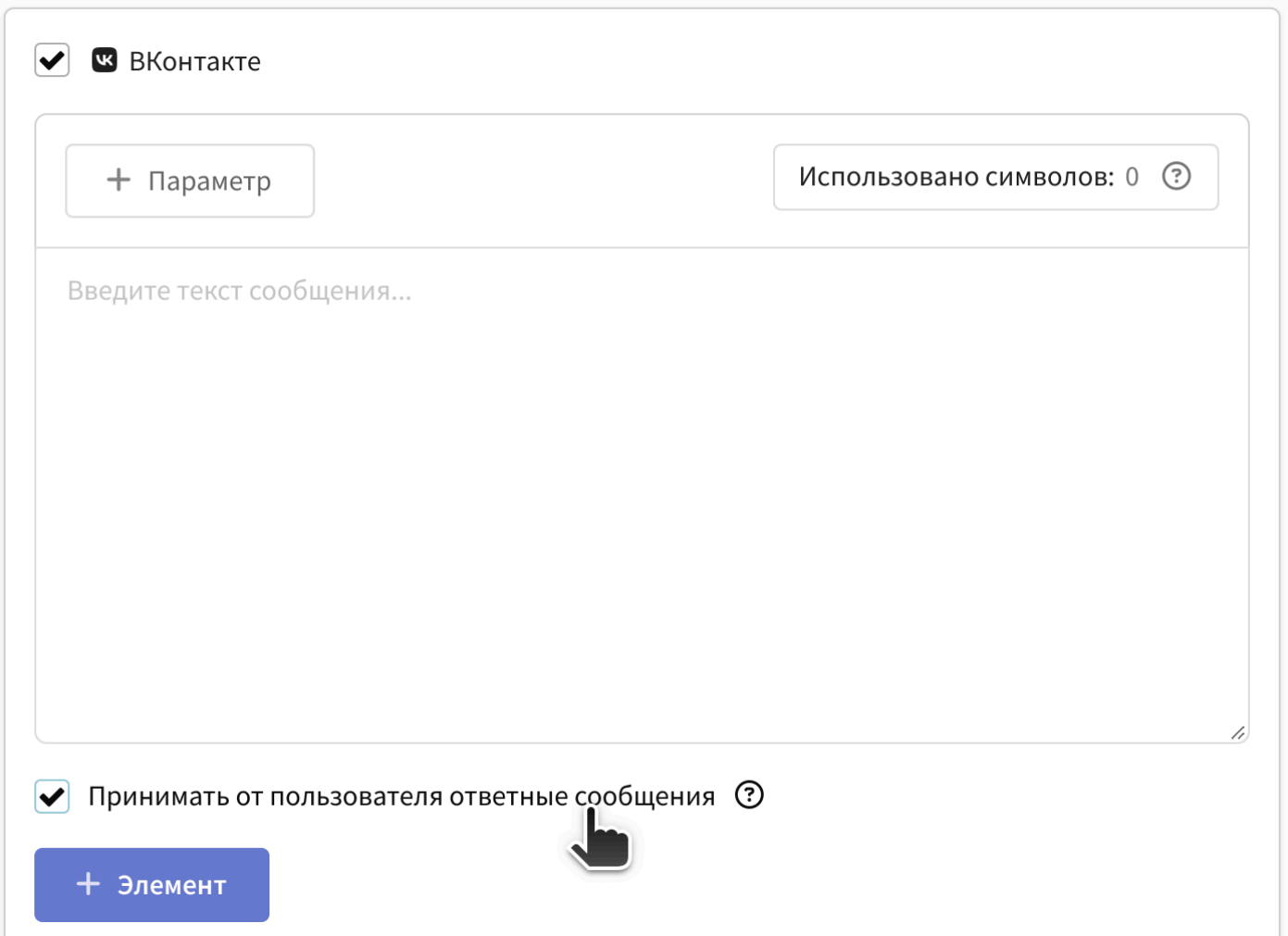
- **Место** — после нажатия открывает диалоговое окно с информацией о местоположении. Пользователь может выбрать своё местоположение в этом окне и отправить его в чат. Администраторы сообщества получат изображение карты в чате.
- **Оплатить через VK Pay** — открывает окно оплаты VK Pay с предопределёнными параметрами. Для кнопки необходимо указать хеш — строку, содержащую параметры платежа и идентификатор приложения.
- **Открыть VK Mini App** — открывает указанное приложение VK Mini Apps. Для настройки кнопки необходимо указать ID приложения, Owner ID, а также хеш для навигации в приложении.
- **Callback-кнопка** — позволяет без отправки сообщения от пользователя получить уведомление о нажатии на кнопку и выполнить необходимое действие. Для кнопки можно указать надпись и цвет.
- **Группа кнопок** — кобминация из вышеперечисленных кнопок.
- **Изображение** — до 5 изображений общим весом не более 50 МБ. Поддерживаемые форматы — JPG, PNG, GIF.

TIP

Подробнее об элементах и их настройке читайте в документации Notify.

В тело сообщения можно вставить **параметры**. В дальнейшем вы сможете подставить в них значения полей из баз данных платформы.

Также вы можете дать возможность клиенту отправлять ответные сообщения в чате с компанией. Для этого активируйте в настройках пункт **“Принимать от пользователя ответные сообщения”**:



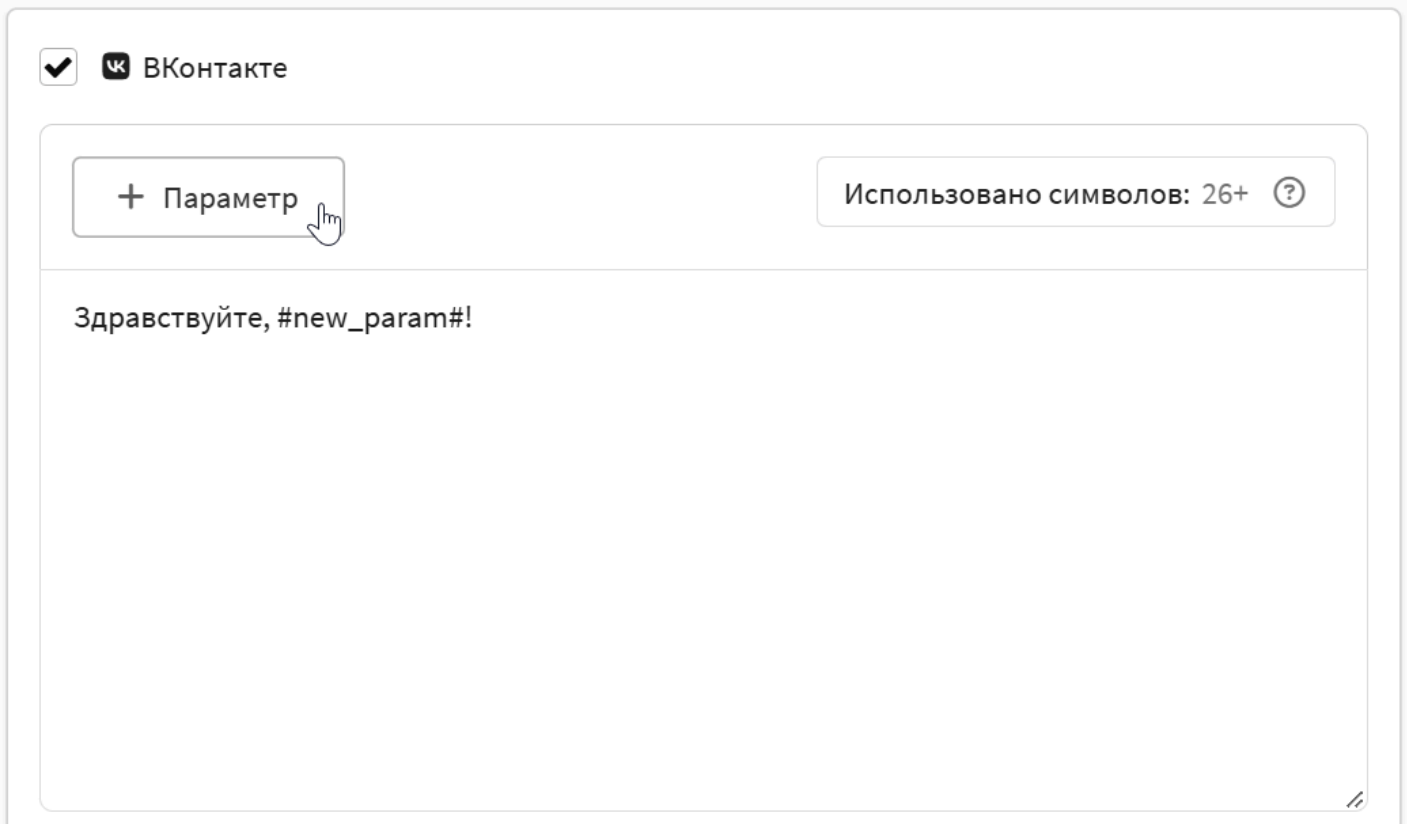
Существует несколько типов сообщений для SMS-отправок — рекламный, идентификационный, сервисный и транзакционный. Тип сообщения влияет на его тарификацию в Notify, подробную информацию об этом вы можете узнать в поддержке сервиса.

После сохранения шаблона он отправится **на модерацию** в сервисе. Если вы не хотите отправлять его на модерацию, активируйте переключатель **“Черновик”**.

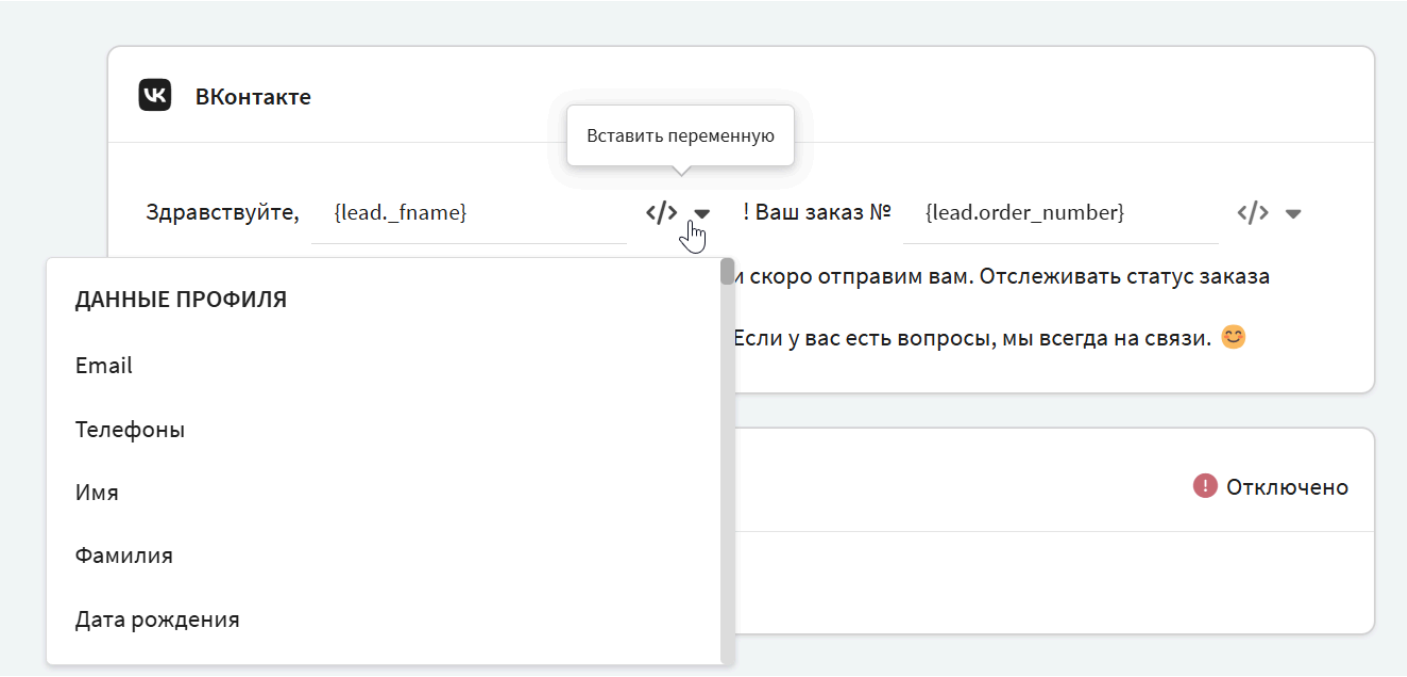
Работа с параметрами

Текстовый блок в Notify состоит из фиксированного текста и параметров. Параметры – это текст шаблона, который меняется для каждого нового сообщения (имя, адрес, название мероприятия, дата и время, ссылка и т.д.).

Параметр может быть добавлен в текстовый блок двумя способами: вручную, введя `#parameter_name#`, где `parameter_name` — это произвольное название параметра, или с помощью кнопки **+ Параметр**, которая автоматически вставит его в текст:



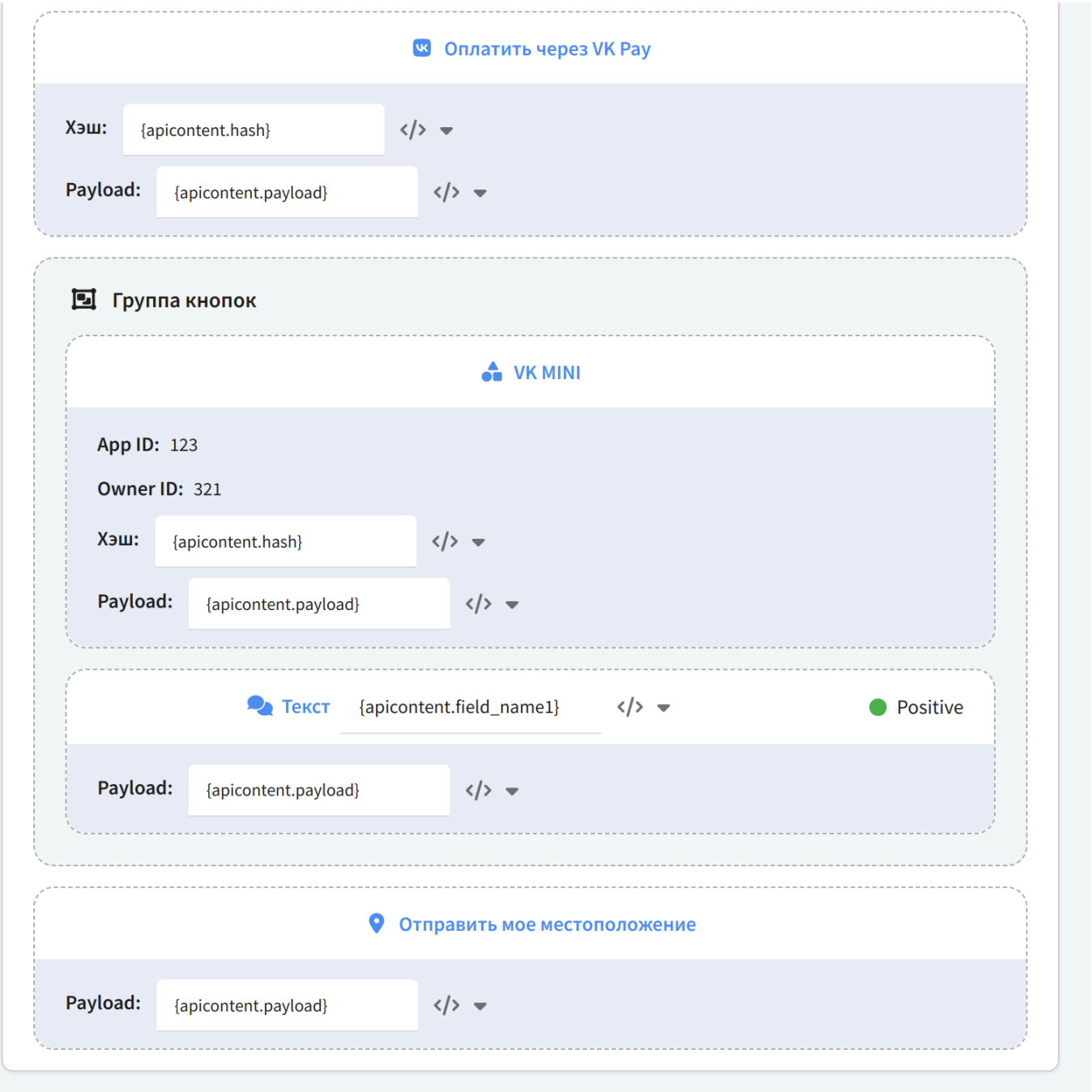
Вы можете вручную прописать значения параметров или добавить переменные Altcraft. Чтобы открыть список переменных, нажмите `</>`. При отправке сообщения введённый текст автоматически подставится в шаблон.



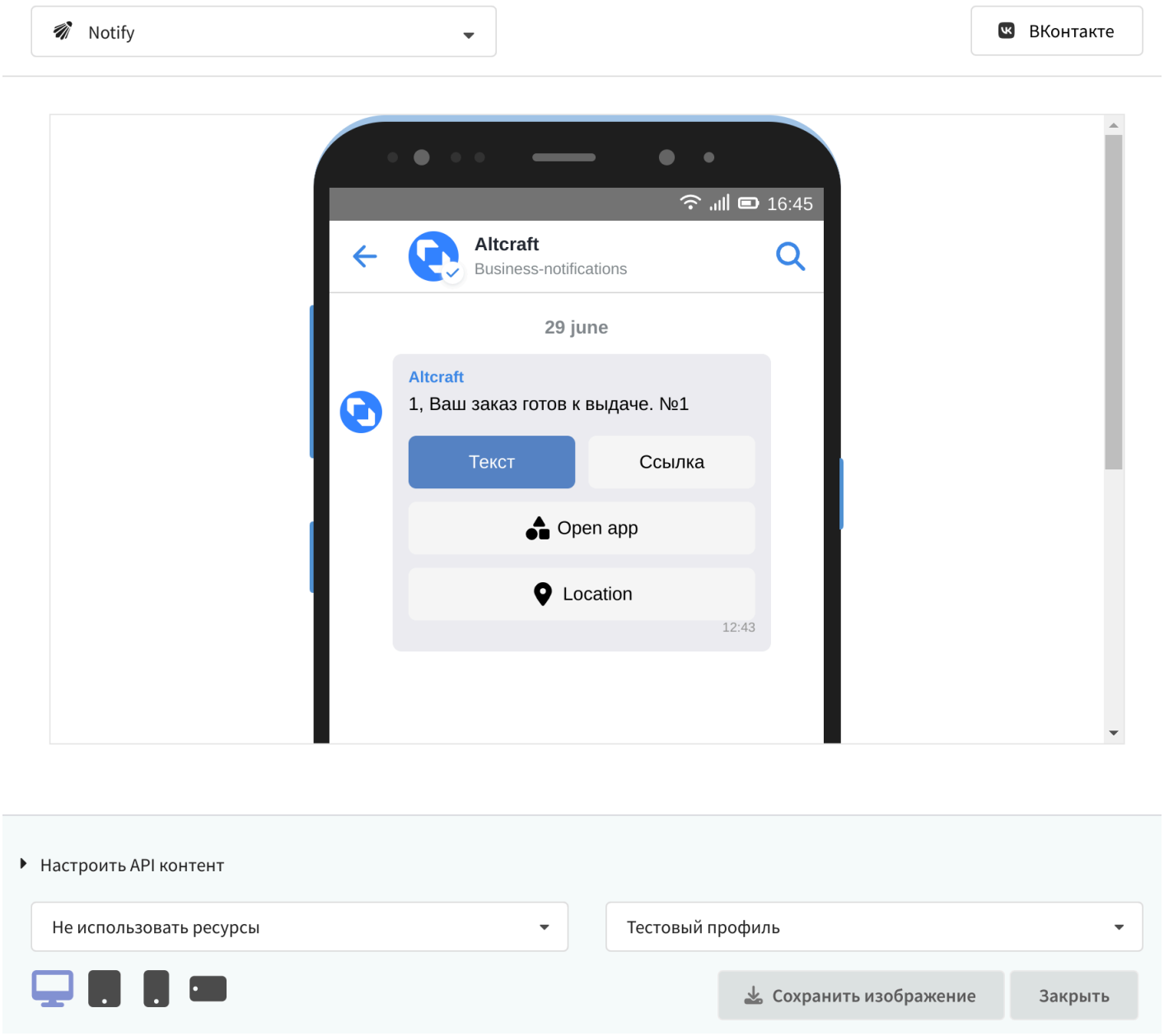
Параметры могут быть использованы не только в текстовом блоке Notify, но и в **элементах**, например, в кнопках:

Предпросмотр шаблона

После выбора одобренного типового шаблона вы можете указать для параметров необходимые **значения** либо **переменные из платформы**:



Для того чтобы узнать, как пользователь увидит полученное сообщение, нажмите кнопку **предпросмотра** в верхней части экрана:



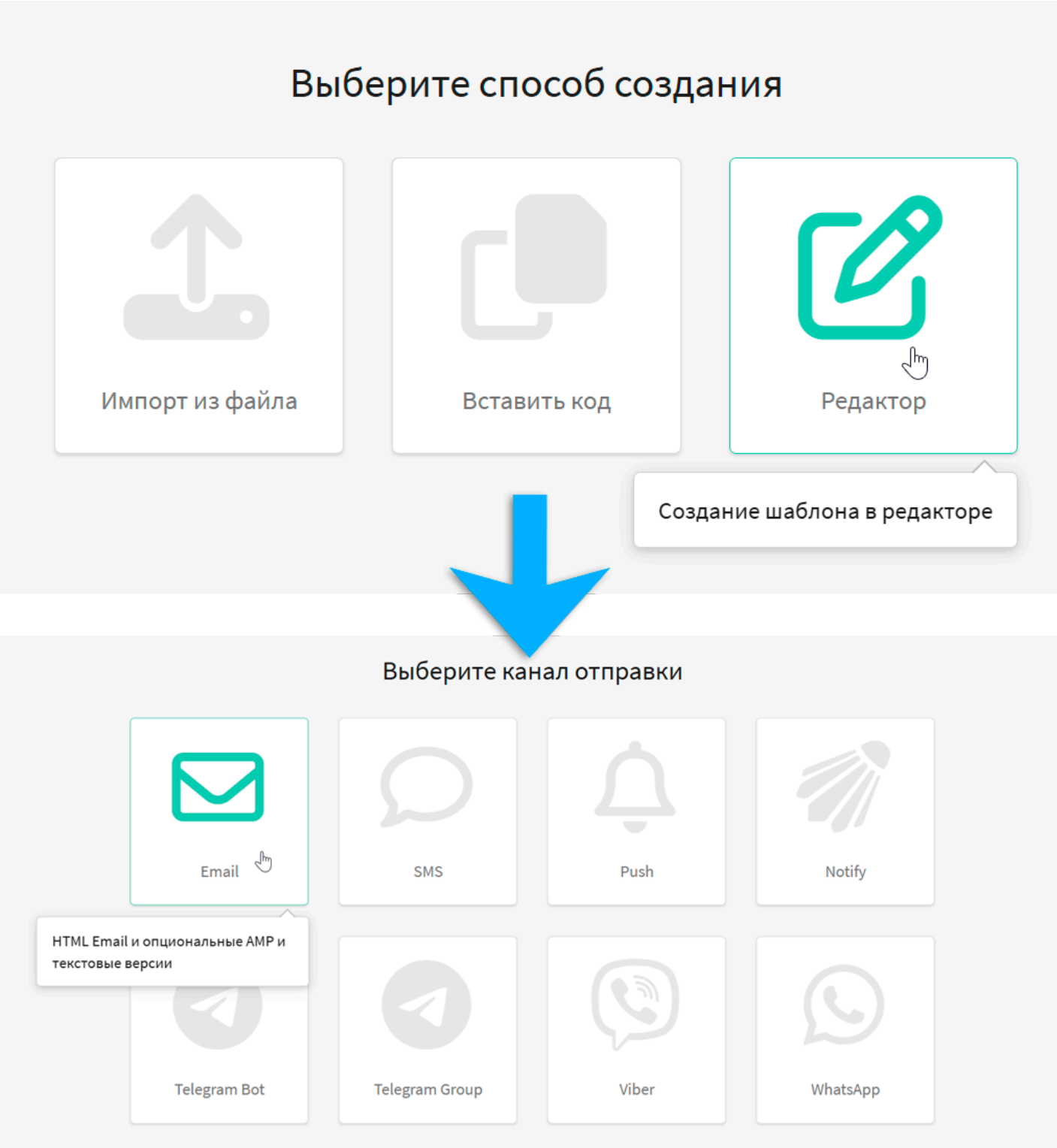
Не забудьте **сохранить** шаблон в платформе после внесения всех изменений. Теперь вы можете использовать этот шаблон для рассылки по каналу Notify.

Визуальный редактор для email-шаблона

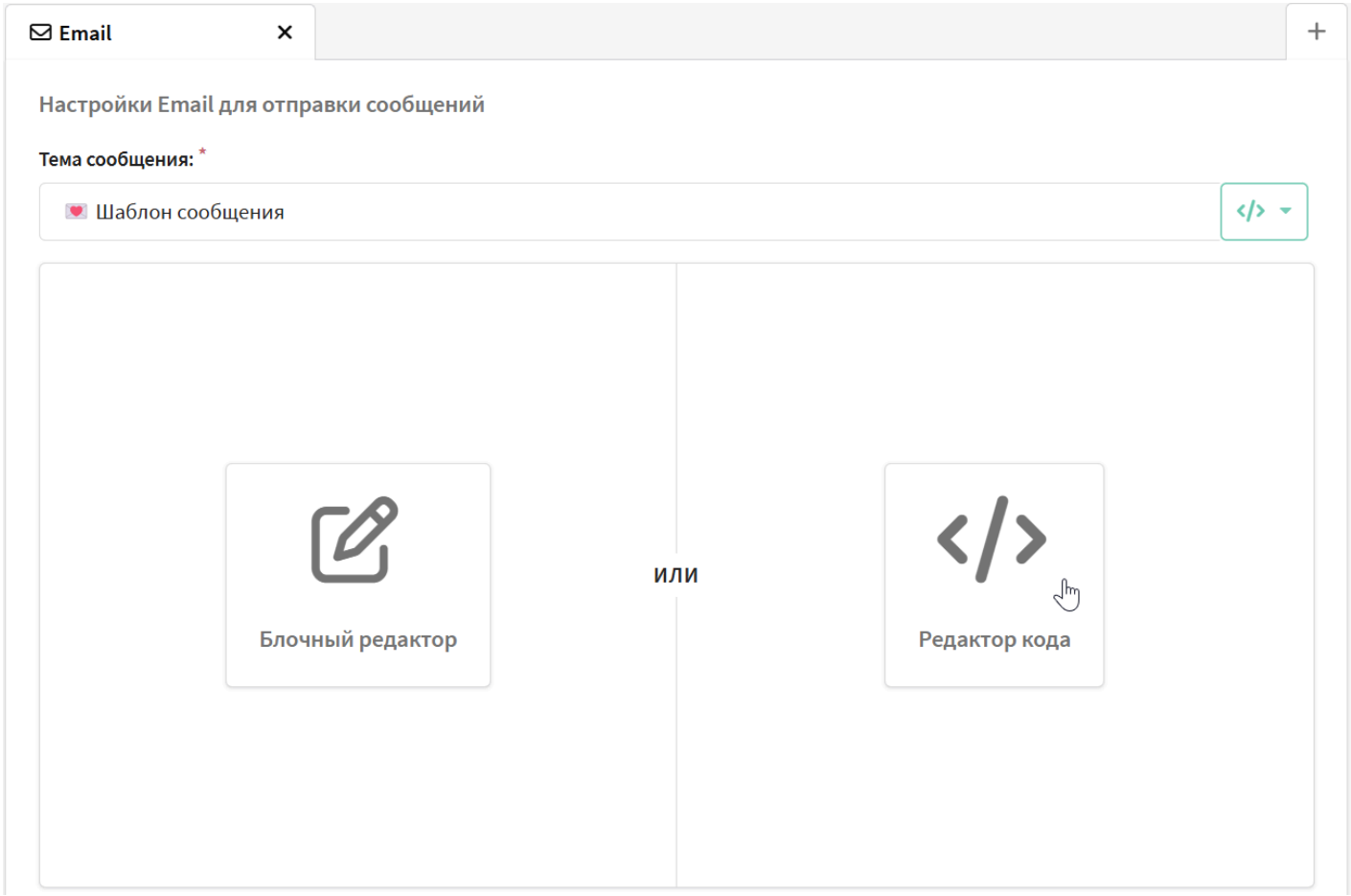
Визуальный редактор для email-шаблона — это инструмент для создания, редактирования и настройки маркетинговых email-шаблонов, который значительно упрощает процесс разработки красивых и функциональных писем. Благодаря удобному интерфейсу, пользователи могут без труда адаптировать дизайн шаблонов под свои нужды, даже без знаний в области программирования. Среди ключевых преимуществ редактора — возможность мгновенного предварительного просмотра, адаптивный дизайн для различных устройств и широкий выбор готовых элементов, которые можно легко комбинировать.

Как начать работу в редакторе?

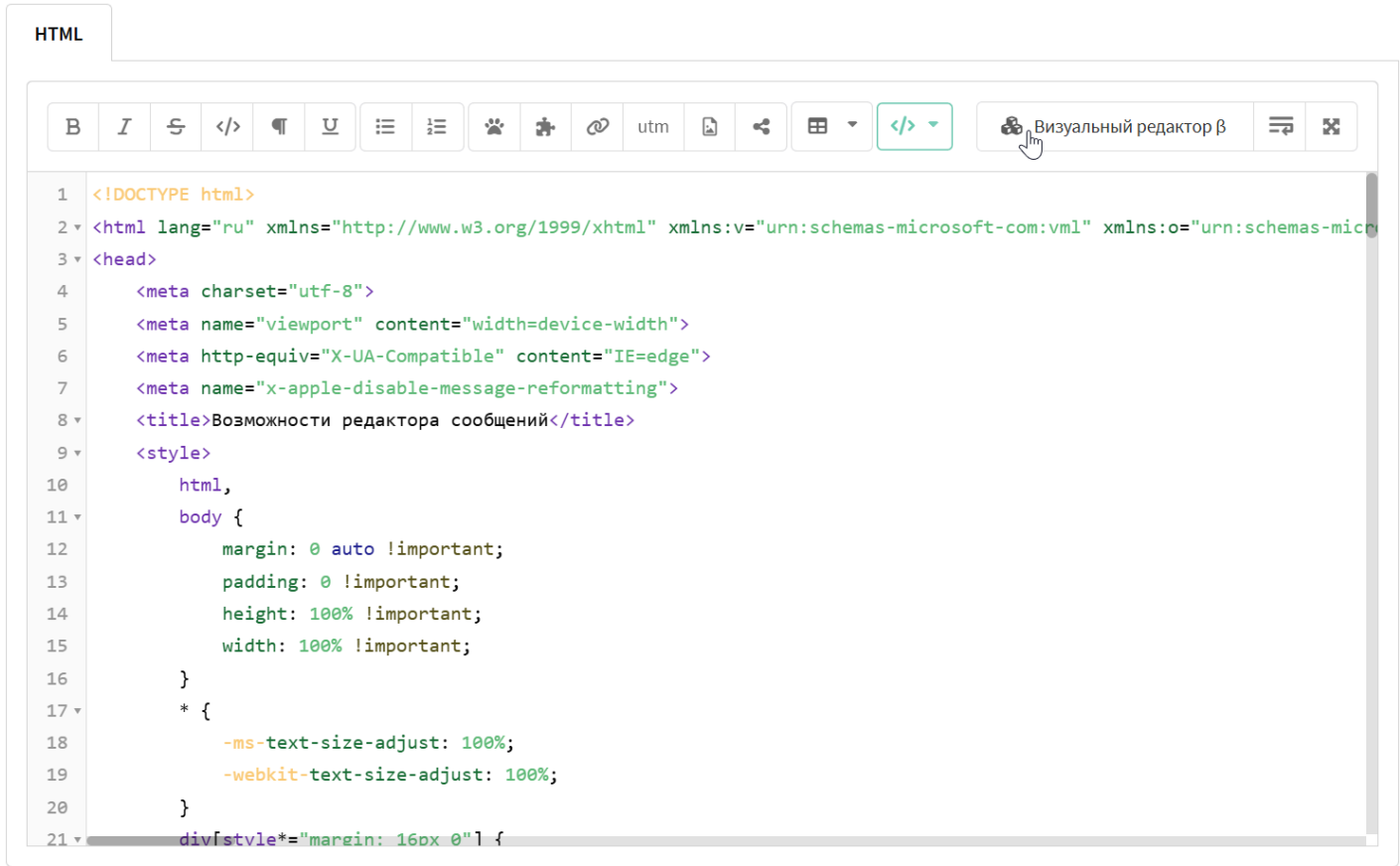
В главном меню откройте раздел **Шаблоны** и создайте новый шаблон сообщения. Выберите **Редактор —> Email**.



Перед вами появится окно настройки email-шаблона, в нём выберите **“Редактор кода”**.



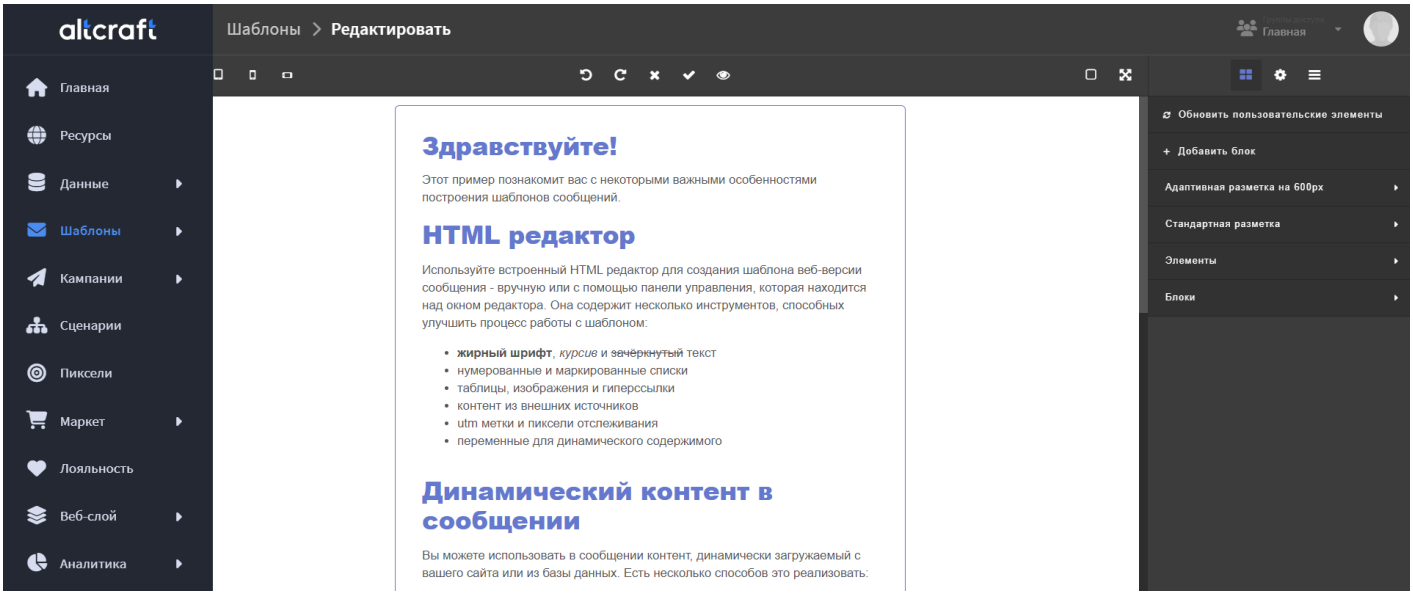
Вы окажетесь в окне редактирования HTML—кода, в нём нажмите на кнопку **“Визуальный редактор”**.



Подробнее о работе с визуальным редактором шаблонов читайте в следующих статьях:

Интерфейс редактора

Редактор разделяет страницу на две части. В центральной части вы видите свой шаблон. Здесь вы можете выделять, перемещать и редактировать блоки. В правой боковой части находится панель, где вы можете работать с элементами шаблона: выбирайте нужные элементы и изменяйте их параметры.



Над центральной частью редактора находится панель инструментов.

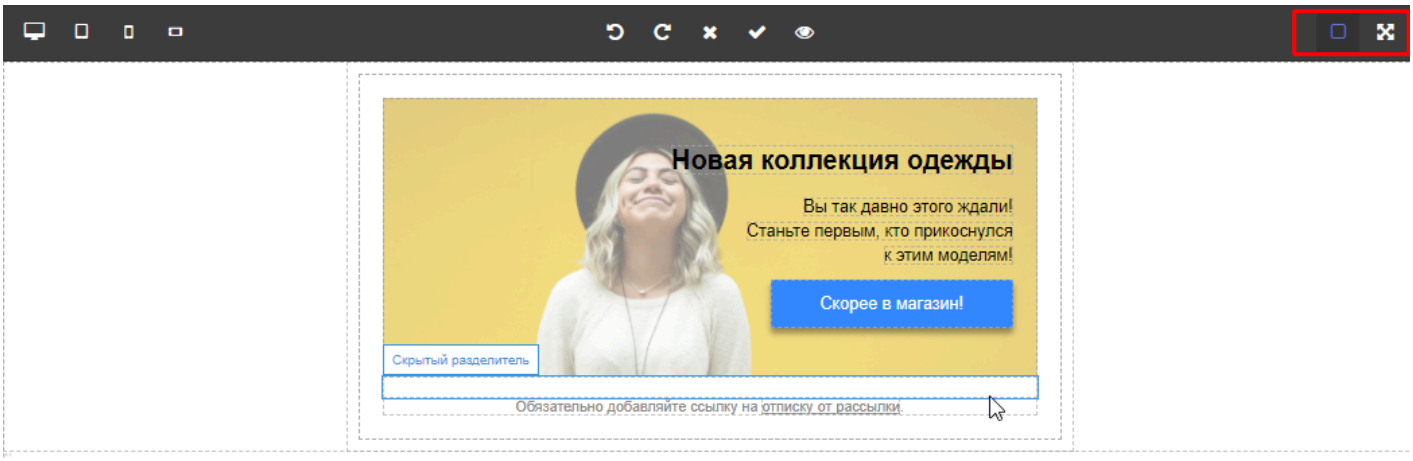
В левой части панели вы можете изменить **режим отображения контента** для предпросмотра на экране компьютера, планшета или смартфона. Таким обраом, у вас есть возможность проверить адаптивность шаблона прямо в редакторе:

Кнопки **в центральной части** панели инструментов позволят вам:

- Отменить или повторить последнее действие.
- Отменить или сохранить изменения в шаблоне.
- Открыть предпросмотр сообщения с возможностью выбора профиля для персонализации.

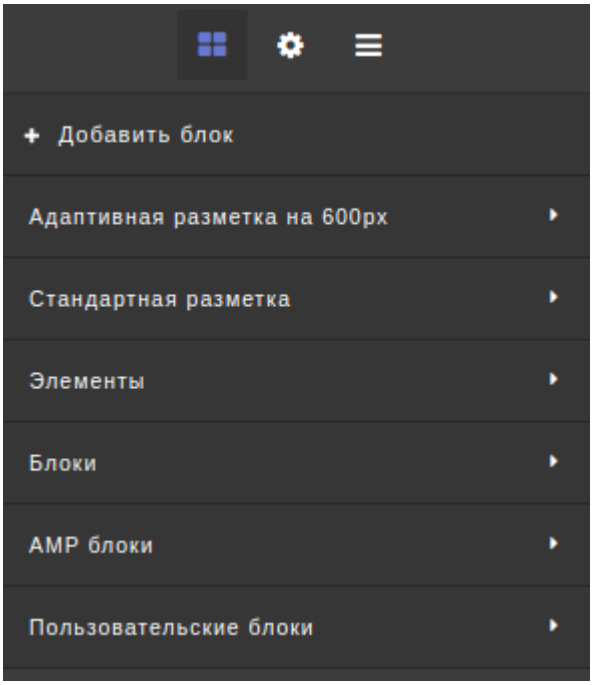
В правой части находятся:

- кнопка отображения границ элементов разметки HTML
- кнопка включения полноэкранного режима редактирования:



Правая панель редактора содержит три вкладки:

1. Добавление элемента
2. Стили элемента
3. Структура элементов



💡

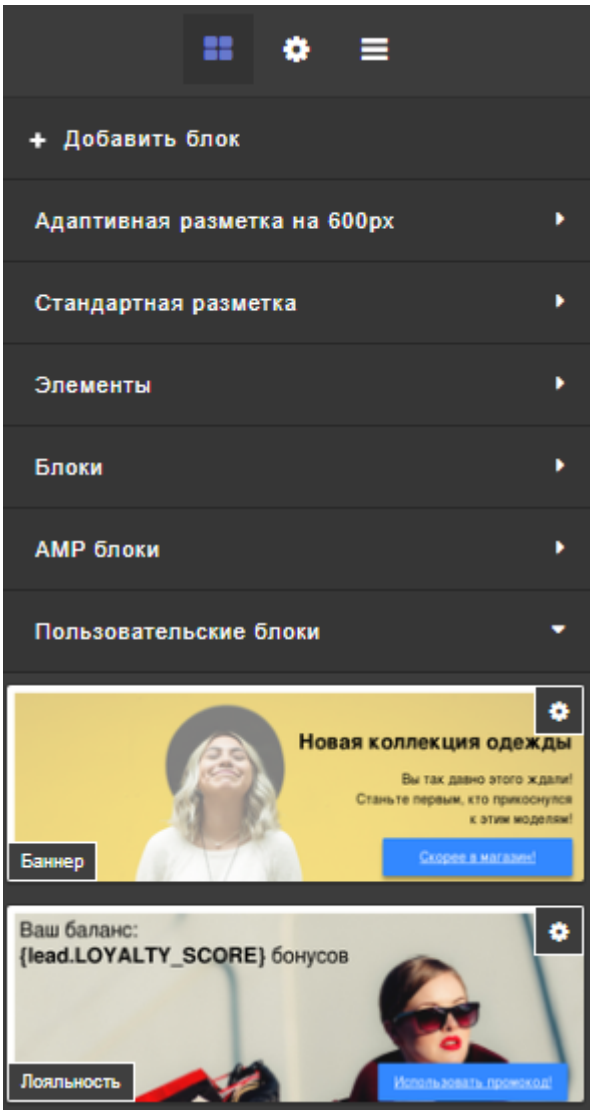
Наш следующий шаг — научиться добавлять элементы в шаблон.

💡

Добавление элементов

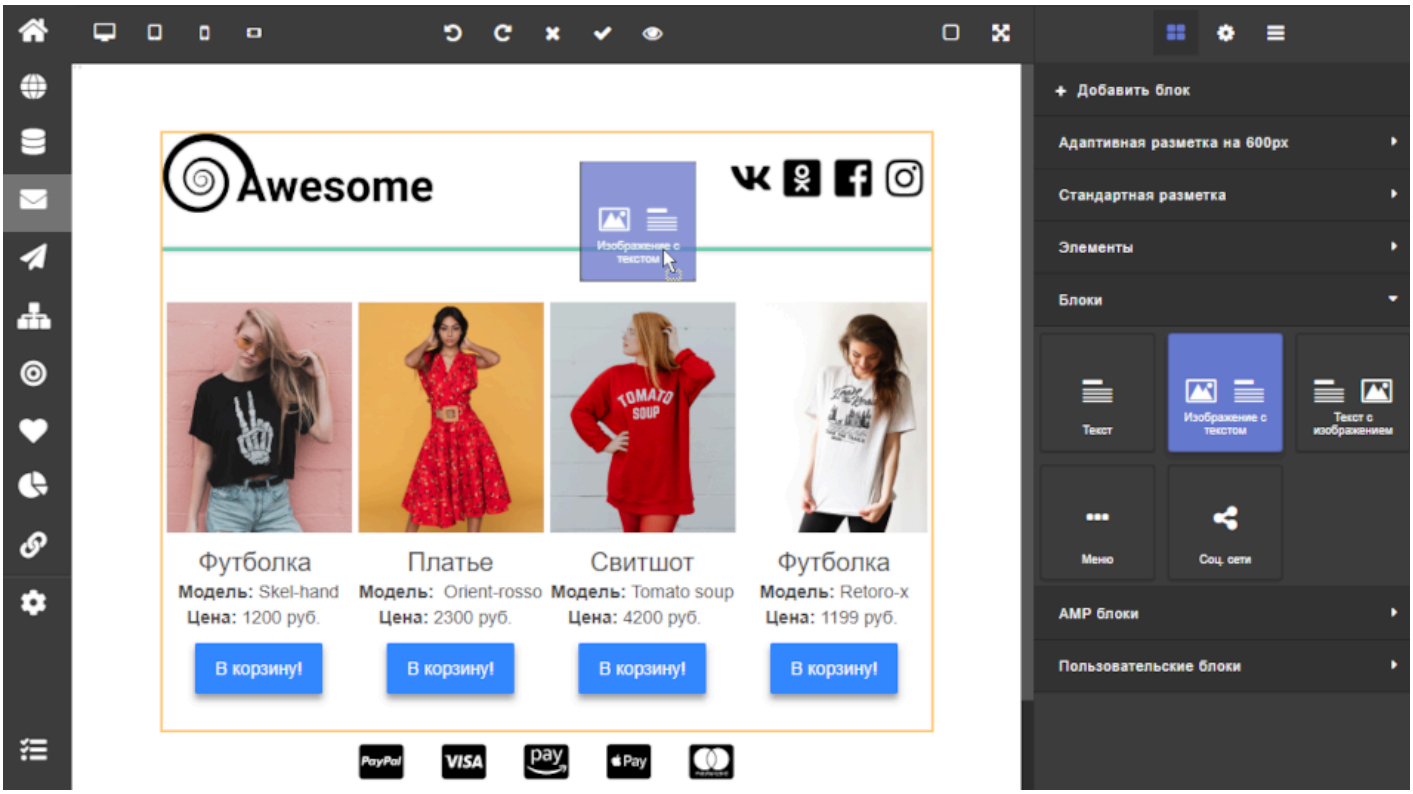
Какие элементы вёрстки Email письма вы можете добавлять в шаблон?

Элемент	Применение
Адаптивная разметка на 600px	Набор строк таблицы, разделённых на несколько ячеек с фиксированной максимальной шириной. Необходим для разделения секции шаблона на несколько колонок. Используется для основных секций Email шаблона со стандартной шириной в 600 пикселей.
Стандартная разметка	Набор строк таблицы, разделённых на несколько ячеек в процентном соотношении (без ограничения ширины). Используется внутри блоков, ограниченных шириной тела шаблона, а также для секций во всю ширину экрана.
Элементы	Отдельные элементы вёрстки, такие как заголовок, текст, разделитель, баннер, видео, ссылка или кнопка.
Блоки	Наборы элементов для быстрого создания частей шаблона. Например, текст с заголовком или текст с заголовком и изображением, меню или ссылки на социальные сети.
AMP блоки	Блоки для интерактивных AMP Email сообщений. Поддерживаются “карусель” и “аккордеон”.
Пользовательские блоки	Вы можете сохранять элементы и целые секции шаблона как пользовательские блоки. Они будут доступны в визуальном редакторе для каждого шаблона — так же, как фрагменты сообщений. Чтобы быстро перенести фрагмент в пользовательские блоки визуального редактора, в верхней части этого меню нажмите **+ Добавить блок** и скопируйте в открывшуюся форму содержимое фрагмента. Подробнее о пользовательсих блоках читайте в этой статье.

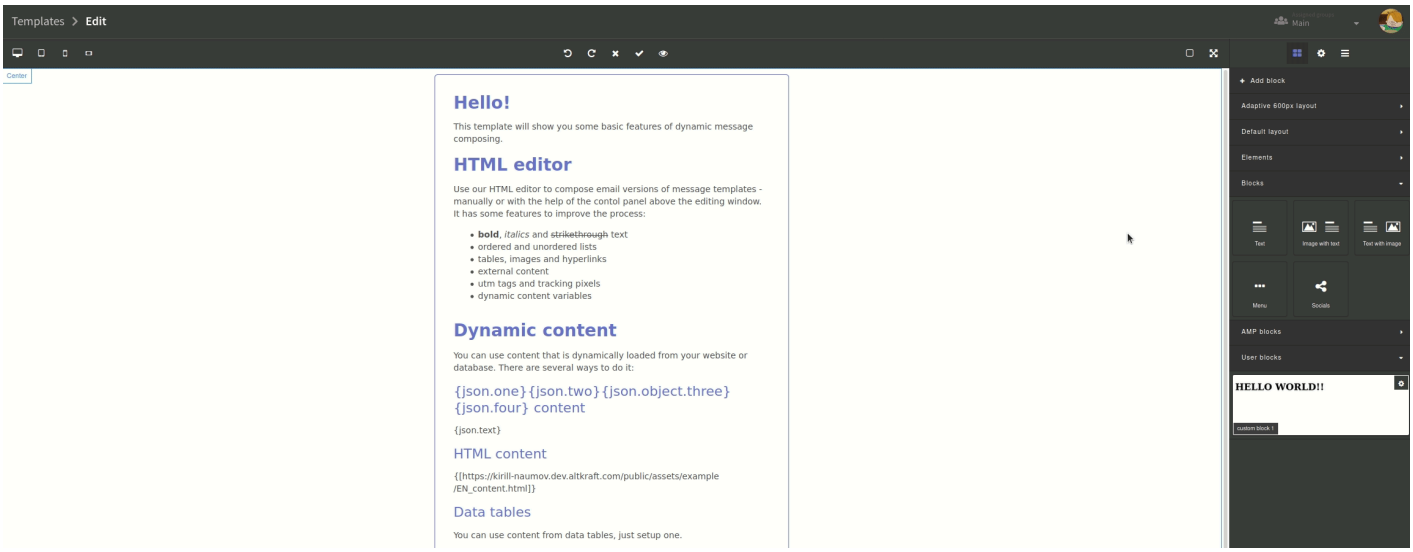


Как добавить элемент в шаблон?

Чтобы добавить блок в шаблон, просто нажмите на него мышкой и перетащите на нужное место. Редактор подсветит жёлтым контуром родительский блок письма, а зелёной линией — место расположения элемента:



GIF “Как добавить элемент?”



Как работать с элементами?

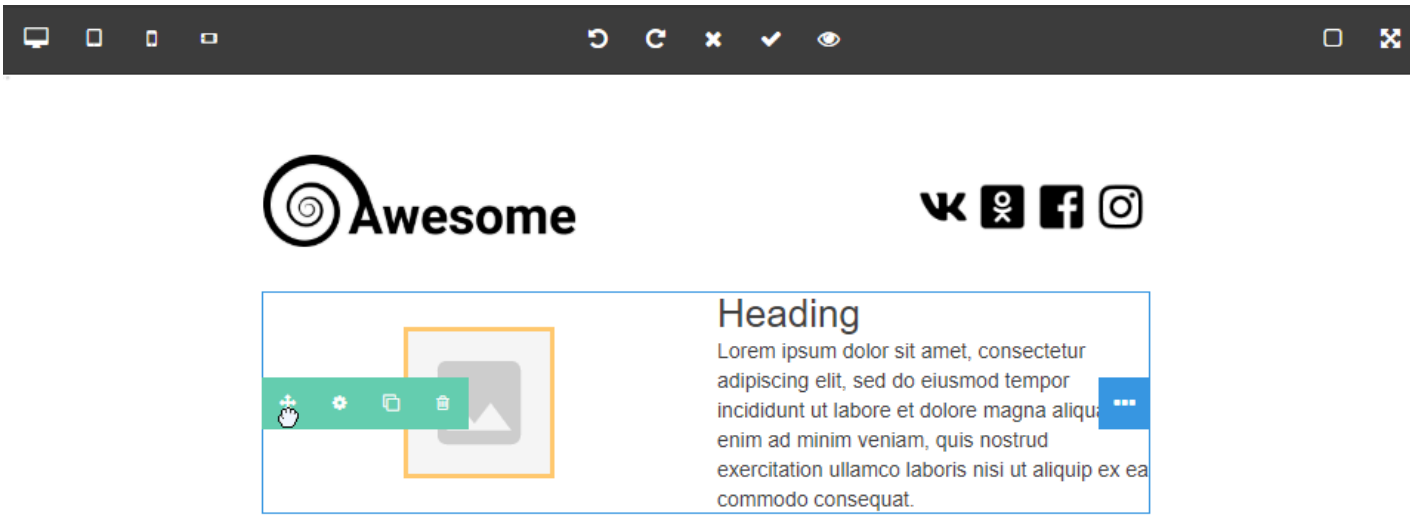
Контент сообщения в визуальном редакторе представлен в виде **элементов** (текст, заголовки, изображения, видео) и **HTML разметки** (ячейки таблиц, в которые, как в контейнеры, помещены элементы).

Основные операции над блоками разделены на два меню:

- меню операций над **контейнерами** — вызывается нажатием на зелёную иконку в левой части контейнера разметки;
- меню операций над **элементами** — вызывается нажатием на синюю иконку в правой части элемента контента или секции письма.

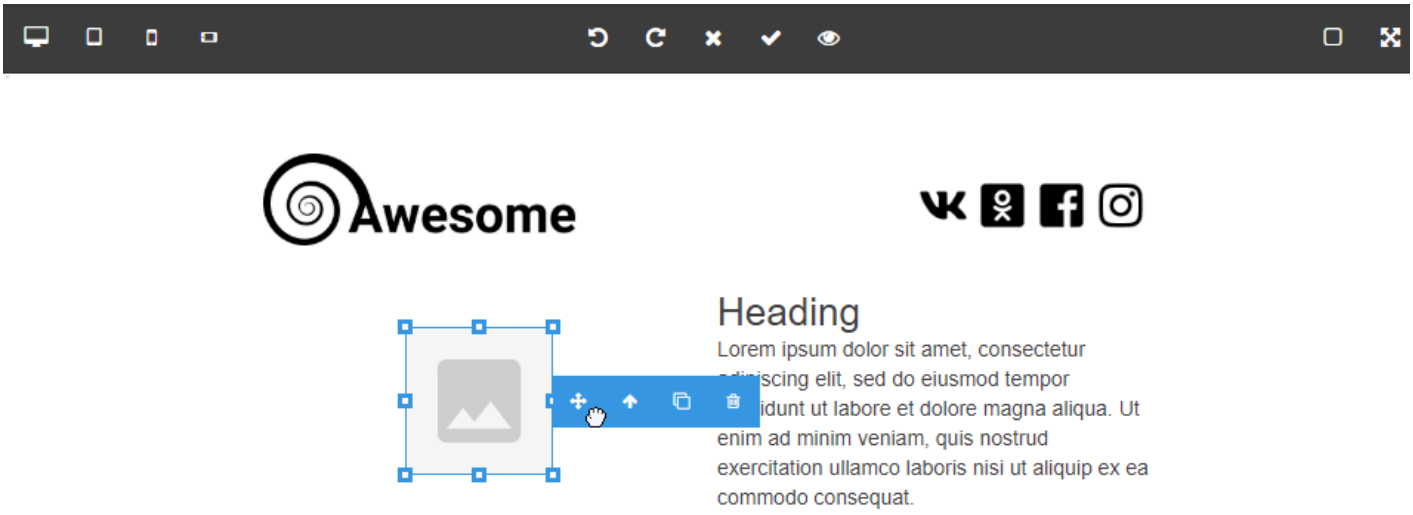
Когда вы добавляете в шаблон ячейку таблицы или другой **контейнер**, в центральной части редактора для него будет доступно меню операций. Выделите необходимую ячейку, и вы сможете **переместить**, **настроить**,

дублировать или **удалить** её с помощью кнопок **зелёного цвета** в левой части:



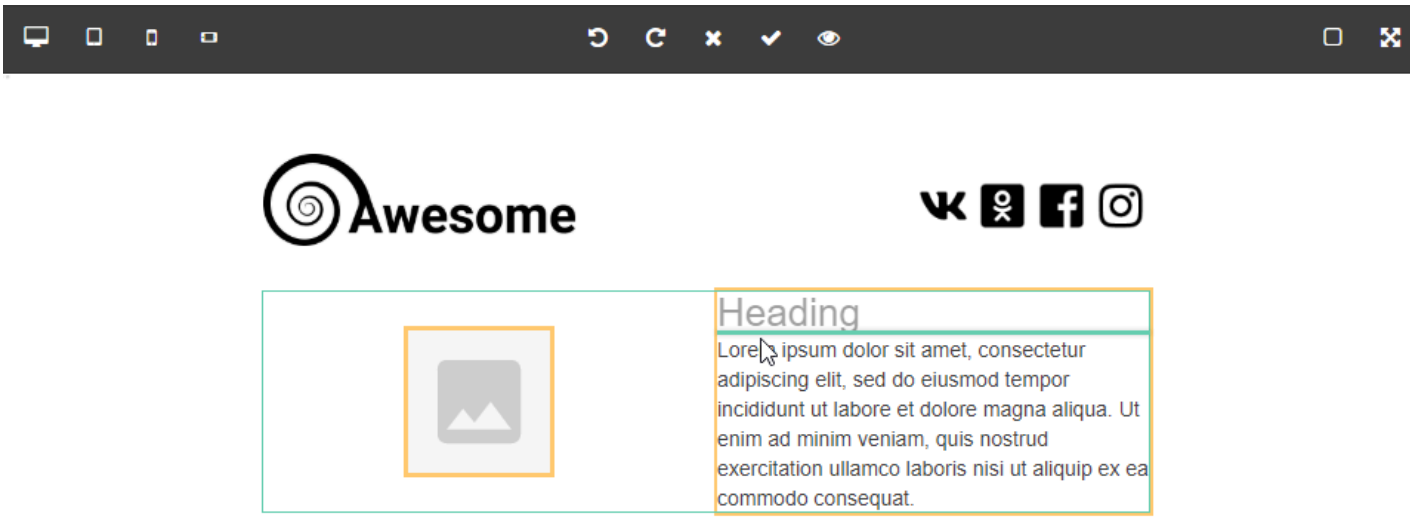
Для каждого **элемента** внутри контейнеров также доступно меню операций. Воспользуйтесь кнопками **синего цвета** в правой части элемента, чтобы:

- переместить элемент,
- вынести его в контейнер следующего уровня,
- дублировать,
- удалить.



При перемещении элемента **зелёным цветом** подсвечивается секция письма, **жёлтым** — родительский контейнер перемещаемого элемента и другие элементы секции.

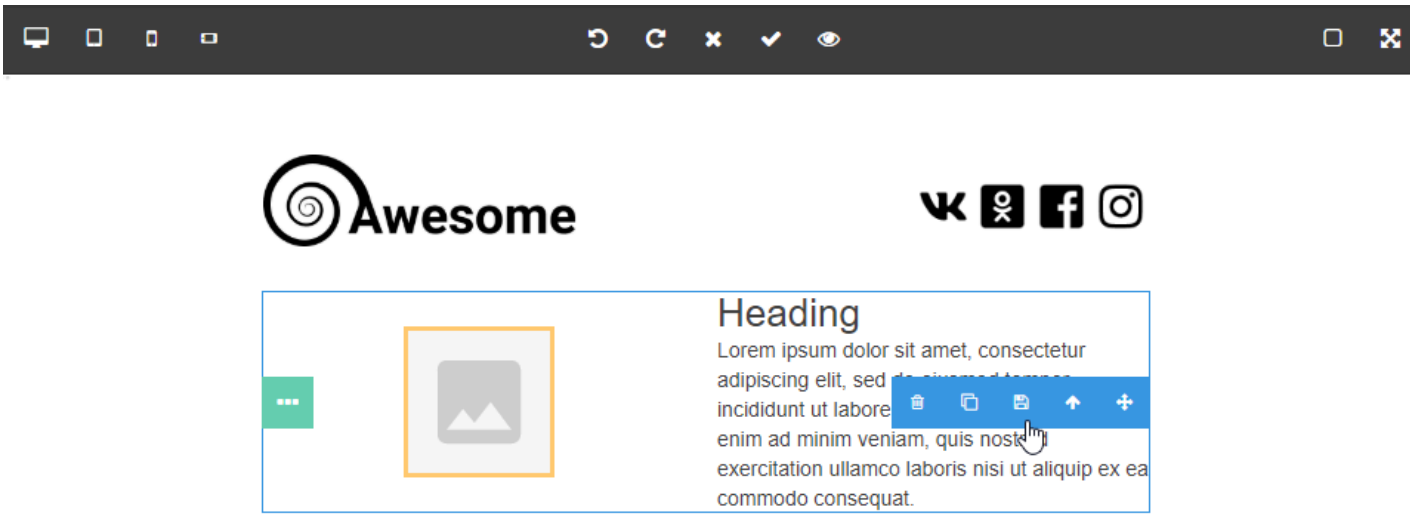
Жирная зелёная линия будет отмечать новое положение элемента или контейнера:



Элементы и группы элементов можно сохранять в виде **пользовательских блоков**. В меню операций с элементом выберите кнопку с **дискетой**.

TIP

Если нужно сохранить содержимое целой секции, выделите её, нажав на кнопку вызова меню **операций с контейнером** в левой части. После этого вызовите в правой части меню операций с элементом.



TIP

В правой части редактора для каждого выделенного элемента доступно меню стилей, в котором вы можете:

- Настроить атрибуты и параметры элемента. Параметры описаны для каждого типа элементов
- Настроить параметры отображения элемента. Стили CSS добавляются непосредственно в тэги разметки, что улучшает читаемость письма клиентами почты. Стили описаны в отдельной статье.

💡

Теперь давайте подробнее разберем каждый элемент.

💡

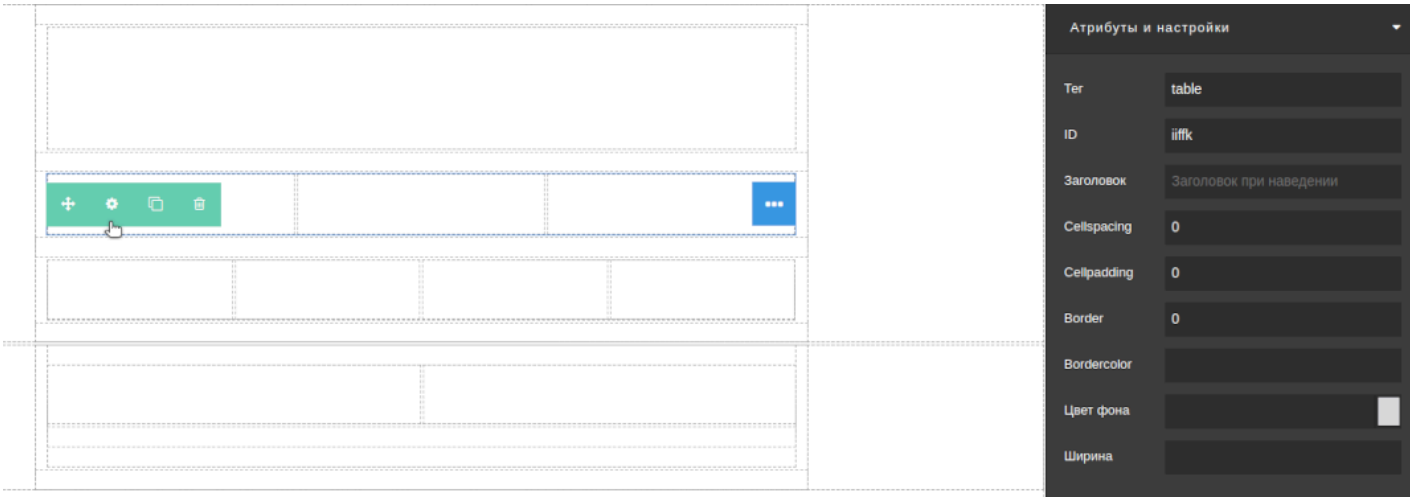
Элементы и их настройки

Адаптивная разметка

При добавлении элементов адаптивной разметки для них будут доступны следующие **атрибуты и настройки**:

- HTML **тег**
- **ID** — идентификатор элемента
- **Заголовок** при наведении
- **Cellspacing** — расстояние между ячейками
- **Cellpadding** — внутренний отступ ячеек
- **Border** — толщина границы
- **Border color** — цвет границы
- **Цвет фона**
- **Ширина** секции

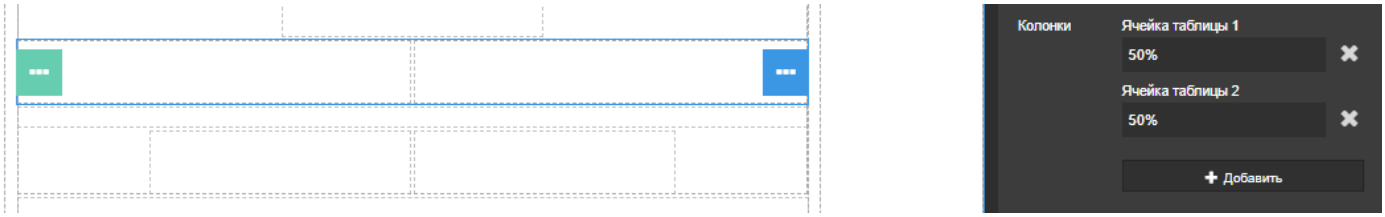
> **TIP**
>
> Чтобы увидеть окно **“Атрибуты и настройки”**, выберите элемент адаптивной разметки, наведите курсор на зеленую иконку в левой части контейнера и в всплывающем окне кликните иконку “Настройки”.



Ячейки адаптивной разметки отображаются как контейнеры, для которых можно установить только идентификатор и заголовок.

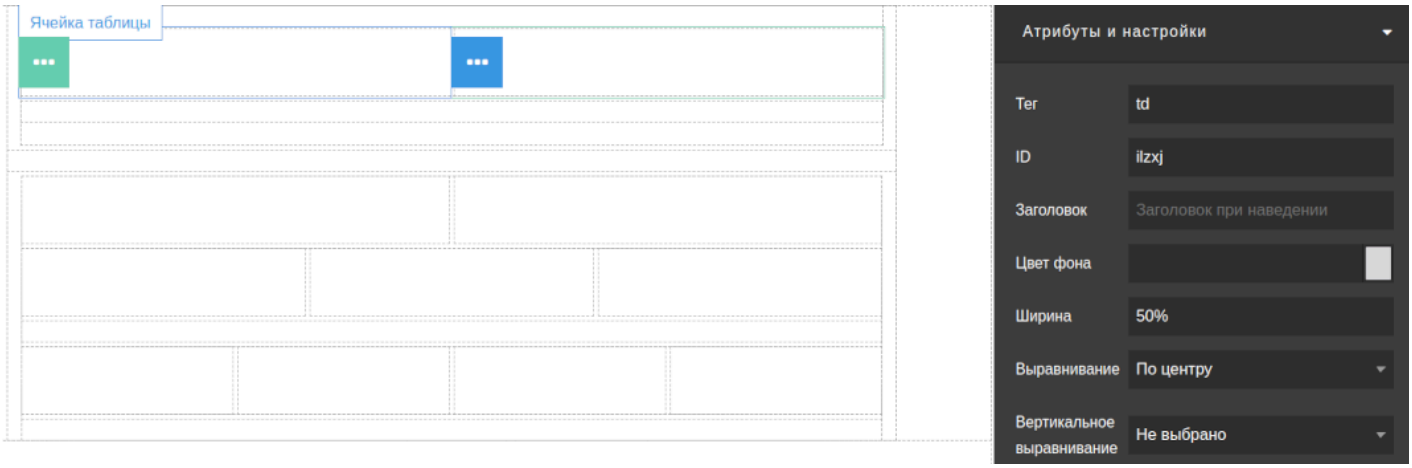
Стандартная разметка

Для элементов **стандартной разметки** также доступно добавление ячеек и изменение процентного соотношения их ширины:



Для ячеек стандартной разметки можно установить:

- **ID** — идентификатор элемента
- **Заголовок** для отображения при наведении курсора
- **Цвет фона**
- **Ширина** контейнера
- **Выравнивание** содержимого по горизонтали и вертикали



Элементы

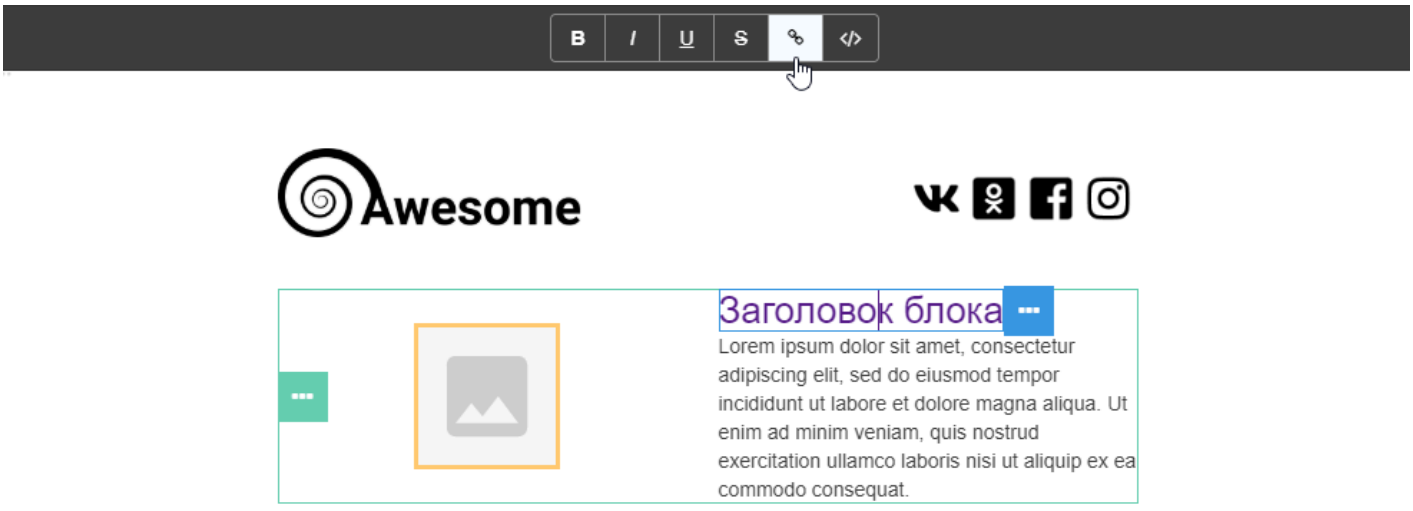
Текст в таких элементах как **заголовок**, **текст, кнопка, ссылка** можно редактировать с помощью **WYSIWYG редактора:**

Редактор WYSIWYG для текстовых элементов

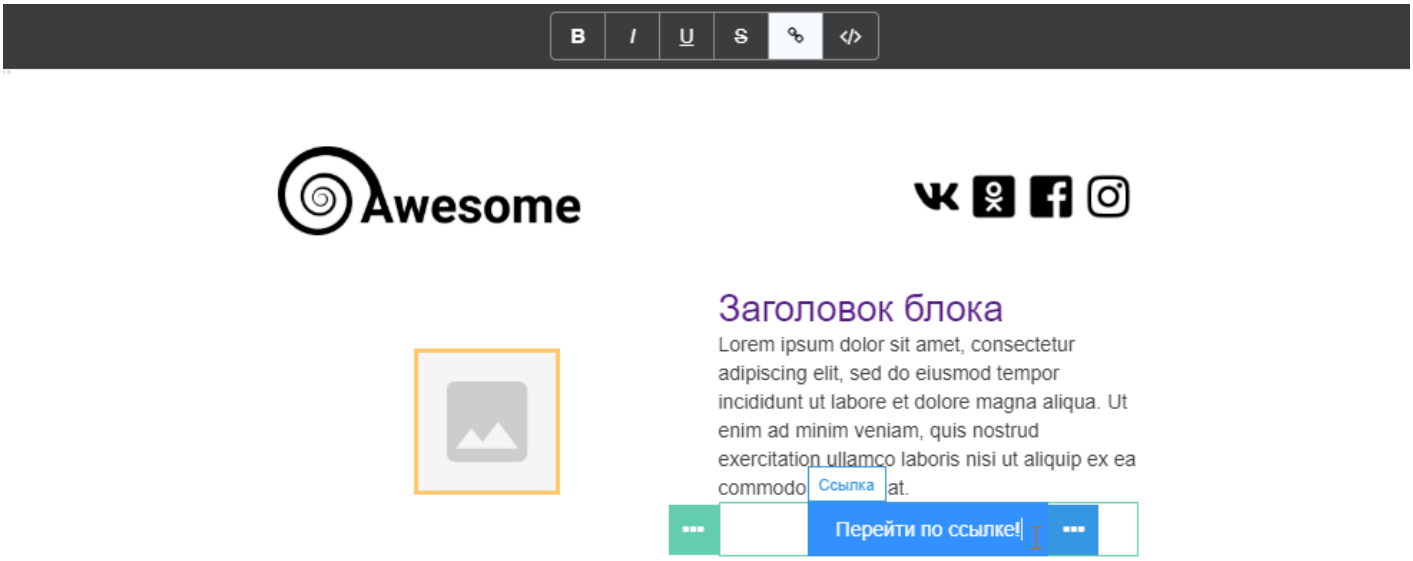
Дважды кликните мышью по текстовому элементу, и панель инструментов в верхней части редактора перейдёт в режим текстового редактора. Вы сможете редактировать текст элемента в центральной части редактора, что сразу позволяет видеть результат.

В панели возможно:

- использовать инструменты форматирования (курсив, жирный / подчеркнутый / зачеркнутый текст);
- добавить ссылки;
- добавить переменные персонализации.



Аналогично вы можете редактировать текст на элементах **Кнопка**:

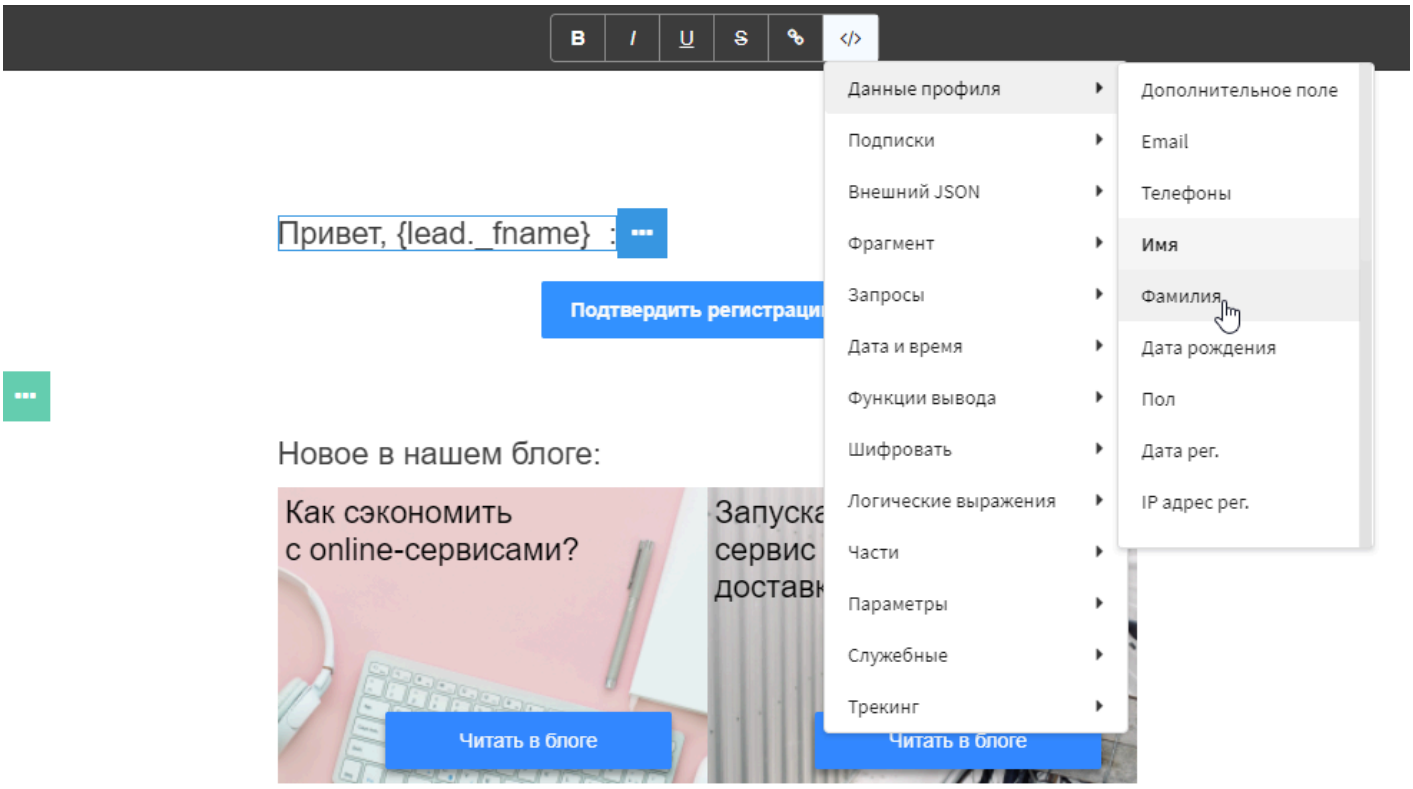


TIP

Параметры отображения текста, а также поведение ссылок и кнопок настраиваются в правой части редактора и описаны в [этой статье](#).

С помощью меню переменных вы можете добавить в шаблон [данные профиля и логические условия для персонализации контента](#):

Для подстановки данных **кастомных полей** профиля используется выражение {lead.custom_field} с полным названием [пользовательского поля данных](#).

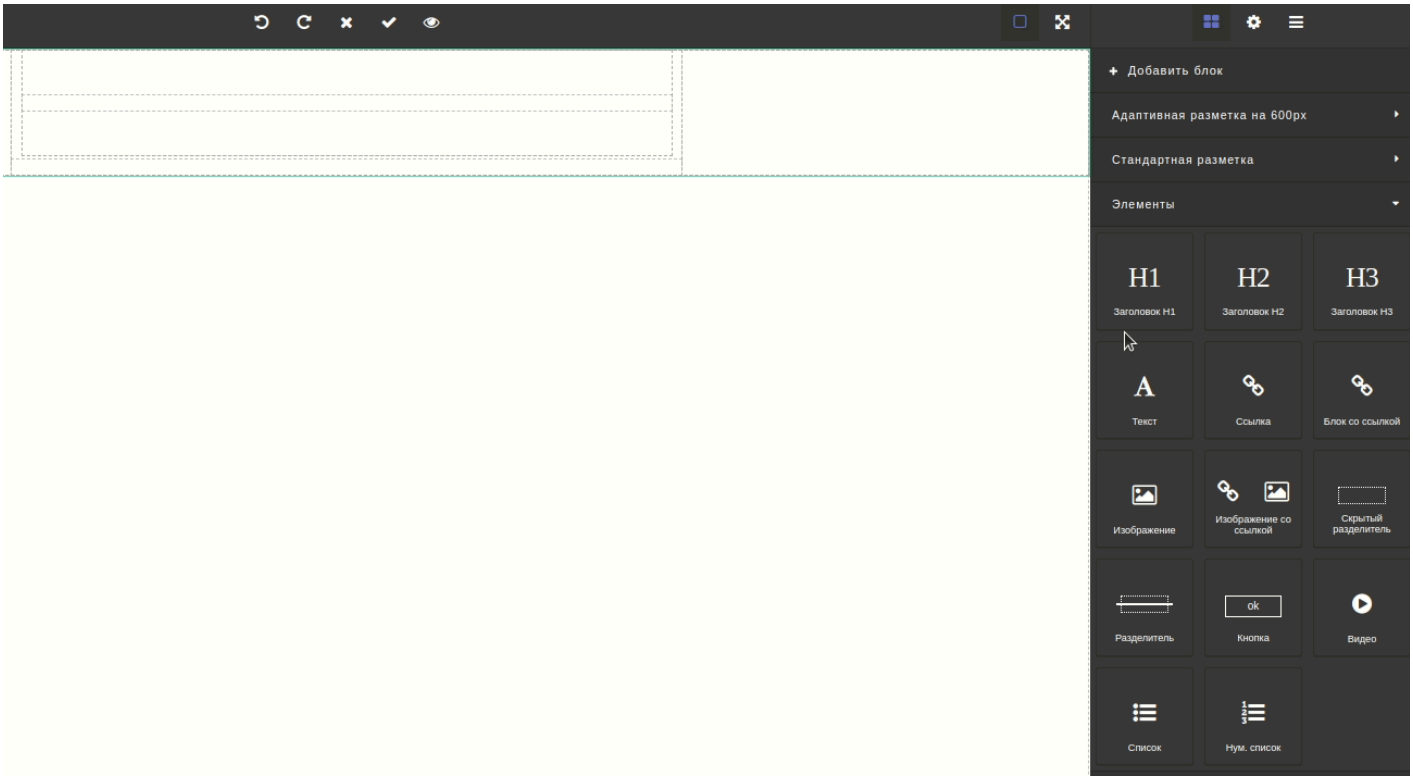


Подробнее о каждом элементе:

Заголовки

При добавлении элемента "**Заголовок"** для него будут доступны следующие **атрибуты и настройки** в правой части редактора:

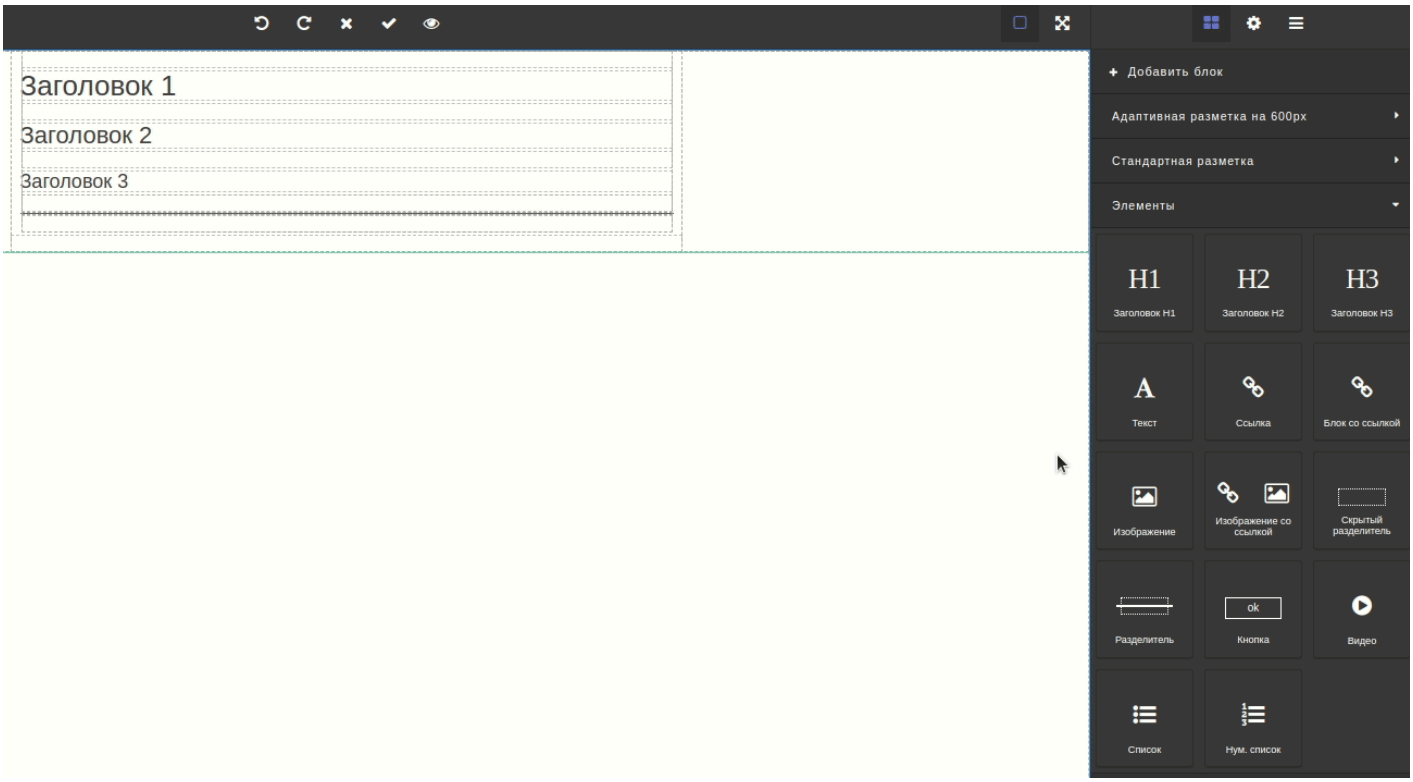
- ID — идентификатор элемента.
- Заголовок для отображения при наведении курсора.



Текст

При добавлении элемента "**Текст"** для него будут доступны следующие **атрибуты и настройки** в правой части редактора:

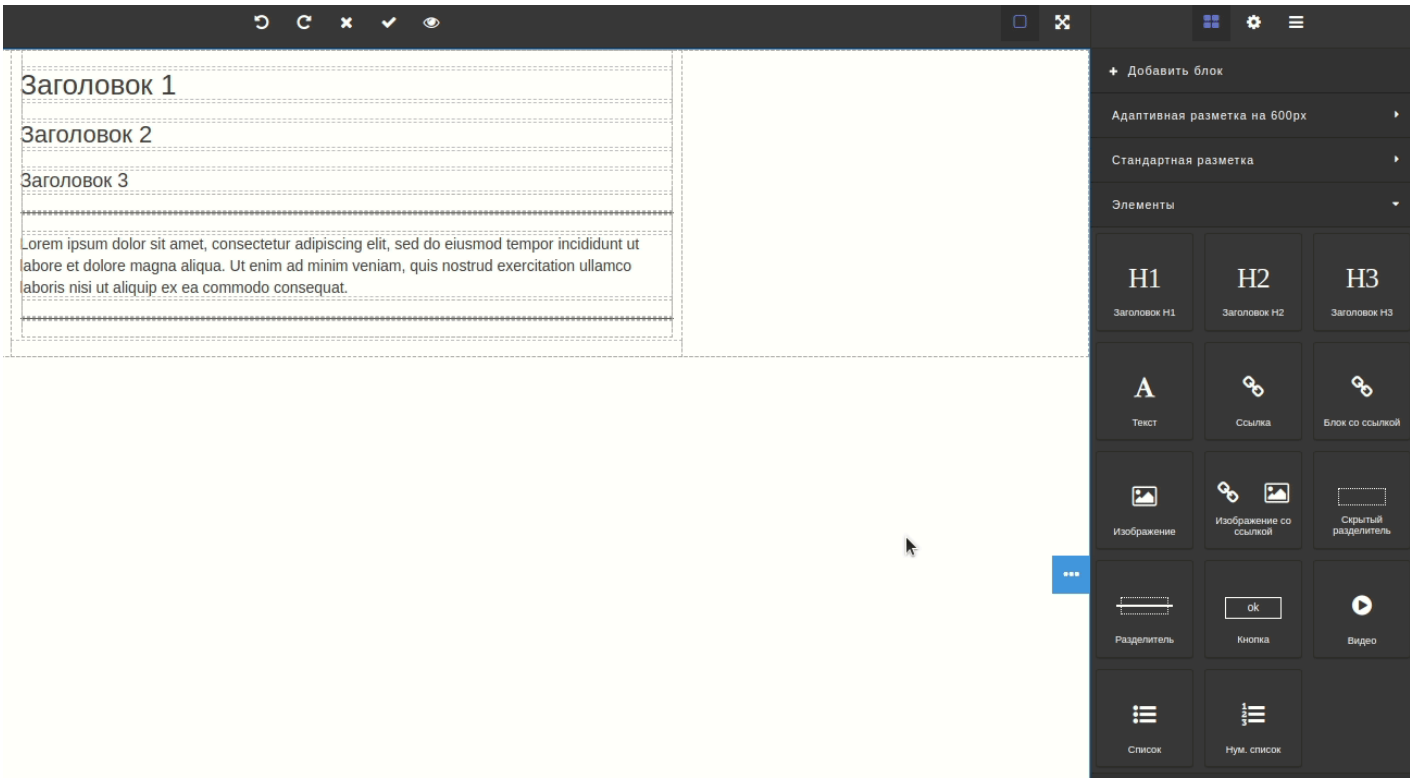
- ID — идентификатор элемента.
- Заголовок для отображения при наведении курсора.



Ссылка и блок со ссылкой

При добавлении элементов **"Ссылка"** для них будут доступны следующие **атрибуты и настройки** в правой части редактора:

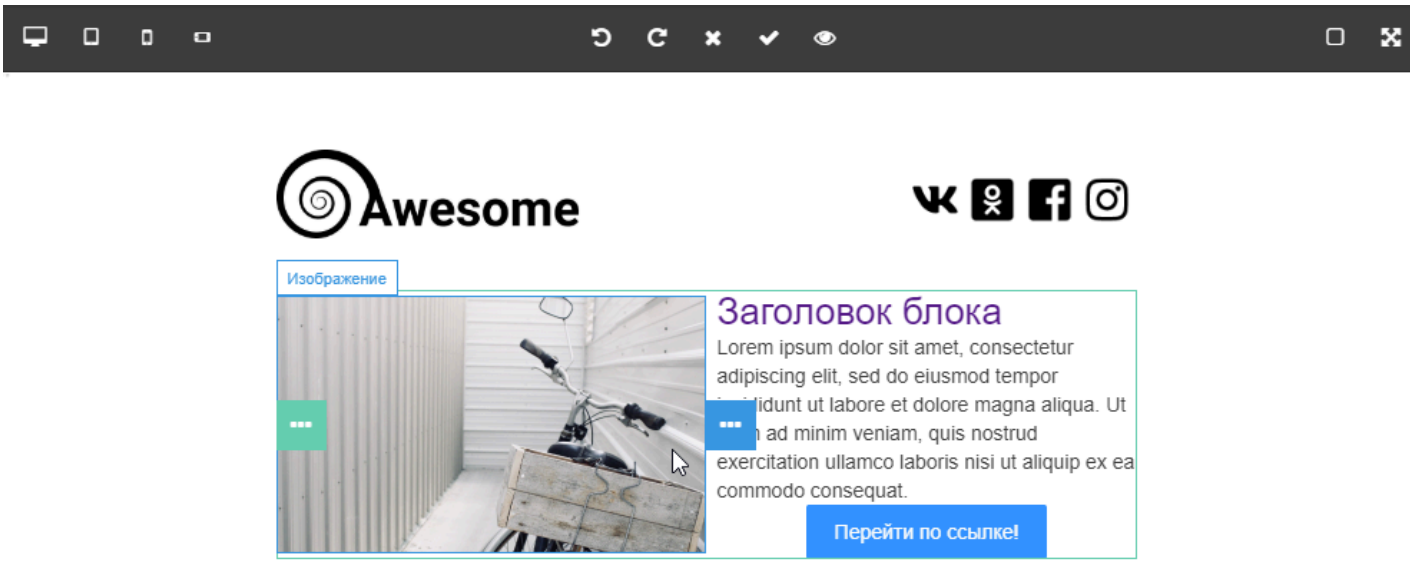
```
    <ul>
      <li>ID – идентификатор элемента.</li>
      <li>Заголовок для отображения при наведении курсора.</li>
      <li>Ссылка для перехода.</li>
      <li>Режим открытия ссылки – в текущем или новом окне.</li>
    </ul>
  </td>
  <td>
  </td>
</tr>
```



Изображение и Изображение со ссылкой

При добавлении элемента **"Изображение"** для него будут доступны следующие **атрибуты и настройки** в правой части редактора:

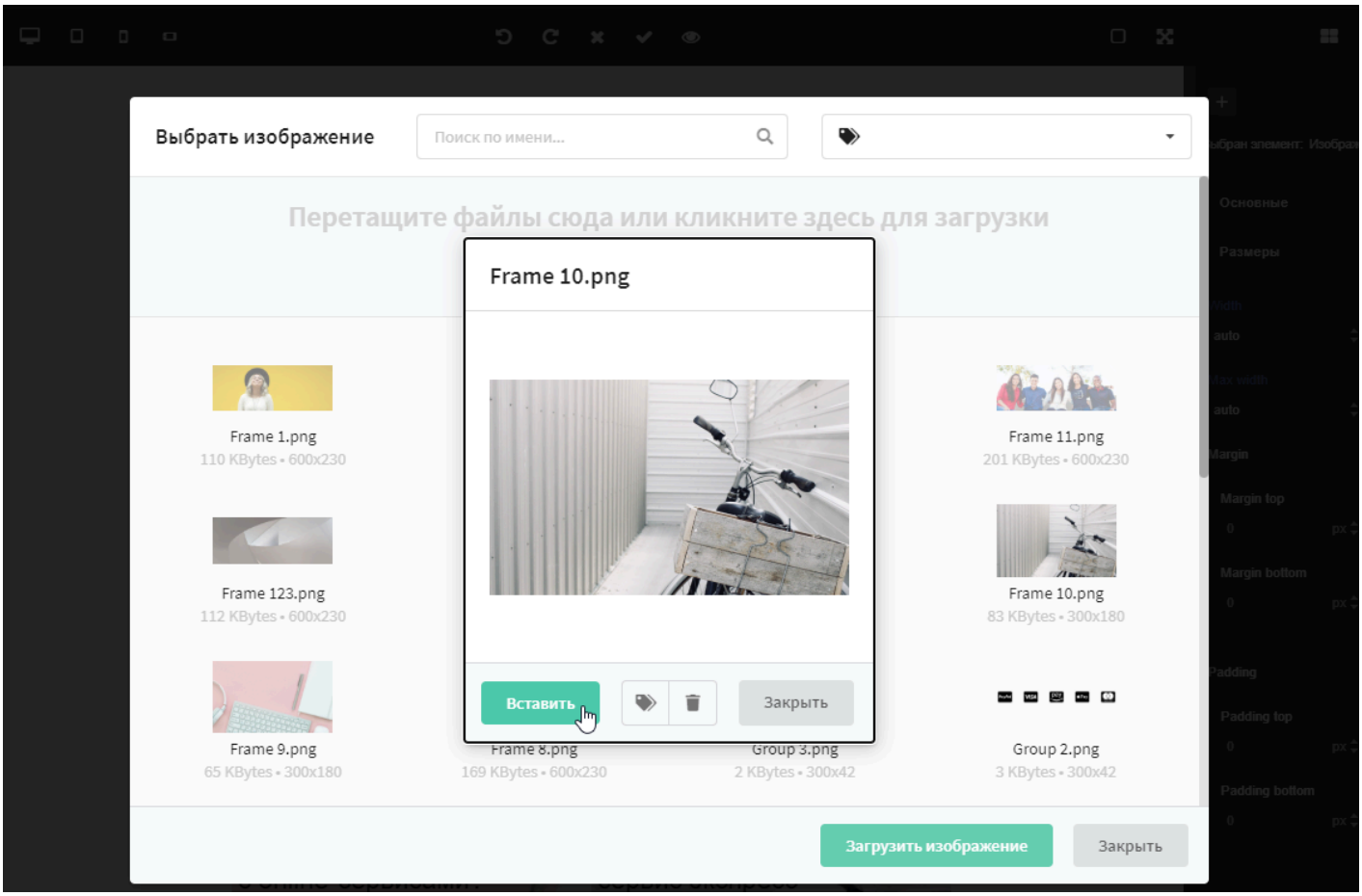
```
    <ul>
      <li>ID – идентификатор элемента.</li>
      <li>Заголовок для отображения при наведении курсора.</li>
      <li>Альтернативный текст (значение атрибута alt).</li>
      <li>Ширина и высота в процентах.</li>
      <li>Ссылка на изображение в платформе или в сети.</li>
    </ul>
  </td>
  <td>
  </td>
</tr>
```



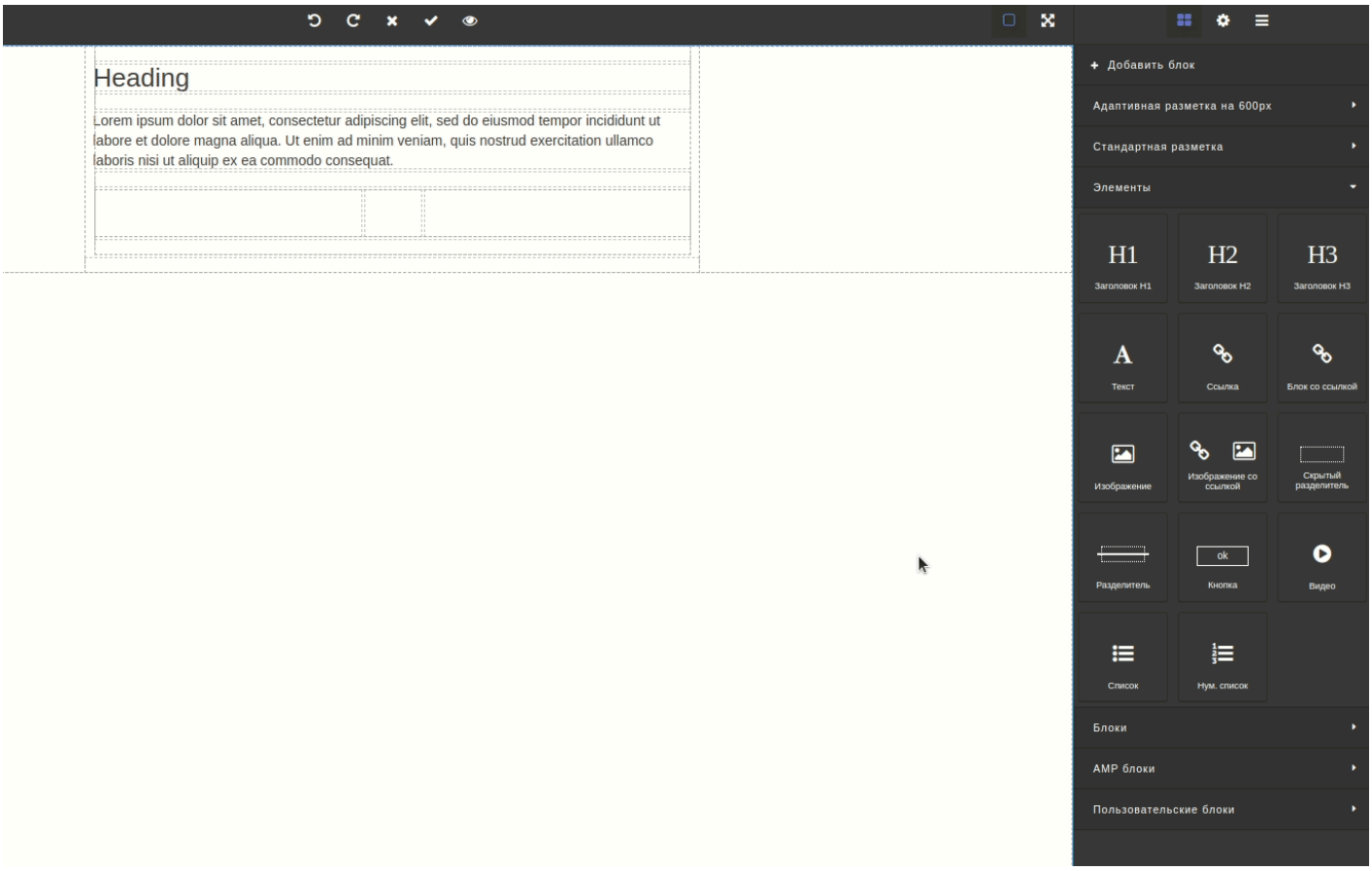
Чтобы открыть галерею изображений платформы, дважды кликните мышью на элемент “Изображение”. В галерее доступен поиск по имени файла и по пользовательским тэгам.

С помощью формы в верхней части вы можете загрузить новое изображение со своего рабочего компьютера: перетащите файл мышью или кликните по форме и укажите нужный файл.

Когда вы нашли нужное изображение, откройте его и кликните **“Вставить”**. Любому изображению вы можете назначить тэги или удалить при необходимости.



После этого вы сможете настроить отображение добавленного изображения в меню **Стилей элемента** в правой части редактора. Работа со стилями описана в следующем разделе.



Разделители

Разделители позволяют визуально отделять элементы друг от друга.

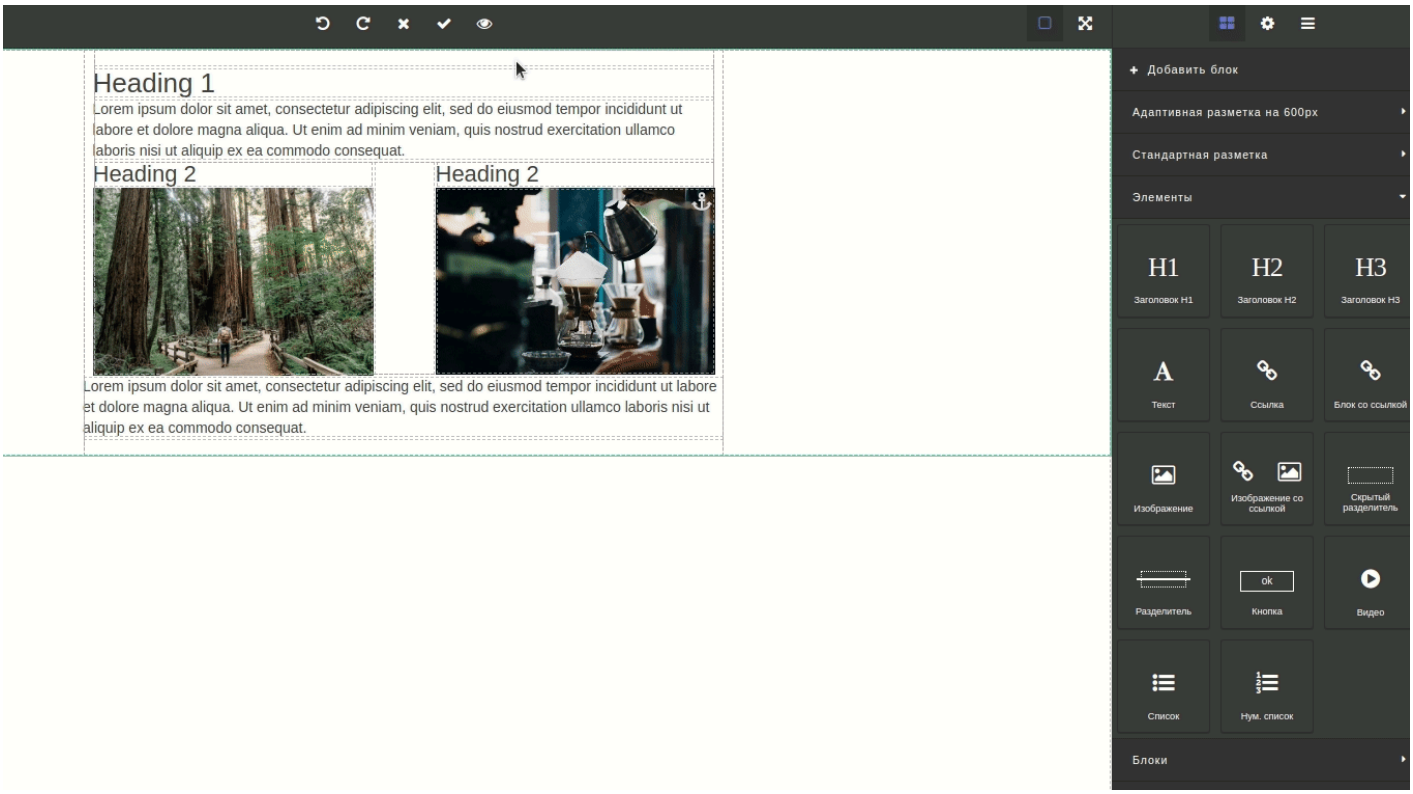
Для **разделителя** вы можете установить **размер** в пикселях и выбрать **цвет:**

Для **скрытого разделителя**настраивается высота в пикселях:

TIP

Элемент “Скрытый разделитель” — это невидимый блок, растягивающийся на всю ширину ячейки.

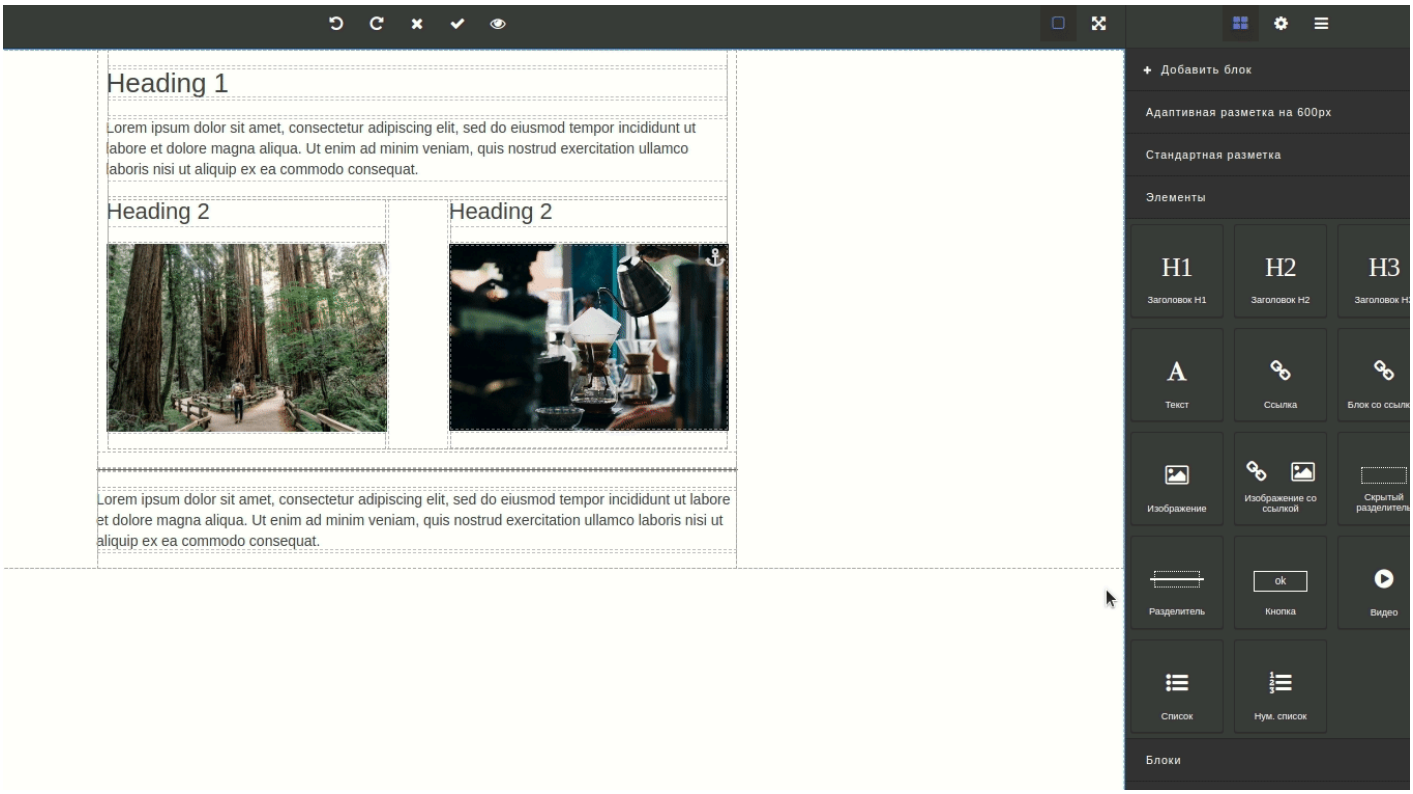
Когда вы используете элемент “Разделитель”, то вы добавляете в свой шаблон два невидимых блока, а также горизонтальную линию, занимающую всю ширину ячейки. Эту линию можно модифицировать: изменять цвет и толщину.



Кнопка

При добавлении элемент а "**Кнопка**" для него будут доступны следующие **атрибуты и настройки** в правой части редактора:

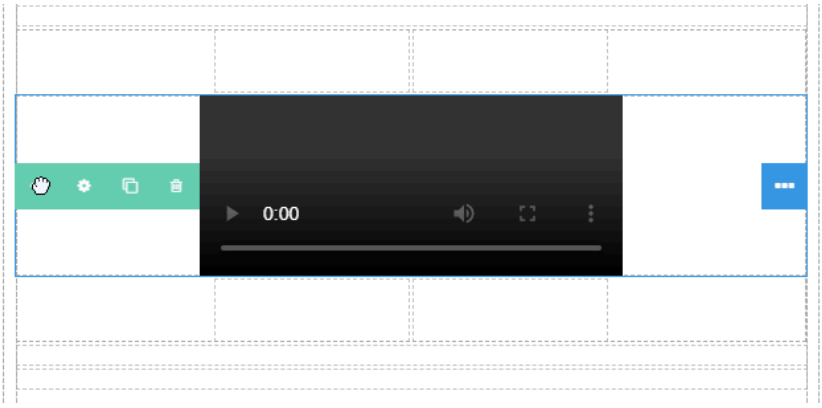
```
    <ul>
      <li>ID — идентификатор элемента.</li>
      <li>Заголовок для отображения при наведении курсора.</li>
      <li>Ссылка для перехода.</li>
      <li>Режим открытия ссылки — в текущем или новом окне.</li>
    </ul>
  </td>
  <td>
  </td>
</tr>
```

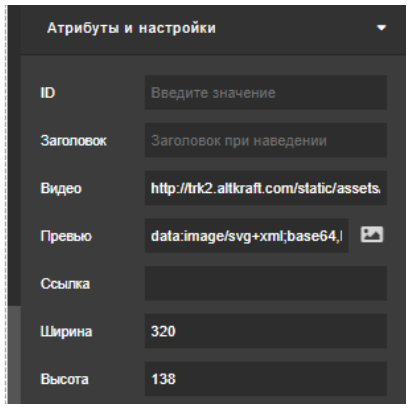


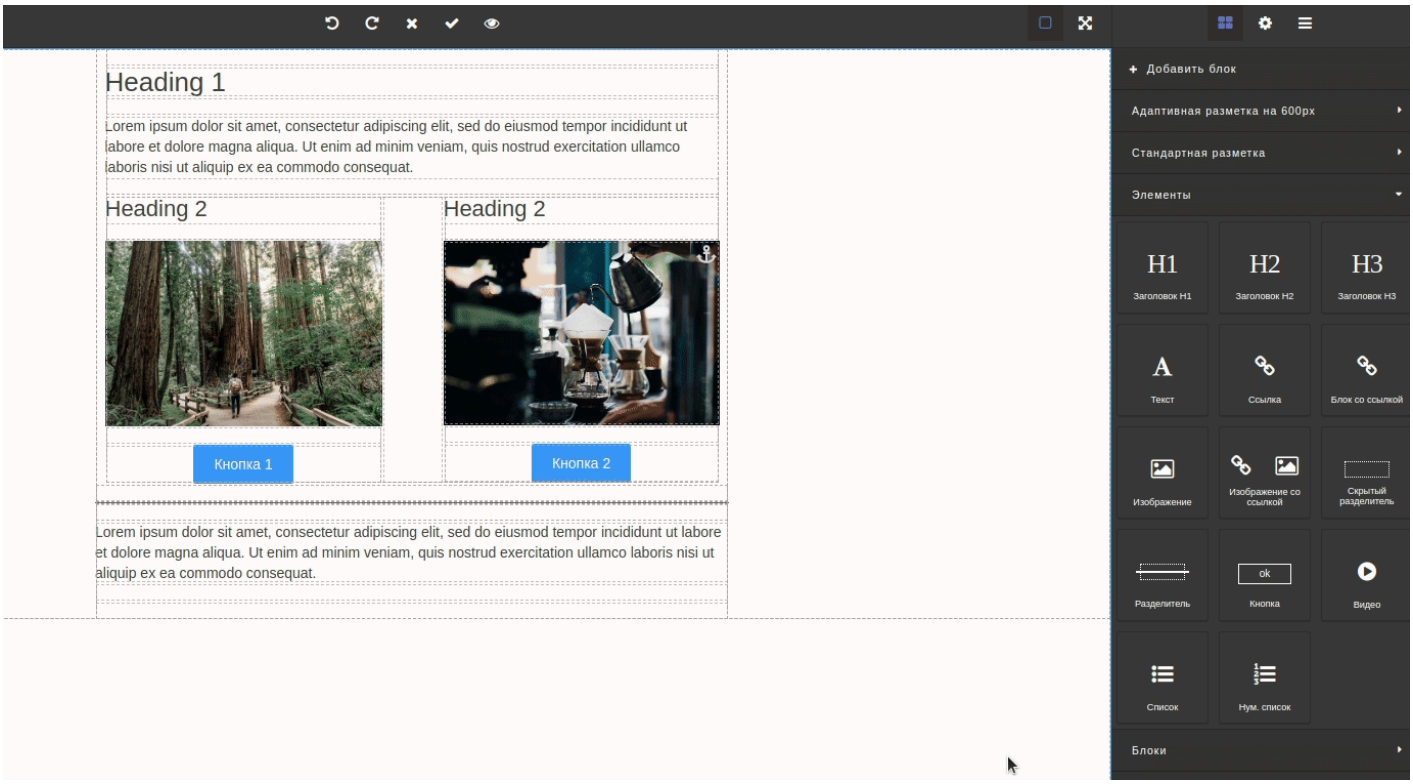
Видео

При добавлении элемента **“Видео”** для него будут доступны следующие **атрибуты и настройки** в правой части редактора:

- ID — идентификатор элемента.
- Заголовок для отображения при наведении курсора.
- Ссылка на видео.
- Изображение для препросмотра видео — из платформы или ссылка.
- Ссылка для перехода по клику.
- Ширина и высота плеера.

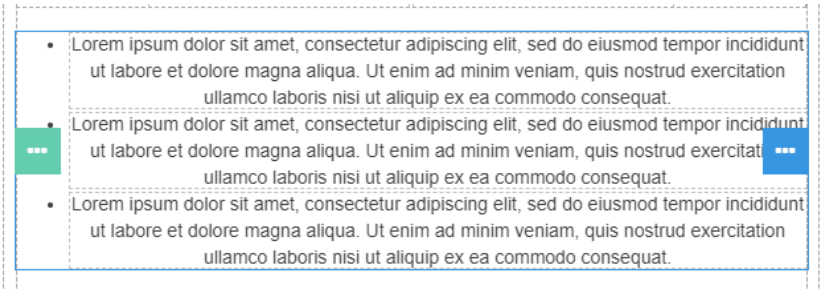


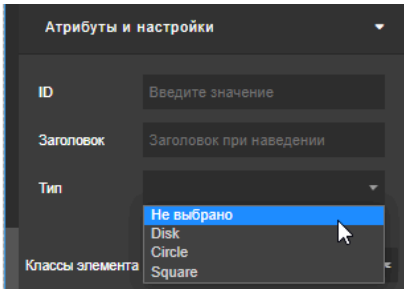




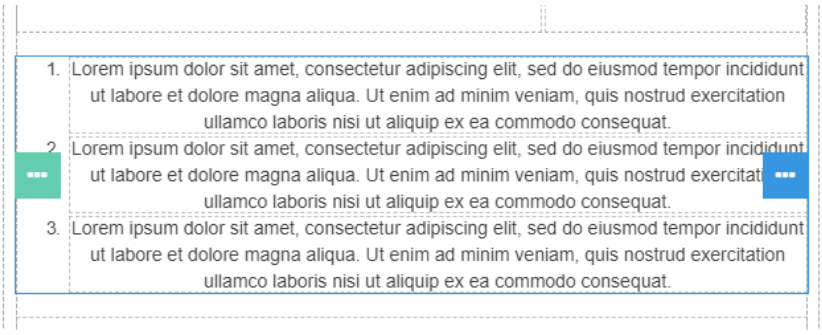
Списки

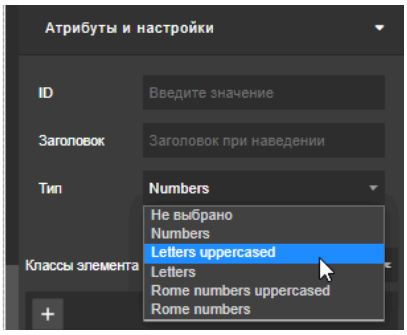
Для **маркированного списка** можно указать ID и заголовок, а также выбрать тип меток: круг, диск или квадрат:



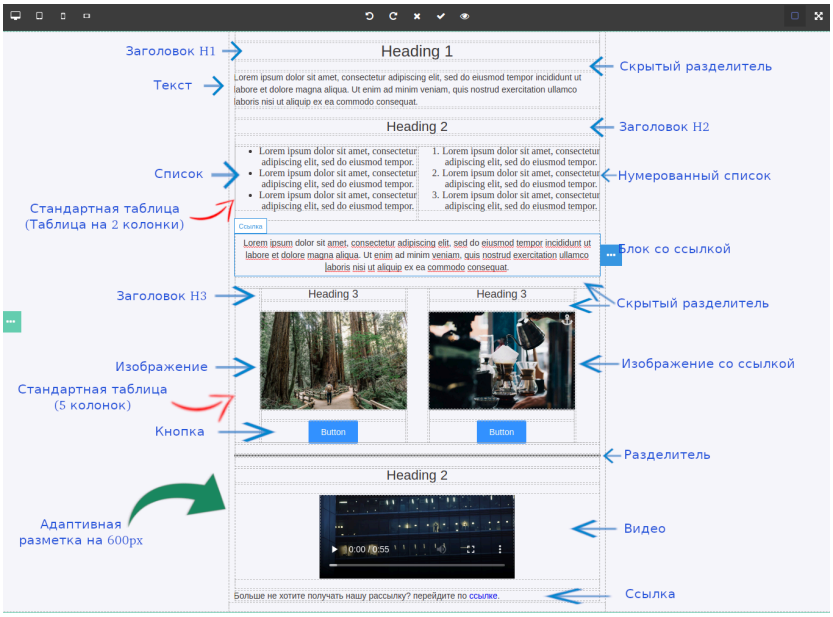
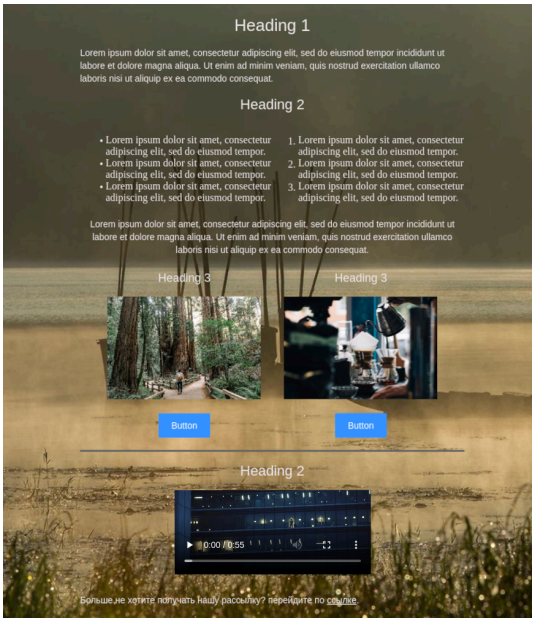


Для **нумерованного списка** можно настроить тип нумерации — римские или арабские цифры — или латинские буквы:





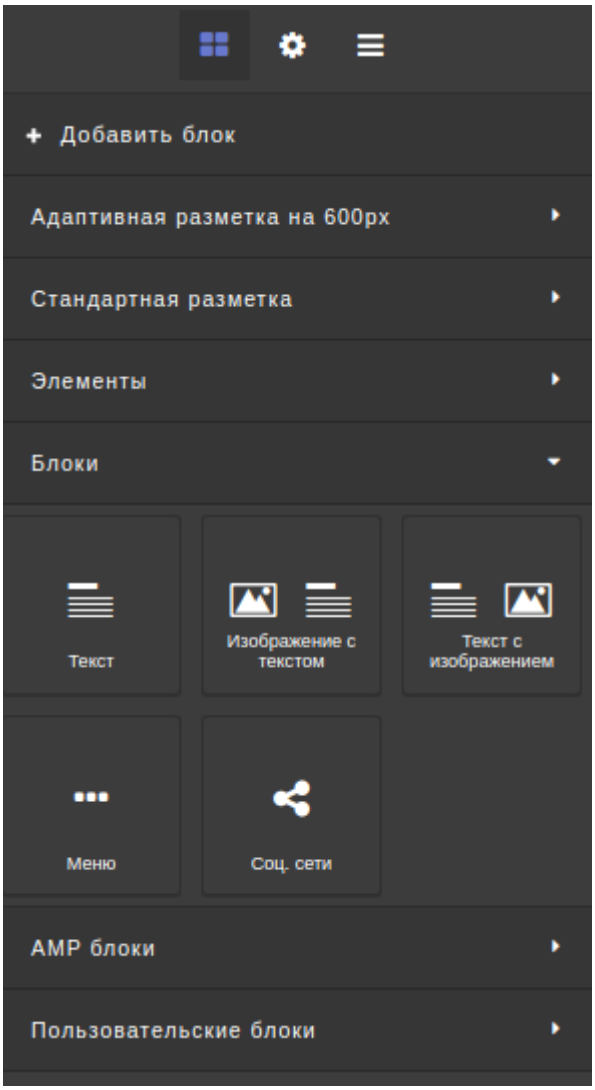
Пример шаблона

Визуальный редактор	Предпросмотр
	

Блоки

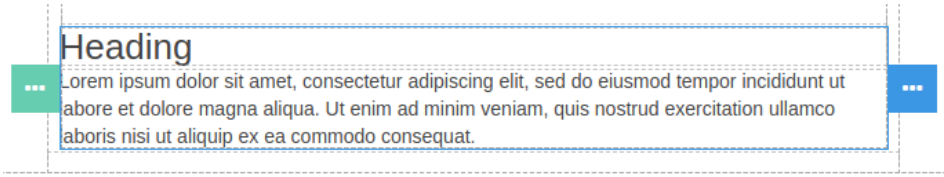
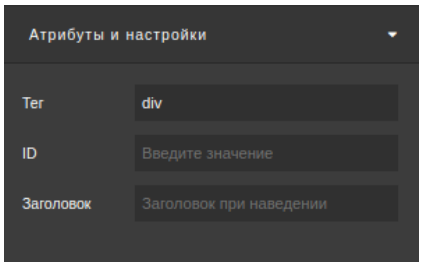
Блоки — это контейнеры, уже содержащие в себе определенные элементы.

Блоки облегчают работу с шаблоном, поскольку представляют собой некие заготовки для пользователя платформы, которые тот, с большой долей вероятности, использует практически в каждом своем Email сообщении. Если вам надоело повторять одно и то же монотонное действие раз за разом, то самое время воспользоваться блоками.



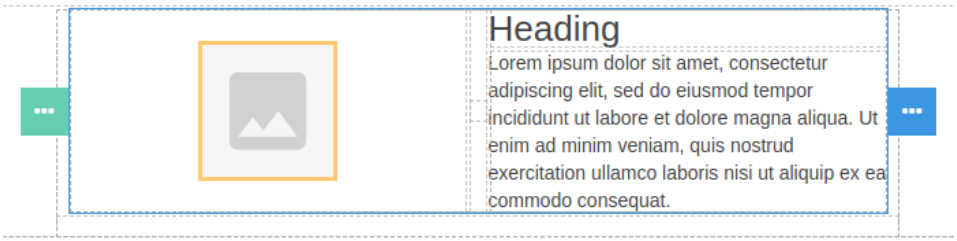
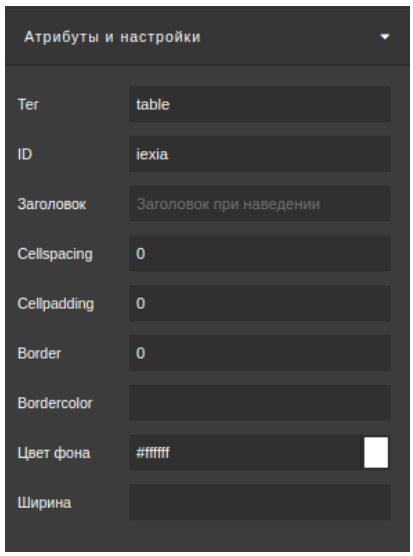
Текст

Блок **“Текст”** — это контейнер, который включает в себя заголовок h1 и параграф.

Внешний вид	Атрибуты и настройки
	

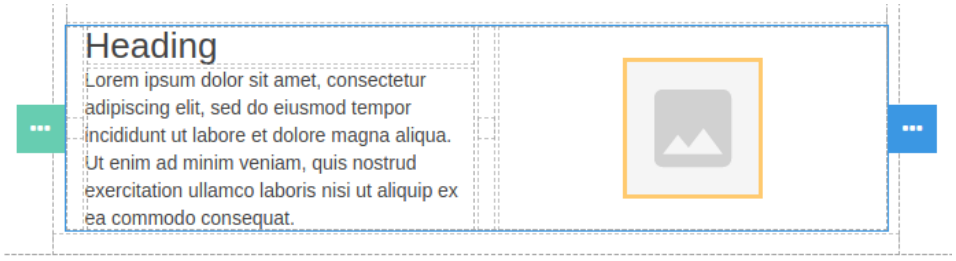
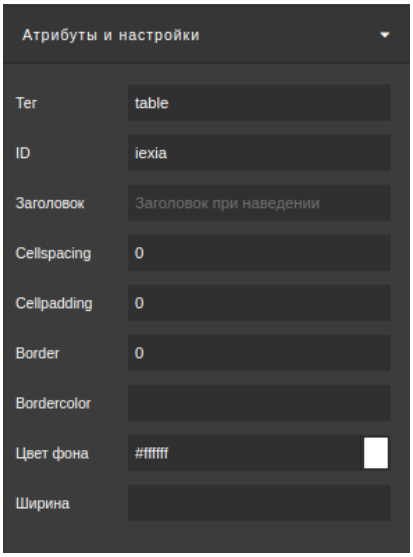
Изображение с текстом

Блок **“Изображение с текстом”** — это таблица, состоящая из двух ячеек. Левая ячейка включает в себя изображение, а правая — заголовок h1 и параграф.

Внешний вид	Атрибуты и настройки
	

Текст с изображением

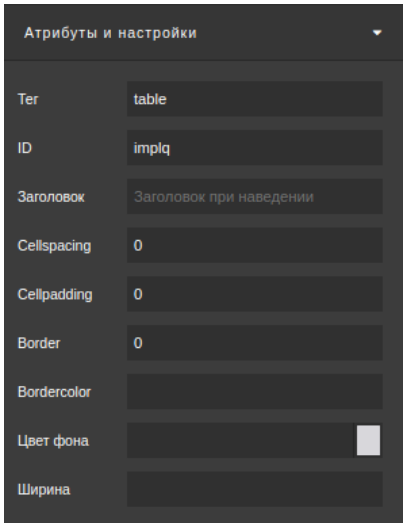
Блок **“Текст с изображением”** — это таблица, состоящая из двух ячеек. Левая ячейка включает в себя заголовок h1 и параграф, а правая — изображение.

Внешний вид	Атрибуты и настройки
	

Меню

Блок **“Меню”** — это таблица, состоящая из 4-х ячеек одинаковой ширины.

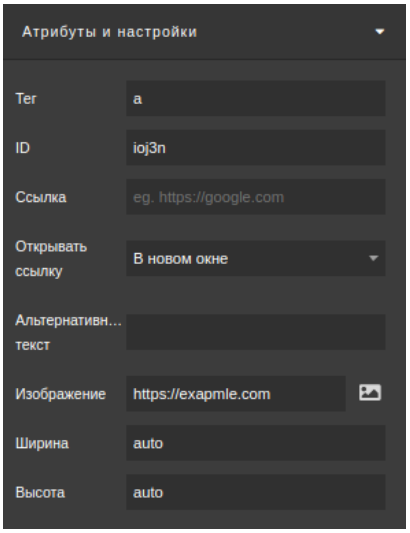
Каждая ячейка включает в себя ссылку, которую вы можете настроить в панели **Атрибуты и настройки**. Там же у вас есть возможность изменить внешние и внутренние отступы между ячейками, настроить ширину границы и общую ширину элемента, а также задать цвет фона.

Внешний вид	Атрибуты и настройки
	

Социальные сети

Блок **“Социальные сети”** — это контейнер, включающий в себя 8 изображений со ссылками на социальные сети.

Для каждого элемента блока “Социальные сети” вы можете задать свою ссылку для перехода, добавить альтернативный текст, настроить изображение, а также изменить ширину и высоту элемента.

Внешний вид	Атрибуты и настройки
	

AMP блоки

AMP-письма — это технология, которая позволяет маркетологам встраивать интерактивные элементы, такие как карусель и аккардеон в свои письма. Пользователь, в свою очередь, может посмотреть данные, даже не открывая новую вкладку для посещения веб-сайта. В платформе Altcraft доступны следующие APM блоки:

Карусель

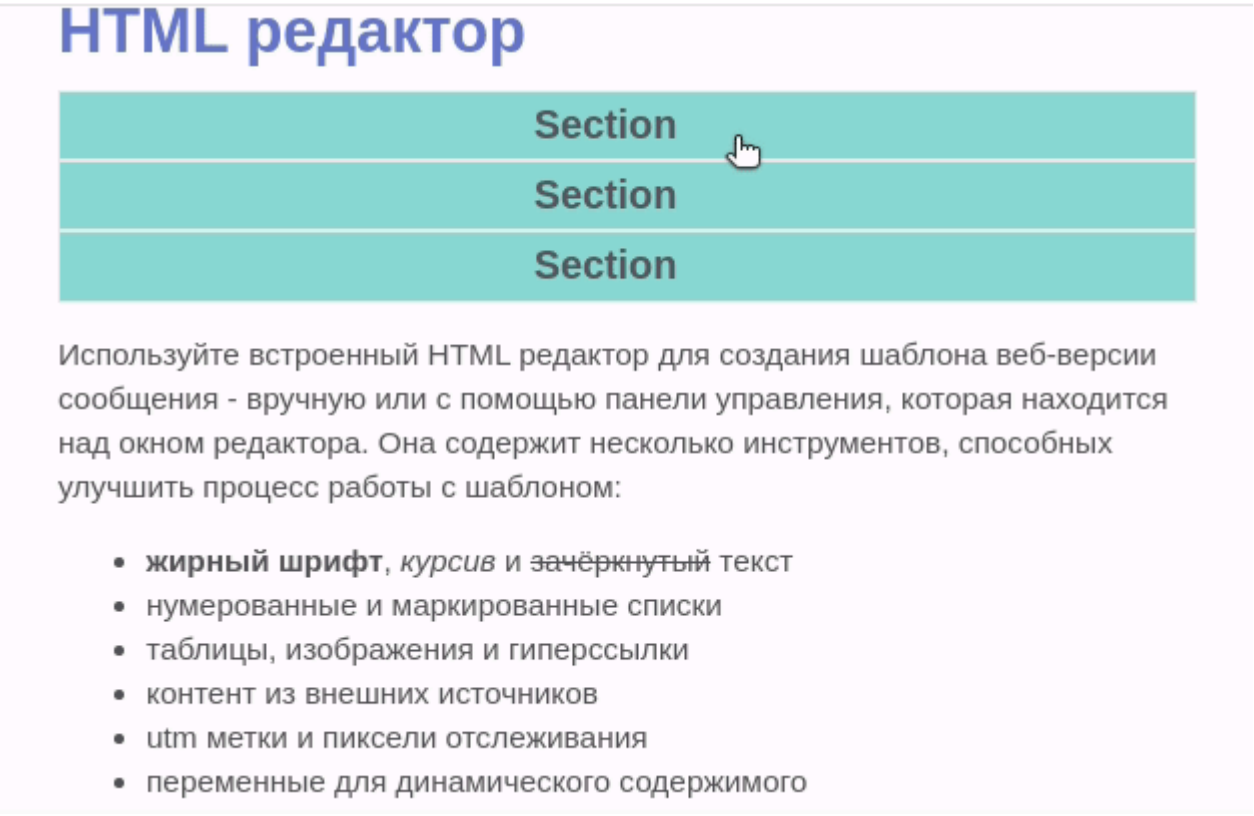
Карусель — это блок, включающий в себя несколько слайдов. Карусель позволяет нескольким фрагментам контента занимать одну и ту же область пространства на странице или экране.

Настройки **AMP карусели** задаются в **меню стилей** элемента. Вы можете задать ID элемента, установить его высоту и добавить слайды, для которых установить изображения и ссылку для перехода. Изображение можно выбрать из галереи или загрузить с компьютера. Также можно указать ссылку на изображение в сети:

Аккордеон

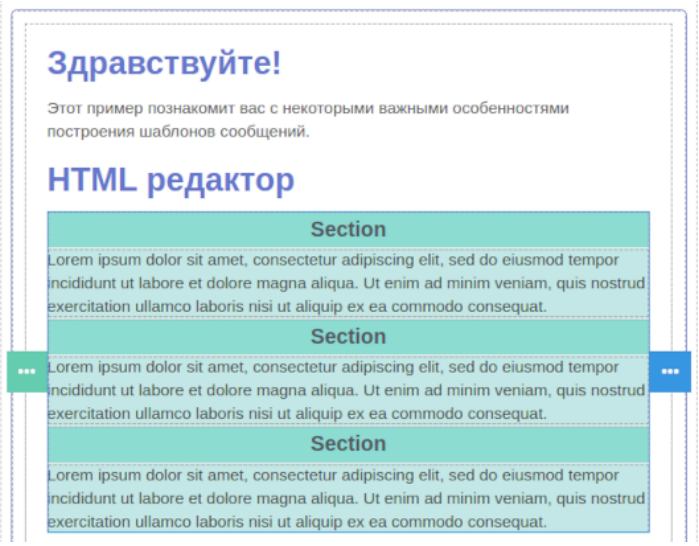
Аккордеон — это блок, состоящий из нескольких вкладок с элементами. Каждый элемент может быть “развёрнут” или “раскрыт” при определенном действии со стороны пользователя (клик, наведение курсора). Вы можете добавлять в секции любые элементы.

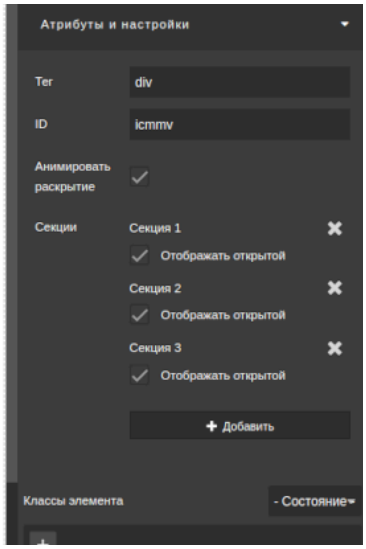
Такой элемент интерфейса как аккардеон не только помогает пользователям быстро перемещаться по материалам, но и позволяет дизайнеру пользовательского интерфейса включать большие объемы информации в ограниченном пространстве.



Настройки **AMP аккордеона** задаются в **меню стилей**.

Выделите блок (зелёное меню) и добавьте необходимое количество секций. Секции можно отображать открытыми или закрытыми по умолчанию. Затем добавьте в секции нужные элементы:





💡

Если вы разобрались, как добавлять и настраивать различные элементы в шаблоне, то осталось последнее — стилизовать элементы.

💡

Пользовательские блоки

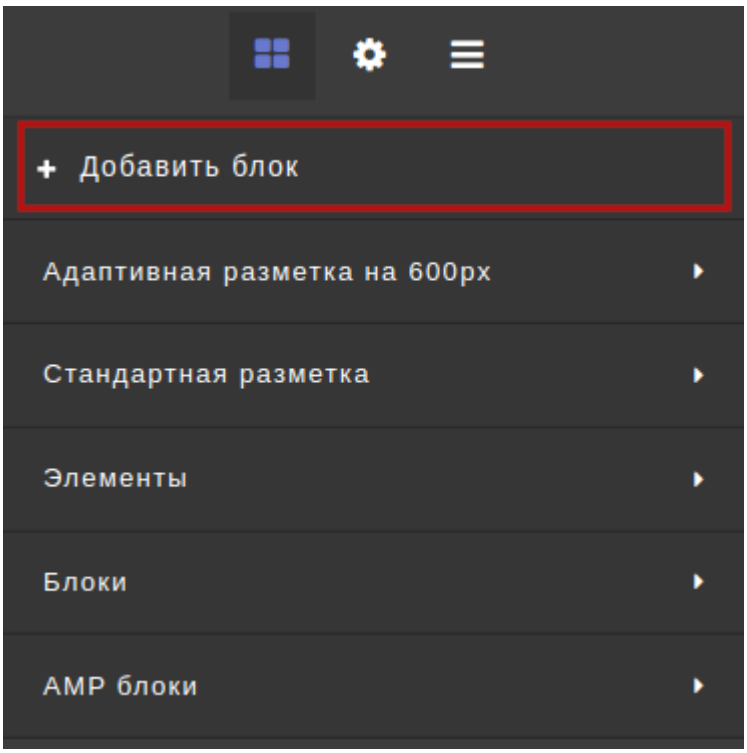
Сохраните целую секцию шаблона как пользовательский блок и потом используйте этот блок снова и снова. Причем использовать вы его можете не только в том шаблоне, где он был создан, но и в любом другом.

Что стоит сохранять как пользовательские блоки? Любой контент, который повторяется из шаблона в шаблон. Например, хедер или футер сообщения.

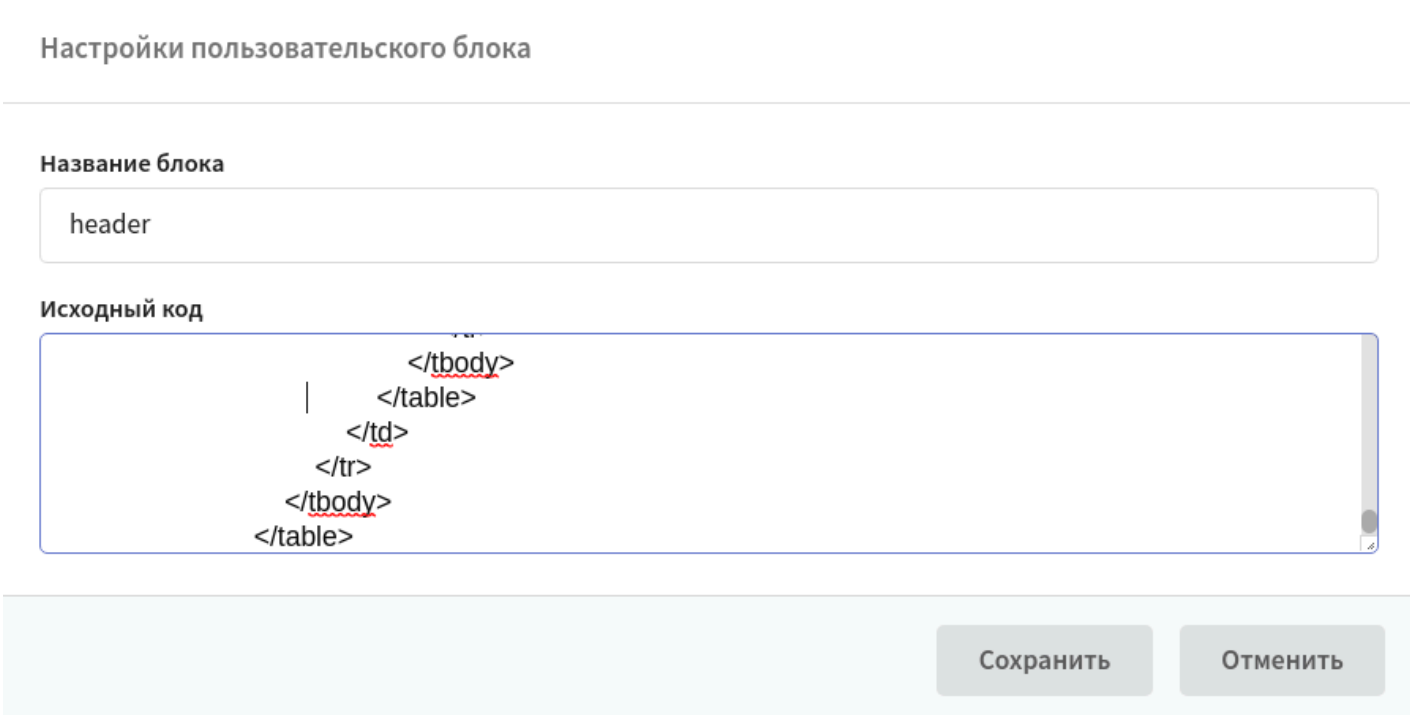
Как создать пользовательский блок?

Вариант 1: Добавить исходный код

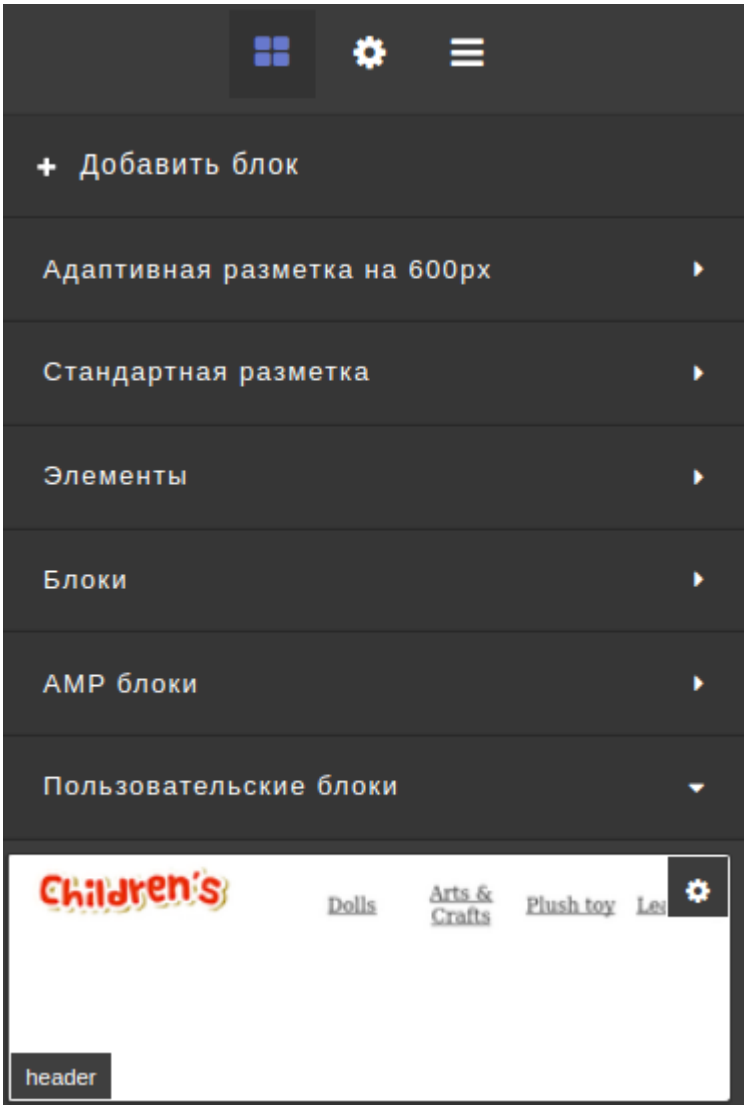
В правой панели визуального редактора перейдите в раздел “Добавить элемент”. Кликните “**+ Добавить блок**”.



Скопируйте в открывшуюся форму исходный код фрагмента и сохраните блок.



Новый пользовательский блок будет отображаться после всех доступных элементов визуального редактора.



Вариант 2: Сохранить контейнер шаблона

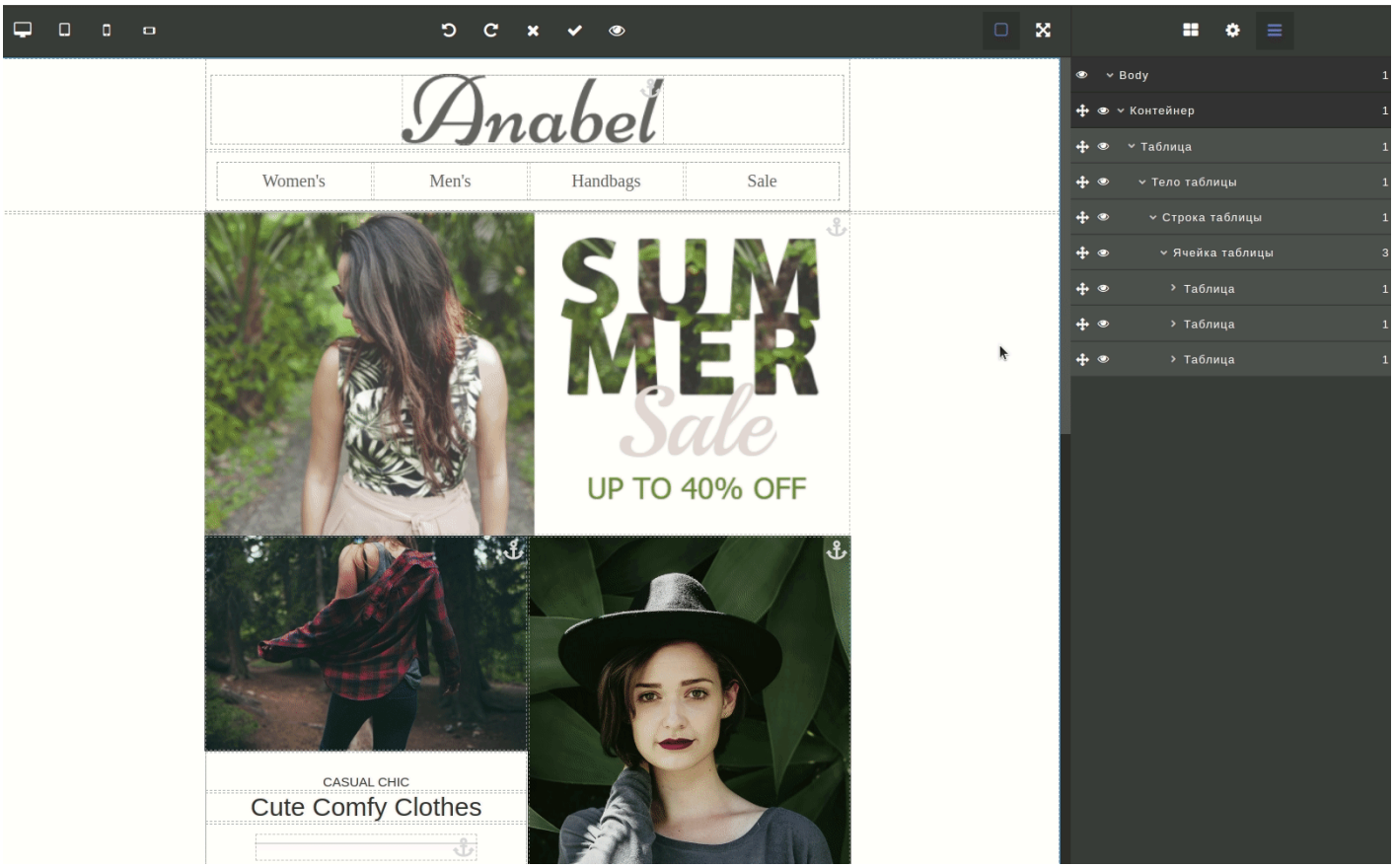
Выберите ту группу элементов, которую хотите сохранить. Вы можете просто навести курсор на нужный контейнер в шаблоне, а можете воспользоваться разделом **“Структура элементов”.**

После того как вы выделите нужную секцию, справа появится кнопка вызова меню операций с контейнером

. Клкините иконку с дискетой

, чтобы сохранить блок.

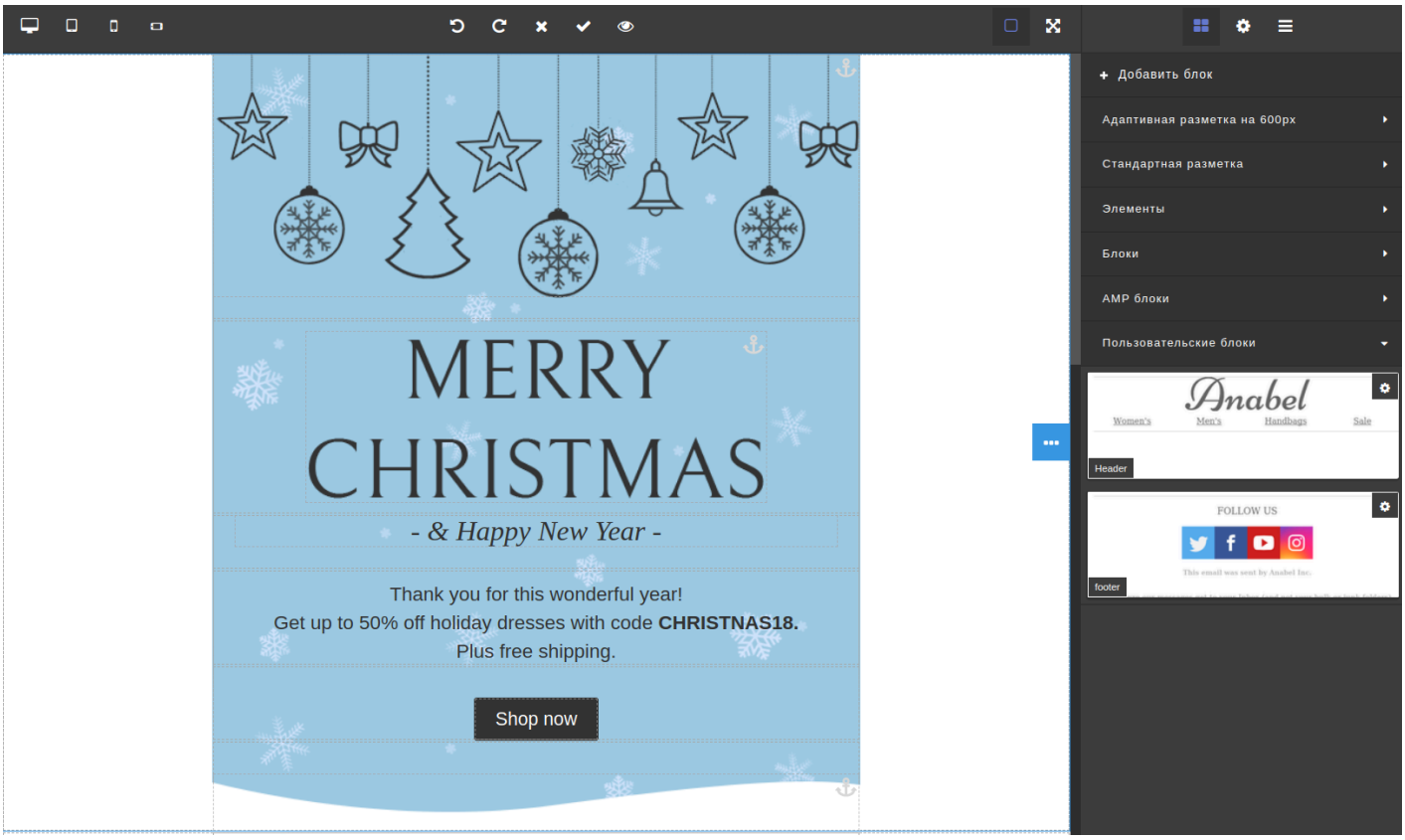
Новый пользовательский блок будет отображаться после всех доступных элементов визуального редактора.



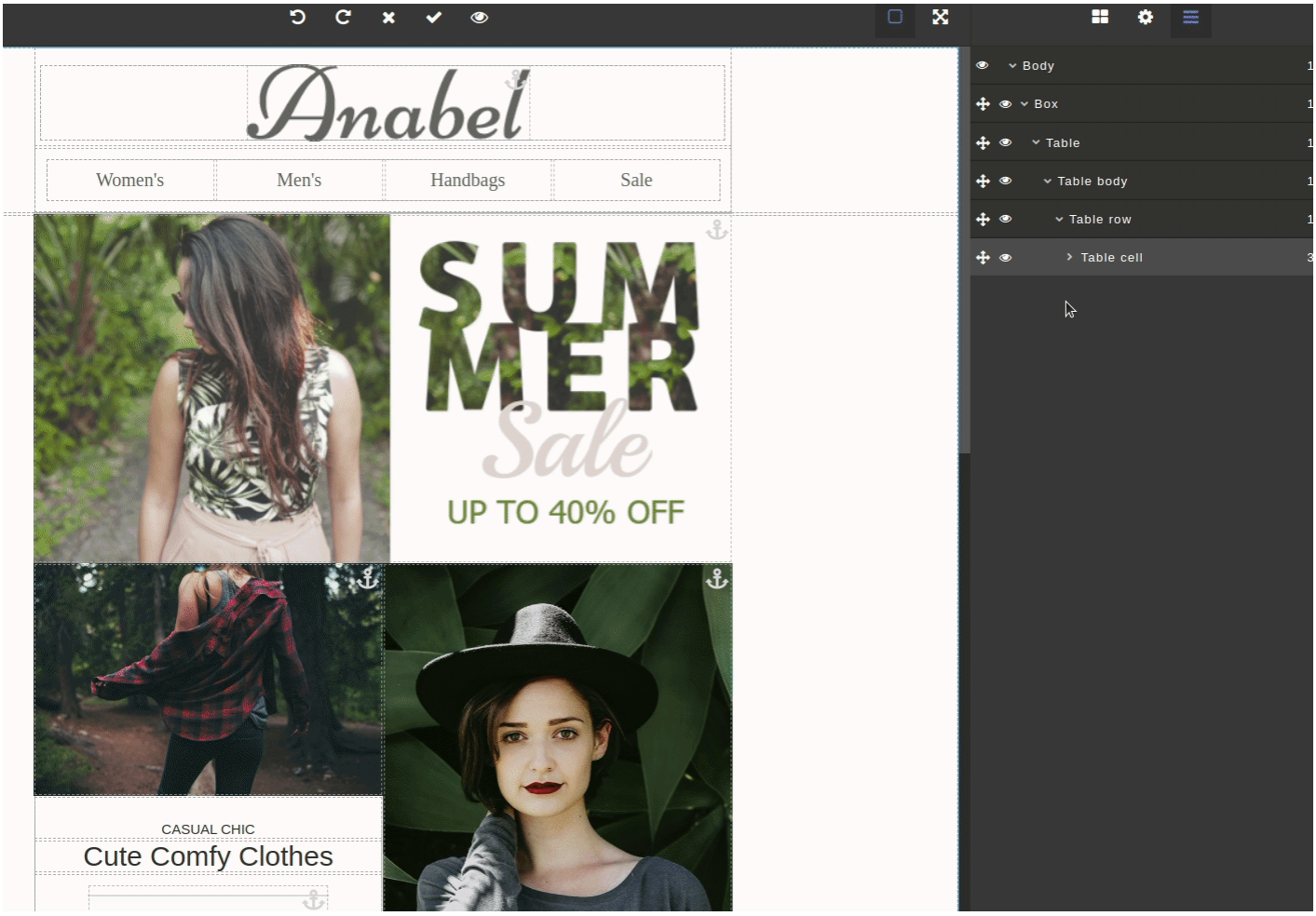
Как добавить пользовательский блок?

Выберите пользовательский блок, кликнув по нему мышкой, а затем перетащите его в нужную вам часть шаблона.

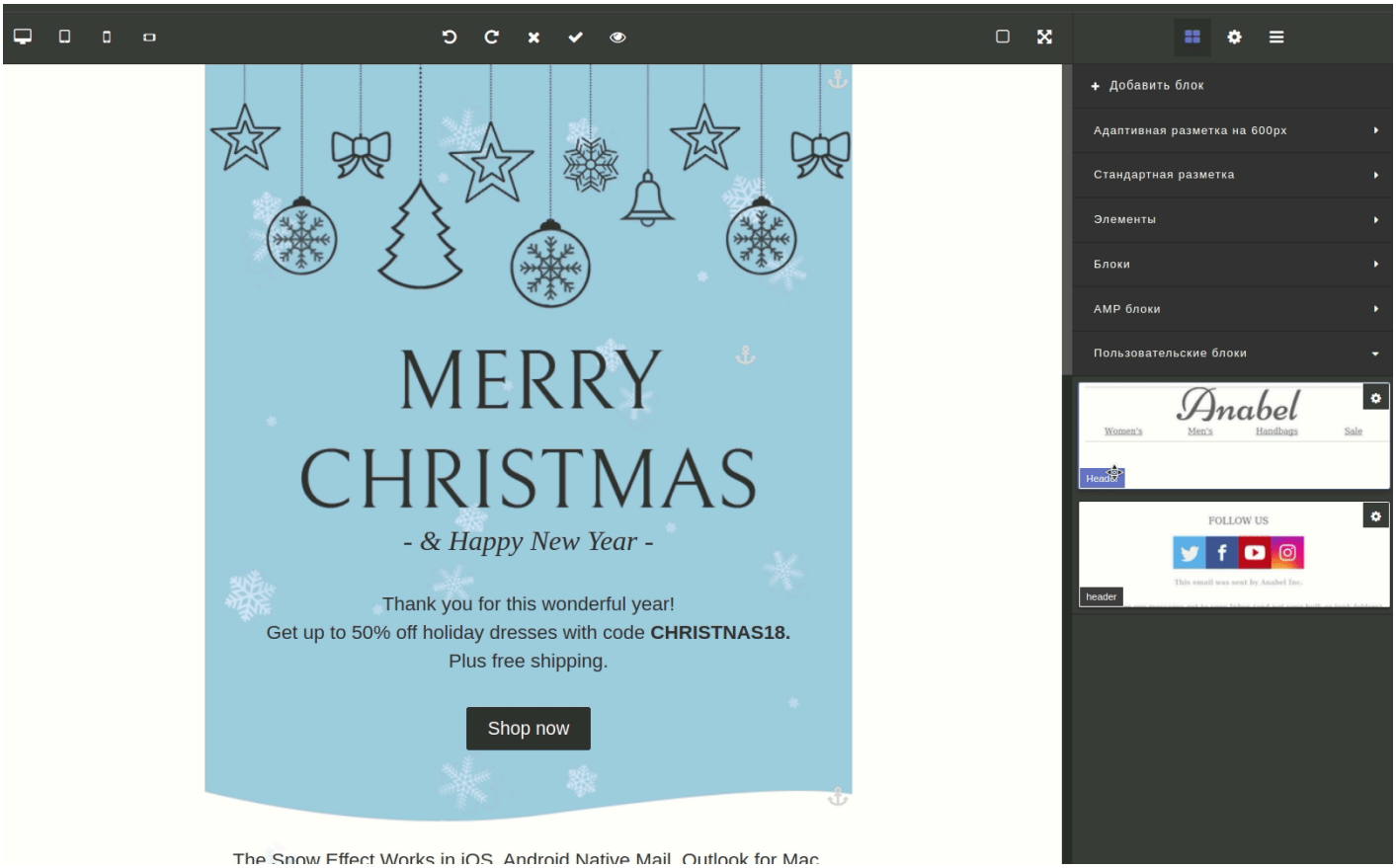
Допустим, мы осуществляем email-рассылки клиентам магазина одежды Anabel. Впереди Рождество, и мы решили подарить нашим подписчиками промокод на скидку. Основная часть шаблона готова:



По сравнению с ежедневными рассылками, основная часть сообщения существенно изменилась, но хедер с логотипом магазина и футер с контактами останутся теми же, что и в ежедневных рассылках. Перейдем в один из старых шаблонов и сохраним хедер и футер как пользовательские блоки.



Осталось добавить пользовательские блоки в рождественский шаблон:

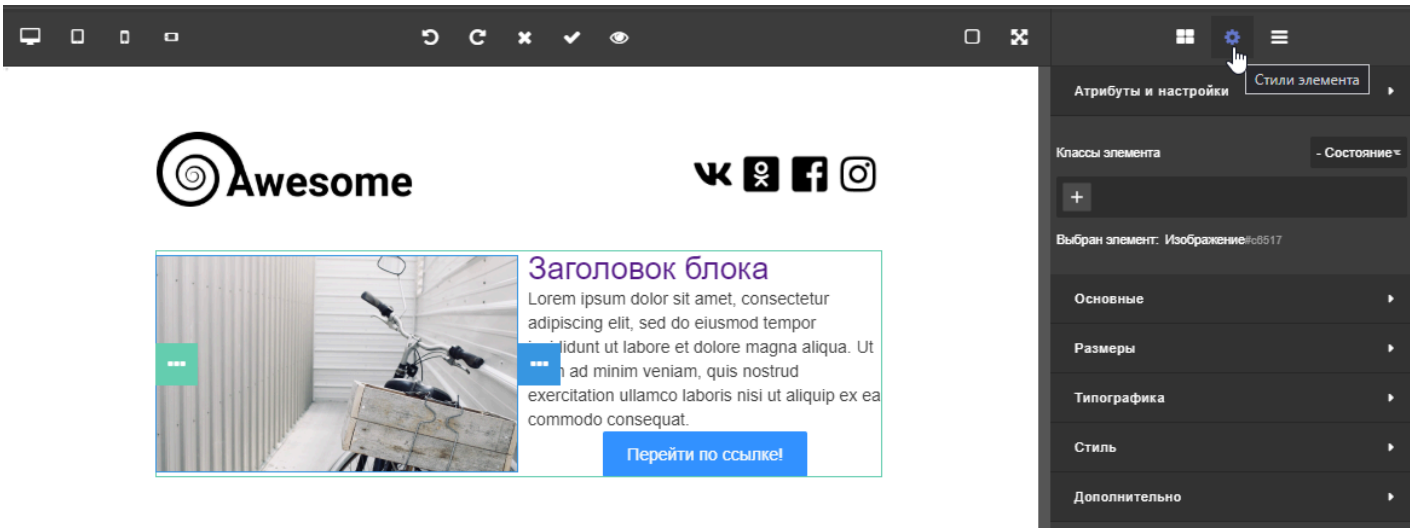


Отлично, шаблон готов, а мы сэкономили немало времени!

👍

Стили элемента

Центральная кнопка в правой части редактора открывает меню стилей выделенного элемента. Набор настроек и атрибутов может меняться в зависимости от элемента. Основные стили остаются доступными для всех элементов и записываются в атрибут `style` HTML разметки, так что вы сможете легко настроить шаблон, который будет одинаково хорошо отображаться во всех почтовых клиентах.



Классы и состояние

В верхней части меню вы можете назначить элементу определённый класс стилей, а также переключить состояние — для добавления эффектов при наведении или нажатии.	

Основные

В основных стилях задаётся расположение элемента в письме:

```
    <ul>
      <li>
        Float — выравнивание блочного элемента по правой или левой стороне
        контейнера.
      </li>
      <li>
        Display — inline, block, inline-block или flex — режим отображения
        элемента.
      </li>
      <li>
        Position, Top, Left, Bottom — режим расположения и смещение элемента
        сверху, справа, слева и снизу.
      </li>
    </ul>
  </td>
  <td>
  </td>
</tr>
```

Размеры

В секции **Размеры** настраиваются следующие параметры отображения:

```
    <ul>
      <li>Width, Height — ширина и высота элемента.</li>
      <li>
        Max width, Max height — максимальные ширина и высота элемента.
      </li>
      <li>Margin — внешняя граница элемента.</li>
      <li>Padding — внутренний отступ от границы элемента.</li>
    </ul>
  </td>
  <td>
  </td>
</tr>
```

Типографика

В секции **Типографика** расположены параметры отображения текстовых элементов:

```
    <ul>
      <li>
        Text align — выравнивание текста по левой, правой стороне
        контейнера, по центру или по ширине.
      </li>
      <li>Font family — семейство шрифтов текста.</li>
      <li>Font size — размер шрифта.</li>
      <li>Font weight — толщина шрифта.</li>
      <li>Letter spacing — расстояние между буквами.</li>
      <li>Color — цвет в rgb или hex.</li>
      <li>Line height — высота строки</li>
      <li>
        Text shadow — добавление тени для текста с настройкой цвета,
        горизонтального и вертикального сдвига, размытия:
      </li>
    </ul>
  </td>
  <td>
  </td>
</tr>
```

Стиль

В секции **Стиль** настраиваются параметры границы элемента, фона и тени элемента:

```
    <ul>
      <li>Border radius – скругление всех углов границы элемента.</li>
      <li>Background color – цвет фона элемента.</li>
      <li>
        Блок Border radius – настройки скругления для каждого угла границы
        элемента.
      </li>
      <li>Блок Border – толщина, стиль и цвет границы элемента.</li>
    </ul>
    <strong>Box shadow</strong> – добавляет внутреннюю или внешнюю тень
    элемента. Вы можете настроить смещение, толщину и размытие тени, а также
    выбрать цвет:

    <strong>Background image</strong> – добавляет фоновое изображение для
    элемента. Вы сможете выбрать изображение из галереи или загрузить с
    компьютера, настроить размер, повторение, и положение фона элемента:

  </td>
  <td>
  </td>
</tr>
```

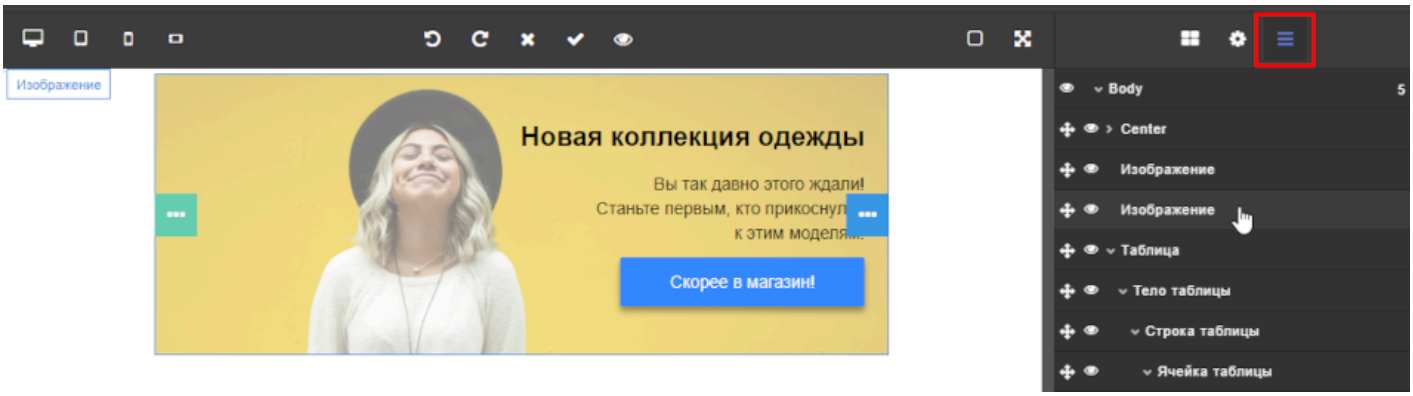
Дополнительно

В дополнительных параметрах вы можете настроить анимацию элемента при переходе в разные состояния. Здесь вы можете настроить AMP элементы, например, карусель изображений.

```
    <ul>
      <li>
        Transition – добавьте анимированный переход, настроив режим и
        длительность.
      </li>
      <li>Perspective – перспектива отображения.</li>
      <li>
        Transform rotate, Transform scale – настройки вращения и увеличения
        элемента по трём осям координат.
      </li>
    </ul>
  </td>
  <td>
  </td>
</tr>
```

Структура элементов

Последний раздел в правой панели редактора — это **“Структура элементов”.**

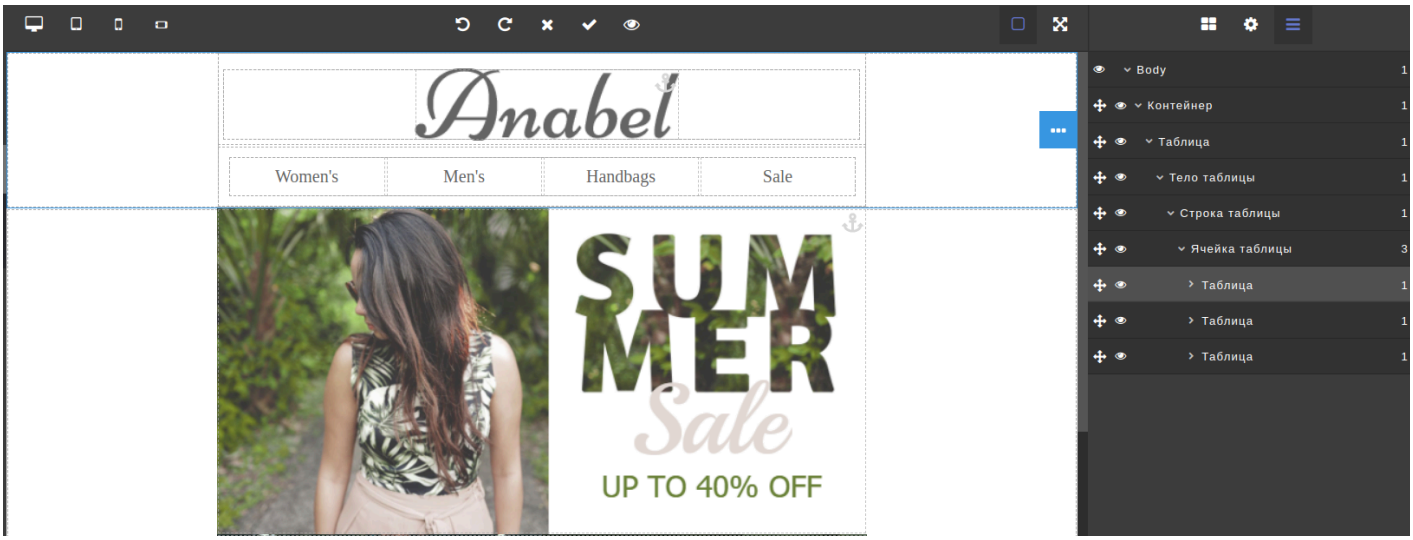


Здесь вы найдете дерево структуры элементов, с помощью которого можно в пару кликов добраться до любого элемента в письме. Такой инструмент может быть очень полезен, особенно если вам приходится работать с шаблоном, который создавали не вы. Один взгляд на структуру элементов позволит вам понять общую схему шаблона и скорректировать его, не углубляясь в детали.

Когда вы открываете структуру элементов, в правой части появляется список всех элементов шаблона. Здесь вы четко можете увидеть, как именно элементы вложены друг в друга.

Цифра напротив каждого элемента показывает, сколько детей есть у родителя.

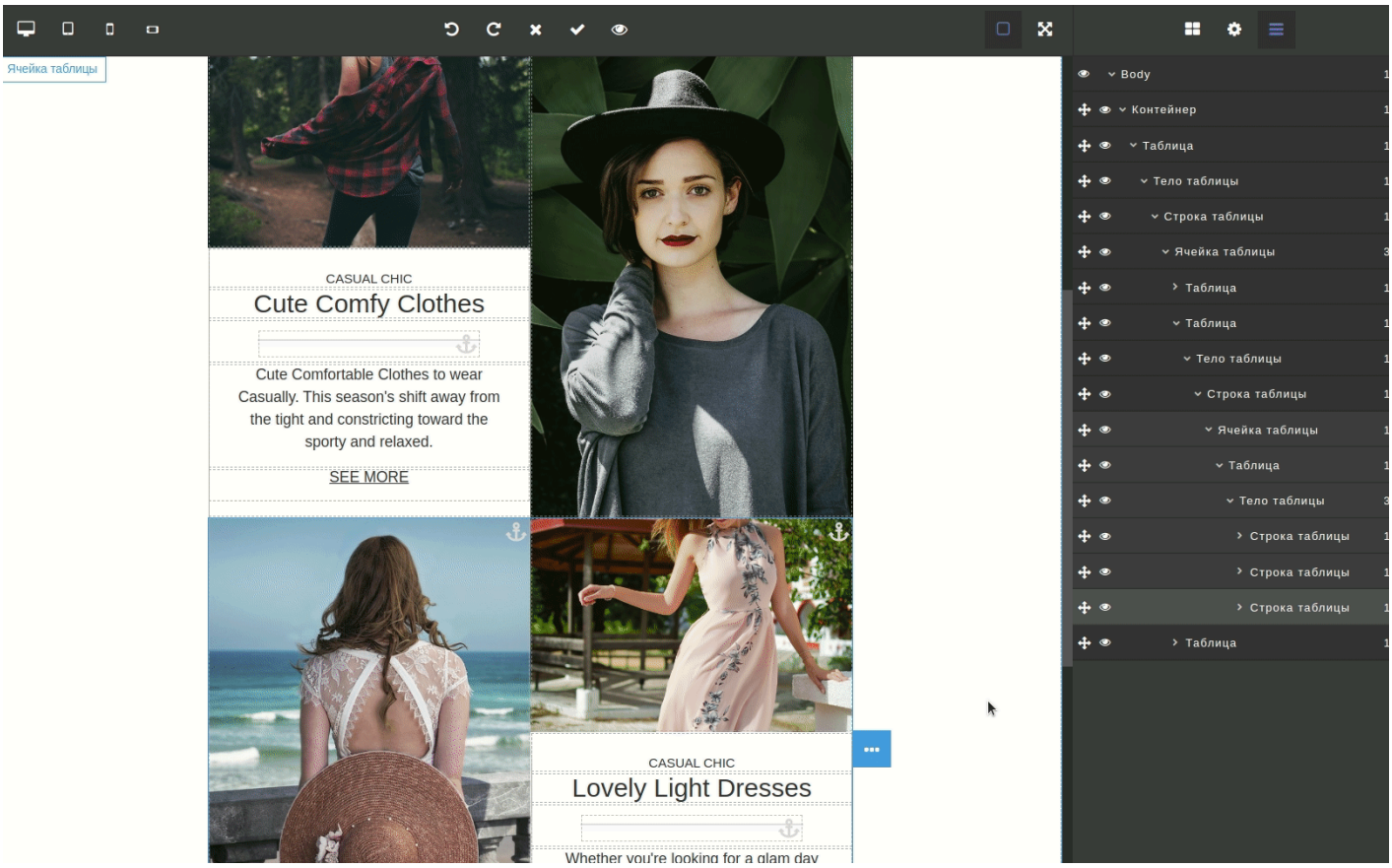
Если вы наведете курсор или кликните на элемент, то в шаблоне занимаемая им область подсветится синим контуром.



Итак, как же мы можем использовать структуру элементов?

(1) Изменяйте расположение элемента или целого контейнера. Кликните на

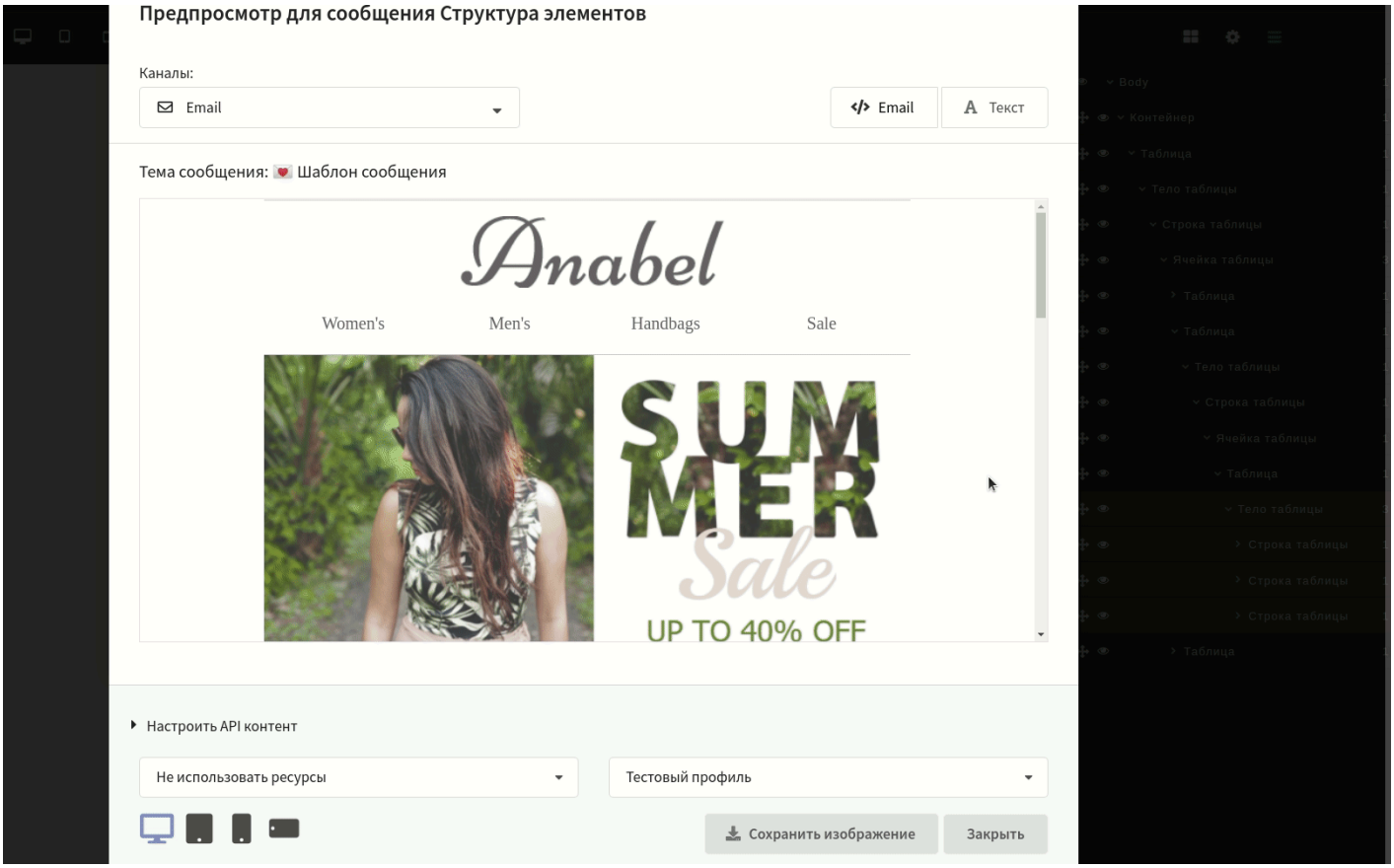
и перетащите элемент на новую позицию.



(2) Скрывайте элементы. Возможно, какой-то элемент письма или целый блок вам пока не нужен или вы хотите посмотреть, как шаблон будет выглядеть без него. Кликните на

, чтобы скрыть элемент. Если передумаете и захотите вернуть элемент, кликните на

.


Предпросмотр для сообщения Структура элементов
Каналы:
Email
Email
Текст
Тема сообщения: Шаблон сообщения
Anabel
Women's
Men's
Handbags
Sale
SUMMER
Sale
UP TO 40% OFF
Настроить API контент
Не использовать ресурсы
Тестовый профиль
Сохранить изображение
Закрыть

Переменные и функции Altcraft

Данные профиля

Описание	Переменная
Email-адрес	`{lead.email}`
Список телефонов	`{lead.phones}`
Имя	`{lead._fname}`
Фамилия	`{lead._lname}`
Дата рождения	`{lead._bdate}`
Пол	`{lead._sex}`
Дата регистрации	`{lead._regdate}`
IP-адрес регистрации	`{lead._regip}`
Город регистрации	`{lead._regcity}`
Страна регистрации	`{lead._regcountry}`
URL-адрес регистрации	`{lead._regurl}`
IP-адрес	`{lead._ip}`
Город	`{lead._city}`
Страна	`{lead._country}`
Регион	`{lead._region}`
Индекс	`{lead._postal_code}`
Временная зона	`{lead._tz}`
Поставщик	`{lead._vendor}`
Идентификатор профиля	`{lead._id}`
XXM-хеш	`{lead._xxm}`
XX-хеш	`{lead._xxh}`
MD5-хеш	`{lead._md5}`
*** Дополнительное поле**	**`{lead.название_поля}`**

Подписки

Переменные подписок содержат данные по **конкретной подписке** и по используемому **каналу отправки**. Например, с её помощью не получится передать номер телефона при отправке по email.

Описание	Переменная
Email-адрес подписки	`{subscription.email}`
Домен email-адреса	`{subscription.email_domain}`
Список email-адресов подписки	`{subscription.emails}`
Телефон подписки	`{subscription.phone}`
Список телефонов подписки	`{subscription.phones}`
Список push-подписок	`{subscription.pushes}`
MD5-хеш email-адреса	`{subscription.email_md5}`
MD5-хеш телефона	`{subscription.phone_md5}`
Hash подписки	`{subscription.hash_id}`
Основное поле подписки	`{subscription_field("channel" "field_name" resource_id)}` channel — строка с указанием на канал (“email”) field_name — имя поля для вставки (“domain”) resource — идентификатор ресурса (10)

Функция `subscription_field` позволяет вставлять поля из подписок, независимо от того, какой канал используется в текущем сообщении. Например, если у профиля есть SMS подписка, вы можете подставить его телефон в email версию сообщения:

`{subscription_field("sms" "phone")}`

Если под условия подходят несколько подписок, выбирается подписка с самым высоким приоритетом.

Доступные поля для подстановки:

Канал	Поле	Описание
email	email	Email-адрес подписки
	domain	Домен email-адреса подписки
	domain_group	Доменная группа
	md5	MD5-хеш email-адреса
sms	phone	Телефон подписки
	md5	MD5-хеш телефона
push	subscription_id	Идентификатор push-подписки
	bundle_id	Идентификатор приложения для iOS APNs
	provider	Push провайдер

TIP

Аналогично в шаблон можно подставлять данные из подписок пользовательских каналов. Поля основных каналов соответствуют sid полей, которые вы указываете при создании канала.

UTM

Если у профиля в его карточке есть UTM, то их можно подставить в шаблон сообщения.

Описание	Переменная
UTM-метка о рассылке	`{lead._utm_campaign}`
UMT-метка об источнике трафика	`{lead._utm_source}`
UMT-метка о типе контента	`{lead._utm_medium}`
UMT-метка о содержании контента	`{lead._utm_content}`
UMT-метка о ключевых словах	`{lead._utm_term}`

Внешний JSON

Внешний JSON-контент — это один из способов создавать динамические сообщения и автоматизировать работу. С его помощью в сообщение добавляется информация, генерируемая на вашем сервере точно в момент отправки каждого сообщения.

Подробнее о том, как работает внешний JSON можно прочесть в статье “Использование JSON-контента в сообщениях”.

В тексте сообщения JSON-контент представлен переменными типа `{json.variable}`. Типы данных `list` и ‘object’ поддерживают циклы, для типа `object` также доступно обращение к элементам по ключу `{json.object.key_name}`.

Маркет

Чтобы использовать информацию о заказах и товарах в рассылках, вы можете добавить переменные маркета в шаблон сообщения. Более подробно использование таких переменных описано в статье “Использование переменных маркета в сообщениях”.

Дата и время

Описание	Переменная	Пример
Форматированная дата	`{format}`	`{format datenow "%Y-%m-%D %H:%M:%S timezone, %I am"}` 2018-12-25 18:56:10 +03:00, 6 pm
Неформатированная дата	`{datenow}`	2018-12-25T18:56:10+0300

Описание	Переменная	Пример
Смещение даты	`{adddate()}`	`{adddate(datenow 0 0 3)}` 2018-12-27T18:56:10+0300
Полный год	`{yearlong}`	2018
Год	`{year}`	18
Месяц	`{month}`	12
День	`{day}`	10
Часы в 24-часовом формате	`{hours}`	18
Часы в 12-часовом формате	`{hours12}`	06
“До полудня”, “После полудня”	`{ampm}`	PM
Минуты	`{minutes}`	56
Секунды	`{seconds}`	10

Функции вывода

Описание	Переменная	Пример
Форматирование	format	`{format 1000000000 ","}` 1,000,000,000
Длина переменной	length	`{length("abc")}` 3
Массив значений	array	`{array[1 "2" 3.3]}` [1 2 3.3]
Случайный элемент массива	randomize	`{randomize(array[1 2 3])}` 1 **Пример**: в дополнительном поле профиля `Favourite_genres` хранится массив `["Детектив", "Юмор", "Фантастика"]`, выражение `{randomize(lead.Favourite_genres)}` выведет один случайный элемент, например `"Детектив"`. Работает с полями Телефоны, Теги и Массив объектов, а также с массивами API-контента и внешнего JSON-контента.

Описание	Переменная	Пример
Лимит массива	limit	`{limit(array[1 2 3] 2)}` [1 2]
Чётность числа	iseven	`{iseven(2)}` => true `{iseven(3)}` => false
Кодирование URL-адреса	urlencode	`{urlencode(lead._regurl)}` http:%2F%2Ftest.testdomain.com%2Fsomething Используется для кодирования недопустимых символов в URL параметрах.
Декодирование URL-адреса	urldecode	`{urldecode(lead._regurl)}` http://test.testdomain.com/something
Верхний регистр	uppercase	`{uppercase("text")}` TEXT
Первый символ в верхнем регистре	uppercasefirst	`{uppercasefirst("john")}` John

Шифрование

Шифрование переменных шаблона применяется когда необходимо передать данные вместе с кликом по URL на целевой сайт, но при этом данные не должны отражаться в журналах веб серверов или каким-то образом у пользователя. Например, в некоторых случаях данные предзаполнения форм достаточно закодировать в base58, чтобы на стороне сайта их раскодировать и подставить в форму.

Чтобы передавать контент шаблона и переменные в зашифрованном или хешированном виде, в редакторе шаблонов предусмотрена функция `crypt`. Чтобы получить доступ к ней, нажмите на кнопку `</>` в редакторе шаблонов и выберите **Шифрование**.

```
{crypt(алгоритм поле ключ)}
```

Можно изменить кодировку или зашифровать данные по одному из этих алгоритмов:

- base58
- base64
- sha1
- sha256
- aes
- blowfish

Для sha1, sha256, aes и Blowfish необходимо указать **ключ**, который будет применяться на другой стороне для дешифровки значения.

Детали шифрования **AES** и **Blowfish**:

	AES	Blowfish
Вектор инициализации	Первые 16 байт строки	Первые 8 байт строки
Режим шифрования	CFB (Cipher Feedback)	CBC (Cipher Block Chaining)
Длина ключа	128 бит — 16 символов	64 бит — 8 символов
Padding	нулевой	нулевой

TIP

Шифрование / дешифрование происходит через кодировку **Base64URL**.

Ниже приведены примеры синтаксиса для изменения кодировки и шифрования:

```
{crypt(base58 lead._fname)}
{crypt(base64 lead._lname)}
{crypt(sha256 lead.custom_field "encrypt_key")}
{crypt(sha1 lead._city "encrypt_key")}
{crypt(aes lead._fname "key1234567891234")}
{crypt(blowfish lead._fname "key12345")}
```

Если необходимо зашифровать строку текста, её нужно заключить в кавычки. Если число — кавычки не нужны:

```
{crypt(base58 "value")}
{crypt(base64 42)}
```

Для хеширования доступны следующие алгоритмы. Для каждого можно использовать соль:

- md5 + соль
- xxh32 + соль
- xxh64 + соль

Синтаксис такой же, как для функций шифрования:

```
{crypt(md5 subscription.email "salt")}
{crypt(xxh32 loyalty.new2.promocode "salt")}
{crypt(xxh64 json.text "salt")}
```

Обратите внимание, что зашифрованные данные уже будут закодированы в base64.

Логические выражения

Использование логических выражений описано в отдельной статье. Для составления простого условия используется:

```
{if lead.age gte 18}
        Если 18 или старше.
{else}
        Если младше 18.
{end}
```

Фрагменты

Фрагменты сообщения

Фрагменты — это блоки сообщений, которые могут использоваться в нескольких сообщениях. Например, это может быть хедер или футер сообщения, блок со ссылками на социальные сети — любой контент, который вы будете отправлять в рамках нескольких рассылок.

Для вызова фрагмента используйте `{fragment.<короткое_имя_фрагмента>}`.

Для вызова случайного фрагмента из диапазона: `{randomfragment[fragment.name1 fragment.name2 fragment.name3]}`.

Использование фрагментов более подробно описано в этой статье.

TIP: Фрагменты и части

В платформе также существует похожий функционал под названием **части**. Части отличаются от фрагментов тем, что настраиваются и используются в рамках одного письма. Чтобы начать работу с частями, добавьте в код шаблона переменную `{part.example_part}`. После этого внизу страницы появится вкладка “Контент”, где вы сможете настроить части шаблона.

Подробнее о частях можно прочесть здесь.

Параметры

Параметры

Описание	Переменная
ID отправленного сообщения (SMID)	`{send_message_id}`
Токен ресурса	`{resource_token}`
ID шаблона	`{msgid}`
ID рассылки	`{campid}`

Описание	Переменная
ID ресурса	`{resid}`
ID Базы данных (базы профилей)	`{listid}`
SUBID рассылки	`{subid}`
Имя ресурса	`{resname}`
Имя БД	`{listname}`
Тип сообщения	`{msg_type}`
URL ресурса	`{resurl}`
Домен URL ресурса	`{resdomain}`
Имя сообщения	`{msgname}`
Имя рассылки	`{campname}`
Имя отправителя SMS (SMS from name)	`{from_name_sms}`
ID сегмента	`{segid}`
Название сегмента	`{segname}`
Значение атрибута	`{attribute_value("attribute")}`, где вместо `"attribute"` — API-название атрибута

Служебные

Описание	Переменная
API-контент	`{apicontent.field_name}`
Отмена отправки сообщения	`{cancel}`

Трекинг

Описание	Переменная
Трекинг домен	`{trkdomain}`
Пиксель, фиксирующий открытие письма	`{pixel}`
Пиксель, фиксирующий чтение письма	`{read}`
Ссылка менеджера подписок	`{preferences}`
Ссылка отписки	`{unsubscribe}`
Ссылка глобальной отписки	`{globalunsubscribe}`
Добавление в стоп-список	`{suppress}`

Описание	Переменная
Ссылка на веб-версию	`{webversion}`

Сценарий

Описание	Переменная
Название сценария	`{workflow_name}`
ID сценария	`{workflow_id}`
ID узла сценария	`{node_id}`
Тип узла сценария	`{node_type}`

События

Переменные событий позволяют работать с данными о действиях пользователей (просмотры страниц, добавления в корзину, покупки и др.), которые собираются через пиксели и цели.

События привязаны к контексту:

- В **триггерных рассылках** переменная содержит только события, относящиеся к конкретному триггеру, который запустил рассылку.
- В **сценариях** переменная содержит список всех событий, которые произошли с профилем с момента входа в сценарий.

> **INFO**
>
> Эти переменные доступны только в триггерных рассылках и сценариях, которые запускаются по триггеру “Активация пикселя”.

Переменные

Описание	Переменная	Примечания
Первое событие	`{event.first}`	Самое старое событие в текущем контексте
Последнее событие	`{event.last}`	Самое новое событие в текущем контексте
Все события	`{event.all}`	Полный список событий в порядке обратном поступлению, где в [0] самое последнее событие
Кол-во событий	`{event.count}`	Количество событий в очереди

Как использовать в шаблонах

Чтобы **получить конкретные данные события**, добавьте к переменной нужный параметр через точку:

```
{event.first.goal} // цель первого события
{event.last.value} // значение последнего события
{event.first.geo_city} // город из первого события
{event.last.utm_source} // источник последнего события
{event.first.browser} // браузер из первого события
{event.first.categories} // категории из первого события
```

Чтобы **перебрать все события пользователя**, используйте цикл:

```
{for $index $event = event.all}
            {if $event.goal equal "cart"}
                  {formatdate $event.date "%Y-%m-%D в %H:%M"} в корзину добавлены
{$event.data.count}x {$event.data.product}.
              {end}
{else}
    <p>Нет событий...</p>
{end}
```

Примеры использования

Для триггерной рассылки после добавления в корзину:

```
Вы забыли товар: {event.first.data.product}
Цена: {event.first.data.price}
Купить: {event.first.data.link}
```

Для сегментации по геолокации:

```
В вашем городе {event.first.geo_city} проходит распродажа!
```

Для анализа эффективности рассылок:

```
Источник: {event.last.utm_source}
Рассылка: {event.last.utm_campaign}
Браузер: {event.last.browser}
```

Для вставки UTM-меток в ссылки:

```
<a href="https://site-example.ru/?utm_source={event.utm_source}&utm_medium=
{event.utm_medium}&utm_campaign={event.utm_campaign}&utm_content={event.utm_content}">
Перейти на сайт
</a>
```

Для использования в сценариях с узлом ожидания события:

```
Здравствуйте, {user.name}!

Вы недавно добавляли новый товар в корзину, но не оформили заказ.

Ваш товар ждёт вас:
{event.last.product_name}
```

TIP

Подробнее об элементе сценария “Ожидание события” можно прочесть здесь.

Параметры событий

Базовые поля событий

Параметр	Тип	Описание	Пример
`id`	string	Уникальный идентификатор события	`{event.first.id}`
`type`	string	Тип события платформы	`{event.last.type}`
`date`	DateTime	Время и дата события	`{event.first.date}`
`test`	bool	Признак тестового события	`{event.last.test}`

Геоданные

Параметр	Тип	Описание	Пример
`geo_country`	string	Страна (полное имя)	`{event.first.geo_country}`
`geo_tld`	string	Двухбуквенный код страны	`{event.first.geo_tld}`
`geo_region`	string	Регион	`{event.first.geo_region}`
`geo_city`	string	Город	`{event.first.geo_city}`
`geo_address`	string	Адрес	`{event.first.geo_address}`
`geo_zip`	string	Почтовый индекс	`{event.first.geo_zip}`
`geo_time_zone`	string	Часовой пояс	`{event.first.geo_time_zone}`
`geo_lat`	float	Широта	`{event.first.geo_lat}`
`geo_lon`	float	Долгота	`{event.first.geo_lon}`

UTM-метки

Параметр	Тип	Описание	Пример
`utm_source`	string	Источник трафика	`{event.first.utm_source}`
`utm_medium`	string	Канал трафика	`{event.first.utm_medium}`
`utm_campaign`	string	Название рассылки	`{event.first.utm_campaign}`
`utm_content`	string	Идентификатор контента	`{event.first.utm_content}`
`utm_term`	string	Ключевое слово	`{event.first.utm_term}`
`utm_keyword`	string	Ключевое слово (альтернативное)	`{event.first.utm_keyword}`

Событие пикселя (Pixel: goal)

Параметр	Тип	Описание	Пример
`pixel_id`	int	Идентификатор пикселя	`{event.first.pixel_id}`
`goal`	string	Название цели	`{event.first.goal}`
`value`	float	Значение цели	`{event.first.value}`
`data`	object	Произвольные данные цели (передаются в объекте `pixel_data`)	`{event.first.data.order_id}`
`browser`	string	Браузер пользователя	`{event.first.browser}`
`user_agent`	string	User-Agent браузера	`{event.first.user_agent}`
`accept_language`	string	Язык браузера	`{event.first.accept_language}`
`count_items`	int	Количество элементов	`{event.first.count_items}`
`categories`	string[]	Массив категорий	`{event.first.categories}`
`send_message_id`	string	ID отправки сообщения	`{event.first.send_message_id}`

Событие API-вызова (API: api_call)

Параметр	Тип	Описание	Пример
`content`	object	Данные из API-вызова	`{event.first.content.order_id}`
`custom_data`	object	Пользовательские данные API	`{event.first.custom_data.field_name}`

Типы событий

Группа	Тип	Описание
Pixel	`goal`	Активация пикселя, цели (включая API)
API	`api_call`	API-вызов через start сценария или триггера

Примечание: Все поля разных типов событий доступны на одном уровне. Например, `{event.last.content}` обращается к полю content события API-вызова, если это текущее событие.

TIP

В документации разработчика вы найдете полную техническую информацию о форматах данных и ограничениях. Это поможет точно настроить сбор данных через API и правильно использовать их в шаблонах.

Программы лояльности

При вставке переменных программ лояльностей через окно вставки в редакторе перед вами откроется контекстное меню, в котором вы сможете выбрать программу лояльности и тип баллов (а также лимит количества транзакций и период активации транзакций для последних двух переменных):

Описание	Переменная
Баланс баллов участника	`{loyalty_program.<ID_программы_лояльности>.points.<ID_баллов>.balance}`
Баланс активных баллов участника	`{loyalty_program.<ID_программы_лояльности>.points.<ID_баллов>.active_balance}`
Баланс неактивных баллов участника	`{loyalty_program.<ID_программы_лояльности>.points.<ID_баллов>.inactive_balance}`
Баланс с названием валюты	`{loyalty_program.<ID_программы_лояльности>.points.<ID_баллов>.balance_named}`
Баланс активных баллов с названием валюты	`{loyalty_program.<ID_программы_лояльности>.points.<ID_баллов>.active_balance_named}`
Баланс неактивных баллов с названием валюты	`{loyalty_program.<ID_программы_лояльности>.points.<ID_баллов>.inactive_balance_named}`
Список ближайших сгораний баллов (где x — лимит количества транзакций, а y — период активации транзакций)	`{loyalty_expiration_points(<ID_программы_лояльности> "<ID_баллов>" x y "desc")}`
Список ближайших активаций баллов (где x — лимит количества транзакций, а y — период активации транзакций)	`{loyalty_activation_points(<ID_программы_лояльности> "<ID_баллов>" x y "acs")}`

emoji

Вы также можете вставить в ваши шаблоны сообщений **emoji**. Для этого выберите соответствующий пункт в меню переменных и нажмите на нужный emoji.

Связи профилей

Общая формула переменной:

`{relation.[короткое_имя_связи].[direct для прямых или reverse для обратных].[имя_свойства].[count – количество, total – сумма, top – топ связей]}`

Количество прямых связей со свойством profit:

```
{relation.managers_loyal.direct.profit.count}
```

Топ обратных связей со свойством `profit`. Используется цикл для вывода нескольких значений:

```
{for $index $item = relation.managers_loyal.reverse.profit.top}
{$item.lead._fname} {$item.lead._lname} ({$item.value} profit)
{else} There is no ‘profit’ for you :(
{end}
```

Сумма значения свойства profit для прямой связи:

```
{relation.managers_loyal.direct.profit.total}
```

Дополнительные функции редактора

Форматирование цен

Также предусмотрена фунция форматирования числовых данных, которая делит число на блоки по три цифры заданным разделителем.

```
Price 8.000 RUR
```

```
<p>Price {format json.price "."} RUR</p>
```

Чтобы вставить функцию, нажмите на кнопку `</>` в редакторе шаблонов и выберите **Функции вывода** —> **Форматирование**.

Смещение даты

Функция `adddate(date, year, month, day)` возращает значение даты после того, как добавит к ней определённый интервал.

```
<p>Event starts on {adddate(lead.notify_date 0 0 10)}</p>
```

`adddate()` может использоваться внутри функции `format()`.

```
<p>Event starts on {format adddate(lead.notify_date 0 0 10) "%Y-%m-%D"}</p>
```

Чтобы вставить функцию, нажмите на кнопку `</>` в редакторе шаблонов и выберите **Дата и время** —> **Смещение дат**.

Одновременно форматировать и сместить дату можно при помощи функции:

```
{format adddate(datenow 0 0 3) "%Y-%m-%D"}
```

Форматирование даты

Данные типа “дата” имеют разное представление в разных странах. Для удобства подписчиков предусмотрена функция форматирования дат.

Например, для того, чтобы представить дату рождения подписчика в виде:

```
Дата рождения 02.11.1998
```

Используется такая функция:

```
<p>Дата рождения {format lead._bdate "%D.%m.%Y"}</p>
```

Одновременно форматировать и сместить дату можно при помощи функции:

```
{format adddate(datenow 0 0 3) "%Y-%m-%D"}
```

Все параметры форматирования даты:

```
# years
"%y": "06"
"%Y": "2006"
"yy": "06"
"YYYY": "2006"

# months
"%m": "01"
"MM": "01"
"%B": "January"
"month": "January"
"%ru_month": "январь"

# days
"%d": "2"
"%D": "02"
"DD": "_2"
"%A": "Monday"
"weekday": "Monday"

# hours
"%H": "15"
"%I": "3"
"hh": "15"
"hours": "15"
"twelve_h": "3"

# mins
"%M": "04"
"mm": "04"

# seconds
"%S": "05"
"ss": "05"

# am/pm indicator
"am": "pm"
"AM": "PM"
"%p": "pm"

# timezone
"tzone": "-07"
"timezone": "-07:00"
"%Z": "Z0700"
```

Чтобы вставить функцию, нажмите на кнопку `</>` в редакторе шаблонов и выберите **Дата и время** —> **Форматированная дата**.

INFO

Если в шаблоне используется контент из внешнего JSON (переменная apicontent или динамический JSON-контент), платформа может преобразовать строки в формате **RFC 3339** в дату и время.

Пример форматирования:

Режим	Формат
JSON	`"date_time": "2023-02-21T11:10:35.141Z"`
Шаблон	`{format apicontent.date_time "%Y-%m-%d %H:%M:%S"}`
Предпросмотр	`2023-02-21 11:10:35`

Использование логических выражений в сообщениях

TIP

Если вы используете несколько логических операторов в одном выражении `if`, обязательно заключайте каждую пару условий в скобки — это необходимо для корректного парсинга.

Пример корректного синтаксиса:

````
<html>
{if param1 and (param2 and (param3 and param4))} <!--правильно-->
if true
{else}
if false
{end}
</html>
```

**Пример, который вызовет ошибку:**

```
<html>
{if param1 and param2 and param3 and param4} <!--неправильно-->
if true
{else}
if false
{end}
</html>
```
````

В сообщениях можно формировать условные блоки, используя данные подписчика, json данные и прочие переменные.

```
{if lead.field in json.promo_actions1}
```

Они расширяют возможности персонификации сообщений, существенно сокращают объём работы.

Например, в сообщении используется подпись, которая меняется от подписчика к подписчику.

`Нашим лучшим клиентам скидка 42%` или **`Спецпредложение! Скидка 20% со второй покупки`!**

Вместо того, чтобы делать разные сообщения, можно использовать два условия:
```

1. Сначала проверяется, есть ли у подписчика поле `"best_buy_type"` и является ли он клиентом нужного магазина. Если нет, то он получает обычное предложение.
2. Затем проверяется значение в этом поле — совершал ли он хоть раз дорогую покупку.

```
<!-- Если в профиле подписчика есть поля "shop_name" и "best_buy_type" -->
{if lead.shop_name and lead.best_buy_type}

    {if lead.shop_name equal "my_shop"}

        <!-- Если подписчик покупал "premium" товары -->
        {if lead.best_buy_type contains "premium"}
                <p>Нашим лучшим клиентам скидка 42%.</p>

        <!-- Если он покупал что-то другое -->
        {else}
                <p>Спецпредложение! Скидка 20% со второй покупки!</p>
        {end}

        <!-- Если подписчик является клиентом другого магазина-->
        {else}
        <p>Спецпредложение! Скидка 20% со второй покупки!</p>
        {end}

<!-- На случай, когда поля в профиле не оказывется -->
{else}
    <p>Спецпредложение! Скидка 20% со второй покупки!</p>
{end}
```

Примеры использования

Условия

Простое условие

```
{if param1}
        if true
{else}
        if false
{end}
```

Комплексное условие

Пример 1:

```
{if param1 in array["par1" "par2"] and ( 5 equal length("par 3") or false )}
        if true
{else}
        if false
{end}
```

Пример 2 с `apicontent`:

```
{if apicontent.field_1 not_equal apicontent.field_2 and ( apicontent.field_1 or
apicontent.field_2 )}
        if true
{else}
        if false
{end}
```

Пример 3:

```
{if crypt("md5" lead._fname "salt") like "123b003aa39d110d1333b37af9889e10" and
crypt("base58" lead.lottery_id) like json.crypted_winner}
        if true
{else}
        if false
{end}
```

Пример 4:

```
{if iseven(lead.count) and iseven(lead.count2) equal false}
        if true
{else}
        if false
{end}
```

Проверка наличия поля и отмена отправки, если поля нет

```
{if lead.some_field}
some_field не пустое
{else}
{cancel}
{end}
```

Циклы

Простой цикл

```
{for array[1 "2" 3.3]}
        {.}
{else}
        if empty array
{end}
```

Соответствие значения

Равно

```
{if param1 equal param2}
        if true
{else}
        if false
{end}
```

Не равно

```
{if param1 not_equal param2}
        if true
{else}
        if false
{end}
```

Строка равна (не учитывается регистр)

```
{if param1 like param2}
        if true
{else}
        if false
{end}
```

Строка не равна (не учитывается регистр)

```
{if param1 not_like param2}
        if true
{else}
        if false
{end}
```

Строка содержит (не учитывается регистр)

```
{if param1 contains param2}
        if true
{else}
        if false
{end}
```

Строка не содержит (не учитывается регистр)

```
{if param1 not_contains param2}
        if true
{else}
        if false
{end}
```

Сравнение чисел и дат

Больше числа/даты

```
{if 0 gt 1}
        if true
{else}
        if false
{end}
```

Больше или равно числа/даты

```
{if 0 gte 1}
        if true
{else}
        if false
{end}
```

Меньше числа/даты

```
{if 0 lt 1}
        if true
{else}
        if false
{end}
```

Меньше или равно числа/даты

```
{if 0 lte 1}
        if true
{else}
        if false
{end}
```

Массивы и [поля с тегами]

Элемент в массиве

```
{if param1 in array["par1" "par2"]}
        if true
{else}
        if false
{end}
```

Элемент в поле тэгов

```
{if param1 in lead.TagsField}
        if true
{else}
        if false
{end}
```

Элемент не в массиве

```
{if param1 not_in array["par1" "par2"]}
        if true
{else}
        if false
{end}
```

Элемент не в поле тэгов

```
{if param1 not_in lead.TagsField}
        if true
{else}
        if false
{end}
```

Все элементы первого массива

```
{if array["par1" "par2"] all_in array["par1" "par2"]}
        if true
{else}
        if false
{end}
```

Не все элементы первого массива

```
{if array["par1" "par2"] not_all_in array["par1" "par2"]}
        if true
{else}
        if false
{end}
```

Хотя бы одно совпадение в массивах

```
{if array["par1" "par2"] any_of array["par1" "par2"]}
        if true
{else}
        if false
{end}
```

Нет совпадений в массивах

```
{if array["par1" "par2"] not_any_of array["par1" "par2"]}
        if true
{else}
        if false
{end}
```

Даты

Равен дню

```
{if "2006-01-01" day_equal "1"}
        if true
{else}
        if false
{end}
```

Равен месяцу

```
{if "2006-01-01" month_equal "1"}
        if true
{else}
        if false
{end}
```

Равен году

```
{if "2006-01-01" year_equal "2006"}
        if true
{else}
        if false
{end}
```

Равен месяцу и году

```
{if "2006-01-01" yearmonth_equal "2006-01"}
        if true
{else}
        if false
{end}
```

Использование циклов в сообщениях

В сообщении реализована поддержка циклов для JSON переменных типа список `(list) []` и объект `(object) {}`, а также обращение к элементу объекта по ключу `{json.object.name}`.

Если в качестве переменной в шаблон вставляются дополнительные поля профиля, то поля типа “Массив объектов” и “Тег” (список строк) тоже могут использоваться в циклах.

Также возможно использовать вложенные циклы.

Пример переменных

```
{
  "Addresses": [
    "улица Барклая, дом 444, корпус 24",
    "улица Суворова, дом 555, корпус 25"
  ],
  "Information": {
    "LJ": [
      {"name": "Optimyss X1", "price": "$10"},
      {"name": "Optimyss X2", "price": "$20"}
    ],
    "Samsung": [
      {"name": "Galaktiko 17", "price": "$50"}
    ]
  }
}
```

Пример циклов

```
<h5>Наши новинки:</h5>

<!-- Цикл по элементам объекта -->
{for $vendor $models = json.Information}
    <p>{$vendor}</p>

    <ol>
        <!-- Вложенный цикл -->
        {for $index $model = $models}

            <!-- Обращение к элементам по ключу -->
            <li>{$model.name} {$model.price}</li>

        {else}
            <p>Empty array!</p>
        {end}
    </ol>

{else}
    <p>Empty array!</p>
{end}

<h5>Адреса:</h5>
<ol>
<!-- Простой цикл по списку строк -->
{for $index $address = json.Addresses}
    <li>{$address}</li>
{else}
    <p>Empty array!</p>
{end}
</ol>
```

Результат

```
<h5>Новинки:</h5>

    <!-- Цикл по элементам объекта -->
    <p>LJ<p>
    <ol>
        <!-- Вложенный цикл, Обращение к элементам по ключу -->
        <li>Optimyss X1 $10</li>
        <li>Optimyss X2 $20</li&gt
    <ol>

    <!-- Цикл по элементам объекта -->
    <p>Samsyng<p>
    <ol>
        <!-- Вложенный цикл, Обращение к элементам по ключу -->
        <li>Galaktiko 17 $50</li>
    <ol>

<h5>Адреса:</h5>
    <ol>
        <!-- Простой цикл по списку строк -->
        <li>улица Барклая, дом 444, корпус 24</li>
        <li>улица Суворова, дом 555, корпус 25</li>
    </ol>
```

Использование переменных маркета в сообщениях

Переменные

В маркете доступны четыре переменные:

- `products` — функция выбирает продукты.

```
{products("<filter>" 10 "manufacturer_name" "desc")}
```

- `sku` — функция выбирает SKU.

```
{sku("<filter>" 10 "manufacturer_name" "desc")}
```

- `orders` — функция выбирает заказы из списка заказов профиля.

```
{orders("<filter>" 10 "status" "asc")}
```

- `order` — функция выбирает один заказ из списка заказов профиля.

```
{order("<filter>")}
```

Аргументы функции

`("<filter>" 10 "manufacturer_name" "desc")`

Аргумент	Обязательный	Описание
`"<filter>"`	Да	Фильтр — это строка условий. Позволяет выбрать продукты, SKU, заказы или позиции заказов, соответствующие заданным условиям. Например, можно выбрать продукты одного производителя или заказы, превышающие конкретную сумму. Подробную информацию смотрите ниже.
`10`	Да	Лимит. Сколько элементов массива необходимо перебрать. Максимальное значение — 30. Например, если указать значение 10, то фильтр выберет первые десять элементов из массива.

Аргумент	Обязательный	Описание
`"manufacturer_name"`	Нет	Сортировка объектов по указанному полю. По умолчанию — без сортировки.
`"desc"`	Нет	Направление сортировки. **`asc`** — по возрастанию (сортировка по умолчанию) **`desc`** — по убыванию

Фильтр

Фильтр состоит из полей и операторов. Поля есть у каждого объекта в маркете. Например, у продуктов и SKU есть поле `name` (наименование товара), а у заказов есть поле `total_price` (общая стоимость заказа). В фильтре вы указываете поле и значение, которому оно должно соответствовать. Если поле объекта соответствует значению, то этот объект будет выбран и его данные можно будет подставить в шаблон.

Пример фильтра

Добавим в шаблон переменную “Заказы”:

```
{orders (("order.total_price == '1000 RUB') && (order.custom_fields.field ==
'lead._region')" 10 "id" "desc")}
```

Выделенная часть переменной является фильтром. Разберем его по частям:

- `order.total_price == '1000 RUB'` — условие выберет все заказы, стоимость которых равна 1000 рублей.
- `order.custom_fields.field == 'lead._region'` — условие выберет заказы с дополнительным полем `"field"`. Вместо `"field"` можно указать любое дополнительное (пользовательское) поле. Указанное равенство означает то, что значение поля `"field"` должно совпадать со значением региона в профиле клиента (`"lead._region"`).
- Оператор `&&` указывает на то, что оба условия должны быть выполнены.
- Число 10 означает количество выводимых объектов, `"id"` указывает на поле фильтрации, а `"desc"` — на сортировку по убыванию.

Ниже приведены поля и операторы, доступные в фильтре.

Операторы и функции в фильтре

Оператор	Описание	Пример	Что будет выбрано
>	больше	`order.delivery_cost > '300 RUB'`	Заказы со стоимостью доставки больше 300 рублей
>=	больше или равно	`order.delivery_cost >= '300 RUB'`	Заказы со стоимостью доставки 300 рублей или больше
<	меньше	`sku.count < '10'`	SKU, которых осталось меньше 10 штук для продажи
<=	меньше или равно	`sku.count <= '10'`	Количество SKU для продажи равно 10 или больше

`==`	равно	`order.status == '1'`
		
		`order.custom_fields('дата') == '2024-12-25'`
			Заказы со статусом “новый”
			
			Заказы с датой 25 декабря 2024

!=	не равно	`product.manufacturer_name != 'Apple'`	Продукты, производителем которых не является Apple
&&	И (оба условия соответствуют)	`product.manufacturer_name == 'Apple' && product.price < '200 USD'`	Продукты производителя Apple со стоимостью больше $200
||	ИЛИ (одно из условий соответствует)	`sku.manufacturer_name == 'MSI'` || `sku.manufacturer_name == 'Acer'`	SKU с производителем MSI или Acer
`in('<field_name>', 'value1', 'value2'...'valueN')`	Значение поля (`<field_name>`) совпадает с одним из указанных значений (`'value1', 'value2'`)	`in(product.external_id,'1','2','3')`	Продукты с идентификатором равным 1, 2 или 3

Оператор	Описание	Пример	Что будет выбрано
`contains('<field_name>', 'value1', 'value2'...'valueN')`	Один из элементов массива совпадает с одним из указанных значений. `<field_name>` — массив `'value1', 'value2'` — значения, совпадение с которыми проверяется. Используется, если поле является массивом, т.е. содержит несколько элементов (например: категории, изображения, дополнительные поля).	`contains(product.categories,'smartphone')`	Продукты с категорией “smartphone”
`len('<field_name>', <оператор>, <число> )`	Проверяет длину массива. Сравнивает количество элементов в массиве с указанным числом. `<field_name>` — поле-массив (например, `product.pictures`) <оператор> — оператор сравнения (`>`, `>=`, `<`, `<=`, `==`, `!=`) <число> — целое число для сравнения.	`len(product.pictures, !=, 0)`	Продукты, у которых количество картинок не равно нулю (т.е. есть хотя бы одна картинка)

Работа с пользовательскими полями

Для фильтрации по пользовательским полям доступно два синтаксиса:

Универсальный синтаксис (рекомендуется)

```
product.custom_fields('external_id')
```

INFO

Мы рекомендуем использовать универсальный синтаксис `product.custom_fields('external_id')`, потому что он поддерживает любые типы идентификаторов — как числовые, так и строковые, включая специальные символы. Этот вариант гарантирует корректную работу фильтрации независимо от формата `external_id` вашего пользовательского поля.

Примеры:

```
{for $product in products("product.custom_fields('123') == 'красный'")}
{for $product in products("in(product.custom_fields('color'), 'red', 'blue')")}
{for $product in products("sku.custom_fields('45') > '100'")}
```

Строковый синтаксис (ограниченный)

```
product.custom_fields.field_name
```

INFO

Старый синтаксис `product.custom_fields.field_name` можно использовать только при условии, что `external_id` поля состоит исключительно из буквенных символов и не содержит цифр, дефисов или других специальных знаков.

Пример:

```
{for $product in products("product.custom_fields.brand == 'Apple'")}
```

Список доступных полей в фильтрах

Продукты

Префикс — **`product.`**

Поле	Описание
`product.external_id`	Идентификатор продукта
`product.name`	Название продукта
`product.manufacturer_name`	Название производителя
`product.url`	URL страницы продукта на сайте магазина
`product.pictures`	URL-ссылка на картинку продукта
`product.count`	Количество продуктов, доступное для продажи
`product.is_available`	Есть ли продукт в наличии у магазина
`product.delivery`	Возможность курьерской доставки (по всем регионам, в которые доставляет магазин)
`product.barcode`	Список штрихкодов продукта от производителя
`product.categories`	Идентификатор категории продукта
`product.custom_fields.<external_id поля>`	Идентификатор дополнительного поля. Для числовых ID используйте `custom_fields('123')`

SKU

Префикс — **`sku.`**

Поле	Описание
`sku.external_id`	Идентификатор SKU
`sku.name`	Название SKU
`sku.manufacturer_name`	Название производителя
`sku.url`	URL страницы SKU на сайте магазина
`sku.pictures`	URL-ссылка на картинку SKU
`sku.count`	Количество SKU, доступное для продажи
`sku.is_available`	Есть ли SKU в наличии у магазина
`sku.delivery`	Возможность курьерской доставки (по всем регионам, в которые доставляет магазин)
`sku.barcode`	Список штрихкодов SKU от производителя
`sku.categories`	Идентификатор категории SKU
`sku.custom_fields.<external_id поля>`	Идентификатор дополнительного поля. Для числовых ID используйте `custom_fields('123')`

Заказы

Префикс — **`order.`**

Поле	Описание
`order.external_id`	Идентификатор заказа
`order.status`	Идентификатор статуса заказа 1 — новый 2 — доставлен 3 — оплачен 4 — отменен 5 — частичный возврат
`order.delivery_cost`	Стоимость доставки
`order.total_price`	Общая стоимость заказа. Складывается из суммы каждой позиции в заказе и стоимости доставки.
`order.custom_fields.<external_id поля>`	Идентификатор дополнительного поля. Для числовых ID используйте `custom_fields('123')`
`order.real_create_time`	Время создание заказа
`order.real_update_time`	Время обновления заказа
`order.region_external_id`	Идентификатор региона

Сортировка по пользовательским полям

Иногда требуется выполнить сортировку объектов по значениям **пользовательских полей** (объект `custom_fields`).
Пользовательские поля хранятся в базе в виде **JSON-строки**, поэтому для доступа к полю внутри JSON при **сортировке** можно использовать оператор `->`.

Пример обращения к ключу:

```
custom_fields -> 'priority'
```

Использование в параметре сортировки функции (на примере `products`):

```
{for $index $el = products("contains(product.categories,'smartphone')" 3 "custom_fields ->
'priority'" "desc")}
{$el.name} — {$el.price}
{else}
if empty array
{end}
```

В этом примере список продуктов предварительно **фильтруется** по категории `smartphone`, затем **сортируется** по значению ключа `priority` внутри `custom_fields` по убыванию, и выводятся первые 3 результата.

INFO

Важно

- Оператор `->` используется **только** в параметре сортировки для обращения к ключу JSON.
- Для **фильтрации** по пользовательским полям применяется точечная нотация из раздела “Фильтр”, например: `order.custom_fields.field == 'value'`.
- Оператор `->` **не является оператором маркета** — это оператор базы данных (PostgreSQL), к которой выполняется запрос, поэтому такой способ сортировки следует рассматривать как **обходное решение**.

Как использовать переменные маркета

Любая переменная в маркете представляет собой массив. Например, переменная `products` будет содержать массив всех продуктов, загруженных в платформу, а переменная `orders` — массив заказов одного профиля. Чтобы перебрать массив, воспользуйтесь циклом с переменными.

Перебор массива продуктов (products)

Выводит в письме 5 продуктов от Apple:

```
Возможно, Вам будут интересны эти товары:
{for $index $el = products("product.manufacturer_name == 'Apple'" 5 "external_id" "desc")}
{$el.name} — {$el.price}
{else}
if empty array
{end}
```

7 самых дешевых смартфонов:

```
Лучшие цены на смартфоны для Вас и Ваших близких:
{for $index $el = products("product.categories == 'smartphone'" 7 "price" "asc")}
{$el.name} — {$el.price}
{else}
if empty array
{end}
```

Товары с доставкой:

```
Доставим до двери:
{for $index $el = products("product.delivery == true" 10 "name" "desc")}
{$el.name} — {$el.price}
{else}
if empty array
{end}
```

Перебор массива SKU (sku)

Смартфоны до 20000:

```
Смартфоны, которые Вам понравятся:
{for $index $el = sku("contains(sku.categories,'smartphone') && sku.price < '20000 RUB'" 3
"price" "asc")}
{$el.name} — {$el.price}
{else}
if empty array
{end}
```

IPhone с объемом памяти больше 128 ГБ:

```
Памяти хватит на всё:
{for $index $el = sku("contains(sku.name,'IPhone') && sku.custom_fields.memory >= '128'" 10
"name" "desc")}
{$el.name} — {$el.price}
{else}
if empty array
{end}
```

Перебор массива всех заказов (orders)

Заказы, прибывшие в пункт выдачи:

```
Заказы ждут Вас в пункте выдачи:
{for $index $el = orders("order.status == '2'" 10 "external_id" "desc")}
Заказ № {$el.external_id}
К оплате: {$el.total_price}
{else}
if empty array
{end}
```

Список новых заказов:

```
Новые заказы за последнюю неделю:
{for $index $el = orders("order.status == '1'" 20 "external_id" "desc")}
Заказ № {$el.external_id}
К оплате: {$el.total_price}
{else}
if empty array
{end}
```

Перебор позиций в заказе с id = ‘0001’ (order + lines)

```
{for $index $el = order("order.external_id == '0001'")}
 {if $index equal "lines"}
  {for $i $line = $el}
    Название позиции: {$line.name}
    Цена позиции: {$line.final_price_of_line}
  {end}
 {end}

 {if $index equal "delivery_cost"}
   Стоимость доставки: {$el}
 {end}
 {if $index equal "total_price"}
   К оплате: {$el}
 {end}

{else}
if empty array
{end}
```

Список доступных полей для подстановки в шаблон

Продукты и SKU (products/sku)

Поле для подстановки	Описание
external_id	Идентификатор товара.
name	Название товара
manufacturer_name	Наименование производителя
url	URL страницы товара на сайте магазина
pictures	Список URL-ссылок на изображения товара (массив)
count	Количество товаров, доступное для продажи
available	Есть ли товар в наличии у магазина
delivery	Возможность курьерской доставки (по всем регионам, в которые доставляет магазин)
barcode	Список штрихкодов продукта от производителя (массив)
categories	Список идентификаторов категорий товара (массив)
custom_fields.<external_id поля>	Идентификатор дополнительного поля. Для числовых ID и специальных символов используйте синтаксис custom_fields(‘external_id’)
price	Актуальная цена продукта
old_price	Старая цена продукта

Поле для подстановки	Описание
regional_data. <region_external_id>.count	Количество товаров, доступное для продажи в указанном регионе. `regional_data` — это данные о продукте, которые могут различаться в зависимости от региона (в формате массива). `<region_external_id>` — идентификатор региона.
regional_data. <region_external_id>.delivery	Возможность курьерской доставки по указанному региону
regional_data. <region_external_id>.availabl e	Есть ли товар в наличии в указанном регионе
regional_data. <region_external_id>.price	Цена товара в указанном регионе

Заказы (orders)

Поле для подстановки	Описание
external_id	Идентификатор заказа
status	Идентификатор статуса заказа
delivery_cost	Стоимость доставки
total_price	Общая стоимость заказа. Складывается из суммы каждой позиции в заказе и стоимости доставки.
custom_fields. <external_id поля>	Идентификатор дополнительного поля. Для числовых ID и специальных символов используйте синтаксис custom_fields(‘external_id’)
create_time	Время создание заказа
update_time	Время обновления заказа
region.external_id	Идентификатор региона
lines	Позиции заказа (массив). Только для функции `order`.

Позиции заказа (lines)

Поле для подстановки	Описание
id	Идентификатор позиции
external_id	Дополнительный идентификатор позиции
custom_status_ex ternal_id	Идентификатор статуса позиции заказа

Поле для подстановки	Описание
base_price_per_item	Базовая цена
final_price_of_line	Конечная цена
count	Количество товара в позиции
custom_fields.<external_id поля>	Дополнительные поля позиции заказа. Идентификатор дополнительного поля в позиции заказа. Для числовых ID и специальных символов используйте синтаксис custom_fields('external_id')
product_external_id	Идентификатор продукта в позиции
name	Название позиции
manufacturer_name	Наименование производителя товара, указанного в позиции
url	URL страницы товара, указанного в позиции, на сайте магазина
pictures	Список URL-ссылок на изображения товара, указанного в позиции (массив)
balance	Остаток товара
available	Есть ли товар, указанный в позиции, в наличии у магазина
delivery	Возможность курьерской доставки позиции заказа
barcodes	Список штрихкодов товара, указанного в позиции (массив)
categories	Список идентификаторов категорий товара, указанного в позиции (массив)

INFO

Приоритет переопределения пользовательских полей:
(определяет значение каких полей будет применятся в первую очередь)

SKU > Продукт > Позиция заказа

Использование функционала JSONPath

Платформа Altcraft поддерживает функционал языка запросов **JSONPath** при работе с JSON-контентом. Синтаксис JSONPath позволяет быстро и просто выбирать объекты внутри кода и использовать их в шаблонах сообщений. Его можно использовать при работе с **API-контентом** и с **внешним JSON-контентом**.

API-контент позволяет менять содержание триггерного сообщения при каждой отправке. Необходимую переменную можно указать в триггерном API-запросе. Подробнее о том, как запускать триггерные рассылки и как работать с API-контентом в сообщениях, можно узнать здесь.

Для подключения внешнего JSON, перейдите в раздел “Опции” во время создания или редактирования шаблона и укажите ссылку на источник динамического JSON-контента. Платформа самостоятельно определит поля внешнего JSON-файла. Более подробную информацию об использовании JSON при работе с шаблонами сообщений вы можете найти здесь.

Аргументы функции

Переменная JSONPath на платформе Altcraft указывается в следующем виде:

```
{jsonpath(json.<корневой_объект> "<путь_к_объекту>" "<формат вывода>")}
```

Аргумент	Описание
Корневой объект	Обращение к JSON-объекту, в котором находится нужный параметр или объект
Путь к объекту	Путь к искомому объекту; может состоять из разделённых точками сегментов, логических операторов, фильтров или функций
Формат вывода	Конечная переменная может быть представлена в двух вариантах: **item** (объект) или **array** (массив). `Item` выведет первый найденный по запросу результат, а если результата нет, то вернёт пустую строку. `Array` возвращает массив всех найденных результатов. Этот режим всегда выводит результат в виде массива, даже если результат один или отсутствует.

Примеры использования

Рассмотрим примеры использования JSONPath. Предположим, что мы имеем JSON-файл со следующим содержанием:

```
{
  "store": {
    "book": [
      {
        "category": "справочная литература",
        "author": "Найджел Риз",
        "title": "Высказывания Века",
        "price": 8.95,
        "images": ["AbeLincoln.png","Churchill.jpg","Thatcher.png"]
      },
      {
        "category": "художественная литература",
        "author": "Ивлин Во",
        "title": "Меч почёта",
        "price": 12.99,
        "images": ["MenAtArms.png","Gentlemen.jpg","Officers.png"]
      },
      {
        "category": "художественная литература",
        "author": "Герман Мелвилл",
        "title": "Моби Дик",
        "isbn": "0-553-21311-3",
        "price": 8.99,
        "images": ["Ahab.png","Ishmael.jpg","whale.png"]

      },
      {
        "category": "художественная литература",
        "author": "Дж.Р.Р. Толкин",
        "title": "Властелин Колец",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ]
  }
}
```

Благодаря функционалу JSONPath небольшое выражение может вернуть **массив**, содержащий всех авторов книг:

`{jsonpath(json.store "$.book[*].author" "array")}`

Другая функция вернёт **один объект** — автора первой книги:

`{jsonpath(json.store "$.book[*].author" "item")}`

Можно построить выраженее посложнее, например, для того, чтобы узнать название книги, цена которой выше 10, мы составим следующий код:

`{jsonpath(json.store "$.book[?(@.price > 10)].title" "item")}`

Чтобы узнать среднее арифметическое значение цены всех книг, используем такую функцию:

`{jsonpath(json.store "$.book[*].price.avg()" "item")}`

Переменные JSONPath также могут работать в циклах:

```
{for $index $imgs = jsonpath(json.store "$.book[*].images" "array")}
    <li>{$imgs}</li>
{else}
    <p>Empty array!</p>
{end}
```

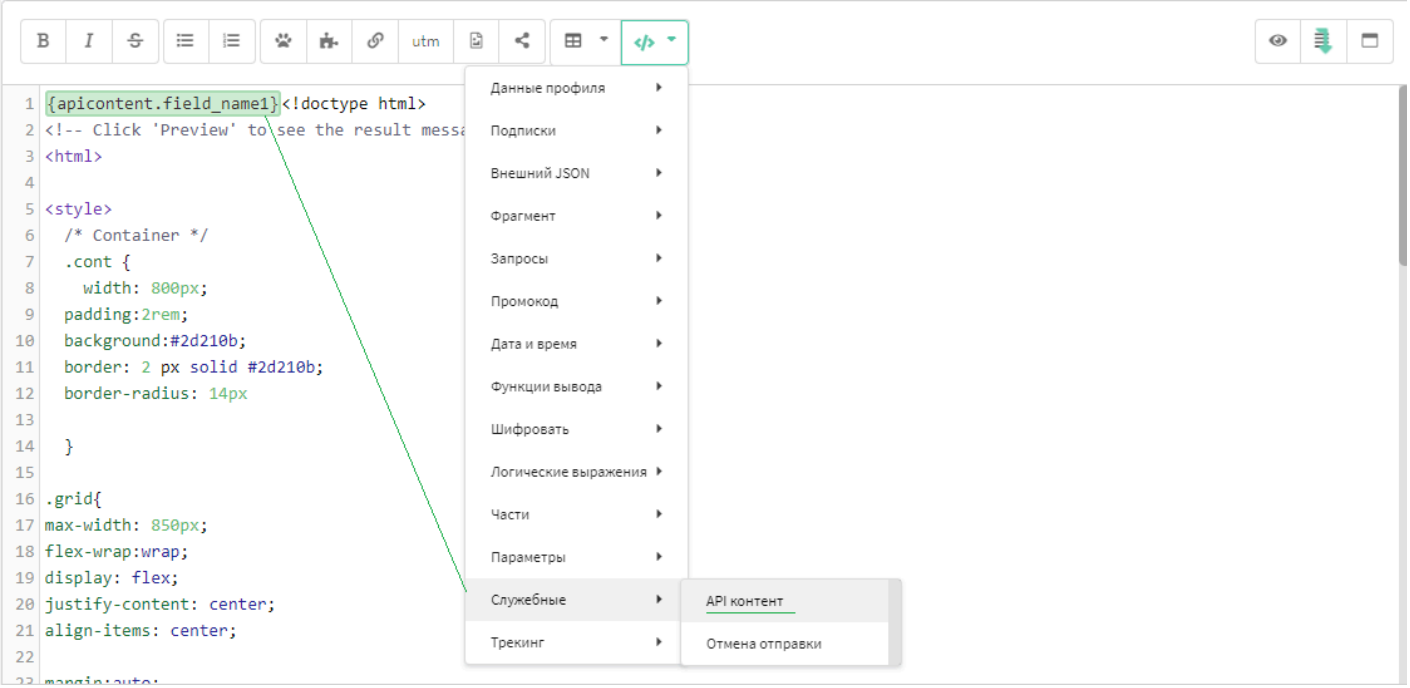
Полный функционал JSONPath в платформе Altcraft Marketing указан в таблице ниже:

Аргумент	Описание	Пример использования	Результат
`$`	Корневой элемент	`{jsonpath(json.store "$.book[0].title" "item")}`	Высказывания Века
`@`	Текущий элемент	`{jsonpath(json.store "$.book[?(@.price > 10)].title" "item")}`	Меч почёта
`..`	Рекурсивный спуск	`{jsonpath(json.store "$..title" "array")}`	[Высказывания Века Меч почёта Моби Дик Властелин Колец]
`*`	Соответствие всем объектам	`{jsonPath(json.store "$.book[*].price" "array")}`	[8.95 12.99 8.99 22.99]
`?()`	Фильтр	`{jsonpath(json.store "$.book[?(@.category == 'справочная литература')].title" "item")}`	Высказывания Века
Логический оператор для фильтрации	**Описание**	**Пример использования**	**Результат**
`==`	Равно	`{jsonpath(json.store "$.book[?(@.category == 'справочная литература')].title" "item")}`	Высказывания Века
`!=`	Не равно	`{jsonpath(json.store "$.book[?(@.category != 'справочная литература')].title" "array")}`	[Меч почёта Моби Дик Властелин Колец]
`<`	Меньше	`{jsonpath(json.store "$.book[?(@.price < 10)].title" "array")}`	[Высказывания Века Моби Дик]
`>`	Больше	`{jsonpath(json.store "$.book[?(@.price > 10)].title" "array")}`	[Меч почёта Властелин Колец]
`<=`	Меньше или равно	`{jsonpath(json.store "$.book[?(@.price >= 12.99)].title" "array")}`	[Меч почёта Властелин Колец]

`>=`	Больше или равно	`{jsonpath(json.store "$.book[?(@.price <= 12.99)].title" "array")}`	[Высказывания Века Меч почёта Моби Дик]
Функция	**Описание**	**Пример использования**	**Результат**
`max`	Максимальное значение в массиве	`{jsonPath(json.store "$.book[*].price.max()" "item")}`	22.99
`min`	Минимальное значение в массиве	`{jsonPath(json.store "$.book[*].price.min()" "item")}`	8.95
`first`	Первый элемент массива	`{jsonPath(json.store "$.book[*].title.first()" "item")}`	Высказывания Века
`last`	Последний элемент массива	`{jsonPath(json.store "$.book[*].title.last()" "item")}`	Властелин Колец
`sum`	Сумма чисел в массиве	`{jsonPath(json.store "$.book[*].price.sum()" "item")}`	53.92
`avg`	Среднее значение из чисел в массиве	`{jsonPath(json.store "$.book[*].price.avg()" "item")}`	13.48
`length`	Количество элементов в массиве	`{jsonPath(json.store "$.book.length()" "item")}`	4

Использование API-контента в сообщениях

При отправке сообщений с помощью API Call триггера, начиная с версии 1.1, возможно передавать данные для сообщения вместе с API-запросом.



С помощью `apicontent` одно сообщение может автоматически персонализироваться под каждого клиента. В `apicontent` можно передать информацию о совершенных операциях, покупках, истории клиента, т.д. Данная информация передается в виде JSON-объекта.

Пример того, как может выглядеть объект для `apicontent`:

```
{
  "product_id": 11,
  "product_name": "Графический дизайн: курс для начинающих",
  "product_price": 1000
}
```

Способы передачи API-контента в платформу

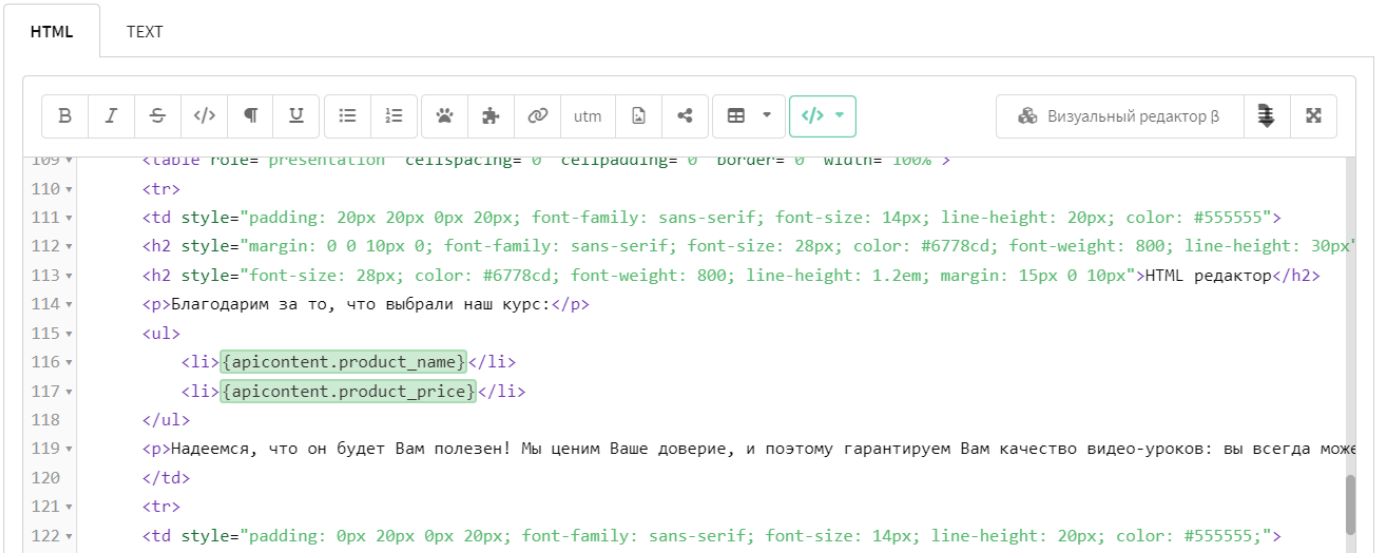
Через API-запрос

Данные в переменную `apicontent` можно передать, запустив триггерную рассылку или сценарий с помощью API-запроса. В запросе необходимые данные передаются в поле `content`. Ниже приведен список запросов, в которых доступно поле `content`:

- Запуск триггерной рассылки
- Импорт профиля + Отправка триггера
- Массовая отправка триггера
- Массовый импорт профилей + отправка триггера
- Отправить профиль клиента в сценарий

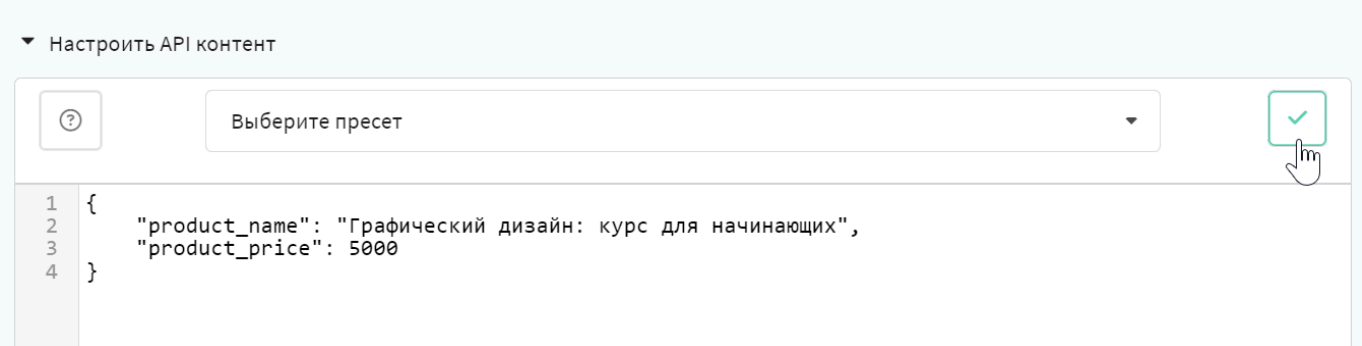
- Одновременный импорт и запуск профиля в сценарий
- Массовый импорт и запуск профилей в сценарий

Далее эти данные можно подставить в шаблон с помощью переменной `apicontent`:



TIP

Предпросмотр сообщения с `apicontent` доступен только после настройки раздела **“Настроить API-контент”**. Для этого перейдите в раздел предпросмотра и вставьте объект, содержащий `apicontent`, в редактор. Нажмите на галочку, чтобы применить изменения.



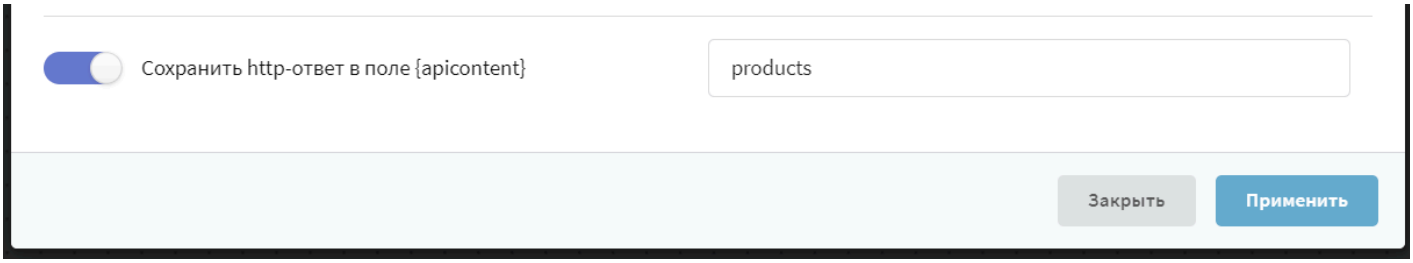
Через элемент “API Запрос” в сценарии

В элементе сценария **“API Запрос”** вы можете получить данные из внешнего источника с помощью запроса и сохранить http-ответ в переменную `apicontent`. Информация из запроса в дальнейшем может использоваться в сценарии.

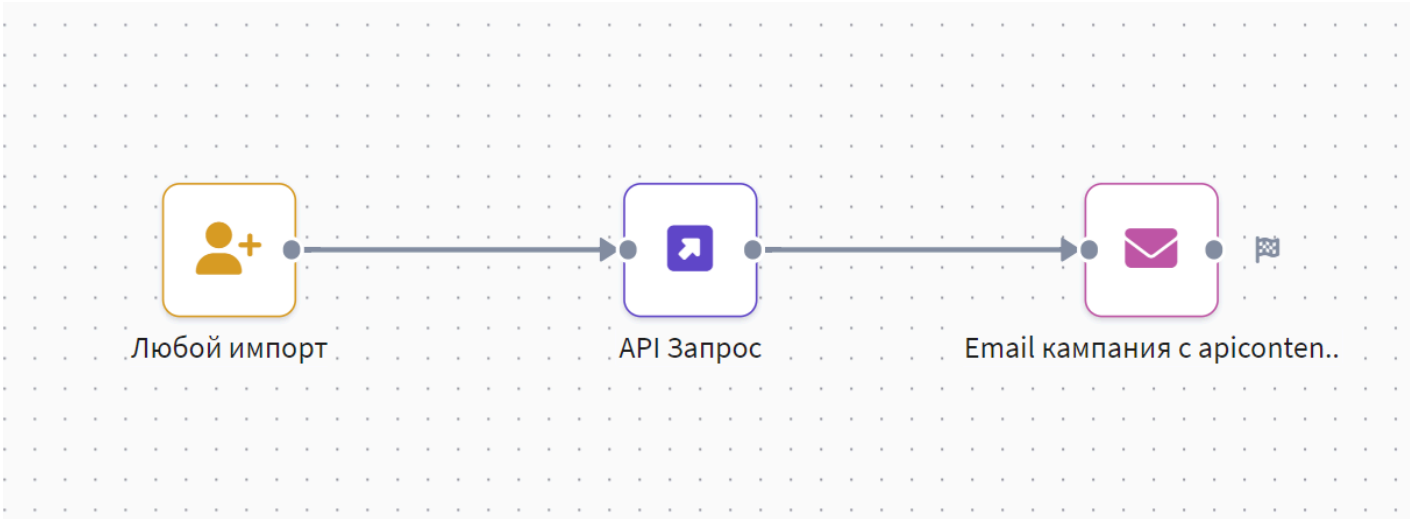
О том, как настроить элемент “API Запрос”, можно прочитать здесь.

Выберите тип запроса и вставьте ссылку на сервис, который будет возвращать JSON-объект:

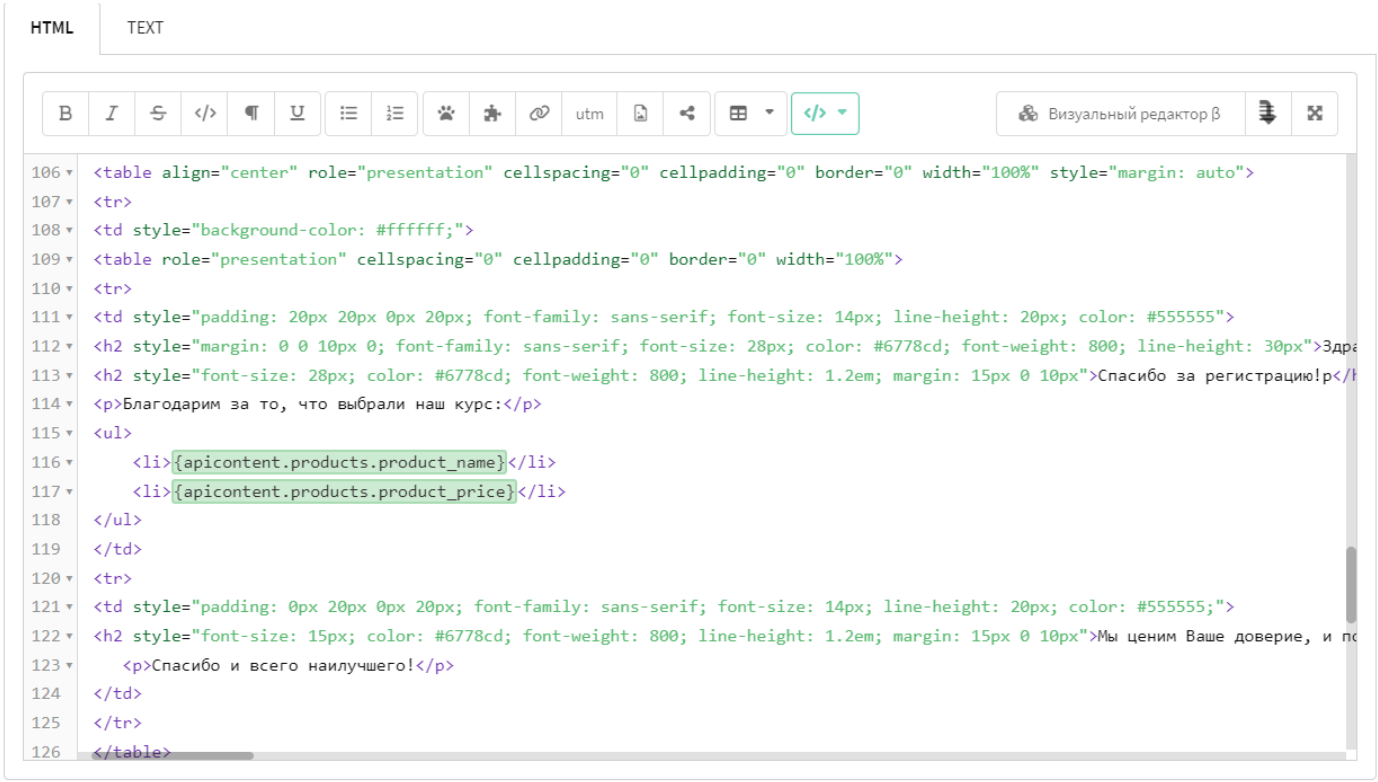
Нажмите на переключатель **“Сохранить http-ответ в поле apicontent”** и введите название поля, в которое будут записаны данные из внешнего источника:



После этого вы сможете использовать данные из `apicontent` в других элементах сценария. Например, вы можете добавить в цепочку элемент **“Email-рассылка”** и выбрать шаблон с переменной `apicontent`.

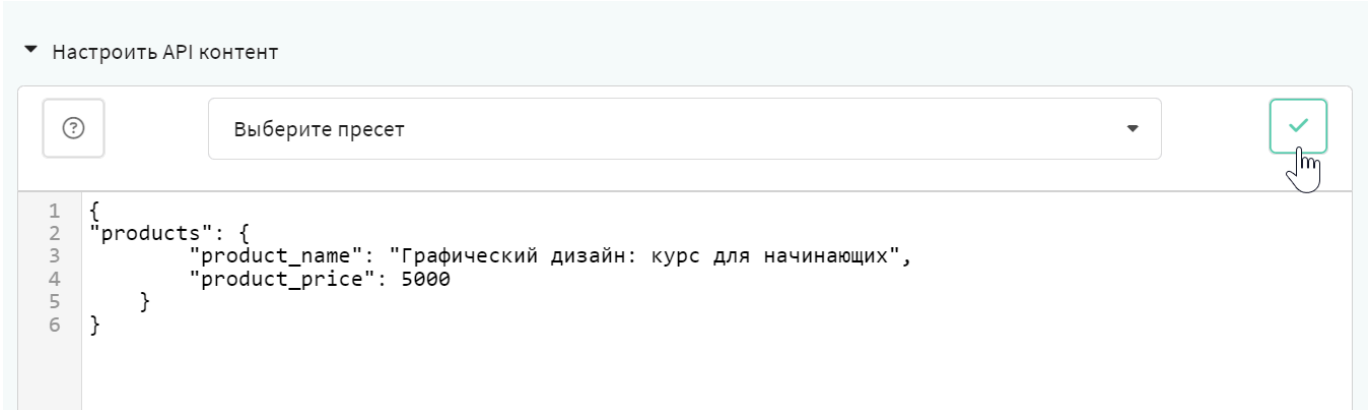


Обратите внимание, что в данном случае данные записываются в поле `products`, поэтому путь к данным в шаблоне немного изменится:



TIP

Предпросмотр сообщения с `apicontent` доступен только после настройки раздела **“Настроить API-контент”**. Для этого перейдите в раздел предпросмотра и вставьте объект, содержащий `apicontent`, в редактор. Нажмите на галочку, чтобы применить изменения.



Использование цикла в шаблоне с apicontent

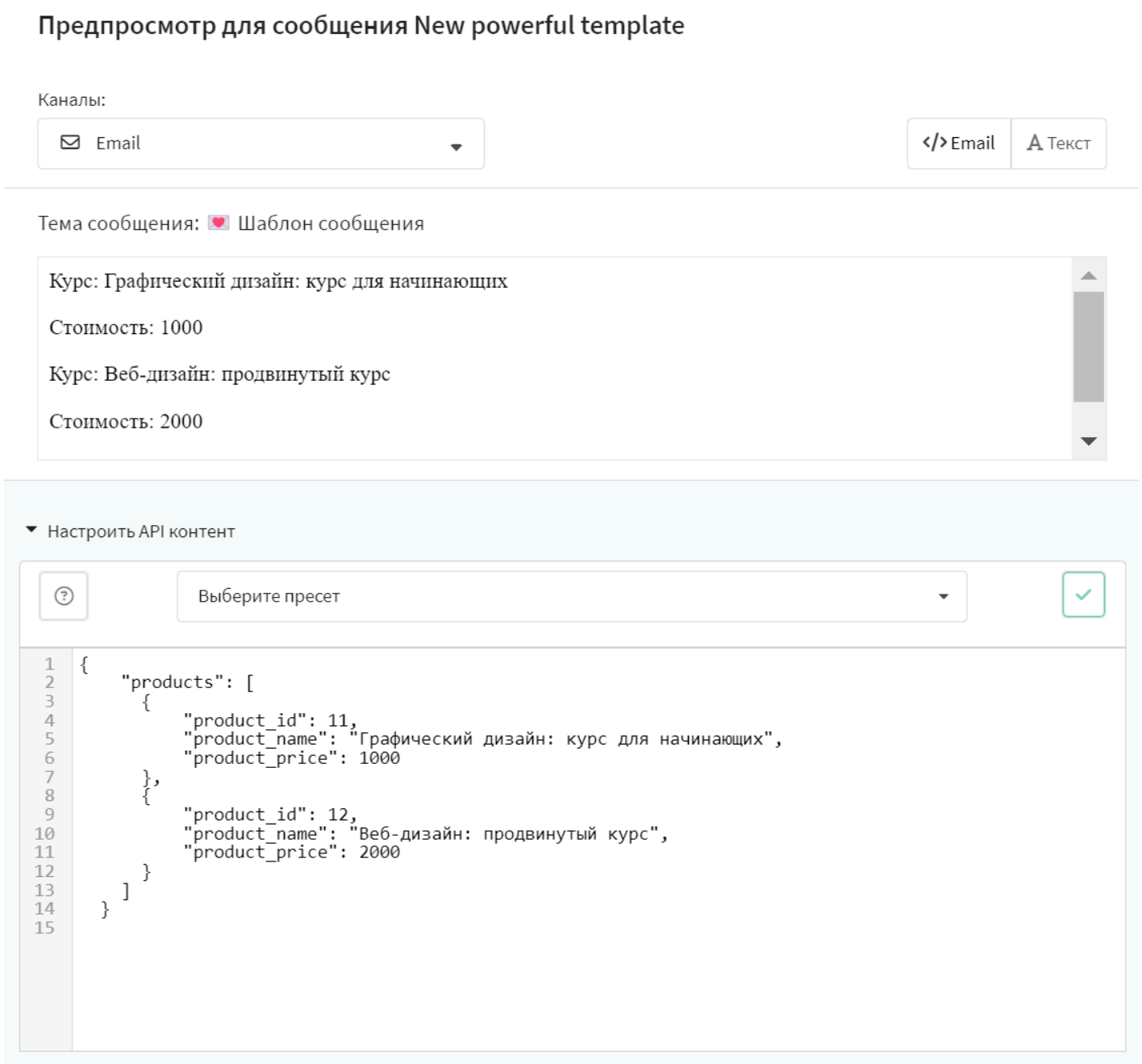
В переменную `apicontent` можно передавать JSON-объект, содержащий массив. Пример:

```
{
  "products": [
    {
        "product_id": 11,
        "product_name": "Графический дизайн: курс для начинающих",
        "product_price": 1000
    },
    {
        "product_id": 12,
        "product_name": "Веб-дизайн: продвинутый курс",
        "product_price": 2000
    }
  ]
}
```

Чтобы подставить данные из массива в шаблон, вам необходимо перебрать массив. Для этого воспользуйтесь [циклом с переменными](#).

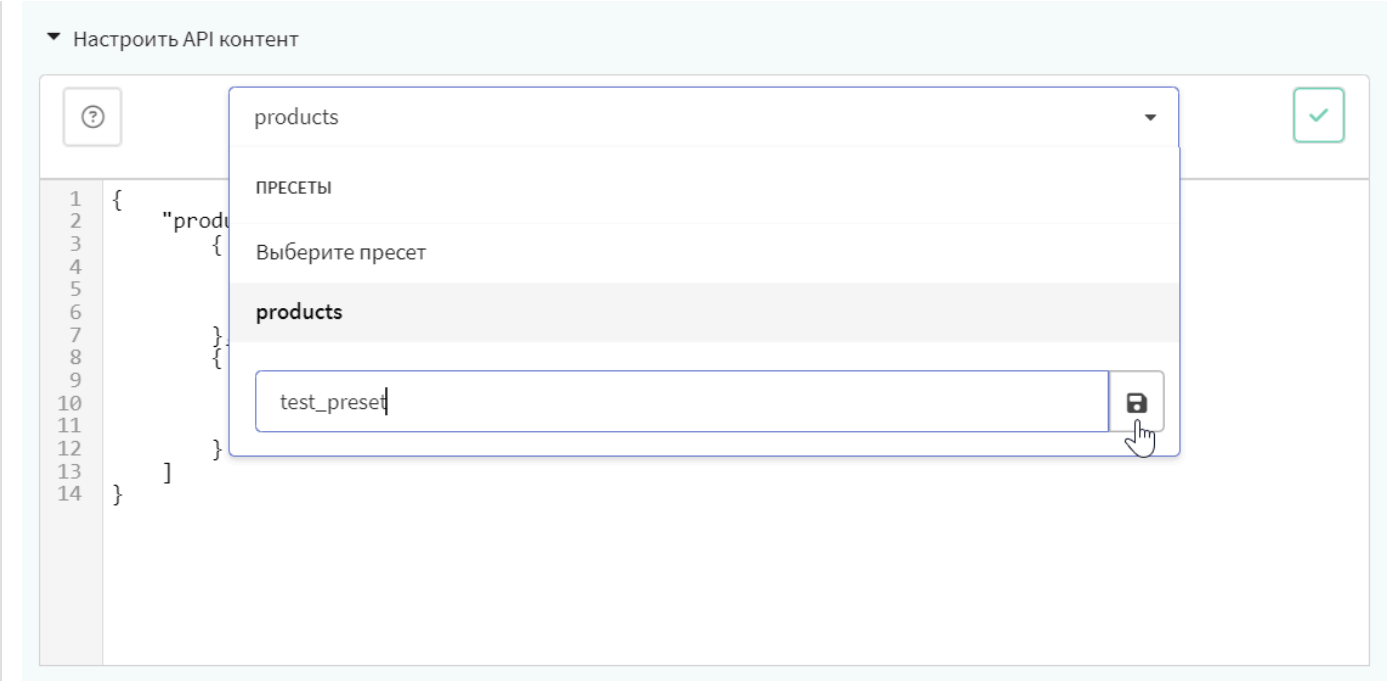
```
{for $index $el = apicontent.products}
<div>
  <p>Курс: {$el.product_name}</p>
  <p>Стоимость: {$el.product_price}</p>
</div>
{else}
  if empty array
{end}
```

Этот код подставит в письмо следующие данные:



TIP

Если вы регулярно используете один и тот же JSON-объект в переменной, то вы можете **сохранить его как пресет** и в дальнешем выбирать этот пресет в окне предпросмотра:



Отмена отправки сообщения, если нет поля

Вы можете настроить отмену отправки сообщения, если в объекте `apicontent` определенное поле отсутствует или содержит пустое значения. Для этого добавьте проверку на наличие поля и используйте переменную `{cancel}`:

```
{if apicontent.field_2}
field_2 не пустое
{else}
{cancel}
{end}
```

При таком условии рассылка не отправит письмо, если `apicontent` будет содержать один из следующих объектов:

```
{
  "field_1": "example"
}
```

```
{
  "field_1": "example",
  "field_2": "" / или 0, null, false
}
```

Этот подход позволяет избежать отправки нерелевантных писем, а также блокировать отправку, если из внешнего источника были получены неполные данные.

Пример использования

Допустим, у вас есть маркетинговая рассылка, предназначенная только для пользователей, которые прошли или начали проходить определенный курс. В запросе приходит объект `apicontent`, который включает информацию о статусе прохождения курса:

```
{
  "user_id": 123,
  "course_id": "abcadefg",
  "course_name": "Графический дизайн: курс для начинающих",
  "course_status": "completed"
}
```

Однако если в запросе поле `course_status` отсутствует или его значение пустое, это может означать, что пользователь не начал курс или не завершил его. В таком случае письмо не должно быть отправлено. Шаблон письма:

```
{if apicontent.course_status}
<div>
  <span>Вы интересовались курсом {apicontent.course_name}.</span>
  <span>Возможно, вам понравятся эти курсы: <список курсов></span>
</div>
{else}
{cancel}
{end}
```

Использование HTML-контента в сообщениях

Динамический HTML-контент — это один из способов создавать динамические сообщения и автоматизировать работу. С его помощью в сообщение добавляется информация, генерируемая на вашем сервере точно в момент отправки каждого сообщения. Это удобно, когда информация в письмах должна меняться в зависимости от каких-то условий и параметров.

Чтобы добавить ссылку на динамический HTML-контент нажмите кнопку “Добавить внешний контент” в шапке редактора сообщения.

Например, если сервер умеет отдавать тексты последних новостей на сайте, то сообщение, содержащее ссылки на него, будет автоматически поддерживаться в актуальном состоянии.

Можно использовать несколько ссылок на HTML-контент в сообщении.

```
{[http://www.example.news?index=1&news_group=movies]}

{[http://www.example.news?index=2&news_group=movies]}
```

В ссылку на динамический HTML-контент можно включать различные переменные. Например, имя подписчика `{lead._fname}` или название базы данных `{listname}`, в которой его профиль находится. То есть вместо статического `&news_group=movies` можно написать `&news_group={lead.NewsGroup}`.

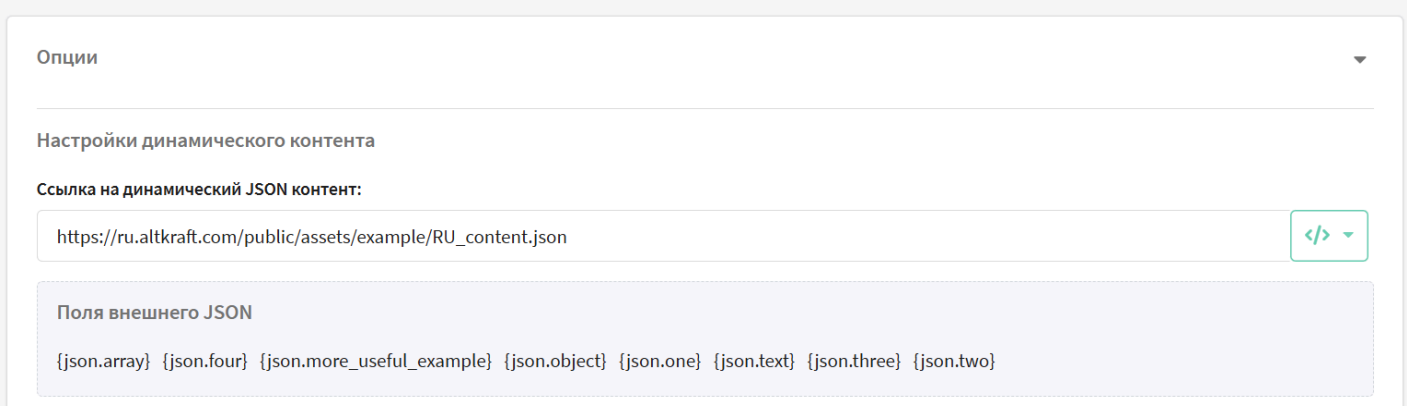
При каждой отправке такого сообщения, с сервера будет запрашиваться динамический контент и подстанавливаться в сообщение. Все подписчики получат свежие новости в зависимости от своих предпочтений.

Использование JSON-контента в сообщениях

Динамический JSON-контент — это один из способов создавать динамические сообщения и автоматизировать работу. С его помощью в сообщение добавляется информация, генерируемая на вашем сервере точно в момент отправки каждого сообщения.

В ссылку на динамический JSON-контент можно включать различные переменные. Например, данные профиля `{lead._fname}`.

Источник динамического JSON-контента устанавливается на вкладке **“Опции”**:



CAUTION

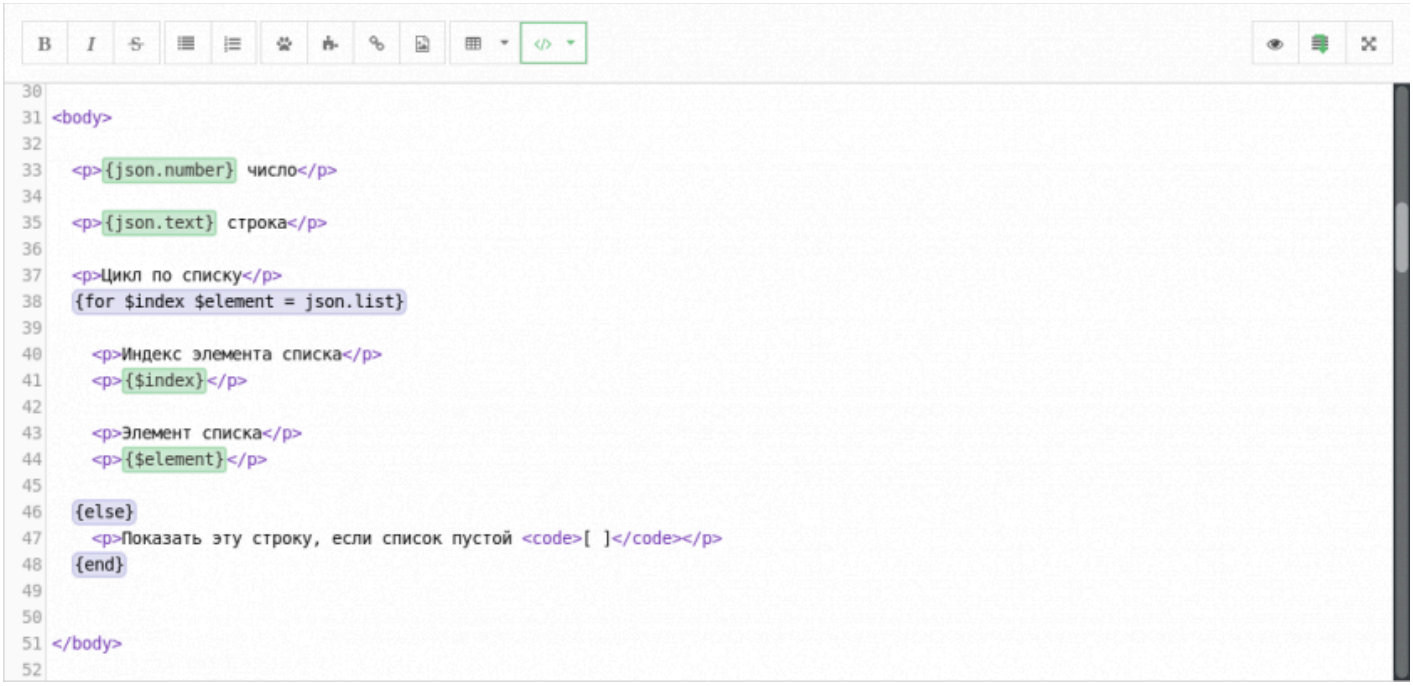
По умолчанию размер динанамического JSON **ограничен 1 МБ**. В **On-premises версии** этот лимит можно изменить в файле конфигурации через параметр `MAX_REMOTE_CONTENT_BYTES`.

Информация, представленная в JSON формате, выглядит так:

```
{
  "number": 10,
  "text": "Hello!",
  "list": ["One", "Two", "Three"],
  "object": { "model": "x17", "price": "$99" },
  "mixed": [
    { "city": "Los Gamburger", "street": "Florence Avenue" },
    { "city": "St. Potato", "street": "Glendale Boulevard" }
  ]
}
```

Чтобы вставить JSON-контент в письмо, нажмите кнопку **“Вставить переменную”** `</>` и в меню выберите **“Внешний JSON”**.

В тексте сообщения JSON-контент представлен переменными типа `{json.variable}`



```
<body>

  <p>{json.number} число</p>

  <p>{json.text} строка</p>

  <p>Цикл по списку</p>
  {for $index $element = json.list}

    <p>Индекс элемента списка</p>
    <p>{$index}</p>

    <p>Элемент списка</p>
    <p>{$element}</p>

  {else}
    <p>Показать эту строку, если список пустой <code>[ ]</code></p>
  {end}


</body>

```

Два типа данных “list” и “object” поддерживают циклы.

Для типа “object” также доступно обращение к элементам по ключу `{json.object.key_name}`

Использование контента из SQL базы данных в сообщениях

Описание

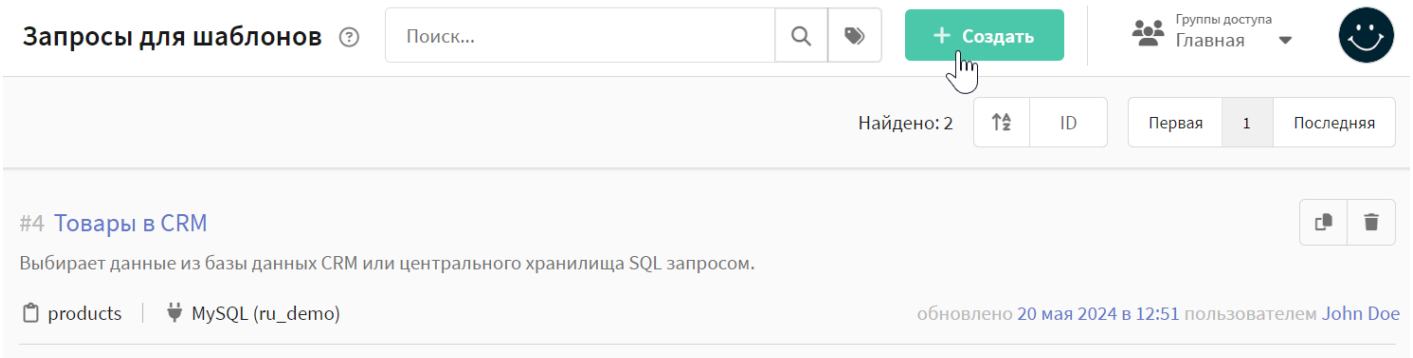
В шаблонах сообщений можно использовать динамический контент из внешних таблиц данных, хранящихся на **MySQL**, **PostgreSQL**, **MSSQL**, **Oracle**, **Hive** и **ClickHouse** серверах. Для этого администратору платформы необходимо в Административной панели Altcraft **создать коннектор к внешней базе данных**.

TIP

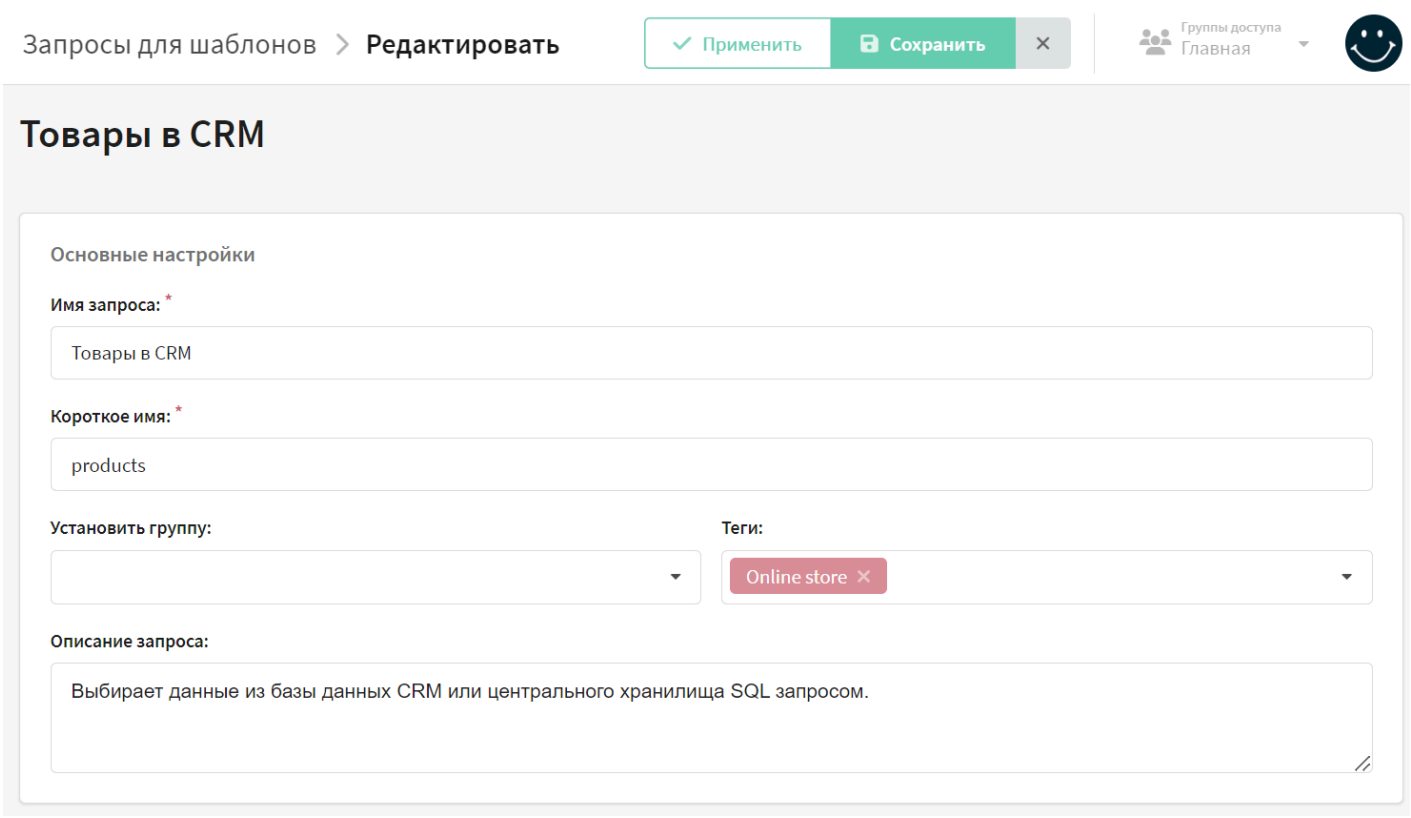
Если вы пользуетесь **облачным решением** — передайте данные для подключения к базе данных нашей службе поддержки: team@altcraft.com. Там же можно уточнить по вопросам, связанным с возможностью подключения к другим внешним службам и БД.

Создание запроса к внешней таблице данных

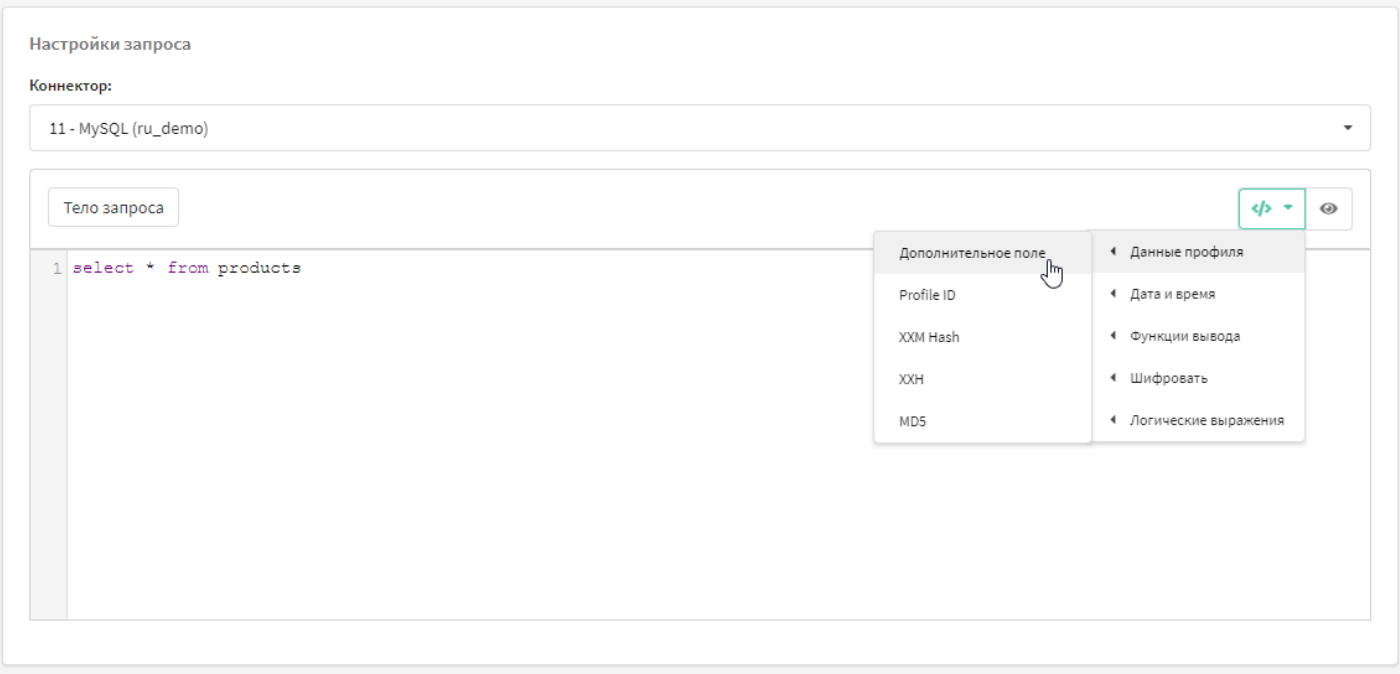
Для создания запроса перейдите на вкладку **Данные —> Запросы для шаблонов** и создайте запрос.



В основных настройках укажите **название** запроса и **короткое имя** для использования в редакторе шаблонов. При необходимости можно добавить **описание**, установить **группы** для ограничения доступа других пользователей к запросу и прикрепить **теги** для быстрого поиска.



После этого выберите в выпадающем меню подходящий **коннектор** к базе данных и введите тело запроса.

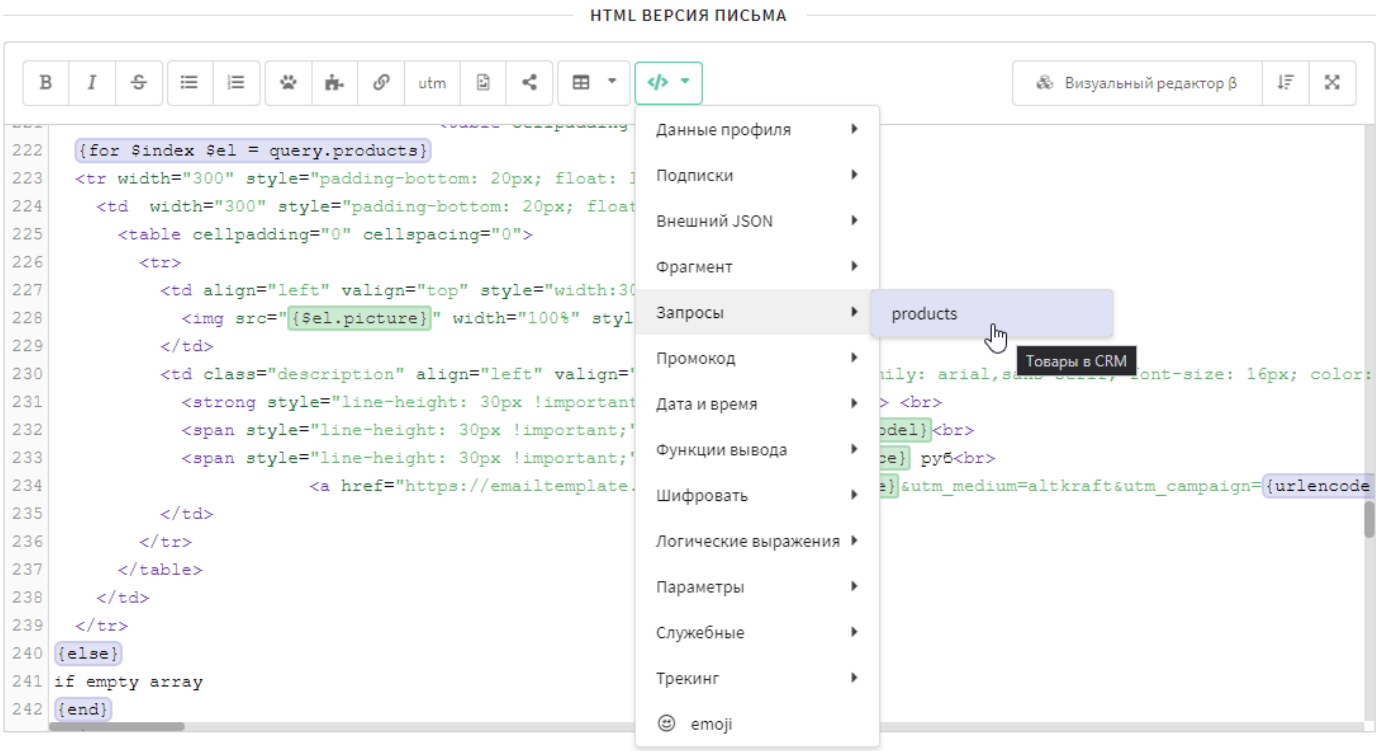


TIP

Второй вариант создания запроса — в меню **“Данные”** открыть пункт **“Таблицы данных”**. Выберите пункт **“Сохранить как запрос для шаблонов”**.

Использование SQL запроса в шаблонах сообщений

В шаблоне сообщения используйте ключевой параметр `query.your_query_name`.



Подробнее о функциональности редактора сообщений – Переменные и функции ALTCRAFT.

Готовый результат вы можете отформатировать по желанию:

```
{for $index $el = query.products}
<tr width="300" style="padding-bottom: 20px; float: left">
  <td width="300" style="padding-bottom: 20px; float: left">
    <table cellpadding="0" cellspacing="0" class="doc-table">
      <tr>
        <td align="left" valign="top" style="width:30%;">
          <center>
          </center>
        </td>
        <td
          class="description"
          align="left"
          valign="middle"
          style="font-family: arial,sans-serif; font-size: 16px; color: #333; padding: 0
16px;"
        >
          <strong style="line-height: 30px !important;">{$el.type} </strong> <br />
          <span style="line-height: 30px !important;">Модель: </span>{$el.model}<br />
          <span style="line-height: 30px !important;">Цена: </span>{$el.price} руб<br />
          <a
            href="https://emailtemplate.ga?utm_source=
{msg_type}&utm_medium=altcraft&utm_campaign={urlencode(campid)}&utm_term=
{urlencode(msgname)}&utm_content={urlencode(msgid)}"
            style="display: block; text-decoration: none; font-weight: 600; margin-top:
15px; text-shadow: 0 2px 15px #000000; color: #313131"
            name="link1"
            >В корзину!</a
          >
        </td>
      </tr>
    </table>
  </td>
</tr>
{else} if empty array {end}
```

Тема сообщения: Новинки в интернет-магазине

Самые горячие новинки сезона:

Платье

Модель: Rosso-Orient

Цена: 3600 руб

В корзину!

Футболка

Модель: Retoro-SE

Цена: 2000 руб

В корзину!

Футболка

Модель: Skel-Hand

Цена: 1500 руб

В корзину!

Свитшот

Модель: Tomato Soup

Цена: 2800 руб

В корзину!

Рассылки

Рассылки в Altcraft управляют отправкой сообщений клиентам.

Рассылка генерирует сообщение подписчикам определённого ресурса на основе выбранных шаблонов сообщений, добавляя информацию о клиенте из базы данных, динамический контент, части и фрагменты.

Если в качестве аудитории рассылки используется Динамический сегмент, он будет рассчитан непосредственно перед каждым запуском. Также перед запуском рассылка **исключает** из рассылки профили, находящиеся в стоп-списках платформы и заблокированные политикой отправки.

Типы рассылок

 Броадкаст	**Броадкаст** Осуществляет однократную рассылку сообщений определённой аудитории подписчиков. При больших объёмах отправку можно разделить на несколько частей по расписанию. В качестве аудитории может выступать база данных подписчиков или сегмент.
 Регулярная	**Регулярная** Рассылает доступный контент определённой аудитории подписчиков на регулярной основе. Регулярная рассылка никогда не останавливается автоматически. В качестве аудитории может выступать база данных подписчиков или сегмент.
 Сплит-тест	**Мультивариантный тест** Рассылка, которая позволяет Вам протестировать маркетинговую эффективность отправляемых сообщений. В рассылку добавляются до 16 шаблонов сообщений. В качестве аудитории может выступать база данных подписчиков или сегмент.
 Триггер	**Триггер** Рассылка, которая отправляет сообщения автоматически подписчикам при совершении подписчиком определённого действия. Также условиями запуска могут быть изменение информации о подписчике, импорт профиля и прямой API запрос.

Календарь рассылок

	Календарь рассылок Altcraft отображает время запуска маркетинговых рассылок: • **Броадкастов** — массовых единоразовых рассылок. • **Регулярных** — периодических рассылок и уведомлений. • **Мультивариантный тест** (A/B/n) — массовых рассылок с возможностью выбрать наиболее эффективный **контент**.

Из интерфейса календаря также можно создать новую рассылку любого типа, в том числе **триггерную**, не имеющую фиксированного времени запуска.

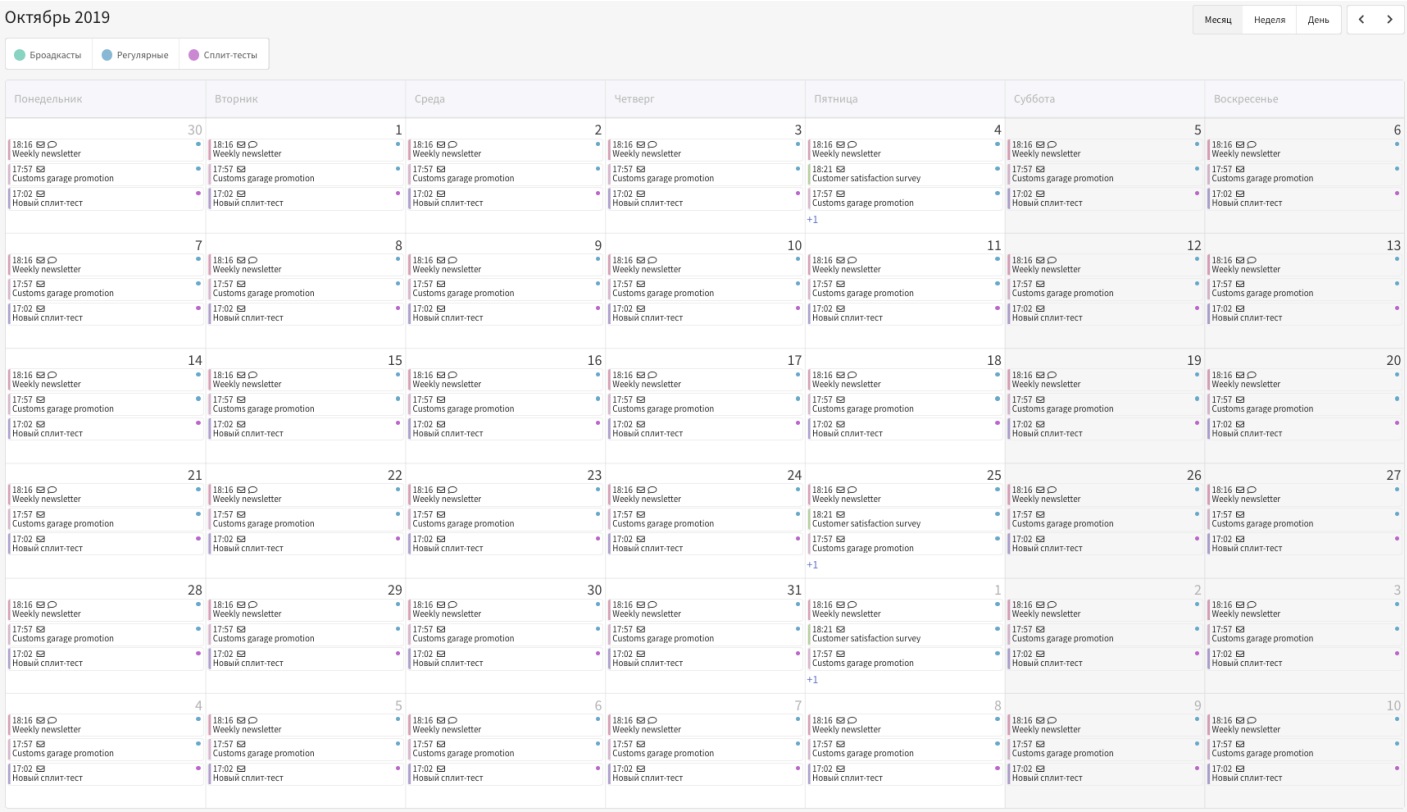
В главном меню откройте **Рассылка** → **Календарь**.

Просмотр информации о запусках

В календаре отображается информация как о **предстоящих** запусках рассылок по расписанию, так и о **прошедших**. Если вы проводите массовую рассылку, разделённую на несколько частей, в календаре будет отображаться запуск каждой порции сообщений. Это поможет контролировать нагрузку на модуль отправки, сетевую инфраструктуру вашей организации, а также ограничения провайдеров email.

Для каждого запуска рассылки отображается время, тип и название рассылки, используемые каналы связи. Доступно три варианта отображения:

- **Месяц** — выводит информацию по всем запускам за месяц. По каждому дню месяца отображается три первых рассылки. Если их больше, внизу плитки появится ссылка на подробную информацию о запусках за день.
- **Неделя** — выводит информацию о запусках за неделю. Рассылки сгруппированы по дню и времени запуска.
- **День** — отображает все запуски рассылок в отдельно взятый день.

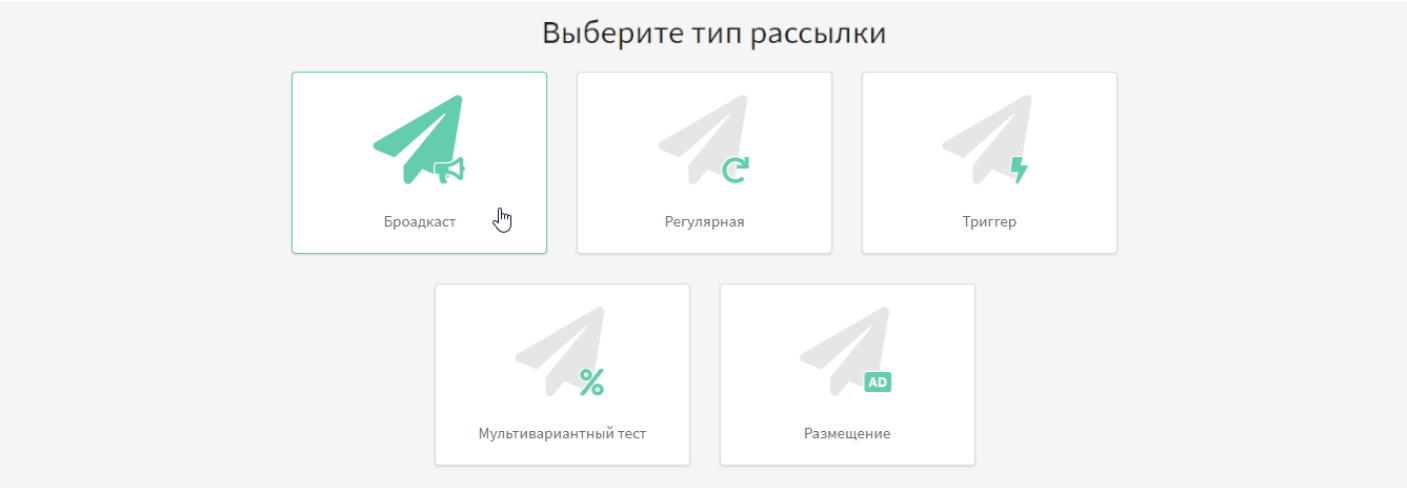


	При нажатии на плитку запуска откроется подробная информация: • Предпросмотр отправляемого контента (email) • Название рассылки • Шаблон сообщения • Точное время запуска • Канал коммуникации • Автор рассылки • Дата создания / изменения • Ресурс Ниже находятся кнопки для **клонирования** или **редактирования** расылки.

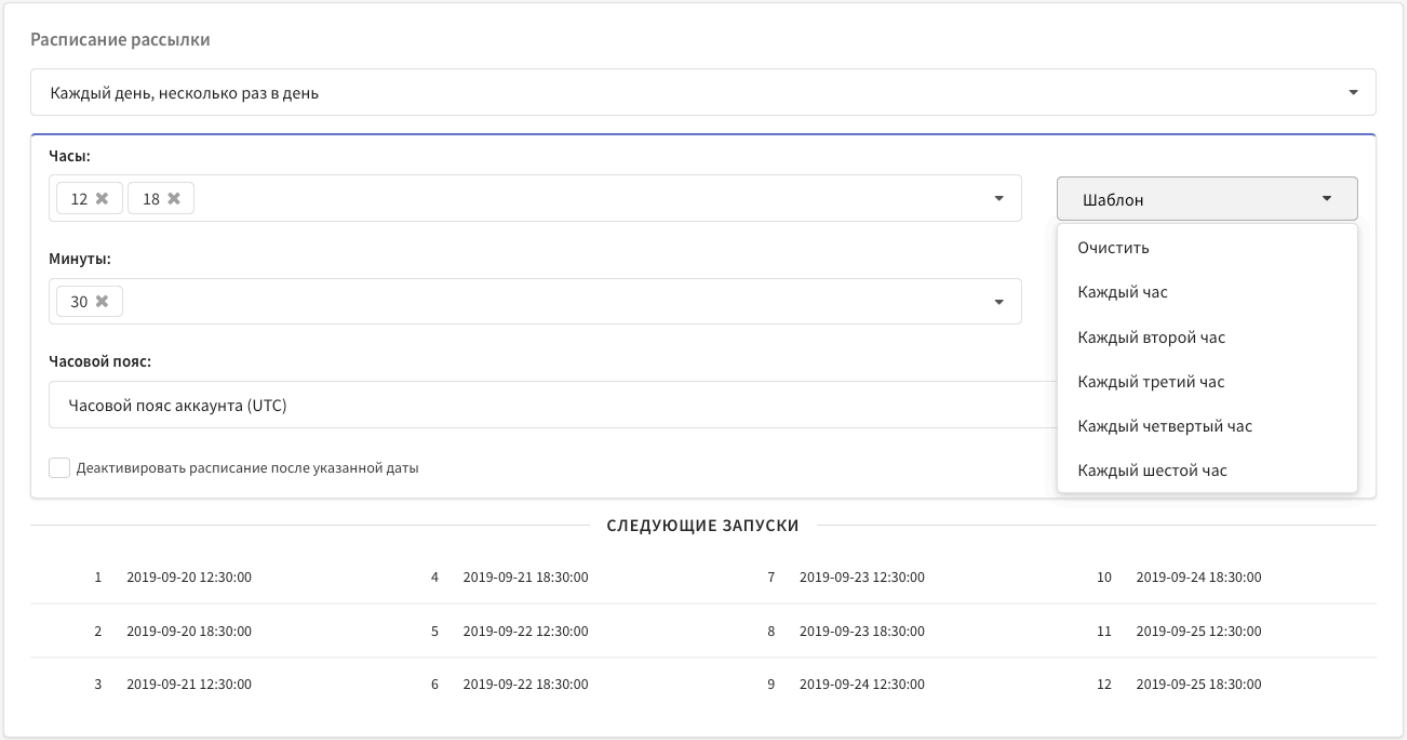
Создание новой рассылки

	Из интерфейса календаря можно создать новую рассылку любого типа. Выберите день первого запуска, кликните мышкой по пустому полю и во всплывающем окне нажмите **+ Добавить рассылку**.

Далее выберите тип рассылки – откроется форма редактирования.



Дата запуска будет автоматически установлена на выбранный вами в календаре день, однако её можно изменить в любой момент, а также установить необходимое количество запусков и расписание:



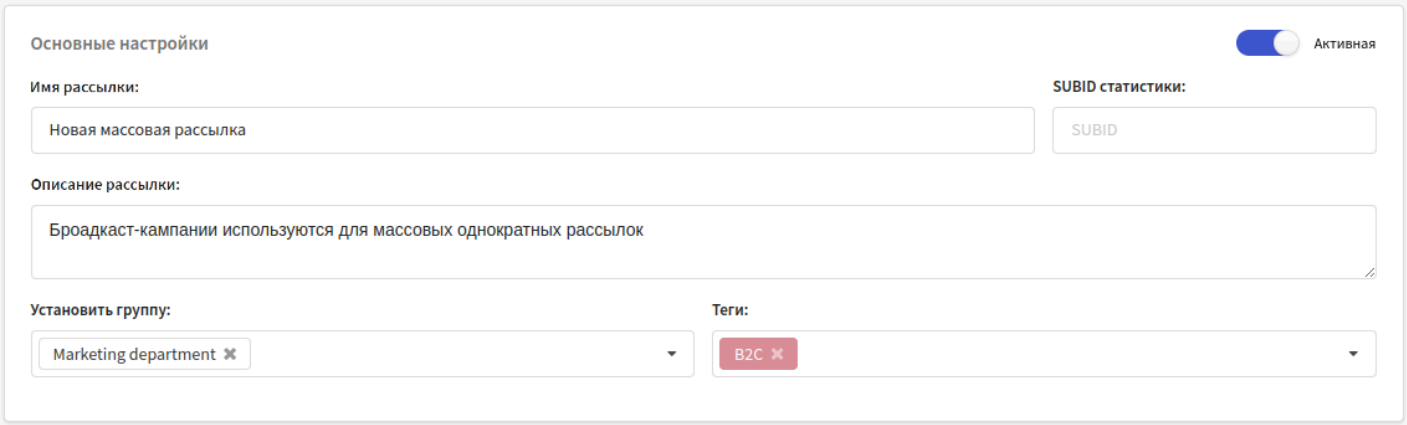
Броадкаст рассылка

Броадкаст рассылка — это инструмент для организации **массовых однократных** рассылок контента подписчикам. Рассылает по одному сообщению всем подписчикам из своего источника данных по заданному расписанию. Когда каждому из получателей будет отправлено одно сообщение, броадкаст-рассылка завершится, в отличие от регулярной рассылки, которая снова начнёт рассылку с начала списка.

	Чтобы создать новый броадкаст, в главном меню выберите **Расслыки** → **Броадкасты** и нажмите кнопку **+ Создать**. Или воспользуйтесь календарём рассылок.

Основные настройки

В основных настройках укажите название и описание новой рассылки, при необходимости установите группу доступа, **теги** и **атрибуты** для дополнительной группировки. Здесь же вы можете указать **SUBID статистики** для рассылки. В верхнем правом углу находится переключатель, активирующий расылку:



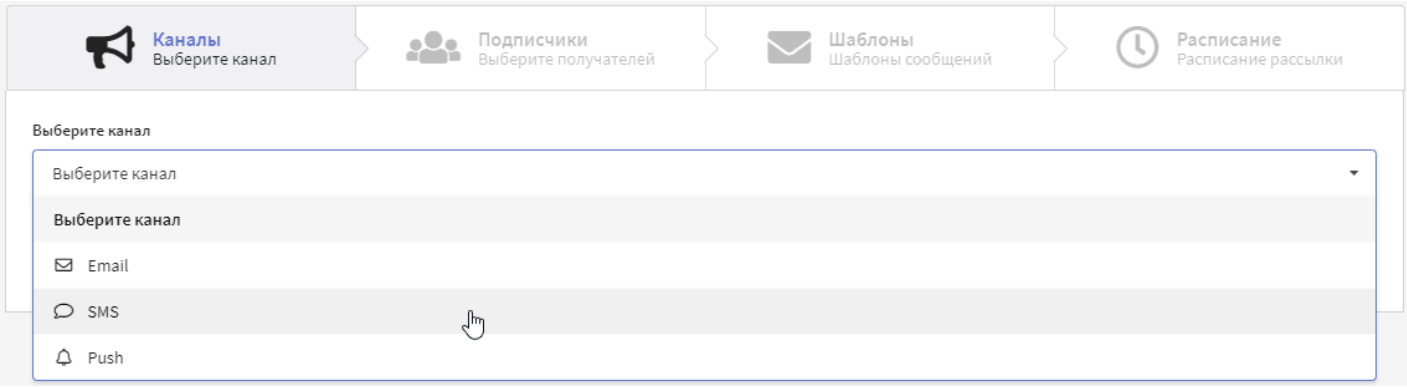
Настройки рассылки

Настройки рассылки объединяют канал, по которому она будет отправлена, получателей сообщений, контент из одного или нескольких шаблонов, а также расписание рассылки.

Каналы

В **первой вкладке** выберите канал связи, по которому хотите отправить рассылку. Канал отправки должен быть доступен хотя бы в одном ресурсе подписок. Ресурс выбирается в следующей вкладке.

> **NOTE**
>
> Для отправки по Email доступно использование основных контактных данных подписчика — поля Email в базе данных.



TIP

При использовании **push-канала** вы можете выбрать конкретных провайдеров для рассылки, а также режим отправки — **fanout**, рассылка на все подписки с учетом фильтров, или **last**, отправка на подписку с самой свежей датой создания. Если несколько подписок были созданы одновременно, то из них будет выбрана одна случайная. Если в рассылке заданы фильтры по провайдеру, дополнительным полям или категориям, то сначала накладываются фильтры, а потом уже ищется самая свежая подписка.

Подписчики

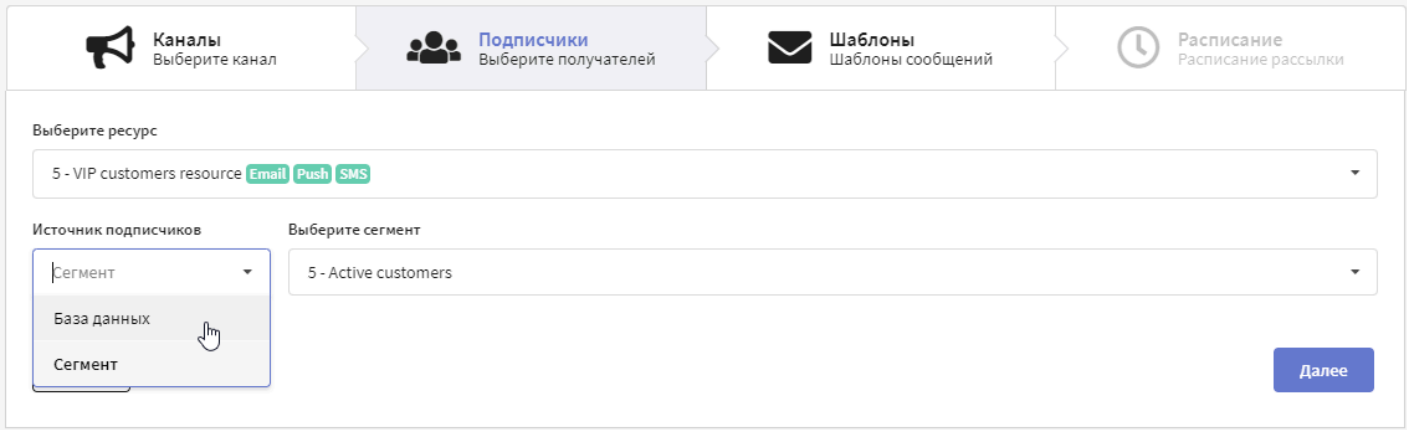
Рассылки в Альткрафт отправляются в рамках определённого ресурса: это может быть отдельный сайт, продукт, акция, или другое направление ваших коммуникаций с клиентами.

Выберите ресурс из выпадающего меню. В ресурсе должен быть настроен выбранный на предыдущем шаге **канал**.

Получателями броадкаст рассылки может быть база профилей или сегмент, состоящий из профилей одной или нескольких баз данных. Установите необходимые значения в полях **Источник данных** и **Название** (сегмента или базы данных).

TIP

При отправке сообщений рассылка использует **подписки клиентов** на выбранный ресурс. Убедитесь, что получатели в вашем источнике данных подписаны на него!

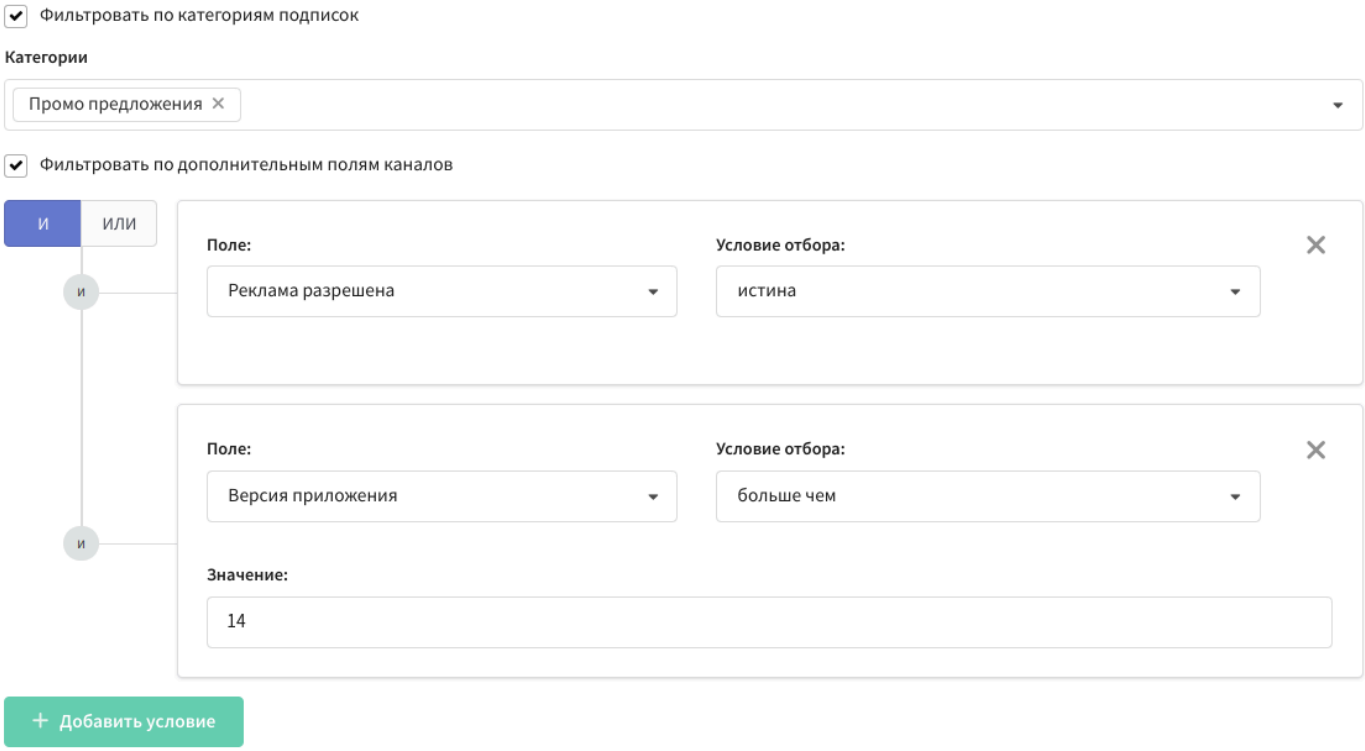


Дополнительно вы можете **отфильтровать получателей по категориям ресурса**. Для этого в ресурсе должны быть созданы категории. В настройках рассылки выберите категории, на которые должен быть

подписан профиль, чтобы получить сообщение.

Также можно отфильтровать получателей рассылки по **дополнительным полям канала** в ресурсе. Чтобы создать фильтр по полям, выберите стандартное или дополнительное поле подписки. В поле **“Условие отбора”** выберите логический оператор и при необходимости укажите значение.

В фильтре по полям возможно добавление нескольких правил и их объединение с помощью логического **И** (все условия должны выполняться) и **ИЛИ** (хотя бы одно условие должно выполняться).



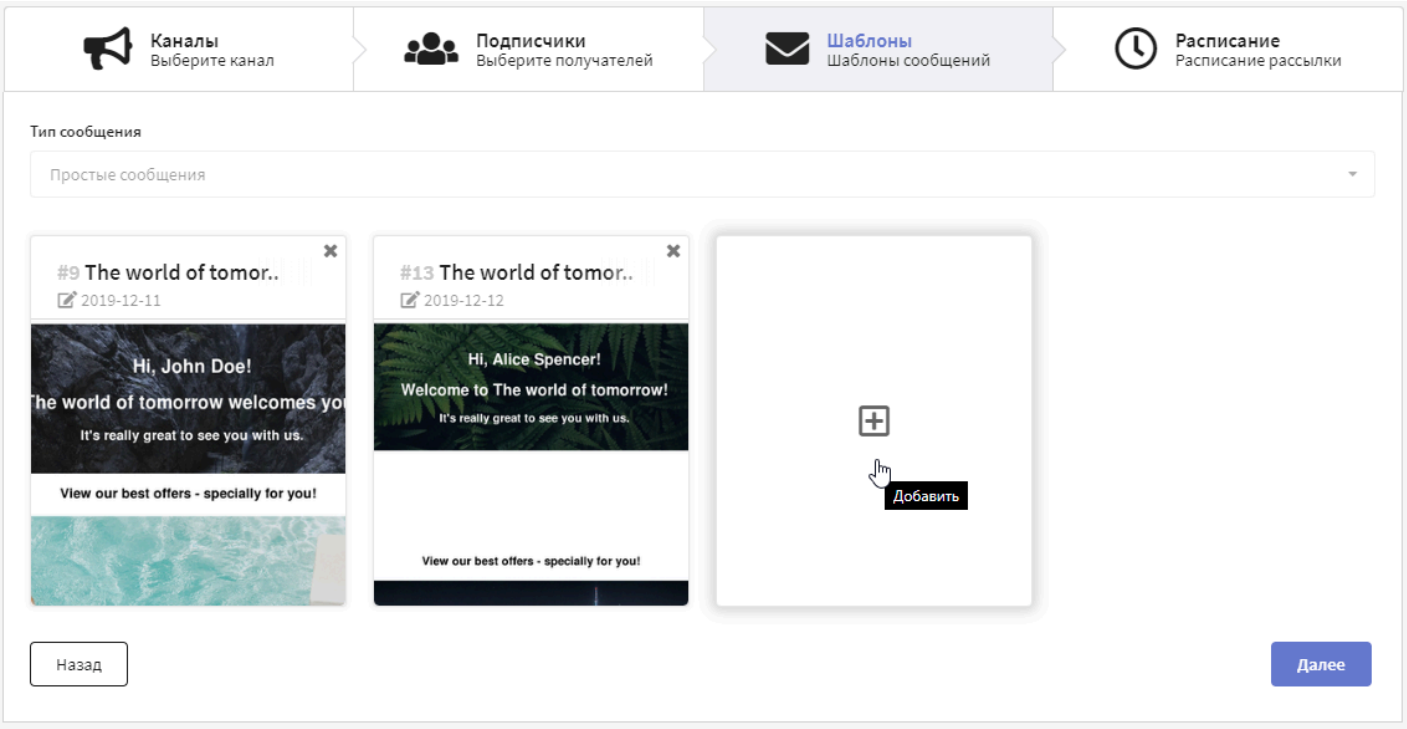
Шаблоны

В этом блоке выберите контент, который хотите использовать в рассылке. Шаблоны сообщений в Альткрафт объединяют контент для отправки по всем доступным каналам.

> **TIP**
>
> Убедитесь, что контент для выбранного в рассылке канала присутствует в шаблоне!

Нажмите на кнопку **Добавить** и выберите подготовленный шаблон. Любой тип рассылки поддерживает до 8 шаблонов сообщений. Броадкаст-рассылка отправит каждому получателю по **одному** сообщению из выбранных в случайном порядке – в равных пропорциях.

По клику на миниатюру шаблона откроется **предпросмотр**. При нажатии на название шаблона он откроется в новой вкладке для редактирования:



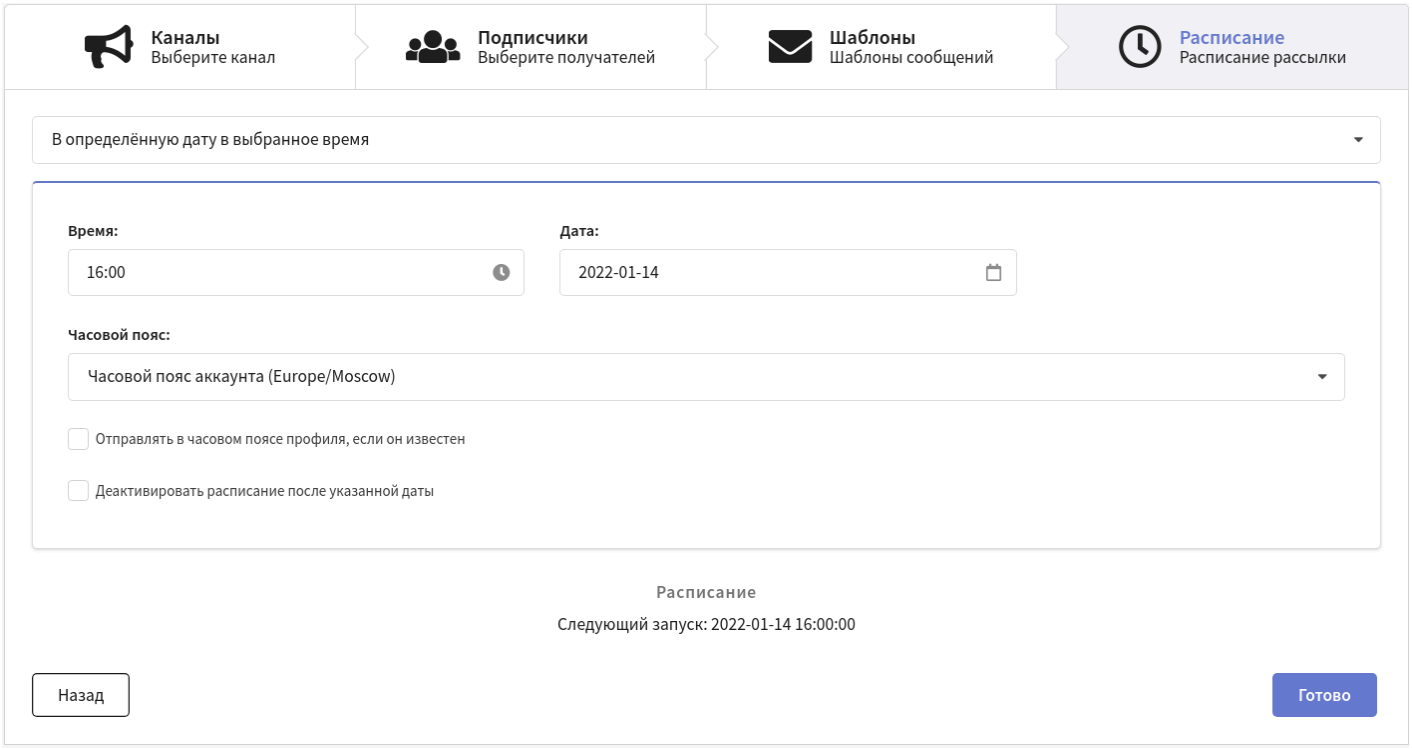
Расписание

> **TIP**
>
> Расписание рассылок подробно описано в отдельной статье.

При проведении массовых рассылок необходимо иметь во внимании такой показатель, как репутация отправителя (домена и IP адресов, с которых ведётся рассылка). Провайдеры электронной почты устанавливают ограничения на массовые рассылки - в зависимости от репутации и количества отправляемых сообщений. Обычно провайдеры устанавливают максимум принимаемых сообщений на 24 часа.

Отправка всех сообщений за один запуск

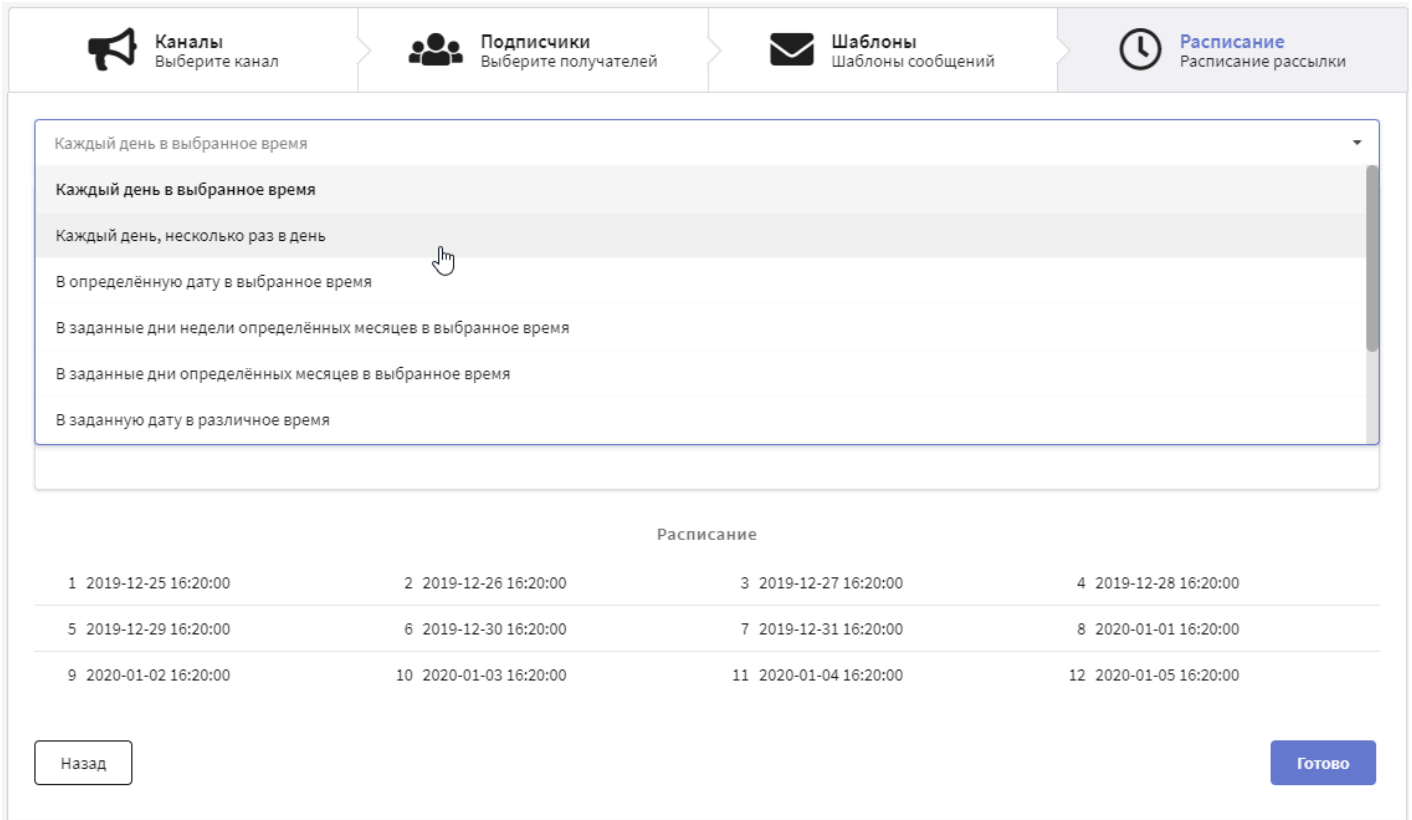
Если репутация отправителя достаточно высока, чтобы отправить все сообщения рассылки за 24 часа, в блоке **Расписание** установите один запуск, **в определённую дату в выбранное время**:



Дополнительно вы можете установить **Часовой пояс** для отправки или отправку в часовом поясе клиента, если такие данные доступны.

Порционная отправка

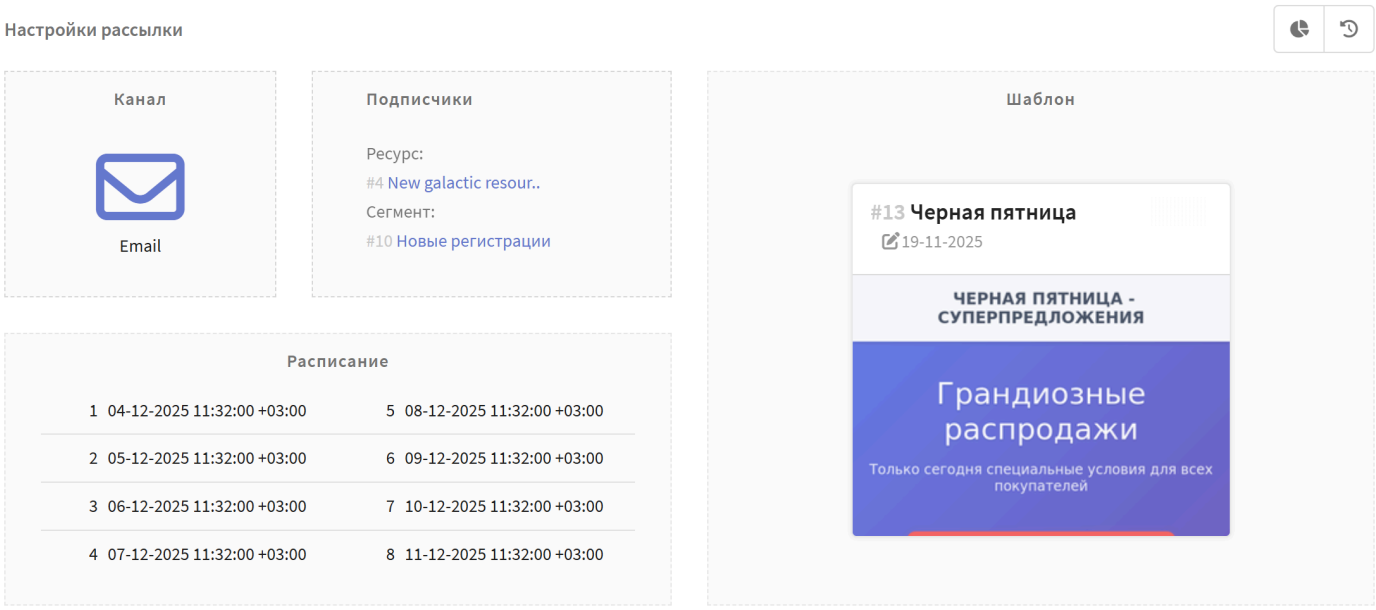
Так как большинство ограничений провайдеров email действует 24 часа, в расписании броадкаста в случаях массовых рассылок, превышающих ограничение, подойдёт план **Каждый день в выбранное время**. Количество сообщений нвстраивается в блоке **Ограничения рассылки** после сохранения изменений.



CAUTION

При порционной отправке с фиксированным количеством запусков убедитесь, что их количество, умноженное на лимит для каждого запуска больше, чем количество получателей. В противном случае, часть сообщений не будет отправлена.

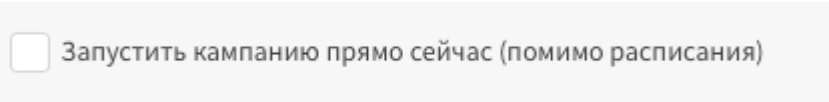
Когда вы завершите настройку, нажмите **Готово**. Выбранные вами параметры будут отображаться в блоке **Настройки рассылки**. Кнопками в правом верхнем углу блока вы можете **редактировать** настройки или перейти к отчету по рассылкам.



Мгновенный запуск рассылки (помимо расписания)

Чтобы запустить рассылку сразу же после создания, в самом низу страницы отметьте соответствующую опцию.

- Если лимиты не установлены – сообщения отправятся всем получателям.
- Если рассылка разбита на несколько частей – произойдёт запуск помимо расписания.



Ограничения рассылки

Если количество сообщений рассылок больше, чем провайдеры электронной почты принимают с ваших почтовых доменов в сутки, броадкаст рассылку можно разделить на несколько частей.

NOTE

Другой сценарий использования — распределение нагрузки на сендер (модуль отправки), или на инфраструктуру вашей организации. Например, при ограничениях канала отправки сообщений.

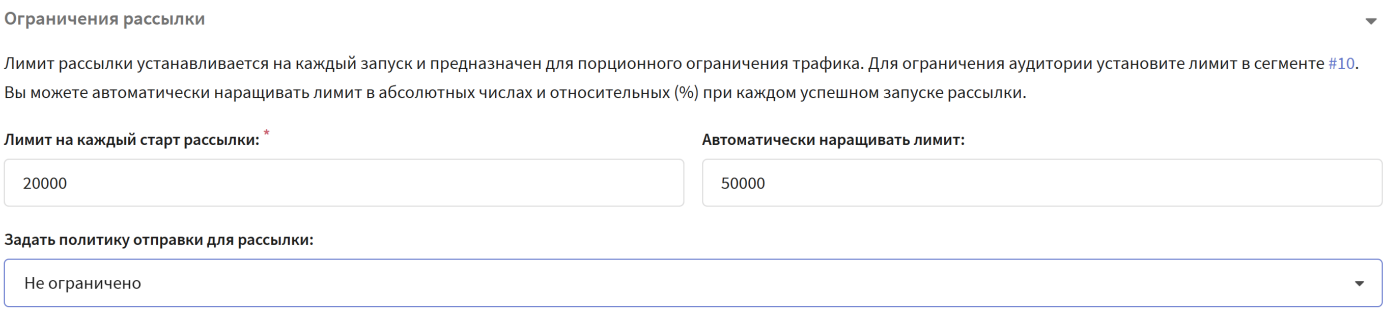
В этих случаях установите в рассылке ограничения рассылки:

- **Лимит на каждый старт рассылки** – количество сообщений, которое будет отправлено в следующий запуск рассылки по расписанию.
- **Автоматически наращивать лимит** – количество сообщений или процент лимита на каждый старт, которое будет добавляться при каждом успешном запуске рассылки.

CAUTION

При активированной опции “Отправлять в часовом поясе профиля, если он известен”, использование лимитов и мгновенный запуск рассылки становятся недоступны.

Политика отправки ограничивает коммуникацию с одним подписчиком. По умолчанию рассылка будет использовать политику, выбранную глобально в аккаунте или в ресурсе, от имени которого отправляется броадкаст.



Smart settings

Блок **Smart-настройки** объединяет следующие дополнительные настройки рассылки:

- **Виртуальный сендер** — позволяет выбрать для рассылки виртуальный сендер (набор правил для отправки сообщений с реальных сендеров платформы).
- **Свои трекинг-домены** — позволяет отслеживать действия с сообщениями рассылки, используя дополнительные трекинг-серверы. Каждый домен указывается на отдельной строке.
- **Список email адресов для скрытой копии** — BCC для отправляемых сообщений. Каждый адрес указывается на отдельной строке.
- **Не проверять контактные данные на уникальность** — отключает проверку на уникальность получателей рассылки по данным подписки.

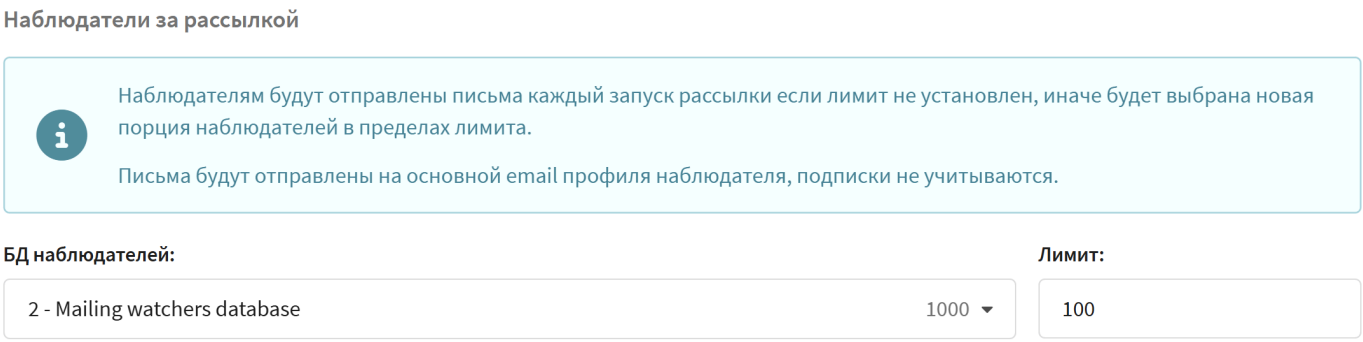
CAUTION

Если вы не уверены, какие настройки необходимо выбрать, проконсультируйтесь с администратором платформы в вашей организации или со специалистами поддержки Altcraft.

Наблюдатели за рассылкой

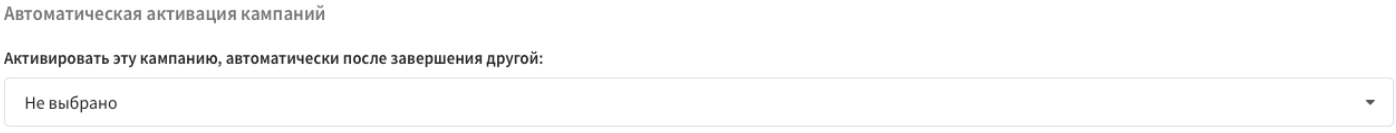
Помимо этого, в умных настройках можно установить **наблюдателей за рассылкой**. Наблюдатели будут получать такие же сообщения, как и получатели рассылки. Для этого вам нужна будет **база данных наблюдателей**. Она создаётся так же, как и обычная база данных, только в Основных настройках необходимо отметить соответствующую опцию.

В Smart-настройках выберите **БД наблюдателей** и лимит отправки сообщений для каждого запуска рассылки:



Последовательный запуск броадкастов

Только для броадкаст-рассылок в блоке Smart settings доступна опция автоматической активации после завершения другого броадкаста. Выберите в выпадающем меню рассылок, после завершения которой запустить текущую:

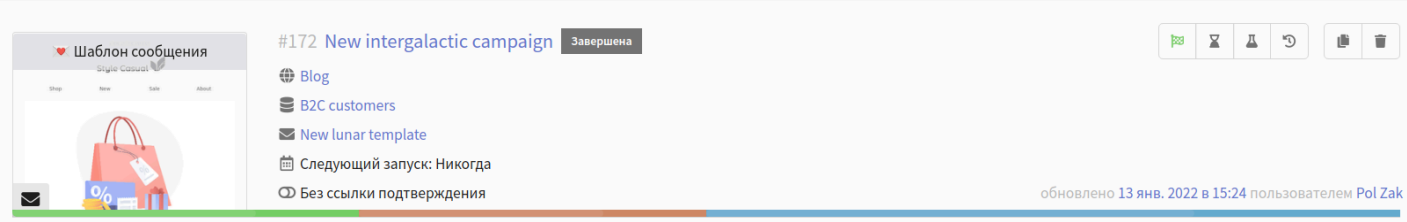


CAUTION

При больших объёмах отправки убедитесь, что рассылка не встречает ограничения со стороны провайдеров email сервисов!

Прогресс отправки рассылок

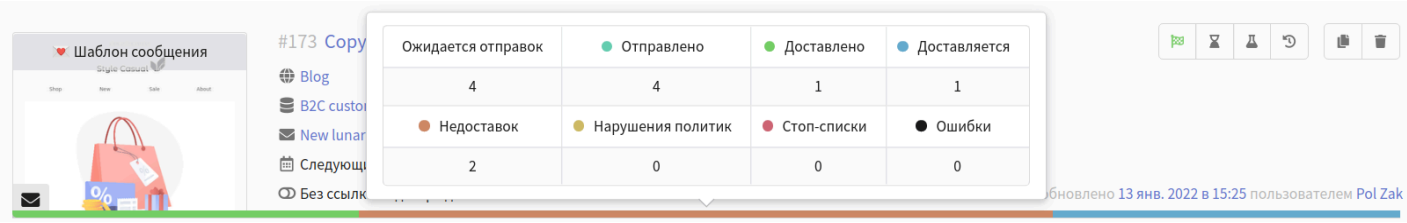
В общем списке броадкаст рассылок отображается прогресс отправки каждой рассылки.



Каждый статус сообщения имеет свой цвет:

- Отправлено – количество профилей, которым были сгенерированы сообщения
- Доставлено – количество профилей, которым было доставлено сообщение
- Доставляется – количество сообщений, находящихся в очереди сендера
- Недоставки – количество профилей, которым не удалось доставить сообщение
- Нарушения политик – количество профилей, попавших под ограничения политик отправки (подробнее)
- Стоп-списки – количество профилей, находящихся в стоп-списке (подробнее)
- Ошибки – иные ошибки при отправках

При нажатии на полосу прогресса откроется таблица со статистикой рассылки:



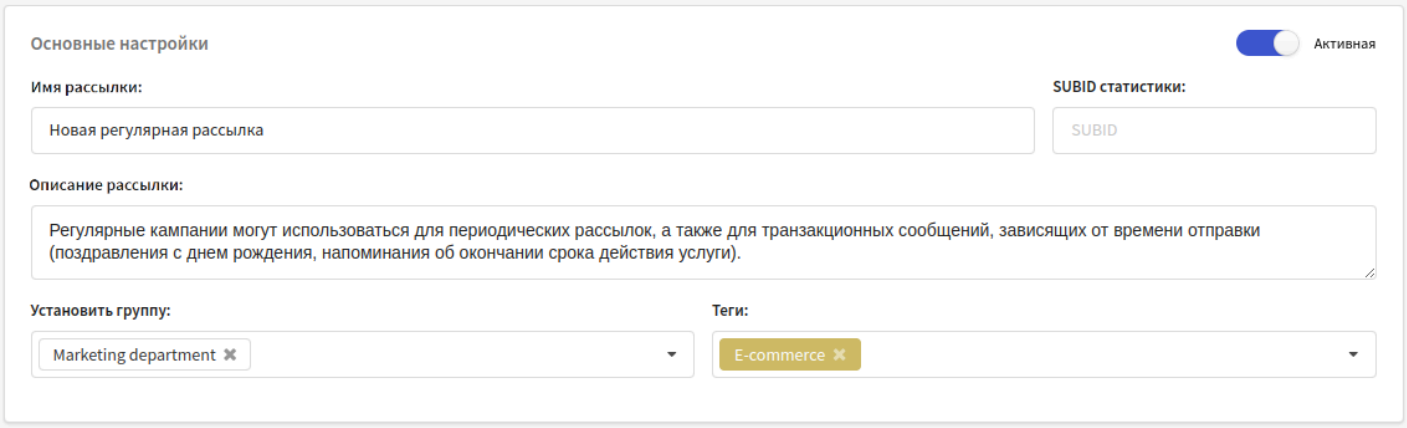
Регулярная рассылка

Регулярная рассылка — инструмент для организации **массовых многократных** рассылок контента подписчикам. **При каждом запуске** по расписанию отправляет по одному сообщению каждому подписчику в источнике данных. Никогда не завершается в отличие от Броадкаст-рассылки, которая обрабатывает базу данных или сегмент подписчиков **только** один раз.

	Чтобы создать новую рассылку, в главном меню выберите **Рассылки** → **Регулярные** и нажмите кнопку **+ Создать**. Или воспользуйтесь календарём рассылок.

Основные настройки

В основных настройках укажите название и описание новой рассылки, при необходимости установите группу доступа, **теги** и **атрибуты** для дополнительной группировки. Здесь же вы можете указать **SUBID статистики** для рассылки. В верхнем правом углу находится переключатель, активирующий рассылку:



Настройки рассылок

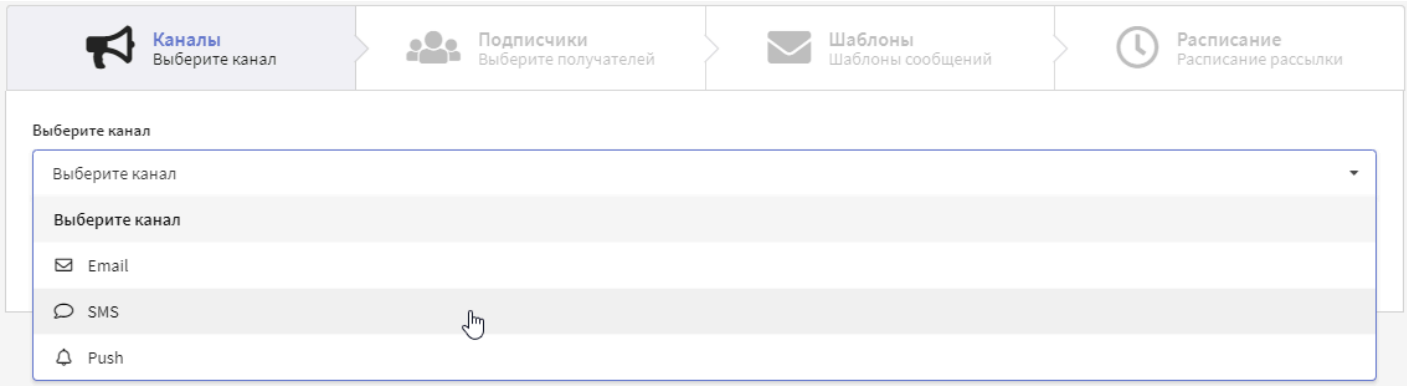
Настройки рассылки объединяют канал, по которому она будет отправлена, получателей сообщений, контент из одного или нескольких шаблонов, а также расписание рассылки.

Каналы

В **первой вкладке** выберите канал связи, по которому хотите отправить рассылку. Канал отправки должен быть доступен хотя бы в одном ресурсе подписок. Ресурс выбирается в следующей вкладке.

NOTE

Для отправки по Email доступно использование основных контактных данных подписчика — поля Email в базе данных.



TIP

При использовании **push-канала** вы можете выбрать конкретных провайдеров для рассылки, а также режим отправки — **fanout**, рассылка на все подписки с учетом фильтров, или **last**, отправка на подписку с самой свежей датой создания. Если несколько подписок были созданы одновременно, то из них будет выбрана одна случайная. Если в рассылке заданы фильтры по провайдеру, дополнительным полям или категориям, то сначала накладываются фильтры, а потом уже ищется самая свежая подписка.

Подписчики

Рассылки в Альткрафт отправляются в рамках определённого ресурса: это может быть отдельный сайт, продукт, акция, или другое направление ваших коммуникаций с клиентами.

Выберите ресурс из выпадающего меню. В ресурсе должен быть настроен выбранный на предыдущем шаге **канал**.

Получателями регулярной рассылки может быть база профилей или сегмент, состоящий из профилей одной или нескольких баз данных. Установите необходимые значения в полях **Источник данных** и **Название** (сегмента или базы данных).

TIP

Источник данных для транзакционных писем

Источник данных регулярной рассылки зависит от её назначения:

- Для периодических рассылок (новости, дайджесты, отчёты) это может быть база данных, статический или динамический сегмент. Отправка будет проходить по настроенной аудитории.
- Для транзакционных сообщений в качестве источника подойдёт только **динамический сегмент**, с выборкой по времени относительно какой-либо даты, например дня роджения подписчика, окончания срока действия услуги.

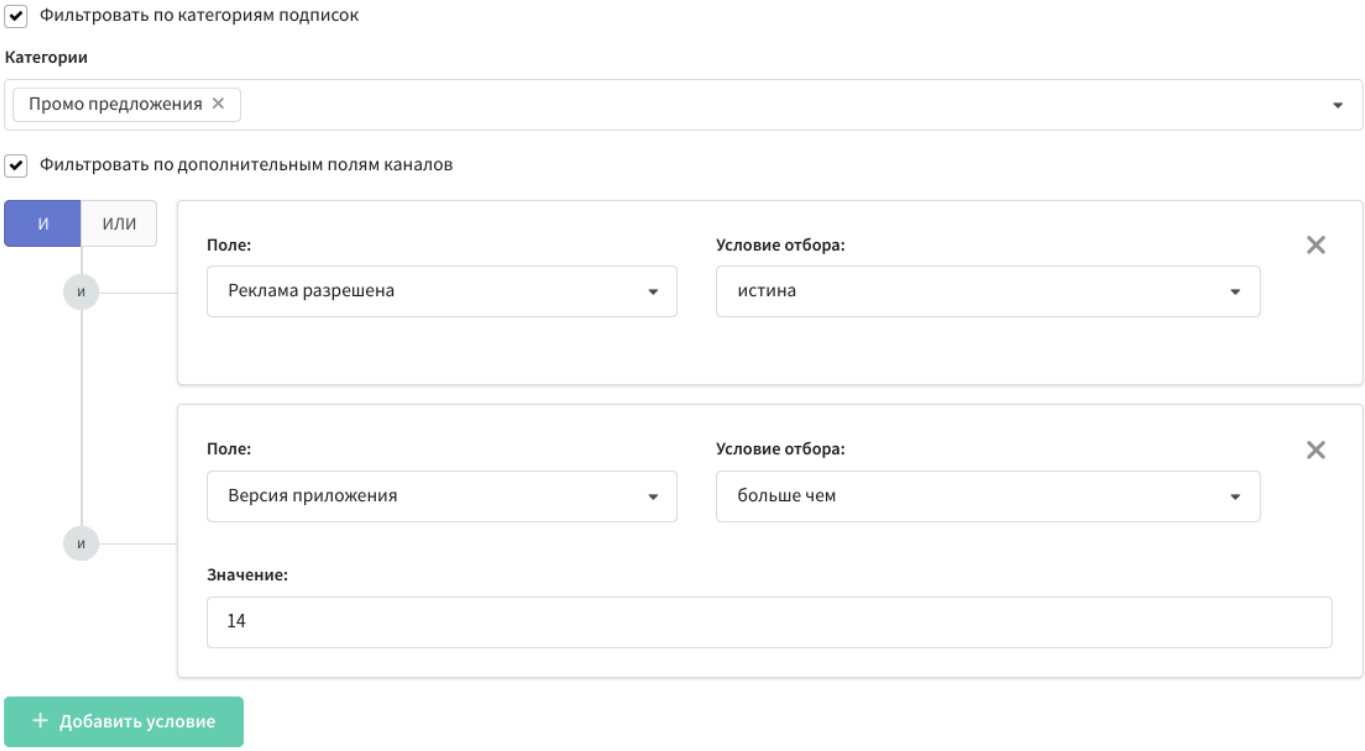
TIP

При отправке сообщений рассылка использует подписки клиентов на выбранный ресурс. Убедитесь, что получатели в вашем источнике данных подписаны на него!

Дополнительно вы можете **отфильтровать получателей по категориям ресурса**. Для этого в ресурсе должны быть созданы категории. В настройках рассылки выберите категории, на которые должен быть подписан профиль, чтобы получить сообщение.

Также можно отфильтровать получателей рассылки по **дополнительным полям канала** в ресурсе. Чтобы создать фильтр по полям, выберите стандартное или дополнительное поле подписки. В поле **“Условие отбора”** выберите логический оператор и при необходимости укажите значение.

В фильтре по полям возможно добавление нескольких правил и их объединение с помощью логического **И** (все условия должны выполняться) и **ИЛИ** (хотя бы одно условие должно выполняться).



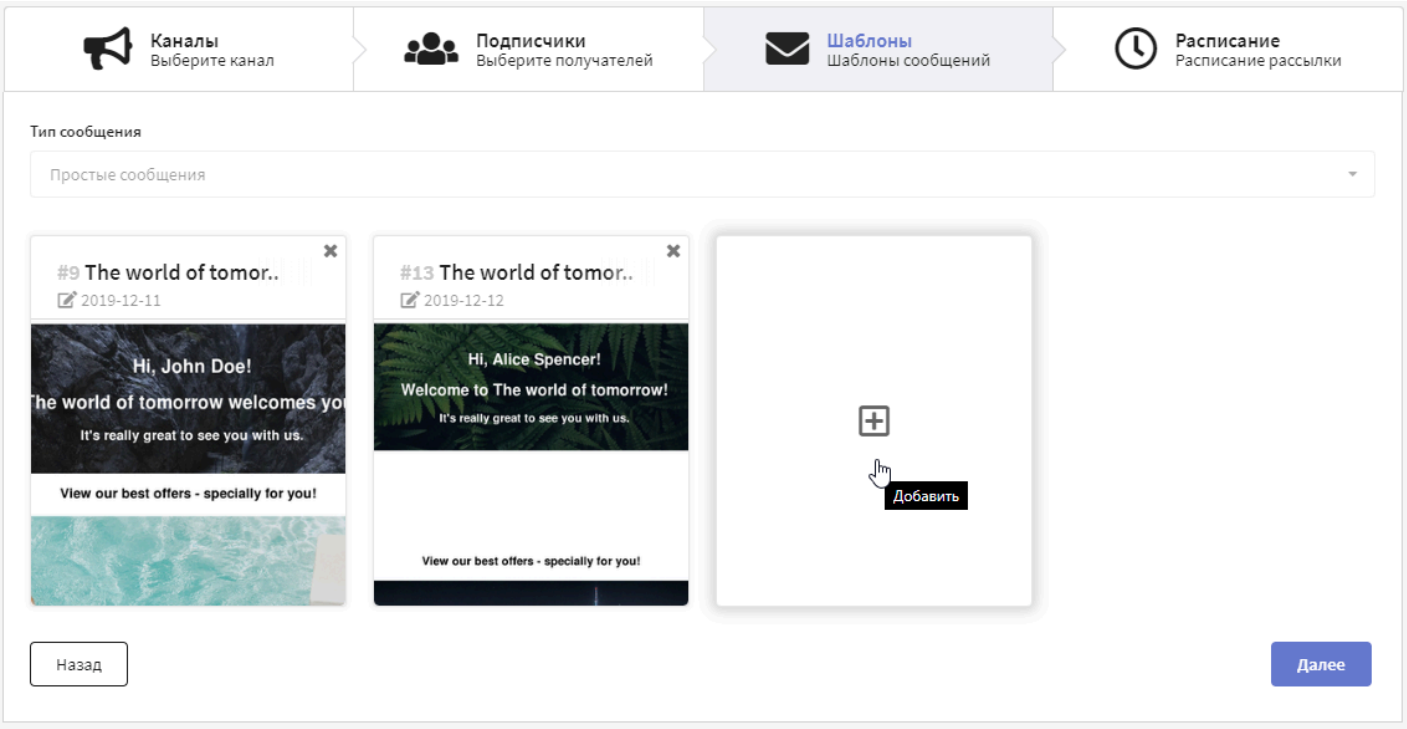
Шаблоны

В этом блоке выберите контент, который хотите использовать в рассылке. Шаблоны сообщений в Альткрафт объединяют контент для отправки по всем доступным каналам.

> **TIP**
>
> Убедитесь, что контент для выбранного в рассылке канала присутствует в шаблоне!

Нажмите на кнопку **Добавить** и выберите подготовленный шаблон. Любой тип рассылки поддерживает до 8 шаблонов сообщений. Регулярная рассылка отправит каждому получателю по **одному** сообщению из выбранных в случайном порядке - в равных пропорциях.

По клику на миниатюру шаблона откроется **предпросмотр**. При нажатии на название шаблона он откроется в новой вкладке для редактирования:



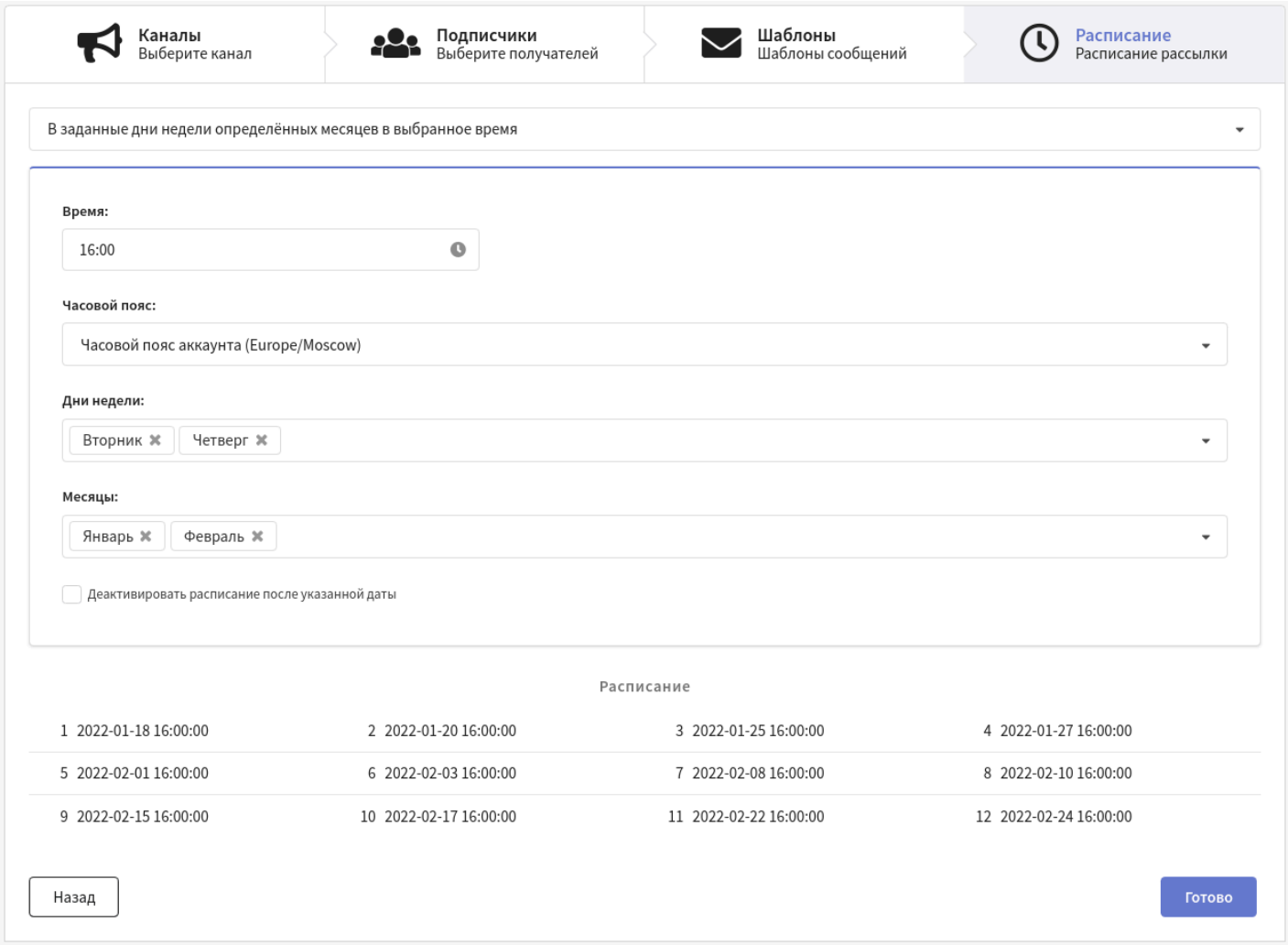
Расписание

> **TIP**
>
> Расписание кампаний подробно описано в отдельной статье.

При проведении массовых рассылок необходимо иметь во внимании такой показатель, как репутация отправителя (домена и IP адресов, с которых ведётся рассылка). Провайдеры электронной почты устанавливают ограничения на массовые рассылки - в зависимости от репутации и количества отправляемых сообщений. Обычно провайдеры устанавливают максимум принимаемых сообщений на 24 часа.

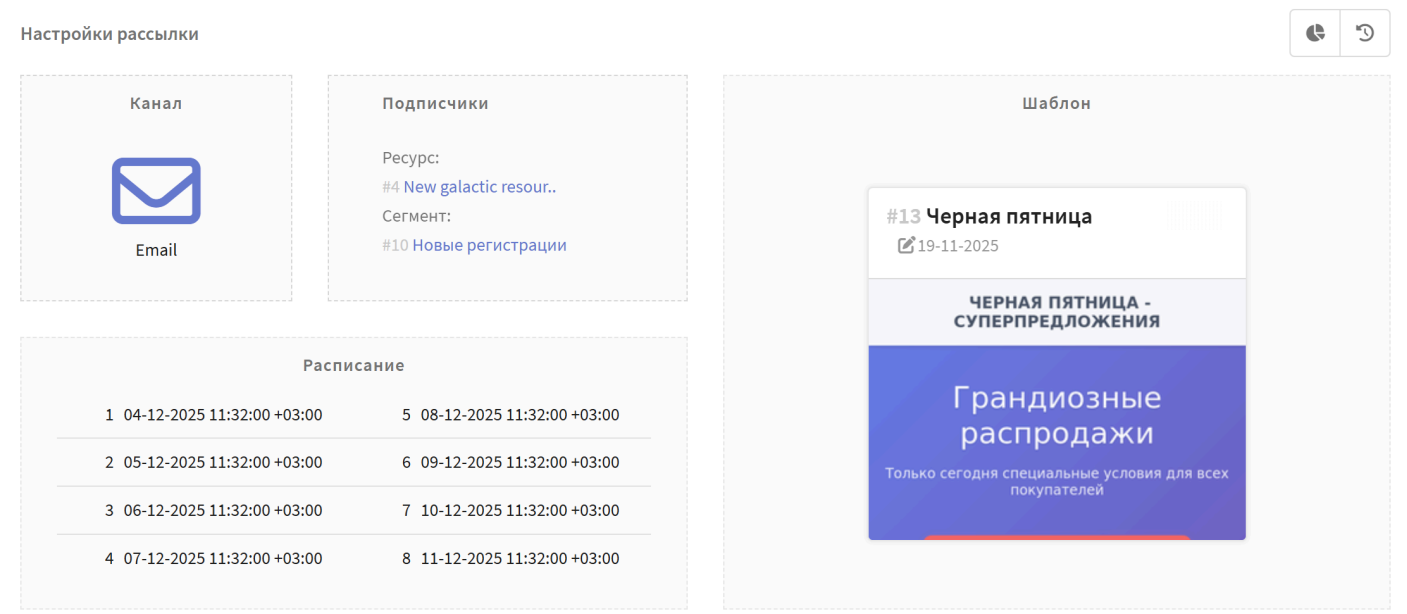
Расписание регулярной рассылки

Расписание регулярной рассылки зависит от того, какой тип рассылки вы хотите организовать. Можно отдельно настроить минуты, часы, дни недели, месяцы и даже определённые дни недели относительно начала месяца:



Дополнительно при некоторых вариантах расписания вы можете установить **часовой пояс** для отправки или отправку в часовом поясе клиента, если такие данные доступны. Если вам нужно прекратить любые рассылки через какое-то время, воспользуйтесь опцией **Деактивировать расписание после указанной даты**.

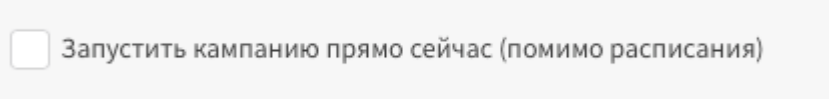
Когда вы завершите настройку, нажмите **Готово**. Выбранные вами параметры будут отображаться в блоке **Настройки рассылки**. Кнопками в правом верхнем углу блока вы можете **редактировать** настройки или перейти к отчету по рассылкам.



Мгновенный запуск рассылки (помимо расписания)

Чтобы запустить рассылку сразу же после создания, в самом низу страницы отметьте соответствующую опцию.

- Если лимиты не установлены – сообщения отправятся всем получателям.
- Если рассылка разбита на несколько частей – произойдёт запуск помимо расписания.



Ограничения рассылки

В этом блоке настраивается лимит рассылки и политка отправки.

Использование лимитов в регулярной рассылке

> **TIP**
>
> Регулярная рассыдлка не останавливается автоматически, отправив по сообщению каждому подписчику. Для массовых **email** рассылок, которые нужно отправлять порционно, лучше использовать **Броадкаст-рассылку**.

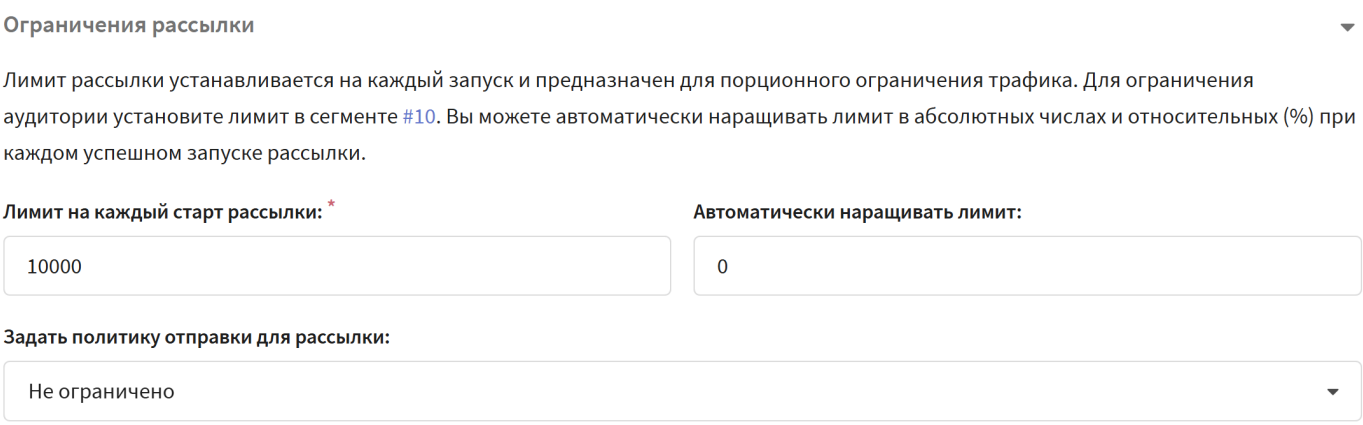
Лимиты в регулярной рассылке можно использовать, когда количество отправок встречает ограничения со стороны почтовых провайдеров.

> **NOTE**
>
> Другой сценарий использования — распределение нагрузки на сендер (модуль отправки), или на инфраструктуру вашей организации. Например, при ограничениях канала отправки сообщений.

В этих случаях установите ограничения рассылки:

- **Лимит на каждый старт расылки**– количество сообщений, которое будет отправлено в следующий запуск рассылки по расписанию.

> **DANGER**
>
> Опция **Автоматически наращивать лимит** не рекомендуется к использованию в регулярных рассылках.



Запуск рассылки по расписанию в этом случае будет отправлять то количество сообщений, которое указано в лимите. Когда количество отправленных сообщений превысит количество получателей, отправка продолжится с начала списка. Чтобы исключить повторную отправку одного и того же сообщения, установите **Политику отправки** (см. следующий пункт).

CAUTION

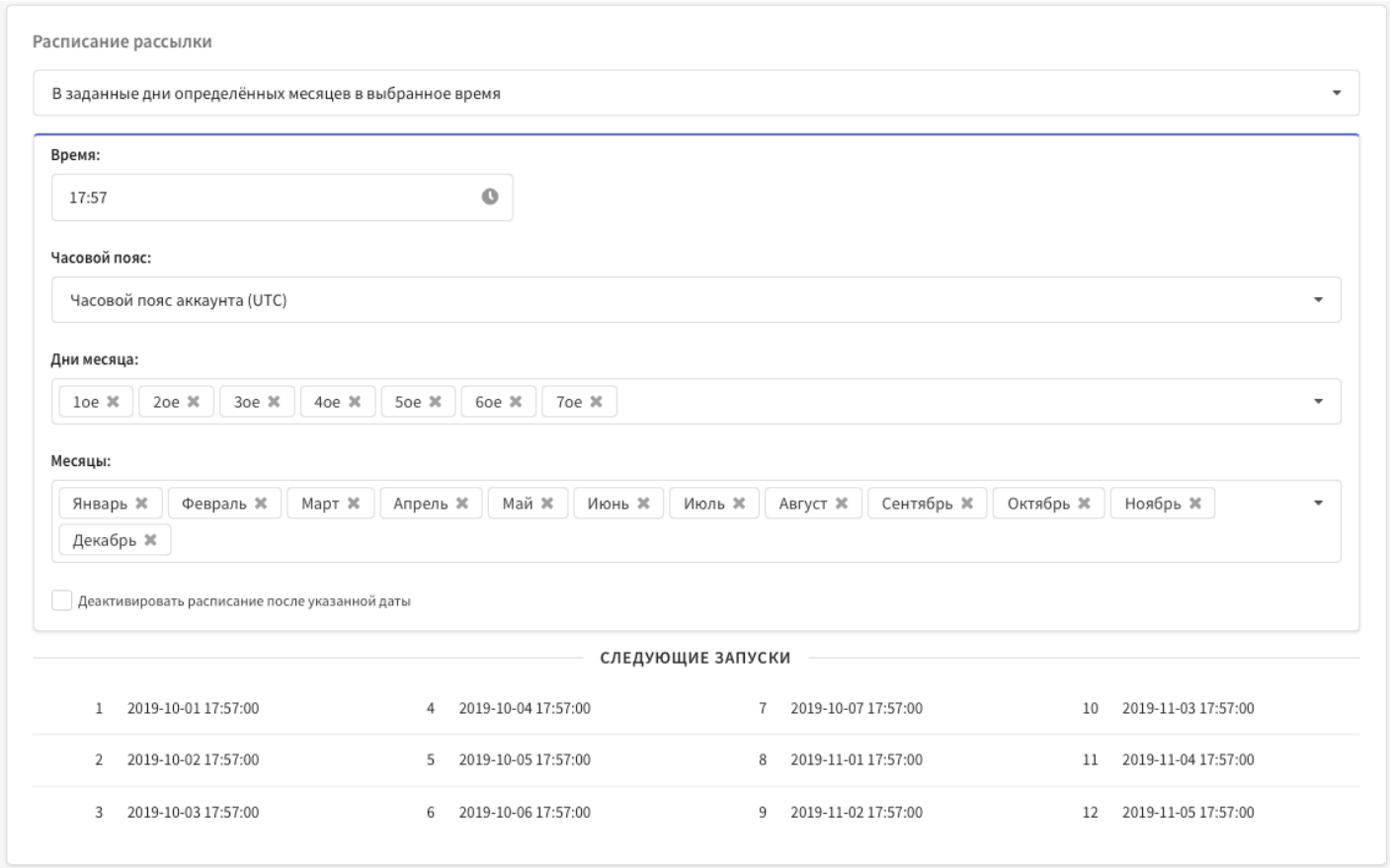
Максимальный срок ограничения в политике – 16 дней, так что период отправки (например, ежемесячной рассылки) не должен превышать это значение.

CAUTION

При активированной опции “Отправлять в часовом поясе профиля, если он известен”, использование лимитов и мгновенный запуск рассылки становятся недоступны.

Ниже приведён пример расписания ежемесячной рассылки **для порционной отправки** в первую неделю месяца. В этом случае необходимо установить:

- Лимит на каждый старт – больше 1/7 аудитории рассылки.
- Политику отправки – максимум одно сообщение подписчику за 7 дней.



Политика отправки

Политика отправки ограничивает коммуникацию с одним подписчиком. Настраивается для рассылки в блоке **Ограничения рассылки** (см. предыдущий пункт). По умолчанию рассылка будет использовать политику, выбранную глобально в аккаунте или в ресурсе, от имени которого отправляется регулярная рассылка.

Smart settings

Блок **Smart-настройки** объединяет следующие дополнительные настройки рассылки:

- **Виртуальный сендер** — позволяет выбрать для рассылки виртуальный сендер (набор правил для отправки сообщений с реальных сендеров платформы).
- **Свои трекинг-домены** — позволяет отслеживать действия с сообщениями рассылки, используя дополнительные трекинг-серверы. Каждый домен указывается на отдельной строке.
- **Список email адресов для скрытой копии** — BCC для отправляемых сообщений. Каждый адрес указывается на отдельной строке.
- **Не проверять контактные данные на уникальность** — отключает проверку на уникальность получателей рассылки по данным подписки.

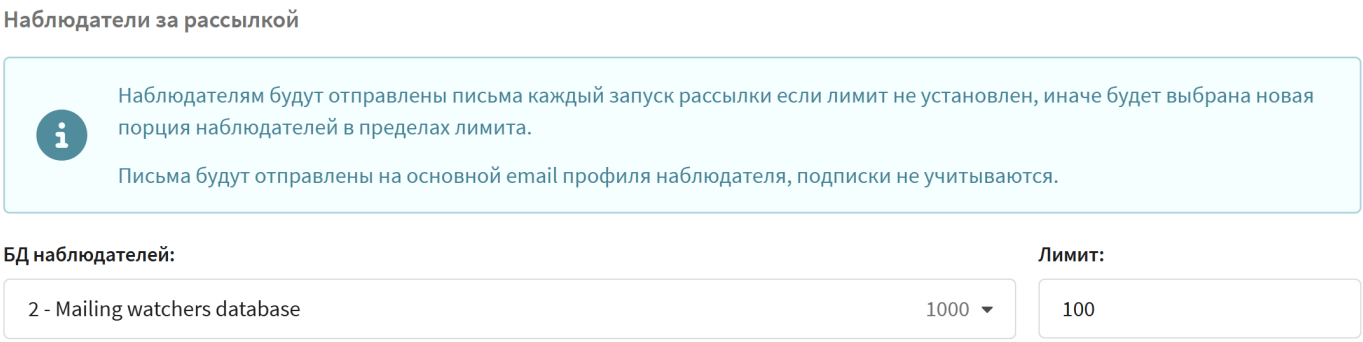
CAUTION

Если вы не уверены, какие настройки необходимо выбрать, проконсультируйтесь с администратором платформы в вашей организации или со специалистами поддержки Altcraft.

Наблюдатели за рассылкой

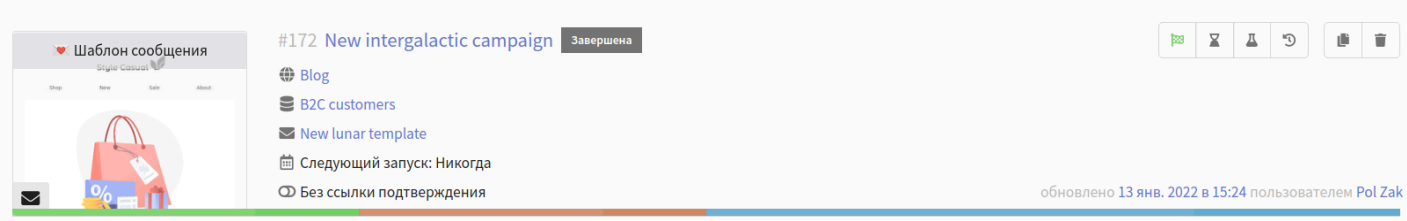
Помимо этого, в умных настройках можно установить **наблюдателей за рассылкой**. Наблюдатели будут получать такие же сообщения, как и получатели рассылки. Для этого вам нужна будет **база данных наблюдателей**. Она создаётся так же, как и обычная база данных, только в Основных настройках необходимо отметить соответствующую опцию.

В Smart-настройках выберите **БД наблюдателей** и лимит отправки сообщений для каждого запуска рассылки:



Прогресс отправки рассылки

В общем списке регулярных рассылок отображается прогресс отправки каждой рассылки.

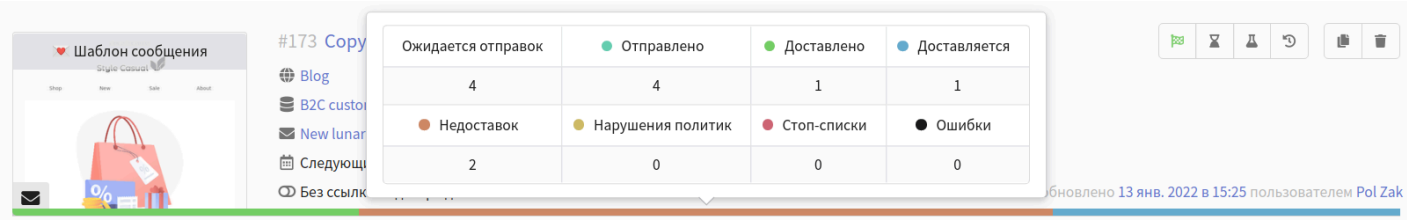


Каждый статус сообщения имеет свой цвет:

- Отправлено — количество профилей, которым были сгенерированы сообщения
- Доставлено – количество профилей, которым было доставлено сообщение
- Доставляется – количество сообщений, находящихся в очереди сендера
- Недоставки – количество профилей, которым не удалось доставить сообщение
- Нарушения политик – количество профилей, попавших под ограничения политик отправки (подробнее)

- Стоп-списки – количество профилей, находящихся в стоп-списке (подробнее)
- Ошибки – иные ошибки при отправках

При нажатии на полосу прогресса откроется таблица со статистикой рассылки:



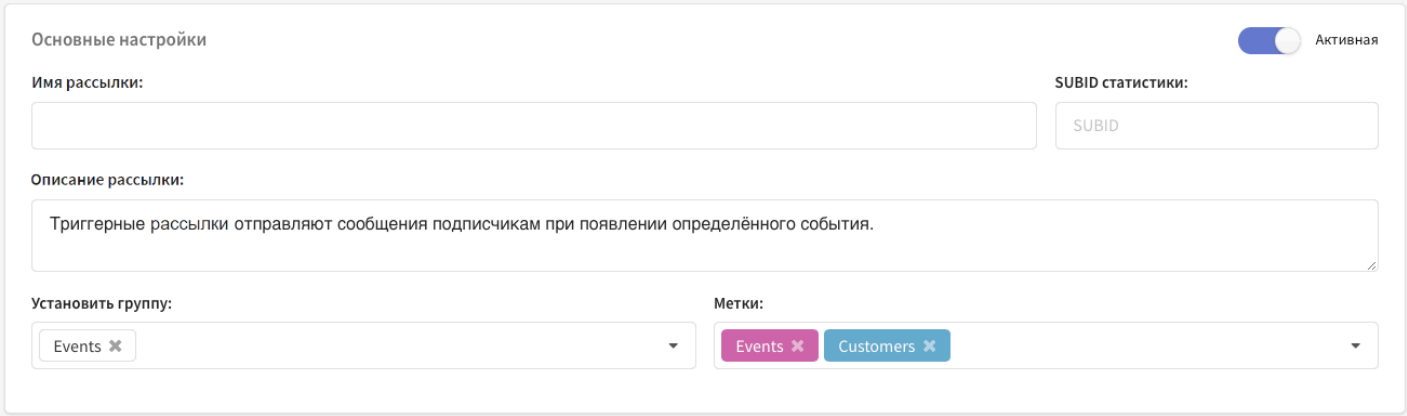
Триггерная рассылка

Триггерная рассылка — это инструмент создания транзакционных сообщений в Altcraft Platform. Триггерная рассылка отличается от других рассылок тем, что не имеет расписания и лимита. Вместо этого триггер постоянно находится в ожидании события, которое его запустит. Рассылает по одному сообщению каждому подписчику, который попал под условие запуска триггера.

	Чтобы создать новую рассылку, в главном меню выберите **Рассылки** → **Триггеры** и нажмите кнопку **+ Создать**. Или воспользуйтесь календарём рассылок.

Основные настройки

В основных настройках укажите название и описание новой рассылки, при необходимости установите группу доступа, **теги** и **атрибуты** для дополнительной группировки. Здесь же вы можете указать **SUBID статистики** для рассылки. В верхнем правом углу находится переключатель, активирующий триггер, переводя его в режим ожидания:



Настройки рассылки

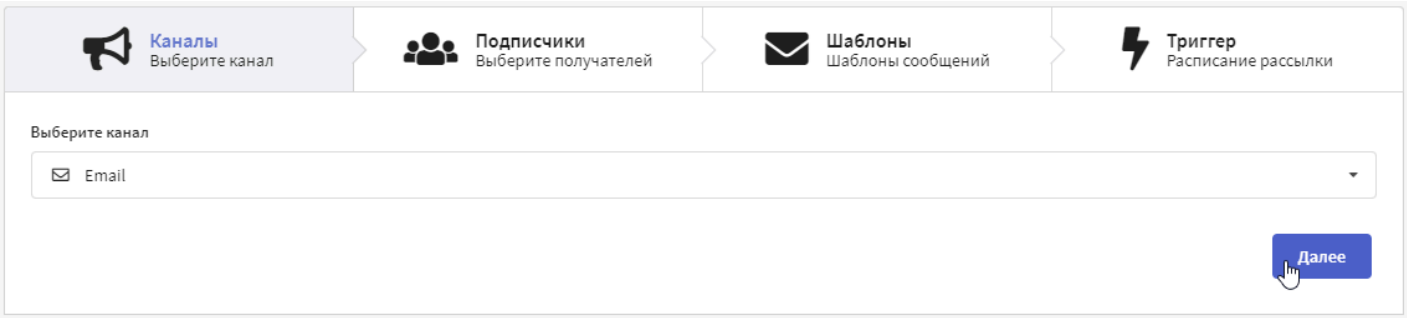
Настройки рассылки объединяют канал, по которому она будет отправлена, получателей сообщений, контент из одного или нескольких шаблонов, а также расписание рассылки.

Каналы

В **первой вкладке** выберите канал связи, по которому хотите отправить рассылку. Канал отправки должен быть доступен хотя бы в одном ресурсе подписок. Ресурс выбирается в следующей вкладке.

NOTE

Для отправки по Email доступно использование основных контактных данных подписчика - поля Email в базе данных.



Подписчики

Рассылки в Альткрафт отправляются в рамках определённого Ресурса: это может быть отдельный сайт, продукт, акция, или другое направление ваших коммуникаций с клиентами.

Выберите ресурс из выпадающего меню. В ресурсе должен быть настроен выбранный на предыдущем шаге **канал**.

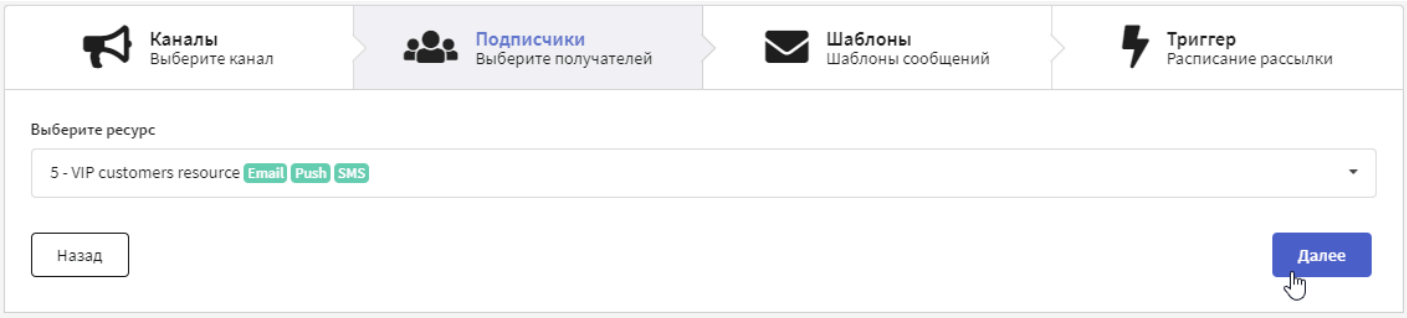
TIP

При отправке сообщений рассылка использует подписки клиентов на выбранный ресурс. Убедитесь, что получатели в вашем источнике данных подписаны на него, или что подписка передаётся при импорте.

Источника данных (база данных или сегмент подписчиков) у триггерной рассылки нет. Аудитория рассылки зависит от запускающего триггер события.

CAUTION

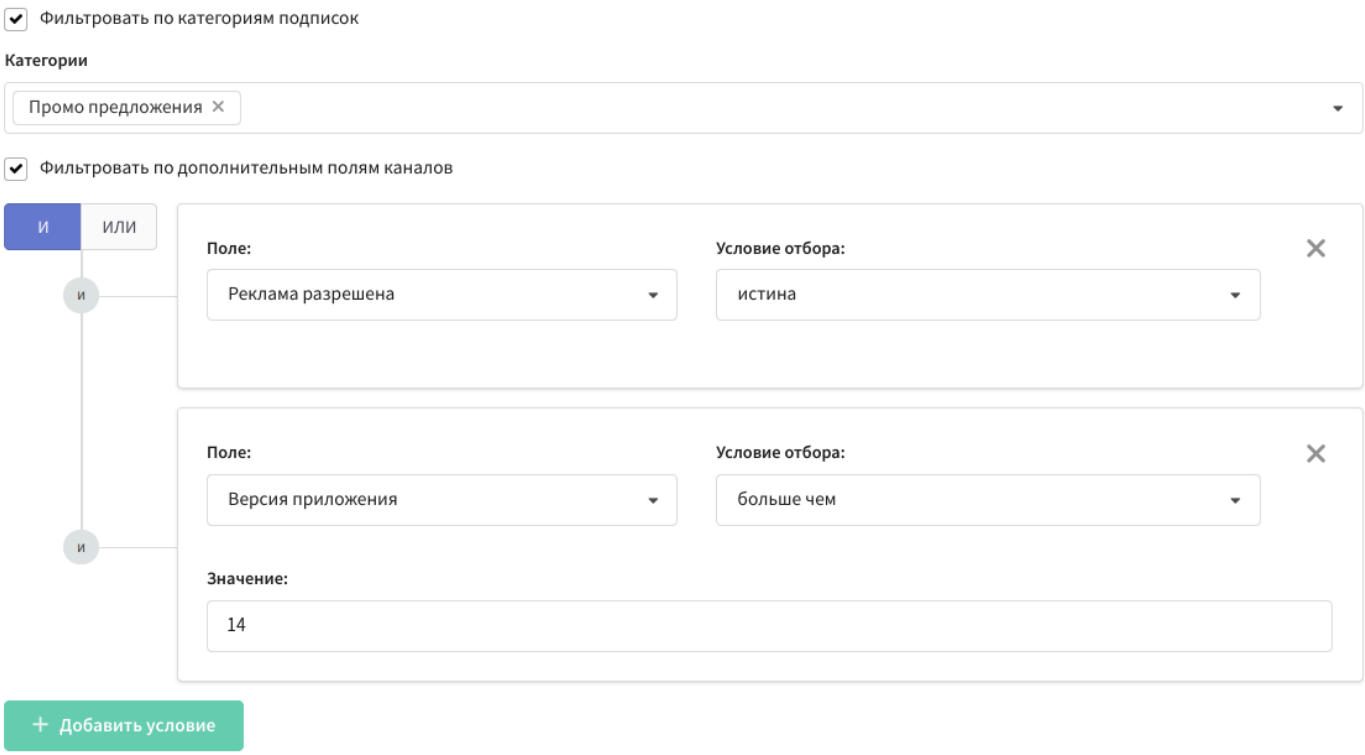
Если триггерное событие — импорт в базу данных, убедитесь, что база данных доступна в выбранном **Ресурсе** рассылки! В противном случае сообщения отправлены не будут.



Дополнительно вы можете **отфильтровать получателей по категориям ресурса**. Для этого в ресурсе должны быть созданы категории. В настройках рассылки выберите категории, на которые должен быть подписан профиль, чтобы получить сообщение.

Также можно отфильтровать получателей рассылки по **дополнительным полям канала** в ресурсе. Чтобы создать фильтр по полям, выберите стандартное или дополнительное поле подписки. В поле **“Условие отбора”** выберите логический оператор и при необходимости укажите значение.

В фильтре по полям возможно добавление нескольких правил и их объединение с помощью логического **И** (все условия должны выполняться) и **ИЛИ** (хотя бы одно условие должно выполняться).



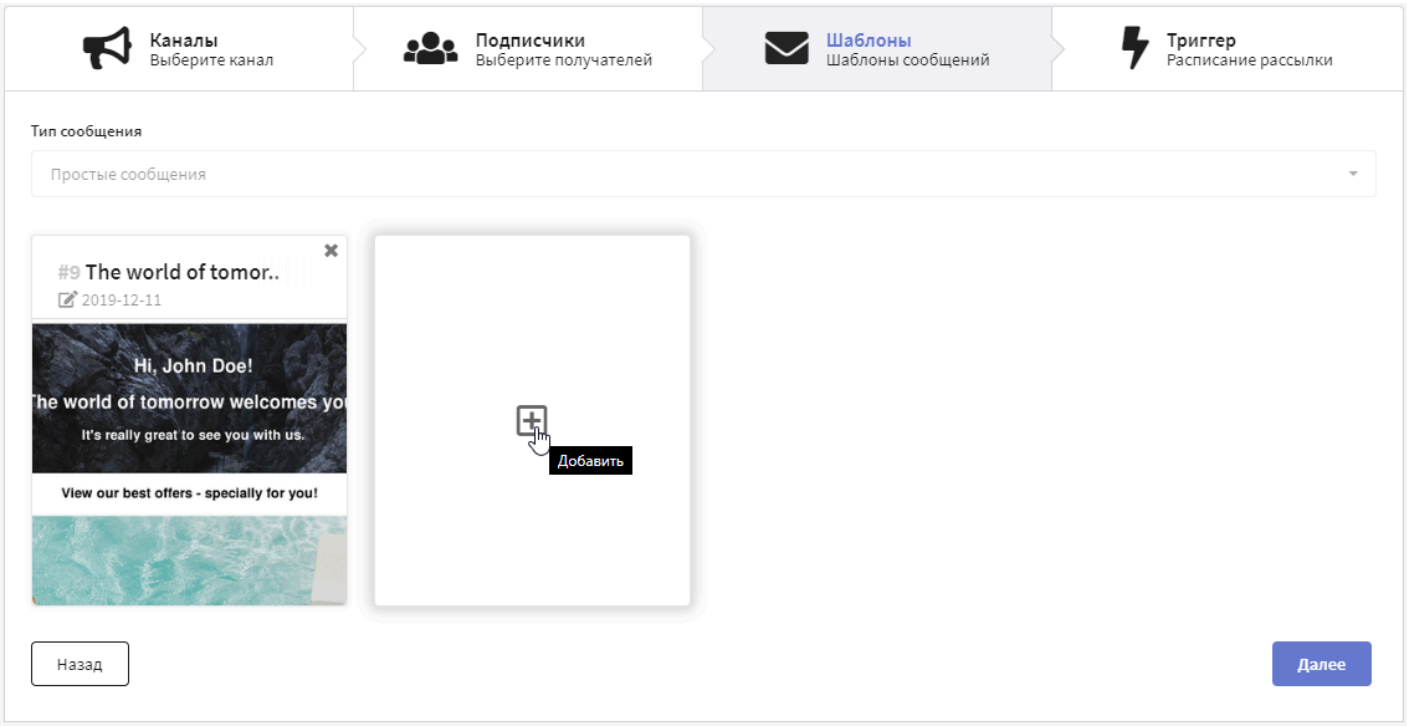
Шаблоны

В этом блоке выберите контент, который хотите использовать в рассылке. Шаблоны сообщений в Альткрафт объединяют контент для отправки по всем доступным каналам.

> **TIP**
>
> Убедитесь, что контент для выбранного в рассылке канала присутствует в шаблоне!

Нажмите на кнопку **Добавить** и выберите подготовленный шаблон. Любой тип рассылки поддерживает до 8 шаблонов сообщений. Триггерная рассылка отправит каждому получателю по **одному** сообщению из выбранных в случайном порядке - в равных пропорциях.

По клику на миниатюру шаблона откроется **предпросмотр**. При нажатии на название шаблона он откроется в новой вкладке для редактирования:



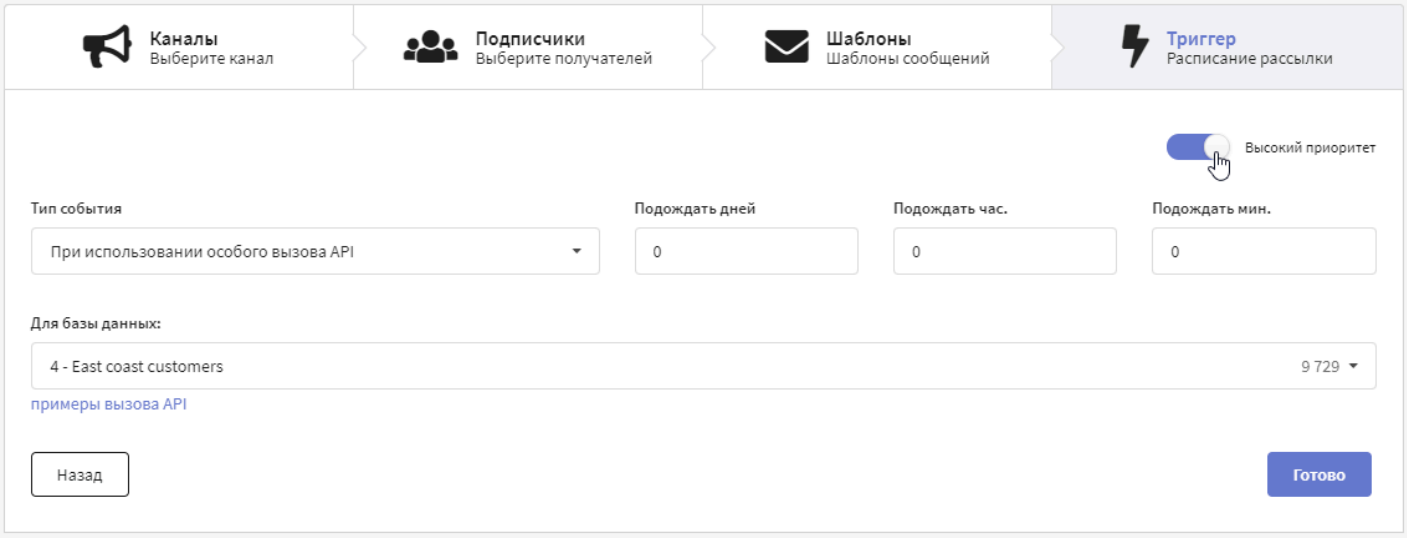
Триггер

Сообщения триггерных рассылок мгновенно отправляются по приоритетному каналу сендера, как только наступает триггерное событие.

Внутри канала также можно установить приоритет отправки опцией **Высокий приоритет**. В этом случае задачи на генерацию контента обрабатываются в отдельной приоритетной очереди в брокере (`oxy_triggers_prior`), что позволяет быстрее подготовить сообщение.

После генерации контента сообщение передаётся в общую очередь на отправку, где приоритет больше не учитывается — сообщения отправляются в порядке поступления. Исключение — отправка через email-канал с использованием AKMTA. В этом случае внутри сендера выделяется отдельная виртуальная очередь для приоритетных сообщений, что ускоряет их отправку. Для остальных каналов приоритет не влияет на порядок отправки на стороне сендера.

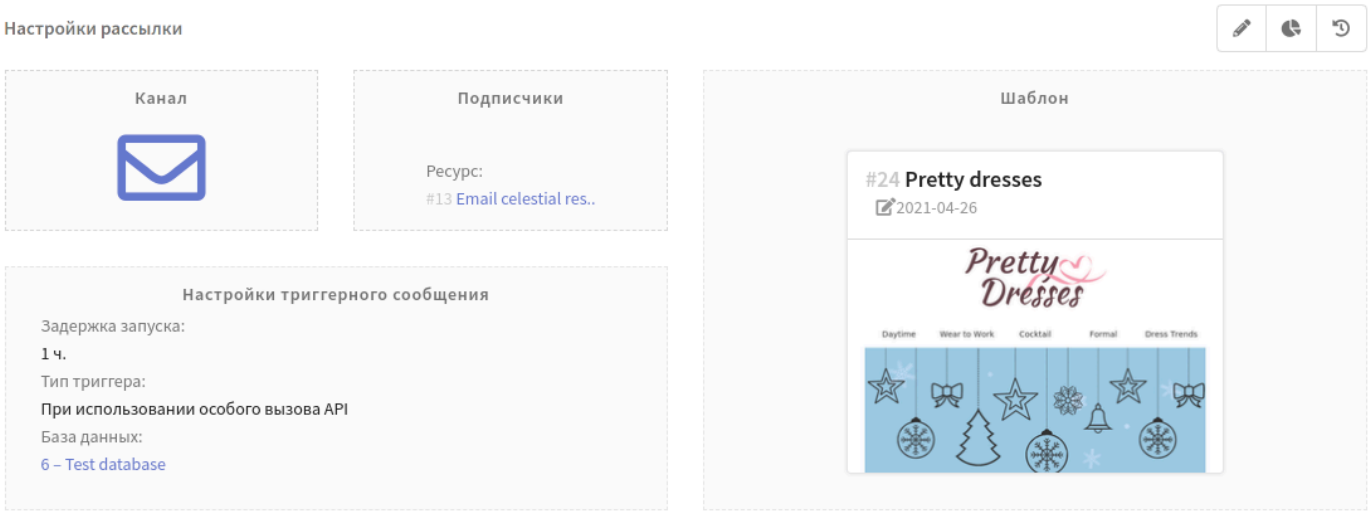
Для каждого типа триггера можно установить **время задержки перед отправкой** сообщения: в днях, часах и минутах.



TIP

Типы триггерных событий описаны в следующем пункте этой статьи!

Когда вы завершите настройку, нажмите **Готово**. Выбранные вами параметры будут отображаться в блоке **Настройки рассылки**. Кнопками в правом верхнем углу блока вы можете **редактировать** настройки, посмотреть журнал рассылки или перейти к его отчету.



NOTE

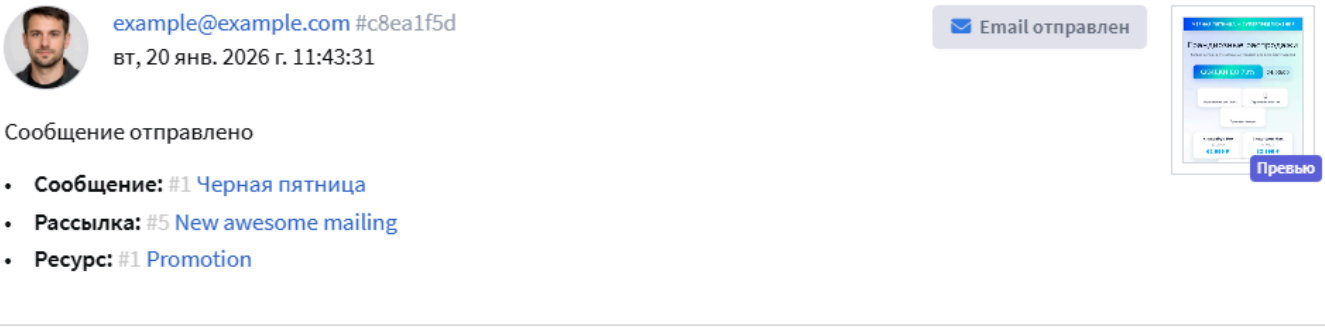
В журнале кампании для триггерных рассылок в поле **“Всего времени затрачено”** может отображаться значение **30 секунд**, даже если сообщение было отправлено практически сразу после наступления события.

Время	Тип	Значение
2026-01-19 11:29:53	Запуск триггера	
2026-01-19 11:29:53	ID базы данных	6
2026-01-19 11:30:23	Количество профилей полученных из источника	1
2026-01-19 11:30:23	Количество найденных подписчиков	1
2026-01-19 11:30:23	Количество подписчиков, кому был отправлен email	1
2026-01-19 11:30:23	Триггер завершен	
2026-01-19 11:30:23	Рассылка завершена	
2026-01-19 11:30:23	Всего времени затрачено	30.066781361s

Для триггерных рассылок значение **“Всего времени затрачено”** фиксируется **в момент завершения кампании**, а не в момент фактической отправки конкретного сообщения. В течение этого интервала триггер продолжает принимать новые события, после чего завершается и записывает общую статистику за период.

Поэтому в журнале отображается одинаковое значение времени, независимо от того, когда именно было отправлено конкретное письмо.

Чтобы узнать **фактическое время генерации и отправки сообщения**, используйте **историю профиля** — в ней фиксируются соответствующие события.

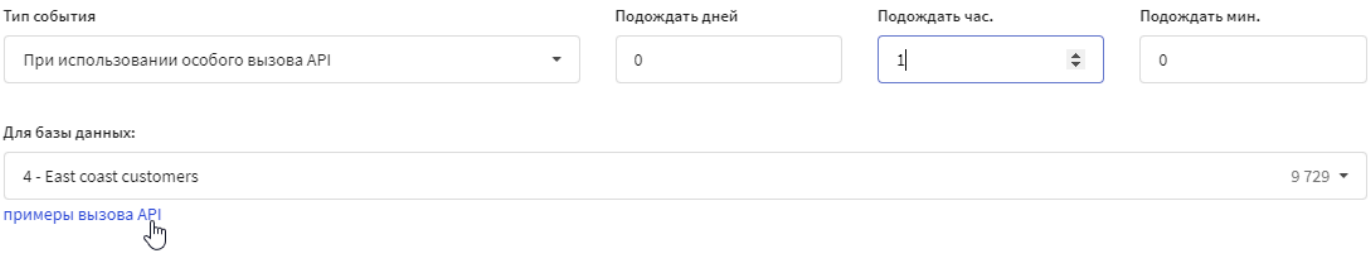


Типы триггерных событий

Тип триггерного события выбирается во вкладке **Триггер** настроек рассылки. Эти же события используются [сценариями автоматизации](#) при индивидуальном запуске.

При использовании особого вызова API

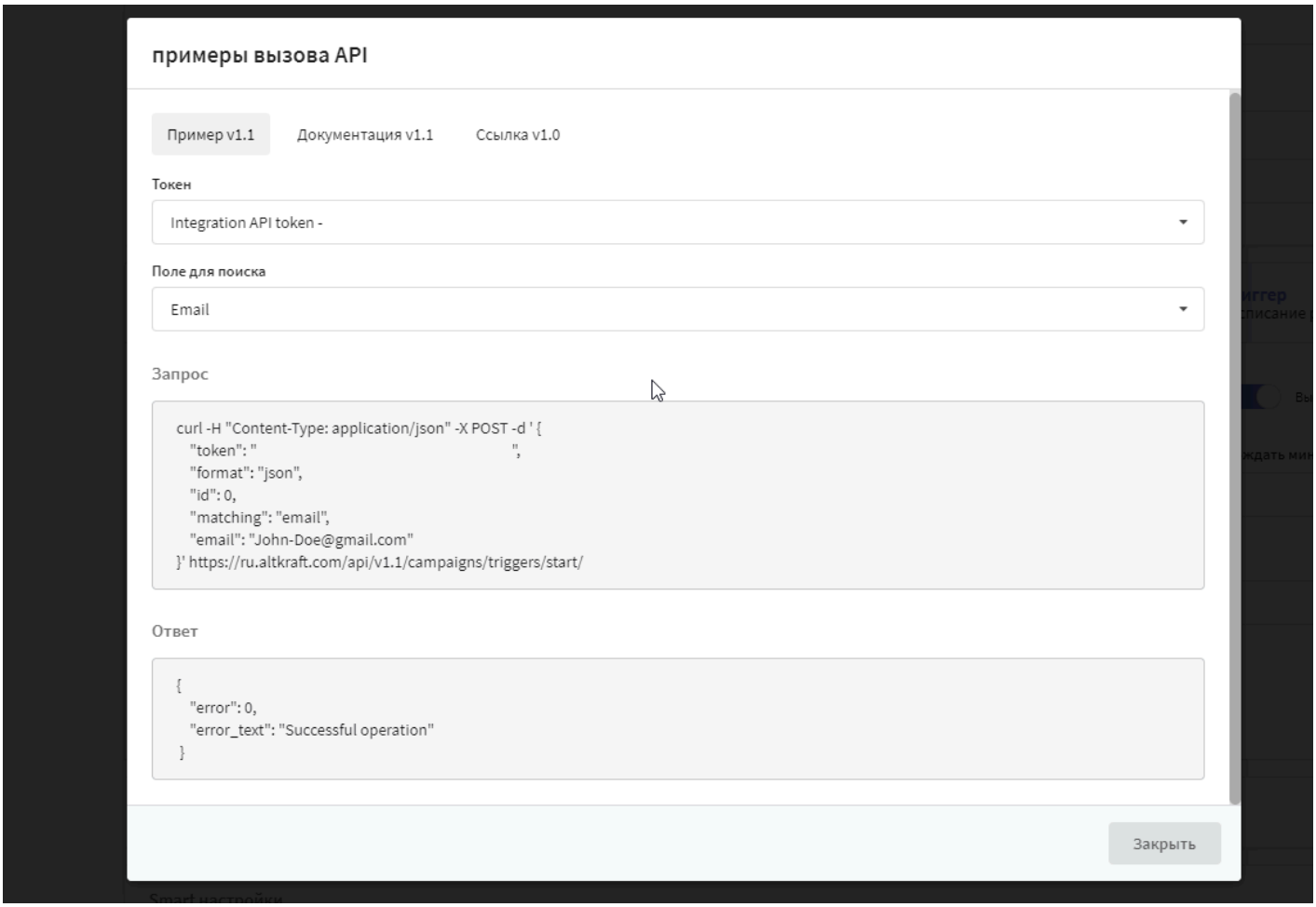
Один из наиболее эффективных вариантов триггера: рассылка запускается [API запросом](#) из внешних систем: с сайта, из приложения, CRM или другого сетевого ресурса. Триггер настраивается для конкретной **базы данных**.



Для запуска триггерной рассылки доступны два варианта запроса к API:

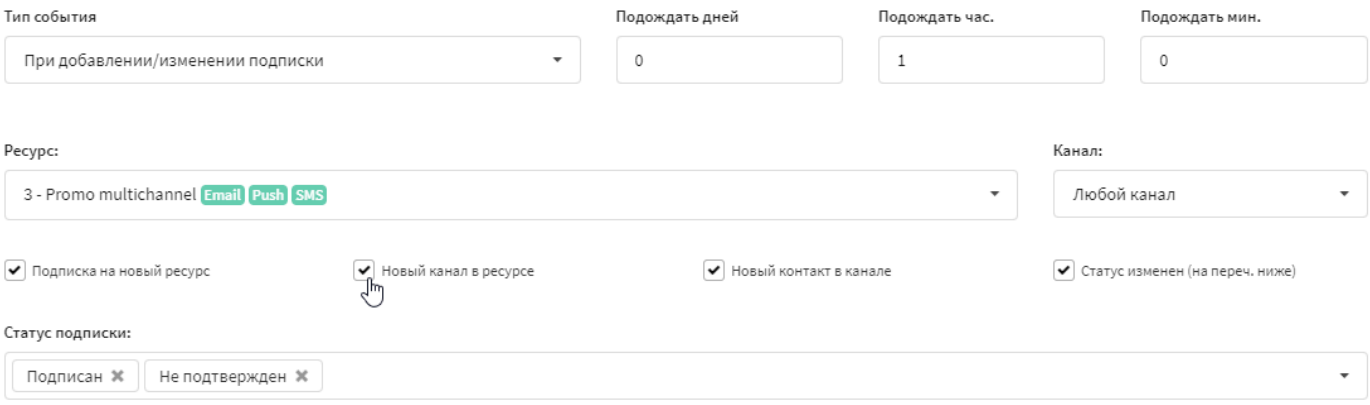
- **В версии 1.0** возможно только запустить рассылку по ссылке, передав email, md5 или xxh нужного профиля. Используется простой GET запрос.
- **В версии 1.1** можно передать рассылке дополнительные параметры.

Нажмите **примеры вызова API** чтобы получить примеры запроса на запуск триггера:



При добавлении/изменении подписки

Активируется, когда для профиля клиента добавляется подписка или изменяется статус существующей. Для триггера можно отдельно указать **Ресурс** и **Канал**, по которым будут отслеживаться изменения. В ином случае рассылка будет ожидать изменений по всем каналам выбранного ресурса, по всем ресурсам для выбранного канала - или по любым подпискам профиля клиента.



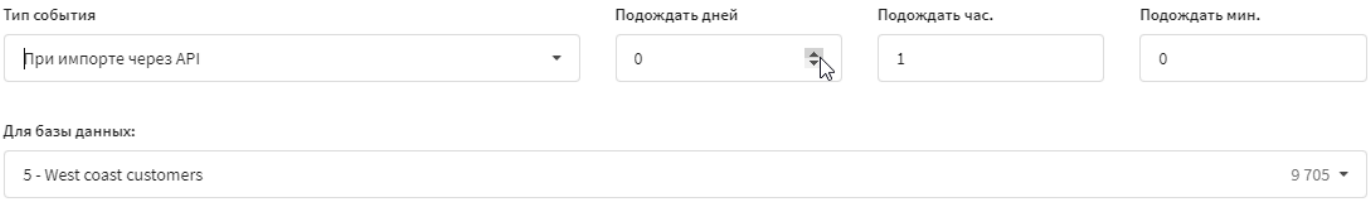
- **Подписка на новый ресурс** — условие сработает только для первой подписки на указанный в рассылке ресурс. При добавлении других подписок на тот же ресурс триггер или сценарий не активируется.
- **Новый канал в ресурсе** — добавлена подписка на новый (для профиля) канал в ресурсе. То есть, была подписка на email рассылки, добавилась ещё и на SMS или Push.

- **Новый контакт в канале** — активирует триггер при каждом появлении новой подписки. Первая подписка на канал также считается “новым контактом в канале”.
- **Статус изменён** — рассылка запустится при изменении статуса подписки на выбранные из выпадающего меню.
- **Первые три пункта выбраны** — триггер сработает при каждой новой подписке.
- **Все галочки сняты** — ничего не происходит.

> **CAUTION**
>
> На подписки, которые добавлены со статусом, отличным от “Подписан”, триггер не активируется.

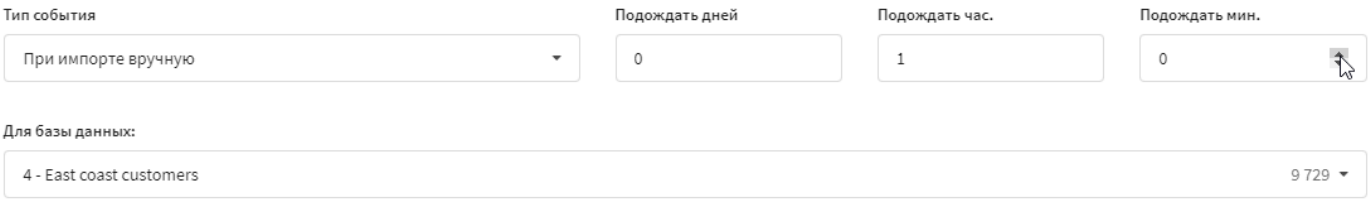
При импорте через API

Активируется, когда в базу данных добавляется новый профиль по API. Триггер настраивается для конкретной **базы данных**.



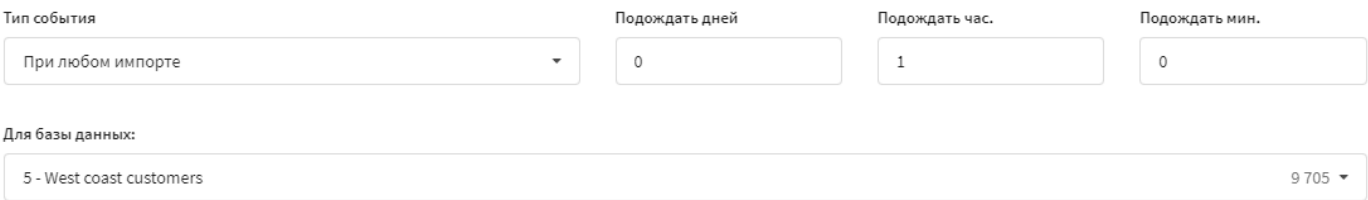
При импорте вручную

Активируется, когда в базу данных добавляется новый профиль с помощью формы импорта или из файла. Триггер настраивается для конкретной **базы данных**.



При любом импорте

Активируется, когда в базу данных добавляется новый профиль любым способом. Триггер настраивается для конкретной **базы данных**.

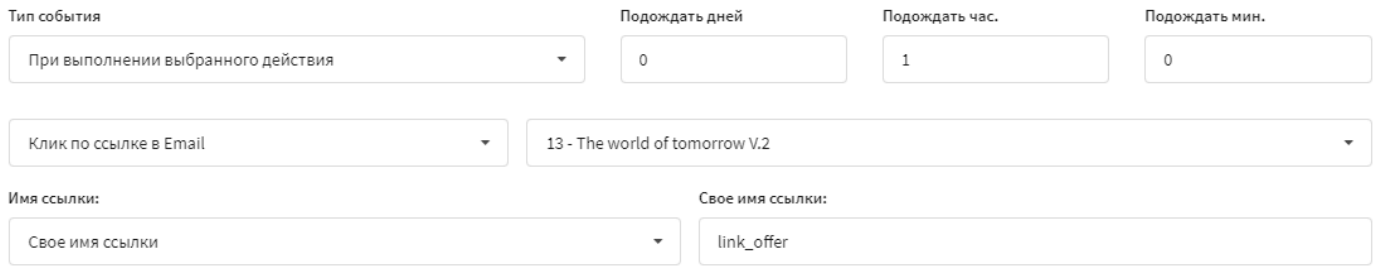


При выполнении выбранного действия

Активируется, когда **подписчик** совершает нужное **действие в сообщении**. Для выбранного действия укажите необходимый шаблон сообщения из выпадающего меню. Для переходов по ссылкам есть два варианта - клик по любой ссылке в сообщении и клик по конкретной ссылке, которую можно указать в соответствующем поле.

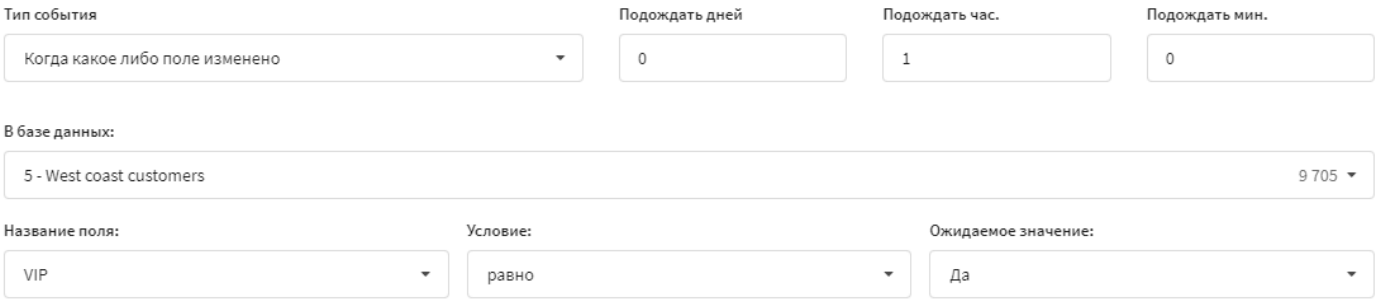
CAUTION

Триггер на **любой клик** будет срабатывать для **первого** перехода по любой ссылке в сообщении за 24 часа. Триггер на **клик по ссылке** будет захватывать первый клик по **заданной** ссылке за последние 24 часа.



Когда какое-либо поле изменено

Активируется, когда заданное поле принимает нужное значение. Необходимо выбрать поле профиля, условие и ожидаемое значение:



Условия различаются в зависимости от типа поля:

Условие	Тип поля	Описание
изменено	все	Любое изменение в поле профиля
равно	все, кроме *теги*, *объект*, *массив объектов*	Новое значение равно ожидаемому

Условие	Тип поля	Описание
не равно	все, кроме *теги*, *объект*, *массив объектов*	Новое значение не равно ожидаемому
не пусто е	все, кроме *теги*	Поле получило любое не пустое значение
пусто е	все, кроме *теги*	Все значения удалены для поля профиля
добав лен любо й тег из списк а	*теги*	Список определяется в триггере
удалё н любо й тег из списк а	*теги*	Список определяется в триггере
свойс тво измен илось	*объект*	Указывается путь к ключу объекта. Если значение ключа изменится, сработает триггер.
свойс тво равно	*объект*	Указывается путь к ключу объекта и ожидаемое значение. Если значение этого ключа примет ожидаемое значение, сработает триггер.
свойс тво не равно	*объект*	Указывается путь к ключу объекта и ожидаемое значение. Если значение этого ключа не примет ожидаемое значение, сработает триггер.

Условие	Тип поля	Описание

свойство элемента изменилось	*массив объектов*	Указывается путь к элементу массива. Если свойство этого элемента изменится, сработает триггер.
		
		При выборе пути к свойству элемента вы также выбираете тип данных, к которому приводить сравниваемое значение. Доступно сравнение таких типов данных, как **string**, **int**, **float**, и **bool**. Сравнение сложных типов данных (объект, массив) невозможно. В примере ниже вы не сможете отследить изменения в свойствах элементов `key_1` and `key_4`.
		`[`
		`{`
		`“key_1”: [1,2,3,4,5],`
		`“key_2”: 2`
		`},`
		`{`
		`“key_3”:“value”,`
		`“key_4”: [1,2,3,4,5]`
		`}`
		`]`
		Обратите внимание, выпадающий список содержит доступные пути к свойству поля, но индекс любого элемента по умолчанию равен 0 — `$[0].key_3`. Чтобы задать путь к нужному элементу, введите индекс вручную в формате jsonpath — `$[1].key_3`.

свойство элемента равно	*массив объектов*	Указывается путь к элементу массива и ожидаемое значение. Если свойство этого элемента примет ожидаемое значение, сработает триггер.
свойство элемента не равно	*массив объектов*	Указывается путь к элементу массива и ожидаемое значение. Если свойство этого элемента не примет ожидаемое значение, сработает триггер.
добавлен элемент	*массив объектов*	В массив добавляется новый элемент.
удален элемент	*массив объектов*	Из массива удаляется элемент.

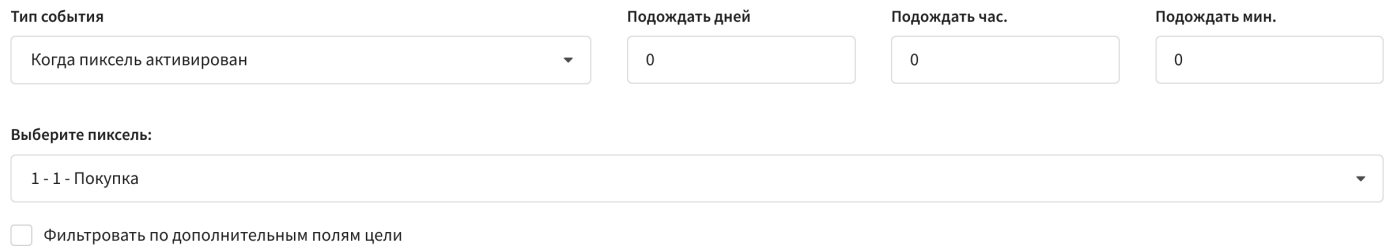
CAUTION

Такой тип триггера можно использовать со ссылками в шаблонах, на которые настроено обновление полей профиля. Обратите внимание, что если помимо обновления поля к **переходам по ссылке** привязаны другие триггеры (пункт выше), они также сработают.

Когда пиксель активирован

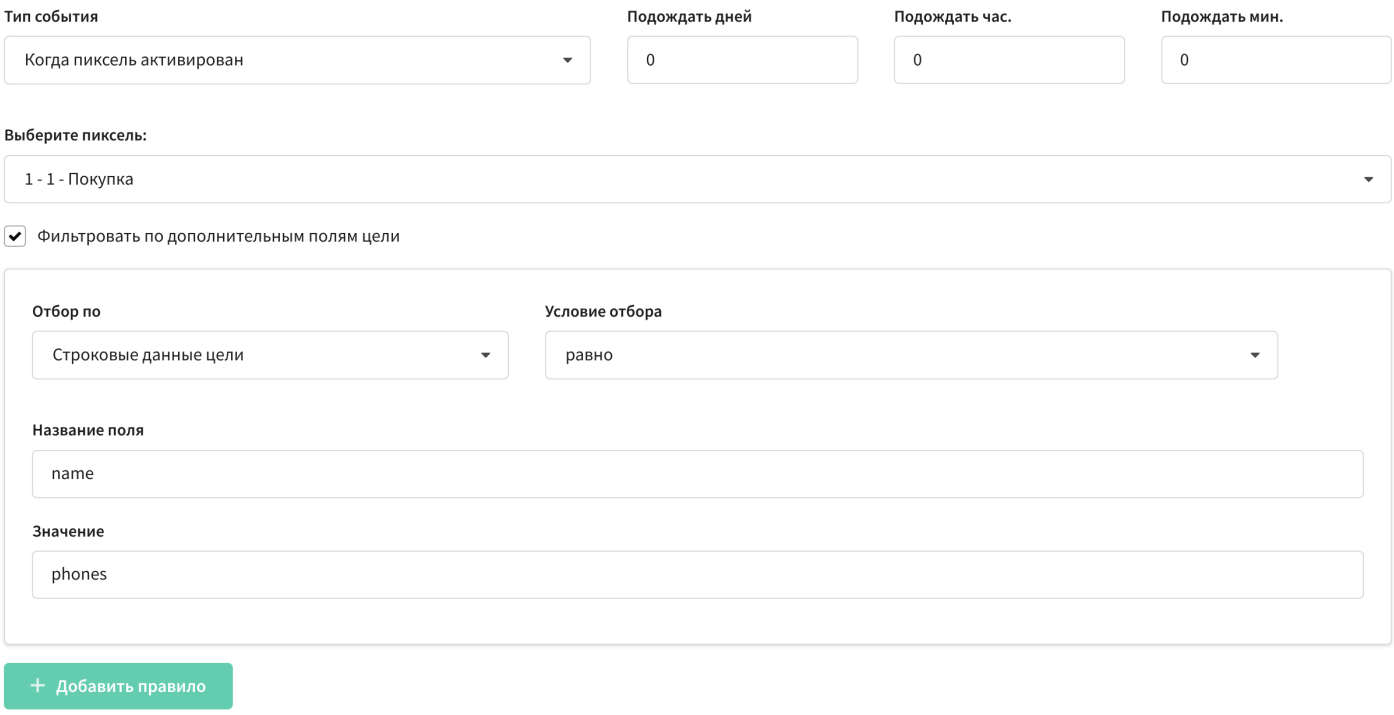
Пиксели Altcraft — это стек технологий для отслеживания действий клиентов на веб-сайтах. Вы можете разместить пиксель на своём веб-ресурсе или передать партнёрам и другим рефералам.

Триггерная рассылка может отслеживать, когда пиксель активируется для клиента с активными подписками на какой-либо канал, и отправлять сообщения сразу или по прошествии заданного времени:



При этом возможно отслеживать не только общее событие срабатывания пикселя. При срабатывании пиксель может регистрировать цели. В рамках события регистрации цели в платформу можно передавать значение цели и другую дополнительную информацию.

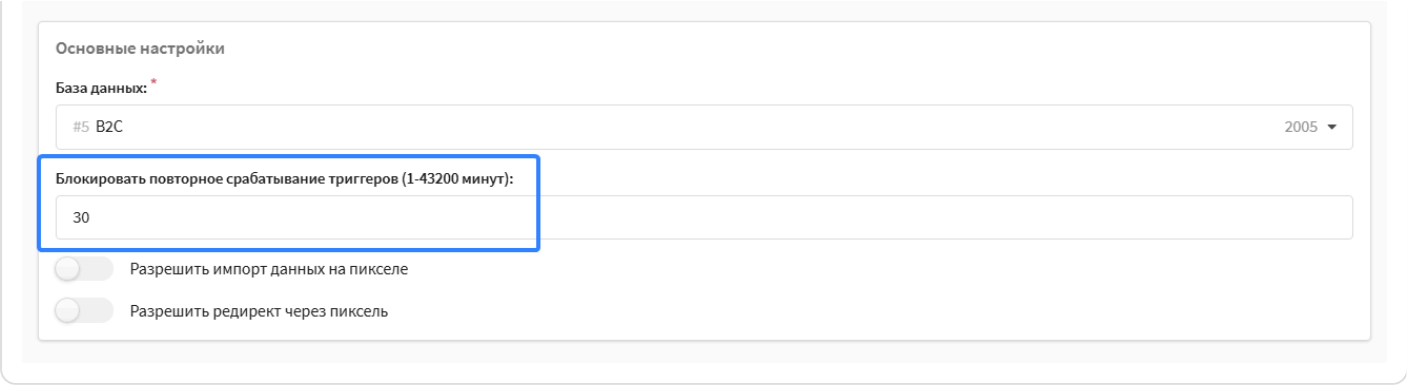
Вы можете указать правила отбора по цели и ее дополнительным полям, например, по значениям этих полей или по местоположению клиента. Это позволяет точнее настраивать рассылку под определенную группу клиентов или конкретные товары:



При отправке сообщений в шаблонах можно добавить **переменные** с данными о событии срабатывания пикселя, которое запустило рассылку. Узнать подробнее об этих переменных можно здесь.

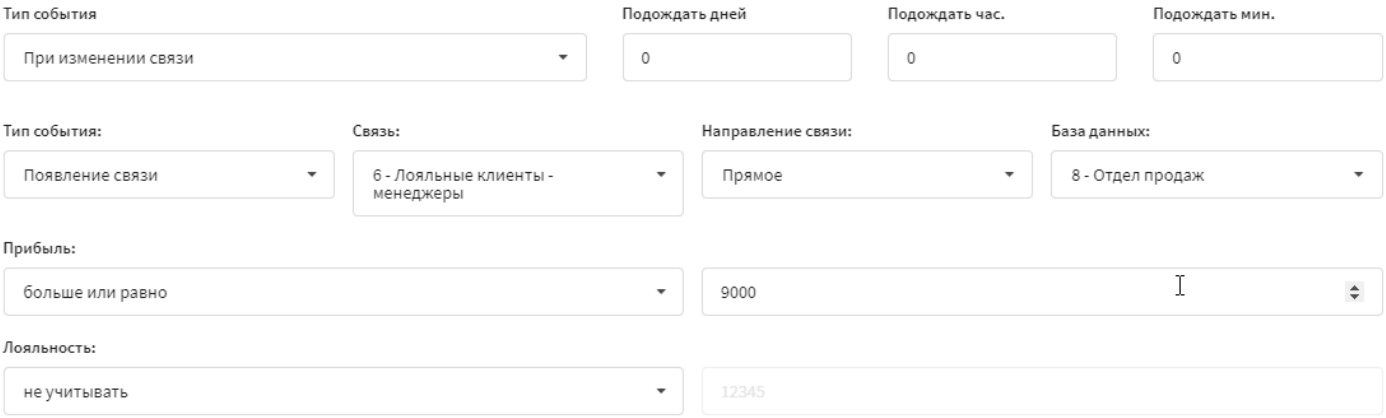
TIP

Если триггер не активируется в рассылках или в сценариях, хотя условия триггера выполняются, рекомендуется проверить **интервал срабатывания пикселя**.



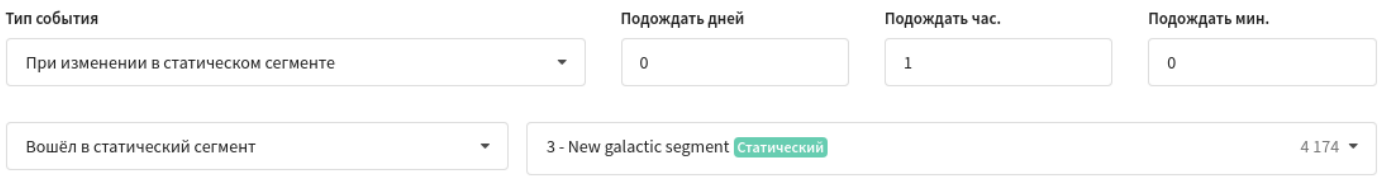
При изменении связи

Триггер может запускаться при изменении связи между профилями: как прямой, так и обратной. При этом может учитываться **появление**, **изменение** или **удаление** связи, а также достижение свойством связи заданного в триггере значения:



При изменении в статическом сегменте

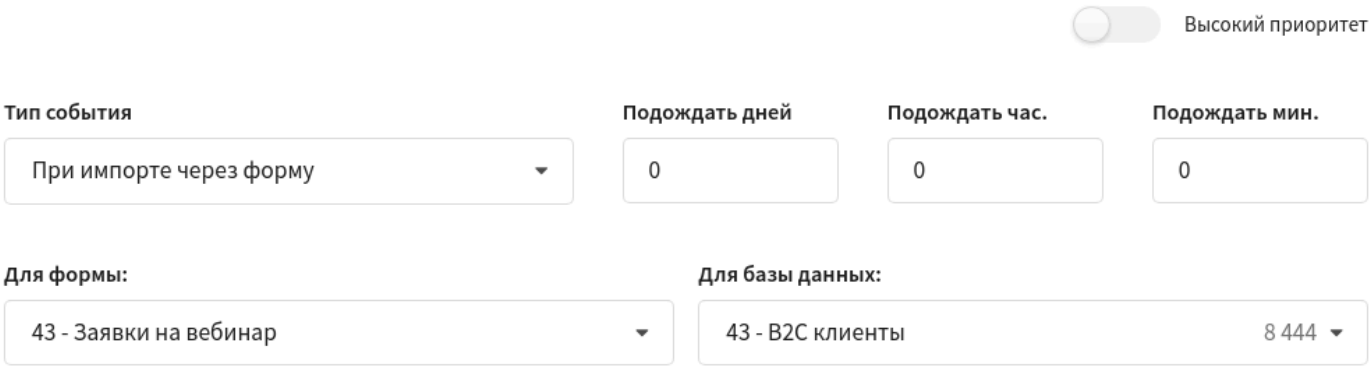
Триггер может запускаться при изменении в статическом сегменте: профиль может как войти в указанный статический сегмент, так и выйти из него.



При импорте через форму

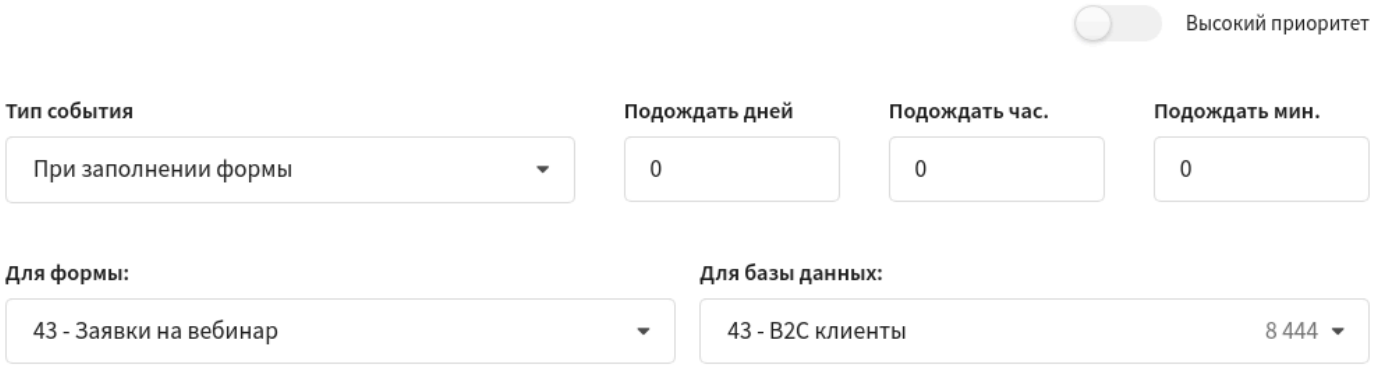
Триггер сработает, если профиль заполнил форму и по этой причине был добавлен в базу данных. Для этого необходимо при настройке формы активировать опцию “**Импорт профилей**” на вкладке “**Действия**”.

Выберите форму и базу профилей:



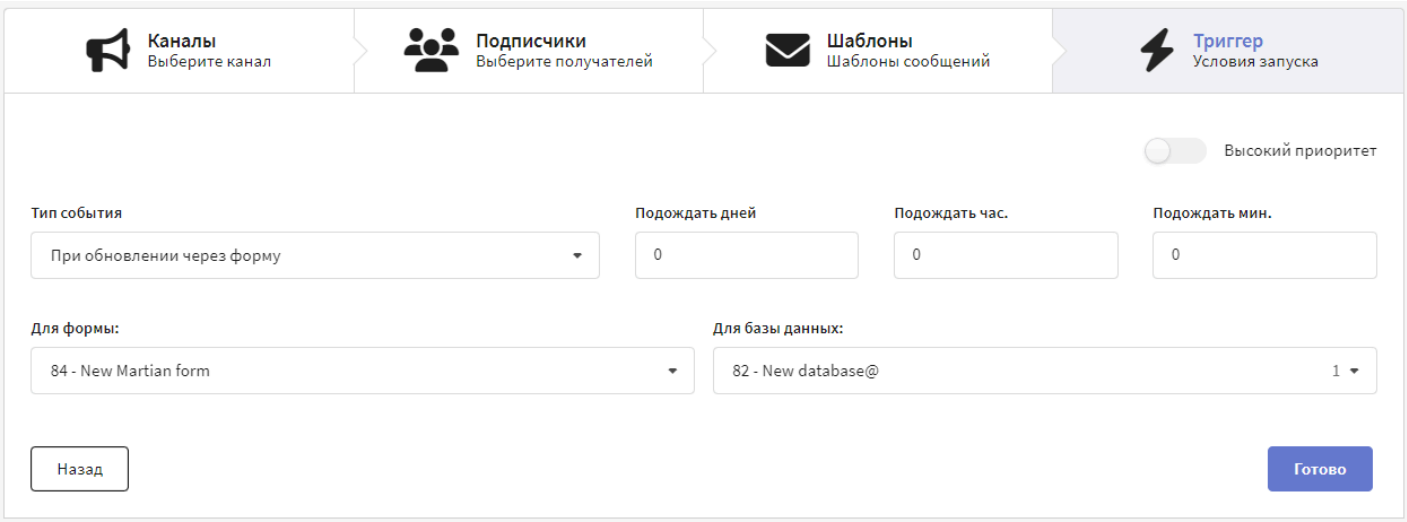
При заполнении формы

Триггер активируется, только если профиль заполнил форму при переходе по ссылке из письма. Выберите форму и базу, где находятся профили, которые будут заполнять форму:



При обновлении через форму

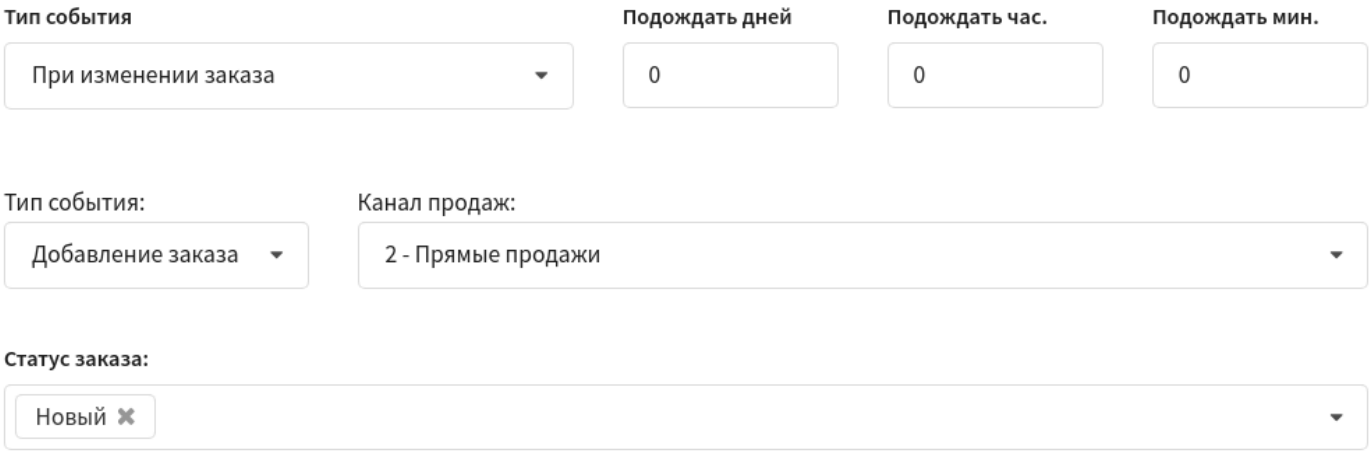
Триггер приводится в действие при изменении данных профиля, который уже есть в базе. Обновление данных осуществляется в результате заполнения формы. Выберите предварительно настроенную форму. Если вы хотите отслеживать изменения данных профилей только в определенной базе, укажите базу.



При изменении заказа

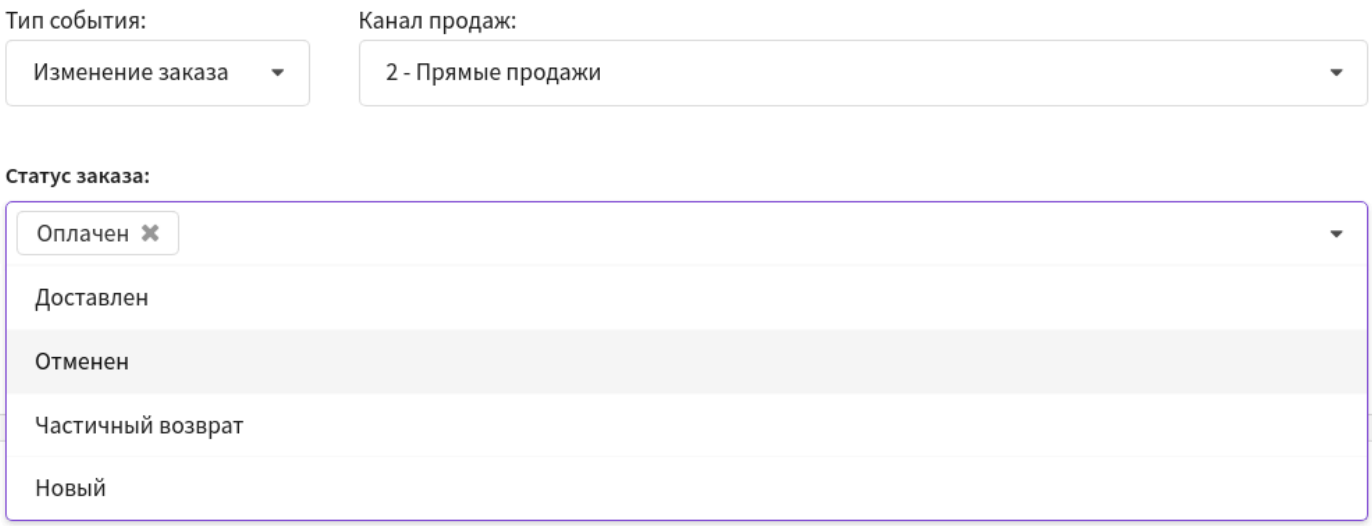
Триггер сработает, если заказ в маркете был изменен. Выберите канал продаж и тип события. Доступно два типа:

- **добавление заказа** (новый заказ был добавлен в маркет)
- **изменение заказа** (изменился статус существующего заказа)



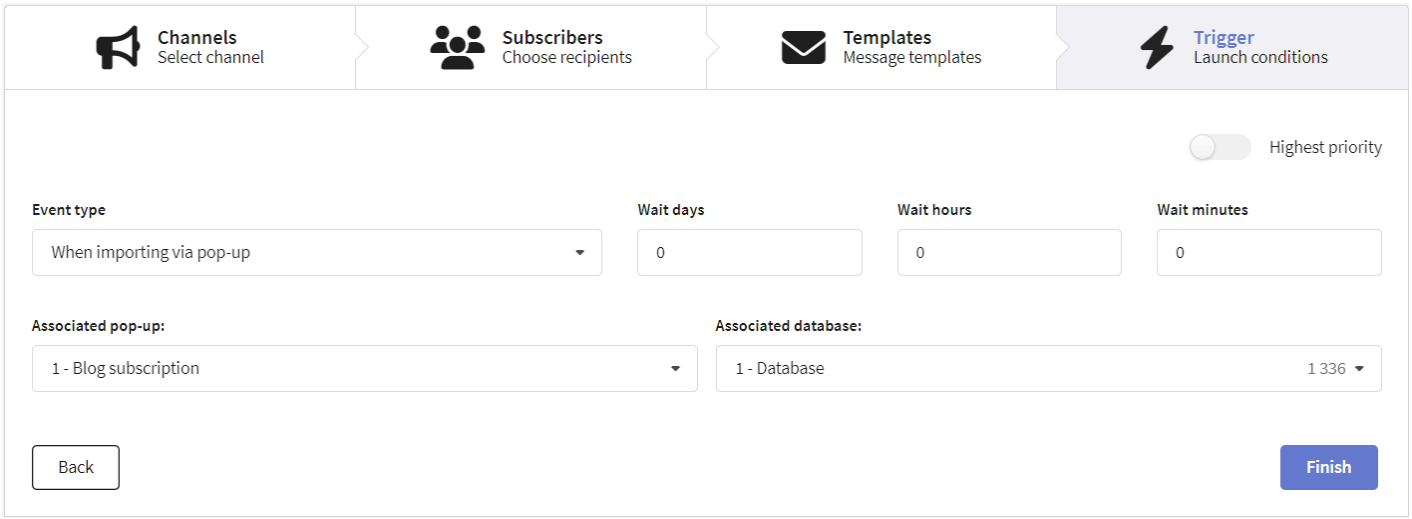
Возможные статусы заказа:

- новый
- доставлен
- оплачен
- отменен
- частичный возврат
- в доставке
- в процессе



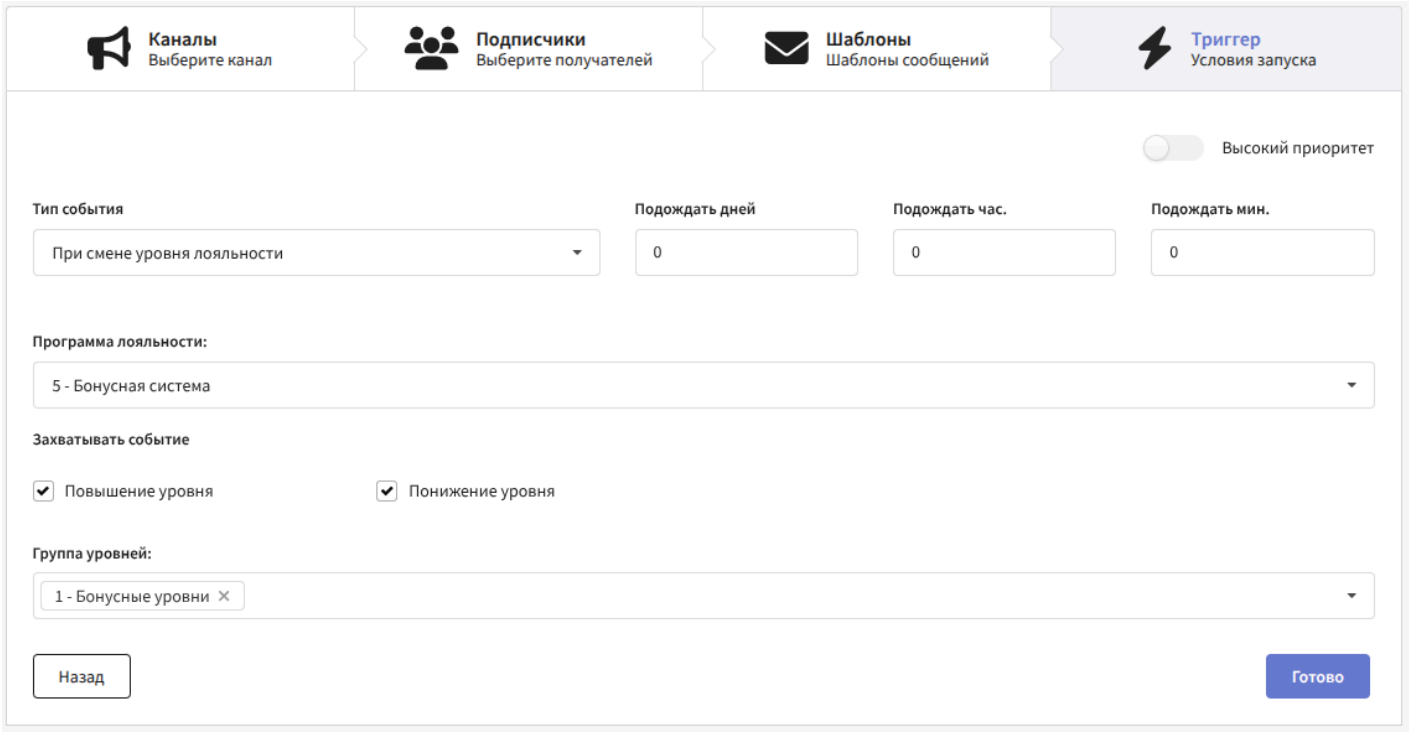
При импорте через попап

Триггер сработает, если данные профиля попадут в базу в результате заполнения и отправки формы в попапе. Выберите **попап**, заполнение которого необходимо отслеживать.



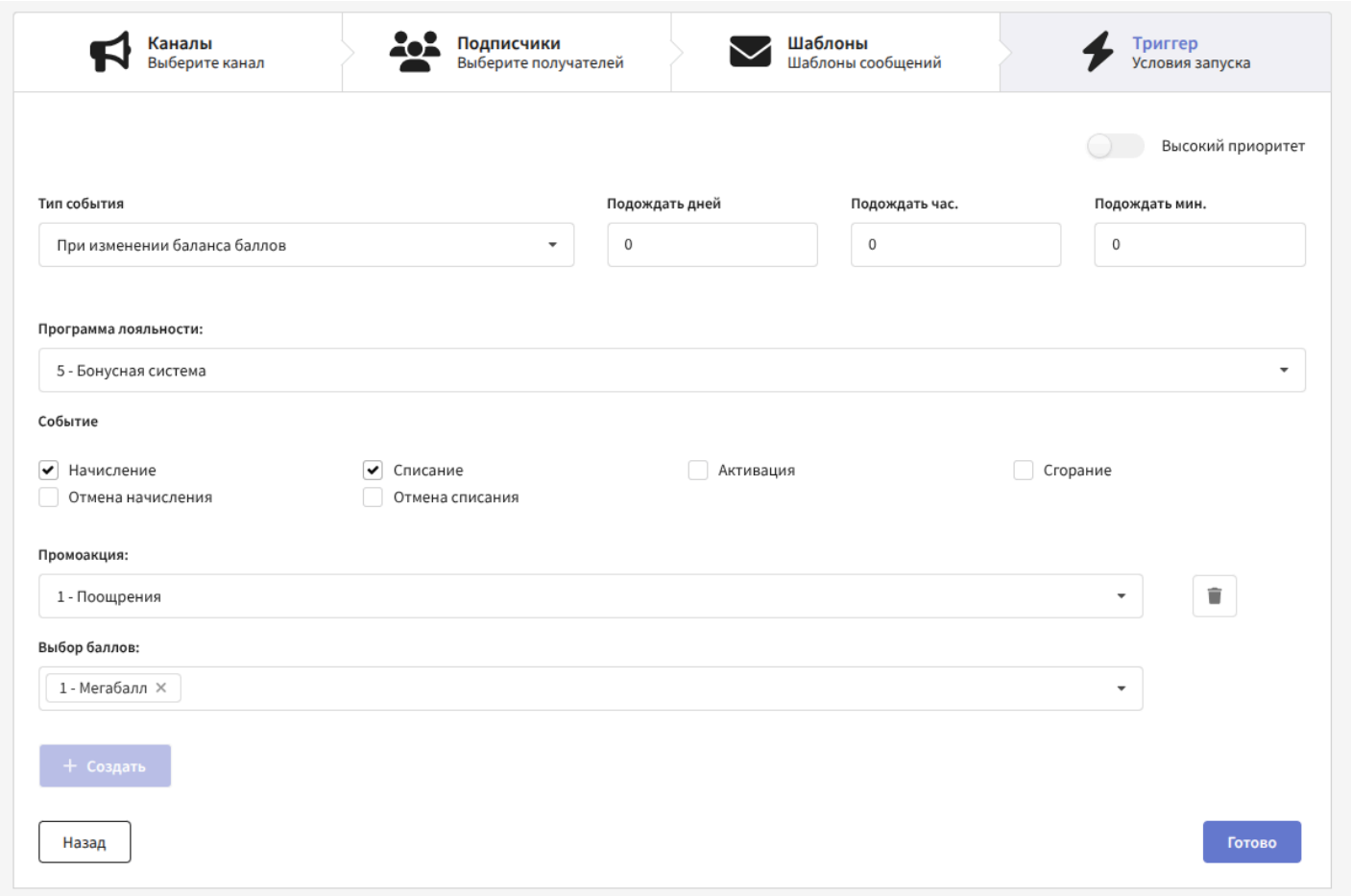
При смене уровня лояльности

Триггер сработает, если профиль перейдет на другой уровень в рамках программы лояльности.



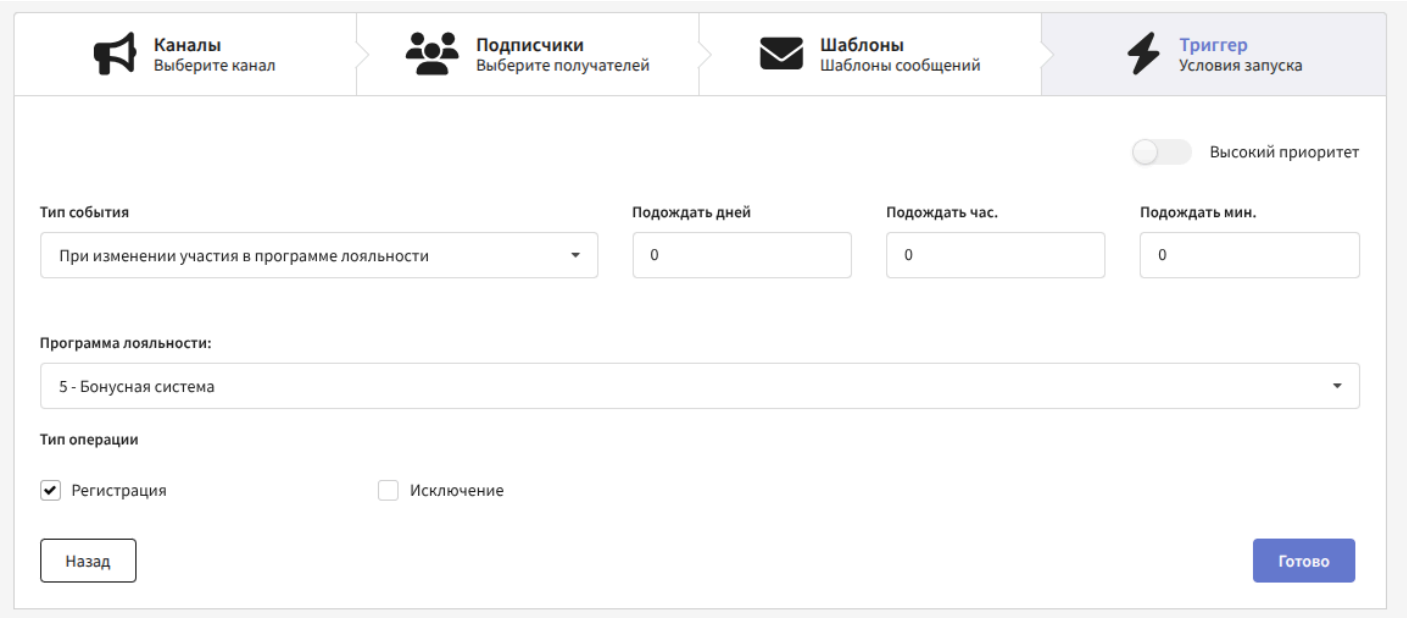
При изменении баланса баллов

Триггер сработает, если у профиля изменится количество бонусных баллов в программе лояльности.



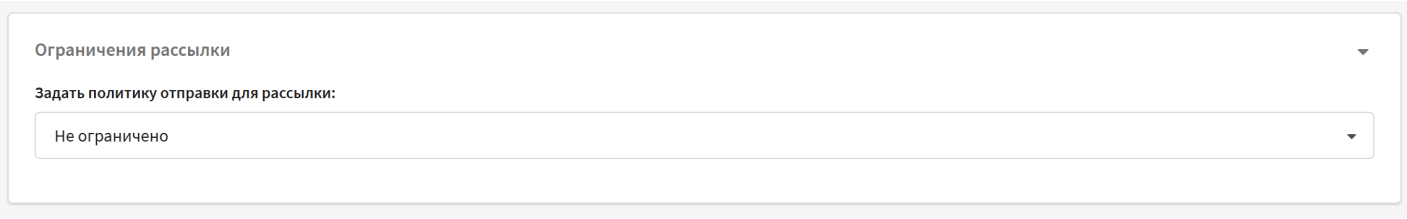
При изменении участия в программе лояльности

Триггер сработает, если профиль будет зарегистрирован или исключен из программы лояльности.



Ограничения рассылки

Так как у триггерных рассылок нет фиксированной аудитории, отправку возможно ограничить только **Политикой отправки**: для одного пользователя за период времени.



Smart settings

Блок **Smart-настройки** объединяет следующие дополнительные настройки рассылки:

- **Виртуальный сендер** — позволяет выбрать для рассылки виртуальный сендер (набор правил для отправки сообщений с реальных сендеров платформы).
- **Свои трекинг-домены** — позволяет отслеживать действия с сообщениями рассылки, используя дополнительные трекинг-серверы. Каждый домен указывается на отдельной строке.
- **Список email адресов для скрытой копии** — BCC для отправляемых сообщений. Каждый адрес указывается на отдельной строке.
- **Не проверять контактные данные на уникальность** — отключает проверку на уникальность получателей рассылки по данным подписки.

CAUTION

Если вы не уверены, какие настройки необходимо выбрать, проконсультируйтесь с администратором платформы в вашей организации или со специалистами поддержки Altcraft.

Наблюдатели за рассылкой

Помимо этого, в умных настройках можно установить **наблюдателей за рассылкой**. Наблюдатели будут получать такие же сообщения, как и получатели рассылки. Для этого вам нужна будет **база данных наблюдателей**. Она создаётся так же, как и обычная база данных, только в Основных настройках необходимо отметить соответствующую опцию.

В Smart-настройках выберите **БД наблюдателей** и лимит отправки сообщений для каждого запуска рассылки:

Наблюдатели за рассылкой

Наблюдателям будут отправлены письма каждый запуск рассылки если лимит не установлен, иначе будет выбрана новая порция наблюдателей в пределах лимита.

Письма будут отправлены на основной email профиля наблюдателя, подписки не учитываются.

БД наблюдателей:

2 - Mailing watchers database 1000

Лимит:

100

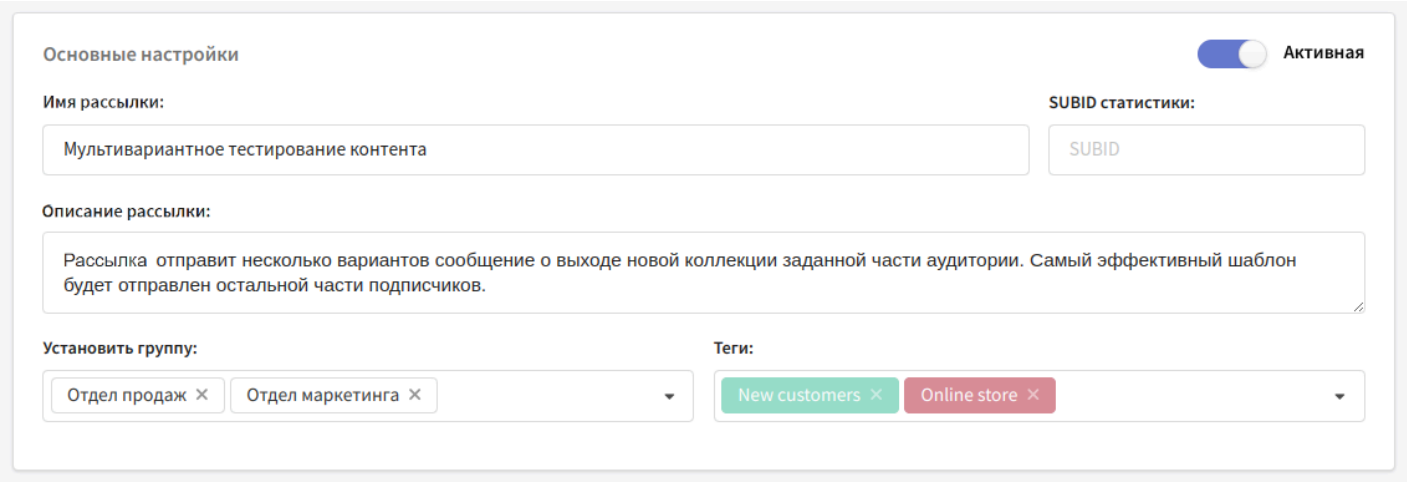
Мультивариантный тест (A/B/n)

Мультивариантный тест — рассылка, в которой несколько (до 8) шаблонов сообщений отправляется части аудитории для сравнения их эффективности. После чего определяется **шаблон-победитель**, который отправляется остальной части аудитории. Рассылает по одному сообщению всем подписчикам из своего источника данных: тестовой части и основной - по заданному расписанию. Когда каждому из получателей будет отправлено одно сообщение, рассылка завершится так же, как броадкаст.

	Чтобы создать новый мультивариантный тест, в главном меню выберите **Рассылки** → **Мультивариантные тесты** и нажмите кнопку **+ Создать**. Или воспользуйтесь календарём рассылок.

Основные настройки

В основных настройках укажите название и описание новой рассылки, при необходимости установите группу доступа, **теги** и **атрибуты** для дополнительной группировки. Здесь же вы можете указать **SUBID статистики** для рассылок. В верхнем правом углу находится переключатель, активирующий рассылку:



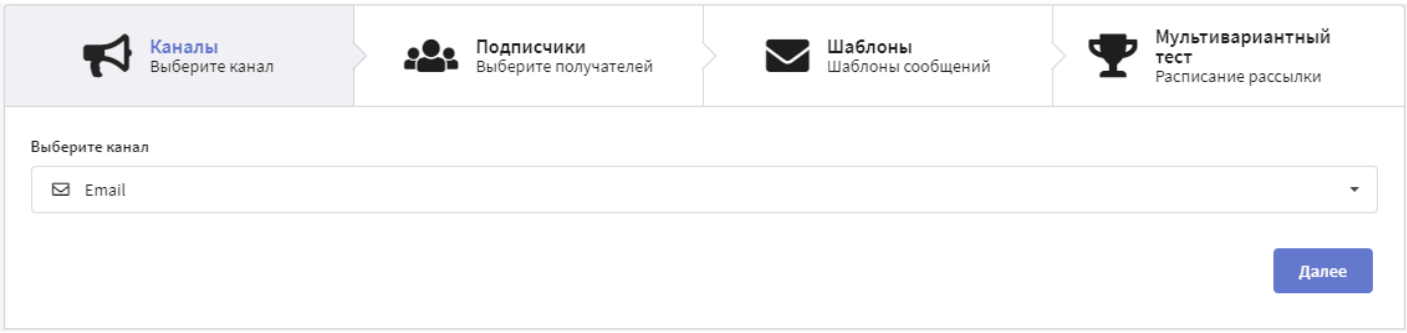
Настройки рассылки

Настройки рассылки включают:

- **канал**, по которому она будет отправлена,
- **получателей** сообщений,
- контент **нескольких шаблонов**,
- размер **аудитории для теста**,
- режим определения **шаблона-победителя**.

Канал отправки

В **первой вкладке** выберите канал связи, по которому хотите отправить рассылку. Канал отправки должен быть доступен хотя бы в одном ресурсе подписок. Ресурс выбирается в следующей вкладке.



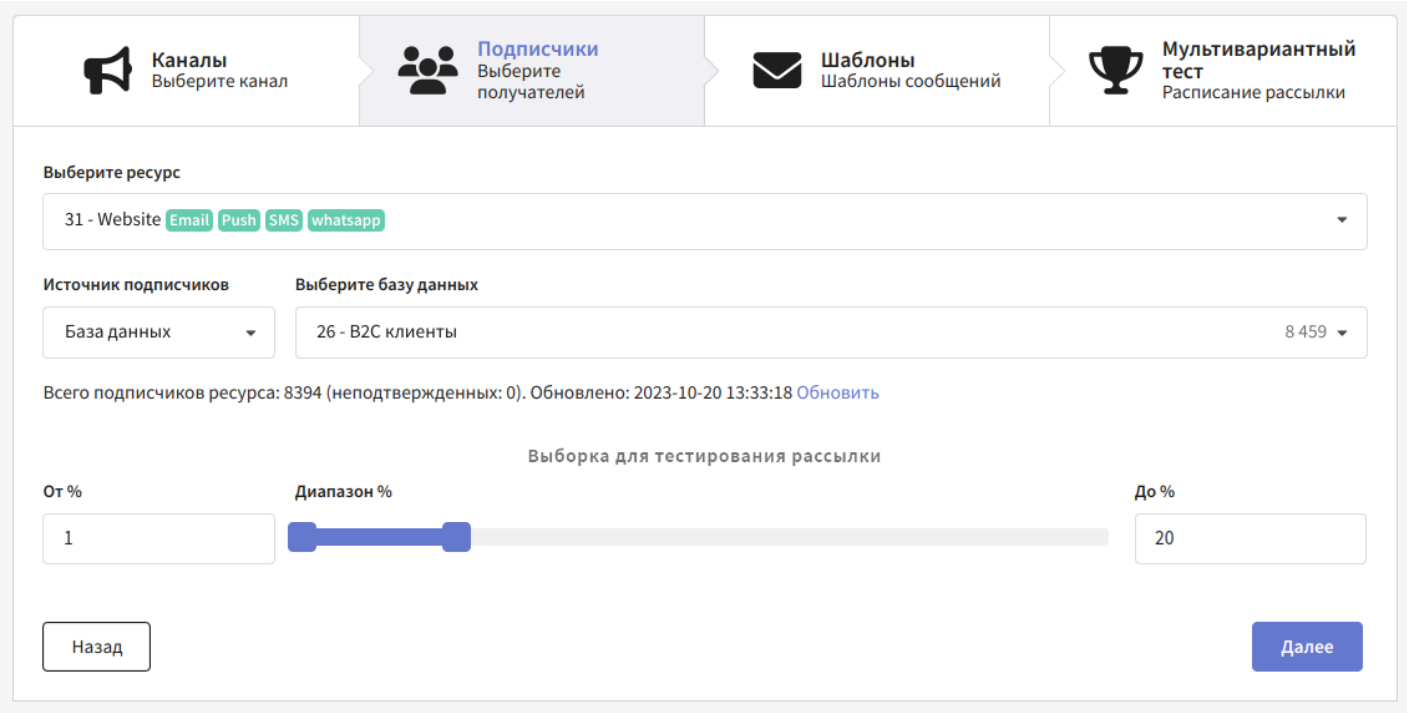
Получатели

Рассылки отправляются в рамках определённого ресурса: это может быть отдельный сайт, продукт, акция, или другое направление ваших коммуникаций с клиентами. Выберите ресурс из выпадающего меню. В ресурсе должен быть настроен выбранный на предыдущем шаге канал.

Далее выберите **источник подписчиков**. Получателями рассылки могут быть профили определенной базы данных или профили, состоящие в сегменте.

> **TIP**
>
> При отправке сообщений рассылка использует подписки клиентов на выбранный ресурс. Убедитесь, что получатели в вашем источнике подписаны на него!

В разделе **Выборка для тестирования рассылки** установите часть аудитории, которой будет отправлено несколько вариантов контента для сравнения эффективности шаблонов.



Шаблоны

В этом блоке выберите **контент**, который хотите использовать для тестирования. Шаблоны сообщений объединяют контент для отправки по всем доступным каналам.

> **TIP**

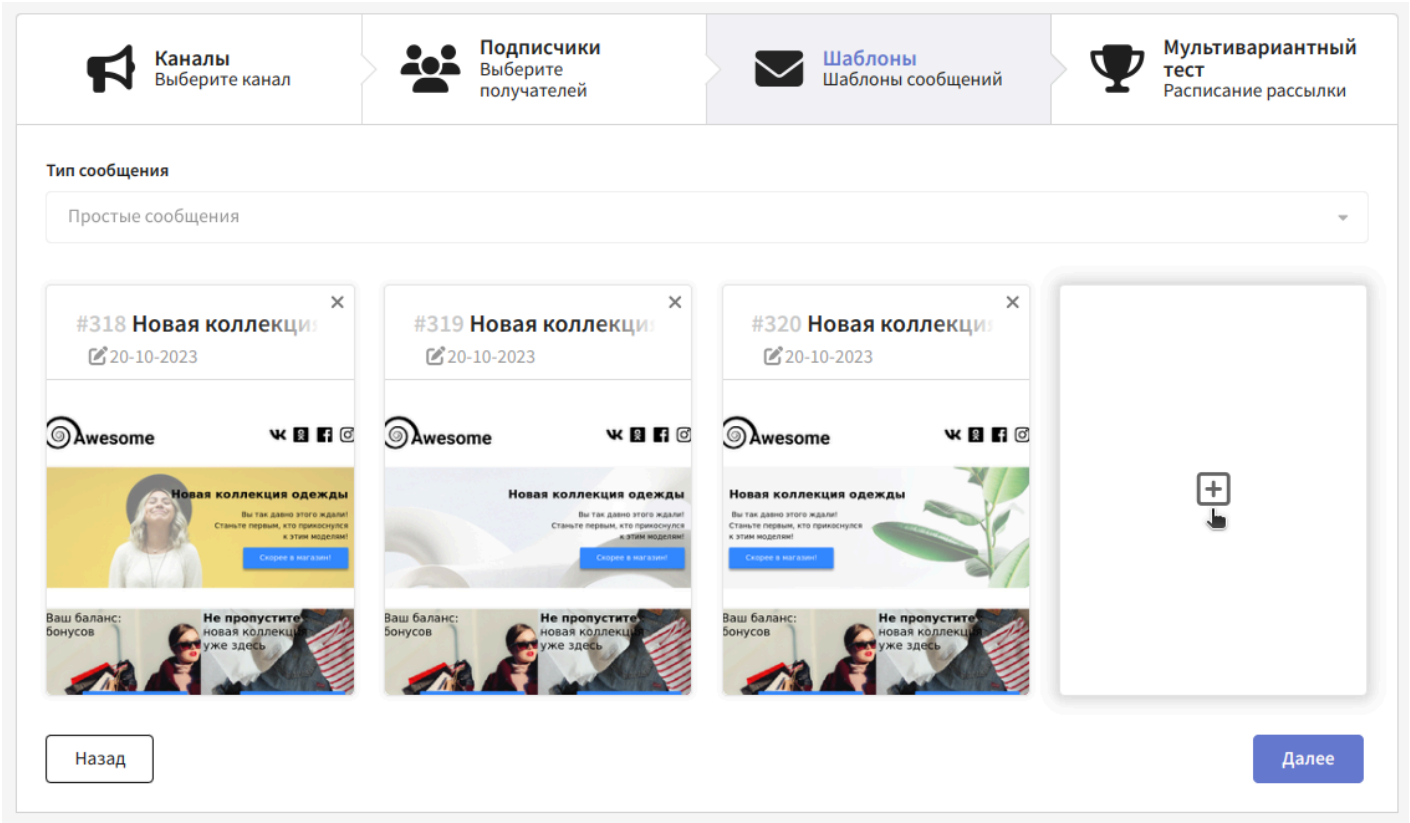
Убедитесь, что контент для выбранного в рассылке канала присутствует в шаблоне!

Нажмите на кнопку **Добавить**

и выберите подготовленные шаблоны.

MVT тест поддерживает **до 8 шаблонов** сообщений. Рассылка отправит каждому получателю по одному сообщению из выбранных в случайном порядке (в равных пропорциях).

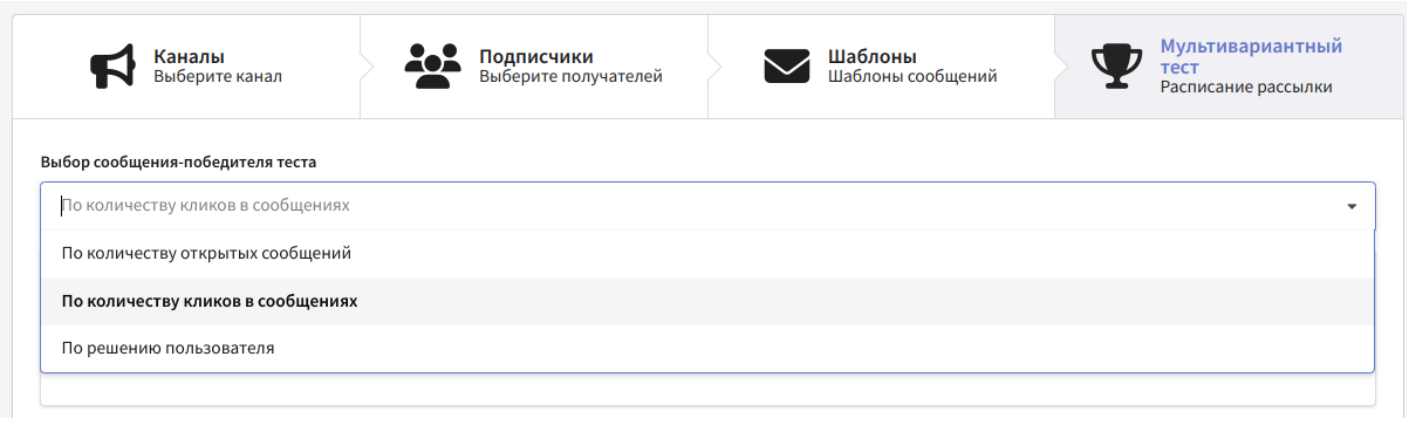
По клику на миниатюру шаблона откроется **предпросмотр**. При нажатии на название шаблона он откроется в новой вкладке для редактирования:



Настройка тестирования

На вкладке **Мультивариантный тест** выберите подходящий вам режим определения шаблона-победителя:

- По количеству открытых сообщений;
- По количеству кликов в сообщениях;
- По решению пользователя – вручную по данным аналитических отчётов.

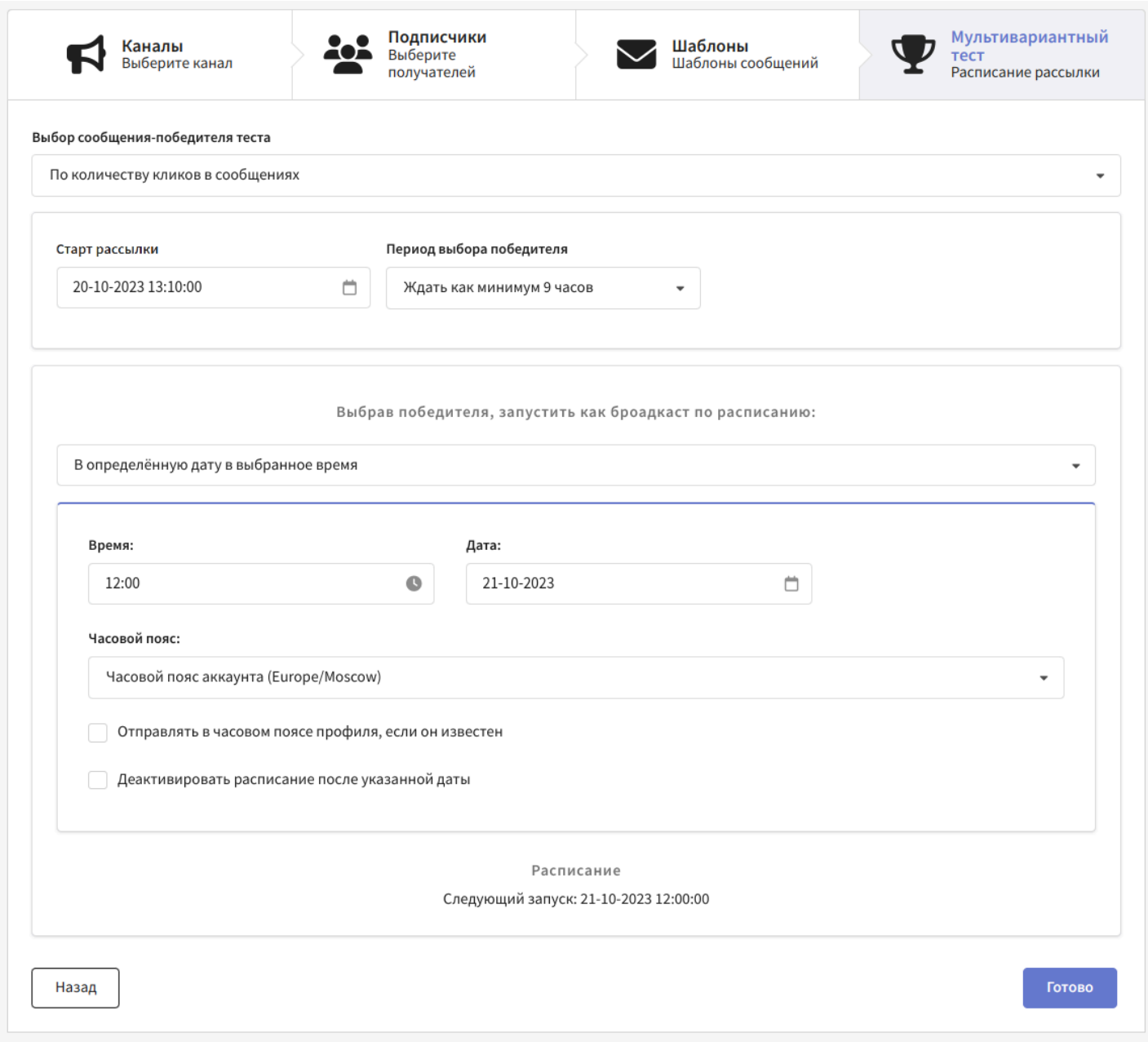


Выбор шаблона-победителя по открытиям/кликам

При автоматическом режиме выбора победителя установите время **Старта рассылки**. Старт рассылки — это время и дата запуска тестовой части рассылки.

Рассылка после запуска будет автоматически собирать статистику по **общему количеству** открытий или переходов по ссылкам в шаблонах. Время для сбора аналитики настраивается параметром **Период выбора победителя**.

После того как рассылка соберёт статистику открытий или кликов, наиболее эффективный по выбранному показателю шаблон сообщения будет отправлен основной части аудитории по заданному **Расписанию рассылки**:



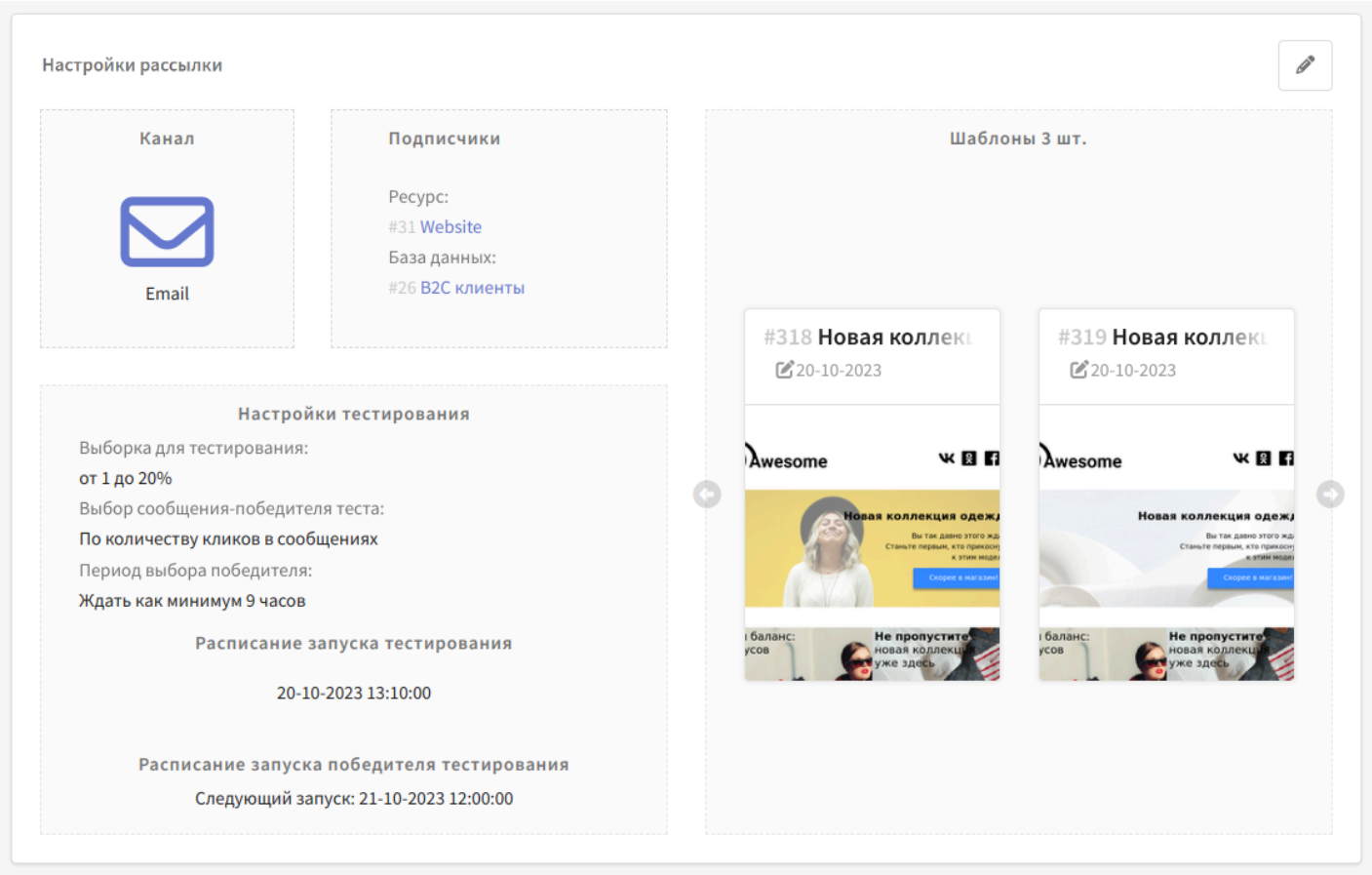
TIP

Расписание рассылок подробно описано в отдельной статье.

CAUTION

Если при настройке расписания вы выбираете часовой пояс, отличный от пояса аккаунта, то по этому часовому поясу отправится только рассылка с шаблоном-победителем. Тестовые сообщения отправятся по часовому поясу аккаунта.

Чтобы завершить настройку, нажмите **Готово**. Выбранные вами параметры будут отображаться в блоке **Настройки рассылки**. Кнопкой в правом верхнем углу блока вы можете **отредактировать** настройки:



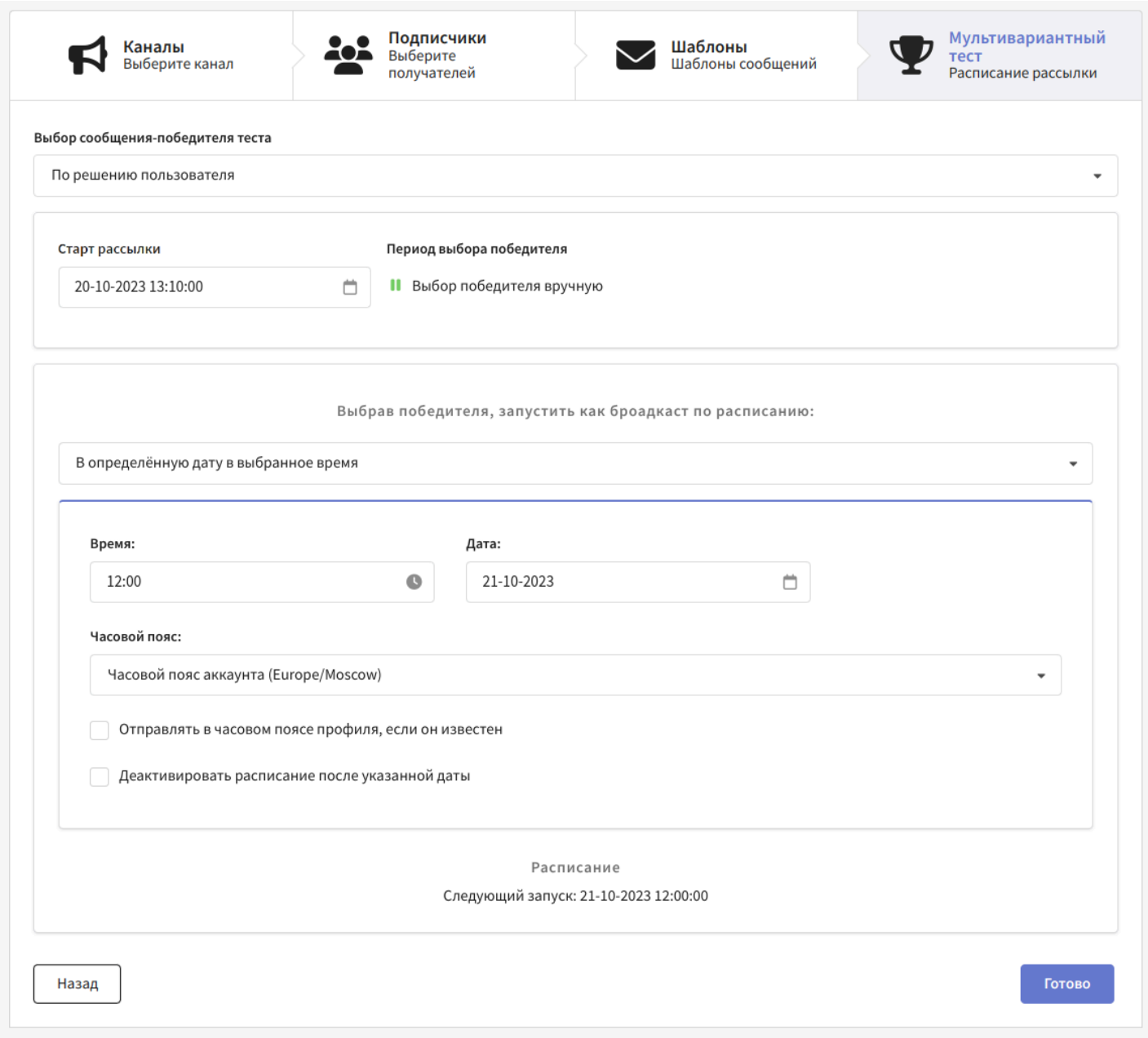
Сохраните рассылку, чтобы активировать её.

Выбор шаблона-победителя вручную

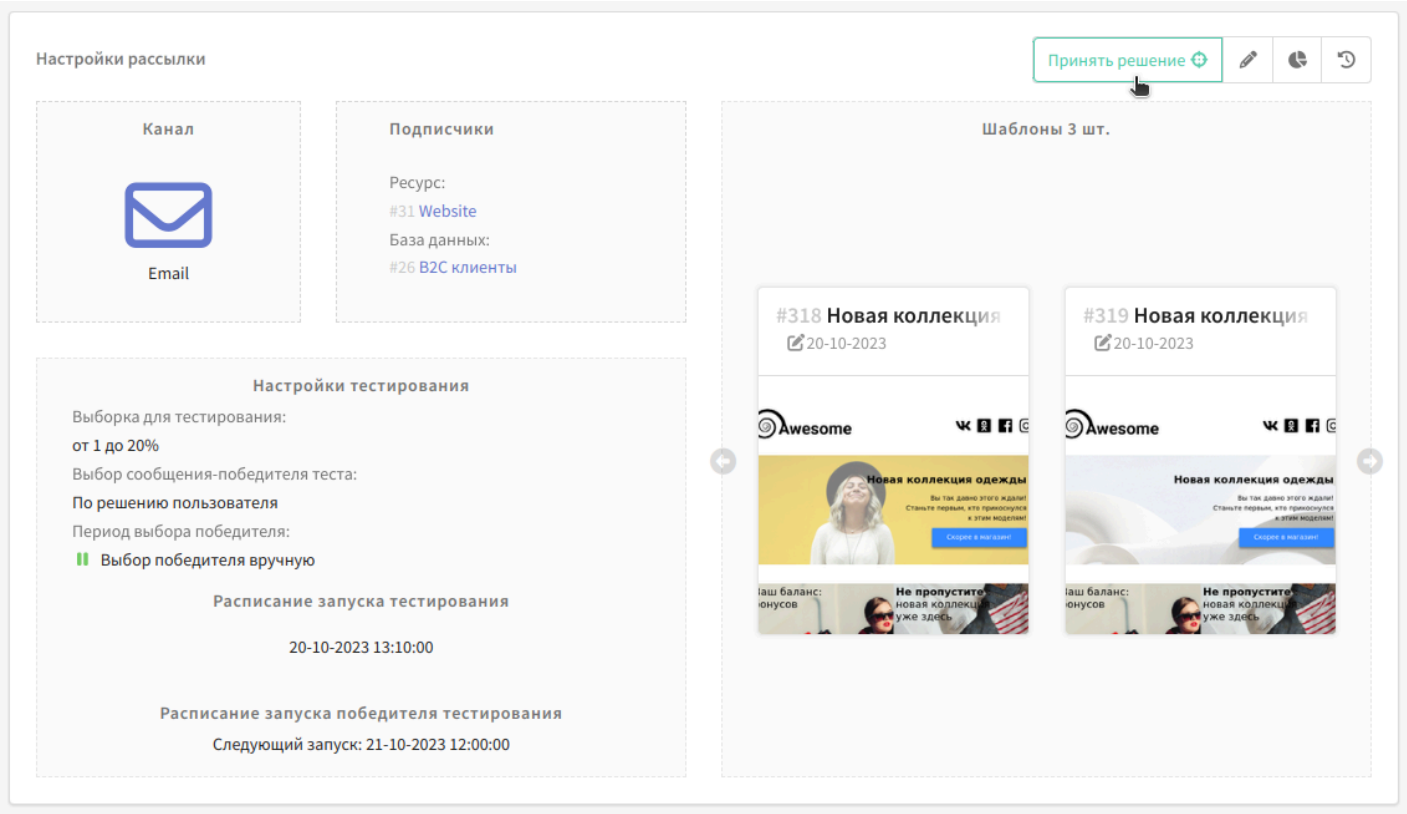
При режиме выбора победителя **По решению пользователя** установите время **Старта рассылки**. Старт рассылки — это время и дата запуска тестовой части рассылки.

Затем задайте **Расписание рассылки.** По этому расписанию шаблон-победитель будет отправлен основной части аудитории.

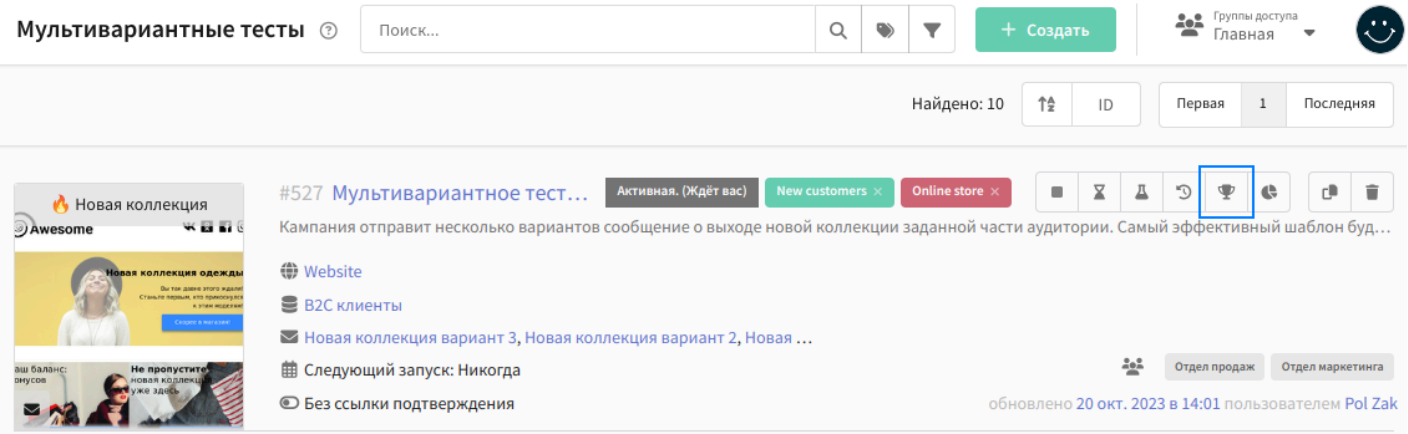
Сохраните рассылку, чтобы активировать её. После того как рассылка отправит тестовые сообщения, она перейдёт в статус **Активная (Ждёт вас)**.



На этом этапе вам необходимо проанализировать показатели каждого шаблона и выбрать шаблон-победитель. Для этого в окне редактирования рассылки нажмите на кнопку **“Принять решение”**:



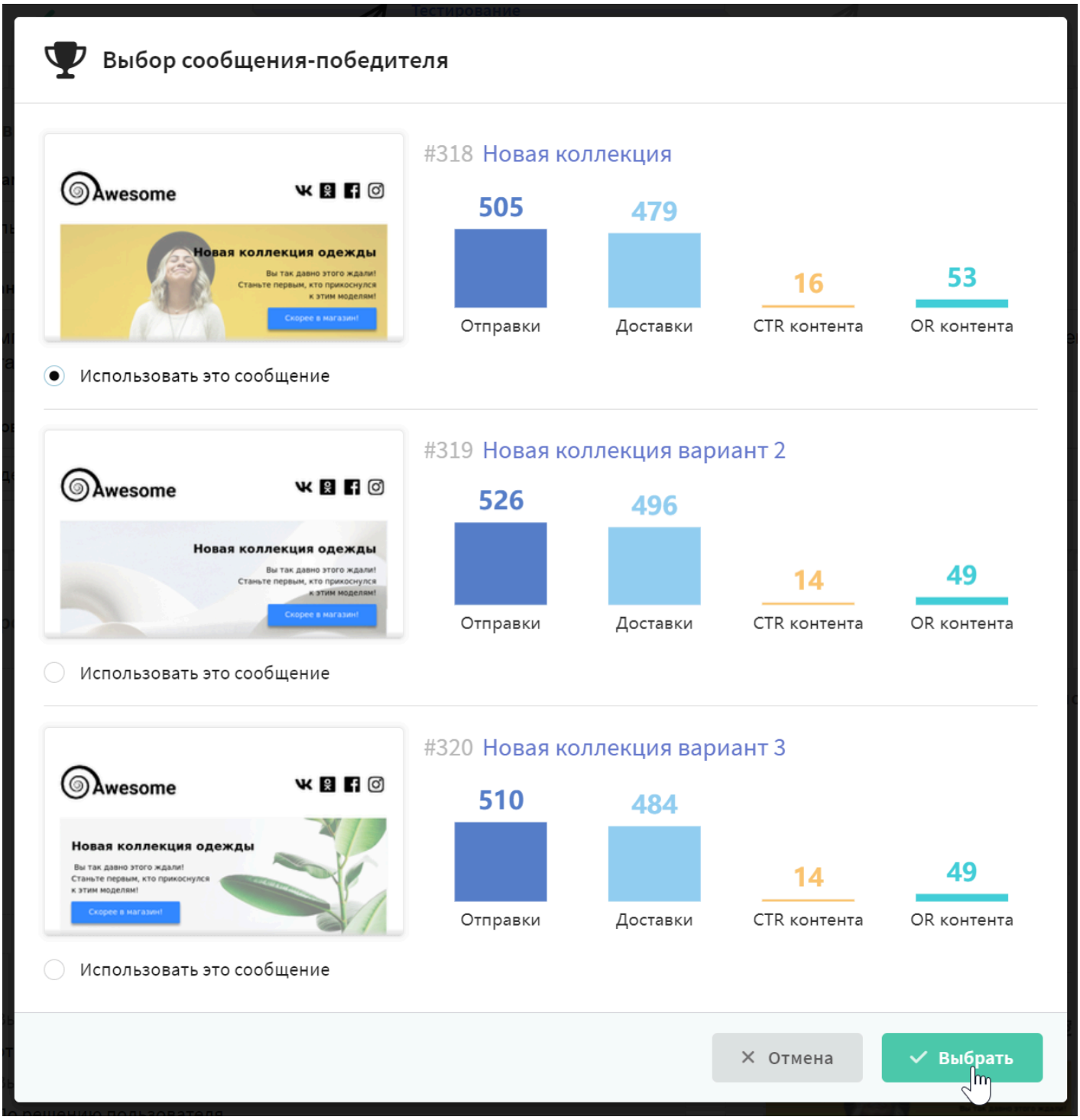
Также шаблон-победитель можно выбрать в окне со списком рассылок, нажав на соответствующую кнопку в контекстном меню:



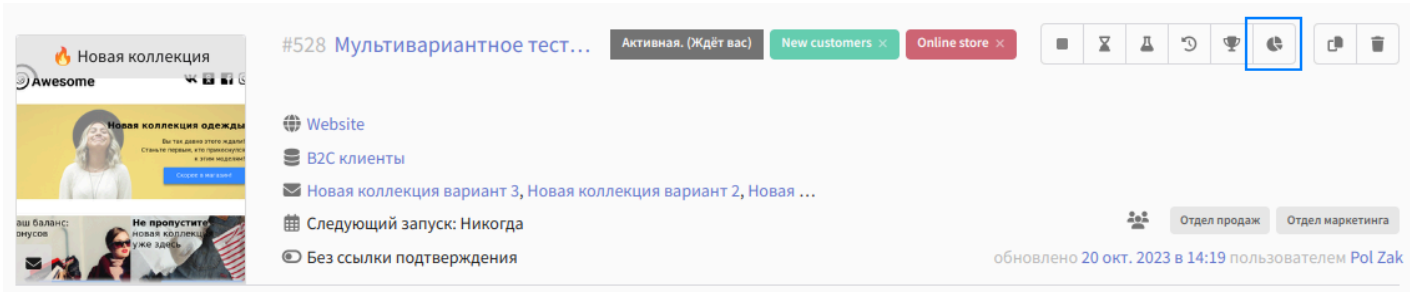
Откроется окно со статистикой по каждому шаблону:

- **Отправки** — количество отправленных сообщений.
- **Доставки** — количество доставленных сообщений.
- **CTR контента** — процент кликов в сообщении. Рассчитывается по формуле: (Количество кликов / Количество доставленных сообщений) * 100%.
- **OR контента** — процент открытий сообщения. Рассчитывается по формуле: (Количество открытий / Количество доставленных сообщений) * 100%.

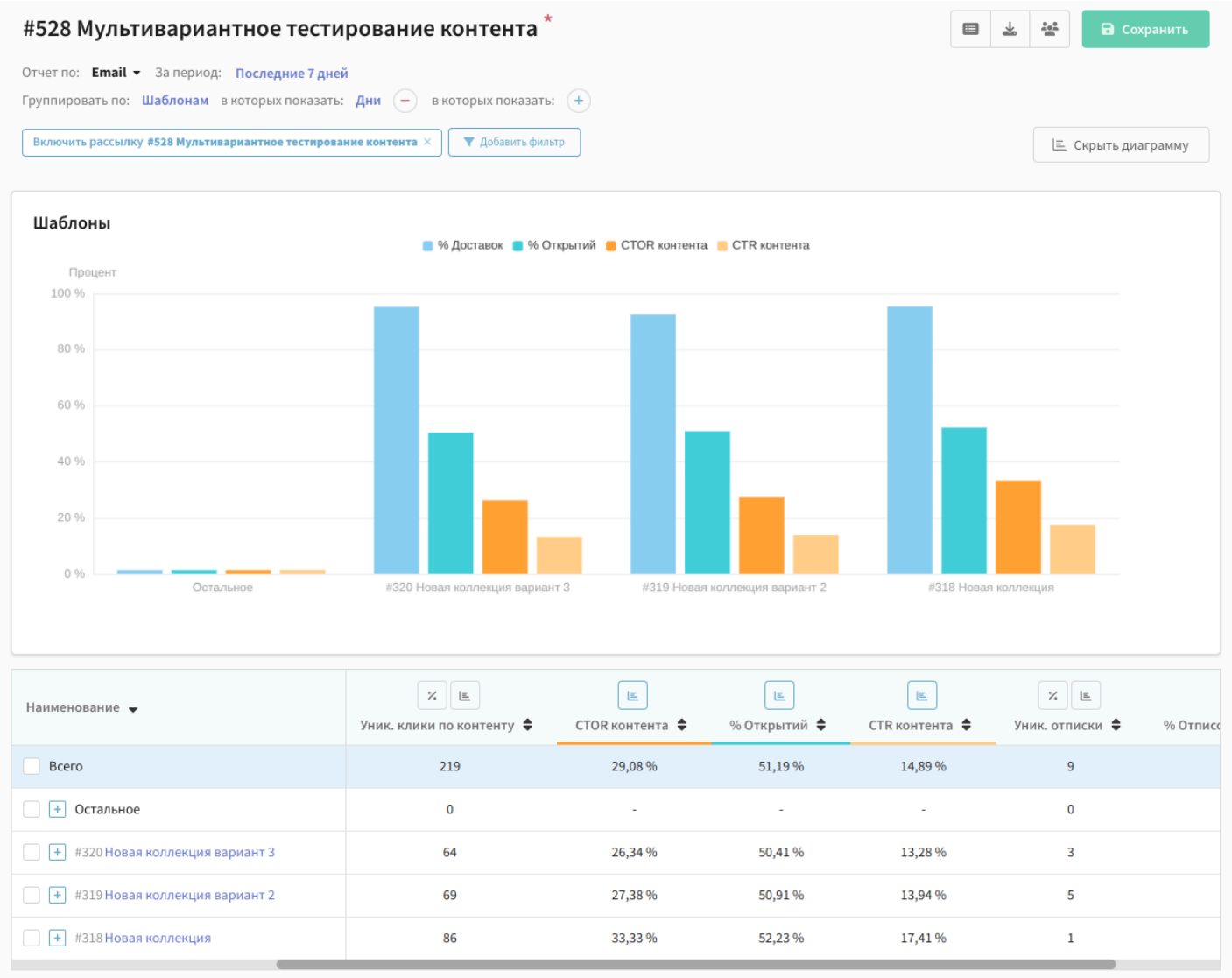
Выберите шаблон, который посчитаете наиболее эффективным, и нажмите кнопку **“Выбрать”**. Этот шаблон будет отправлен основной части аудитории по заданному расписанию.



Если для выбора шаблона-победителя вам недостаточно предоставленных данных, вы можете воспользоваться **Отчетом по каналам**. Чтобы открыть отчёт, нажмите на иконку аналитики:



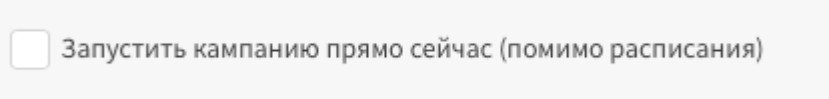
В отчёте выберите необходимые для оценки эффективности параметры. Это могут быть абсолютные или относительные значения (такие как Open Rate или CTR/CTOR). Можно оценить и негативную реакцию по количеству отписок или жалоб:



Мгновенный запуск рассылки (помимо расписания)

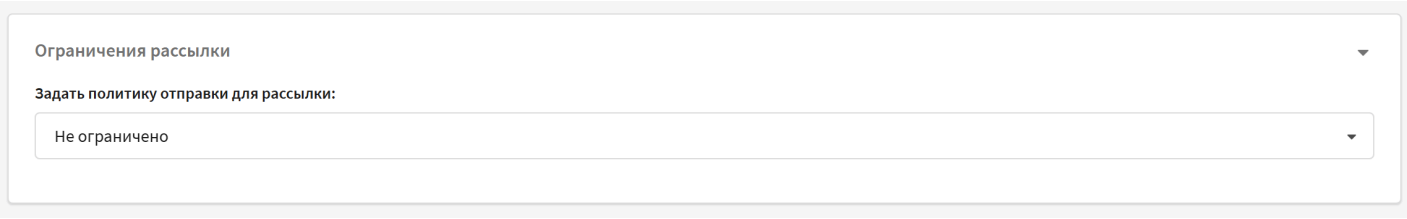
Чтобы запустить мультивариантный тест сразу же после создания, в самом низу страницы отметьте соответствующую опцию.

- Если лимиты не установлены — сообщения отправятся всем получателям.
- Если рассылка разбита на несколько частей — произойдёт запуск помимо расписания.



Политика отправки

Политика отправки ограничивает коммуникацию с одним подписчиком. Настраивается для рассылки в блоке **Ограничения рассылки**. По умолчанию рассылка будет использовать политику, выбранную глобально в аккаунте или в Ресурсе, от имени которого отправляется броадкаст.



Smart settings

Блок **Smart-настройки** объединяет следующие дополнительные настройки рассылки:

- **Виртуальный сендер** — позволяет выбрать для рассылки виртуальный сендер (набор правил для отправки сообщений с реальных сендеров платформы).
- **Свои трекинг-домены** — позволяет отслеживать действия с сообщениями рассылки, используя дополнительные трекинг-серверы. Каждый домен указывается на отдельной строке.
- **Список email адресов для скрытой копии** — BCC для отправляемых сообщений. Каждый адрес указывается на отдельной строке.
- **Не проверять контактные данные на уникальность** — отключает проверку на уникальность получателей рассылки по данным подписки.

CAUTION

Если вы не уверены, какие настройки необходимо выбрать, проконсультируйтесь с администратором платформы в вашей организации или со специалистами поддержки Altcraft.

Наблюдатели за рассылкой

Помимо этого, в умных настройках можно установить **наблюдателей за рассылкой**. Наблюдатели будут получать такие же сообщения, как и получатели рассылки. Для этого вам нужна будет **база данных наблюдателей**. Она создаётся так же, как и обычная база данных, только в Основных настройках необходимо отметить соответствующую опцию.

В Smart-настройках выберите **БД наблюдателей** и лимит отправки сообщений для каждого запуска рассылки:

Наблюдатели за рассылкой

Наблюдателям будут отправлены письма каждый запуск рассылки если лимит не установлен, иначе будет выбрана новая порция наблюдателей в пределах лимита.

Письма будут отправлены на основной email профиля наблюдателя, подписки не учитываются.

БД наблюдателей:

2 - Mailing watchers database 1000

Лимит:

100

Тестирование расылок

TIP

Видео-гайд по тестовой отправке сообщения доступен в **центре поддержки**.

Описание

Тестирование рассылок необходимо проводить перед каждой реальной отправкой. Это позволяет убедиться, что:

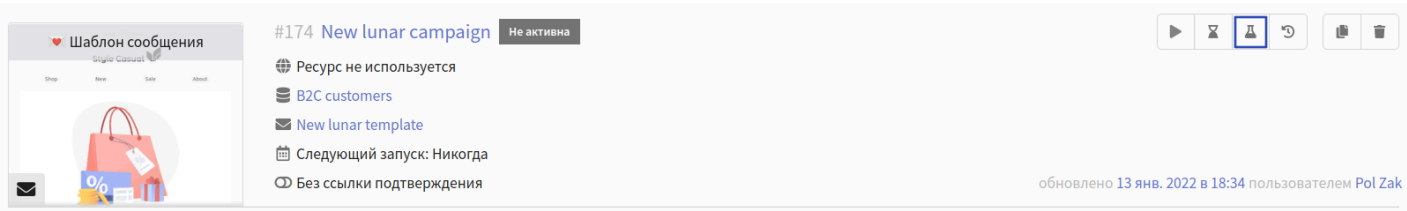
- техническая часть отправки и доставки сообщения в порядке;
- сообщение этой рассылок в различных почтовых системах выглядит должным образом;
- все подстановки и сложная логика внутри шаблона работает правильно;
- email и sms провайдеры не блокируют сообщение из-за контента.

Используя возможность тестирования рассылок, вы можете отправлять письма на проверку ответственному за ваши рассылки лицу или использовать сервисы, оценивающие качество сообщений, инфраструктуру доставки и.т.д.

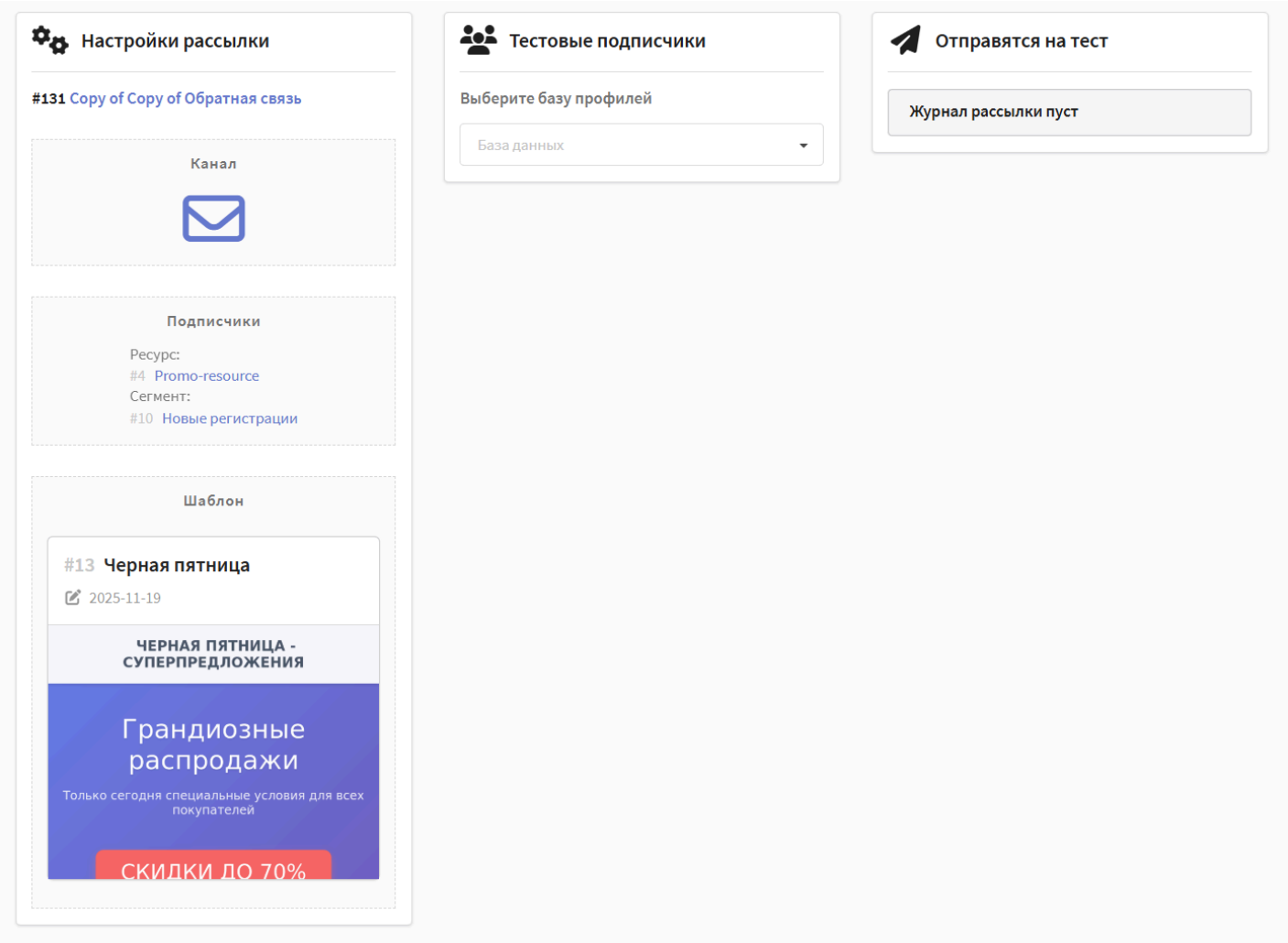
Как запустить тестирование

Зайдите на страницу **Рассылки** и в меню действий нужной рассылки нажмите

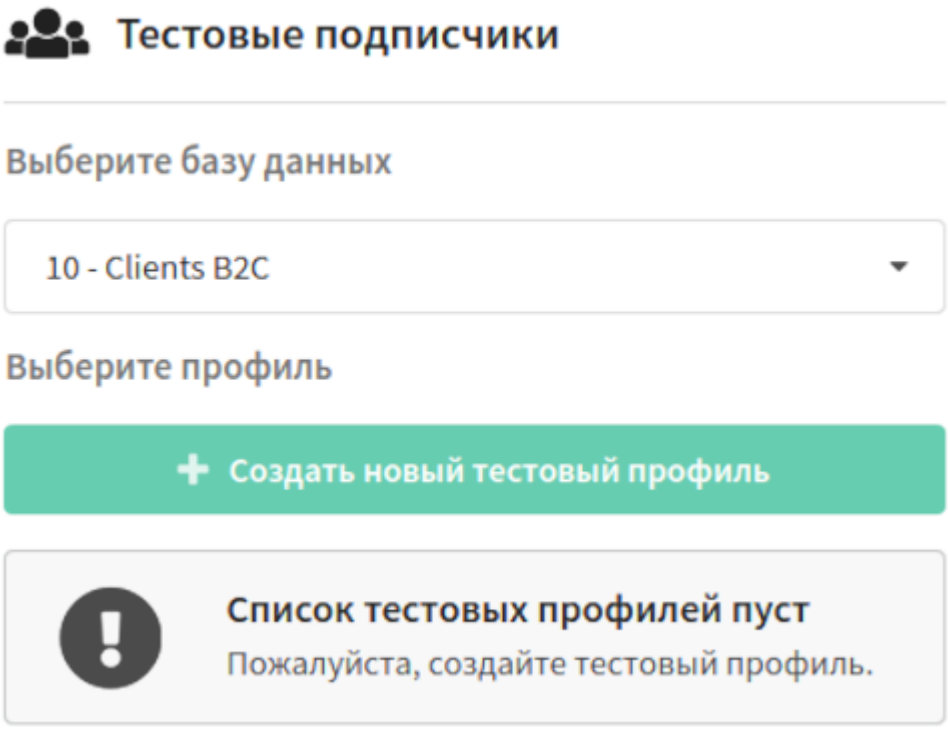
:



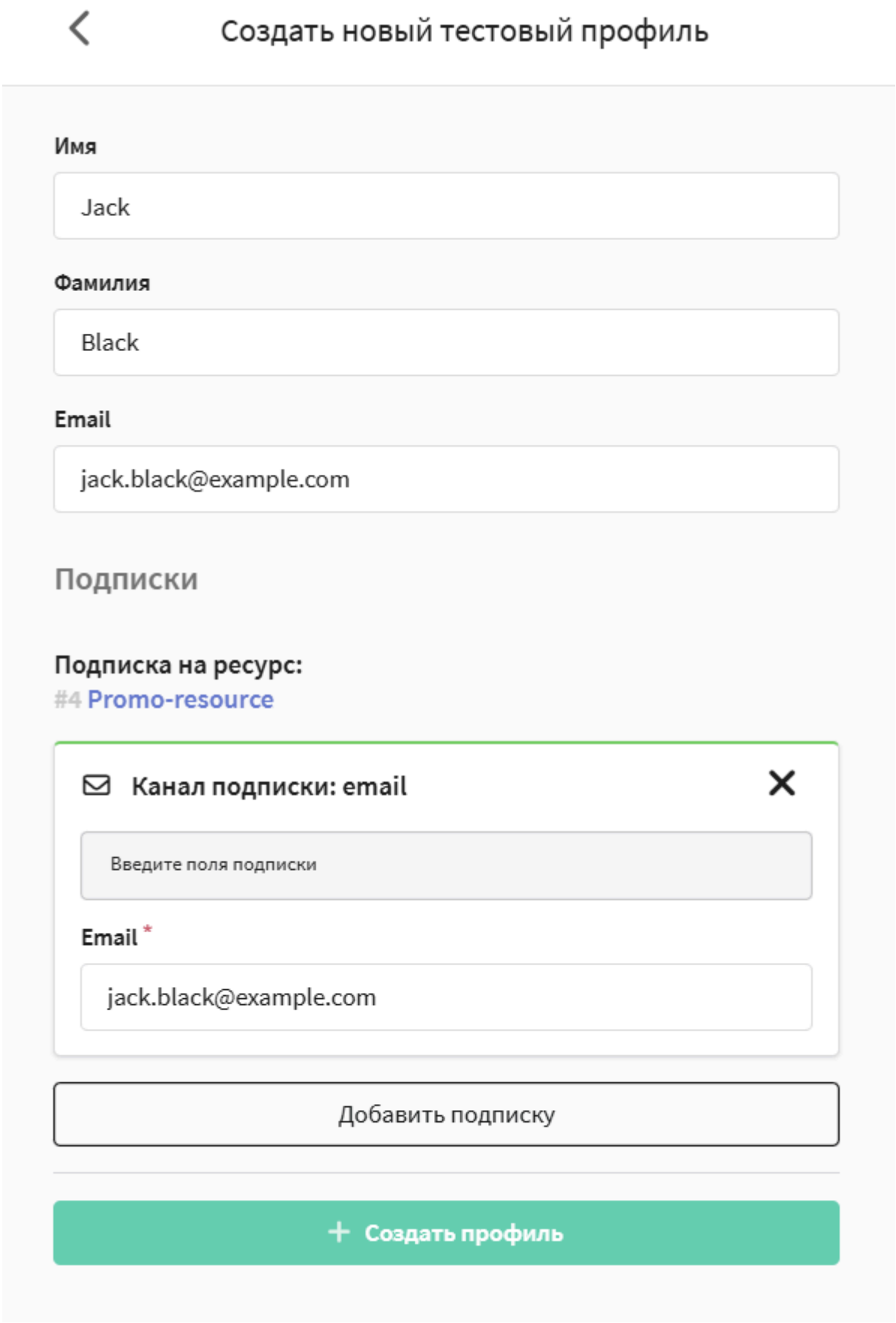
Появится окно, где вы сможете посмотреть все настройки тестируемой рассылки, а также выбрать тестовых подписчиков.



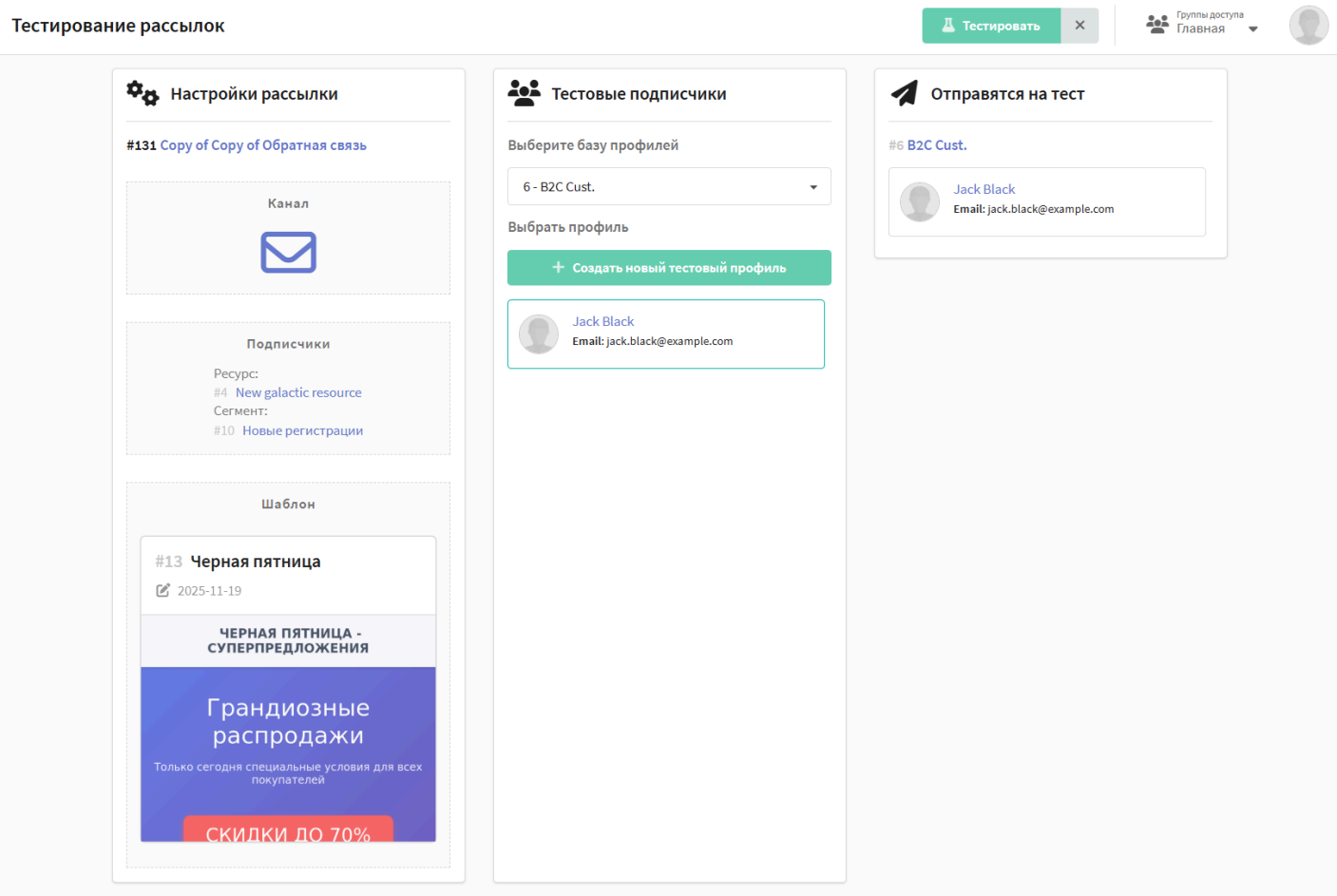
Если у вас пока нет тестовых email, их можно добавить. Выберите базу профилей в выпадающем списке. Поскольку профиль тестового подпичика еще не создан, платформа тут же предложит создать профиль в выбранной базе. Кликните **“Создать новый тестовый профиль”:**



Справа выплывет окно с интерфейсом создания тестового профиля:



Заполните поля соответствующими данными и нажмите кнопку **“Создать профиль”**. Профиль отобразится в списке тестовых подписчиков. Кликните на него, чтобы “отправить на тест”.



Нажмите кнопку **“Тестировать”** — произойдет тестовая отправка рассылки. На экране появится новое окно с журналом рассылки:

Время	Тип	Значение
2025-12-04 12:02:13	Запуск рассылки	
2025-12-04 12:02:13	Общий лимит	1
2025-12-04 12:02:13	Источник подписчиков	segment
2025-12-04 12:02:13	ID сегмента	10
2025-12-04 12:02:13	Время получения сегмента	20.759646ms
2025-12-04 12:02:13	Количество профилей полученных из источника	1
2025-12-04 12:02:13	Количество найденных подписчиков	1
2025-12-04 12:02:13	Количество подписчиков, кому был отправлен email	1
2025-12-04 12:02:13	Рассылка завершена	
2025-12-04 12:02:13	Всего времени затрачено	169.82222ms

TIP

Рекомендуем иметь по одному тестовому адресу для всех основных email провайдеров!

Расписание рассылок

Расписание применяется в броадкастах, регулярных рассылок, мультивариантных тестах, а также в сценариях, импортах по расписанию и при синхранизации статических сегментов.

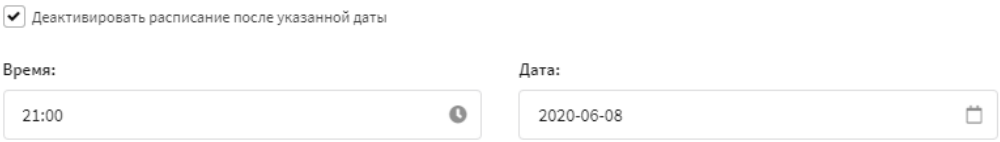
В расписании доступно несколько **планов** в зависимости от типа рассылки. Ниже в статье описаны возможные варианты.

NOTE

Для каждого плана расписания можно указать часовой пояс отправки.

NOTE

С помощью опции **Деактивировать расписание после указанной даты** вы можете остановить рассылок:

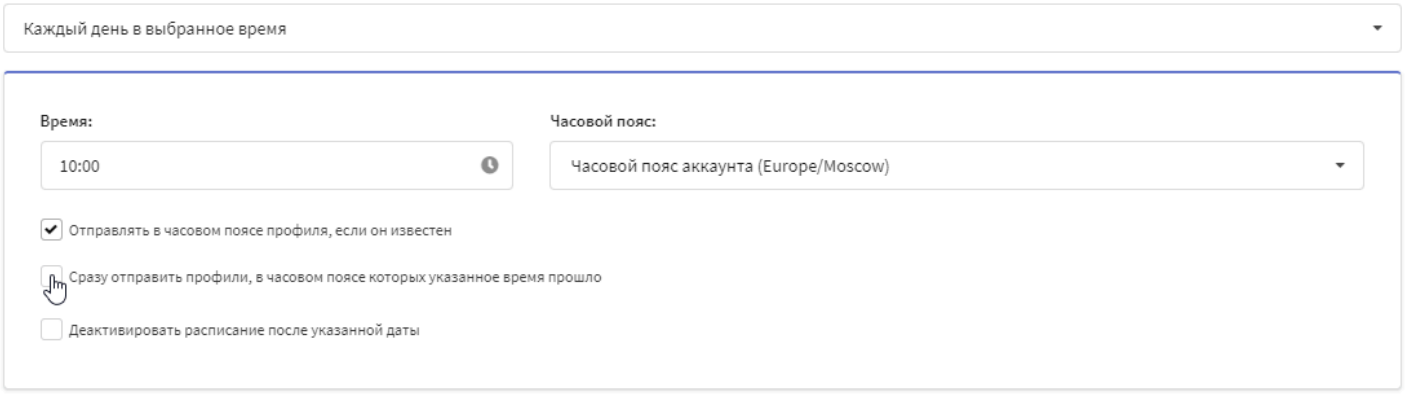


Каждый день в выбранное время

Этот вариант подойдёт для динамических регулярных рассылок и цепочек: рассылка будет запускаться каждый день в указанное время.

TIP

Для этого плана расписания доступна отправка в часовом поясе клиента. Если для кого-то из подписчиков время прошло на момент создания рассылок - есть возможность отправить им сообщения сразу.



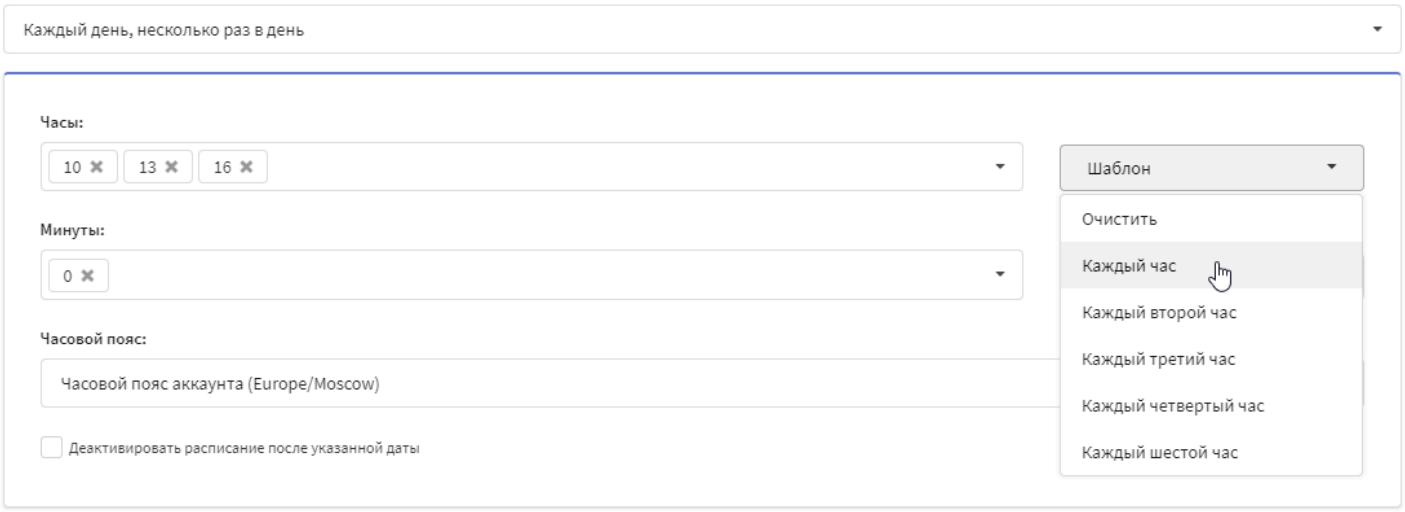
TIP

Этот план расписания также удобно использовать при разделении больших броадкаст-рассылок, установив часть аудитории для отправки каждый запуск.

Каждый день, несколько раз в день

Если каждый день необходимо запускать рассылку несколько раз. Укажите, в какие часы и минуты каспания будет запускаться.

В правой части вы можете выбрать шаблоны для часов и минут.

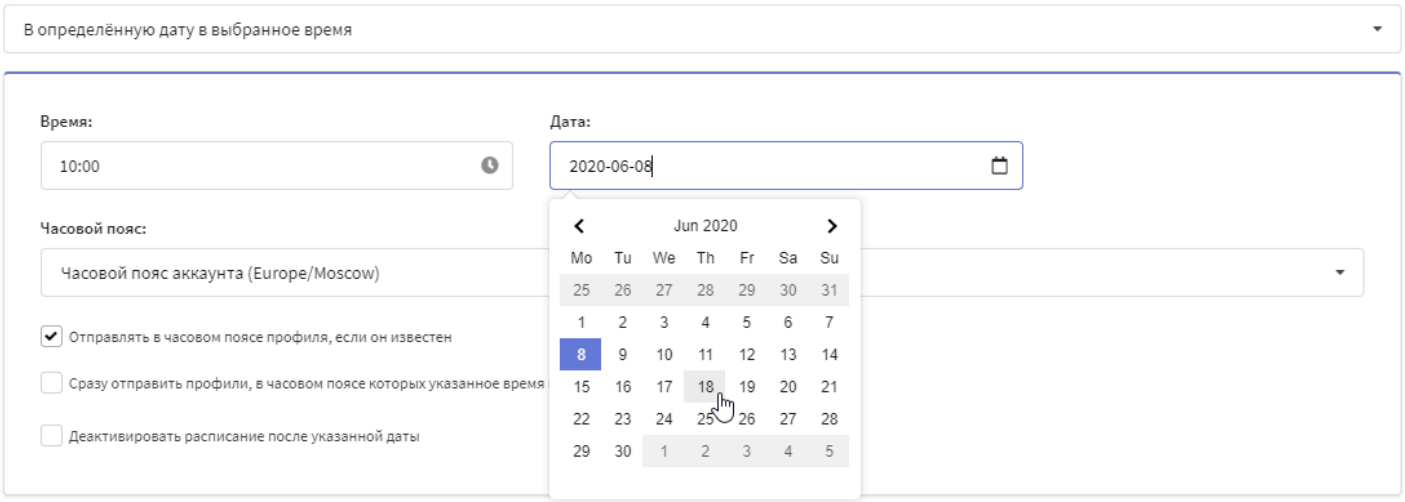


В определённую дату в выбранное время

Этот план отлично подходит для разовых рассылок: выберите дату и установите время запуска рассылок.

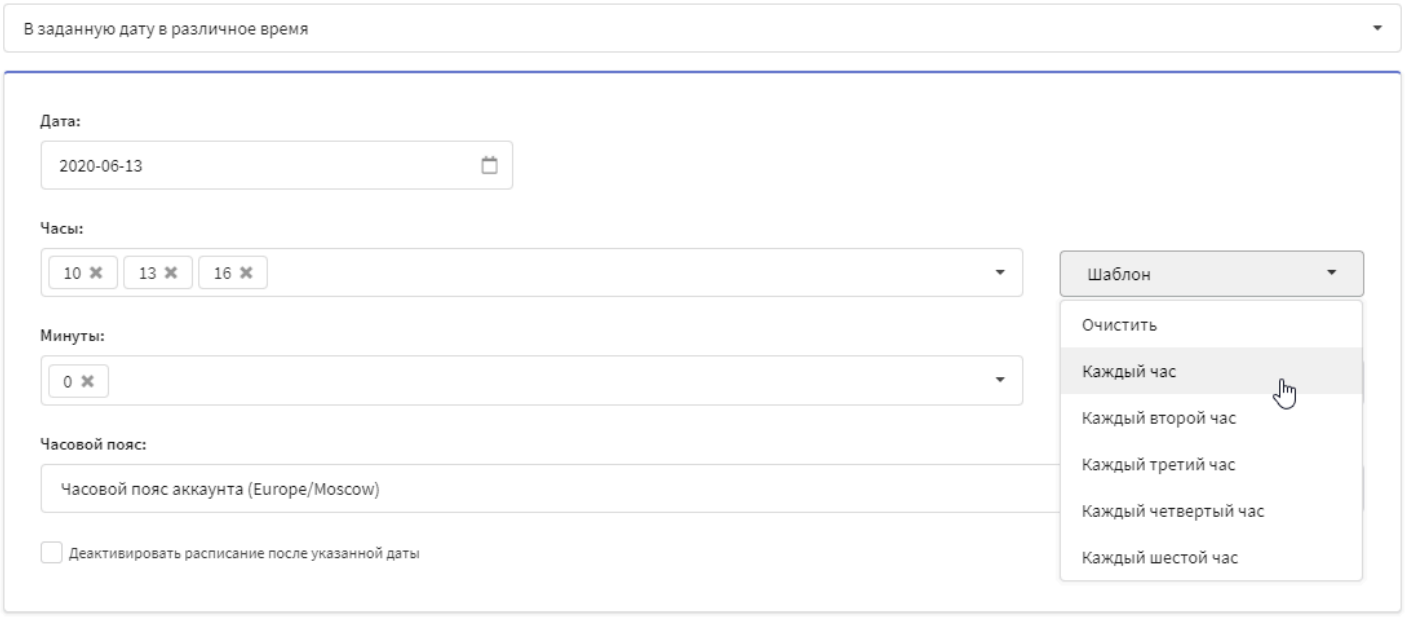
TIP

Для этого плана расписания доступна отправка в часовом поясе клиента. Если для кого-то из подписчиков время прошло на момент создания рассылки - есть возможность отправить им сообщения сразу.



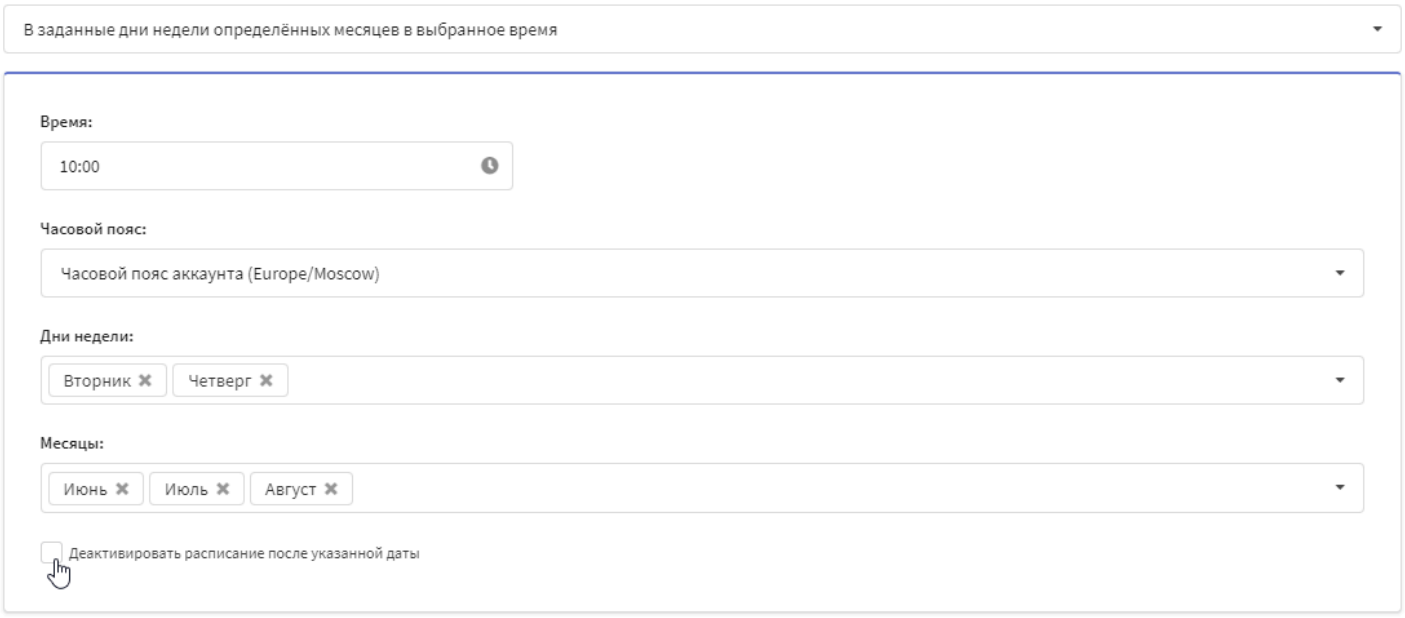
В заданную дату в различное время

Этот план отличается от предыдущего тем, что вы можете добавить несколько запусков в одну определённую дату.



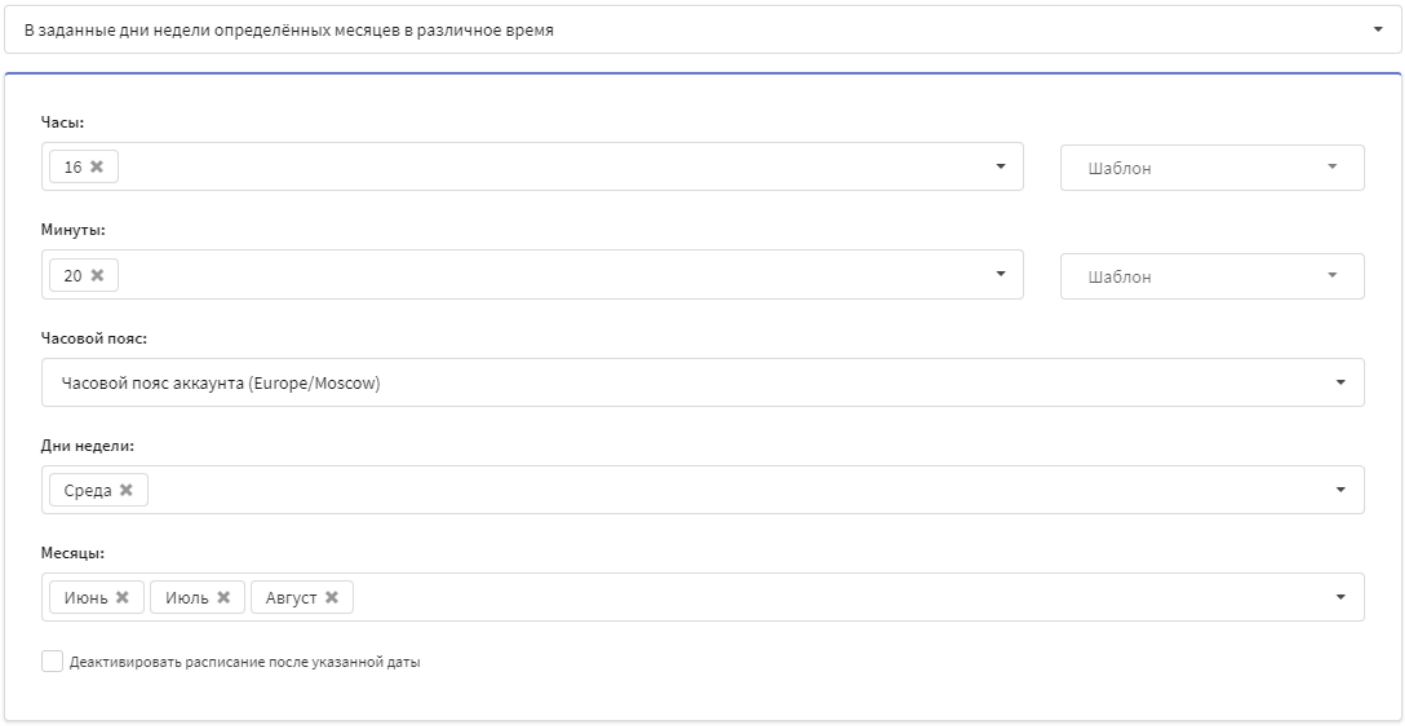
В заданые дни недели определённых месяцев в выбранное время

Подойдёт для еженедельных рассылок в определённые дни недели. Укажите месяцы, в которые рассылка будет активна и день недели (или несколько) для отправки.



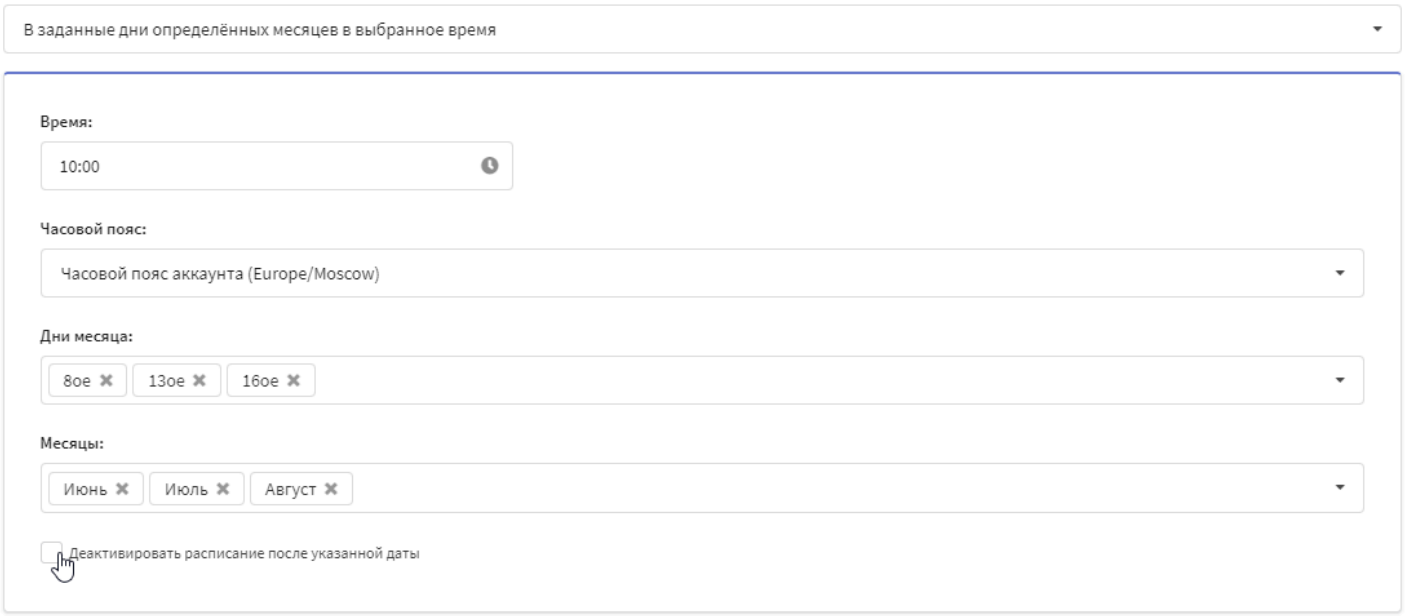
В заданые дни недели определённых месяцев в различное время

Отличается от предыдущего количеством запусков в один день.



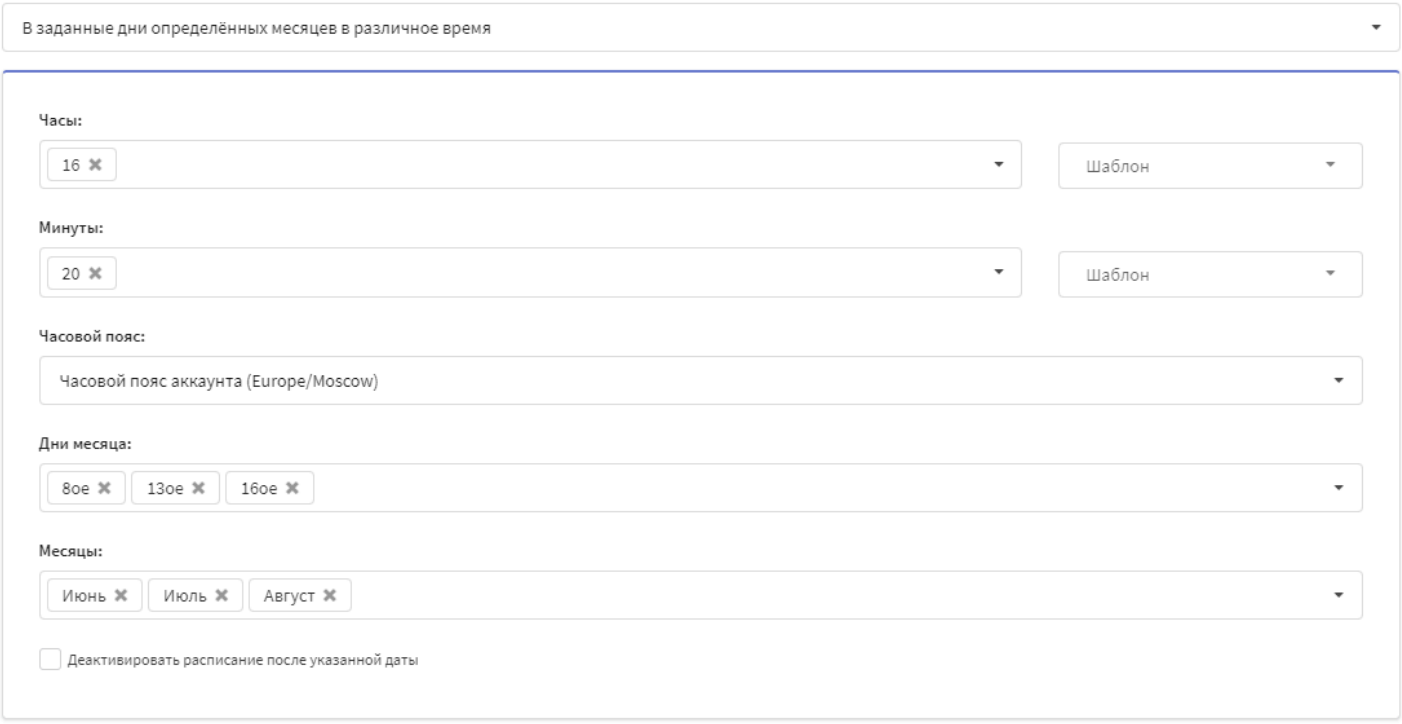
В заданые дни определённых месяцев в выбранное время

В этом плане вместо дней недели устанавливается день от начала месяца. Подойдёт для периодических ежемесячных или ежеквартальных рассылок.



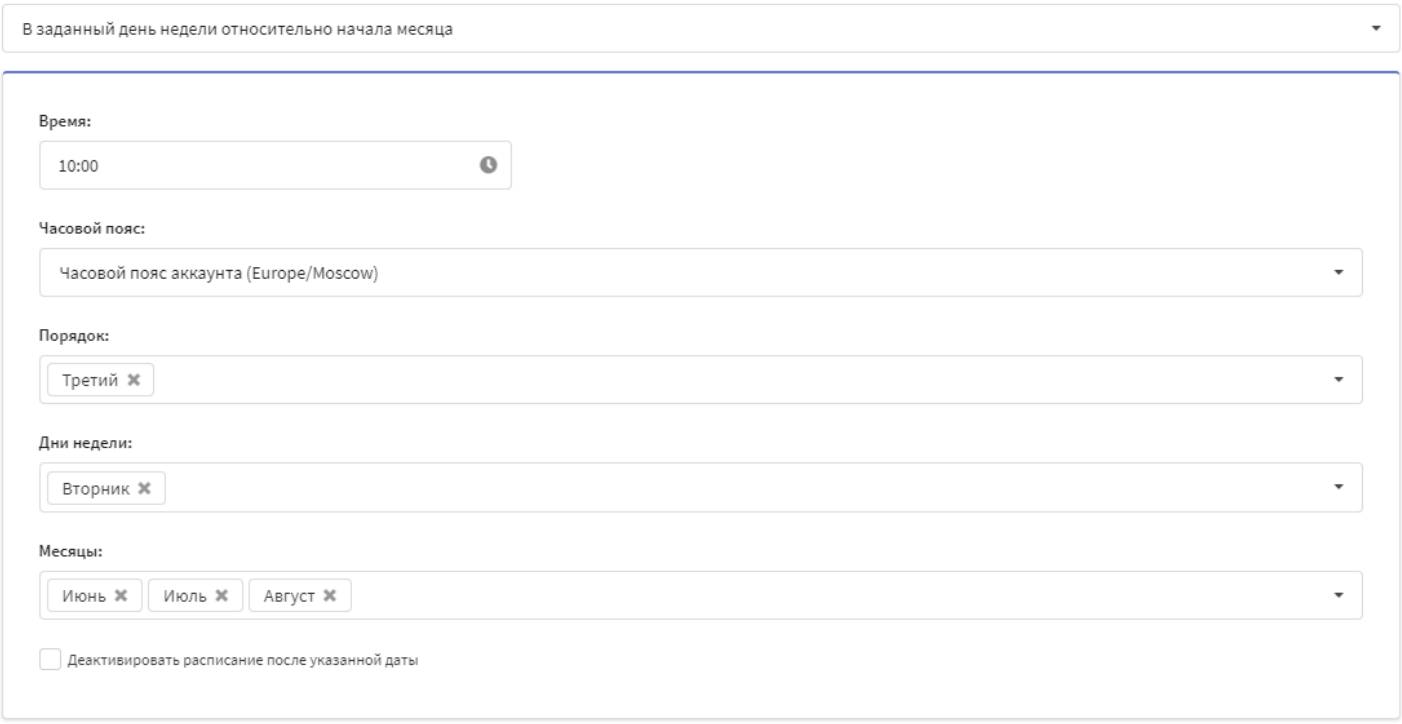
В заданые дни определённых месяцев в различное время

Отличается от предыдущего количеством запусков в один день.



В заданный день недели относительно начала месяца

Если вам нужно организовать рассылку, например, во второй четверг ноября, или во второй вторник каждого месяца, этот план расписания решает такие задачи:



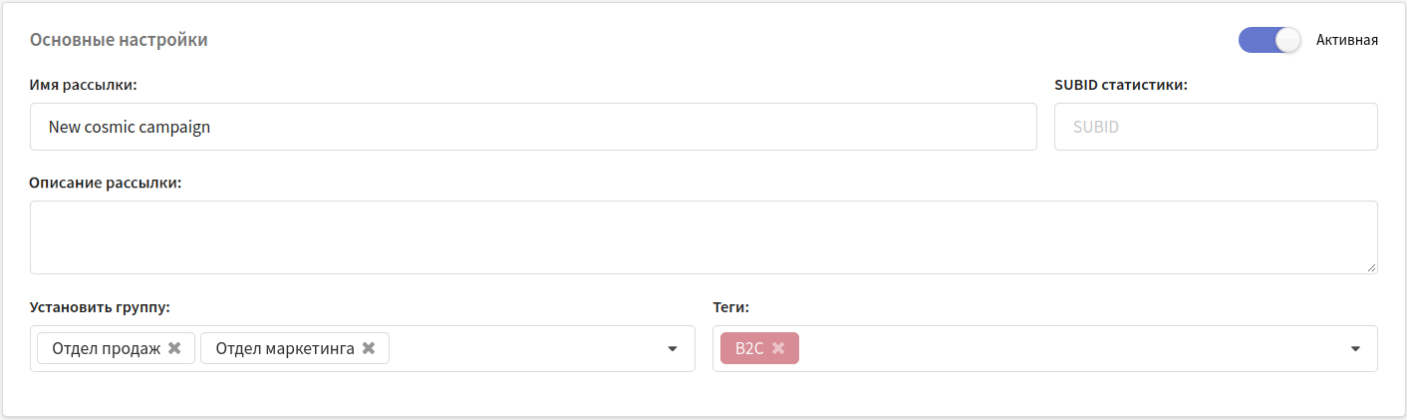
Размещения

Размещения — инструмент для размещения контента подписчикам на определенном ресурсе с возможностью последующей отмены. При запуске отправляет контент каждому подписчику в источнике данных (в случае старта по расписанию) или конкретному лиду, инициировавшему определенное событие (при триггерном условии старта). Если в канале настроена возможность отмены рассылки, то размещенный контент может быть убран с ресурса в ручную или по достижению определенной даты и времени.

	Чтобы создать новую рассылку, в главном меню выберите **Рассылки** → **Размещения** и нажмите кнопку**+ Создать**.

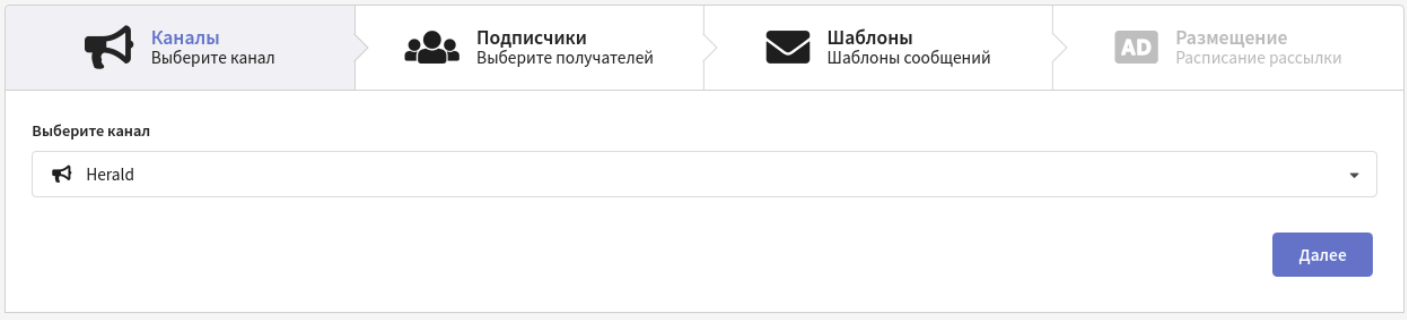
Основные настройки

В основных настройках укажите название и описание новой рассылки. При необходимости установите группу доступа, **теги** и **атрибуты** для дополнительной группировки. В верхнем правом углу находится переключатель, активирующий рассылку:



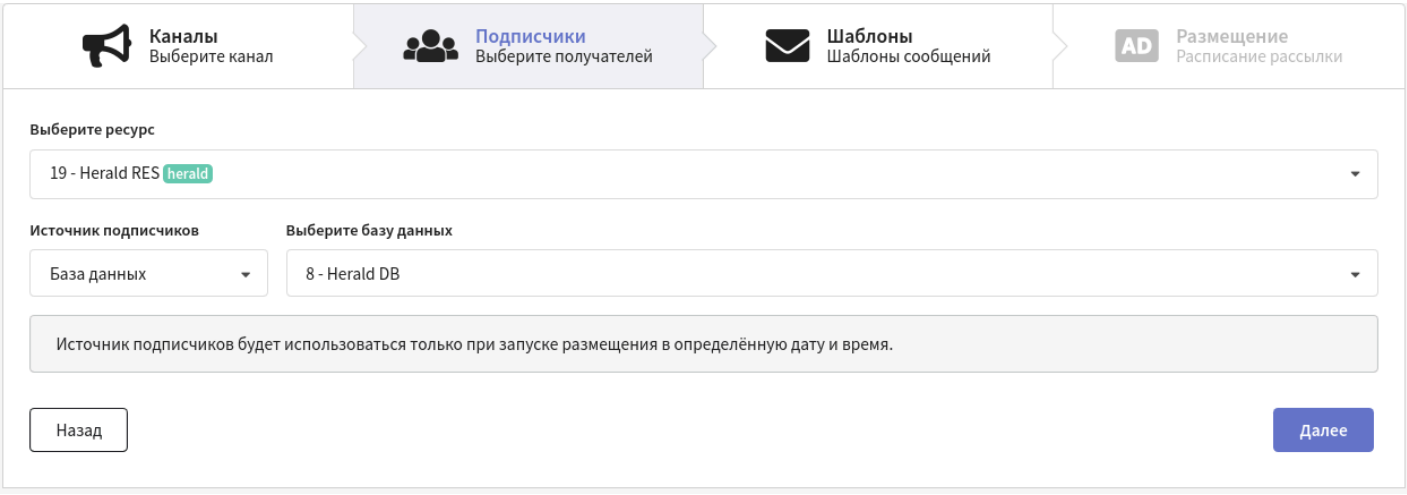
Канал

Выберете канал отправки для которого доступен данный тип рассылки из выпадающего меню.

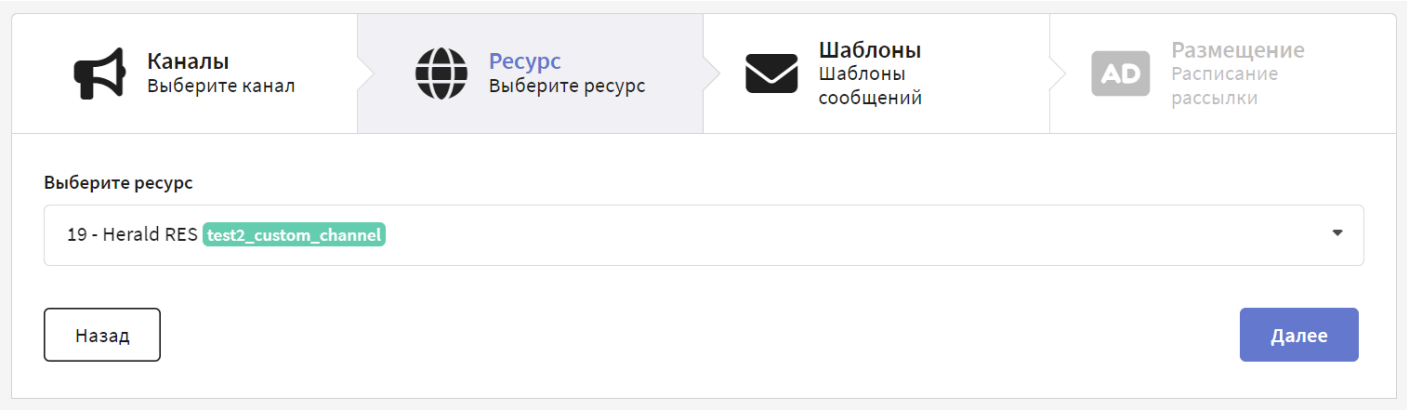


Подписчики

Выберите получателей рассылки. Рассылки в Altcraft отправляются в рамках определённого ресурса: это может быть отдельный сайт, продукт, акция, или другое направление ваших коммуникаций с клиентами. Выберите **ресурс** и **источник подписчиков**. Получателями размещения может быть база профилей или сегмент, состоящий из профилей одной или нескольких баз данных.



Если вы используете размещение для отправки сообщений через кастомный канал, настройки рассылки могут отличаться в зависимости от конфигурации канала. Например, для некоторых каналов может потребоваться выбрать только ресурс, не указывая источник подписчиков:



TIP

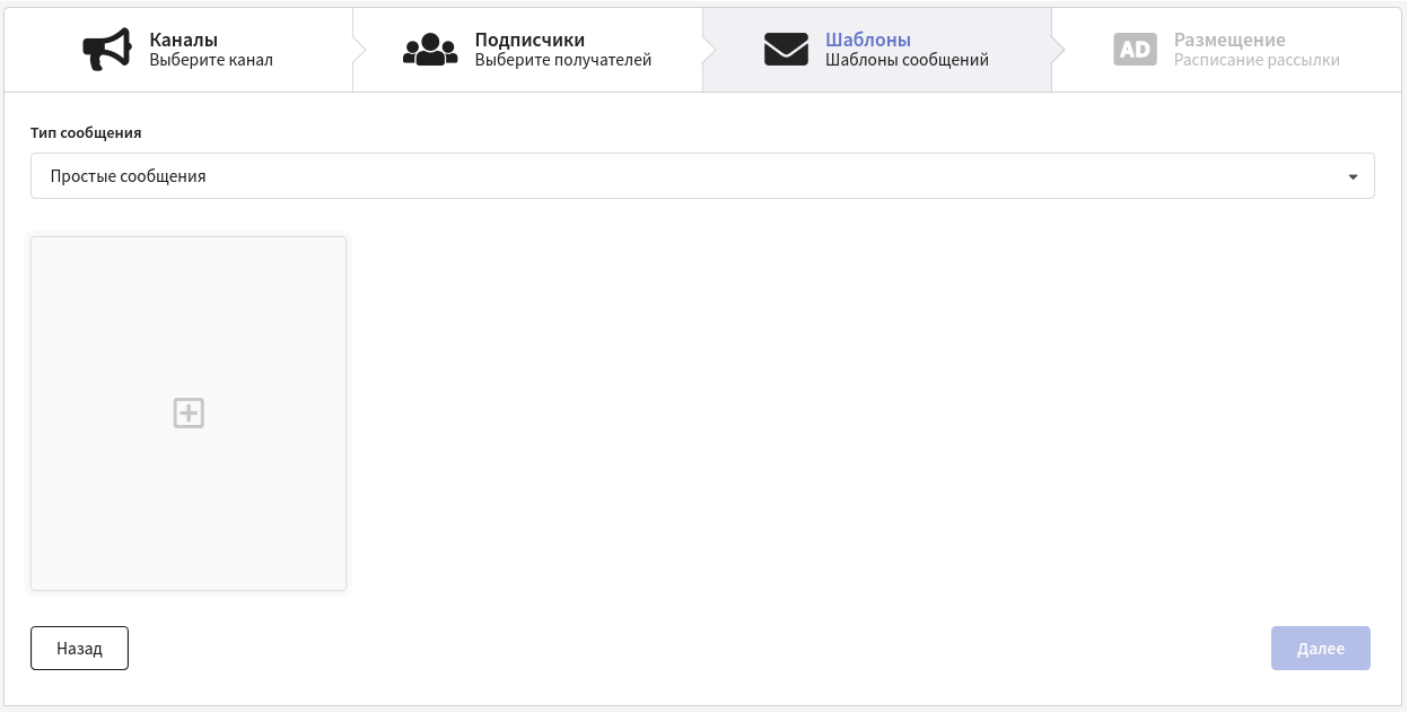
При отправке сообщений рассылка использует **подписки клиентов** на выбранный ресурс. Убедитесь, что получатели в вашем источнике данных подписаны на него!

Шаблоны

На этом шаге выберите контент, который хотите использовать в размещении.

Нажмите на кнопку **Добавить контент** и выберите подготовленный шаблон. Любой тип рассылки поддерживает до 8 шаблонов сообщений. Рассылка отправит каждому получателю по **одному** сообщению из выбранных в случайном порядке - в равных пропорциях.

По клику на миниатюру шаблона откроется предпросмотр сообщения.



Размещение

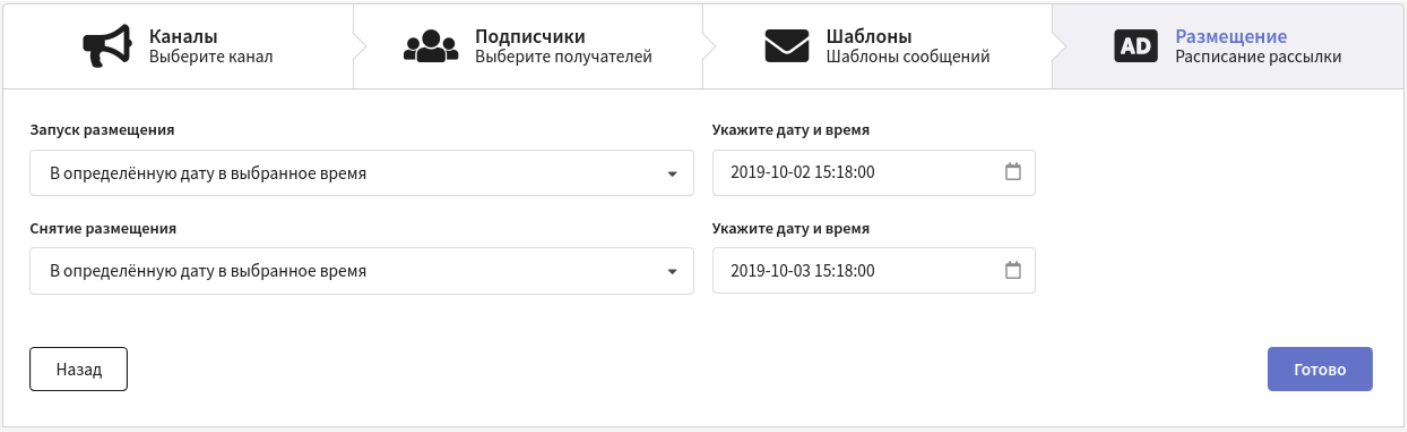
Настройка старта и остановки

Размещение может запускаться двумя способами:

- в определенную дату и время для всех подписчиков в источнике
- по событию для одного подписчика

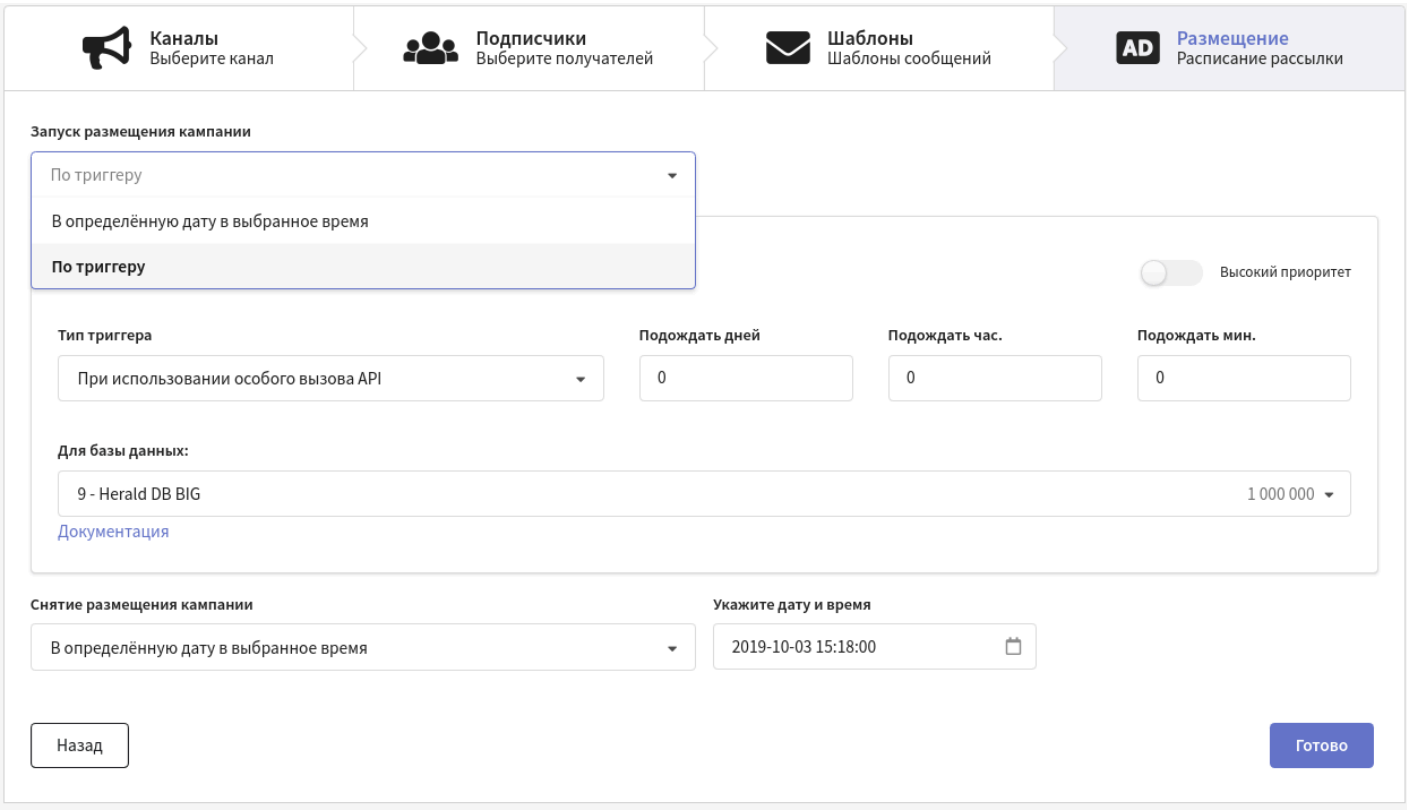
Старт в определенную дату и время

Рассылка запустится один раз в указанное время. После этого рассылка не будет доступна для перезапуска.



Старт по событию

Настройка аналогична триггерной рассылке. Контент будет отправляться каждому подписчику, который будет захвачен триггером



Отмена размещения

Отмена размещения доступна только для каналов, в которых она настроена. Если отмена рассылки не была предусмотрена администратором при конфигурации канала, данный пункт будет не доступен для редактирования в рассылке.

Отмена может быть двух типов:

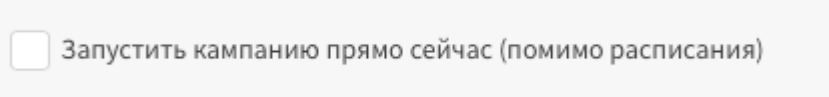
- в определенную дату и время
- вручную

Оба варианта отменят размещения для всех подписчиков, кому был отправлен контент. Не зависит от типа старта. Если старт был настроен на триггер, то триггер будет деактивирован перед отменой.

Вручную рассылка размещения может быть отменена, только если она имеет статус "**Размещена** " (“Deployed”).

Мгновенный запуск рассылки (помимо расписания)

Чтобы запустить рассылку сразу же после создания, в самом низу страницы отметьте соответствующую опцию:



Smart settings

Блок **Smart-настройки** объединяет следующие дополнительные настройки рассылки:

- **Виртуальный сендер** — позволяет выбрать для рассылки виртуальный сендер (набор правил для отправки сообщений с реальных сендеров платформы).

- **Свои трекинг-домены** — позволяет отслеживать действия с сообщениями рассылки, используя дополнительные трекинг-серверы. Каждый домен указывается на отдельной строке.
- **Список email адресов для скрытой копии** — BCC для отправляемых сообщений. Каждый адрес указывается на отдельной строке.
- **Не проверять контактные данные на уникальность** — отключает проверку на уникальность получателей рассылки по данным подписки.

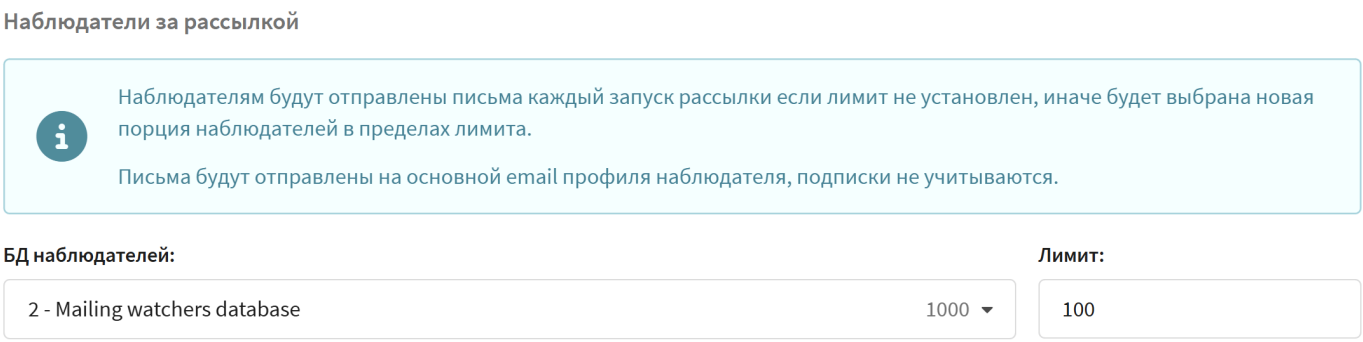
CAUTION

Если вы не уверены, какие настройки необходимо выбрать, проконсультируйтесь с администратором платформы в вашей организации или со специалистами поддержки Altcraft.

Наблюдатели за рассылкой

Помимо этого, в умных настройках можно установить **наблюдателей за рассылкой**. Наблюдатели будут получать такие же сообщения, как и получатели рассылки. Для этого вам нужна будет **база данных наблюдателей**. Она создаётся так же, как и обычная база данных, только в Основных настройках необходимо отметить соответствующую опцию.

В Smart-настройках выберите **БД наблюдателей** и лимит отправки сообщений для каждого запуска рассылки:



Кампании

INFO: Обновление терминологии

Раньше **“Кампаниями”** в платформе назывались все виды маркетинговых рассылок. В новой версии платформы они переименованы в **“Рассылки”**, а новые **“Кампании”** представляют собой более широкий инструмент.

Кампании — это инструмент для комплексного управления маркетинговыми активностями в едином пространстве.

Позволяет настраивать аудиторию, сегментировать базы данных, управлять профилями пользователей, контрольными группами и сценариями.

В рамках кампаний можно как создавать, так и переносить уже готовые маркетинговые активности: сценарии, броадкасты, регулярные, триггерные, мультивариантные и размещения.

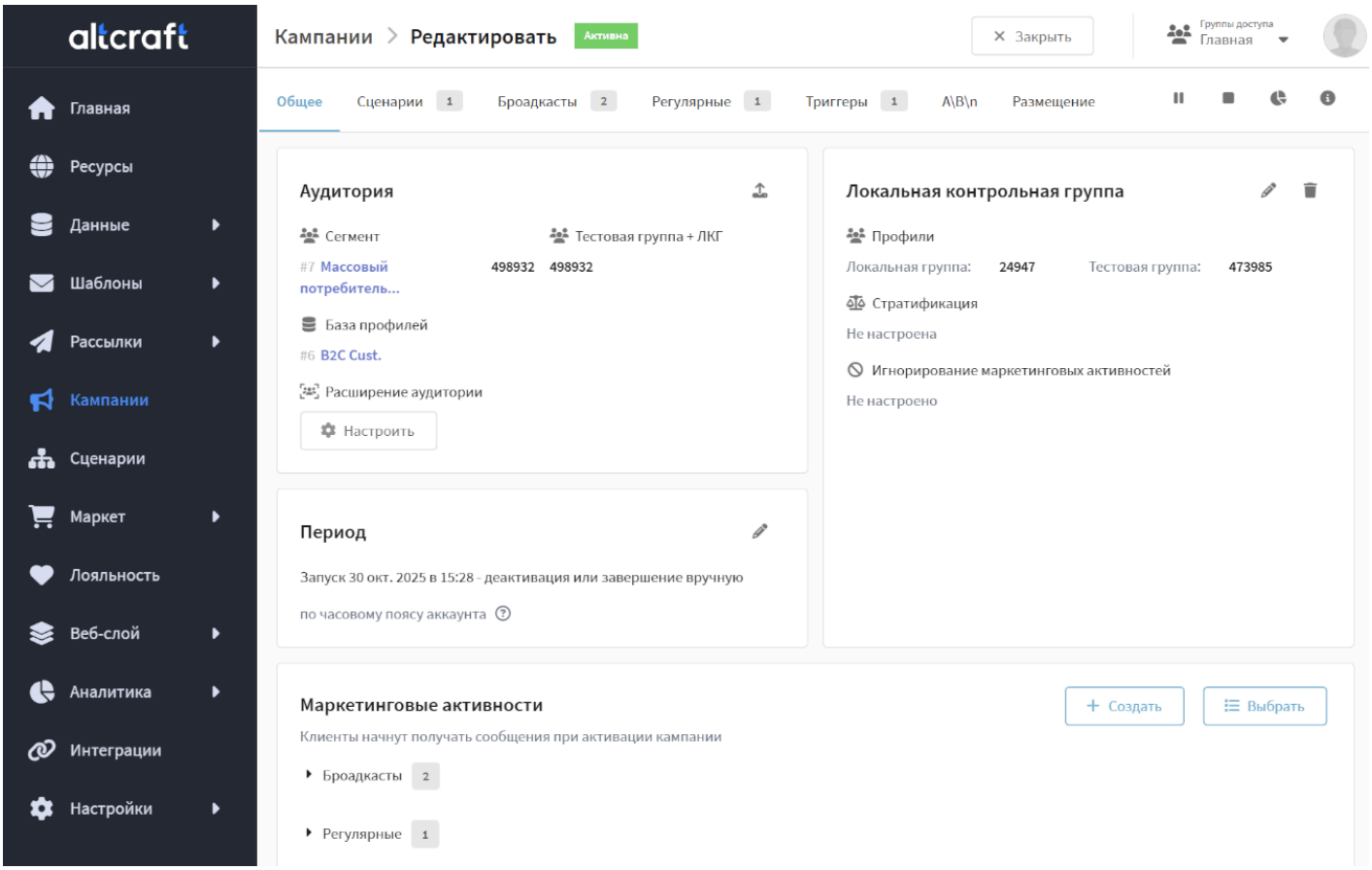
Работа с Кампаниями

Описание

INFO: Обновление терминологии

Раньше **“Кампаниями”** в платформе назывались все виды маркетинговых рассылок. В новой версии платформы они переименованы в **“Рассылки”**, а новые **“Кампании”** представляют собой более широкий инструмент.

Кампании — это инструмент для объединения маркетинговых активностей в единую стратегию и управления ими из одного места. В кампании можно собирать броадкасты, регулярные и триггерные рассылки, сценарии, A/B-тесты и размещения, настраивать тестовые и контрольные группы для достижения маркетинговых целей, отслеживать эффективность стратегии и управлять жизненным циклом активностей из единого окна:



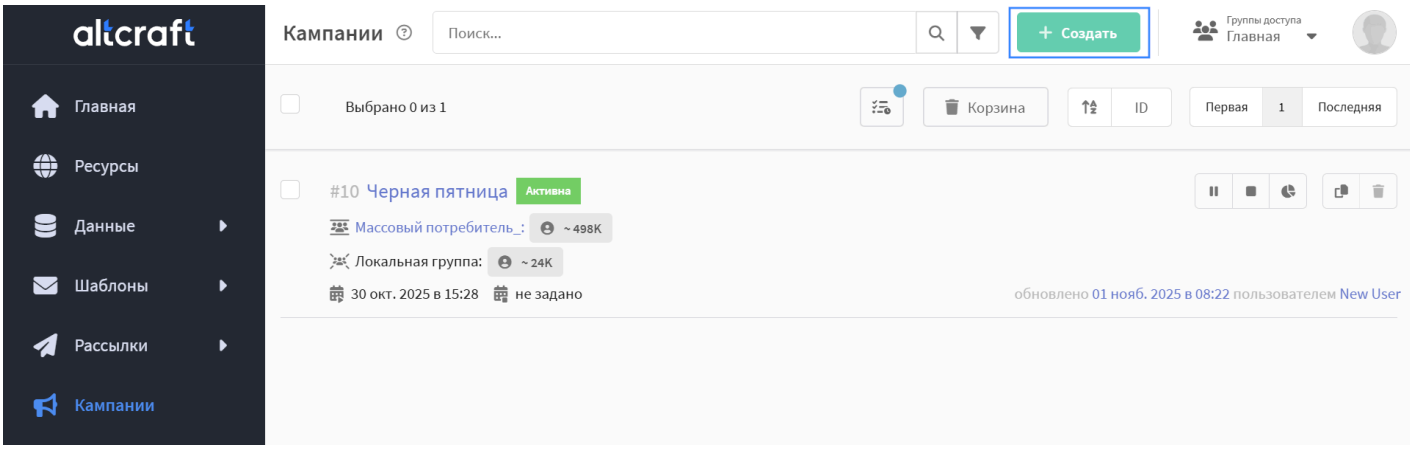
Создание кампании

Перед созданием подготовьте статический или обновляемый сегмент для целевой аудитории. Он понадобится при настройке кампании и определяет, кто будет участвовать в активностях.

INFO: Обратите внимание:

При использовании обновляемого сегмента в качестве основного сегмента кампании функция расширения аудитории недоступна.

Откройте раздел **Кампании** в главном меню и нажмите `+ Создать` в списке кампаний:



На первом этапе укажите **название**, выберите **базу профилей** и **статический сегмент**, который определяет целевую аудиторию.

Если в системе настроены обязательные атрибуты, их также нужно выбрать на этом этапе.

> **NOTE: Обратите внимание**
>
> Выбранный **сегмент** определяет аудиторию кампании. При активации система разделит её на две части:
>
> - **Тестовая группа** — профили, которые будут получать активности кампании;
> - **Локальная контрольная группа (ЛКГ)** — профили, которые не получают активности, **если ЛКГ включена в настройках**.

> **CAUTION**
>
> После создания кампании изменить сегмент и базу профилей нельзя.

Управление маркетинговыми активностями

В разделе **“Маркетинговые активности”** можно управлять всеми активностями, которые входят в кампанию.

Активности можно создавать прямо внутри кампании или добавлять уже существующие.

> **NOTE: Обратите внимание**
>
> При выборе активности **в список не попадают:**
>
> - активности из другой базы профилей;
> - рассылки и сценарии в статусе “Активна” или “Завершена”;
> - сценарии с активными профилями;
> - активности, уже используемые в других кампаниях;
> - броадкасты и регулярные рассылки, которые хотя бы раз запускались.

Доступны следующие типы активностей:

- **Броадкасты** — разовые массовые отправки;
- **Регулярные рассылки** — отправки по расписанию;
- **Триггерные рассылки** — автоматические отправки по событиям;
- **A/B-тесты** — сравнение вариантов для достижения маркетинговых целей;
- **Сценарии** — многошаговые автоматизации;
- **Размещения** — показ контента подписчикам на выбранном ресурсе.

При создании активности можно выбрать любой **сегмент рассылки**, если требуется уточнить аудиторию. Выбрать любую **базу профилей** нельзя — она всегда совпадает с базой кампании. Если выбранный сегмент активности отличается от сегмента кампании, система проверяет пересечение между ними: сообщение будет отправлено **только тем профилям, которые входят одновременно в оба сегмента**.

Как работает пересечение

Частичное совпадение сегментов

Кампания настроена на базу профилей `CRM_База` и сегмент `Постоянные клиенты`, в котором **100 профилей** активных клиентов.

Маркетолог хочет отправить дополнительное сообщение только тем, кто недавно оформлял заказ. Для этого при создании броадкаста он выбирает другой сегмент — `Недавние покупатели`, где 60 профилей. Из них **40 совпадают** с пользователями из сегмента `Постоянные клиенты`.

В этом случае сообщение **получат только эти 40 профилей** — те, кто входит одновременно в сегмент `Постоянные клиенты` и `Недавние покупатели`. Остальные участники кампании сообщение не получат.

Это полезно, когда нужно настроить отдельную активность для определённой группы пользователей, не изменяя основную кампанию.
Например, можно выбрать более узкий сегмент — активных клиентов, участников акции или пользователей из конкретного региона — и отправить им сообщение, связанное с их интересами или текущими маркетинговыми целями.

Сегменты не пересекаются

Кампания использует базу профилей `CRM_База` и сегмент `Постоянные клиенты` (100 профилей).
Создаётся броадкаст с сегментом `Новые регистрации` (50 профилей), куда входят пользователи, которые только создали аккаунт и ещё не совершали действий.

Так как **общих пользователей между сегментами нет**, сообщение никто не получит.

Перемещение активности

Добавить или исключить активность можно также из меню рассылок. Для этого найдите нужную активность в списке и нажмите **“Переместить”**:

В открывшемся окне выберите, куда нужно переместить активность — во **внешний список** или в **нужную кампанию**:

> **INFO: Обратите внимание**
>
> Не все активности можно переместить:
>
> - Активности в статусе “Активна” или уже запускавшиеся;

- Сценарии с профилями;
- Нельзя переместить активность в завершенную кампанию.

Настройка локальной контрольной группы (ЛКГ)

Что такое Локальная контрольная группа

Локальная контрольная группа (ЛКГ) — это часть аудитории кампании, которая не получает активности из этой кампании. Она используется для более точного управления аудиторией и анализа влияния кампаний на пользователей. Сейчас функция аналитики ЛКГ в платформе недоступна.

TIP: ЛКГ и ГКГ

Локальная контрольная группа (ЛКГ) работает только внутри одной кампании и настраивается в её параметрах.
Глобальная контрольная группа (ГКГ) действует для всей базы профилей и исключает выбранных пользователей из всех активностей на платформе.

Если профиль входит в ГКГ, он не сможет участвовать в кампании, кроме случаев, когда эта кампания добавлена в исключения.
ГКГ настраивается в параметрах базы профилей. Подробнее — в **отдельной статье**.

Базовая настройка ЛКГ

В настройках кампании можно выбрать, какая часть аудитории попадёт в локальную контрольную группу.

Для настройки ЛКГ укажите процент пользователей, которых нужно исключить из отправок в рамках кампании. Если включить опцию **“Лимит аудитории”**, можно задать не процент, а точное количество профилей, которые войдут в ЛКГ.

TIP

Если вы планируете использовать локальную контрольную группу, рекомендуется настроить ЛКГ **до запуска кампании**: После изменения параметров запускается перерасчёт аудитории: в это время все броадкасты и регулярные рассылки приостанавливаются до завершения процесса.
Чтобы избежать задержек и ошибок в работе кампании, настройте контрольную группу заранее, а уже потом активируйте кампанию.

В случае если кампания настроена на обновляемый сегмент, разметка аудитории будет запускаться **при каждом пересчете**.

После активации кампании и завершения разметки в локальную контрольную группу будут попадать только **новые пользователи**. Пользователи, уже попавшие в тестовую группу на момент активации, не перемещаются в ЛКГ — состав тестовой аудитории фиксируется при разметке.

Стратификация

Стратификация — это способ более точного формирования локальной контрольной группы.
При стратификации аудитория разбивается на подгруппы по выбранным характеристикам (например, возраст, город, тип клиента), и из каждой из них случайно выбирается часть профилей для ЛКГ.

Настройка стратификации доступна в окне параметров ЛКГ через выпадающее меню. Выберите одно или несколько полей, по которым нужно задать страты, и определите условия для каждой группы (например, диапазоны значений или списки элементов):

Пример использования:
Маркетологу необходимо включить в контрольную группу мужчин и женщин двух возрастных категорий:

- от 20 до 25 лет (дата рождения от **01.01.2000** до **01.01.2005**);
- от 25 до 40 лет (дата рождения от **01.01.1985** до **31.12.1999**).

Для этого он выбирает поля **“Пол”** и **“Дата рождения”**, задаёт два диапазона по дате рождения и добавляет значения пола (**мужчина** и **женщина**). Система сформирует четыре страты — мужчины 20-25, женщины 20-25, мужчины 25–40 и женщины 25–40 — и распределит участников между тестовой и контрольной группами согласно настройкам.

Игнорирование маркетинговых активностей

Игнорирование маркетинговых активностей позволяет выбрать внутри кампании рассылки или сценарии, которые будут отправляться всем пользователям, **включая участников ЛКГ.** Это полезно для сообщений, которые должны доходить до всей аудитории независимо от участия в контрольной группе. Например, вам нужно, чтобы **уведомления о заказах** доходили до получателя даже если он попал в контрольную группу.

Для этого в настройках ЛКГ через выпадающее меню выберите нужные активности и отметьте их как исключения. Такие рассылки будут отправляться всем пользователям без ограничений:

Подробнее о настройке в **отдельной статье о ЛКГ**.

Управление статусами

Активация кампании

> **CAUTION**
>
> Активация доступна только для кампаний, в которых есть хотя бы одна маркетинговая активность. Если в кампании не создано ни одной активности, запустить её невозможно.

Для активации кампании нажмите кнопку ▶ на верхней панели редактора. После нажатия откроется окно настройки параметров запуска.

Доступно два варианта активации:

- **Активировать сейчас** — кампания запускается сразу после сохранения. Начинается процесс разметки аудитории на тестовую и контрольную группы, после чего все активности внутри кампании начинают работать согласно своим настройкам.
- **Активировать в определённую дату** — кампания запланирована к запуску в указанное время. Система автоматически активирует кампанию в выбранную дату и время.

Будьте внимательны при выборе даты и времени для запуска кампании. Если вы укажете прошедшую дату, кампания будет бесконечно ожидать запуска и не выведет предупреждение.

> **NOTE**
>
> Активация кампании **не запускает активности внутри неё автоматически**.
> Если рассылки, сценарии или другие активности находятся в статусе “Неактивна”, их нужно запустить вручную.

Настройка завершения работы

В этом же окне можно задать условия автоматической деактивации или завершения кампании:

- **Деактивация** — временная остановка кампании в указанную дату. Все активные рассылки приостанавливаются, профили сохраняют свою групповую принадлежность.
- **Завершение** — полное окончание кампании в указанную дату. Все профили исключаются из аудитории, а активности прекращают работу.

> **NOTE**
>
> Повторно активировать завершённую кампанию **невозможно**.

Подробнее о статусах кампании

Статус	Описание и влияние на активности
Неактивна	Присваивается после создания кампании и сохраняется до её запуска. Активности внутри кампании не выполняются.
Формирование аудитории	Проводится расчёт аудитории и распределение профилей между тестовой и контрольной группами. Пока идёт расчёт, броадкасты и регулярные рассылки приостанавливаются.
В ожидании	Кампания активирована с отложенным стартом. Запуск кампании начнётся автоматически в указанную дату.
Активна	Кампания запущена, аудитория рассчитана. Активности работают согласно своим настройкам.
Исключение аудитории	Система исключает профили из кампании и отвязывает их от сценариев. Новые рассылки не запускаются.
Завершена	Кампания полностью остановлена. Активности прекращают работу, **повторный запуск невозможен.**

Работа с профилями

Формирование тестовой и контрольной групп

После активации кампании начинается процесс **разметки аудитории** — система распределяет профили между тестовой группой (ТГ) и локальной контрольной группой (ЛКГ), если она настроена.

Если ЛКГ не используется, вся аудитория кампании становится **тестовой группой**

NOTE

Разметка аудитории запускается только **после активации** кампании.

Расширение аудитории

Расширение аудитории — это механизм, который позволяет автоматически добавлять в кампанию новых пользователей, если они участвуют в выбранных активностях.
Механизм помогает увеличивать охват кампании и поддерживать актуальность аудитории без ручного обновления сегментов.

INFO: Обратите внимание:

При использовании обновляемого сегмента в качестве основного сегмента кампании функция расширения аудитории недоступна. Новые профили попадают в аудиторию кампании только в результате пересчета сегмента.

При запуске активности с включённым расширением система проверяет, входят ли получатели в кампанию, и добавляет тех, кто ещё не участвует в ней. Новые профили автоматически распределяются между тестовой группой и локальной контрольной группой (если ЛКГ включена). Пользователи из тестовой группы получат активность, а участники ЛКГ — нет.

Примеры расширения аудитории

Сценарий без расширения аудитории

Кампания использует базу профилей `CRM_База` и сегмент `Постоянные клиенты` (100 профилей).
Создаётся броадкаст с сегментом `Новые регистрации` (50 профилей). Эти пользователи не входят в кампанию и не пересекаются с её основным сегментом.

При запуске броадкаста сообщение **не отправится никому**, потому что между сегментами нет пересечения — в кампании нет этих профилей.

Сценарий с расширением аудитории

Если добавить броадкаст в список **“Расширение аудитории”** в настройках кампании, система включит в кампанию всех получателей из сегмента `Новые регистрации`.

После этого профили автоматически распределятся между тестовой группой и локальной контрольной группой.

Пользователи из тестовой группы получат сообщение, а те, кто попал в ЛКГ, не получат активностей.

Таким образом, активность, добавленная в список расширения аудитории, не только выполняет отправку, но и пополняет кампанию новыми участниками.

Подробнее о настройке и работе механизма читайте в **отдельной статье о расширении аудитории**.

Ручное добавление профилей

INFO

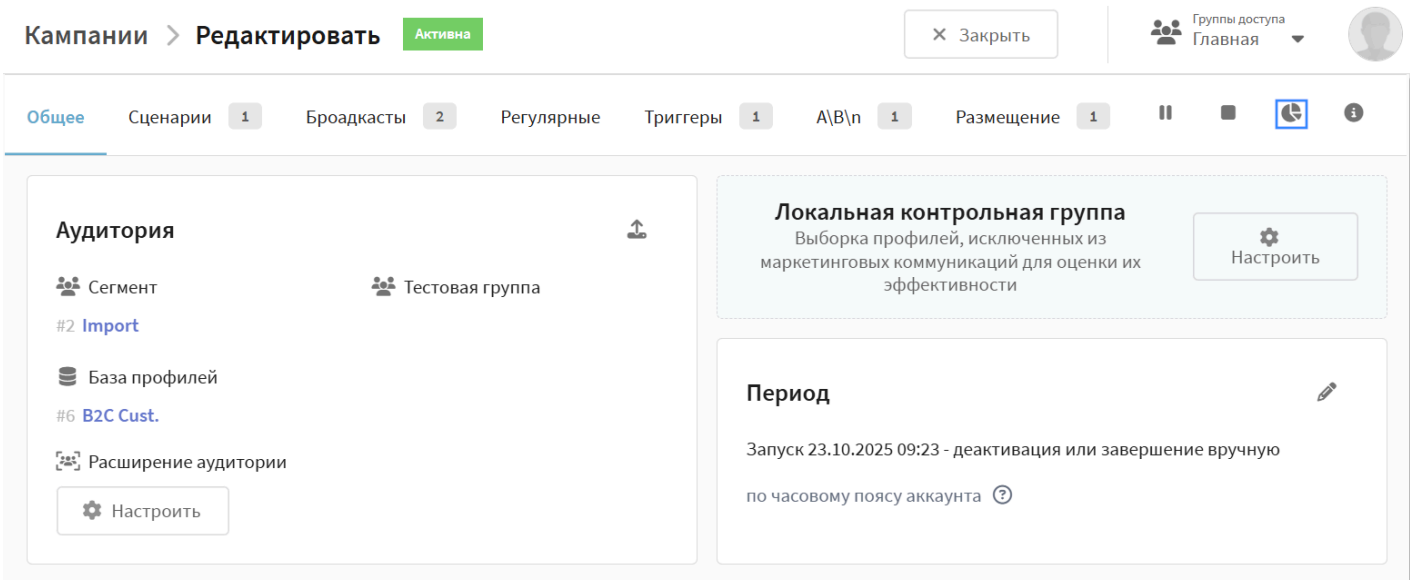
Ручное добавление профилей в аудиторию кампании доступно только для статических сегментов.

При необходимости отдельные профили можно вручную включить в кампанию, например для тестирования или проверки работы активностей. Добавление выполняется из **карточки профиля**: в блоке **“Взаимодействия”**, раздел **“Кампании”**, выберите нужную кампанию:

Чтобы профиль участвовал в активностях, установите флажок **“Отправить профиль в тестовую группу, даже если он находится в контрольной группе”** — в этом случае профиль будет принудительно помещен в тестовую группу, даже если состоял в локальной или глобальной контрольных группах.

Аналитика кампаний

Для просмотра отчета по кампании нажмите кнопку **“Аналитика”** в верхней панели редактора.



После нажатия система автоматически откроет раздел **"Аналитика** с уже применённым фильтром по выбранной кампании.

В этом отчёте можно просмотреть показатели по каждой рассылке и сценарию, входящим в кампанию:

Дополнительные возможности

Экспорт профилей

Доступен экспорт профилей из кампании с различными параметрами фильтрации. Функция находится в разделе **“Аудитория”**:

Доступные параметры экспорта:

- Фильтрация по статусу подписки
- Выбор аудитории (Тестовая группа, Локальная контрольная группа, Глобальная контрольная группа)
- Выбор полей для выгрузки (основные, поведенческие, системные, пользовательские)
- Настройка лимита количества профилей

INFO

Экспорт профилей доступен только после разметки аудитории

Подробнее о работе с экспортом данных в **этой статье**.

Клонирование кампании

Клонирование кампании создаёт её полную копию со всеми настройками и маркетинговыми активностями. Это полезно, когда нужно запустить похожую стратегию для другой аудитории или протестировать изменения на копии работающей кампании.

Чтобы клонировать кампанию, в списке кампаний нажмите иконку

“Клонировать” в контекстном меню нужной кампании.

После клонирования откроется редактор новой кампании, где вы можете изменить её параметры. При клонировании сохраняются все настройки оригинала, включая локальную контрольную группу, стратификацию, маркетинговые активности и расширение аудитории.

CAUTION: Обратите внимание

Базу данных изменить при клонировании нельзя — новая кампания будет использовать ту же базу, что и оригинал. Однако вы можете изменить сегмент аудитории для настройки тестовой группы.

Новая кампания создаётся в статусе “Неактивная” с неактивными маркетинговыми активностями. Для запуска потребуется отдельная активация после настройки всех параметров.

Удаление кампании

Удаление кампании перемещает её, а также все связанные с ней маркетинговые активности, в корзину. Удалить кампанию можно в статусе “Завершена” или “Неактивна” — при условии, что в ней нет размеченных профилей.

Чтобы удалить кампанию, откройте список кампаний и нажмите

“Удалить кампанию”. Затем подтвердите действие в открывшемся диалоговом окне — кампания будет перемещена в корзину.

CAUTION

После удаления кампания и все входящие в неё активности перемещаются в корзину.
Кампанию можно восстановить, но она будет возвращена в статусе **“Завершена”** и **не сможет быть повторно активирована**.

Локальные контрольные группы (ЛКГ)

Описание

Локальная контрольная группа (ЛКГ) — это часть аудитории **кампании**, которая не участвует в маркетинговых активностях. Она используется для более точного управления аудиторией и анализа влияния кампаний на получателей. В данный момент формирование отчета по ЛКГ в платформе недоступно.

Глобальная контрольная группа (ГКГ) действует для всей **базы профилей** и исключает выбранную часть профилей из **всех активностей на платформе.** Если профиль входит в ГКГ, он не сможет участвовать в кампании, кроме случаев, когда эта кампания добавлена в исключения. ГКГ настраивается в параметрах базы профилей. Подробнее — в **отдельной статье о ГКГ**.

Настройка локальной контрольной группы

Настройте локальную контрольную группу через виджет **“Локальная контрольная группа”** на главном экране редактора кампании, нажав кнопку **“Настроить”**:

Настройка размера группы

Укажите размер контрольной группы в процентах. Для точной настройки используйте дробные значения:

При активации опции **“Лимит аудитории”** появляется возможность задать **максимальное количество профилей** для контрольной группы. Фактический размер группы не превысит это значение, но может быть меньше в зависимости от процента и доступной аудитории.

> **INFO**
>
> Рекомендуемый размер ЛКГ — 5% от аудитории кампании. Слишком маленькая группа может не дать статистически значимых результатов, слишком большая — снижает охват кампании.

Стратификация

Описание

Стратификация — это метод случайного отбора профилей с предварительным разделением аудитории на подгруппы (страты) по заданным характеристикам. Аудитория разбивается на группы так, чтобы каждая из них отражала состав всей совокупности.

Из каждой группы случайным образом выбирается часть профилей, пропорциональная размеру этой группы. Это помогает сформировать контрольную группу с таким же распределением характеристик, как у тестовой, что делает результаты кампании более точными. Максимальное количество страт зависит от числа выбранных характеристик и количества групп внутри них. Эти ограничения можно настроить в системе.

Страта — это группа условий, выделенная по одному или нескольким критериям. Каждая уникальная комбинация критериев образует отдельную страту.

Поле — это критерий, по которому будут задаваться страты.

Пример стратификации

Допустим, маркетологу нужно разделить аудиторию по трём критериям:

- **Город**: Москва, Санкт-Петербург
- **Пол**: мужской, женский
- **Возраст**: 18-30 лет

Система создаст страты по всем комбинациям критериев:

- Мужчины 18-30 лет из Москвы
- Мужчины 18-30 лет из Санкт-Петербурга
- Женщины 18-30 лет из Москвы
- Женщины 18-30 лет из Санкт-Петербурга

Расчёт доли страт: Предположим, в вашей целевой аудитории:

- 50% мужчин и 50% женщин
- 40% людей 18-30 лет
- 60% из Москвы, 40% из Санкт-Петербурга

Тогда доля страты “Мужчины 18-30 лет из Москвы” **от всей аудитории** составит: 0.5 (пол) × 0.4 (возраст) × 0.6 (город) = 0.12 **(12%)**

В контрольную группу профили отбираются из каждой страты пропорционально её рассчитанной доле.

Настройка стратификации

Настройка стратификации находится в том же окне, что и основные параметры ЛКГ, и доступна через выпадающее меню

Задайте страты по одному или нескольким полям, определив условия для каждой группы: например, **диапазоны дат**, **числовые значения**, **группы элементов** из списка и т.д. Настройка стратификации осуществляется через пользовательский интерфейс, где можно последовательно добавлять страты и задавать параметры для каждой из них:

Поддерживаемые типы полей

Для стратификации могут использоваться как основные, так и дополнительные поля следующих типов:

- **enum** — список фиксированных значений
- **bool** — логическое значение
- **int/float** — числовые значения
- **string** — строковые значения
- **date** — дата

Условия для типов полей

Для каждого типа поля доступны различные условия:

- **Число** — условие отбора “в диапазоне” с полями “от” и “до”
- **Число с точкой** — условие “в диапазоне” с выбором “включительно/не включительно”
- **Строка** — условие отбора “равно” с возможностью ввода списка значений

- **Boolean** — условия по значениям “истина”, “ложь”, “отсутствует”
- **Список** — условие отбора “равно” с возможностью множественного выбора
- **Дата** — условия “в диапазоне относительная” (год/месяц/день) и “в диапазоне” (точные даты)

Ограничения стратификации

- **Минимальное количество условий** — 2 в каждом поле
- **Максимальное количество условий** — 64 на всю стратификацию
- **Максимальное количество полей данных** — 32

Лимит рассчитывается перемножением количества условий в выбранных полях. Например: поле “Город” с 3 условиями × поле “Возраст” с 2 условиями = 6 комбинаций.

TIP: Обратите внимание

Значения в пределах одного поля не должны полностью пересекаться:

- **Даты** — диапазоны в пределах одного поля не должны пересекаться
- **Числа** — диапазоны не должны полностью перекрывать друг друга
- **Строки/списки** — не допускается полное дублирование значений

Рекомендации по количеству страт

- **Сегмент до 100 тыс** — 5-10 страт с использованием ключевых метрик (пол, возраст, регион)
- **Сегмент от 100 тыс до 1 млн** — 10-20 страт с использованием ключевых и поведенческих метрик
- **Сегмент от 1 млн** — 20-30 страт с глубокой сегментацией по демографии, поведению и RFM

Рекомендуемое количество условий в одном поле — от 2 до 4. Более 5 условий может привести к слишком узким стратам.

Игнорирование маркетинговых активностей

Описание

Настройка **“Игнорирование маркетинговых активностей”** позволяет выбрать конкретные активности внутри кампании, которые будут отправлять сообщения всем профилям, включая участников контрольной группы. Это полезно для критически важных уведомлений, которые должны доходить до всей аудитории независимо от участия в ЛКГ.

TIP: Примеры использования

- **Критически важные уведомления** — транзакционные emails, уведомления о статусе заказа
- **Юридические коммуникации** — уведомления об изменении условий, политики конфиденциальности
- **Технические уведомления** — сообщения о технических работах, обновлениях системы
- **Экстренные коммуникации** — важные оповещения, касающиеся всех пользователей

Настройка игнорирования

Настройка игнорирования активностей находится в том же окне настроек ЛКГ и доступна через выпадающее меню:

Для редактирования списка активностей, игнорирующих ЛКГ, в настройках ЛКГ нажмите

“Редактировать”:

Процесс разметки аудитории

После настройки параметров ЛКГ при активации кампании система запускает процесс разметки аудитории. Также этот процесс запускается при изменении настроек ЛКГ в уже активной кампании. В это время:

- Профили распределяются между тестовой и контрольной группами
- Рассылки приостанавливаются до завершения пересчета
- Кампании присваивается дополнительный статус **“Формирование аудитории”**, при этом её основной статус сохраняется

Обратите внимание, что при включении ЛКГ в кампании, которая уже была активна, состав контрольной группы будет формироваться только из профилей, попавших в кампанию после ее активации, например в результате расширения аудитории. Профили, попавшие в тестовую группе при первой разметке, остаются в тестовой группе.

> **CAUTION: Внимание**
>
> Если на момент запуска пересчета аудитории в системе были активны броадкасты или регулярные рассылки, они будут приостановлены до его окончания.

Управление ЛКГ

После настройки ЛКГ в интерфейсе кампании доступны следующие действия:

- **Редактировать настройки** — изменение параметров ЛКГ, стратификации и исключений;
- **Удалить ЛКГ** — полное удаление локальной контрольной группы. **Вместе с ЛКГ будут удалены все связанные настройки** (Стратификация и Игнорирование маркетинговых активностей). Операция запускает пересчёт аудитории, в результате которого профили из контрольной группы будут перемещены в тестовую;
- **Экспорт профилей** — возможность выгрузки участников ЛКГ. После завершения формирования аудитории при настройке экспорта профилей в фильтрах можно выбрать участников локальной контрольной группы для выгрузки. Экспорт доступен внутри виджета **“Аудитория”**. Подробнее про экспорт профилей Кампании вы можете узнать здесь.

Особенности работы ЛКГ при пересчете сегментов

При использовании обновляемого сегмента в качестве аудитории кампании, или при пересчете аудитории статического сегмента механизм ЛКГ имеет ряд особенностей:

Разметка новых профилей Разметка аудитории запускается автоматически после каждого пересчета сегмента. Система распределяет между тестовой и контрольной группами только новые профили, добавленные в сегмент. Профили, уже состоящие в кампании, сохраняют свою группу.

Выход профилей из кампании Если профиль перестает соответствовать условиям сегмента, при следующем пересчете он автоматически удаляется из кампании. При этом:

- Профиль получает событие «Выход из кампании»
- Все активные сценарии кампании для этого профиля деактивируются
- Если профиль позже снова начнет соответствовать условиям, он будет добавлен в кампанию как новый и пройдет повторную разметку

Изменение настроек ЛКГ При изменении параметров контрольной группы (процента или стратификации) новые настройки применяются только к профилям, которые будут добавлены в кампанию после следующих пересчетов сегмента. Уже распределенные профили не меняют свою группу.

Расширение аудитории в кампании

Описание

Расширение аудитории — это механизм, который автоматически добавляет в кампанию новые профили из сегментов активностей, входящих в неё. Механизм расширения помогает увеличивать охват кампании и поддерживать актуальность аудитории без ручного обновления исходного сегмента.

Каждая кампания создаётся на основе **статического сегмента**, который определяет целевую аудиторию. Однако внутри кампании каждая активность может быть настроена на **любой сегмент**, даже если он не пересекается с основным.

Например, кампания ориентирована на сегмент **“Постоянные клиенты”**, но внутри неё создаётся рассылка, нацеленная на сегмент **“Новые регистрации”**. Без расширения аудитории такие пользователи не получат сообщение, потому что их нет в основной аудитории кампании. Если же для этой рассылки включено **расширение аудитории**, система автоматически добавит получателей из сегмента **“Новые регистрации”** в кампанию и распределит их между **тестовой группой** и **локальной контрольной группой**, если она включена.

INFO: Ключевые ограничения

- **Только активности этой кампании** — нельзя выбрать рассылку или сценарий из других кампаний
- **Только незавершённые активности** — нельзя выбрать активности в статусе “Завершена”
- **Только профили из Базы Профилей кампании** — даже если сегмент активности включает профили из других баз, в аудиторию кампанию попадут только профили из её целевой базы
- **Только новые профили** — добавляются только те профили, которых ещё нет в аудитории кампании

Как работает расширение аудитории

Механизм расширения аудитории позволяет динамически пополнять кампанию новыми профилями непосредственно из аудитории запускаемых активностей. Его работу можно описать следующими ключевыми принципами:

- При запуске активности система определяет аудиторию этой активности (сегмент или всю базу) и автоматически добавляет в аудиторию кампании профили, которые в ней ещё не состоят;
- Новые участники сразу распределяются между тестовой и контрольной группой кампании;
- Пользователи, не из базы профилей кампании, сообщение не получат;
- Если профиль попадает в сегменты нескольких активностей **расширения аудитории** одной кампании, он будет добавлен в её аудиторию только **один раз**. Его группа (тестовая или контрольная) будет закреплена в момент первого добавления. Это исключает повторное попадание профилей в аудиторию кампании и предотвращает их случайный переход из тестовой группы в контрольную при срабатывании последующих активностей;
- Расширение аудитории работает только для активных кампаний.

CAUTION

Так как расширение аудитории добавляет профили в статический сегмент кампании (её исходную аудиторию), последующий перерасчёт этого сегмента приведёт к их исключению. Все профили, попавшие в кампанию через расширение аудитории, будут удалены из её аудитории при обновлении статического сегмента. Во избежание этого **не рекомендуется** включать функцию **“Обновлять сегмент ежедневно”** для выбранного сегмента кампании.

Процесс добавления профилей

Для каждого профиля в аудитории запускаемой активности система проверяет:

- **Принадлежность к Базе Профилей кампании:** Профиль должен входить в ту же базу профилей, что и кампания. Если нет — он не попадет в аудиторию кампании и не получит сообщение.
- **Участие в кампании:** Проверяется, состоит ли профиль в аудитории кампании в момент запуска активности.
    - Если **да** — он получит сообщение (если не состоит в ЛКГ).
    - Если **нет** — профиль распределится в тестовую или контрольную группу согласно настройкам локальной контрольной группы. Профили, попавшие в тестовую группу получат сообщение, попавшие в контрольную группу — не получат.

Пример использования

Сценарий без расширения аудитории

Кампания использует базу профилей `CRM_База` и сегмент `Постоянные клиенты` (100 профилей).
Создаётся броадкаст с сегментом `Новые регистрации` (50 профилей). Эти пользователи не входят в кампанию и не пересекаются с её основным сегментом.

При запуске броадкаста сообщение **не отправится никому**, потому что между сегментами нет пересечения — в кампании нет этих профилей.

Сценарий с расширением аудитории

Если добавить броадкаст в список **“Расширение аудитории”** в настройках кампании, система включит в кампанию всех получателей из сегмента `Новые регистрации`.
После этого профили автоматически распределятся между тестовой группой и локальной контрольной группой.

Настройка расширения аудитории

Настройка расширения аудитории находится в разделе **“Аудитория”** редактора кампании в блоке **“Расширение аудитории”**:

Выбор активностей

В блоке “Расширение аудитории” доступны для выбора различные типы активностей:

- **Броадкасты** — разовые массовые рассылки

- **Регулярные рассылки** — периодические коммуникации по расписанию
- **Триггерные рассылки** — автоматические отправки по событиям
- **A/B-тесты** — сравнение эффективности вариантов
- **Сценарии** — многошаговые автоматизации
- **Размещения** — инструмент для размещения контента

TIP

Вы можете выбрать несколько активностей. Каждая из них будет независимо добавлять новых пользователей в кампанию при своём запуске.

Разметка аудитории в кампаниях

Описание процесса

Формирование аудитории — это процесс распределения профилей кампании между тестовой и контрольной группами согласно заданным настройкам.

Система обрабатывает только профили, принадлежащие к базе данных, выбранной при создании кампании. Если в статический сегмент добавлены профили из разных баз, кампания будет учитывать только тех, кто относится к её базе.

Процесс запускается в следующих случаях:

- **При активации кампании** — первоначальное распределение профилей согласно настройкам локальной контрольной группы (ЛКГ)
- **При изменении параметров активной кампании** — редактирование настроек ЛКГ запускает пересчет распределения профилей
- **При пересчете сегмента** — новые профили, добавленные в сегмент, проходят “доразметку”. Этот процесс происходит в фоновом режиме.

Работа с неактивными кампаниями

В неактивной кампании действия с профилями имеют особенности. При удалении профиля **из сегмента:**

- Событие **“Выход из сегмента”** фиксируется в карточке профиля сразу
- Событие **Выхода из кампании** появляется только после активации кампании

Если профиль удалили из сегмента, а затем вернули обратно до активации кампании, система при разметке учитывает только его конечное состояние — нахождение в сегменте. События входа и выхода **из кампании** для таких промежуточных действий **не фиксируются**, однако в карточке профиля сохраняется полная история входов и выходов **из сегмента**.

Изменения ЛКГ в активных кампаниях

Включение Локальной контрольной группы в уже активной кампании запустит формирование аудитории, в результате чего активные рассылки по расписанию (броадкасты, регулярные) будут приостановлены. При этом в ЛКГ смогут попасть только новые профили, попавшие в аудиторию кампании (например, через расширение аудитории).

Удаление ЛКГ в активной кампании вызывает перераспределение аудитории — все профили из ЛКГ переводятся в тестовую группу.

Завершение кампании

Когда кампания завершается, система автоматически добавляет всем участвовавшим профилям событие **“Выход из кампании”**.

Сценарии автоматизации

Сценарии автоматизации

Сценарии — программы автоматизации маркетинговых процессов. С помощью сценариев можно управлять лояльностью ваших клиентов, мотивировать их совершить целевое действие за определённый период времени, проводить комплексные интерактивные мероприятия, а также реализовывать такие маркетинговые процессы, как напоминание об брошенной корзине, приветственная цепочка, реактивация и ретаргетинг.

Сценарий может выполняться:

- для каждого клиента индивидуально при наступлении определённого события — так же, как триггерная рассылка.
- для определённой группы клиентов (сегмента) по заданному расписанию — аналогично регулярной рассылки.

В этом разделе

Работа со Сценариями автоматизации

INFO

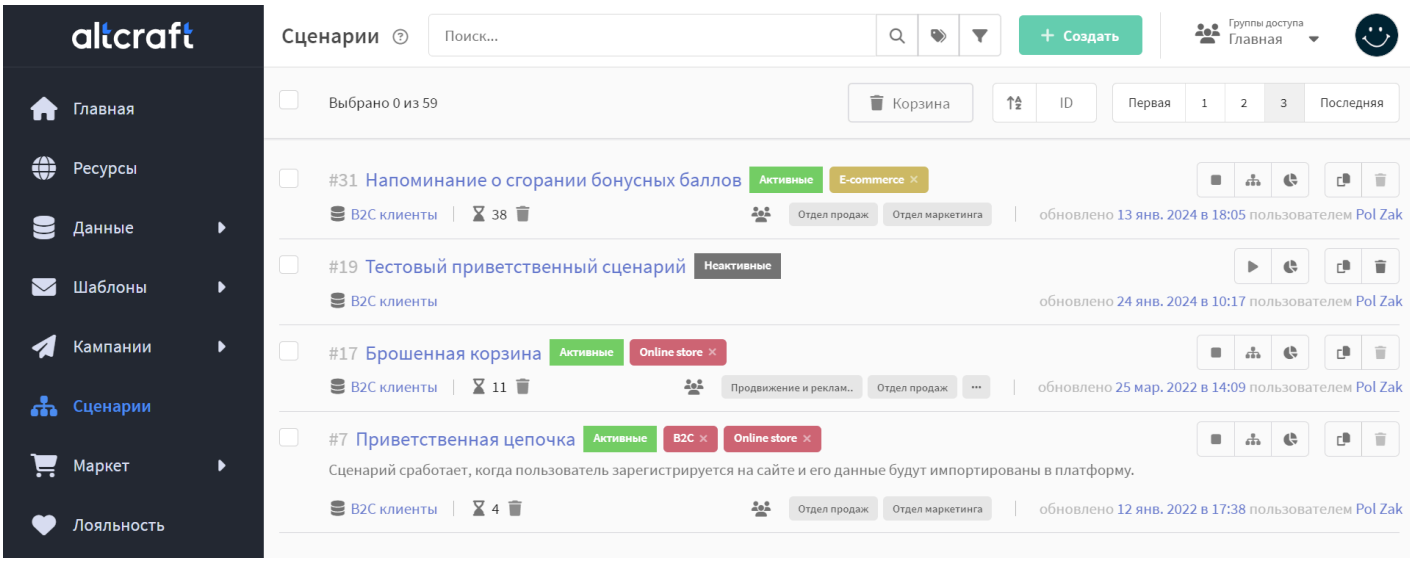
Элементы (шаги) сценария описаны **в отдельной статье**.

INFO

Сценариями автоматизации можно управлять по **API**. Информация доступна по **этой ссылке**.

Управление сценариями

Чтобы начать работу со сценариями взаимодействия с клиентами, в главном меню выберите **Сценарии**. В открывшемся списке отображается краткая информация по существующим сценариям, их статусы активности, а также количество профилей, которые в данный момент находятся в сценариях.



С помощью контекстного меню вы можете:

— активировать сценарий.

— остановить сценарий. Если в сценарии на момент остановки остались профили, их можно сохранить. При следующей активации сценария они “проснуться” и продолжат проходить сценарий с того элемента, на котором остановились.

— отправить профили из динамического сегмента в сценарий (в сегменте должна использоваться та же база профилей, что и в сценарии).

— клонировать сценарий.

— переместить сценарий в корзину (доступно только для неактивных сценариев).

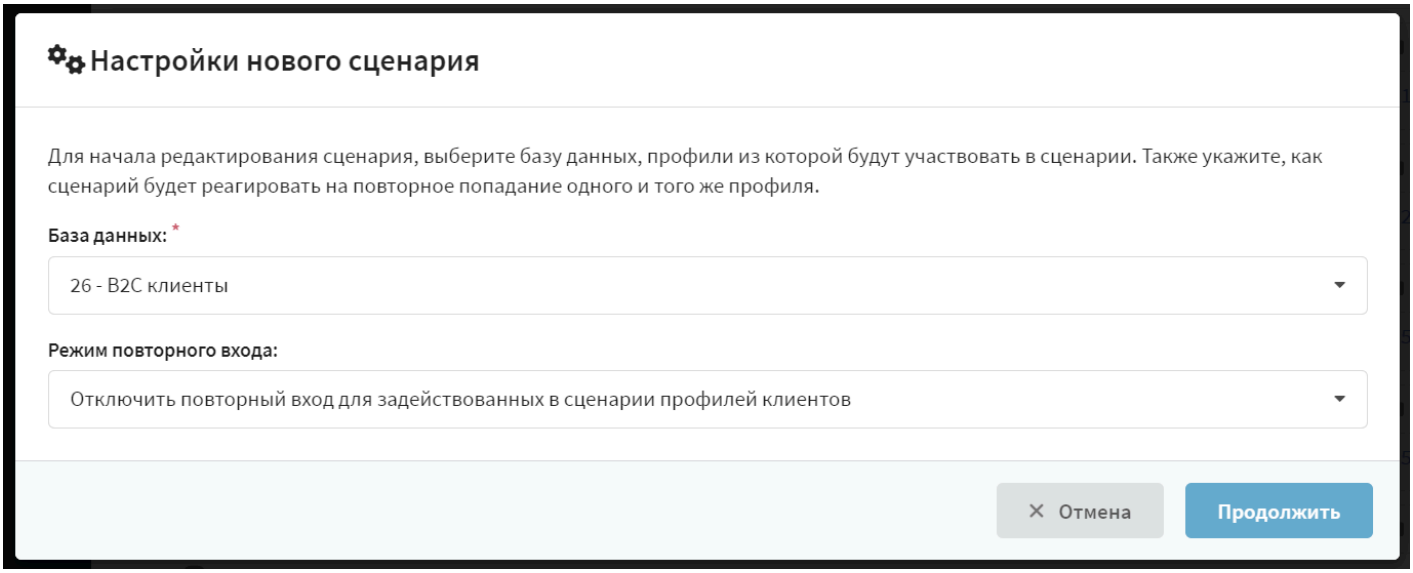
— перейти в отчет по сценарию.

Начало работы со сценарием

Для добавления нового сценария нажмите **+ Создать** в верхней части страницы.

Предварительная настройки

В открывшемся окне выберите **базу данных**, профили в которой будут захвачены в сценарий, и **режим повторного входа в сценарий**. Затем нажмите кнопку **“Продолжить”**. Позже эти настройки можно будет изменить.



Режим повторного входа определяет поведение сценария **при неоднократном захвате** профиля клиента. Выберите один из режимов:

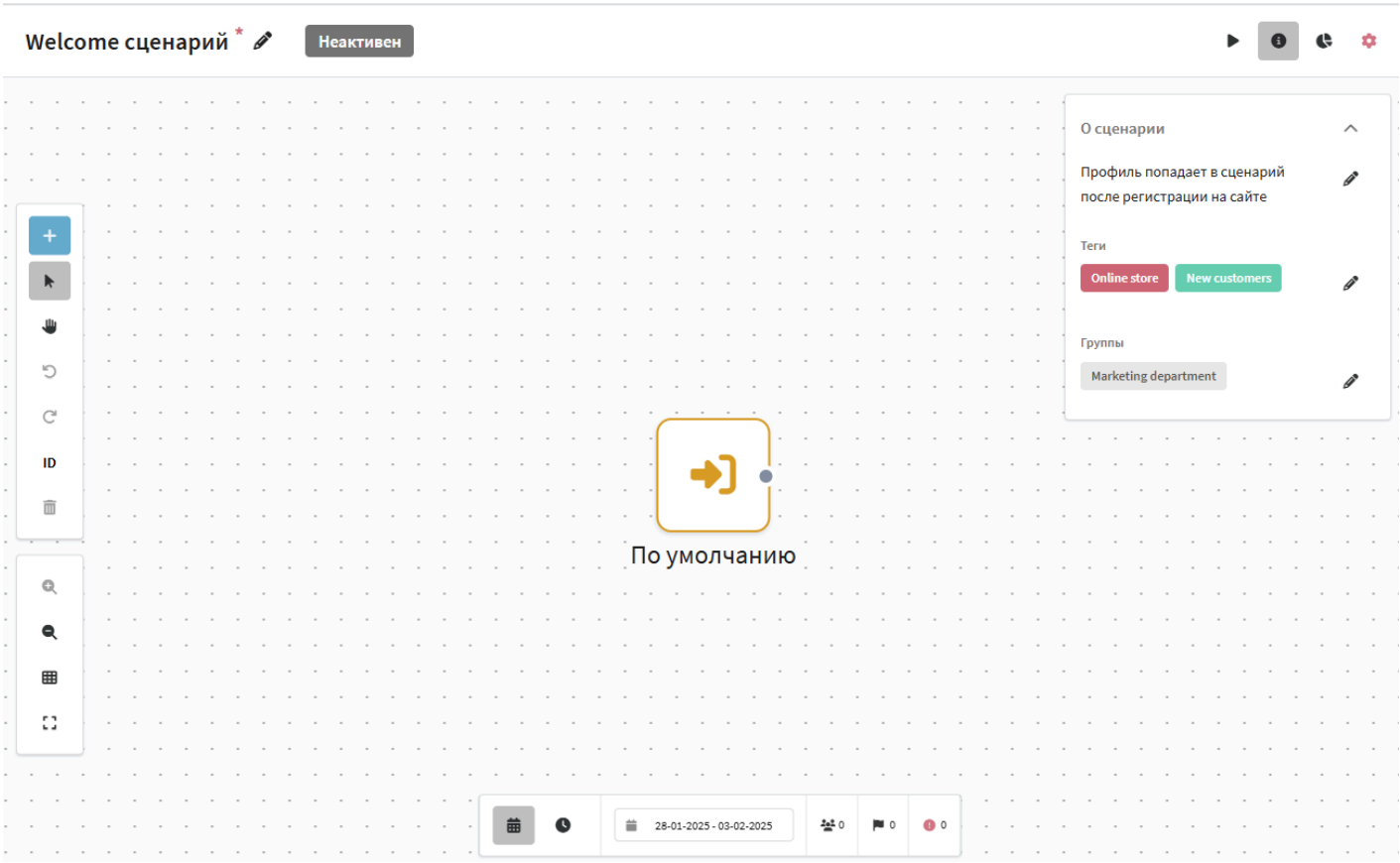
- **Отключить повторный вход в сценарий для профиля клиента** — в этом случае профиль может только один раз попасть в сценарий, и даже если он из него выйдет, то не сможет попасть в него снова. Никогда.
- **Отключить повторный вход для задействованных в сценарии профилей клиента** — если профиль в данный момент уже в сценарии, он не может попасть в него еще раз, но как только выйдет — сможет.
- **Прекратить существующее задание для профиля при появлении нового** — если профиль клиента повторно захватывается сценарием, его текущее задание в сценарии будет немедленно завершено и заменено новым. Задание — это индивидуальный процесс прохождения сценария отдельным профилем. Все данные, связанные с текущим заданием (`apicontent` и т.д.), будут удалены. Профиль начнёт прохождение сценария с самого начала, в рамках нового задания.
- **Обрабатывать каждую запись профиля в сценарии отдельно** — независимо от того, сколько раз профиль присутствует в сценарии, он попадет в него снова, когда выполнится условие захвата. При этом оба попадания будут обрабатываться полностью.

Основные настройки

Вы перейдете в редактор. На верхней панели укажите **название** сценария. В блоке **“О сценарии”** можно добавить **описание**, прикрепить **теги** для быстрого поиска и назначить **группы** доступа, а также прикрепить **атрибуты** и указать их **значения**.

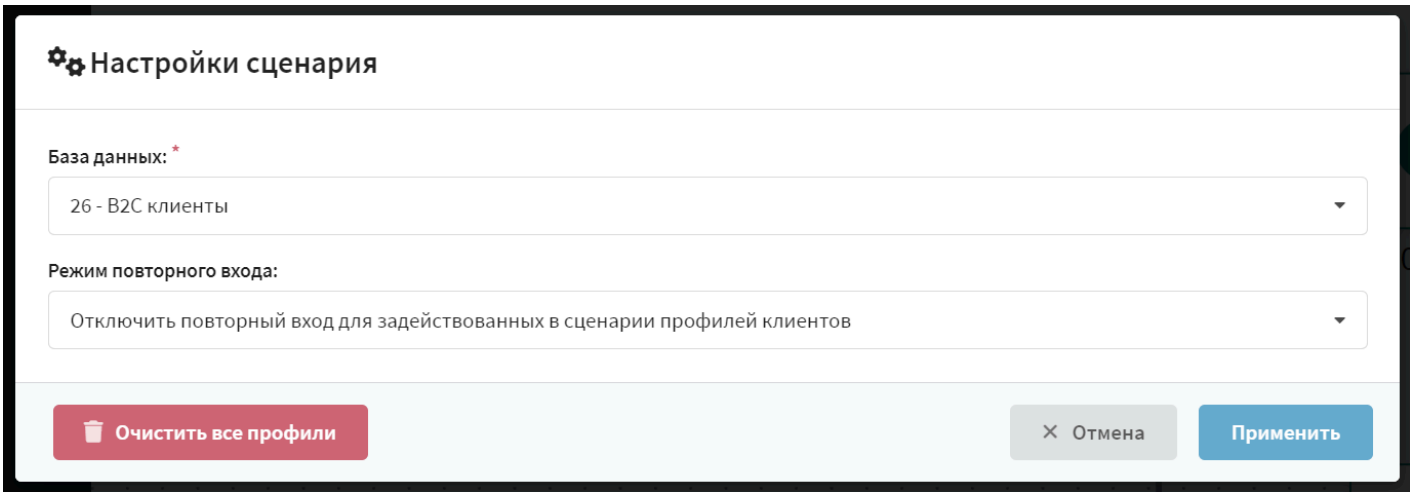
Чтобы отобразить или скрыть блок **“О сценарии”**, используйте кнопку

на панели инструментов:



Если вы хотите изменить базу профилей в сценарии или выбрать другой режим повторного входа, то на верхней панели нажмите

и отредактируйте настройки. В этом же окне доступна опция **“Очистить все профили”** — она позволяет удалить из сценария профили, которые находятся там в настоящий момент.

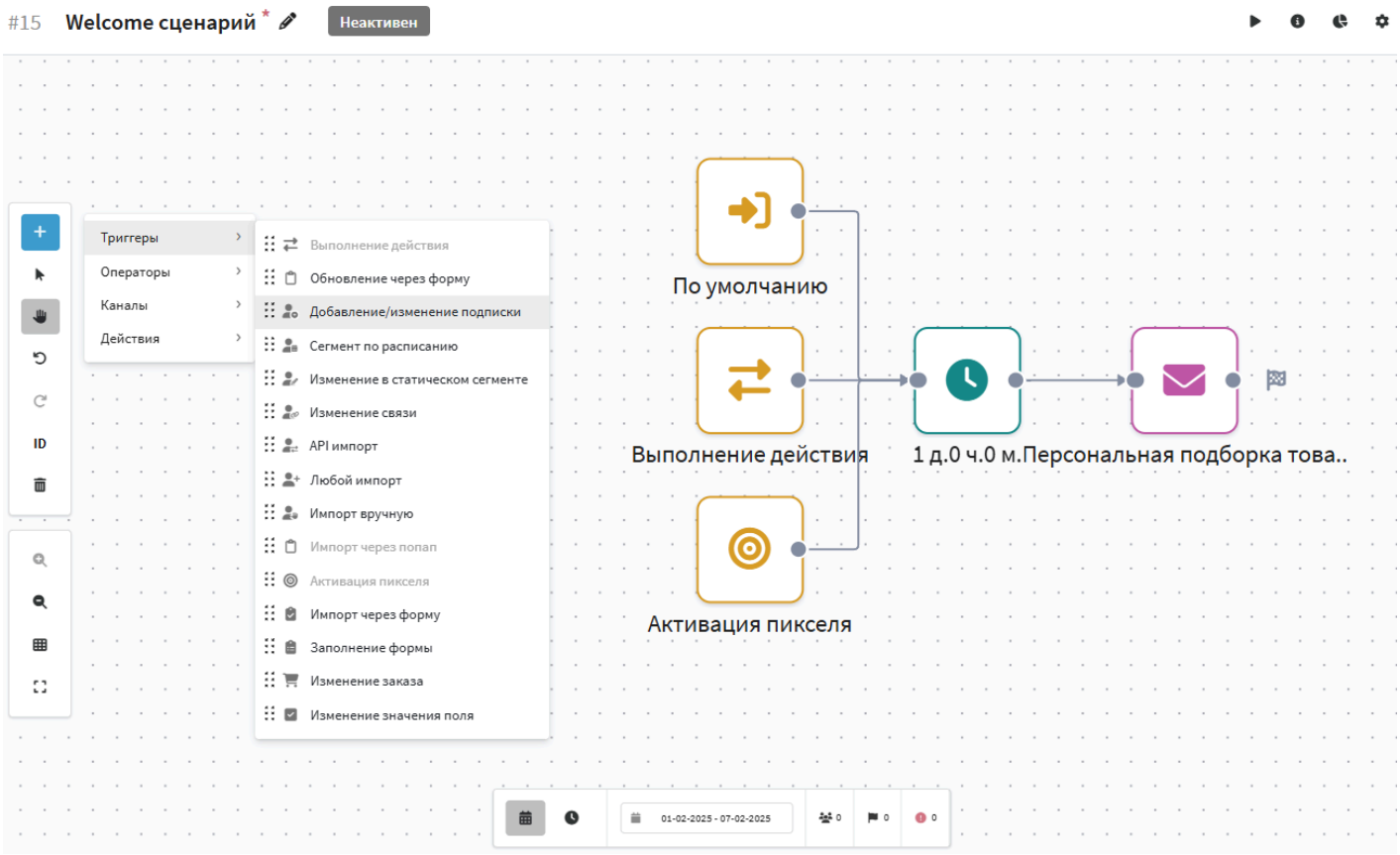


В случае, если для сценариев активирован какой-либо **обязательный атрибут**, то вам необходимо сразу выбрать для него значение.

Настройка захвата профилей в сценарий

Условия попадания профиля в сценарий определяет триггер. **Триггер** — это **стартовый элемент**, который запускает сценарий. В одном сценарии вы можете настроить **до 10 триггеров**.

Чтобы добавить триггер, нажмите **“+ Элемент”** и выберите нужный элемент из категории **“Триггеры”**:



TIP

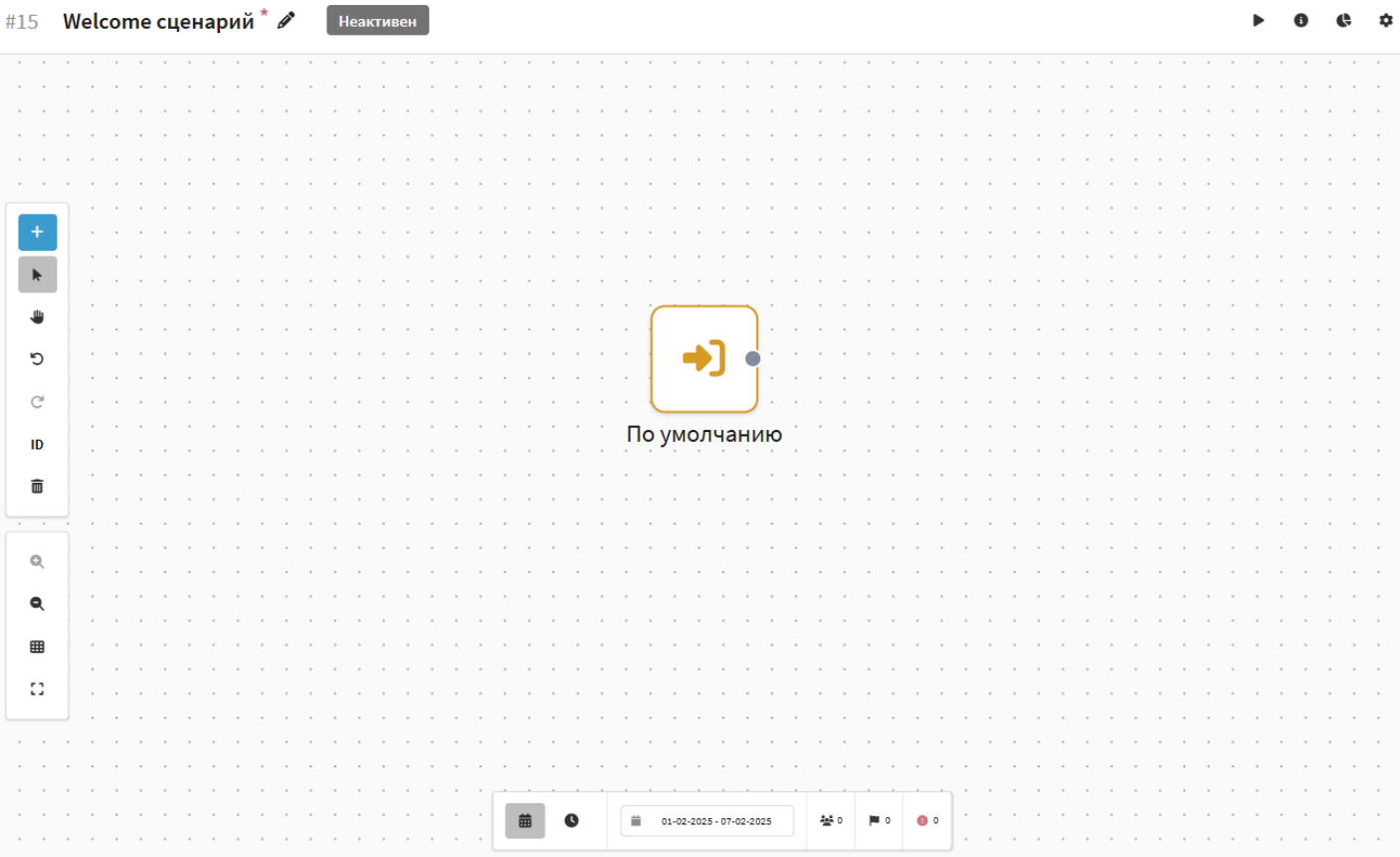
Подробную информацию по каждому триггеру можно найти в [этой статье](#).

CAUTION

Чтобы **сохранить сценарий**, необходимо связать все стартовые элементы с каким-либо элементом.

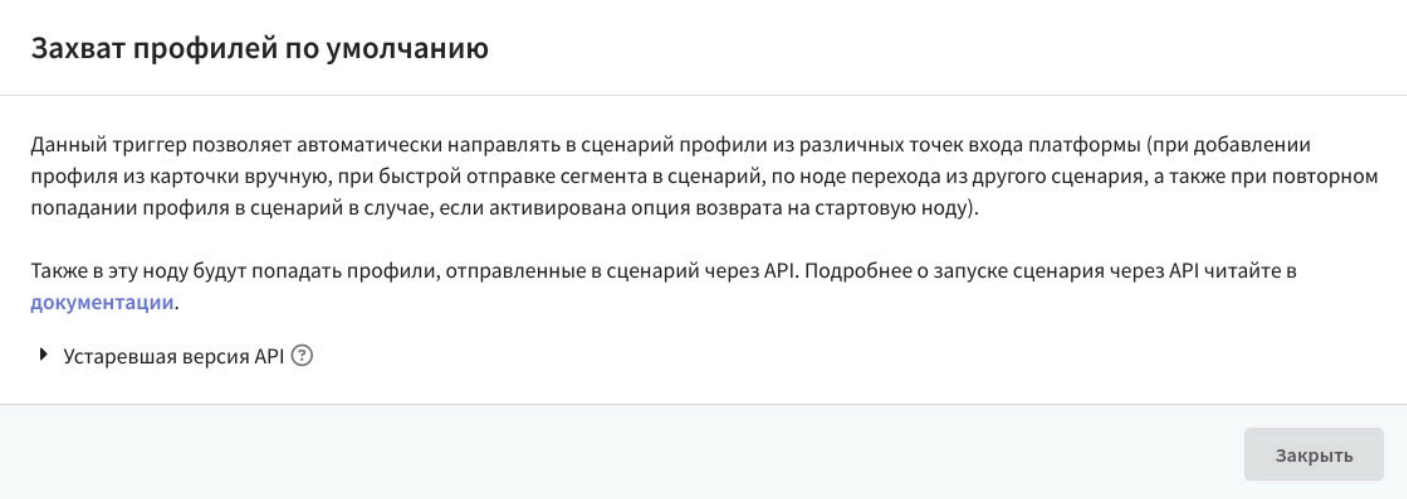
Захват профилей по умолчанию

В любом сценарии есть элемент **“Захват профилей по умолчанию”**. Этот элемент является встроенной функцией сценария и **не подлежит удалению**. Дважды кликните на него, чтобы прочитать описание.

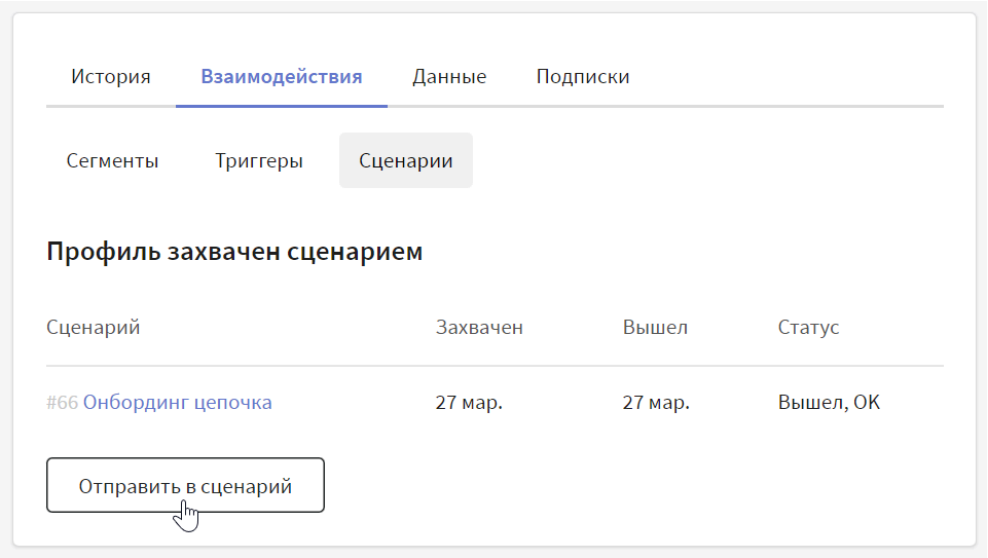


С помощью элемента “По умолчанию” профили могут попадать в сценарий из разных точек платформы:

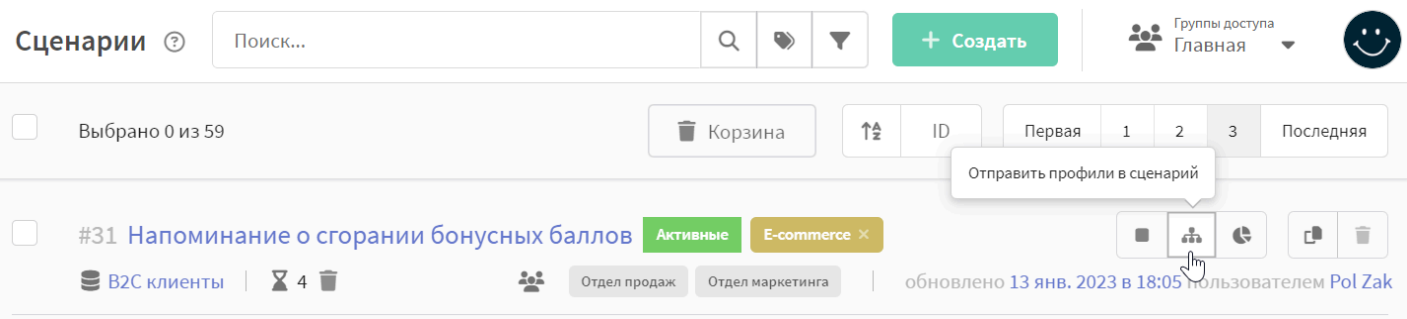
1. При отправке профиля в сценарий через API. После сохранения сценария в окне элемента отобразится URL, который можно использовать для запуска сценария. Однако рекомендуется использовать более новую версию API. Подробнее в **документации**.



2. При добавлении профиля в сценарий вручную:



3. При быстрой отправке сегмента в сценарий:

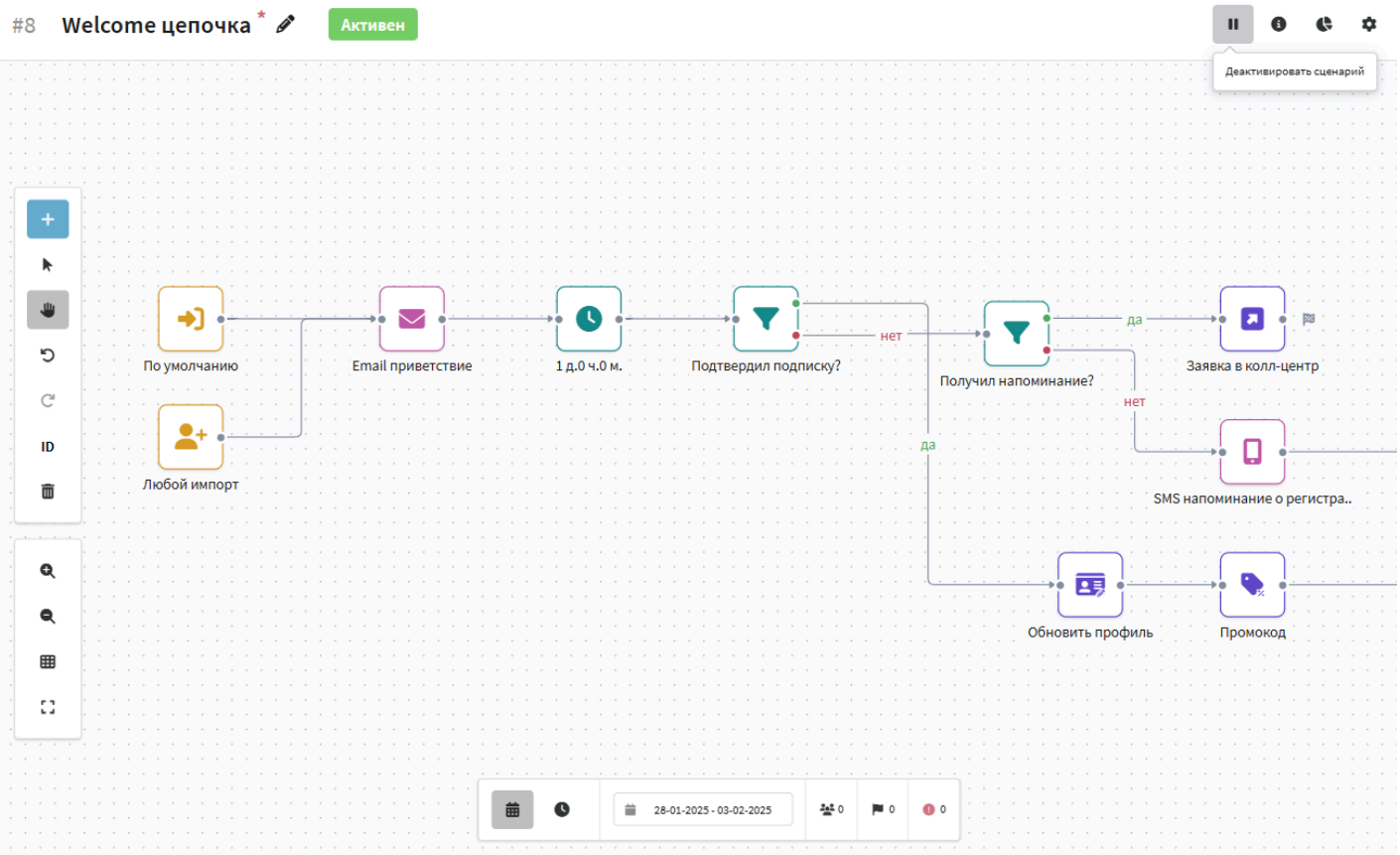


4. Через [узел перехода] из другого сценария:

Активация и деактивация сценария

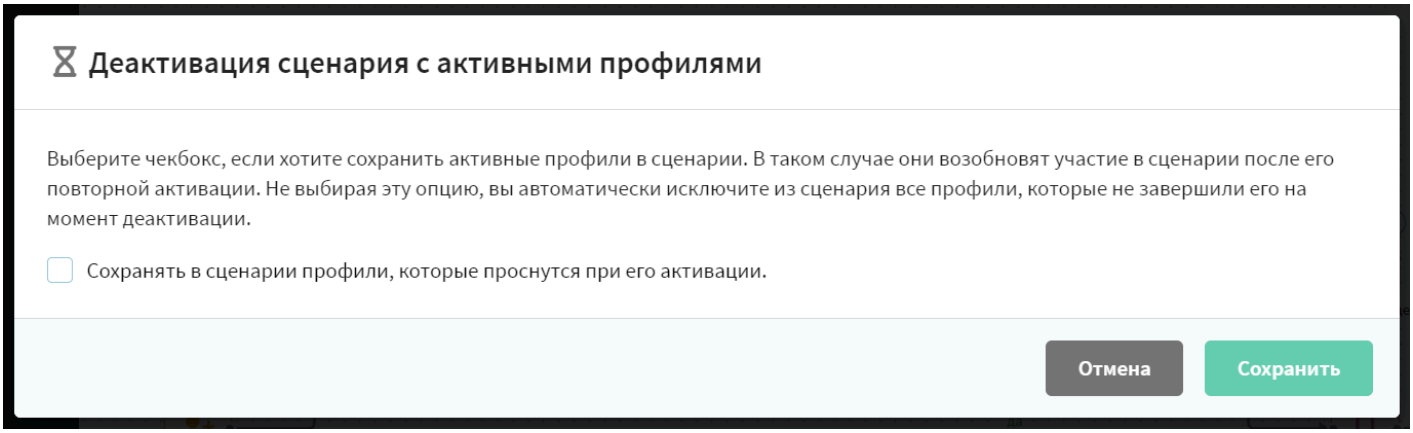
Чтобы активировать сценарий, нажмите кнопку

, а затем сохраните сценарий. После активации сценарий начнёт захватывать профили, согласно заданным условиям.



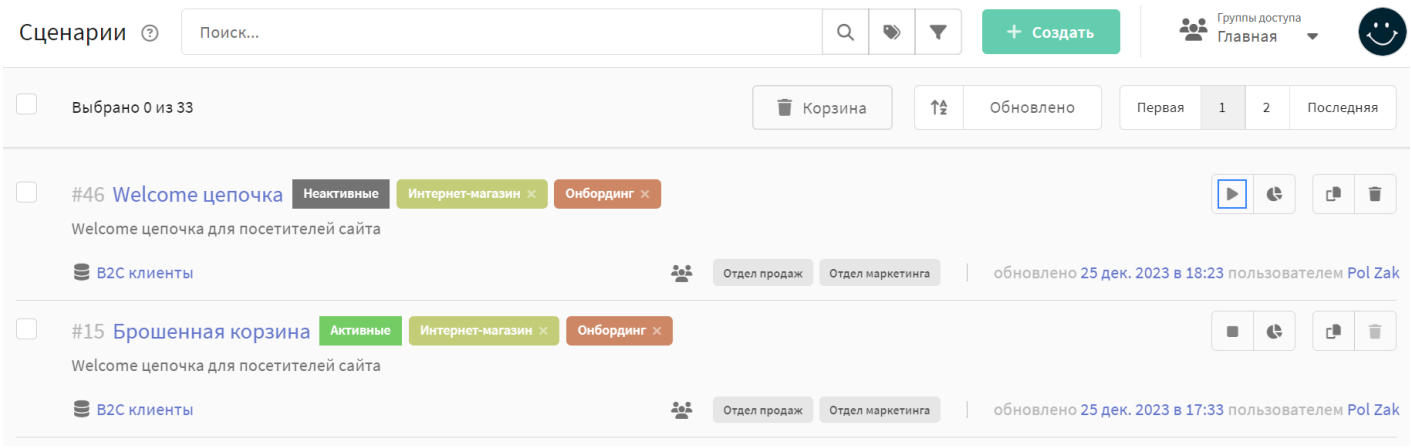
Чтобы остановить сценарий, нажмите кнопку

и сохраните сценарий. Если в сценарии на момент деактивации находятся профили, то вы увидите окно с предупреждением:



По умолчанию после деактивации сценария все профили выйдут из него. Чтобы избежать этого, включите настройку **“Сохранять в сценарии профили, которые проснуться при его активации”**. В таком случае все профили в сценарии будут заморожены, и после повторной активации сценария они продолжат идти по нему.

Также активировать и деактивировать сценарий можно из общего списка:

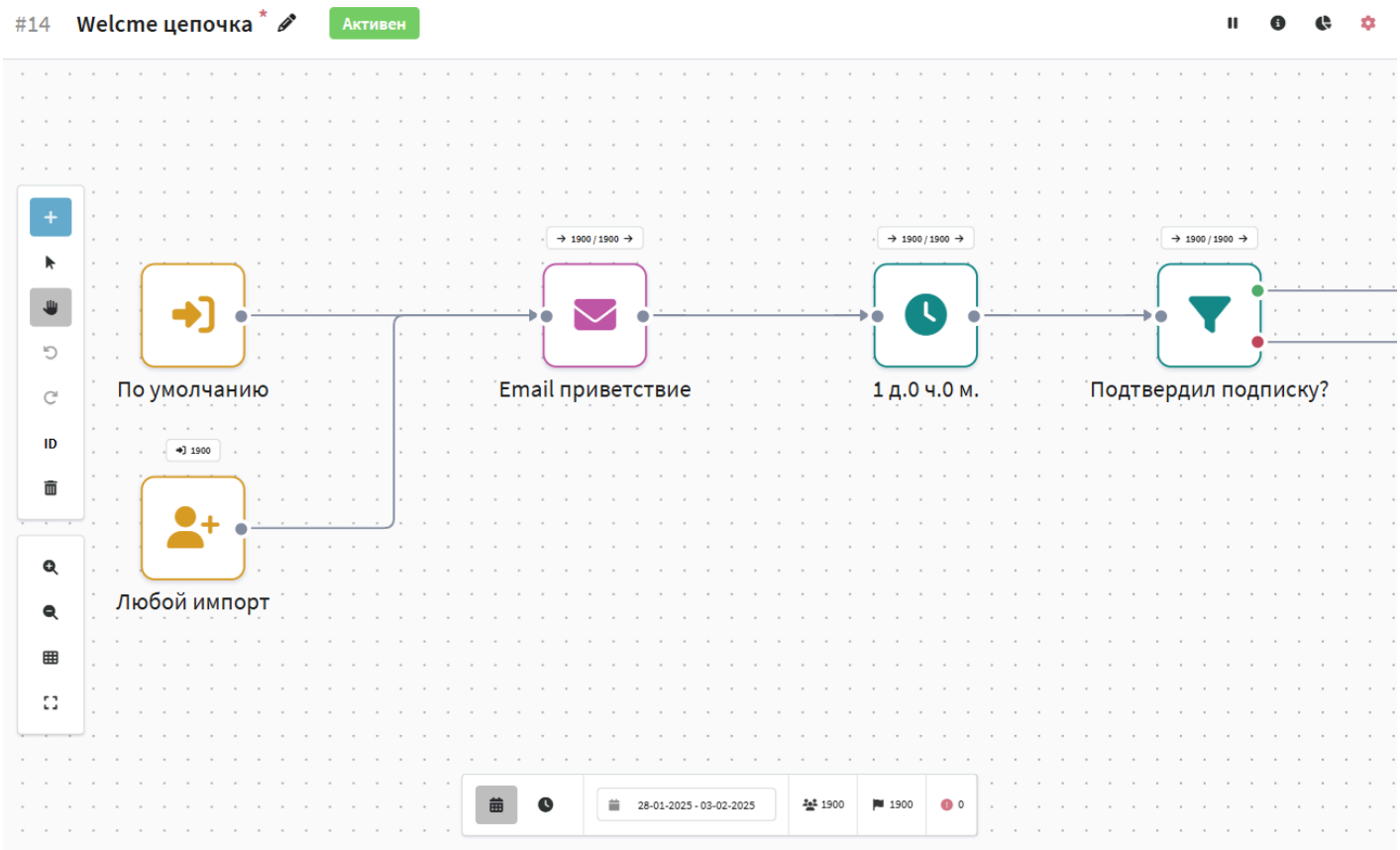


Аналитика сценариев

Анализировать эффективность ваших сценариев вы можете непосредственно в редакторе сценария, а увидеть подробную статистику по рассылкам, использованным в цепочке, позволит **отчёт по каналам**.

Статистика прохождения сценария клиентами

Панель статистики находится в нижней части редактора. На этой панели собрана информация о прохождении сценария профилями.



Просмотр статистики доступен в двух режимах:

— статистика за период

— статистика в реальном времени

Статистика за период

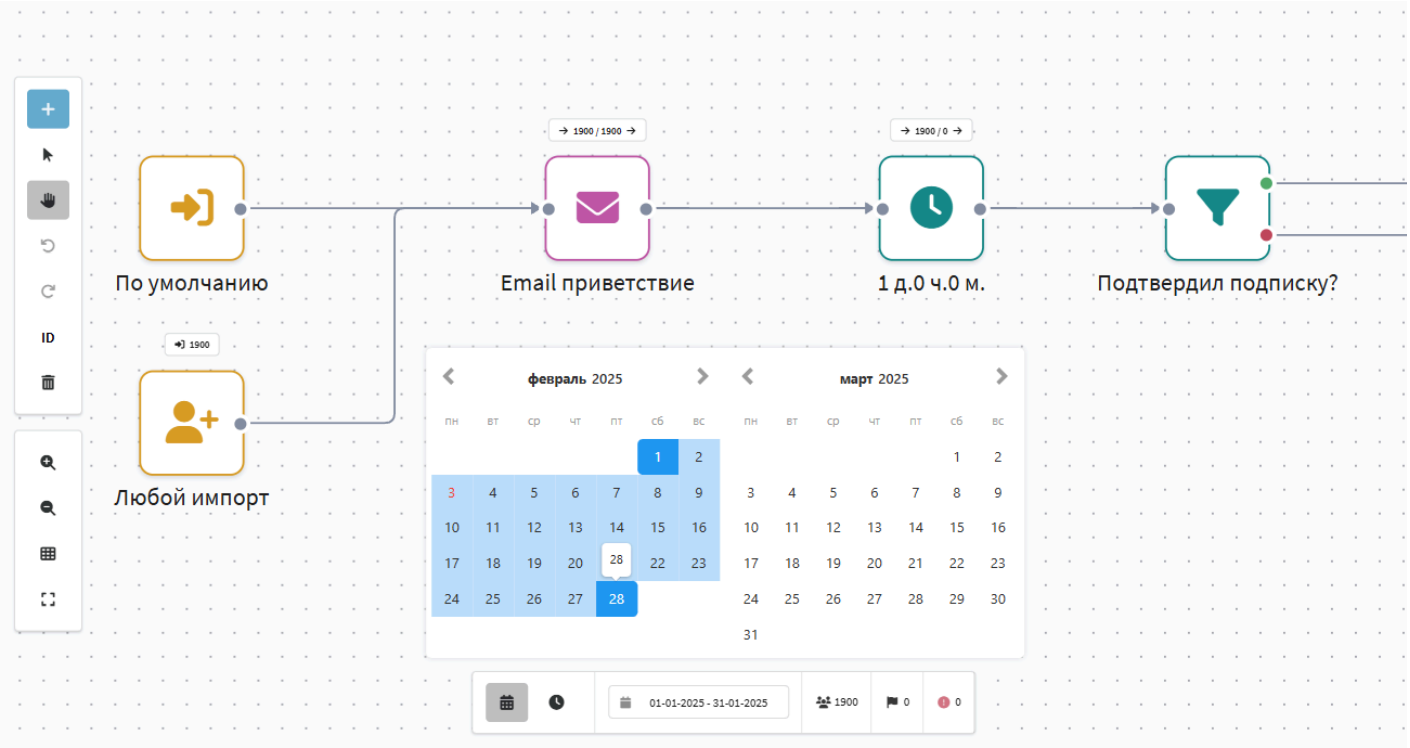
Здесь вы можете просмотреть данные по сценарию за определенный период времени:

— общее количество профилей, вошедших в сценарий.

— количество профилей, завершивших сценарий.

— ошибки, возникшие в процессе работы сценария.

По умолчанию панель отображает данные за последние 7 дней. Чтобы изменить период, введите нужные даты вручную или выберите их в календаре:



Также над каждым элементов сценария показано, сколько профилей вошло в элемент и сколько вышло.

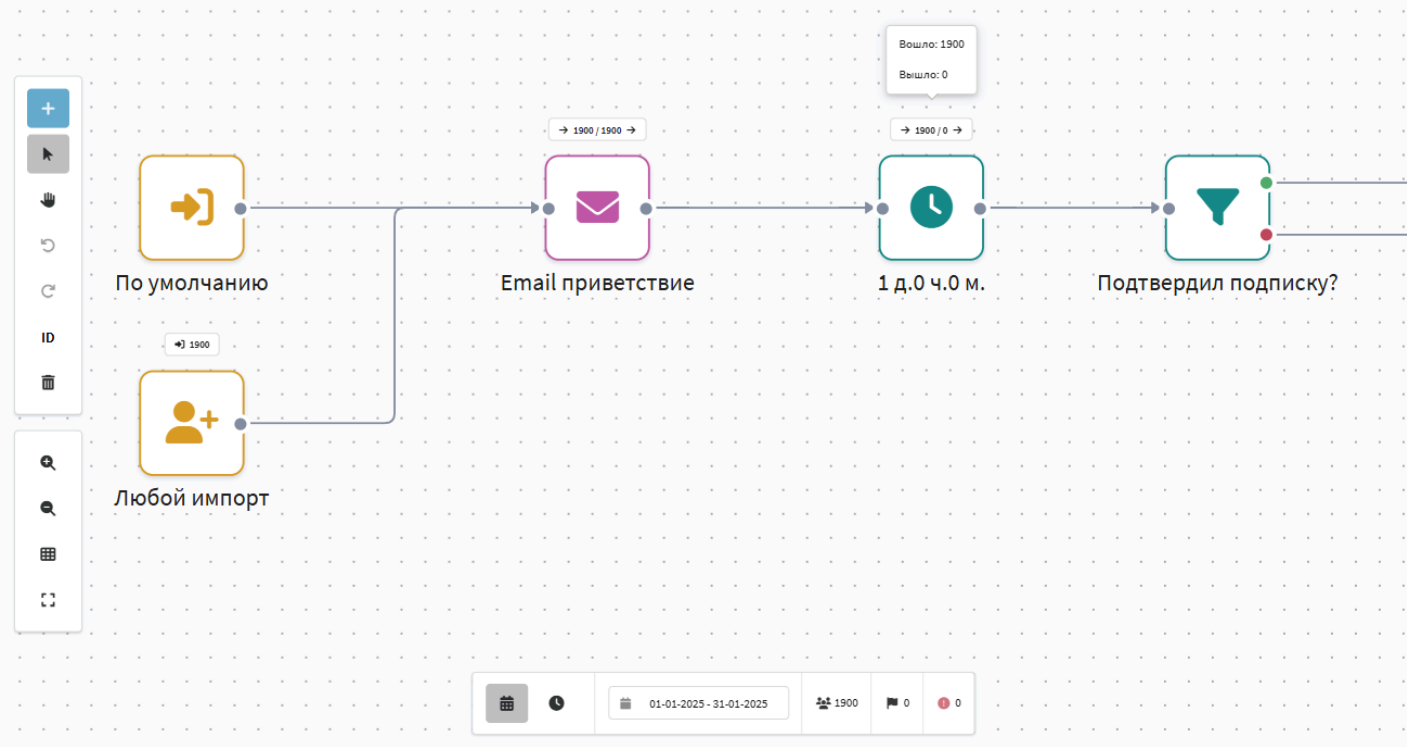
TIP

Чтобы узнать, на каком именно элементе профиль покинул сценарий, откройте карточку профиля, перейдите на вкладку **Взаимодействия —> Сценарии** и наведите курсор на статус “Вышел”. Отобразится идентификатор элемента:

Статистика в реальном времени

Здесь вы можете получить информацию о профилях, которые сейчас находятся в сценарии.

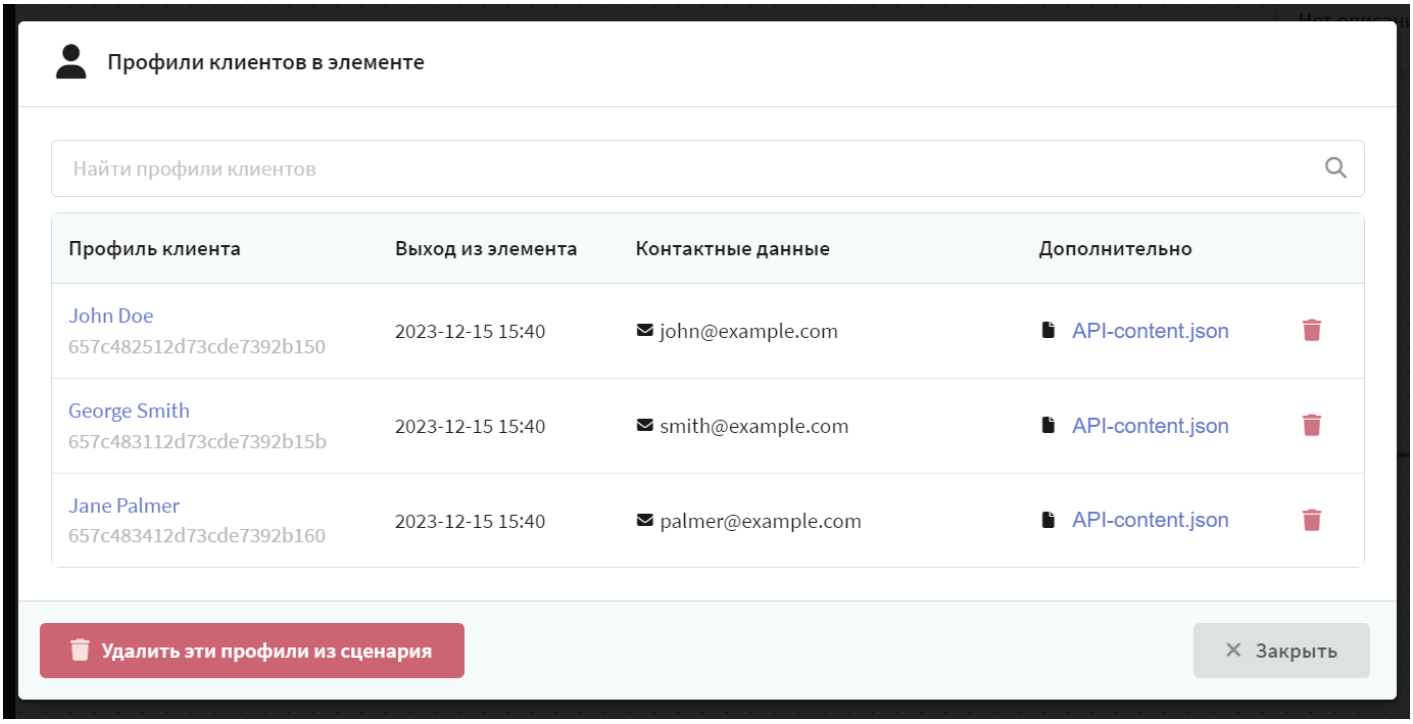
— количество профилей, вошедших в сценарий, но еще не завершивших его.



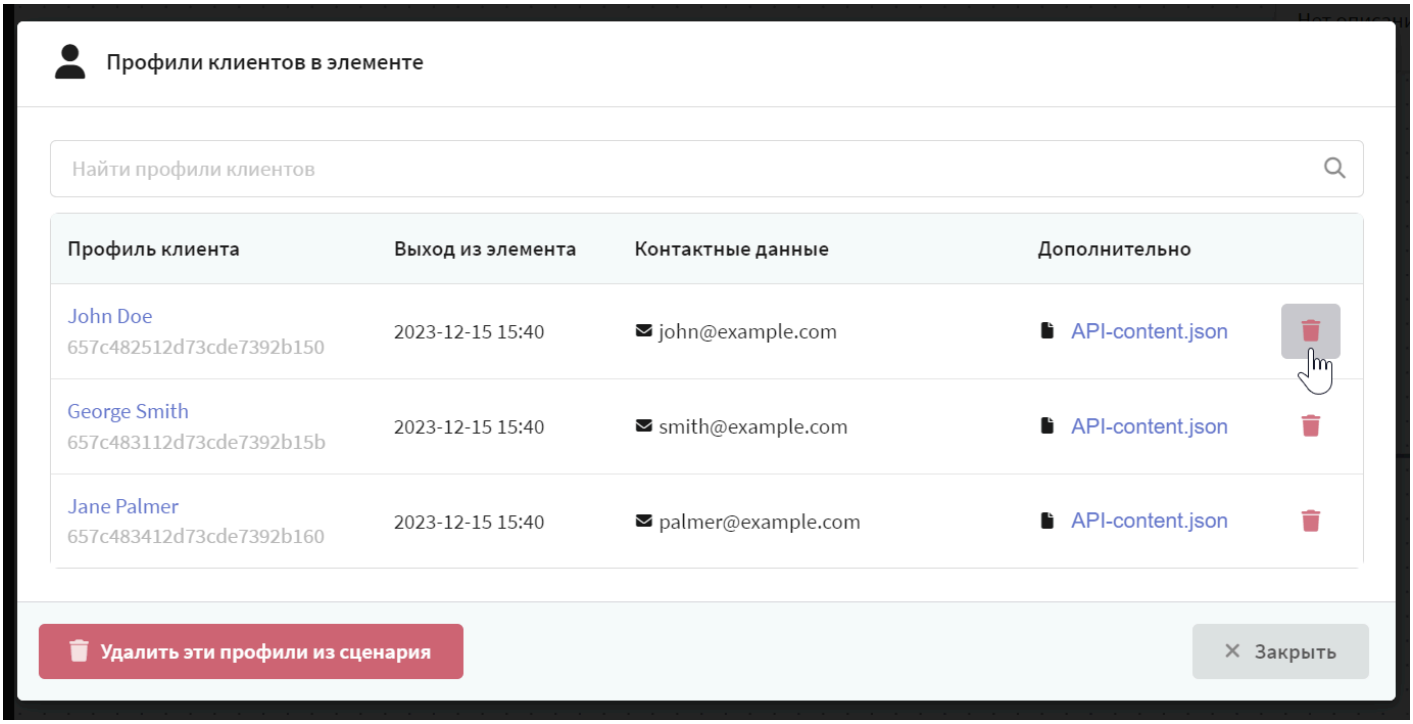
Если в сценарии в данный момент есть профили, то вы увидите цифру над элементом, в котором они находятся. Цифра показывает количество профилей в элементе. Таких элементов может быть несколько. В большинстве случаев это элементы ожидания (Пауза, Лучшее время доставки или Условие с задержкой).

Чтобы узнать информацию о профилях в элементе, нажмите на эту цифру. Откроется модальное окно, в котором можно:

- перейти в профиль клиента в базе;
- узнать точное время, когда профиль выйдет из элемента;
- скачать файл с API-контентом, если соответствующий параметр был передан в **API-запросе**.



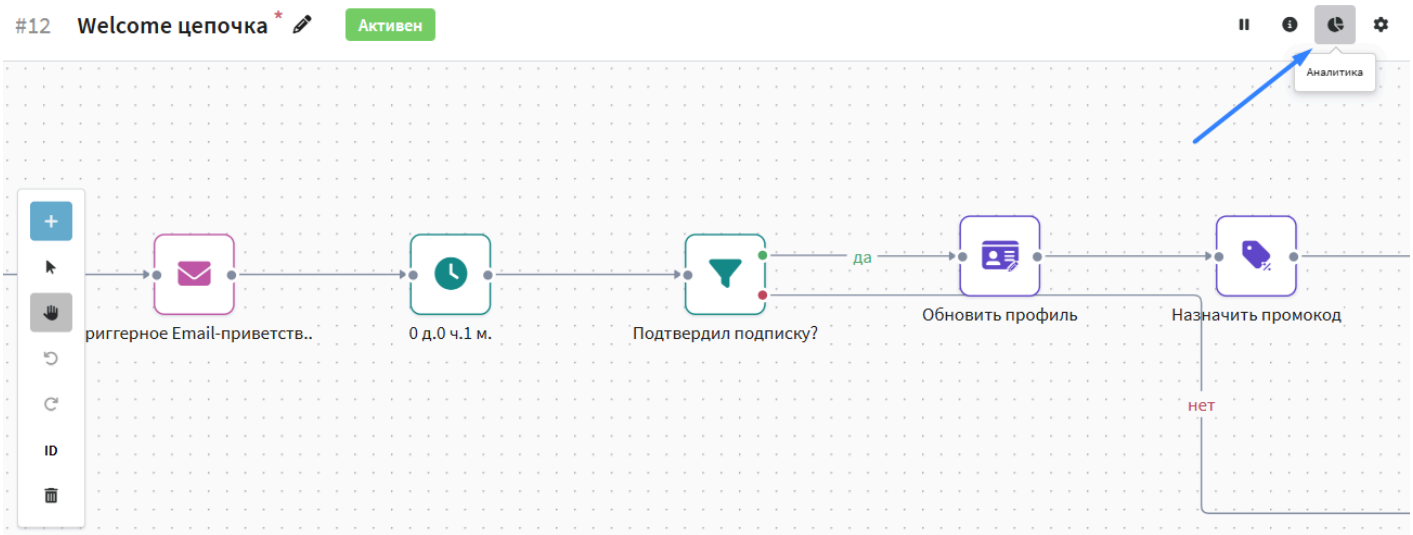
Также доступно **удаление профилей из сценария**. Если вы хотите удалить определенный профиль из сценария, нажмите на иконку корзины напротив его имени:



Чтобы удалить из сценария все профили, которые в данный момент находятся в этом элементе, нажмите кнопку **“Удалить эти профили из сценария”**.

Аналитика рассылок в сценарии

Чтобы просмотреть отчёт о рассылках, которые запускает сценарий, откройте **Отчёт по каналам**:



Используйте группировки и фильтры, чтобы отобразить нужные вам данные. Например, если вы хотите посмотреть статистику рассылок, задействованных в сценарии, вам подойдет такая настройка:

#50 Welcome цепочка *

Сохранить

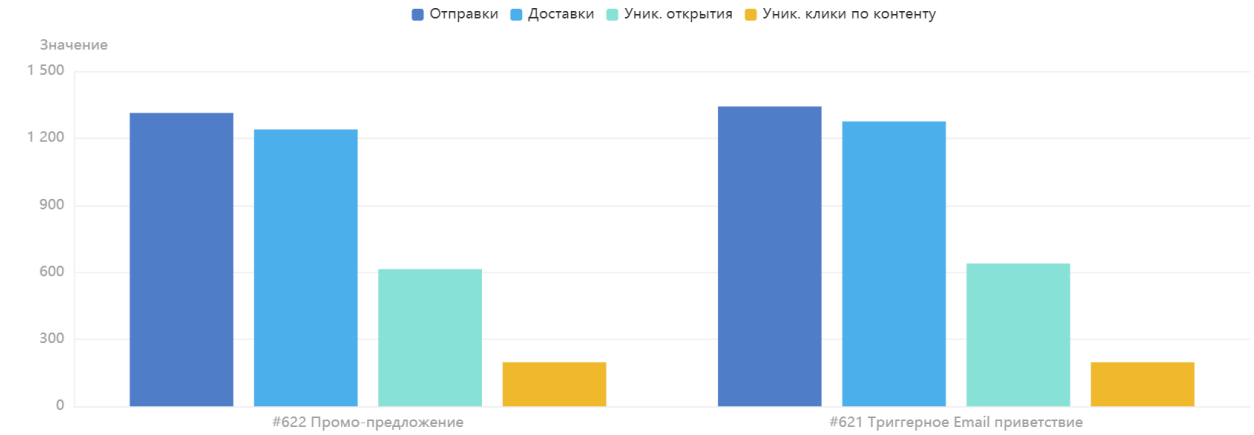
Отчет по: **Email** За период: Последние 7 дней

Группировать по: Рассылкам в которых показать: +

Включить цепочку #50 Welcome цепочка × | Добавить фильтр

Скрыть диаграмму

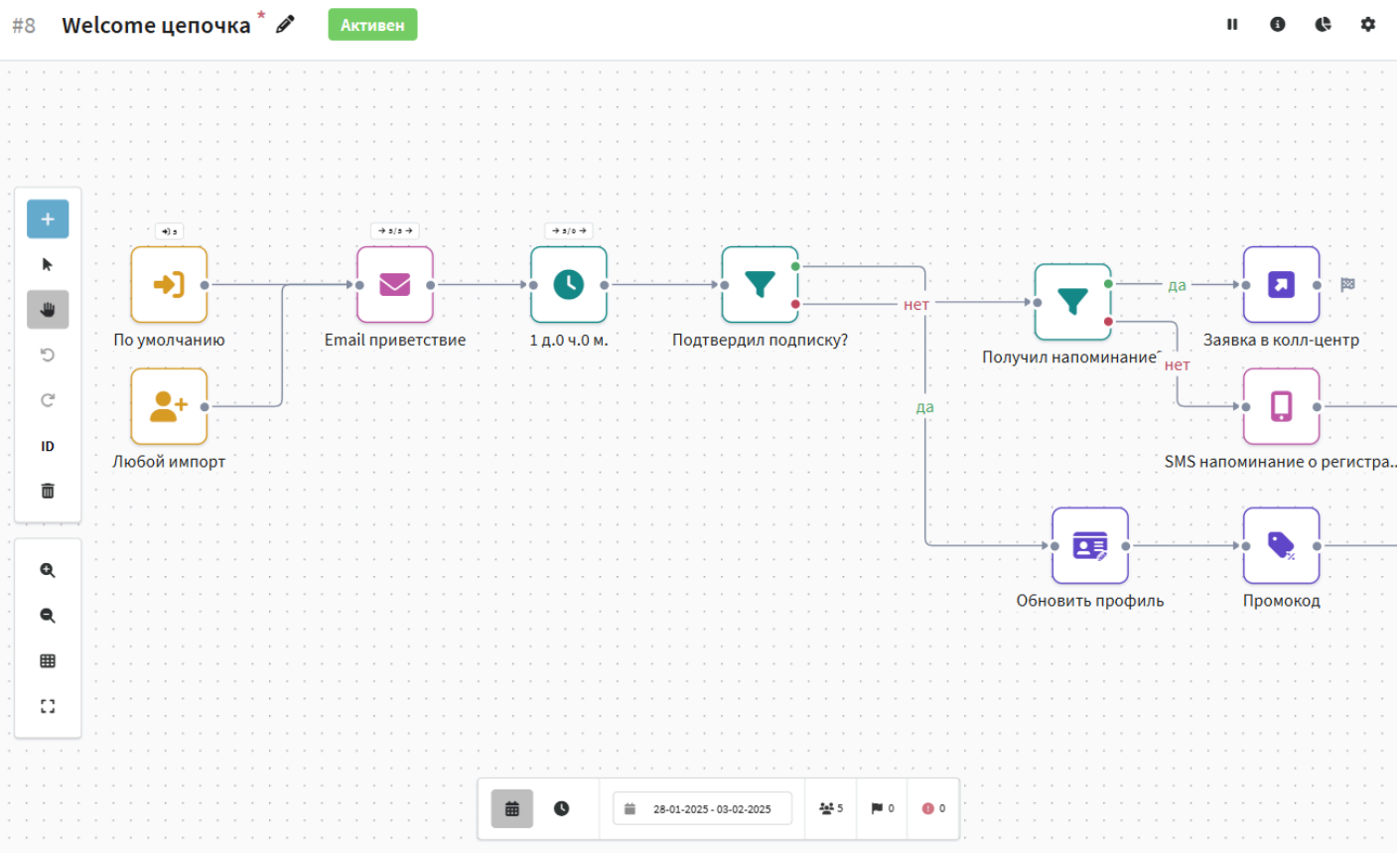
Кампании



Наименование	% Доставок	Открытия	Прочтения	Клик по контенту	Уник. открытия	OR контента
Всего	94,69 %	1 253	1 022	2 758	1 253	49,84 %
#622 Промо-предложение	94,36 %	614	496	1 379	614	49,56 %
#621 Триггерное Email приветствие	95,01 %	639	526	1 379	639	50,12 %

Узлы сценария

Сценарий состоит из **узлов** — специальных элементов, отвечающих за каждый шаг сценария. Каждый узел выполняет одно действие и затем передаёт управление следующему.

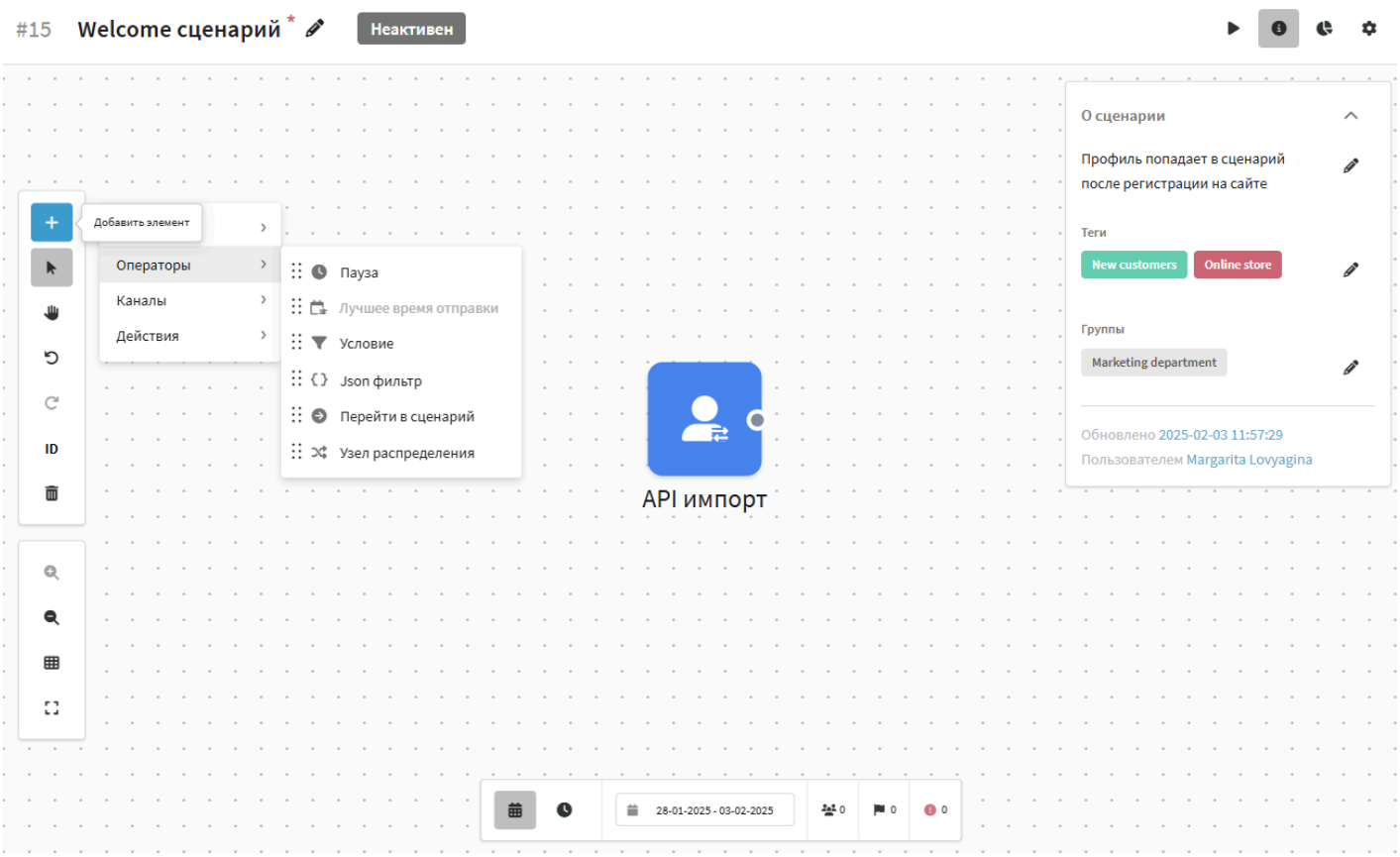


Любой сценарий должен содержать как минимум 2 узла: стартовый узел и узел, связанный с ним.

Управление узлами

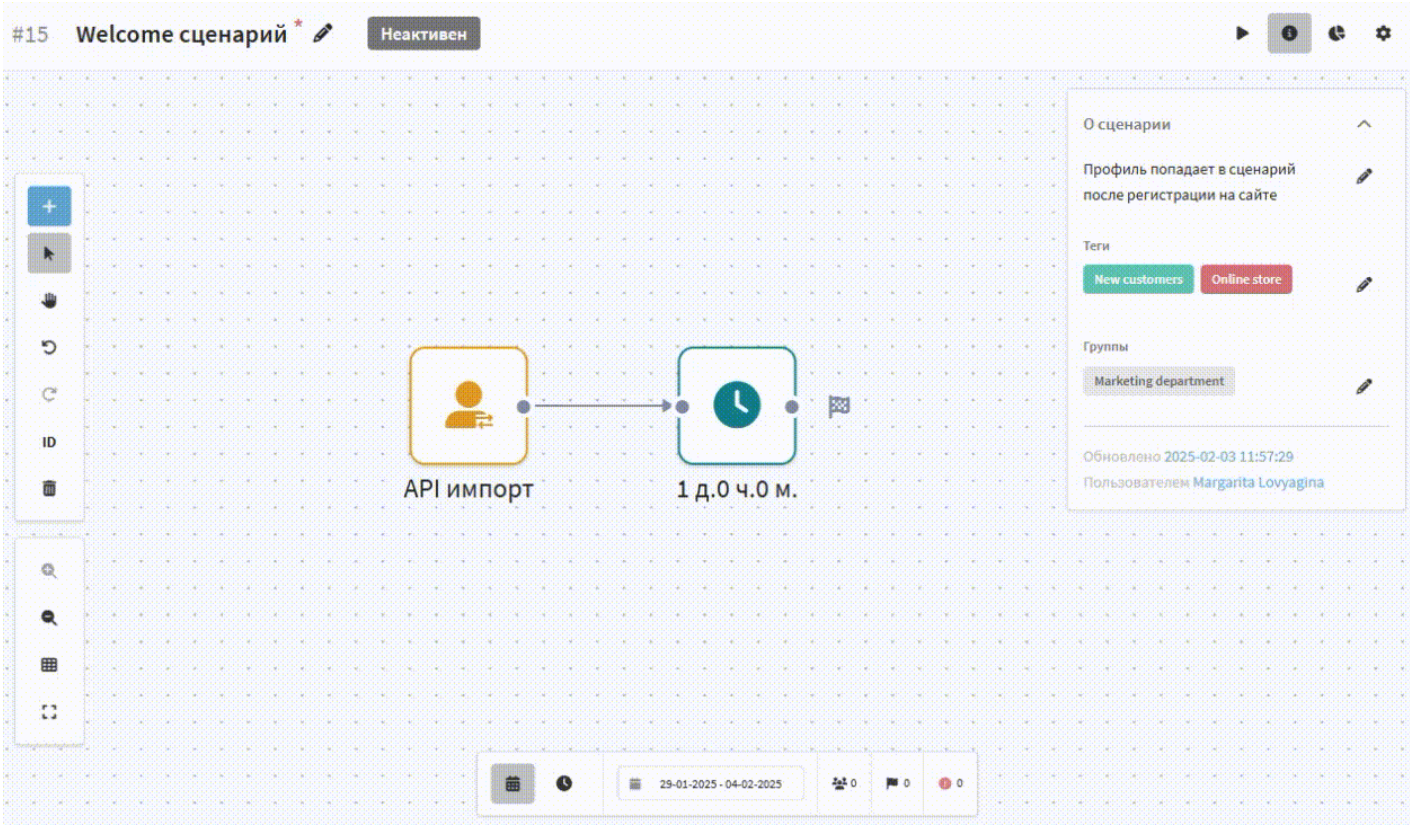
Добавление узла

Чтобы добавить узел в сценарий, нажмите кнопку **“+ Элемент”** на панели инструментов. В открывшемся списке нажмите на необходимый элемент или перетащите его на рабочее пространство, удерживая левую кнопку мыши.



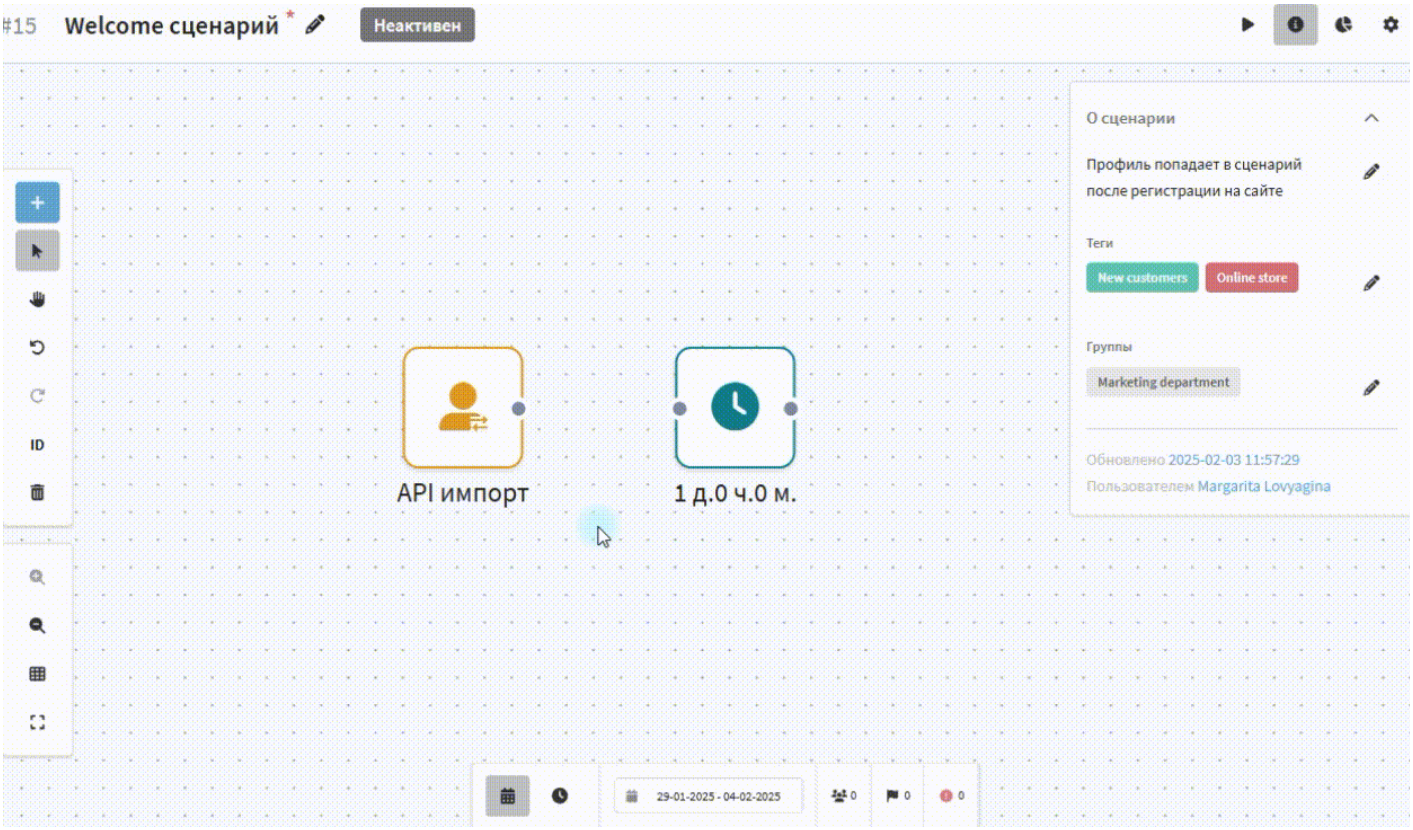
Настройка узла

Для каждого узла сценария задаются настройки. Чтобы открыть окно настроек, дважды кликните на нужный элемент. Для сохранения настроек нажмите “Применить”.

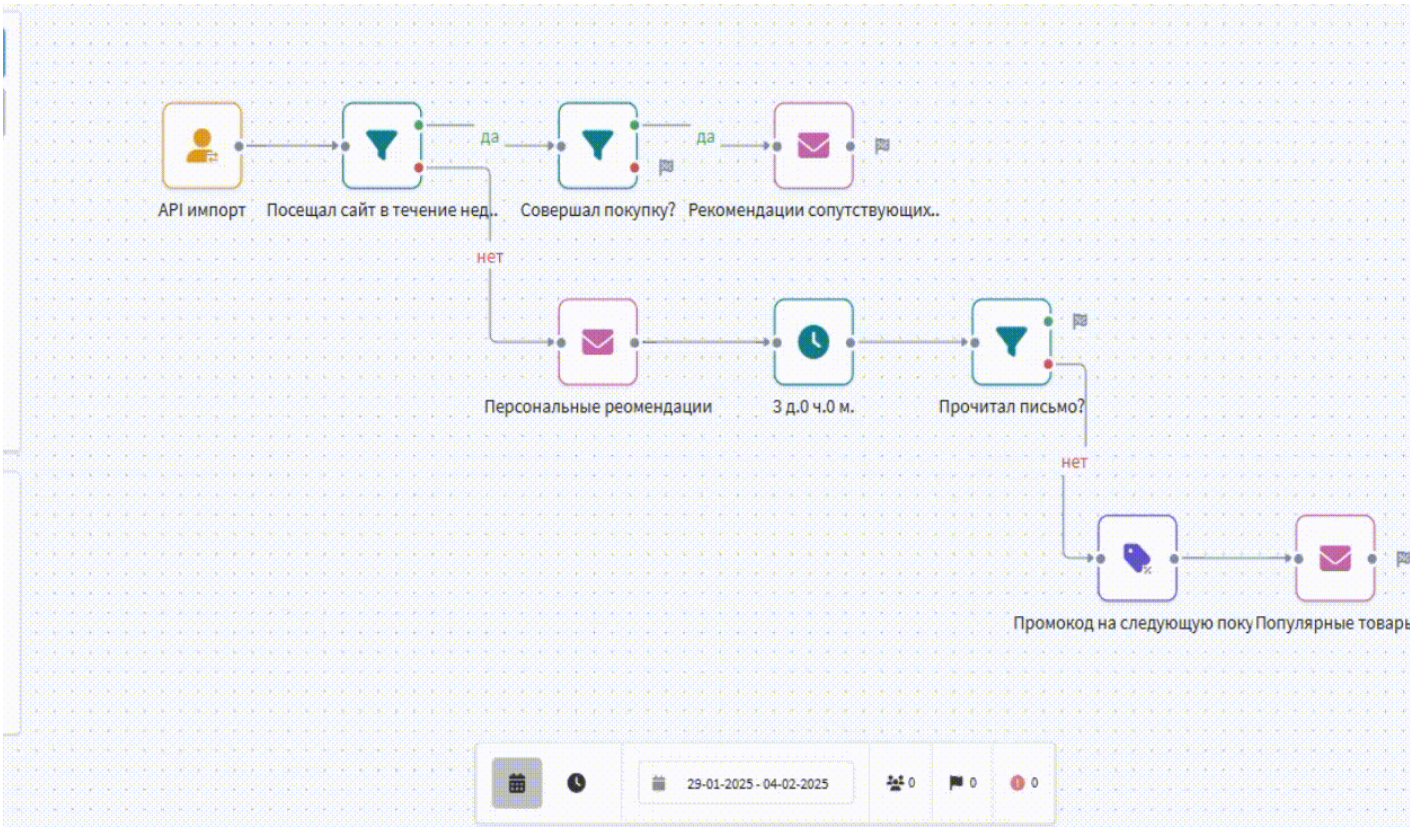


Добавление связи между узлами

Переход профиля из одного узла сценария в другой задается с помощью связей. Чтобы связать один узел с другим, протяните связь между ними:

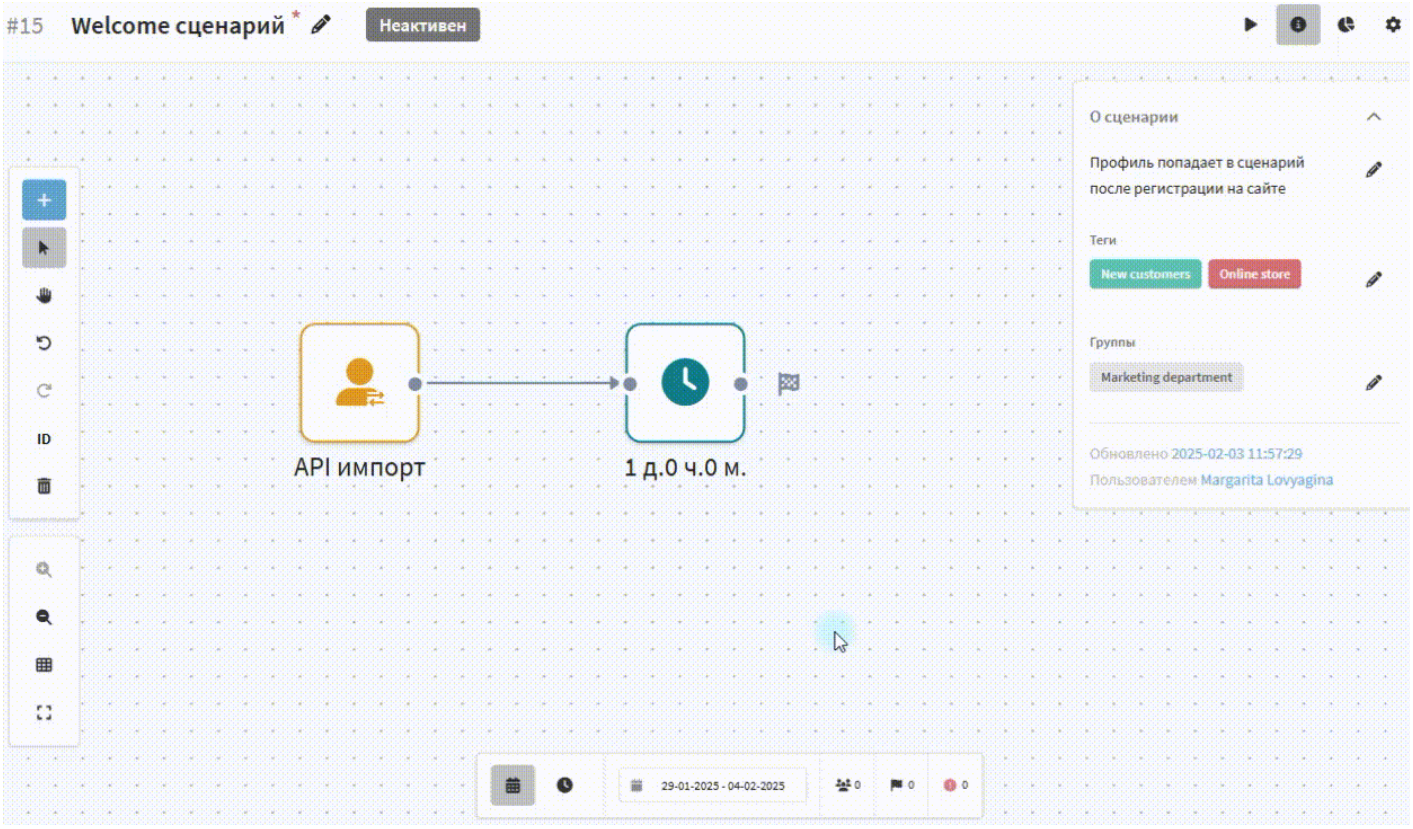


Узлы можно связывать не только последовательно. С помощью связей можно организовать многократное прохождение профилем одного и того же участка сценария или переключение на альтернативную последовательность действий:

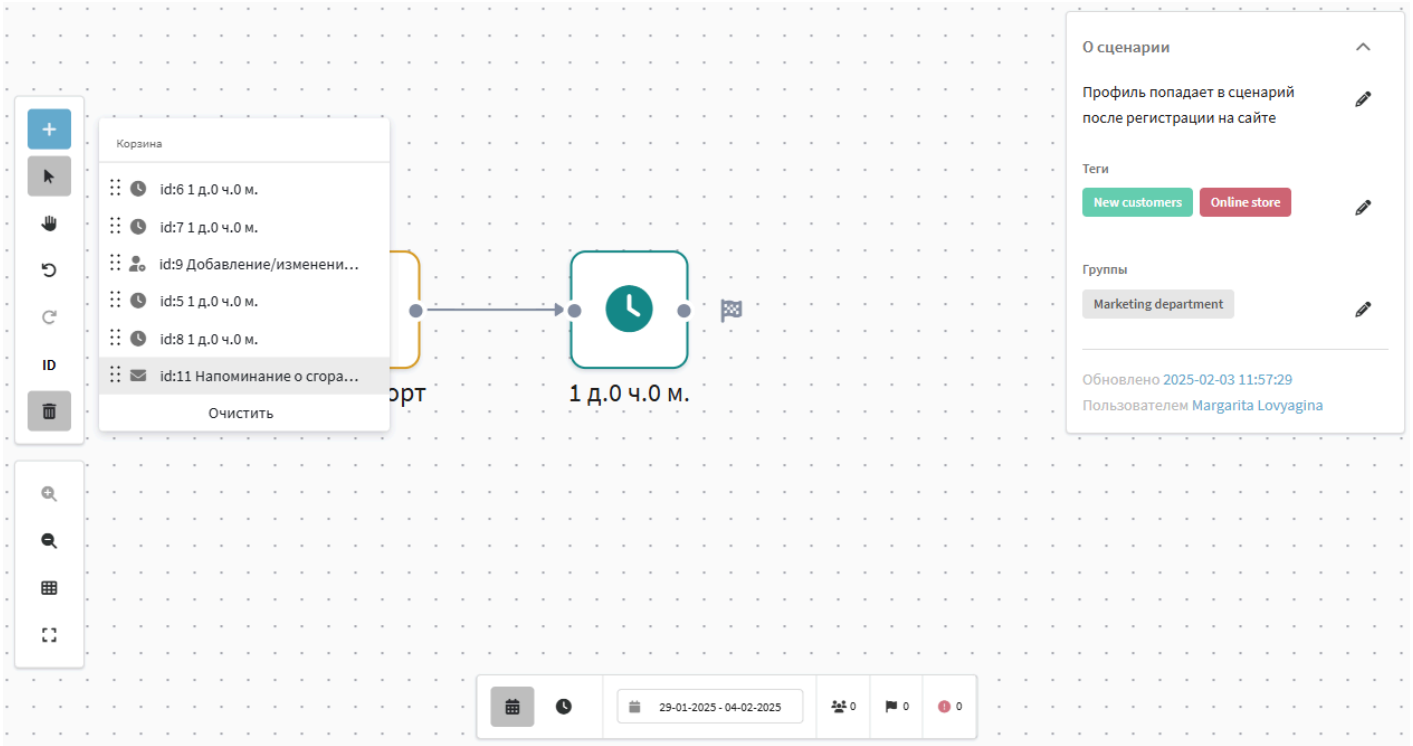


Удаление узла

Чтобы удалить узел или связь между узлами, выделите объект, кликнув на него. Затем нажмите правую кнопку мыши и в выпадающем меню выберите **“Переместить в корзину”** или воспользуйтесь `Backspace/Delete`. Доступно удаление нескольких узлов, включая связи между ними.



Если вы удалили узел из сценария, вы можете восстановить его. Для этого в правом нижнем углу нажмите кнопку **“Корзина”**. Появится список со всеми удаленными узлами. Выберите узел, который хотите вернуть в сценарий. После восстановления узел сохранит прежние настройки.



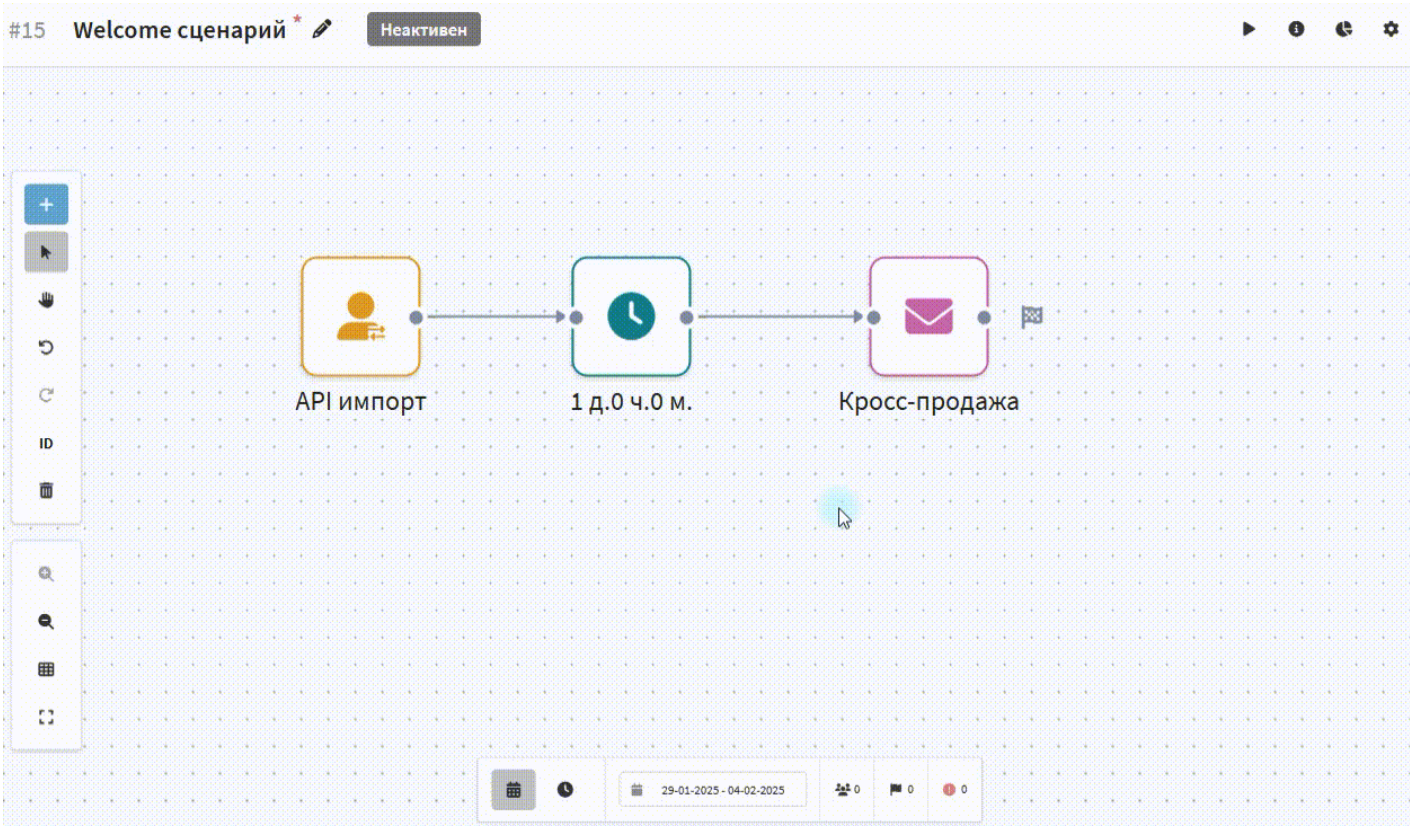
Также вы можете безвозвратно удалить все узлы из корзины, нажав кнопку **“Очистить”**.

CAUTION

Удаление узла, в котором в данный момент находятся профили, приведет к тому, что эти профили будут исключены из сценария. Если в узле нет задействованных профилей, то при удалении профили продолжат идти по обновлённому сценарию.

Клонирование узла

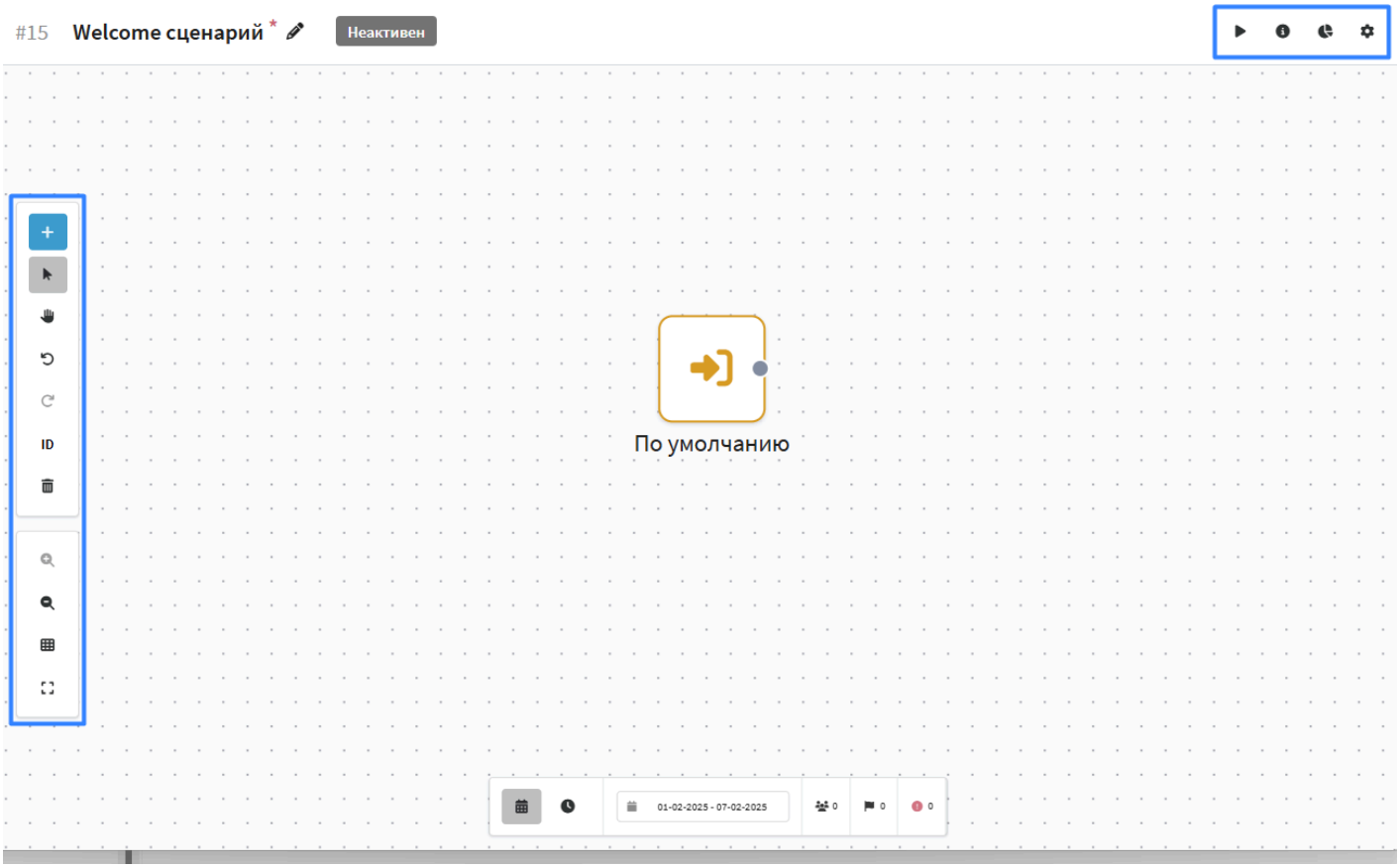
Чтобы клонировать узел, выделите его, щелкните правой кнопкой мыши на нем и выберите **“Копировать”** или используйте комбинацию клавиш `Ctrl` + `C`. Затем нажмите правую кнопку мыши и выберите **“Вставить”** или используйте комбинацию клавиш `Ctrl` + `V`. Таким же образом можно клонировать несколько узлов, включая связи между ними.



TIP

Для выделения нескольких узлов сценария активируйте инструмент “Указатель”. Затем зажмите левую кнопку мыши и выделите область с нужными объектами или зажмите клавишу `Ctrl` и, удерживая ее, последовательно выберите нужные объекты правой кнопкой мыши.

Панель инструментов



Инструменты для управления узлами

— активировать инструмент “Указатель”. Используется для управления узлами сценария: перемещение, соединение, выделение и т.д.

— активировать инструмент “Рука” (Space + drag). Используется для перемещения по рабочему пространству сценария.

— отменить последнее действие `Ctrl` + `Z`.

— вернуть последнее отмененное действие `Ctrl` + `Shift` + `Z`.

— открыть “Корзину”. “Корзина” содержит все удаленные из сценария узлы.

ID

— отобразить ID каждого узла вместо названия.

Инструменты для настройки сценария

— активировать сценарий.

— деактивировать сценарий.

— показать информацию о сценарии (описание, теги и группы для сценария).

— просмотреть аналитику сценария в отчете по каналам.

— показать настройки сценария (режим повторного входа профиля в сценарий и др.).

Инструменты для управления рабочим пространством

— увеличить масштаб рабочего пространства `Ctrl +`.

— уменьшить масштаб рабочего пространства `Ctrl -`.

— применить выравнивание. Узлы выравниваются относительно друг друга внутри рабочего пространства.

— подогнать вид. Масштаб изменяется на оптимальный, что позволяет увидеть все узлы внутри рабочего пространства.

Список узлов

Все узлы сценария поделены на четыре категории:

- **Триггеры** — узлы, определяющие способ захвата профиля в сценарий.
- **Операторы** — узлы, влияющие на движение профиля по сценарию.
- **Каналы** — узлы, с помощью которых профилю отправляется сообщение.
- **Действия** — узлы, запускающие определенное действие в платформе в момент вхождения профиля.

Триггеры

Триггер — это узел, который запускает сценарий. Можно настроить один или несколько триггеров. Максимальное количество триггеров в одном сценарии — 10. Один тип триггера можно использовать несколько раз в одном сценарии.

INFO

Триггеры в контексте данной статьи используются в сценариях для запуска профилей в них. Не следует путать их с триггерами в рассылках.

Захват профилей по умолчанию

	Этот узел является встроенной функцией сценария и не подлежит удалению. С его помощью профили могут попадать в сценарий из разных точек платформы.

Подробнее

Данный узел обеспечивает захват профилей при следующих действиях:

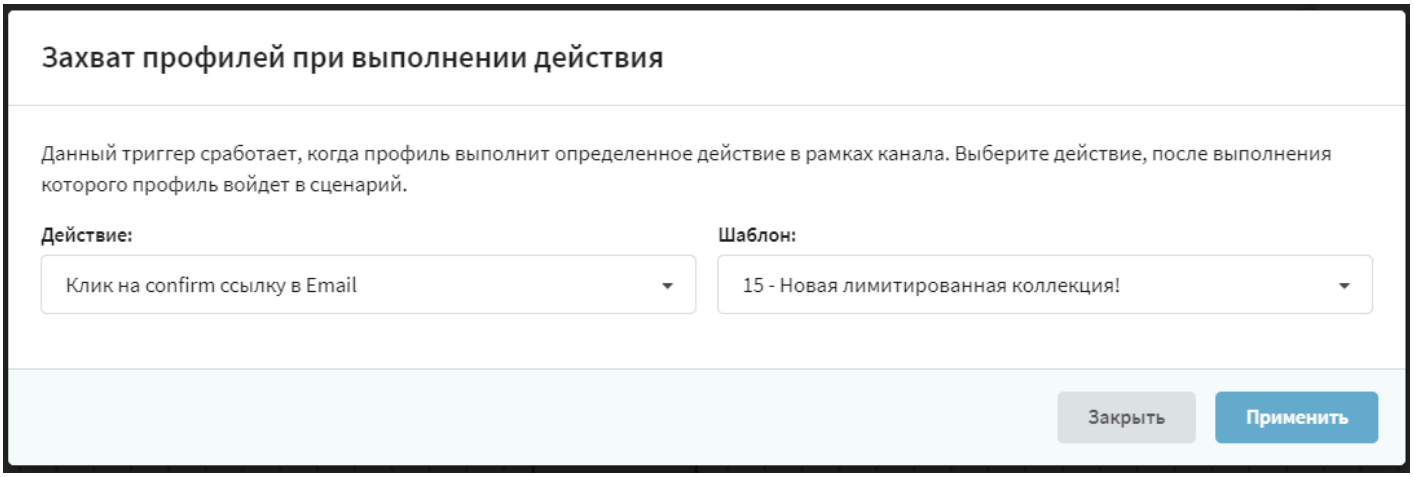
- **При отправке профиля в сценарий через API.** После сохранения сценария в окне узла отобразится URL, который можно использовать для запуска сценария. Однако рекомендуется использовать более новую версию API. Подробнее в документации.
- **При добавлении профиля в сценарий вручную.**
- **При отправке сегмента в сценарий** в панели быстрых действий.
- **При переходе через узел из другого сценария.**

Выполнение действия

	Активируется, когда подписчик совершает нужное действие в сообщении.

Подробнее

В выпадающем списке выберите **действие** в одном из каналов отправки, а также укажите **шаблон**, в котором хотите отслеживать это действие:



Действие	Описание
Любой клик в Email	Получатель кликнул по любой ссылке в выбранном шаблоне сообщения.
Клик по ссылке в Email	Получатель кликнул по конкретной ссылке в выбранном шаблоне сообщения.
Открыт Email	Пиксель отслеживания зафиксировал открытие письма по выбранному шаблону.

Прочитан Email	Пиксель отслеживания зафиксировал прочтение письма по выбранному шаблону.
Клик на confirm ссылку в Email	Получатель кликнул по ссылке подтверждения подписки в выбранном шаблоне.
Клик в SMS	Платформа зарегистрировала клик по любой ссылке в выбранном шаблоне.
Клик по ссылке в SMS	Платформа зарегистрировала первый переход по конкретной ссылке в выбранном шаблоне.
Открыт Push	Платформа зарегистрировала открытие выбранного шаблона Push-уведомления. Push-уведомления обычно открываются автоматически, если разрешены в браузере или ОС мобильного устройства.
Любой клик в Push	Клик по любой ссылке в выбранном шаблоне Push-уведомления.
Открыт попап	Платформа зарегистрировала показ выбранного шаблона попапа пользователю.
Клик по попапу	Пользователь кликнул по выбранному шаблону попапа.
Закрыт попап	Платформа зарегистрировала клик по кнопке закрытия попапа.
Пользовательское событие попапа	Платформа зарегистрировала пользовательское событие в попапе.
Заполнен попап	Платформа зарегистрировала заполнение полей в выбранном шаблоне попапа пользователем.
Notify клик	Платформа зарегистрировала клик по любой ссылке в выбранном шаблоне Notify. Триггер срабатывает при каждом переходе и повторно отправляет профиль в сценарий, если не указано иное.
Notify уник. клик	Платформа зарегистрировала первый переход по любой ссылке в шаблоне Notify.
Notify уник. клик по ссылке	Платформа зарегистрировала первый переход по конкретной ссылке в шаблоне Notify.
Notify прочтение	Платформа зарегистрировала прочтение шаблона Notify.
Whatsapp*/Viber™/Telegram bot клик	Платформа зарегистрировала любой переход по ссылке в выбранном шаблоне. Триггер срабатывает при каждом переходе и повторно отправляет профиль в сценарий, если не указано иное.
Whatsapp*/Viber™/Telegram bot/ уник. клик	Платформа зарегистрировала первый переход по любой ссылке в шаблоне.
Whatsapp*/Viber™/Telegram bot уник. клик по ссылке	Платформа зарегистрировала первый переход по конкретной ссылке в выбранном шаблоне.
Whatsapp*/Viber™/Telegram bot подтверждение подписки	Платформа зарегистрировала клик по ссылке подтверждения подписки в шаблоне.

Обновление через форму

	Триггер захватывает в сценарий профили, чьи данные будут обновлены в результате заполнения формы. Срабатывает только при обновлении существующих профилей.

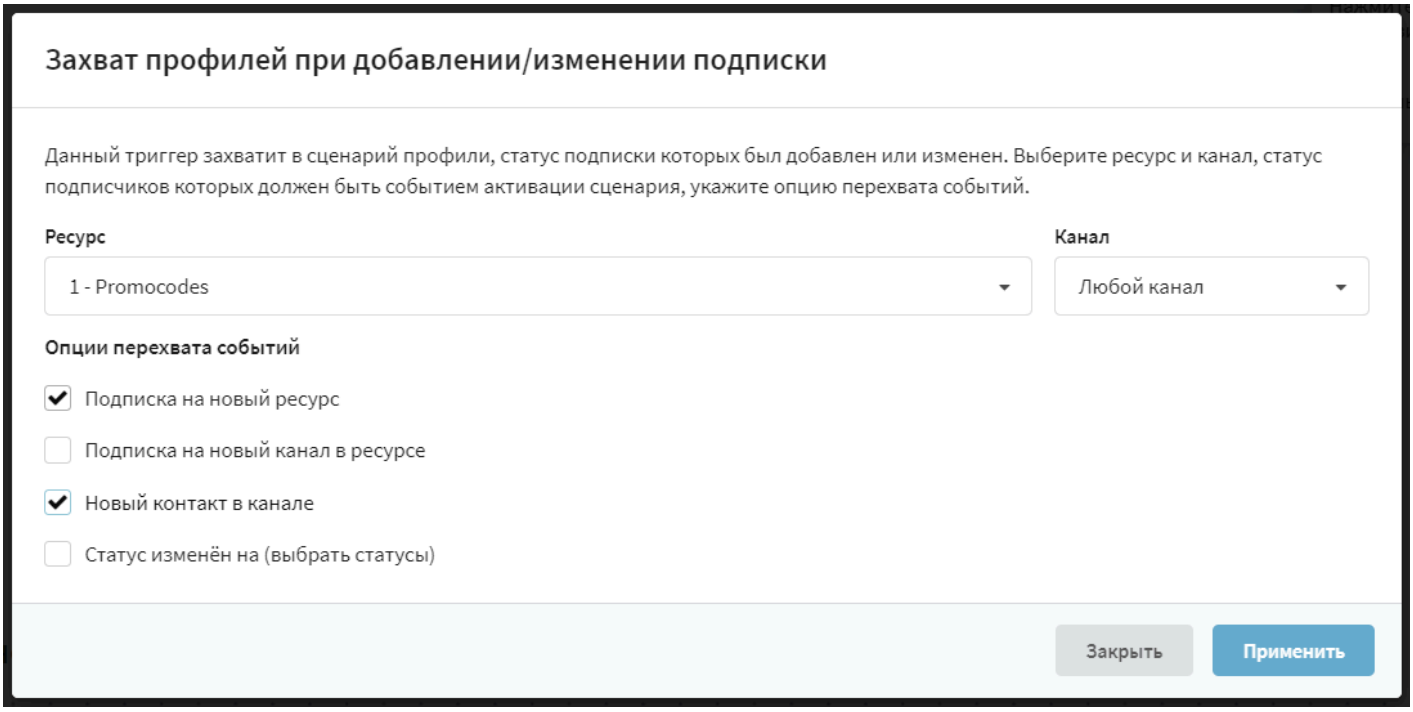
Подробнее

В выпадающем окне укажите форму, к которой будет привязан триггер:

Добавление/изменение подписки

	Триггер активируется, когда профилю добавляется новая подписка или меняется статус существующей. Для триггера можно отдельно указать **Ресурс** и **Канал**, по которым будут отслеживаться изменения. В противном случае, рассылка будет ожидать изменений по всем каналам выбранного ресурса, по всем ресурсам для выбранного канала или по любым подпискам профиля.

Подробнее



- **Подписка на новый ресурс** — условие сработает только для первой подписки на указанный в рассылке ресурс. При добавлении других подписок на тот же ресурс сценарий не активируется.
- **Подписка на новый канал в ресурсе** — добавлена подписка на новый (для профиля) канал в ресурсе. То есть, была подписка на email рассылки, добавилась ещё и на SMS или Push.
- **Новый контакт в канале** — активирует триггер при каждом появлении новой подписки. Первая подписка на канал также считается “новым контактом в канале”.
- **Статус изменён на (выбрать статусы)** — запустится при изменении статуса подписки на выбранные из выпадающего меню.
- **Первые три пункта выбраны** — триггер сработает при каждой новой подписке.
- **Все галочки сняты** — ничего не происходит.

NOTE

На подписки, которые добавлены со статусом, отличным от “Подписан”, триггер не активируется.

Сегмент по расписанию

	Группа профилей, соответствующая определенным условиям, попадает в сценарий по заданному расписанию.

Подробнее

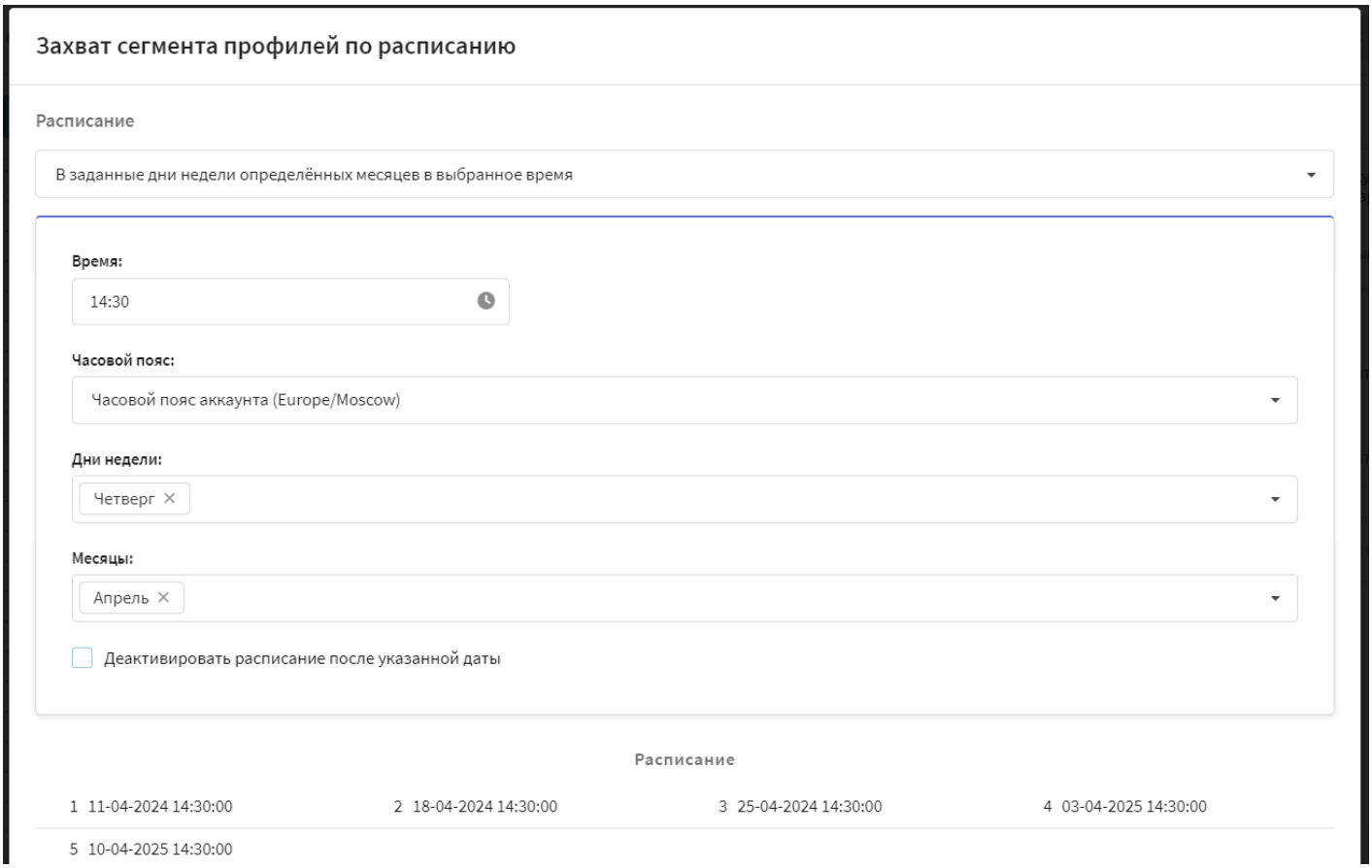
Для начала необходимо настроить **расписание захвата**. Выберите тип расписания и настройте время, когда сценарий будет запускаться для сегмента.

Дополнительно вы можете **Деактивировать расписание** после определённой даты или сменить часовой пояс для отсчёта времени запусков.

После внесения изменений под расписанием отобразится таблица со **следующими запусками** сценария.

TIP

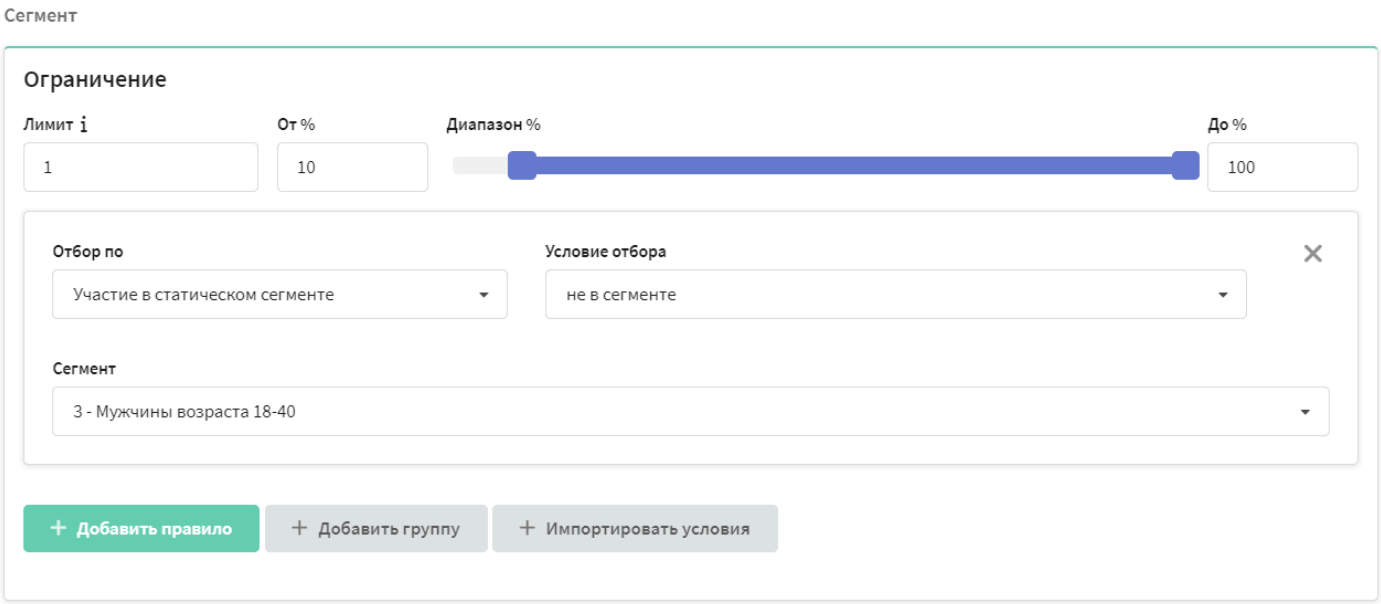
Настройка расписания подробно описана в **отдельной статье**.



Далее настраивается **выборка профилей**. Составьте запрос сегментации к базе данных, добавляя правила или группы правил. Доступна сегментация по:

- **основным данным профиля**
- **истории действий**
- **подпискам**
- и другим параметрам.

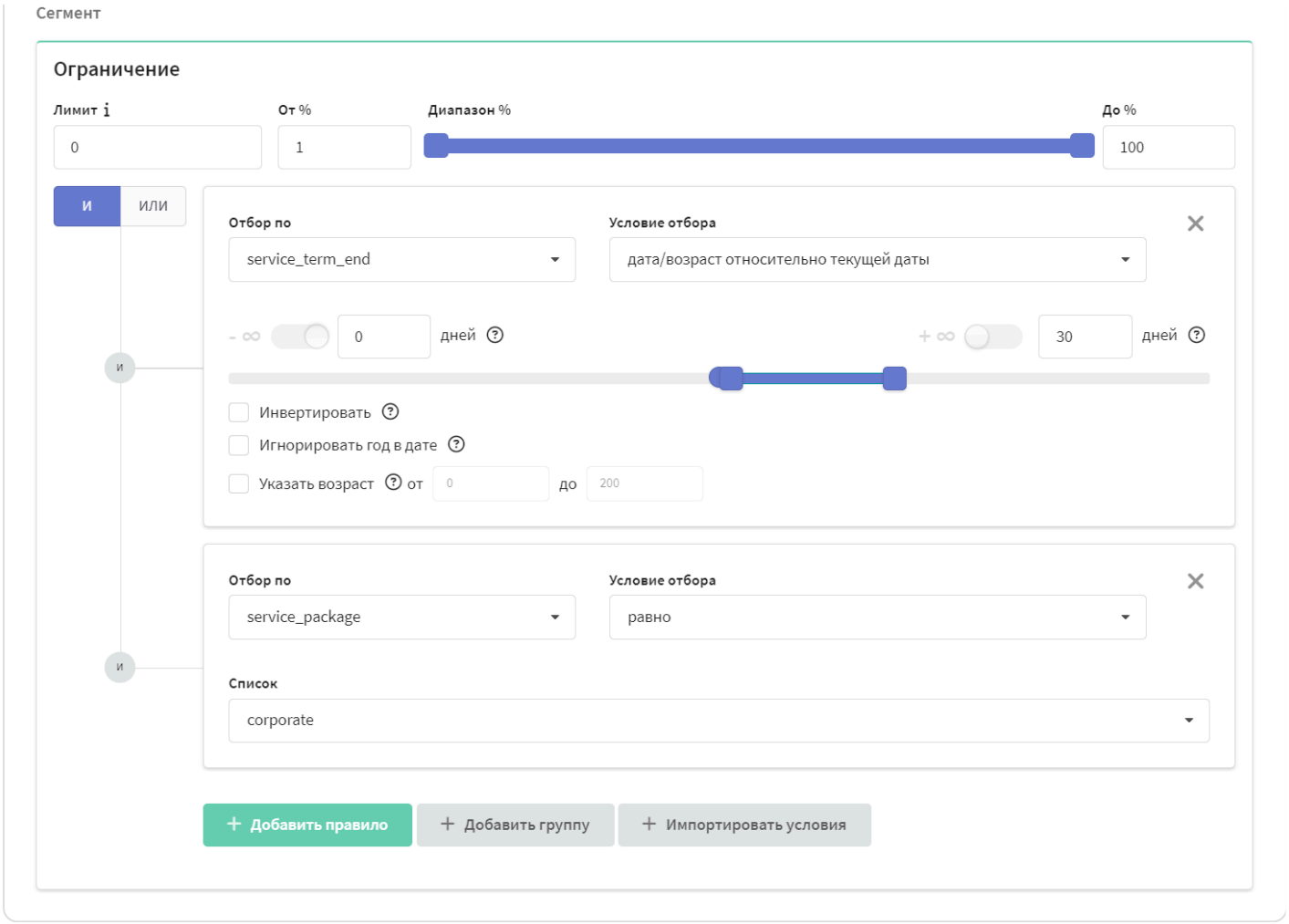
Подробнее сегментация описана в **соответствующем разделе документации**. Дополнительно можно установить **ограничение** получившейся выборки в процентах.



TIP

Сценарий по расписанию запускается аналогично регулярной рассылке. Сегмент клиентов рассчитывается динамически при каждом запуске.

С помощью таких сценариев можно организовать последовательность коммуникаций для клиентов, у которых заканчивается срок предоставления каких-либо услуг: сценарий по сегменту (на примере ниже) будет каждый день выбирать только тех клиентов, у кого до окончания срока остался 1 месяц.

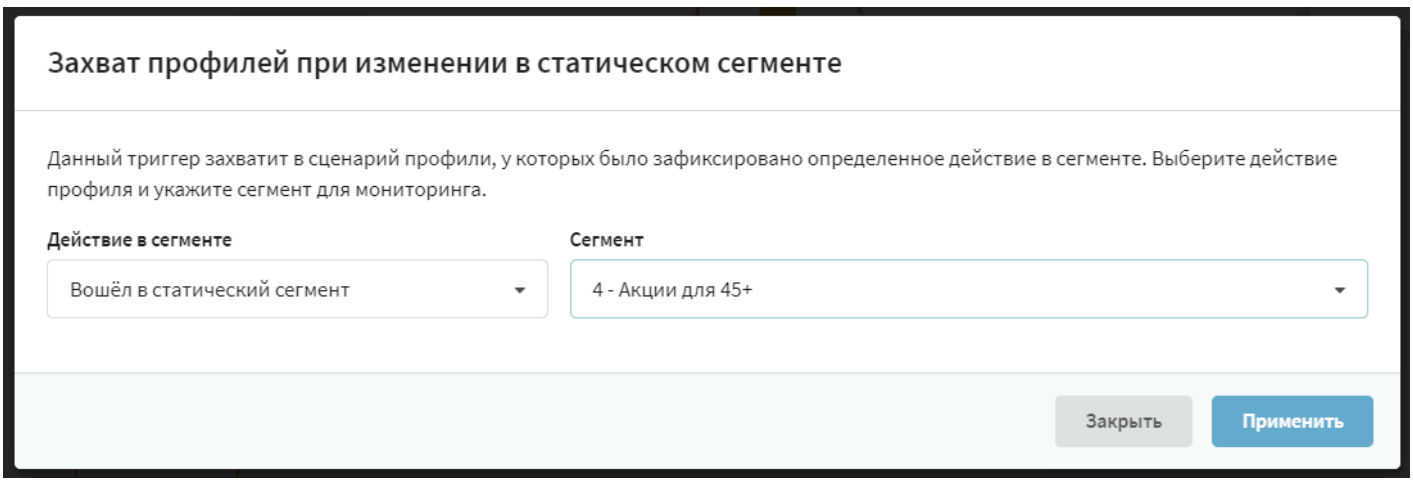


Изменение в статическом сегменте

	Триггер запускается при изменении в [**статическом сегменте**](/user-guide/segmentation/9407279): профиль может как войти в указанный статический сегмент, так и выйти из него.

Подробнее

В выпадающем окне выберите **действие** (вход или выход из статического сегмента) и **сегмент**, который хотите отслеживать:



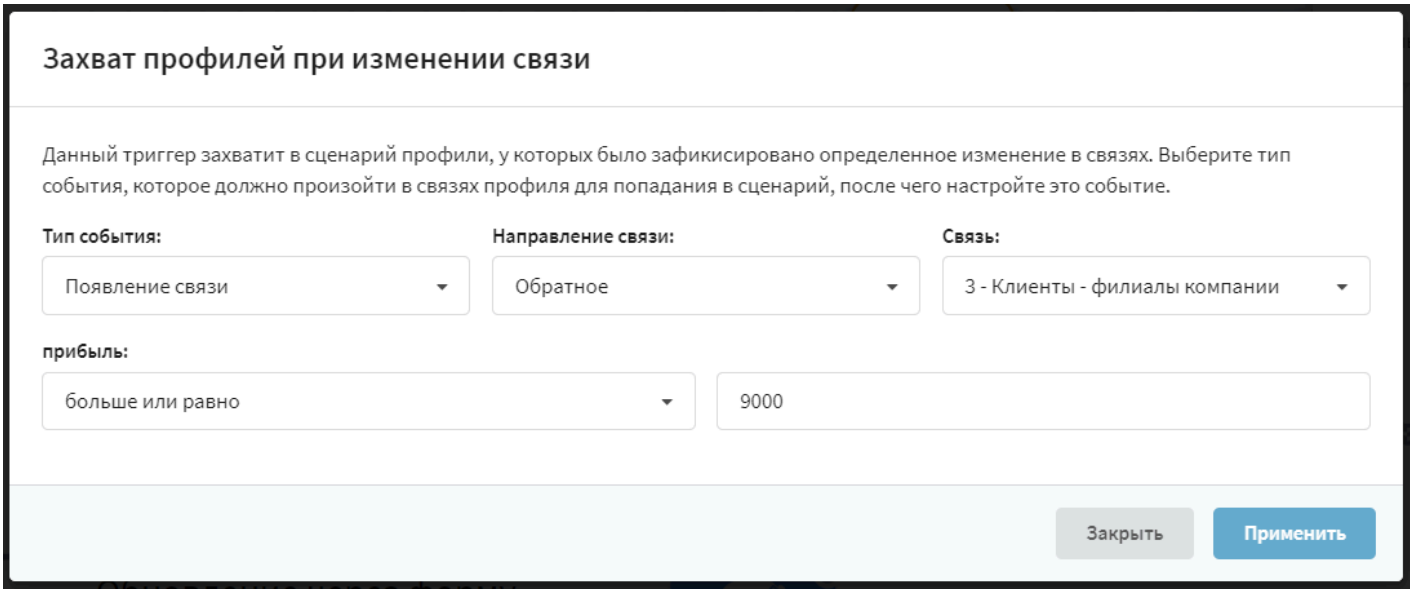
Изменение связи

	Триггер запускается при изменении [**связи**](/user-guide/profiles-and-databases/20480122) между профилями: как прямой, так и обратной. При этом может учитываться появление, изменение или удаление связи, а также достижение свойством связи заданного в триггере значения.`Delete`/`Backspace`

Подробнее

Чтобы настроить триггер, укажите:

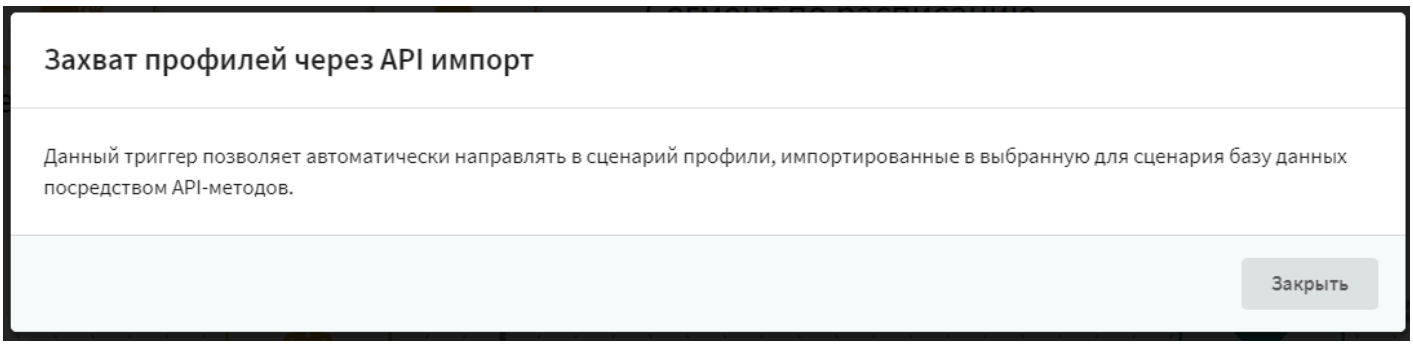
- связь,
- тип события, привязанного к ней,
- направление связи,
- свойство связи,
- значение свойства.



API импорт

	Триггер активируется, когда в базу данных добавляется новый профиль через [**API запрос**](/developer-guide/profiles/2195477). Профиль должен добавляться в базу, выбранную для сценария.

Подробнее



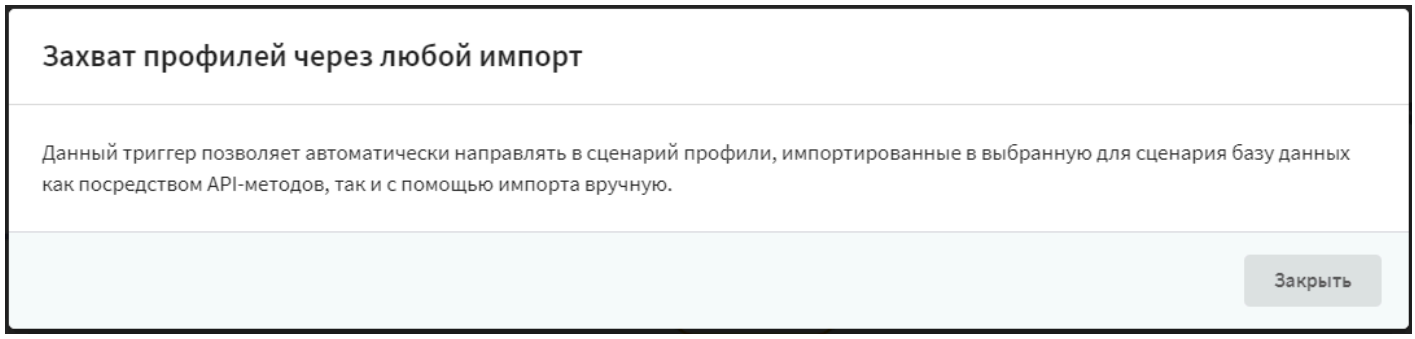
CAUTION

При выборе способа **“Любой импорт”** этот триггер становится недоступен.

Любой импорт

	Триггер активируется, когда в базу данных добавляется новый профиль любым способом: посредством [**API-запроса**](/developer-guide/profiles/2195477) или [вручную](/user-guide/profiles-and-databases/9405665#Scenarios) из карточки профиля.

Подробнее



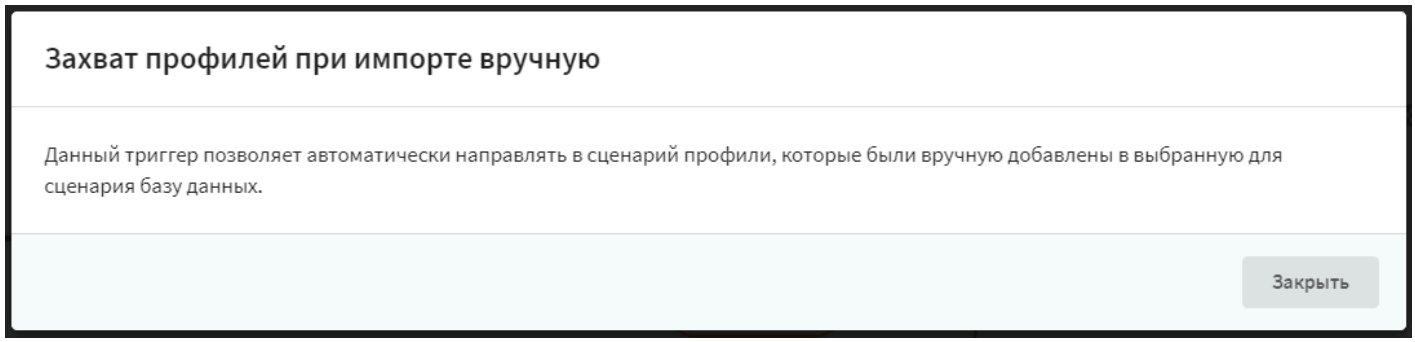
CAUTION

При выборе способа **“Любой импорт”** остальные триггеры на импорт становятся недоступными.

Импорт вручную

	Триггер активируется, когда в базу данных [**вручную добавляется**](/user-guide/profiles-and-databases/9405665) новый профиль клиента.

Подробнее



CAUTION

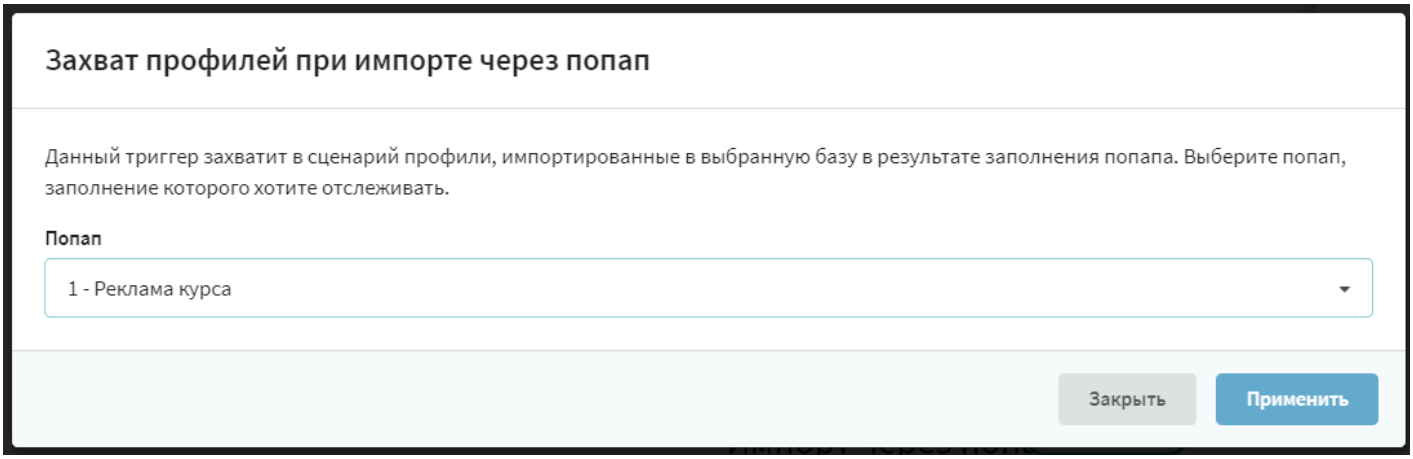
При выборе способа **“Любой импорт”** этот триггер становится недоступен.

Импорт через попап

	Триггер сработает, если в базе будет создан новый профиль в результате того, что пользователь заполнил и отправил форму в pop-up окне.

Подробнее

Выберите попап, заполнение которого необходимо отслеживать. Pop-up окно можно создать в разделе **Веб-слой** —> **Попапы**, а настроить импорт профилей в базу можно в pop-up редакторе в разделе **“Действия”**.



CAUTION

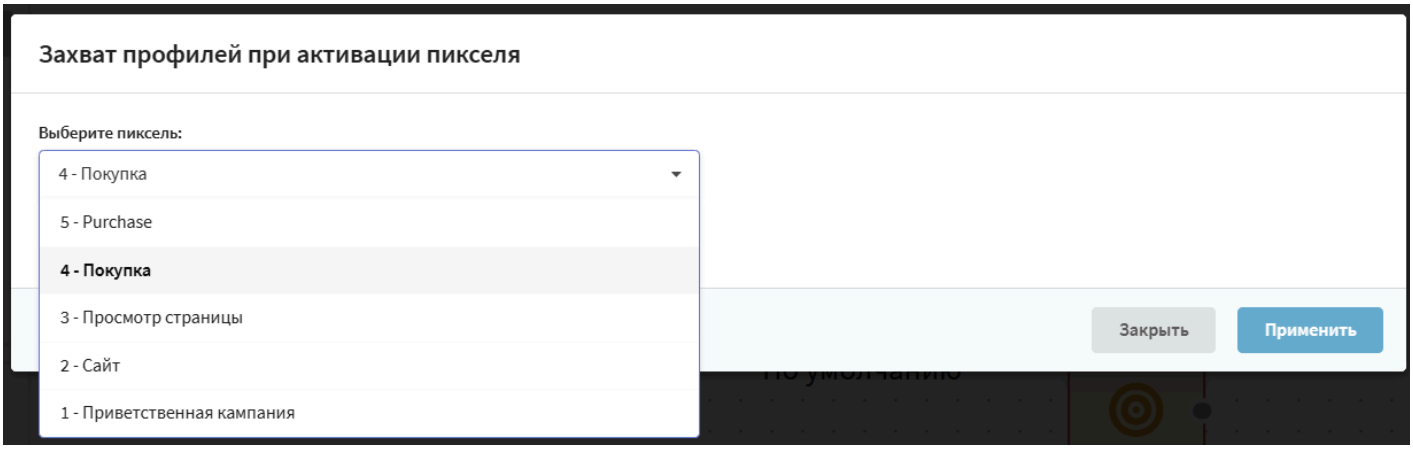
При выборе способа **“Любой импорт”** этот триггер становится недоступен.

Активация пикселя

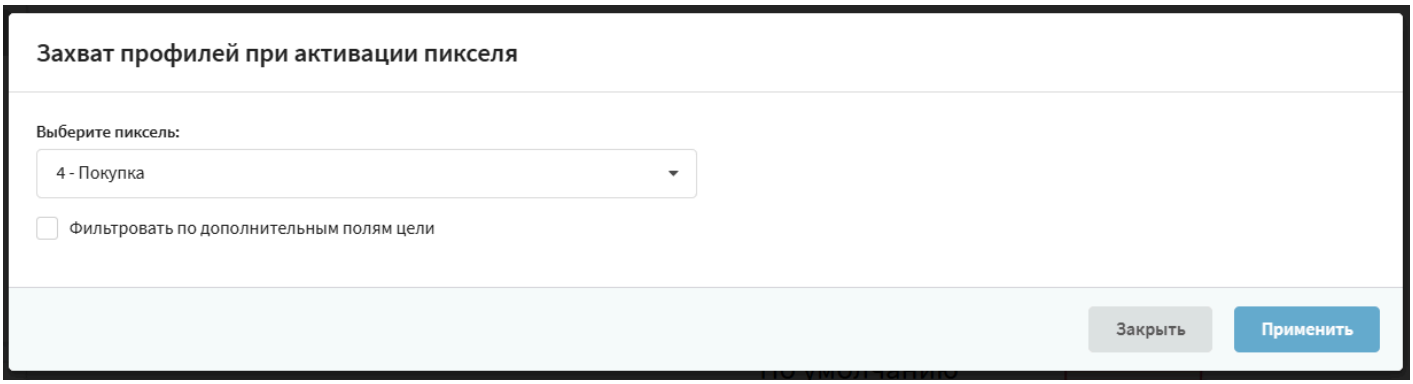
	Профиль попадает в сценарий, когда активируется пиксель.

Подробнее

В выпадающем окне выберите необходимый пиксель:

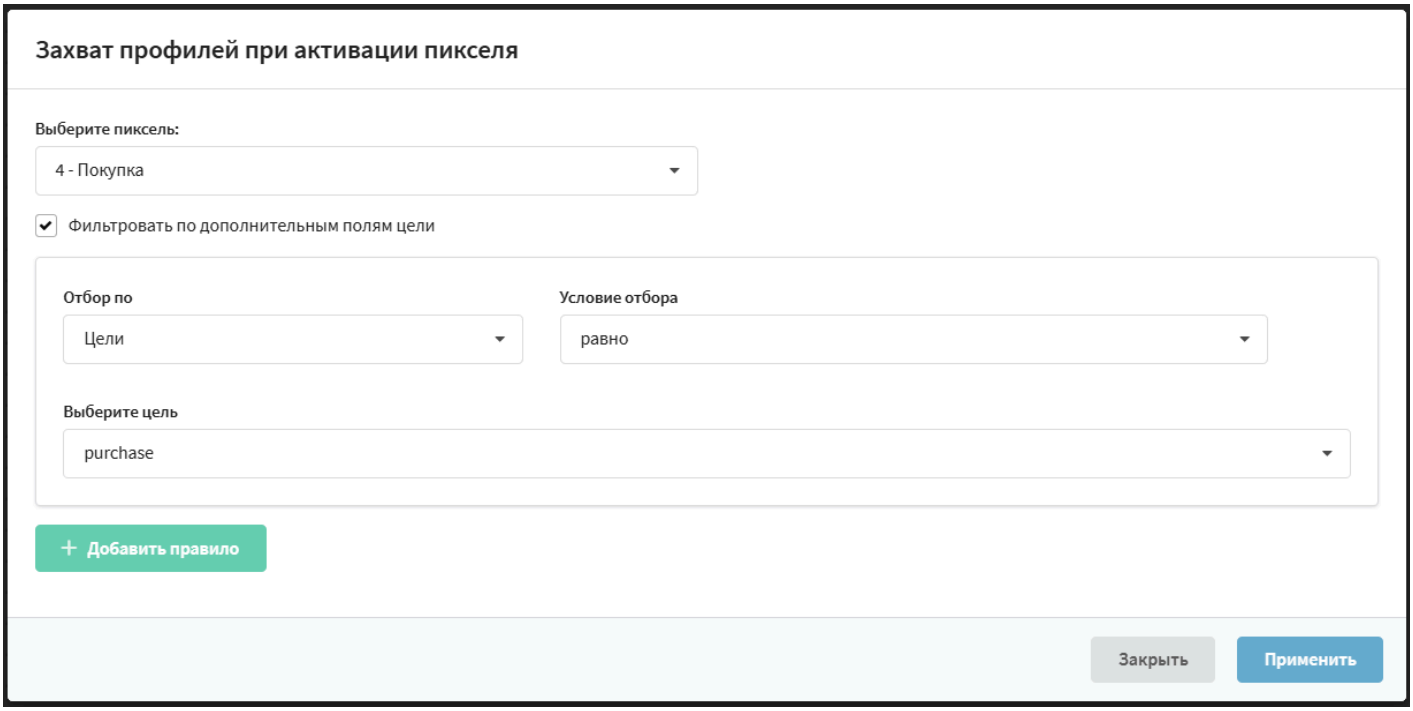


Данный триггер может отслеживать срабатывание пикселя по умолчанию:



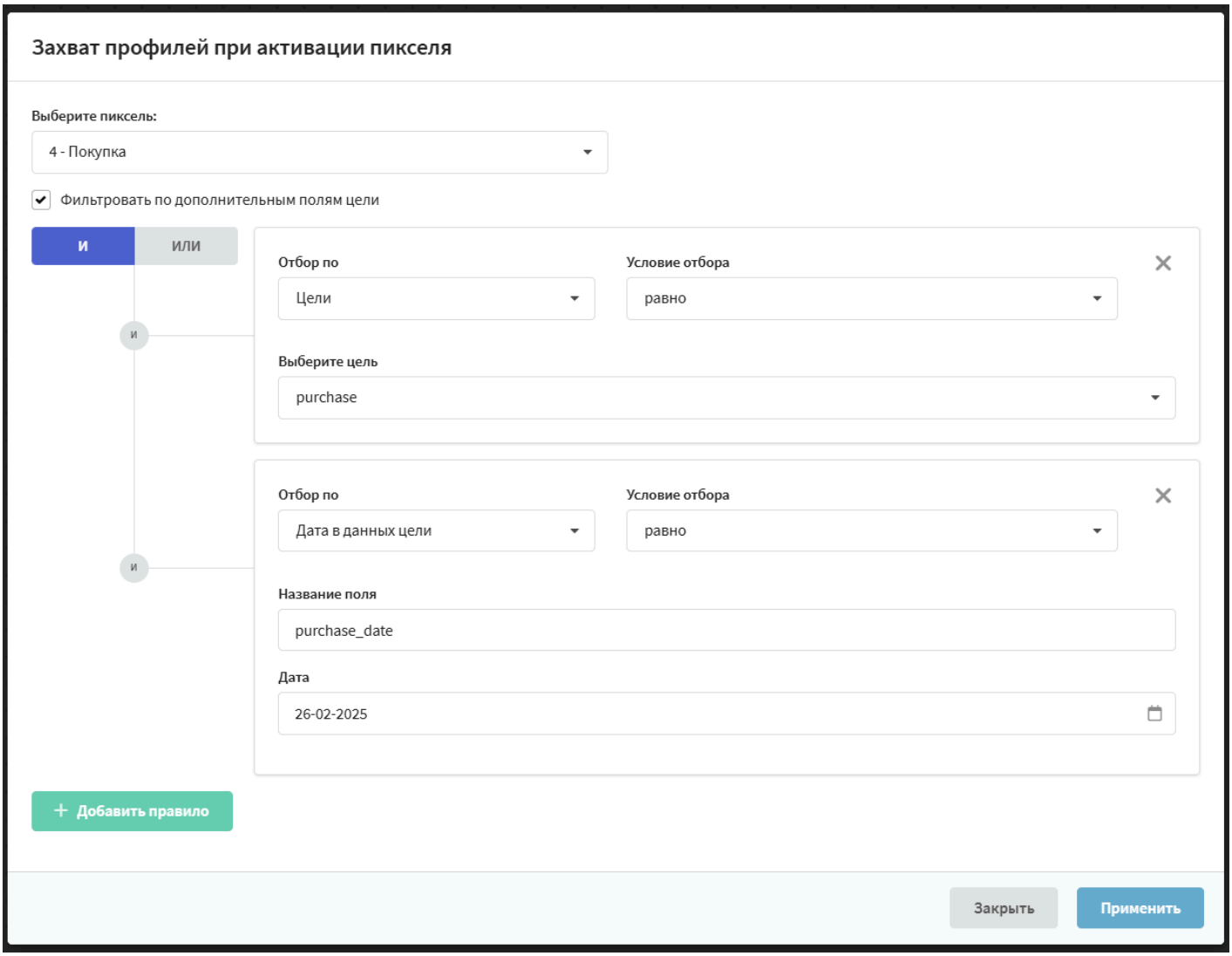
Если вам необходимо отслеживать только определенные события, то можно настроить триггер для регистрации конкретной цели. Для этого нужно установить галочку в поле **“Фильтровать по дополнительным полям цели”** и настроить необходимые параметры отбора.

С помощью такого триггера пиксель будет фиксировать только те события, которые связаны с конкретной целью. Например, вы можете настроить фильтрацию так, чтобы отслеживались только регистрации профилей, покупки или другие ключевые действия.



Для более точной настройки можно использовать объединение нескольких правил отбора. Для этого нужно установить галочку в поле **“Фильтровать по дополнительным полям цели”**, нажать **“Добавить правило”** и настроить два или более правил отбора.

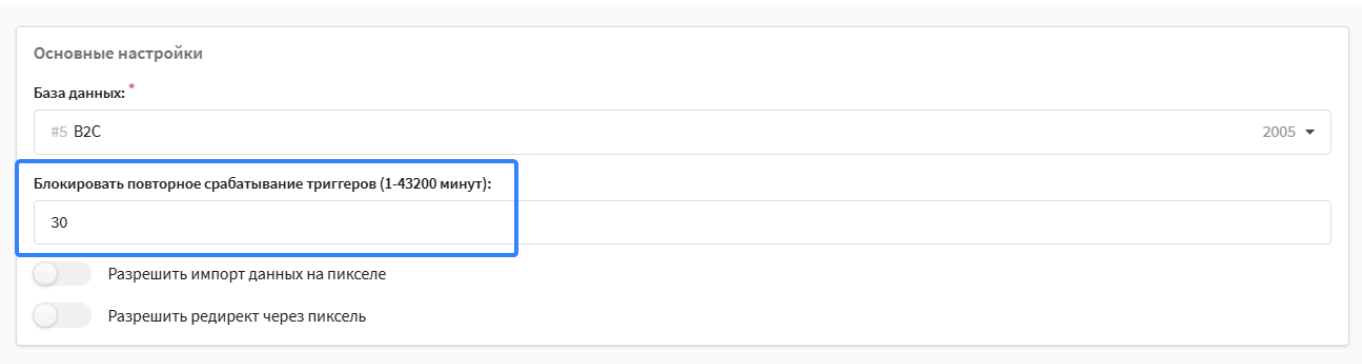
Используйте **“И”**, когда хотите, чтобы событие строго подходило под оба условия. Используйте **“ИЛИ”**, если хотите, чтобы событие подходило хотя бы под одно из условий.



Также можно отфильтровать профили по дополнительным полям. Значения дополнительных полей [передаются посредством API-запроса](#).

TIP

Если триггер не активируется в рассылках или в сценариях, хотя условия триггера выполняются, рекомендуется проверить **интервал срабатывания пикселя**.

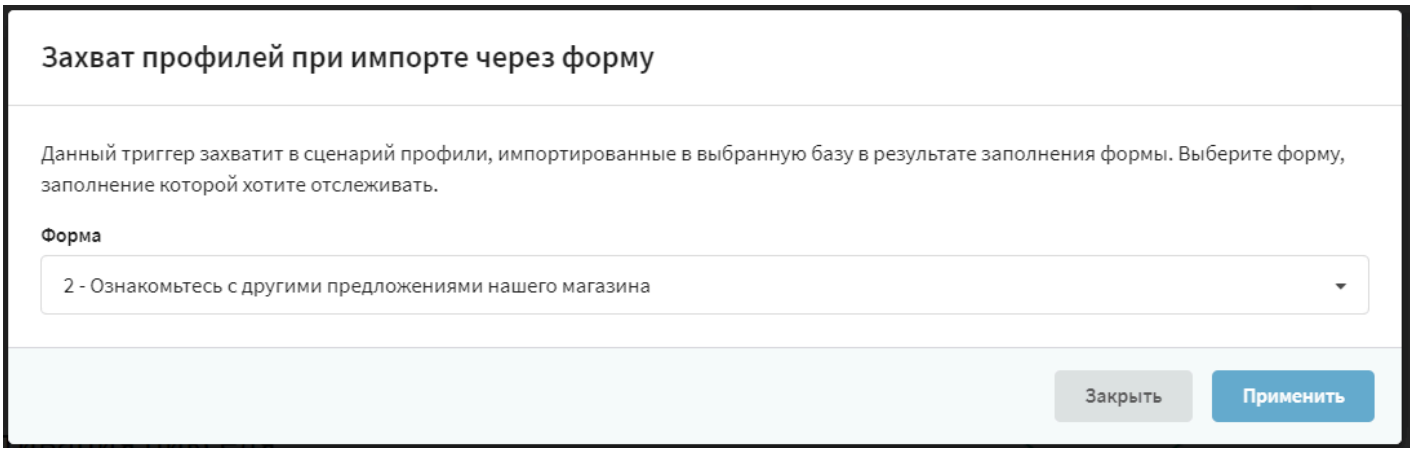


Импорт через форму

	Триггер сработает, если в базе будет создан новый профиль в результате того, что пользователь заполнил и отправил форму.

Подробнее

Чтобы использовать этот триггер, вам необходимо при настройке формы активировать опцию **“Импорт профилей”** на вкладке **“Действия”**. Импорт должен происходить в ту же базу, которая используется в сценарии.



CAUTION

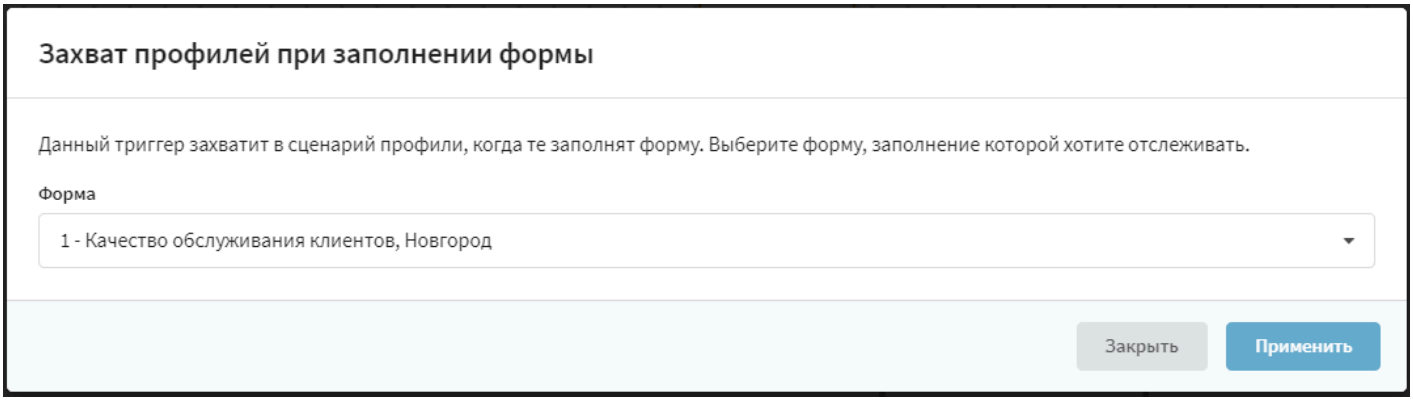
При выборе способа **“Любой импорт”** этот триггер становится недоступен.

Заполнение формы

	Триггер сработает только в случае, если форма была заполнена по переходу из письма.

Подробнее

В выпадающем списке укажите форму, заполнение которой хотите отслеживать:



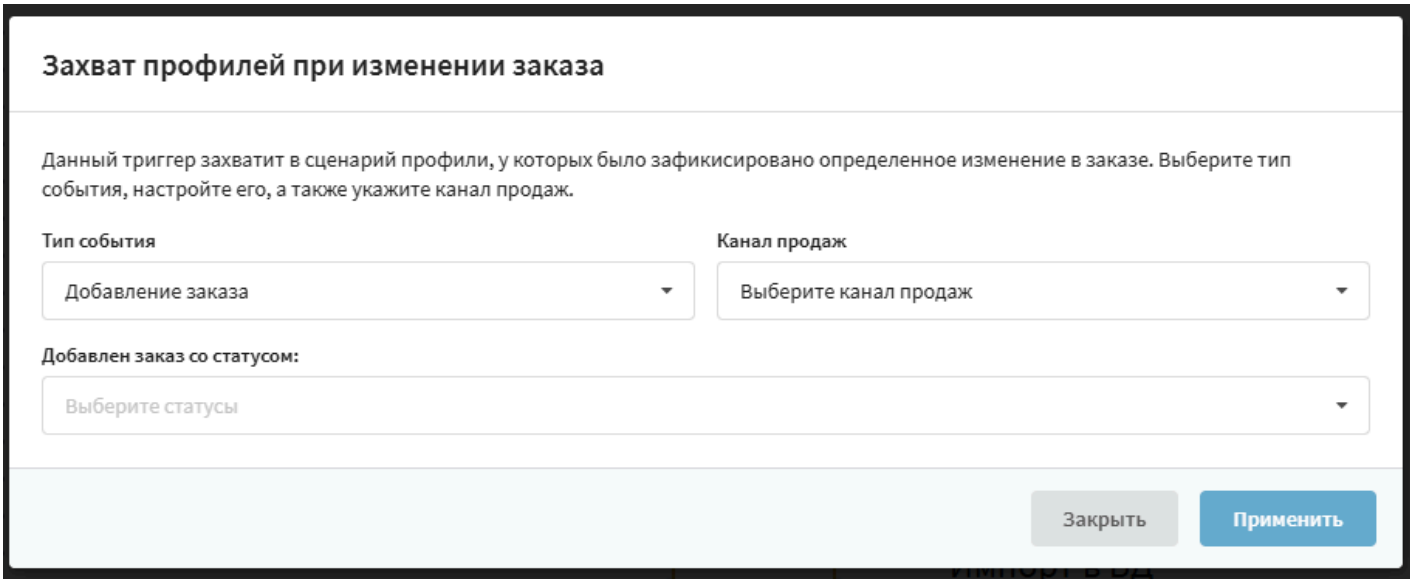
TIP

Подбронее о том, как отправить профилю ссылку на форму, читайте в **этой статье**.

Изменение заказа

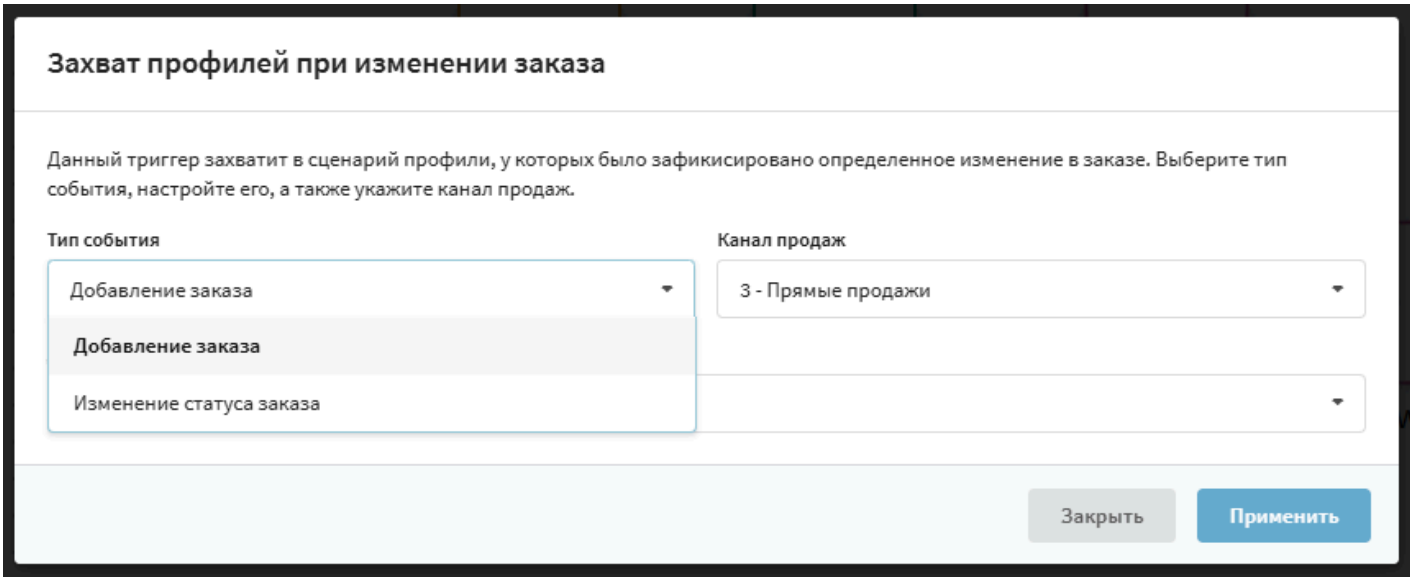
	Данный триггер захватит в сценарий профили, у которых было зафикисировано определенное изменение в заказе. Триггер сработает только в случае, если настроен канал продаж, привязанный к базе данных, выбранной для данного сценария.

Подробнее



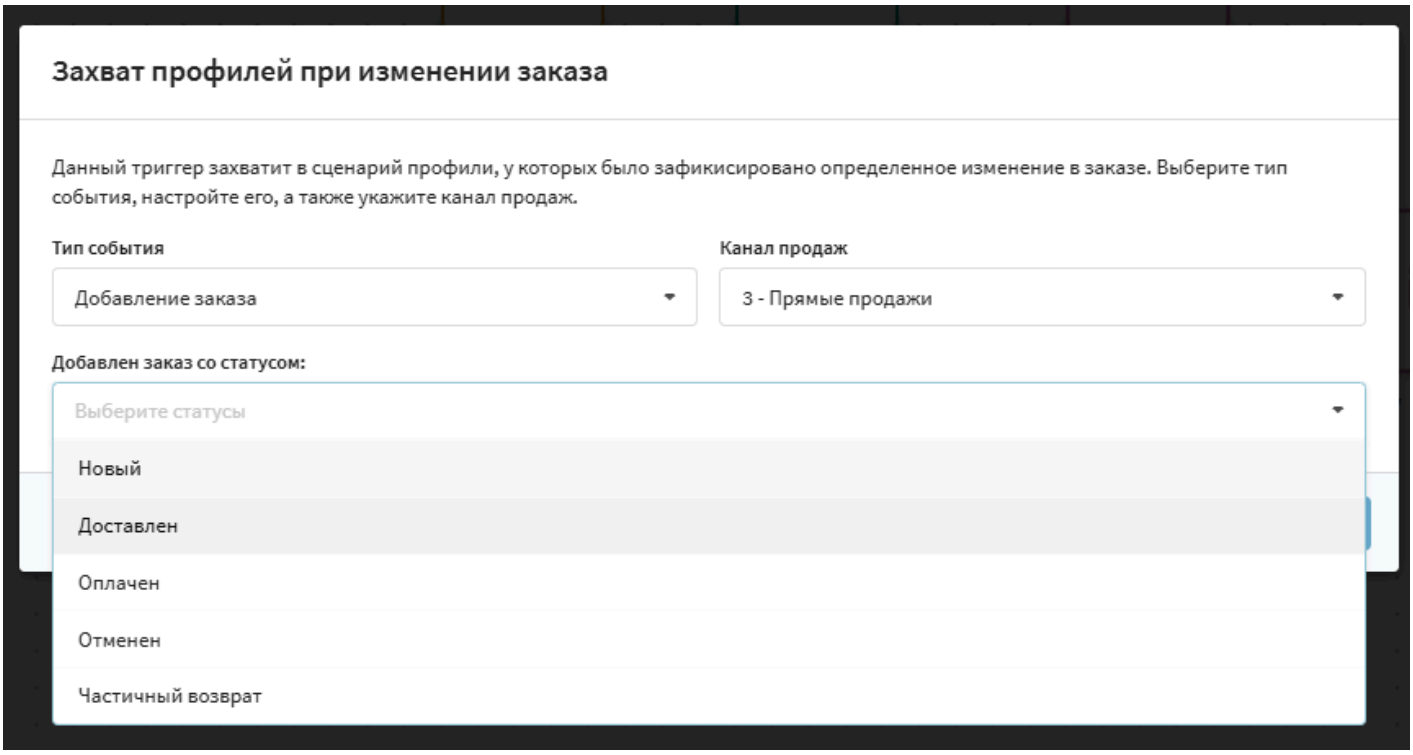
В выпадающем списке укажите тип события, канал продаж и статус заказа.

Доступны следующие варианты типов событий:



- **Добавление заказа** — триггер сработает при добавлении в маркет заказа с одним из указанных статусов.
- **Изменение статуса заказа** — триггер сработает, если статус существующего заказа примет одно из выбранных значений.

Можно выбрать один или несколько статусов заказа, приводящих к срабатыванию триггера:



TIP

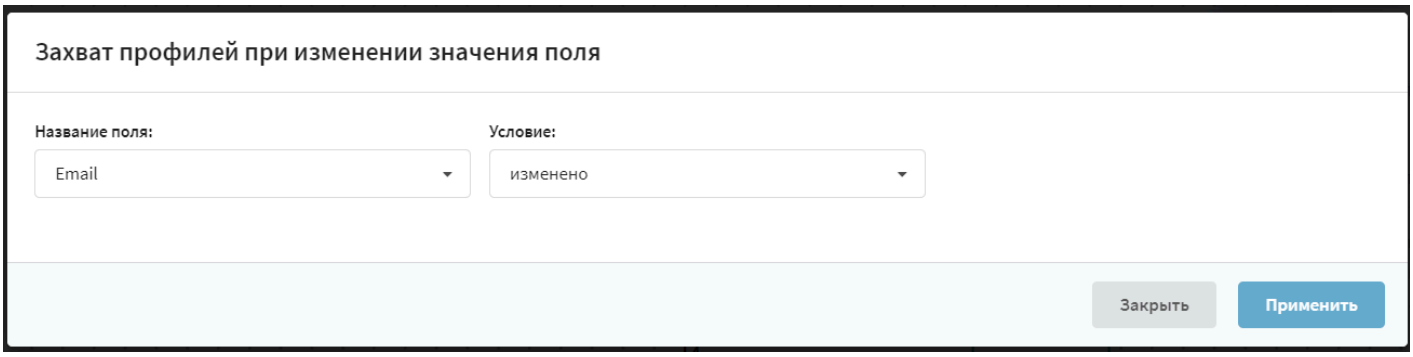
Настройка заказов подробно описана в **отдельной статье**.

Изменение значения поля

	Триггер активируется, когда заданное [**поле профиля**](/user-guide/profiles-and-databases/66420#standard-fields) принимает нужное значение.

Подробнее

Выберите поле профиля, условие и ожидаемое значение:



Условия различаются в зависимости от типа поля:

Условие	Тип поля	Описание
изменено	все	Любое изменение в поле профиля

Условие	Тип поля	Описание
равно	все, кроме *теги*, *объект*, *массив объектов*	Новое значение равно ожидаемому
не равно	все, кроме *теги*, *объект*, *массив объектов*	Новое значение не равно ожидаемому
не пусто е	все, кроме *теги*	Поле получило любое не пустое значение
пусто е	все, кроме *теги*	Все значения удалены для поля профиля
добавлен любой тег из списка	*теги*	Список определяется в триггере
удалён любой тег из списка	*теги*	Список определяется в триггере
свойство изменилось	*объект*	Указывается путь к ключу объекта. Если значение ключа изменится, сработает триггер.
свойство равно	*объект*	Указывается путь к ключу объекта и ожидаемое значение. Если значение этого ключа примет ожидаемое значение, сработает триггер.

Условие	Тип поля	Описание
свойство не равно	*объект*	Указывается путь к ключу объекта и ожидаемое значение. Если значение этого ключа не примет ожидаемое значение, сработает триггер.

свойство элемента изменилось	*массив объектов*	Указывается путь к элементу массива. Если свойство этого элемента изменится, сработает триггер.
		
		При выборе пути к свойству элемента вы также выбираете тип данных, к которому приводить сравниваемое значение. Доступно сравнение таких типов данных, как **string**, **int**, **float**, и **bool**. Сравнение сложных типов данных (объект, массив) невозможно. В примере ниже вы не сможете отследить изменения в свойствах элементов `key_1` and `key_4`.
		`[`
		`{`
		`“key_1”: [1,2,3,4,5],`
		`“key_2”: 2`
		`},`
		`{`
		`“key_3”:“value”,`
		`“key_4”: [1,2,3,4,5]`
		`}`
		`]`
		Обратите внимание, выпадающий список содержит доступные пути к свойству поля, но индекс любого элемента по умолчанию равен 0 — `$[0].key_3`. Чтобы задать путь к нужному элементу, введите индекс вручную в формате jsonpath — `$[1].key_3`.

свойство элемента равно	*массив объектов*	Указывается путь к элементу массива и ожидаемое значение. Если свойство этого элемента примет ожидаемое значение, сработает триггер.
свойство элемента не равно	*массив объектов*	Указывается путь к элементу массива и ожидаемое значение. Если свойство этого элемента не примет ожидаемое значение, сработает триггер.
добавлен элемент	*массив объектов*	В массив добавляется новый элемент.
удален элемент	*массив объектов*	Из массива удаляется элемент.

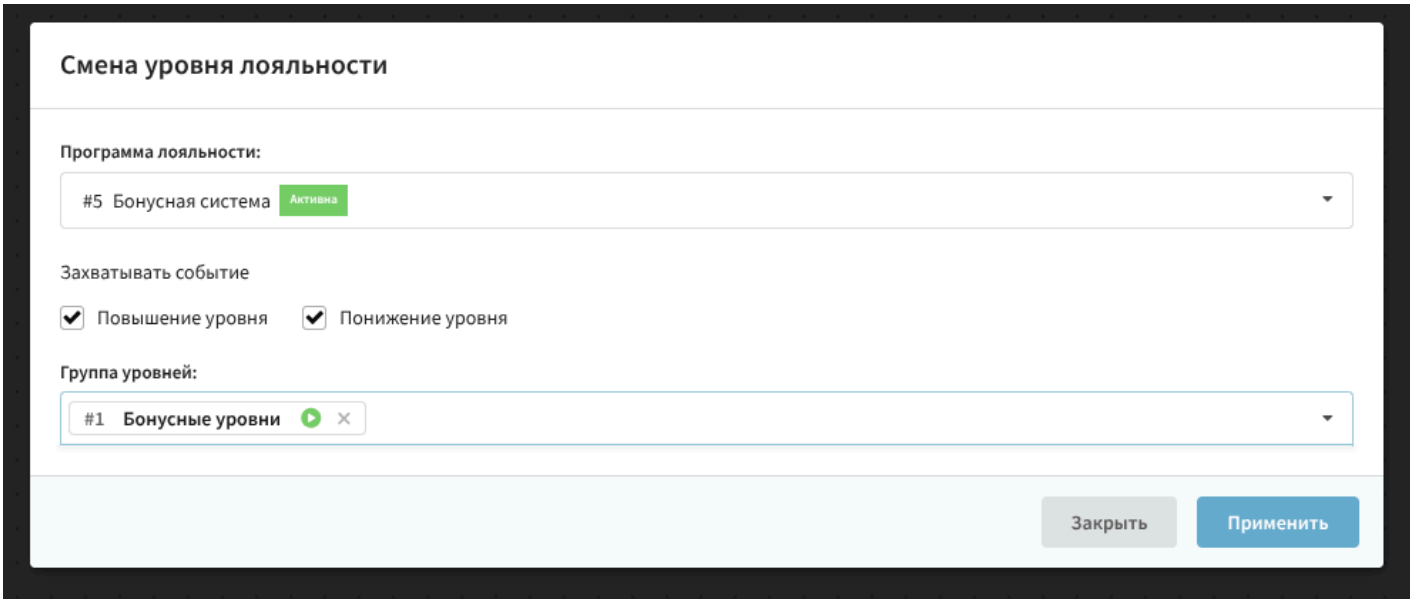
CAUTION

Такой тип триггера можно использовать со ссылками в шаблонах, на которые настроено **обновление полей профиля**. Обратите внимание, что если помимо обновления поля к **переходам по ссылке** привязаны другие триггеры, они также сработают.

Смена уровня лояльности

	Триггер активируется, когда профиль переходит на другой уровень в [**программе лояльности**](/user-guide/loyalty-programs).

Подробнее

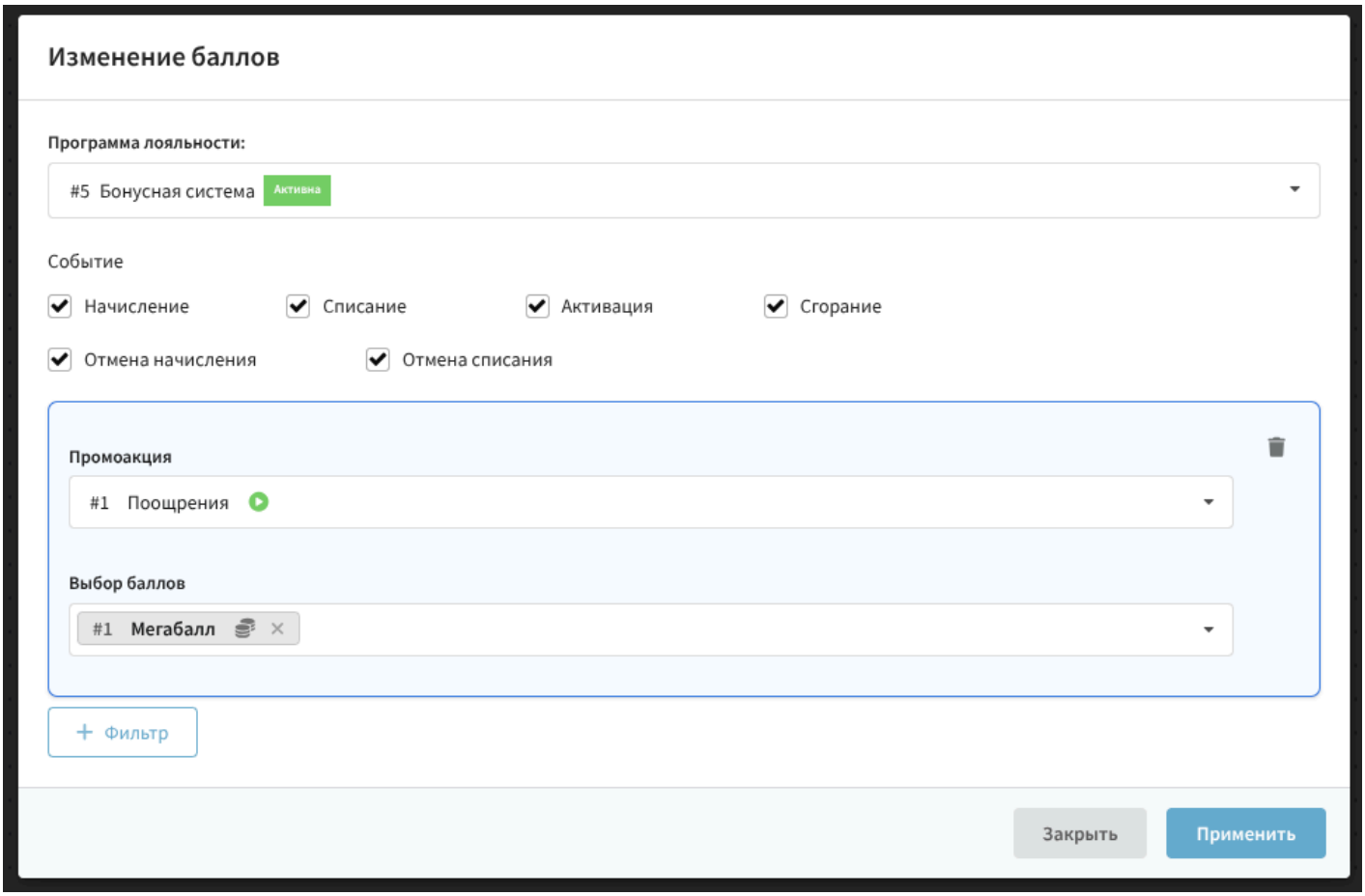


Для настройки триггера выберите заранее созданную **программу лояльности** и **группы уровней**. Вы можете настроить срабатывание триггера при повышении, понижении или любом изменении уровня.

Изменение баллов

	Триггер активируется, когда у профиля изменяется количество баллов в [**программе лояльности**](/user-guide/loyalty-programs).

Подробнее



Для настройки триггера выберите заранее созданную **программу лояльности** и **действия**, которые будут активировать триггер:

- Начисление
- Списание
- Активация
- Сгорание
- Отмена начисления
- Отмена списания

Вы также можете добавить фильтр, чтобы выбрать промоакцию и вид баллов, для которых будет срабатывать этот триггер.

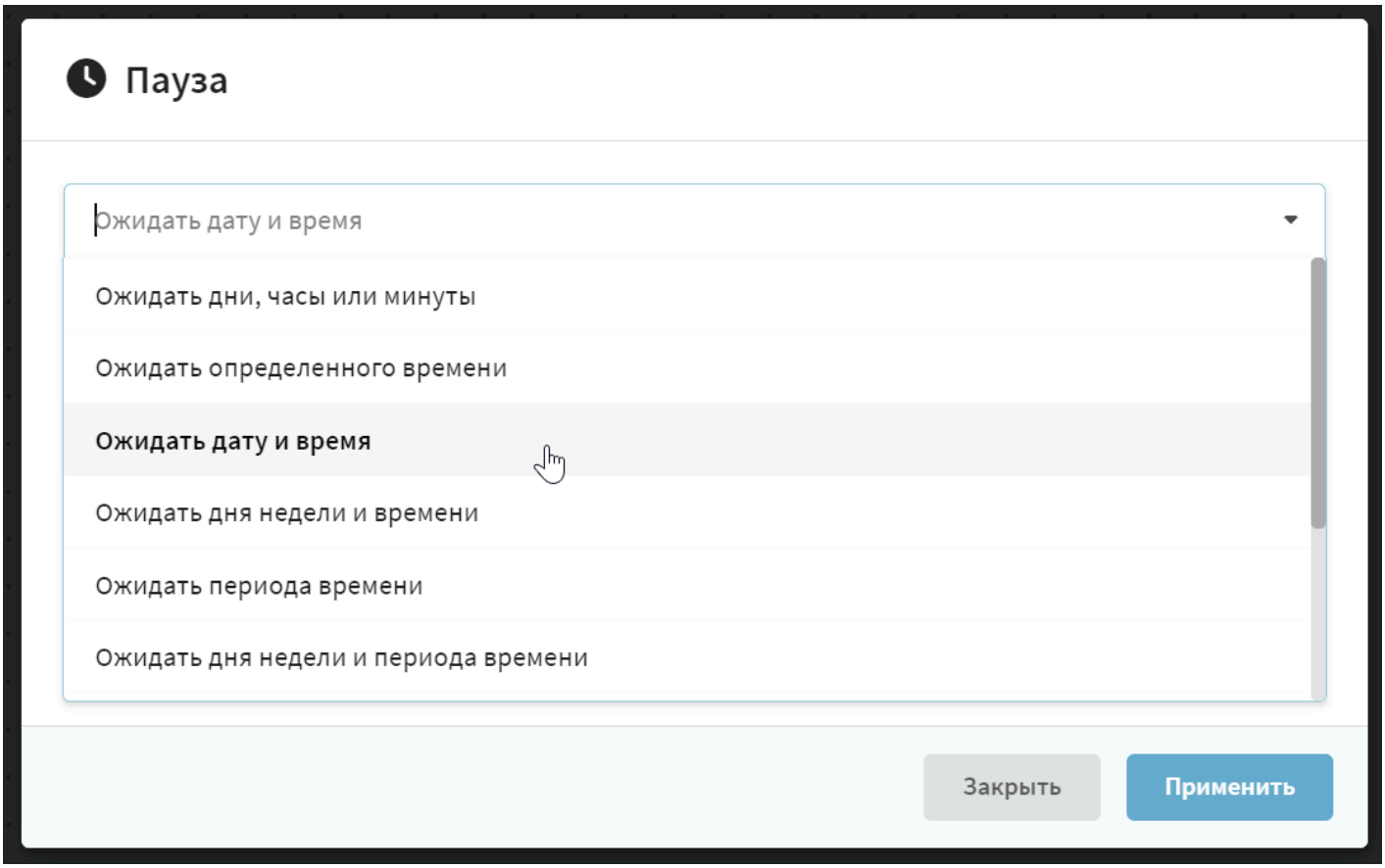
Операторы

Пауза

	Устанавливает интервалы между действиями сценария. С помощью этого узла вы можете настроить время запуска узла, следующего за ним. Это позволяет, в первую очередь, более деликатно подходить к вопросу времени коммуникации с клиентом. Также узел может ожидать определенных событий: начала рекламной акции, праздника и т.д.

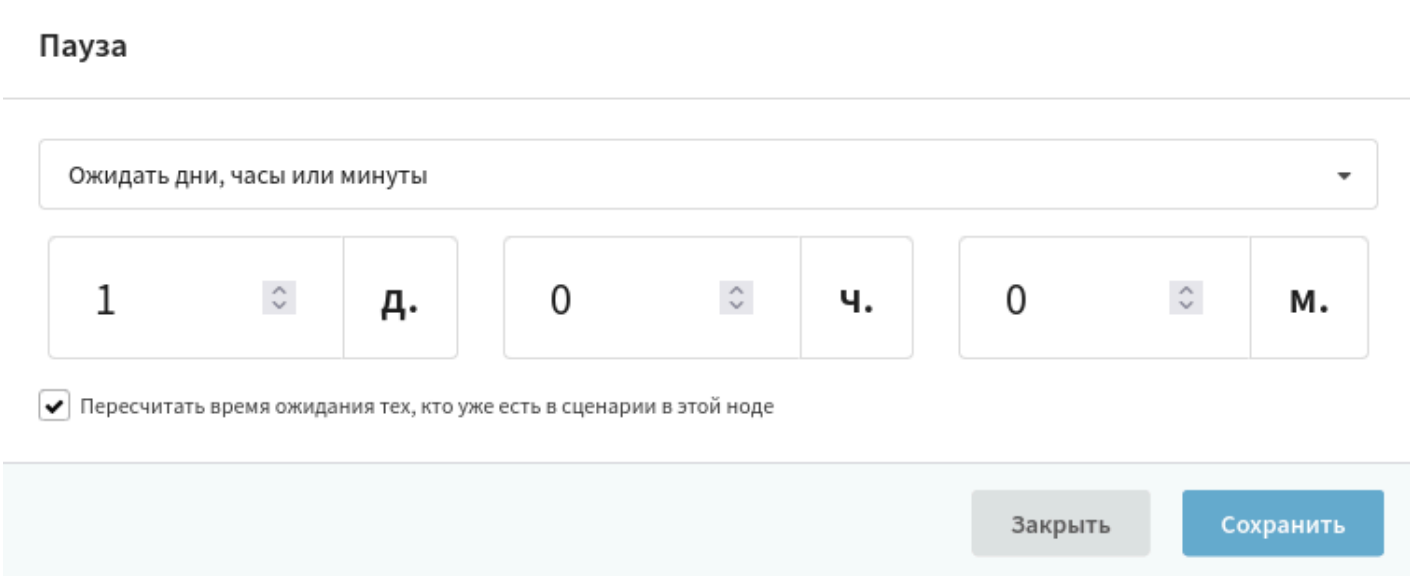
Подробнее

Доступны следующие варианты:



Пауза относительно предыдущего действия

Чтобы ожидать определённое время после захвата профиля клиента сценарием или после выполнения предыдущего действия, выберите вариант “**Ожидать дни, часы или минуты**”:



TIP: Пересчитать время ожидания

При использовании функции пересчета времени ожидания, платформа будет пересчитывать время паузы для тех, кто уже есть в этом узле сценария. Например, изначально вы установили паузу длительностью 10 дней, а затем увеличили время ожидания до 20 дней. В таком случае профиль, который захвачен в сценарий после внесения изменений, будет ожидать 20 дней. Что касается профиля, который уже находился в этом узле 8 дней до внесения изменений, время его ожидания составит еще 12 дней, а не 20.

Если функция отключена, то при изменении узла “Пауза” время ожидания пересчитываться **не будет**.

Пауза до наступления определённого времени

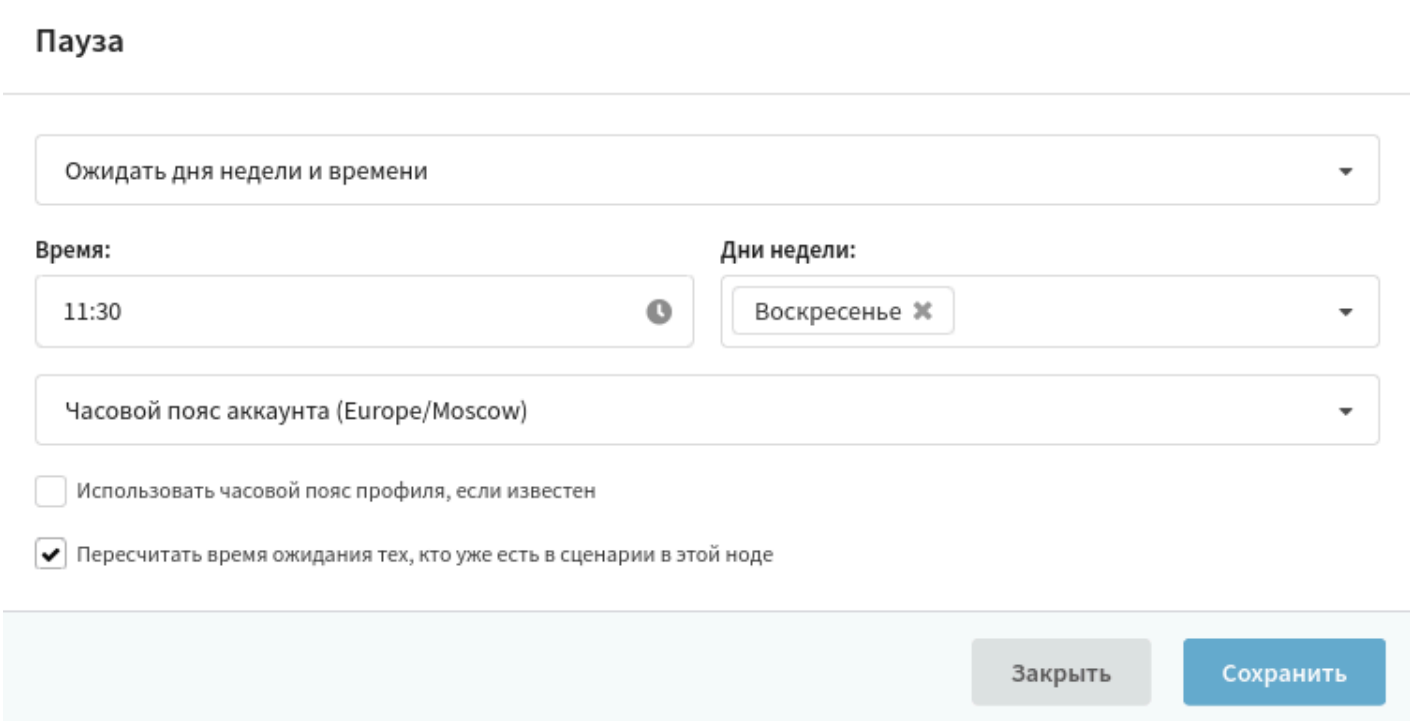
Эти варианты ставят сценарий на паузу до наступления определённого времени суток — в ближайшее выбранное время, в конкретную дату, в ближайший день недели. Причем, если известен часовой пояс профиля, его тоже можно учитывать.

- **Ожидать определенного времени** — указывается точное время перехода профиля на следующий узел.
- **Ожидать дату и время** — указывается точное время и дата перехода на следующий узел.
- **Ожидать дня недели и времени** — настраивается точное время и один или несколько дней недели.
- **Ожидать даты из списка** — выбирается точное время и несколько дат, когда профиль переходит на следующий узел. Каждый профиль при попадании "останавливается" до **ближайшей даты** из списка.

TIP

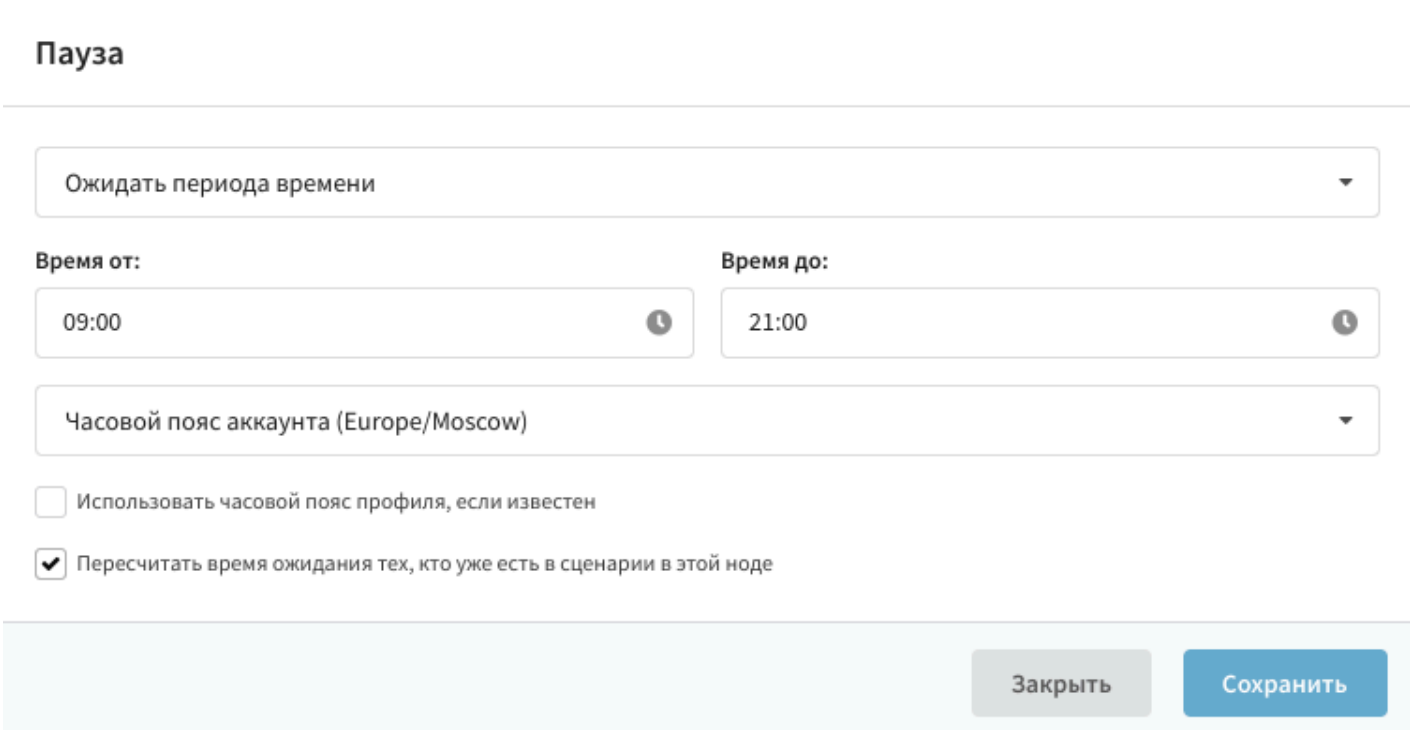
Приведем пример, как работает список дат в узле “Пауза”.

Предположим, вы установили следующие даты: “2023-02-01”, “2023-07-01” и “2023-10-01”. Если профиль попал на узел 1 января 2023, то он продолжит движение по сценарию 1 февраля. Если профиль попал на узел 1 марта 2023, то он пойдет дальше 1 июля. Если же профиль попал на узел в августе, то он будет ожидать до 1 октября. Каждый профиль **единожды** ожидает до ближайшей к моменту попадания на узел даты.

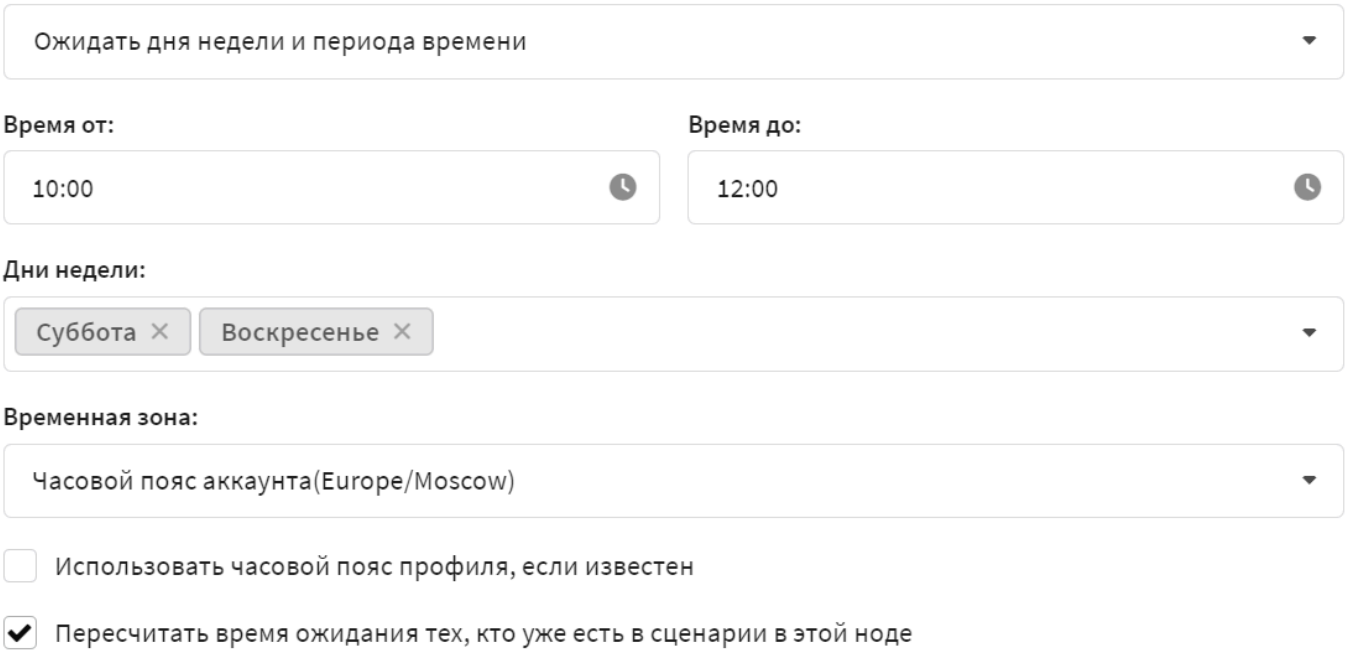


Пауза до наступления определённого периода времени

Вариант **“Ожидать периода времени”** устанавливает часы активности следующего узла сценария. Например, если вы хотите, чтобы определённая рассылка запускалась с 9 утра до 9 вечера:



Вариант **"Ожидать дня недели и периода времени"** устанавливает дни и часы активности следующего узла сценария. Например, если необходимо, чтобы определённая рассылка запускалась только по субботам и воскресеньям с 10 до 12.



Ожидание даты и времени из профиля

Эти настройки позволят поставить сценарий на паузу до наступления даты или времени, которые указаны в одном из полей профиля:

- **Ожидать дату из профиля** — профиль перейдет на следующий узел, когда наступит дата, указанная в выбранном поле, и время, указанное в настройках узла “Пауза”.
- **Ожидать время из профиля (независимо от даты)** — профиль перейдет на следующий узел, когда наступит время, указанное в выбранном поле. Дата может не совпадать с текущей.

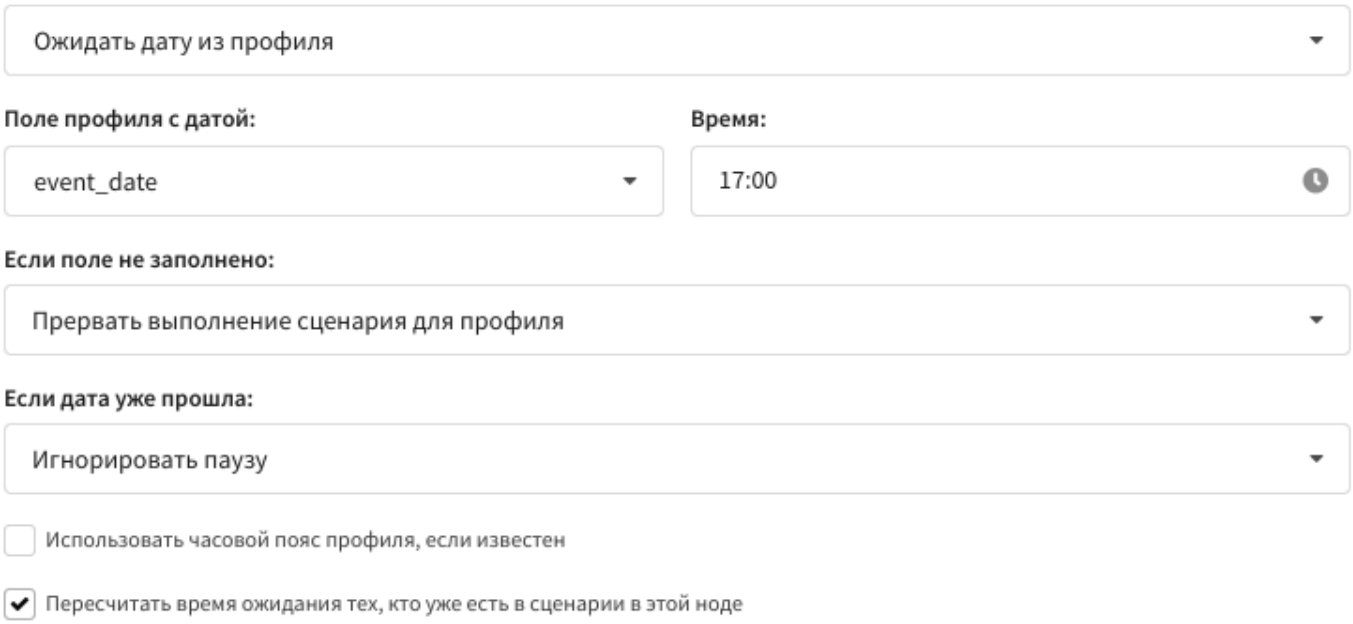
- **Ожидать дату и время из профиля** — профиль перейдет на следующий узел, когда наступит даты и время, указанные в выбранном поле.

Выберите поле в базе с нужными данными. Это может быть дата рождения (`_bdate`), дата регистрации в системе (`_regdate`) или любое кастомное поле с типом “Дата” (например, дата запланированного события).

Далее настройте работу сценария в случае, если у профиля не заполнено выбранное поле:

- **Игнорировать паузу** — профиль сразу перейдет на следующий узел, не дожидаясь наступления определенного времени.
- **Прервать выполнение сценария для профиля** — профиль выйдет из сценария.

Если дата, указанная в поле профиля, уже прошла, вы можете либо сразу перебросить профиль на следующий узел сценария, проигнорировав паузу, либо прервать сценарий для профиля, либо перенести дату на следующий год.



Лучшее время отправки

	Устанавливает перед следующим узлом сценария паузу, рассчитанную на основе данных ML-сервиса. Профиль не перейдёт на следующий узел, пока не наступит оптимальное время для отправки ему сообщения. Оптимальный вариант работы – добавить после узла "Ожидание лучшего времени доставки" узел из категории "Каналы". В таком случае рассылка отправит профилю сообщение в наиболее удобное для него время. </td>

Подробнее

INFO

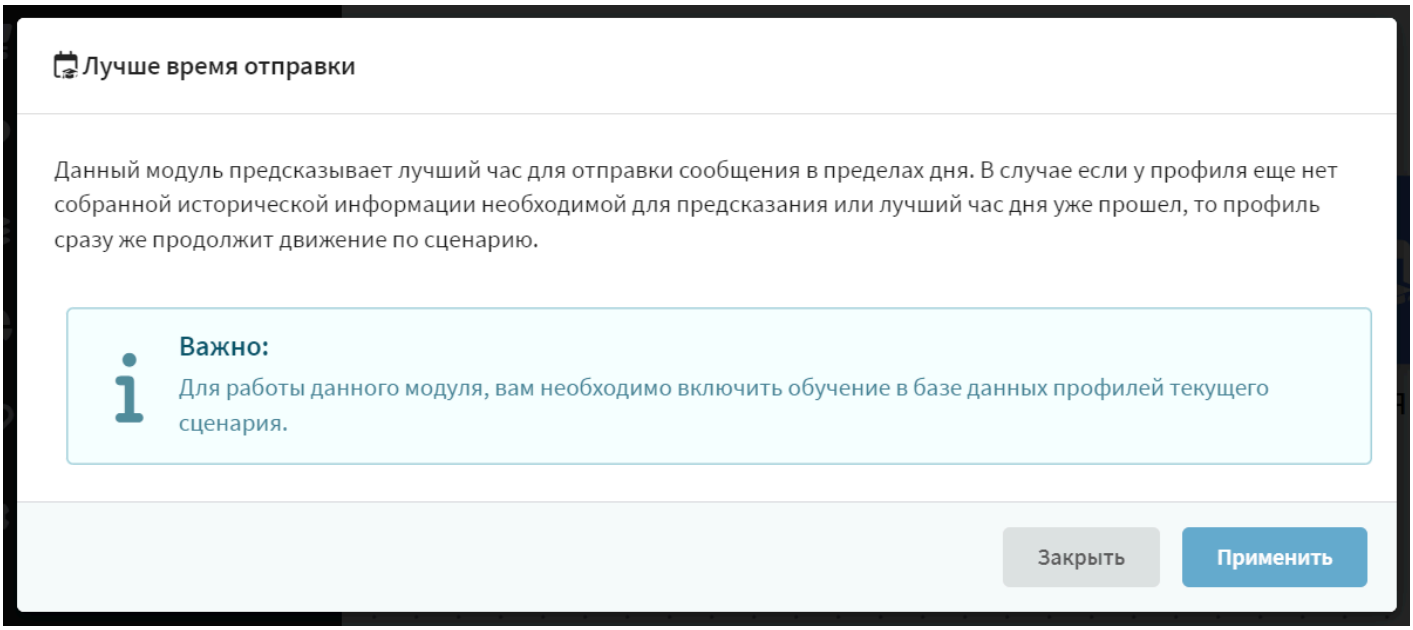
Расчёт лучшего времени происходит через алгоритмы машинного обучения. ML-сервис определяет лучший час, когда вероятность открытия сообщения профилем самая высокая, а также подбирает массив с несколькими хорошими часами отправки.

Узел доступен, если захват профилей в сценарий происходит из базы, для которой включено машинное обучение.

Профиль перейдет на следующий узел, когда наступит его лучший час отправки.

Если в момент попадания профиля в узел лучший час уже прошёл, узел будет дожидаться ближайшего хорошего часа отправки. Если хорошие часы также прошли, то профиль сразу перейдёт на следующий узел.

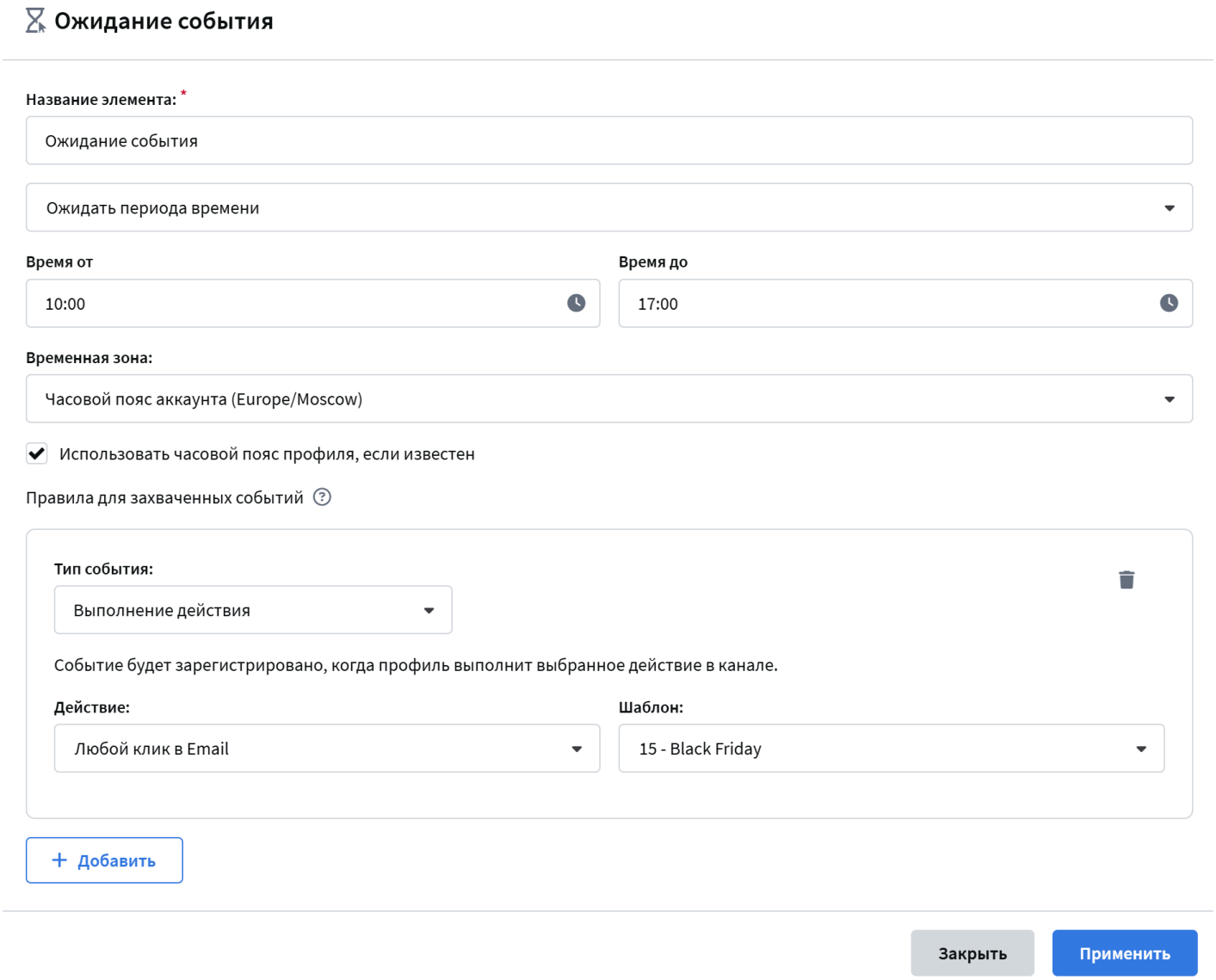
Если в узел попадёт профиль, у которого нет данных по лучшему времени отправки, он также будет сразу же направлен на следующий узел.



Ожидание события

	Останавливает движение профиля по сценарию до тех пор, **пока профилем не будет выполнено определенное действие**. Как только **любое** из указанных событий зарегистрируется платформой, профиль перейдет на следующий узел по верхней (зеленой) ветке. Если же время ожидания превысит заданные значения, профиль пойдет по нижней (красной) ветке.

Подробнее



Введите название узла для отображения в сценарии. Затем настройте время ожидания события. Профиль будет находиться на узле в течение заданного периода. Если за это время ни одно из указанных событий не произойдет, профиль пройдет по нижней ветке сценария.

Настройки времени ожидания:

- **Ожидать дни, часы или минуты** — устанавливает фиксированную продолжительность ожидания события. Профиль перейдет по нижней ветке, если событие не произойдет в течение указанного времени.

  ```
  ![](/img/attachments/9407058/days-hours-minutes.png)
  ```

- **Ожидать определенного времени** — профиль перейдет по нижней ветке, если событие не произойдет до указанного времени.

  ```
  ![](/img/attachments/9407058/certain-time.png)
  ```

- **Ожидать дату и время** — профиль перейдет по нижней ветке, если событие не произойдет до конкретной даты и времени.

  ```
  ![](/img/attachments/9407058/date-and-time.png)
  ```

- **Ожидать дня недели и времени** — настраивается один или несколько дней недели и точное время. Профиль перейдет по нижней ветке, если событие не произойдет до ближайшего выбранного дня и времени.

  ```
  ![](/img/attachments/9407058/day-and-time.png)
  ```

- **Ожидать даты из списка** — выбирается несколько дат и время. Профиль перейдет по нижней ветке, если событие не произойдет до ближайшей даты из списка. Каждый профиль ожидает до ближайшей к моменту попадания на узел даты.

  ```
  ![](/img/attachments/9407058/date-list.png)
  ```

- **Ожидать дату из профиля** — профиль перейдет по нижней ветке, если событие не произойдет до даты, указанной в выбранном поле профиля.

  ```
  ![](/img/attachments/9407058/from-profile.png)
  ```

- **Ожидать время из профиля (независимо от даты)** — профиль перейдет по нижней ветке, если событие не произойдет до времени, указанного в выбранном поле профиля. Дата может отличаться от текущей.

  ```
  ![](/img/attachments/9407058/from-profile-witohut-date.png)
  ```

- **Ожидать дату и время из профиля** — профиль перейдет по нижней ветке, если событие не произойдет до даты и времени, указанных в выбранном поле профиля.

  ```
  ![](/img/attachments/9407058/from-profile-date-and-time.png)
  ```

Далее добавьте события, которые будет ожидать платформа. Если наступит **любое** из этих событий, профиль пойдет дальше по сценарию по зеленой ветке. Полный список событий представлен в таблице ниже:

Тип события	Описание
Выполнение действия	Событие будет зарегистрировано, когда профиль выполнит выбранное действие в канале. Полный список действий можно найти в описании узла ["Выполнение действия"](/user-guide/automation-scenarios/9407058#action-execution).
Обновление через форму	Событие будет зарегистрировано при обновлении полей профиля после заполнения формы. Для активации этого триггера требуется форма, которая будет импортировать захваченные профили в выбранную для данного сценария базу данных.
Добавление/ изменение подписки	Событие будет зарегистрировано при добавлении или изменении статуса подписки профиля.
Изменение в статическом сегменте	Событие будет зарегистрировано, когда для профиля будет зафиксировано определенное изменение в сегменте.
Изменение связи	Событие будет зарегистрировано при изменении связей профиля.

Активация пикселя	Событие будет зарегистрировано при активации пикселя для профиля.
Заполнение формы	Событие будет зарегистрировано при заполнении формы пользователем с существующим профилем.
Изменение/ добавление заказа	Событие будет зарегистрировано при изменении статуса заказа у профиля. Для активации этого триггера необходим канал продаж, привязанный к базе данных, выбранной для данного сценария.

Условие

	Реализует логический выбор следующего шага сценария для профилей клиентов. Условие выбора определяется правилами **Сегментации**. При выполнении условия профиль клиента продолжит двигаться по сценарию вправо. Если условие не выполнено, профиль переместится по сценарию вниз, и для него может быть настроена альтернативная последовательность действий. Узел может также ожидать выполнения условия в течение определённого времени.

TIP

Если связать несколько узлов **“Условие”** по вертикали, можно создать несколько альтернативных цепочек в одной — в зависимости от результатов проверок.

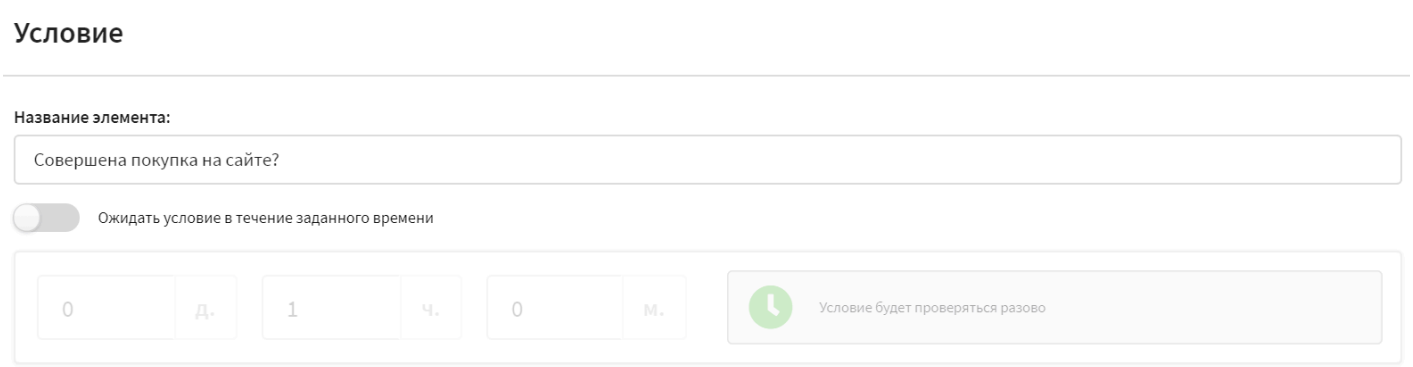
Подробнее

Основные настройки

Введите название узла для отображения в сценарии.

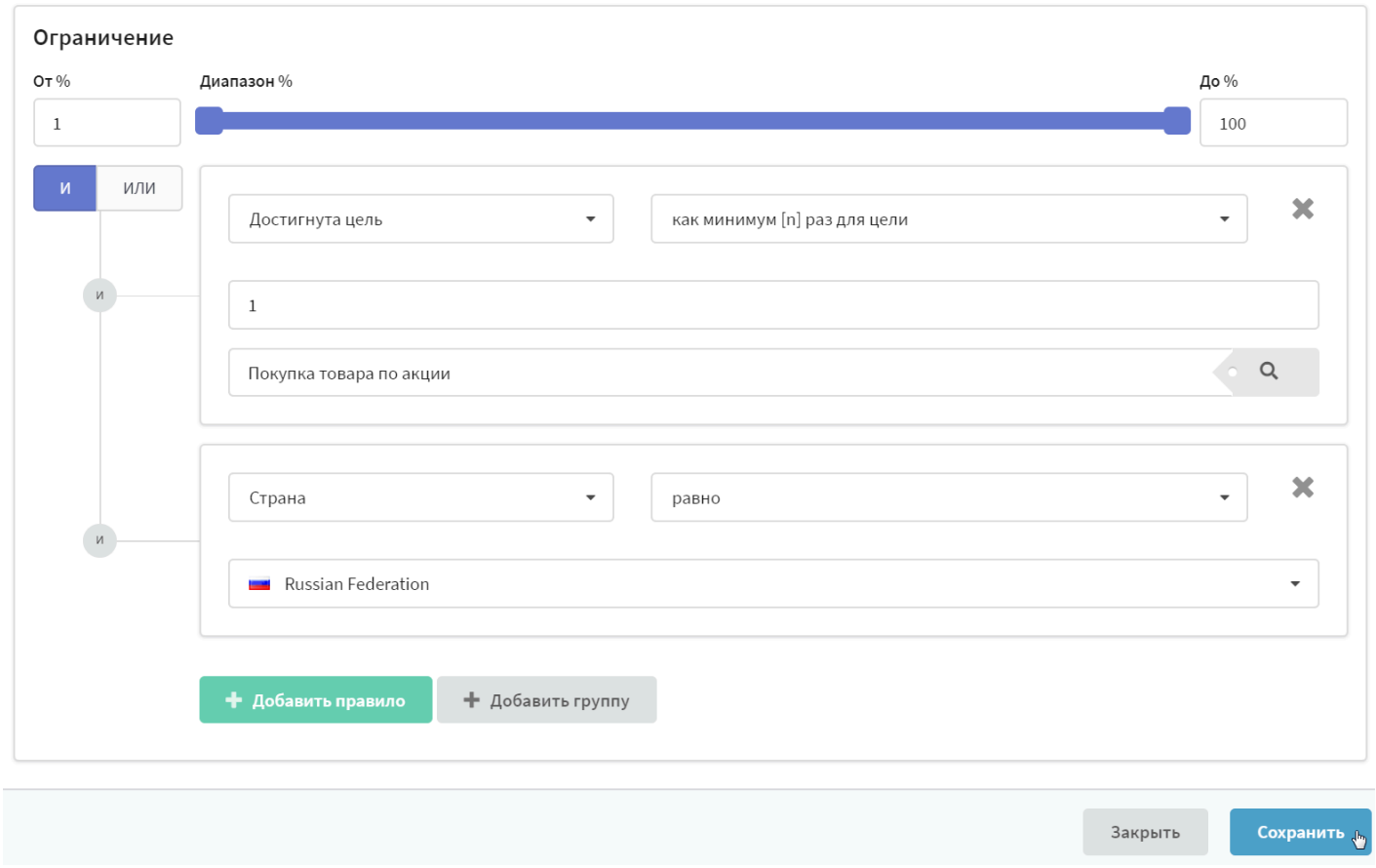
TIP

Если необходимо динамически проверять выполнение условия, включите опцию “**Ожидать условие в течение заданного времени**”. Так проверка будет происходить несколько раз. Если время ожидания фиксированное, не включайте эту опцию, а вместо этого добавьте перед условием узел “**Пауза**”.



Правила выполнения условия

Условие составляется правилами и группами правил, аналогичными правилам сегментации. Подписчики могут распределяться по личным данным, данным подписок, истории действий:



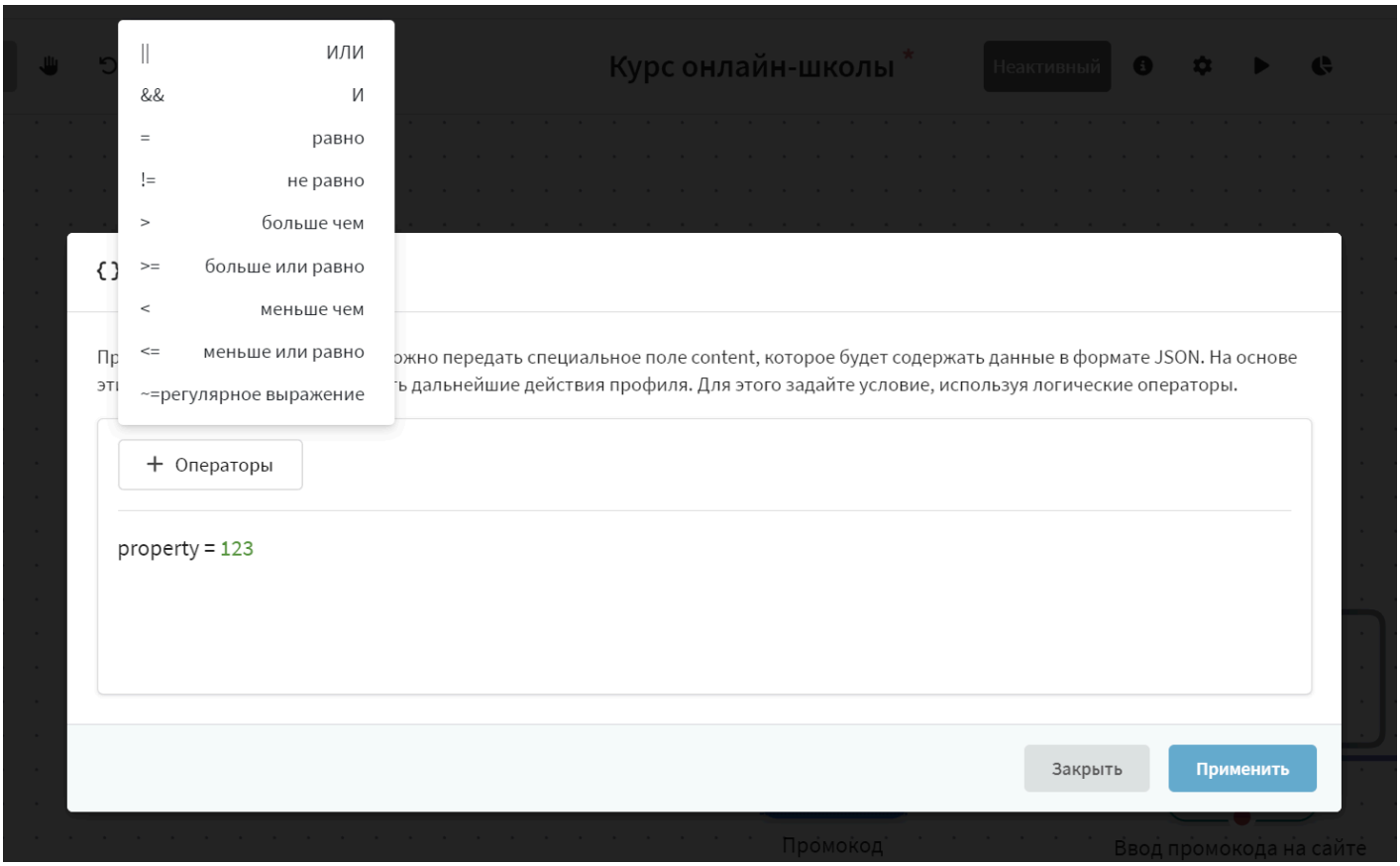
JSON фильтр

	В случае вызова сценария через API можно передать специальное поле `content`, с неопределенной структурой в формате JSON (её определяете вы). Эти данные можно использовать для вставки в сообщения, возврата в Action Hook или для построения условий в сценарии. Например, когда пользователь совершает какое-то действие, веб-сайт (или приложение) активирует сценарий и передает туда все данные, связанные с действием. Узел "JSON фильтр" определяет дальнейшие действия для профиля на основе проверки этих данных.

Подробнее

JSON-фильтр работает так же, как и обычное условие. Можно неоднократно использовать этот узел для построения сложных схем.

Вы можете задавать условия в редакторе кода с помощью логических операторов и выражений.



Пример поля `"content"` в API запросе:

```
{
  "content": {
    "customer_id": "12345",
    "total_purchases": 5,
    "last_purchase_amount": 250,
    "average_purchase_amount": 100,
    "preferred_category": "electronics",
    "loyalty_points": 300,
    "membership_status": "premium",
    "discount_coupons_used": 2,
    "active_coupon": "win0513",
    "promo": null
  }
}
```

Обратите внимание, ключ JSON-объекта указывается без кавычек (например, `membership_status="premium"`).

Оператор	Пример	Описание
`||` ИЛИ	`(preferred_category = "electronics") || (preferred_category = "computers")`	Условие выполнится, если клиент интересуется категориями “Электроника” или “Компьютеры”.

Оператор	Пример	Описание
`&&` И	`(total_purchases >= 5) && (average_purchase_amount >= 100)`	Условие выполнится, если клиент совершил 5 или более покупок и средний чек превышает 100.
`=` Равно	`membership_status = 'premium'`	Условие выполнится, если у клиента premium-статус.
`!=` Не равно	`membership_status != 'standard'`	Условие выполнится, если у клиента не standard-статус.
`>` Больше чем	`last_purchase_amount > (2 * average_purchase_amount)`	Условие выполнится, если у клиента сумма последней покупки превышает в два раза среднюю сумму покупок.
`>=` Больше или равно	`last_purchase_amount >= 250`	Условие выполнится, если сумма последней покупки равна или превышает 250.
`<` Меньше чем	`loyalty_points < 500`	Условие выполнится, если количество бонусных баллов меньше 500.
`<=` Меньше или равно	`loyalty_points <= 500`	Условие выполнится, если количество бонусных баллов 500 или меньше.
`~=` Регулярное выражение	`active_coupon ~= '^win'`	Условие выполнится, если промокод начинается с “win”.
`!~=` Обратное регулярное выражение	`active_coupon !~= '^win'`	Условие выполнится, если промокод НЕ начинается с “win”.
`+` Сложение	`(loyalty_points + bonus_points) < 200'`	Оператор арифметического сложения.
`-` Вычитание	`(loyalty_points - penalty_points) > 100'`	Оператор арифметического вычитания.
`*` Умножение	`(bonus_points * years) > 360`	Оператор арифметического умножения.
`/` Деление	`(total_purchase / 365) > 5`	Оператор арифметического деления.
`%` Деление по модулю	`(total_sum % 15) = 5`	Оператор вычисления остатка от деления.
`^` Возведение в степень	`(promotion_points ^ 3) > 360'`	Оператор возведения числа в степень.
`is defined` Проверка существования поля	`promo is defined`	Условие выполнится, если поле “promo” существует, вне зависимости от того, указано для него значение или нет.
`isnot defined` Проверка отсутствия поля	`promo isnot defined`	Условие выполнится, если поле “promo” отсутствует в коде.
`is null` Проверка пустого значения поля	`promo is null`	Условие выполнится, если у поля “promo” нет значения (его значение равно `null`).
`isnot null` Проверка значения поля	`promo isnot null`	Условие выполнится, если поле “promo” имеет какое-либо значение (равно чему угодно, кроме `null`).

Оператор	Пример	Описание
`contains` Проверка, содержится ли значение справа в поле слева	Строка: `"hello" contains "he"` Массив: `[1, 2, 3] contains 2`	Условие выполнится, если поле “hello” содержит значение “he”.
`in` Проверка, содержится ли значение слева в поле или массиве справа	Строка: `"he" in "hello"` Массив: `2 in [1, 2, 3]`	Условие выполнится, если значение 2 содержится в массиве [1, 2, 3].
`notin` Проверка, НЕ содержит ли строка или массив значение поля слева	`"bye" notin "hello"`	Условие выполнится, если поле “hello” НЕ содержит значение “bye”.

Для обращения к вложенному полю объекта используйте следующий синтаксис: `object.field`.

Для обращения к элементам массива по индексу используйте следующий синтакис: `array.[0].field`.

Примеры использования:

```
// Объект:

{
  "content": {
    "customer_id": "12345",
    "total_purchases": 5,
    "promo_campaign": {
      "tickets": 115
    },
    "tickets_info": [
      {
        "ticket_id": "T001",
        "price": 50,
        "date": "2024-08-30"
      },
      {
        "ticket_id": "T002",
        "price": 75,
        "date": "2024-08-31"
      }
    ]
  }
}

// Обращение к вложенному полю объекта:

promo_campaign.tickets >= 100

// Обращение к первому элементу массива:

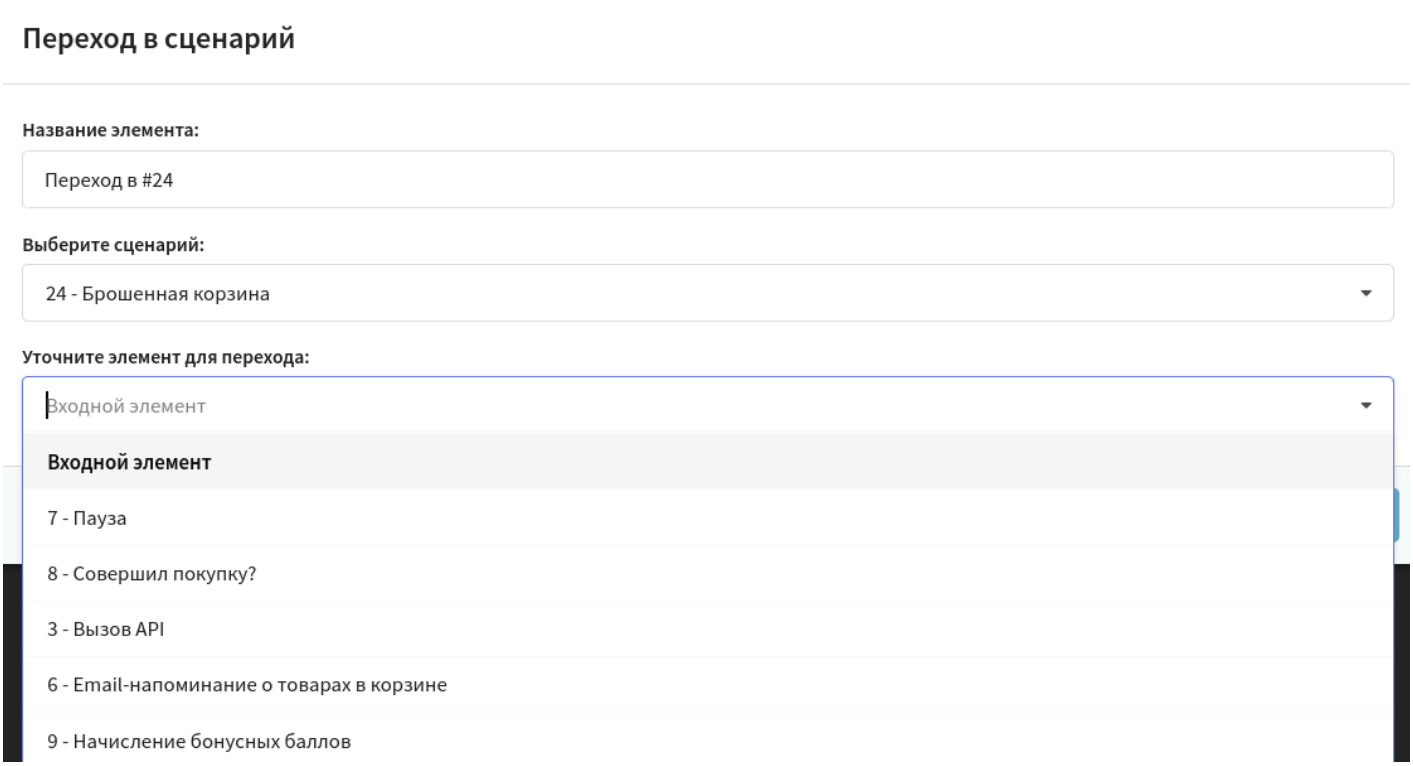
tickets_info.[0].ticket_id
```

Перейти в сценарий

	Отправляет профиль в начало другого сценария или в определённый узел другого сценария. При этом текущий сценарий и сценарий, в который профиль переходит, должны использовать одну базу данных.Если после узла перехода цепочка в сценарии продолжается, то профиль переходит в другой сценарий и в то же время продолжает двигаться по текущему.

Подробнее

Введите **название** узла. Выберите **сценарий,** в который необходимо отправить профиль. Если вы хотите, чтобы профиль перешёл в начало другого сценария, выберите **входной узел** как элемент для перехода. Если вы хотите, чтобы профиль сразу же оказался на опредённом узле другого сценария, выберите этот узел в списке.



Узел распределения

	Создаёт несколько узлов, разделяющих варианты развития сценария для профилей клиентов независимо от каких-либо условий. Доступно распределение перебором и статическое распределение.

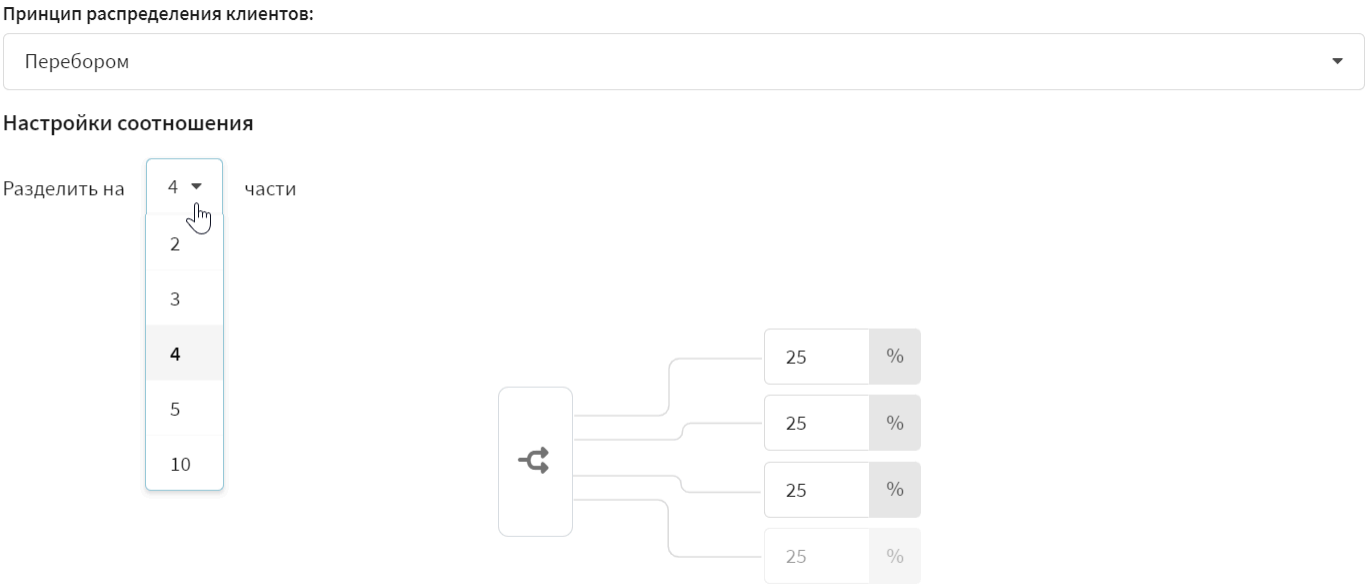
Подробнее

Основные настройки

Выберите принцип распределения и количество частей, на которые будет распределяться аудитория. Процентное соотношение можно менять, редактируя элементы распределения.

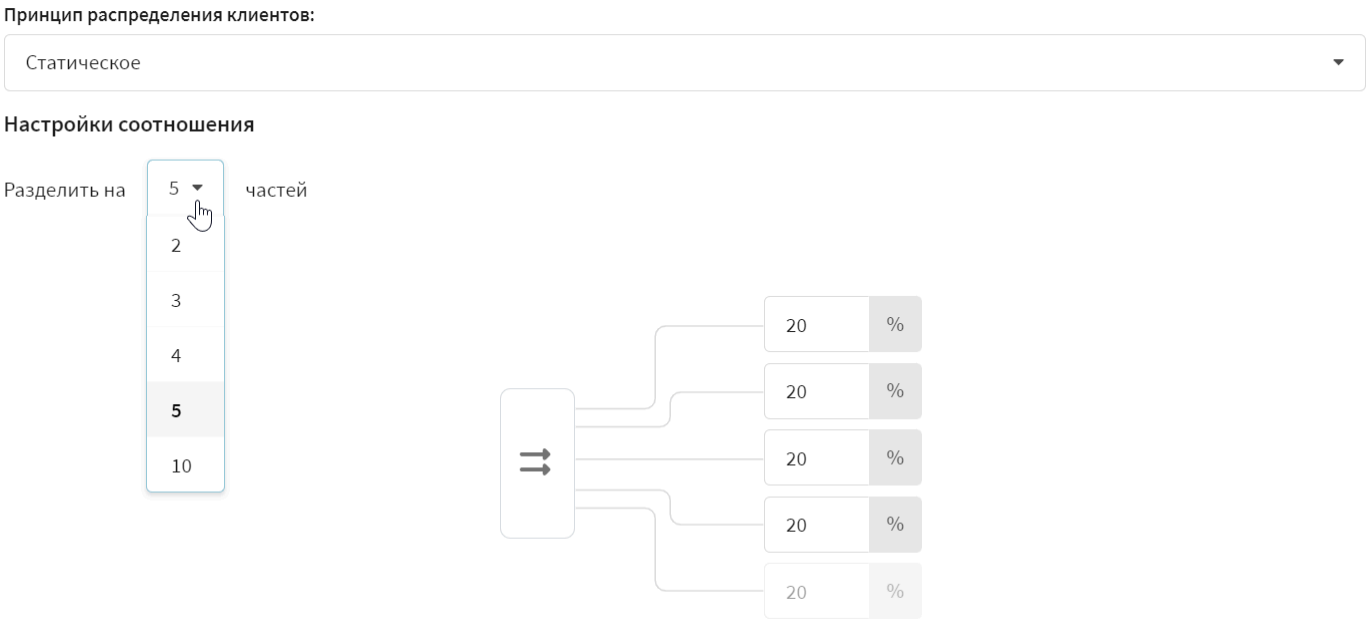
Распределение перебором

Распределение профилей происходит в момент попадания в узел сценария в соответствии с указанными пропорциями. В его основе лежит механизм Smooth Weighted round-robin balancing.



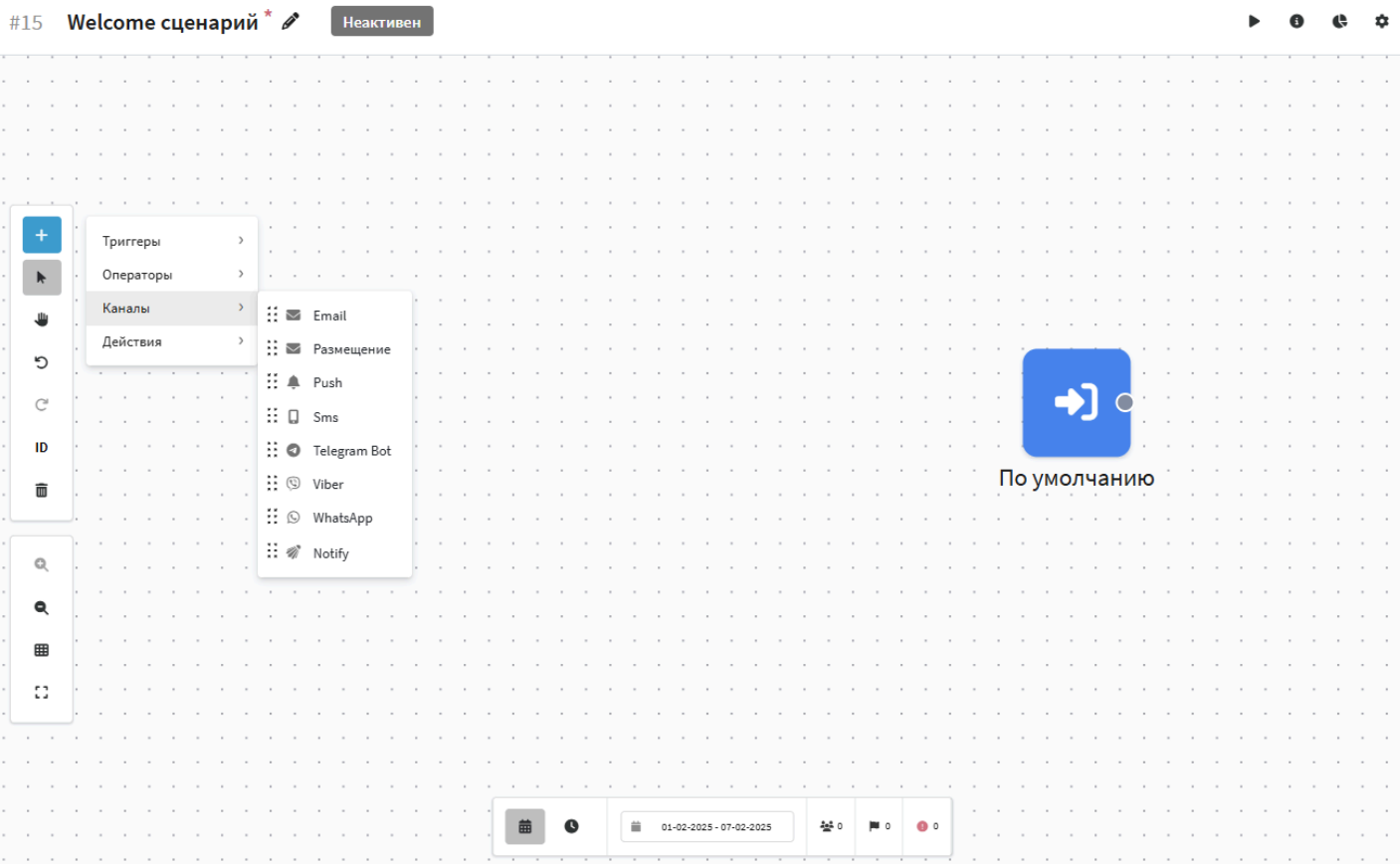
Статическое распределение

При использовании статического распределения профиль всегда будет попадать в один и тот же процент распределения, который нельзя изменить. Статическое распределение **не рекомендуется использовать для небольшого количества профилей (до 10 000)**.



Каналы

Узел этой категории запускают **триггерную рассылку** через определенный канал. Новая рассылка создаётся непосредственно в редакторе узла.



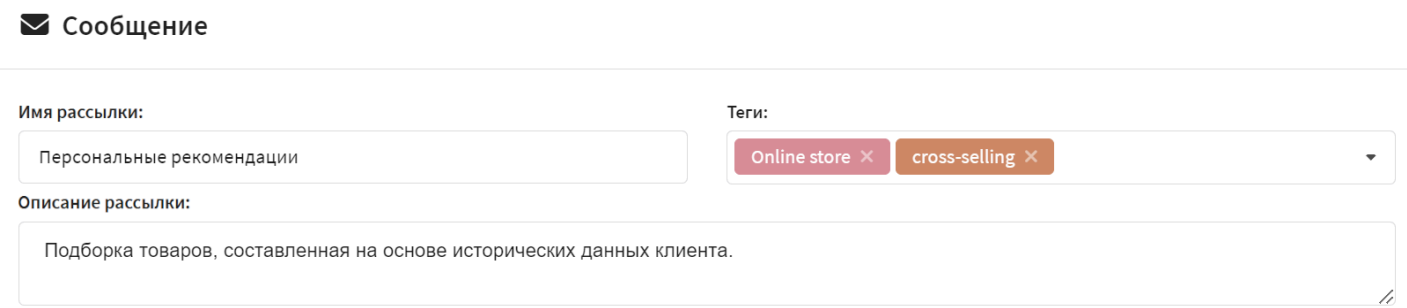
TIP

Узел **“Размещение”** позволяет запустить рассылку-размещение в сценарии. Настройка такой рассылки описана в этой статье.

Подробнее

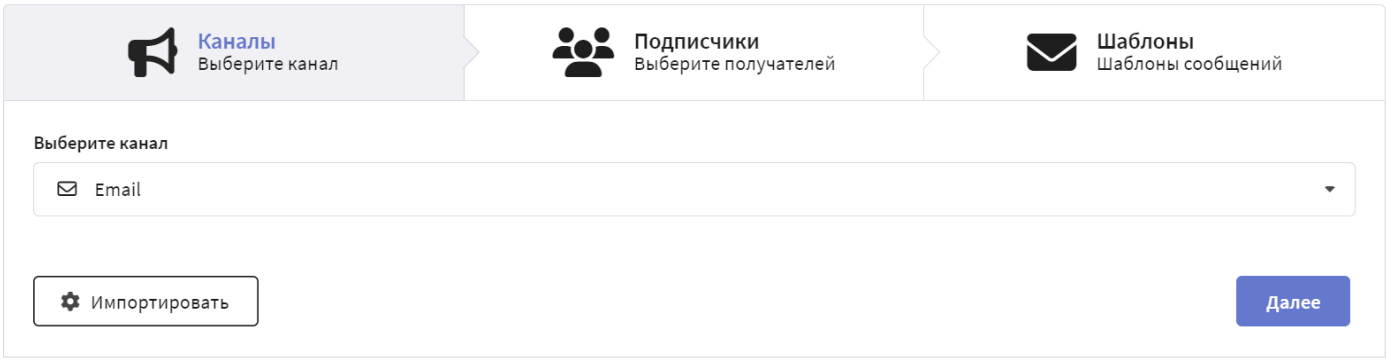
Основные настройки

Введите имя новой рассылки, при необходимости добавьте описание. Здесь же вы можете задать пользовательские теги и установить **высокий приоритет** отправки:

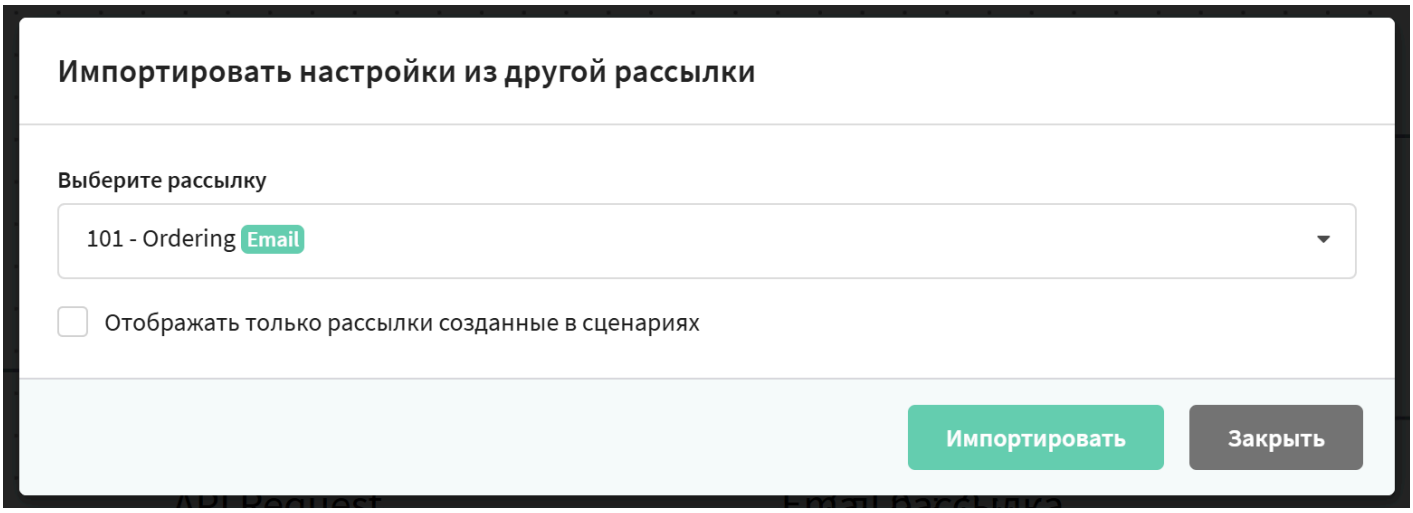


Настройки рассылки

На первой вкладке настроек выберите канал коммуникации: каждая рассылка может отправляться только по одному каналу.

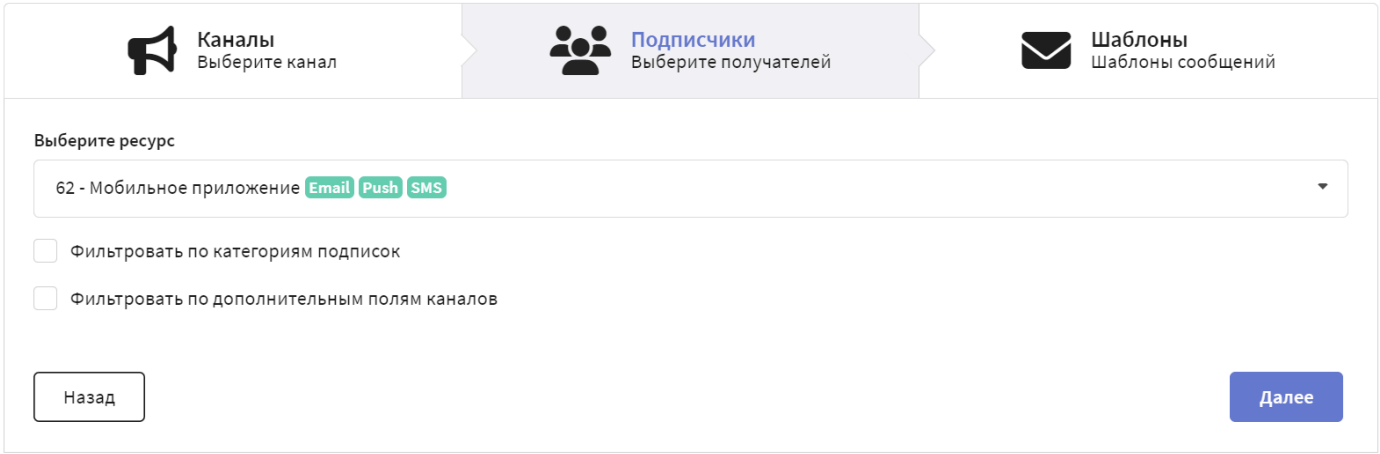


На этой же вкладке вы можете **импортировать** в узел существующую рассылку. Это может быть как отдельная рассылка, так и рассылка, входящая в другие сценарии:



Подписчики

Триггер в цепочке будет использовать контактные данные подписчиков одного из ваших **ресурсов**. Выберите ресурс, на который подписана ваша целевая аудитория и перейдите к следующему шагу:

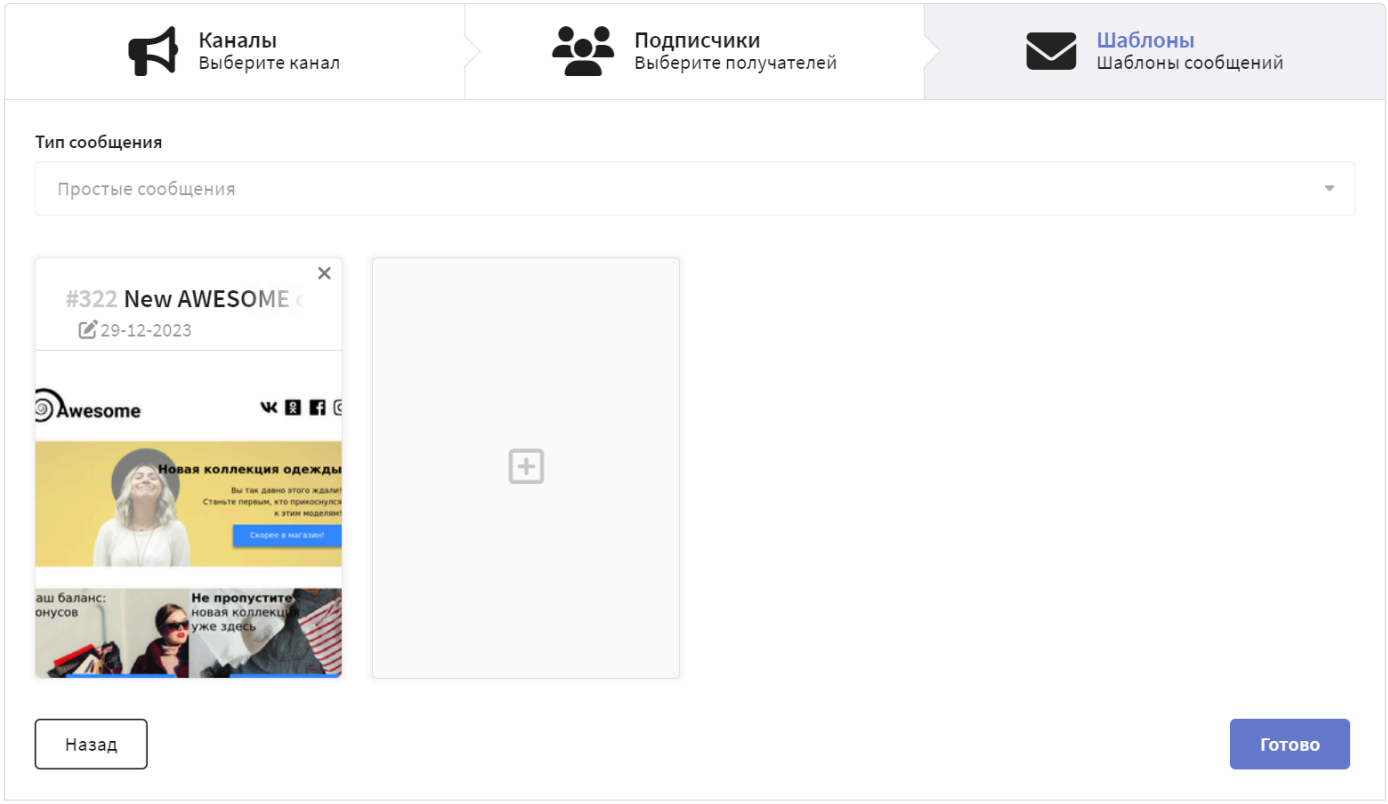


Дополнительно вы можете **отфильтровать получателей по категориям ресурса**. Для этого в ресурсе должны быть созданы категории. В настройках рассылки выберите категории, на которые должен быть подписан профиль, чтобы получить сообщение.

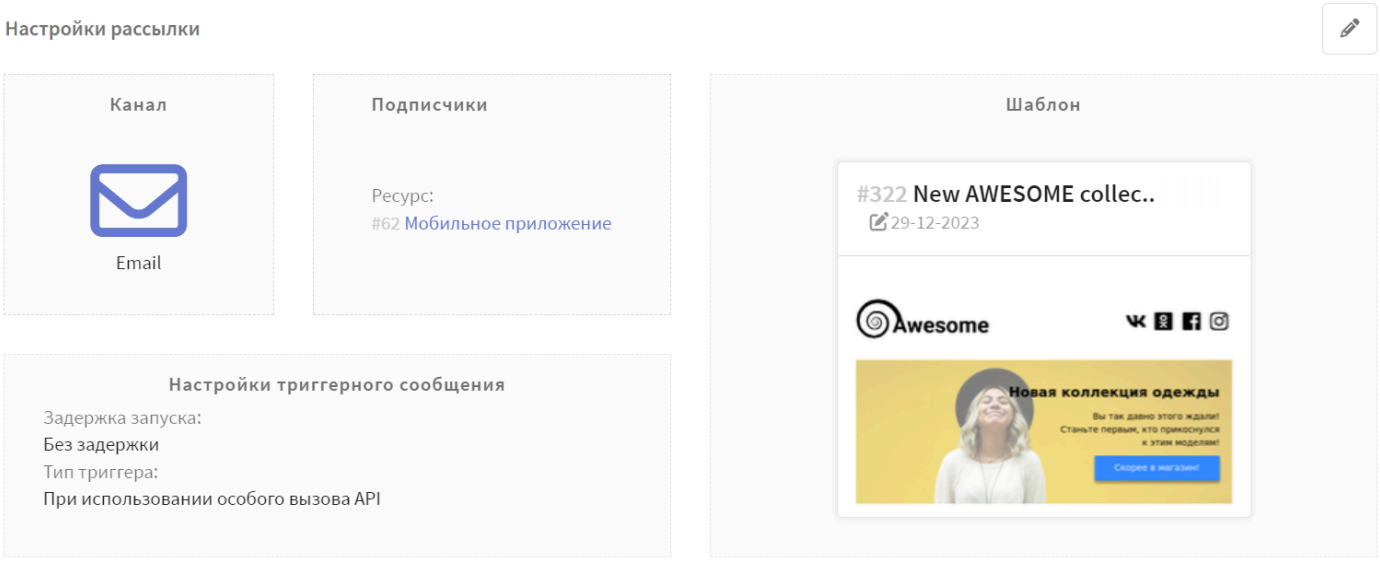
Также можно **отфильтровать получателей рассылки по дополнительным полям канала** в ресурсе. Чтобы создать фильтр по полям, выберите стандартное или дополнительное поле подписки. В поле “Условие отбора” выберите логический оператор и при необходимости укажите значение.

Контент

На третьей вкладке выберите **Шаблон сообщения**, который хотите отправить подписчикам. Есть возможность выбрать сразу несколько шаблонов (до 16 шаблонов в одной рассылки). В таком случаи они будут отправляться в случайном порядке. Как и при создании рассылки, вы можете открыть предпросмотр нужного шаблона и перейти к его редактированию, если это необходимо:

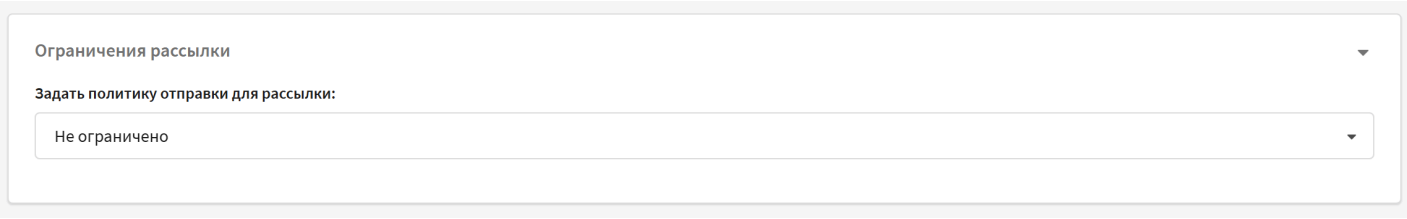


После выбора контента нажмите кнопку “**Готово**”. Настройки рассылки отобразятся на одной вкладке, и вы сможете внести необходимые изменения в любой момент по нажатию на кнопку редактирования:



Ограничения рассылки

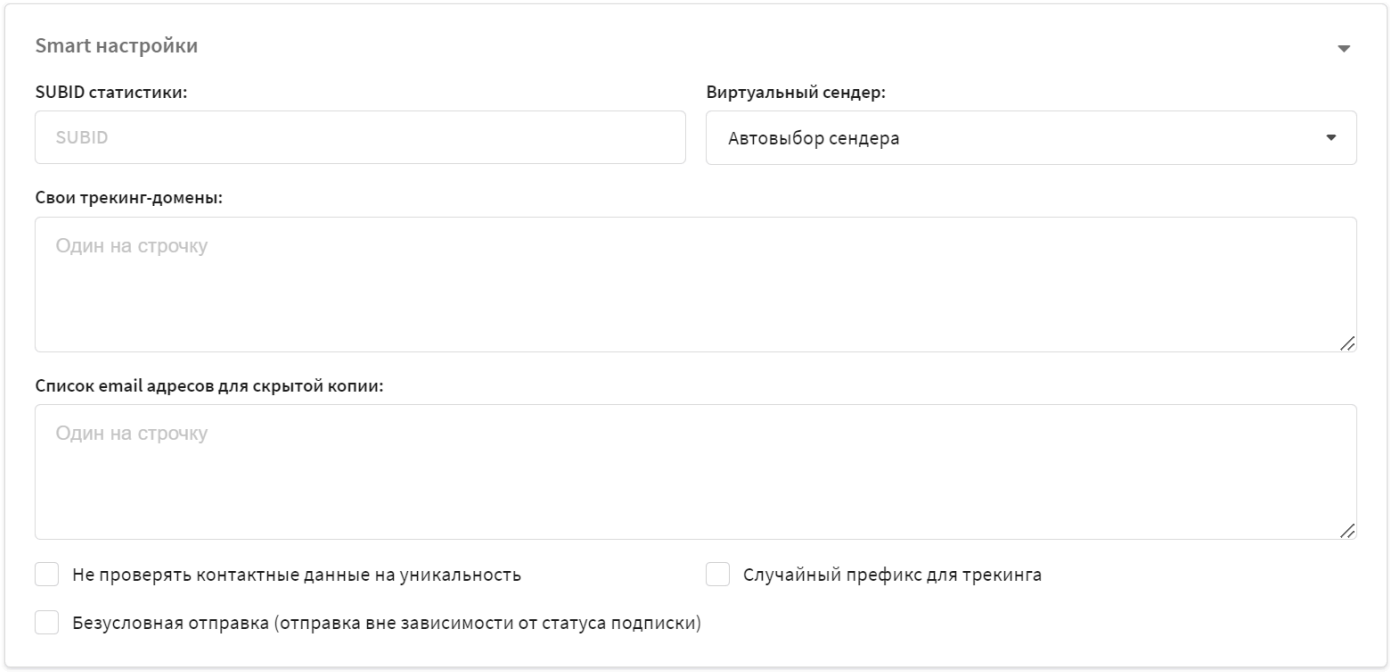
В этом открывающемся блоке вы можете задать **политику отправки** рассылки. По умолчанию применяется политика выбранного ресурса или глобальная политика аккаунта:



Smart настройки

В умных настройках вы можете дополнительно настроить рассылку:

- SUBID статистики
- Выбор виртуального сендера
- Пользовательские домены для отслеживания событий (помимо настроенного в ресурсе)
- Список адресов для скрытой копии к отправляемым письмам
- Другие опции в зависимости от конфигурации платформы



CAUTION

Если при отправке рассылки возникла ошибка, профиль продолжит идти по сценарию.

Действия

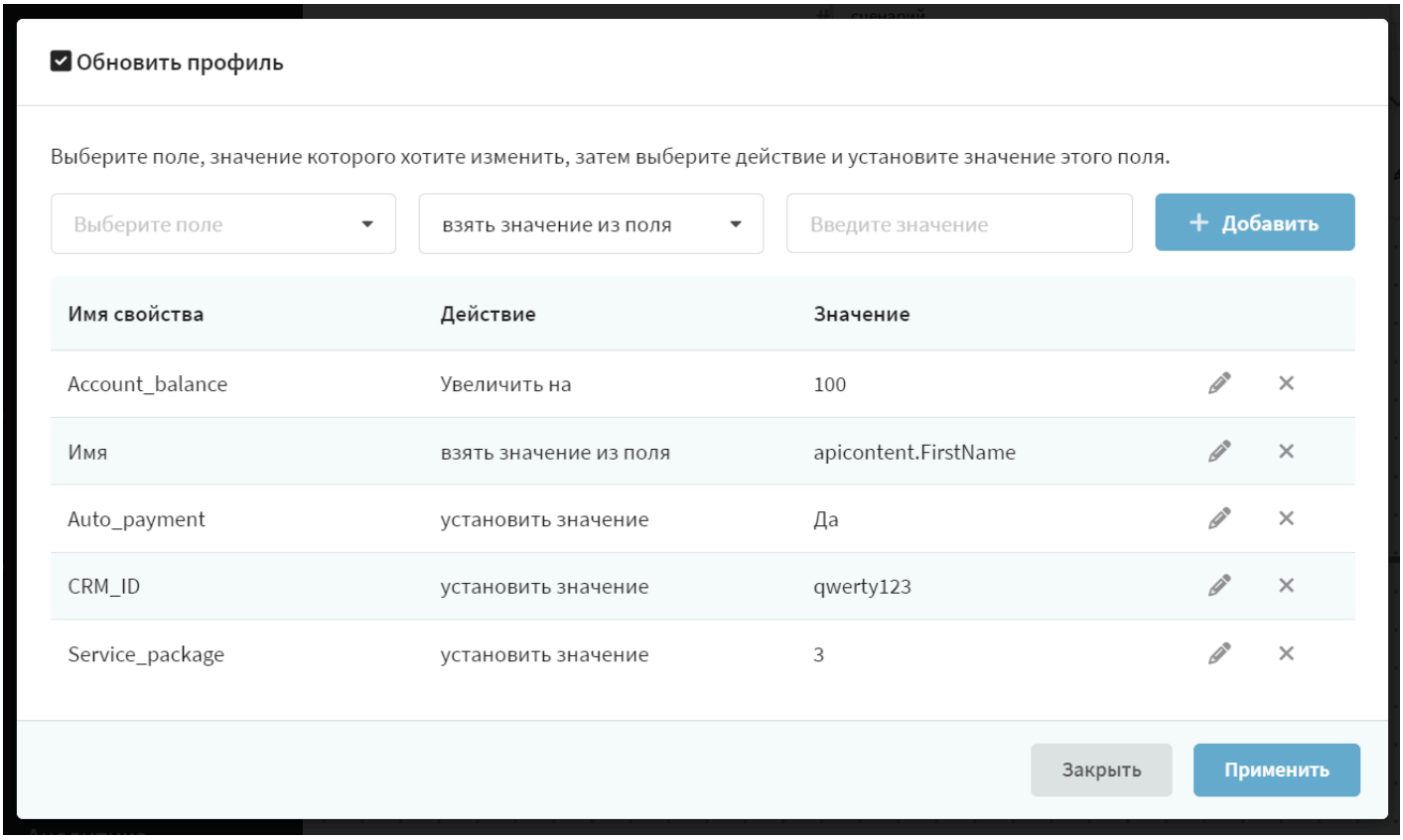
Обновить профиль

	Обновляет данные выбранных полей профиля подписчика или удаляет значения в этих полях. Для разных типов полей доступны связанные с ними действия. Узел может добавлять теги, изменять значения полей с выбором опций и числовых полей.

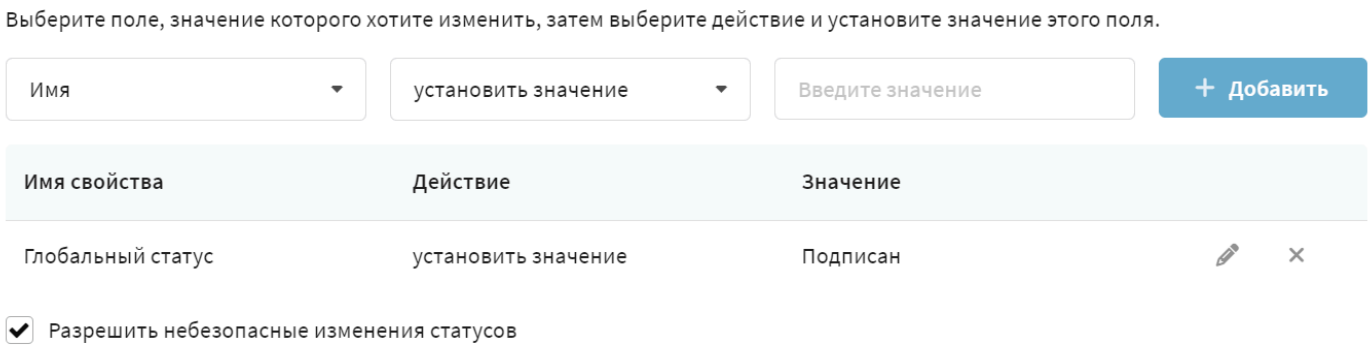
Подробнее

В выпадающем списке выберите поле, которое хотите обновить. Затем выберите действие и установите значение этого поля. Нажмите **“+ Добавить”**. Добавленное поле можно отредактировать, нажав на иконку карандаша, а также удалить из узла.

Действие	Тип поля	Описание
Установить значение	Все	Вручную указывается новое значение поля. Для полей типа Список (Enum) значение меняется на одно из заданных в списке, например, выбранный пакет услуг. Для полей типа Boolean значение меняется на Да или Нет.
Удалить значение	Все	Удаляет значение из выбранного поля.
Взять значение из поля	Все	Значение берется из поля `apicontent`. Используется вместе с узлом “API Запрос”, который способен сохранить результат запроса в объект `apicontent`.
Увеличить на	Целое число	Увеличивает или уменьшает значение. Для уменьшения значения нужно ввести отрицательное число.
Добавить элемент	Теги	Записать в поле еще один тег.
Удалить элемент	Теги	Удалить из поля один из существующих тегов.



С помощью этого узла вы также можете **изменить глобальный статус профиля**. Если вы разрешите небезопасные изменения статусов, то узел будет обновлять профили со статусами, отличными от “Подписан”.



INFO

При обращении к порядковому номеру элемента массива из `apicontent` необходимо использовать знак **%**. Счет в таких массивах начинается с нуля. Например, мы имеем следующий объект с массивом:

```
{
    "results": [
        {
            "objectId": "12asd34",
            "name": "Организация 1"
        }
    ]
}
```

Для того чтобы получить название первой организации, нам нужно использовать следующий запрос — `res.results.%0.name`, в котором `res` является полем для хранения ответа из узла “Вызов API”. Также обратите внимание, что запросы **не подерживают JSONPath**.

Пример **корректной** структуры запроса: `resp.data.data.%0.profile._region` Пример **некорректной** структуры запроса: `{jsonpath(resp.data "$.data[0].profile._region" "item")}`

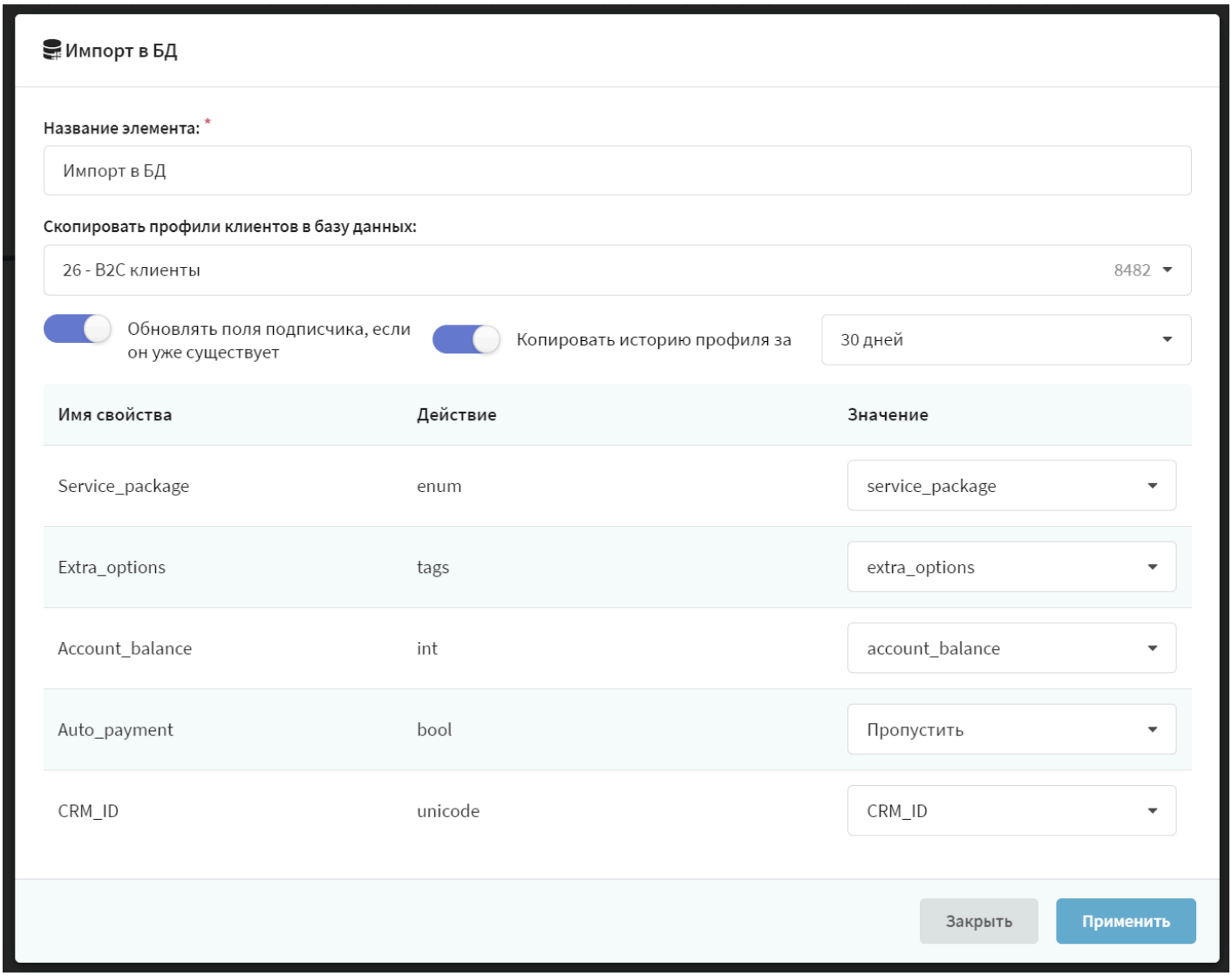
Импорт в базу данных

	Копирует профиль подписчика из базы профилей, для которой создан сценарий, в другую. Также можно копировать историю профиля за определенный период и перенести пользовательские поля профилей.

Подробнее

Введите **название** узла сценария и выберите **базу профилей**, в которую необходимо скопировать профиль клиента. Если вы хотите обновить данные профиля, который уже есть в целевой базе — активируйте опцию “**Обновлять поля подписчика, если он уже существует**”. Если в исходной базе есть **дополнительные поля**, их данные можно записать в поля подходящего типа целевой базы. Для этого настройте соответствие полей.

Вы также можете перенести историю профилю из исходной базы в целевую. Для этого воспользуйтесь опцией “**Копировать историю профиля**” и выберите период, данные за который хотите перенести. История профиля включает действия в рассылках (отправки, открытия, клики и др.), события подписок, импорта, пикселей, назначение промокодов и участие в статических сегментах.



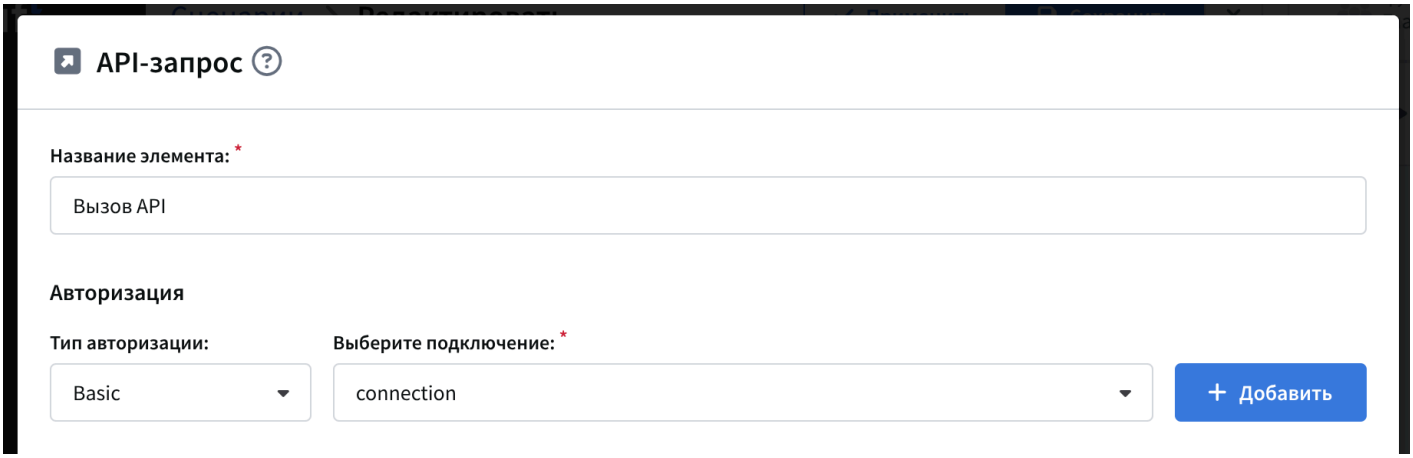
Вызов API

	Послать GET или POST запрос в другую систему. В запрос можно добавлять параметры со статическими данными или взятыми из подписчика. Используется в целях интеграции с другими системами.

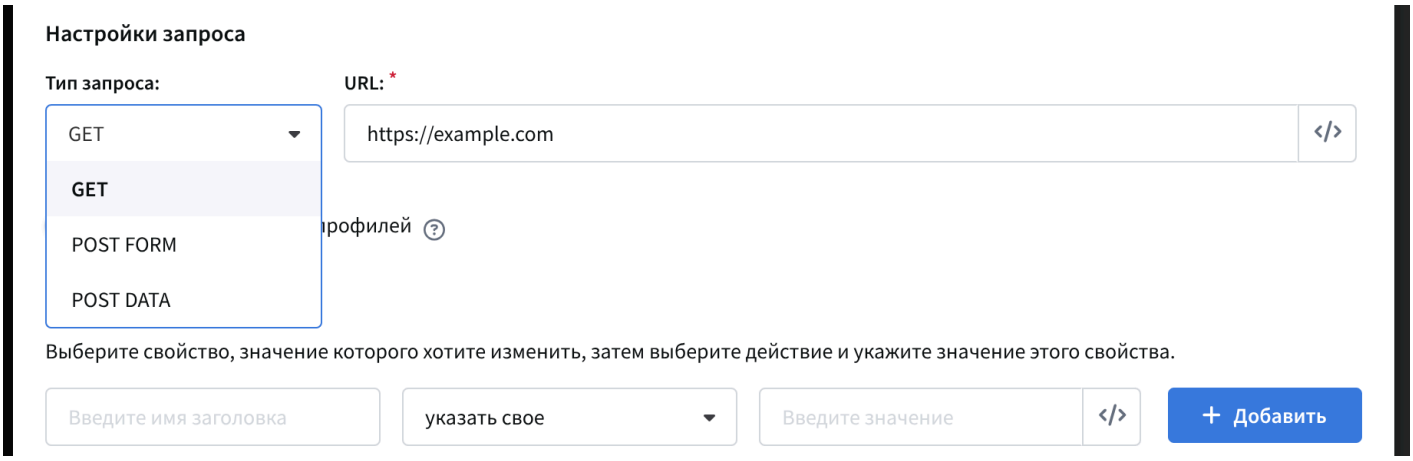
Подробнее

Введите **название** узла.

Выберите необходимый **тип авторизации**. Это необходимо, для того чтобы передавать данные профилей во внешние сервисы, которые используют защищенную авторизацию. Вы можете предварительно связать внешний сервис с платформой Altcraft через подключение, либо перейти к его созданию прямо из этого меню:



Выберите **тип запроса** к внешней системе. В поле **URL** введите адрес сервиса, принимающего данные:



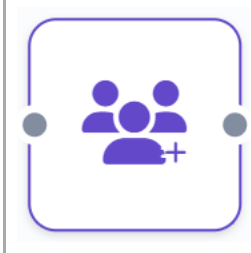
В URL запроса также используются **переменные**. Нажмите кнопку `</>` или введите значение вручную. Пример использования **промокода** в адресе запроса:

```
http://www.example.org/api/?Promocode={loyalty.black_friday.promocode}
```

Пользователь может ограничить количество профилей, проходящих через этот узел за определенный промежуток времени. Это необходимо для обеспечения стабильной работы сценария без перегрузок. Для этого активируйте соответствующий переключатель, укажите число профилей, а также временной промежуток (минута, секунда или час):

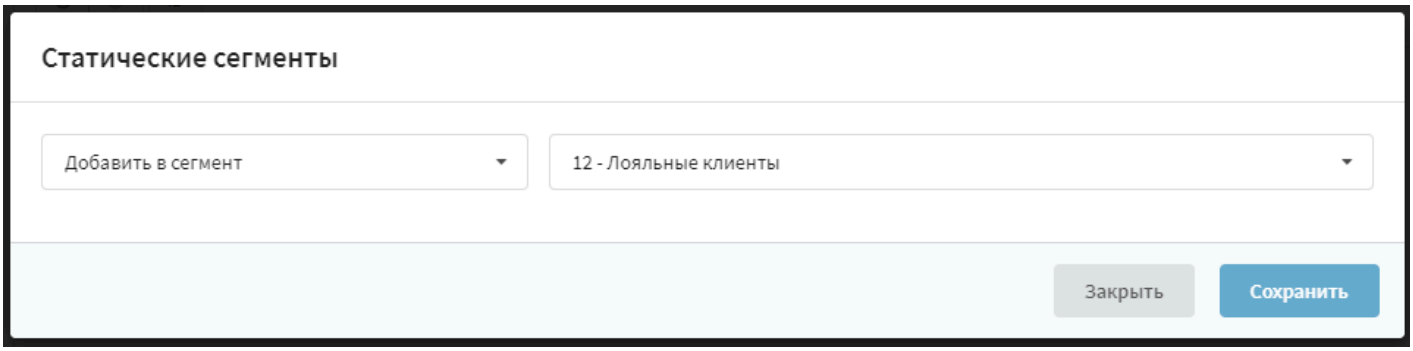
▶ **Принцип работы ограничителя**

Обновление статистического сегмента

	Добавляет профиль клиента в статический сегмент аудитории или удаляет из сегмента.

Подробнее

В настройках узла выберите добавление или удаление и доступный статический сегмент:



Подписки

	Подписывает клиента на определённый ресурс или обновляет данные существующей подписки. В качестве источника подписок могут использоваться данные профиля или данные подписок на другой ресурс.

Подробнее

В системе используются **два типа каналов подписки:**

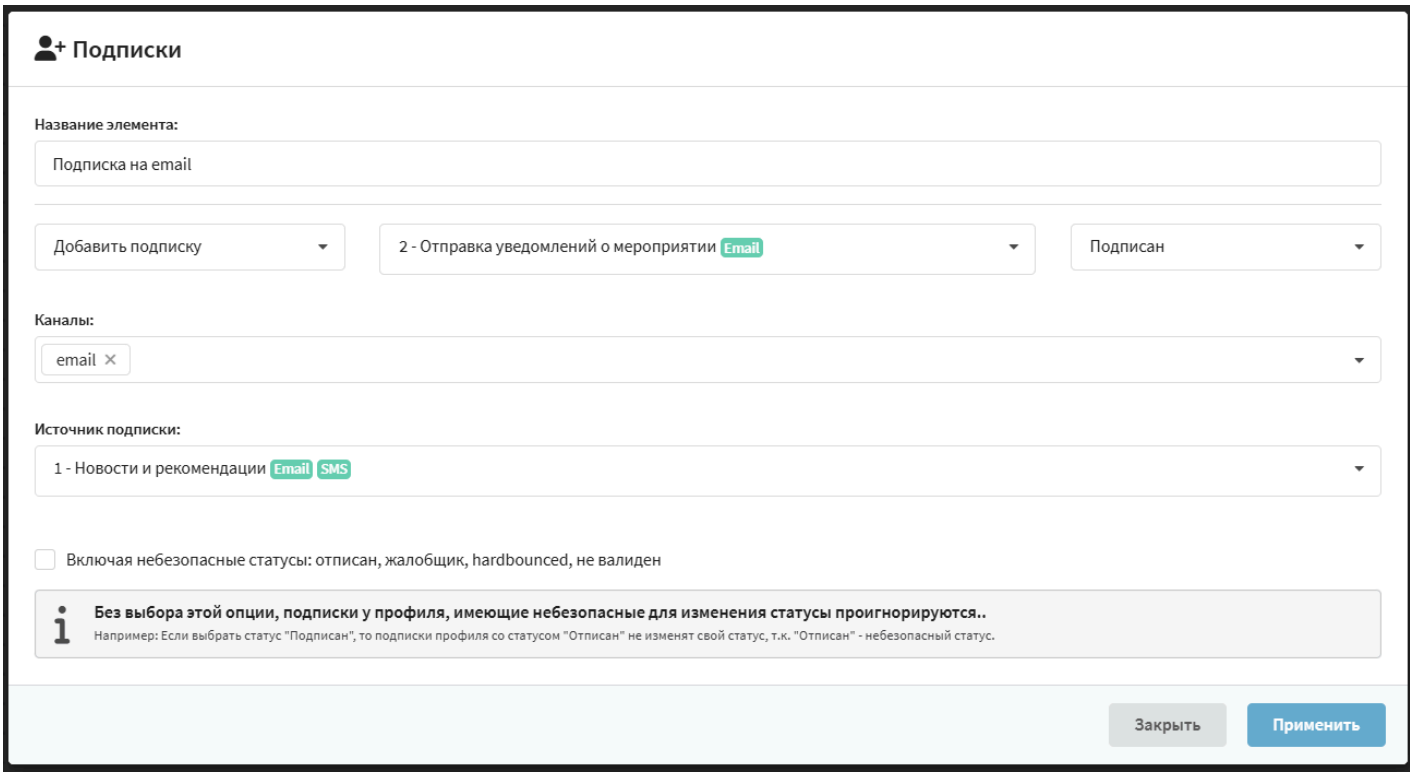
- **Indirect-канал** — отправляет сообщение неопределенному количеству пользователей. Используется только для рассылки в кастомные каналы и в Telegram Group-канал.
- **Direct-канал** — отправляет персонализированное сообщение конкретному получателю.

Добавление и обновление подписки

- Подписку можно **добавить**, если клиент ещё не подписан на ресурс.
- Можно **изменить статус** подписки. Например, при подтверждении подписки или при использовании в кастомных сценариях отписки.

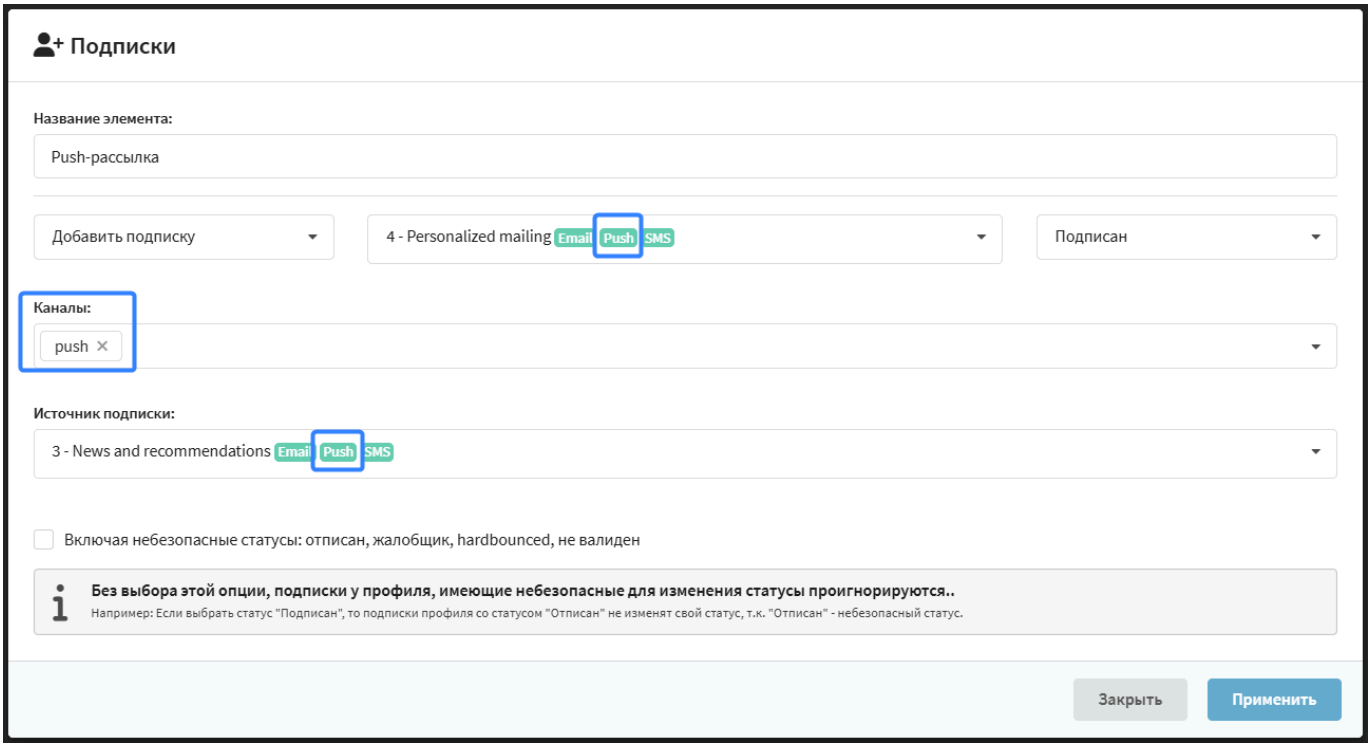
Как **добавить** или **обновить подписку** на ресурс:

1. Введите **название узла**.
2. Нажмите **Добавить подписку** или **Обновить**.
3. Укажите **ресурс** из выпадающего списка.
4. При необходимости измените **статус подписки**.
5. Выберите доступные **каналы связи**.
6. В качестве источника подписки используйте данные профиля (например, **Email**, **Телефон**) или данные подписок другого ресурса.



TIP

Подписаться на **Push-уведомления** можно только если в качестве **источника подписки** выбран ресурс, для которого уже настроена отправка **Push**.



Промокод

	Назначает профилю клиента промокод от выбранной программы лояльности. Промокод может быть затем отправлен клиенту с помощью одного из узлов в категории **"Каналы"** или использован для расчёта персональных предложений.

Подробнее

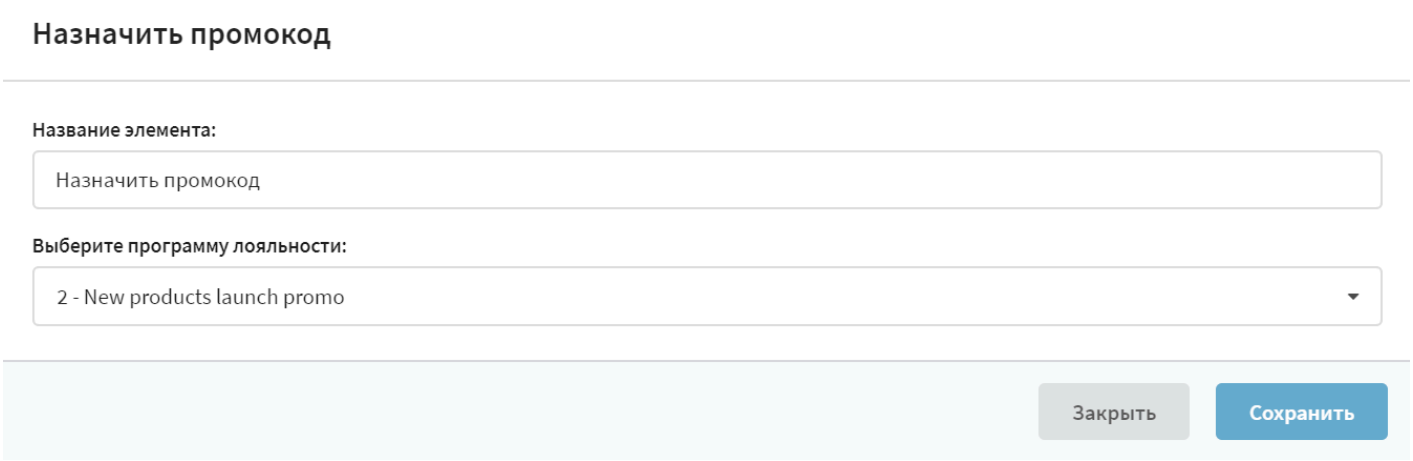
Введите **название** узла сценария и выберите необходимую **программу лояльности**. В программе должно быть сгенерировано достаточное количество промокодов или должна быть активна опция создания промокодов сценарием.

CAUTION

Если профиль уже имеет промокод этой программы лояльности, сценарий продолжит выполняться.

Если промокод не может быть назначен по одной из причин ниже, сценарий закончится на этом узле:

- достигнут лимит промокодов
- программа лояльности не активна
- программа лояльности не имеет свободных промокодов, а её опция "Сценарий может создавать промокоды, если их не хватает" отключена



Регистрация цели

	Записывает в профиль клиента достижение определённой **цели** и связанную **стоимость** действия. Может использоваться для **скоринга** подписчиков, **LTV** и **RFM** сегментации, в отчётах по **конверсии** и **прибыли**.

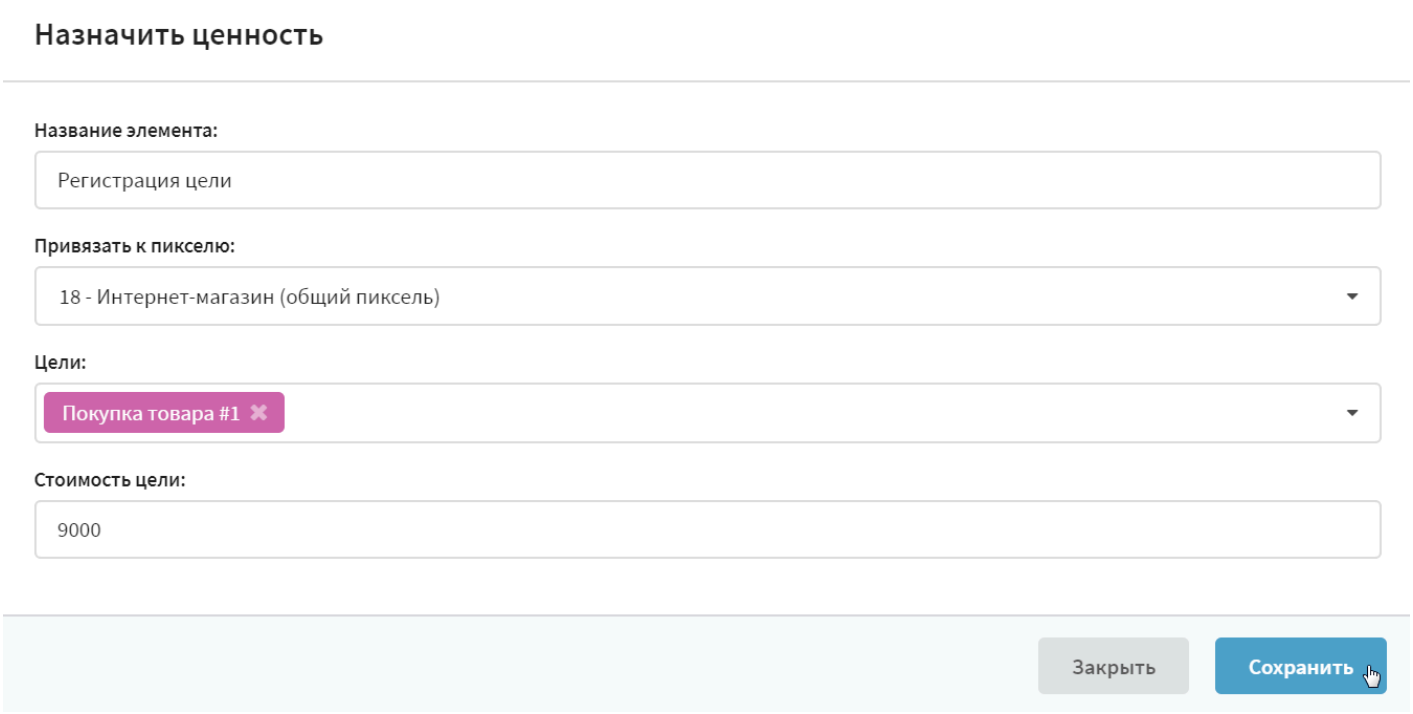
Подробнее

Введите **название** узла и пиксель, к которому будет привязана цель. Регистрация одинаковых целей на разные пиксели может быть использована в партнёрских программах или для A/B тестирования интерфейса вашего сайта или других ресурсов.

Ниже выберите существующие **цели** или добавьте новые. Укажите **стоимость**, которую необходимо передать для целей.

TIP

Значение для всех целей будет записано одно. Если необходимо передать несколько целей с разными значениями, добавьте несколько таких **узлов** - один за другим, с разными целями.



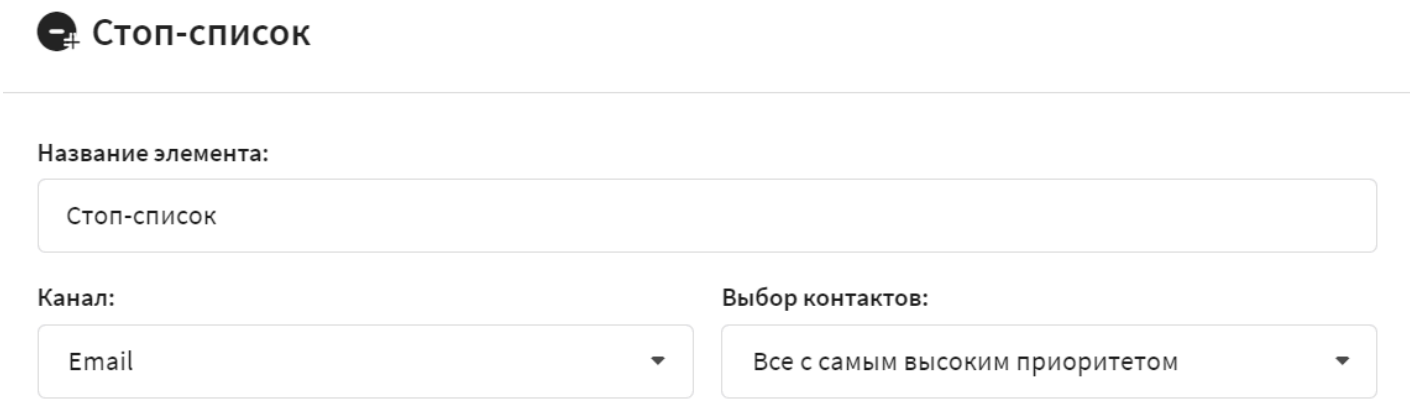
При необходимости, вы можете передавать внутри шаблонов переменные, связанные с данными о срабатывании пикселя. Узнать подробнее об этих переменных можно здесь.

Стоп-список

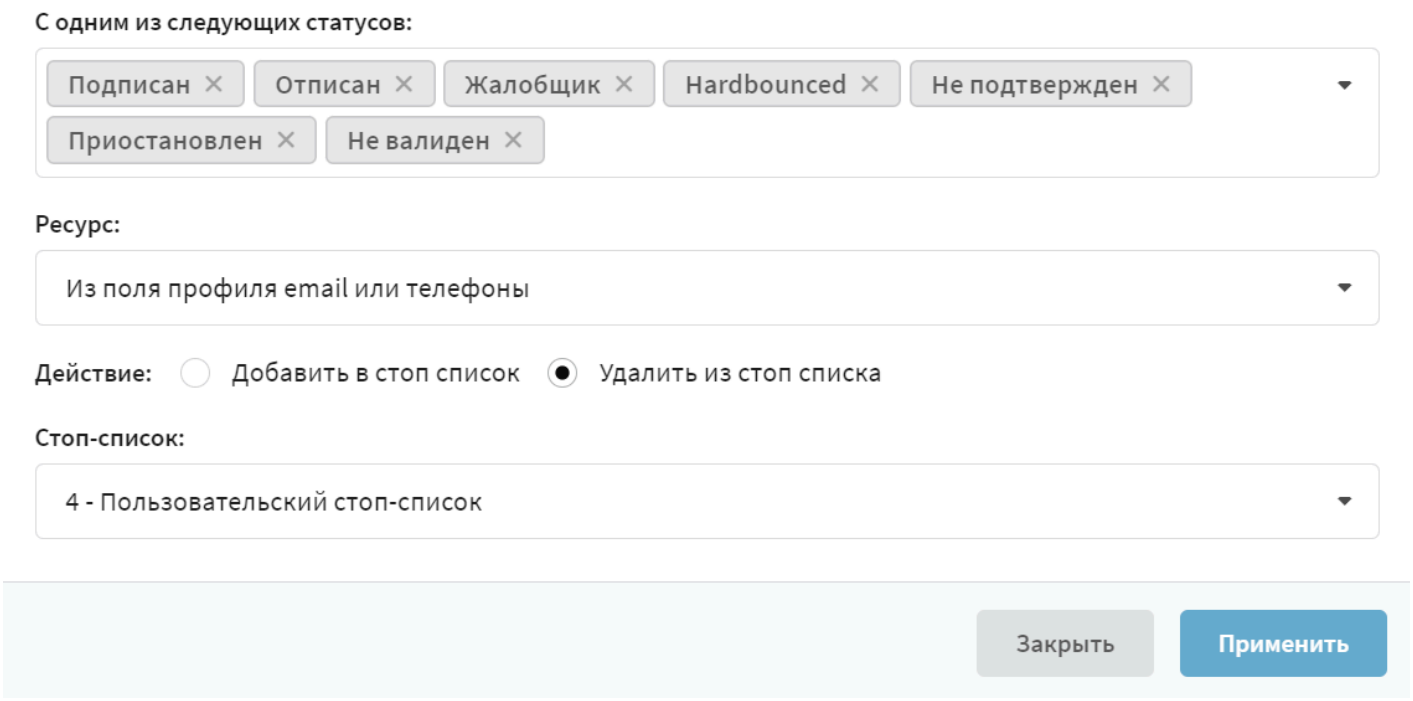
	Добавляет определённые контактные данные клиента **(email адрес и номер телефона)** в стоп-список платформы. Коммуникация с клиентом по этим данным будет заблокирована. Или наоборот, **удаляет** контактные данные из стоп-списка, восстанавливая возможность коммуникации с клиентом.

Подробнее

Введите название узла для отображения в сценарии. Далее необходимо выбрать, контактные данные каких подписок необходимо заблокировать или восстановить. Выберите **канал коммуникации** и **приоритет** подписок для изменения:



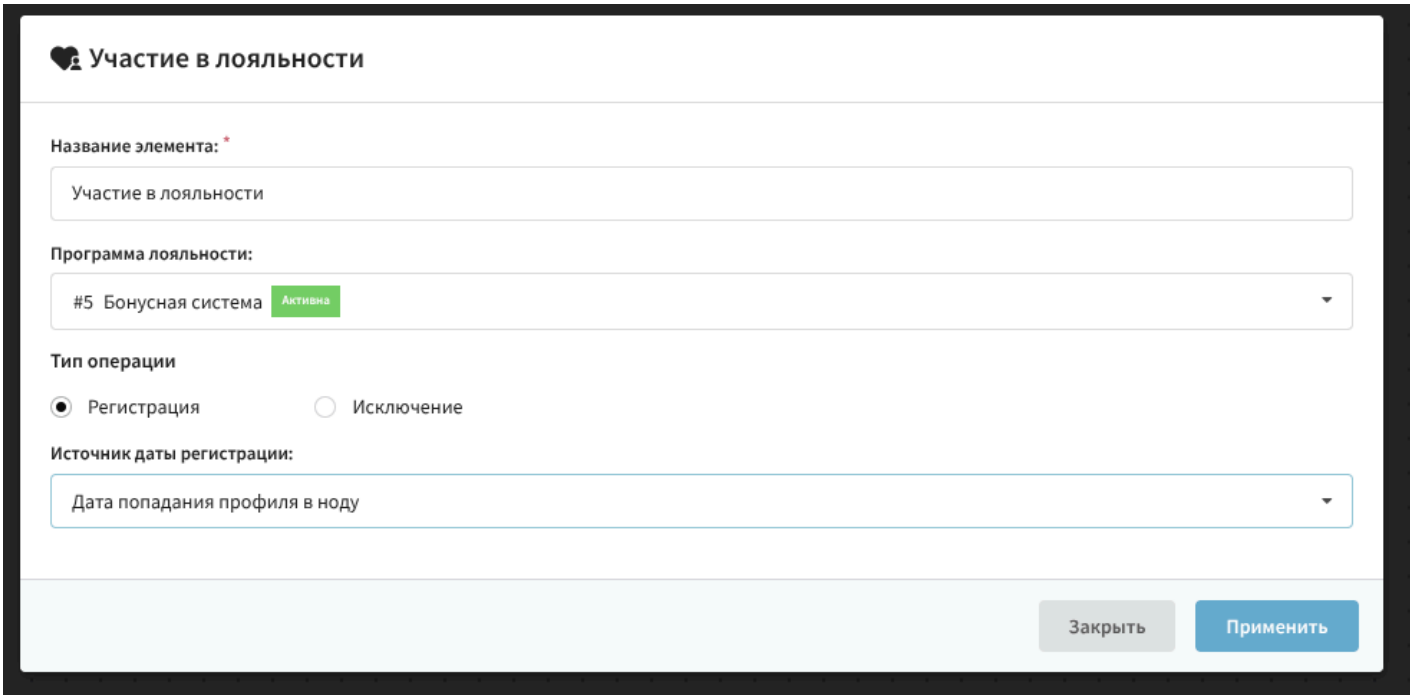
Выберите ресурс и статусы подписок, контактные данные которых необходимо **добавить в стоп-список** или **удалить из стоп-списка**. Внизу формы выберите стоп-список для добавления или удаления:



Участие в лояльности

	Регистрирует или исключает пользователя из [**программы лояльности**](/user-guide/loyalty-programs).

Подробнее



Для настройки узла выберите заранее созданную **программу лояльности** и операцию, которая будет произведена с профилем — **регистрация** или **исключение** из программы лояльности.

При регистрации профиля можно указать источник даты регистрации участника в программе лояльности:

- **Дата попадания профиля в узел** — время, в которое профиль оказался на этом узле сценария;
- **Дата из поля профиля** — время регистрации будет взято из выбранного поля профиля в формате даты;

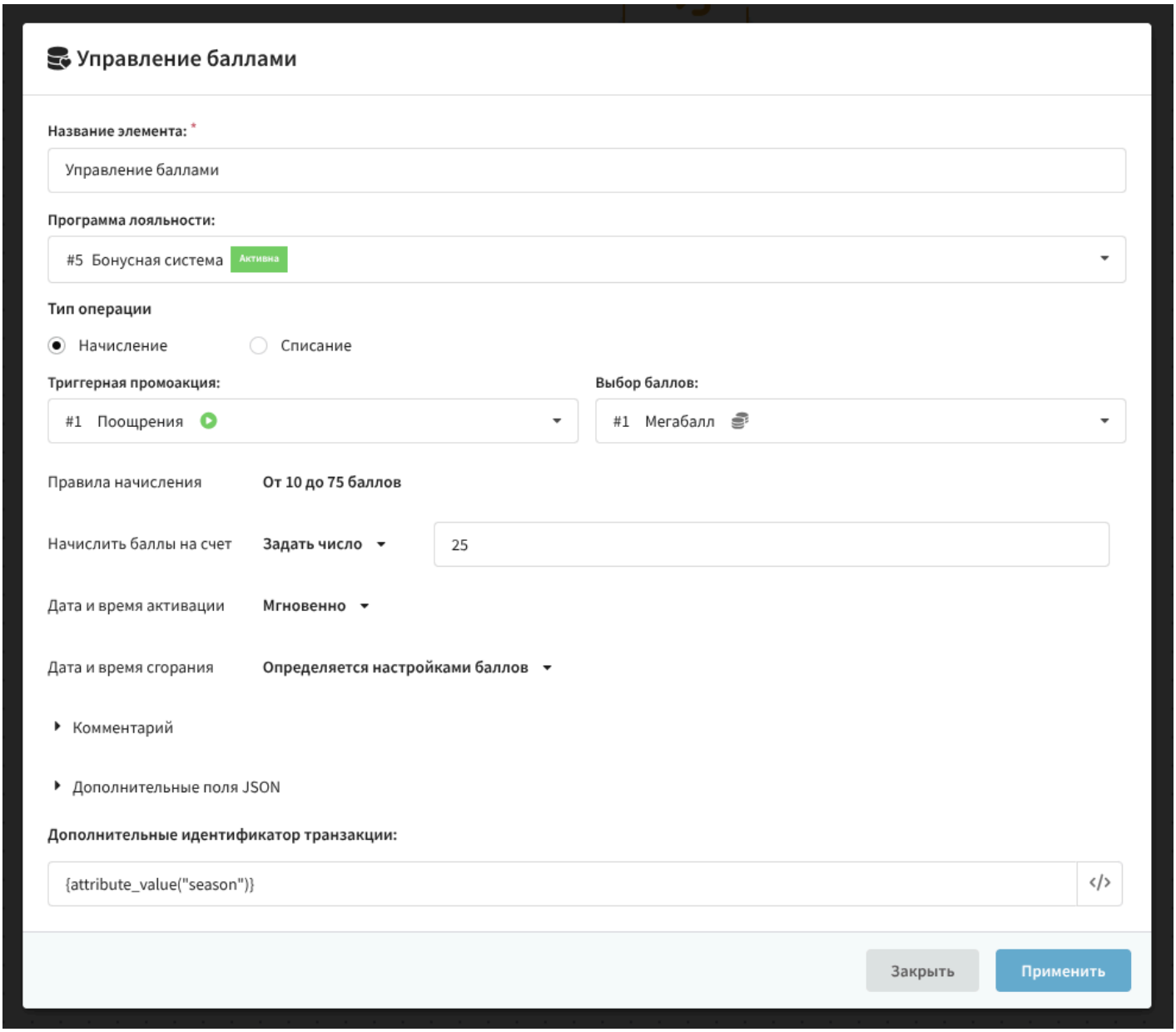
- **API-контент** — время регистрации будет взято из указанного API-поля.

Для последних двух вариантов доступны два варианта обработки некорретных данных: использование даты попадания профиля в узел или регистрация ошибки.

Управление баллами

	Изменяет количество баллов профиля в программе лояльности.

Подробнее



Для настройки узла выберите заранее созданную **программу лояльности** и тип операции — **начисление** или **списание** баллов лояльности.

Затем укажите промоакцию, в рамках которой будет проводиться начисление, и вид баллов. Количество баллов можно указать либо числом, либо взять из поля профиля, либо передать из API-content.

Помимо этого можно указать:

- **Дата и время активации** (мгновенно, по полю из профиля или по значению из API-content);

- **Дата и время сгорания** (в соответствии с настройками баллов, по полю из профиля или по значению из API-content);
- **Комментарий** к транзакции;
- **Дополнительные поля JSON**;
- **Дополнительный идентификатор транзакции**.

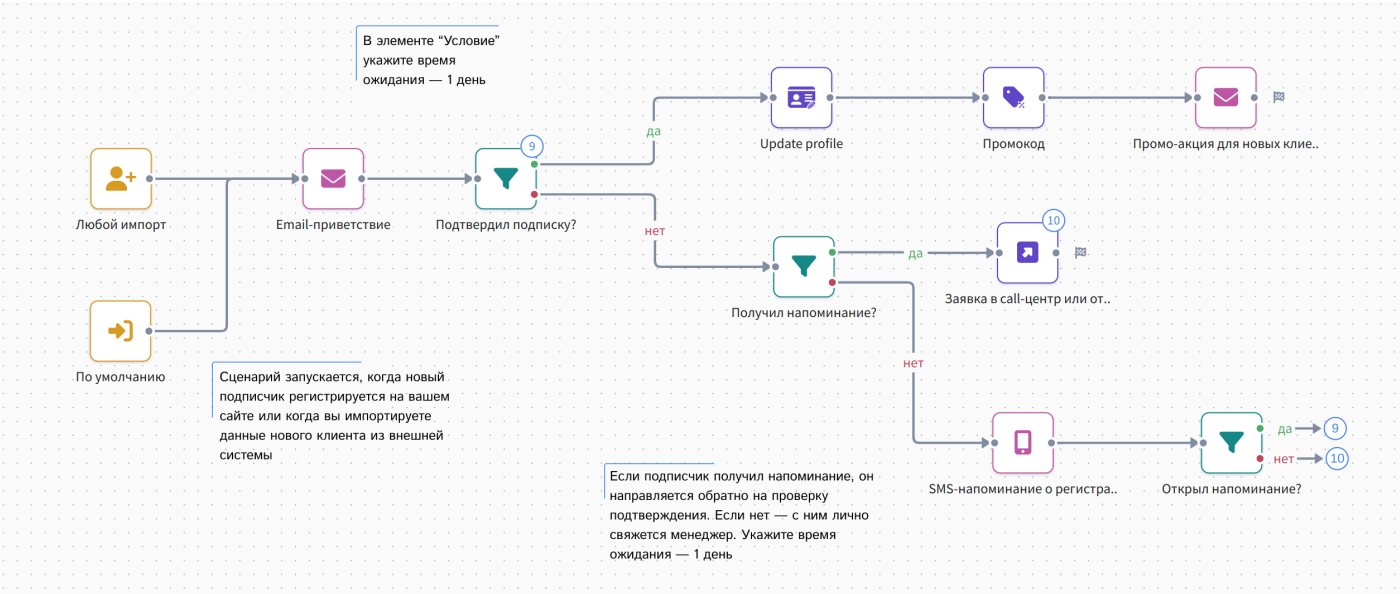
**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ. ** Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ*

Классические сценарии автоматизации маркетинга

Приветственная цепочка

Приветственные цепочки используются для повышения вовлечённости клиентов с момента регистрации. С помощью **сценария**, пример которого представлен ниже, вы можете:

- автоматизировать процесс подтверждения email подписки
- напоминать о регистрации по SMS
- назначать приветственные бонусы и промокоды
- отправлять запрос в колл-центр или отдел продаж по неактивным пользователям



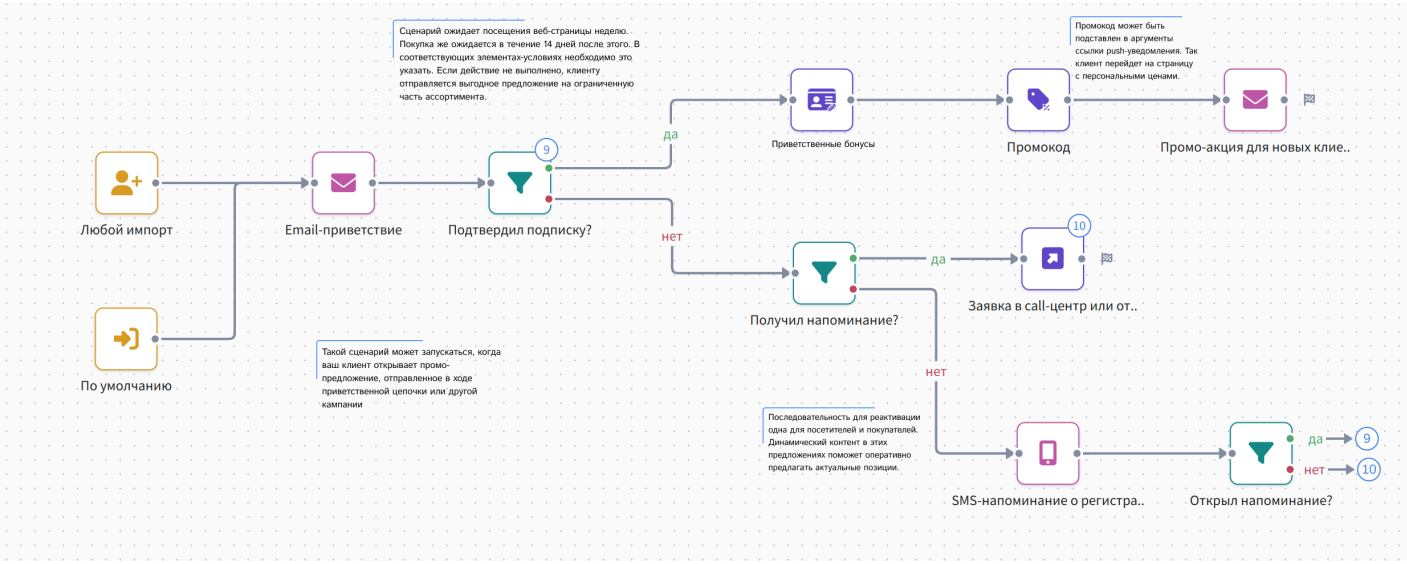
TIP

[Здесь вы найдёте](#) пошаговое руководство по созданию приветственного сценария.

Онбординг: первая покупка

Онбординговые сценарии помогут вам вовлечь клиентов с момента получения первой рассылки. Например, после массовой рассылки о выходе новой продуктовой линейки, вы можете стимулировать активность подписчиков следующей цепочкой действий:

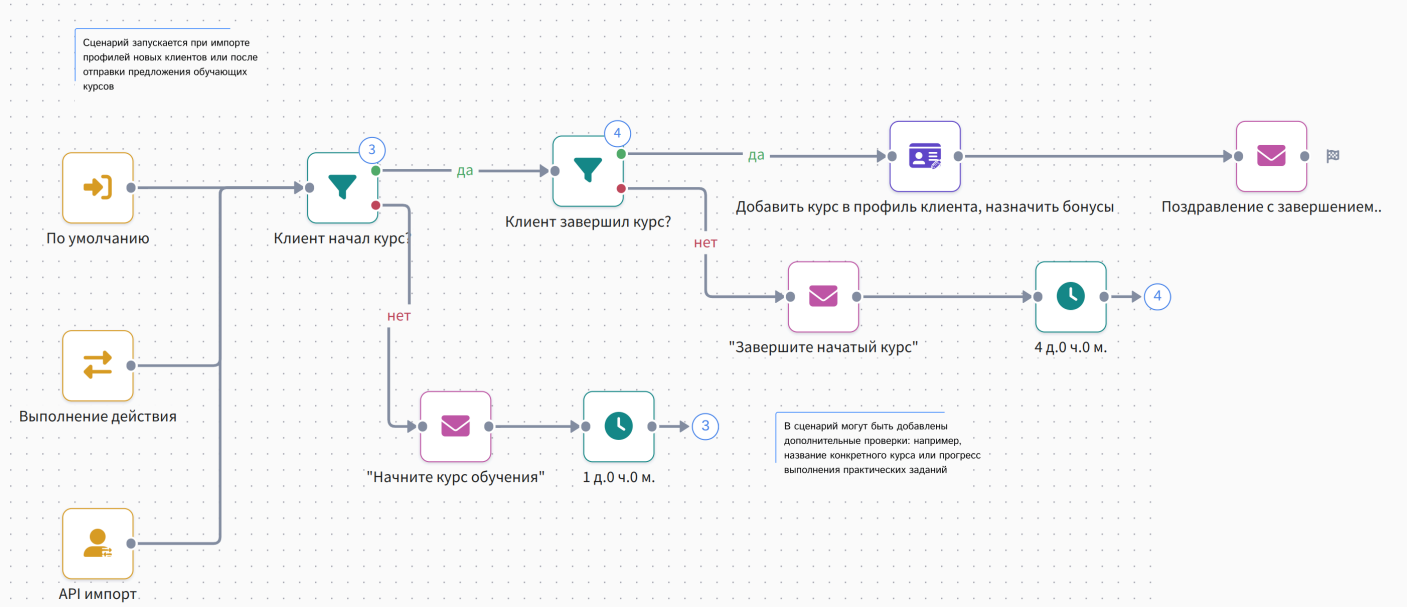
- запуск сценария после выполнения целевого действия
- реактивация клиентов, не проявивших интереса к рассылке
- реактивация клиентов, открывших сайт магазина, но не сделавших покупку
- назначение промокодов на ограниченную часть ассортимента, например - особые промо-товары



Онбординг: обучающие курсы

Один из лучших способов собрать лояльную аудиторию клиентов - научить их самостоятельно разбираться в услугах, которые вы им предоставляете. Наиболее важно это для финансового сектора, так как в разы повышает доверие клиентов к вашей компании: вы становитесь для клиентов не только продавцом услуг, но и авторитетом в своей отрасли.

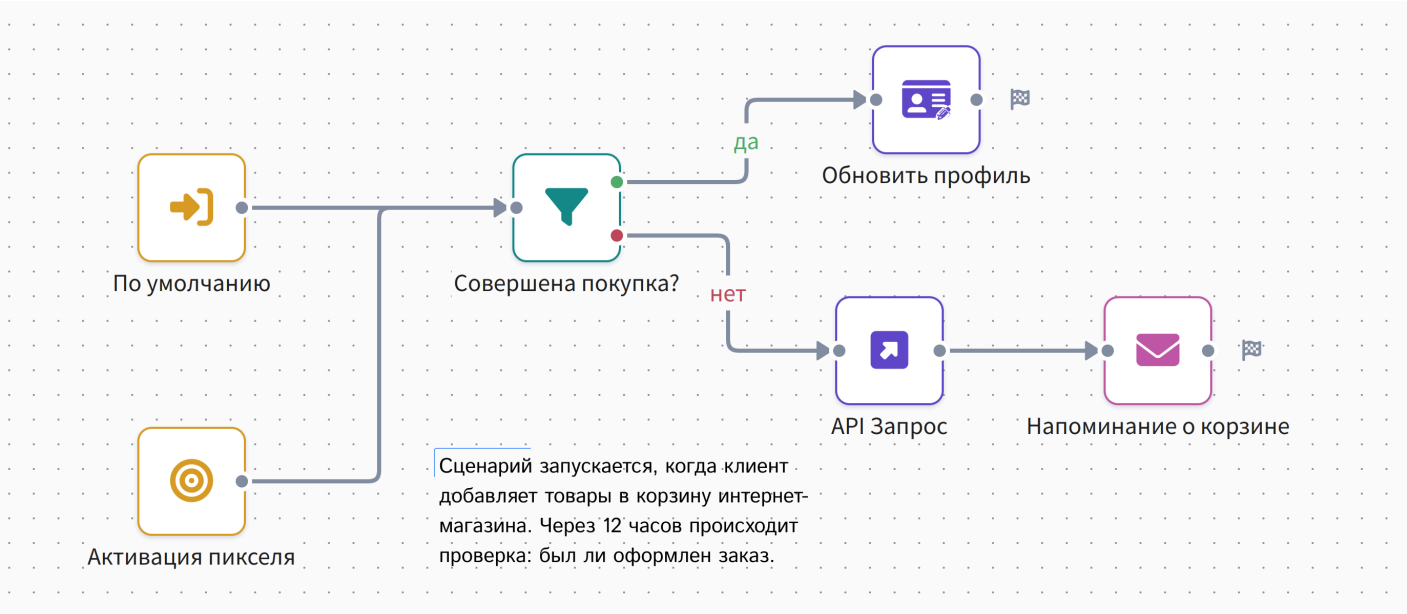
С помощью сценариев Altcraft Marketing вы можете сделать процесс обучения интерактивным, своевременно назначая клиентам релевантные курсы обучения - и проверяя их выполнение:



Брошенная корзина

Если потенциальный клиент заинтересовался вашим предложением настолько, что добавил товары в “корзину” - можно увеличить шансы на оформление заказа, напомнив о нём через некоторое время. Если на часть товаров можно предоставить скидку ради удержания клиента - сценарий может автоматически назначить промокод и отправить персональное предложение.

Для запуска сценария используется **вызов API платформы**.

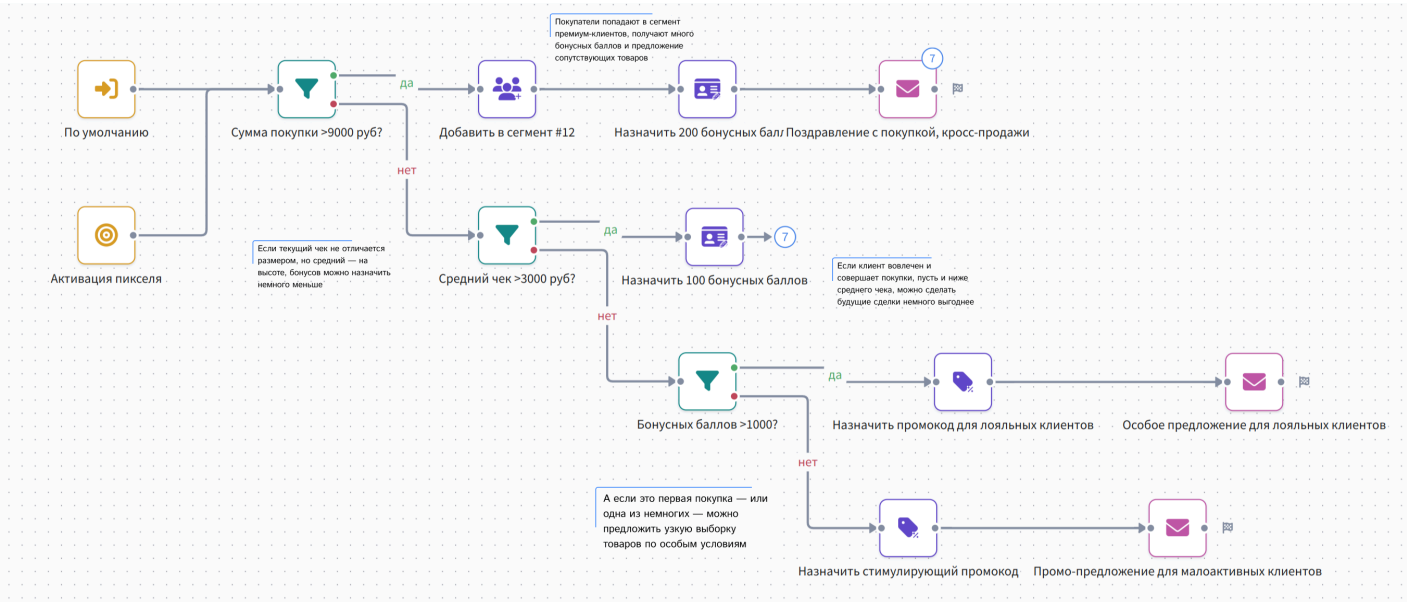


TIP

Пример реализации сценария брошенной корзины можно найти в **отдельной статье**.

Управление лояльностью клиентов

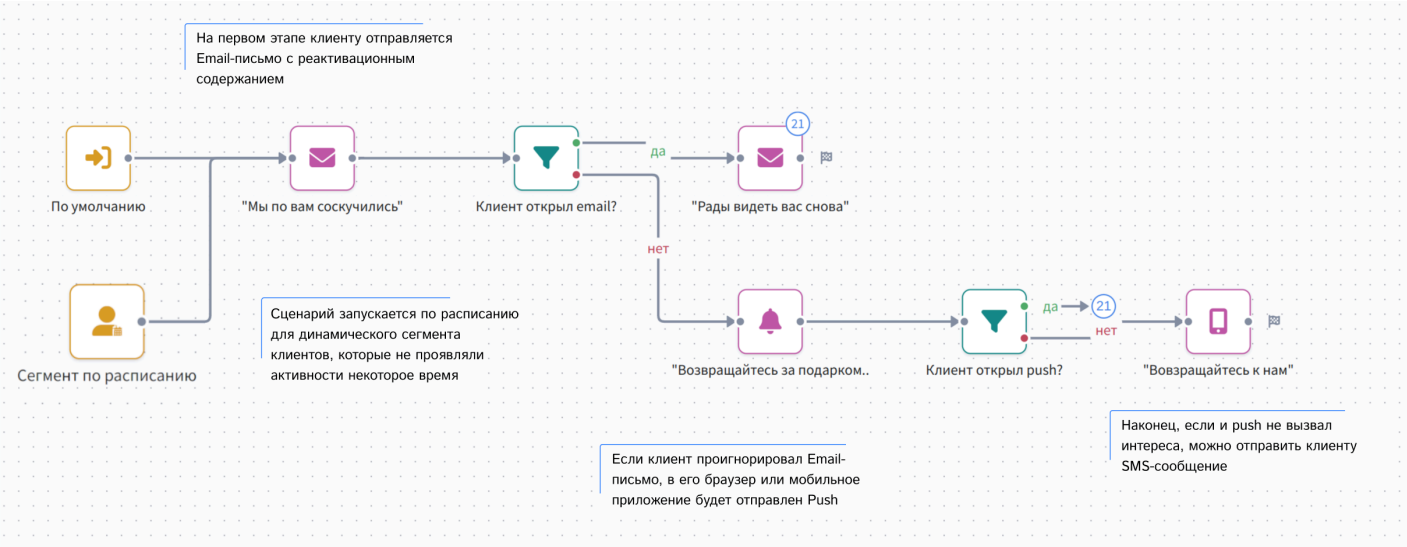
Лояльность клиентов - один из решающих факторов современного рынка. Геймифицируйте процесс коммуникации с клиентом: настройте сценарий, который при каждой покупке будет проверять лояльность по данным заказа, **RFM показателям** (средний чек), или уже накопленным бонусным баллам. Такой сценарий повысит активность ваших покупателей:



Омниканальная реактивация клиентов

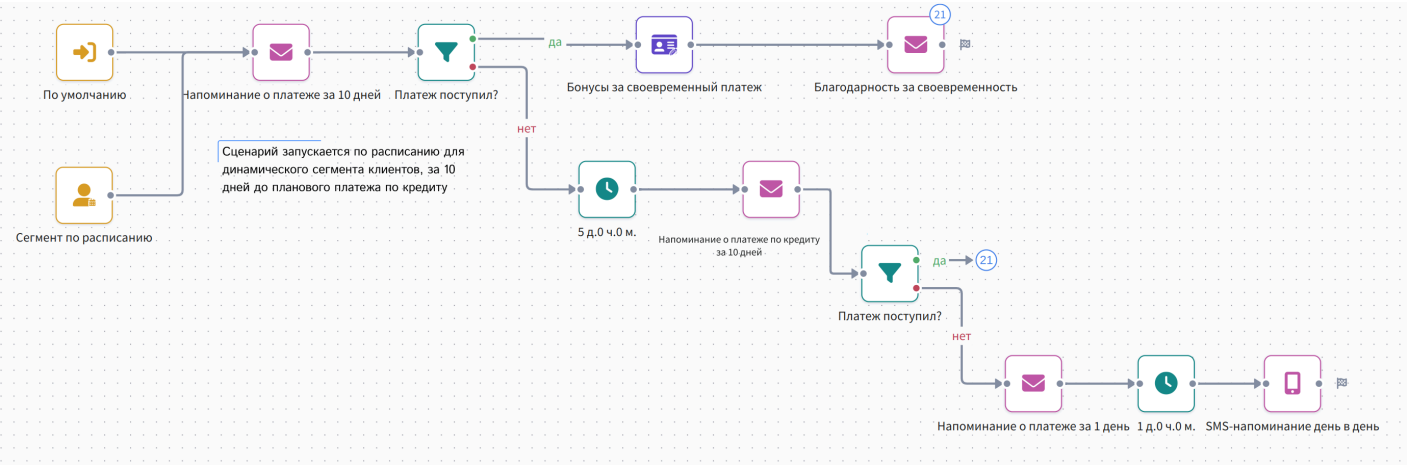
Для малоактивных клиентов вы можете настроить сценарий автоматической реактивации. Для этого используется запуск сценария для сегмента по расписанию. **Динамический сегмент** каждый запуск будет выбирать неактивных клиентов и отправлять их профили в сценарий.

Последовательность действий представляет собой каскадную омниканальную рассылку. Вы сможете использовать её не только для реактивации - но и как замену **регулярной рассылки**, если вам нужно донести сообщение по любому из доступных каналов связи:



Напоминание о платеже

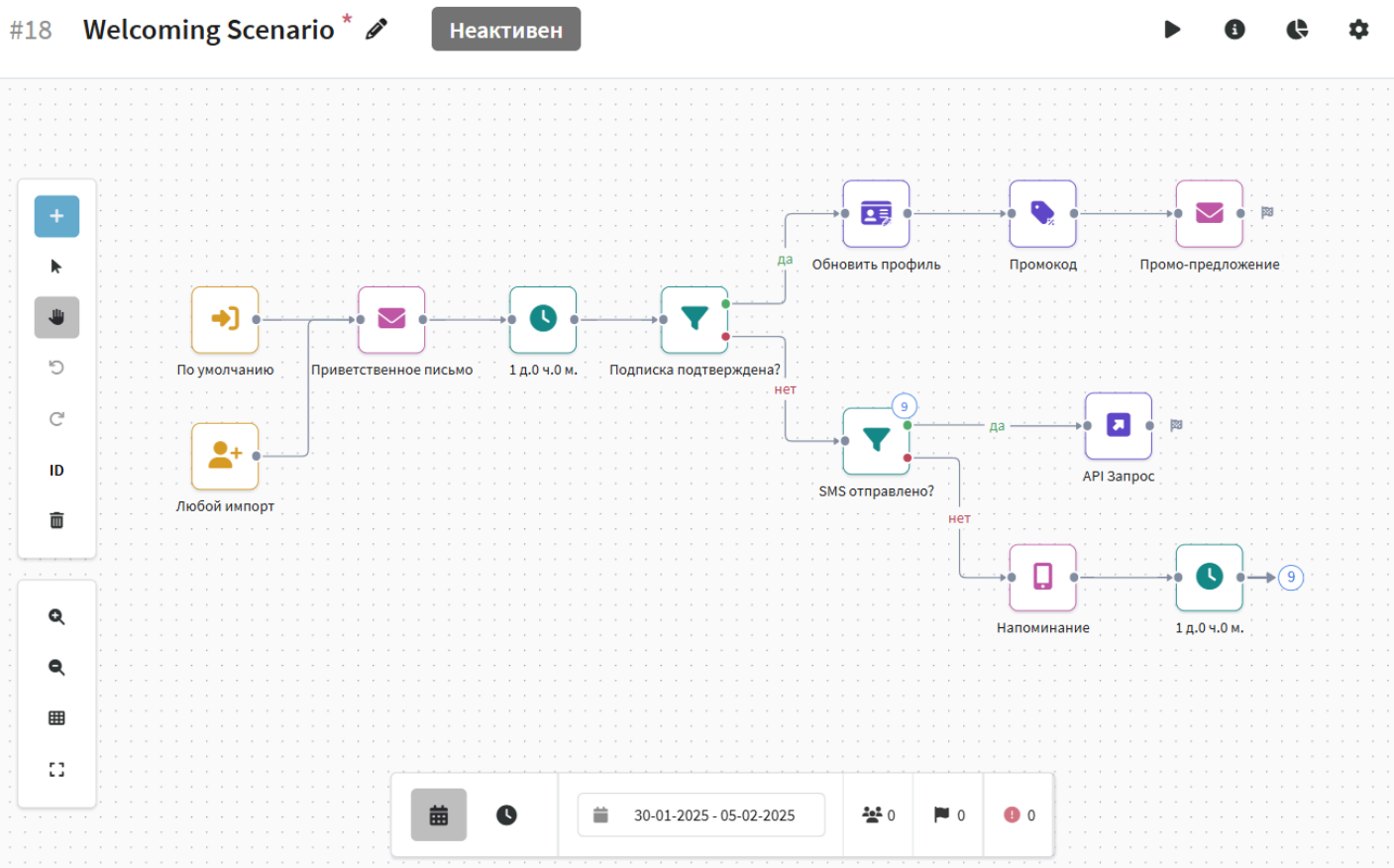
С помощью сценариев Altcraft Marketing можно автоматизировать напоминания о продлении услуг или внесении обязательных платежей по рассрочке или кредиту. **Динамический сегмент** будет выбирать только актуальные данные на момент запуска цепочки. После этого сценарий отправляет серию напоминаний, проверяя после каждого - поступили ли средства:



Приветственный сценарий: пошаговая настройка

Этот гайд поможет вам настроить сценарий автоматизации, который будет взаимодействовать с клиентом от момента регистрации на вашем сайте до совершения первых покупок. Описаны принципы, по которым подбирается тот или иной шаг Сценария – в зависимости от ваших средств и целей.

Приветственная цепочка – один из классических сценариев автоматизации, который можно реализовать ещё эффективнее с Altcraft Platform.

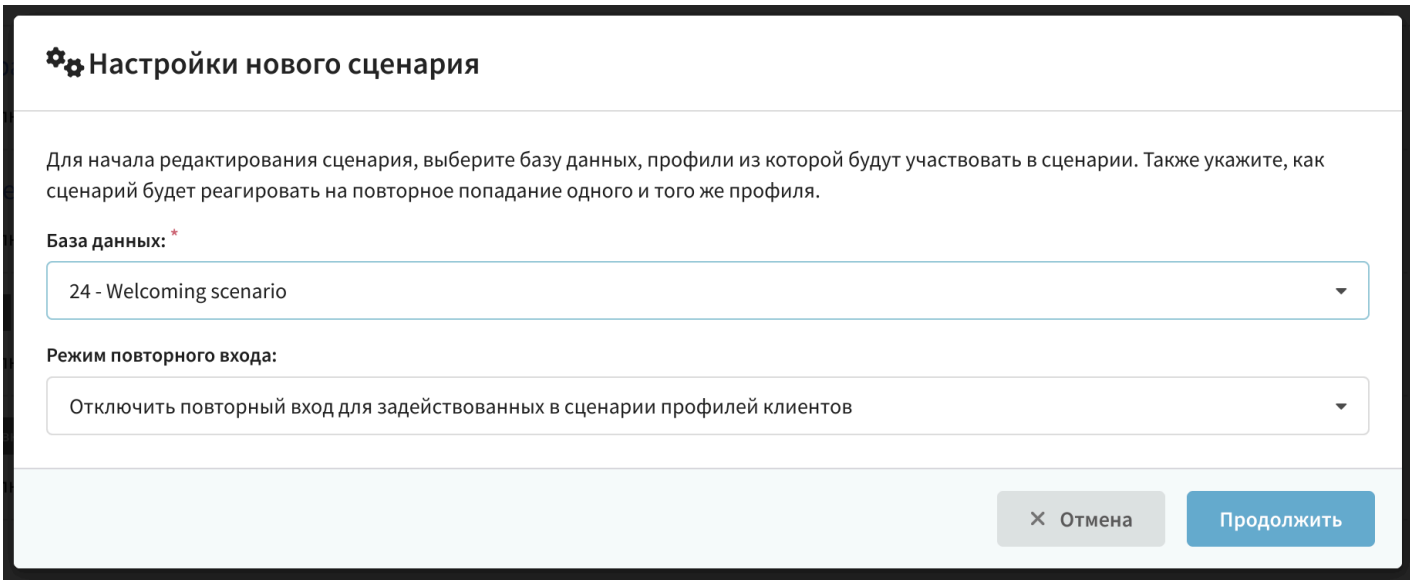


TIP

Работа со сценариями автоматизации описана в этой статье.
Подробное описание узлов сценария вы найдёте здесь.

Шаг 0: Создайте новый сценарий

В разделе “Сценарии” создайте новый сценарий, нажав на соответствующую кнопку в верхней части экрана. Укажите **базу профилей** и **отключите** повторный вход для задействованных в сценарии профилей, поскольку приветствие – событие разовое:



Шаг 1: Настройте захват профилей клиентов

Настроим сценарий так, чтобы профили клиентов попадали в него в момент регистрации в платформе. Условие захвата (**событие**) будет зависеть от того, как вы импортируете данные клиентов:

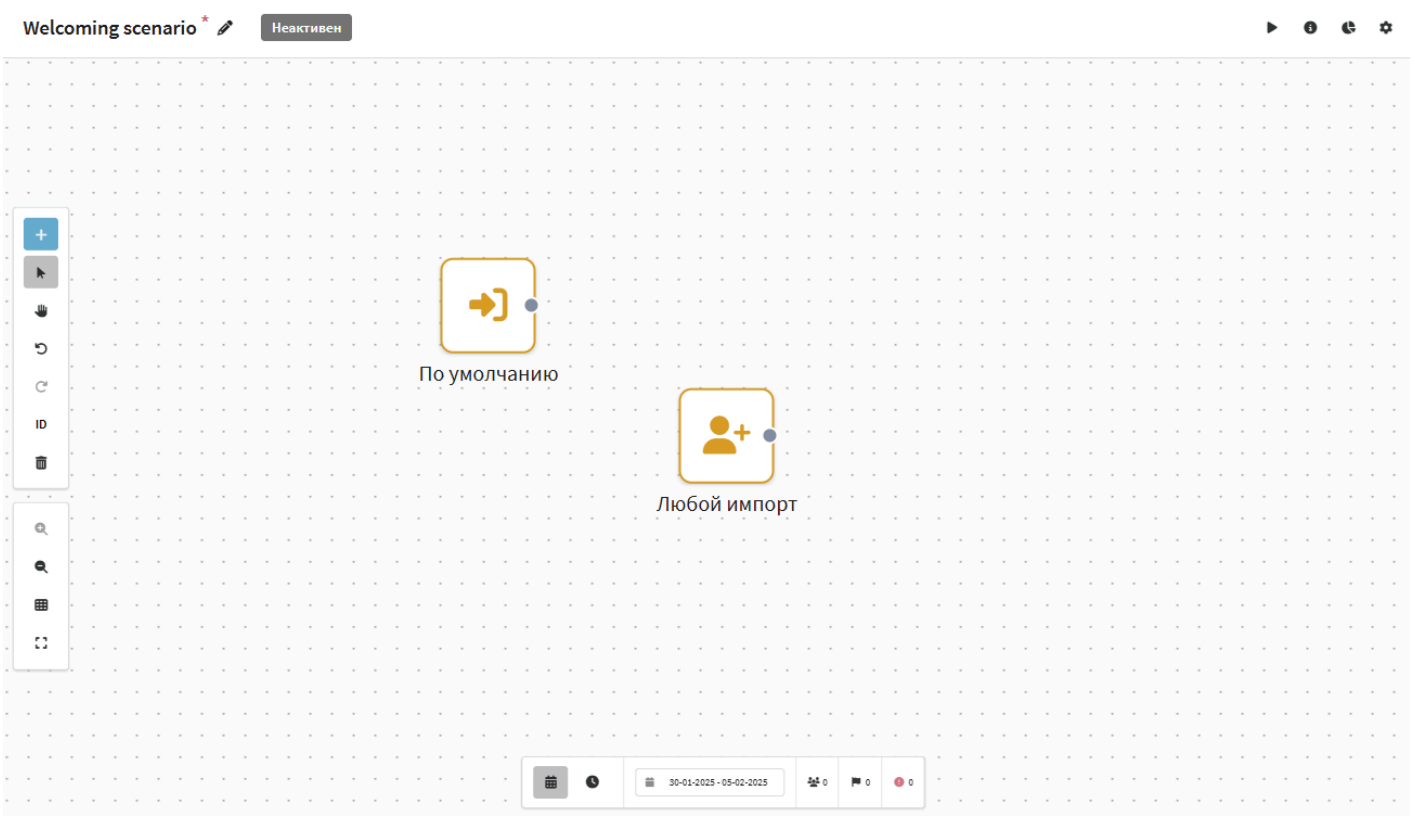
- **При любом импорте** – срабатывет при добавлении любого нового профиля.
- **При импорте вручную** – при создании нового профиля вручную.
- **При импорте через API** – при использовании вызовов API импорта. Подойдёт для регистрации профилей на сайте и передачи данных из интеграционного ПО (ETL).
- **При импорте через форму** – при заполнении полей формы. О том, как создать форму в платформе, читайте [здесь](/user-guide/weblayers/forms/).
- **При импорте через попап** – при заполнении формы в попапе. Подробнее о создании и настройке попапов можно узнать [в этой статье](/user-guide/weblayers/popups/).

Также в любом сценарии существует элемент **захвата профилей по умолчанию**. Он необходим для того, чтобы захватывать профили, добавленные вручную, через API и попавшие в цепочку из другого сценария.

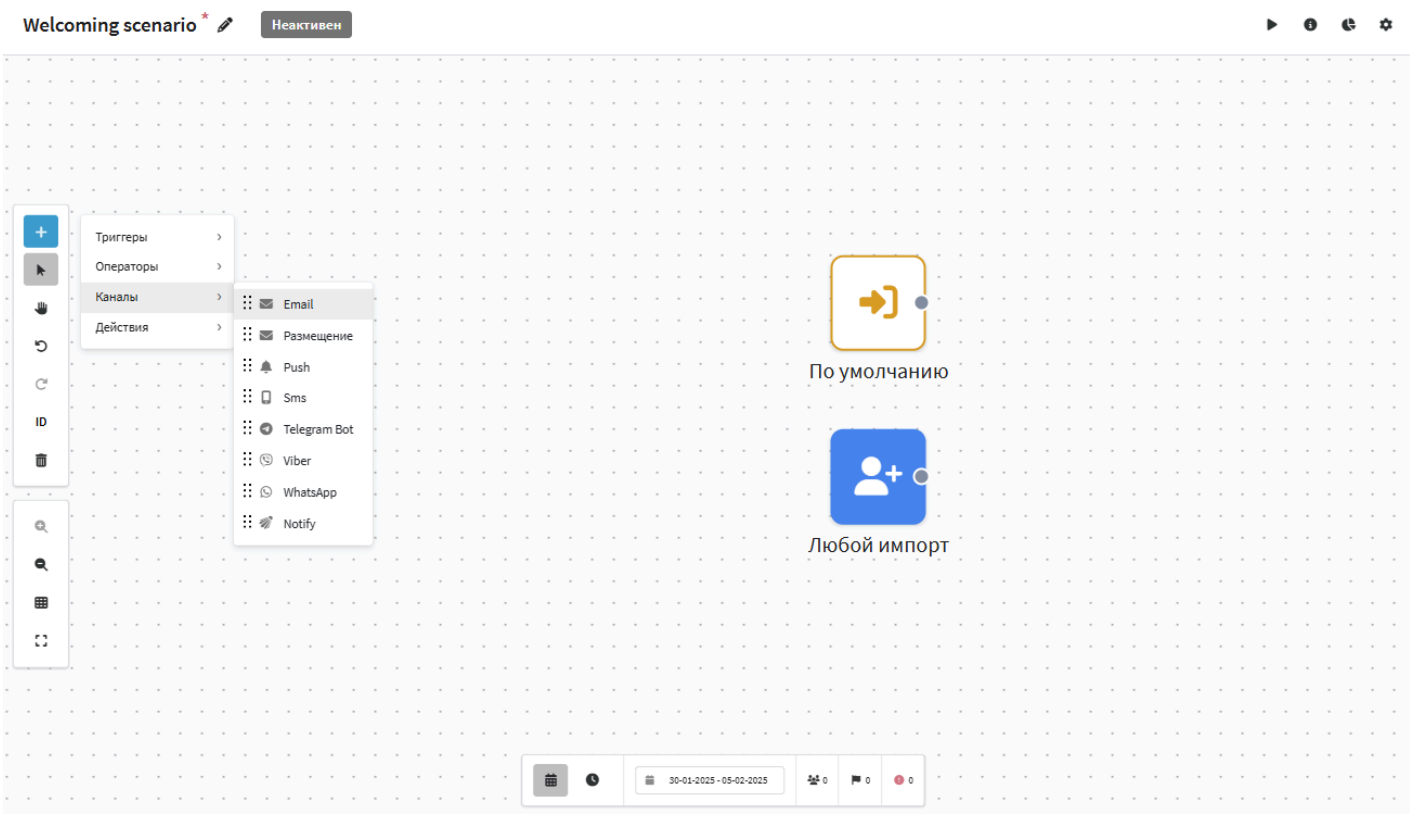
TIP

Различные варианты импорта описаны в этой статье.
Отдельная статья посвящена импорту данных по расписанию.
Вызовы API описаны в этом разделе документации.

Для того чтобы все новые профили из базы попадали в сценарий, используем триггер **“При любом импорте”**:



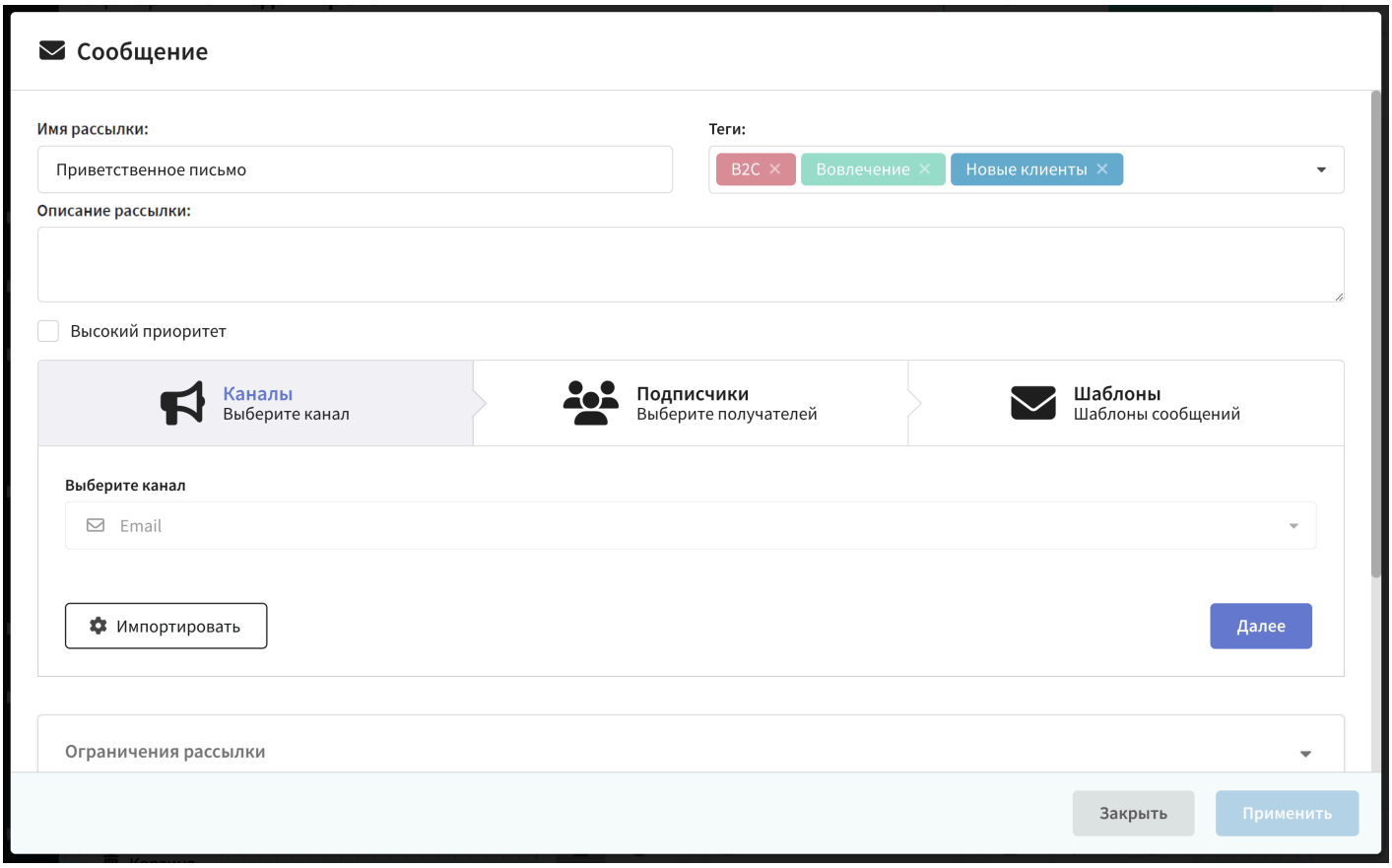
Шаг 2: Создайте приветственное сообщение



Добавьте следующий элемент в сценарий – **рассылку по email**. Для этого нажмите на **“+ Элемент – Каналы – Email”**. Откроется окно настройки рассылки. Здесь вы можете указать **канал, ресурс и шаблон сообщения**.

TIP

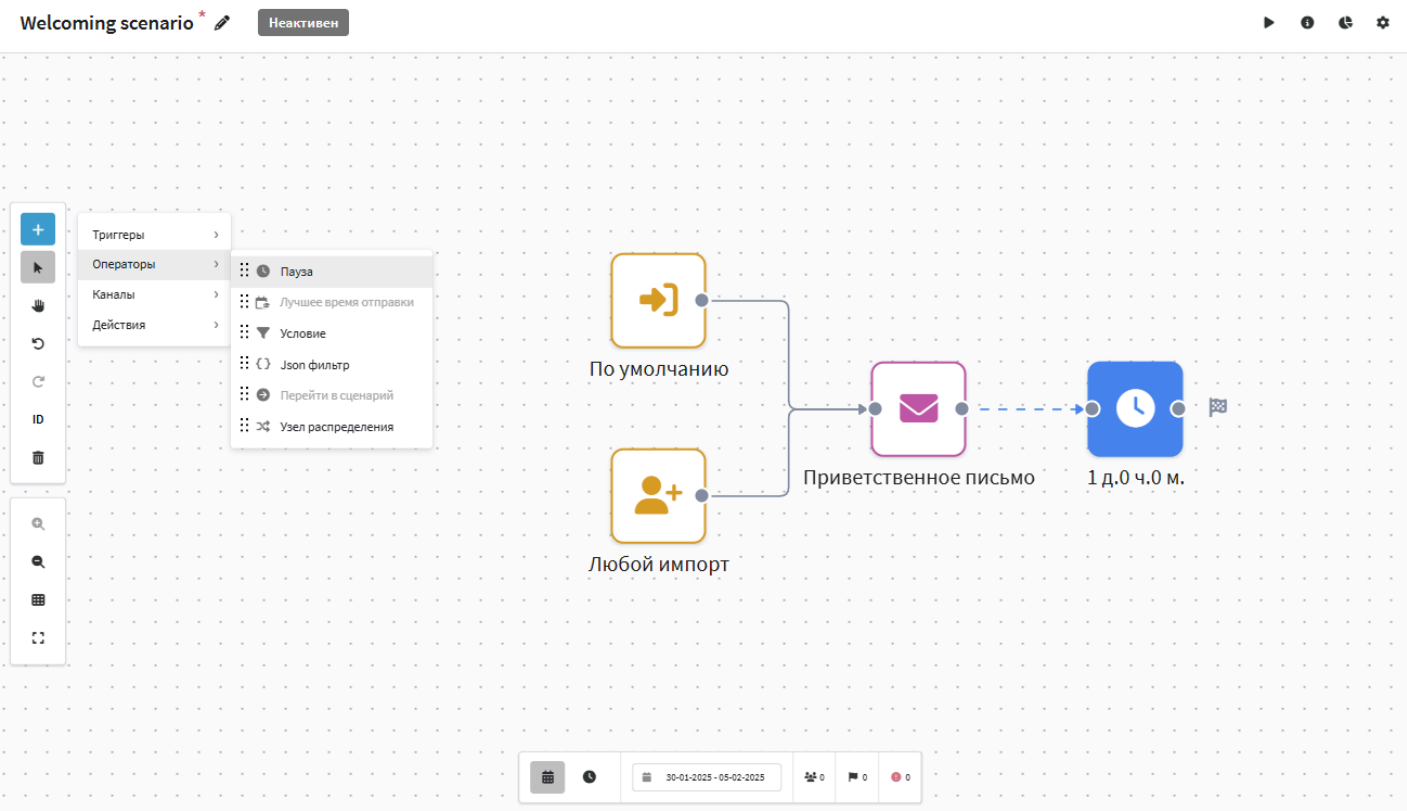
Пошаговое создание **Email-рассылки** вы можете найти здесь.
Работа с **ресурсами подписок** описана в отдельной статье.
А в этом разделе вы найдёте больше информации о работе с **шаблонами сообщений**.



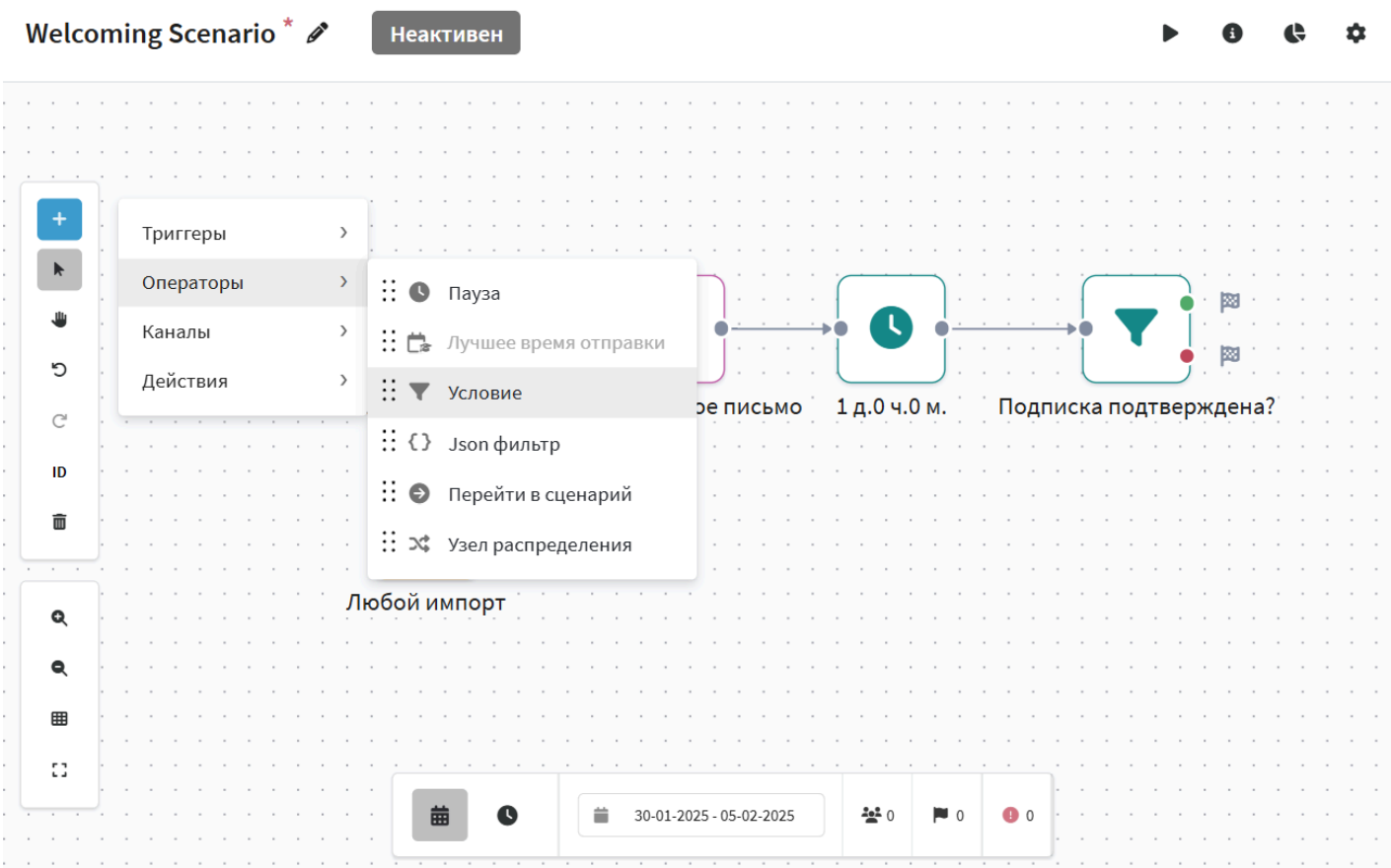
Шаг 3: Настройте цепочку для активных клиентов

После того, как вы отправили приветственное письмо, часть клиентов его прочитает, перейдёт по ссылке и совершит покупку. Процент тех, кто прошёл по этому пути называется процентом **конверсии** вашей приветственной рассылки. Теперь, чтобы повысить этот процент, нужно разнообразить **путь клиента** (Customer Journey).

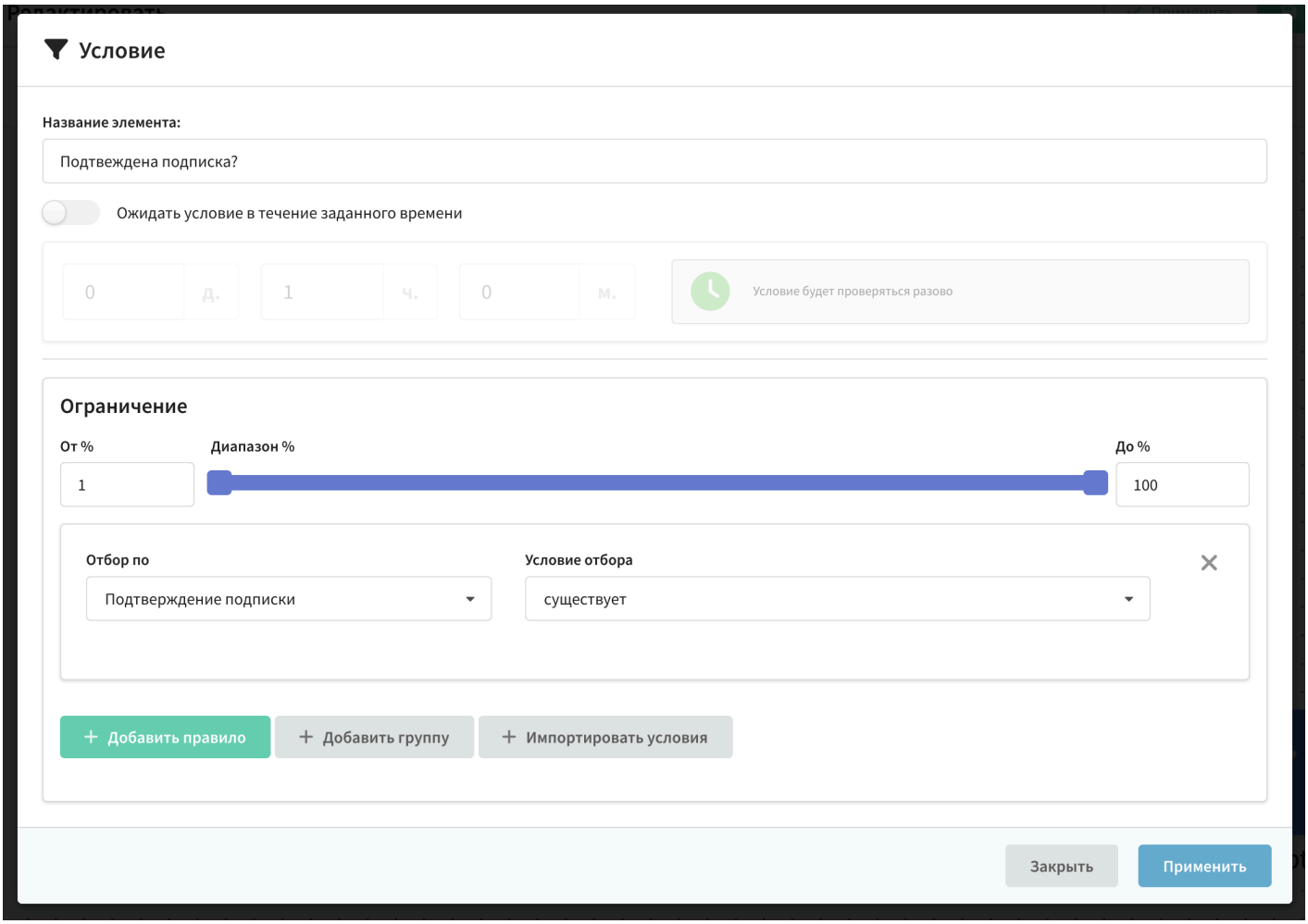
Следующим элементом будет **пауза**. С её помощью мы дадим клиентам время на прочтение письма. Элемент “Пауза” находится в категории “Операторы”:



Дальше с помощью элемента **Условие** отделим тех подписчиков, кто открыл приветственное сообщение и подтвердил регистрацию в первый же день:

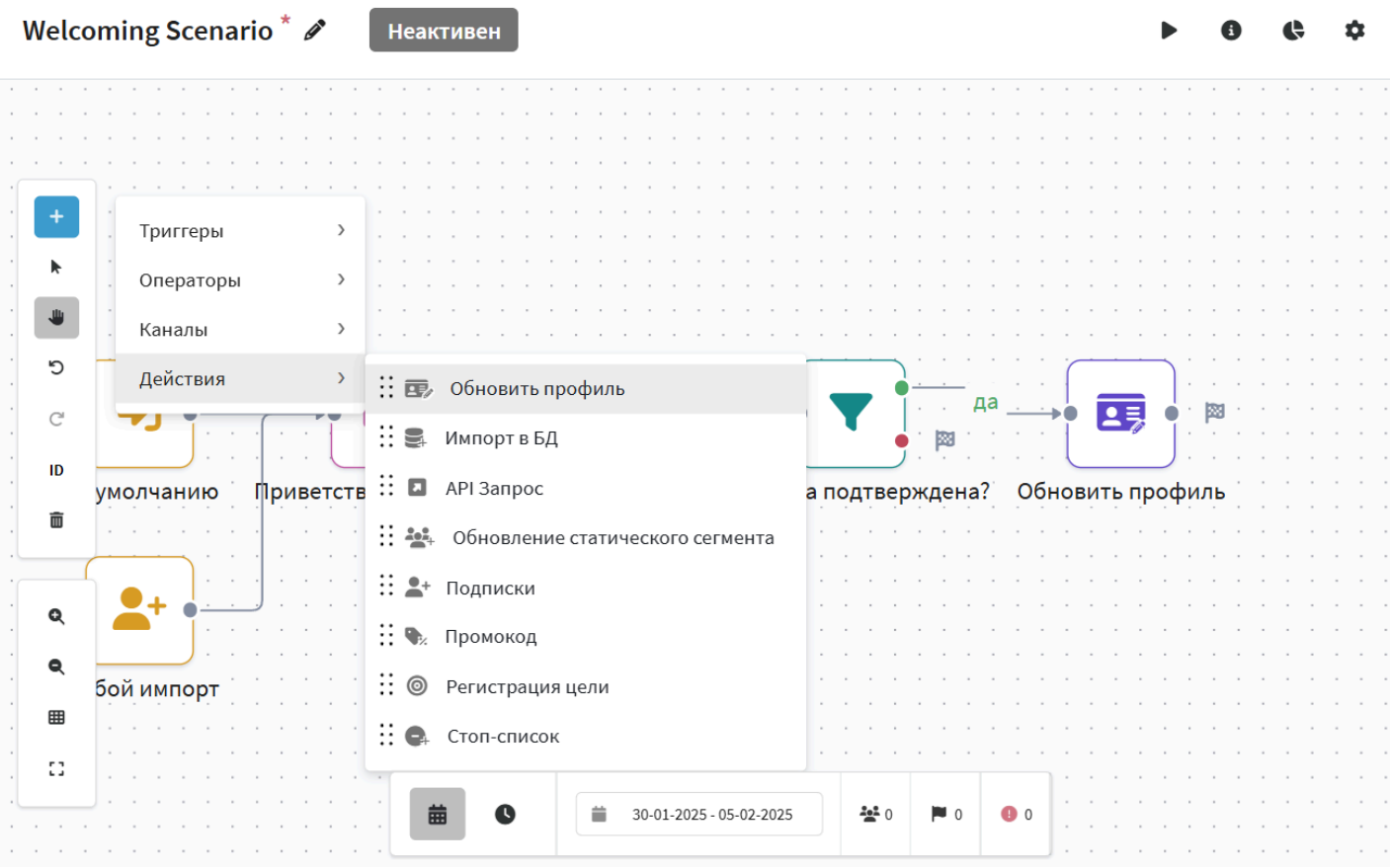


Вместо подтверждения регистрации вы можете использовать правило “Открыт Email” или “Клик на ссылку”. В примере использовано подтверждение подписки в рамках стратегии [DOI (double opt-in)]:

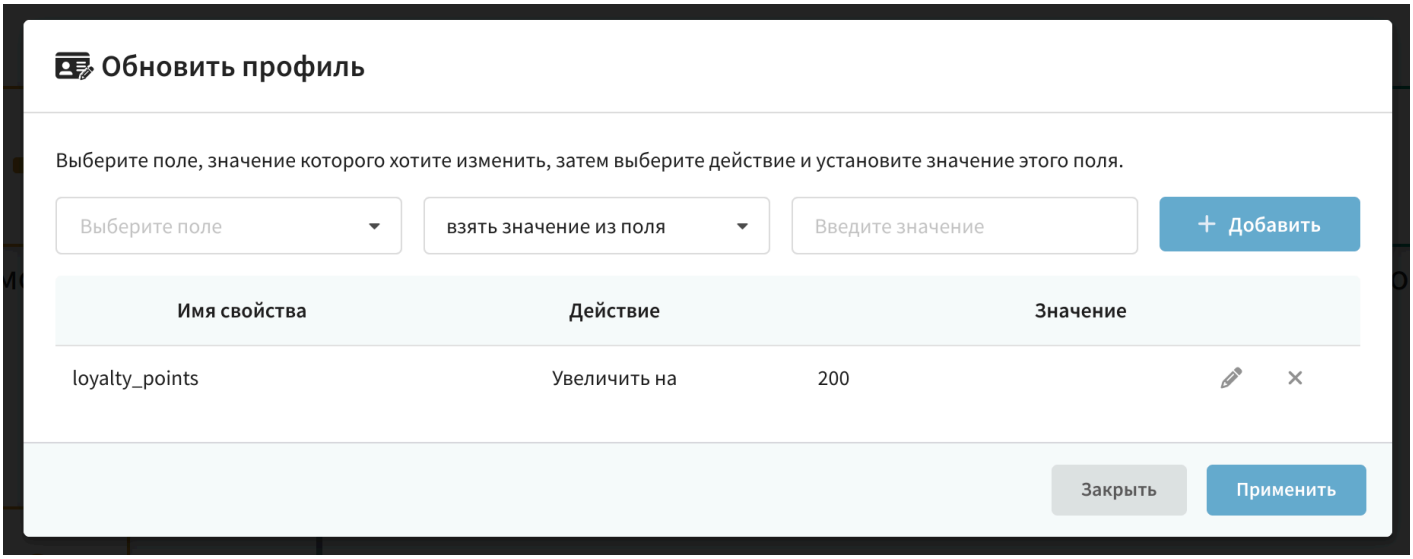


Эти клиенты, скорее всего, сильно заинтересованы в покупке. Хорошей идей будет вознаградить таких клиентов, например, вы можете сразу предоставить им бонусные баллы. В дальнейшем их можно использовать для назначения промо-кодов, скидок или подарков.

Такие действия в маркетинге называются **скоринг**. Скоринг – один из элементов геймификации продаж, способный существенно повысить конверсию. Для его реализации в платформе необходимо предварительно создать в базе профилей числовое поле для значения баллов лояльности. Обновлять эти баллы в сценарии можно при помощи элемента **“Обновить профиль”**:



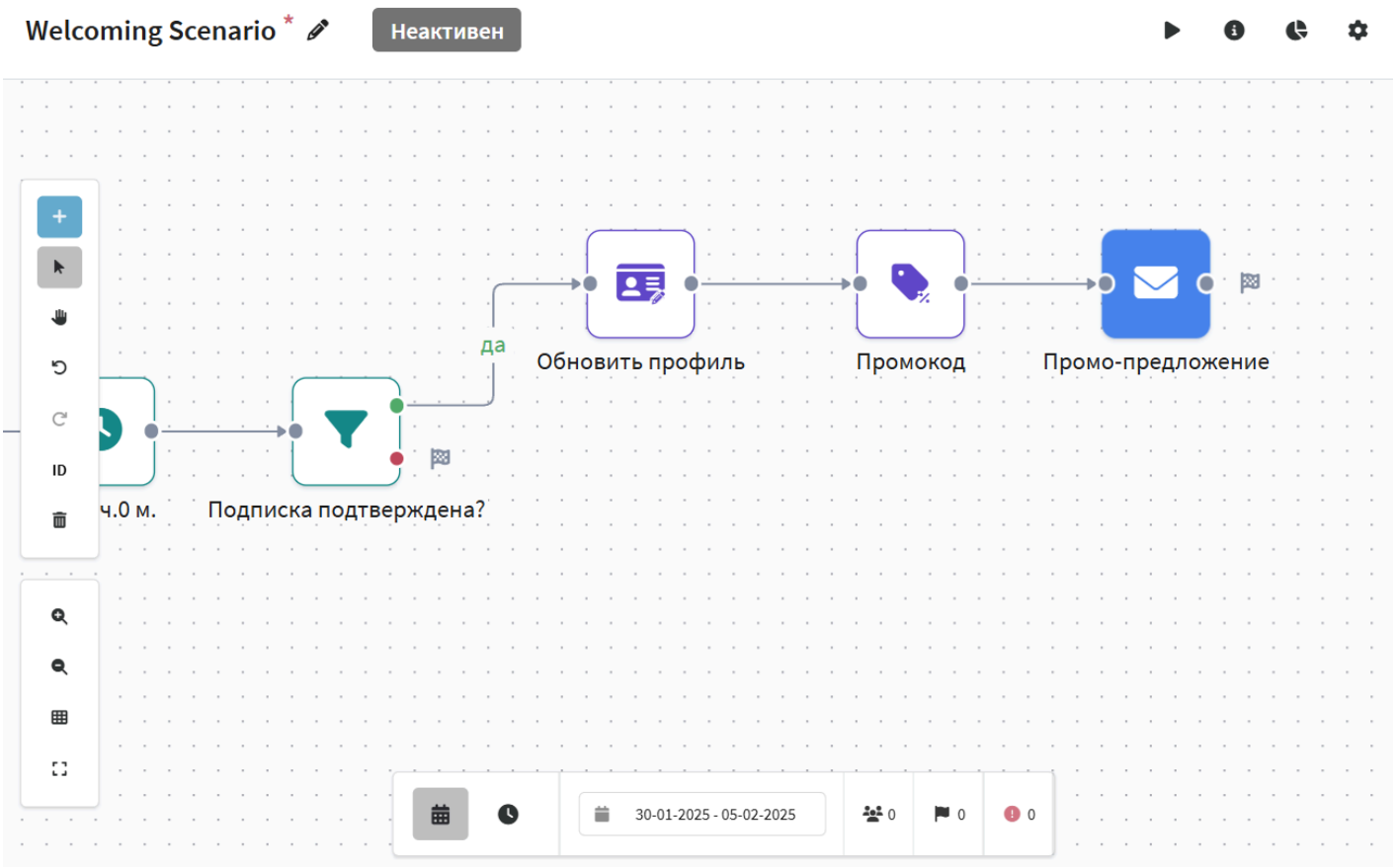
Добавьте элемент на верхнюю (зелёную) ветку, идущую от условия. В настройках элемента укажите поле, в котором будут содержаться баллы лояльности, и увеличьте его значение на необходимую величину:



Далее вы можете назначить новым клиентам **промокоды** (например – на часть товара, служащего специально для разогрева продаж) и отправить клиентам **промо-предложения**. Для этого добавьте элемент **Промокод** и **сообщение по email-каналу**.

TIP

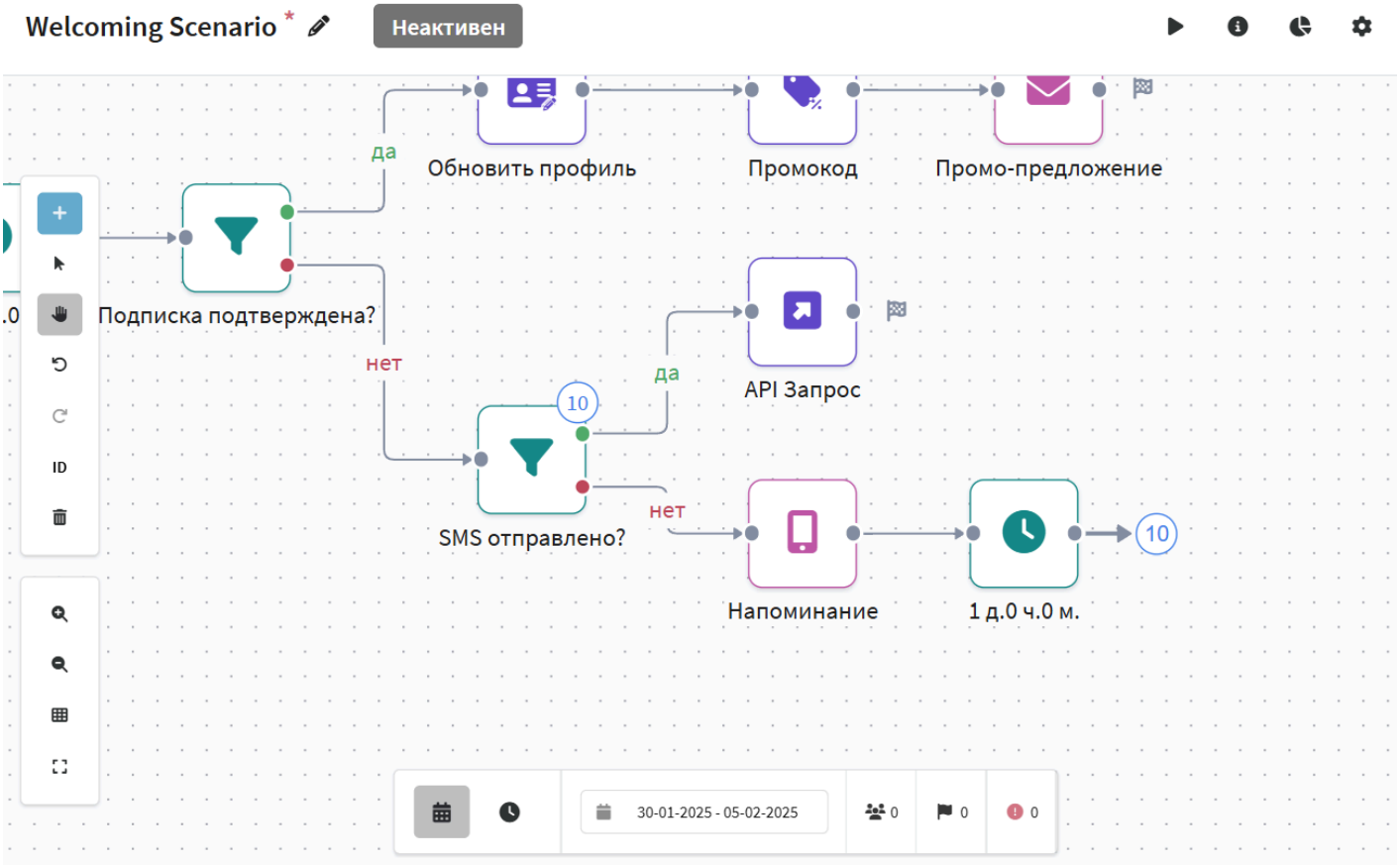
Промокоды создаются **программами лояльности Altcraft Marketing**. Подробнее вы можете прочитать в этой статье.



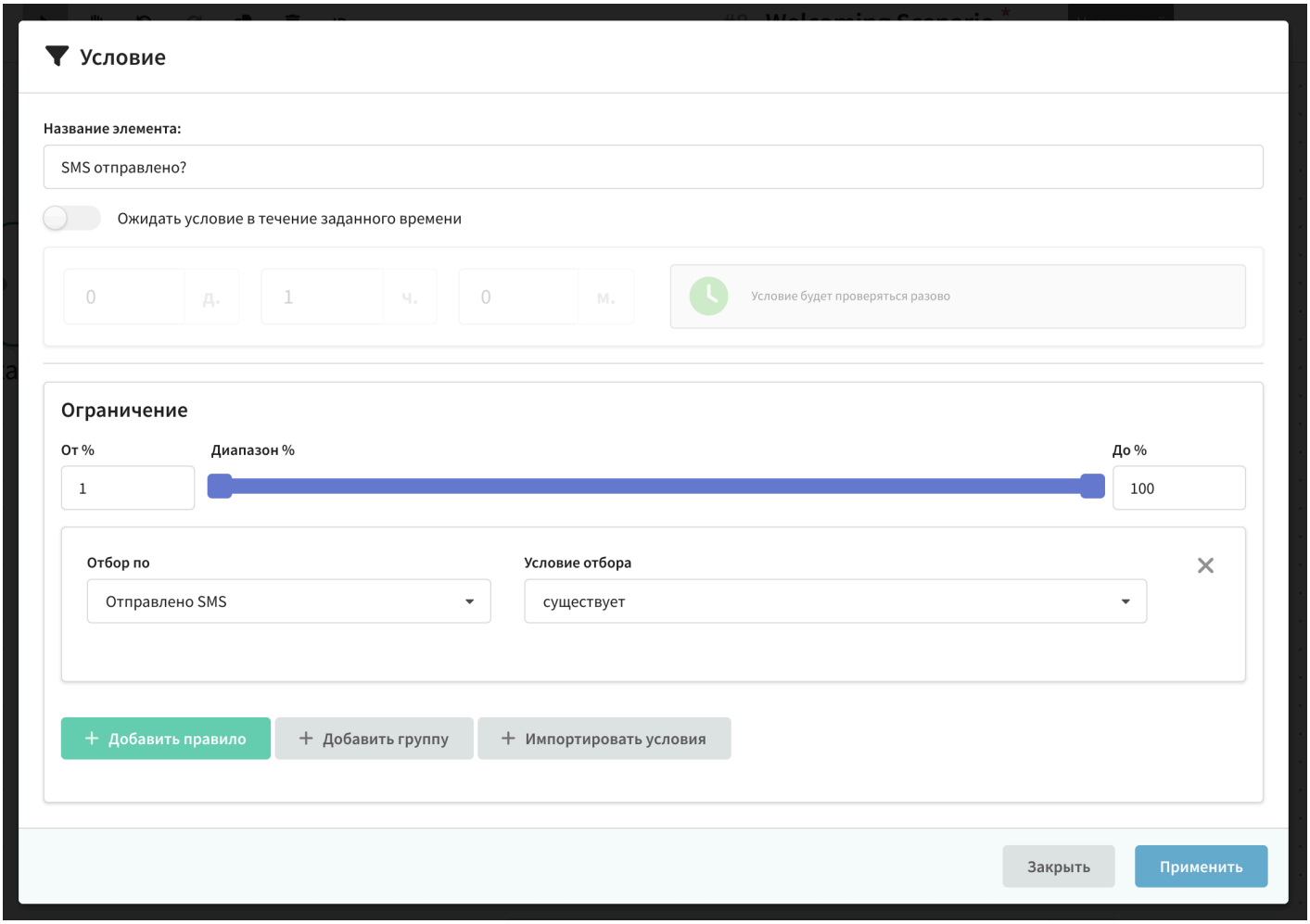
Шаг 4: Настройте цепочку для неактивных клиентов

Часть клиентов, которым вы отправили приветствие, могли забыть о нём, не получить, отложить или просто не заинтересоваться предложением. В таком случае вы можете напомнить им о незаконченной регистрации. Лучше всего использовать **SMS-канал**: телефон есть у каждого и всегда находится под рукой.

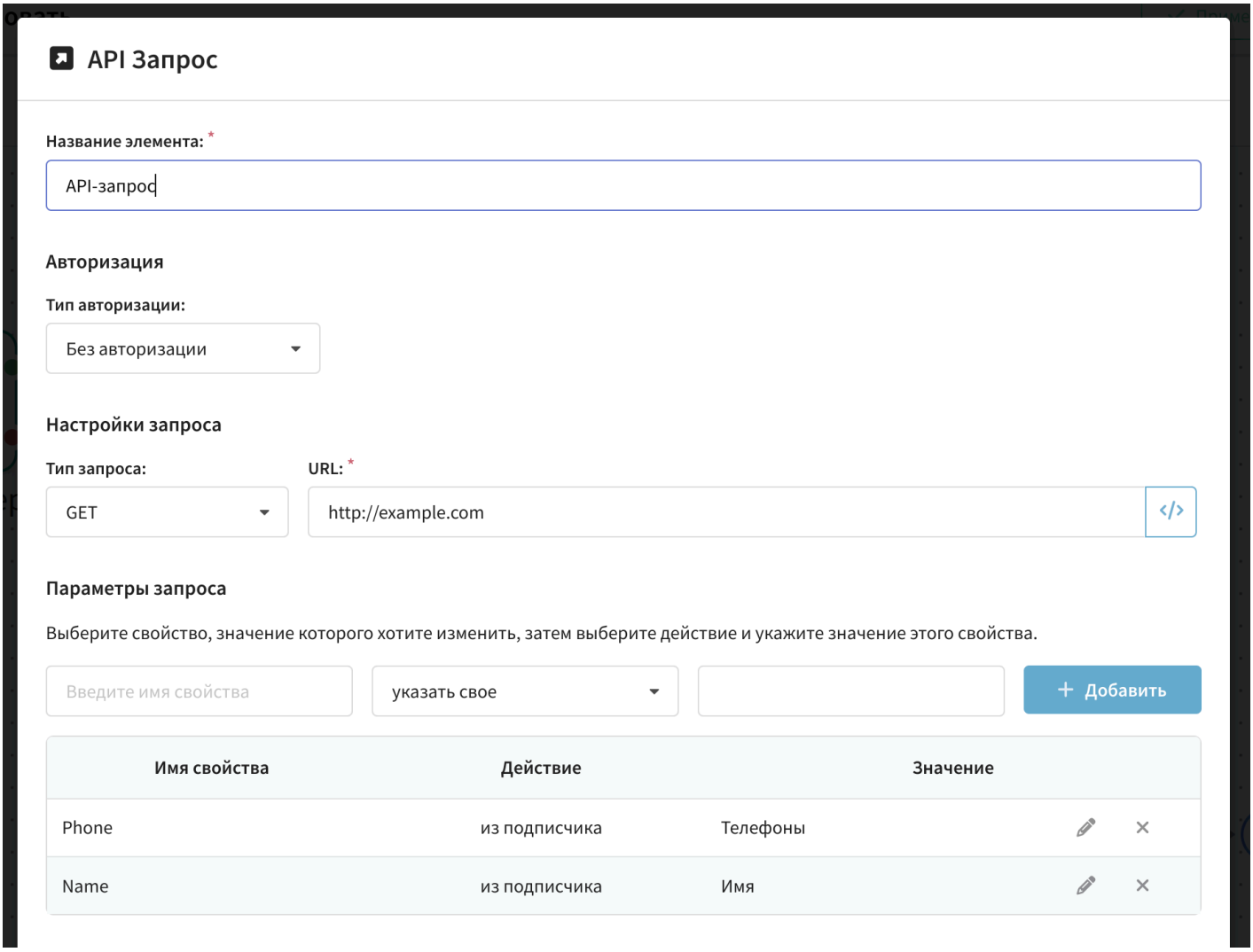
Однако, обычно отправлять одно и то же напоминание больше одного раза не всегда целесообразно. Гораздо эффективнее будет использовать различные тактики реактивации. В примере применим две: сначала отправляется **SMS**, а если на него нет реакции – менеджерам или операторам колл-центра отправляется **уведомление** с контактными данными из профиля, чтобы сотрудник мог лично пообщаться с клиентом. Наша “ветвь” для неактивных клиентов будет выглядеть так:



Чтобы применить две тактики одновременно, добавьте элемент **Условие** для тех, кому хотя бы раз было отправлено SMS-напоминание. Выберите правило **“Существует отправка SMS”**:



Сверху (по зелёной ветке) добавьте элемент **Вызов API**. В ней укажите сетевой адрес, тип запроса и данные, которые нужно передать в колл-центр. Свяжитесь с системными администраторами вашей компании, чтобы получить подробности:

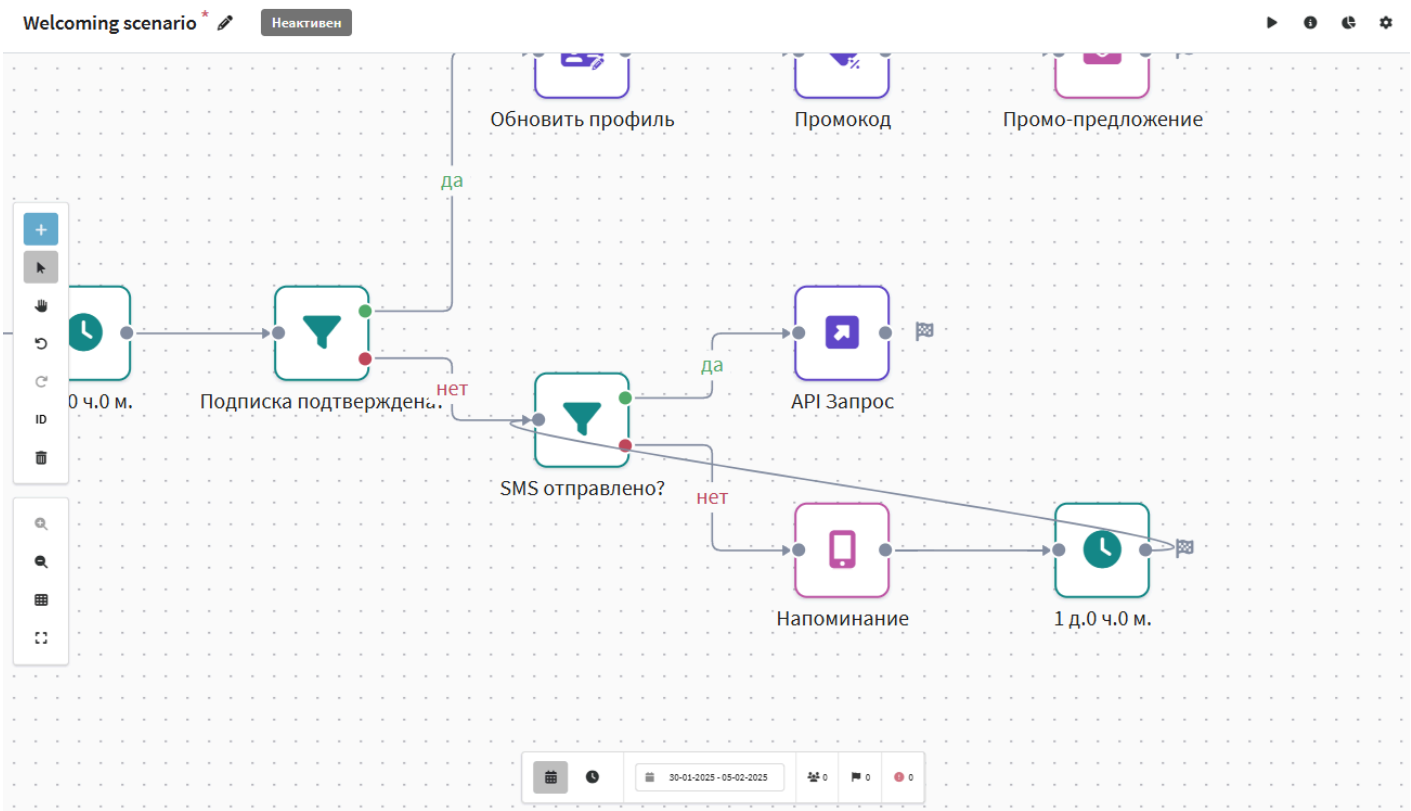


Вниз от условия (по красной ветке) добавьте элемент с SMS-сообщением, в которой создайте новую или импортируйте готовую SMS рассылку.

TIP

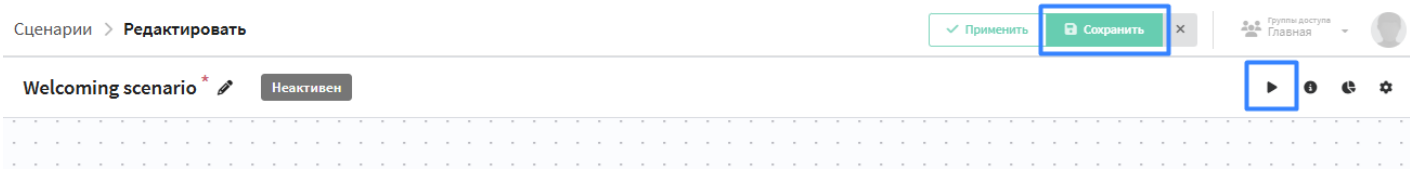
Пошаговое руководство по созданию SMS-рассылки вы найдёте здесь.

После сообщения добавьте **Паузу** – так же как после Email-приветствия. Когда время после отправки SMS заканчивается, клиента нужно отправить в начало сценария, на повторную проверку подтверждения подписки в элементе **Условие**. Для этого протяните курсор от правой точки элемента “Пауза” к условию по отправке SMS:

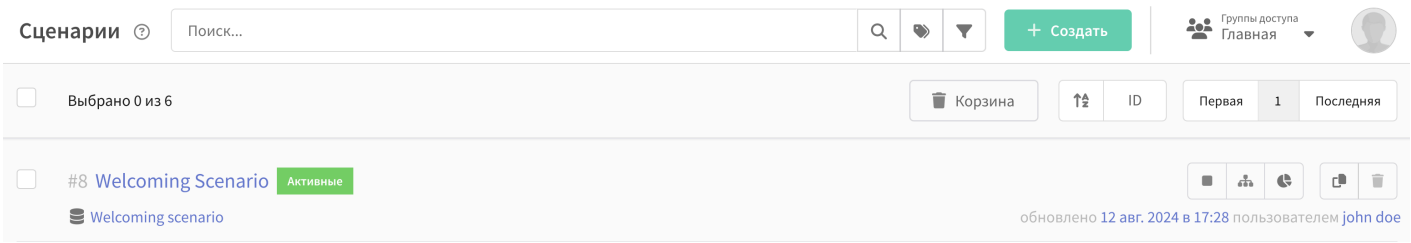


Шаг 5: Сохраните и активируйте сценарий

После того, как вы внесли необходимые настройки и сконфигурировали шаги сценария, **сохраните** и **активируйте его**, нажав на соответствующие кнопки в верхней части экрана:



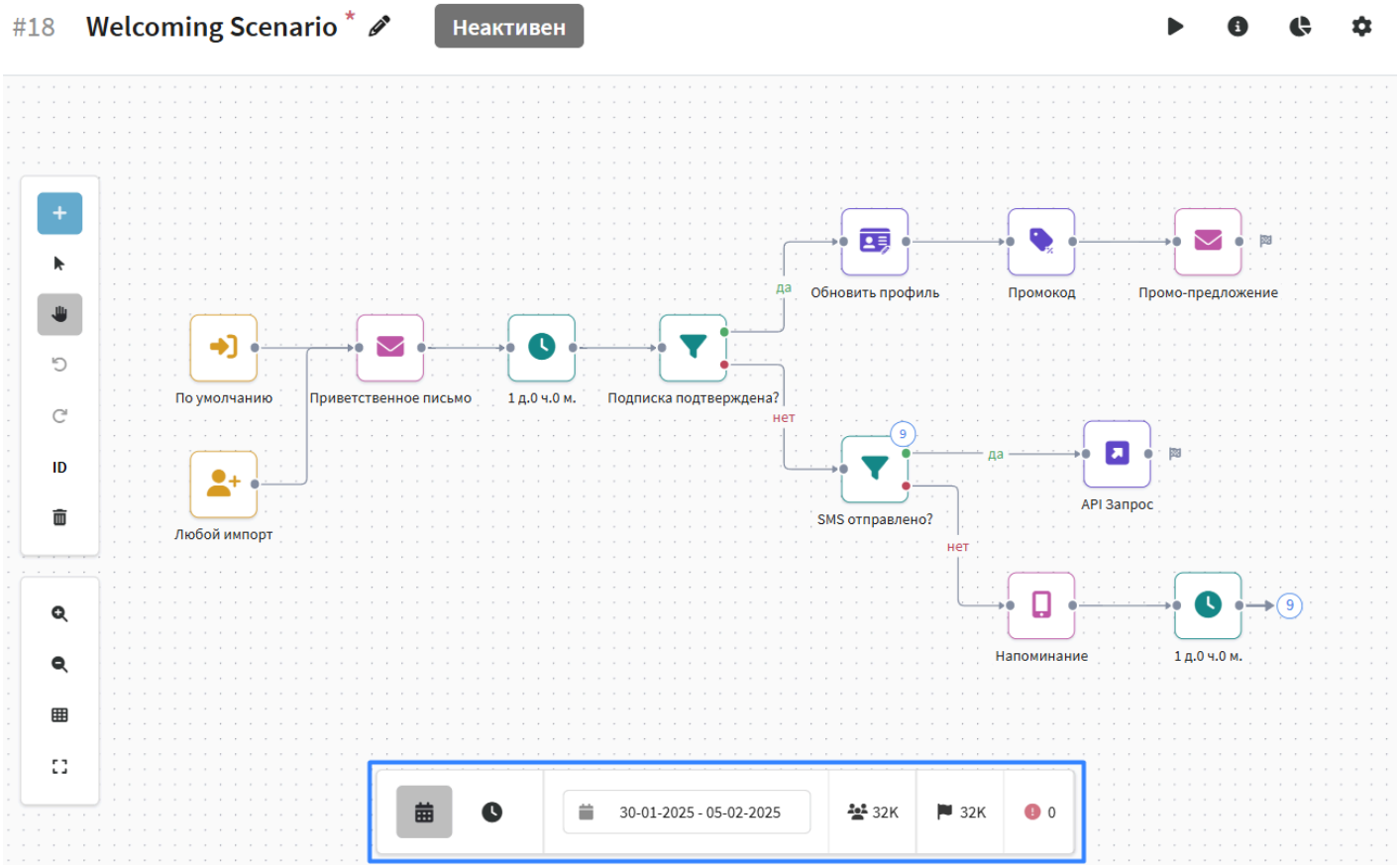
Приветственная последовательность станет отличным началом для эффективного долгосрочного общения с клиентами. Открыть сценарий для просмотра и редактирования вы можете нажав на его название:



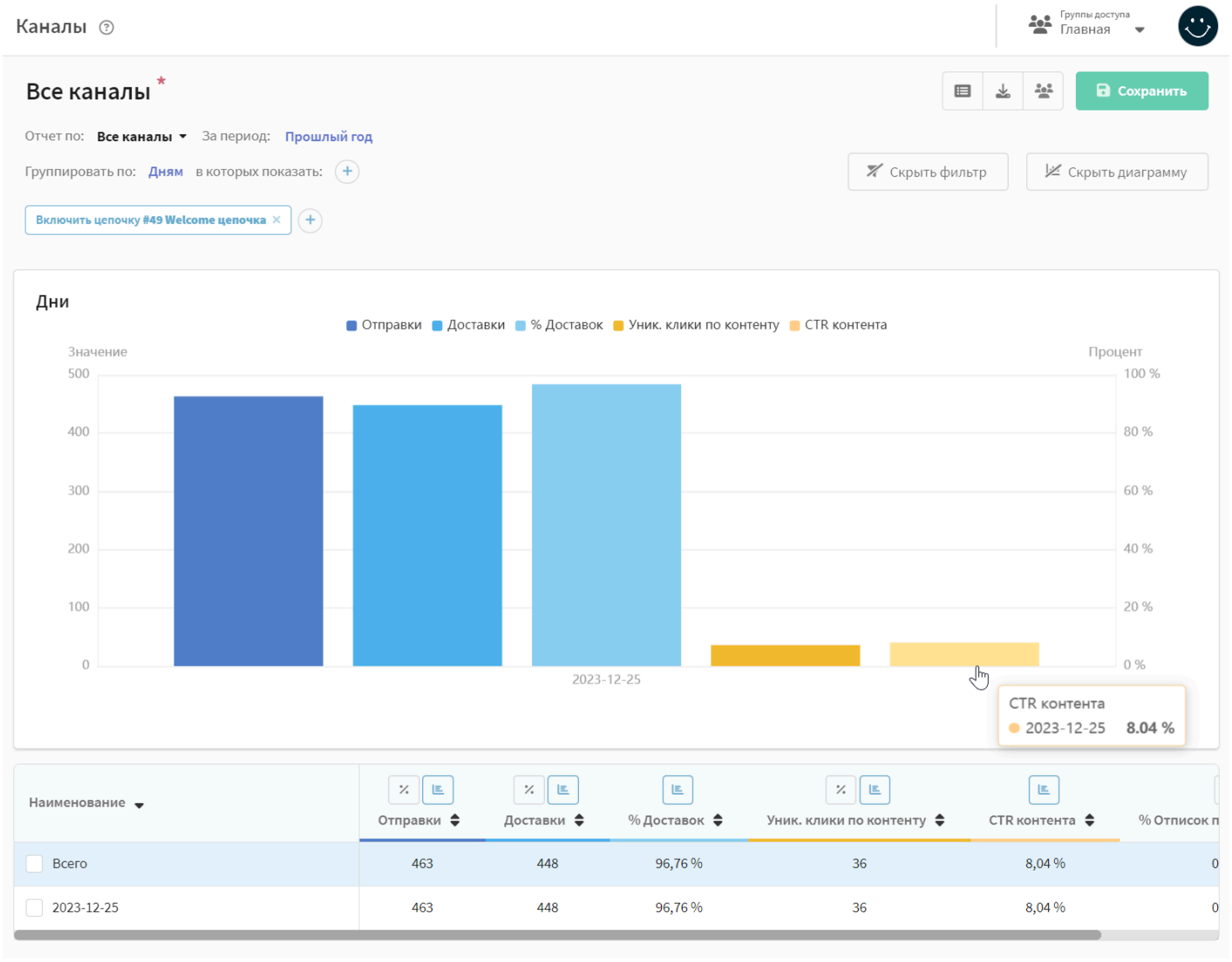
Шаг 6: Оцените эффективность

Оценить эффективность сценария вы сможете, используя несколько аналитических инструментов.

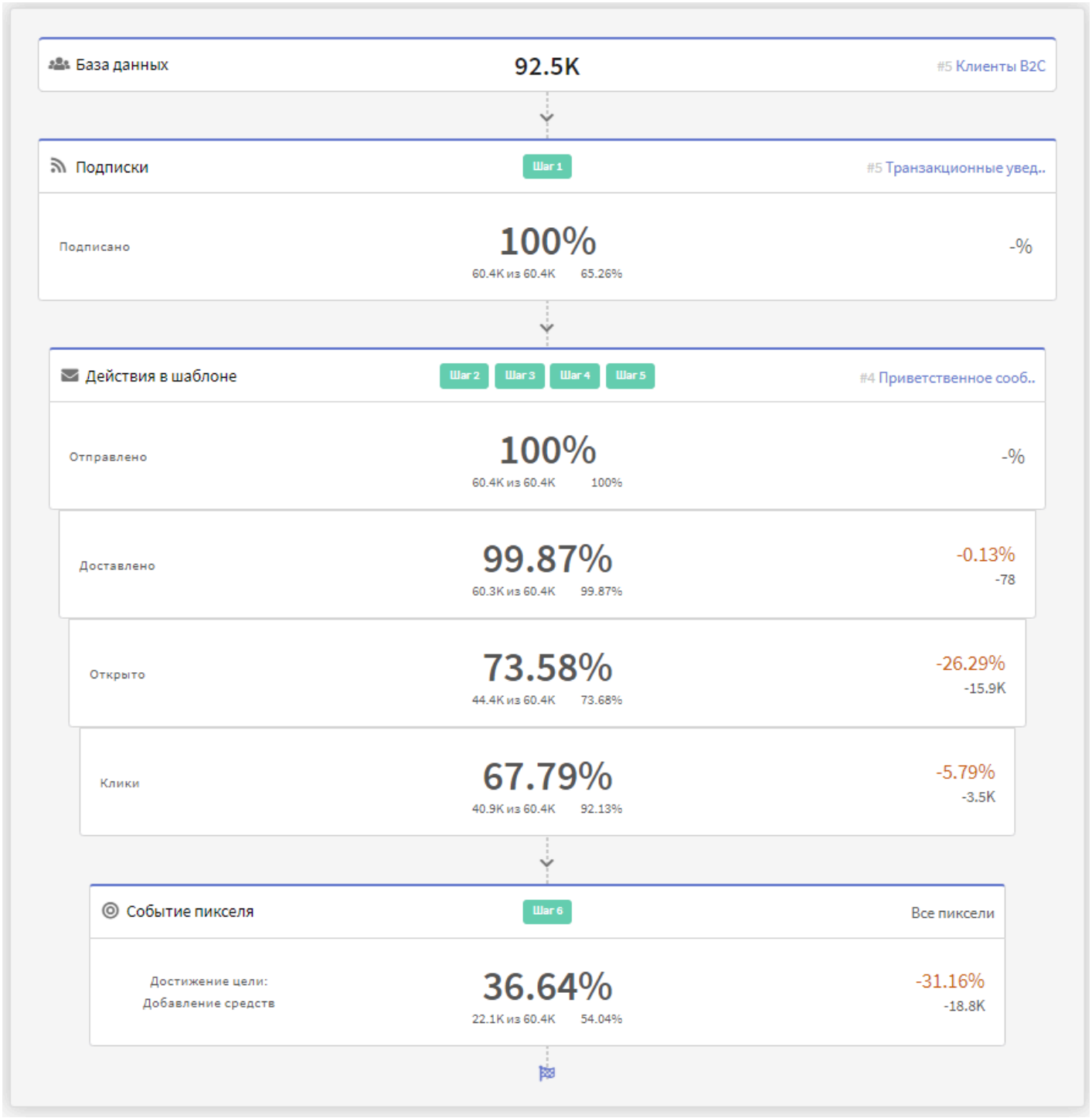
Первый из них – **статистика** внутри сценария. Выберите нужный период и просмотрите показатели прохождения:



Второй – [отчёт по каналам](#) сценария. Он предоставит подробную информацию по рассылкам в рамках сценария:



Наконец, вы можете добавить рассылки и другие события сценария в отчёт-воронку, чтобы ещё увереннее контролировать показатели конверсии. Шагом воронки может быть попадание клиента под условия выборки или достижение определённой цели (Покупка, Внесение средств).


База данных
92.5K
#5 Клиенты B2C
Подписки
Шаг 1
#5 Транзакционные увед..
Подписано
100%
60.4K из 60.4K 65.26%
-%
Действия в шаблоне
Шаг 2
Шаг 3
Шаг 4
Шаг 5
#4 Приветственное сооб..
Отправлено
100%
60.4K из 60.4K 100%
-%
Доставлено
99.87%
60.3K из 60.4K 99.87%
-0.13%
-78
Открыто
73.58%
44.4K из 60.4K 73.68%
-26.29%
-15.9K
Клики
67.79%
40.9K из 60.4K 92.13%
-5.79%
-3.5K
Событие пикселя
Шаг 6
Все пиксели
Достижение цели:
Добавление средств
36.64%
22.1K из 60.4K 54.04%
-31.16%
-18.8K

Автоматическое оповещение менеджера через сценарий

Рассмотрим типовой кейс, который может быть реализован с помощью сценариев в платформе Altcraft MP.

Задача — необходимо, чтобы при совершении пользователем какого-либо целевого действия менеджеру приходило письмо с информацией о клиенте.

Рассмотрим реализацию данного функционала через вызов API из сценария. В данном примере предполагается, что база лидов уже создана и наполнена.

Шаг 1. Создание базы менеджеров

Прежде всего, необходимо создать базу данных (для примера назовем **“Менеджеры”**). Для базы необходимо задать настройки базы и структуру ее полей. После того, как база создана, нужно создать в ней профиль.

Подробнее о базах профилей можно узнать здесь.

Шаг 2. Создание ресурса и подписок

После того, как создана база менеджеров и в нее добавлены профили, необходимо создать ресурс (в данном примере назовем его **“Оповещения менеджеров”**). При создании ресурса нужно указать в его настройках базу **“Менеджеры”** . После того, как ресурс создан, необходимо подписать на него профиль менеджера. В примере добавим подписку вручную:

Подробнее о ресурсах можно узнать здесь.

Шаг 3. Создание шаблона

На этом этапе мы создадим шаблон письма (для примера назовем его **“Оповещение”**), которое будет приходить менеджеру. Для персонализации сообщений мы будем использовать переменные **apicontent** :

Email клиента: {apicontent.mail}

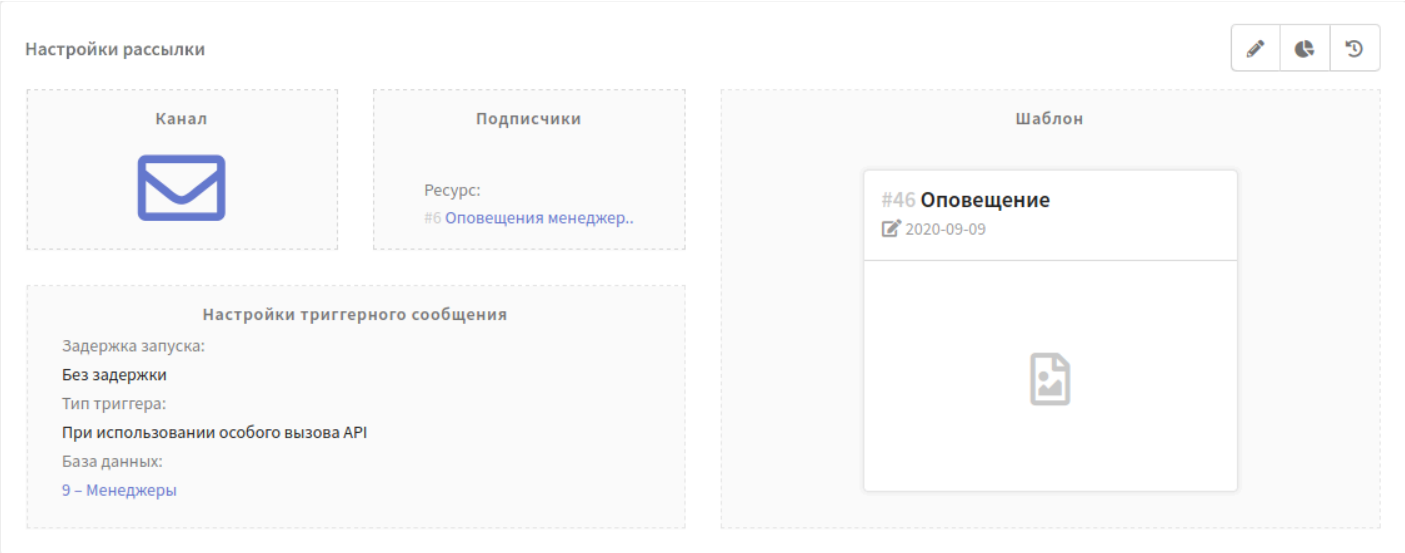
Номер телефона клиента: {apicontent.ph_number}

Пример шаблона для оповещения менеджера:

Подробнее о шаблонах можно узнать здесь.

Шаг 4. Создание триггера

В качестве средства оповещения менеджера будет использоваться триггерная рассылка. При ее создании нужно выбрать ранее созданный ресурс **“Оповещения менеджеров”,** шаблон **“Оповещение”** , условие запуска триггера — “При использовании особого API вызова” и базу данных **“Менеджеры”** .



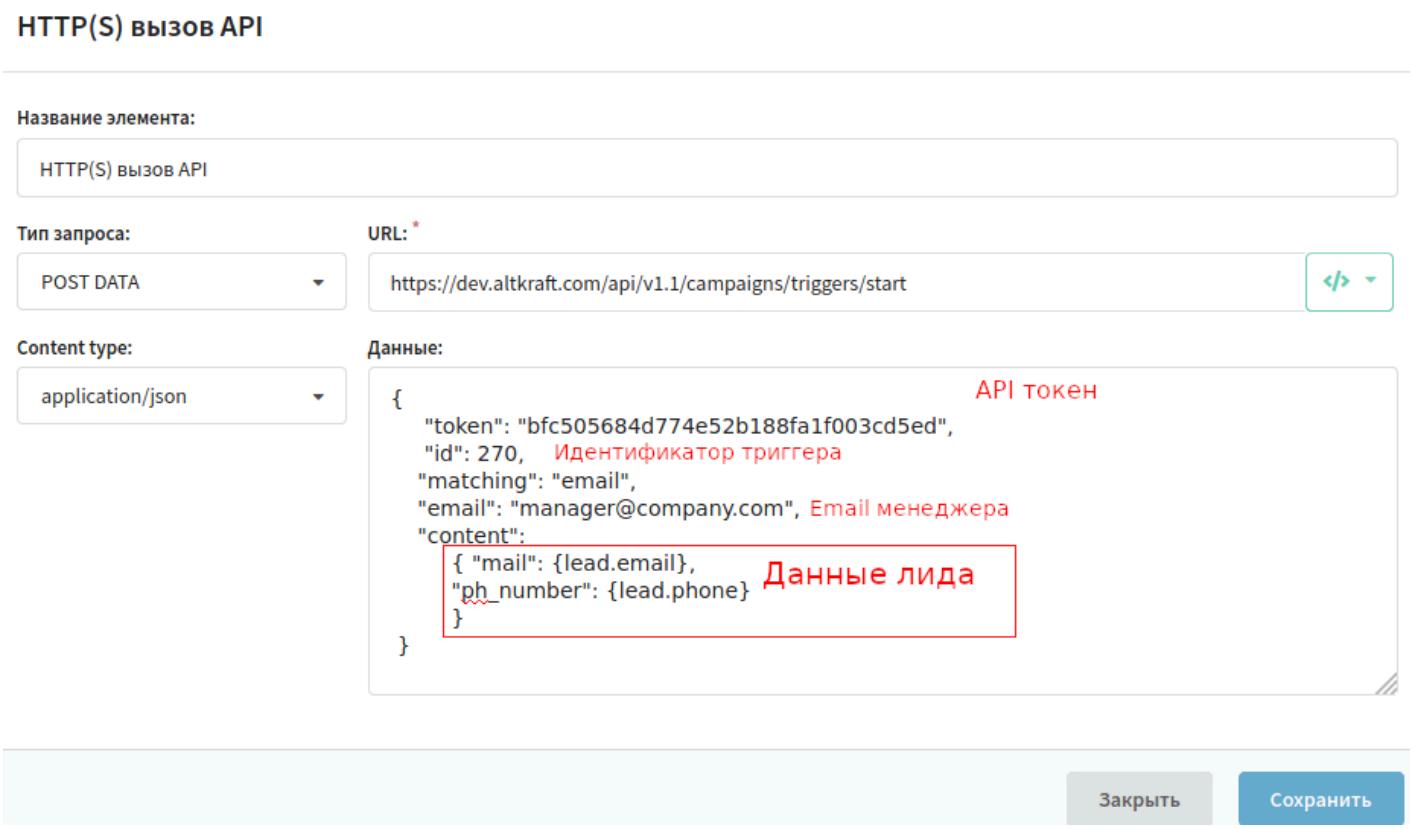
Подробнее о триггерных рассылках можно узнать здесь.

Шаг 5. Создание сценария

Ключевым элементом, который будет обеспечивать вызов триггера с оповещением для менеджером, является узел **“HTTP(S) Вызов API”** . В качестве типа запроса выбираем POST DATA, Content type - JSON либо XML (в данном примере JSON). В URL указывается адрес, по которому будет вызваться триггер (*api/v1.1/triggers/start*). В разделе **Данные** помещается тело API запроса, по которому будет вызываться **“Триггер - оповещение”** (подробнее про API можно узнать здесь).

```
{
  "token": "bfc505684d774e52b188fa1f003cd5ed",
  "id": 270,
  "matching": "email",
  "email": "manager@company.com",
  "content": {
    "mail": "{lead.email}",
    "phones": "{lead.phones}"
  }
}
```

В данном запросе указывается API-токен для авторизации, идентификатор триггера, который будет отправлен менеджеру, email менеджера, а также email и телефон лида, о котором будет оповещен менеджер.



CAUTION

Для авторизации при передаче запроса необходим API-токен, который должен быть создан в платформе. Подробнее можно узнать здесь.

На этом настройка сценария автоматизации оповещения менеджера завершена!

Сценарий брошенной корзины

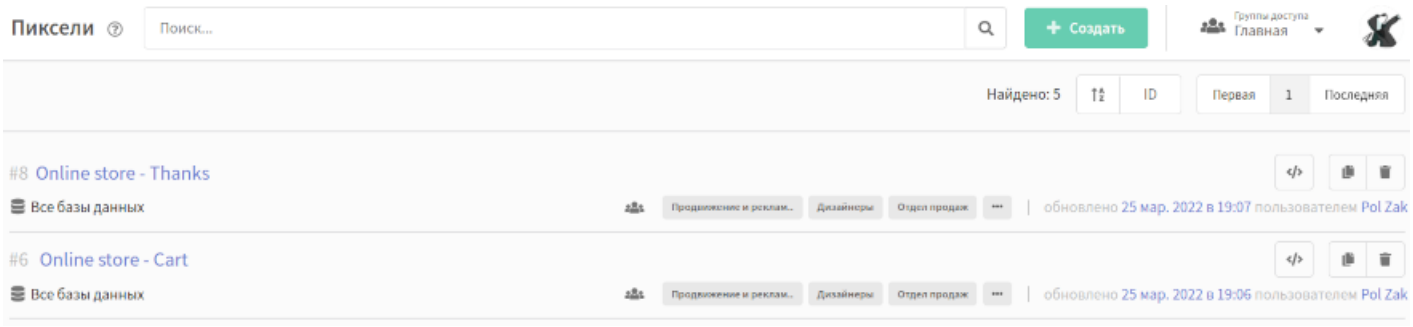
Нередко посетители интернет-магазина добавляют товар в корзину, но не оформляют заказ. Однако если напомнить через некоторое время о брошенных товарах, то шансы на завершение покупки увеличатся. Этот гайд поможет вам настроить короткий сценарий, который автоматически будет отправлять клиенту email-письмо с напоминанием о товарах в корзине.

Шаг 1: Разместите пиксели отслеживания на страницах сайта

Допустим, на нашем сайте пользователь добавляет товары в корзину, переходит в корзину, оплачивает покупку и затем его перенаправляют на страницу с благодарностью. Напоминание о брошенной корзине подписчик получит, если он посетил страницу с корзиной, но после этого не попал на страницу с благодарностью, т.е. не завершил оформление покупки. Отслеживать посещение определенных страниц сайта помогут пиксели.

В разделе **Пиксели** создайте 2 пикселя:

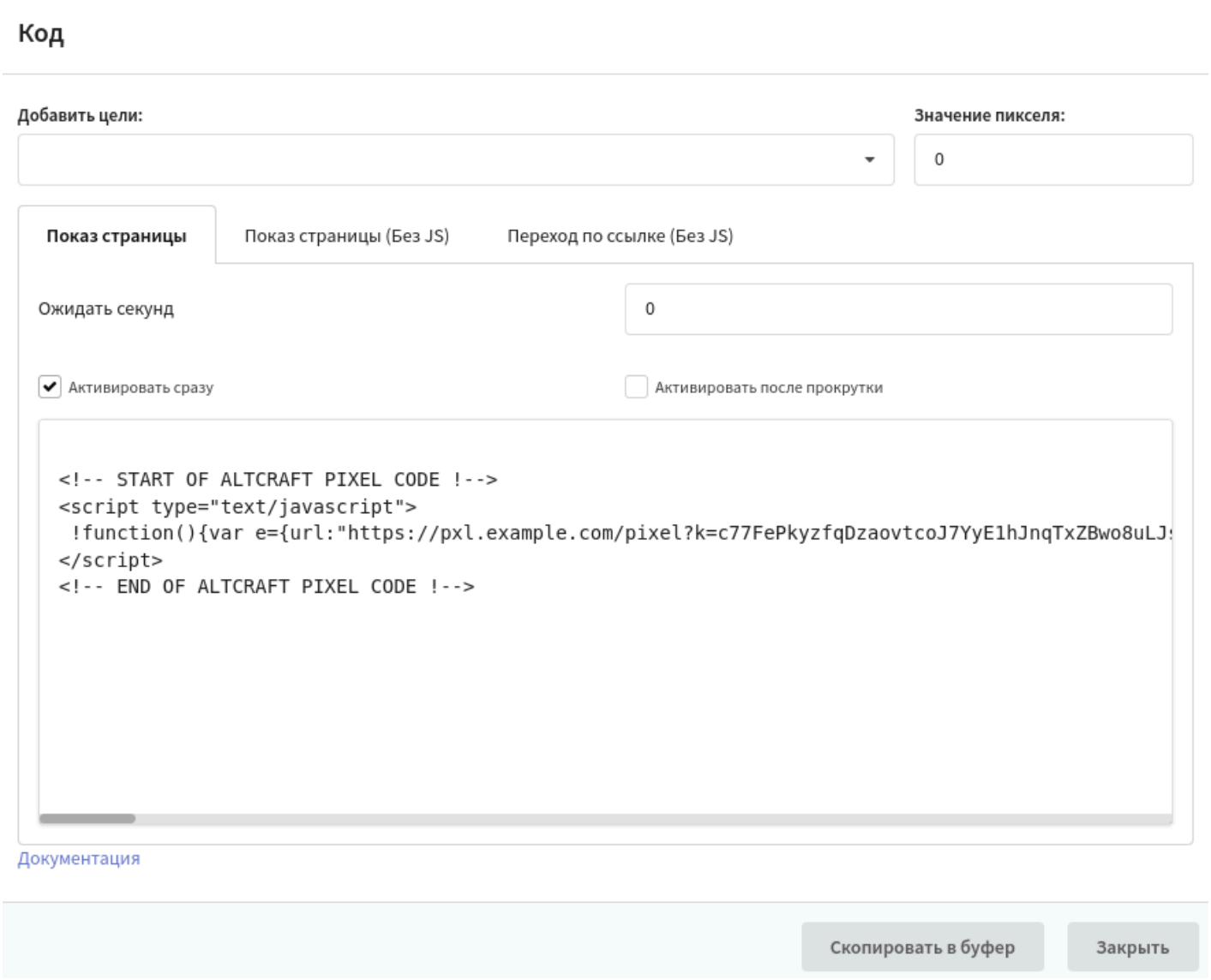
- **Online store - Cart**: отслеживает посещение страницы с корзиной
- **Online store - Thanks**: отслеживает посещение страницы с благодарностью



TIP

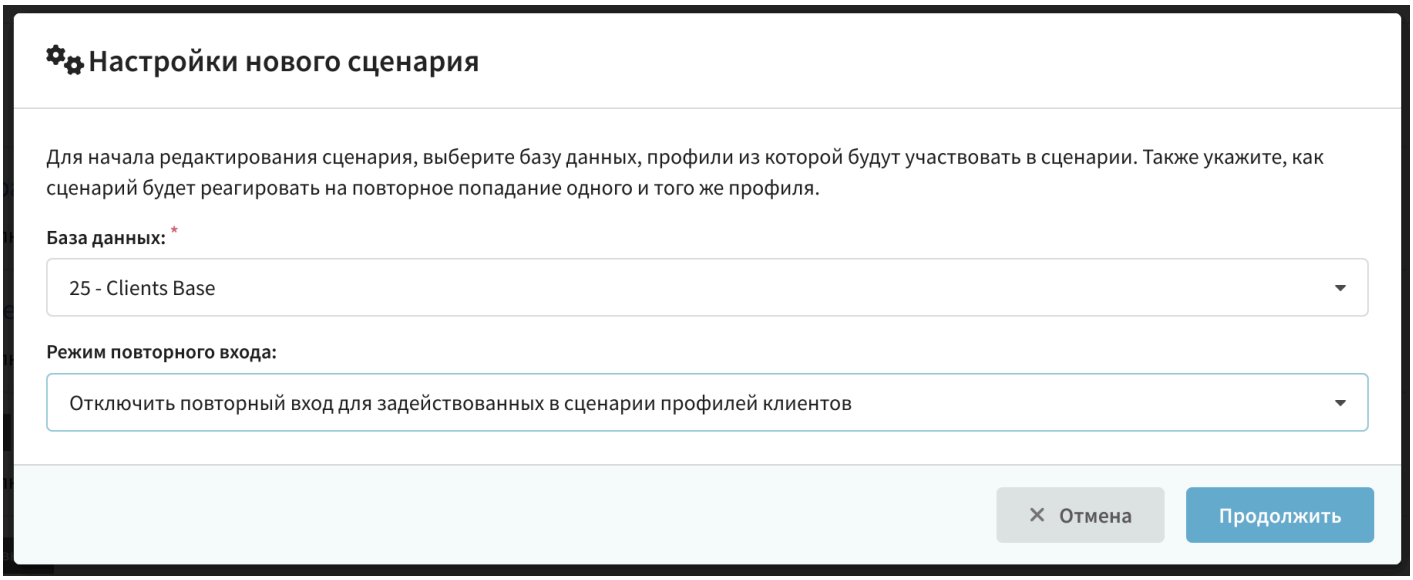
В некоторых ситуациях посетитель сайта может открывать корзину не один раз. Чтобы не заспамить почту клиента, отправляя несколько email-напоминаний в одну неделю, **блокируйте повторное срабатывание** пикселя для корзины в течение 7200 минут (5 дней).

Чтобы посмотреть код пикселя, нажмите кнопку “**Показать код**”. Скопируйте код из открывшегося окна и разместите на соответствующих страницах сайта:

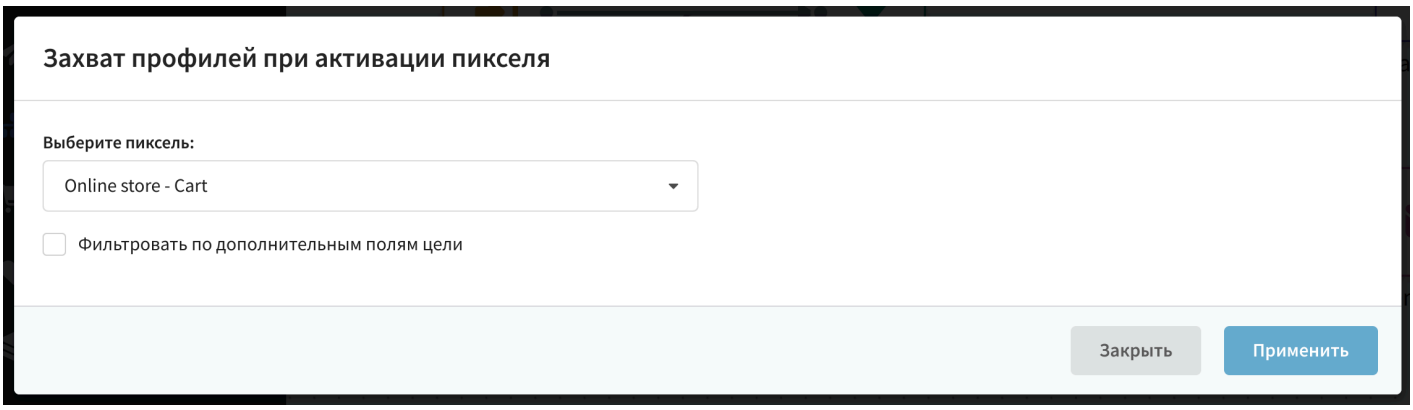


Шаг 2: Настройте захват профилей клиентов

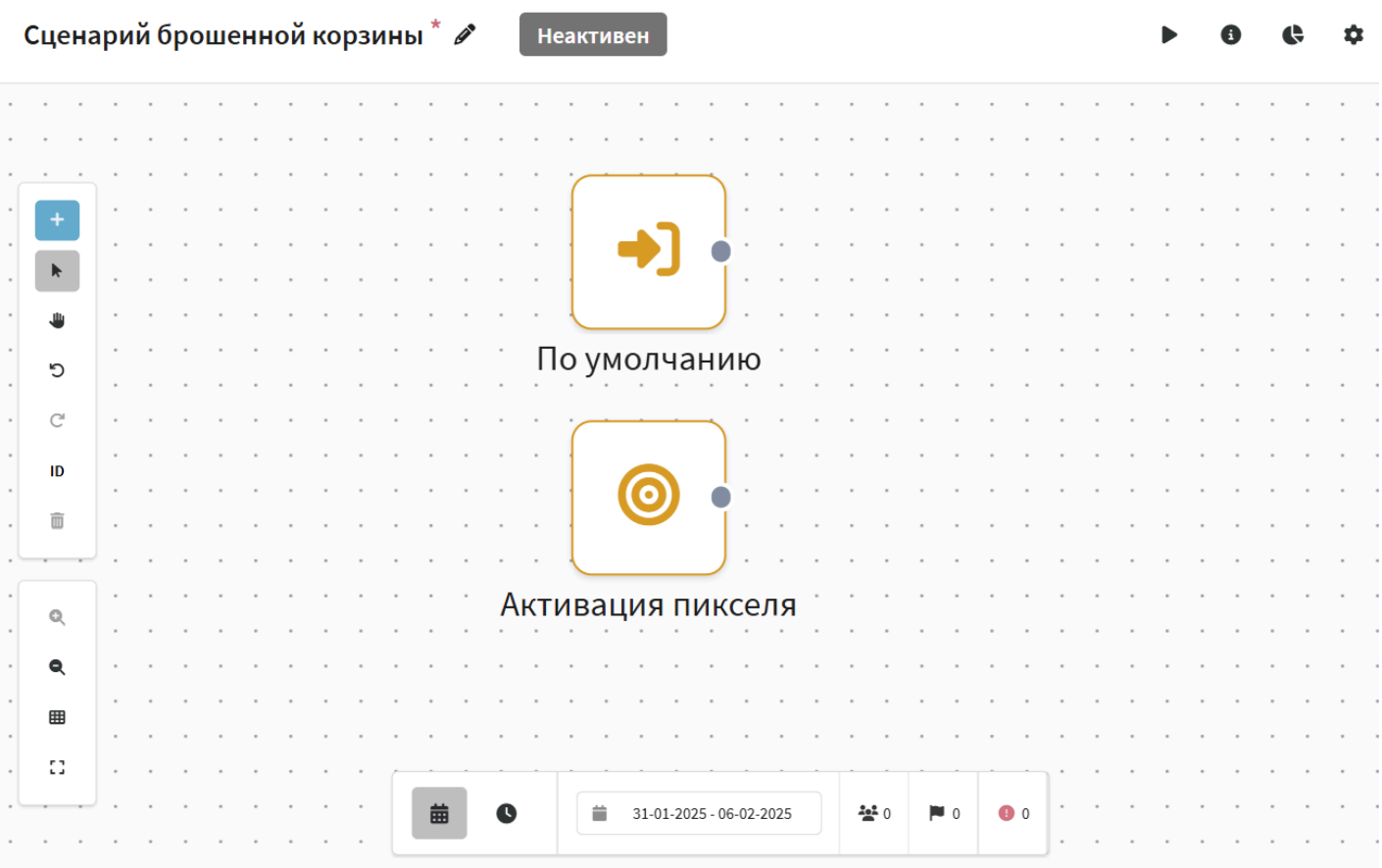
Создайте новый сценарий. Укажите базу — профили этих клиентов будут попадать в сценарий, если выполнится условие захвата. Выберите, что делать с профилем при неоднократном попадании. Нам подойдёт опция **“Отключить повторный вход для задействованных в сценарии профилей клиента”**. Так клиент сможет попасть в него снова, но только когда он соберется совершить следующую покупку спустя какое-то время.



Для того чтобы пользователи попадали в сценарий, добавьте элемент-триггер **“Когда пиксель активирован”**. Активироваться должен тот пиксель, который мы разместили на странице с корзиной (**Online store - Cart**).

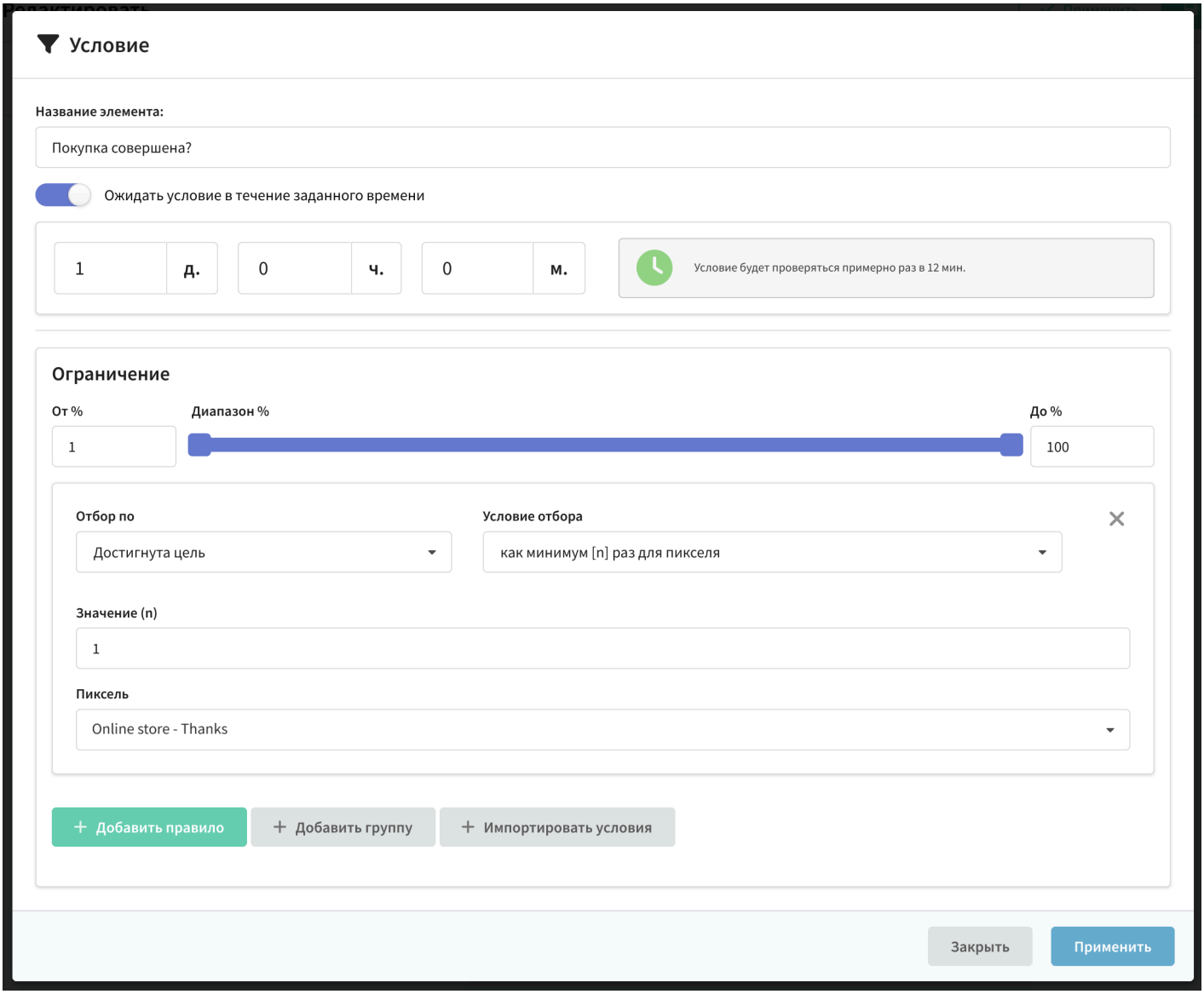


В любом сценарии автоматически создается элемент **“Импорт по умолчанию”**. Он необходим для импорта профилей вручную, через API и при переходе из другого сценария. Это системный элемент, удалить его **невозможно**:

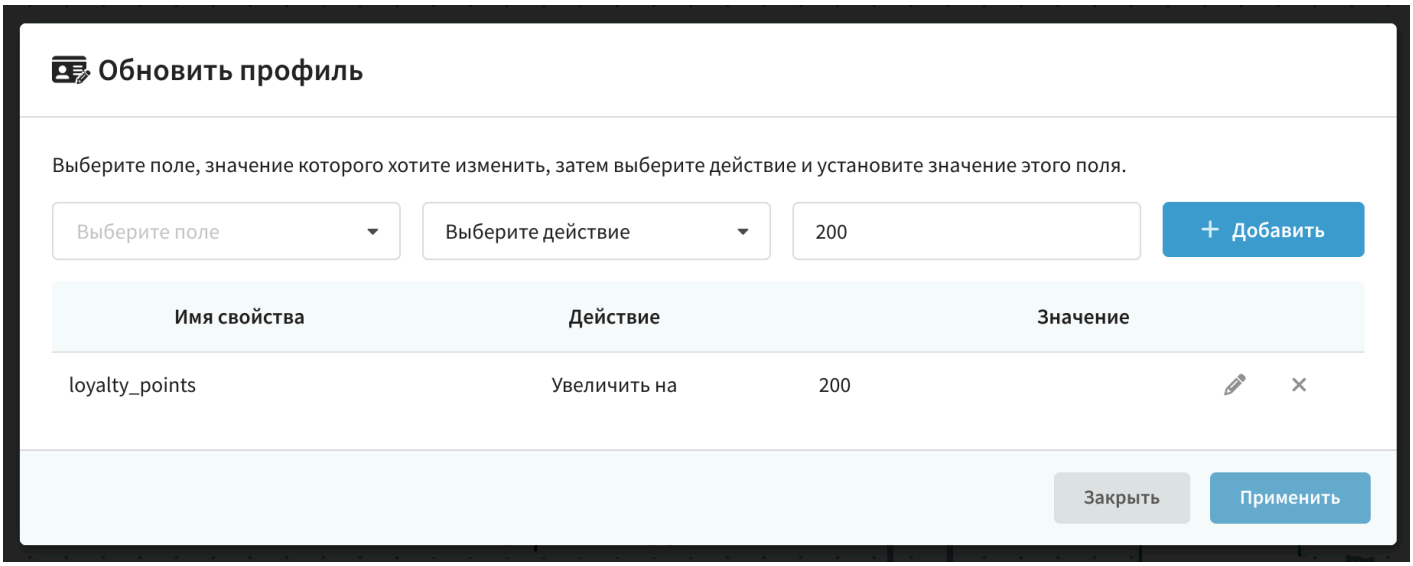


Шаг 3: Настройте цепочку для профилей, оформивших заказ

Далее добавьте элемент-оператор **Условие**, чтобы проверить, совершил пользователь покупку или нет. Условие будет выполнено, если он перешёл на страницу благодарности, т.е. был активирован пиксель **“Online store - Thanks”**. При настройке элемента переключите ползунок **“Ожидать условие в течение заданного времени”** и укажите время ожидания — **1 день**. Платформа в течение этого времени будет проверять, совершил ли клиент покупку:



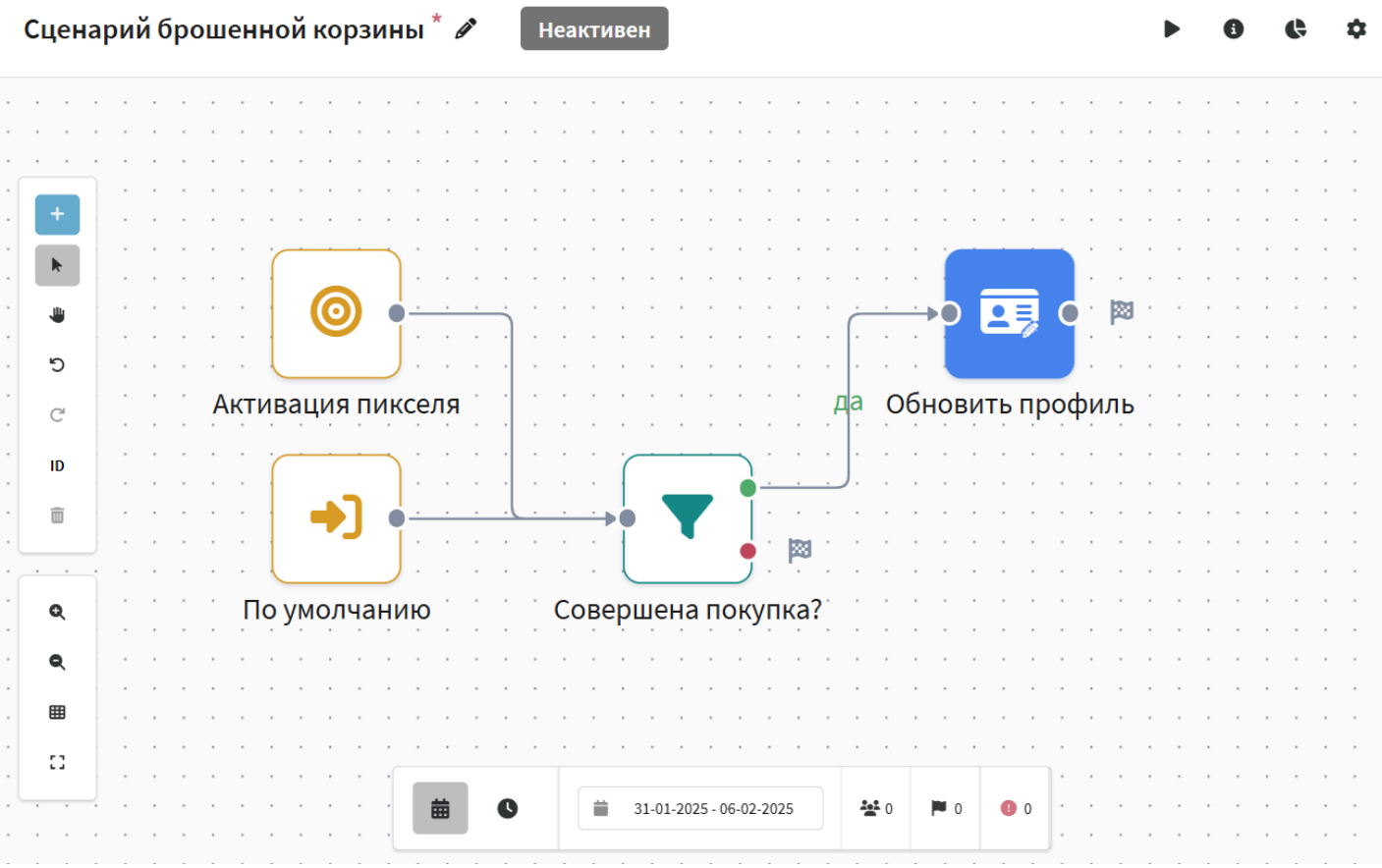
Если клиент совершил покупку, начислите ему бонусные баллы. Для этого в базе профилей должно быть дополнительное числовое поле, например, **loyalty_points**. Добавьте в сценарий элемент **Обновить данные** и увеличьте значение поля:



TIP

Если вы хотите изменять количество бонусных баллов в зависимости от суммы покупки, используйте действие **“Взять значение из поля `{apicontent}`”** вместо “Увеличить на”. Для этого вам надо добавить элемент **Вызов API** на предыдущем шаге и сохранить http-ответ с нужными данными.

Итак, положительное развитие сценария готово:

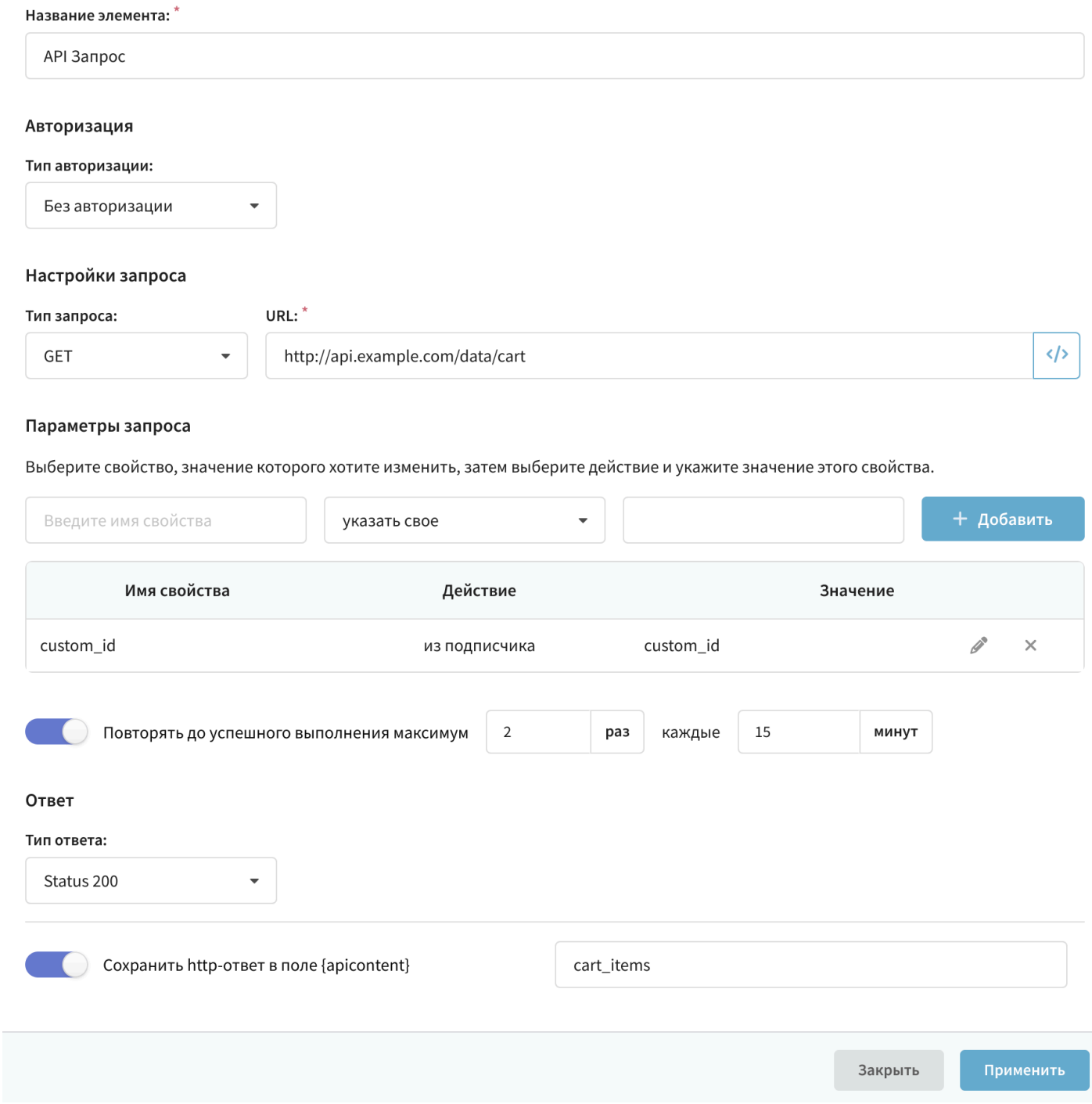


Шаг 4. Настройте цепочку для профилей, покинувших сайт

Вернёмся к клиентам, которые покинули сайт, не завершив оформление заказа. После элемента условия сценарий перенаправит их по красной ветке.

С помощью API-вызова надо передать в платформу данные о товарах, брошенных в корзине, а затем подставить эти данные в шаблон письма. Добавьте элемент **Вызов API**. Укажите URL, по которому хранятся данные о товарах в корзине. Тип запроса — **GET**.

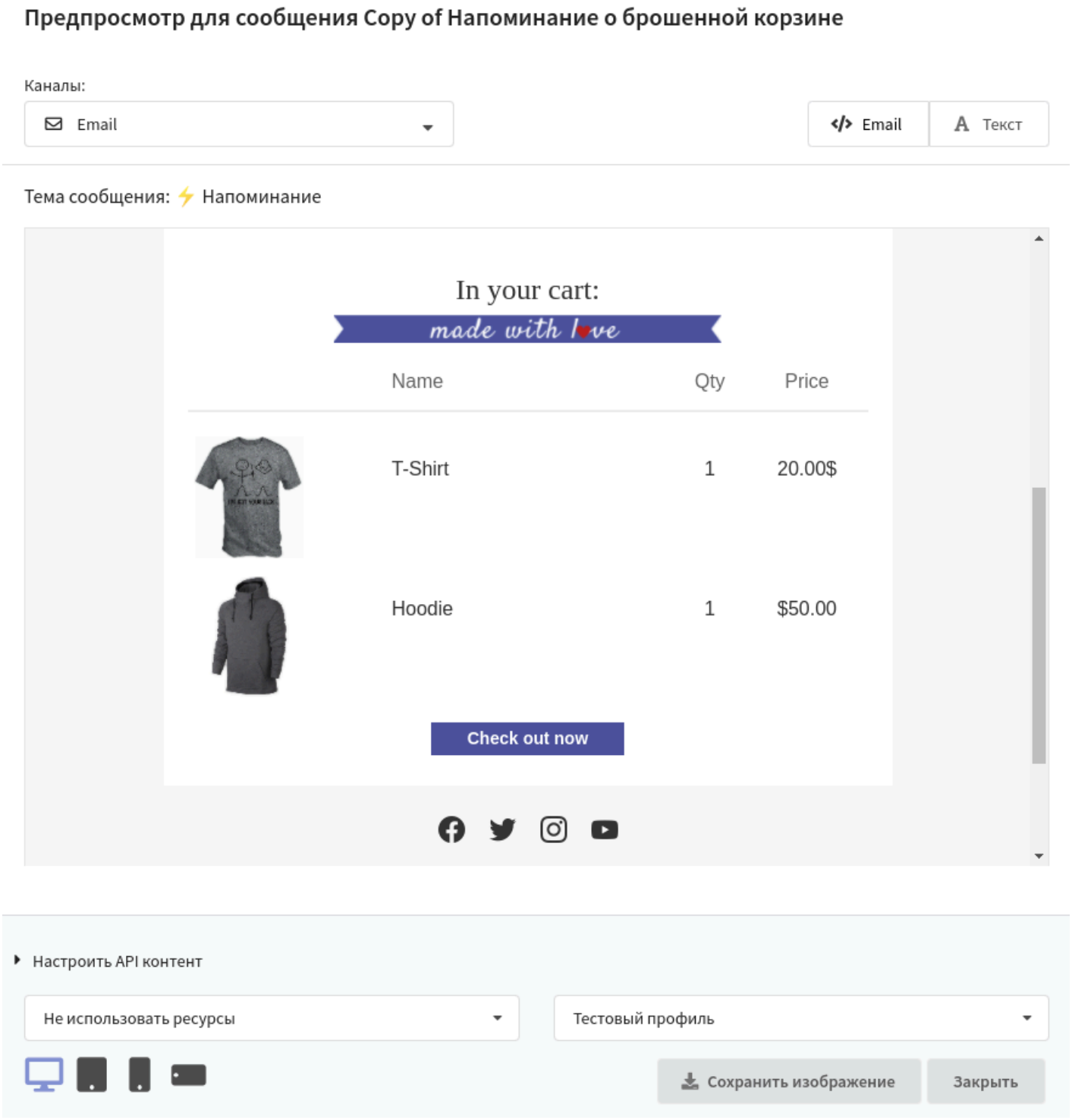
Далее активируйте поле “**Сохранить http-ответ в поле {apicontent}**” и пропишите имя поля, в котором будет сохранен ответ. Название поля может быть произвольным, например, `cart_items`. Именно это поле позже будет использоваться при подстановке переменной в шаблон. Добавьте необходимые параметры запроса, например, **id** или **email профиля**, с помощью которого можно будет найти нужную корзину:



Сохраните сценарий на этом этапе. Осталось создать **шаблон сообщения** и добавить элемент с **рассылкой** в сценарий.

Шаг 5: Создайте шаблон сообщения

Перейдите в раздел **Шаблоны** и создайте новый шаблон для канала отправки email:



Данные в письме должны динамически обновляться в зависимости от получателя. Для этого используется переменная `{apicontent}`. В этой переменной сохраняются данные, которые вы получаете с помощью элемента сценария **Вызов API**. Например, JSON-объект с данными о товарах в корзине покупателя.

Пример JSON-корзины

```
[
  {
    "product_id": 1,
    "product_name": "T-Shirt",
    "product_price": "20.00",
    "product_quantity": 1,
    "product_image_url": "https://example.com/pic/jersey.png"
  },
  {
    "product_id": 2,
    "product_name": "Hoodie",
    "product_price": "50.00",
    "product_quantity": 1,
    "product_image_url": "https://example.com/pic/hoodie.png"
  }
]
```

Сохранив JSON-объект в поле apicontent, вы сможете обращаться к элементам этого объекта по ключу, например — `{apicontent.cart_items.field_name1}`. Чтобы добавить переменную, выберите в списке ({ `</>` }) **Служебные** —> **API-контент**.

Если данные о товарах лежат в массиве, используйте циклы:

```
Товары в корзине:

{for $i $item = apicontent.cart_items}

<p><img src="{$item.product_image_url}"/></p>

<p>Название: {$item.product_name}</p>
<p>Количество: {$item.product_quantity}</p>
<p>Стоимость: {$item.product_price}</p>

{end}

<!--{unsubscribe} -->
```

HTML TEXT

B I S utm Визуальный редактор β

```
Товары в корзине:

{for $i $item = apicontent.cart_items}

<p><img src="{$item.product_image_url}"/></p>

<p>Название: {$item.product_name}</p>
<p>Количество: {$item.product_quantity}</p>
<p>Стоимость: {$item.product_price}</p>

{end}

<!--{unsubscribe} -->
```

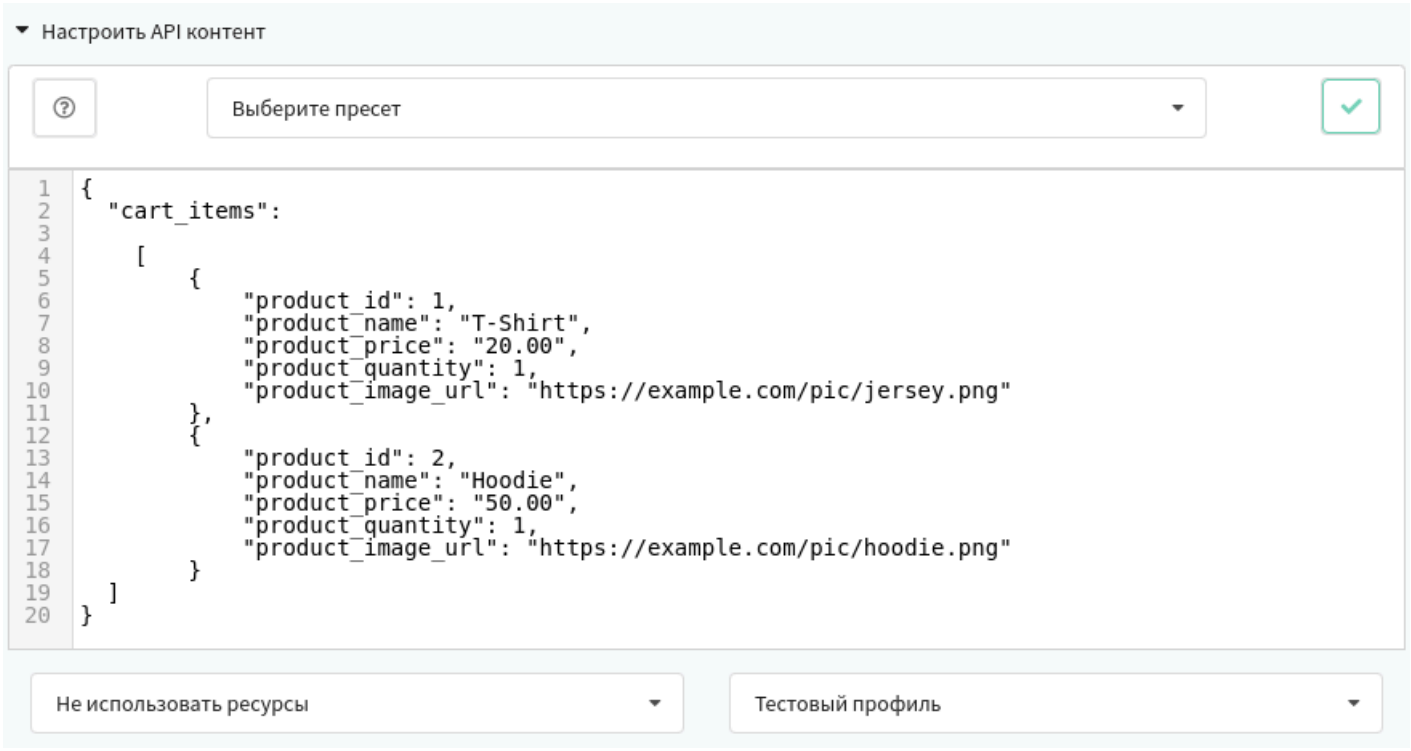
Шаг 6: Проверьте отображение переменной apicontent

При добавлении переменной apicontent предпросмотр сообщения не будет работать, так как в переменной ничего нет. Чтобы посмотреть, как будет выглядеть сообщение, необходимо дополнительно настроить окно предпросмотра.

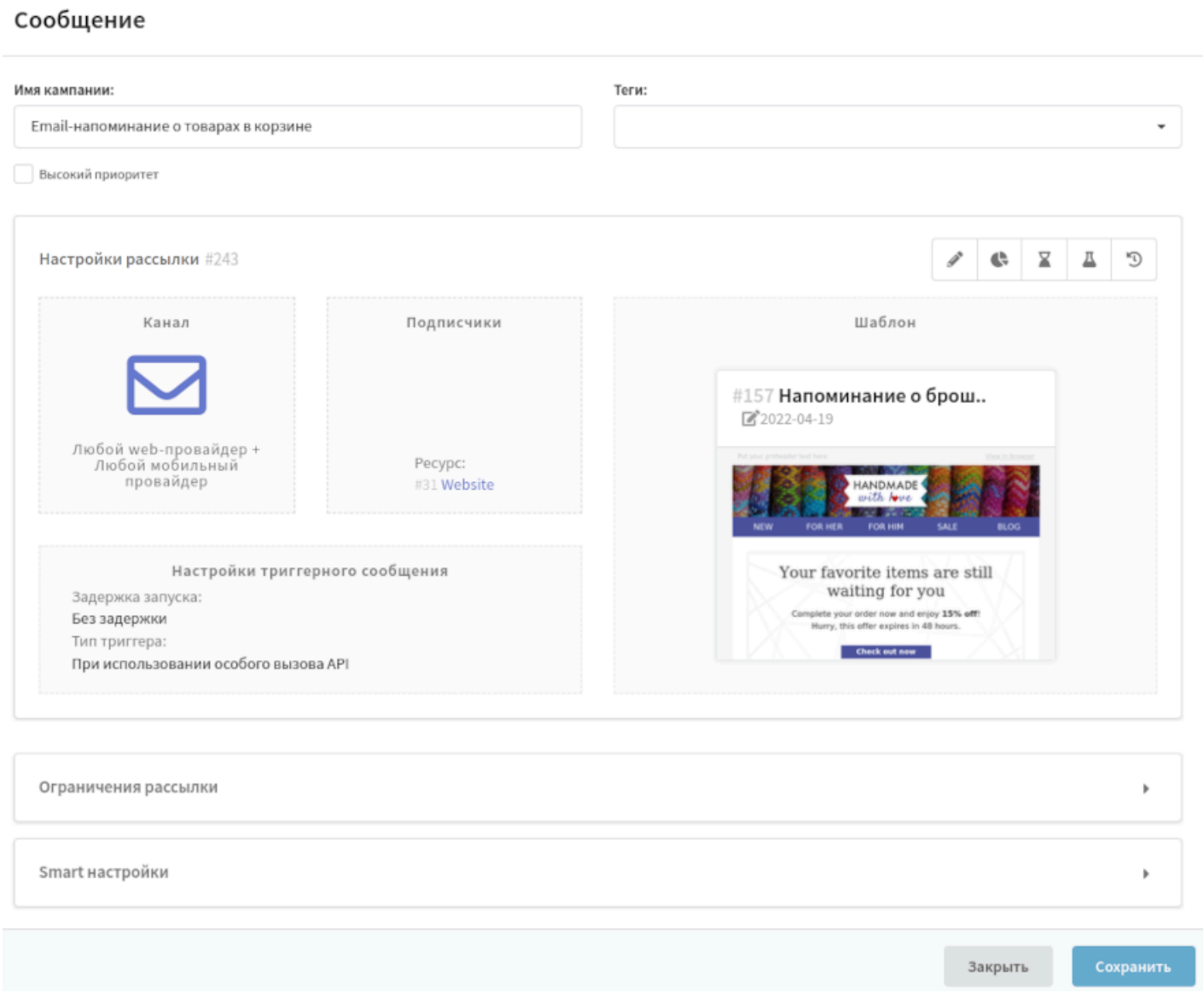
Откройте предпросмотр

и внизу найдите блок “**Настроить API-контент**”. Добавьте JSON-объект, аналогичный тому, что вы получите после API-запроса в сценарии.

Обратите внимание, тестовый массив товаров должен храниться в поле `cart_items`. Нажмите на галочку. Значения полей подставятся в шаблон:



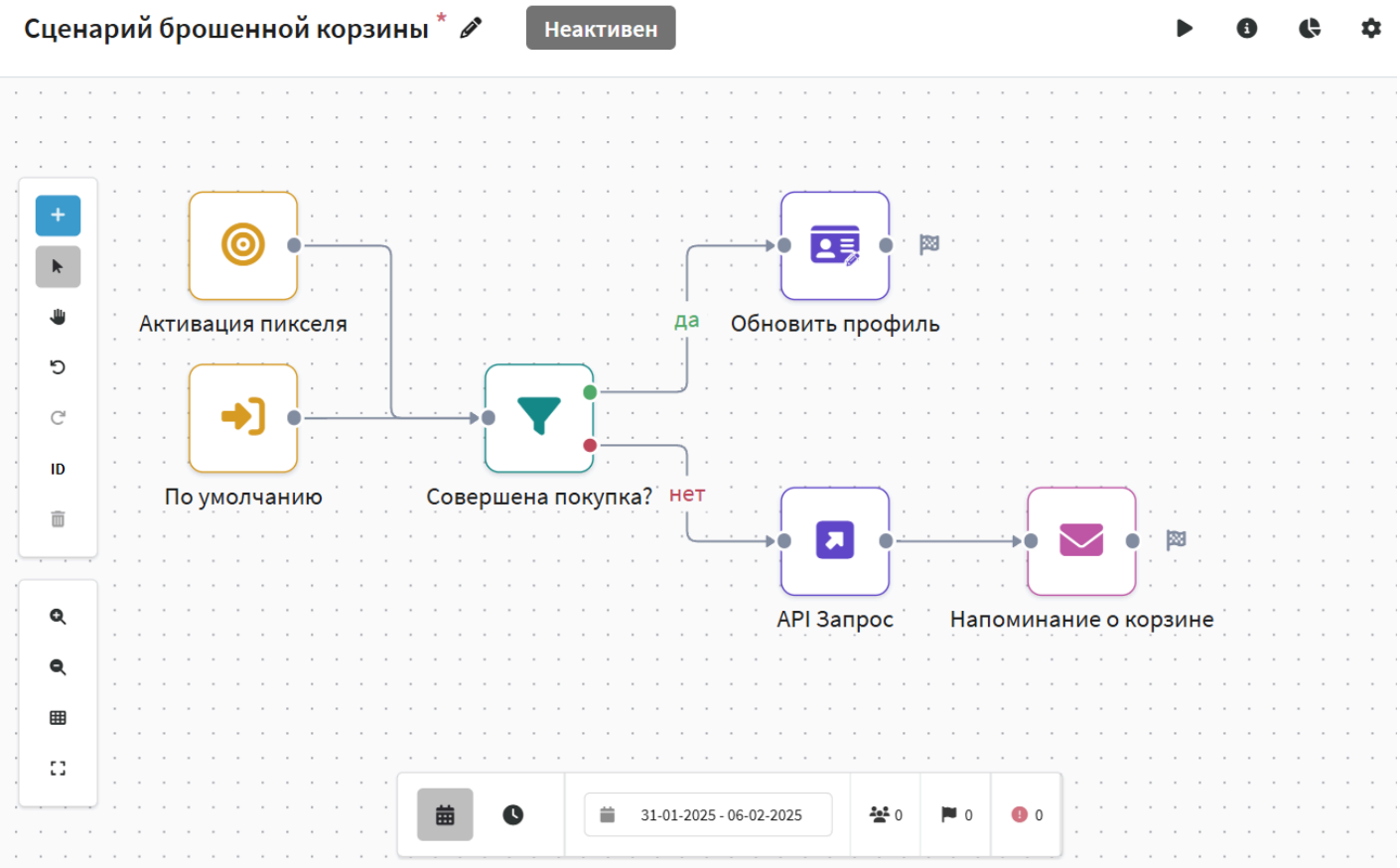
Вернитесь в сценарий и добавьте элемент **Email** из категории **“Каналы”**. В настройках выберите email-канал, ресурс подписчиков и созданный шаблон:



Сохраните сценарий, нажав на соответствующую кнопку в верхней части экрана.

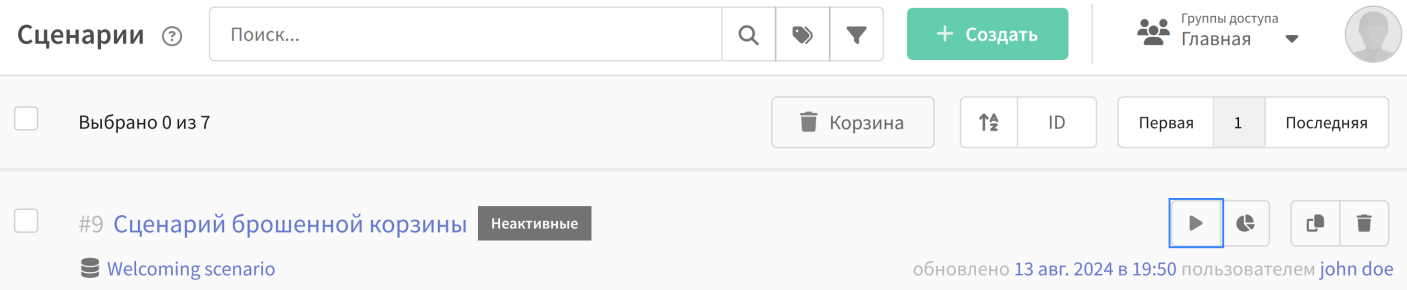
Шаг 7: Активируйте сценарий и оцените эффективность

Готовый сценарий выглядит следующим образом:



При желании сценарий можно расширить. Например, если подписчик не завершил покупку, назначьте ему промокод на скидку или отправьте подборку сопутствующих товаров.

После сохранения **активируйте** сценарий в меню быстрых действий:



Оценить эффективность сценария вы можете нажав на соответствующий значок в меню быстрых действий или в [отчёте по каналам](#).

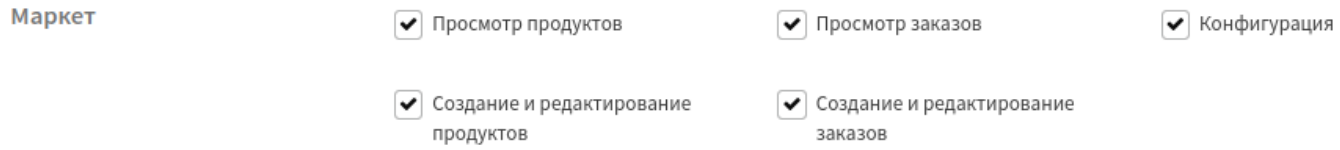
Маркет

Маркет позволяет передавать в платформу данные о товарах и заказах и использовать их в коммуникациях с клиентами. Данные могут передаваться автоматически — через API или товарный фид, а также вручную — путём загрузки файла.

Полученные данные можно использовать для:

- подстановки информации о заказе в шаблоны сообщений;
- запуска триггеров при изменении статуса заказа;
- сегментации клиентов на основе данных о покупках.

Прежде чем начать работу, убедитесь, что у вас включены необходимые роли для пользователя и токена:



Для работы с маркетом мы рекомендуем использовать API-запросы. Если вы работаете в интерфейсе платформы, следуйте инструкциям ниже:

Настройки маркета

	В данной инструкции описано, как настроить такие сущности маркета, как канал продаж, дополнительные поля, статусы позиций заказа, регионы, категории и валюты. Чтобы начать работу, выберите в главном меню раздел "**Маркет**" —> " **Настройки**".

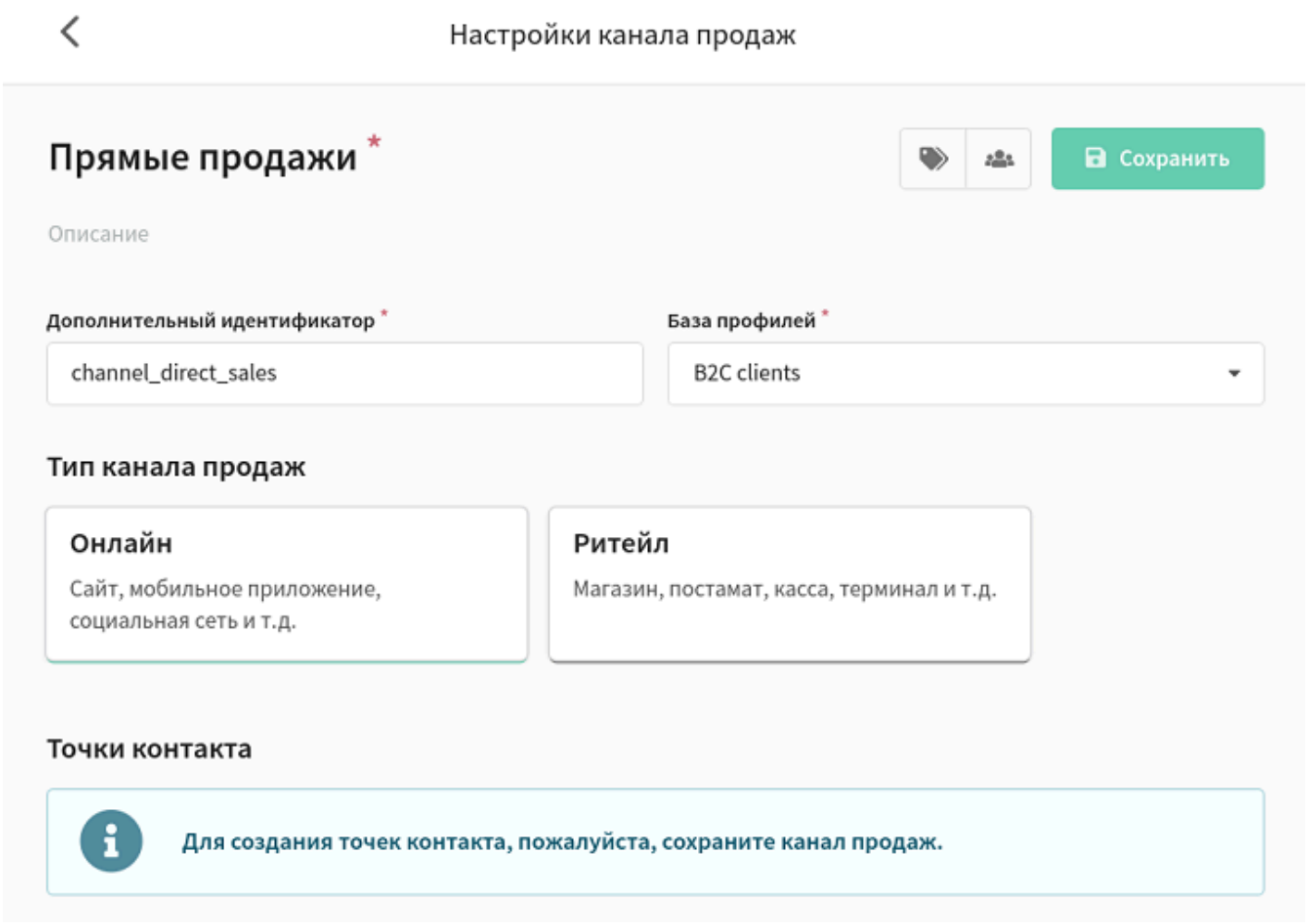
Каналы продаж

Канал продаж – это канал, через который клиенты попадают в компанию, чтобы заказать интересующие их товары. Он объединяет площадки и инструменты, используемые для продажи (магазин, сайт, терминал или касса, где был оформлен заказ). Главная задача канала – определить базу покупателей и точки контакта.

Чтобы создать новый канал продаж, кликните “**Создать**”. В появившемся окне настройте канал:

- введите имя и описание канала
- дополнительный идентификатор
- добавьте теги

  и группы доступа
- укажите базу профилей (покупатели канала)
- выберите тип канала (онлайн или ритейл)



CAUTION

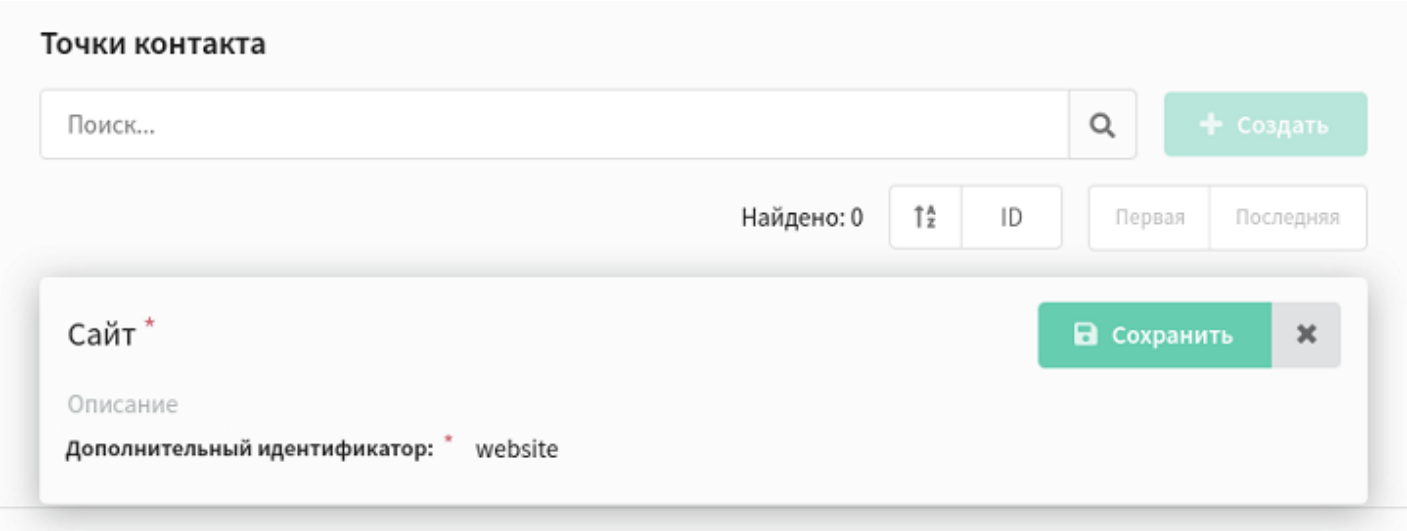
После создания канала продаж нельзя изменить заданную базу профилей и дополнительный идентификатор.

Теперь сохраните канал, а затем выберите его в общем списке, чтобы создать точку контакта.

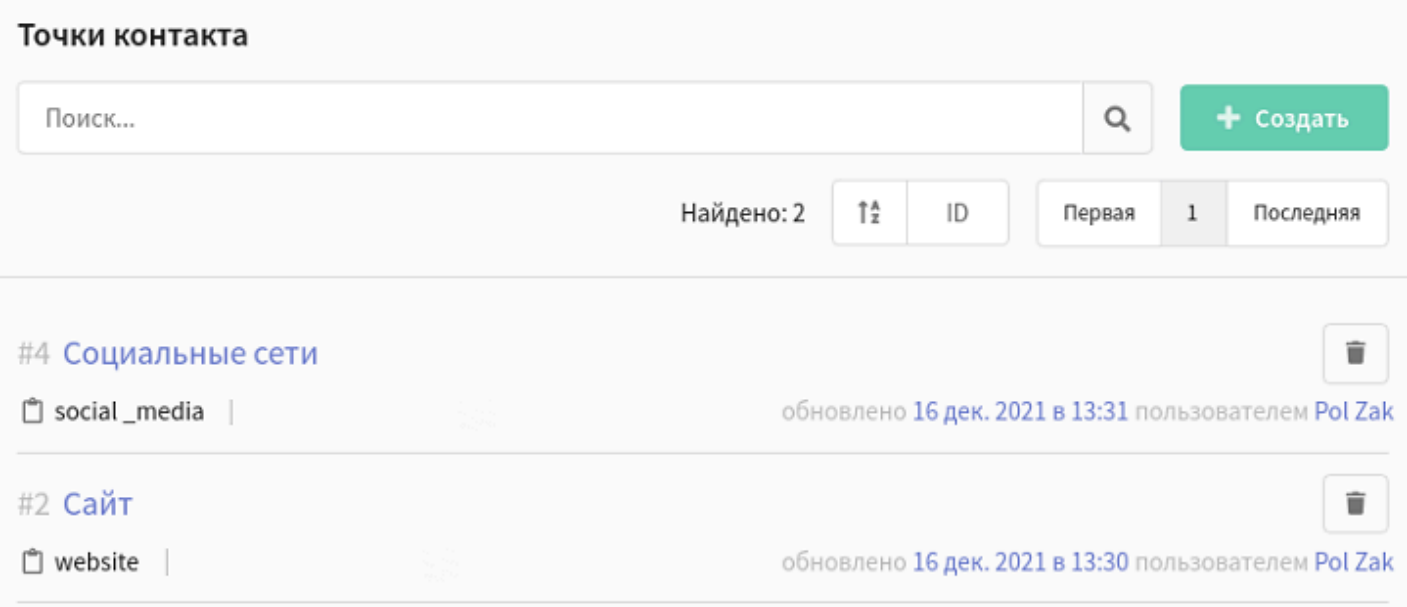
TIP

Точка контакта — это определенная площадка или инструмент, с помощью которого вы продаете клиенту продукт. Точка контакта связывает канал продаж и заказ. Это может быть сайт, магазин, постамат, касса, где оформлен заказ, отдельный терминал, соцсеть, call-центр и т.д.

Введите название точки, описание и дополнительный идентификатор (используется в API-вызовах). Дополнительный идентификатор **нельзя будет изменить** после сохранения точки. Затем нажмите “**Сохранить**”:



Каждая точка может быть привязана только к одному каналу. При этом у канала может быть несколько точек контакта:



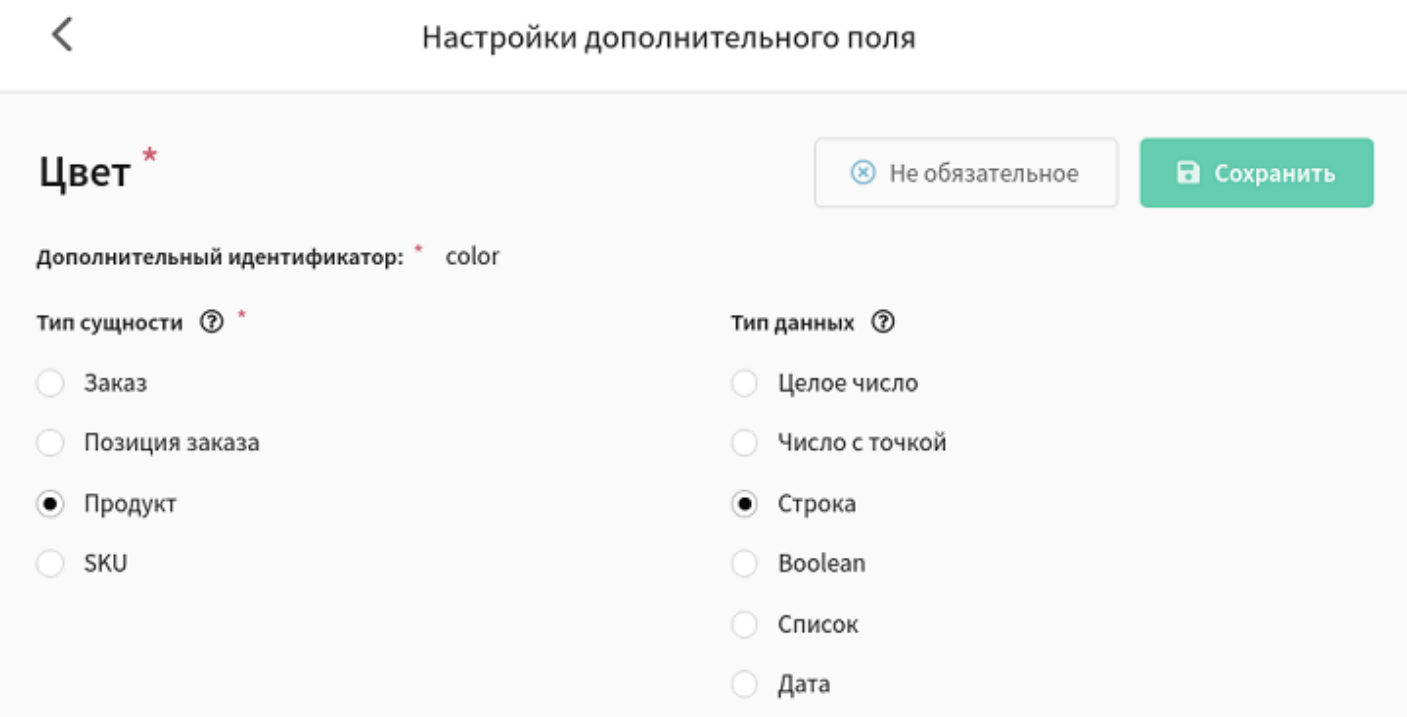
Дополнительные поля

Дополнительные поля задаются для таких сущностей, как заказ, позиция заказа, продукт и SKU. В них можно хранить информацию о важных характеристиках товара — цвет, размер, объем, материал, модель, вес и т. д.

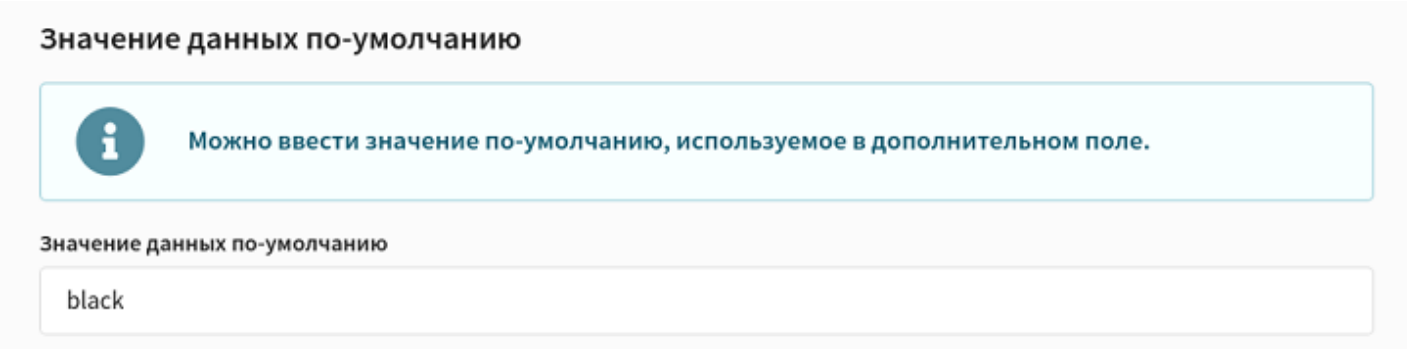
Введите имя и дополнительный идентификатор поля, который будет использовать в API-запросах. Дополнительный идентификатор **нельзя будет изменить** после сохранения.

Выберите **тип сущности** и укажите **тип данных**, которые будут храниться в новом поле:

- целое число
- число с точкой
- строка
- boolean
- список
- дата



Есть возможность задать значение данных по умолчанию – значение будет автоматически подставляться в дополнительное поле указанной сущности.

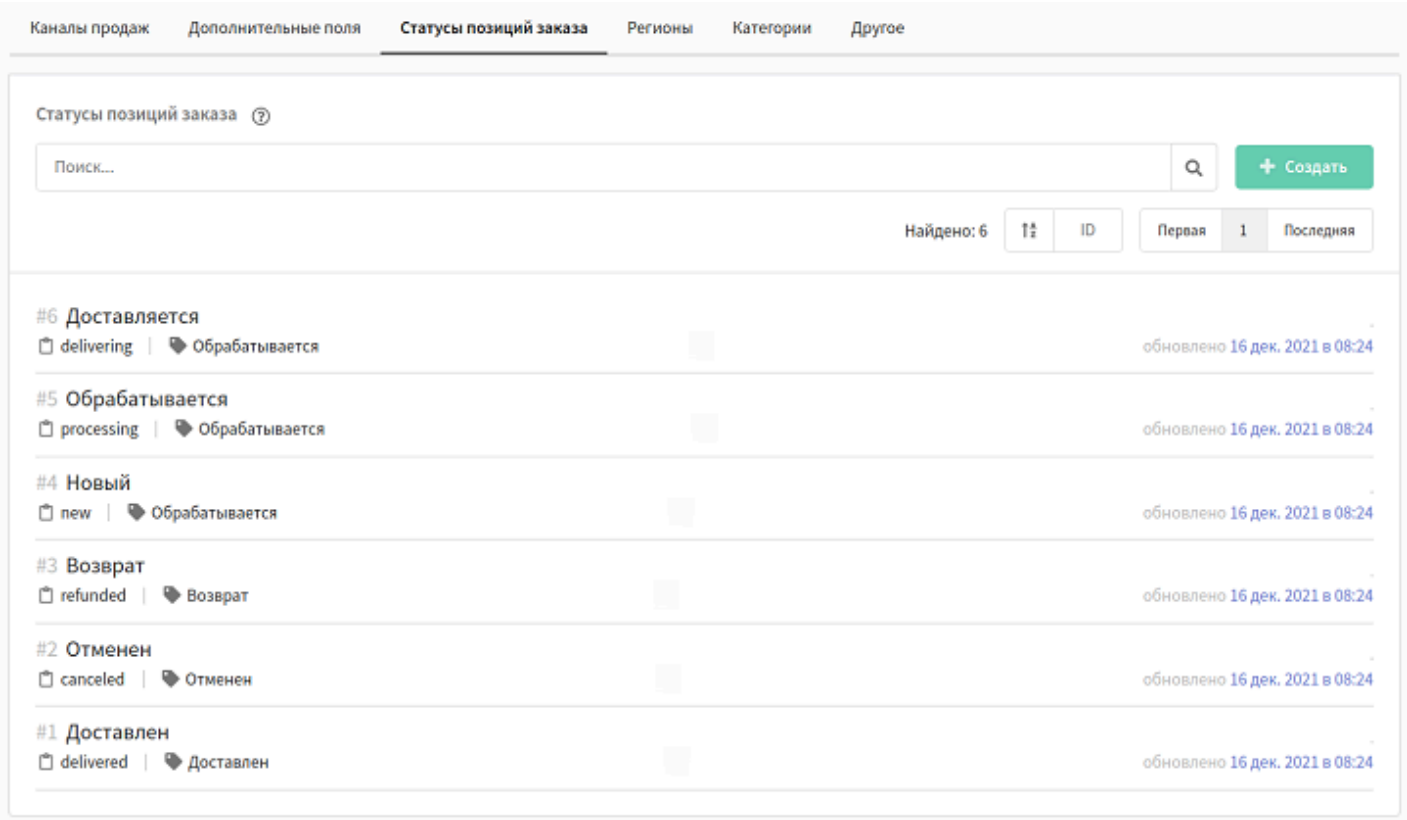


Если необходимо сделать поле обязательным для заполнения, переключите кнопку “**Не обязательное**” —> “**Обязательное**”:

Статусы позиций заказа

По умолчанию доступно 6 статусов позиций:

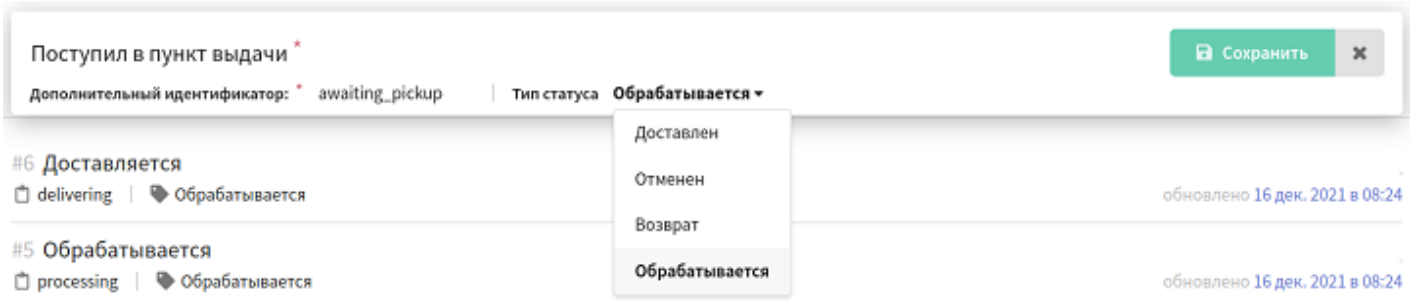
- доставляется
- обрабатывается
- новый
- возврат
- отменен
- доставлен



Также можно создать свои статусы. Например, если вам необходимо разделить процесс доставки на несколько этапов: *“Получен курьерской службой”, “Поступил в пункт выдачи”, “Передан курьеру”.*

Чтобы создать новый статус, воспользуйтесь кнопкой “**Создать**”. Затем введите имя статуса, его идентификатор и тип. Доступно 4 типа статуса:

- доставлен
- отменен
- возврат
- обрабатывается



CAUTION

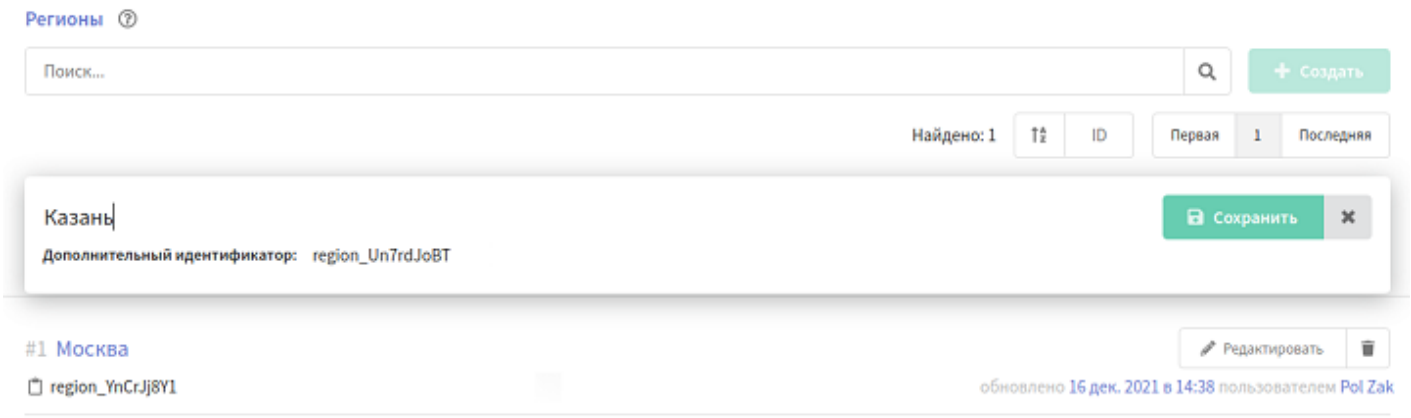
После сохранения статуса дополнительный идентификатор нельзя будет изменить.

После заполнения всех полей нажмите “**Сохранить**”.

Регионы

Регионы указываются при настройке заказа. В продуктах и SKU для каждого региона можно задать региональные данные — информация о товаре, которая отличается в зависимости от региона (наличие, цена, возможность доставки и др.).

При создании региона введите его название и при необходимости отредактируйте идентификатор, который будет передаваться в API-запросах. Затем нажмите “**Сохранить**”:



TIP

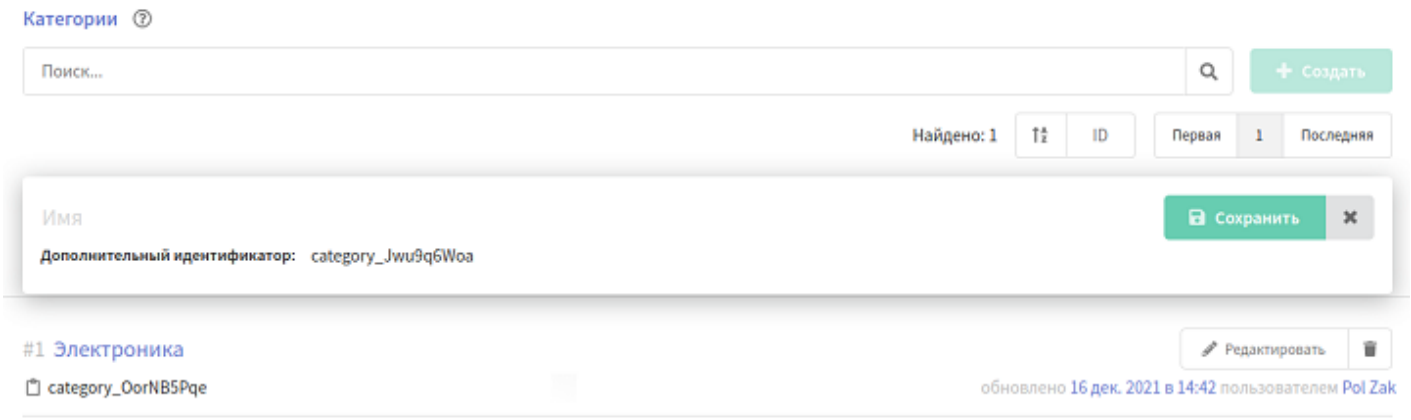
Например, вы добавляете в платформу продукт “Canon EOS”. При настройке продукта вы указываете цену 15 000 RUB и добавляете для продукта несколько регионов “Москва”, “Казань”, “Екатеринбург”, “Новосибирск” и др. Затем для каждого региона уточняете региональные данные, а именно — повышаете цену продукта для регионов “Новосибирск” и “Екатеринбург” до 16 500 RUB.

После этого вы создаете заказ. Для любого заказа можно указать регион. Цена товара изменится в зависимости от настроек региональных данных. Если вы укажете в заказе регион “Новосибирск”, то цена продукта автоматически изменится на 16 500 RUB. Если же вы выберете регион “Казань”, то цена останется 15 000 RUB.

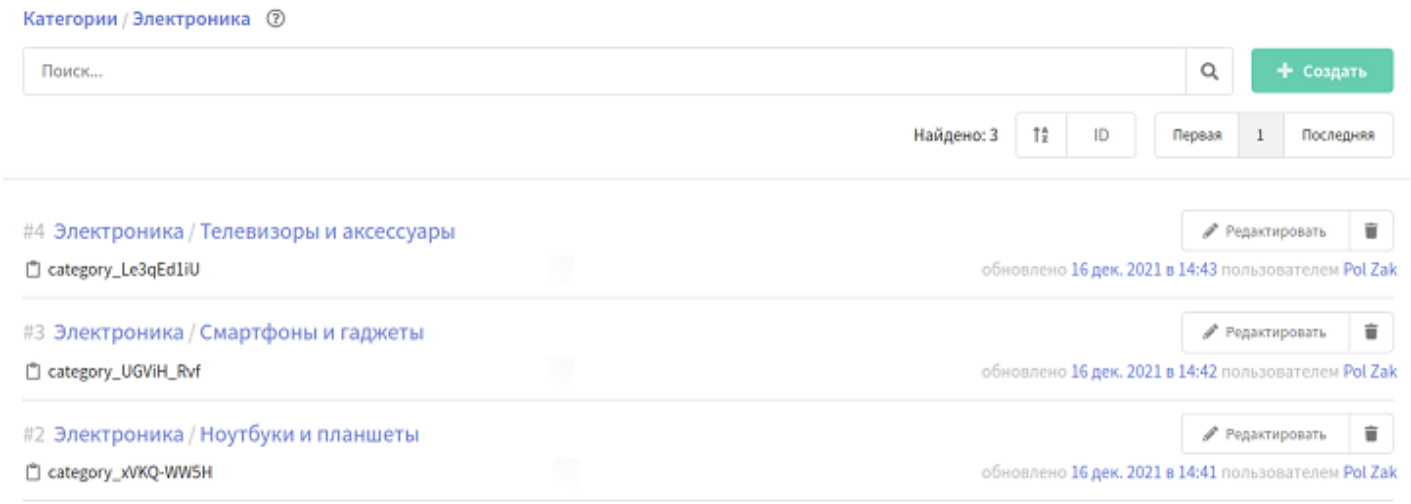
Категории

Категория — это группа товаров, которые покупатель воспринимает как схожие между собой. В категории можно объединять продукты и SKU.

Нажмите “**Создать**”, чтобы создать новую категорию. Введите название категории и при необходимости измените идентификатор. Затем сохраните категорию:



У каждой категории может быть несколько подкатегорий. Нажмите на название категории, внутри которой хотите создать подкатегорию. Создайте и сохраните объект:

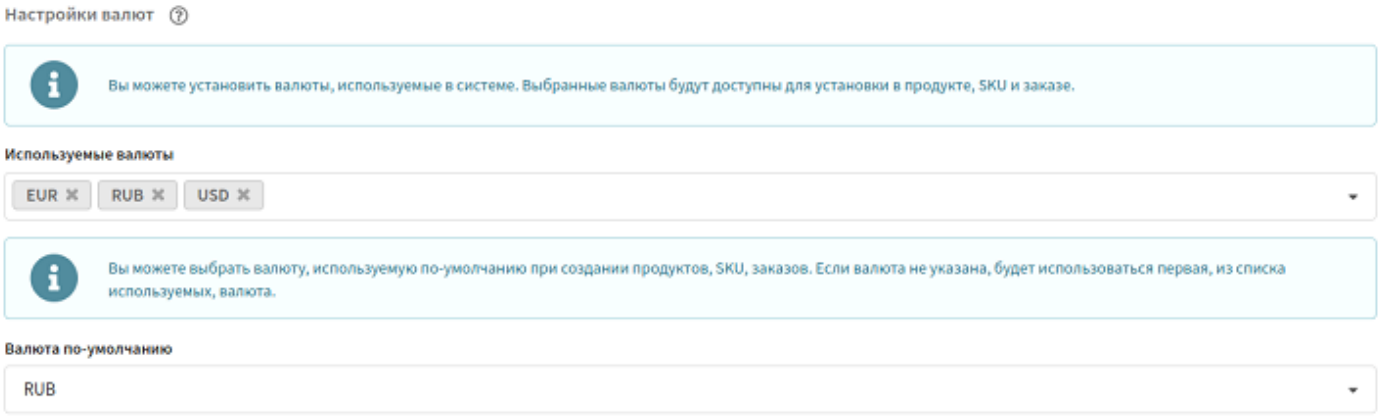


Другое

На этой вкладке настраиваются валюты, используемые в маркете. Выберите необходимые валюты из выпадающего списка.

CAUTION

При импорте из файла или через API продукт или SKU не будет импортирован, если для него указана валюта, которая не задана в поле “**Используемые валюты**”.



Можно выбрать **валюту по умолчанию** — она будет автоматически устанавливаться при создании продуктов, SKU и заказов.

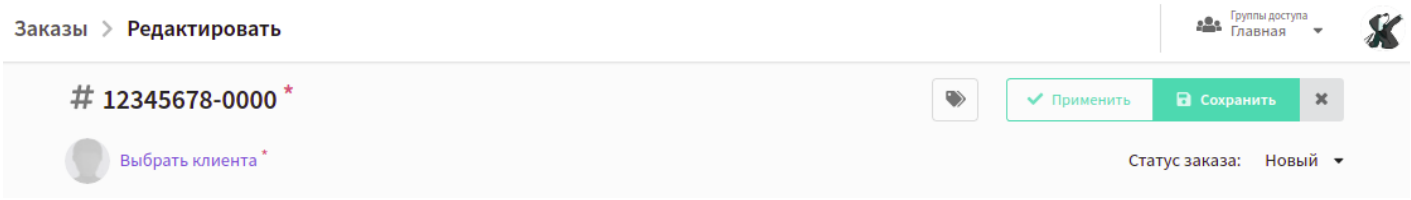
Заказы

Раздел **"Заказы"** позволяет создавать и обрабатывать заказы, поступающие из различных точек продаж. Здесь хранится информация по уже оформленным заказам, включая их текущие статусы. Для создания нового заказа необходимо выбрать клиента и добавить в заказ продукты или SKU.

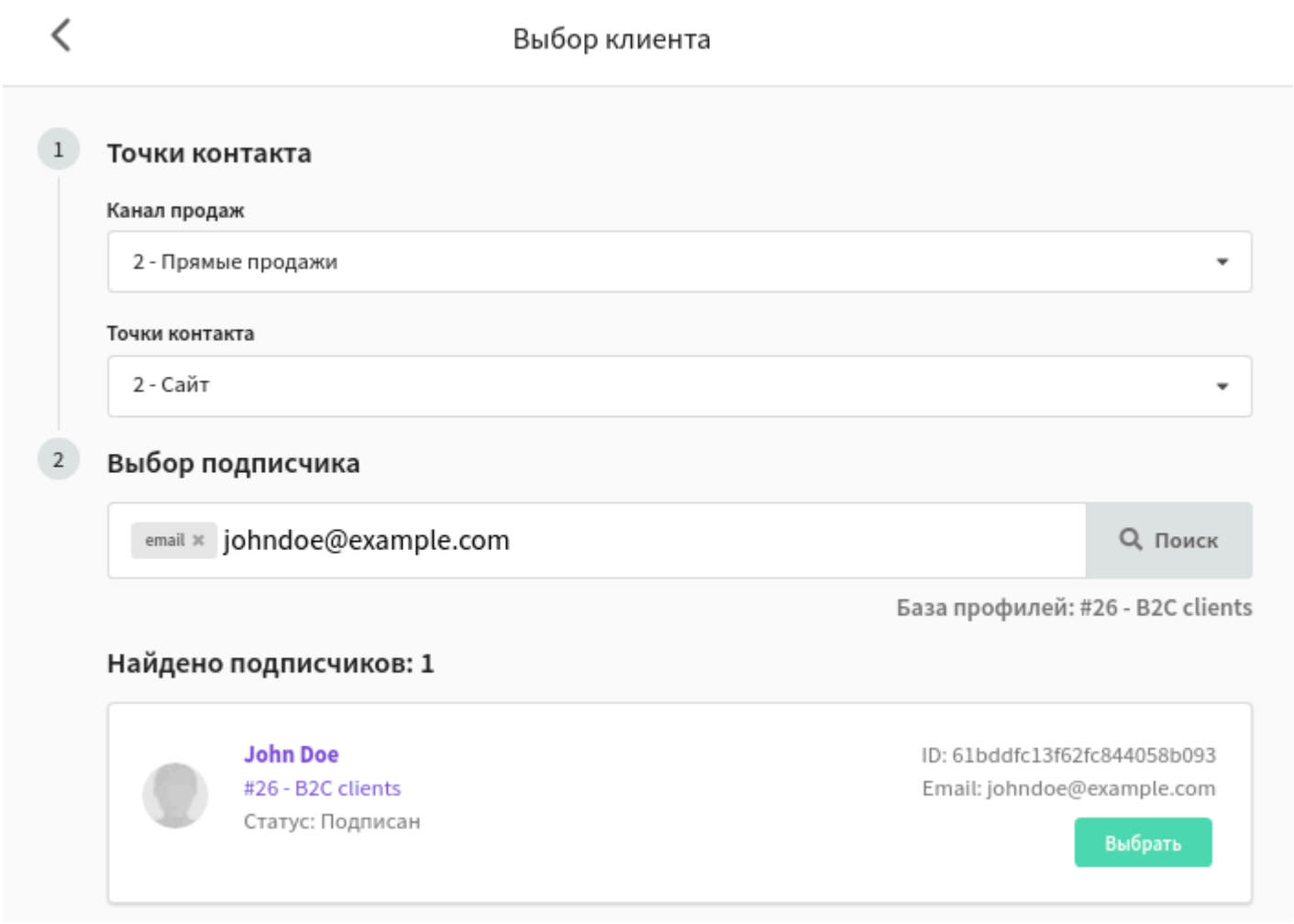
Заказ можно импортировать через API или создать вручную в интерфейсе платформы. Чтобы начать работу с заказами, перейдите в раздел **"Маркет" —> "Заказы"**.

Выбор клиента

Для создания заказа вручную введите номер заказа. Затем укажите покупателя, кликнув “**Выбрать клиента**”:



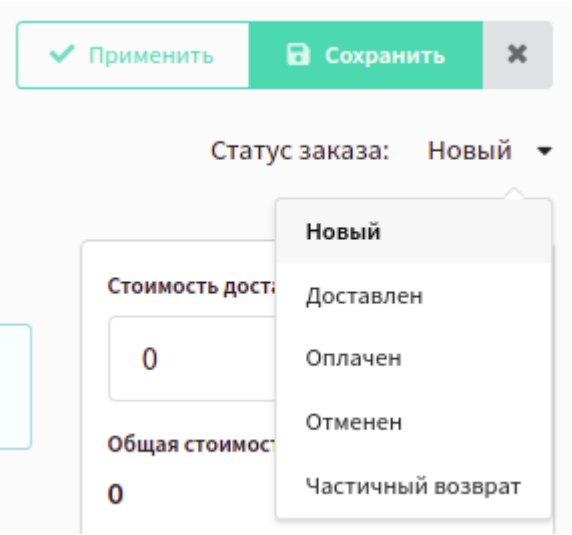
В новом окне выберите канал продаж и точку контакта. К каждому каналу привязывается своя база профилей. Используйте поле “**Выбор подписчика**”, чтобы найти в базе нужный профиль по email, номеру телефона, кастомному полю или с помощью любого другого доступного идентификатора. Окно автоматически закроется, когда вы нажмете “**Выбрать**”:



Текущий статус заказа

Доступные статусы заказа:

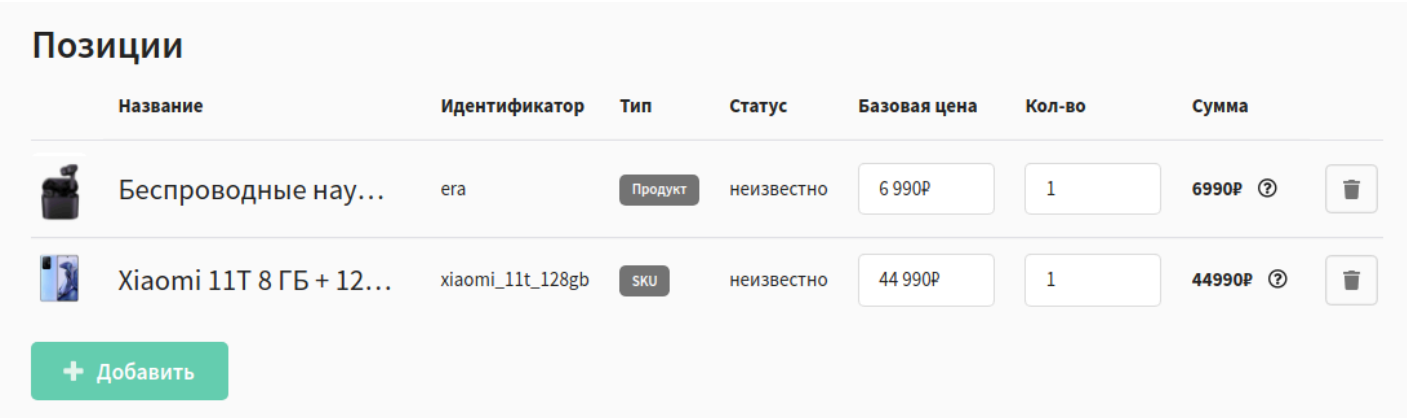
- новый
- доставлен
- оплачен
- отменен
- частичный возврат



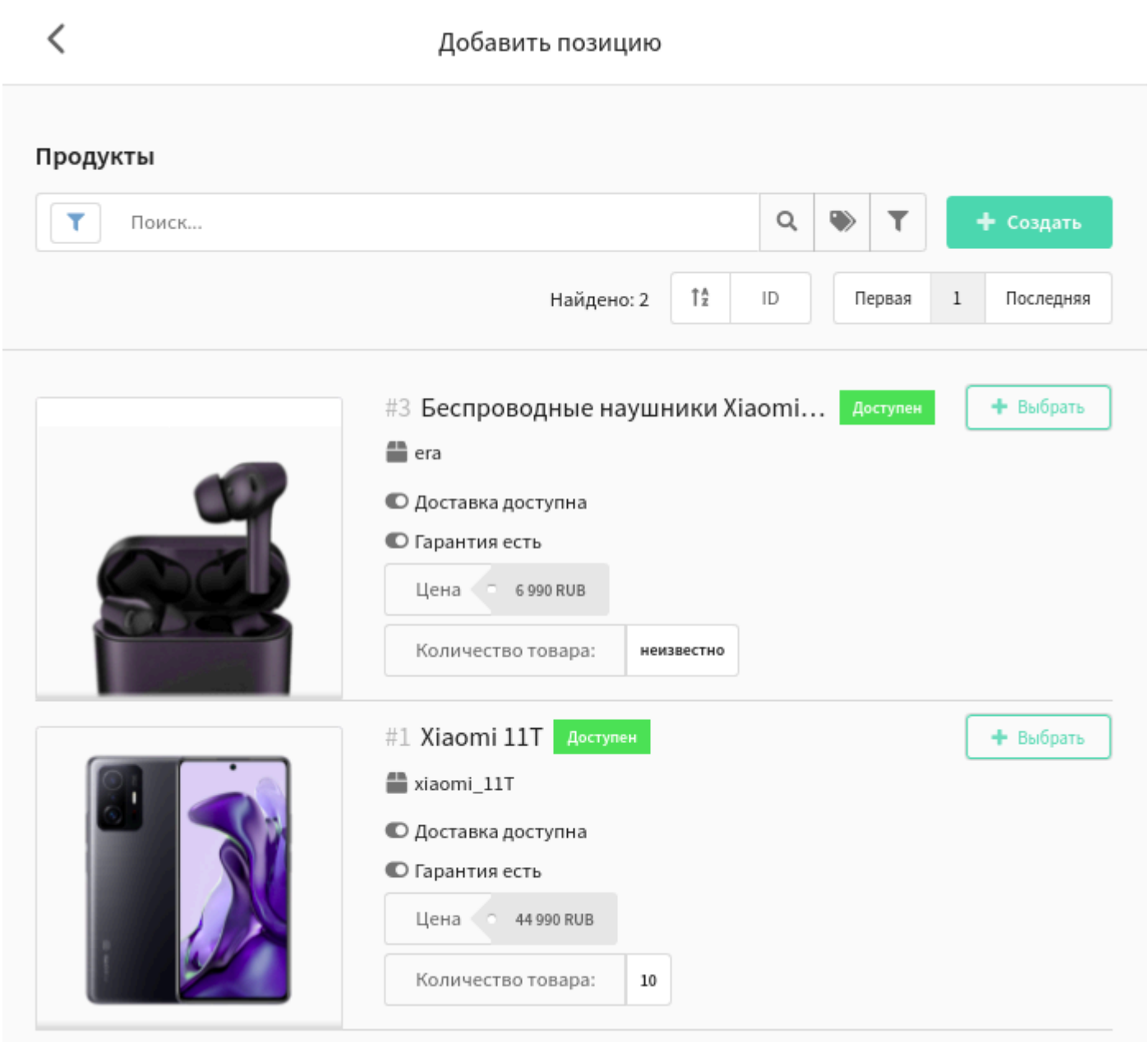
Добавление позиции в заказ

Заказ состоит из позиций. Позициями могут быть продукты или SKU. Несколько единиц одного товара в заказе считаются за одну позицию.

Чтобы добавить новые позиции в заказ, нажмите “**Добавить**”:



В новом окне выберите необходимые позиции:

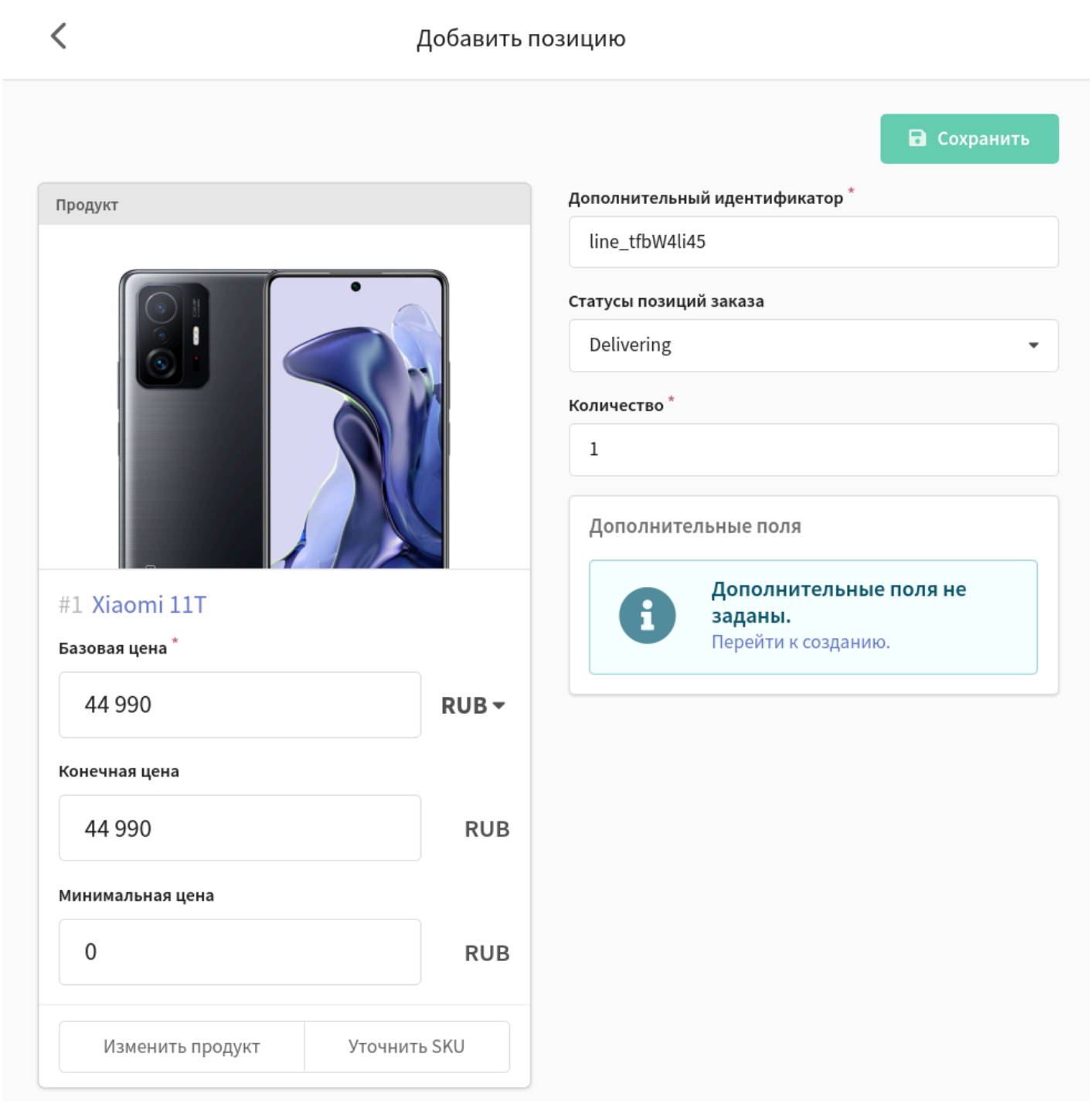


Настройка позиции

При необходимости вы можете отредактировать каждую позицию. Для этого кликните на позицию в общем списке – справа появится окно для редактирования следующих полей:

- цена товара и дополнительный идентификатор;
- текущий статус товара;
- количество товаров в позиции;
- дополнительные поля.

В этом же окне вы можете **изменить продукт** или **уточнить SKU**.



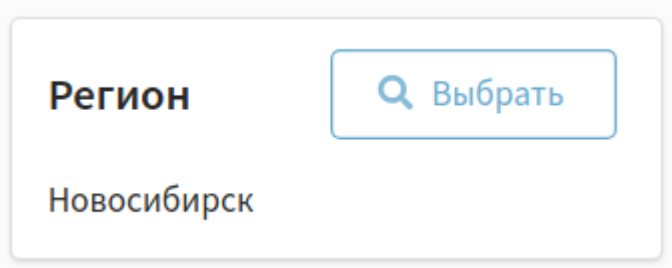
Стоимость заказа

Общая стоимость заказа показана в окне справа. Она складывается из стоимости доставки и стоимости каждой позиции товара.

Если вам необходимо пересчитать стоимость заказа, активируйте опцию “**Авто-расчет**”.

Регион

Выберите регион, где был оформлен заказ. Если вы уточняли региональные данные для продукта или SKU, то в зависимости от региона могут измениться некоторые характеристики товара (цена, значения дополнительных полей, наличие на складе).

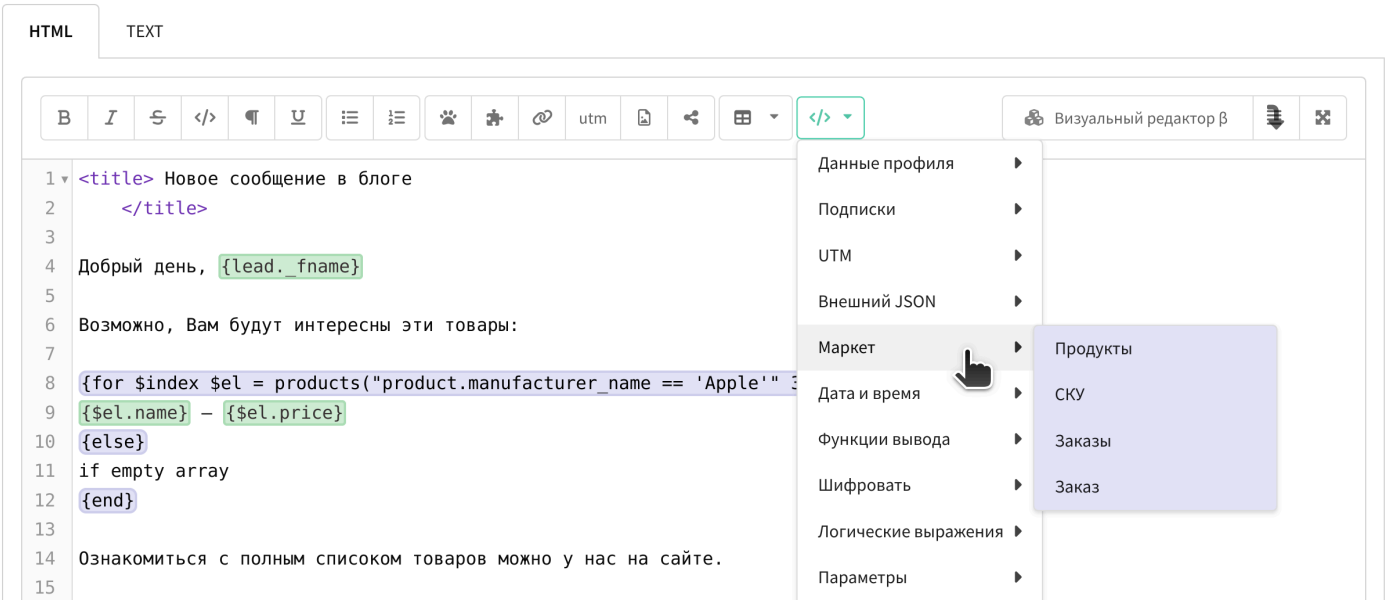


Настройка заказа завершена. Сохраните его, воспользовавшись кнопками в верхней панели.

Переменные маркета в шаблонах

Чтобы использовать информацию о заказах и товарах в рассылках, необходимо добавить переменные маркета в шаблон сообщения.

Откройте шаблон сообщения и нажмите кнопку `</>` в редакторе. В списке выберите “**Маркет**” и вставьте одну из переменных:



TIP

Подробнее об использовании переменных маркета в сообщениях читайте в этом разделе.

Руководство: как отправить письмо подтверждения заказа

Этот гайд покажет алгоритм работы в маркете Altcraft Platform.

В качестве примера представим, что у нас есть интернет-магазин смартфонов. Покупатель заходит на сайт, добавляет товар в корзину и оформляет заказ. Наша задача — отправить ему письмо подверждения заказа, где будет указан номер заказа и сумма к оплате.

В инструкции описаны следующие шаги:

1. Создание канала продаж и точек контакта
2. Импорт данных о продуктах и SKU с помощью yml файла
3. Настройка API-запроса для добавления данных заказа в маркет
4. Создание шаблона и подстановка данных заказа с помощью переменных
5. Настройка триггерной рассылки

Шаг 1. Настройка канала продаж

Работа в маркете начинается с создания канала продаж. Канал определяет базу профилей и точки контакта.

База профилей — это покупатели, которые делают заказы в определенном канале. Необходимо импортировать их данные в новую или существующую базу. Для этого воспользуйтесь импортом из файла или через API.

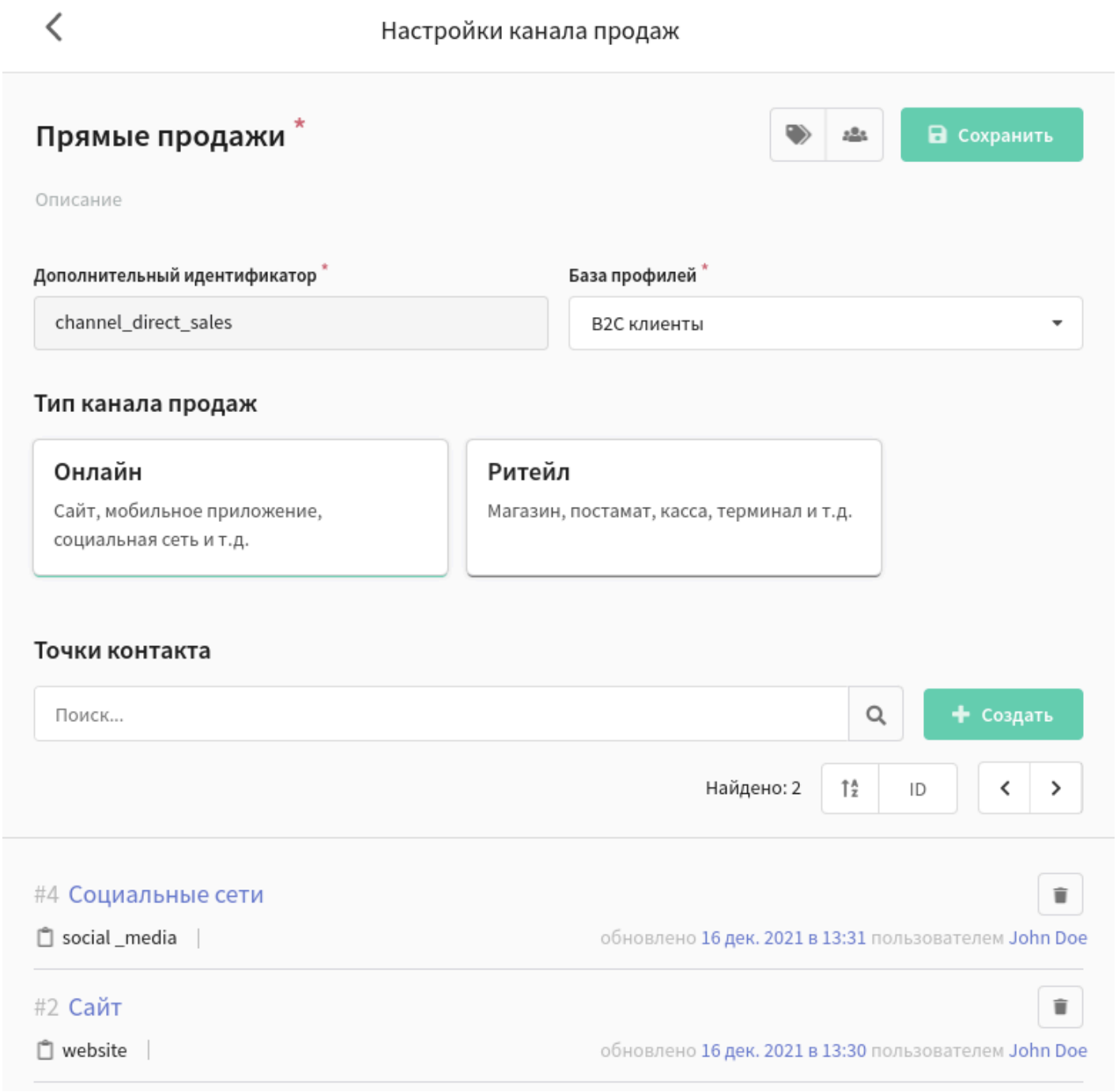
Точка контакта — это площадка или инструмент, с помощью которого вы продаете клиенту товары или услуги (сайт, магазин, постамат, отдельный терминал, соцсеть, call-центр и др.). Точка контакта связывает канал продаж и заказ.

Чтобы создать канал, в главном меню перейдите в раздел **Маркет** —> **Настройки**. На вкладке **Каналы продаж** создайте новый канал. Укажите:

- название канала
- дополнительный идентификатор
- базу профилей
- тип канала: онлайн или ритейл

Далее сохраните канал и снова перейдите в его настройку. Создайте точки контакта.

Обратите внимание, каждая точка может быть привязана только к одному каналу. Идентификатор точки контакта будет использоваться в API-запросе для импорта заказа.



Шаг 2. Загрузка в платформу данных о товарах

Товары в Маркете представлены двумя сущностями: продукты и SKU.

TIP

Продукт — это товар или услуга для продажи. **SKU** — это модификатор продукта, т.е. группа товаров, которая выделяется в одном продукте по некоторым параметрам (цвет, размер, вес, модель, комплектация и т.д.).

SKU привязывается к определенному продукту и не может существовать без него. Например, `Apple iPhone 12` — это продукт, а `Apple iPhone 12 128GB` и `Apple iPhone 12 256GB` — это его SKU.

Добавить данные о товарах в платформу можно тремя способами:

- создание вручную
- импорт из файла
- импорт через API

Давайте импортируем товары из файла. Для этого вам необходимо подготовить файл в **формате yml**.

> **TIP**
>
> Если возникнут вопросы по импорту, читайте подробную инструкцию.

Ниже показаны примеры yml-файлов:

Продукт

Файл с таким содержимым добавит в маркет два продукта: `Apple iPhone 12` и `Xiaomi 12X`.

```
<?xml version="1.0" encoding="UTF-8"?>
<yml_catalog date="2017-02-05 17:22">
    <shop>
    <name>BestSeller</name>
    <company>Tne Best inc.</company>
    <url>http://best.seller.ru</url>
    <categories>
      <category id="10">Смартфоны</category>
      <category id="101" parentId="10">Apple</category>
      <category id="102" parentId="10">Xiaomi</category>
    </categories>
    <offers>
      <offer id="iphone12" available="true">
        <name>Apple iPhone 12</name>
        <vendor>Apple</vendor>
        <url>http://best.seller.ru/product_page.asp?pid=12348</url>
        <price>83990</price>
        <oldprice>89990</oldprice>
        <currencyId>RUB</currencyId>
        <categoryId>101</categoryId>
        <picture>https://example.com/img/apple_phone.png</picture>
        <delivery>true</delivery>
        <manufacturer_warranty>true</manufacturer_warranty>
        <barcode>0156789012</barcode>
          </offer>
      <offer id="xiaomi12" available="true">
                <name>Xiaomi 12X</name>
        <vendor>Xiaomi</vendor>
        <url>http://best.seller.ru/product_page.asp?pid=12345</url>
        <price>62990</price>
        <oldprice>69990</oldprice>
        <currencyId>RUB</currencyId>
        <categoryId>102</categoryId>
        <picture>https://example.com/img/xiaomi_phone.png</picture>
        <delivery>true</delivery>
        <manufacturer_warranty>true</manufacturer_warranty>
        <barcode>0123456789379</barcode>
          </offer>
    </offers>
  </shop>
</yml_catalog>
```

SKU

Файл добавит два SKU для продукта `Apple iPhone 12`: `Apple iPhone 12 128GB` и `Apple iPhone 12 256GB`.

```
<?xml version="1.0" encoding="UTF-8"?>
<yml_catalog date="2017-02-05 17:22">
    <shop>
    <name>BestSeller</name>
    <company>Tne Best inc.</company>
    <url>http://best.seller.ru</url>
    <offers>
      <offer id="apple12_128gb" group_id="iphone12" available="true">
        <name>Apple iPhone 12 128GB</name>
        <vendor>Apple</vendor>
        <url>http://best.seller.ru/product_page.asp?pid=12358</url>
        <price>71990</price>
        <oldprice>72990</oldprice>
        <currencyId>RUB</currencyId>
        <categoryId>101</categoryId>
        <picture>https://example.com/img/apple_phone_128.png</picture>
        <delivery>true</delivery>
        <manufacturer_warranty>true</manufacturer_warranty>
        <barcode>0156789012</barcode>
          </offer>
      <offer id="apple12_256gb" group_id="iphone12" available="true">
                <name>Apple iPhone 12 256GB</name>
        <vendor>Apple</vendor>
        <url>http://best.seller.ru/product_page.asp?pid=12359</url>
        <price>83990</price>
        <oldprice>89990</oldprice>
        <currencyId>RUB</currencyId>
        <categoryId>102</categoryId>
        <picture>https://example.com/img/apple_phone_256.png</picture>
        <delivery>true</delivery>
        <manufacturer_warranty>true</manufacturer_warranty>
        <barcode>0156789012</barcode>
          </offer>
    </offers>
  </shop>
</yml_catalog>
```

TIP

Список всех продуктов и SKU хранится в элементе **offers.** Один продукт или SKU записывается в отдельном элементе **offer**.

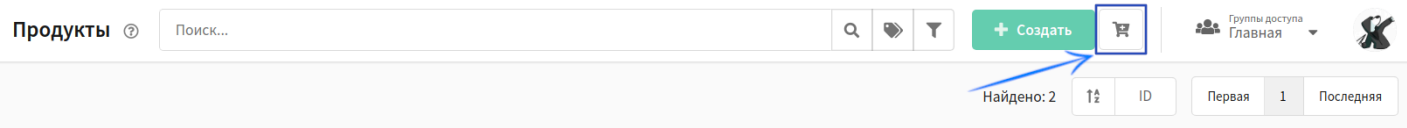
Для элемента **offer** указывается атрибут **id ,** уникальный для каждого товара.

Структура yml файла для продукта и SKU практически не отличается. Единственное отличие — при импорте SKU для элемента **offer** указывается не только **id,** но также **group_id** – идентификатор продукта.

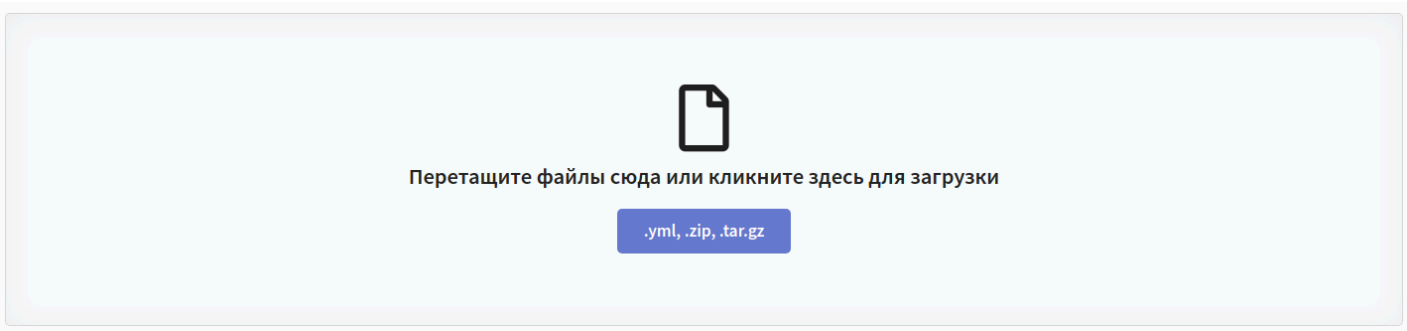
Обратите внимание, SKU – это различные вариации одного продукта, поэтому значение атрибута **group_id** должно быть одинаковым для всех SKU одного продукта.

Чтобы импортировать файл со списком продуктов, перейдите в раздел **Маркет** —> **Продукты.** Нажмите на иконку

в верхней панели, а затем выберите, что именно хотите импортировать — продукт.



Кликните на область загрузки и выберите файл на своем устройстве:



Запустите процесс импорта, нажав кнопку “**Импортировать**” в верхней панели.

Затем импортируйте SKU, повторив все действия, описанные в этом пункте.

Шаг 3. Настройка API запроса на добавление заказа

Когда посетитель интернет-магазина оформит заказ, необходимо отправить API-запрос в платформу с данными о заказе.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/orders/import`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен. Убедитесь, что для токена доступны роли, позволяющие работать с маркетом.
data	array of objects	[{ … }]	Да	Данные о заказе. Подробнее.

Параметр	Тип	Пример	Обязательный	Описание
Объект data				
external_id	string	“1234001”	Да	Идентификатор заказа
matching	string	“email”		Режим поиска подписчика. По умолчанию – email. Для каждого типа матчинга в теле запроса необходимо передавать определенные поля. Подробнее об этих полях можно узнать здесь.
endpoint	string	“website”	Да	Идентификатор точки продажи
status	string	“new” “delivered” “paid” “canceled” “part_refunded”	Да	Идентификатор статуса заказа
total_price	string	“71990”	Да	Общая стоимость заказа. Складывается из суммы каждой позиции в заказе и стоимости доставки.
currency	string	“RUB”	Да	Валюта, в которой указана стоимость заказа.
create_time	string	“2022-04-29T12:48:23Z”	Нет	Время создания заказа
update_time	string	“2022-04-29T12:48:23Z”	Нет	Время обновления заказа

lines	array	[
		{
		…
		}
		]
			Да	Данные о позициях заказа. Минимум одна позиция.

Объект lines				
external_id	string	“line_01”	Да	Идентификатор позиции
product	string	“iphone12”	Не должен присутствовать в запросе, если задан `sku`	Идентификатор продукта

Параметр	Тип	Пример	Обязательный	Описание
sku	string	“apple12_128gb”	Не должен присутствовать в запросе, если задан `product`	Идентификатор SKU
base_price_per_item	string	“71990”	Да	Базовая цена
min_price_per_item	string	“71990”	Да	Минимальная цена
final_price_per_item	string	“71990”	Да	Конечная цена
count	int	1	Да	Количество товара в позиции

Пример запроса

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "data": [
    {
      "external_id": "1234001",
      "matching": "email",
      "email": "johndoe@example.com",
      "endpoint": "website",
      "status": "new",
      "total_price": "71990",
      "create_time": "2022-04-29T12:48:23Z",
      "update_time": "2022-04-29T12:48:23Z",
      "currency": "RUB",
      "lines": [
        {
          "external_id": "line_01",
          "product": "iphone12",
          "base_price_per_item": "71990",
          "min_price_per_item": "71990",
          "final_price_per_item": "71990",
          "count": 1
        }
      ]
    }
  ]
}
```

Пример ответа


```
{
  "error": 0,
  "error_text": "",
  "result": [
    {
      "error": 0,
      "external_id": "1234000"
    }
  ]
}
```


Возвращаемые параметры

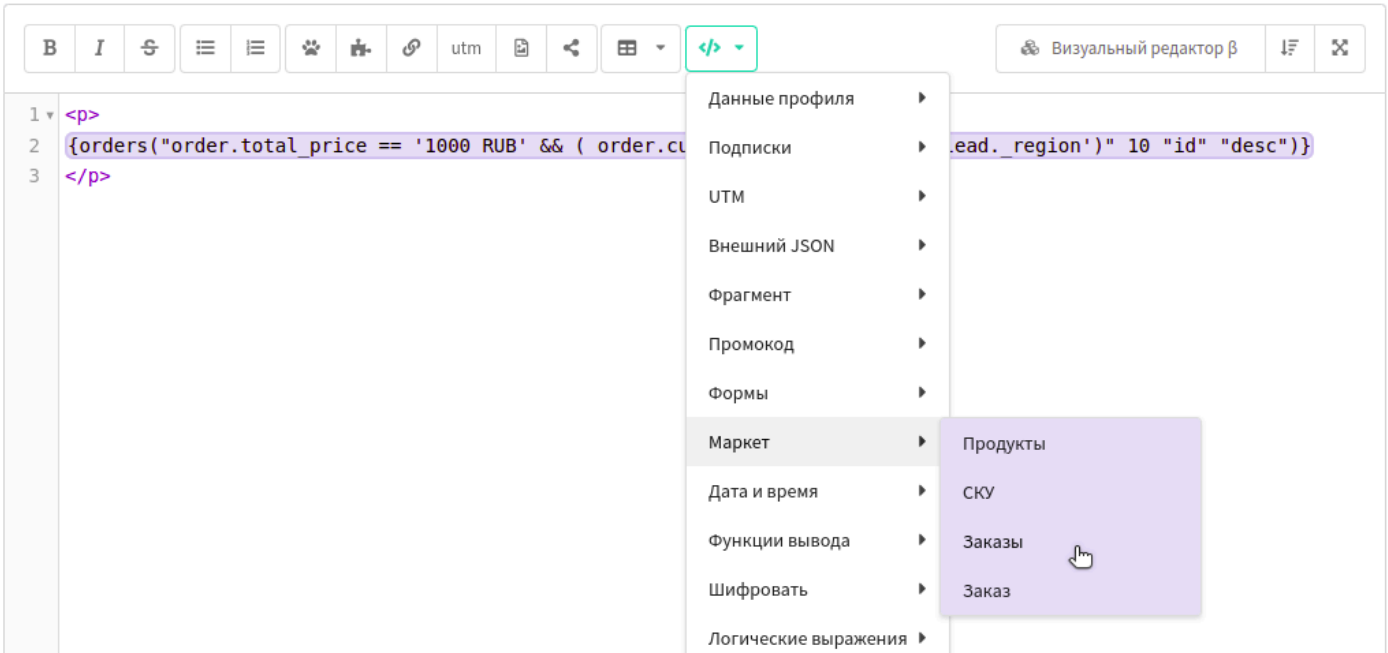
Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
result	array	Идентификаторы заказов, которые были успешно импортированы

Шаг 4. Создание шаблона сообщения

Перейдите в раздел **Шаблоны**. Добавьте новый шаблон для email канала.

Чтобы использовать информацию о заказе в рассылке, необходимо добавить переменные маркета в шаблон письма. Для этого в редакторе нажмите `</>`. В списке выберите “**Маркет**” и вставьте переменную “**Заказы**”.

В редактор добавится паттерн функции `orders`. Эта функция выбирает заказы из списка заказов профиля.



Нам надо добавить в письмо только информацию о последнем заказе со статусом “Новый”. Для этого потребуется изменить аргументы функции:

```
{orders("order.status == '1'" 1 "create_time" "desc")}
```

Функция состоит из четырех аргументов:

Аргумент	Пример	Описание
Фильтр	`"order.status == '1'"`	Функция выберет только те заказы, которые соответствуют условию “статус заказа новый”.
Лимит	`1`	Функция выберет только первый элемент в списке заказов.
Поле для сортировки	`"create_time"`	Выбранные элементы будут отсортированы по полю “create_time” (время создания заказа).
Направление сортировки	`"desc"` `"asc"`	Элементы будут отсортированы по убыванию значения в поле “create_time”.

Функция **orders** получит массив заказов. В данном случае это будет массив с одним элементом. Используйте цикл, чтобы перебрать массив.

В шаблоне письма подставьте 2 переменные:

- `$el.external_id` — дополнительный идентификатор заказа.
- `$el.total_price` — общая сумма заказа.

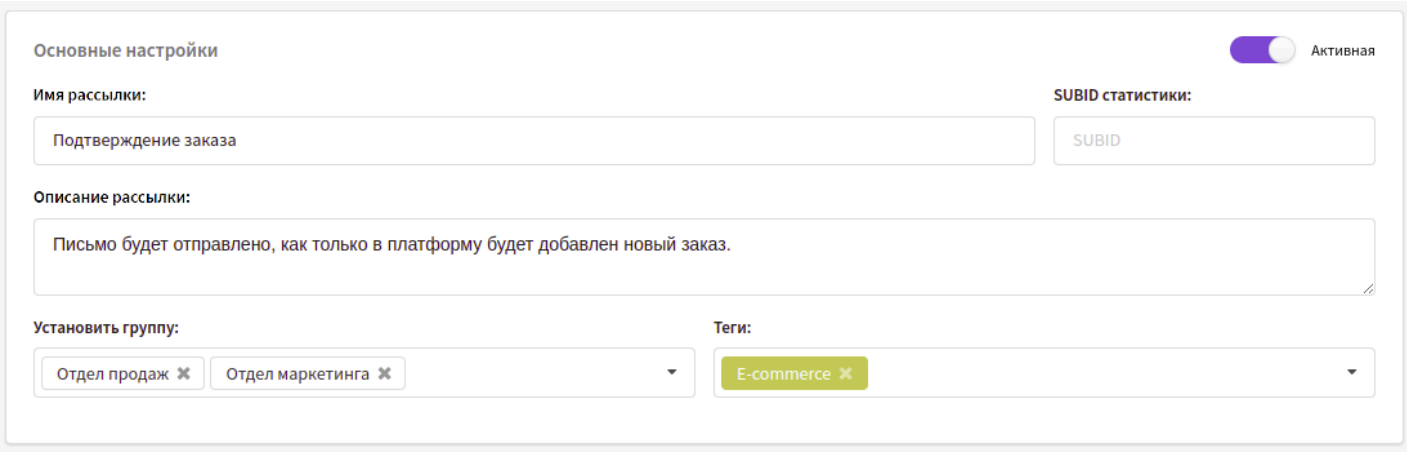
```
{for $index $el = orders("order.status == '1'" 1 "create_time" "desc")}
  Спасибо! Ваш заказ №{$el.external_id} подтвержден.
  К оплате: {$el.total_price}
{else}
  if empty array
{end}
```

TIP

Подробнее о работе с переменными маркета написано в этой статье. Там вы найдете информацию о полях, которые можно использовать в фильтре или подставлять в шаблон.

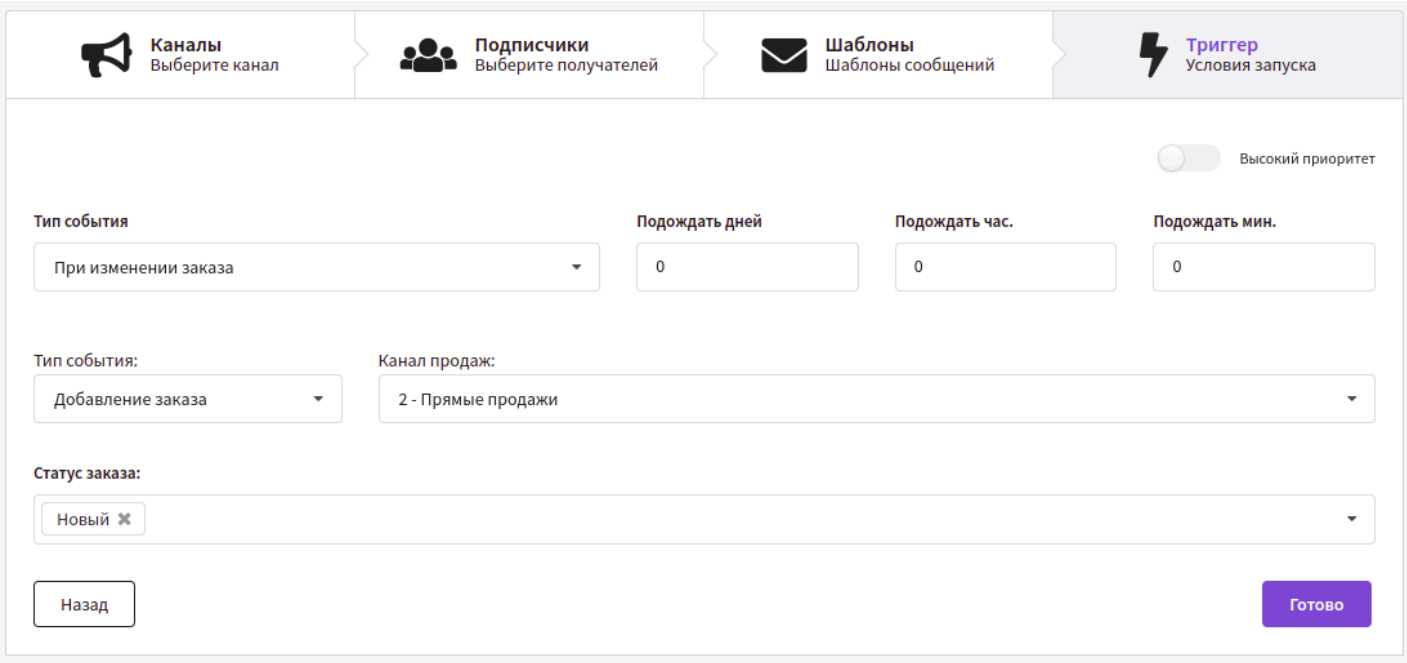
Шаг 5. Настройка триггерной рассылки

Откройте раздел **Рассылки** → **Триггеры.** Создайте новую рассылку. Не забудьте её активировать.



Выберите email-канал и добавьте нужный шаблон сообщения. На вкладке **Триггер** укажите:

- **Тип события**: “При изменении заказа” —> “Добавление заказа”. Триггер сработает, как только в маркет будет добавлен новый заказ.
- **Канал продаж**: в этом канале должна быть указана база профилей, где находятся данные покупателей вашего интернет-магазина.
- **Статус заказа:** “Новый”. Триггер сработает, если в маркет будет добавлен заказ со статусом “новый”.



Нажмите **Готово** и сохраните рассылку.

TIP

Чтобы протестировать работу триггера, вы можете добавить заказ вручную.

Продукты

	Продукты — это товар или услуга для продажи. Для каждого продукта создается отдельная карточка с названием, описанием, изображением и другими данными о товаре. В один продукт можно добавить несколько SKU (модификатор товара). Чтобы начать работу с продуктами, откройте раздел **"Маркет" —> "Продукты"**.

Создать продукт или SKU можно несколькими способами:

1. **Вручную**
2. **Импорт продукта из файла**
3. **Импорт через фид по расписанию**
4. **Импорт через API**

Создание продукта вручную

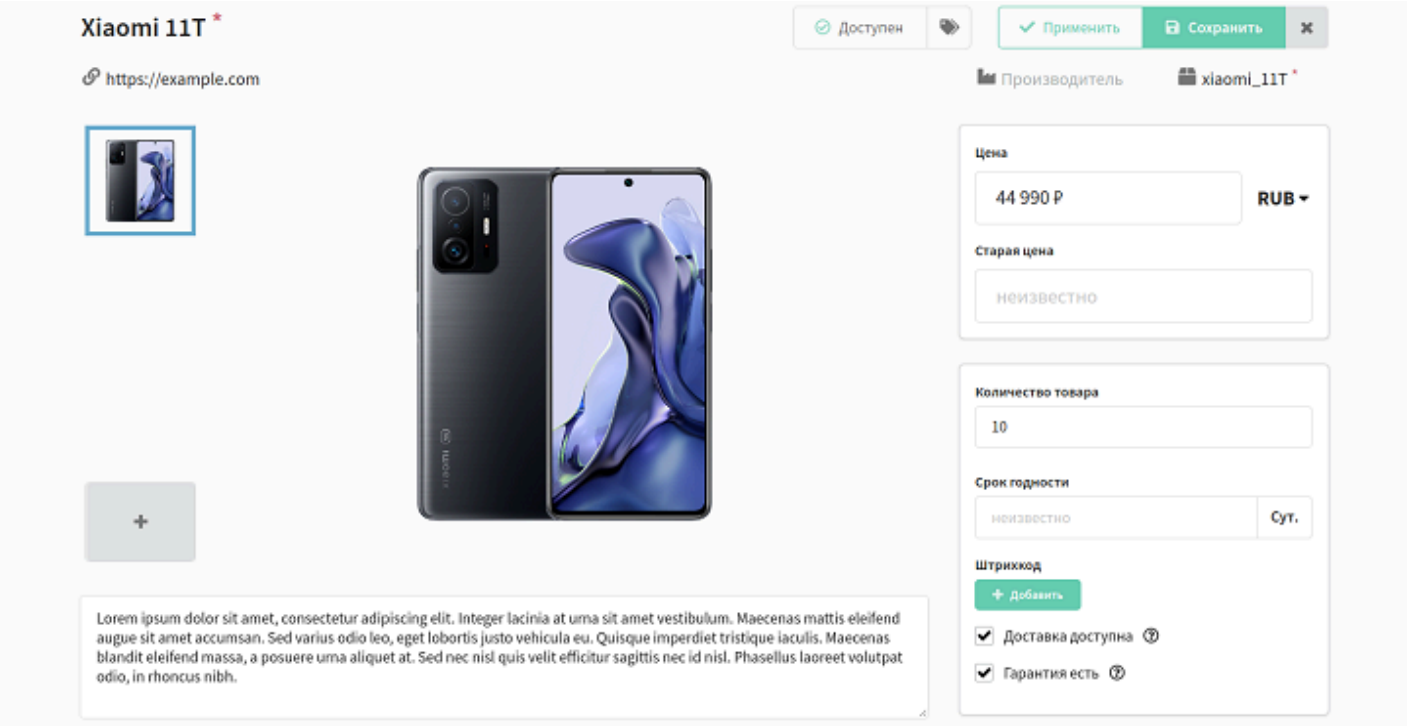
В этой статье описывается, как создать продукт или SKU вручную в маркете Altcraft. Чтобы начать работу, перейдите в раздел “**Продукты**” и нажмите кнопку “**Создать**” в верхней панели.

Основная информация о продукте

При создании нового продукта необходимо указать его **название** и **дополнительный идентификатор**. Идентификатор может состоять только из цифр и латинских букв. Максимальная длина — 20 символов.

Остальные поля не являются обязательными для заполнения:

- **ссылка на продукт** – URL страницы товара на сайте магазина
- **изображения** товара
- **описание**
- **доступен/недоступен** – есть ли товар в наличии
- **теги** для быстрого поиска
- **производитель** – название производителя
- **цена** (выберите валюту) – актуальная цена товара
- **старая цена**
- **количество** товара доступного для продажи
- **срок годности** или службы (в сутках)
- **штрихкод** товара от производителя в одном из форматов (EAN-13, EAN-8, UPC-A, UPC-E), можно добавлять несколько штрихкодов.
- **доставка** – возможность курьерской доставки (по всем регионам, в которые доставляет магазин).
- **гарантия** – есть или нет у товара официальной гарантии производителя



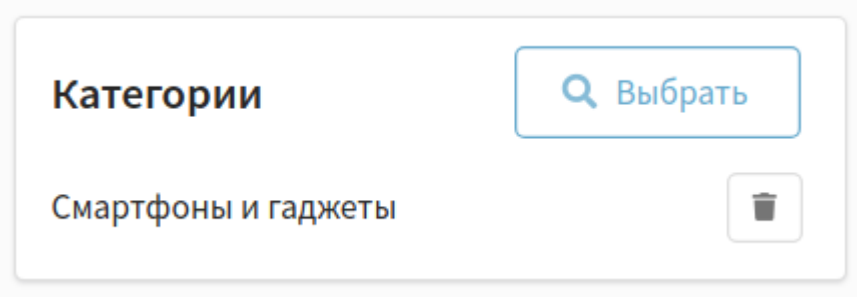
Дополнительные поля

Заполните дополнительные поля продукта. В них можно записыватьважные характеристики товара — цвет, размер, объем, материал, вес, модель и т. д.

Дополнительные поля создаются в разделе **“Настройки”**.

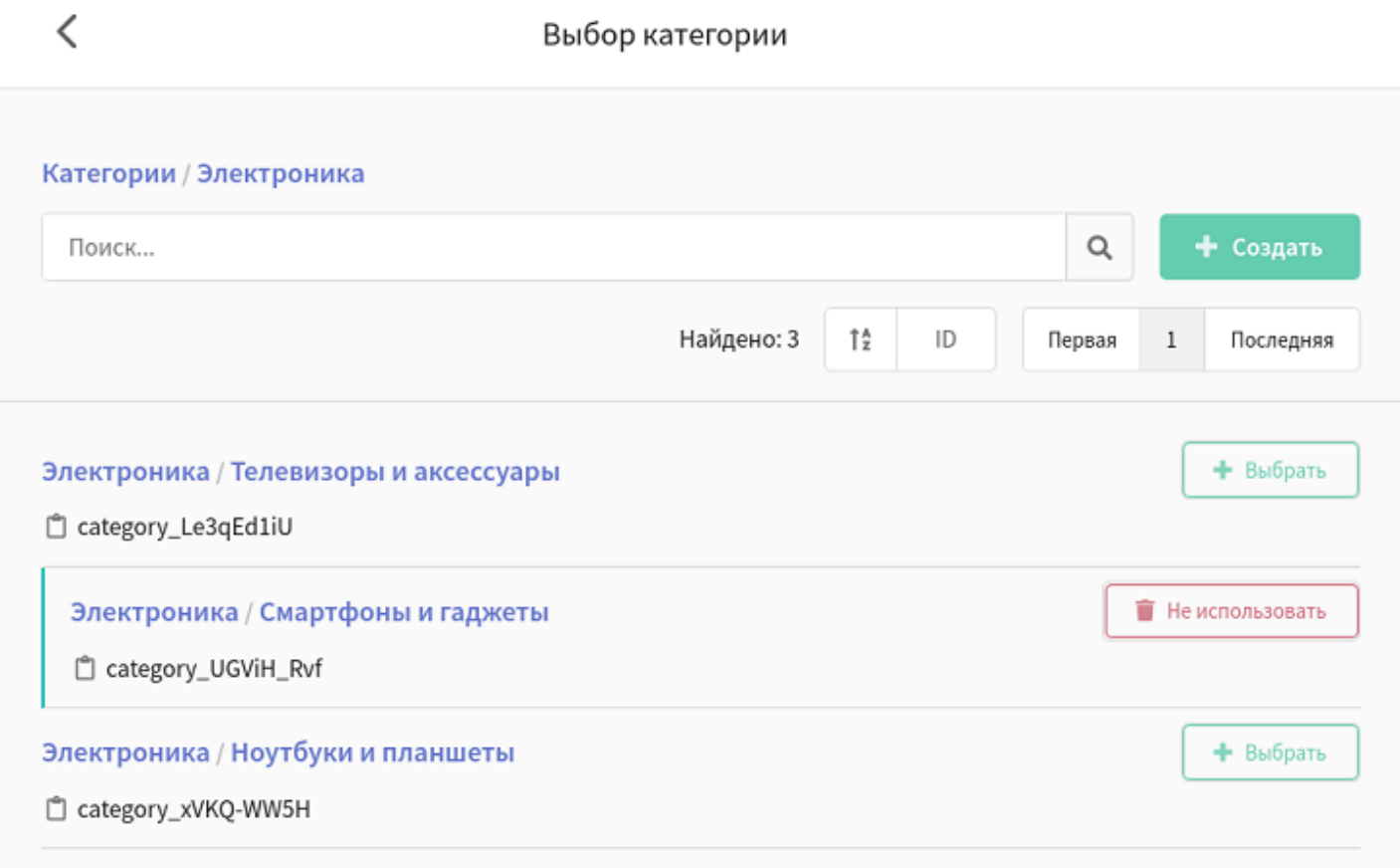
Категории

Укажите категории для продукта, нажав на кнопку “**Выбрать**”:



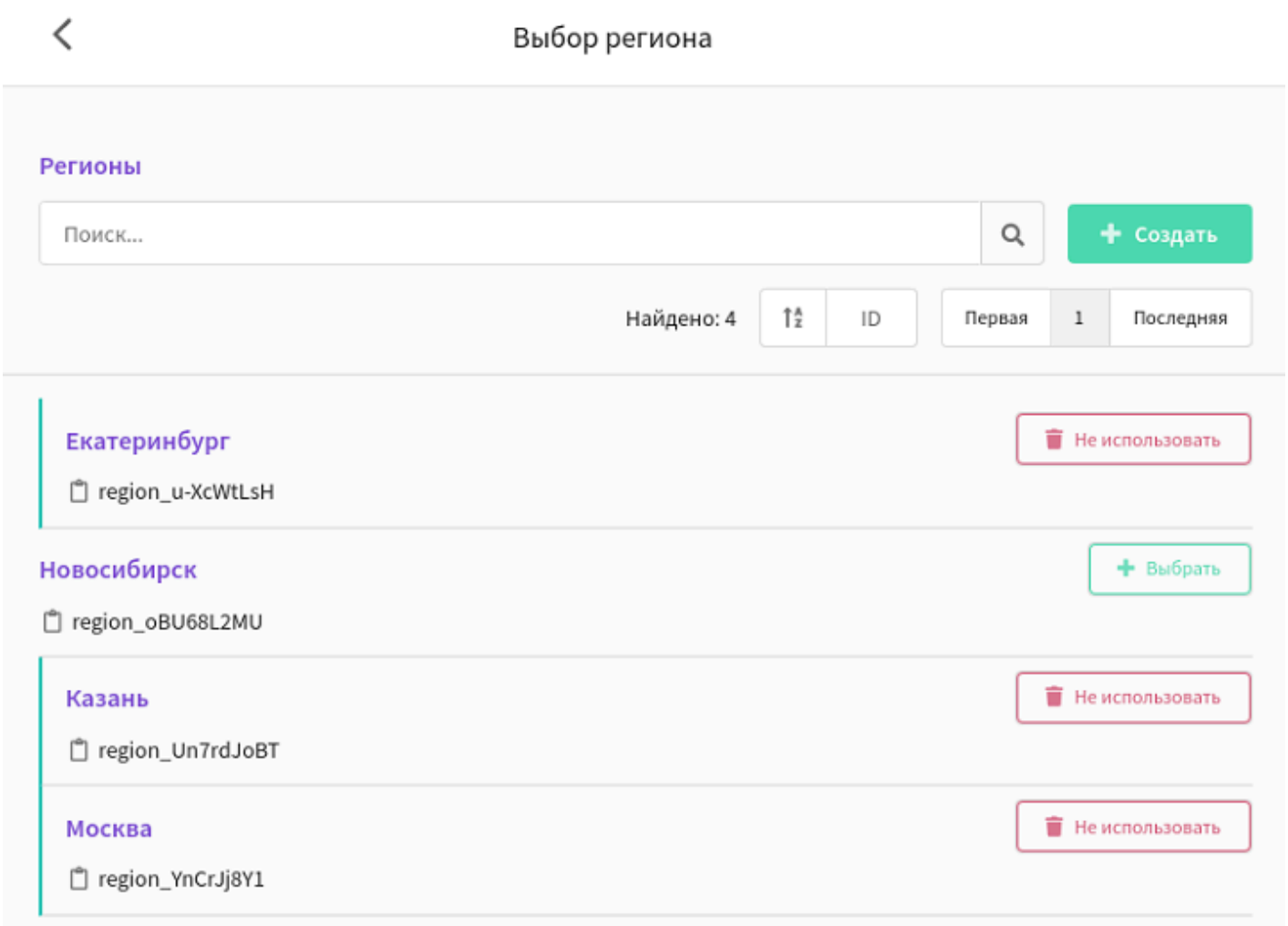
Категории создаются в настройках маркета. Вы можете быстро перейти в этот раздел прямо из окна создания продукта, кликнув “**Создать**”.

Если вы уже создали необходимые категории, добавьте в них продукт с помощью кнопки “**Выбрать**”:



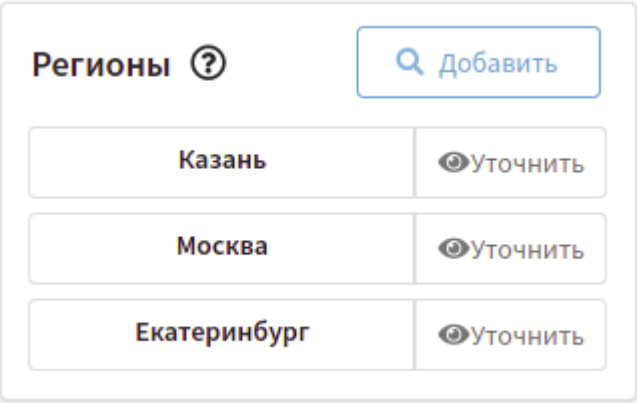
Регионы

Добавьте регионы, где возможна продажа продукта:



Параметры продукта можно изменять в зависимости от региона. Например, если вы не осуществляете доставку в Казань, измените параметр “Доставка доступна” для этого региона.

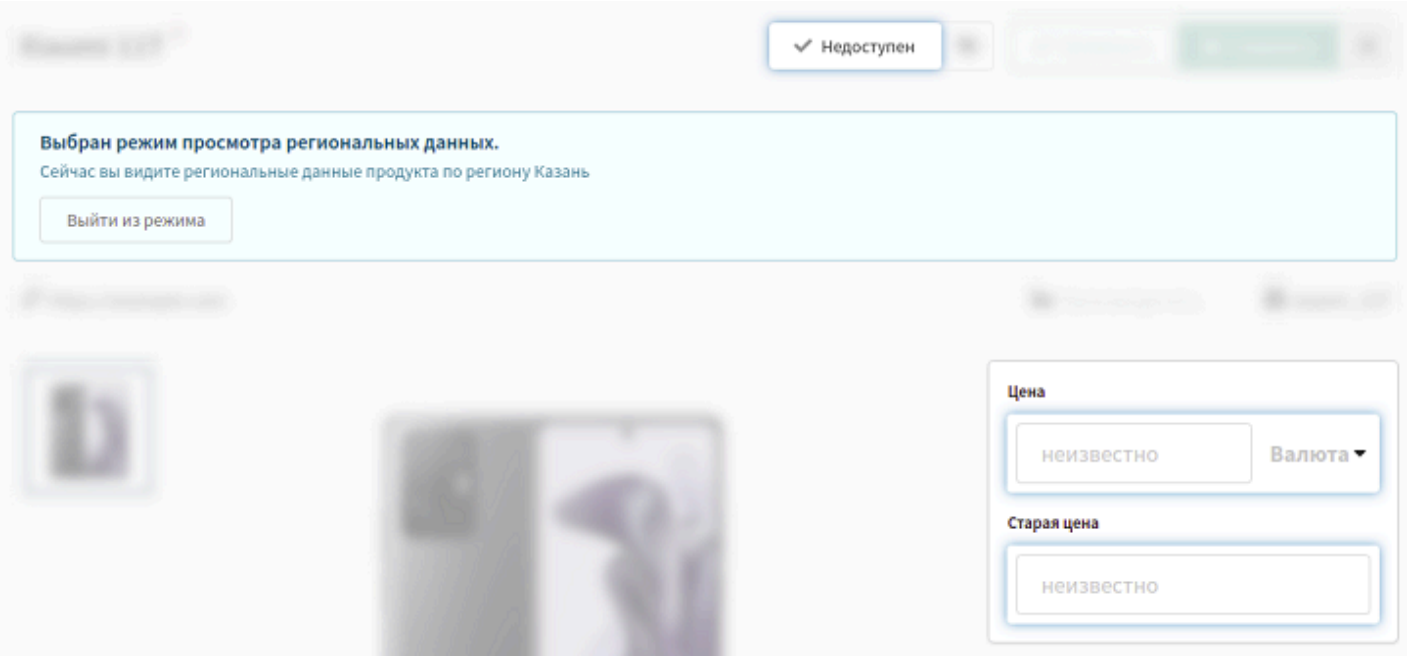
Чтобы отредактировать информацию о продукте для конкретного региона, нажмите “**Уточнить**” – и вы перейдете в режим просмотра региональных данных:



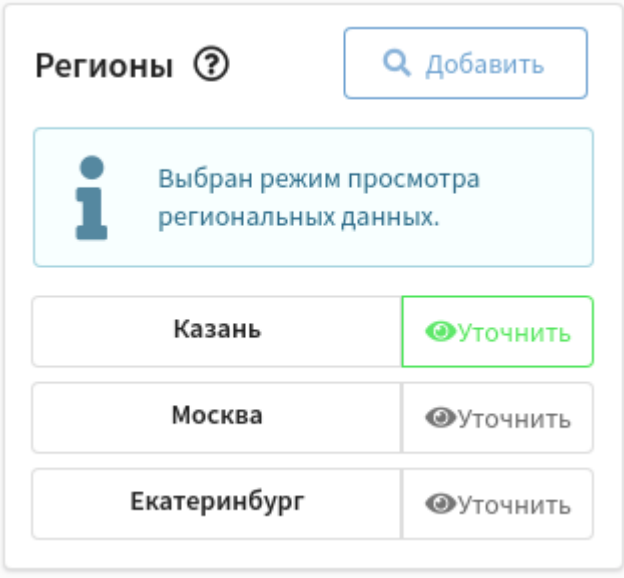
В этом режиме вы можете изменить следующую информацию о продукте:

- доступен или недоступен продукт для региона;
- цена;
- старая цена;

- возможность доставки в регион;
- дополнительные поля.



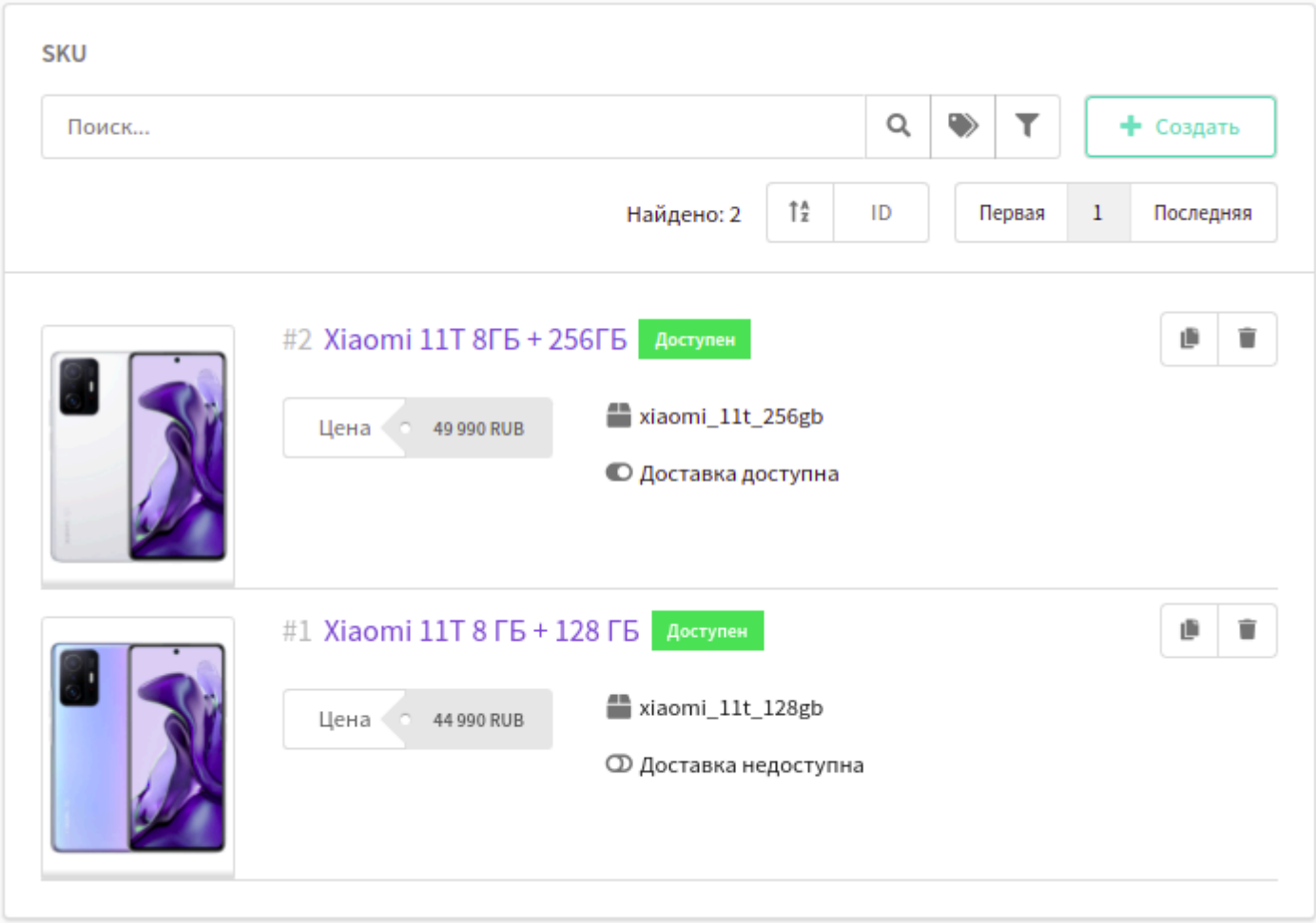
После заполнения региональных данных нажмите “**Уточнить**” напротив конкретного региона, чтобы сохранить изменения:



SKU

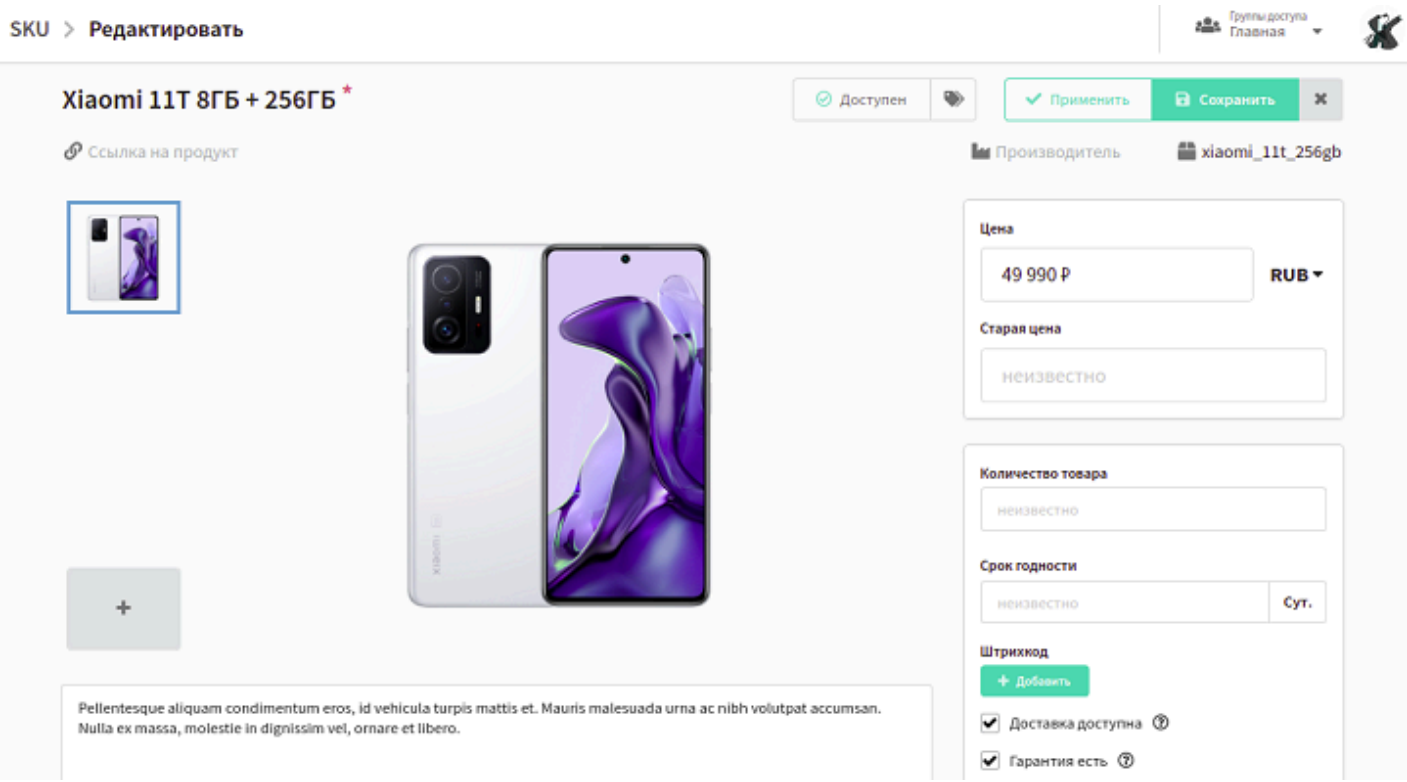
TIP

SKU — это модификатор продукта, т.е. группа товаров, которая выделяется в одном продукте по некоторым параметрам (цвет, размер, вес, модель, комплектация и т.д.). SKU привязывается к определенному продукту и не может существовать без него.



Процесс создания SKU аналогичен созданию продукта:

- введите название и дополнительный идентификатор;
- заполните карточку товара (ссылка, изображение, производитель, цена, доставка, гарантия и др. данные);
- добавьте SKU в определенную категорию товаров;
- добавьте регионы и укажите для каждого из них свои региональные данные;
- заполните дополнительные поля.



После завершения настройки не забудьте сохранить продукт. Теперь вы можете использовать продукт или его SKU при создании заказа.

Импорт продукта из файла

В этой статье описывается, как загрузить данные о товарах из файла в маркет платформы Altcraft. Товары в Маркете представлены двумя сущностями: **продукты** и **SKU**.

Продукт — это товар или услуга для продажи. **SKU** — это модификатор продукта, т.е. группа товаров, которая выделяется в одном продукте по некоторым параметрам (цвет, размер, вес, модель, комплектация и т.д.).

SKU привязывается к определенному продукту и не может существовать без него. Например, Apple iPhone 12 — это продукт, а Apple iPhone 12 64GB и Apple iPhone 12 128GB — это его SKU.

Для загрузки данных о товарах в платформу можно использовать **файл в формате yml** или **.zip/.tar.gz архив**, содержащий yml-файлы.

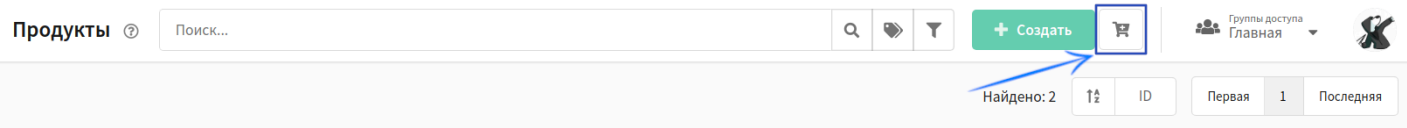
INFO

Формат **YML (Yandex Market Language)** — это формат данных для товарных платформ, разработанный Яндексом. Подробную информацию о структуре yml-файла, о его параметрах и свойствах, а также примеры готовых yml-фидов с товарами, можно найти **здесь**.

Загрузка файлов в платформу

Чтобы импортировать продукт или SKU из файла, нажмите на иконку

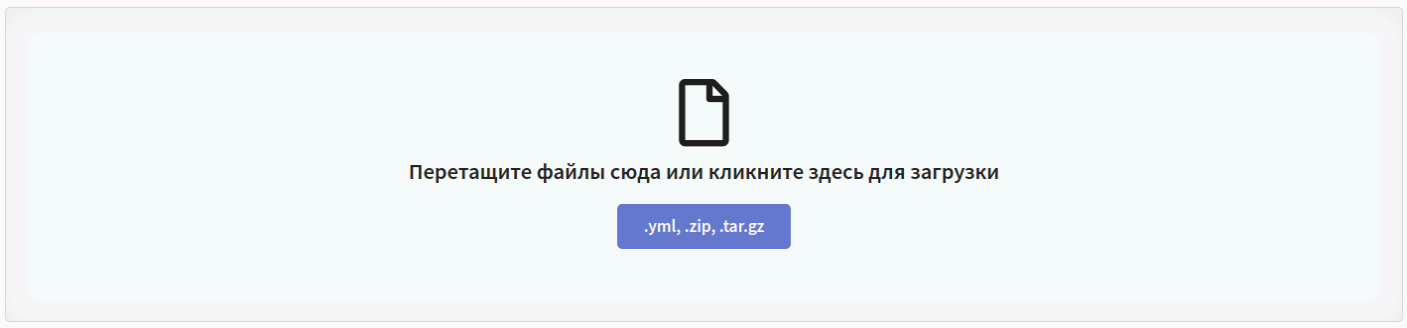
в верхней панели, а затем выберите, что именно хотите импортировать: продукт или SKU.



TIP

Если вы храните данные о продуктах и данные о SKU в одном файле, вам не потребуется редактировать его. В процессе импорта будут загружены данные только той сущности, которую вы выберете.

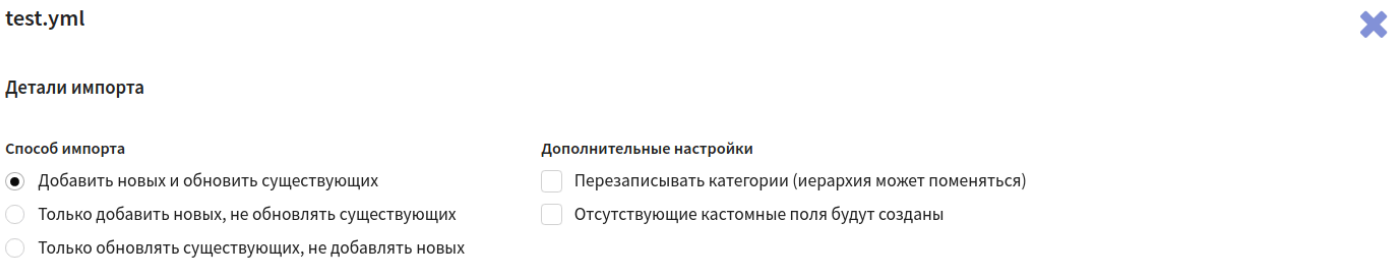
Нажмите на область загрузки и выберите файл на своем устройстве:



Детали импорта

Укажите **способ импорта** – он будет зависеть от того, добавляете вы данные о новых товарах или обновляете товары, которые уже есть в маркете:

- Добавить новых и обновить существующих
- Только добавить новых, не обновлять существующих
- Только обновлять существующих, не добавлять новых



В блоке **Детали импорта** активируются дополнительные опции:

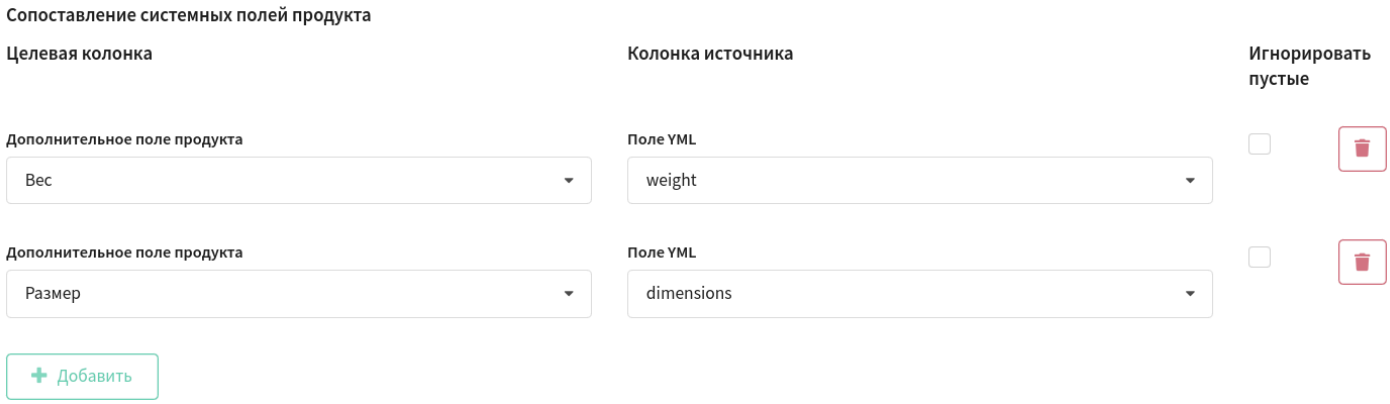
- **Перезаписывать категории** (иерархия категорий может измениться)
- **Отсутствующие пользовательские поля будут созданы** – в платформе автоматически сгенерируются дополнительные поля для элемента `param` в yml-файле. Altcraft подберет колонки для элементов `param`, если их наименование будет такое же, как у дополнительных полей в маркете.

Сопоставление системных полей

В этом блоке вы можете настроить импорт дополнительной информации о товаре в платформу.

Целевая колонка — это дополнительное поле в маркете, которое создаётся в разделе “Настройки”.

Колонка источника — это дополнительные поля в структуре yml-файла.



На этом настройка импорта завершена. Запустите процесс, нажав кнопку **“Импортировать”** в верхней панели. Результат импорта вы можете посмотреть в разделе **“Задания”**.

Импорт по расписанию

В этой статье описывается, как настроить синхронизацию продуктов и SKU в Маркете с данными вашего товарного фида. Данная функция позволяет не загружать новый yml-файл каждый раз, когда ваш фид обновляется.

После настройки импорта новая информация о продуктах будет автоматически подгружаться в Маркет по указанному расписанию. Информация будет подгружаться из yml-фида с товарами, на который будет необходимо указать ссылку.

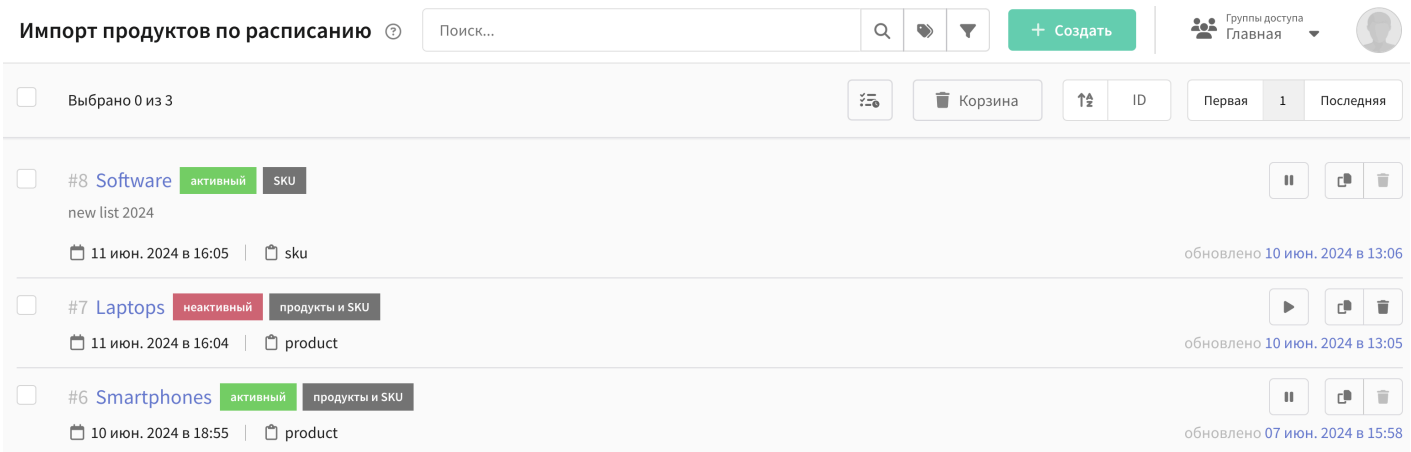
INFO

Формат YML (Yandex Market Language) — это формат данных для товарных платформ, разработанный Яндексом. Подробную информацию о структуре yml-файла, о его параметрах и свойствах, а также примеры готовых yml-фидов с товарами, можно найти **здесь**.

Шаг 1. Создайте новый импорт

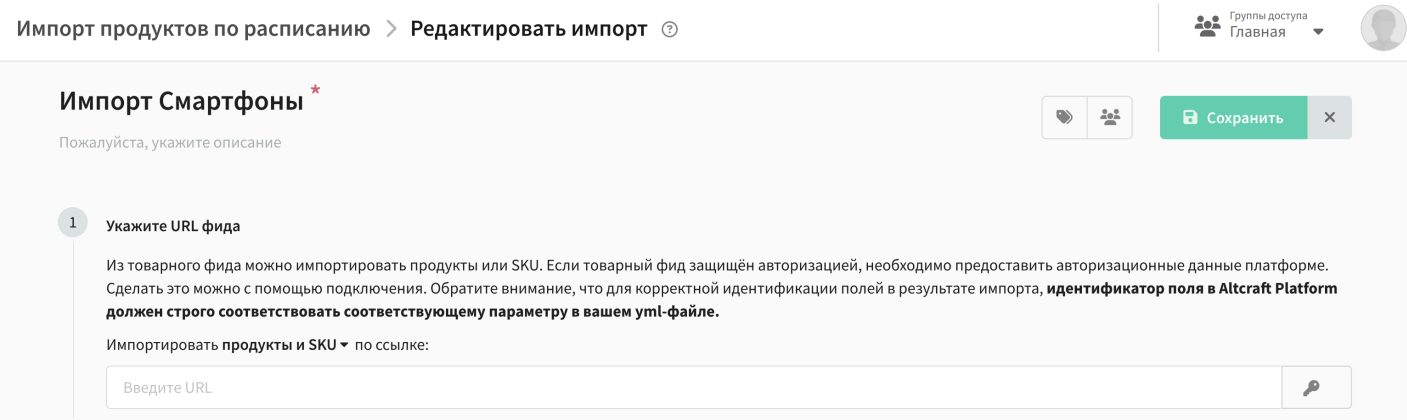
Перейдите в раздел **“Маркет”**, выберите **“Импорт продуктов по расписанию”**. Перед вами появится список существующих импортов с указанием краткой информации о них. Здесь же вы можете запустить, приостановить, клонировать или удалить импорт.

Для того, чтобы создать новый импорт, нажмите кнопку **“Создать”** в правом верхнем углу:



Шаг 2. Укажите URL фида

В соответствующее поле вставьте URL-адрес вашего фида. В пункте **“Импортировать продукты и SKU по ссылке”** можно выбрать то, какой тип объектов будет загружаться из фида — продукты и их SKU или только SKU. В первом случае платформа вначале импортирует все продукты, а потом добавит все соответствующие SKU, во втором — новые продукты будут полностью проигнорированы, а SKU будут загружаться для уже существующих продуктов:

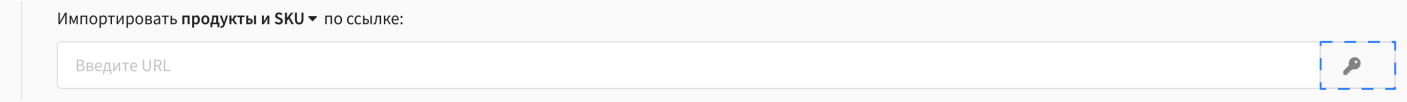


CAUTION

Если в вашем товарном фиде есть поля, которые не предусмотрены по умолчанию в Маркет платформы, то убедитесь, что вы создали соответствующие дополнительные поля в **Настройках маркета**, иначе импорт пройдет с ошибками.

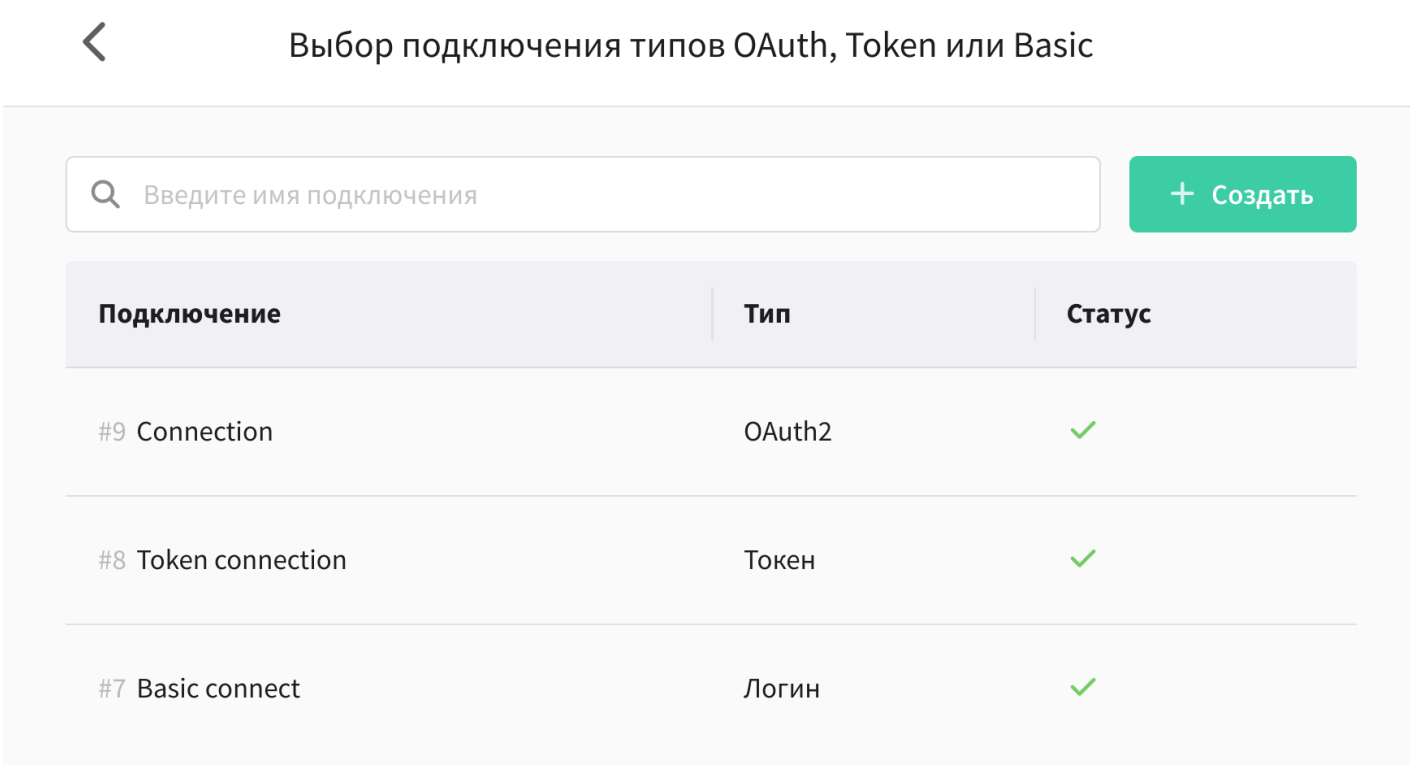
Выбор подключения

В случае, если ваш товарный фид защищен авторизацией, вам нужно добавить к импорту **подключение**. Сделать это можно, нажав на кнопку с ключом:



Перед вами появится окно выбора подключения. На платформе Altcraft сущуствует три типа подключений для авторизации:

- **OAuth2** — протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. При этом пользователю не надо доверять приложению логин и пароль.
- **Basic Authentication** — протокол HTTP-аутентификации, использующий логин и пароль для получения доступа к защищенной информации.
- **Token Authentication** — протокол аутентификации, использующий специальный токен, который предоставляется пользователю после успешной аутентификации для доступа к защищенным ресурсам.



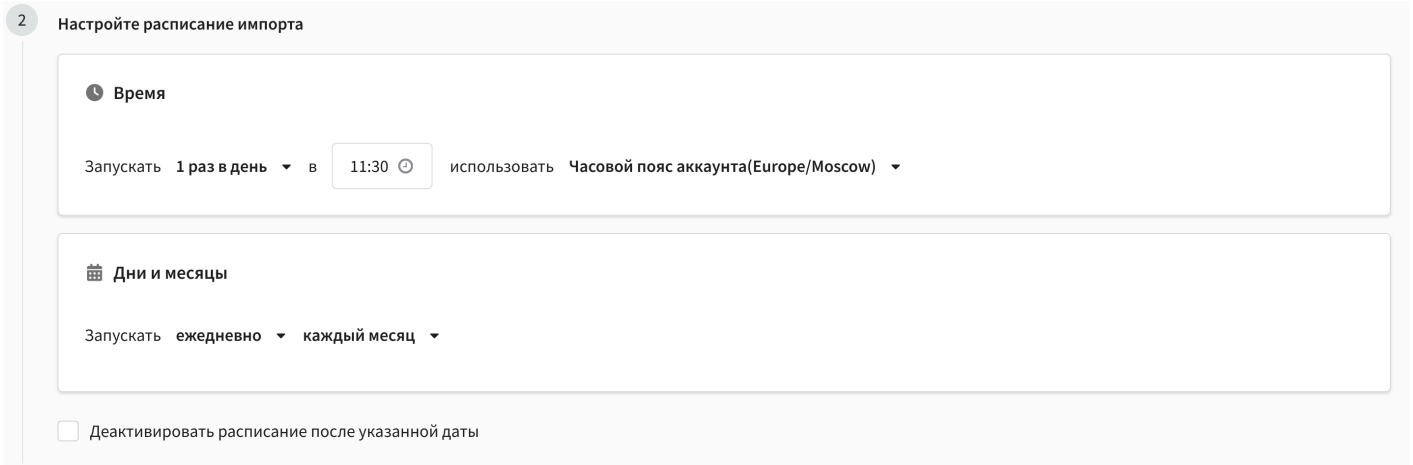
Если подключения отсутствуют, вы можете создать новое прямо в этом окне. Для этого нажмите **+ Создать** и выберите тип подключения: **OAuth2**, **Basic Authentication** или **Token Authentication**. Заполните требуемые поля и нажмите **“Сохранить”**.

TIP

Более подробная информация о создании и настройке подключения доступна в разделе **“Подключения”**.

Шаг 3. Настройте расписание импорта

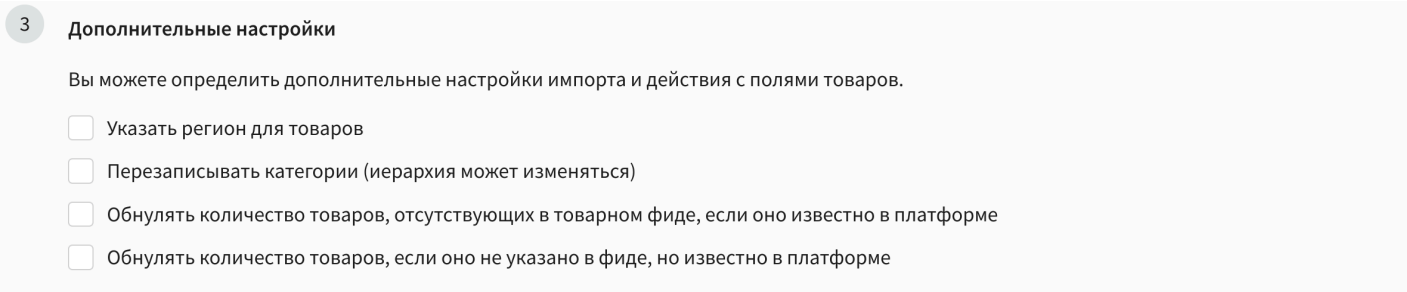
Из выпадающего меню выберите план расписания и настройте в зависимости от того, насколько часто вы хотите обновлять список продуктов. Для каждого плана расписания можно установить часовой пояс для расчёта. Также можно указать дату деактивации импорта:



Шаг 4. Укажите дополнительные настройки

После настройки основных параметров вы можете указать необходимые дополнительные опции:

- **Указать регион для товаров** — устанавливает регион для импортированных продуктов. Регион должен быть создан на платформе заранее.
- **Перезаписывать категории (иерархия может измениться)** — категории товаров будут перезаписаны в соответствии с содержанием фида.
- **Обнулять количество товаров, отсутствующих в товарном фиде, если оно известно в платформе** — если в фиде не указан какой-либо товар, то его количество на платформе станет равно 0.
- **Обнулять количество товаров, если оно не указано в фиде, но известно в платформе** — если в фиде не указано количество товара, то его количество на платформе станет равно 0.



Шаг 5. Сохранение и запуск импорта

Для сохранения созданного импорта нажмите кнопку **“Сохранить”**. Вы попадете на страницу обзора вашего импорта:

В контекстном меню доступны следующие функции:

— редактировать импорт

— запустить и активировать импорт прямо сейчас (помимо расписания)

— активировать/деактивировать импорт

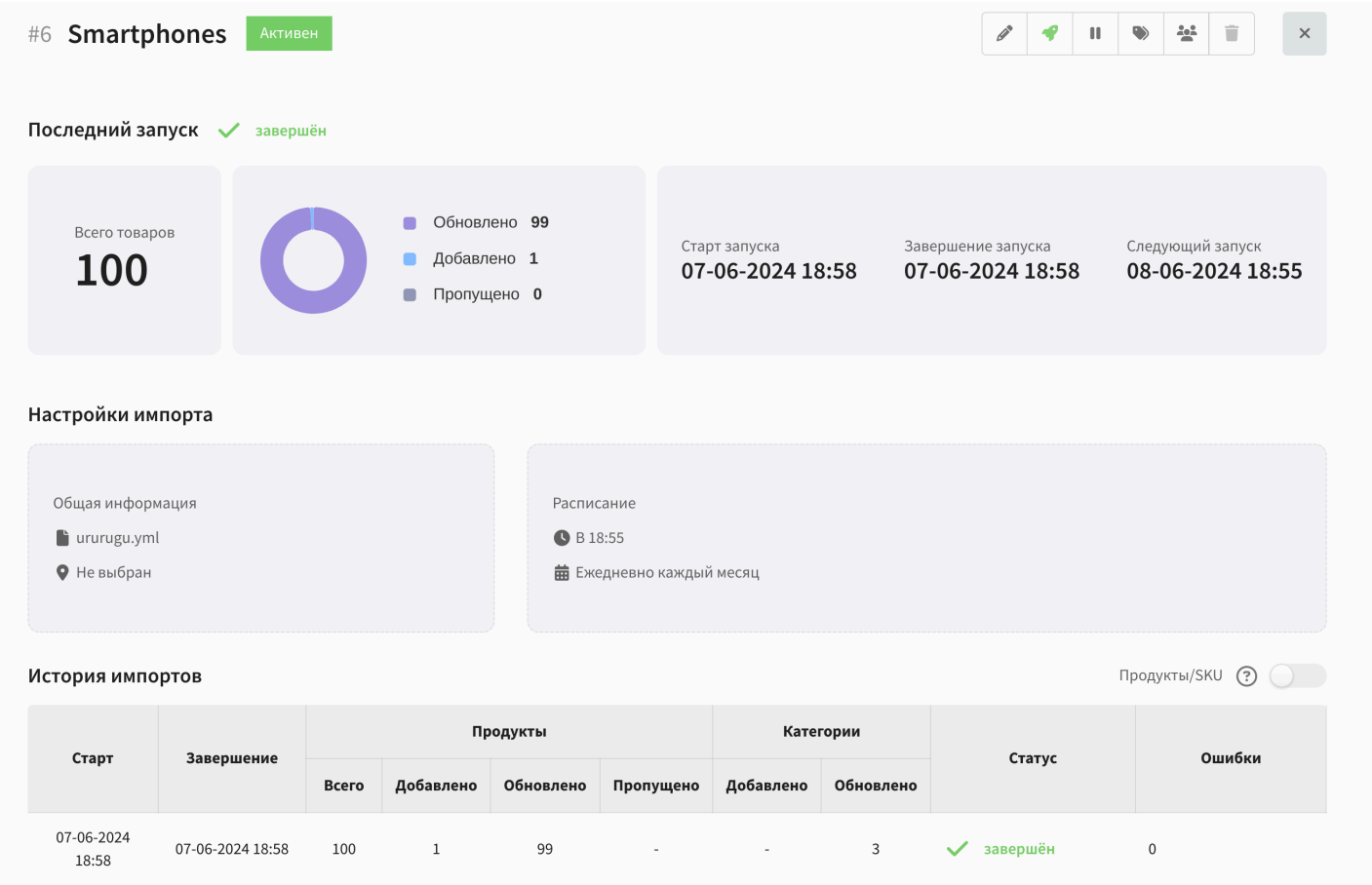
— назначить тег

— назначить группы доступа к данному импорту

— удалить импорт

После первого запуска в таблице ниже появится история импорта и информация о добавленных товарах. Там же можно посмотреть статус и количество ошибок, возникших в ходе импорта. Для того чтобы переключить тип

отображаемых данных (данные о продуктах или данные о SKU), активируйте переключатель **“Продукты/SKU”**:



Старт	Завершение	Продукты				Категории		Статус	Ошибки
		Всего	Добавлено	Обновлено	Пропущено	Добавлено	Обновлено		
07-06-2024 18:58	07-06-2024 18:58	100	1	99	-	-	3	завершён	0

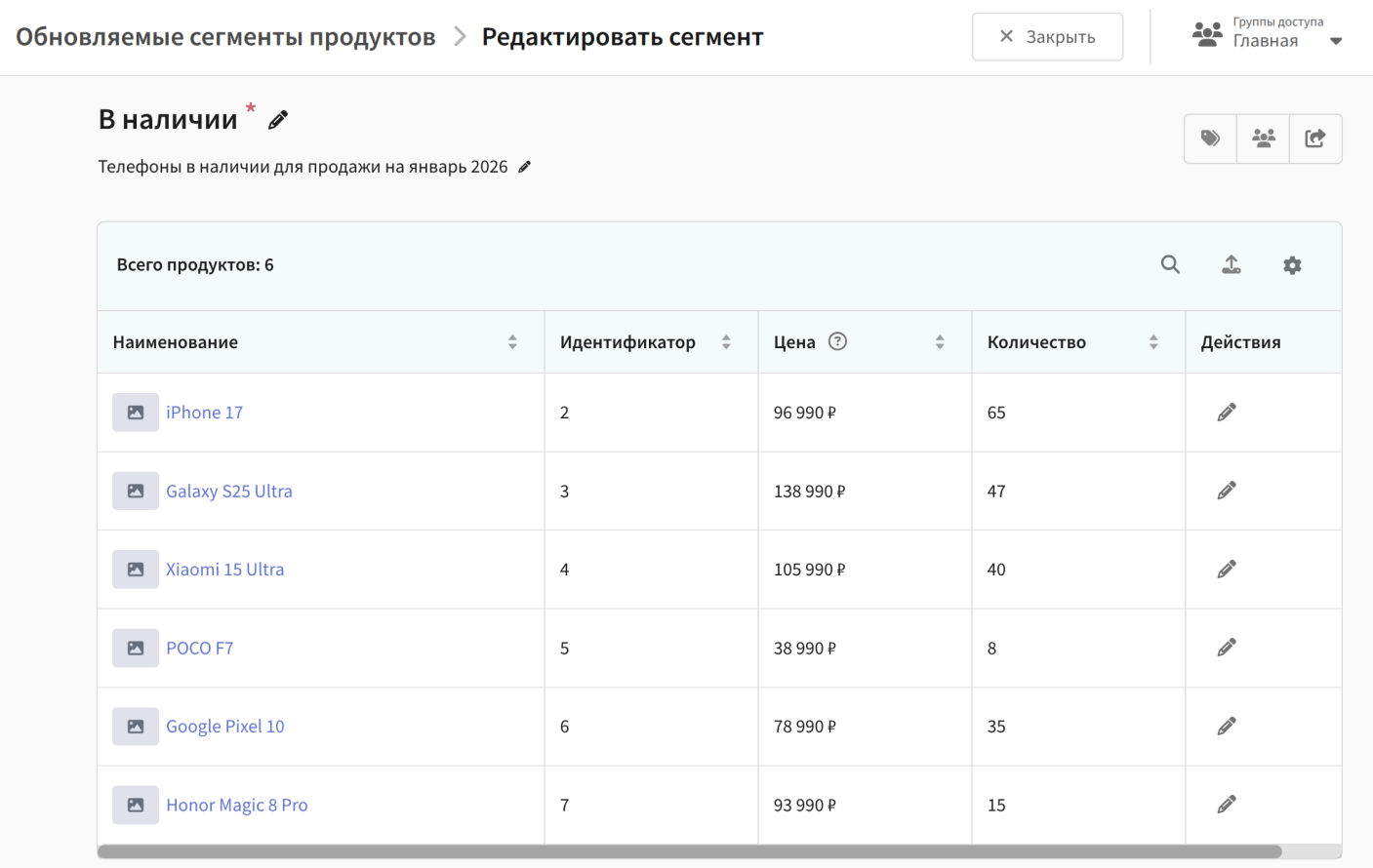
Сегменты продуктов и SKU

TIP: Сегментация в Altcraft

Сегментация продуктов и SKU происходит так же, как и **сегментация профилей** в платформе. Более подробная информация о конструкторе условий и принципах работы с сегментами доступна в этом разделе.

Описание

Сегменты продуктов и SKU — набор продуктов и их модификаторов (SKU), объединенных в группу по определенному признаку. Готовый сегмент можно выгрузить и использовать для анализа данных о ваших продажах. Сегменты бывают двух видов: **статические** и **обновляемые**.

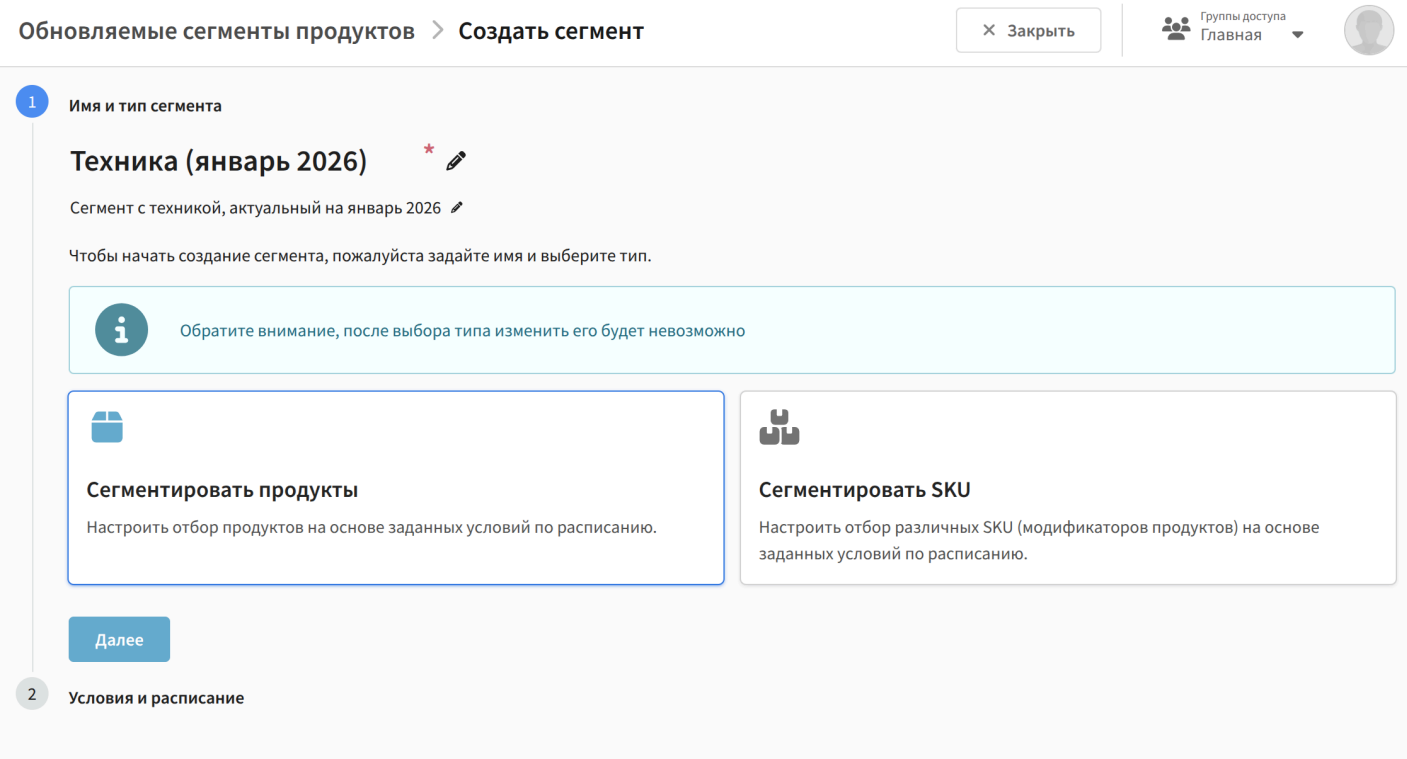


Обновляемые сегменты

Обновляемые сегменты — обновляемые **по расписанию** наборы продуктов и их модификаторов (SKU), отобранных по определенным условиям. Такой сегмент можно также пересчитывать вне расписания самостоятельно.

Для создания обновляемого сегмента перейдите в раздел **Маркет — Обновляемые сегменты**:

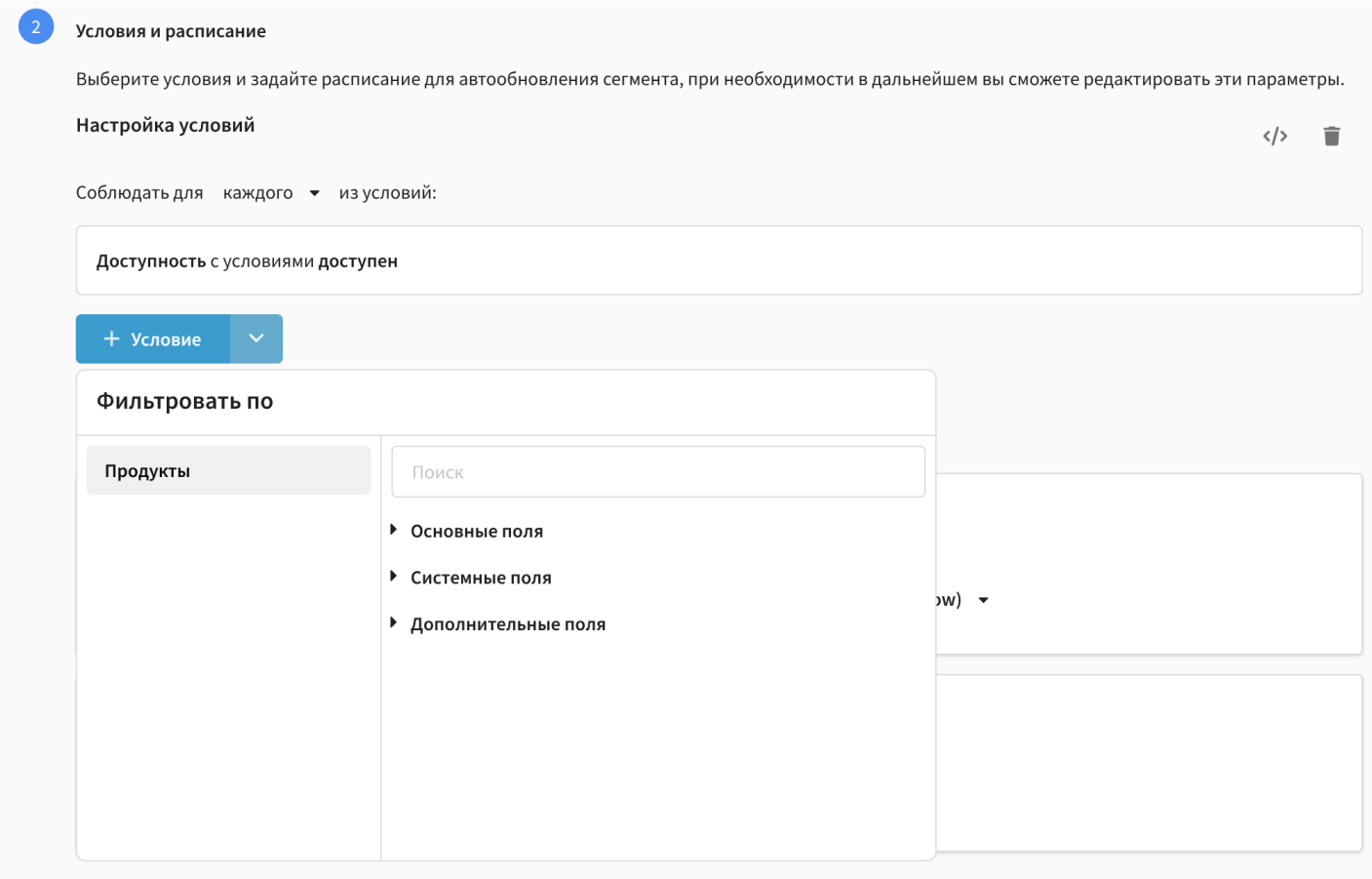
Нажмите “Создать”. Укажите название, описание и выберите сущности для сегментации, **продукты** или **SKU**. После выбора изменить тип сегмента **нельзя**:



Затем откроется блок **“Условия и расписание”**. Выберите условия или группу условий и задайте расписание для автообновления сегмента. После настройки условий внутри группы их можно комбинировать по логике **“соблюдать для каждого из условий”** (И) или **“соблюдать для одного из условий”** (ИЛИ). При необходимости в дальнейшем вы сможете редактировать эти параметры.

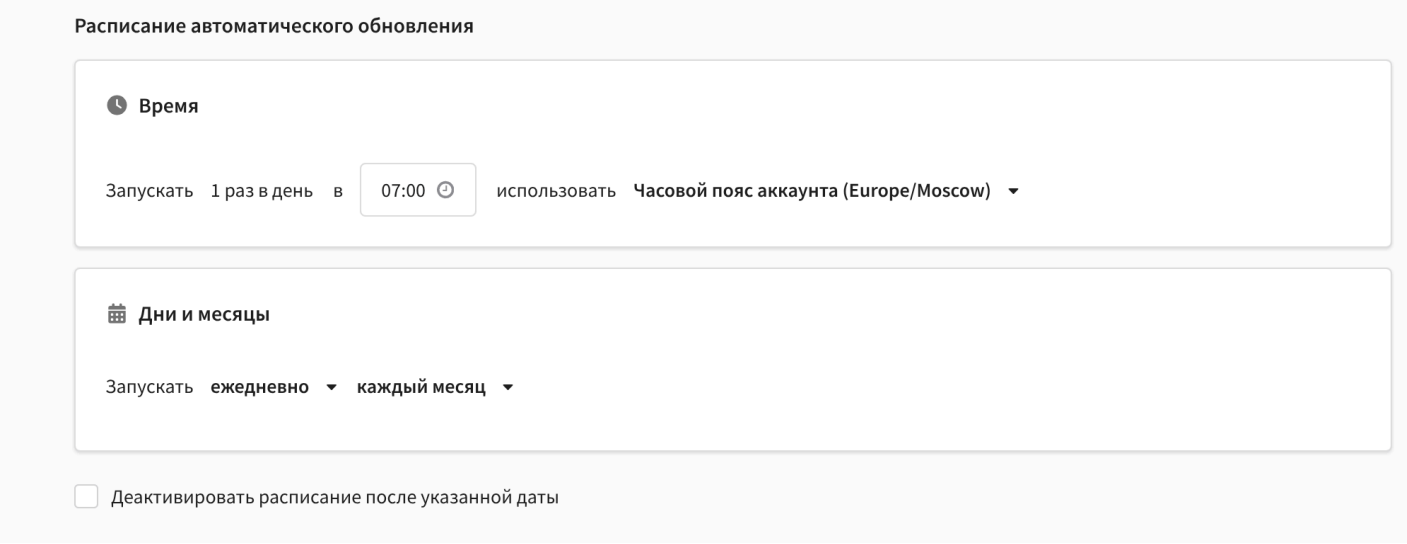
Условия делятся на три группы:

- Основные поля — отбор по наименованию, региону, категории, количеству товаров, сроку годности, цене, доставке, гарантии, доступности, производителю и по тегам.
- Системные поля — отбор по идентификатору, дополнительному (строковому) идентификатору, дате создания и обновления.
- Дополнительные поля — отбор по полям, созданным пользователем.



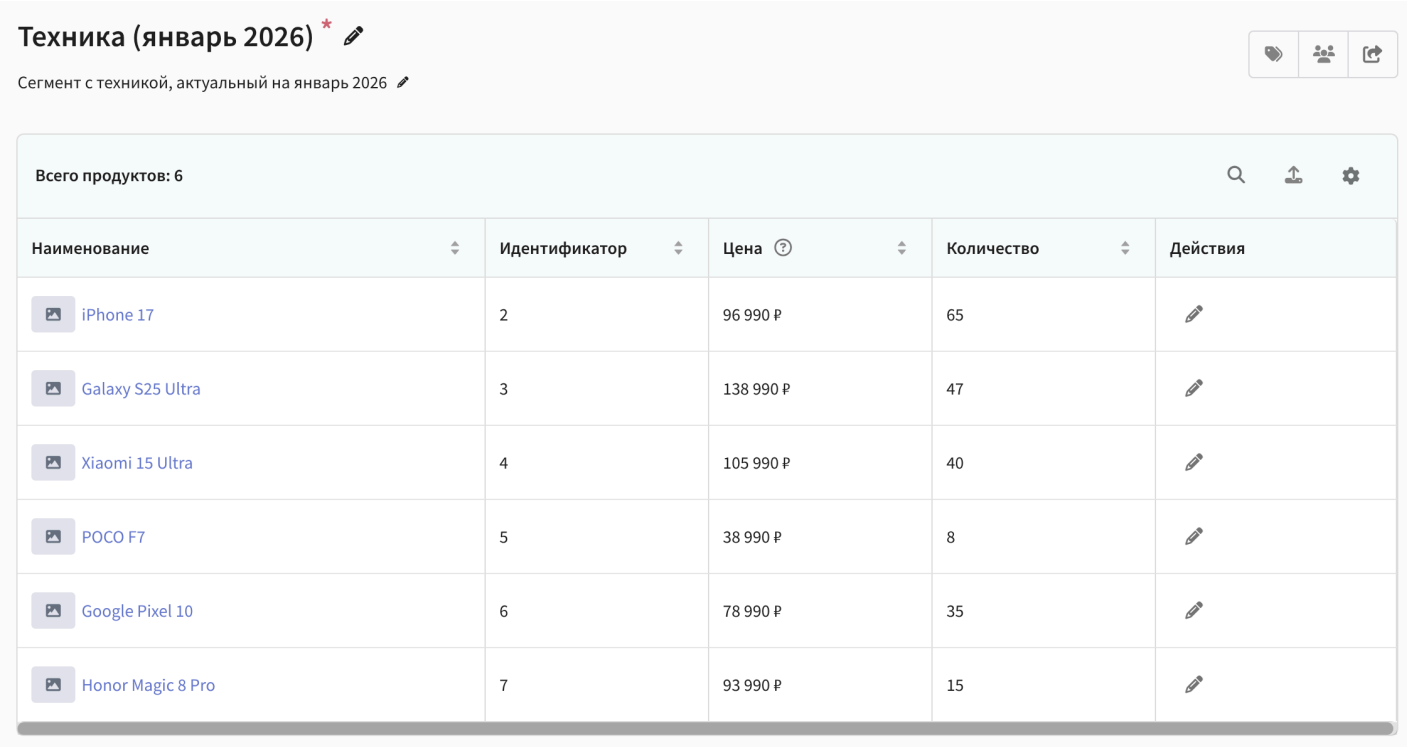
В этом же блоке доступна кнопка **“Скрипт”**, открывающая редактор для ручного написания условий. После настройки условий можно задать ограничение на количество продуктов. Это ограничение применяется к результату выборки **после** отбора по условиям.

Сегмент может обновляться ежедневно либо в указанные дни. Время обновления и часовой пояс также задаются пользователем:



Вы также можете настроить деактивацию сегмента после указанной даты.

После расчёта в окне сегмента отображается таблица со всеми продуктами, входящими в сегмент. В ней отображаются поля “Наименование”, “Идентификатор”, “Цена”, “Количество”, а также кнопка для перехода к редактированию продукта:



Доступны следующие действия для работы с этой таблицей:

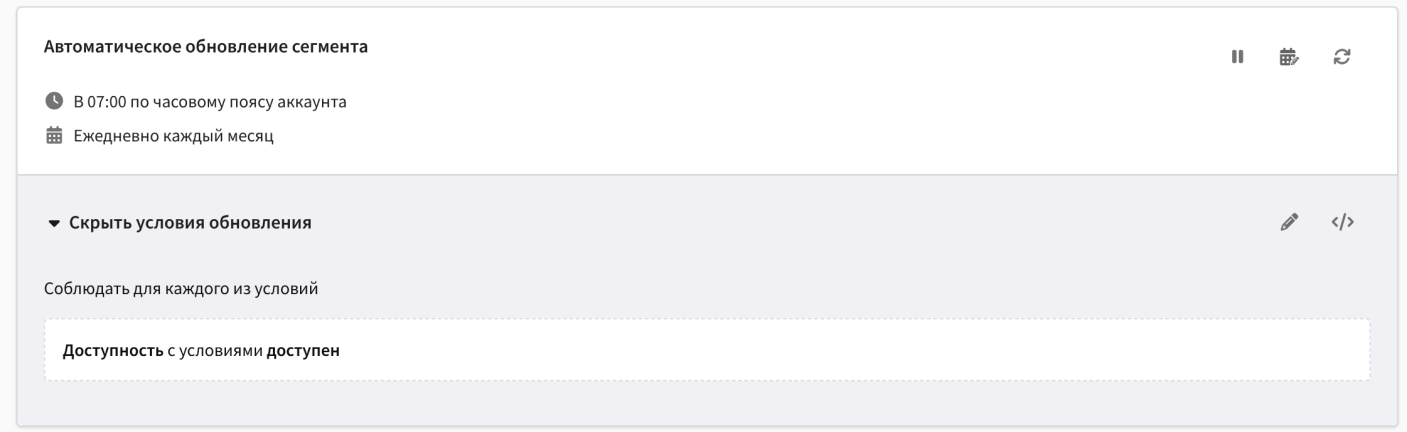
- **Поиск**

  : найти конкретный продукт.
- **Экспортировать продукты**

  : выгрузить отображаемый список во внешний файл.
- **Настройка отображаемых полей**

  : чтобы добавить или убрать колонки в таблице, перетащите нужное **Поле** из боковой колонки **“Добавить поле”** в область **“Отображаемые поля”** или обратно.

Под таблицей продуктов расположено расписание и условия обновления сегмента. Здесь можно **отредактировать** эти условия или пересчитать сегмент **вне расписания**:

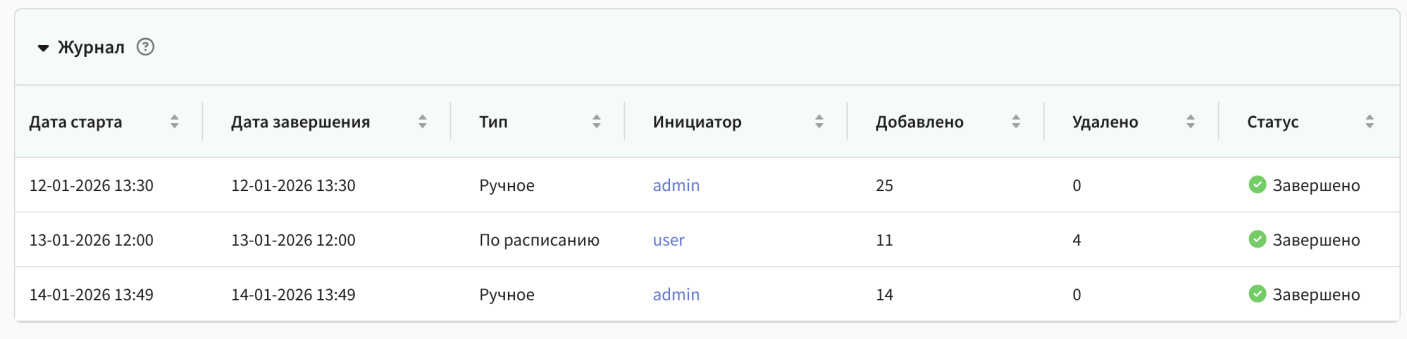


Ниже находится блок **“История обновлений”**. В нём наглядно представлена информация об изменении общего количества продуктов в сегменте за выбранный период. Вы можете указать нужный период (дату

начала и окончания) и сгруппировать данные по дням или неделям.

Ниже истории обновлений находится **“Журнал”**. Это детальный лог всех операций, повлиявших на состав сегмента. По каждой операции отображается:

- **Дата начала**: время запуска процесса сегментации (добавления/удаления по условиям).
- **Дата завершения**: время окончания процесса.
- **Инициатор**: имя пользователя, который инициировал операцию.
- **Добавлено**: количество продуктов, добавленных в результате операции.
- **Удалено**: количество продуктов, удалённых в результате операции.
- **Статус**: текущее состояние операции (например, “Завершено”, “В процессе”, “Ошибка”).



TIP: Действия с сегментом

В верхней части экрана доступно **меню быстрых действий**, с помощью которого можно **назначить теги**, **настроить группы доступа** и **скопировать продукты в другой сегмент**.

Статические сегменты

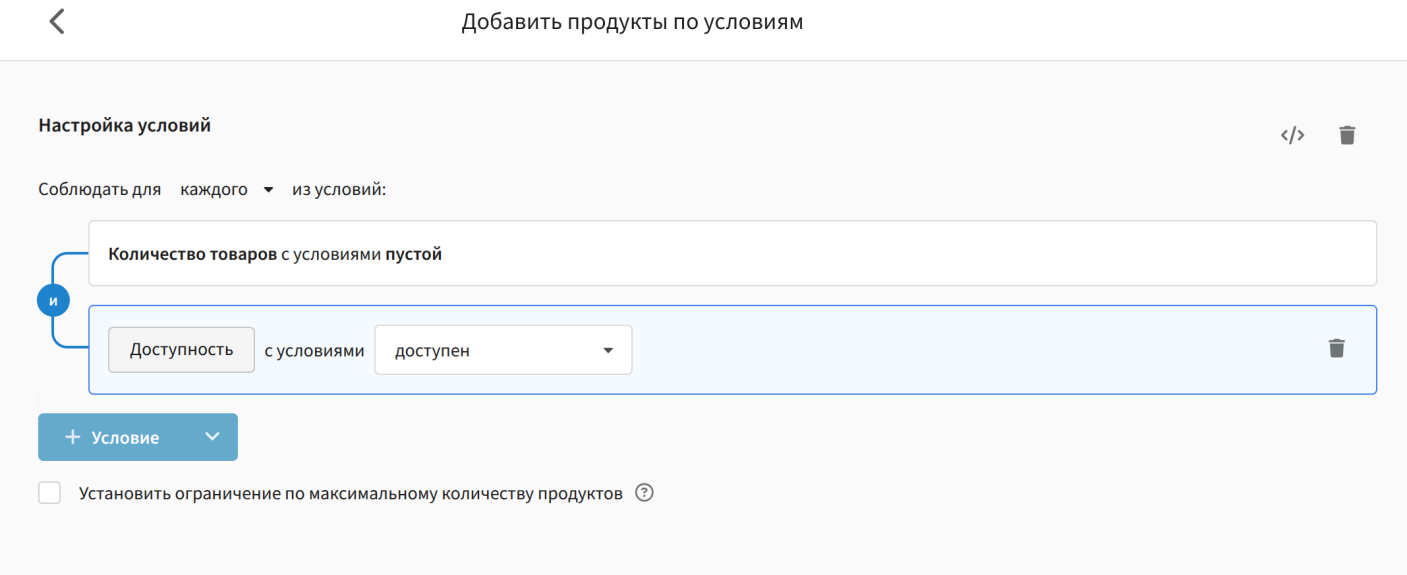
Статические сегменты продуктов — наборы продуктов и их модификаторов (SKU), объединенных в группу **через API** или вручную при помощи **условий отбора**.

Для создания статического сегмента перейдите в раздел **Маркет — Статические сегменты**:

Нажмите “Создать”. Укажите название, описание и выберите сущности для сегментации, **продукты** или **SKU**. После выбора изменить тип сегмента **нельзя**:

Сегмент создан. Чтобы добавить в него продукты, нажмите кнопку “Добавить”, откроется блок **“Условия”**.

Для добавления правила нажмите кнопку **“+ Условие”**. В выпадающем меню доступна опция **“Добавить группу условий”**. После настройки условий внутри группы их можно комбинировать по логике **“соблюдать для каждого из условий”** (И) или **“соблюдать для одного из условий”** (ИЛИ):



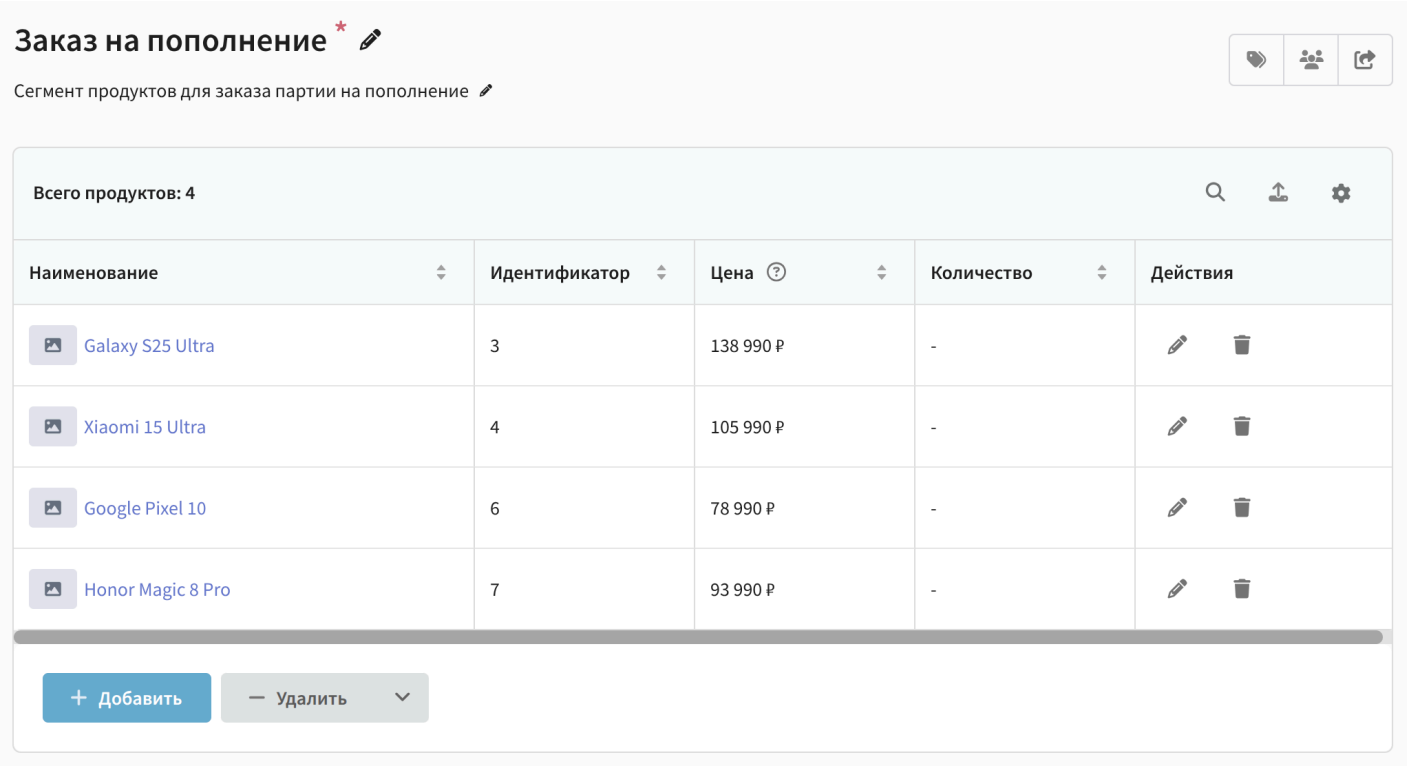
Условия делятся на три группы:

- Основные поля — отбор по наименованию, региону, категории, количеству товаров, сроку годности, цене, доставке, гарантии, доступности, производителю и по тегам;
- Системные поля — отбор по идентификатору, дополнительному (строковому) идентификатору, дате создания и обновления;
- Дополнительные поля — отбор по полям, созданным пользователем.

В этом же блоке доступны кнопки **“Удаление”**, которая полностью очищает все заданные правила, и **“Скрипт”**, открывающая редактор для ручного написания условий. После настройки условий можно задать ограничение на количество продуктов. Это ограничение применяется к результату выборки **после** отбора по условиям.

После нажатия кнопки “Добавить” в сегменте появятся продукты, соответствующие правилам отбора. Для того чтобы удалить продукты из сегмента нажмите **“Удалить”**. Функция удаления аналогична функции добавления продуктов.

После расчёта в окне статического сегмента отображается таблица со всеми продуктами, входящими в сегмент. В ней отображаются поля “Наименование”, “Идентификатор”, “Цена”, “Количество”, а также меню быстрых действий для удаления или переходу к редактированию продукта:



Доступны следующие действия для работы с этой таблицей:

- **Поиск**

  : найти конкретный продукт.
- **Экспортировать продукты**

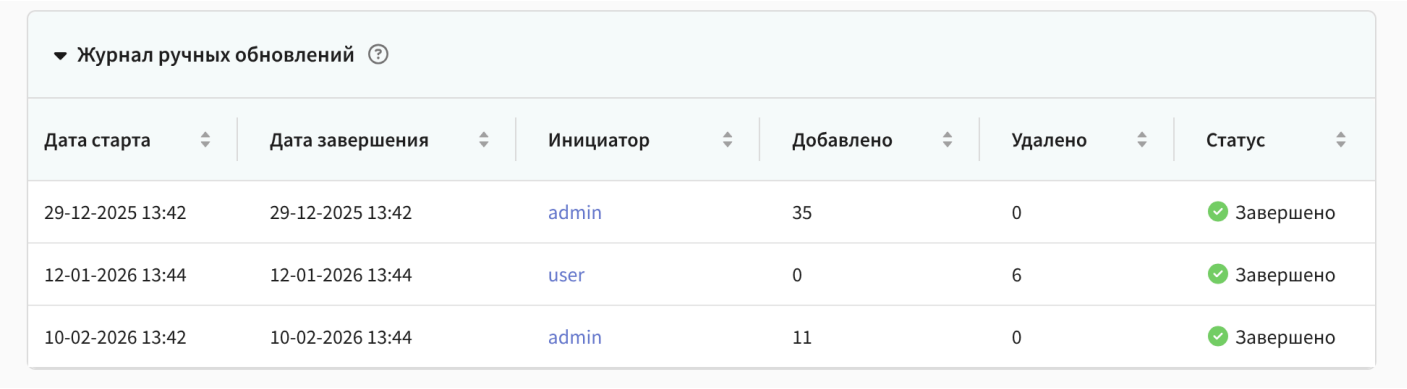
  : выгрузить отображаемый список во внешний файл.
- **Настройка отображаемых полей**

  : чтобы добавить или убрать колонки в таблице, перетащите нужное **Поле** из боковой колонки **“Добавить поле”** в область **“Отображаемые поля”** или обратно.

Ниже находится блок **“История обновлений”**. В нём наглядно представлена информация об изменении общего количества продуктов в сегменте за выбранный период. Вы можете указать нужный период (дату начала и окончания) и сгруппировать данные по дням или неделям.

Ниже истории обновлений находится **“Журнал ручных изменений”**. Это детальный лог всех операций, повлиявших на состав сегмента. По каждой операции отображается:

- **Дата начала**: время запуска процесса сегментации (добавления/удаления по условиям).
- **Дата завершения**: время окончания процесса.
- **Инициатор**: имя пользователя, который инициировал операцию.
- **Добавлено**: количество продуктов, добавленных в результате операции.
- **Удалено**: количество продуктов, удалённых в результате операции.
- **Статус**: текущее состояние операции (например, “Завершено”, “В процессе”, “Ошибка”).



TIP: Действия с сегментом

В верхней части экрана доступно **меню быстрых действий**, с помощью которого можно **назначить теги**, **настроить группы доступа** и **скопировать продукты в другой сегмент**.

Условия

Условия для отбора продуктов и SKU разделены на три группы: **Основные поля**, **Системные поля** и **Дополнительные поля**. Доступный набор операторов зависит от типа выбранного поля.

Основные поля

К основным относятся стандартные поля продукта или SKU: наименование, категория, цена, количество и другие.

Наименование

Строковое поле, содержащее название продукта.

- равно/не равно
- содержит/не содержит
- начинается с
- в списке/не в списке
- в таблице данных/не в таблице данных
- в файле в колонке/не в файле в колонке
- в ответе на post url/не в ответе на post url

Регион

Поле для указания региона продаж или наличия товара.

- существует/не существует
- равно/не равно
- в списке/не в списке

Категории

Поле, содержащее одну или несколько категорий, к которым относится товар.

- существует/не существует
- содержит хотя бы одно значение из списка

- не содержит ни одного значения из списка
- содержит все значения из списка
- не содержит одновременно все значения из списка

Количество товаров

Целое число, отражающее остаток или доступное количество.

- пустой/не пустой
- равно/не равно
- больше чем/меньше чем
- больше или равно/меньше или равно
- в диапазоне

Срок годности

Поле типа *дата*, указывающее срок годности товара.

- равно/не равно
- больше чем/меньше чем
- больше или равно/меньше или равно
- существует/не существует
- в диапазоне

Цена

Числовое поле с текущей ценой продукта.

- существует/не существует
- равно/не равно
- больше чем/меньше чем
- больше или равно/меньше или равно
- в списке/не в списке
- в диапазоне

Старая цена

Числовое поле с ценой до скидки (если применимо).

- существует/не существует
- равно/не равно
- больше чем/меньше чем
- больше или равно/меньше или равно
- в списке/не в списке
- в диапазоне

Доставка

Логическое поле (true/false), указывающее, требуется ли доставка для товара или доступна ли она.

- истина
- ложь

Гарантия

Логическое поле (true/false), указывающее, есть ли на товар гарантия.

- истина
- ложь

Доступность

Поле-переключатель, отражающее статус наличия товара.

- доступен
- не доступен

Производитель

Строковое поле с названием компании-производителя.

- существует/не существует
- равно/не равно
- содержит/не содержит
- в списке/не в списке
- в таблице данных/не в таблице данных
- в файле в колонке/не в файле в колонке
- в ответе на post url/не в ответе на post url

Теги

Поле-список, используемое для гибкой маркировки товаров.

- существует/не существует
- содержит хотя бы одно значение из списка
- не содержит ни одного значения из списка
- содержит все значения из списка
- не содержит одновременно все значения из списка

Системные поля

Поля, заполняемые автоматически платформой или служащие для внутренней идентификации.

Идентификатор

Уникальный числовой ID продукта в системе.

- равно/не равно
- в списке/не в списке
- в таблице данных/не в таблице данных
- в файле в колонке/не в файле в колонке
- в ответе на post url/не в ответе на post url

Дополнительный идентификатор

Строковое поле для хранения внешнего ID (например, артикула из 1С или кода товара).

- равно/не равно
- в списке/не в списке
- в таблице данных/не в таблице данных

- в файле в колонке/не в файле в колонке
- в ответе на post url/не в ответе на post url

Дата создания продукта/SKU

Поле типа **дата**, фиксирующее момент создания записи в платформе.

- равно/не равно
- больше чем/меньше чем
- больше или равно/меньше или равно
- месяц равен/не равен
- месяц тот же что и сегодня
- день месяца равен/не равен/больше чем/меньше чем/такой же что и текущий
- день и месяц равны/не равны/больше чем/меньше чем/тот же что и текущий
- в списке/не в списке
- дата в диапазоне
- дата относительно текущей даты — выборка по моменту в прошлом или будущем (например, “создан за последние 7 дней”)

Дата обновления продукта/SKU

Поле типа **дата**, фиксирующее момент последнего изменения записи.

- равно/не равно
- больше чем/меньше чем
- больше или равно/меньше или равно
- месяц равен/не равен
- месяц тот же что и сегодня
- день месяца равен/не равен/больше чем/меньше чем/такой же что и текущий
- день и месяц равны/не равны/больше чем/меньше чем/тот же что и текущий
- в списке/не в списке
- дата в диапазоне
- дата относительно текущей даты

Пользовательские поля

Вы можете создавать в структуре каталога собственные поля для хранения специфичной информации о продуктах. Для сегментации доступны операторы, соответствующие типу созданного поля (строковые, числовые, дата, булевы и т.д.).

Подготовка YML-файла

Товары в Маркете представлены двумя сущностями: продукты и SKU.

Продукт — это товар или услуга для продажи. **SKU** — это модификатор продукта, т.е. группа товаров, которая выделяется в одном продукте по некоторым параметрам (цвет, размер, вес, модель, комплектация и т.д.).

Для загрузки данных о товарах в платформу используется **файл в формате yml** или **.zip/.tar.gz архив**, содержащий yml-файлы. В случае, если вы настраиваете импорт по расписанию, вы должны указать ссылку на yml-файл.

Формат YML (Yandex Market Language) — это формат данных для товарных платформ, разработанный Яндексом. Ознакомиться с его полной структурой вы можете по этой ссылке.

Структура таких файлов выглядит следующим образом:

```
<?xml version="1.0" encoding="UTF-8"?>
<yml_catalog date="2017-02-05 17:22">
  <shop>
    ...
    <offers>
      ...
    </offers>
    <gifts>
      ...
    </gifts>
    <promos>
      ...
    </promos>
  </shop>
</yml_catalog>
```

Список всех товаров для импорта передаётся в элементе `offers` (предложения магазина). Этот элемент, в свою очередь, содержит элементы `offer`.

TIP

Больше информации об элементе `offer` вы найдёте здесь.

Каждый продукт и SKU описывается в отдельном элементе `offer`. Для элемента указывается атрибут `id`, уникальный для каждого товара:

```
<offer id="9012">
    <!-- данные продукта #1 -->
</offer>
<offer id="9013">
    <!-- данные продукта #2 -->
</offer>
```

Структура файлов для импорта продукта и для импорта SKU практически не отличается. Единственное отличие — при импорте SKU для элемента offer указывается не только `id`, но также `group_id` – идентификатор продукта. Это обусловлено тем, что SKU всегда привязывается к определенному продукту.

```
<offer id="1001" group_id="9012">
    <!-- данные SKU #1 в продукте #1 -->
</offer>
<offer id="1002" group_id="9012">
    <!-- данные SKU #2 в продукте #1 -->
</offer>
```

CAUTION

Обратите внимание, SKU – это различные вариации одного продукта, поэтому значение атрибута `group_id` должно быть одинаковым для всех SKU одного продукта.

Структура элемента offer

Атрибуты offer

Пример записи атрибута:

```
<offer id="iphone12" available="true">
```

Элемент	Пример	Обязательный	Описание
`id`	`apple12_128gb`	Да	Дополнительный идентификатор продукта или SKU.
`group_id`	`iphone12`	Да, если импортируете SKU	Элемент объединяет все SKU, которые являются вариациями одного продукта и должен иметь одинаковое значение.
`available`	`true`	Нет	Товар есть в наличии или нет.

Стандартные поля

Стандартные поля — это поля, которые уже встроены в маркет Altcraft Platform. Они отображаются в карточке продукта или SKU.

Элемент	Пример	Обязательный	Описание
`name`	`Apple iPhone 12 128GB`	Да	Название продукта или SKU
`vendor`	`Apple`	Нет	Название производителя
`url`	`http://best.seller.ru/product_page.asp?pid=12348`	Нет	URL страницы товара на сайте магазина.
`price`	`71990`	Да	Актуальная цена товара.
`oldprice`	`72990`	Нет	Старая цена товара, должна быть выше текущей.
`currencyId`	`RUB`	Да	Идентификатор валюты, в которой указана цена товара. Валюта должна быть задана в настройках маркета, иначе товар не будет импортирован.

`categoryId`	`102`	Нет	Идентификатор категории товара.
			
			Через yml файл можно добавлять категории и подкатегории в маркет. Код ниже добавляется в элемент `yml_catalog` и создаёт категорию “Смартфоны” и две подкатегории “Apple” и “Xiaomi”.
			
			`<categories id=“1”>`
			` <category id=“101” parentId=“1”>Смартфоны</category>`
			` <category id=“102” parentId=“1”>Apple</category>`
			` <category id=“102” parentId=“1”>Xiaomi</category>`
			`</categories>`

`picture`	`https://example.com/img/apple_phone_128.png`	Нет	URL-адрес изображения товара
`delivery`	`false`	Нет	Доставка доступна или нет
`manufacturer_warranty`	`true`	Нет	Гарантия производителя
`barcode`	`0156789012`	Нет	Штрихкод

Дополнительные поля

Дополнительные поля — это стандартные элементы в структуре yml-файла, для которых по умолчанию не предусмотрены поля в маркете Altcraft Platform.

Прежде, чем импортировать товары с этими полями, необходимо предварительно создать дополнительные поля в разделе “Настройки”. Далее при импорте настройте **сопоставление системных полей.**

Элемент	Пример	Обязательный	Описание
`type`	`book`	Нет	Особый тип товара. Является атрибутом элемента offer.
`store`	`true`	Нет	Возможность купить товар без предварительного заказа.
`pickup`	`true`	Нет	Возможность самовывоза из пунктов выдачи (во всех регионах, в которые доставляет магазин).
`vendorCode`	`A1234567B`	Нет	Код производителя для данного товара.
`model`	`3811`	Нет	Модель товара.
`sales_notes`	`Необходима предоплата.`	Нет	Условия продажи товара. Используется, когда необходимо сообщить об ограничениях при заказе (минимальная сумма заказа, минимальное количество товаров или необходимость предоплаты)
`country_of_origin`	`Россия`	Нет	Страна производства товара
`downloadable`	`false`	Нет	Продукт можно скачать.
`adult`	`false`	Нет	Товар имеет отношение к удовлетворению сексуальных потребностей.
`weight`	`10`	Нет	Вес товара
`dimensions`	`8.27/5.31/1.73`	Нет	Габариты товара

Элементы param

Чтобы добавить в маркет поля для элементов `param`, в деталях импорта отметьте галочкой опцию “**Отсутствующие пользовательские поля будут созданы**”.

Для каждого элемента будет создано новое дополнительное поле. Наименованием поля станет значение атрибута `name`.

Элемент	Пример	Обязательный	Описание

param	`<offer>`
	`<!-- элементы, входящие в offer -->`
	`<param name="Размер экрана" unit="дюйм">27</param>`
	`<param name="Материал">алюминий</param>`
	`<param name="Wi-Fi">есть</param>`
	`</offer>`
		Нет	Все важные характеристики товара — цвет, размер, объем, материал, вес, возраст, пол, и т. д. В YML элемент offer может содержать несколько элементов param (один элемент param — одна характеристика).
			Подробнее

Примеры элемента offer

Пример offer для продукта

При импорте файла с таким содержимым в маркет будут добавлены два продукта: `Apple iPhone 12` и `Xiaomi 12X`.

```
<?xml version="1.0" encoding="UTF-8"?>
<yml_catalog date="2017-02-05 17:22">
    <shop>
    <name>BestSeller</name>
    <company>Tne Best inc.</company>
    <url>http://best.seller.ru</url>
    <categories>
      <category id="1">Смартфоны</category>
      <category id="101" parentId="1">Apple</category>
      <category id="102" parentId="1">Xiaomi</category>
    </categories>
    <offers>
      <offer id="iphone12" available="true">
        <name>Apple iPhone 12</name>
        <vendor>Apple</vendor>
        <url>http://best.seller.ru/product_page.asp?pid=12348</url>
        <price>83990</price>
        <oldprice>89990</oldprice>
        <currencyId>RUB</currencyId>
        <categoryId>101</categoryId>
        <picture>https://example.com/img/apple_phone.png</picture>
        <delivery>true</delivery>
        <manufacturer_warranty>true</manufacturer_warranty>
        <barcode>0156789012</barcode>
        <dimensions>8.27/5.31/1.73</dimensions>
        <weight>135</weight>
        <param name="Камера">Dual 12MP</param>
        <param name="ОС">iOS 14</param>
          </offer>
      <offer id="xiaomi12" available="true">
                <name>Xiaomi 12X</name>
        <vendor>Xiaomi</vendor>
        <url>http://best.seller.ru/product_page.asp?pid=12345</url>
        <price>62990</price>
        <oldprice>69990</oldprice>
        <currencyId>RUB</currencyId>
        <categoryId>102</categoryId>
        <picture>https://example.com/img/xiaomi_phone.png</picture>
        <delivery>true</delivery>
        <manufacturer_warranty>true</manufacturer_warranty>
        <barcode>0123456789379</barcode>
        <dimensions>152,7/69,9/8,16</dimensions>
        <weight>176</weight>
        <param name="Камера">50MP</param>
        <param name="ОС">Snapdragon 870</param>
          </offer>
    </offers>
  </shop>
</yml_catalog>
```

```
</yml_catalog>
```

Пример offer для SKU

При импорте из этого файла будут добавлены два SKU для продукта `Apple iPhone 12`: `Apple iPhone 12 128GB` и `Apple iPhone 12 256GB`.

```
<?xml version="1.0" encoding="UTF-8"?>
<yml_catalog date="2017-02-05 17:22">
    <shop>
    <name>BestSeller</name>
    <company>Tne Best inc.</company>
    <url>http://best.seller.ru</url>
    <offers>
      <offer id="apple12_128gb" group_id="iphone12" available="true">
        <name>Apple iPhone 12 128GB</name>
        <vendor>Apple</vendor>
        <url>http://best.seller.ru/product_page.asp?pid=12358</url>
        <price>71990</price>
        <oldprice>72990</oldprice>
        <currencyId>RUB</currencyId>
        <categoryId>101</categoryId>
        <picture>https://example.com/img/apple_phone_128.png</picture>
        <delivery>true</delivery>
        <manufacturer_warranty>true</manufacturer_warranty>
        <barcode>0156789012</barcode>
        <dimensions>8.27/5.31/1.73</dimensions>
        <weight>135</weight>
        <param name="Камера">Dual 12MP</param>
        <param name="ОС">iOS 14</param>
          </offer>
      <offer id="apple12_256gb" group_id="iphone12" available="true">
                <name>Apple iPhone 12 256GB</name>
        <vendor>Apple</vendor>
        <url>http://best.seller.ru/product_page.asp?pid=12359</url>
        <price>83990</price>
        <oldprice>89990</oldprice>
        <currencyId>RUB</currencyId>
        <categoryId>102</categoryId>
        <picture>https://example.com/img/apple_phone_256.png</picture>
        <delivery>true</delivery>
        <manufacturer_warranty>true</manufacturer_warranty>
        <barcode>0156789012</barcode>
        <dimensions>8.27/5.31/1.73</dimensions>
        <weight>135</weight>
        <param name="Камера">Dual 12MP</param>
        <param name="ОС">iOS 14</param>
          </offer>
    </offers>
  </shop>
</yml_catalog>
```

Лояльность

В этом разделе

Программы лояльности

	[**Программы лояльности**](/user-guide/loyalty-programs/loyalties) — дополнительный модуль для создания систем поощрения клиентов за их действия: покупки, регистрацию, просмотр контента и другие взаимодействия с брендом.

Промокоды

	Промокоды могут предоставлять вашим клиентам доступ к специальным предложениям, акциям, скидкам. Промокоды также могут использоваться в шаблонах сообщений и сценариях автоматизации. Также управление промокодами доступно по API.

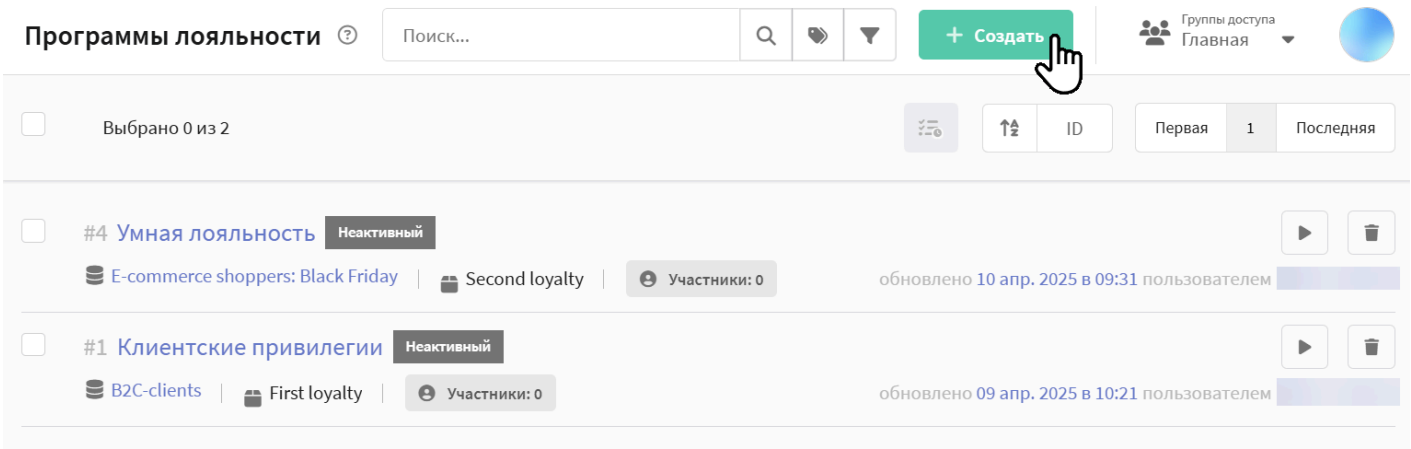
Создание и настройка программы лояльности

Программа лояльности — это система поощрения клиентов за целевые действия: покупки, регистрацию, просмотр контента и другое. Клиенты получают баллы, которые могут потратить на скидки, фильмы, тарифы и услуги. В программе предусмотрены уровни (бронза, серебро и т.д.), отчетность, правила начисления и сгорания баллов, а также интеграции с другими сервисами и каналами коммуникации.

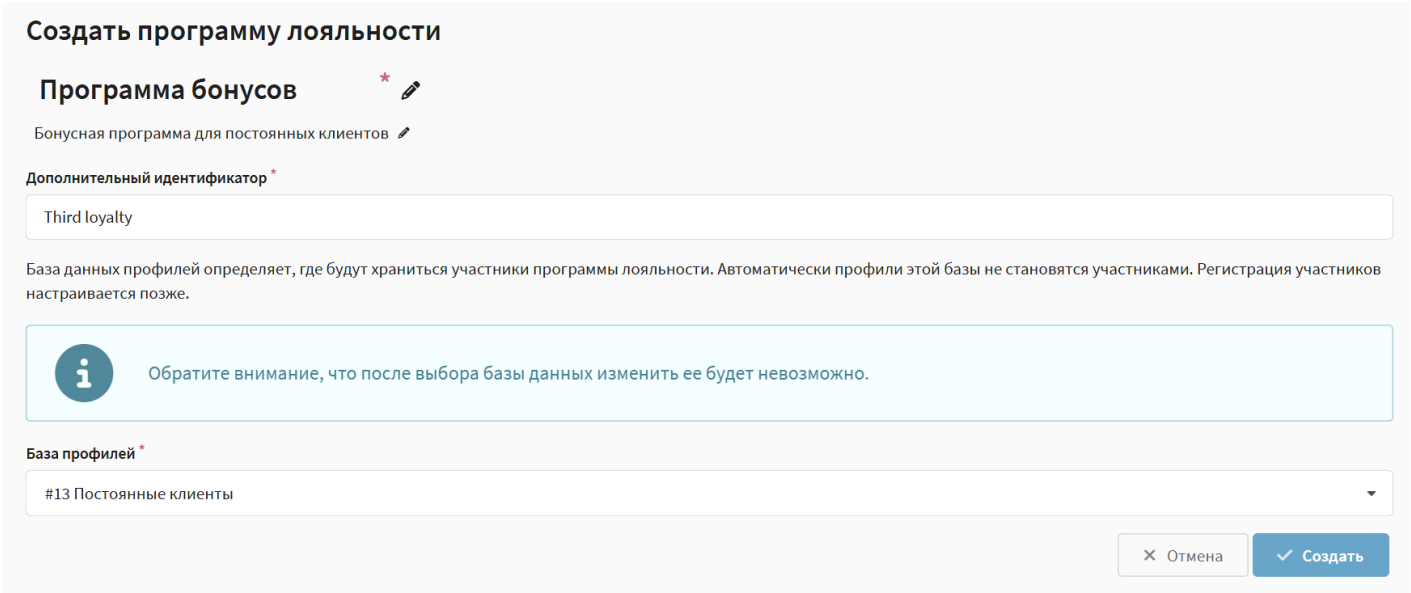
Создание программы лояльности

	Чтобы перейти к управлению программами лояльности, в главном меню перейдите в раздел **Лояльность — Программы лояльности**. Откроется список с существующими программами лояльности. Для редактирования программы нажмите на ее заголовок.

Чтобы начать новую программу лояльности, вверху страницы нажмите кнопку **+ Создать**:



Перед вами появится окно настройки программы лояльности. Укажите **название** вашей программы и ее **дополнительный идентификатор**. При необходимости добавьте **описание**:



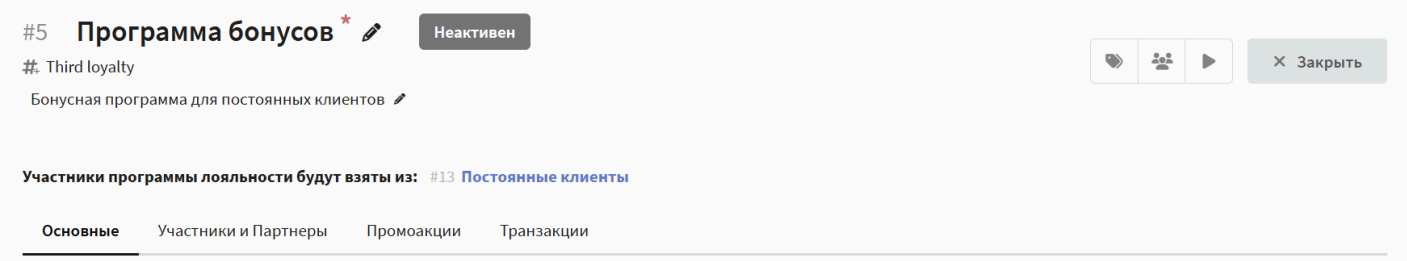
Выберите **базу данных**, из которой будут браться участники.

CAUTION

Имейте в виду:
после выбора базу данных сменить уже **нельзя**;
в одной базе данных одновременно может быть активна **только одна** программа лояльности.

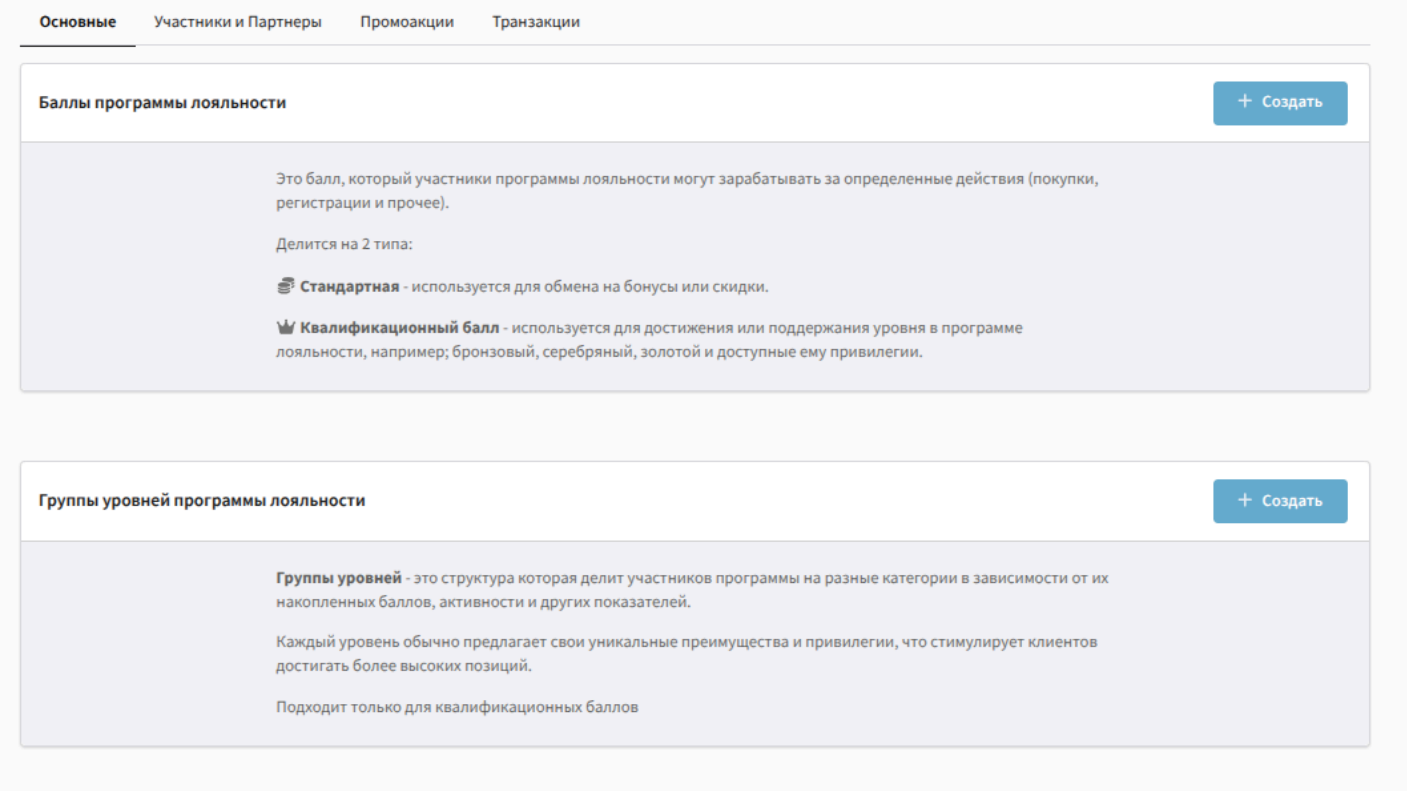
Редактирование программы лояльности

После создания вы попадете в **редактор программы лояльности**. Он состоит из нескольких вкладок: **Основные**, **Участники и партнеры**, **Промоакции** и **Транзакции**:



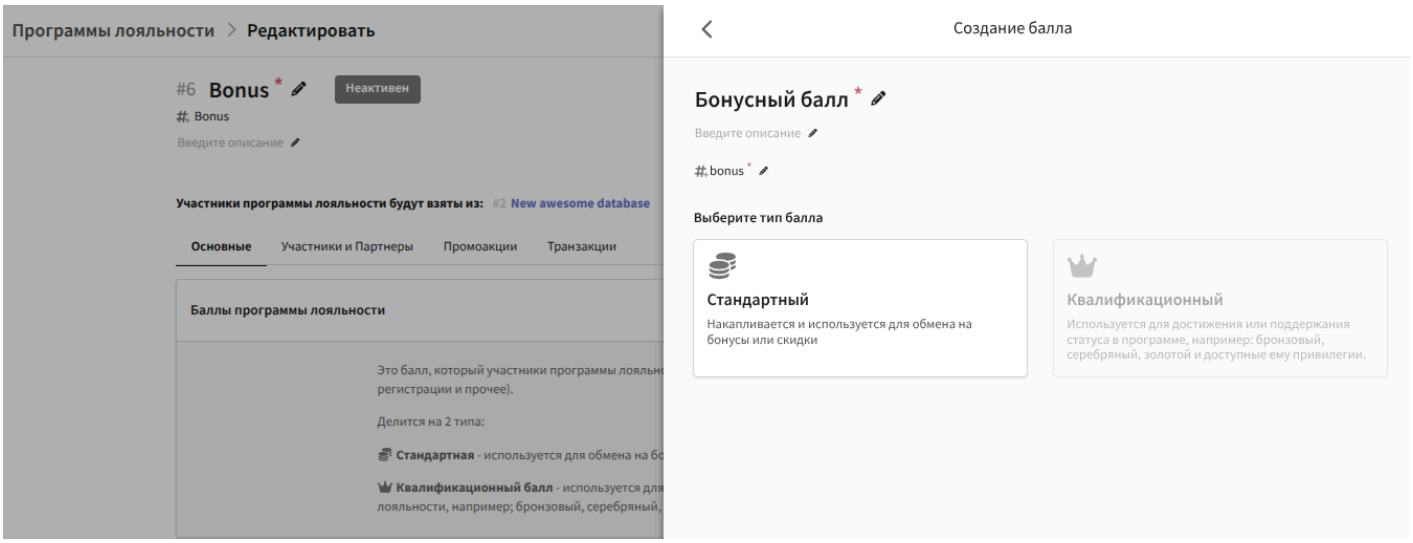
Основные

Раздел **Основные** предназначен для базовой настройки структуры программы лояльности. Здесь задаются ключевые параметры, которые определяют правила начисления, использования и отслеживания баллов, а также механизмы продвижения клиентов по уровням привилегий.



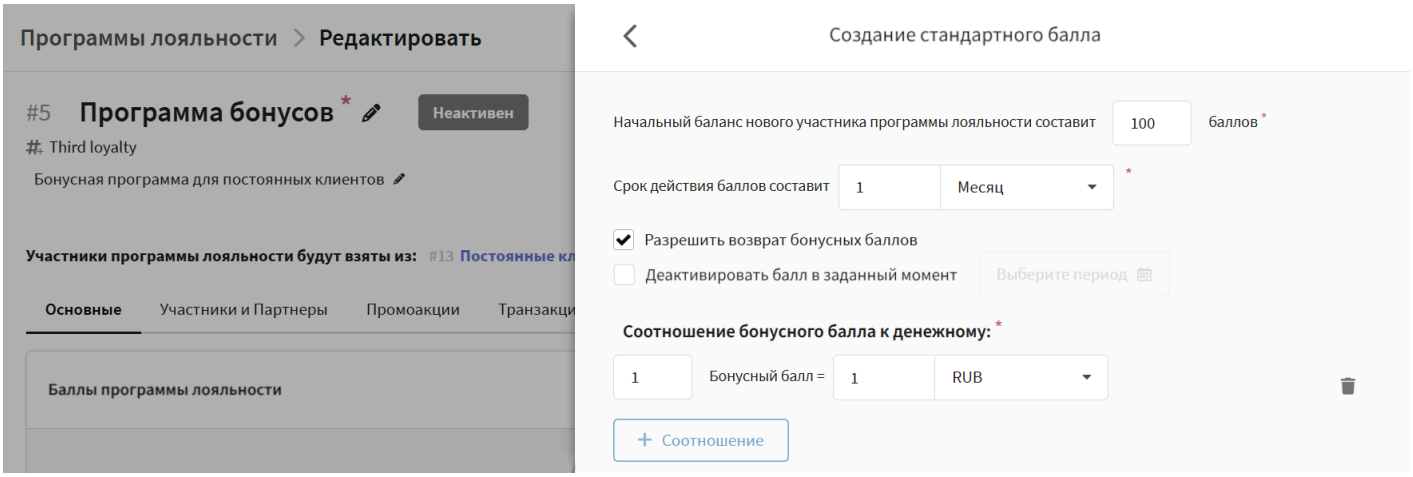
Баллы программы лояльности

Баллы программы лояльности — этот блок отвечает за создание и управление типами баллов, которые используются в рамках программы лояльности. Укажите имя и дополнительный идентификатор ваших баллов, при необходимости добавьте описание.



Баллы могут быть двух типов:

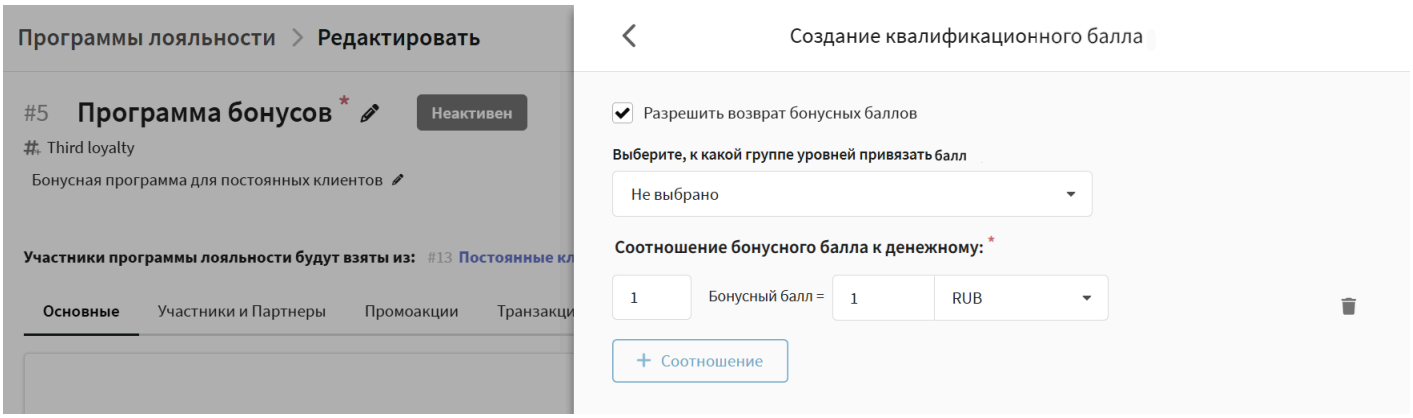
Стандартный балл — начисляемые на бонусный счет баллы, с помощью которых клиент в дальнейшем сможет полностью или частично оплатить покупки.



При создании стандартного балла можно задать следующие параметры:

- **Начальный баланс нового участника программы лояльности** — указывается количество бонусных баллов, которое будет автоматически начислено новому участнику при его вступлении в программу. Значение задается в баллах.
- **Срок действия баллов** — определяет, как долго начисленные баллы будут действительными. Можно задать срок в днях, неделях, месяцах или годах. После истечения указанного срока баллы аннулируются.
- **Разрешить возврат бонусных баллов** — позволяет возвращать бонусные баллы участнику, например, при возврате товара или отмене услуги.
- **Деактивировать балл в заданный момент** — устанавливает дату, после которой балл перестает быть активным (не начисляется и не используется). Полезно для временных акций или сезонных программ.
- **Соотношение бонусного балла к денежному** — обязательный параметр, определяющий, сколько бонусных баллов соответствует определенной денежной сумме. Например: 1 балл = 1 рубль или 10 баллов = 1 доллар. Это соотношение используется при расчете скидок или обмена баллов на товары/услуги.

Квалификационный балл — предназначен для оценки активности клиента и определения его уровня в программе лояльности (бронзовый, серебряный, золотой и т.п.).



При создании квалификационного балла можно задать следующие параметры:

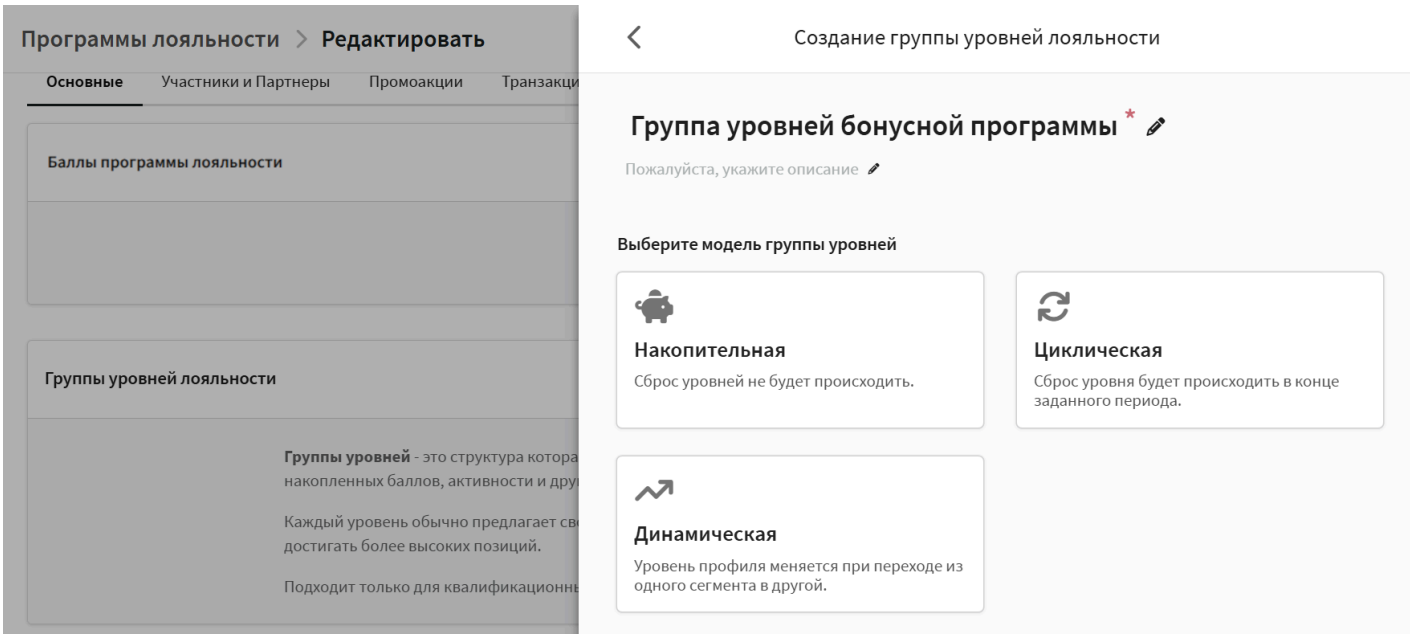
- **Разрешить возврат бонусных баллов** — позволяет возвращать бонусные баллы участнику, например, при возврате товара или отмене услуги.
- **Привязка к группе уровней** — позволяет связать балл с определенной группой уровней лояльности (например: Bronze, Silver, Gold). Это может быть использовано для настройки разных правил начисления или курса в зависимости от уровня клиента.
- **Соотношение бонусного балла к денежному** — обязательный параметр, определяющий, сколько бонусных баллов соответствует определенной денежной сумме. Например: 1 балл = 1 рубль или 10 баллов = 1 доллар. Это соотношение используется при расчете скидок или обмена баллов на товары/услуги.

Не забудьте **активировать** созданный вами балл. При необходимости, его можно клонировать и на его основе создать другой балл.

Группы уровней лояльности

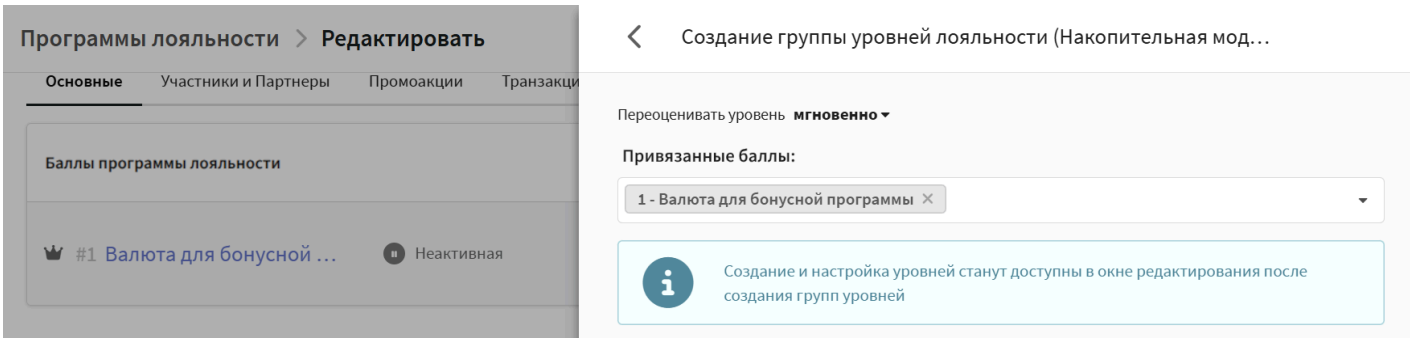
Группы уровней лояльности — это механизм, который позволяет сегментировать участников программы лояльности на различные уровни или статусы в зависимости от их активности, накопленных баллов или других показателей.

Укажите имя и дополнительный идентификатор вашей группы уровней лояльности, при необходимости добавьте описание:



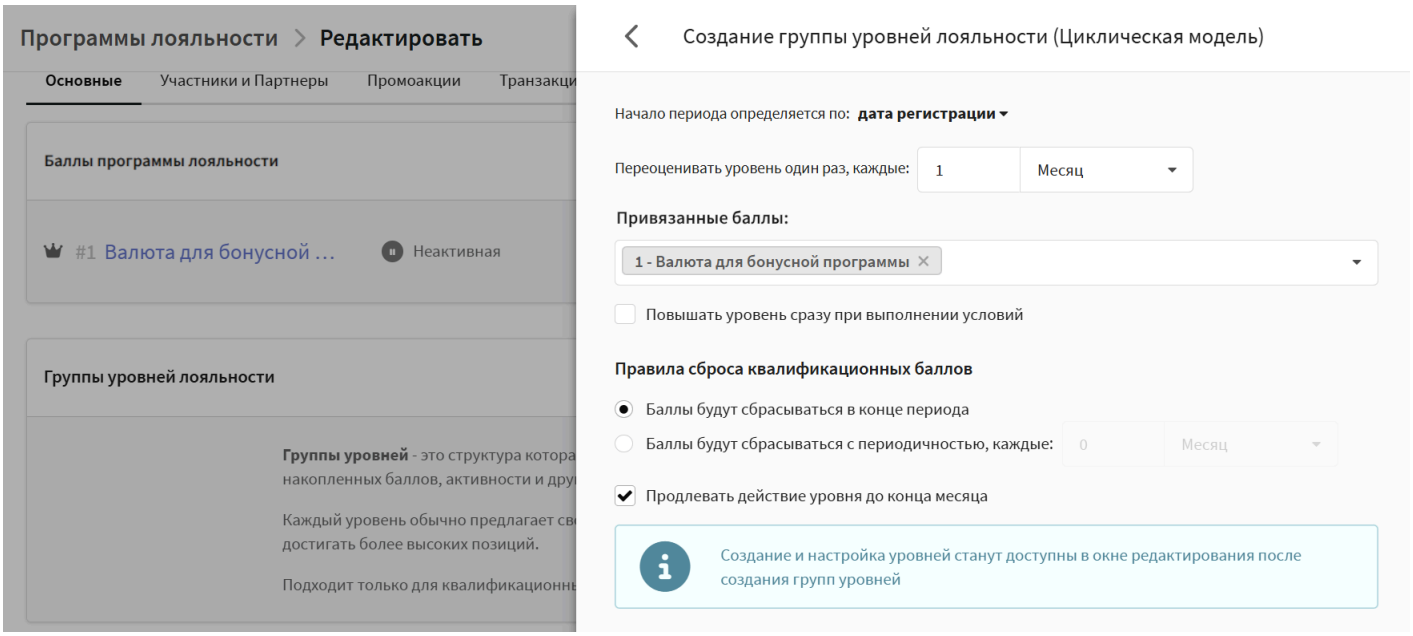
Доступно несколько моделей групп уровней лояльности:

- **Накопительная модель** — участник программы сохраняет достигнутый уровень навсегда. После того как пользователь накапливает достаточно баллов или выполняет необходимые условия, он получает статус, который остается с ним и не сбрасывается со временем. Это удобно, если программа нацелена на долгосрочную мотивацию и поощрение постоянства.



Укажите **балл**, будет привязан к данной модели.

- **Циклическая модель** — достигнутый участником уровень действует лишь в течение установленного периода. По его завершении система автоматически пересматривает статус клиента: если он не подтвердил свою активность или не выполнил нужные условия, уровень может быть понижен.



Настройте модель:

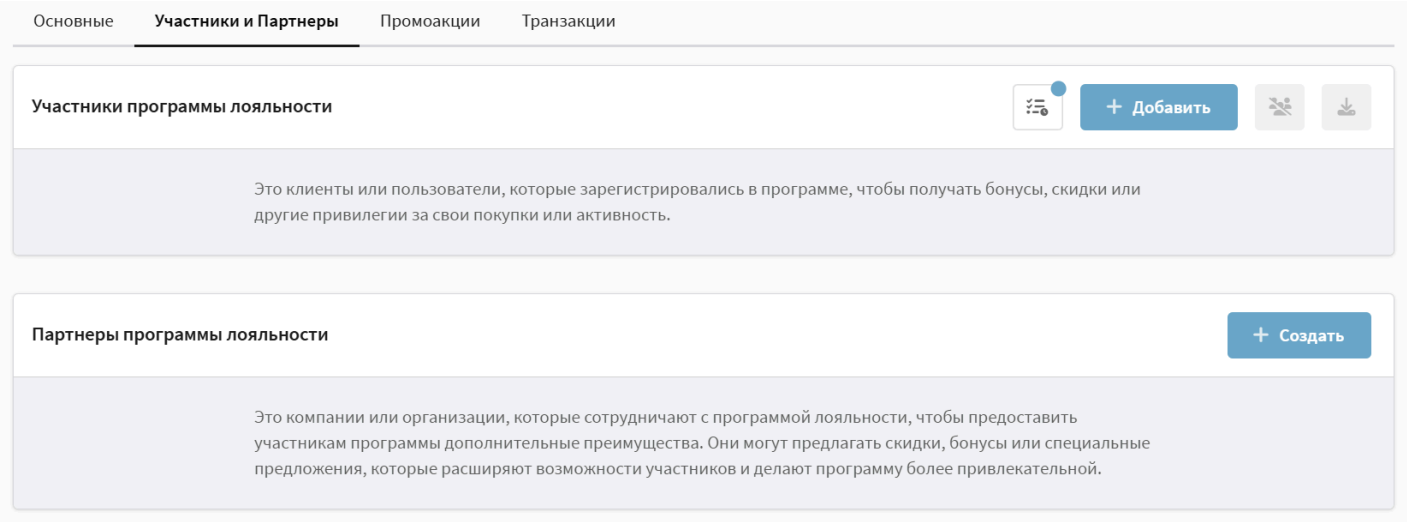
- Задайте **интервал**, через который будет пересматриваться текущий уровень пользователя;
- Укажите **балл**, будет привязан к данной модели.
- Поставьте галочку **“Повышать уровень сразу при выполнении условий”**, тогда участник поднимется на новый уровень сразу после достижения необходимых показателей, не дожидаясь окончания текущего цикла;
- Установите **Правила сброса квалификационных баллов**, можно выбрать сброс в конце каждого периода или через заданные интервалы (например, каждые 6 месяцев);
- Также можно включить опцию **“Продлевать действие уровня до конца месяца”** — уровень сохранится до завершения текущего календарного месяца, даже если участник не выполнил условия для его поддержания;

- **Динамическая модель** — уровень участника напрямую зависит от его текущих характеристик или активности. Как только изменяются условия, по которым сегментируются клиенты, происходит автоматическое изменение уровня.

После установки основных настроек перейдите в раздел **Участники и партнеры**.

Участники и партнеры

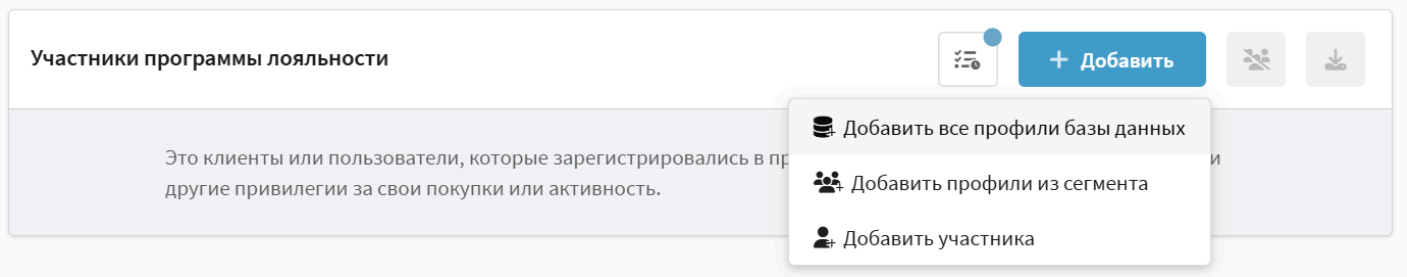
Раздел **Участники и партнеры** предназначен для управления базой участников программы лояльности (конечных пользователей) и партнеров (компаний, предоставляющих дополнительные преимущества участникам).



Данный раздел включает следующие элементы:

Участники программы лояльности — участниками являются клиенты, зарегистрированные в системе лояльности. Они получают баллы за выполнение определенных целевых действий (например, оплата услуг, просмотр контента, подтверждение данных), и могут использовать накопленные баллы для получения скидок, покупки контента или активации дополнительных опций.

Чтобы пополнить список участников нажмите кнопку **+ Добавить**, доступно 3 варианта добавления профилей:



1. Добавить все профили базы данных — массовая регистрация всех клиентов, находящихся в текущей базе.

2. Добавить профили из сегмента — позволяет выбрать ограниченную группу пользователей, например, активных клиентов, недавно оплативших услугу и т.п.

3. Добавить участника — ручная регистрация одного клиента.

INFO

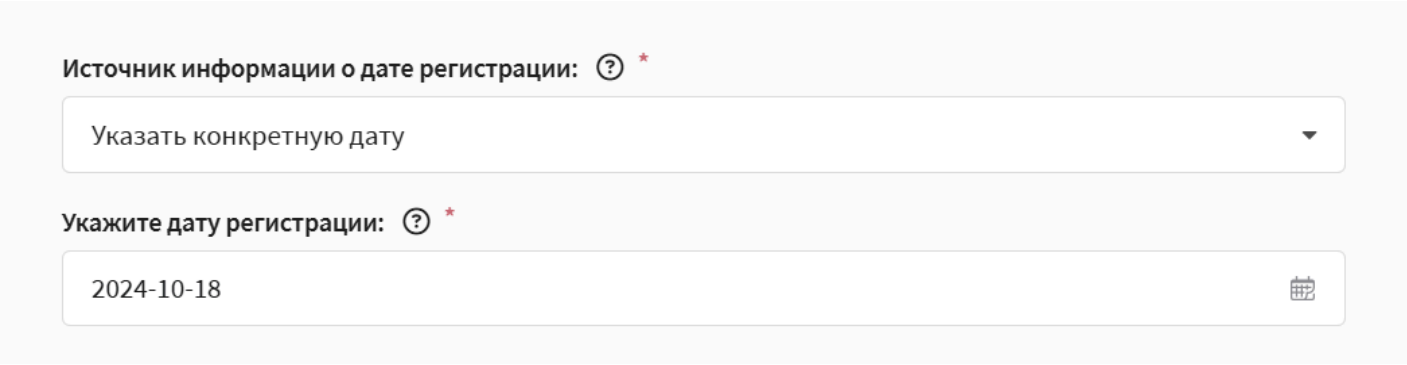
Дополнительно доступны следующие способы пополнения списка участников:

- Автоматизация через узел “Участие в лояльности” — автоматическое добавление клиентов в рамках сценариев при выполнении заданных условий, например после отправки формы.

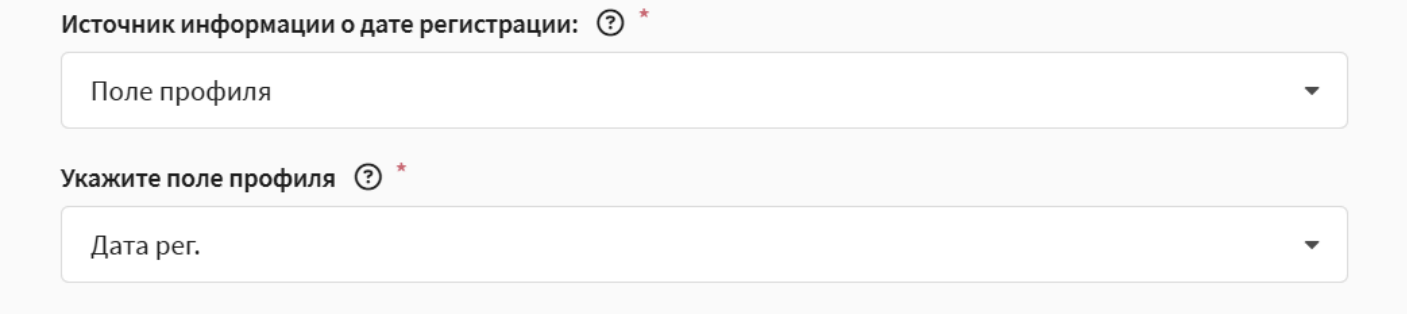
- Интеграция через API — добавление участников программным способом с использованием API-вызова. Подходит для подключения внешних систем и сервисов.

Укажите **источник информации о дате регистрации**.

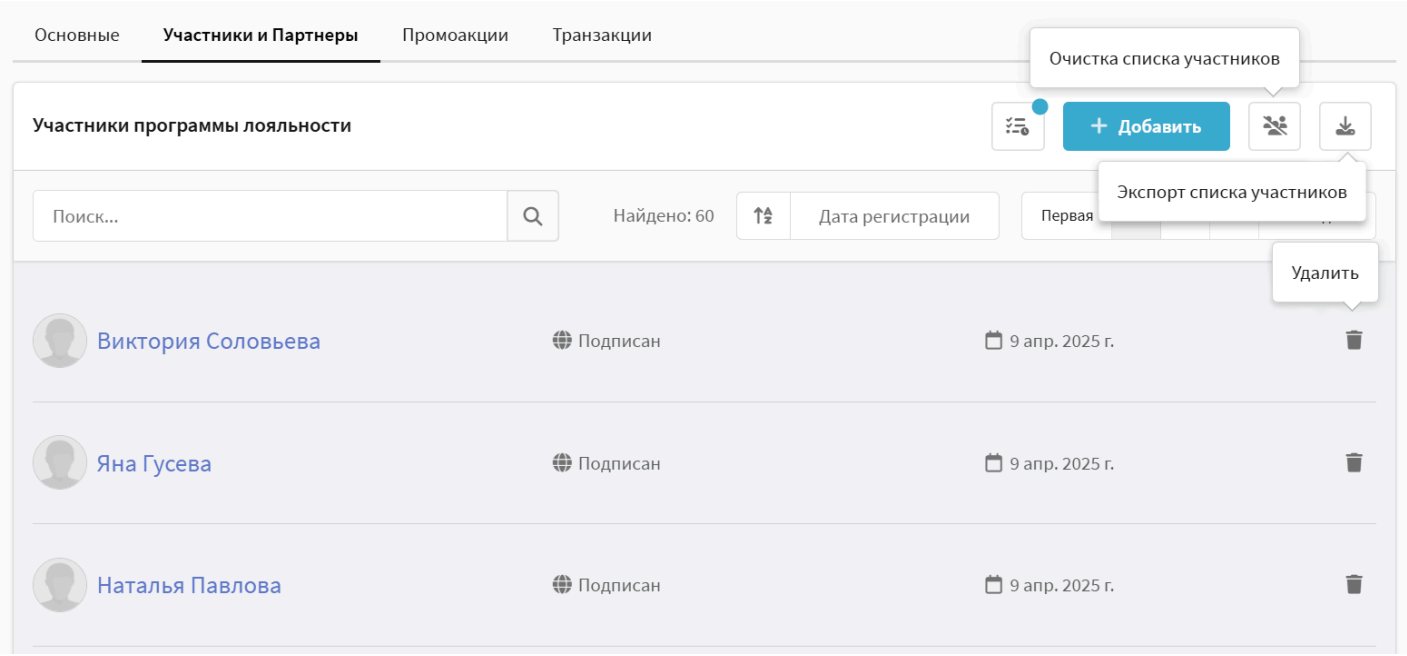
Им может быть **конкретная дата**:



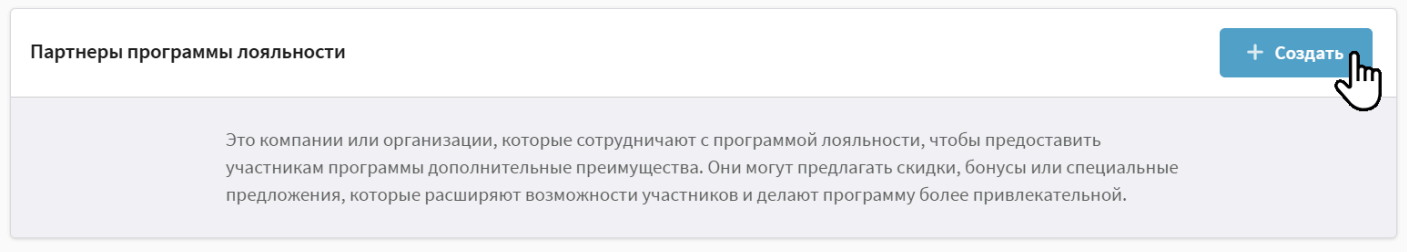
Или **поле профиля**:



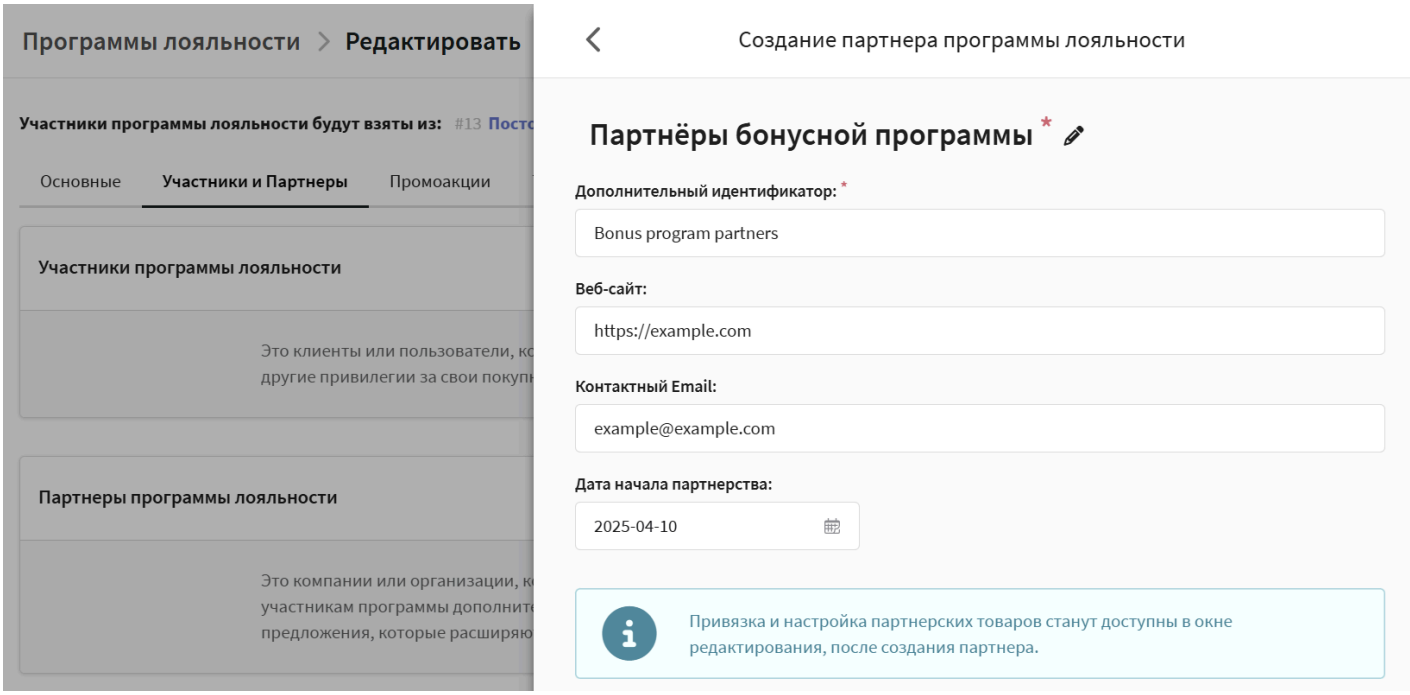
После того как участники добавлены, вы можете удалить отдельных участников или сразу всех, воспользовавшись функцией **“Очистка списка участников”**, а также экспортировать список участников, нажав на одноименную кнопку.



Партнеры программы лояльности — раздел предназначен для управления организациями и компаниями, которые сотрудничают с программой лояльности с целью расширения ее возможностей. Партнеры предоставляют участникам дополнительные привилегии — от скидок до эксклюзивных предложений.



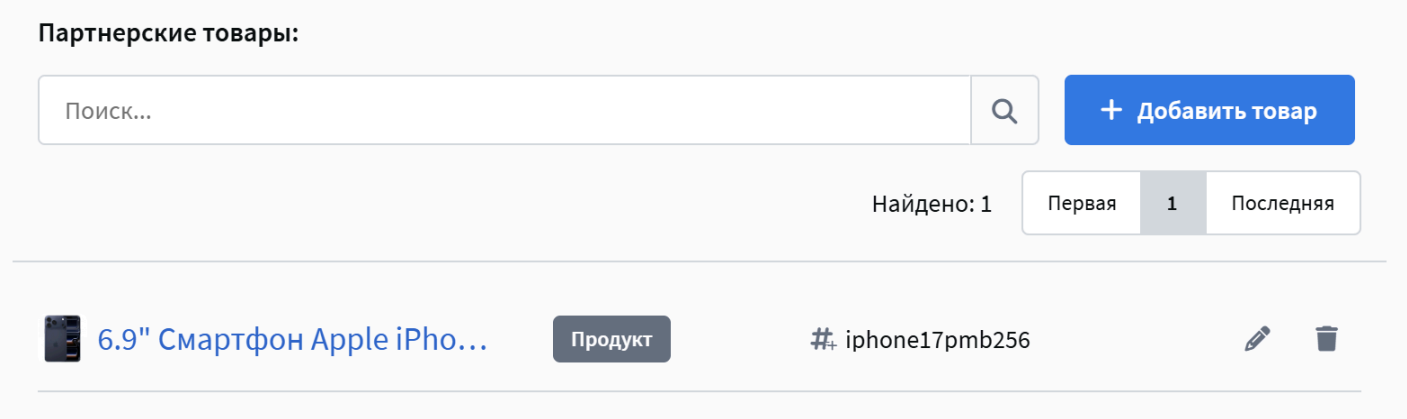
Чтобы добавить нового партнера, нажмите кнопку **+ Создать**. Перед вами появится окно настройки партнера:



Укажите **имя** и **дополнительный идентификатор** партнера программы лояльности. По желанию добавьте ссылку на **официальный сайт** партнера для справочной информации, укажите **контактный email** вашего партнера и установите **дату начала партнерства**.

После создания и сохранения партнера вы можете привязать **Партнёрские товары** из маркета, к которым будут применяться партнерские условия.

Для каждого товара настраивается **скидка в процентах**, которая будет предоставляться участникам программы лояльности при покупке партнерских товаров. Эти настройки используются в промоакциях с бенефитом **“Скидка от партнера”** и учитываются при расчете заказа.



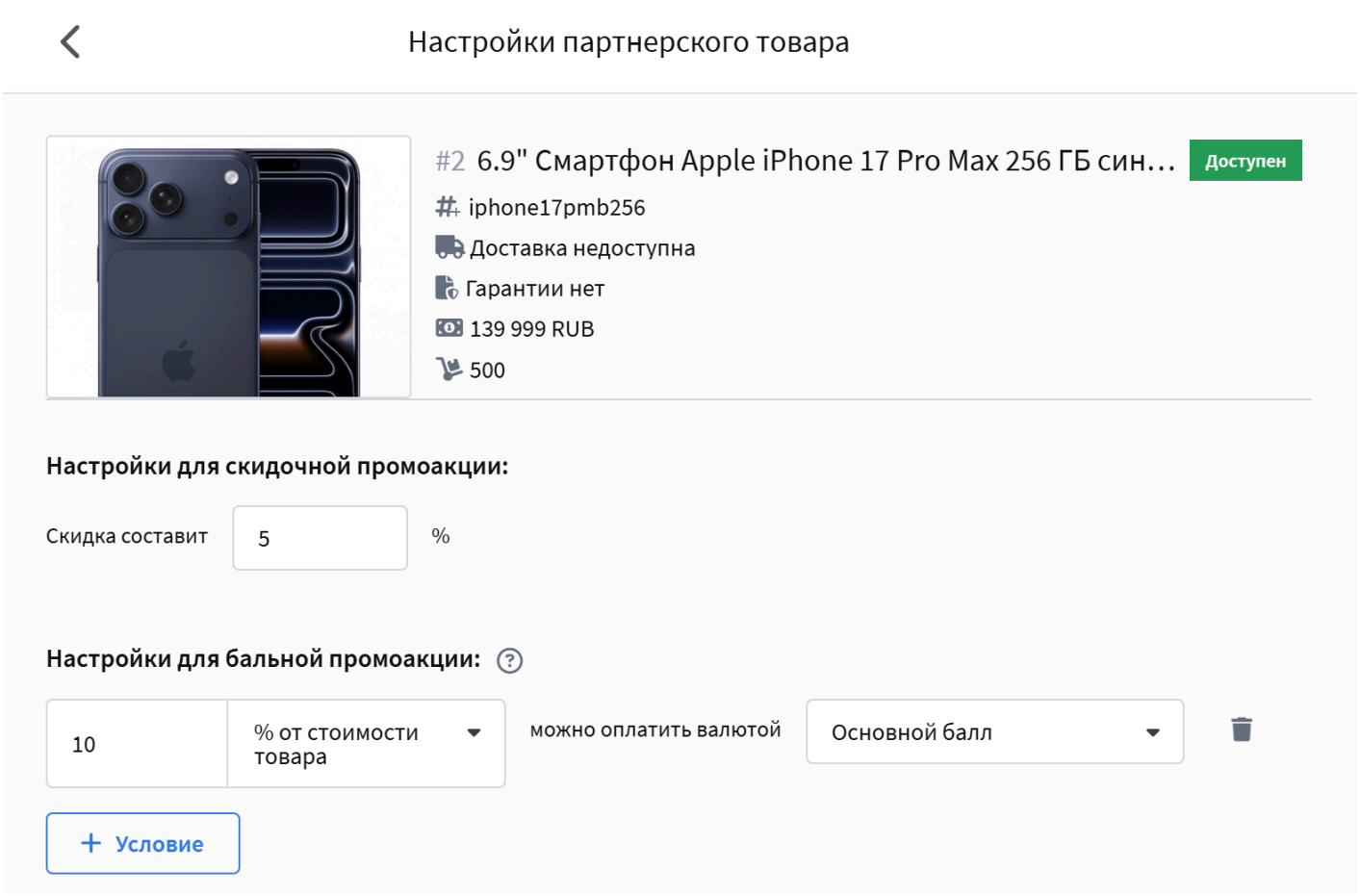
При добавлении партнерских товаров также можно задать **условия применения** в зависимости от типа промоакции.

Настройки для скидочной промоакции:

- **Скидка составит %** — тип скидки;

Настройки для балльной промоакции:

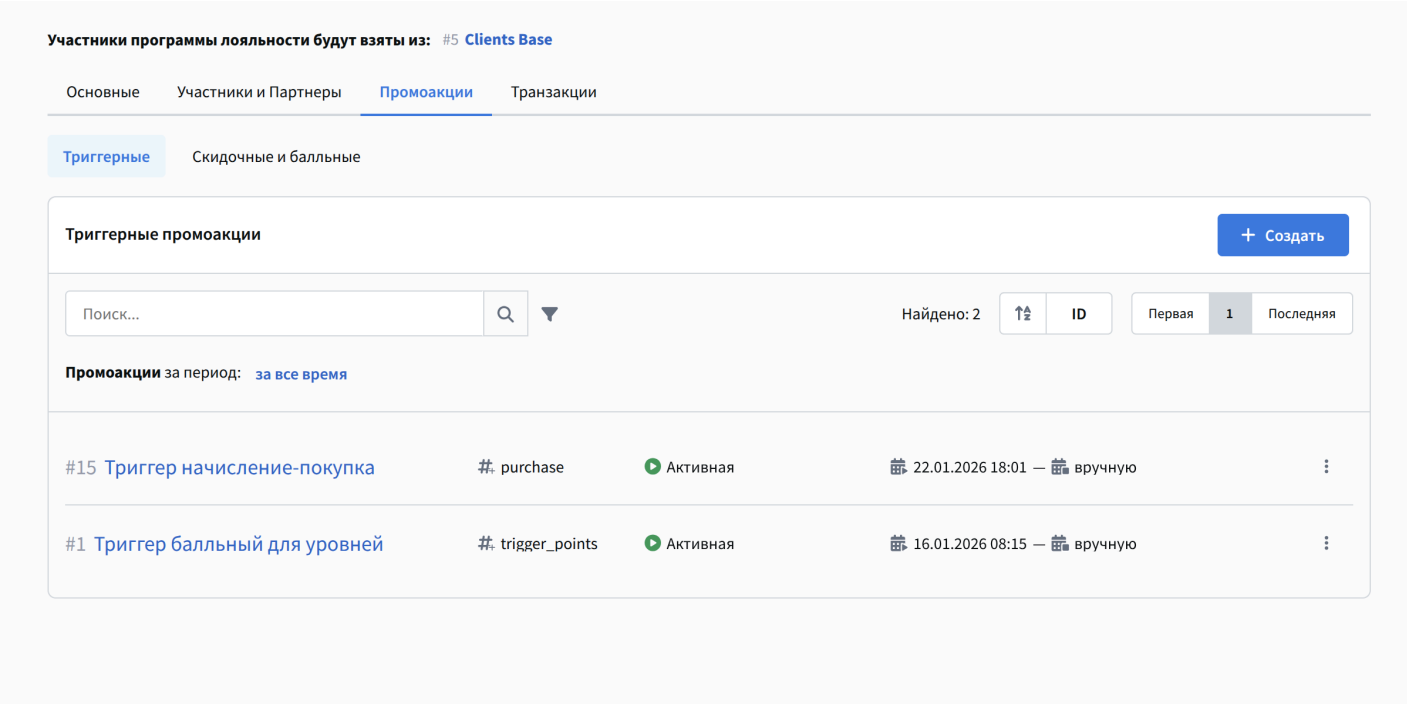
- **Значение** — количество баллов **или** процент от стоимости товара, который будет использован для расчета;
- **Баллы / % от стоимости товара** — выбор механики расчета: начисление/списание **баллов** либо применение **скидки в %**;
- **Можно оплатить валютой** — определяет, сколько баллов можно использовать как эквивалент денежной оплаты:



Партнерские товары используются для формирования **специальных предложений** в рамках программы лояльности. Такие предложения доступны только участникам, для которых применяются условия партнерства, и настраиваются через соответствующие промоакции с бенефитом “Скидка от партнера”.

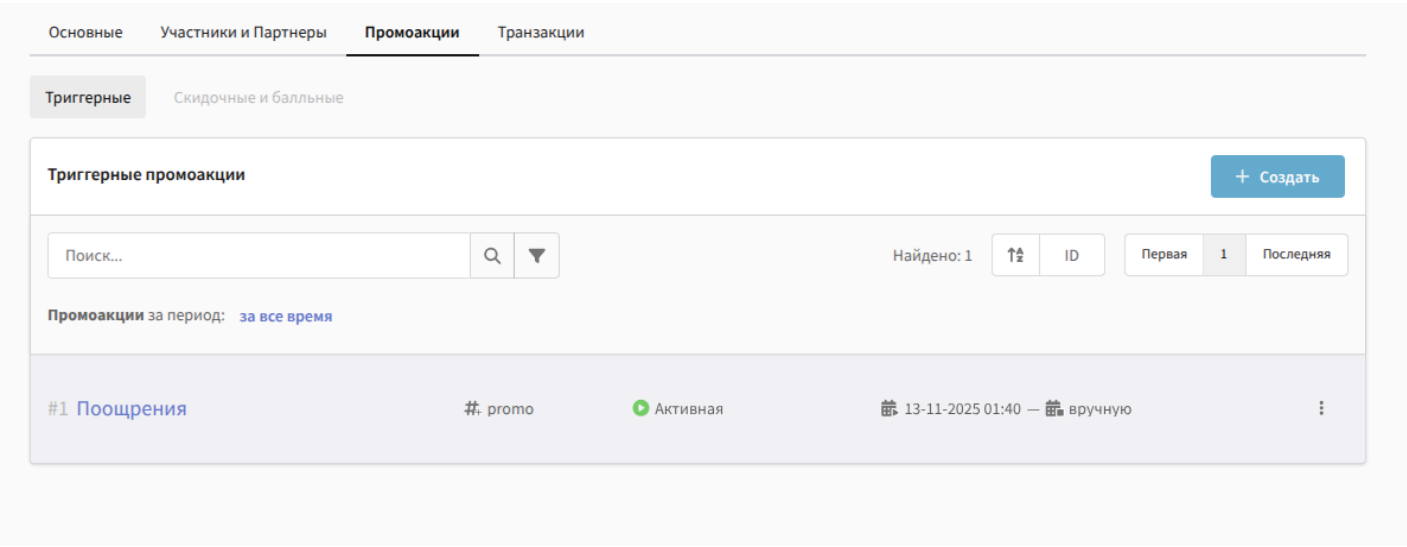
Промоакции

Раздел Промоакции предназначен для создания и управления акциями программы лояльности, которые автоматически начисляют и списывают баллы, применяют скидки и другие преимущества к заказам и товарам в Маркете.



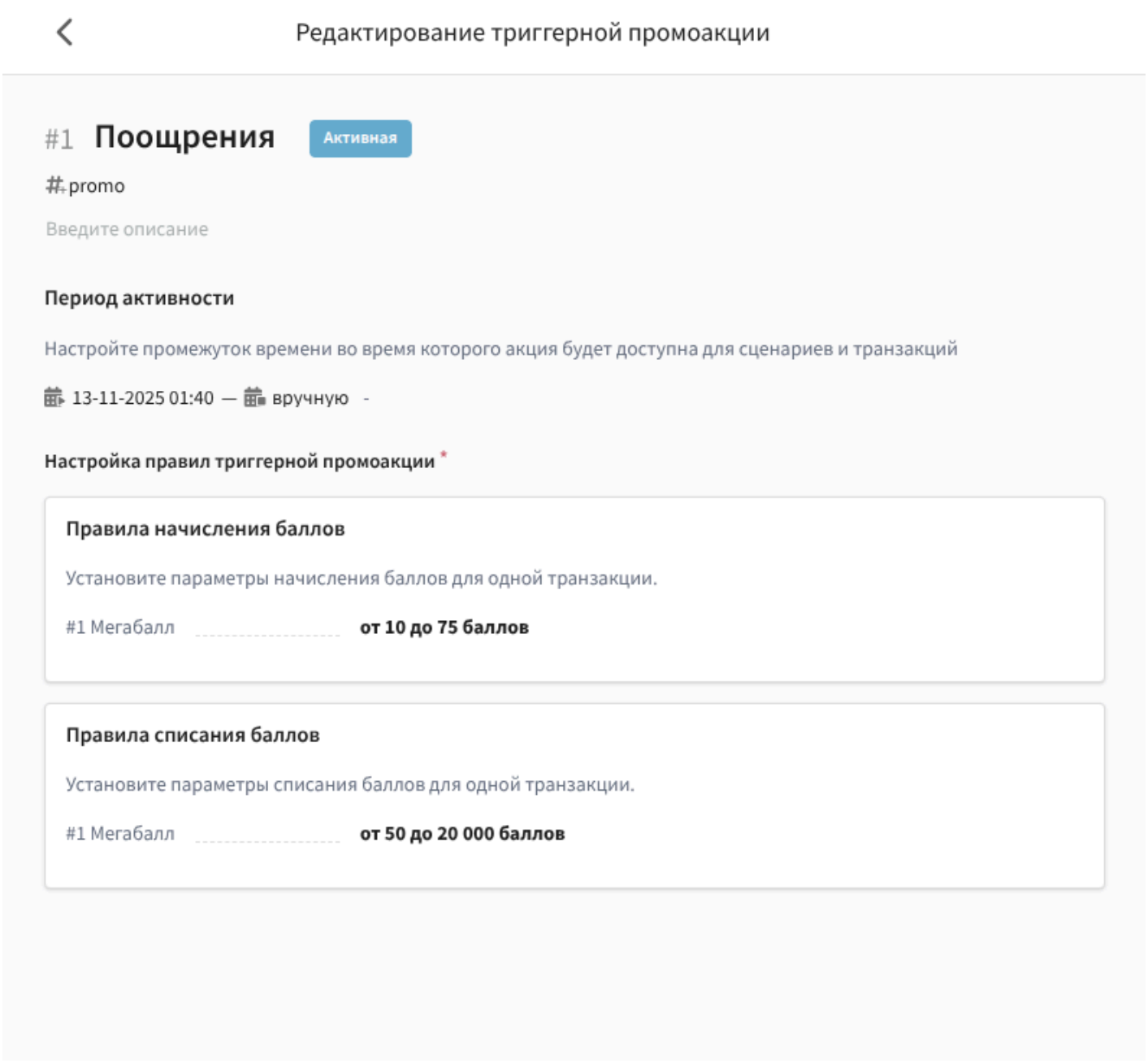
Промоакции делятся на две группы: **триггерные** и **скидочные и балльные** акции.

Триггерные промоакции



Триггерные промоакции — маркетинговые мероприятия, при **соблюдении правил (активации триггеров)** которых участник приобретает право на получение бонусных баллов. Иными словами, это набор правил, в соответствии с которым клиент может заработать (а затем и потерять) баллы лояльности. Такие промоакции связываются с триггерами при создании рассылок и [сценариев](#).

Чтобы создать новую триггерную промоакцию, нажмите **+ Создать**. Перед вами появится окно настройки промоакции:

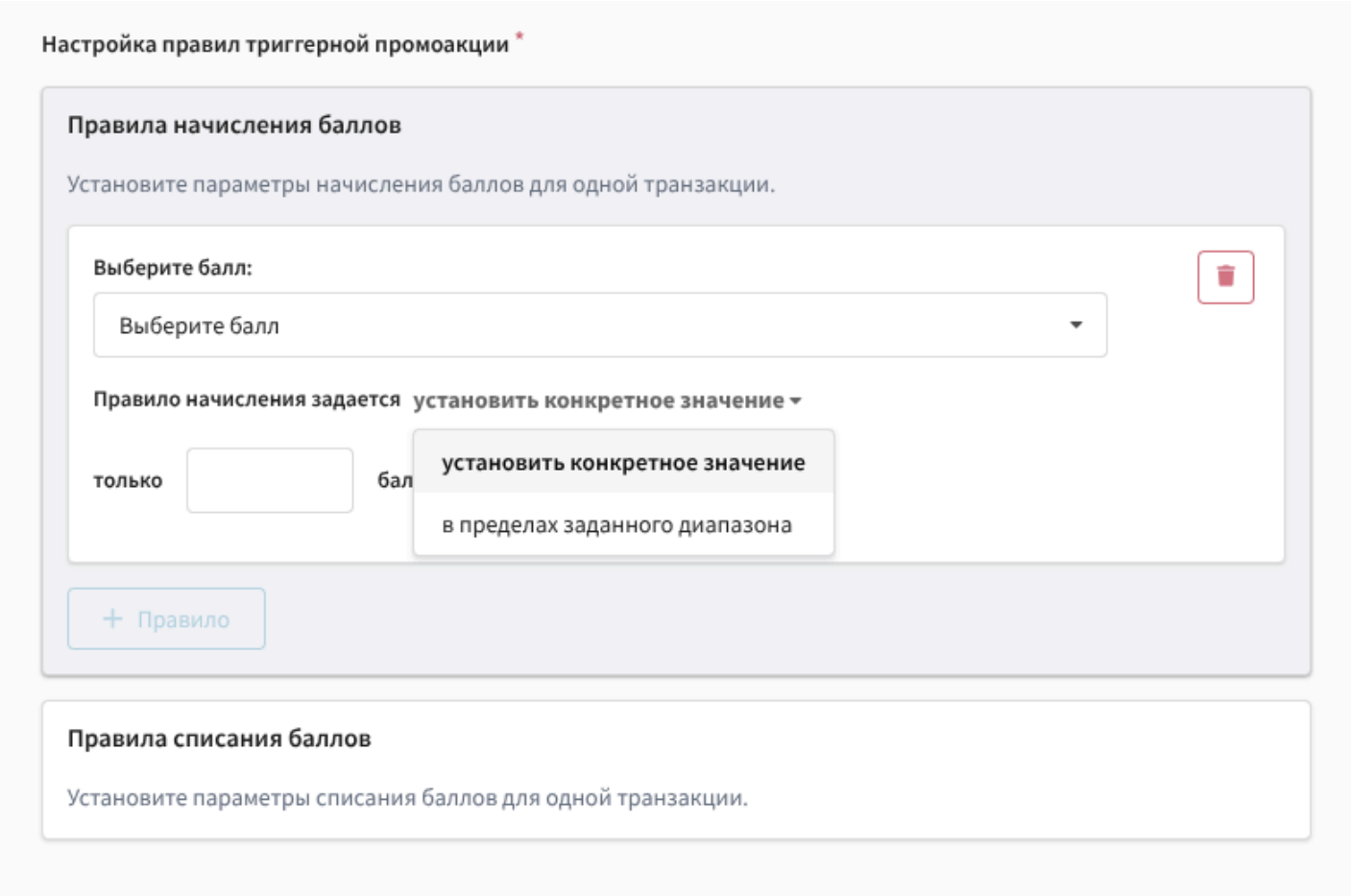


Укажите **название** промоакции и ее **дополнительный идентификатор**. При необходимости добавьте **описание**.

Затем обозначьте дату **старта** и **окончания** промоакции:

- **Старт промоакции** — начнется сразу после активации или в конкретный день и время;
- **Окончание промоакции** — будет окончена вручную, закончится в конкретный день и время или закончится спустя указанное время.

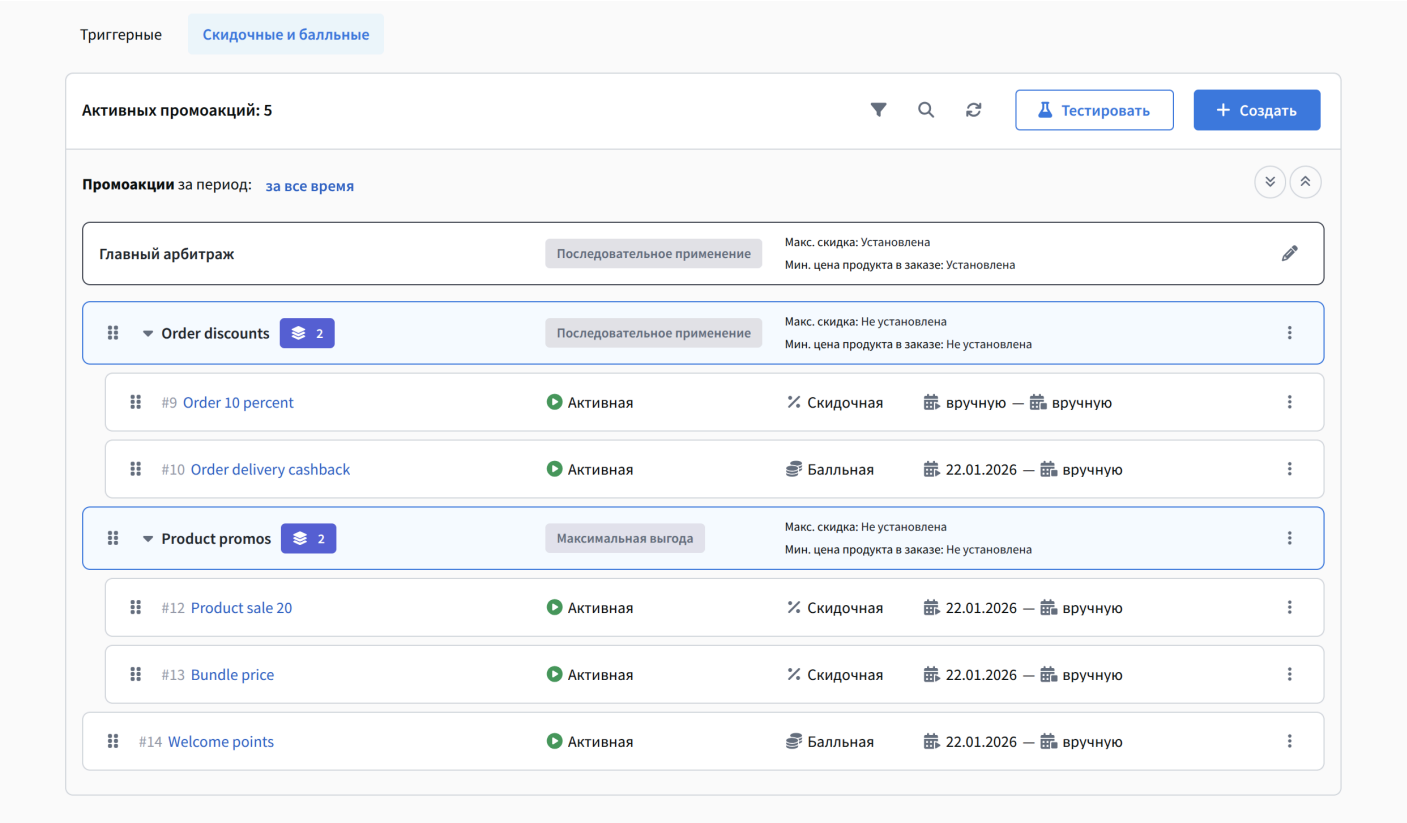
Далее установите правила промоакции. Они регулируют то, как могут начисляться и списываться баллы у участников. Правила начисления и списания настраиваются **отдельными** условиями. Укажите тип баллов, к которым будут относиться правила и конкретное значение/диапазон баллов для изменения:



После сохранения и активации промоакции вы можете использовать ее в триггерных рассылках и сценариях, чтобы автоматизировать начисление баллов. Например, вы можете настроить акцию, при которой участнику начисляются бонусные баллы за оформление первого заказа, за покупку товаров из выбранной категории или за достижение определенной суммы заказа.

Скидочные и балльные промоакции

Балльные и скидочные промоакции — это акции программы лояльности, которые автоматически применяются в момент оформления заказа или при расчете заказа через Маркет. Они управляют начислением и списанием баллов, применением скидок и других преимуществ на основе настроенных условий, аудитории и правил применения.



Балльные промоакции управляют балансом участника программы лояльности. Они начисляют или списывают баллы, как стандартные так и квалификационные при выполнении заданных условий, например, покупке товаров, достижении суммы заказа, участии в акции или использовании баллов при оплате. Такие промоакции не изменяют цену товаров напрямую, а влияют на будущую выгоду участника и его уровень в программе лояльности.В рамках балльных промоакций можно начислять **как стандартные, так и квалификационные баллы**, в зависимости от выбранного типа баллов и настроенных бенефитов. Это позволяет одновременно управлять бонусной выгодой участника и его уровнем в программе лояльности.

Скидочные промоакции применяются к заказам или отдельным товарам и изменяют их стоимость. Они используются для предоставления выгоды при покупке: скидок, специальных цен и других преимуществ.

Арбитраж

Промоакции могут действовать одновременно и применяться к одному заказу или товару, поэтому системе требуется механизм управления их совместной работой. Для этого в программе лояльности предусмотрен **арбитраж**.

Арбитраж — это правило применения промоакций, которое определяет, как система обрабатывает несколько подходящих акций одновременно. Он определяет:

- какие промоакции будут применены;
- в каком порядке они применяются;
- могут ли акции комбинироваться между собой;
- какие акции применяются вместе, а какие — нет.

Арбитраж используется каждый раз, когда один заказ или товар подходит под условия нескольких промоакций. Он настраивается на нескольких уровнях и работает иерархически.

Главный арбитраж — это основной арбитраж, который применяется ко всем балльным и скидочным промоакциям внутри программы лояльности. Он задает общий принцип расчета выгод и всегда влияет на итоговый результат, независимо от структуры акций.

Даже если промоакции объединены в группы или имеют собственные правила, финальный расчет **всегда** проходит через главный арбитраж.

Арбитраж групп — промоакции можно объединять в группы и настраивать для каждой группы собственный арбитраж. Он определяет, как именно акции внутри группы будут взаимодействовать между собой. Внутри групп можно также создавать подгруппы акций.

Групповой арбитраж работает локально — только внутри своей группы. Результат его работы затем передается на уровень выше и обрабатывается главным арбитражем программы лояльности.

Арбитраж внутри промоакции — внутри одной промоакции может быть несколько бенефитов (преимуществ и выгод). Для них также задается правило обработки, которое определяет, применяются ли бенефиты параллельно или последовательно.

Существует несколько типов арбитража:

- **Последовательное применение** — промоакции применяются по очереди, в том порядке, в котором они расположены в группе. Каждая следующая акция рассчитывается от результата предыдущей. Если сначала применена скидка, а затем еще одна, вторая считается уже от уменьшенной стоимости. Подходит для сценариев, где важен порядок действий и пошаговое изменение цены или баланса.
- **Несовместимость** — из всех подходящих промоакций применяется только одна. На уровне продукта или заказа срабатывает та акция, которая первой выполнила свои условия. Остальные акции игнорируются. Используется, когда акции логически исключают друг друга, например: “или скидка, или подарок”.
- **Суммирование** — все подходящие промоакции применяются одновременно. Скидки складываются между собой, начисленные баллы суммируются. Результат рассчитывается как совокупный эффект всех акций в группе. Подходит для сценариев, где несколько акций должны усиливать друг друга.
- **Максимальная выгода** — система перебирает возможные комбинации акций и выбирает ту, которая дает наибольшую выгоду участнику. Используется в сложных промо-механиках, где важно автоматически выбрать лучший вариант без ручной настройки приоритетов.

CAUTION: Ограничения типов арбитража

Не все бенефиты поддерживаются некоторыми типами арбитража:

Для арбитража “Максимальная выгода” недоступны:

- Отобразить шаблонизированный текст
- Выдать промокод

Для арбитража “Суммирование” недоступны:

- Округлить цену / Округлить цену заказа
- Использовать цену из поля
- Начислить баллы из пользовательского поля

Дополнительно для арбитража можно задать **ограничения**, которые контролируют итоговый результат и защищают от чрезмерных скидок и слишком низкой цены:

- **Максимальная скидка** — ограничивает предельный размер скидки, который может быть применен, вне зависимости от количества и комбинации сработавших акций. Если суммарная скидка превышает установленное значение, система автоматически ее ограничивает.
- **Минимальная цена продукта** — задает минимально допустимую цену, ниже которой стоимость товара не может опуститься после применения всех акций. Если расчет по промоакциям приводит к цене ниже заданного минимума, система корректирует результат и останавливает дальнейшее снижение.

Ограничения, установленные в **главном арбитраже**, действуют на все акции в программе лояльности.

Арбитраж делает результат применения акций предсказуемым и контролируемым, даже при сложной структуре скидок и бонусов.

TIP: Практический пример

Предположим, пользователь хочет купить себе моноблок за 100 000 ₽. Ему доступны скидка 20% в день рождения и списание 15% цены бонусами. Вот как это будет работать с разными арбитражами:

Последовательное применение
100 000 —> −20% = 80 000 —> −15% = **68 000 ₽**
Акции считаются по очереди от результата предыдущей.

Несовместимость
100 000 —> либо −20% = **80 000 ₽**, либо −15% = **85 000 ₽**
Применяется только одна акция.

Суммирование
100 000 —> −(20% + 15%) = **65 000 ₽** Все акции складываются.

Максимальная выгода
Сравнение вариантов: 80 000 / 85 000 / 68 000 / **65 000 ₽**
Выбирается минимальная цена.

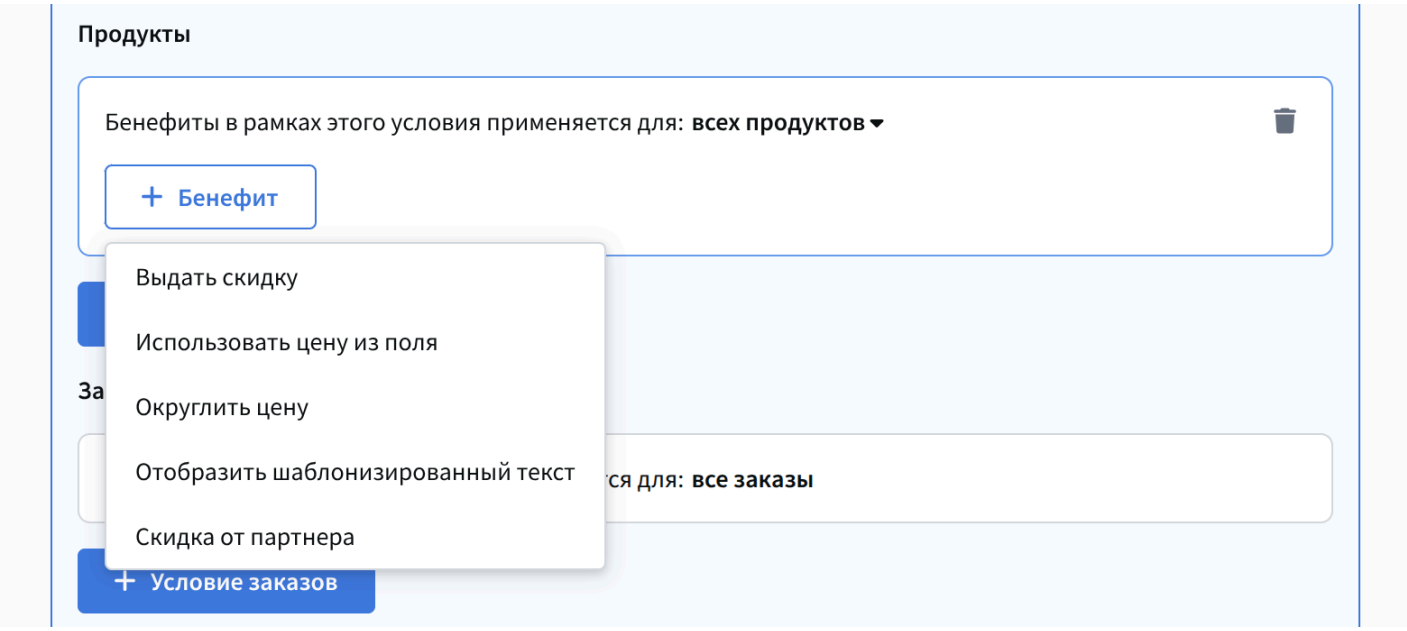
Бенефиты для скидочных акций

Для товаров:

- **Выдать скидку** — скидка выдается в указанном размере либо в процентах от стоимости товара;
- **Использовать цену из поля** — будет использована цена из пользовательского поля продукта. Эту цену можно умножить на указанный множитель, например, на 0,75, если вы хотите сделать 25-процентную скидку;
- **Округлить цену** — округляет цену продукта до целого, половины или целого, ближайшего целого вниз, до десятков или до ближайшего десятка;
- **Отобразить шаблонизированный текст** — передает с заказом текст, в который можно добавить переменные и emoji. Этот текст передается вместе с данными заказа и может быть передан через API, например, в электронное письмо или на кассовый терминал;
- **Скидка от партнера** — выдает скидку на товары выбранных партнеров согласно настройкам партнерских товаров.

Для заказа:

- **Выдать скидку** — скидка выдается в указанном размере либо в процентах от стоимости заказа. Можно ограничить количество единиц товара, на которое распространяется скидка в рамках одного заказа;
- **Округлить цену заказа** — округляет итоговую стоимость заказа;
- **Отобразить шаблонизированный текст** — передает с заказом текст, в который можно добавить переменные и emoji;
- **Выдать промокод** — выбирает промокод из указанного пула и привязывает его к участнику. Дополнительно можно настроить шаблон сообщения, которое получит участник вместе с промокодом.



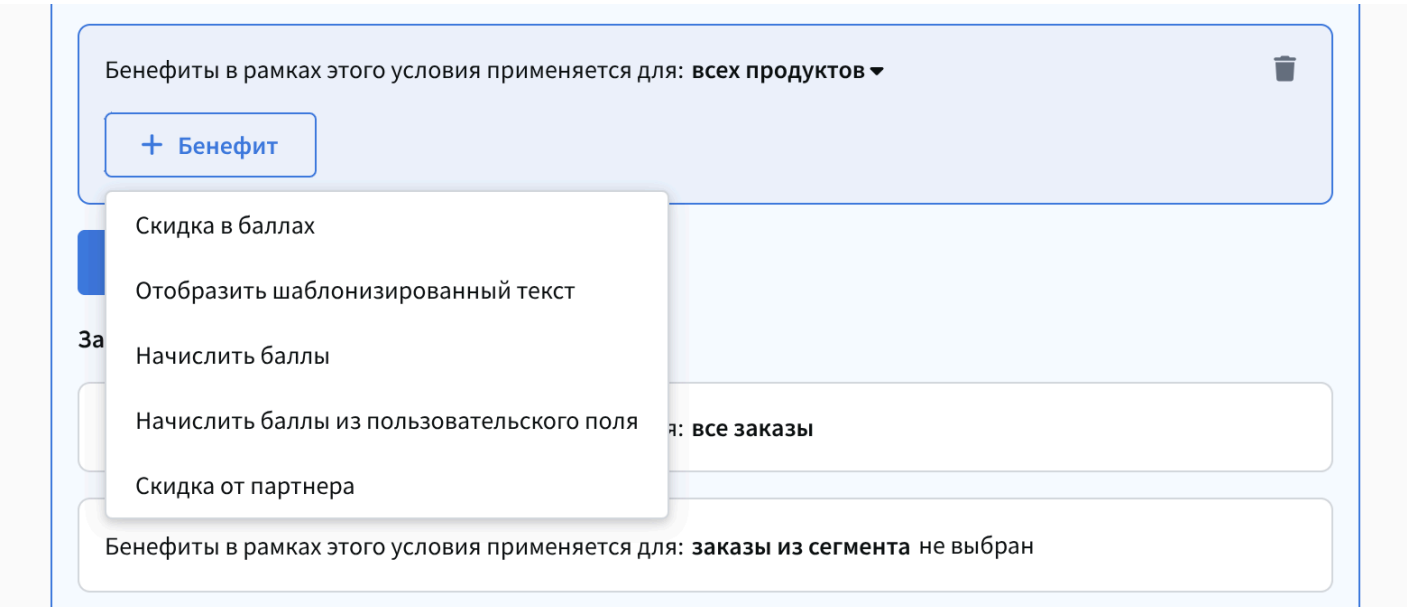
Бенефиты для балльных акций

Для товаров:

- **Скидка в баллах** — скидка выдается **баллами** в указанном размере либо в процентах от стоимости товара, либо фиксированным количеством баллов. Можно ограничить количество единиц товара, на которое распространяется скидка в рамках одного заказа;
- **Отобразить шаблонизированный текст** — передает с заказом текст, в который можно добавить переменные и emoji;
- **Начислить баллы** — на счет профиля поступают указанные баллы лояльности. Можно настроить момент начисления, например, сразу при создании заказа или после его доставки;
- **Начислить баллы из пользовательского поля** — будет начислено количество баллов, взятое из пользовательского поля продукта. Также можно зачислять определенное количество баллов за каждую потраченную единицу валюты, например, начислять 200 баллов за каждые потраченные 1500 рублей;
- **Скидка от партнера** — выдает скидку в баллах на товары выбранных партнеров согласно настройкам партнерских товаров.

Для заказа:

- **Скидка в баллах** — скидка выдается баллами в процентах от стоимости заказа или фиксированным количеством баллов;
- **Отобразить шаблонизированный текст** — передает с заказом текст;
- **Начислить баллы** — начисляет баллы за весь заказ целиком.

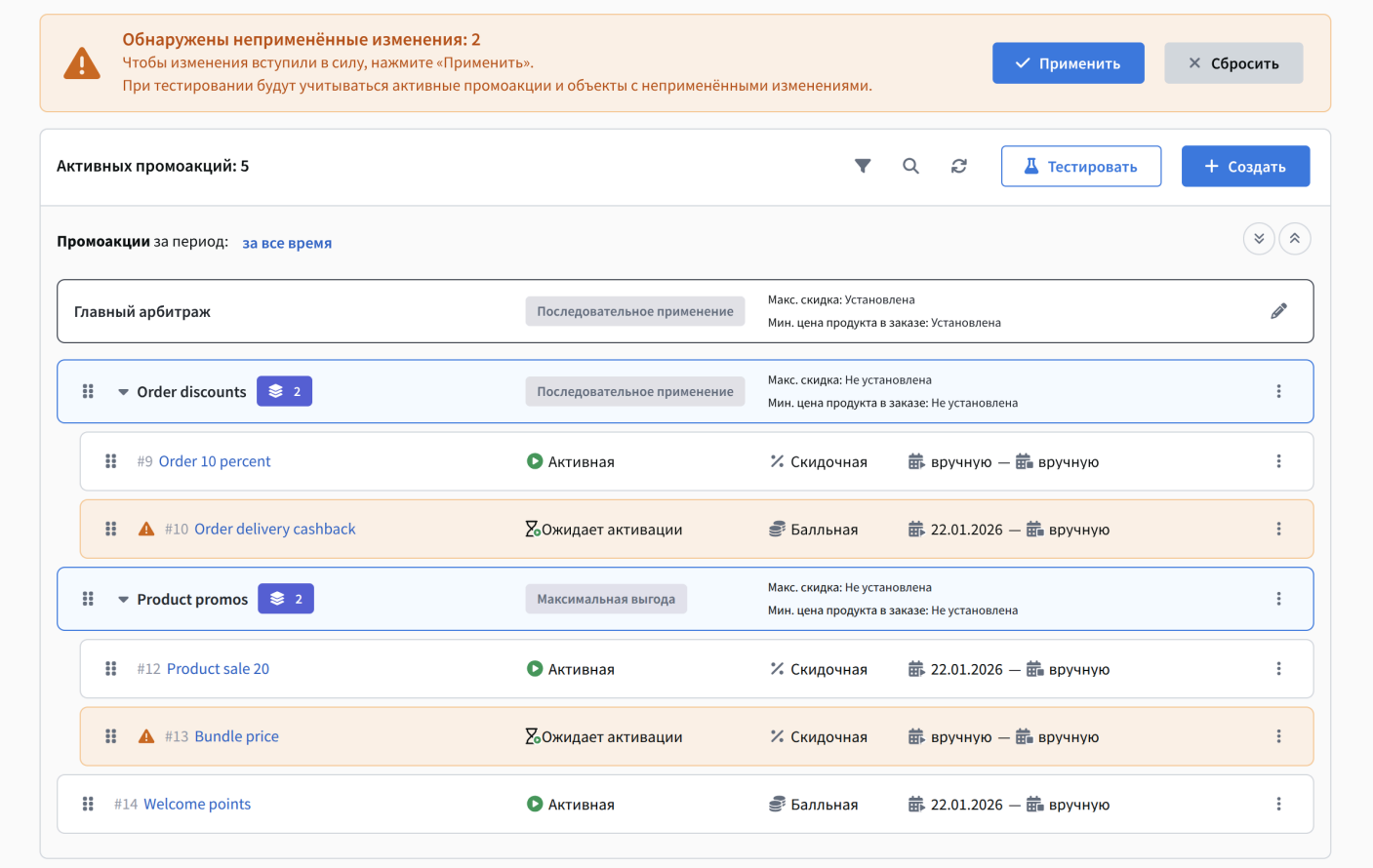


После заполнения всех обязательных блоков промоакцию можно **сохранить** или **активировать**. Активированная скидочная промоакция начинает участвовать в расчетах заказов в соответствии с настроенным арбитражем и условиями. Если вы выберете сохранение, промоакция появится как **черновик** в списке акций. В дальнейшем такую акцию нужно будет **активировать**.

При работе с промоакциями сверху страницы отображается **статус изменений** и **быстрые действия**.

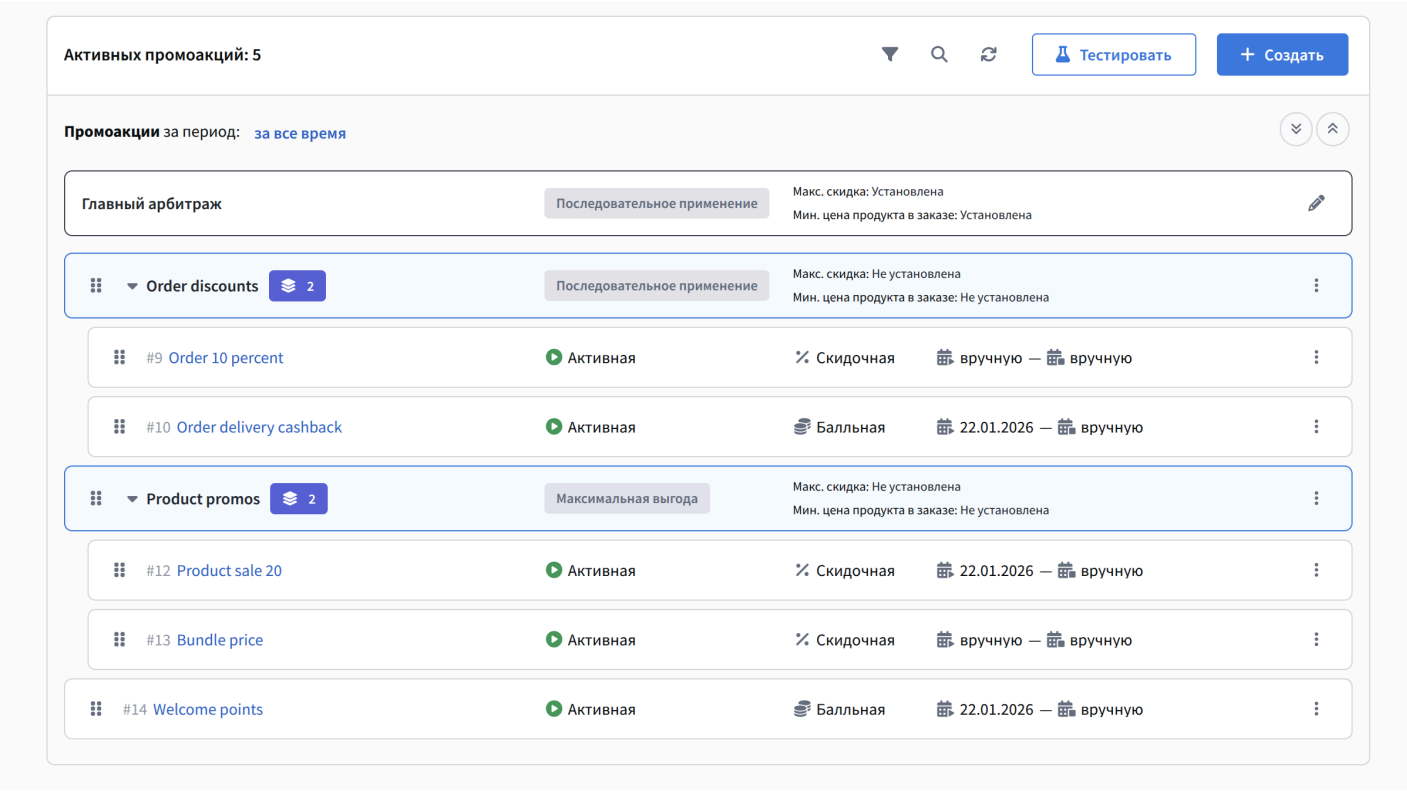
Если в дереве есть изменения, которые еще не введены в силу, система покажет уведомление “Обнаружены неприменненные изменения”. В нем доступны две кнопки:

- **Применить** — вводит изменения в действие и делает их рабочими для реальных расчетов.
- **Сбросить** — отменяет неприменненные изменения и возвращает состояние к последней примененной версии.



Непримененные изменения **учитываются** при работе с промоакцией в режиме тестирования. В реальных расчётах изменения начнут учитываться только после нажатия **Применить**.

Ниже расположен список промоакций и групп:

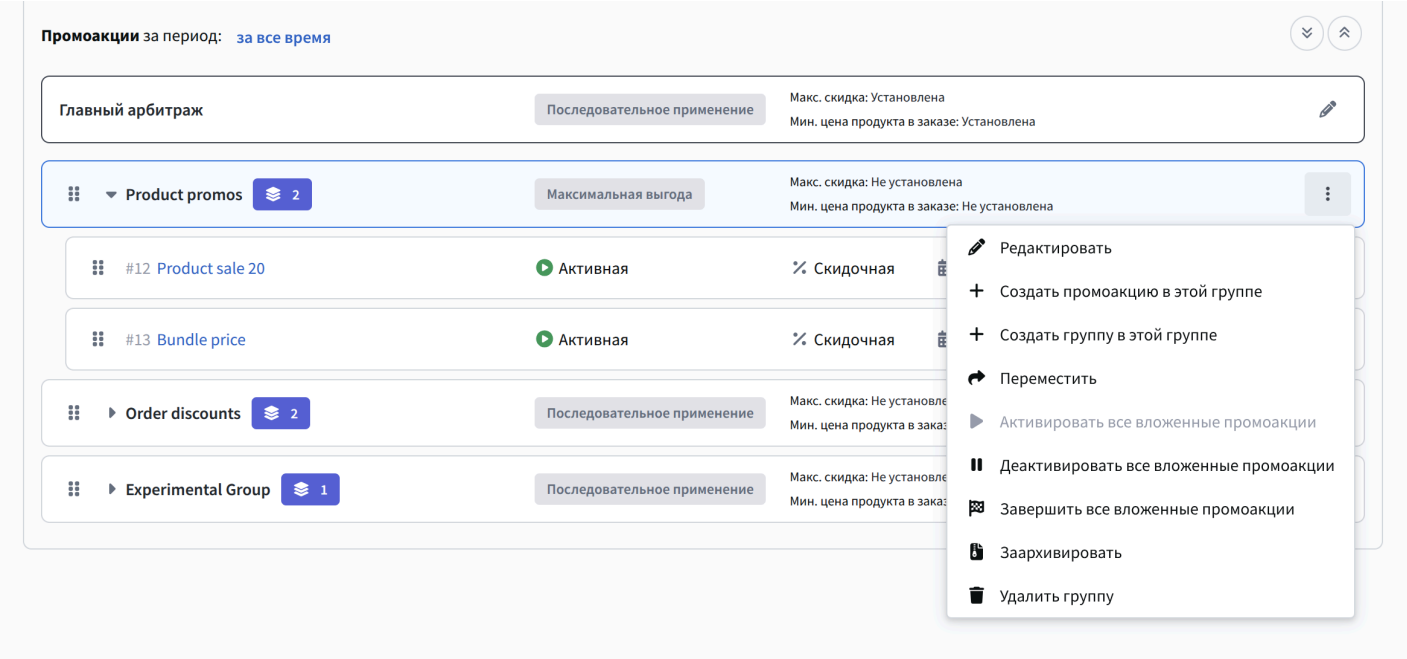


- в заголовке блока видно количество активных промоакций;
- справа доступны быстрые действия: фильтры/поиск, Тестировать, Создать;
- в дереве отображается Главный арбитраж и настройки его ограничений (например, максимальная скидка и минимальная цена продукта);
- у групп и промоакций есть меню действий, а сами элементы можно перетаскивать внутри дерева и перемещать между группами.

Для созданных акций доступно **меню быстрых действий**. С его помощью вы можете:

- редактировать акцию
- деактивировать акцию (если она активна)
- переместить акцию в другую группу
- клонировать акцию
- завершить акцию (все процессы, связанные с ней, будут завершены и ее невозможно будет активировать повторно)

Управление группами



Группы промоакций используются для логической организации акций и управления их совместной работой через отдельный арбитраж.

Группа — это узел в дереве промоакций, внутри которого задается собственный тип арбитража, настраиваются ограничения по максимальной скидке и минимальной цене, а также определяется порядок применения входящих в нее акций.

Группы могут быть вложенными (но не более 9 уровней) и располагаться внутри других групп или в главном арбитраже программы лояльности.

При создании группы указываются:

- название группы;
- родительская группа или главный арбитраж;
- тип арбитража и его ограничения.

После создания группы промоакции можно:

- перемещать в нее при создании или редактировании;
- перетаскивать между группами прямо в списке акций;
- менять порядок акций внутри группы для влияния на результат (кроме суммирования);
- через меню быстрых действий можно деактивировать все акции внутри группы, а также заархивировать ее.

Как и для промоакций, изменения в структуре групп и порядке акций должны быть применены. До применения изменения будут учитываться только в **тестовом режиме**. Финальный результат применения акций всегда дополнительно обрабатывается главным арбитражем программы лояльности. Все действия, примененные к группе (активация, деактивация, архивация), каскадно применяются ко всем вложенным промоакциям и подгруппам.

INFO

Группа удаляется только при выполнении одного из условий:

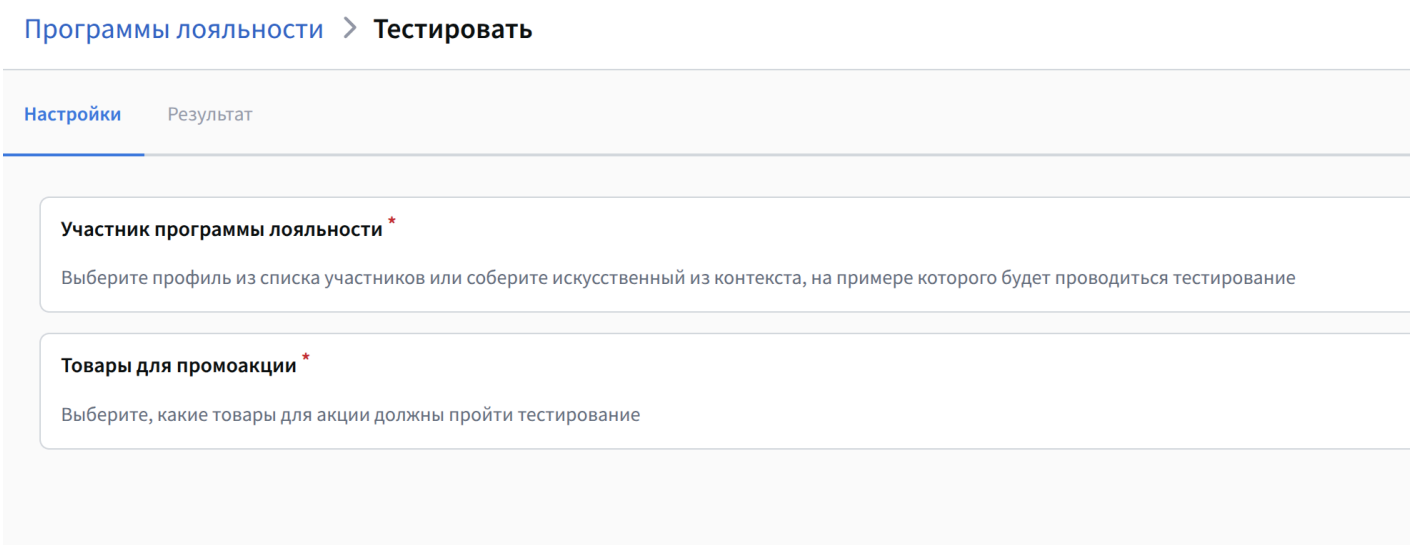
- внутри группы нет ни одной промоакции;

- все промоакции внутри группы находятся в статусе “Черновик”.

Тестирование промоакций

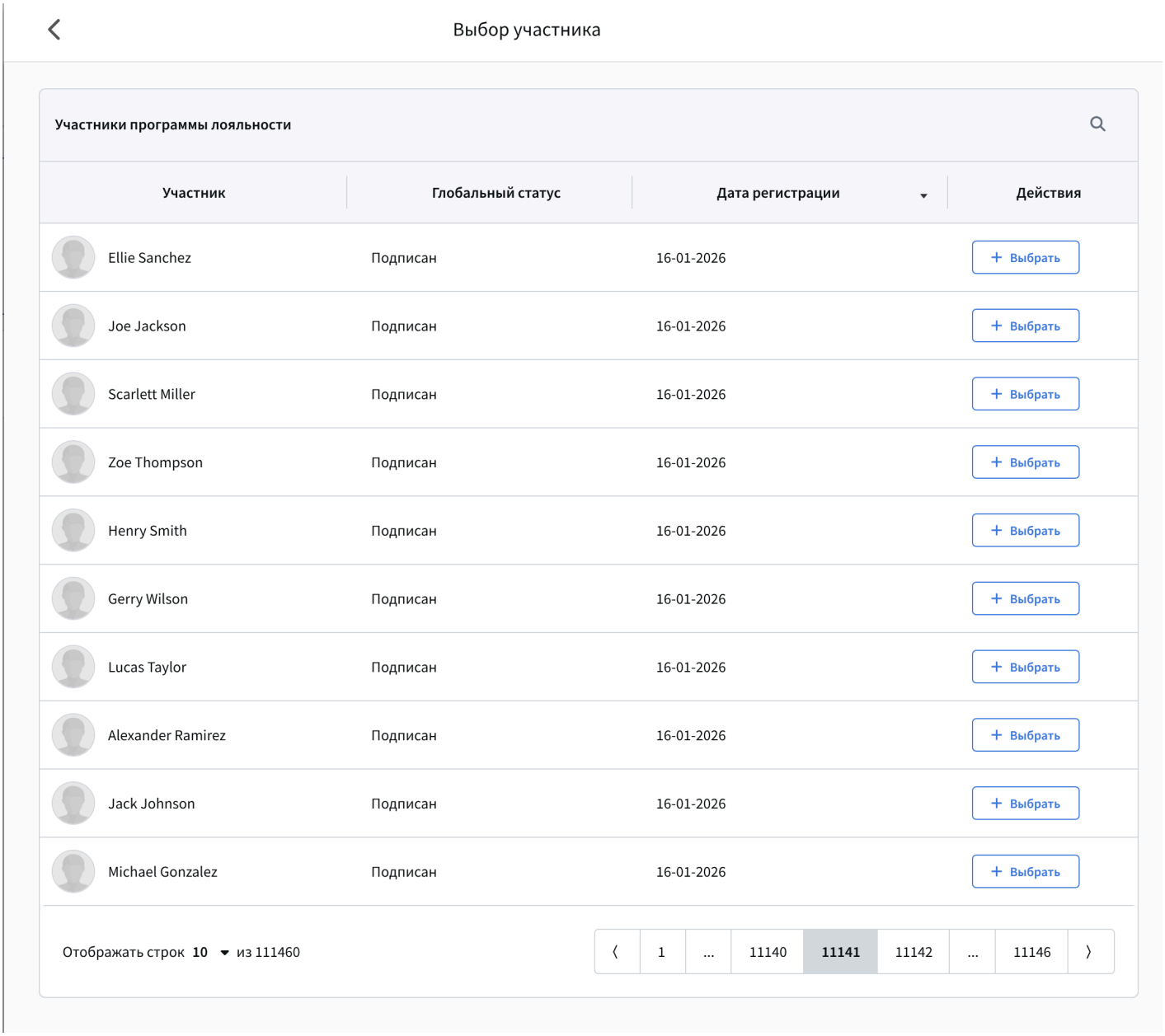
Чтобы проверить, как промоакции сработают на заказе, нажмите **Тестировать** — откроется режим тестирования, где можно задать параметры заказа и участника и увидеть результат применения акций (в том числе с учетом неприменных изменений).

Выберите **участника** и **товары** для тестирования:



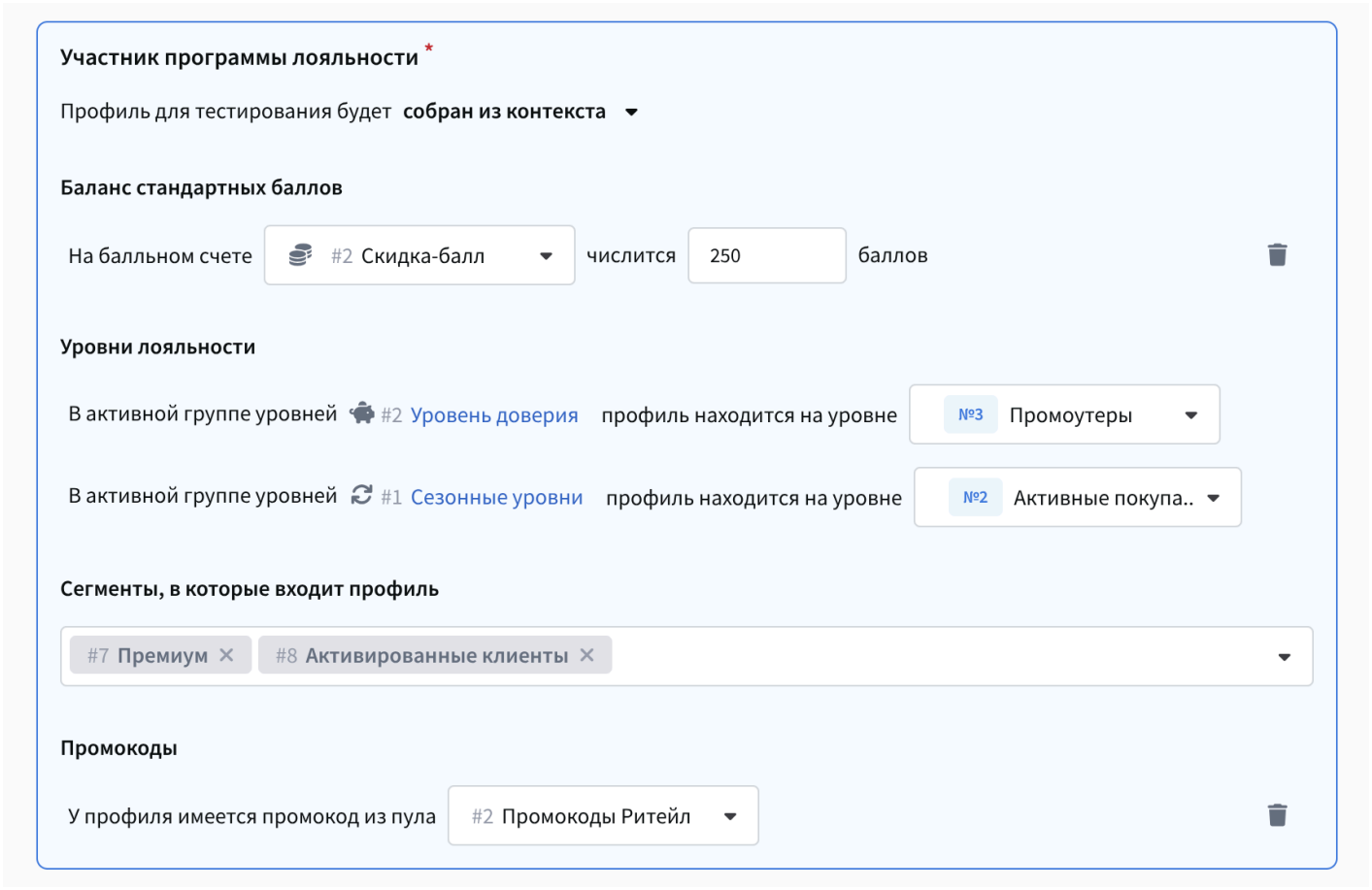
Вы можете выбрать **реальный профиль участника** или **собрать профиль из контекста**.

Если вы хотите выбрать существующий профиль: нажмите **Выбрать**, перед вами появится список участников программы. После выбора участника, он зафиксируется в тестировании:

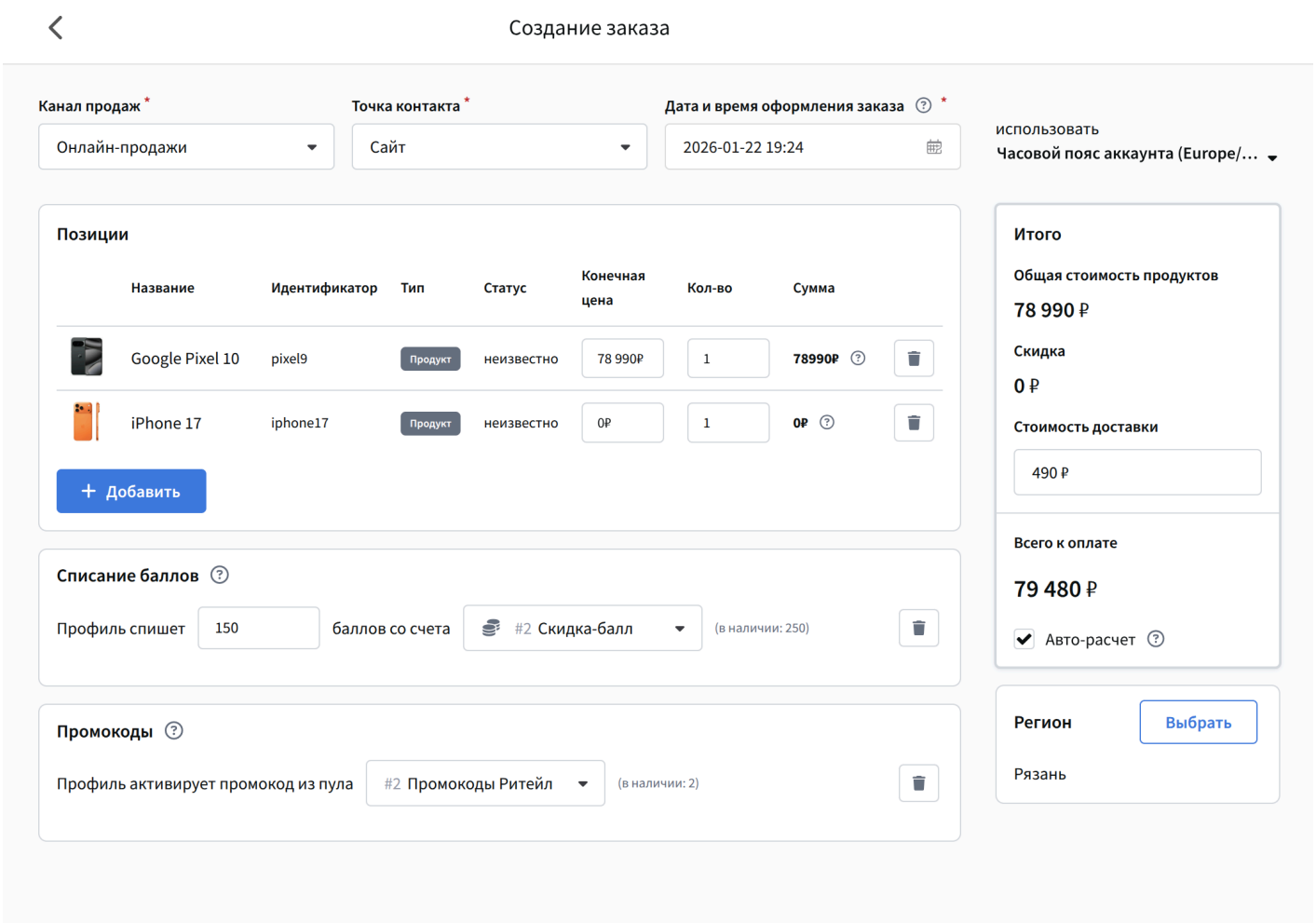


В случае, если вы собираете профиль из контекста, будет создан тестовый участник, для которого можно указать:

- его бонусный баланс;
- на каких уровнях он находится в программе лояльности;
- в какие сегменты входит профиль;
- какие промокоды привязаны к профилю.



После настройки участника необходимо определить товары для промоакции. Они задаются в рамках **заказа**:

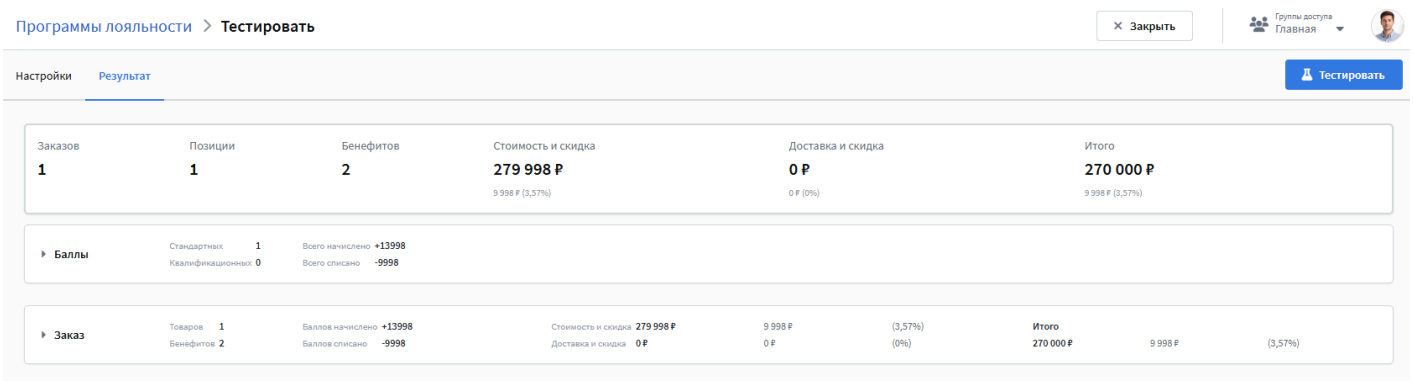


Для тестового заказа необходимо указать:

- **Канал и точку продаж** — канал, по которому клиент совершил заказ.
- **Дата и время оформление заказа** — дата, время и часовой пояс.
- **Позиции** — товары для заказа. Можно выбрать продукт, количество, а также конечную цену. Выбрав товар, вы можете перейти к его редактированию. Изменения применяются только в тестовом заказе и не меняют данные Маркета. Это полезно, когда нужно проверить промоакции на новых ценах.
- **Списание баллов** — количество и тип баллов, которые будут списаны в рамках заказа.
- **Промокоды** — активируемые профилем промокоды.
- **Стоимость доставки** — цена доставки товара.
- **Регион** — регион заказа.

Если активирован чекбокс **Авто-расчет**, итоговая цена рассчитывается как сумма товаров и цены доставки.

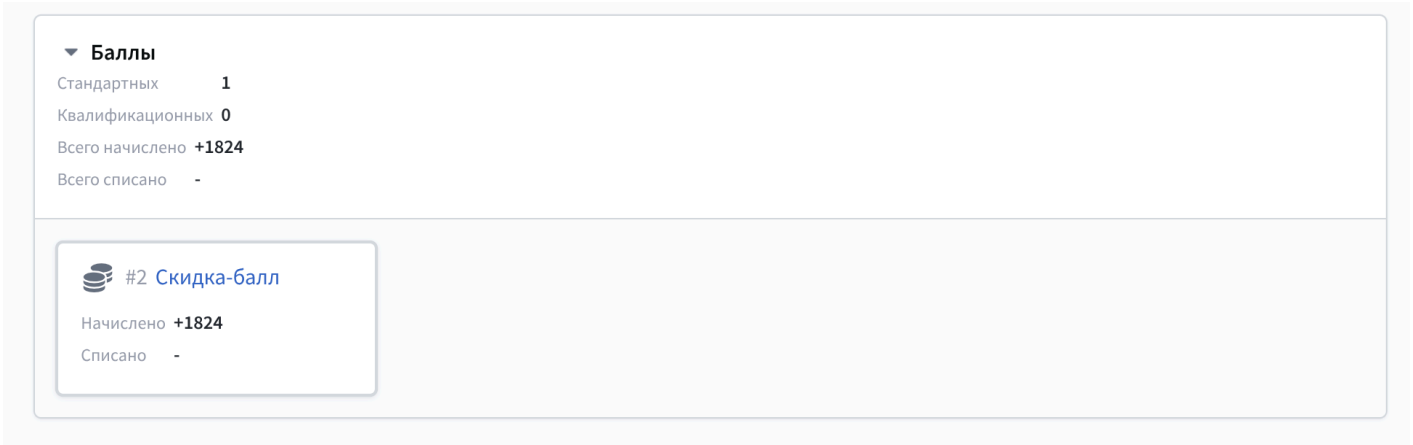
После завершения настройки участника и товаров, нажмите на кнопку **Тестировать**. После нажатия на нее, вы перейдете на вкладку **Результат**, на которой находится итог применения бенефитов:



На панели отображены:

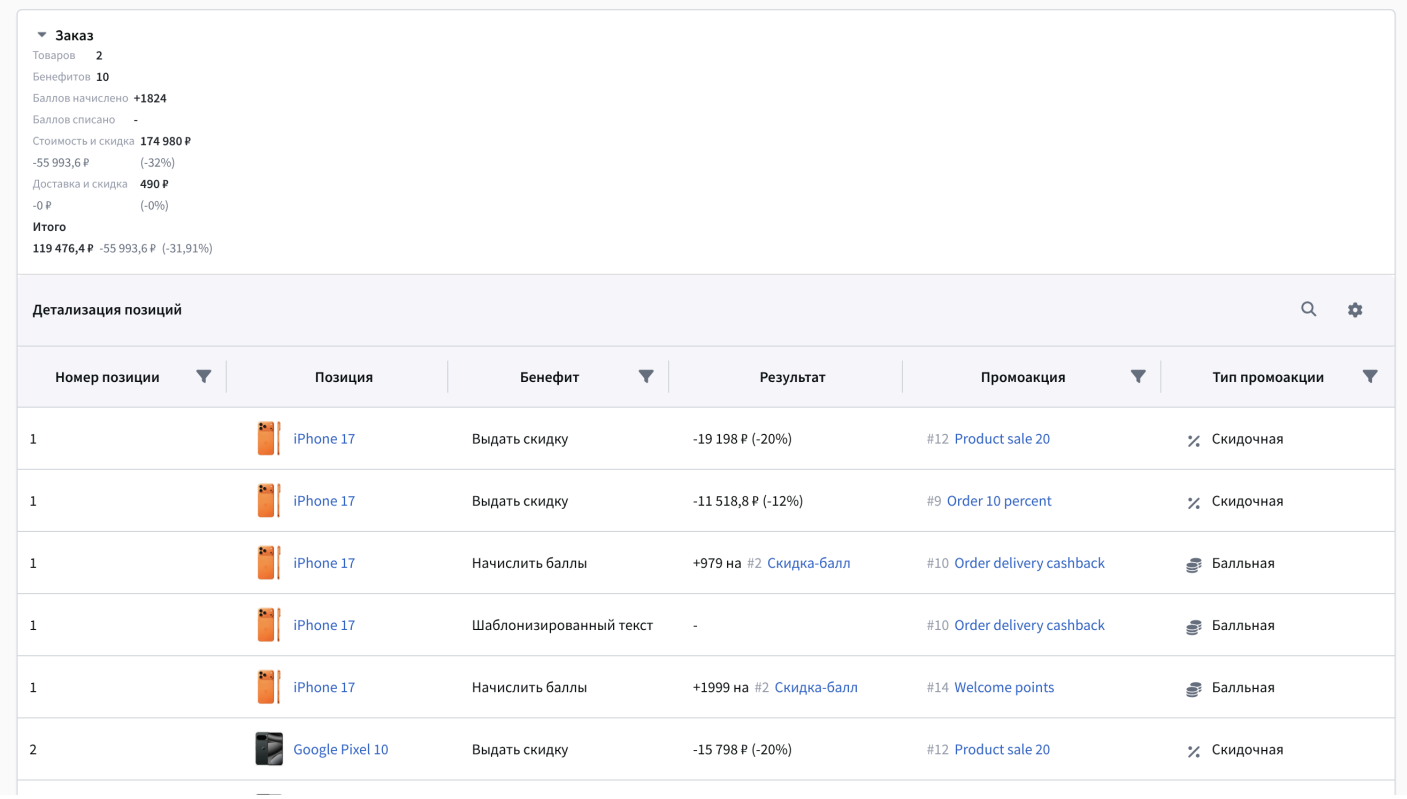
- Количество заказов
- Количество позиций в заказах
- Количество примененных бенефитов
- Общая стоимость заказов (без скидки) и размер скидки (под стоимостью)
- Стоимость доставки и скидка
- Итоговая сумма заказа с учетом примененной скидки (размер скидки расположен под суммой)

На вкладке **Баллы** видны изменения балльного баланса по итогу заказа:



Вкладка **Заказ** содержит таблицу с операциями, связанными с бенефитами заказа. Каждый элемент списка содержит следующую информацию:

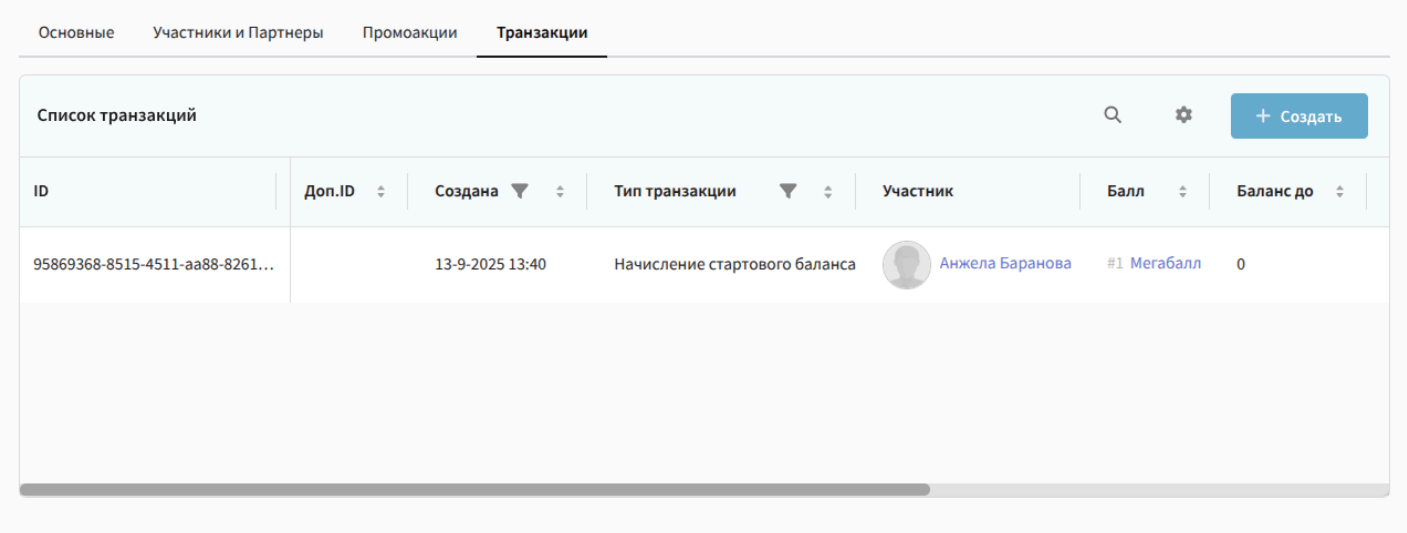
- Номер позиции в заказе
- Позиция (продукт)
- Бенефит
- Результат бенефита
- В рамках какой промоакции применялся бенефит
- Тип промоакции



Пользователю доступен поиск по таблице, а также он может настроить отображаемые поля в меню действий с таблицей.

Транзакции

Раздел **Транзакции** содержит список всех транзакций, проведенных в рамках программы лояльности. Здесь же вы можете создать новую транзакцию и начислить или списать баллы у участника программы.



В таблице транзакций доступна подробная информация о каждом изменении бонусного баланса профилей. Таблица содержит следующие колонки:

- **ID** — идентификатор транзакции
- **Дополнительный ID** — дополнительный идентификатор транзакции
- **Дата создания** — дата создания транзакции
- **Тип транзакции** — тип транзакции
- **Участник** — профиль участника программы лояльности
- **Балл** — тип балла, с которым проведена транзакция
- **Баланс до** — баланс до проведения транзакции
- **Значение** — количество начисленных/снятых баллов
- **Баланс после** — баланс после проведения транзакции
- **Тип источника** — тип сущности, из которой была запущена транзакция
- **Источник** — перейти к подробной сущности-источника

В этом разделе вы также можете создать **новую транзакцию**, для этого нажмите кнопку **+ Создать**. Перед вами появится окно настройки:

Создание балльной транзакции

#transaction

Участник промоакции *
Участник, с чьим счетом необходимо произвести манипуляцию

Анжела Баранова

Триггерная промоакция *
Выберите ТПА в рамках которой провести транзакцию

1 - Поощрения

Баллы *
Выберите баллы на чей счет провести транзакцию

1 - Мегабалл

Значение операции *
Настройте тип и значение ручной транзакции выбранного участника

- (•) **Начислить баллы на счет**
  Укажите кол-во начисляемых баллов на счет, и настройте активацию и сгорание начисленных баллов

  Начислить 35 Баллов

  от 10 до 75 баллов по правилу начисления

  Баллы активируются **сразу**

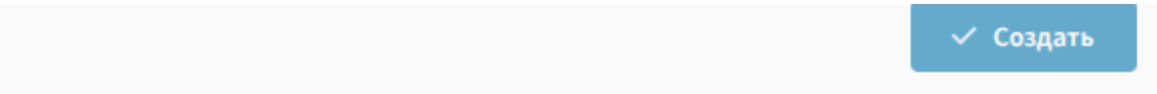
  Баллы сгорят **по заданной в баллах настройке**

- ( ) **Списать баллы с счета**
  Укажите кол-во списываеммых баллов с счета

Комментарий *

Бонусные баллы за выполнение цели

▸ **Дополнительные поля JSON**



- При необходимости укажите **дополнительный идентификатор**;
- Выберите **участника** транзакции. Это может быть любой профиль, который зарегистрирован в этой программе лояльности;
- Выберите **триггерную промоакцию**, в рамках которой будет проведена транзакция;
- Выберите **баллы**, с которыми будет проведена операция;
- Укажите **тип операции**, зачисление или списание;
- Добавьте **комментарий** к транзакции;
- При необходимости укажите **дополнительные поля JSON**.

Интеграция лояльности с внешними системами

Интеграция программ лояльности во внешнюю систему заказов

Программа лояльности в Altcraft Platform может использоваться в двух вариантах — в зависимости от того, передаются ли в платформу данные о товарах и заказах через модуль Маркет.

Маркет позволяет передавать в платформу данные о товарах и заказах и использовать их в коммуникациях с клиентами. Данные могут передаваться автоматически — через API или товарный фид, а также вручную — путём загрузки файла.

От наличия данных в Маркете зависит, какие механизмы лояльности вам доступны:

- **Без Маркета** вы не сможете **автоматически рассчитывать скидки**, начислять и списывать баллы при оформлении заказов. Операции начисления и списания баллов доступны только через триггерные промоакции и API лояльности.
- **С Маркетом** помимо базовых операций через триггерные промоакции вам становятся доступны скидочно-балльные промоакции. Они позволяют автоматически начислять и списывать баллы при оформлении заказов, применять скидки и рассчитывать все применяемые бенефиты в соответствии с настроенным арбитражем. Также становится доступен функционал партнерских товаров — для отдельных товаров можно настроить особые условия, например повышенный кешбэк или эксклюзивную скидку для участников программы. Партнерские товары учитываются при расчете заказа через `preorder` / `preorder_commit` наравне с обычными промоакциями.

Лояльность с интеграцией Маркета

Этот вариант подходит для автоматического начисления и списания баллов при оформлении заказов, а также применения скидок через балльные и скидочные промоакции. Для работы этого подхода необходимо:

- Загрузить данные о товарах в Маркет. Загрузить данные можно через API, товарный фид или из файла.
- Создать скидочно-балльные промоакции — именно они будут автоматически применяться при расчете заказа.

Оформление заказов с расчетом применённых бенефитов происходит через двухэтапный API:

- **На первом этапе** вызывается метод `preorder` — он выполняет **предварительный расчет** в момент, когда клиент собирает корзину. В запросе передаются состав заказа и данные участника (email, phone, profile_id). Платформа возвращает итоговую цену со скидками, сумму начисления и списания баллов, а также список сработавших промоакций. Заказ еще не создается на данном этапе — это только расчет.
- **На втором этапе**, после оформления или подтверждения заказа, вызывается метод `preorder_commit`. Платформа **создает заказ**, начисляет баллы на счет участника, списывает использованные баллы и фиксирует все примененные промоакции.

INFO: Обратите внимание

Автоматическое начисление и списание баллов при оформлении заказов работает **только** при использовании двухэтапного API (`preorder` → `preorder_commit`).

Если заказ создается любым другим способом (например, через импорт, прямое добавление в интерфейсе или собственную логику), механизмы лояльности не срабатывают. В таких случаях управление баллами нужно реализовывать отдельно — через сценарии с триггерными промоакциями или прямые вызовы API начисления и списания.

Лояльность без интеграции с Маркетом

В этом варианте начисление и списание баллов выполняется за действия пользователя: заполнение форм, установку приложений, достижения. **Скидочно-балльные промоакции** и связанные с ними функции (тестирование, расчет применения бенефитов в рамках арбитража) **недоступны**.

Управление баллами строится на использовании триггерных промоакций — они задают правила начисления и списания, но не привязаны к заказам. Момент выполнения промоакции определяется либо в сценариях автоматизации (через узел “Управление баллами”), либо через прямые вызовы API лояльности (начисление, списание, регистрация участников). Методы `preorder` и `preorder_commit` в этом варианте не используются — они **доступны только при работе с Маркетом**.

Примеры реализации этого подхода — начисление баллов за установку приложения, заполнение формы и уведомление о сгорании баллов — вы можете найти в статье Базовые кейсы использования программы лояльности.

Базовые кейсы использования программы лояльности

Базовые кейсы использования программы лояльности

Программа лояльности в Altcraft может использоваться для автоматического поощрения пользователей и персонализированных коммуникаций на разных этапах их взаимодействия с продуктом. Ниже приведены базовые практические кейсы, которые показывают, как настраивать начисление баллов, добавление профилей в программу лояльности и отправку уведомлений на основе действий пользователей и состояния их баланса. Все примеры ориентированы на типовые сценарии и могут быть адаптированы под конкретные бизнес-задачи.

CAUTION

Altcraft Platform предоставляет технические инструменты для хранения и обработки данных, включая пользовательские поля, которые могут использоваться для фиксации согласия на участие в программе лояльности. Платформа не несёт ответственности за юридическую корректность способа получения, хранения и использования такого согласия. Вопросы соответствия требованиям законодательства (в том числе в части персональных данных и маркетинговых коммуникаций) пользователь решает самостоятельно, с учётом применимых норм и рекомендаций юридических специалистов.

Поощрение профиля при заполнении формы

Этот кейс позволяет автоматически добавлять пользователя в программу лояльности сразу после того, как он заполнил форму на сайте или в приложении. При этом логика работает корректна как для новых участников, так и для уже зарегистрированных участников программы лояльности, также поощряя их за заполнение формы:

INFO: Обратите внимание

Если в настройках балла **начальный баланс равен нулю**, необходимо после узла “Участие в лояльности” также добавить узел “Управление баллами” с триггерной промоакцией, чтобы начислить бонус за заполнение формы

Лояльность

Перед созданием сценария необходимо подготовить базовую конфигурацию программы лояльности. В разделе **Лояльность — Программы лояльности** создайте саму программу.

Настройте стандартные баллы. При создании стандартного балла необходимо заполнить обязательные параметры:

- название балла и дополнительный идентификатор;
- начальный баланс нового участника программы лояльности;
- срок сгорания баллов (через заданное количество дней, месяцев или лет после начисления);
- соотношение баллов.

Эти параметры определяют, с каким балансом профиль будет добавляться в программу и как дальше будут учитываться начисленные бонусы:

Триггерная промоакция используется как универсальный механизм начисления или списания баллов. Несмотря на название, собственного триггера у такой промоакции нет — момент её выполнения определяется сценарием.

Настройте правила начисления баллов.

- Выберите созданный балл;
- Задайте диапазон или фиксированное количество:

> **INFO**
>
> Если указан диапазон, точное количество начисляемых баллов определяется уже в настройках узла сценария. Это позволяет использовать одну и ту же триггерную промоакцию для разных кейсов, например для начисления 500 баллов за скачивание мобильного приложения и 300 баллов за заполнение формы, не создавая отдельные промоакции под каждый случай.

После сохранения и активации программы лояльности триггерная промоакция становится доступной в сценариях автоматизации. В сценарии она добавляется как узел Управление баллами, и именно там определяется, в какой момент она должна выполниться и какое точное количество баллов будет начислено или списано.

База данных и форма

В **базе профилей** создайте **дополнительное поле** типа **Логический**, которое будет фиксировать согласие пользователя на участие в программе лояльности:

Затем создайте Форму и добавьте в неё это логическое поле. **Поле рекомендуется сделать обязательным,** чтобы профиль не мог отправить форму без подтверждения согласия. После публикации формы данные из неё будут сохраняться в профиль пользователя, включая значение поля согласия:

Сценарий

Когда программа лояльности и триггерная промоакция уже настроены, можно переходить к созданию сценария. В сценарии отключите повторный вход профилей, чтобы один и тот же профиль не мог попасть в сценарий повторно. В качестве триггера входа выберите **Заполнение формы**:

В сценарии добавьте элемент проверки условия, который проверяет участие профиля в программе лояльности:

Если профиль уже состоит в программе, элемент “Участие в лояльности” не используется, а сразу вызывается триггерная промоакция, которая начисляет, например, 500 бонусных баллов за заполнение формы:

Если профиль еще не является участником программы лояльности, сценарий выполняет добавление профиля в программу. Так как поле согласия в форме является обязательным, дополнительная проверка согласия не требуется:

При добавлении профиля в программу баллы начисляются автоматически — в соответствии с установленным значением начального баланса стандартного балла.

INFO: Обратите внимание

Если в настройках балла при регистрации указан **нулевой начальный баланс**, необходимо добавить узел **“Управление баллами”** (по аналогии с предыдущим), чтобы вручную начислить требуемое количество баллов.

Автоматическое поощрение профиля за установку мобильного приложения

Этот кейс позволяет автоматически добавлять пользователя в программу лояльности или начислять баллы сразу после того, как он установил мобильное приложение. Для этого используется **пиксель с целью**, который срабатывает при установке приложения.

INFO: Обратите внимание

Если в настройках балла **начальный баланс равен нулю**, необходимо после узла “Участие в лояльности” также добавить узел “Управление баллами” с триггерной промоакцией, чтобы начислить бонус за скачанное приложение.

Лояльность

Перед созданием сценария необходимо подготовить базовую конфигурацию программы лояльности. В разделе **Лояльность — Программы лояльности** создайте саму программу.

Настройте стандартные баллы. При создании стандартного балла необходимо заполнить обязательные параметры:

- название балла и дополнительный идентификатор;
- начальный баланс нового участника программы лояльности;
- срок сгорания баллов (через заданное количество дней, месяцев или лет после начисления);
- соотношение баллов.

Эти параметры определяют, сколько баллов получит профиль после установки приложения:

Триггерная промоакция используется как универсальный механизм начисления или списания баллов. Момент её выполнения определяется сценарием.

Настройте правила начисления баллов:

- Выберите созданный балл;
- Задайте диапазон или фиксированное количество.

INFO

Если указан диапазон, точное количество начисляемых баллов определяется в сценарии в момент вызова промоакции. Это позволяет использовать одну и ту же триггерную промоакцию для разных кейсов: например, 300 баллов за установку iOS и 500 баллов за установку Android.

После сохранения и активации программы лояльности триггерная промоакция становится доступной в сценариях автоматизации. В сценарии она добавляется как узел Управление баллами, где определяется момент начисления баллов.

Пиксель

Создайте **пиксель** для мобильного приложения. Пиксель используется с одной целью — фиксировать факт установки приложения.

- Тип события: **Цель установки приложения**.
- Активируется автоматически при установке.
- Для корректной работы убедитесь, что пиксель интегрирован в мобильное приложение через mSDK.

Сценарий

Когда программа лояльности и триггерная промоакция уже настроены, создайте сценарий. В сценарии отключите повторный вход профилей, чтобы один и тот же профиль не мог попасть в сценарий повторно.

В качестве триггера входа выберите **Событие пикселя — установка приложения**:

В сценарии добавьте элемент проверки условия, который проверяет, состоит ли профиль уже в программе лояльности:

Если профиль уже участвует в программе, элемент “Участие в лольяности” не используется, а сразу вызывается триггерная промоакция для начисления баллов (например, 500 баллов за установку приложения):

Если профиль ещё не является участником программы лояльности, сценарий выполняет добавление профиля в программу. После этого автоматически начисляются стартовые баллы за счет настроенного начального баланса стандартного балла.

Уведомление о скором сгорании баллов программы лояльности

В этом кейсе пользователи получают **Уведомление** о том, что их баллы в программе лояльности скоро сгорят. Для этого используется **динамический сегмент**, который отбирает профили с ближайшим сгоранием баллов, и **регулярная рассылка**, выполняемая по расписанию.

Сегмент

Создайте **динамический сегмент** для профилей, у которых баллы скоро сгорят.

В условиях сегмента используйте фильтр **Взаимодействие с сущностями** — **Лояльность** — **Сгорание балов**

- Условие: **в ближайшие [X] дней**
- Значение **X**: `10`
- Программа лояльности: выберите нужную программу
- Балл: укажите необходимый балл

Такой сегмент автоматически включает профили, у которых в течение ближайших 10 дней ожидается сгорание баллов указанной валюты.

Шаблон SMS

Создайте **шаблон SMS-сообщения**, в котором будет отображаться информация о ближайшем сгорании баллов.

Для вывода данных используйте переменную: `{loyalty_expiration_points(<ID_программы_лояльности> "<ID_баллов>" x y "desc")}`

Переменная возвращает информацию о ближайшем сгорании баллов по выбранной валюте:

- `<ID_программы_лояльности>` — идентификатор программы лояльности;
- `<ID_баллов>` — SID балла;
- `x` — период активности баллов;
- `y` — лимит количества транзакций;
- `"desc"` — сортировка по дате транзакций (по убыванию).

Пример текста SMS:

```
{for $i $p = loyalty_expiration_points(2 "bonus1" 10 1 "desc")}{lead._fname}, {$p.Amount} бонусов сгорят через 10 дней. Потратьте их сейчас!
{end}
```

Рассылка

Создайте **регулярную рассылку**, в качестве аудитории выберите созданный динамический сегмент. Выберите SMS-шаблон с информацией о сгорании баллов. Настройте расписание (например “Каждый день в выбранное время” “Отправлять в часовом поясе профиля, если он известен”)

После активации рассылки система будет ежедневно отправлять уведомления только тем профилям, у которых баллы действительно находятся в зоне сгорания.

INFO

Использование динамического сегмента и регулярной рассылки позволяет уведомлять пользователей о сгорании баллов без сценариев и ручного контроля. Уведомления отправляются только актуальной аудитории и автоматически прекращаются после изменения условий.

Промокоды

Программа лояльности Altcraft позволяет создавать персональные или универсальные промокоды, которые затем можно применять в различных механиках управления лояльностью клиентов:

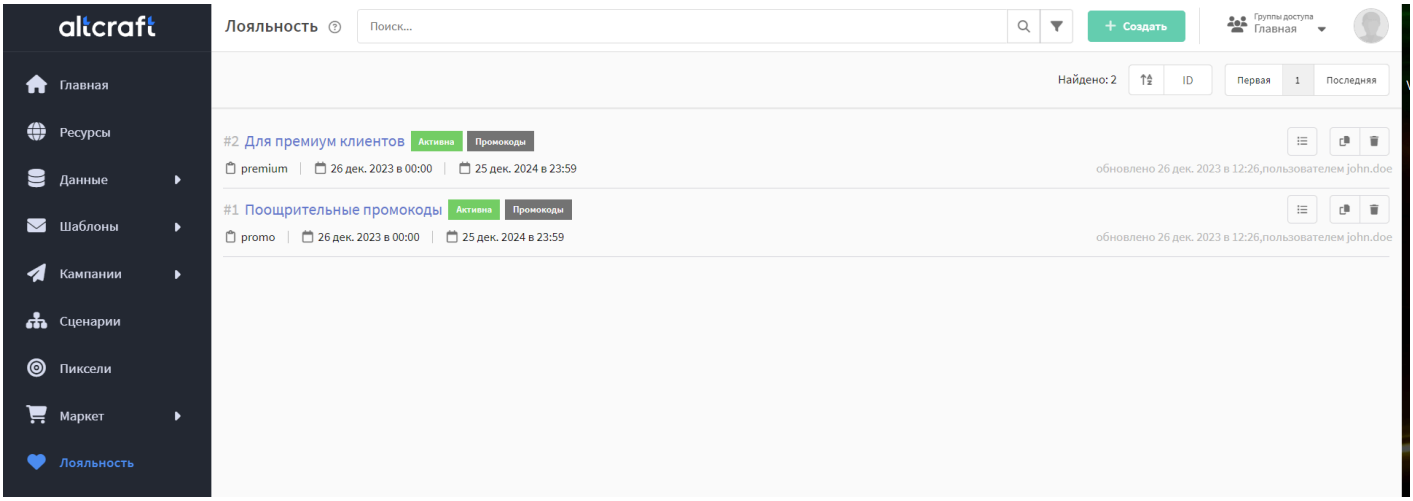
- **Персональный** промокод **привязывается** к профилю конкретного клиента для предоставления личных акций, скидок, предложений.
- **Универсальный** промокод может быть привязан к нескольким профилям клиентов. Может быть использован при организации массовых акций с общими условиями.

Назначение профилям клиентов промокодов доступно вручную и автоматически (при помощи рассылок и сценариев Altcraft). Промокоды можно сгенерировать в интерфейсе **Программ лояльности** или импортировать из файла.

> **TIP**
>
> Расширенное управление промокодами доступно по API. Например, можно отслеживать клиентов, которые воспользовались предложением. Документация API программ лояльности **доступна по этой ссылке.**

Чтобы перейти к управлению промокодами, в главном меню перейдите в раздел **Лояльность**. Откроется список с существующими программами лояльности. Для каждой доступно контекстное меню, с помощью которого вы сможете клонировать или удалить запись, активировать или остановить программу, а также просмотреть список созданных промокодов. Для редактирования программы нажмите на её заголовок.

Чтобы начать новую программу лояльности, вверху страницы нажмите кнопку **+ Создать**:



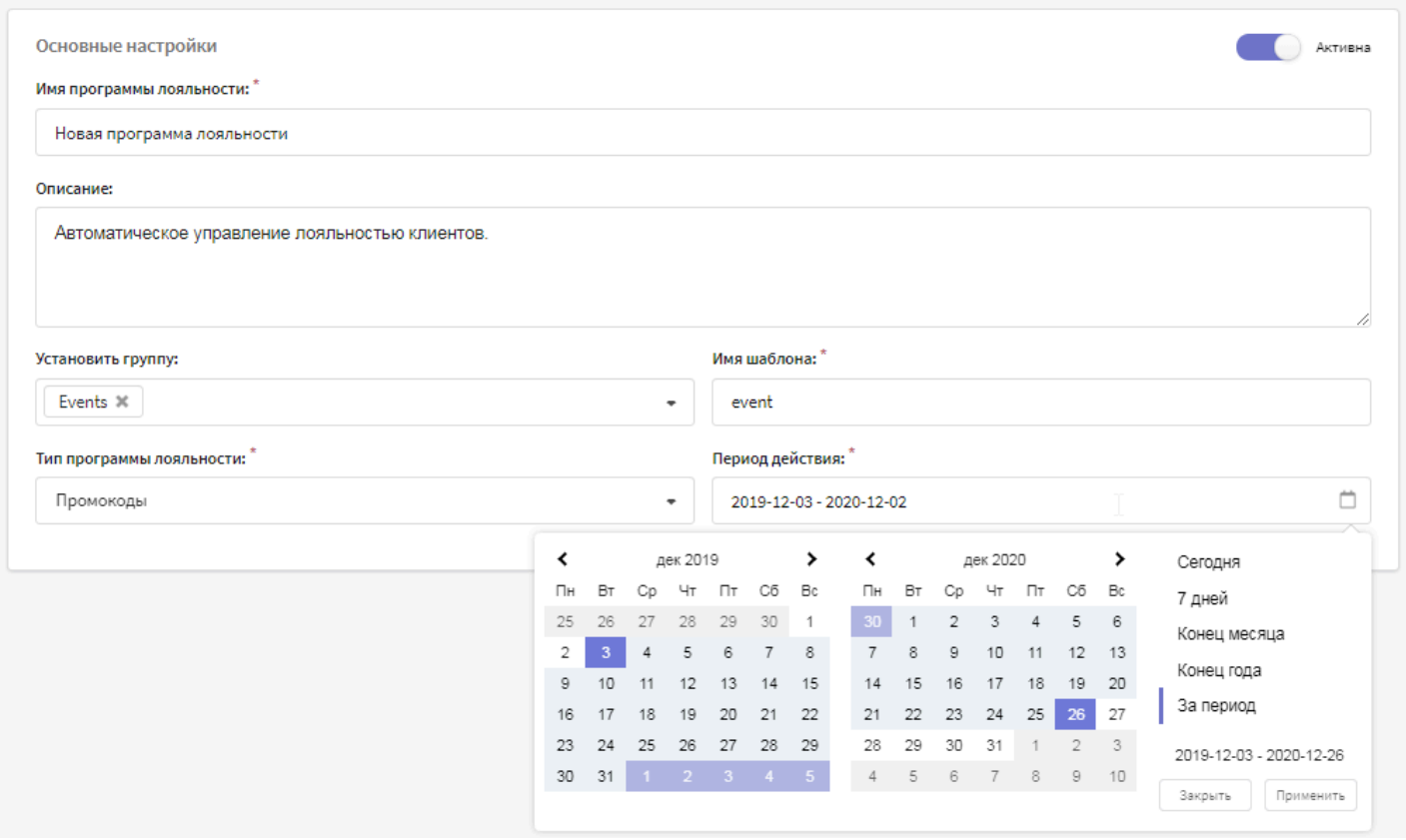
Данные промокода

Каждый промокод помимо собственно кода для отправки клиентам содержит следующие данные:

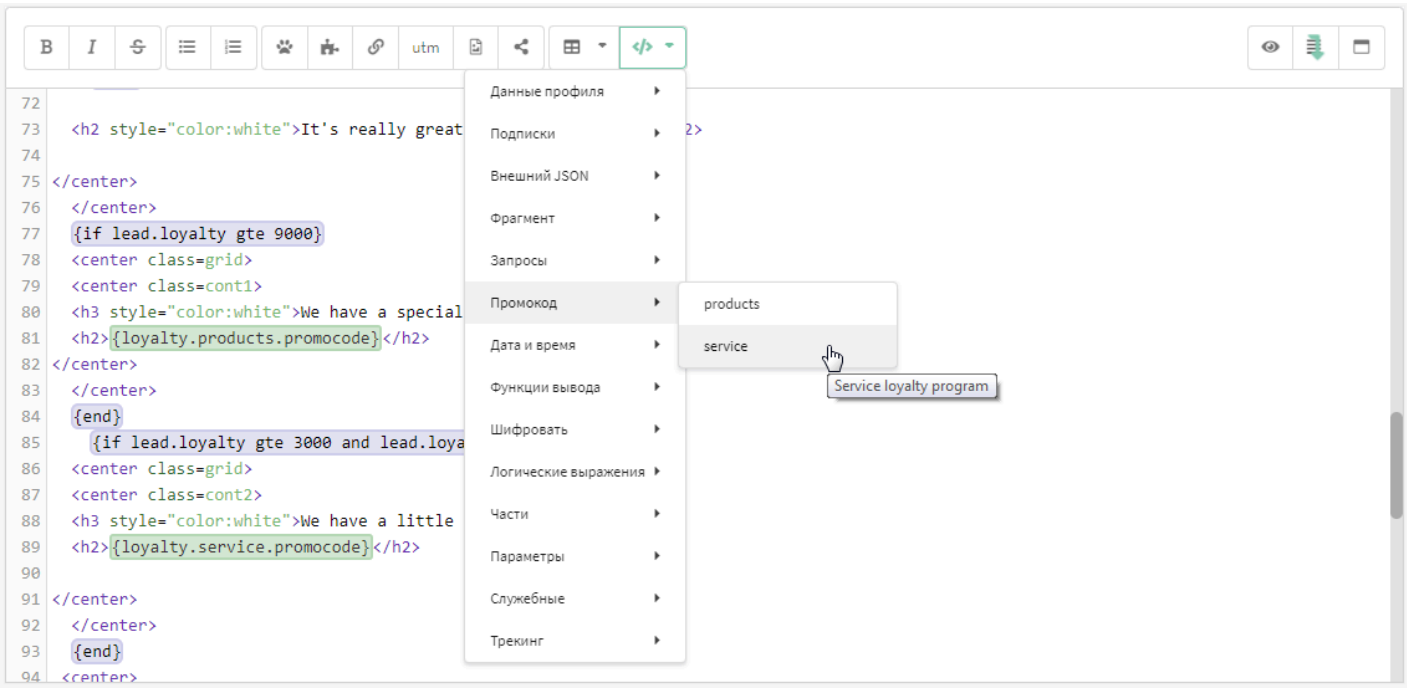
- дата создания и ID программы лояльности
- признак и дата привязки промокода к профилю клиента
- срок действия, информация об истечении срока
- признак и дата активации кода клиентом
- дополнительные поля данных

Основные настройки

В основных настройках укажите **Имя программы лояльности** и введите **Описание**. Также при необходимости **установите группу** доступа:



В поле **Имя шаблона** введите название программы лояльности для использования в [шаблонах сообщений](). В редакторе шаблона программа лояльности будет доступна в меню переменных **{ </> }** или макросом {loyalty.**name**.promocode}:



Период действия позволит ограничить возможность создания и отправки промокодов в рассылках и сценариях. Этот параметр не относится к времени действия уже отправленных клиентам промокодов. Установите необходимый диапазон дат и нажмите **Применить**.

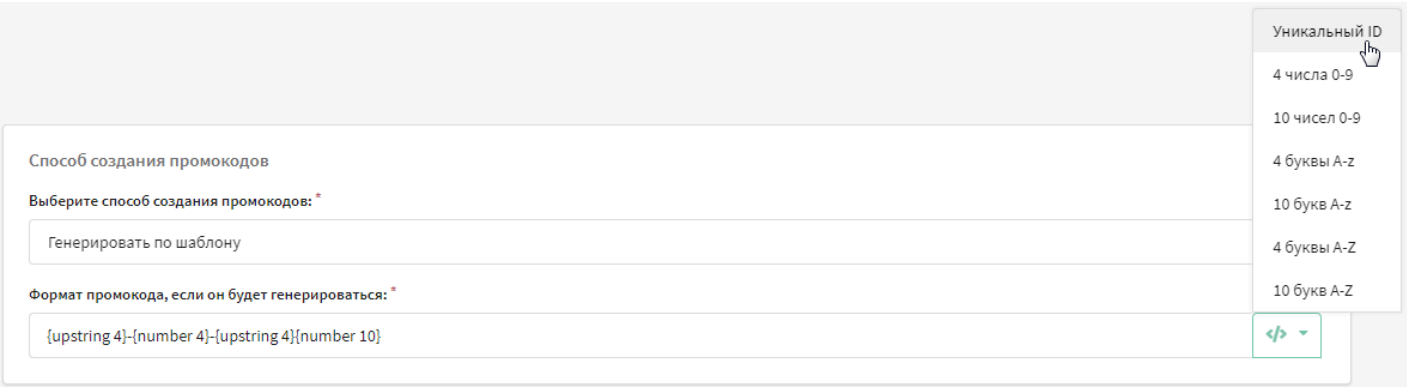
Генерация промокодов

> **CAUTION**
>
> Если вы планируете загрузить готовый список промокодов из файла, можете пропустить этот пункт.

В настройках типа генерации задается способ создания промокодов: средствами Altcraft или с использованием внешнего сервиса.

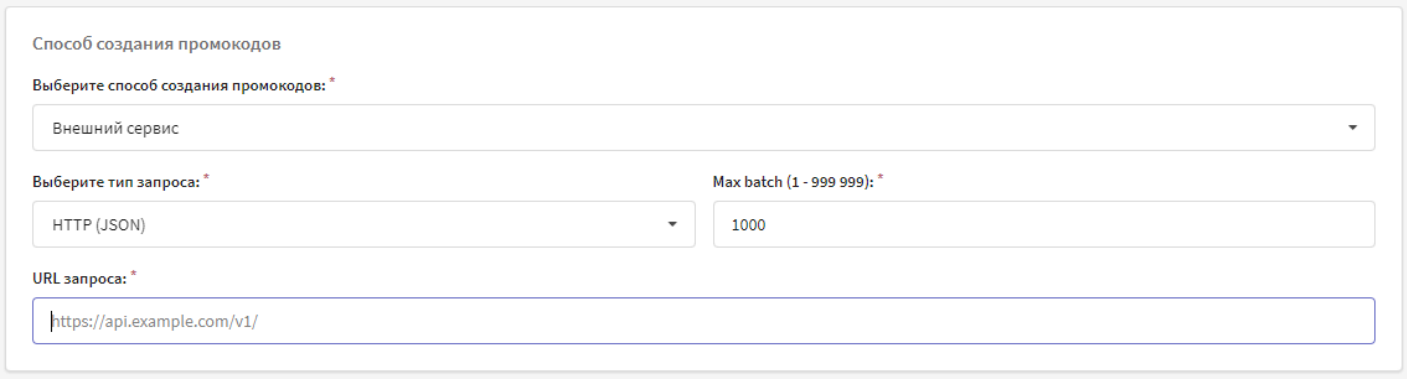
Шаблон для генерации промокодов

Если у вас нет действующей системы управления промокодами или фиксированного списка кодов, вы можете **сгенерировать их в интерфейсе** Альткрафт. Выберите необходимую комбинацию символов из выпадающего меню **{** `</>` **}**:



Обращение к внешнему сервису

Альтернативой является запрос к **внешнему сервису**, который создаст промокоды на вашей стороне:



Указывается **URL** сервиса и максимальное количество промокодов в одном запросе. Используется **POST** запрос с **JSON** данными, формат строго определен:

```
{
    account_id: Число (Идентификатор аккаунта),
    loyalty_id: Число (Идентификатор программы лояльности),
    loyalty_name: Строка (Название программы лояльности),
    loyalty_shortname: Строка (Короткое имя программы лояльности),
    campaign_id: Число (Идентификатор рассылки, если она запрашивает промокоды),
    campaign_name: Строка (Имя рассылки, если она запрашивает промокоды),
    workflow_id: Число (Идентификатор сценария, если он запрашивает промокоды),
    workflow_name: Строка (Имя сценария, если он запрашивает промокоды),
    count: Число (Количество промокодов, которое необходимо сгенерировать),
    is_test: Bool (Признак теста),
    fields: {
        <Имя поля>: <Значение>
    }
}
```

Если промокоды запрашивает **рассылка** или **сценарий**, добавьте их идентификаторы в запрос. Если передан флаг **is_test**, то промокод не должен быть настоящим, он просто используется для предпросмотра. Все поля, определенные в программе лояльности, будут передаваться вместе с запросом.

В ответе ожидается **список** промокодов, тоже в **JSON**:

```
{
    codes: Массив строк (промокоды)
}
```

Если есть необходимость с промокодами вернуть **дополнительные поля**, то промокоды нужно возвращать в виде массива обьектов:

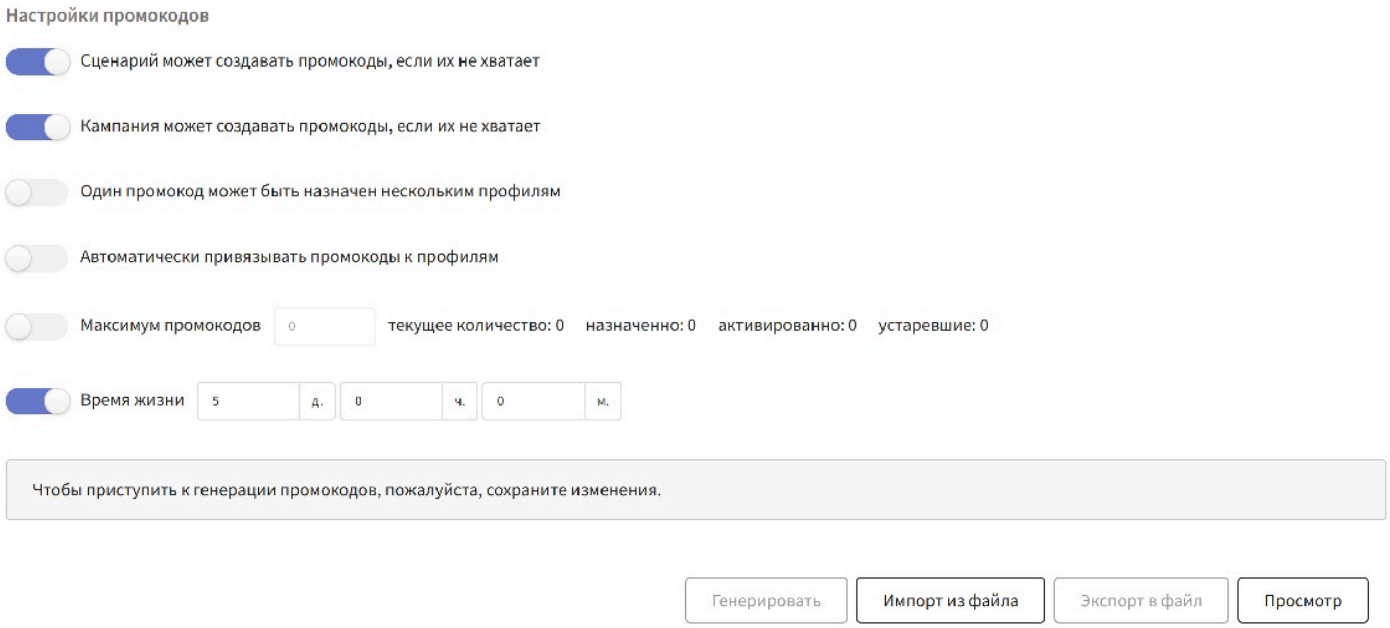
```
{
    codes: Массив обьектов {
        code: Строка (промокод),
        fields: {
            <Имя поля>: <Значение>
        }
    }
}
```

TIP

Для проверки сервиса можно использовать кнопку **Генерировать** в блоке **Настройки промокодов**, при этом на сервис будет отправлен запрос с флагом is_test. Если сервис не отвечает или отвечает HTTP ошибкой, то система делает еще несколько попыток и затем возвращает ошибку.

Настройки промокодов

Сконфигурируйте программу лояльности под вашу бизнес-логику в блоке **Настройки промокодов**. Помимо настроек в этом блоке доступны команды генерации, импорта, экспорта и просмотра промокодов.



Доступны следующие настройки:

- **Сценарий может создавать промокоды, если их не хватает** — разрешает сценариям автоматизации генерировать промокоды по шаблону. В сценарии нужно добавить шаг **Назначить промокод**. Сами коды генерировать в таком случае нет необходимости.
- **Рассылка может создавать промокоды, если их не хватает** — разрешает рассылкам генерировать промокоды по шаблону. При отправке клиенту шаблона с использованием переменной программы лояльности будет сгенерирован и отправлен новый промокод. Сами коды генерировать в таком случае нет необходимости.

CAUTION

При тестировании рассылки программа лояльности генерирует новый промокод, а не расходует уже созданный.

- **Один промокод может быть назначен нескольким профилям** — с помощью этой опции вы можете использовать один промокод для группы клиентов. Рассылка или сценарий в таком случае будут назначать первый активный промокод программы лояльности.
- **Автоматически привязывать промокоды к профилям** — при генерации или импорте из файла промокоды будут назначены определённым профилям.
- **Максимум промокодов** — общее возможное количество промокодов в программе лояльности.
- **Время жизни** — срок действия промокодов программы лояльности. По его истечении код считается истёкшим и не может быть прикреплён, отправлен или активирован.

CAUTION

Время жизни промокода может быть дольше, чем период действия программы лояльности. Если программа завершена, коды нельзя будет назначить и отправить клиентам, однако отправленные коды могут быть использованы клиентами, пока срок их действия не истёк.

Автоматическая привязка промокодов к профилю

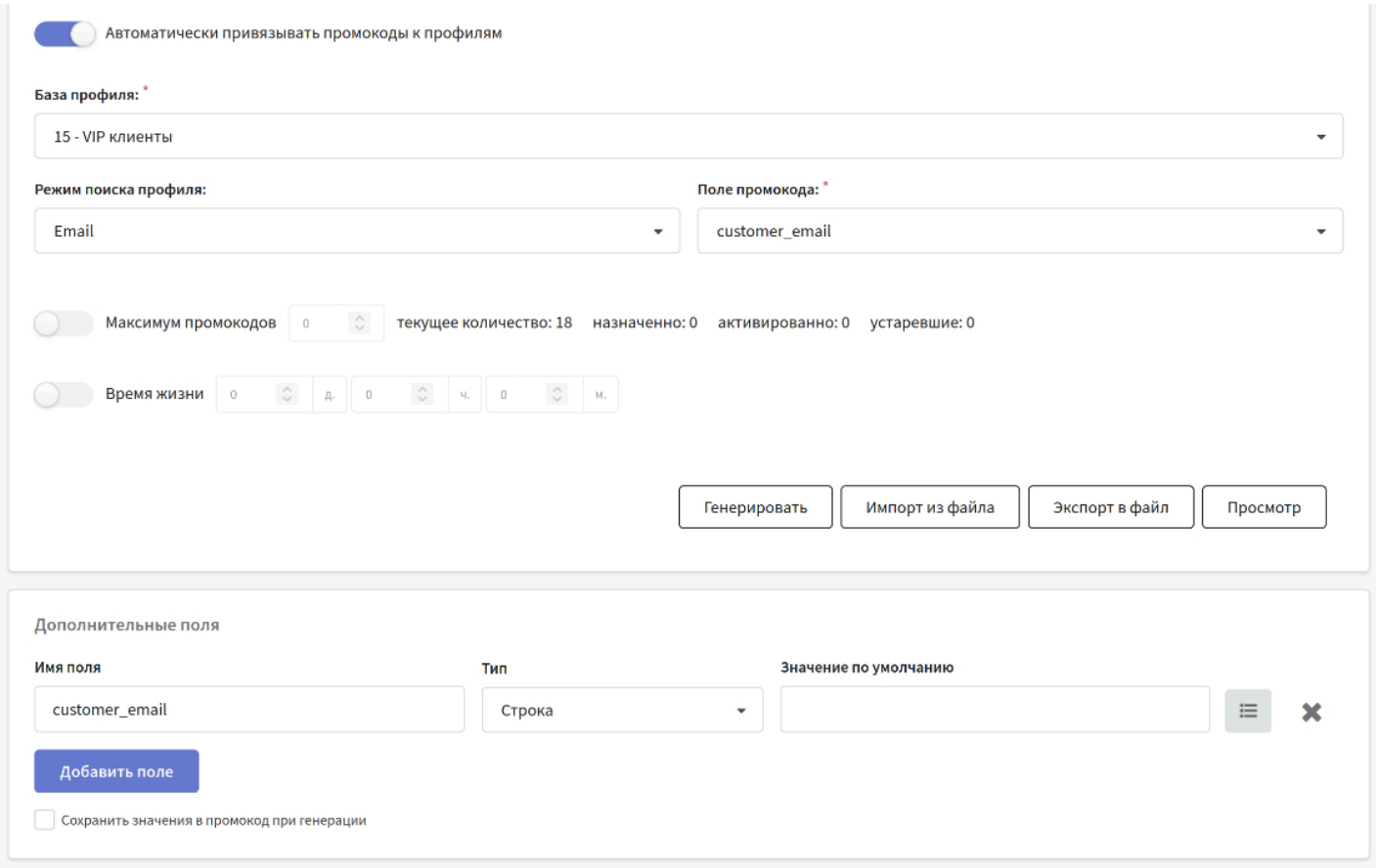
Выберите **базу** с профилями, которым необходимо назначить промокоды. Далее укажите **режим поиска профиля** и **поле промокода**. Режим поиска определяет, по каким данным платформа будет искать профили. Доступны следующие режимы:

- Email — по электронному адресу.
- Phone — по номеру телефона.
- By custom field — по значению дополнительного поля.

Если значение в поле промокода совпадёт со значением поля, указанного в качестве режима поиска, то платформа привяжет промокод найденному профилю.

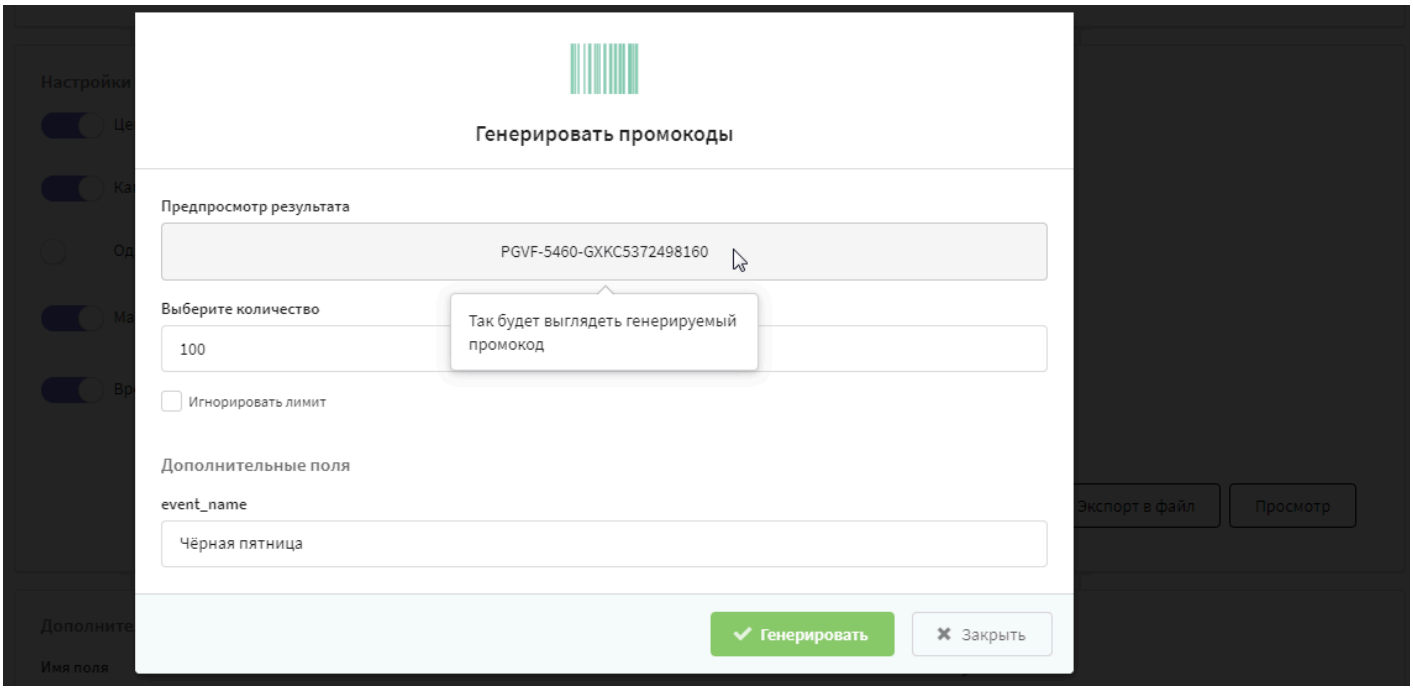
TIP

Поле промокода создаётся в блоке **Дополнительные поля** внизу страницы.



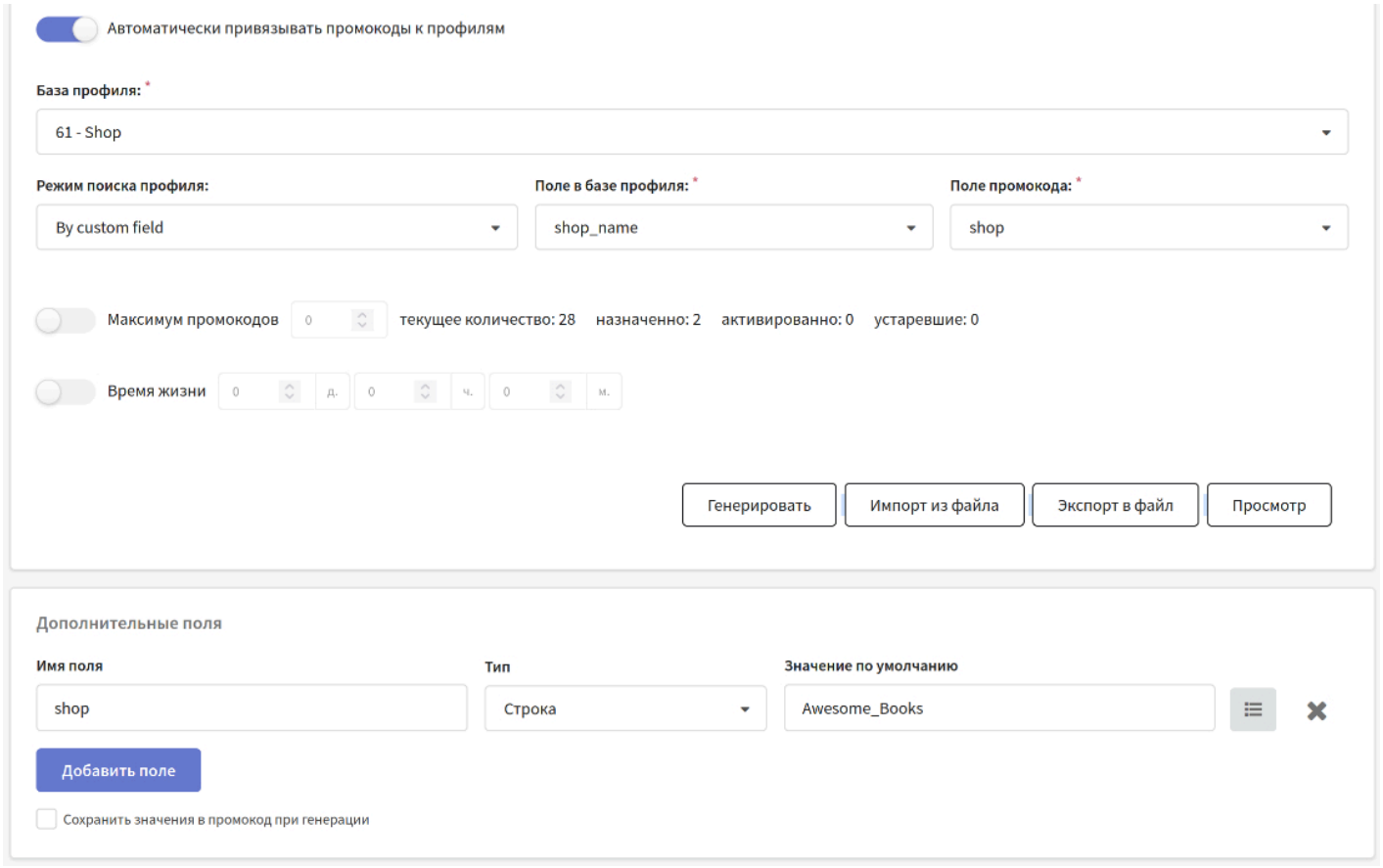
Генерация промокодов по шаблону:

При нажатии кнопки **Генерировать** откроется модальное окно с предпросмотром промокода по установленному шаблону. Будет предложено выбрать количество кодов и заполнить дополнительные поля, если они есть в программе лояльности.



Чтобы автоматически привязать промокоды профилям при генерации, используйте режим поиска по дополнительному полю. Создайте дополнительное поле промокода и выберите его в настройках матчинга. Укажите значение по умолчанию. Если значение дополнительного поля в базе совпадёт со значением дополнительного поля промокода, промокод будет назначен найденному профилю.

В примере ниже промокоды будут привязаны профилям, у которых дополнительное поле “`shop_name`” содержит значение “`Awesome_Books`”:



TIP

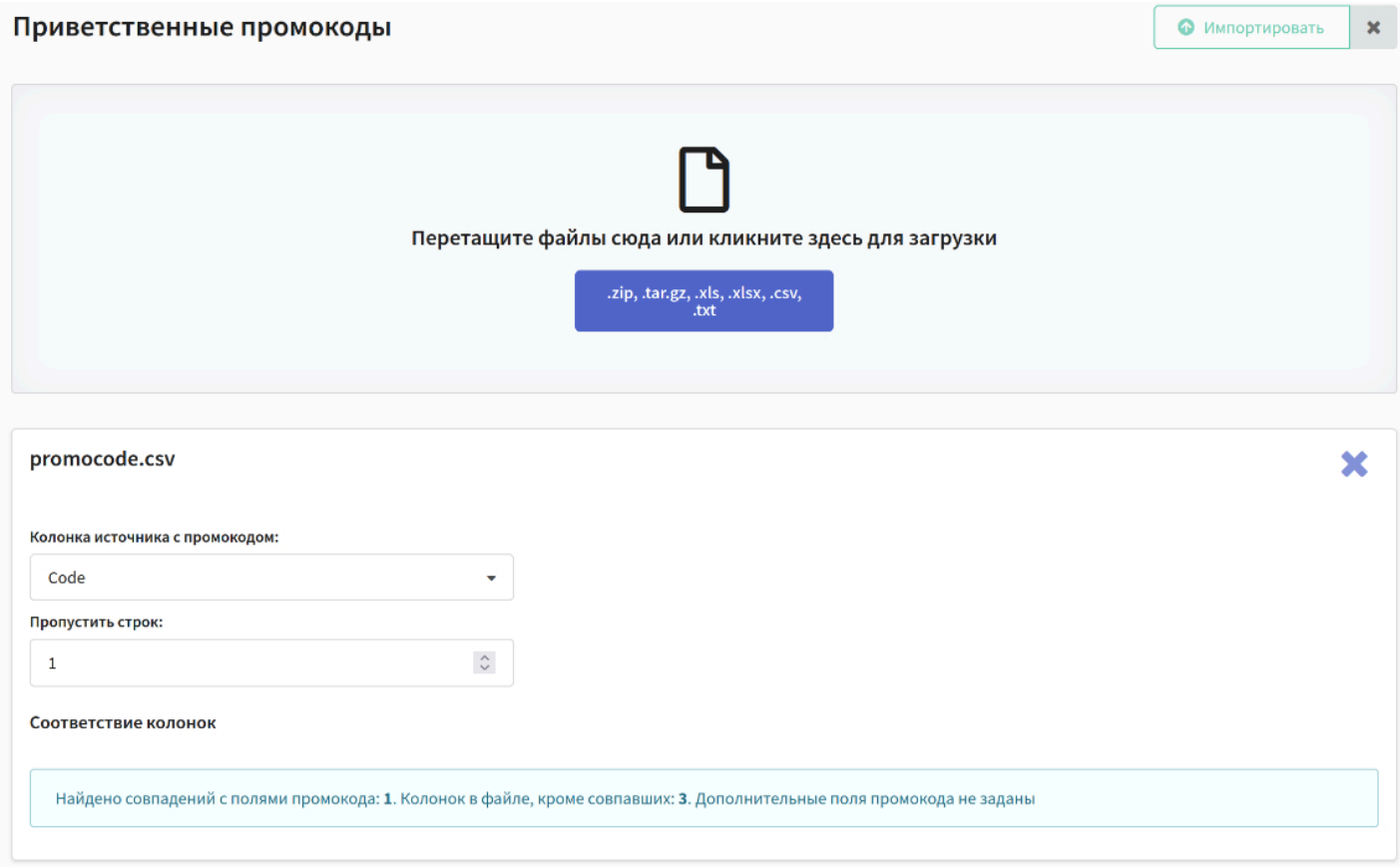
После генерации, в зависимости от количества промокодов, они сразу обновятся в интерфейсе или будет создана задача на их создание.

Импорт из файла

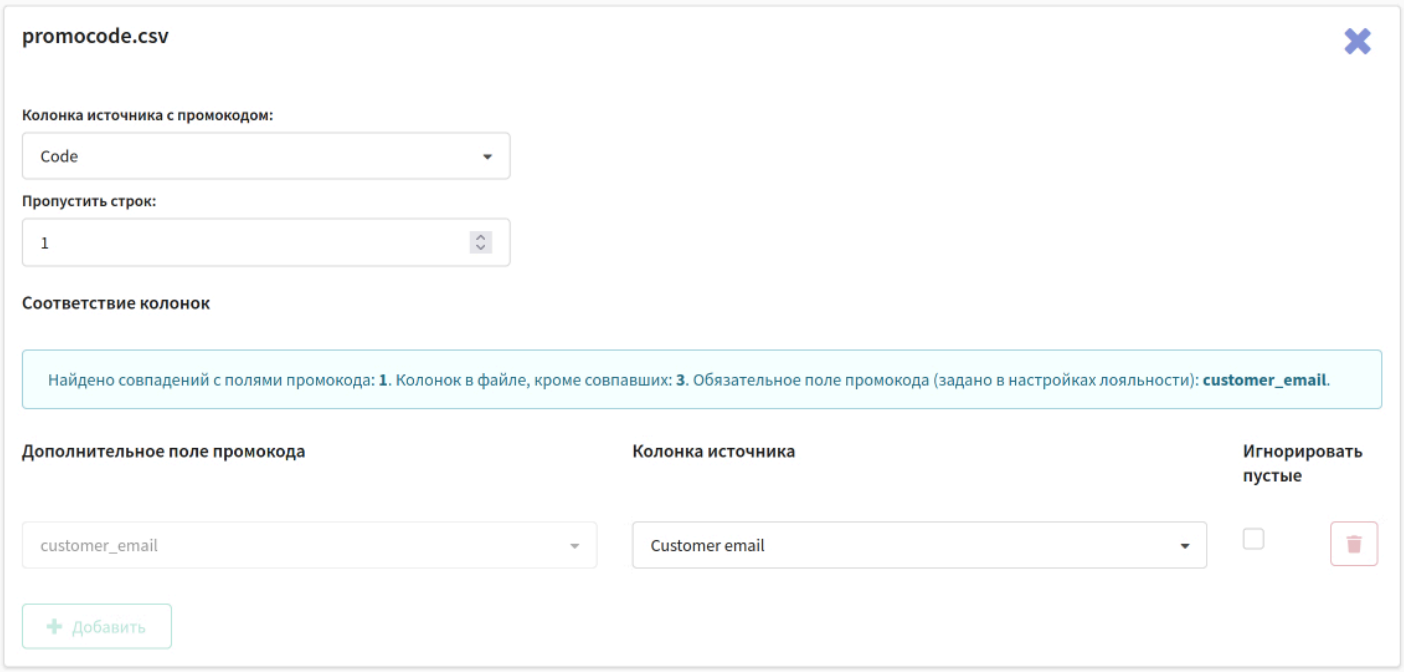
TIP

Кнопка импорта доступна только после сохранения программы лояльности.

При клике на “**Импорт из файла**” будет предложено закачать колоночный файл (CSV, XLS) с промокодами в систему, выбрать колонку с промокодами и запустить импорт кнопкой вверху страницы:



Чтобы автоматически привязать промокоды профилям в процессе импорта из файла, настройте соответствие колонок. В качестве колонки источника выберите колонку в файле, по которой будет происходить поиск нужного профиля:

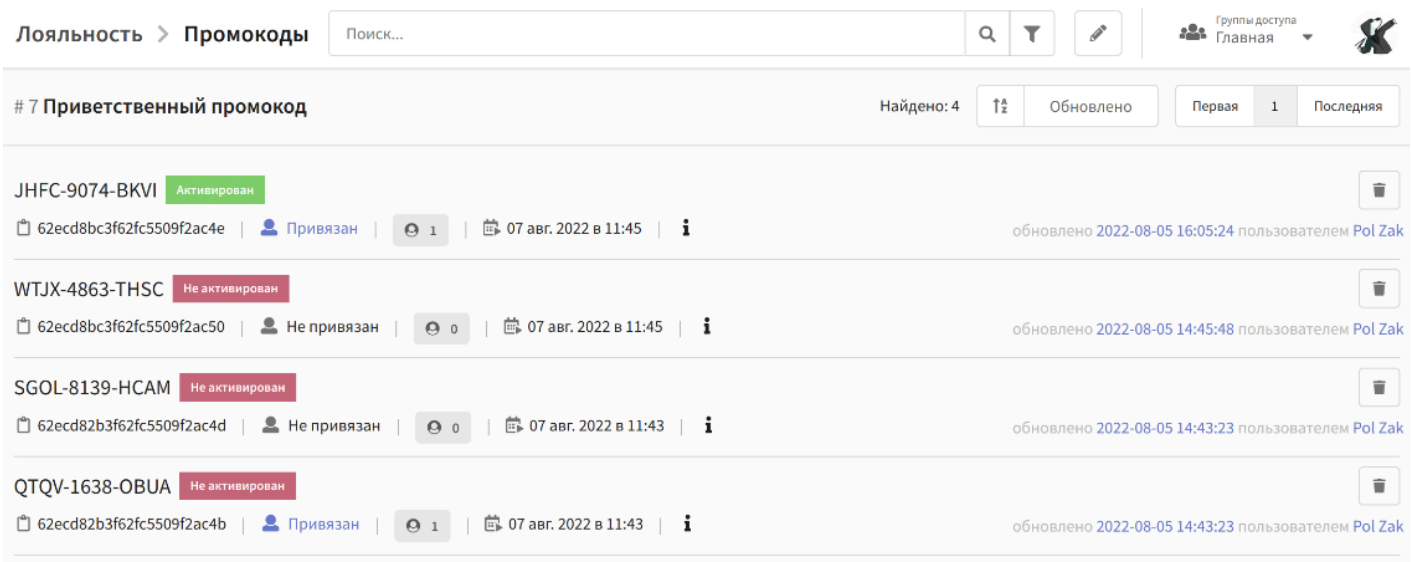


Экспорт в файл

При **Экспорте** вместе с промокодами можно выгрузить поля профилей, которым эти коды назначены. Выберите поля из выпадающего меню и нажмите **OK**, чтобы экспортировать файл в формате CSV:

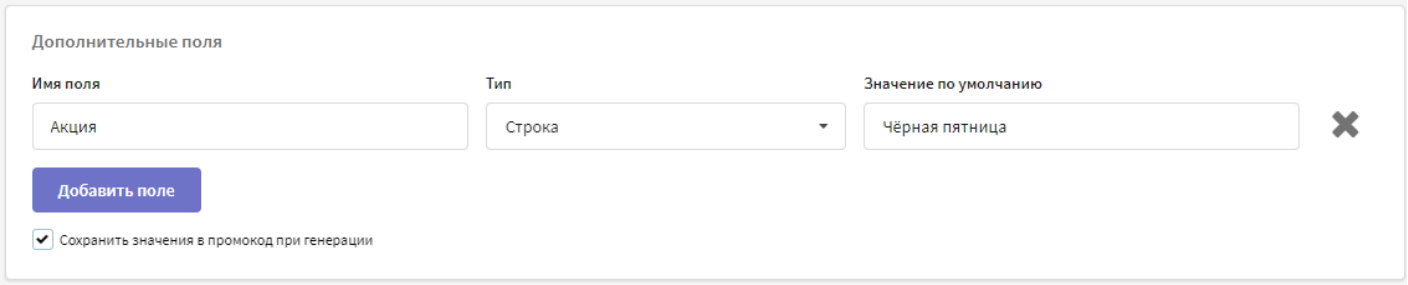
Просмотр

По нажатию кнопки **Просмотр** откроется список созданных в программе лояльности промокодов. По каждому будет отображаться краткая информация. Из списка можно удалить любой промокод вручную:



Дополнительные поля данных

Дополнительные поля запрашиваются при генерации и будут отправлены на внешний сервис, если он используется. При необходимости эти поля могут быть сохранены в промокод. Если внешний сервис возвращает поля с такими же названиями, то они также будут обновлены в промокоде.



Формы

Формы — это инструмент для сбора данных от ваших клиентов. С их помощью вы можете реализовывать различные механизмы с использованием внешних данных, например, добавлять новые профили в вашу базу, управлять подписками на различные ресурсы и получать обратную связь от клиентов.

Для форм доступен специальный конструктор, настройки времени публикации и настройки действия, выполняемого при активации формы.

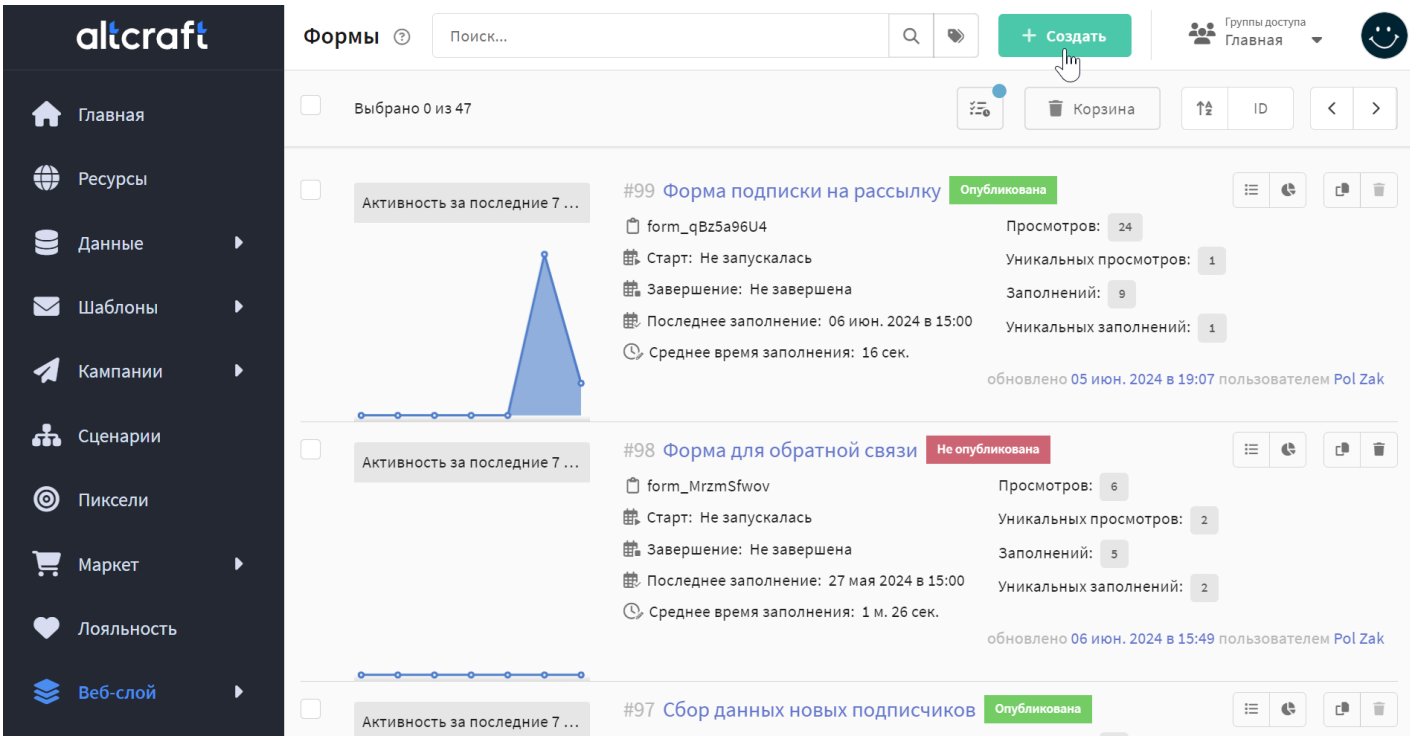
TIP

Видео-гайд по созданию формы доступен в **центре поддержки**.

В этом разделе

Создание формы

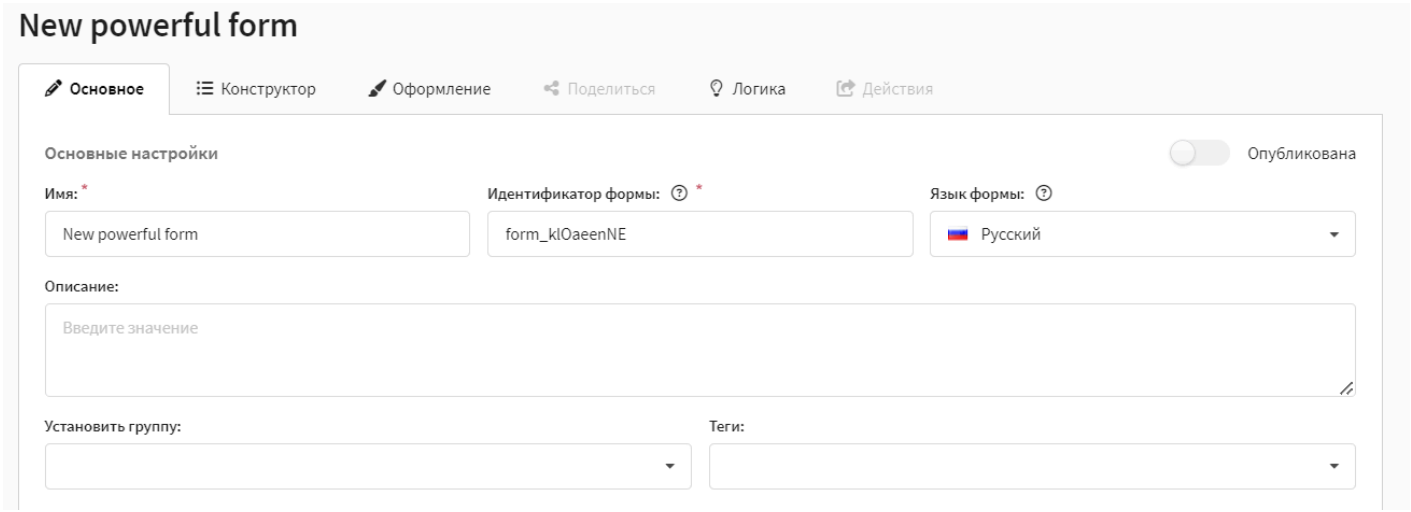
Для создания формы зайдите в раздел **“Формы”** в главном меню и нажмите на кнопку **“Создать”**.



Основные настройки

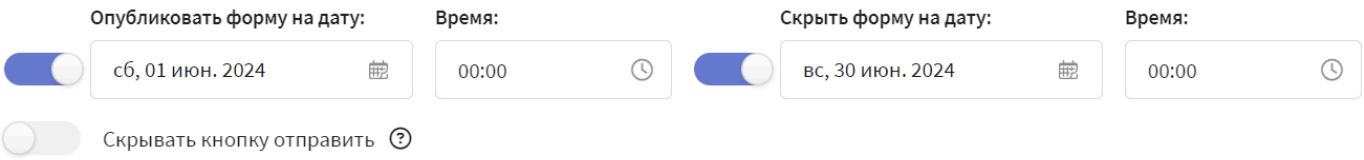
На первой вкладке **“Основные”** доступны главные настройки формы: **имя**, **описание**, **идентификатор**, **язык**. Для формы можно установить **группу** доступа и задать **теги**.

Чтобы форма стала доступна по URL-адресу или как iframe-элемент, необходимо активировать переключатель **“Опубликована”** в правом верхнем углу. Обратите внимание, что пока форма не снята с публикации, её нельзя удалить.



Также на этой вкладке можно настроить **дату и время начала и окончания публикации формы**, если она должна иметь временный характер. Форма станет доступна для заполнения в указанную дату и время и будет снята с публикации, если установлена дата скрытия. При этом переключатель **“Опубликована”** в правом

верхнем углу должен оставаться неактивным. Его состояние автоматически будет обновляться при наступлении указанных дат.



INFO

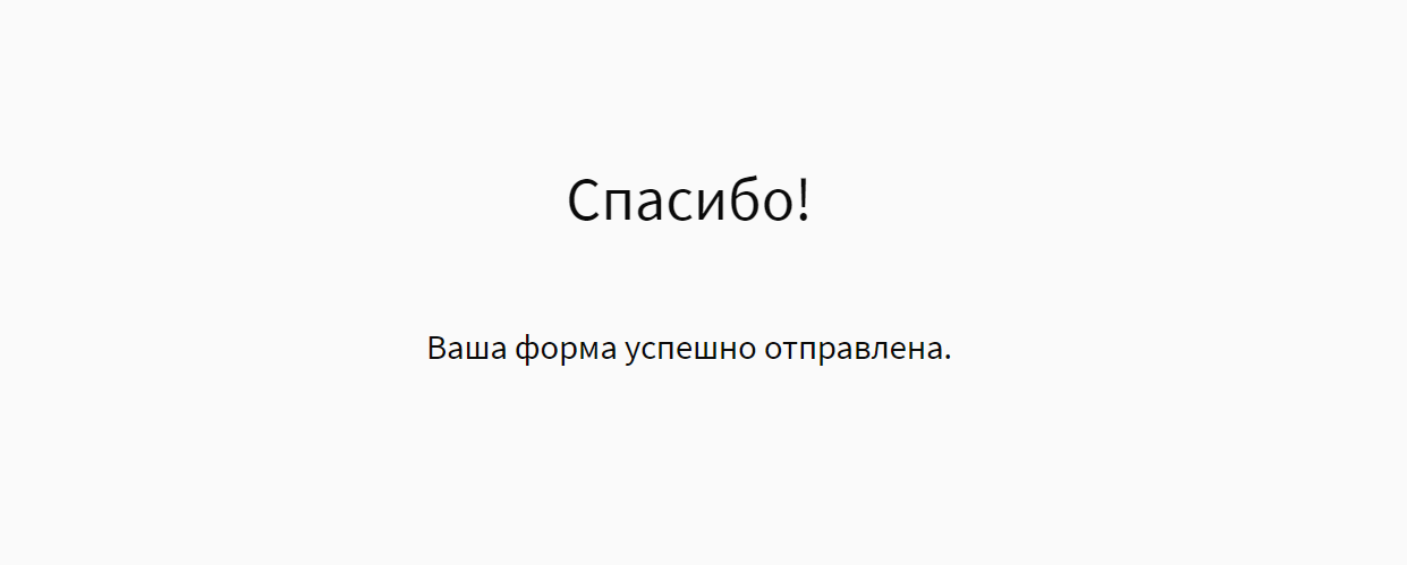
При загрузке формы система использует кэш для её хранения. Длительность хранения формы в кэше составляет 3 минуты. Это может привести к задержке в публикации формы до 3 минут после запланированного времени. Аналогично, форма может оставаться доступной в течение 3 минут после её снятия с публикации.

Когда форма не опубликована, пользователи, пытающиеся получить к ней доступ по URL-адресу или через iframe, будут автоматически перенаправлены на специальную страницу с сообщением **“Форма недоступна”**.

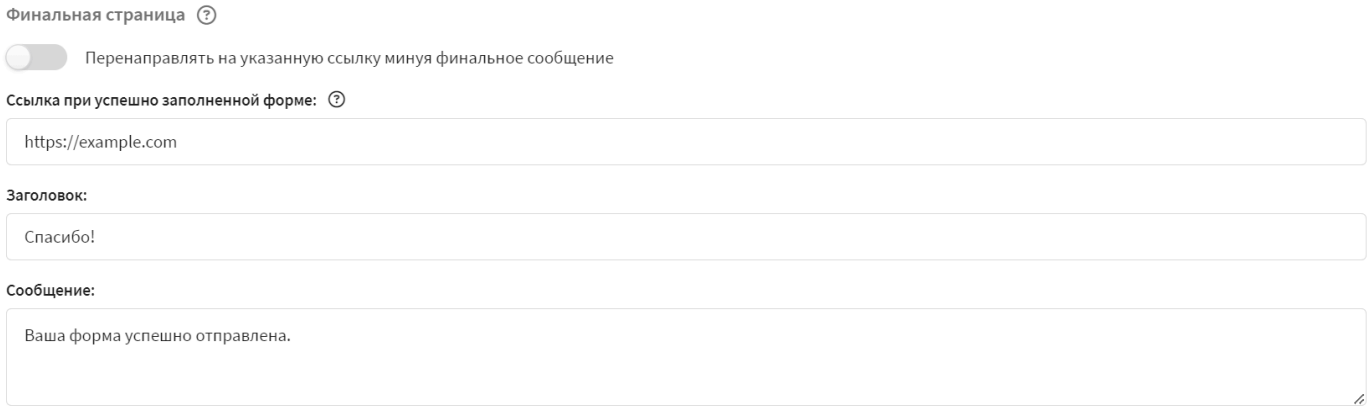
Если вы не хотите показывать пользователю кнопки “Отправить” и “Далее”, пока он не заполнит все обязательные поля на странице с формой, то активируйте опцию **“Скрывать кнопку отправить”**.

Настройка финальной страницы

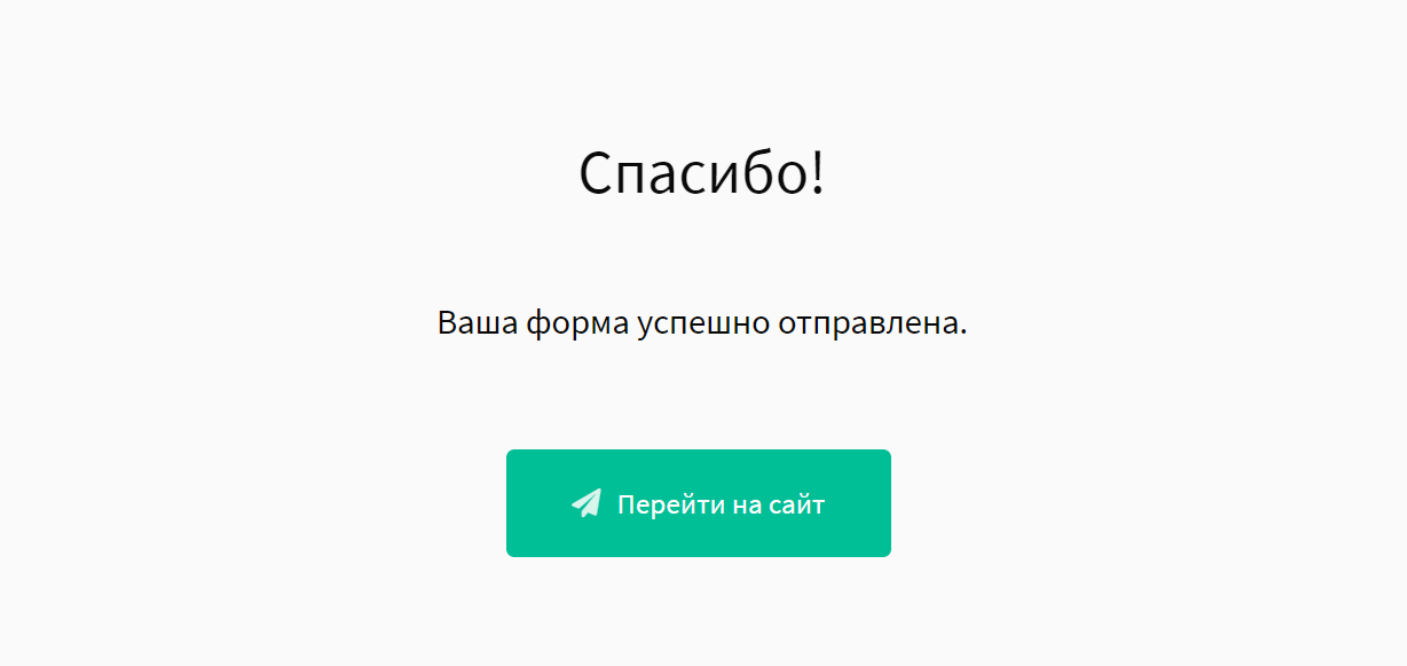
Финальная страница — это страница, которую увидит пользователь после того как успешно отправит заполненную форму. По умолчанию страница выглядит так:



Вы можете изменить **Заголовок** и **Сообщение** на этой странице, заполнив соответствующие поля:



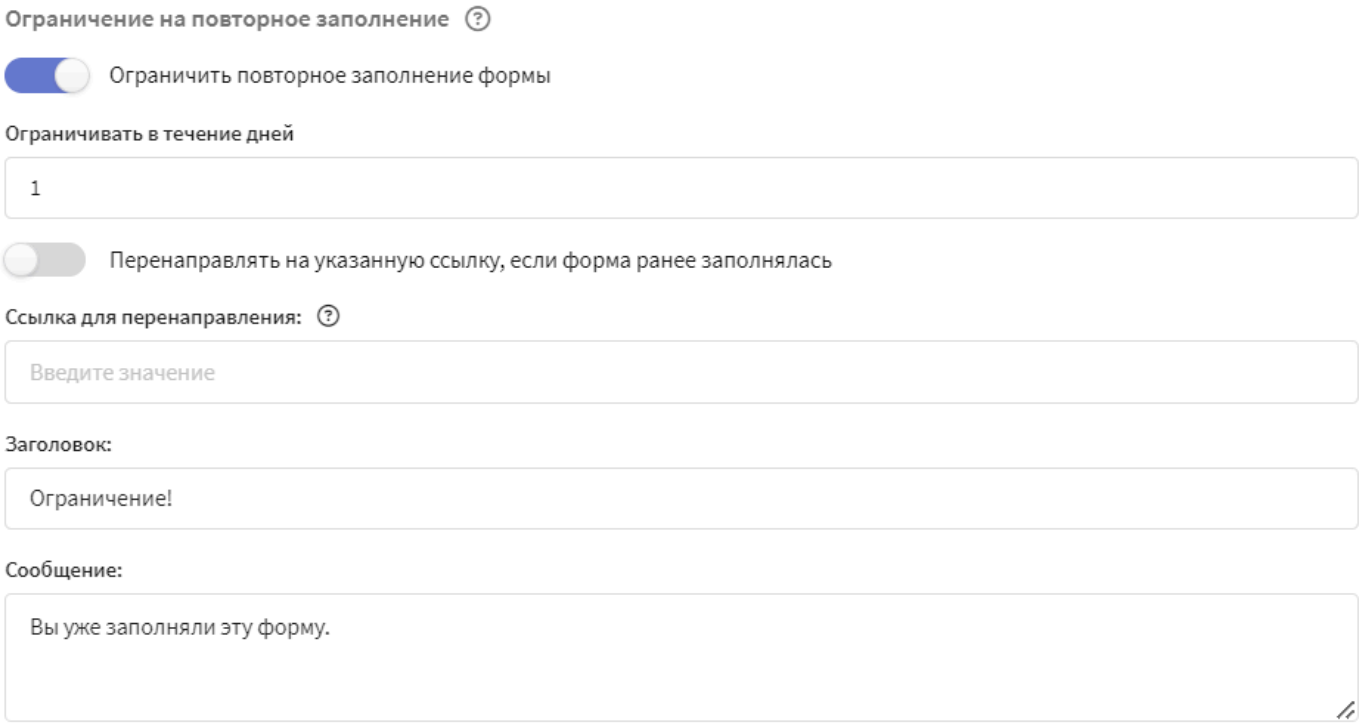
Ссылка при успешно заполненной форме — это ссылка, по которой может перейти пользователь, когда заполнит форму. Ссылка будет отображена в виде кнопки **“Перейти на сайт”**. Название кнопки может быть изменено на вкладке **“Оформление”**.



Если нажать на переключатель **“Перейти на указанную ссылку, минуя финальное сообщение”**, то пользователь сразу после отправки формы перейдет по указанной ссылке.

Ограничение на повторное заполнение

Если вы хотите ограничить повторное заполнение формы одним и тем же пользователем, активируйте соответствующий переключатель и задайте количество дней, в течение которых заполнение формы должно быть ограничено:



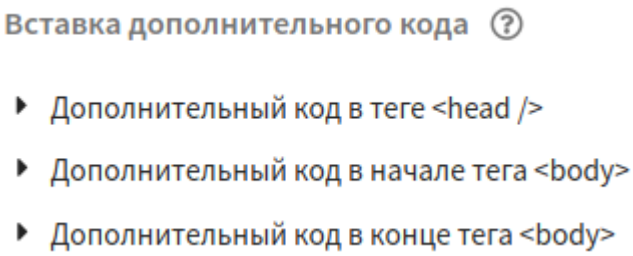
Данные о заполнении формы записываются в файлы cookie. При повторном открытии страницы с формой сервер считает данные из cookie и вместо формы отобразит сообщение: “Вы уже заполнили эту форму, повторное заполнение невозможно”. Текст этого сообщения можно изменить, заполнив поля **“Заголовок”** и **“Сообщение”**.

Вы также можете предоставить пользователю возможность перейти на определенный сайт при попытке повторного заполнения. Для этого укажите ссылку для перенаправления. В таком случае на странице с предупреждением появится кнопка “Перейти на сайт”. Название кнопки можно изменить на вкладке **“Оформление”**.

Если вы хотите, чтобы пользователь при повторной попытке открыть уже заполненную форму сразу переходил на другую страницу, нажмите на переключатель **“Перенаправлять на указанную ссылку, если форма ранее заполнялась”**.

Кастомизация формы

У вас есть возможность добавить в форму дополнительный код. Вы можете вставить код в тег `<head>` после открывающего тега `<body>` или перед закрывающим тегом `</body>`.



INFO: Для чего можно использовать дополнительный код?

- для оформления страницы с формой;

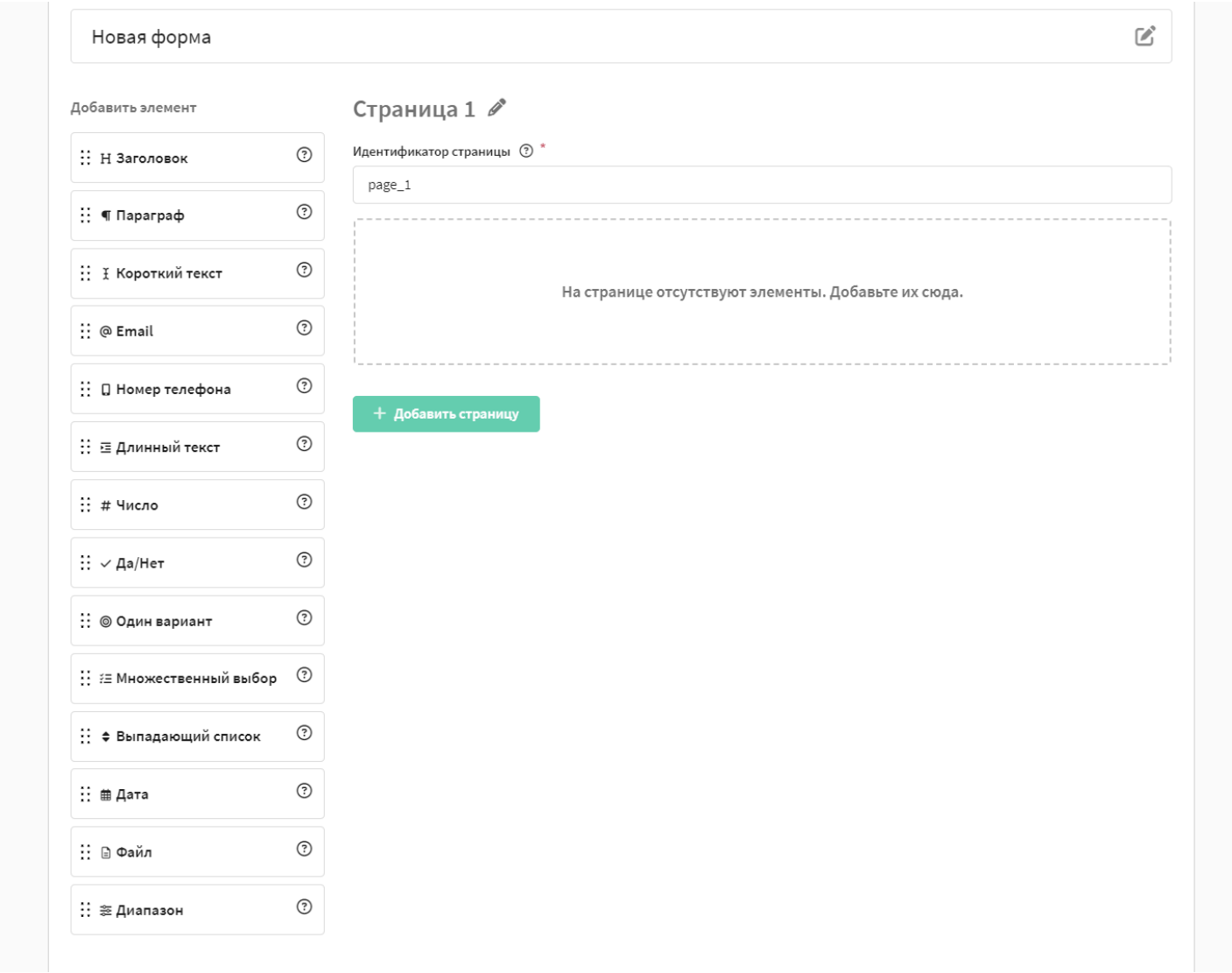
- для добавления скриптов (например, скриптов аналитики или регистрации цели).

TIP

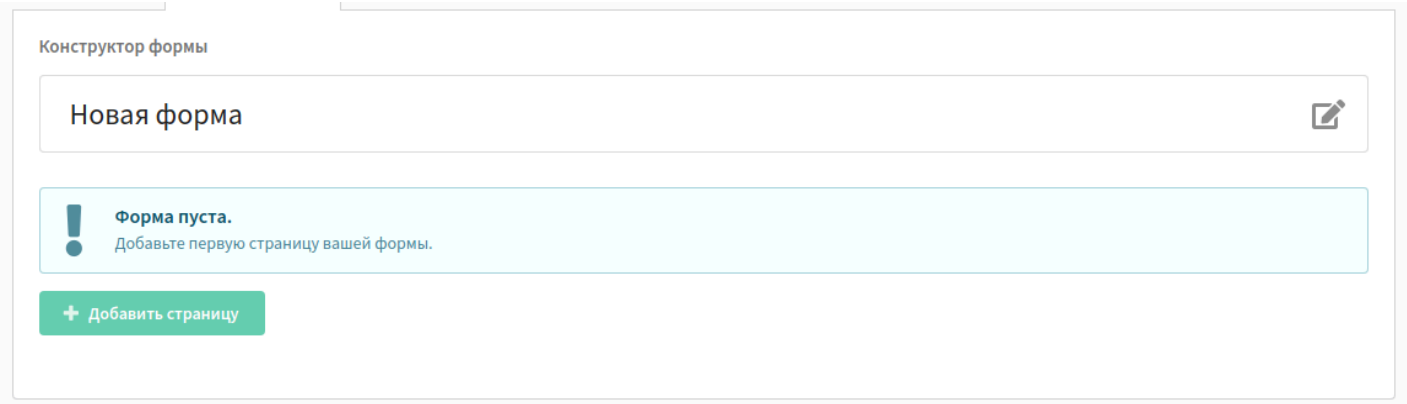
В **on-premise версии платформы** администраторы могут настраивать внешний вид и поведение форм (страница ошибок, страница неактивной формы и т.д.), изменяя шаблонные файлы. Подробнее в документации для администратора.

Конструктор формы

На второй вкладке **“Конструктор”** происходит настройка структуры формы и ее полей.



В правой части экрана отображаются страницы формы и ее элементы. Работа в конструкторе начинается с создания страницы — нажмите **“Добавить страницу”**:

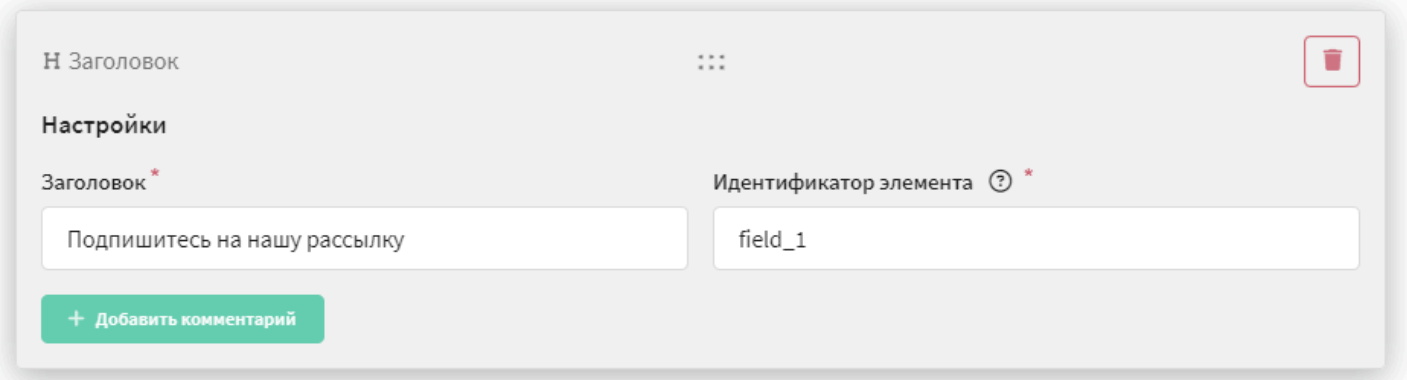


В левом столбце показаны типы элементов, которые могут быть добавлены в форму.

Элементы и их описание

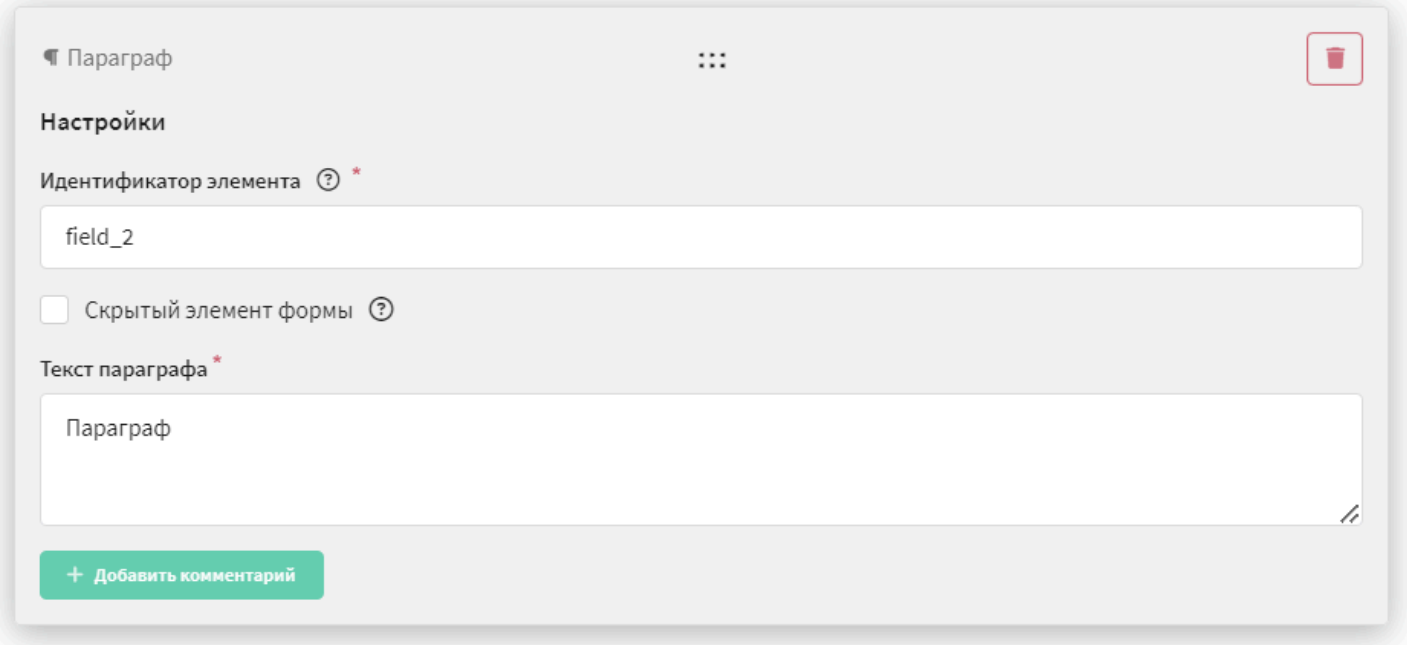
Заголовок

Заголовок — текстовый элемент, где вы можете указать основную тему своей формы.



Параграф

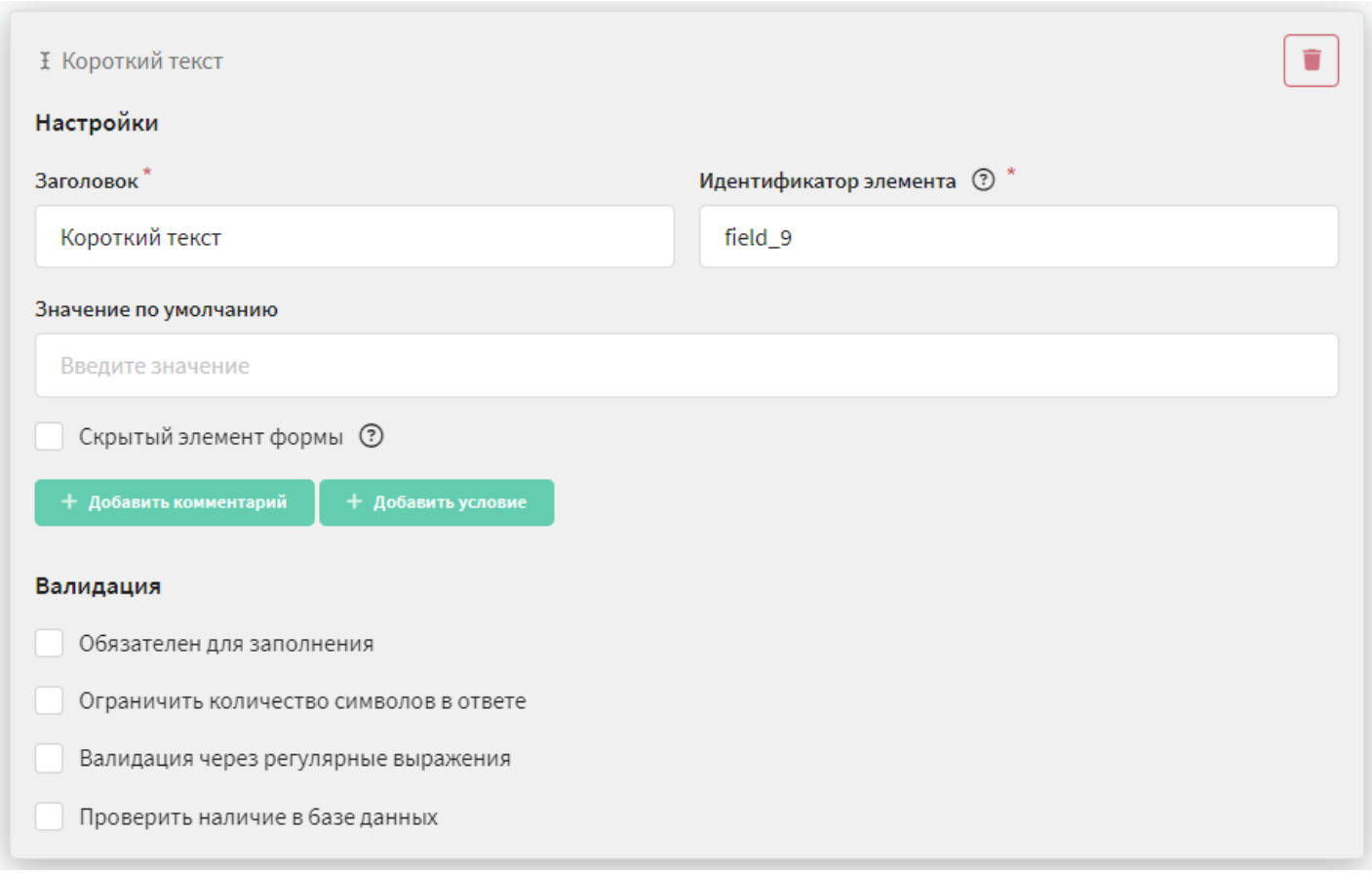
Параграф — текстовый элемент. В нем может быть размещена какая-либо информация, которая должна отображаться в форме.



Короткий текст

Короткий текст — текстовое поле, в котором может быть задан вопрос с коротким ответом. Например, имя или фамилия клиента.

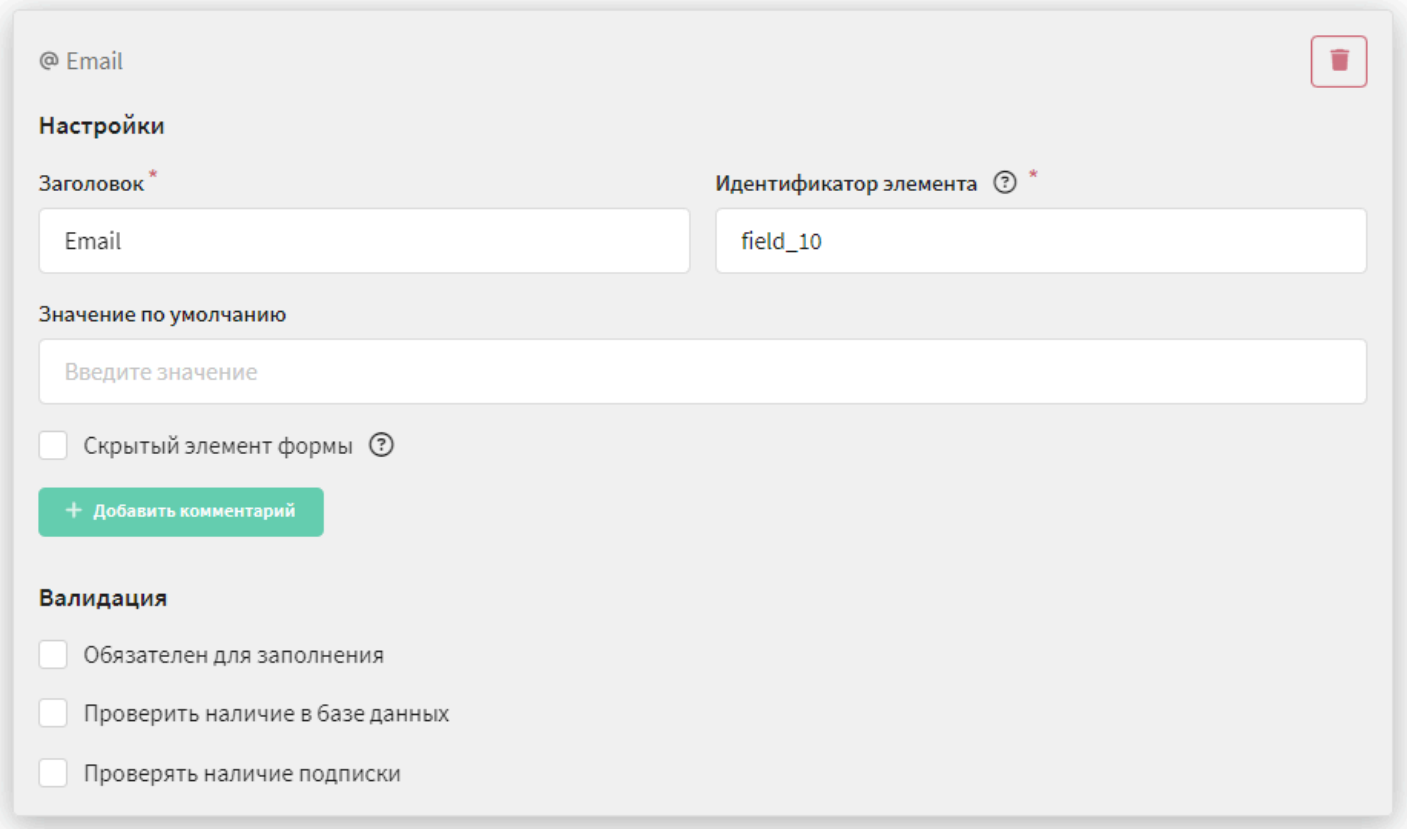
Для данного поля может быть задан заголовок, значение по умолчанию, а также можно настроить видимость элемента и валидацию данных.



Email

Email — поле, в котором необходимо указать email.

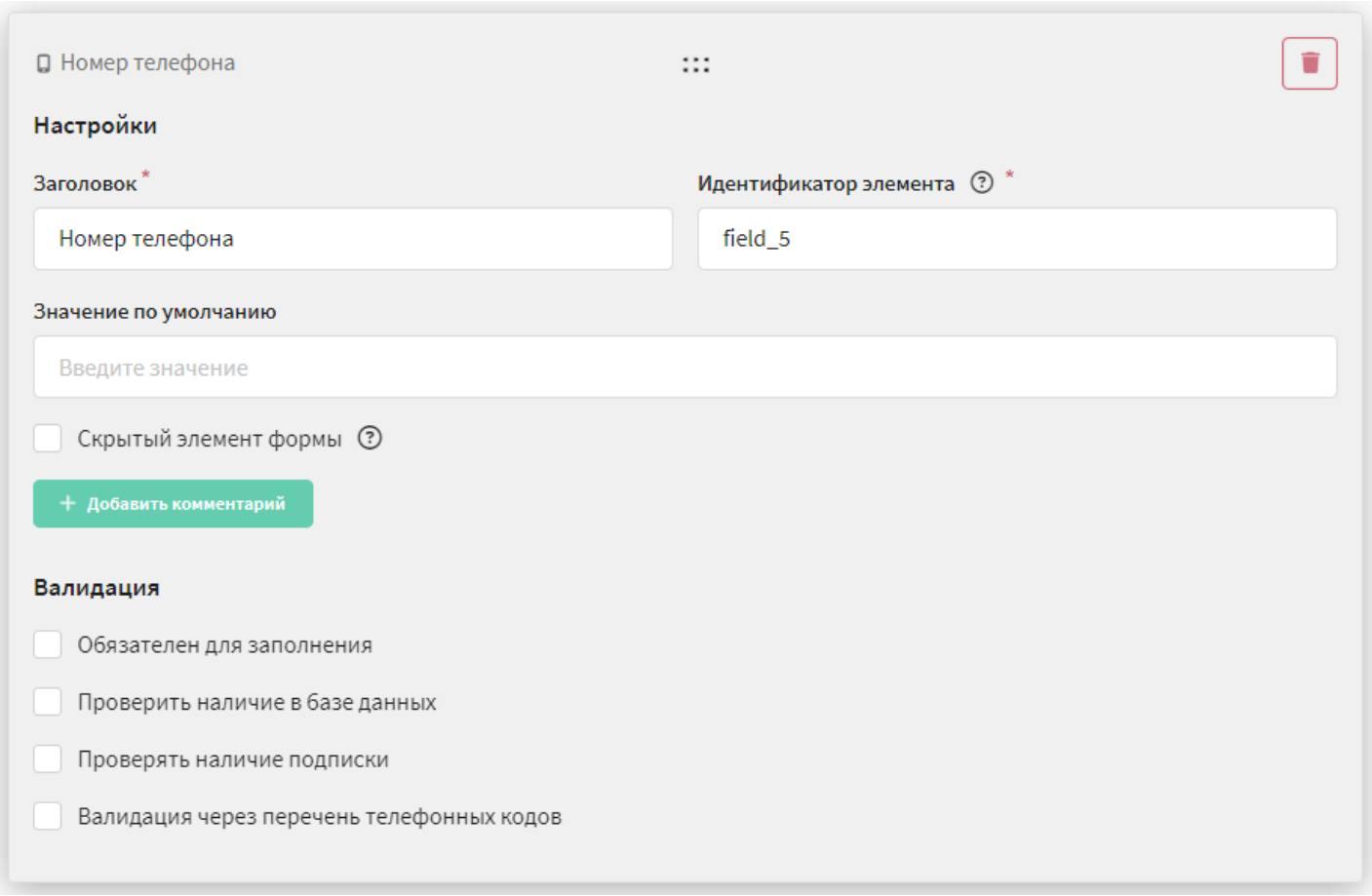
Для данного поля может быть задан заголовок, значение по умолчанию, а также можно настроить видимость элемента и валидацию данных.



Номер телефона

Номер телефона — поле, в котором можно указать номер телефона.

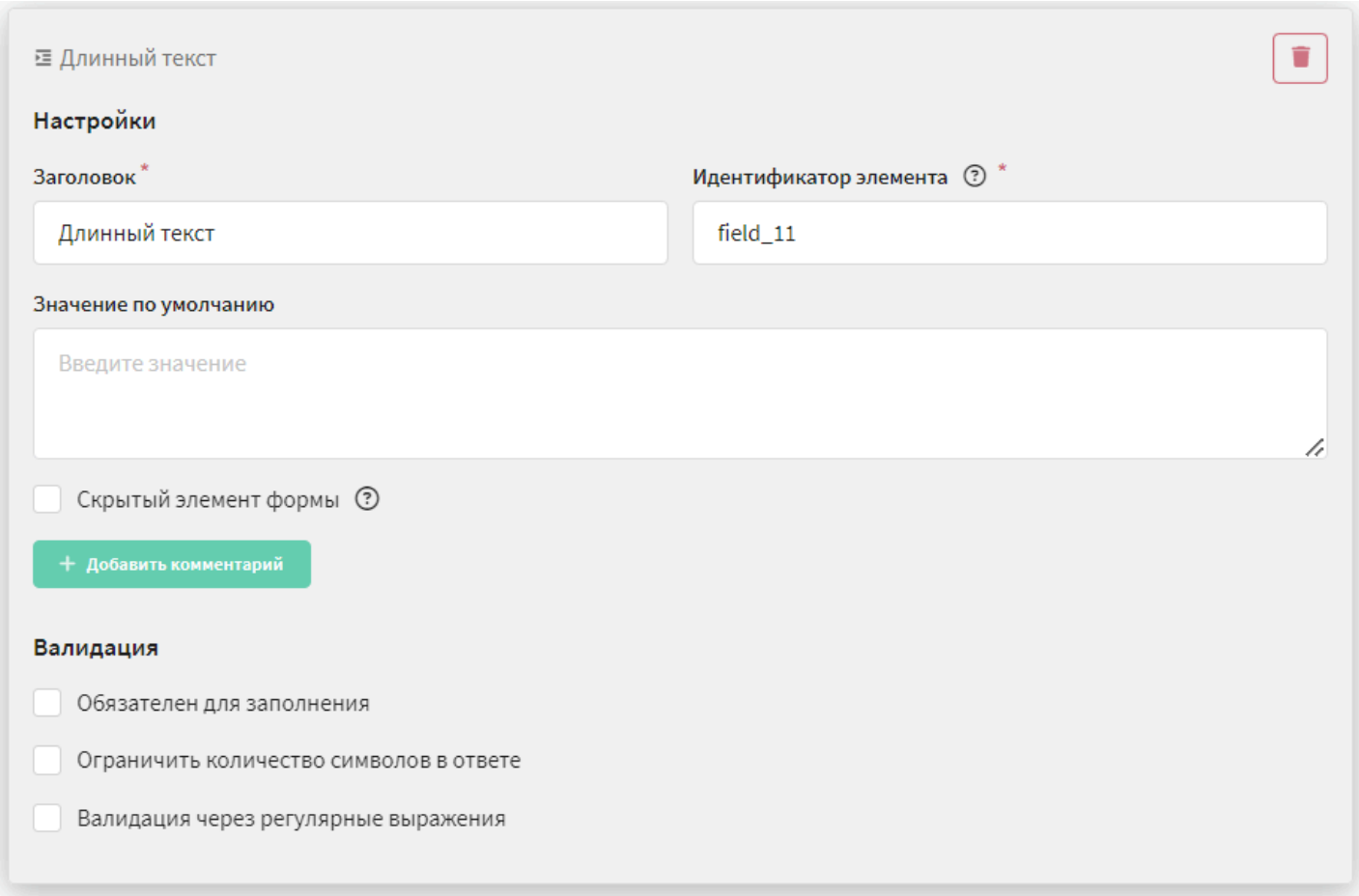
Для данного поля может быть задан заголовок, значение по умолчанию, а также можно настроить видимость элемента и валидацию.



Длинный текст

Длинный текст — текстовое поле, в котором может быть задан вопрос с длинным ответом, например, отзыв о продукте или мероприятии.

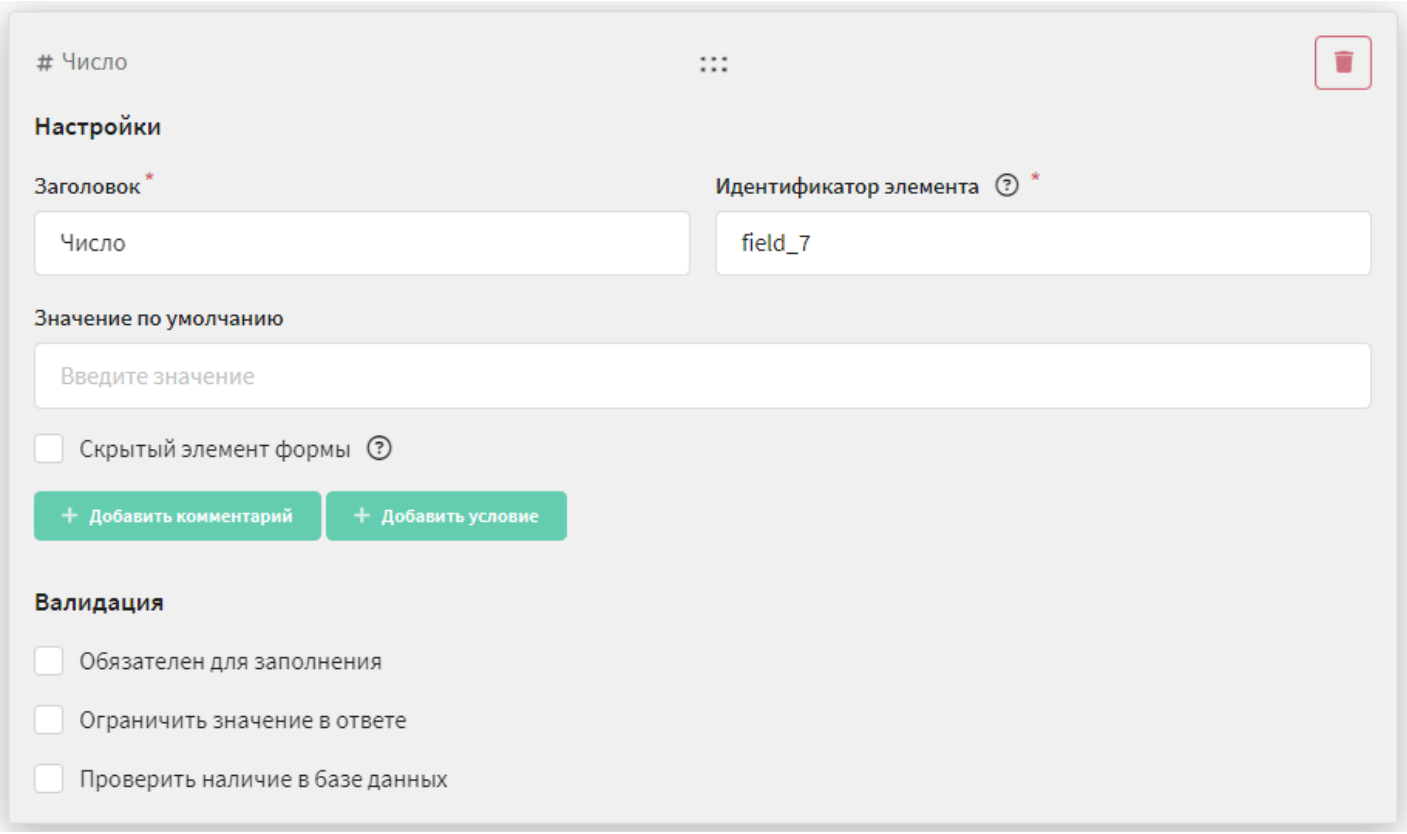
Для данного поля может быть задан заголовок, значение по умолчанию, а также можно настроить видимость элемента и валидацию данных.



Число

Число — поле, в котором может быть задан вопрос с ответом в виде целого или дробного числа, например, возраст.

Для данного поля может быть задан заголовок, значение по умолчанию, а также можно настроить видимость элемента и валидацию данных.

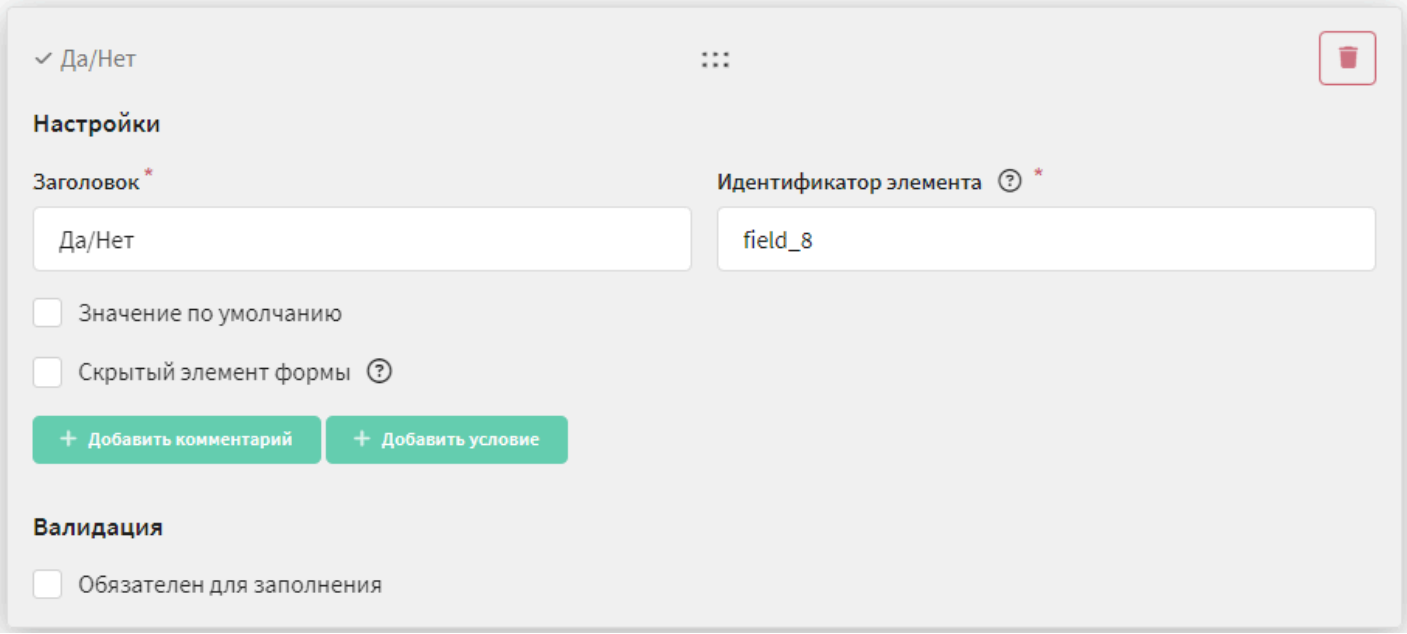


Да/Нет

Да/Нет — поле, в котором может быть задан вопрос с целью получения согласия, например, на получение почтовой рассылки.

Для данного поля может быть задан заголовок, значение по умолчанию, а также можно настроить видимость элемента.

Если вы поставите галочку в поле **“Значение по умолчанию”**, то в форме по умолчанию будет ответ “да”.



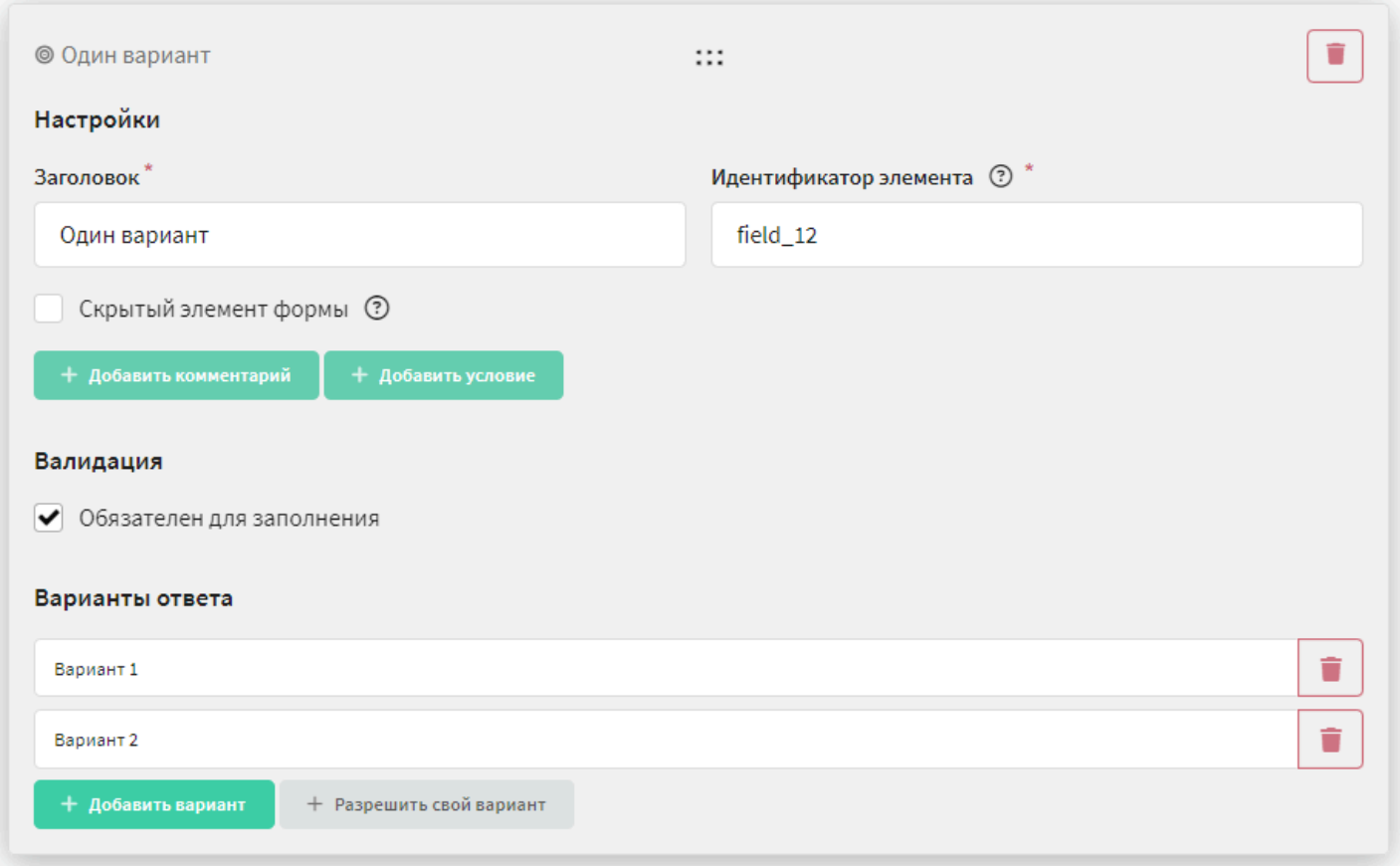
Один вариант

Один вариант — поле, в котором может быть предложено выбрать один из нескольких вариантов ответа. Например, цвет товара — красный или синий.

Для данного поля может быть задан заголовок, варианты ответов, а также можно настроить видимость элемента и валидацию данных.

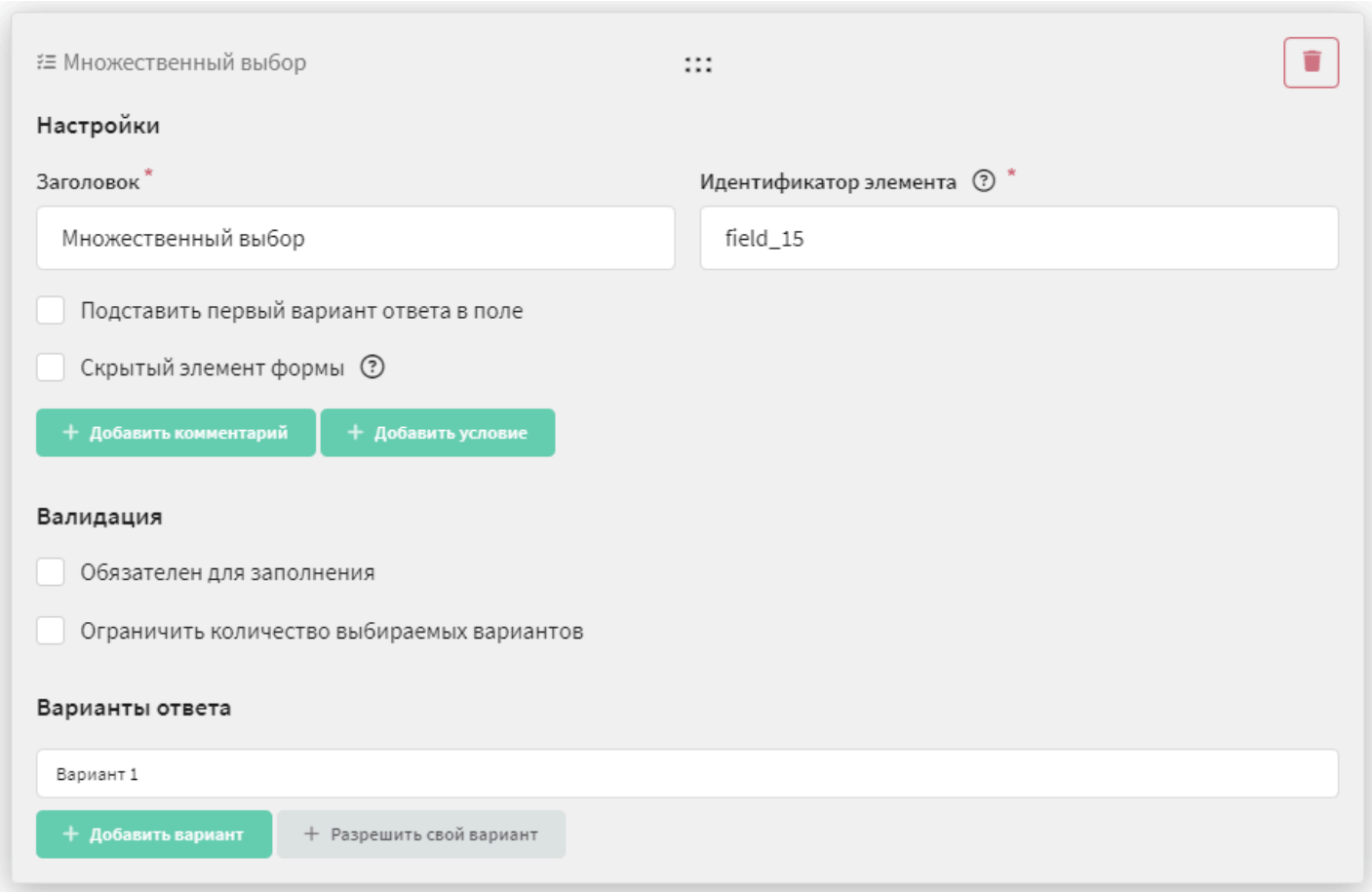
Нажмите на кнопку **“Добавить вариант”** и введите значение. Количество вариантов не ограничено.

Вы можете разрешить пользователю ввести свой вариант ответа, нажав на кнопку **“Разрешить свой вариант”**.

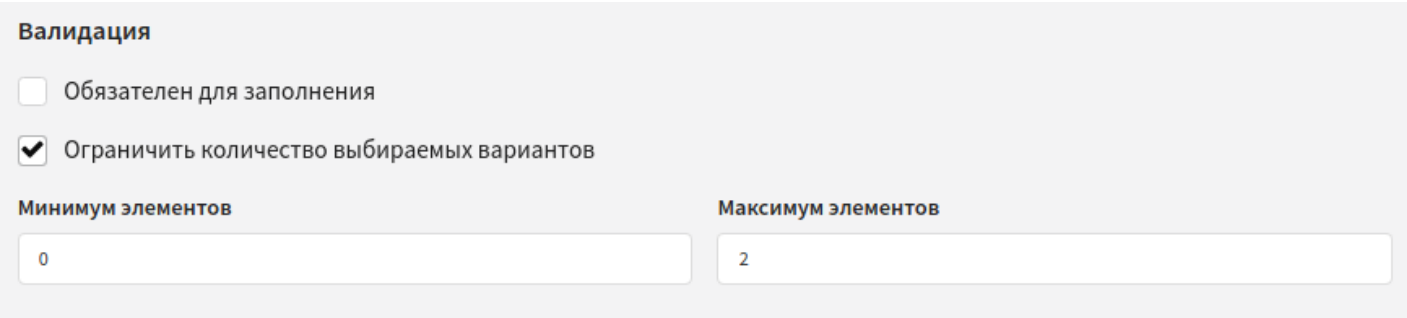


Множественный выбор

Элемент позволяет выбрать несколько вариантов ответа из предложенных. С помощью кнопки **“Добавить вариант”** добавьте возможные ответы.



Элемент можно сделать обязательным, ограничить количество выбираемых вариантов, а также можно подставить первый вариант ответа в поле:



Выпадающий список

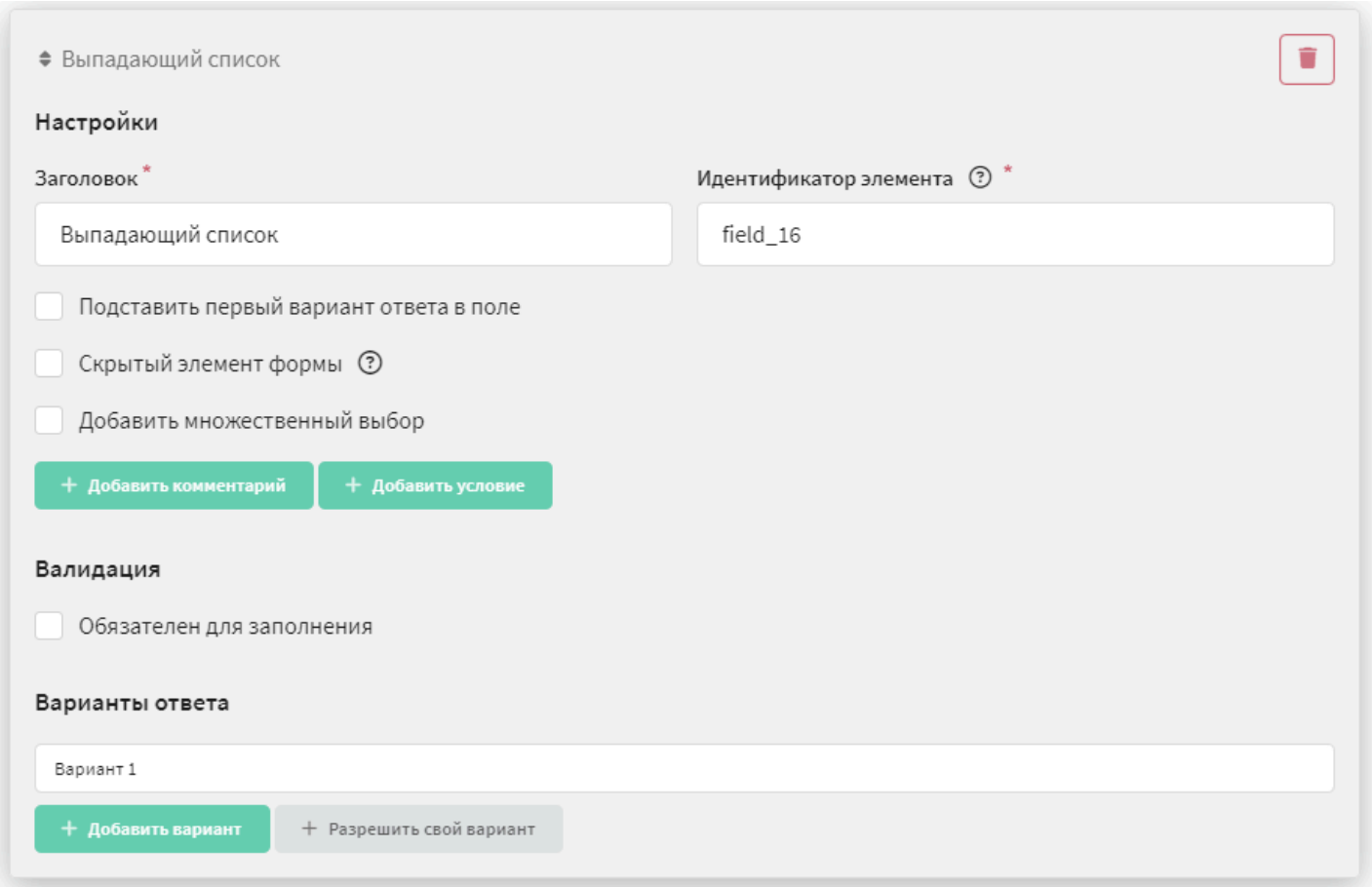
Выпадающий список — поле, в котором пользователю надо выбрать один вариант из выпадающего списка.

Для данного поля необходимо задать заголовок и варианты ответов. Нажмите на кнопку **“Добавить вариант”** и введите значение. Количество вариантов не ограничено.

Вы можете разрешить пользователю ввести свой вариант ответа, нажав на кнопку **“Разрешить свой вариант”**. При выборе этого варианта откроется дополнительное поле для ввода текста.

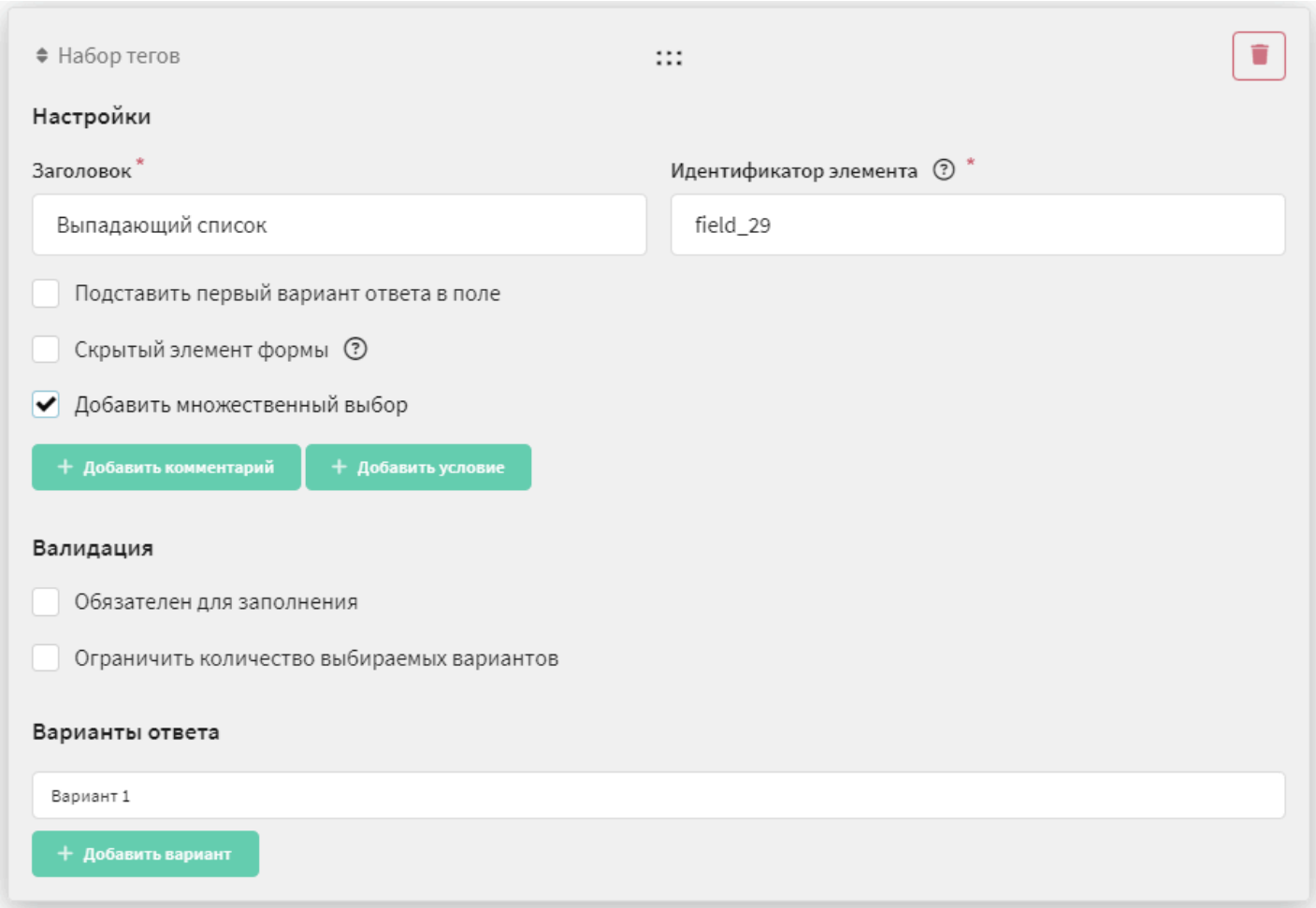
Кроме того, есть возможность **подставить первый вариант ответа в поле**, нажав на соответствующую кнопку.

Также можно настроить видимость элемента и валидацию данных.



Если вы хотите предоставить пользователю возможность выбрать несколько вариантов ответа, кликните **“Добавить множественный выбор”**.

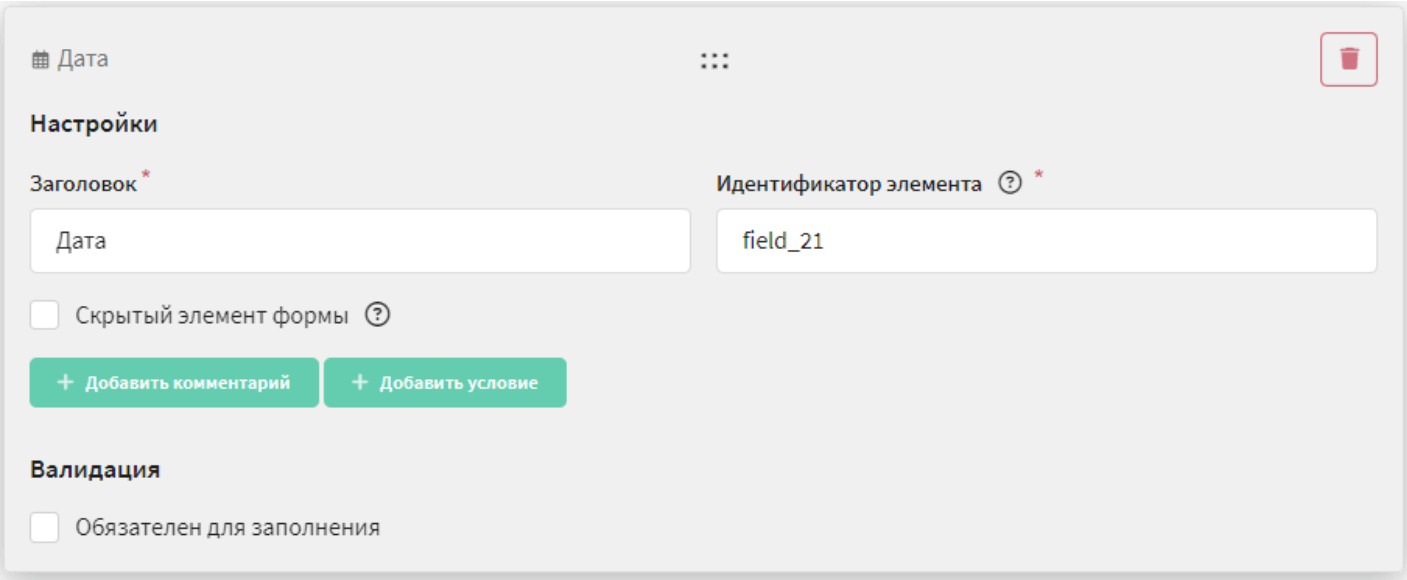
Обратите внимания, что при множественном выборе появляется новый тип валидации — **“Ограничить количество выбираемых вариантов”**.



Дата

Дата — поле, в котором предлагается указать дату, например, день рождения клиента.

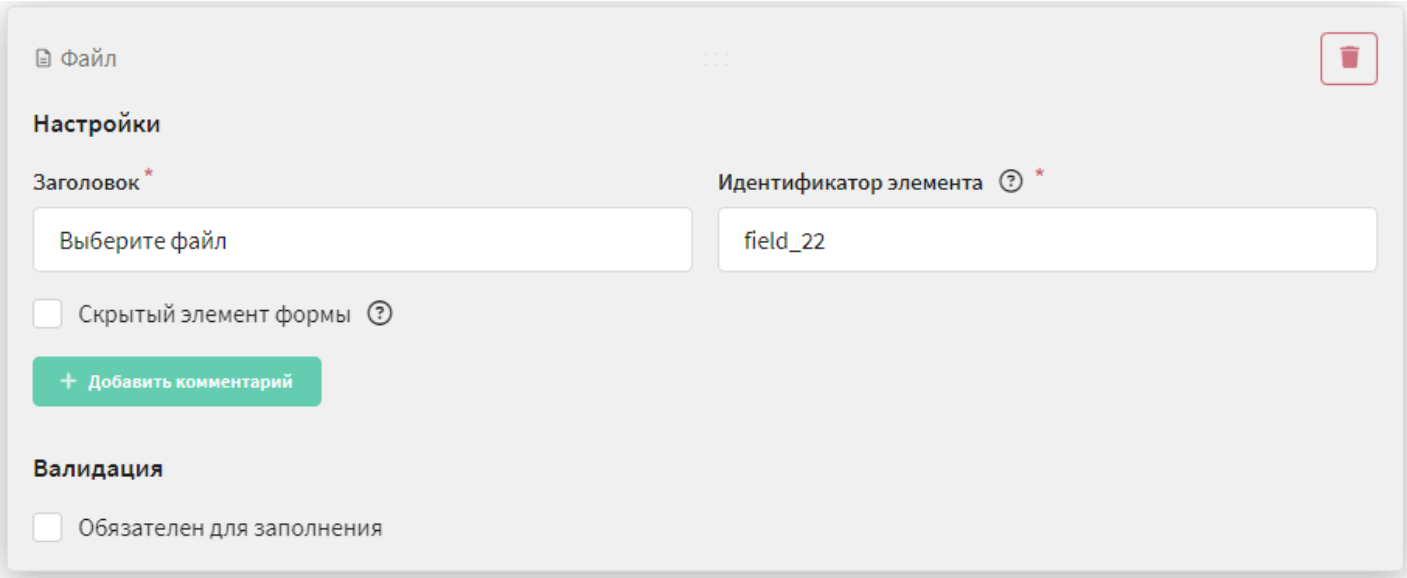
Для данного поля может быть задан заголовок, а также можно настроить видимость элемента и валидацию данных.



Файл

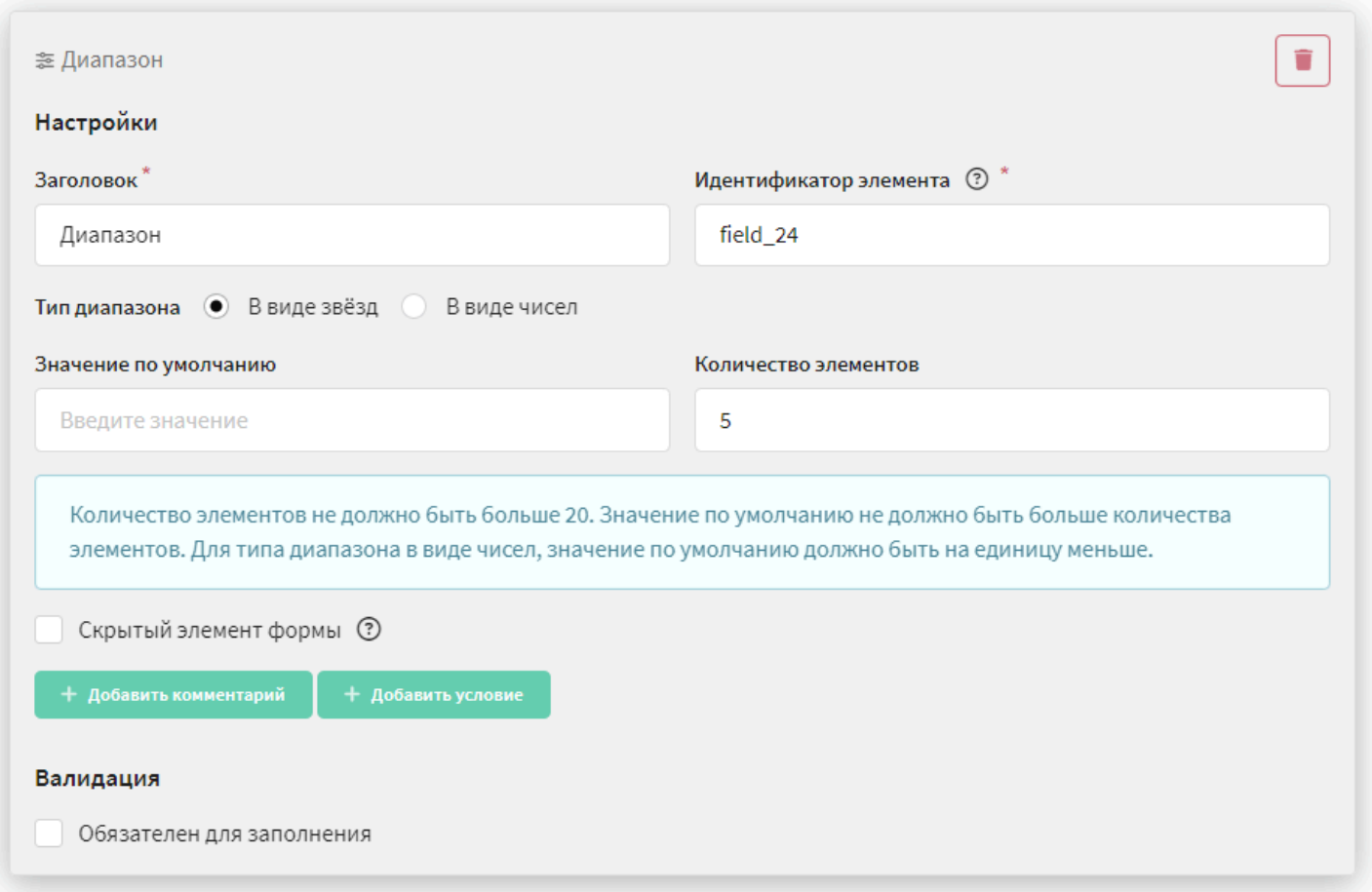
Файл — поле, в котором к форме могут быть прикреплены файлы, например, фото. Максимальный размер файла – 20 мб.

Для данного поля может быть задан заголовок, а также можно настроить видимость элемента и валидацию данных.



Диапазон

Диапазон — диапазон выбора оценки. Для данного поля может быть задан заголовок, а также можно настроить видимость элемента и валидацию данных.

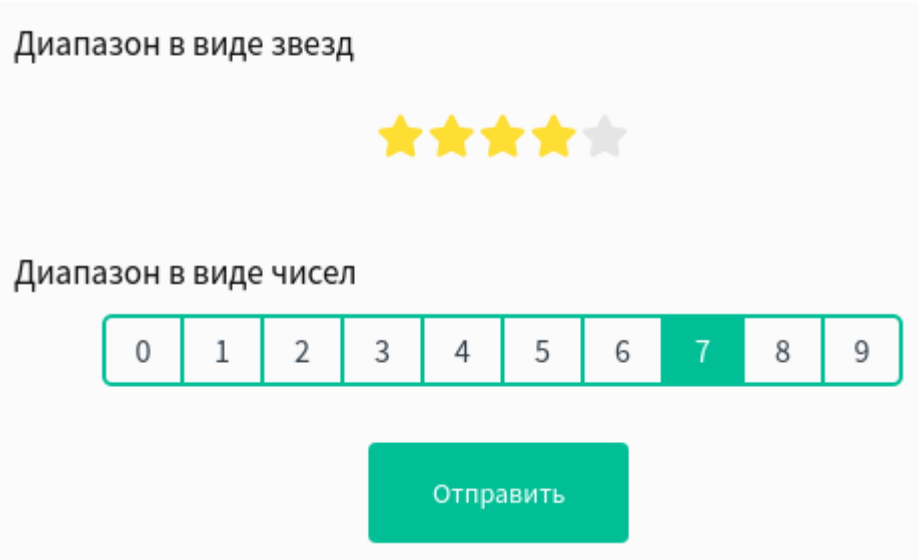


Кроме того, вы можете выбрать **тип диапазона: в виде звезд** или **в виде чисел**.

Вы также можете указать количество элементов (например, 5 звезд или 10 чисел).

TIP

Если в качестве типа диапазона вы выбираете числа, то обратите внимание, что отсчет начинается с 0.

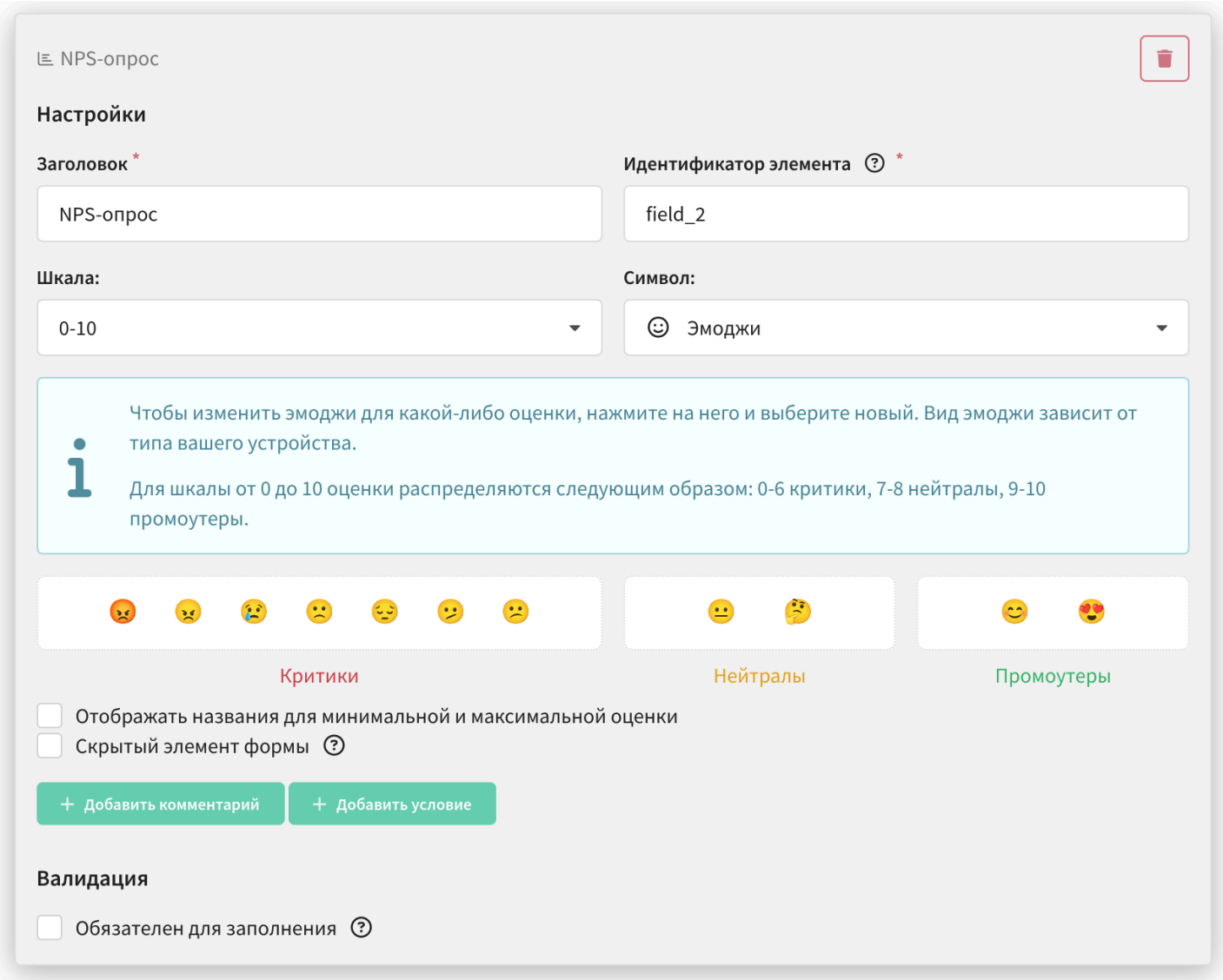


NPS-опрос

NPS-опрос — тестирование для получения индекса потребительской лояльности. Для данного поля может быть задан заголовок, настроена видимость элемента и валидация данных, а также названия для максимальной и минимальной оценки.

TIP

Узнать подробнее о том, что такое NPS-тестирование и как его создать, вы можете в этой статье.

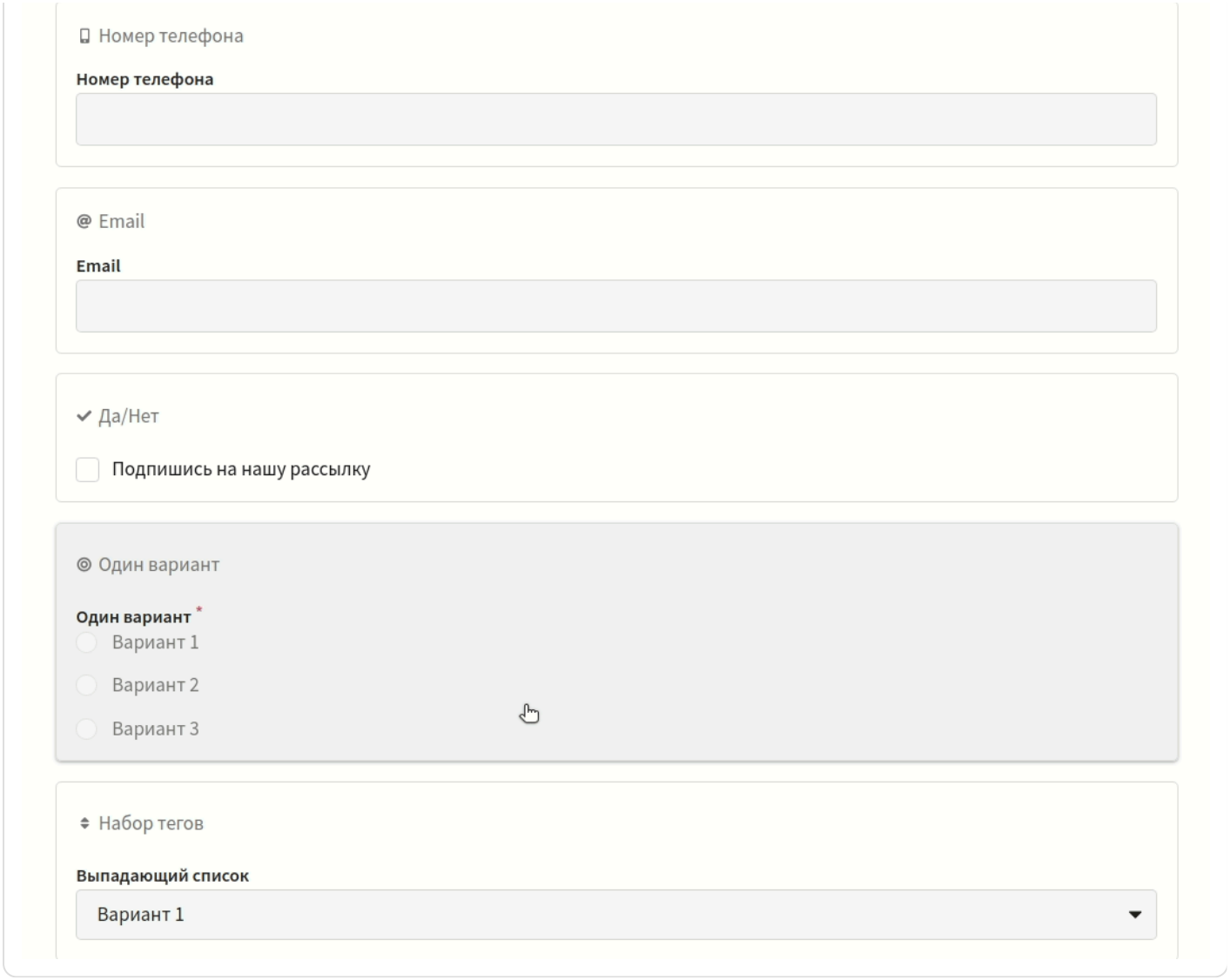


Вы можете выбрать **тип символов**: в виде **звезд**, **сердец**, **эмодзи** или в виде **чисел**.

Для шкалы также можно указать размер — **от 0 до 5**, **от 0 до 10**, **от 1 до 5** и **от 1 до 10**.

TIP

Вы можете перетаскивать элементы формы, без особого труда располагая их в том порядке, который вам нужен. При наведении на элемент в верхней части появляется скопление точек. Наведите на них курсор, зажмите левую кнопку мыши и перетащите элемент, куда пожелаете.



Валидация полей формы

У большинства полей доступен один или несколько типов валидации:

Валидация	Поле
Обязателен для заполнения	Короткий текст, email, номер телефона, длинный текст, число, один вариант, выпадающий список, дата, файл, диапазон
Ограничить количество символов в ответе	Короткий текст, длинный текст
Валидация через регулярные выражения	Короткий текст, длинный текст
Проверить наличие в базе данных	Короткий текст, email, номер телефона, число
Проверить наличие подписки	Email, номер телефона
Валидация через перечень телефонных кодов (перечислите через символ “+” телефонные коды, которые разрешено указывать в форме)	Номер телефона
Ограничить значение в ответе	Число

Валидация	Поле
Ограничить количество выбираемых элементов (укажите минимальное и максимальное количество элементов, которые может выбрать пользователь)	Выпадающий список

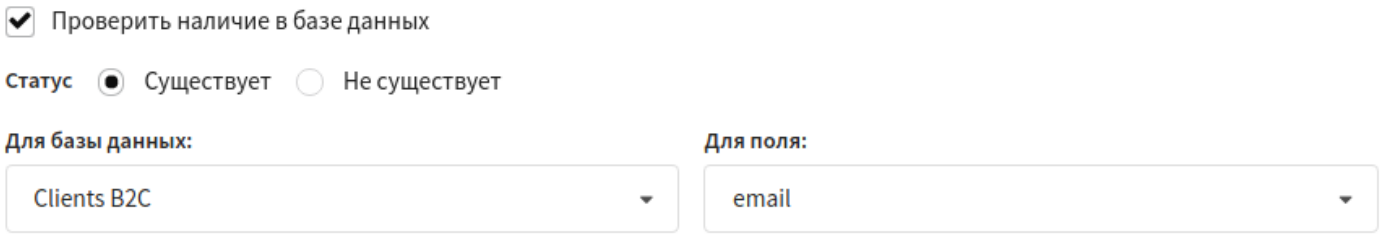
Типы валидации **“Проверить наличие в базе данных”** и **“Проверить наличие подписки”** позволяют вам ограничивать отправку формы для некоторых пользователей. Например, вы можете разрешить отправку только зарегистрированным пользователям или запретить повторную отправку формы одним и тем же пользователем.

Проверить наличие в базе данных

В момент отправки формы платформа проверит, существует ли профиль человека, заполнившего форму, в базе данных. Если профиля с указанными данными нет, то пользователь не сможет отправить форму.

При настройке формы необходимо выбрать:

- статус, который вы хотите проверять (существует или не существует),
- базу данных, в которой следует искать профиль,
- поле в базе данных для поиска существующего профиля.

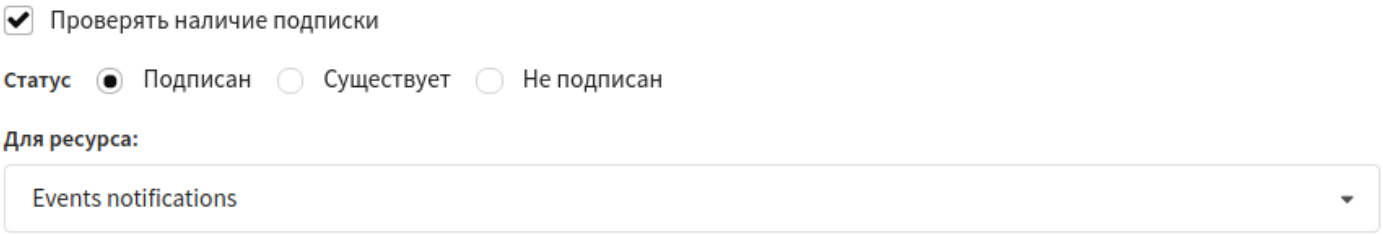


Проверить наличие подписки

В момент отправки формы платформа проверит, подписан ли профиль, заполнивший форму, на определенный ресурс. Если профиля с указанной подпиской нет, то пользователь не сможет отправить форму.

При настройке формы необходимо выбрать:

- статус, который вы хотите проверять (подписан, существует, не подписан),
- ресурс, подписку на который следует искать.

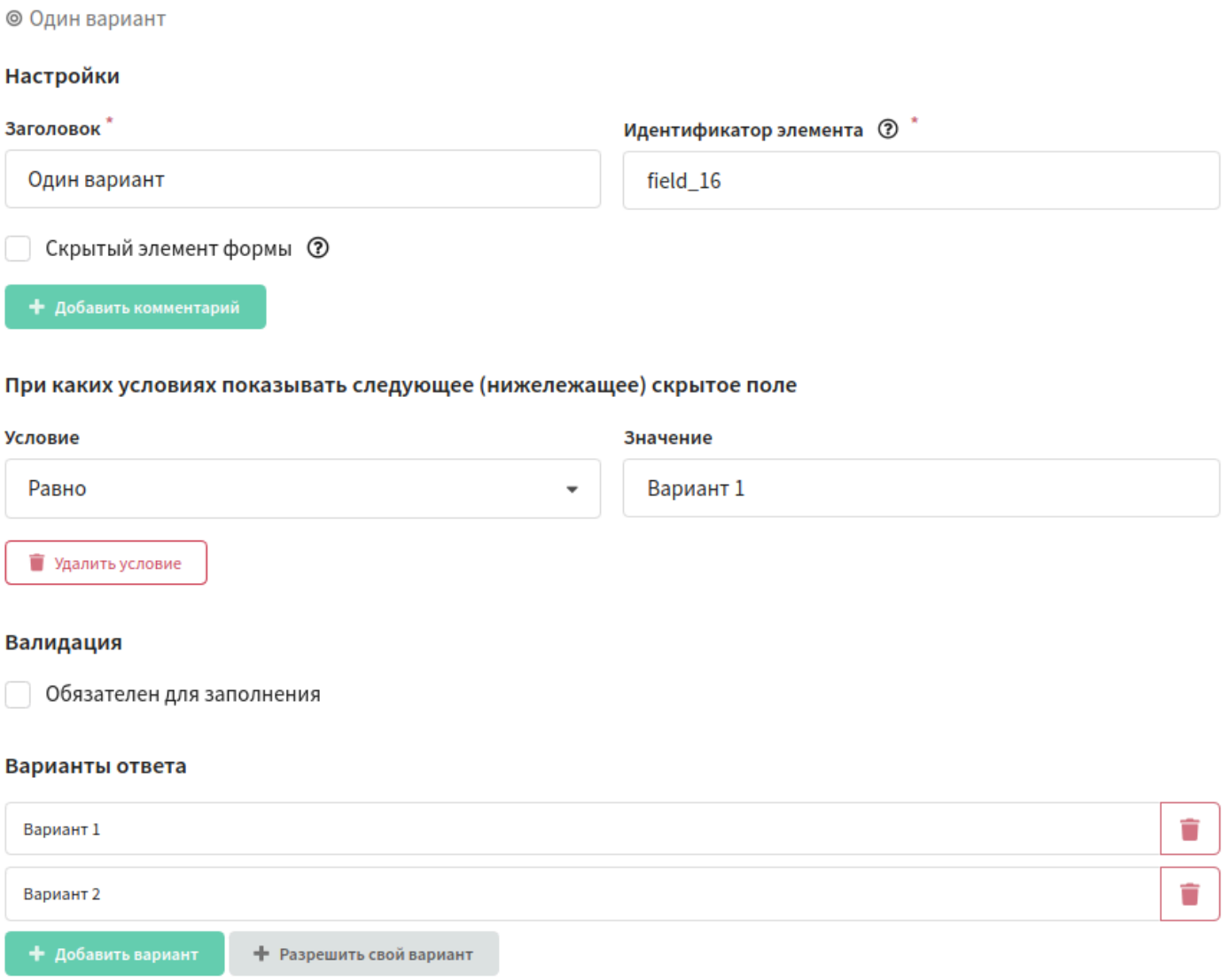


Условие показа элемента формы

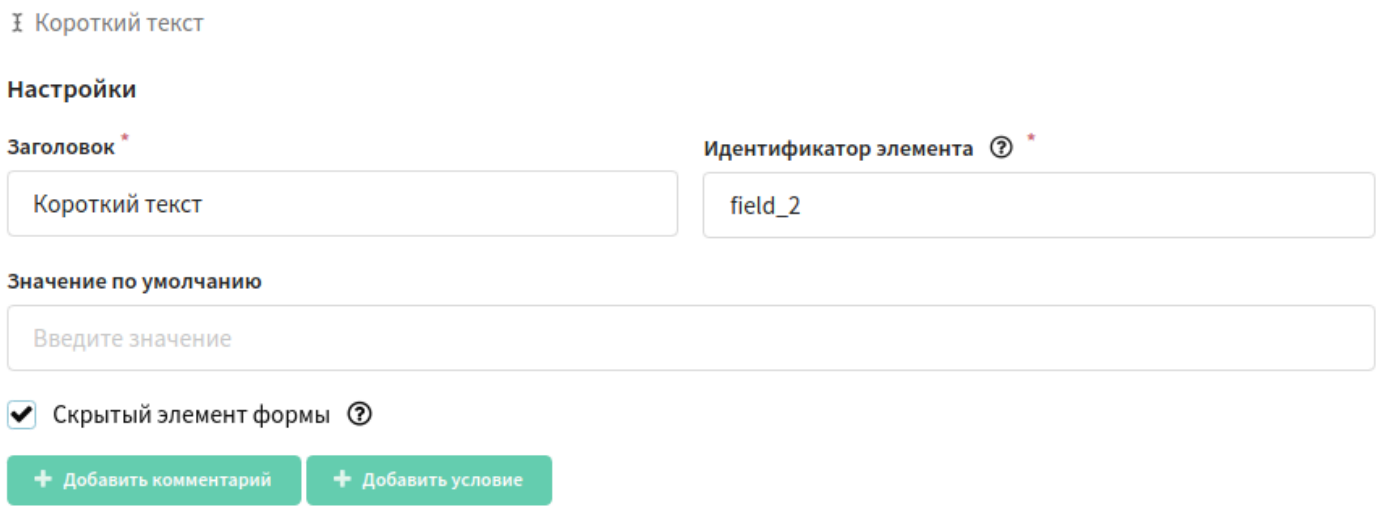
Вы можете показывать пользователю какое-то поле формы, только если он дал определенный на предыдущий вопрос. Для этого при настройке элемента нажмите на кнопку **“Добавить условие”** — откроются

дополнительные поля (**При каких условиях показывать следующее скрытое поле**).

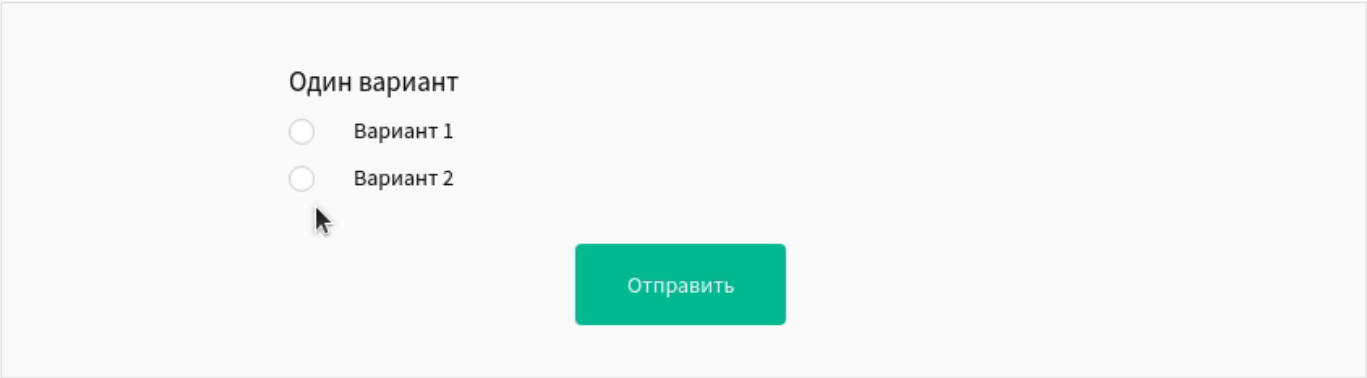
Укажите условие и значение, которое должен выбрать пользователь.



Далее добавьте элемент, который хотите показать, если пользователь выберет определенное значение в вопросе выше. Активируйте опцию **“Скрытый элемент формы”**:



Теперь если пользователь выбрал определенный вариант ответа, для него будет показан скрытый элемент. В противном случае, он перейдет сразу к следующему вопросу.

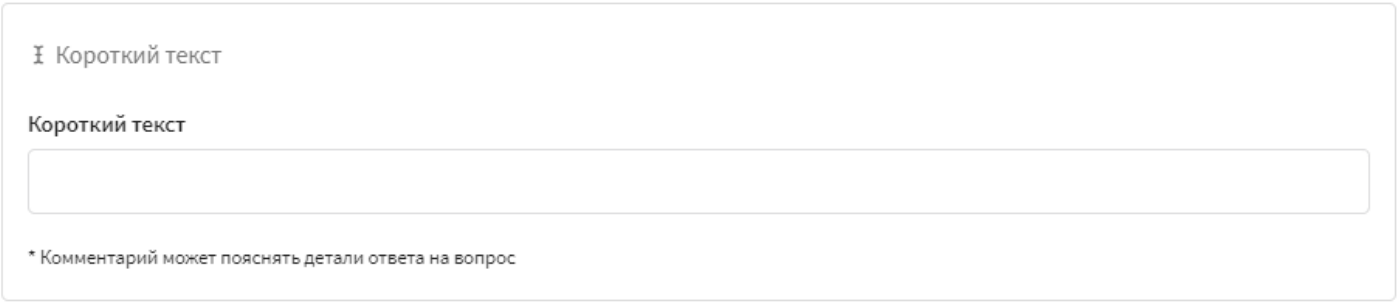


TIP

Для добавления более сложных условий, например, переход между страницами формы в зависимости от ответов, настройте блок **“Логика”**.

Комментарий

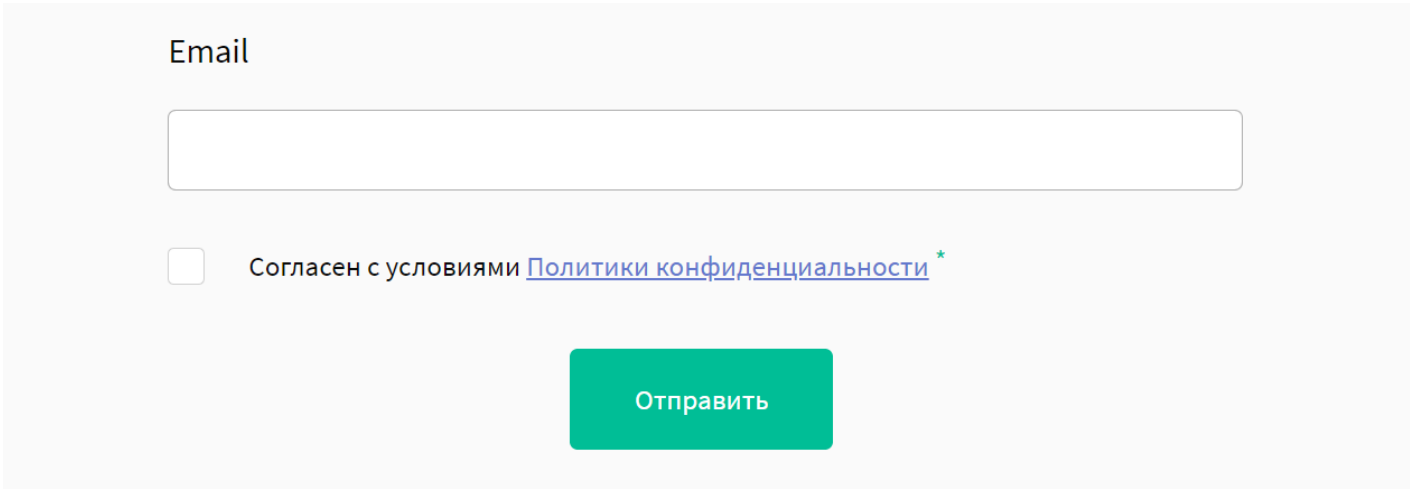
Для каждого элемента можно **добавить комментарий**, который будет отображаться под полем формы.



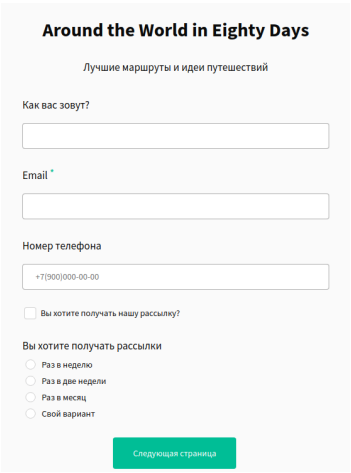
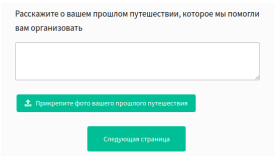
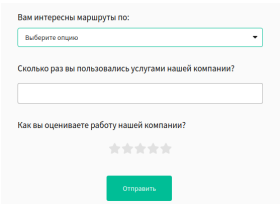
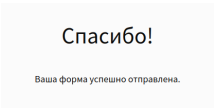
Ссылки в полях формы

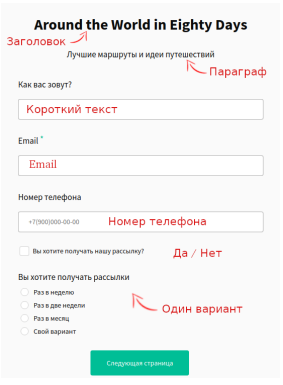
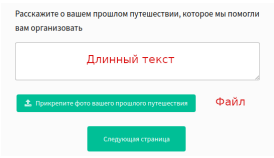
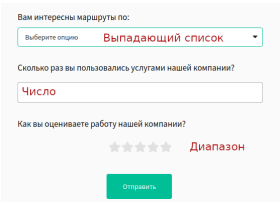
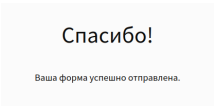
В любой элемент формы можно добавить ссылку с помощью HTML:

```
Согласен с условиями <a target="_blank" href="#">Политики конфиденциальности</a>
```



Пример формы

Страница 1	Страница 2	Страница 3	Финальная страница
			

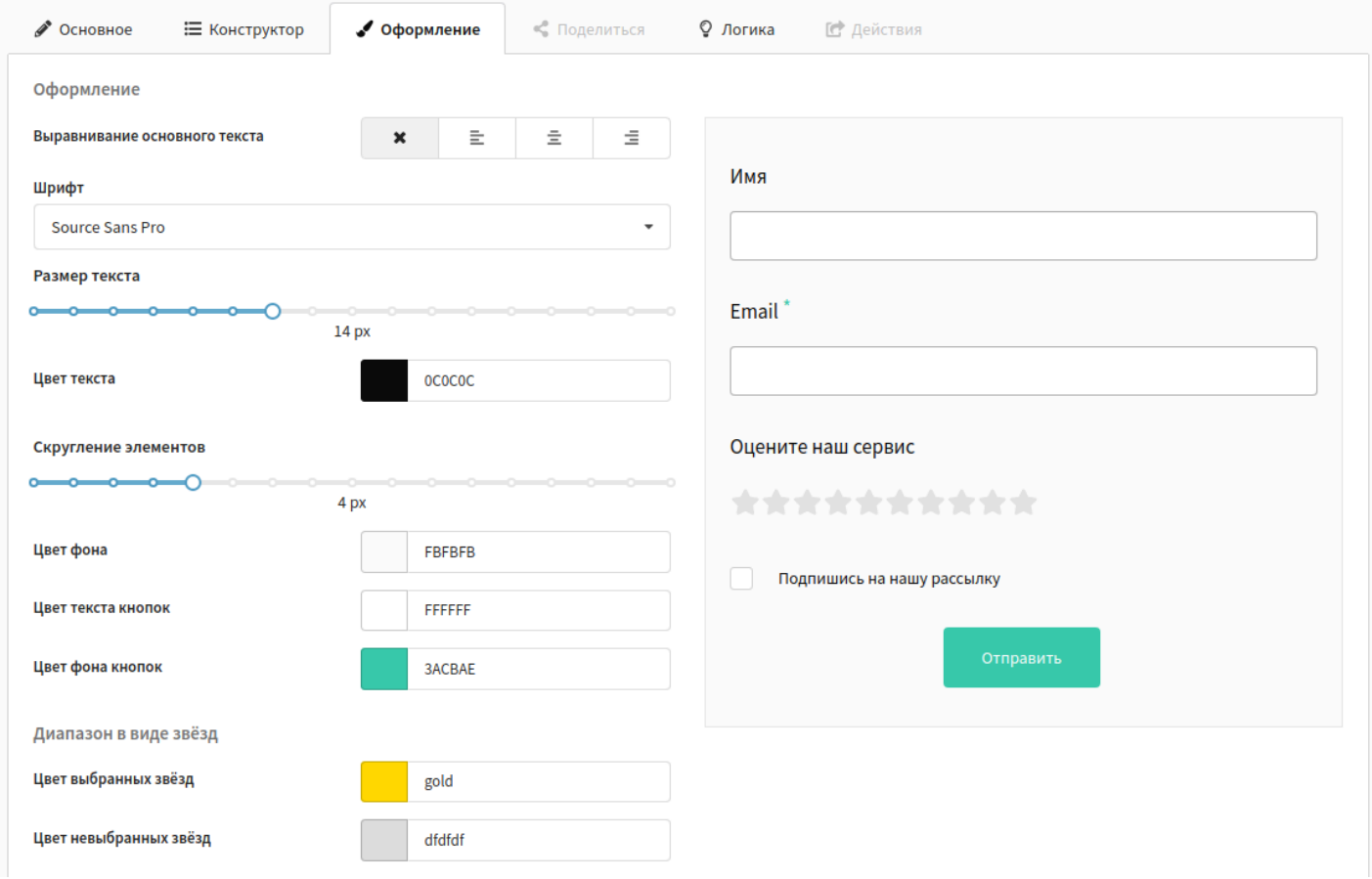
Страница 1	Страница 2	Страница 3	Финальная страница
			

Оформление формы

На данной вкладке настраивается оформление формы:

- шрифты (семейство, размер)
- цветовая схема (текст, фон, кнопки)
- выравнивание (отсутствие выравнивания, по левому/правому краю, по центру)

В левой части отображается предпросмотр формы в реальном времени.

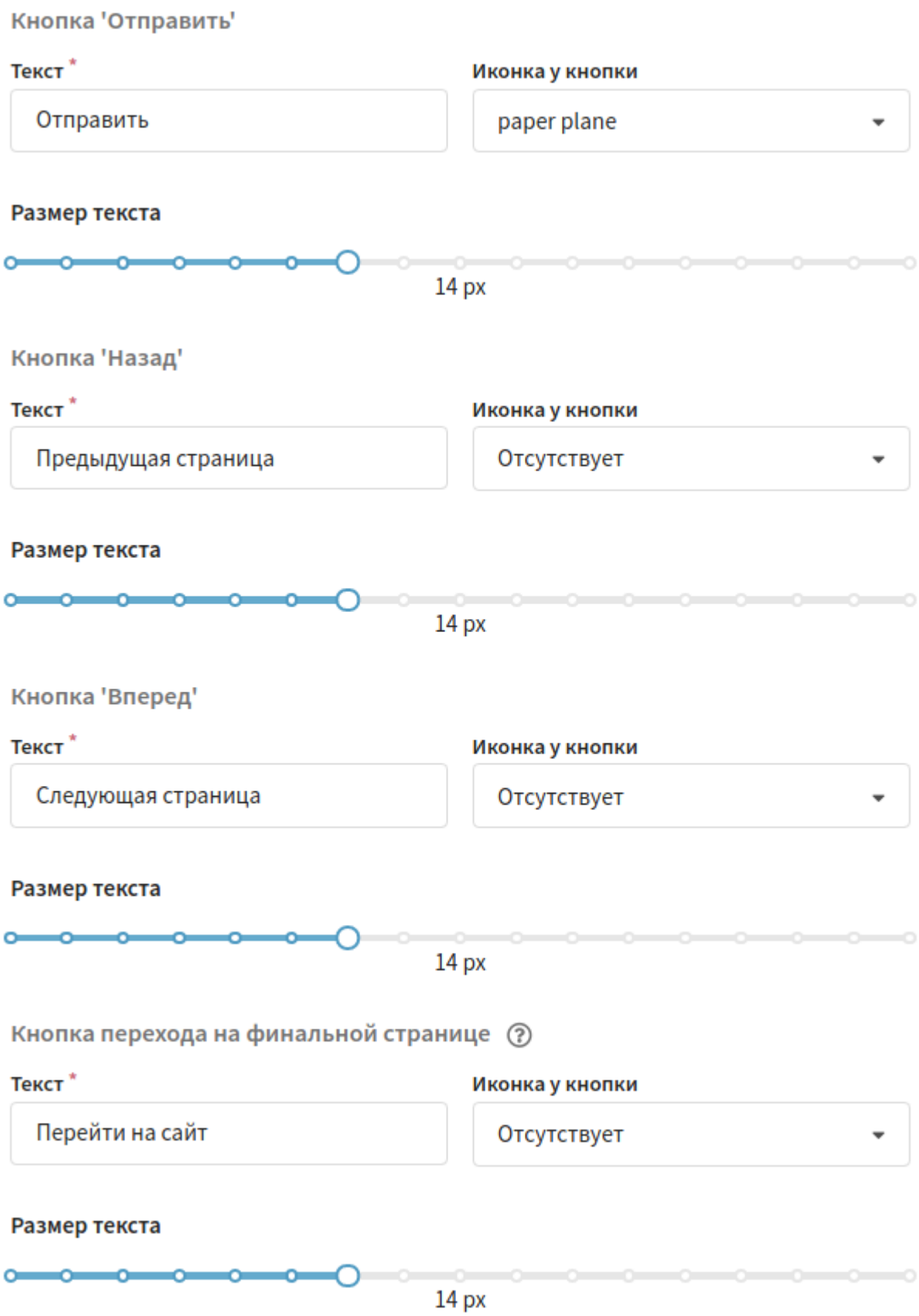


В форме доступна стилизация кнопок:

- "Отправить" (для отправления формы)
- "Назад" (для перехода на следующую страницу формы)
- "Вперед" (для перехода на предыдущую страницу формы)
- Кнопка перехода на сайт на финальной странице

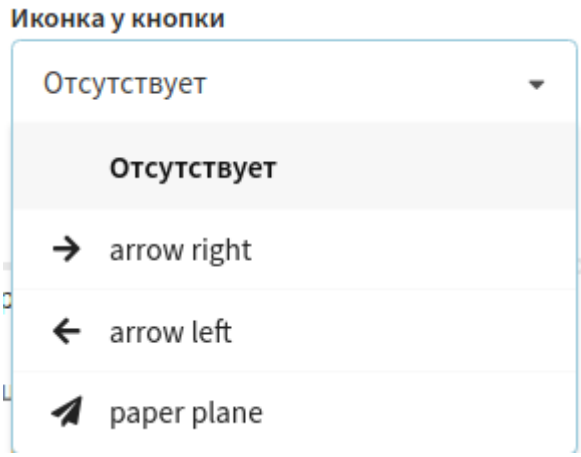
TIP: Кнопка перехода на финальной странице

Если вы оставите поле с текстом кнопки пустым или на вкладке **“Основное”** в настройках финальной страницы не укажете ссылку для перехода после заполнения формы, то кнопка не будет отображаться.



Для каждой кнопки вы можете указать:

- текст, который будет отображаться в кнопке
- размер шрифта (в пикселях)
- иконку



Когда вы зададите основные настройки формы, структуру полей и оформление, форму необходимо **сохранить.** Только после этого станут доступны настройки **действия** при активации формы и возможность ее **публикации**.

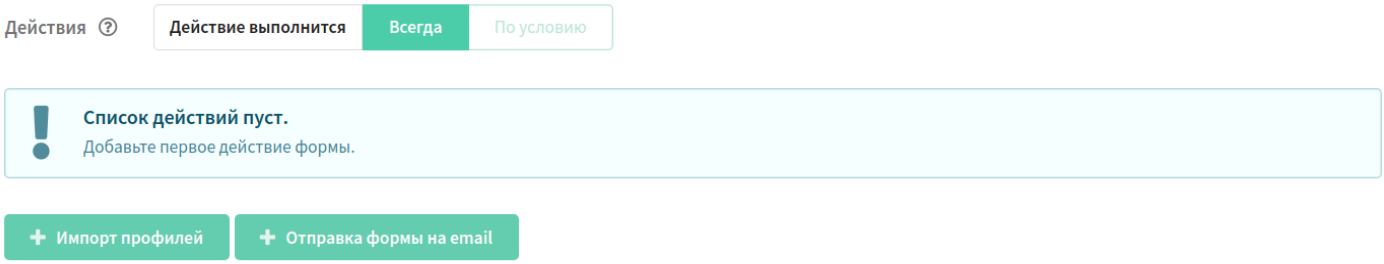
Действия при активации формы

TIP

Прежде чем приступить к работе с действиями, сохраните форму.

На вкладке **“Действия”** указываются действия, которые будут выполняться при успешном заполнении формы:

- **Импорт профилей** позволяет загрузить данные пользователя, заполнившего форму, в базу;
- **Отправка формы на email** включает отправку оповещений о заполнении формы на указанные email-адреса.



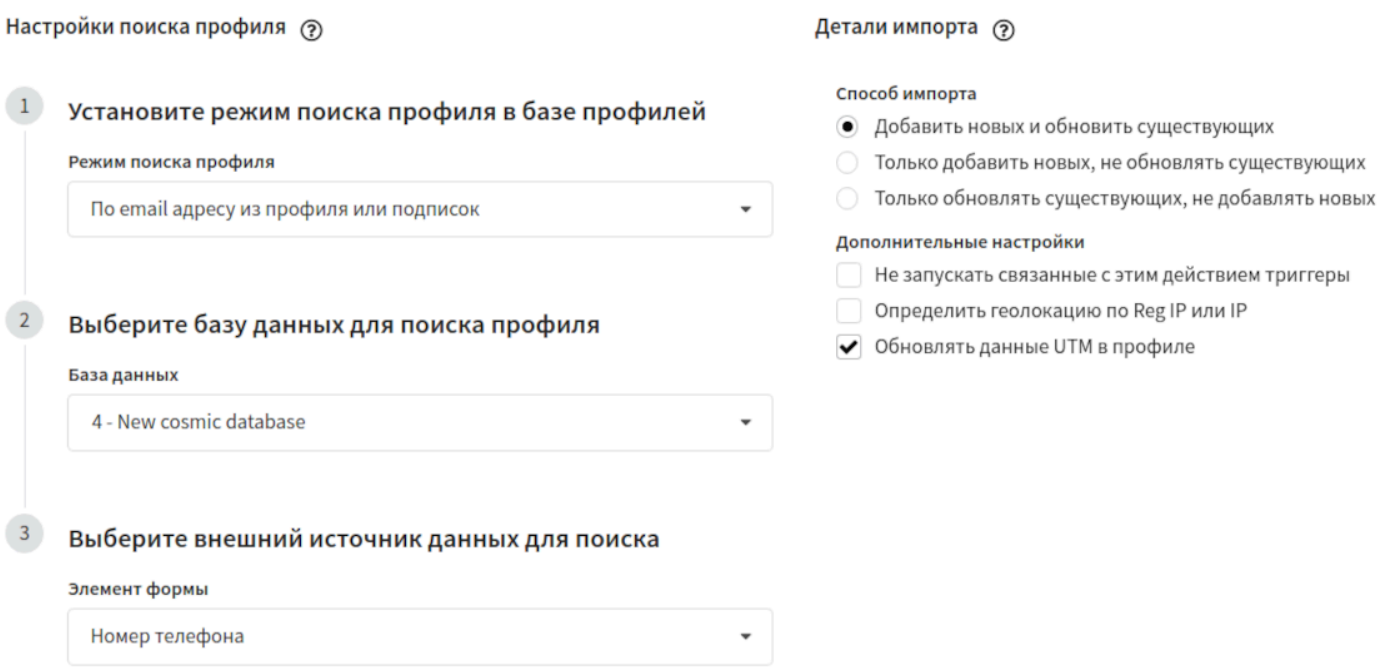
Импорт профилей

Настройки поиска профиля

Укажите режим поиска профиля в базе:

- **по email адресу из профиля или подписок** — по полю профиля **`Email`** или по email адресам из подписок
- **по email адресу из профиля** — только по полю профиля **`Email`**
- **по Message ID** — по идентификатору шаблона сообщения
- **по дополнительному полю** — по пользовательскому полю базы данных

Далее выберите базу данных, в которую будет происходить импорт профиля, и элемент формы, откуда необходимо брать данные для поиска:



В правой части окна можно уточнить **способ импорта:**

- Добавить новых и обновить существующих;
- Только добавить новых, не обновлять существующих;
- Только обновлять существующих, не добавлять новых.

Также есть возможность добавить **дополнительные правила:**

- Не запускать связанные с этим действием **триггеры**: отключить для нового профиля триггерные оповещения при импорте в базу данных;
- Определить **геолокацию** по Reg IP или IP: включить автоматическое определение геолокации клиента по данным в полях **Reg IP** (IP адрес регистрации) или **IP** (последний известный IP адрес);
- Обновлять данные **UTM** в профиле: если форма содержит UTM-метки, то информация об этих метках автоматически обновляется в структуре профиля после импорта.

Соответствие колонок

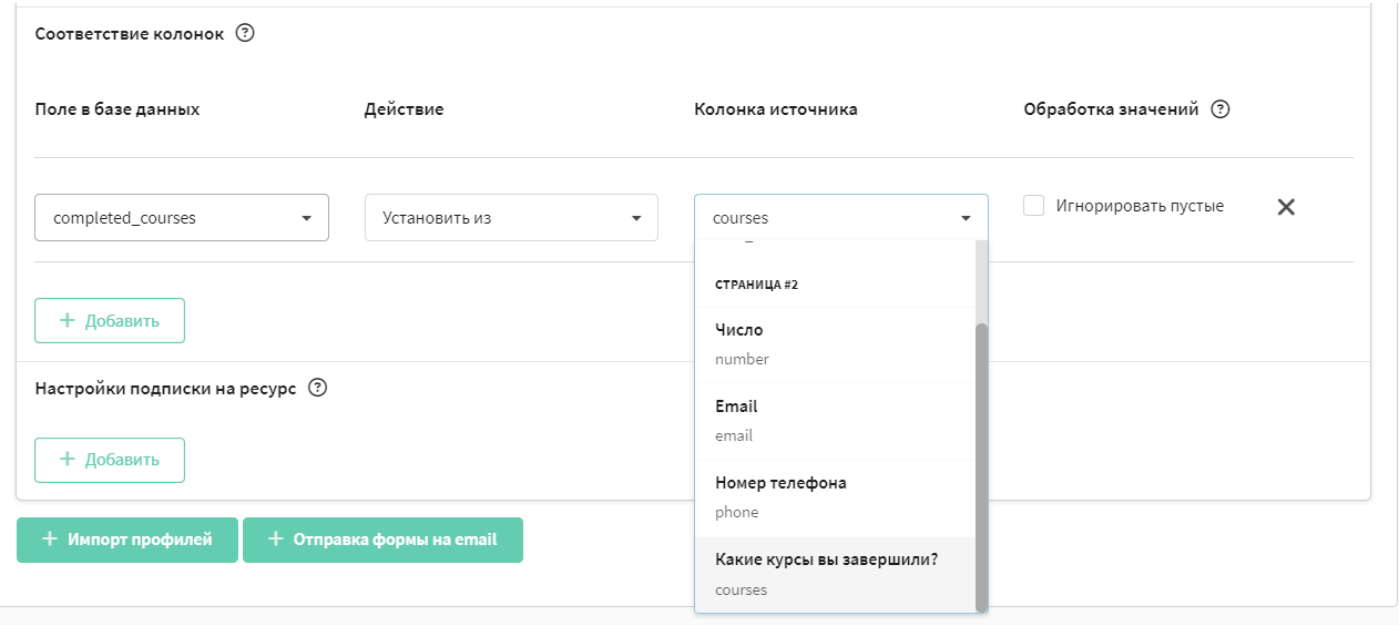
Вы можете установить соответствие между элементами формы и полями в карточке профиля.

В открывающемся меню **“Поле в базе данных”** доступны поля из базы данных: системные и пользовательские. Они необходимы для сопоставления с основными полями формы. Например, поле Email в базе данных соотносится с тем полей формы, в котором требуется указать Email.

В меню **“Действия”** вы можете выбрать три варианта действий: установить из, установить свое или удалить значение.

“Колонка источника” содержит поля, которые существуют в форме. Вы можете задать им имена самостоятельно для более простого сопоставления. Рекомендуем называть их так же, как называются поля, системные и пользовательские, в базе данных.

“Обработка значений” содержит действие **“Игнорировать пустые”**. Если данное поле в форме не заполнено, то его значение не перезаписывает то значение в базе, которое установлено по умолчанию.

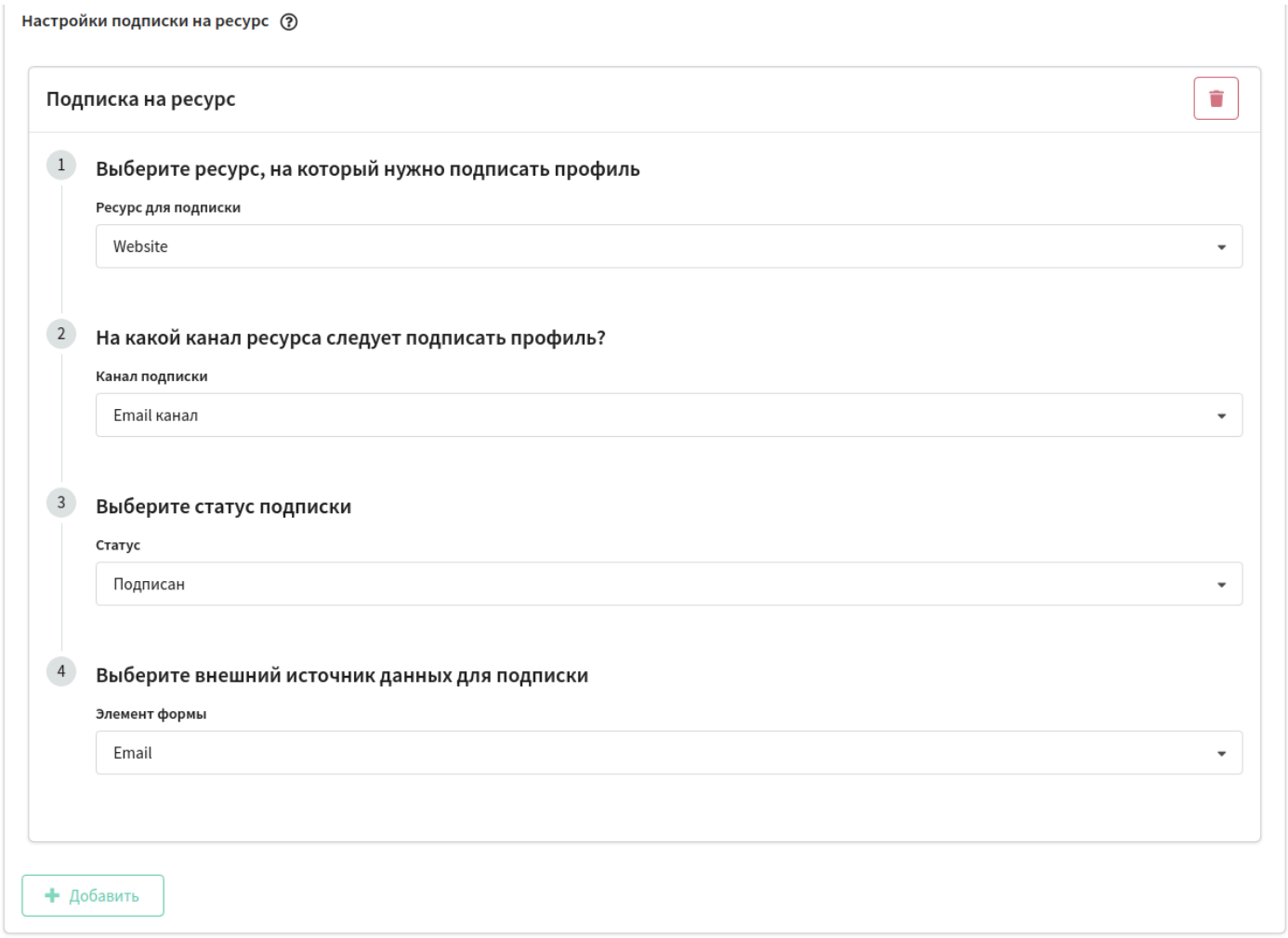


Для того, чтобы при переносе ответов из формы в карточку профиля не возникло ошибки, необходимо верно указать тип поля при создании базы данных и сопоставить их с полями формы.

Элемент формы	Тип поля в базе
Короткий текст	Строка
Email	Строка
Номер телефона	Список телефонов / Строка
Длинный текст	Строка
Число	Целое число
Да/Нет	Boolean
Один вариант	Строка
Множественный выбор	Теги
Выпадающий список	Теги
Дата	Дата (формат RFC 3339 (ISO 8601))
Диапазон	Целое число

Настройки подписки на ресурс

При импорте профилей из форм вы можете сразу добавить клиенту подписку на какой-либо из ваших ресурсов. Выберите ресурс для подписки, канал и статус подписки, а также поле из формы, которое будет использоваться для подписки (например, email).

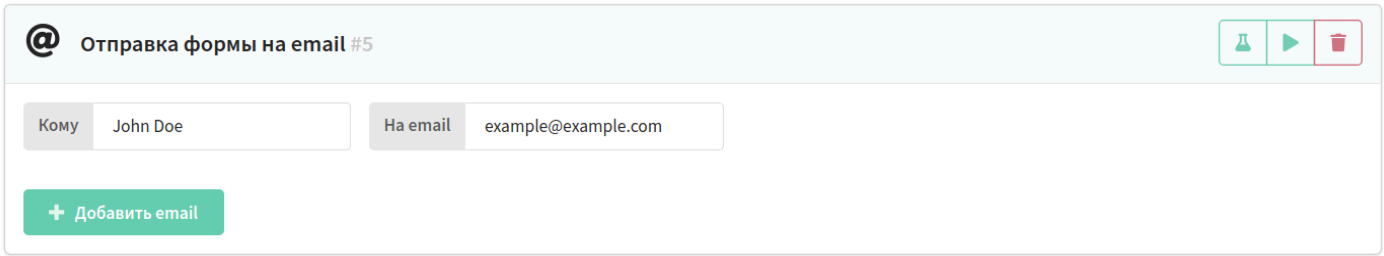


TIP

Можно подписать профили клиентов сразу на несколько ресурсов. Для добавления еще одной подписки нажмите на кнопку **+ Добавить**.

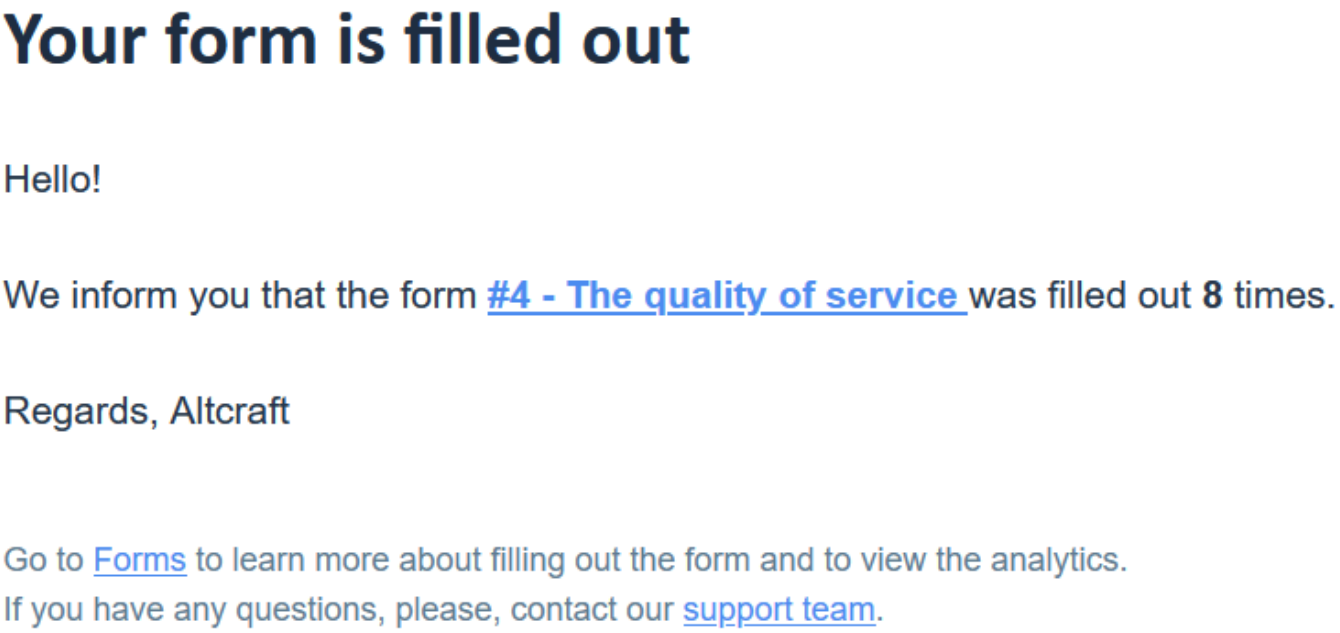
Отправка формы на email

Добавьте это действие, если хотите получать оповещения о заполнении формы. Укажите имя и email-адрес. Можно добавить несколько получателей.



Каждый раз, когда пользователь будет заполнять форму, на указанную почту будет приходить email-письмо со следующими данными:

- оповещение о заполнении формы
- название и идентификатор формы
- ссылка на форму (*Go to view collected data*)



— пробное действие (отправьте тестовое сообщение на указанный email-адрес, чтобы убедиться в правильности введенных данных)

— запустить действие (на почту придет письмо с информацией о том, сколько раз заполняли форму, и архивом, содержащим данные о заполнении)

— удалить действие

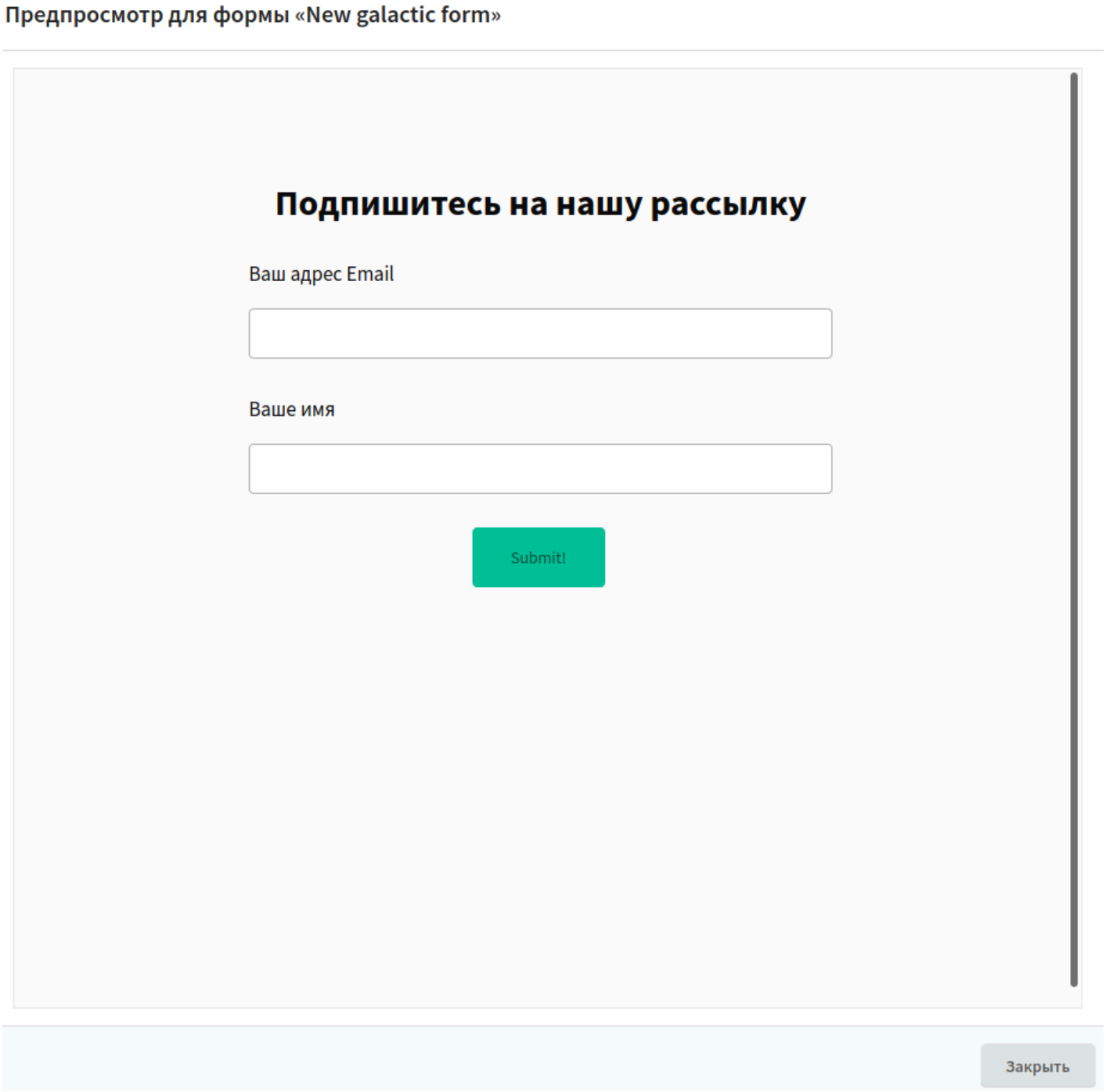
TIP

Кнопки **“Пробное действие”** и **“Запустить действие”** работают при условии, что форма опубликована.

Предпросмотр формы

В любой момент создания и редактирования формы доступен ее предпросмотр. Чтобы увидеть, как будет выглядеть опубликованная форма, нажмите на кнопку предпросмотра в правом верхнем углу страницы

, после чего откроется форма в том виде, в каком ее увидит пользователь.



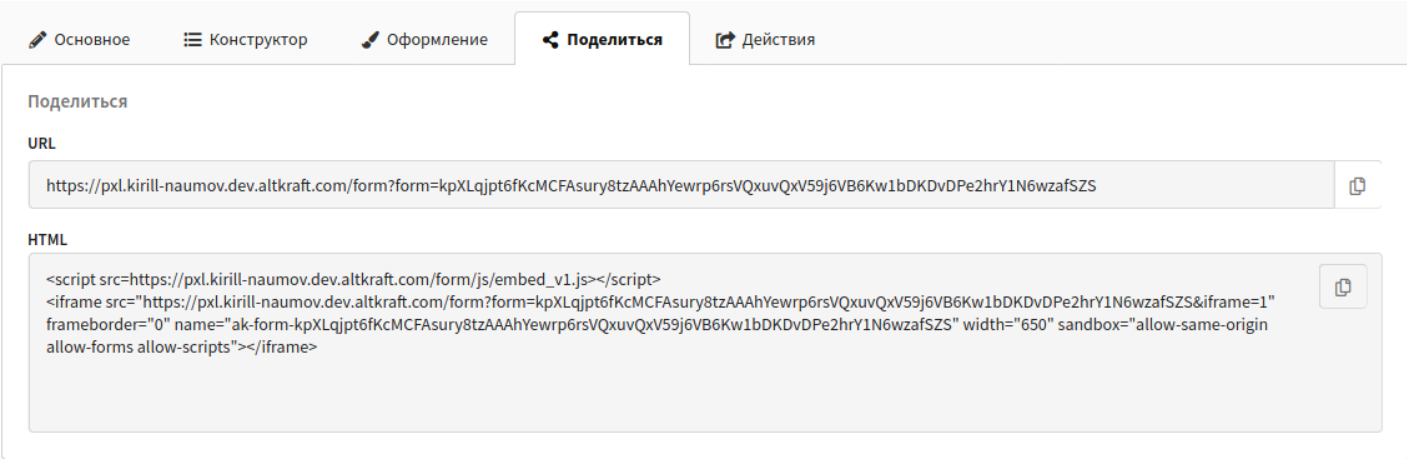
Поделиться

Когда форма создана, ее можно опубликовать. Для этого необходимо активировать переключатель **“Опубликовать”** на вкладке **“Основное”**.

После того как форма опубликована и сохранена, её можно использовать двумя способами:

- **URL:** платформа генерирует URL-адрес, который ведет на самостоятельную страницу с формой. Вы можете отправить эту ссылку подписчикам через любой канал связи, добавив в шаблон сообщения переменную формы. Подробнее.
- **HTML:** платформа генерирует HTML-код, который необходимо добавить на страницу вашего сайта. Форма будет встроена в виде iframe-элемента.

URL и HTML-код формы доступны на вкладке **“Поделиться”**:



TIP

После заполнения и отправки формы пользователь может быть перенаправлен на системную финальную страницу или на любую другую страницу, указанную в настройках формы. Обратите внимание, что если вы встраиваете форму на сайт, то эта страница откроется не внутри iframe-элемента на текущей странице, а произойдет редирект пользователя на новую страницу.

Аналитика данных

Описание

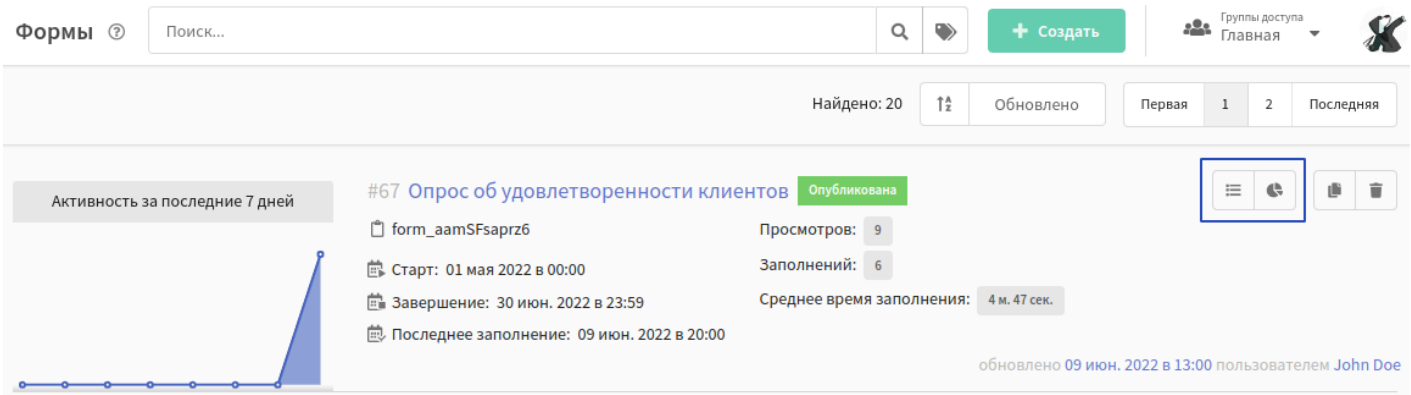
Проанализировать результаты, полученные после заполнения формы, вы можете в разделе “Собранные данные” или в разделе “Аналитика”.

- **Собранные данные**

  показывают ответы определенного пользователя.
- **Аналитика**

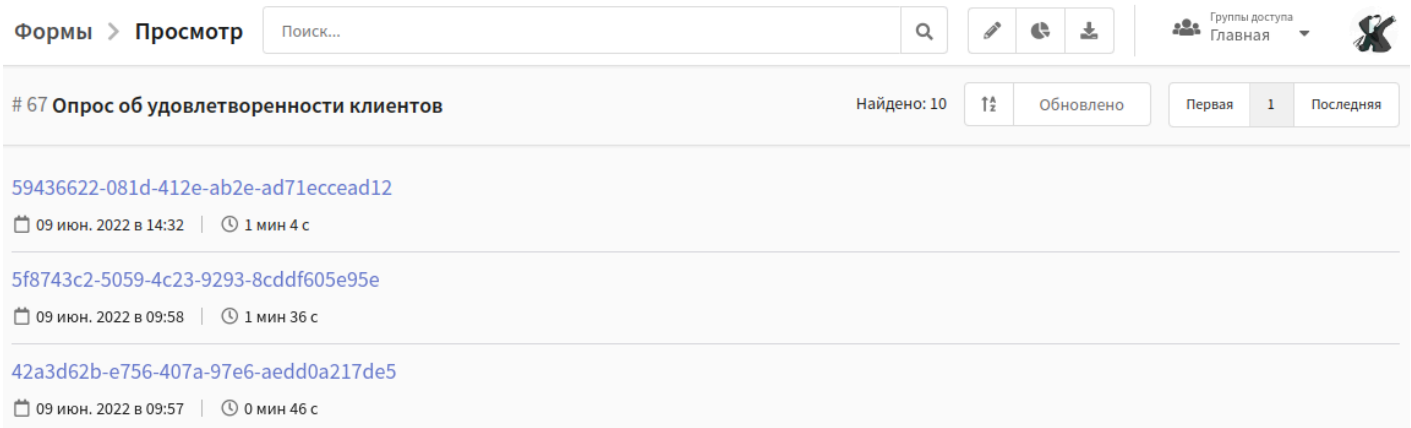
  предоставляет общую сводку по форме (просмотры, заполнения) и статистику по конкретному вопросу.

Чтобы перейти в один из разделов, найдите нужную форму в общем списке и нажмите соответствующую иконку на панели инструментов:



Собранные данные

Откройте форму и перейдите в раздел **“Собранные данные”.** Здесь вы увидите список идентификаторов всех заполненных форм. Нажмите на идентификатор, чтобы получить подробную информацию.



INFO

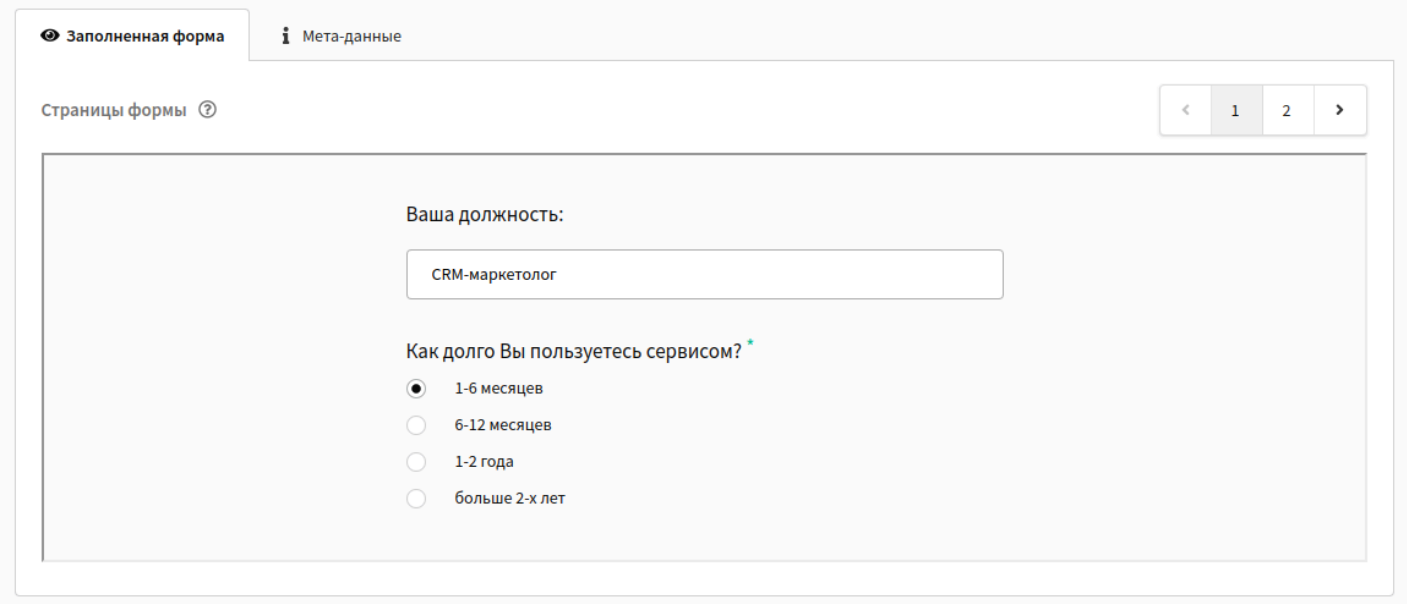
В правом верхнем углу расположены следующие кнопки:

— перейти в окно редактирования формы.

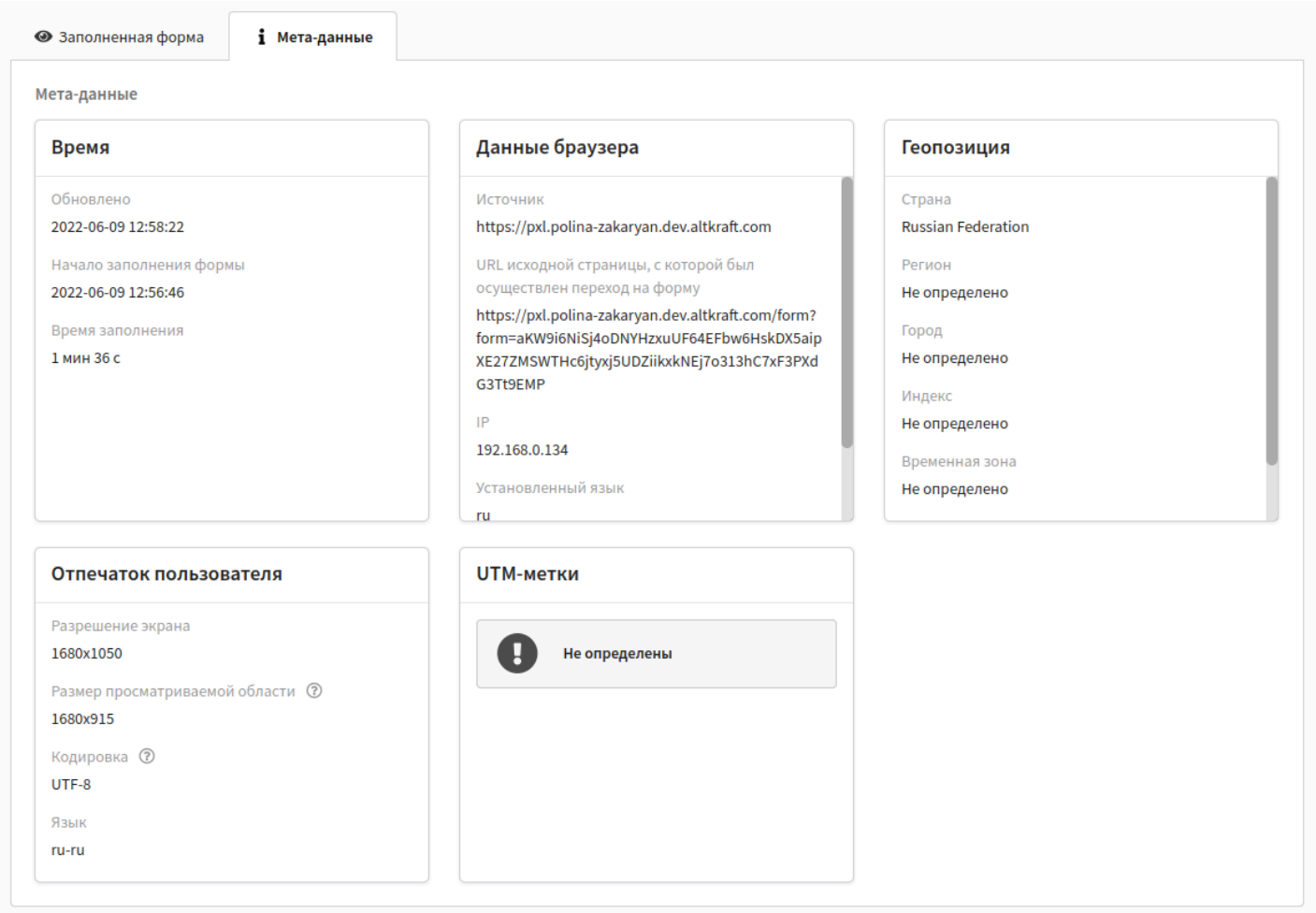
— открыть аналитику по форме.

— экпортировать данные о заполнениях формы за определенный период.

На вкладке **“Заполненная форма”** вы можете узнать, как именно пользователь заполнил каждое поле:



Вкладка **“Мета-данные”** отображает детальную информацию о заполнении формы: время, данные браузера, геопозиция, отпечаток пользователя и UTM-метки.



Аналитика формы

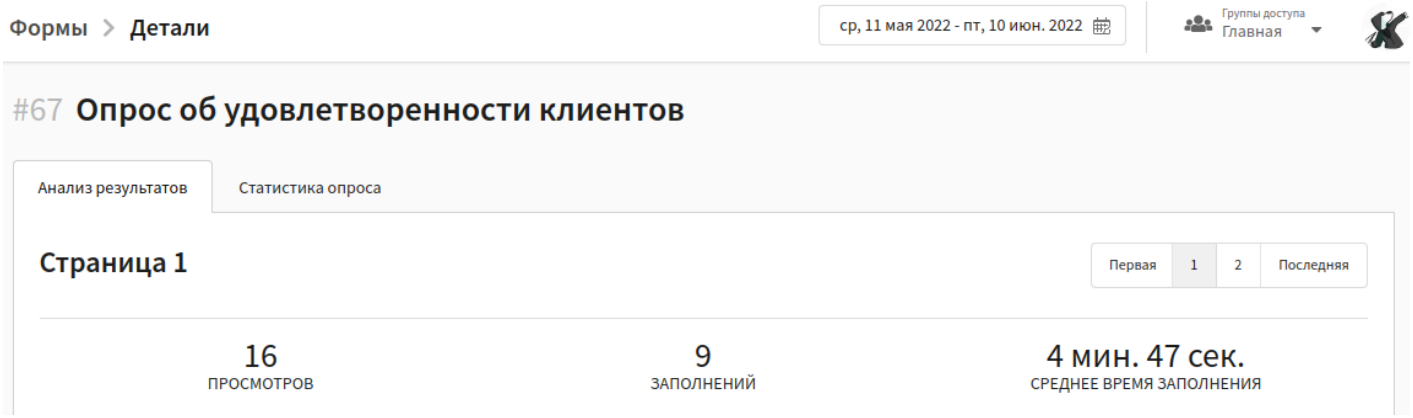
Откройте форму и перейдите в раздел **“Аналитика”**. На верхней панели откройте календарь и выберите период, за который хотите получить данные.

Аналитика предоставляет следующую информацию:

- **Просмотры** (сколько раз была загружена страница с формой).
- **Заполнения** (сколько раз форма была заполнена и отправлена).
- **Среднее время заполнения**.

INFO

Данные показатели являются неуникальными. Если один и тот же пользователь просматривал или заполнял форму несколько раз, будут засчитаны все просмотры и заполнения.



В аналитике формы доступны две вкладки:

- **Анализ результатов** — статистика по конкретному полю в форме.
- **Статистика опроса** — общая сводка по форме.

Анализ результатов

На этой вкладке отображается статистика по заполнению конкретного поля в форме. Для переключения между страницами формы используйте кнопки в правом верхнем углу.

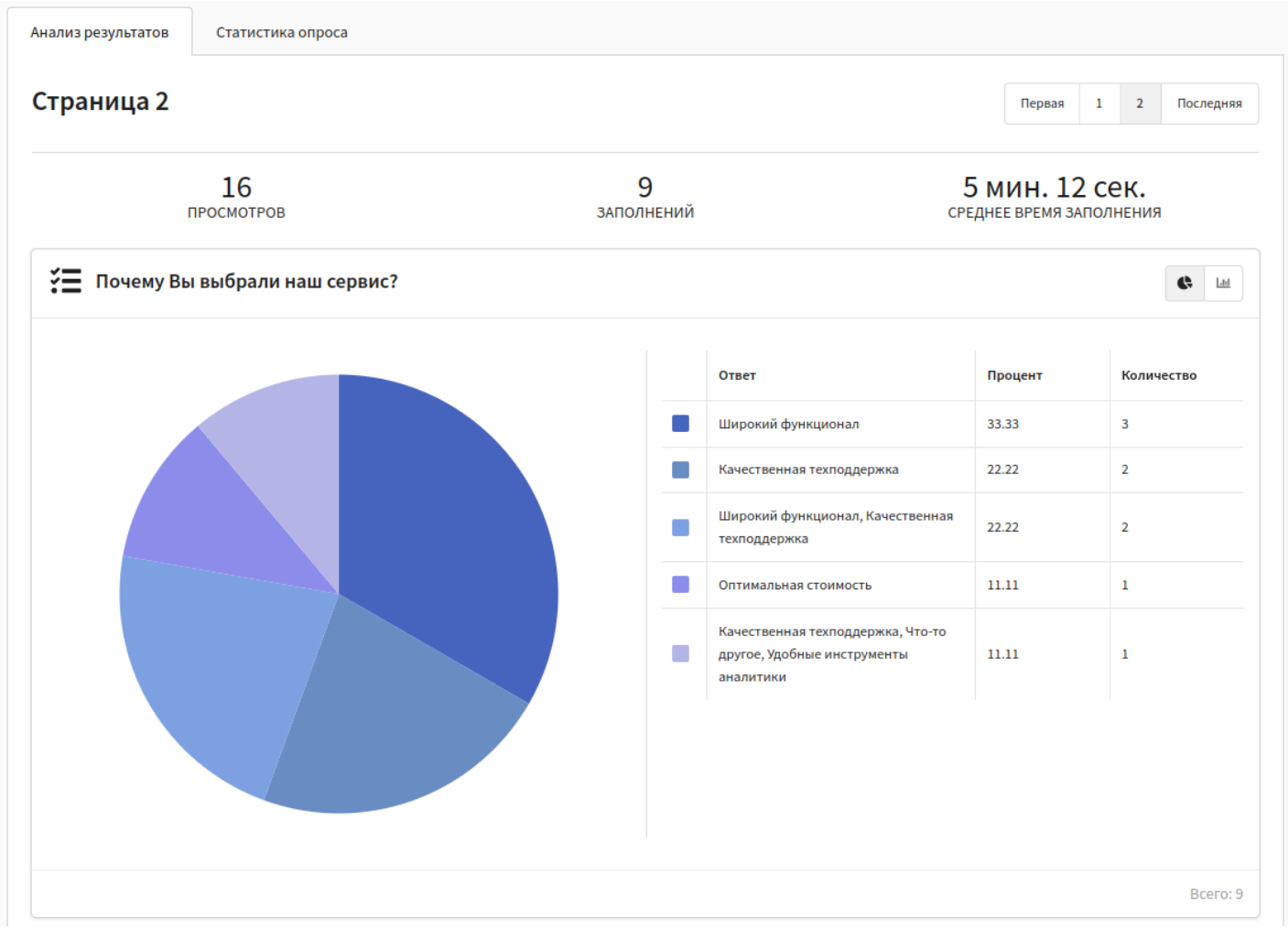
Данные по каждому полю собраны в отдельном блоке. Здесь вы можете посмотреть вопрос и ответы, которые дали пользователи. Внизу блока показано, сколько всего раз было заполнено поле.

Большинство данных представлены в процентном и количественном соотношении.

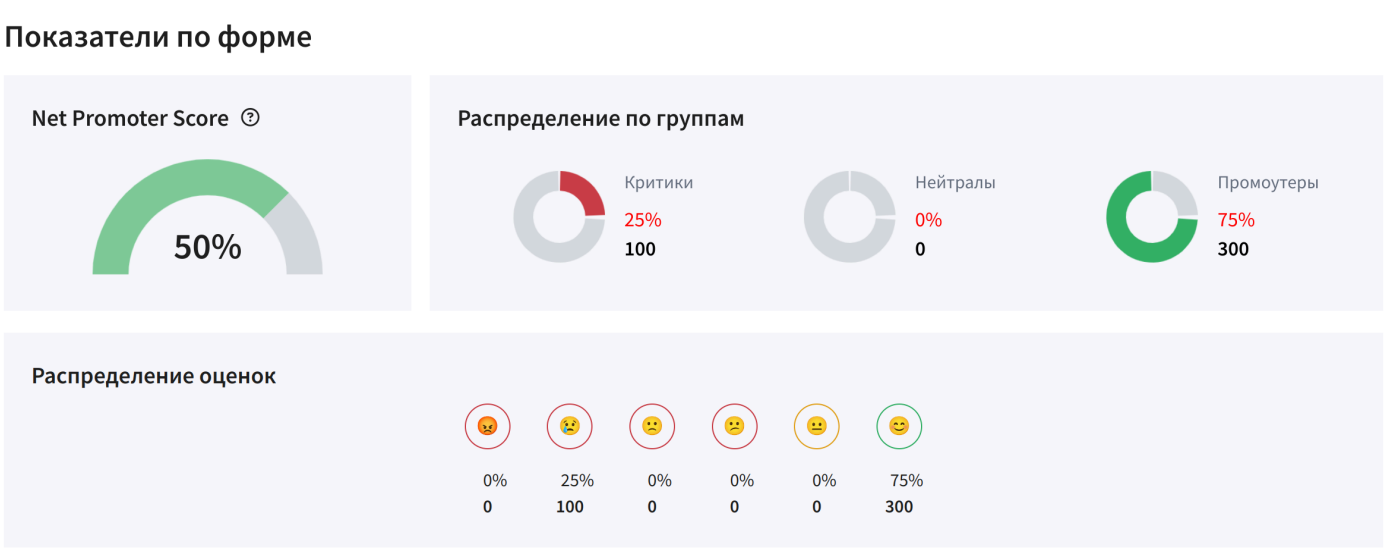
Дополнительно возможна их визуализация в виде бар —

или пайчарта

.



Для форм, содержащих NPS-тестирование, будет представлен счетчик Net Promoter Score и распределение ответов по группам:



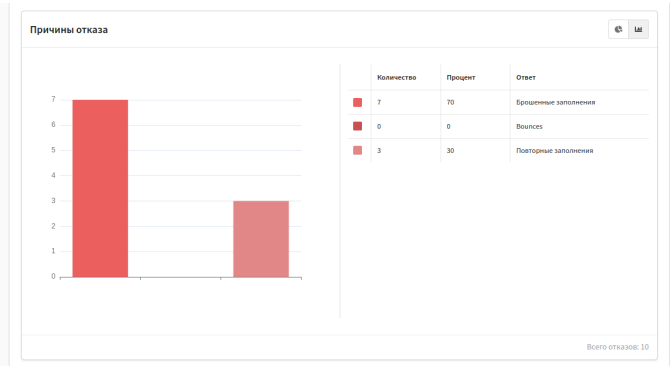
Статистика опроса

На этой вкладке вы можете найти следующую информацию:

Просмотры страниц

	Просмотры покажут, сколько раз была загружена каждая страница. С помощью этого показателя вы можете узнать, на какой странице пользователи чаще всего бросают форму, и в дальнейшем оптимизировать её.

Причины отказа

	1. **Брошенные заполнения** — пользователь частично или полностью заполняет форму, но не отправляет ее. 2. **Bounces** — при заполнении формы возникли технические проблемы (например, остановка сервера). 3. **Повторные заполнения** — один и тот же пользователь заполняет форму несколько раз. Хотя такие заполнения отображаются в причинах отказа, они также учитываются в общей сводке.

Аналитические срезы отчёта

Ниже вы можете дополнительно сгруппировать респондентов по следующим критериям:

- по рефереру (с какого источника был совершен переход на форму)
- по браузеру (в каком браузере была открыта форма)
- по ОС
- по языку
- по часовому поясу
- UTM content
- UTM medium
- UTM source
- UTM campaign
- UTM term

Экспорт формы

Выберите форму. Перейдите в раздел **Собранные данные**. В верхней части страницы расположена кнопка экспорта

. Нажмите на неё.

В календаре укажите, за какой период экспортировать данные. Нажмите **Применить**. Перейдите в **Задания** и скачайте файл.

Поиск...

Группы доступа
Главная

67 **Опрос об удовлетворенности клиентов**

Первая 1 Последняя

Экспортировать данные за период

2022-05-01 - 2022-05-10

May 2022

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

Закрыть Применить

2ca9b70c-4135-48bd-9e67-45ce769386bb
18 мая 2022 в 11:25 | 41 мин 2 с

41c99b8c-def6-41fa-b008-8c39c90d198e
18 мая 2022 в 11:39 | 0 мин 49 с

1de38069-e044-4956-a2f2-865c7b212901
19 мая 2022 в 14:33 | 0 мин 54 с

c317a8c5-b868-4894-9800-a060661e1284
20 мая 2022 в 10:16 | 0 мин 39 с

c41467b8-eeb7-4360-956e-55a4f9f863fb
09 июн. 2022 в 09:49 | 0 мин 18 с

Связывание данных канала и формы

Описание

Если данные профиля, который заполняет форму, уже есть в одной из ваших баз данных, то платформа может идентифицировать пользователя в тот момент, когда он откроет форму. Это позволяет настроить автоматическое заполнение полей формы данными из профиля.

Например, в форме у вас есть поля, где вы просите пользователя ввести электронную почту и номер телефона, но вы и так уже знаете эту информацию. Вы можете облегчить жизнь пользователю, отправив ему ссылку на форму, где уже будут указаны его email и телефон.

Данная операция возможна, только если вы рассылаете форму с помощью одного из каналов, а не встраиваете её HTML-код в структуру сайта. Настройка автоматической идентификации профиля при заполнении формы происходит в момент создания шаблона для рассылки.

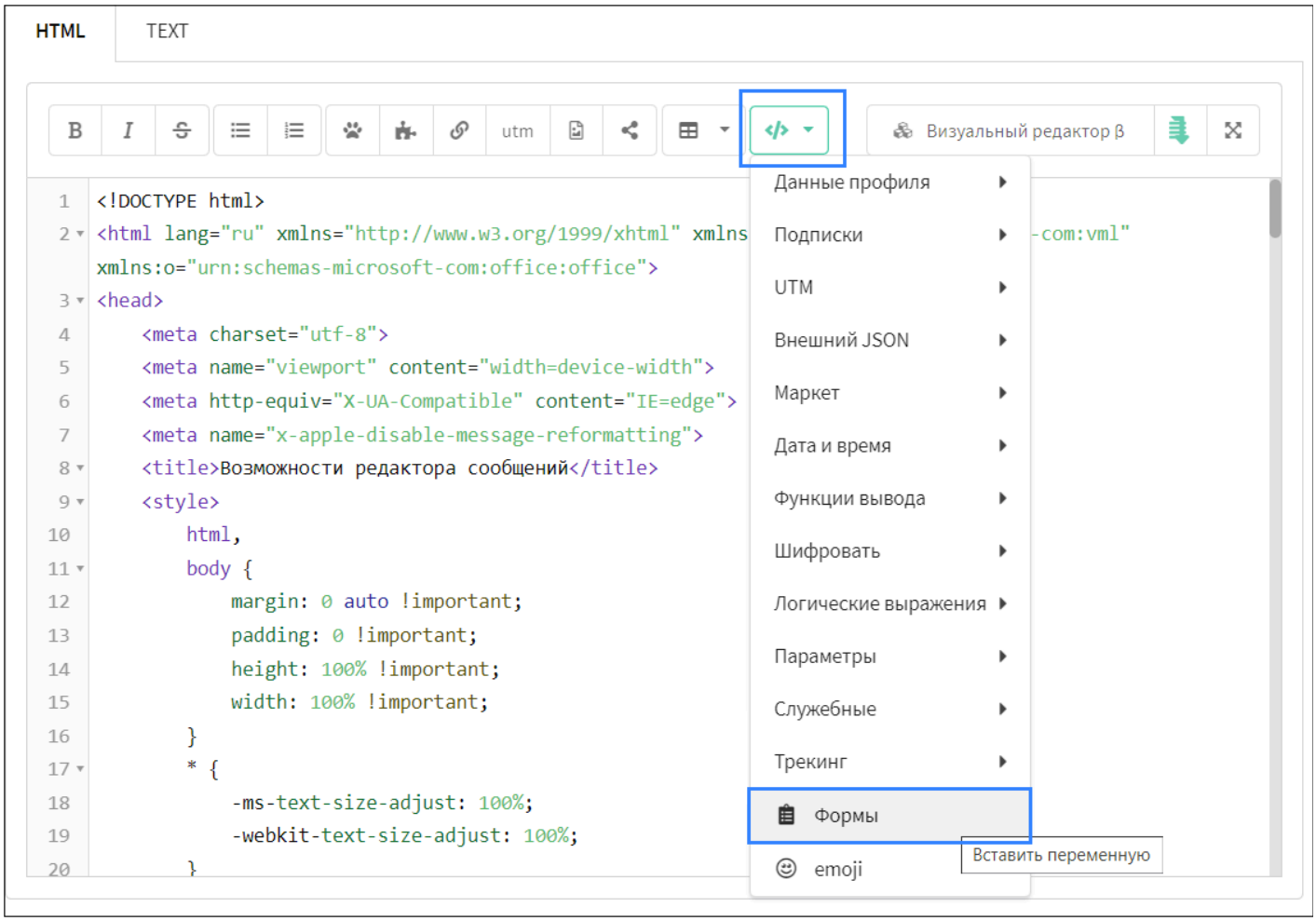
TIP

Видео-гайд по настройке автозаполнения полей формы доступен в **центре поддержки**.

Как добавить ссылку на форму в шаблон

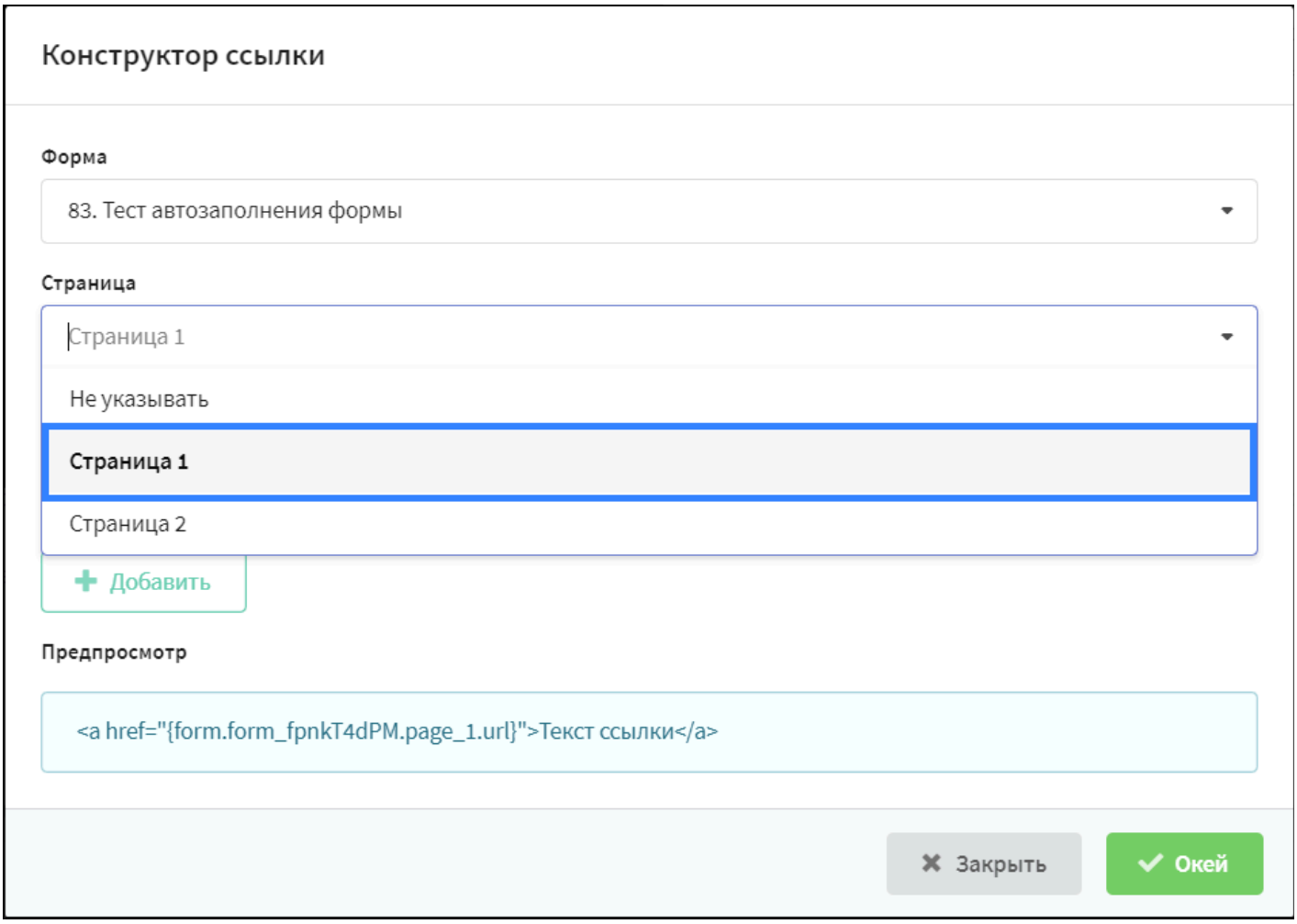
1. В разделе **“Шаблоны”** создайте новый шаблон и перейдите в редактор.

2. Установите курсор на ту строку, куда хотите вставить ссылку на форму. Далее откройте список переменных с помощью кнопки `</>` и выберите пункт **“Формы”:**



3. В модальном окне **“Конструктор ссылки”** выберите форму.

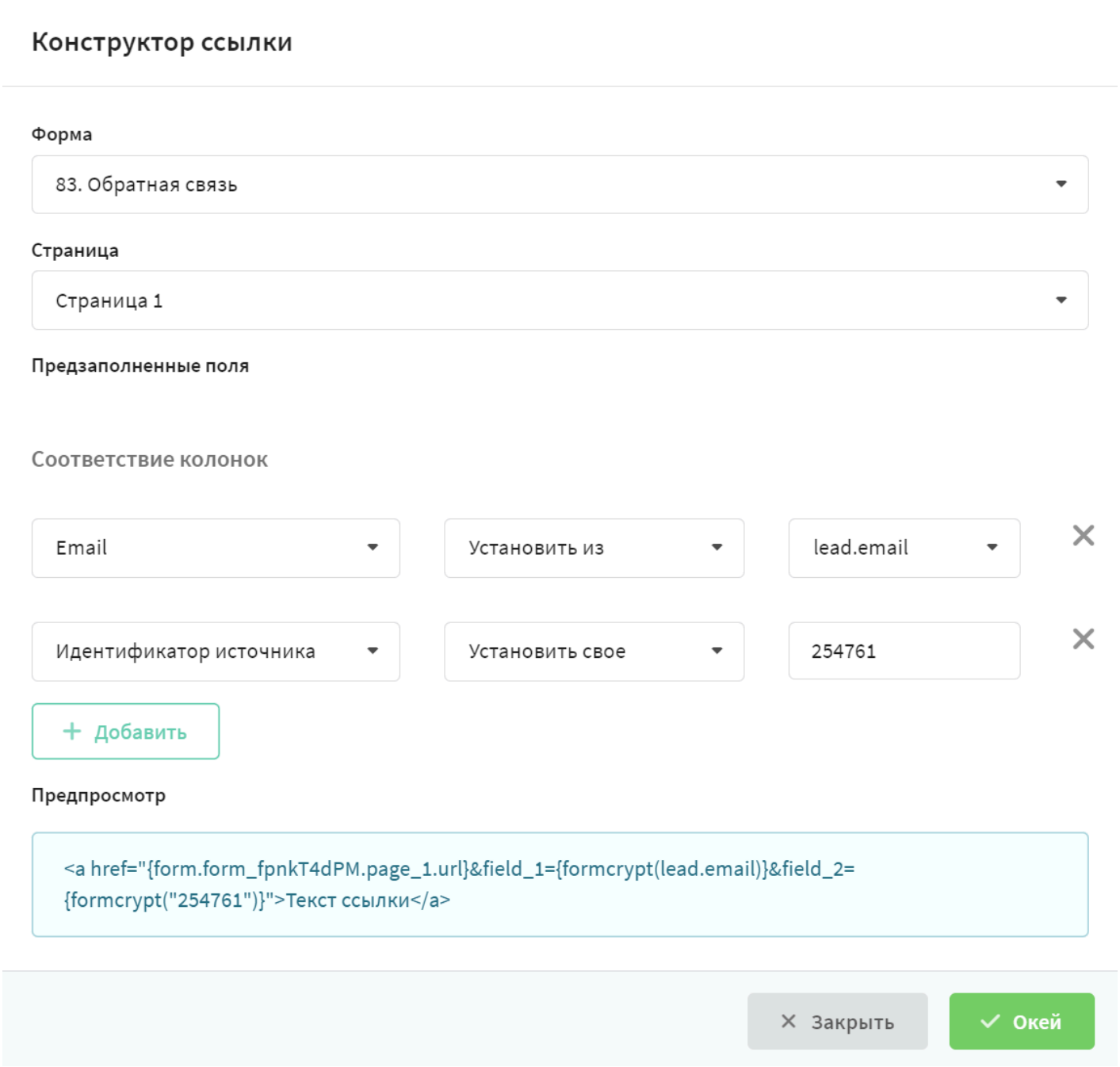
Если в форме несколько страниц, то вы можете выбрать страницу, на которую хотите направить пользователя после перехода по ссылке. По умолчанию пользователь перейдет на первую страницу формы.



4. Нажмите кнопку **“Добавить”**. В первом выпадающем списке выберите поле формы, в которое хотите подставить данные.

4.1. Если вы хотите заполнить поле данными профиля из базы, то во втором выпадающем списке выберите **“Установить из”**. Затем из третьего выпадающего списка выберите поле базы, в котором хранятся нужные данные о клиенте.

4.2. Если вы хотите подставить в поле фиксированное значение, то выберите **“Установить свое”** и введите значение вручную.



В разделе **“Предпросмотр”** отобразится код ссылки, например:

```
<a href="{form.form_fpnkT4dPM.page_1.url}&field_1={formcrypt(lead.email)}&field_2=
{formcrypt("254761")}">Текст ссылки</a>
```

- **Текст ссылки** — по умолчанию текст ссылки выглядит таким образом. Вы можете изменить его прямо в теле шаблона, написав нужный.
- `form_fpnkT4dPM` — это идентификатор формы.
- `page_1` — страница формы, на которую перейдет пользователь при клике на ссылку.
- `field_1` и `field_2` — идентификаторы полей “Email” и “Идентификатор источника” в форме.
- `(lead.email)` — переменная платформы, которая подтянет email-адрес профиля из базы в поле формы. Какие переменные доступны, читайте здесь.
- `"254761"` — фиксированное значение для предзаполнения поля.
- `{formcrypt}` — функция шифрования данных.

CAUTION

При передаче данных профиля использование функции `{formcrypt}` для шифрования обязательно.

5. Нажмите кнопку “Окей” — и ссылка, отображающаяся в предпросмотре, будет добавлена в тело сообщения.

Автозаполнение элементов формы без шифрования

В примерах выше значения параметров в ссылке шифровались с помощью функции `{formcrypt}`. Платформа знает алгоритм шифрования и может декодировать такие параметры. Однако в некоторых случаях в ссылку на форму требуется подставить параметр со своим значением, не зашифрованным функцией `{formcrypt}`. Данный параметр не может быть декодирован платформой.

Чтобы подставить значение такого параметра в форму, необходимо добавить дополнительный параметр `prefill` и передать ему любое значение (`prefill=true` или `prefill=1`). Такой подход часто используется для автозаполнения скрытых полей формы.

```
<a href="{form.form_BMteOxecM.url}&custom_field=client_id&prefill=true"
name="link1">Ссылка</a>
```

Условная постраничная логика в формах и опросах

Описание

Условная постраничная логика позволяет настраивать логику переходов между страницами формы. В зависимости от ответов, которые дает пользователь, можно изменять порядок показа страниц, а также задавать условия показа той или иной страницы.

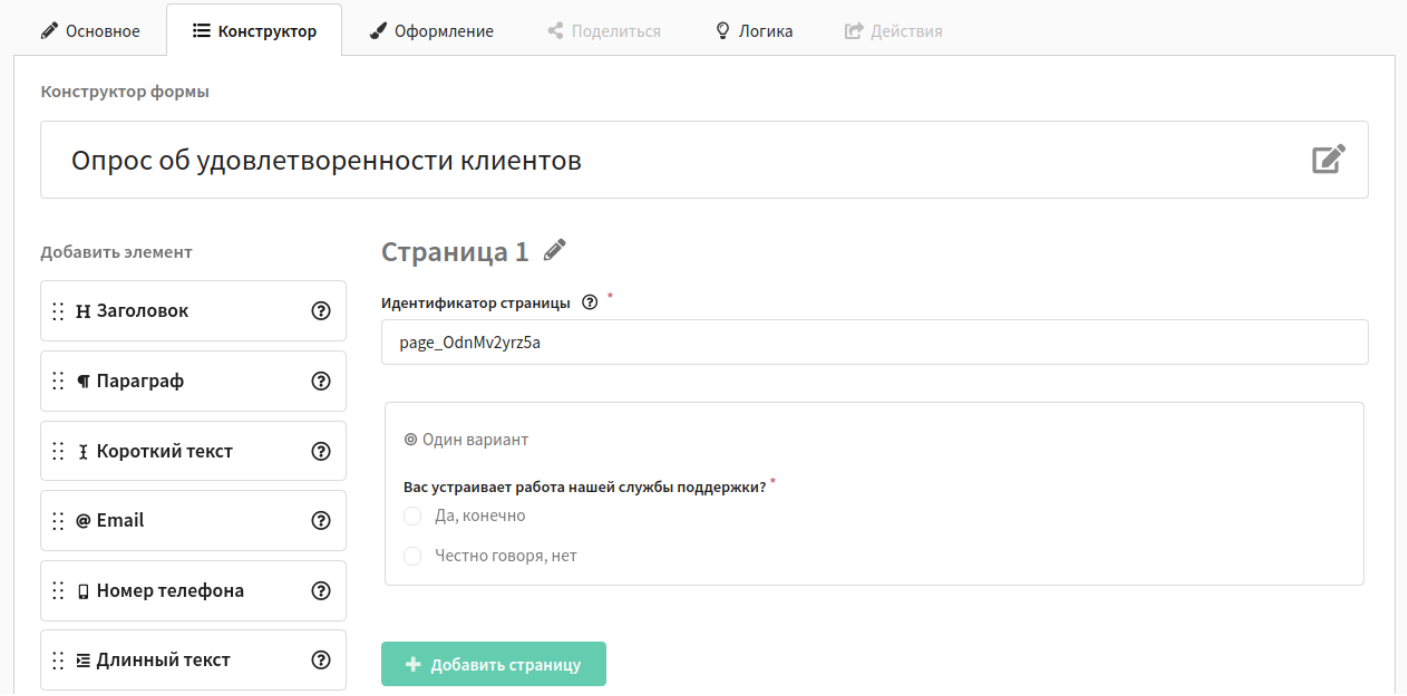
Как настроить логику

Шаг 1. Активация логики

Создайте форму и заполните основные настройки. После этого перейдите на вкладку **“Логика”** и активируйте опцию, кликнув на переключатель в правом верхнем углу.

Шаг 2. Добавление страниц и элементов

Вернитесь на вкладку **“Конструктор”**. Создайте страницы формы и добавьте в них нужные элементы:

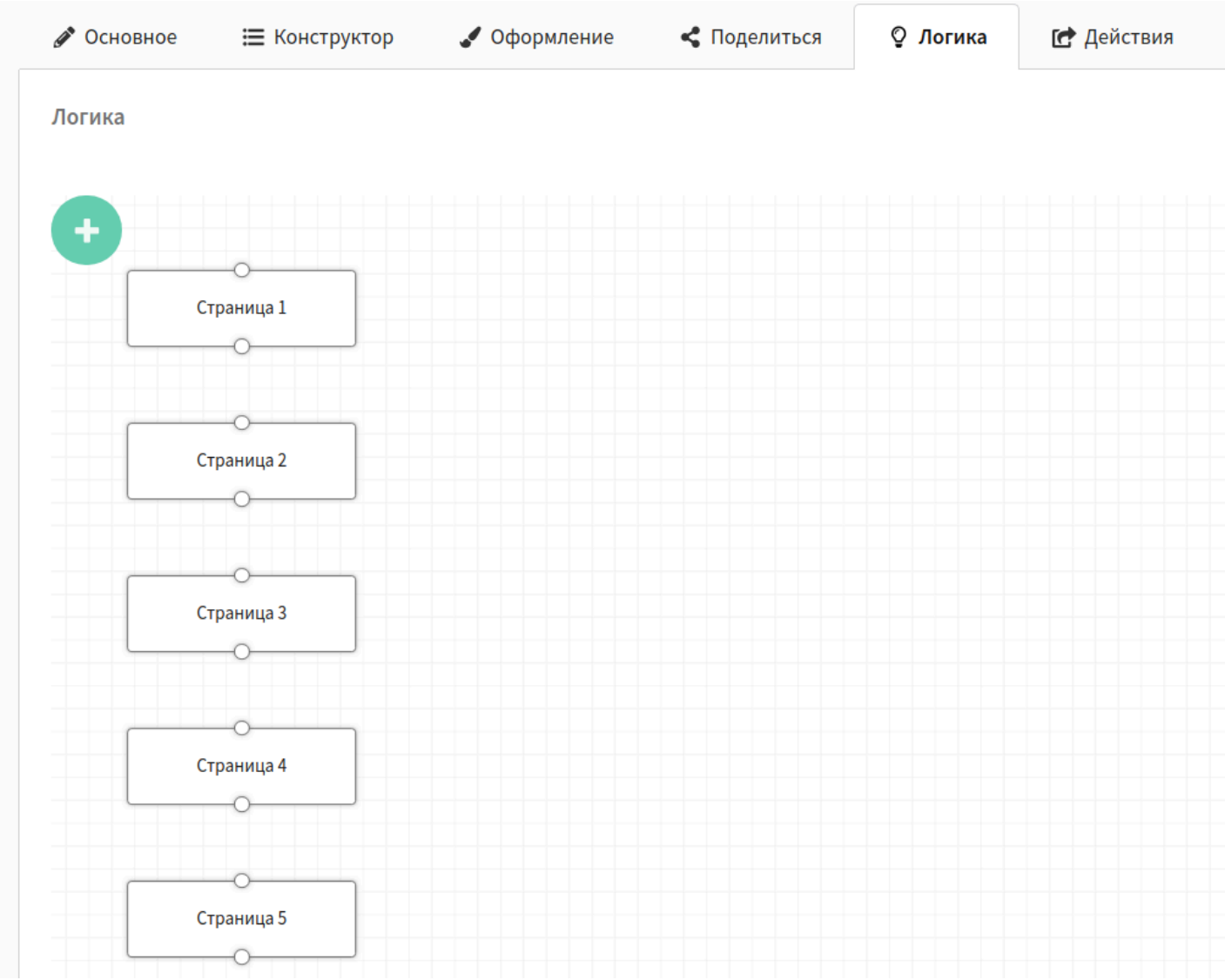


TIP

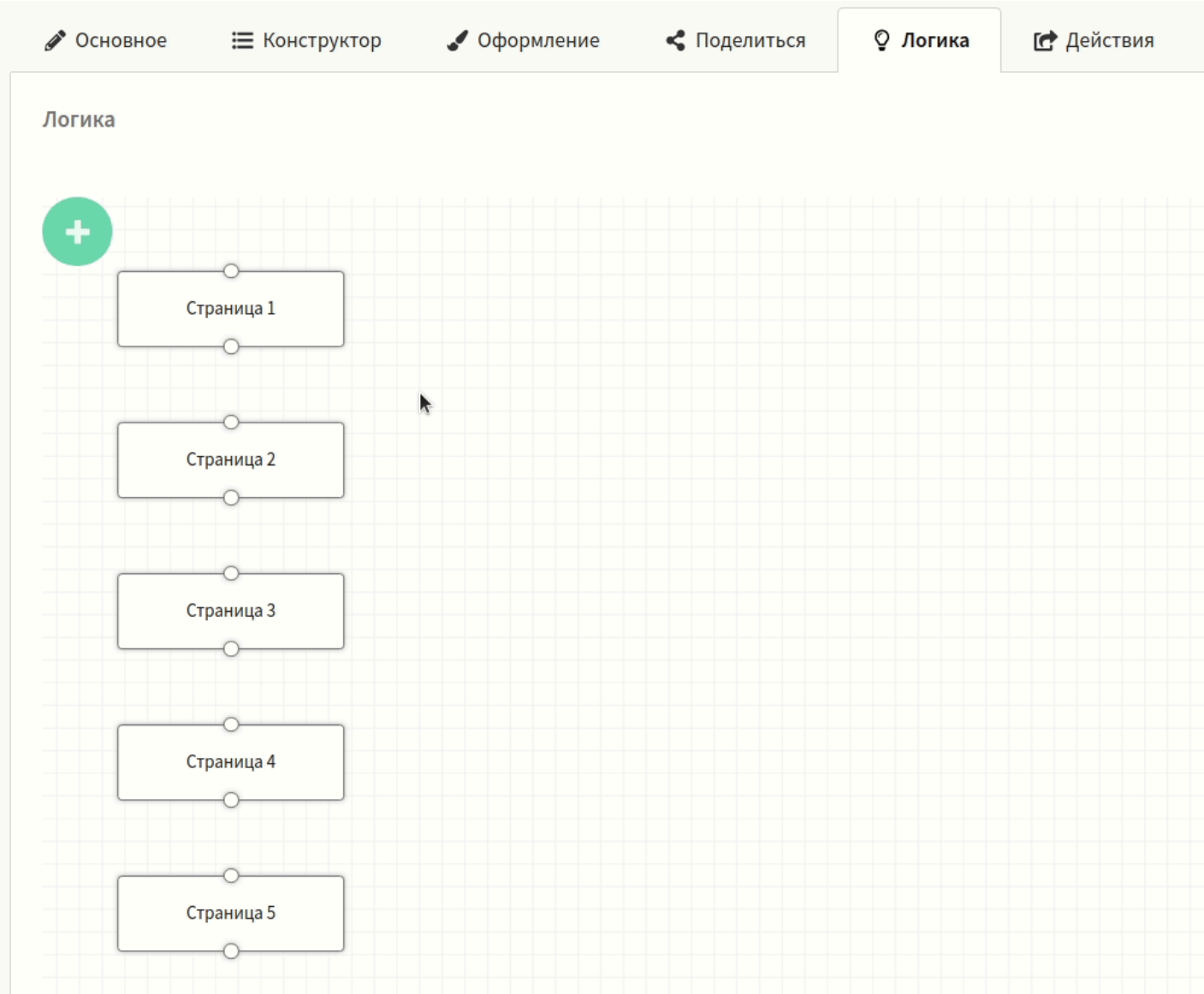
Обратите внимание, что вы настраиваете логику переходов **между страницами** формы. Чтобы после определенного ответа на вопрос пользователь переходил на нужную страницу, необходимо создавать отдельные страницы для каждого вопроса, после которого произойдет переход.

Шаг 3. Добавление условия

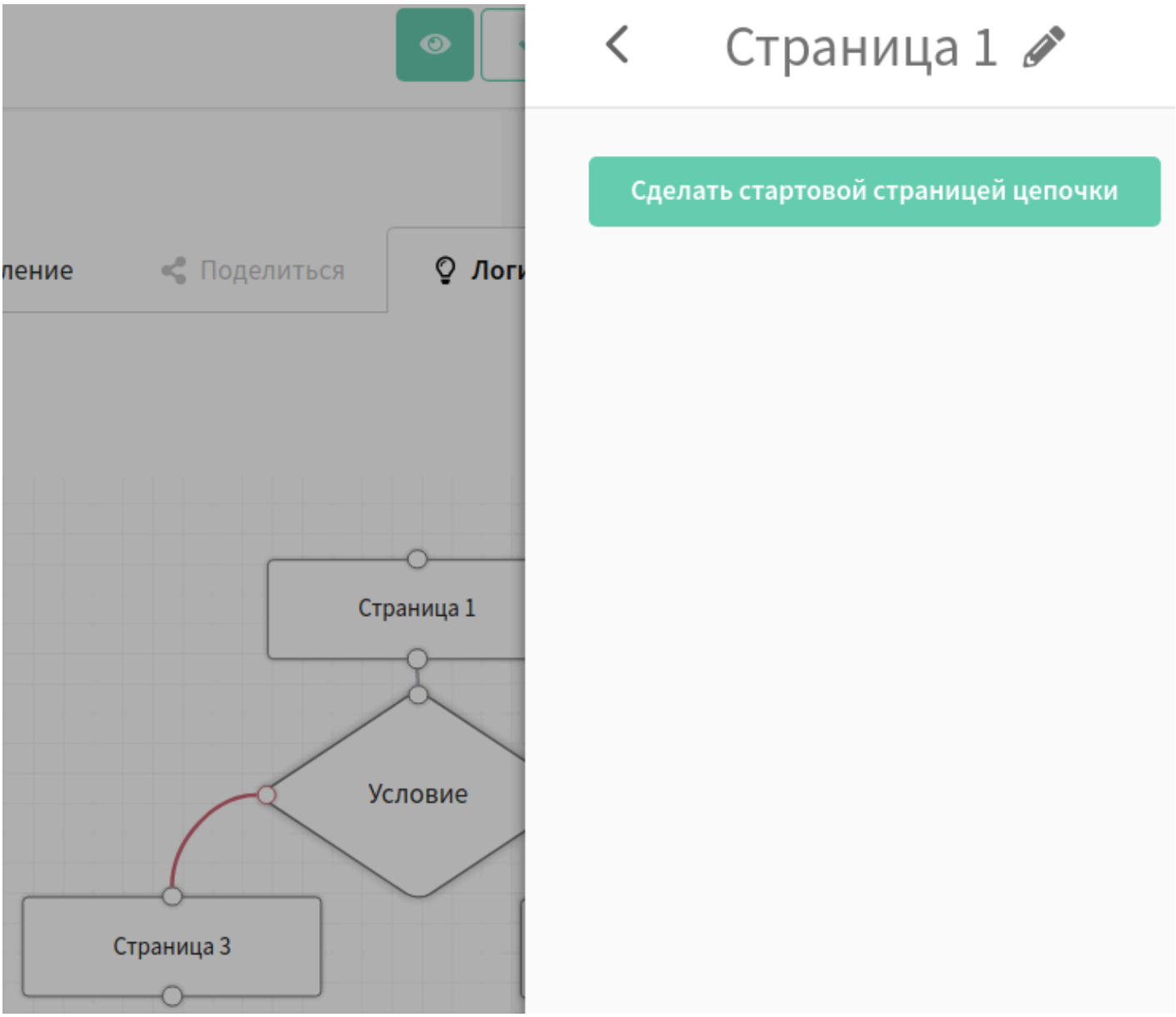
Снова перейдите на вкладку **“Логика”**. Вы увидите, что все созданные страницы выстроились друг за другом:



Чтобы изменить порядок, кликните на блок страницы и, зажав левую кнопку мыши, перетащите его.



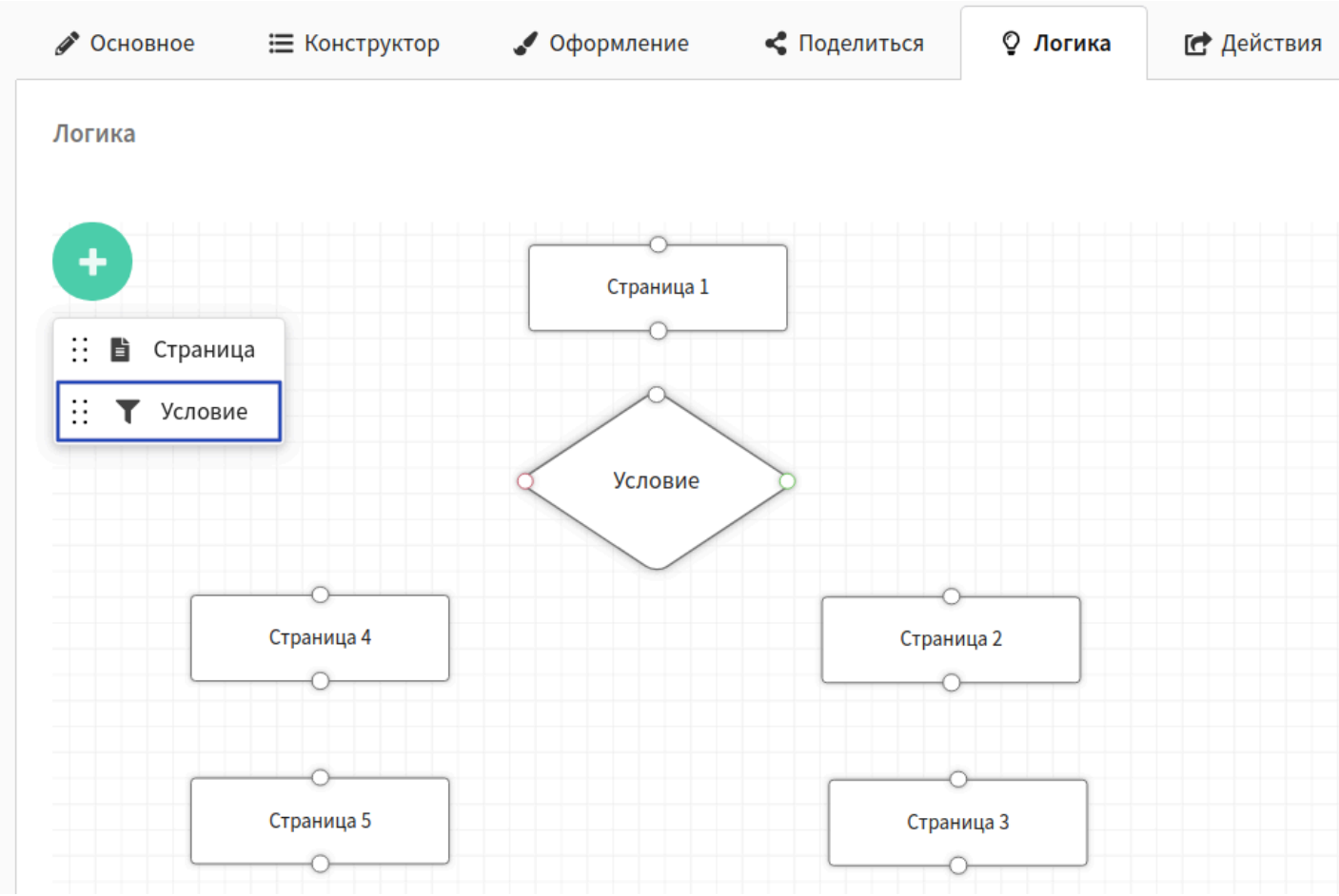
Выберите страницу, с которой пользователь начнет проходить опрос. Для этого нажмите на блок страницы — справа появится окно. Кликните кнопку **“Сделать стартовой страницей цепочки”**. Стартовой страницей может быть любая страница в форме.



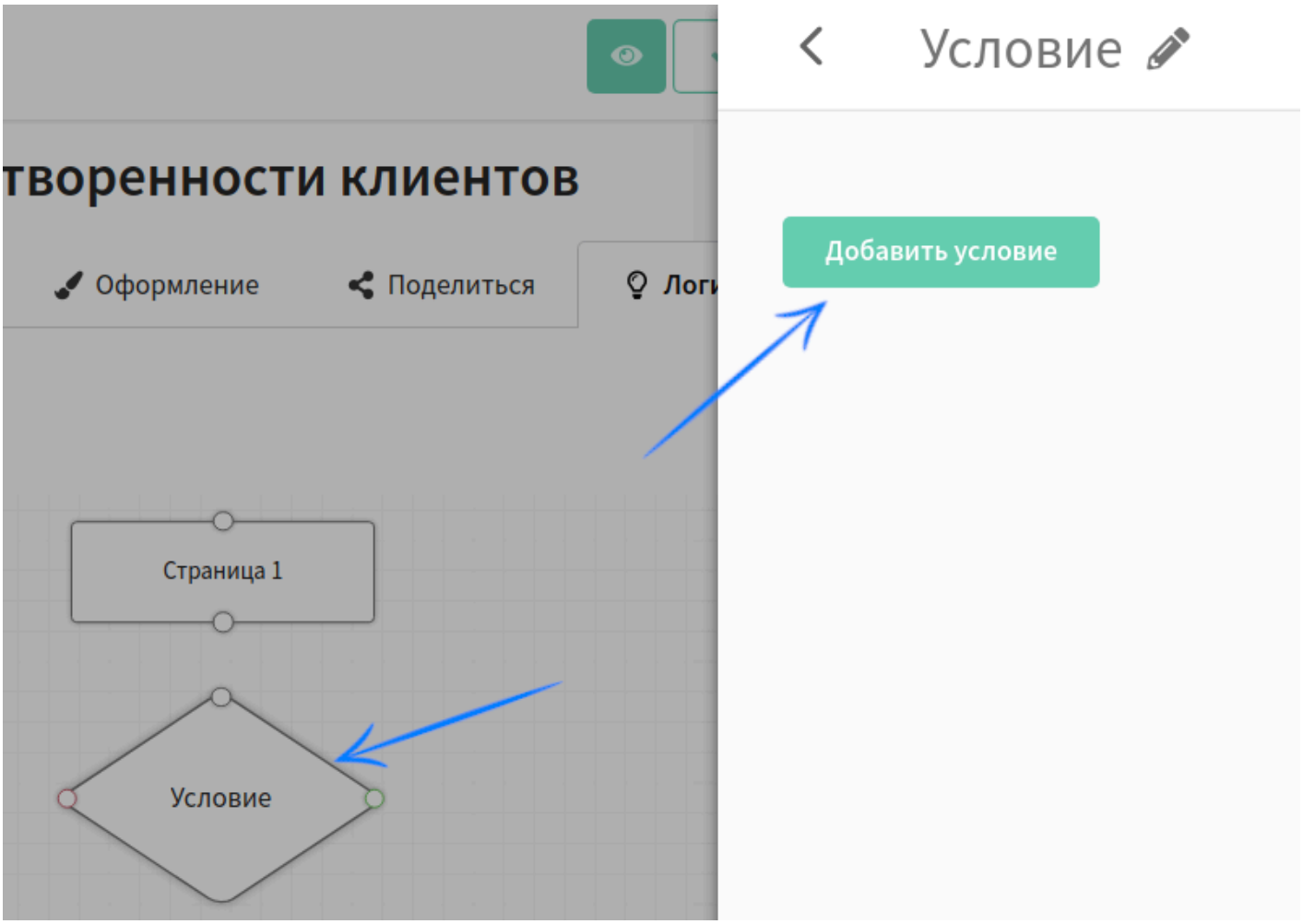
Теперь необходимо добавить условие. Условие — это элемент на странице формы, предполагающий выбор пользователя (“Да/нет”, “Один вариант”, “Выпадающий список”, “Диапазон”).

Чтобы добавить блок условия, нажмите на кнопку

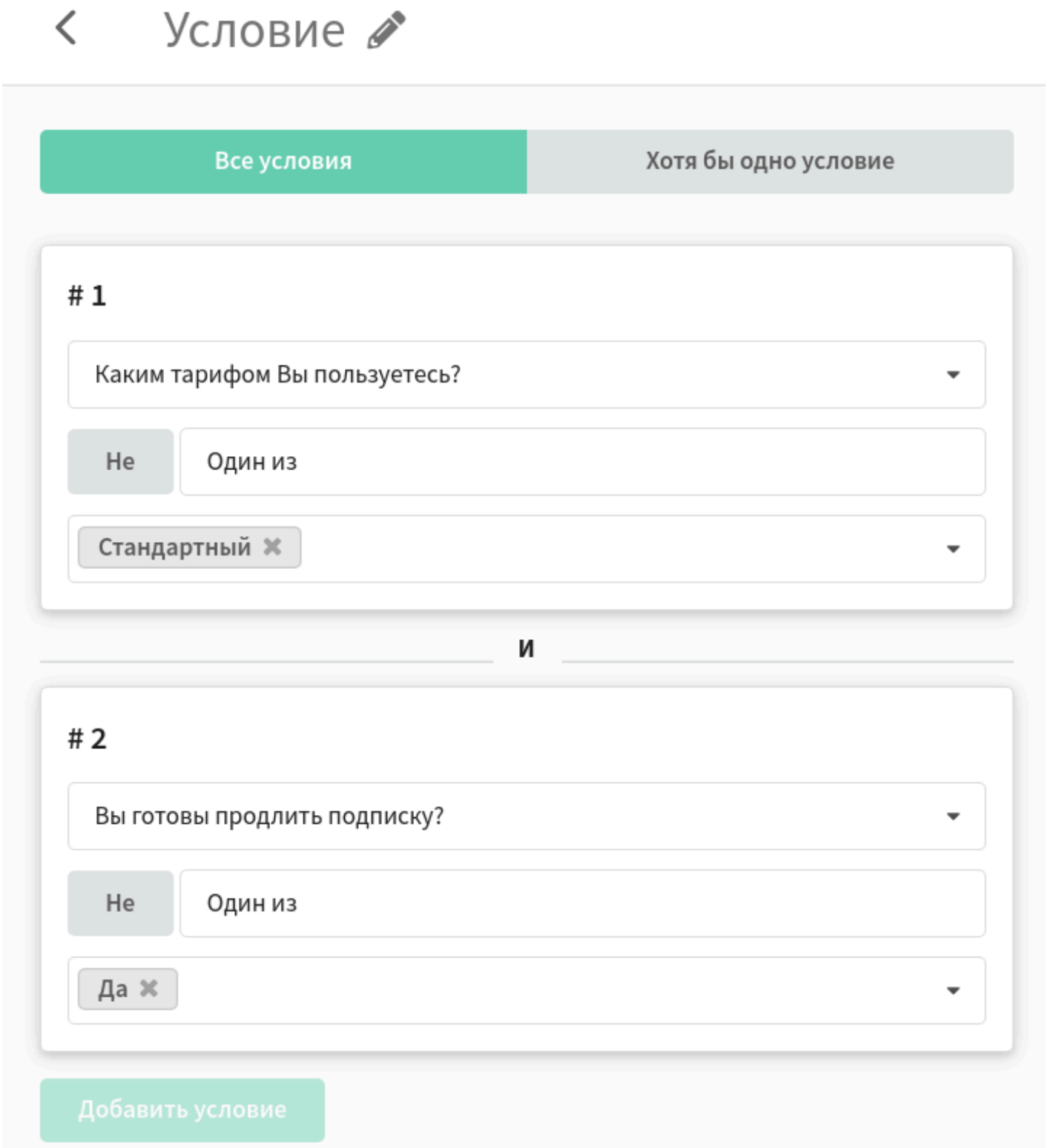
и выберите **“Условие”:**



Кликните на блок условия (ромб). После этого нажмите на кнопку **“Добавить условие”** в появившемся справа окне:



Для одного блока можно добавлять сразу несколько условий. Есть возможность проверять ответ пользователя на соответствие всем условиям (**“Все условия”**) или на соответствие хотя бы одному из них (**“Хотя бы одно условие”**).

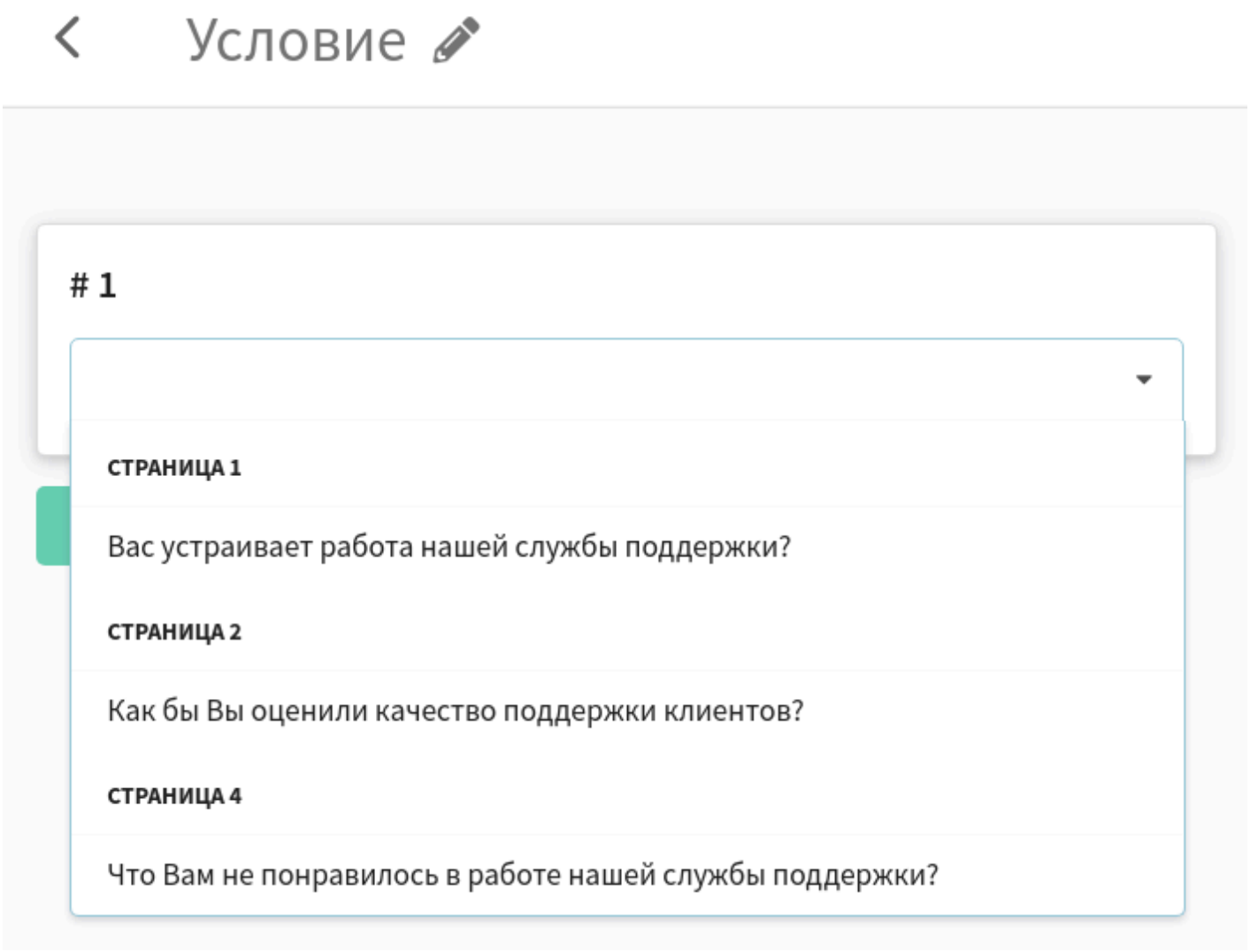


TIP

Для удаления всего блока условия из схемы воспользуйтесь иконкой корзины в правом верхнем углу.

Шаг 4. Настройка условия

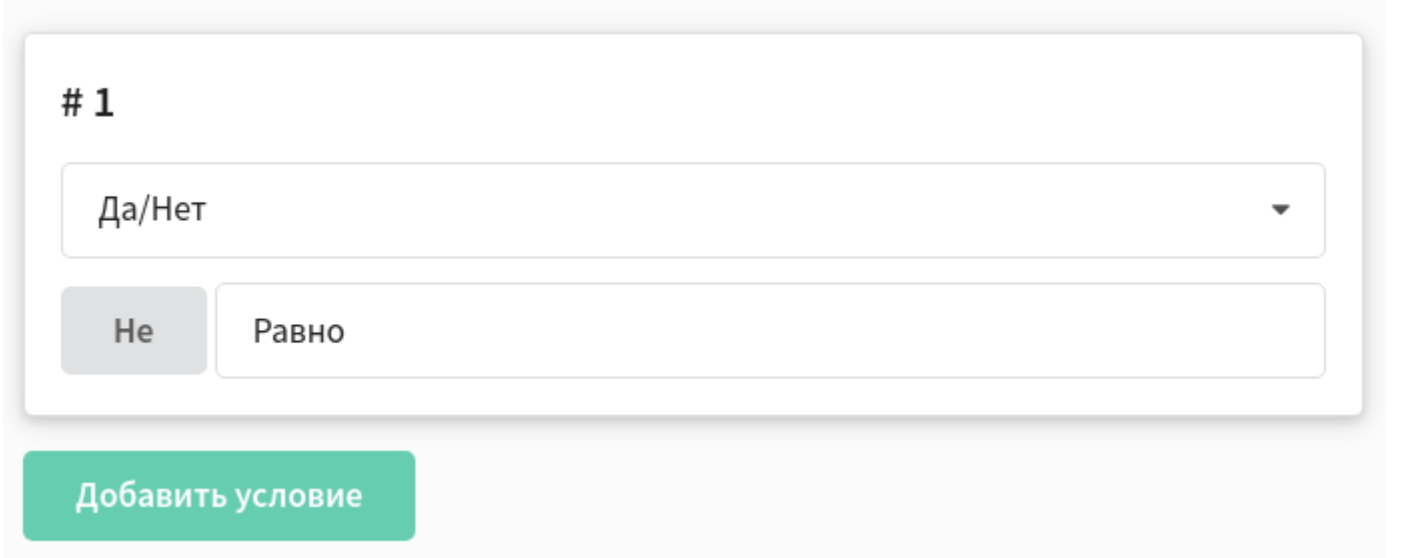
Сначала необходимо выбрать в выпадающем списке элемент на странице, который будет использоваться в качестве условия:



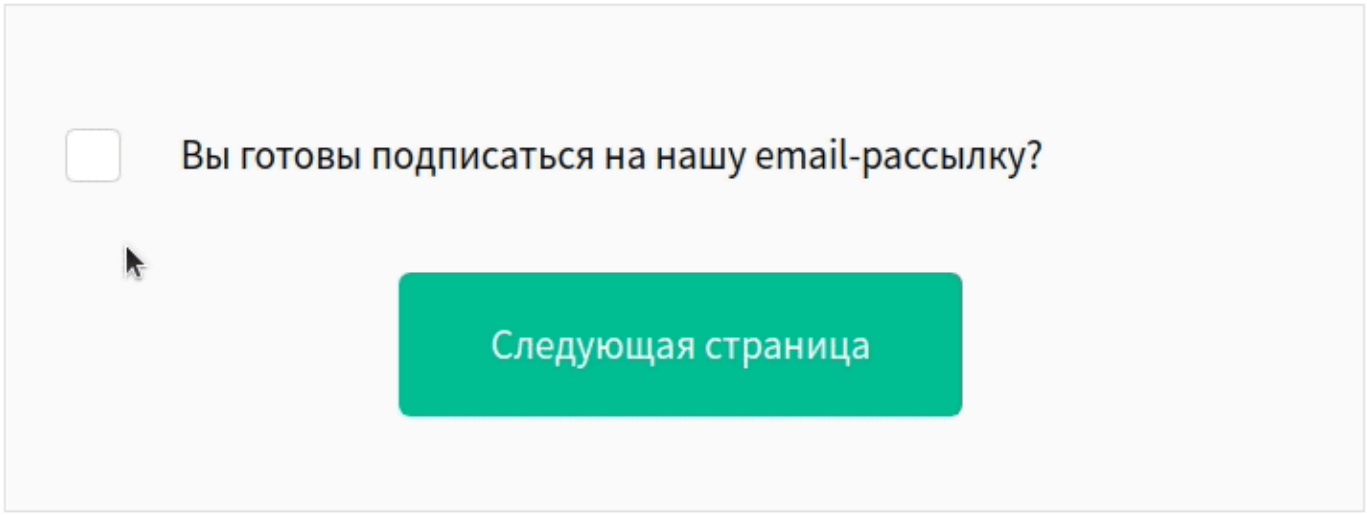
Условие можно задавать следующим элементам:

Да/Нет

“Да/Нет” является самым простым условием. Если пользователь дал утвердительный ответ, он перейдет на одну страницу; если отрицательный — на другую.

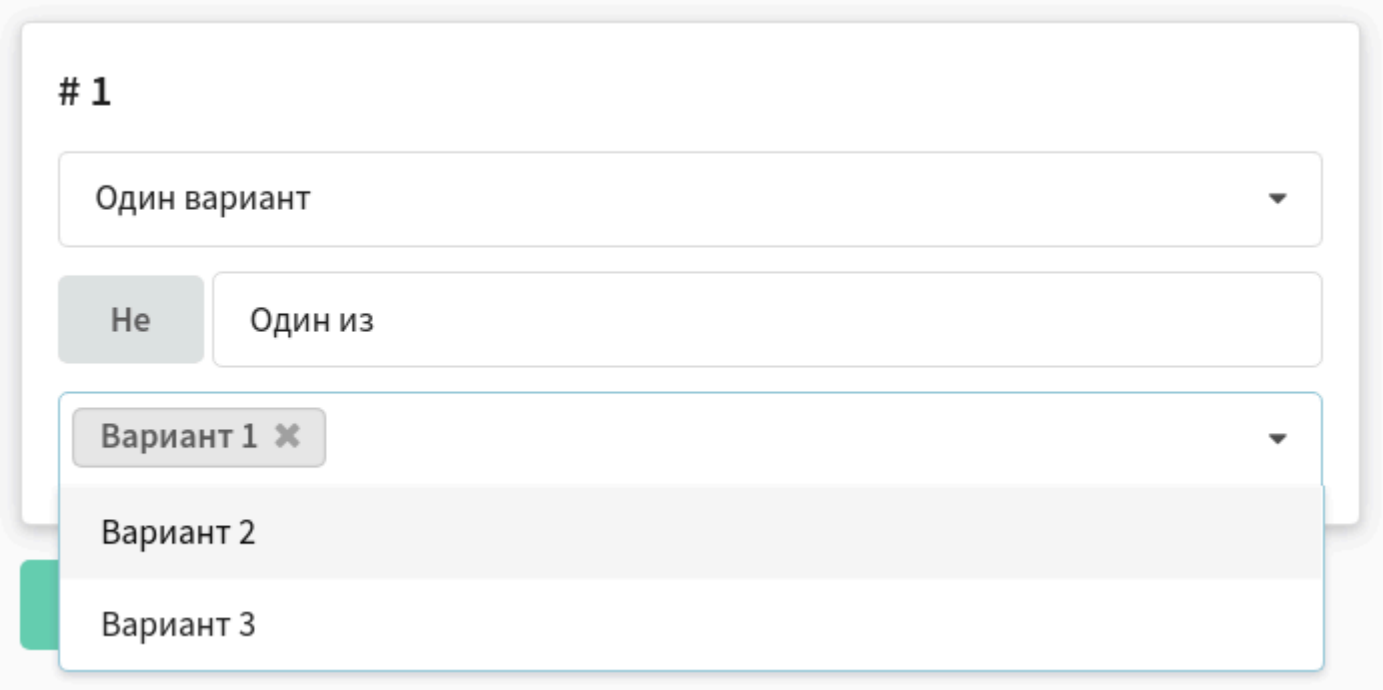


Пример

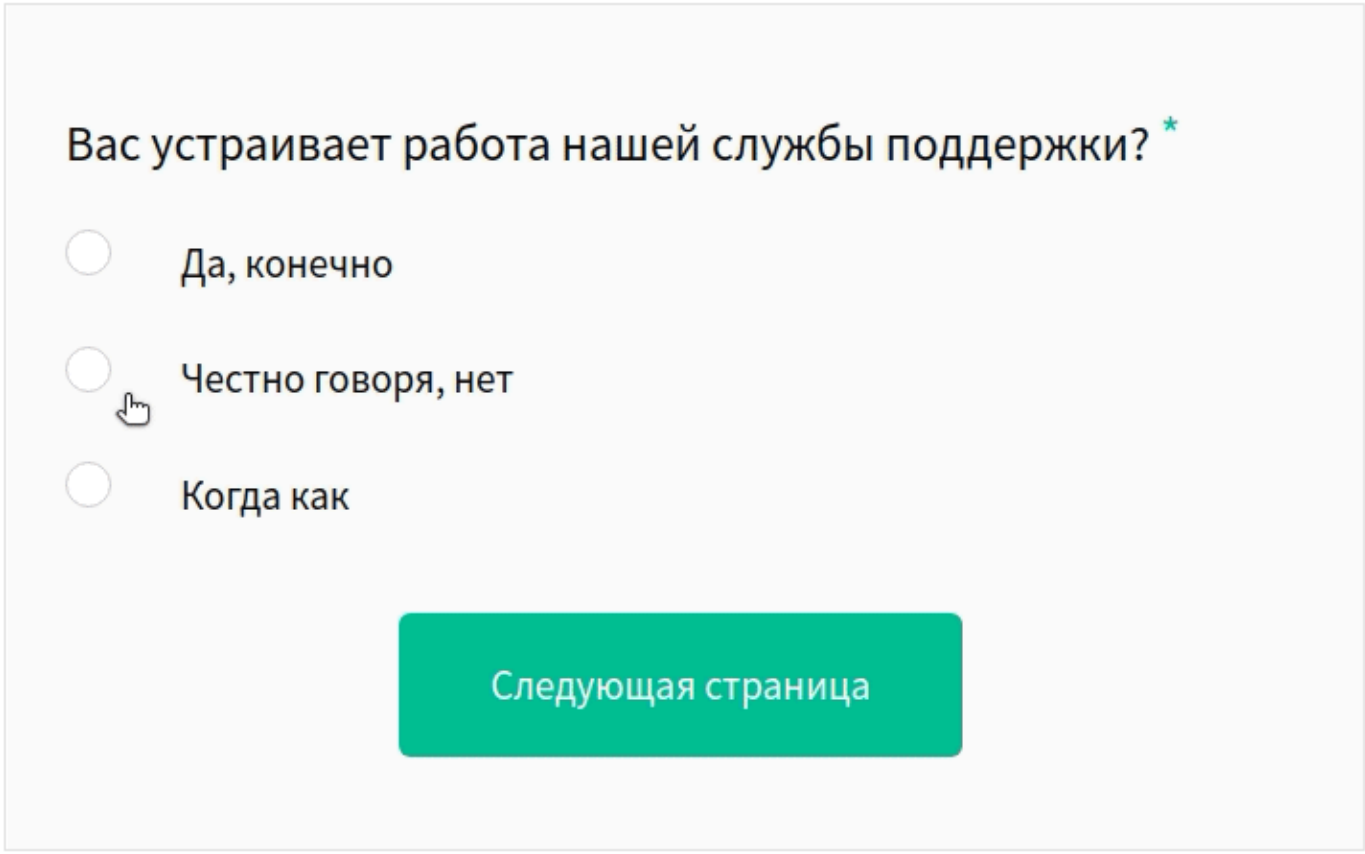


Один вариант

Элемент позволяет перенаправить пользователя на определенную страницу в зависимости от варианта ответа, который он выбрал.



Пример

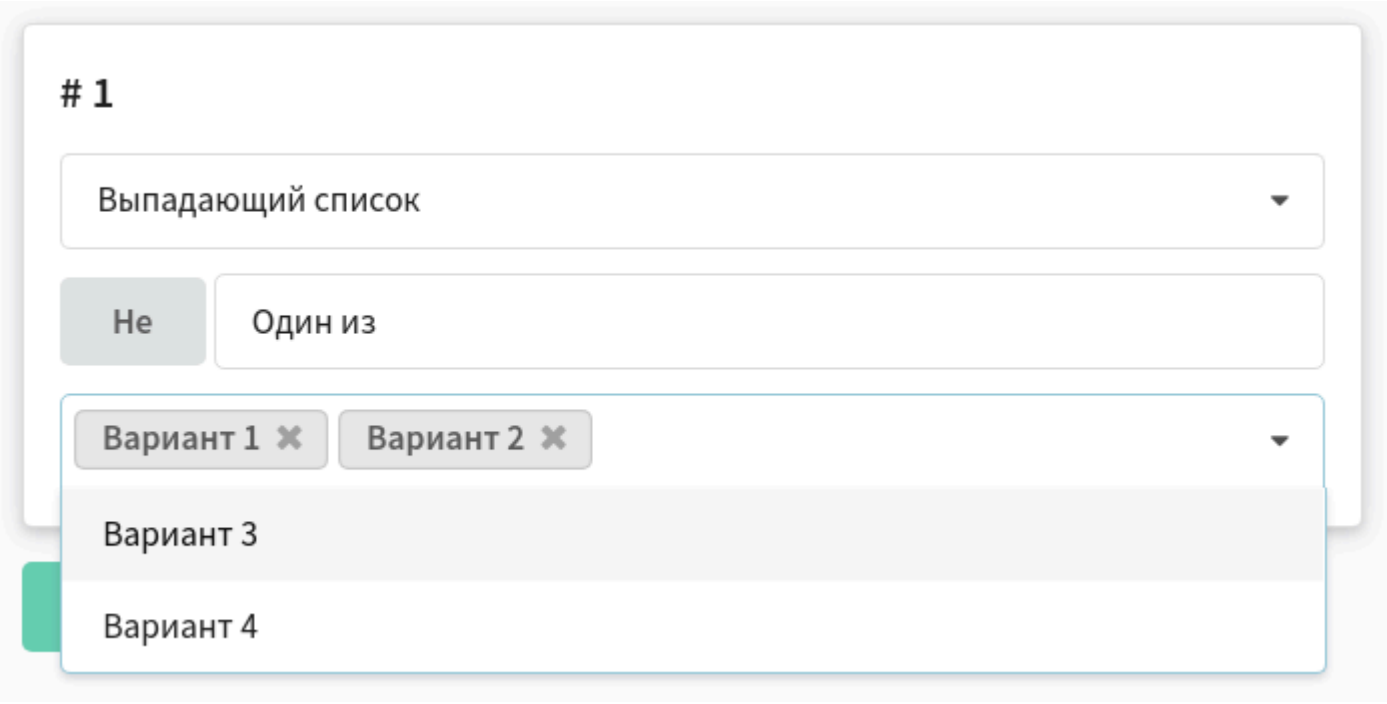


Выпадающий список

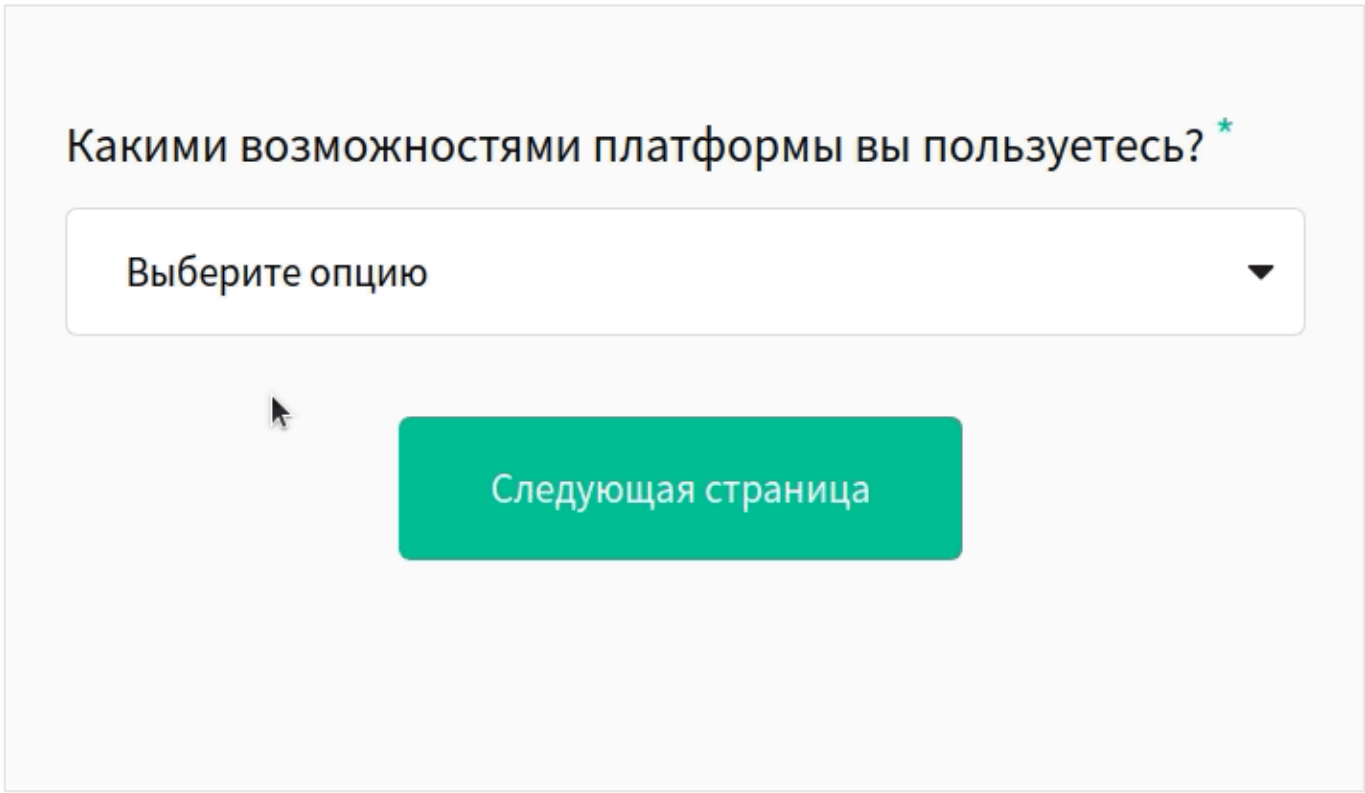
Выпадающий список позволяет пользователю выбрать один или несколько вариантов ответа из предложенных. Вы можете настроить логику так, чтобы при выборе определенных вариантов пользователь переходил на одну страницу формы, а при выборе других — на другую.

Для **выпадающего списка со множественным выбором** доступны следующие условия:

- **Один из** — хотя бы один из вариантов ответа, выбранных пользователем, соответствует указанному значению.
- **Все** — все варианты ответа, выбранные пользователем, соответствуют указанному значению.



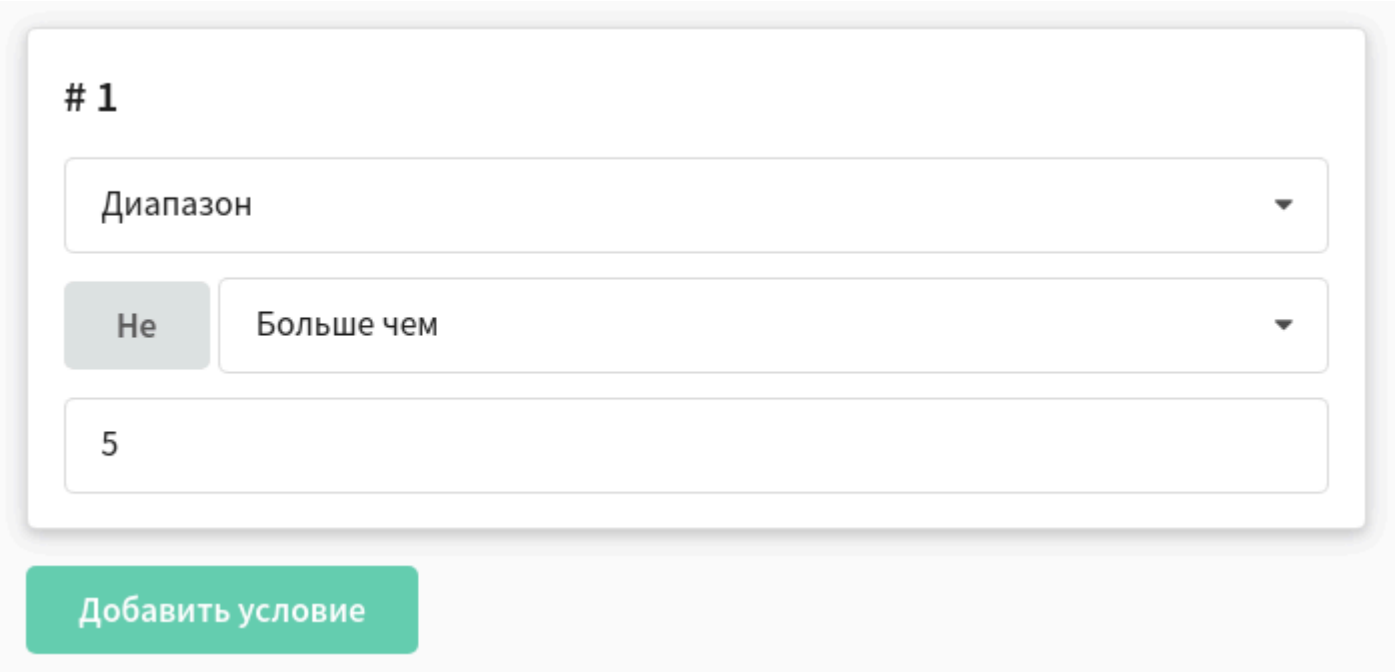
Пример



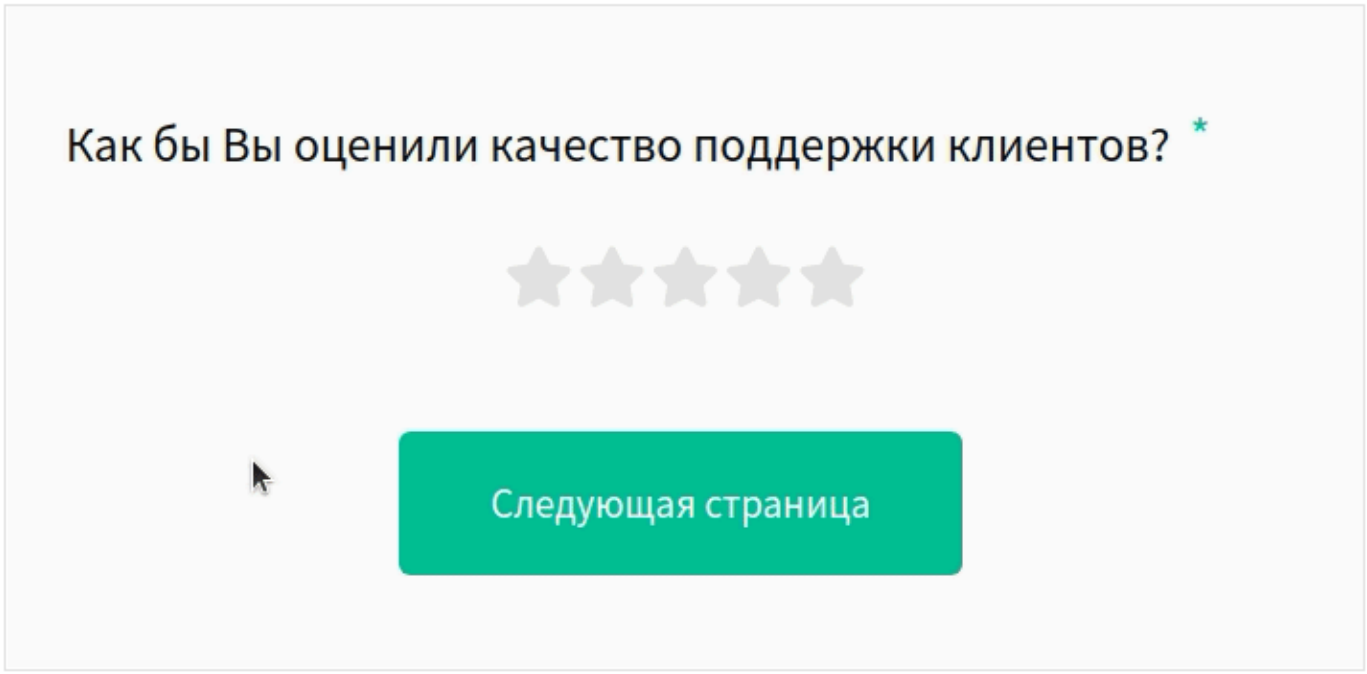
Диапазон

Пользователь перейдет на определенную страницу в зависимости от того, какое значение диапазона отметит. Требуется выбрать:

- **значение**, с которым будет сравниваться ответ пользователя
- **оператор сравнения** (равно, больше чем, меньше чем)



Пример

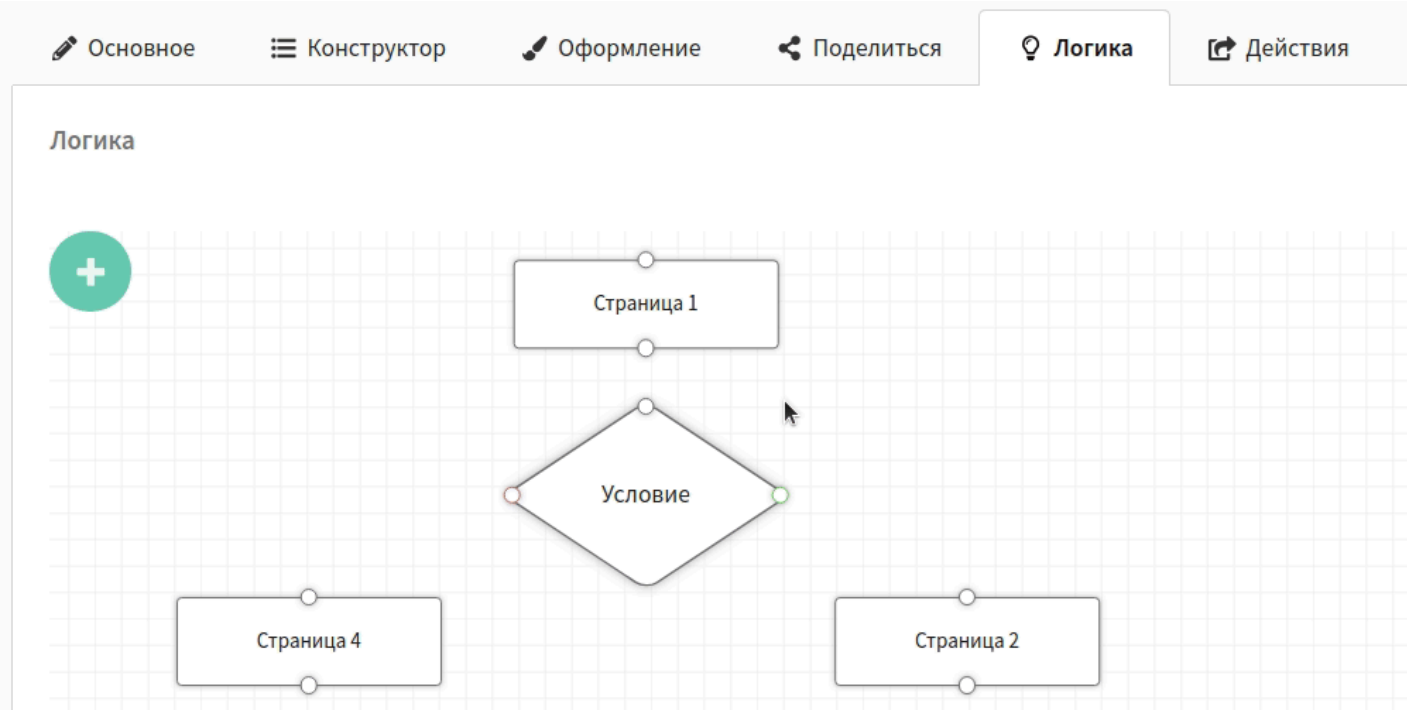


Шаг 5. Настройка связей между блоками

Осталось связать страницы формы в нужной последовательности. Для этого зажмите иконку

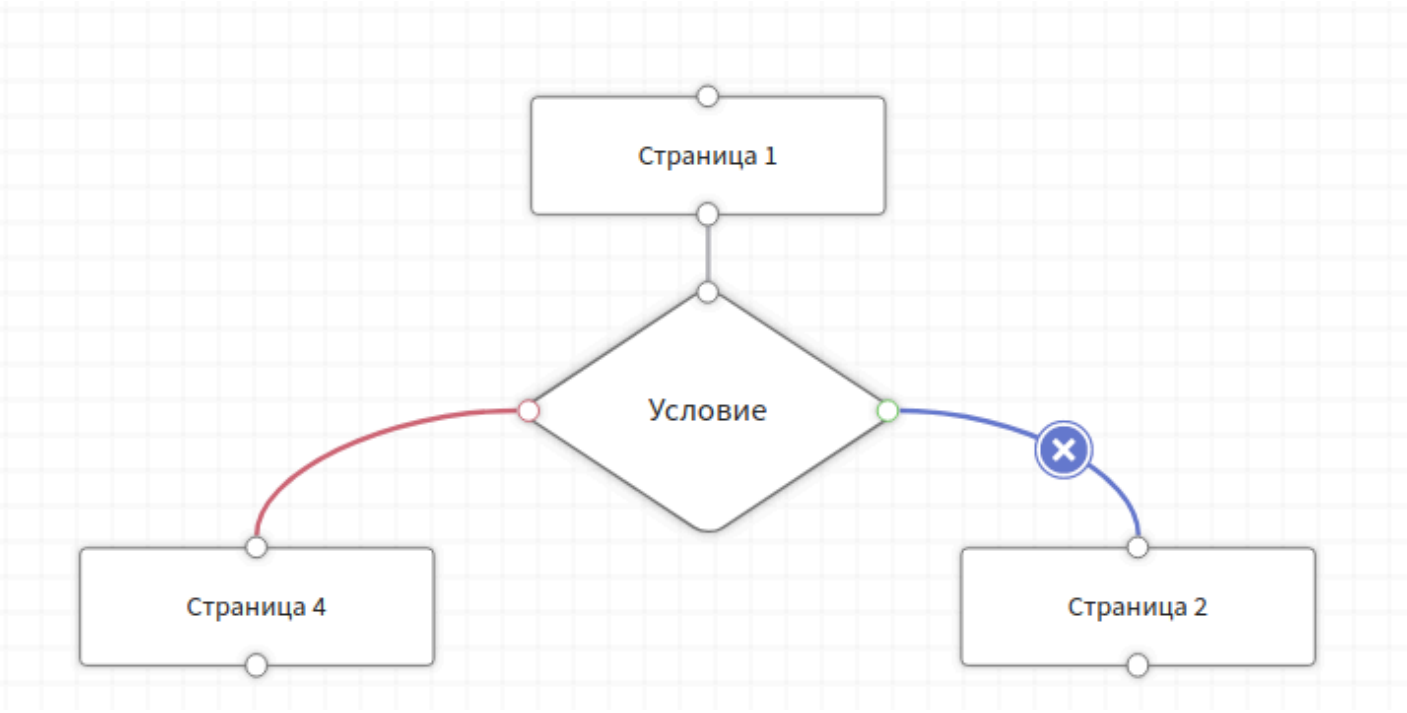
на границе блока и проведите линию до страницы, на которую должен перейти пользователь.

Серая линия связывает переход между страницами без условий. Зеленая — переход, если ответ пользователя соответствует условию. Красная — если условие не соблюдено.



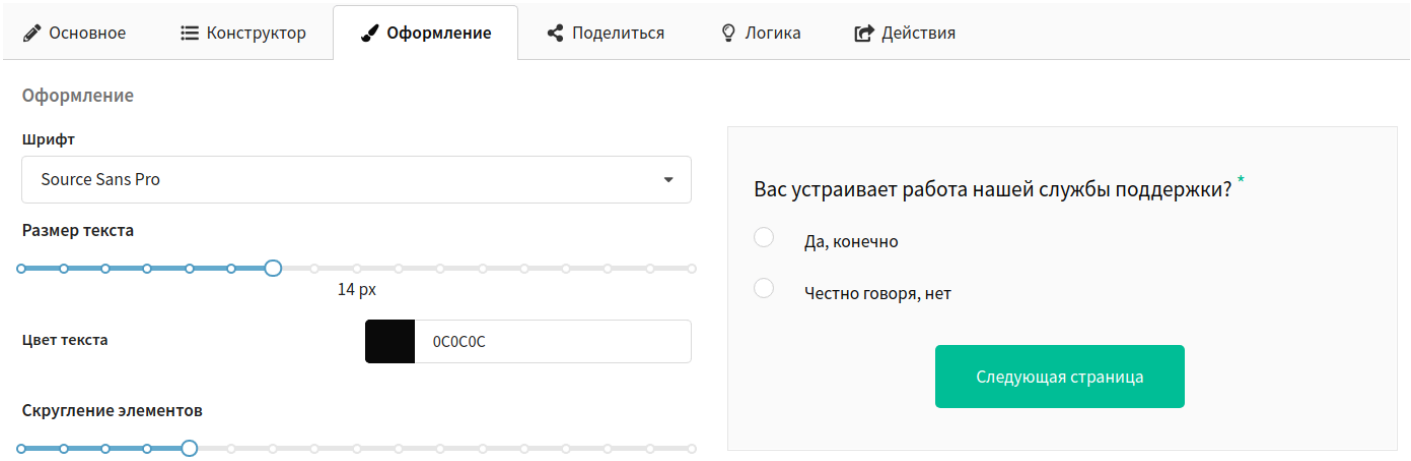
Чтобы удалить связь, нажмите на линию, а затем на иконку

:



Шаг 6. Предпросмотр формы

Проверить, правильно ли выстроена логика формы, можно на вкладке **“Оформление”**:



Если всё настроено верно, сохраните форму. После этого ее можно опубликовать и использовать на сайте или отправить клиенту ссылку на страницу формы в письме.

NPS-тестирование

Индекс Net Promoter Score (индекс потребительской лояльности) — метрика, показывающая приверженность клиентов компании или товару. Иными словами, индекс NPS отражает готовность покупателей рекомендовать ваш продукт или услуги. Этот показатель является одним из главных индексов измерения клиентской лояльности. NPS позволяет проанализировать сильные и слабые стороны продукта и скорректировать маркетинговую стратегию.

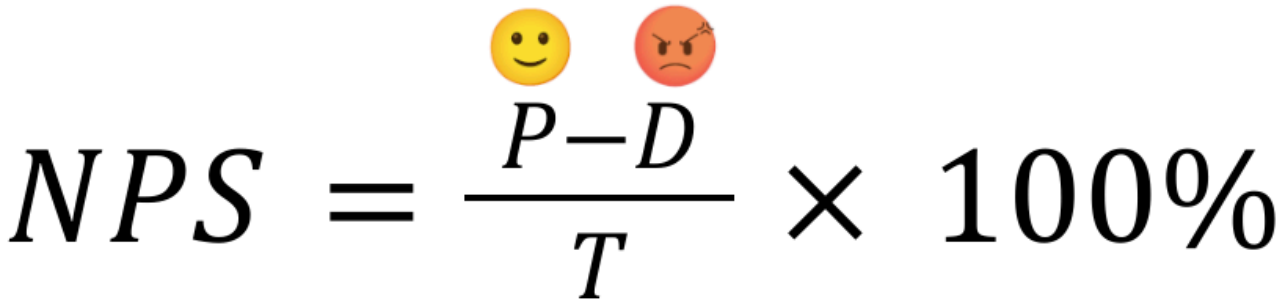
Обычно NPS-тестирование представляет собой форму с вопросом **“Насколько вы готовы рекомендовать нас по шкале от 1 до 10?”** и вариантами ответа. В зависимости от ответа различают несколько типов респондентов:

- **Промоутеры** — лояльные клиенты, готовые оставлять положительные отзывы и рекомендовать продукт
- **Нейтралы** — нейтральные клиенты, которые пользуются вашим продуктом сейчас, но прекратят, если найдут предложение лучше
- **Критики** — клиенты, которые получили негативный опыт взаимодействия с вами и теперь готовы оставлять негативные отзывы о вашей компании.

Приемлемыми считаются значения NPS от **30% и выше**. Значение между 0% и 30% свидетельствует о наличии некоторых проблем с продуктом. Отрицательные значения индекса говорят **о серьезных проблемах**, которые требуют немедленного вмешательства.

Расчет NPS

В математическом смысле NPS – это **разница между процентными показателями промоутеров и критиков**. Выражение можно представить так:



где **P** – промоутеры, **D** – критики, **T** – общее число респондентов.

При аналитике полученных данных платформа может использовать либо **средний**, либо **агрегированный** индекс NPS. Средний NPS — это среднее арифметическое всех индексов NPS внутри формы. Агрегированный NPS — общий индекс NPS, полученный на основе ответов **всех** респондентов в каждом вопросе. Их можно вычислить по формулам:

$$NPS_{\text{средн}} = \frac{NPS_1 + NPS_2 + NPS_3 + \ldots + NPS_n}{n}$$

$$NPS_{\text{агр}} = \frac{\Sigma(P_i - D_i)}{\Sigma T_i} \times 100\%$$

Рассмотрим пример нахождения индекса NPS. Допустим, мы провели 3 тестирования со следующими результатами:

- Тест #1 – 6 критиков, 9 нейтралов и 27 промоутеров, всего 42 респондента
- Тест #2 – 10 критиков, 15 нейтралов и 25 промоутеров, всего 50 респондентов
- Тест #3 – 21 критик, 35 нейтралов и 14 промоутеров, всего 70 респондентов

Расчет NPS для этих тестов будет выглядеть так:

🙂 😡

$$NPS_1 = \frac{27-6}{42} \times 100\% = 50\%$$

🙂 😡

$$NPS_2 = \frac{25-10}{50} \times 100\% = 30\%$$

🙂 😡

$$NPS_3 = \frac{14-21}{70} \times 100\% = -10\%$$

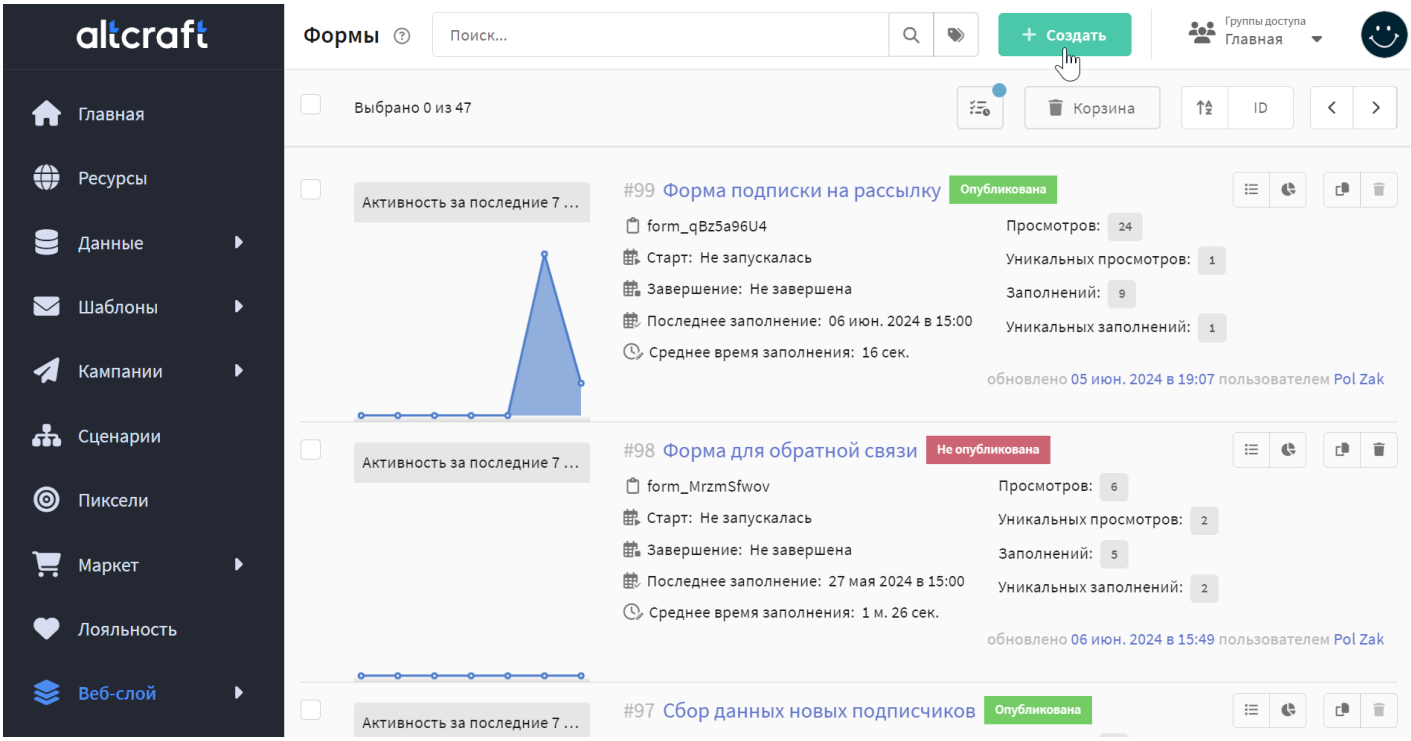
На основе этих данных можно найти средний и агрегированный NPS:

$$NPS_{\text{средн}} = \frac{50\%+30\%-10\%}{3} = 23.3\%$$

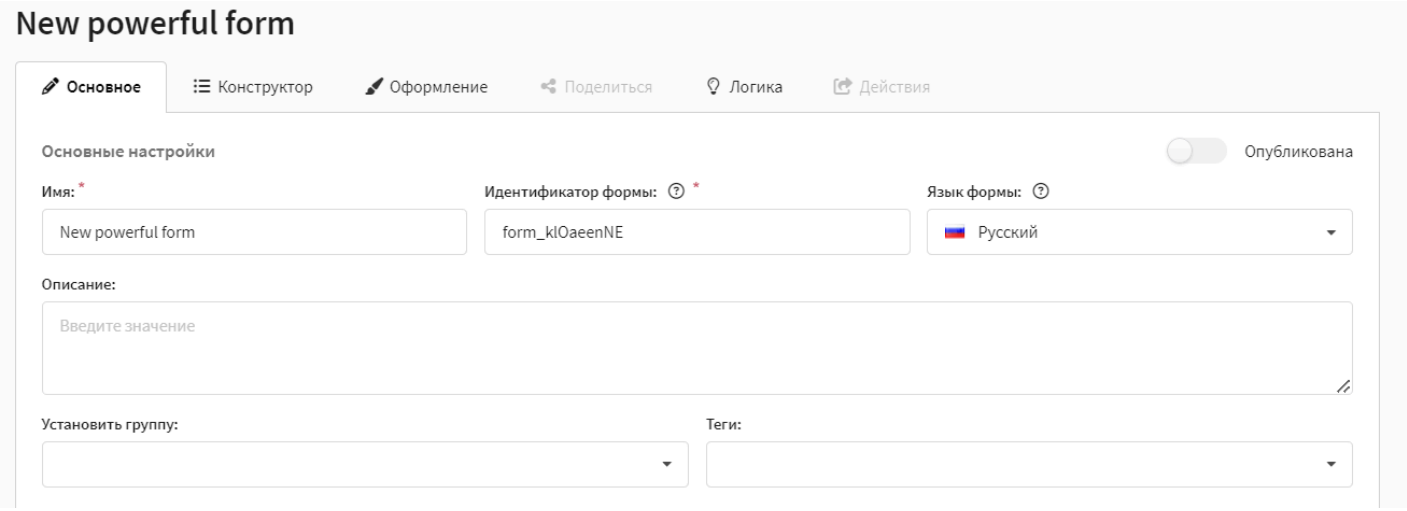
$$NPS_{\text{агр}} = \frac{(27+25+14)-(6+10+21)}{42+50+70} \times 100\% = 17.9\%$$

Создание NPS-тестирования в платформе

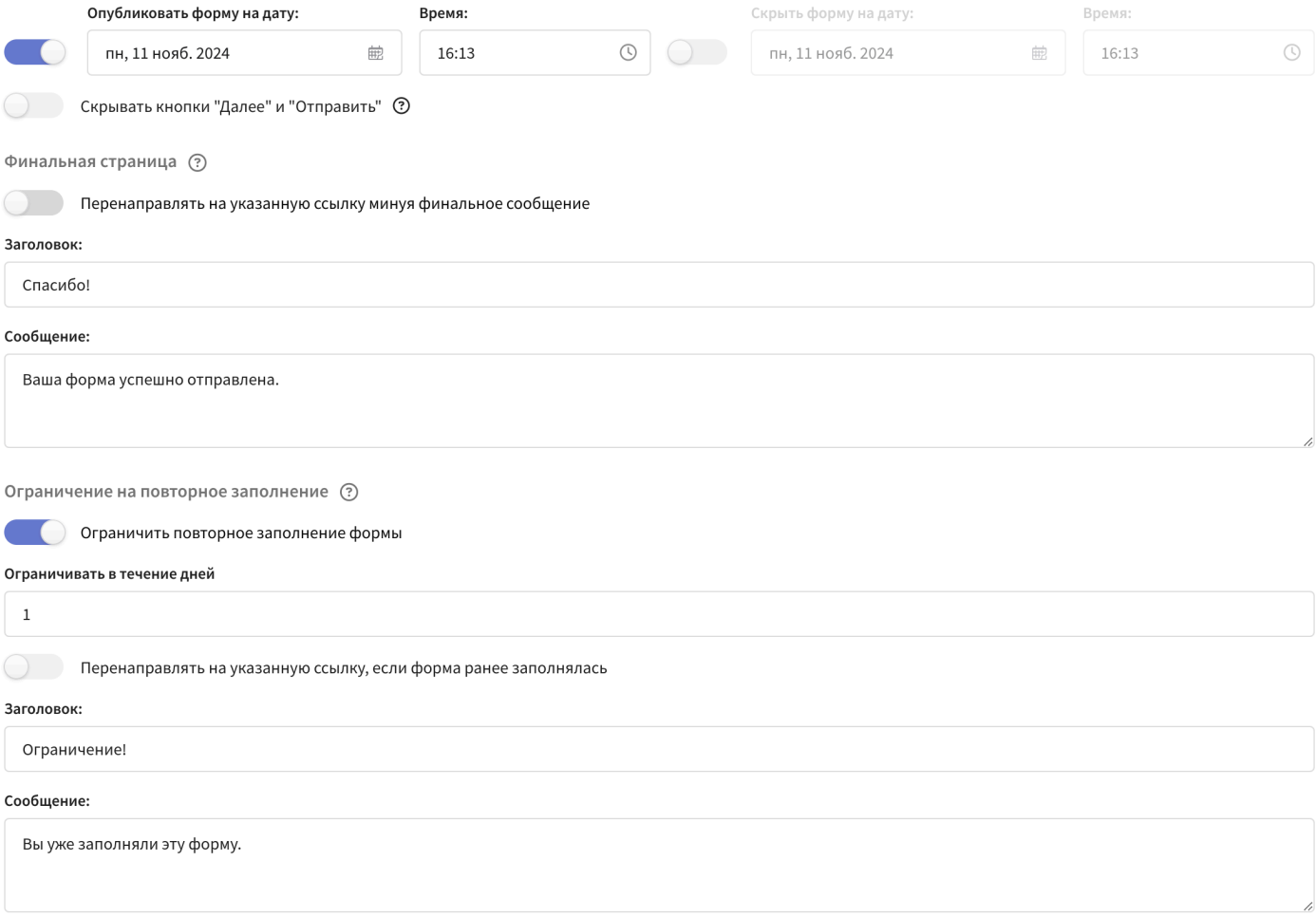
NPS-тестирование реализовано в платформе Altcraft с помощью **веб-форм**. Чтобы создать её, зайдите в раздел **“Веб-слой – Формы”** в главном меню и нажмите на кнопку **“Создать”**:



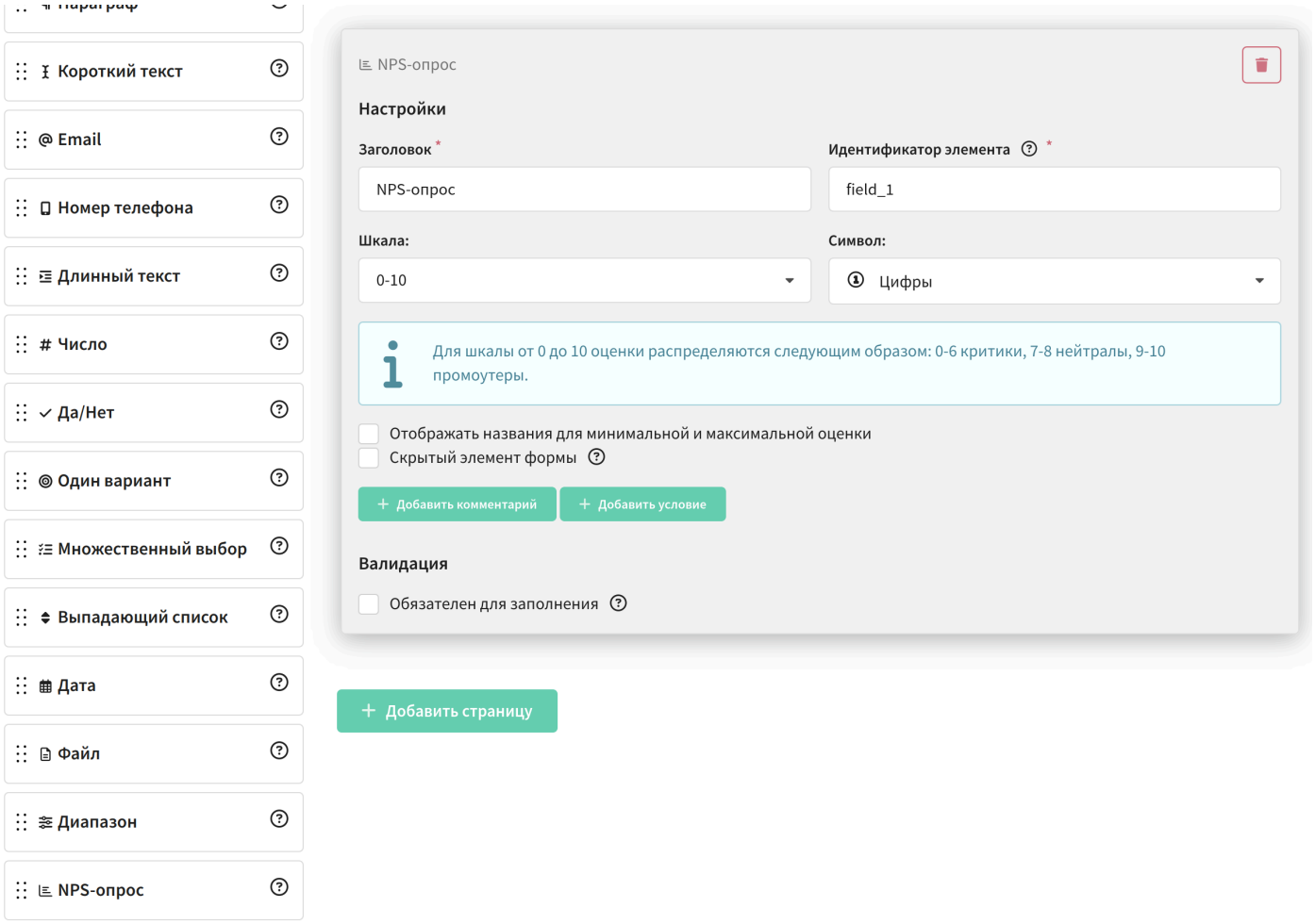
На первой вкладке **“Основные”** доступны главные настройки формы: **имя**, **описание**, **идентификатор**, **язык**. Чтобы форма стала доступна по URL-адресу или как iframe-элемент, необходимо активировать переключатель **“Опубликована”** в правом верхнем углу. Обратите внимание, что пока форма не снята с публикации, её нельзя удалить:



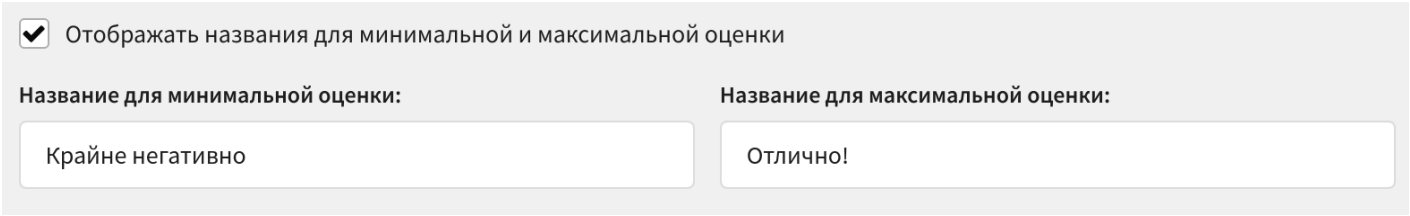
Ниже вы можете настроить такие параметры, как **расписание публикации, финальная страница формы**, а также установить **ограничение на повторное заполнение**:



Чтобы добавить NPS-опрос в форму, перейдите на вкладку **“Конструктор”** и добавьте элемент **NPS-опрос**:



Здесь вы можете изменить заголовок, шкалу и символы для шкалы. Также можно сделать элемент обязательным для заполнения и добавить названия для минимальной и максимальной оценки:



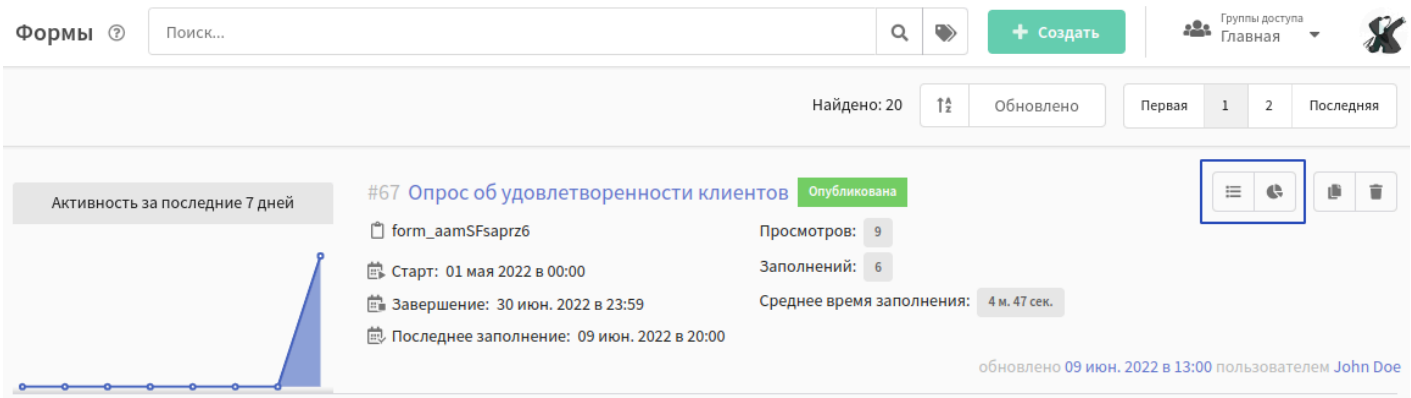
При желании вы можете добавить в форму другие элементы. После внесения изменений **сохраните форму** и переключите параметр **“Опубликована”**. NPS-опрос готов.

TIP

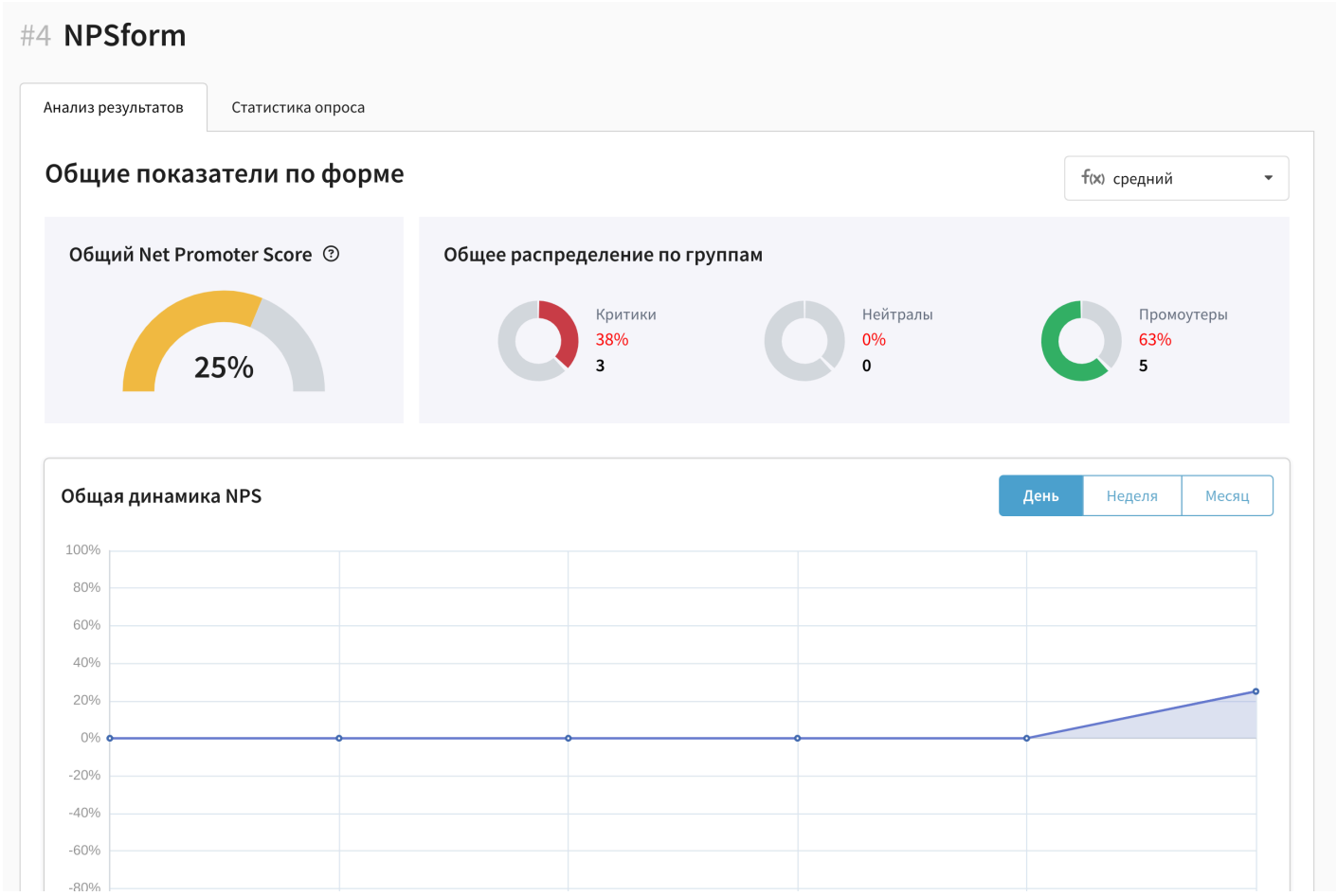
Подробнее прочесть о создании и настройке формы можно на этой странице.

Результаты тестирования

Вам не нужно вручную собирать информацию из форм и высчитывать индекс NPS, результаты тестирования и их интерпретацию можно посмотреть в **статистике по форме**. Для этого откройте раздел **“Веб-слой — Формы”**, найдите нужную вам форму и нажмите на кнопку **“Аналитика”** в меню быстрых действий:



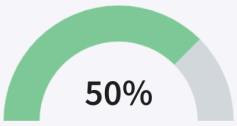
На экране отобразятся общие показатели по форме — **Net Promoter Score, распределение по группам** и **динамика NPS**:



Также вы можете посмотреть информацию по отдельно взятому элементу на каждой странице:

Показатели по форме

Net Promoter Score



Распределение по группам

Критики
25%
100

Нейтралы
0%
0

Промоутеры
75%
300

Распределение оценок

0%	25%	0%	0%	0%	75%
0	100	0	0	0	300

Пиксели

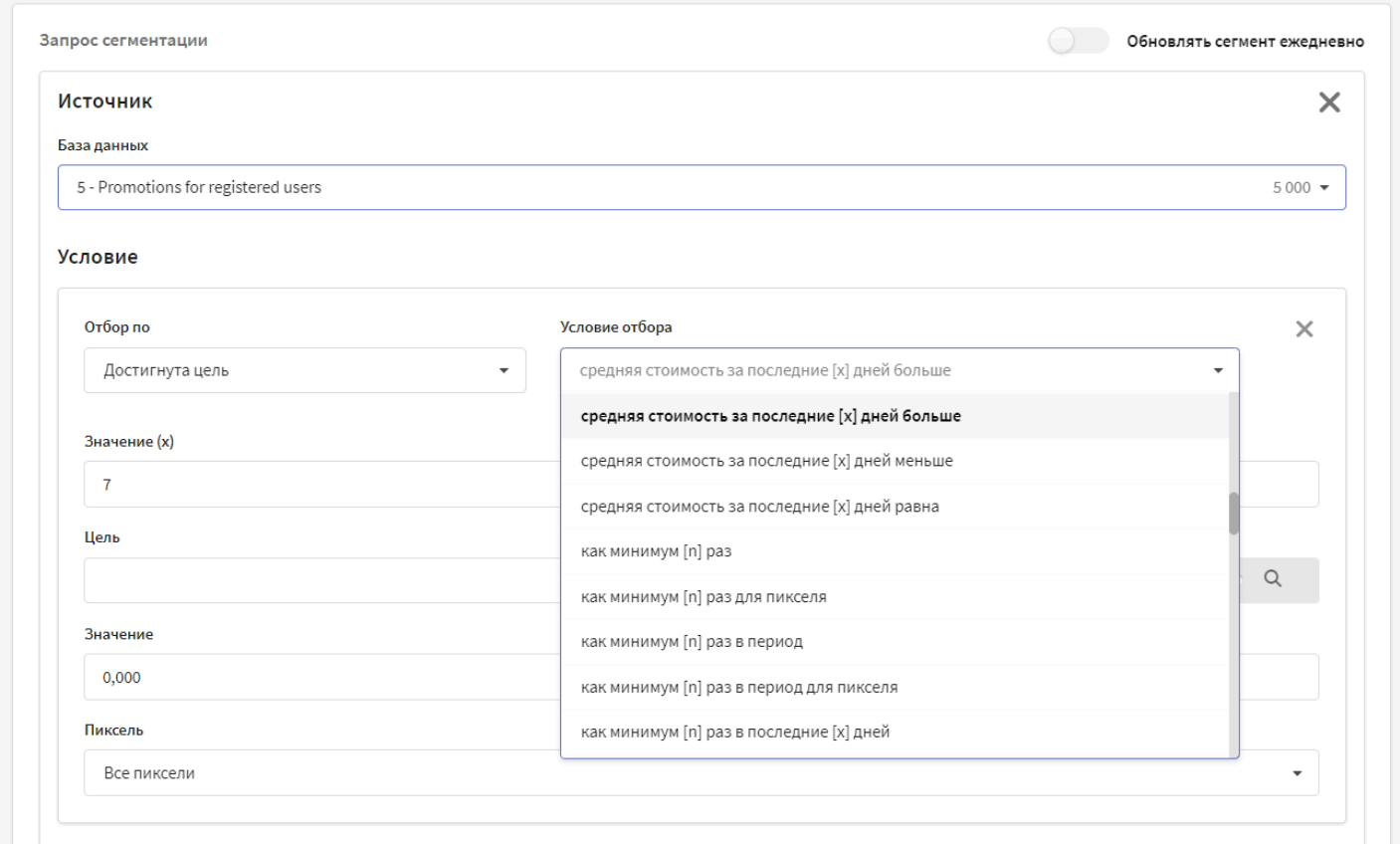
Трекинг-пиксели — инструмент для сбора данных о действиях пользователей на сайте. Пиксель представляет собой небольшой **фрагмент кода**, чаще всего JS-скрипт или прозрачное изображение, незаметное для посетителя. Его можно привязать к загрузке страницы или к любому интерактивному элементу — кнопке, форме или всплывающему окну.

Когда пользователь своими действиями активирует пиксель, на трекинг-сервер платформы отправляется запрос с информацией о событии. С помощью пикселя можно отслеживать:

- географию показов и кликов (страна, город, регион), определяемую по IP;
- браузер и платформу устройства;
- действия подписчика на сайте до и после подписки;
- URL страницы, с которой выполнен переход;
- интересы и предпочтения подписчиков.

Эти данные собираются из заголовков запросов. Дополнительно пиксель может передавать бизнес-информацию через **цели** и их **значения**. Цель — любое важное действие на сайте (например, покупка товара), а значение — связанная с ней числовая величина (например, сумма покупки). Цели и значения задаются вручную в соответствии с вашими задачами.

Собранные данные сохраняются в профиле пользователя в базе, если платформа смогла его идентифицировать. В дальнейшем эту информацию можно использовать для сегментации, построения сценариев и запуска триггерных рассылок. Например, при создании сегмента профилей по событиям можно учитывать добавленные к пикселям цели, адреса сайтов, временные промежутки и количество срабатываний пикселя:



Достижение целей по тем или иным пикселям вы сможете наблюдать в отчёте “Цели”.

TIP

Более подробно использование пикселей рассмотрено в этой статье.

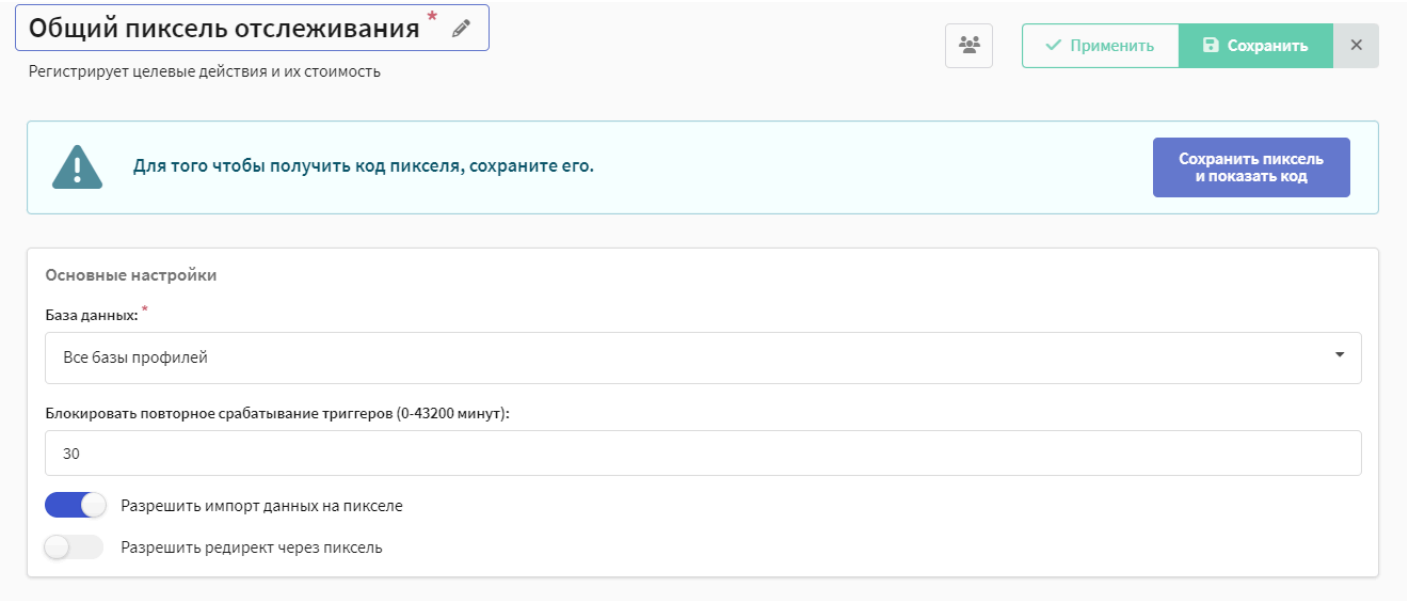
Включение пикселей

	При использованиии Altcraft Platform на собственном оборудовании для работы с пикселями необходимо включить опцию **"Enable Pixel"** в настройках аккаунта, в Панели администратора платформы. Действия выполняет администратор платформы. **NOTE** Кроме того, для использования пикселей необходимо настроить HTTPS (см. руководство для системных администраторов). После включения опции в пользовательском интерфейсе в разделе “Веб-слой” должен появится соответствующий подраздел.

Создание пикселей

Создайте новый пиксель и задайте основные настройки: имя пикселя, описание, группы доступа и связанные базы данных.

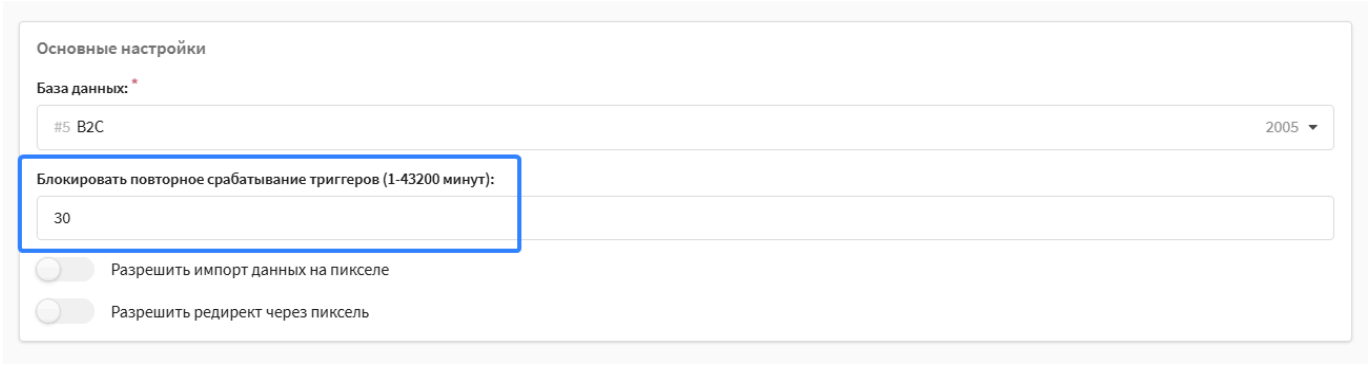
Если нужно блокировать повторное срабатывание триггера, установите нужный временной интервал:



Триггер будет блокироваться для каждой отдельной цели, а не для всего пикселя. Это значит, что вы можете без ограничений по времени использовать один и тот же пиксель, но с разными целями для отслеживания.

TIP

Если триггер не активируется в рассылках или в сценариях, хотя условия триггера выполняются, рекомендуется проверить **интервал срабатывания пикселя**.



INFO

Импорт данных профилей при активации пикселя включает создание новых профилей клиентов (в том числе анонимных) при срабатывании кода отслеживания.

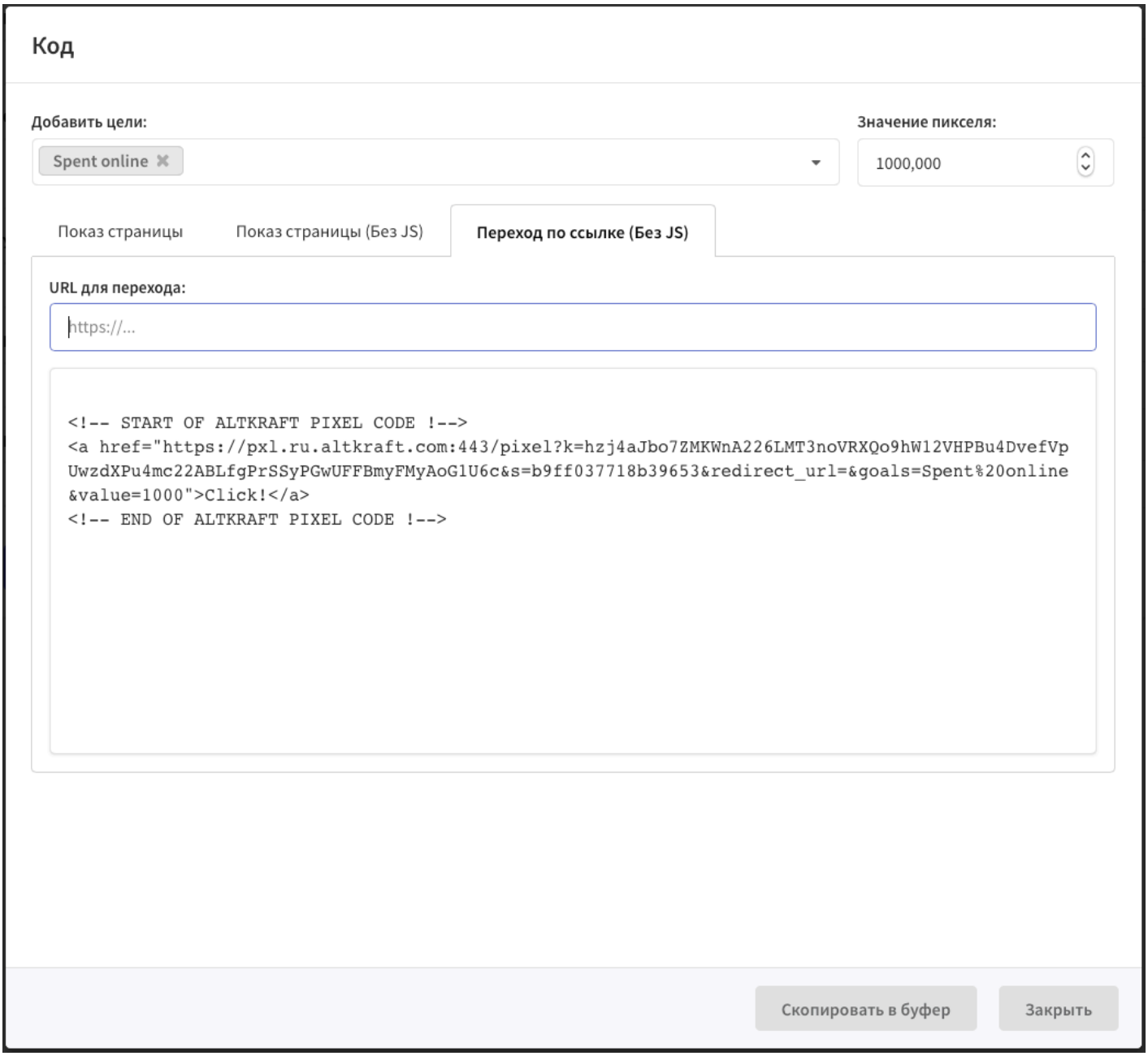
При импорте пиксель может создавать новый профиль или обновлять существующий:

- **HTML пиксели** (изображение и редирект ссылки) используют данные сессии (cookie) для поиска профиля. Если профиль с известными cookie (например, открывал почту с того же устройства) не найден, будет создан новый **анонимный профиль.**
- **JavaScript пиксели** могут использовать как данные сессии, так и непосредственно данные профилей в платформе. Если профиль найден — данные будут обновлены, а если нет — будет добавлен новый профиль с теми данными, которые вы укажете. Настройки импорта для JS кода описаны **ниже в этой статье**.

После сохранения пикселя станет доступен его SID и интерфейс конфигурации кода отслеживания:

Скрипты отслеживания доступны в трёх вариантах:

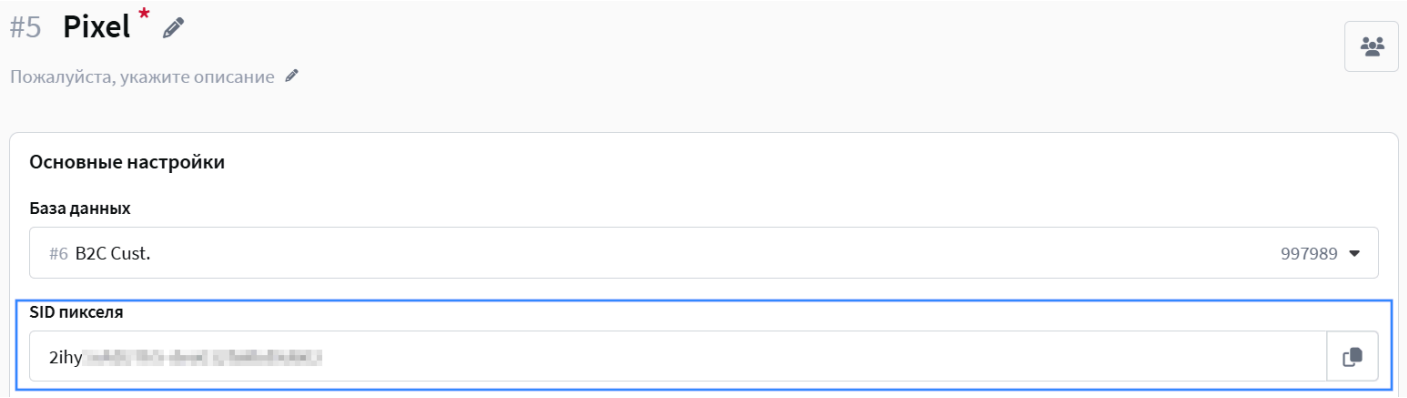
- **Показ страницы** — JavaScript для вставки в код сайта. Поддерживает срабатывание при прокрутке страницы и время ожидания.
- **Показ страницы (Без JS)** — прозрачное изображение, загружаемое с трекинг-сервера. Подойдёт как для сайта, так и для сообщений — не требуется JavaScript.
- **Переход по ссылке (Без JS)** — перенаправление на трекинг для ссылок. Подойдёт для любых ссылок, JavaScript не требуется. Может использоваться в соцсетях и любых внешних ресурсах.



Для каждого скрипта вы можете выбрать несколько целевых действий, а для каждой **цели** — задать своё **значение** (value). Целями могут быть продажа, интерес, заявка, а их значения могут выражаться в баллах или стоимости.

При размещении на странице код зарегистрирует указанные цели, дополнительных действий не требуется. Каждый код, созданный в пикселе, будет ассоциировать цели именно с этим пикселем. Такую логику можно использовать для сравнения конверсии по разным лендингам, сайтам, формам сбора данных.

SID пикселя — служебный строковый идентификатор пикселя. SID **формируется автоматически после сохранения пикселя** и отображается в его редакторе. Он используется платформой для регистрации событий и **не требуется для стандартного отслеживания действий на сайте**.



SID может понадобиться при настройке мобильных событий через mSDK.

Настройка отслеживания действий на сайте

Конфигурация кода отслеживания

Скрипт добавляется **в код страниц сайта**, на которых вы хотите отслеживать действия клиентов. При посещении страницы пиксель будет регистрировать цели, которые вы можете выбрать в меню “**Добавить цели**”. Для всех целей вы также можете передать **значение** или стоимость действия.

CAUTION

В названиях целей можно использовать:

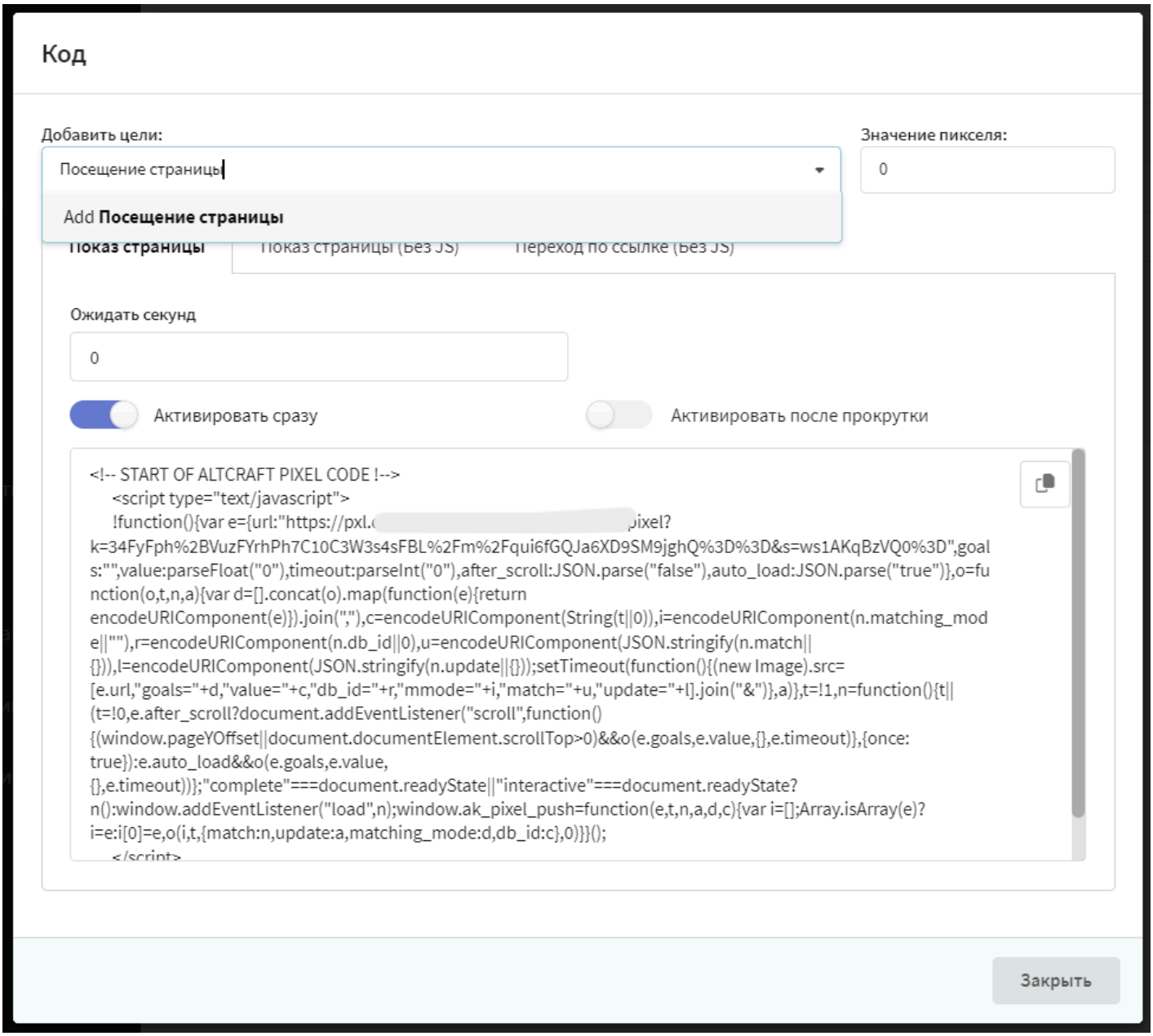
- буквы латинского и русского алфавита (любой регистр)
- пробелы и табуляцию
- символы валют
- `! # $ % & * + - . , : ; '`
- `= ? @ ^ _ ( ) [ ] { } / \ |`

Максимальная длина названия — 128 символов.

Ниже вы можете настроить **время ожидания** или включить опцию “**Активировать после прокрутки**” — чтобы исключить из выборки случайных посетителей страницы.

TIP

После того как вы сконфигурировали пиксель, передайте его код из вкладки **"Показ страницы"** разработчикам вашего сайта вместе с информацией из этого раздела.



Вызов API регистрации целей

На страницах сайта, где будет размещён **код пикселя**, можно отслеживать любые действия клиента по **событиям браузера**. Например, наведение курсора мыши, открытие выпадающего меню или формы. При наступлении события можно передавать в платформу любые цели и значения. В значение можно передавать стоимость действия или затраты по нему.

На страницах с размещённым JavaScript кодом отслеживания, используйте метод `ak_pixel_push()`:

Для него доступны следующие аргументы:

- **tags** — название целевого действия. Может содержать строку или массив строк.
- **value** — значение или стоимость целевого действия. Может содержать положительное или отрицательное число с плавающей точкой.

Описание	Метод
Простой вызов	`ak_pixel_push()`
Вызов с действием	`ak_pixel_push("click")`

Описание	Метод
Вызов с действием, с некоторым значением	`ak_pixel_push("click", 100)`
Вызов с несколькими действиями, с отрицательным значением. Обоим целям будет назначено одинаковое значение.	`ak_pixel_push(["click", "cars"], -100.40)`

Импорт профилей клиентов при активации пикселя

Если для пикселя разрешён импорт данных профилей, вы можете добавить параметры для поиска и импорта **ak_pixel_push(tags, value, match, update, matching_mode)**:

- **match** — используется для поиска профиля. Представляет собой объект с одним или двумя параметрами: `"поле для поиска": "значение поля"` и `"db_id": "ID базы данных"` — только если база данных **не указана** в настройках пикселя.

> **TIP**
>
> **db_id** можно не указывать, если База профилей выбрана в настройках пикселя.

- **update** — может содержать любые данные профиля, в том числе подписки. Структура объекта и названия полей данных профиля описаны на этой странице.
- **matching_mode** — может содержать любой тип матчинга.

> **TIP**
>
> **matching_mode** можно не указывать при поиске по кастомному полю.

Пример регистрации цели с импортом профиля

Ниже представлены примеры регистрации цели с различным типом матчинга:

email

```
ak_pixel_push(
  "click",
  100,
  {
    db_id: 1,
    email: "user@mail.ru"
  },
    {
        "_fname": "John",
        "_lname": "Doe",
        "subscriptions": [
            {
                "resource_id": 1,
                "channel": "email",
                "email": "john.doe@example.com"
            }
        ]
    }
);
```

email_profile

```
ak_pixel_push(
  "click",
  100,
  {
    db_id: 1,
    email: "user@mail.ru"
  },
    {
        "_fname": "John",
        "_lname": "Doe",
        "subscriptions": [
            {
                "resource_id": 1,
                "channel": "email",
                "email": "john.doe@example.com"
            }
        ]
    },
  "email_profile"
);
```

custom

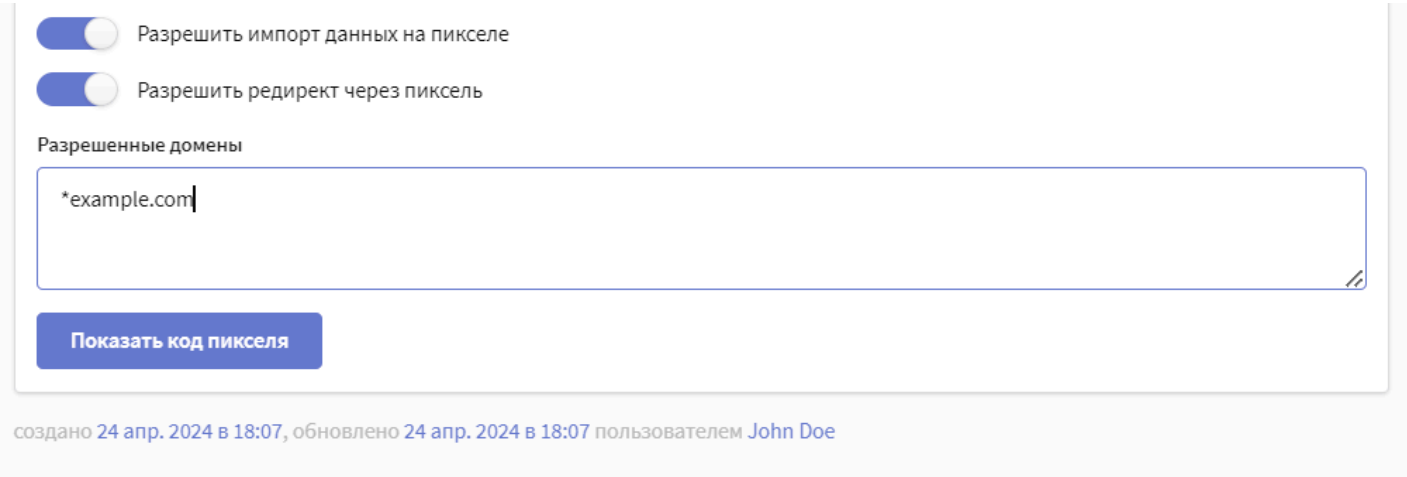
```
ak_pixel_push(
    "click",
    100,
    {
      db_id: 1,
      field_name: "client_id",
      field_value: 1,
    },
    {
      _fname: "John",
      _lname: "Doe",
      subscriptions: [
        {
          resource_id: 1,
          channel: "email",
          email: "john.doe@example.com",
        },
      ],
    },
  "custom"
);
```

phone

```
ak_pixel_push(
    "click",
    100,
    {
      db_id: 1,
      phone: "+79000000000"
    },
      {
          "_fname": "John",
          "_lname": "Doe",
          "subscriptions": [
              {
                  "resource_id": 1,
                  "channel": "email",
                  "email": "john.doe@example.com"
              }
          ]
      },
    "phone"
);
```

Настройка допустимых для редиректа доменов

Если вы разрешите **редирект через пиксель**, то в тот момент, когда пользователь попадет на ваш пиксель, его перебросит на указанный вами URL:



Используя редирект-ссылку, злоумышленник может отправить фишинговое письмо с подстановкой в redirect_url – вредоносного сайта. Для недопущения подобной уязвимости, пиксель в платформе по умолчанию работает в режиме отключенного редиректа.

Когда опция активирована, появится текстовое поле, в котором можно ввести список доменов, редирект на которые разрешен.

После этого в сообщениях, использующих данный пиксель, будет доступен редирект только на домены, указанные в списке. Перенаправление на другие домены будет запрещено.

Редирект доступен только если вы внедряете пиксель на ваш сайт таким способом как **“Переход по ссылке”.**

Настройка регистрации целей Altcraft пикселя с помощью Google Tag Manager (GTM)

Для удобства размещения пикселя Altcraft на страницах сайта можно использовать контейнеры Google Tag Manager (если установлен на сайте).

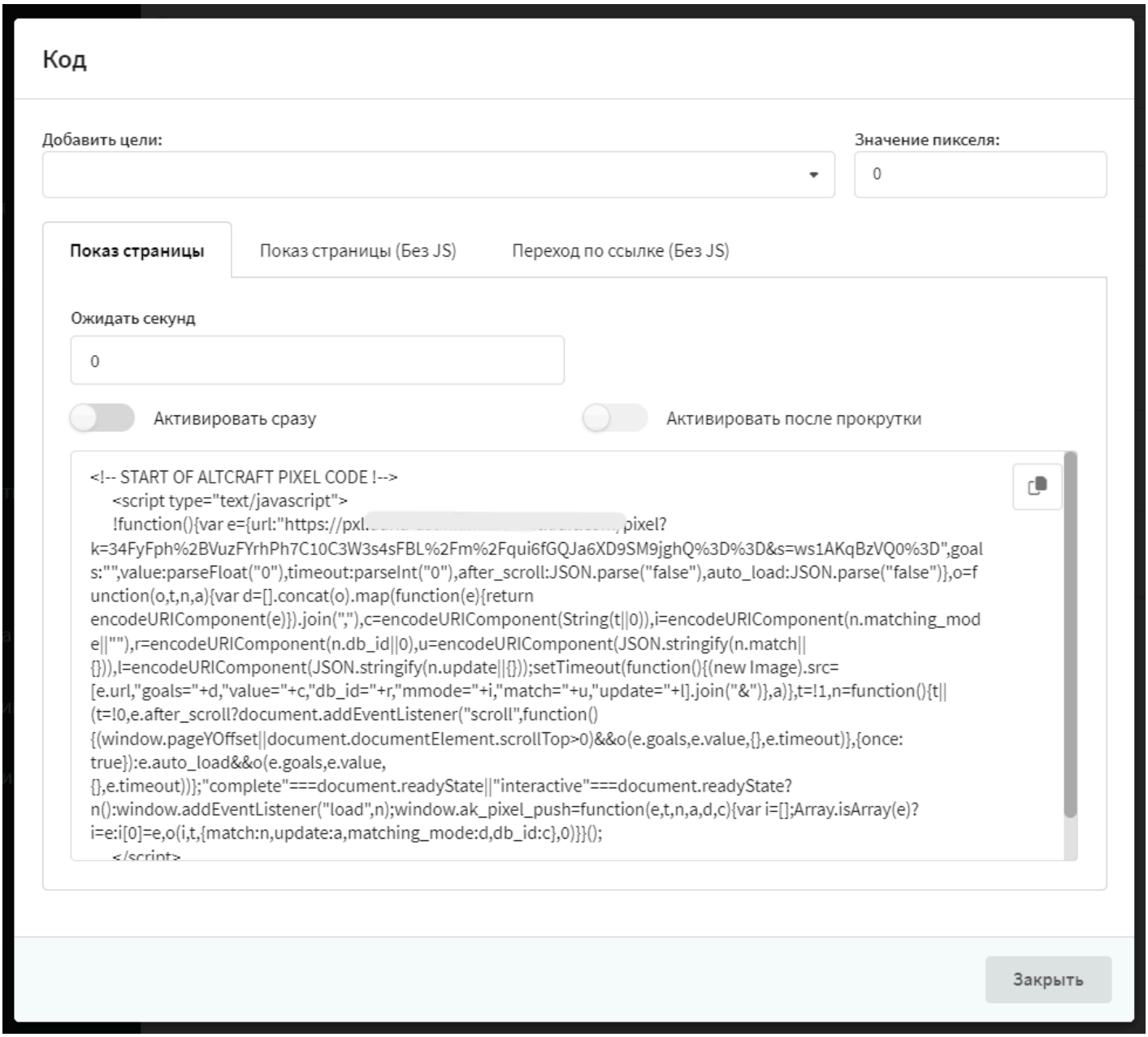
Для этого:

1. Создайте новый пиксель и задайте основные настройки: имя пикселя, описание, группы доступа и связанные базы данных.

2. После сохранения пикселя будет доступен интерфейс конфигурации кода отслеживания. Скопируйте код пикселя с вкладки “**Показ страницы**”.

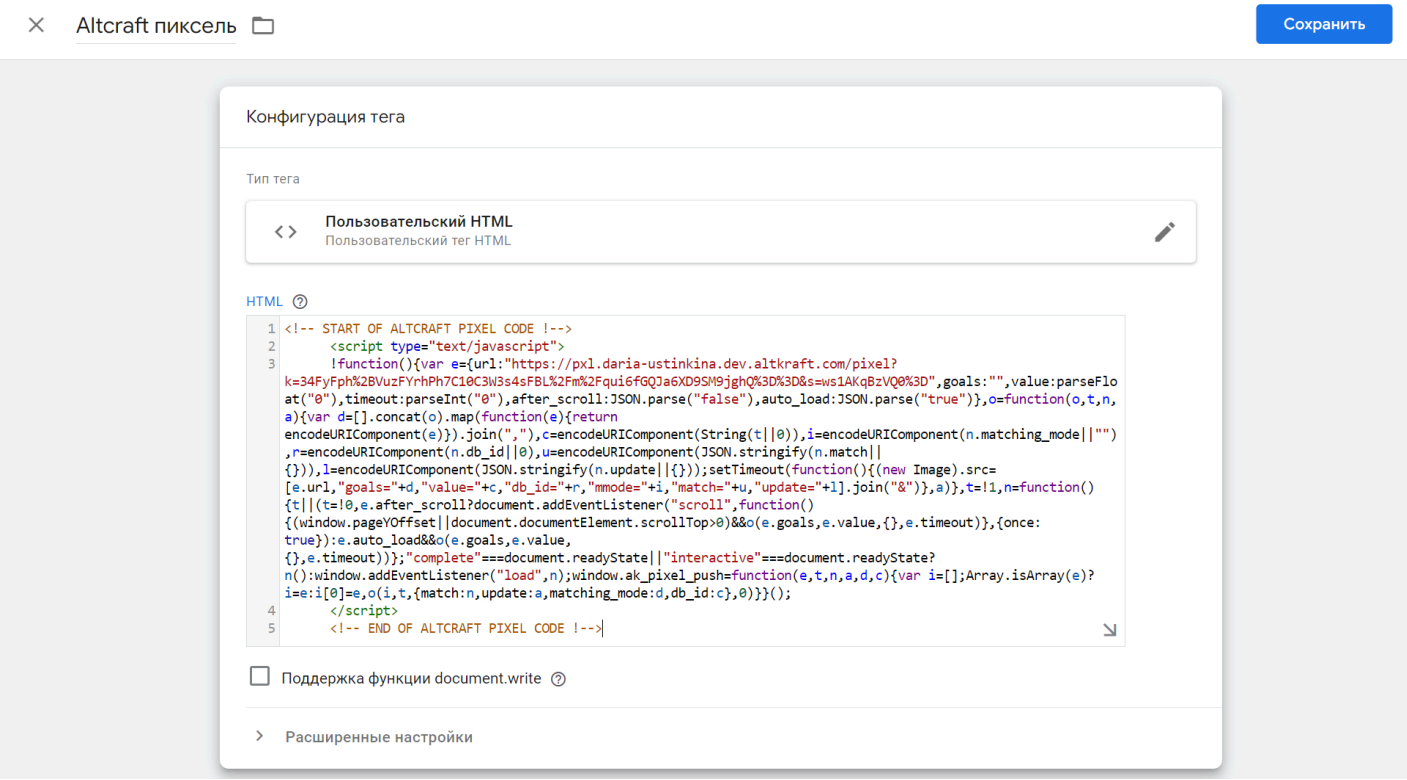
CAUTION

Необходимо перед копированием отменить **“Активировать сразу”**. Это позволит использовать код отслеживания в качестве библиотеки.



3. Далее перейдите в рабочую область в GTM и создайте новый Тег с типом **Пользовательский HTML.**

4. Добавьте код пикселя Altcraft в поле Тега, укажите **Триггер** для активации **All Pages** и сохраните.



После публикации контейнер с кодом пикселя Altcraft будет загружаться на всех страницах вашего сайта.

5. Теперь для передачи в Altcraft целей и, при необходимости, их значений — добавьте нужное количество Тегов типа **Пользовательский HTML**, настроенных на нужные вам Триггеры.

```
<script>
        ak_pixel_push("PageView", 0);
</script>
```

INFO

Аргументы добавляются по аналогии с Вызов API регистрации целей.

Altcraft просмотр страниц сайта

Сохранить

Конфигурация тега

Тип тега

Пользовательский HTML
Пользовательский тег HTML

HTML

```
<script>
  ak_pixel_push("PageView",0);
</script>
```

Поддержка функции document.write

Расширенные настройки

Триггеры

Триггеры активации

Изменение в истории
Изменение в истории

Целевые действия клиентов и скоринг

Пиксели Альткрафт позволяют отслеживать действия клиентов на сторонних сетевых ресурсах, таких как:

- Веб сайты и посадочные страницы, в том числе партнёрские.
- Форумы, социальные сети, записи в блогах и подобных сервисах.
- Мобильные и десктопные приложения.
- CRM и интеграционные сервисы компании.

Каждое зарегистрированное действие клиента содержит два параметра:

- **Цель (goal)** — название целевого действия.
- **Стоимость (value)** — связанное с целевым действием числовое значение.

Сам **Пиксель** при этом группирует цели по типу или источнику трафика, что может быть использовано в различных механиках, описанных в этой статье.

CAUTION

В названиях целей можно использовать:

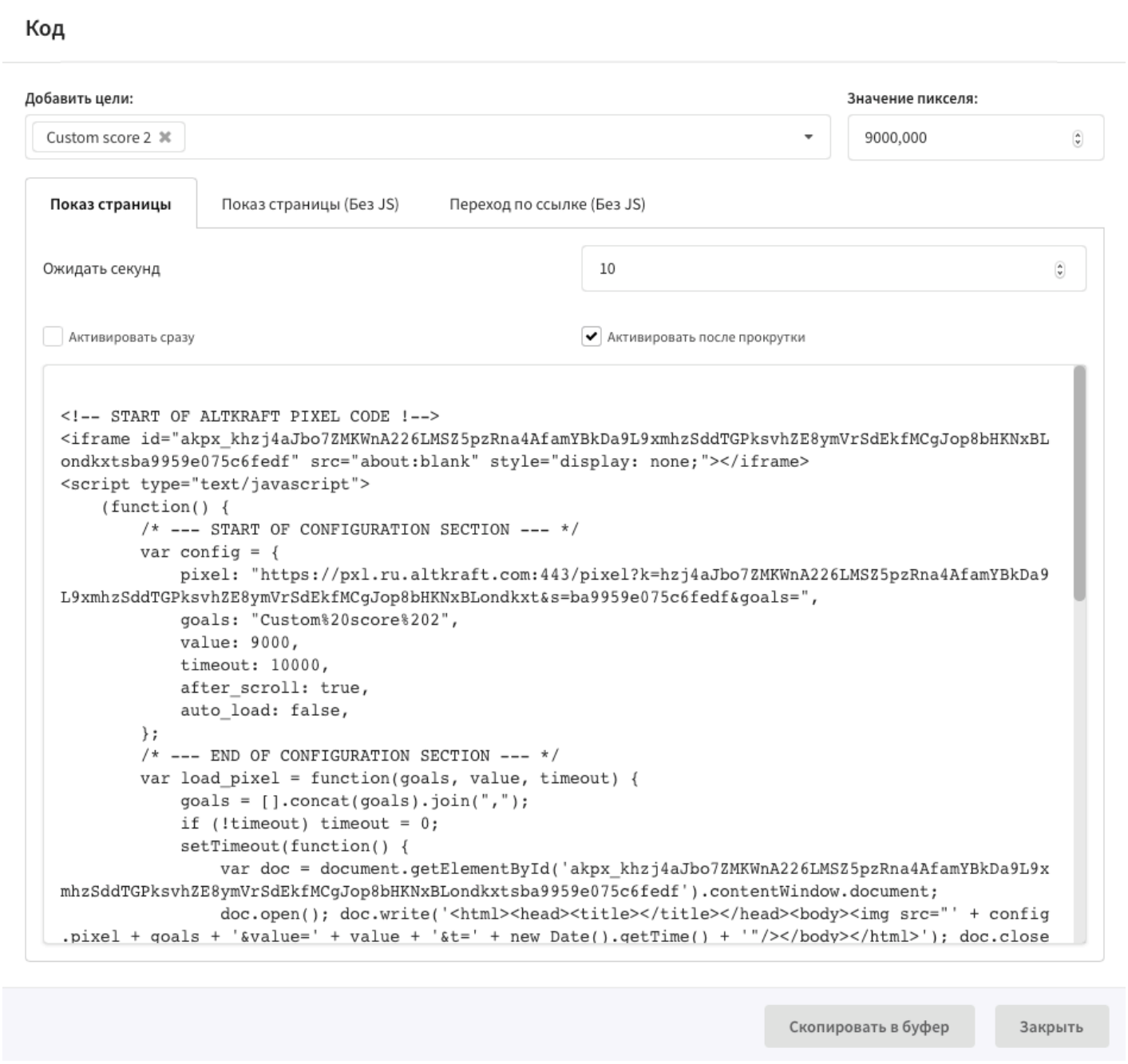
- буквы латинского и русского алфавита (любой регистр)
- пробелы и табуляцию
- символы валют
- `! # $ % & * + - . , : ; '`
- `= ? @ ^ _ ( ) [ ] { } / \ |`

Максимальная длина названия — 128 символов.

Целевые действия при просмотре web-страницы

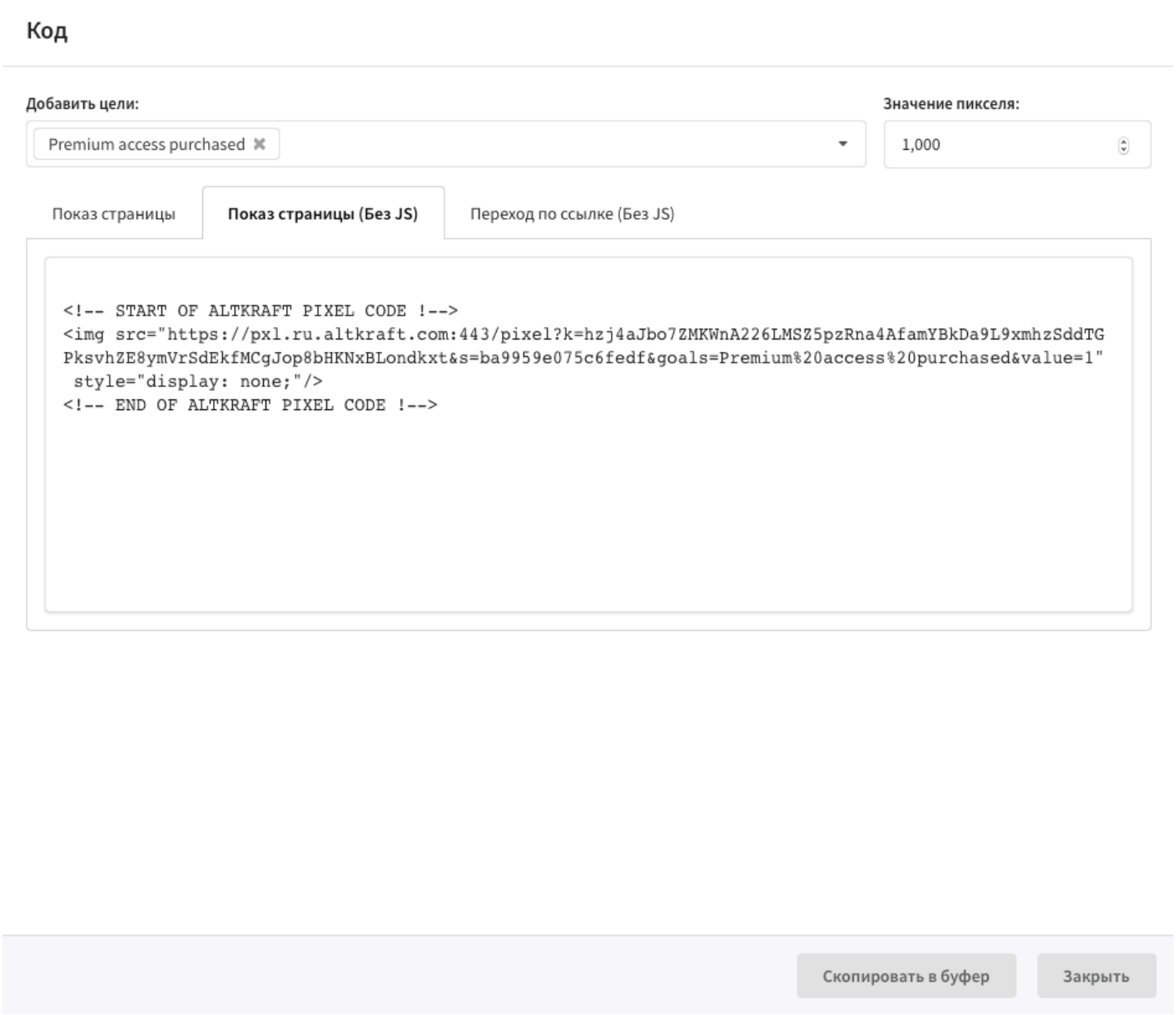
Для регистрации целевых действий посетителей определённой страницы при её просмотре можно использовать два типа скриптов:

- **Показ страницы** — для этого вида отслеживания применяется JavaScript. Активация скрипта настраивается, например, можно установить срабатывание только после прокрутки страницы пользователем. Выберите необходимый пиксель, нажмите “**Показать код пикселя**” и в соответствующей вкладке сконфигурируйте параметры скрипта: Цели, Значение, Ожидание и необходимость прокрутки.



После этого скопируйте код и передайте его разработчикам сайта.

- **Показ страницы (Без JS)** — это самый простой способ, прозрачное изображение, которое хранится на трекинг-сервере платформы. Вставляется в любую часть веб-страницы и при загрузке страницы клиентом регистрирует указанное целевое действие и связанное с ним значение.



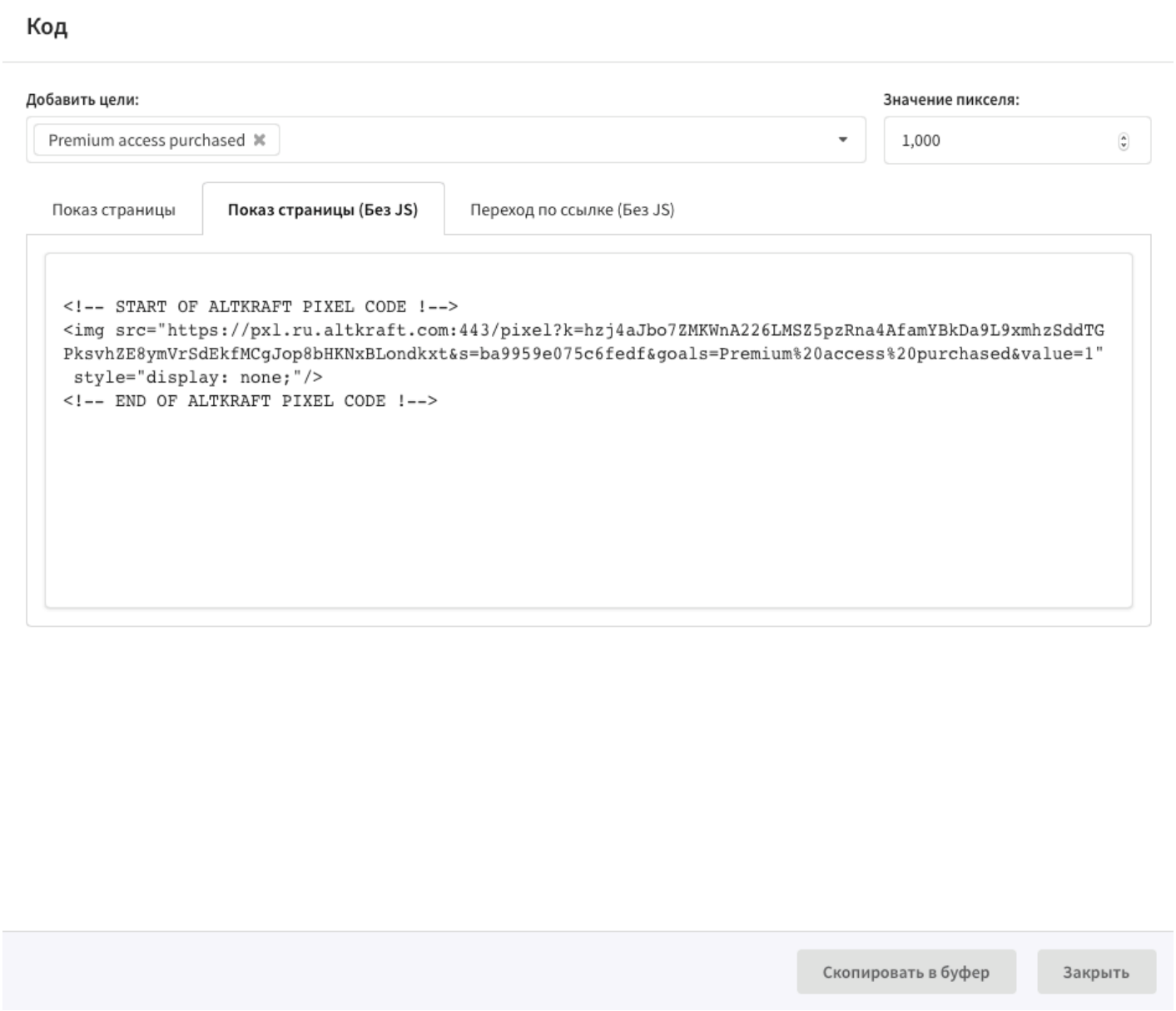
TIP

Если необходимо регистрировать несколько целей, каждое со своим значением, а также если нужен более продвинутый алгоритм регистрации целевых действий, воспользуйтесь API регистрации целей.

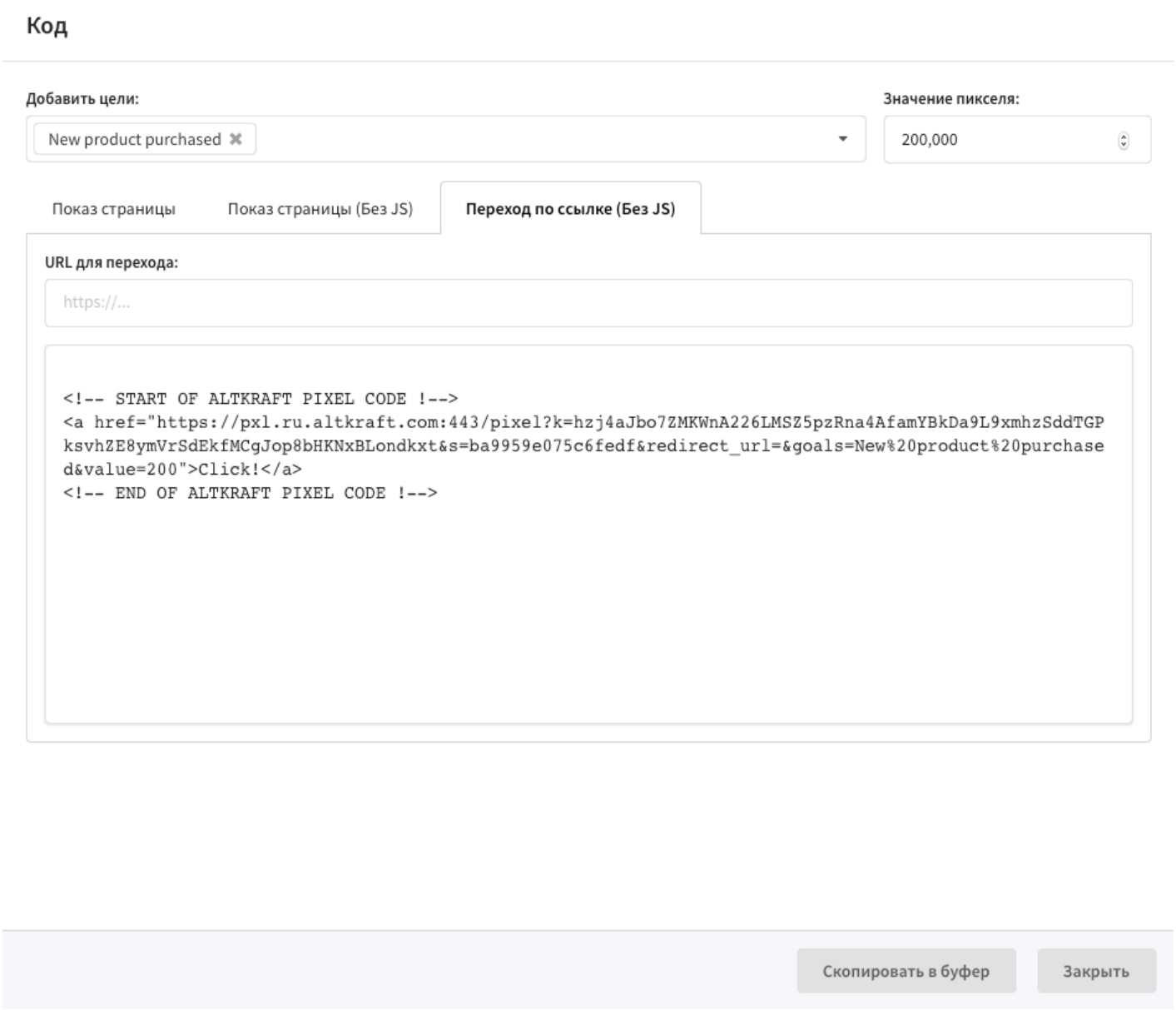
Целевые действия в соцсетях и на других публичных ресурсах

С помощью пикселей можно отследить действия клиентов в социальных сетях, на форумах и подобных веб-ресурсах. Подойдут два варианта:

- **Показ страницы (Без JS)** — прозрачное изображение, загружаемое с трекинг-сервера платформы, вставляется в форму, поддерживающую изображения с внешних источников. При просмотре клиентом такой записи регистрируется целевое действие.



- **Переход по ссылке (Без JS)** — добавьте перенаправление на трекинг-сервер платформы для любой ссылки, которую хотите разместить в публикации — в соцсети или на схожем ресурсе. Переходы по такой ссылке будут регистрировать заданное целевое действие и его стоимость.



Целевые действия в десктопных, мобильных и веб приложениях

Трекинг событий в приложениях осуществляется запросами к API регистрации целей. Отличительной особенностью запроса к API является группировка целей и стоимости. В отличии от готовых скриптов, в запросе поддерживаются пары “цель” + “значение”, так что можно передать несколько целей, у каждой из которых будет свое значение.

Попапы

	Попап (от англ. "pop-up", всплывающее окно) — это небольшое окно, которое появляется поверх основного содержимого веб-страницы. Его используют, чтобы привлечь внимание пользователя к определенной информации или мотивировать его совершить целевое действие. Например, с помощью попапов можно предложить подписаться на рассылку, рассказать о скидке или акции, собрать заявки или оказать техподдержку.

INFO

Попапы всплывают на экране устройства при наступлении определенного события — например, при загрузке страницы, при клике на элемент или при скролле страницы до заданного порога. Чтобы задать условия появление попапа, рекомендуем установить на сайт Менеджер тегов Altcraft.

В этом разделе

Создание и публикация попапа

Данное руководство содержит инструкции по созданию и настройке всплывающих окон (попапов) для вашего сайта с использованием платформы Altcraft. Вы узнаете, как настроить внешний вид окна, задать правила его показа и проанализировать результаты.

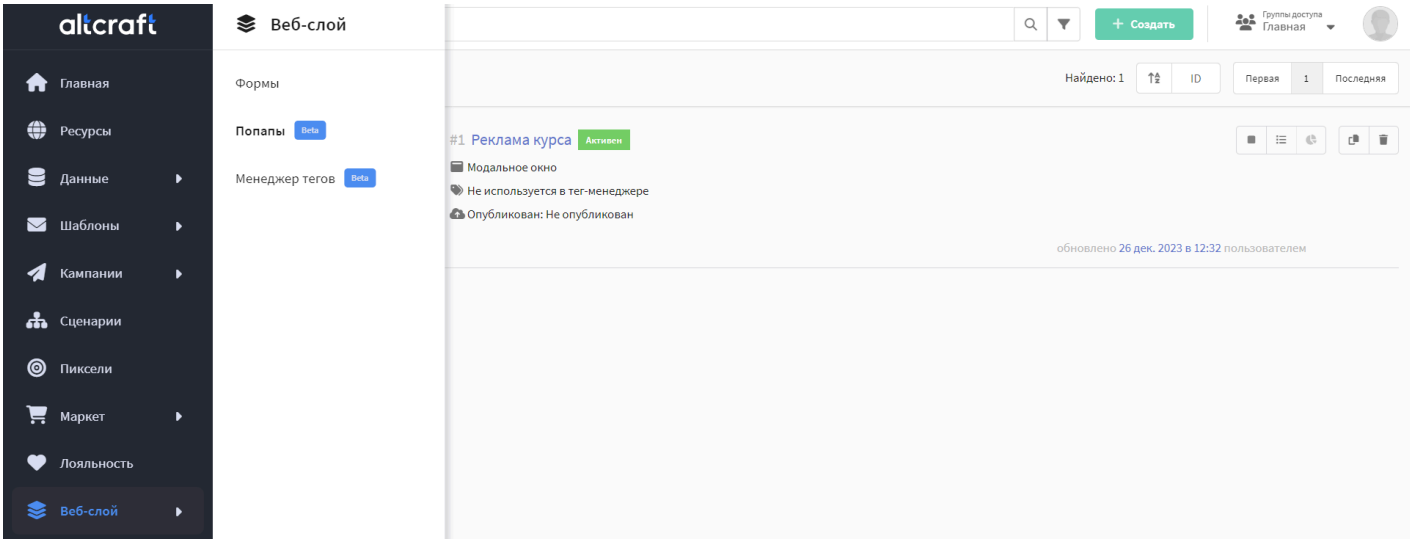
Полный алгоритм создания и настройки попапа

- **Шаг 1: Создание попапа**
  Выбор типа попапа (модальное окно, плашка, всплывающее окно) и базового шаблона для быстрого старта
- **Шаг 2: Настройка контента и дизайна**
  Работа с визуальным редактором: добавление текста, медиа, форм и изменение внешнего вида под бренд
- **Шаг 3: Настройка действий после отправки формы**
  Отображение сообщения после заполнения формы, перенаправление пользователя и выбор формата уведомлений.
- **Шаг 4: Публикация и настройка показа**
  Интеграция кода на сайт и настройка правил отображения (время, триггеры, целевая аудитория)
- **Шаг 5: Аналитика результатов**
  Мониторинг ключевых метрик (конверсия, показы, клики) для оценки эффективности и принятия решений

Каждый из этих шагов мы подробно разберем ниже.

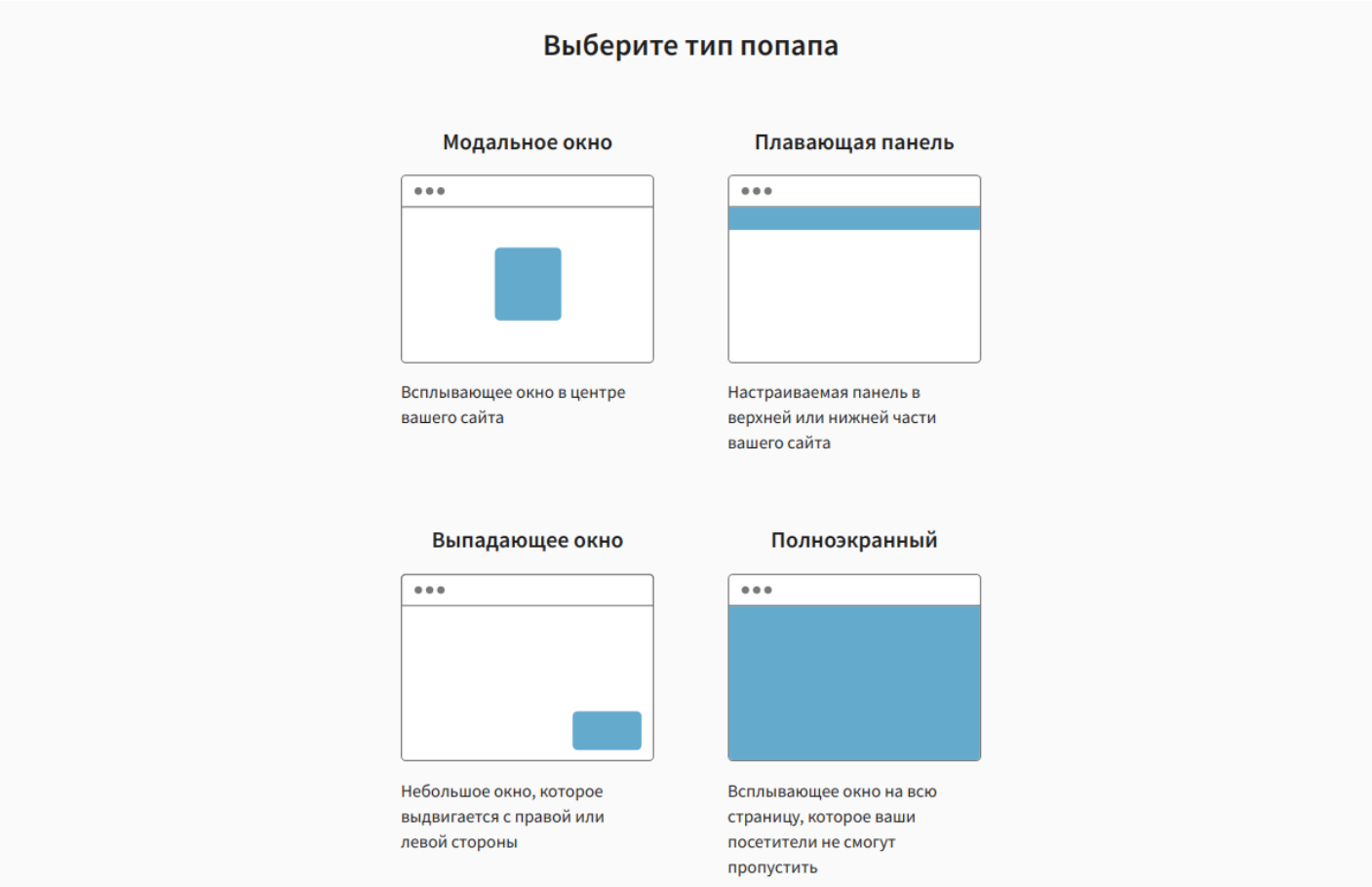
Создание попапа

Перейдите в раздел **“Веб-слой”** → **“Попапы”** и нажмите **“+ Создать”** в верхней части страницы:

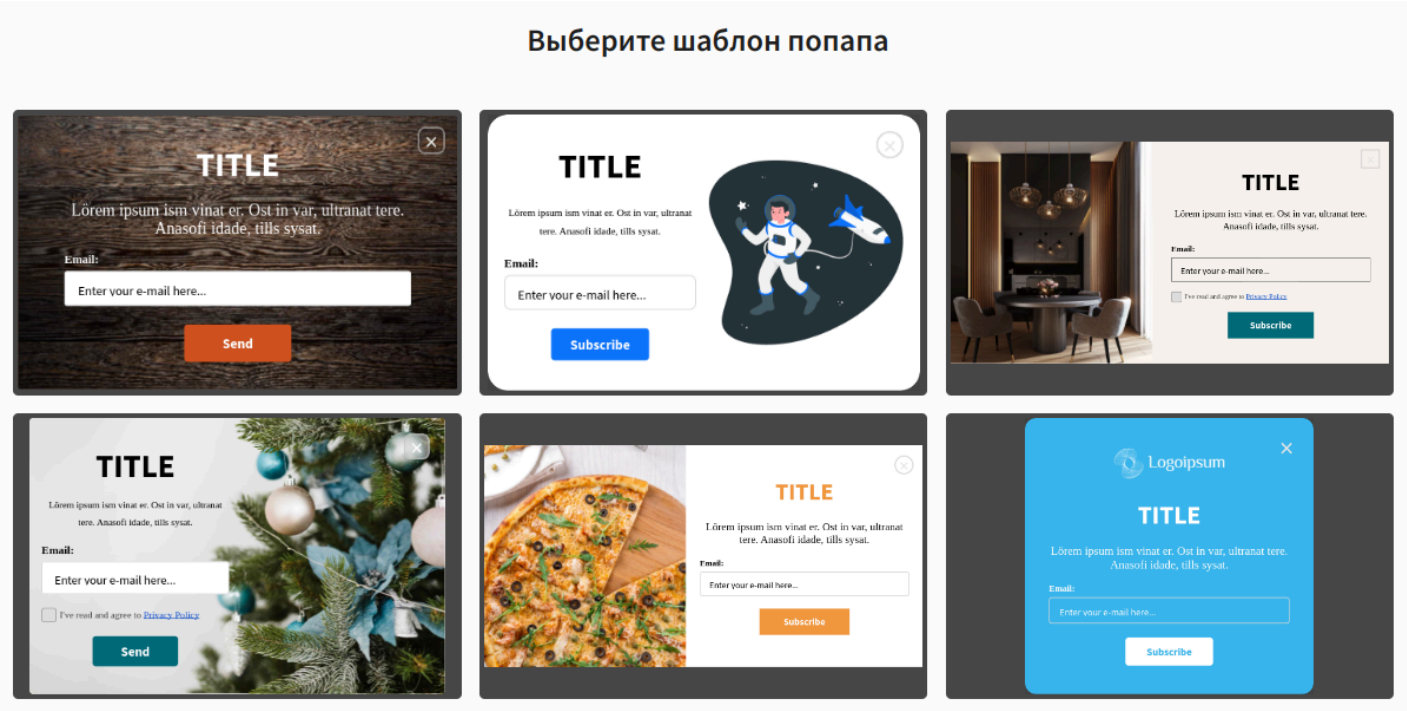


Выберите **тип попапа:**

- **Модальное окно** — всплывающее окно, которое появляется в центре экрана. Оно привлекает внимание пользователя, так как затемняет фон браузера. Обычно используется для сбора email-адресов.
- **Плавающая панель** — настраиваемая панель, которая располагается в верхней или нижней части страницы. Остаётся видимой на экране даже при прокрутке страницы. Подходит для информирования клиентов о новостях или о лимитированных предложениях.
- **Выпадающее окно** — небольшое окно, которое появляется в правом нижнем углу страницы. Оно менее навязчиво и не мешает посетителю просматривать содержимое страницы. Выпадающие окна могут использоваться для вывода дополнительной информации, предложений или форм обратной связи.
- **Полноэкранный** — всплывающее окно, которое занимает весь экран браузера. Это гарантирует, что посетители не пропустят представленную информацию. Однако использовать полноэкранные окна следует осторожно, так как они могут быть восприняты как навязчивая реклама и вызвать недовольство у пользователей.



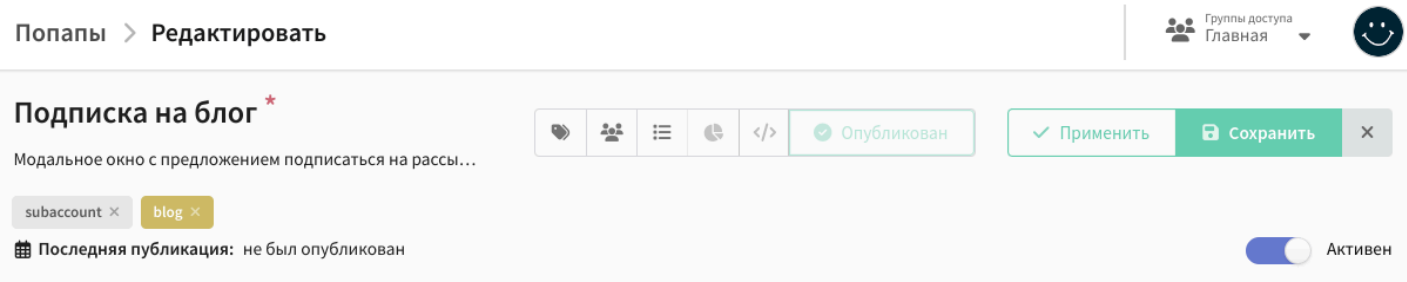
Выберите **шаблон попапа**. В платформе уже есть несколько готовых шаблонов. Используйте любой из них как основу. На следующем шаге вы сможете полностью изменить шаблон (задать текст, стили, настроить поля и т. д.) и подстроить его под свои задачи.



Задайте системное **имя** попапа и при необходимости добавьте **описание**. Также можно установить [теги для быстрого поиска]

и назначить [группы доступа]

.



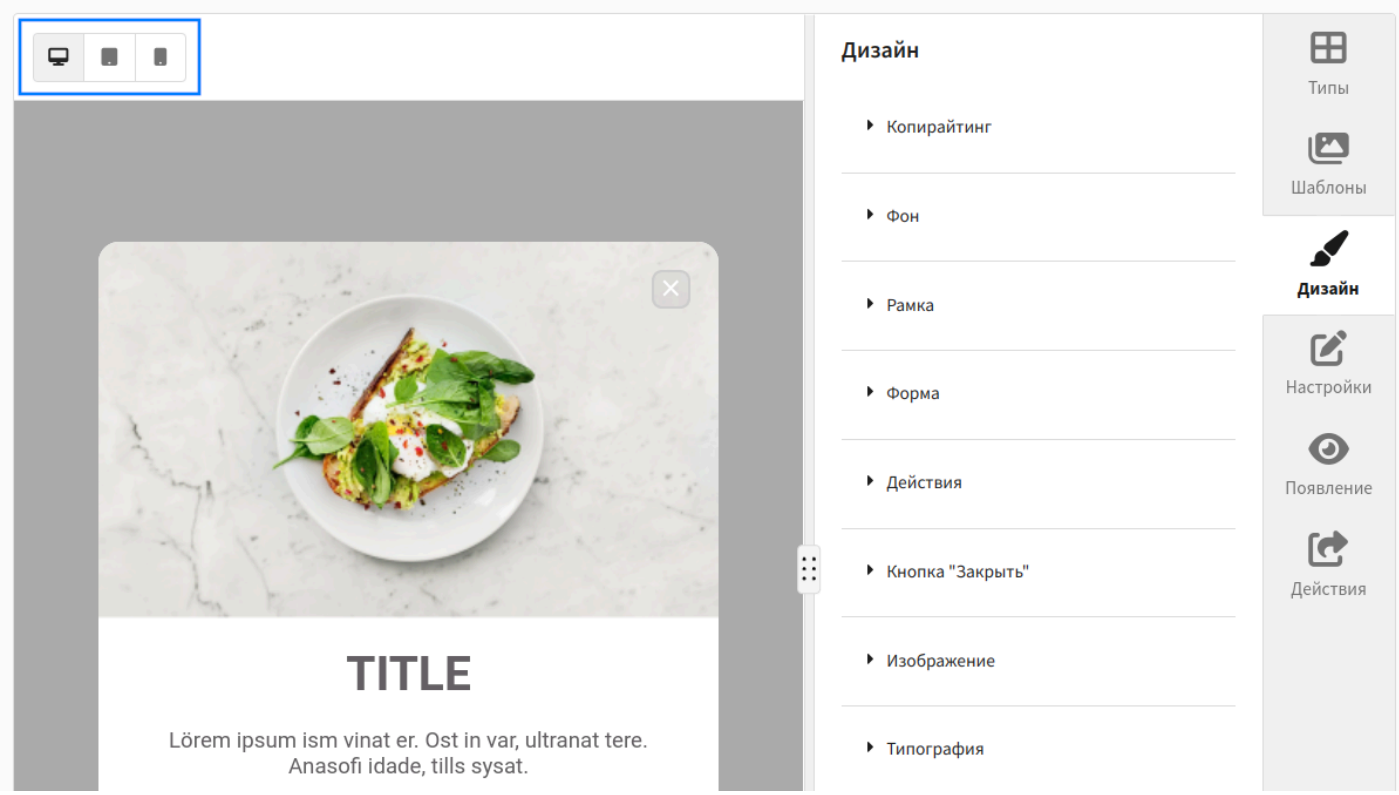
Настройка контента и дизайна

Внешний вид попапа задается с помощью параметров. Параметры содержат текст, изображение и CSS-стили различных элементов. Например, параметр `text` содержит основной текст попапа, параметр `backgroundColor` — его фоновый цвет.

Доступно два инструмента для работы с контентом попапа:

- **Редактирование параметров в визуальном редакторе.** Подходит для пользователей, не знакомых с кодом. Для изменения внешнего вида попапа достаточно ввести текст, выбрать изображения в галерее или использовать палитры цветов.
- **Изменение HTML, CSS и JavaScript в редакторе кода.** Подходит для пользователей, которые хотят работать напрямую с кодом попапа.

Любые изменения, внесенные в редакторе, отображаются в окне предпросмотра в режиме реального времени. Чтобы проверить, как попап будет выглядеть на разных устройствах, используйте кнопки в левом верхнем углу:



Визуальный редактор

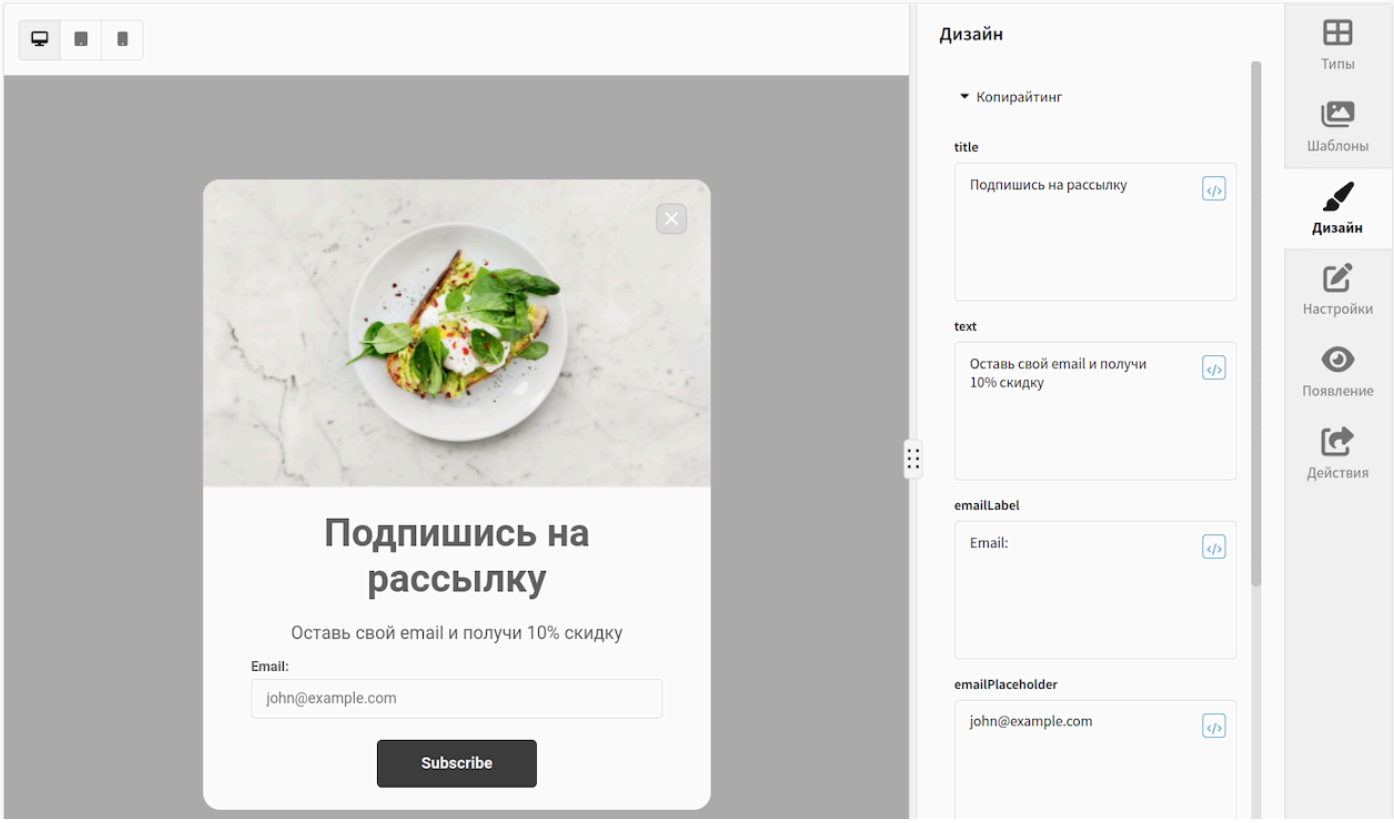
Чтобы настроить текст и изменить внешний вид всплывающего окна в визуальном редакторе, перейдите в раздел **“Дизайн”**. В этом разделе все параметры разбиты на категории:

Копирайтинг

На вкладке **“Копирайтинг”** задаётся текст попапа. Доступны следующие параметры:

- **title** — заголовок;
- **text** — основной текст;
- **<field>Label** — лейбл для полей формы;
- **<field>Placeholder** — текстовая подсказка внутри поля формы.

В любой текстовый элемент можно добавить переменную `</>`, чтобы динамически подставлять в попап данные профиля. Описание переменных вы найдете в этой статье.

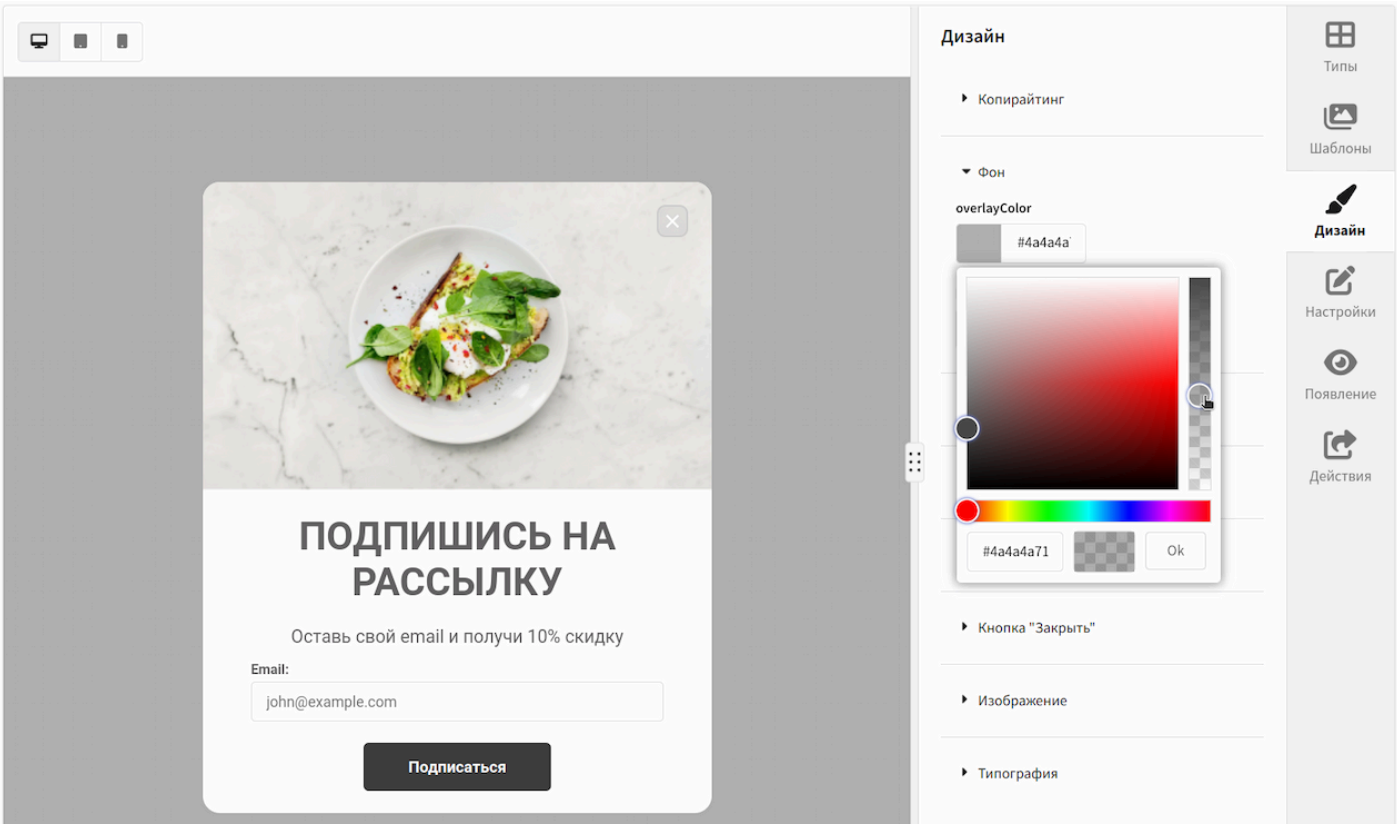


Фон

На вкладке **“Фон”** доступны следующие параметры:

- **backgroundColor** — фоновый цвет попапа;
- **backgroundImage URL** — фоновое изображение попапа. Загрузите изображение в галерею платформы

  или укажите URL-адрес;
- **overlay** — фоновый цвет страницы браузера. При появлении попапа фон страницы затемняется, и с помощью палитры вы можете изменить цвет и прозрачность затемнения.

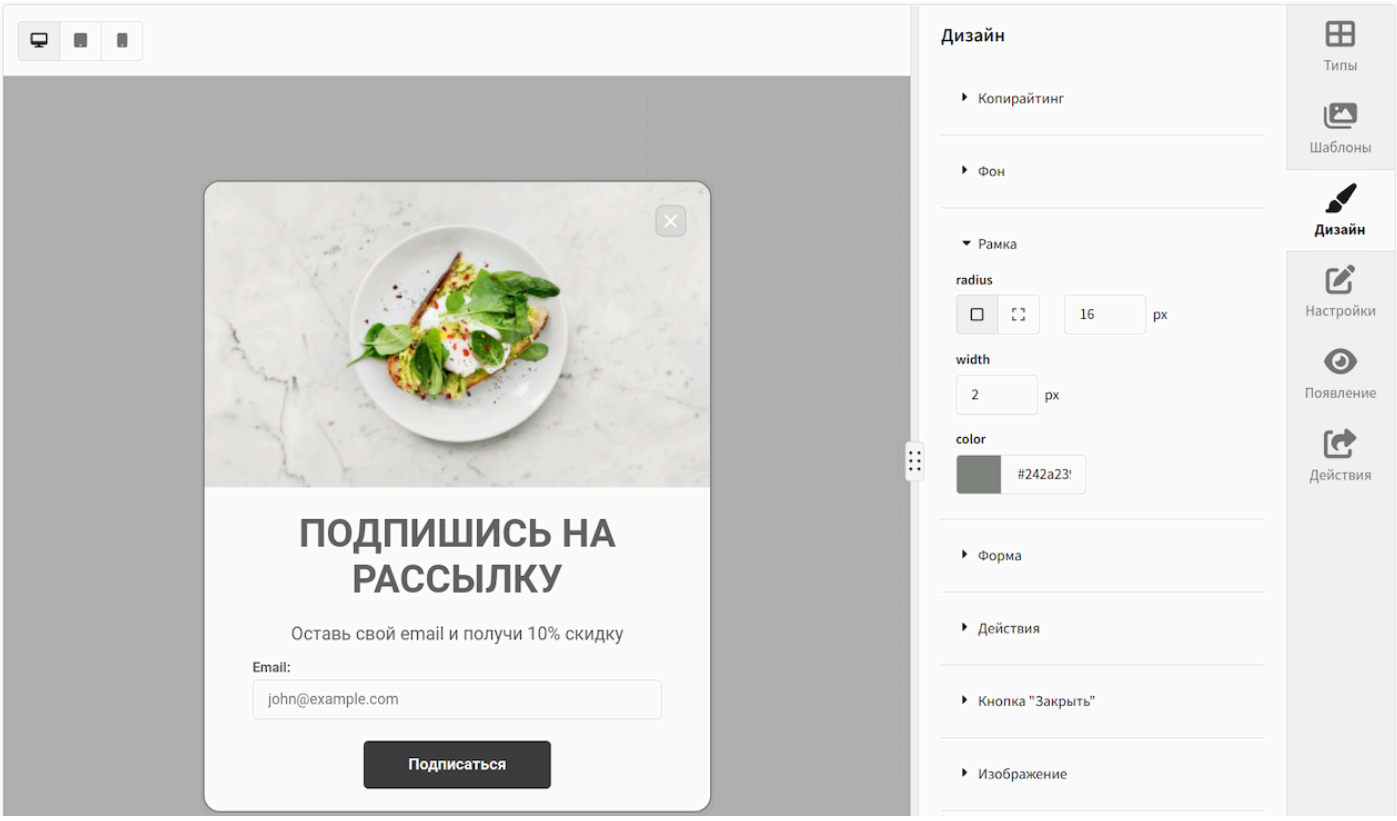


Рамка

На вкладке **“Рамка”** настраиваются границы всплывающего окна. Доступны следующие параметры:

- **borderRadius** – скругление границ попапа в пикселях. Можно задать скругление для каждого угла отдельно

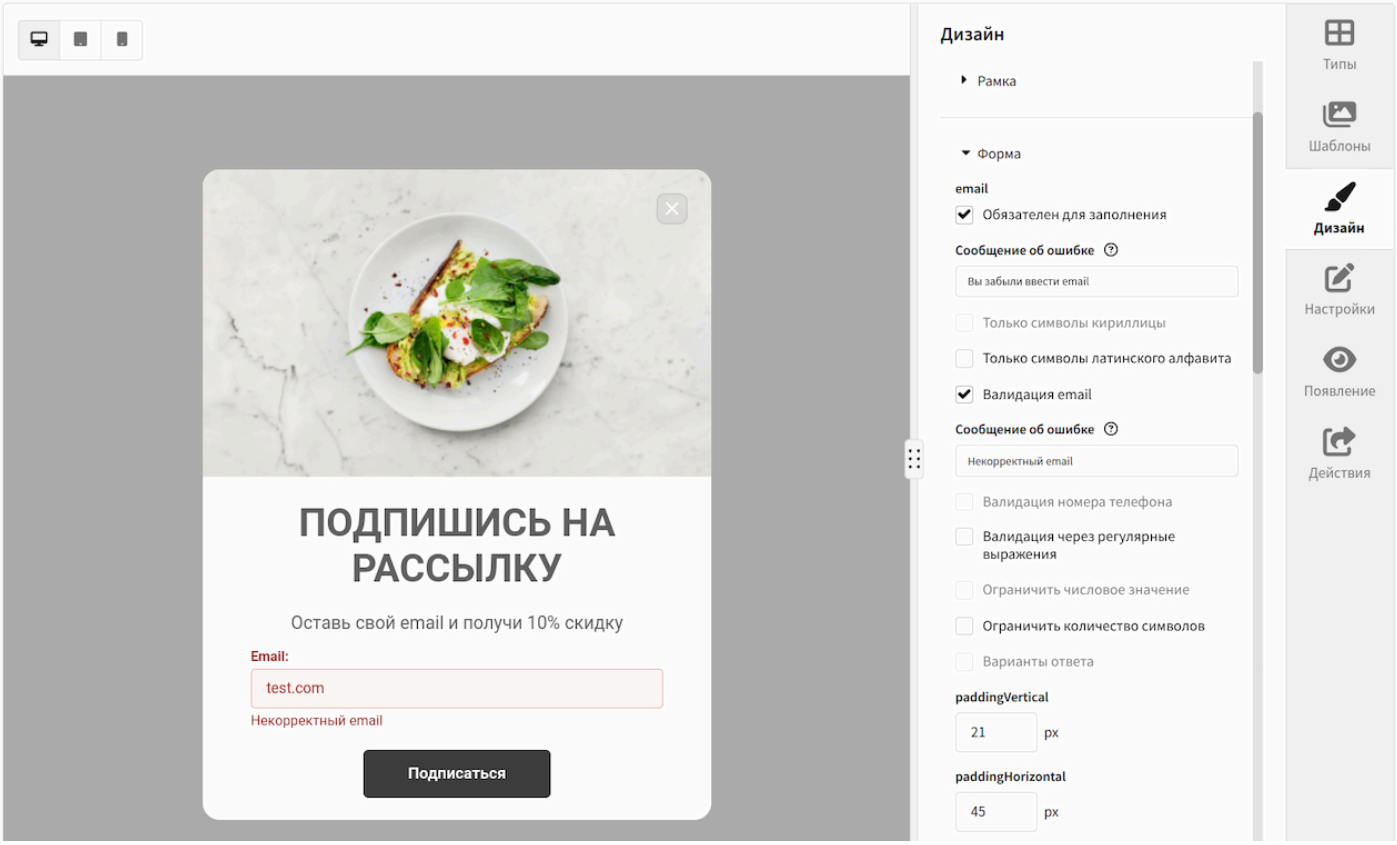
  ;
- **width** — толщина границы;
- **color** — цвет границы.



Форма

На вкладке **“Форма”** настраиваются поля для ввода данных:

- задаётся валидация;
- указывается текст ошибки, который увидит пользователь, если введет некорректные данные;
- редактируются CSS-стили полей формы.



Валидация:

- **Обязателен для заполнения** — пользователь должен заполнить это поле перед отправкой формы. Если поле останется пустым, он увидит сообщение об ошибке;
- **Только символы кириллицы** — разрешен ввод только символов кириллицы;
- **Только символы латинского алфавита** — разрешен ввод только латинских символов;
- **Валидация email** — введенное значение должно соответствовать формату электронной почты;
- **Валидация номера телефона** — введенное значение должно соответствовать формату номера телефона;
- **Валидация через регулярные выражения** — возможно использование пользовательских регулярных выражений для определения собственных правил валидации поля ввода;
- **Ограничить числовое значение** — задает минимальное и максимальное значение для числового поля;
- **Ограничить количество символов** — ограничивает количество символов, которое можно ввести в поле;
- **Варианты ответа** — используется для полей с предопределенными вариантами ответа. Вы можете указать допустимые варианты ответа, и пользователь должен выбрать один из них. Если пользователь выбирает недопустимый вариант, будет отображено сообщение об ошибке;
- **Сообщение об ошибке** — этот текст увидит пользователь, если введёт данные, которые не соответствуют условиям валидации. Сообщение об ошибке задаётся для каждого поле отдельно.

Стили окна:

- **paddingVertical** — отступ между кнопкой и нижней границей попапа;
- **paddingHorizontal** — отступ между полями и левой границей попапа;
- **width** — ширина попапа;
- **fieldsGap** — расстояние между полями формы, в т.ч. между полем и кнопкой;
- **borderWidth** — ширина границы полей для ввода;
- **borderColor** — цвет границы полей для ввода;
- **borderRadius** — скругление границ полей для ввода. Можно задать скругление для каждого угла отдельно

  .

Стили поля с типом input:

- **inputPaddingVertical** — отступ placeholder-текста от верхней и нижней границы поля для ввода;
- **inputPaddingHorizontal** — отступ placeholder-текста от левой границы поля для ввода;
- **inputFontSize** — размер шрифта placeholder-текста;
- **inputBackgroundColor** — фоновый цвет поля для ввода;
- **inputColor** — цвет текста, который пользователь вводит в поле;
- **inputHoverColor** — цвет границы поля для ввода при наведении курсора;
- **inputActiveColor** — цвет границы поля для ввода при нажатии кнопкой мыши.

Стили поля с типом checkbox:

- **checkboxSize** — размер шрифта чекбокса;
- **checkboxLabelFormatting** — форматирование текста чекбокса.

Стили сообщения об ошибке:

- **errorColor** — цвет сообщения об ошибке;
- **errorFontSize** — размер шрифта в сообщении об ошибке;

- **inputErrorBorderColor** — цвет границы поля для ввода при появлении сообщения об ошибке;
- **inputErrorBackgroundColor** — фоновый цвет поля для ввода при появлении сообщения об ошибке.

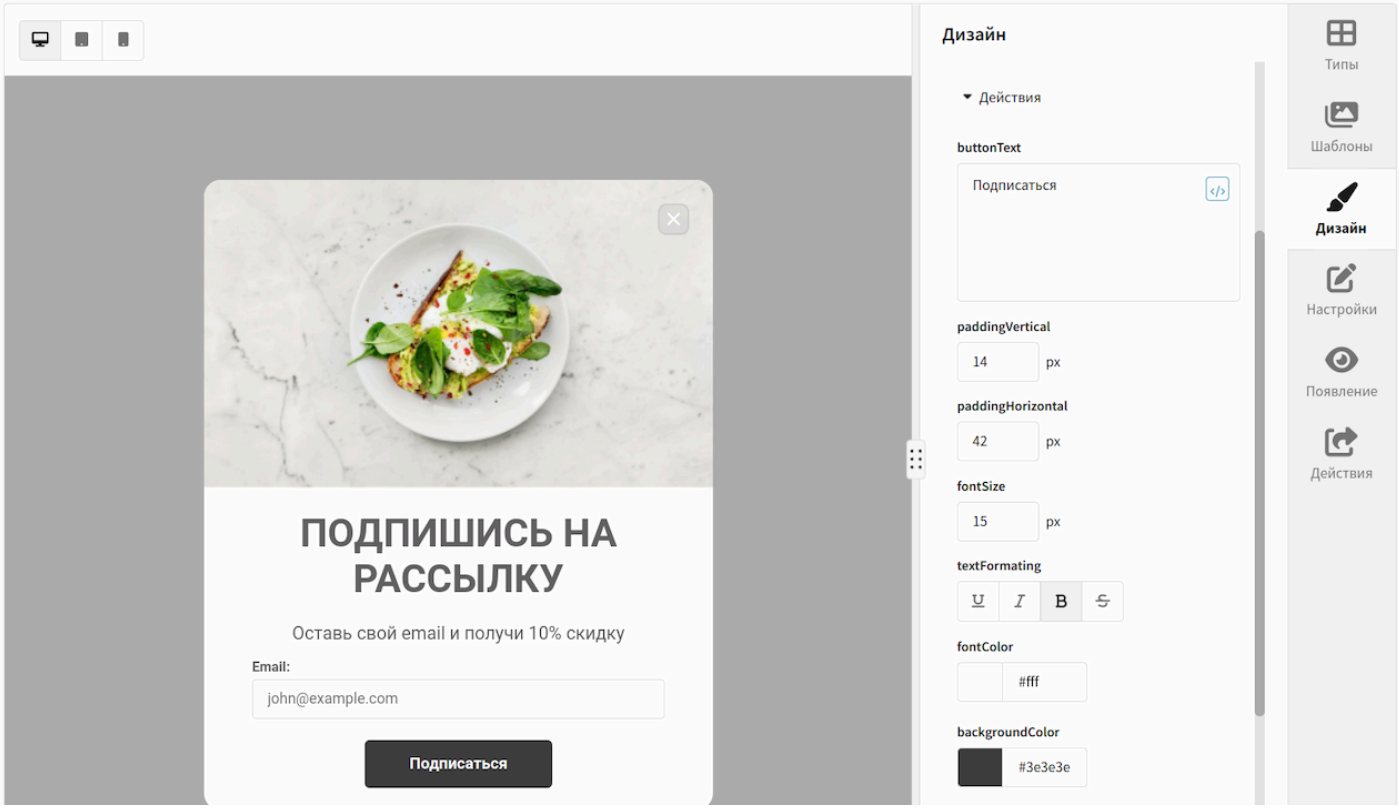
Обратите внимание, что при работе в визуальном редакторе может быть создана **только одна форма**. Для того чтобы добавить больше, используйте **редактор кода**.

Действия

На вкладке **“Действия”** редактируется текст кнопки и её CSS-стили. Доступны следующие параметры:

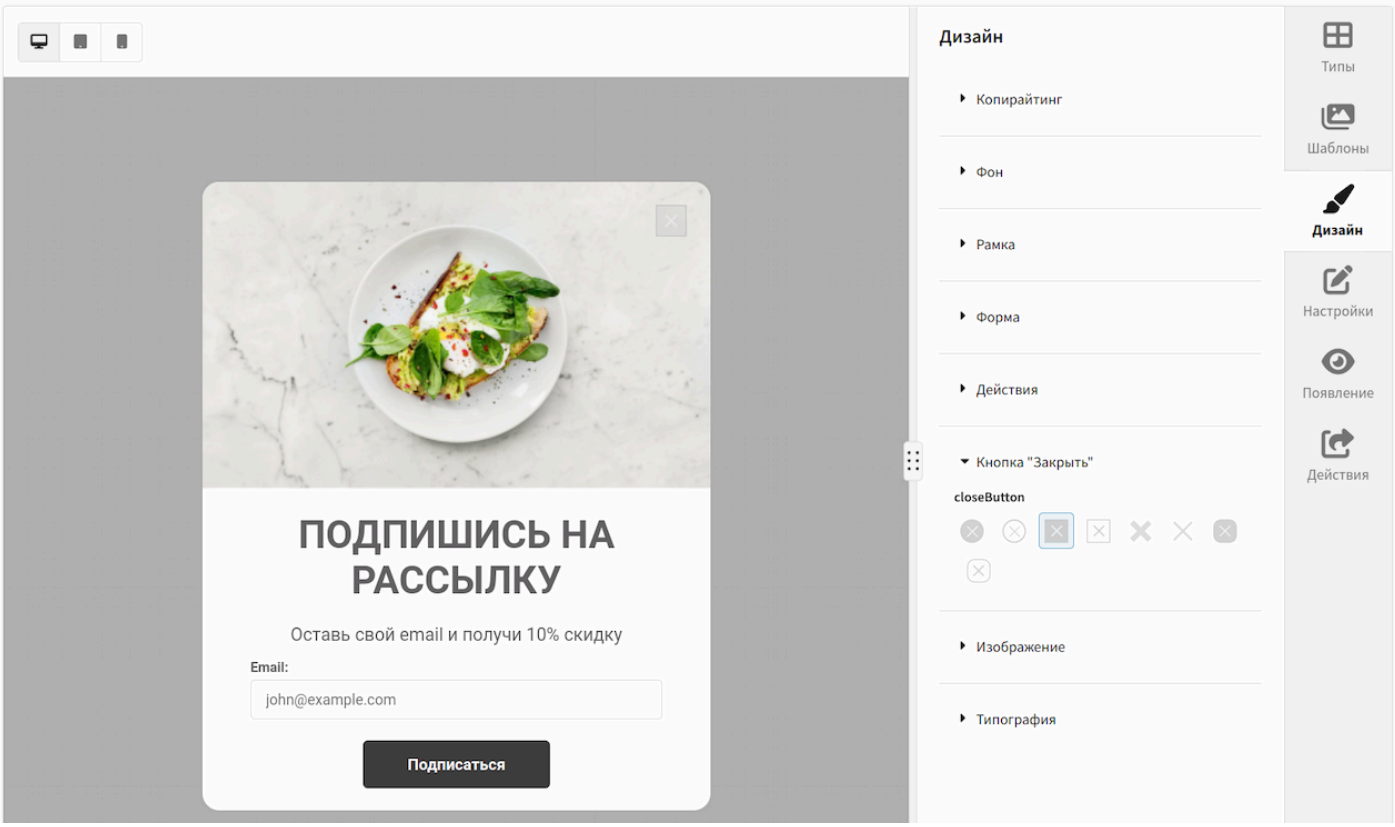
- **text** — текст кнопки для отправки формы;
- **textFormatting** — форматирование текста в кнопке;
- **borderRadius** — скругление границ кнопки. Можно задать скругление для каждого угла отдельно

  ;
- **backgroundColor** — фоновый цвет кнопки;
- **hoverBackgroundColor** — фоновый цвет кнопки при наведении;
- **color** — цвет текста в кнопке;
- **fontSize** — размер текста в кнопке;
- **paddingVertical** — отступ текста кнопки от верхней и нижней границы;
- **paddingHorizontal** — отступ текста кнопки от правой и левой границы.



Кнопка ‘Закрыть’

На этой вкладке выбирается внешний вид кнопки, с помощью которой пользователь может закрыть всплывающее окно. Параметр — **closeButton**.

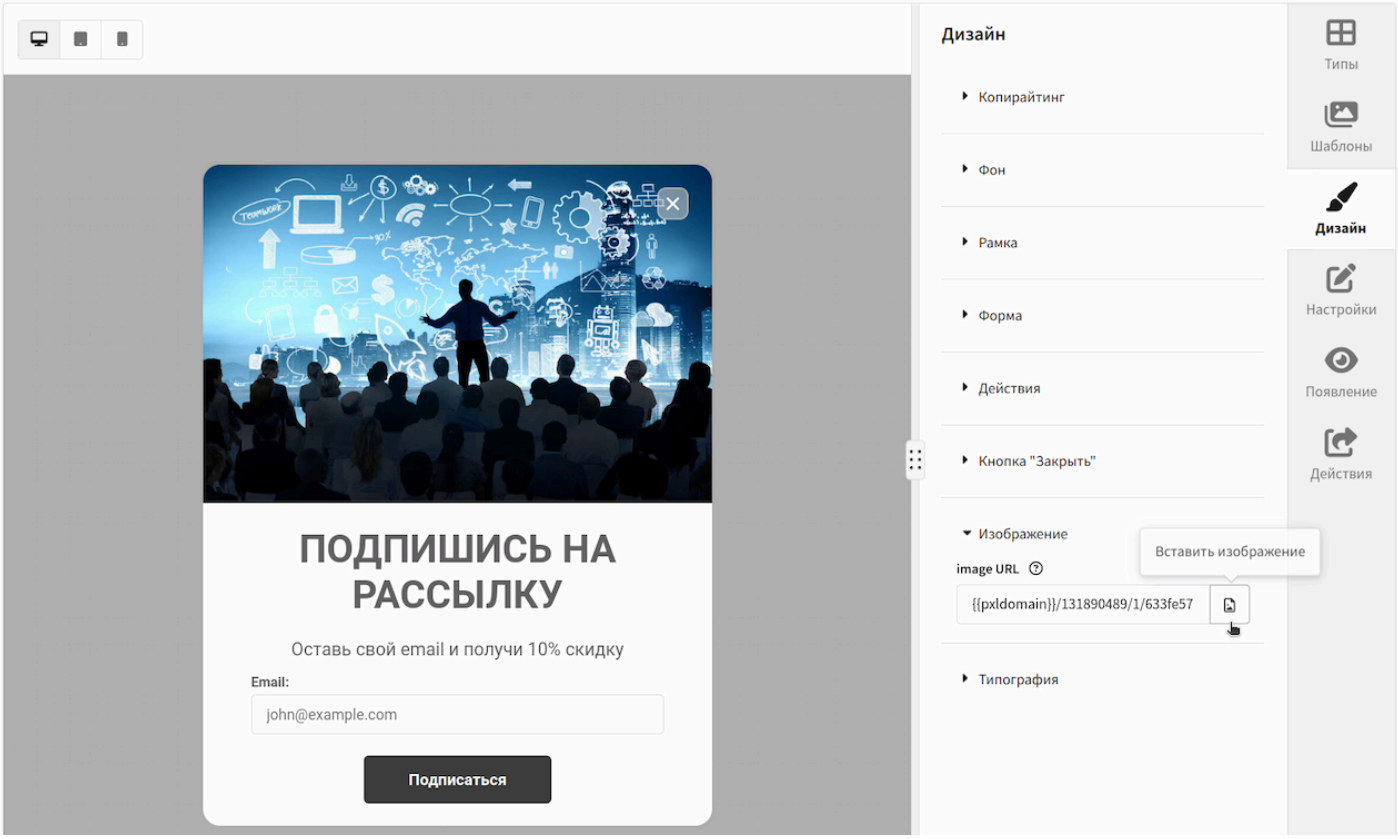


Изображение

На этой вкладке в попап добавляется изображение. Не все готовые шаблоны предоставляют такую возможность.

- **image URL** — URL-адрес изображения. Доступна загрузка изображения из галереи платформы

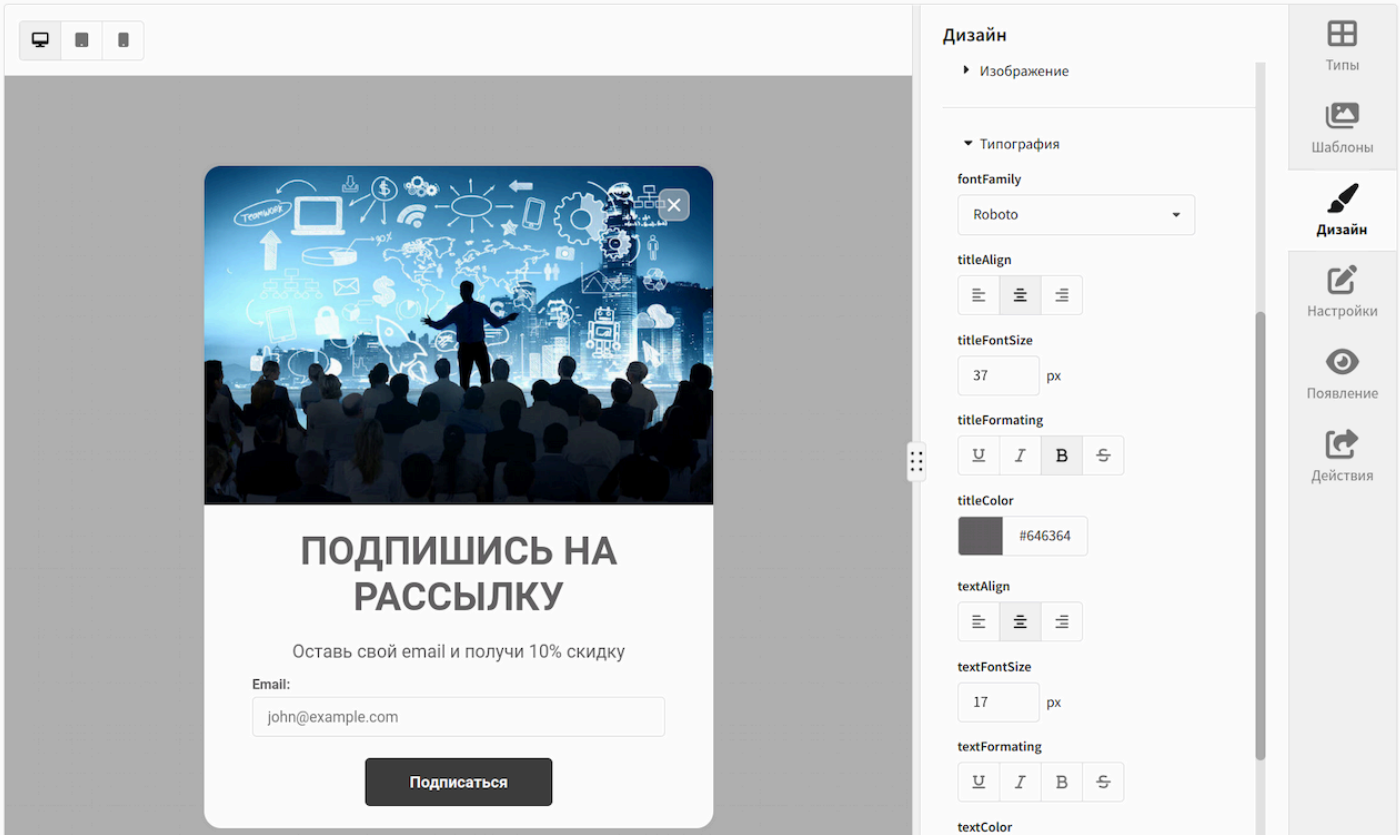
  ;
- **width** — ширина изображения в пикселях. Высота меняется пропорционально заданной ширине.



Типографика

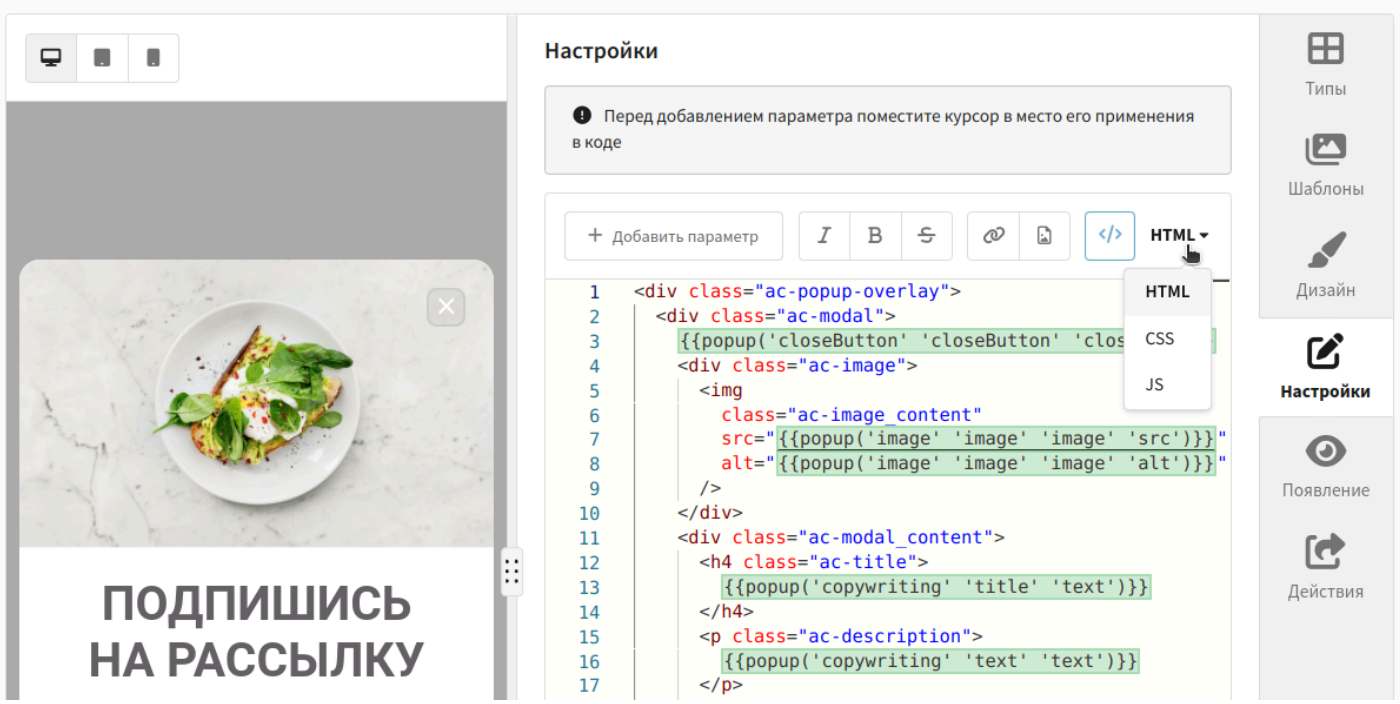
На вкладке **“Типографика”** вы можете изменить CSS-стили всех текстовых элементов:

Категория	Параметр	Описание
Общий шрифт	fontFamily	Семейство шрифта, которое используется во всех текстовых элементах попапа.
Заголовок	titleAlign	Выравнивание заголовка (по левому краю, по центру, по правому краю)
	titleFontSize	Размер шрифта в заголовке
	titleFormatting	Форматирование текста в заголовке (подчеркивание, курсив, жирный, перечеркнутый)
	titleColor	Цвет текста в заголовке
Основной текст	textFontSize	Размер шрифта в основном тексте попапа
	textAlign	Выравнивание основного текста (по левому краю, по центру, по правому краю)
	textFormatting	Форматирование основного текста (подчеркивание, курсив, жирный, перечеркнутый)
	textColor	Цвет основного текста
Лейбл	labelFontSize	Размер шрифта в лейбле поля
	labelFormatting	Форматирование текста в лейбле (подчеркивание, курсив, жирный, перечеркнутый)
	labelColor	Цвет лейбла

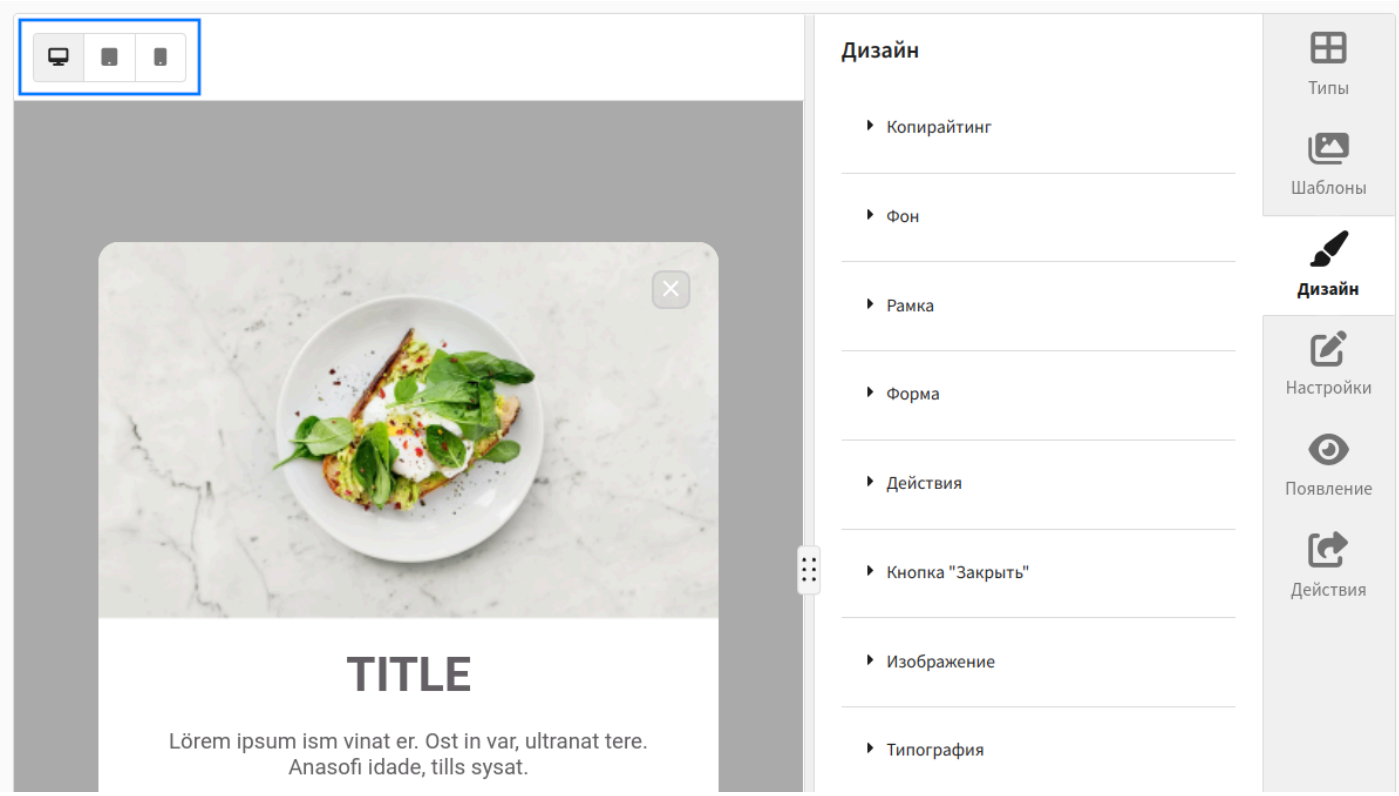


Редактор кода

Чтобы напрямую работать с кодом, перейдите в раздел **“Настройки”**. Рекомендации по вёрстке попапа в редакторе собраны в [этой статье].



Любые изменения, внесенные в редакторе, отображаются в окне предпросмотра в режиме реального времени. Чтобы проверить, как попап будет выглядеть на разных устройствах, используйте кнопки в левом верхнем углу:



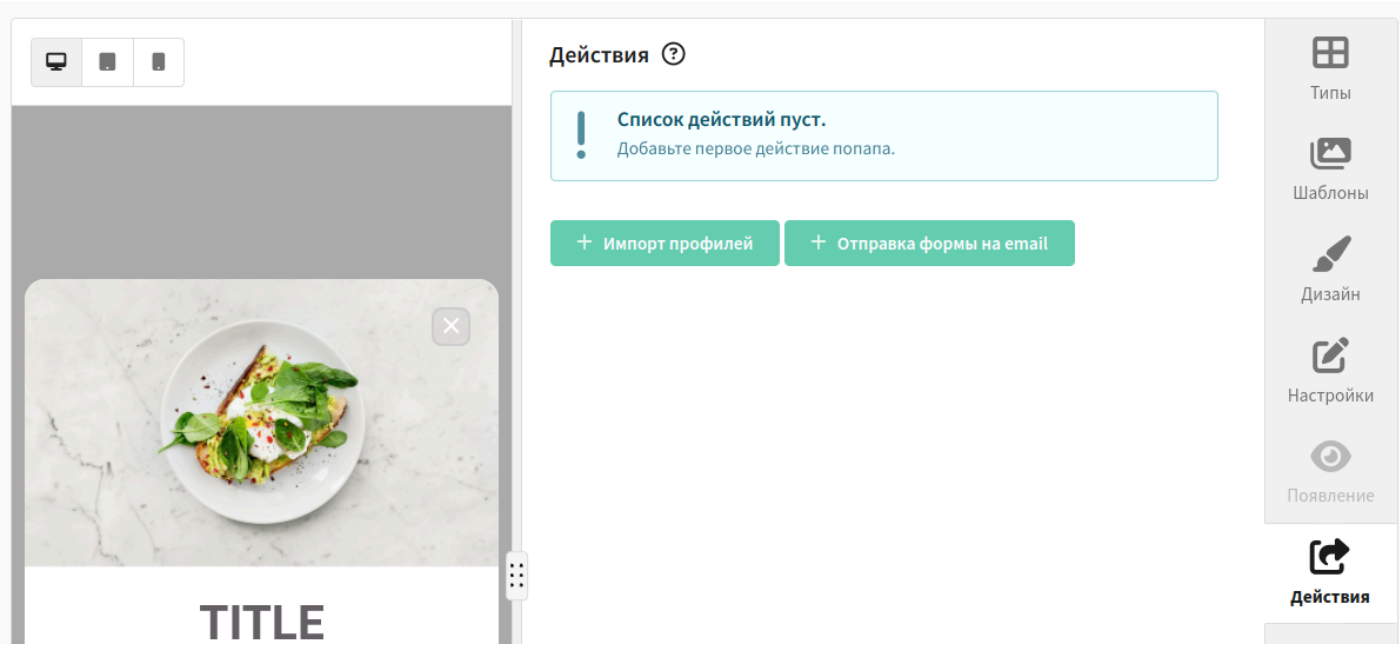
Настройка действий после заполнения формы

Перейдите в раздел **“Действия”**. Здесь вы можете определить действия, которые нужно выполнить после того, как пользователь заполнит и успешно отправит форму попапа:

- **Импорт профилей** позволяет загрузить данные пользователя, заполнившего форму, в одну или несколько баз;
- **Отправка формы на email** активирует отправку оповещений о заполнении формы на указанные email-адреса.

TIP

Действия после заполнения формы попапа аналогичны действиям после заполнения стандартной веб-формы. Подробнее про настройку действий вы можете прочитать в этой статье.



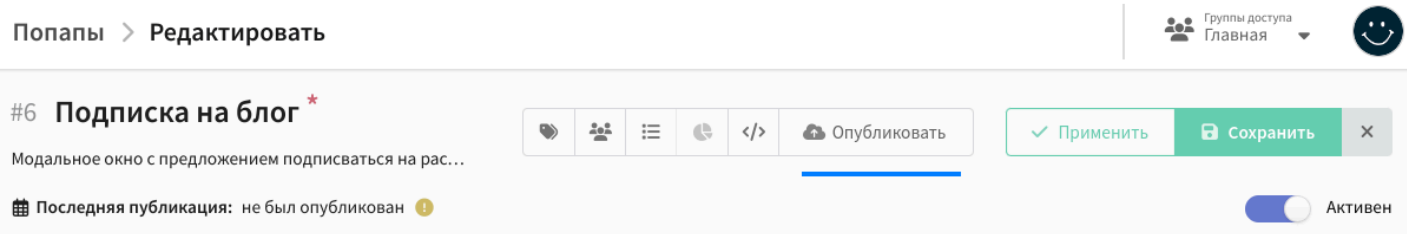
Настройка публикации и показа попапа

Доступно два механизма показа попапа на сайте:

- **Добавление скрипта на страницу сайта вручную** — этот способ используется для прямого доступа к управлению попапом. Условия показа попапа необходимо реализовывать самостоятельно.
- **Добавление скрипта через Менеджер тегов** — более простой способ. Достаточно один раз разместить код контейнера на страницах сайта, а все дальнейшие настройки попапа, в том числе условия его появления, задаются в интерфейсе платформы.

Добавление скрипта вручную

Сохраните все заданные настройки и убедитесь, что статус попапа **“Активен”**. Затем нажмите кнопку **“Опубликовать”**:



В меню дополнительных действий нажмите

</>

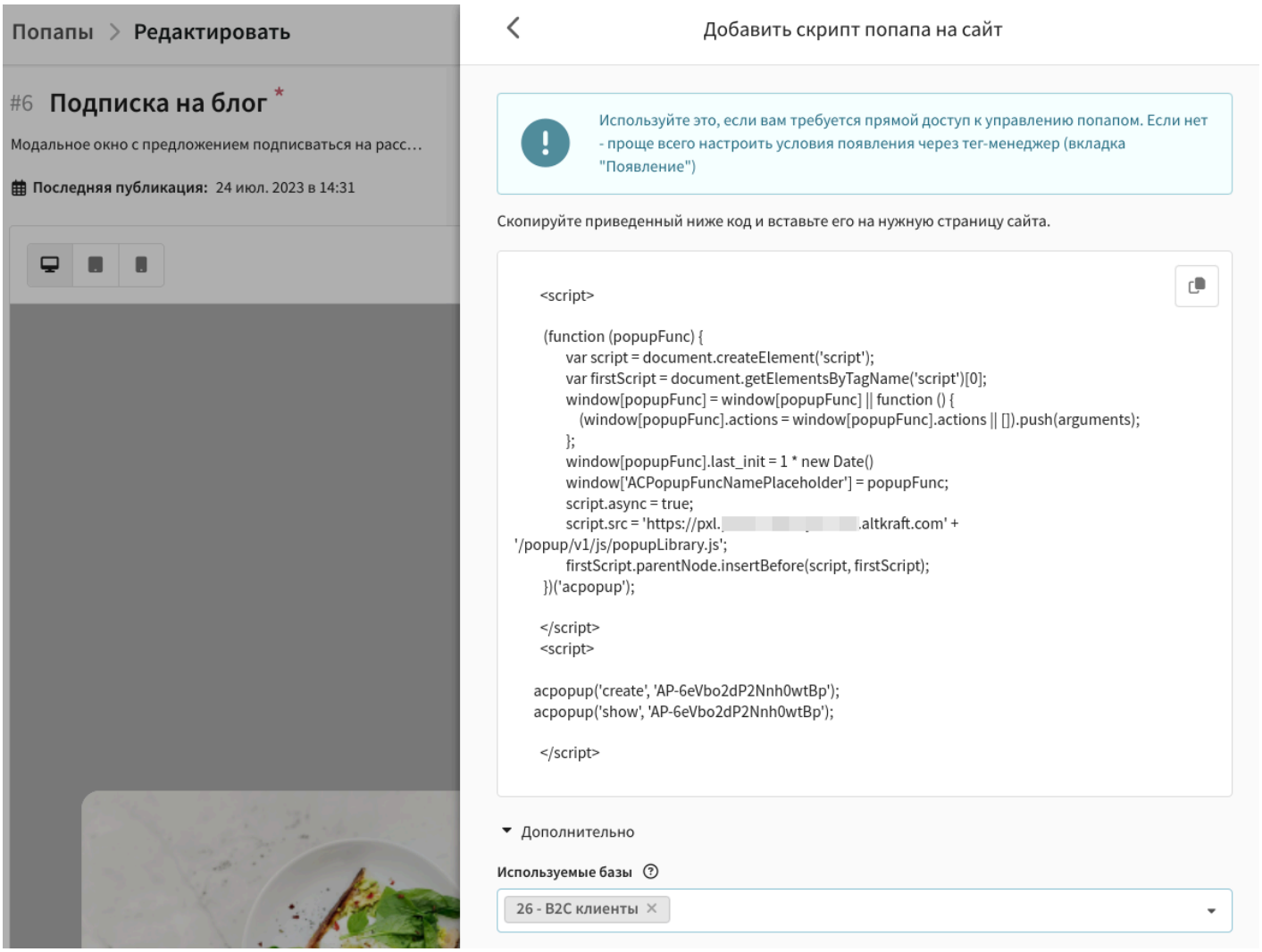
. Справа появится окно, где необходимо выбрать **базы для поиска профилей**. Данные для матчинга можно передать вместе с [запросом данных попапа с сервера]. Если профиль с такими данными будет найден в одной из выбранных баз, его данные обновятся; в противном случае — в базе будет создан новый профиль.

CAUTION

Обратите внимание, что здесь вы задаете базы для поиска существующих профилей. Данные пользователей, заполнивших форму в попапе, будут импортироваться в базу, указанную в разделе

“Действия”.

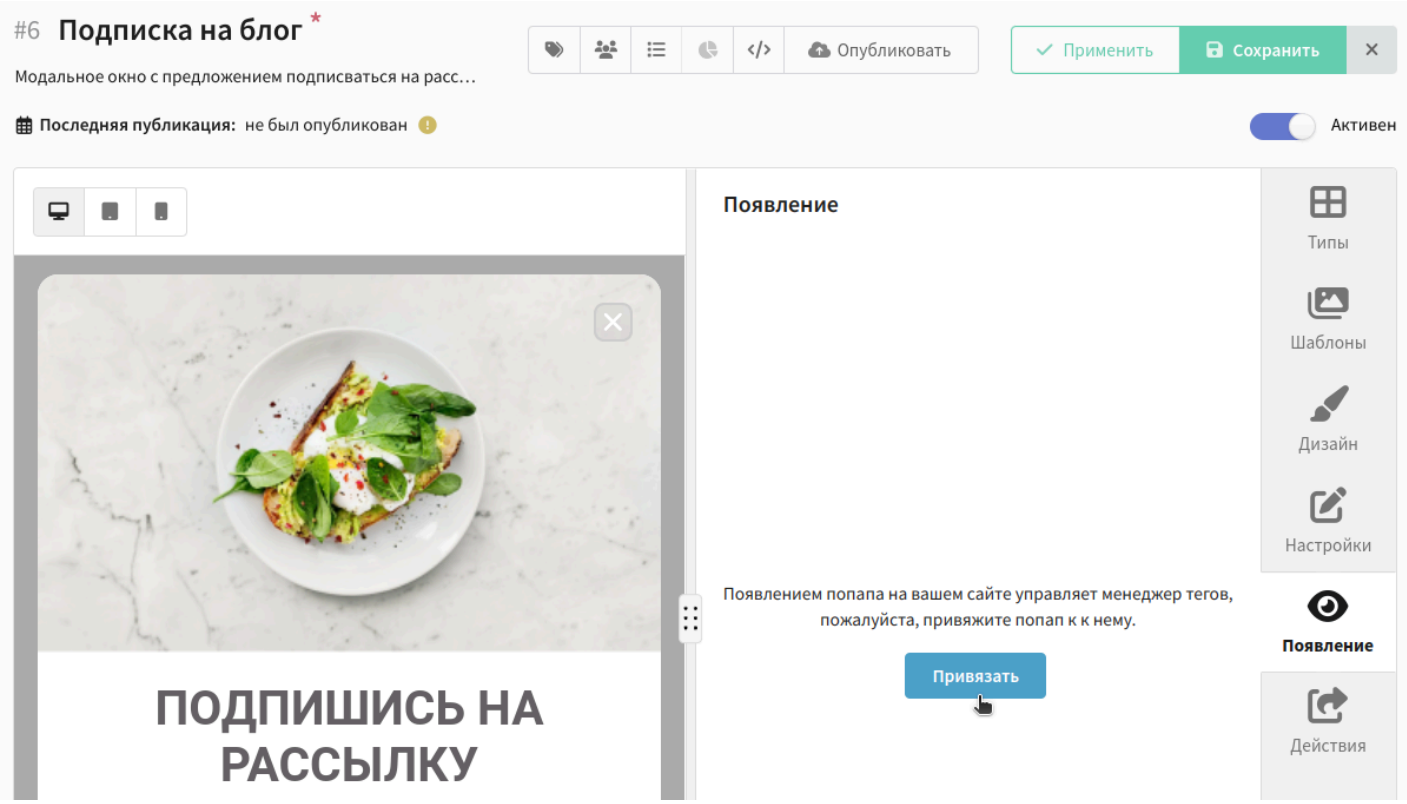
Нажмите кнопку **“Показать код”** внизу страницы. Скопируйте код попапа и передайте его разработчикам, чтобы они добавили его на нужные страницы сайта:



Добавление через Менеджер тегов

Привязка к контейнеру

Сохраните попап с помощью кнопки на верхней панели. После этого раздел **“Появление”** станет активным. Перейдите в него. Нажмите **“Привязать”**, чтобы привязать попап к контейнеру в Менеджере тегов:

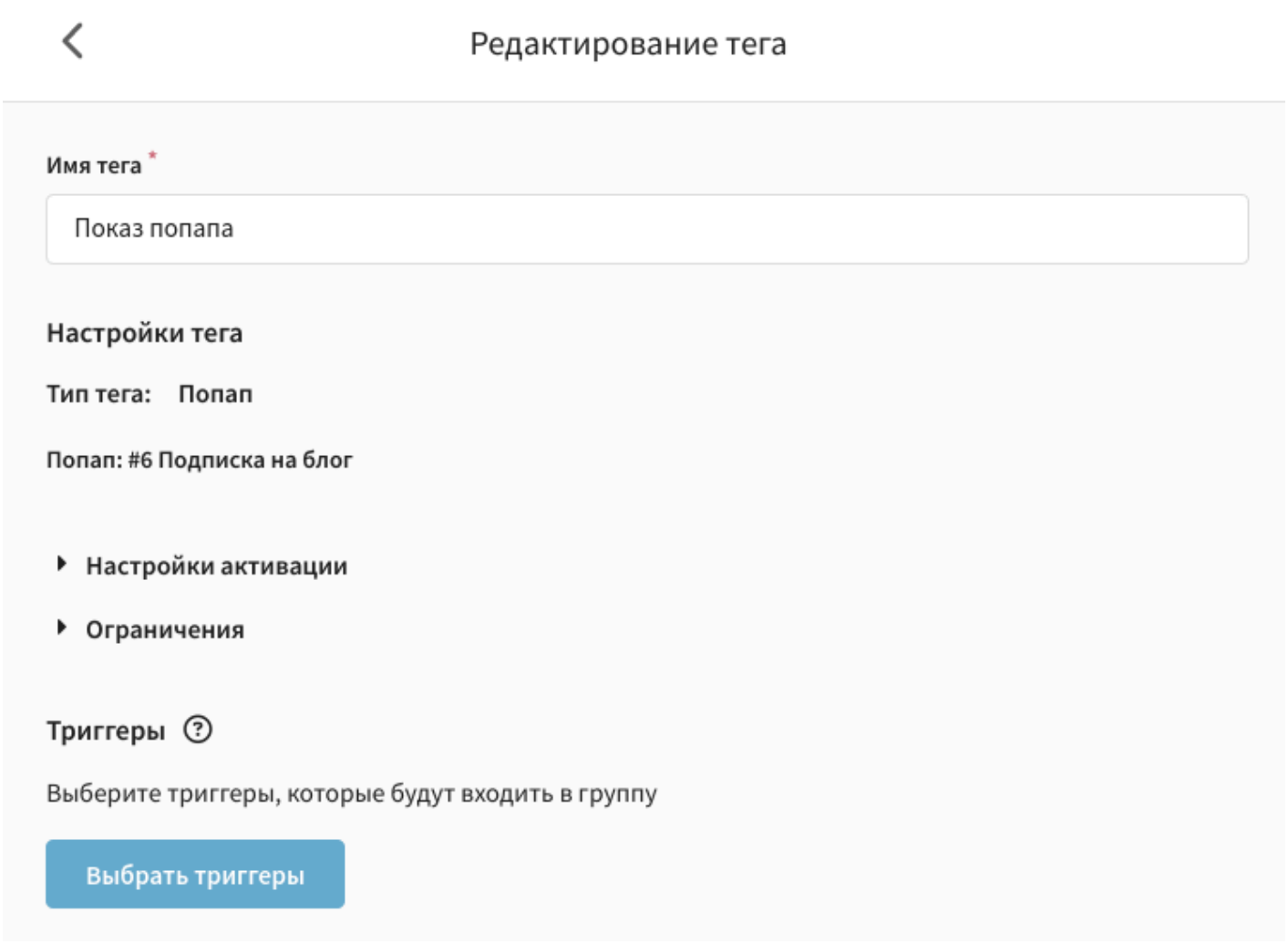


Выберите **контейнер** или создайте новый.

Введите **имя тега**. Тег — это фрагмент кода, который выполняется на сайте при определенных условиях. В данном случае тег — это показ попапа.

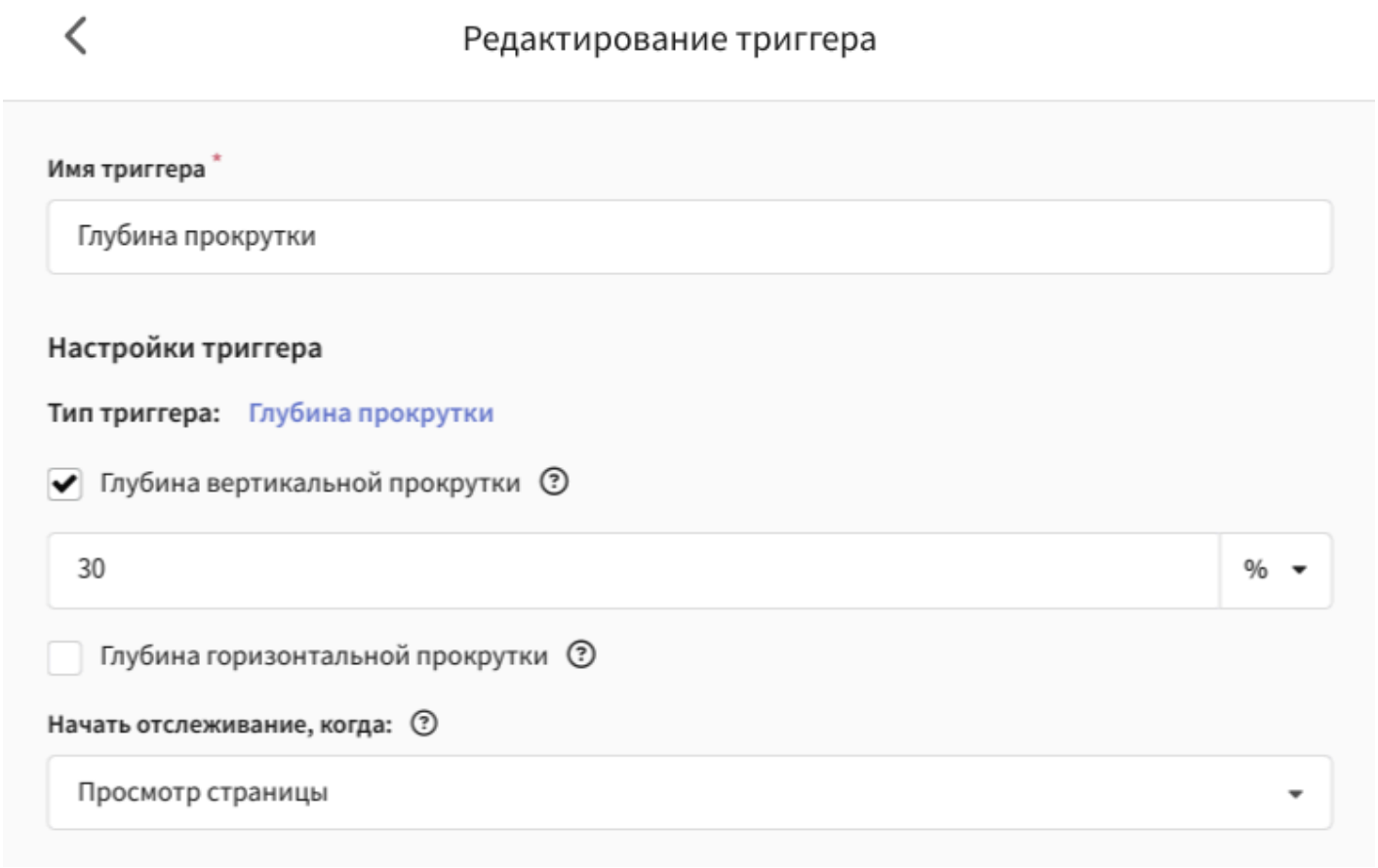
Настройка активации тега

Задайте **настройки активации** и **ограничения** для срабатывания тега:

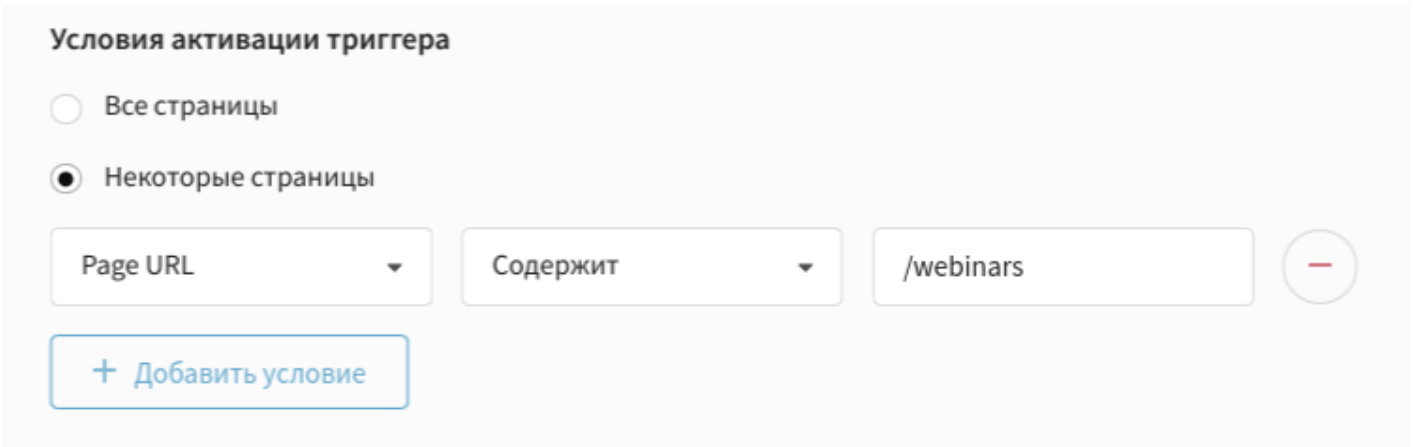


Настройка триггеров показа

Выберите **триггеры** или создайте новый. Триггер — это условие выполнения тега. Триггеры отслеживают определенные события на сайте (например, клики или загрузка страницы) и активируют теги в ответ на эти события. В данном случае триггер запустит показ попапа:

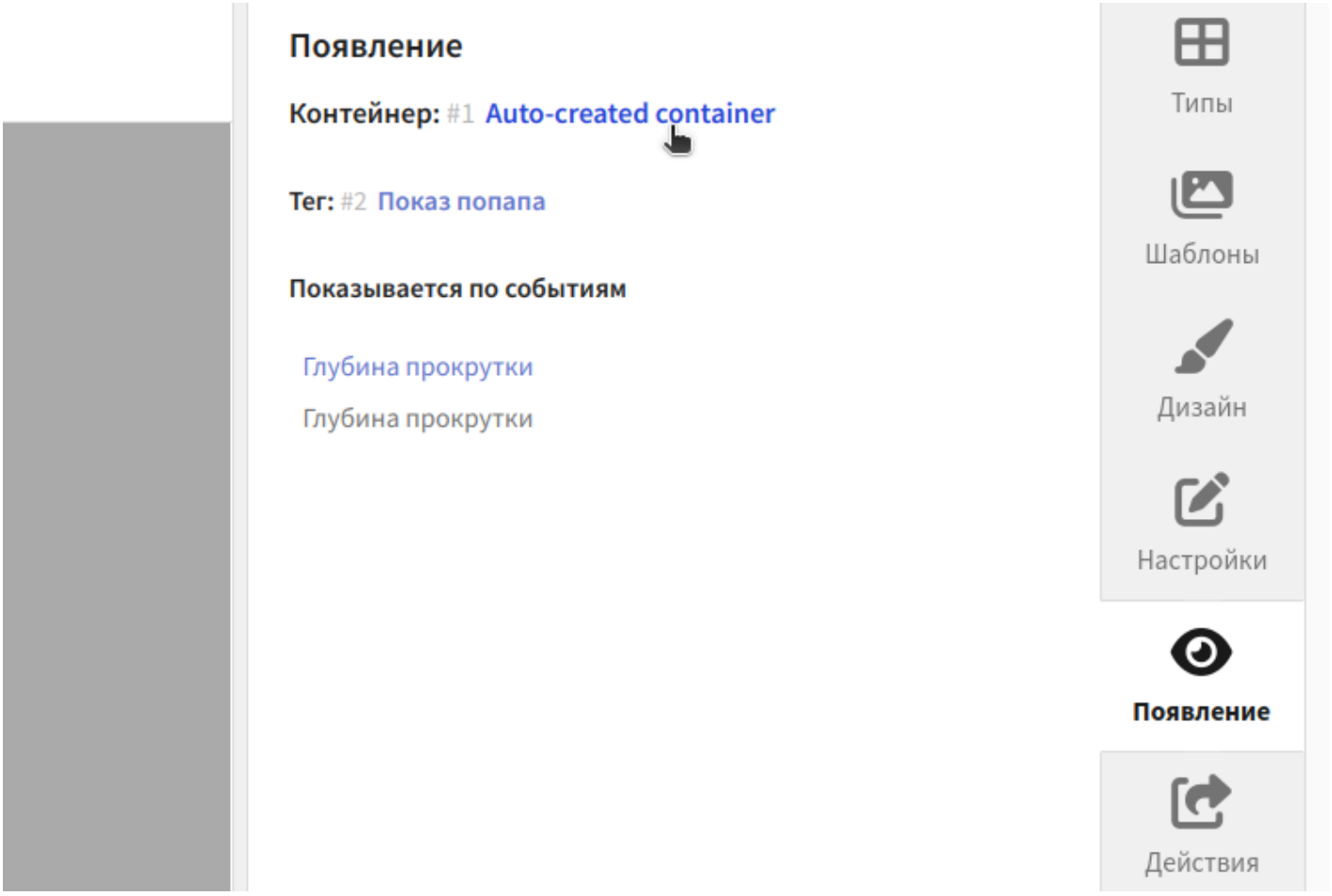


Чтобы задать более конкретные условия для появления попапа, используйте фильтр. Для этого в блоке **“Условия активации триггера”** выберите **“Некоторые события”** и настройте фильтр:

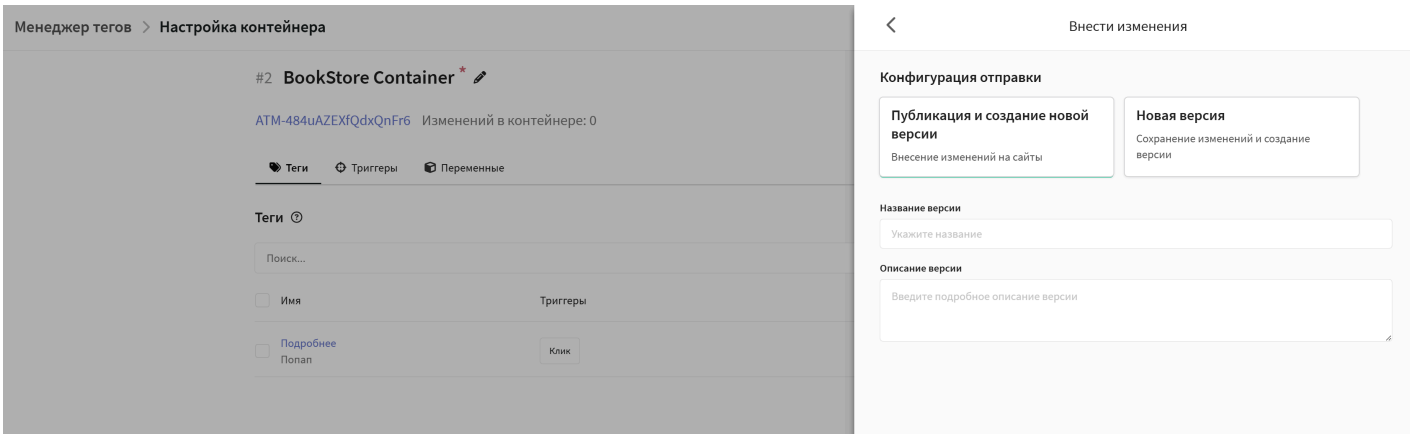


Публикация изменений

Сохраните тег. Заданное условие для показа попапа отобразится в разделе **“Появление”**:



Сохраните попап и **опубликуйте**. Перейдите в привязанный контейнер и **опубликуйте новую версию**. После этого попап будет показываться посетителям сайта при наступлении события, заданного в триггере:



TIP

Больше информации о том, как создать контейнер и настроить теги, триггеры и переменные в Менеджере тегов, вы найдёте в этой статье.

Матчинг профилей

Вы можете самостоятельно идентифицировать посетителя сайта и передать данные о нём в платформу с помощью метода `acDataLayer.push()`. Если система найдёт профиль в базе, то обновит его данные.

Пример:

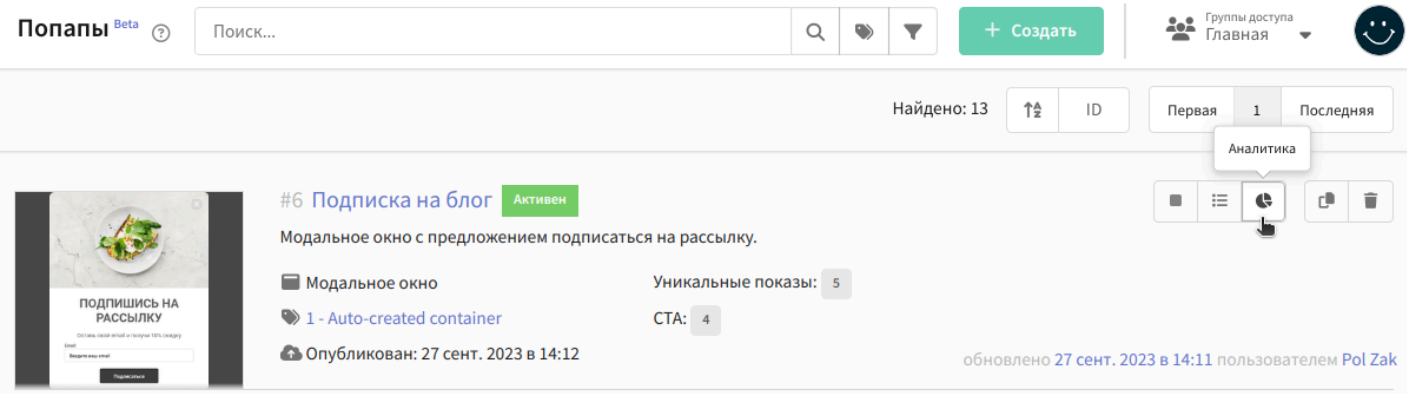
```
acDataLayer.push({ matching:{ matching_mode: 'email', matching: { email:
'test@example.com', db_ids: [1,2] } }})
```

- **matching_mode** — режим поиска профиля; может принимать значения: `"email"`, `"email_profile"`, `"phone"`, `"profile_id"` или `"custom"`;
- **matching** — объект с одним или двумя параметрами: `"поле для поиска": "значение поля"` и `"db_id": "ID базы данных"` — базы данных, в которых необходимо искать профиль.

Аналитика и отслеживание результатов

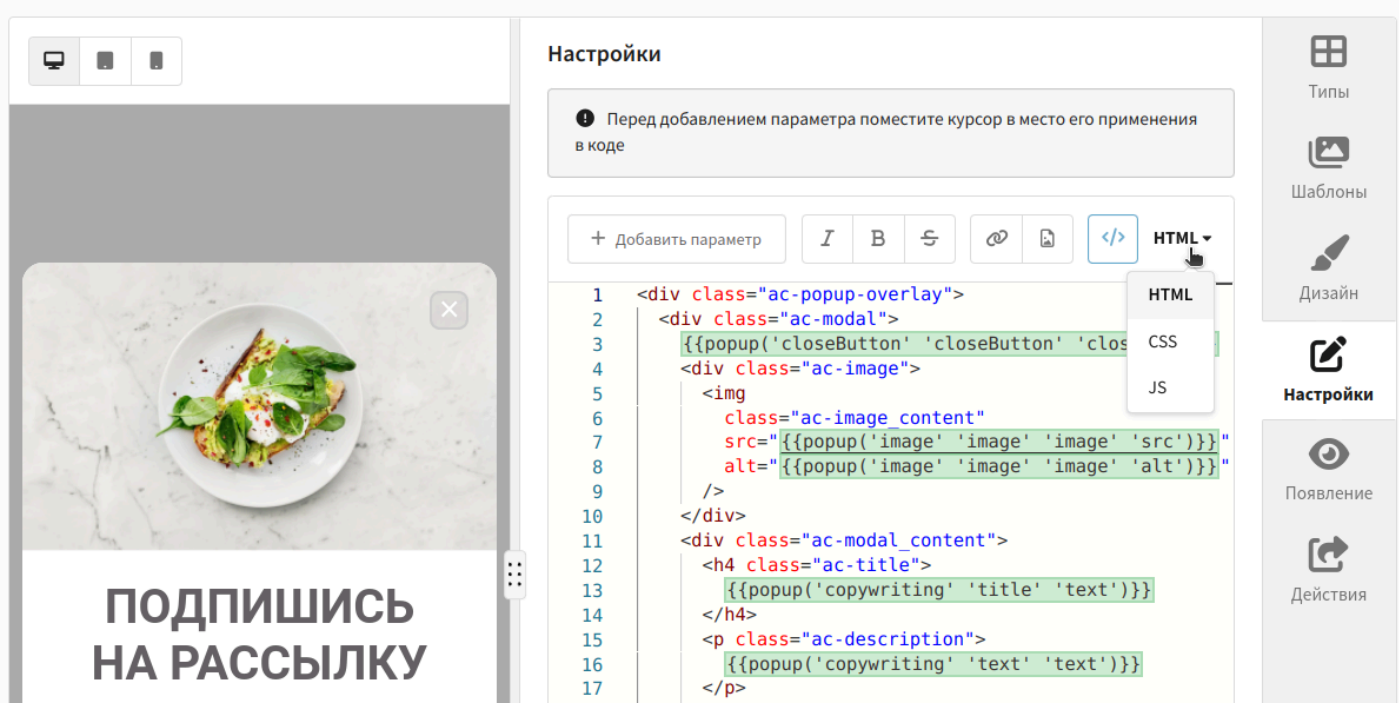
Проанализировать статистику по попапам и просмотреть результаты заполнения формы вы можете в разделе **Аналитика**. Для этого нажмите на иконку

в окне редактирования попапа или в общем списке попапов:



Настройка попапа в редакторе кода

Для работы напрямую с кодом попапа перейдите в раздел **“Настройки”**. В правом верхнем углу доступно переключение между HTML, CSS и JavaScript-кодом:



Для удобства работы в редакторе используйте следующие функции в панели инструментов:

B

— **жирный текст**

I

— *курсив*

~~S~~

— ~~зачёркнутый текст~~

— добавление или редактирование **ссылок**

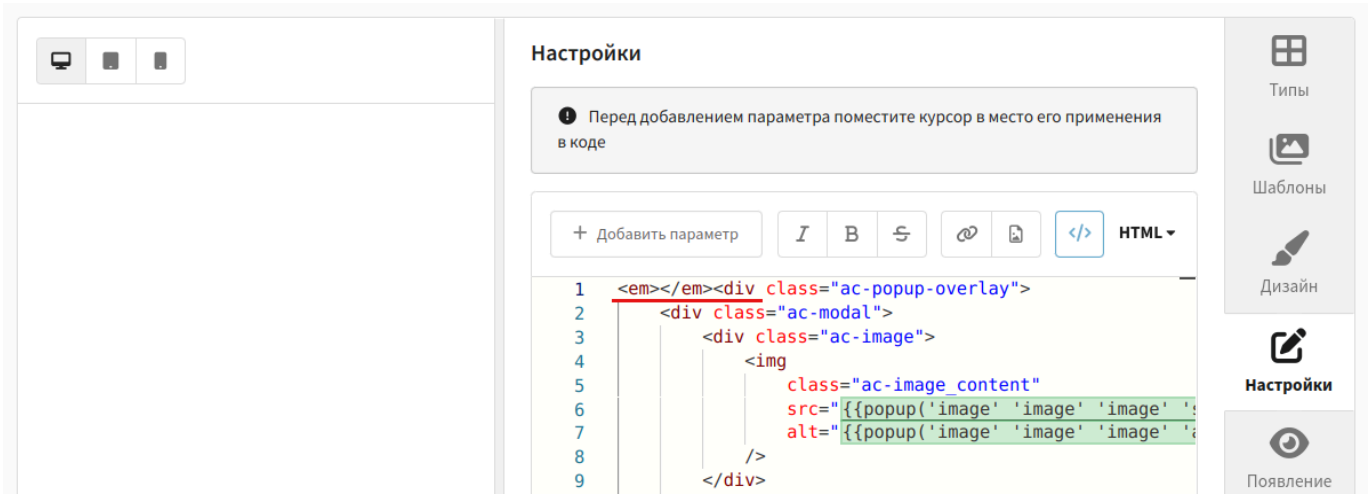
— добавление **изображений** из галереи платформы

</>

— список переменных и функций редактора. С помощью этой кнопки вы можете добавить в попап динамический контент.

INFO

Перед верхнеуровневым элементом `div` нельзя добавлять другие html-элементы, иначе превью шаблона станет недоступным.

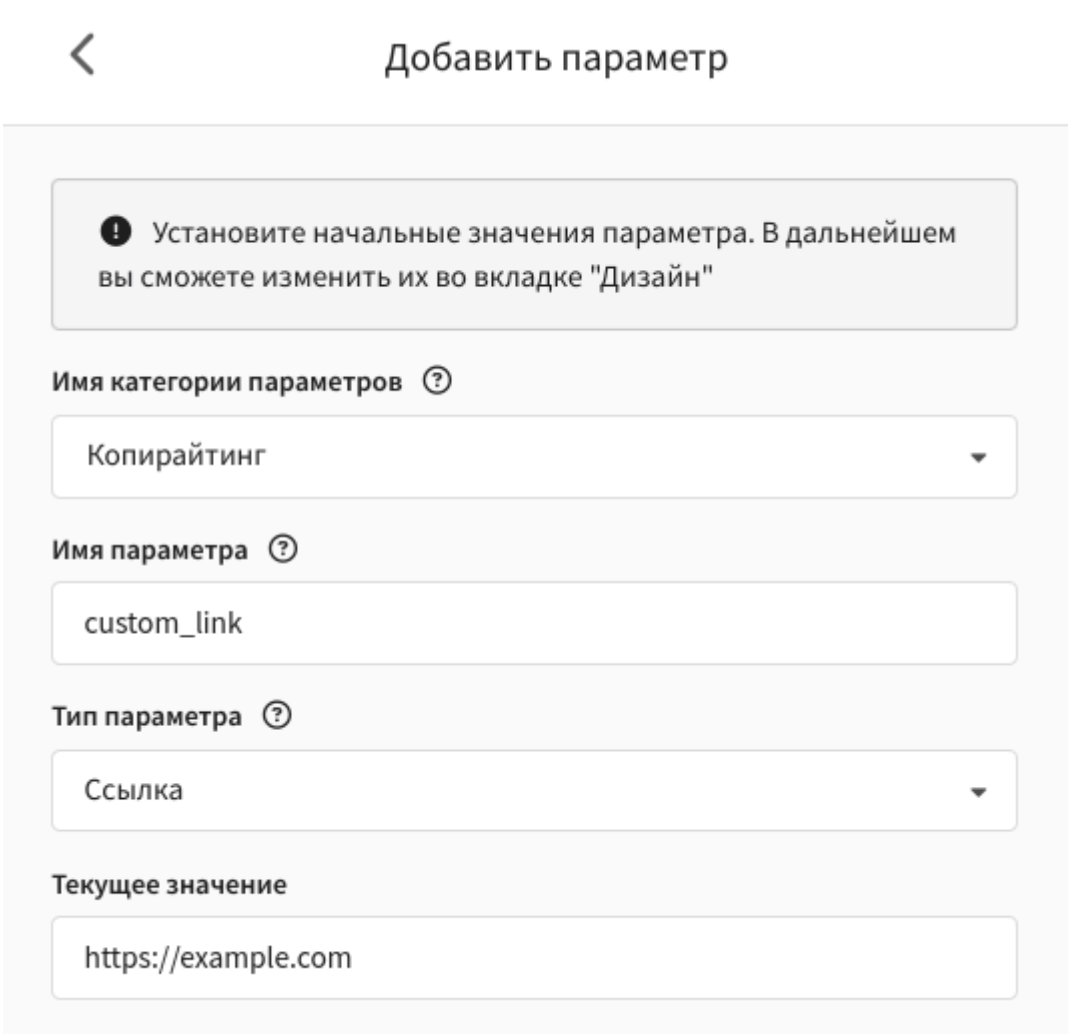


Создание параметра

В этом разделе также можно создать новые параметры. При сохранении они отобразятся на вкладке **“Дизайн”**. Таким образом, пользователи, не знакомые с кодом, смогут менять значения кастомных параметров в визуальном редакторе.

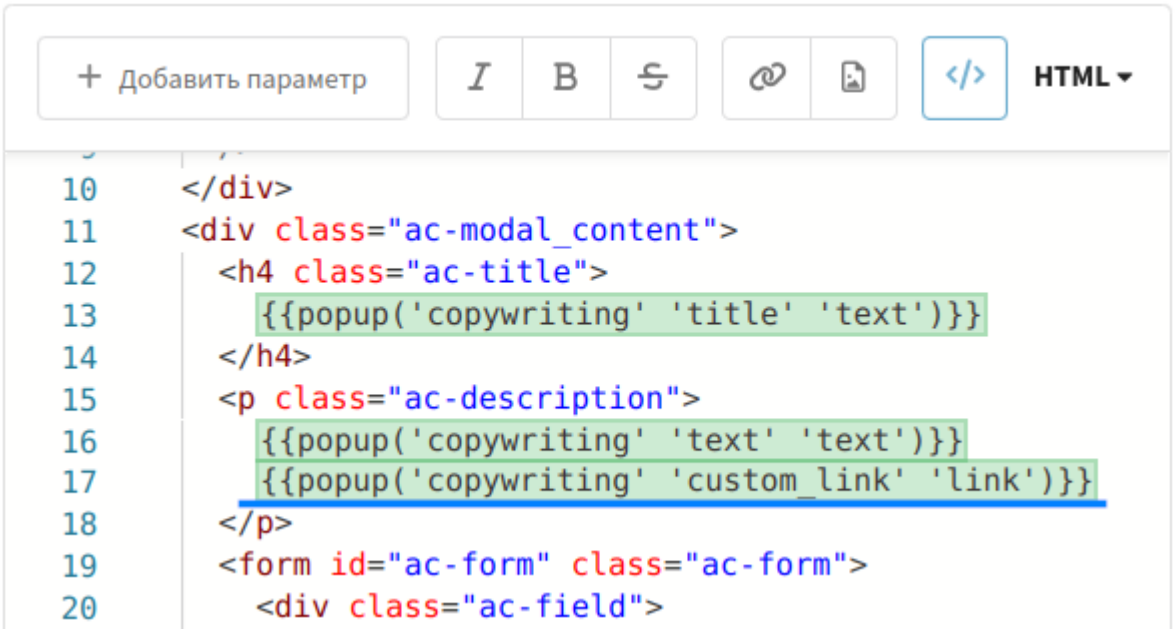
Чтобы создать новый параметр, установите курсор в строке редактора, куда хотите добавить параметр. Нажмите кнопку **“Добавить параметр”** в левом верхнем углу редактора. В новом окне выберите:

- **Имя категории параметров.** Встроенные параметры в визуальном редакторе сгруппированы по категориям. Вам необходимо выбрать одну из категорий для кастомного параметра. Также можно создать свою категорию. Для этого введите имя категории и нажмите Enter.
- **Имя параметра.** Введите название параметра, которое будет использоваться в редакторе. Имя должно быть уникальным в рамках категории.
- **Тип параметра.** Выберите тип значения, которое будет содержать параметр (например, текст, числовое значение, семейство шрифта, параметры поля для ввода и др.).
- **Текущее значение.** Укажите значение параметра по умолчанию. Это поле не является обязательным.

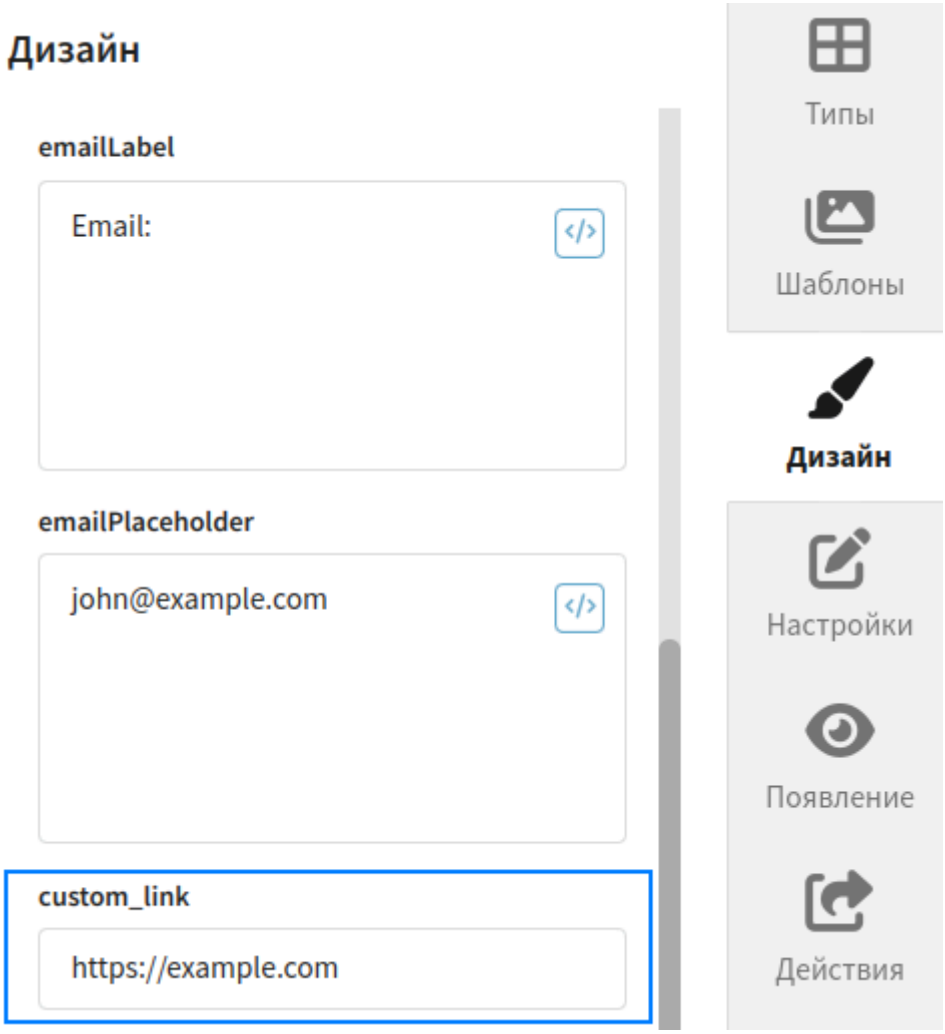


После настройки нажмите кнопку **“Вставить”** внизу страницы. Параметр добавится в следующем виде:

```
{{popup('<имя_категории>' '<имя_параметра>' '<тип_параметра>')}}
```



Созданный параметр появится в визуальном редакторе в выбранной категории. После сохранения попапа значение нового параметра смогут редактировать другие пользователи платформы:



Настройка элементов попапа

При создании попапа платформа предлагает множество готовых шаблонов, однако иногда необходимо внести изменения в существующий шаблон. Ниже собраны рекомендации, как это можно сделать.

INFO

В имени полей (атрибут `name`) до и после выражения не должно быть лишних символов:

Пример корректного использования — `name="{{popup('form' 'week-field' 'fieldData')}}"` Пример неправильного использования — `name="{{popup('form' 'week-field' 'fieldData')}}-"`

Перекрывающий слой

Перекрывающий слой предназначен для визуального выделения попапа. Определяется элементом с классом `ac-popup-overlay`. По умолчанию, этот элемент имеет полупрозрачный фон и занимает всю доступную область экрана.

Пример использования:

```
<div class="ac-popup-overlay">
    <!--Содержимое попапа-->
</div>
```

Текстовое наполнение

По умолчанию доступно два вида текста: заголовок и параграф.

Пример использования:

```
<!-- Заголовок -->
<h4 class="ac-title">
    {{popup('copywriting' 'title' 'text')}}
</h4>

<!-- Параграф -->
<div class="ac-description">
    {{popup('copywriting' 'text' 'text')}}
</div>
```

Изображения

Добавить изображение в попап можно двумя способами:

- в виде элемента с тегом `<img>`
- с помощью CSS-свойства `background-image`

Пример добавления изображения с помощью тега `<img>`:

```
<div class="ac-image">
    <img
        class="ac-image_content"
        src="{{popup('image' 'image' 'image' 'src')}}"
        alt="{{popup('image' 'image' 'image' 'alt')}}"
    />
</div>
```

Пример добавления изображения с помощью CSS-свойства `background-image`:

```
background-image: url("{{popup('background' 'backgroundImage' 'image' 'src')}}");
```

Форма

Для корректной работы попапа все поля для ввода и кнопка отправки формы должны размещаться в элементе с тегом `<form>`.

Пример использования:

```
<form id="ac-form" class="ac-form">
    <!-- Содержимое формы -->
</form>
```

Поля для ввода

Для корректного отображения каждое поле для ввода должно находиться внутри элемента с классом `ac-field`. Для добавления заголовка рекомендуется использовать элемент с классом `ac-label`. Обязательно укажите атрибут `name`, если он не будет указан, то значение поля ввода не отправится на сервер и не сохранится в платформе.

Если предполагается использование правил валидации, необходимо добавить элемент с классом `ac-validation-msg`.

Пример поля для ввода email:

```
<div class="ac-field">
    <label class="ac-label" for="email">
        {{popup('copywriting' 'emailLabel' 'text')}}
    </label>
    <input
        id="email"
        type="email"
        class="ac-input"
        name="{{popup('form' 'email' 'fieldData')}}"
        placeholder="{{popup('copywriting' 'emailPlaceholder' 'text')}}"
    />
    <div class="ac-validation-msg" for="email"></div>
</div>
```

Пример поля для ввода телефона:

```
<div class="ac-field">
    <label class="ac-label" for="phone">
        {{popup('copywriting' 'phoneLabel' 'text')}}
    </label>
    <input
        id="phone"
        type="tel"
        class="ac-input"
        name="{{popup('form' 'phone' 'fieldData')}}"
        placeholder="{{popup('copywriting' 'phonePlaceholder' 'text')}}"
    />
    <div class="ac-validation-msg" for="phone"></div>
</div>
```

CAUTION

Для корректного сбора статистики для полей ввода следует установить атрибут `type`, значение которого будет соответствовать типу вводимых данных.

Доступны следующие типы полей:

- `<input type="text">` — базовое текстовое поле в виде строки.
- `<input type="email">` — поле для адреса электронной почты. Перед отправкой проверяется, что входное значение содержит либо пустую строку, либо один действительный адрес электронной почты.
- `<input type="tel">` — поле для ввода номера телефона; разрывы строк автоматически удаляются из входного значения.
- `<input type="number">` — поле ввода для числа формата float.
- `<input type="search">` — однострочное текстовое поле для ввода строк поиска.
- `<input type="url">` — поле для редактирования URL. Введённое значение должно содержать либо пустую строку, либо допустимый абсолютный URL.
- `<input type="color">` — элемент управления цветом.
- `<input type="date">` — элемент управления для ввода даты (год, месяц и день, без времени).
- `<input type="datetime">` — элемент управления для ввода даты и времени (час, минута, секунда и доля секунды) в соответствии с часовым поясом UTC.
- `<input type="datetime-local">` — элемент управления для ввода даты и времени без часового пояса.
- `<input type="week">` — элемент управления для ввода даты, содержащей число неделя-год и номер недели без часового пояса.
- `<input type="month">` — элемент управления для ввода месяца и года без часового пояса.

Кнопка отправки формы

Для создания кнопки отправки формы рекомендуется использовать элемент с классом `ac-form_submit`, который содержит элемент кнопки с типом `submit` и классом `ac-button`.

Пример использования:

```
<div class="ac-form_submit">
    <button type="submit" class="ac-button">
        {{popup('cta' 'text' 'text')}}
    </button>
</div>
```

Кнопка закрытия

Чтобы создать кнопку закрытия попапа, рекомендуется использовать функцию popup, передав в неё соответствующие параметры.

Пример использования:

```
{{popup('closeButton' 'closeButton' 'closeBtn')}}
```

Элемент имеет несколько различных вариантов, посмотреть которые можно на вкладке **Дизайн —> Кнопка “Закрыть”**.

Дополнительное отслеживание событий

Для трекинга событий необходимо в html-элемент, с которым связано событие, добавить атрибут `data-tracking-event="[event-name]"`. Вместо `[event-name]` укажите название события. Поддерживаются следующие события:

- `click` — клик на элемент
- `close` — закрытие попапа

Пример регистрации кликов по элементу попапа:

```
<div data-tracking-event="click"></div>
```

TIP

Если попап содержит html-элемент form, то при нажатии на кнопку отправки формы автоматически будет регистрироваться действие заполнения формы. То есть в данном случае для регистрации события не требуется добавлять дополнительный атрибут на кнопку формы.

Управление попапами вручную через скрипт

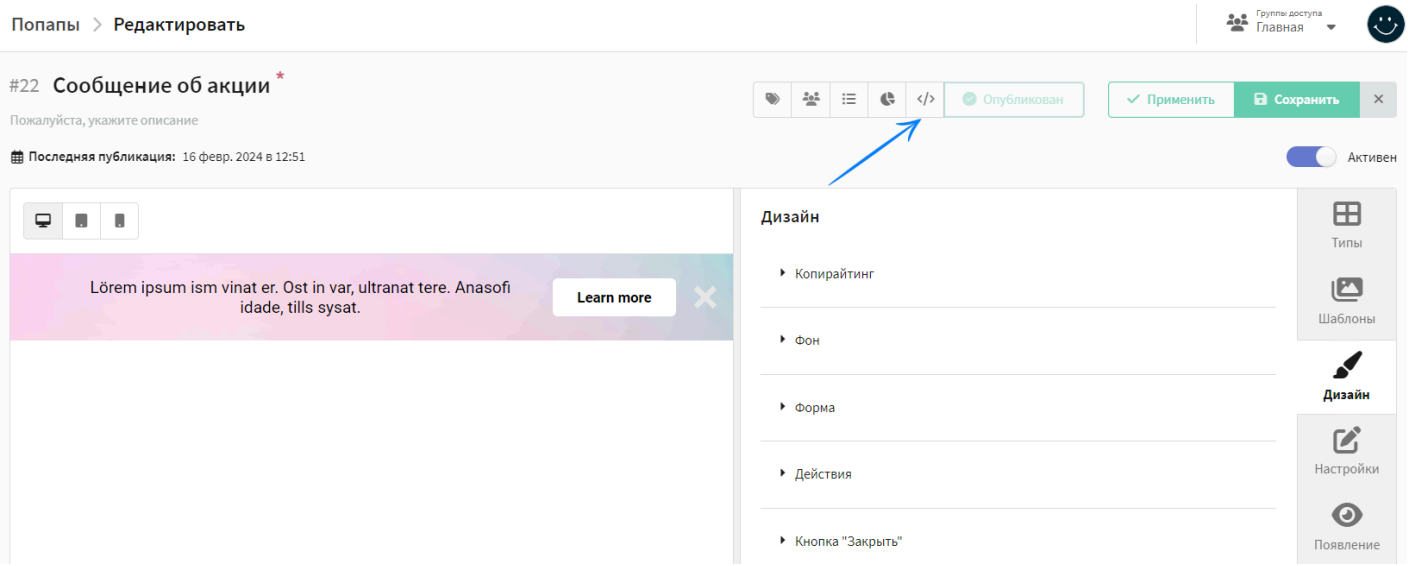
Данная инструкция предназначена для пользователей платформы, которые размещают попап на сайте через скрипт, а не через Менеджер тегов Altcraft.

Получение скрипта

Для получения скрипта перейдите в редактор попапа и в меню дополнительных действий нажмите

</>

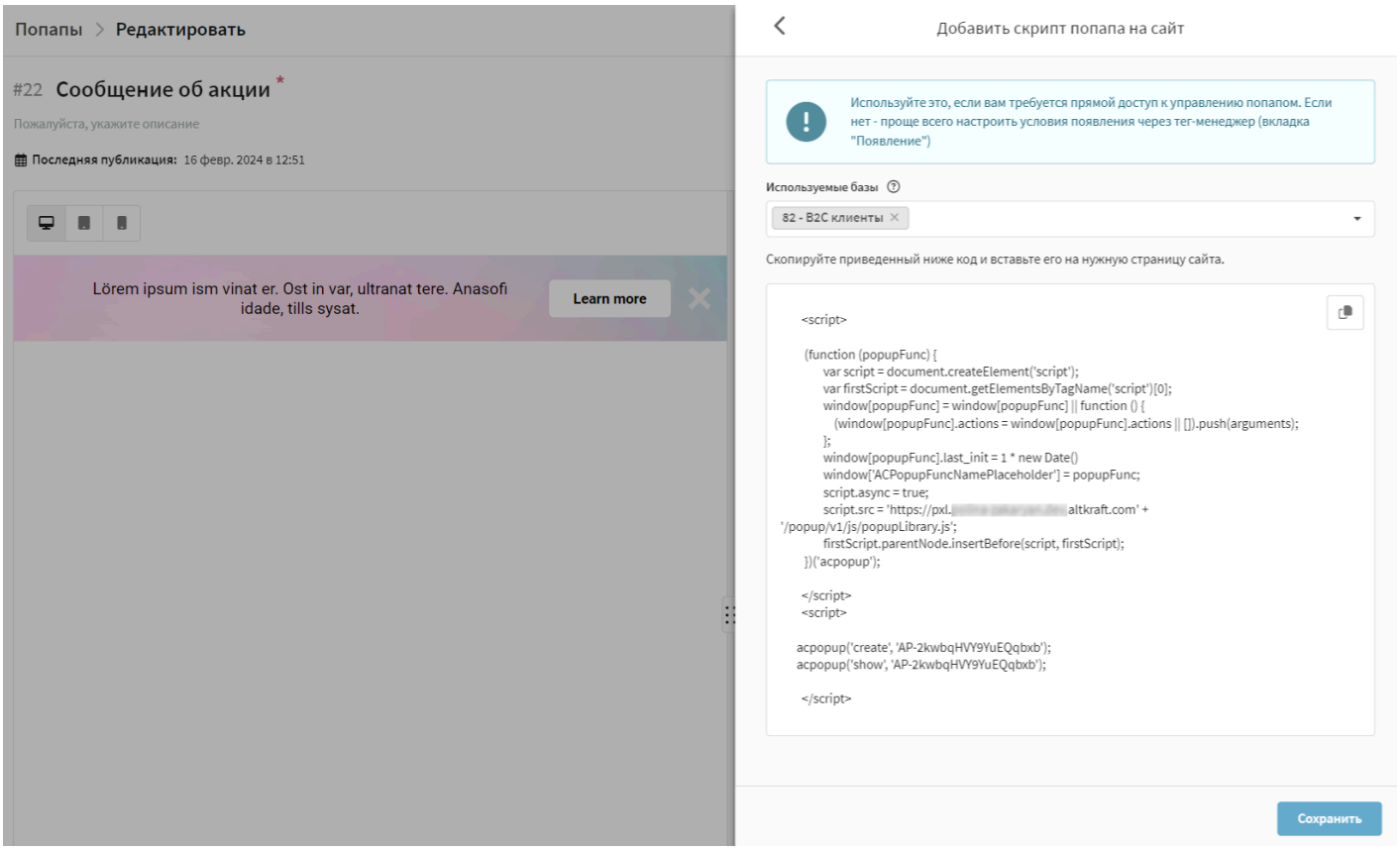
:



В появившемся окне выберите **базы для поиска профилей**. Когда пользователь заполняет форму попапа, его данные передаются в платформу. Затем платформа проверяет, есть ли профиль с такими данными в одной из выбранных баз. Если есть — данные существующего профиля обновляются; если нет — в базе создается новый профиль.

Обратите внимание, что здесь вы задаете базы для поиска существующих профилей. Данные пользователей, заполнивших форму в попапе, будут импортироваться в базу, указанную в редакторе попапа в разделе **“Действия”**.

Нажмите кнопку **“Показать код”**. Этот код необходимо разместить на тех страницах сайта, где вы хотите показывать попап. Ниже вы найдете более подробную информацию о том, как настроить логику появления попапа.



Управление через скрипт

Для работы с попапами на сайте используется следующий код:

```
<!-- Загрузка скрипта на страницу -->
<script>

  (function (popupFunc) {
        var script = document.createElement('script');
        var firstScript = document.getElementsByTagName('script')[0];
        window[popupFunc] = window[popupFunc] || function () {
            (window[popupFunc].actions = window[popupFunc].actions || []).push(arguments);
        };
        window[popupFunc].last_init = 1 * new Date()
        window['ACPopupFuncNamePlaceholder'] = popupFunc;
        script.async = true;
        script.src = 'https://pxl.altkraft.com' + '/popup/v1/js/popupLibrary.js';
        firstScript.parentNode.insertBefore(script, firstScript);
  })('acpopup');

</script>

<!-- Запрос данных попапа с сервера + показ попапа на странице -->
<script>

  acpopup('create', '<popup_id>');
  acpopup('show', '<popup_id>');

</script>
```

При установке этого кода на сайт посетитель увидит попап сразу после загрузки страницы из-за наличия скрипта:

```
<script>
  acpopup('create', '<popup_id>');
  acpopup('show', '<popup_id>');
</script>
```

Вам необходимо самостоятельно реализовать условия создания и появления попапа.

Пример появления попапа через 10 секунд после загрузки страницы:

```
<script>
    window.onload = function() {
        setTimeout(function() {
            // Запрос данных попапа с сервера
            acpopup('create', '<popup_id>');

            // Показ попапа
            acpopup('show', '<popup_id>');
        }, 10000);
    };
</script>
```

Пример появления попапа при клике на кнопку:

```
<button onclick="showPopupClick()">Показать попап</button>

<script>
    function showPopupClick() {
        acpopup('create', '<popup_id>');
        acpopup('show', '<popup_id>');
    }
</script>
```

Для управления попапом используйте функцию `acpopup`. Она работает в 3 режимах:

- create
- show
- sendEvent

create — запросить данные попапа с сервера

В функцию `acpopup` вторым аргументом требуется передать **ID попапа**. Третьим аргументом можно передать **объект с информацией о матчинге** и **SMID**. В таблице описана структура этого объекта:

Параметр	Тип	Обязательный	Описание
`matching_mode`	string	Нет	Режим поиска профиля. Может принимать значения: `"email"`, `"email_profile"`, `"phone"`, `"profile_id"` или `"custom"`.
`matching`	JSON object	Нет	Объект с одним или двумя параметрами: `"поле для поиска": "значение поля"` и `"db_id": "ID базы"` — массив баз данных, в которых необходимо искать профиль. `db_id` можно не указывать, если база профилей выбрана в настройках попапа.
`smid`	string	Нет	Уникальный идентификатор отправки сообщения профилю.

Пример:

```
acpopup('create', 'AP-6eVbo2dP2Nnh0wtBp', {
    matching_mode: 'email',
    matching: {
        email: 'john@example.com',
        db_ids: [1]
    },
    smid: 'send_message_id',
})
```

show — показать попап пользователю

В функцию `acpopup` вторым аргументом требуется передать **ID попапа**. Третьим аргументом может быть **объект с функцией** `onShow`, которая будет вызвана после появления попапа на экране.

Пример:

```
acpopup('show', 'AP-6eVbo2dP2Nnh0wtBp', {
  onShow: function () {
      console.log('Look at the screen')
  },
})
```

sendEvent — отправить пользовательское событие попапа

В функцию `acpopup` вторым аргументом требуется передать **ID попапа**. Третьим аргументом должен быть **объект с данными о пользовательском событии**. Также в этот объект можно передать **матчинг** для конкретного события и **SMID**. В таблице описана структура объекта:

Параметр	Тип	Обязательный	Описание
`web_event_name`	string	Да	Имя события
`web_event_value`	int	Нет	Числовое значение для события
`web_event_data`	JSON object	Нет	Объект с произвольными данными для события

Параметр	Тип	Обязательный	Описание
`matching_mode`	string	Нет	Режим поиска профиля. Может принимать значения: `"email"`, `"email_profile"`, `"phone"`, `"profile_id"` или `"custom"`.
`matching`	JSON object	Нет	Объект с одним или двумя параметрами: `"поле для поиска": "значение поля"` и `"db_id": "ID базы"` — базы данных, в которых необходимо искать профиль. `db_id` можно не указывать, если база профилей выбрана в настройках попапа.
`smid`	string	Нет	Уникальный идентификатор отправки сообщения профилю.

Пример:

```
acpopup('sendEvent', 'AP-6eVbo2dP2Nnh0wtBp', {
    web_event_name: 'new_event',
    web_event_value: 500,
    web_event_data: {
        status: 'active'
    },
    matching_mode: 'email',
    matching: {
        email: 'john@example.com',
        db_ids: [1]
    },
    smid: 'send_message_id',
})
```

TIP

Если вы добавляете на одну страницу несколько попапов, то для управления ими достаточно один раз разместить код загрузки скрипта на страницу, а в дальнейшем пользоваться функцией `acpopup`.

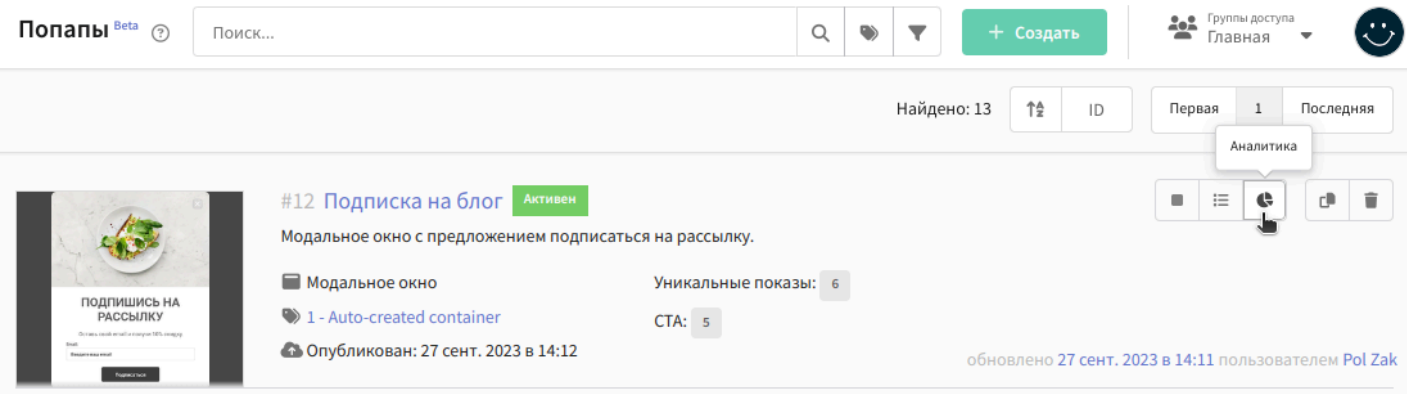
Аналитика попапов

Описание

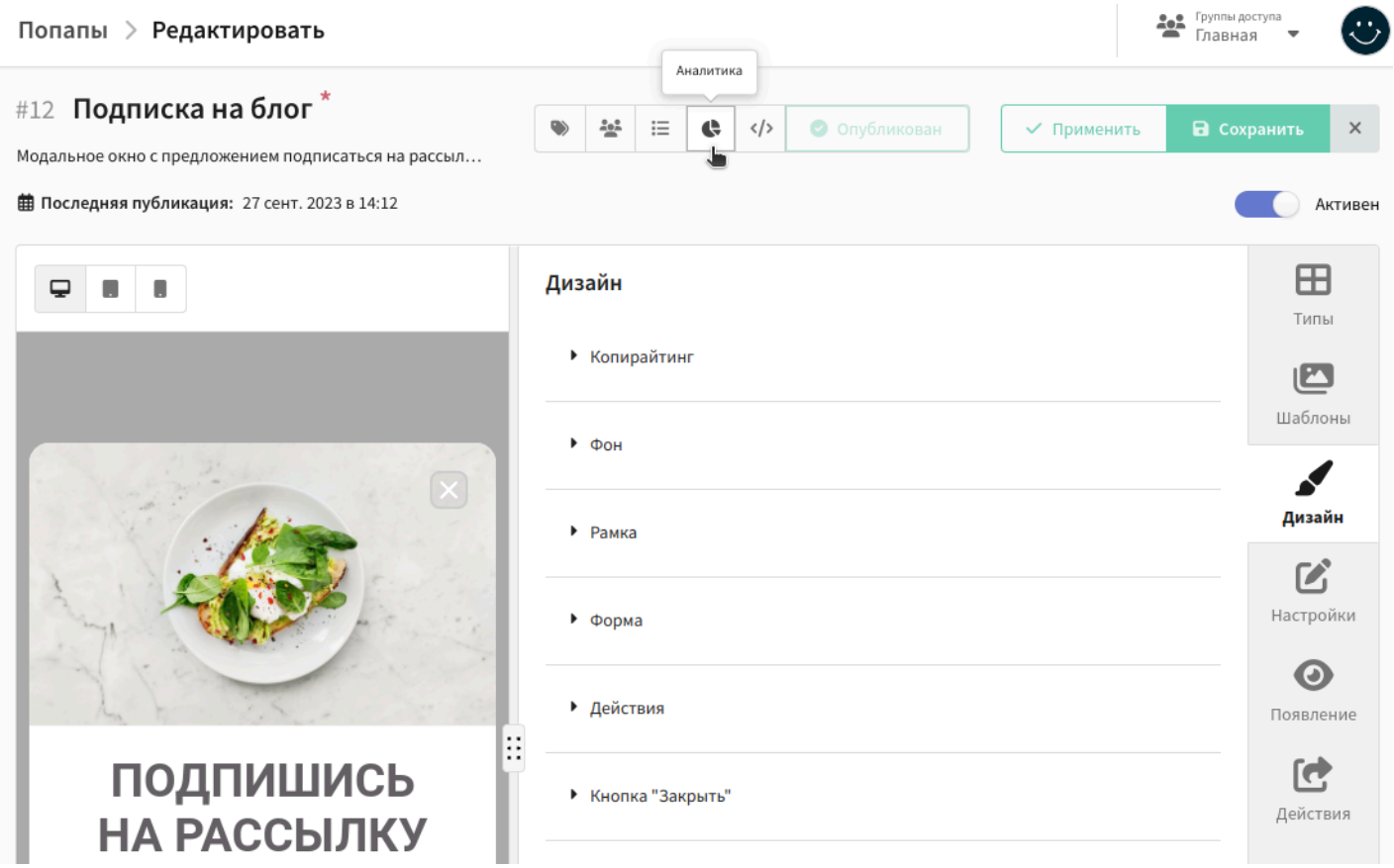
Аналитические инструменты позволяют просмотреть ключевые показатели попапа (попадание в таргетинг, показы, заполнения формы), а также ответы пользователей. Данные представлены в виде графика и в виде воронки. Любой показатель можно сгруппировать по различным критериям, например, по гео-данным пользователя, по браузеру или UTM-меткам.

Чтобы перейти в **Аналитику**, найдите нужный попап в общем списке и нажмите на иконку

:



Также можно перейти в Аналитику непосредственно из окна редактирования попапа:



В аналитике доступны две вкладки:

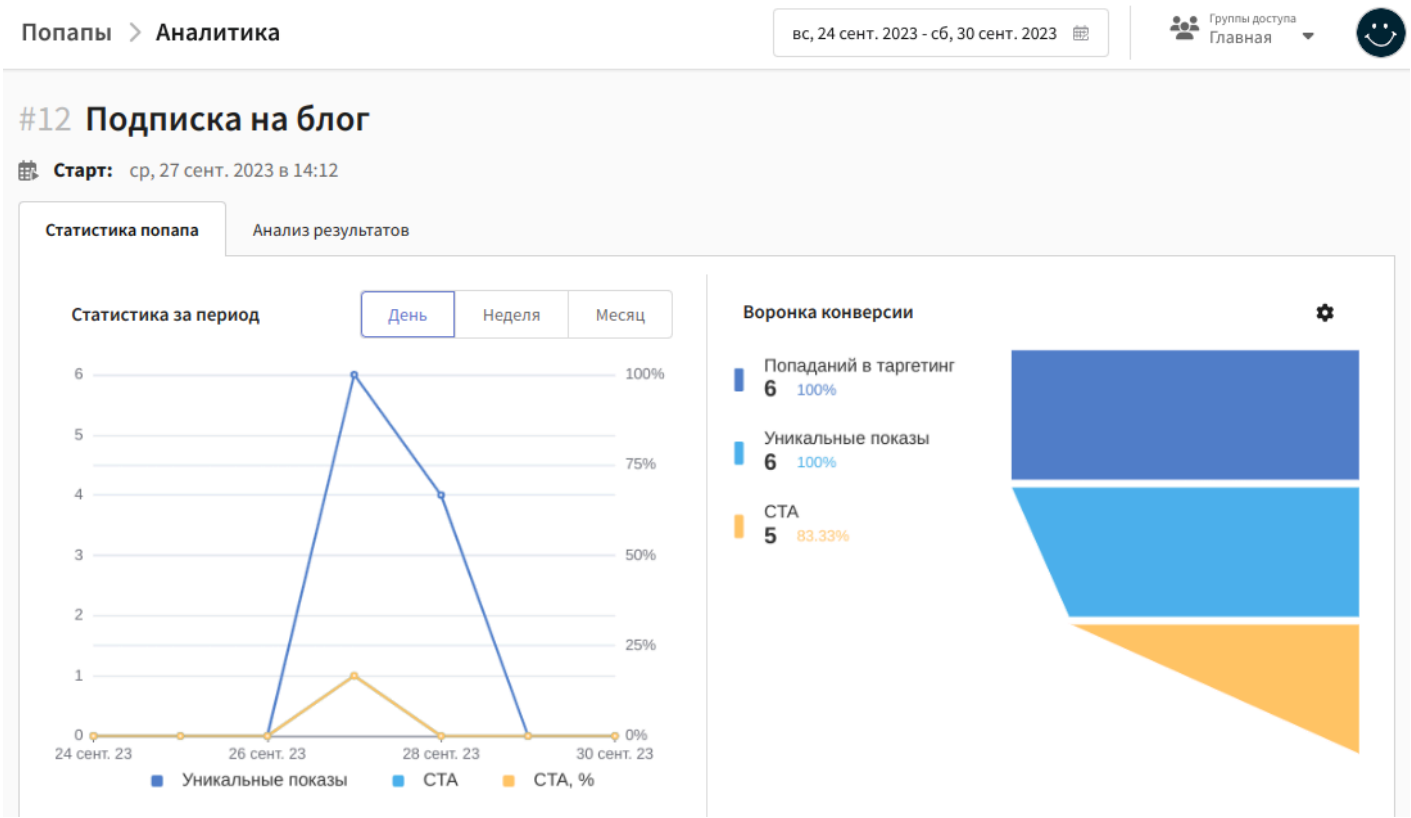
- **Статистика попапа** — общая сводка по попапу.
- **Анализ результатов** — статистика по заполнению полей в форме.

Статистика попапа

На этой вкладке собрана информация о показах попапа и заполнениях формы. На верхней панели вы можете указать период, за который хотите получить данные.

Статистика попапа разделена на три блока:

1. Статистика за период
2. Воронка конверсии
3. Аналитические срезы



Статистика за период

График отображает динамику изменения показателей за выбранный период:

- **Уникальные показы** — количество показов попапа пользователям. Если попап показан пользователю несколько раз в рамках одной сессии, фиксируется только один уникальный показ.
- **CTA** — пока событие отправки формы не зарегистрировано, показатель отображает клики на любой элемент формы. Если форма была хотя бы один раз заполнена и отправлена, показатель отображает количество заполнений.
- **CTA, %** — (Уник. показы / CTA) * 100%.

Данные можно сгруппировать по дням, неделям или месяцам. Если хотите скрыть показатель с графика, нажмите на его название.

Воронка конверсии

Воронка представляет собой последовательность шагов, по которым проходит группа посетителей сайта. Эта группа определяется первым шагом воронки — попадание в таргетинг. Далее отчёт показывает, какое количество клиентов перешло на следующий этап. С каждым шагом количество пользователей уменьшается.

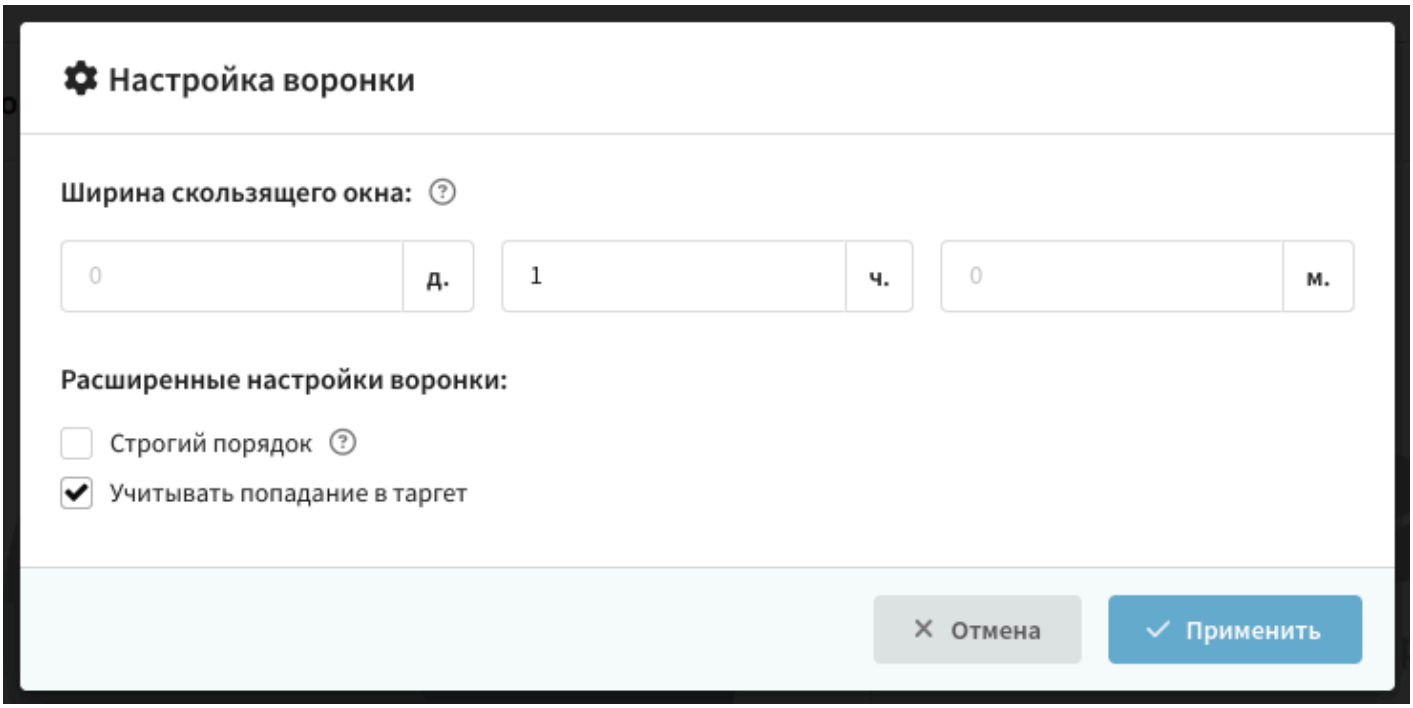
- **Попаданий в таргетинг** — количество раз, когда были выполнены условия для появления попапа, заданные в триггере. При этом попап мог быть не показан пользователю из-за лимитов тега, заданных в блоках “Настройки активации” и “Ограничения”.
- **Уникальные показы** — количество показов попапа пользователям.
- **CTA** — пока событие отправки формы не зарегистрировано, показатель отображает клики на любой элемент формы. Если форма была хотя бы один раз заполнена и отправлена, показатель отображает количество заполнений.

TIP

События в воронке являются **уникальными**. Например, если пользователю попап был показан несколько раз в рамках одной сессии, будет засчитан только один уникальный показ.

Дополнительно вы можете настроить воронку под свои задачи. Чтобы открыть окно настройки, нажмите

в правом верхнем углу.



Ширина скользящего окна — это временный период между первым и последним условиями в воронке. По умолчанию — 1 час. Если с момента попадания в таргетинг до момента регистрации события CTA пройдет больше 1 часа, то последний шаг воронки не будет засчитан.

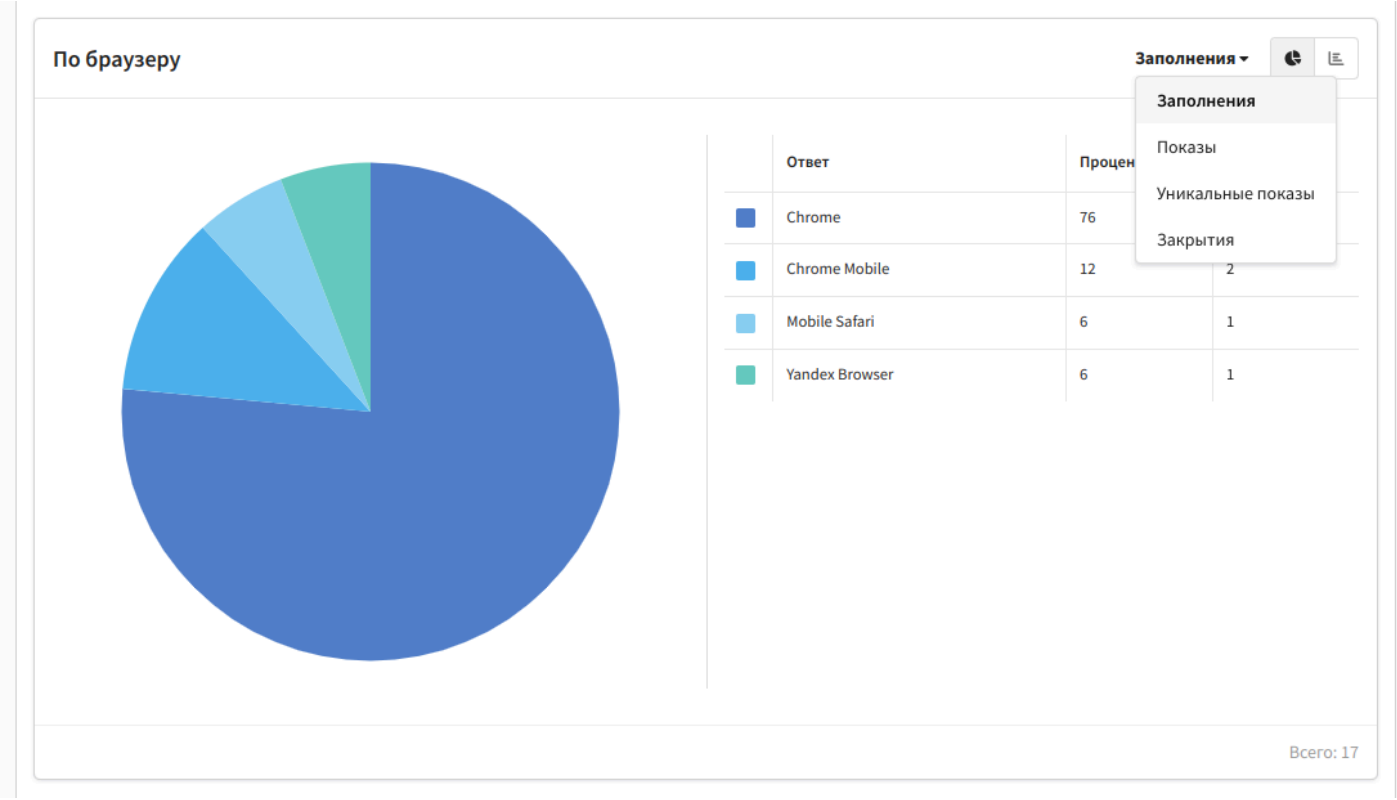
Строгий порядок запрещает посторонние события в искомой последовательности. Например, если в воронке с последовательностью *Попадание в таргетинг —> Уник. показ —> CTA* между первым и вторым шагом будет зафикисровано еще одно событие, то все последующие события не будут засчитаны. При этом событие, которые нарушило порядок, тоже не будет засчитано.

Учитывать попадание в таргет — сняв галочку, вы уберете попадание в таргетинг как шаг воронки.

Аналитические срезы отчёта

Ниже представлены показатели попапа, сгруппированные по следующим критериям:

- По часовому поясу
- По городам
- По странам
- По языку (язык браузера)
- По ОС (операционная система на устройстве пользователя)
- По браузеру
- По рефереру (URL страницы, на которой был показан попап)
- По UTM меткам (content, medium, source, campaign, term)



Для любого среза можно выбрать конкретный показатель:

- **CTA** — пока событие отправки формы не зарегистрировано, показатель отображает клики на любой элемент формы. Если форма была хотя бы один раз заполнена и отправлена, показатель изменится на “Заполнения”.
- **Заполнения** — пользователь заполнил и отправил форму попапа.
- **Показы** — попап показан пользователю.
- **Уникальные показы** — попап показан пользователю. Если попап был показан пользователю несколько раз в рамках одной сессии, будет засчитан один уникальный показ.
- **Закрытия** — клик на кнопку закрытия в попапе.

В таблице справа данные представлены в процентном и количественном соотношении. Внизу блока показано, сколько всего раз было зарегистрировано выбранное событие. Дополнительно можно изменить способ

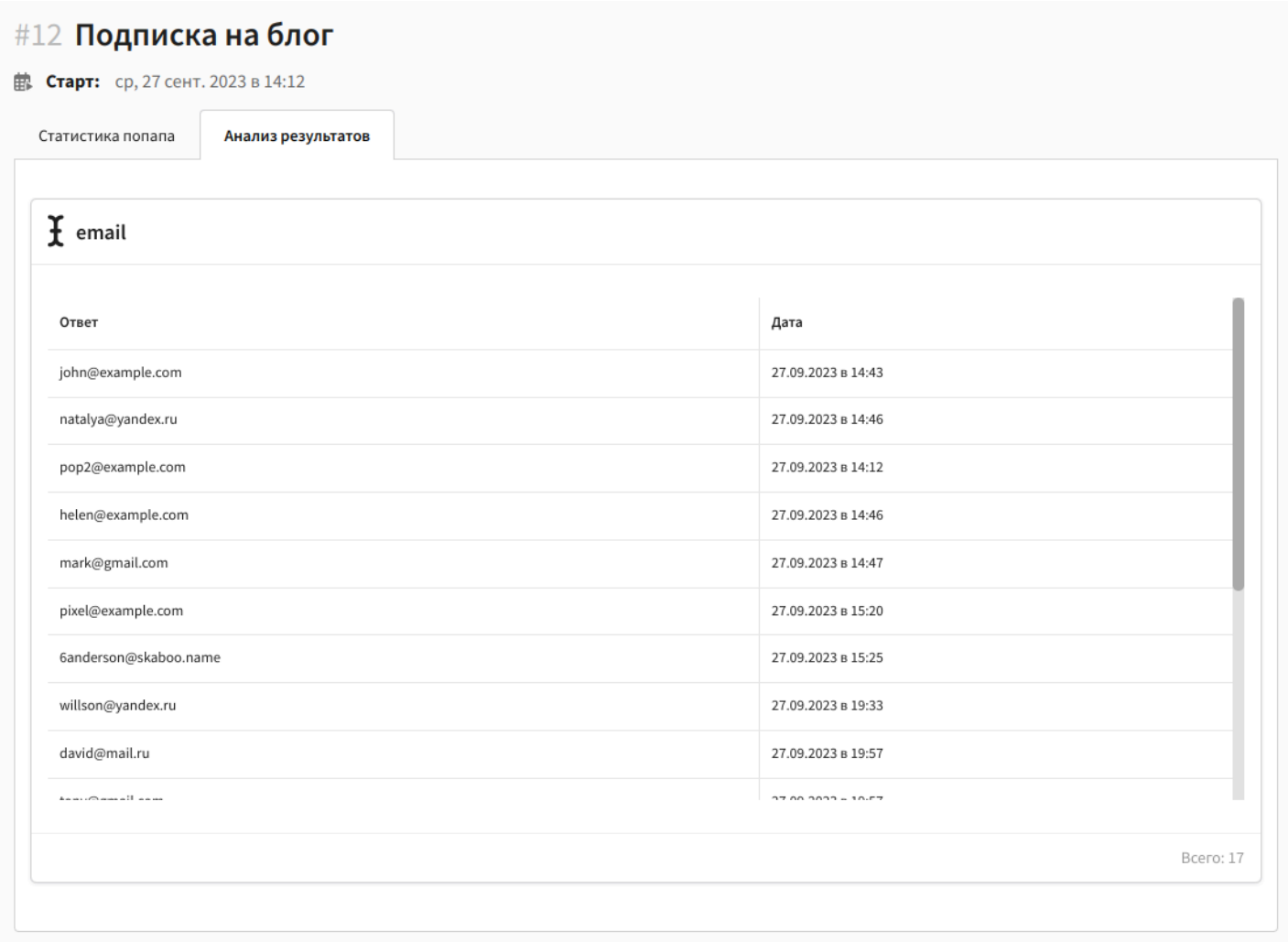
отображения данных:

— круговая диаграмма

— столбчатая диаграмма

Анализ результатов

На этой вкладке вы можете получить информацию о заполнении полей формы в попапе:



Статистика формируется в зависимости от типа поля для ввода, указанного в коде попапа: `<input type="<тип_поля">`. Для корректного сбора статистики следует установить атрибут `type`, значение которого будет соответствовать типу вводимых данных. Подробнее о полях для ввода вы можете прочитать в этой статье.

Руководство: попап для подписки на push

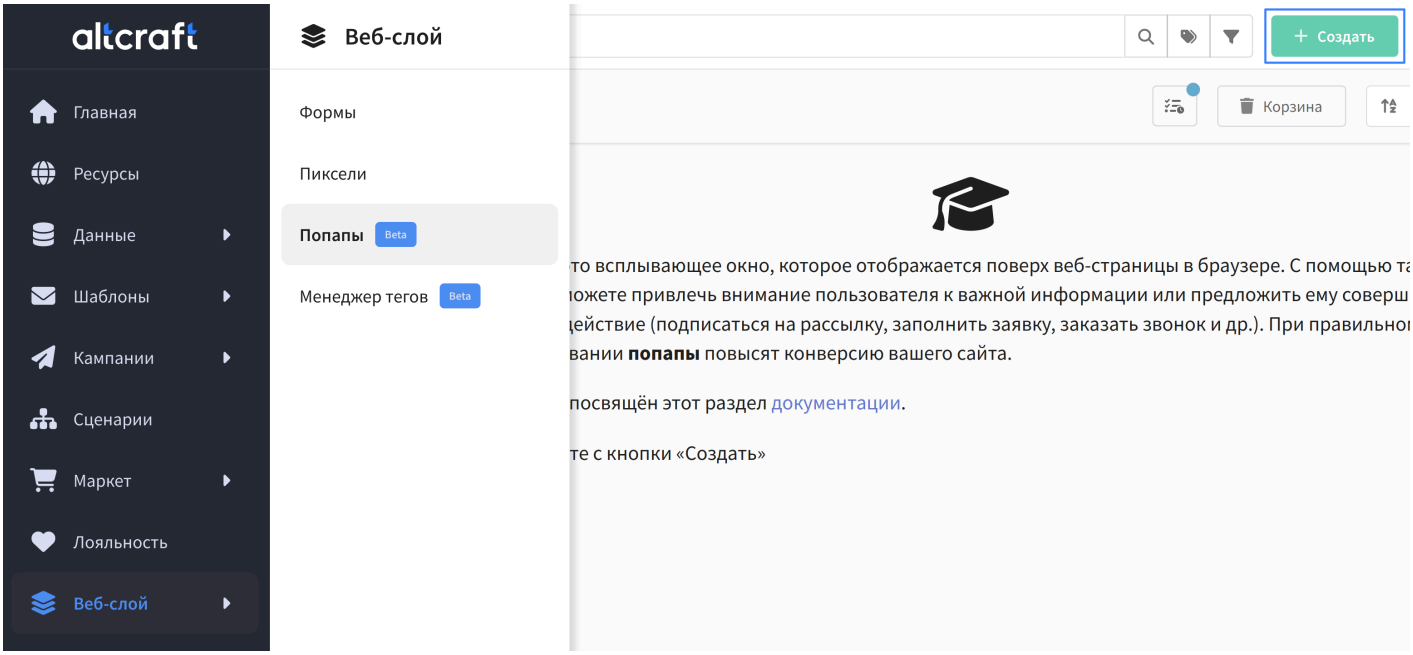
Один из самых распространенных видов попапов — попап подписки на push-уведомления. Такое всплывающее окно поможет вам быстро подписать посетителей вашего сайта на ресурс платформы Altcraft. В этой статье мы покажем, как создать и настроить попап, который будет появляться при загрузке страницы для тех пользователей, которые еще не давали разрешения на показ push-уведомлений на сайте.

INFO

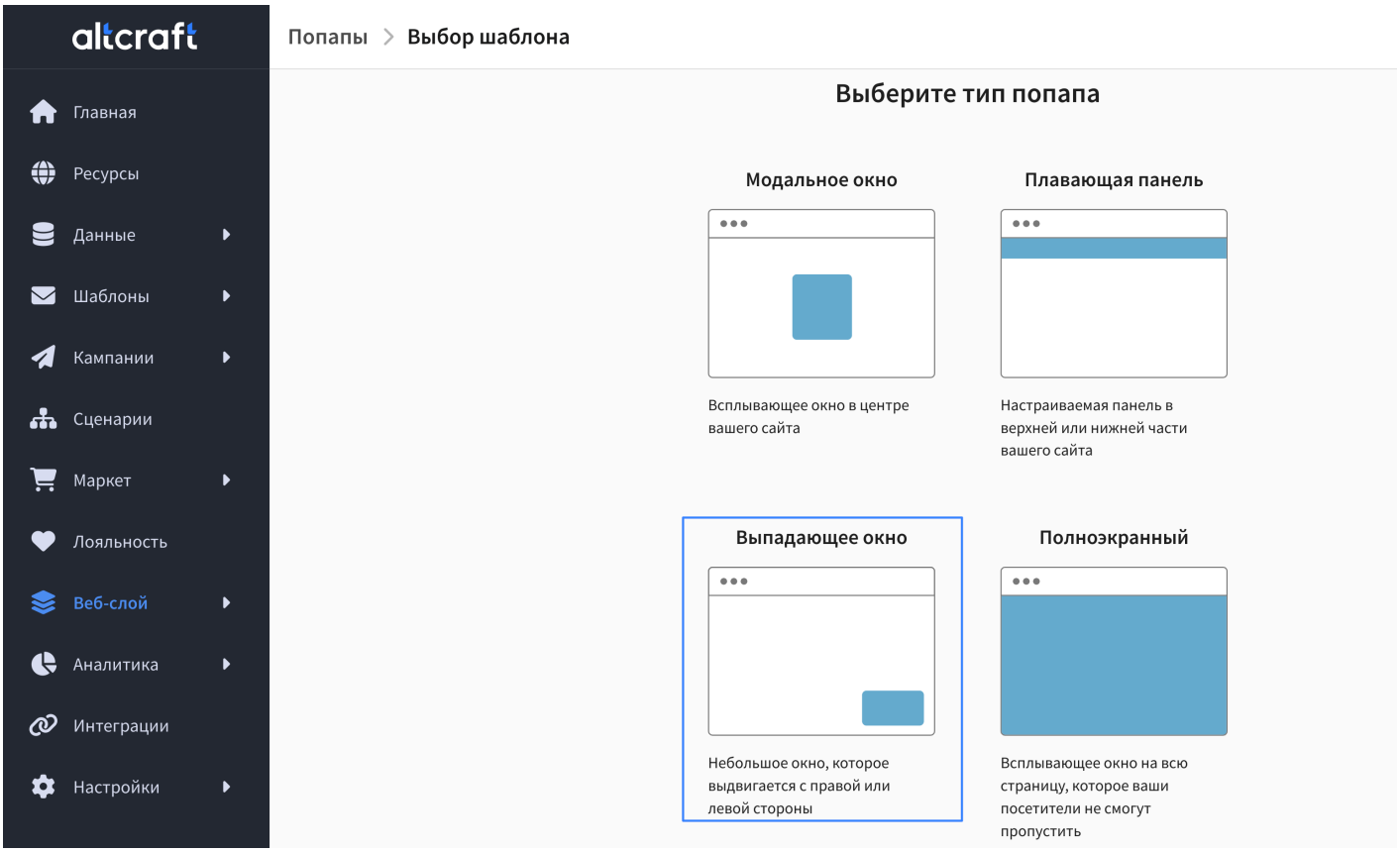
Для корректной работы подписки на push-уведомления вы должны предварительно настроить **ресурс** на платформе, а также добавить соответствующий **скрипт** себе на сайт. Подробнее о настройке push-уведомлений можно узнать здесь.

Создание и публикация попапа

Перейдите в раздел **“Веб-слой” — “Попапы”**. Создайте новый попап:



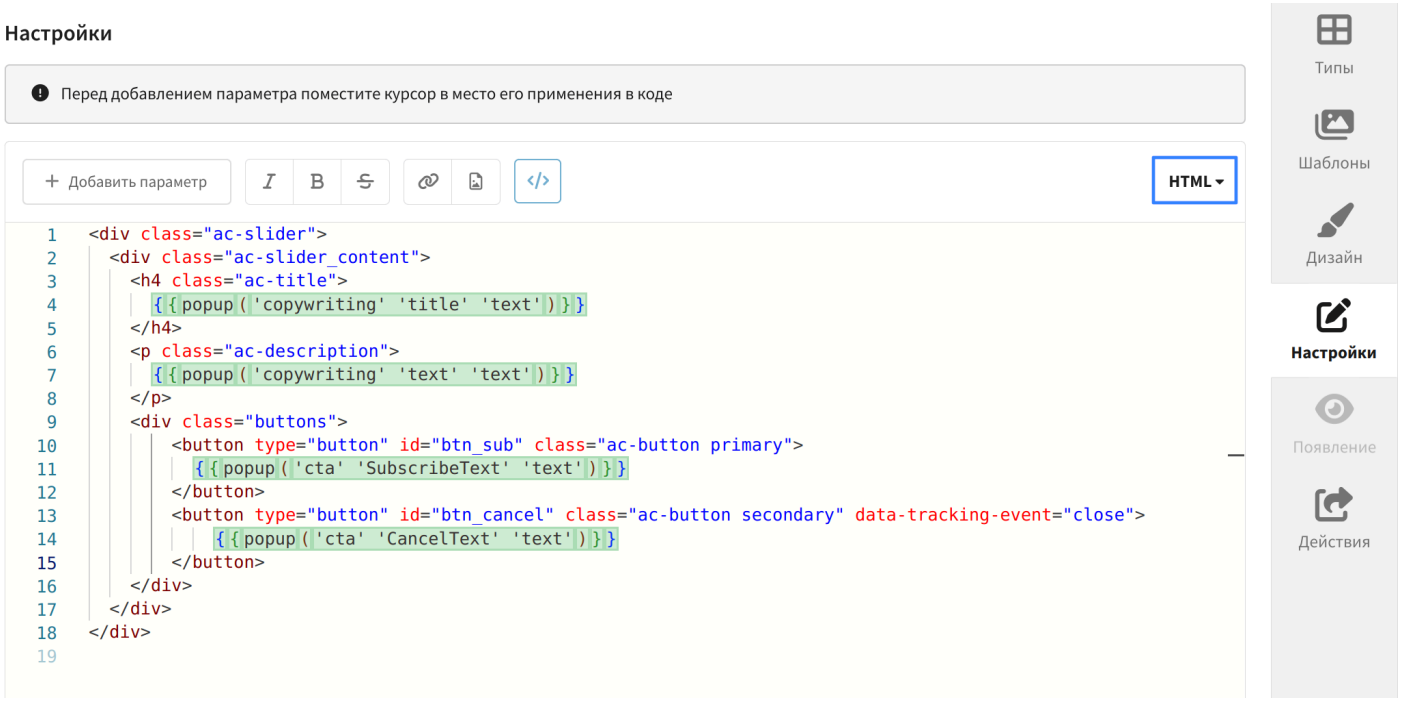
Выберите тип **“Выпадающее окно”**:



Выберите **шаблон попапа**. Затем перейдите в раздел “Настройки” открывшегося меню в правой части экрана. Замените текущий HTML-код на следующий:

```
<div class="ac-slider">
  <div class="ac-slider_content">
    <h4 class="ac-title">
      {{popup('copywriting' 'title' 'text')}}
    </h4>
    <p class="ac-description">
      {{popup('copywriting' 'text' 'text')}}
    </p>
    <div class="buttons">
        <button type="button" id="btn_sub" class="ac-button primary">
          {{popup('cta' 'SubscribeText' 'text')}}
        </button>
        <button type="button" id="btn_cancel" class="ac-button secondary" data-tracking-
event="close">
            {{popup('cta' 'CancelText' 'text')}}
        </button>
    </div>
  </div>
</div>
```

В итоге тело попапа должно выглядеть следующим образом:



Смените формат на JS и добавьте код, который будет инициализировать подписку на push при клике на кнопку “Подписаться”:

```
const popupContainer = document.querySelector('[data-ac-popup-container]');

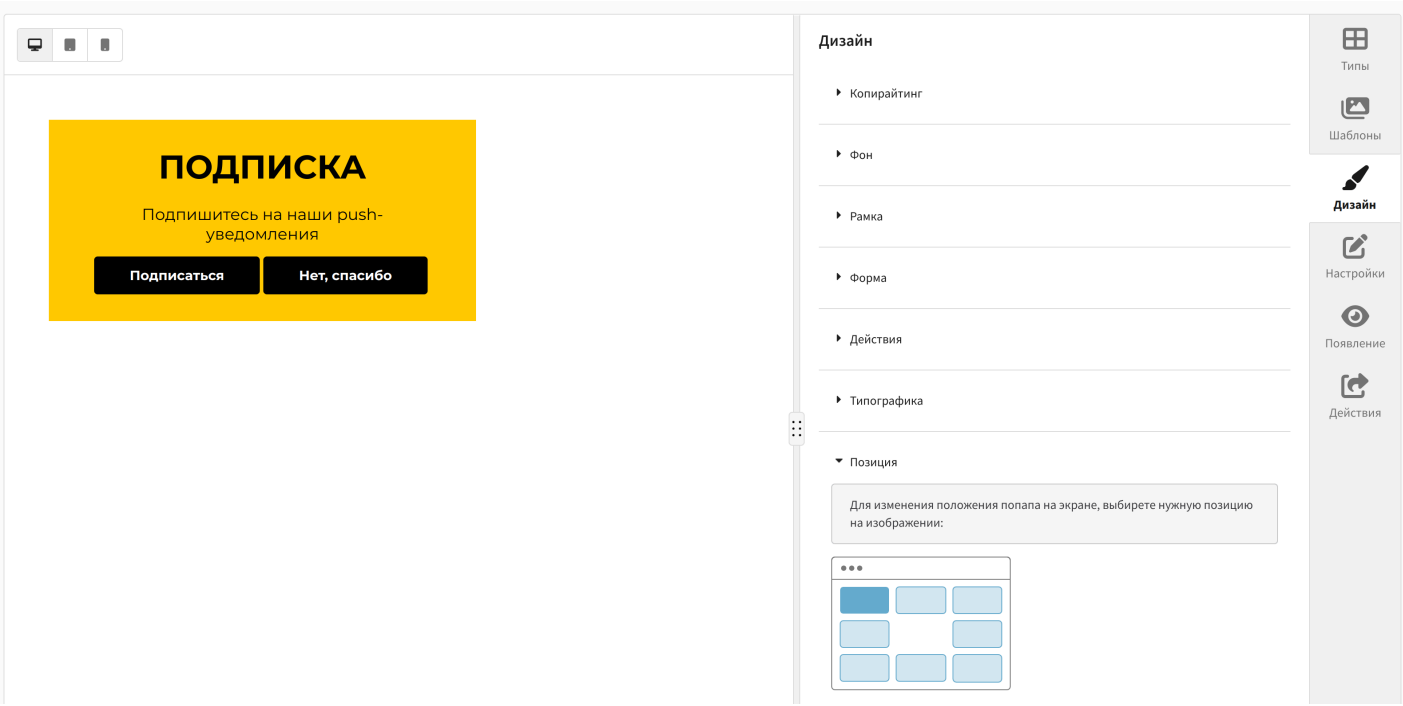
if (popupContainer) {
    const popupHost = popupContainer.shadowRoot || document;
    const subscribeBtn = popupHost.getElementById('btn_sub');

    if (subscribeBtn) {
      subscribeBtn.addEventListener('click', () => {
          // Поиск профиля по email. Также будет записано в профиль, если совпадений не
найдётся.
          // new AKPush().initSubscription({ email: 'example@email.com' });
          new AKPush().initSubscription();
          popupContainer.style.display = 'none';
      });
    }
}
```

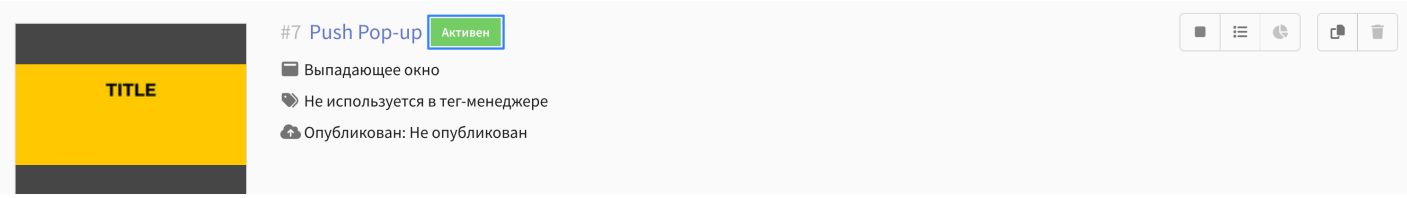
TIP

Узнать подробнее о вёрстке и настройке попапа в Редакторе кода вы можете в этой статье.

В разделе “Дизайн” вы можете настроить отображение элементов, их цвет, текст надписей, а также расположение окна на экране:



Когда все изменения в попап будут внесены, укажите его имя, а затем нажмите кнопку **“Применить”** и **“Опубликовать”**. В списке попапов проверьте, активен ли он:



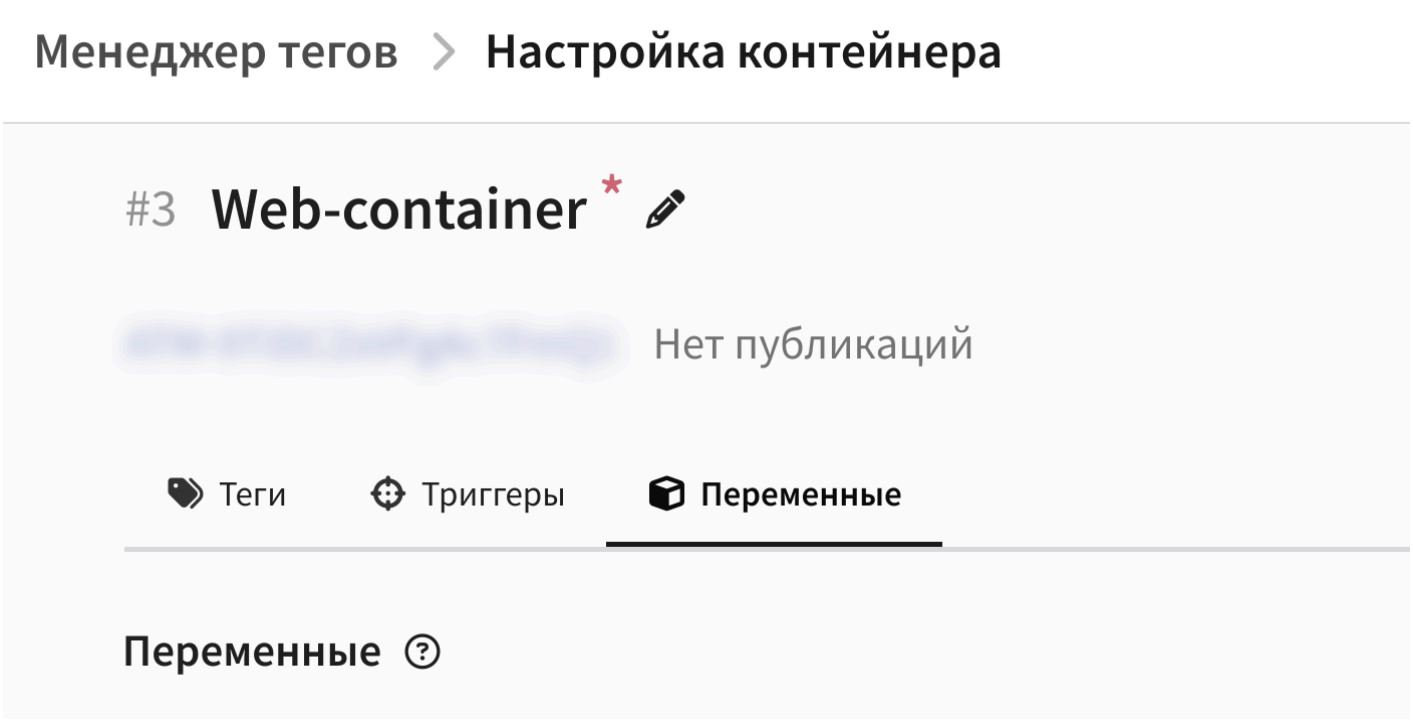
Настройка Менеджера тегов

INFO

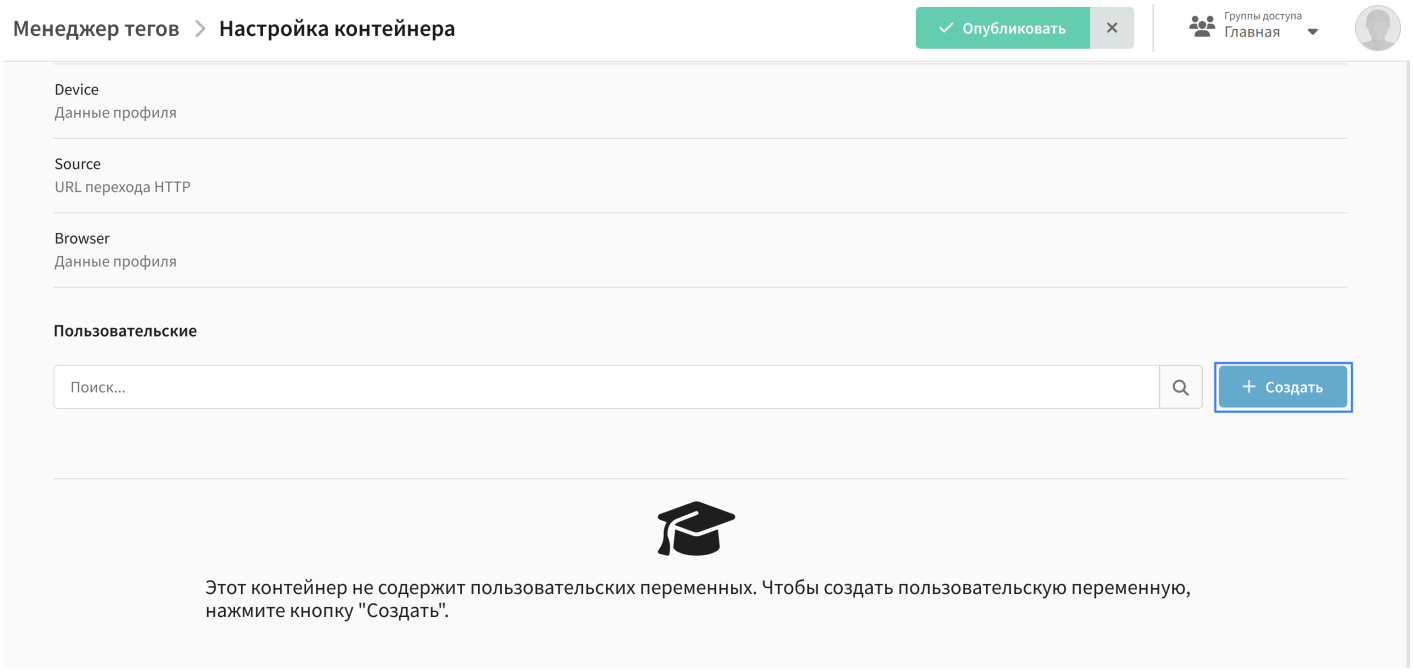
Если до этого вы не работали с **Менеджером тегов** и не настраивали **контейнер**, вы можете ознакомиться с инструкцией по настройке этих элементов здесь.

Создание переменной

Перейдите в раздел **“Веб-слой” — “Менеджер тегов”**. Откройте используемый для сайта контейнер:



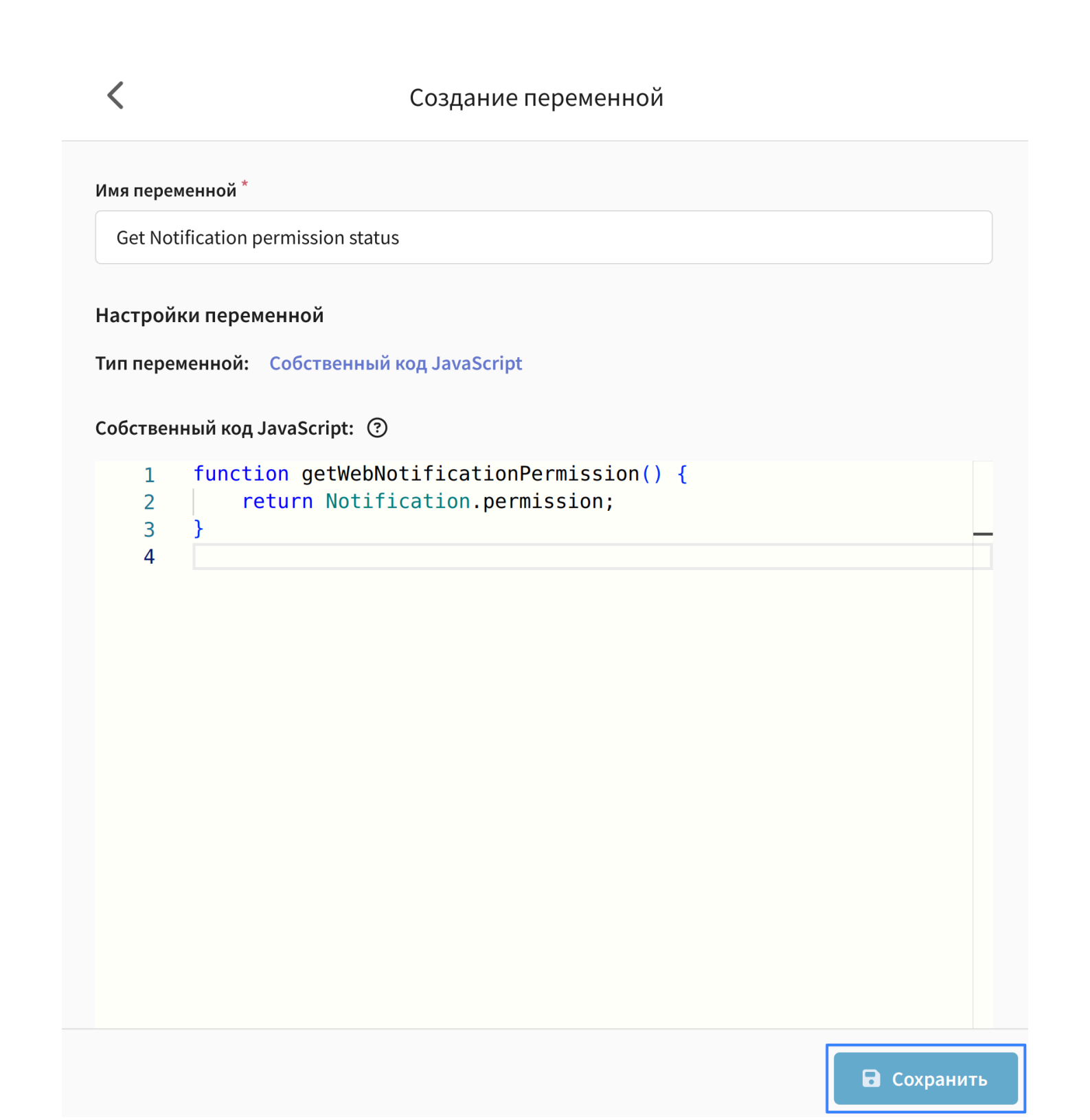
Выберите пункт **“Переменные”** и в самом низу страницы создайте пользовательскую переменную:



Укажите название, например, “Get Notification permission status”, затем выберите тип **Собственный код JavaScript**. В поле ввода вставьте следующий код:

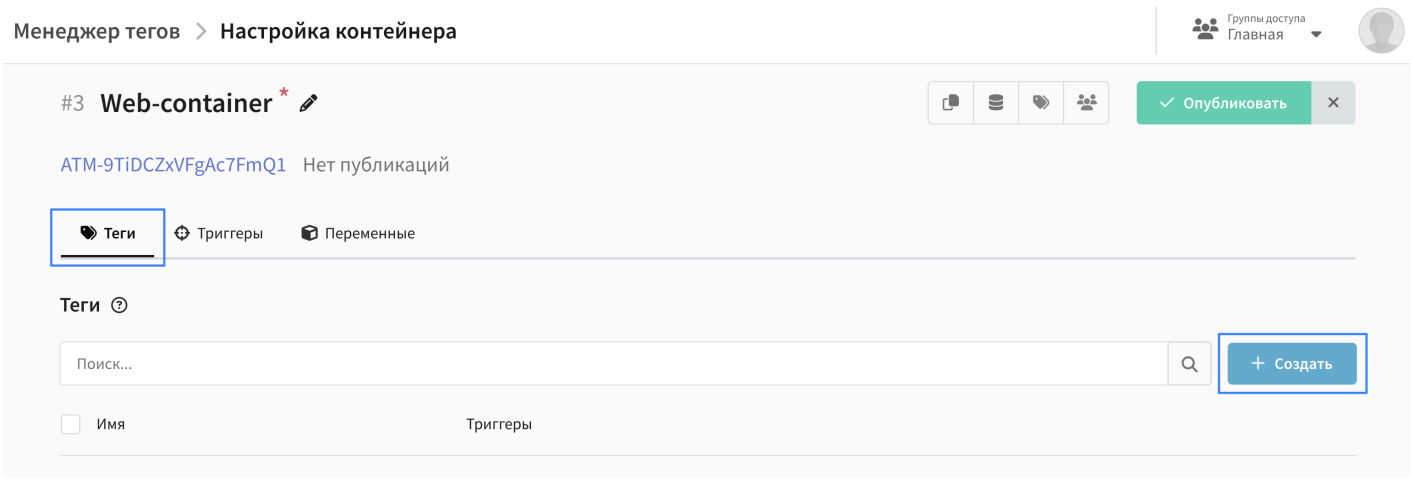
```
function getWebNotificationPermission() {
    return Notification.permission;
}
```

Сохраните созданную переменную:

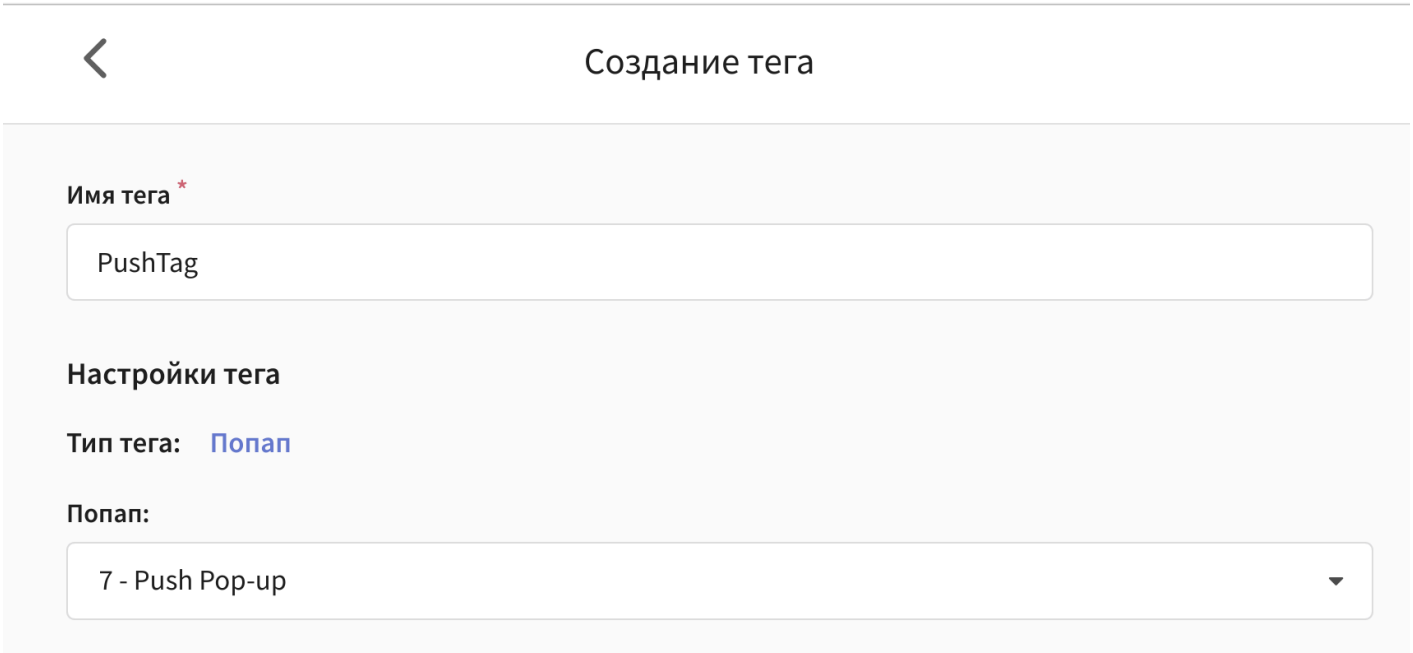


Создание тега

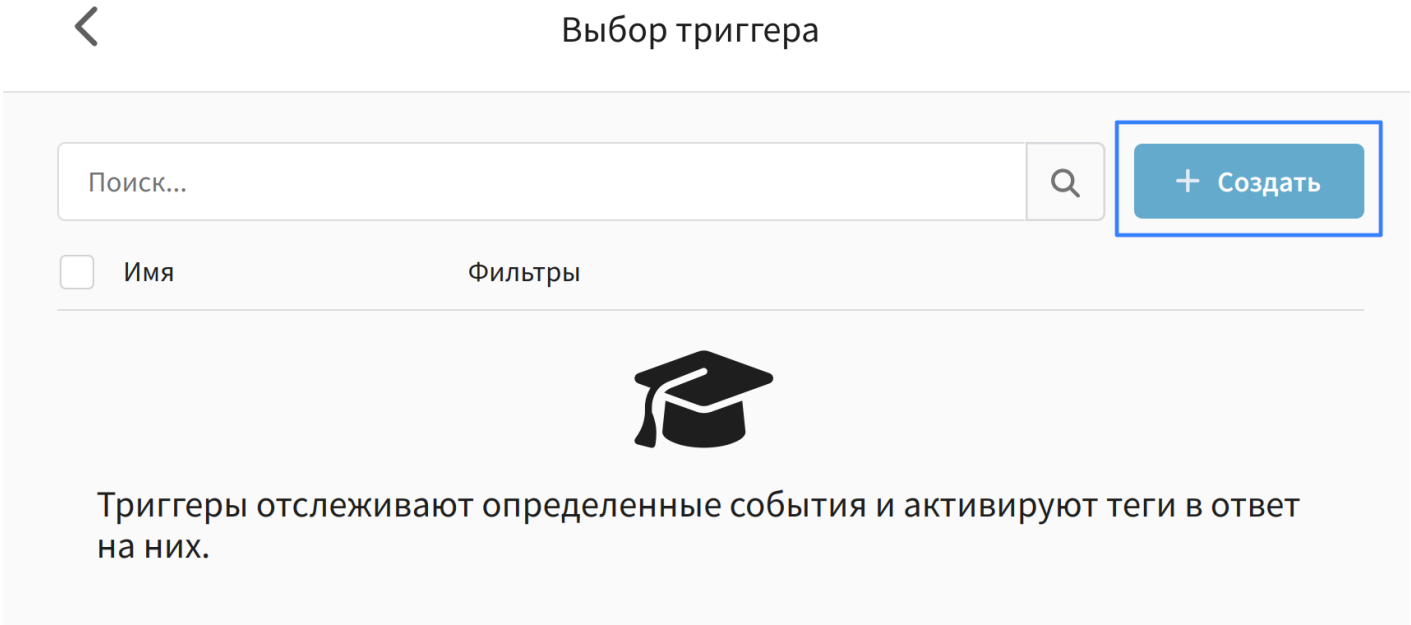
Перейдите в раздел **“Теги”** и создайте новый тег:



Назовите его, укажите тип **“Попап”** и выберите созданный вами на предыдущем этапе попап:

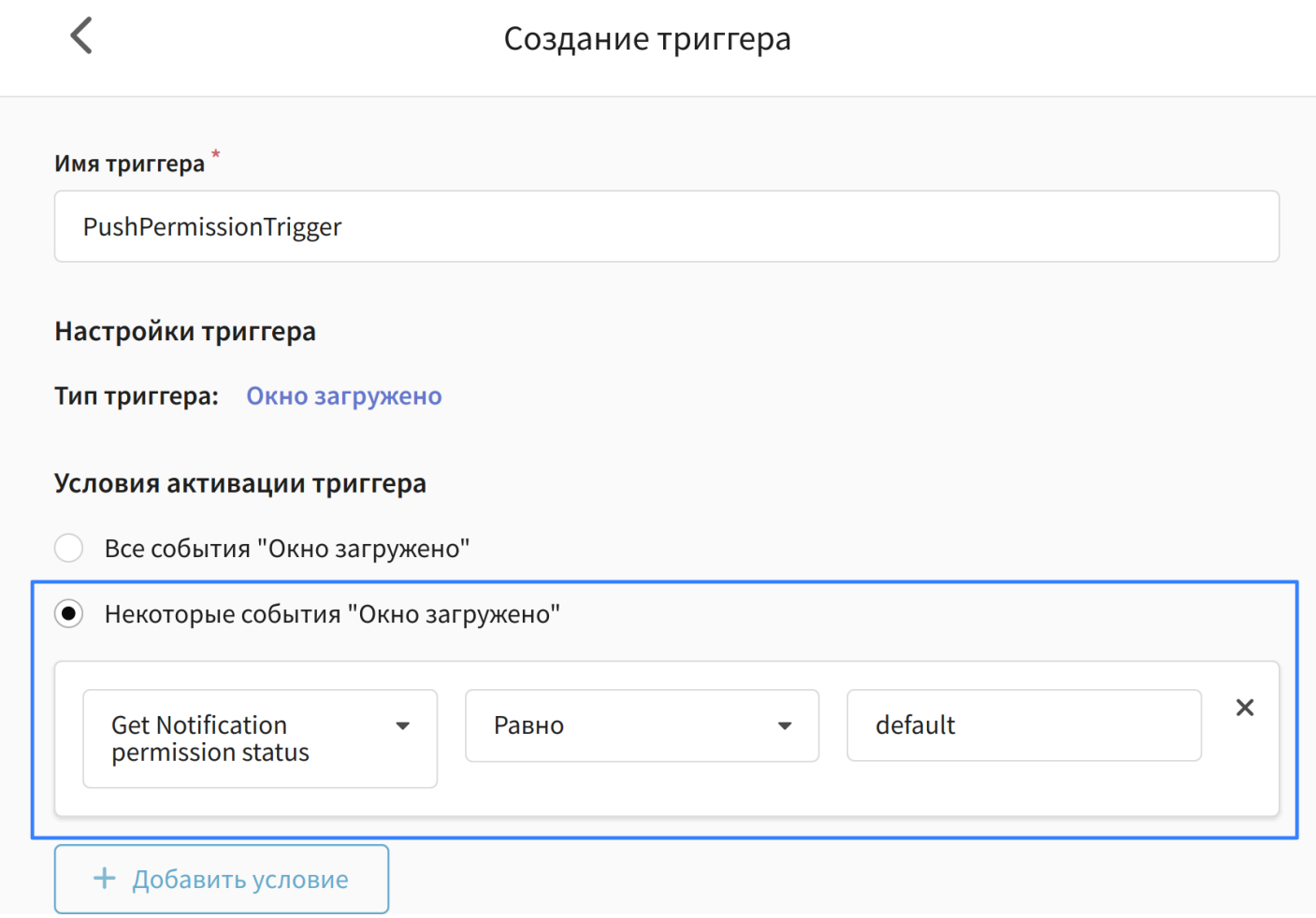


При необходимости настройте настройки активации и ограничения на срабатывания. Затем выберите **триггеры** для запуска. Если вы еще не создавали триггеры, вы можете создать новый прямо из этого окна:

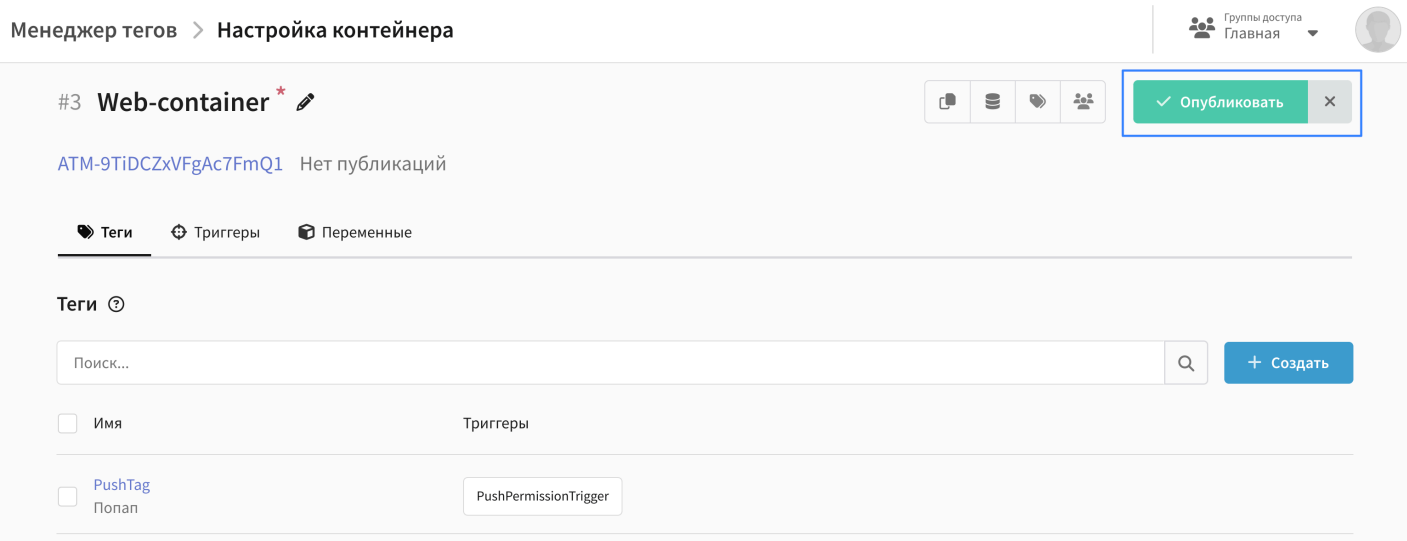


Вы можете настроить триггер самостоятельно на любое необходимое действие. В нашем примере мы используем тип “Окно загружено”.

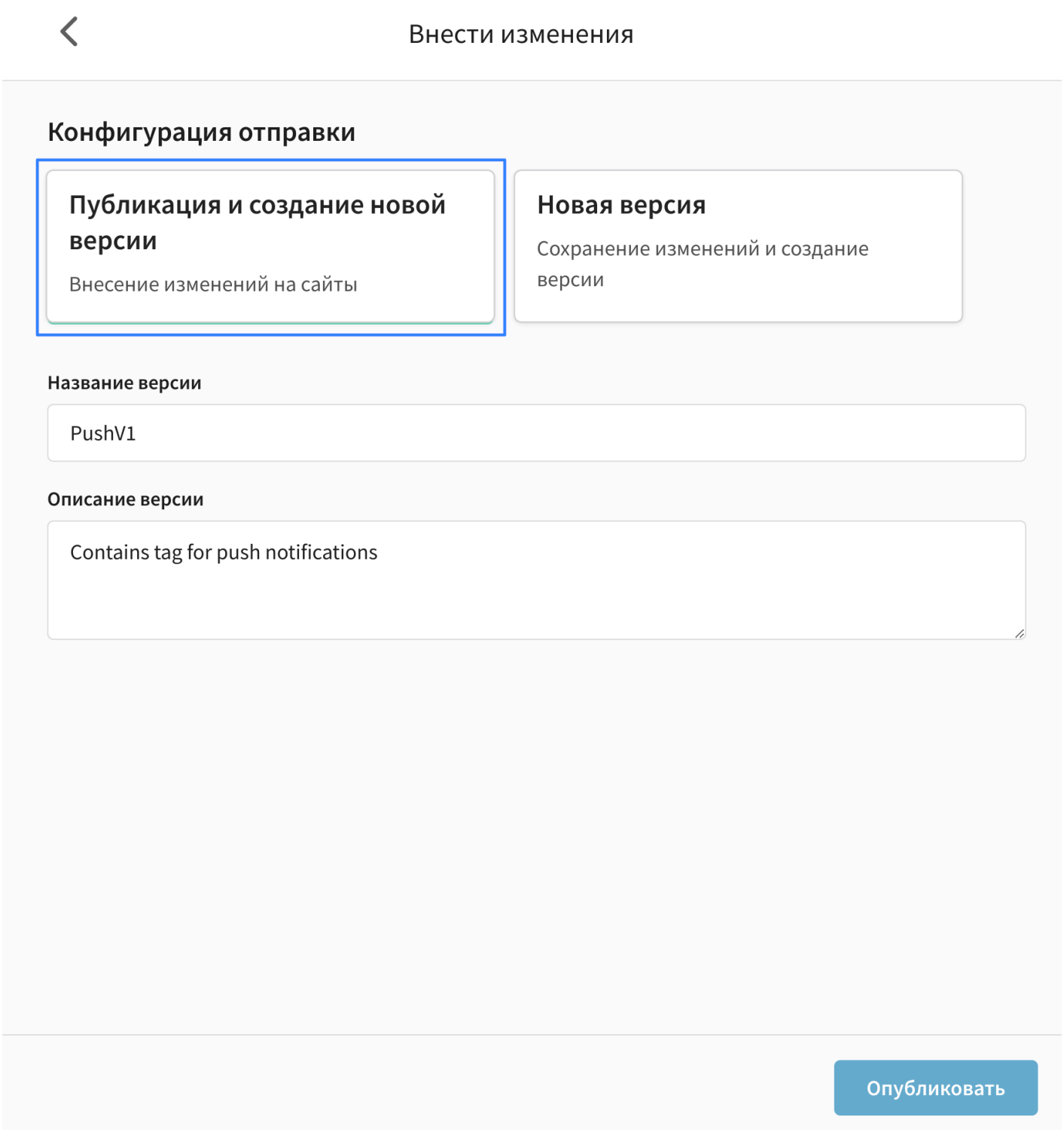
В условиях активации укажите **“Некоторые события”** и созданную раннее переменную. Она отвечает за **статус разрешения** на показ уведомлений. Переменная будет равна `default` в тех случаях, когда браузер ещё не запрашивал разрешение на показ у пользователя. В таких ситуациях и будет срабатывать попап:



Придумайте для триггера название и сохраните его. Выберите его как триггер для показа вашего попапа и сохраните тег. Когда тег будет сохранен и все изменения в контейнер будут внесены, нажмите кнопку **“Опубликовать”**:



Выберите опцию “Публикация и создание новой версии”, укажите название и описание версии и опубликуйте ее:



Работа всплывающего окна настроена. Теперь при попадании на страницу, пользователь увидит окно с предложением подписаться на push-уведомления.

Базовые кейсы размещения попапа через Менеджер тегов

Настройка триггеров срабатывания попапа

Доступны четыре основных сценария срабатывания попапа:

- При загрузке страницы
- При скролле
- При клике
- Через определенное время нахождения на странице

> **INFO**
>
> Для размещения попапа необходимо добавить его в Менеджер тегов Altcraft и установить код контейнера на сайт.

Показ попапа при загрузке страницы

Чтобы настроить показ всплывающего окна (попапа) сразу после загрузки страницы, создайте соответствующий триггер в Менеджере тегов. Тип триггера зависит от архитектуры вашего сайта:

Архитектура сайта	Триггер
Для MPA-сайтов (с полной перезагрузкой страницы)	**Просмотр страницы**
Для SPA-сайтов (без перезагрузки страницы)	**Изменение истории**

Для настройки создайте триггер в Менеджере тегов: перейдите во вкладку **“Триггеры”** —> **“Создать”.**

Выберите соответствующий тип триггера и установите условия активации:

- **Все просмотры страницы** — попап будет показан на любой странице с кодом контейнера.
- **Некоторые просмотры страницы** — например: `Page URL –> равно –> https://site.com/contact`. Попап будет показан только на странице контактов.

После настройки нажмите “Сохранить” и опубликуйте изменения в Менеджере тегов.

Показ попапа при скролле

Попап будет показан при прокрутке страницы на указанную глубину. Для настройки создайте триггер типа **Глубина прокрутки"** и укажите значение в процентах (например, 50%). Триггер может срабатывать на глубину как вертикальной, так и горизонтальной прокрутки.

Установите условия активации:

- **Все просмотры страницы** — срабатывание на любой странице.

- **Некоторые просмотры страницы** — например: `Page Path –> содержит –> /blog/`. Попап будет показан только при скролле страниц блога.

После настройки нажмите “Сохранить” и опубликуйте изменения в Менеджере тегов.

Показ попапа при клике

Попап будет показан при клике на элемент страницы. Для настройки создайте триггер типа **“Клик по элементу”**.

Установите условия активации:

- **Все клики** — срабатывание при любом клике на странице.
- **Некоторые клики** — например: `Click ID –> равно –> subscribe-button`. Попап будет показан только при клике на кнопку с ID “subscribe-button”.

После настройки нажмите “Сохранить” и опубликуйте изменения в Менеджере тегов.

Показ попапа через определенное время

При такой настройке триггера срабатывания, попап будет показан по прошествии указанного времени.

Для этого создайте триггер в Менеджере тегов. Перейдите во вкладку триггеры —> создать Введите название триггера и выберите его тип **“Таймер”**.

Настройки триггера включают:

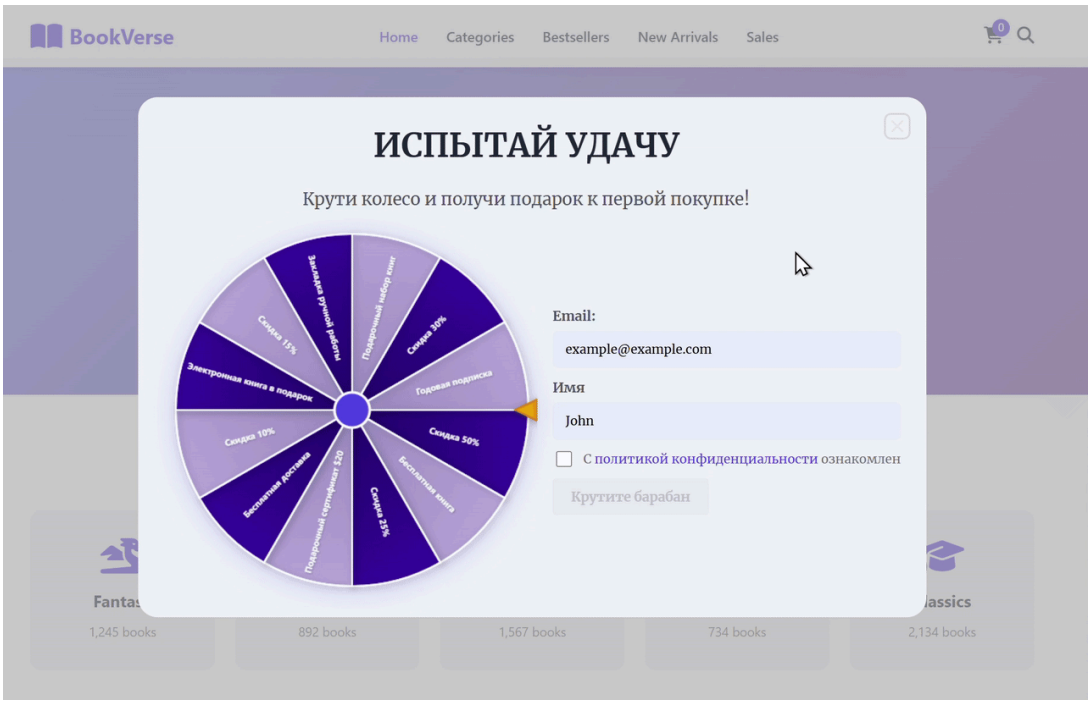
- **Название события по умолчанию** — `actm.timer`. В большинстве случаев название менять не нужно. Вы можете указать другое название, чтобы создать триггер на основе пользовательского события.
- **Интервал** — время между срабатываниями события (в миллисекундах). Например, если указать 5000, то событие будет срабатывать каждые 5 секунд. Первое срабатывание также произойдет через указанное время.
- **Ограничение** — максимальное количество срабатываний. Если оставить поле пустым, триггер будет срабатывать до тех пор, пока пользователь не покинет страницу.

Например, для показа попапа через 10 секунд после загрузки страницы укажите интервал 10000 и ограничение 1.

После настройки нажмите “Сохранить” и опубликуйте изменения в Менеджере тегов.

Кейс: Создание попапа с виджетом “Колесо фортуны”

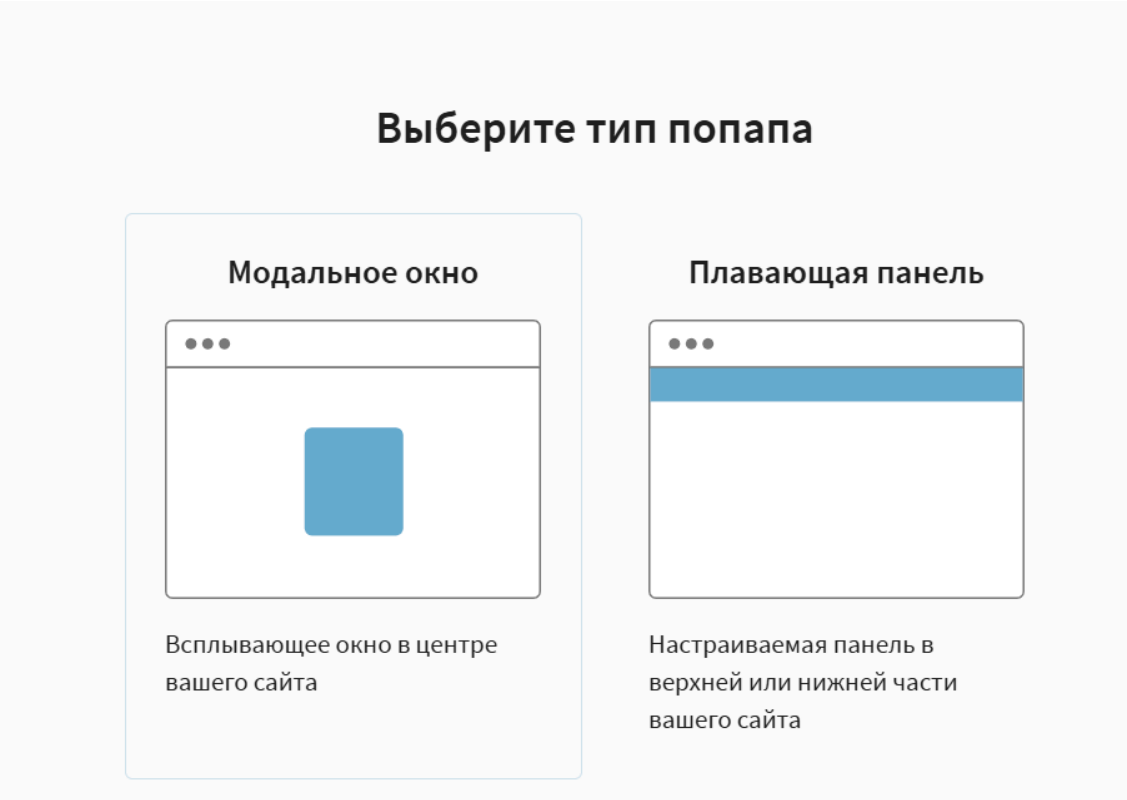
Специальный виджет “Колесо фортуны” позволяет создать интерактивный попап, который вовлекает посетителей сайта в игру с призами. Пользователь заполняет простую форму и вращает колесо, чтобы получить случайный подарок: скидку, промокод или другой бонус. Данное руководство описывает процесс создания такого попапа в конструкторе Altcraft:



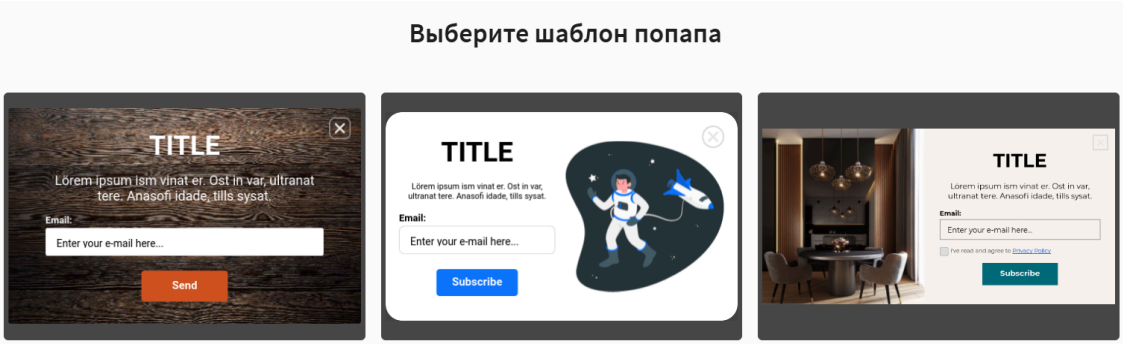
Перед настройкой колеса рекомендуется создать поле в базе профилей, в которое будет сохраняться информация о выигранном призе. О том, как создавать дополнительные поля, вы можете узнать в этой статье.

Создание попапа и выбор шаблона

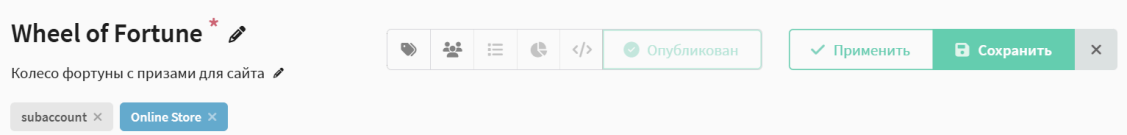
Начните с создания нового попапа в разделе **Веб-слой — Попапы**, нажав кнопку **Создать**. Выберите **тип попапа.** Для колеса фортуны идеально подходит **Модальное окно** — оно появляется в центре экрана:



Выберите шаблон. Воспользуйтесь любым из готовых шаблонов как основой. Для упрощения кастомизации рекомендуется использовать шаблоны с минималистичным дизайном (например, с космонавтом):



Задайте **имя и описание.** Укажите системное имя и при необходимости — описание:



Интеграция кода виджета

Виджет “Колесо фортуны” реализован на HTML, CSS и JavaScript. Для его добавления необходимо заменить код стандартного шаблона на предоставленный готовый код.

CSS

Перейдите на вкладку **CSS**. **Не удаляйте существующие стили.** Прокрутите редактор в самый конец и добавьте предоставленный CSS-код для стилизации колеса и его контейнера.

```
.formWrapper {
    display: flex;
    align-items: center;
    justify-content: space-around;
    flex-wrap: wrap;
}

.ac-button:disabled {
    opacity: 0.6;
    cursor: not-allowed;
    background-color: #cccccc;
    color: #666666;
    transform: none;
    box-shadow: none;
}

.ac-button:disabled:hover {
    background-color: #cccccc;
    color: #666666;
    transform: none;
    box-shadow: none;
}
```

HTML

В разделе **“Настройки”** откройте вкладку **HTML**. Полностью удалите текущее содержимое редактора и вставьте предоставленный HTML-код для колеса.

```
<div class="ac-popup-overlay">
    <div class="ac-modal">
        <div class="ac-modal_content">
            <div style="display: none;" data-popupId="{{popup('copywriting' 'popupId'
'text')}}"></div>
            <div id="firstPage">
                <h4 class="ac-title">
                    {{popup('copywriting' 'title' 'text')}}
                </h4>
                <div class="ac-description">
                    {{popup('copywriting' 'text' 'text')}}
                </div>
                <div class="formWrapper">
                    <div class="canvas-container">
                        <canvas id="canvas" width="420" height="420"></canvas>
                    </div>
                    <form id="ac-form" class="ac-form" action>
                            <div class="ac-field">
                            <label class="ac-label" for="email">
                                {{popup('copywriting' 'emailLabel' 'text')}}
                            </label>
                            <input
                                id="email"
                                type="email"
                                class="ac-input"
                                name="{{popup('form' 'email' 'fieldData')}}"
                                placeholder="{{popup('copywriting' 'emailPlaceholder'
'text')}}"
                            />
                            </div>
                            <div class="ac-field">
                            <label class="ac-label" for="name">
                                {{popup('copywriting' 'nameLabel' 'text')}}
                            </label>
                            <input
                                id="name"
                                class="ac-input"
                                name="{{popup('form' 'name' 'fieldData')}}"
                                placeholder="{{popup('copywriting' 'namePlaceholder'
'text')}}"
                            />
                            </div>
                            <div class="ac-field">
                            <label>
                                <input type="checkbox" name="{{popup('form' 'privacyPolicy'
'fieldData')}}" id="privacyPolicy" required>
                                <span>C <a href="{{popup('copywriting' 'policy' 'link')}}"
target="_blank">политикой конфиденциальности</a> ознакомлен</span>
                            </label>
```

```
                            </label>
                            </div>
                            <div class="ac-validation-msg" for="email"></div>
                            <input type="text" style="display: none;" class="ac-input"
id="prizeField" name="{{popup('form' 'prize' 'fieldData')}}">
                        <button class="ac-button" type="button" id="winBtn" disabled>
                            {{popup('copywriting' 'winBtn' 'text')}}
                        </button>

                        <div class="ac-form_submit" style="display: none;">
                            <button type="submit" class="ac-button" id="submitBtn">
                                {{popup('cta' 'text' 'text')}}
                            </button>
                        </div>
                    </form>
                </div>
            </div>
             <div id="secondPage" style="display: none;">
                <h4 class="ac-title">
                    {{popup('copywriting' 'title-congrats' 'text')}}
                </h4>
                <div class="ac-description">
                    {{popup('copywriting' 'text-before-prize' 'text')}}
                    <span id="prizeName"></span>
                    <div>
                        {{popup('copywriting' 'text-after-prize' 'text')}}
                    </div>
                </div>
            </div>
        </div>
        {{popup('closeButton' 'closeButton' 'closeBtn')}}
    </div>
</div>
```

JavaScript

Переключитесь на вкладку **JavaScript**. Полностью удалите текущее содержимое и вставьте предоставленный JS-код, который отвечает за логику вращения колеса, определение выигрыша и работу с формой.

```
// Список выигрышей. Длина может варьироваться
// id - Идентификатор, отправляемый на сервер
// text - Подпись у сектора колеса
const options = [
  {id: 1, text: '$100'},
  {id: 2, text: '$10'},
  {id: 3, text: '$25'},
  {id: 4, text: '$250'},
  {id: 5, text: '$30'},
  {id: 6, text: '$1000'},
  {id: 7, text: '$1'},
  {id: 8, text: '$200'},
  {id: 9, text: '$45'},
  {id: 10, text: '$500'},
  {id: 11, text: '$5'},
  {id: 12, text: '$20'}
]

// Цвета секторов колеса. Если цветов меньше, чем секторов, то цвета повторяются циклично
const colors = ['white', '#3583FF']

// Начальный угол поворота колеса
let currentAngle = 0;

// Длительность вращения колеса задаётся случайно
// Минимальное время вращения (сек)
const minSpinTime = 4;

// Максимальное время вращения (сек)
const maxSpinTime = 7;

// Задержка между остановкой колеса и сообщением о выигрыше (сек)
const pageSwitchDelay = 2;

// Цвет подписей секторов на колесе
const textColor = "black";

let isPreview = true;
let myPopup = null;

const allPopups = document.querySelectorAll('[data-ac-popup-container]');

allPopups.forEach(item =>{
    const idContainer = item.shadowRoot.querySelector('[data-popupId={{popup('copywriting'
'popupId' 'text')}}]');
    if(idContainer){
        myPopup = item.shadowRoot;
        isPreview = false;
    }
```

```
    }
})

if (!myPopup) {
  myPopup = document;
}

const winBtn = myPopup.getElementById('winBtn');

function initInputs(){
  let emailInput = myPopup.getElementById('email');
  let nameInput = myPopup.getElementById('name');
  let policyInput = myPopup.getElementById('privacyPolicy');

  function updateButtonState() {
      const isEmailEmpty = !emailInput.value.trim();
      const isNameEmpty = !nameInput.value.trim();
      const isPolicyChecked = policyInput.checked;

      console.log(myPopup.getElementById('winBtn'))
      myPopup.getElementById('winBtn').disabled = isEmailEmpty || isNameEmpty ||
!isPolicyChecked
  }

  emailInput.addEventListener('input', updateButtonState);
  nameInput.addEventListener('input', updateButtonState);
  policyInput.addEventListener('input', updateButtonState);
  myPopup.getElementById("winBtn").addEventListener("click", spin);

  updateButtonState();
}
if(isPreview) {
  setTimeout(()=>{initInputs()}, 1000);
} else {
  initInputs();
}

let isInitialized = false;

function initializeWheel() {
    if (isInitialized) return;

    const canvas = myPopup.getElementById("canvas");
    if (!canvas) return;

    const ctx = canvas.getContext("2d");
    if (ctx && !canvas.dataset.initialized) {
        drawRouletteWheel();
        canvas.dataset.initialized = "true";
        isInitialized = true;
```

```
    }
}

const observer = new MutationObserver((mutations) => {
    for (let mutation of mutations) {
        if (mutation.type === 'childList') {
            mutation.addedNodes.forEach((node) => {
                if (node.nodeType === 1 && (node.id === 'canvas' || node.querySelector?.
('#canvas'))) {
                    setTimeout(initializeWheel, 100);
                }
            });
        }
    }
});

if (document.body) {
    observer.observe(document.body, { childList: true, subtree: true });
}

document.addEventListener('DOMContentLoaded', initializeWheel);

let segmentAngle = Math.PI / (options.length / 2);
let spinTimeout = null;

let spinAngleStart = 0;

let spinTime = 0;
let spinTimeTotal = 0;

let ctx;

function drawRouletteWheel() {
  const canvas = myPopup.getElementById("canvas");

  if (canvas.getContext) {
    const centerX = 210;
    const centerY = 210;
    const outsideRadius = 200;
    const textRadius = 130;
    const insideRadius = 0;

    ctx = canvas.getContext("2d");
    ctx.clearRect(0, 0, 400, 400);

    ctx.strokeStyle = "black";
    ctx.lineWidth = 2;
    ctx.font = 'bold 12px Helvetica, Arial';
```

```
    ctx.strokeStyle = "white";
    ctx.lineWidth = 15;
    ctx.beginPath();
    ctx.arc(centerX, centerY, outsideRadius, 0, 2 * Math.PI);
    ctx.stroke();

    ctx.strokeStyle = "black";
    ctx.lineWidth = 2;

    for(let i = 0; i < options.length; i++) {
      const angle = currentAngle + i * segmentAngle;
      const segmentMiddleAngle = angle + segmentAngle / 2;
      ctx.fillStyle = colors[i % colors.length];
      ctx.beginPath();
      ctx.arc(centerX, centerY, outsideRadius, angle, angle + segmentAngle, false);
      ctx.arc(centerX, centerY, insideRadius, angle + segmentAngle, angle, true);
      ctx.stroke();
      ctx.fill();

      ctx.save();
      ctx.fillStyle = textColor;

      const textX = centerX + Math.cos(segmentMiddleAngle) * textRadius;
      const textY = centerY + Math.sin(segmentMiddleAngle) * textRadius;

      ctx.translate(textX, textY);
      ctx.rotate(segmentMiddleAngle);

      const text = options[i].text;
      ctx.fillText(text, -ctx.measureText(text).width / 2, 0);
      ctx.restore();
    }

    ctx.fillStyle = "white";
    ctx.strokeStyle = "#666666";
    ctx.lineWidth = 2;

    ctx.beginPath();
    ctx.moveTo(centerX - 10, centerY - 15);
    ctx.lineTo(centerX + 30, centerY);
    ctx.lineTo(centerX - 10, centerY + 15);
    ctx.closePath();
    ctx.fill();
    ctx.stroke();
  }
}

function rotateWheel() {
  spinTime += 30;
  if(spinTime >= spinTimeTotal) {
```

```
  if(spinTime >= spinTimeTotal) {
    stopRotateWheel();
    return;
  }
  const spinAngle = spinAngleStart - easeOut(spinTime, 0, spinAngleStart, spinTimeTotal);
  currentAngle += (spinAngle * Math.PI / 180);
  drawRouletteWheel();
  spinTimeout = setTimeout(rotateWheel, 30);
}

function spin(e) {
  e.preventDefault();
  const form = myPopup.getElementById('ac-form');
  const isValid = form.checkValidity();

  if(!isValid){
    myPopup.getElementById('submitBtn').click();
    return;
  }
  winBtn.removeEventListener('click', spin);
  spinAngleStart = Math.random() * 10 + 10;
  spinTime = 0;
  spinTimeTotal = Math.random() * (maxSpinTime - minSpinTime) + minSpinTime * 1000;
  rotateWheel();
}

function handlePrize(prize){
  myPopup.getElementById("prizeField").value = prize.id;
  myPopup.getElementById('submitBtn').click();
  myPopup.getElementById('prizeName').innerText = prize.text;
}

function showSecondPage() {
  myPopup.getElementById('firstPage').style.display = 'none';
  myPopup.getElementById('secondPage').style.display = 'block';
}

function stopRotateWheel() {
  clearTimeout(spinTimeout);

  const degrees = currentAngle * 180 / Math.PI;
  const segmentDegrees = segmentAngle * 180 / Math.PI;
  const index = Math.floor((360 - degrees % 360) / segmentDegrees);

  handlePrize(options[index]);

  const form = myPopup.getElementById('ac-form');
  const isValid = form.checkValidity();

  setTimeout(()=>{
```

```
    showSecondPage();
  }, pageSwitchDelay * 1000);

}

function easeOut(currentTime, startValue, changeInValue, duration) {
    const normalizedTime = currentTime / duration;
    const squaredTime = normalizedTime * normalizedTime;
    const cubedTime = squaredTime * normalizedTime;

    return startValue + changeInValue * (cubedTime + -3 * squaredTime + 3 *
normalizedTime);
}

drawRouletteWheel();

myPopup.getElementById('ac-form').submit = () => {}
```

Настройка контента и дизайна в визуальном редакторе

После интеграции кода перейдите в раздел **Дизайн**. Здесь вы можете наполнить попап текстом и настроить его внешний вид, используя привычный визуальный интерфейс, без правки кода.

Ширина окна

Первым делом задайте **ширину** окна (**Форма — width**). Размеры холста с колесом фиксированы (420 px). Поэтому ширина всего попапа должна быть **не менее 420 px**. Если вы хотите разместить колесо и форму для ввода данных на одной горизонтальной линии, установите ширину **не менее 810 px**.

Текстовое наполнение

На вкладке **Копирайтинг** заполните ключевые текстовые параметры:

- `title` — главный заголовок на первом экране (например, “Испытай удачу!”).
- `text` — поясняющий текст или описание акции.
- `emailLabel` и `emailPlaceholder` — подпись и подсказка для поля ввода email.
- `nameLabel` и `namePlaceholder` — подпись и подсказка для поля ввода имени.
- `policy` — **обязательная ссылка** на страницу с политикой конфиденциальности. **Без заполнения этого поля сохранить попап не получится.**
- `winBtn` — текст на кнопке, которая запускает вращение колеса (например, “Крутить колесо!”).
- `popupId` — **обязательный параметр.** Уникальный текстовый идентификатор попапа (например, `bookverse_wheel`). Необходим для корректной работы, если на странице используется несколько попапов.

Попап состоит из двух экранов. Параметры для экрана с сообщением о выигрыше:

- `title-congrats` — заголовок поздравления.
- `text-before-prize` — текст, который будет показан перед названием выигранного приза.
- `text-after-prize` — дополнительный текст под сообщением о призе.

Внешний вид

Используйте остальные вкладки раздела **Дизайн** для тонкой настройки внешнего вида:

- **Фон** — измените цвет фона попапа и прозрачность затемнения страницы (`overlay`).
- **Рамка** — настройте скругление углов всплывающего окна.
- **Форма** — задайте стили для полей ввода: отступы, цвета границ, фона и текста.
- **Действия** — настройте внешний вид кнопок (цвет, скругление, шрифт).
- **Типографика** — выберите общий шрифт и стили для всех текстовых элементов.

Любые изменения в визуальном редакторе сразу отображаются в окне предпросмотра справа.

Кастомизация колеса фортуны

Внешний вид и поведение самого колеса (призы, цвета секторов, скорость вращения) настраиваются путём правки переменных в JavaScript-коде.

Вернитесь в раздел **Настройки — JavaScript**. В начале файла найдите и отредактируйте следующие переменные.

Призы (options)

Переменная `options` определяет список призов на колесе. Это массив объектов, где каждый объект содержит `id` и текст приза `text`.

```
const options = [
    { id: 1, text: '$100' },
    { id: 2, text: '$10' },
    { id: 3, text: '$25' },
    { id: 4, text: '$250' },
    { id: 5, text: '$30' },
    { id: 6, text: '$1000' },
    { id: 7, text: '$1' },
    { id: 8, text: '$200' },
    { id: 9, text: '$45' },
    { id: 10, text: '$500' },
    { id: 11, text: '$5' },
    { id: 12, text: '$20' }
];
```

Как изменить:

- **Текст приза:** Замените значение `text` на нужное (например, `'10% скидка'` или `'Бесплатная доставка'`).
- **Количество призов:** Добавьте новые объекты в массив или удалите существующие. Количество секторов на колесе изменится автоматически.

Цвета секторов (colors)

Переменная `colors` — это массив цветов для заливки секторов колеса. Цвета применяются по порядку, начиная с первого сектора.

```
const colors = ['black', 'orange'];
```

Как изменить: Замените цвета в массиве на нужные. Вы можете использовать:

- Названия цветов на английском: `'red'`, `'blue'`, `'green'`
- HEX-коды: `'#4f46e5'`, `'#f59e0b'`, `'#10b981'`
- RGB/RGBA значения: `'rgb(79, 70, 229)'`

Скорость вращения

Длительность вращения колеса задаётся двумя переменными:

- **`minSpinTime`** — минимальное время вращения (в секундах).
- **`maxSpinTime`** — максимальное время вращения (в секундах).

Фактическое время каждого вращения будет случайным числом в этом диапазоне.

```
const minSpinTime = 4;
const maxSpinTime = 7;
```

Как изменить: Увеличьте значения, чтобы колесо вращалось дольше, или уменьшите значения для более быстрого результата.

Задержка показа

Переменная **`pageSwitchDelay`** задаёт паузу (в секундах) между остановкой колеса и автоматическим переходом на экран с сообщением о выигрыше.

```
const pageSwitchDelay = 2;
```

Как изменить: Измените значение на необходимое количество секунд.

Цвет текста (textColor)

Переменная **`textColor`** определяет цвет надписей с названиями призов на секторах колеса.

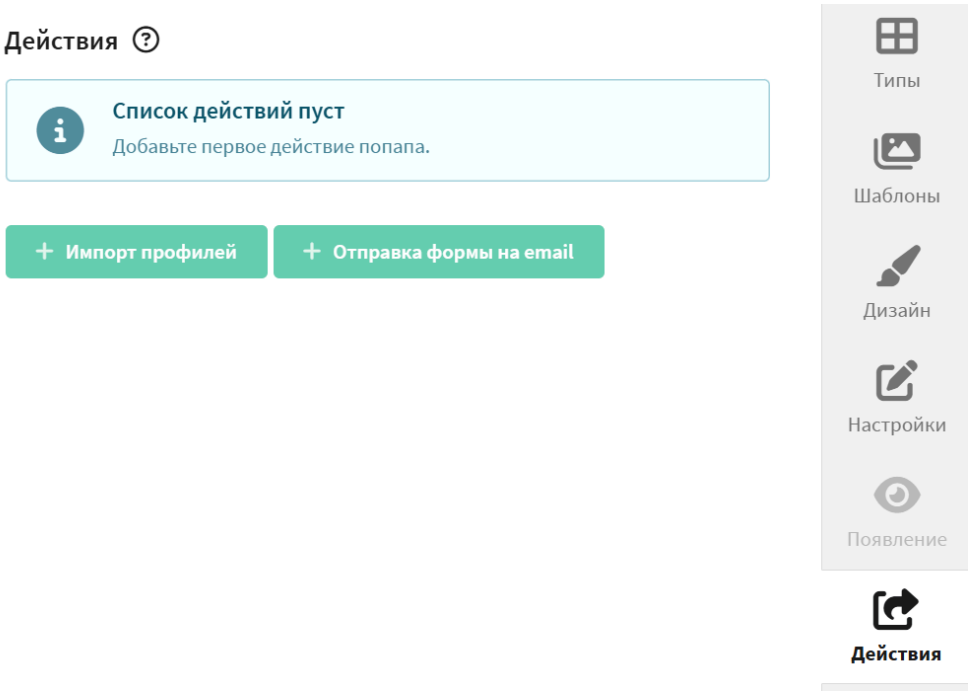
```
const textColor = "white";
```

Как изменить: Замените значение на любой валидный CSS-цвет, чтобы текст хорошо читался на фоне сектора. Например, `"#ffffff"`, `"black"`, `"#1f2937"`.

Сохранение данных о выигранном призе

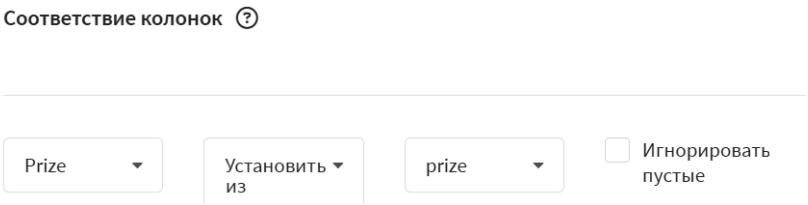
После того как пользователь вращает колесо и выигрывает приз, информация о выигрыше автоматически заполняется в скрытом поле формы `prize`. Чтобы сохранить эти данные в профиле пользователя в Altcraft, необходимо настроить действие импорта в разделе **Действия**.

В редакторе попапа откройте вкладку **Действия**:



Настройте импорт профилей:

- Следуйте стандартной процедуре настройки импорта, описанной в статье о создании попапов.
- В настройках импорта обязательно настройте сопоставление поля `prize` из формы с нужным полем в вашей базе данных Altcraft.



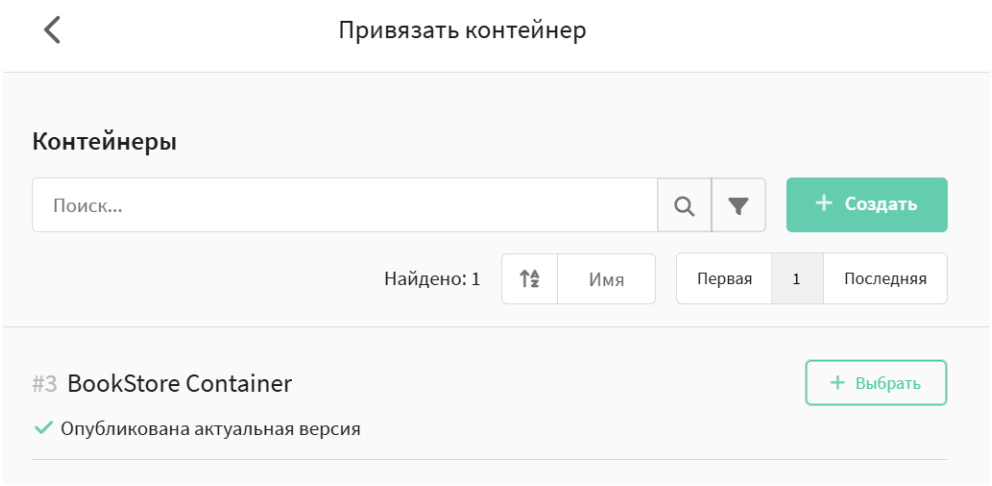
По умолчанию JavaScript-код записывает в поле `prize` текст выигранного приза (`prize.text`). Если вам нужно сохранять идентификатор приза (`prize.id`), отредактируйте функцию `handlePrize()` в JS-коде, заменив `prize.text` на `prize.id`.

После настройки импорта при каждой отправке формы данные о выигрыше (текст или ID приза) будут сохраняться в профиле пользователя вместе с его email и именем.

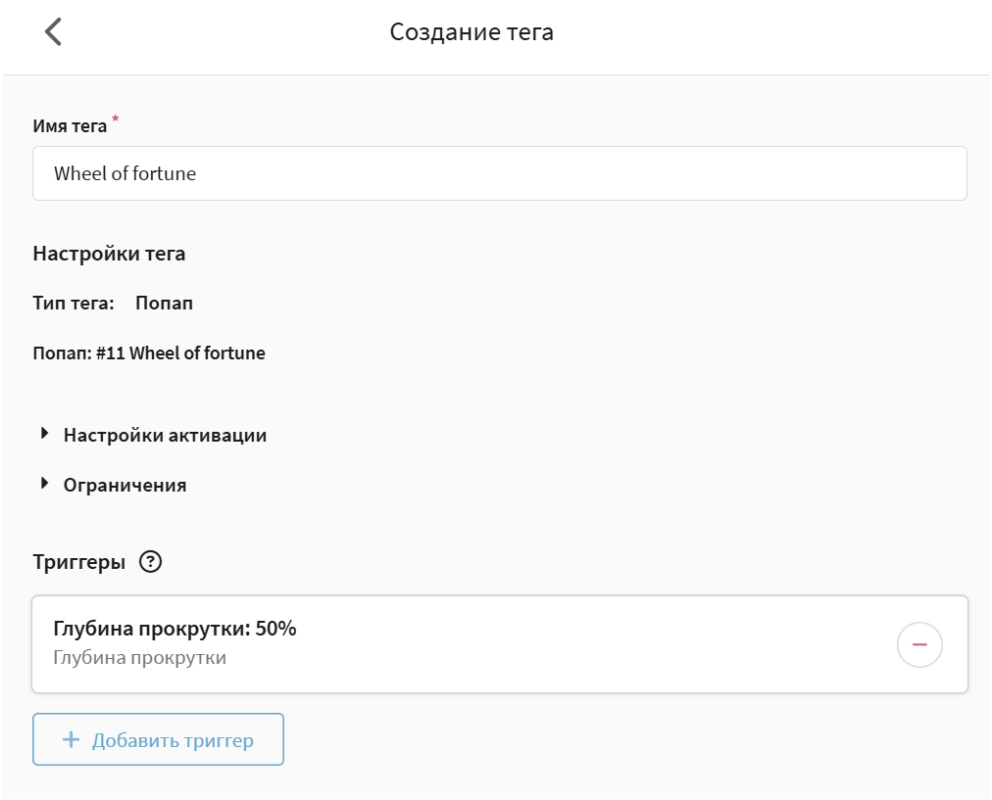
Публикация попапа

После полной настройки сохраните попап и убедитесь, что его статус **Активен**.

В разделе **Появление** привяжите попап к контейнеру Менеджера тегов:



Задайте **Имя** тега и настройте **триггер** — условие, при котором попап будет показан (например, **Таймер** или **Глубина прокрутки**):



Подробнее о настройке триггеров читайте в отдельной статье.

После полной настройки сохраните попап и убедитесь, что его статус установлен в **Активен**. После публикации попап с колесом фортуны будет готов к работе и начнёт появляться у посетителей вашего сайта согласно заданным правилам.

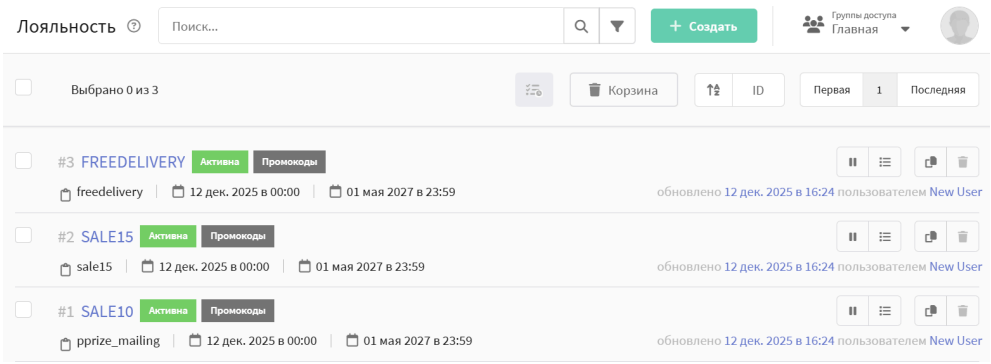
Автоматическая отправка персонализированных призов через сценарии

Для автоматической отправки промокодов, соответствующих выигранным призам, настройте сценарий с триггером на изменение поля `Prize`. Вам потребуется создать отдельные промокоды для каждого типа приза и настроить условные ветвления в сценарии.

Создание промокодов

Перейдите в **Лояльность — Создать** для создания новой программы лояльности. В настройках программы создайте **промокоды** для каждого выигрыша. Например:

- **Скидка 10%** — промокод `BOOK10`
- **Скидка 15%** — промокод `BOOK15`
- **Бесплатная доставка** — промокод `FREESHIP`
- И т. д. для каждого приза



Подробнее о промокодах можно узнать в этой статье.

Создание шаблона письма

Создайте универсальный шаблон письма, который будет показывать разный текст в зависимости от выигранного приза. Используйте **логические выражения** для проверки значения поля `Prize`. Подробнее о работе с логическими выражениями и переменными в шаблонах можно узнать в этой статье.

В шаблоне проверяется, какое значение содержит поле с призом (например `Prize`) в профиле пользователя (`{lead.Prize}`), и в зависимости от этого показывается соответствующий текст. Для каждого типа приза в текст письма нужно вставить **переменную промокода**:

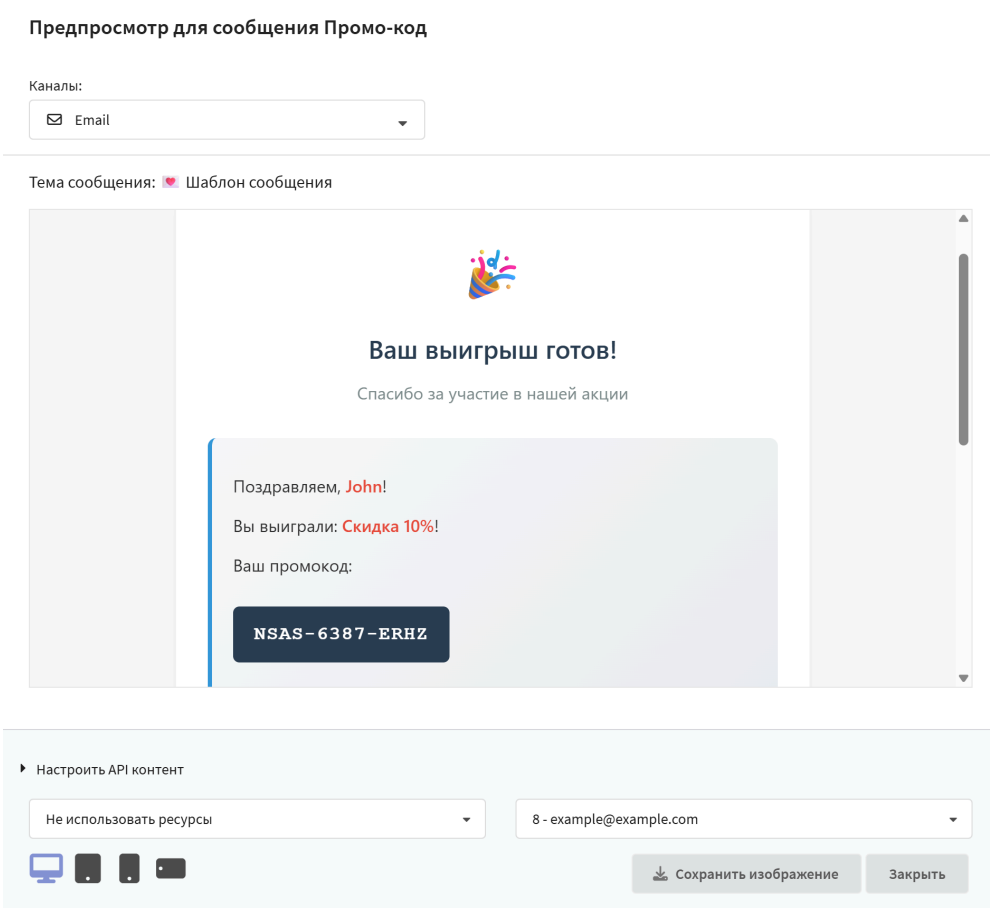
```
{if lead.Prize}

    {if lead.Prize equal "1" or lead.Prize equal "$100"}
        <p>Поздравляем, {lead._fname}!</p>
        <p>Вы выиграли денежный приз в размере $100!</p>
        <p>Ваш промокод: {loyalty.prize100.promocode}</p>
        <p>Промокод действует на все товары в нашем магазине.</p>
    {end}

    {if lead.Prize equal "2" or lead.Prize equal "$10"}
        <p>Поздравляем, {lead._fname}!</p>
        <p>Вы выиграли денежный приз в размере $10!</p>
        <p>Ваш промокод: {loyalty.prize10.promocode}</p>
        <p>Промокод действует на все товары в нашем магазине.</p>
    {end}

    {if lead.Prize equal "3" or lead.Prize equal "скидка 10%"}
        <p>Поздравляем, {lead._fname}!</p>
        <p>Вы выиграли скидку 10% на следующую покупку!</p>
        <p>Ваш промокод: {loyalty.discount10.promocode}</p>
        <p>Скидка действует на все товары в нашем каталоге.</p>
    {end}
{else}
    <p>Поздравляем, {lead._fname}!</p>
    <p>Вы выиграли приз в нашей акции!</p>
    <p>Подробности уточняйте у менеджера.</p>
{end}
```

В результате у вас получится шаблон, где для каждого условия указан свой промокод:



Создание сценария

Перейдите в раздел **Сценарии — Создать**. Выберите базу данных, в которой хранятся профили участников акции. В настройках входа установите опцию **Отключить повторный вход в сценарий для профиля клиента**. Это гарантирует, что сценарий будет запущен для каждого пользователя только один раз при каждом новом выигрыше.

После создания сценария добавьте **триггер входа**. Выберите тип триггера **Изменение профиля**, укажите поле `Prize` и установите оператор **Изменено**. Это означает, что сценарий будет запускаться каждый раз, когда в профиле пользователя появляется или изменяется значение поля с призом.

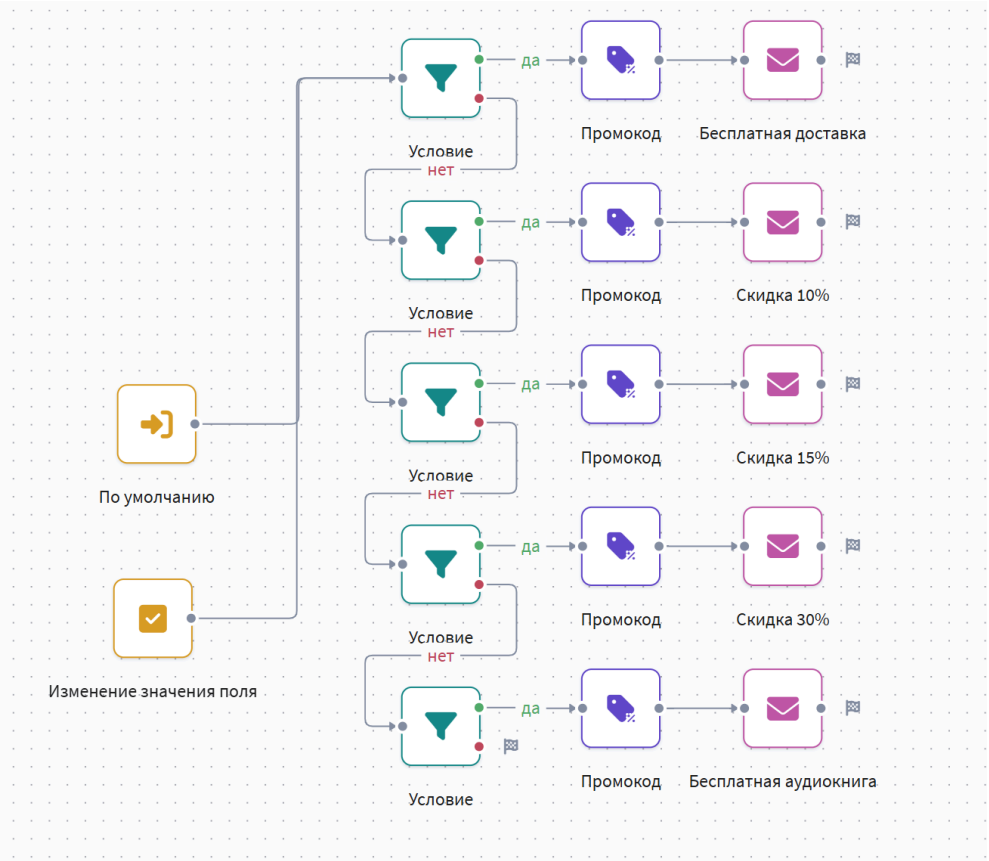
После настройки триггера добавьте в сценарий элемент **Условие**. В условии проверяйте значение поля `Prize`. Например, укажите **Поле** — `Prize`, **Оператор** — `equal`, и введите значение `1` (если в поле сохраняется ID приза) или текст первого приза, например `$100`.

Если условие выполняется (пользователь выиграл приз с ID 1 или “$100”), добавьте два узла:

- **Промокод** — выберите созданную программу лояльности и конкретный промокод, соответствующий этому призу.
- **Отправить email** — выберите созданный универсальный шаблон письма.

Если условие не выполняется, добавьте следующий элемент **Условие** для проверки второго приза. Настройте его аналогично: проверяйте, содержит ли поле `Prize` значение `2` или текст второго приза.

Повторите эту логику для каждого приза, создав цепочку условий и соответствующих действий по назначению промокодов и отправке писем:



В конце не забудьте **сохранить** и **активировать** сценарий.

Менеджер тегов

	Менеджер тегов Altcraft — это инструмент, который позволяет вам управлять HTML тегами и JavaScript кодом без необходимости привлекать разработчиков. С его помощью вы можете создавать, добавлять и обновлять фрагменты кода на сайте, чтобы отслеживать действия пользователей и передавать данные в платформу.

TIP

Менеджер тегов Altcraft полностью совместим с Google Tag Manager (GTM). Вы можете использовать оба инструмента на одном сайте без риска возникновения конфликтов.

Основные понятия

Контейнер — это оболочка, в которой хранятся теги, триггеры и переменные. Он предоставляет код, который необходимо разместить на сайте для активации тегов.

Тег — это фрагмент кода, который выполняет определенную функцию (например, показ попапа посетителям сайта).

Триггер — это условие срабатывания тега. Триггеры отслеживают определенные события на сайте (например, клики или загрузка страницы) и активируют теги в ответ на них. Для запуска тега должен быть настроен хотя бы один триггер.

Переменная — это параметр, которому передаются определенные значения. Переменные могут применяться как в триггерах, так и в тегах. В триггерах они используются в качестве фильтра активации, то есть определяют, когда следует запускать тот или иной тег. Переменные в тегах используются для получения динамических значений.

В этом разделе

Настройка и установка Менеджера тегов

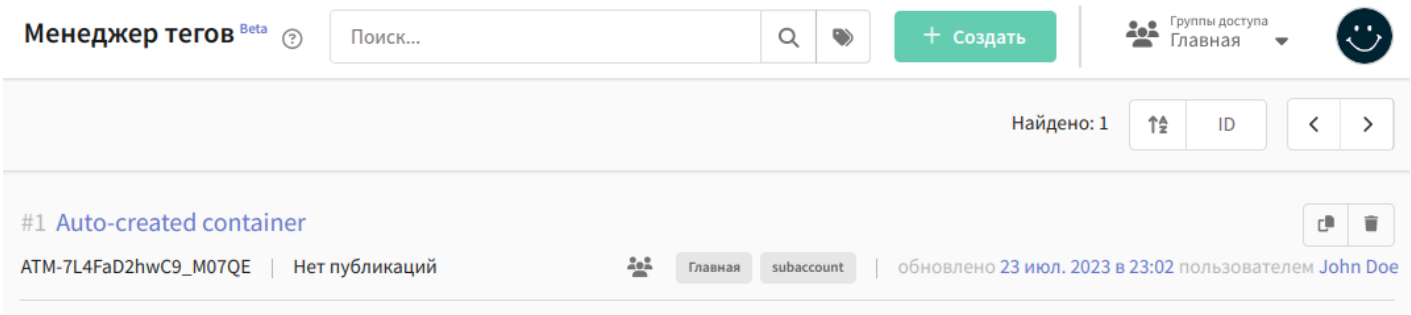
Чтобы начать работу с Менеджером тегов, перейдите в раздел **Веб-слой → Менеджер тегов**.

Шаг 1. Создание контейнера и его установка на сайт

TIP

Контейнер хранит теги, триггеры и переменные. Он представляет собой фрагмент кода, который устанавливается на сайте. После установки все последующие настройки отслеживания проводятся в интерфейсе платформы, без необходимости внесения изменений в исходный код.

По умолчанию в вашем аккаунте уже создан контейнер. Вы можете работать в нём или добавить новый. Для каждого сайта или мобильного приложения рекомендуется настраивать отдельный контейнер.



Из меню действий в правой части доступны следующие действия:

— Клонирование контейнера: создаётся новый контейнер, в который переносятся настройки, теги, триггеры и переменные текущего контейнера (история версий не переносится).

— Удаление контейнера со всеми созданными тегами, триггерами и переменными. Теги из этого контейнера больше не будут активироваться на сайте.

Создание нового контейнера

Для создания нового контейнера выполните следующие действия:

1. Нажмите кнопку **+ Создать** в правом верхнем углу.

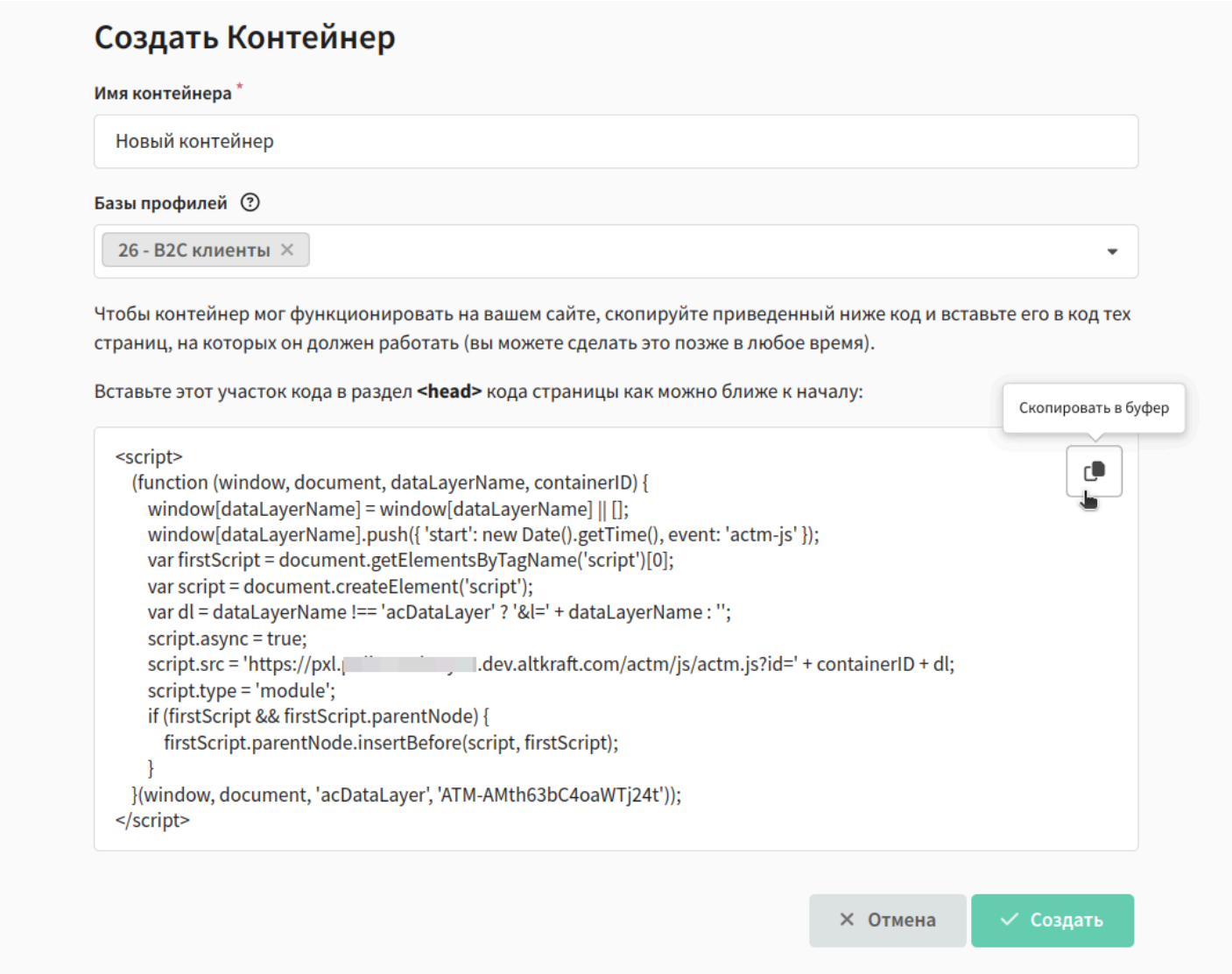
2. Введите произвольное **название** контейнера.

3. Выберите **базы профилей**. Теги отслеживают поведение посетителей сайта и передают данные в систему Altcraft. Полученные данные могут быть использованы для поиска уже существующих профилей в базе. Укажите как минимум одну базу для поиска.

4. **Установите контейнер на сайте**. Для этого скопируйте предоставленный код и передайте его разработчикам. Они должны разместить код контейнера на каждой странице сайта, где вы планируете

использовать теги отслеживания. Код контейнера следует добавить в раздел `<head>`, как можно ближе к открывающему тегу.

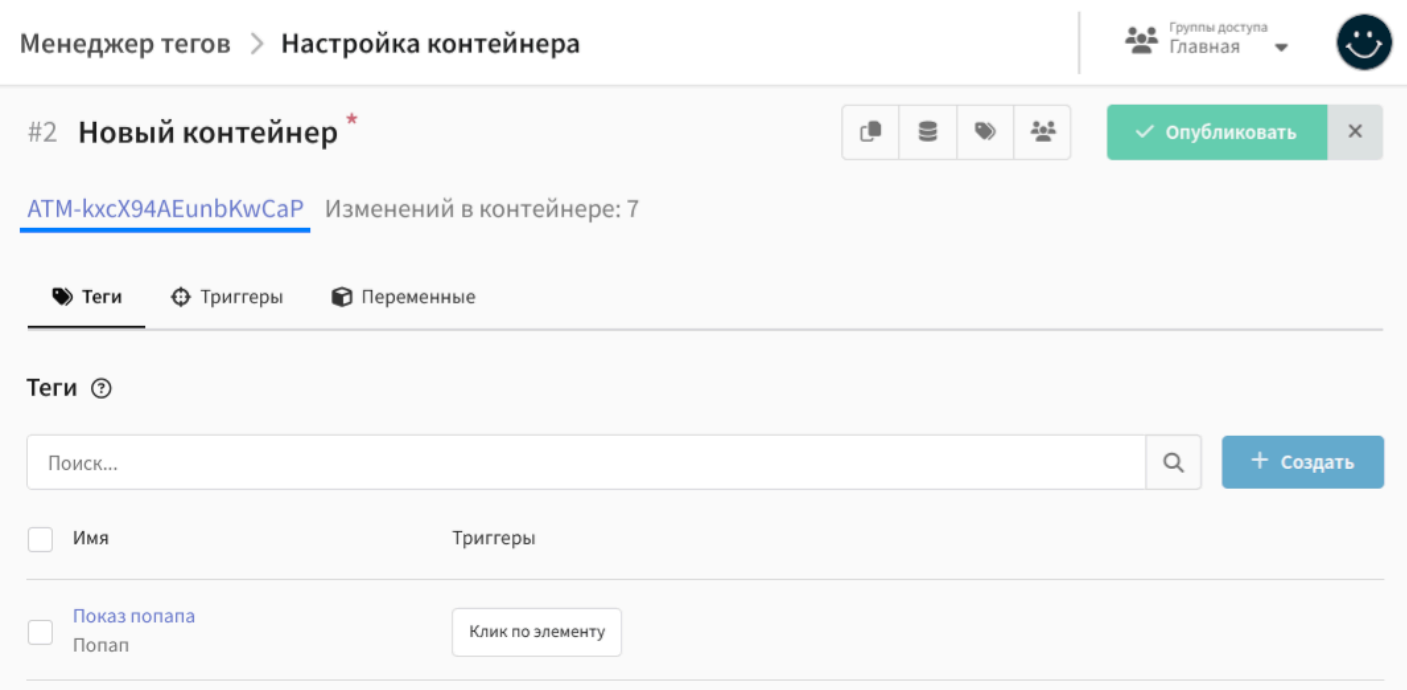
5. Нажмите кнопку **“Создать”**, чтобы сохранить контейнер.



Работа в контейнере

Чтобы перейти к редактированию существующего контейнера, нажмите на его название. При клике на идентификатор контейнера (ATM-xxxxxxx) откроется окно с кодом. Если вы еще не разместили контейнер на сайте, передайте код разработчикам.

Справа от идентификатора находится счётчик, который показывает количество неопубликованных изменений. После публикации версии счётчик обнуляется.



Для дополнительных настроек справа доступно меню быстрых действий:

— просмотр истории версий

— выбор баз для поиска профилей

— назначение тегов для быстрого поиска

— установка групп доступа

INFO

Любые изменения в настройках контейнера (редактирование имени или групп доступа, настройка тегов и триггеров) автоматически сохраняются в платформе, но не вносятся на сайт до тех пор, пока вы не опубликуете контейнер.

Шаг 2. Добавление тегов и настройка триггеров

TIP

Тег — это фрагмент кода, который выполняется на сайте при определенном условии. **Триггер** — это условие срабатывания тега.

Триггеры отслеживают определенные события на сайте (например, клики, загрузку страницы или отправку формы) и активируют теги в ответ на эти события.

Для запуска тега должен быть настроен хотя бы один триггер.

Чтобы создать тег и настроить условия его запуска, выполните следующие действия:

1. На вкладке **“Теги”** нажмите **+ Создать**.

2. Укажите **название** тега.

3. Выберите **тип** тега:

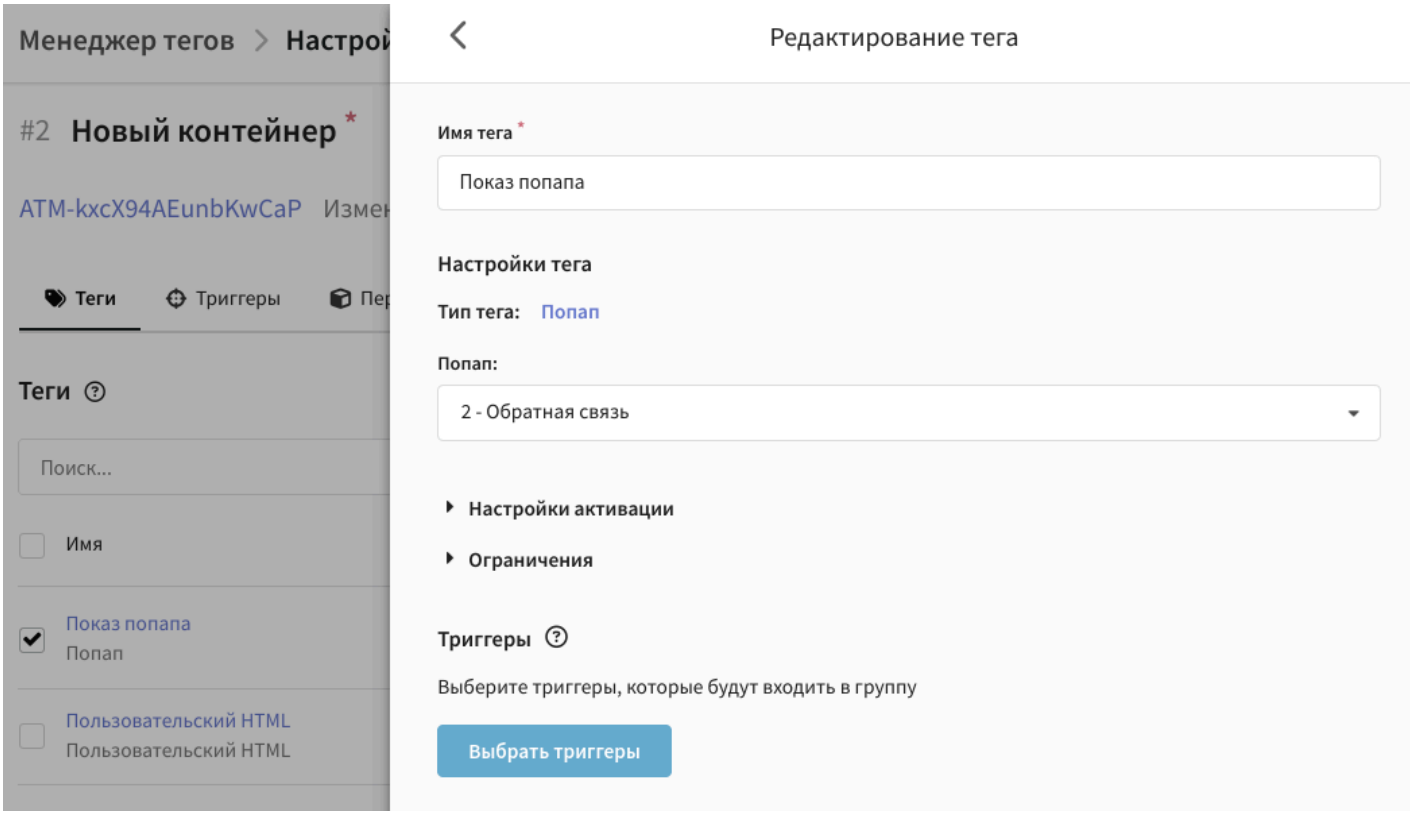
- **Попап** — используется для показа посетителям сайта всплывающего окна. Предварительно необходимо создать попап, а затем выбрать его в поле **“Попап”**. Подробнее о попапах читайте здесь.
- **Пользовательский HTML** — используется для создания собственного тега. Необходимо вставить ваш HTML-код в окно редактора.

NOTE: Использование переменных в коде тега

Переменные контейнера можно использовать в HTML или JavaScript коде тега для динамической подстановки значений. Для этого применяйте синтаксис `{{Имя переменной}}`.

```
<h2>{{Заголовок баннера}}</h2>
<script>
console.log('Текущая страница:', {{Page URL}});
</script>
```

Подробнее о работе с переменными вы можете узнать в этой статье

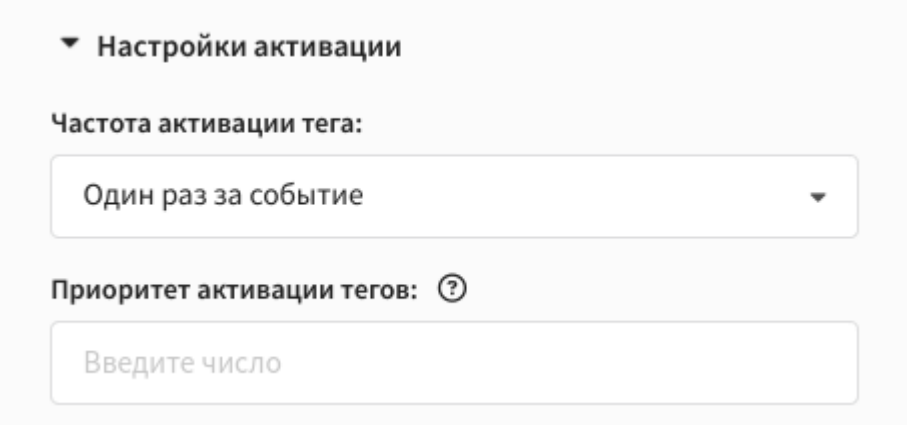


4. Задайте **настройки активации** тега и установите **ограничения**.

Настройки активации тега

Частота активации тега:

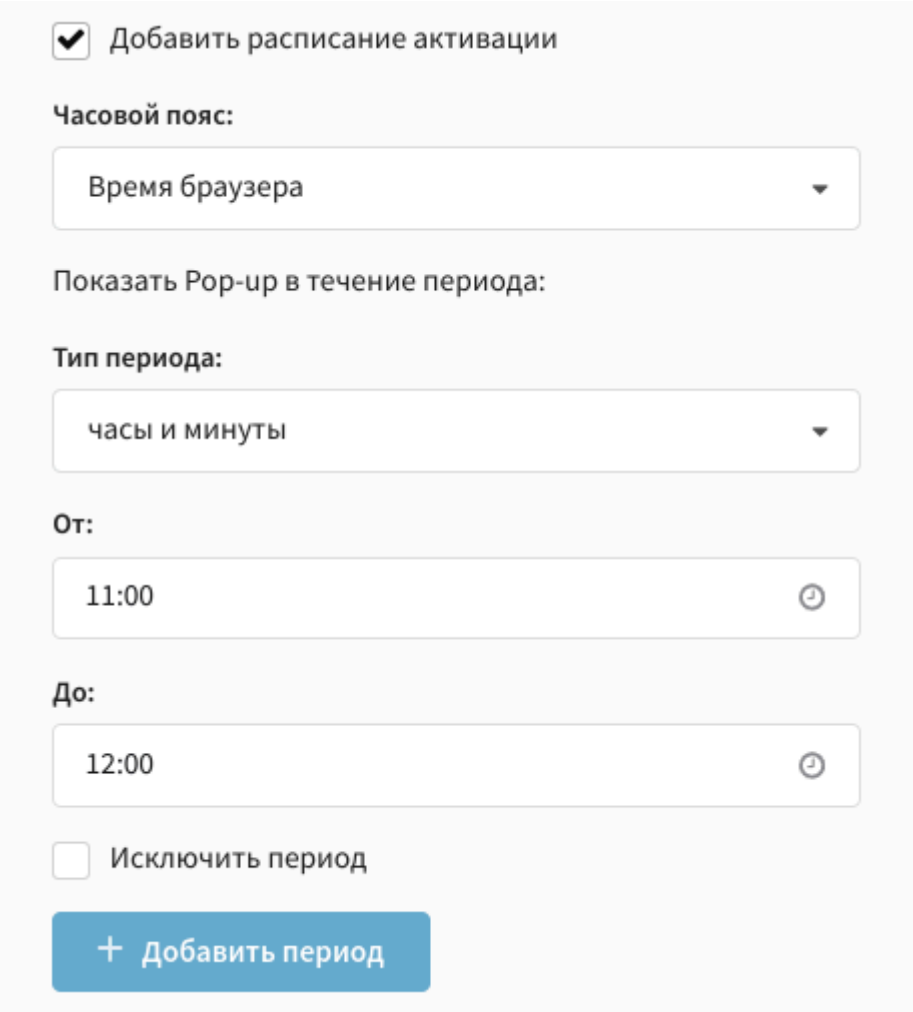
- **Один раз за событие** — тег активируется только один раз при выполнении определенного события.
- **Один раз за страницу** — тег сработает только один раз на страницу, независимо от количества событий.
- **Без ограничений** — тег активируется при наступлении каждого события, заданного в триггере.



Приоритет активации тегов. Устанавливает очередность срабатывания тегов. По умолчанию приоритет тега равен 0. Чтобы изменить очередность, укажите положительное или отрицательного целое число. Например, тег с приоритетом 2 сработает раньше тега с приоритетом 1. Теги запускаются асинхронно.

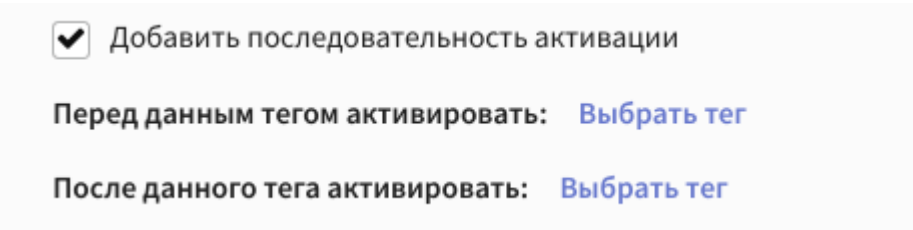
Добавить расписание активации. Позволяет настроить активацию тега в течение определенного периода. Например, вы проводите рекламную акцию, которая длится всего один час. В таком случае баннер с информацией необходимо показывать посетителям сайта только в течение часа.

Выберите **тип периода**, задайте **время/дату начала** и **время/дату окончания**. Если вы, напротив, хотите, чтобы тег оставался неактивным в определенный период, установите флаг в поле **“Исключить период”**. Вы также можете объединить несколько временных периодов — нажмите кнопку **“Добавить период”**.



Добавить последовательность активации:

- **Перед данным тегом активировать.** Выберите тег, который будет активирован перед текущим тегом. Если выбранный тег не сработает, то тег, который вы создаете в данный момент, не будет активирован.
- **После данного тега активировать.** Выберите тег, который будет активирован после выполнения текущего тега. Если тег, который вы создаете в данный момент, не сработает, то выбранный тег не будет активирован.

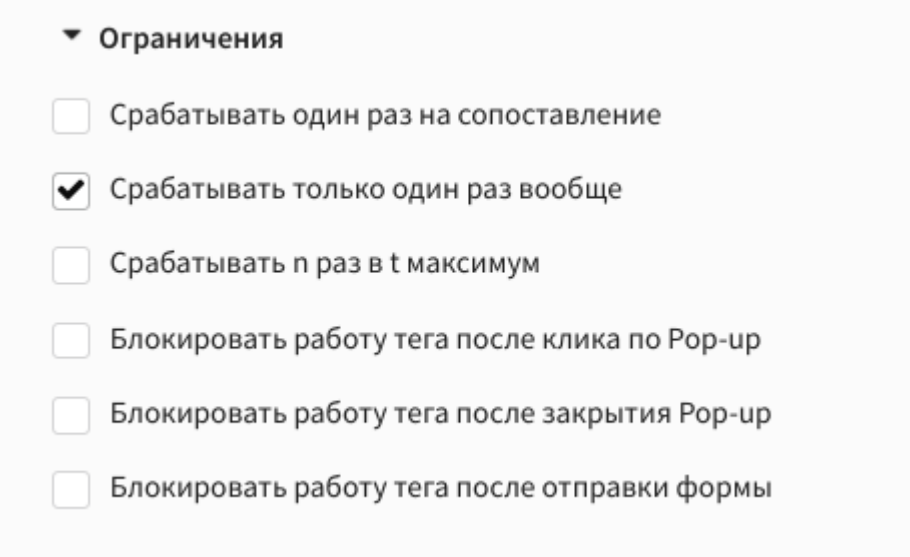


Ограничения тега

Ограничения

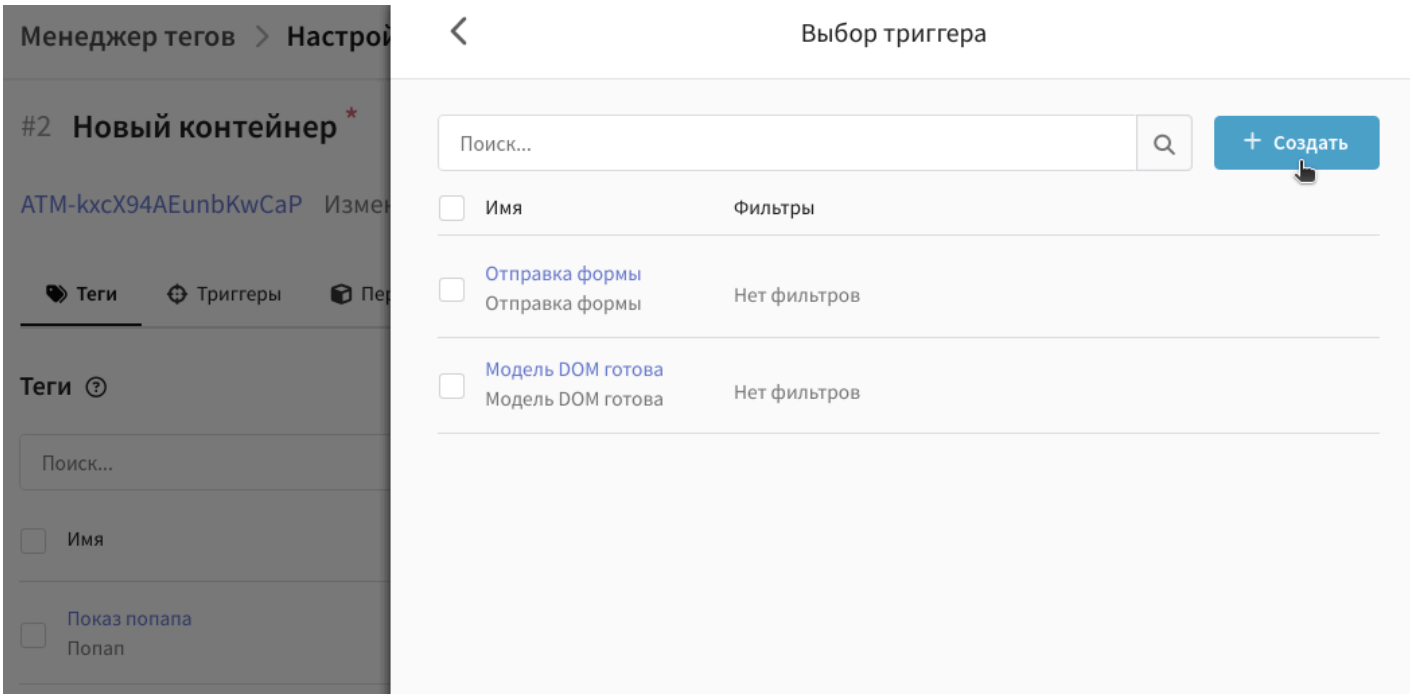
- **Срабатывать один раз на сопоставление** — тег будет выполнен только один раз при каждом срабатывании условия, которое соответствует его настройкам. Если условие снова становится истинным, тег не будет повторно выполнен. Это относится к условиям, связанным с сопоставлением, таким как URL страницы, заголовок страницы и другие параметры, которые могут изменяться во время просмотра страницы.

- **Срабатывать только один раз вообще** — тег будет выполнен только один раз, независимо от того, какие условия срабатывают. Это ограничение можно использовать для инициализации скриптов, которые должны быть выполнены только один раз.
- **Срабатывать n раз в t максимум** — ограничивает количество срабатываний триггера за определенный промежуток времени.
- **Блокировать работу тега после клика по pop-up** — после нажатия на любой элемент попапа, тег не будет срабатывать повторно, даже если событие, на которое он реагирует, происходит снова. Это предотвращает множественные срабатывания тега при повторных кликах на элементы попапа.
- **Блокировать работу тега после закрытия pop-up** — после нажатия на кнопку “Закрыть” в попапе триггер не будет срабатывать повторно, несмотря на наступление события.
- **Блокировать работу тега после клика отправки формы** — после отправки формы, связанной с тегом, триггер не будет срабатывать повторно, даже если условия для его срабатывания снова выполняются.



5. Нажмите **“Выбрать триггеры”**. Откроется новое окно со всеми доступными триггерами. Если у вас уже создан триггер, можно перейти сразу к пункту 9.

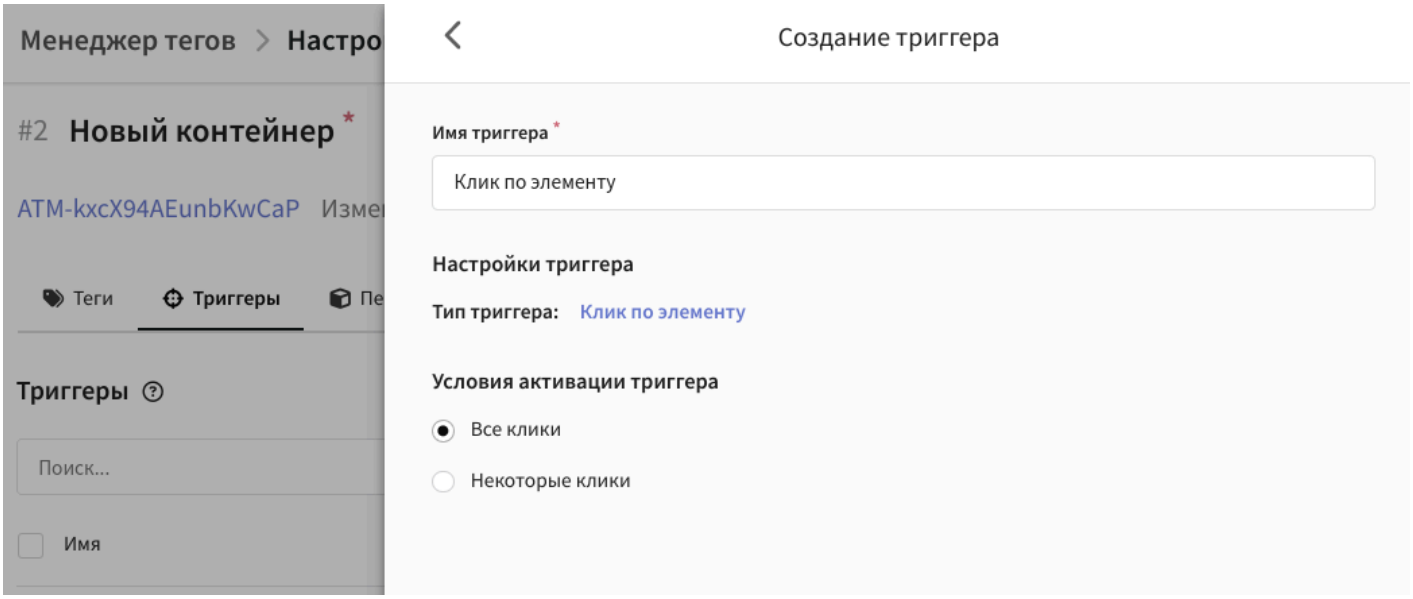
6. Чтобы настроить новый триггер, нажмите **+ Создать**:



7. Введите **название** триггера и выберите его **тип**.

TIP

Подробнее о типах триггера читайте в [этой статье].



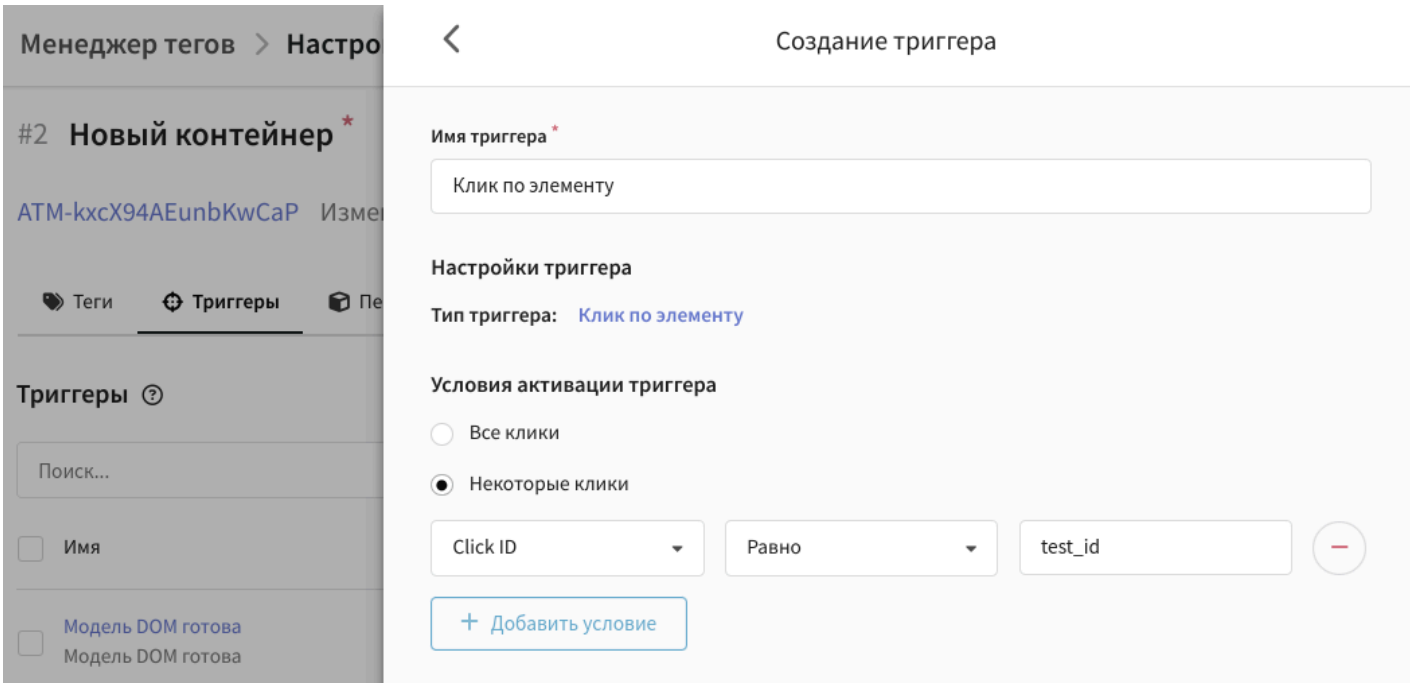
8. По умолчанию триггеры активируются в ответ на все события определенного типа. Чтобы задать дополнительные условия активации триггера, вы можете настроить фильтр. Для этого в блоке **“Условия активации триггера”** выберите **“Некоторые события”**. После настройки фильтра **сохраните** триггер с помощью кнопки внизу страницы.

INFO

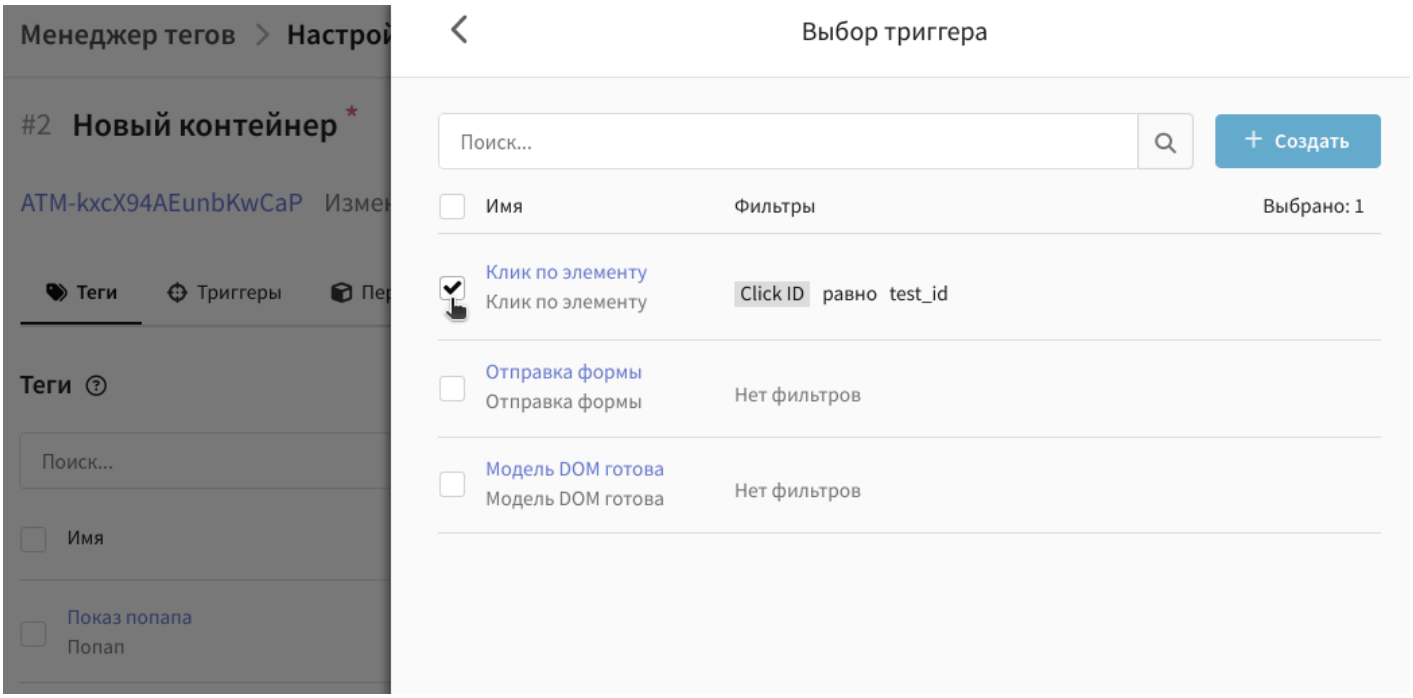
Фильтр состоит из **переменной**, **оператора** и **значения**. Выберите переменную (встроенную или пользовательскую) и оператор. Затем выберите или введите вручную текстовое значение переменной.

Каждое событие на сайте связано с набором переменных, которые содержат информацию о нём. Когда происходит событие, триггер считывает определенную переменную. Менеджер тегов сравнивает значение этой переменной с тем значением, что задано в фильтре. Если значение переменной соответствует заданным в фильтре параметрам, триггер активируется и запускает выполнение тега.

Таким образом, с помощью фильтра вы можете ограничить аудиторию, для которой будет активирован тег.



9. В окне **“Выбор триггера”** выделите один или несколько триггеров и нажмите кнопку **“Выбрать”** внизу страницы. При выборе нескольких триггеров тег будет срабатывать при активации любого из них.

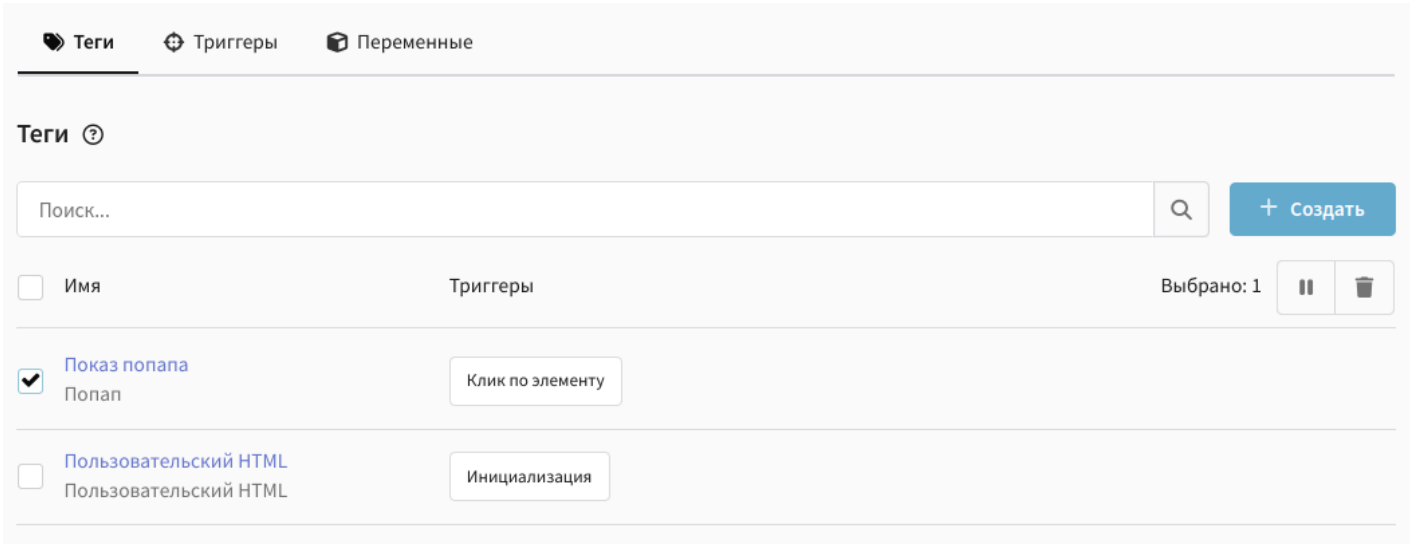


10. В окне **“Создание тега”** проверьте заданные настройки и нажмите **“Сохранить”**.

Созданный тег отобразится на вкладке **“Теги”**. Если вам потребуется изменить его настройки, нажмите на его название. Чтобы временно остановить работу тегов на странице сайта, выделите теги и нажмите

, а затем опубликуйте новую версию контейнера. Вы также можете удалить лишние теги из контейнера

.



Шаг 3. Публикация версии контейнера

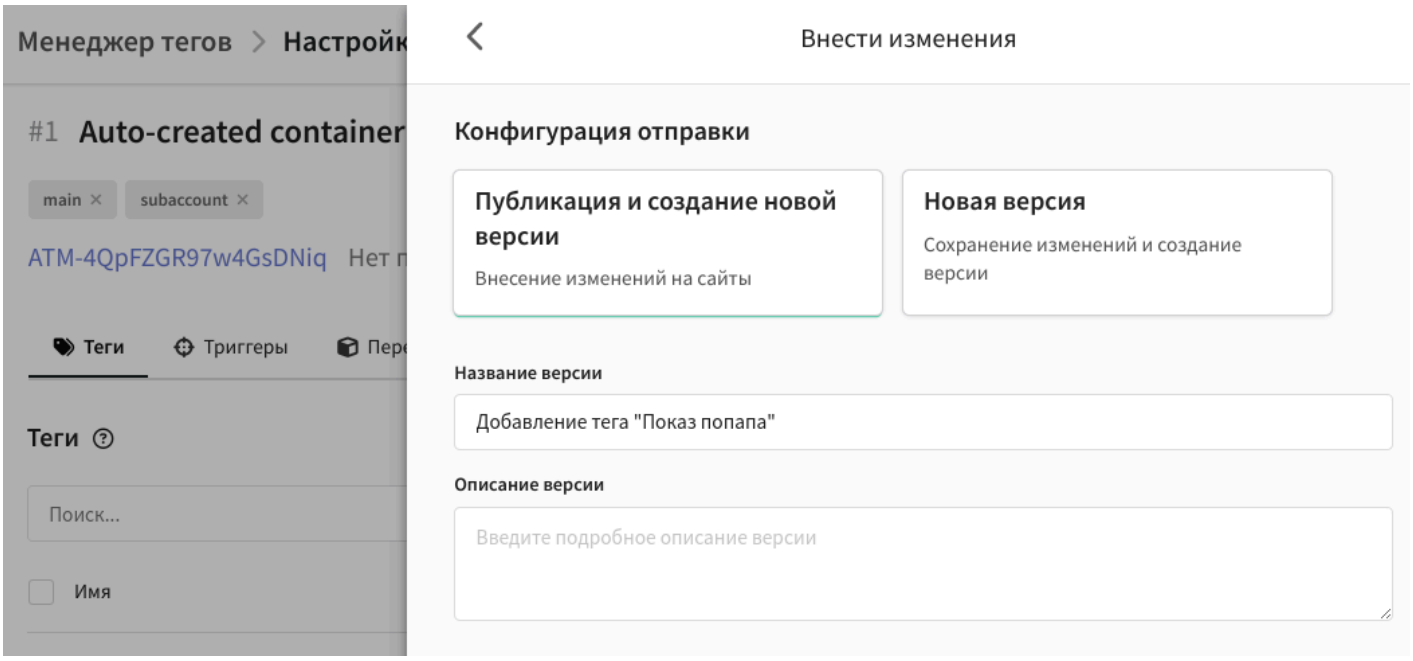
Версия – это сохранённая копия контейнера в текущий момент времени. Если вы внесли какие-то изменения в контейнер (добавили тег, изменили триггер), можно сохранить эти изменения в виде версии. Для этого в окне **“Настройка контейнера”** нажмите кнопку **“Опубликовать”**. Доступны две **конфигурации отправки**:

- **Публикация и создание новой версии** — создаётся новая версия контейнера, и все изменения вносятся на сайт.
- **Новая версия** — создаётся новая версия, в которой сохраняются все изменения, но контейнер не публикуется, и изменения на сайт не вносятся.

CAUTION: Проверка переменных перед публикацией

Перед публикацией система проверяет, что все переменные, используемые в HTML-тегах, существуют в контейнере. Если будут найдены неизвестные переменные (например, удалённые или переименованные), публикация будет отменена с сообщением об ошибке.

Рекомендуется указать название и описание версии — так вы зафиксируете, какие именно изменения были внесены. Затем подтвердите сохранение с помощью кнопки **“Опубликовать”** внизу страницы.

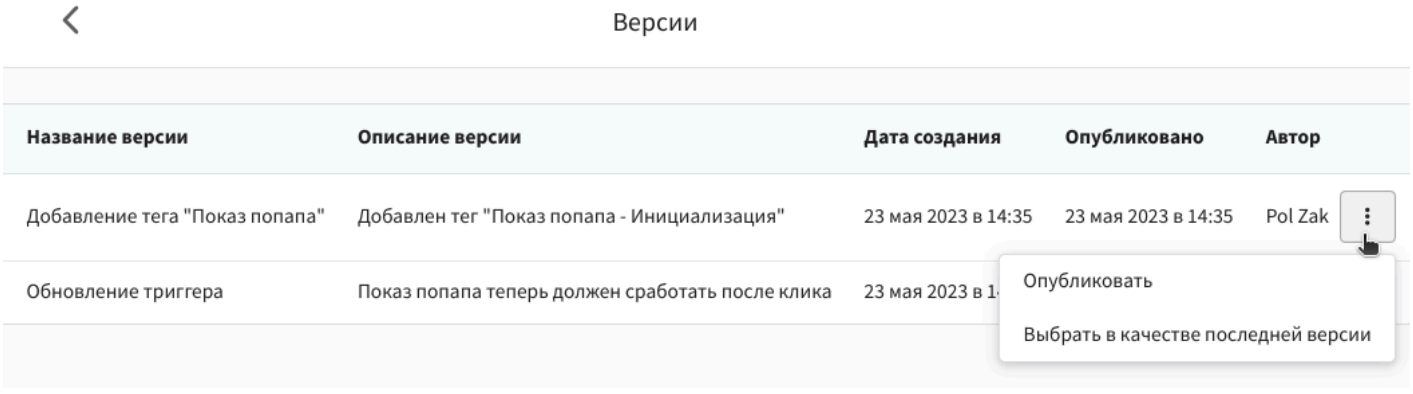


Чтобы перейти к управлению версиями контейнера, воспользуйтесь кнопкой

в меню быстрых действий. В таблице отображается информация о том, какой пользователь и когда создал или опубликовал ту или иную версию.

Вы можете переключиться на другую версию или опубликовать её вместо текущей. Для этого откройте меню дополнительных действий

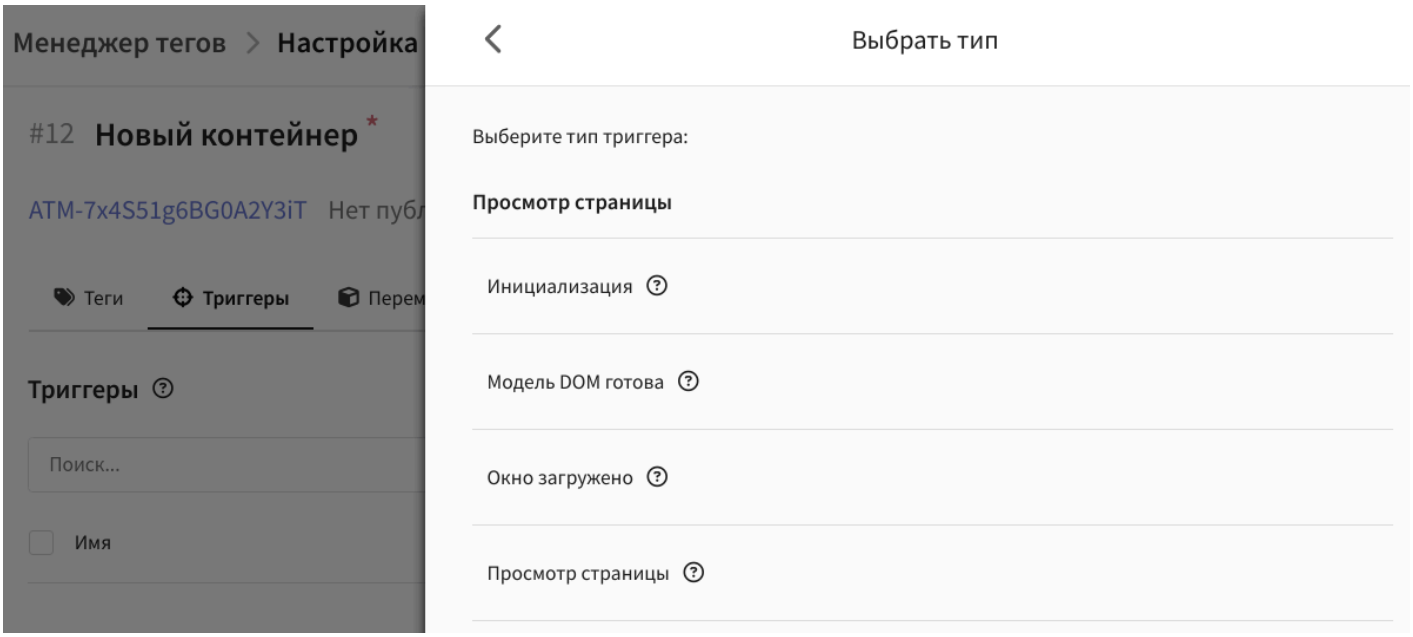
и нажмите **“Выбрать в качестве последней версии”**:



Первая версия контейнера опубликована, и на вашем сайте установлен Менеджер тегов Altcraft. Не забудьте проверить корректность работы тега.

Типы триггеров

Просмотр страницы



Инициализация

Триггер срабатывает перед активацией всех остальных триггеров. Используйте этот триггер, чтобы он активировал нужные теги раньше других триггеров.

Модель DOM готова

Триггер срабатывает после завершения обработки HTML-кода страницы браузером.

Браузер полностью загрузил HTML, было построено DOM-дерево, но внешние ресурсы, такие как изображения `<img>` и стили, могут быть ещё не загружены.

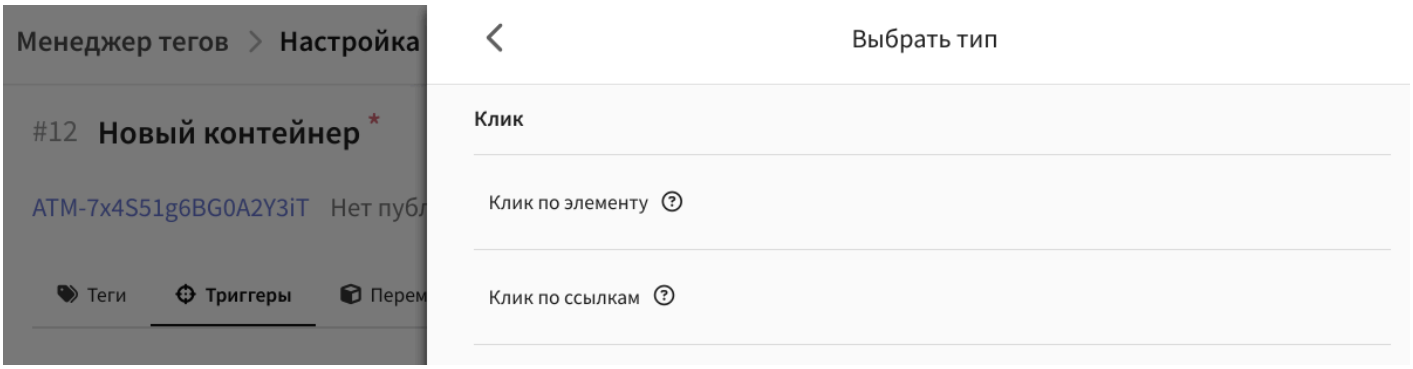
Окно загружено

Триггер срабатывает после полной загрузки страницы, включая изображения, скрипты и другие встроенные объекты.

Просмотр страницы

Триггер срабатывает, когда страница начинает загружаться. Используется для получения данных только по показам страницы.

Клик



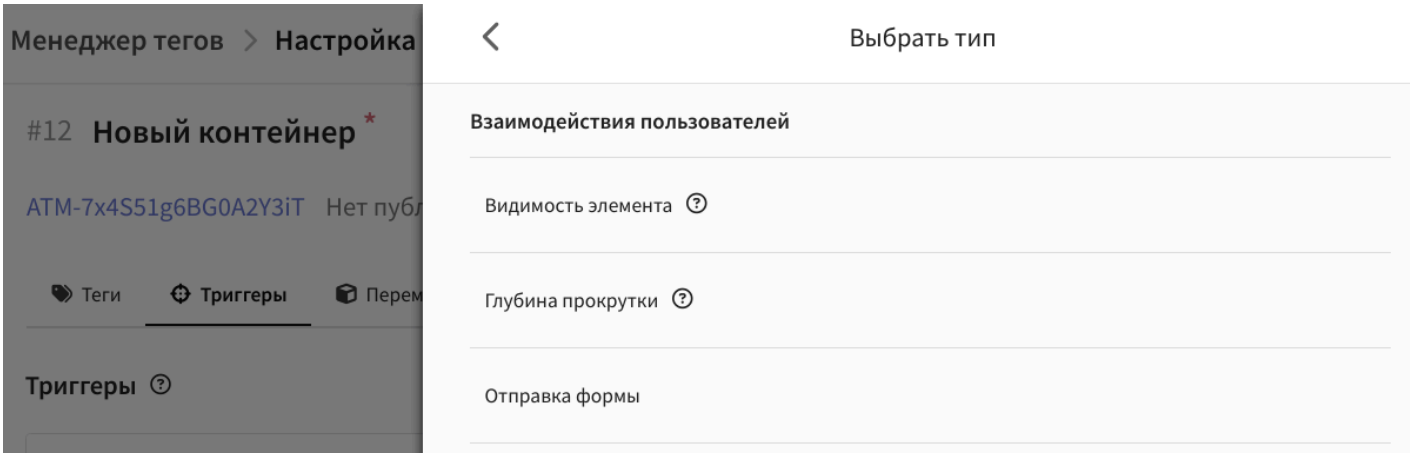
Клик по элементу

Триггер срабатывает при клике на любой элемент страницы, включая ссылки, изображения, кнопки и др.

Клик по ссылкам

Триггер срабатывает при клике на HTML-ссылки, например, `<a hrefs="https://example.com"> Ссылка </a>`

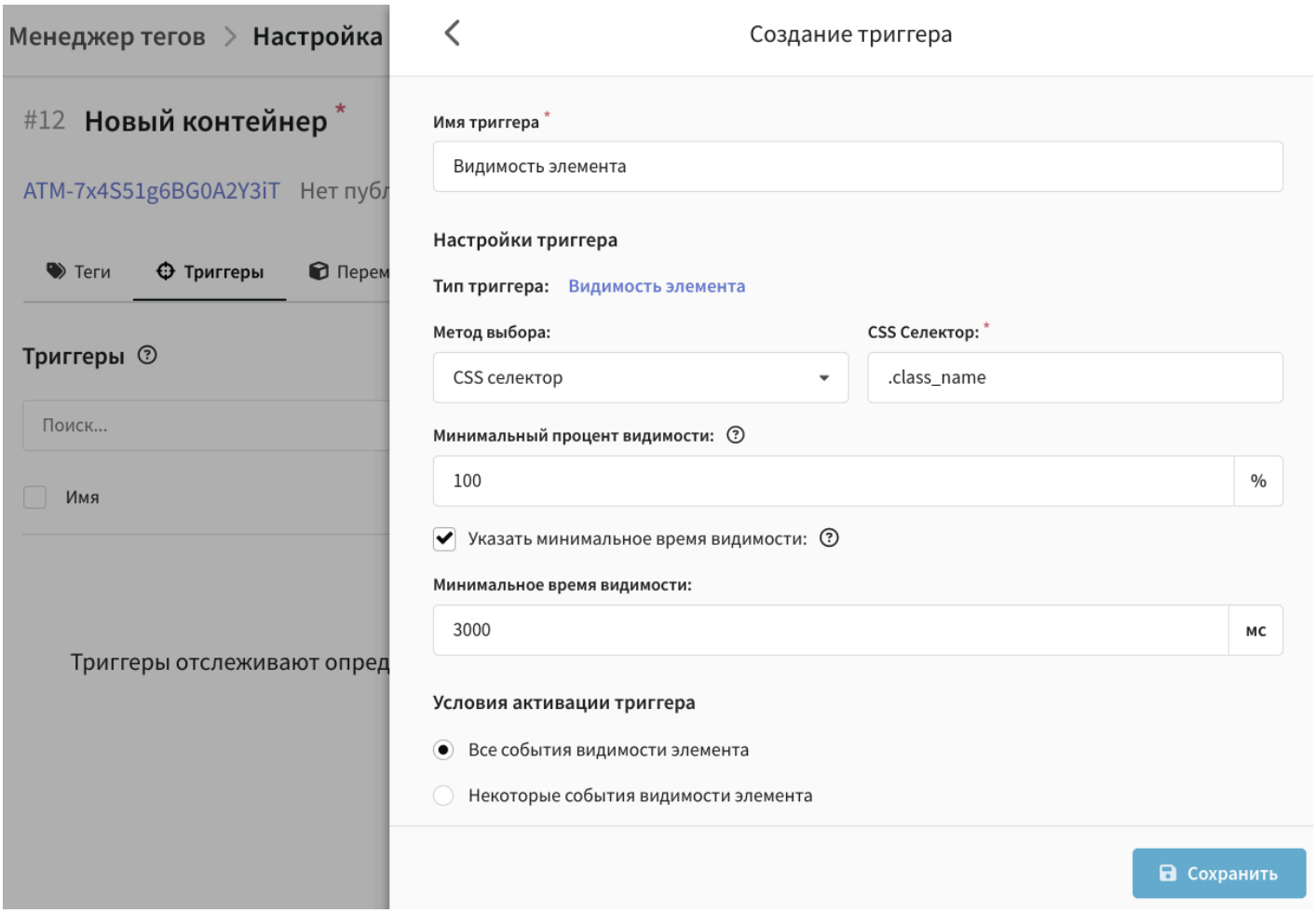
Взаимодействия пользователей



Видимость элементов

Триггер срабатывает, когда определенный элемент становится видимым в области просмотра.

Видимость элемента может измениться в результате различных действий: загрузка страницы, прокрутка, переключение на фоновую вкладку браузера, изменение позиции элемента через код и др.



Метод выбора:

- **Идентификатор** — используется, чтобы отслеживать видимость одного элемента с определенным значением атрибута `id`. Например, если указать `test`, то будет выбран элемент с `id="test"`.
- **CSS селектор** — используется, чтобы отслеживать видимость нескольких элементов, которые соответствуют заданному шаблону CSS селектора. Например, `.class_name` выберет элементы на странице с `class="class_name"`.

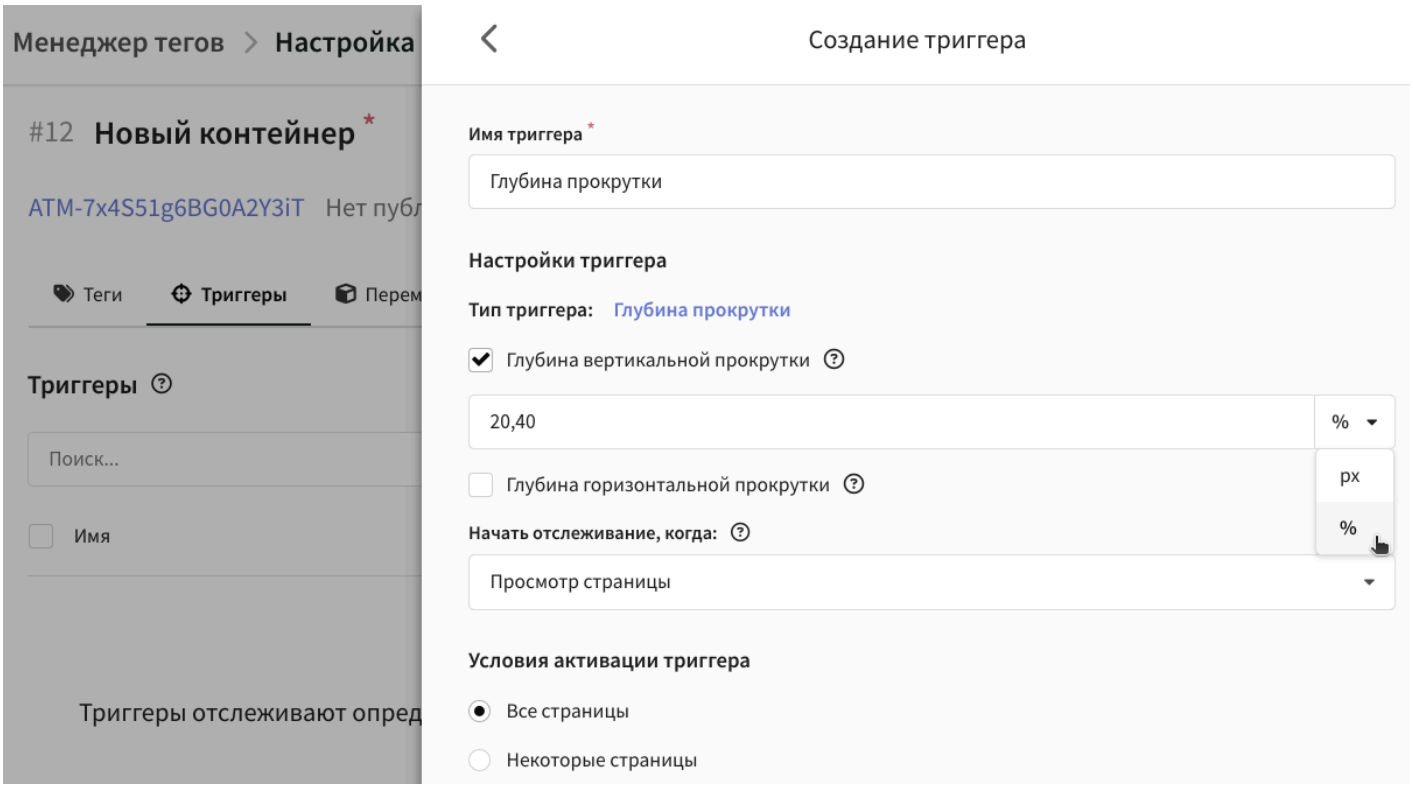
Минимальный процент видимости — процент выбранного элемента, который должен быть виден на экране пользователя для того, чтобы сработал триггер. По умолчанию — 100%, то есть элемент должен полностью отобразиться на экране.

Также можно **указать минимальное время видимости**. Минимальное время видимости — это период времени, в течение которого элемент должен оставаться видимым на экране для того, чтобы сработал триггер.

> **TIP**
>
> Обратите внимание, что периоды времени суммируются. Если элемент был виден в течение 5000 мс, затем скрыт и снова виден в течение 3000 мс, то время видимости составит 8000 мс.

Глубина прокрутки

Триггер срабатывает в зависимости от того, насколько пользователь скроллит страницу вниз или вправо.



Для настройки триггера необходимо выбрать хотя бы один из параметров прокрутки:

- **Глубина вертикальной прокрутки** — триггер активируется на основе данных о том, насколько пользователь прокрутил страницу вниз.
- **Глубина горизонтальной прокрутки** — триггер активируется на основе данных о том, насколько пользователь прокрутил страницу вправо.

Можно использовать оба параметра одновременно. Значение глубины прокрутки указывается в пикселях или в процентах (от высоты и ширины страницы). Если вы указываете несколько значений через запятую, то триггер активируется при достижении каждого из них.

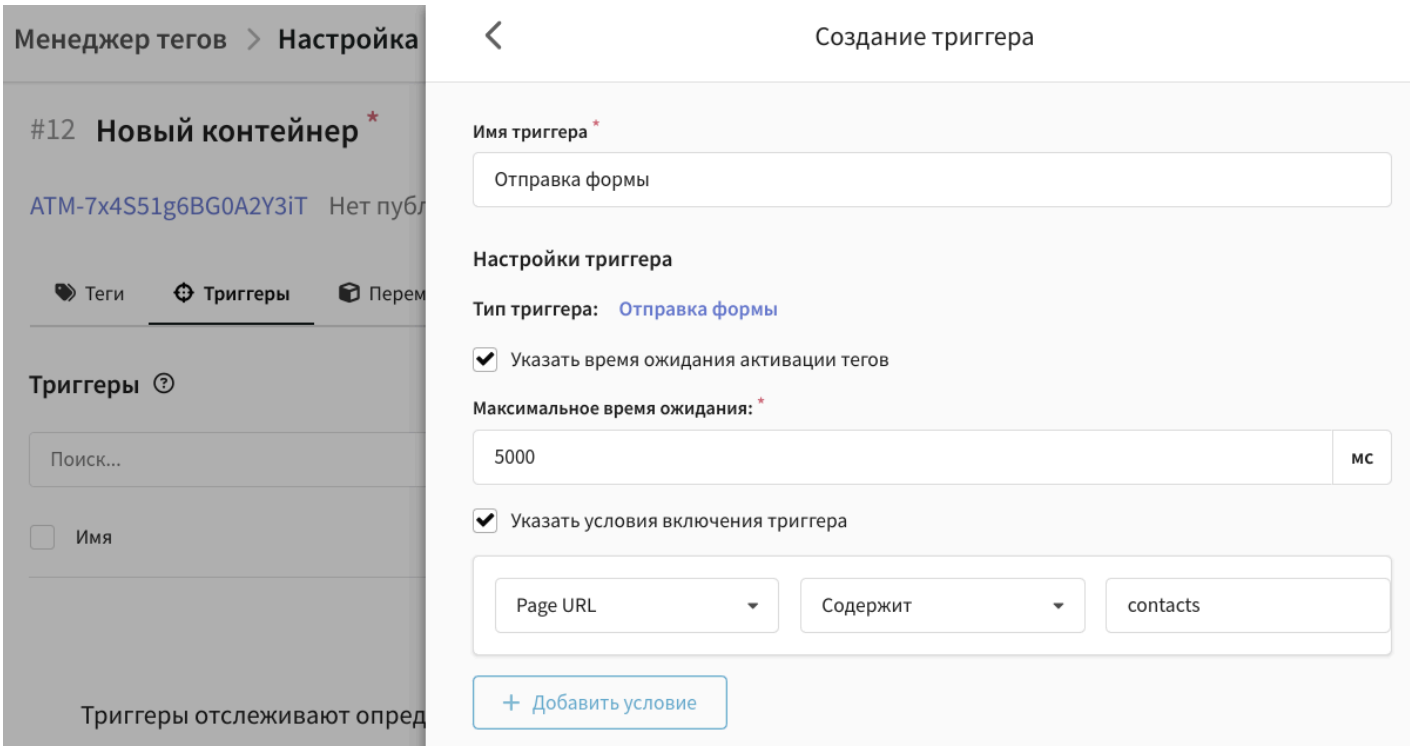
Также можно указать, на каком этапе загрузки страницы вы хотите **начать отслеживать** событие:

- Окно загружено
- Модель DOM готова
- Просмотр страницы

> **TIP**
>
> Если указанная глубина прокрутки входит в область просмотра при загрузке страницы, триггер срабатывает без прокручивания.

Отправка формы

Триггер срабатывает при каждой попытке пользователя отправить форму.



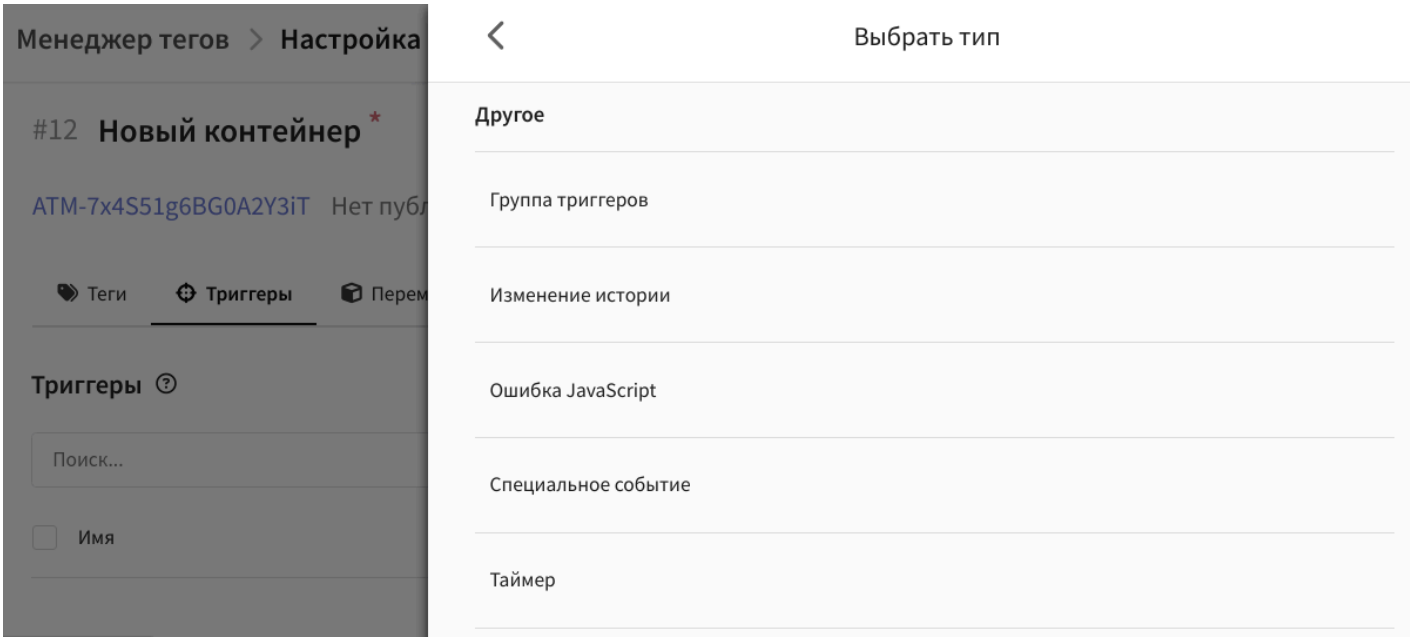
В настройках триггера вы можете:

- **Указать время ожидания активации тегов:** отправка формы блокируется до тех пор, пока не истечет указанное время ожидания или пока не будут активированы все теги, зависящее от этого триггера. Время ожидания задается в миллисекундах (5000 мс = 5 сек).
- **Указать условия включения триггера:** с помощью переменных выберите, при каких условиях триггер должен отслеживать события.

TIP

В триггере рекомендуется использовать фильтр с указанием URL страницы, на которой размещена форма.

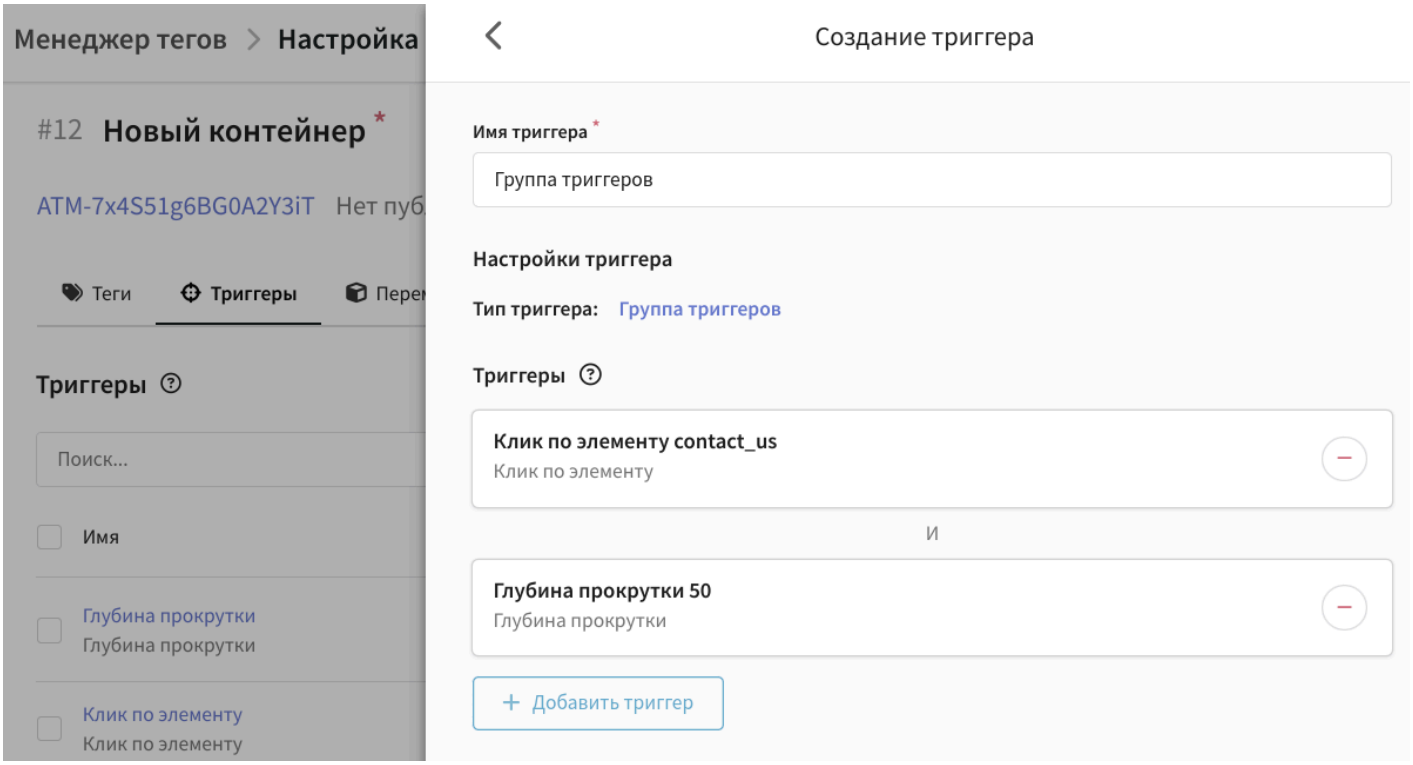
Другое



Группа триггеров

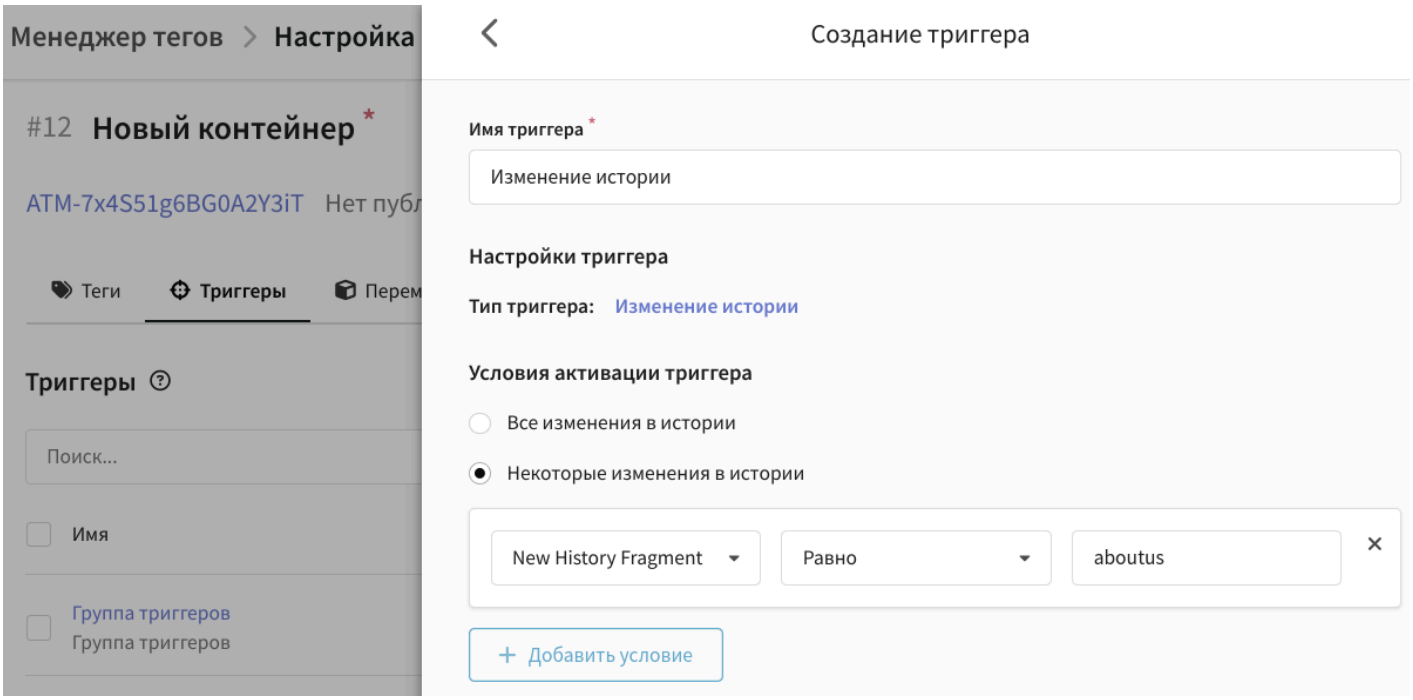
Триггер срабатывает после того, как все выбранные теги были активированы хотя бы один раз.

С помощью кнопки **“Добавить триггер”** выберите необходимые триггеры. Количество триггеров в группе не ограничено.



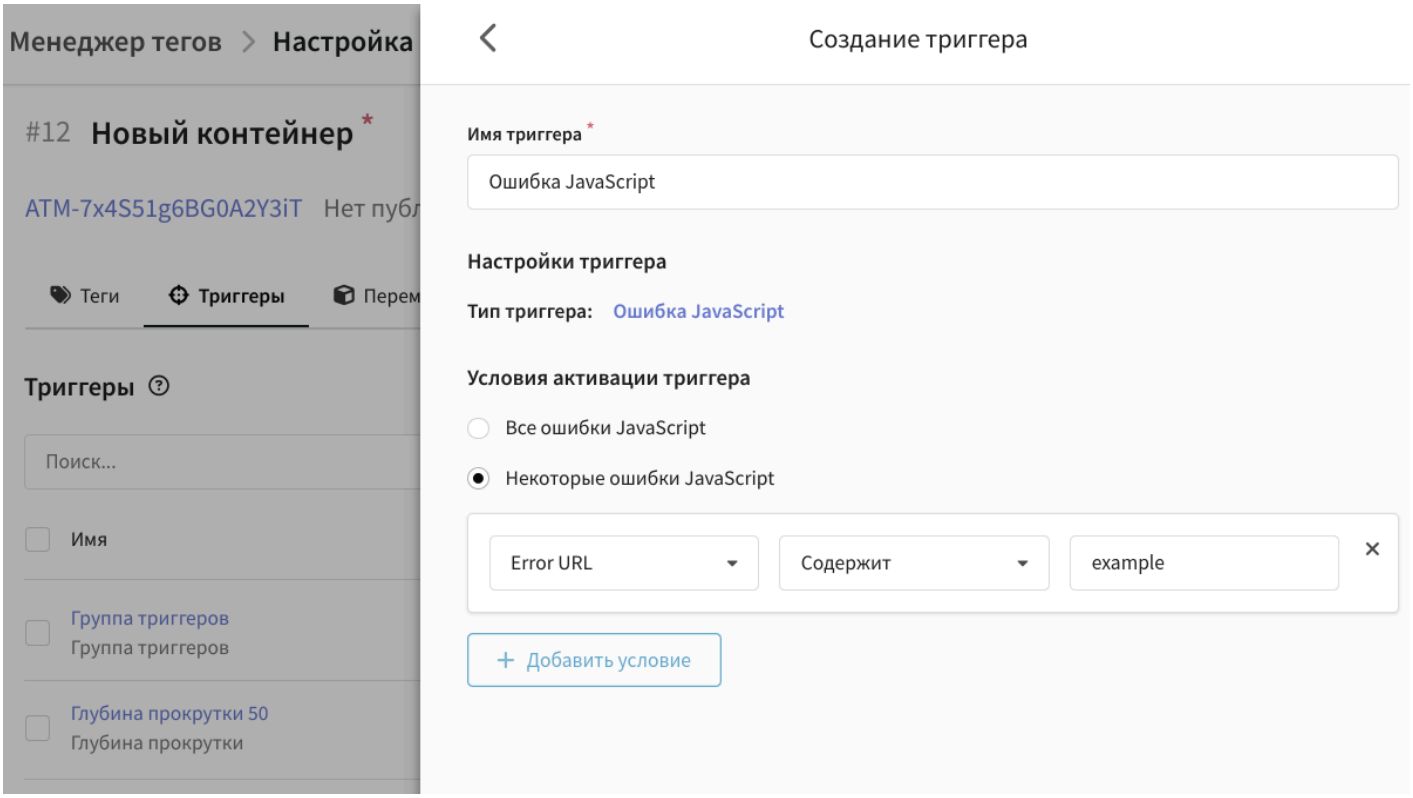
Изменение истории

Триггер срабатывает при изменении в фрагменте URL (хеше) или при использовании API pushState HTML5 на сайте.



Ошибка JavaScript

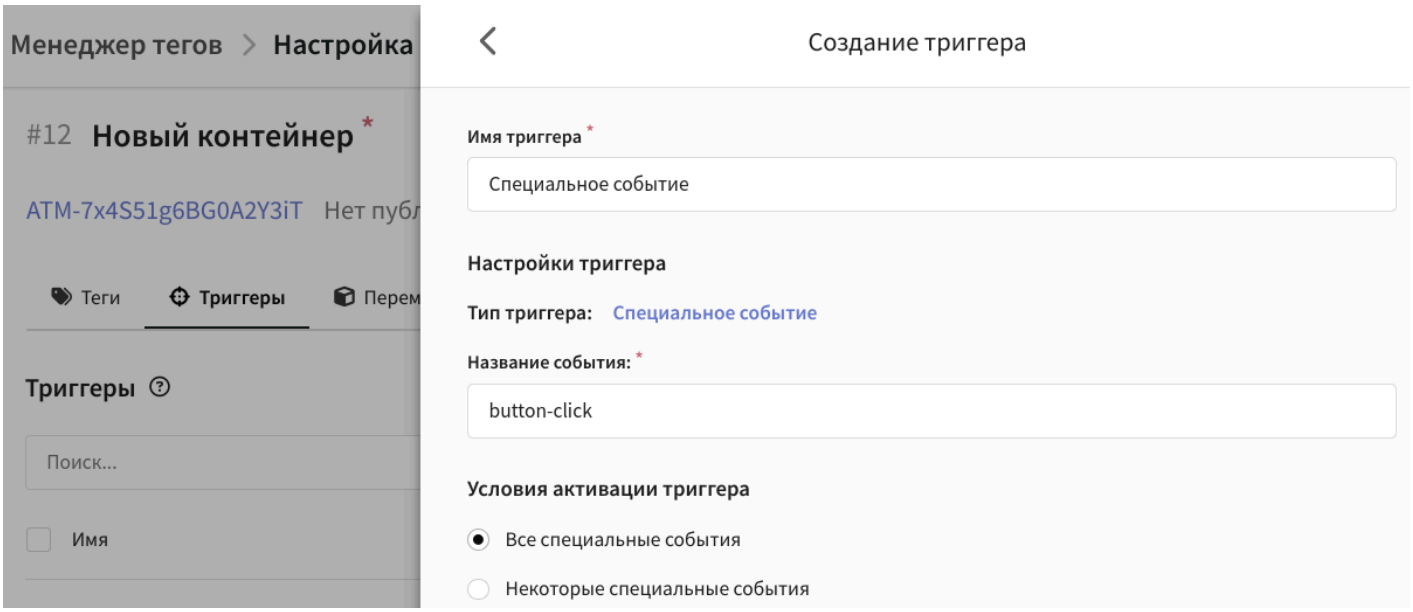
Триггер срабатывает, когда на сайте возникает ошибка JavaScript, которая не была перехвачена обработчиком событий window.onError.



Специальное событие

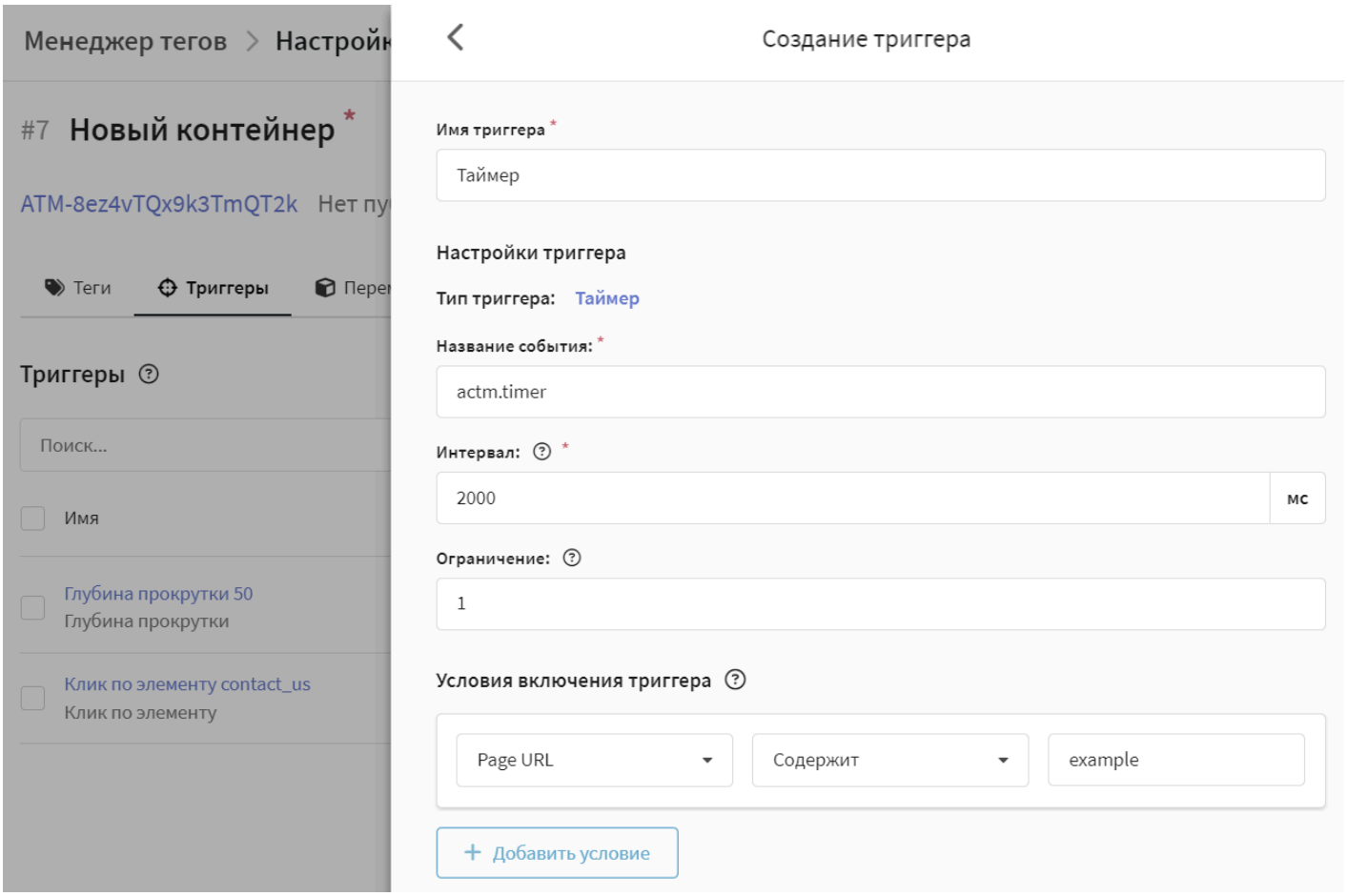
Триггер позволяет отслеживать взаимодействия с сайтом или приложением, для которых не подходят стандартные методы. Например, если вы хотите отслеживать отправку формы, для которой переопределили поведение события “submit” в браузере.

Для настройки триггера необходимо указать название события, например, `button-click`.



Таймер

Триггер срабатывает через заданный интервал времени.



Настройки триггера включают:

- **Название события по умолчанию** — actm.timer. В большинстве случаев название менять не нужно. Вы можете указать другое название, чтобы создать триггер на основе пользовательского события.
- **Интервал** — время между срабатываниями события (в миллисекундах). Например, если указать 2000, то событие будет срабатывать каждые 2 секунды. Первое срабатывание также произойдет через указанное время.
- **Ограничение** — максимальное количество срабатываний. Если оставить поле пустым, триггер будет срабатывать до тех пор, пока пользователь не покинет страницу.

TIP

Если указать интервал — 2000 мс и ограничение — 1, то первое и последнее срабатывание произойдет через 2 секунды.

Типы переменных

Описание

Переменная — это объект, которому дано имя и который может принимать различные значения во время выполнения кода на сайте (пара “ключ: значение”). Имя переменной постоянно, а значение меняется в зависимости от того, какие действия происходят на сайте. В переменной могут храниться данные о содержимом какой-либо страницы. Например, переменная “Page URL” возвращает URL текущей страницы.

Переменные используются как в триггерах, так и в тегах:

- В **триггерах** переменные применяются для настройки фильтров, которые определяют условия запуска определенных тегов. В этом случае переменные служат для таргетинга — задавая условия для активации тега, они ограничивают аудиторию, которой необходимо показать определенный контент на сайте, например, попап.
- В **тегах** переменные используются для получения динамических значений.

Для управления переменными контейнера перейдите на вкладку “Переменные”. В Менеджере тегов доступны встроенные и пользовательские переменные. Для большинства конфигураций тегов и триггеров достаточно встроенных переменных. Однако вы также можете создавать собственные пользовательские переменные для решения ваших задач.

CAUTION

Обратите внимание, при настройке триггера не рекомендуется использовать одну переменную несколько раз.

Использование переменных в коде тегов

Встроенные и пользовательские переменные можно использовать в теге “Пользовательский HTML”:

HTML

```
<div class="banner">
  <h3>{{Название акции}}</h3>
  <p>Специальное предложение для {{City}}</p>
</div>
```

JS

```
<script>
console.log('Текущий пользователь:', {{User ID}});
analytics.track('page_view', { url: {{Page URL}} });
</script>
```

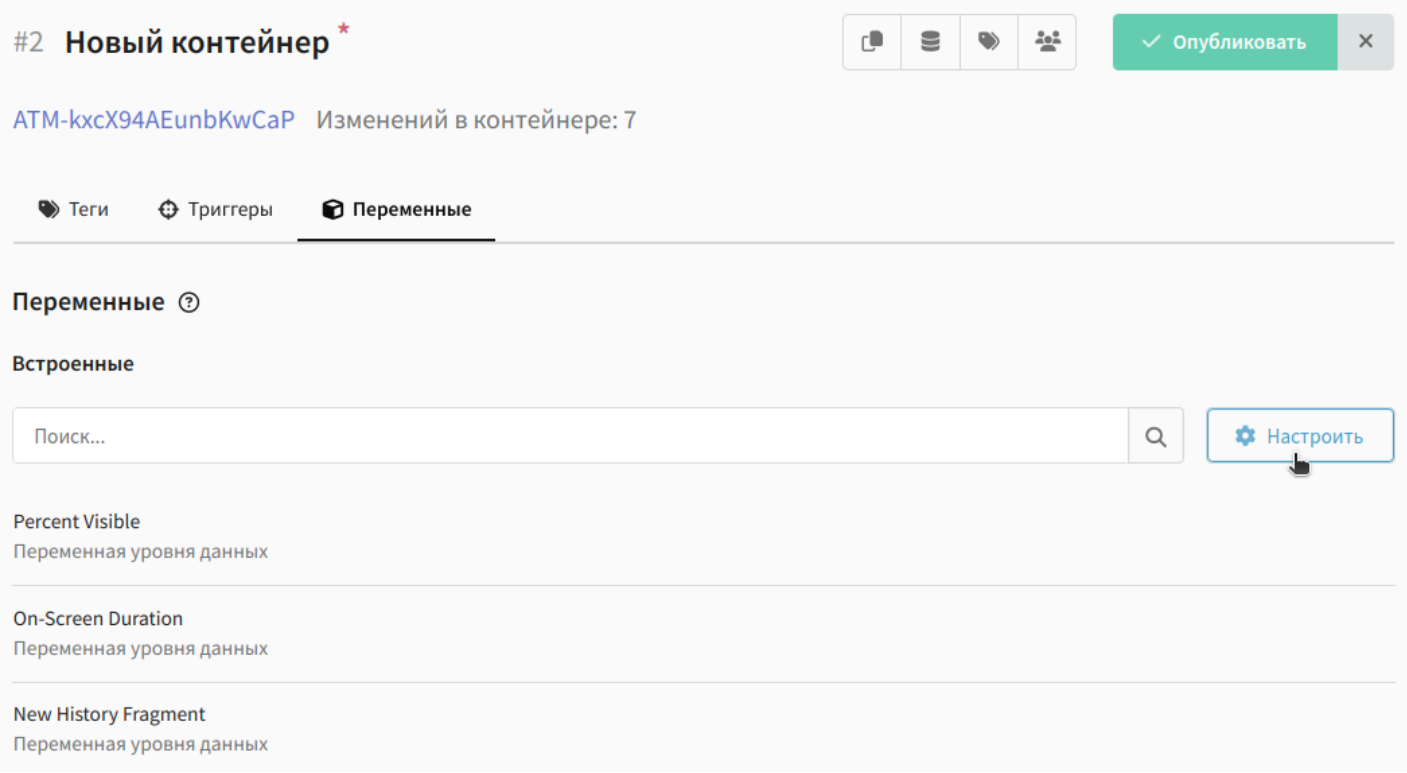
При публикации контейнера система проверяет, что все используемые переменные существуют. Если найдены неизвестные переменные (например, переменная была удалена, переименована или еще не создана), публикация будет отменена с сообщением об ошибке.

CAUTION

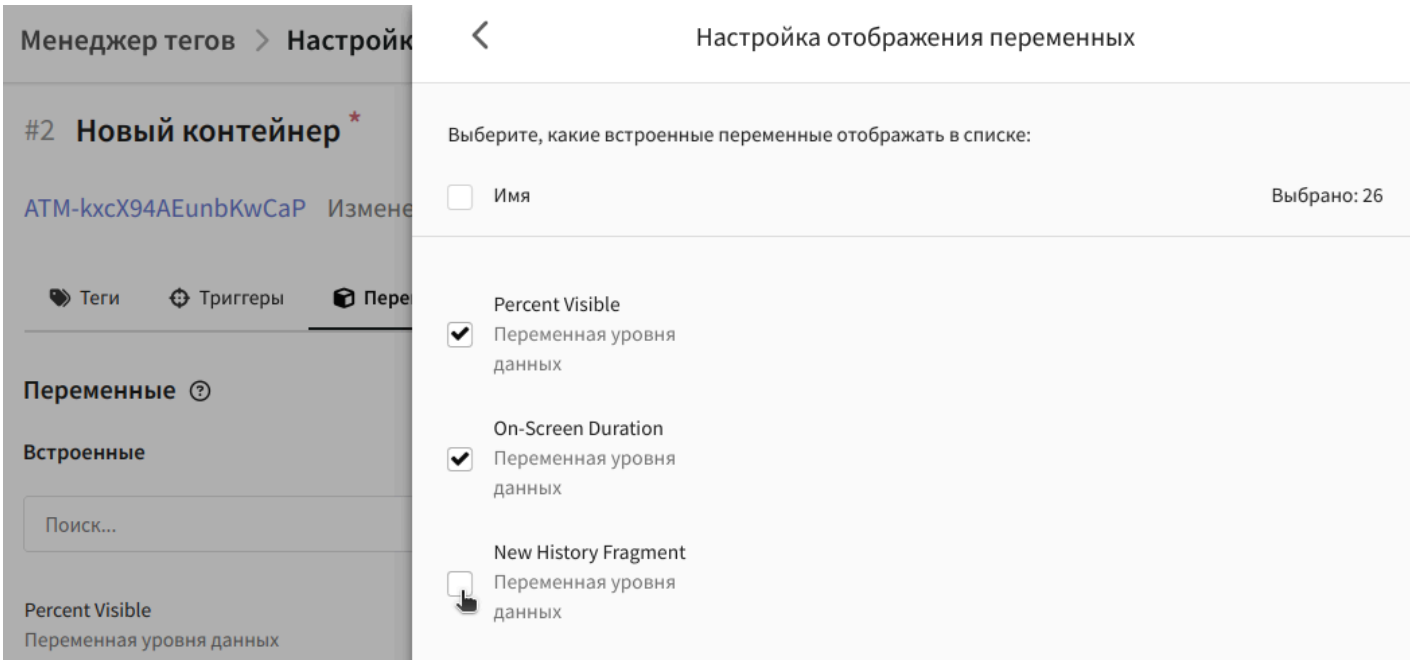
Если вы измените имя переменной, необходимо обновить его во всех тегах, где она используется.

Встроенные переменные

Встроенные переменные — это готовые переменные, которые автоматически добавляются в новый контейнер. Такие переменные нельзя изменить.



По умолчанию все встроенные переменные в контейнере активны. Если вы не хотите отображать какую-то переменную в списке, нажмите кнопку **“Настроить”** и в открывшемся окне снимите галочку напротив ее названия. Затем нажмите кнопку **“Выбрать”** внизу страницы.



Типы встроенных переменных

Страницы

- **Page URL.** Переменная возвращает полный URL-адрес текущей страницы, но без хеша. Например, если пользователь загрузил страницу `https://example.com/test?param=value#hash`, то переменная вернёт значение `https://example.com/test?param=value`.
- **Page Hostname.** Переменная возвращает доменное имя в URL текущей страницы. Например, если пользователь загрузил страницу `https://example.com/test?param=value`, то переменная вернёт значение `example.com`.
- **Page Path.** Переменная возвращает путь к странице в текущем URL без учёта GET-параметров. Например, если пользователь загрузил страницу `https://example.com/test?param=value`, то переменная вернёт значение `/test`.
- **Referrer.** Переменная возвращает полный URL страницы, с которой пользователь перешёл на текущую страницу.

Утилиты

- **Container ID.** Переменная возвращает номер контейнера в менеджере тегов. Например, ATM-4ynTVvH4Ga0O.

Ошибки

- **Error Message.** Переменная возвращает сообщение об ошибке, отправленное с помощью триггера “Ошибка JavaScript”.
- **Error URL.** Переменная возвращает URL скрипта, в котором обнаружена ошибка.
- **Error Line.** Переменная возвращает номер строки файла, в которой произошла ошибка.
- **Debug Mode.** Переменная возвращает значение “true”, если активирован режим отладки в Менеджере тегов.

Клики

- **Click Element.** Переменная возвращает HTML-элемент, на который кликнул пользователь.
- **Click Classes.** Переменная возвращает значение атрибута `class` для HTML-элемента, по которому кликнул пользователь.
- **Click ID.** Переменная возвращает значение атрибута `id` для HTML-элемента, по которому кликнул пользователь.
- **Click Target.** Переменная возвращает значение атрибута `target` для HTML-элемента, по которому кликнул пользователь.
- **Click URL.** Переменная возвращает значение атрибута `href` для HTML-элемента, по которому кликнул пользователь.
- **Click Text.** Переменная возвращает значение атрибута `textContent` для HTML-элемента, по которому кликнул пользователь.

Формы

- **Form Element.** Переменная возвращает JavaScript-объект для отправленной формы. Например, атрибуты class, ID, данные о родительских и дочерних элементах.
- **Form Classes.** Значение атрибута `class` для отправленной формы.
- **Form ID.** Переменная возвращает значение атрибута `id` для отправленной формы.
- **Form Target.** Переменная возвращает значение атрибута `target` для отправленной формы.
- **Form URL.** Переменная возвращает значение атрибута `action` для отправленной формы.
- **Form Text.** Переменная возвращает текст, содержащийся в отправленной форме и ее потомках.

История

- **New History Fragment.** Переменная возвращает новое значение хеша (#) URL-сайта после совершения пользовательского события “Изменение в истории”.
- **Old History Fragment.** Переменная возвращает предыдущее значение хеша URL-сайта до совершения пользовательского события.
- **History Source.** Переменная возвращает строку, содержащую событие, которое привело к изменению объекта истории.

Прокрутка

- **Scroll Depth Threshold.** Переменная возвращает числовое значение порога прокрутки, при пересечении которого был запущен триггер “Глубина прокрутки” (в процентах или в пикселях).
- **Scroll Depth Units.** Переменная возвращает единицу измерения (“пиксели” или “проценты”) для порогового значения, запустившего триггер “Глубина прокрутки”.
- **Scroll Direction.** Переменная возвращает параметр, указывающий на направление прокрутки (“вертикальное” или “горизонтальное”).

Видимость элемента

- **Percent Visible.** Переменная возвращает числовое значение от 0 до 100, которое показывает, какой процент выбранного элемента был виден при срабатывании триггера “Видимость элемента”.
- **On-Screen Duration.** Переменная возвращает числовое значение, которое показывает, как долго выбранный элемент был виден при срабатывании триггера “Видимость элемента”.

Гео-переменные

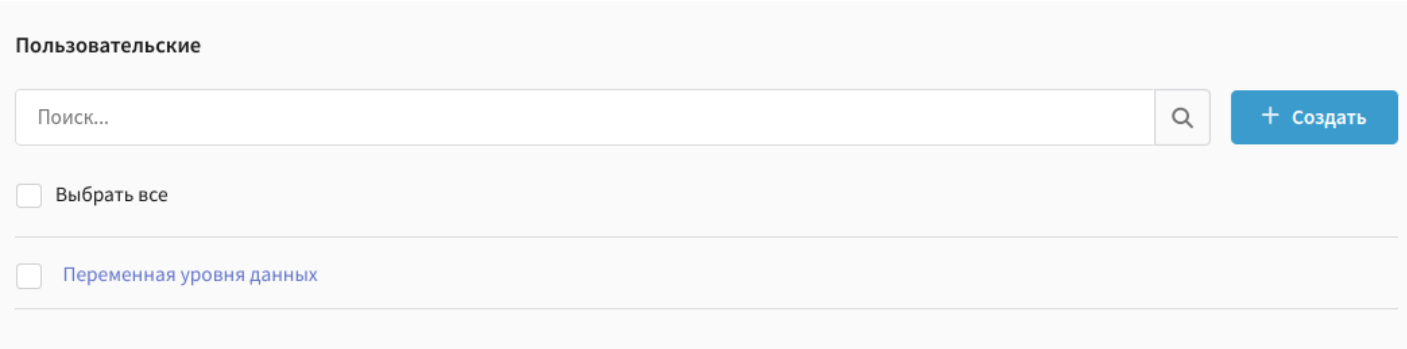
- **Country.** Переменная возвращает страну профиля.
- **City.** Переменная возвращает город профиля.

Агент

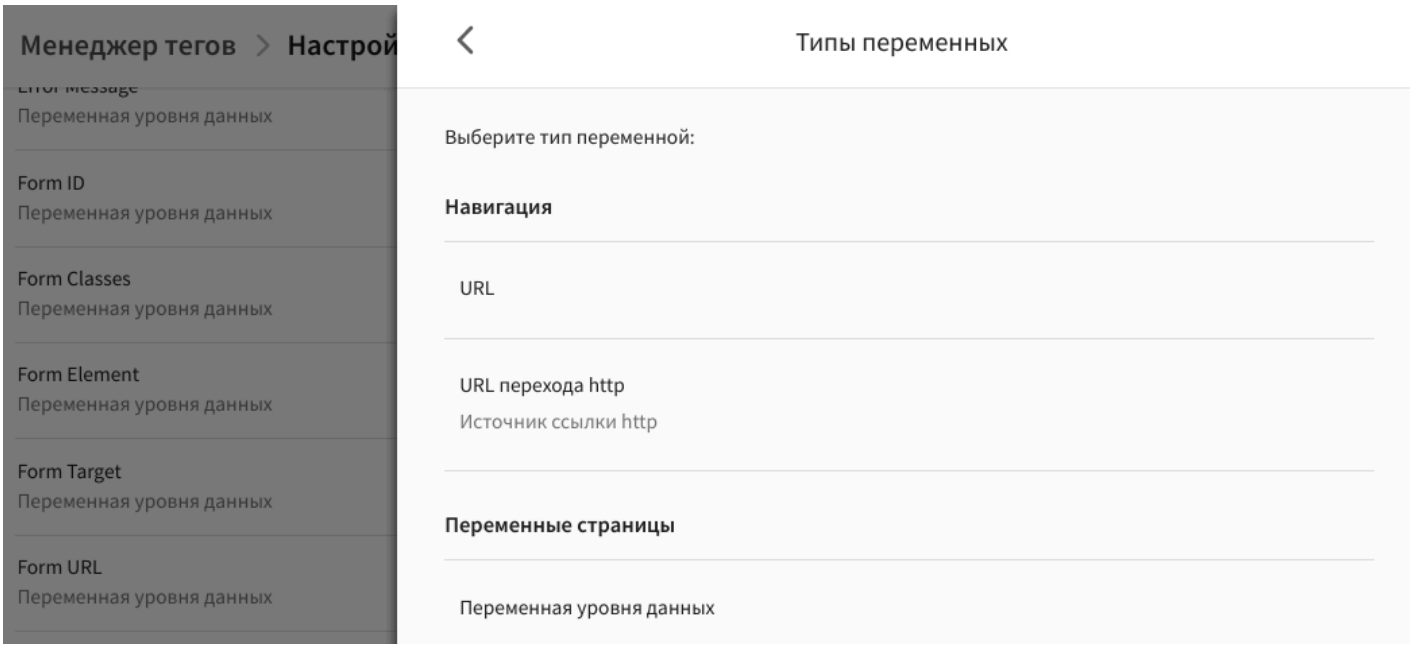
- **Device.** Переменная возвращает устройство профиля.
- **Browser.** Переменная возвращает браузер профиля.

Пользовательские переменные

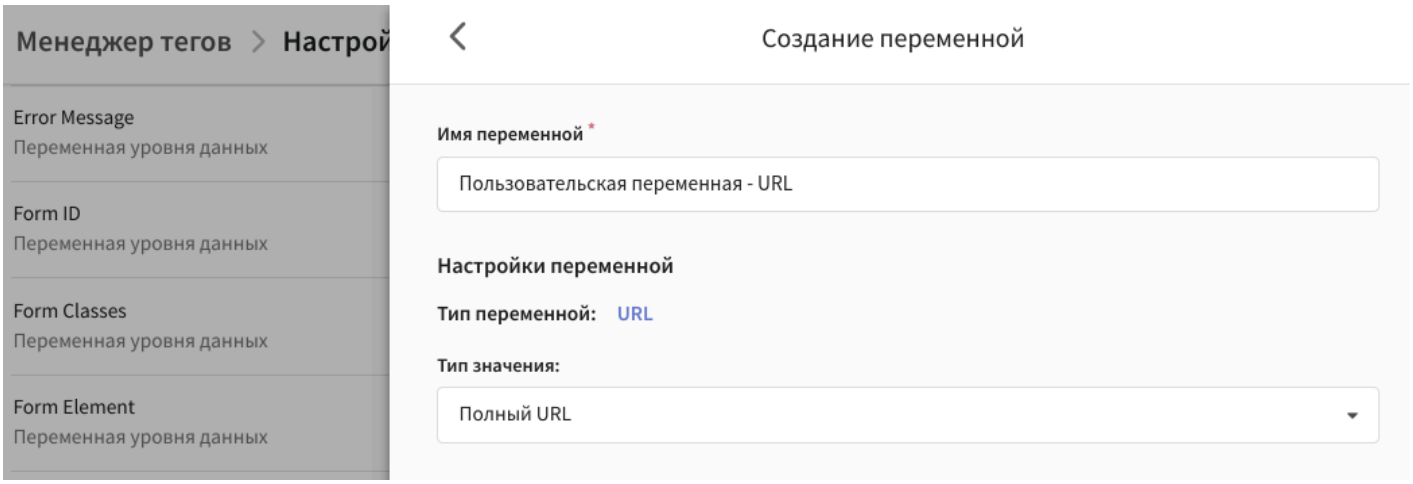
Для создания новой переменной в окне **“Настройка контейнера”** перейдите на вкладку **“Переменные”**. Найдите блок **“Пользовательские”** и нажмите кнопку **+ Создать**:



В появившемся окне укажите произвольное имя переменной и выберите её **тип**:

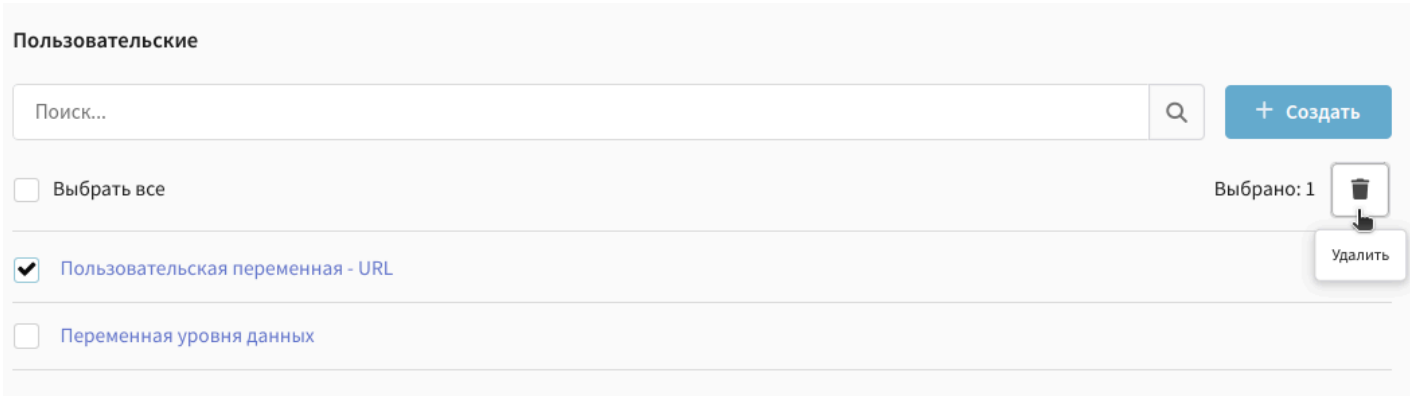


Задайте **настройки переменной** и **сохраните** с помощью кнопки внизу страницы.



После сохранения переменная появится в списке. Чтобы изменить настройки переменной, нажмите на её название. Вы также можете удалить ненужные переменные: выделите их и нажмите

:

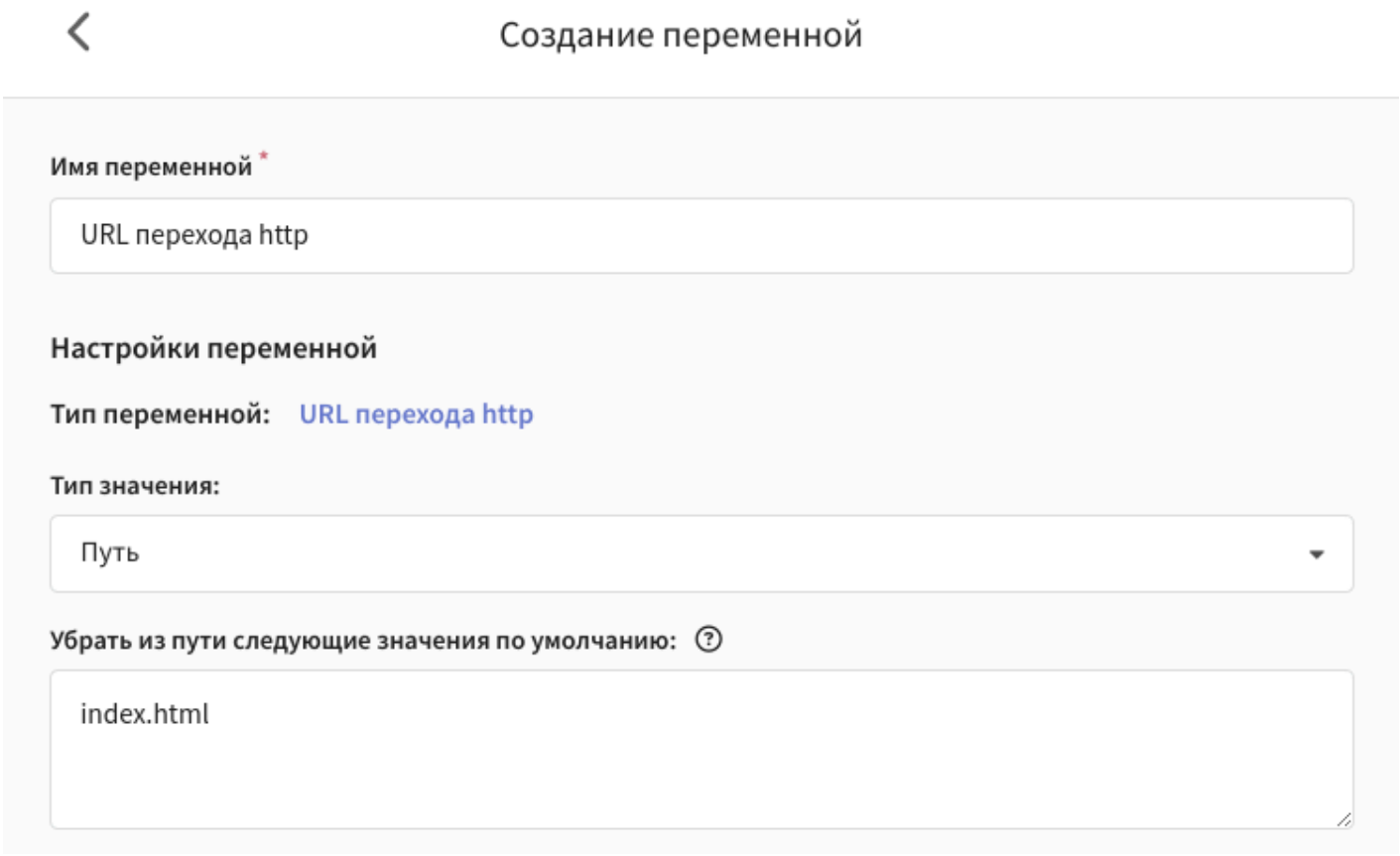


Типы пользовательских переменных

Навигация

Данный тип переменных позволяет получить доступ к компонентам URL:

- **URL** — Аналог встроенных переменных Page URL, Page Hostname, Page Path. Входное значение для переменной этого типа – URL страницы, на которой находится пользователь. Значение извлекается из `document.location`
- **URL перехода https** — Аналог встроенной переменной Referrer. Входное значение для переменной этого типа – URL-адрес предыдущей страницы. Значение извлекается из `document.referrer`.



Тип значения (на примере URL: `http://example.com:80/path/index.html?param1=value1&param2=value2#hash`):

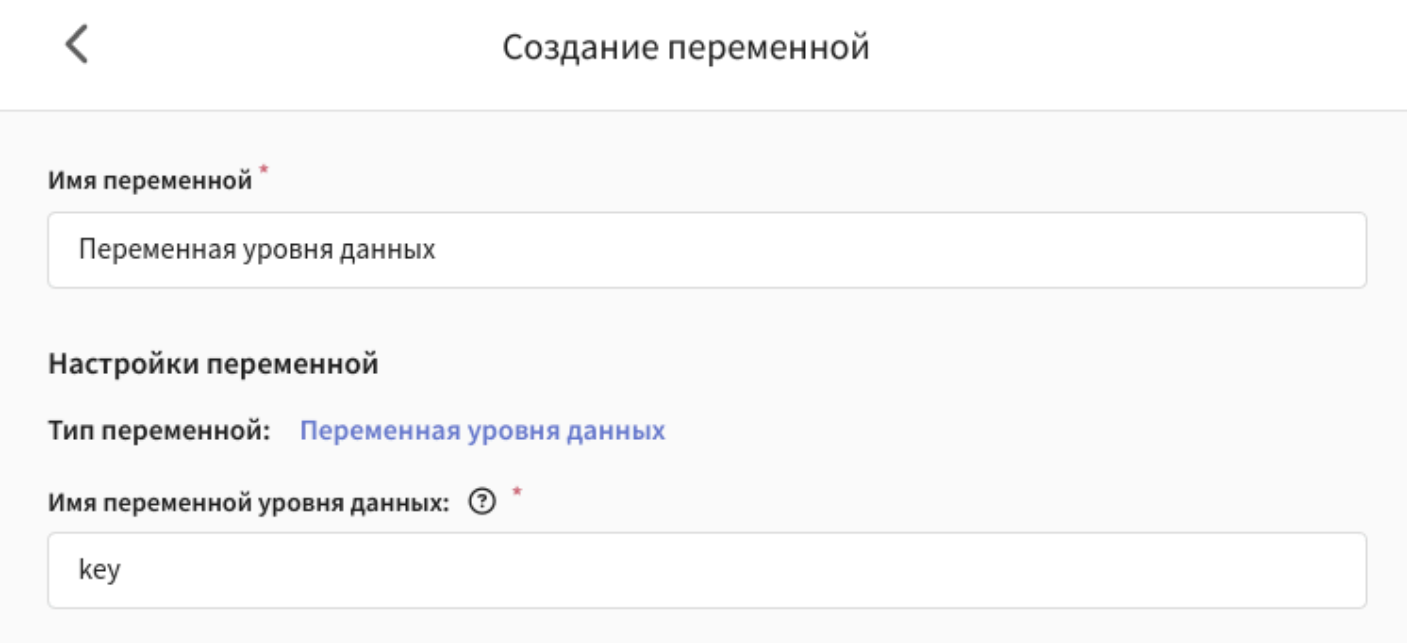
Тип значения	Пример	Описание
Полный URL	http://example.com:80/path/index.html?param1=value1&param2=value2	Переменная возвращает полный URL-адрес без фрагмента хеширования.
Имя хоста	example.com	Переменная возвращает имя хоста URL-адреса без номера порта. Если активировать поле "убрать www. из имени хоста", переменная вернёт одинаковое значение для www.example.com и example.com. </td>
Запрос	param1=value1&param2=value2	Переменная возвращает параметры запроса (без "?"). Если указать ключ запроса, переменная будет возвращать только значение этого ключа. например, для ключа <code>param1</code> переменная вернёт значение <code> value1 </code>. </td>

Путь	/path/index.html	Переменная возвращает путь в URL-адресе. В поле **"Убрать из пути следующие значения по умолчанию"** можно указать название индексного файла — `index.html`. В таком случае страницы `http://example.com/path/index.html` и `http://example.com/path/` будут рассматриваться как одна страница. Также можно указать несколько значений по умолчанию — каждое значение на отдельной строке. в таком случае если последний сегмент пути соответствует значению по умолчанию и не является каталогом, он будет удален.
Протокол	http:	Переменная возвращает протокол URL.
Порт	80	Переменная возвращает номер порта.
Расширение файла	html	Переменная возвращает расширение файла.
Фрагмент	hash	Переменная возвращает фрагмент хеширования (без "#").

Переменные страницы

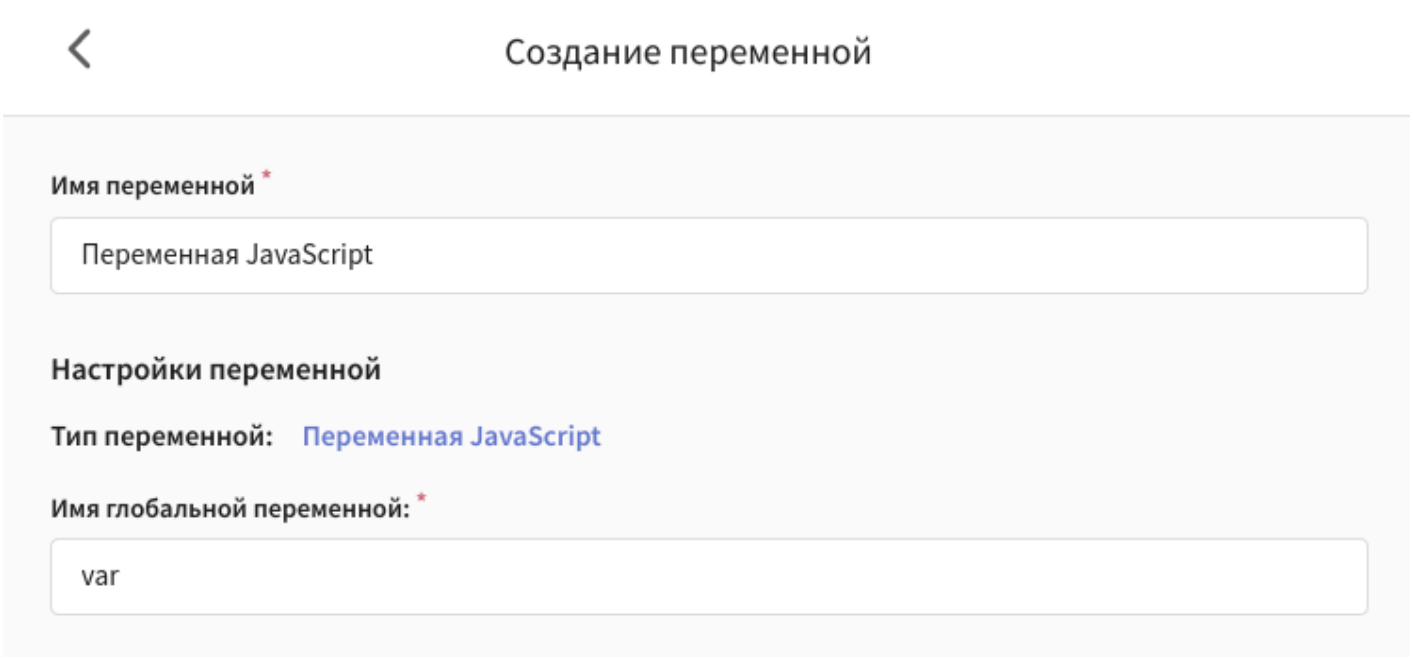
Переменная уровня данных

Переменная принимает значение в момент передачи информации на уровень данных с помощью метода `acDataLayer.push({'var':'value'})`. Информация передаётся в формате `key:value`, где `key` — это имя переменной, которое задаётся в поле “Имя переменной уровня данных”, а `value` — это передаваемое значение.



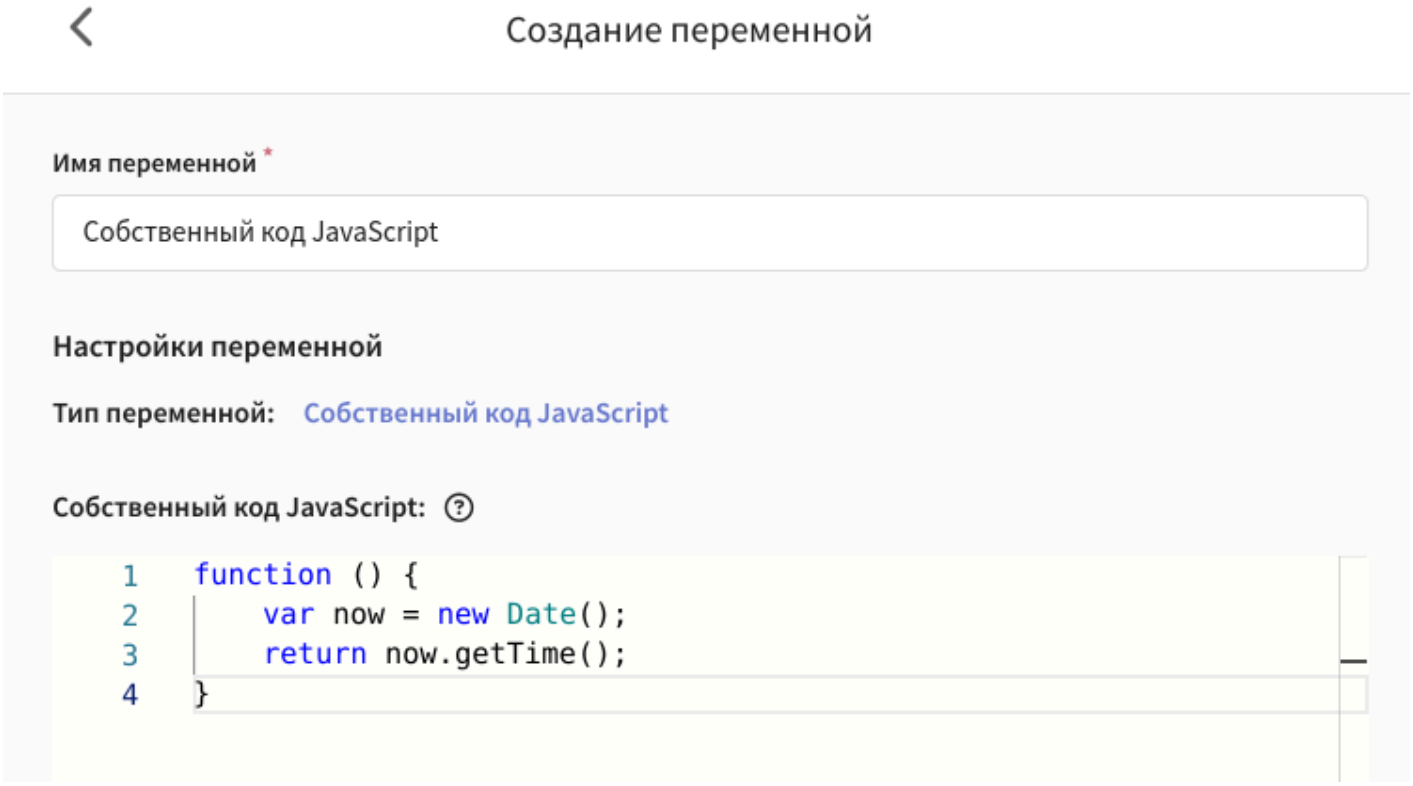
Переменная JavaScript

Переменная принимает значение глобальной переменной JavaScript, имя которой задаётся в поле “Имя глобальной переменной”. Если такой переменной не существует, вернётся значение undefined.



Собственный код JavaScript

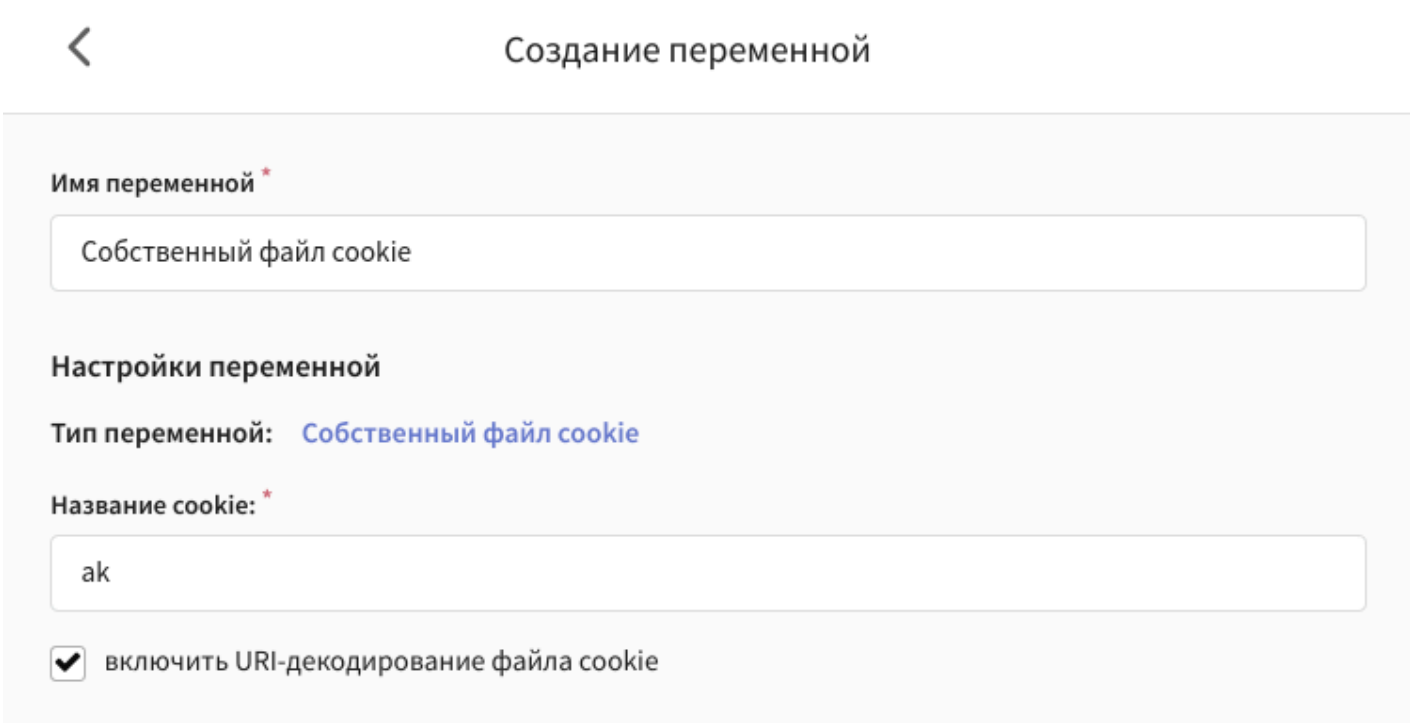
Переменная принимает значение, которое возвращает анонимная функция JavaScript.



Собственный файл cookie

Переменная принимает значение cookie пользователя, доступное для текущего сайта. Значение извлекается из `document.cookie`. Необходимо указать имя cookie, значение которой вы хотите передавать в переменную. Например, если на сайте установлена cookie `"ak": "cookieValue"`, то в поле “Название cookie” можно указать `ak`.

Значение cookie кодируется — активируйте поле **“Включить URI-декодирование файла cookie”** для декодирования.



Элементы страницы

Видимость элемента

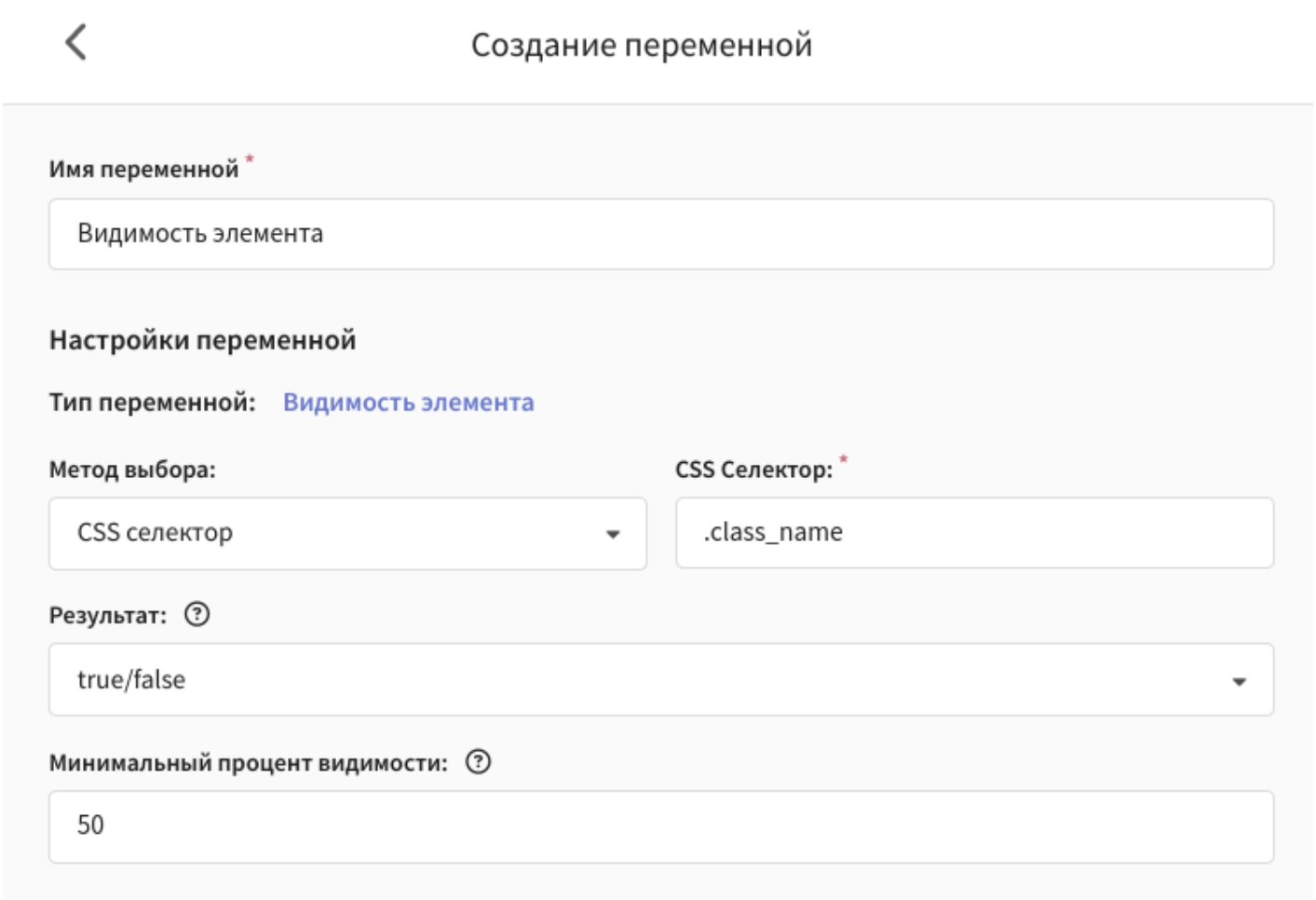
Переменная позволяет определить, какой элемент был виден в браузере пользователя. Сначала укажите **метод выбора**:

- **Идентификатор** — отслеживается видимость одного элемента с определенным значением атрибута `id`. Например, если указать `test`, то будет выбран элемент с `id="test"`.
- **CSS селектор** — отслеживается видимость нескольких элементов, которые соответствуют заданному шаблону CSS селектора. Например, .`class_name` выберет элементы на странице с `class="class_name"`.

Результат определяет возвращаемое значение:

- **True/false** — переменная примет значение true, если будет достигнут заданный минимальный процент видимости элемента. В противном случае — false.
- **Процент видимости** — переменная примет значение, равное проценту видимости.

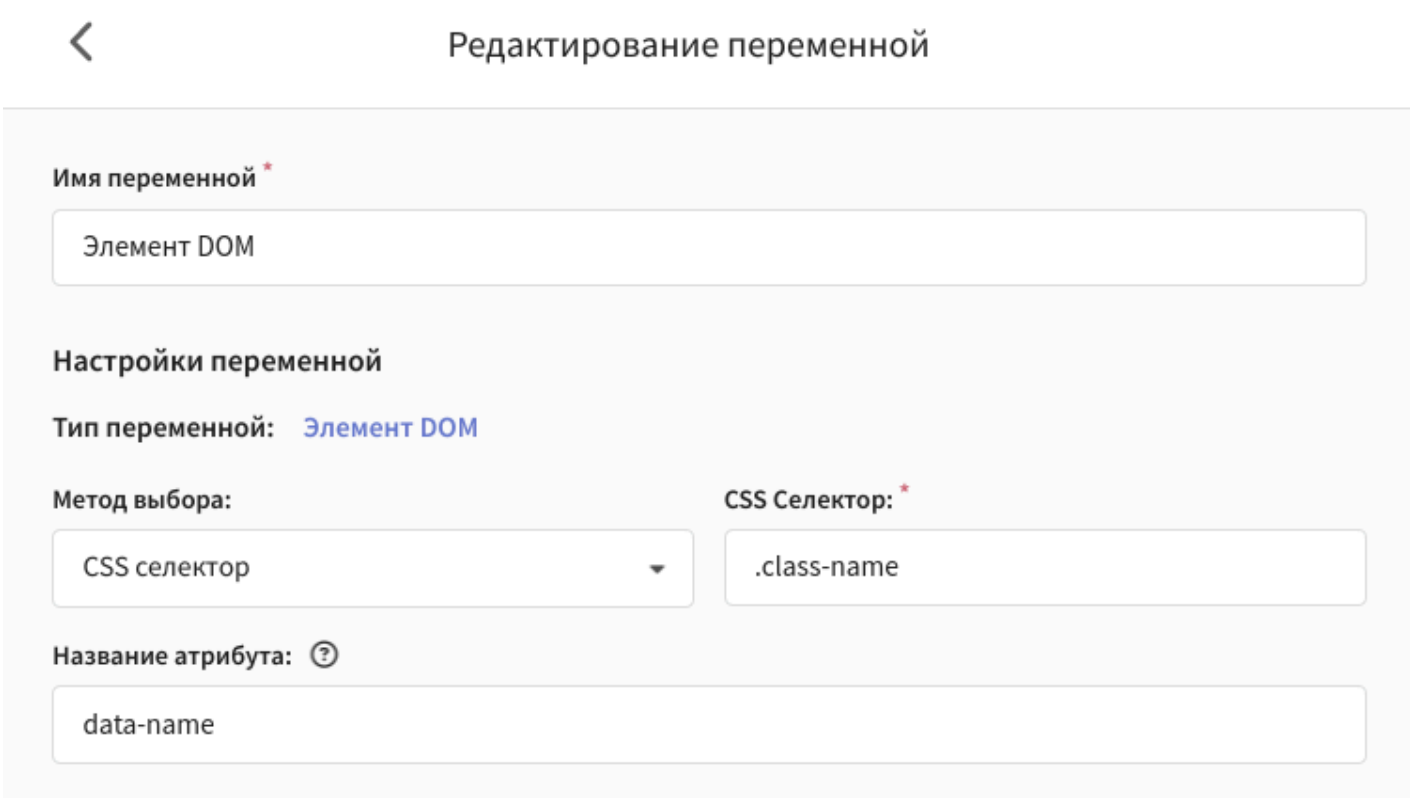
В примере ниже переменная вернет значение `true`, когда половина элемента с классом `class_name` отобразится на экране пользователя:



Элемент DOM

Переменная используется для получения текстового содержимого любого DOM-элемента или для извлечения значения любого атрибута DOM-элемента. Определите, какие элементы должна отслеживать переменная. Для этого укажите метод выбора и задайте идентификатор элемента или шаблон CSS селектора.

Если задать **название атрибута**, переменной будет присвоено значение атрибута (например, `data-name="value"`). Если название атрибута не указано, переменная будет возвращать текст DOM-элемента.



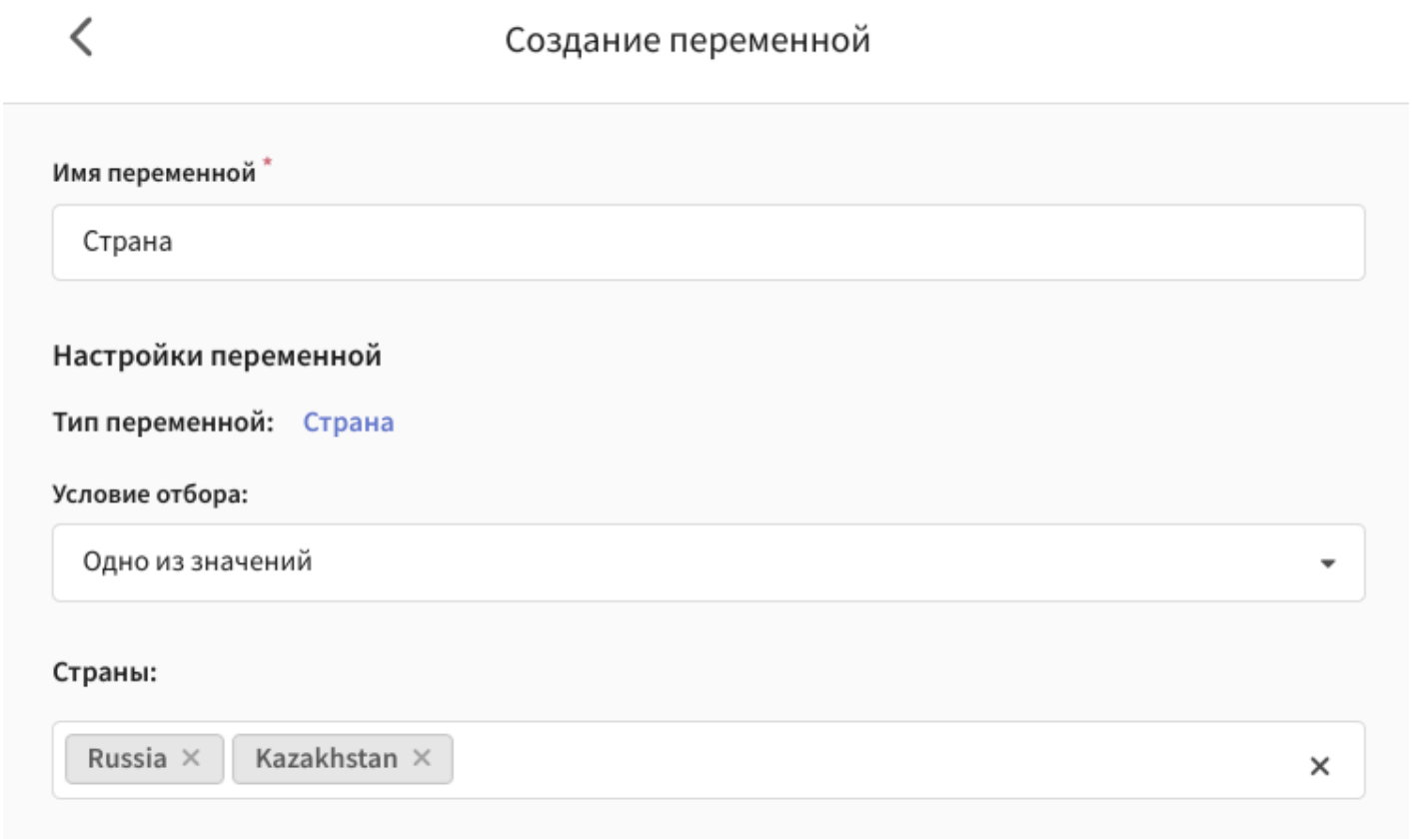
Прочее

Страна

Переменная принимает значение `true`, если страна пользователя соответствует заданному условию отбора, и `false` — если не соответствует.

Условие отбора:

- **Одно из значений** — переменная вернёт true, если страна пользователя соответствует одной из выбранных.
- **Ни одно из значений** — переменная вернёт true, если страна пользователя не соответствует ни одной из выбранных.

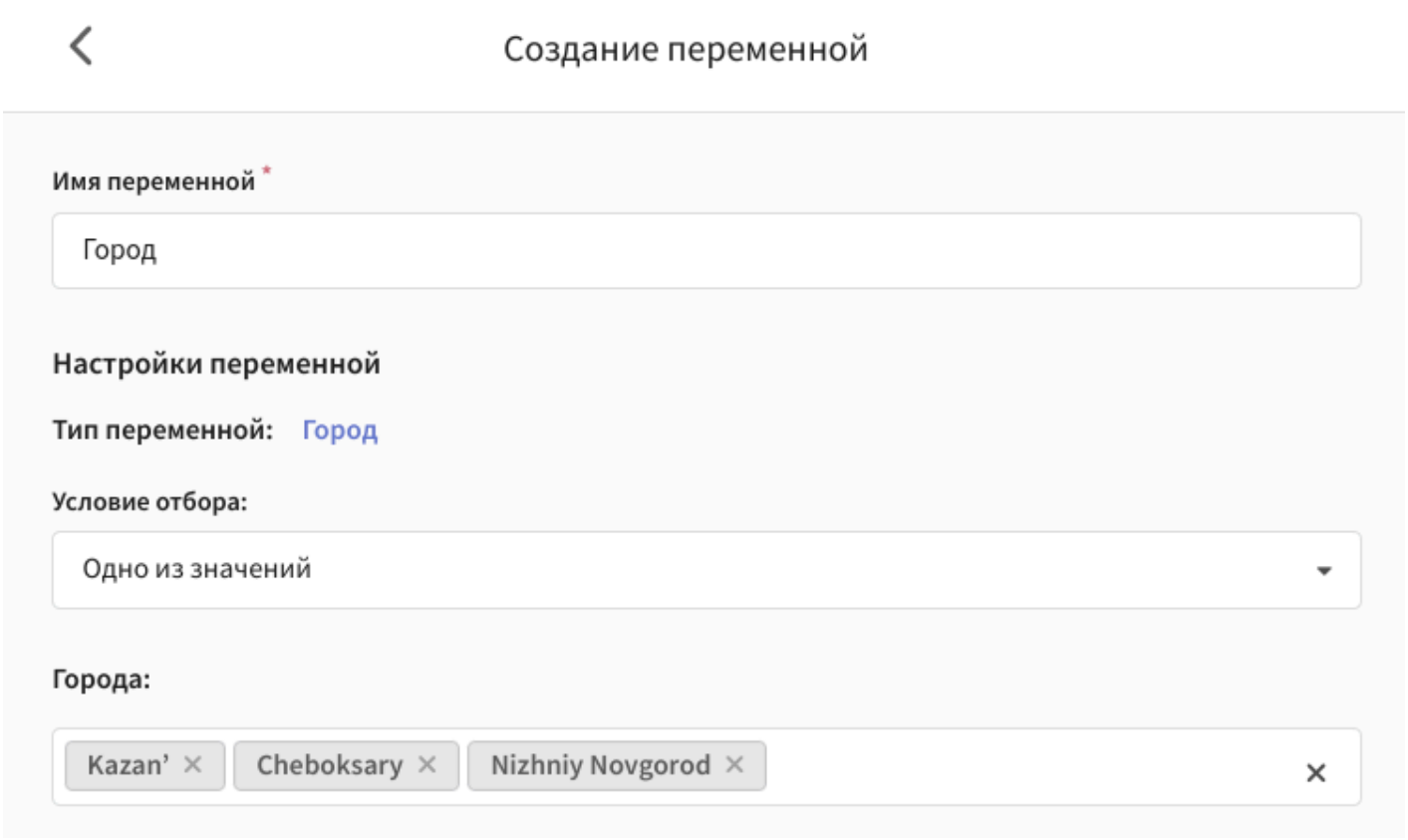


Город

Переменная принимает значение `true`, если город пользователя соответствует заданному условию отбора, и `false` — если не соответствует.

Условие отбора:

- **Одно из значений** — переменная вернёт true, если город пользователя соответствует одному из выбранных.
- **Ни одно из значений** — переменная вернёт true, если город пользователя не соответствует ни одному из выбранных.

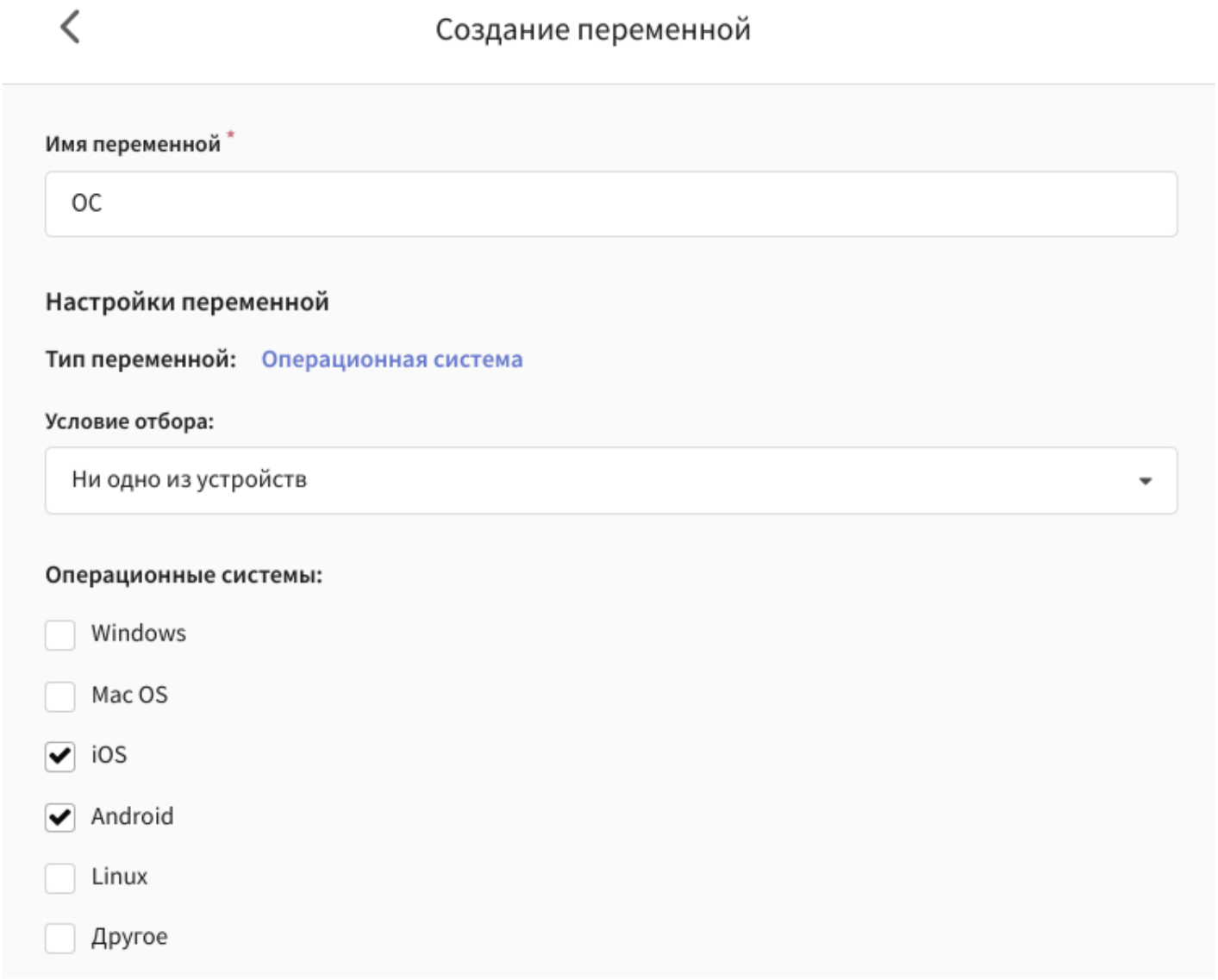


Операционная система

Переменная принимает значение `true`, если ОС пользователя соответствует заданному условию отбора, и `false` — если не соответствует.

Условие отбора:

- **Одно из устройств** — переменная вернёт true, если пользователь использует одну из выбранных ОС.
- **Ни одно из устройств** — переменная вернёт true, если пользователь не использует ни одну из выбранных ОС.

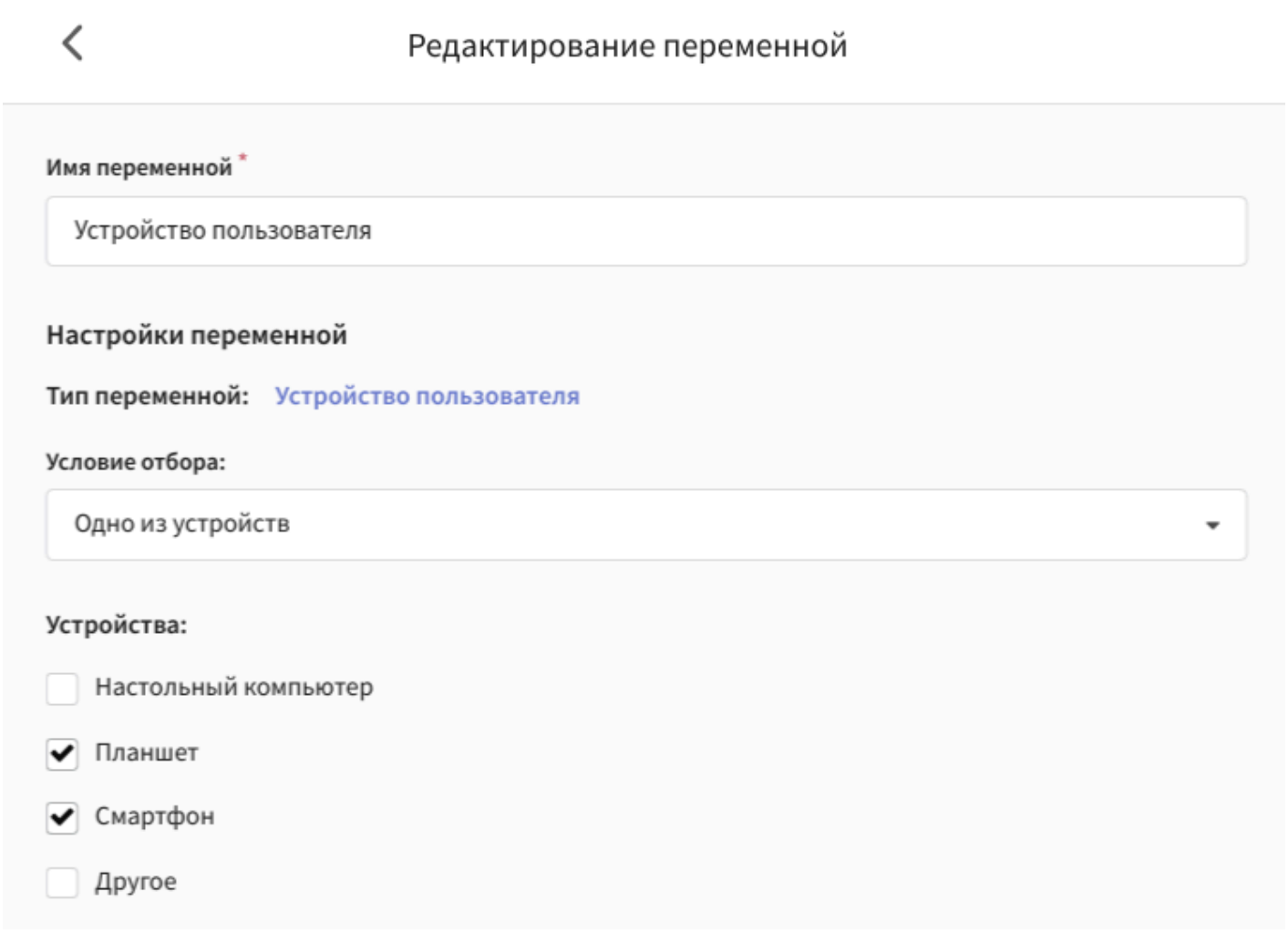


Устройство пользователя

Переменная принимает значение `true`, если устройство пользователя соответствует заданному условию отбора, и `false` — если не соответствует.

Условие отбора:

- **Одно из устройств** — переменная вернёт true, если пользователь использует одно из выбранных устройств.
- **Ни одно из устройств** — переменная вернёт true, если пользователь не использует ни одно из выбранных устройств.

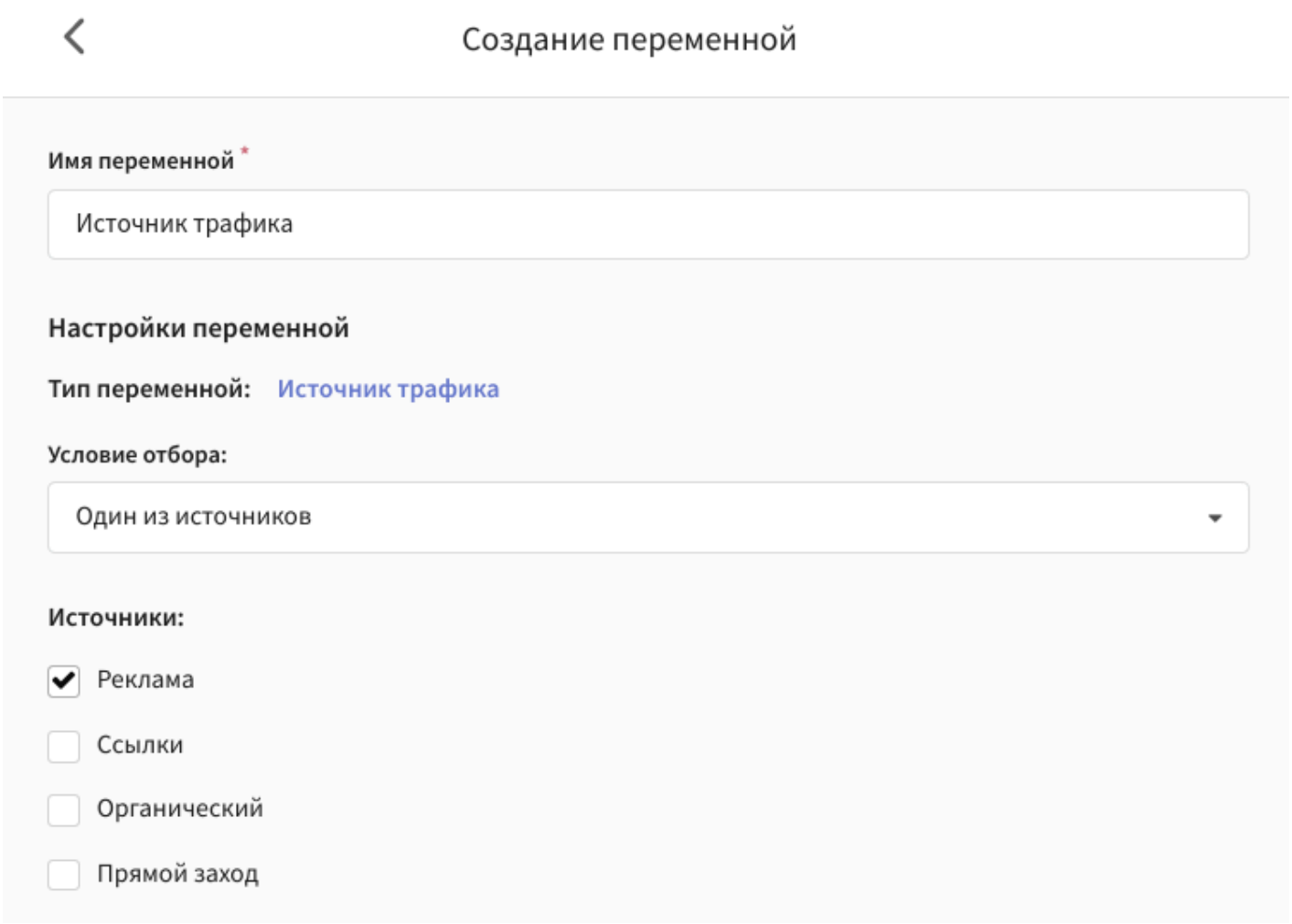


Источник трафика

Переменная принимает значение `true`, если источник перехода пользователя на сайт соответствует заданному условию отбора, и `false` — если не соответствует.

Условие отбора:

- **Одно из устройств** — переменная вернёт true, если источник трафика соответствует одному из выбранных.
- **Ни одно из устройств** — переменная вернёт true, если источник трафика не соответствует ни одному из выбранных.



UTM-метка

Переменная принимает значение выбранного UTM-параметра.

Создание переменной

Имя переменной *

UTM метка

Настройки переменной

Тип переменной: UTM метка

Параметр:

source

- content
- **source**
- campaign
- medium
- term

Связывание пикселя и Менеджера тегов

Менеджер тегов — удобный способ разместить **пиксели отслеживания** на сайте.

Когда вы используете пиксели как отдельный инструмент, вам приходится вносить изменения в исходный код сайта каждый раз, когда вы создаёте новый пиксель. При использовании Менеджера тегов этого можно избежать.

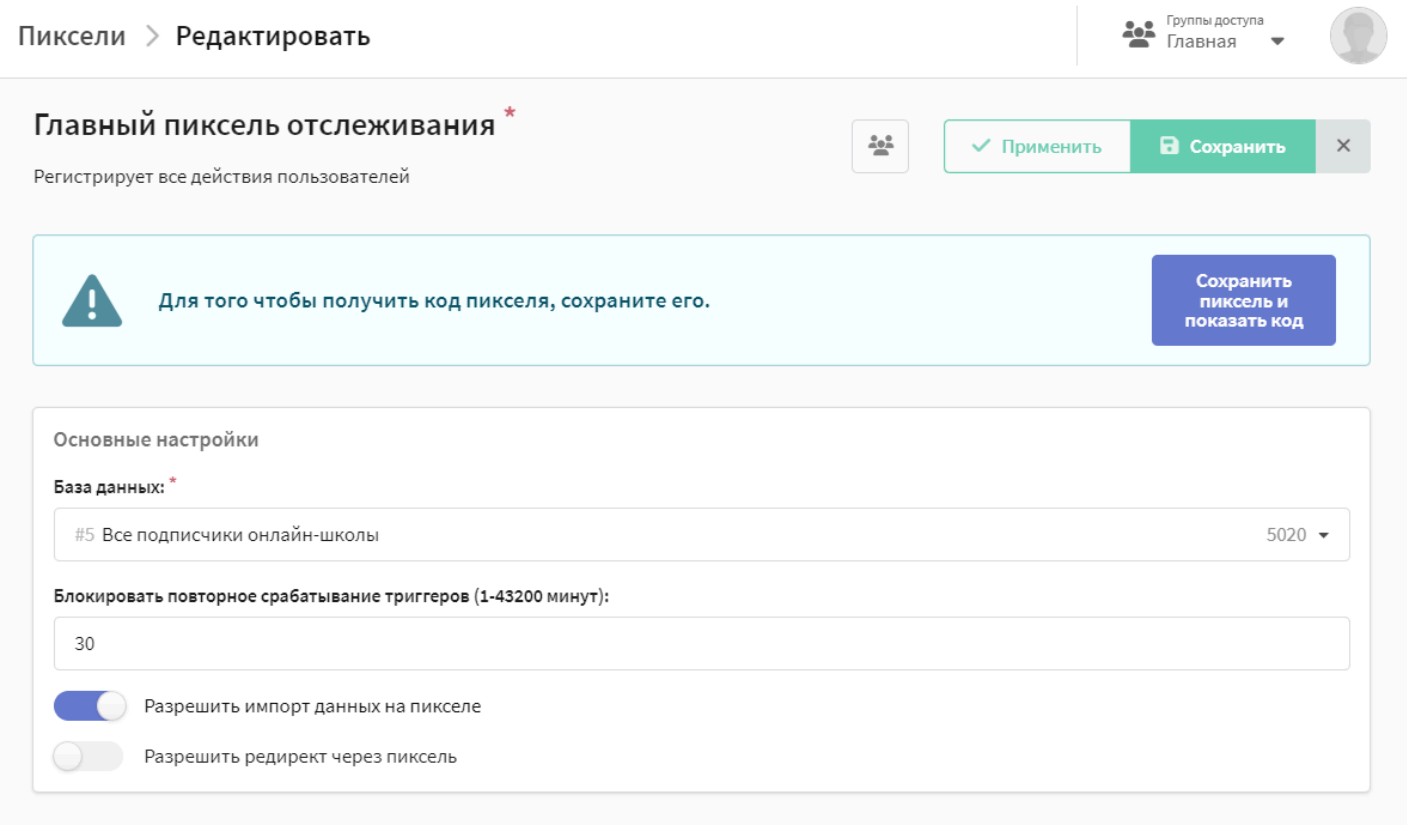
Вы создаёте контейнер, добавляете в него тег с HTML-кодом пикселя и триггеры, которые определяют условия срабатывания пикселя. Далее вы размещаете код контейнера на страницах сайта. Если в дальнейшем вы захотите внести какие-то изменения в работу пикселей (создать новый пиксель или приостановить работу существующего), достаточно будет отредактировать тег в интерфейсе платформы.

Данное руководство поможет вам связать Пиксели отслеживания и Менеджер тегов в системе Altcraft.

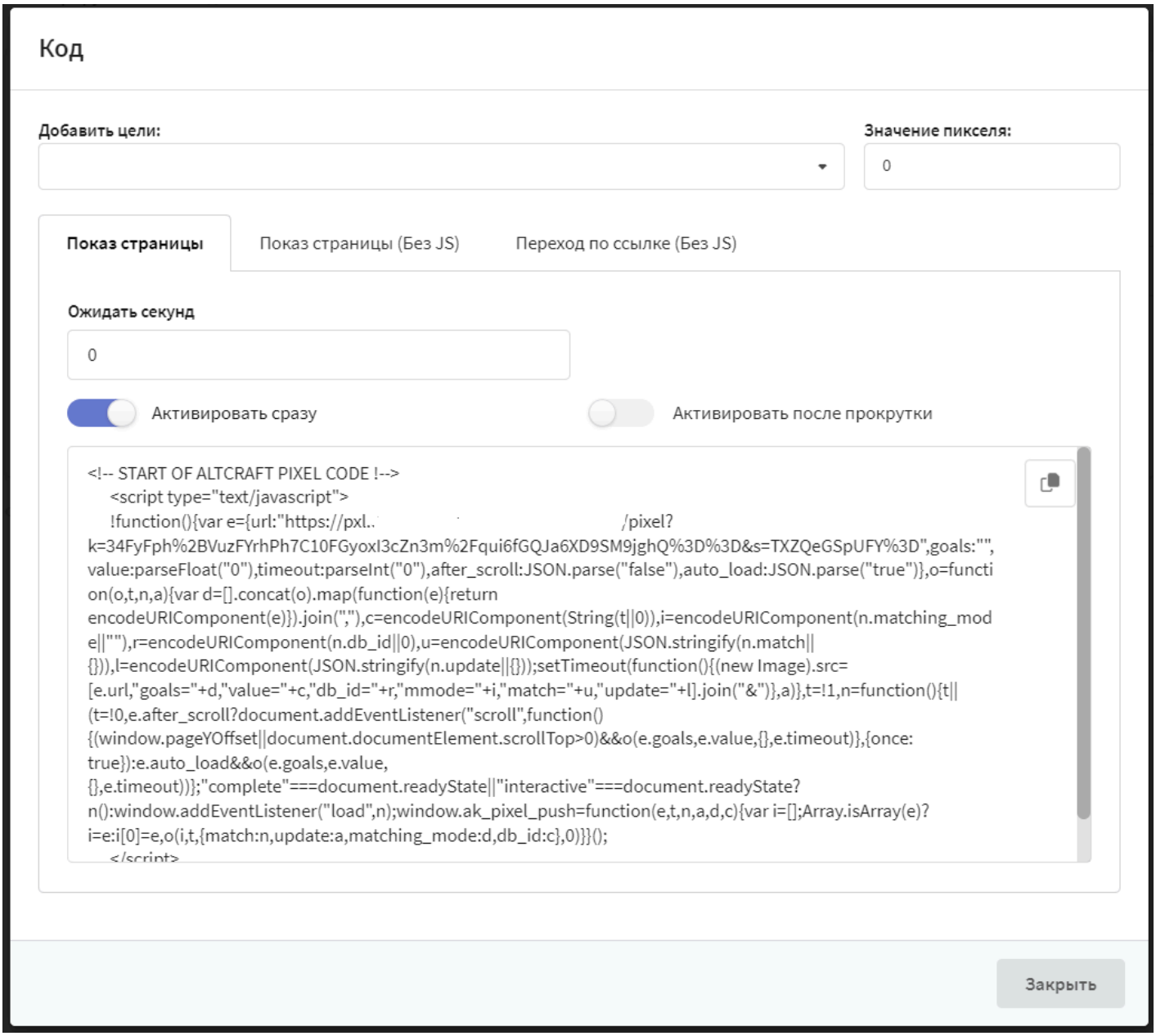
Шаг 1. Создание пикселя

1. Перейдите в раздел **“Пиксели”** и нажмите **“Создать”**.

2. Настройте и сохраните пиксель.



3. Нажмите **“Показать код пикселя”**, добавьте цели и скопируйте код.



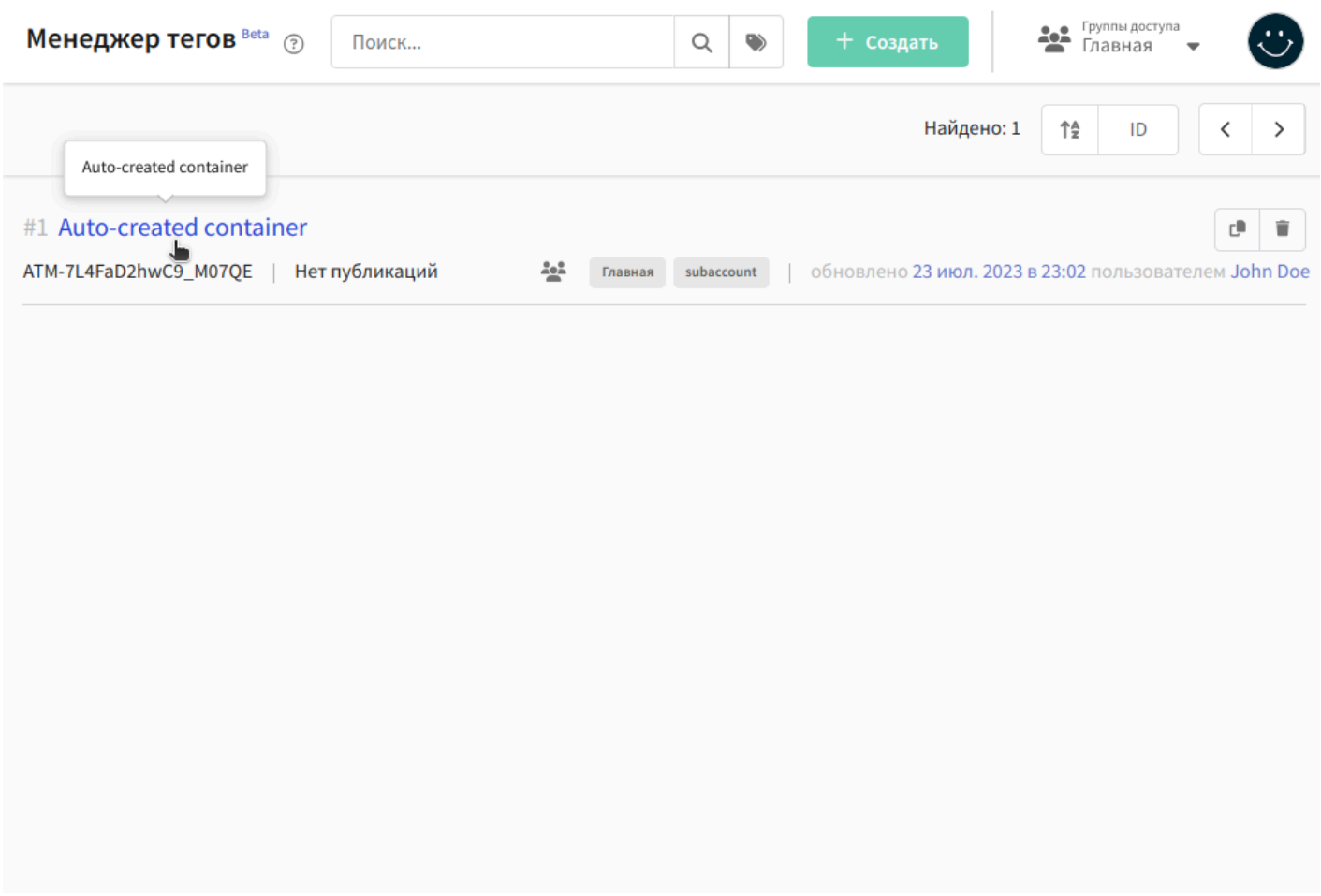
TIP

Больше информации о создании и настройке пикселя вы найдёте в [этой статье](#).

Шаг 2. Создание тега в контейнере

1. Перейдите в раздел **Веб-слой —> Менеджер тегов**.

2. Откройте контейнер:



3. На вкладке **“Теги”** нажмите **“+ Создать”**.

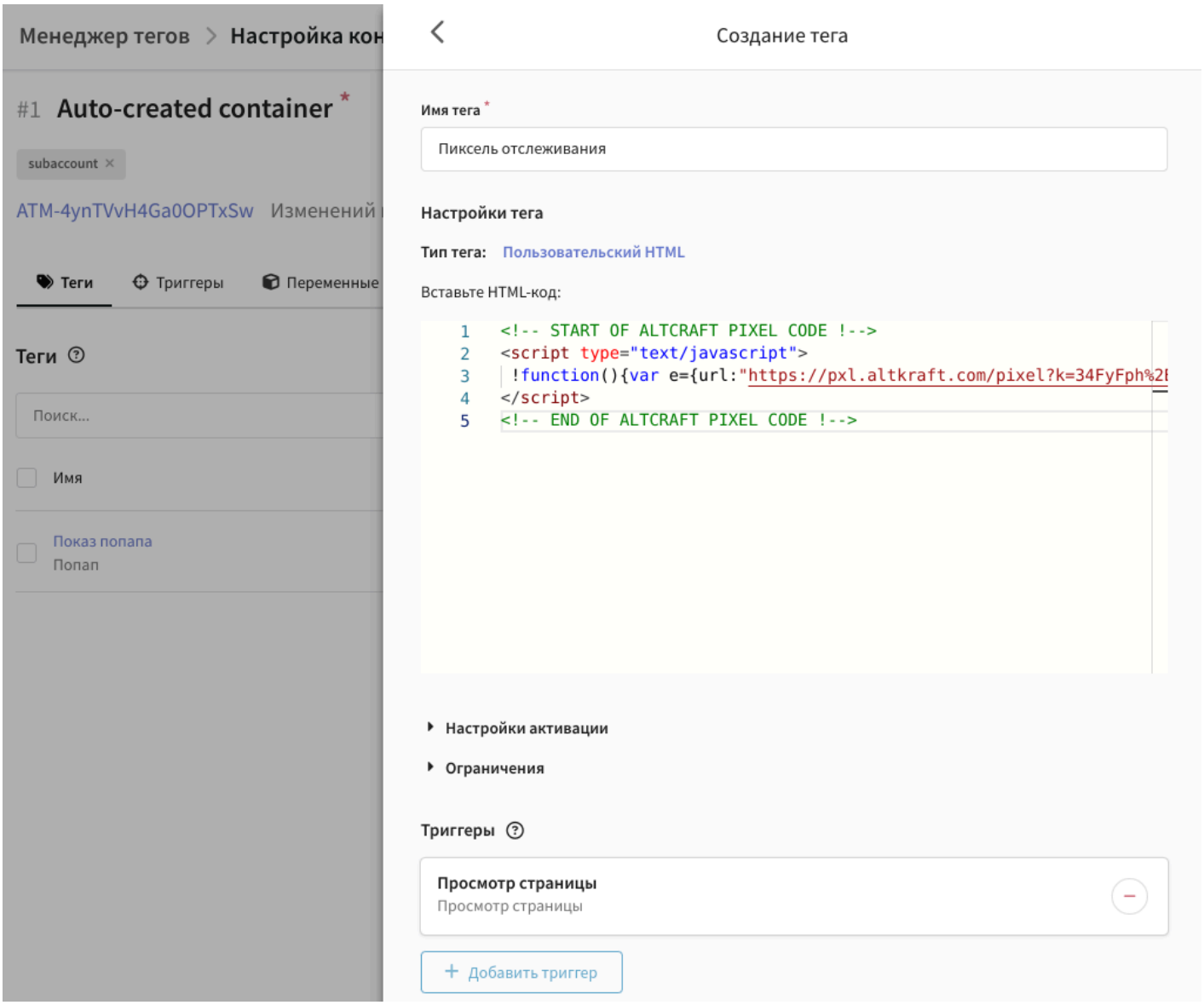
4. Введите произвольное **имя** тега.

5. В качестве типа тега выберите **“Пользовательский HTML”**.

6. Вставьте код пикселя в окно редактора.

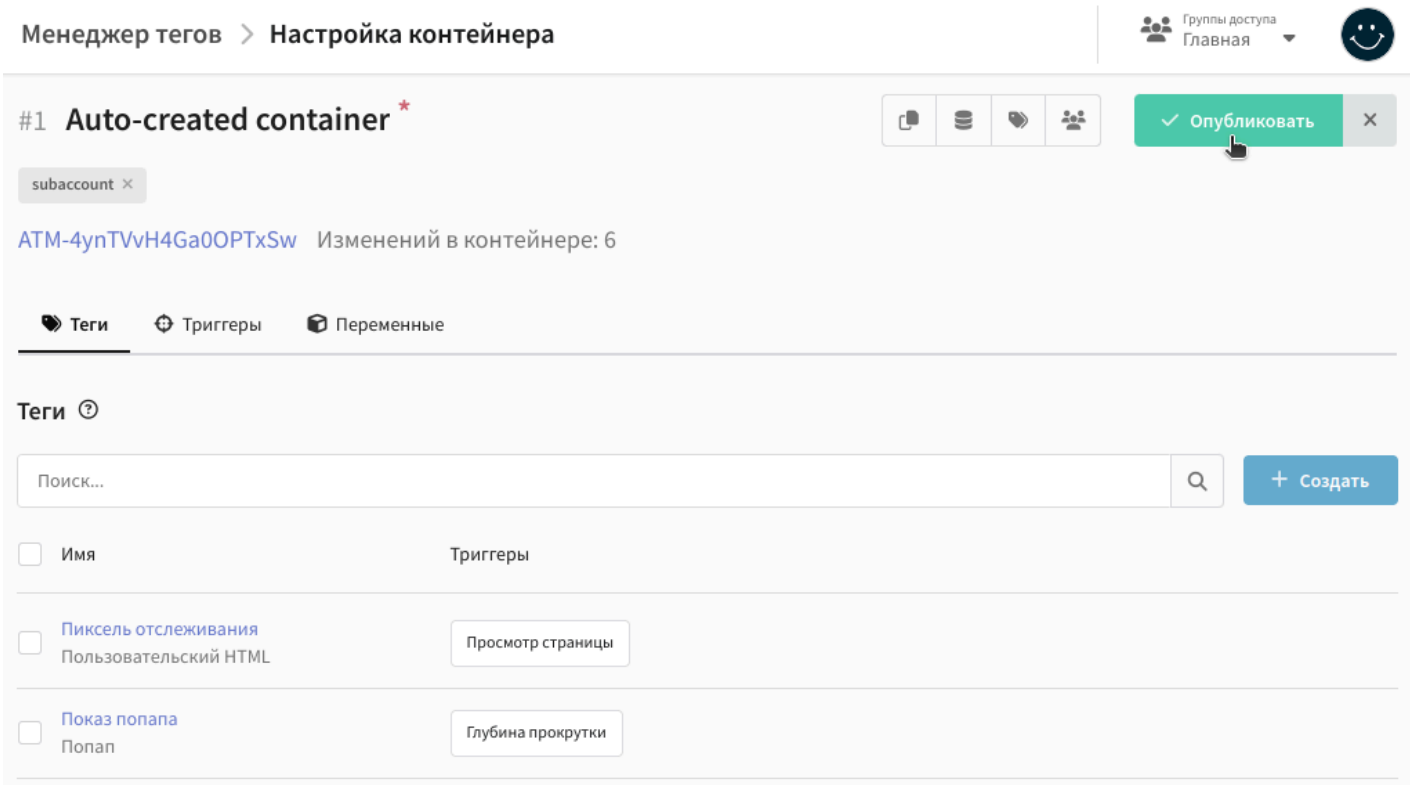
7. Задайте **настройки активации** и **ограничения** для тега.

8. Выберите и **настройте триггер**. Тег будет активироваться при наступлении условий, заданных в триггере. Если вы хотите ограничить аудиторию пользователей, для которых будет срабатывать триггер, дополнительно **настройте фильтр**.



9. Сохраните тег.

10. Нажмите **“Опубликовать”** —> **“Публикация и создание новой версии”**. Все изменения контейнера будут внесены на сайт, и пиксель начнёт отслеживание действий пользователей.



TIP

Больше информации о создании контейнера, настройке тегов и триггеров вы найдёте в **этой статье**.

Отчеты и аналитика

NOTE

Все отчёты можно экспортировать в форматах CSV и XLSX, для этого нажмите кнопку **“Выгрузить отчёт”** в меню действий.

Отчет по каналам

Отчет по каналам предоставляет данные по всем событиям, которые отслеживаются в рассылках: отправки, доставки, открытия, клики и др. Информацию можно получить как по одному каналу, так и по всем сразу. Вы можете сгруппировать данные по различным признакам, детализировать отчет или использовать фильтры для получения только необходимой информации. Также есть возможность сравнить показатели, например, за разные периоды времени или при использовании разных шаблонов.

Чтобы открыть отчёт, в главном меню выберите **Аналитика** —> **Каналы**.

> **INFO**
>
> Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

Настройка отчёта

Настройки включают:

- группировку данных;
- добавление фильтров;
- сохранение отчёта.

Справа находятся следующие кнопки:

Список отчётов — открыть список готовых отчетов (системных и сохраненных);

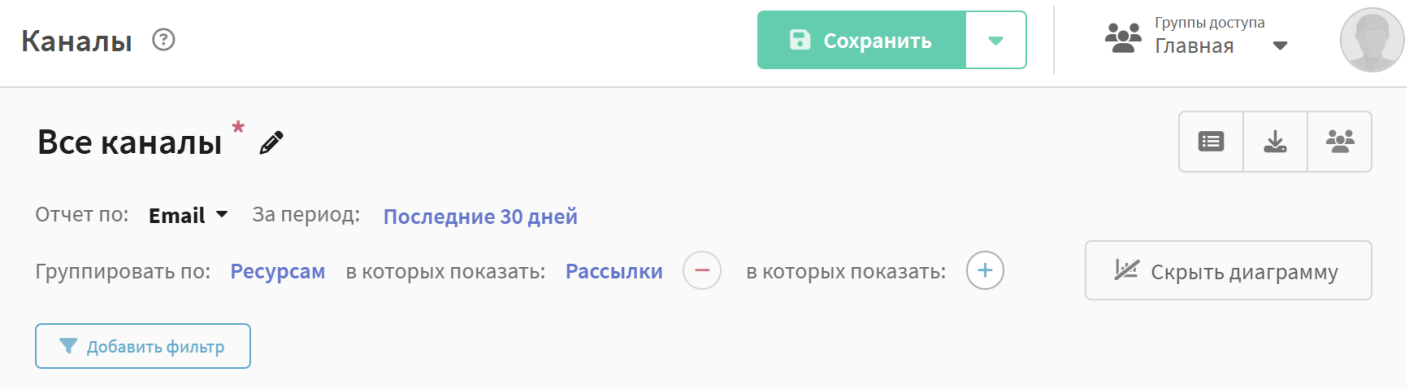
Выгрузить отчёт — экспортировать отчёт в csv или xlsx форматах;

Группы доступа — назначить группы доступа

Группировка данных

По умолчанию отчёт строится по всем каналам: Email, SMS, Push, Telegram, WhatsApp*, Viber™, Notify, а также пользовательским. Чтобы получить данные по конкретному каналу, выберите его в поле “**Отчет по**”.

Далее опередилите **период**. Вы можете вручную указать диапазон дат в календаре или выбрать один из готовых пресетов.



Выберите признак, по которому необходимо **сгруппировать** данные в отчете. После этого можно детализировать группировку, последовательно заполняя поля “**в которых показать**”. Группировка возможна до трёх уровней.

Признаки для группировки появятся в окне справа. Они разделены на три категории:

- **Дата и время**— группировать данные по часам, дням, неделям, месяцам, кварталам или годам;
- **Канал** — группировать по объектам платформы (ресурсы, сценарии, кампании, рассылки, сегменты и др.);
- **Событие** — группировать по ссылке, домену, доменной группе или поставщику.

Добавление фильтров

Категории	Доступные фильтры	Применение
Дата и время	• Часы(24) • Дни • Дни недели • Год и недели • Месяца • Кварталы • Года • Год и кварталы	С помощью фильтров можно добавить в отчёт или исключить из него данные за некоторый отрезок времени. Такие фильтры удобны, чтобы просмотреть статистику только за определенные дни или убрать из отчёта статистику за периоды праздников и выходных дней. Также можно исключить данные, собранные в период, когда на вашей стороне велись технические работы и рассылки не отправлялись.

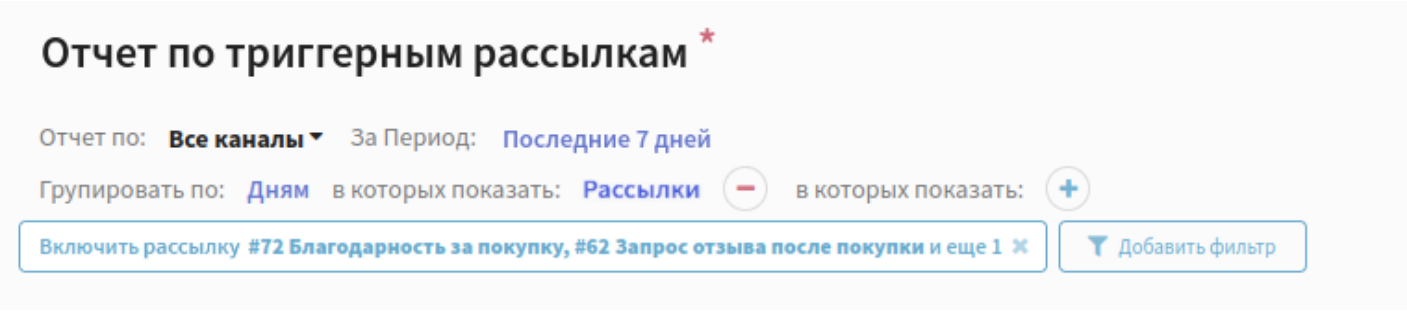
Канал	• **Кампании** — статистика по всем рассылкам и сценариям, входящим в выбранные кампании • **Каналы** — фильтр используется, чтобы отобразить данные по определенному каналу, если изначально отчёт строился по всем каналам. • **Ресурсы** — статистика по всем рассылкам, запускаемым в рамках ресурса. • **Сценарии** — статистика по всем рассылкам, запускаемым в рамках сценария автоматизации. • **Рассылки** — статистика по определенным рассылкам. • **Базы данных** — статистика по рассылкам, получатели которых входят в выбранную базу. • **Сегменты** — статистика по рассылкам, получатели которых входят в выбранные сегменты. • **Шаблоны** — статистика по всем рассылкам, в которых используются выбранные шаблоны. • **Отправщики** — виртуальные сендеры.	Фильтр по рассылкам поможет получить данные только по определенному типу рассылок. Например, если вы хотите проанализировать только показатели триггерных рассылок, выберите в списке соответствующие рассылки. Для поиска нужных рассылок можно ввести их название или id в поисковую строку или указать теги. Чтобы просмотреть показатели рассылок с определенным тегом, переключите Единичный отбор на Отбор по тегам. Подобная опция предусмотрена для любого объекта в категории "Канал". </td>
Событие	• **Поставщик** — добавить или исключить статистику рассылок в зависимости от значения поля Поставщик (vendor) у профиля в БД. • **Доменная группа** — данные по отправкам на определенную доменную группу (Gmail, Hotmail, Yandex). • **Домен** — данные по отправкам на определенный домен (gmail.com, example.com, altcraft.com и т. д.). • **Ссылка** — события рассылки, связанные с конкретными ссылками в шаблоне. Ссылки выбираются по атрибуту `name`.	При проблемах с доставляемостью можно просмотреть данные по каждому ISP отдельно, настроив фильтр **по доменной группе**. Если вы не хотите, чтобы в отчет попадали данные получателей с вашей корпоративной почты, выберите фильтр **по домену**, введите корпоративный домен и исключите его.

Чтобы отфильтровать данные, нажмите кнопку “**Добавить фильтр**”. Справа появится окно, где вы можете выбрать, какие данные включить в отчёт.

Если необходимо, напротив, убрать некоторые данные из отчёта, нажмите на переключатель **“Исключить”** в правом верхнем углу.

После настройки фильтра нажмите кнопку **“Применить”** внизу страницы.

Добавленные фильтры отобразятся в отчёте. Количество используемых фильтров не ограничено. Для удаления фильтра нажмите



Сохранение отчета

Вы можете сохранить текущие настройки отчёта. Для этого отредактируйте название отчёта (по умолчанию – “Все каналы”) и нажмите кнопку **“Сохранить”**. Это удобно, если вы часто запрашиваете данные, используя одни и те же группировки и фильтры.

Сохранённый отчёт будет доступен всем пользователей в рамках видимости групп.

Чтобы открыть список отчётов, кликните

. В новом окне отобразятся системные и сохраненные отчёты. Нажмите на название отчёта, чтобы просмотреть его.

Для удаления готового отчета нажмите

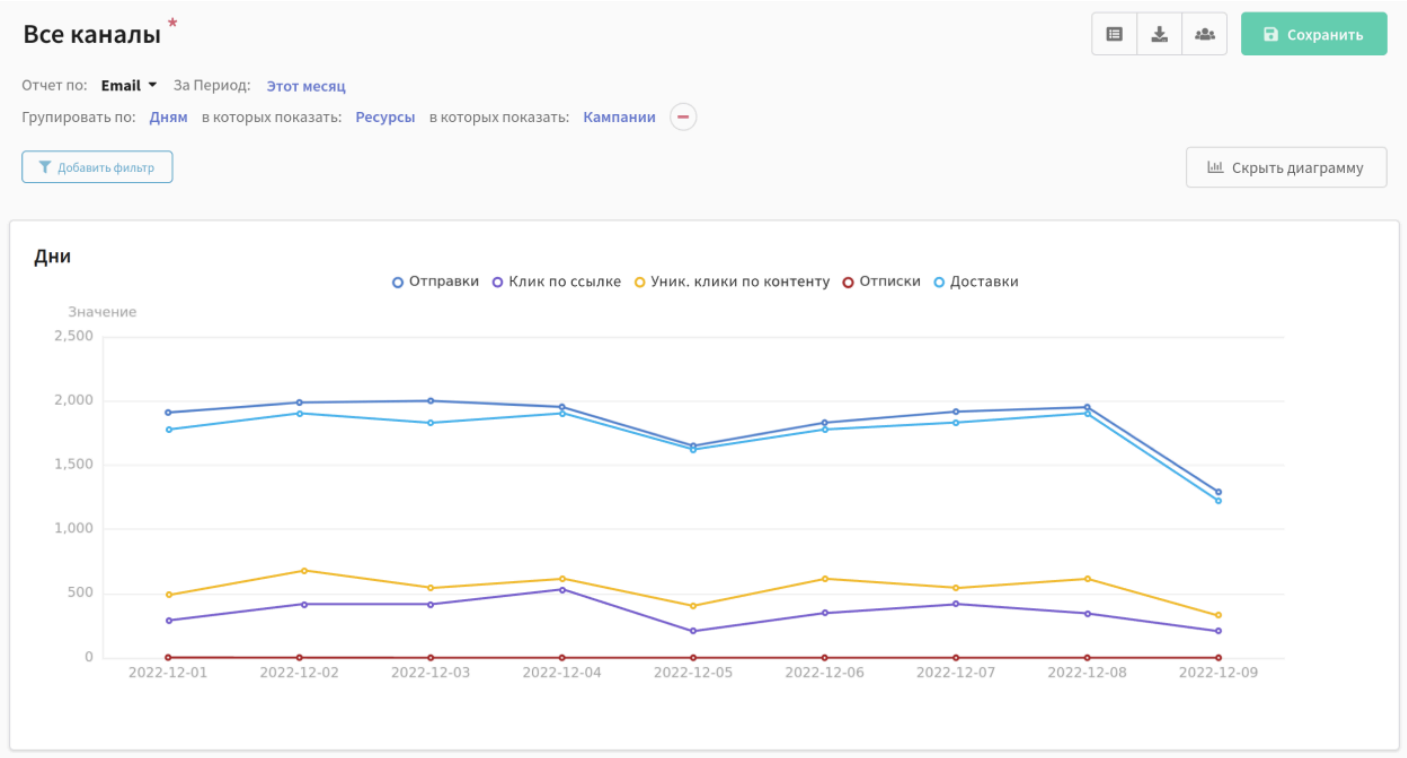
. При этом нельзя удалять отчёт, который вы просматриваете в данный момент.

Просмотр отчёта

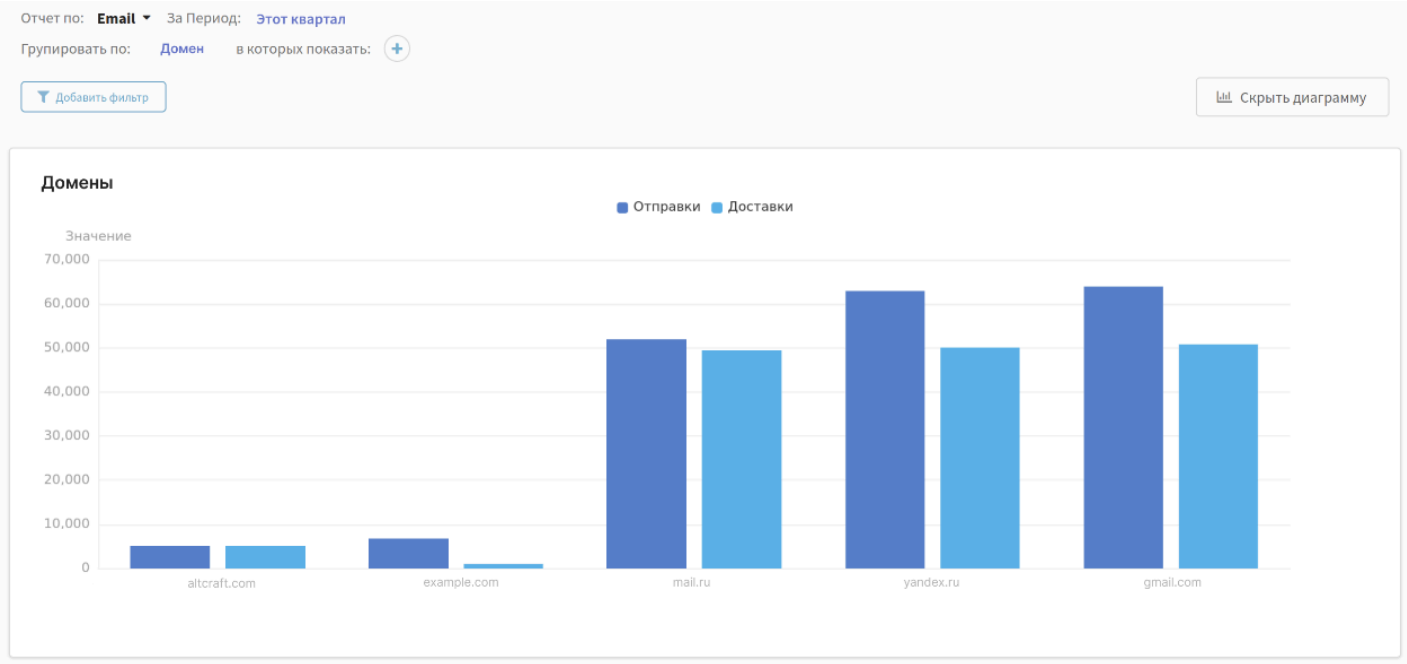
Отчет состоит из таблицы с данными и соответствующей им диаграммы. Диаграмму можно свернуть, нажав на кнопку “**Скрыть диаграмму**”.

Диаграмма

Диаграмма отображает данные на основе группировки. При группировке отчета **по дням, неделям и кварталам** диаграмма показывает динамику изменения показателей за выбранный период:



При группировке **по часам, дням недели, месяцам, годам,** а также **по каналу** или **событию** диаграмма показывает соотношение количественных или процентных показателей в выбранный период:



При наведении курсора на диаграмму вы увидите точные показатели.

В шапке перечислены показатели, отображаемые на диаграмме. На диаграмму может быть выведен любой столбец таблицы, но максимальное количество показателей — 7. Чтобы скрыть показатель, кликните по нему.

Таблица данных

В таблице собраны количественные и процентные показатели рассылок (отправки, доставки, клики, отписки, жалобы и т. д.). Описание показателей вы найдете ниже.

Показатели группируются на основе настроек отчёта. В зависимости от детализации группировки, структура таблицы может иметь вложенные элементы.

Например, при настройке “Группировать по **кампаниям**, в которых показать **Расслыки**, в которых показать **Каналы**” таблица отобразит иерархическую структуру эффективности маркетинговых стратегий.

При настройке “Группировать по **дням,** в которых показать **Ресурсы**, в которых показать **Рассылки**” таблица отобразит следующие данные:

Наименование	Отправки	Открытия	Прочтения	Клик по ссылке	Уник. открытия	Уник. клики по ссылке
Всего	16 494	17 127	16 549	16 434	16 476	16 434
2022-12-09	1 291	546	266	296	519	250
#18 Website	1 291	546	266	296	519	250
#266 Подборка рекомендованных то…	1 189	473	217	261	451	218
#62 Запрос отзыва после покупки	102	73	49	35	68	32
2022-12-08	1 952	2 047	1 951	1 951	1 951	1 951
2022-12-07	1 917	1 974	1 917	1 917	1 917	1 917
2022-12-06	1 831	1 897	1 830	1 820	1 829	1 820
2022-12-05	1 650	1 703	1 709	1 610	1 639	1 610
2022-12-04	1 954	2 021	1 954	1 954	1 954	1 954
2022-12-03	2 001	2 089	2 001	2 001	2 001	2 001

Чтобы **добавить новую метрику** в таблицу, наведите курсор на название столбца и нажмите

. В открывшемся списке выберите метрику. Если необходимо **удалить столбец,** нажмите

.

Для **перемещения столбца** по таблице также наведите курсор на название столбца, зажмите левую кнопку мыши и перетащите столбец в удобное место.

Наименование	Отправки	Доставки	Открытия	Клик по контенту	Клик по ссылке	Softbounced
Всего	0	0	0	0	0	0
2022-12-09	0	0	0	0	0	0

Другие опции:

— переключение количественных показателей в процентные;

— отображение столбца на диаграмме.

Сравнение данных

Режим сравнения позволяет увидеть данные по нескольким объектам одновременно. Например, можно сравнить эффективность двух шаблонов или оценить изменение показателей рассылки от периода к периоду.

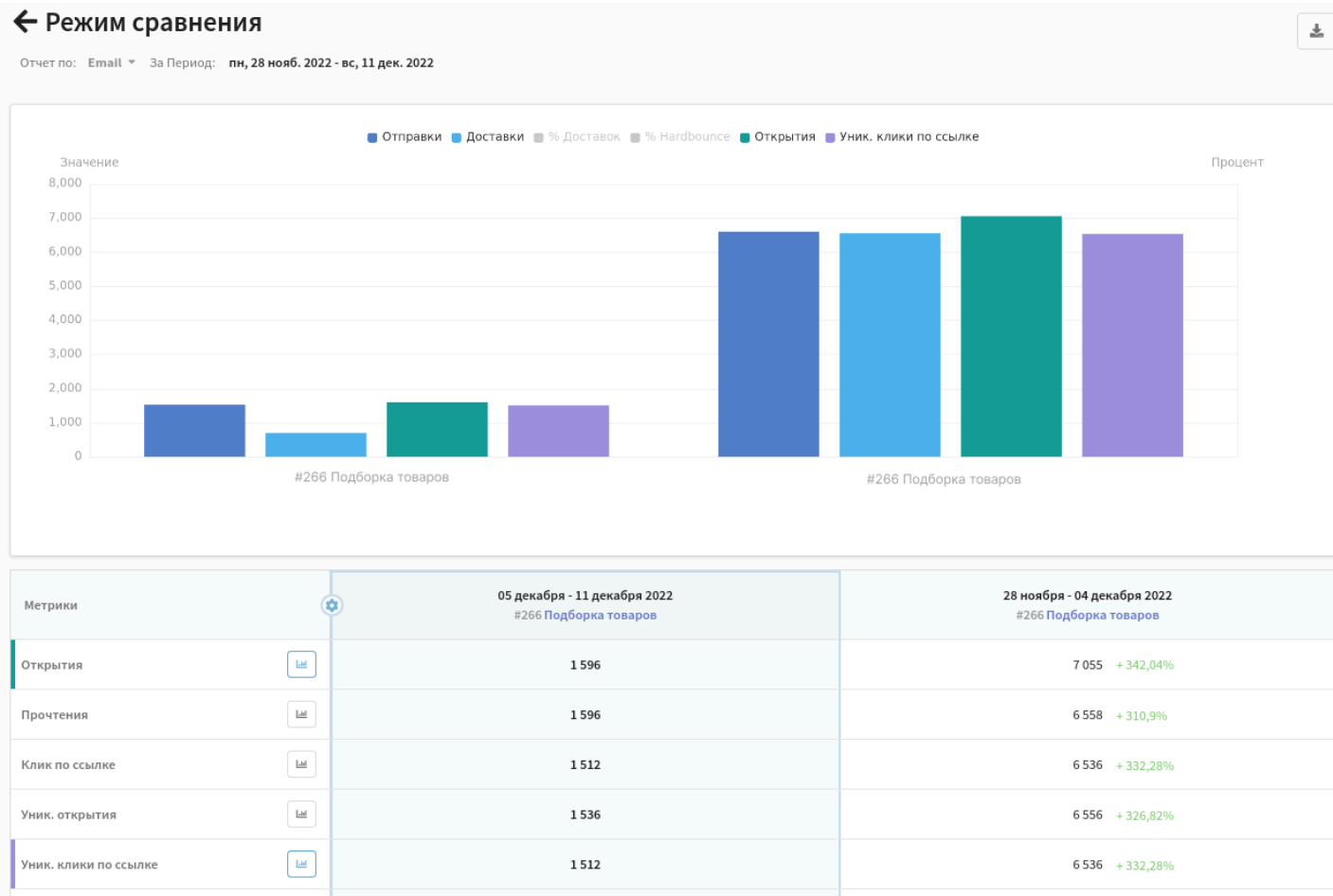
Для перехода в режим сравнения выделите в таблице объекты, которые хотите сравнить. Нажмите **“Сравнить”.** Возможно сравнивать объекты на разных уровнях вложенности.

Наименование — Сравнить	Отправки	Стоп список	Softbounced	Hardbounced
Всего	43 283	24	28	5
05 декабря - 11 декабря 2022	16 663	8	9	2
#266 Подборка товаров	8 134	4	7	1
#404 Новинки в ассортименте	8 529	4	2	1
28 ноября - 04 декабря 2022	16 609	6	10	0
#266 Подборка товаров	8 091	3	6	0
#404 Новинки в ассортименте	8 518	3	4	0

Режим сравнения также содержит диаграмму и таблицу с метриками. На диаграмме показатели по каждому объекту сгруппированы вместе.

Первый объект в таблице — это столбец сравнения. С этим объектом сравниваются оставшиеся объекты. Процент роста или снижения метрики высчитывается относительно этого объекта. Чтобы изменить столбец сравнения, нажмите

и выберите другой объект.



Чтобы вернуться в общий отчёт, нажмите ← **Режим сравнения**.

Показатели отчёта

INFO: Уникальное событие

Уникальность любого события ограничена временем, заданным в конфигурации платформы (параметр `UNIQUE_ACTIONS_TTL_MIN`), и по умолчанию составляет 1440 минут. Это значит, что повторное событие через сутки будет снова зачтено как уникальное. Уточните у администратора платформы, какое значение для уникальности установлено.

Все показатели

Показатель	Канал	Описание
Отправки	Все	Общее количество отправленных сообщений по всем каналам.
Недоставки	Все	Общее количество недоставленных сообщений по всем каналам. В недоставки входят Softbounced- и Hardbounced-сообщения, ошибки транспорта, а также сообщения с неопределенной причиной недоставки (например, неизвестные ошибки сети). Подробнее о недоставках в отчетах вы можете прочитать [здесь](/user-guide/reports-and-analytics/undeliveries).

% Недоставок	Все	Процент недоставленных сообщений по всем каналам.
Доставки	Все	Общее количество доставленных сообщений по всем каналам.
% Доставок	Все	Процент доставленных сообщений по всем каналам.
Открытия	Email Push	Количество открытых сообщений. Если один и тот же профиль открыл письмо несколько раз, будет зафиксировано каждое открытие.
Уник. Открытия	Email Push	Общее количество уникальных открытий сообщений по всем каналам. Если один профиль открыл письмо несколько раз, будет зафиксировано только одно уник. открытие.
% Открытий	Email Push	Процент получателей, открывших сообщение. (Уник. открытия / Доставки) * 100%.
Ответы	Email	Количество ответов на сообщение.
Прочтения	Email Whatsap p Viber Notify	Количество прочитанных сообщений.
Уник. прочтения	Email	Количество уникальных прочтений email-сообщения.
% Прочтений	Email	Процент уникальных прочтений email-сообщения. (Уник. Прочтения / Доставки) * 100%.
Просмотры	Email	Событие регистрируется, если пользователь открывает email-сообщение и просматривает его не более 4 секунд.
Беглые взгляды	Email	Событие регистрируется, если получатель просматривает email-сообщение от 4 до 8 секунд.
Клик по контенту	Все	Количество кликов по любым ссылкам в сообщении. При неоднократном клике получателем по одной ссылке, фиксируется каждое событие клика. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 6 кликов.
Уник. клики по контенту	Все, кроме Telegram Group	Количество получателей, кликнувших по любой ссылке в сообщении хотя бы один раз. Фиксирует факт взаимодействия получателя с ссылками в сообщении. По этому показателю рассчитывается CTOR и CTR контента.

Уник. клики по ссылке	Email Telegram Bot Notify	Количество первых переходов пользователя по каждой ссылке в сообщении. Считает только один клик на каждую ссылку, даже если пользователь переходил по ней многократно. По этот показатель рассчитывается CTOR и CTR ссылки.
CTR контента	Все, кроме Telegram Group	Процент кликов на любую ссылку в сообщении от доставок. (Уник. Клики по контенту / Доставки) * 100%.
CTOR контента	Email Push	Процент кликов на любую ссылку в сообщении от открытий. (Уник. клики по контенту / Уник. открытия) * 100%.
CTR ссылки	Email Telegram bot Whatsap Viber Notify	Процент переходов по ссылке от доставок. (Уник. клики по ссылке / Доставки) * 100%.
CTOR ссылки	Email	Процент переходов по ссылке от открытий. (Уник. клики по ссылке / Уник. открытия) * 100%.
Ошибки генерации	Все	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Все	Ошибка в работе протоколов при отправке сообщений.
Стоп-список	Все	Количество сообщений, которые не были отправлены, так как email-адрес или номер телефона получателя ранее были внесены в стоп-список.
Превышение политик	Все	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Softbounced	Email	Количество email, недоставленных из-за переполненности ящика или временной недоступности почтового сервера получателя.
% Softbounce	Email	Процент email, недоставленных из-за переполненности ящика или временной недоступности почтового сервера получателя.
Hardbounced	Email SMS Push	Количество сообщений, которые не были доставлены из-за недействительного email-адреса, номера телефона или идентификатора push-подписки.
% Hardbounce	Email SMS Push	Процент сообщений, которые не были доставлены из-за недействительного email-адреса, номера телефона или идентификатора push-подписки.

Жалобы	Email SMS Push	Количество сообщений, на которые последовали жалобы на спам (FBL).
% Жалоб	Email	Процент сообщений, на которые последовали жалобы на спам. (Жалобы / Доставки) * 100%.
Подписки	Все, кроме Telegram Group	Количество подписок на ресурсы в различных каналах.
Отписки	Email SMS Push Viber Notify	Количество отписок от ресурсов, т.е. переходов по ссылке отписки в сообщении.
Уник. отписки	Email SMS Push	Процент уникальных отписок от ресурсов.
% Отписок по доставкам	Email SMS Push	(Отписки / Доставки) * 100%.
% Отписок по открытиям	Email Push	(Отписки / Открытия) * 100%.
Undefined	Все	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети)

Email-канал

Показатель	Описание
Отправки	Количество отправленных email-сообщений.
Недоставки	Количество недоставленных email-сообщений.
% Недоставок	Процент недоставленных email-сообщений.
Доставки	Количество доставленных email-сообщений.

% Доставок	Процент доставленных email-сообщений.
Открытия	Количество открытых email-сообщений. Если один и тот же профиль открыл письмо несколько раз, будет зафиксировано каждое открытие.
Уник. открытия	Количество уникальных открытий email-сообщений. Если один профиль открыл письмо несколько раз, будет зафиксировано только одно открытие.
% Открытий	Процент получателей, открывших email. (Уник. открытия / Доставки) * 100%.
Ответы	Количество ответов на сообщение.
Прочтения	Количество прочитанных email-сообщений. Событие регистрируется, если получатель открывает письмо и просматривает его не менее 8 секунд.
Уник. прочтения	Количество уникальных прочтений email-сообщения. Если один профиль прочитал письмо несколько раз, будет зафиксировано только одно прочтение.
% Прочтений	Процент уникальных прочтений email-сообщения. (Уник. Прочтения / Доставки) * 100%.
Просмотры	Событие регистрируется, если пользователь открывает письмо и просматривает его не более 4 секунд.
Беглые взгляды	Событие регистрируется, если получатель просматривает письмо от 4 до 8 секунд.
Клик по контенту	Количество кликов по любым ссылкам в сообщении. При неоднократном клике получателем по одной ссылке, фиксируется каждое событие клика. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 6 кликов.
Уник. клики по контенту	Количество получателей, кликнувших по любой из ссылок в сообщении хотя бы один раз. Отслеживает факт взаимодействия получателя с ссылками в сообщении. Учитывает только первый клик пользователя независимо от количества последующих взаимодействий. При группировке по ссылкам позволяет отслеживать, какие ссылки пользователи выбирают первыми при взаимодействии с сообщением.
Уник. клики по ссылке	Количество первых переходов пользователя по каждой ссылке в сообщении. Считает только один клик на каждую ссылку, даже если пользователь переходил по ней многократно.
CTR контента	Процент кликов на любую ссылку в сообщении от доставок. (Уник. Клики по контенту / Доставки) * 100%.
CTOR контента	Процент кликов на любую ссылку в сообщении от открытий. (Уник. клики по контенту / Уник. открытия) * 100%.

CTR ссылки	Процент переходов по ссылке от доставок. (Уник. клики по ссылке / Доставки) * 100%.
CTOR ссылки	Процент переходов по ссылке от открытий. (Уник. клики по ссылке / Уник. открытия) * 100%.
Ошибки генерации	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Ошибка в работе протоколов при отправке сообщений.
Стоп список	Количество сообщений, которые не были отправлены, так как email-адрес получателя ранее был внесен в стоп список.
Превышение политик	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Softbounced	Количество email, недоставленных из-за переполненности ящика или временной недоступности почтового сервера получателя.
% Softbounce	Процент email, недоставленных из-за переполненности ящика или временной недоступности почтового сервера получателя.
Hardbounced	Количество сообщений, которые не были доставлены из-за недействительного email-адреса.
% Hardbounce	Процент сообщений, которые не были доставлены из-за недействительного email-адреса.
Жалобы	Количество сообщений, на которые последовали жалобы на спам (FBL).
% Жалоб	Процент сообщений, на которые последовали жалобы на спам. (Жалобы / Доставки) * 100
Подписки	Количество подписок на ресурсы в email-каналах.
Отписки	Количество отписок от ресурсов, т.е. переходов по ссылке отписки в сообщении.
Уник. отписки	Процент уникальных отписок от ресурсов.
% Отписок по доставкам	(Отписки / Доставки) * 100%.

% Отписок по открытиям	(Отписки / Открытия) * 100%.
Undefined	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети)

SMS-канал

Показатель	Описание
Отправки	Количество отправленных SMS-сообщений.
Недоставки	Количество недоставленных SMS-сообщений.
%Недоставок	Процент недоставленных SMS-сообщений.
Доставки	Количество доставленных SMS-сообщений.
% Доставок	Процент доставленных SMS-сообщений.
Клик по контенту	Количество кликов по ссылкам в сообщении.
Уник. клики по контенту	Количество получателей, кликнувших по любой из ссылок в сообщении хотя бы один раз. Отслеживает факт взаимодействия получателя с ссылками в сообщении. Учитывает только первый клик пользователя независимо от количества последующих взаимодействий.
CTR контента	Процент кликов от доставок. (Уник. Клики по контенту / Доставки) * 100%.
Ошибки генерации	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Ошибка в работе протоколов при отправке сообщений.
Стоп список	Количество сообщений, которые не были отправлены, так как номер телефона получателя ранее был внесен в стоп-список.
Превышение политик	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Hardbounced	Количество сообщений, которые не были доставлены из-за недействительного номера телефона.
% Hardbounce	Процент сообщений, которые не были доставлены из-за недействительного номера телефона.

Жалобы	Количество сообщений, на которые последовали жалобы на спам (FBL).
Подписки	Количество подписок на SMS-канал в ресурсах.
Отписки	Количество отписок от ресурсов, т.е. переходов по ссылке отписки в сообщении.
Уник. отписки	Процент уникальных отписок от ресурсов.
% Отписок по доставкам	(Отписки / Доставки) * 100%.
Undefined	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети)

Push-канал

Показатель	Описание
Отправки	Количество отправленных push-уведомлений.
Недоставки	Количество отправленных push-уведомлений, которые не удалось доставить до устройства получателя, включая hardbounces.
%Недоставок	Процент недоставленных push-уведомлений.
Доставки	Количество отправленных push-уведомлений, которые были успешно доставлены до устройства получателя.
% Доставок	Процент доставленных push-уведомлений.
Открытия	Количество зарегистрированных открытий push-уведомлений. **Веб push:** Количество открытий равно количеству доставок. **Мобильные push:** Регистрация открытий зависит от того, как реализована обработка event-ссылок на стороне клиента. Открытия фиксируются самим приложением и не зависят от наличия ссылок в push-уведомлении.
Уник. открытия	Общее количество уникальных открытий push-уведомлений. Если один профиль открыл уведомление несколько раз, будет зафиксировано только одно открытие.
% Открытий	Процент получателей, открывших push. (Уник. открытия / Доставки) * 100%.
Клик по контенту	Количество кликов в push-уведомлении. Для мобильных push учитываются, только если ссылка обернута в трекинг. Используется для отслеживания переходов на сайт или внешние ресурсы.
Уник. клики по контенту	Если один и тот же профиль кликнул несколько раз, будет зафиксирован только один клик по контенту.
CTR контента	Процент кликов от доставок. (Уник. Клики по контенту / Доставки) * 100%.

CTOR контента	Процент кликов от открытий. (Уник. клики по контенту / Уник. открытия) * 100%.
Ошибки генерации	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Ошибка в работе протоколов при отправке сообщений.
Превышение политик	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Hardbounced	Количество сообщений, которые не были доставлены из-за недействительного идентификатора push-подписки.
% Hardbounce	Процент сообщений, которые не были доставлены из-за недействительного идентификатора push-подписки.
Жалобы	Количество сообщений, на которые последовали жалобы на спам (FBL).
Подписки	Количество подписок на push-канал ресурсов.
Отписки	Количество отписок от ресурсов, т.е. переходов по ссылке отписки в сообщении.
Уник. отписки	Процент уникальных отписок от ресурсов.
% Отписок по доставкам	(Отписки / Доставки) * 100%.
% Отписок по открытиям	(Отписки / Открытия) * 100%.
Undefined	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети)

Telegram Bot канал

Показатель	Описание
Отправки	Количество отправленных сообщений подписчику в чат с Telegram ботом.
Недоставки	Количество недоставленных сообщений в чат с Telegram ботом
% Недоставок	Процент недоставленных сообщений в чат с Telegram ботом
Доставки	Количество доставленных сообщений в чат с Telegram ботом
% Доставок	Процент доставленных сообщений в чат с Telegram ботом
Клик по контенту	При неоднократном клике получателем по одной ссылке, фиксируется каждое событие клика. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 6 кликов.

Уник. клики по контенту	Событие регистрируется, если получатель кликнул по любой ссылке в сообщении. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксирован 1 уникальный клик по контенту.
Уник. клики по ссылке	Если получатель несколько раз кликнул по одной и той же ссылке, фиксируется только одно событие клика для каждой ссылки. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 3 уникальных клика по ссылке.
CTR контента	Процент кликов на любую ссылку в сообщении от доставок. (Уник. Клики по контенту / Доставки) * 100%.
CTR ссылки	Процент переходов по ссылке от доставок. (Уник. клики по ссылке / Доставки) * 100%.
Ошибки генерации	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Ошибка в работе протоколов при отправке сообщений.
Превышение политик	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Подписки	Профиль подписан на Telegram бот.
Undefined	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети)

Telegram Group канал

Показатель	Описание
Отправки	Общее количество отправленных сообщений в Telegram группу
Недоставки	Общее количество недоставленных сообщений в Telegram группу
% Недоставок	Процент недоставленных сообщений в Telegram группу
Доставки	Общее количество доставленных сообщений в Telegram группу
% Доставок	Процент доставленных сообщений в Telegram группу
Клик по контенту	Количество кликов по любым ссылкам в сообщении.
Ошибки генерации	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Ошибка в работе протоколов при отправке сообщений.
Превышение политик	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Undefined	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети)

WhatsApp-канал

Показатель	Описание
Отправки	Количество отправленных в WhatsApp* сообщений
Недоставки	Количество недоставленных в WhatsApp* сообщений
% Недоставок	Процент недоставленных в WhatsApp* сообщений
Доставки	Количество доставленных в WhatsApp* сообщений
% Доставок	Процент доставленных в WhatsApp* сообщений
Прочтения	Количество сообщений, прочитанных пользователем WhatsApp*.
Клик по контенту	При неоднократном клике получателем по одной ссылке, фиксируется каждое событие клика. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 6 кликов.
Уник. клики по контенту	Событие регистрируется, если получатель кликнул по любой ссылке в сообщении. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксирован 1 уникальный клик по контенту.
Уник. клики по ссылке	Если получатель несколько раз кликнул по одной и той же ссылке, фиксируется только одно событие клика для каждой ссылки. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 3 уникальных клика по ссылке.
CTR контента	Процент кликов по любой ссылке сообщения от доставок. (Уник. Клики по контенту / Доставки) * 100%.
CTR ссылки	Процент переходов по ссылке от доставок. (Уник. клики по ссылке / Доставки) * 100%.
CTOR ссылки	Процент переходов по ссылке от открытий. (Уник. клики по ссылке / Уник. открытия) * 100%
Ошибки генерации	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Ошибка в работе протоколов при отправке сообщений.
Стоп список	Количество сообщений, которые не были отправлены, так как номер телефона получателя ранее был внесен в стоп-список.
Превышение политик	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Подписки	Профиль подписался на WhatsApp*-канал.

Undefined	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети)

Viber™-канал

Показатель	Описание
Отправки	Сообщение отправлено для доставки подписчику в Viber™ чат.
Недоставки	Сообщение было отклонено Viber.
% Недоставок	Процент недоставленных сообщений в Viber™ чат.
Доставки	Сообщение доставлено получателю в Viber™ чат.
% Доставок	Процент доставленных сообщений в Viber™ чат.
Прочтения	Сообщение прочитано получателем.
Уник. Прочтения	Количество уникальных прочтений в чате Viber™.
Клик по контенту	При неоднократном клике получателем по одной ссылке, фиксируется каждое событие клика. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 6 кликов.
Уник. клики по контенту	Событие регистрируется, если получатель кликнул по любой ссылке в сообщении. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксирован 1 уникальный клик по контенту.
Уник. клики по ссылке	Если получатель несколько раз кликнул по одной и той же ссылке, фиксируется только одно событие клика для каждой ссылки. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 3 уникальных клика по ссылке.
CTR контента	Процент кликов на любую ссылку в сообщении от доставок. (Уник. Клики по контенту / Доставки) * 100%.
CTR ссылки	Процент переходов по ссылке от доставок. (Уник. клики по ссылке / Доставки) * 100%.
Ошибки генерации	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Ошибка в работе протоколов при отправке сообщений.
Стоп список	Количество сообщений, которые не были отправлены, так как номер телефона получателя ранее был внесен в стоп-список.

Превышение политик	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Подписки	Количество подписок на ресурсы в этом канале.
Отписки	Количество отписок от ресурсов, т.е. переходов по ссылке отписки в сообщении.
Undefined	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети)

Notify-канал

Показатель	Описание
Отправки	Сообщение отправлено для доставки подписчику по каналу Notify.
Недоставки	Сообщение было отклонено Notify.
% Недоставок	Процент недоставленных сообщений по каналу Notify.
Доставки	Сообщение доставлено получателю впо каналу Notify.
% Доставок	Процент доставленных сообщений по каналу Notify.
Прочтения	Сообщение прочитано получателем.
Клик по контенту	При неоднократном клике получателем по одной ссылке, фиксируется каждое событие клика. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 6 кликов.
Уник. клики по контенту	Событие регистрируется, если получатель кликнул по любой ссылке в сообщении. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксирован 1 уникальный клик по контенту.
Уник. клики по ссылке	Если получатель несколько раз кликнул по одной и той же ссылке, фиксируется только одно событие клика для каждой ссылки. Например, если в сообщении 3 ссылки и по каждой из них получатель перешёл 2 раза, то будет зафиксировано 3 уникальных клика по ссылке.
CTR контента	Процент кликов на любую ссылку в сообщении от доставок. (Уник. Клики по контенту / Доставки) * 100%.
CTR ссылки	Процент переходов по ссылке от доставок. (Уник. клики по ссылке / Доставки) * 100%.
Ошибки генерации	Ошибка при формировании шаблона. Возникает, когда у профиля отсутствует подписка на выбранный в рассылке ресурс.
Ошибки транспорта	Ошибка в работе протоколов при отправке сообщений.

Стоп список	Количество сообщений, которые не были отправлены, так как номер телефона получателя ранее был внесен в стоп-список.
Превышение политик	Количество сообщений, не отправленных по причине превышения лимита, установленного в политике отправки.
Подписки	Количество подписок на ресурсы в этом канале.
Отписки	Количество отписок от ресурсов, т.е. переходов по ссылке отписки в сообщении.
Undefined	Не удалось определить причину недоставки (например, произошла неизвестная ошибка сети).

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Отчёт по трафику

Отчёт по трафику предоставляет данные о трафике профилей из таких источников как каналы, формы, пиксели и попапы. Вы можете сгруппировать данные по определённым признакам, детализировать отчёт или использовать фильтры, чтобы убрать лишнюю информацию. Также есть возможность сравнить данные, например, за разные периоды.

В основе отчёта лежит hit — единица измерения трафика. Hit засчитывается, когда в результате действия пользователя на сайте или в приложении отправляется запрос к серверу. Действием может быть регистрация цели или клик в сообщении.

Настройка отчёта

Настройки включают:

- группировку данных;
- добавление фильтров;
- сохранение отчёта.

Справа находятся следующие кнопки:

Список отчётов — открыть список готовых отчетов (системных и сохраненных);

Выгрузить отчёт — экспортировать отчёт в csv или xlsx форматах;

Группы доступа — назначить группы доступа

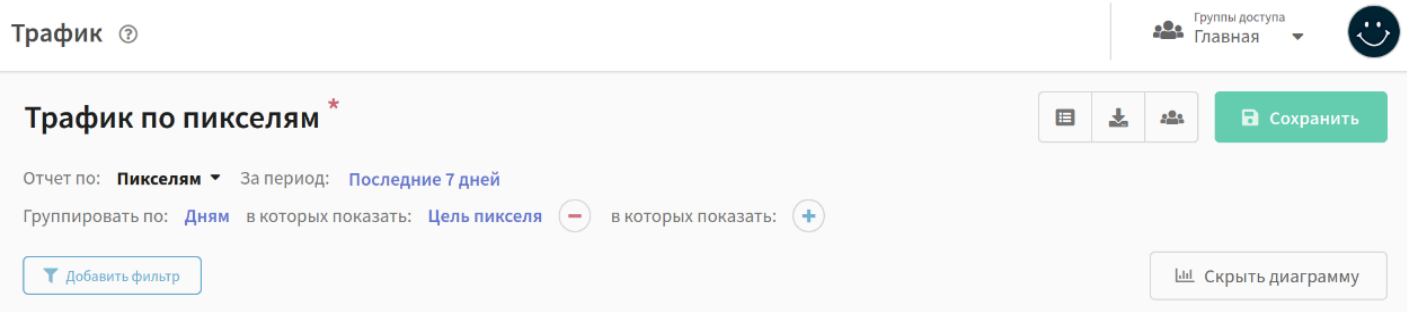
Группировка данных

Отчёт предоставляет данные по следующим источникам трафика:

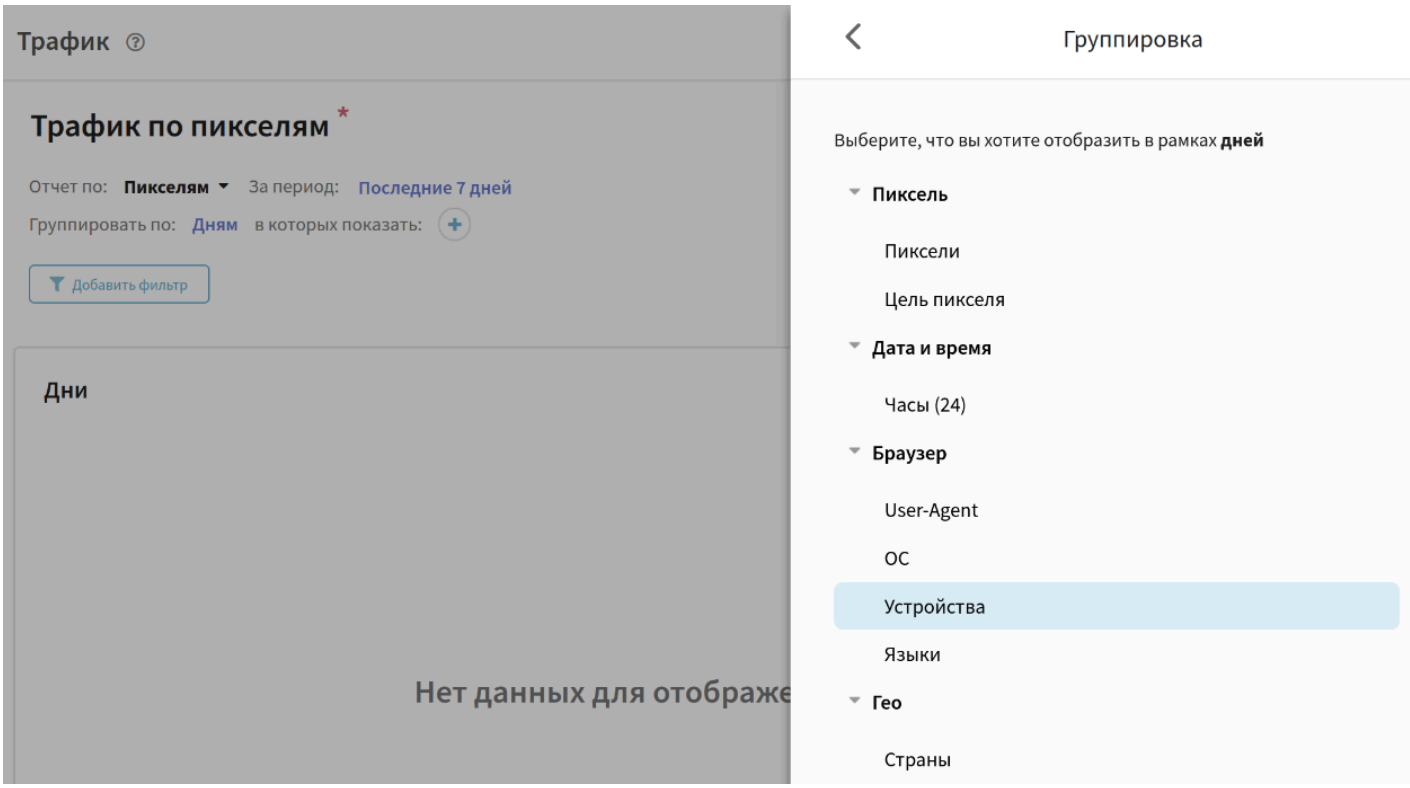
- **Каналы** — статистика по подпискам и отпискам, открытиям и кликам в сообщениях.
- **Пиксели** — статистика по целям, которые зарегистрированы пикселями.
- **Формы** — статистика по загрузкам и отправкам веб-форм.
- **Попапы** — статистика по показам и заполнениям попапов.

Чтобы получить данные по конкретному источнику трафика, выберите его в поле “**Отчет по**”.

Далее опеределите **период**. Вы можете вручную указать диапазон дат в календаре или выбрать один из готовых пресетов.



Выберите признак, по которому необходимо **сгруппировать** данные в отчёте. После этого можно детализировать группировку, последовательно заполняя поля “**в которых показать**”. Группировка возможна до трёх уровней. Варианты группировки появятся в окне справа:



Подробнее о группировке

Категория	Группирует по
Канал (только для отчёта по каналам)	Ресурсы Цепочки (сценарии) Рассылки Базы данных Сегменты Отправщики (сендеры) Шаблоны Каналы
Пиксель (только для отчёта по пикселям)	Пиксель Цель пикселя
Форма (только для отчёта по формам)	Формы Страницы формы

Категория	Группирует по
Попап (только для отчёта по попапам)	Попапы
Дата и время	Часы Дни Дни недели Недели Месяцы Кварталы Год и кварталы Годы
Браузер	User-Agent — информация о браузере профиля. ОС — операционная система. Устройства — веб или мобильное устройство. Языки — язык браузера.
Гео	Страны Страны и города Временные зоны Гео-точки
UTM	UTM Content — параметр ссылки utm_content UTM Medium — параметр ссылки utm_medium UTM Source — параметр ссылки utm_source UTM Campaign — параметр ссылки utm_campaign UTM Term — параметр ссылки utm_term
Referer	URL, с которого была отправлена форма или зарегистрирована цель. Например, если форма отправлена с URL `https://pxl.example.com/form?form=8fucSs16eZDhkYDKkKfq`, то: • Referer домен — это `pxl.example.com` • Referer путь — это `/form?form=8fucSs16eZDhkYDKkKfq`
Origin	Источник показывает домен сайта, с которого произошёл запрос к серверу при отправке формы или активации пикселя. • Origin домены • Origin пути
Событие	Домен — почтовые домены профилей (gmail.com, yandex.ru, mail.ru и др.). Доменная группа — например: Gmail, Hotmail, Yandex. Поставщик — значение поля `Поставщик (vendor)` у профиля в БД. Ссылки — группировка по атрибуту `name` в ссылках сообщения.

TIP

Не все браузеры присылают ответы с уточнением устройства, браузера и операционной системы в заголовке. Группировки по “Устройствам”, “ОС” и “User Agent” содержат в себе “Остальное” для таких случаев.

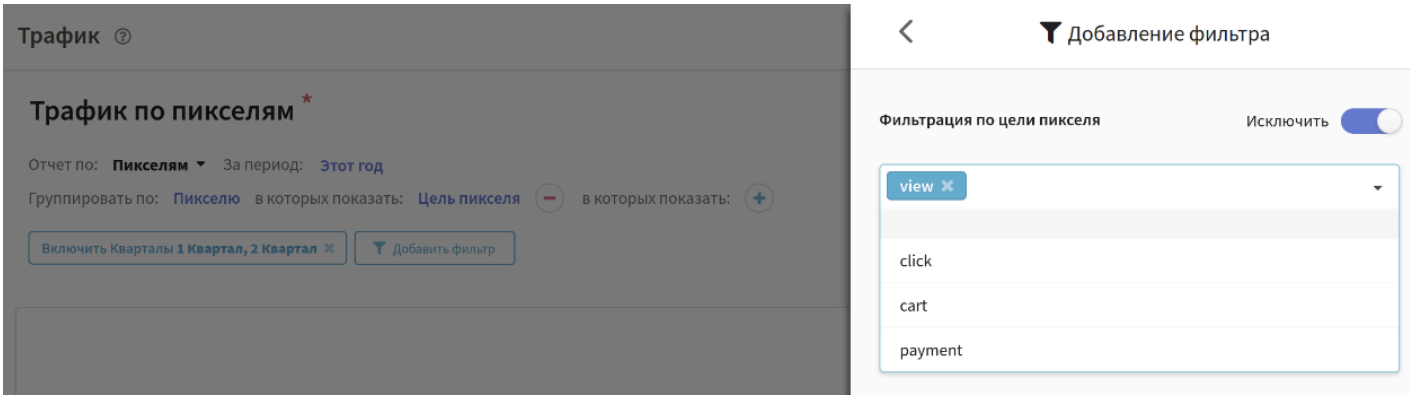
Если же заголовок был передан, но система не смогла считать его, то информация о событии будет записана в “Other”.

Добавление фильтров

С помощью фильтров вы можете оставлять в отчёте только необходимую информацию. Используйте фильтры, если хотите просмотреть данные по определённым формам, целям, ресурсам и т. д. Категории фильтров совпадают с категориями группировок.

Чтобы отфильтровать данные, нажмите кнопку “**Добавить фильтр**”. Справа появится окно, где вы можете выбрать, какие данные включить в отчёт.

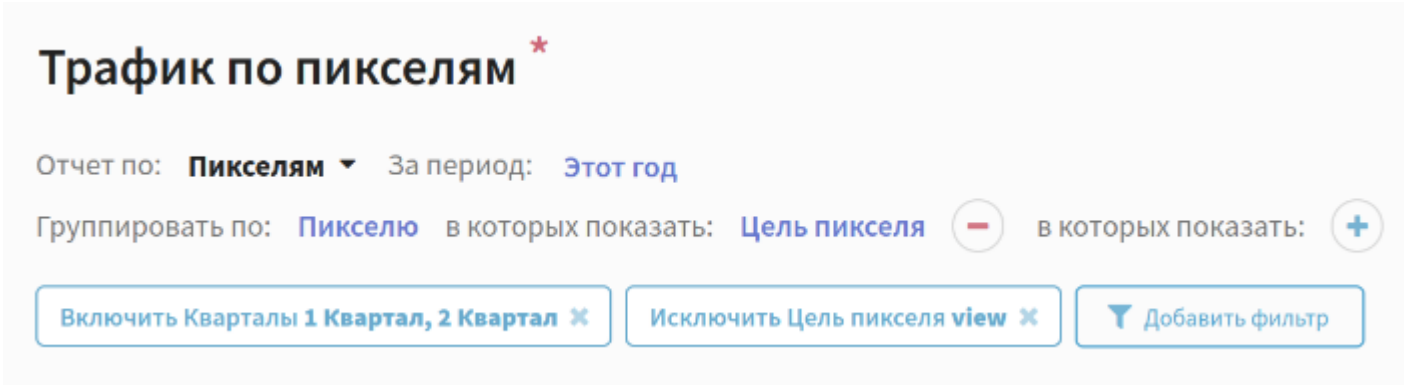
Если необходимо убрать некоторые данные из отчёта, нажмите **“Исключить”** в правом верхнем углу.



После настройки фильтра нажмите кнопку **“Применить”** внизу страницы.

Добавленные фильтры отобразятся в отчёте. Количество используемых фильтров не ограничено. Для удаления фильтра нажмите

.



Сохранение отчета

Вы можете сохранить текущие настройки отчёта. Для этого отредактируйте название отчёта (по умолчанию – “*Трафик каналов*”) и нажмите кнопку **“Сохранить”**. Это удобно, если вы часто запрашиваете данные,

используя одни и те же группировки и фильтры.

Сохранённый отчёт будет доступен всем пользователям в рамках видимости групп.

Чтобы открыть список отчётов, кликните

. В новом окне отобразятся системные и сохраненные отчёты. Нажмите на название отчёта, чтобы просмотреть его.

Для удаления готового отчета нажмите

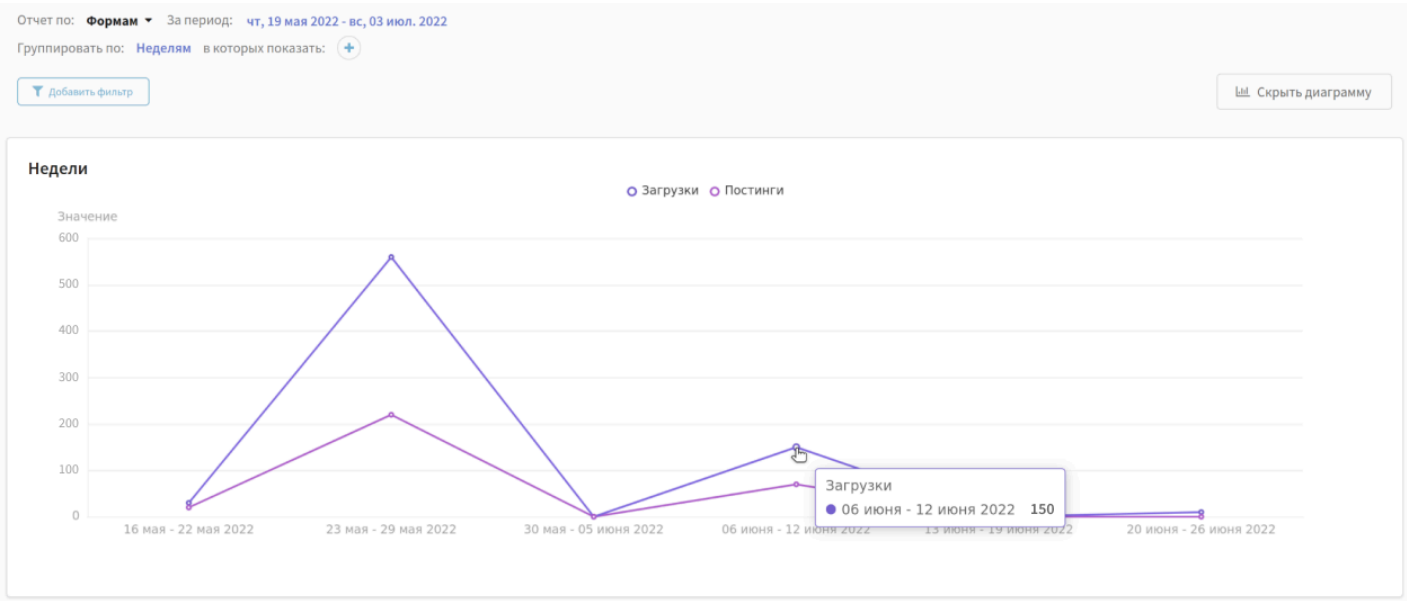
. При этом нельзя удалять отчёт, который вы просматриваете в данный момент.

Просмотр отчёта

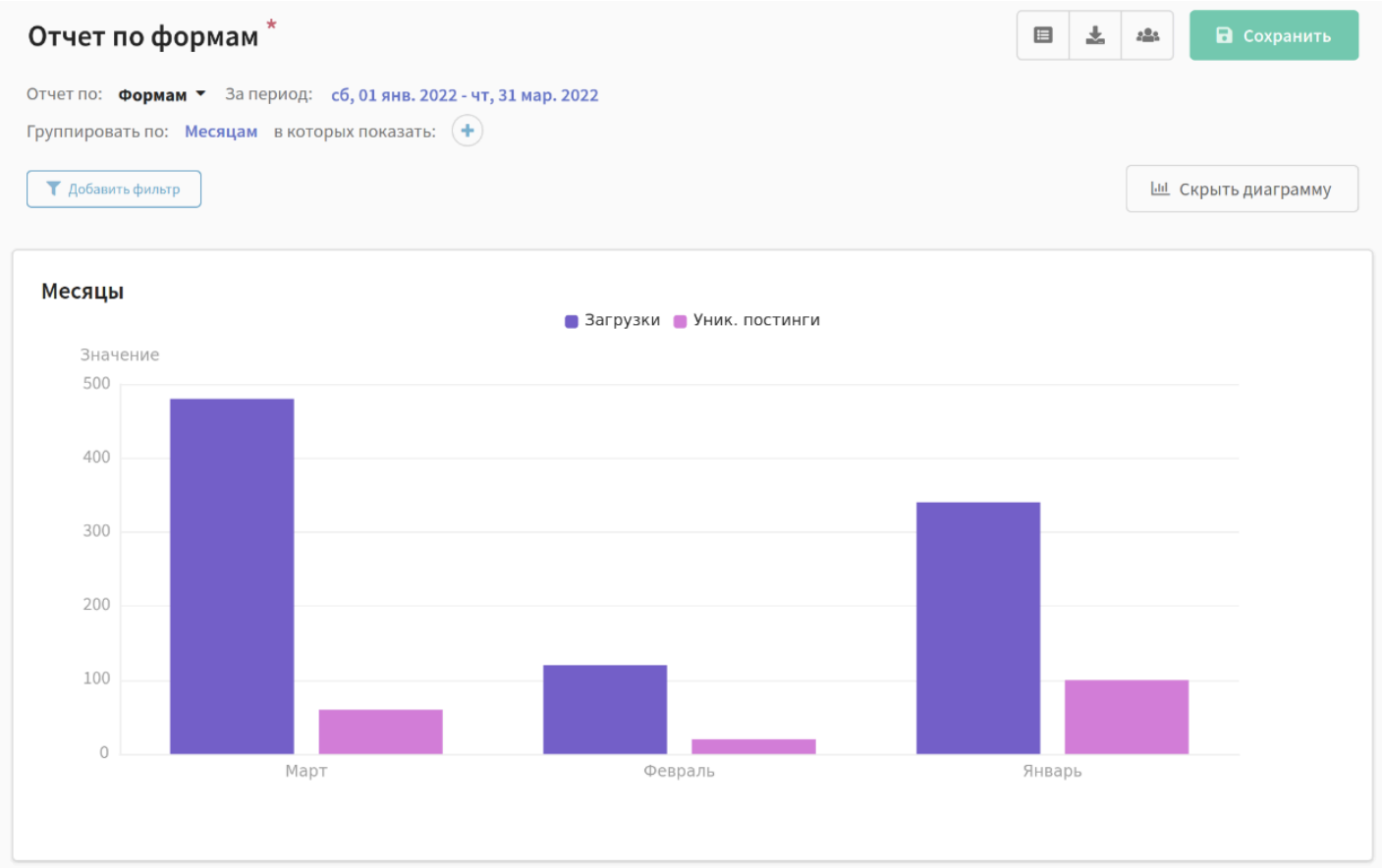
Отчет состоит из таблицы с данными и соответствующей им диаграммы. Диаграмму можно свернуть, нажав на кнопку “**Скрыть диаграмму**”.

Диаграмма

Диаграмма отображает данные на основе группировки. При группировке отчета **по дням, неделям и кварталам** диаграмма показывает динамику изменения показателей во времени:



В остальных случаях диаграмма показывает соотношение количественных или процентных показателей в выбранный период:



При наведении курсора на диаграмму вы увидите точные показатели.

В шапке перечислены показатели, отображаемые на диаграмме. На диаграмму может быть выведен любой столбец таблицы. Чтобы скрыть показатель, кликните по нему.

Таблица данных

В таблице собраны количественные и процентные показатели по источникам трафика.

Показатели группируются на основе настроек отчёта. В зависимости от детализации группировки, структура таблицы может иметь вложенные элементы.

Например, при настройке `“Группировать по **Дням**, в которых показать **Пиксели**, в которых показать **Цель пикселя**”` таблица отобразит следующие данные:

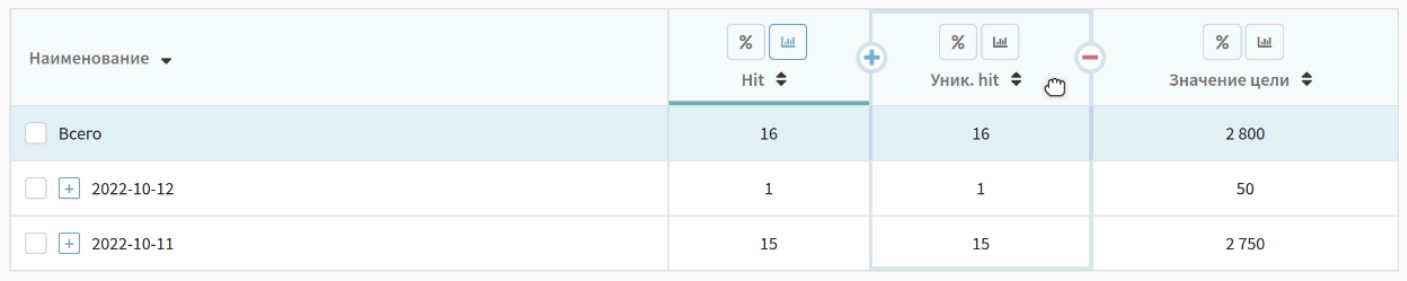
Наименование	Hit	Уник. hit	Значение цели
Всего	16	16	2 800
2022-10-12	1	1	50
2022-10-11	15	15	2 750
#18 Тестовый пиксель	2	2	150
#17 Интернет-магазин	6	6	2 200
view	2	2	200
payment	3	3	1 500

Чтобы **добавить новую метрику** в таблицу, наведите курсор на название столбца и нажмите

. В открывшемся списке выберите метрику. Если необходимо **удалить столбец,** нажмите

.

Для **перемещения столбца** по таблице также наведите курсор на название столбца, зажмите левую кнопку мыши и перетащите столбец в удобное место.



Другие опции:

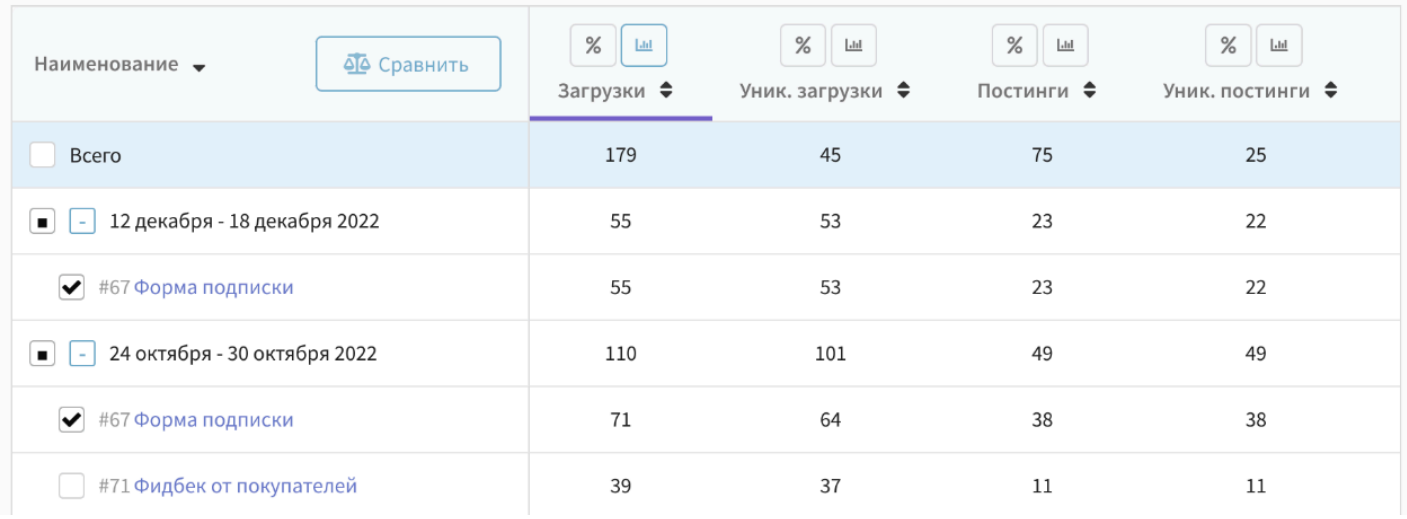
— переключение количественных показателей в процентные;

— отображение столбца на диаграмме.

Сравнение данных

Режим сравнения позволяет просмотреть данные по нескольким объектам одновременно. Например, можно сравнить динамику заполнения формы в разные периоды.

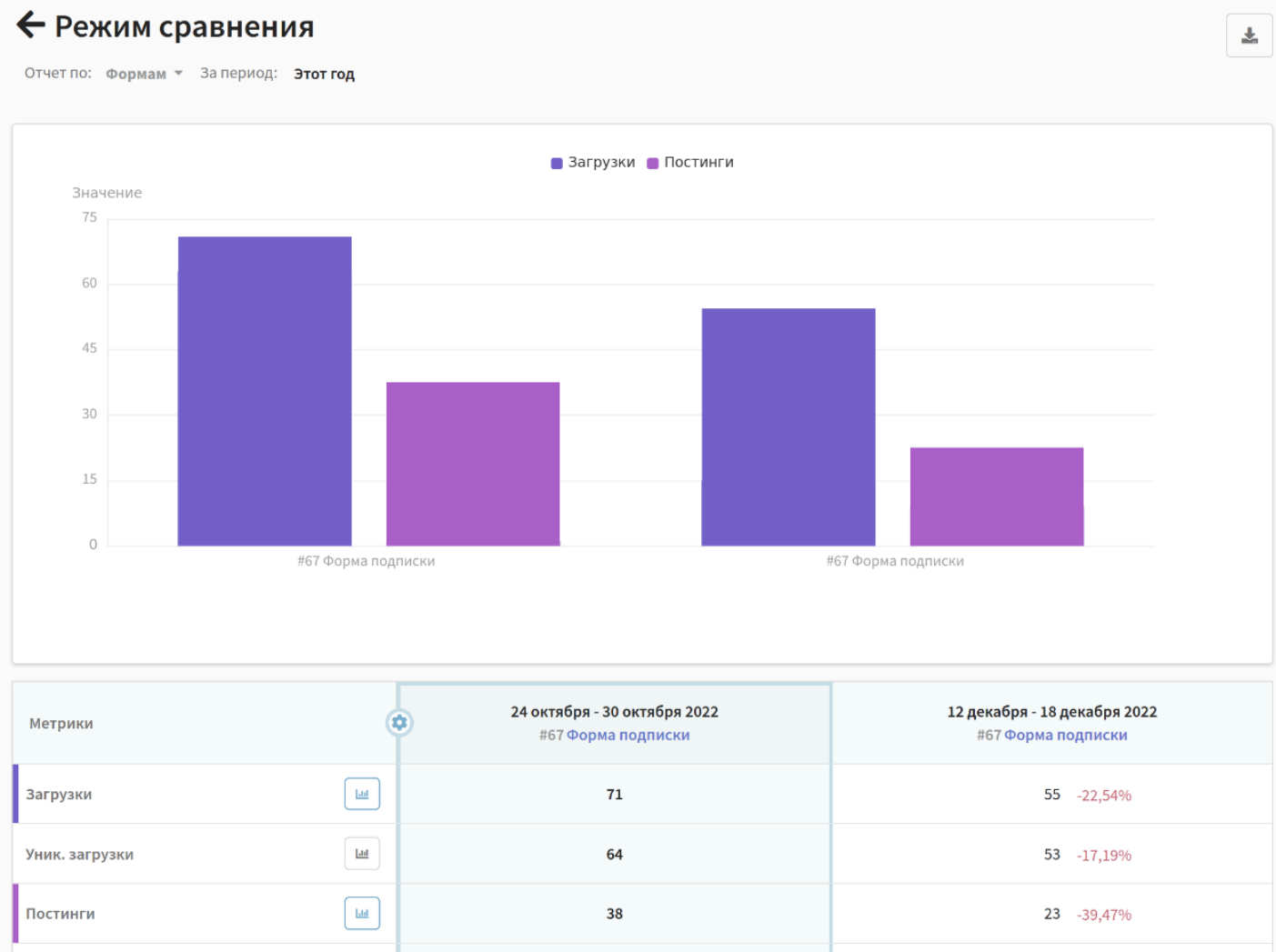
Для перехода в режим сравнения отметьте в таблице объекты, которые хотите сравнить друг с другом. Нажмите **“Сравнить”.** Возможно сравнивать любое количество объектов на разных уровнях вложенности.



Режим сравнения также содержит диаграмму и таблицу с метриками. На диаграмме показатели по каждому объекту сгруппированы вместе.

Первый объект в таблице — это столбец сравнения. С этим объектом сравниваются остальные объекты. Процент роста или снижения метрики высчитывается относительно этого объекта. Чтобы изменить столбец сравнения, нажмите

и выберите другой объект.



Чтобы вернуться в общий отчёт, нажмите ← **Режим сравнения**.

Показатели отчёта по трафику

Отчёт по	Показатель	Описание

Каналы	Открытия	Количество сообщений, открытых профилями.
	Уник. открытия	Количество уникальных открытий. Если один профиль открывает сообщение два раза, фиксируется только одно уникальное открытие.
	Отписки	Количество отписок от ресурсов в различных каналах.
	Подписки	Количество подписок на ресурсы в различных каналах.
	Подтверждения	Количество переходов по ссылке подтверждения (ссылка с атрибутом confirm-link).
	Hit	Количество кликов по ссылке в сообщениях.

	Уник. hit	Количество уникальных кликов по ссылке в сообщениях.
		`Если один профиль кликает по ссылке в сообщении два раза, фиксируется только один уникальный hit.`
		`</td>`

Пиксели	Hit	Количество целей, зарегистрированных пикселями.
	Уник. hit	Количество уникальных регистраций цели. Если для одного профиля цель регистрируется несколько раз, фиксируется только один уникальный hit.
	Значение цели	Сумма значений всех целей, зарегистрированных пикселями.
	Значение за hit	(Значение цели / Hit) * 100%
Формы	Загрузки	Количество загрузок страницы с формой.

	Уник. загрузки	Количество уникальных загрузок.
		`Если один профиль загружает форму несколько раз, фиксируется только одна уникальная загрузка.`
		`</td>`

	Постинги	Количество успешно отправленных форм.

	Уник. постинги	Количество уникальных отправок.
		`Если один профиль отправляет форму несколько раз, фиксируется один уникальный постинг.`
		`</td>`

	% Постингов	(Уник. постинги / уник. загрузки) * 100%
	Время заполнения	Продолжительность взаимодействия с формами (в секундах). Суммируется время, затраченное каждым профилем на заполнение формы.

Попапы	Попаданий в таргетинг	Количество раз, когда были выполнены условия для появления попапа, заданные в триггере. При этом попап мог быть не показан пользователю из-за лимитов тега, заданных в блоках "Настройки активации" и "Ограничения".
	Показы	Количество показов попапа пользователям.
	Уник. показы	Если попап показан пользователю несколько раз в рамках одной сессии, фиксируется только один уникальный показ.
	Уник. CTA	Количество кликов, пока нет заполнений, или количество заполнений, если они были.
	Закрытия	Количество кликов по кнопке "Закрыть" (closeButton).
	Уник. закрытия	Если пользователь несколько раз закрыл попап в рамках одной сессии, фиксируется одно уникальное закрытие.

INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

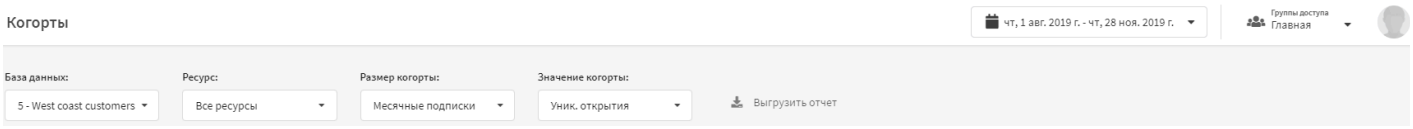
Когортный отчёт

Когорта – это группа подписчиков, привлечённых в определённый временной промежуток. Панельный анализ помогает проследить изменения в активности каждой когорты с течением времени.

> **TIP**
>
> Сравнивая когорты и даты акций по привлечению клиентов, можно спрогнозировать эффективность будущих предложений и проанализировать результаты недавно проведённых акций.

Формирование отчёта

Начните работу с выбора **периода** для составления отчёта (в верхней части страницы). Далее выберите **базу профилей** для исследования и **ресурс**, на который подписаны клиенты:



Размер когорты может быть равен **неделе** или **месяцу**. Это и есть временной промежуток, когда клиенты были импортированы в базу данных.

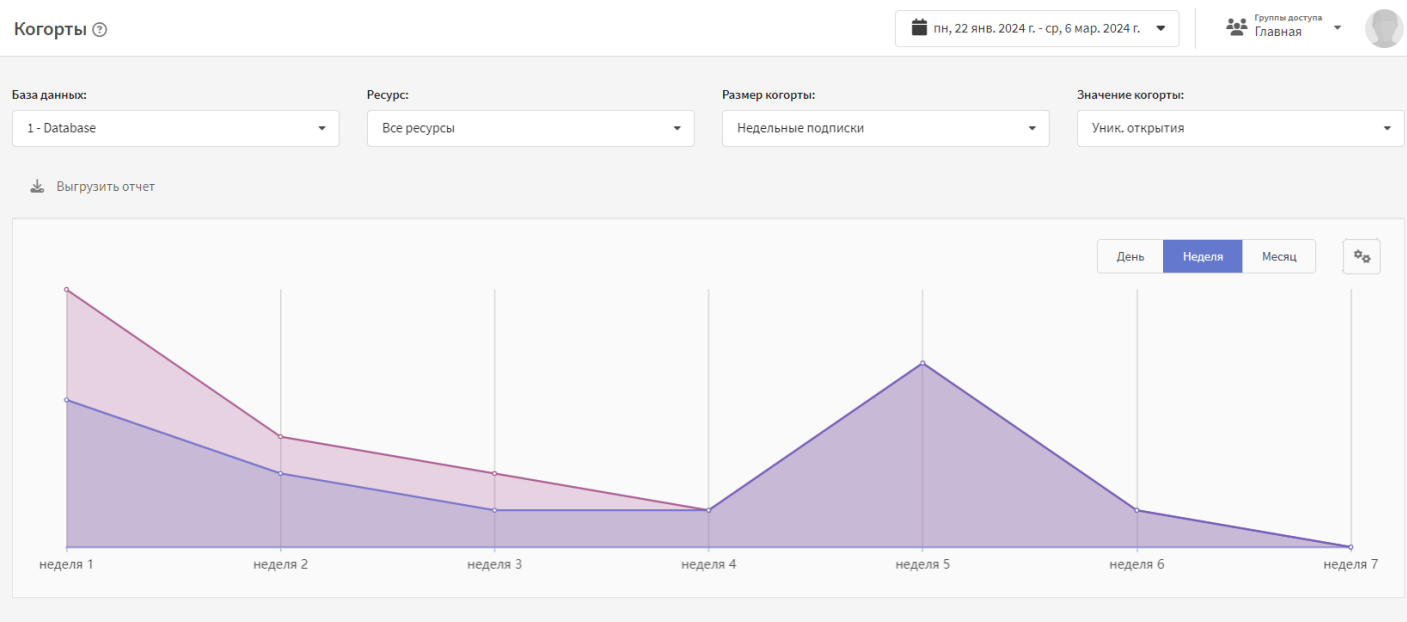
В меню **Значение когорты** выберите показатель, по которому хотите произвести оценку активности когорты:

- количество уникальных кликов;
- количество уникальных открытий;
- отношение уникальных кликов к открытиям;
- отношение уникальных открытий к отправленным сообщениям;
- отношение уникальных кликов к отправленным сообщениям.

> **TIP**
>
> Справа от настроек находится кнопка **Выгрузить отчёт**, с помощью которой вы можете экспортировать данные в формате XLS или CSV.

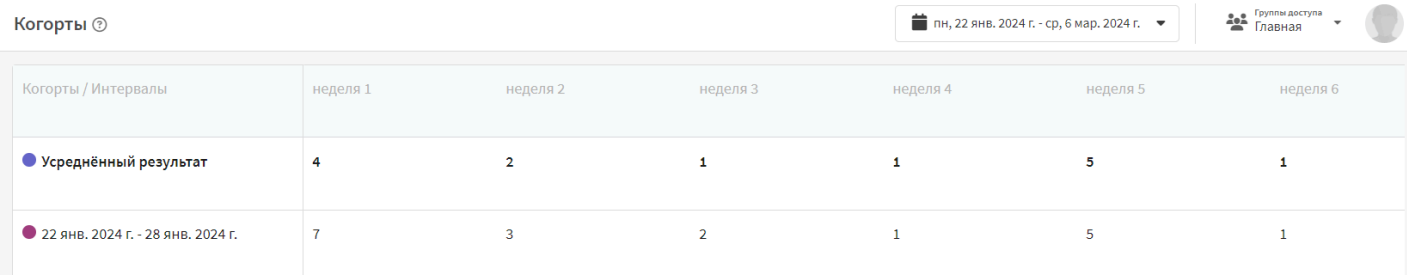
График

На графике наглядно отображается активность когорт подписчиков: доступны режимы визуализации по дням, неделям и месяцам **периода** формирования отчёта. С помощью кнопки настройки графика вы можете убрать или добавить определённые когорты или усреднённый показатель по всем группам клиентов:



Показатели активности

В таблице под графиком отображаются показатели активности по всем группам подписчиков за выбранный **период** (в зависимости от **Значения когорты**), а также усредненный результат.



Когорты / Интервалы	неделя 1	неделя 2	неделя 3	неделя 4	неделя 5	неделя 6
Усреднённый результат	**4**	**2**	**1**	**1**	**5**	**1**
22 янв. 2024 г. - 28 янв. 2024 г.	7	3	2	1	5	1

INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

Время жизни

Время жизни показывает среднее время жизни подписчика в [когорте](). Подписчик считается активным пока кликает на ссылки в сообщениях.

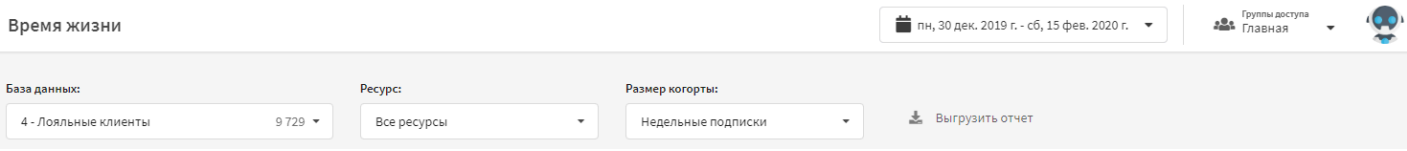
Подсчёт времени происходит с учётом количества активных подписчиков, за каждый день.

- В когорте выбирается максимальное количество кликов за день.
- Затем клики за каждый день в отчётном периоде делятся на это число. Таким образом учитывается вес каждого дня в общей сумме.
- Сумма результатов деления — это время жизни когорты подписчиков (в днях).
- Если количество кликов за какой-то день < 5% от максимума, то он не попадает в сумму.

Настройки отчёта

В верхней части страницы выберите **период** для формирования отчёта. Чтобы узнать время жизни определенной когорты, нужно выбрать **базу данных**, в которую импортировались пофили. Если вам нужно отследить активность по конкретному **ресурсу подписки**, выберите ресурс в выпадающем меню.

Далее выбирается **размер когорты**: новые клиенты за неделю или за месяц.

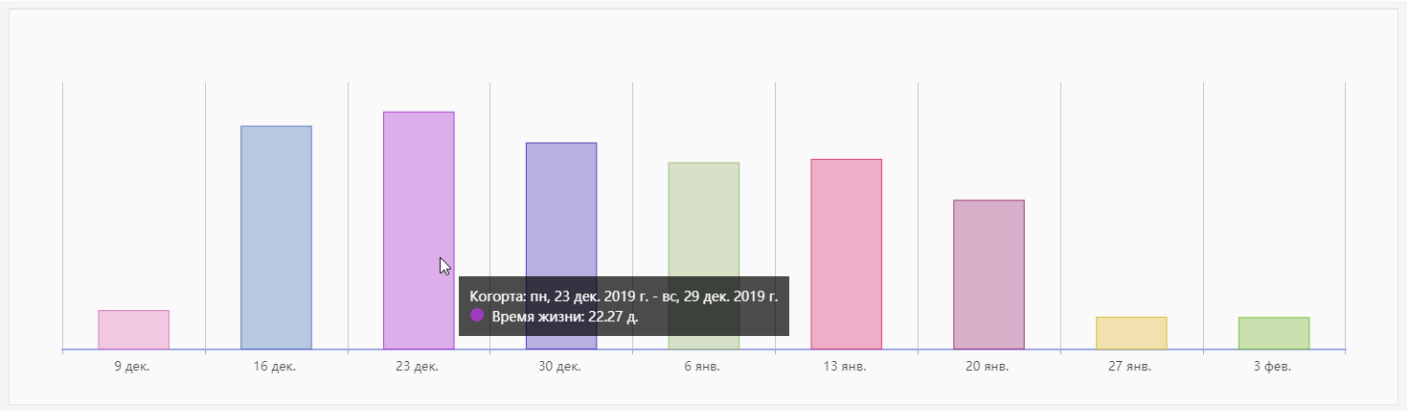


TIP

Справа от настроек находится кнопка Выгрузить отчёт, с помощью которой вы можете экспортировать данные в формате XLSX или CSV.

График когорт по времени жизни

График отображает когорты клиентов по максимальному времени жизни: так вы можете отследить, когда привлечение новых подписчиков оказалось наиболее эффективно в долгосрочной перспективе. При наведении на когорту отображается её среднее время активности:



Динамика активности клиентов

Строки таблицы показывают активность подписчиков когорты относительно максимальной активности в этой когорте — для каждого **дня**, **недели** или **месяца периода** отчёта. Отслеживая динамику активности, вы сможете принять меры по увеличению вовлечённости клиентов:

День | Неделя | Месяц

Когорты / Интервалы	День 1	День 2	День 3	День 4	День 5	День 6
9 дек. 2019 г. - 15 дек. 2019 г.	37.09%	15.30%	8.56%	6.84%	8.46%	6.31%
16 дек. 2019 г. - 22 дек. 2019 г.	58.72%	69.72%	87.16%	51.38%	100.00%	92.66%
23 дек. 2019 г. - 29 дек. 2019 г.	66.39%	82.35%	57.98%	82.35%	85.71%	64.71%
30 дек. 2019 г. - 5 янв. 2020 г.	44.68%	57.45%	79.79%	92.55%	100.00%	40.43%
6 янв. 2020 г. - 12 янв. 2020 г.	36.43%	34.88%	69.77%	100.00%	89.92%	72.87%
13 янв. 2020 г. - 19 янв. 2020 г.	70.63%	67.46%	74.60%	92.06%	100.00%	71.43%
20 янв. 2020 г. - 26 янв. 2020 г.	66.67%	63.25%	98.29%	100.00%	88.89%	68.38%
27 янв. 2020 г. - 2 фев. 2020 г.	100.00%	36.43%	19.94%	20.36%	16.07%	12.88%
3 фев. 2020 г. - 9 фев. 2020 г.	100.00%	23.97%	25.77%	21.91%	26.55%	15.98%

INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

Воронка конверсии

Воронка — наиболее полезный отчёт, если вы хотите отследить конверсию всей вашей маркетинговой стратегии или отдельных рассылок (в широком смысле слова).

Отчёт представляет собой последовательность шагов, по которым проходит группа ваших клиентов. Эта группа определяется первым настроенным шагом воронки. Далее отчёт показывает, какое количество клиентов перешло на следующий шаг, тем самым сужая выборку.

Анализируя динамику прохождения воронки, вы можете оперативно принять меры по повышению конверсии на определённых этапах маркетинговой коммуникации.

Расчёт периода конверсии

Так как отчёт-воронка собирает данные из разных источников, расчёт может в некоторых случаях занять продолжительное время. Чтобы оптимизировать процесс работы с отчётом, периоды дат для него рассчитываются и сохраняются в платформе.

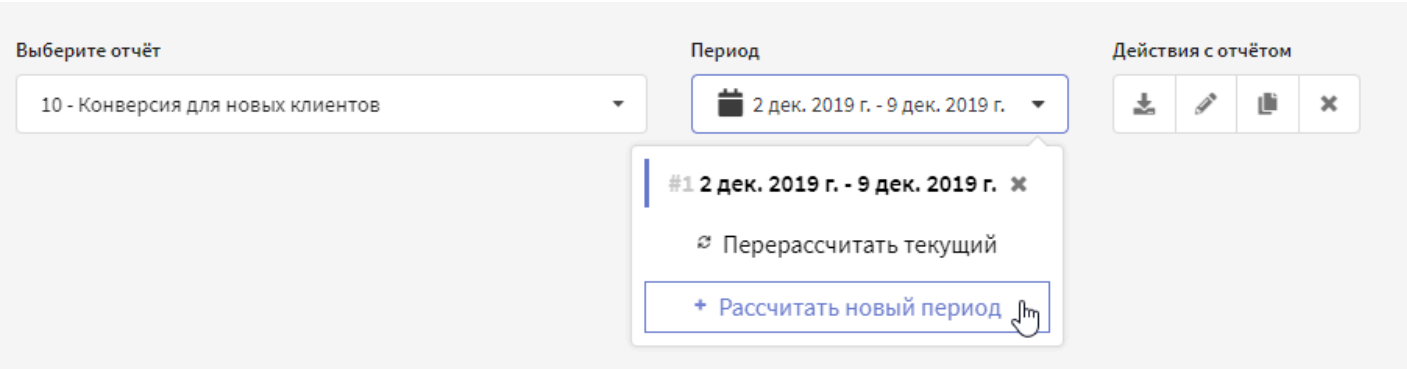
В меню воронок **выберите** готовый **отчёт** и интересующий вас расчётный **период**. Вы также можете добавить новый период отчёта или пересчитать текущий.

Справа находятся следующие кнопки:

Выгрузить отчёт — экспортировать отчёт в csv или xlsx форматах;

Редактировать отчёт — внести изменения в отчёт;

Клонировать отчёт — создать копию отчёта



Создание и настройка отчёта

Чтобы добавить новый отчёт по конверсии, в верхнем правом углу страницы нажмите кнопку **Создать**.

Основные настройки

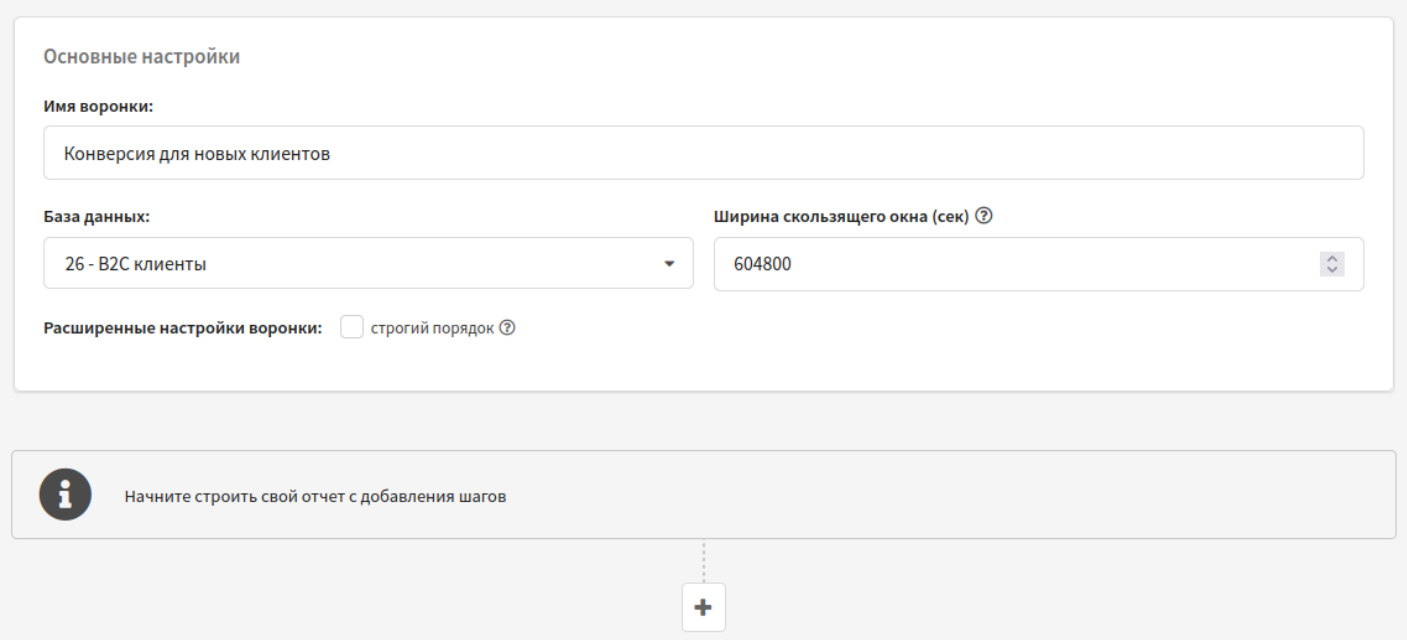
В основных настройках введите название и описание новой воронки, а также выберите **базу данных**, к которой относится интересующая вас аудитория клиентов.

Далее укажите **ширину скользящего окна** — это временный период между первым и последним условиями в воронке. Период указывается в секундах и по умолчанию составляет 604800 секунд (7 дней).

Например, вы оставляете ширину скользящего окна по умочанию (604800 секунд) и добавляете в воронку следующие шаги: `Шаблон #1 доставлен –> Шаблон #1 открыт –> Достижение цели "purchase"`. В таком случае первое условие выполнится, когда сообщение будет доставлено профилю, а последнее — когда профиль своими действиями активирует пиксель. Профиль попадет в последний шаг воронки, только если с момента доставки сообщения до момента активации пикселя пройдет не больше 7 дней. При этом должен быть сохранен порядок действий, заданный в воронке.

TIP

30 дней — 2592000 секунд
14 дней — 1209600 секунд
1 день — 86400 секунд
1 час — 3600 секунд



Дополнительно можно включить **строгий порядок** в воронке. Такая настройка запретит посторонние события в искомой последовательности.

Например, вы строите воронку: `Рассылка #1 доставлена → Рассылка #1 открыта → Достижение цели "purchase"`. Если между вторым и третьим шагом платформа зафиксировала ещё одно событие в истории профиля, профиль не перейдет на третий шаг.

Используйте строгий порядок, если вам важно, чтобы профиль выполнил четкую последовательность действий за короткий временной промежуток.

Построение воронки

Добавьте блоки воронки в последовательности, необходимой для решения поставленных задач:

- **Сегмент** — вхождение клиента в определённую выборку;
- **Действия в шаблоне**;
- **Действия в рассылке**;
- **Событие пикселя** — событие трекинга: посещение или достижение цели;
- **Подписки** — подписка на определённый ресурс.

CAUTION

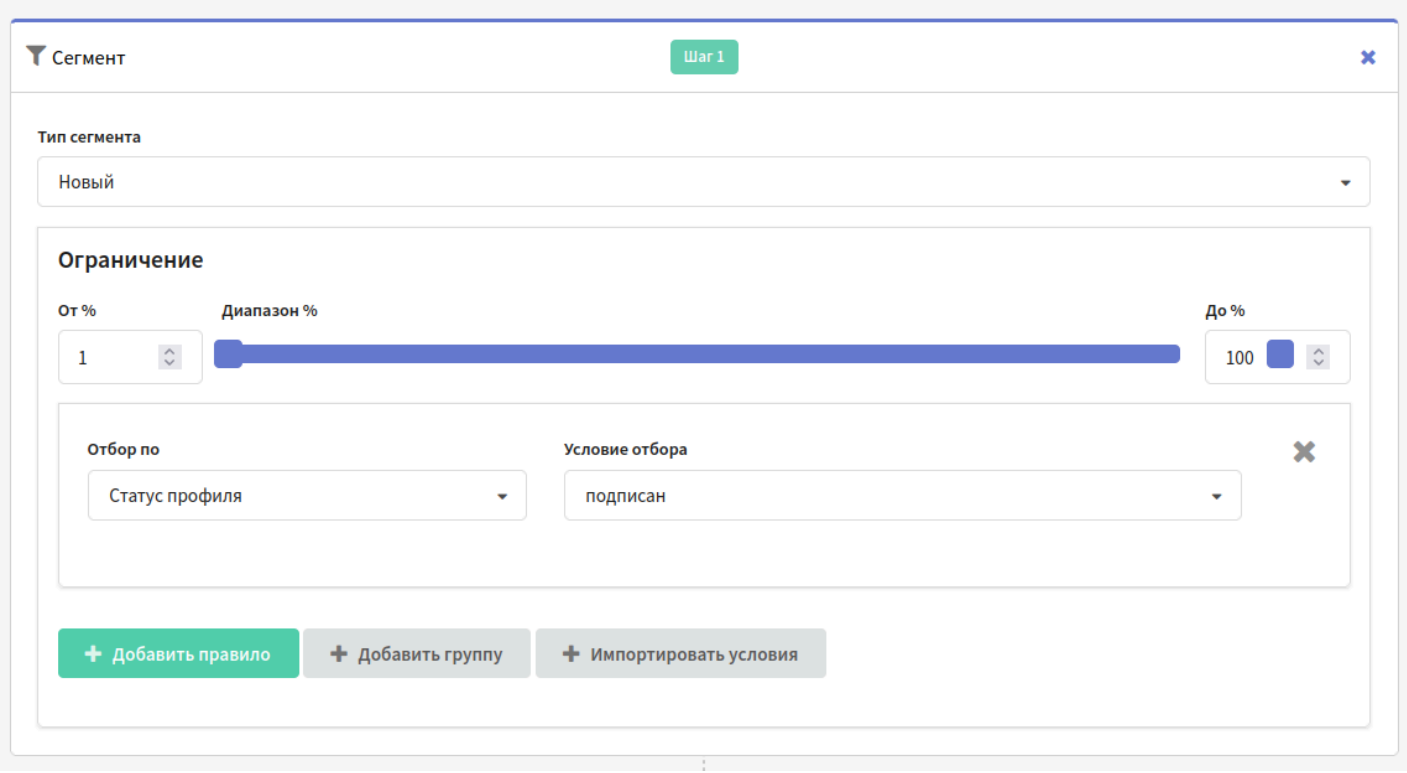
Воронка считает количество профилей, а не количество событий. Например, если платформа зафиксировала два клика профиля в шаблоне, воронка покажет только один профиль на данном шаге.

Сегмент

Сегмент как условие воронки можно добавлять только на первом шаге. Это позволяет сузить наблюдаемую в воронке выборку профилей. Вы можете настроить новый сегмент или использовать уже существующий.

TIP

Если вы используете существующий сегмент, привязанный к нескольким базам, в отчёт попадут профили только из базы, которая указана в основных настройках воронки.



CAUTION

Шаг воронки “сегмент” не учитывает выбранный период, он работает в реальном времени, как фильтр.

Действия в шаблоне

Действия в шаблоне позволяют зафиксировать профили, совершившие заданные действия в выбранном шаблоне. Выберите канал и действия (отправлено, доставлено, открыто, клики), которые должен совершить профиль, чтобы перейти на данный этап воронки.

TIP

Каждый тип действия добавляет новый шаг воронки.

CAUTION

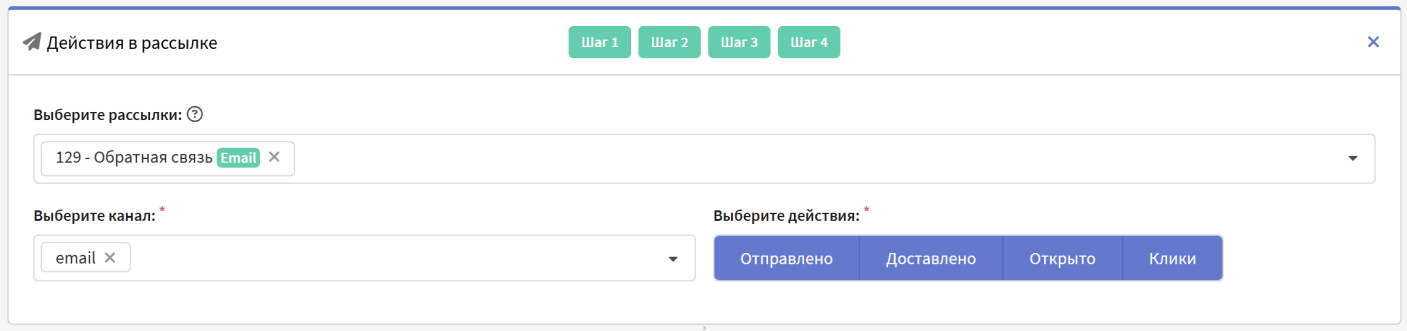
Если в SMS-канале было зарегистрировано событие клика, но при этом отсутствует событие доставки, то событие доставки сгенерируется автоматически — на 100 миллисекунд раньше, чем событие клика.

Действия в рассылке

Данный шаг позволяет осуществить подсчет профилей, совершивших определенные действия в рассылке по выбранному каналу связи.

TIP

Каждый тип действия добавляет новый шаг воронки.



CAUTION

Если в SMS-канале было зарегистрировано событие клика, но при этом отсутствует событие доставки, то событие доставки сгенерируется автоматически — на 100 миллисекунд раньше, чем событие клика.

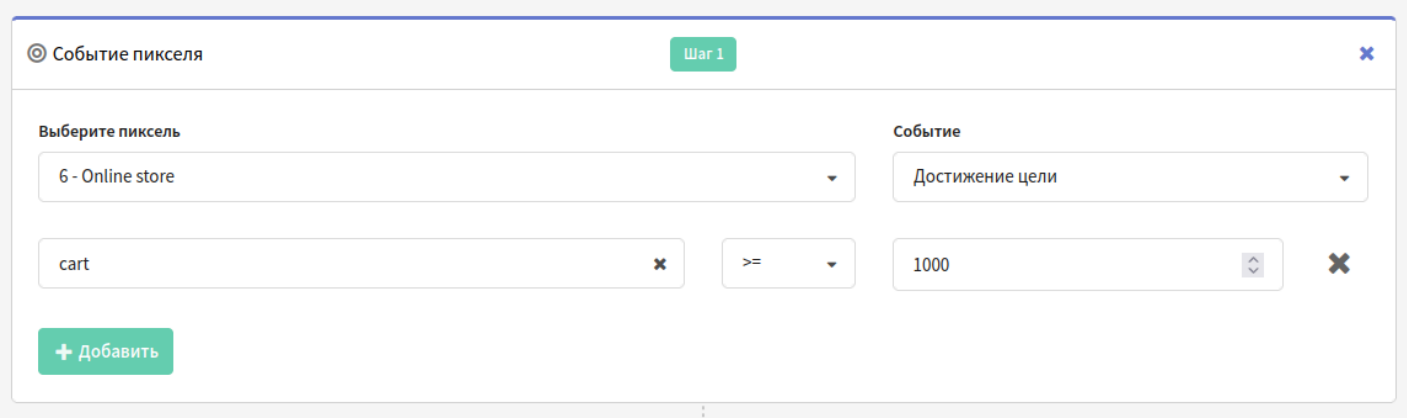
Событие пикселя

Информация о посещении сайта или выполнении определённого действия. Выберите **пиксель**, по которому будут отобраны события. Это могут быть:

- **Посещение пикселя** — любое событие выбранного пикселя.
- **Достижение цели** — событие достижения клиентом одной или нескольких **целей**.

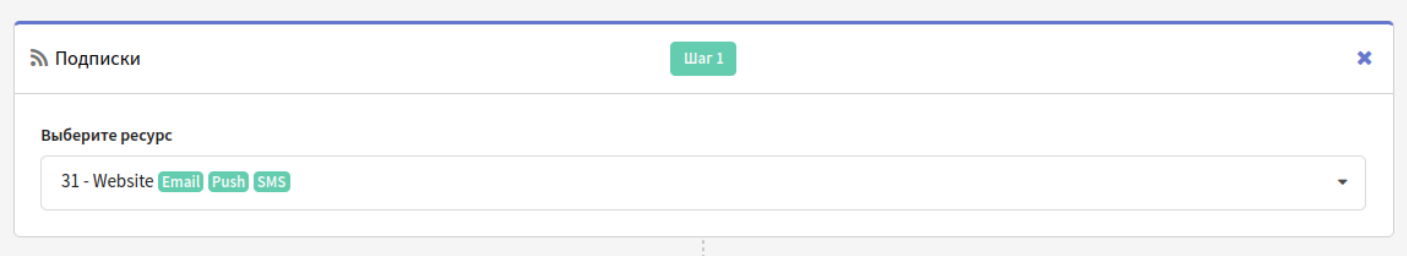
Дополнительно для цели можно указать значение пикселя, которое должно быть зафиксировано у профиля для попадания на этот шаг воронки. Доступны операторы сравнения:

- больше или равно
- меньше или равно
- больше
- меньше
- равно



Подписки

Этот шаг фиксирует профили тех, кто подписался на любой канал в выбранном ресурсе. Информация по подпискам даёт сведения о количестве подписавшихся.



INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

Цели

Цели — это отчёт о выполнении вашими клиентами целевых действий и о ценности их совершения. Цели могут регистрироваться профилям клиентов с использованием нескольких инструментов Altcraft:

- Скрипты пикселей отслеживания, размещённые на страницах сайта и других ресурсах
- Узел **Регистрация цели** в Сценарии (цепочке)
- API регистрации целей

> **TIP**
>
> Отчёт автоматически складывает **ценность действий** с положительными и отрицательными значениями и выводит в виде показателя “**Доход**”.

Настройка отчета

В верхней части страницы выберите **период** для формирования отчёта.

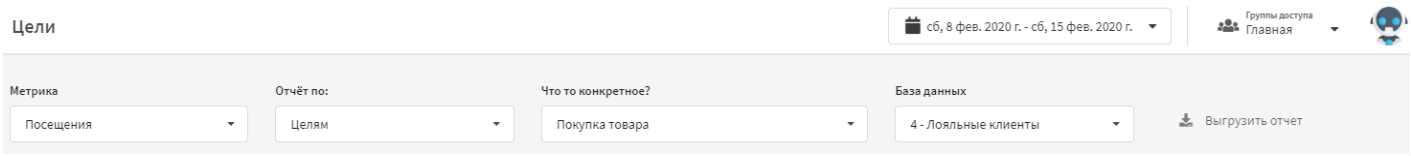
В отчёте доступно две основных метрики:

- **Посещения** — количество совершения целевых действий;
- **Доход** — сумма ценности совершённых действий.

Также выберите необходимый режим отчёта:

- **Отчёт по пикселям** — аналитика по пикселям, для которых регистрируются цели;
- **Отчёт по целям** — аналитика по целям, зарегистрированным на разные пиксели.

В зависимости от выбора, в поле “**Что-то конкретное?**” выберите пиксель или цель для анализа. При выборе конкретного пикселя или цели вы также сможете уточнить Базу профилей — для известных клиентов:



> **TIP**
>
> Выгрузить получившийся отчёт в формате MS Excel и CSV можно, нажав кнопку **“Выгрузить отчёт”.**

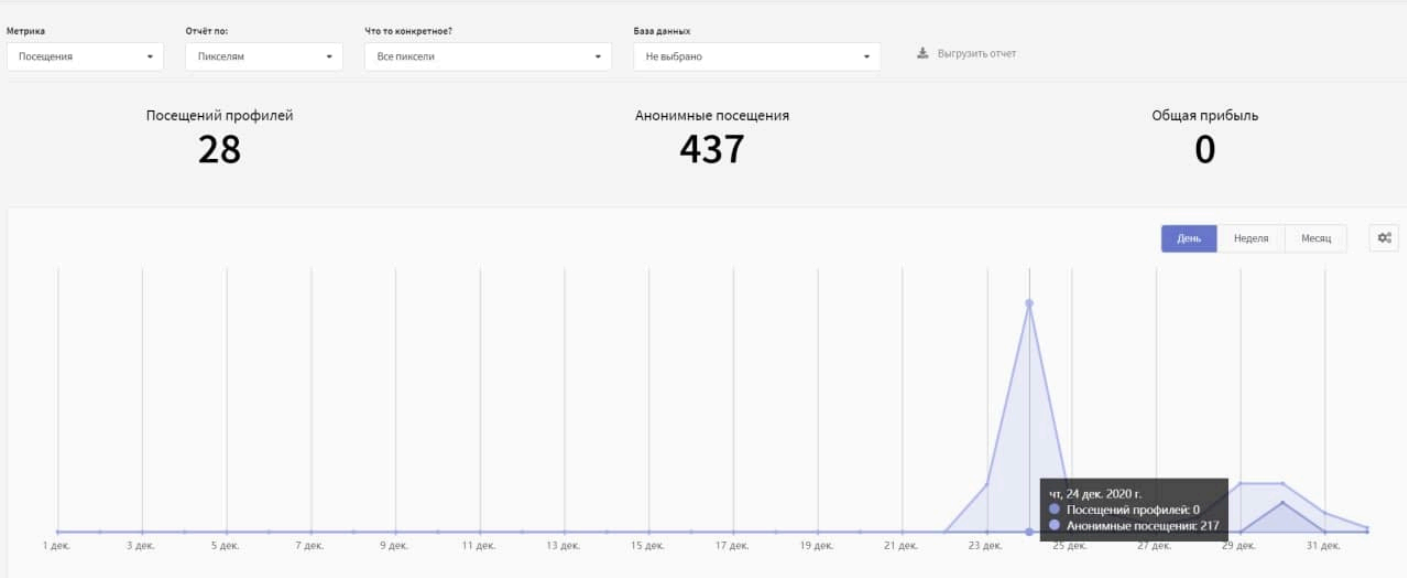
Динамика прибыли и достижения целей

График отчёта о целях показывает количество **посещений** известных **профилей** и количество **анонимных посещений**, а также общую прибыль от выполненных действий.

> **TIP**

Цели могут регистрироваться для клиентов, чьих контактных и личных данных данных у вас ещё нет. Такие события называются **анонимными**. Друг от друга Altcraft отличает их по данным о соединении (cookie). Когда клиент после этого регистрируется из того же браузера, его данные записываются в уже существующий профиль.

Если вы выберите метрику **“Посещения”,** то вы получите информацию о количестве посещений тех профилей, что есть в ваших базах данных, и об анонимных посещениях.



Если вы выберете такую метрику как **“Доход”**, то вы получите информацию об **общем доходе, общих затратах** на привлечение клиентов и **общей прибыли**. **Общая прибыль** — это разность между общим доходом и затратами.

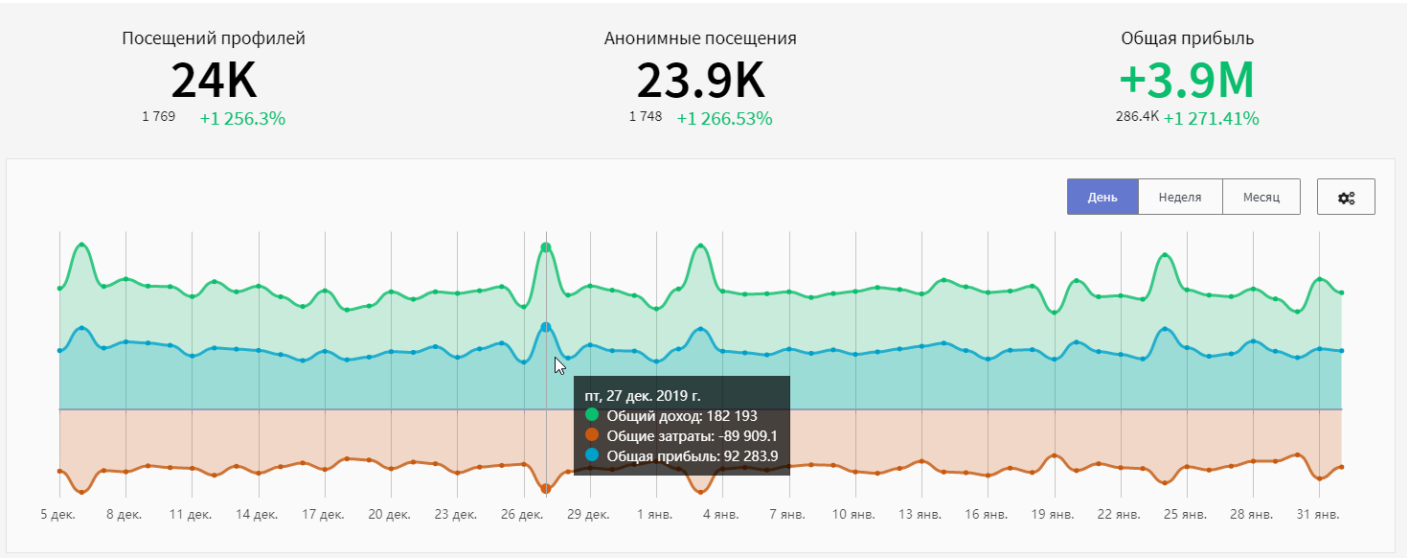


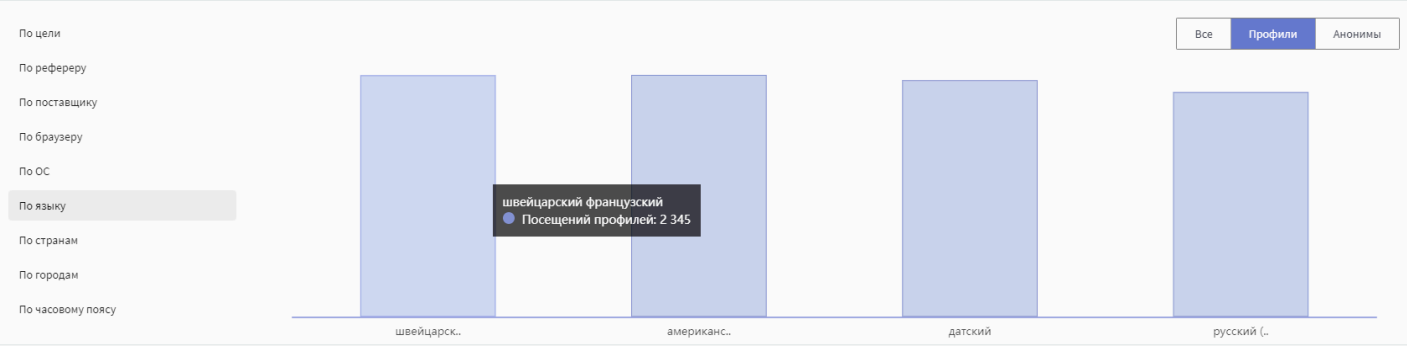
График строится по **дням**, **неделям** или **месяцам периода** отчёта. С помощью кнопки

вы можете убрать ненужные показатели с графика или, наоборот, добавить.

Аналитические срезы отчёта

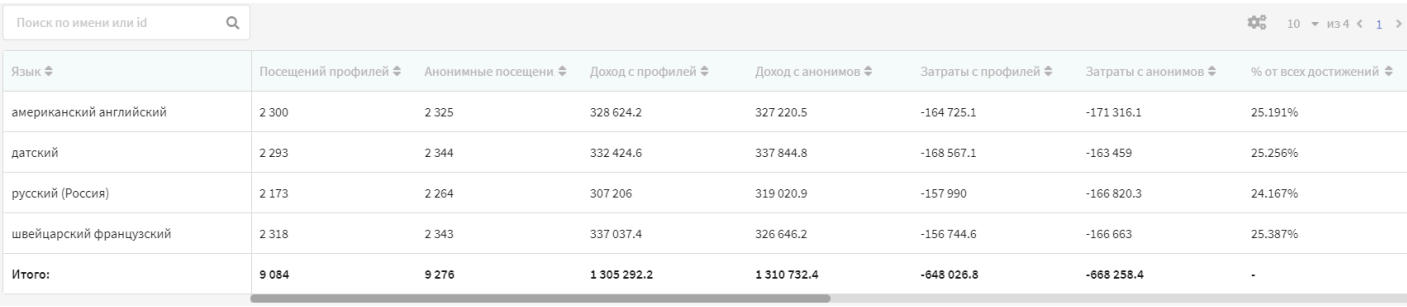
По умолчанию отчёт по пикселям сгруппирован по наиболее популярным целям, и наоборот. Помимо этого, доступен ряд других срезов:

При регистрации целевого действия Altcraft собирает информацию об устройстве, которым пользовался клиент, а также о сетевом соединении и рефererе. На основе этих данных вы можете дополнительно сгруппировать получившуюся аудиторию по геолокации, рефереру, а также по браузеру или операционной системе:



Показатели отчёта

Таблица в нижней части страницы содержит сводные данные отчёта в зависимости от выбранных вами настроек, срезов и показателей. Именно эти данные будут выгружены при экспорте. С помощью кнопки настройки можно добавить колонки в таблицу или скрыть ненужные:



Язык	Посещений профилей	Анонимные посещени	Доход с профилей	Доход с анонимов	Затраты с профилей	Затраты с анонимов	% от всех достижений
американский английский	2 300	2 325	328 624.2	327 220.5	-164 725.1	-171 316.1	25.191%
датский	2 293	2 344	332 424.6	337 844.8	-168 567.1	-163 459	25.256%
русский (Россия)	2 173	2 264	307 206	319 020.9	-157 990	-166 820.3	24.167%
швейцарский французский	2 318	2 343	337 037.4	326 646.2	-156 744.6	-166 663	25.387%
Итого:	**9 084**	**9 276**	**1 305 292.2**	**1 310 732.4**	**-648 026.8**	**-668 258.4**	-

INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

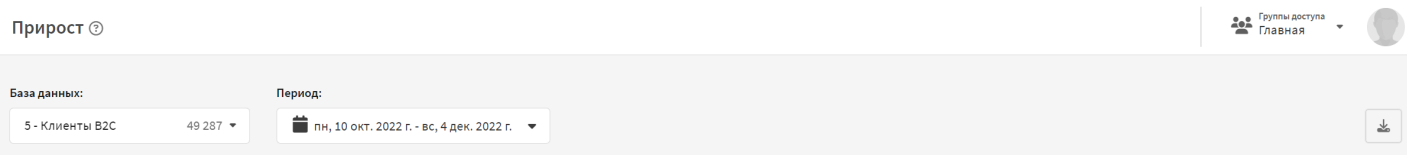
Прирост аудитории

Отчёт по приросту позволяет отслеживать динамику роста аудитории профилей в вашей базе данных. Этот отчёт фиксирует только события добавления новых профилей подписчиков, не учитывая отписки или удаления.

Настройка отчёта

В верхней части страницы выберите **период** для формирования отчёта.

По умолчанию отчёт формируется для всех доступных **баз профилей**. Вы можете использовать выпадающее меню, чтобы выбрать конкретную базу клиентов:

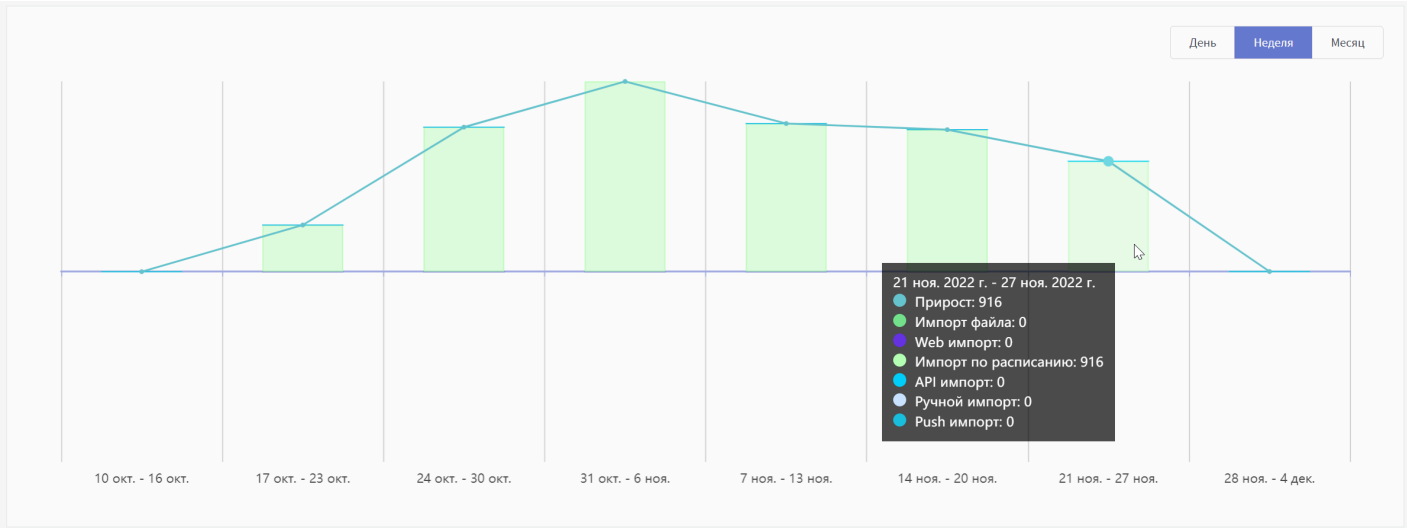


TIP

С помощью кнопки **“Выгрузить отчёт”** вы можете экспортировать данные по приросту в формате CSV или XLS.

График прироста

Визуальная часть отчёта отображает динамику роста аудитории. Данные могут быть представлены в виде диаграмм, сгруппированных по дням, неделям или месяцам выбранного периода.



Показатели динамики аудитории

Прирост аудитории представлен следующими метриками **импорта**:

- **Прирост** — чистое увеличение количества профилей в базе (без вычета отписок или других факторов, уменьшающих количество профилей).

- **Импорт файла** — массовое добавление профилей из табличного файла.
- **Web импорт** — создание новых профилей при подписке через форму или попап на веб-сайте.
- **Импорт по расписанию** — создание профилей на основе данных из внешнего SQL хранилища по расписанию.
- **API импорт** — создание новых профилей через API.
- **Ручной импорт** — создание профилей через интерфейс платформы.
- **Push импорт** — создание анонимных и персональных профилей при подписке на **push-канал** (например, при подписке на получение уведомлений в браузере).

Дата	Прирост	Импорт файла	Web импорт	Импорт по расписани	API импорт	Ручной импорт
28 ноя. 2022 г. - 4 дек. 2022 г.	+1	0	1	0	0	0
21 ноя. 2022 г. - 27 ноя. 2022 г.	+916	0	0	916	0	0
14 ноя. 2022 г. - 20 ноя. 2022 г.	+1 178	0	0	1 178	0	0
7 ноя. 2022 г. - 13 ноя. 2022 г.	+1 229	0	0	1 229	0	0
31 окт. 2022 г. - 6 ноя. 2022 г.	+1 578	0	0	1 578	0	0
24 окт. 2022 г. - 30 окт. 2022 г.	+1 199	0	0	1 199	0	0
17 окт. 2022 г. - 23 окт. 2022 г.	+387	0	0	387	0	0
10 окт. 2022 г. - 16 окт. 2022 г.	0	0	0	0	0	0
Итого:	6 488	0	1	6 487	0	0

INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчётов

TIP

Отчёт по приросту отображает увеличение количества профилей в базе данных. Если вам нужно отслеживать изменения в количестве подписок на ресурс, используйте **Отчёт по трафику** с применением фильтра по ресурсу.

Карта кликов (Email)

Карта кликов — это отчет по переходам ваших клиентов по ссылкам в отправленном Email-сообщении. Карта кликов отображает только **уникальные** клики по всем ссылкам Email-сообщения, включая ссылки отписки. Она помогает определить качество контента шаблона и выявить наиболее удачное месторасположение и оформление ссылок.

> **INFO**
>
> Карта кликов работает только для **Email-сообщений**. Для других каналов отправки (SMS, Push) эта функция не поддерживается.

> **TIP**
>
> Необходимо иметь в виду, что отчет строится именно по **шаблону**, не по **рассылкам**. Если шаблон использовался в нескольких рассылках, то данные будут суммироваться.

Особенности сбора статистики

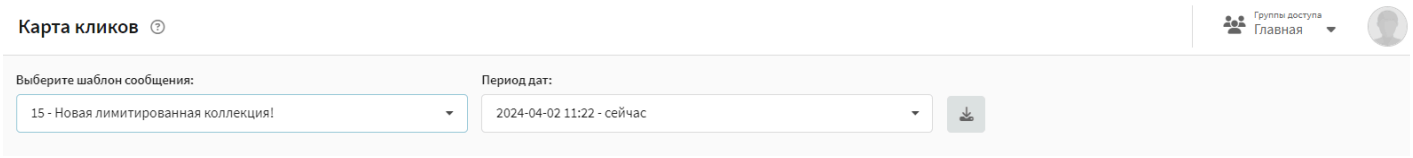
При анализе данных карты кликов учитывайте:

- **Общее количество уникальных кликов** (отображается вверху) — это уникальные клики по всему сообщению за выбранный период
- **Клики по отдельным ссылкам** (в таблице) — уникальные клики по каждой конкретной ссылке
- **Уникальность кликов** определяется в течение 24 часов — повторные клики от одного пользователя в течение суток не учитываются
- **Учитываются все клики**, включая переходы по ссылкам отписки, которые могли быть выполнены через кнопки почтового клиента
- **Динамические ссылки** — для персонализированных URL статистика собирается по конечному домену, а не по исходному шаблону ссылки

> **CAUTION**
>
> Цифры в общей статистике и детальной таблице могут различаться по нескольким причинам: разные периоды подсчета уникальности, учет всех типов кликов (включая отписки по ссылке), особенности обработки динамических ссылок и обработки данных платформой.

Составление карты кликов

Для начала выберите **шаблон сообщения** для сбора статистики. Если вы изменяли сообщение — например, для разных рассылок — выберите необходимую версию в выпадающем меню **Период дат**:



TIP

Экспортировать получившийся отчёт в формате MS Excel и CSV можно, нажав кнопку **“Выгрузить отчёт”.**

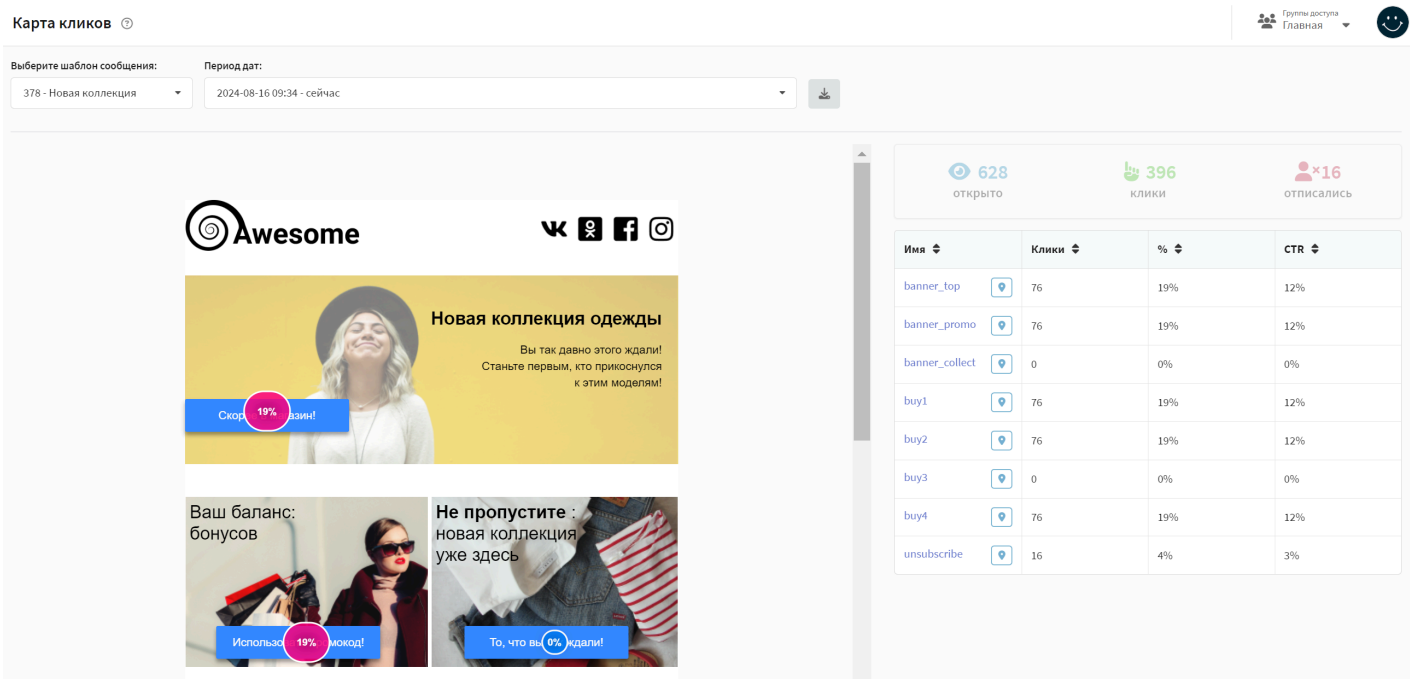
Тепловая карта и диаграмма переходов

При выборе шаблона и версии будет составлена тепловая карта, отмечающая процентные соотношения кликов по ссылкам в письме.

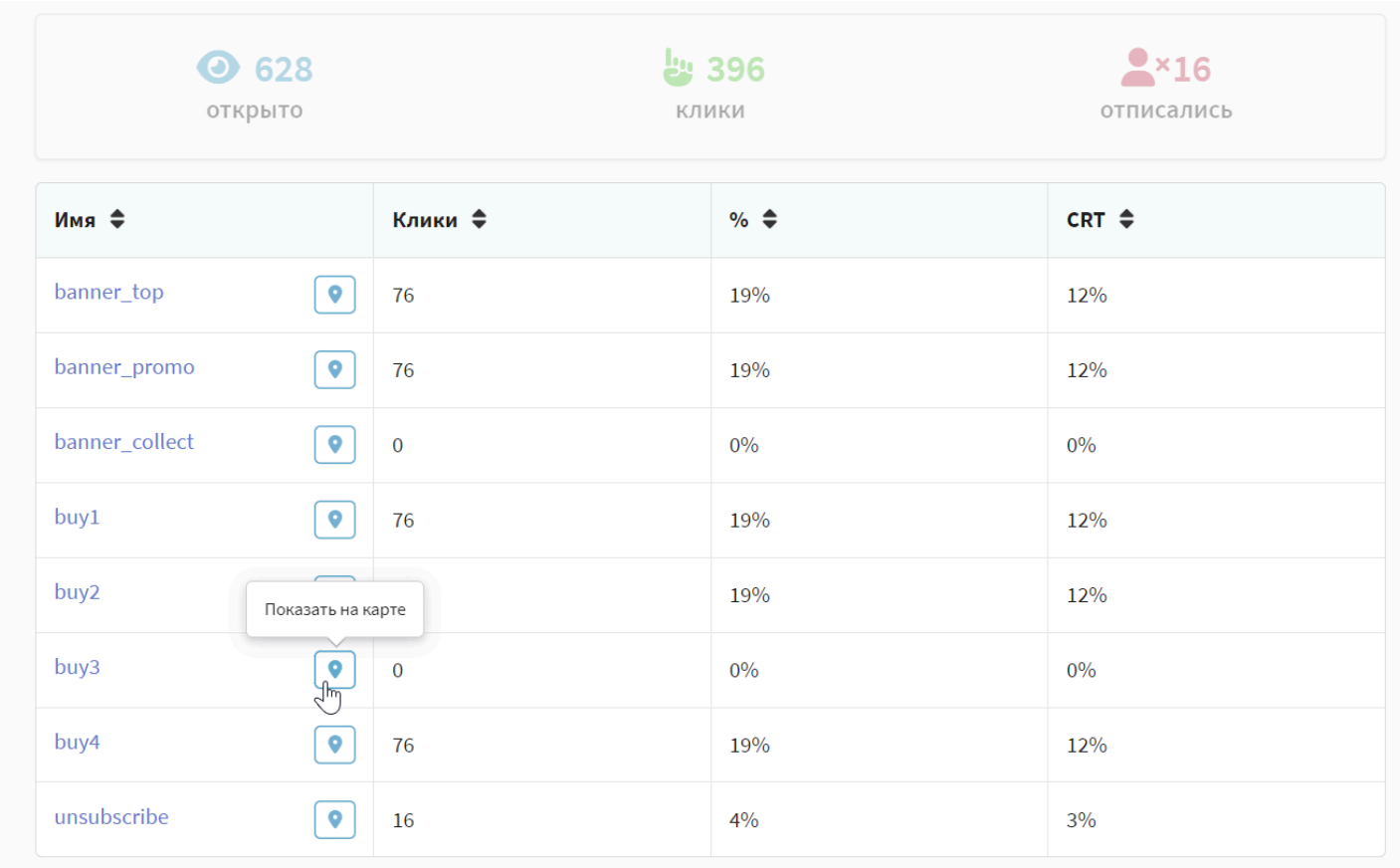
Справа отображается дополнительная информация:

- Количество уникальных открытий шаблона
- Общее количество уникальных кликов
- Количество отписавшихся клиентов

Ниже представлена таблица с количеством кликов, переходов по определённой ссылке, процентом от общего количества кликов и **CTR** для каждой отдельной ссылки сообщения.



Рядом с каждой ссылкой находится кнопка с геопозицией. При нажатии на нее вы переноситесь в ту область письма, где находится ссылка.



Это делает поиск ссылок более удобным, особенно если в шаблоне их много или вы анализируете большое письмо, по которому “разбросаны” ссылки.

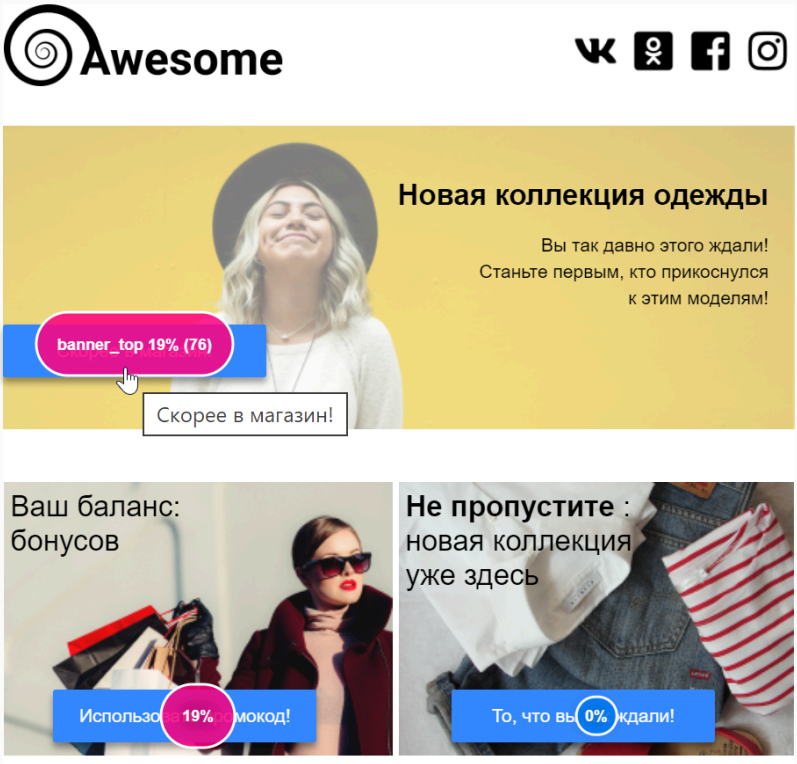
Карта кликов

Выберите шаблон сообщения:

378 - Новая коллекция

Период дат:

2024-08-16 09:34 - сейчас



INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

Отчет по программам лояльности

Отчет по программе лояльности отображает ключевые метрики программы лояльности — использование баллов, транзакции, покупки и активность участников. Такой отчет помогает оценить вовлечённость клиентов и эффективность программы.

Чтобы открыть отчёт, в главном меню выберите **Аналитика** → **Лояльность**.

Основные показатели отчета

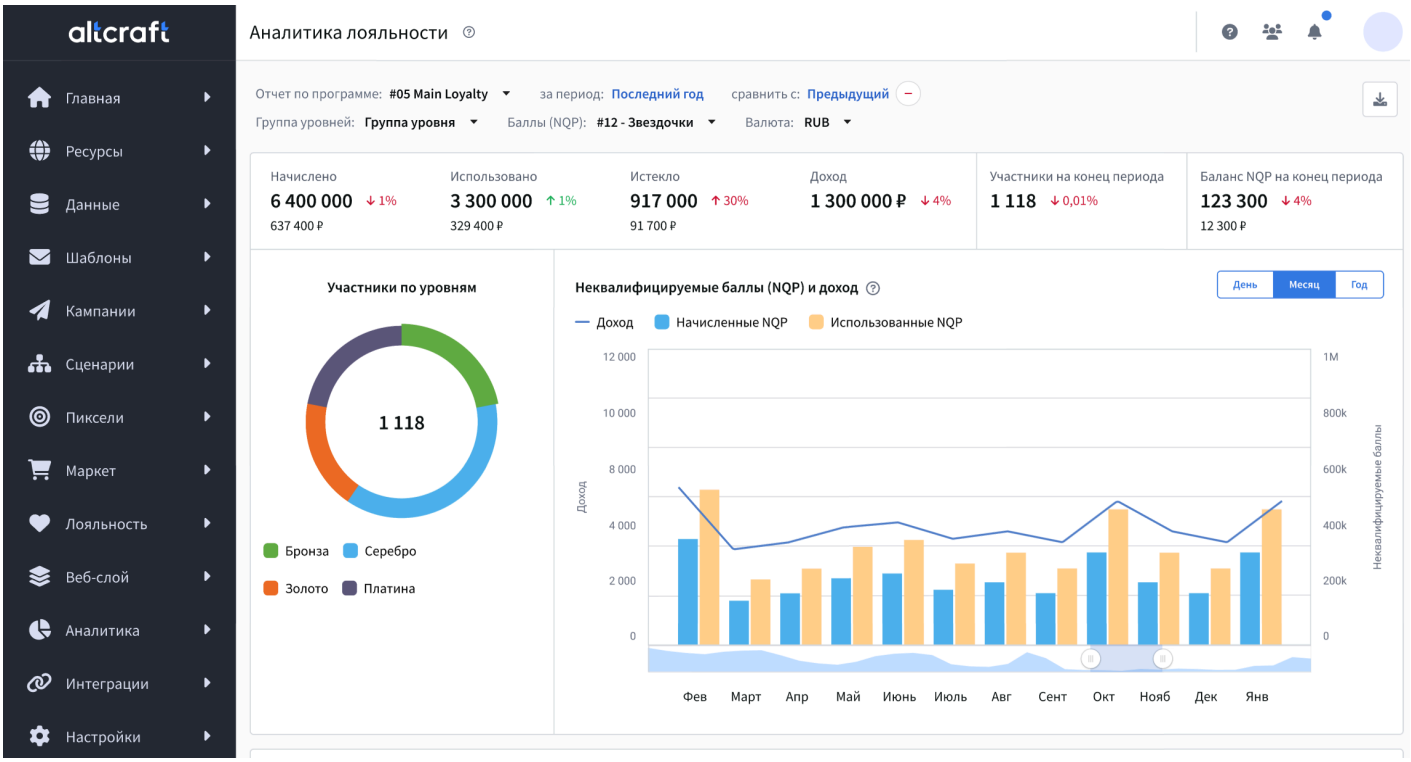
> **TIP: Что такое NQP**
>
> Одна из главных метрик, используемых в аналитике — **баллы NQP**. NQP — это баллы, не участвующие в расчете лояльности, но влияющие на активность участников. Иными словами, это бонусные баллы, которые участник накапливает и тратит.

Укажите программу лояльности, период, за который будет отображаться статистика, и период для сравнения. Также можно выбрать группу уровней лояльности, тип начисляемых баллов и валюту.

Основные показатели отчета:

- **Начислено** — количество NQP, начисленных участникам за период;
- **Использовано** — количество NQP, потраченных участниками за период;
- **Истекло** — количество “сгоревших” NQP за период;
- **Доход** — денежная сумма заказов, которые были сделаны участниками лояльности за период;
- **Участники на конец периода** — количество участников программы лояльности (пересчитывается в конце суток);
- **Баланс NQP на конец периода** — общее количество NQP, имеющихся у участников программы лояльности (пересчитывается в конце суток).

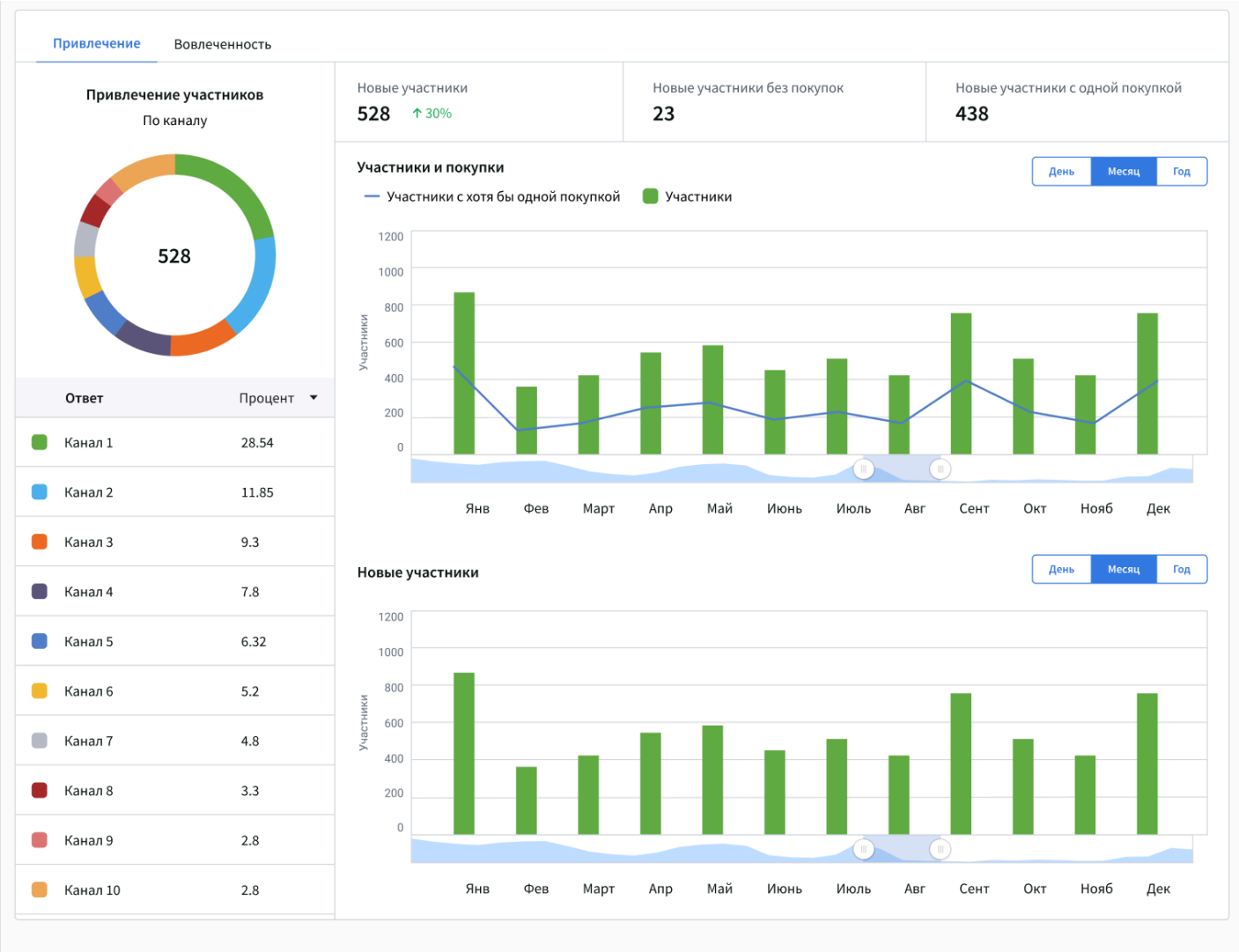
Под показателями лояльности находятся диаграммы: распределение участников **по уровням** и график **дохода, начисленных** и **использованных NQP**:



Привлечение

В этом разделе находятся данные по привлечению новых участников в программу лояльности:

- Диаграмма по каналам привлечения;
- Число новых участников, а также новых участников без покупок и как минимум с одной покупкой;
- Диаграмма по новым участникам.



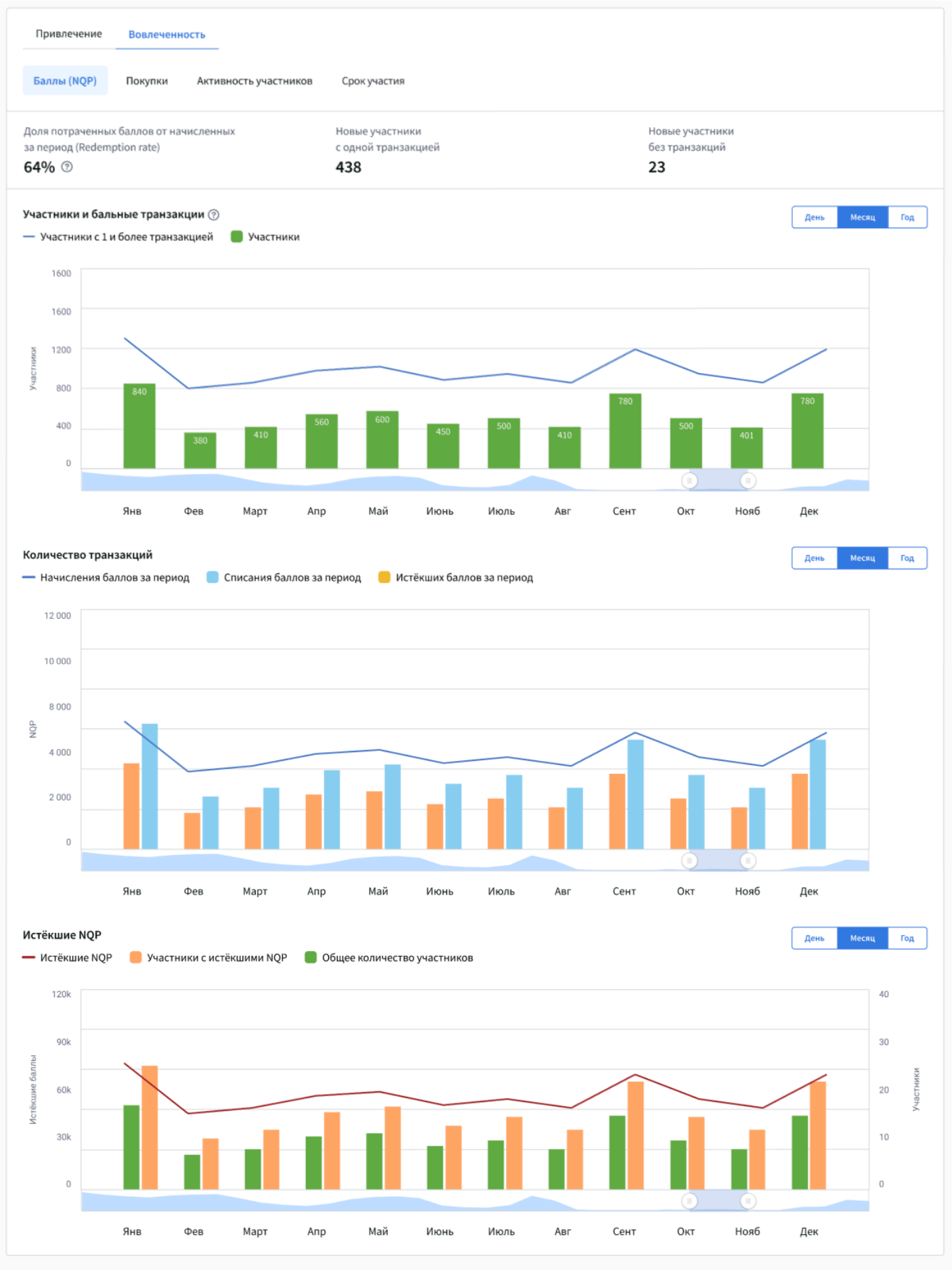
Вовлеченность

Информация в этом разделе показывает вовлеченность участников программы на основе их активности. Содержит вкладки **Баллы**, **Покупки**, **Активность участников** и **Срок участия**.

Баллы (NQP)

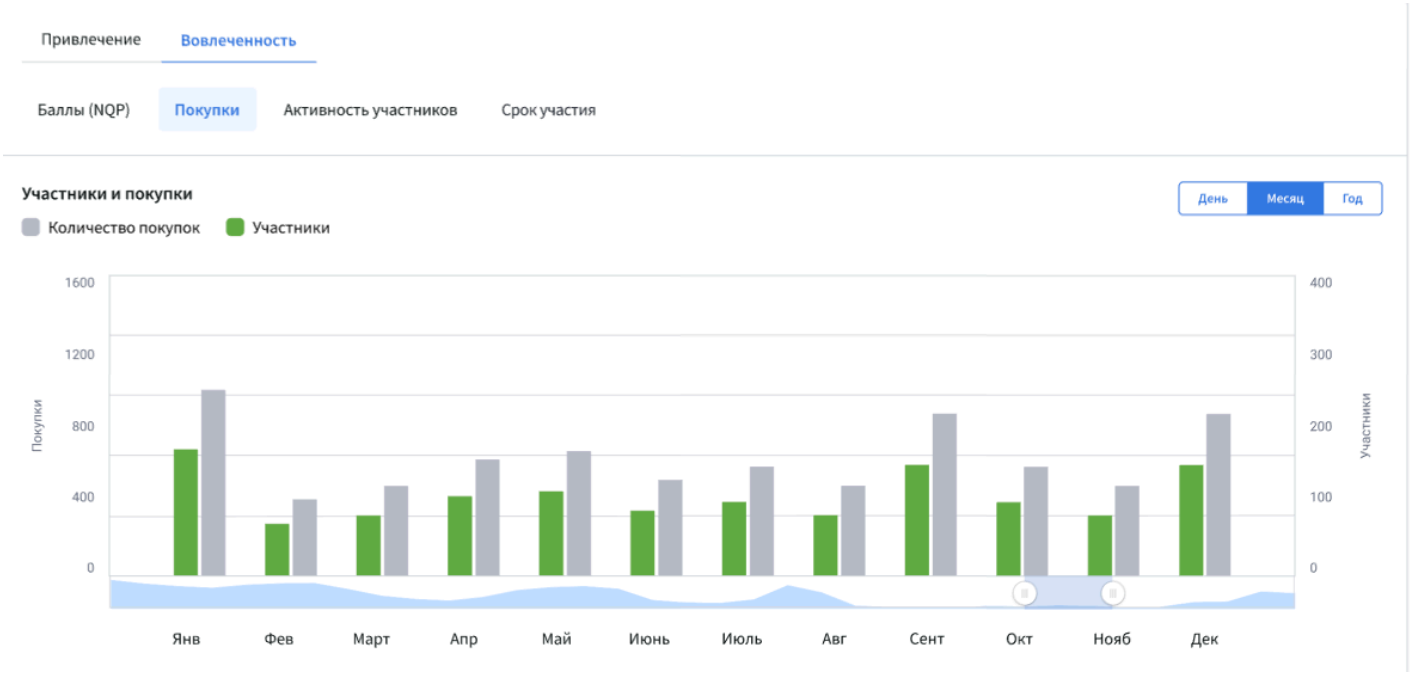
Здесь представлены следующие данные:

- Информация по доле потраченных баллов от начисленных (за период), по новым участникам с транзакциями и без;
- Графики по участникам программы и их транзакциям;
- График по истекшим NQP.



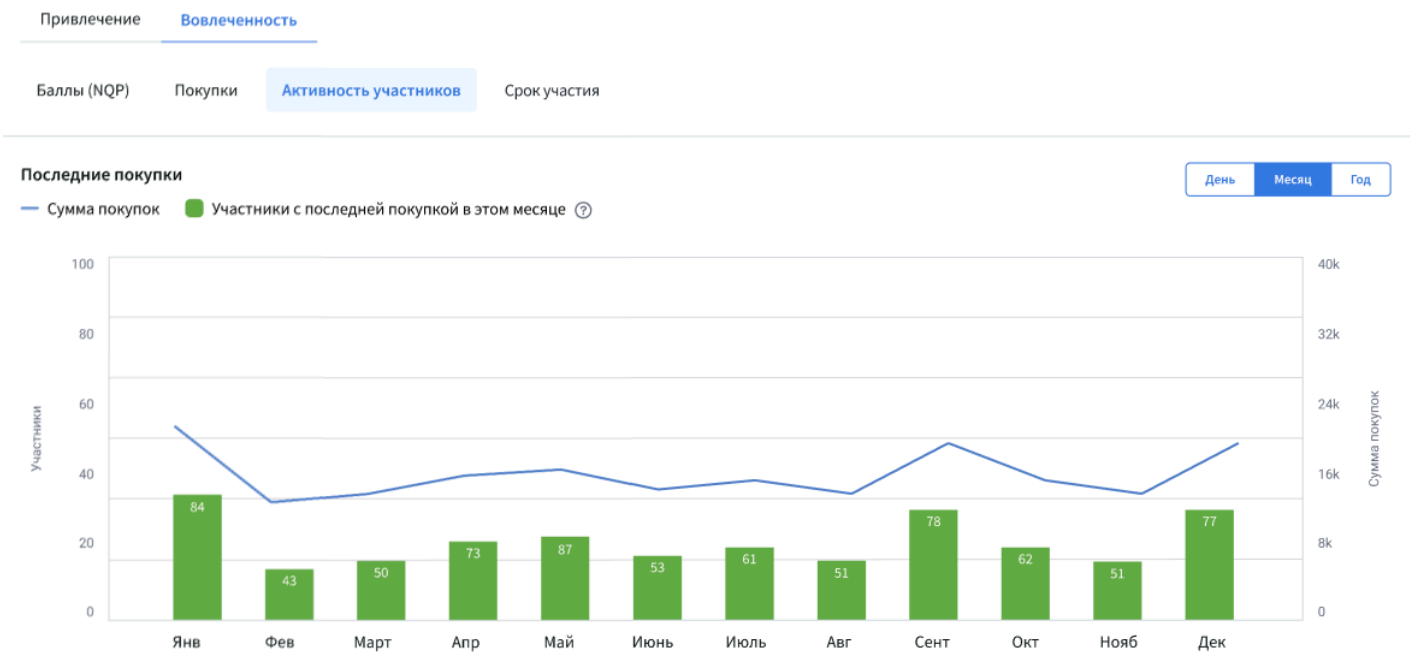
Покупки

В этом разделе отображаются данные по покупкам участников программы лояльности:



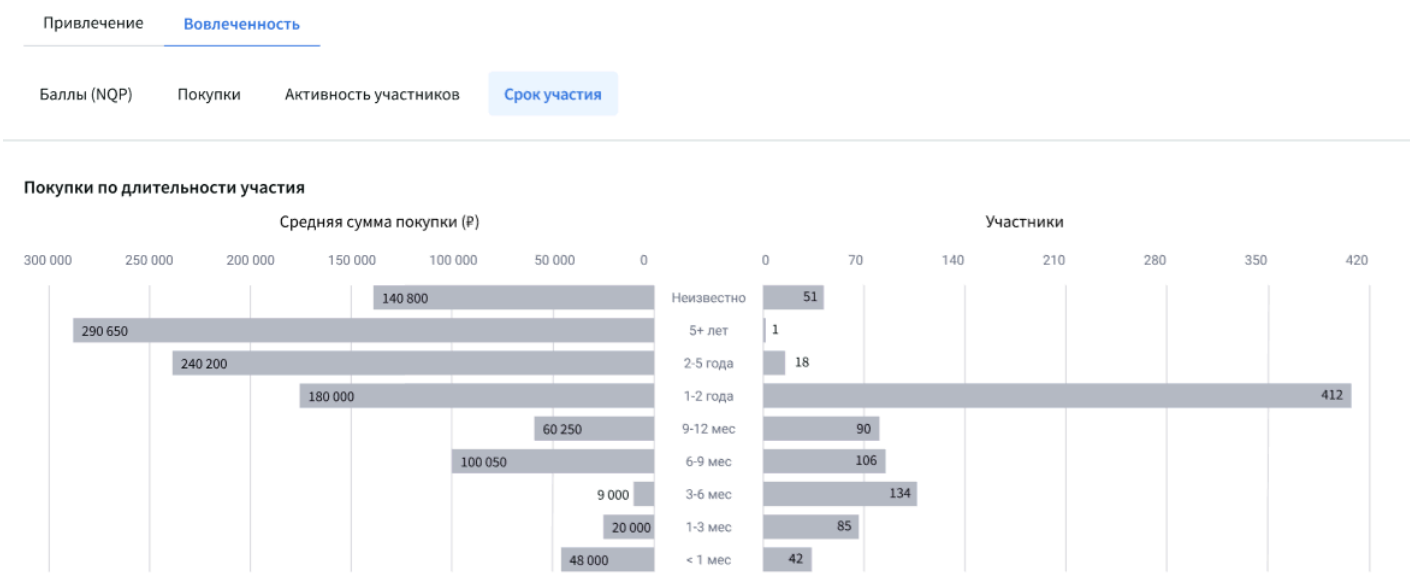
Активность участников

Здесь находится график, отображающий сумму покупок по месяцам и участников с последней покупкой в соответствующем месяце:



Срок участия

В разделе содержится гистограмма с информацией по участникам программы, их средней сумме покупки и длительностью участия в программе лояльности:



Экспорт отчета

Чтобы выгрузить отчет, нажмите на соответствующую иконку в верхнем правом углу экрана. Назовите отчет и нажмите **Выгрузить**:

Будет создано задание на экспорт отчета по программе лояльности. Скачать его можно в разделе **Задачи**.

Отчёт о возвратах

Hard и soft bounce

В тех случаях, когда email-письмо не может быть доставлено получателю, почтовый сервер отправляет обратно сообщение об ошибке. Существует два вида подобных ошибок: **Hard bounce** и **Soft bounce**.

Hard bounce — ошибки серверного класса (5xx). Такие ошибки возникают в тех случаях, когда отправка на почтовый сервер невозможна, например, указанного email-адреса не существует. Когда сервер возвращает Hard bounce ошибку, платформа меняет статус профиля на **“Hard bounced”** и добавляет email в глобальный стоп-список. Пока адрес находится в этом списке, отправить на него письмо не получится. Убрать адрес из глобального стоп-списка можно только в Панели администратора в разделе **“Supression list”**.

> **CAUTION**
>
> Почтовые адреса с Hard bounce ошибками попадают в стоп-список, находящийся **в Панели администратора**. Не следует путать его со стоп-списком в панели пользователя, который по умолчанию называется “Global suppress list”.

Soft bounce — ошибки клиента отправки (4xx). В отличие от Hard bounce, Soft bounce не влияет на статусы подписок клиентов. В случае получения Soft bounce платформа пробует повторно отправить сообщение по **правилам повторной отправки**. Если доставить сообщение не получится, платформа зафиксирует событие недоставки.

Причинами Soft bounce могут быть:

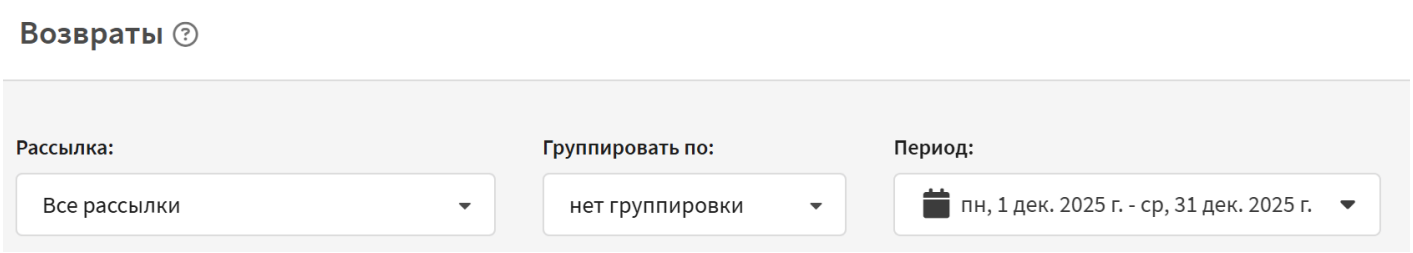
- Переполненный ящик получателя не может принять письмо
- Вложения слишком тяжелые для сервера получателя
- Временная недоступность сервиса
- Карантин сообщения в связи с жалобами на другие сообщения рассылки

Отчёт о возвратах позволит вам отследить количество Email сообщений, которые не доставились получателям с первого раза (сообщения с ошибками Soft bounce).

Настройка отчёта

В верхней части страницы выберите **период** для формирования отчёта.

Отчёт предоставляет информацию о всех временных ошибках отправки для всех **рассылок** за период или для одной конкретной. Дополнительно, можно использовать группировку по **from домену** или **IP-адресу** сендера, с которого производилась отправка:



TIP

Экспортировать получившийся отчёт в формате MS Excel и CSV можно, нажав кнопку **“Выгрузить отчёт”.**

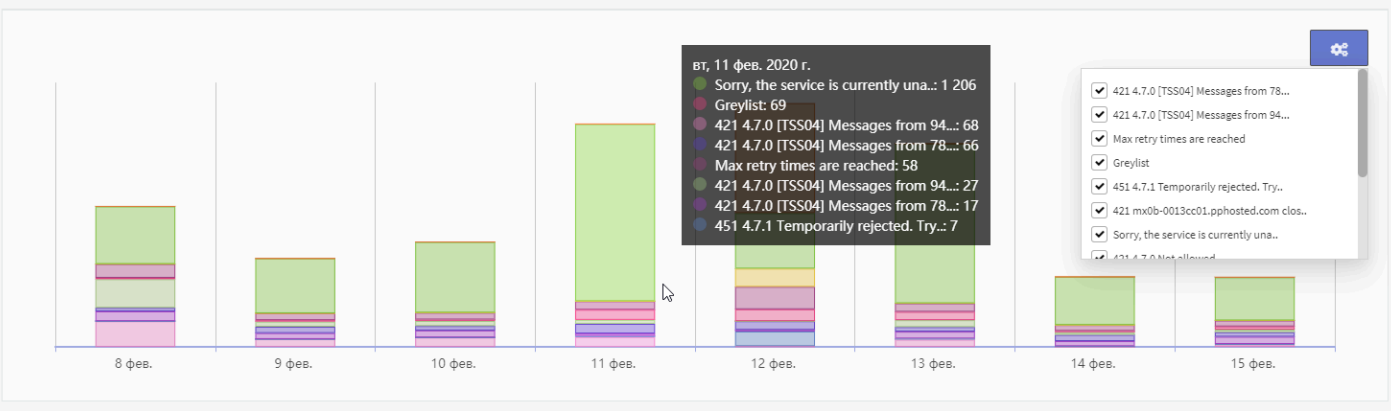
График и статистика

Отчёт сортрует данные по количеству ошибок с одинаковым ответом от провайдеров Email. На основе текста ошибок также работают правила повторной отправки.

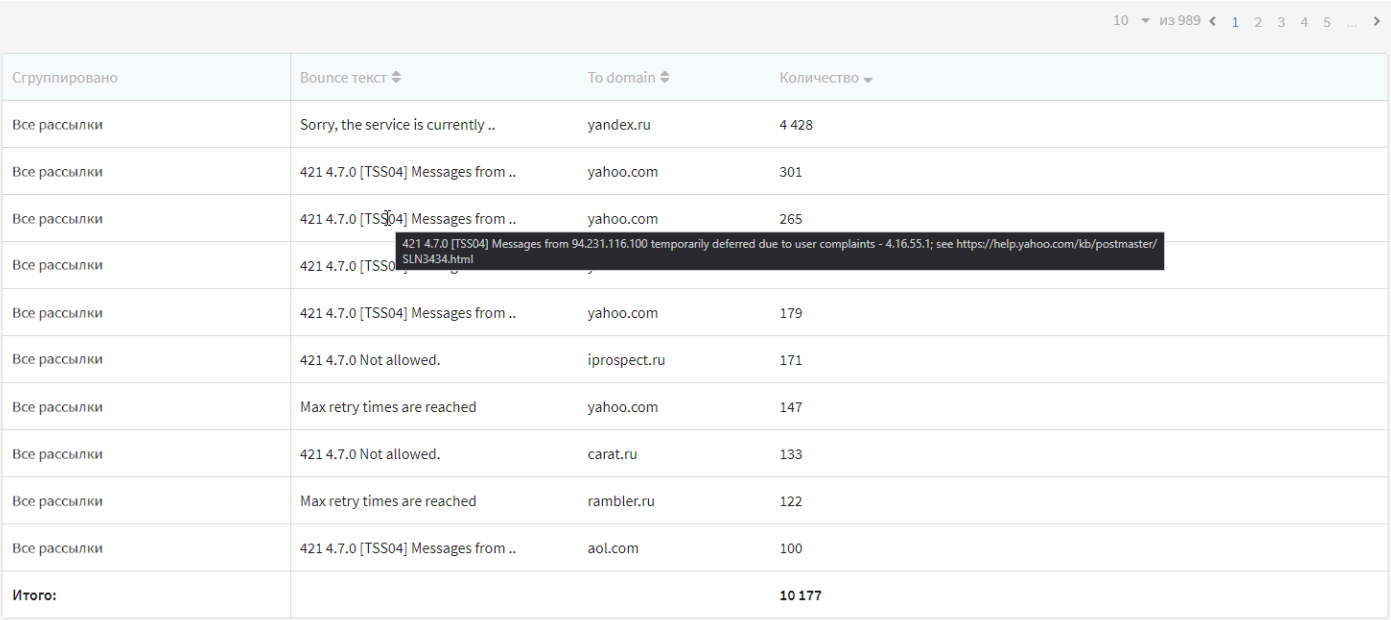
CAUTION

Одно сообщение для одного получателя может быть возвращено несколько раз: каждый раз будет регистрироваться новый возврат. Даже если в итоге письмо доставлено, каждый возврат будет отображён в отчёте. Так можно диагностировать и устранить проблемы с доставляемостью, если из-за большого количества временных ошибок письмо не доставляется в принципе.

На графике представлены наиболее частые типы возвратов за выбранный период. С помощью кнопки **настройки** вы можете убрать с графика тот или иной тип ошибки:



В таблице под графиком вы можете дополнительно отсортировать данные, а также просмотреть полный текст ошибки: для этого наведите на часть, которая отображается в отчёте:



Сгруппировано	Bounce текст	To domain	Количество
Все рассылки	Sorry, the service is currently ..	yandex.ru	4 428
Все рассылки	421 4.7.0 [TSS04] Messages from ..	yahoo.com	301
Все рассылки	421 4.7.0 [TSS04] Messages from ..	yahoo.com	265
Все рассылки	421 4.7.0 [TSS0		
Все рассылки	421 4.7.0 [TSS04] Messages from ..	yahoo.com	179
Все рассылки	421 4.7.0 Not allowed.	iprospect.ru	171
Все рассылки	Max retry times are reached	yahoo.com	147
Все рассылки	421 4.7.0 Not allowed.	carat.ru	133
Все рассылки	Max retry times are reached	rambler.ru	122
Все рассылки	421 4.7.0 [TSS04] Messages from ..	aol.com	100
Итого:			**10 177**

Чтобы ознакомится с полным текстом ошибки, наведите курсор на Bounce-текст. Список распространённых причин soft bounce и их описание представлены в таблице:

Bounce-текст	Описание
`421 Message source blocked temporarily. One or more sender authentication rules matched.`	Сообщения от данного отправителя временно заблокированы из-за несоответствия правилам email-аутентификации. Это может произойти, если отправитель не прошел проверку SPF, DKIM или DMARC. Необходимо проверить настройки аутентификации для From-домена.
`421 Service Temporarily Unavailable`	Сервер временно недоступен. Это может быть связано с перегрузкой или техническими работами.
`451 Please try again later`	Временная проблема на стороне сервера, связанная с перегрузкой или техническими работами.
`451 <<EMAIL>>: Recipient address rejected: Intentional policy rejection, please try again later`	Email-адрес получателя был отклонен на основе политики сервера. Это может происходить из-за неправильных настроек на стороне получателя или из-за того, что отправитель был ранее помечен как спамер.
`450 <<EMAIL>>: Sender address rejected: Greylisted for 120 seconds! Please, see <URL>`	Адрес отправителя временно отклонен из-за попадания в greylist (серый список). Сервер временно отказывается принимать сообщения от отправителя и ожидает, что отправитель повторит попытку через указанное время. Платформа автоматически повторит попытку через 120 секунд. Если проблема не исчезнет после повторной попытки, стоит проверить настройки отправителя.

INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

Отчёт о недоставках

Отчёт по недоставкам предоставляет подробную информацию по **Email, SMS**, **Push**, **Telegram Bot, Viber™, WhatsApp*** сообщениям, которые не были доставлены получателям по итогам одной или нескольких рассылок. Отчёт поможет выявить и устранить причины недостаточной доставляемости рассылок.

Наиболее частые причины недоставки:

Канал	Причины недоставки
Email	Достигнуто максимальное количество **попыток переотправки** сообщения. Сообщение не успело отправиться и было удалено на сендере (время жизни настраиваемое). Сообщение было удалено из очереди пользователем. Сообщение было отправлено на несуществующий ящик (**hard bounce**).
SMS	Изучите документацию вашего провайдера (причины могут отличаться)
Push	Невалидный push-токен в подписке на ресурс (токен устарел)
Telegram Bot	Невалидный chat_id
Viber™	Изучите документацию вашего провайдера (причины могут отличаться)
WhatsApp*	Изучите документацию вашего провайдера (причины могут отличаться)
Notify	Изучите документацию вашего провайдера (причины могут отличаться)

В некоторых случаях, причину недоставки определить невозможно. Такие недоставки могут происходить по неизвестным ошибкам сети. Во всех отчетах подобные ошибки маркируются **“undefined”**.

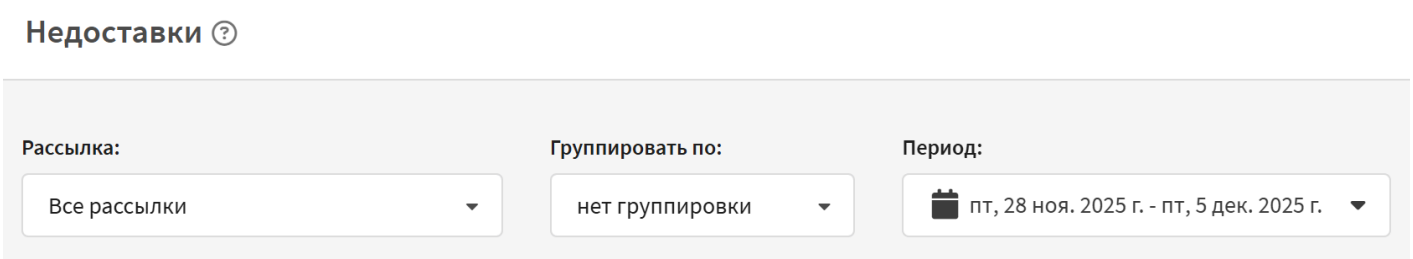
INFO

Платформа Altcraft допускает погрешность 0,02% при составлении отчетов

Настройка отчёта

В верхней части страницы выберите **период** для формирования отчёта.

В отчете предлагается группировка по одной или по всем **Рассылкам**. Дополнительно можно использовать группировку по **from домену** или **IP-адресу**, с которого производилась отправка:

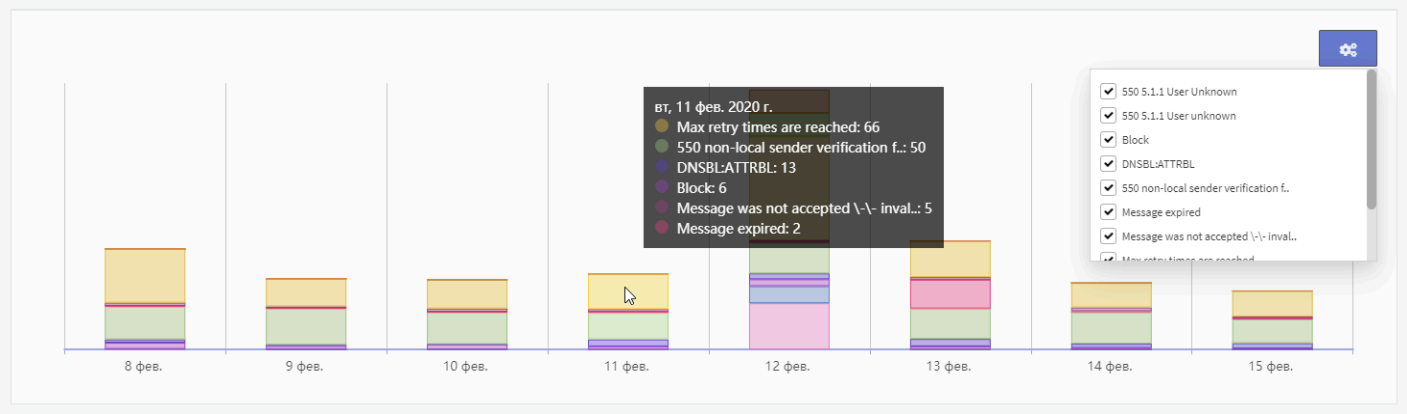


TIP

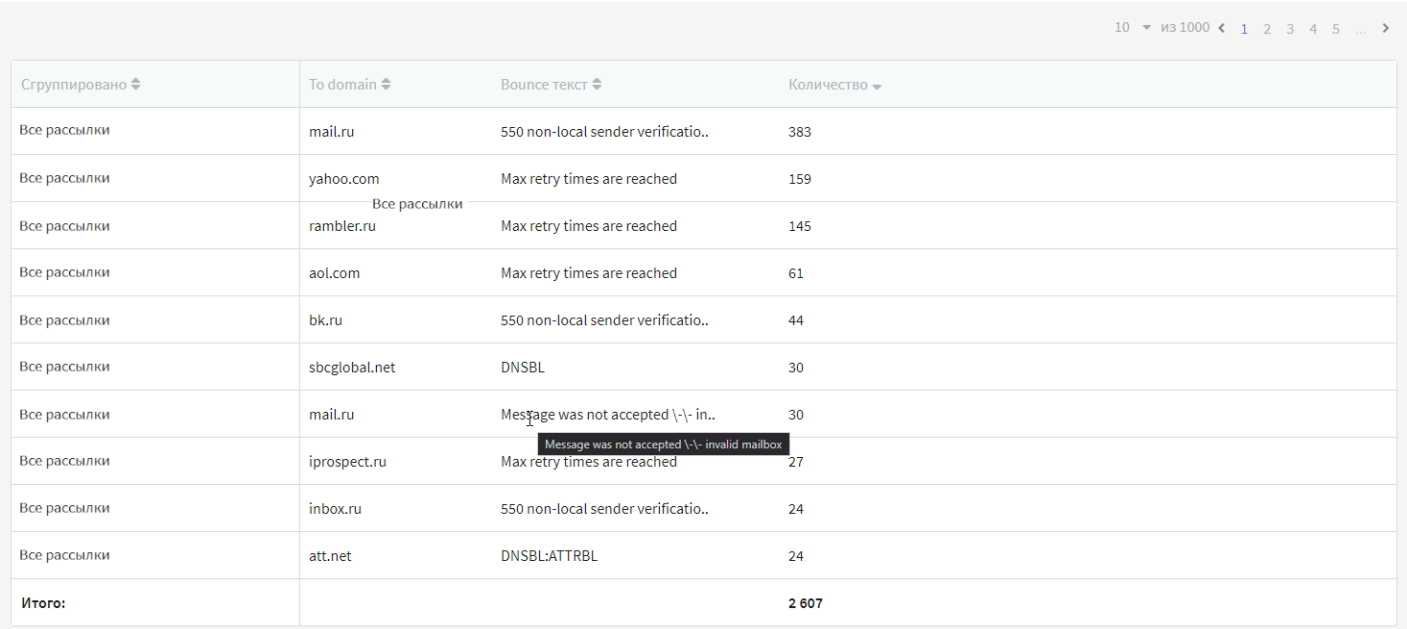
Экспортировать получившийся отчёт в формате MS Excel и CSV можно, нажав кнопку **“Выгрузить отчёт”.**

График и статистика

На графике представлены наиболее частые типы возвратов за выбранный период. С помощью кнопки **настройки** вы можете убрать с графика тот или иной тип ошибки:



В таблице под графиком вы можете дополнительно отсортировать данные, а также просмотреть полный текст ошибки — для этого наведите на часть, которая отображается в отчёте:



Чтобы ознакомится с полным текстом ошибки, наведите курсор на Bounce-текст. Список распространённых ошибок доставки и их описание представлены в таблице:

Bounce-текст	Описание
`500 dial tcp <IP4:PORT>-><IP4:PORT>: i/o timeout`	Сетевая проблема, не удалось подключиться к принимающему серверу
`500 lookup <DOMAIN> on <IP4:PORT>: no such host for domain:<DOMAIN>`	Домен не существует.

Bounce-текст	Описание
`540 <<EMAIL>>: recipient address rejected: Inactive`	Адрес получателя существует, но он был деактивирован (например, отключен или временно приостановлен). Такой адрес может быть снова активирован позже.
`550 relay not permitted`	Неправильно настроена пересылка на почтовом ящике получателя
`550 No such user!`	Email-адрес не существует на сервере (либо аккаунт был удален, либо никогда не существовал).
`550 Unrouteable address`	Не маршрутизируемый адрес (например, если на домене получателя нет A/PTR записей).
`550 Message was not accepted -- invalid mailbox. Local mailbox <EMAIL> is unavailable: user is terminated`	Почтовый ящик пользователя удалён.
`550 Message was not accepted -- invalid mailbox. Local mailbox <EMAIL> is unavailable: account is disabled`	Почтовый ящик пользователя отключён.
`552 5.2.2 <<EMAIL>>: user is over quota`	Почтовый ящик получателя переполнен.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Отчет по глобальным контрольным группам

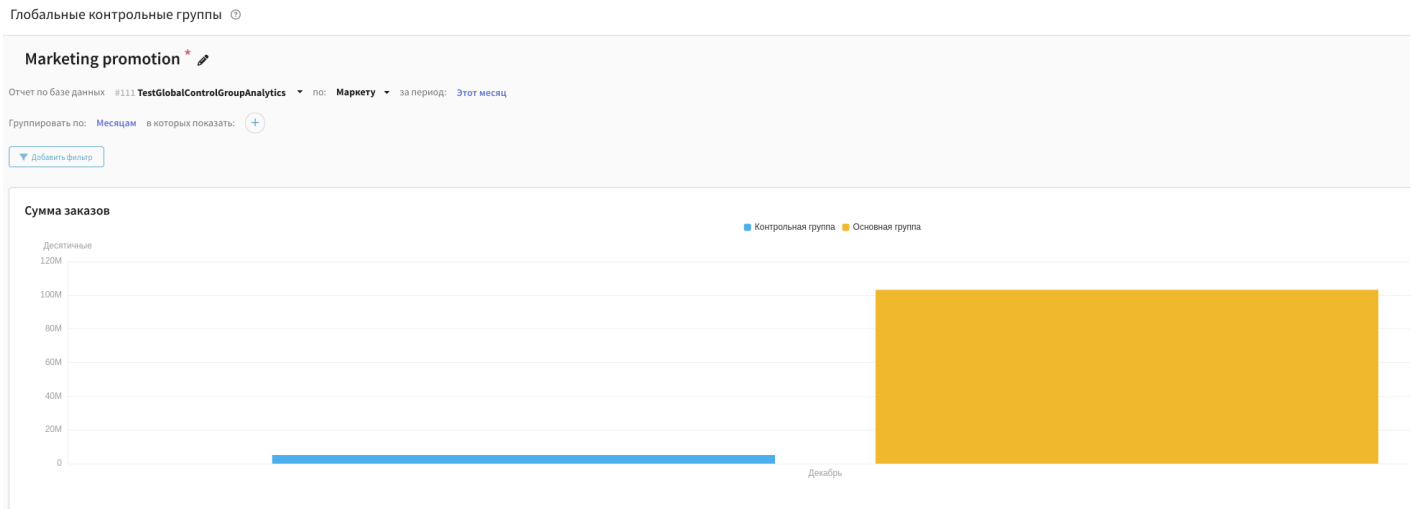
Глобальные контрольные группы (Global control groups) — часть клиентской базы, которая временно не получает маркетинговые коммуникации с целью анализа эффективности маркетинговых стратегий. Подробнее о контрольных группах вы можете узнать на этой странице.

Отчет по глобальным контрольным группам доступна на соответствующей вкладке раздела **Аналитика**:

TIP

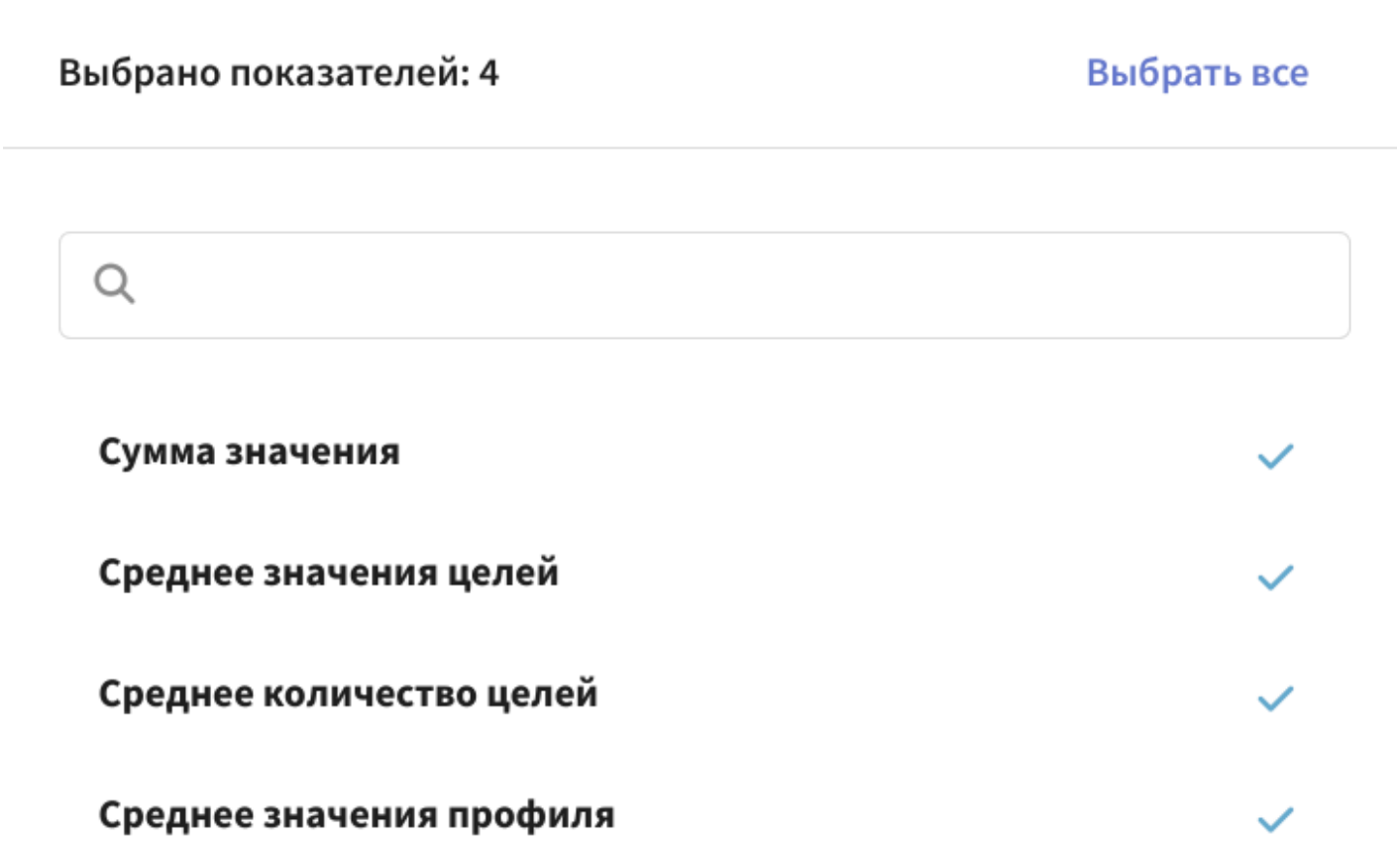
Аналитика по глобальным контрольным группам основана на показателях, собранных либо по данным Маркета, либо по данным целей пикселей. Для этого необходимо предварительно **настроить их** в платформе. Подробнее о Маркете можно прочитать здесь, а в этом разделе находится информация о пикселях отслеживания.

Отчет представляет сравнение данных по контрольной и основной группам. Информацию можно получить по маркету и по целям. Для метрик маркета можно выбрать **канал продаж**, а для целей доступен выбор **пикселей** и **целей пикселей**. На экране будет отображен график, основанный на указанных показателях обеих групп:



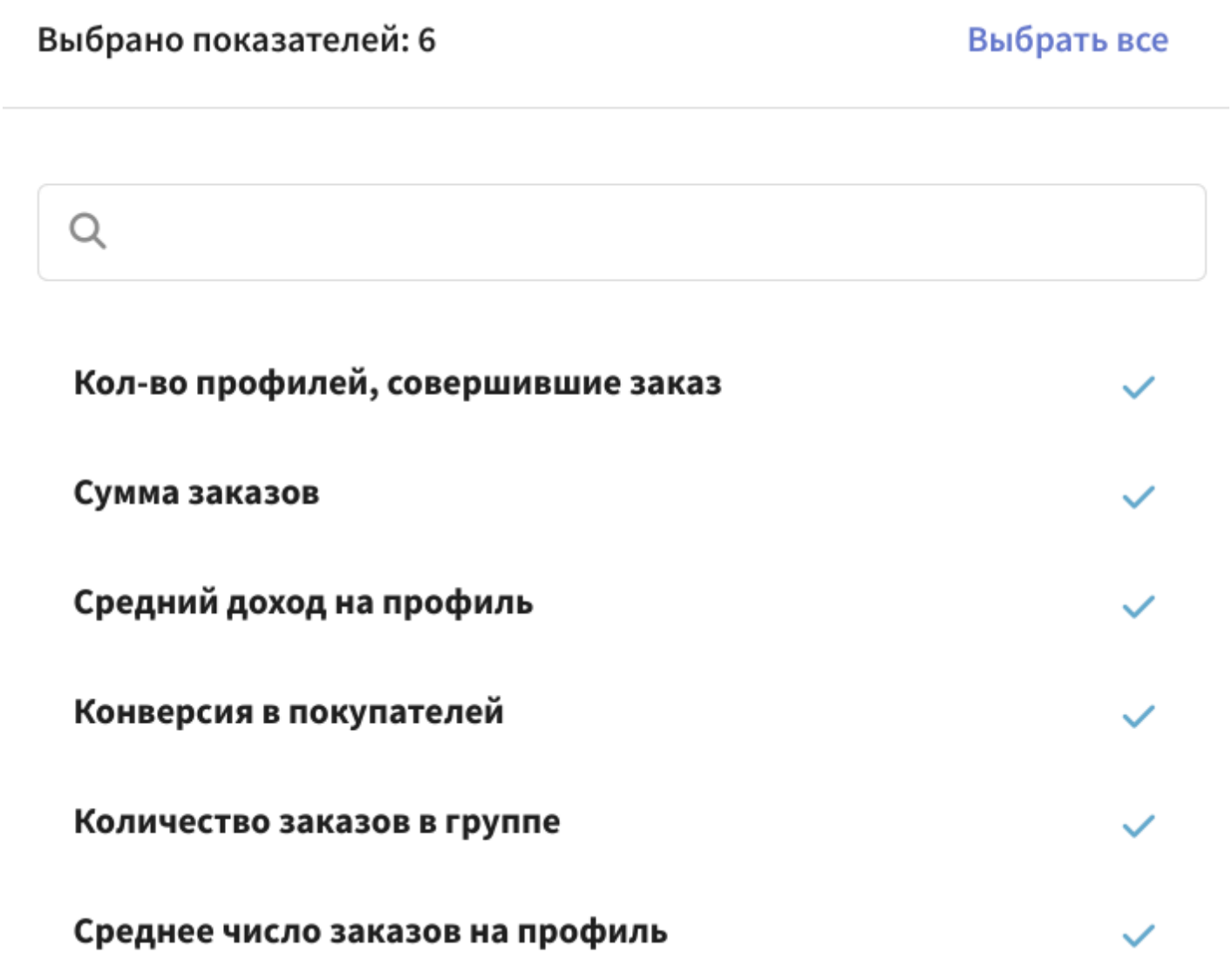
В таблице ниже вы можете увидеть значения показателей. По **целям** доступны следующие метрики:

- Сумма значение
- Среднее значение целей
- Среднее количество целей
- Среднее значение профиля

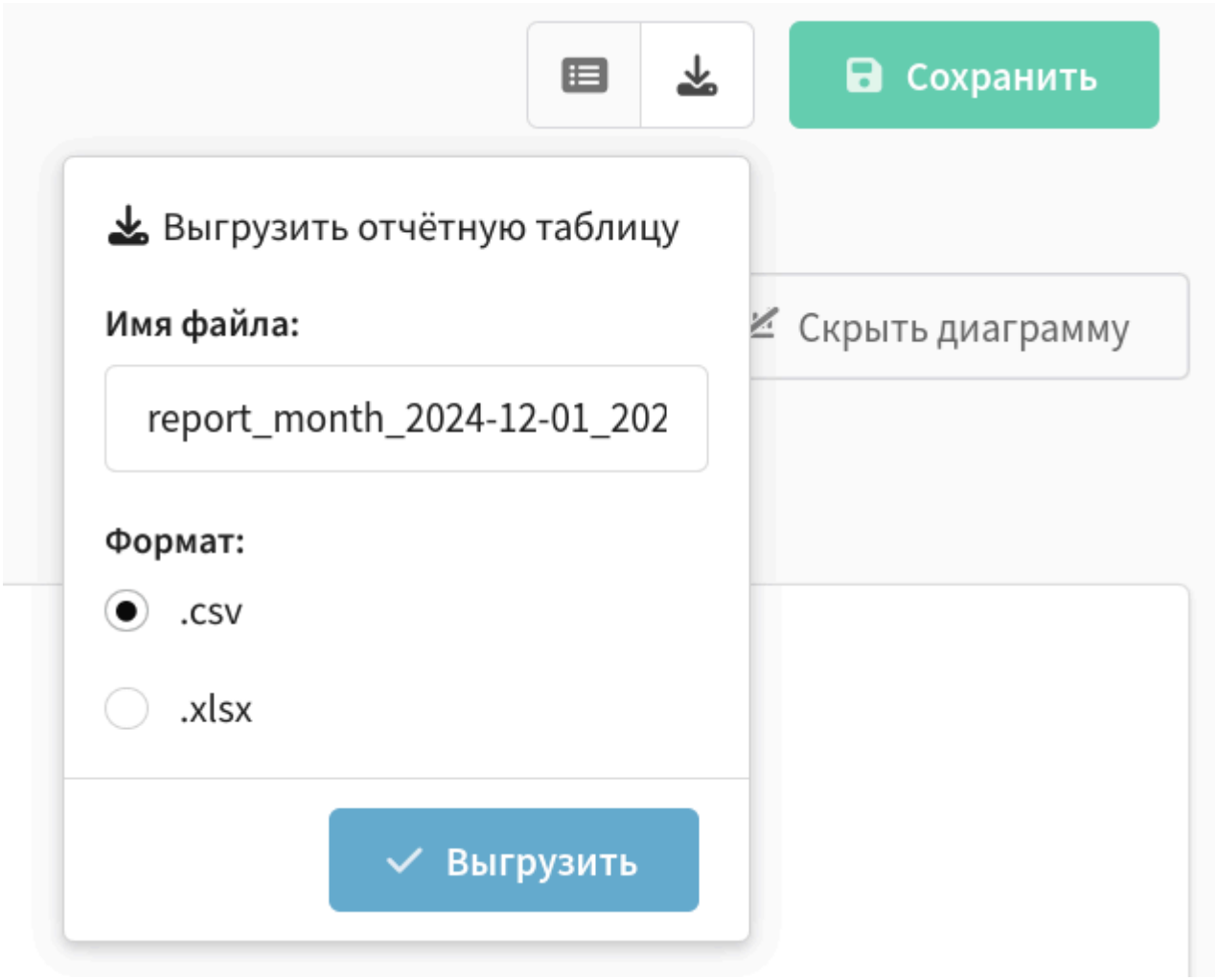


Для **маркета** доступны такие показатели, как:

- Кол-во профилей, совершивших заказ
- Сумма заказов
- Средний доход на профиль
- Конверсия в покупателей
- Количество заказов в группе
- Среднее число заказов на профиль



Вы можете **сохранить** отчет или **выгрузить** его в любом удобном формате, нажав на соответствующую кнопку в верхней части экрана:



Сохраненный отчет будет доступен в **списке отчетов**, доступном рядом с кнопками выгрузки и сохранения:

Отчеты по каналам

Сохраненные отчеты

Поиск...

Marketing promotion

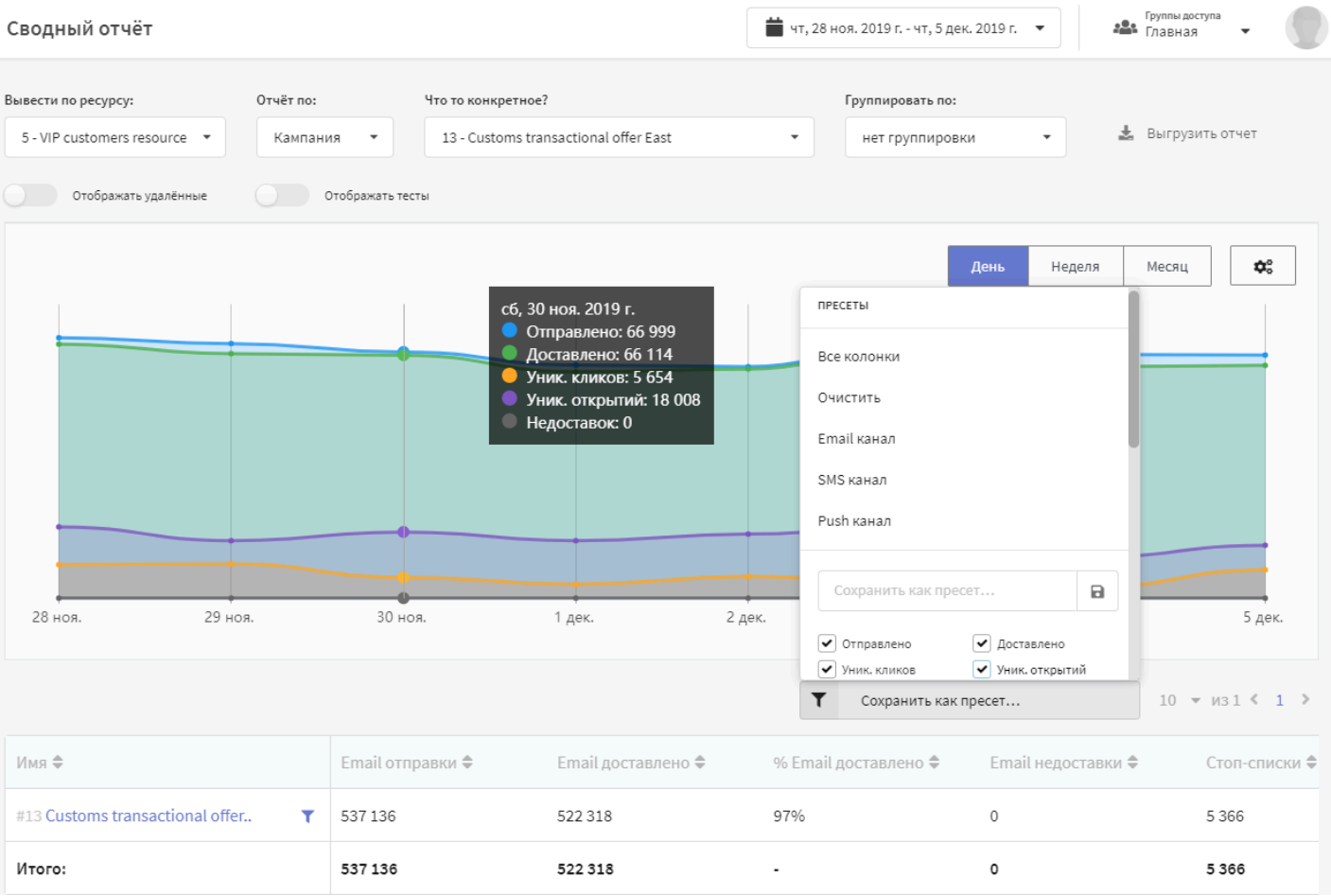
× Закрыть

Сводный отчёт

CAUTION

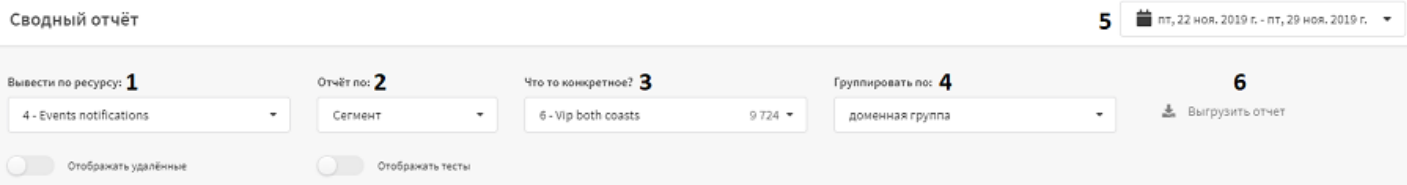
Поддержка сводного отчёта прекращена. Отчёт недоступен в версиях 65 и выше.

Сводный отчёт предоставляет детальную статистику по коммуникациям с числовыми и процентными (CTR и OR) показателями. В отчёте доступны пресеты для **Email**, **Push** и **SMS** каналов связи и различные виды группировки показателей: по ресурсам, базам данных, шаблонам сообщений, доменным группам Email.



Группировка объектов для анализа

По умолчанию сводный отчёт показывает данные по всем рассылкам. С помощью настроек в верхней части страницы вы можете выбрать только необходимую информацию:



В зависимости от выбранного типа отчёта данные можно дополнительно **сгруппировать**, чтобы просмотреть статистику по разным срезам и наборам показателей. **Зелёным цветом** выделены объекты, для которых доступны дополнительные группировки:

Ресурс (1)	Тип объекта (2)	Объект (3)	Группировка (4)
Отчёт по конкретному ресурсу или по всем доступным	Рассылка	Выберите конкретный объект для формирования отчёта. Исключение здесь составляют **домен, поставщик, день, месяц, неделя, окружение** — такие отчёты аналогичны соответствующим типам группировки — в рамках всех рассылок.	домен
	Сценарий		
	База данных		доменная группа
	Сегмент		Рассылка
	Шаблон сообщения		база данных
	Ресурс		сообщения
	Доменная группа (email)		ресурс
	Домен(email)		push провайдер
	Поставщик		поставщик
	День		по дням
	Неделя		по неделям
	Месяц		по месяцам
	Окружение		окружение
	Сендер		сендер

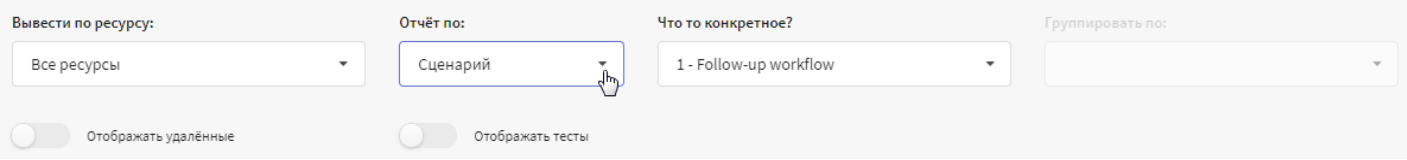
Кнопка **"Отображать удаленные"** добавит в отчёт информацию по удаленным объектам. Кнопка **“Отображать тесты”** добавит тестовые отправки.

Период, за который вы хотите получить отчёт, настраивается в верхней части страницы (**5**).

Экспортировать отчет по таблице в виде MS Excel и CSV можно, нажав кнопку **Выгрузить отчёт** (**6**).

Сводный отчёт по рассылкам в сценарии

Чтобы просмотреть детальную статистику по всем рассылкам, созданным в рамках Сценария автоматизации, выберите тип объекта **Сценарий** и необходимую последовательность взаимодействия.



График

На основе выбранных данных сводный отчёт формирует график, позволяющих отследить эффективность ваших действий в динамике: по **дням**, **неделям** и **месяцам** выбранного периода. Показатели, которые необходимо визуализировать, вы можете выбрать, нажав на кнопку настройки:

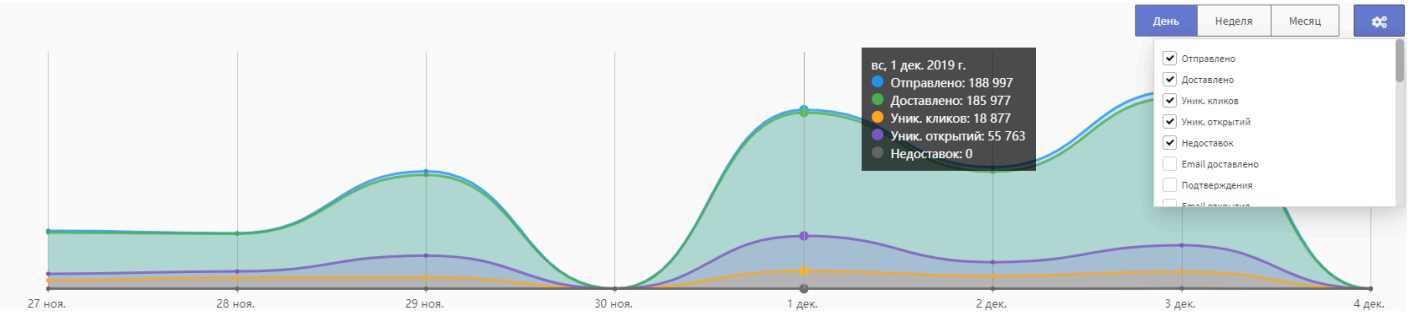
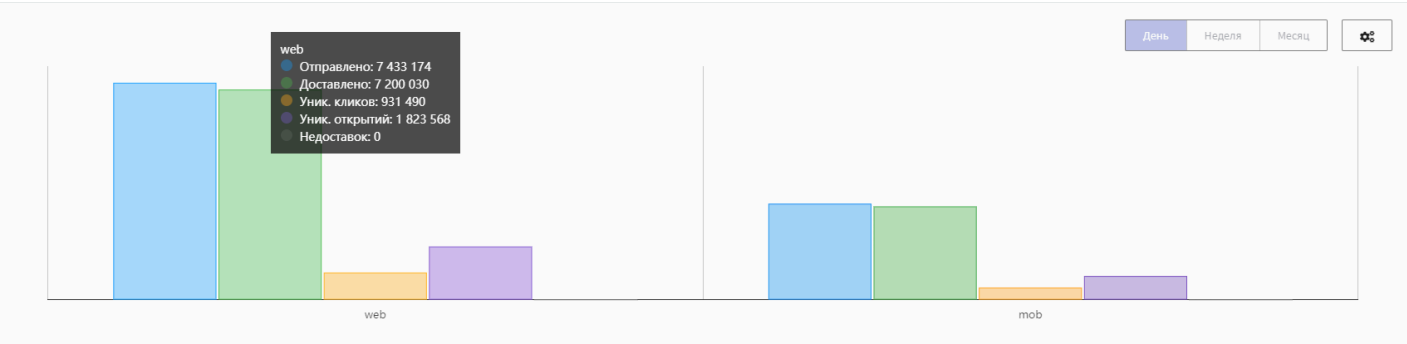


График для **доменных групп, доменов**, **поставщиков** и **окружения** представлен в виде столбцов:



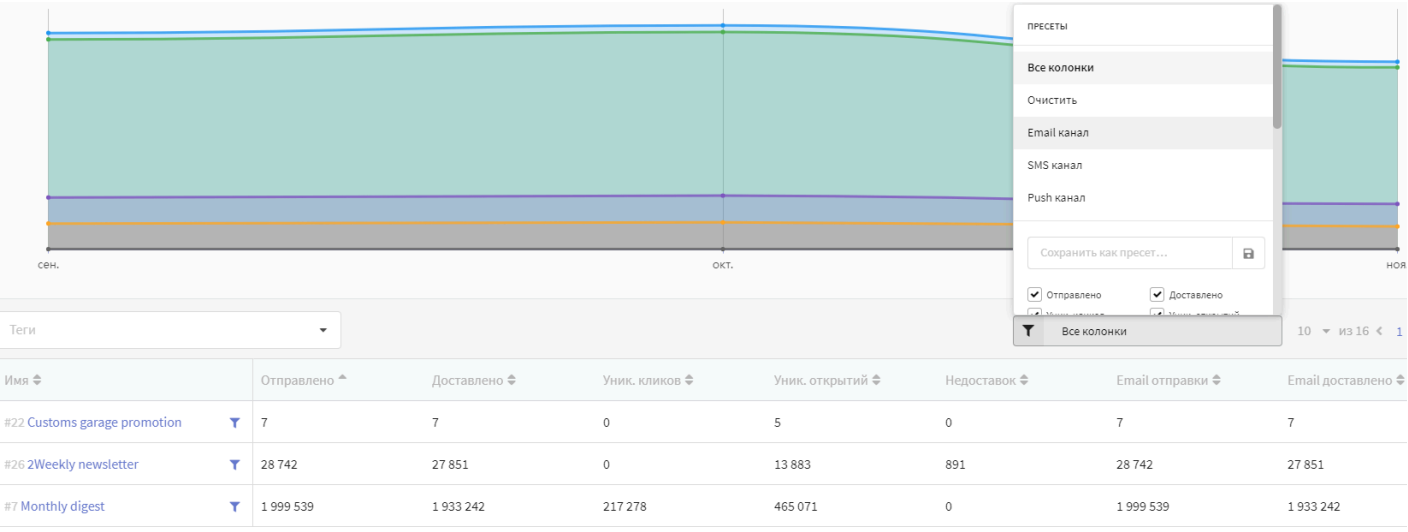
Показатели эффективности

Сводные данные отчёта отображаются в виде таблицы в нижней части страницы. Доступно более 70 различных показателей эффективности. Вы можете выбрать только необходимые вам метрики в выпадающем меню. Для удобства можно использовать **пресеты -** готовые наборы показателей.

Подробнее о показателях сводной таблицы можно узнать в разделе [“Все показатели сводного отчета”](#).

TIP

Вы можете воспользоваться поиском по тегам, чтобы дополнительно сгруппировать данные. Вы можете добавить произвольные теги к любой рассылке, базе данных, шаблону и другим объектам.



Все показатели сводного отчета

Канал	Показатель	Имя в API	Описание	Формула
Все	Отправлено	sends_total	Общее количество отправленных сообщений по всем каналам	sends + sends_push + sends_sms
Все	Доставлено	delivs_total	Общее количество доставленных сообщений по всем каналам	delivs + delivs_push + delivs_sms
Все	Уник. кликов	uclicks_total	Общее количество уникальных кликов в сообщениях, доставленных по всем каналам	uclicks + uclicks_push + uclicks_sms
Все	Уник. открытий	uopens_total	Общее количество уникальных открытий сообщений по всем каналам	uopens + uopens_push
Все	Недоставок	undelivs_total	Общее количество недоставленных сообщений по всем каналам. Помимо Softbounced и Hardbounced сообщений, туда также входят недоставки с неопределенными причинами (например, проблемы с сетью)	undelivs + undelivs_sms + undelivs_push
Все	Нарушение политик	offence	Количество нарушений коммуникационной политики	—
Email	Email отправки	sends	Количество отправленных email	—
Email	Email доставлено	delivs	Количество доставленных email	—
Email	Email открытия	opens	Количество зафиксированных открытий email. Если один и тот же пользователь открыл письмо несколько раз, будут зафиксированы все открытия.	—
Email	Email клики	clicks	Количество зафиксированных кликов в email	—
Email	Уник. email открытия	uopens	Количество уникальных открытий email. Значение равно количеству пользователей, открывших письмо. Если один и тот же пользователь открыл письмо несколько раз, будет зафиксировано только одно открытие.	—

Канал	Показатель	Имя в API	Описание	Формула
Email	Уник. email клики	uclicks	Количество уникальных кликов по ссылкам в email-сообщении. Если пользователь кликнул по ссылке в письме, будет зафиксирован один уникальный клик. Даже если в письме несколько ссылок и пользователь кликнул по каждой из них, будет также засчитан только один уникальный клик.	—
Email	Уник. email переходы	ulclicks	Количество уникальных переходов по ссылкам в email-сообщении. Если в письме несколько ссылок, будет фиксироваться уникальный переход по каждой из них.	—
Email	Email прочтений	reads	Количество прочитанных email. Событие прочтения регистрируется, если получатель открыл сообщение и не закрывал его как минимум 12 секунд.	—
Email	Email ответы	replies	Количество полученных ответов	—
Email	Email жалобы	complains	Количество email, на которые последовали жалобы на спам (FBL)	—
Email	Email hard bounces	hbounces	Количество email, которые возвращены из-за недействительных адресов или неожиданной ошибки во время отправки	—
Email	Email soft bounces	sbounces	Количество email, возвращенных из-за переполненности ящика или временной недоступности сервера получателя	—
Email	Подтверждения	confirms	Количество подтвержденных email-сообщений	—
Email	Email отписки	unsubs	Количество отписавшихся, т.е. пользователей, которые перешли по ссылке отписки в письме.	—
Email	Стоп-списки	suppresses	Количество email, которые не были отправлены, так как адрес получателя ранее был внесен в стоп-список	—
Email	Email недоставки	undelivs	Количество недоставленных email. Помимо Softbounced и Hardbounced сообщений, туда также входят недоставки с неопределенными причинами (например, проблемы с сетью)	—
Email	% Email доставлено	delivery_rate	Процент email, которые были успешно доставлены	(delivs / sends) * 100
Email	% Email открытий	u_open_rate	Процент получателей, открывших email	(uopens / delivs) * 100
Email	% Email CTR	u_click_rate	Процент уникальных кликов в email от email-доставок	(ulclicks / delivs) * 100

Канал	Показатель	Имя в API	Описание	Формула
Email	% Email CTOR	u_open_click_rate	Процент уникальных кликов в email от уникальных открытий email	(ulclicks / uopens) * 100
Email	% Email прочтений	email_read_rate	Процент получателей, прочитавших email, т.е тех, кто открыл письмо и не закрывал его как минимум 12 секунд.	(reads / delivs) * 100
Email	% Email ответов	email_reply_rate	Процент получателей, ответивших на email	(replies / delivs) * 100
Email	% Email жалоб	complain_rate	Процент email, на которые последовали жалобы на спам	(complains / delivs) * 100
Email	% Email hard bounces	hb_rate	Процент email, которые возвращены из-за недействительных адресов или неожиданной ошибки во время отправки	(hbounces / sends) * 100
Email	% Email soft bounces	sb_rate	Процент email, возвращенных из-за переполненности ящика или временной недоступности сервера получателя	(sbounces / sends) * 100
Email	% Email отписок	email_unsub_rate	Процент получателей, отписавшихся от рассылки в email-канале	(unsubs / delivs) * 100
SMS	SMS отправки	sends_sms	Количество отправленных SMS	—
SMS	SMS доставлено	delivs_sms	Количество доставленных SMS	—
SMS	SMS клики	clicks_sms	Общее количество кликов в SMS	—
SMS	Уник. SMS клики	uclicks_sms	Количество зафиксированных уникальных кликов в SMS. Если пользователь перешел по сссылке несколько раз, будет зафиксирован только один уникальный клик.	—
SMS	Стоп-списки SMS	suppresses_sms	Количество SMS, которые не были отправлены, так как номер телефона получателя ранее был внесен в стоп-список	
SMS	% SMS доставлено	sms_deliv_rate	Процент SMS, которые были успешно доставлены	(delivs_sms / sends_sms) * 100
SMS	Недоставки SMS	undelivs_sms	Количество недоставленных SMS	—

Канал	Показатель	Имя в API	Описание	Формула
SMS	% SMS CTR	sms_ctr	Процент уникальных кликов в SMS от общего числа доставленных SMS	(uclicks_sms / delivs_sms) * 100
SMS	SMS отписки	unsubs_sms	Количество профилей, которые перешли по ссылке отписке в SMS сообщении.	—
Push	Push отправлено	sends_push	Количество отправленных push	—
Push	Push открытия	opens_push	Количество зафиксированных открытий push	—
Push	Уник. push открытия	uopens_push	Количество зарегистрированных открытий push-уведомлений.	—
Push	Push доставлено	delivs_push	Количество доставленных push	—
Push	Недоставки Push	undelivs_push	Количество недоставленных push	—
Push	Push клики	clicks_push	Количество кликов в push-уведомлениях. Событие регистрируется только для веб-пушей.	—
Push	Уник. push клики	uclicks_push	Количество кликов в push-уведомлениях. Событие регистрируется только для веб-пушей.	—
Push	% Push открытий	push_open_rate	Процент открытых push-уведомлений	(uopens_push / delivs_push) * 100
Push	% Push доставлено	push_deliv_rate	Процент успешно доставленных push-уведомлений	(delivs_push / sends_push) * 100
Push	% Push CTR	push_ctr	Процент кликов в push от открытий push	(uclicks_push / uopens_push) * 100
Push	Push soft bounces	sbounce_push	Ответ о временной невозможности доставки сообщения (сетевые проблемы, недоступность push-сервиса по любым причинам). Это событие означает, что платформа продолжит попытки доставить сообщение.	—

Канал	Показатель	Имя в API	Описание	Формула
Push	% Push soft bounces	sb_push_rate	Процентное соотношение количества push-уведомлений, которые являются soft bounces, к общему числу отправленных уведомлений.	(sbounce_push / sends) * 100
Push	Push hard bounces	hbounce_push	Ответ о невозможности доставки push сообщения (недействительный идентификатор подписки). После получения такого ответа платформа не будет пытаться доставить сообщение повторно.	—
Push	% Push hard bounces	hb_push_rate	Процентное соотношение количества push-уведомлений, которые являются hard bounces, к общему числу отправленных уведомлений.	(hbounce_push / sends) * 100
WhatsApp*	Отправлено	whatsapp_send	Количество сообщений, отправленных в WhatsApp*	—
WhatsApp*	Доставлено	whatsapp_deliv	Количество сообщений, доставленных в WhatsApp*	—
WhatsApp*	Не доставлено	whatsapp_undeliv	Количество отправленных, но не доставленных сообщений в WhatsApp*	—
WhatsApp*	WhatsApp* подписка	whatsapp_subscribe	Количество новых подписок на WhatsApp*-канал	—
WhatsApp*	WhatsApp* отписка	whatsapp_unsub	Количество отписок в WhatsApp*-канале. Событие отписки фиксируется, если получатель перешёл по ссылке отписки в сообщении. Если он заблокировал бота, отписка не фиксируется.	—

INFO

Уникальное событие

Уникальность любого события ограничена временем, заданным в конфигурации платформы (параметр `UNIQUE_ACTIONS_TTL_MIN`), и по умолчанию составляет 1440 минут. Это значит, что повторное событие через сутки будет снова зачтено как уникальное. Уточните у администратора платформы, какое значение для уникальности установлено.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Синхронизация статических сегментов

Данные статических сегментов можно синхронизировать с внешними SQL таблицами данных. Такая интеграция используется, когда вам нужно поддерживать во внешних системах актуальную информацию о группах клиентов. Обновление данных во внешней системе будет происходить на основе определённых событий:

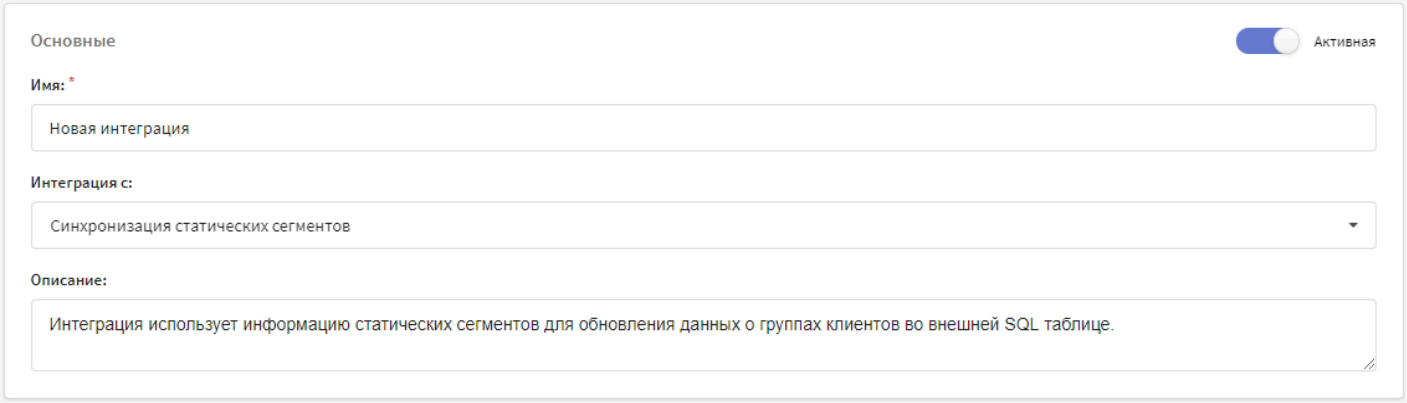
- Добавление профилей в статический сегмент.
- Удаление профилей из статического сегмента.
- Очистка (удаление всех профилей) статического сегмента.
- Создание или обновление статического сегмента.
- Удаление статического сегмента.

Чтобы добавить синхронизацию, в главном меню выберите **Интеграции** и в верхней части страницы нажмите кнопку **+ Создать**.

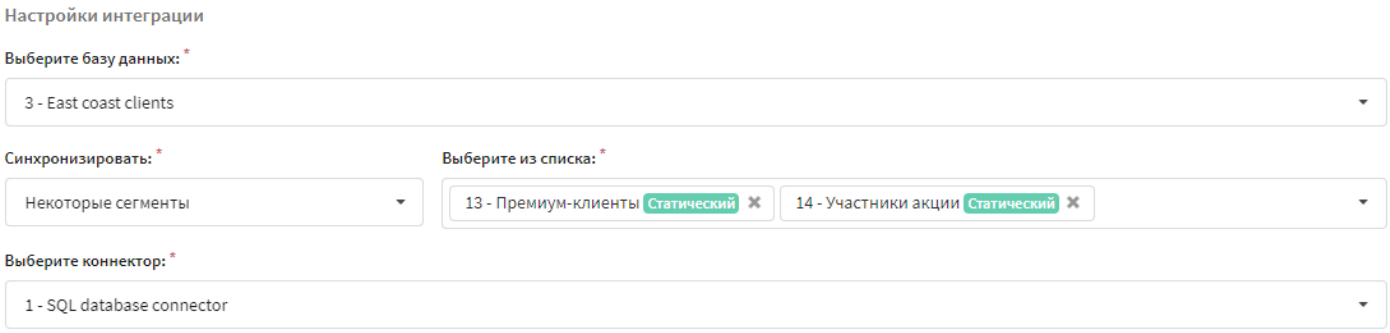
> **TIP**
>
> Для синхронизации в панели администратора Altcraft необходимо создать коннектор к внешнему SQL хранилищу. Инструкцию по добавлению коннектора вы можете найти здесь.

Настройки интеграции

В основных настройках интеграции введите **Имя** и **Описание**, и выберите тип: **Синхронизация статических сегментов**. Здесь же находится переключатель, с помощью которого вы можете активировать или остановить интеграцию:



Теперь задайте область видимости интеграции. Ниже в этом же блоке выберите SQL коннектор для хранилища с таблицей, в которую будут экспортироваться данные:



SQL-запросы для синхронизации событий

Ниже необходимо настроить SQL-запросы, которые будут добавлять или удалять данные во внешней таблице на основе событий, связанных с выбранными статическими сегментами.

TIP

В запросах могут использоваться дополнительные данные профиля и логические выражения. Соответствие полей профиля и колонок таблицы данных определяется оператором WHERE.

Добавление и удаление профилей, очистка сегмента

CAUTION

Запросы на добавление и удаление профилей являются **обязательными** для интеграции.

События добавления или удаления профилей клиентов в статический сегмент можно экспортировать по одному или батчем. В случае использования батча устанавливается количество добавленных профилей (**Размер батча)** и время **(Таймаут),** по прошествии которого синхронизация будет запущена, даже если необходимое количество профилей не накопилось. Значение таймаута по умолчанию — **60 секунд**.

Пример структуры таблицы данных

INFO

`segId` — идентификатор статического сегмента

`clientId` — уникальный идентификатор клиента в датацентре компании

`updated` — время добавления или удаления профиля

```
mysegment

+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| segId    | int(11)      | YES  |     | NULL    |       |
| clientId | varchar(255) | YES  |     | NULL    |       |
| updated  | datetime     | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

Пример запроса на добавление профилей батчем

INFO

`{segment_id}` — идентификатор сегмента в платформе

`{$lead.CustomID}` — кастомное (дополнительное) поле профиля.

Батч настраивается следующим циклом:

```
{for $index, $lead = $batch}
    ({segment_id}, "{$lead.CustomID}", NOW()){autocomma} // Здесь определяются поля для экспорта
  {end}
```

Полный список переменных Altcraft.

```
INSERT INTO mysegment (segId, clientId, updated)
  VALUES
  {for $index, $lead = $batch}
    ({segment_id}, "{$lead.CustomID}", NOW()){autocomma}
  {end}
  ON DUPLICATE KEY UPDATE
    updated = NOW();
```

Пример запроса на удаление профилей батчем

INFO

`{segment_id}` — идентификатор сегмента в платформе.

`{$lead.CustomID}` — кастомное (дополнительное) поле профиля.

Полный список переменных Altcraft.

```
DELETE FROM mysegment
WHERE segment_id = {segment_id}
AND clientId IN ({batch_in "CustomID"})
```

Пример запроса на удаление всех профилей из сегмента

INFO

`{segment_id}` — идентификатор сегмента в платформе.

Полный список переменных Altcraft.

```
DELETE FROM mysegment
WHERE segment_id = {segment_id}
```

Добавление/обновление и удаление сегмента

Дополнительно вы можете настроить экспорт событий о добавлении нового сегмента, обновлении данных существующего и удалении сегмента. Данные о сегментах экспортируются в отдельную SQL таблицу:

Пример структуры таблицы данных

INFO

`name` — название (Имя) статического сегмента

`segment_id` — уникальный ID сегмента в Altcraft

`updated` — дата события

```
segments

+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| name        | varchar(255) | YES  |     | NULL    |       |
| segment_id  | int(11)      | YES  |     | NULL    |       |
| updated     | datetime     | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
```

Пример запроса на добавление/обновление сегмента

```
INSERT INTO segments (segment_id,name,updated)
  VALUES
     ({segment_id}, "{segment_name}", NOW())
  ON DUPLICATE KEY UPDATE
    name = "{segment_name}",
    updated = NOW();
```

Пример запроса на удаление сегмента

INFO

`{segment_id}` — идентификатор сегмента в платформе

Полный список переменных Altcraft.

```
DELETE FROM segments
WHERE segment_id = {segment_id}
LIMIT 1
```

INFO

Если вы передаете данные профиля во внешнюю базу через переменные, то одинарные кавычки `'` будут экранироваться. В зависимости от типа базы для экранирования используются разные символы:

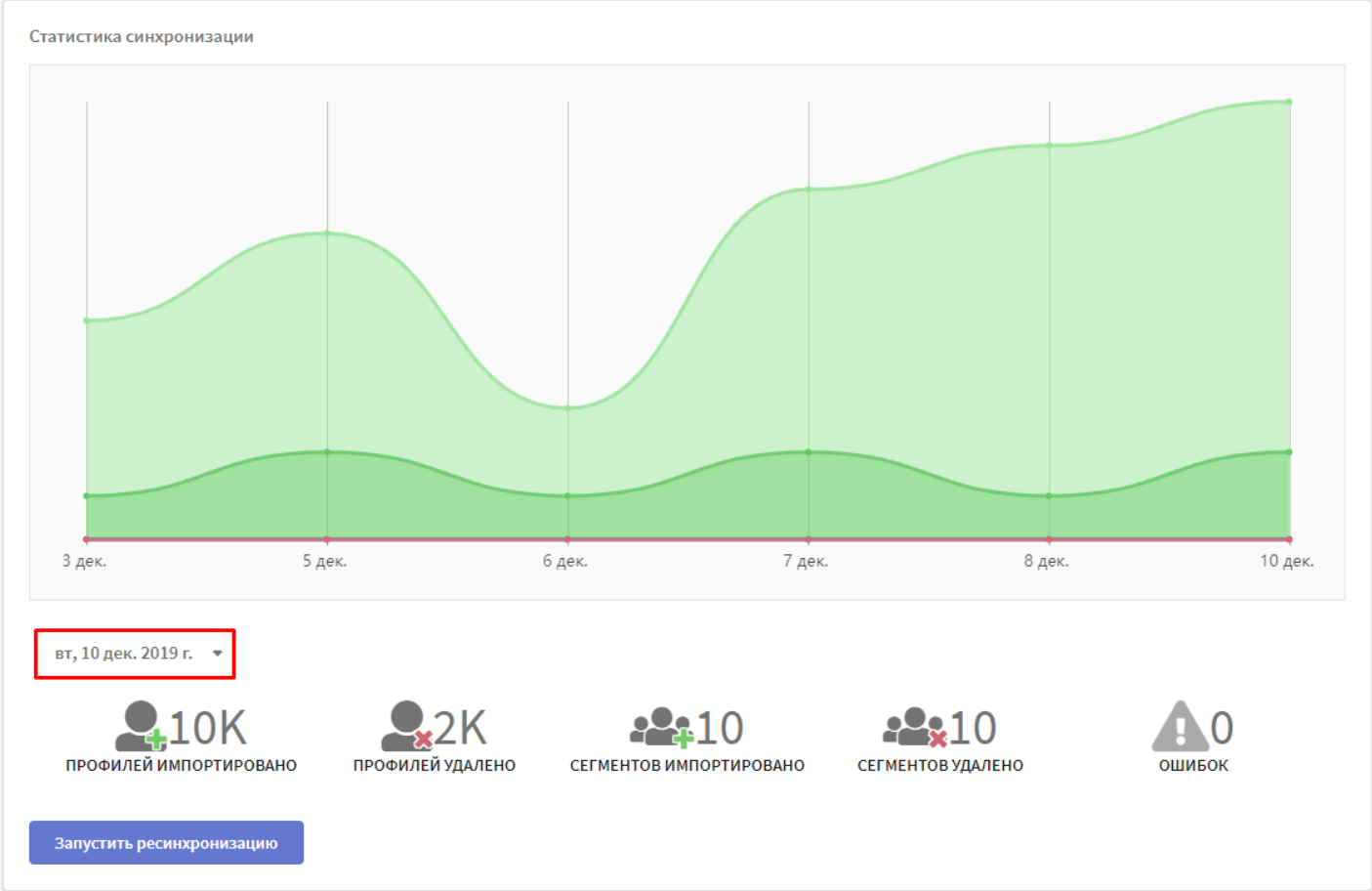
БД	Символ экранирования	Пример
MySQL, MSSQL, PostgreSQL, Oracle	`'` (одинарная кавычка)	`value''s`

БД	Символ экранирования	Пример
ClickHouse	`\` (обратный слеш)	`value\'s`

Статистика синхронизации

В блоке **Статистика синхронизации** будет отображаться информация по количеству успешных SQL запросов за **указанное время**, а также ошибки выполнения синхронизации.

С помощью кнопки **Запустить ресинхронизацию** вы можете перерассчитать данные сегментов для выгрузки.



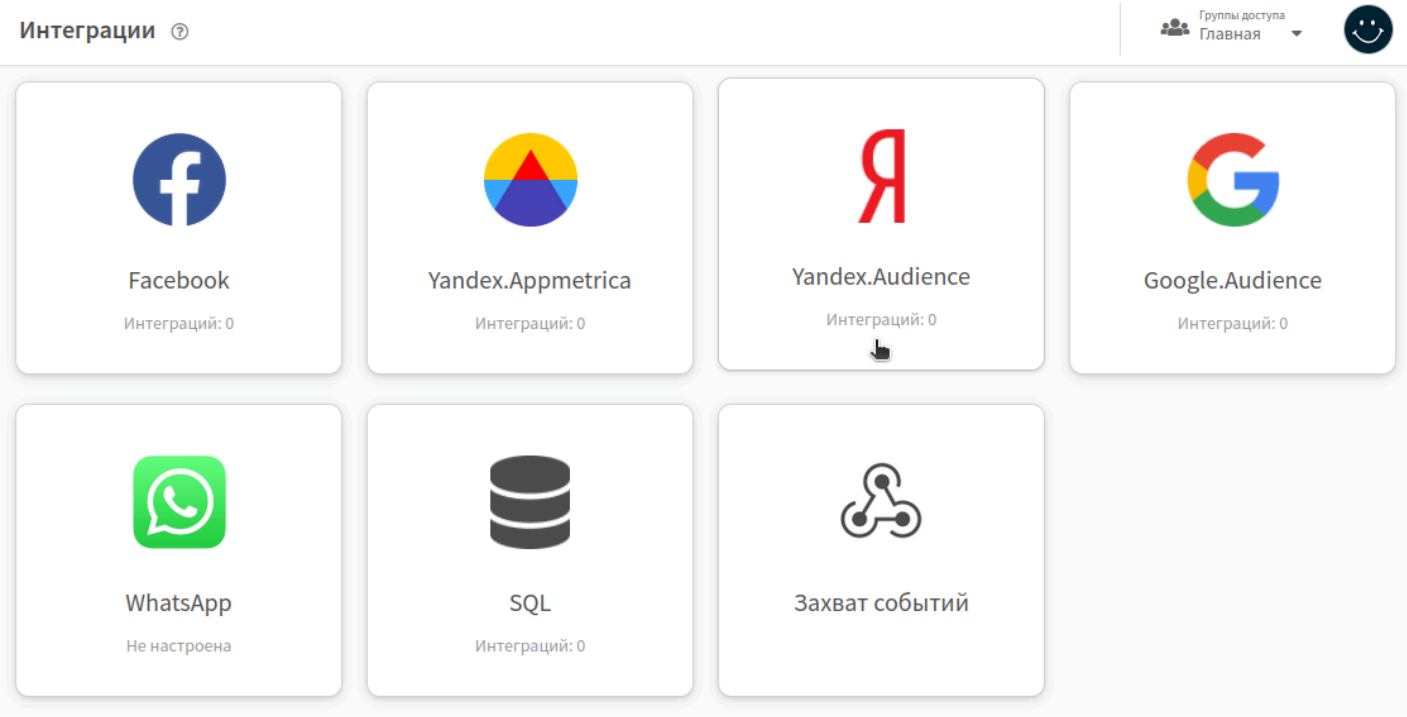
Яндекс.Аудитории™

Интеграция с Яндекс.Аудитории™ позволяет синхронизировать данные статических сегментов Altcraft Platform с сегментами в Яндекс.Аудитории™. Другими словами, вы переносите сегменты из платформы в Яндекс.Аудитории™, чтобы затем этот сервис, используя полученные данные о профилях, нашел нужных людей и запустил для них таргетированную рекламу.

Чтобы добавить синхронизацию, в главном меню кликните "**Интеграции**". Выберите тип интеграции — Yandex.Audience, а затем в верхней части страницы нажмите кнопку **+ Создать**.

TIP

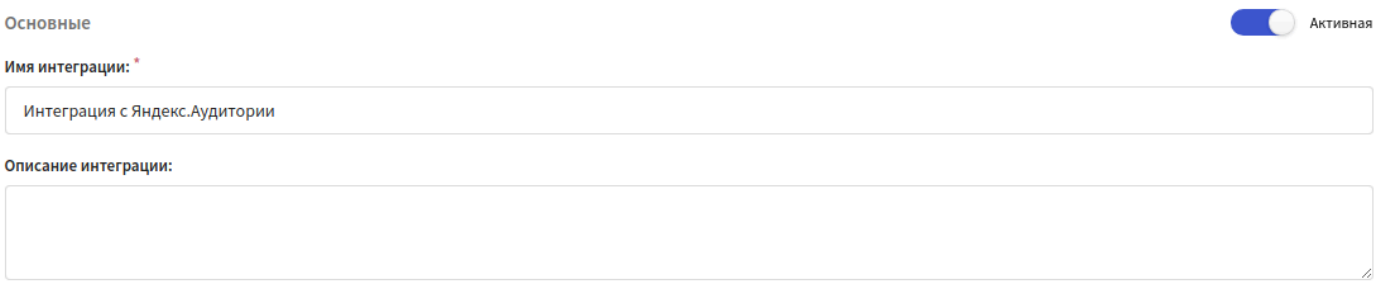
Для интеграции с Яндекс.Аудитории™ необходимо настроить **подключение** в платформе Altcraft. Подробную инструкции вы найдете здесь.



Настройка

Основные

На вкладке “Основные” введите **Имя** и **Описание**. Здесь же находится переключатель, с помощью которого вы можете активировать или остановить интеграцию:

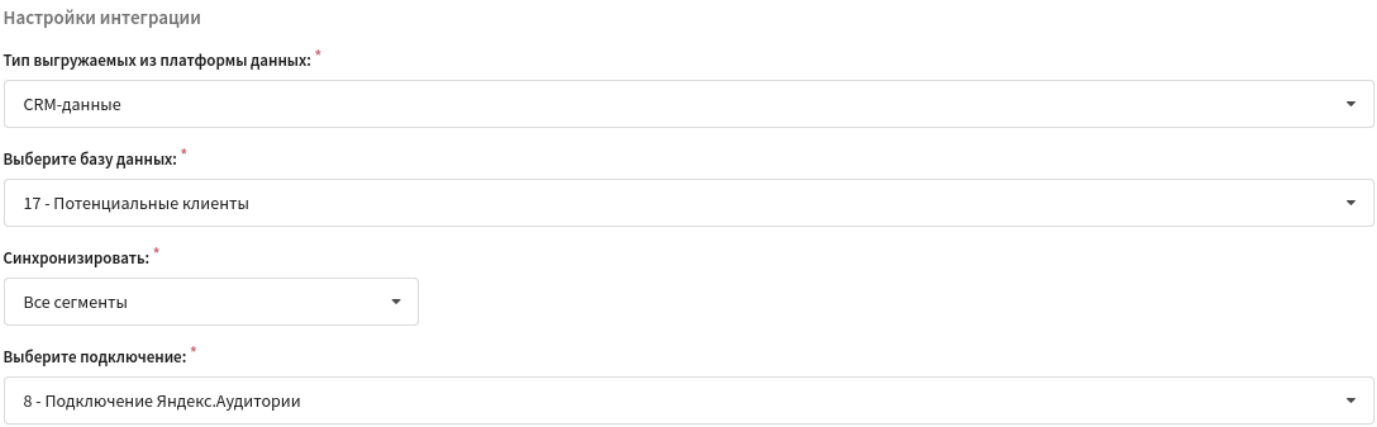


Настройки интеграции

Ниже выберите **тип выгружаемых из платформы данных:**

- **CRM-данные**: email и номер телефона.
- **Идентификаторы мобильных устройств**: данные берутся из подписок профилей. Яндекс.Аудитории принимает такие типы идентификаторов, как **AppMetrica with Google AID** и **AppMetrica with iOS IFA**.
- **MAC-адреса**: в базе данных должно быть дополнительное поле, в котором хранится mac-адрес профиля.

Задайте область видимости интеграции. Выберите подключение к Яндекс.Аудитории™, которое вы настроили ранее:

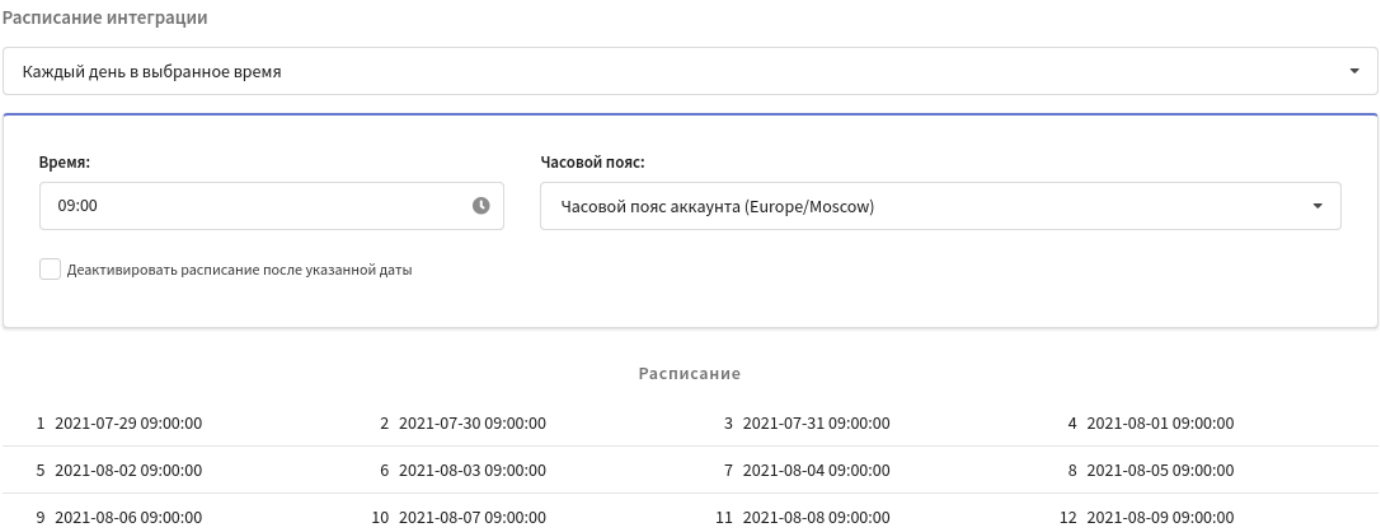


TIP

Обратите внимание, что в Яндекс.Аудитории™ нельзя построить сегмент, который будет содержать менее 100 профилей.

Расписание интеграции

Установите, с какой периодичностью вы хотите обновлять данные сегментов. Подробнее о том, как настроить расписание в платформе, читайте в этом разделе.

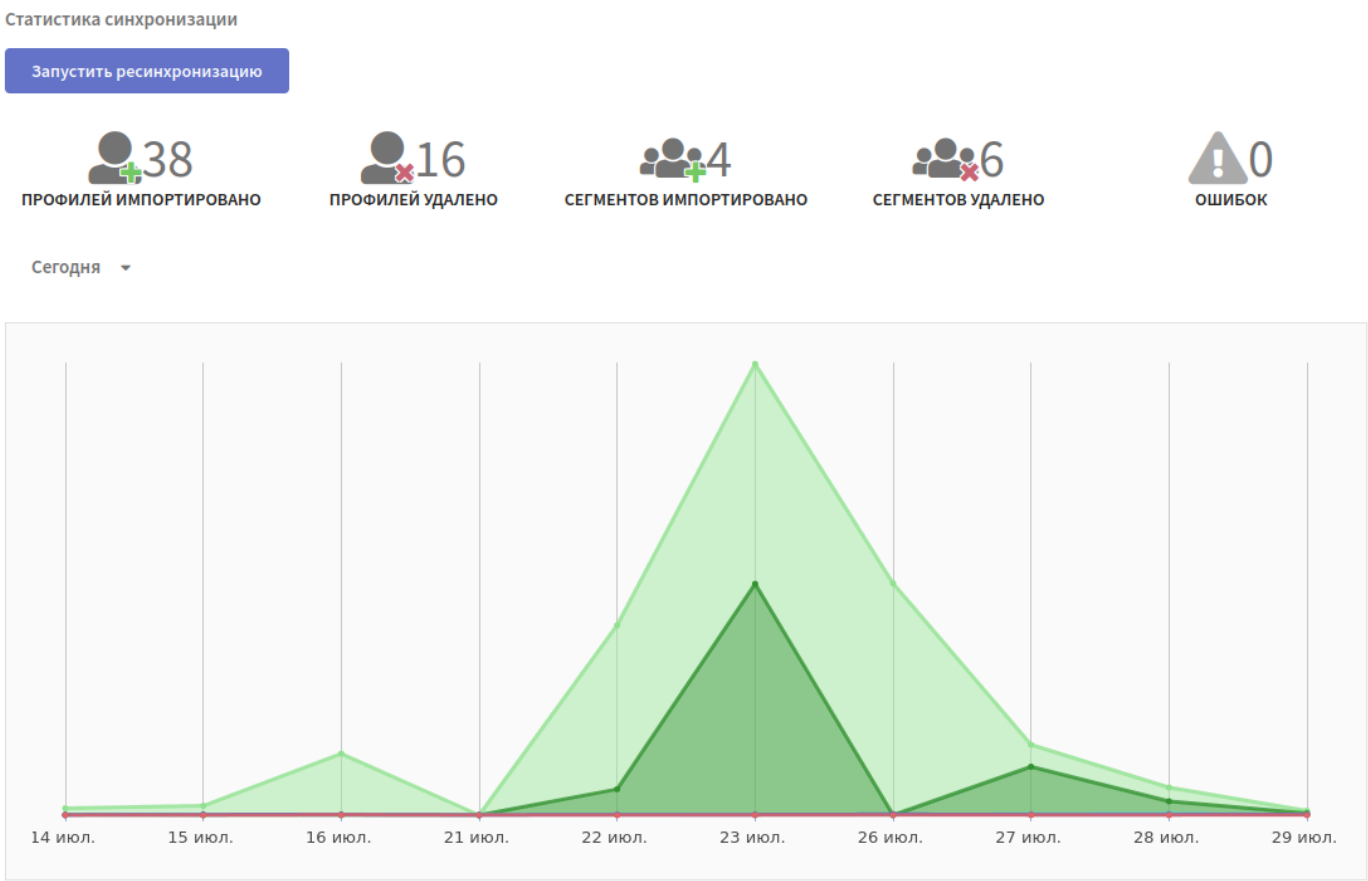


Статистика синхронизации

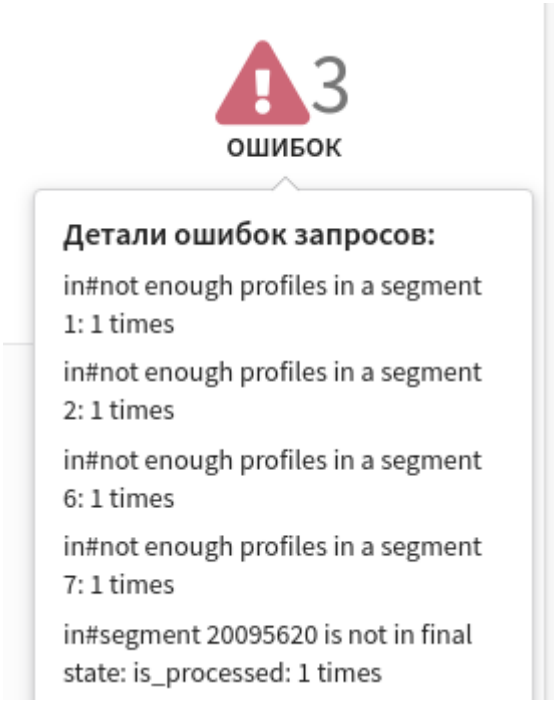
В этом блоке отображается информация о количестве успешных операций за указанное время, а именно, сколько:

- **профилей импортировано** (количество профилей в платформе, которые вошли в сегмент);
- **профилей удалено** (количество профилей в платформе, которые вышли из сегмента);
- **сегментов импортировано** (количество сегментов платформы, которые включили в синхронизацию с Яндекс.Аудиторией);
- **сегментов удалено** (количество сегментов платформы, которые исключили из синхронизации).

С помощью кнопки “**Запустить ресинхронизацию**” вы можете заново отправить данные по сегментам в Яндекс.Аудитории™.



Кроме того, здесь отображаются ошибки, возникшие при синхронизации. Кликните на соответствующую иконку, чтобы посмотреть детали ошибок:



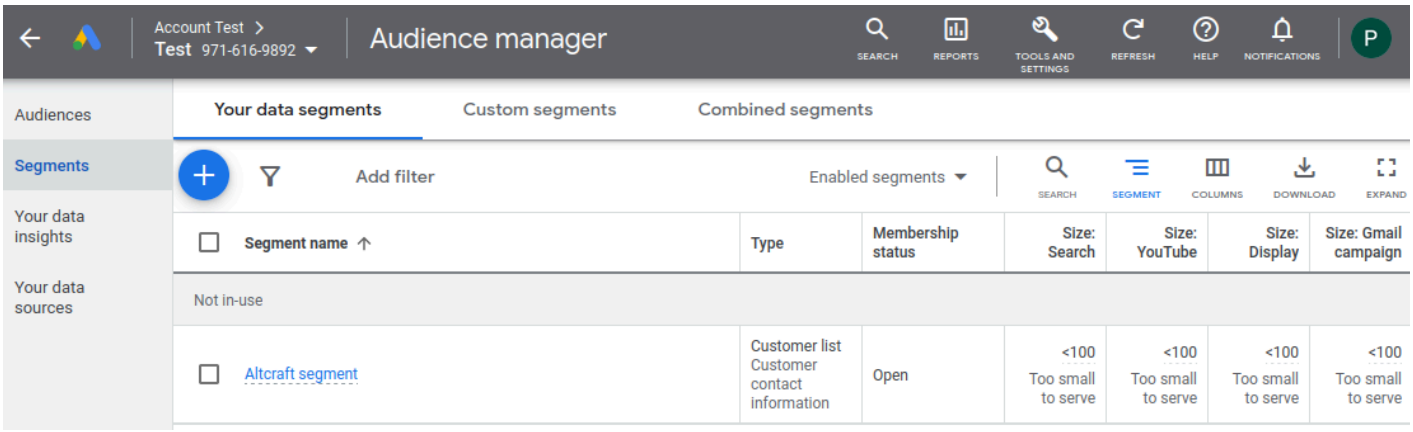
В таблице показаны текущие статусы сегментов, в которых они пребывают в Яндекс.Аудитории™ на данный момент:

Информация об аудиториях

Сегмент	Аудитория	Количество	Статус	Последняя синхронизация
#91 seg 1 ya.Au	seg 1 ya.Au (20054702)	772	Готов	2021-07-21 08:59:21
#92 seg 2 ya.Au	seg 2 ya.Au (20054703)	806	Готов	2021-07-21 08:59:22
#93 seg 3 ya.Au	seg 3 ya.Au (20054704)	877	Готов	2021-07-21 08:59:23

Аудитории Google Ads

Интеграция с Google Ads позволяет синхронизировать данные клиентов из статических сегментов Altcraft Platform с аудиториями в Google Ads. После экспорта сегментов в рекламный кабинет вы сможете использовать полученные данные для рекламы в поиске Google, Gmail, YouTube и контекстно-медийной сети Google.

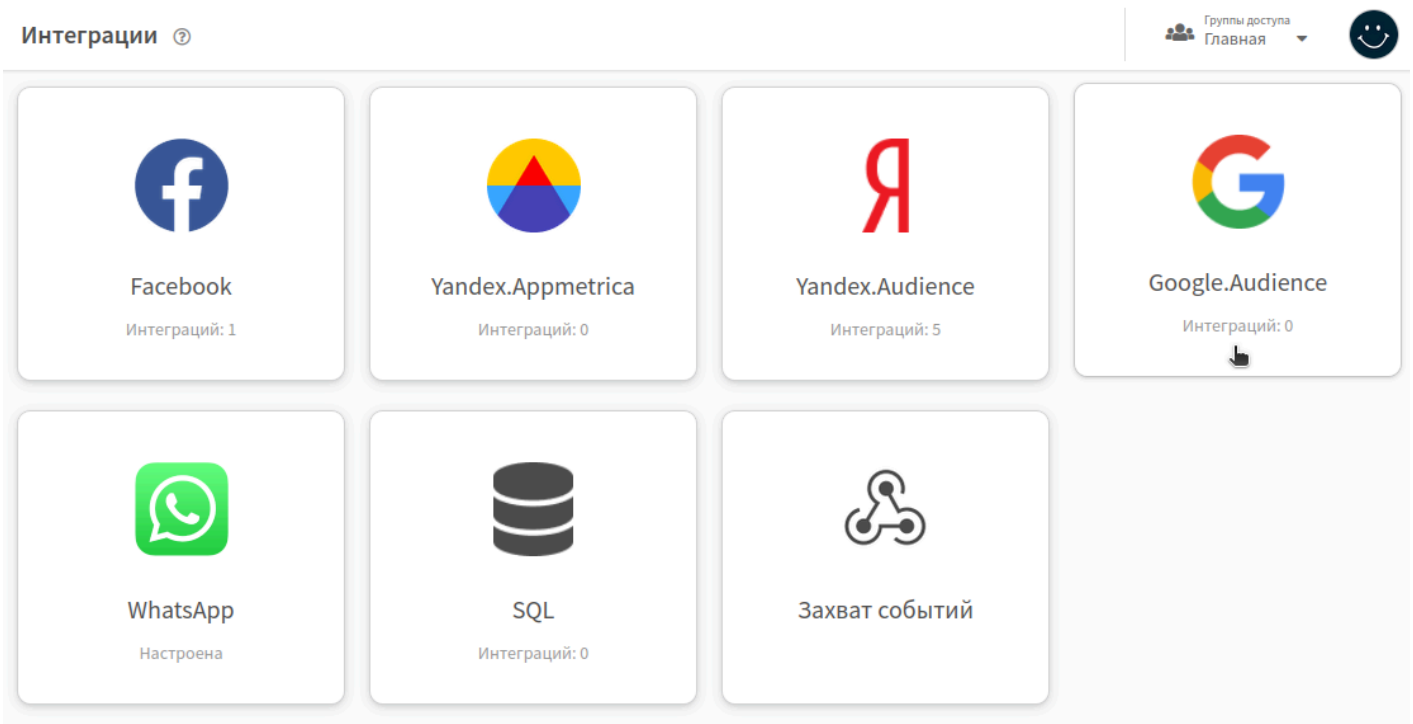


Для создания интеграции необходима учетная запись Google Ads и соответствующие учетные данные администратора.

Чтобы начать работу, перейдите в раздел "**Интеграции**". Выберите тип интеграции — Google.Audience. Откроется список готовых интеграций. Чтобы настроить новую, нажмите кнопку **+ Создать** на верхней панели .

TIP

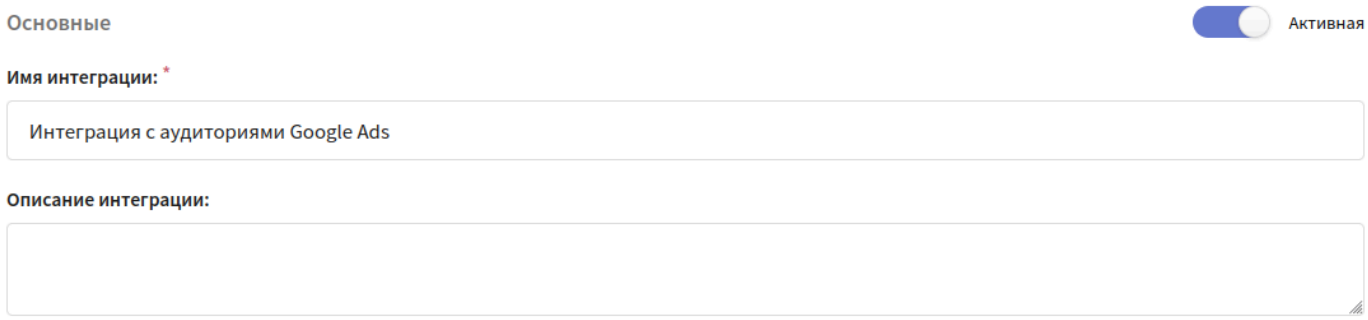
Для интеграции с аудиториями Google Ads необходимо настроить подключение в платформе Altcraft. Подробную инструкции вы найдете в этой статье.



Настройка

Основные настройки

На вкладке “Основные” введите **имя** и **описание**. Здесь же находится переключатель, с помощью которого вы можете активировать или остановить интеграцию:

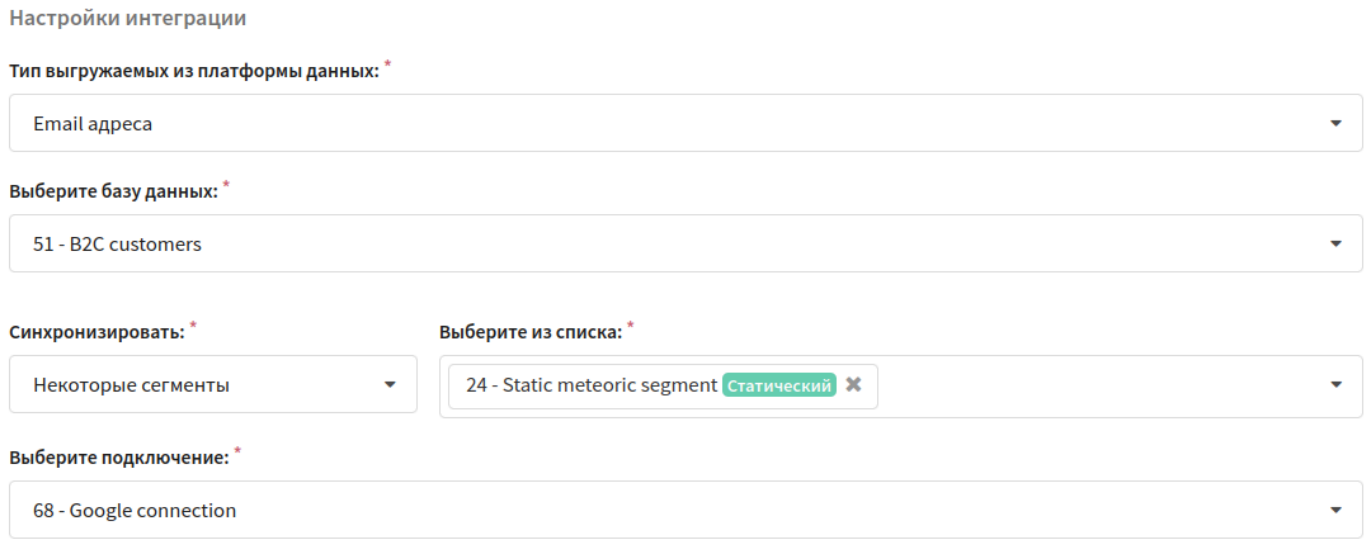


Настройки интеграции

Ниже выберите **тип выгружаемых из платформы данных:**

- Email адреса
- Рекламные идентификаторы (в базе данных должно быть дополнительное поле, в котором хранится рекламный идентификатор профиля)
- Номера телефонов

Задайте область видимости интеграции. Выберите подключение к Google Ads, которое вы настроили ранее.

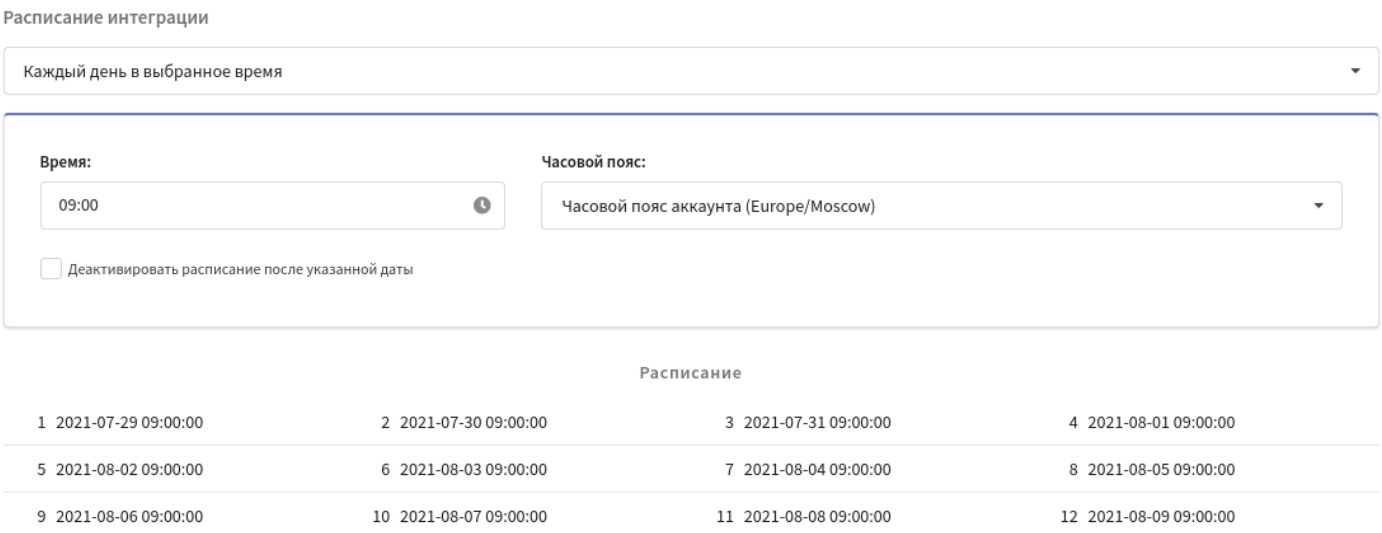


TIP

Обратите внимание, что в Google Ads нельзя показывать рекламу для сегмента, который содержит менее 100 профилей.

Расписание интеграции

Установите, с какой периодичностью вы хотите обновлять данные сегментов. Подробнее о том, как настроить расписание в платформе, читайте в этом разделе.



Статистика синхронизации

В этом блоке отображается информация о количестве аудиторий, которые были успешно добавлены или обновлены. Если в ходе синхронизации возникли ошибки, вы можете изучить детали, кликнув на соответствующую иконку.

С помощью кнопки **“Запустить ресинхронизацию”** можно заново отправить данные по сегментам в аудитории Google.

В таблице ниже собрана информация по каждому сегменту: название сегмента в Altcraft и сегмент в Google Ads, с которым он синхронизируется, а также размер аудитории и статус последней синхронизации. Вы можете посмотреть детальную историю всех синхронизаций, нажав на иконку

.

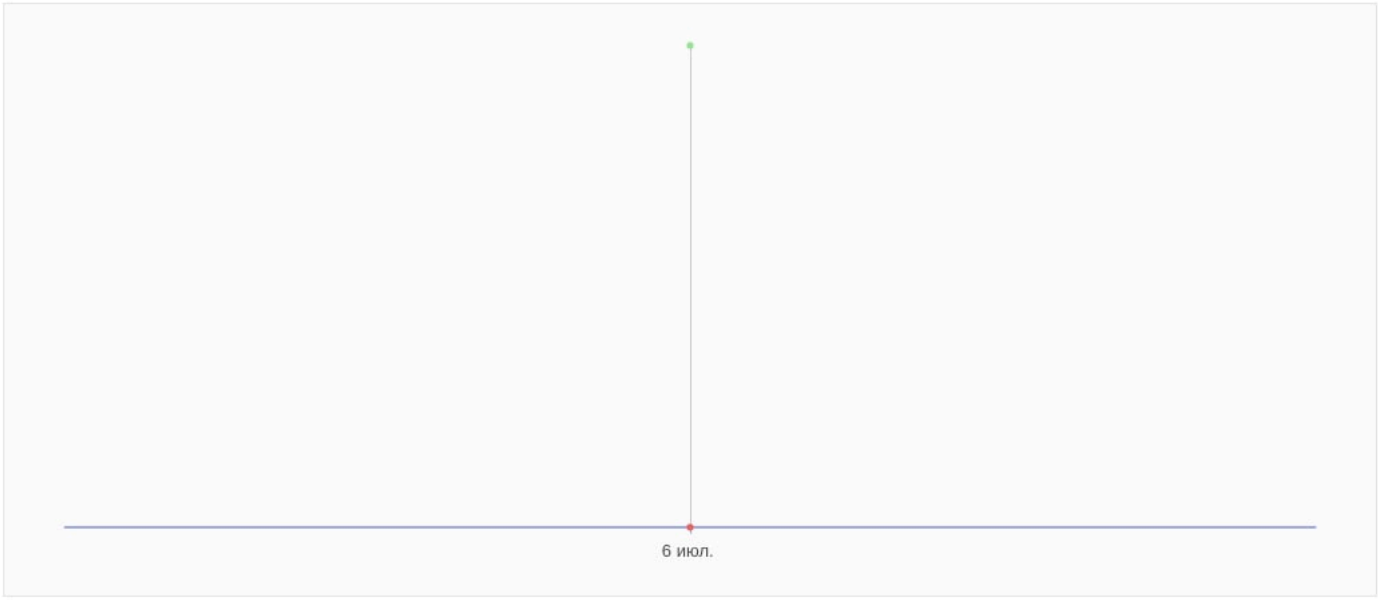
Статистика синхронизации

Запустить ресинхронизацию

2
АУДИТОРИЙ ДОБАВЛЕНО/ ОБНОВЛЕНО

0
ОШИБОК

Сегодня



Информация об аудиториях

Сегмент	Аудитория	Количество	Статус	Задания
#92 For Google Ads	(92) For Google Ads (emails) (customers/9547395462/userLists/7275945698)	0		

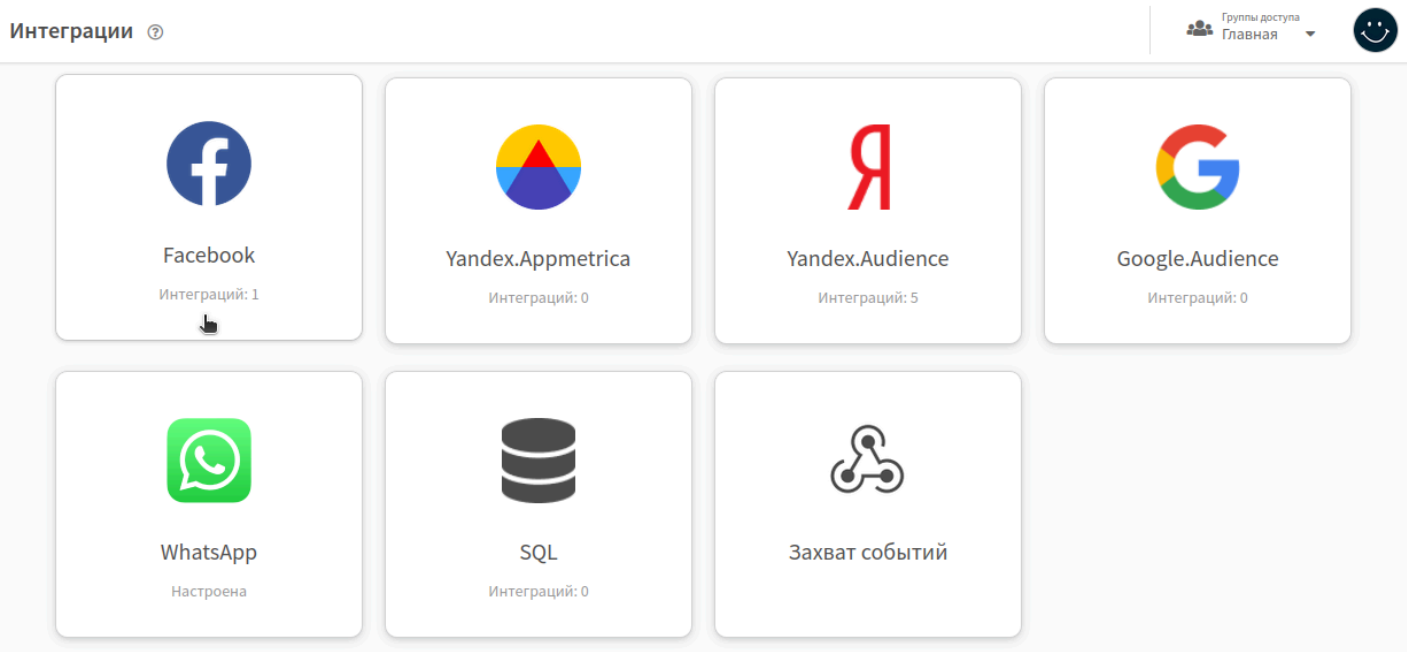
Facebook Ads Manager™

Интеграция с Facebook*™ Ads Manager позволяет синхронизировать данные статических сегментов Altcraft Platform с пользовательскими аудиториями в Facebook*™. Для этого вам необходимо создать отдельный статический сегмент в платформе Altcraft для каждой пользовательской аудитории Facebook*™.

Чтобы добавить синхронизацию, перейдите в раздел **"Интеграции"**. Выберите тип интеграции — Facebook*™. Откроется список готовых интеграций. Чтобы добавить новую, нажмите кнопку **+ Создать** в верхней части страницы.

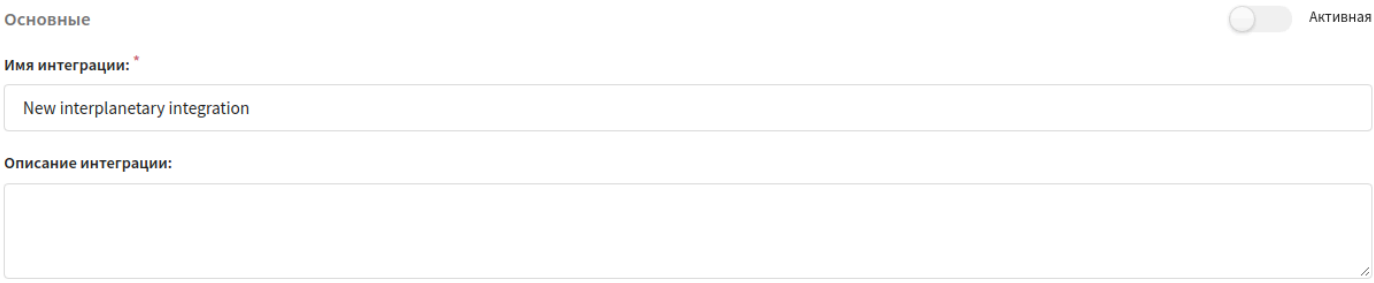
TIP

Для интеграции с Facebook* Ads Manager™* необходимо настроить подключение в платформе Altcrfat. Подробную инструкции вы найдете здесь .

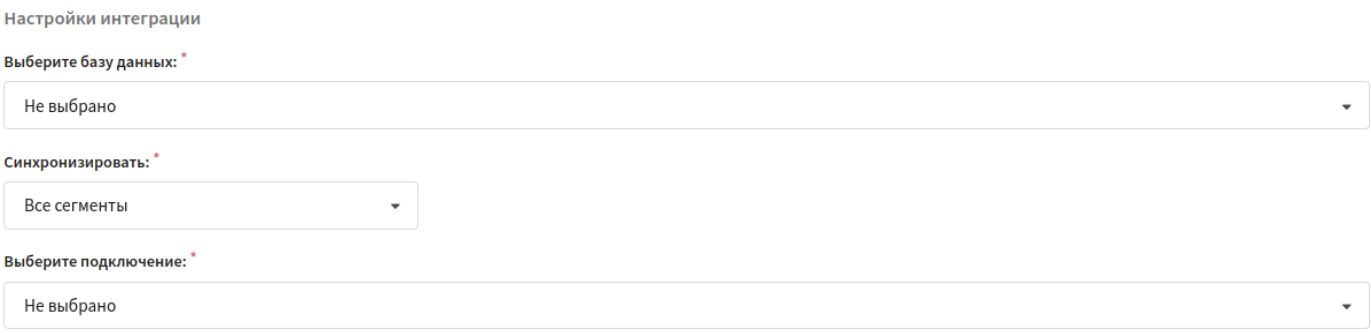


Настройки интеграции

На вкладке “Основные” введите **Имя** и **Описание**. Здесь же находится переключатель, с помощью которого вы можете активировать или остановить интеграцию:



Ниже задайте область видимости интеграции. Выберите подключение к Facebook*, которое вы настроили ранее:



Тип данных для поиска профилей

Теперь необходимо выбрать **тип данных для поиска профилей**:

Тип данных для поиска профилей — это информация, которую платформа Altcraft предоставляет для Facebook*™. Facebook*™ будет искать совпадение переданных данных с данными своих пользователей. Как только совпадение будет найдено, данный пользователь попадет в вашу пользовательскую аудиторию.

Тип данных для поиска	Описание
Контактные данные	Email и телефоны (в том числе из подписок)
Гео-данные	Страна, город
Личные данные	Имя, фамилия, дата рождения, пол
Внешний идентификатор	Матчинг по выбранному полю профиля

По умолчанию поиск профилей настроен по **контактным данным.**

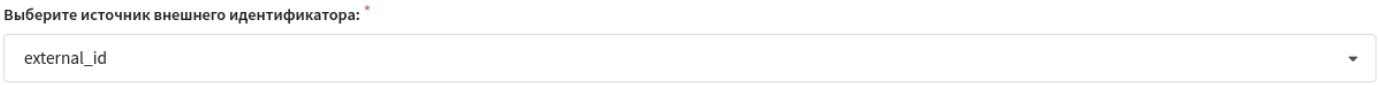
Для поиска профиля необходим идентификатор. Идентификатором могут выступать либо **контактные данные**, либо **внешний идентификатор**. По этой причине вы не можете в качестве типа данных для поиска профилей выбрать только **гео данные** или **личные данные** — вместе с ними обязательно должно быть отмечено одно из двух полей: контактные данные или внешний идентификатор.

TIP

Иногда в пользовательской аудитории Facebook*™ для одного подписчика, который был импортирован из статического сегмента Altcraft, создается несколько профилей. Такое поведение вызвано тем, что в платформе Altcraft этот профиль был подписан на несколько ресурсов и каждая подписка была перенесена в Facebook* по отдельности.

Внешний идентификатор

Если вы укажете поиск профилей по внешнему идентификатору, то ниже у вас появится новое поле, в котором вам необходимо **выбрать источник внешнего идентификатора**.



Внешний идентификатор необходимо завести не только в платформе Altcraft, но и в Facebook*™.

Внешний идентификатор в Altcraft — это кастомное поле в базе профилей. Добавьте это поле в базу, а затем выберите его как источник внешнего идентификатора.

Внешний идентификатор в Facebook*™ (External ID) — это уникальный строковой идентификатор, который вы передаете в Facebook*™ вместе с другими данными о пользователе. После этого Facebook*™ находит пользователя и привязывает этот id к нему. Подробнее об External ID в Facebook* читайте здесь.

Статистика интеграции с Facebook™

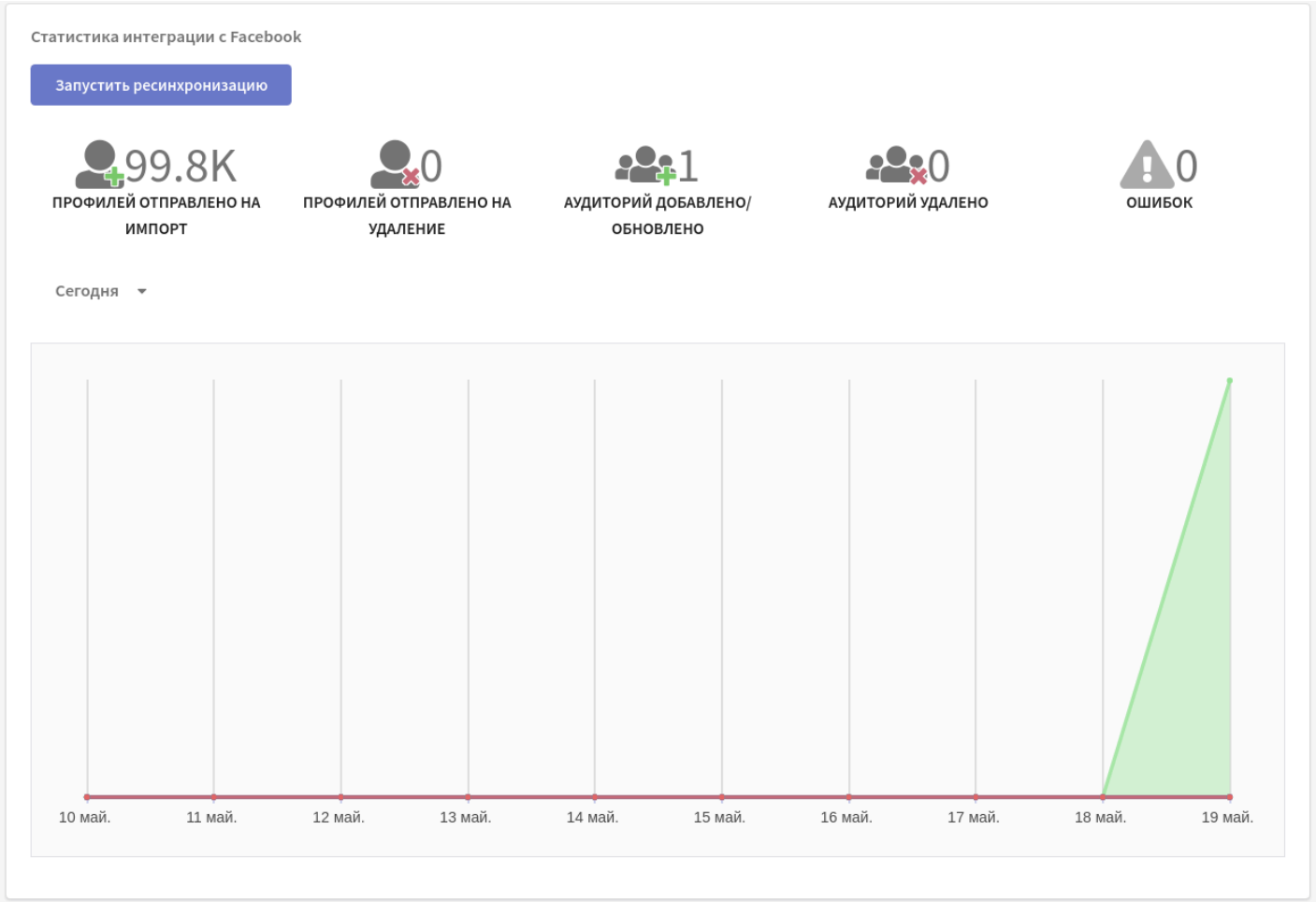
TIP: Расписание интеграции

Для этой интеграции невозможно настроить расписание ресинхронизации. Запуск ресинхронизации начинается **автоматически**, когда в привязанный к интеграции сегмент попадают новые профили.

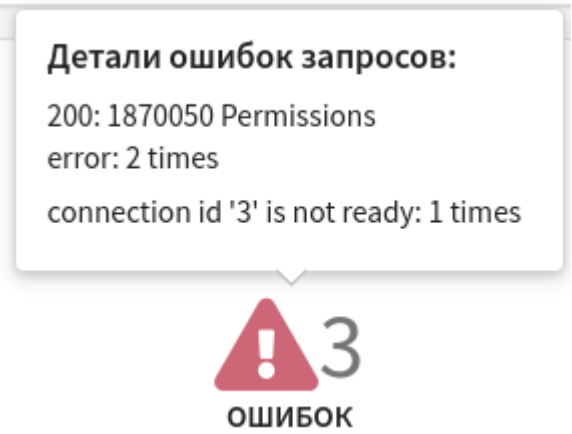
В этом блоке отображается информация о количестве успешных операций за указанное время, а именно, сколько:

- профилей отправлено на импорт
- профилей отправлено на удаление
- аудиторий добавлено/обновлено
- аудиторий удалено

С помощью кнопки “**Запустить ресинхронизацию**” вы можете заново отправить данные по сегментам в Facebook*™.



Кроме того, здесь отображаются ошибки, возникшие при синхронизации. Кликните на соответствующую иконку, чтобы посмотреть детали ошибок:

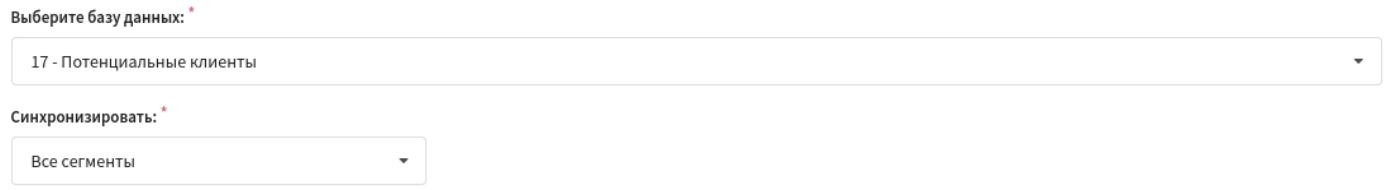


**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Область видимости интеграции

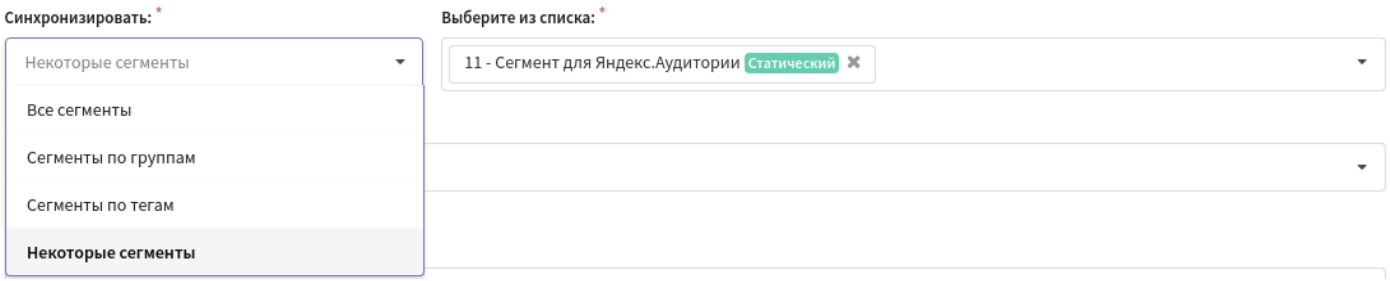
Область видимости интеграции — это область, в пределах которой находятся профили, данные которых вы хотите экспортировать во внешние системы. Она ограничивается базой профилей и статическими сегментами.

Настройка области видимости необходима при всех интеграциях, где происходит синхронизация со статическими сегментами платформы Altcraft Marketing.



Сегменты можно сгруппировать следующим образом:

- **Все сегменты** — будут выгружаться данные всех статических сегментов, связанных с выбранной базой данных.
- **Сегменты по группам** — будут выгружаться данные сегментов, для которых установлена определённая группа доступа. В меню справа выберите необходимые группы.
- **Сегменты по тегам** — будут выгружаться данные сегментов, для которых добавлены определённые пользовательские метки (**теги**). Теги можно добавить в основных настройках сегмента. В меню справа выберите необходимые метки.
- **Некоторые сегменты** — в меню справа вручную выберите необходимые сегменты.



WhatsApp

Интеграция с WhatsApp* позволит автоматизировать коммуникацию с подписчиками в мессенджере. Настроив этот канал, вы сможете:

- отправлять подписчикам персонализированные сообщения в приложении;
- сегментировать аудиторию на основе поведения пользователей;
- использовать WhatsApp*-канал для настройки сценария;
- собирать статистику и анализировать полученные данные.

CAUTION

Для отправки сообщения пользователю в WhatsApp* необходимо его согласие на получение сообщений от вашей компании. Подробнее.

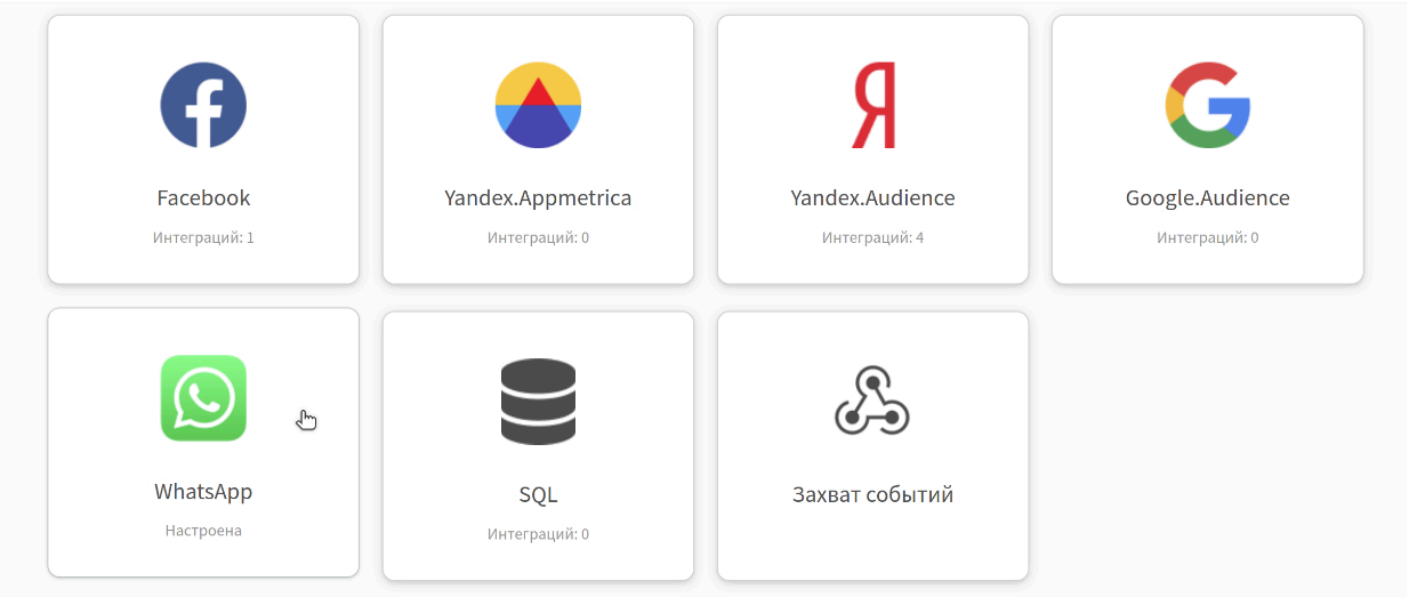
Интеграция реализована через официальных провайдеров: Edna или 360dialog. Зарегистрируйте аккаунт у одного из провайдеров, а затем свяжите аккаунт с Altcraft Platform через подключение. Инструкция по подключению к Edna и 360dialog доступна на портале документации.

TIP

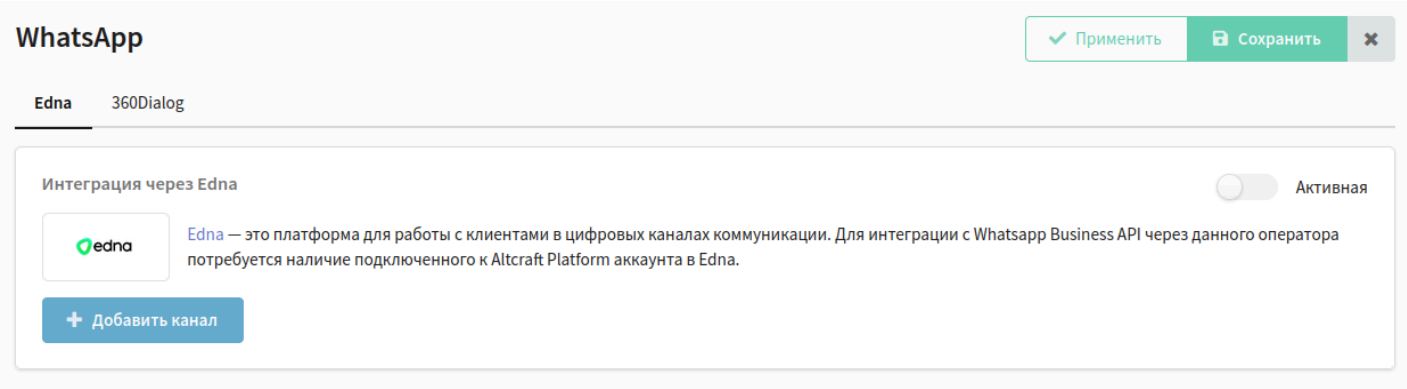
Помимо услуг провайдера, вам также придётся отдельно оплачивать сообщения, отправленные пользователям. Подробнее о стоимости сообщений в WhatsApp* вы можете узнать здесь.

Настройка интеграции

Откройте раздел **"Интеграции"** и выберите тип интеграции — **WhatsApp***:



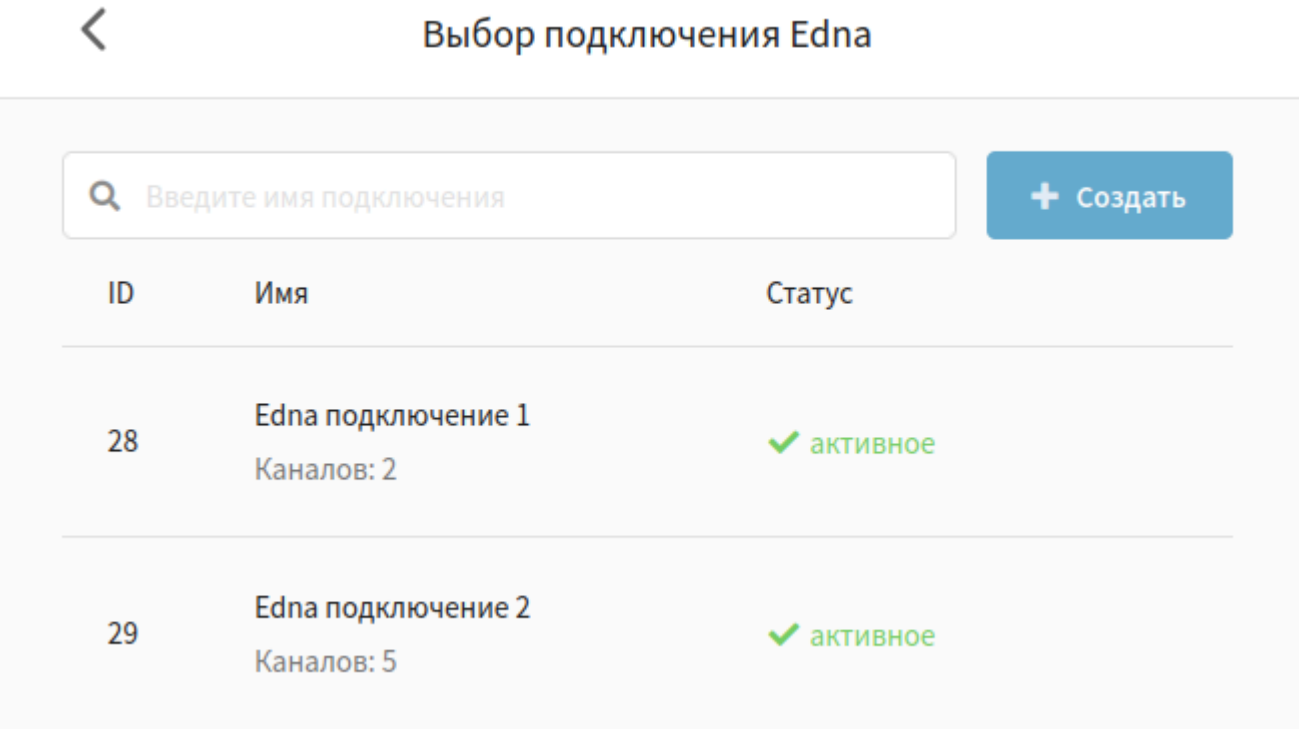
Перейдите на вкладку вашего провайдера:



Edna

Активируйте интеграцию с помощью переключателя в правом верхнем углу. Нажмите **“Добавить канал”.**

Справа появится окно, где сначала необходимо выбрать доступное подключение:



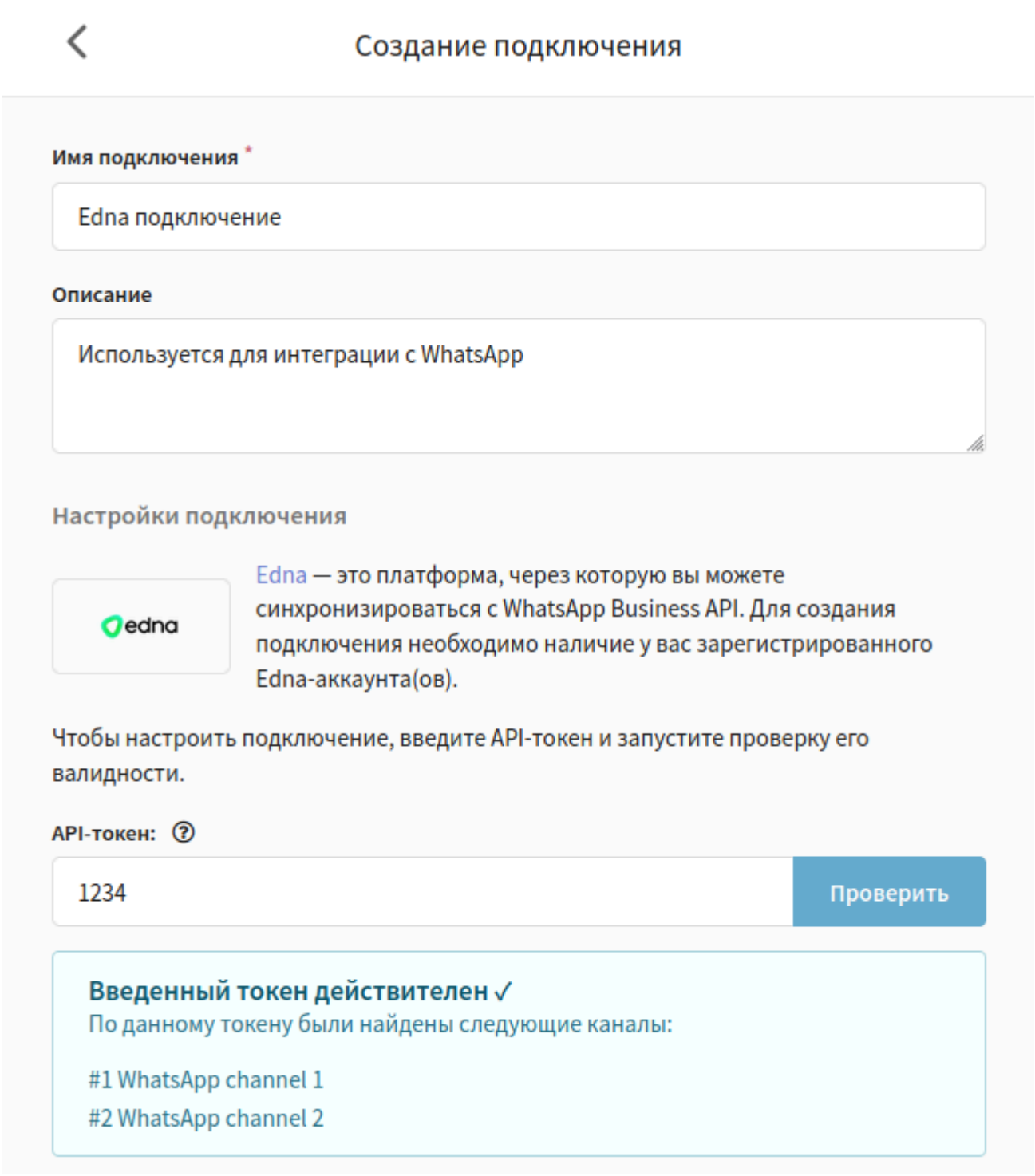
Нет доступных подключений

Если у вас ещё не настроено подключение к Edna, вы можете создать его непосредственно в этом окне.

Нажмите **“Создать”.** Введите имя и описание подключения. Далее укажите API-токен, сгенерированный в личном кабинете Edna, и запустите проверку его валидности. Если токен прошёл проверку, нажмите кнопку **“Создать”** в правом нижнем углу.

TIP

Подробнее о создании подключения к Edna можно прочитать здесь.



Далее выберите активный **канал**. Канал должен находиться в статусе “Работает исправно”.

После этого укажите **каскад.** Edna каскад – это сценарий, который задаёт последовательность отправки сообщений в несколько каналов, если в предыдущий канал сообщение не было доставлено.

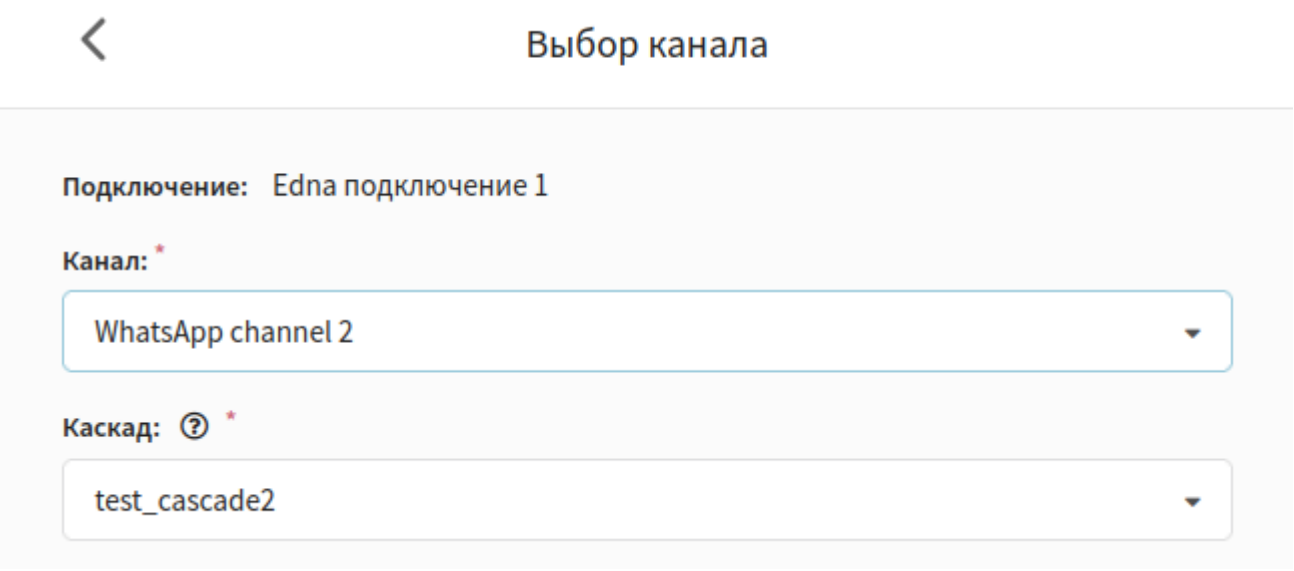
CAUTION

Некоторые каскады в списке могут быть недоступны. Такие каскады настроены на несколько каналов, и платформа считает их невалидными. Вы можете выбрать только каскады, в которых используется один канал.

При необходимости вы можете **включить захват событий в Edna** — это позволит передавать в Altcraft Platform данные о статусах сообщений. Если параметр активирован, после сохранения интеграции в личный

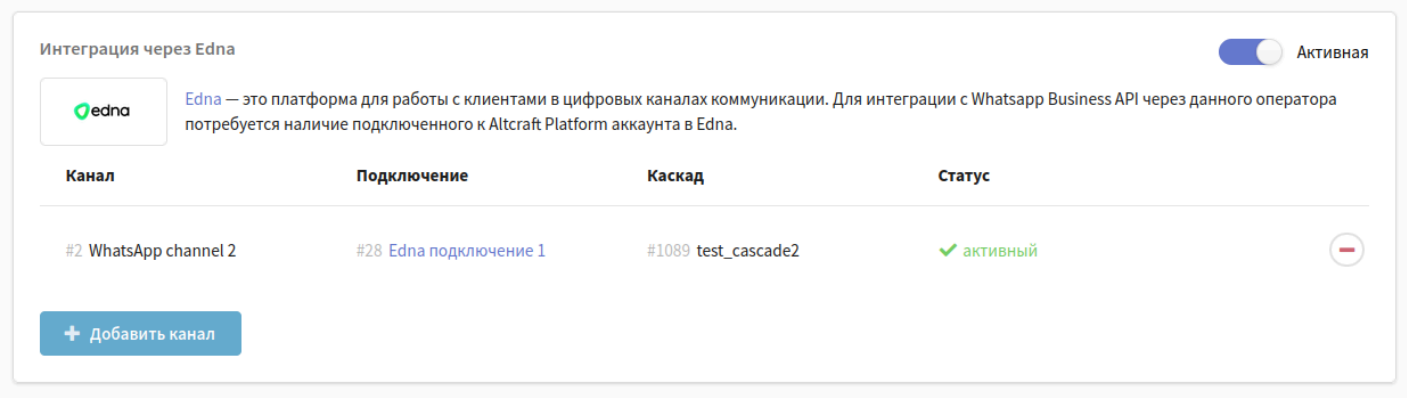
кабинет Edna автоматически будет добавлен Callback URL, через который события WhatsApp передаются в Altcraft. Если отключить этот параметр, аналитика по доставкам и прочтениям будет недоступна в платформе.

Настроив канал и каскад, нажмите **“Добавить”** в правом нижнем углу.



Данные по добавленному каналу отобразятся в интеграции. При необходимости вы можете добавить несколько каналов. Чтобы убрать канал из интеграции, нажмите

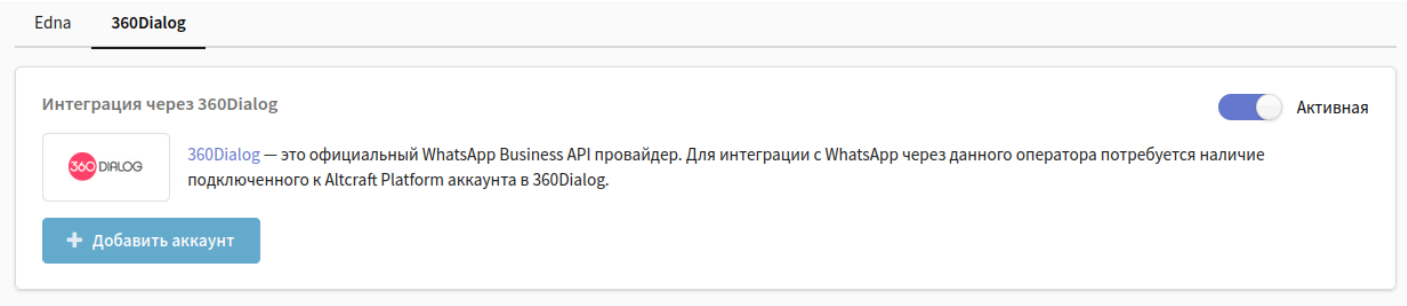
.



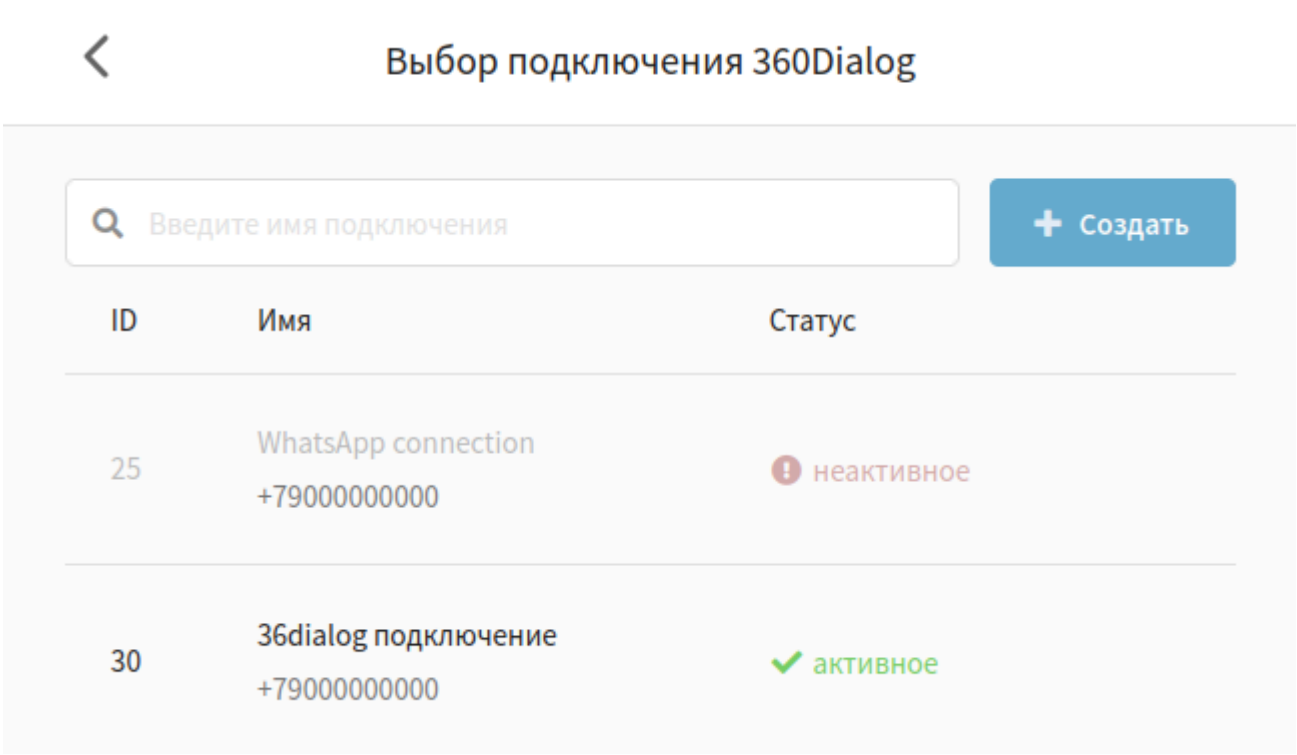
Не забудьте **сохранить интеграцию,** воспользовавшись кнопками в правом верхнем углу.

360dialog

Активируйте интеграцию с помощью переключателя в правом верхнем углу. Нажмите **“Добавить аккаунт”.**



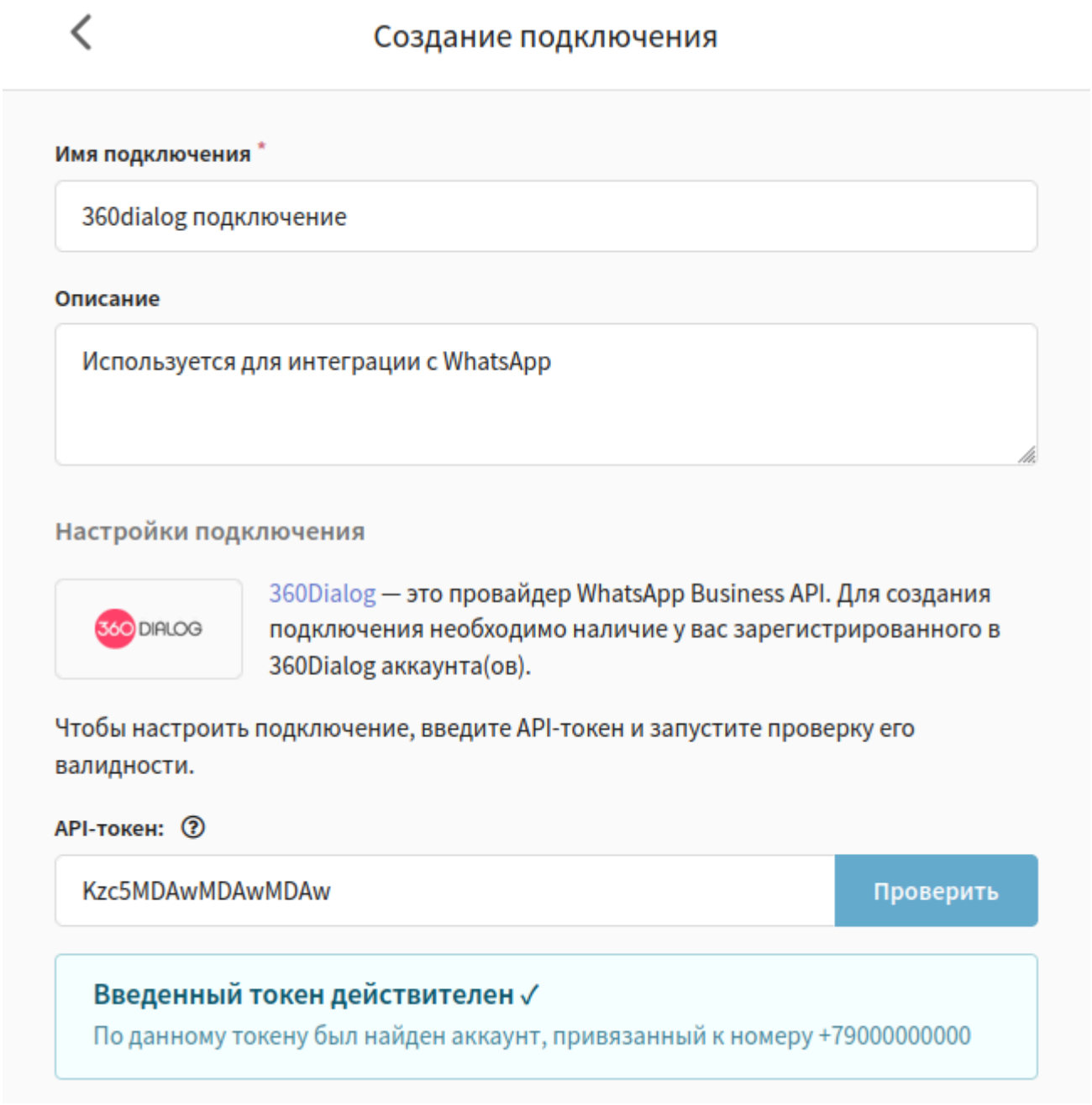
Выберите доступное **подключение** к 360dialog:



Нет доступных подключений

Если вы ещё не настраивали подключение к 360dialog, вы можете создать его непосредственно в этом окне.

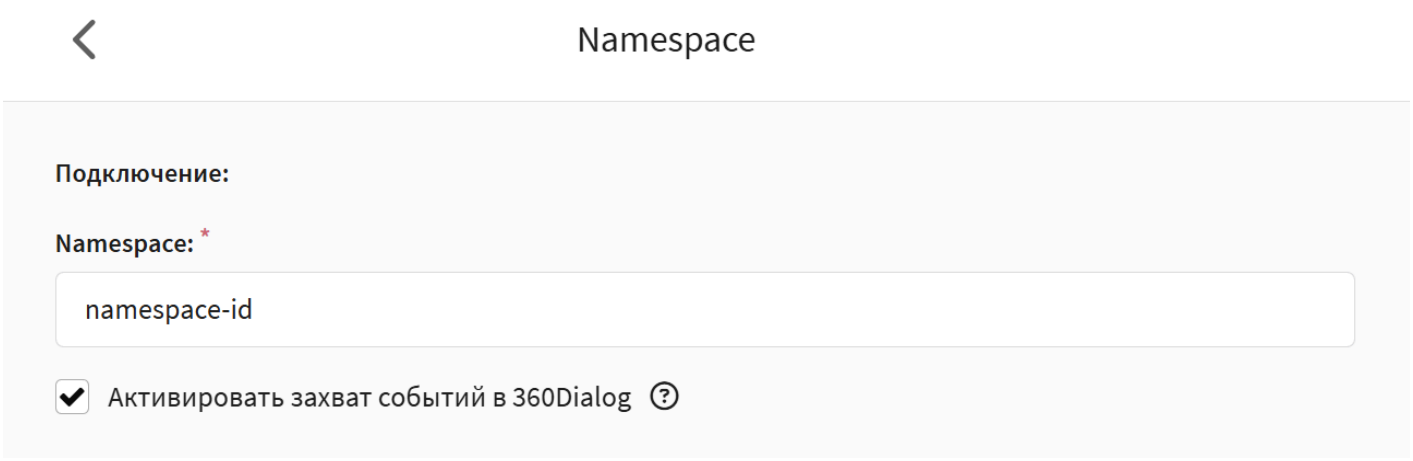
Нажмите **“Создать”**. Введите имя и описание подключения. Далее укажите API-токен, сгенерированный в личном кабинете 360dialog, и запустите проверку его валидности. Если токен прошёл проверку, нажмите кнопку **“Создать”** в правом нижнем углу.



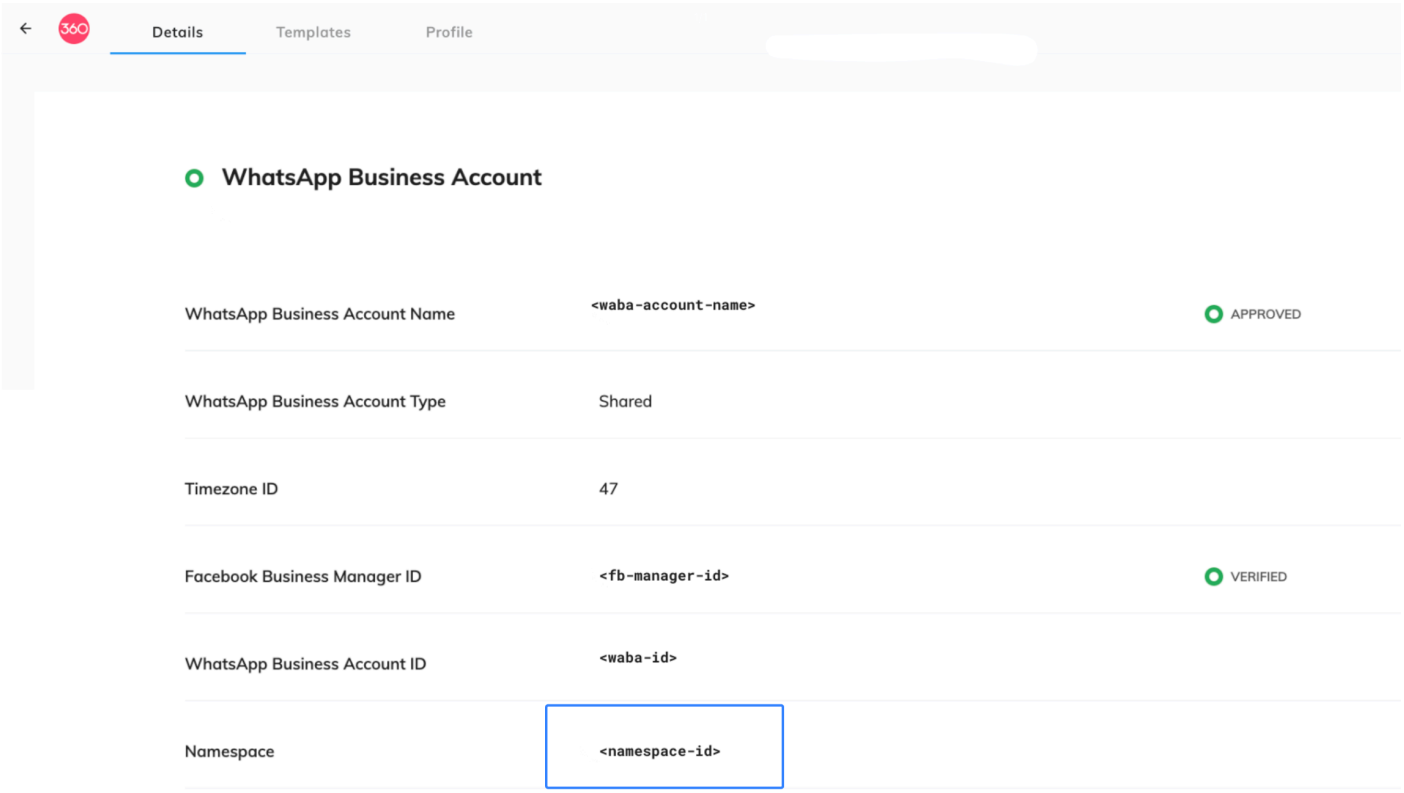
TIP

Подробнее о создании подключения можно прочитать здесь.

Далее заполните поле **“Namespace”** — это уникальная строка, который используется для идентификации всех шаблонов, созданных в рамках определенного WhatsApp Business аккаунта.



Namespace можно найти в личном кабинете 360Dialog:



При необходимости вы можете **включить захват событий в 360dialog** — это позволит передавать в Altcraft Platform данные о статусах сообщений. Если параметр активирован, после сохранения интеграции в личный кабинет 360dialog автоматически будет добавлен Callback URL, через который события WhatsApp передаются в Altcraft. Если отключить этот параметр, аналитика по доставкам и прочтениям будет недоступна в платформе.

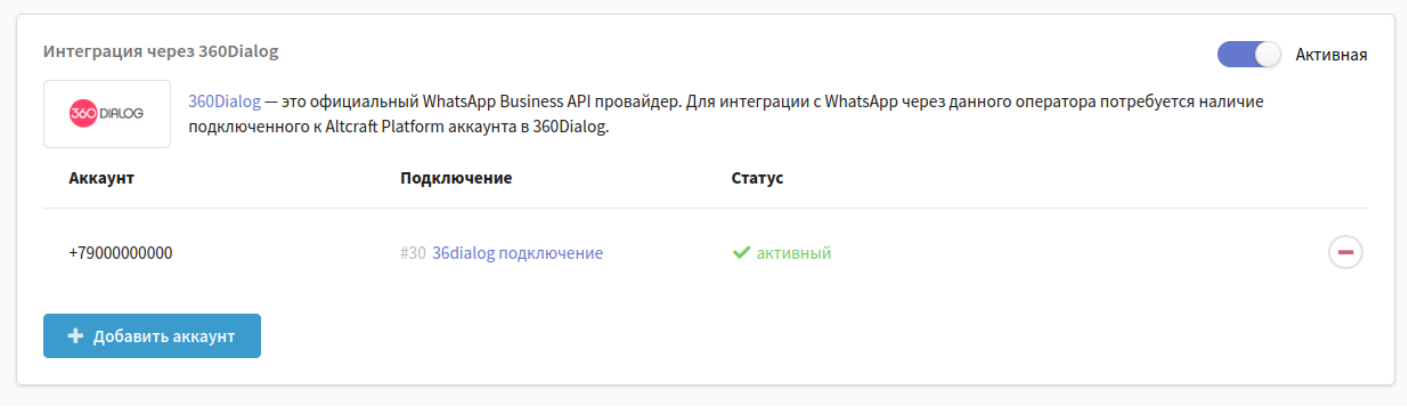
Нажмите **“Добавить”** в правом нижнем углу.

TIP

Подробнее о создании подключения можно прочитать здесь.

Данные по добавленному аккаунту отобразятся в интеграции. При необходимости вы можете добавить несколько аккаунтов.
Чтобы убрать аккаунт из интеграции, нажмите

.



Нажмите кнопку **“Сохранить”** в правом верхнем углу, чтобы сохранить настроенную интеграцию.

После настройки интеграции вы можете использовать её для добавления Viber™-канала в ресурс.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Viber™

Интеграция с Viber™ позволит автоматизировать коммуникацию с подписчиками в мессенджере. Настроив этот канал, вы сможете:

- отправлять подписчикам персонализированные сообщения в мессенджер;
- сегментировать аудиторию на основе поведения пользователей;
- использовать Viber™ канал для настройки сценария;
- собирать статистику и анализировать полученные данные.

Интеграция может быть настроена несколькими способами: при помощи **Devino Telecom**, **SMS Traffic**, **MTC OmniChannel** или **Rapporto**. Прежде чем настраивать интеграцию, вам необходимо:

1. Зарегистрировать аккаунт в выбранном вами сервисе.
2. Создать подключение, чтобы связать аккаунт с Altcraft Platform.

TIP

- Создание подключения через **Devino** описано в **этой статье**.
- Создание подключения через **SMSTraffic** описано в **этой статье**.
- Создание подключения через **MTC OmniChannel** описано в **этой статье**.
- Создание подключения через **Rapporto** описано в **этой статье**.

INFO

Стоимость сообщений определяется на стороне провайдера. Altcraft Platform не взимает дополнительной платы за отправку сообщений в Viber™.

Настройка интеграции

Откройте раздел **“Интеграции”** и выберите тип интеграции — **Viber™.** Перейдите на вкладку вашего провайдера:

Интеграция с Viber

Применить | Сохранить | ×

Devino telecom | SMS Traffic | MTS OmniChannel | Rapporto

Интеграция через Devino telecom

Активная

Devino Telecom — разработчик облачных платформ для общения бизнеса со своей аудиторией в цифровых каналах коммуникации. Для интеграции с Viber через данного оператора потребуется наличие подключенного к Altcraft Platform аккаунта в Devino Telecom.

+ Добавить подключение

Интеграция с Viber

Применить | Сохранить | ×

Devino telecom | SMS Traffic | MTS OmniChannel | Rapporto

Интеграция через SMS Traffic

Активная

SMS Traffic — разработчик облачных платформ для взаимодействия бизнеса со своей аудиторией в цифровых каналах коммуникации. Для интеграции с Viber через этого провайдера требуется наличие аккаунта в SMS Traffic, подключенного к Altcraft Platform.

+ Добавить подключение

Callback URL:

https://click. /fbp/v1/viber/1?acc=4h6NRD82Qgo%3D&integ_type=sms_traffic

Интеграция с Viber

Применить | Сохранить | ×

Devino telecom | SMS Traffic | MTS OmniChannel | Rapporto

Интеграция через MTS OmniChannel

Активная

MTC OmniChannel платформа для рассылки сообщений через различные цифровые каналы. Для интеграции с Viber через данного оператора потребуется наличие подключенного к Altcraft Platform аккаунта в MTC OmniChannel.

Параметр class — это опциональный параметр для управления кастомизированной маршрутизацией сообщений.

Этот параметр задается в сервисе MTC OmniChannel и принимает числовое значение. Введите число, соответствующее классу отправки сообщений в Viber.

Подключение	Статус	Лимит	Class	
#11 Viber connection	✓ активный	100		×

Интеграция с Viber

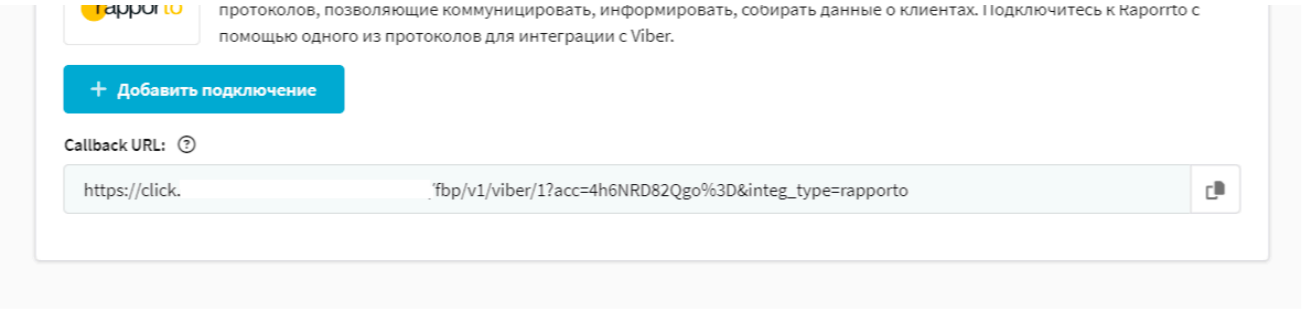
Применить | Сохранить | ×

Devino telecom | SMS Traffic | MTS OmniChannel | Rapporto

Интеграция через Rapporto

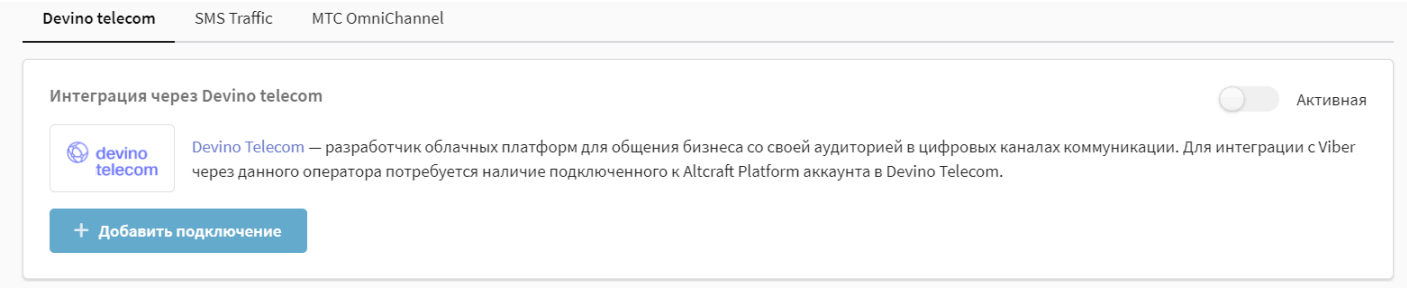
Активная

Rapporto — платформа для интеграции с мобильными каналами коммуникации. Для подключения доступны разные виды

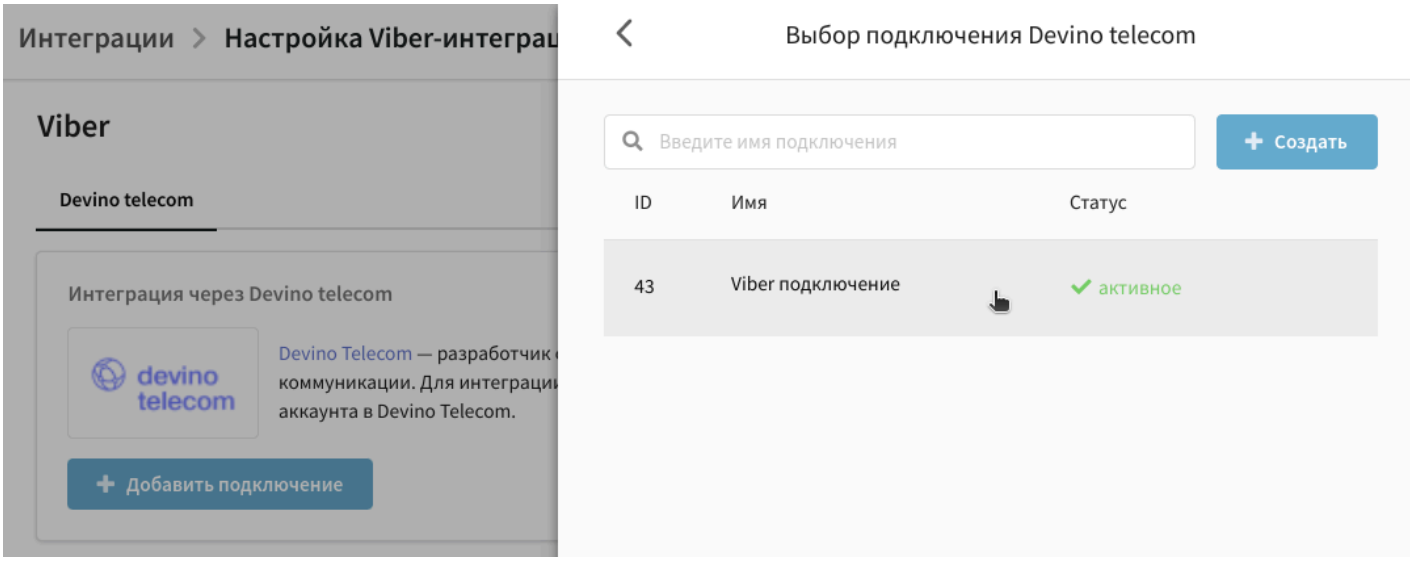


Devino

Активируйте интеграцию с помощью переключателя в правом верхнем углу. Нажмите **“Добавить подключение”:**



Справа появится окно, где необходимо выбрать доступное подключение:



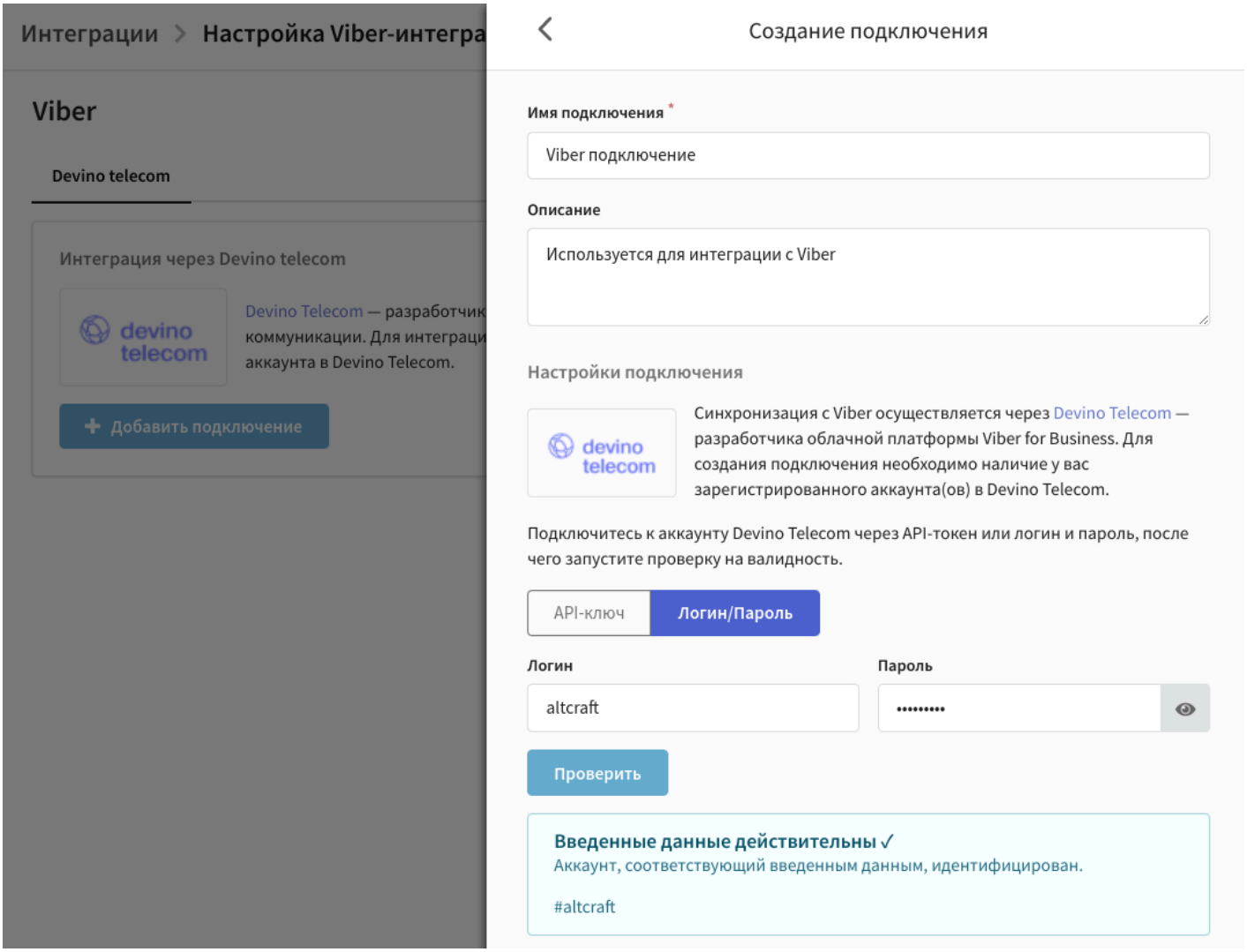
Нет доступных подключений

Если у вас ещё не настроено подключение к провайдеру, вы можете создать его непосредственно в этом окне.

1. Нажмите **“Создать”**.
2. Введите **имя** и описание подключения.
3. Введите данные для авторизации (для Devino доступно два варианта авторизации).
4. Нажмите **“Проверить”**, чтобы запустить проверку валидности введенных данных.
5. Если введенные данные прошли проверку, нажмите кнопку **“Создать”** в правом нижнем углу.

TIP

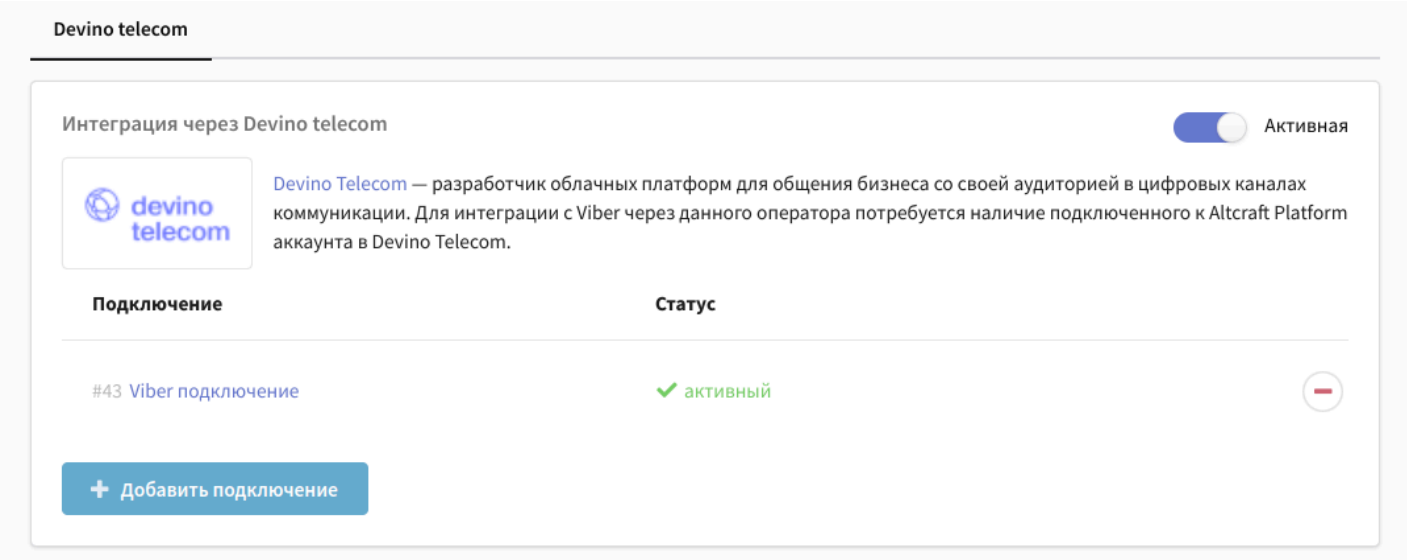
На портале документации опубликована подробная инструкция по созданию подключения через Devino Telecom.



Информация о добавленном подключении отобразится в интеграции. Убедитесь, что статус подключения — “активный”. В противном случае проверьте статус вашего аккаунта в личном кабинете провайдера. Имейте в виду, что аккаунт может быть заблокирован, если не внесена абонентская плата.

При необходимости вы можете добавить несколько аккаунтов. Чтобы убрать аккаунт из интеграции, нажмите

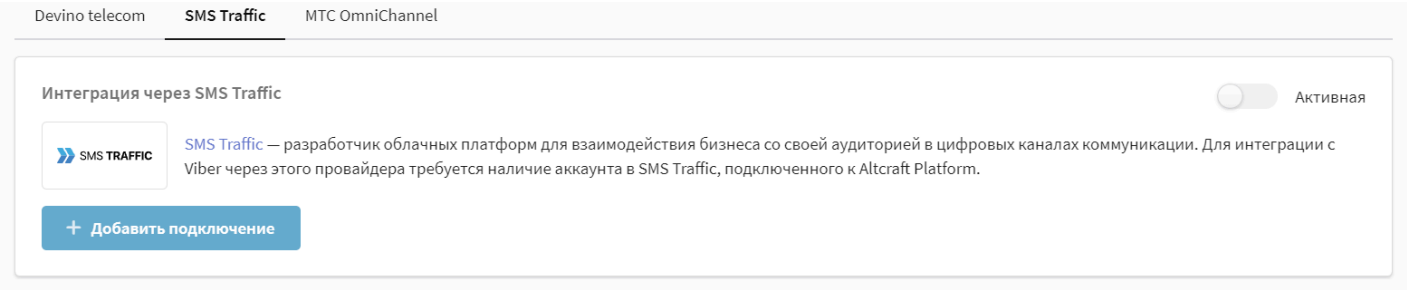
.



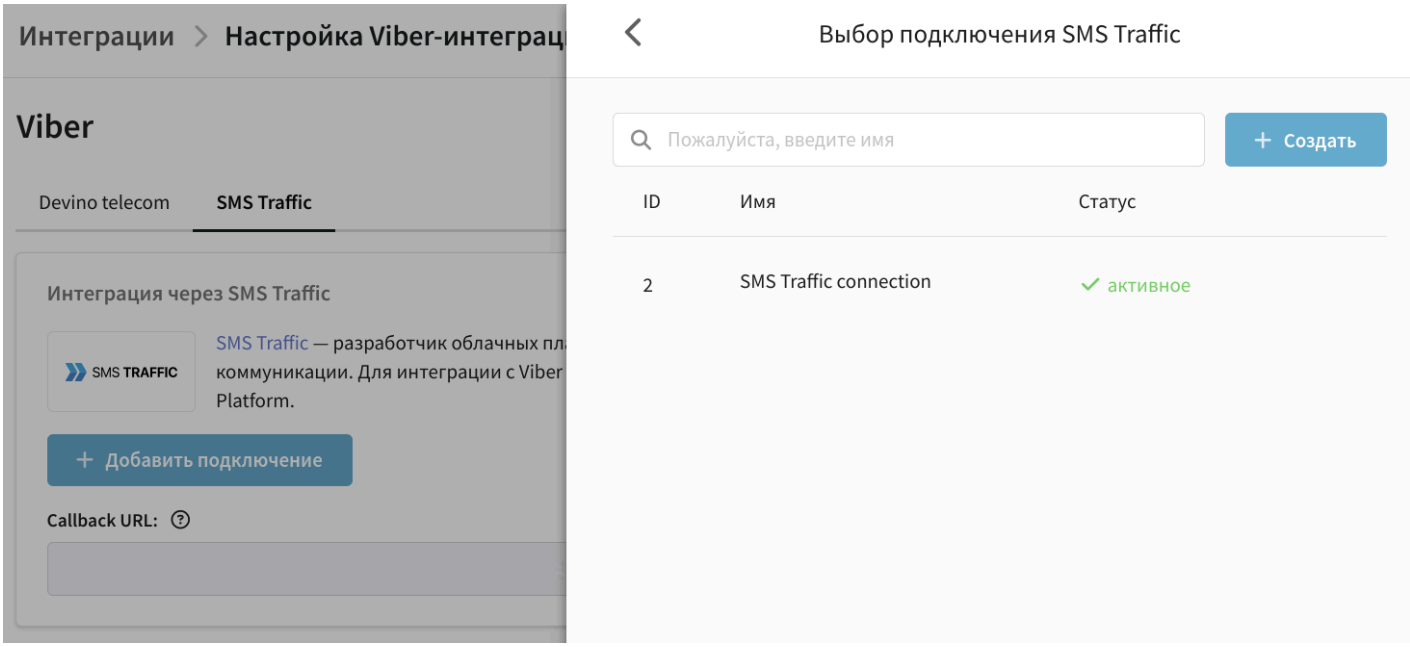
Сохраните интеграцию с помощью кнопок в правом верхнем углу.

SMSTraffic

Активируйте интеграцию с помощью переключателя в правом верхнем углу. Нажмите **“Добавить подключение”:**



Справа появится окно, где необходимо выбрать доступное подключение:



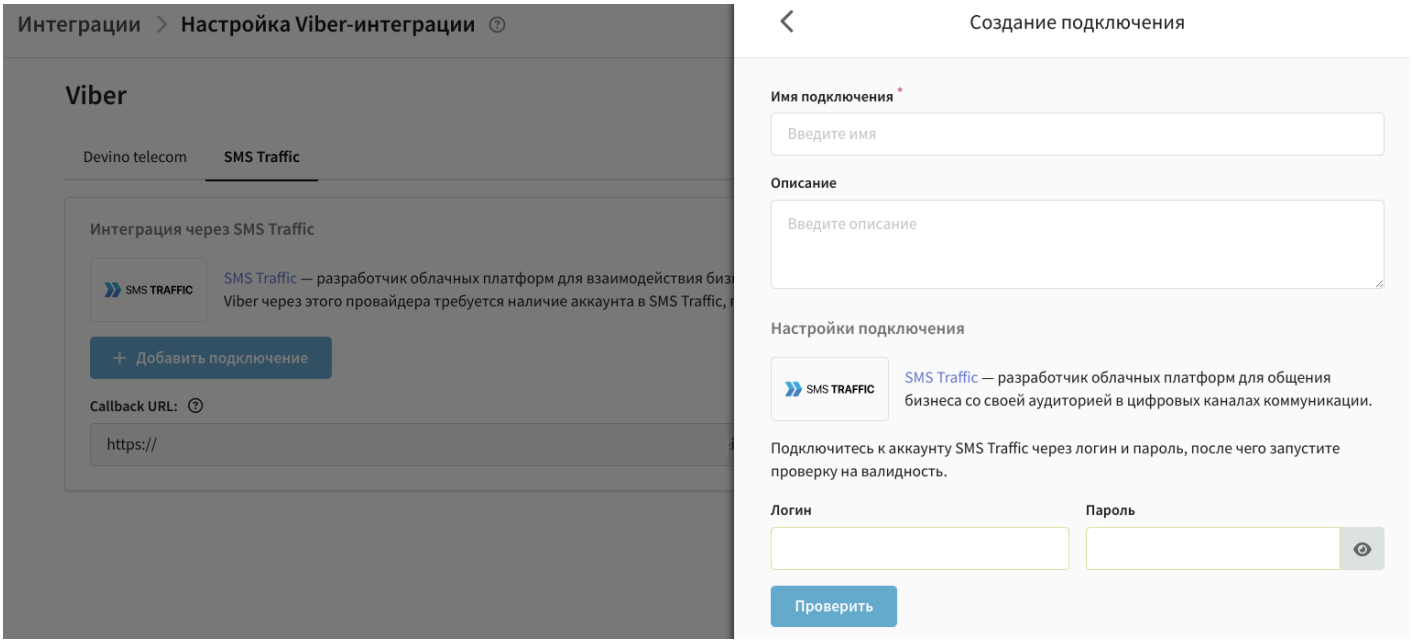
Нет доступных подключений

Если у вас ещё не настроено подключение к провайдеру, вы можете создать его непосредственно в этом окне.

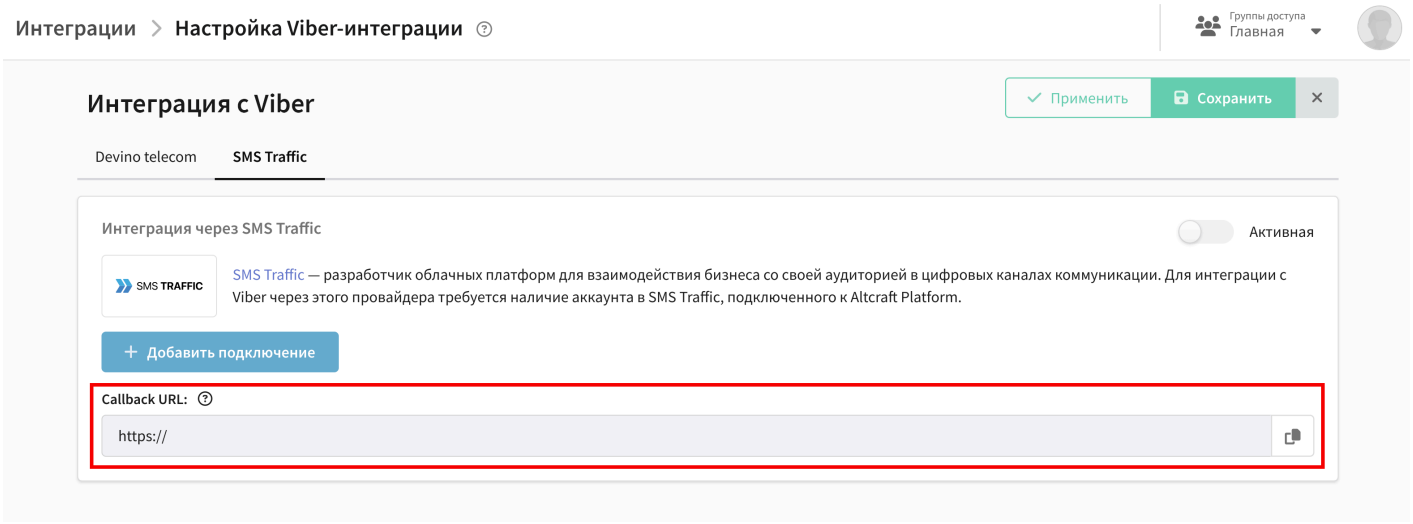
1. Нажмите **“Создать”**.
2. Введите **имя** и описание подключения.
3. Введите данные для авторизации.
4. Нажмите **“Проверить”**, чтобы запустить проверку валидности введенных данных.
5. Если введенные данные прошли проверку, нажмите кнопку **“Создать”** в правом нижнем углу.

TIP

На портале документации опубликованы подробные инструкции по созданию подключения через SMSTraffic.



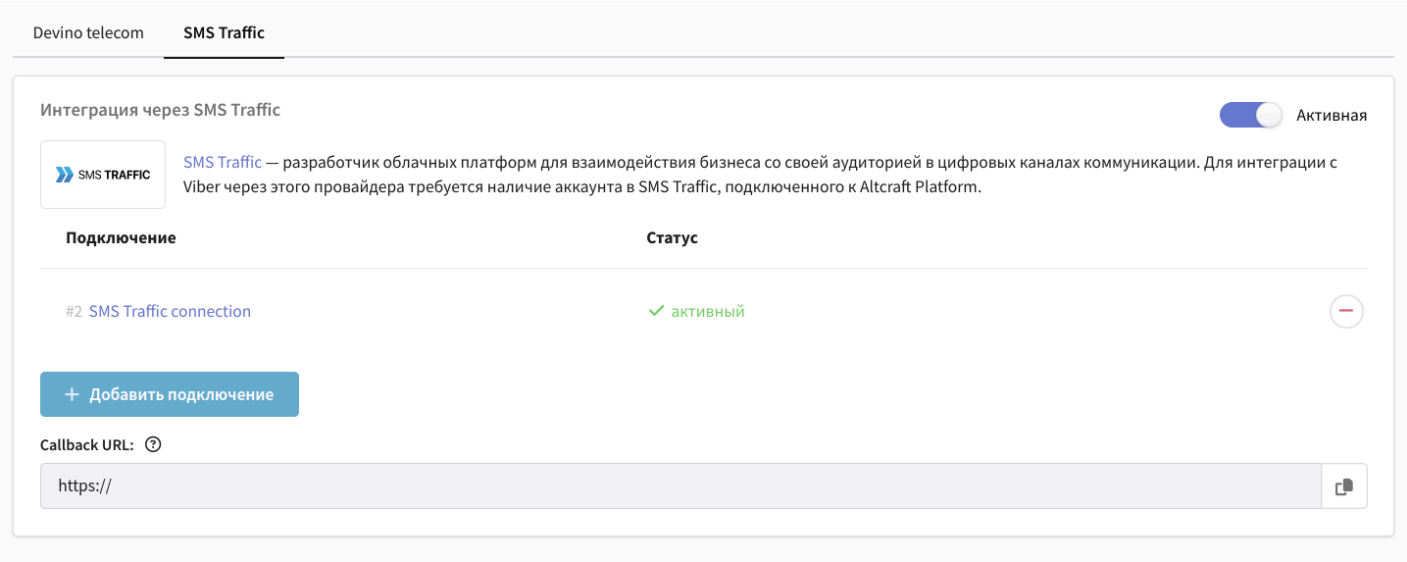
Для корректной работы интеграции вам также необходимо отправить в техническую поддержку ваш **Callback URL**, указанный в интеграции.



Информация о добавленном подключении отобразится в интеграции. Убедитесь, что статус подключения — “активный”. В противном случае проверьте статус вашего аккаунта в личном кабинете провайдера. Имейте в виду, что аккаунт может быть заблокирован, если не внесена абонентская плата.

При необходимости вы можете добавить несколько аккаунтов. Чтобы убрать аккаунт из интеграции, нажмите

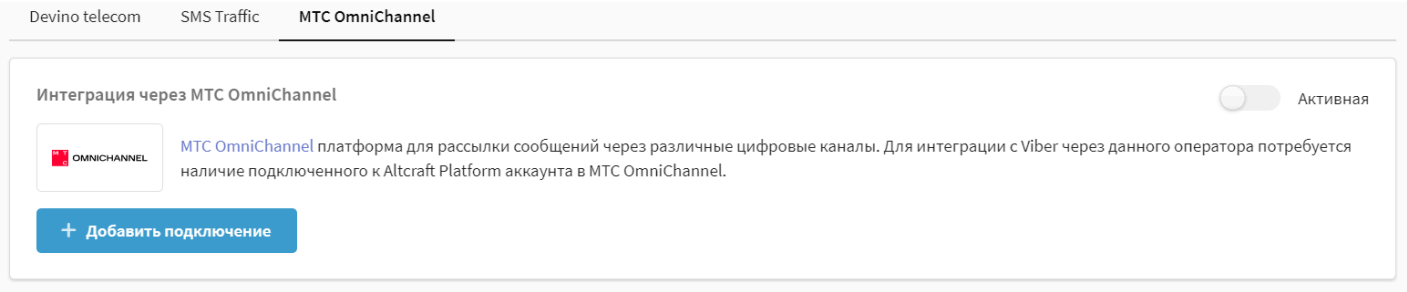
.



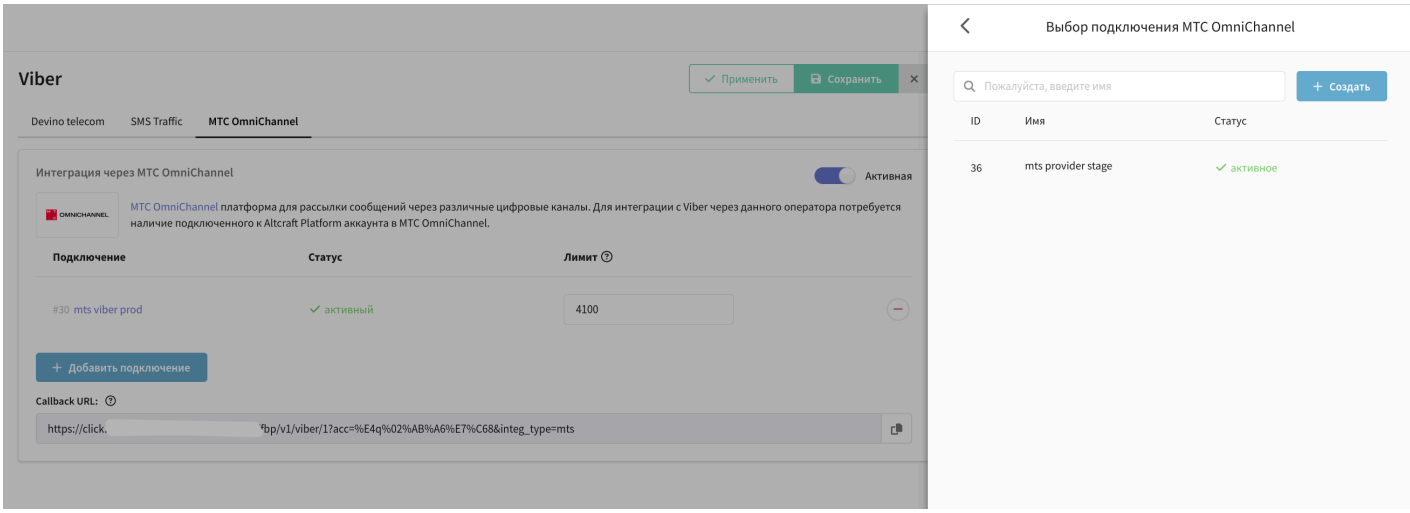
Сохраните интеграцию с помощью кнопок в правом верхнем углу.

MTC OmniChannel

Активируйте интеграцию с помощью переключателя в правом верхнем углу. Нажмите **“Добавить подключение”**:



Справа появится окно, где необходимо выбрать доступное подключение:



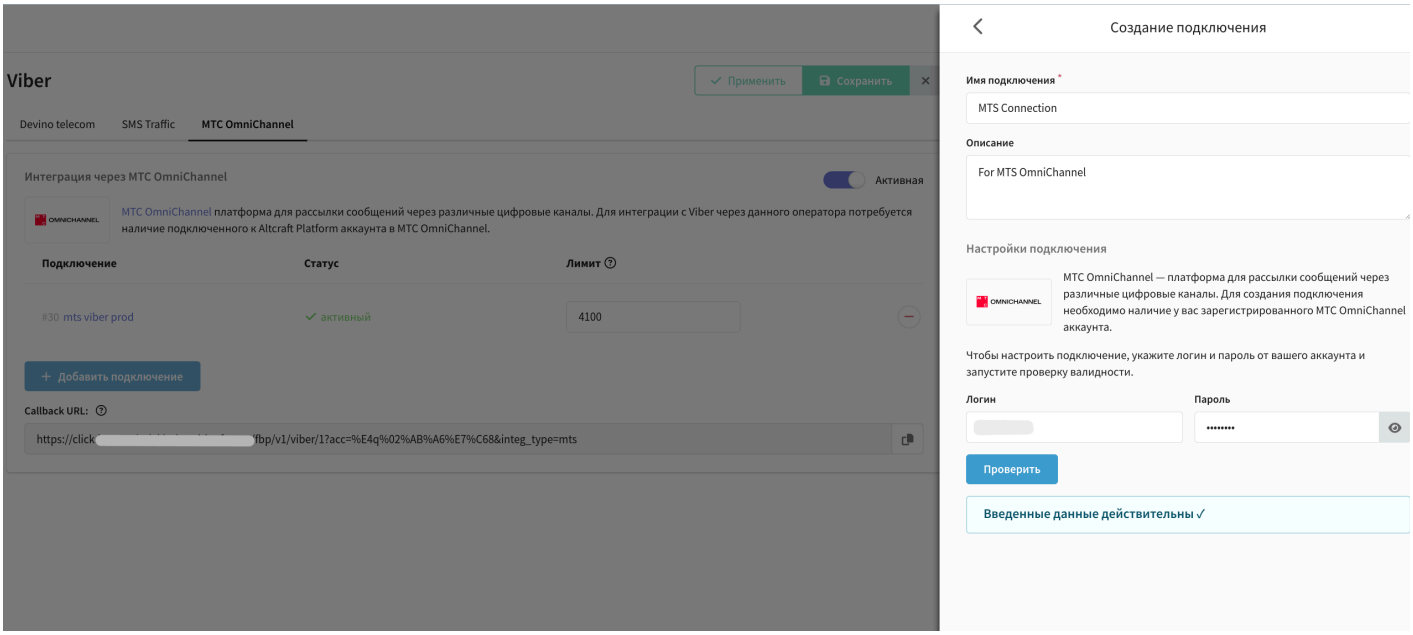
Нет доступных подключений

Если у вас ещё не настроено подключение к провайдеру, вы можете создать его непосредственно в этом окне.

1. Нажмите **“Создать”**.
2. Введите **имя** и описание подключения.
3. Введите данные для авторизации.
4. Нажмите **“Проверить”**, чтобы запустить проверку валидности введенных данных.
5. Если введенные данные прошли проверку, нажмите кнопку **“Создать”** в правом нижнем углу.

TIP

На портале документации опубликованы подробные инструкции по созданию подключения через MTC OmniChannel.

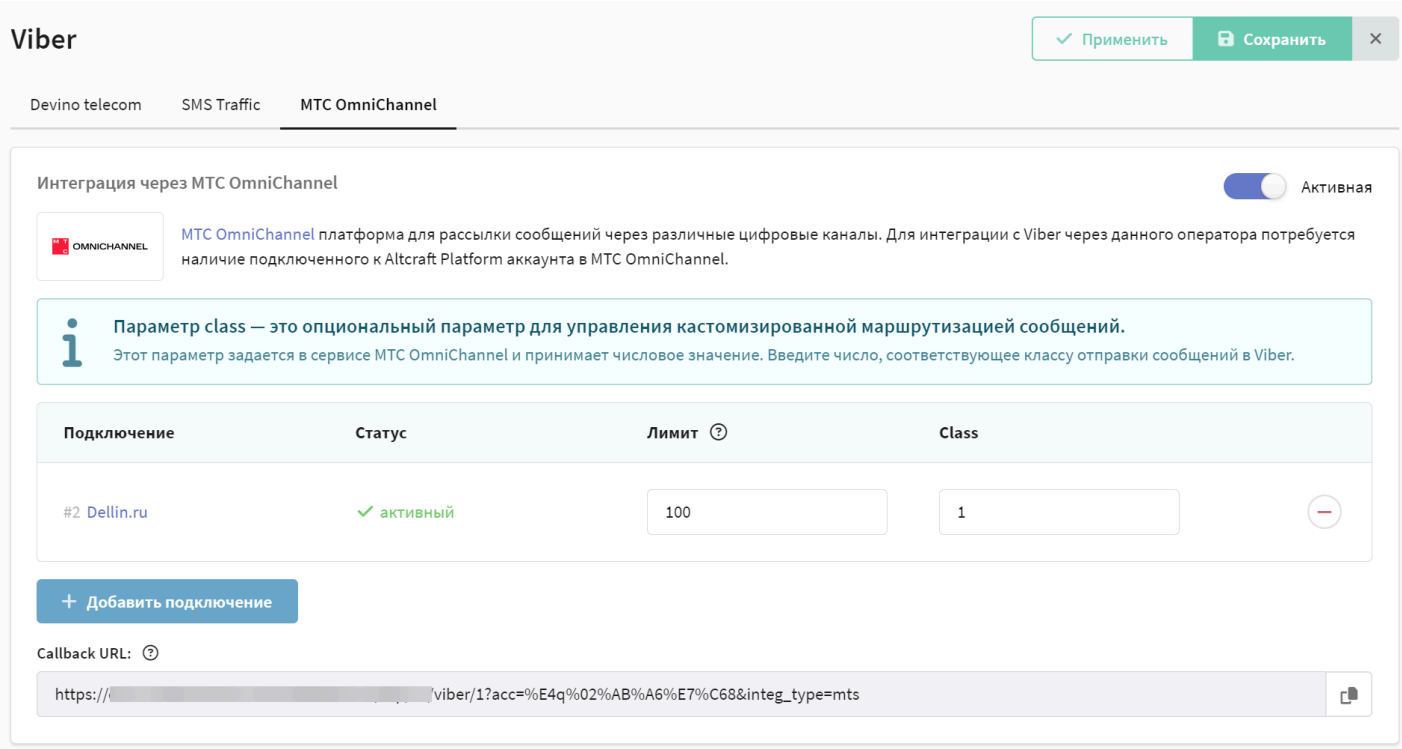


Для корректной работы интеграции вам также необходимо сообщить в техническую поддержку ваш **Callback URL**, указанный в интеграции.

Информация о добавленном подключении отобразится в интеграции. Убедитесь, что статус подключения — “активный”. В противном случае проверьте статус вашего аккаунта в личном кабинете провайдера. Имейте в виду, что аккаунт может быть заблокирован, если не внесена абонентская плата.

При необходимости вы можете добавить несколько аккаунтов. Чтобы убрать аккаунт из интеграции, нажмите

.



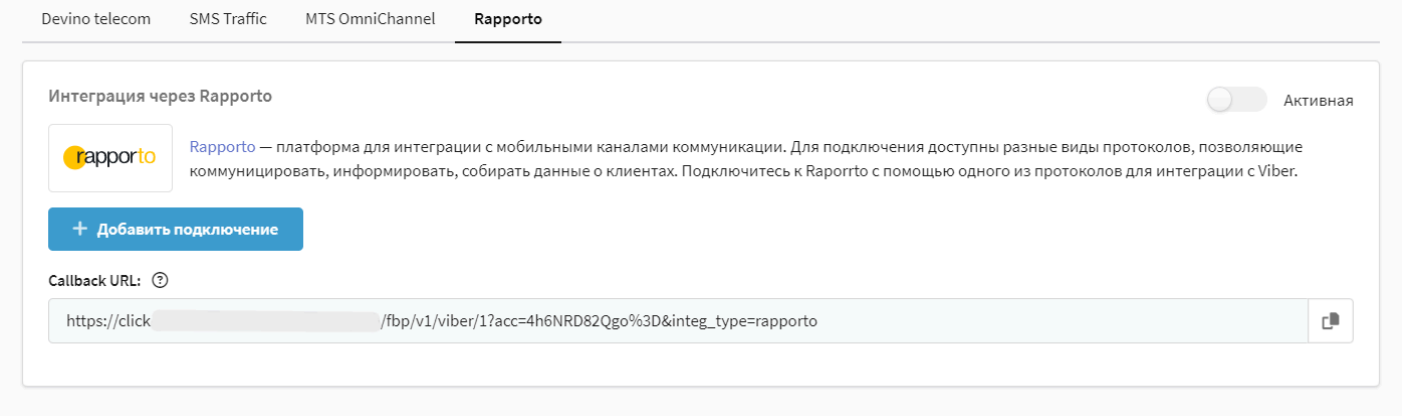
Укажите **лимит сообщений**, отправляемых в секунду (по умолчанию 0 — лимит отсутствует). Уточнить лимит можно у менеджера MTC OmniChannel.

В поле **“class”** укажите параметр маршрутизации сообщений. Этот параметр задается в сервисе MTC OmniChannel и принимает числовое значение. Введите число, соответствующее классу отправки сообщений в Viber™.

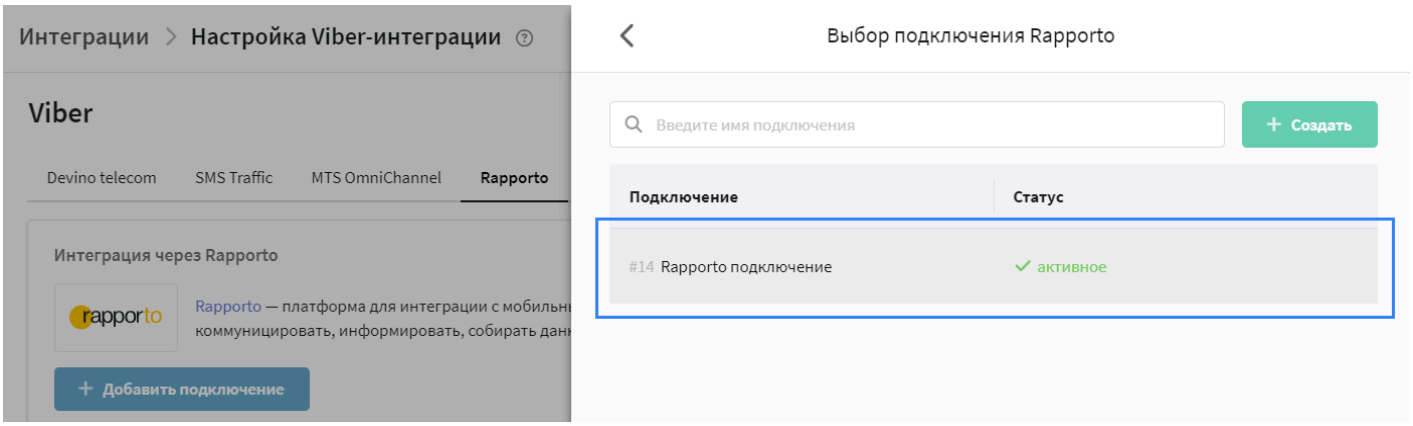
Сохраните интеграцию с помощью кнопок в правом верхнем углу.

Rapporto

Активируйте интеграцию с помощью переключателя в правом верхнем углу. Нажмите **“Добавить подключение”**:



Справа появится окно, где необходимо выбрать доступное подключение:



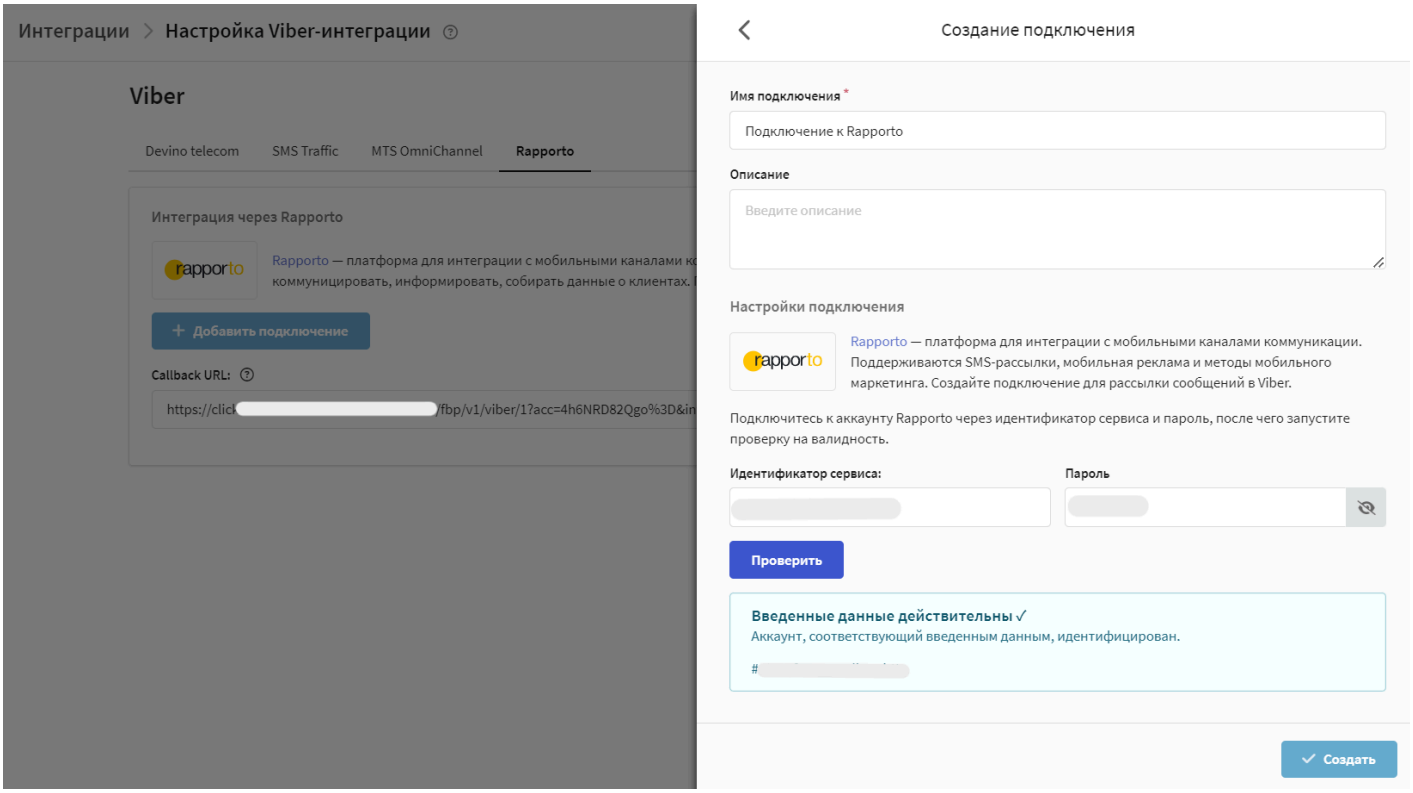
Нет доступных подключений

Если у вас ещё не настроено подключение к провайдеру, вы можете создать его непосредственно в этом окне.

1. Нажмите **“Создать”**.
2. Введите **имя** и описание подключения.
3. Введите данные для авторизации.
4. Нажмите **“Проверить”**, чтобы запустить проверку валидности введенных данных.
5. Если введенные данные прошли проверку, нажмите кнопку **“Создать”** в правом нижнем углу.

TIP

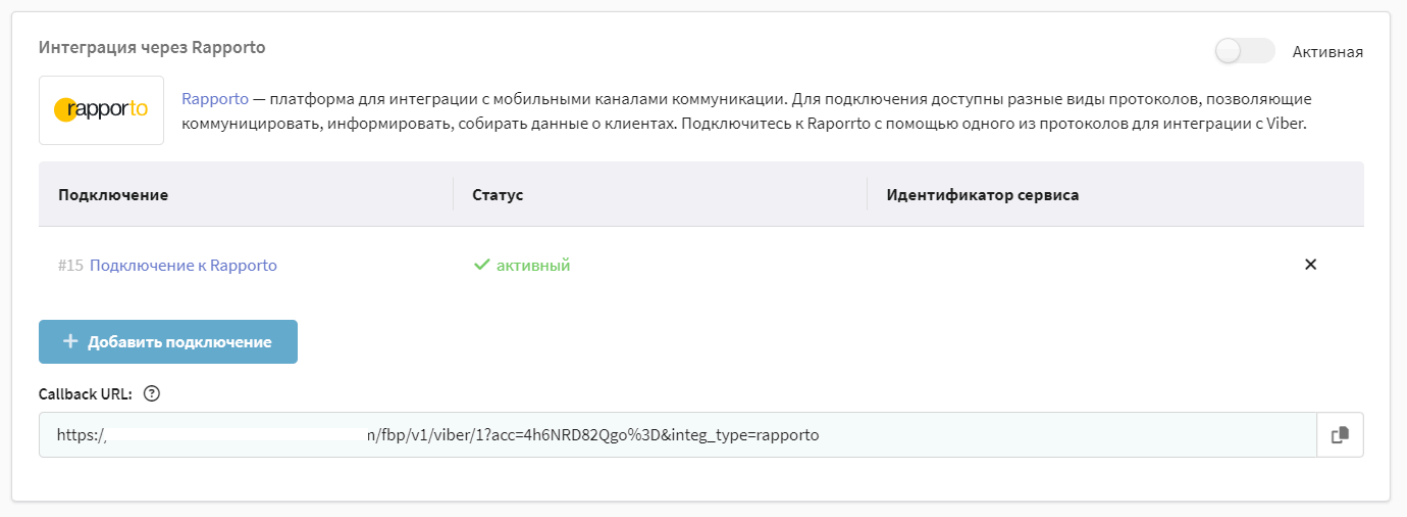
На портале документации опубликованы подробные инструкции по созданию подключения через Rapporto.



Для корректной работы интеграции вам также необходимо сообщить в техническую поддержку ваш **Callback URL**, указанный в интеграции.

Информация о добавленном подключении отобразится в интеграции. Убедитесь, что статус подключения — “активный”.

При необходимости вы можете добавить несколько аккаунтов. Чтобы убрать аккаунт из интеграции, нажмите на крестик.



Сохраните интеграцию с помощью кнопок в правом верхнем углу.

Завершение настройки

После настройки интеграции вы можете использовать её для добавления Viber™-канала в **ресурс**.

---

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Tilda™

Интеграция с Tilda для импорта профиля клиента

В Altcraft Platform можно передавать данные клиента, который заполнил форму на сайте, сконструированном в сервисе Tilda. Для этого необходимо воспользоваться инструментом Webhooks. Подробнее о Webhooks в Tilda вы можете прочитать в этой статье.

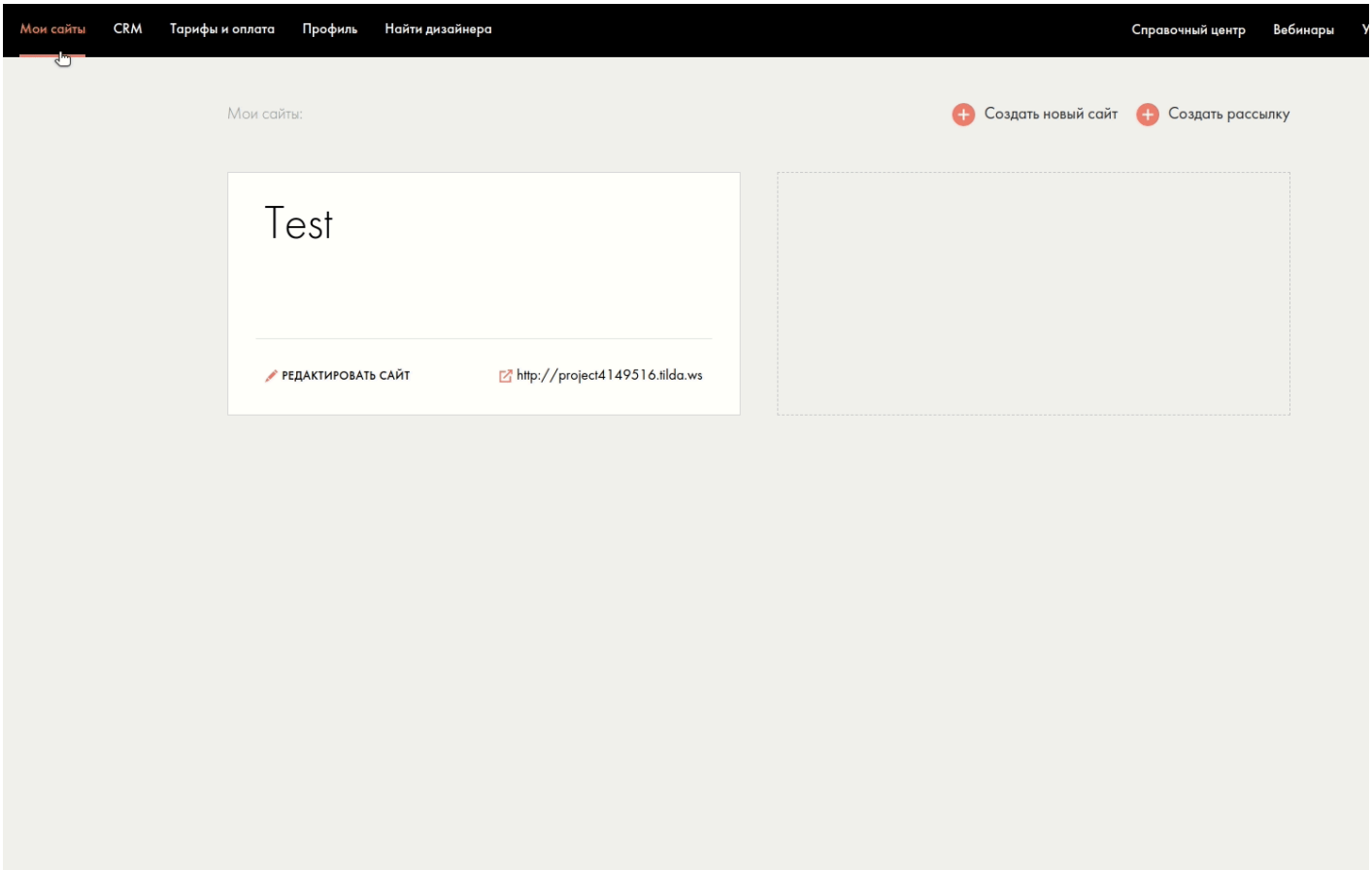
> **NOTE**
>
> Обратите внимание, что интеграция с Tilda позволяет интегрировать только подписки пользователей на email и SMS ресурсы.

Для успешной интеграции вам потребуется следующая информация.

- URL панели Altcraft, например: `https://ru.altcraft.com`.
- API-ключ (API-токен), имеющий права на отправку данных в Altcraft. Подготовить такой токен сможет пользователь вашего аккаунта с мастер-правами.
- Идентификаторы базы данных и ресурса. Если необходимости подписывать на ресурс нет, то только базы данных.

Шаг 1. Создайте Webhook в Tilda

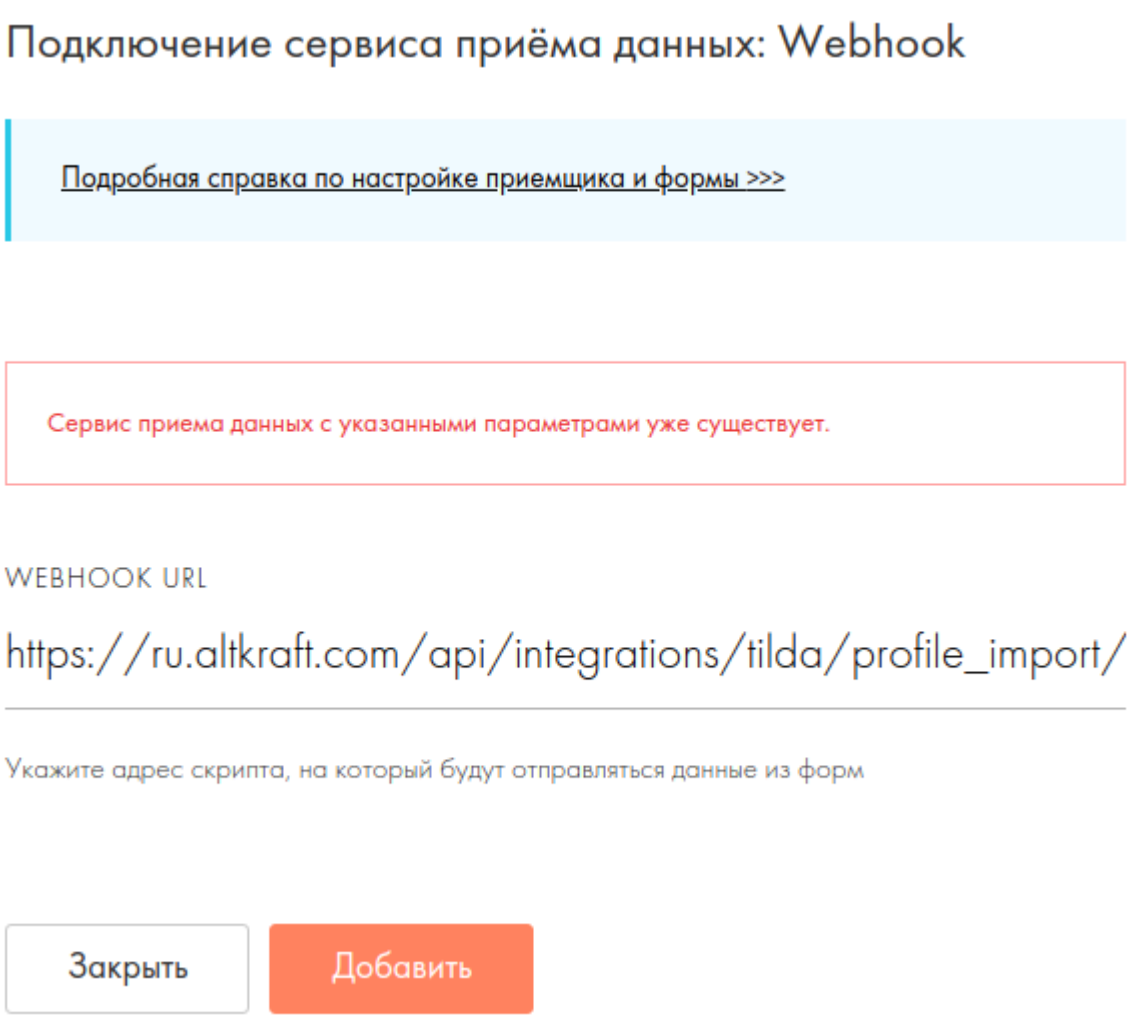
Перейдите в раздел: **Мои сайты —> Сайт —> Настройки сайта —> Форма.** В самом низу вы найдете **Другое —> Webhook**.



Заполните поле **WEBHOOK URL** следующим образом:

`<your.domain>/api/integrations/tilda/profile_import/?token=<api_token>`

Здесь вам надо заменить `<your.domain>` на URL панели Altcraft, а `<api_token>` на предоставленный вам токен:

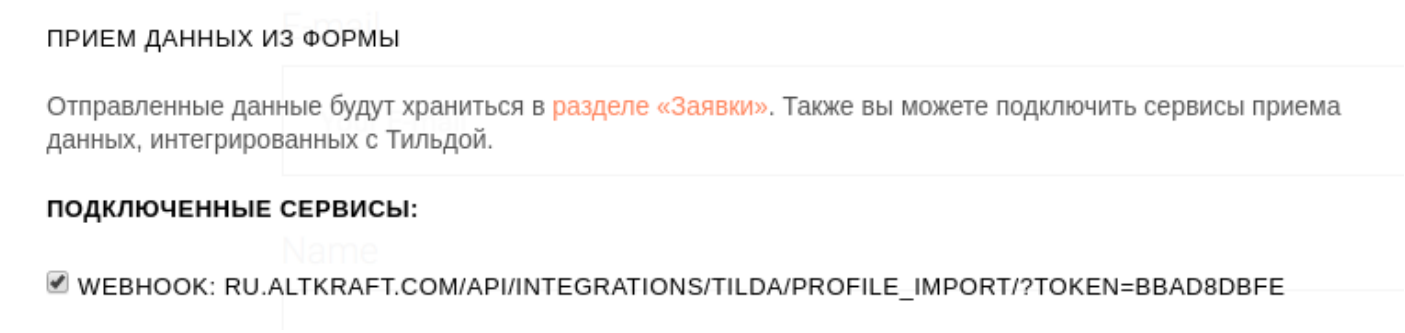


Нажмите **“Добавить”**. После этого вам предложат добавить приемщик данных ко всем формам на сайте. Пропустите этот шаг, кликнув кнопку **“Закрыть”**. Итак, Webhook создан.

Шаг 2. Настройте форму в Tilda

Откройте окно редактирования страницы сайта и найдите форму, данные из которой хотите передать в Altcraft. Затем откройте вкладку **“Контент”:**

Найдите раздел “**ПРИЕМ ДАННЫХ ИЗ ФОРМЫ**” и выберите ваш Webhook:



В Tilda существует два типа формы:

- простая форма с кнопкой,
- форма с несколькими полями.

Настройки для каждой из них несколько отличаются.

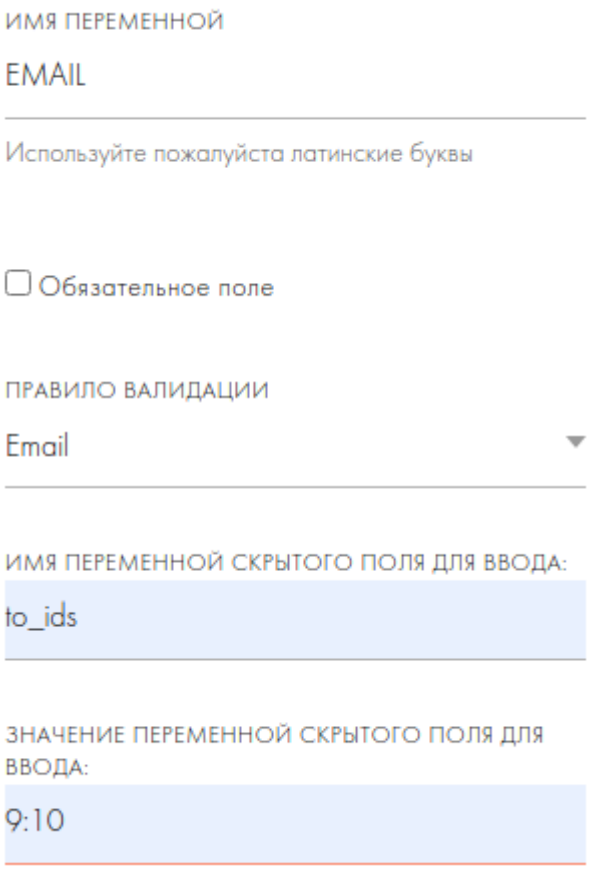
Простая форма

Если у вас форма с одним полем, то перейдите в **Настройки** вашей простой формы — кнопка слева от **Контента**.

Затем проскрольте настройки, пока не найдете раздел с полями. Здесь нужно обратить внимание на имя переменной в каждом из полей.

Поле	Описание
Имя переменной	Если вы хотите, чтобы это поле попало в Altcraft, то его имя должно соотвествовать такому же полю в базе данных. Имя переменной должно быть email или phone. Имя не чувствительно к регистру, например, если в БД Altcraft есть поле “email”, а в Tilda поле называется “EMAIL”, то данные из формы все равно будут переданы успешно.
Имя переменной скрытого поля для ввода	Назовите его `to_ids`
Значение переменной скрытого поля для ввода	Введите через двоеточие идентификатор базы данных и идентификатор ресурса. Если на ресурс подписывать не нужно, то только идентификатор базы данных. На скриншоте ниже, в примере указан `9` — идентификатор БД, `10` — идентификатор ресурса.

Пример заполнения полей:



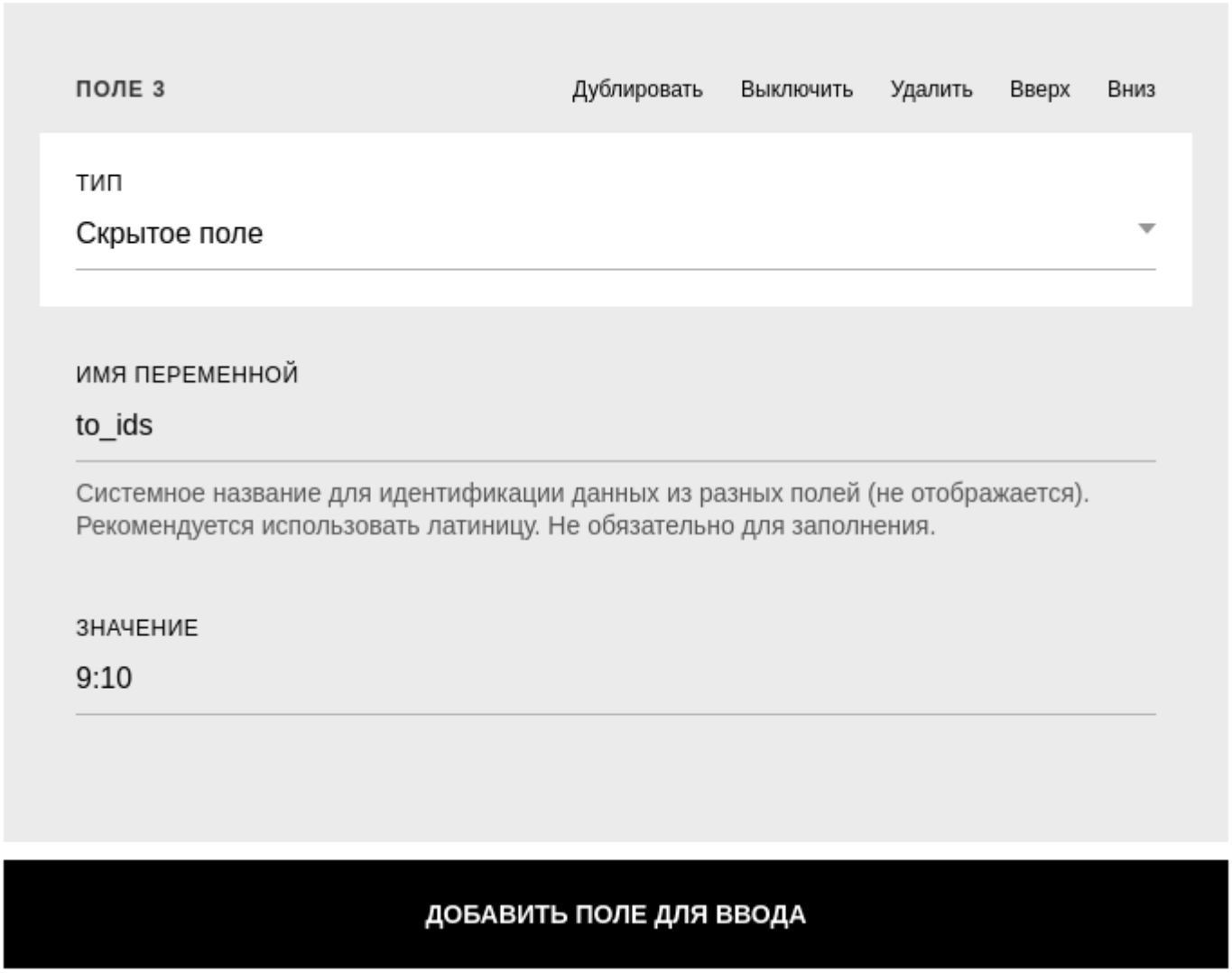
Форма с несколькими полями

Если вы работаете с формой, которая содержит несколько полей, то вам необходимо открыть вкладку **“Контент”**. Там разверните вкладку **“ПОЛЯ ДЛЯ ВВОДА”.**

Обратите внимание на имена всех переменных, которые содержит форма. Если вы хотите, чтобы поле попало в Altcraft, его имя должно соответствовать такому же полю в базе данных. Имя переменной должно быть email или phone. Имя переменной не чувствительно к регистру, например, **EMAIL** тоже будет работать.

После настройки видимых полей формы, добавьте еще одно поле. Тип — **скрытое**. Назовите его `to_ids`. Введите через двоеточие идентификатор базы данных и идентификатор ресурса. Если на ресурс подписывать не нужно, введите только идентификатор базы данных.

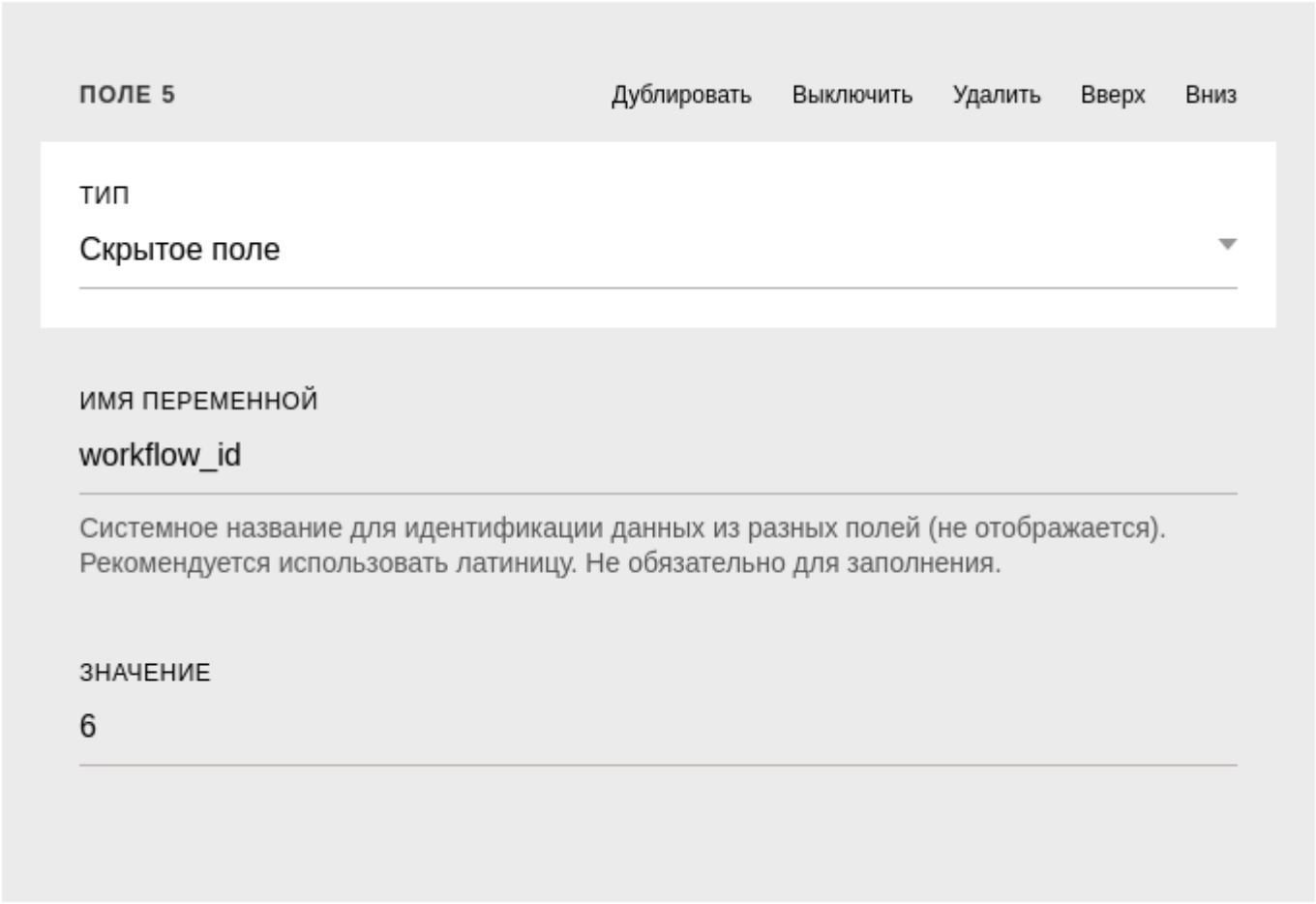
В примере на скриншоте ниже указан `9` — идентификатор БД, `10` — идентификатор ресурса.



Интеграция с Tilda позволяет не только импортировать новые профили в базы данных Altcraft, но и сразу запустить триггерное событие. Так, вы можете запустить триггерную рассылку сразу после импорта нового профиля, чтобы отправить ему приветственный email, или добавить профиль в сценарий приветственной цепочки.

Для этого вам надо добавить дополнительные скрытые поля в форме:

Действие после импорта	**Имя переменной**	**Значение**
Запустить триггерную рассылку	trigger_id	Идентификатор триггерной рассылки
Добавить профиль в сценарий	workflow_id	Идентификатор сценария



Передача статуса подписки

Когда вы импортируете профиль из формы с несколькими полями, вы можете передать определенный статус подписки на ресурс. Именно этот статус отобразится в цифровом профиле подписчика в платформе Altcraft.

Для передачи статуса профиля вам необходимо добавить еще одно скрытое поле в настройках вашей формы. В качестве **имени переменной** необходимо указать `status_sub`, а в качестве **значения** выбрать один из статусов, зарезервированных платформой Altcraft.

Доступные статусы:

Статус	**Значение поля “status_sub”**
Подписан	0
Отписан	1
Жалобщик	2
Hardbounced	3
Не подтвержден	4
Приостановлен	5
Не валиден	6

Шаг 3. Протестируйте форму

После того как вы настроите поля для формы, нажмите “**Сохранить и закрыть**”. Опубликуйте страницу. После публикации протестируйте лендинг, отправив в форму тестовые данные.

> **TIP**
>
> Tilda не сразу обновит форму, поэтому лучше тестировать ее через пару минут после публикации.

Затем откройте **Ваш проект —> Заявки из форм.**

Date	Email	Data
15/08/2018 11:35	qeeqawewq@sdfsdfasfd.cc	to_ids: 27:4

На скриншоте пример успешно отправленной заявки. Если вы видите ошибку, необходимо проверить следующие моменты:

- URL в Webhook прописан верный и содержит верный API Ключ
- База данных существует и она выбрана в ресурсе, который вы используете для подписки.
- В форме присутствует поле email или phone

Дополнительно

Вы также можете передавать поле **matching**, как и в других API запросах на импорт, для уточнения способа идентификации профиля. По умолчанию используется email **матчинг**, который осуществляет поиск и по email профиля, и по email из подписок. При использовании **matching** типа **custom** необходимо передать поле, по которому осуществляется поиск в **field_name**.

Вместо использования `to_ids`, можно применять поля `db_id` и `resource_id`. Поля можно прописать в Webhook URL заранее, если вам необходимо с нескольких форм отправлять данные в одну базу данных или подписывать на один ресурс.

Любое поле должно соответствовать полю в БД, если хотите, чтобы оно было сохранено, в противном случае оно будет проигнорировано. Например поля `name` может не быть в базе данных, чтобы передать имя в БД Altcraft вам следует назвать поле `_fname` или добавить в базу данных поле `name`.

По умолчанию идентификатор формы передается и сохраняется в поле **Поставщик** (`_vendor`). Если вы хотите изменить это поведение, то добавьте скрытое поле `_vendor` с необходимым значением, тогда идентификатор формы будет проигнорирован. В системе также доступны отчеты по полю **Поставщик**.

При обновление в БД поля профиля с типом “теги” поле полностью перезаписывается. Чтобы добавить новые значения к уже существующим, добавьте инструкцию `-add` к имени переменной при настройке поля в Tilda. Например, для обновления поля `tag_field`, используйте имя переменной `tag_field-add`.

ПОЛЕ 10 Дублировать Выключить Удалить Вверх Вниз

ТИП

Скрытое поле

ИМЯ ПЕРЕМЕННОЙ

tag_field-add

Системное название для идентификации данных из разных полей (не отображается). Рекомендуется использовать латиницу. Не обязательно для заполнения.

ЗНАЧЕНИЕ

tag1, tag2

Yandex AppMetrica™

Yandex AppMetrica™ — платформа аналитики для мобильных приложений с возможностью отправки push уведомлений на устройства пользователей.

Сайт AppMetrica™: https://appmetrica.yandex.ru

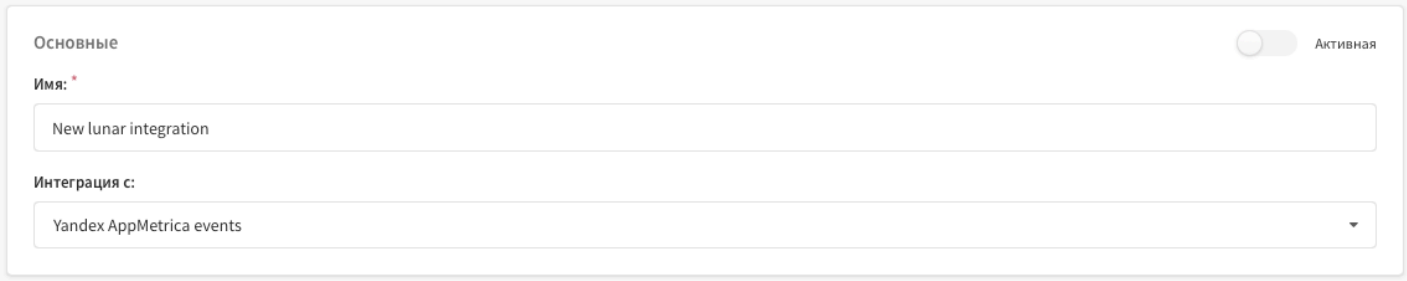
Altcraft MP может захватывать **события** и **установки** в AppMetrica™ и использовать их для:

- регистрации целевых действий и их ценности
- импорта или обновления профилей подписчиков

Чтобы интегрировать данные AppMetrica в Altcraft, в главном меню перейдите в раздел **Интеграции**.

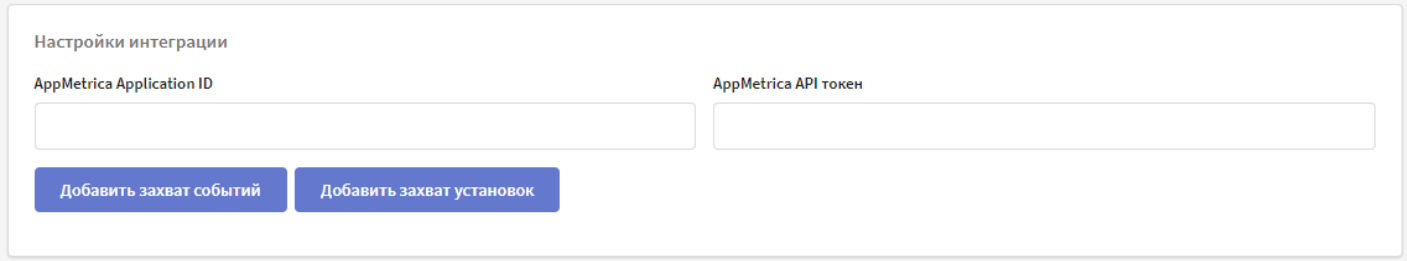
Подключение к API AppMetrica™

Создайте новую интеграцию, в основных настройках укажите произвольное название и в выпадающем меню выберите **Yandex AppMetrica events**. Здесь же находится кнопка активации/деактивации захватов событий:



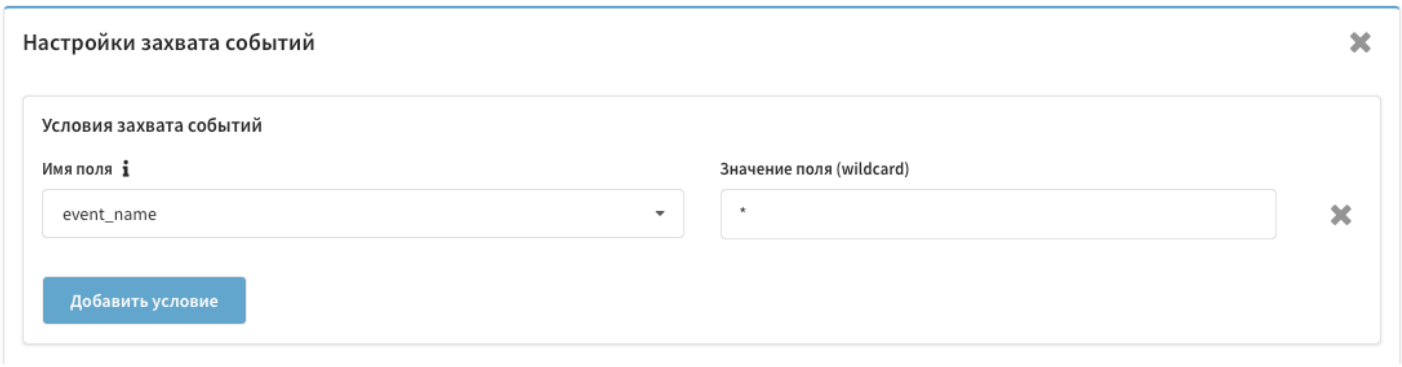
В настройках интеграции необходимо указать следующие параметры:

- **Application ID** — уникальный числовой идентификатор приложения в AppMetrica™.
- **AppMetrica API token** — токен для получения доступа к данным. Получить его можно в разделе **Настройки** приложения в AppMetrica™.



После внесения этих настроек вы сможете добавлять в интеграцию захваты **событий** и захваты **установок** приложения. Для каждого из них можно задать определённые условия срабатывания, основанные на значении полей AppMetrica™.

Доступна фильтрация по части значения поля, при этом изменяющаяся часть заменяется символом шаблона поиска `*` (wildcard). Например, значение `*@mail.ru` будет срабатывать на все адреса доменной группы mail.ru.



Поля события, получаемые из Yandex AppMetrica™

Интеграция поддерживает все поля, доступные в AppMetrica Logs API. Полный список передаваемых полей с описанием доступен по ссылке в документации AppMetrica™:

- **Для захатов событий** — https://appmetrica.yandex.ru/docs/mobile-api/logs/ref/events.html
- **Для захватов установок** — https://appmetrica.yandex.ru/docs/mobile-api/logs/ref/installations.html

INFO

Дополнительные параметры, передаваемые в **event_json**, необходимо вводить в **Имя поля** вручную, разделяя уровни вложенности данных точкой. Например:

- **event_json.email** — если поле email передаётся в AppMetrica в основной структуре “event_json”:

```
"event_json" :
{"email" : "example@example.com"}
```

- **event_json.contacts.email** — если поле включено в подструктуру “contacts”:

```
"event_json" :
{"contacts":
{"email" : "example@example.com",
 "phone" : "+12345678910"}
}
```

При передаче событий приложения в интерфейс платформы AppMetrica доступно 5 уровней вложенности, при передаче в API AppMetrica — 10. Altcraft поддерживает максимальное значение.

Документация по отправке собственных событий из приложения в AppMetrica: https://appmetrica.yandex.ru/docs/mobile-events/concepts/events.html

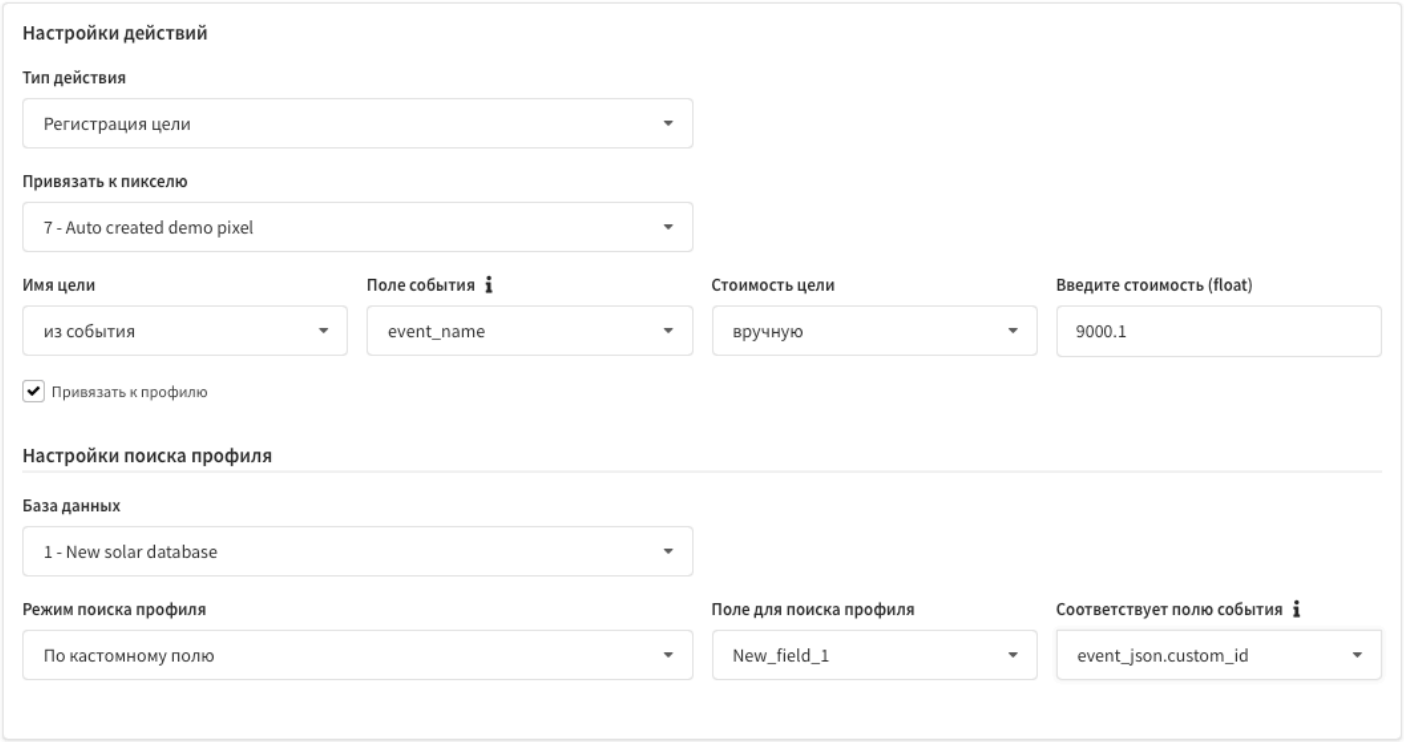
Регистрация цели

Событие, переданное в AppMetrica™ из мобильного приложения может регистрировать в Альткрафт достижение целевого действия и его ценность. Для этого выберите **Тип действия: Регистрация цели.**

Цели в Альткрафт организованы по Пикселям. Регистрируя одинаковые цели по нескольким пикселям, можно оценить эффективность разных вариантов интерфейса приложения или определённых последовательностей действий пользователей. Выберите пиксель в выпадающем меню.

Имя цели и её стоимость можно указать вручную, или записать значения из данных события.

- Значения, введённые вручную, статичны. При соответствии захвата событий условиям будут регистрироваться предопределённые значения.
- Значения из полей события зависят от конкретного действия в приложении и передают динамические данные.



Каждая регистрируемая цель может быть связана или не связана с определённым профилем клиента в Альткрафт:

- Анонимные цели доступны только в **отчётах по достижению целей**, формируя статистику по неизвестным профилям.
- Цели, связанные с профилем, могут использоваться для RFM и LTV сегментации пользователей, начисления баллов и управления лояльностью. Достижения целей отображаются в разделе **История действий** профиля клиента.

Чтобы записать цель определённому пользователю, включите опцию **Привязать к профилю**, выберите базу данных, в которой хранится его профиль и установите **режим поиска**:

- **По email адресу из профиля** — адрес из полей профиля клиента в базе данных.
- **По email адресу из профиля или подписок** — адрес из полей профиля или из контактных данных подписок клиента.
- **По ID AppMetreica** — по значению `appmetrica_device_id`. Должен быть заранее передан профилям. При импорте через события AppMetrica устанавливается автоматически.
- **По кастомному полю** — по значению дополнительного поля профиля клиента в Альткрафт. Выберите название поля в базе данных и поле события для поиска значения.

Импорт профиля

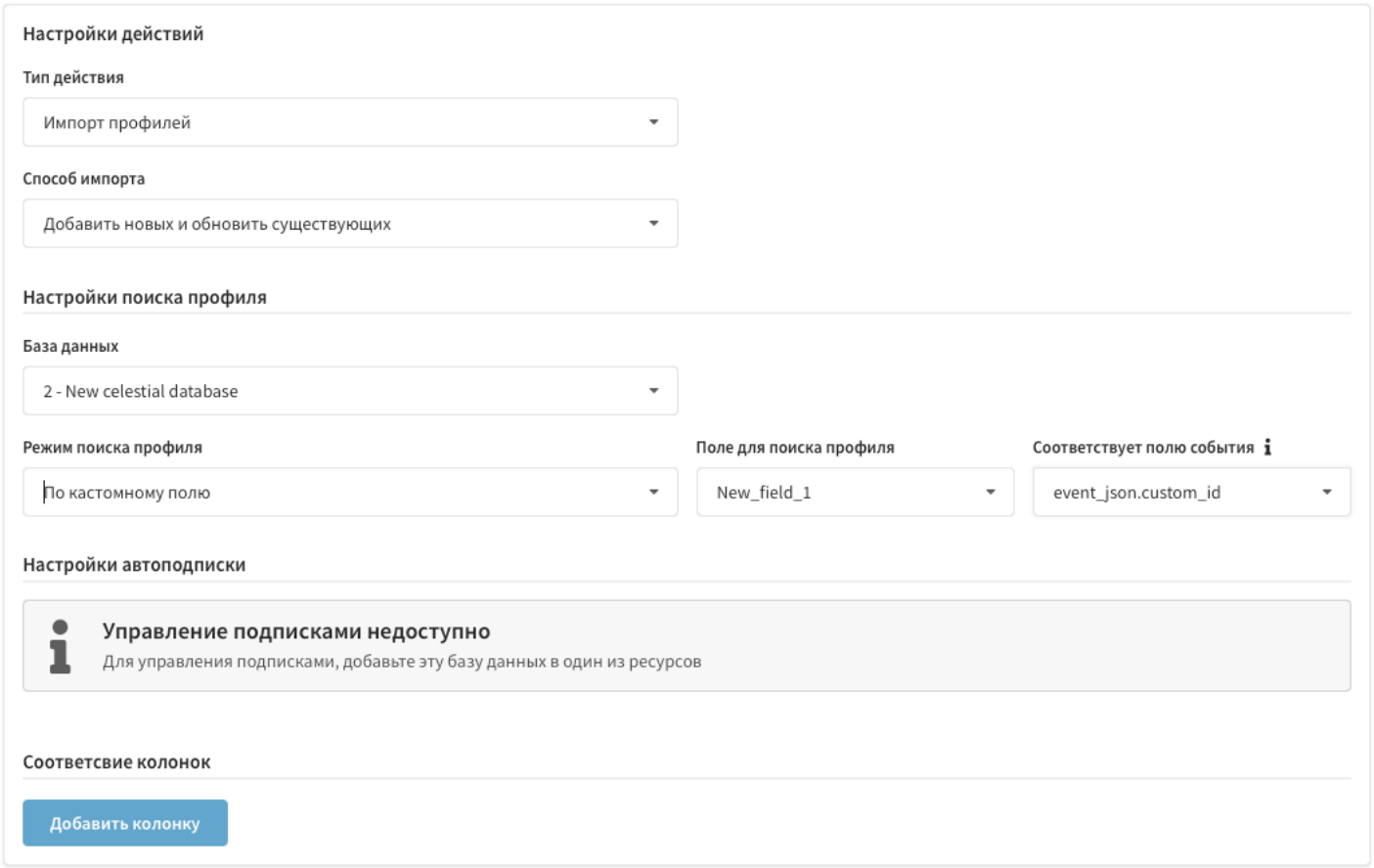
Второе применение событий из AppMetrica™ — импорт и обновление данных профилей пользователей в базах данных Altcraft. В захвате действий для этого нужно выбрать **Тип действия: Импорт профилей**. Доступно три режима импорта:

- Добавить новых и обновить существующих
- Только добавить новых, не обновлять существующих
- Только обновлять существующих, не добавлять новых

Для поиска профиля укажите базу данных Альткрафт и установите режим поиска аналогично поиску профиля при регистрации цели (выше).

CAUTION

В рамках одной базы данных для импорта нежелательно менять режим поиска с email или кастом на `appmetrica_device_id`. Используйте этот режим сразу во избежание потери данных.



Автоподписка

При импорте или обновлении профилей можно подписывать пользователей на Email, SMS или Push. Для этого необходимо:

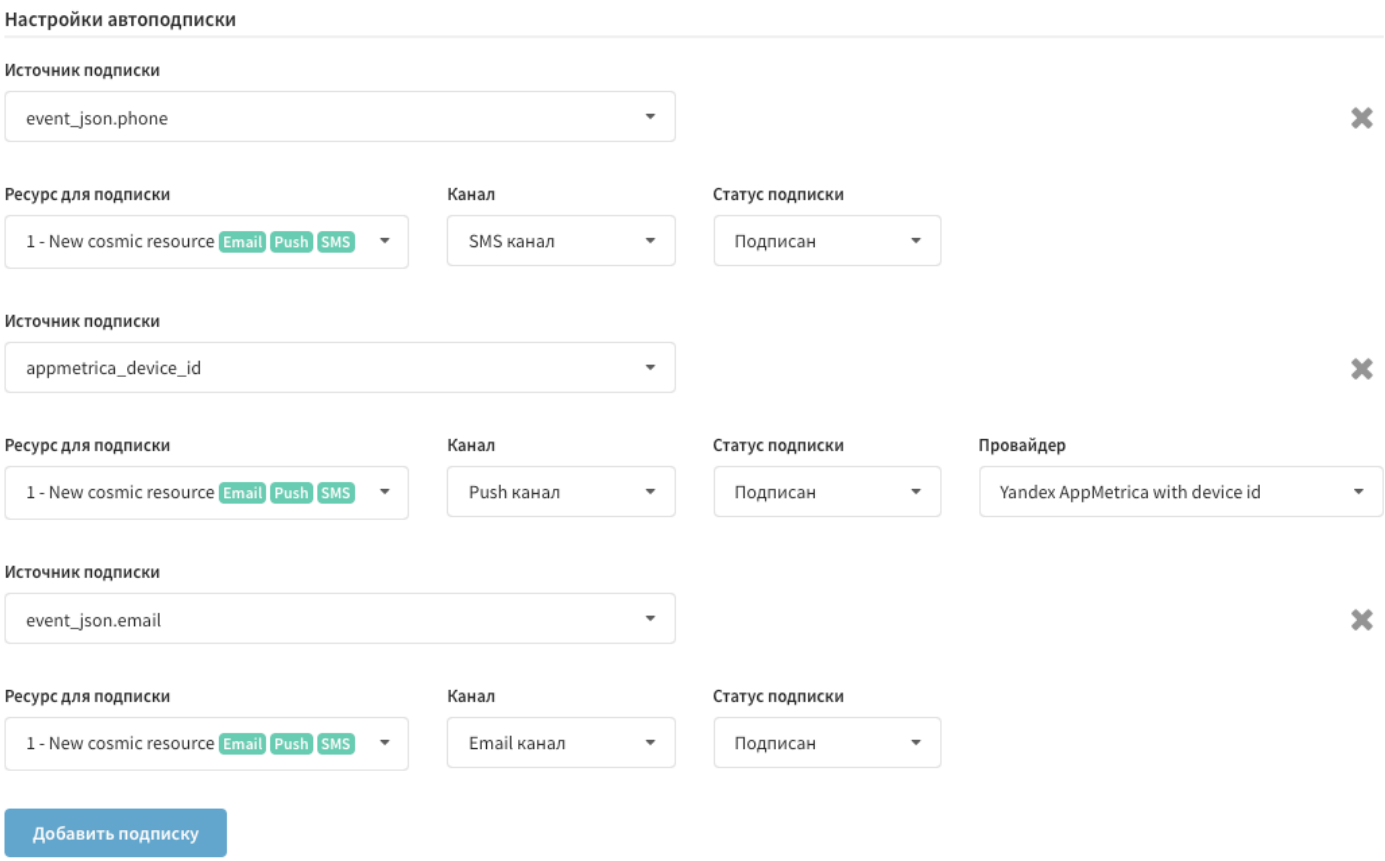
1. Создать ресурс и настроить в нём нужные каналы или использовать существующие ресурсы.
2. Добавить в настроенный ресурс Базу данных, в которую будут импортироваться данные пользователей. В рамках одного импорта можно передавать подписки на разные ресурсы. Базу данных при этом нужно добавить во все используемые ресурсы.

После этого в захвате событий станет доступна кнопка **Добавить подписку:**

Для каждой записи выберите Ресурс, Канал и Источник подписки. Для Push подписок из AppMetrica доступны стандартные поля события:

Провайдер Push	Поле события
Yandex AppMetrica with device id	appmetrica_device_id
Yandex AppMetrica with iOS IFA	ios_ifa
Yandex AppMetrica with Google AID	google_aid

Для других провайдеров Push, а также для Email и SMS подписок, данные необходимо передавать из приложения в дополнительных параметрах `event_json`.



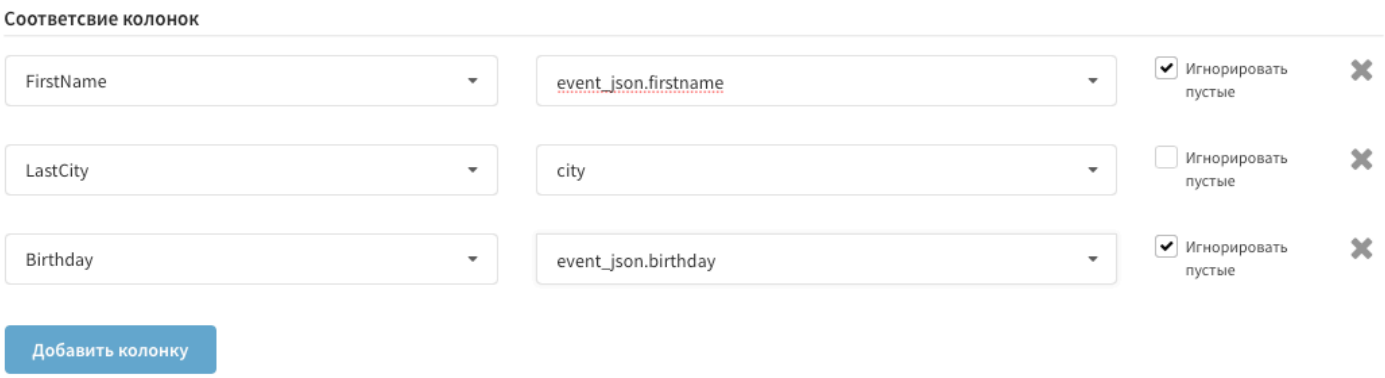
Для каждой подписки можно установить статус:

- при импорте можно передавать статус “Подписан” для Single opt-in, или “Не подтверждён” для Double opt-in логики
- при обновлении данных можно менять статус подписки в зависимости от действий пользователя, например приостановить, возобновить или отменить подписку

Соответствие данных AppMetrica и данных профиля Altcraft

Для каждого нового профиля будет создаваться уникальный идентификатор Altcraft и записываться поле, по которому происходит поиск (если по нему профиль не найден). Если режим поиска настроен на email, профилю будет также записываться системное поле `appmetrica_device_id`.

Для основных и дополнительных полей профилей Altcraft нужно установить соответствие с полями событий, получаемых от AppMetrica™:



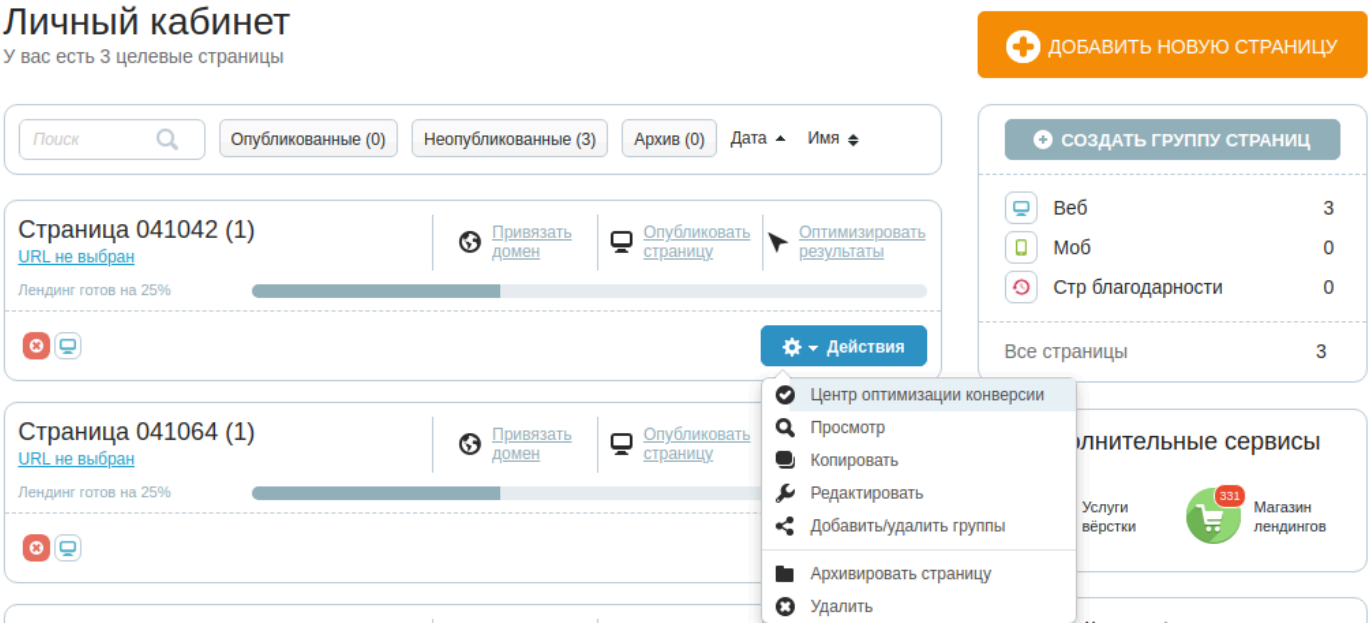
Пользовательские поля задаются в структуре Баз данных Altcraft. Пользовательские поля события передаются из приложения в объекте `event_json`.

Опция **Игнорировать пустые** запрещает запись пустого значения в поле профиля при обновлении или импорте.

Lpgenerator™

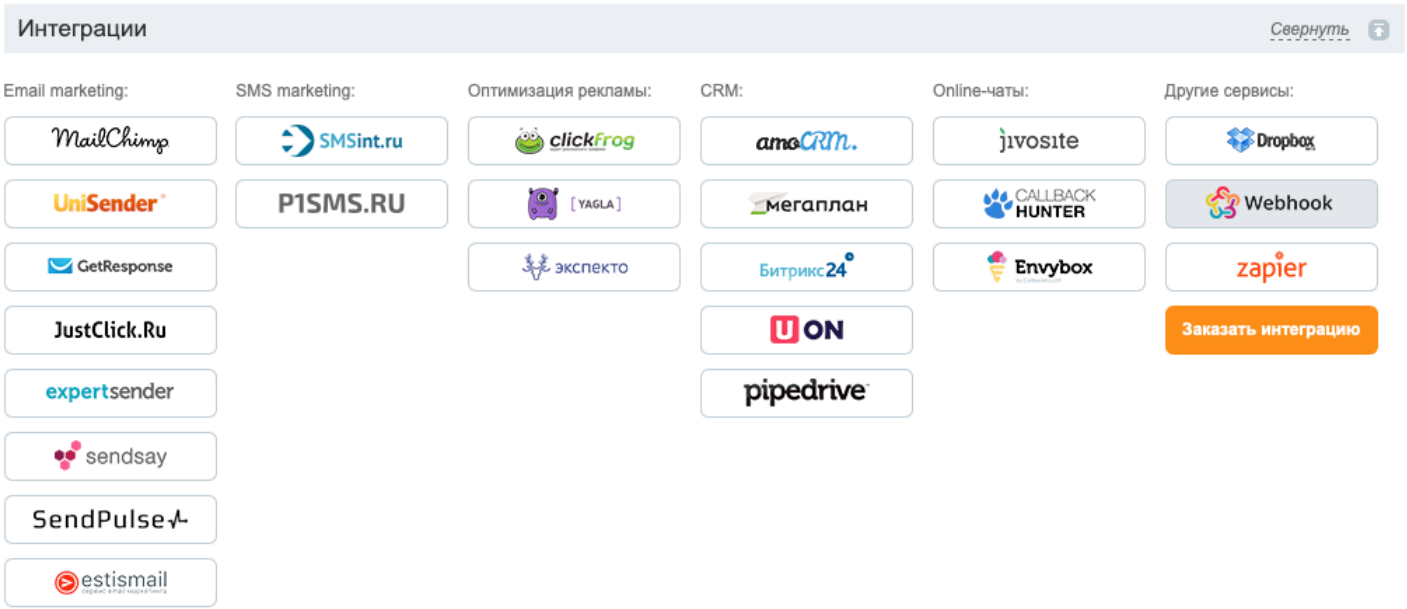
Если форма размещена на страницах сайта, созданного в сервисе LPgenerator, то данные клиента, заполнившего форму, можно передать в Altcraft Platform. Так, вы можете создать профиль для нового клиента или обновить данные уже существующего.

Чтобы подключить формы LPGenerator к аккаунту Altcraft, перейдите в личный кабинет LPGenerator. Кликните “**Действия**” для выбранной страницы, а затем выберите “**Центр оптимизации конверсии**”.



После этого в блоке “**Интеграции**” в разделе “**Другие сервисы**” выберите **Webhook**.

Описание работы Webhook на стороне сервиса: https://lpgenerator.ru/blog/2018/04/03/webhook-samostoyatelnaya-integraciya-storonnih-servisov/



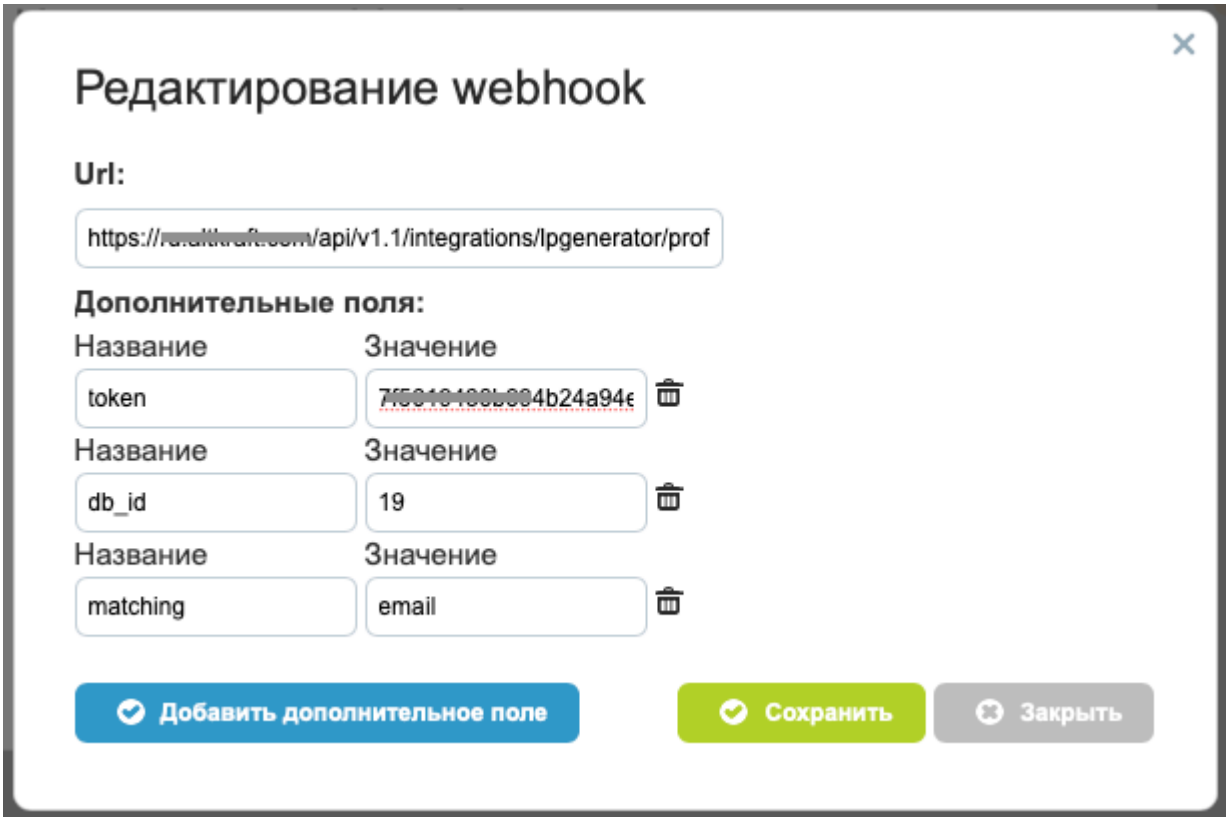
Настройка Webhook интеграции

Добавьте новый Webhook и укажите следующие данные:

URL: `https://example.com/api/v1.1/integrations/lpgenerator/profile_import`

Дополнительные поля:

Параметр	Тип	Пример	Обязательный	Описание
token	string	abcdefghijklmnqrstuvwxyz	Да	API-токен для доступа к Altcraft
db_id	string	1	Да	Идентификатор базы данных для импорта
matching	string	email phone custom	Нет	Тип поиска профиля в Altcraft для обновления: по email или телефону из полей данных или подписок профиля, - или по значению дополнительного поля профиля Altcraft.
field_name	string	custom_ID	Нет	Имя кастомного поля БД Altcraft при **matching: custom** . В форме сбора данных для поиска нужно поле с таким же именем. Например, `custom_ID`
resource_id	string	3	Нет	При передаче контактных данных (email или phone) профилям будет добавляться подписка на указанный ресурс.



Поля LPGenerator по умолчанию

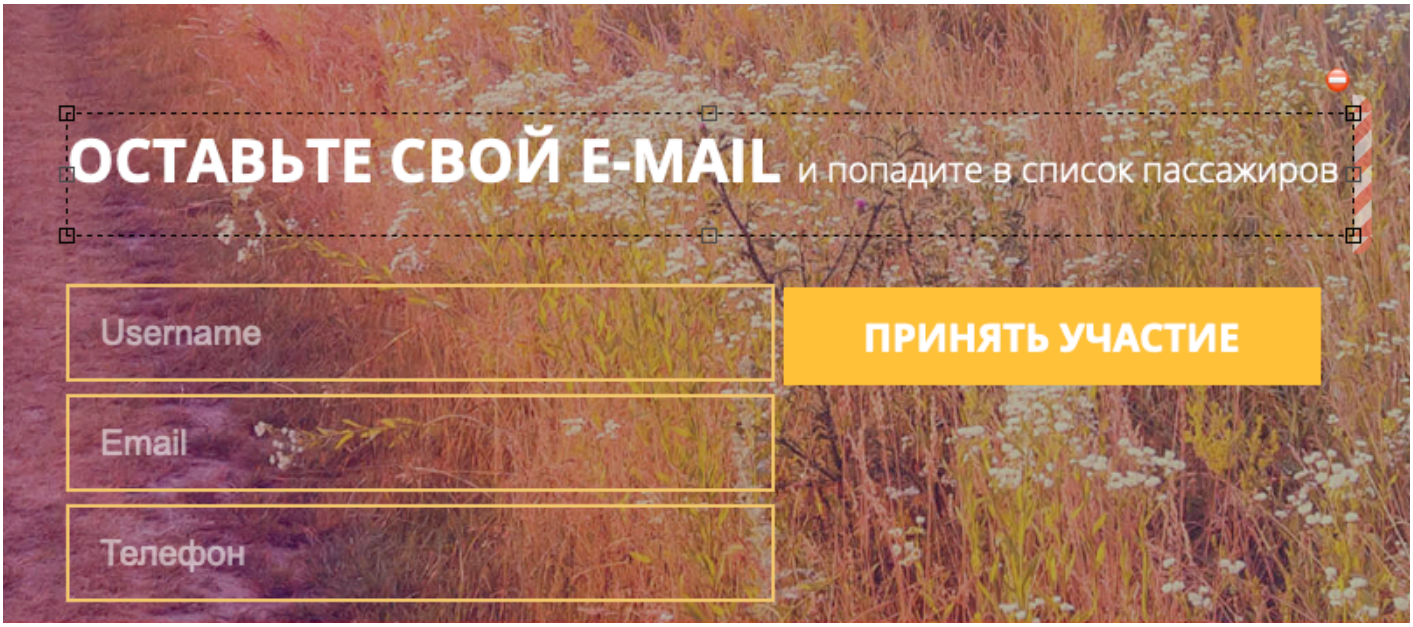
Ниже приведены неизменяемые системные поля запроса, передаваемого в Altcraft.

Параметр	Тип	Пример	Описание
id	string	“777777”	ID формы сбора данных

Параметр	Тип	Пример	Описание
meta_data	JSON object	“meta_data” : {…}	Метаданные. Используются для автоопределения IP, геолокации подписчиков, браузера и операционной системы.
utm_data	JSON object	“utm_data” : {…}	Предопределенные данные utm (часто пусто).
lead_data	JSON object	“lead_data” : {…}	Данные подписчика из формы сбора.

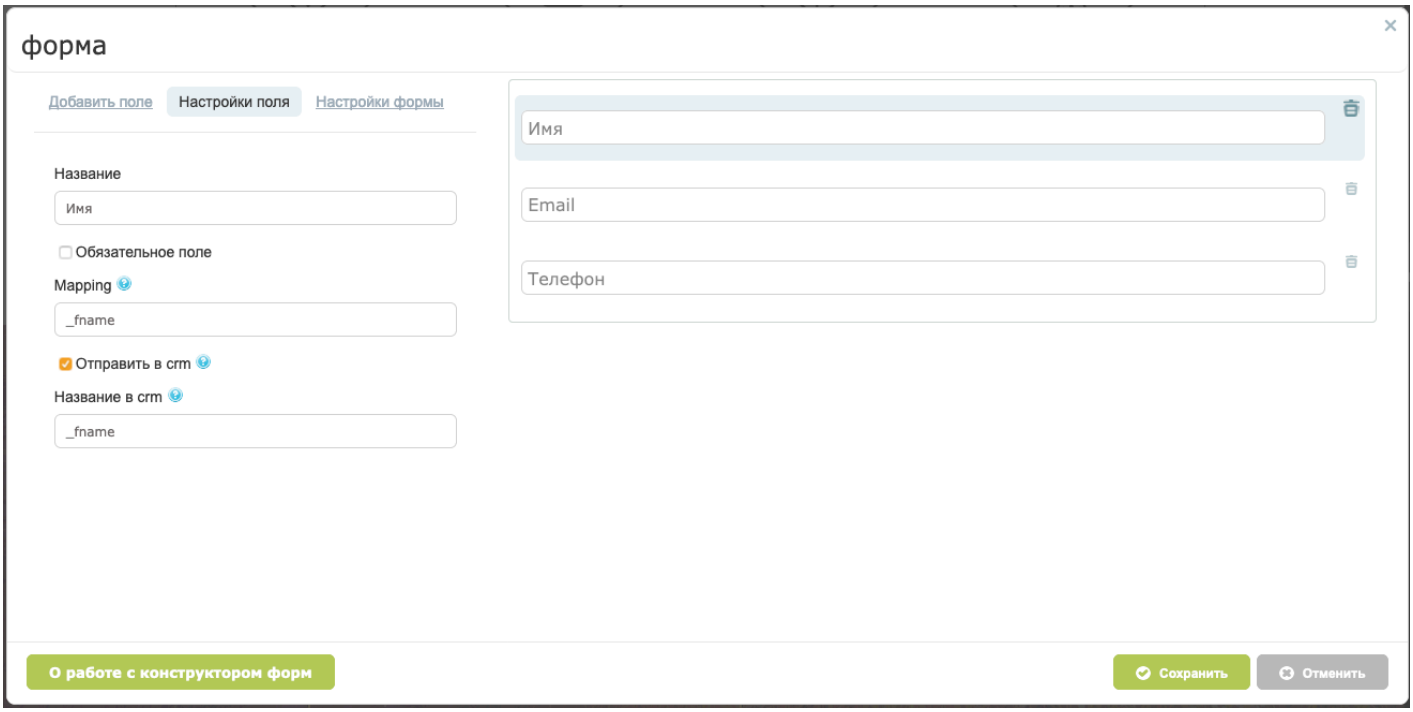
Добавление формы сбора данных

В конструкторе страниц добавьте форму:



По двойному клику на область формы откроется редактор полей формы. Добавьте необходимые поля и укажите их название для отображения на сайте.

Отметьте опцию “**Отправить в crm**”, а в поле "**Название в crm**"укажите название соответствующего поля в базе данных Altcraft. Для кастомных полей укажите те же названия полей, которые использовали в платформе, учитывая регистр. Список системных полей профиля Altcraft, таких как Имя или Email, доступен по этой ссылке.



Сохраните настройки и опубликуйте форму, чтобы начать передавать данные.

Примеры структуры запросов

Импорт подписчика по email-адресу:

```
{
        "id": 77777777,
        "lead_data": {
                "email": "qwerty@dron.ff",
        },
        "meta_data": {
                "HTTP_USER_AGENT": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
                "LANGUAGE": "en-US",
                "form_name": "форма 1",
                "from": "Страница 107570 (Первый вариант)",
                "page_id": 1642169,
                "user_ip": "96.123.123.123",
                "variant_id": 1935651
        },
        "utm_data": {}
    "db_id": "2",
    "resource_id": "2",
    "matching": "email",
    "token": "abcdefghijklmnqrstuvwxyz"
}
```

Обновление полей подписчика с подпиской на ресурс (с поиском подписчика по кастомному полю `custom_ID`):

```
{
        "id": 27773744,
        "lead_data": {
                "email": "ytrewq@dron.ff",
        "custom_ID": "Bagdasaryan"
        },
        "meta_data": {
                "HTTP_USER_AGENT": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36",
                "LANGUAGE": "en-US",
                "form_name": "форма 1",
                "from": "Страница 207571",
                "page_id": 1642169,
                "user_ip": "96.123.123.123",
                "variant_id": 1935651
        },
        "utm_data": {}
    "db_id": "2",
    "resource_id": "2",
    "matching": "custom",
    "field_name": "custom_ID",
    "token": "abcdefghijklmnqrstuvwxyz"
}
```


Импорт подписчика с передачей статуса подписки:

```
{
  "db_id": "75",
  "id": 64368040,
  "workflow_id": 4,
  "trigger_id": 1,
  "field_for_APIContent": "any data",
  "lead_data": {
    "_status": 0,
    "_fname": "By phone (explicitly) After",
    "phone": "+234257929700231073",
    "New_field_5": 6.2,
    "subscriptions": [
      {
        "resource_id": 5,
        "channel": 2,
        "channel_sid": "sms",
        "phone_data": {
          "phone": "+234257929700231073",
          "md5": "33890b49c501f101965c6ac4fb01e409"
        },
        "priority": 0,
        "status": 2,
        "hash_id": "ceffaf99",
        "is_delete": false
      }
    ]
  },
  "matching": "phone",
  "meta_data": {
    "HTTP_USER_AGENT": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/67.0.3396.99 Safari/537.36",
    "LANGUAGE": "en-US",
    "form_name": "\\u0444\\u043e\\u0440\\u043c\\u0430 1",
    "from": "\\u0421\\u0442\\u0440\\u0430\\u043d\\u0438\\u0446\\u0430 518658
(\\u041f\\u0435\\u0440\\u0432\\u044b\\u0439
\\u0432\\u0430\\u0440\\u0438\\u0430\\u043d\\u0442)",
    "page_id": 1623331,
    "user_ip": "94.231.119.122",
    "variant_id": 1937033
  },
  "resource_id": "5",
  "token": "a3c31d8a15714531afebfd1541eeba41",
  "utm_data": {}
}
```


Поле	Значение	Обазательное	Описание

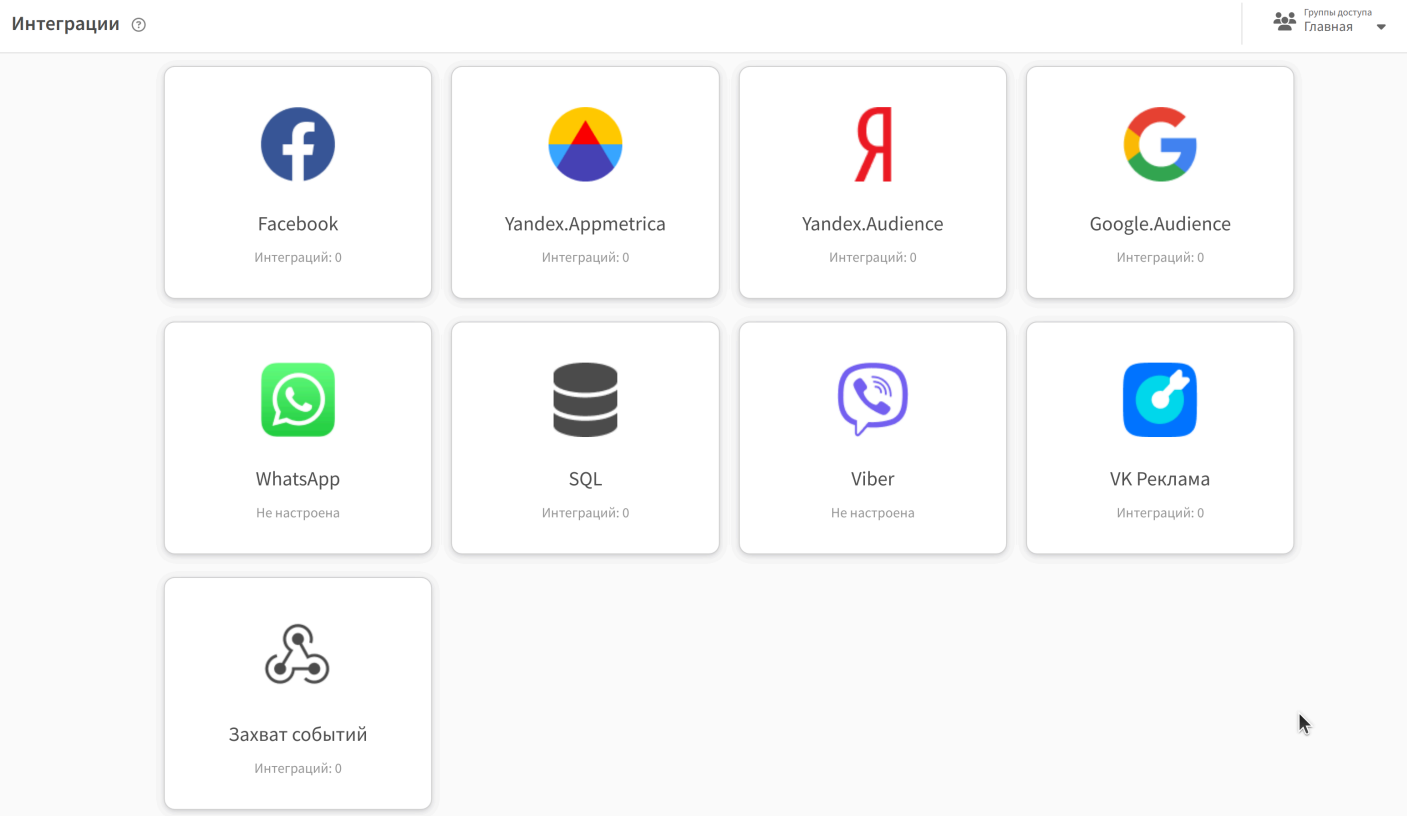
workflow_id	идентификатор сценария	Нет	Добавить профиль в сценарий сразу после импорта профиля
trigger_id	идентификатор триггерной рассылки	Нет	Запустить триггерную рассылку сразу после импорта профиля
_status	0 — Подписан 1 — Отписан 2 — Жалобщик 3 — Hardbounced 4 — Не подтвержден 5 — Приостановлен 6 — Не валиден	Нет	Глобальная подписка профиля
status		Нет	Статус подписки на определенный ресурс

VK Реклама™

Интеграция с VK Рекламой™ позволяет синхронизировать данные статических сегментов Altcraft Platform с аудиториями в рекламном кабинете Вконтакте. Другими словами, вы переносите сегменты из платформы в **VK Рекламу™**, чтобы затем этот сервис, используя полученные данные о профилях, нашел нужных людей и запустил для них таргетированную рекламу.

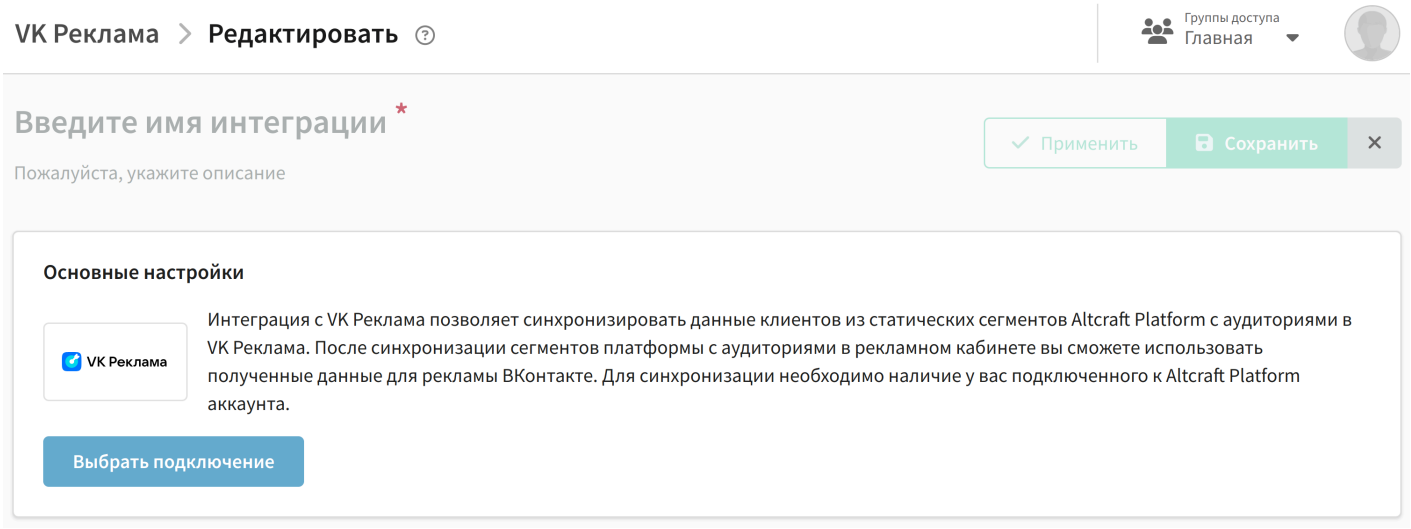
> **TIP**
>
> Для интеграции с VK Рекламой™ необходимо настроить **подключение** в платформе Altcraft. Подробную инструкции вы найдете здесь.

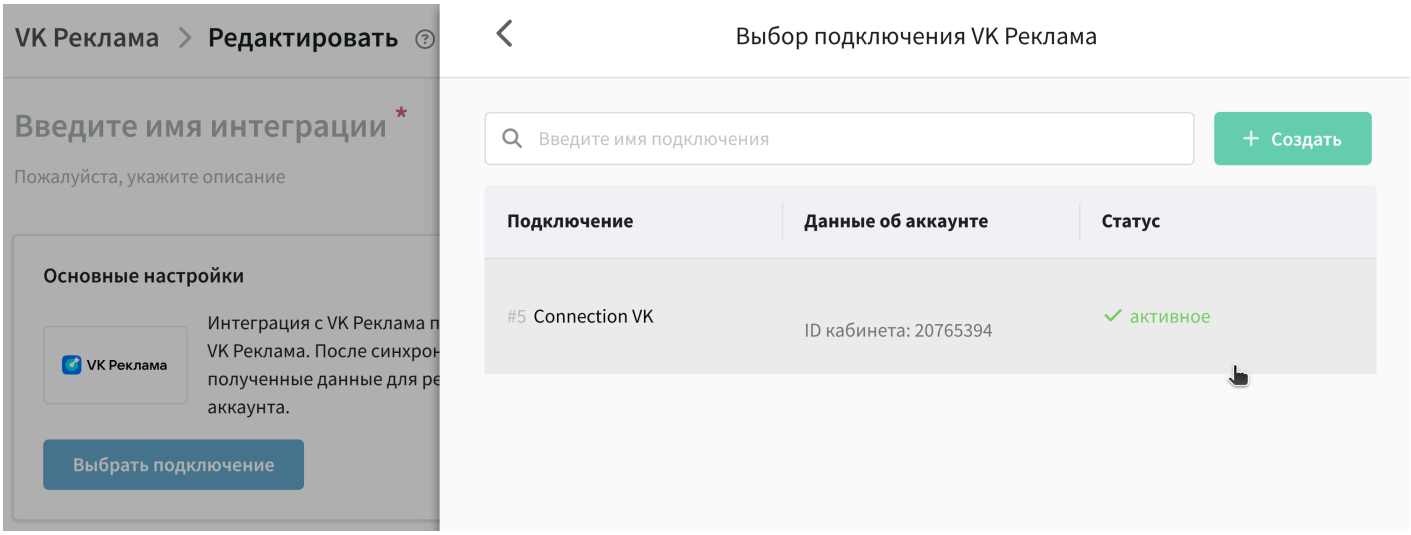
Чтобы добавить синхронизацию, в главном меню кликните "**Интеграции**". Выберите тип интеграции — **VK Реклама**, а затем в верхней части страницы нажмите кнопку **“Создать”**.



Настройка

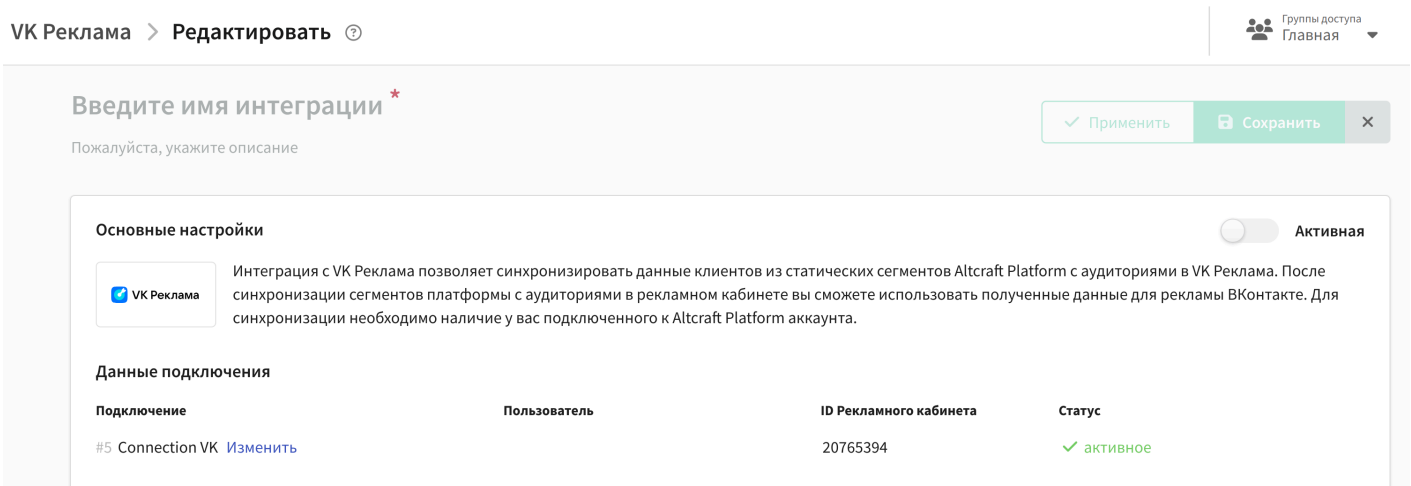
Перед вами откроется окно создания интеграции. Нажмите на кнопку **“Выбрать подключение”** для просмотра существующих подключений:





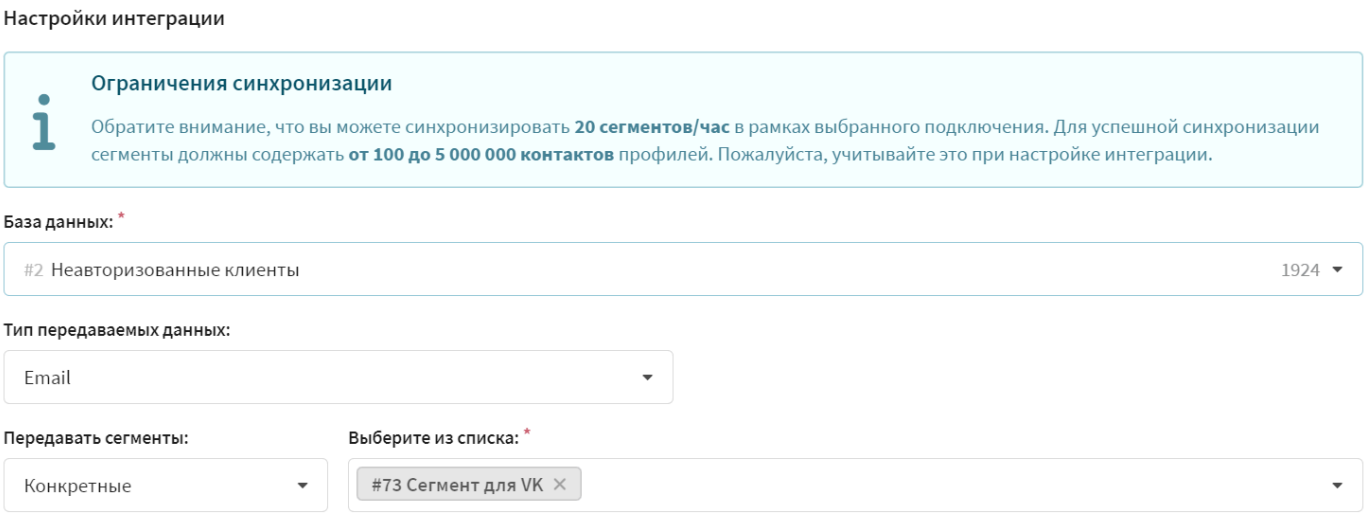
Из этого меню вы можете создать новое подключение. С подробной инструкцией по созданию подключения вы можете ознакомиться в этой статье.

Основные настройки



На вкладке “Основные” введите **Имя** и **Описание**. Здесь же находится переключатель, с помощью которого вы можете активировать или остановить интеграцию.

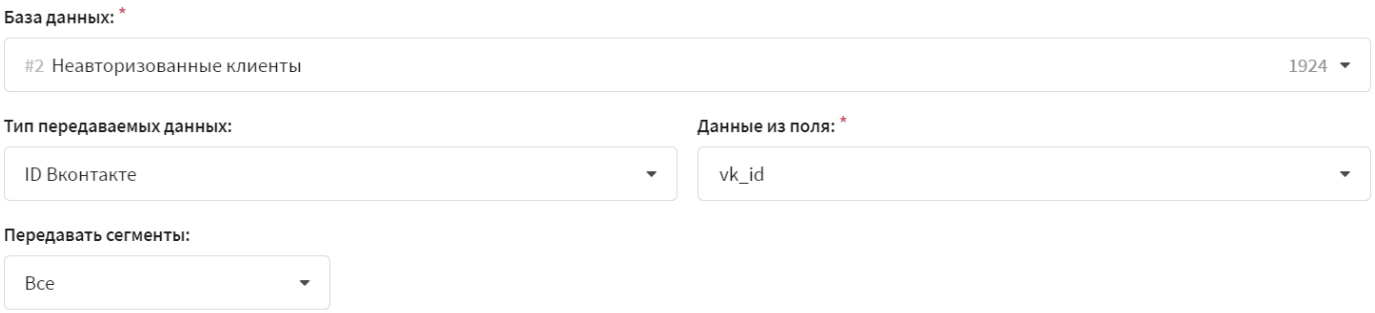
Настройки интеграции



В данном разделе вы можете выбрать **одну** базу и до 20 сегментов выгрузки, выбрать **тип передаваемых данных**. Обратите внимание, что нельзя перенести сегмент, который будет содержать **менее 100 профилей**. VK Реклама™ поддерживает интеграцию следующих типов данных:

- ID Вконтакте
- Email
- ID Одноклассники
- Телефонные номера
- Google Advertising ID
- IDFA устройства

Для каждого типа, кроме email и номера телефона, вам понадобится дополнительное поле в базе данных. После выбора типа передаваемых данных появится окно с выбором поля, из которого будет выгружаться информация с платформы.

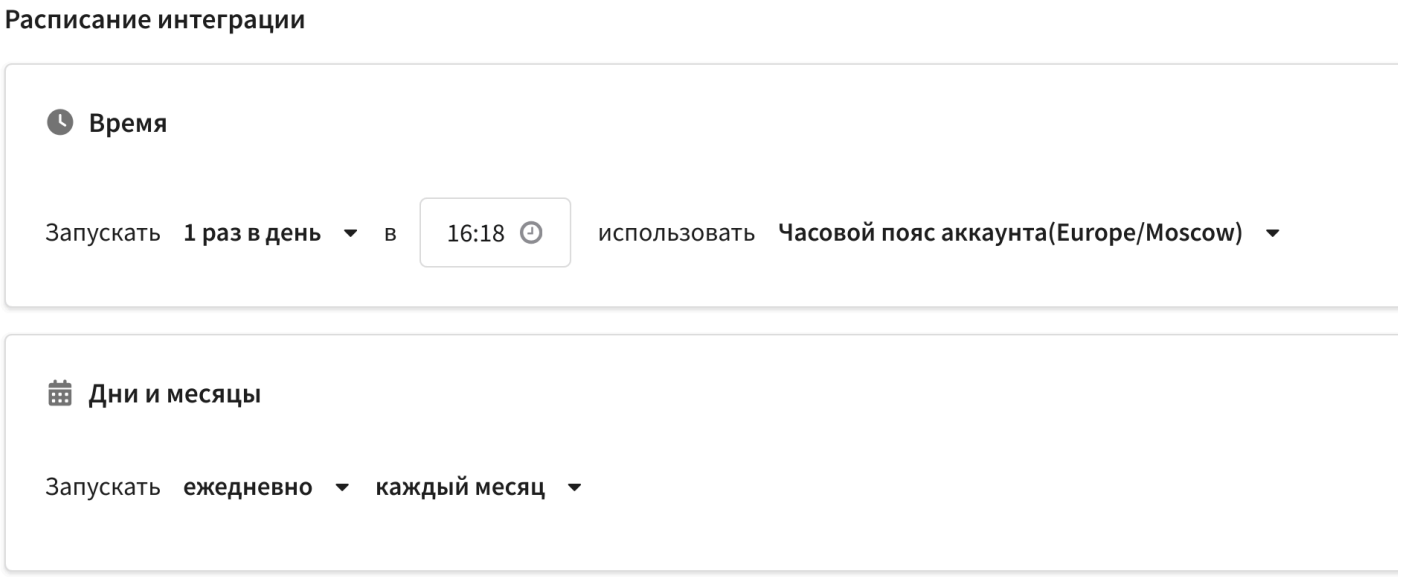


CAUTION

Интеграция будет выгружать данные **только из выбранной базы данных** . Если ваш сегмент затрагивает несколько баз данных, то импортированы будут только те профили, которые входят в изначально указанную вами базу.

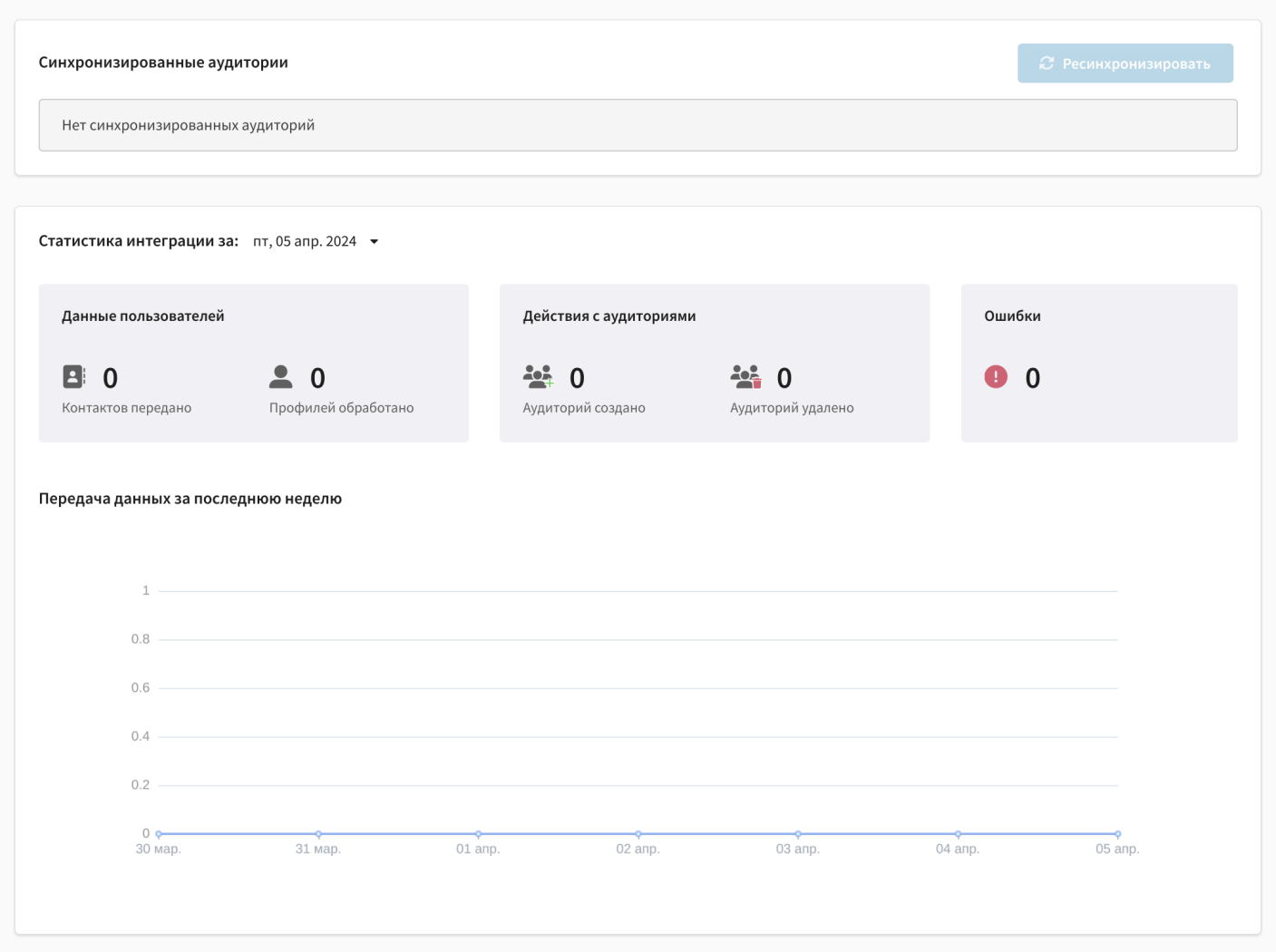
Расписание интеграции

Здесь вы можете настроить то, с какой переодичностью будет происходить интеграция. Подробнее о том, как настроить расписание в платформе, читайте в этом разделе.



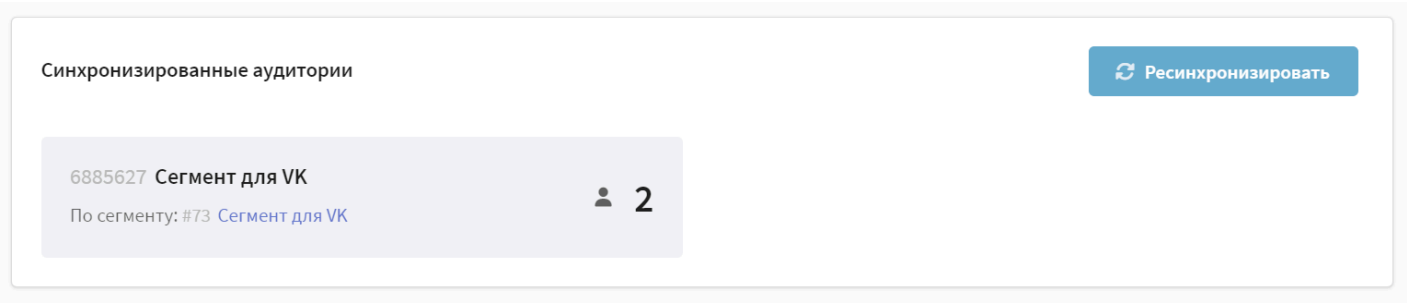
Синхронизация и статистика

После того, как вы сохраните интеграцию, на странице настроек появятся разделы **Синхронизированные аудитории** и **Статистика интеграции**:



В разделе **Синхронизированные аудитории** появится список перенесённых в VK Рекламу™ аудиторий. Также вы сможете синхронизировать аудитории нажатием кнопки **“Ресинхронизировать”**. После нажатия

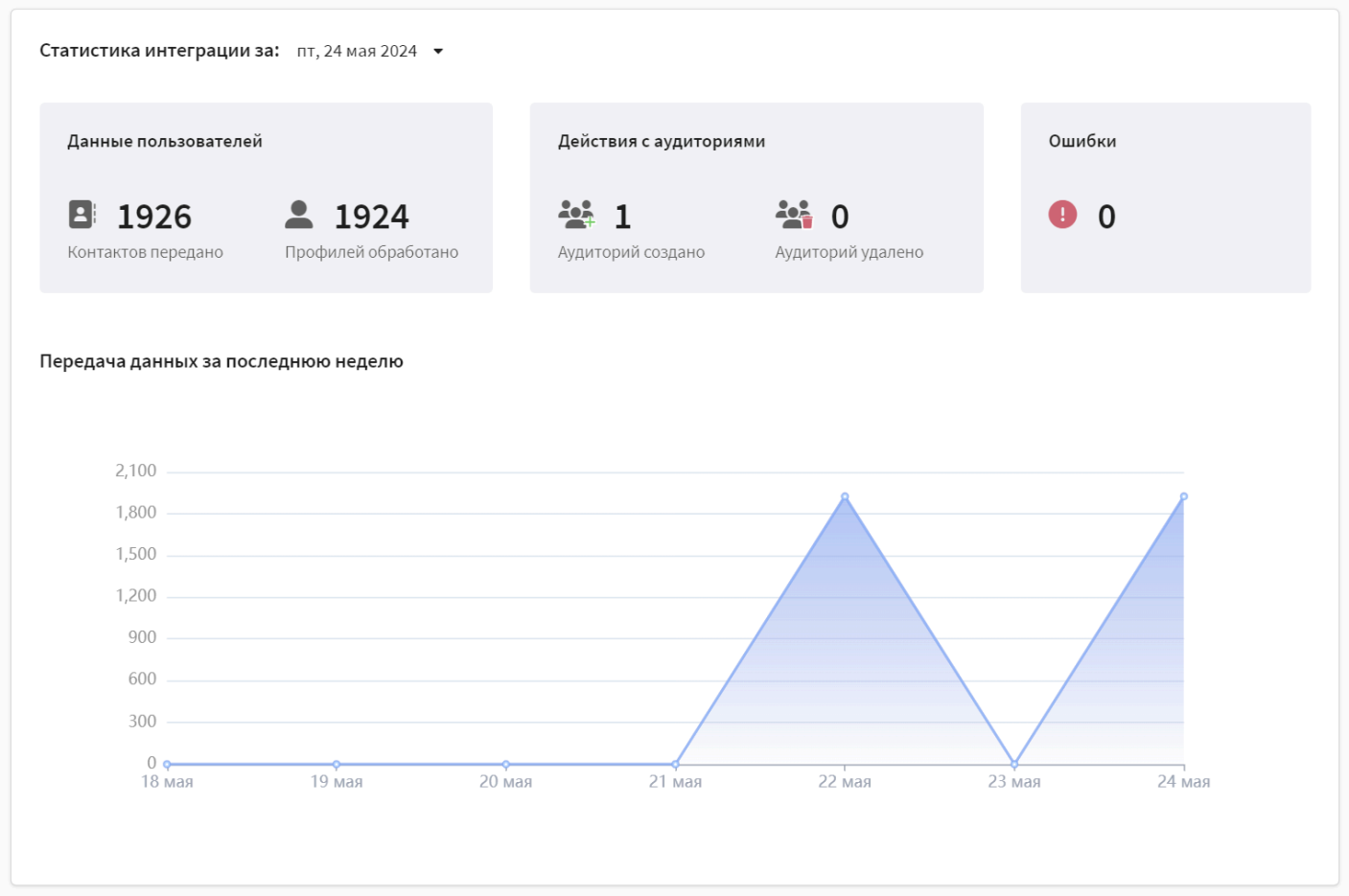
будут удалены все когда-либо синхронизированные аудитории и заново загружены все сегменты, которые попали в область видимости интеграции.



TIP

Интеграция переносит только те профили, которые появились с момента её запуска. Для наиболее полного переноса аудиторий, рекомендуем запустить **Ресинхронизацию** перед запуском интеграции

В разделе **Статистика интеграции** вы сможете увидеть подробную информацию о синхронизациях за последнюю неделю, какие действия совершались с ними и ошибки, возникшие во время синхронизации.



TIP

Пункт “Контактов передано” засчитывает каждый переданный **канал связи** от профиля. Например, если у профиля указаны телефон, email и ID Вконтакте, то в статистику попадёт 3 контакта, а не 1.

Передаваемые при синхронизации данные

Для синхронизации сегментов на основе аудиторий из внешних источников (Google Audience, Яндекс Аудитории™, VK Реклама™) в соответствующие системы передаются данные профилей в захешированном виде.

В статье приведён перечень данных, передаваемых при синхронизации аудиторий. Эта информация может потребоваться пользователям платформы при запросе со стороны службы информационной безопасности.

Яндекс.Аудитории™

Авторизация	
Идентификатор приложения	Client ID приложения из Яндекс.OAuth
Секрет приложения	Client Secret приложения из Яндекс.OAuth
Передаваемые данные	
email	Адрес электронной почты. Передается в хешированном виде **(MD5)**
phone	Телефон. Передается в хешированном виде **(MD5)**
externalIDValue	Идентификатор сегмента в Яндекс.Аудиториях™. Передается в хешированном виде **(MD5)**
PushData.SubscriptionID	ID push-подписки. Передается в хешированном виде **(MD5)**

TIP: Хеширование данных

Хеширование данных можно отключить, для этого измените значение параметра конфигурации `YAAUDIENCES_SYNC_DO_HASHING` на `false`.

Google.Audience

Авторизация	
Идентификатор приложения	Client ID
Секретный ключ	Client secret
Токен	Developer token
Передаваемые данные	
email	Email. Передается в хешированном виде **(SHA256)**
phone	Phone. Передается в хешированном виде **(SHA256)**
Рекламные идентификаторы	IDFA/Google Advertising ID и другие рекламные ID
Данные о сегменте	ID и название сегмента

VK Реклама™

Авторизация	
Идентификатор клиента	client_id
Секретный ключ	client_secret
Передаваемые данные	
vk	Идентификатор пользователя VK
email	Адрес электронной почты. Передается в хешированном виде **(MD5)**
phone	Номер телефона. Передается в хешированном виде **(MD5)**
ok	Идентификатор пользователя Одноклассники
advertising_id	Идентификатор Google Advertising ID
idfa	Идентификатор мобильного устройства Apple

Facebook Ads Manager*™

Авторизация	
Идентификатор приложения	App ID
Секретный ключ приложения	App Secret
Передаваемые данные	
em	Email. Передается в хешированном виде **(SHA256)** после нормализации
ph	Телефон. Передается в хешированном виде **(SHA256)** после нормализации
ct	Город. Передается в хешированном виде **(SHA256)** после нормализации
country	Страна. Передается в хешированном виде **(SHA256)** (ISO 3166-1 alpha-2)
fn	Имя. Передается в хешированном виде **(SHA256)** после нормализации
ln	Фамилия. Передается в хешированном виде **(SHA256)** после нормализации
ge	Пол. Передается в хешированном виде **(SHA256)**, `m`/`f`
db	Дата рождения. Передается в хешированном виде **(SHA256)** в формате YYYYMMDD
external_id	Передается без хеширования (рекомендуется передавать в исходном формате)

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Notify

INFO

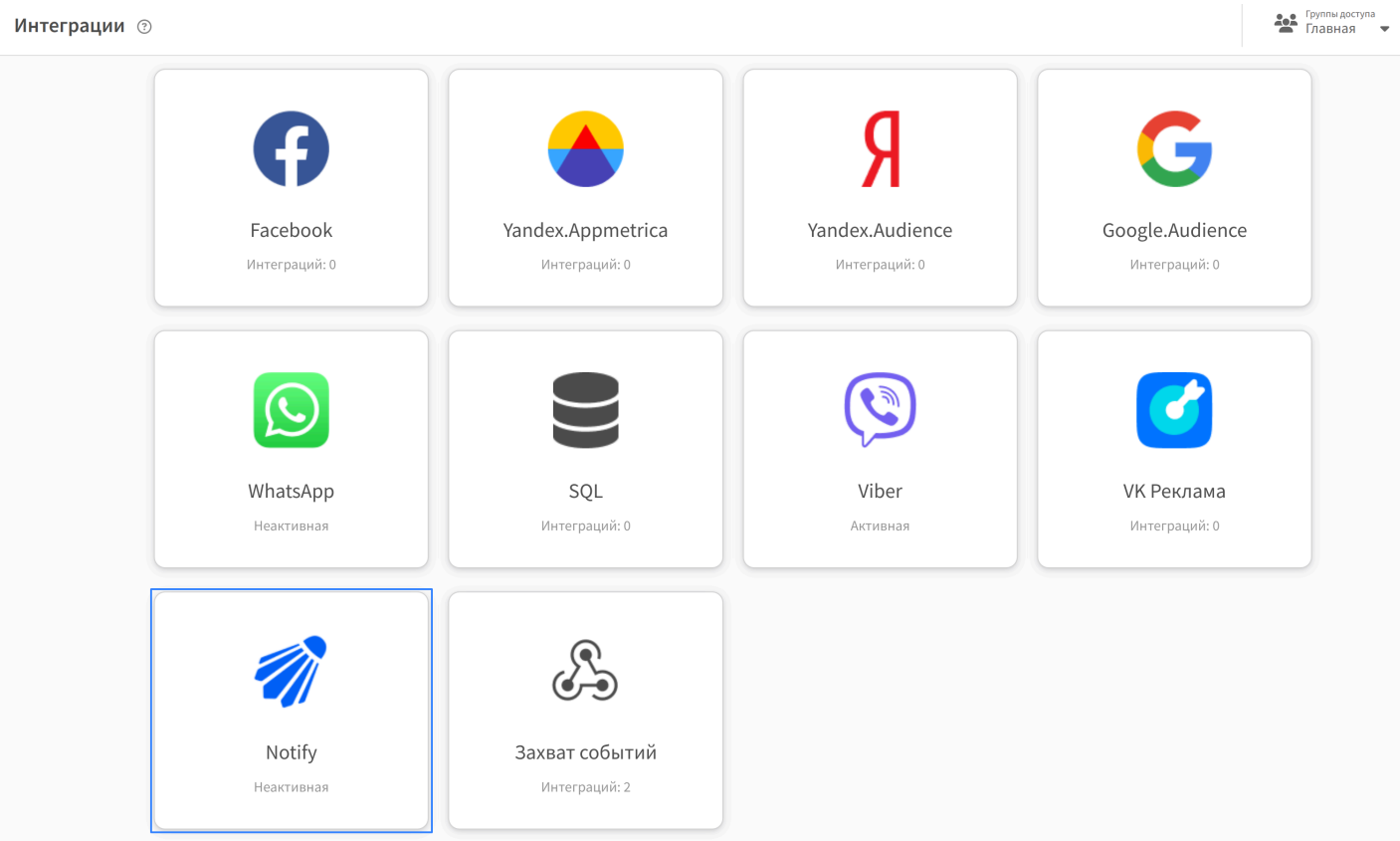
Согласно политике Notify, в данный момент новые доступы к личному кабинету сервиса **не предоставляются.**

Поддержка текущих пользователей доступна через техподдержку MAX Business.

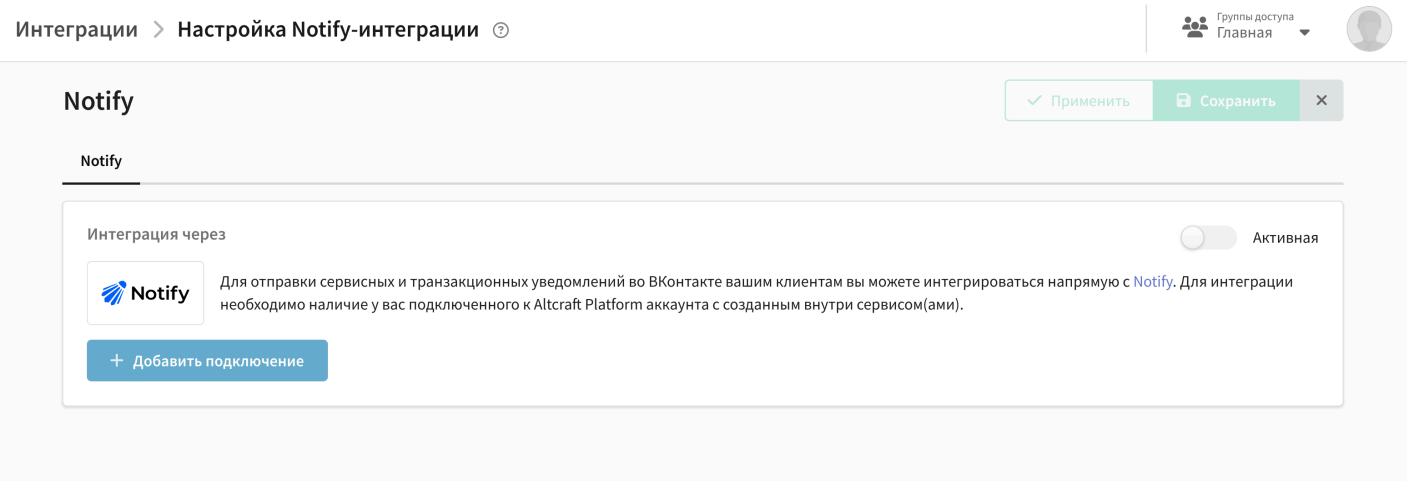
Интеграция с **Notify** позволит использовать сервис как канал коммуникации для рассылки транзакционных и сервисных уведомлений. Для создания интеграции требуется предварительно настроить подключение к сервису. Инструкция по подключению к Notify доступна на **этой странице**.

Настройка интеграции

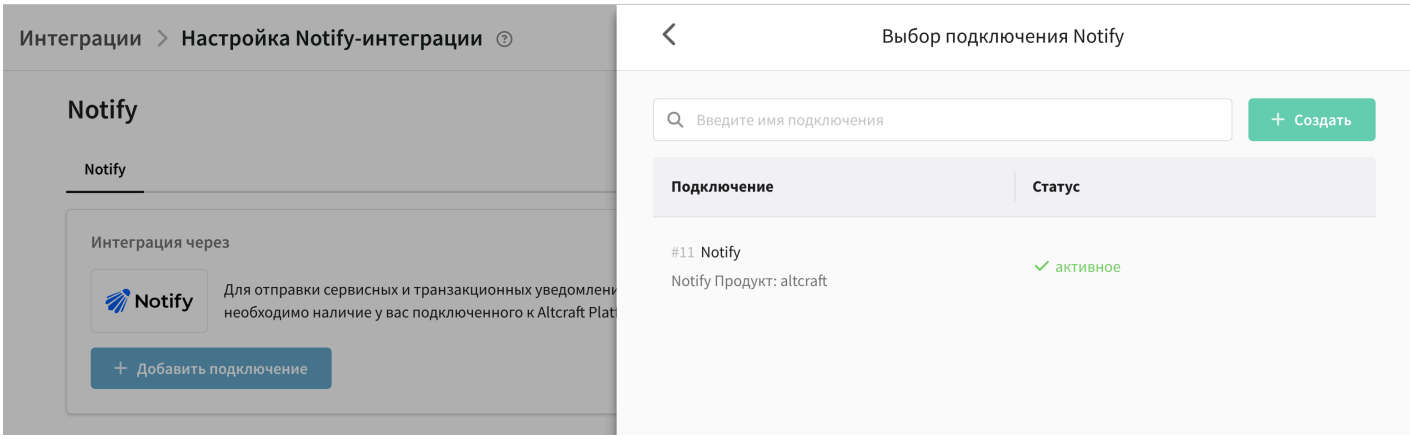
Откройте раздел **"Интеграции"** и выберите тип интеграции — **Notify**:



Нажмите **“Добавить подключение”**:



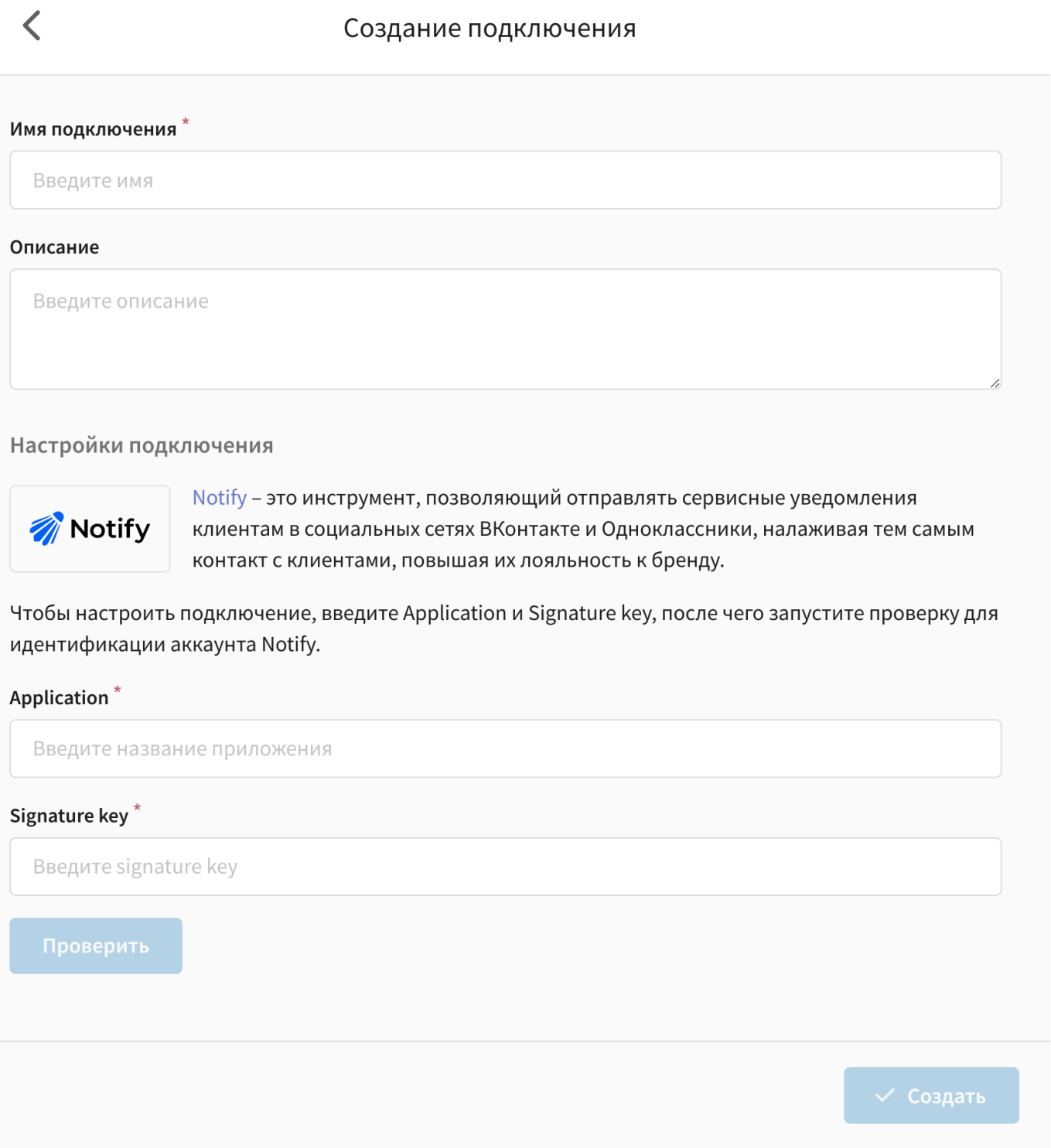
Выберите доступное **подключение** к Notify:



Нет доступных подключений

Если вы ещё не настраивали подключение к Notify, вы можете создать его непосредственно в этом окне.

Нажмите **“Создать”**. Введите имя и описание подключения. Далее укажите параметры **Application** и **Signature key** и нажмите кнопку **“Проверить”**, чтобы проверить правильность введенных данных. Если данные прошли проверку, нажмите кнопку **“Создать”** в правом нижнем углу:



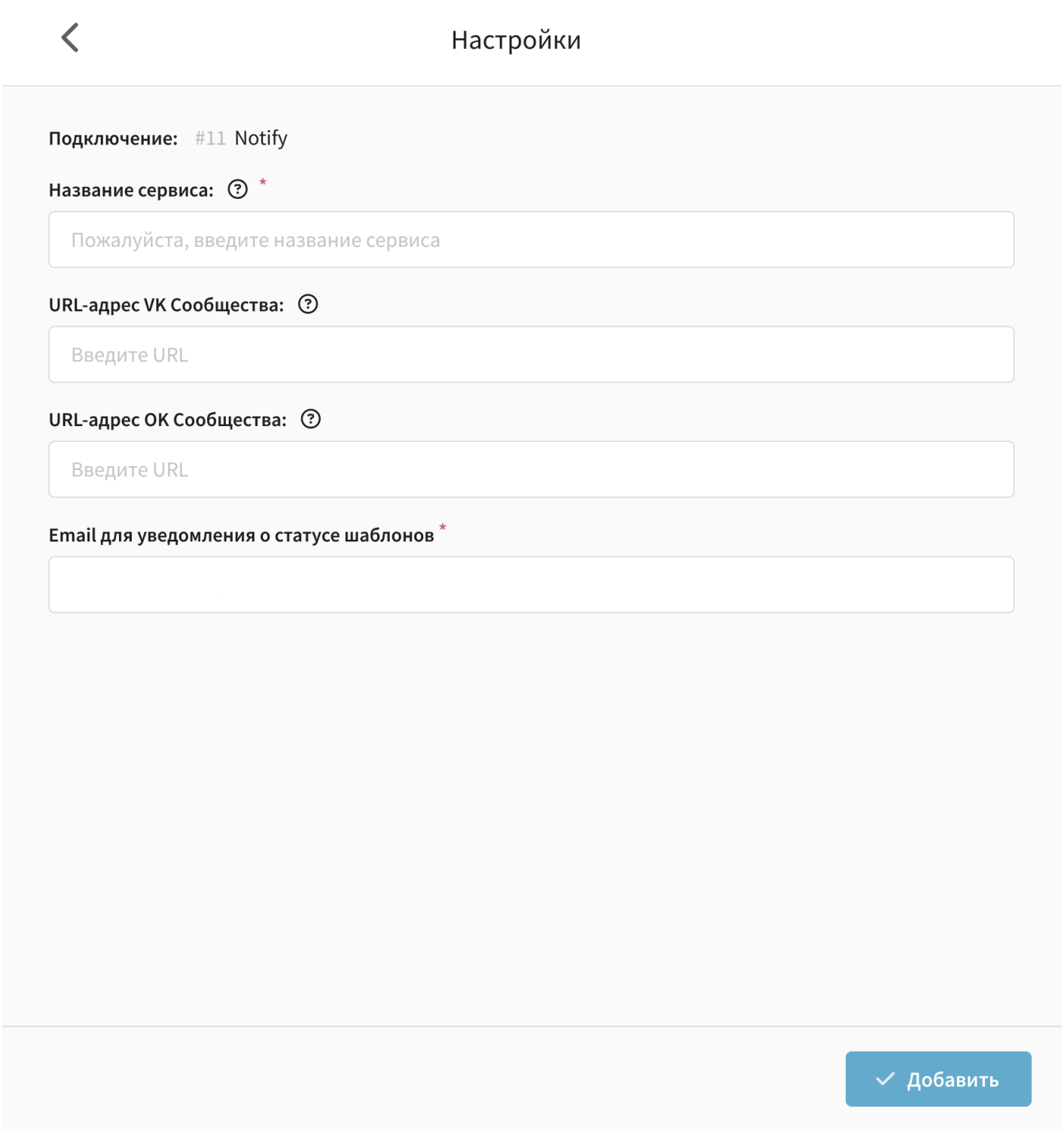
TIP

Подробнее о создании подключения можно прочитать **здесь**.

Далее укажите **параметры интеграции**, для корректной работы они должны **точно совпадать** с соответствующими параметрами из личного кабинета Notify:

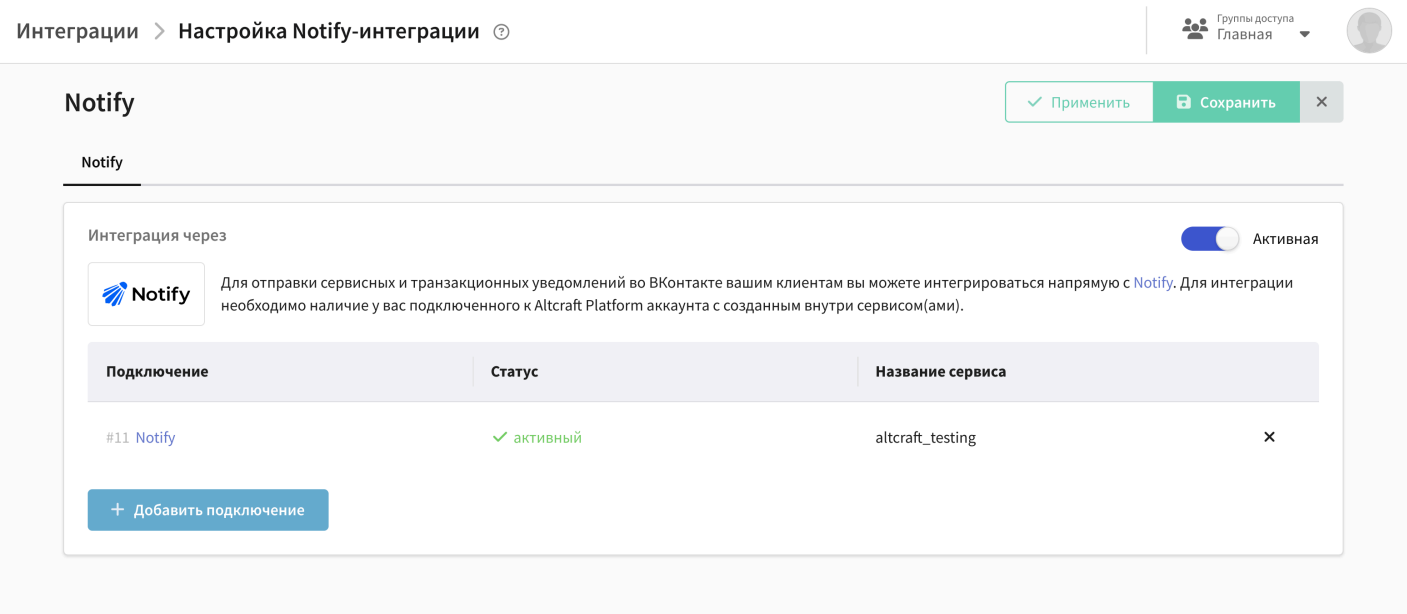
- **Название сервиса** — название сервиса для рассылки, обязательное поле.
- **URL-адрес VK Сообщества** — ссылка на сообщество Вконтакте, от имени которого будет происходить рассылка.
- **URL-адрес OK Сообщества** — ссылка на сообщество в Одноклассники, от имени которого будет происходить рассылка.

- **Email для уведомления о статусе шаблонов** — электронный почтовый адрес, на который будут приходить отчеты о статусах шаблонов. Обязательное поле.



После заполнения полей нажмите кнопку **“Добавить”**. Если введенные данные некорректны, платформа попросит проверить их правильность. Если же данные верны, то параметры сохранятся.

Активируйте интеграцию с помощью переключателя и **сохраните** её с помощью кнопок в правом верхнем углу.



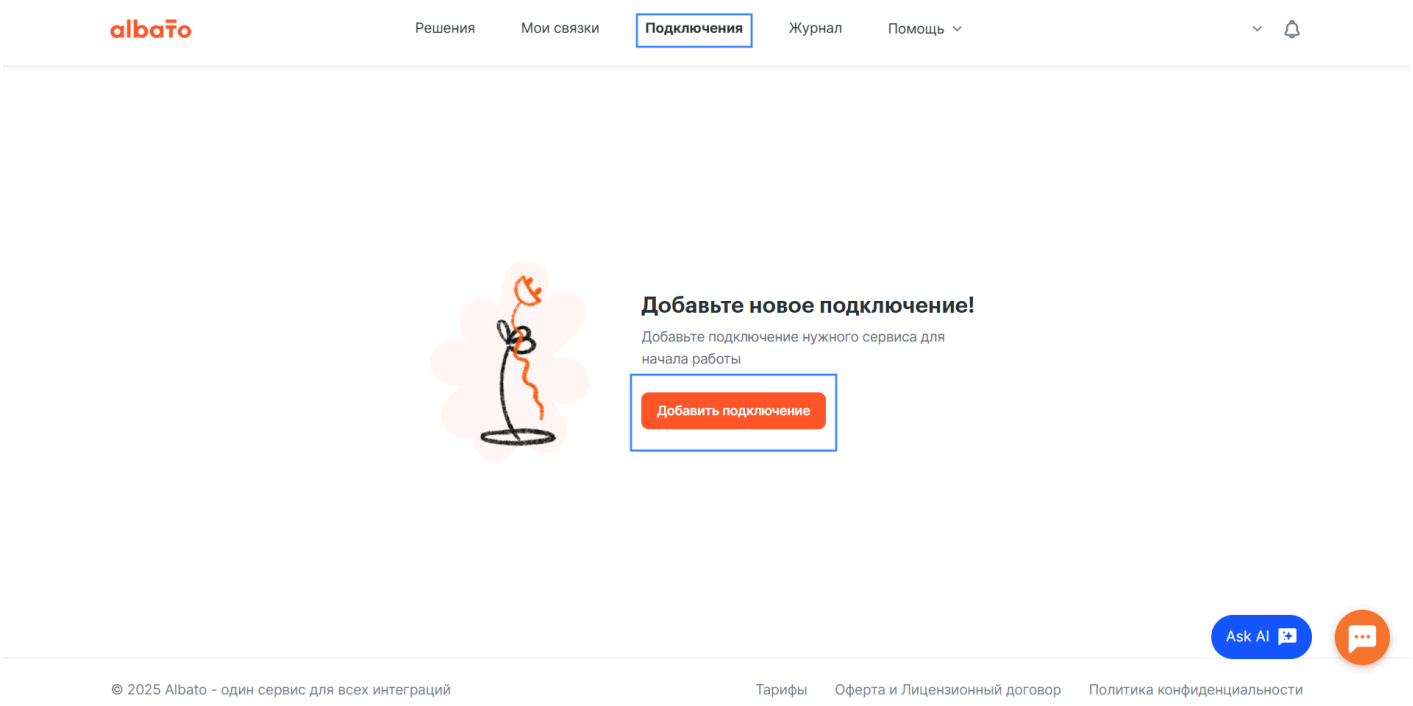
После настройки интеграции вы можете использовать её для добавления Notify-канала в **ресурс**.

Подключение Altcraft к Albato

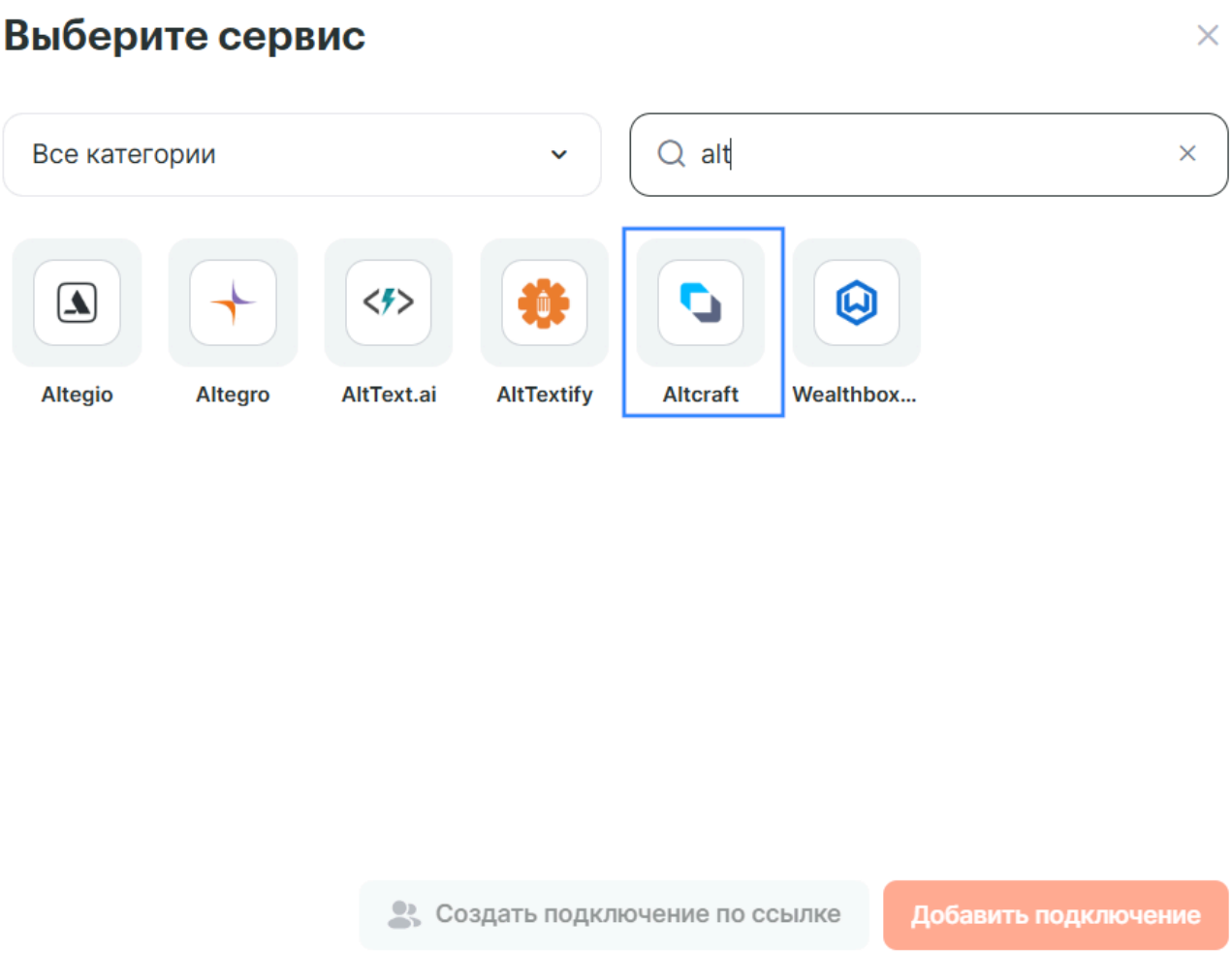
Перед тем как настроить интеграцию с нужным сервисом, необходимо подключить Altcraft к Albato. Для этого необходимо войти в аккаунт Albato.

Далее в верхнем меню выбрать раздел **“Подключения”**.

В окне “Подключения” в меню слева нажимаем на кнопку **“Добавить подключение”:**



В открывшемся окне в поле **“Выберите сервис”** введите в поисковой строке название платформы — **Altcraft.** Выберите найденный сервис из списка и нажмите кнопку “Добавить подключение” для подтверждения выбора:

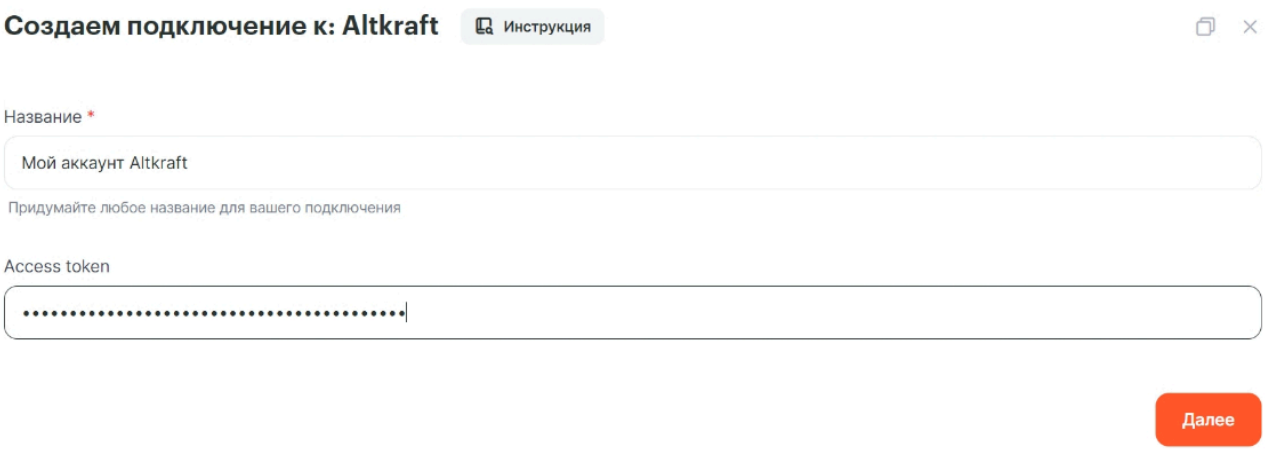


После успешного добавления сервиса откроется окно для ввода параметров подключения. В соответствующих полях необходимо указать:

- **Наименование подключения** (опционально, присваивается автоматически)
- **Ключ доступа (Access Token),** предоставляемый платформой Altcraft (является обязательным)

TIP

Получение Access Token в платформе Altcraft описано в документации по ссылке.



После ввода ключа доступа (Access Token) система отобразит детали созданного подключения.

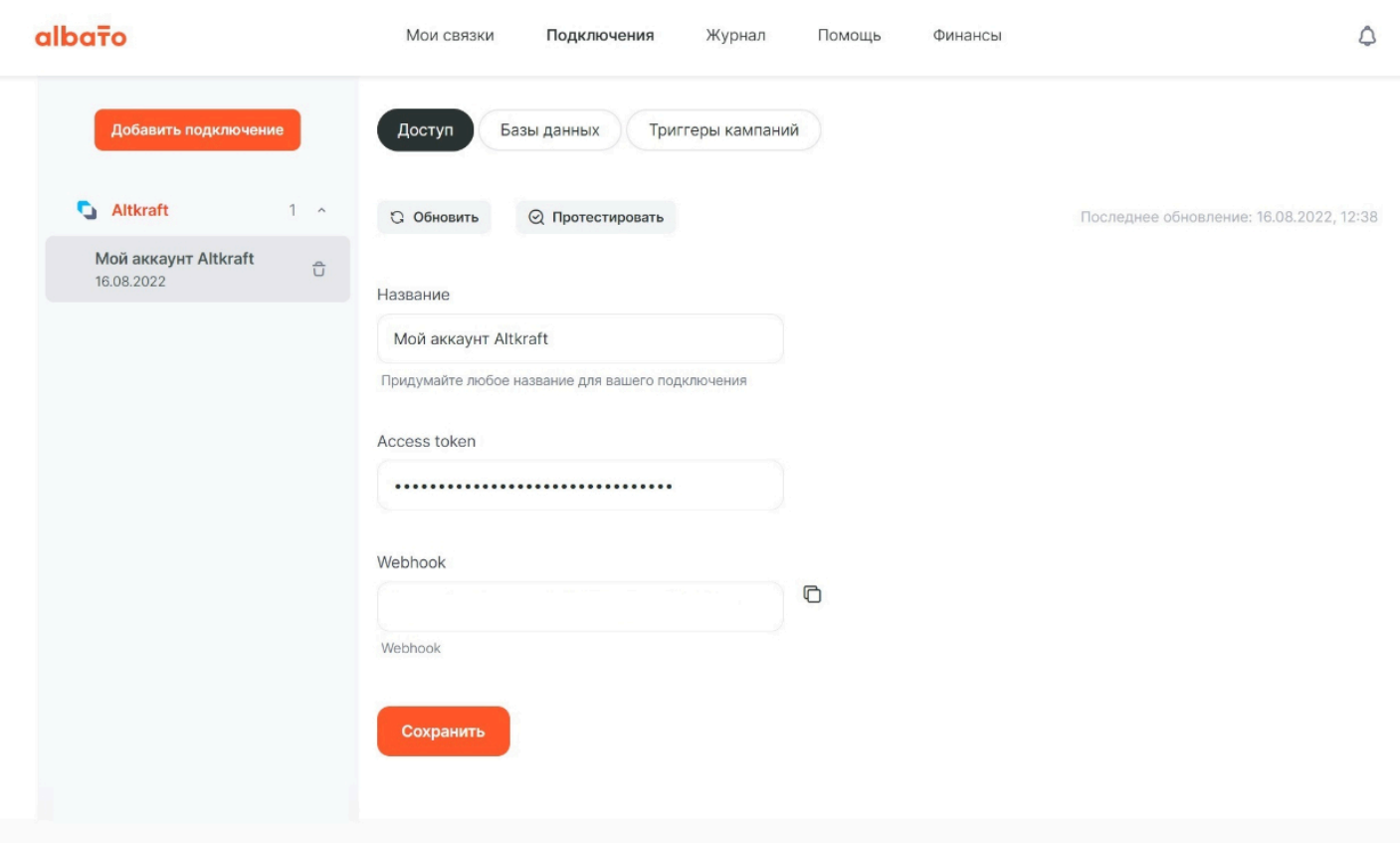
При успешном завершении процесса подключение “Altcraft” будет отображено в списке подключений в левом меню раздела “Подключения”.

CAUTION: Важно!

В Albato можно подключить несколько пользователей Altcraft с ограниченным доступом. Для этого необходимо сгенерировать новый Access Token, присвоить необходимые права и создать новое подключение по описанию выше.

Также необходимо указать наименования для каждого подключения, чтобы впоследствии было удобно просматривать и тестировать подключения.

Чтобы протестировать корректность подключения, необходимо выбрать подключение из списка (если их несколько). Далее откроется окно с информацией по подключению:



В этом окне можно посмотреть список доступных баз данных (могут быть ограничения в зависимости от указанной роли Access Token), а также список доступных триггерных рассылок и протестировать подключение.

Запуск приветственного сценария через Albato

Albato позволяет настроить передачу данных из внешнего сервиса в платформу Altcraft. С помощью специального API-запроса (подробнее о них можно прочитать в разделе для разработчика), который поступает от Albato, платформа решает, что делать с данными клиента.

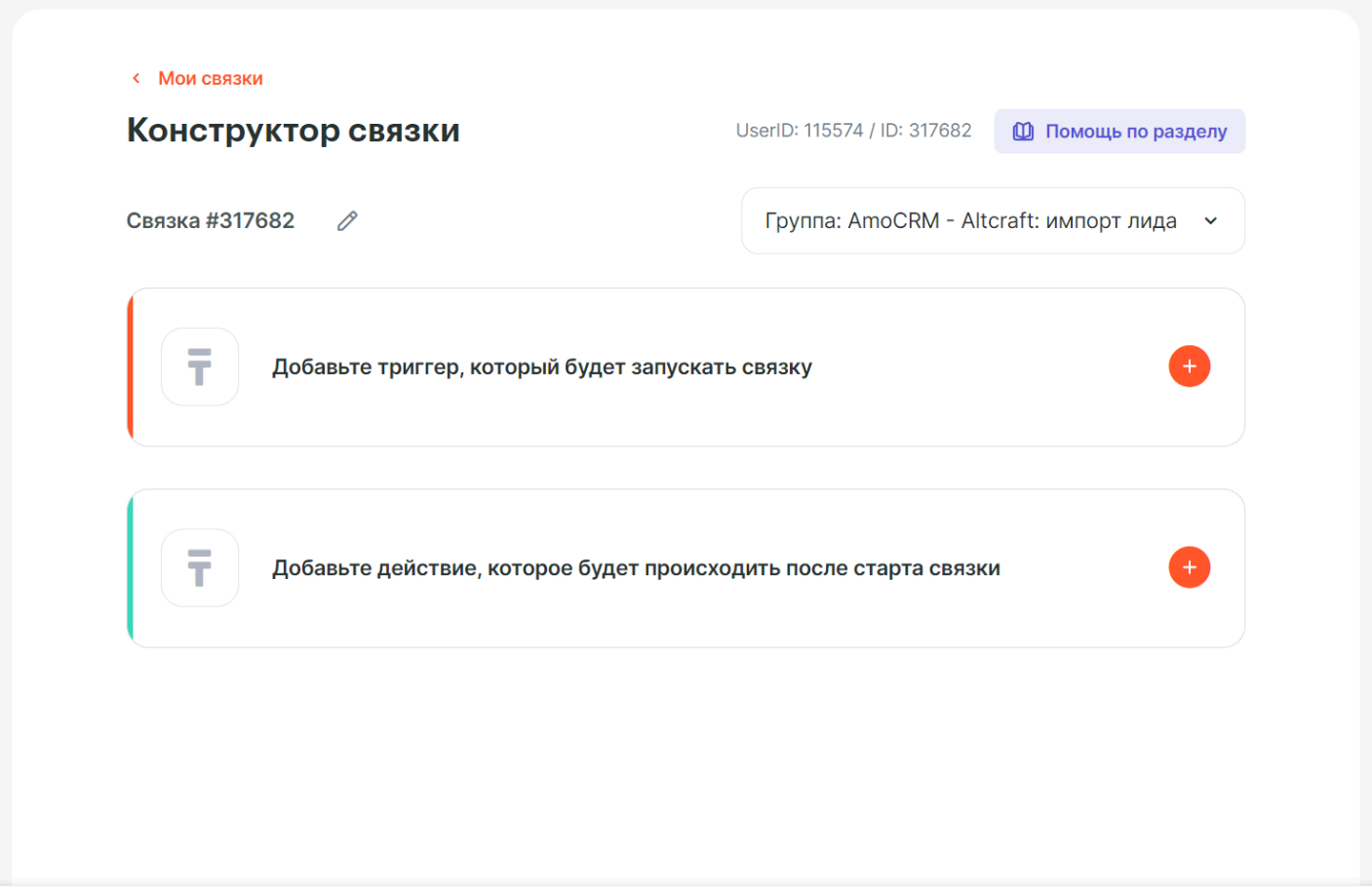
Платформа может:

- посредством импорта обновить данные в профиле клиента
- добавить профиль в сегмент
- отправить клиента в сценарий

В данной статье разберем, как с помощью Albato добавить профиль клиента в приветственный сценарий

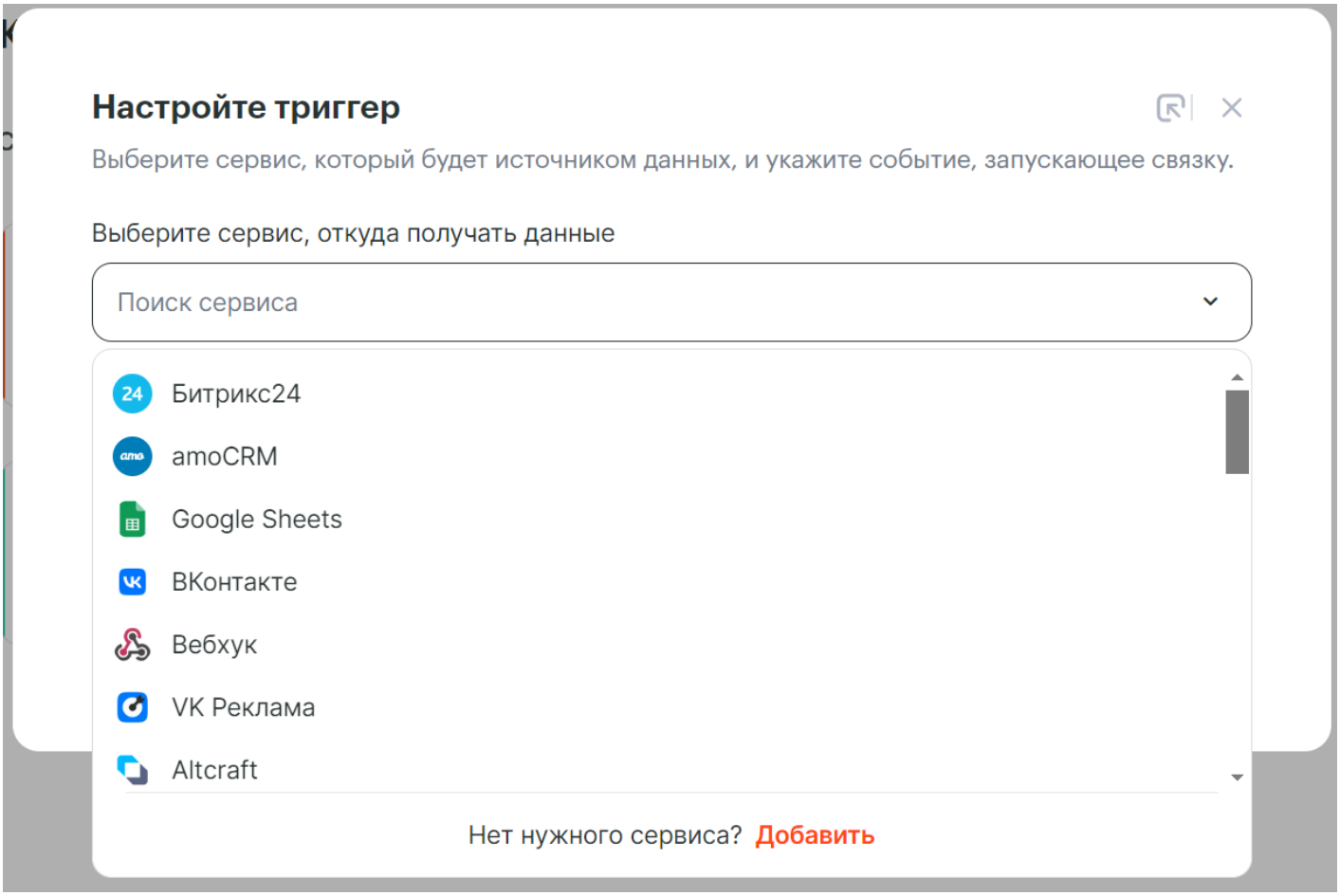
Создание новой связки в Albato

Связка в Albato состоит из двух компонентов: из триггера, который запускает передачу данных из внешнего сервиса на Albato, и действия, происходящего после запуска триггера:

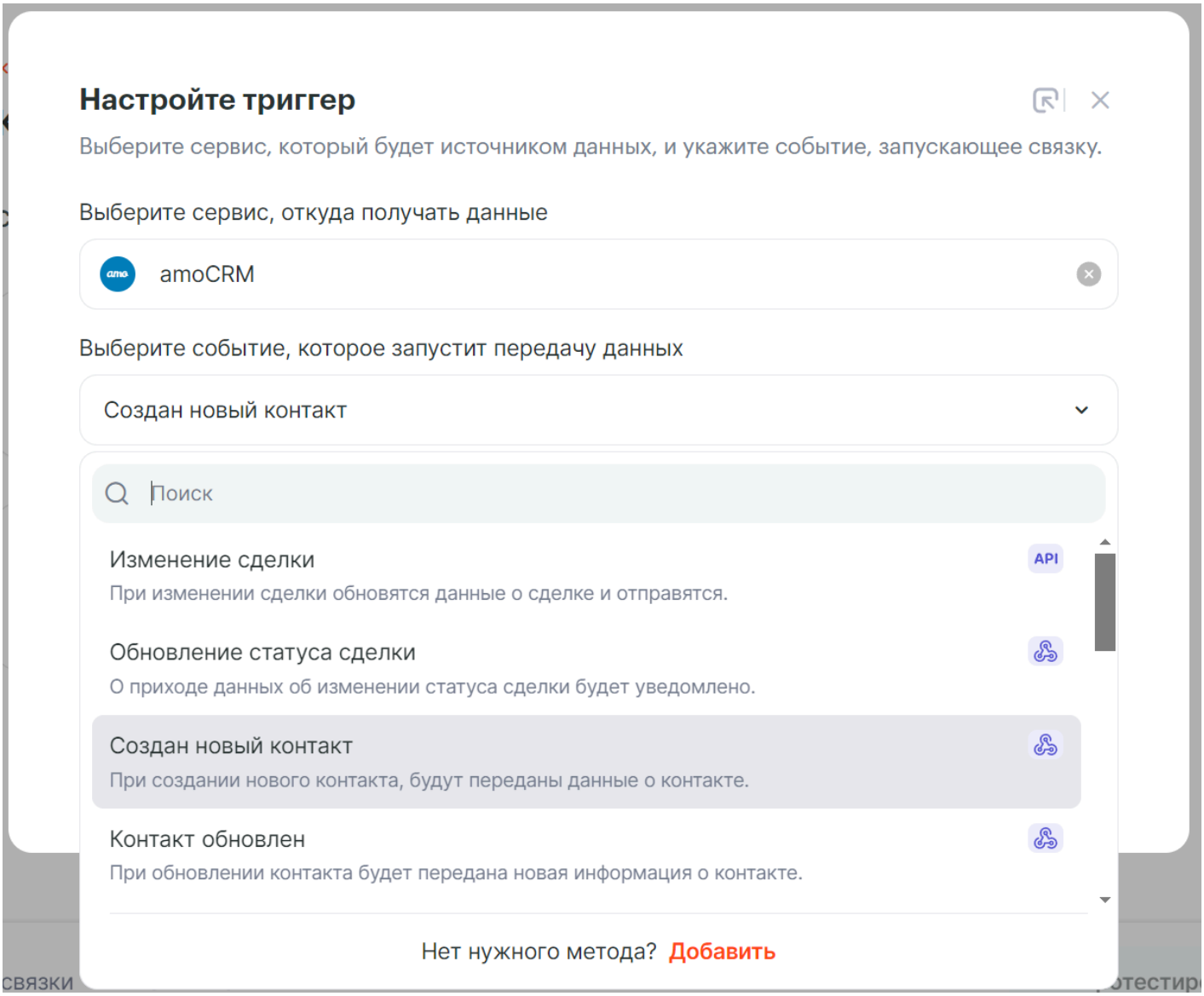


Добавление триггера

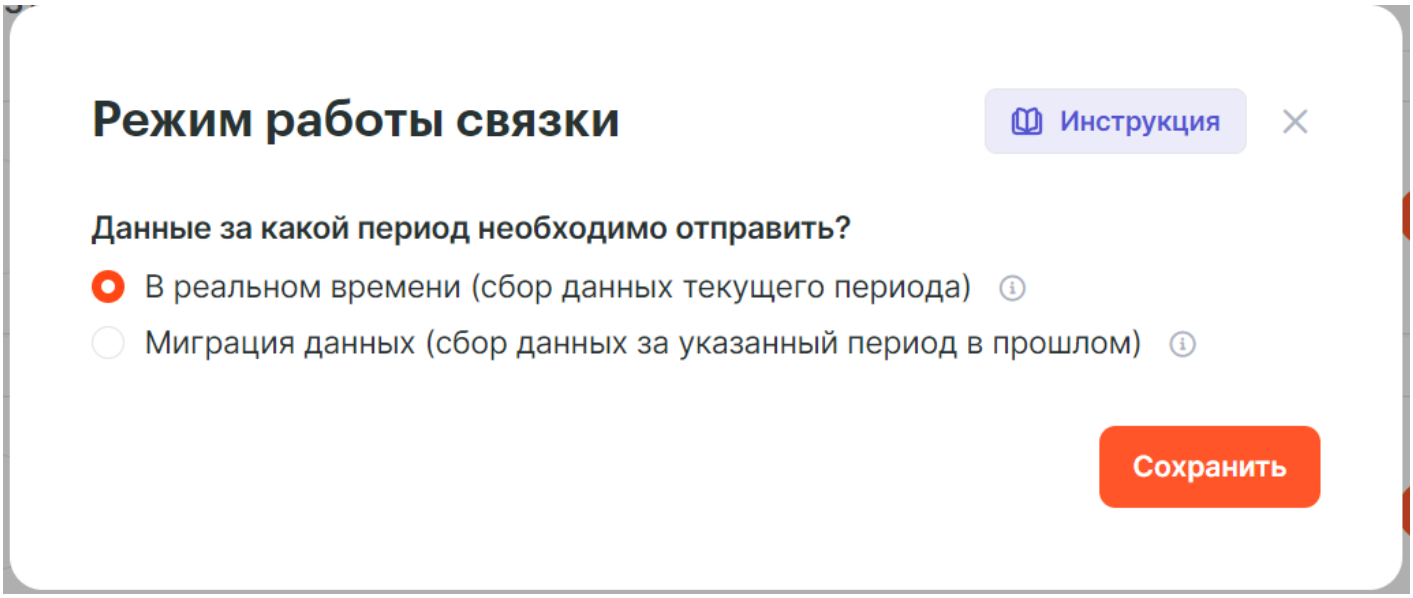
Выберите сервис, откуда Albato будет получать данные:



Выберите событие, которое запустит передачу данных. В нашем случае, действием будет создание нового контакта в amoCRM:

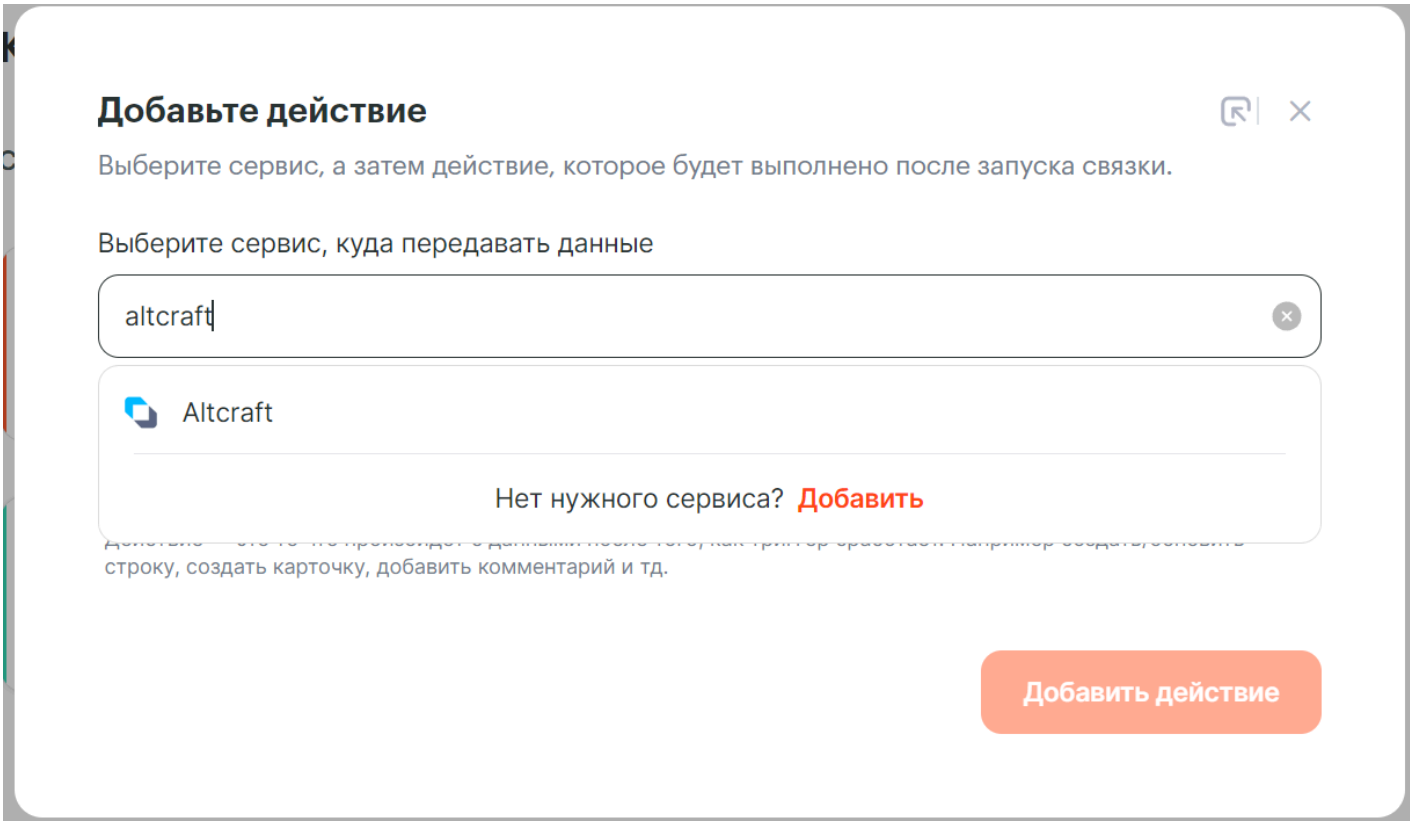


Выберите режим работы связки: данные можно отправлять в Altcraft в реальном времени или указать, за какой период их собрать:

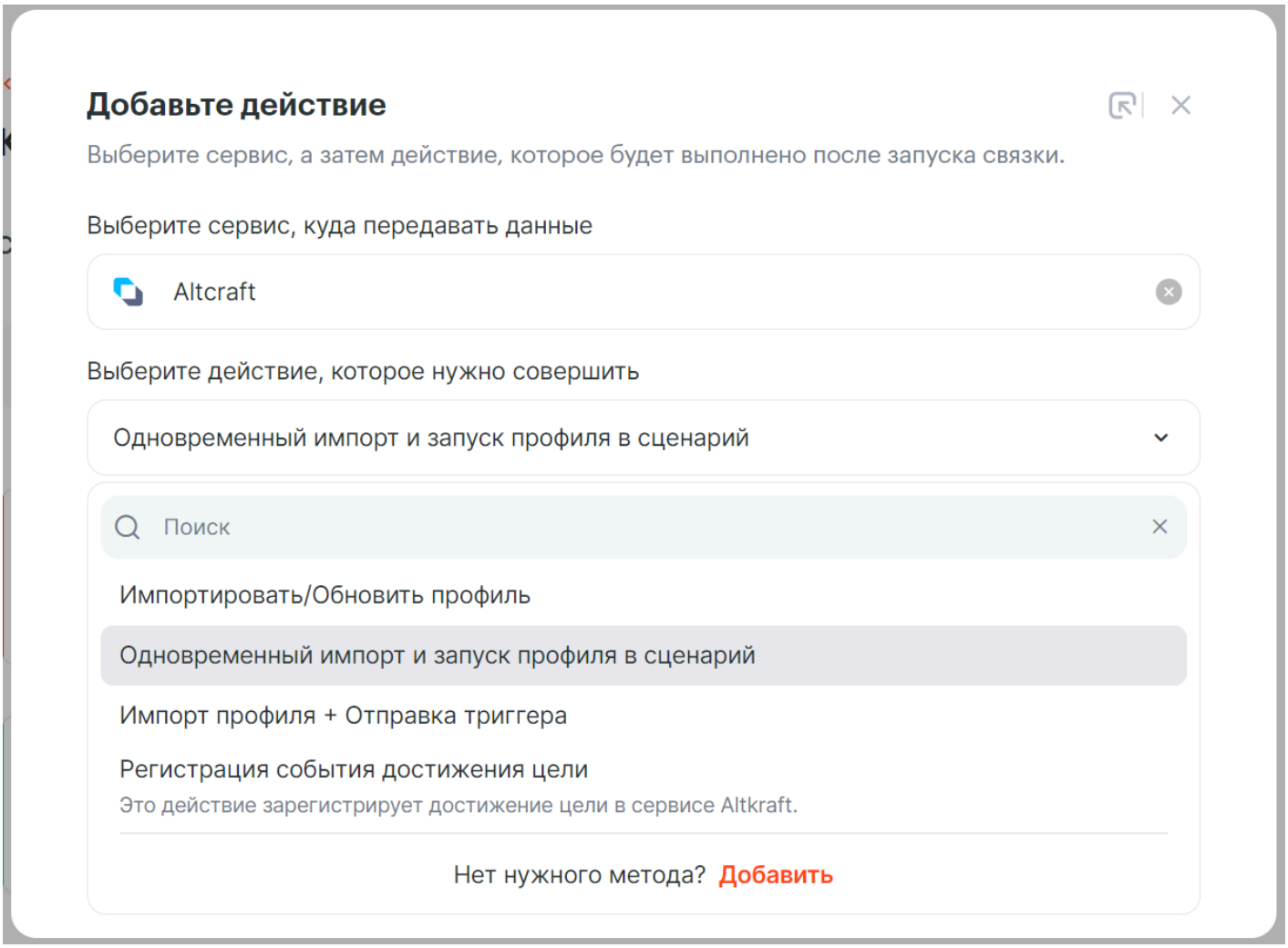


Добавление действия

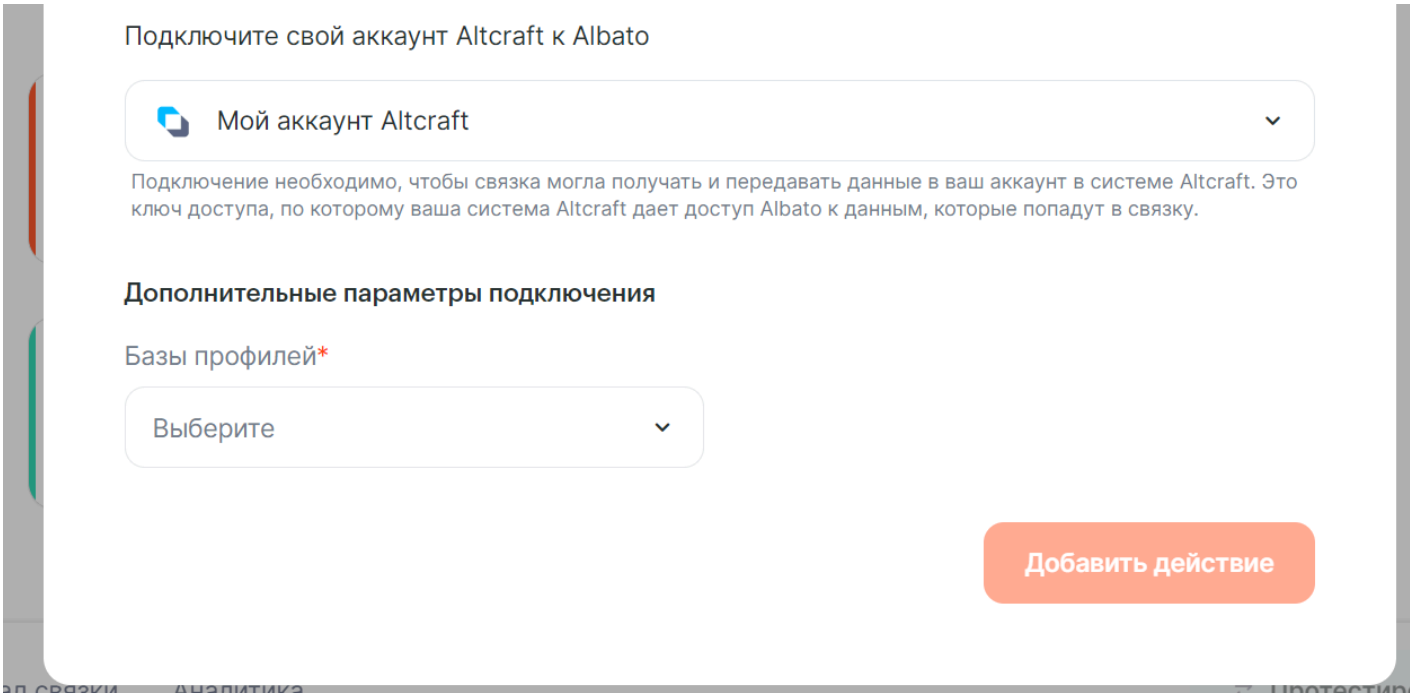
Выберите платформу **Altcraft** как сервер, который будет получать данные из amoCRM:



Для того чтобы импортировать или обновить данные о клиенте и перенести профиль в сценарий, выберите **“Одновременный импорт и запуск профиля в сценарий”:**

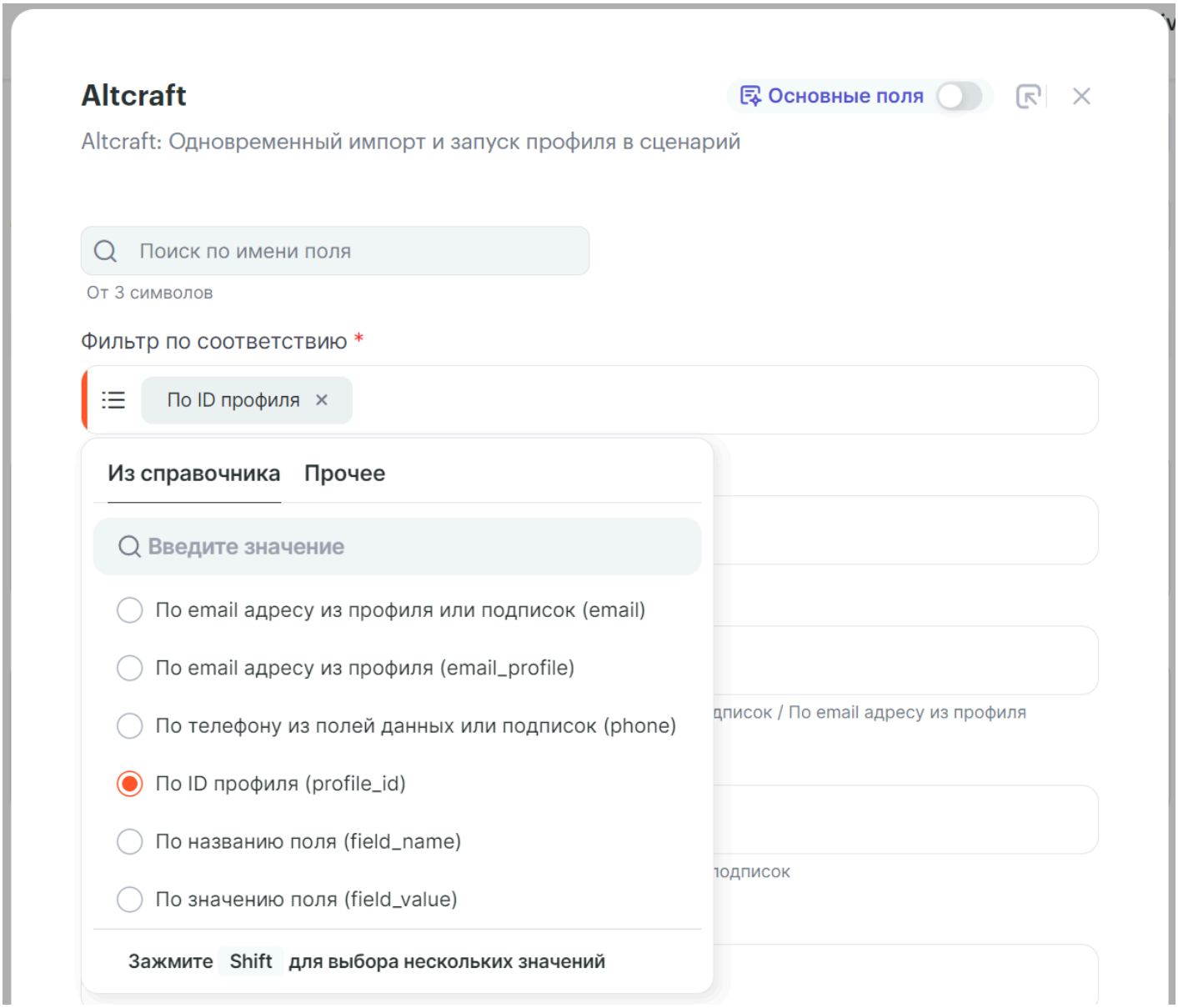


Подключите аккаунт Altcraft и укажите базу данных для взаимодействия. Нажмите **“Добавить действие”:**.

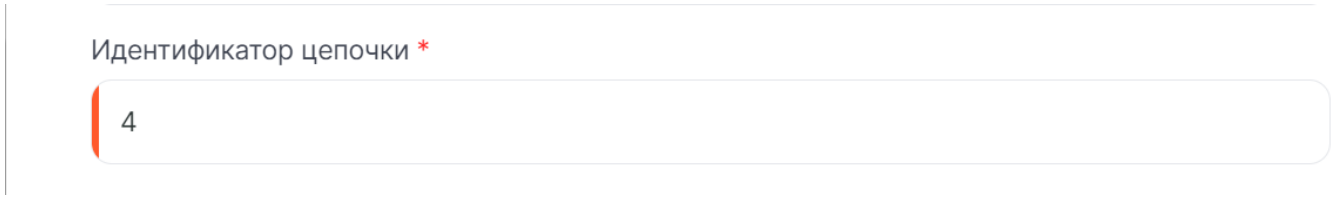


Настройка полей

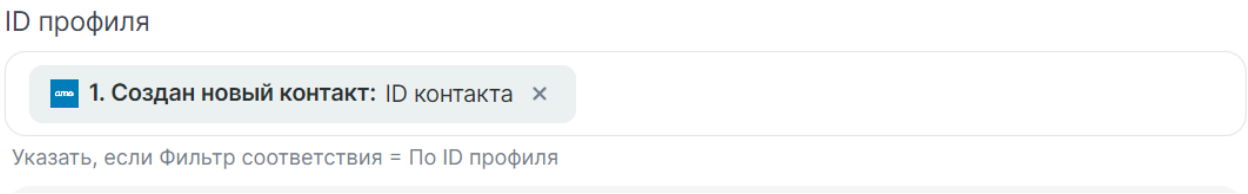
Укажите фильтр, по которому будет осуществляться поиск профиля в базе данных:



Укажите **ID сценария,** в который будет отправлен профиль:



В зависимости от выбранного фильтра для матчинга укажите поле, по которому будет осуществляться поиск:



Опционально укажите дополнительные поля для соответствия. После настройки вы можете протестировать или запустить связку.

Работа со сценарием на платформе

После создания связки данные о клиенте будут переноситься через Albato на платформу, когда будет наступать триггерное событие. Профиль клиента попадет в заранее подготовленный приветственный сценарий.

Подробнее о том, как на платформе регистрируются события для возврата данных на Albato, вы можете прочитать в следующей статье.

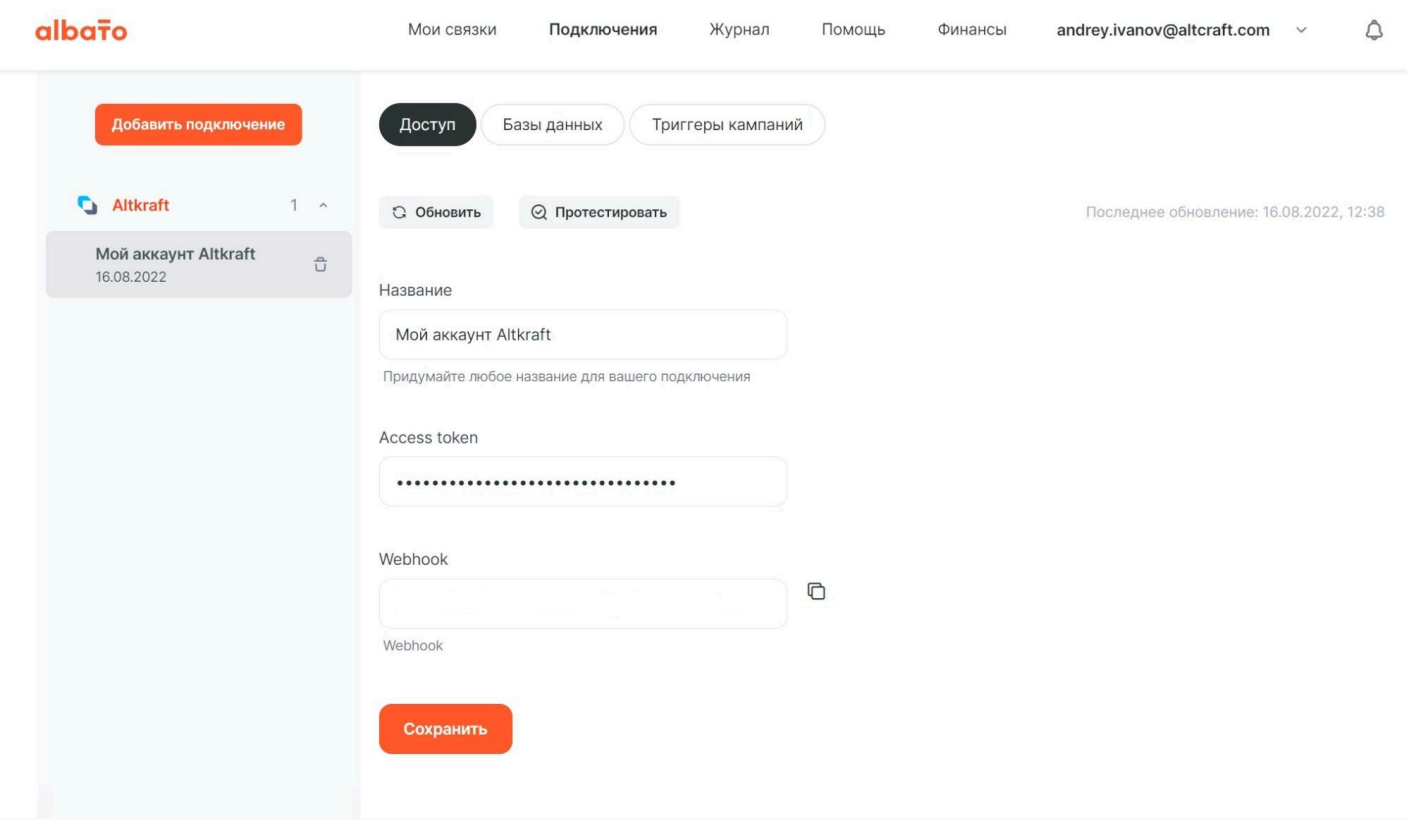
Передача данных о событии

Событие — это действие, связанное как с доставкой сообщений или импортом профилей подписчиков, так и действия клиентов, отслеженные пикселями платформы Altcraft на Вашем сайте или в приложении.

Данные о событиях можно передать в различные сервисы с помощью захвата событий в том числе в Albato.

Настройте захват событий на платформе

Основные этапы настройки захвата событий описаны в этой статье. В поле URL укажите адрес, который находится в **Albato** —> **Подключения** —> **Altcraft** —> **Мой аккаунт Altcraft**

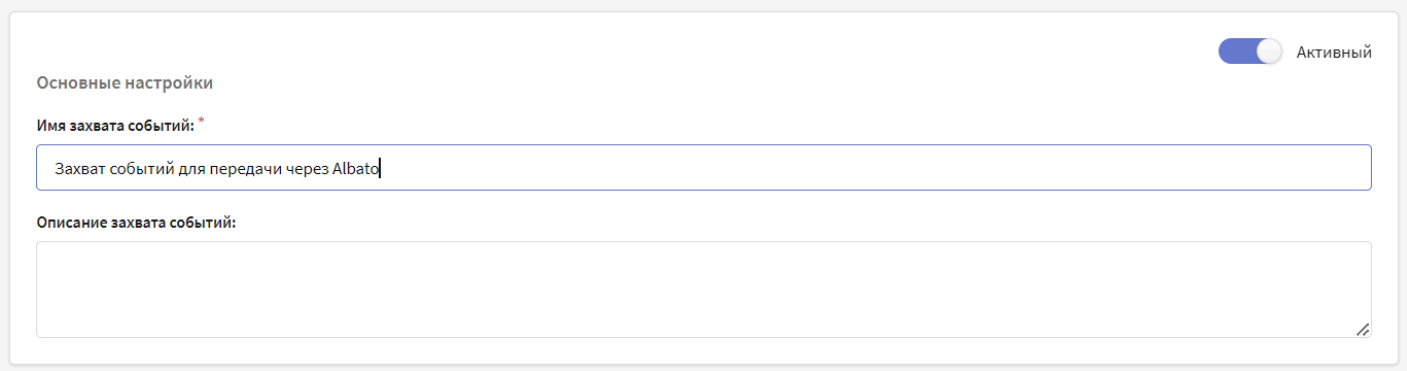


Список доступных событий:

- Открыт AK-Pixel на сайте
- Отправщик не смог доставить Email-сообщение подписчику
- Запущена рассылка
- SMS-сообщение отправлено для доставки подписчику
- Неуникальный клик по ссылке в Email сообщении
- SMS-сообщение доставлено
- Клик по ссылке подтверждения подписки в double opt-in сообщении
- SMS-сообщение не доставлено
- Подписчик перешел по ссылке отписки в письме и был отписан
- Push-сообщение отправлено для доставки подписчику
- Не уникальное открытие Push-сообщения
- Push-сообщение доставлено
- Не уникальный клик по ссылке в SMS сообщении
- Не уникальный клик по ссылке в Push сообщении

- Промокод активирован
- Неуникальное открытие Email-сообщения
- Email-сообщение отправлено для доставки подписчику
- Подписчику назначен промокод
- Отправщик доставил Email-сообщение подписчику

Активировать захват событий можно с помощью переключателя вверху экрана:

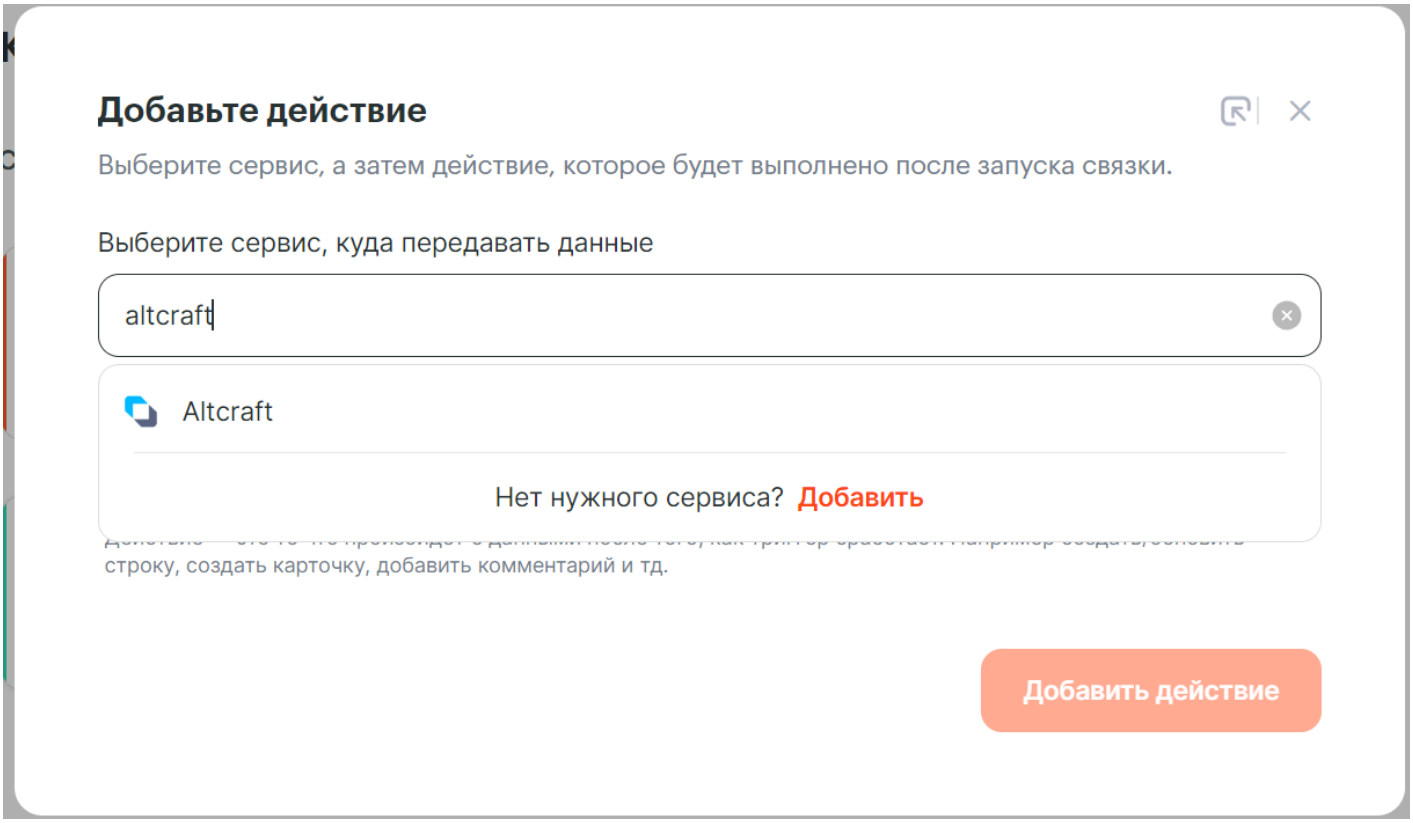


Настройте связку в Albato

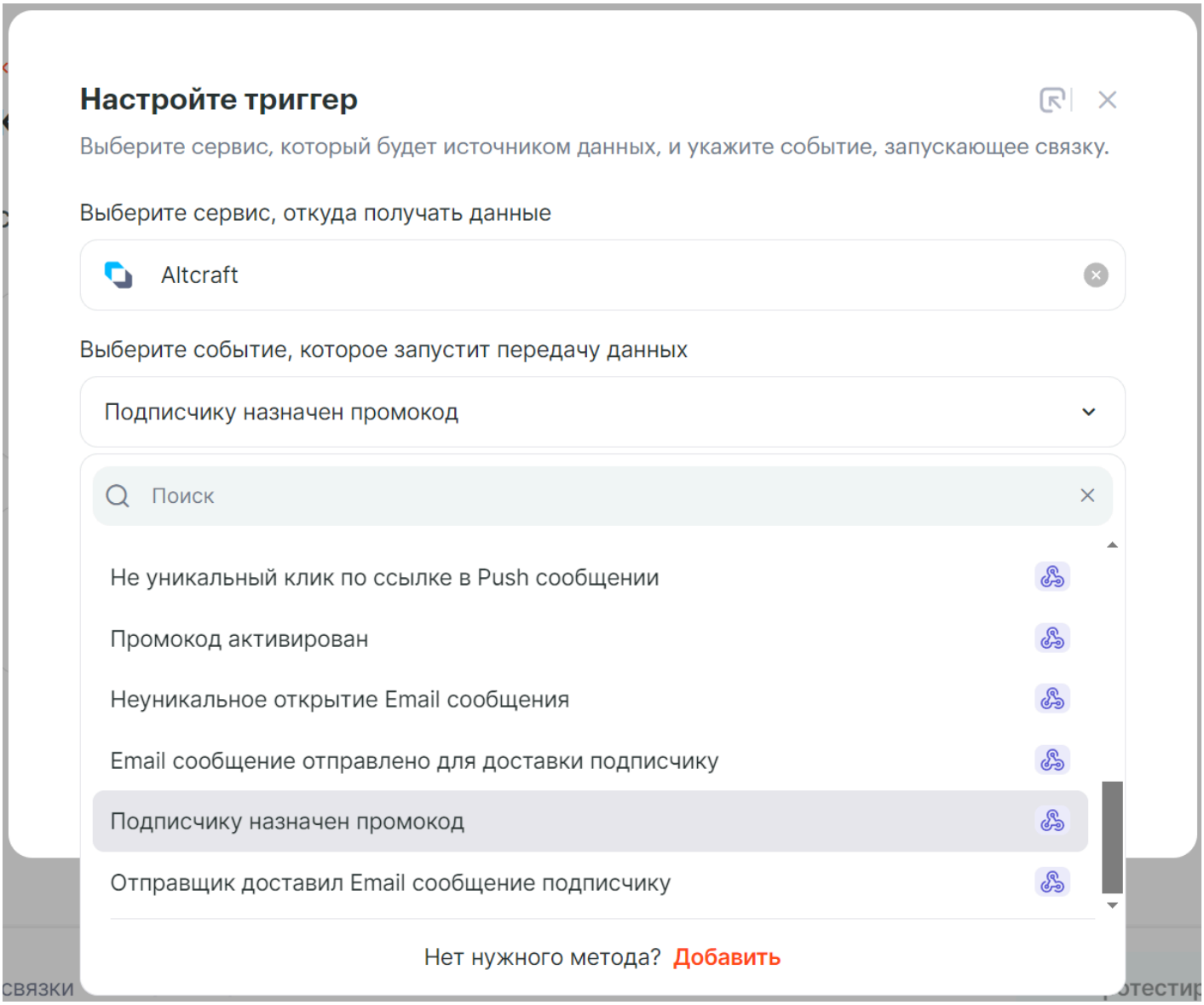
Для того чтобы настроить связку для передачи данных из Altcraft во внешний сервис, нажмите **“Создать связку”**

Настройте триггер

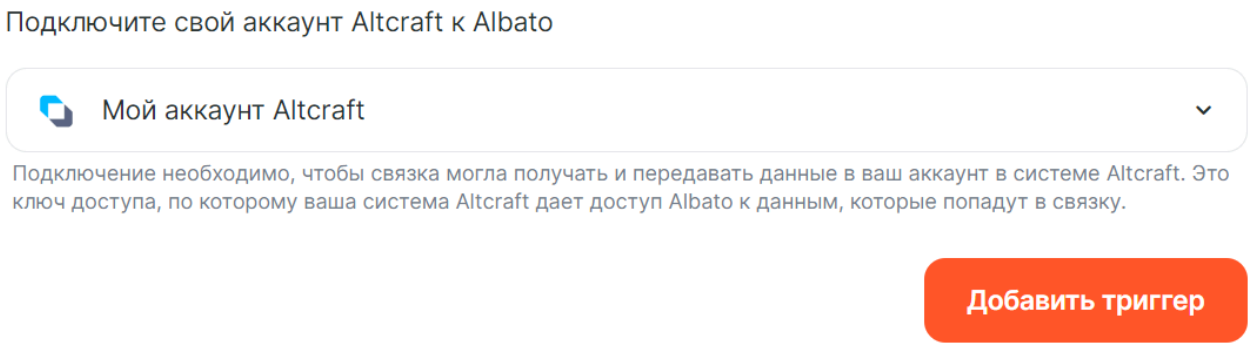
Выберите **Altcraft** в качестве сервиса, который будет передавать данные в Albato:



Выберите событие, наступление которого запустит передачу данных в Albato:

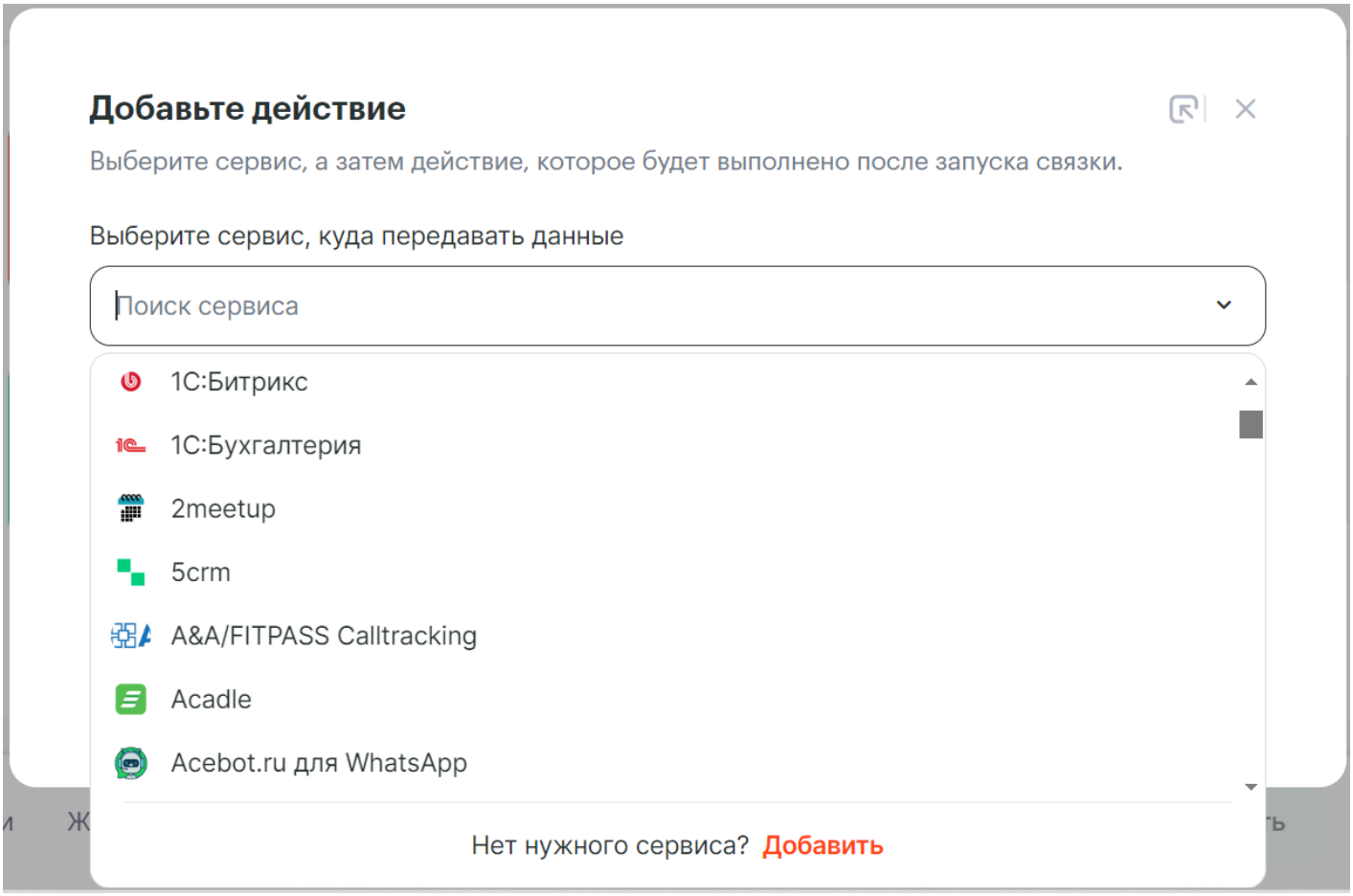


Подключите свой аккаунт:

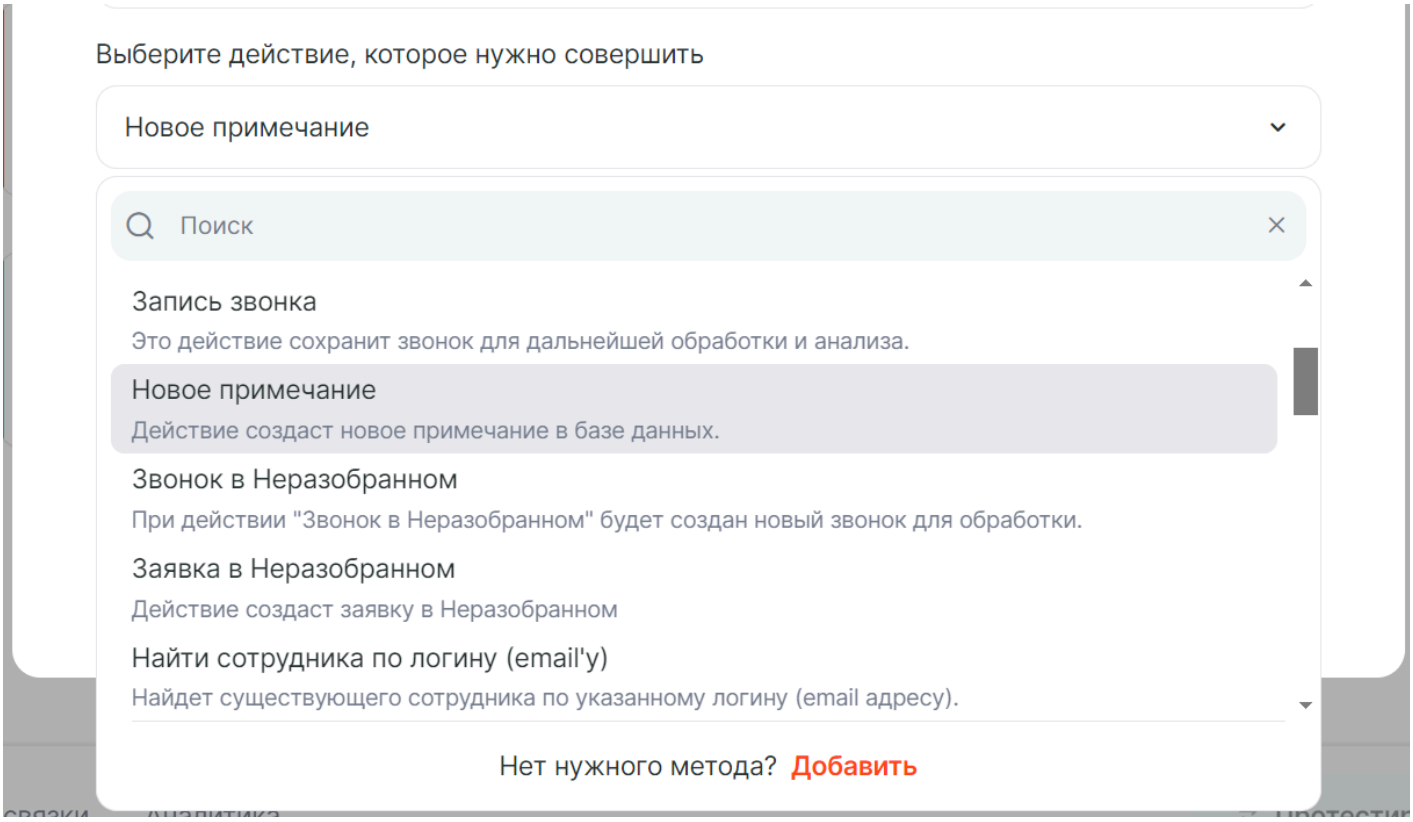


Добавьте действие, которое будет происходить после запуска триггера

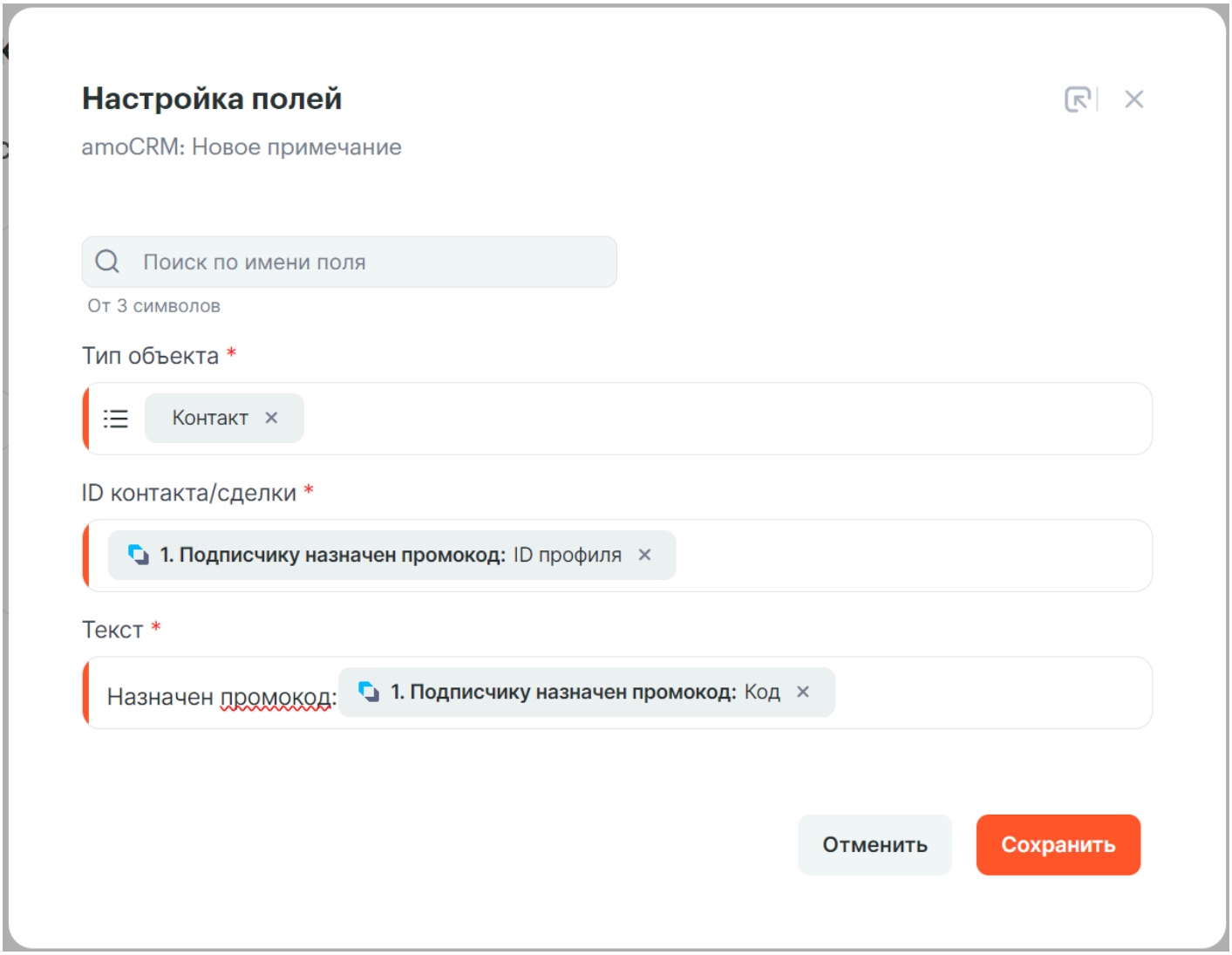
Выберите сервис, куда будут поступать данные:



Назначьте событие, которое будет совершено:



Подключите свой аккаунт к Albato. Нажмите сохранить и после настройте поля:



Активировать связку можно, нажав “Запустить”. После этого привязанные к событию на платформе данные будут отображаться в карточке клиента во внешнем сервисе.

Отправка триггерной рассылки

С помощью Albato вы можете создать связку которая будет запускать триггерную рассылку на платформе.

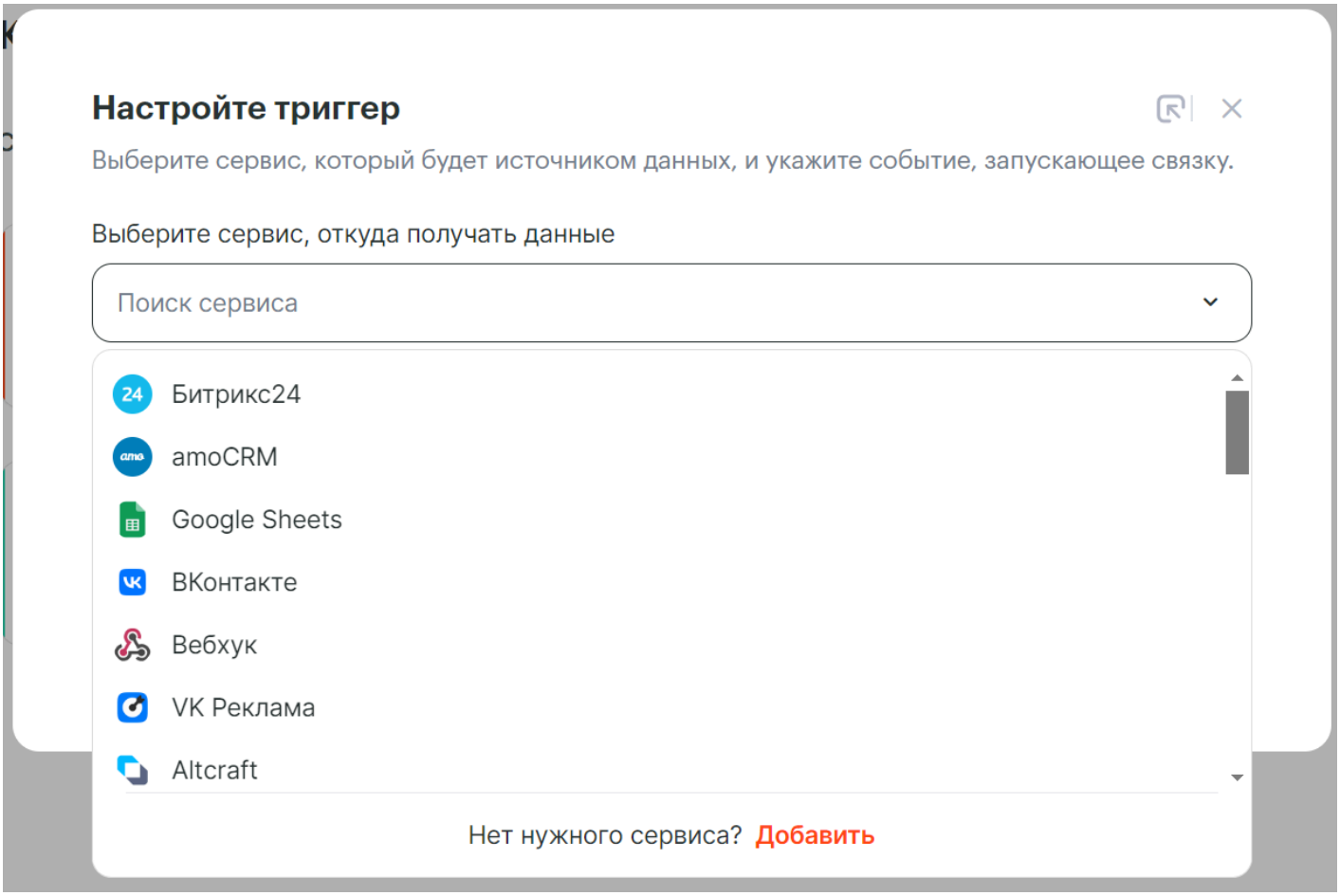
Рассмотрим запуск данного типа рассылки на примере отправки счета клиенту.

Настройте связку в Albato

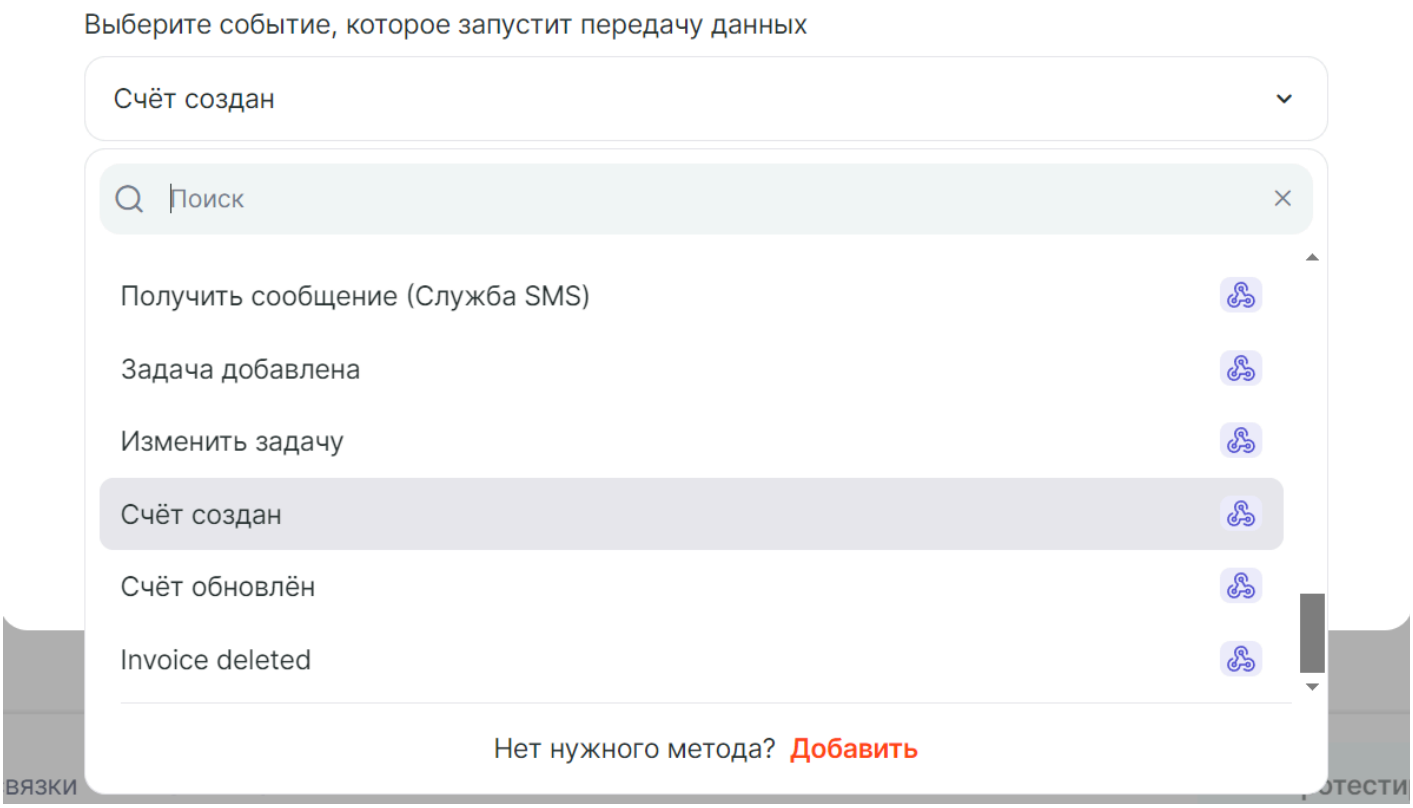
Настройте связку **Внешний сервис** —> **Albato** —> **Altcraft**

Добавьте триггер

Выберите внешний сервис, с которого будут поступать данные:

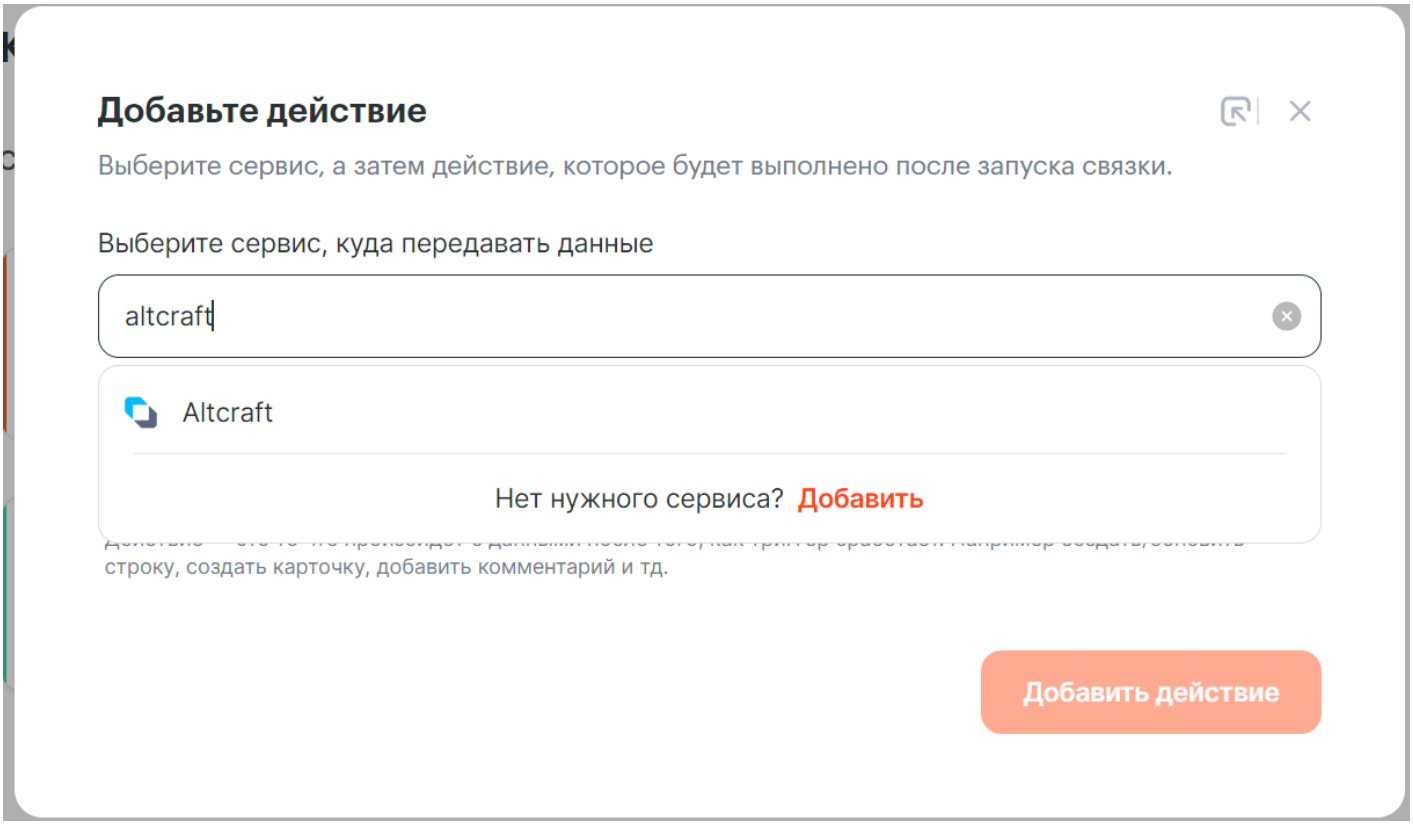


Выберите событие, которое запустит передачу данных:

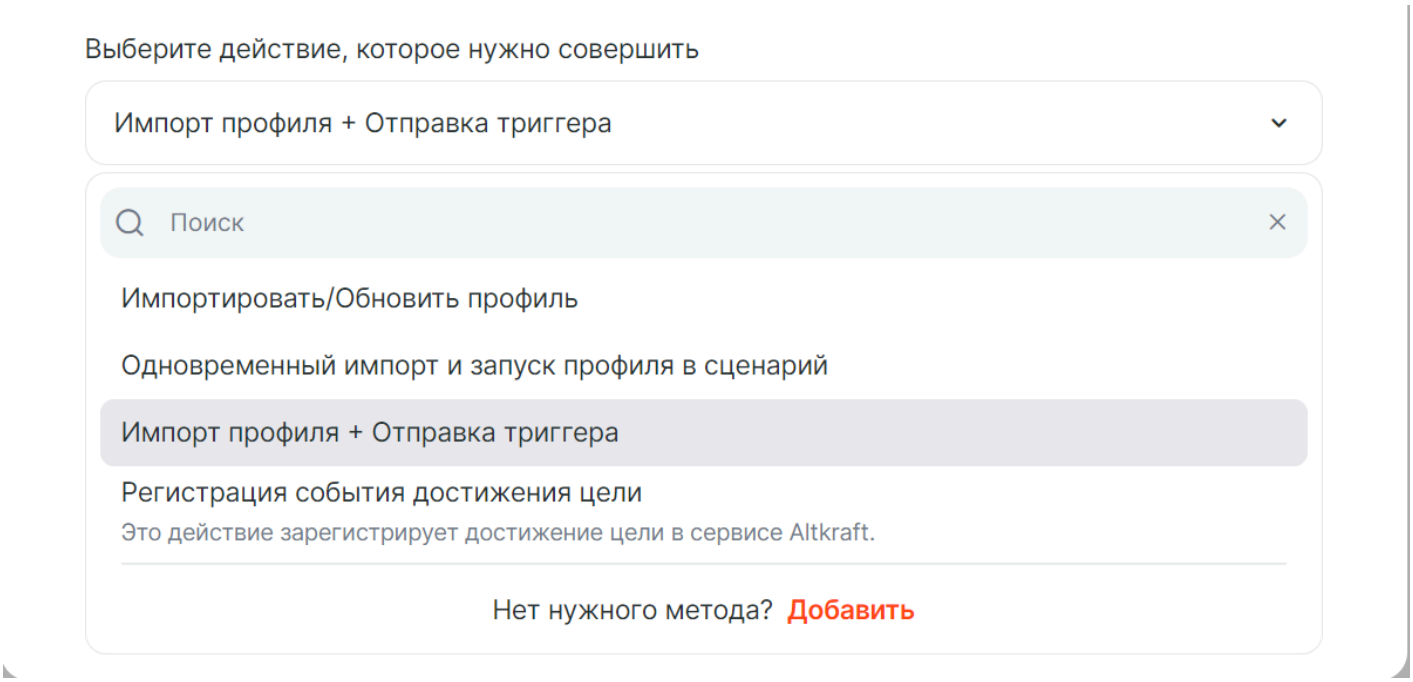


Добавьте действие

Укажите **Altcraft** как сервис для получения данных:

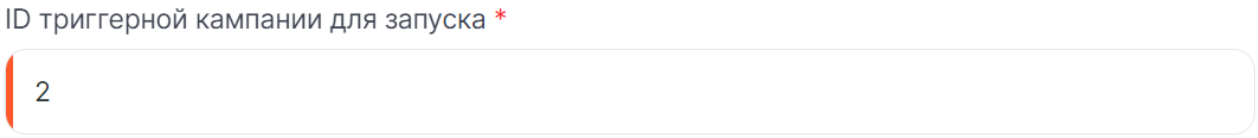


Назначьте действие, которое будет происходить после наступления события:

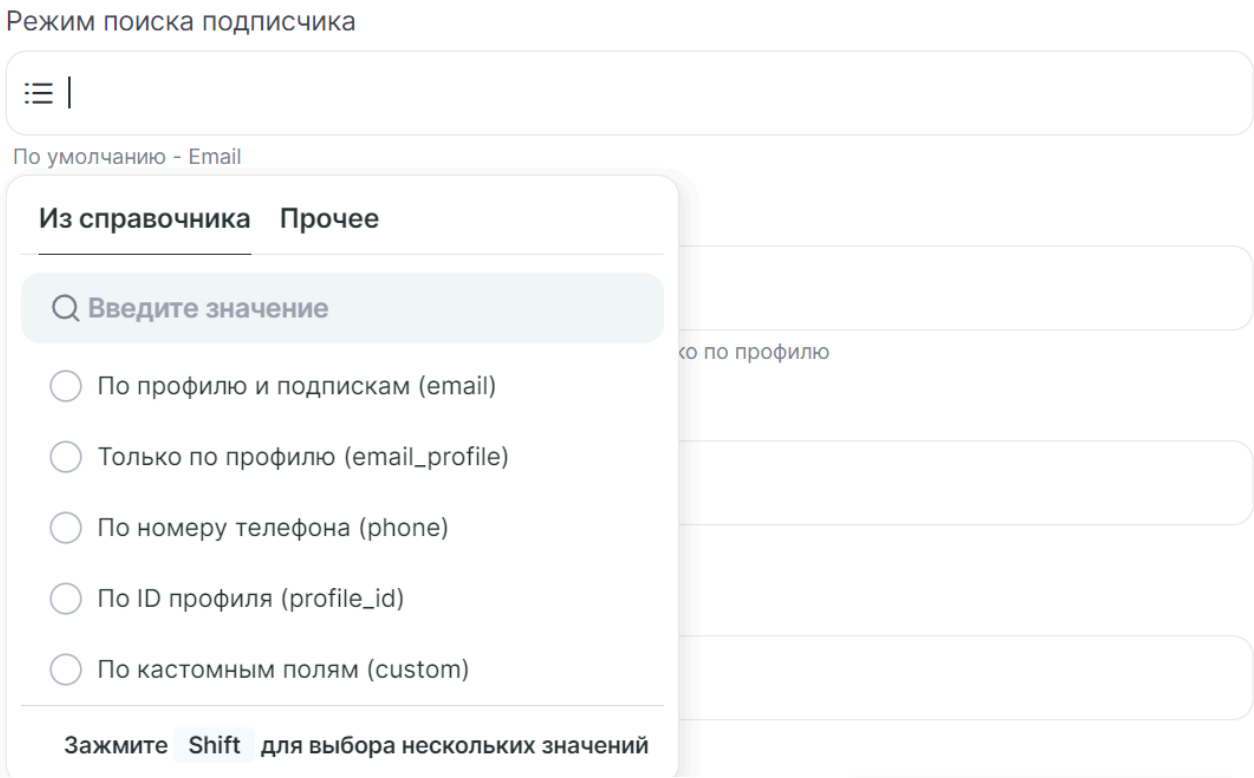


Подключите свой аккаунт и выберите базу профилей.

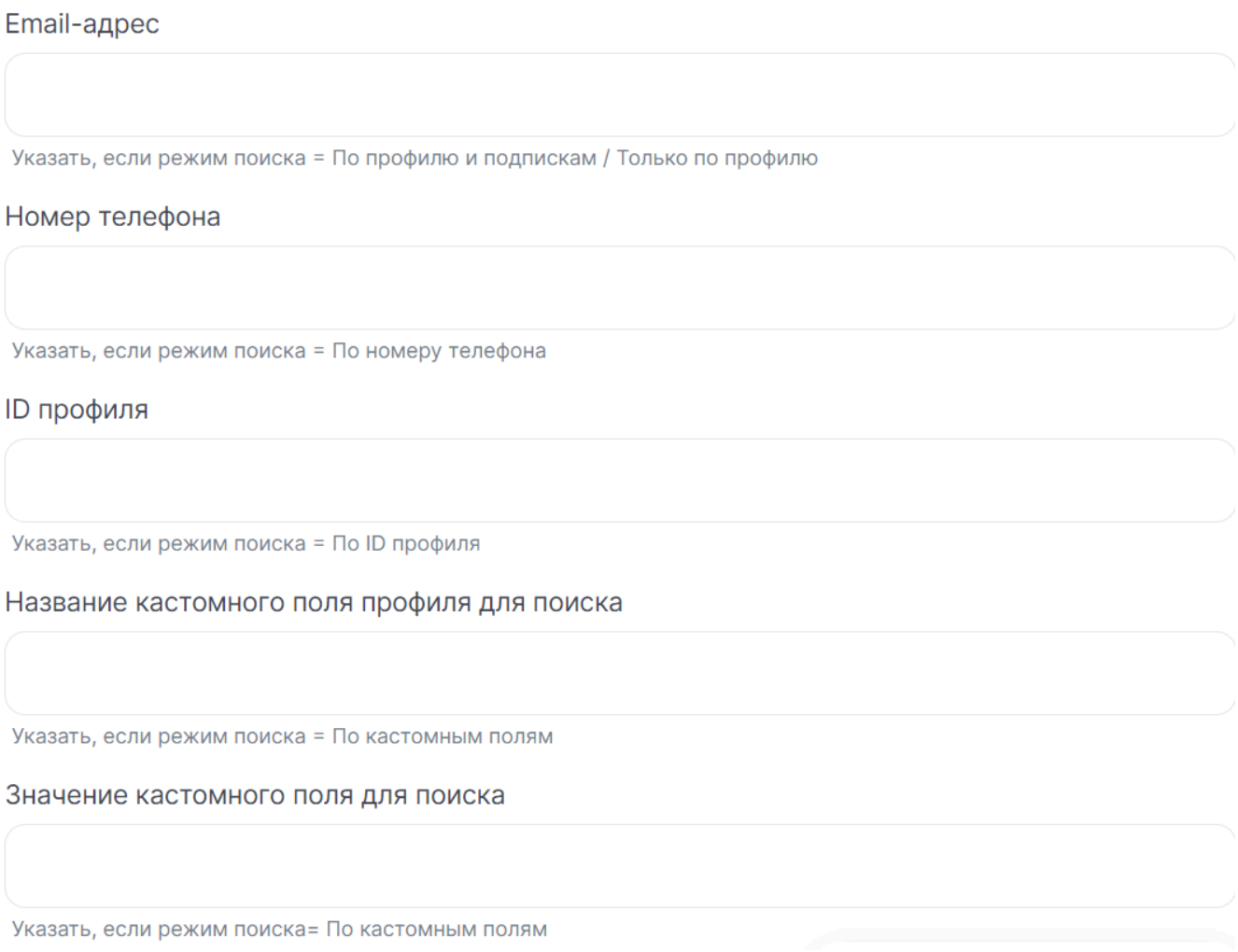
Далее укажите ID триггерной рассылки, которую вы хотели бы запускать:



Выберите параметр, по которому подписчик будет находиться в базе данных:



В зависимости от выбранного параметра укажите поле для поиска:

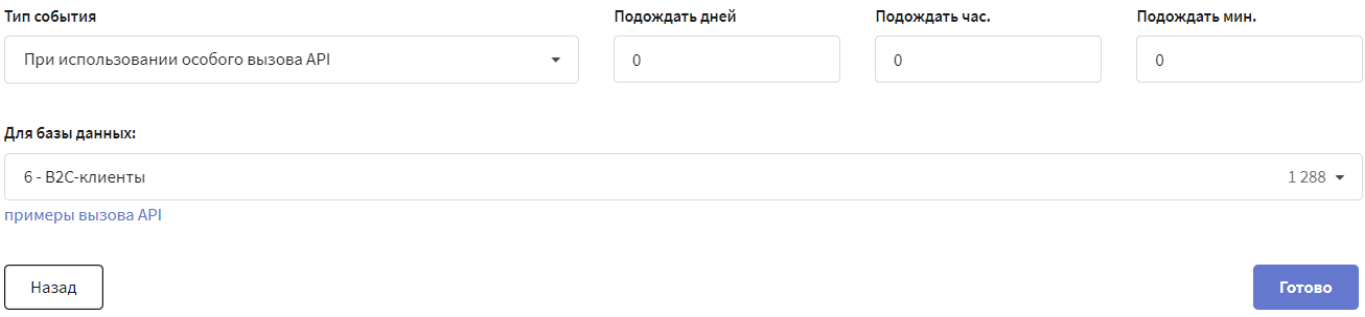


Опционально укажите поля для матчинга.

Настройте триггерную рассылку на платформе

Для начала необходимо настроить триггерную рассылку.

В разделе **Триггер** укажите **тип события**: **При использовании особого вызова API** (принимает вызовы API из внешних сервисов). Выберите базу данных и, опционально, время ожидания:



Сохраните и запустите рассылку.

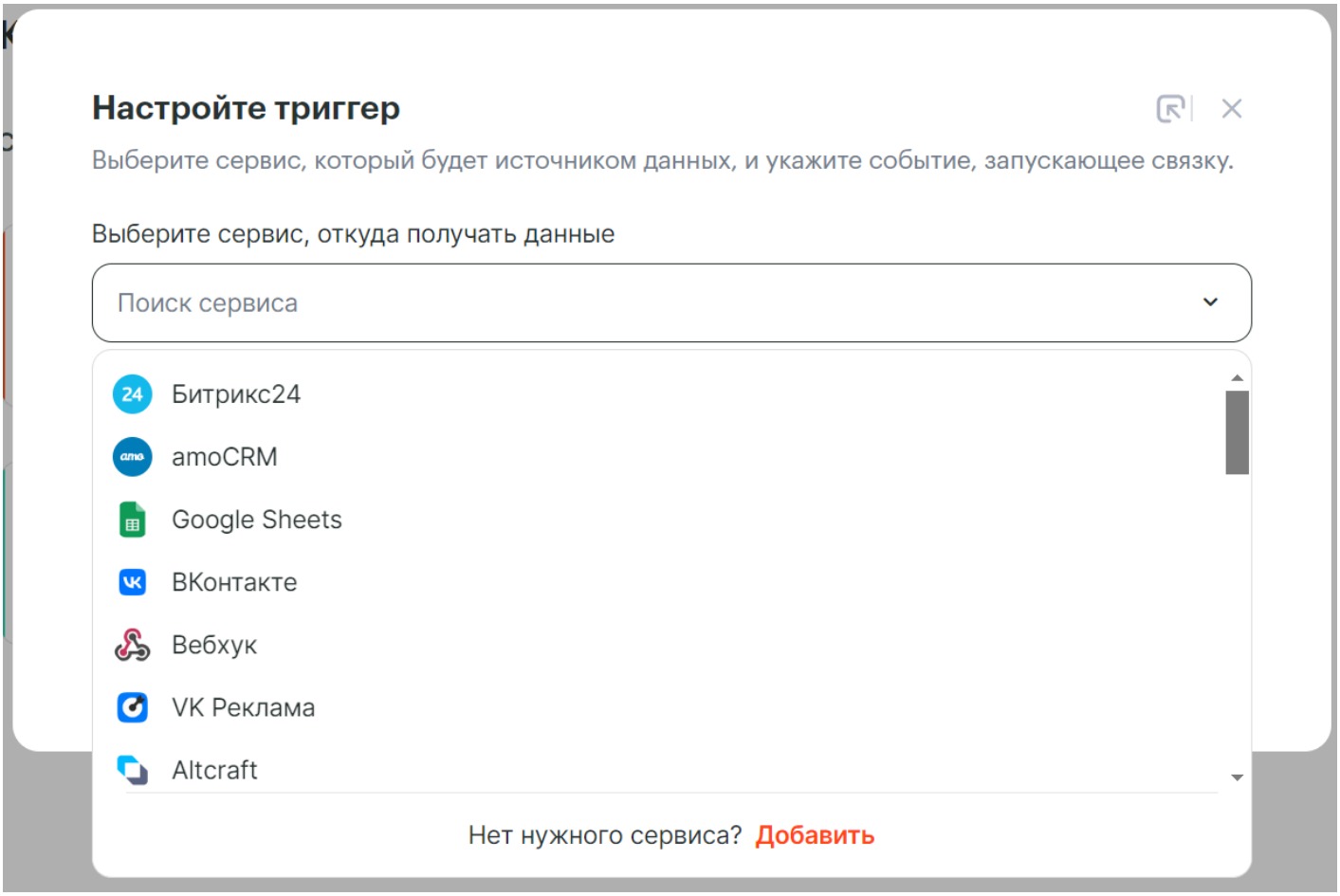
Регистрация событий

Albato позволяет вам переносить данные о клиенте в профиль подписчика на платформе.

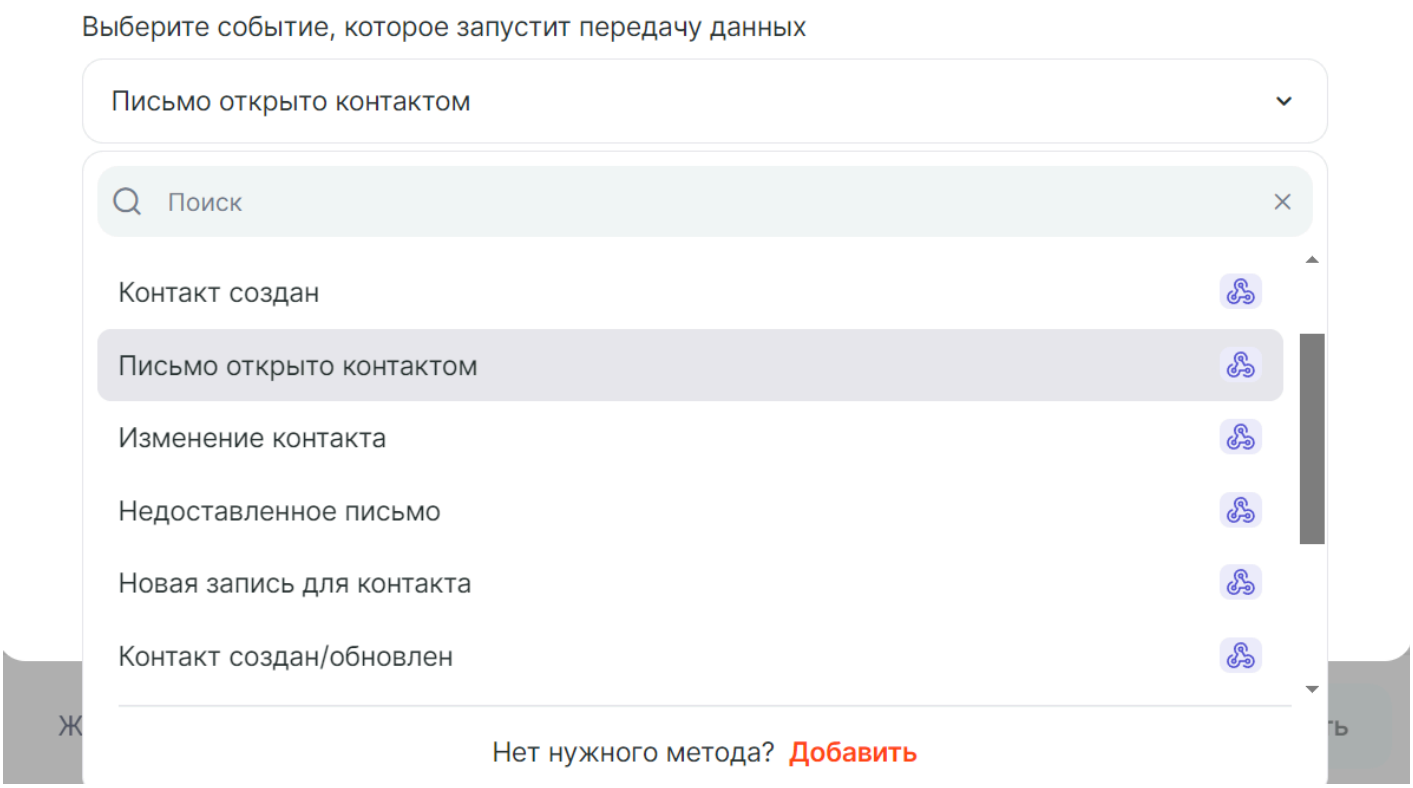
Для этого необходимо настроить связку **Внешний сервис** —> **Albato** —> **Altcraft**

Настройте триггер

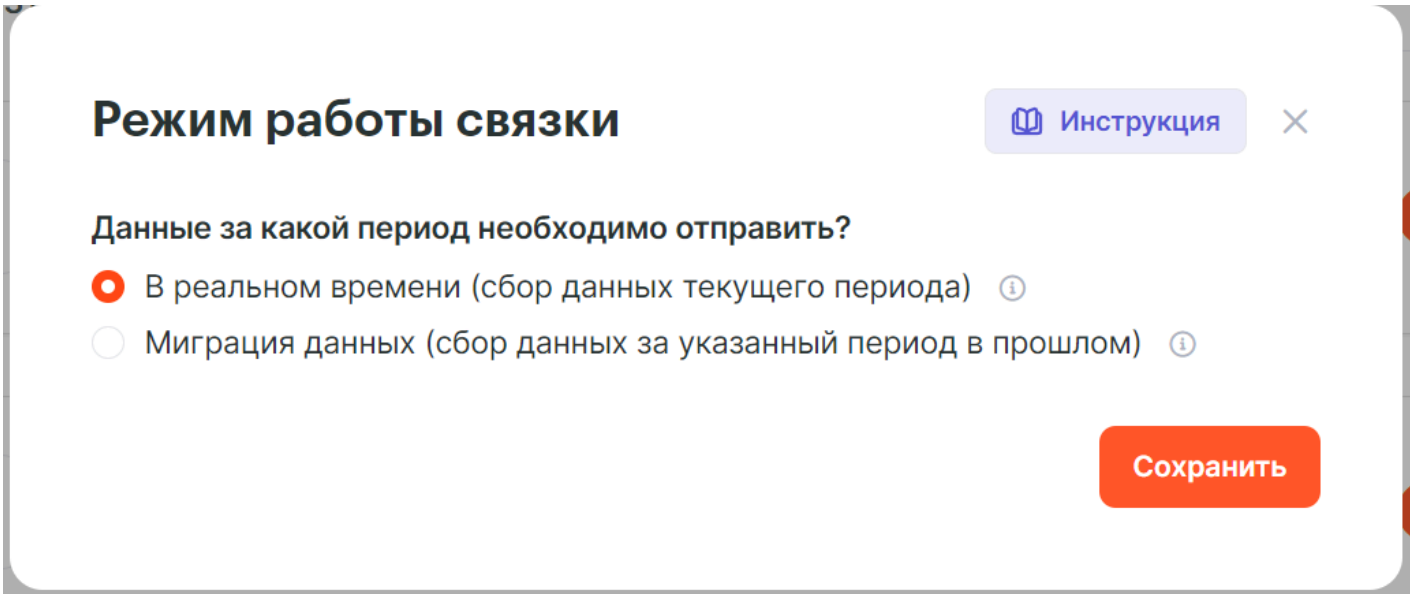
Укажите внешний сервис, который будет передавать данные на платформу:



Выберите событие, которое запустит передачу данных:

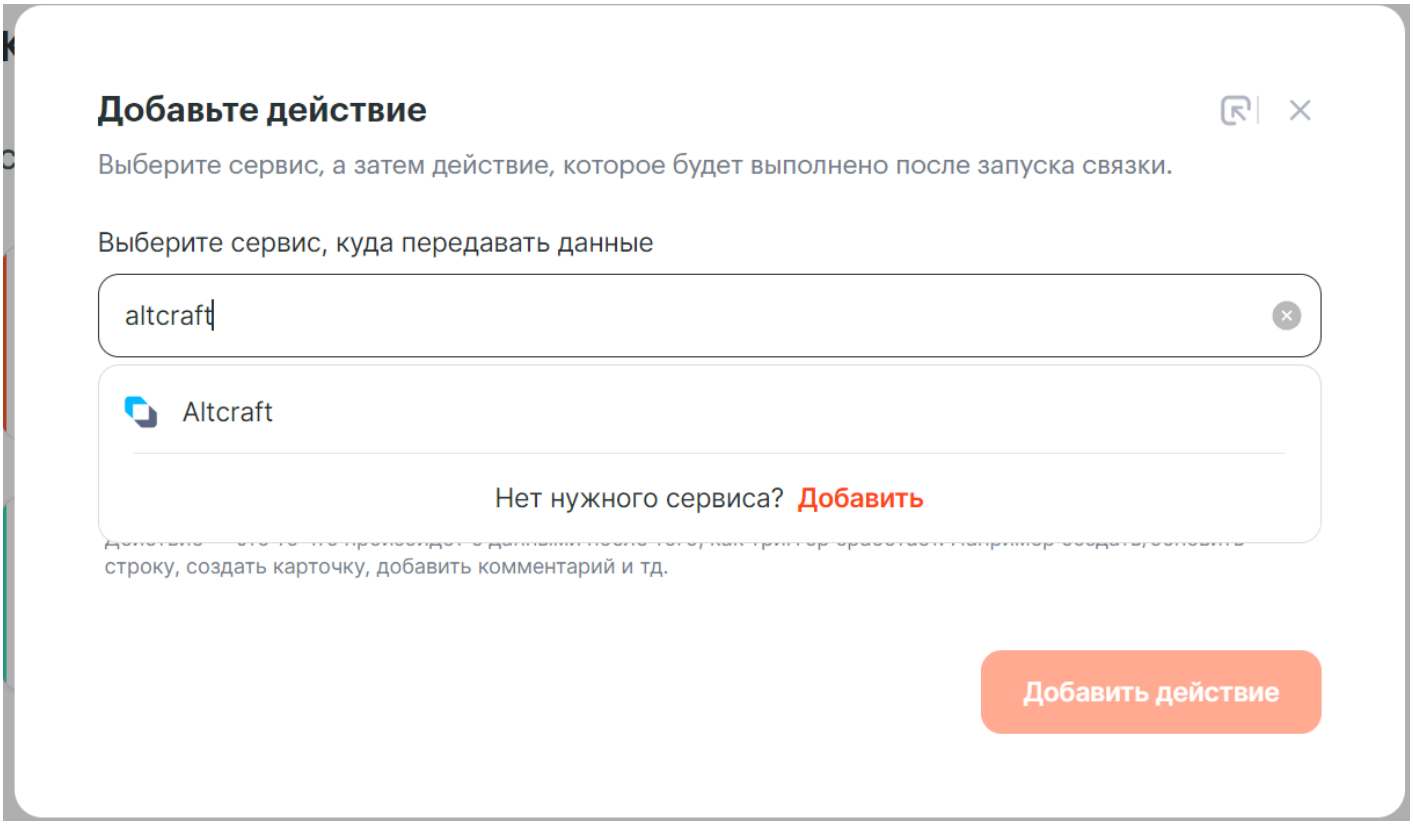


Выберите режим работы связки: данные можно отправлять в **Altcraft** в реальном времени или указать, за какой период их собрать:

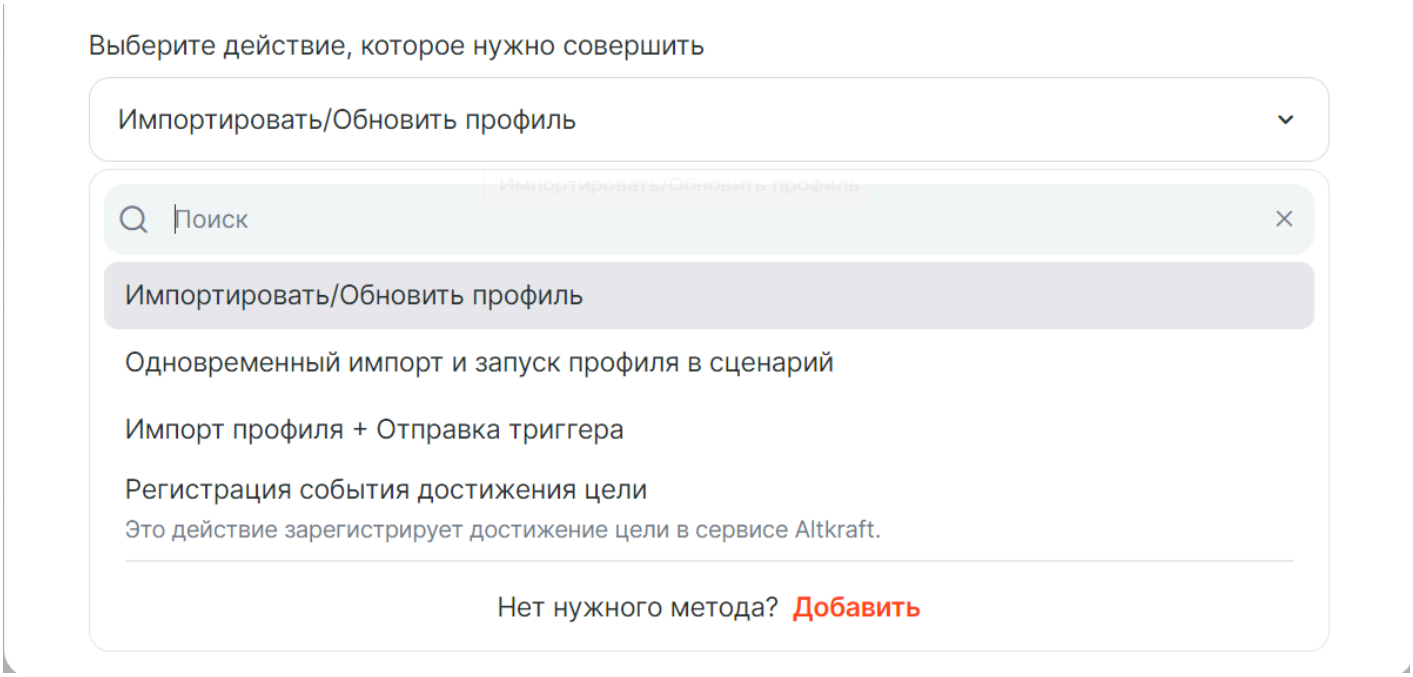


Настройте действие

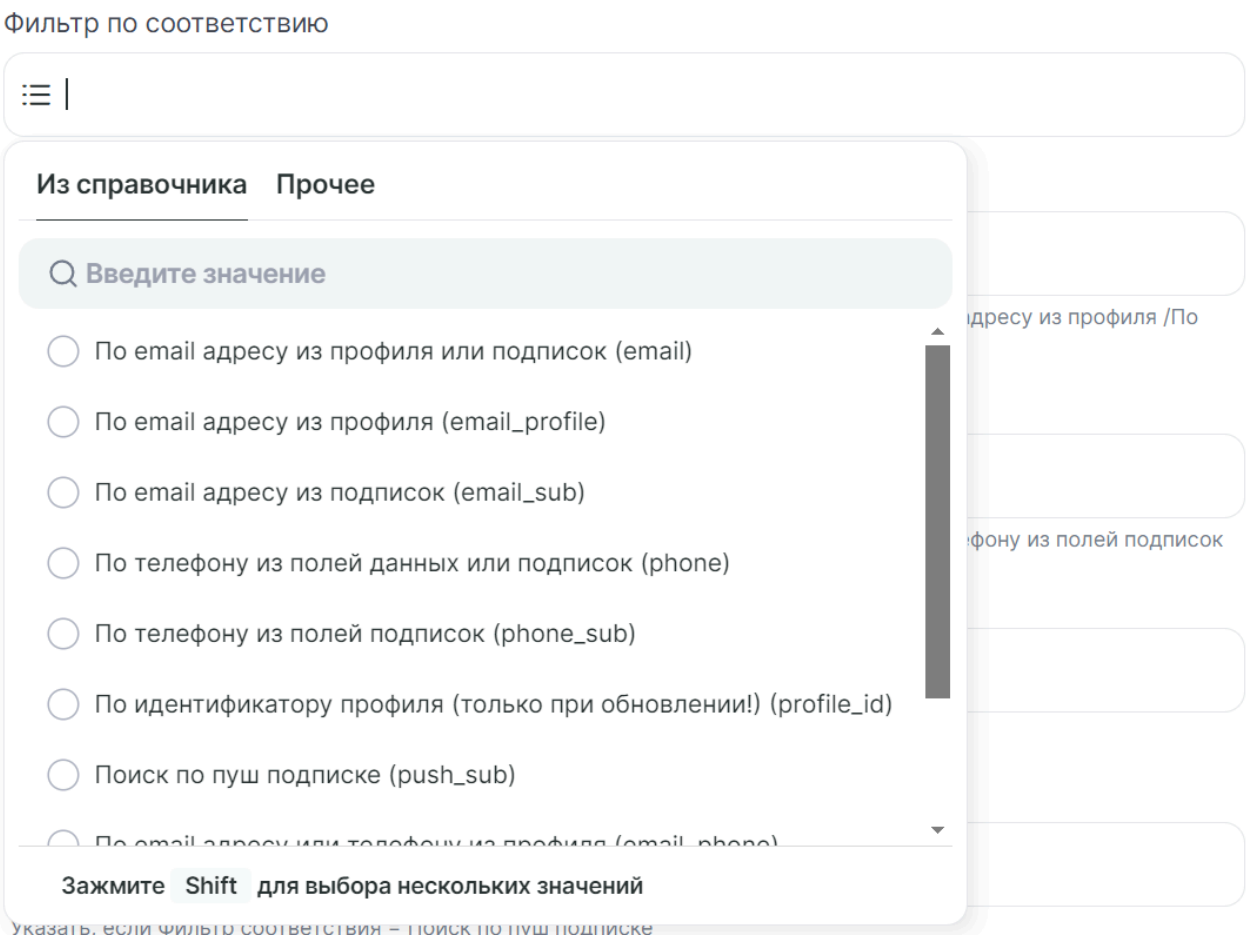
Укажите **Altcraft** как сервис для получения данных:



Назначьте действие, которое будет происходить после наступления события. Для того чтобы данные импортировались в профиль подписчика, выберите **Импортировать/Обновить профиль**:



Настройте фильтр соответствия:



Опционально настройте дополнительные поля.

После сохранения связки, данные будут передаваться в карточку профиля клиента при срабатывании триггера.

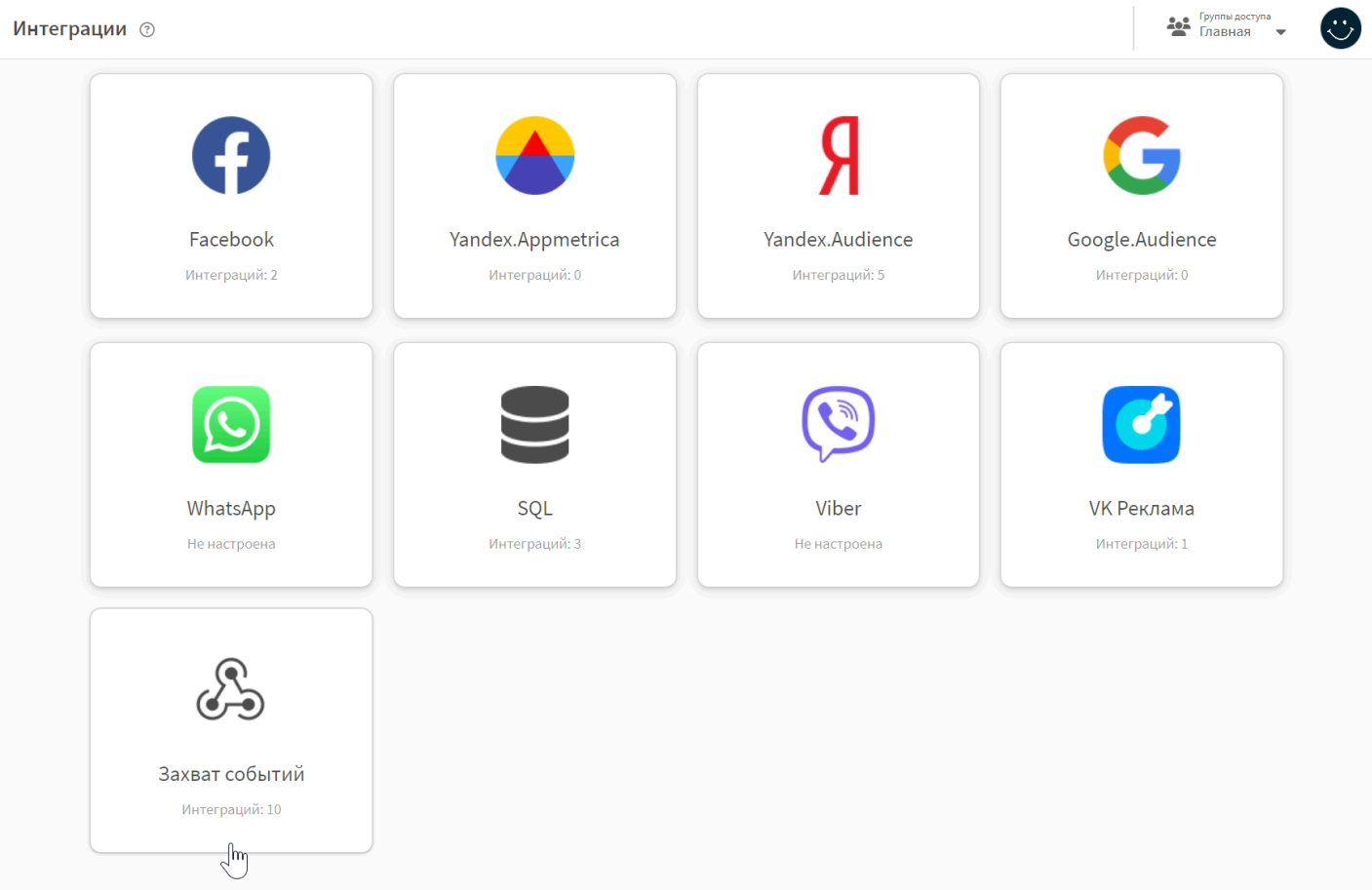
Захват событий Altcraft

Захват событий (Action Hooks) — это механизм уведомления внешних систем о событиях платформы. Это могут быть как системные действия, связанные с доставкой сообщений или импортом профилей подписчиков, так и действия клиентов, отслеженные пикселями платформы Altcraft на Вашем сайте или в приложении.

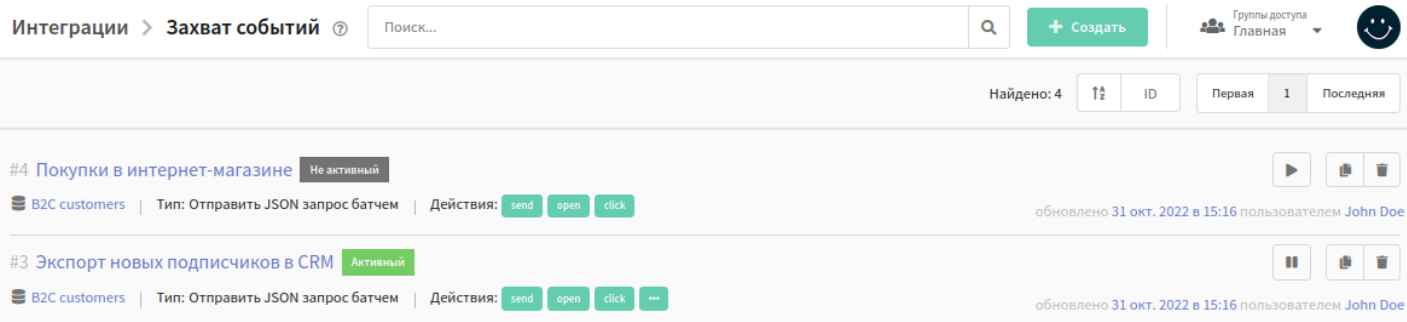
Захват событий может работать со всеми **базами данных** или с какой-либо конкретной. Захваченные события могут передаваться **батчем** в формате JSON по HTTP или отправляться в очередь брокера **асинхронно**.

Конфигурируется мастер-пользователем аккаунта для определенной базы данных или для всех сразу.

Чтобы начать работу, в главном меню выберите **Интеграции** → **Захват событий**:

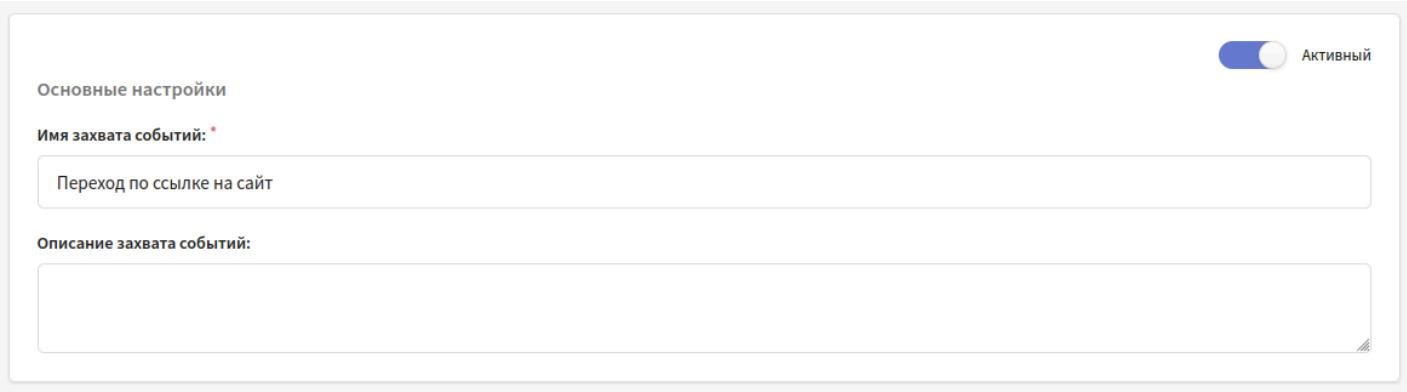


Существующие захваты событий вы можете активировать или остановить, клонировать или удалить с помощью контекстного меню. Если вы хотите отредактировать захват, нажмите на его заголовок. Чтобы добавить новый, нажмите **+ Создать** в верхней части страницы:



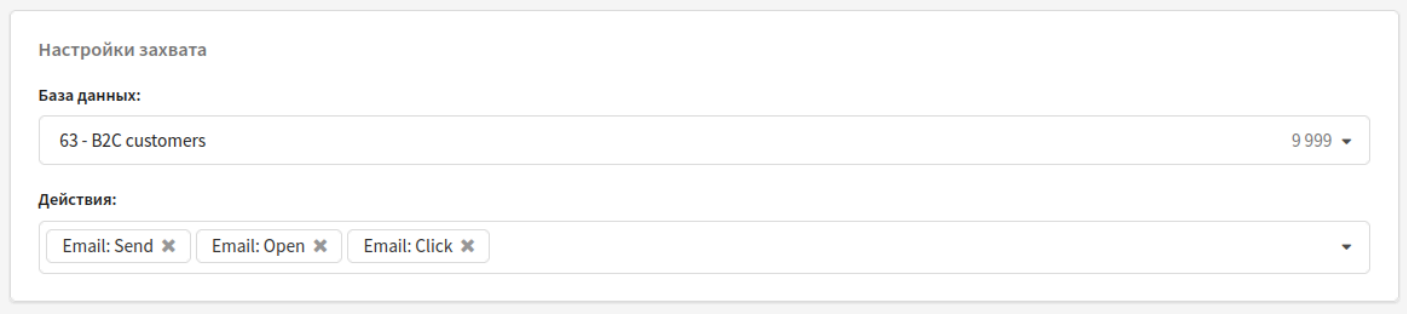
Основные настройки захвата событий

В **Основных настройках** введите имя нового захвата. Здесь же находится переключатель для активации процесса отправки событий:



Выбор действий для захвата

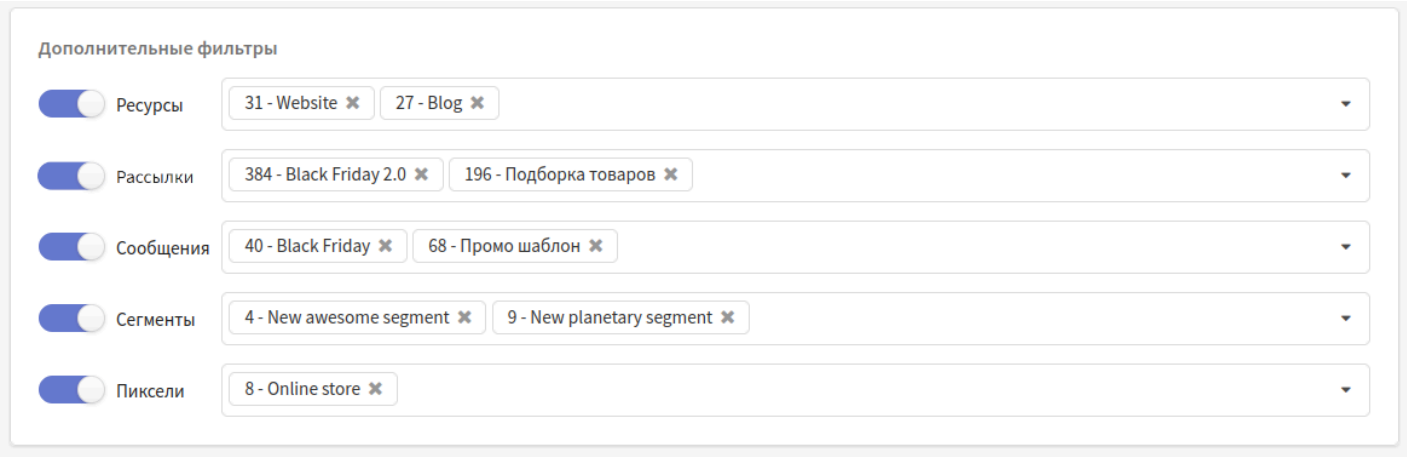
Захваты событий могут работать со всеми Базами профилей сразу или с одной конкретной. В блоке **Настройки захвата** выберите необходимый вариант в выпадающем меню **База данных**:



В поле **Действия** выберите из выпадающего меню события, которые будут передаваться во внешние системы. Типы событий, их обозначения в меню и в запросе захвата, а также описания вы найдёте в этой статье.

По умолчанию Захват регистрирует все события выбранных вами типов. Ограничить события так, чтобы получать более узкую выборку, можно в блоке **Дополнительные фильтры:**

- **Ресурсы** — будут захвачены события, связанные с коммуникациями в рамках определённых ресурсов.
- **Рассылки** — будут захвачены события определённых маркетинговых рассылок.
- **Сообщения** — будут захвачены события рассылок, использующих определённые шаблоны сообщений.
- **Сегменты** — будут захвачены события, связанные с подписчиками, входящими в определённые сегменты.
- **Пиксели** — будут захвачены действия подписчиков на внешних ресурсах, зарегистрированные определёнными пикселями.



Настройки обратного запроса

Доступны следующие **типы запроса:**

1. Отправить JSON запрос батчем. Подробнее.
2. Отправить сообщение в очередь RabbitMQ. Подробнее.
3. Отправить сообщение в exchange RabbitMQ. Подробнее.
4. Отправить сообщение в Kafka broker. Подробнее.

INFO

Функция повторной отправки

Если во время отправки сообщения по каким-то причинам произошла ошибка (например, сбой сети), через определенное время сообщение будет отправлено снова. Количество повторных отправок и их периодичность устанавливаются в конфигурационном файле системы “main.json”:

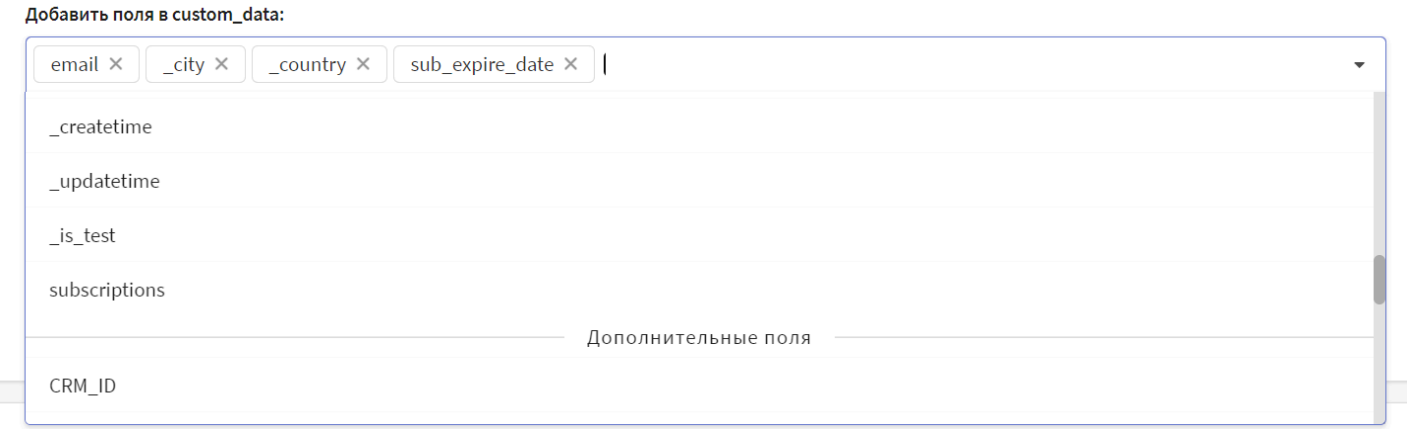
`"HOOK_RETRY_TIMES"`: устанавливает максимальное количество попыток отправки (по умолчанию — 4)

`"HOOK_RETRY_PERIOD_SEC"`: Периоды между попытками отправки, в секундах (по умолчанию — 15)

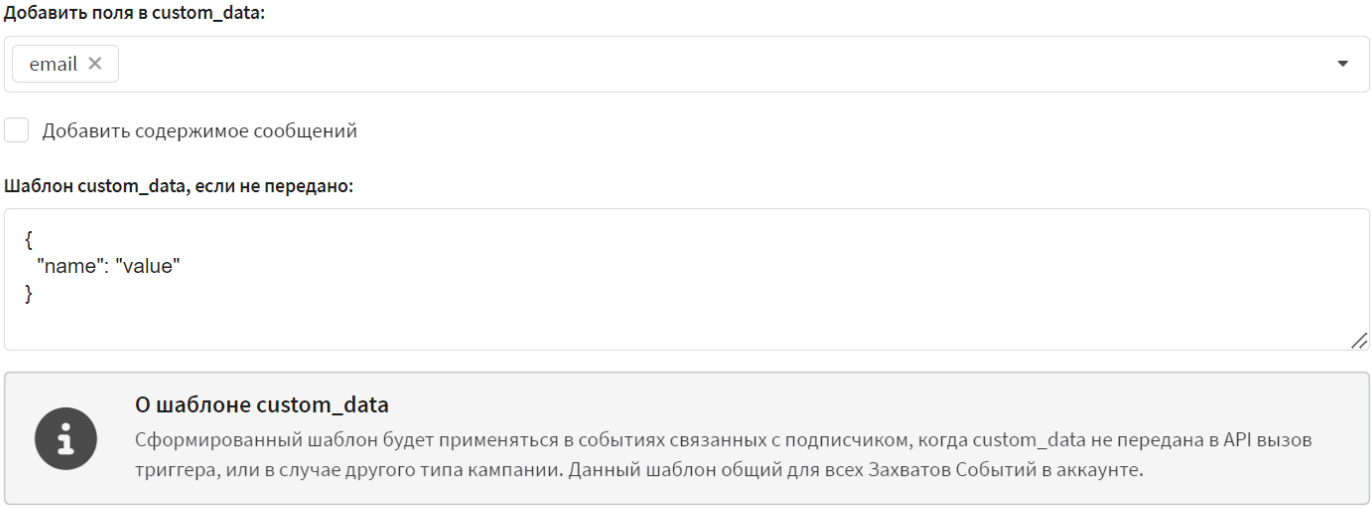
Передача дополнительных данных о событии в захвате

Данные профиля клиента в захвате

Вместе с информацией о захваченном событии можно передавать значения стандартных и дополнительных полей профиля. Набор доступных дополнительных полей зависит от выбранной базы данных. Выберите в выпадающем списке поля профиля, которые хотите **добавить в custom_data:**



Ниже можно задать **шаблон custom_data**. Платформа будет использовать в захват значение из этого шаблона, если в API-запросе на запуск триггера не переданы данные в поле `custom_data`.



Дополнительные данные в захвате

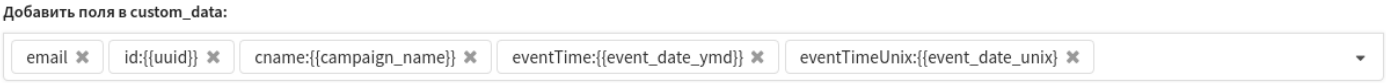
Помимо данных профиля, в объект `custom_data` можно передавать любые данные в формате `mykey:myvalue`. Для этого вручную введите ключ и значение, а затем нажмите Enter ↵.

Доступны шаблонные параметры:

Параметр	Описание
`{{uuid}}`	Генерирует новый UUIDv4
`{{event_date}}`	Дата события в формате “2006-01-25T15:04:05.999999Z”
`{{event_date_ymd}}`	Дата события в формате “2006-01-25”
`{{event_date_rfc3339}}`	Дата события в формате RFC 3339
`{{event_date_unix}}`	Дата события во времени unix, в строке
`{{event_date_unixnano}}`	Дата события во времени unix (наносекунды), в строке
`{{campaign_name}}`	Извлекает имя рассылки, если оно актуально для события, иначе пустое
`{{campaign_subid}}`	Извлекает SubID рассылки, если оно актуально для события, иначе пустое

Параметр	Описание
`{{database_name}}`	Извлекает имя базы данных, если оно актуально для события, иначе пустое
`{{segment_name}}`	Извлекает имя сегмента, если оно актуально для события, иначе пустое
`{{message_name}}`	Извлекает имя шаблона, если оно актуально для события, иначе пустое
`{{message_subject}}`	Извлекает тему сообщения в шаблоне, если оно актуально для события, иначе пустое
`{{resource_name}}`	Извлекает имя ресурса, если оно актуально для события, иначе пустое

Например, вы добавляете в `custom_data` поля:



В этом случае данные будут переданы следующим образом:

```
"custom_data": {
        "cname": "Black Friday",
        "email": "test@example.com",
        "eventTime": "2023-03-17",
        "eventTimeUnix": "1615988820",
        "id": "dc876134-87a4-4437-aabb-d2fc8a3d4213",
}
```

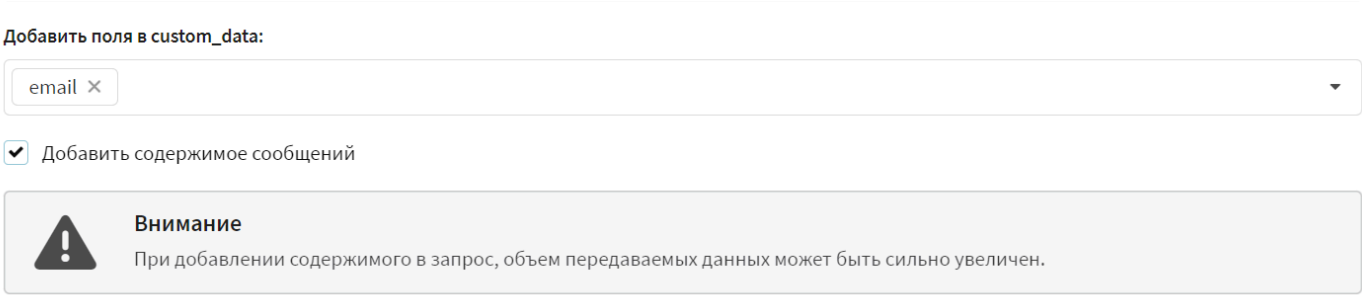
Поле `custom_data` может содержать любые произвольные данные. Передаваемые в нём поля не обязаны существовать в базе данных профилей и используются только в рамках самого события. В событии они отображаются в следующем виде:

```
"custom_data": {
    "test_field": "test"
}
```

Важно учитывать, что поля в `custom_data` не связаны напрямую со структурой базы профилей. При настройке захвата событий может возникнуть ощущение, что такие поля должны существовать в базе данных, поскольку они передаются в API-запросе. Однако это не требуется. Функция **“Добавить поля в custom_data”** используется для дополнительной передачи данных — например, чтобы добавить в `custom_data` значения из профиля или других источников, даже если эти поля изначально отсутствовали в самом запросе.

Контент сообщения в захвате

Для того чтобы вместе с событиями передавать контент всех сообщений, активируйте опцию **“Добавить содержимое сообщений”**:



Содержимое будет передаваться в поле `"content"`. Структура поля для каждого канала описана в этой статье.

CAUTION

Обратите внимание, что при этом все отправленные сообщения будут помещаться в хранилище на фиксированное время, что потребует дополнительного дискового пространства. По умолчанию время хранения — 72 часа и устанавливается глобальной настройкой `HOOK_CONTENT_CLEANUP_HOURS` в файле main.json. По истечению этого времени события будут передаваться уже без связанного с ними контента.

Экспорт исторических событий за период

Функция позволяет выгрузить в подключение данные о событиях, которые уже произошли в платформе. Это полезно для восстановления данных, если захват событий не был настроен вовремя или произошел сбой в работе.

NOTE

Для экспорта исторических событий недоступны типы событий: `campaign_launch`, `profile_merge`, `profile_split`

Настройка экспорта

Чтобы запустить экспорт исторических событий, активируйте и сохраните захват событий с нужными настройками, затем в блоке управления выберите опцию **“Захватывать исторические события за период”**, укажите даты начала и окончания периода и нажмите **“Запустить”:**

INFO: Обратите внимание

Период для экспорта исторических событий ограничен: минимальная дата — 01-01-1970, максимальная — предыдущий день. Текущая дата недоступна для выбора.

Для проверки работы экспорта используйте инструмент тестирования — система сгенерирует пример события выбранного типа.

Управление процессом

После запуска вы можете отслеживать прогресс и управлять выполнением:

- **Приостановить** — временно остановить выгрузку
- **Возобновить** — продолжить выгрузку
- **Прервать** — полностью отменить выгрузку

Статусы выполнения

- **Выполняется** — `Идет процесс экспорта исторических событий за период`
- **На паузе** — `Процесс экспорта исторических событий был приостановлен`
- **Завершено** — `Экспорт исторических событий за период выполнен`
- **Ошибка** — `Возникла ошибка при экспорте исторических событий`

> **NOTE**
>
> Если вы деактивируете захват событий во время выполнения экспорта исторических событий, процесс будет приостановлен до повторной активации.

Типы событий для захвата

TIP

При изменении статуса подписки (в интерфейсе, через API или при импорте) регистрируются события только для следующих статусов:

- Подписан
- Отписан
- Жалобщик
- Hardbounced

При этом, при изменении статуса на “Подписан”, событие регистрируется только если предыдущий статус подписки был “Отписан”, “Жалобщик” или “Hardbounced”.

Например, если выставить статус подписки “Приостановлен”, а затем изменить на статус “Подписан”, то событие **не зарегистрируется**.

По нижеуказанным статусам события **не регистрируются**:

- Не подтвержден
- Приостановлен
- Не валиден

```
<tr>
  <td colspan="3">
    <strong>События, связанные с [программами лояльности](/user-guide/loyalty-programs)
</strong>
  </td>
</tr>
<tr>
  <td>Программа лояльности: Регистрация</td>
  <td>
    <strong>loyalty_register</strong>
  </td>
  <td>Регистрация профиля в программе лояльности.</td>
</tr>
<tr>
  <td>Программа лояльности: Исключение</td>
  <td>
    <strong>loyalty_unregister</strong>
  </td>
  <td>Исключение профиля из программы лояльности.</td>
</tr>
<tr>
  <td>Программа лояльности: Повышение уровня</td>
  <td>
    <strong>level_up</strong>
  </td>
  <td>
    Повышение уровня лояльности.
  </td>
  </tr>
  <tr>
  <td>Программа лояльности: Понижение уровня</td>
  <td>
    <strong>level_down</strong>
  </td>
  <td>
    Понижение уровня лояльности.
  </td>
  </tr>
  <tr>
  <td>Программа лояльности: Начисление</td>
  <td>
    <strong>points_accrual</strong>
  </td>
  <td>
    Начисление баллов лояльности.
  </td>
  </tr>
  <tr>
  <td>Программа лояльности: Активация</td>
```

```
<td>Программа лояльности: Активация</td>
<td>
  <strong>points_activate</strong>
</td>
<td>
  Активация баллов лояльности.
</td>
</tr>
<tr>
<td>Программа лояльности: Сгорание</td>
<td>
  <strong>points_expire</strong>
</td>
<td>
  Сгорание баллов лояльности.
</td>
</tr>
<tr>
<td>Программа лояльности: Списание</td>
<td>
  <strong>points_redeem</strong>
</td>
<td>
  Списание баллов лояльности.
</td>
</tr>
<tr>
<td>Программа лояльности: Отмена начисления</td>
<td>
  <strong>points_cancel_accrual</strong>
</td>
<td>
  Отмена начисления баллов лояльности.
</td>
</tr>
<tr>
<td>Программа лояльности: Отмена  списания</td>
<td>
  <strong>points_cancel_redeem</strong>
</td>
<td>
  Отмена списания баллов лояльности.
</td>
</tr>
<tr>
<td>Программа лояльности: Сброс баллов</td>
<td>
  <strong>points_q_reset</strong>
</td>
<td>
```

```
    Сброс баллов лояльности.
  </td>
</tr>

<tr>
  <td colspan="3">
    <strong>События, связанные с <a href="/user-guide/profiles-and-databases/37847201">
      автоматическим созданием статического сегмента на импорте</a></strong>
  </td>
</tr>
<tr>
  <td>Static Segment Added</td>
  <td>
    <strong> segs_add</strong>
  </td>
  <td>Профиль вошел в статический сегмент.</td>
</tr>
<tr>
  <td>Static Segment Removed</td>
  <td>
    <strong> segs_remove</strong>
  </td>
  <td>Профиль вышел из статического сегмента.</td>
</tr>
<tr>
  <td colspan="3">
    <strong>

      События <a href="/developer-guide/profiles/41484440">объединения</a> нескольких
профилей или
      <a href="/developer-guide/profiles/41484445">разделения</a> одного профиля
    </strong>
  </td>
</tr>
<tr>
  <td>Profile merge</td>
  <td>
    <strong> profile_merge</strong>
  </td>
  <td>
    Несколько профилей объединены в новый или уже существующий профиль через запрос к API.
Недоступно для экспорта [исторических событий.](/user-guide/integrations/action-
hooks/37849833#export)
  </td>
</tr>
<tr>
  <td>Profile split</td>
  <td>
    <strong> profile_split</strong>
```

```
  </td>
  <td>Профиль разделен на несколько новых или уже существующих профилей через запрос к API. Недоступно для экспорта [исторических событий.](/user-guide/integrations/action-hooks/37849833#export)</td>
</tr>
<tr>
  <td colspan="3">
    <strong> События Telegram Bot канала</strong>
  </td>
</tr>
<tr>
  <td>Telegram bot отправки</td>
  <td>
    <strong> telegram_bot_send</strong>
  </td>
  <td>Сообщение отправлено для доставки подписчику в чат с Telegram ботом.</td>
</tr>
<tr>
  <td>Telegram bot доставки</td>
  <td>
    <strong> telegram_bot_deliv</strong>
  </td>
  <td>Сообщение доставлено в чат с Telegram ботом.</td>
</tr>
<tr>
  <td>Telegram bot клики</td>
  <td>
    <strong> telegram_bot_click</strong>
  </td>
  <td>Неуникальный клик по ссылке в сообщении в чате с Telegram ботом.</td>
</tr>
<tr>
  <td>Telegram bot уник. клики</td>
  <td>
    <strong> telegram_bot_uclick</strong>
  </td>
  <td>Уникальный клик по ссылке в сообщении.</td>
</tr>
<tr>
  <td>Telegram bot недоставки</td>
  <td>
    <strong> telegram_bot_undeliv</strong>
  </td>
  <td>Сообщение не доставлено в чат с Telegram ботом.</td>
</tr>
<tr>
  <td>Telegram bot подписка</td>
  <td>
    <strong> telegram_bot_subscribe</strong>
  </td>
```

```
  </td>
  <td>
    Профиль подписан на Telegram бота.
  </td>
</tr>
<tr>
  <td colspan="3">
    <strong> События Telegram Group канала</strong>
  </td>
</tr>
<tr>
  <td>Telegram group отправки</td>
  <td>
    <strong> telegram_group_send</strong>
  </td>
  <td>Сообщение отправлено для доставки подписчику в Telegram группу.</td>
</tr>
<tr>
  <td>Telegram group доставки</td>
  <td>
    <strong> telegram_group_deliv</strong>
  </td>
  <td>Сообщение доставлено в Telegram группу.</td>
</tr>
<tr>
  <td>Telegram group клики</td>
  <td>
    <strong> telegram_group_click</strong>
  </td>
  <td>Клик по ссылке в сообщении в Telegram группе.</td>
</tr>
<tr>
  <td>Telegram group недоставки</td>
  <td>
    <strong> telegram_group_undeliv</strong>
  </td>
  <td>Сообщение не доставлено в Telegram группу.</td>
</tr>
<tr>
  <td colspan="3">
    <strong> События WhatsApp*-канала</strong>
  </td>
</tr>
<tr>
  <td>WhatsApp* отправки</td>
  <td>
    <strong> whatsapp_send</strong>
  </td>
  <td>Сообщение отправлено для доставки подписчику в whatsapp чат.</td>
</tr>
```

```
<tr>
  <td>WhatsApp* доставки</td>
  <td>
    <strong> whatsapp_deliv</strong>
  </td>
  <td>Сообщение доставлено в whatsapp чат.</td>
</tr>
<tr>
  <td>WhatsApp* клики</td>
  <td>
    <strong> whatsapp_click</strong>
  </td>
  <td>Неуникальный клик по ссылке в сообщении в whatsapp чате.</td>
</tr>
<tr>
  <td>WhatsApp* уник. клики</td>
  <td>
    <strong> whatsapp_uclick</strong>
  </td>
  <td>Уникальный клик по ссылке в сообщении.</td>
</tr>
<tr>
  <td>WhatsApp* недоставки</td>
  <td>
    <strong> whatsapp_undeliv</strong>
  </td>
  <td>Сообщение не доставлено в whatsapp чат.</td>
</tr>
<tr>
  <td>
    WhatsApp* прочтения
  </td>
  <td>
    <strong> whatsapp_read</strong>
  </td>
  <td>
    Сообщение прочитано пользователем.
  </td>
</tr>
<tr>
  <td>WhatsApp* subscription</td>
  <td>
    <strong> whatsapp_subscribe</strong>
  </td>
  <td>
    Профиль подписался на WhatsApp*-канал.
  </td>
</tr>
<tr>
```

```
  <td>WhatsApp* unsubscribe</td>
  <td>
    <strong> whatsapp_unsub</strong>
  </td>
  <td>
    Профиль отписался от WhatsApp*-канала.
  </td>
</tr>
<tr>
  <td colspan="3">
    <strong> События Viber™ канала</strong>
  </td>
</tr>
<tr>
  <td>Viber™ отправки</td>
  <td>
    <strong> viber_send</strong>
  </td>
  <td>Сообщение отправлено для доставки подписчику в Viber™ чат.</td>
</tr>
<tr>
  <td>Viber™ доставки</td>
  <td>
    <strong> viber_deliv</strong>
  </td>
  <td>Сообщение доставлено получателю.</td>
</tr>
<tr>
  <td>Viber™ клики</td>
  <td>
    <strong> viber_click</strong>
  </td>
  <td>
    Переход по ссылке в сообщении. Если получатель несколько раз кликнул по одной и той же
    ссылке, фиксируется каждое событие клика.
  </td>
</tr>
<tr>
  <td>Viber™ уник. клики</td>
  <td>
    <strong> viber_uclick</strong>
  </td>
  <td>
    Уникальный переход по ссылке в сообщении. Если получатель хотя бы один раз перешел по
любой
    ссылке, фиксируется одно событие уникального клика.
  </td>
</tr>
<tr>
  <td>Viber™ недоставки</td>
```

```
  <td>Viber™ недоставки</td>
  <td>
    <strong> viber_undeliv</strong>
  </td>
  <td>Сообщение было отклонено Viber™ или Devino.Online.</td>
</tr>
<tr>
  <td>
    Viber™ прочтения
  </td>
  <td>
    <strong> viber_read</strong>
  </td>
  <td>
    Сообщение прочитано пользователем.
  </td>
</tr>
<tr>
  <td>Viber™ subscription</td>
  <td>
    <strong> viber_subscribe</strong>
  </td>
  <td>
    Профиль подписался на Viber™-канал, т.е. согласился на получение сообщений от бизнес
    аккаунта.
  </td>
</tr>
<tr>
  <td>Viber™ unsubscribe</td>
  <td>
    <strong> viber_unsub</strong>
  </td>
  <td>
    Профиль отписался от Viber™-канала, т.е. заблокировал получение сообщений от бизнес
аккаунта.
  </td>
</tr>
<tr>
  <td colspan="3">
    <strong> События Notify</strong>
  </td>
</tr>
   <tr>
  <td>Notify доставки</td>
  <td>
    <strong>notify_deliv</strong>
  </td>
  <td>Сообщение доставлено.</td>
</tr>
<tr>
```

```
  <td>Notify недоставки</td>
  <td>
    <strong>notify_undeliv</strong>
  </td>
  <td>Сообщение не доставлено.</td>
</tr>
<tr>
  <td>Notify отправки</td>
  <td>
    <strong>notify_send</strong>
  </td>
  <td>Сообщение отправлено.</td>
</tr>
<tr>
  <td>Notify клики</td>
  <td>
    <strong>notify_click</strong>
  </td>
  <td>
    Переход по ссылке в сообщении. Если получатель несколько раз кликнул по одной и той же
   ссылке, фиксируется каждое событие клика.
  </td>
</tr>
<tr>
  <td>Notify уник. клики</td>
  <td>
    <strong>notify_uclick</strong>
  </td>
  <td>Уникальный переход по ссылке в сообщении. Если получатель хотя бы один раз перешел по
любой
    ссылке, фиксируется одно событие уникального клика.</td>
</tr>
<tr>
  <td>Notify подписки</td>
  <td>
    <strong>notify_subscribe</strong>
  </td>
  <td>Профиль подписался на Notify-канал.</td>
</tr>
<tr>
  <td>Notify прочтения</td>
  <td>
    <strong>notify_read</strong>
  </td>
  <td>Сообщение прочитано пользователем.</td>
</tr>
 <tr>
  <td>Notify отписки</td>
  <td>
```

```
    <strong>notify_unsub</strong>
  </td>
  <td>Профиль отписался от Notify-канала.</td>
</tr>
 <tr>
  <td>Notify жалобы</td>
  <td>
    <strong>notify_complain</strong>
  </td>
  <td>Подписчик пожаловался на сообщение.</td>
</tr>
<tr>
  <td colspan="3">
    <strong> События Попапа</strong>
  </td>
</tr>
<tr>
  <td>Popup: Post</td>
  <td>
    <strong>popup_post</strong>
  </td>
  <td>Форма в попапе заполнена и отправлена.</td>
</tr>
<tr>
  <td>Popup: Click</td>
  <td>
    <strong>popup_click</strong>
  </td>
  <td>Выполнен клик по любому элементу попапа, кроме кнопки "Закрыть".</td>
</tr>
<tr>
  <td>Popup: Error</td>
  <td>
    <strong>popup_error</strong>
  </td>
  <td>Любая ошибка в попапе.</td>
</tr>
<tr>
  <td>Popup: Bounce</td>
  <td>
    <strong>popup_bounce</strong>
  </td>
  <td>
    В процессе отправки формы попапа возникли технические проблемы (например, сбой на
сервере).
  </td>
</tr>
<tr>
  <td>Popup: Close</td>
```

```
  <td>
    <strong>popup_close</strong>
  </td>
  <td>Выполнен клик по кнопку "Закрыть" в попапе.</td>
</tr>
<tr>
  <td>Popup: Open</td>
  <td>
    <strong>popup_open</strong>
  </td>
  <td>Попап показан пользователю.</td>
</tr>
<tr>
  <td>Popup: Custom</td>
  <td>
    <strong>popup_custom</strong>
  </td>
  <td>Зарегистрировано событие, настроенное на стороне клиента.</td>
</tr>
<tr>
  <td colspan="3">
    <strong>События разметки аудитории в кампаниях</strong>
  </td>
</tr>
<tr>
  <td>Попал в аудиторию кампании</td>
  <td>
    <strong>sc_add</strong>
  </td>
  <td>Профиль добавлен в аудиторию кампании</td>
</tr>
<tr>
  <td>Вошел в тестовую группу</td>
  <td>
    <strong>sc_test_group_assign</strong>
  </td>
  <td>Профиль распределен в тестовую группу</td>
</tr>
<tr>
  <td>Вошел в контрольную группу</td>
  <td>
    <strong>sc_control_group_assign</strong>
  </td>
  <td>Профиль распределен в контрольную группу</td>
</tr>
<tr>
  <td>Вышел из аудитории кампании</td>
  <td>
    <strong>sc_remove</strong>
  </td>
```

```
  <td>Профиль исключен из аудитории кампании</td>
</tr>
<tr>
  <td>Исключен ГКГ из кампании</td>
  <td>
    <strong>sc_gcg_suppress</strong>
  </td>
  <td>Профиль не добавлен в аудиторию кампании из-за нахождения в ГКГ</td>
</tr>
```

В интерфейсе	В запросе	Описание
События Email канала		
Send	**send**	Email сообщение отправлено для доставки подписчику.
Reply	**reply**	Ответ подписчика на полученное Email сообщение.
Unique Open	**uopen**	Уникальное открытие email сообщения.
Open	**open**	Неуникальное открытие email сообщения.
Unique Click	**uclick**	Уникальный клик по ссылке в email сообщении. Создаёт также события **open** и **uopen*, если событие **open** не было получено.
Click	**click**	Неуникальный клик по ссылке в email сообщении.
Confirm	**confirm**	Клик по ссылке подтверждения в обычном сообщении. Клик по confirm ссылке создаёт также событие **click** и **uclick**, если это первый клик в сообщениию.
Confirm Subscription	**confirmsub**	Клик по ссылке подтверждения подписки в double opt-in сообщении.
Unsubscribe	**unsub**	Подписчик перешел по ссылке отписки в письме и был отписан. `{unsubscribe}` или `{globalunsubscribe}`
Complain	**complain**	Подписчик пожаловался на сообщение.
SoftBounce	**sbounce**	Событие ответа о временной невозможности доставки сообщения.
HardBounce	**hbounce**	Ответ о невозможности доставки сообщения.
Suppression	**suppress**	Сообщение не было отправлено подписчику по причине нахождения в стоп-листе.
Delivered	**deliv**	Отправщик доставил email сообщение подписчику.

В интерфейсе	В запросе	Описание
Undelivered	**undeliv**	Отправщик не смог доставить сообщение подписчику.
Read Message	**read**	Сообщение открыто пользователем в течении **8 и более секунд.**
Glanced Message	**glanced**	Сообщение открыто **до 4 секунд.**
Skimmed Message	**skimmed**	Сообщение открыто **до 8 секунд.**
События импорта профилей подписчиков		
Manual Import	**import_manual**	Подписчик был добавлен через интерфейс системы.
File Import	**import_file**	Подписчик был добавлен через интерфейс системы, из файла.
API Import	**import_api**	Подписчик был добавлен через API.
События для SMS канала		
Send SMS	**send_sms**	SMS сообщение отправлено для доставки подписчику.
Click SMS	**click_sms**	Не уникальный клик по ссылке в SMS сообщении.
Unique Click SMS	**uclick_sms**	Уникальный клик по ссылке в SMS сообщении.
Delivered SMS	**deliv_sms**	SMS сообщение доставлено.
Undelivered SMS	**undeliv_sms**	SMS сообщение не доставлено.
Unsubscribe SMS	**unsub_sms**	Подписчик перешел по ссылке отписки в сообщении и был отписан. `{unsubscribe}` или `{globalunsubscribe}`
События для Push канала		
Unique Open Push	**uopen_push**	Уникальное открытие push сообщения.
Open Push	**open_push**	Не уникальное открытие push сообщения.

В интерфейсе	В запросе	Описание
Send Push	**send_push**	Push сообщение отправлено для доставки подписчику.
Click Push	**click_push**	Не уникальный клик по ссылке в push сообщении.
Unique Click Push	**uclick_push**	Уникальный клик по ссылке в push сообщении.
Delivered Push	**deliv_push**	Push сообщение доставлено.
Undelivered Push	**undeliv_push**	Push сообщение не доставлено.
Softbounce Push	**sbounce_push**	Ответ о временной невозможности доставки сообщения (сетевые проблемы, недоступность push-сервиса по любым причинам). Это событие означает, что платформа продолжит попытки доставить сообщение.
Hardbounce Push	**hbounce_push**	Ответ о невозможности доставки push сообщения (недействительный идентификатор подписки). После получения такого ответа платформа не будет пытаться доставить сообщение повторно.
Кастомная (API) отписка от рассылок		
API unsubscribe	**unsub_api**	Подписчик отписан от рассылок через API.
Запуск рассылки		
Campaign launch	**campaign_launch**	Запущена рассылка (имеет особую структуру hook-сообщения).
Ошибки рассылки		
General: Error	**camp_error**	Ошибки, возникшие в процессе работы рассылки.
События, захваченные Пикселями на внешних ресурсах		
Pixel Open	**pixel_open**	Открыт AK-Pixel на сайте. Узнать подробнее о параметрах данного запроса можно [здесь] (/developer-guide/goals/2195843).
События, связанные с промокодами		
Promocode Attach	**promocode_attach**	Подписчику назначен промокод.
Promocode Detach	**promocode_detach**	Промокод отвязан от профиля подписчика.

В интерфейсе	В запросе	Описание
Promocode Activate	**promocode_activate**	Промокод активирован.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Структуры сообщений захвата событий

Общая структура hook-сообщения для всех типов событий:

Параметр	Тип	Пример	Описание
event_type	string	“click”	Название события как в таблице типы событий
event_date	string	“2018-03-04T13:13:19.034315796Z”	Дата возникновения события в формате RFC 3339
profile_id	string	“abcdefgh12345678”	ID профиля получателя
_xxh	string	“ad1850e0d1d9481e”	Хеш от email подписчика, используемый для идентификации в Altcraft
_md5	string	“145bb30f7ddf1184cb560bc4b7705646”	MD5 хеш от email подписчика
list_id	int	111	ID Базы данных, в рамках которого возникло событие
account_id	int	23	ID Аккаунта, в рамках которого возникло событие
sends_trying_num	int	2	Число попыток отправки сообщения (присутствует, только если количество попыток больше 0)
is_test	bool	false	Было ли событие вызвано тестовой отправкой сообщения
is_static	bool	false	Использовались ли динамические подстановки в отправляемый контент
subscription_hash_id	string	“0a4b910312345”	хеш подписки
custom_data	JSON object	{“alice@example.org”}	Набор данных о подписчике, настраиваемый в интерфейсе редактирования Action Hook
event_data	JSON object	{“browser”: “Firefox”}	Набор данных, зависящий от типа события (есть два типа — событие в трекинге и возникающее в процессе доставки)

Параметр	Тип	Пример	Описание
subscription	array	[{ “resource_id”: 123, “channel”: “email”, “email”: “user@example.com”, “phone”: “+79991234567”, “subscription_id”: “abc123”, “provider”: “androidfirebase”, “priority”: 1, “status”: “subscribed”, “ccsub”: { “custom_key”: “custom_value” } }]	Массив с деталями подписки, включающий информацию о ресурсе, канале связи, получателе и статусе подписки. Структура описана ниже.

INFO: Обратите внимание

При экспорте исторических событий контент сообщений не передается, события отправляются тем же форматом, что и в реальном времени, а для удаленных профилей передается `null`.

Структура event_data

События системы трекинга

Для событий следующих типов:

- **uopen, open, uclick, click, confirm, confirmsub, unsub, read, glanced, skimmed,**
- **uclick_sms, click_sms, uclick_sms, unsub_sms,**
- **uopen_push, open_push, click_push, uclick_push.**

Параметр	Тип	Пример	Описание
message_id	int	17	ID сообщения в системе Altcraft

Параметр	Тип	Пример	Описание
campaign_id	int	973	ID рассылки в системе Altcraft
scampaign_id	int	973	ID кампании в системе Altcraft
segment_id	int	0	ID сегмента, если доступно
resource_id	int	3	ID ресурса
ip	string	“10.9.0.14”	IPv4 адрес подписчика
country	string	“.us” “.ru”	Страна в TLD виде
city	string	“New York” “Moscow”	Город подписчика
postal_code	string	“101000”	Почтовый код подписчика
region	string	“Ohio” “Moskovskaya oblast”	Регион подписчика
tz	string	“Europe/Vaduz”	Временная зона подписчика
lat	float	55.244075	Географическая широта
lon	float	55.244075	Географическая долгота
user_agent	string	“Mozilla/5.0 (X11; Linux x86_64; rv:59.0) Gecko/20100101 Firefox/59.0”	Строка в заголовке User-Agent
os	string	“Linux x86_64”	Операционная система
browser	string	“Firefox”	Браузер, версия
device	string	“web”	Устройство
link_url	string	“http://www.example.com”	HTTP ссылка, на которую был осуществлен редирект
launch_id	string	“00000017000003cd5a9bf01a”	ID запуска
send_message_id	string	“5a9bf01a17_3cd_11_6f_0_2.5a8c045fc1dde44cbf718f78”	ID отправленного сообщения

Параметр	Тип	Пример	Описание
full_message_id	string	“fQIQs4FgAAEBIcARAxDxAQAAEJECFQADuZJygDTgWXpQ_BVrkejVzu2bsMzQ3NDkxODlKR8-gA”	Полный SendMessageID
email_message_id	string	“mzi5c1~+9vqekqmy63mh8xhgt2qjwfkad6lmiafbyn-0te1_uzckfqyi4ry”	Email-safe SendMessageID (только для событий email-канала)
extra_uid	string	“ew4Sgf41Qwjq8_2_2E_w_k___t.2SDvJv8bvBYQFtSDx_7bTV2f.2_2E_4Sgf41DXc7Z_D”	Extra UID (только для событий push-канала)
provider	string	“YandexAppMetricaDeviceIDAndroid”	Провайдер
prev_st	int	1	Идентификатор группы: 1 - тестовая, 2 - ЛКГ

События отправки в очередь

Для событий следующих типов:

- **send, deliv, suppress, complain,**
- **send_sms, deliv_sms,**
- **send_push, deliv_push.**

Параметр	Тип	Пример	Описание
message_id	int	17	ID сообщения в системе Altcraft
campaign_id	int	973	ID рассылки в системе Altcraft
segment_id	int	0	ID сегмента, если доступно
resource_id	int	3	ID ресурса в системе Altcraft
launch_id	string	“00000017000003cd5a9bf01a”	ID запуска
send_message_id	string	“5a9bf01a17_3cd_11_6f_0_2.5a8c045fc1dde44cbf718f78”	ID отправленного сообщения
full_message_id	string	“fQIQs4FgAAEBIcARAxDxAQAAEJECFQADuZJygDTgWXpQ_BVrkejVzu2bsMzQ3NDkxODlKR8-gA”	Полный SendMessageID
email_message_id	string	“mzi5c1~+9vqekqmy63mh8xhgt2qjwfkad6lmiafbyn-0te1_uzckfqyi4ry”	Email-safe SendMessageID (только для событий email-канала)
extra_uid	string	“ew4Sgf41Qwjq8_2_2E_w_k___t.2SDvJv8bvBYQFtSDx_7bTV2f.2_2E_4Sgf41DXc7Z_D”	Extra UID (только для событий push-канала)
provider	string	“YandexAppMetricaDeviceIDAndroid”	Провайдер

Для событий **sbounce, hbounce, sbounce_push, hbounce_push** также передаются следующие параметры:

Параметр	Тип	Пример	Описание
bounce_code	int	500	Код ответа от сервера
bounce_message	string	“No such email”	Ответ от сервера

Для событий **undeliv, undeliv_sms, undeliv_push** передаются следующие параметры:

Параметр	Тип	Пример	Описание
undeliv_code	int	400	Код ошибки
undeliv_message	string	“Invalid Caller ID”	Сообщение об ошибке

События пикселей

Для событий типа **pixel_open**:

Параметр	Тип	Пример	Описание
ip	string	“10.9.0.14”	IPv4 адрес подписчика
country	string	".ru "	Страна в TLD виде
city	string	“Moscow”	Город подписчика
pixel_id	int	41	ID пикселя
postal_code	string	“390000”	Почтовый код подписчика
region	string	"Ryazanskaya obl"ast	Регион подписчика
tz	string	“Europe/Vaduz”	Временная зона подписчика
lat	float	55.244075	Географическая широта
lon	float	55.244075	Географическая долгота
user_agent	string	“Mozilla/5.0 (X11; Linux x86_64; rv:59.0) Gecko/20100101 Firefox/59.0”	Строка в заголовке User-Agent
os	string	“Linux x86_64”	Операционная система
browser	string	“Firefox”	Браузер, версия
device	string	“web”	Устройство

Полный список возможных параметров указан в этой статье.

События попапов

Для всех событий типа **popup_post, popup_click, popup_error, popup_bounce, popup_close, popup_open**:

Параметр	Тип	Пример	Описание
popup_id	int	123	ID попапа
referer	string	“api.example.com”	Заголовок запроса Referer
smid	string	“fGmXiWU9Y4NSIpCwl4QQQQEQEAABCRA”	Идентификатор отправляемого сообщения
utm_campaign	string	“utm_campaign”	UTM-метка рассылки
utm_content	string	“utm_content”	UTM-метка контента
utm_source	string	“utm_source”	UTM-метка источника

У некоторых событий есть свои уникальные параметры:

Параметр	**Тип**	**Пример**	**Описание**
popup_post			
post_uid	string	"7bc142b5-7f38-4272-bc7e-7539f70b25b1"	Уникальный идентификатор заполнения формы
popup_click			
name	string	"main_block"	Название элемента, по которому был совершен клик
url	string	"example.com/page"	Значение атрибута `href` элемента, по которому был совершен клик
popup_error, popup_bounce			
error	string	"object_not_found"	Текст ошибки
error_category	int	404	Номер ошибки

События промокодов

Для событий типа **promocode_attach, promocode_detach, promocode_activate**:

Параметр	Тип	Пример	Описание
attached	string	“2020-10-25T13:28:52Z”	Дата и время прикрепления промокода к профилю
detached	string	“2020-10-25T13:28:52Z”	Дата и время отсоединения промокода
activated	string	“2020-10-25T13:28:52Z”	Дата активации промокода
code	string	“TEST-XOUH-KBJM-J5K9-24I1”	Промокод

Параметр	Тип	Пример	Описание
db_id	int	123456	Идентификатор базы профиля
loyalty_id	int	9	Идентификатор программы лояльности
profile_id	string	“000000000000000000000000”	Идентификатор профиля

Все даты указываются в формате **RFC 3339**.

События создания сегмента при импорте

Для событий типа **segs_add, segs_remove**:

Параметр	Тип	Пример	Описание
campaign_id	int	9	ID рассылки в системе Altcraft
launch_id	string	“”	ID запуска рассылки
message_id	int	0	ID сообщения в системе Altcraft
segment_id	int	80	ID сегмента, если доступно
send_message_id	string	“”	ID отправленного сообщения
full_message_id	string	“fQIQs4FgAAEBIcARAxDxAQAAEJECFQADuZJygDTgWXpQ_BVrkejVzu2bsMzQ3NDkxODlKR8-gA”	Полный SendMessageID

События объединения нескольких профилей или разделения одного профиля

Для событий типа **profile_merge**, **profile_split**:

Параметр	Тип	Пример	Описание
destination	array	[{ “custom_data”: { “age”: 25 }, “db_id”: 94, “profile_id”: “608a8960e3224a6be26a71df” }]	Если происходит объединение профилей, то содержит данные матчинга профиля, который получится в результате объединения. Если происходит разделение профиля, то содержит данные матчинга профилей, которые получатся в результате разделения.

Параметр	Тип	Пример	Описание

source	array	[
		{
		“custom_data”: {
		“age”: 25
		},
		“db_id”: 94,
		“profile_id”: “608a8960e3224a6be26a71df”
		}
		]
			При объединении профилей содержит данные матчинга о профилях, которые нужно объединить.
			
			При разделении профиля содержит данные матчинга о профиле, который нужно разделить.

События каналов

Для событий следующих каналов:

- **Telegram**: `telegram_bot_send, telegram_bot_deliv, telegram_bot_click, telegram_bot_uclick, telegram_bot_undeliv, telegram_group_send, telegram_group_deliv, telegram_group_click, telegram_group_undeliv`
- **WhatsApp***: `whatsapp_deliv, whatsapp_undeliv, whatsapp_send, whatsapp_click, whatsapp_uclick, whatsapp_subscribe, whatsapp_read, whatsapp_unsub`
- **Viber***: `viber_deliv, viber_undeliv, viber_send, viber_click, viber_uclick, viber_subscribe, viber_read, viber_unsub, viber_complain`
- **Notify**: `notify_deliv, notify_undeliv, notify_send, notify_click, notify_uclick, notify_subscribe, notify_read, notify_unsub, notify_complain`

Параметр	Тип	Пример	Описание
campaign_id	int	9	ID рассылки в системе Altcraft
launch_id	string	“”	ID запуска рассылки
message_id	int	0	ID сообщения в системе Altcraft
resource_id	int	3	ID ресурса в системе Altcraft
segment_id	int	80	ID сегмента, если доступно
send_message_id	string	“”	ID отправленного сообщения
full_message_id	string	“fQIQs4FgAAEBIcARAxDxAQAAEJECFQADuZJygDTgWXpQ_BVrkejVzu2bsMzQ3NDkxODIKR8-gA”	Полный SendMessageID
city	string	“Moscow”	Город подписчика
browser	string	“Firefox”	Браузер, версия

Параметр	Тип	Пример	Описание
country	string	".ru "	Страна в TLD виде
device	string	“web”	Устройство
ip	string	“10.9.0.14”	IPv4 адрес подписчика
lat	float	55.244075	Географическая широта
link_url	string	“http://www.example.com”	HTTP ссылка, на которую был осуществлен редирект
lon	float	55.244075	Географическая долгота
os	string	“Linux x86_64”	Операционная система
postal_code	string	“390000”	Почтовый код подписчика
region	string	"Ryazanskaya obl"ast	Регион подписчика
tz	string	“Europe/Vaduz”	Временная зона подписчика
undeliv_code	int	500	Код ошибки недоставки
undeliv_message	string	“Content is required”	Информация об ошибке

События ошибок рассылки

Для событий типа **camp_error**:

Параметр	Тип	Пример	Описание
campaign_id	int	9	ID рассылки в системе Altcraft
error_code	int	500	Код ошибки
error_text	string	“LOG_INTERNAL_ERROR”	Текст ошибки
launch_id	string	“”	ID запуска

Параметр	Тип	Пример	Описание
message_id	int	0	ID сообщения в системе Altcraft
resource_id	int	3	ID ресурса в системе Altcraft
segment_id	int	80	ID сегмента, если доступно
send_message_id	string	“”	ID отправленного сообщения
full_message_id	string	“fQIQs4FgAAEBIcARAxDxAQAAEJECFQADuZJygDTgWXpQ_BVrkejVzu2bsMzQ3NDkxODlKR8-gA”	Полный SendMessageID

Структура campaign_launch

Общая структура campaign_launch:

Параметр	Тип	Пример	Описание
event_type	string	“campaign_launch”	Тип события
event_date	string	“2018-03-25T13:13:19.034315796Z”	Дата события в формате RFC 3339
account_id	int	23	ID аккаунта платформы
is_test	bool	false	Признак тестовой отправки события
event_data	JSON object	{}	Объект с данными о рассылке

Структура event_data для campaign_launch

Параметр	Тип	Пример	Описание
campaign_id	int	42	Идентификатор рассылки
campaign_type	string	“broadcast”	Тип рассылки
campaign_name	string	“name”	Название рассылки
campaign_sub_id	string	“x11”	Идентификатор статистики
data_type	string	“segment”	Тип источника данных
data_id	int	3	Идентификатор источника данных
launch_id	string	“00000017000003cd5a9bf01a”	ID запуска
content	JSON object	{}	Информация о шаблонах сообщений рассылки

Пример события запуска рассылки

```
{
  "event_type": "campaign_launch",
  "event_date": "2016-12-09T10:11:24.934079204Z",
  "is_test": false,
  "account_id": 161,
  "event_data": {
    "campaign_id": 1472,
    "campaign_type": "broadcast",
    "campaign_name": "Campaugn Launch Hook",
    "campaign_sub_id": "x11",
    "data_type": "segment",
    "data_id": 1221,
    "launch_id": "000000a1000005c0584a834c",
    "content": [
      {
        "message_name": "Hello",
        "message_id": 1412,
        "suppress_id": 271,
        "replyto_email": "",
        "replyto_name": "",
        "text": "текстовая версия письма",
        "html": "<html> html версия письма </html>",
        "from_name": "",
        "from_email": "",
        "attach_links": [],
        "subject": "Hello, {your_name}!"
      },
      {
        "message_name": "Hi",
        "message_id": 1413,
        "suppress_id": 272,
        "replyto_email": "monty@altcraft.com",
        "replyto_name": "Monty",
        "text": "текстовая версия письма",
        "html": "<html> html версия письма </html>",
        "from_name": "",
        "from_email": "",
        "attach_links": [
          {
            "name": "text_file.txt",
            "url": "http://test.altcraft.com/api/attach/link/text_file.txt?token=eyJh..."
          },
          {
            "name": "picture_file.png",
            "url": "http://test.altcraft.com/api/attach/link/picture_file.png?
token=eyJh..."
          }
        ],
        "subject": "Hi, {your_name}!"
```

```
          subject : "Hi, {your_name}!
        }
      ]
    }
}
```

Структура subscription

В этом объекте передается информация о подписке на ресурс, через который произведена отправка:

Параметр	**Тип**	**Пример**	**Описание**
Общие для всех каналов			
resource_id	int	12	ID ресурса
channel	string	"email"	Канал коммуникации
priority	int	0	Приоритет подписки
status	string	"subscribed"	Статус подписки
Для email-канала			
email	string	"john@example.com"	Email подписчика
Для SMS-канала			
post_uid	string	"+79991112233"	Телефон подписчика
Для push-канала			
provider	string	"YandexAppMetricaDeviceIDAndroid"	Push-провайдер
subscription_id	string	"123ab667960efec47b1afe64b56bfa386a3"	ID подписки провайдера

Структура content

Если в настройках вебхука активирована опция “Добавить содержимое сообщений”, то в общую структуру hook-сообщения добавляется поле `content`.

TIP

Ссылки на веб-версии всегда попадают с хуком, вне зависимости от того, активирована ли опция или нет. Единственное исключение — campaign_launch, для него веб-версий нет.

Email

```
{
    "content": {
        "amp": "<html amp4email></html>",
        "from_email": "no-reply@altcraft.com",
        "from_name": "Altcraft",
        "html": "<html></html>",
        "replyto_email": "example@example.com",
        "replyto_name": "Altcraft",
        "subject": "Weather alert",
        "text": "Text version",
        "to_email": "profile@example.com",
        "to_name": "Fname Lname",
        "webversion_email_html": "http://click.altkraft.com/0000/path.html",
        "webversion_email_text": "http://click.altkraft.com/0000/path.txt"
    },
}
```

Параметр	Тип	Пример	Описание
amp	string	<html amp4email></html>	AMP-версия письма
from_email	string	“no-reply@altcraft.com”	Email отправителя
from_name	string	“Altcraft”	Имя отправителя
html	string	<html></html>	HTML-версия письма
replyto_email	string	“example@example.com”	Email для обратной связи
replyto_name	string	“Altcraft”	Имя обратного получателя
subject	string	“Weather alert”	Тема письма
text	string	“Text version is important…”	Текстовая верия письма
to_email	string	“profile@example.com”	Email адрес профиля получателя
to_name	string	"Fname Lname "	Имя профиля получателя
webversion_email_html	string	http://click.altkraft.com	Ссылка на web версию письма
webversion_email_text	string	http://click.altkraft.com	Ссылка на текстовую веб-версию письма

SMS

```
{
    "content": {
        "fromnamesms": "abcdefghijklmnqrstuvwxyz",
        "sms": "Hello, Aaron! See the new items for you: http://click.altkraft.com/s/path",
        "webversion_sms_text": "http://click.altkraft.com/000000/path.txt"
    }
}
```

Параметр	Тип	Пример	Описание
fromnamesms	string	“abcdefghijklmnqrstuvwxyz”	Caller ID
sms	string	“Переходи по ссылке и забери приветственные бонусы!”	Текст сообщения
webversion_sms_text	string	http://click.altkraft.com	Ссылка на веб-версию сообщения

Push

```
{
    "content": {
        "app_push_settings": {
            "background_color": "#00CED1",
            "badges": 0,
            "banner": "",
            "critical_push": false,
            "deliver_prior": 0,
            "expir_time": 604800,
            "led": "",
            "media": "",
            "silent_push": false,
            "sound": "",
            "subtitle": "",
            "vibro": false
        },
        "body": "See if our new items are fit to make you even more awesome!",
        "click_url": "https://click.altkraft.com/0000000",
        "custom_push_json": {},
        "icon": "",
        "result_content": {
            "content_available": true,
            "data": {
                "hub_link": {
                    "ack": "https://pxl.altkraft.com/crypto_post_events?/bFOoNcxW7lpDA",
                    "open": "https://pxl.altkraft.com/crypto_post_events?/LbFOoNcxW7lpDA"
                },
                "image": "",
                "launch_id": "2_3J_4Z2ZTRaQr7p",
                "mediaUrl": "",
                "message": "See if our new items are fit to make you even more awesome!",
                "uid": "fATFSGbgLjinkRE59ZXDOSd0BcAABARnhIxDBBAAAEbAQowRH"
            },
            "mutable_content": true,
            "notification": {
                "body": "See if our new items are fit to make you even more awesome!",
                "click_action":
"https://click.altkraft.com/0000000/abcdefghijklmnqrstuvwxyz",
                "color": "",
                "icon": "",
                "sound": "default"
            },
            "title": "Just look, Alex! These are awesome!",
            "priority": "normal",
            "time_to_live": 604800,
            "to": "5dbDUxVXHV"
        },
        "title": "Just look, Alex! These are awesome!",
        "webversion_push_body": "http://click.altkraft.com/000000/path.txt"
```

```
        webversion_push_body : "http://click.altkraft.com/000000/path.txt"
    }
}
```

Параметр	Тип	Описание
body	string	Тело push сообщения
click_url	string	Ссылка для перехода по клику
custom_push_json	JSON object	Кастомный JSON сообщения
icon	string	Путь до иконки push-сообщения
title	string	Заголовок сообщения
result_content	JSON object	Хранит структуру данных, отправленных в push-сервис
webversion_push_body	string	Ссылка на веб-версию push-сообщения
app_push_settings	JSON object	Расширенные настройки для мобильных пушей
web_push_settings	JSON object	Расширенные настройки для браузерных пушей

Структура app_push_settings

Параметр	Тип	Пример	Описание
background_color	string	#00CED1	Цвет фона push-уведомления
badges	int	0	Индикатор событий
banner	string	http://example.com/path.img"	Путь к изображению-баннеру
critical_push	bool	false	Активация критического режима в push-уведомлении
deliver_prior	int	0,1	Приоритет доставки
expir_time	int	604800	TTL push-уведомления (задается в секундах)
led	string	#00CED2	Цвет светодиодного индикатора
media	string	http://example.com/path	Путь к локальному файлу в приложении
silent_push	bool	false	Включение/отключение тихого режима
sound	string	/sound/sound.mp3	Путь к локальному файлу в приложении

Параметр	Тип	Пример	Описание
subtitle	string	“Только для вас!”	Подзаголовок сообщения (для iOS)
vibro	bool	true	Вибрация при поступлении push-уведомления
soundless	bool	true	Указывает, будет ли push-сообщение сопровождаться звуком (используется провайдерами: android firebase, android huawei, android rustore)
actions	array	`[ { “label”: “Открыть”, “action”: “https://example.com” } ]`	Кнопки в push-уведомлениях

Структура web_push_settings

Параметр	Тип	Пример	Описание
actions	JSON object	`[]`	Массив содержит кнопки пуша: ссылка для перехода (action) + текст кнопки (label)
button_label	string	“Конечно”	Текст кнопки действия (значение передается только для провайдера Safari)
large_image	string	“https://.altkraft.com/284834790/1/664c6da207767c769a/image.png”	Путь к изображению-баннеру
expir_time	string	604800	TTL push-уведомленния (задается в секундах)
large_image	string	“https://click.example.com”	Путь к локальному изображению баннера

INFO

Ссылка на веб-версию сообщения передается в объекте `content` только для события **send** в каналах email, sms и push. Для остальных событий, а также для события **send** в кастомном канале объект `content` содержит поле `webversion_link` с пустой строкой:

```
"content": {
        "webversion_link": ""
}
```

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

***Viber является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

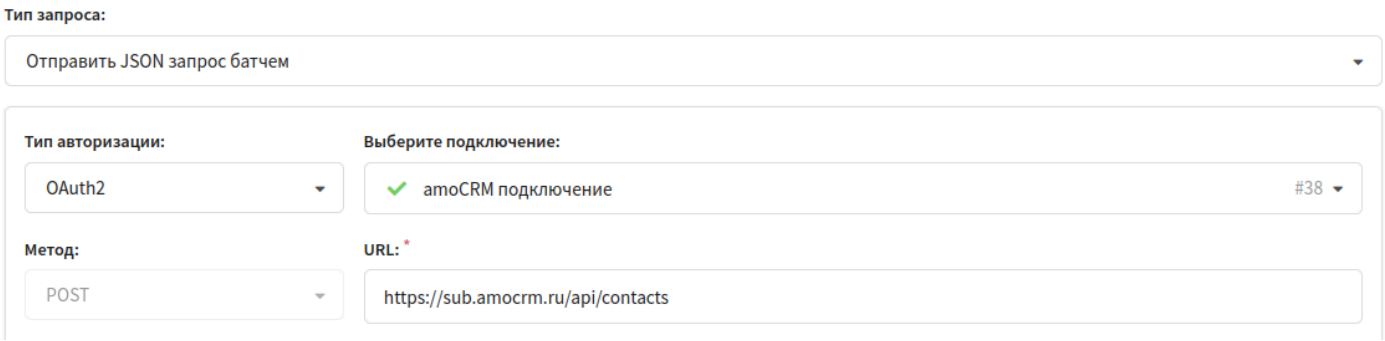
Отправить JSON-запрос батчем

Batch-запросы позволяют объединить множество HTTP-запросов в один, указав в теле каждый запрос как отдельный объект. Сервер возвращает один HTTP-ответ, внутри которого находятся ответы на каждый переданный запрос.

Когда мы выбираем этот тип запроса, мы отправляем полученные события в формате JSON в теле POST-запроса к внешнему API. При отправке будет сгенерирован JSON-массив с событиями.

Настройка запроса

Стандартный запрос отправляется без авторизации. Однако если вы хотите передавать информацию во внешней сервис, который в качестве метода авторизации использует протокол OAuth 2.0, измените **тип авторизации** на OAuth2. В таком случае потребуется выбрать подключение, связывающее платформу Altcraft и сервис.



В настройках запроса необходимо указать:

- **URL** сервиса, принимающего данные.
- **Максимальный размер батча** — ограничение количества событий в JSON-массиве. POST-запрос будет отправлен, как только наберётся достаточно событий.

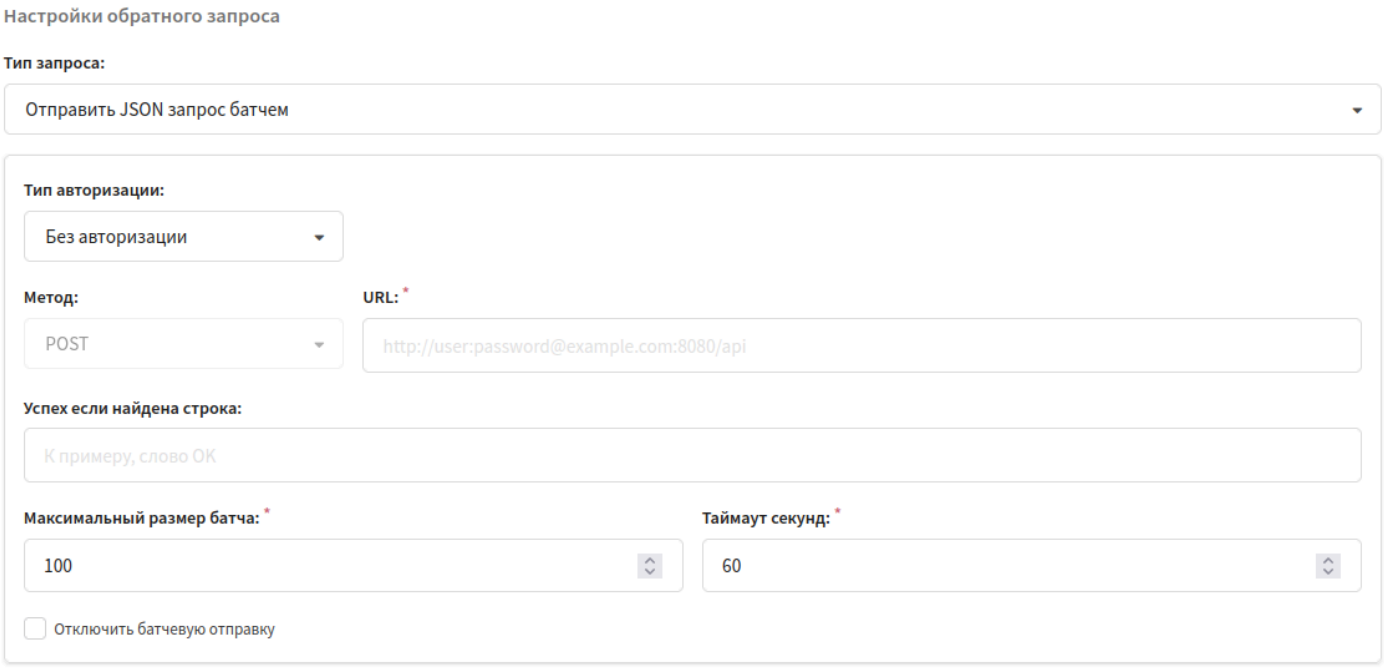
```
[{"event": "send", ... }, { }, ... ]
```

- **Таймаут секунд** — период времени, в течение которого накапливать события в батче для отправки на сервер. Если за это время максимальный размер батча так и не будет достигнут, POST-запрос всё равно будет отправлен. В таком случае JSON-массив будет содержать количество событий, которое накопилось к моменту истечения таймаута.
- **Успех если найдена строка** — ответ от сервера должен попасть в этот Regex, только тогда запрос на сервер будет признан успешным. Например:

```
.*OK.*
```

CAUTION

Если оставить поле “Успех если найдена строка” пустым, то любой ответ со статусом `(status code) 200` будет считаться правильным.



Пример запроса для двух событий доставки

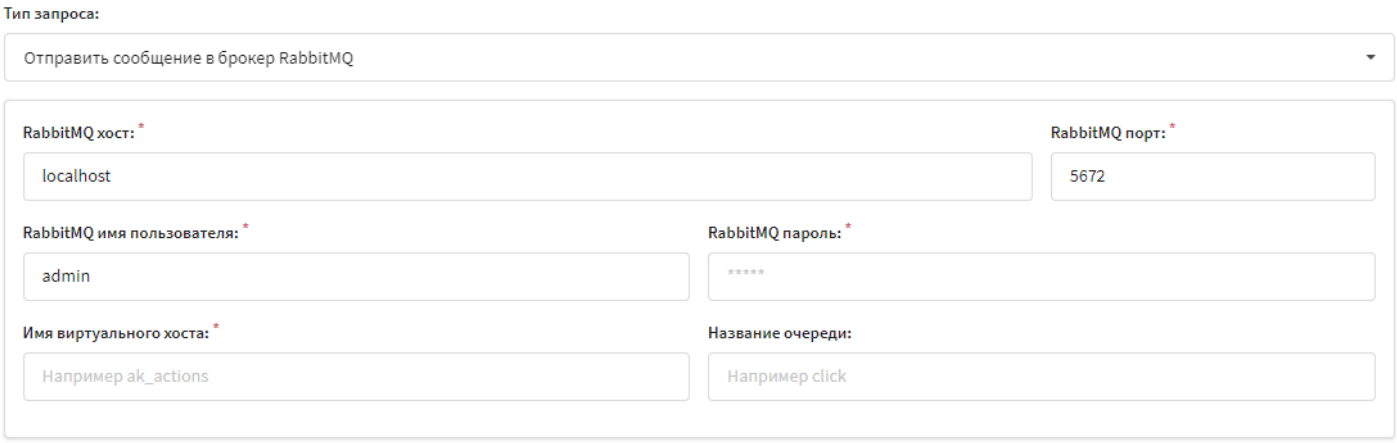
```
[
  {
    "event_type": "deliv",
    "list_id": 50,
    "account_id": 161,
    "_xxh": "5ada5e6ea3c6da5f",
    "_md5": "a5656dc1e3f603063a1918d582a10f58",
    "event_data": {
      "message_id": 5,
      "segment_id": 0,
      "campaign_id": 130
    },
    "event_date": "2016-11-15T09:12:28.514388956Z",
    "custom_data": {
      "email": "eddie@test-ukr.net",
      "_fname": "Eddie"
    },
    "is_test": false
  },
  {
    "event_type": "deliv",
    "list_id": 50,
    "account_id": 161,
    "_xxh": "bb42e541c7267fa6",
    "_md5": "593b250db256a764da8d2da4be694b31",
    "event_data": {
      "message_id": 5,
      "segment_id": 0,
      "campaign_id": 130
    },
    "event_date": "2016-11-15T09:12:28.569124547Z",
    "custom_data": {
      "email": "simon@test-gmail.com",
      "_fname": "Simon"
    },
    "is_test": false
  }
]
```

Отправить сообщение в очередь RabbitMQ

Каждое единичное событие будет отправлено немедленно в очередь брокера сообщений. Может использоваться как локальный RabbitMQ, входящий в состав платформы, так и расположенный на внешнем хосте брокер. В настройках укажите **хост** и **порт** брокера, **имя** и **пароль** пользователя для авторизации, имя **виртуального хоста** и **название очереди**, в которую будут попадать сообщения о событиях.

CAUTION

RabbitMQ пользователь должен иметь доступ к выбранному виртуальному хосту. Создать *Виртуальный хост (virtual host)* и дать пользователю RabbitMQ право его использовать нужно в самой RabbitMQ.



TIP

При отправке в RMQ можно указывать или не указывать имя очереди. Если имя очереди не будет передано, то оно будет установлено в соответствии с именем события (см. таблицу **Типы событий для захвата**).

Пример сообщения о клике на ссылку в Email

```
{
  "event_type": "click",
  "event_date": "2018-03-04T13:13:19.034315796Z",
  "list_id": 111,
  "_xxh": "ad1850e0d1d9481e",
  "_md5": "145bb30f7ddf1184cb560bc4b7705646",
  "is_test": false,
  "event_data": {
    "browser": "Firefox",
    "campaign_id": 973,
    "city": "",
    "country": "",
    "device": "web",
    "ip": "10.9.0.14",
    "lat": 0,
    "launch_id": "00000017000003cd5a9bf01a",
    "link_url": "http://www.ford.com",
    "lon": 0,
    "message_id": 17,
    "os": "Linux x86_64",
    "postal_code": "",
    "region": "",
    "segment_id": 0,
    "send_message_id": "5a9bf01a17_3cd_11_6f_0_2.5a8c045fc1dde44cbf718f78",
    "tz": "",
    "user_agent": "Mozilla/5.0 (X11; Linux x86_64; rv:59.0) Gecko/20100101 Firefox/59.0"
  },
  "custom_data": {
    "email": "alice@example.org"
  },
  "account_id": 23
}
```

Отправить сообщение в exchange RabbitMQ

Exchange — это точка обмена, другими словами, еще один игрок между отправителем сообщения и очередью. Эта точка обмена распределяет сообщения в одну или несколько очередей.

Для настройки отправки сообщения в exchange RabbitMQ укажите хост, порт, имя пользователя, пароль и имя виртуально хоста.

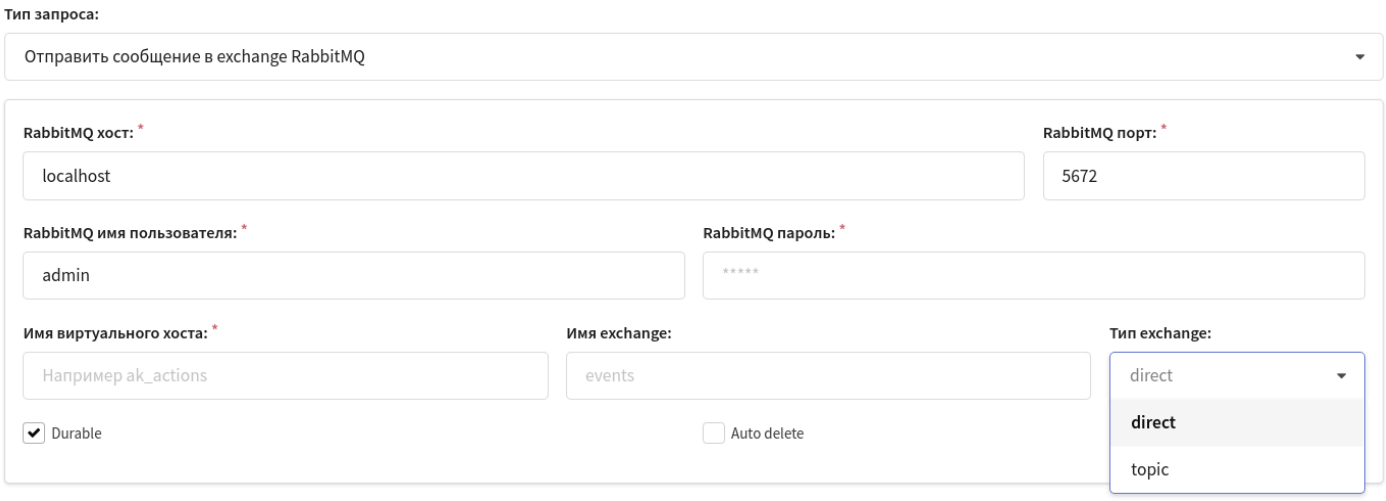
> **TIP**
>
> При отправке в RMQ можно указывать или не указывать имя exchange. Если имя exchange не будет передано, то оно будет установлено в соответствии с именем события (см. таблицу **Типы событий для захвата**).

Далее нас просят выбрать **тип exchange:**

1. **Тип direct** позволяет отправлять сообщения только одной очереди. Причем в ту очередь, которая подписана на соответствующий ключ маршрутизации (routing key). Ключ формируется автоматически по шаблону: `<db_id>.<event_type>`.
2. **Тип topic** позволяет отправлять сообщения в несколько очередей, которые подписаны на шаблон ключа маршрутизации.

По умолчанию активирована опция **durable**. Это означает, что точка обмена будет постоянно храниться на диске и сможет пережить перезапуск сервера. Если убрать галочку с этого поля, то exchange станет временным и в случае перезапуска сервера удалится.

Также стоит обратить внимание на поле **Auto delete** (автоудаление). Если активирвоать это поле, то exchange будет удален, как только будут удалены все связанные с ним очереди.



Пример сообщения об уникальном клике в Email

```
{
  "event_type": "uclick",
  "event_date": "2021-03-05T14:08:47.18953529+03:00",
  "list_id": 123456,
  "profile_id": "000000000000000000000000",
  "_xxh": "61be4d0b871502d4",
  "_md5": "a42c44fc77533b43ffd569ed4c83e377",
  "is_test": true,
  "event_data": {
    "browser": "Firefox",
    "campaign_id": 286,
    "city": "",
    "country": "",
    "device": "web",
    "ip": "192.168.0.58",
    "lat": 0,
    "launch_id": "2_5W_4HECAgq2o6H",
    "link_url": "https://emailtemplate.ga/?
utm_source=email&utm_medium=&utm_campaign=286&utm_term=kirk.mckinney83@example.com&utm_cont
ent=9",
    "lon": 0,
    "message_id": 9,
    "os": "Ubuntu",
    "postal_code": "",
    "region": "",
    "segment_id": 0,
    "send_message_id": "w4HECAgxdpsf_2_5W_a_6__3_.2MpNdKgZFnZxJbCiJ",
    "tz": "",
    "user_agent": "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:81.0) Gecko/20100101
Firefox/81.0"
  },
  "custom_data": {
    "email": "kirk.mckinney83@example.com"
  },
  "account_id": 1,
  "sends_trying_num": 1,
  "is_static": false
}
```

Отправить сообщение в Kafka broker

Для настройки отправки событий в Apache Kafka, укажите:

- хост
- порт
- название темы
- идентификатор раздела (partition)

Активируйте **TLS** или **SASL**, если это необходимо.

Для **SASL-аутентификации** поддерживаются следующие типы:

- **SASL Plaintext**
- **SCRAM-SHA-256**
- **SCRAM-SHA-512**

При применении самоподписанного сертификата, если он не внесен в доверенные, укажите цепочку CA-сертификатов в PEM формате. При этом также учитывается глобальная опция `ALLOW_INSECURE_REQUESTS` в файле конфигурации `main.json`. Если она установлена, валидность сертификатов не проверяется. Если применяется авторизация с клиентским сертификатом, то необходимо указать приватный ключ клиента и сертификат в PEM-формате.

Скрипт для создания сертификатов Expand source

```
#!/usr/bin/env bash

set -e

KEYSTORE_FILENAME="kafka.keystore.jks"
VALIDITY_IN_DAYS=3650
DEFAULT_TRUSTSTORE_FILENAME="kafka.truststore.jks"
TRUSTSTORE_WORKING_DIRECTORY="truststore"
KEYSTORE_WORKING_DIRECTORY="keystore"
CA_CERT_FILE="ca-cert"
KEYSTORE_SIGN_REQUEST="cert-file"
KEYSTORE_SIGN_REQUEST_SRL="ca-cert.srl"
KEYSTORE_SIGNED_CERT="cert-signed"

function file_exists_and_exit() {
echo "'$1' cannot exist. Move or delete it before"
echo "re-running this script."
exit 1
}

if [ -e "$KEYSTORE_WORKING_DIRECTORY" ]; then
file_exists_and_exit $KEYSTORE_WORKING_DIRECTORY
fi

if [ -e "$CA_CERT_FILE" ]; then
file_exists_and_exit $CA_CERT_FILE
fi

if [ -e "$KEYSTORE_SIGN_REQUEST" ]; then
file_exists_and_exit $KEYSTORE_SIGN_REQUEST
fi

if [ -e "$KEYSTORE_SIGN_REQUEST_SRL" ]; then
file_exists_and_exit $KEYSTORE_SIGN_REQUEST_SRL
fi

if [ -e "$KEYSTORE_SIGNED_CERT" ]; then
file_exists_and_exit $KEYSTORE_SIGNED_CERT
fi

echo
echo "Welcome to the Kafka SSL keystore and truststore generator script."

echo
echo "First, do you need to generate a trust store and associated private key,"
echo "or do you already have a trust store file and private key?"
echo
echo -n "Do you need to generate a trust store and associated private key? [yn] "
```

```
echo -n "Do you need to generate a trust store and associated private key? [yn]"
read generate_trust_store

trust_store_file=""
trust_store_private_key_file=""

if [ "$generate_trust_store" == "y" ]; then
if [ -e "$TRUSTSTORE_WORKING_DIRECTORY" ]; then
file_exists_and_exit $TRUSTSTORE_WORKING_DIRECTORY
fi

mkdir $TRUSTSTORE_WORKING_DIRECTORY
echo
echo "OK, we'll generate a trust store and associated private key."
echo
echo "First, the private key."
echo
echo "You will be prompted for:"
echo " - A password for the private key. Remember this."
echo " - Information about you and your company."
echo " - NOTE that the Common Name (CN) is currently not important."

openssl req -new -x509 -keyout $TRUSTSTORE_WORKING_DIRECTORY/ca-key \
-out $TRUSTSTORE_WORKING_DIRECTORY/ca-cert -days $VALIDITY_IN_DAYS

trust_store_private_key_file="$TRUSTSTORE_WORKING_DIRECTORY/ca-key"

echo
echo "Two files were created:"
echo " - $TRUSTSTORE_WORKING_DIRECTORY/ca-key -- the private key used later to"
echo " sign certificates"
echo " - $TRUSTSTORE_WORKING_DIRECTORY/ca-cert -- the certificate that will be"
echo " stored in the trust store in a moment and serve as the certificate"
echo " authority (CA). Once this certificate has been stored in the trust"
echo " store, it will be deleted. It can be retrieved from the trust store via:"
echo " $ keytool -keystore <trust-store-file> -export -alias CARoot -rfc"

echo
echo "Now the trust store will be generated from the certificate."
echo
echo "You will be prompted for:"
echo " - the trust store's password (labeled 'keystore'). Remember this"
echo " - a confirmation that you want to import the certificate"

keytool -keystore $TRUSTSTORE_WORKING_DIRECTORY/$DEFAULT_TRUSTSTORE_FILENAME \
-alias CARoot -import -file $TRUSTSTORE_WORKING_DIRECTORY/ca-cert

trust_store_file="$TRUSTSTORE_WORKING_DIRECTORY/$DEFAULT_TRUSTSTORE_FILENAME"

echo
```

```
echo "$TRUSTSTORE_WORKING_DIRECTORY/$DEFAULT_TRUSTSTORE_FILENAME was created."

# don't need the cert because it's in the trust store.
# rm $TRUSTSTORE_WORKING_DIRECTORY/$CA_CERT_FILE
else
echo
echo -n "Enter the path of the trust store file. "
read -e trust_store_file

if ! [ -f $trust_store_file ]; then
echo "$trust_store_file isn't a file. Exiting."
exit 1
fi

echo -n "Enter the path of the trust store's private key. "
read -e trust_store_private_key_file

if ! [ -f $trust_store_private_key_file ]; then
echo "$trust_store_private_key_file isn't a file. Exiting."
exit 1
fi
fi

echo
echo "Continuing with:"
echo " - trust store file: $trust_store_file"
echo " - trust store private key: $trust_store_private_key_file"

mkdir $KEYSTORE_WORKING_DIRECTORY

echo
echo "Now, a keystore will be generated. Each broker and logical client needs its own"
echo "keystore. This script will create only one keystore. Run this script multiple"
echo "times for multiple keystores."
echo
echo "You will be prompted for the following:"
echo " - A keystore password. Remember it."
echo " - Personal information, such as your name."
echo " NOTE: currently in Kafka, the Common Name (CN) does not need to be the FQDN of"
echo " this host. However, at some point, this may change. As such, make the CN"
echo " the FQDN. Some operating systems call the CN prompt 'first / last name'"
echo " - A key password, for the key being generated within the keystore. Remember this."

# To learn more about CNs and FQDNs, read:
# https://docs.oracle.com/javase/7/docs/api/javax/net/ssl/X509ExtendedTrustManager.html

keytool -keystore $KEYSTORE_WORKING_DIRECTORY/$KEYSTORE_FILENAME \
-alias localhost -validity $VALIDITY_IN_DAYS -genkey -keyalg RSA
```

```
echo
echo "'$KEYSTORE_WORKING_DIRECTORY/$KEYSTORE_FILENAME' now contains a key pair and a"
echo "self-signed certificate. Again, this keystore can only be used for one broker or"
echo "one logical client. Other brokers or clients need to generate their own keystores."

echo
echo "Fetching the certificate from the trust store and storing in $CA_CERT_FILE."
echo
echo "You will be prompted for the trust store's password (labeled 'keystore')"

keytool -keystore $trust_store_file -export -alias CARoot -rfc -file $CA_CERT_FILE

echo
echo "Now a certificate signing request will be made to the keystore."
echo
echo "You will be prompted for the keystore's password."
keytool -keystore $KEYSTORE_WORKING_DIRECTORY/$KEYSTORE_FILENAME -alias localhost \
-certreq -file $KEYSTORE_SIGN_REQUEST

echo
echo "Now the trust store's private key (CA) will sign the keystore's certificate."
echo
echo "You will be prompted for the trust store's private key password."
openssl x509 -req -CA $CA_CERT_FILE -CAkey $trust_store_private_key_file \
-in $KEYSTORE_SIGN_REQUEST -out $KEYSTORE_SIGNED_CERT \
-days $VALIDITY_IN_DAYS -CAcreateserial
# creates $KEYSTORE_SIGN_REQUEST_SRL which is never used or needed.

echo
echo "Now the CA will be imported into the keystore."
echo
echo "You will be prompted for the keystore's password and a confirmation that you want to"
echo "import the certificate."
keytool -keystore $KEYSTORE_WORKING_DIRECTORY/$KEYSTORE_FILENAME -alias CARoot \
-import -file $CA_CERT_FILE
rm $CA_CERT_FILE # delete the trust store cert because it's stored in the trust store.

echo
echo "Now the keystore's signed certificate will be imported back into the keystore."
echo
echo "You will be prompted for the keystore's password."
keytool -keystore $KEYSTORE_WORKING_DIRECTORY/$KEYSTORE_FILENAME -alias localhost -import \
-file $KEYSTORE_SIGNED_CERT

echo

echo "Creating keys directory and copy needed files to them"
echo
mkdir platform_keys
mv truststore/ca-cert platform_keys/
```

```
mv truststore/ca-cert platform_keys/
mv cert-signed platform_keys/

echo "Create keys for platform"
echo
echo
echo "Convert kafka.keystore.jks to kafka.keystore.p12"
echo
keytool -importkeystore -srckeystore keystore/kafka.keystore.jks \
-destkeystore kafka.keystore.p12 -srcalias localhost \
-srcstoretype jks -deststoretype pkcs12
echo
echo "Convert kafka.keystore.p12 to kafka.keystore.pem"
echo
openssl pkcs12 -in kafka.keystore.p12 -out kafka.keystore.pem

echo
echo "Clear password on .pem file"

echo

openssl rsa -in kafka.keystore.pem -out platform_keys/key.pem

echo
echo "All Done!"
echo
```

Настройки обратного запроса

Тип запроса:

Отправить сообщение в Кафка брокер

Kafka хост: *

kafka.example.com

Kafka порт: *

9092

Kafka название темы: *

test

Kafka идентификатор раздела: *

0

Включить TLS

Клиентский сертификат | CA сертификат

PEM данные:

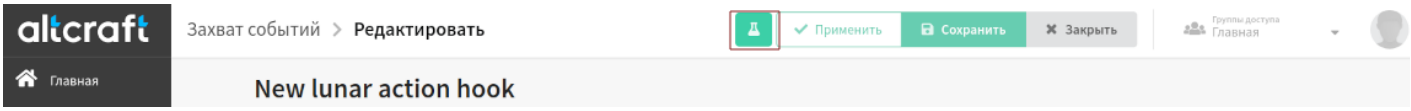
```
-----BEGIN CERTIFICATE-----
MIIDODCCAiACFCkMYJlQHcB/6oTyRsOL4GdqD0V9MA0GCSqGSIb3DQEBCwUAMEUx
CzAJBgNVBAYTAkFVMRMwEQYDVQQIDApTb21lLVN0YXRlMSEwHwYDVQQKDBhJbnRl
cm5ldCBXaWRnaXRzIFB0eSBMdGQwHhcNMjAwOTI0MTEyNDUzWhcNMzAwOTIyMTEy
NDUzWjBsMRAwDgYDVQQGEwdVbmtub3duMRAwDgYDVQQIEwdVbmtub3duMRAwDgYD
VQQHEwdVbmtub3duMRAwDgYDVQQKEwdVbmtub3duMRAwDgYDVQQLEwdVbmtub3du
MRAwDgYDVQQDEwdVbmtub3duMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKC
AQEAu8czHOvP5sqSoNq3PjG1ykzGFfeEHwlFBCtPUdcgUfgOg8xu9wRadiJOXvu+
uTGu8WdXL4hDbg+eIi8UP4iHAyO63IXkRV3QKKuQlul5KMQERtFJAK6W/t11pizq
```

Приватный ключ:

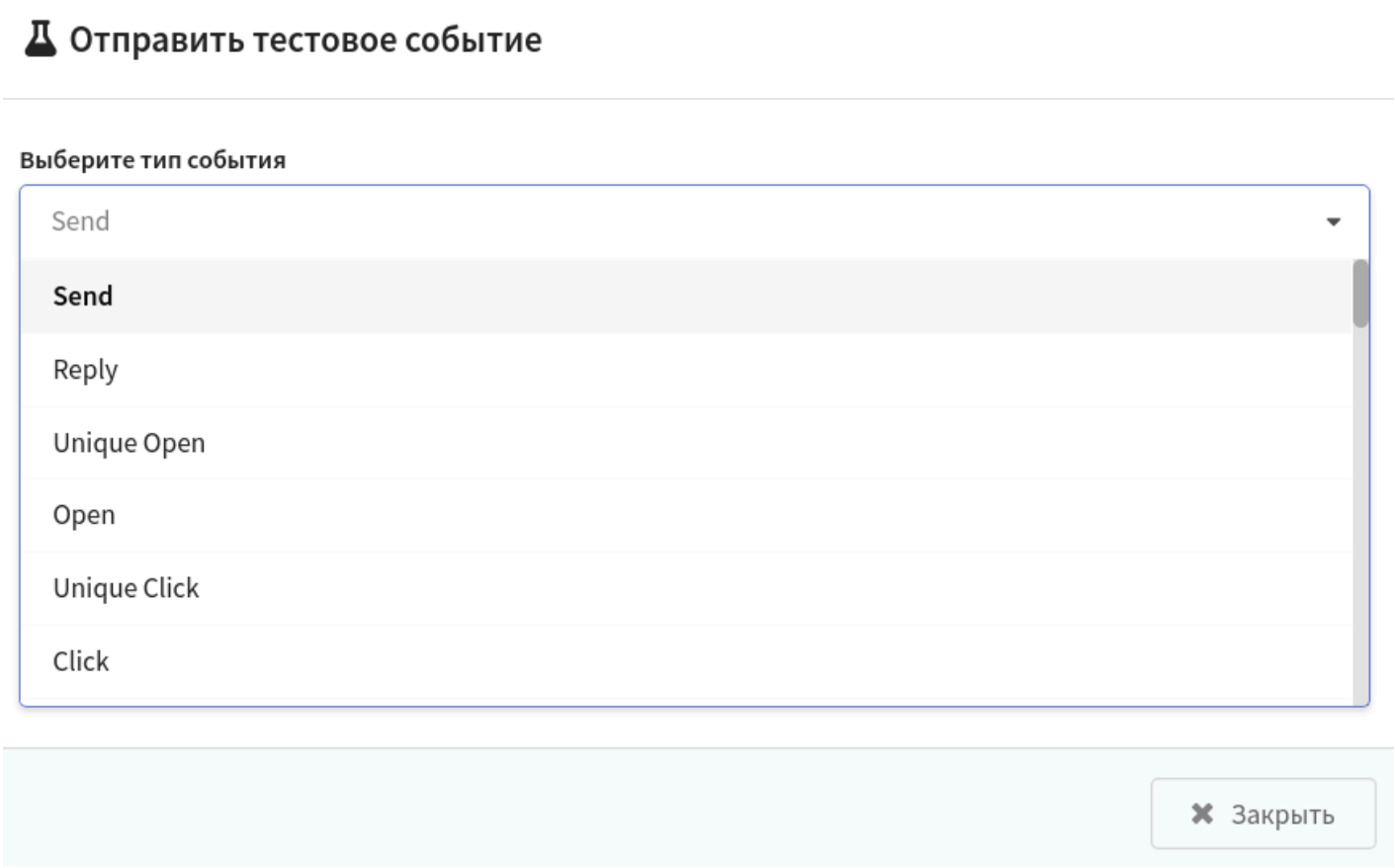
```
-----BEGIN RSA PRIVATE KEY-----
MIIEpgIBAAKCAQEAu8czHOvP5sqSoNq3PjG1ykzGFfeEHwlFBCtPUdcgUfgOg8xu
9wRadiJOXvu+uTGu8WdXL4bDbg+eIi8UP4iHAyO63IXkRV3QKKuQlul5KMQERtFJ
AK6W/t11njzqM7PNsJ/51J65DairMUjIpihm6Ny8F1ng6rCB3DwuzFBm1OPgXc/4
PuSb6Z14POQzM8+Yp/d+Ev5BH+GjCk4+/9Y8XqNswrdA6Mt5rnhcFVmRzskdKXzs
2QZD/+ul9eBHi+Ng7S04UOpuecM+iauZdAO0CBMmh0lcHnhM1WmTeqCC+KOn7bTN
ZMhcMBq3gCJRXvkXs3pFmsnRDQxZg6bj9cuKQwIDAQABAoIBAQC6DaJoSoIBM7/a
3zrmxcxQuIcgZtGjCsU2X09FULj2XqjLEDj2SXr0SAOM+zS7PC4RhIoPTiC/IXxW
dH1Fqhr8El/odkV6MXv5f1xRGt3wdV+iZ2RE57g2gOkMM2D+f24TTi7025uRpAib
```

Предварительное тестирование события

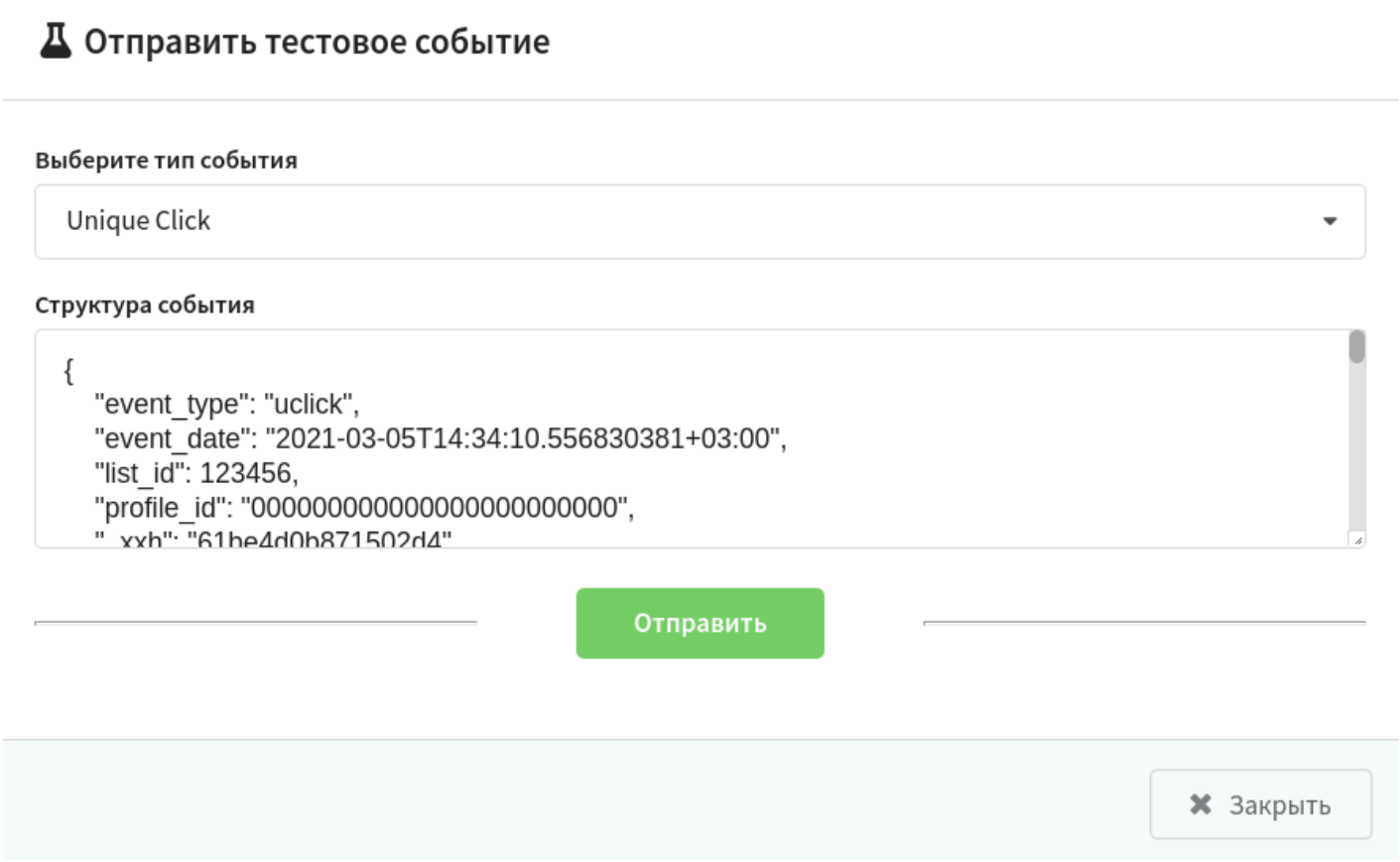
При работе с захватом событий есть возможность тестирования любого события непосредственно из интерфейса платформы. Для этого необходимо нажать на иконку в виде колбы в верхней панели.



В новом окне укажите тип события, которое вы хотите протестировать:



Автоматически будет сгенерирован пример структуры выбранного события. В данную структуру можно вносить изменения.

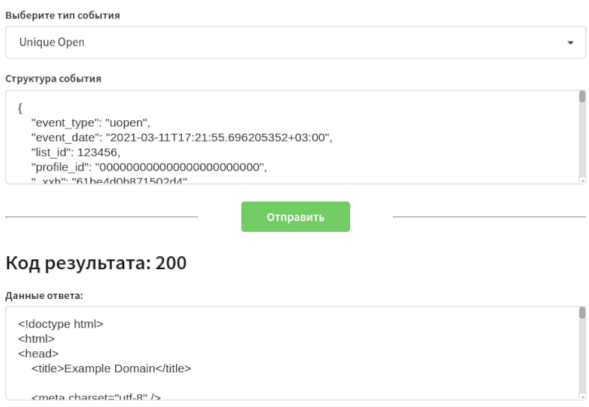
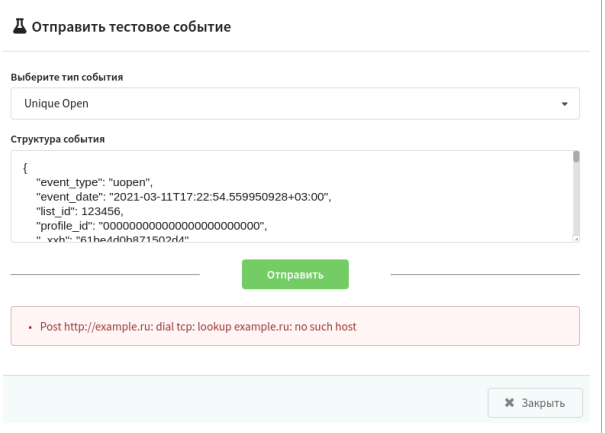


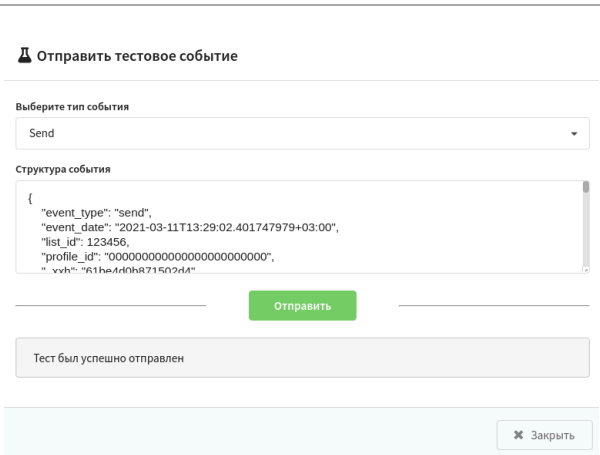
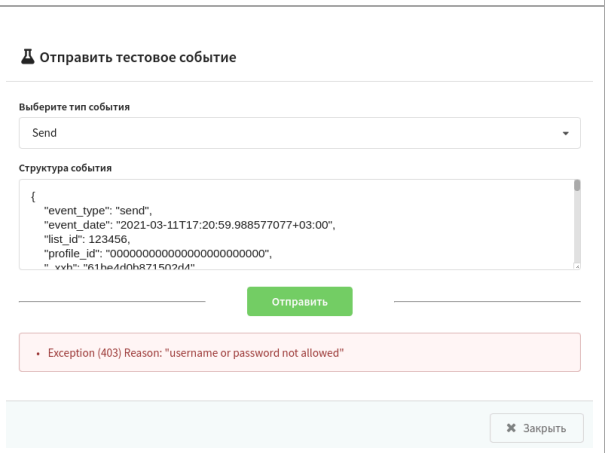
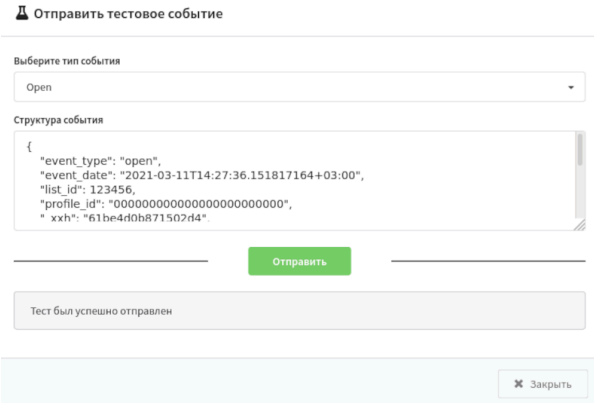
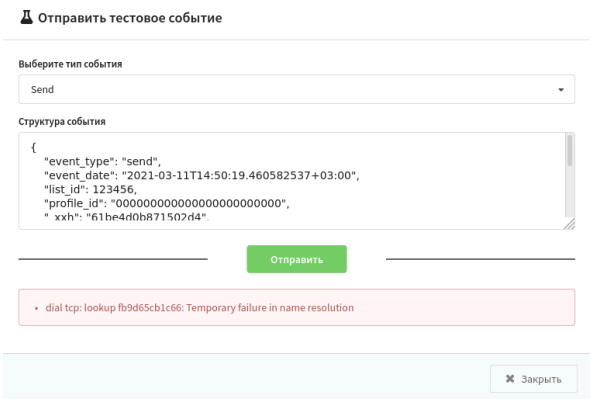
Нажмите кнопку **Отправить**, чтобы протестировать событие.

Если сообщение было **успешно** отправлено посредством HTTP запросом, то вы увидите **код резутата: 200**, а также данные ответа.

Если же сообщение было успешно отправлено в очередь или exchange RabbitMQ или в Kafka broker, то вы увидете текст: **“Тест был успешно отправлен”**.

В противном случае, внизу появится красное поле, где вы сможете узнать код ошибки (в случае HTTP запроса) и ее причину.

Тип запроса	Сообщение отправлено успешно	Ошибка при отправке сообщения
JSON запрос батчем		

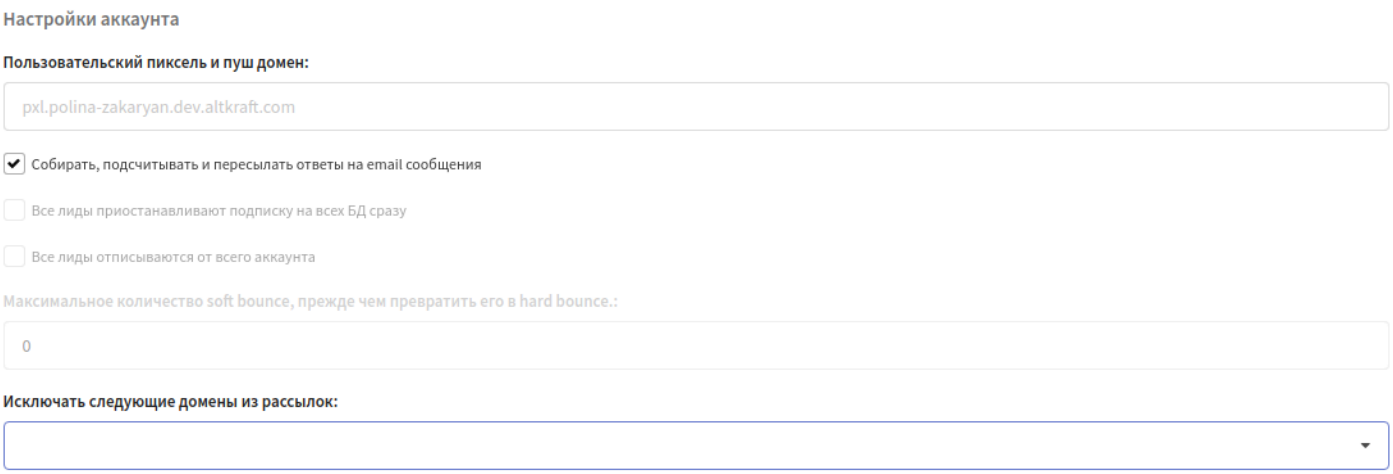
Тип запроса	Сообщение отправлено успешно	Ошибка при отправке сообщения
В очередь / exchange RabbitMQ		
В Kafka broker		

Настройки аккаунта

	В меню **Настройки аккаунта** собраны параметры конфигурации, которые применяются глобально — ко всем процессам и объектам в аккаунте. Настройки влияют на различный функционал.

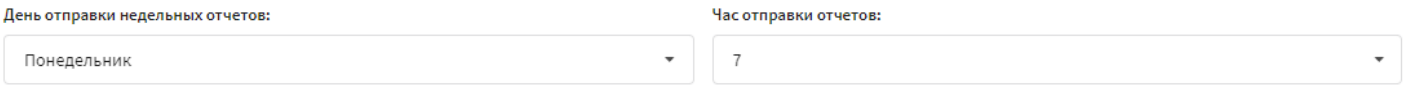
Администрирование

Здесь собраны настройки, связанные с отправкой email и подписками:



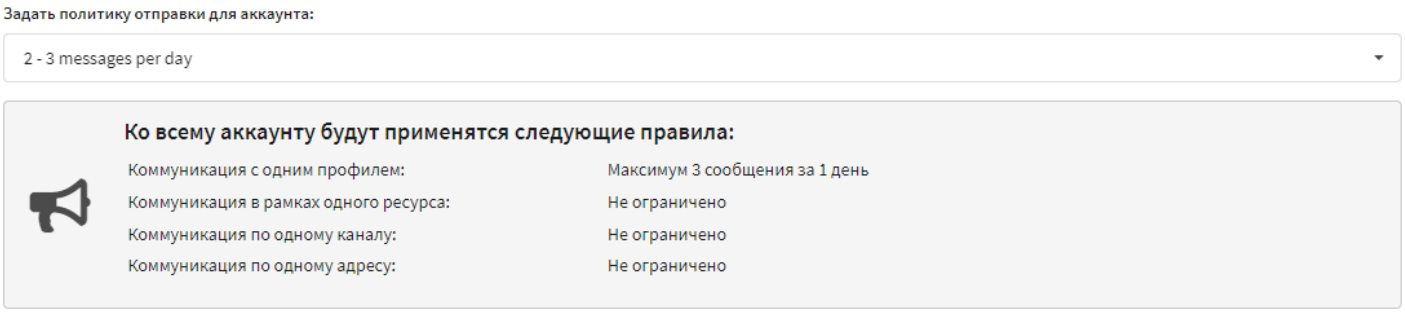
Настройка отчётов

В этой части вы можете установить день недели и час для отправки отчётов на электронную почту пользователей платформы. Содержание отчётов настраивается для каждого пользователя отдельно.



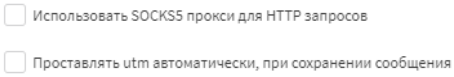
Глобальная политика отправки

В следующем блоке вы можете установить глобальную политику отправки. Она будет применяться ко всем ресурсам и рассылкам, для которых не установлена собственная политика:



Дополнительные опции

В нижней части формы находятся опции **проксирования** запросов трекинга и автопроставления **utm меток** при сохранении [шаблона](#) сообщения:

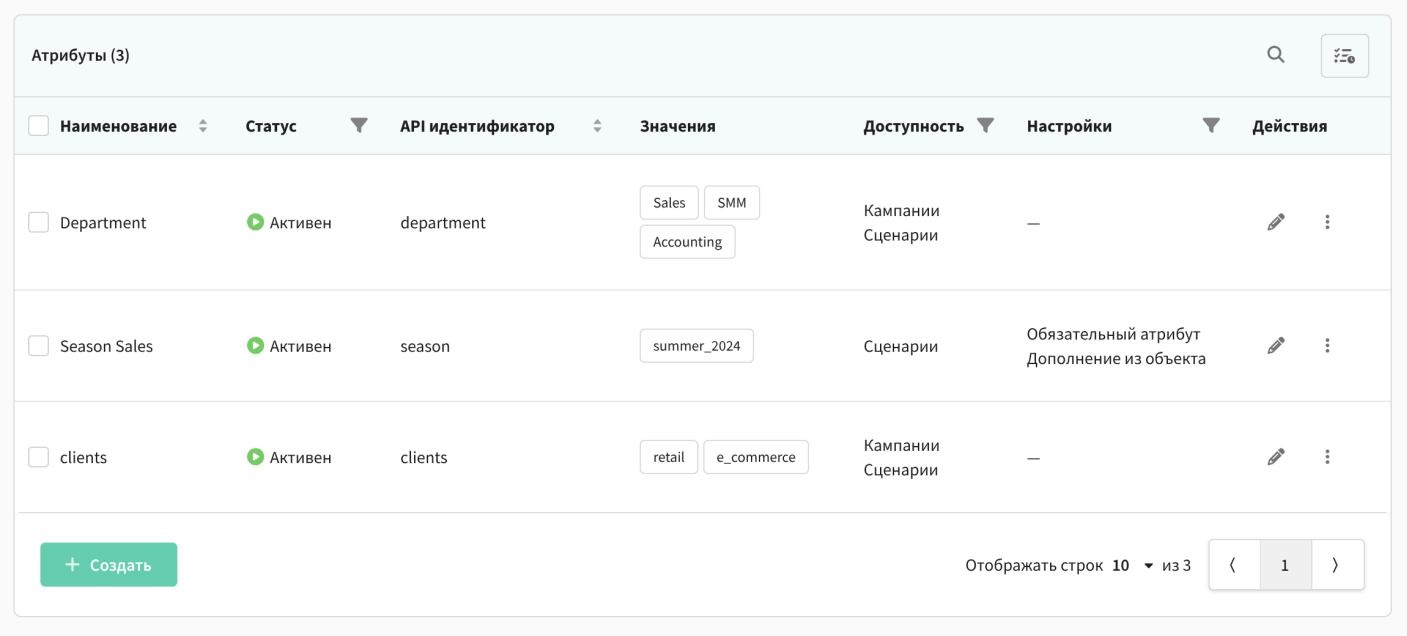


Настройки атрибутов

Что такое атрибуты сущностей

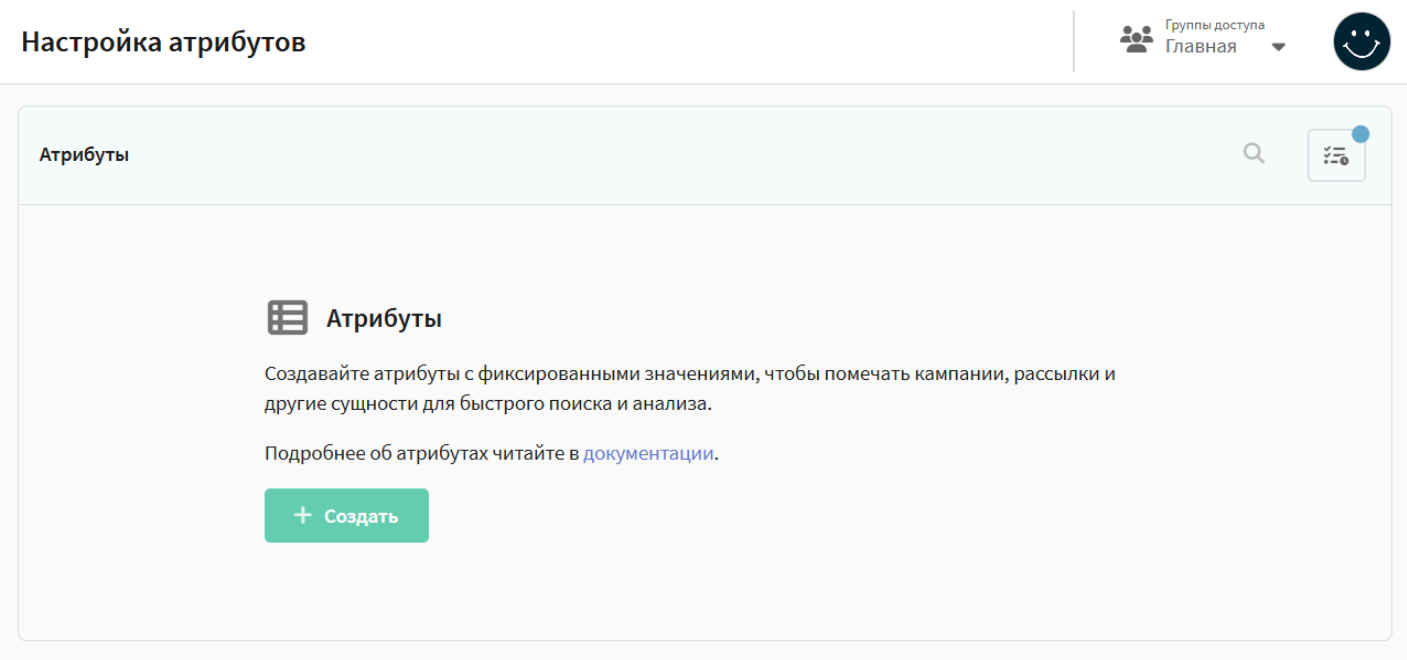
Атрибуты — специальные пользовательские поля, добавляемые к сущностям платформы. Они представляют собой определённый **параметр** с набором возможных **значений**. С помощью атрибутов можно не только маркировать для поиска определённые группы сущностей, но и использовать их как признак сегментации, переменную или включать их значения в API-запросы. Атрибуты можно назначать для **кампаний**, **рассылок** и **сценариев**.

> **TIP: Наследование атрибутов**
>
> Атрибуты иерархически **наследуются** по цепочке: **Кампания** → **Сценарий** → **Рассылка**. Это означает, что если для рассылки не задано значение атрибута, будет использовано значение сценария, а если не задано и там — значение кампании.

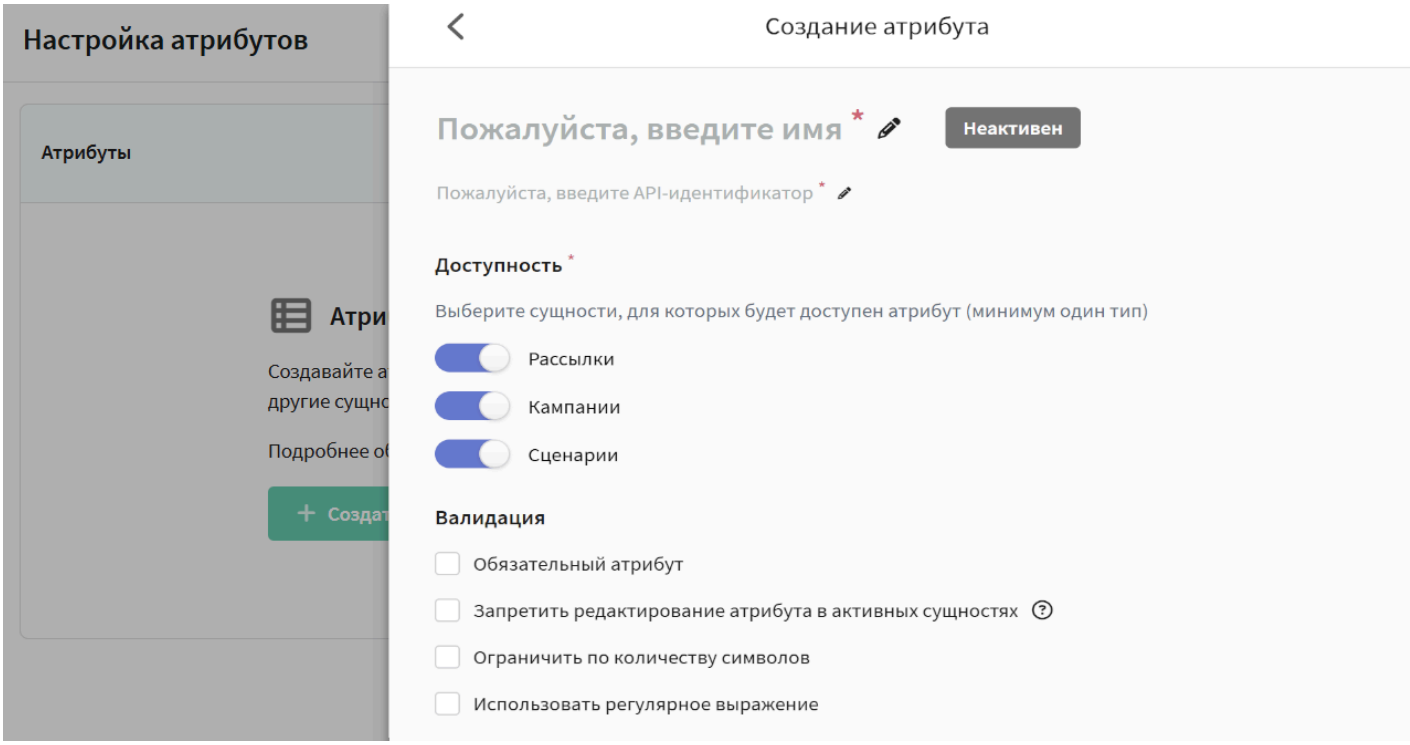


Создание и настройка

Чтобы перейти к созданию и настройке атрибутов, откройте раздел **Настройки** → **Настройка атрибутов**. Откроется панель управления атрибутами. Создайте первый атрибут, нажав на соответствующую кнопку:



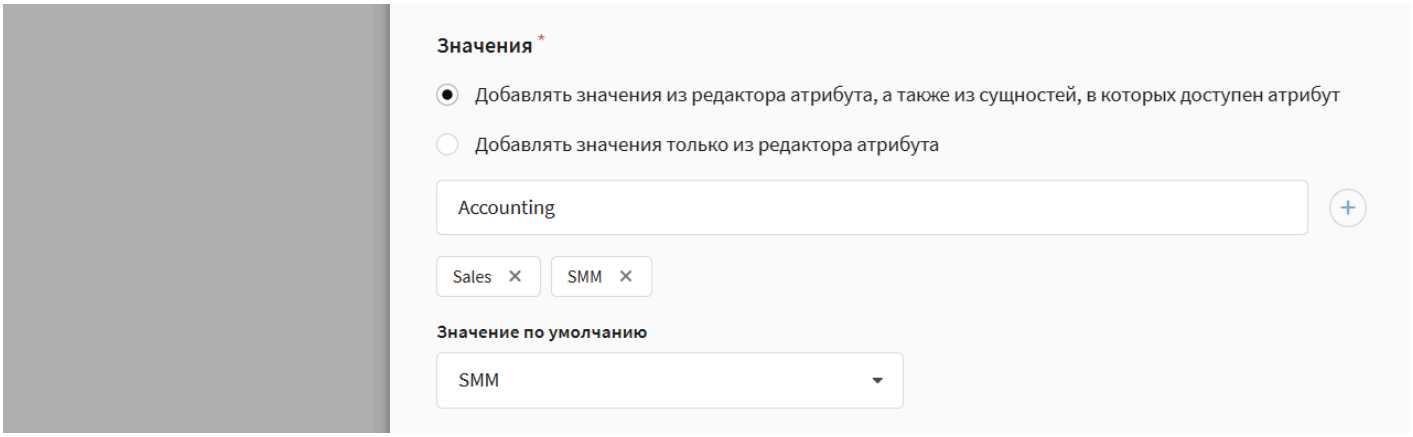
Появится окно с настройками атрибута. Прежде всего задайте для него **название** и **API-идентификатор**. Далее отметьте те сущности, для которых будет доступен атрибут — **кампании**, **рассылки** или **сценарии** (как минимум один тип сущности):



Затем отметьте нужные параметры:

- **Обязательный атрибут** — для любой созданной сущности, в которой доступен атрибут, необходимо указать его значение. Без этого кампания, рассылка или сценарий не будут сохранены.
- **Запретить редактирование атрибута в активных сущностях** — запрещает изменение значения атрибута в уже активированных сущностях.
- **Ограничить по количеству символов** — устанавливает максимальную длину значений атрибута.
- **Использовать регулярное выражение** — проверяет значение атрибута на соответствие регулярному выражению.

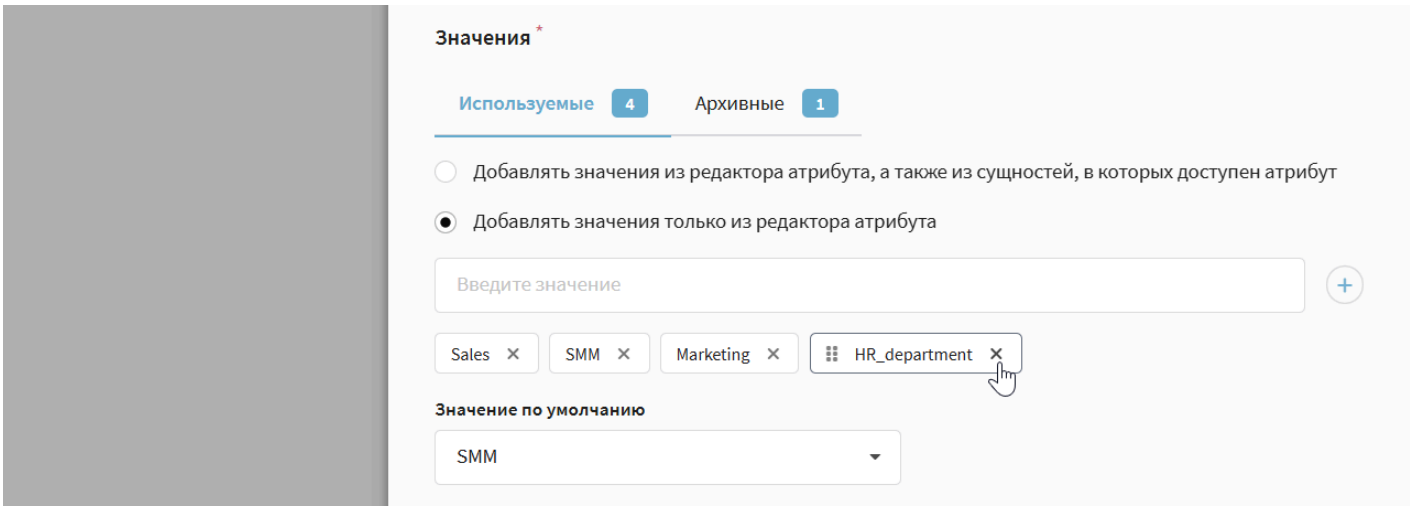
Следующим шагом определите **значения** атрибута. Вы также можете разрешить пользователю не только выбирать значение из заранее созданных, но и добавлять новые при редактировании сущности. Для каждого атрибута можно установить до 30 значений:



TIP

Атрибуты иерархически **наследуются** по цепочке: **Кампания** → **Сценарий** → **Рассылка**. Это означает, что если для рассылки не задано значение атрибута, будет использовано значение сценария, а если не задано и там — значение кампании.

В дальнейшем значения можно не только удалять, но и **архивировать**:



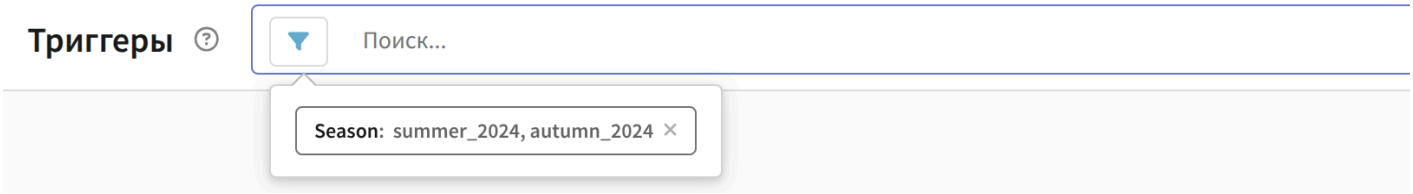
Архивированное значение нельзя добавить новым сущностям, но оно сохраняется в уже существующих и может использоваться при фильтрации и поиске. Такие значения отмечаются специальным образом:

При необходимости назначьте значение атрибута по умолчанию. Затем **сохраните** и **активируйте** атрибут. После сохранения вы можете деактивировать, клонировать или удалить его из списка.

Использование атрибутов в платформе

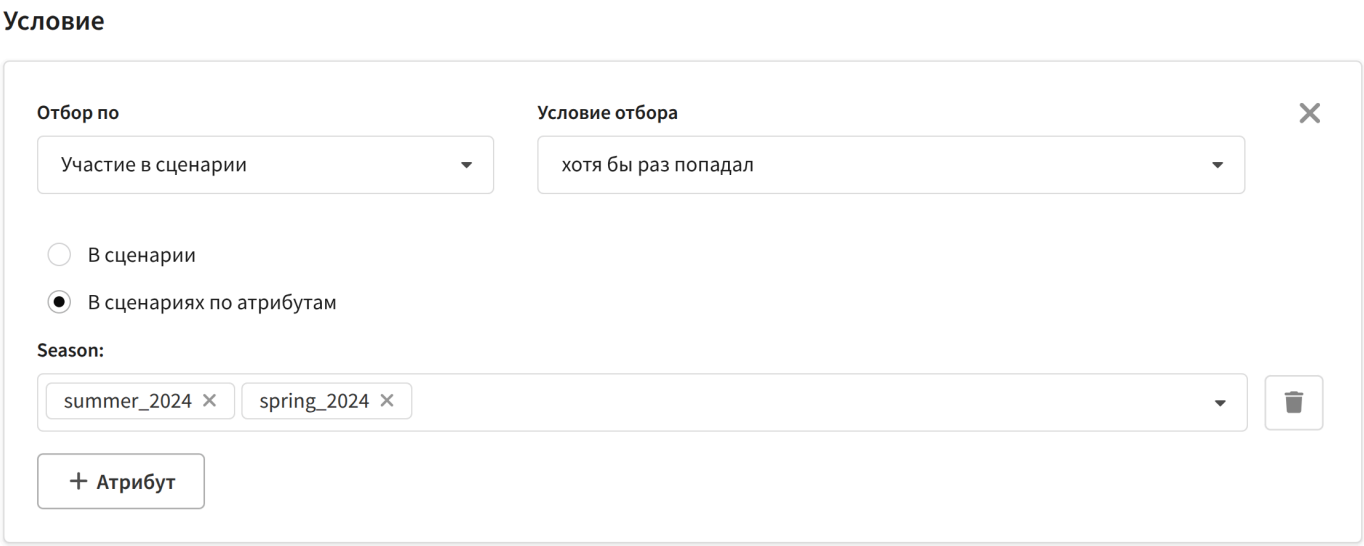
Поиск и фильтрация

Значения атрибутов используются для поиска и фильтрации сущностей платформы:



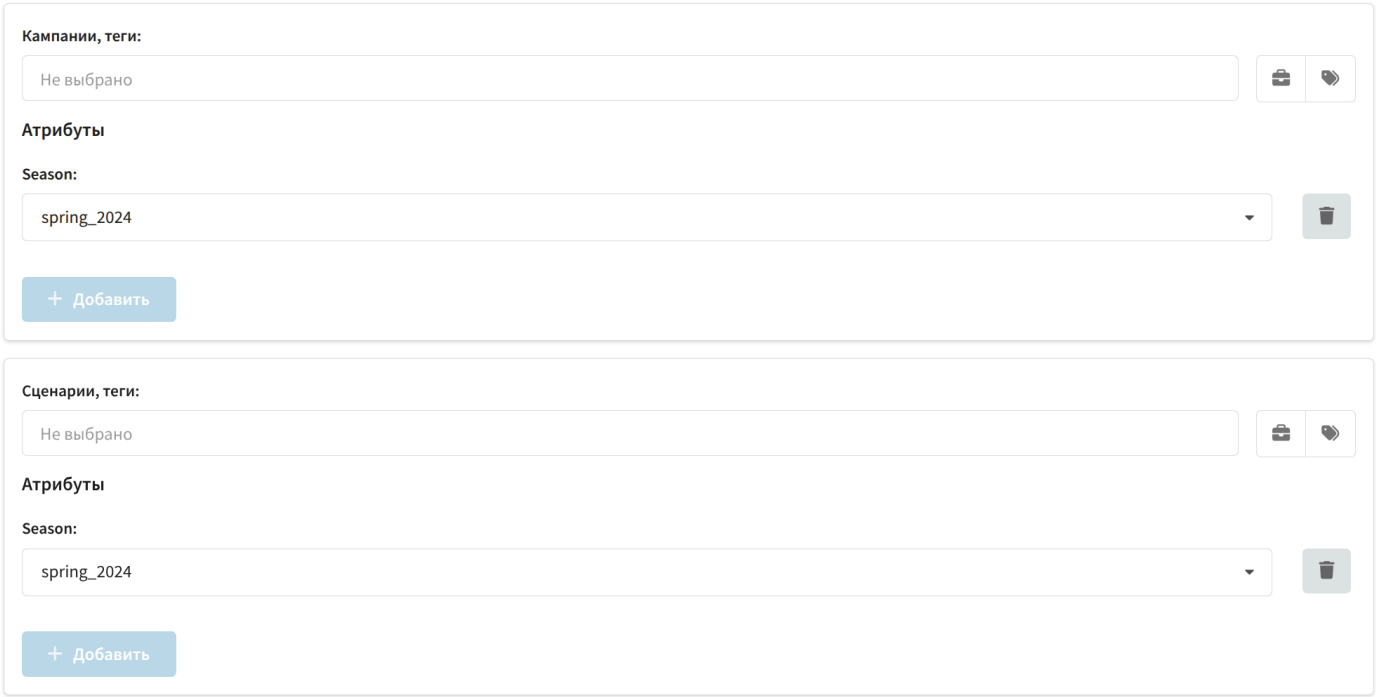
Сегментация профилей

При сегментации по действиям подписчиков можно использовать значения атрибутов для выбора кампаний, рассылок и сценариев для отбора профилей:



Исключения ГКГ

Пользователь может отключить глобальные контрольные группы для сценариев, рассылок и кампаний с определённым значением атрибута:



Исключения ЛКГ

В настройках локальной контрольной группы (ЛКГ) можно фильтровать профили по значениям атрибутов кампаний, сценариев и рассылок. Это позволяет гибко настраивать правила исключения профилей из контрольной группы.

Подстановка переменной

Значение атрибута можно подставлять в шаблоны и сценарии с помощью переменной `{attribute_value("attribute_name")}`, где вместо `"attribute_name"` — API-название атрибута.

Примеры использования:

- `{attribute_value("test_attribute")}` — подставит значение атрибута “test_attribute”
- `{attribute_value("campaign_type")}` — подставит значение атрибута “campaign_type”
- `{attribute_value("priority")}` — подставит значение атрибута “priority”

API-запросы

Параметры с названием и значением атрибута доступны в некоторых API-запросах к платформе. Подробнее об API можно прочесть в этой статье.

Поисковые теги: создание и применение

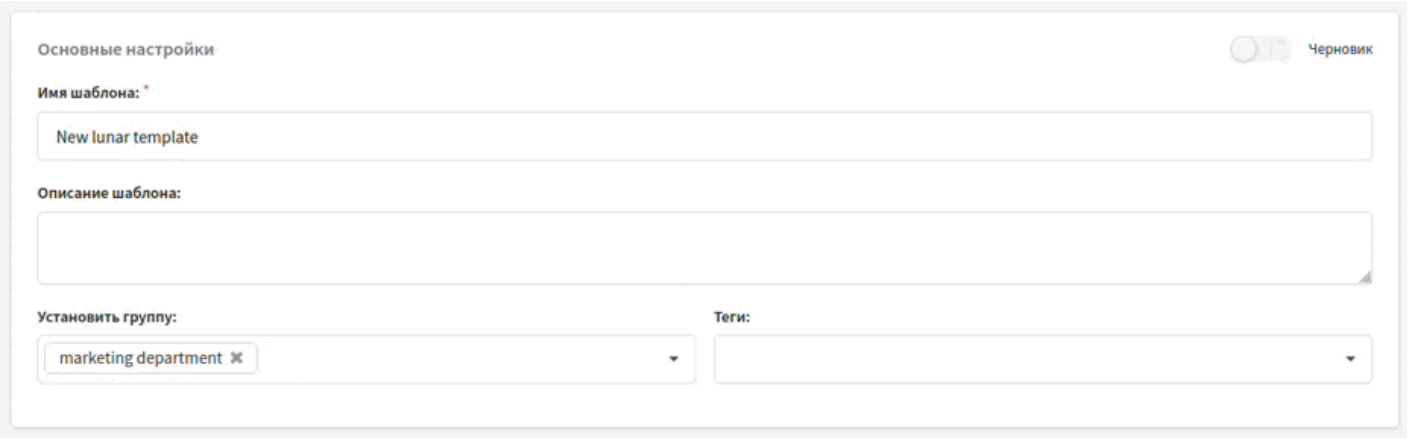
Тег — это метка, которую можно назначать некоторым объектам платформы, чтобы в дальнейшем быстро находить эти объекты. Тегами обычно связывают объекты с общей тематикой. Например, рассылки, которые запускаются в рамках одной акции, или шаблоны, нацеленные на одну аудиторию.

Каким объектам можно назначать теги:

- базы данных
- ресурсы
- сегменты
- шаблоны
- импорты по расписанию
- стоп-списки
- рассылки
- кампании
- сценарии
- формы
- продукты и заказы

Как создать тег

Тег создается непосредственно в объекте, которому вы этот тег хотите назначить. Перейдите в окно редактирования объекта, установите курсор в поле **“Теги”** и введите текст, затем нажмите **Add** или **↵ Enter**.



Чтобы открепить тег от объекта, нажмите крестик.

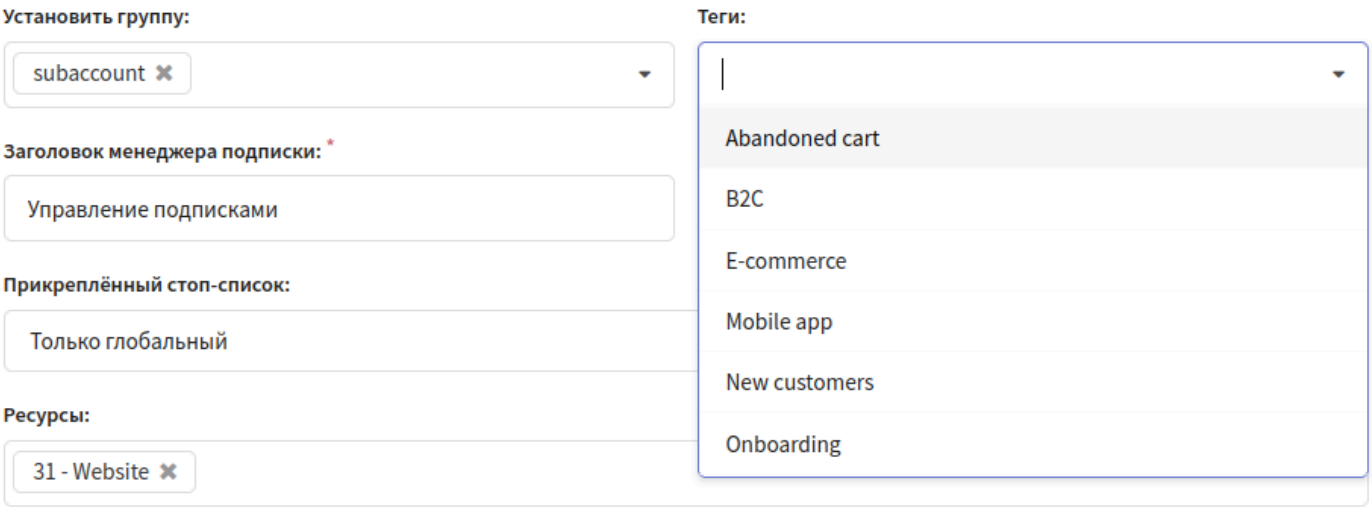
Объекту можно назначить любое количество тегов. Названия тегов никак не ограничены:

- любое количество символов
- кириллица и латиница
- пробелы
- специальные символы

Примеры тегов: `new2023, брошенная корзина, abandoned_cart.`

Где можно посмотреть созданные теги

Когда вы сохраняете объект с назначенным тегом, сам тег также сохраняется. Отдельного окна, где можно посмотреть все теги, нет. Однако при редактировании другого объекта все ранее созданные теги будут доступны в выпадающем списке:



Хотя сохраненный тег можно открепить от объекта, полностью удалить его из платформы нельзя. Все пользователи аккаунта видят все существующие теги.

Все пользователи аккаунта видят все существующие теги.

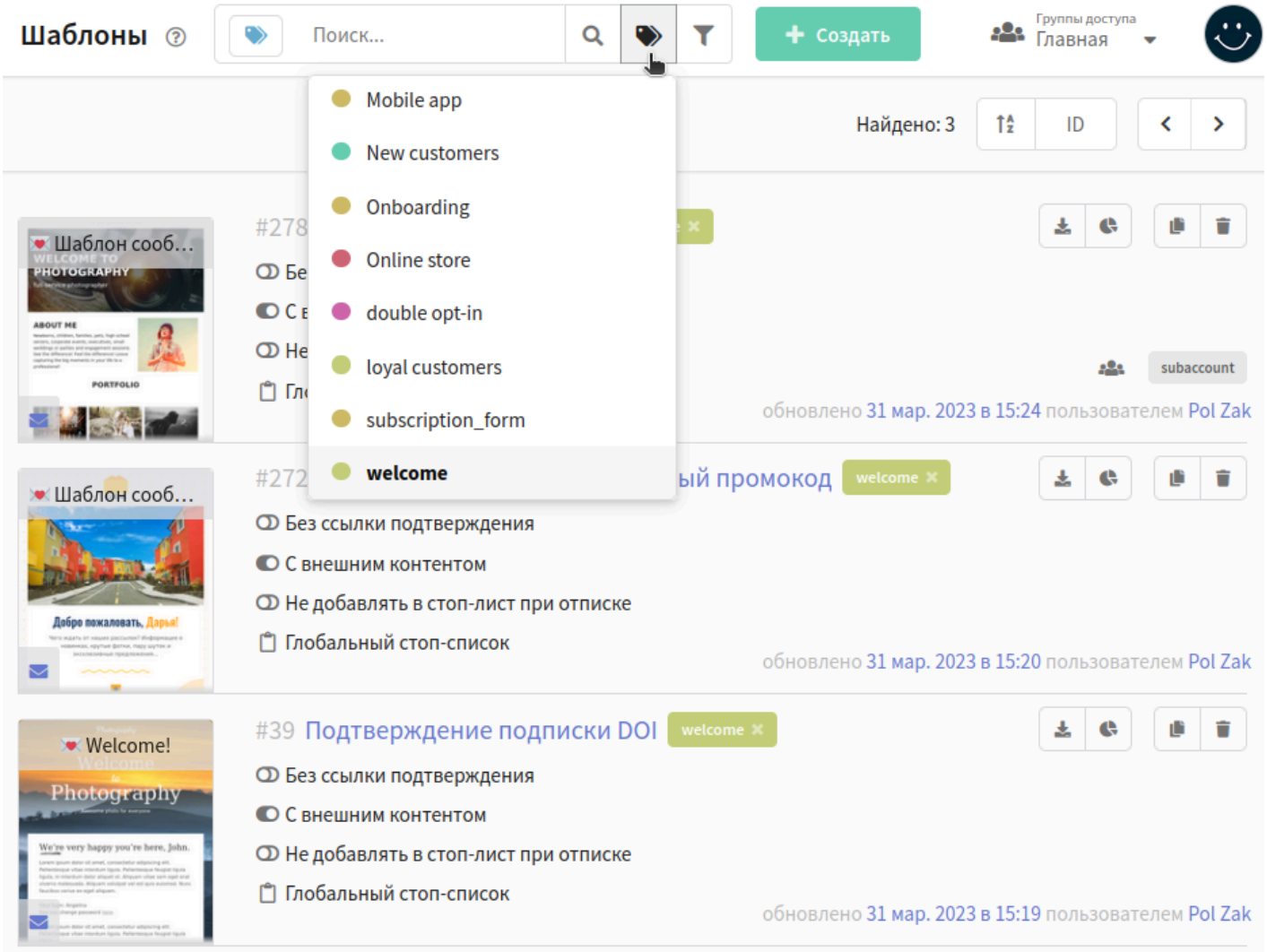
Если тег не будет прикреплен ни к одному объекту платформы, он автоматчиески удалится.

Где использовать теги

1. Поиск объектов в общем списке

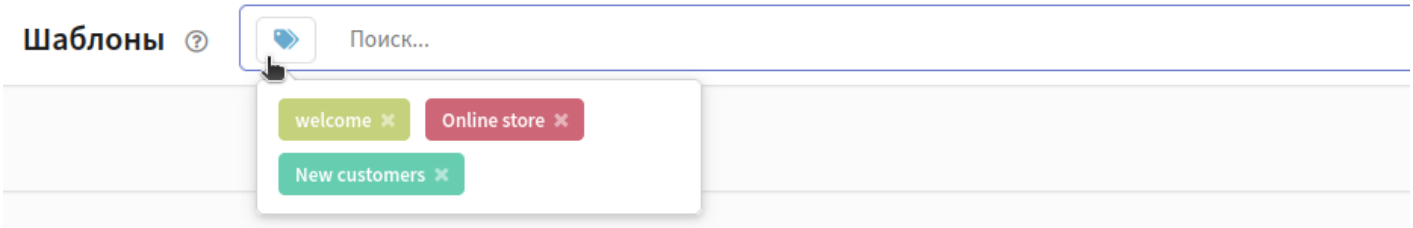
Перейдите в нужный раздел (например, в Шаблоны) и нажмите на иконку

справа от строки поиска. Выберите один или несколько тегов.



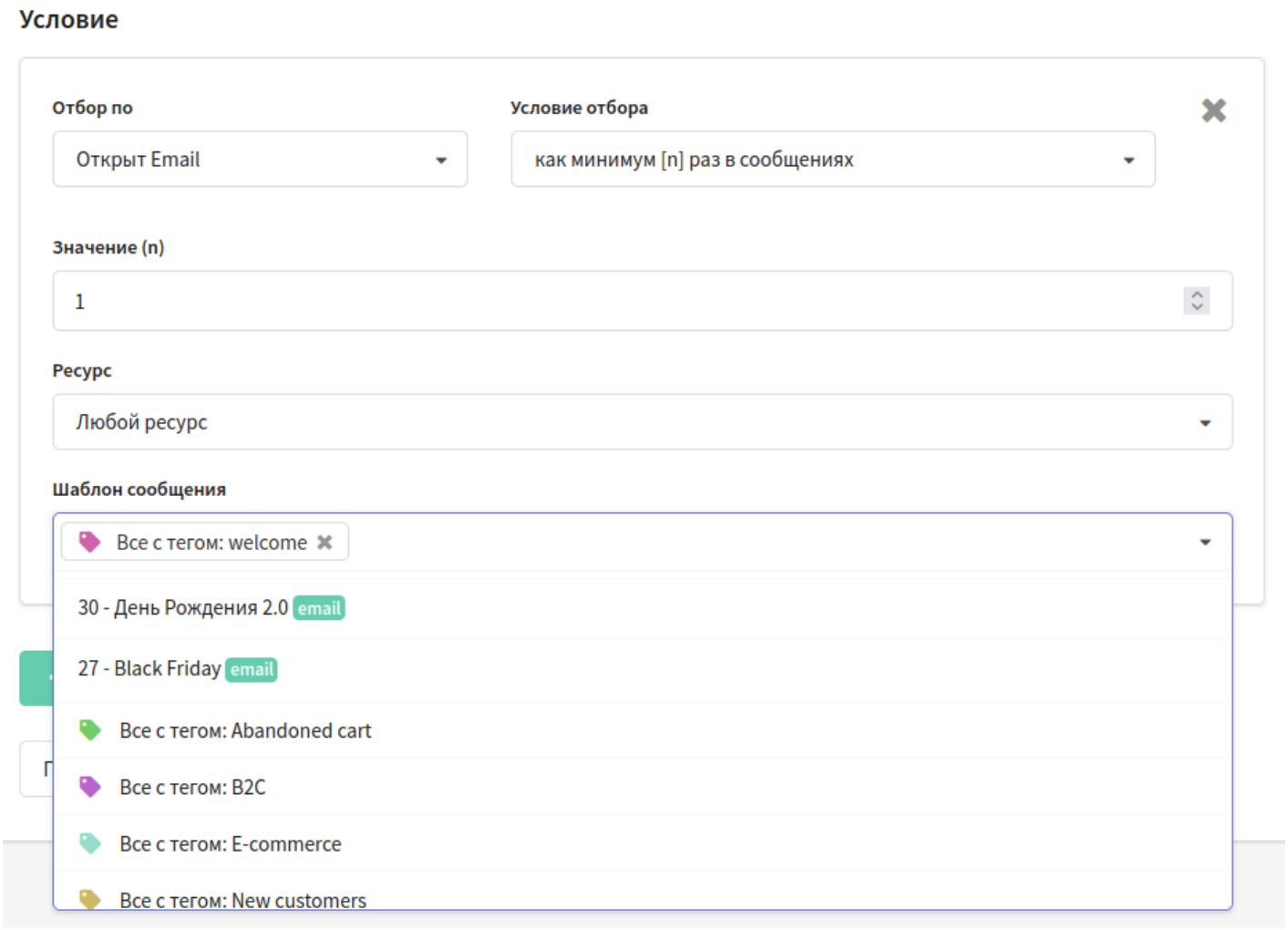
Чтобы посмотреть, какие теги добавлены в поиск, наведите курсор на

:



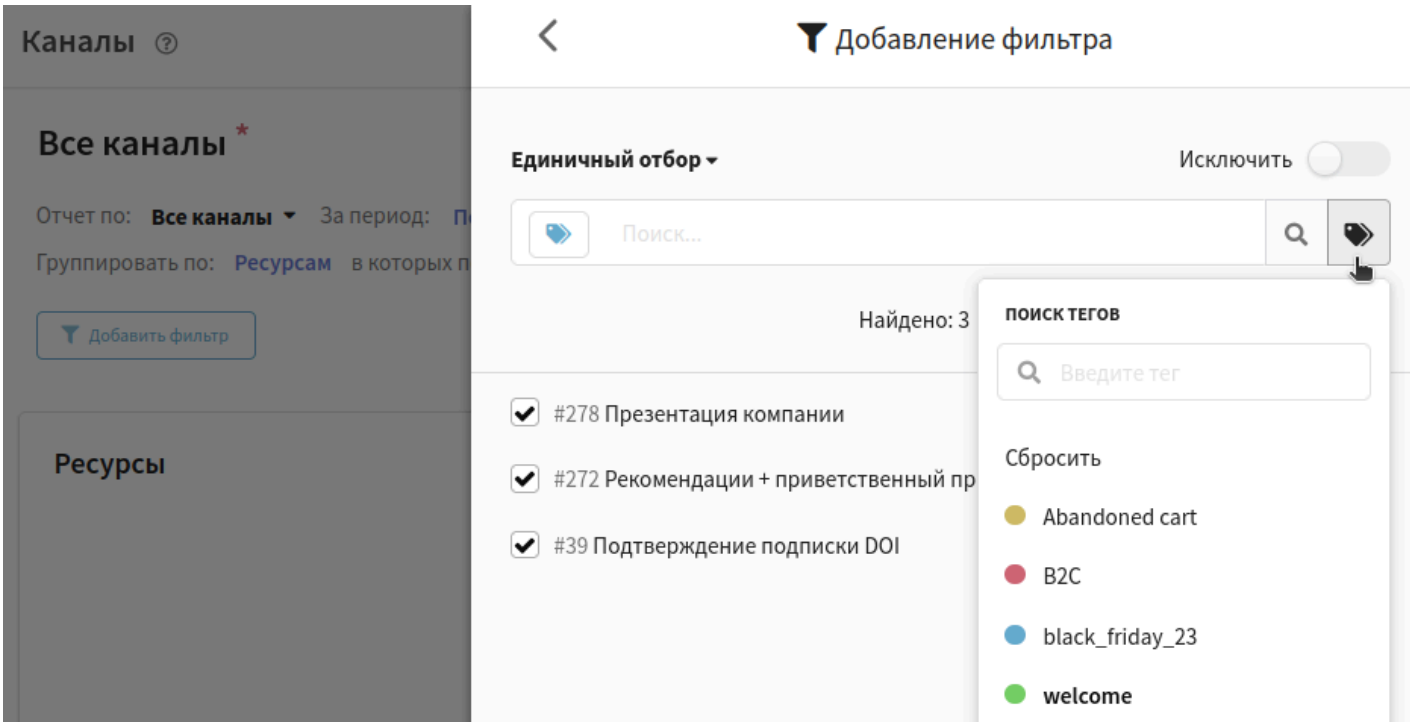
2. Сегментация профилей по событиям в рассылках или в шаблонах (отправки, доставки, открытия, клики)

При таком условии отбора вы можете выбрать все рассылки или все шаблоны с определённым тегом или тегами:

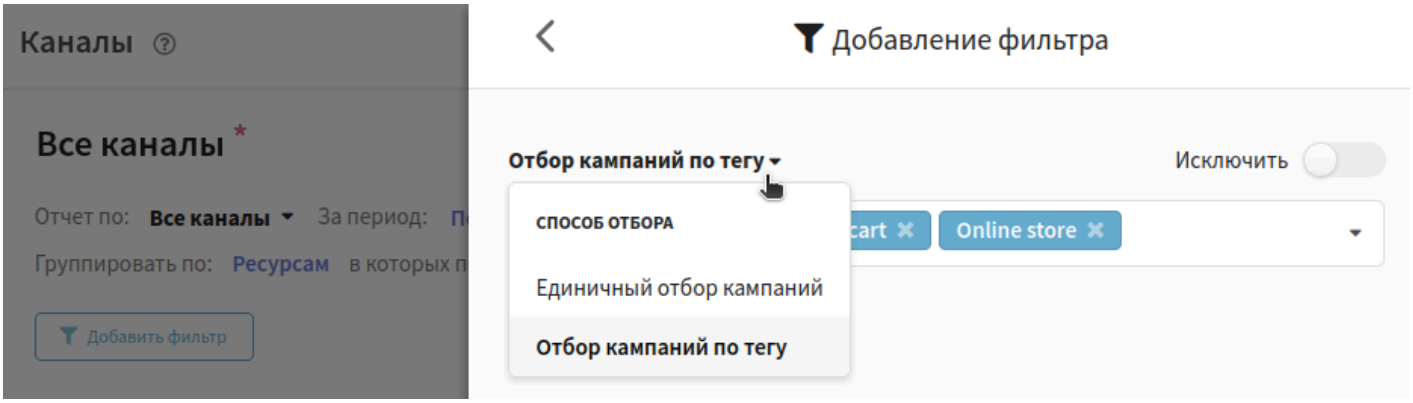


3. Аналитические отчёты

В отчете по каналам и в отчёте по трафику теги можно использовать в фильтрах. Например, для быстрого поиска нужного объекта:



Кроме того, можно сразу выбрать все объекты одного типа, которым назначен один или несколько тегов. Для этого переключите режим отбора: Единичный отбор рассылок → **Отбор рассылок по тегу**, а затем выберите теги в выпадающем списке и примените фильтр:

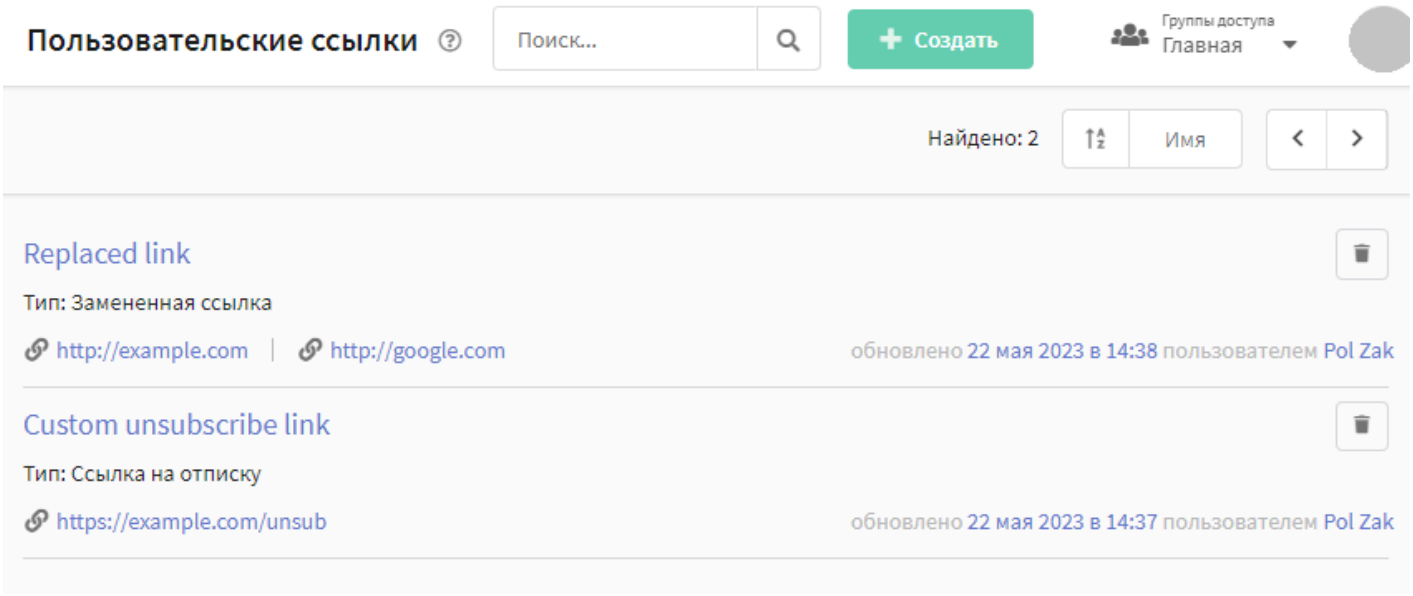


Пользовательские ссылки

Доступно два типа пользовательских ссылок: ссылка отписки и замененная ссылка. Пользовательская **ссылка отписки** нужна для того, чтобы получатели email сообщений при нажатии на кнопку “Отписаться” перенаправлялись в личный кабинет сайта или на любую веб-страницу. **Замененная ссылка** позволяет заменить адреса ссылок в уже отправленных сообщениях.

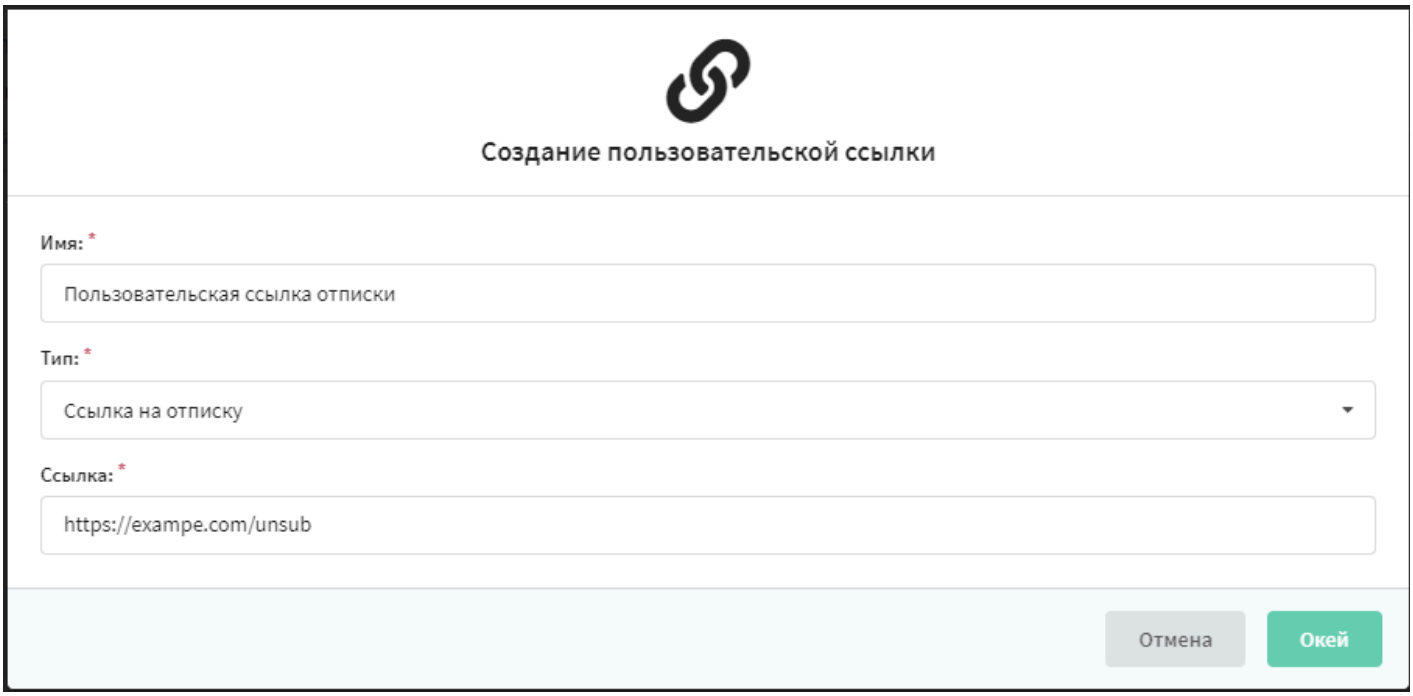
Создание ссылок отписки

В главном меню выберите **Настройки —> Пользовательские ссылки**. Для добавления новой ссылки нажмите **+ Создать**:

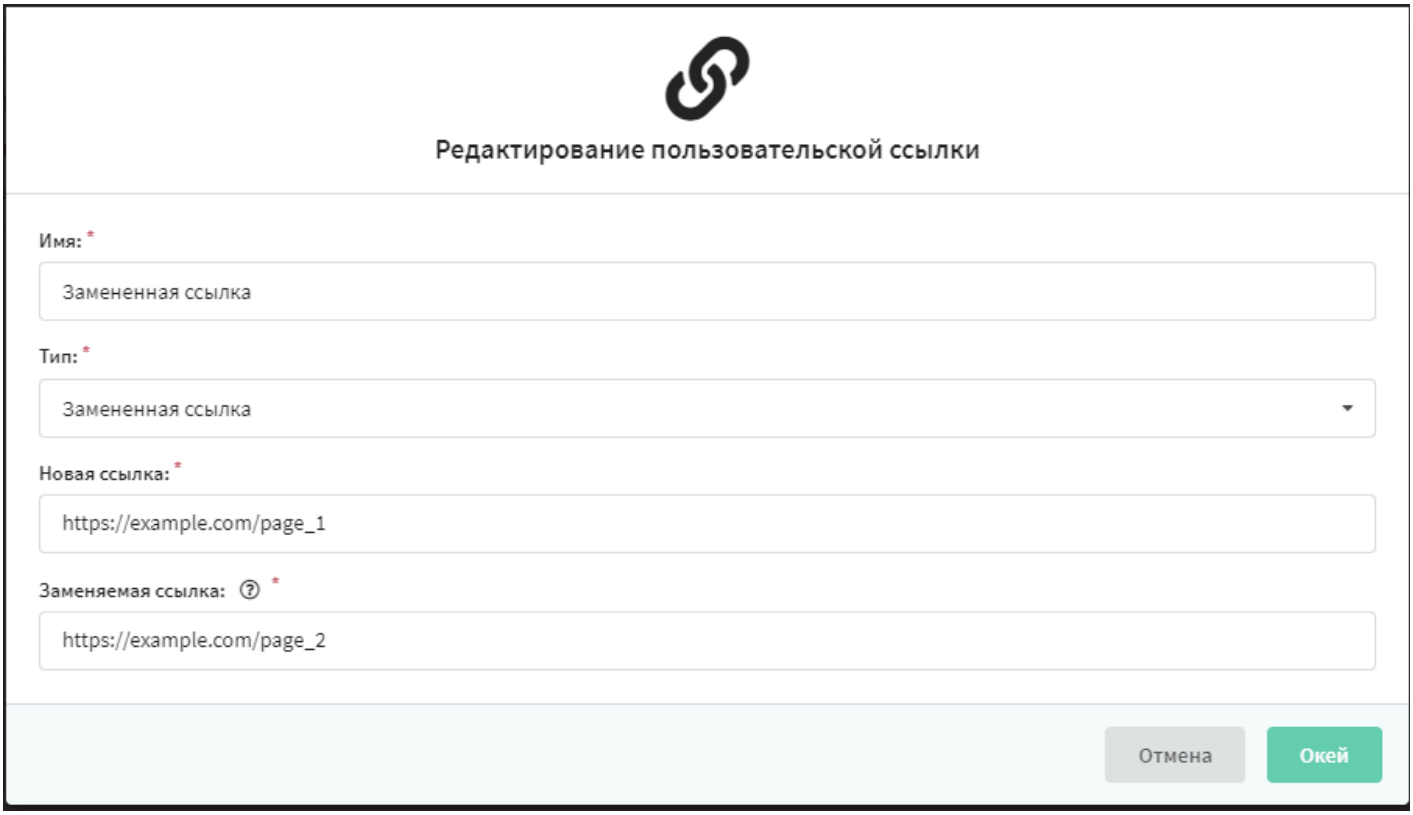


В открывшейся форме введите имя ссылки и выберите тип (ссылка отписки или замененная ссылка).

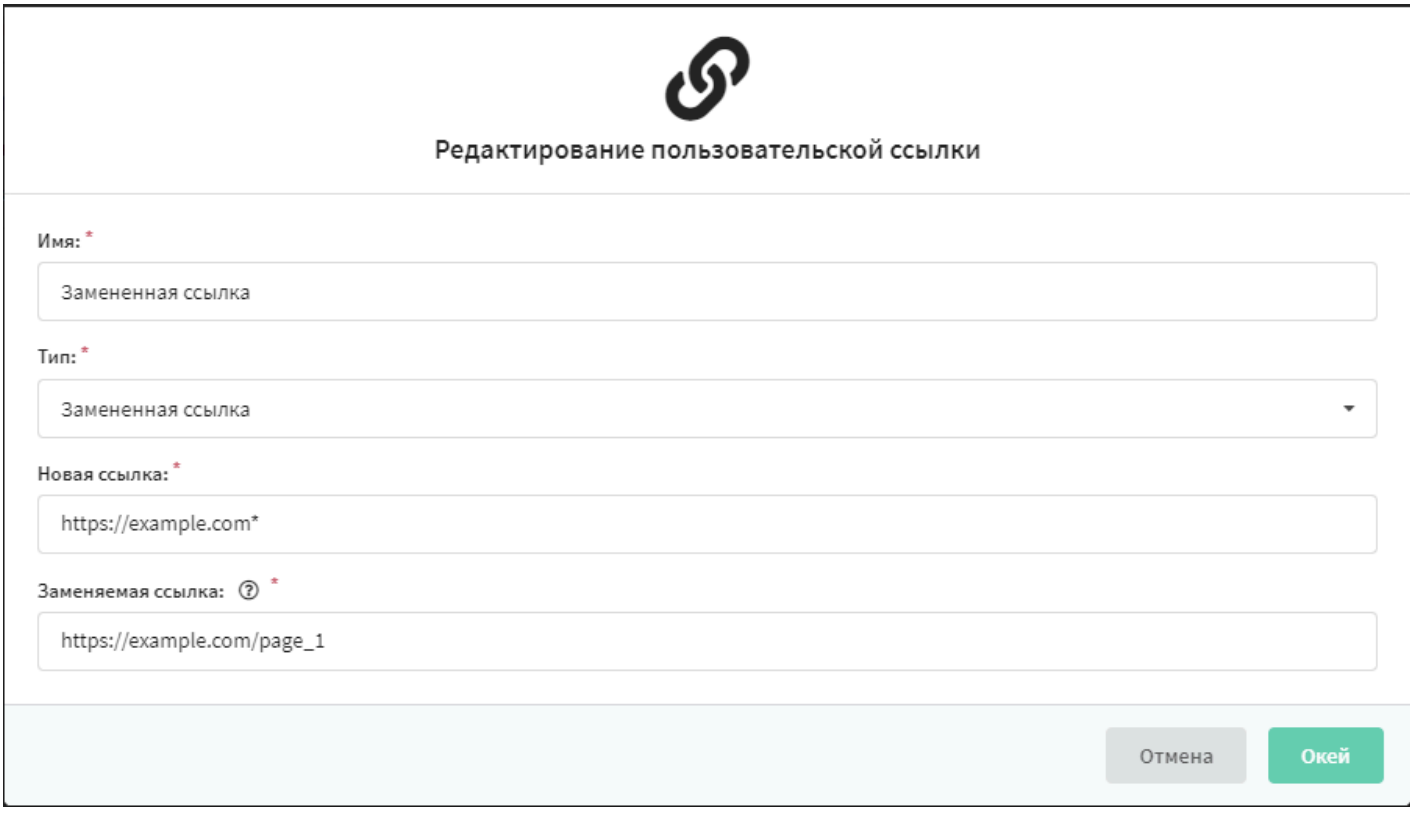
Для **ссылки отписки** укажите URL для перенаправления:



Для **замененной ссылки** в поле “Заменяемая ссылка” укажите ссылку в шаблоне, которую хотите заменить. В поле “**Новая ссылка**” введите новый URL. Ссылка в отправленных сообщениях не изменится, но при клике по ней подписчик будет перенаправлен по новому URL адресу:



Также можно заменить сразу несколько ссылок. Для этого в поле “Заменная ссылка” задайте условие поиска, используя символ wildcard (`*`). Так вы сможете найти все ссылки, адрес которых содержит определенную строку. Например, если указать `https://example.com/*`, то платформа заменит все ссылки в шаблонах, которые начинаются с `https://example.com`.



Виртуальные сендеры

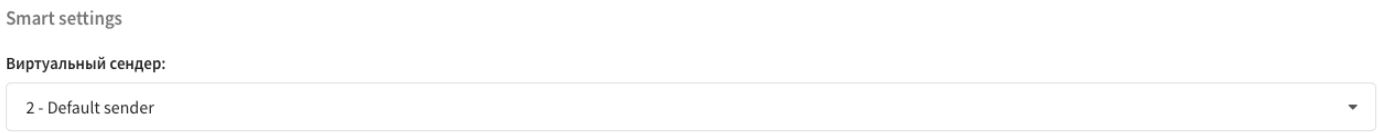
Виртуальные сендеры объединяют реальные модули отправки платформы в группы и устанавливает правила и пропорции рассылки через них. Управление учётными записями пользователей аккаунта доступно как в панели администратора Altcraft, так и в пользовательском интерфейсе. В этой статье рассмотрен функционал Пользовательского интерфейса.

	Чтобы перейти к управлению виртуальными сендерами аккаунта, в главном меню выберите **Настройки —> Виртуальные сендеры**. Нажмите **+ Создать**, чтобы добавить новую запись.

NOTE

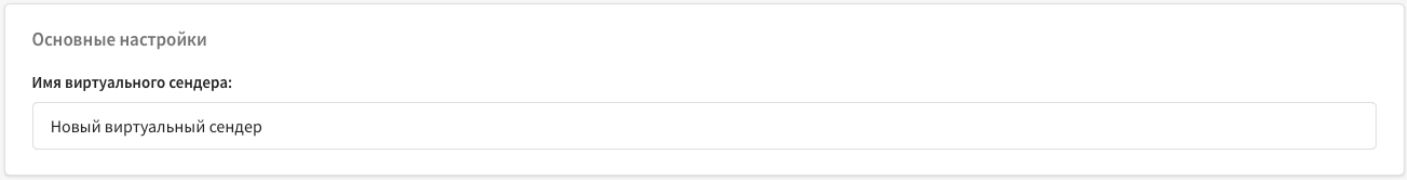
Виртуальные сендеры в Пользовательском интерфейсе могут создавать только мастер-пользователи.

Пользователи могут выбирать виртуальный сендер в Smart настройках рассылки. Для этого в аккаунте должна быть активирована опция **Smart settings**. Если Smart настройки не активированы, для рассылок будет использоваться первый подходящий Виртуальный сендер:

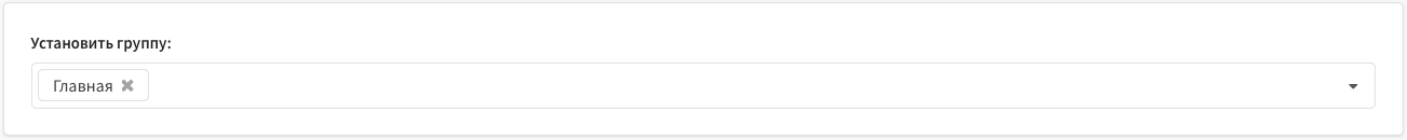


Основные настройки и группы доступа

В основных настройках для каждого виртуального сендера нужно указать уникальное имя:



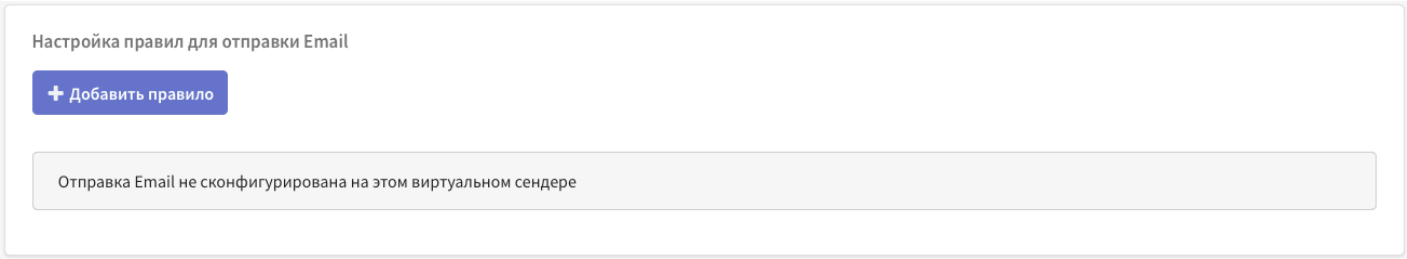
Назначить Группы объектов аккаунта можно в отдельном блоке внизу страницы:



Правила отправки для разных каналов связи

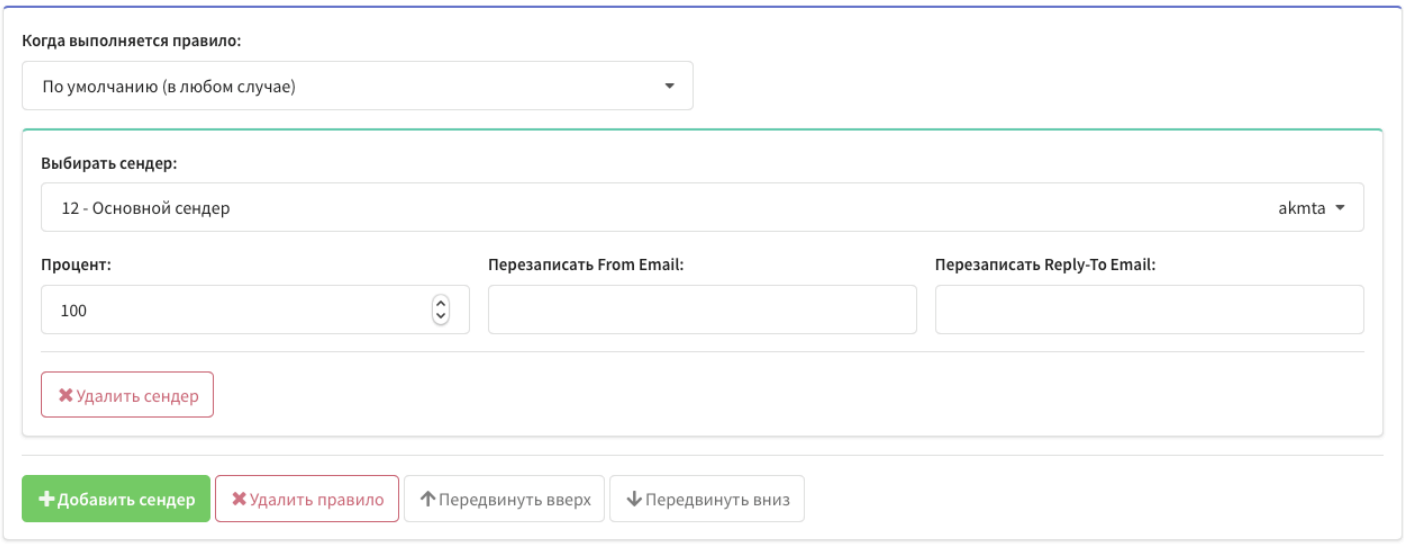
Правила отправки Email

Правила для email сендеров определяют, какие реальные отправщики и в каких пропорциях будут использоваться при тех или иных условиях. Чтобы добавить правило, нажмите **+ Добавить правило**.



Выберите подходящее условие в выпадающем меню **Когда правило применяется:**

- **По умолчанию (в любом случае)** — правило применяется при любых отправках.
- **Group соответствует значению** — правило применяется к email-рассылкам, входящим в определённую Группу объектов аккаунта.
- **From domain соответствует значению** — правило применяется при отправках с определённого домена.
- **To domain соответствует значению** — правило применяется при отправках на определённые домены email.
- **From email соответствует значению** — правило применяется при отправках с определённого email-адреса.



Приоритет применения правил устанавливается их положением на странице: сверху вниз. Правило **По умолчанию (в любом случае)** обязательно должно быть последним в списке. Приоритет остальных правил устанавливается кнопками **Передвинуть вверх** и **Передвинуть вниз**.

CAUTION

Правило по умолчанию **По умолчанию (в любом случае)** невозможно удалить или переместить вверх по приоритету, так как оно обрабатывает все случаи, не попавшие под остальные правила.

Кнопками **Добавить сендер** и **Удалить сендер** вы можете добавлять и удалять сендеры в рамках правила отправки. Для каждого сендера настраиваются следующие поля:

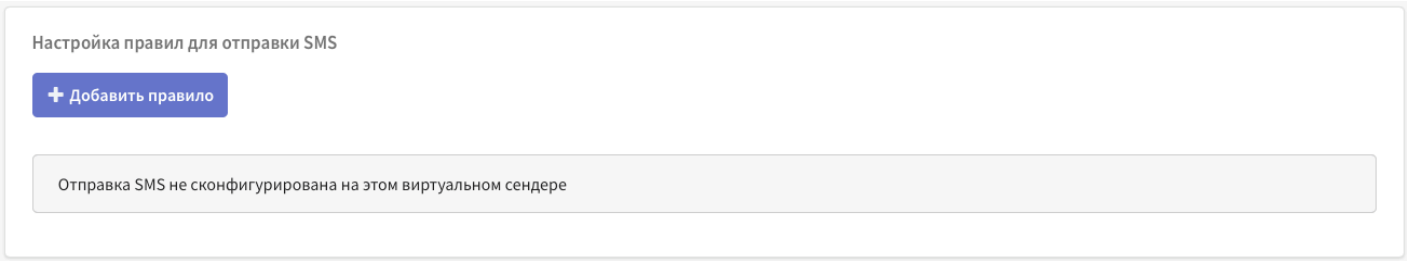
- **Выбирать сендер** — выберите реальный сендер для отправки.
- **Процент** — установите для сендера процент отправляемых сообщений в рамках правила.
- **Перезаписать From Email** — переопределить значение From email для отправляемых сообщений.
- **Перезаписать Reply-to Email** — переопределить значение Reply-to email для отправляемых сообщений.

NOTE

Сумма процентов для сендеров в рамках правила должна быть 100.

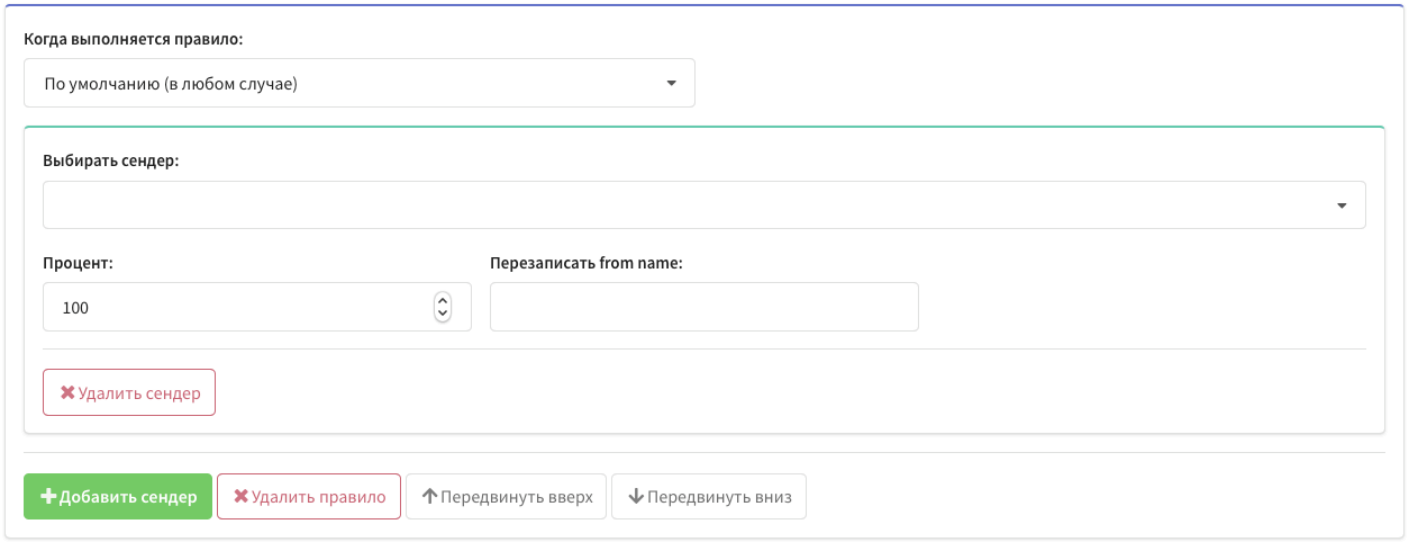
Правила отправки SMS

Правила для SMS сендеров определяют, какие реальные отправщики и в каких пропорциях будут использоваться при тех или иных условиях. Чтобы добавить правило, нажмите **+ Добавить правило**.



Выберите подходящее условие в выпадающем меню **Когда выполняется правило:**

- **Group соответствует значению** — правило применяется к SMS-рассылкам, входящим в определённую Группу объектов аккаунта.
- **По умолчанию (в любом случае)** — правило применяется при любых отправках.



Приоритет применения правил устанавливается их положением на странице: сверху вниз. Правило **По умолчанию (в любом случае)** обязательно должно быть последним в списке. Приоритет остальных правил устанавливается кнопками **Передвинуть вверх** и **Передвинуть вниз**.

CAUTION

Правило по умолчанию **По умолчанию (в любом случае)** невозможно удалить или переместить вверх по приоритету, так как оно обрабатывает все случаи, не попавшие под остальные правила.

Кнопками **Добавить сендер** и **Удалить сендер** вы можете добавлять и удалять сендеры в рамках правила отправки. Для каждого сендера настраиваются следующие поля:

- **Выбирать сендер** — выберите реальный сендер для отправки.

- **Процент** — установите для сендера процент отправляемых сообщений в рамках правила.
- **Перезаписать from name** — переопределить значение `Caller_ID` для отправляемых сообщений.

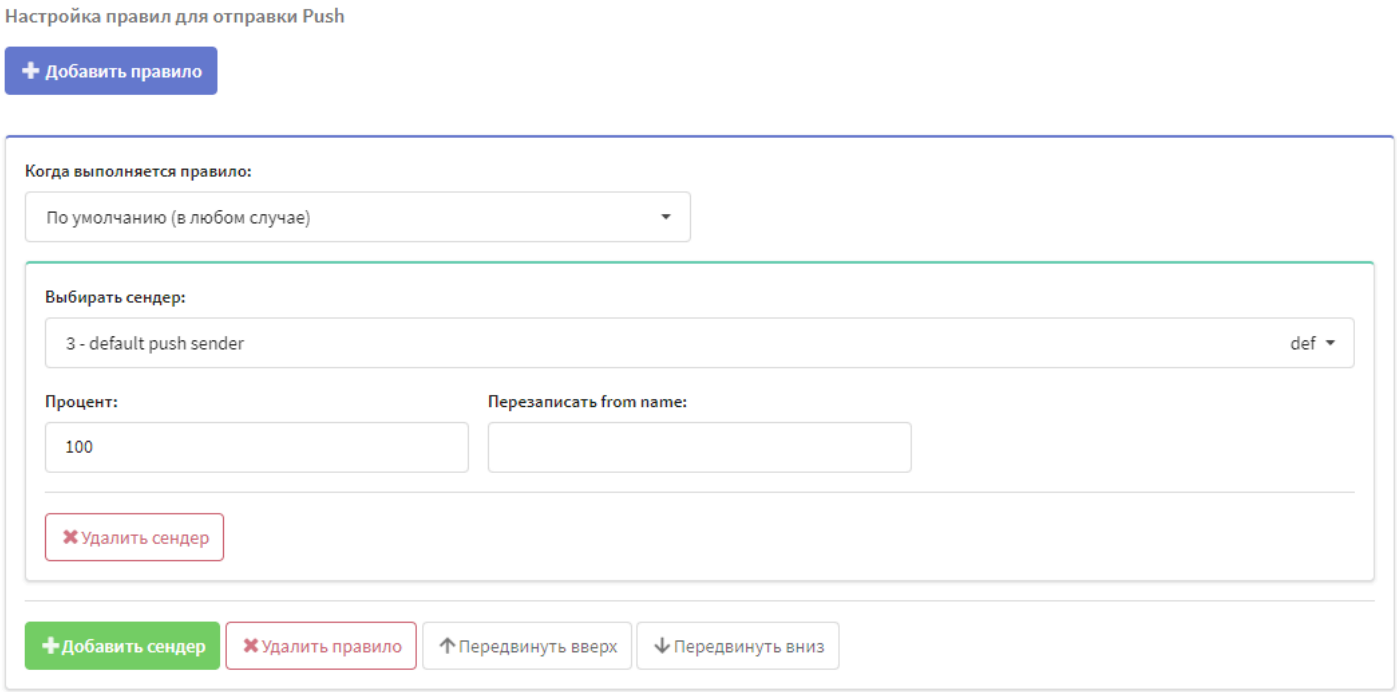
> **NOTE**
>
> Сумма процентов для сендеров в рамках правила должна быть 100.

Правила отправки Push

Правила для Push сендеров определяют, какие реальные отправщики и в каких пропорциях будут использоваться при тех или иных условиях. Чтобы добавить правило, нажмите **+ Добавить правило**.

Выберите подходящее условие в выпадающем меню **Когда выполняется правило:**

- **Group соответствует значению** — правило применяется к Push-рассылкам, входящим в определённую Группу объектов аккаунта.
- **По умолчанию (в любом случае)** — правило применяется при любых отправках.



Приоритет применения правил устанавливается их положением на странице: сверху вниз. Правило **По умолчанию (в любом случае)** обязательно должно быть последним в списке. Приоритет остальных правил устанавливается кнопками **Передвинуть вверх** и **Передвинуть вниз**.

> **CAUTION**

Правило по умолчанию **По умолчанию (в любом случае)** невозможно удалить или переместить вверх по приоритету, так как оно обрабатывает все случаи, не попавшие под остальные правила.

Кнопками **Добавить сендер** и **Удалить сендер** вы можете добавлять и удалять сендеры в рамках правила отправки. Для каждого сендера настраиваются следующие поля:

- **Выбирать сендер** — выберите реальный сендер для отправки.
- **Процент** — установите для сендера процент отправляемых сообщений в рамках правила.

NOTE

Сумма процентов для сендеров в рамках правила должна быть 100.

Политики отправки

Политики отправки ограничивают количество отправляемых одному получателю сообщений. Политику отправки можно назначить:

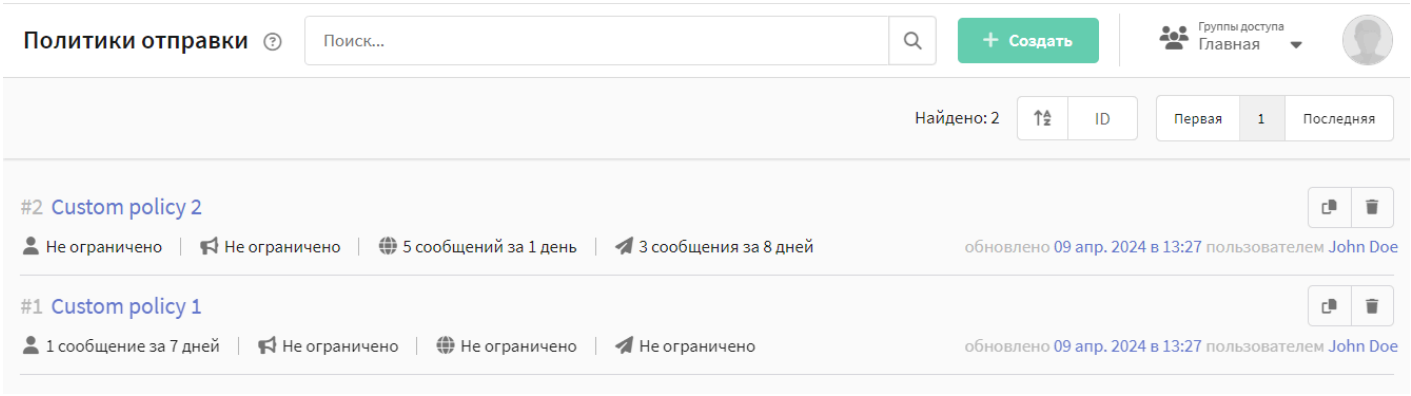
- **В аккаунте** — политика будет использоваться глобально.
- **В ресурсе** — политика будет применяться ко всем рассылкам ресурса, переопределяя настройки аккаунта.
- **В рассылке** — политика будет применяться к выбранной рассылке, переопределяя настройки ресурса.

> **CAUTION**
>
> Политику отправки нельзя установить для конкретного профиля. Политика настраивается и действует для всех профилей в рамках аккаунта/ресурса/рассылки.

Управление политиками отправки

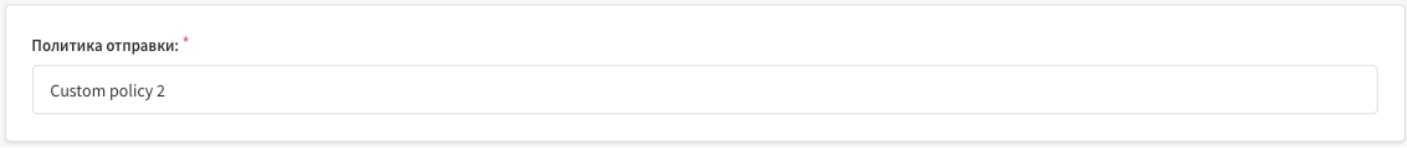
> **CAUTION**
>
> Политики отправки, как и другой функционал меню **Настройки**, доступен только **мастер-пользователям** аккаунта.

Чтобы создать, редактировать или удалить политику отправки, в главном меню выберите **Настройки —> Политики отправки**. В открывшемся списке отображаются существующие политики отправки:



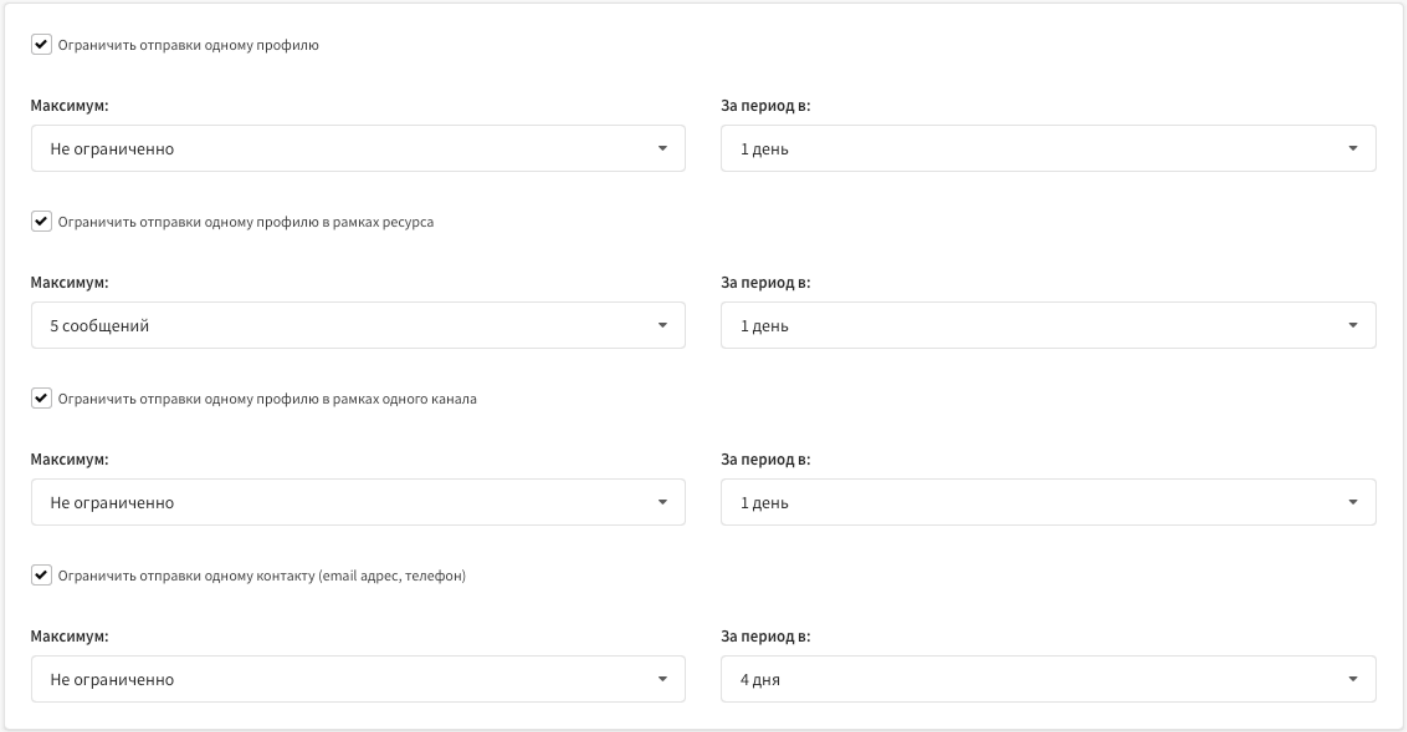
Политику можно клонировать или удалить из контекстного меню в правой части плитки. Перейти к редактированию можно нажав на название политики.

Чтобы создать новую политику отправки, нажмите **+ Создать** в верхней части страницы. Для каждой политики есть три блока настроек. В первом введите уникальное имя:



Второй блок содержит ограничения отправки. Для каждого типа ограничения установите максимальное количество сообщений (**от 1 до 15**) и временной период (**от 1 до 16 дней**):

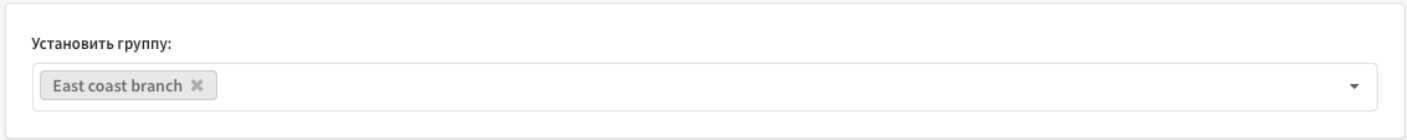
- **Ограничить отправки одному профилю** — будут учитываться отправки по любым контактам и подпискам в рамках одного профиля подписчика.
- **Ограничить отправки одному профилю в рамках ресурса** — будут учитываться все подписки **одного профиля** на определённый Ресурс.
- **Ограничить отправки одному профилю в рамках одного канала** — будут учитываться все подписки на все ресурсы **одного профиля** по одному каналу коммуникации (email, sms, push)
- **Ограничить отправки одному контакту (email адрес, телефон)** — будут учитываться все подписки, содержащие одинаковый контакт. При этом подписки могут относиться к **разным профилям**.



TIP

Политика, где не выбрано ни одно ограничение, является **безлимитной**. Если вам нужно установить глобальное ограничение в рамках аккаунта и разрешить ресурсу или рассылке неограниченную отправку — создайте безлимитную политику, чтобы переопределить глобальные настройки.

В третьем блоке настроек установите при необходимости группы доступа, в ресурсах и рассылках которых будет доступна для выбора эта политика отправки.

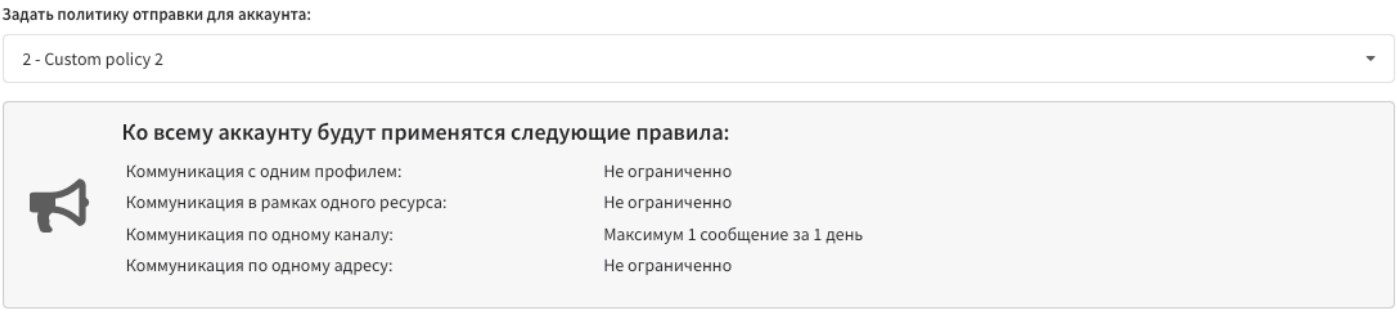


Назначение политики отправки

Политику отправки можно назначить глобально для всего аккаунта, в рамках одного ресурса или для одной конкретной рассылки.

В рамках всего аккаунта

Чтобы установить глобальную политику, в главном меню откройте **Настройки —> Настройки аккаунта** и выберите созданную ранее политику в этом выпадающем меню:

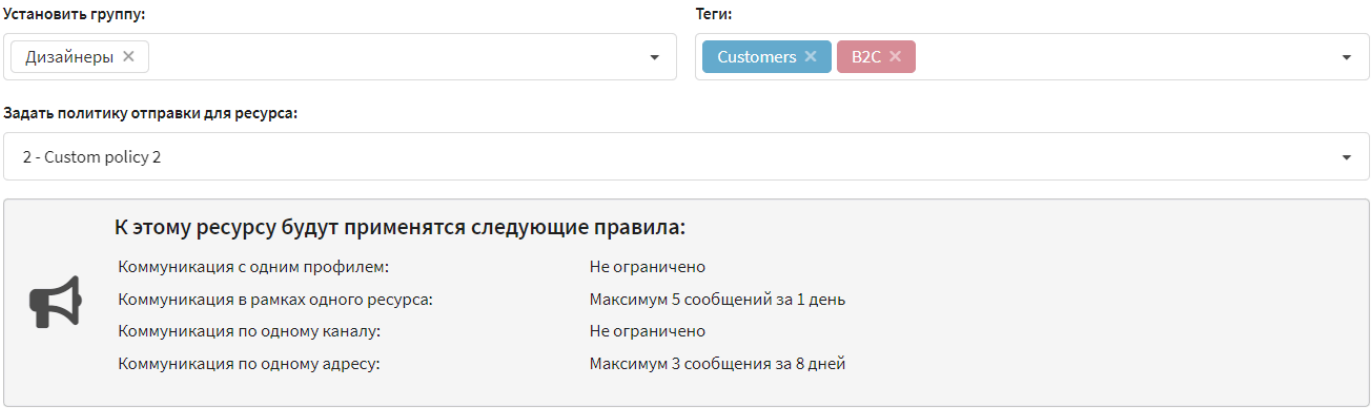


CAUTION

Глобальную политику отправки для аккаунта может установить только **мастер-пользователь**.

В рамках одного ресурса

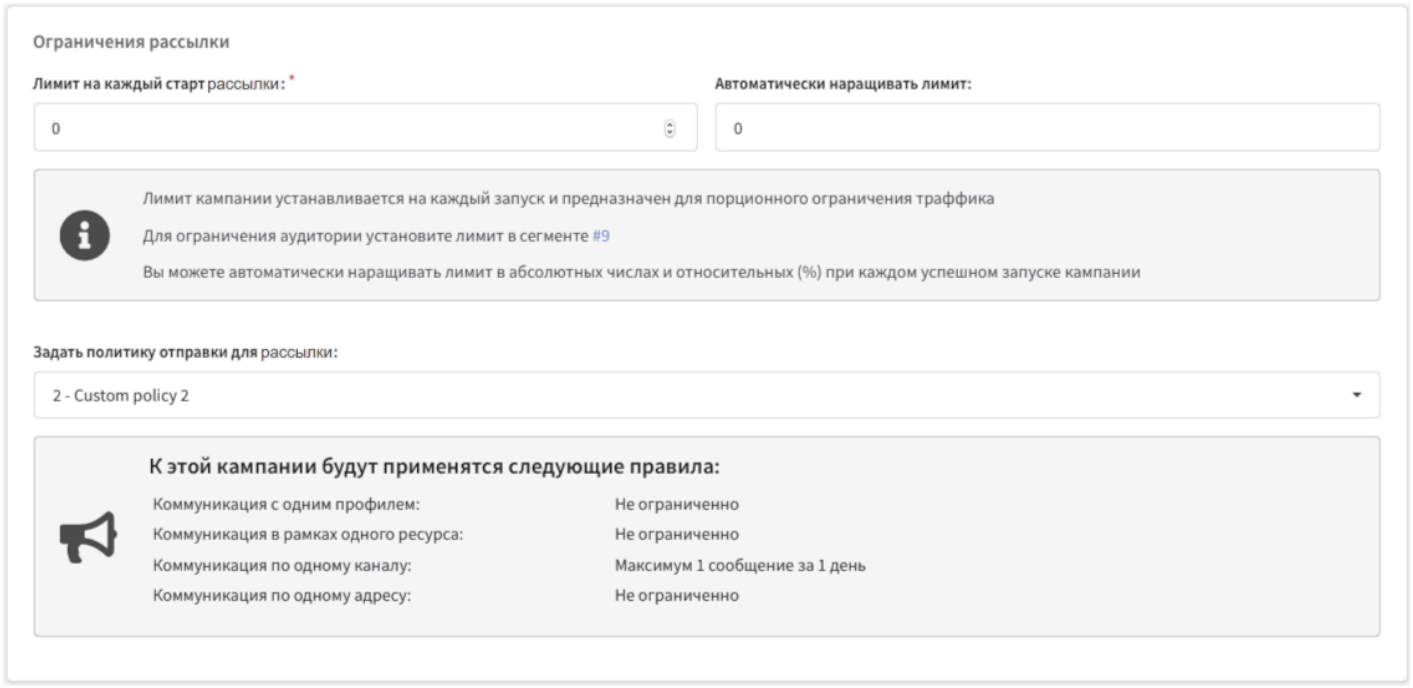
Политика отправки ресурса переопределит глобальные настройки. В главном меню откройте страницу **Ресурсы** и в **Основных настройках** нужного ресурса выберите созданную ранее политику отправки:



В рамках одной рассылки

Политика отправки, которую вы используете в рассылке, переопределит как глобальные настройки, так и политику ресурса. Откройте нужную рассылку и в блоке **Ограничения рассылки** выберите созданную ранее политику отправки.

Обратите внимание, ограничения политики распространяться на профиль и его контакты, т.е. выбранная рассылка не сможет отправить одному профилю больше сообщений, чем указано в политике.



INFO

Сообщения, попавшие под ограничения, будут формировать в журнале рассылки записи типа **Отправка запрещена политикой**.

Журнал аудита

Журнал аудита – это важный элемент безопасности и мониторинга, регистрирующий действия пользователей в платформе. Доступ к аудиту есть только у пользователей с правами **мастера**. Чтобы перейти к аудиту, откройте **Настройки** → **Журнал аудита**.

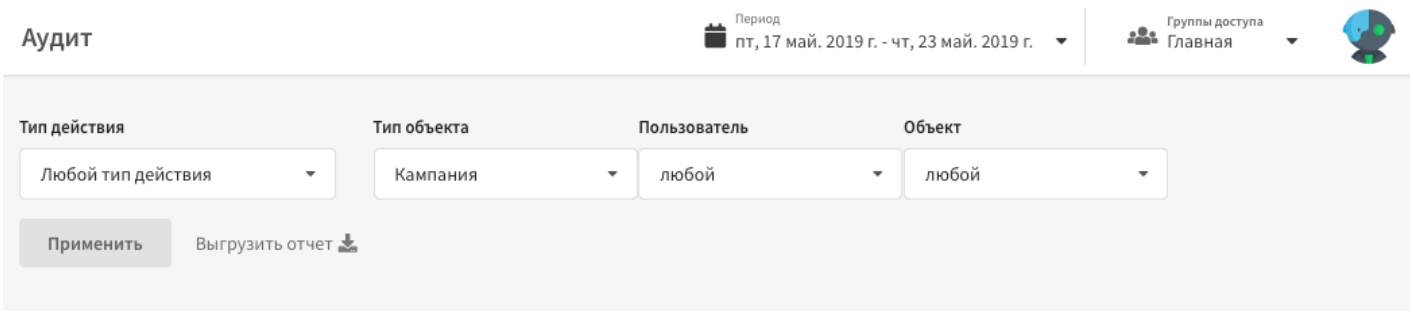
Лог аудита хранится в течение 90 дней. Чтобы изменить это значение, в файл конфигурации `main.json` необходимо добавить запись `"AUDIT_LOG_TTL" : 90`, изменив значение на необходимое.

Настройки журнала аудита

Вы можете настроить запрос аудита под текущие задачи:

- **Период** — временные рамки для отображения зарегистрированных действий.
- **Тип действия** — тип события, зарегистрированного системой. Действия связаны с авторизацией в платформе, созданием, редактированием и активацией объектов в аккаунте.
- **Тип объекта** — позволяет просмотреть действия, связанные с определённым типом сущностей в Altcraft. Это могут быть пользователи, группы и роли доступа, базы данных и сегменты, ресурсы и рассылки, кампании, шаблоны и их части, а также другие объекты.
- **Объект** — вы можете выбрать конкретный объект аккаунта и получить информацию по действиям с ним.

Чтобы просмотреть отчёт аудита, нажмите **Применить**. Чтобы выгрузить отчёт аудита в .XLS или .CSV, нажмите **Выгрузить отчёт:**

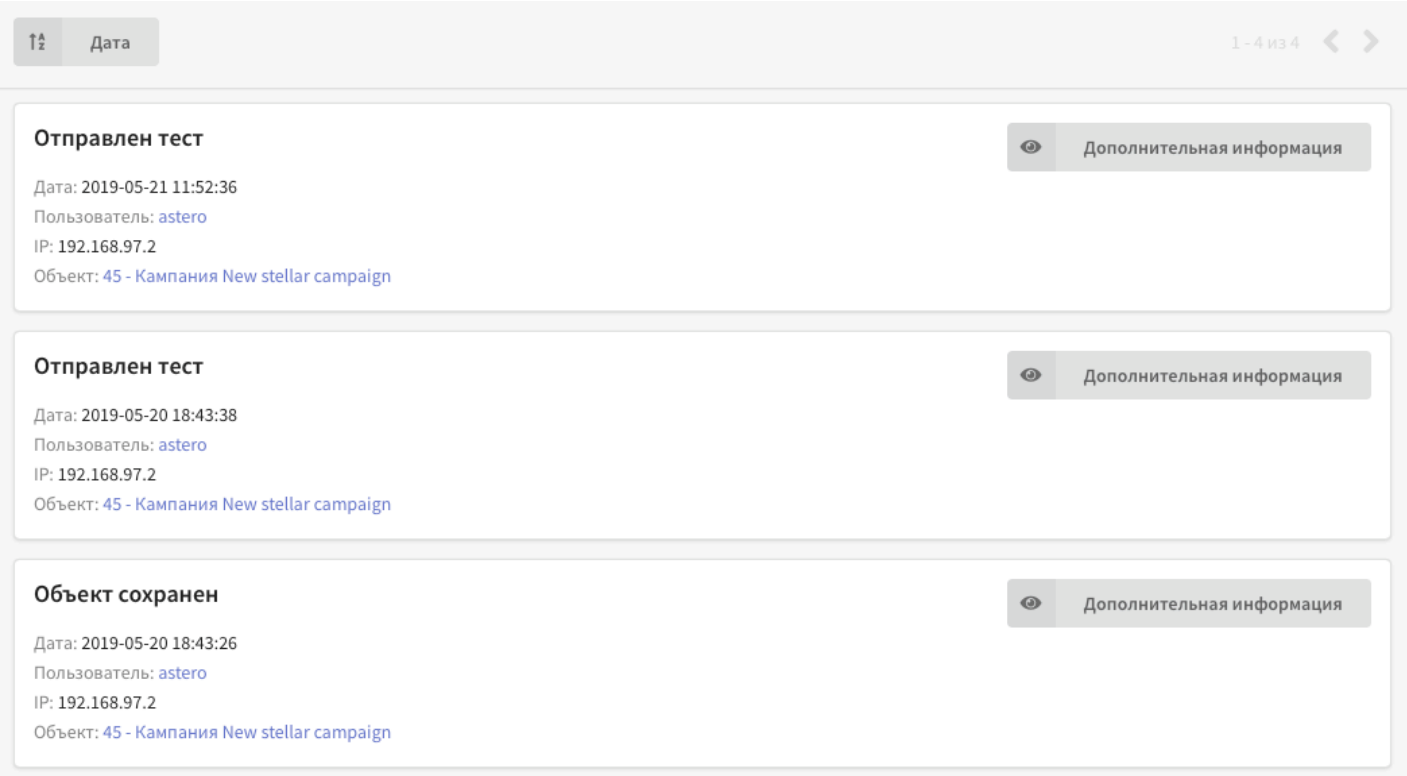


Просмотр событий аудита

Действия, подходящие под условия запроса, отображаются списком плиток. Заголовок каждой плитки указывает на тип зарегистрированного действия. Для действия отображается дата, пользователь, совершивший действие, его IP адрес и объект, с которым действие связано.

Дополнительная информация содержит более подробное описание события, например, email адреса, на которые был отправлен тест.

Действия в отчёте можно отсортировать по дате, пользователю, IP адресу или объекту.



Действия с кампаниями

Система регистрирует следующие действия с кампаниями (объединениями рассылок и сценариев):

Основные операции с кампаниями

- **Создание кампании** — создание новой кампании
- **Обновление кампании** — изменение настроек кампании
- **Удаление кампании** — удаление кампании
- **Активация кампании** — запуск кампании
- **Деактивация кампании** — остановка кампании
- **Завершение кампании** — окончательное завершение работы кампании

Управление локальными контрольными группами (ЛКГ)

- **Создание ЛКГ** — настройка локальной контрольной группы
- **Обновление ЛКГ** — изменение параметров ЛКГ
- **Удаление ЛКГ** — удаление локальной контрольной группы

Работа с маркетинговыми активностями

- **Привязка активности** — добавление рассылки или сценария в кампанию
- **Отвязка активности** — удаление рассылки или сценария из кампании

Работа с сегментами

- **Привязка сегмента** — назначение сегмента для кампании
- **Отвязка сегмента** — удаление сегмента из кампании

Работа с программами лояльности

- **Создание программы лояльности** — создание новой программы лояльности
- **Обновление программы лояльности** — изменение настроек программы лояльности
- **Удаление программы лояльности** — удаление программы лояльности
- **Активация программы лояльности** — запуск программы лояльности
- **Деактивация программы лояльности** — остановка программы лояльности

Все вышеописанные действия доступны также для **типов баллов** и **уровней лояльности**.

Действия с партнерами ПЛ:

- **Создание партнера в программе лояльности** — создание нового партнера программы лояльности
- **Обновление партнера в программе лояльности** — изменение партнера программы лояльности
- **Удаление партнера в программе лояльности** — удаление партнера программы лояльности

Действия с промоакциями:

- **Создание промоакции** — создание новой промоакции
- **Обновление промоакции** — изменение настроек промоакции
- **Удаление промоакции** — удаление промоакции
- **Активация промоакции** — запуск промоакции
- **Деактивация промоакции** — остановка промоакции
- **Завершение промоакции** — окончательное завершение работы промоакции

Также журнал аудита хранит информацию о прикреплении и откреплении программы лояльности к профилю подписчика.

TIP

При работе с рассылками и сценариями внутри кампаний система также регистрирует стандартные действия (удаление, деактивация, завершение) с указанием идентификатора родительской кампании.

Пользователи и разграничение доступа

В Altcraft можно ограничить функционал учётных записей **пользователей**, а также **токенов**, использующихся в **API** платформы для авторизации. Доступ определяется группами и ролями пользователя или токена:

- **Группы** определяют видимость таких объектов, как базы данных, шаблоны, кампании и сценарии и др.
- **Роли** определяют набор доступных действий с видимыми объектами.

CAUTION

Управление пользователями аккаунта и разграничением доступа доступно только для пользователей с правами **мастера**.

Группы доступа

С помощью **Групп** в аккаунте Альткрафт можно ограничивать доступ пользователей к тем или иным объектам аккаунта, например, к шаблонам, кампаниям, базам данных и сегментам. Группу можно представить как контейнер для объектов. Таким образом пользователи будут работать внутри своих контейнеров и не смогут случайно или намеренно повлиять на объекты в других группах.

Использовать группы удобно, когда вам нужно в рамках одного аккаунта с общей статистикой организовать работу нескольких подразделений организации, или создать условия для работы сторонних специалистов, например — верстальщиков для работы с шаблонами или разработчиков сайта/приложения для отладки интеграции.

INFO

Если вам нужно сильнее разделить зоны ответственности нескольких департаментов компании, используйте несколько аккаунтов платформы. Аккаунты могут использовать общие или выделенные ресурсы и компоненты. На все аккаунты одной платформы создаётся общий стоп-список email.

Главная группа

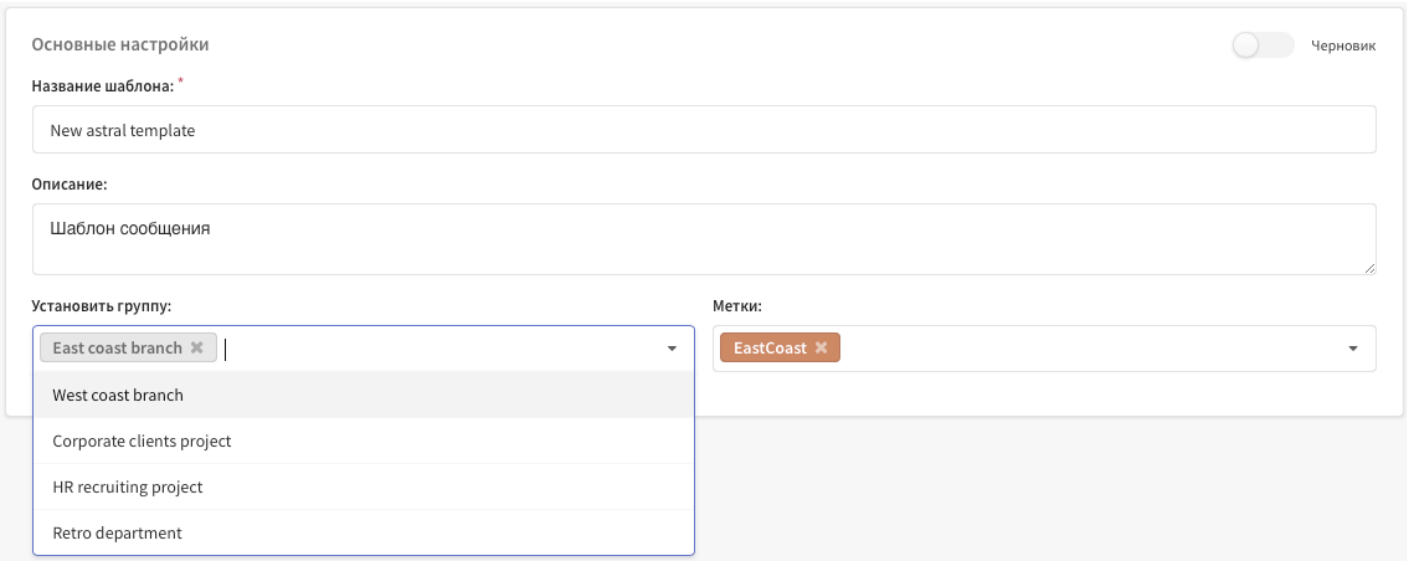
Главная группа создаётся системно и содержит все объекты аккаунта, в том числе объекты, входящие в другие группы. Если вы удаляете пользовательскую группу, объекты в ней не удаляются, но переходят в Главную группу. В Главной группе можно работать так же, как и в любой другой, если разделение по группам не требуется.

CAUTION

Мастер-доступ подразумевает полный доступ к Главной группе аккаунта вне зависимости от назначенных дополнительно групп и ролей. Единственное исключение — **доступ к удалению профилей**. Роль на удаление профилей назначается **отдельно**.

Создать / назначить группу

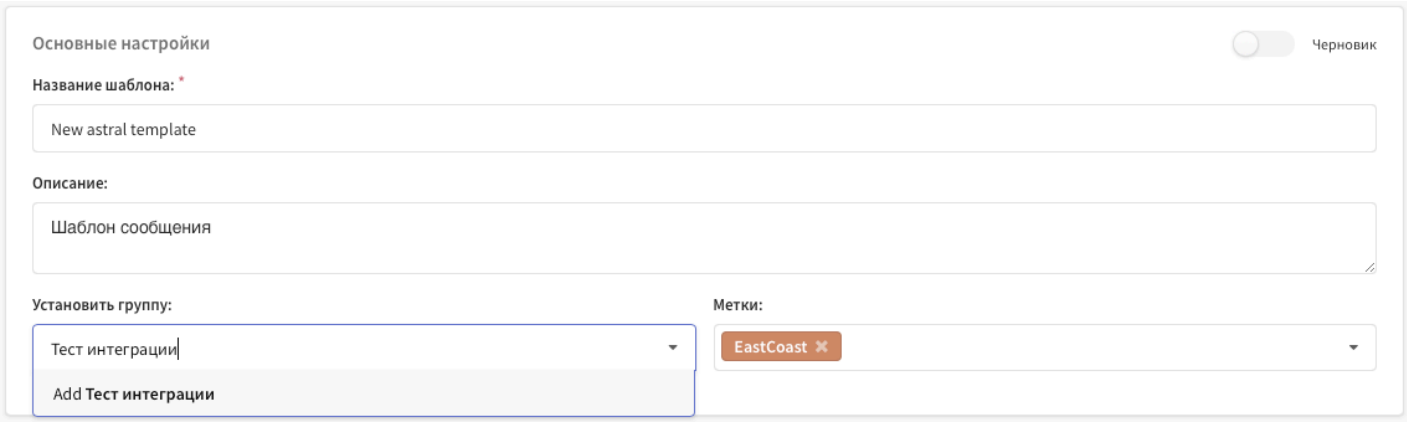
Группы назначаются и создаются динамически при редактировании объектов аккаунта — в блоке **Основные настройки**. Чтобы назначить существующую Группу, в поле **Установить группу** выберите её из выпадающего списка. Например, для Шаблона сообщения:



CAUTION

Создавать и устанавливать группы может только Мастер-пользователь.

Чтобы создать новую группу, в этом же поле для любого объекта аккаунта введите её название и нажмите Enter на клавиатуре или **Add [имя_группы]** в выпадающем списке:



CAUTION

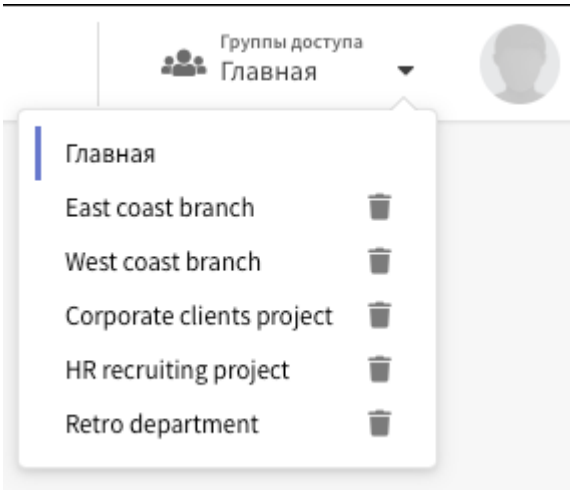
Каждому объекту, который создаёт пользователь, присваивается группа, в которой он находился в этот момент.

Переключить / удалить группу

Группы, доступные пользователю, отображаются в меню в верхнем правом углу приложения, рядом с аватаром учётной записи. Переключиться с группы на группу можно из любого меню аккаунта.

NOTE

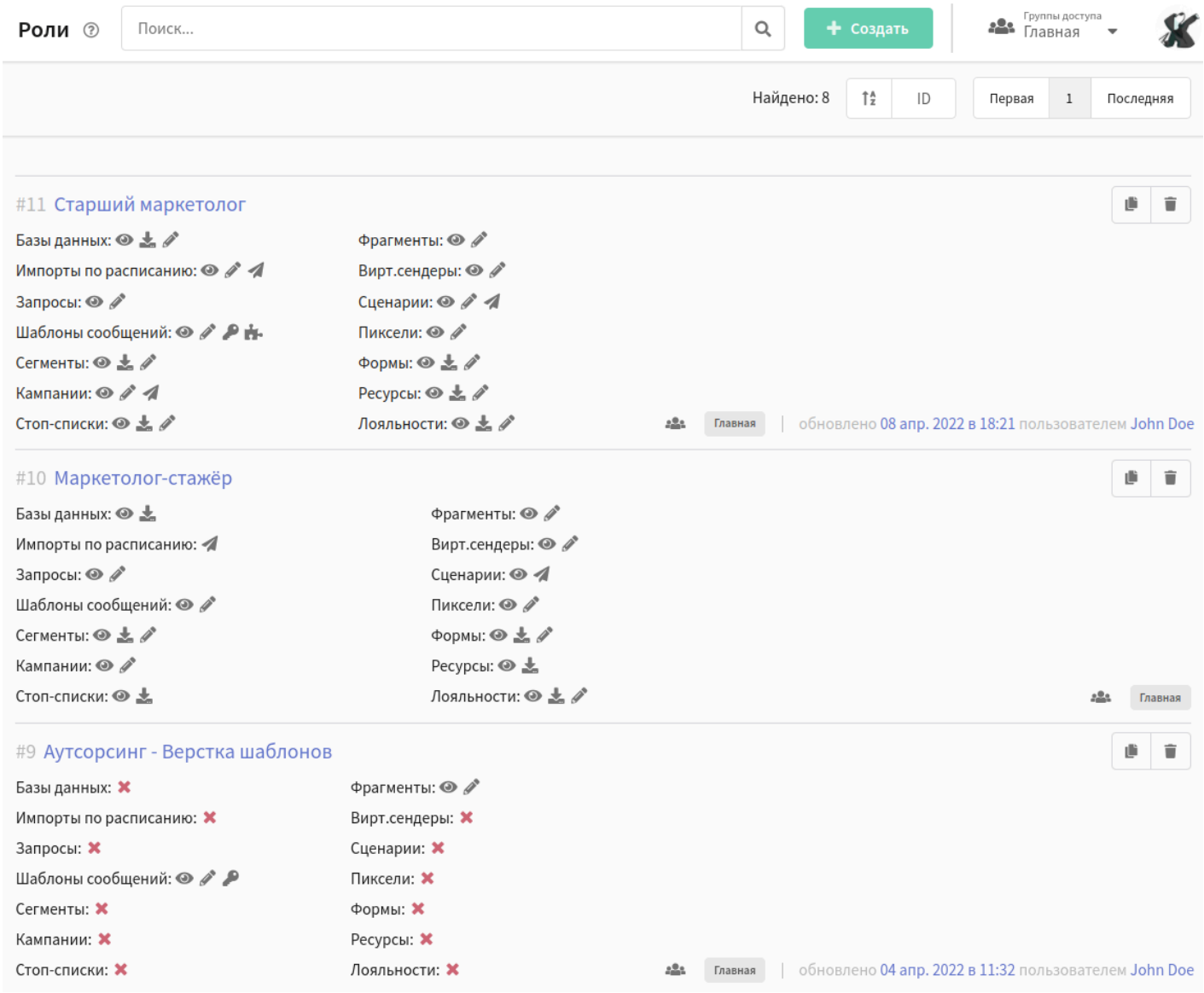
Удаление группы доступно в этом же списке — только **мастер-пользователям**.



Роли доступа

Роль доступа определяет, какие действия с объектами аккаунта разрешены пользователю. Чтобы перейти к управлению ролями, в главном меню выберите **Настройки —> Роли.**

В открывшемся списке отображаются существующие роли. Роль можно клонировать или удалить из контекстного меню в правой части плитки. Если вы хотите отредактировать роль, нажмите на её название.



Чтобы создать новую роль, нажмите **+ Создать** в верхней части страницы.

Для каждой роли настраиваются три параметра:

- **Название**, которое видят **мастер-пользователи** в списках ролей.
- **Группы**, на которые будут действовать ограничения роли.
- **Правила доступа** к объектам выбранных Групп.

CAUTION

Если в роль включить **Главную группу**, эта роль будет применяться ко всем объектам аккаунта. На **мастер-пользователя** ограничения ролей и групп не распространяются. Исключением является возможность **удаления профилей**, ее необходимо активировать самостоятельно, в том числе и для мастер-пользователя.

Для всех объектов аккаунта можно разрешить **Просмотр**, а также **Создание и редактирование**. Для баз данных, сегментов, стоп-списков и ресурсов доступны также **Чтение и выгрузка** — скачивание списков данных.

NOTE

Роль без опции **Модерация** в блоке **Доступ к сообщениям** разрешит сохранять шаблоны сообщений только как черновики. Такие черновики будут доступны в рассылках только после активации — пользователем с соответствующими правами доступа. Такую роль удобно устанавливать для учётных записей подрядчиков, создающих шаблоны рассылок.

Для Рассылок и Сценариев можно запретить **Запуск**: пользователи с такой ролью смогут только готовить рассылки, а проверять и запускать их будет в таком случае пользователь с достаточными правами доступа.

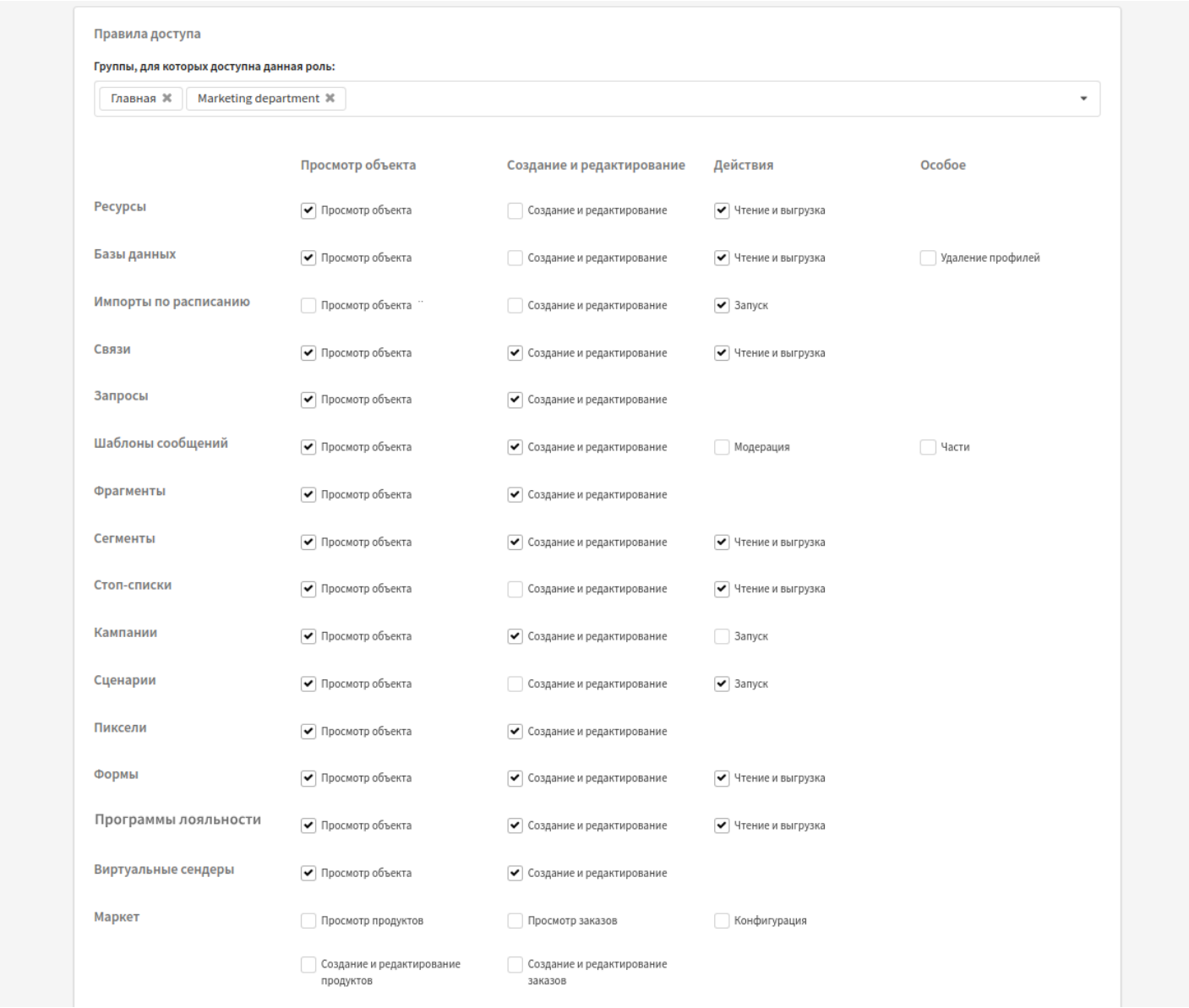
Для Кампаний доступны отдельные права доступа:

- **Кампании - Просмотр объекта** — просмотр кампаний и их настроек
- **Кампании - Создание и редактирование** — создание и изменение кампаний
- **Кампании - Запуск** — активация, деактивация и завершение кампаний

TIP: Зависимости пермишенов

Для работы с Кампаниями требуются дополнительные права доступа:

- **Кампании - Просмотр объекта** требует прав просмотра на: Рассылки, Сценарии, Сегменты, Базы данных
- **Кампании - Запуск** требует прав запуска на: Рассылки и Сценарии



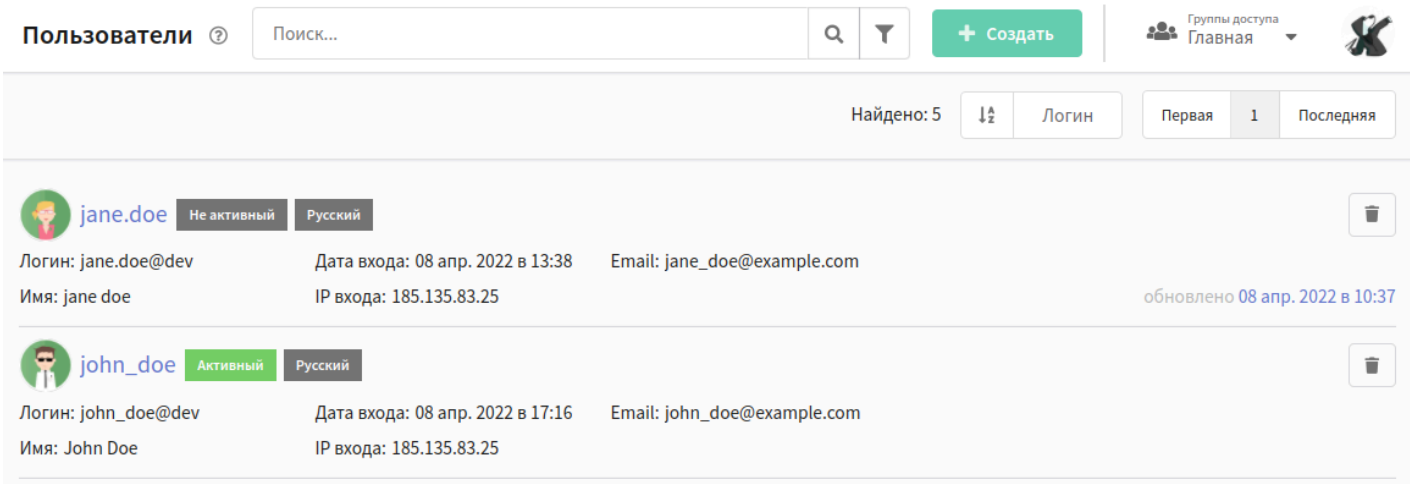
Роли могут ограничить действия пользователя со следующими **объектами платформы:**

- Ресурсы — инструмент для управления подписками клиентов на различный контент по определённым каналам связи.
- Базы данных — база для хранения данных о клиентах.
- Импорты в базу данных по расписанию — автоматическая синхронизация базы профилей Altcraft с внешней базой (например, корпоративная CRM или SQL база).
- Связи — используются для регистрации взаимодействий между вашими клиентами, или клиентами и менеджерами компании.
- Запросы — SQL-запросы к внешним базам. Роль регулирует работу пользователя с запросами для шаблонов и с запросами сегментации.
- Шаблоны сообщений — контент, который вы отправляете подписчикам в рамках рассылки.
- Фрагменты сообщений — часть шаблона, которую можно использовать в других шаблонах.
- Сегменты — часть базы данных, которая включает профили, подходящие под определённые условия. Роль регулирует действия со статическими и динамическими сегментами.
- Стоп-списки — список email-адресов, доменов и телефонных номеров, исключенных из рассылки.
- Рассылки — броадкасты, регулярные и триггерные рассылки, мультивариантные тесты, размещения.
- Кампании — объединение рассылок, сценариев и других активностей в единую стратегию.
- Сценарии — автоматизированная цепочка из условий и действий, которые последуют при их выполнении.

- Пиксели отслеживания — HTML или JavaScript-код, размещенный на веб-сайте или в приложении, который предназначен для отслеживания действий пользователей.
- Формы — элемент на странице сайта, с помощью которого пользователь вводит и отправляет вам свои данные.
- Программы лояльности — модуль для управления промоакциями и промокодами.
- Виртуальные сендеры — набор правил отправки сообщений для реальных модулей отправки email и SMS сообщений.
- Маркет — модуль для работы с данными о товарах и заказах интернет-магазина.

Учётные записи пользователей

Управление учётными записями пользователей аккаунта Altcraft доступно как в пользовательском интерфейсе, так и в Панели администратора. В этой статье рассмотрен функционал пользовательского интерфейса. Чтобы перейти к управлению учётными записями, в главном меню выберите **Настройки —> Пользователи**. В открывшемся списке отображаются существующие учётные записи.



Перейти к изменению настроек пользователя можно, нажав на название записи. Если вы хотите удалить пользователя, нажмите на иконку корзины справа

.

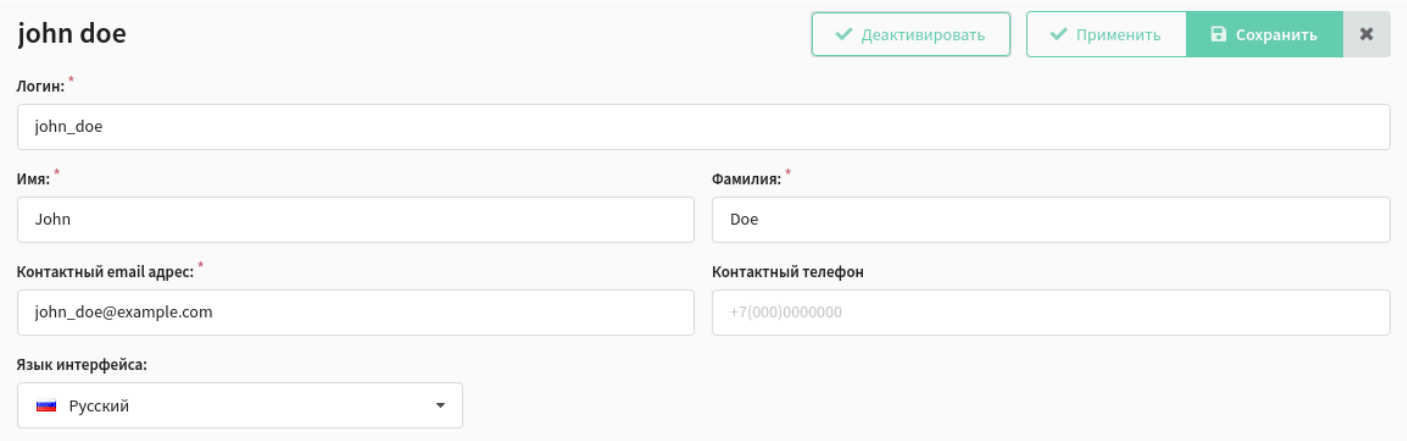
Как создать новую учётную запись

Чтобы добавить нового пользователя в аккаунт, нажмите **+ Создать** в верхней части страницы.

Основные настройки

В основных настройках укажите следующие данные:

- **Логин** — имя учётной записи. Логин используется для входа в аккаунт, вместе с алиасом аккаунта, в виде `login@account`.
- **Имя** и **фамилия** пользователя.
- **Контактный email адрес** — адрес, на который будут отправлены данные доступа.
- **Контактный телефон** — телефон пользователя.
- **Язык интерфейса** — выберите между английским и русским языком.



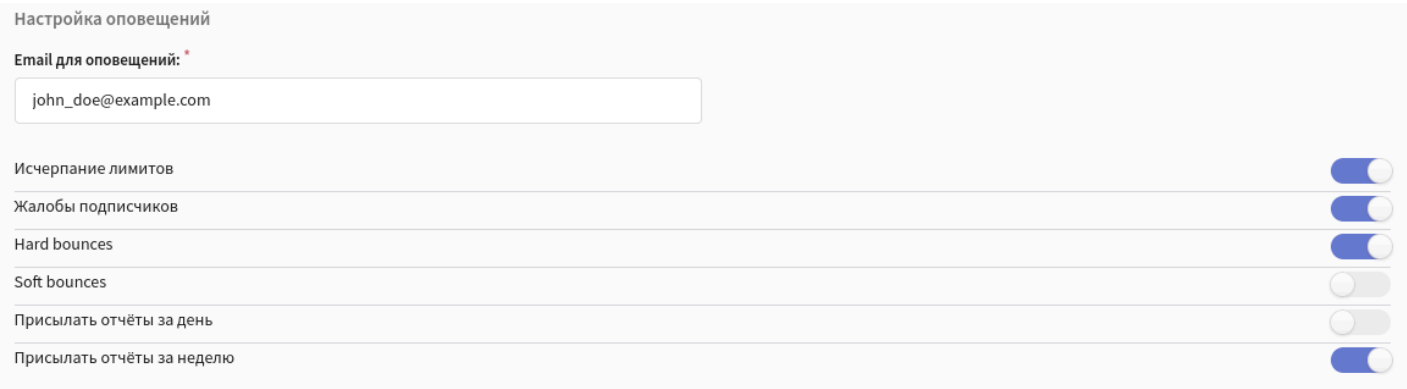
TIP

Если доступ в аккаунты платформы настроен по протоколу LDAP, через сервис директории организации, для авторизации в платформе будет использоваться логин сервиса директории и алиас аккаунта: user@company**@account**. Пароль при этом также задаётся в едином сервисе директории.

Настройка оповещений

Введите email-адрес, на который пользователю будут приходить опощения. Также выберите системные уведомления, о которых следует оповещать:

- **Исчерпание лимитов** — оповещение придет, если в тарифе аккаунта настроены ограничения, и количество профилей или количество отправленных сообщений достигло максимума;
- **Жалобы подписчиков** — процент жалоб на сообщения превысил допустимое значение.
- **Hard bounces** — процент hard bounce превысил допустимое значение.
- **Soft bounces** — процент soft bounce превысил допустимое значение.
- **Присылать отчёты за день/за неделю** — на почту пользователя с определённой периодичностью будет приходить отчёт, содержащий основные показатели рассылок, а также данные о приросте аудитории и достижении целей.

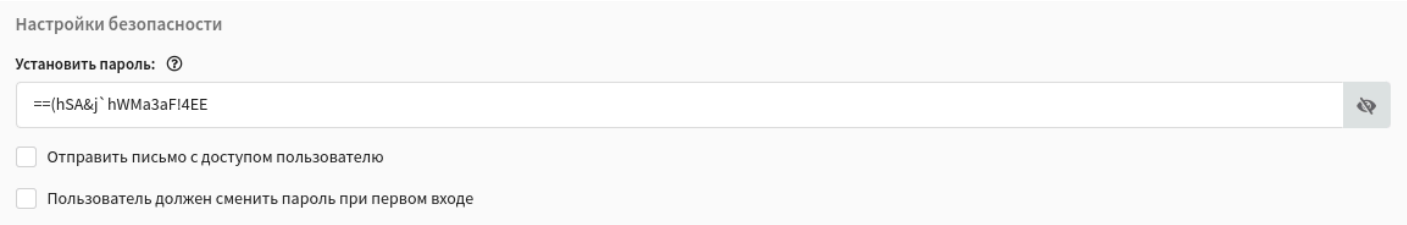


Настройка безопасности

Установите пароль для входа пользователя в учётную записи. Пароль должен содержать от 8 до 32 символов, включая цифры и спецсимволы.

После создания нового пользователя пароль можно передать владельцу автоматически, отметив опцию **Отправить письмо с доступом пользователю**. Чтобы сделать возможным редактирование этого поля, внесите какие-либо изменения в пароль. Например, можно добавить точку и тут же удалить её.

Если вы активируете поле **“Пользователь должен сменить пароль при первом входе”**, пользователю при входе придется нажать кнопку “**Forget password**”, ввести предоставленный логин и создать новый пароль.



NOTE

Если доступ в платформу настроен только по LDAP, управление паролями в интерфейсе не используется.

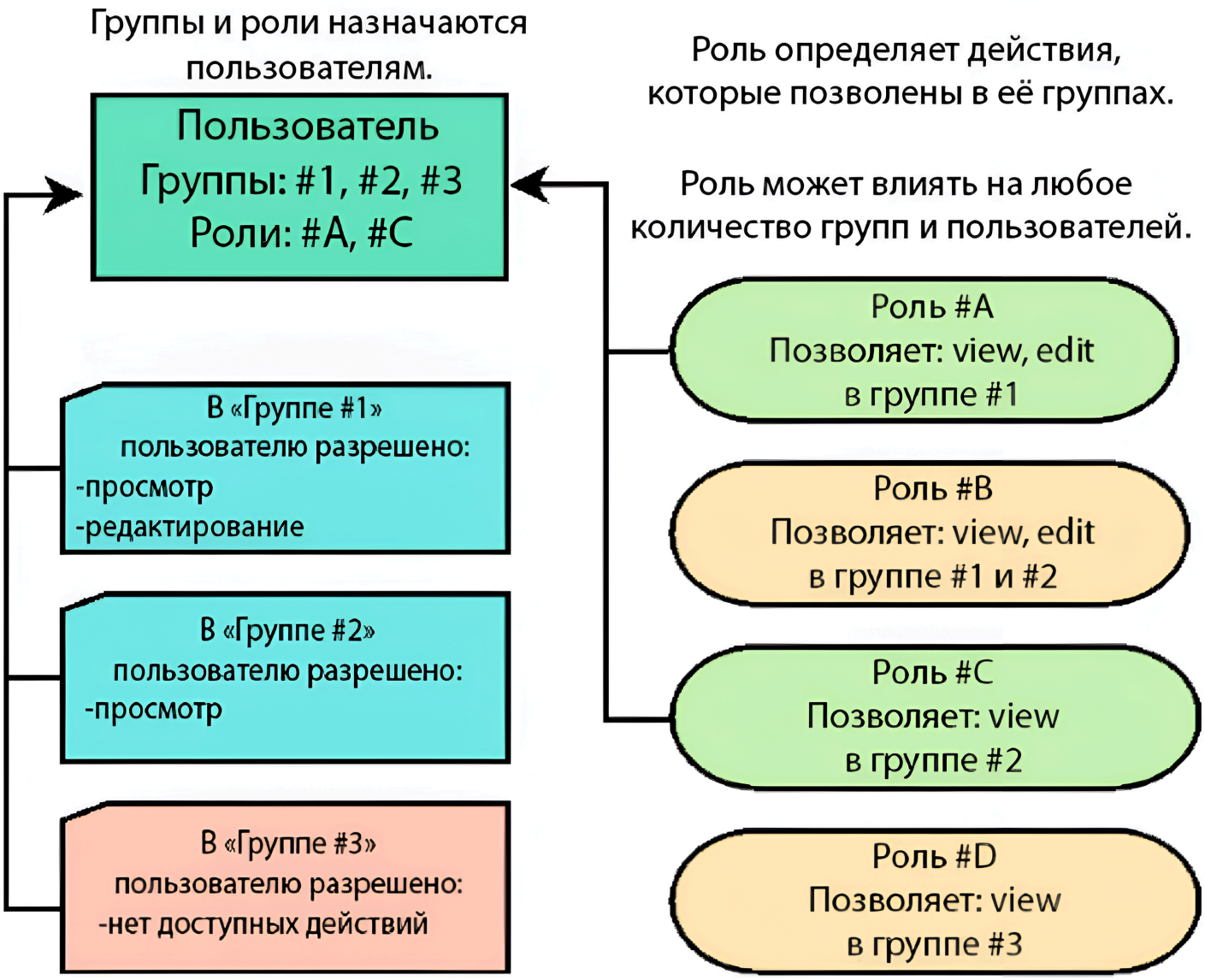
Доступ к объектам аккаунта

В блоке **Доступ** настраивается, какие действия может совершать пользователь с объектами тех или иных **Групп** аккаунта:

- **Мастер** — пользователю с такой настройкой будут доступны любые действия в **Главной группе** аккаунта. Соответственно, у него будет доступ ко всем объектам всех групп. При этом ограничения ролей никак не будут действовать. Мастер-пользователи имеют доступ к блоку **“Настройки”** в главном меню, где могут управлять аккаунтами других пользователей: удалять пользователей, редактировать их доступ, скачивать персональные данные, а также изменять настройки оповещений и безопасности.
- **Роли** — созданные ранее роли доступа.
- **Группы доступа** — группы объектов аккаунта, с которыми сможет взаимодействовать пользователь.

Для того чтобы пользователь мог авторизоваться в аккаунте, его учётная запись должна быть **активной**, и для неё должны быть установлены как минимум одна роль и как минимум одна группа.

На графике справа показано взаимодействие групп и ролей пользователя:

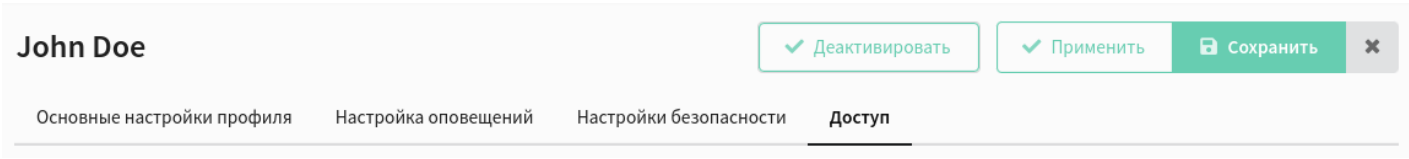


CAUTION

Если в роли не задана группа, доступная пользователю, правила роли не будут работать, и пользователь не сможет взаимодействовать с объектами группы.

Если в роли задана группа, которая недоступна пользователю — она останется недоступной.

После завершения настройки нажмите кнопку **“Сохранить”** или **“Применить”** в верхней части страницы:



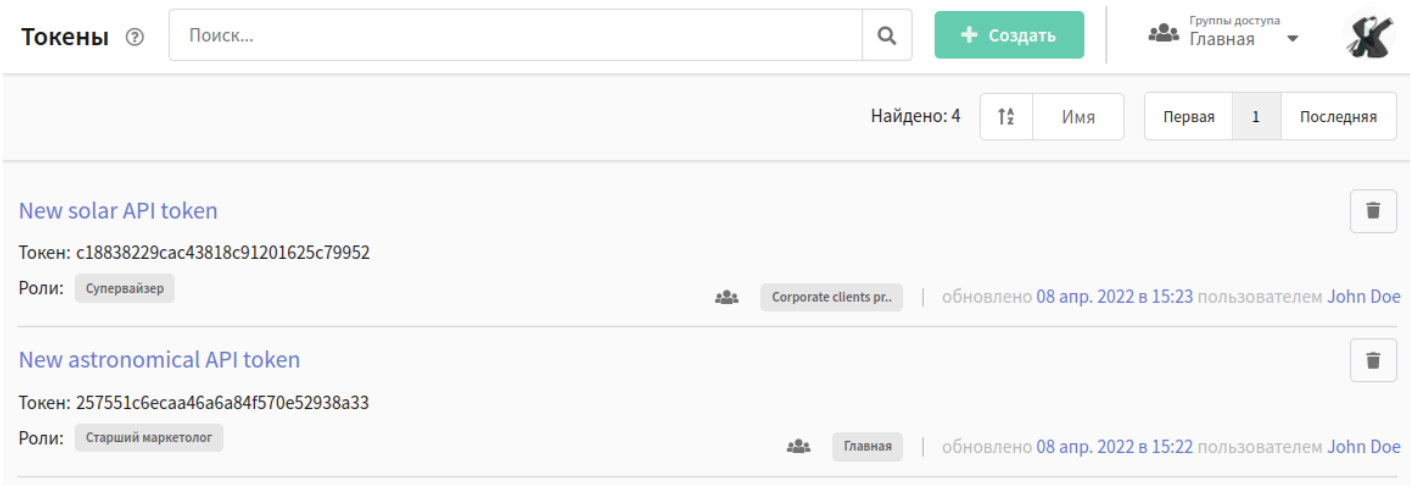
Кнопка **Деактивировать** сделает учетную запись неактивной. Пользователи с такой настройкой не смогут авторизоваться в платформе.

API-токены

API-токены используются для авторизации в аккаунте при использовании API. Чтобы перейти к управлению токенами, в главном меню выберите **Настройки —> Токены.**

В открывшемся списке отображаются существующие токены. Перейти к редактированию токена можно, нажав на его название. Если вы хотите удалить токен, воспользуйтесь кнопкой

справа. Чтобы создать новый токен, нажмите **+ Создать** в верхней части страницы.



Для нового токена подберите подходящее название или оставьте системное. Сам токен станет доступен после сохранения записи. Для токена, как и для учётной записи пользователя, необходимо установить группы видимости объектов аккаунта и связанные с ними роли доступа.

CAUTION

Если токену открыть доступ в **Главную группу**, для него будут доступны все объекты аккаунта, в том числе входящие в другие группы.

New astronomical API token

Обновлено 2022-04-08 15:22:23 пользователем john

Основные настройки

Имя токена: *

New astronomical API token

Токен
257551c6ecaa46a6a84f570e52938a33

Права доступа

Роли:

Старший маркетолог ✖

Группа:

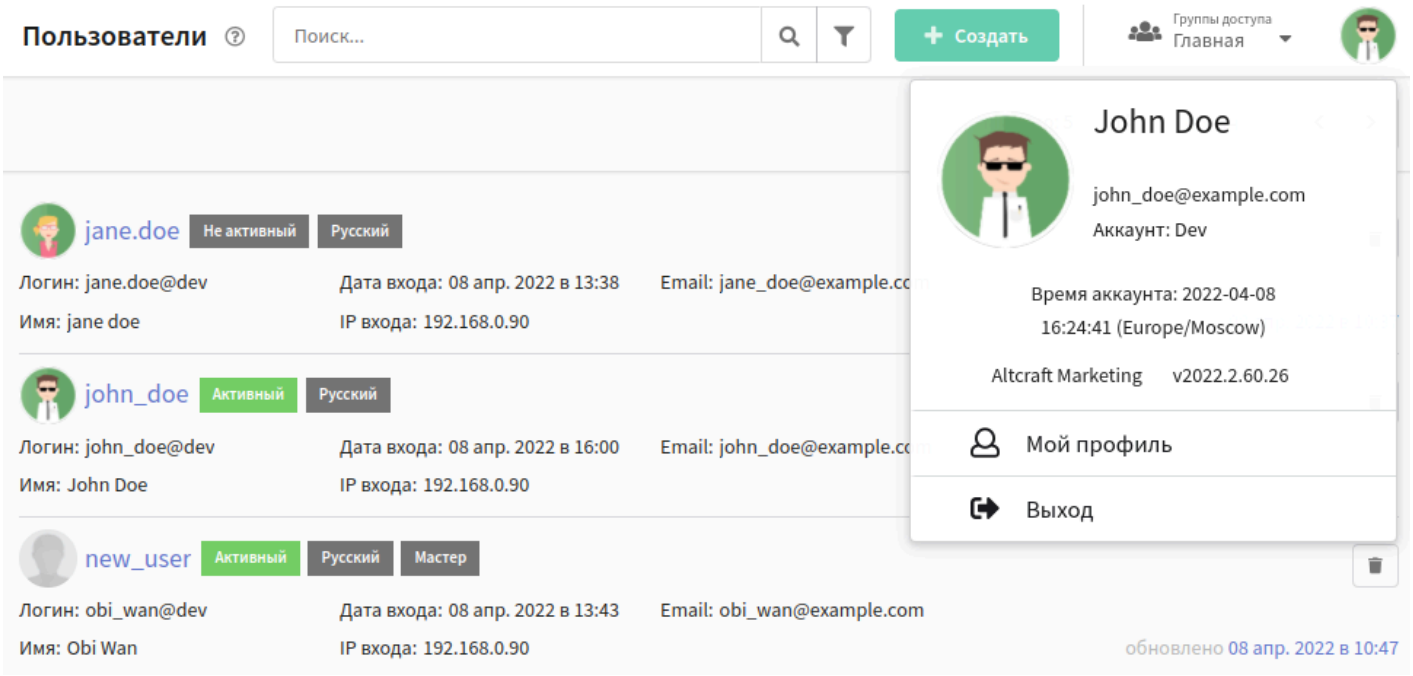
Главная

Двухфакторная аутентификация (2FA)

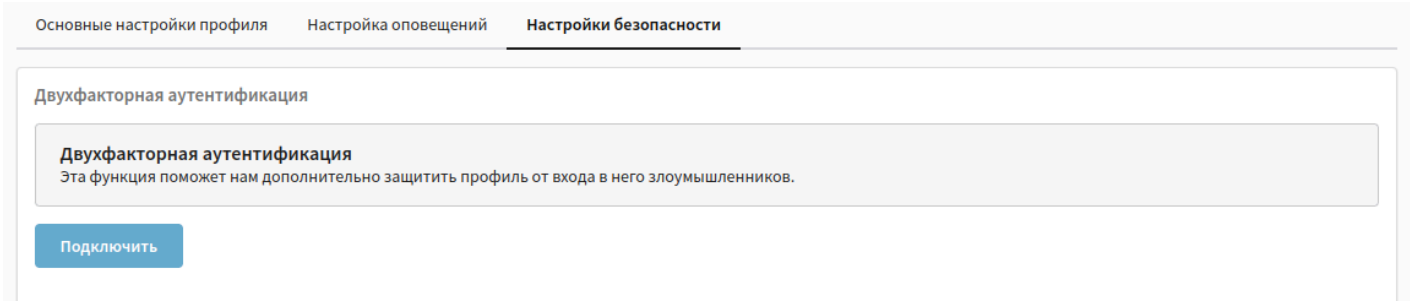
Двухфакторная аутентификация, или 2FA — это средство защиты вашей учетной записи не только посредством ввода логина и пароля, но и при помощи кода, известного только вам. Для прохождения аутентификации потребуется приложение-генератор одноразового кода, например, Google Authenticator. После включения 2FA ввод шестизначного кода станет обязательным шагом при каждом входе в платформу.

Как включить 2FA для отдельного пользователя

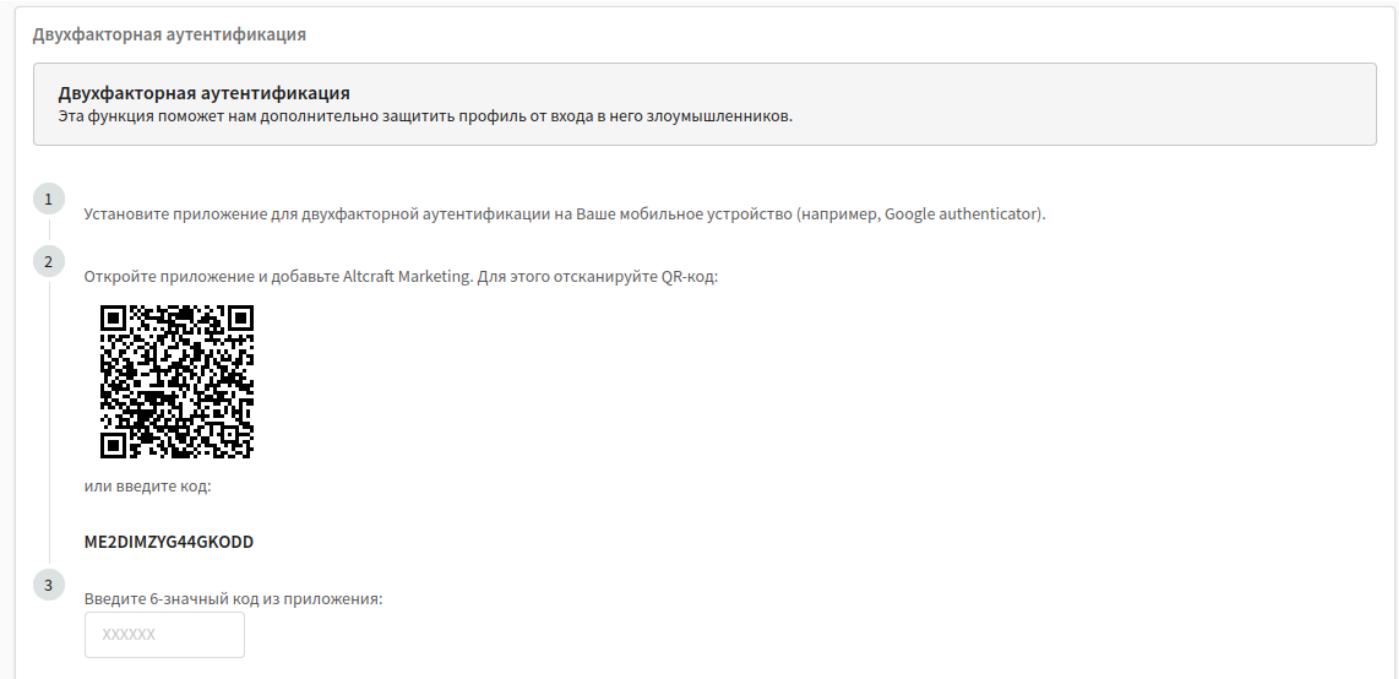
1. Зайдите в настройки своего профиля. Для этого кликните на аватарку в правом верхнем углу. Далее —> **Мой профиль**.



2. Перейдите на вкладку **Настройки безопасности**. В блоке **Двухфакторная аутентификация** нажмите “**Подключить**”:



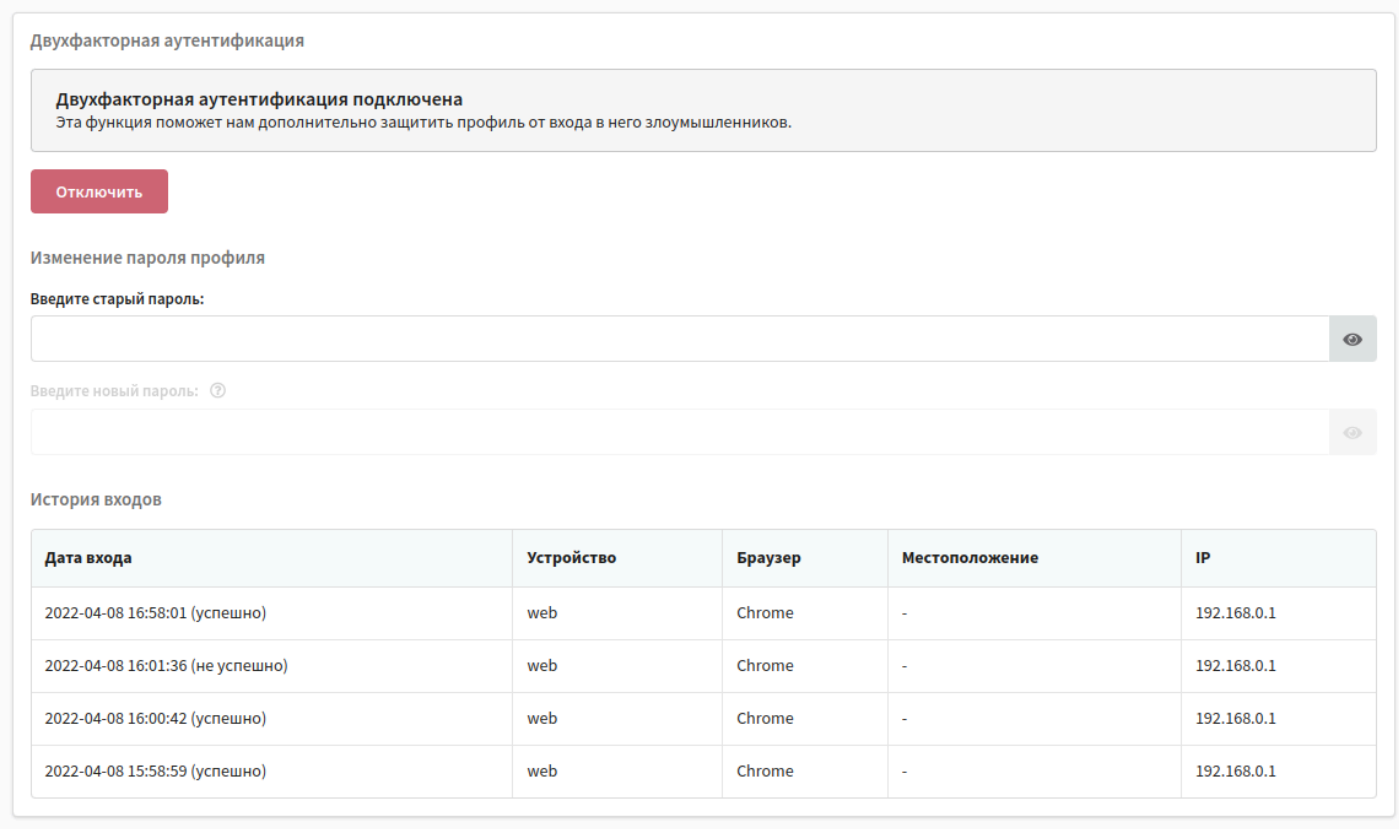
3. Откроется инструкция для подключения. Следуйте указанным в ней шагам: скачайте приложение, добавьте в него Altcraft Marketing и введите полученный 6-значный код. Затем нажмите “**Подключить**”.



4. На странице появится список **резервных кодов**. С помощью них вы сможете восстановить доступ к своей учётной записи в случае потери устройства, на котором установлено 2FA приложение. Каждый резервный код можно использовать только один раз. В дальнейшем вы не сможете увидеть этот список в платформе, поэтому скопируйте его и храните в доступном месте.

Настройка 2FA завершена. Теперь каждый раз при входе в платформу пользователь должен будет ввести не только логин и пароль, но также 6-значный код, полученный в приложении для аутентификации.

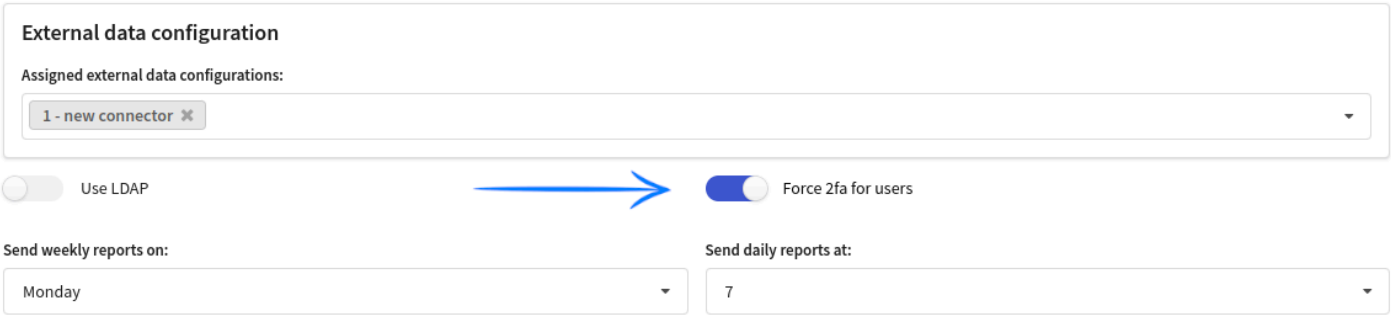
Информация об успешных и неуспешных попытках входа будет отображаться в блоке **История входов**:



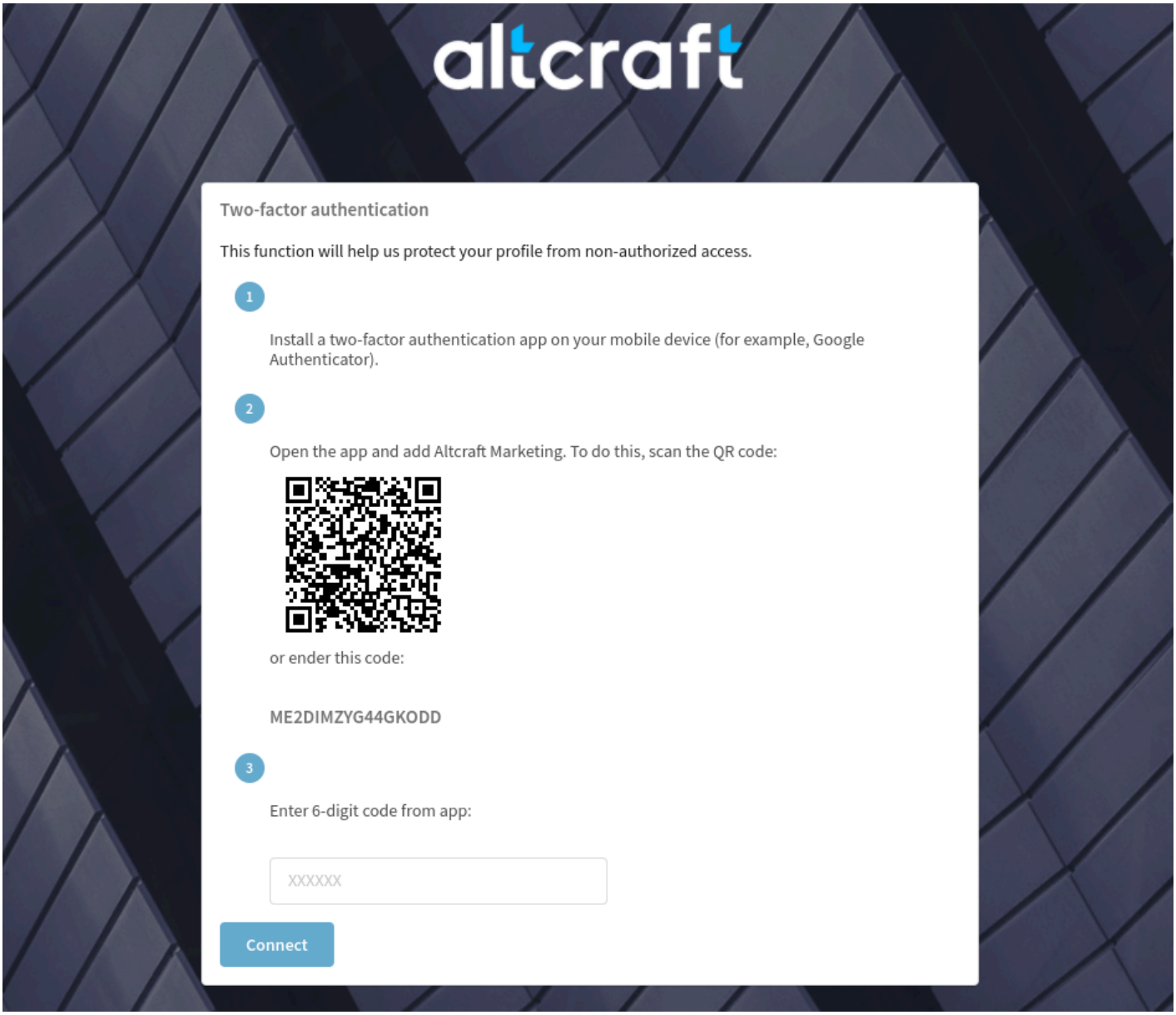
Если вы решите, что двухфакторная аутентификация вам больше не нужна, нажмите на кнопку “**Отключить**”.

Как включить 2FA для всех пользователей аккаунта

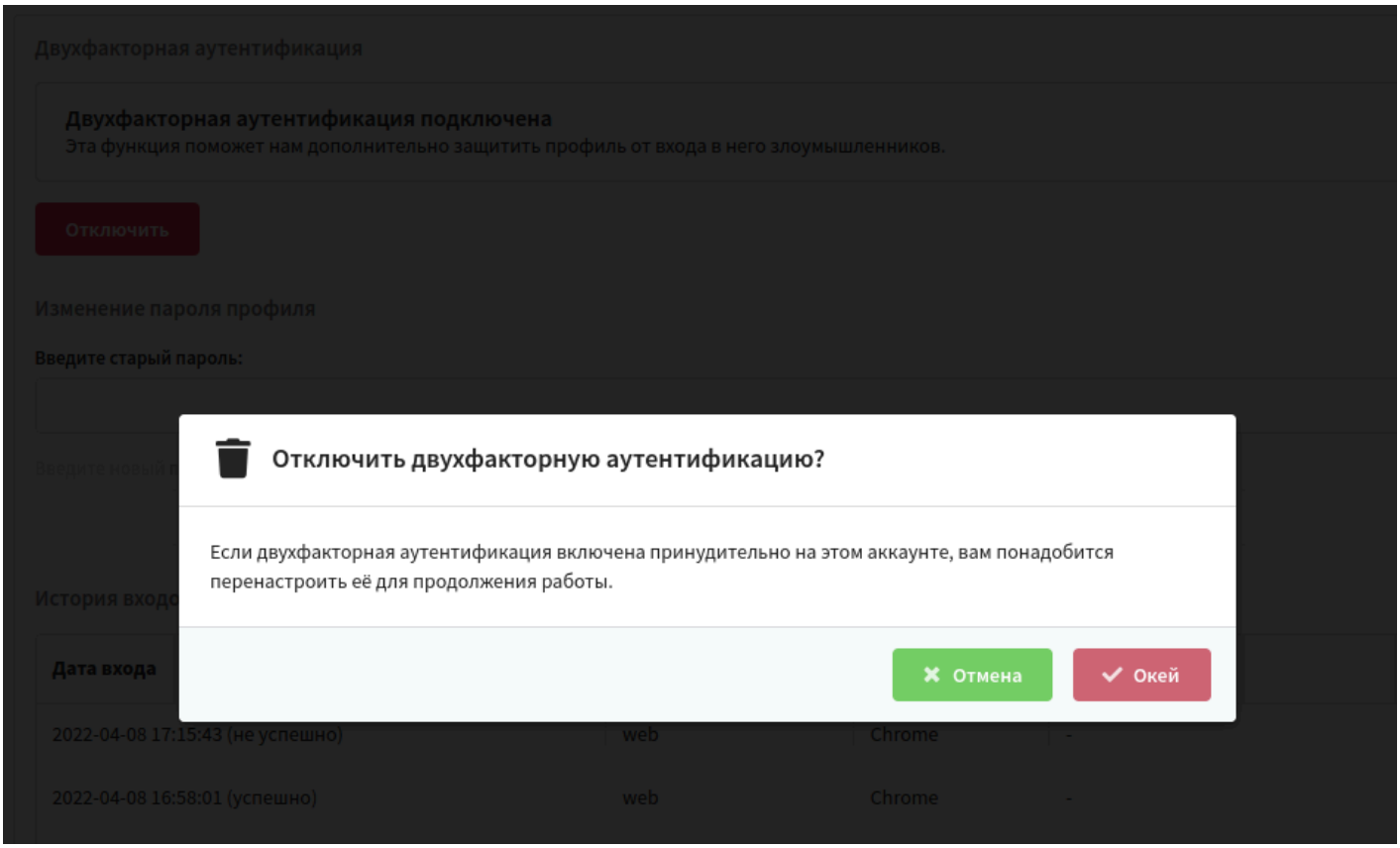
Чтобы включить принудительную двухфакторную аутентификацию для всех учётных записей аккаунта, администратор платформы должен активировать поле **Force 2fa for users** в настройках аккаунта в Панели администратора:



После активации принудительной 2FA любому пользователю для входа в платформу потребуется настроить двухфакторную аутентификацию. Окно для настройки откроется при вводе корректных логина и пароля:



Если двухфакторная аутентификация является обязательной для всех учетных записей, пользователь не сможет отключить её. При клике на кнопку “**Отключить**” всплывёт предупреждающее окно:



Если пользователь всё-таки отключит 2FA, он принудительно покинет платформу. Для входа ему потребуется заново настроить 2FA.

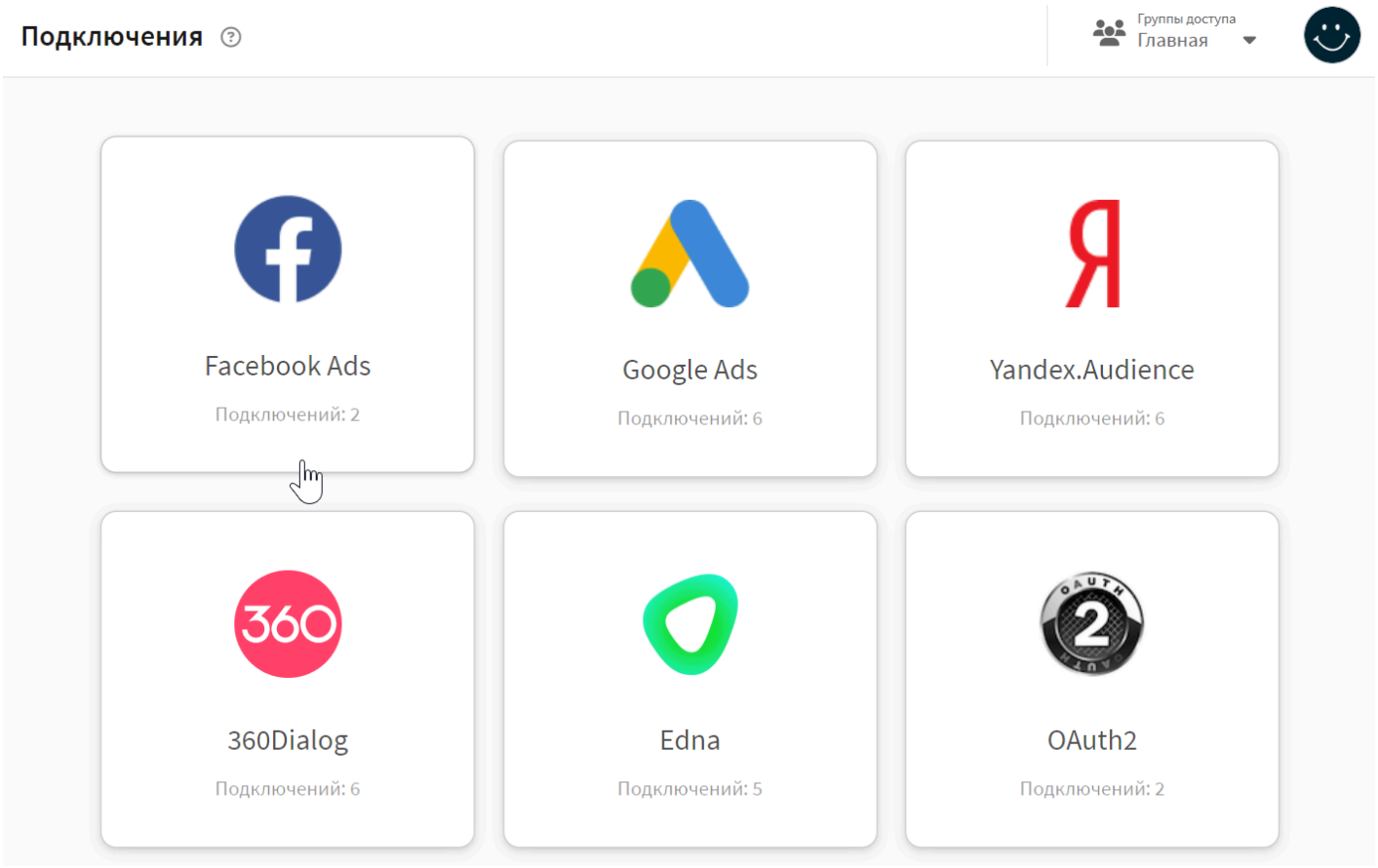
Подключения

	Подключения позволяют авторизоваться во внешних системах для расширения возможностей Altcraft Marketing функционалом этих систем. Обычно для создания такого подключения требуется аккаунт во внешней системе с правами доступа через API.
	`Подключения настраиваются в разделе <strong>`
	`Настройки –> Подключения.</strong>`
	`</td>`
	`</tr>`

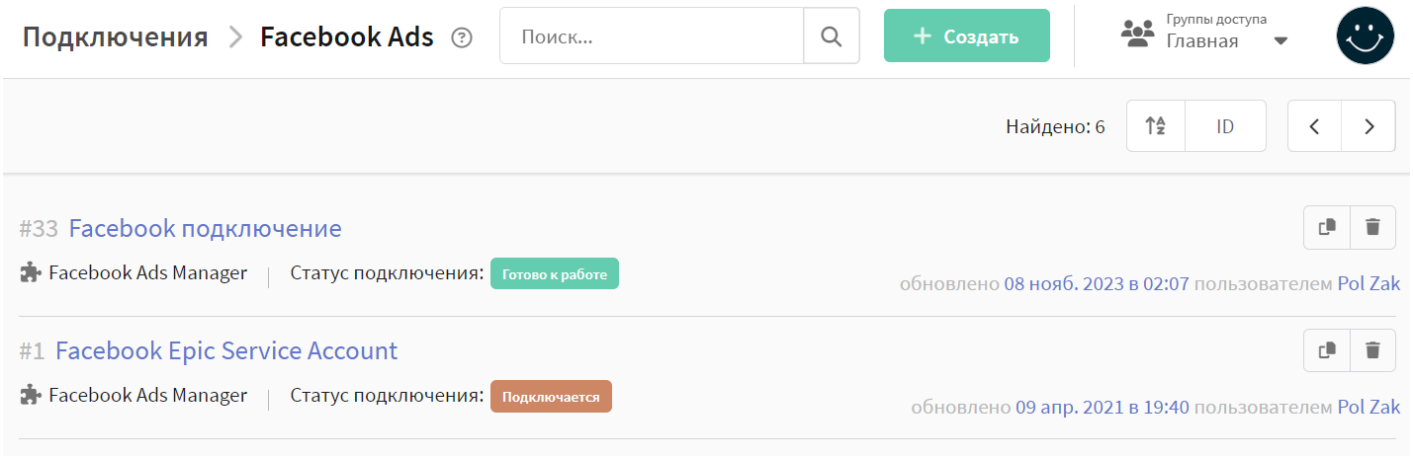
Доступные подключения

Подключение к Facebook Ads

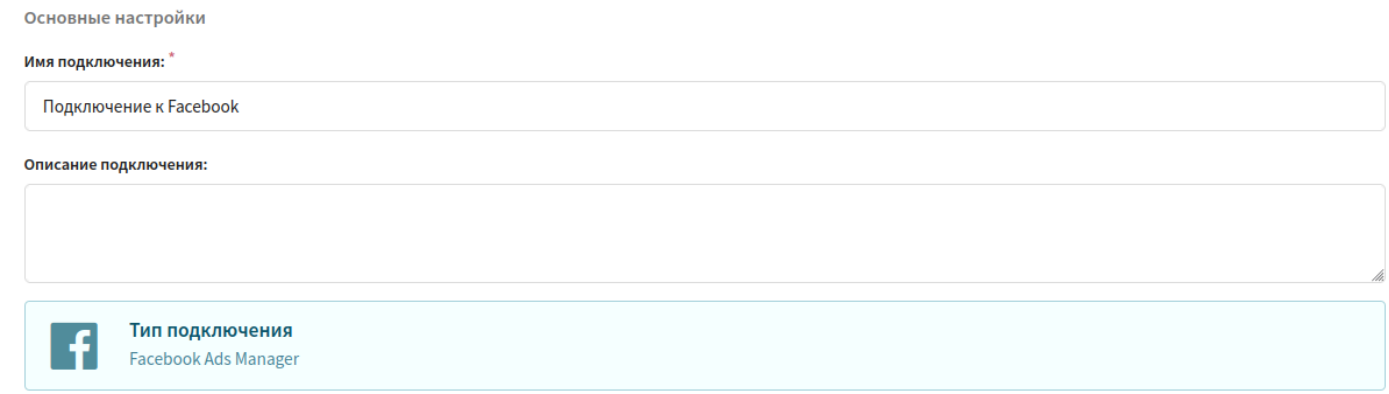
Для добавления нового подключения откройте раздел **Настройки** —> **Подключения**. Выберите тип подключения *Facebook Ads**.



Откроется список всех настроенных Facebook*-подключений. Чтобы добавить новое, нажмите **+ Создать**.



В основных настройках укажите название и описание нового подключения:



Для подключения к Facebook* требуется данные Facebook* приложения, через которое платформа будет взаимодействовать с Facebook* Marketing API.

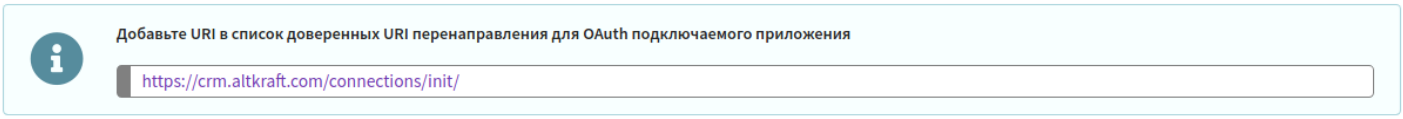
Создать приложение можно на странице https://developers.facebook.com/apps/.

Необходимо приложение с типом **Business**. Документацию можно найти здесь.

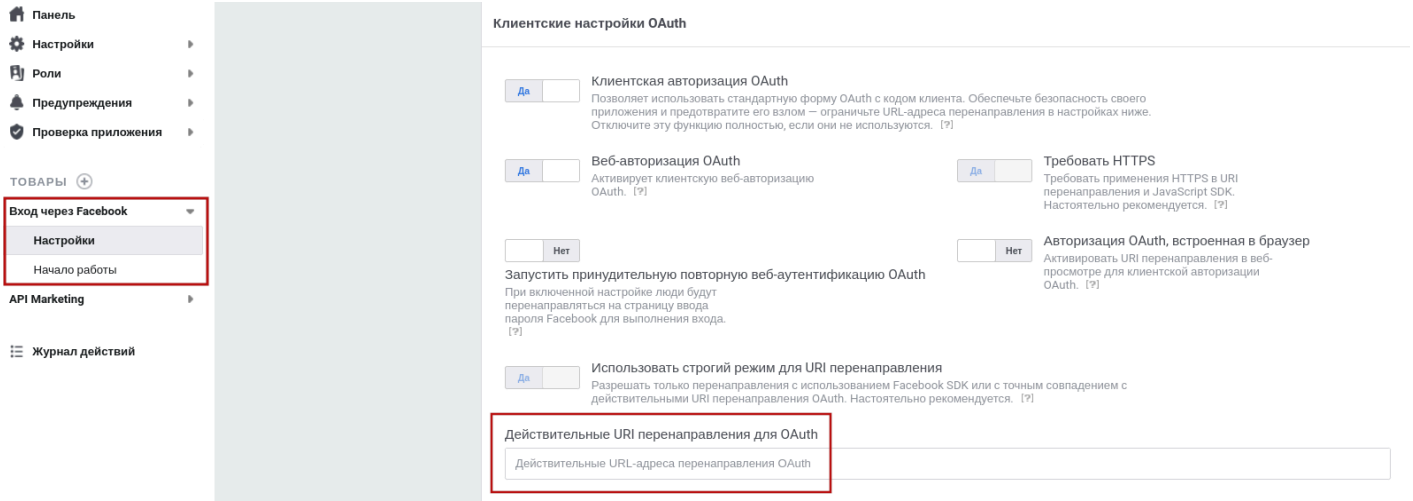
Прежде всего, на панели приложения в разделе **Products** нужно добавить:

- API Marketing
- Вход через Facebook*

В настройках последнего продукта нужно добавить URI, указанные в новом подключении, в список **действительных URI перенаправления для OAuth**:



URI необходимо добавить в это поле:



CAUTION

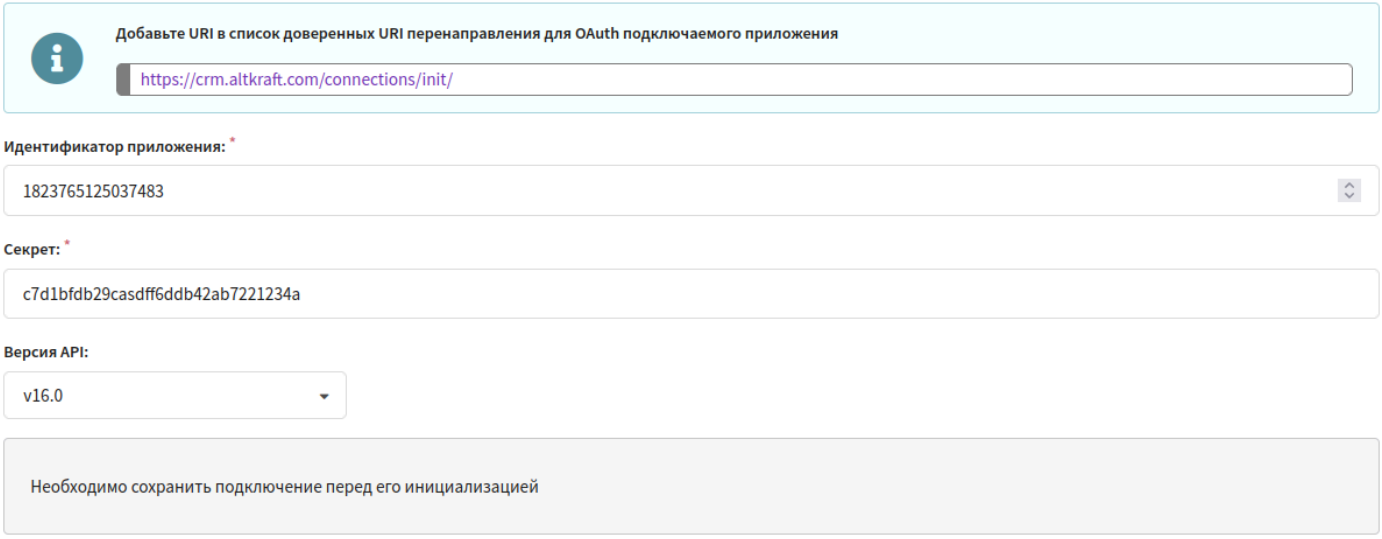
Приложение должно иметь права ads_management. В большинстве ситуаций стандартного доступа будет достаточно (см. получение разрешений).

Конфигурация на стороне Altcraft Platform состоит из следующих шагов:

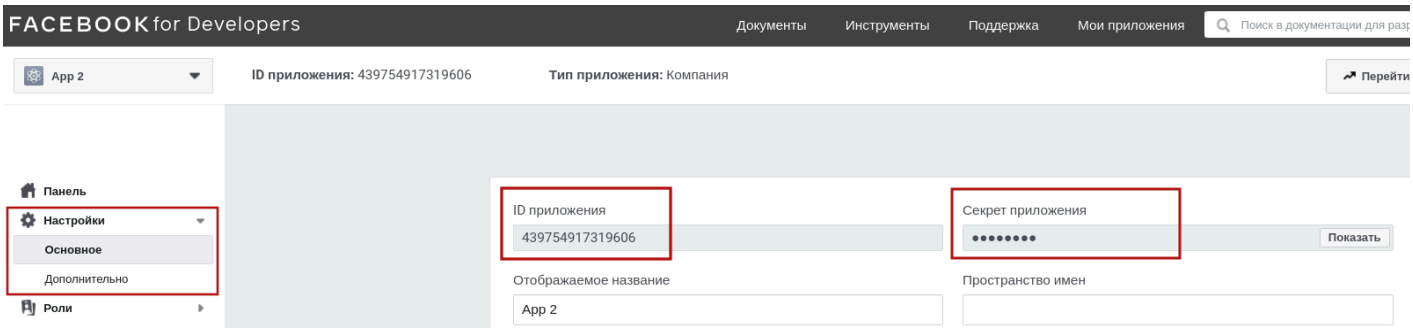
- Предварительная настройка Facebook* подключения
- Инициализация подключения
- Выбор рекламного аккаунта

Предварительная настройка Facebook подключения

Перед инициализацией подключения нужно заполнить поля следующей формы:

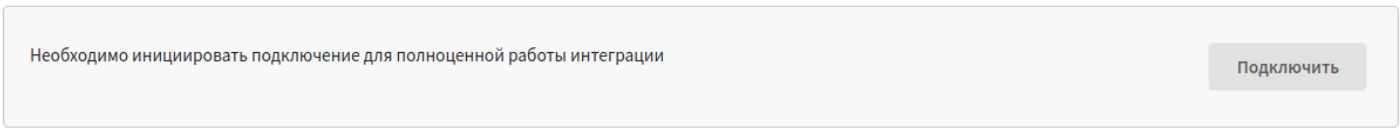


Идентификатор приложения и **Секрет** можно найти на панели приложения в настройках.



Версию API можно посмотреть в разделе **Настройки —> Дополнительно**.

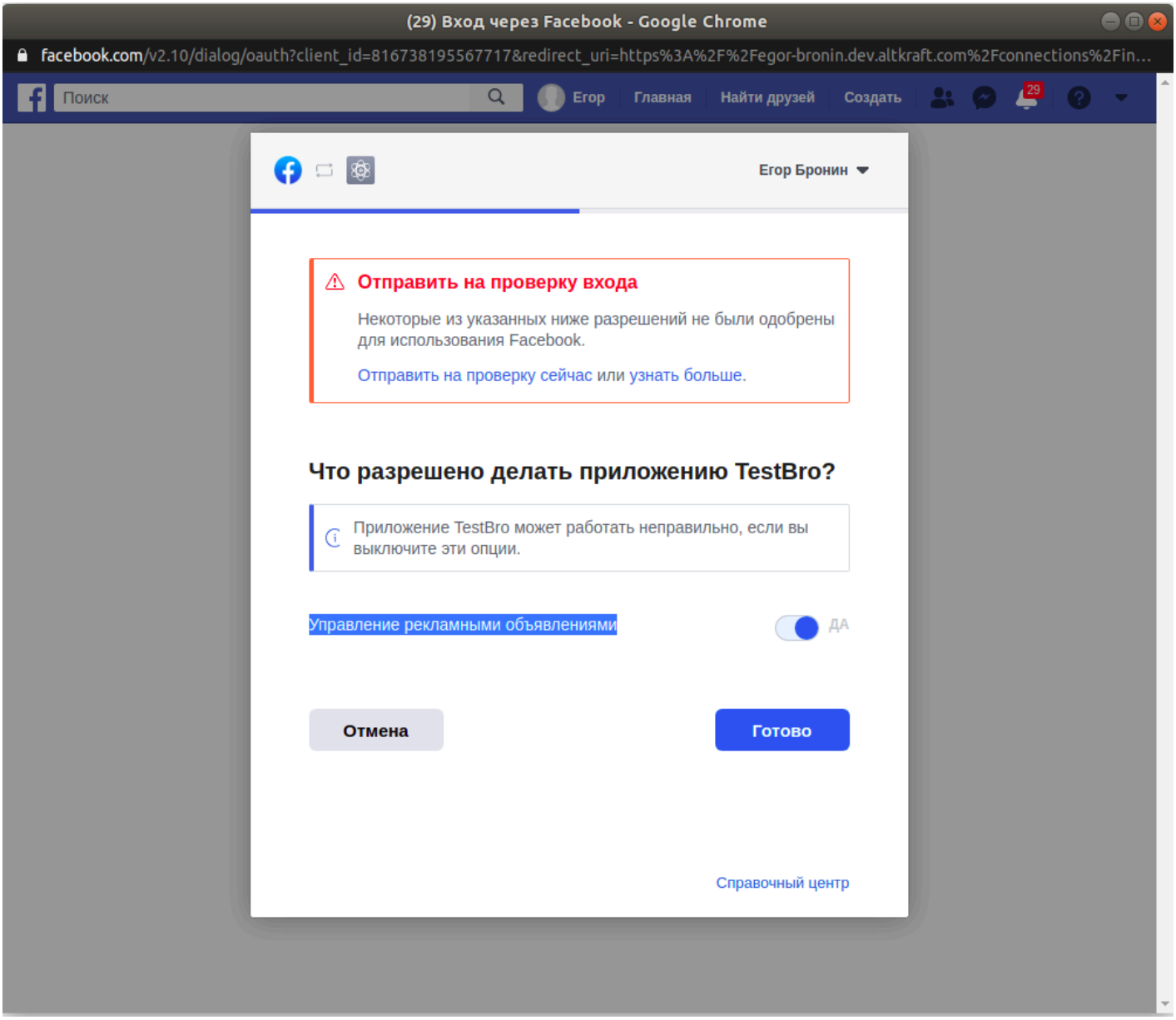
Чтобы продолжить, необходимо сохранить подключение. Только после этого появится кнопка “**Подключить**”:



Инициализация Facebook подключения

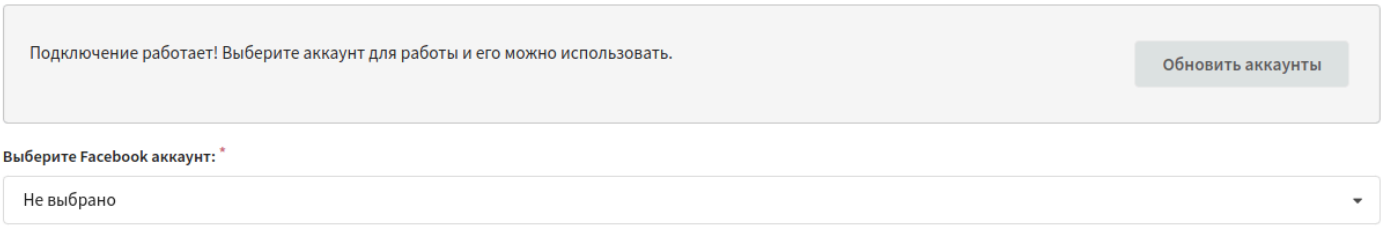
При нажатии на кнопку “Подключить” произойдет редирект в Facebook* с предложением авторизоваться. Авторизовываться нужно под профилем, у которого есть права на управление рассылками в целевом рекламном аккаунте.

После этого будут запрошены права на управление рекламными объявлениями для вашего приложения от лица и с правами профиля.

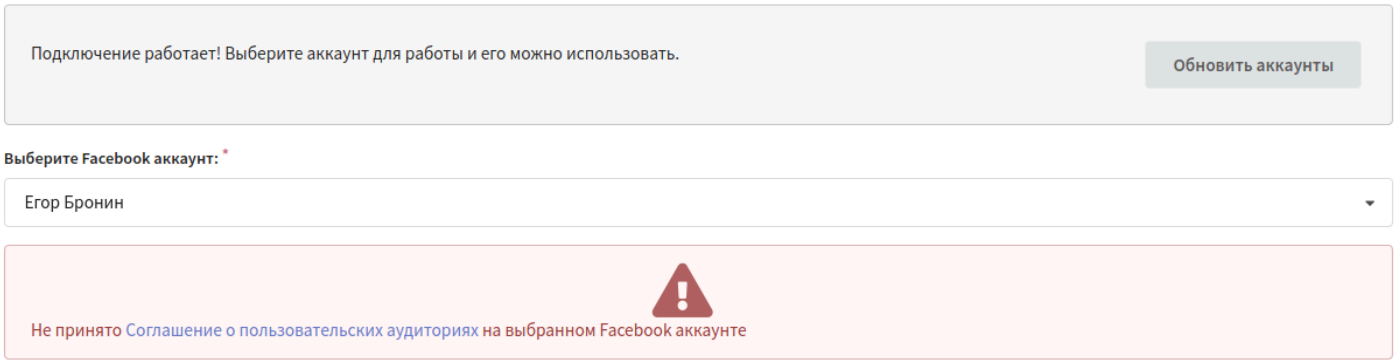


Выбор рекламного аккаунта Facebook

Если предыдущие шаги завершились без ошибок, то останется просто выбрать рекламный аккаунт, с которым можно взаимодействовать с этим подключением.



При выборе аккаунта может возникнуть следующая ошибка:



Необходимо принять указанные соглашения и **обновить аккаунты.**

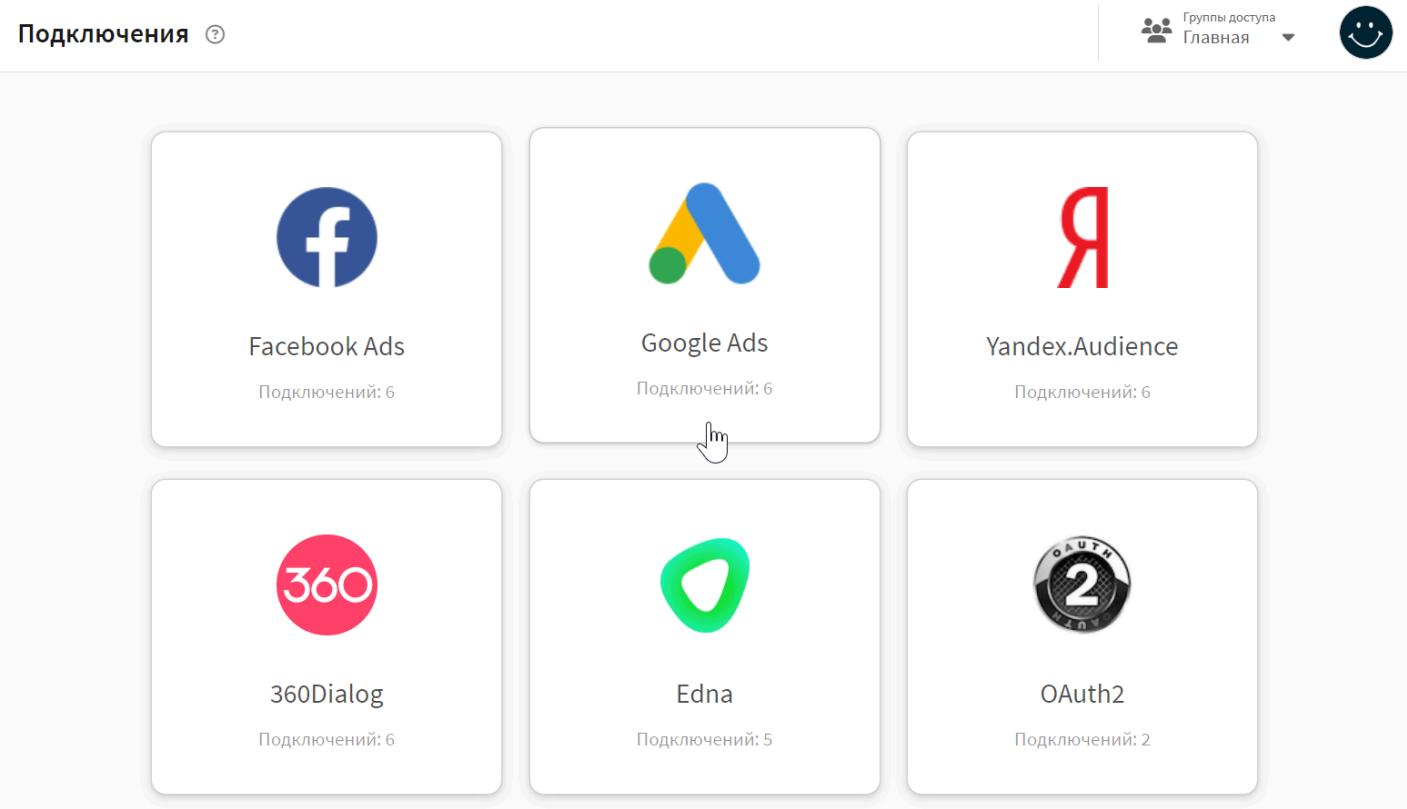
**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Подключение к Google Ads

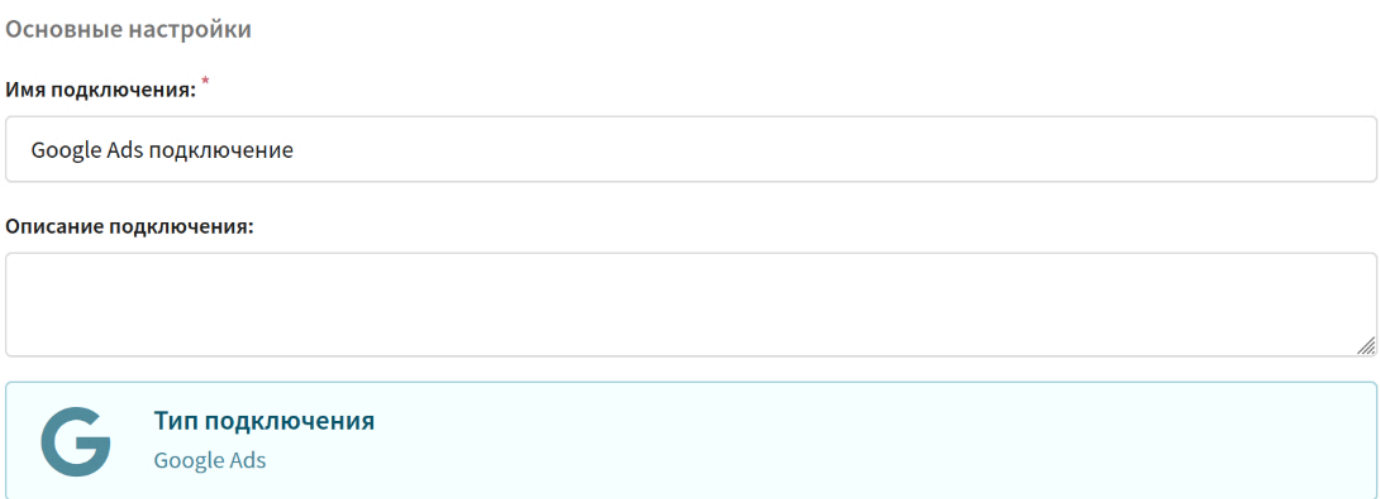
Google Ads — это сервис для размещения контекстной рекламы на страницах поисковой системы Google и на партнерских сайтах контекстно-медийной сети (Gmail, YouTube и др.). Подключение свяжет ваш аккаунт Google Ads с Altcraft Platform. Далее вы сможете использовать это подключение, чтобы настроить интеграцию с Google Ads и синхронизировать статические сегменты Altcraft с аудиториями рекламного кабинета Google.

Как создать подключение

В главном меню перейдите в раздел **Настройки** —> **Подключения**. Выберите тип подключения — **Google Ads**. Откроется список всех настроенных к данному сервису подключений. Чтобы добавить новое, нажмите кнопку **“Создать”** на верхней панели.



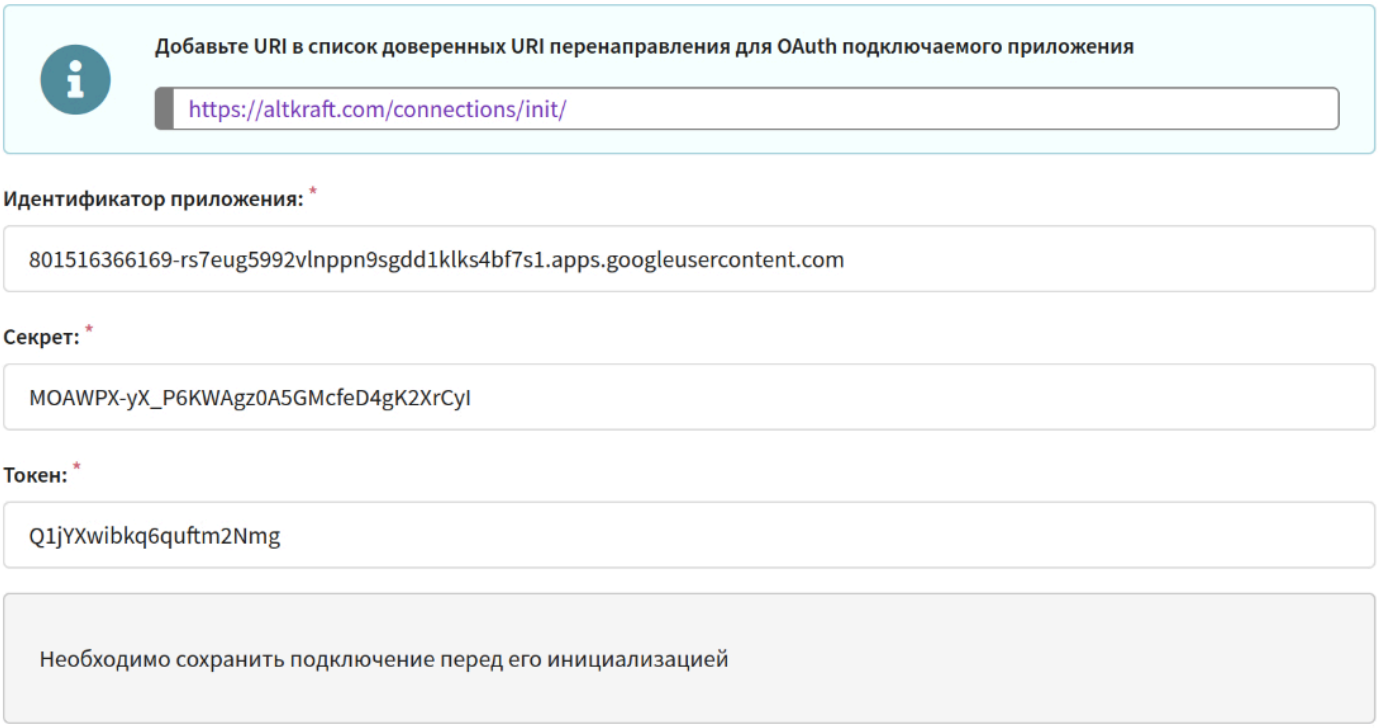
В основных настройках укажите имя подключения и при необходимости его описание:



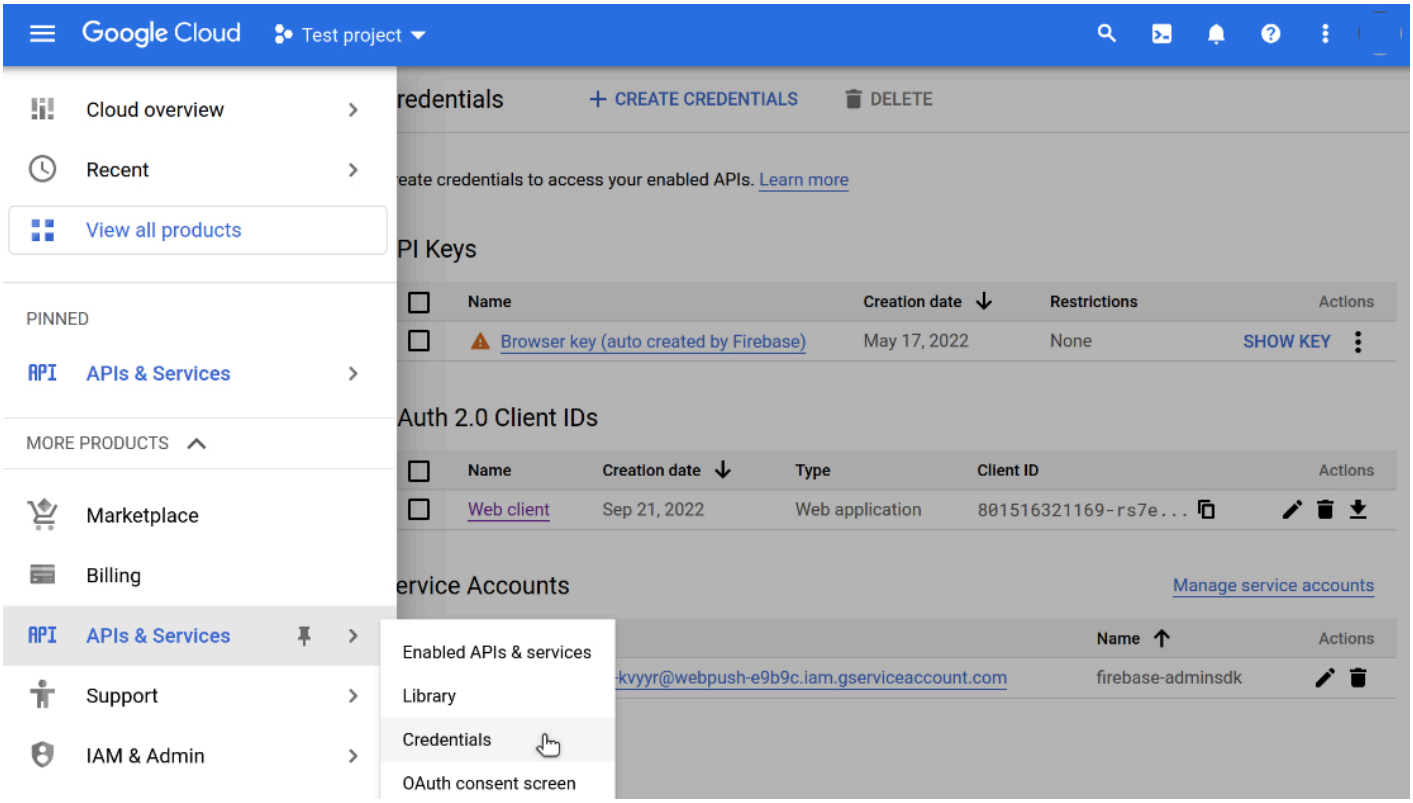
Шаг 1. Предварительная настройка Google подключения

Для подключения к Google требуется данные Google приложения, через которое платформа будет взаимодействовать с Google Ads. Если у вас еще нет приложения, то создайте его, перейдя по ссылке: https://console.cloud.google.com/.

Внесите данные приложения в платформу:



Идентификатор приложения и **Секрет** вы можете найти в Google Cloud. Для этого перейдите в раздел **APIs & Services** —> **Credentials** и кликните на нужный **OAuth 2.0 Client ID**, чтобы просмотреть данные:



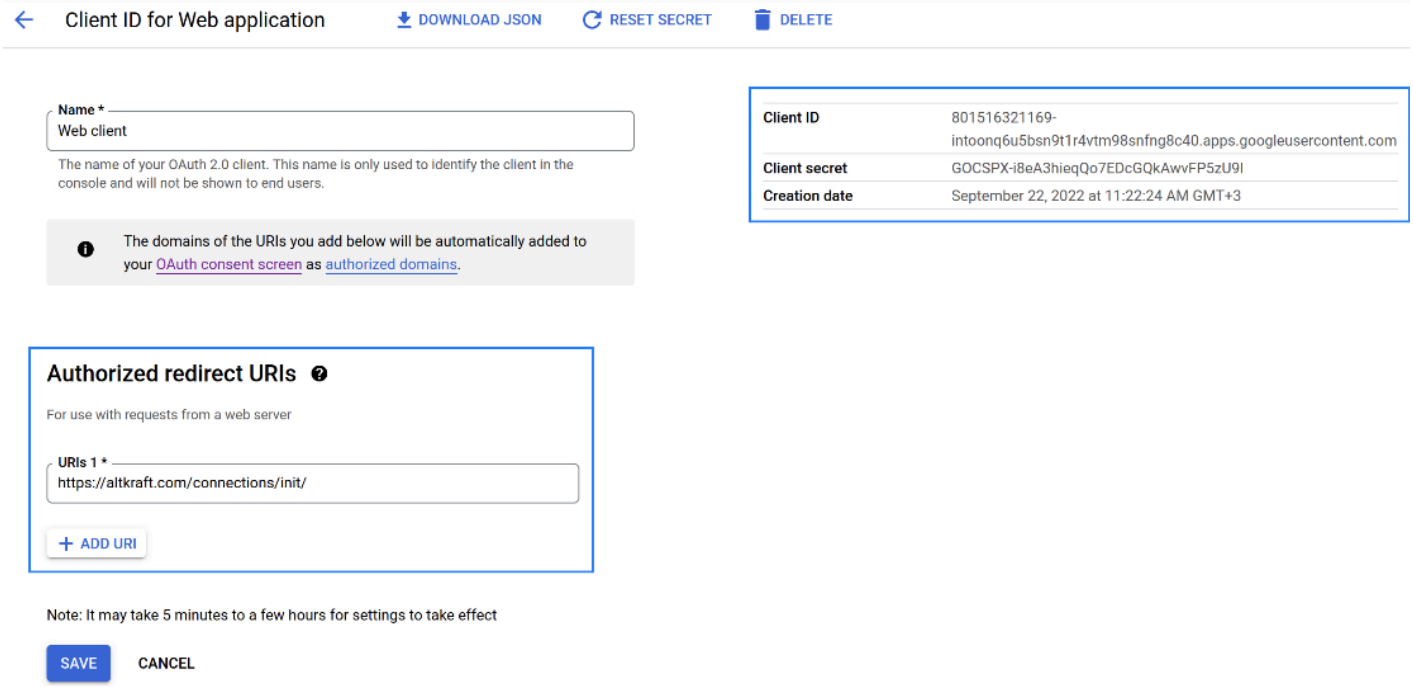
TIP

Чтобы создать новый OAuth 2.0 Client IDs, нажмите кнопку “**Create credentials**” в верхней части страницы. Выберите **OAuth 2.0 Client ID** и укажите **Web application** как тип приложения.

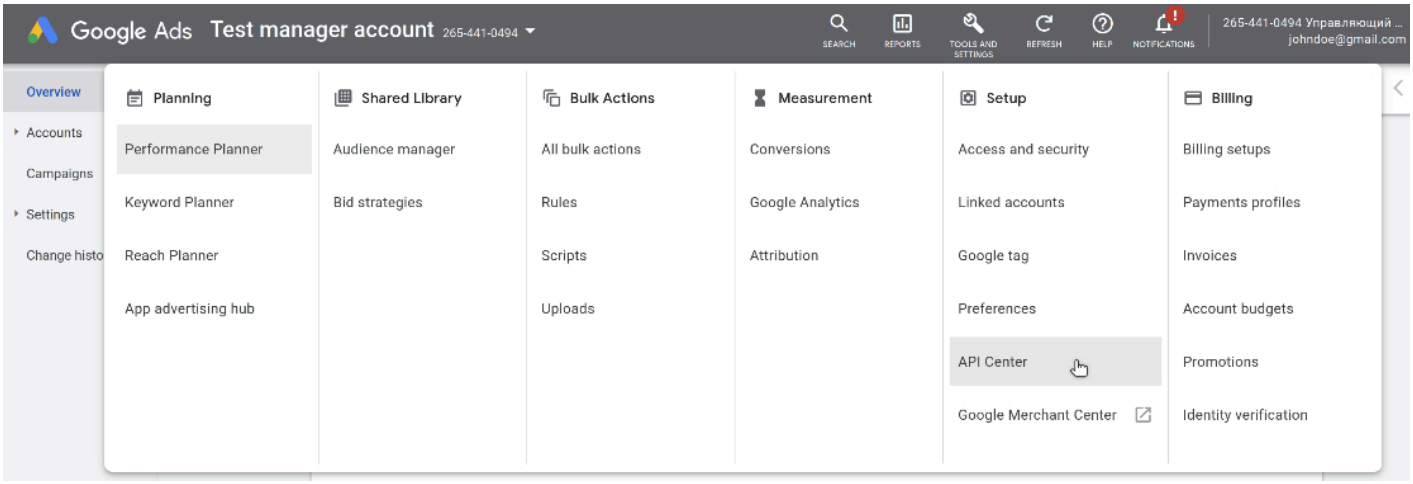
При просмотре созданного OAuth 2.0 Client ID в блоке справа вы найдете:

- **Client ID** — идентификатор приложения,
- **Client secret** — секрет.

В блоке “**Authorized redirect URIs**” добавьте **URI**, полученный в платформе Altcraft при настройке подключения. Не забудьте сохранить изменения.



Чтобы получить **токен**, войдите в управляющий аккаунт Google Ads (manager account). Далее откройте раздел **Tools and Settings** —> **API Center** и скопируйте значение из поля **Developer token**.

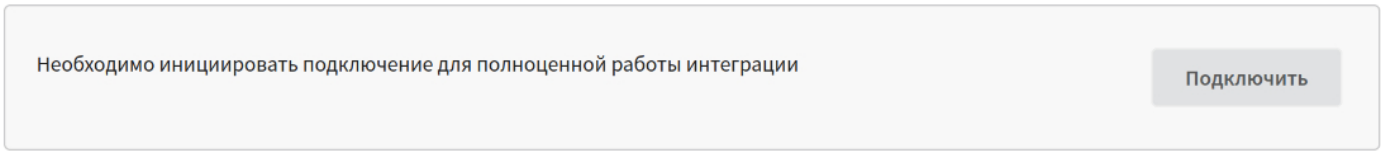


TIP

Инструкцию по созданию управляющего аккаунта можно найти по этой ссылке.
Подробнее о получении токена читайте в документации.

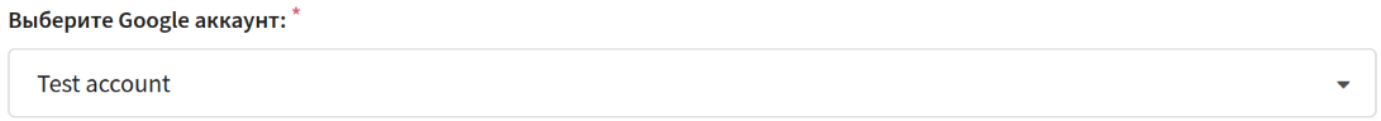
Шаг 2. Инициализация Google Ads подключения

Когда вы внесете данные Google приложения в платформу, сохраните подключение, нажав “**Применить**”. Только после этого появится кнопка “**Подключить**”. При нажатии на неё произойдет редирект в Google с предложением авторизоваться. Авторизуйтесь, чтобы подтвердить права доступа для платформы Altcraft.



Шаг 3. Выбор рекламного аккаунта Google

Если предыдущие шаги завершились без ошибок, выберите рекламный аккаунт, с которым нужно будет взаимодействовать через созданное подключение:



Статусы подключения

Статус	Описание
Готово к работе	Подключение настроено и может быть использовано для выгрузки статических сегментов в Google Аудитории.
Подключается	Происходит подключение. Возможно, вы не авторизовались в Google и не предоставили права, когда при подключении появилось дополнительное окно авторизации.
Подключено	Подключение настроено не полностью, требуются дополнительные шаги. Например, необходимо выбрать Google аккаунт.
Ошибка	В процессе работы с API возникла ошибка, которая требует переподключения. В этом случае необходимо убедиться в наличии прав доступа у пользователя.
Ошибка при подключении	Попытка настройки подключения завершилась ошибкой.

Подключение к Яндекс.Аудиториям™

Подключение к Яндекс.Аудиториям™ необходимо для настройки соответствующей интеграции.

Шаг 1. Регистрация веб-приложения в Yandex

Перейдите на страницу [Яндекс.OAuth](https://oauth.yandex.ru/) и войдите в свой аккаунт Яндекс. Перейдите в раздел **Ваши приложения** и нажмите **Создать**.	
В появившемся окне выберите **Для авторизации пользователей**.	
При необходимости пройдите верификацию через сервис Госуслуги. Это необязательный шаг, сделать это можно позже.	
Дайте вашему приложению название, загрузите иконку и укажите вашу контактную почту.	
В окне "Платформы приложений" выберите **Веб-сервисы**. В поле "URL, куда направим пользователя после того, как он разрешил или отказал приложению в доступе" добавьте следующий URL: `https://<домен_платформы>/connections/init/`, где вместо `<домен_платформы>` вставьте URL, по которому вы пользуетесь платформой Altcraft.	
Добавьте необходимые права. Для корректной работы подключения необходимо добавить: • право на **чтение параметров настройки своих и доверенных сегментов** (`audience:read` в поиске) • право на **создание сегментов, изменение параметров настройки своих и доверенных сегментов** (`audience:write` в поиске)	
Проверьте корректность данных и нажмите **Всё верно, создать приложение**. Не закрывайте страницу приложения. На следующем этапе вам понадобятся **Client ID** и **Client Secret** с этой страницы.	

TIP: Уже есть созданное приложение

Если у вас уже есть созданное приложение в Яндекс.OAuth, просто измените его настройки в соответствии с описанными выше параметрами.

Шаг 2. Создание подключения в платформе Altcraft

Перейдите в раздел **Настройки — Подключения**. Выберите **Yandex.Audience**.	

Нажмите кнопку **Создать** в правом верхнем углу экрана.	

Укажите название и описание подключения. В поле "Идентификатор приложения" укажите **Client ID** приложения, а в поле "Секрет" — **Client Secret**.	

Сохраните подключение. После этого выберите его в списке подключений и нажмите **Подключить** в нижней части меню.	

Произойдёт редирект на страницу Яндекса. Подтвердите подключение. В случае успеха, перед вами появится соответствующее сообщение.	

Подключение готово к работе. На этой странице можно прочесть о создании интеграции с Яндекс.Аудиториями™.

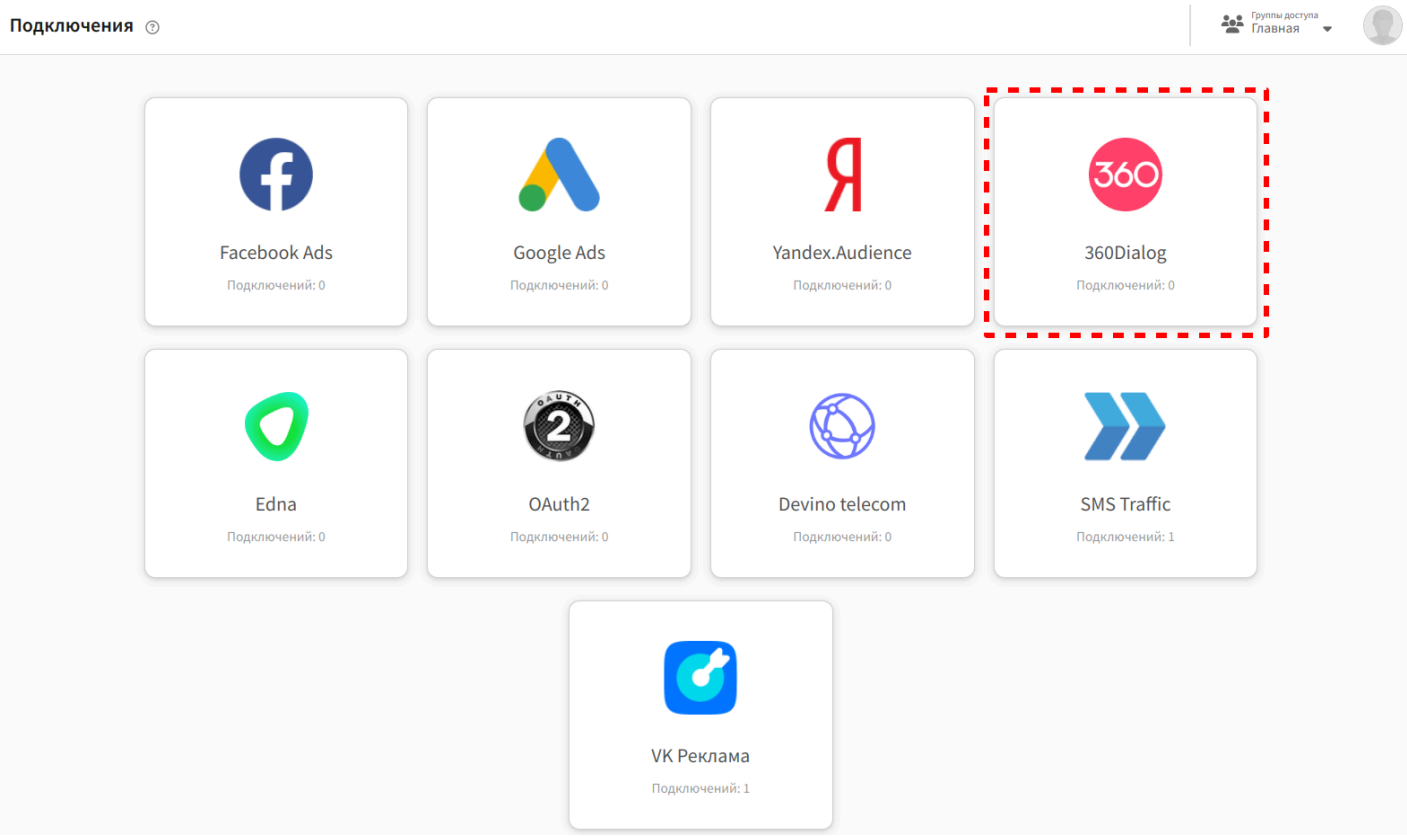
Подключение к 360dialog

Для интеграции платформы с мессенджером WhatsApp* необходимо настроить подключение к одному из провайдеров WhatsApp* Business API. В этой статье описывается, как настроить подключение к провайдеру **360dialog**.

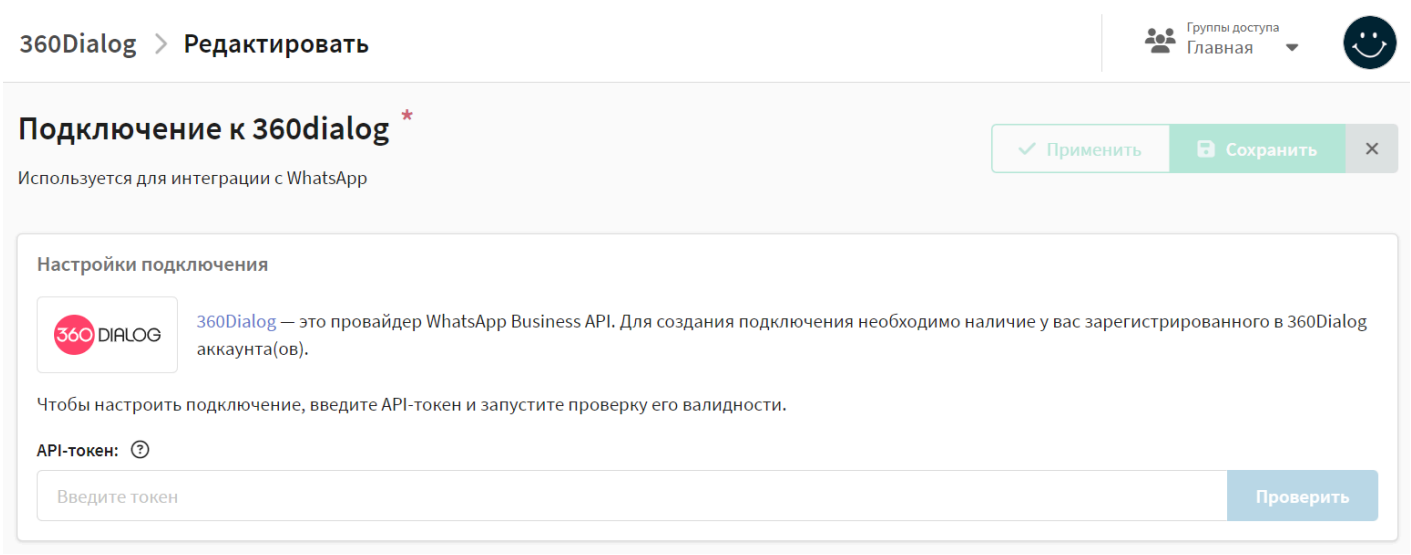
TIP

Предварительно вам необходимо подать заявку на регистрацию аккаунта в 360dialog через **эту форму**.

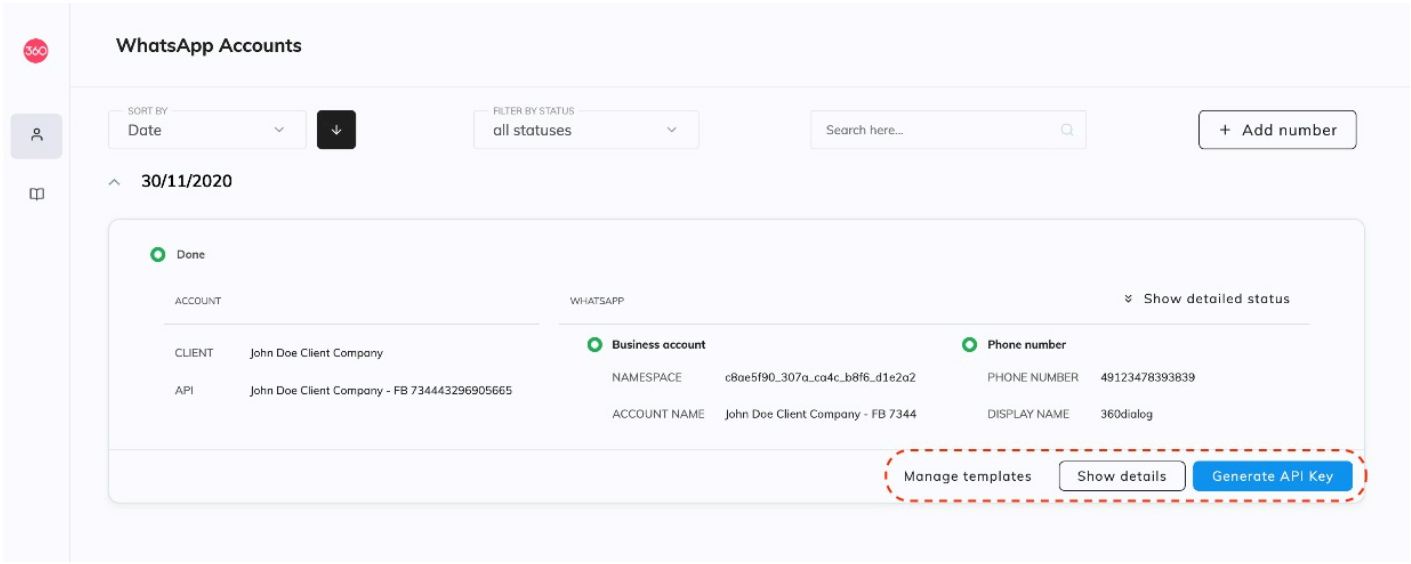
В главном меню перейдите в раздел **Настройки** —> **Подключения**. Выберите тип подключения — **360dialog**. Нажмите кнопку **“Создать”** на верхней панели.



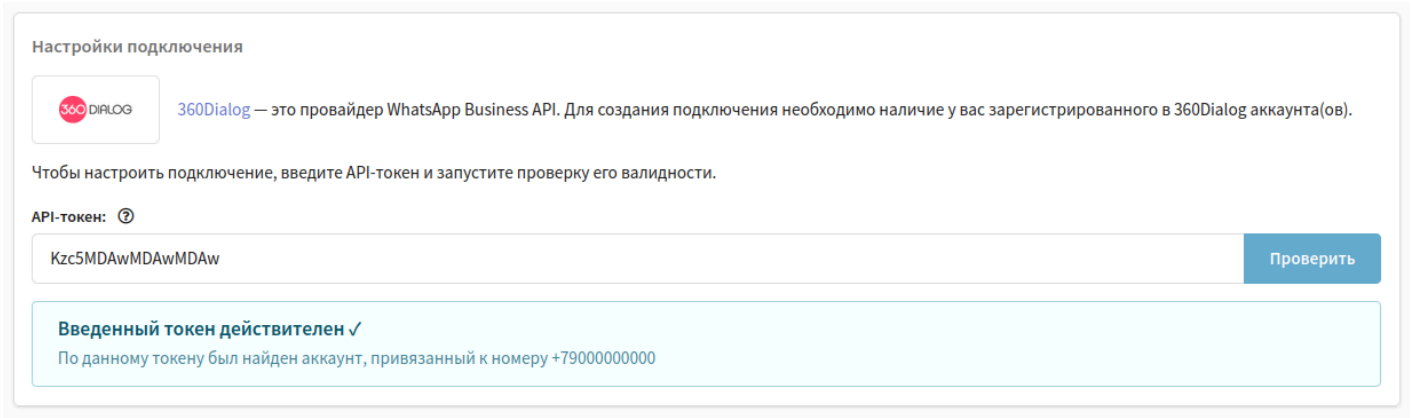
Задайте имя и описание подключения в верхней части страницы.



Для создания подключения вам понадобится **API-токен** от провайдера. Его можно найти в личном кабинете 360 Client Hub. Ключ можно сгенерировать сразу после завершения регистрации в 360dialog. Однако если вы не сохранили ключ, вы можете сгенерировать новый. Для этого перейдите в раздел *"WhatsApp Accounts"** и нажмите на кнопку **“Generate API key”**.



В настройках подключения введите **API-токен** и нажмите на кнопку **“Проверить”**, чтобы подтвердить его валидность. Если токен прошёл проверку, ниже появится информация о том, к какому номеру телефона привязан подключенный аккаунт:



Сохраните подключение, нажав на кнопку **“Сохранить”**. После этого можно переходить к **настройке интеграции с WhatsApp***.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Подключение к Edna

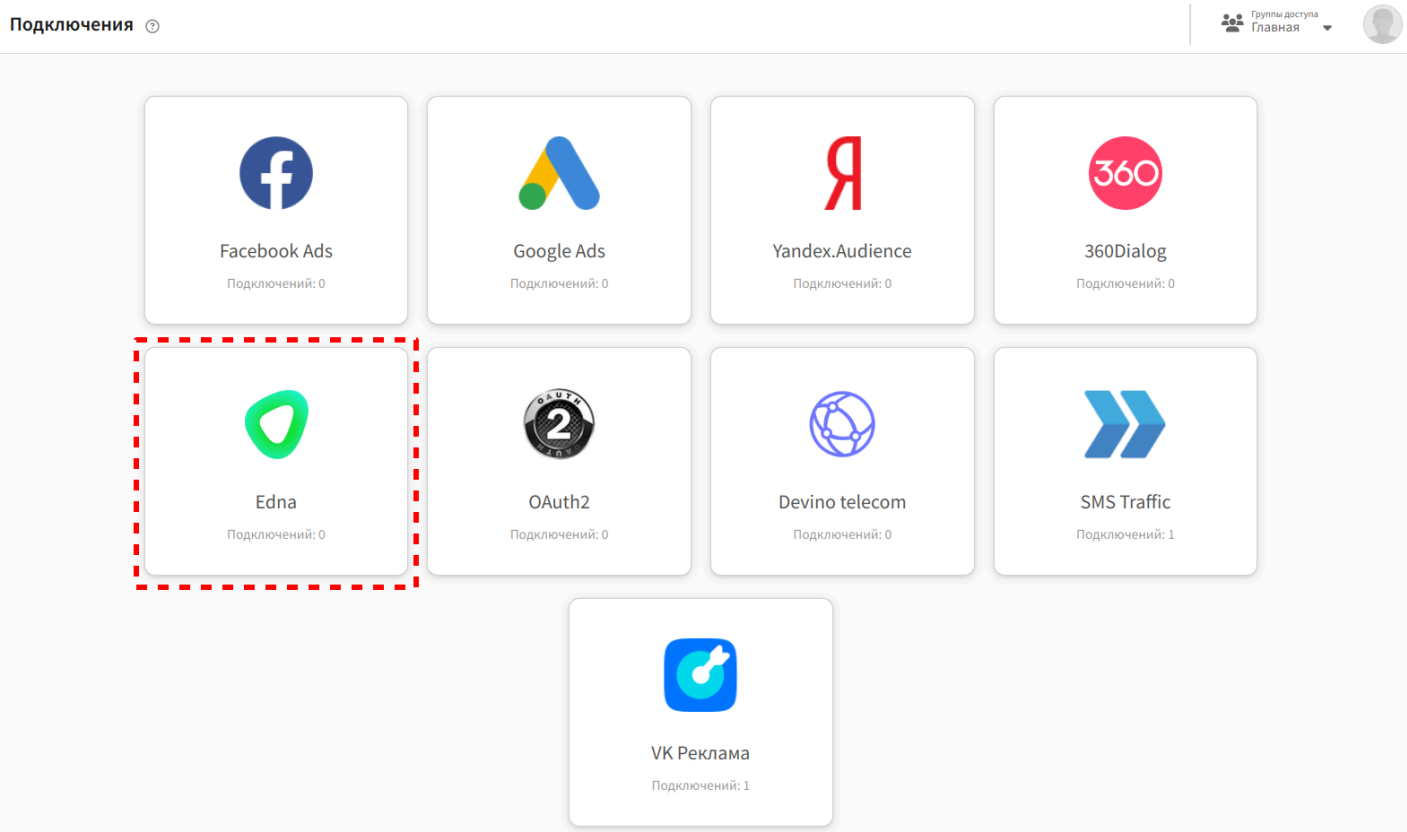
Для интеграции платформы с мессенджером WhatsApp* необходимо настроить подключение к одному из провайдеров WhatsApp* Business API. В этой статье описывается, как настроить подключение к провайдеру **Edna**.

TIP

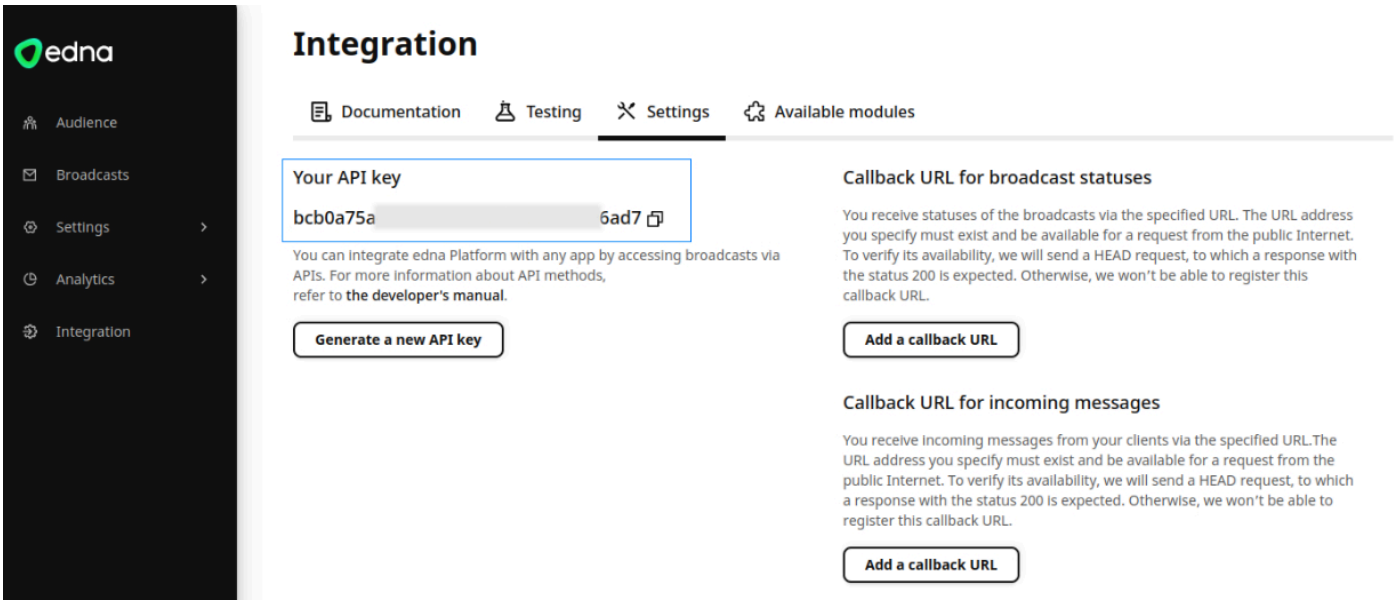
Предварительно вам необходимо подать заявку на регистрацию аккаунта в Edna через **эту форму**.

После регистрации WhatsApp* Business API, перейдите в панель управления на платформе Altcraft.

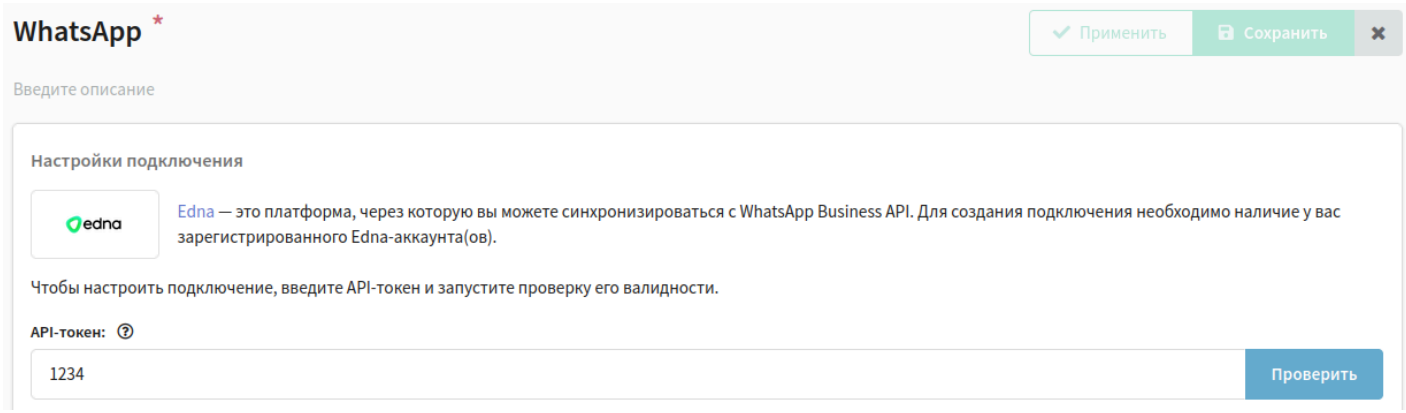
В главном меню перейдите в раздел **Настройки** —> **Подключения**. Выберите тип подключения — **Edna**. Нажмите кнопку “**Создать**” на верхней панели.



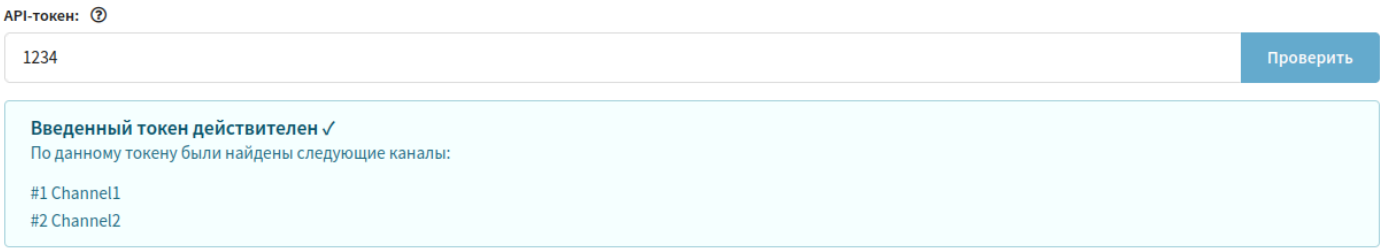
Для создания подключения вам понадобится ввести **API-токен**, сгенерированный провайдером. В личном кабинете Edna откройте раздел **“Интеграции”** и перейдите на вкладку **“Настройки”**. Скопируйте API-токен.



Введите токен и нажмите на кнопку **“Проверить”**, чтобы подтвердить его валидность.



Если токен прошёл проверку, ниже появится информация о WhatsApp*-каналах в аккаунте Edna:



Задайте имя и описание подключения в верхней части страницы. Сохраните подключение, нажав на кнопку **“Сохранить”**.

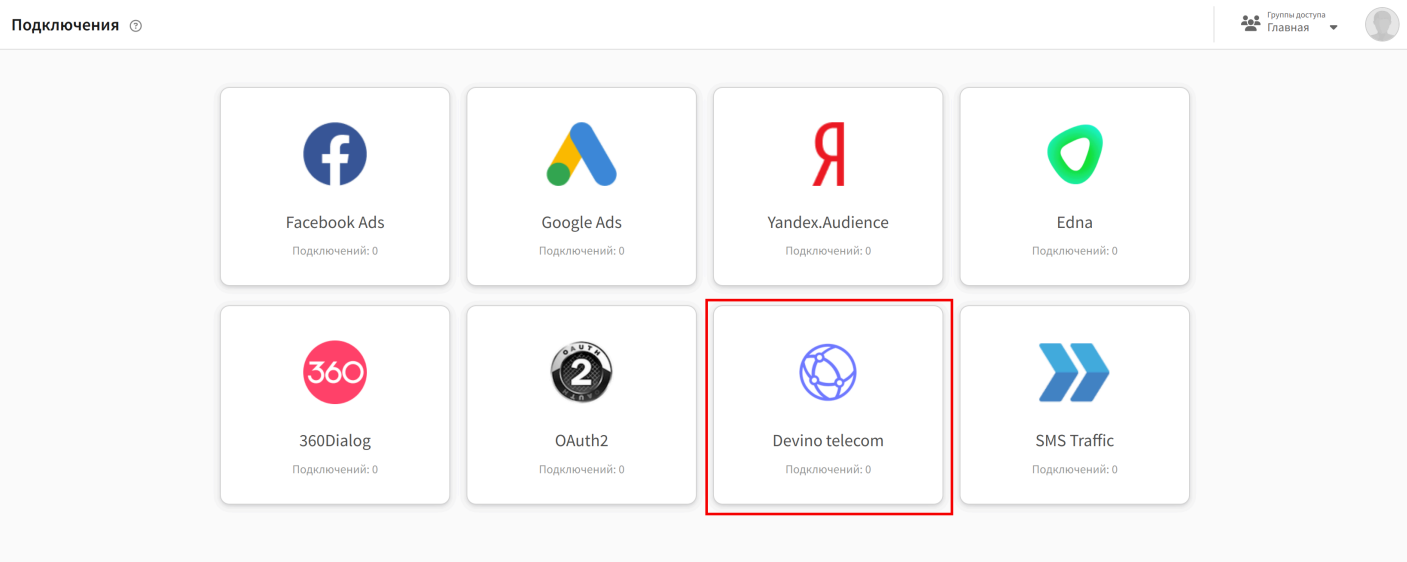
**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Подключение к Devino Telecom

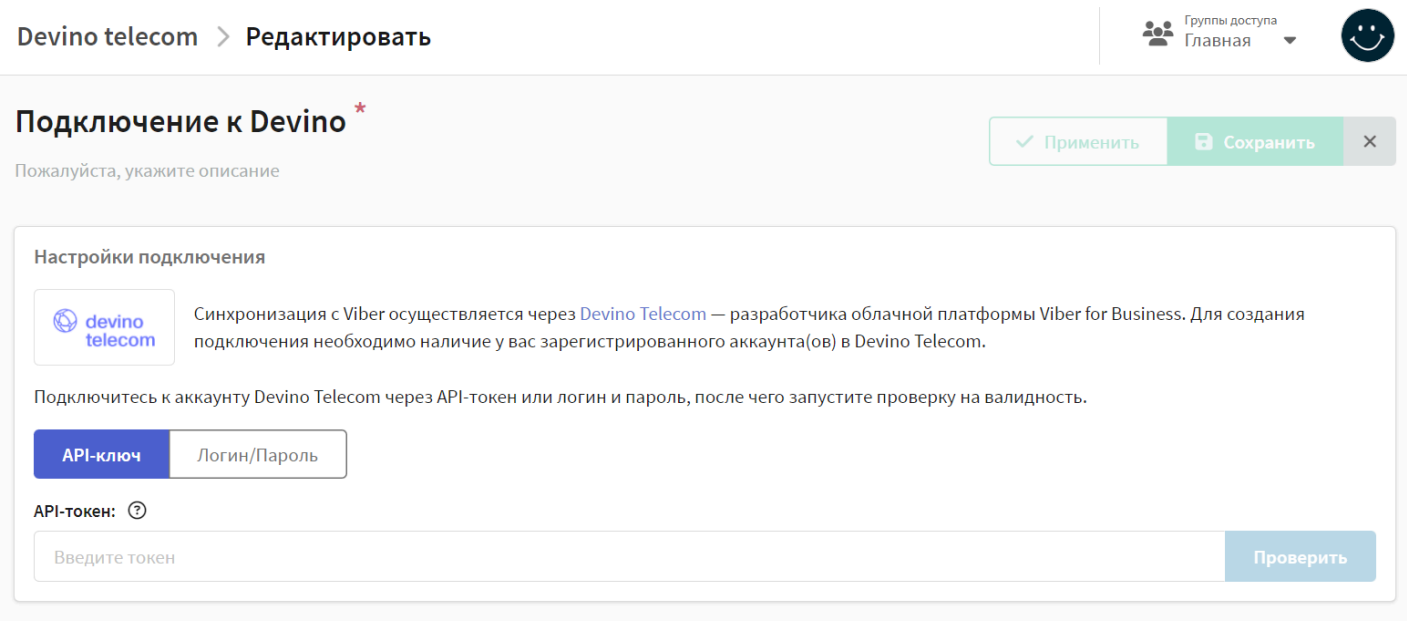
Для интеграции платформы с мессенджером Viber™ необходимо настроить подключение к одному из провайдеров. В этой статье описывается, как настроить подключение к провайдеру Devino Telecom.

Зарегистрируйсь в сервисе, а затем перейдите в панель управления на платформе Altcraft.

В главном меню перейдите в раздел **Настройки —> Подключения.** Выберите тип подключения — **Devino telecom**. Нажмите кнопку **“Создать”** на верхней панели.



Задайте системное имя подключения и при необходимости укажите описание.



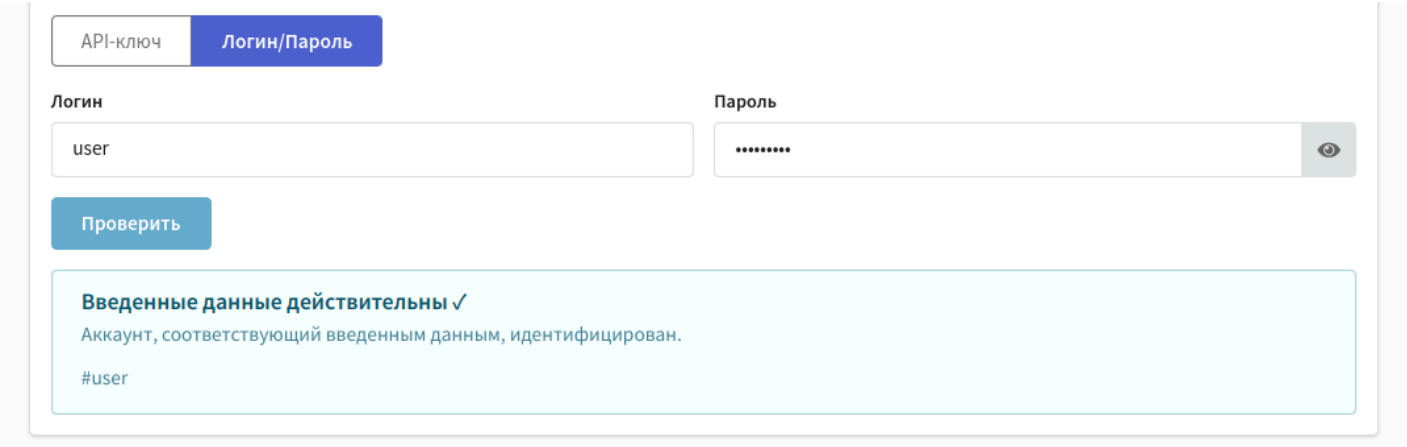
Далее выберите удобный для вас способ авторизации в аккаунте Devino:

- **API-ключ** — с помощью сгенерированного API ключа.
- **Логин и пароль** — комбинация логина и пароля учётной записи Devino.Online в кодировке Base64.

TIP

Больше информации об этих способах авторизации вы найдёте в документации Devino Telecom.

Введите данные для авторизации. Затем необходимо **проверить валидность** введенных данных. Чтобы запустить проверку, нажмите кнопку **“Проверить”**. Если данные действительны, ниже появится информация об успешно идентифицированном аккаунте.



Сохраните подключение, используя кнопки в правом верхнем углу.

---

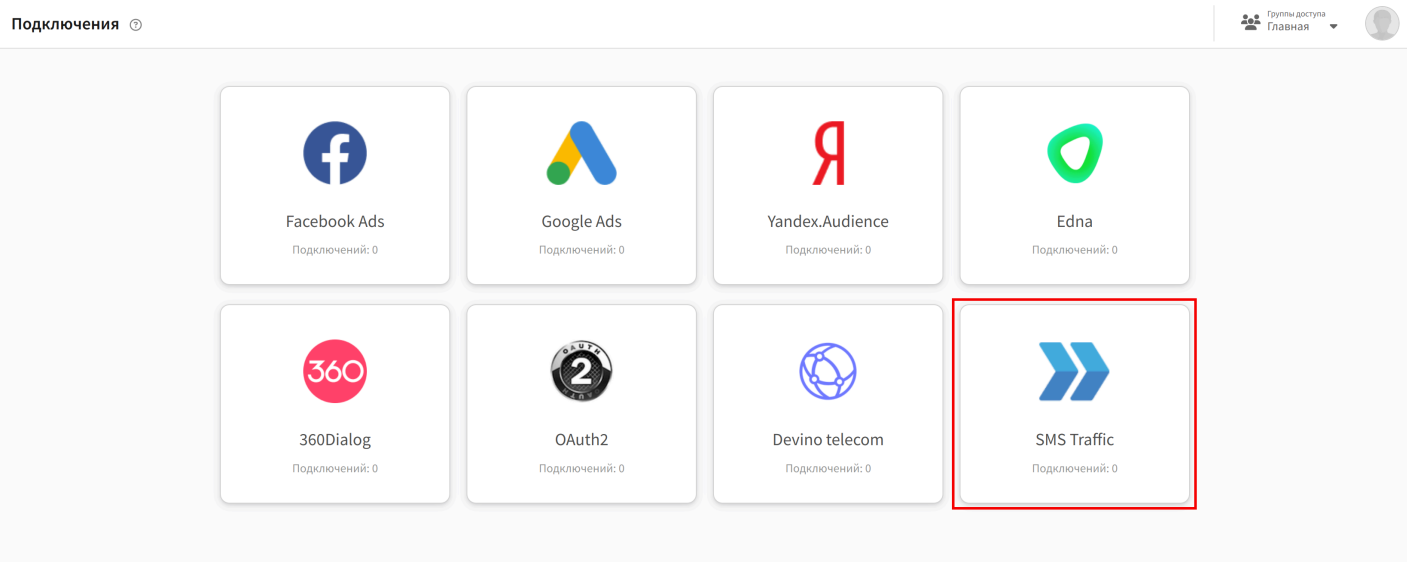
* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Подключение к SMS Traffic

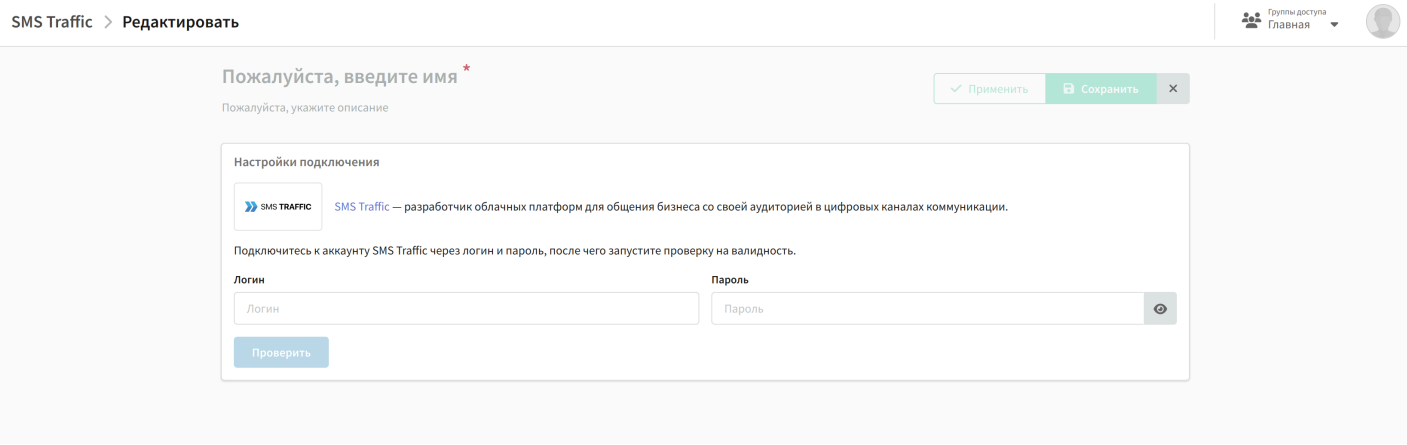
Для интеграции платформы с мессенджером Viber™ необходимо настроить подключение к одному из провайдеров. В этой статье описывается, как настроить подключение к провайдеру **SMS Traffic**.

Зарегистрируйсь в сервисе, а затем перейдите в панель управления на платформе Altcraft.

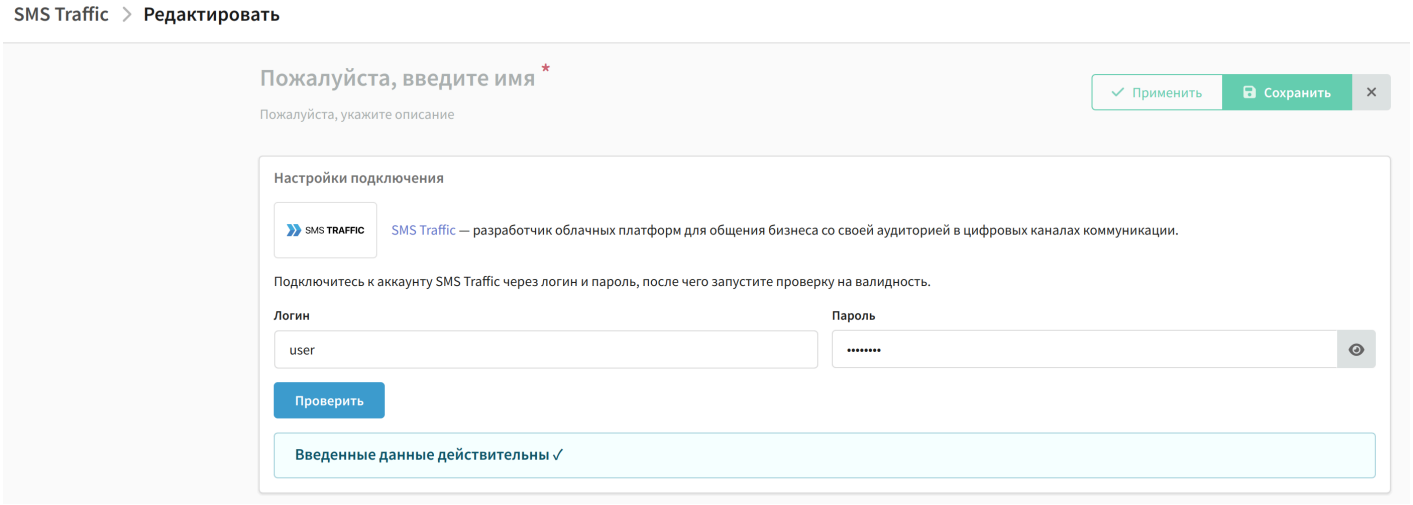
В главном меню перейдите в раздел **Настройки —> Подключения.** Выберите тип подключения — **SMS Traffic**. Нажмите кнопку **“Создать”** на верхней панели.



Задайте системное имя подключения и при необходимости укажите описание. Далее авторизуйтесь при помощи логина и пароля учетной записи SMS Traffic.



Затем необходимо **проверить валидность** введенных данных. Чтобы запустить проверку, нажмите кнопку **“Проверить”**. Если данные действительны, ниже появится информация об успешно идентифицированном аккаунте.



---

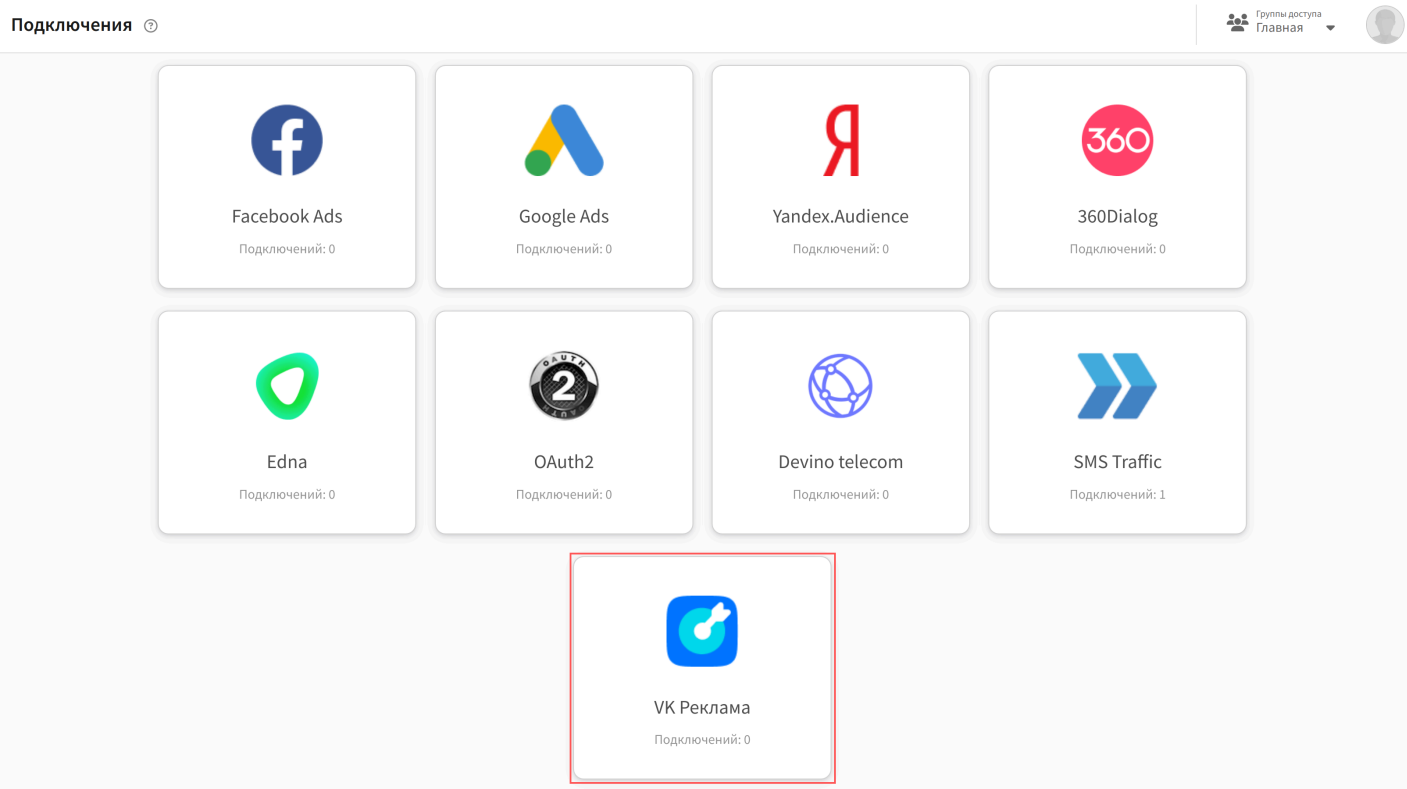
* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Подключение к VK Рекламе™

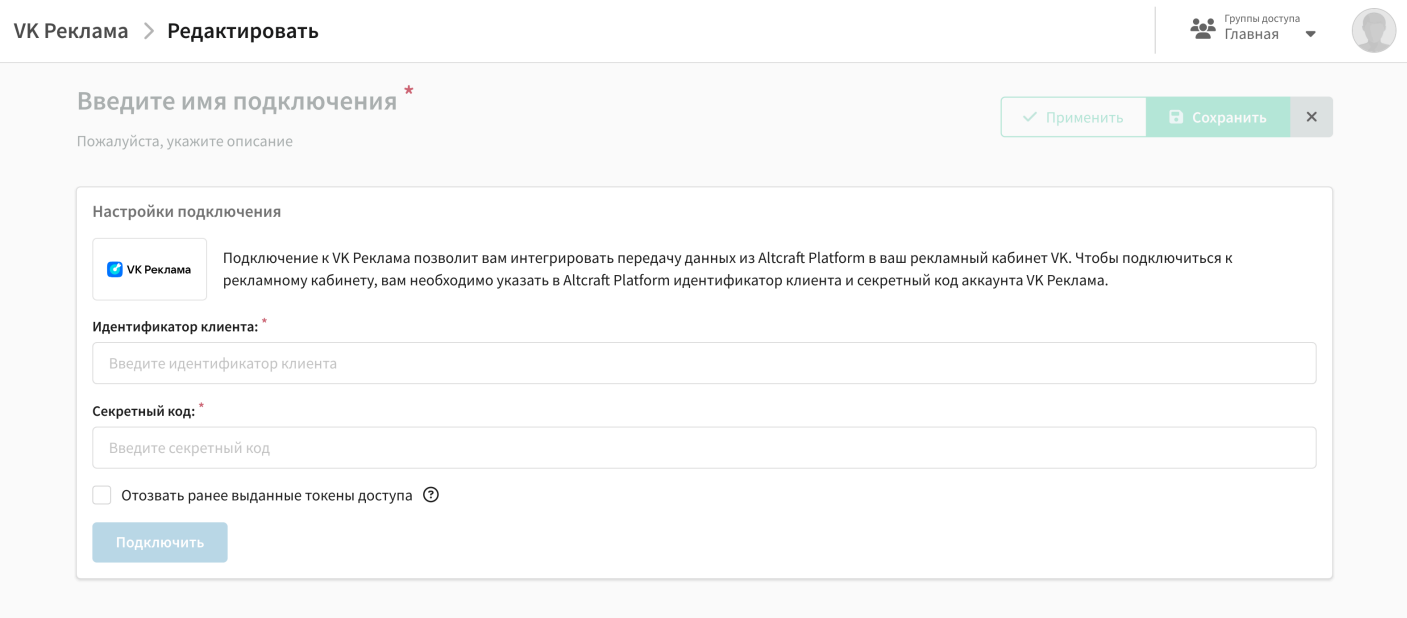
INFO

Для настройки подключения вам понадобится аккаунт в VK Рекламе™ со статусом “Рекламодатель”. Аккаунт также должен пройти модерацию в сервисе. Если вы у вас ещё нет аккаунта, зарегистрируйтесь по ссылке. Полная документация по VK Рекламе™ доступна здесь.

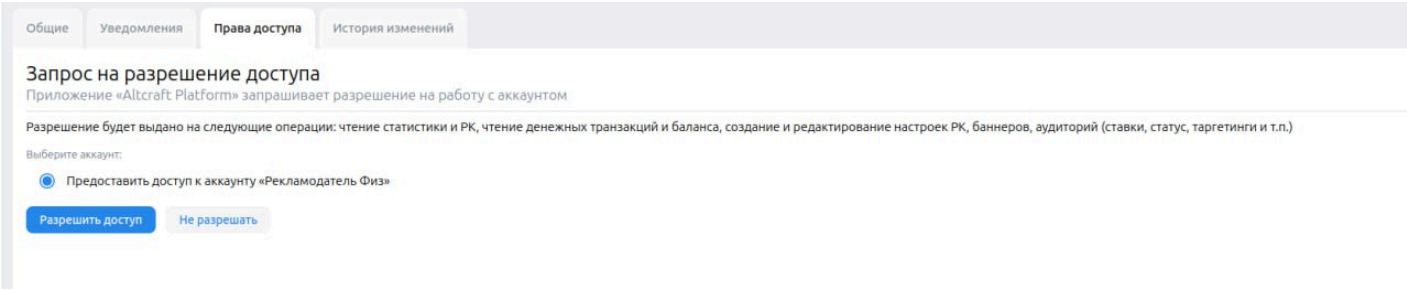
В главном меню выберите **“Настройки” —> “Подключения”**. Выберите тип подключения — **VK Реклама™**.



Откроется список всех настроенных подключений к VK. Чтобы добавить новое, нажмите **“Создать”**. Перед вами появится окно создания нового подключения. В полях в верхней части экрана укажите **название** и **описание** подключения.

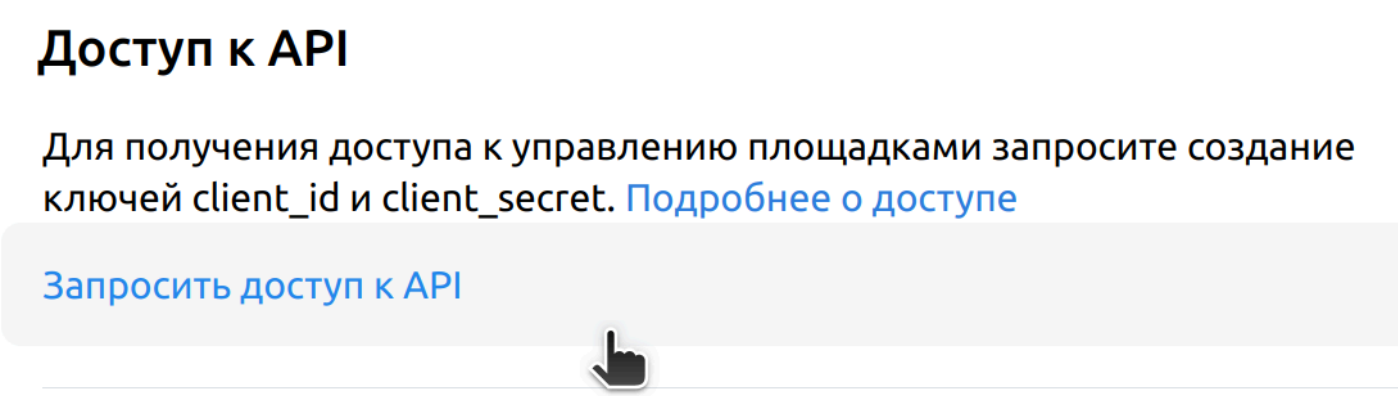


Если вы пользуетесь **облачной версией** платформы Altcraft, при попытке подключить сервис в браузере откроется страница настроек VK Рекламы™. Сервис запросит у вас разрешение для приложения Altcraft Platform. Нажмите **“Разрешить доступ”**. В списке подключений на платформе появится ваше подключение.

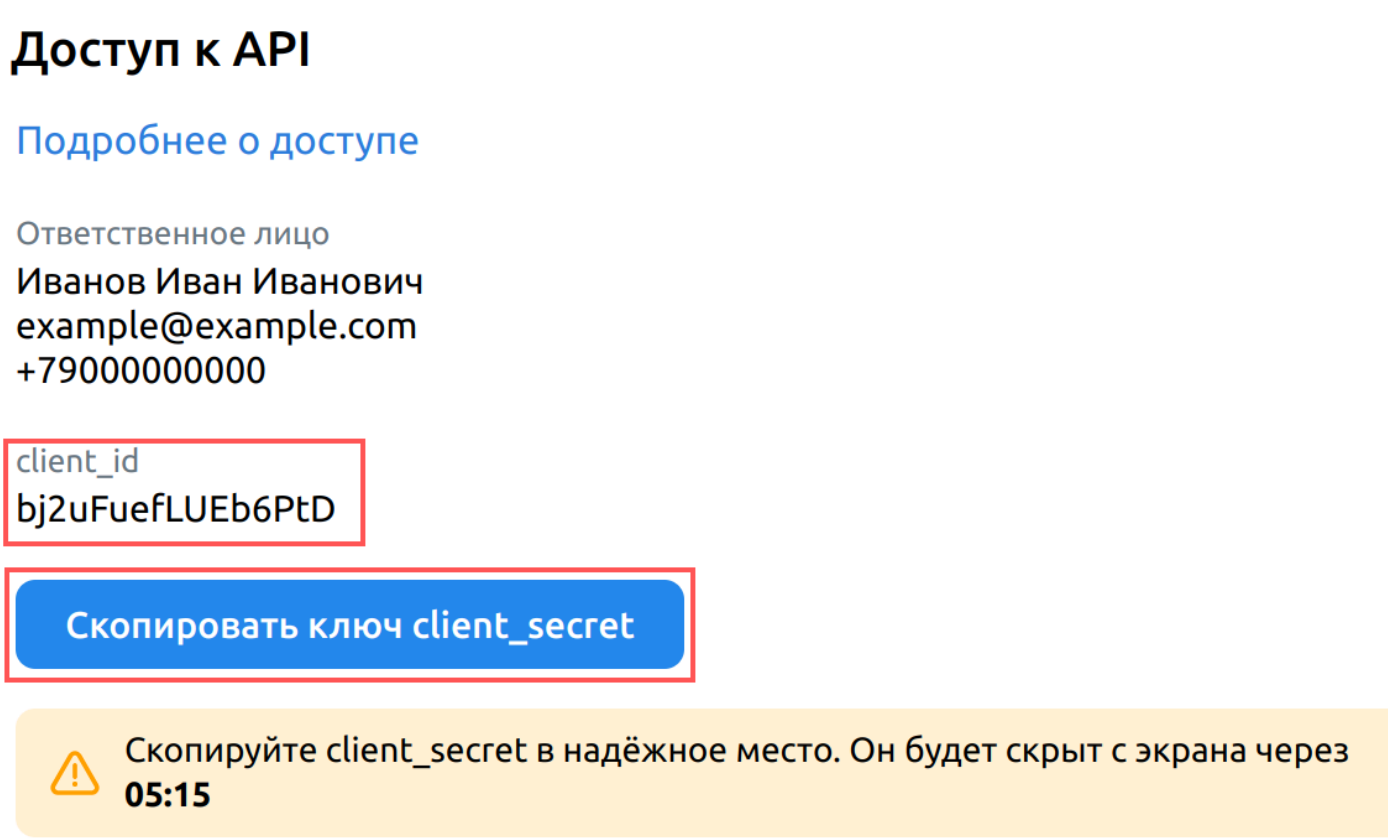


Если вы пользуетесь **on-premise версией** платформы, для подключения вам необходимо указать **Идентификатор клиента** и **Секретный код**.

Их значения можно найти в настройках рекламного кабинета VK на вкладке **“Общие”** в разделе **“Доступ к API”**. Для того, чтобы их получить, необходимо нажать на кнопку **“Запросить доступ к API”**:



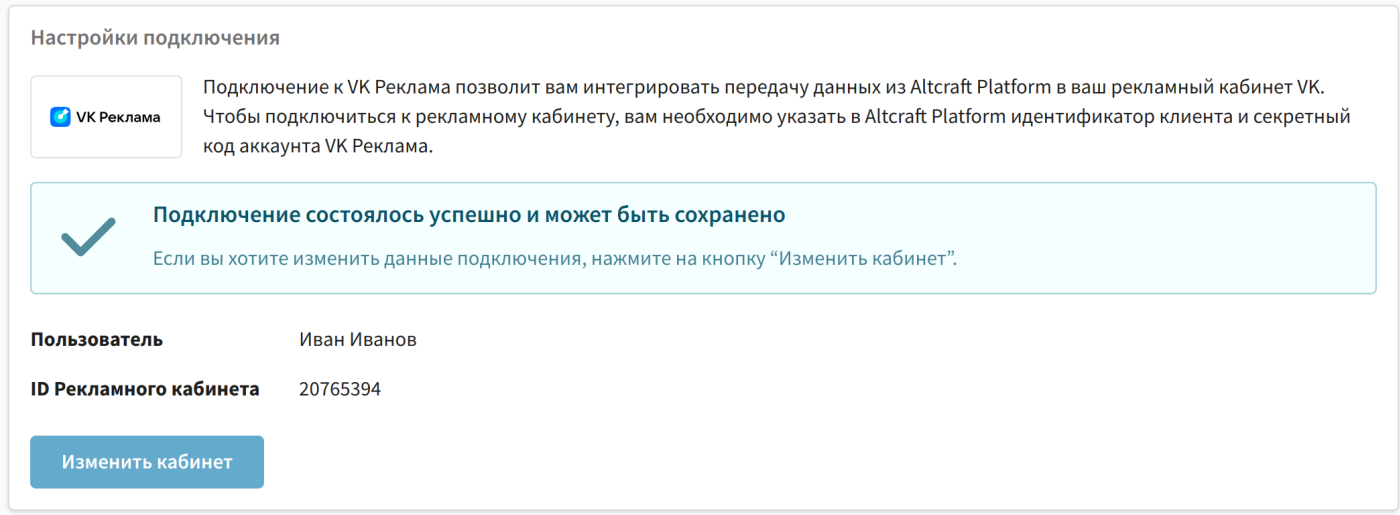
После того, как вы подтвердите контактную информацию о себе, вы получите доступ к своему **client_id**, он используется на платформе Altcraft для поля “Идентификатор клиента”. Чтобы скопировать значение **client_secret** для поля “Секретный ключ”, нажмите на соответствующую кнопку. **Сохраните его**, через 10 минут эта кнопка исчезнет и вы больше не сможете его узнать:



Когда все поля будут заполнены, нажмите кнопку **“Подключить”**. В случае успеха, на экране появится подтверждение. Если вместо этого появляется сообщение об ошибке, проверьте правильность введенных данных и попробуйте снова.

INFO

Каждый раз, когда вы производите успешную попытку подключения к VK Рекламе™, платформа создает **токен доступа**. Даже если вы не сохраняете это подключение, токен остается в системе. Количество токенов **не может быть больше 5**. Если система не позволяет вам создать ещё одно подключение, активируйте параметр **“Отозвать раннее выданные токены доступа”** при создании подключения.



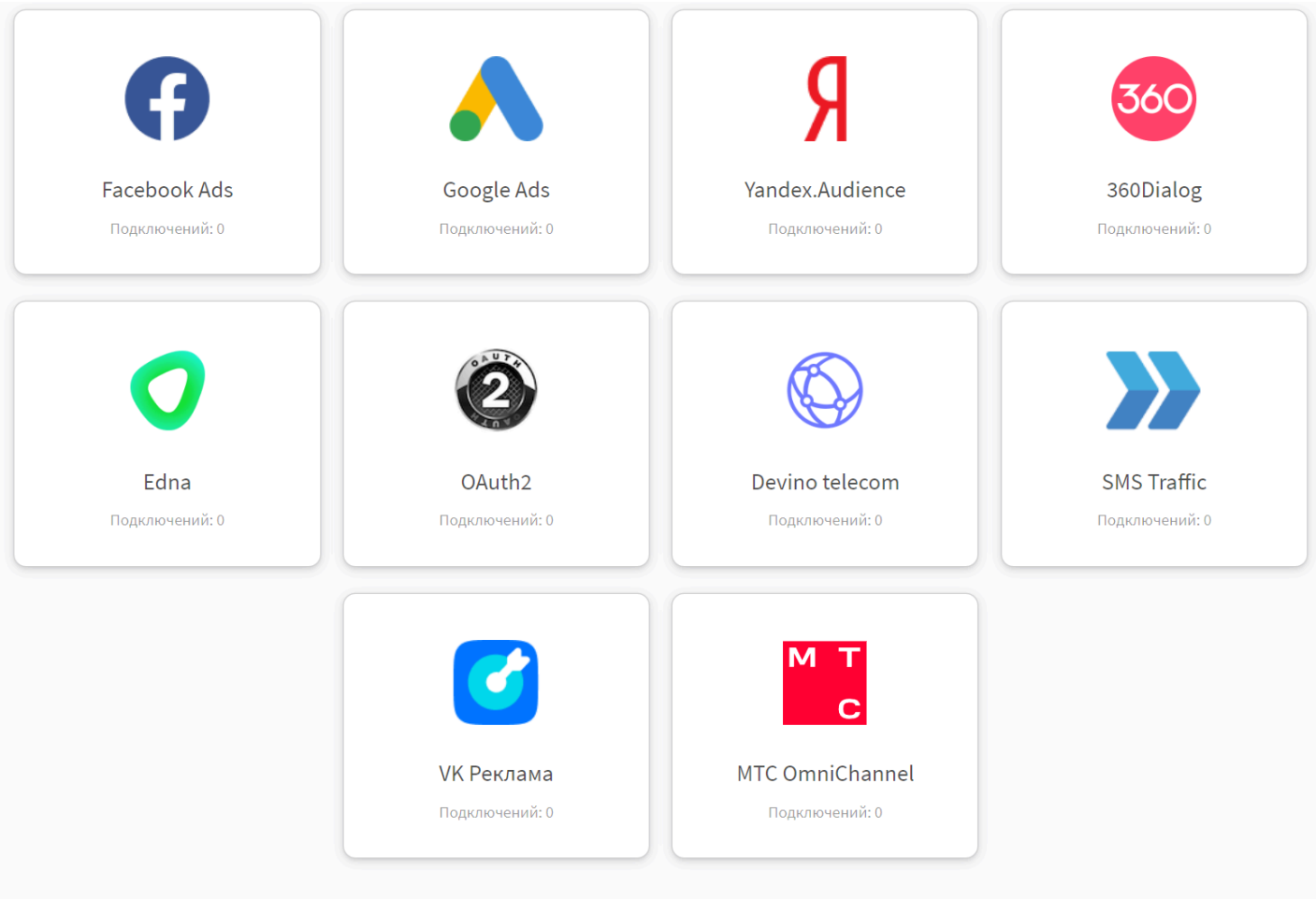
Если соединение было установлено, нажмите кнопку **“Сохранить”**. После сохранения подключения вы можете переходить к настройке интеграции.

Подключение к MTS OmniChannel

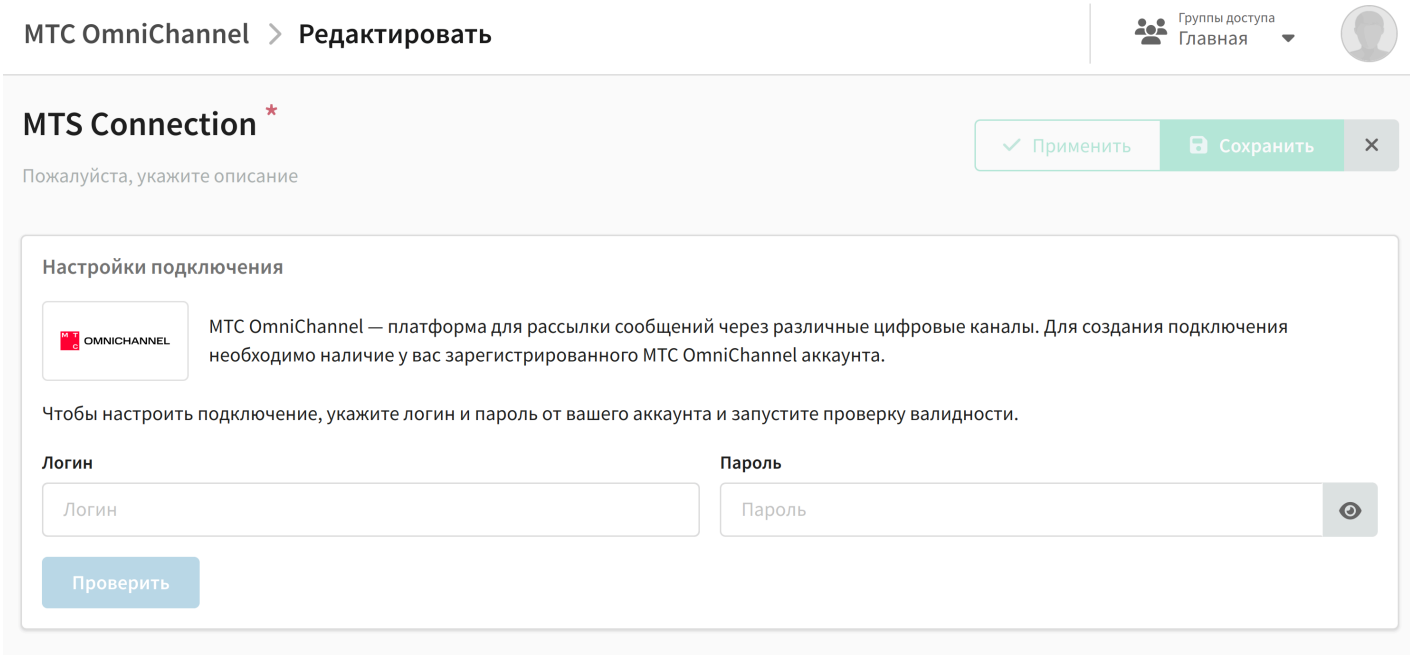
Для интеграции платформы с мессенджером Viber™ необходимо настроить подключение к одному из провайдеров. В этой статье описывается, как настроить подключение к провайдеру **MTS OmniChannel**.

Зарегистрируйте аккаунт на платформе MTS OmniChannel, а затем перейдите в панель управления платформы Altcraft.

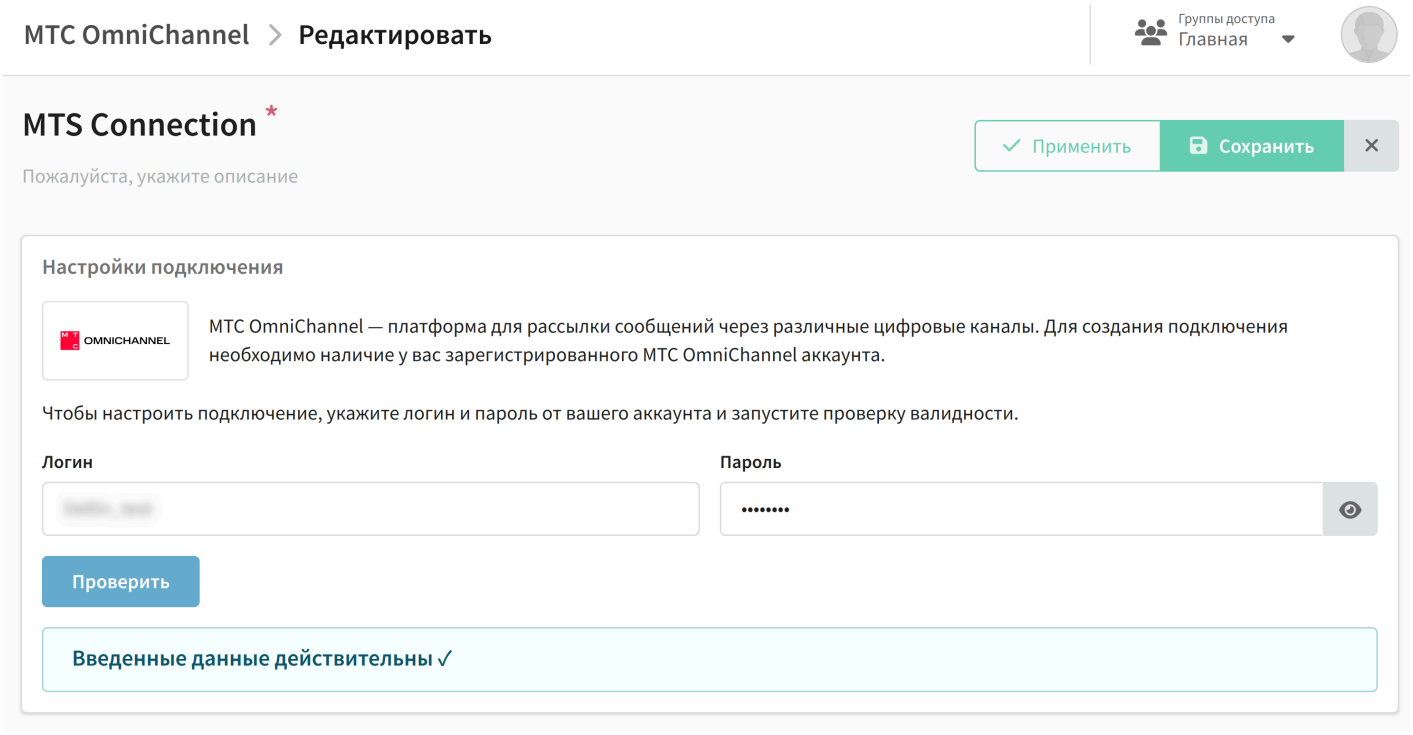
В главном меню перейдите в раздел **Настройки** —> **Подключения**. Выберите тип подключения — **MTS OmniChannel**. Нажмите кнопку **“Создать”** на верхней панели.



Задайте системное имя подключения и при необходимости укажите описание. Далее авторизуйтесь при помощи логина и пароля учетной записи MTS OmniChannel.



Затем необходимо **проверить валидность** введенных данных. Чтобы запустить проверку, нажмите кнопку **“Проверить”**. Если данные действительны, ниже появится информация об успешно идентифицированном аккаунте.



Сохраните подключение, нажав на кнопку **“Сохранить”**. После этого можно переходить к [настройке интеграции с Viber™](#).

---

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Подключение через OAuth2

Описание

OAuth 2.0 — это протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. При этом пользователю не надо доверять приложению логин и пароль.

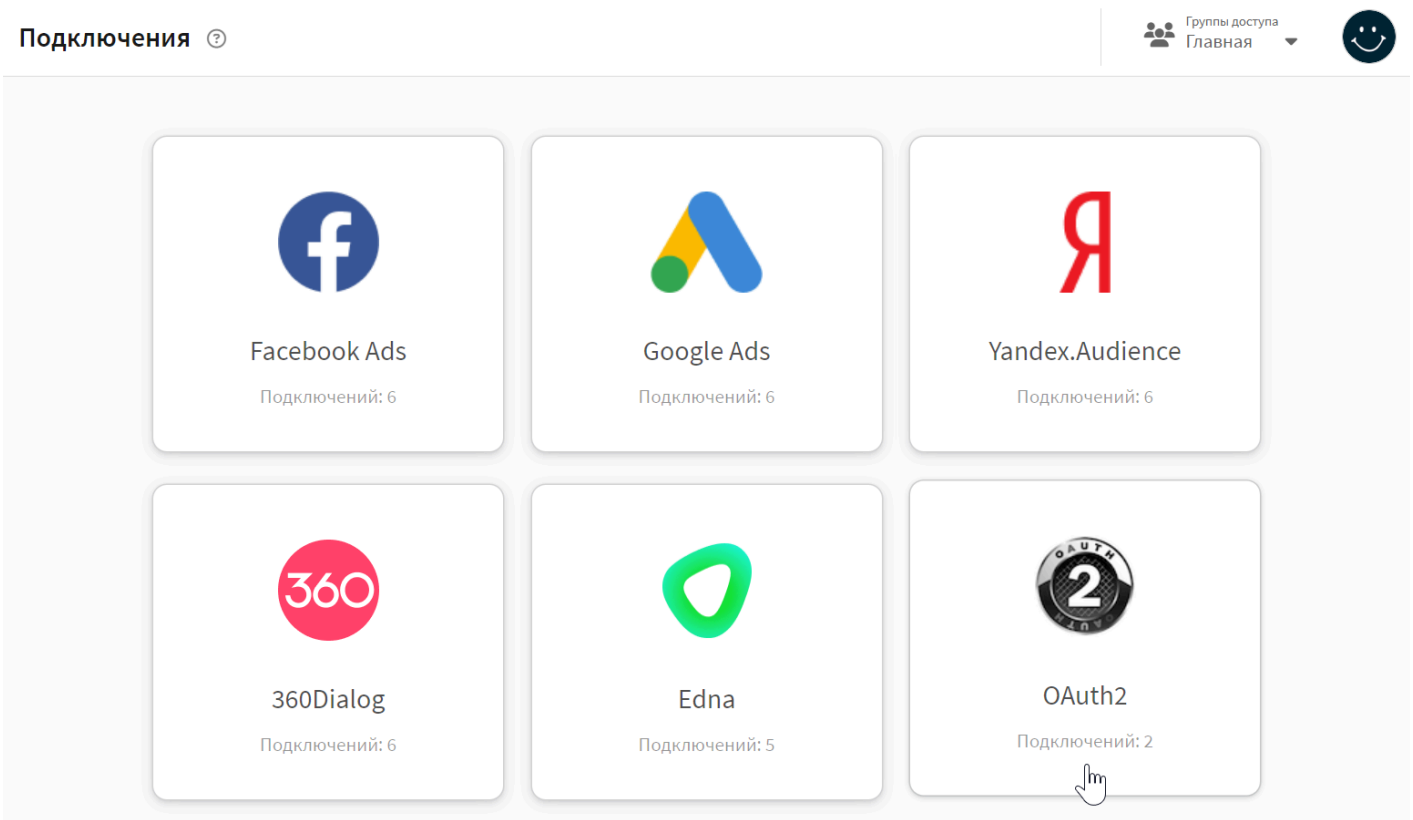
С помощью этого типа подключения вы можете связать платформу Altcraft с любым внешним сервисом, который в качестве метода авторизации использует OAuth2. Например, с Facebook*, Яндекс приложением или CRM-системой.

Где используется подключение:

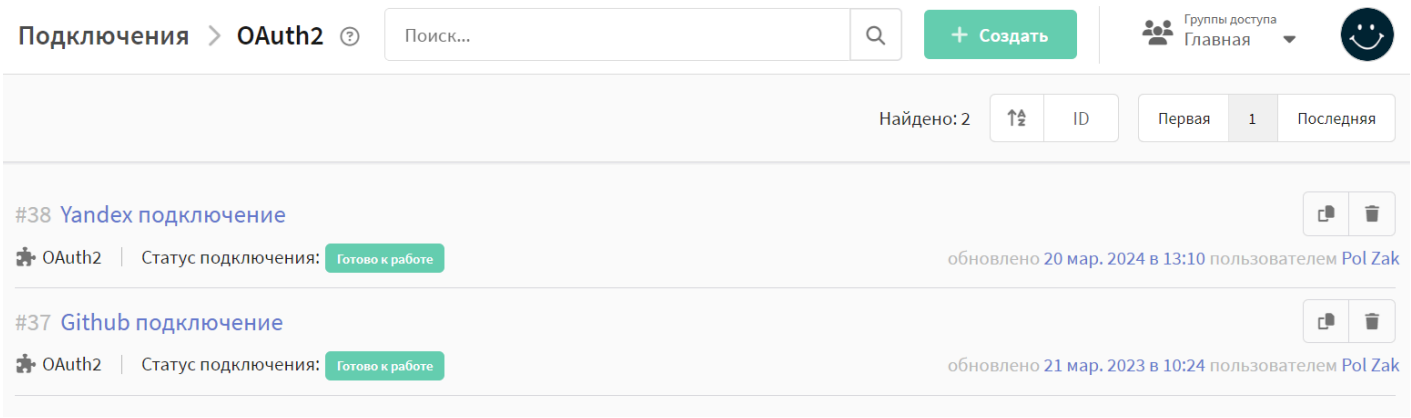
- **Узел “Вызов API”**. С его помощью можно передавать в подключенный сервис информацию из сценария или, наоборот, запрашивать из сервиса необходимый для подстановки API контент. Удобно, если нужно передать в CRM-систему данные профиля, отправленного в сценарий.
- **Action Hooks.** Механизм захвата событий позволит передать во внешний сервис информацию о событии, а также данные профиля.
- **Импорт данных по расписанию в Маркете.** Для доступа к защищенному товарному фиду доступна авторизация через OAuth2.

Создание подключения

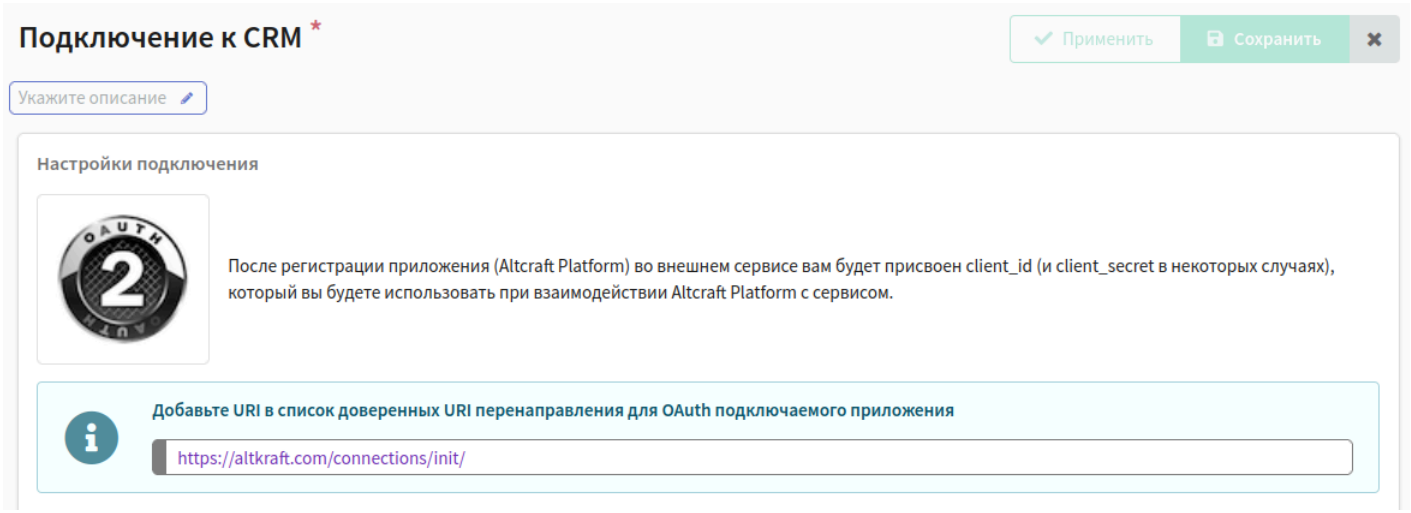
Перейдите в раздел **“Настройки”** —> **“Подключения”** и выберите тип подключения **OAuth 2**:



Откроется список всех настроенных OAuth2-подключений. Чтобы добавить новое, нажмите **“+ Создать”**.



Задайте **имя** нового подключения и при необходимости добавьте **описание:**



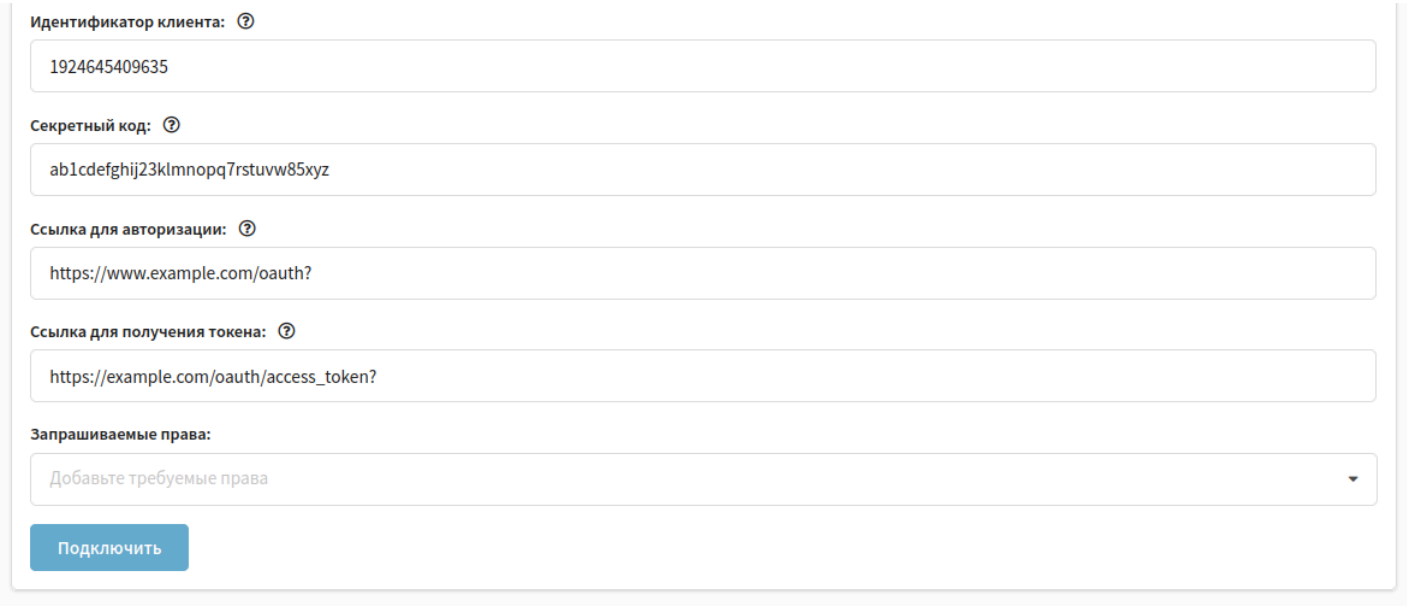
Платформа предоставит **URI перенаправления** — URL адрес, на который будет перенаправлен пользователь, когда предоставит Altcraft разрешение на доступ к своим данным в подключаемом приложении. Скопируйте URI и добавьте его в список доверенных URI перенаправления для приложения.

Далее укажите данные вашего приложения, необходимые, чтобы платформа могла получить доступ к его ресурсам. Эти данные вы получаете при регистрации приложения и можете найти в его настройках:

- **Идентификатор клиента** (client_id) — это общедоступный идентификатор приложений. Данное поле является обязательным для заполнения.
- **Секретный код** (client_secret) — пароль приложения, известный только приложению и серверу авторизации. Некоторые сервисы не выдают секретный код для проведения OAuth авторизации — в таком случае оставьте это поле пустым.

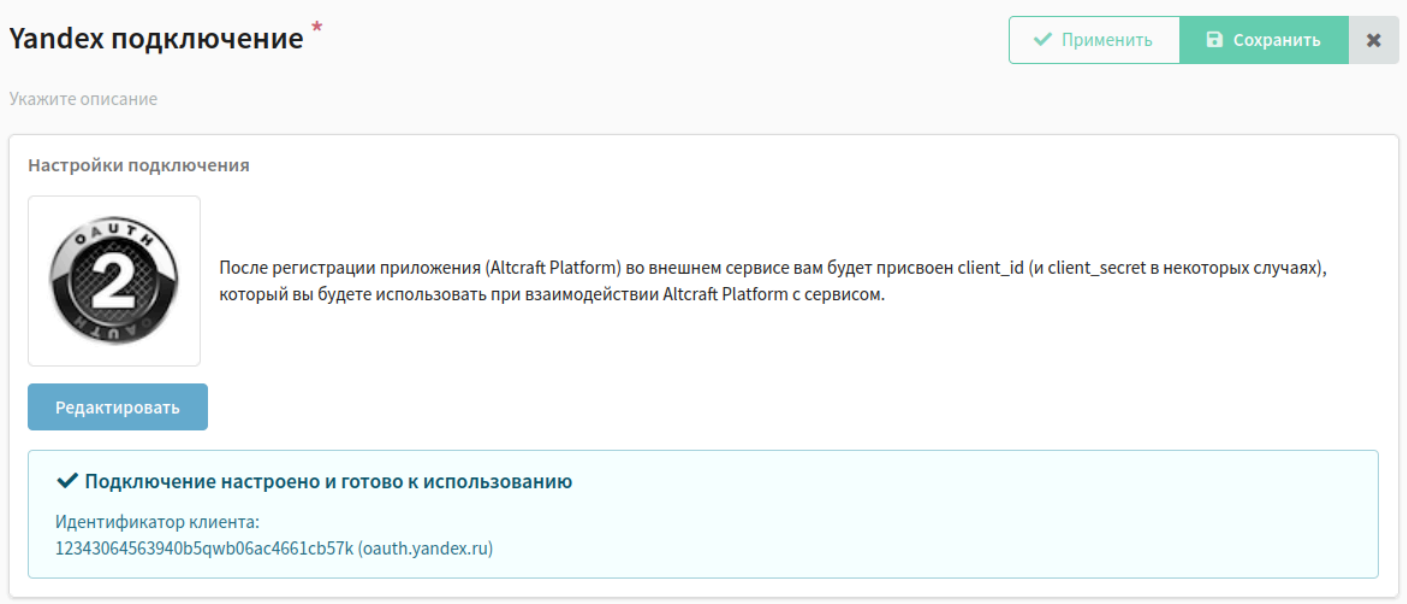
Также для OAuth операций требуется указать **ссылку для авторизации** и **ссылку для получения токена**. Чтобы определить, какие ссылки отвечают за эти операции, изучите документацию подключаемого сервиса.

При необходимости вы можете ограничить доступ платформы к данным, выбрав **запрашиваемые права:**



Нажмите **“Подключить”.** Вас перенаправят в подключаемый сервис, где необходимо авторизоваться и подтвердить выдачу прав платформе.

Если подключение к сервису завершилось успешно, **сохраните подключение** с помощью кнопок в правом верхнем углу.



Настройки подключения можно изменить в любой момент. Для этого нажмите **“Редактировать”**.

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Подключение через Basic Authentication

Описание

Basic Authentication — это протокол HTTP-аутентификации, использующий логин и пароль для получения доступа к защищенной информации. Это один из самых простых и распространенных методов авторизации на сайтах и серверах.

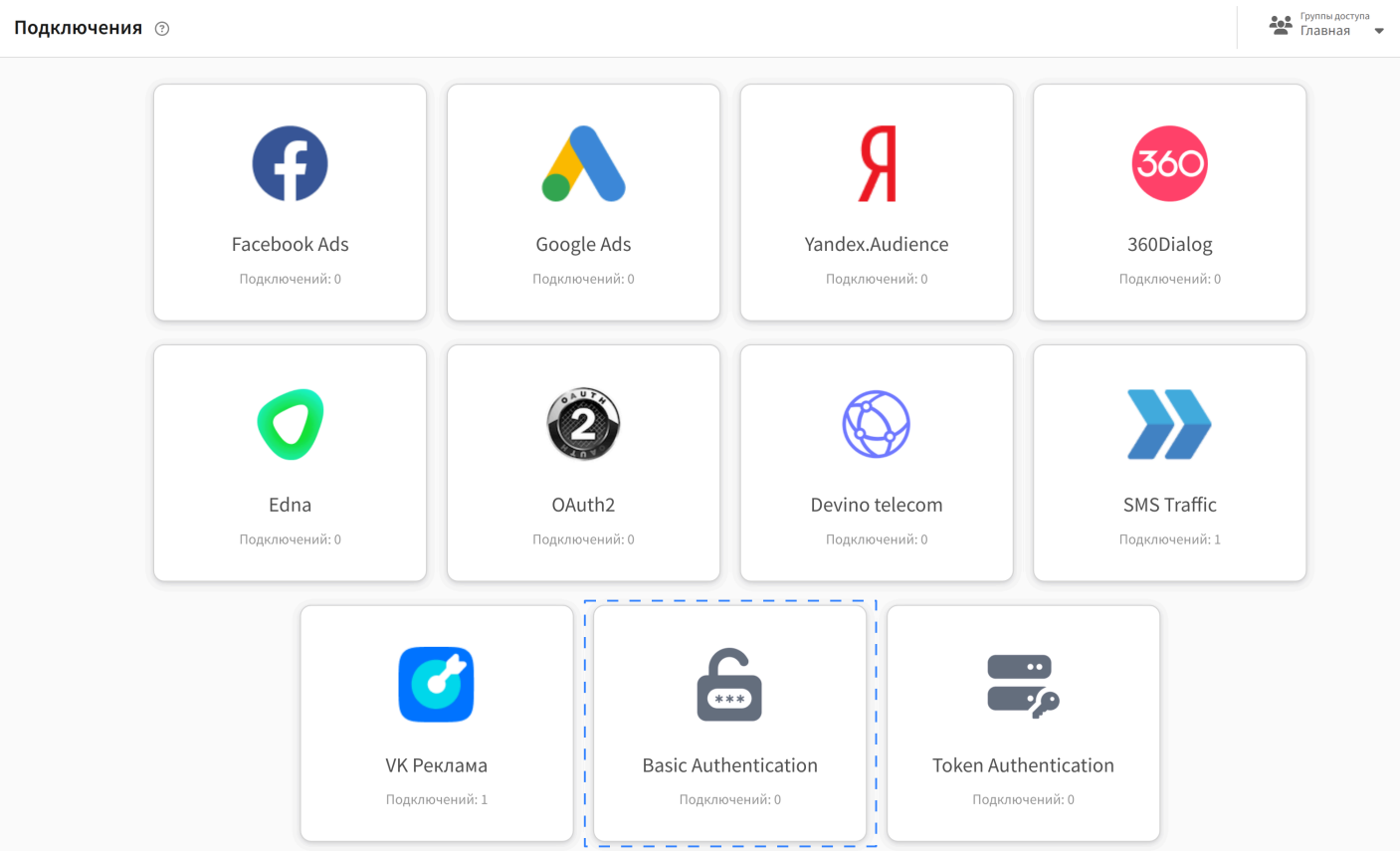
С помощью этого типа подключения вы можете связать платформу Altcraft с любым внешним сервисом, который в качестве метода авторизации использует Basic Authentication.

Где используется подключение:

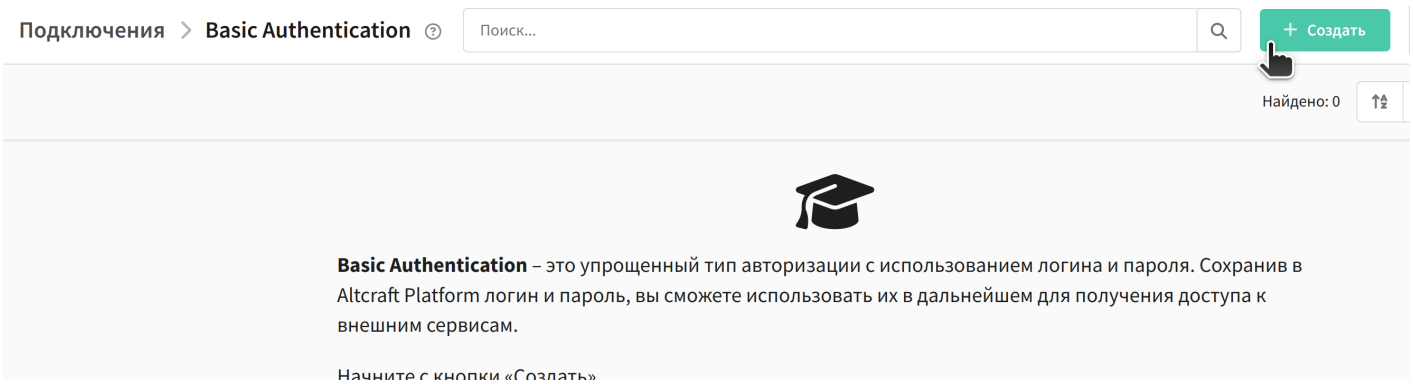
- **Узел сценария “Вызов API”**. С его помощью можно передавать в подключенный сервис информацию из сценария или, наоборот, запрашивать из сервиса необходимый для подстановки API-контент. Удобно, если нужно передать в CRM-систему данные профиля, отправленного в сценарий.
- **Action Hooks**. Механизм захвата событий позволит передать во внешний сервис информацию о событии, а также данные профиля.
- **Импорт данных по расписанию в Маркете**. Для доступа к защищенному товарному фиду доступна авторизация через логин и пароль.

Создание подключения

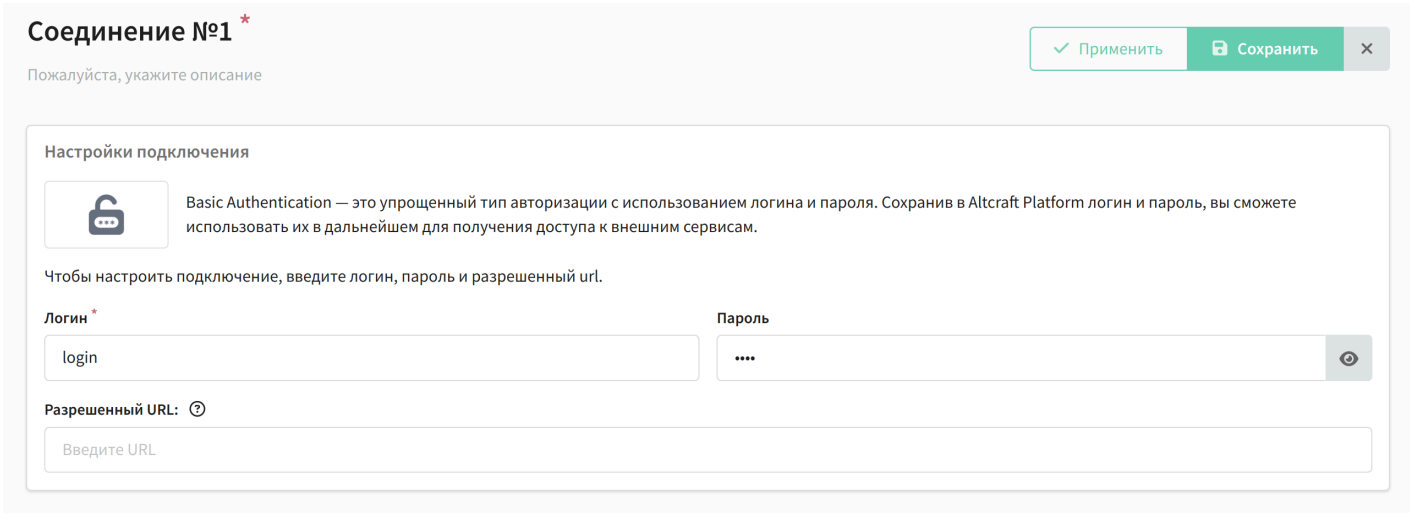
Перейдите в раздел **Настройки — Подключения** и выберите тип подключения **Basic Authentication**:



Откроется список всех настроенных Basic-подключений. Чтобы добавить новое, нажмите **+ Создать**.



Задайте **имя** нового подключения и при необходимости добавьте **описание**. Введите **логин** и **пароль**, которые будет использовать подключение. Если вы хотите предоставить доступ только для определенного сервиса, укажите его URL в поле **Разрешенный URL**:



Сохраните подключение с помощью кнопок в правом верхнем углу. Настройки подключения можно изменить в любой момент. Для этого нажмите **Редактировать**.

Подключение через Token Authentication

Описание

Token Authentication — это протокол аутентификации, использующий специальный Bearer-токен — тип авторизационного токена, который предоставляется пользователю после успешной аутентификации для доступа к защищенным ресурсам. Данный способ надежднее, чем авторизация с помощью логина и пароля.

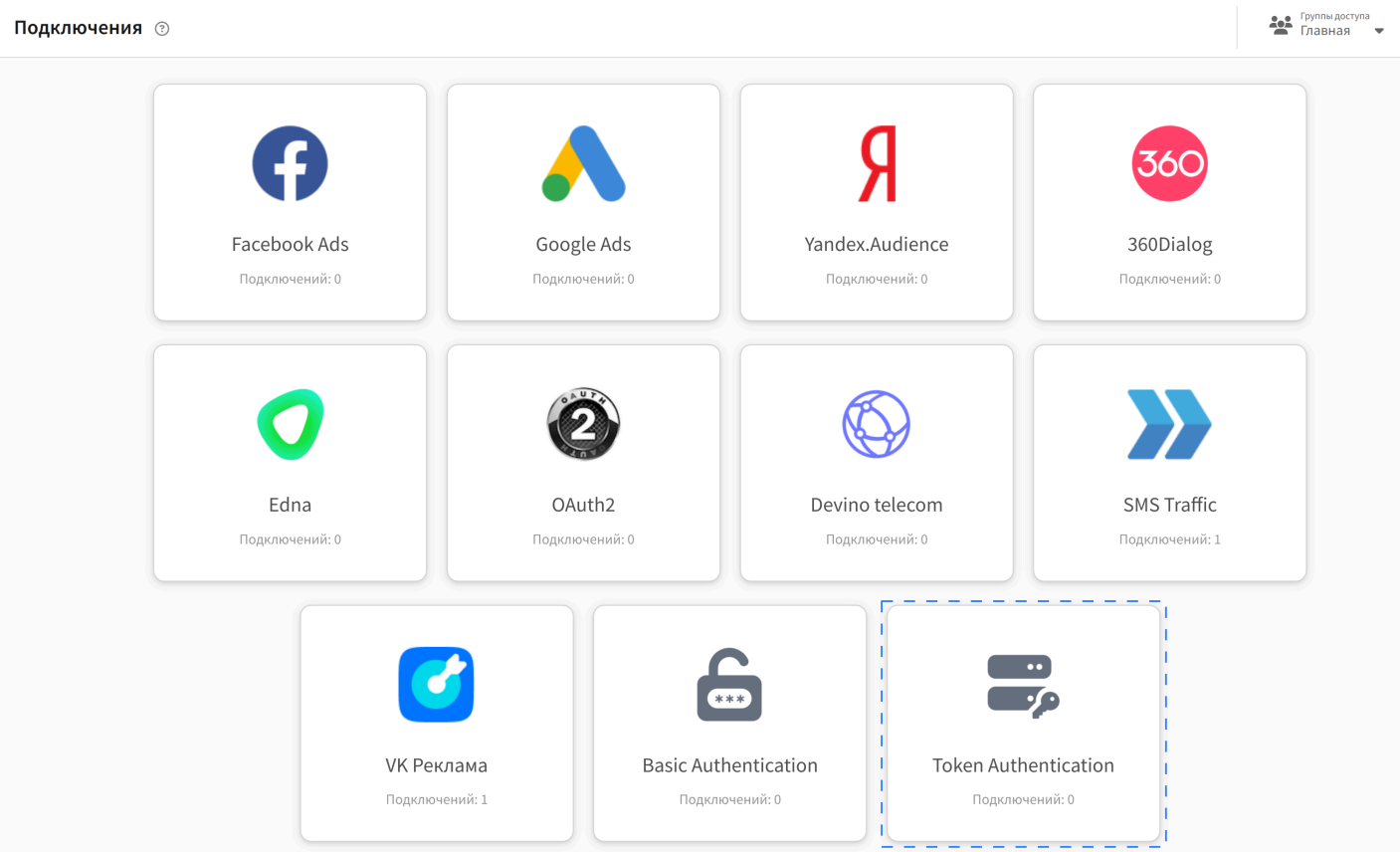
С помощью этого типа подключения вы можете связать платформу Altcraft с любым внешним сервисом, который в качестве метода авторизации использует Token Authentication.

Где используется подключение:

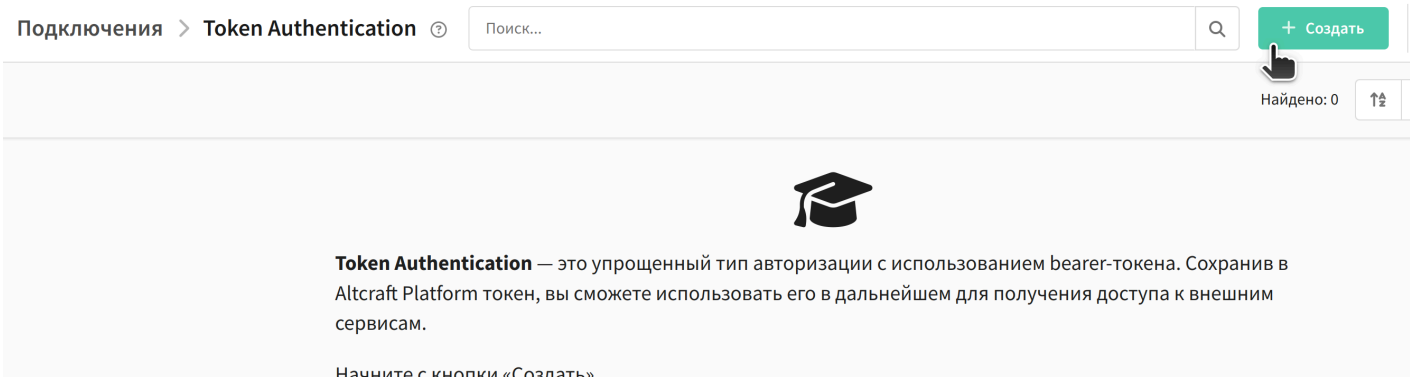
- **Узел сценария “Вызов API”**. С его помощью можно передавать в подключенный сервис информацию из сценария или, наоборот, запрашивать из сервиса необходимый для подстановки API-контент. Удобно, если нужно передать в CRM-систему данные профиля, отправленного в сценарий.
- **Action Hooks**. Механизм захвата событий позволит передать во внешний сервис информацию о событии, а также данные профиля.
- **Импорт данных по расписанию в Маркете**. Для доступа к защищенному товарному фиду доступна авторизация с помощью токена.

Создание подключения

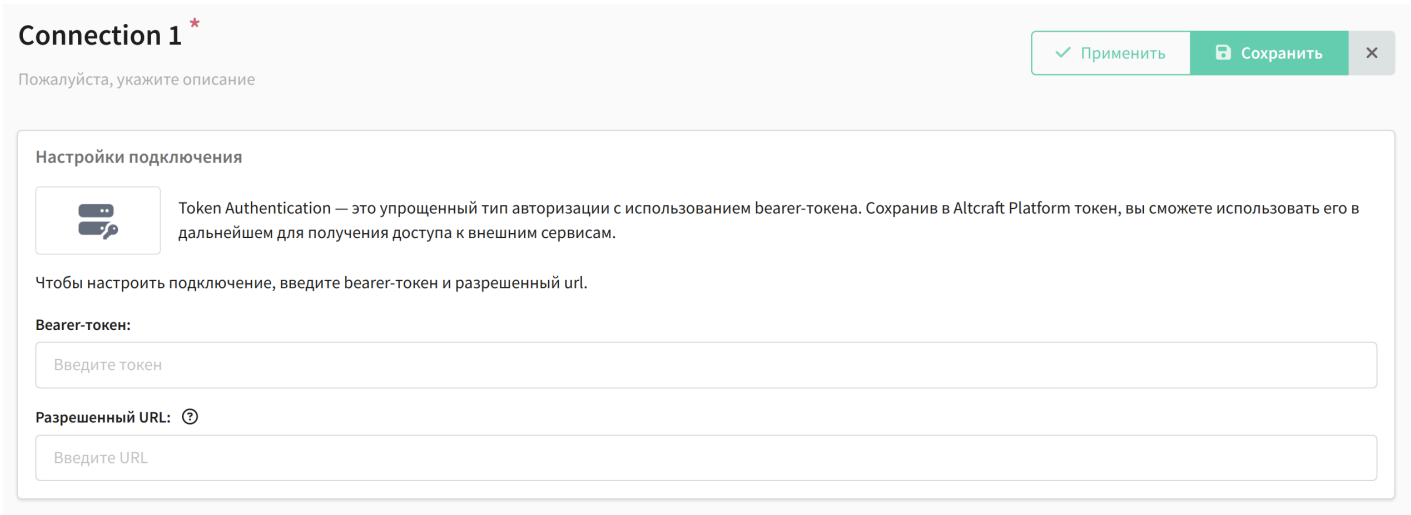
Перейдите в раздел **Настройки — Подключения** и выберите тип подключения **Token Authentication**:



Откроется список всех настроенных Token-подключений. Чтобы добавить новое, нажмите **+ Создать**.



Задайте **имя** нового подключения и при необходимости добавьте **описание**. Укажите **Bearer-токен**. Как правило, он указан в личном кабинете или настройках аккаунта того или иного сервиса. Если вы хотите предоставить доступ только для определенного сервиса, укажите его URL в поле **Разрешенный URL**:



Сохраните подключение с помощью кнопок в правом верхнем углу. Настройки подключения можно изменить в любой момент. Для этого нажмите **Редактировать**.

Подключение через Custom Authentication

Описание

Custom Authentication — это тип подключения, который позволяет настроить пользовательские HTTP-заголовки для API-запросов, включая предустановленный заголовок Authorization. Это решение предоставляет гибкость в настройке и отправке API-запросов, а также поддерживает валидацию и обработку ошибок.

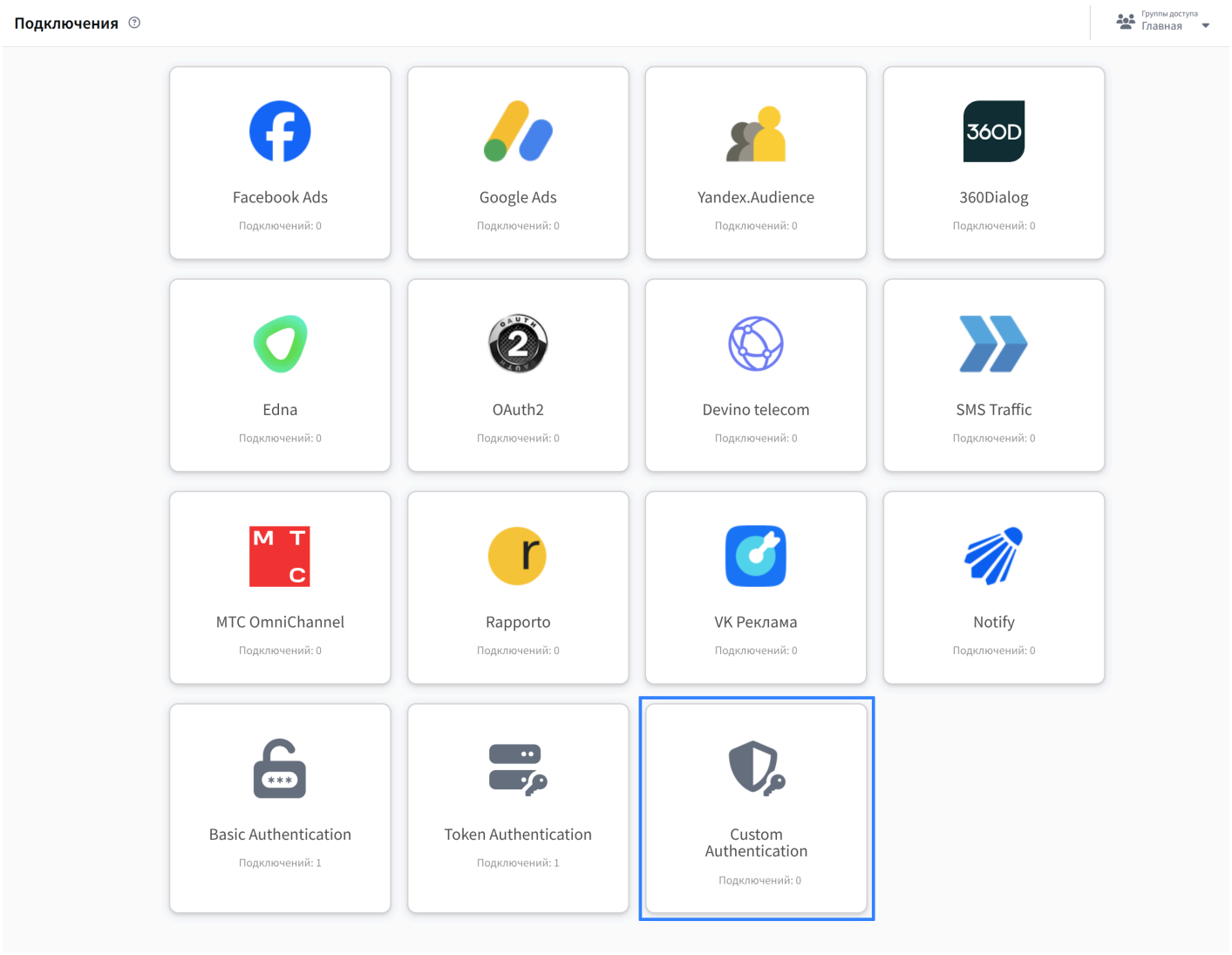
С помощью этого типа подключения вы можете связать платформу Altcraft с любым внешним сервисом, который требует пользовательских HTTP-заголовков для аутентификации или других целей, включая API-запросы с ключом в заголовке.

Где используется подключение:

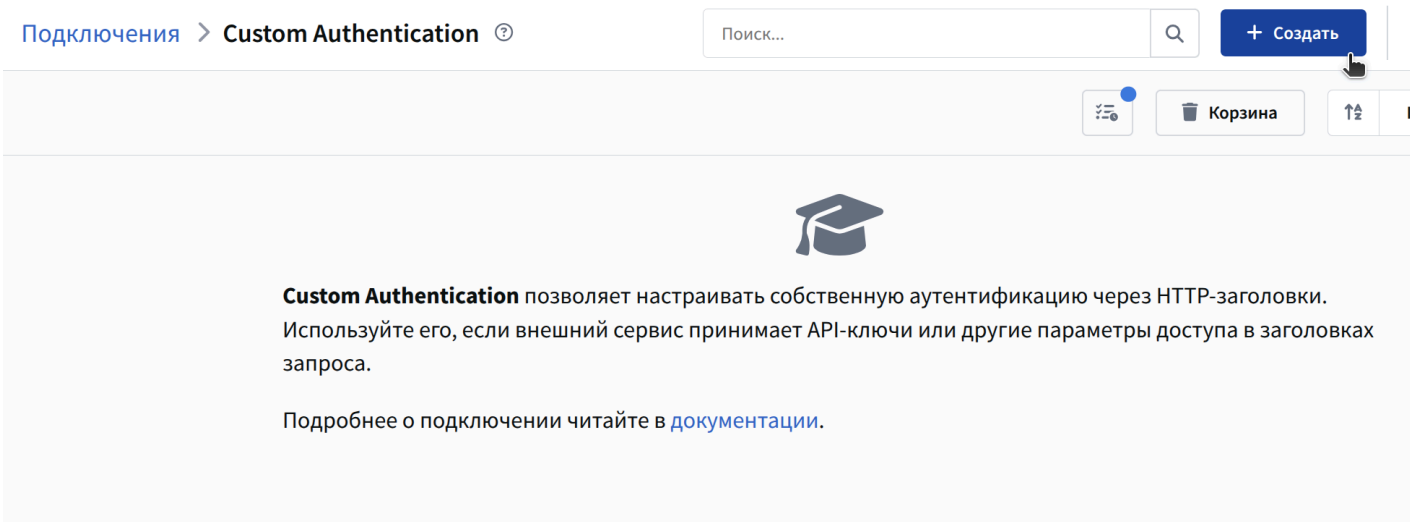
- **Узел сценария “Вызов API”**. С его помощью можно передавать в подключенный сервис информацию из сценария или, наоборот, запрашивать из сервиса необходимый для подстановки API-контент. Удобно, если нужно передать в CRM-систему данные профиля, отправленного в сценарий.
- **Action Hooks.** Механизм захвата событий позволит передать во внешний сервис информацию о событии, а также данные профиля.
- **Импорт данных по расписанию в Маркете.** При помощи пользовательской аутентификации можно настроить подключение к внешнему фиду продуктов.

Создание подключения

Перейдите в раздел **Настройки — Подключения** и выберите тип подключения **Custom Authentication**:



Откроется список всех настроенных Custom-подключений. Чтобы добавить новое, нажмите **+ Создать**.



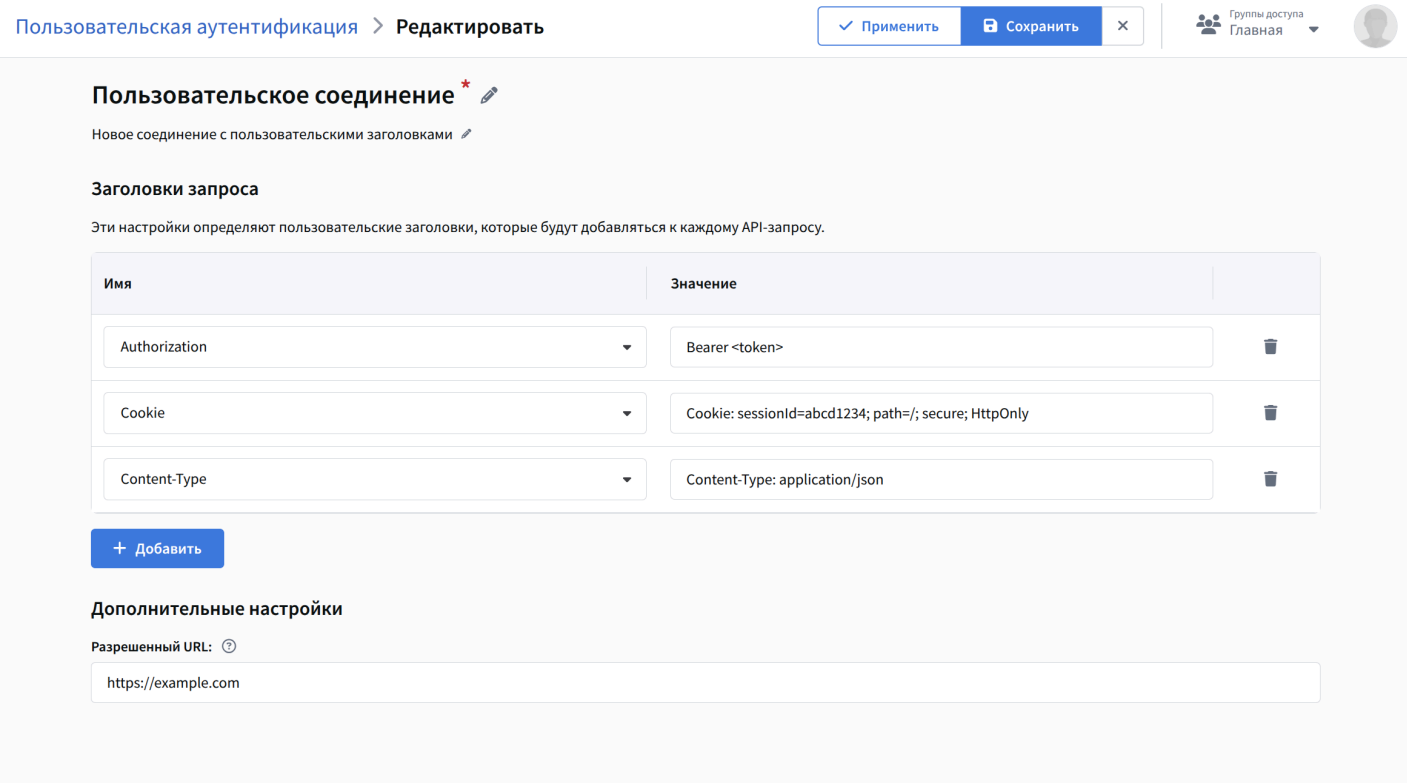
Задайте **имя** нового подключения и при необходимости добавьте **описание**. Затем добавьте желаемые **заголовки** и укажите их **значение**. Вы можете выбрать заголовки из выпадающего списка или указать собственные.

TIP: Запрещенные заголовки

Некоторые системные заголовки запрещены к добавлению, так как их использование может привести к неисправностям в работе подключения. Список этих заголовков:

- “Host”
- “Connection”
- “Keep-Alive”
- “Accept-Encoding”
- “Content-Length”
- “Transfer-Encoding”
- “Trailer”
- “Upgrade”
- “Date”
- “Location”
- “Server”

Если вы хотите предоставить доступ только для определенного сервиса, укажите его URL в поле **Разрешенный URL**:



Сохраните подключение с помощью кнопок в правом верхнем углу. Настройки подключения можно изменить в любой момент. Для этого нажмите **Редактировать**.

Подключение к Notify

INFO

Согласно политике Notify, в данный момент новые доступы к личному кабинету сервиса **не предоставляются.**

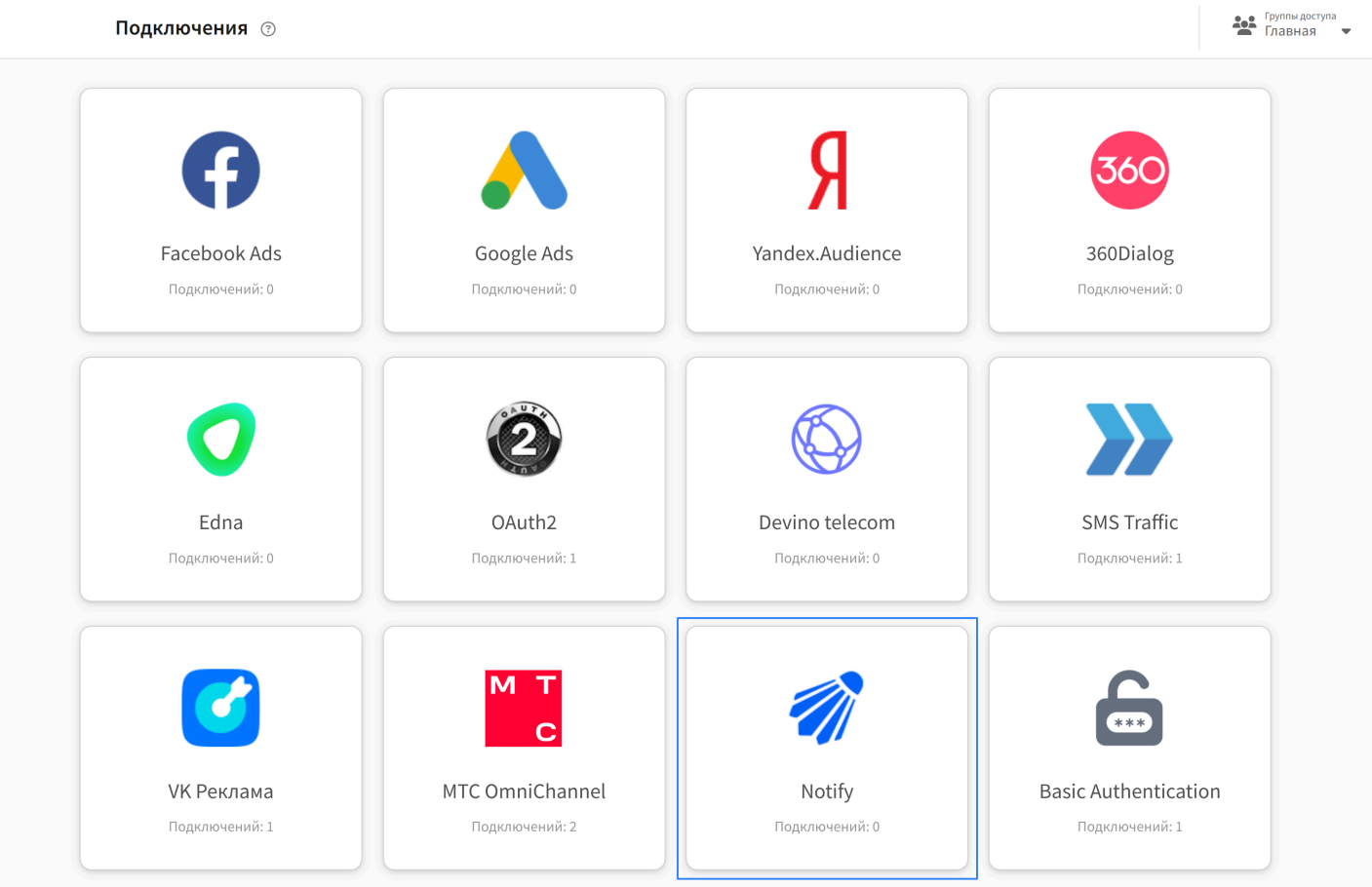
Поддержка текущих пользователей доступна через техподдержку MAX Business.

Подключение к **Notify** используется для создания интеграции и дальнейшего использования сервиса в качестве канала коммуникации с клиентами. В этой статье описывается, как создать и настроить такое подключение.

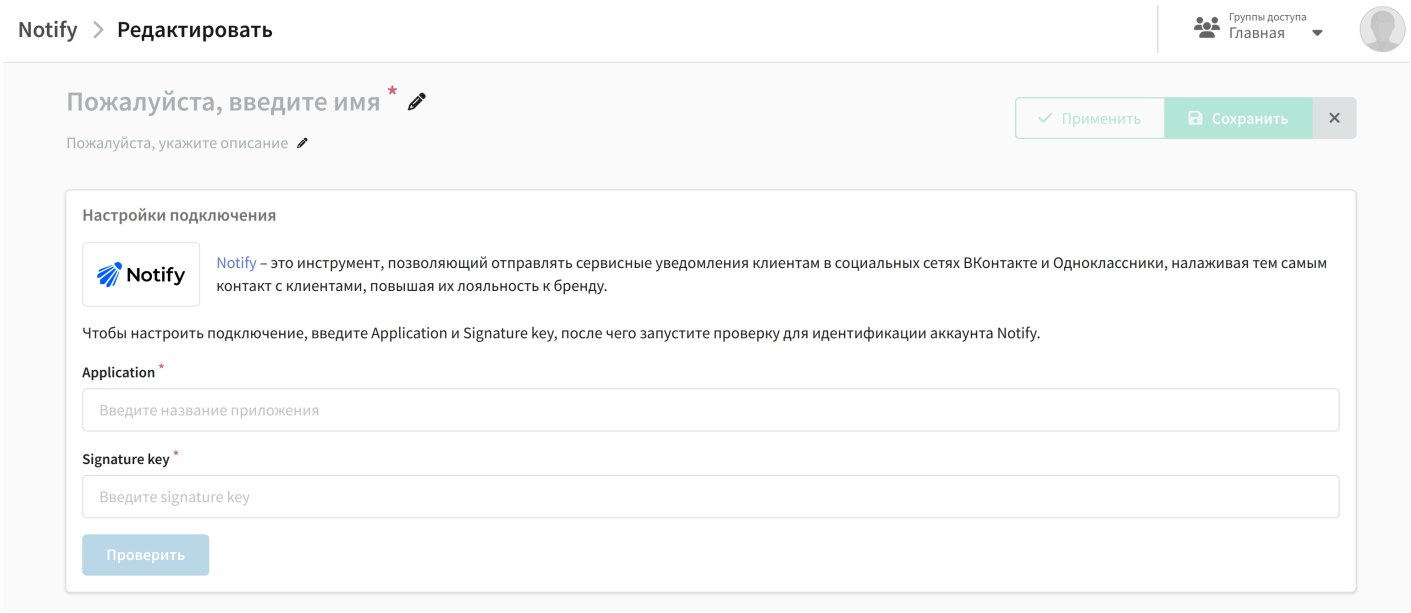
TIP

Подключение доступно клиентам, которые уже имеют прямой доступ к сервису, т.е. тем, у кого уже есть **учётная запись Notify**. В данный момент Notify не позволяет создать новые аккаунты в сервисе.

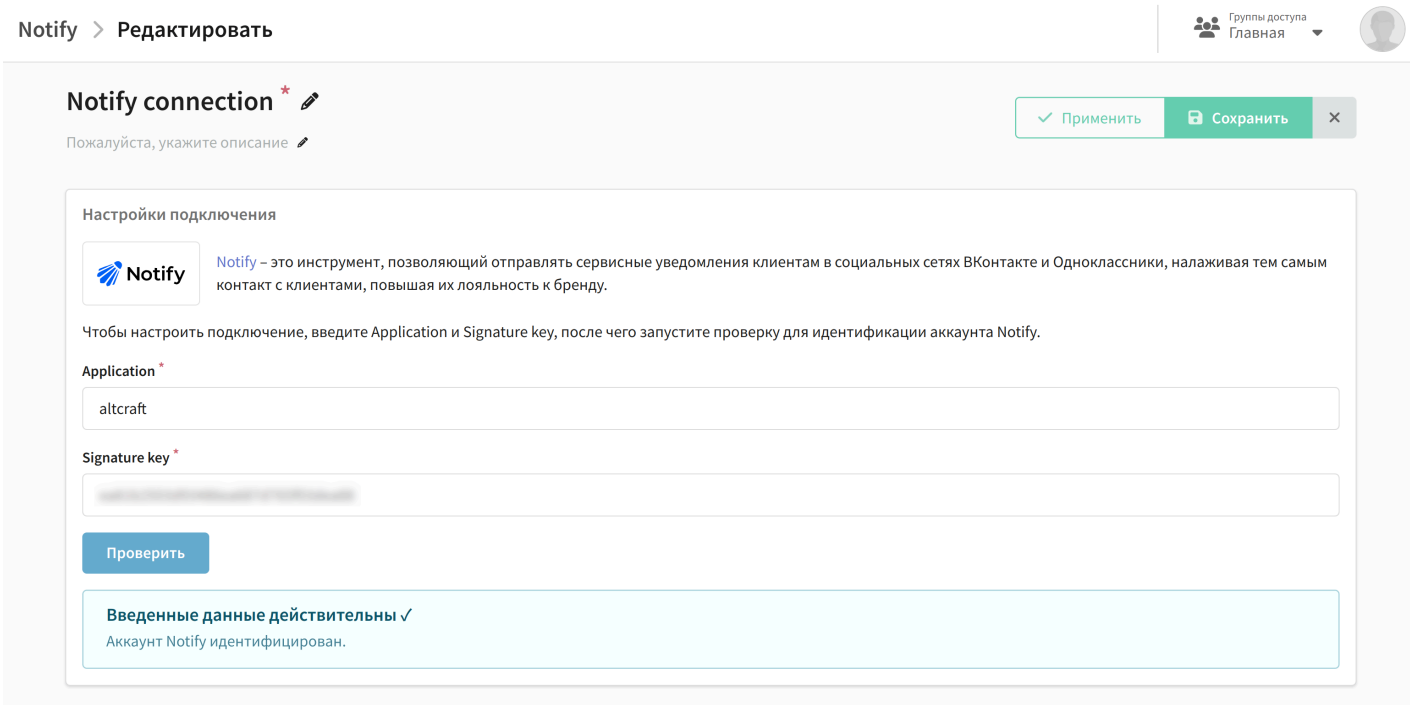
В главном меню перейдите в раздел **Настройки** —> **Подключения**. Выберите тип подключения — **Notify**:



Нажмите кнопку **“Создать”** на верхней панели. В открывшемся окне задайте имя и описание подключения:



Для создания подключения вам понадобятся параметры **Application** и **Signature key** из личного кабинета Notify. После заполнения этих полей, нажмите кнопку **“Проверить”**. Если введенные данные действительны, платформа отобразит соответствующее сообщение:



Сохраните подключение, нажав на кнопку **“Сохранить”**. После этого можно переходить к **настройке интеграции с Notify**.

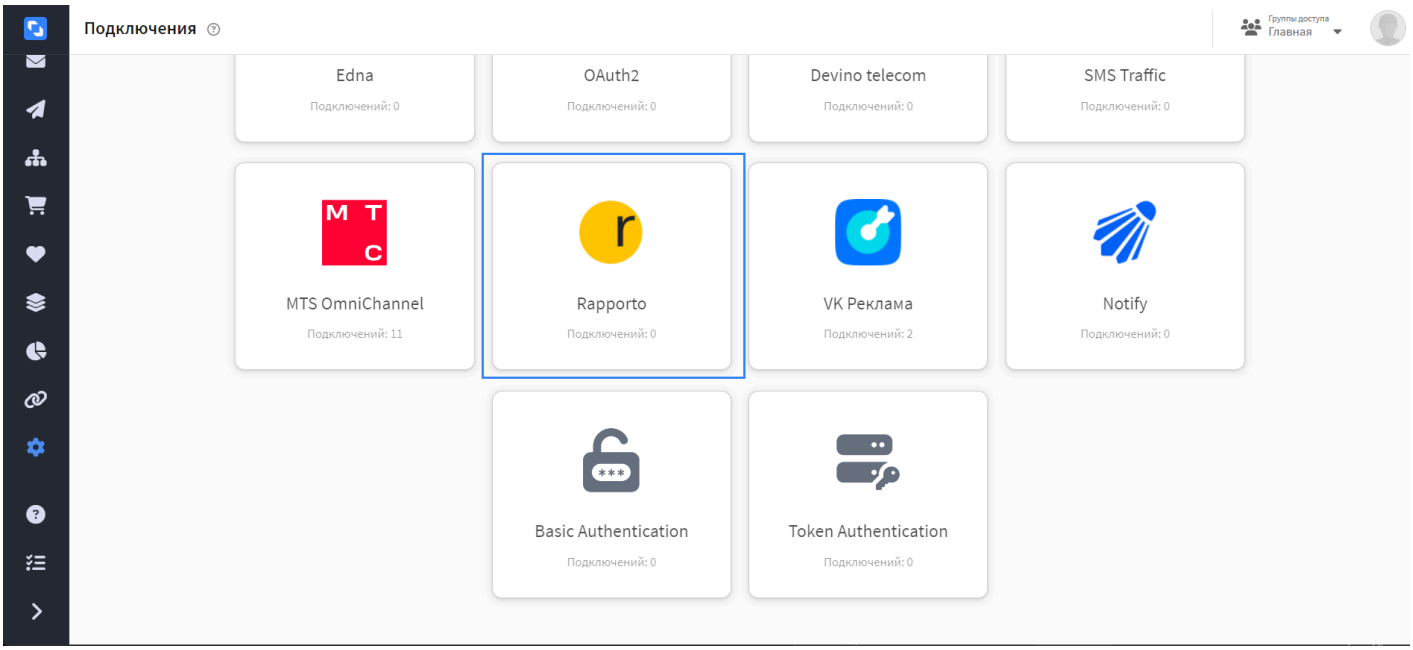
Подключение к Rapporto

Для интеграции платформы с мессенджером Viber™ необходимо настроить подключение к одному из провайдеров. В этой статье описывается, как настроить подключение к провайдеру Rapporto.

TIP

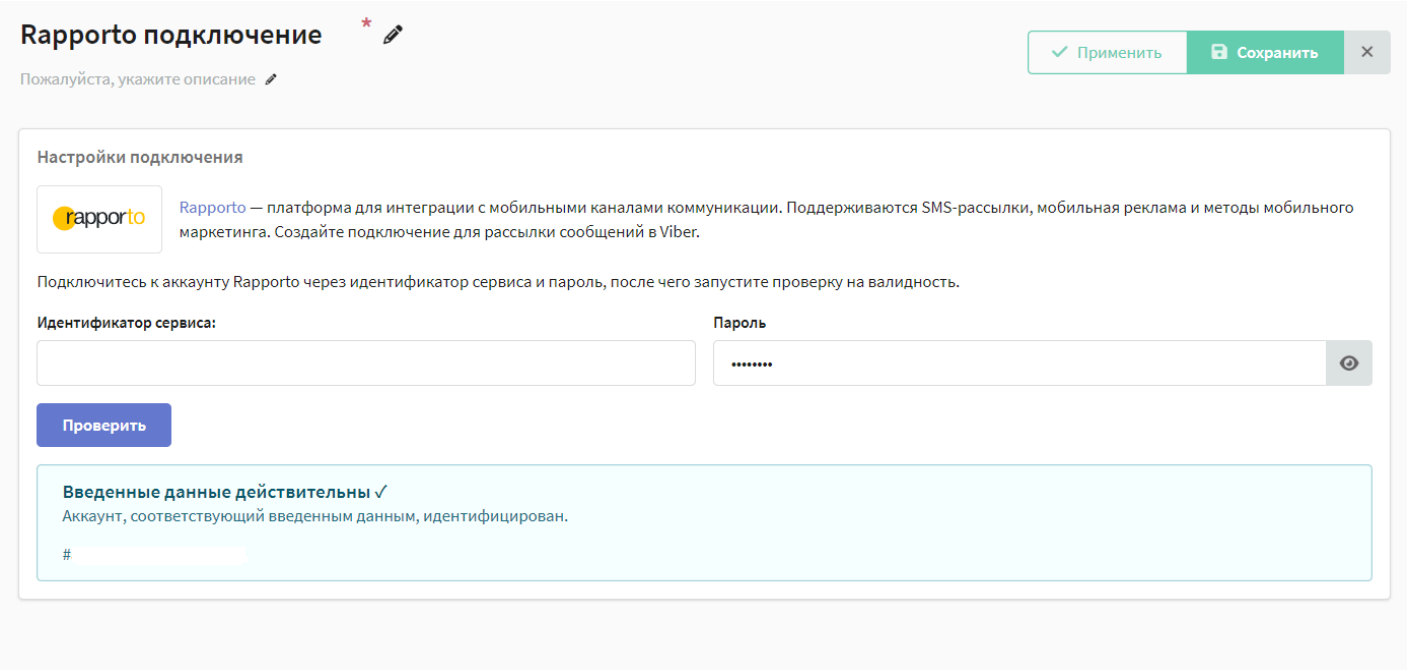
Перед тем, как настраивать подключение на платформе, ознакомьтесь с документацией Rapporto для того, чтобы настроить интеграцию и запросить **идентификатор сервиса** и **пароль**.

В главном меню перейдите в раздел **Настройки** —> **Подключения**. Выберите тип подключения — **Rapporto**:



Нажмите кнопку “**Создать**” на верхней панели. Откроется окно настройки подключения: введите **имя подключения**, по желанию добавьте **описание**.

Далее необходимо пройти проверку на валидность. Введите **идентификатор сервиса** и **пароль**, нажмите “**Проверить**”. При успешном подключении появится следующее:



Нажмите на кнопку “**Сохранить**”, чтобы сохранить подключение. После этого вы можете переходить к **настройке интеграции с Viber™**.

---

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

API-запросы: с чего начать

API-запросы

Большинство пользователей платформы Altcraft взаимодействуют с ней через веб-интерфейс в браузере. Однако для более гибкой настройки и автоматизации процессов можно использовать API.

API (Application Programming Interface) — это набор правил, позволяющий программам взаимодействовать между собой. С помощью API вы можете автоматизировать различные задачи и интегрировать платформу Altcraft с другими системами.

Как правило, написанием API-запросов занимаются разработчики. Однако знание основ работы API маркетологом упрощает его взаимодействие с разработчиками. Маркетолог сможет более четко ставить задачи, если будет понимать, какими средствами можно её реализовать.

В этом разделе вы найдете руководства по созданию базовых API-запросов, таких как импорт профиля в базу, запуск триггерной рассылки и отправка профиля в сценарий.

В этом разделе

Импорт и обновление профиля

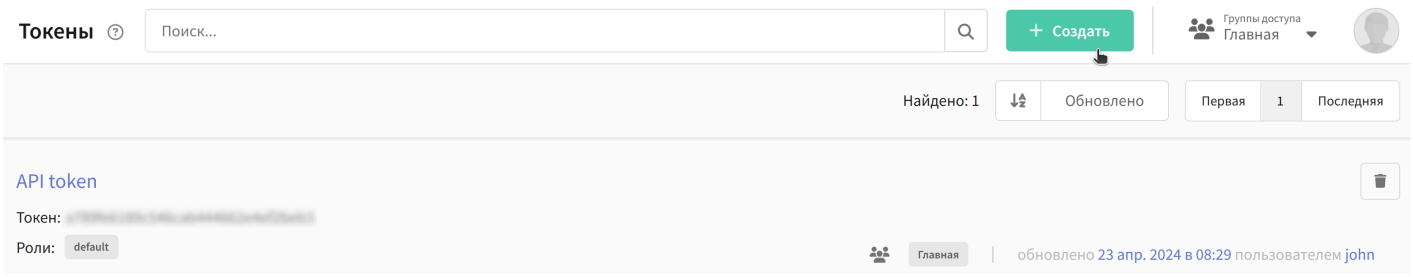
В этой статье покажем, как написать и отправить API-запросы для выполнения следующих задач:

- Добавление профиля в базу данных
- Обновление данных существующего профиля
- Подписка профиля на различные каналы

Получение API-токена

API-токен — это ключ доступа к платформе через API. Он используется в любом API-запросе.

Найти токен можно в настройках платформы Altcraft, в разделе **“Токены”**. Если у вас ещё нет токена, создайте его, нажав на соответствующую кнопку:



CAUTION

API-токен могут создать только **мастер-пользователи**. Если у вас нет необходимых прав, обратитесь к пользователю с правами мастера.

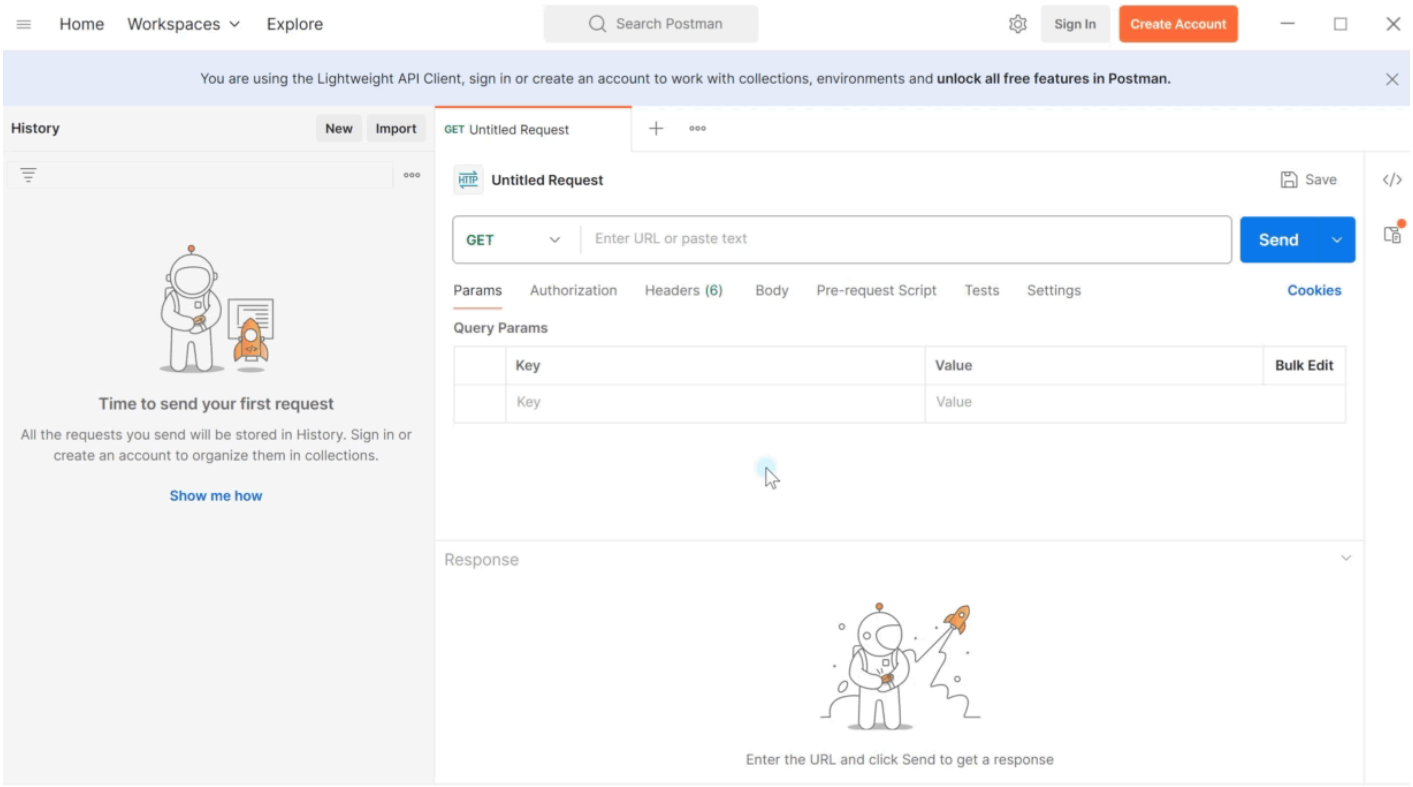
Установка и запуск Postman

Для работы с API-запросами потребуется приложение, позволяющее отправлять и принимать их.

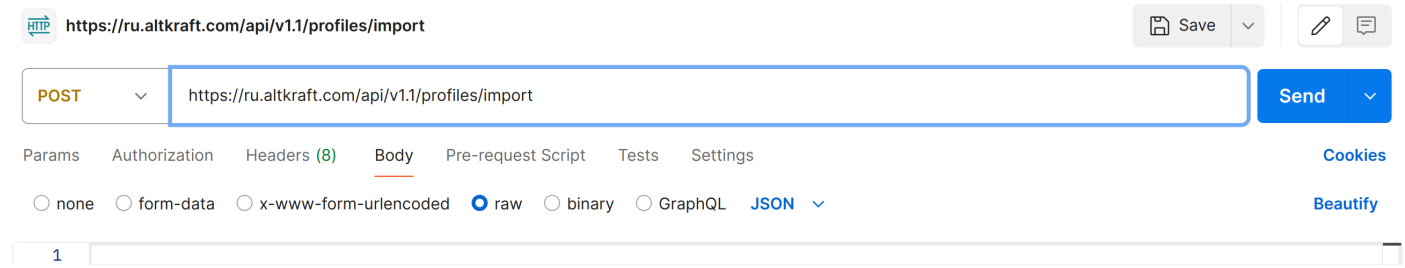
1. Скачайте и установите приложение Postman или откройте его веб-версию.

2. Выберите метод **POST** для вашего запроса.

3. Перейдите на вкладку **Body** и выберите режим **raw**. Убедитесь, что выбран формат **JSON**.



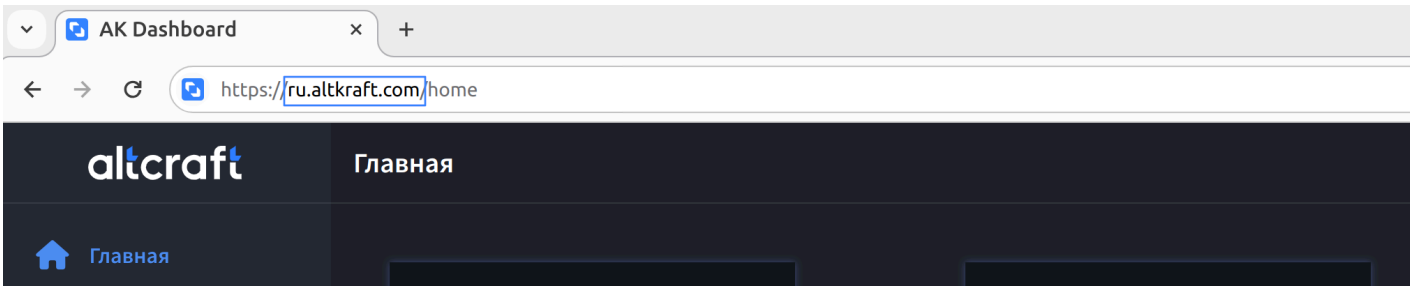
URL-адрес запроса



Для отправки запроса на сервер вам необходимо ввести его **URL**. Он индивидуален для каждого запроса. Например, для запроса на импорт профиля URL-адрес будет таким:

`https://example.com/api/v1.1/profiles/import`

Замените `example.com` на домен, на котором развернута ваша платформа. Вы можете скопировать адрес прямо из адресной строки:



Создание запроса

Добавление нового профиля

Перейдем к написанию запроса. Предположим, мы хотим добавить в базу #5 профиль клиента с электронной почтой `i.petrov@example.com`. Составим запрос:

```
{
     "token": "abcdefghklmnopqrstzxc",
     "db_id": 5,
     "matching": "email",
     "email": "i.petrov@example.com",
     "data": {
        "email": "i.petrov@example.com"
    }
}
```

- **“token”** — API-токен. Подробнее о том, где можно его найти, можно прочитать в начале статьи.
- **“db_id”** — номер базы, в которую добавляется профиль. Это число указано рядом с названием базы профилей справа от знака #.
- **“data”** — объект, содержащий данные профиля. В примере выше мы передаём в этом объекте только email-адрес клиента.
- **“matching”** и **“email”** — параметр для поиска профиля в базе. В данном случае для поиска профиля используется email-адрес. Если профиль с таким адресом уже есть в базе, то запрос обновит его данные, если нет — создаст новый профиль. Параметр “matching” может принимать различные значения. При этом он не является обязательным и, если его не указать, поиск профиля произойдёт по email.

Подробнее о параметре `matching`

Параметр `matching` определяет, по какому полю будет происходить поиск профиля в базе данных. В платформе доступно несколько режимов поиска, например, платформа может искать профиль по email-адресу, номеру телефона, системному идентификатору или по любому значению, которое вы храните в дополнительном поле базы.

Главное, чтобы поле, по которому вы запускаете поиск в базе, было уникальным для профиля, т.е. значение этого поля не должно совпадать у разных профилей в одной базе.

Если вы хотите найти профиль по его email, то `matching` будет выглядеть так:

```
{
     "matching": "email",
     "email": "example@example.com"
}
```

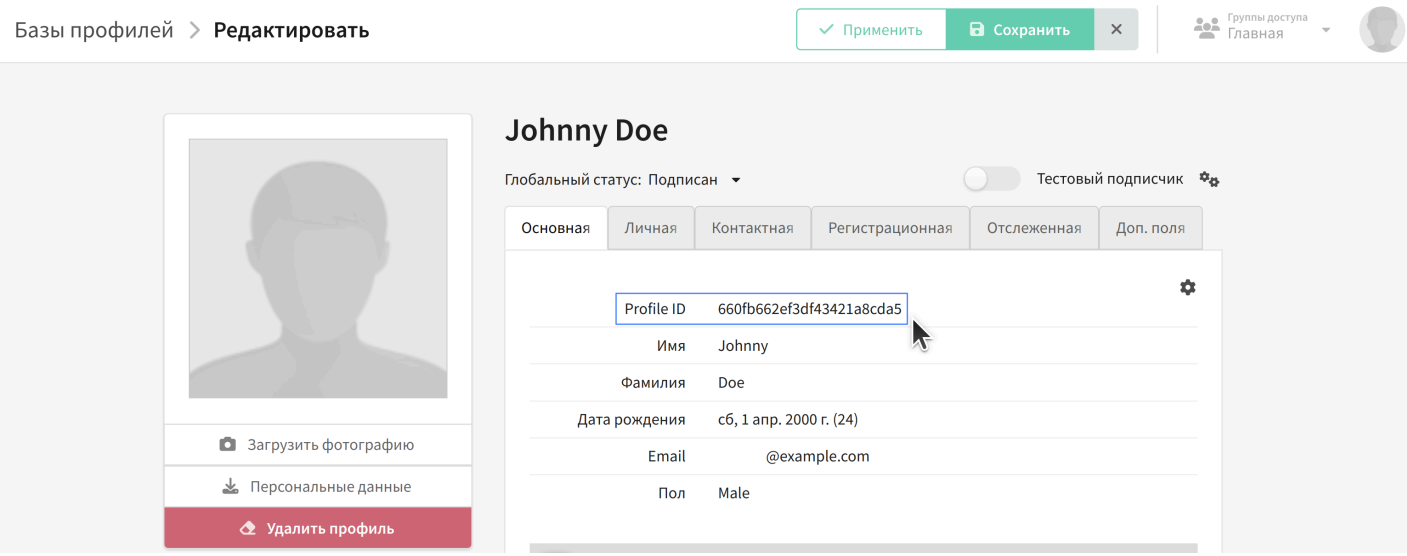
Поиск профиля по телефону:

```
{
     "matching": "phone",
     "phone": "+79000000000"
}
```

Поиск профиля по идентификатору (только при обновлении профиля):

```
{
     "matching": "profile_id",
     "profile_id": "012452gf62fcd71fjh8ec0"
}
```

Идентификатор профиля можно найти здесь:



Если для поиска профиля в базе вы используете дополнительное поле, то ваш запрос будет выглядеть так:

```
{
     "matching": "custom",
     "field_name": "CRM_ID",
     "field_value": "12345"
}
```

где `CRM_ID` — это название поля, а `12345` — значение этого поля у профиля.

TIP

Больше информации о матчинге вы найдете в **этой статье**.

Обновление данных профиля

Допустим, мы получили новые данные о клиенте: его зовут Иван Петров, родился 15.01.1990, живёт в Москве, телефон — +79000000000. Теперь нам необходимо обновить профиль и добавить эту информацию. Запрос будет выглядеть так:

```
{
     "token": "abcdefghklmnopqrstzxc",
     "db_id": 5,
     "matching": "email",
     "email": "i.petrov@example.com",
     "data": {
        "email": "i.petrov@example.com",
        "_fname": "Ivan",
        "_lname": "Petrov",
        "_sex": "male",
        "_bdate": "1990-01-15T12:00:00Z",
        "_city": "Moscow",
        "phones": ["+79000000000"]
    }
}
```

Мы расширили параметр `"data"` следующими данными:

- **“_fname”** — имя
- **“_lname”** — фамилия
- **“_bdate”** — дата рождения в формате “1990-01-15T12:00:00Z”
- **“_sex”** — пол, “male” для мужчины, “female” для женщины
- **“_city”** — город
- **“phones”** — номер телефона.

Добавление подписок в профиль

Мы обновили профиль и теперь нам нужно подписать его на email, sms и push-каналы в ресурсе #1. Для этого нужно добавить в `data` массив `subsriptions`:

```
{
    "token": "abcdefghklmnopqrstzxc",
    "db_id": 5,
    "matching": "email",
    "email": "i.petrov@example.com",
    "data": {
      "email": "i.petrov@example.com",
      "_fname": "Ivan",
      "_lname": "Petrov",
      "_sex": "male",
      "_bdate": "1990-01-15T12:00:00Z",
      "_city": "Moscow",
      "phones": ["+79000000000"],
      "subscriptions": [
            {
                "channel": "email",
                "email": "i.petrov@example.com",
                "resource_id": 1
            },
            {
                "channel": "sms",
                "phone": "+79000000000",
                "resource_id": 1
            },
            {
                "channel": "push",
                "provider": "Firefox",
                "resource_id": 1,
                "subscription_id":"abcdefghijklmnqrstuvwxyz"
            }
        ]
    }
}
```

Для каждого канала связи, через который будет осуществляться рассылка, необходимо указать следующие параметры:

- **“channel”** — тип канала, например, **“email”**, **“sms”**, **“push”**
- **“resource_id”** — идентификатор ресурса, который необходимо настроить заранее.

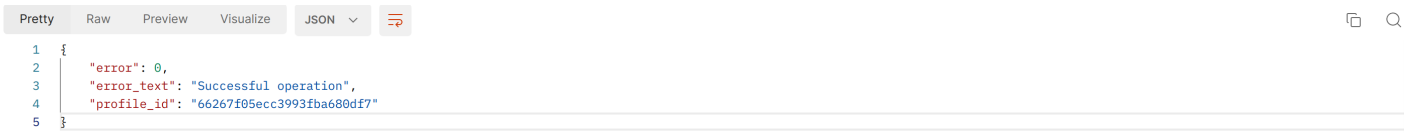
Также в зависимости от канала, на который вы подписываете профиль, потребуется указать определенные контактные данные. Например, для email-канала — это email-адрес, для sms-канала — номер телефона, а для push-канала — провайдер (`provider`) и push-токен (`subscription_id`).

TIP

Обратите внимание на синтаксис вашего запроса. Лишние символы или незакрытые скобки могут привести к ошибке.

Отправка запроса

Чтобы отправить запрос, нажмите кнопку **Send** в Postman. При успешном выполнении вы получите сообщение об успешной отправке запроса:



Если что-то пошло не так, в ответном сообщении будет указан код ошибки и ее описание:

Если у вас не получилось самостоятельно устранить ошибку, передайте код и описание ошибки в **службу поддержки Altcraft**.

TIP

Хотя в статье описано поэтапное добавление и обновление данных профиля, на практике вы можете сразу отправить финальный запрос. Если профиль с указанными данными уже существует в базе, он будет обновлен, а если нет — будет создан новый профиль с полными данными и подписками на ресурс.

TIP

В статье описывается лишь часть возможных параметров для данного запроса. С остальными параметрами можно ознакомиться **здесь**.

Запуск триггерной рассылки

В этой статье мы покажем, как отправить сообщению профилю, запустив триггерную рассылку через API.

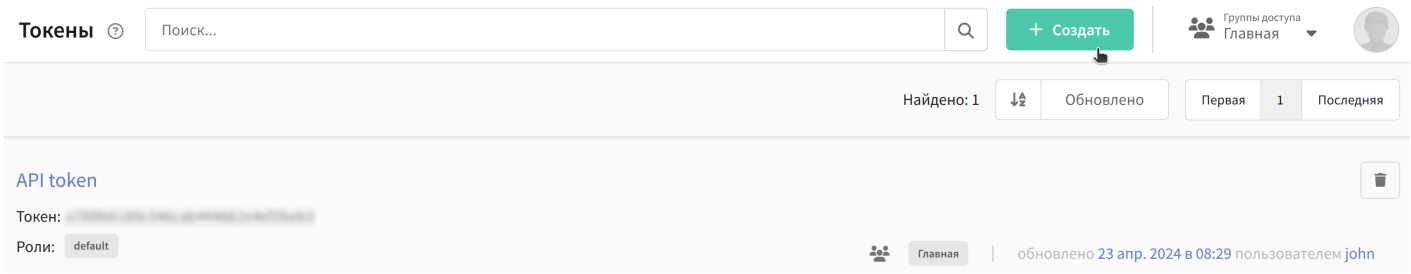
TIP

Через API можно запустить только **активную** триггерную рассылку, настроенную на событие **“При использовании особого вызова API”**. Создание триггерной рассылки описано в **этой статье**.

Получение API-токена

API-токен — это ключ доступа к платформе через API. Он используется в любом API-запросе.

Найти токен можно в настройках платформы Altcraft, в разделе **“Токены”**. Если у вас ещё нет токена, создайте его, нажав на соответствующую кнопку:



CAUTION

API-токен могут создать только **мастер-пользователи**. Если у вас нет прав для создания токена, обратитесь к пользователю с правами мастера.

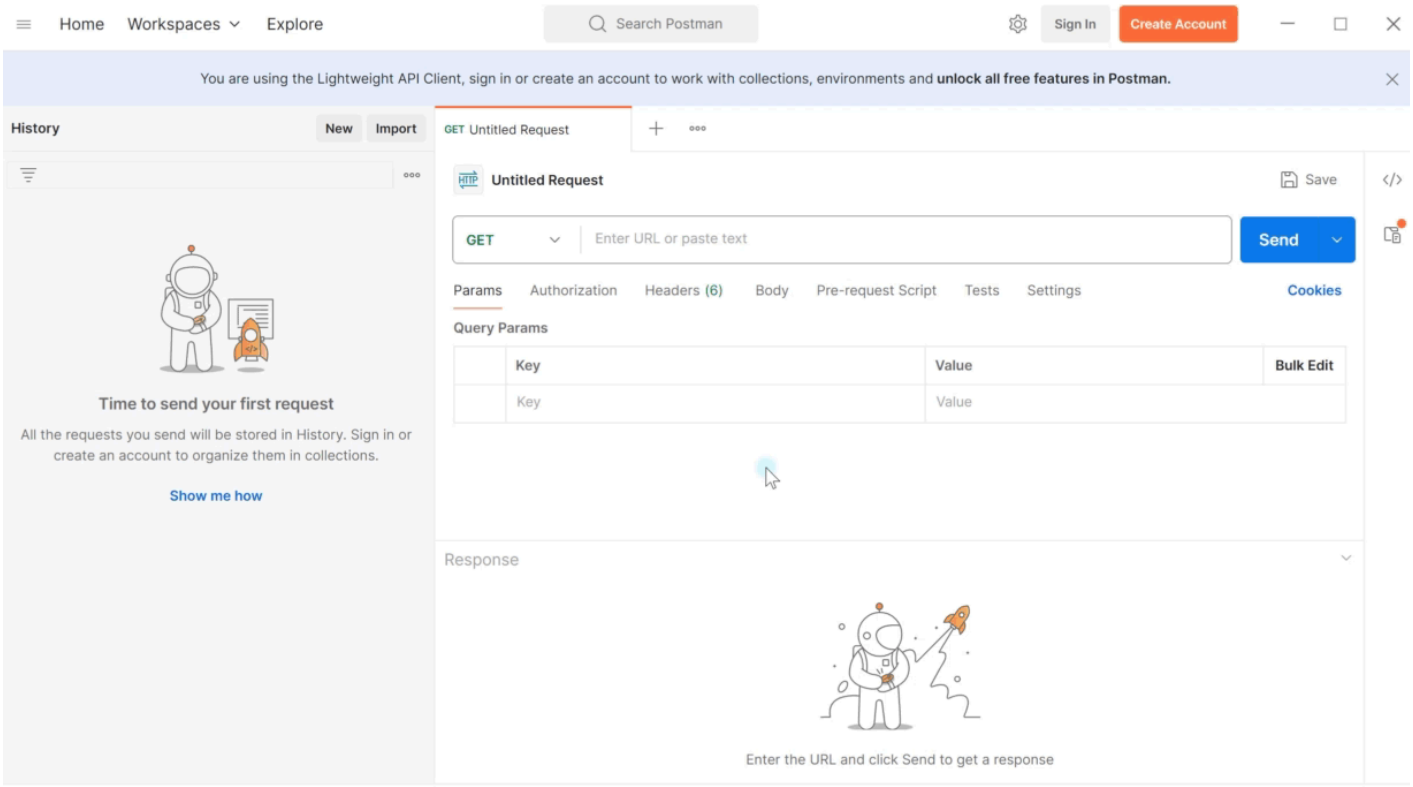
Установка и запуск Postman

Для работы с API-запросами вам понадобится приложение, позволяющее отправлять и принимать их.

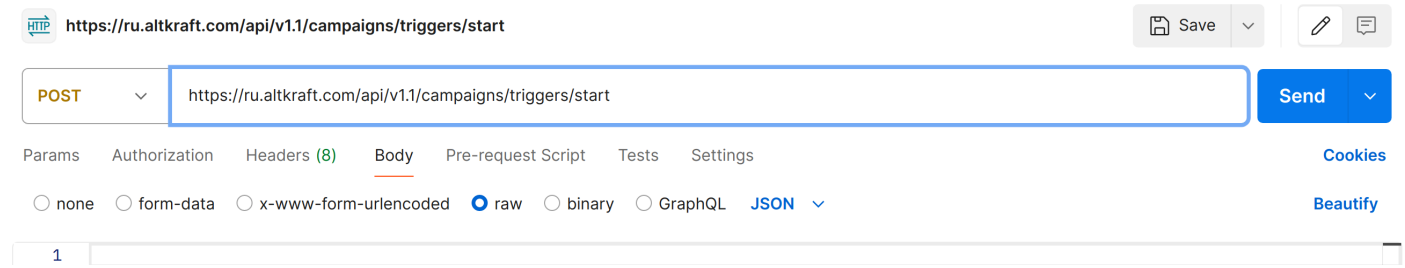
1. Скачайте и установите приложение Postman или откройте его веб-версию.

2. Выберите метод **POST** для вашего запроса.

3. Перейдите на вкладку **Body** и выберите режим **raw**. Убедитесь, что выбран формат **JSON**.



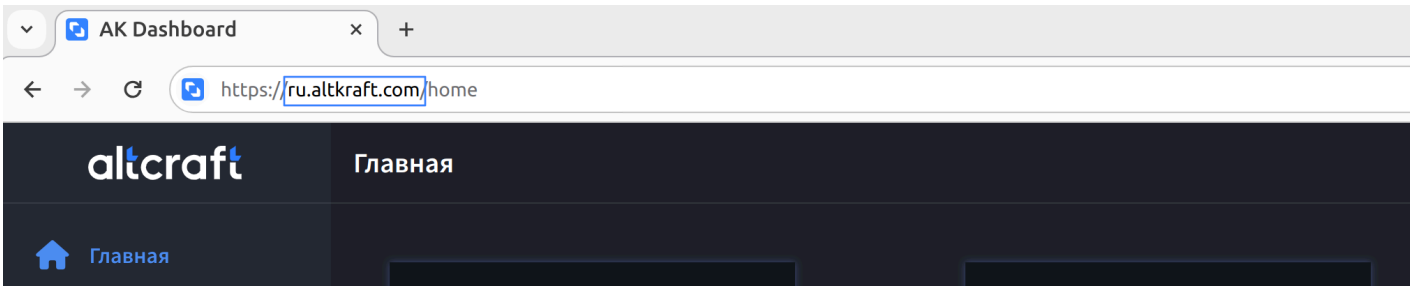
URL-адрес запроса



Для отправки запроса на сервер вам необходимо ввести его **URL**. Он индивидуален для каждого запроса. Например, для запроса на импорт профиля URL-адрес будет таким:

`https://example.com/api/v1.1/campaigns/triggers/start`

Замените `example.com` на домен, на котором развернута ваша платформа. Вы можете скопировать адрес прямо из адресной строки:



Создание запроса

Перейдем к написанию запроса.

Поиск профиля по email

Предположим, в базе есть подписчик с email-адресом `zaycev@example.com`. Мы хотим отправить ему на почту сообщение, запустив триггерную рассылку #1. Наш запрос может выглядеть так:

```
{
  "token": "abcdefghklmnopqrstzxc",
  "id": 1,
  "matching": "email",
  "email": "zaycev@example.com"
}
```

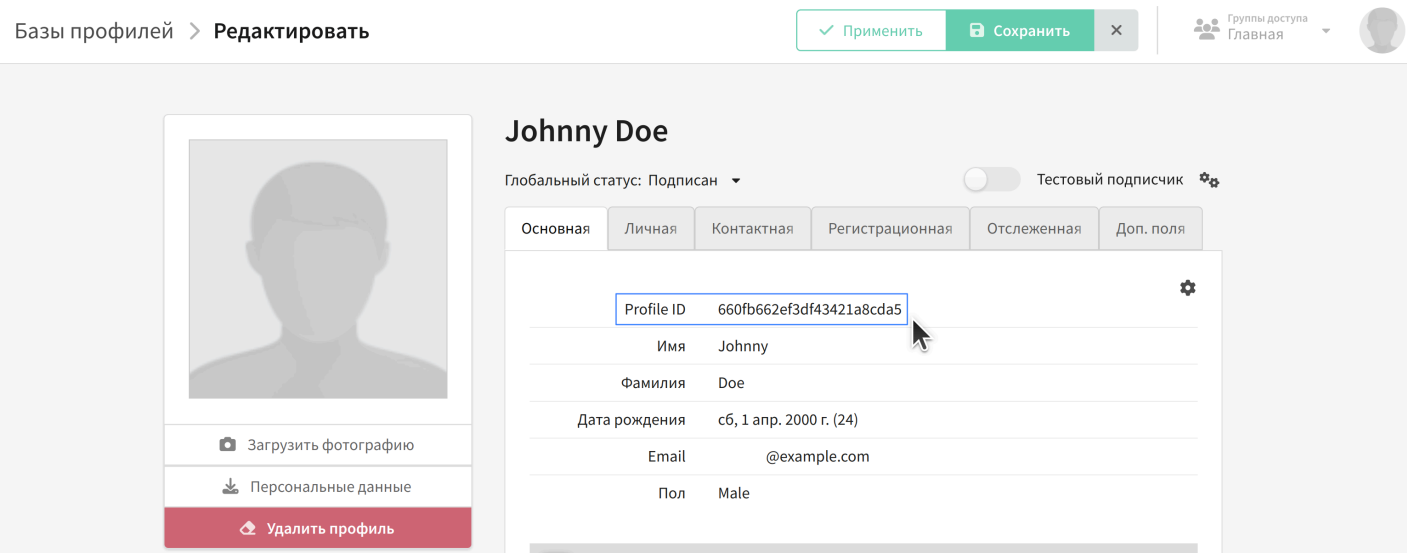
- **“token”** — ваш API-токен. Подробнее о том, где можно его найти, написано в начале статьи.
- **“id”** — идентификатор триггерной рассылки. Он указан рядом с названием рассылки справа от знака **#** . `id` триггерной рассылки — это **всегда число**, не путайте его с названием.
- **“matching”** и **“email”** — параметры, по которым запрос найдёт в базе необходимый профиль. В данном случае используется поиск по email-адресу профиля.

Поиск профиля по profile_id

Если email-адрес клиента неизвестен, вы можете найти его профиль в базе, используя внутренний идентификатор (`profile_id`):

```
{
  "token": "abcdefghklmnopqrstzxc",
  "id": 1,
  "matching": "profile_id",
  "email": "660fb662ef3df43421a8cda5"
}
```

Идентификатор профиля можно найти в его карточке:



Поиск профиля по дополнительному полю

Если для идентификации профиля клиента в базе используются данные из **дополнительного поля**, вы можете использовать их для запуска триггерной рассылки.

Предположим, мы хотим отправить сообщение клиенту, который поставил лайк на запись Вконтакте. Но мы не знаем о нем ничего, кроме его ID Вконтакте, который записан в нашей базе данных в поле `"vk_id"`. В этом случае запрос будет выглядеть следующим образом:

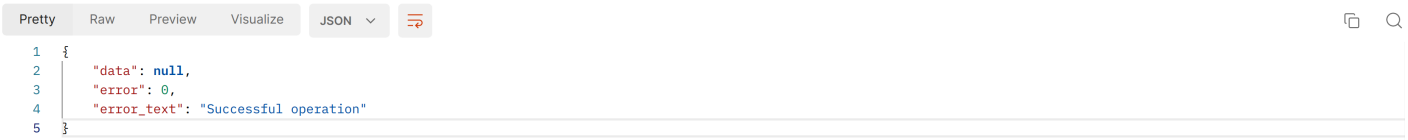
```
{
  "token": "abcdefghklmnopqrstzxc",
  "id": 5,
  "matching": "custom",
  "field_name": "vk_id",
  "field_value": "123456"
}
```

TIP

Обратите внимание на синтаксис вашего запроса. Лишние символы или незакрытые скобки могут привести к ошибке.

Отправка запроса

Чтобы отправить запрос, нажмите кнопку **Send** в Postman. При успешном выполнении вы получите сообщение об успешной отправке запроса:



Если что-то пошло не так, в ответном сообщении будет указан номер и описание ошибки:

Если у вас не получилось самостоятельно устранить ошибку, передайте номер и описание ошибки в **[службу поддержки Altcraft](#)**.

TIP

В статье описывается лишь часть возможных параметров для данного запроса. С остальными параметрами можно ознакомиться **[здесь](#)**.

Отправка профиля клиента в сценарий

В этой статье мы расскажем, как с помощью API-запроса отправить в сценарий профиль, который уже есть в базе.

TIP

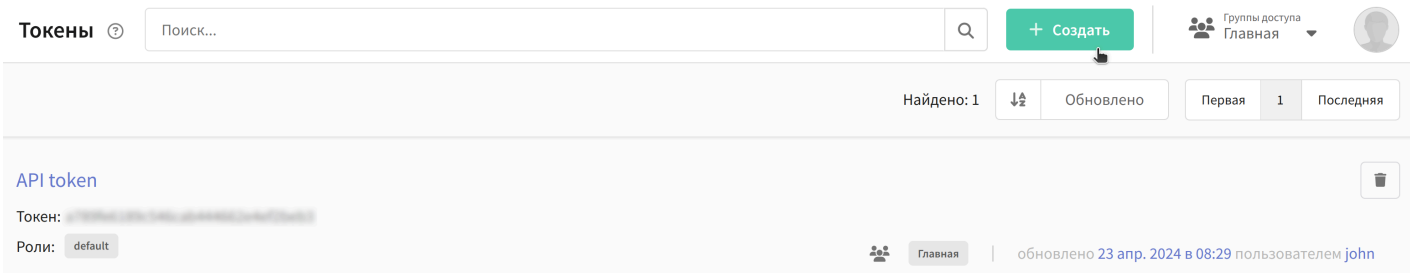
Для того чтобы запустить профиль в сценарий, сценарий должен быть создан и **активирован**. **В этой статье** вы можете прочитать о том, как работать со сценариями автоматизации.

Обратите внимание, что профили при таком запуске входят в сценарий через **узел “По умолчанию”**.

Получение API-токена

API-токен — это ключ доступа к платформе через API. Он используется в любом API-запросе.

Найти токен можно в настройках платформы Altcraft, в разделе **“Токены”**. Если у вас ещё нет токена, создайте его, нажав на соответствующую кнопку:



CAUTION

API-токен могут создать только **мастер-пользователи**. Если у вас нет прав для создания токена, обратитесь к пользователю с правами мастера.

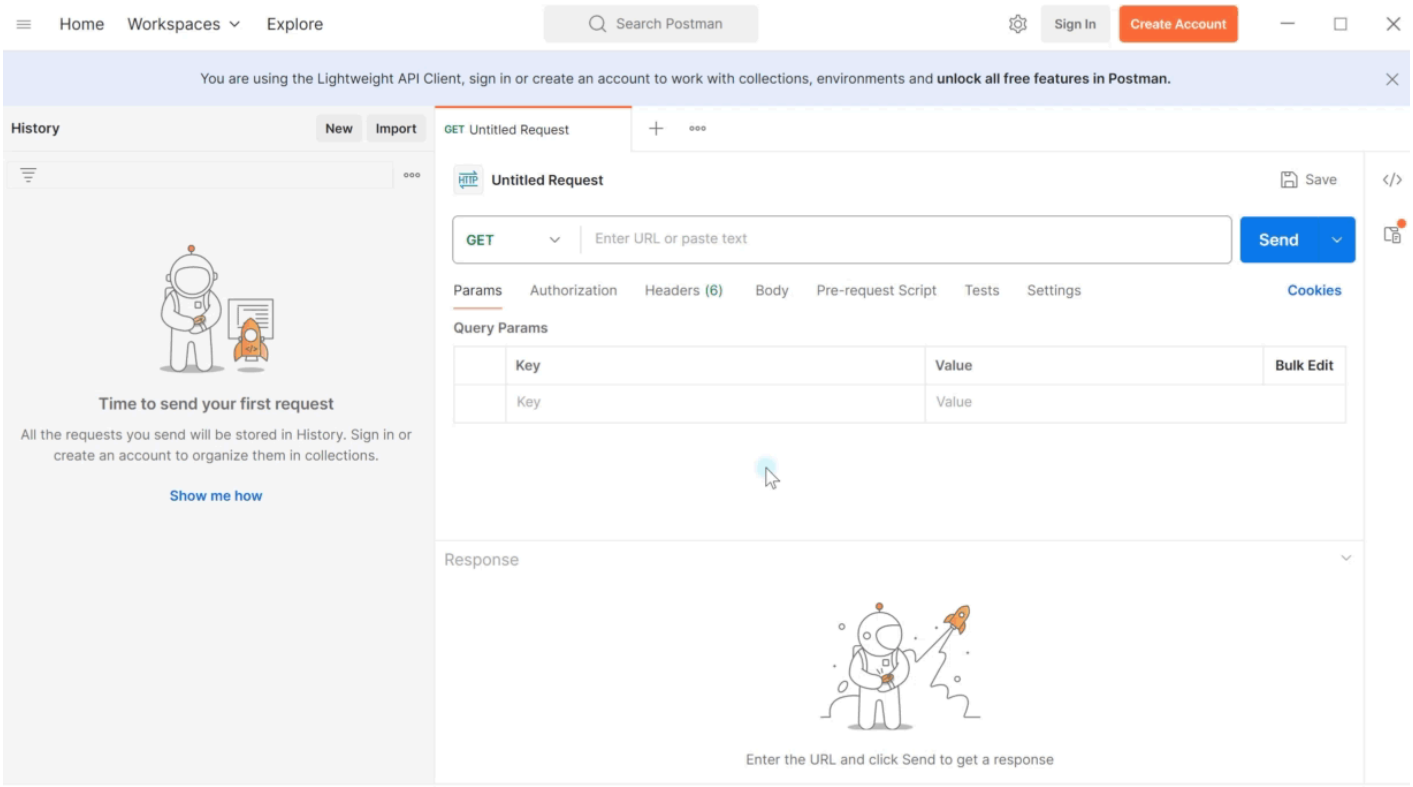
Установка и запуск Postman

Для работы с API-запросами вам понадобится приложение, позволяющее отправлять и принимать их.

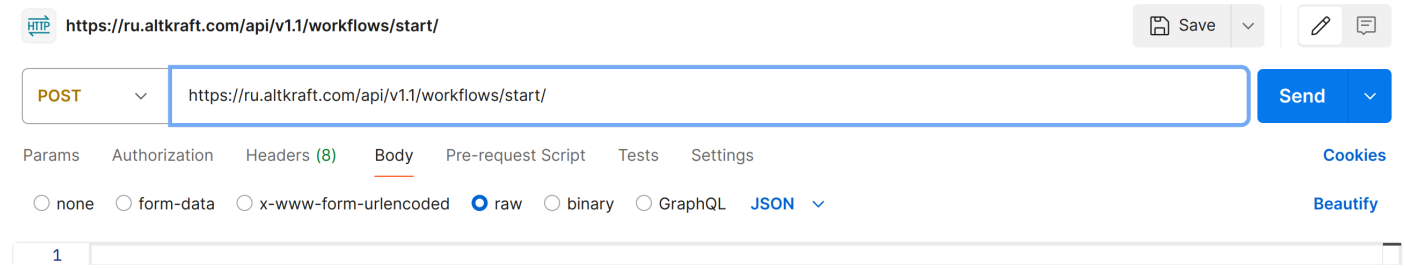
1. Скачайте и установите приложение Postman или откройте его веб-версию.

2. Выберите метод **POST** для вашего запроса.

3. Перейдите на вкладку **Body** и выберите режим **raw**. Убедитесь, что выбран формат **JSON**.



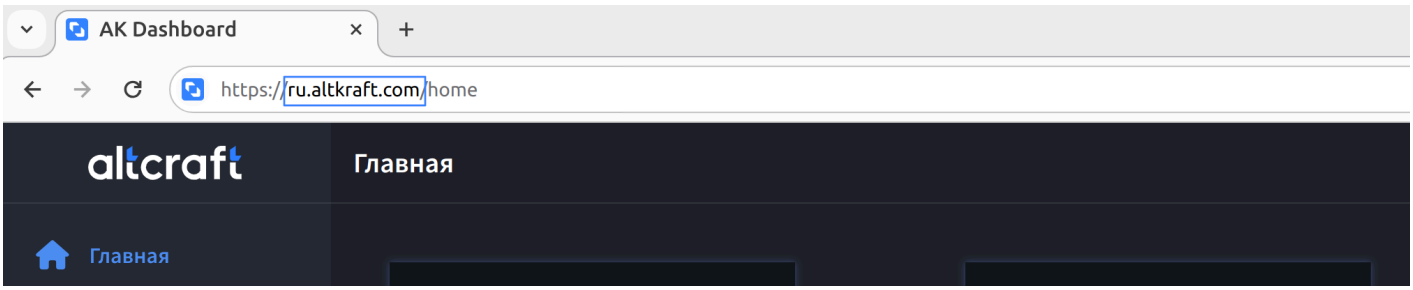
URL-адрес запроса



Для отправки запроса на сервер вам необходимо ввести его **URL**. Он индивидуален для каждого запроса. Например, для запроса на импорт профиля URL-адрес будет таким:

`https://example.com/api/v1.1/workflows/start/`

Замените `example.com` на домен, на котором развернута ваша платформа. Вы можете скопировать адрес прямо из адресной строки:



Создание запроса

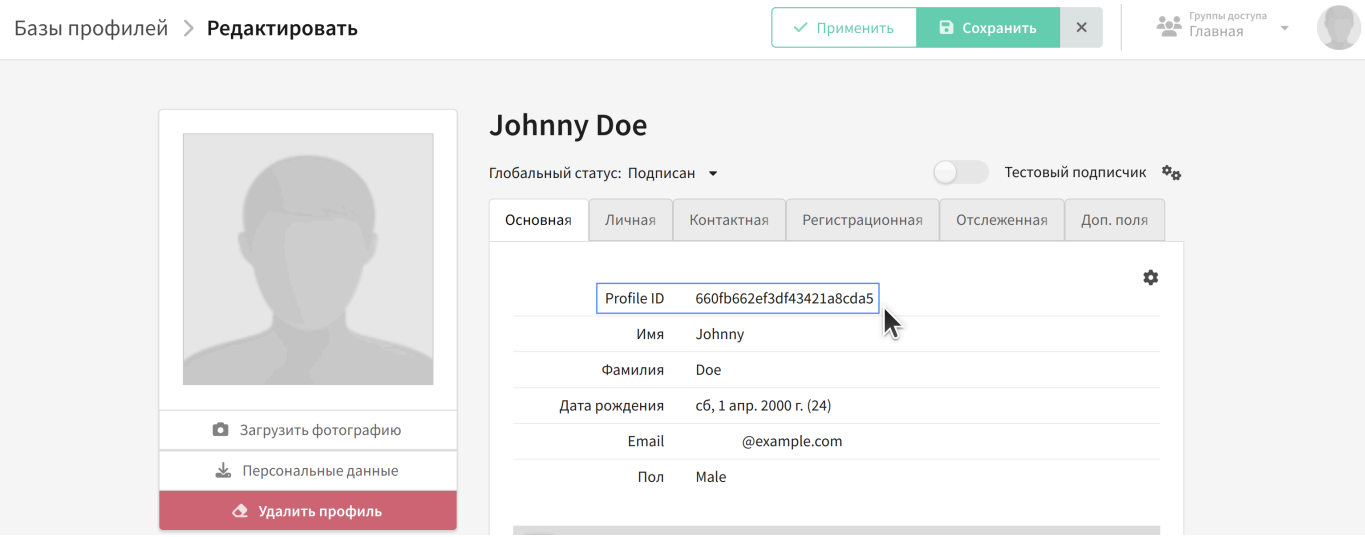
Перейдем к написанию запроса. Предположим, мы хотим отправить в сценарий #3 профиль подписчика с ID `66acb337240d5213037986d8`. Запрос может выглядеть так:

```
{
  "token": "abcdefghklmnopqrstzxc",
  "id": 3,
  "matching": "profile_id",
  "profile_id": "66acb337240d5213037986d8"
}
```

- **“token”** — ваш API-токен. Подробнее о том, где можно его найти, написано в начале статьи.
- **“id”** — идентификатор сценария. Он указан рядом с названием сценария справа от знака **#** . `id` сценария — это **всегда число**, не путайте его с названием.
- **“matching”** и **“profile_id”** — параметры, по которым запрос найдёт в базе необходимый профиль. В данном случае используется поиск по идентификатору профиля.

INFO

Идентификатор профиля можно найти здесь:



Подробнее о параметре `matching`

Параметр `matching` определяет, по какому полю будет происходить поиск профиля в базе данных. В платформе доступно несколько режимов поиска, например, платформа может искать профиль по email-адресу, номеру телефона, системному идентификатору или по любому значению, которое вы храните в дополнительном поле базы.

Главное, чтобы поле, по которому вы запускаете поиск в базе, было уникальным для профиля, т.е. значение этого поля не должно совпадать у разных профилей в одной базе.

Если вы хотите найти профиль по его email, то `matching` будет выглядеть так:

```
{
     "matching": "email",
     "email": "example@example.com"
}
```

Поиск профиля по телефону:

```
{
     "matching": "phone",
     "phone": "+79000000000"
}
```

Поиск профиля по идентификатору:

```
{
     "matching": "profile_id",
     "profile_id": "012452gf62fcd71fjh8ec0"
}
```

Поиск профиля по значению дополнительного поля:

```
{
     "matching": "custom",
     "field_name": "CRM_ID",
     "field_value": "12345"
}
```

где `CRM_ID` — это название поля, а `12345` — значение этого поля у профиля.

TIP

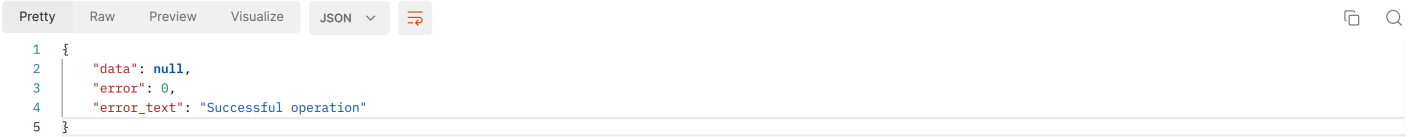
Больше информации о матчинге вы найдете в **этой статье**.

TIP

Обратите внимание на синтаксис вашего запроса. Лишние символы или незакрытые скобки могут привести к ошибке.

Отправка запроса

Чтобы отправить запрос, нажмите кнопку **Send** в Postman. При успешном выполнении вы получите сообщение об успешной отправке запроса:



Если что-то пошло не так, в ответном сообщении будет указан номер и описание ошибки:

Если у вас не получилось самостоятельно устранить ошибку, передайте номер и описание ошибки в **службу поддержки Altcraft**.

TIP

В статье описывается лишь часть возможных параметров для данного запроса. С остальными параметрами можно ознакомиться **здесь**.

API / Developer Guide: Содержание

Содержание раздела.

Взаимодействие с API ..... api-interaction.mdx

Матчинг 37848489.mdx

Список API-методов ..... api-endpoints-list.mdx

Импорт и настройка коллекции API-методов в Postman ..... api-collection.mdx

Импортировать профиль ..... profiles/import.mdx

Обновить профиль ..... profiles/update.mdx

Добавить профиль в базу данных profiles/2195477.mdx

Получить информацию о профиле profiles/2195482.mdx

Импортировать профиль в RabbitMQ profiles/2195492.mdx

Массовое добавление профилей в базу данных profiles/2195496.mdx

Массовое обновление профилей profiles/2195501.mdx

Массовый импорт профилей profiles/2195504.mdx

Приостановить все профили в базе данных profiles/2195553.mdx

Восстановить все приостановленные профили в базе данных profiles/2195556.mdx

Удалить профиль profiles/28936545.mdx

Упрощенный импорт профиля profiles/3113182.mdx

Функциональное обновление полей базы profiles/3113720.mdx

Функциональное обновление полей подписки profiles/3113721.mdx

Выгрузка профилей в файл profiles/3115796.mdx

Получение данных по нескольким профилям profiles/37847989.mdx

Объединение нескольких профилей profiles/41484440.mdx

Отписать профиль от ресурса profiles/41484441.mdx

Разделение профиля profiles/41484445.mdx

Получить историю одного профиля profiles/action-history/2195546.mdx

Получить историю нескольких профилей profiles/action-history/2195548.mdx

Добавить связь profiles/profile-relations/9406148.mdx

Удалить связь profiles/profile-relations/9406152.mdx

Усилить/ослабить связь profiles/profile-relations/9406155.mdx

Перезаписать значения свойств связи profiles/profile-relations/9406157.mdx

Получить информацию о связях профиля profiles/profile-relations/9406159.mdx
Получить список связей профиля ..... profiles/profile-relations/get-relations-list.mdx
Добавить или редактировать подписку profiles/subscriptions/2195519.mdx
Получить все подписки профиля profiles/subscriptions/2195522.mdx
Получить все подписки нескольких профилей profiles/subscriptions/2195524.mdx
Получить информацию о подписке профиля profiles/subscriptions/2195527.mdx
Удалить подписку профиля profiles/subscriptions/2195529.mdx
Восстановить удаленную подписку профиля profiles/subscriptions/2195531.mdx
Приостановить все подписки profiles/subscriptions/2195533.mdx
Восстановить все приостановленные подписки profiles/subscriptions/2195539.mdx
Получить список баз данных databases/2195563.mdx
Получить информацию о базе данных databases/2195565.mdx
Получить информацию о полях базы данных databases/2195567.mdx
Очистка базы данных для тестирования databases/2195815.mdx
Получить статистику по базе данных ..... databases/get-counters.mdx
Обновить статистику по базе данных ..... databases/refresh-counters.mdx
Получить список ресурсов resources/2195587.mdx
Получить информацию о ресурсе resources/2195590.mdx
Получить информацию о полях подписки ресурса resources/2195591.mdx
Получить статистику по ресурсам ..... resources/get-counters.mdx
Обновить статистику по ресурсам ..... resources/refresh-counters.mdx
Добавить сегмент segments/2195594.mdx
Обновить сегмент segments/2195596.mdx
Получить информацию о сегменте segments/2195598.mdx
Получить список сегментов segments/2195600.mdx
Удалить сегмент segments/2195602.mdx
Получить статистику по сегментам ..... segments/get-counters.mdx
Обновить статистику по сегментам ..... segments/refresh-counters.mdx
Добавить или удалить профиль segments/9407390.mdx
Получить данные профилей статического или обновляемого сегмента segments/9407394.mdx
Добавить стоп-список suppression-lists/2195613.mdx
Переименовать стоп-список suppression-lists/2195615.mdx

Получить информацию о стоп-списке suppression-lists/2195617.mdx
Получить информацию о нескольких стоп-списков suppression-lists/2195619.mdx
Удалить стоп-список suppression-lists/2195621.mdx
Выгрузить данные из стоп-списка в файл suppression-lists/76809125.mdx
Добавление и удаление из стоп-списка ..... suppression-lists/suppression-list-actions/index.mdx
Проверить email-адрес в стоп-списке suppression-lists/suppression-list-actions/2195633.mdx
Добавить email-адрес в стоп-список suppression-lists/suppression-list-actions/2195625.mdx
Добавить один или несколько email-адресов в стоп-список suppression-lists/suppression-list-actions/2195627.mdx
Удалить email-адрес из стоп-списка suppression-lists/suppression-list-actions/2195637.mdx
Удалить все email-адреса из стоп-списка suppression-lists/suppression-list-actions/2195641.mdx
Проверить домен в стоп-списке suppression-lists/suppression-list-actions/2195635.mdx
Добавить домен в стоп-список suppression-lists/suppression-list-actions/2195629.mdx
Добавить один или несколько доменов в стоп-список suppression-lists/suppression-list-actions/2195631.mdx
Удалить домен из стоп-списка suppression-lists/suppression-list-actions/2195639.mdx
Удалить все домены из стоп-списка suppression-lists/suppression-list-actions/2195643.mdx
Проверить номер телефона в стоп-списке suppression-lists/suppression-list-actions/76809084.mdx
Добавить номер телефона в стоп-список suppression-lists/suppression-list-actions/76809087.mdx
Добавить один или несколько номеров в стоп-список suppression-lists/suppression-list-actions/76809091.mdx
Удалить номер из стоп-списка suppression-lists/suppression-list-actions/76809096.mdx
Удалить все номера из стоп-списка suppression-lists/suppression-list-actions/76809100.mdx
Получить список шаблонов templates/2195606.mdx
Получить информацию о шаблоне templates/2195608.mdx
Удалить шаблон templates/56255090.mdx
Добавить шаблон сообщения templates/56255095.mdx
Обновить шаблон сообщения templates/56255277.mdx
Channel object ..... templates/Channel-object_56255164.mdx
Получить список рассылок campaigns/2195648.mdx
Получить информацию о рассылке campaigns/2195650.mdx
Получить лог рассылки campaigns/2195652.mdx
Клонировать рассылку campaigns/2195654.mdx
Удалить рассылку campaigns/2195656.mdx
Активировать рассылку campaigns/2195660.mdx

Деактивировать рассылку campaigns/2195662.mdx
Получить статус рассылки campaigns/2195664.mdx
Получить список броадкаст рассылок campaigns/broadcast-campaigns/2195885.mdx
Получить информацию о броадкаст рассылке campaigns/broadcast-campaigns/2195887.mdx
Добавить броадкаст рассылку campaigns/broadcast-campaigns/2195891.mdx
Обновить броадкаст рассылку campaigns/broadcast-campaigns/2195895.mdx
Запустить броадкаст рассылку campaigns/broadcast-campaigns/3116196.mdx
Получить список регулярных рассылок campaigns/regular-campaigns/2195899.mdx
Получить информацию о регулярной рассылке campaigns/regular-campaigns/2195901.mdx
Добавить регулярную рассылку campaigns/regular-campaigns/2195903.mdx
Обновить регулярную рассылку campaigns/regular-campaigns/2195905.mdx
Запустить регулярную рассылку campaigns/regular-campaigns/3116201.mdx
Получить список триггерных рассылок campaigns/trigger-campaigns/2195915.mdx
Получить информацию о триггерной рассылке campaigns/trigger-campaigns/2195917.mdx
Добавить триггерную рассылку campaigns/trigger-campaigns/2195922.mdx
Обновить триггерную рассылку campaigns/trigger-campaigns/2195925.mdx
Запуск триггерной рассылки (API call) campaigns/trigger-campaigns/2196177.mdx
Импорт профиля + Отправка триггера campaigns/trigger-campaigns/22708362.mdx
Задание на массовую отправку триггера campaigns/trigger-campaigns/33063451.mdx
Задание на массовый импорт профилей + отправка триггера campaigns/trigger-campaigns/33063474.mdx
Массовая отправка триггера campaigns/trigger-campaigns/37847471.mdx
Массовый импорт профилей + отправка триггера campaigns/trigger-campaigns/37847473.mdx
Клонировать триггер рассылку campaigns/trigger-campaigns/56258725.mdx
Data array ..... campaigns/trigger-campaigns/Data-array_41484531.mdx
Получить информацию о кампании ..... super-campaigns/get-scampaigns-info.mdx
Получить список кампаний ..... super-campaigns/get-scampaigns-list.mdx
Активация кампании ..... super-campaigns/scampaigns-activation.mdx
Завершение кампании ..... super-campaigns/scampaigns-complete.mdx
Деактивация кампании ..... super-campaigns/scampaigns-deactivation.mdx
Получить статус кампании ..... super-campaigns/scampaigns-status.mdx
Отправить профиль клиента в сценарий automation-scenarios/2195668.mdx
Одновременный импорт и запуск профиля в сценарий automation-scenarios/28934227.mdx

Массовый импорт и запуск профилей в сценарий automation-scenarios/33063492.mdx

Задание на массовый импорт и запуск профилей в сценарий automation-scenarios/33063493.mdx

Получить список сценариев automation-scenarios/95715617.mdx

Активировать сценарий ..... automation-scenarios/activate-workflow.mdx

Деактивировать сценарий ..... automation-scenarios/deactivate-workflow.mdx

Импортировать промокоды promocodes/2195573.mdx

Получить информацию о промокоде promocodes/2195575.mdx

Активировать промокод promocodes/2195577.mdx

Обновить промокод promocodes/2195579.mdx

Привязать промокод к профилю promocodes/2195581.mdx

Отвязать промокод от профиля promocodes/2195583.mdx

Получить все промокоды promocodes/3113099.mdx

Получить уровень профиля в программе лояльности ..... loyalty/tiers.mdx

Экспорт транзакций баллов ..... loyalty/transactions-export.mdx

Сгораемые баллы за период ..... loyalty/expiring-points.mdx

Получение транзакций по счёту профиля ..... loyalty/transactions-list.mdx

Получение списка триггерных промоакций ..... loyalty/trigger-promotions.mdx

Начисление баллов участнику ..... loyalty/accrue-points.mdx

Списание баллов участника ..... loyalty/redeem-points.mdx

Подтверждение временной транзакции ..... loyalty/commit-transaction.mdx

Предварительный расчет заказа ..... loyalty/preorder.mdx

Подтверждение заказа ..... loyalty/preorder-commit.mdx

Отмена временной транзакции ..... loyalty/rollback-transaction.mdx

Отмена балльной транзакции ..... loyalty/cancel-transation.mdx

Получение баланса балльного счёта ..... loyalty/balance.mdx

Регистрация участника в программе лояльности ..... loyalty/register-member.mdx

Удаление участника из программы лояльности ..... loyalty/delete-member.mdx

Регистрация события достижения цели goals/2195843.mdx

Обработка и добавление подписки application-push-notifications/2196103.mdx

Добавить события с app push application-push-notifications/2196102.mdx

Структура заказа (order data object) market/market-objects/56249763.mdx

Product data object ..... market/market-objects/Product-data-object_56249759.mdx

Структура SKU (SKU data object) ..... market/market-objects/SKU-data-object_56249769.mdx
Категории (categories array) ..... market/market-objects/Categories-array_56249780.mdx
Custom fields array ..... market/market-objects/Custom-fields-array_56249775.mdx
Импорт заказа и статусов позиций market/orders/56249746.mdx
Получить список заказов market/orders/56249749.mdx
Удалить заказ market/orders/76808865.mdx
Получить статус заказа ..... market/orders/get-status.mdx
Изменение статуса позиции заказа ..... market/orders/update-order-line-status.mdx
Импорт продуктов, SKU и категорий market/products-and-sku/56249718.mdx
Получение списка продуктов market/products-and-sku/56249723.mdx
Получение списка SKU market/products-and-sku/56249724.mdx
Импорт SKU и категорий market/products-and-sku/56249733.mdx
Удалить продукты ..... market/products-and-sku/delete-product.mdx
Удалить SKU ..... market/products-and-sku/delete-sku.mdx
Получить сводный отчет analytic-reports/2195743.mdx
Получить отчет о возвратах analytic-reports/2195748.mdx
Получить отчет о недоставках analytic-reports/2195750.mdx
Получить список сендеров senders/2195754.mdx
Получить список виртуальных сендеров senders/virtual-senders/2195762.mdx
Получить информацию о виртуальном сендере senders/virtual-senders/2195765.mdx
Клонировать виртуальный сендер senders/virtual-senders/2195767.mdx
Добавить виртуальный сендер senders/virtual-senders/2195769.mdx
Обновить виртуальный сендер senders/virtual-senders/2195771.mdx
Удалить виртуальный сендер senders/virtual-senders/2195773.mdx
AKMTA object ..... objects/AKMTA-object_2195811.mdx
Content object ..... objects/Content-object_2195781.mdx
Email rule object ..... objects/Email-rule-object_2195791.mdx
File object ..... objects/File-object_2195839.mdx
Profile data object ..... objects/Profile-data-object_3116008.mdx
Sender object ..... objects/Sender-object_2195784.mdx
Sender types ..... objects/Sender-types_2195789.mdx
SMS rule object ..... objects/SMS-rule-object_2195796.mdx

Start schedule object ..... objects/Start-schedule-object_2195802.mdx

Subscription object ..... objects/Subscription-status_2195807.mdx

Trigger types ..... objects/Trigger-types_2195809.mdx

Добавить запрос сегментации ..... datatables/query_segments/add.mdx

Обновить запрос сегментации ..... datatables/query_segments/update.mdx

Получить информацию о запросе сегментации ..... datatables/query_segments/get.mdx

Получить список запросов на сегментацию ..... datatables/query_segments/list.mdx

Удалить запрос сегментации ..... datatables/query_segments/delete.mdx

Добавить запрос для шаблонов ..... datatables/query_templates/add.mdx

Обновить запрос для шаблонов ..... datatables/query_templates/update.mdx

Получить информацию о запросе для шаблонов ..... datatables/query_templates/get.mdx

Получить список запросов для шаблонов ..... datatables/query_templates/list.mdx

Удалить запрос для шаблонов ..... datatables/query_templates/delete.mdx

Загрузить файл miscellaneous/2195777.mdx

Получить веб-версию сообщения miscellaneous/2196089.mdx

Push провайдеры miscellaneous/37848490.mdx

Дедупликация запросов miscellaneous/41484312.mdx

Работа с API через RabbitMQ miscellaneous/44662785.mdx

Список гендерных идентификаций miscellaneous/9405775.mdx

Получить допустимые значения полей browsers, devices, tz, oses, languages ..... miscellaneous/dictionary.mdx

mSDK ..... sdk/mobile/index.mdx

Работа с ролевым и JWT-токеном ..... sdk/mobile/jwt-r-token.mdx

Быстрый старт ..... sdk/mobile/android/quick-start.mdx

Конфигурация SDK ..... sdk/mobile/android/setup.mdx

Функционал SDK ..... sdk/mobile/android/functionality.mdx

Публичный API SDK ..... sdk/mobile/android/api.mdx

Firebase Cloud Messaging ..... sdk/mobile/android/providers/fcm.mdx

Huawei Mobile Services ..... sdk/mobile/android/providers/hms.mdx

RuStore ..... sdk/mobile/android/providers/rustore.mdx

Быстрый старт ..... sdk/mobile/ios/quick-start.mdx

Конфигурация SDK ..... sdk/mobile/ios/setup.mdx

Функционал SDK ..... sdk/mobile/ios/functionality.mdx

Публичный API SDK ..... sdk/mobile/ios/api.mdx

Apple Push Notification Service ..... sdk/mobile/ios/providers/apns.mdx

Firebase Cloud Messaging ..... sdk/mobile/ios/providers/fcm.mdx

Huawei Mobile Services ..... sdk/mobile/ios/providers/hms.mdx

Быстрый старт ..... sdk/mobile/react-native/quick-start.mdx

Конфигурация SDK ..... sdk/mobile/react-native/setup.mdx

Функционал SDK ..... sdk/mobile/react-native/functionality.mdx

Публичный API SDK ..... sdk/mobile/react-native/api.mdx

Настройка провайдеров ..... sdk/mobile/react-native/providers.mdx

Взаимодействие с API

TIP

По **этой ссылке** доступна коллекция API Altcraft для Postman.

Формат запросов

Данные в запросах передаются методом POST по протоколу HTTP/1.1. Метод запроса также указан на странице с описанием каждого запроса. Параметры запроса помещаются в передаваемую структуру. Часть параметров может быть передана в URL (API key, format).

Формат входных данных указывается в HTTP-заголовке Content-Type.

Возможные значения заголовка:

- application/json — формат JSON
- application/xml — формат XML

Кодировка символов — UTF-8

```
POST https://example.com/api/v1.1/<имя метода>
```

Если вы работаете в **облачном** аккаунте Altcraft Platform, используйте URL-адрес панели управления: `ru.altkraft.com`. При установке платформы на ваше оборудование (**on-Premise**) — используйте ваш собственный URL.

Описание формата JSON вы найдёте в RFC 8259: https://www.rfc-editor.org/rfc/rfc8259.

Не забывайте кодировать символы, которые не могут быть непосредственно записаны в JSON.

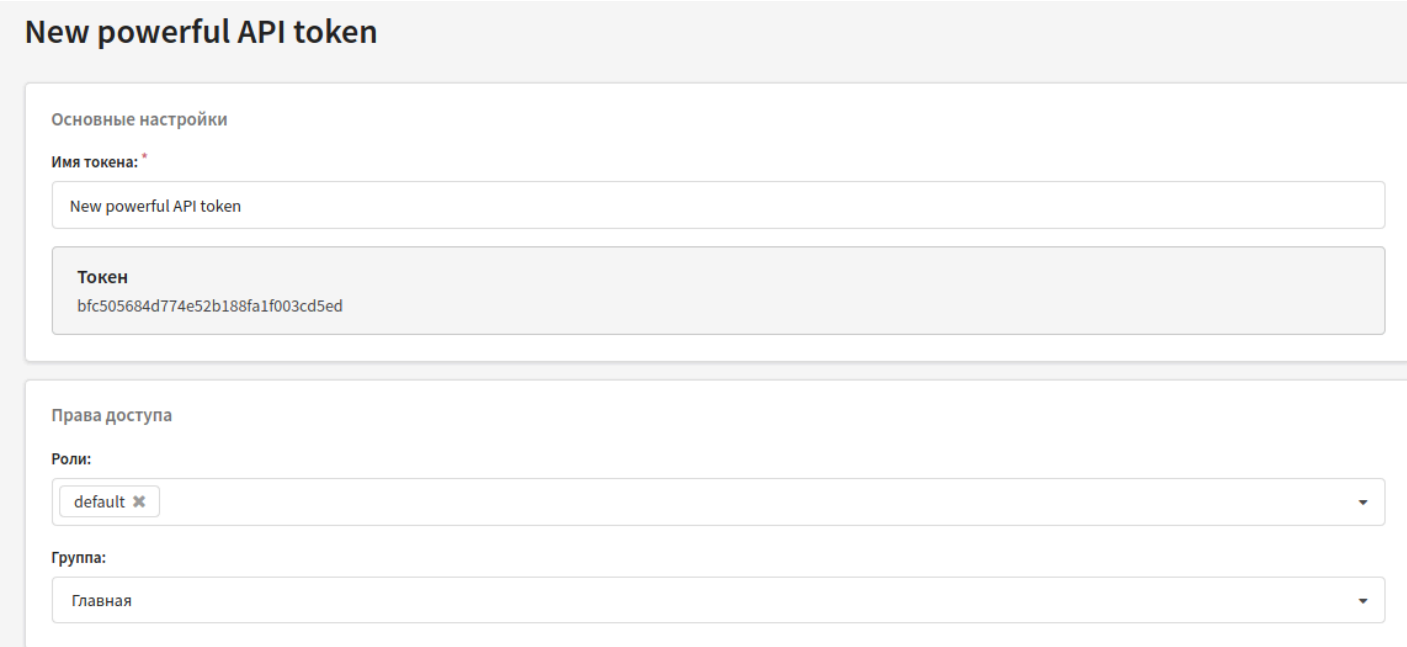
Авторизация

Для авторизации при отправке запроса вам необходимо передать в платформу **токен**. Для этого существует несколько способов, например, указать его в теле запроса:

```
{
  "token": "bfc505684d774e52b188fa1f003cd5ed",
  "db_id": 1,
  "resource_id": 1,
  "matching": "email",
  "email": "example@example.com",
  "data": {
    "_status": 0,
    "_fname": "Jim",
    "_lname": "Jones",
    "email": "example@example.com",
    "phones": ["+79000000000"]
  }
}
```

Также вы можете передать токен в одном из **заголовков запроса**. Для этого создайте заголовок с ключом **Authorization**, а в его значении укажите `Bearer <token>`, где вместо `<token>` вставьте ваш токен.

Токен может быть создан в панели пользователя платформы, раздел “Настройки” — “Токены”. Для создания токена необходимы права мастер-пользователя:



Токен генерируется автоматически после сохранения. Также в данном интерфейсе можно задать имя токена, и настроить права доступа (уточняются в роли) пользователей к данному токену, и группу, в рамках которой токену доступны объекты.

CAUTION

Мы **не рекомендуем** передавать токен в URL вашего запроса. Этот метод небезопасен и в будущем не будет поддерживаться платформой.

Формат ответа

Формат ответа может быть передан в заголовке или в параметрах запроса.

Пример ответа (успешное выполнение операции):

```
{
  "error": 0,
  "error_text": "Successful operation",
  "profile_id": "5f4fa1a5ce9448665fef548e"
}
```

В ответе возвращаются следующие параметры:

- `error` — код ошибки
- `error_text` — описание ошибки
- `profile_id` — идентификатор профиля, к которому был применены действия согласно запросу. Получается при успешном выполнении операции.

INFO

Код 200 — это код успеха на HTTP транспорте. Однако, если внутри HTTP-транспорта есть информация об ошибке, то дополнительные детали об ошибке предоставляются в JSON-объекте в поле `error` с пояснениями в `error_text`.

Коды ответа

Код	Описание
0	Операция выполнена успешно
400	Некорректный запрос
401	Требуется API-токен
402	Достигнуто ограничение тарифа
403	Нет прав на действие
404	Объект не найден в зоне видимости токена
409	Существует другая запись с такими же признаками уникальности
413	База данных недоступна в выбранном ресурсе
415	Запрашиваемый Content-Type не поддерживается
429	Превышено количество запросов, установленное в файле конфигурации (параметр `API_MAX_REQUESTS_COUNT`)
435	Неоднозначный поиск, есть несколько объектов с заданными характеристиками
441	Объект принадлежит другой группе

Код	Описание
450	Ошибка валидации запроса
500	Внутренняя ошибка сервиса
501	Такой метод отсутствует

Атрибуты тегов XML

XML-запросы строятся на основе JSON-запросов, в связи с этим в некоторых ситуациях требуется вписывать **дополнительные атрибуты** внутри тегов:

- `array='true'` — атрибут, указывающий на то, что тег является **массивом**. Используется в случаях, когда внутри массива находится либо 1 элемент, либо элементов нет совсем.
- `string='true'` — атрибут используется для указания на то, что используемое значение является **строкой** (тип данных **string**). Необходим в случаях, когда в строке находится число или слово “true”/“false”.
- `json='true'` — атрибут, необходимый в тех случаях, когда внутри xml-запроса мы должны передать **json-запрос** (например, параметры отбора при запросе на получение данных по нескольким профилям).

Дедупликация запросов

Если в момент получения данных произошёл сбой соединения, может быть отправлен повторный запрос. Платформа не будет воспринимать повторный запрос, если он приводит к изменению данных, чтобы исключить дублирование событий. Подробнее о повторных запросах читайте здесь.

Матчинг

Поле `matching` определяет режим поиска подписчика для обновления. По умолчанию — `email`.

При обновлении данных следует применять общий подход к матчингу.

Режим поиска подписчика	matching	Необходимые поля	Тип поля	Пример	Описание поля
По email адресу из профиля или подписок	"email"	email	string	"john@example.com"	Email-адрес
По email адресу из профиля	"email_profile"	email	string	"john@example.com"	Email-адрес
По email адресу из подписок	"email_sub"	email	string	"john@example.com"	Email-адрес
По телефону из полей данных или подписок	"phone"	phone	string	"+79000000000"	Номер телефона
По телефону из полей подписок	"phone_sub"	phone	string	"+79000000000"	Номер телефона
По идентификатору профиля (только при обновлении!)	"profile_id"	profile_id	string	"abcdefghijklmnqrstuvwxyz"	Идентификатор профиля
Поиск по пуш подписке	"push_sub"	provider	string	"android-firebase"	Провайдер
		subscription_id	string	"1234"	Идентификатор подписки
По пользовательскому полю	"custom"	field_name	string	"CRM_ID"	Название пользовательског поля профиля для поиска.
		field_value	int/string	"example123"	Значение пользовательского поля для поиска. Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке через запятую: "тег_1, тег_2".
Поиск профиля по подписке на пользовательский канал	"custom_sub"	field_name	string	"CRM_ID"	Название пользовательского поля профиля для поиска.
		field_value	int/string	"example123"	Значение пользовательского поля для поиска.
По email адресу или телефону из профиля	"email_phone"	email / phone	string	"john@example.com" / "+79000000000"	Email-адрес / Номер телефона

По email адресу или телефону из подписок	"email_phone_sub"	email / phone	string	"john@example.com" / "+79000000000"	Email-адрес / Номер телефона

INFO

Unclear matching

Если поле, которое вы передаете в теле запроса, не является уникальным, т.е. значение этого поля совпадает у нескольких профилей, то запрос вернет ошибку **“Unclear matching”**.

INFO

Duplicate unique data

Ошибка “**Duplicate unique data**” возникает, если при импорте профиля вы используете пользовательское уникальное поле для матчинга и при этом нарушаете уникальность системного поля.

Например, для матчинга используется уникальное пользовательское поле `client_id`. В базе есть профиль с полями: *email: test@example.com*, *client_id: 100*. Вы импортируете ещё один профиль с полями: *email: test@example.com*, *client_id: 101*. Запрос вернет ошибку “Duplicate unique data”, потому что нарушается уникальность системного поля `email`.

В таких случаях рекомендуется передавать поле email в объекте `subscription` (подписка):

```
{
    "token": "m1883gf9cacngh18c912016r7c7995a",
    "db_id": 1,
    "matching": "custom",
    "field_name": "client_id",
    "field_value": "100",
    "data": {
            "_fname": "John",
            "_lname": "Doe",
            "subscriptions": [
                {
                    "channel": "email",
                    "email": "test@example.com",
                    "resource_id": 1
                }
            ]
        }
}
```

Пример импорта/обновления подписки при определенном матчинге

email, email_profile, email_sub, email_phone, email_phone_sub

JSON

```
{
     "token": "m1883gf9cacngh18c912016r7c7995a",
     "db_id": 1,
     "matching": "email",
     "email": "example@example.com",
     "data": {
        "subscriptions": [
            {
                "channel": "email",
                "email": "example@example.com",
                "resource_id": 1
            }
        ]
     }
}
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <token>m1883gf9cacngh18c912016r7c7995a</token>
  <db_id>1</db_id>
  <matching>email</matching>
  <email>example@example.com</email>
  <data>
    <subscriptions>
      <list>
        <channel>email</channel>
        <email>example@example.com</email>
        <resource_id>1</resource_id>
      </list>
    </subscriptions>
  </data>
</xml>
```

phone, phone_sub, email_phone, email_phone_sub

JSON

```
{
     "token": "m1883gf9cacngh18c912016r7c7995a",
     "db_id": 1,
     "matching": "phone",
     "phone": "+79000000000",
     "data": {
        "subscriptions": [
                        {
                "channel": "sms",
                "phone": "+79000000000",
                "resource_id": 1
            }
        ]
     }
}
```


XML


```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <token>m1883gf9cacngh18c912016r7c7995a</token>
  <db_id>1</db_id>
  <matching>phone</matching>
  <phone>+79000000000</phone>
  <data>
    <subscriptions>
      <list>
        <channel>sms</channel>
        <phone>+79000000000</phone>
        <resource_id>1</resource_id>
      </list>
    </subscriptions>
  </data>
</xml>
```


profile_id

JSON

```
{
     "token": "91f1dfa81c264a938b475677c60ce967",
     "db_id": 1,
     "matching": "profile_id",
     "profile_id": "012452gf62fcd71fjh8ec0",
     "data": {
        "subscriptions": [
            {
                "channel": "email",
                "email": "example@example.com",
                "resource_id": 1
            }
        ]
    }
}
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <token>91f1dfa81c264a938b475677c60ce967</token>
  <db_id>1</db_id>
  <matching>profile_id</matching>
  <profile_id>012452gf62fcd71fjh8ec0</profile_id>
  <data>
    <subscriptions>
      <list>
        <channel>email</channel>
        <email>example@example.com</email>
        <resource_id>1</resource_id>
      </list>
    </subscriptions>
  </data>
</xml>
```

push_sub

JSON

```
{
     "token": "91f1dfa81c264a938b475677c60ce967",
     "db_id": 1,
     "matching": "push_sub",
     "provider": "android-firebase",
     "subscription_id": "<token>",
     "data": {
        "subscriptions": [
            {
                "channel": "push",
                "subscription_id": "<token>",
                "provider": "android-firebase",
                "resource_id": 1
            }
        ]
    }
}
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <token>91f1dfa81c264a938b475677c60ce967</token>
  <db_id>1</db_id>
  <matching>push_sub</matching>
  <provider>android-firebase</provider>
  <subscription_id>&lt;token&gt;</subscription_id>
  <data>
    <subscriptions>
      <list>
        <channel>push</channel>
        <subscription_id>&lt;token&gt;</subscription_id>
        <provider>android-firebase</provider>
        <resource_id>1</resource_id>
      </list>
    </subscriptions>
  </data>
</xml>
```

custom, custom_sub

JSON

```
{
     "token": "91f1dfa81c264a938b475677c60ce967",
     "db_id": 1,
         "matching": "custom",
     "field_name": "CRM_id",
     "field_value": "12345",
     "data": {
        "subscriptions": [
            {
                "channel": "email",
                "email": "example@example.com",
                "resource_id": 1
            }
        ]
    }
}
```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<xml>
  <token>91f1dfa81c264a938b475677c60ce967</token>
  <db_id>1</db_id>
  <matching>custom</matching>
  <field_name>CRM_id</field_name>
  <field_value>12345</field_value>
  <data>
    <subscriptions>
      <list>
        <channel>email</channel>
        <email>example@example.com</email>
        <resource_id>1</resource_id>
      </list>
    </subscriptions>
  </data>
</xml>
```

TIP

Объект subscriptions

Объект `subscription` передаёт данные о подписках профиля на ресурс. Подробнее.

Список API-методов

Профиль

Импорт и обновление профилей	
Импортировать профиль	/api/v1.1/profiles/import
Массовый импорт профилей	/api/v1.1/profiles/import_batch
Добавить профиль в базу данных	/api/v1.1/profiles/add
Массовое добавление профилей в базу данных	/api/v1.1/profiles/add_batch
Упрощенный импорт профиля	/api/integrations/any/profile_import
Обновить профиль	/api/v1.1/profiles/update
Массовое обновление профилей	/api/v1.1/profiles/update_batch
Получить информацию о профиле	/api/v1.1/profiles/get
Получение данных по нескольким профилям	/api/v1.1/profiles/list
Выгрузка профилей в файл	/api/v1.1/profiles/get_all
Удалить профиль	/api/v1.1/profiles/delete
Объединение нескольких профилей	/api/v1.1/profiles/merge
Разделение профиля	/api/v1.1/profiles/split
Отписать профиль от ресурса	/api/v1.1/profiles/message_unsub
Приостановить все профили в базе данных	/api/v1.0/lists/{list_id}/suspend_all?token={token}
Восстановить все приостановленные профили в базе данных	/api/v1.0/lists/{list_id}/unsuspend_all?token={token}
История	
Получить историю одного профиля	/api/v1.1/subscribers/history_get
Получить историю нескольких профилей	/api/v1.1/subscribers/history_get_all
Связи профилей	
Добавить связь	/api/v1.1/relations/attach
Удалить связь	/api/v1.1/relations/detach
Усилить/ослабить связь	/api/v1.1/relations/strengthen
Перезаписать значения свойств связи	m/api/v1.1/relations/edit
Получить информацию о связях профиля	/api/v1.1/relations/get_counter
Подписки	
Добавить или редактировать подписку	/api/v1.1/subscriptions/set

Получить все подписки профиля	/api/v1.1/subscriptions/list
Получить все подписки нескольких профилей	/api/v1.1/subscriptions/list_all
Получить информацию о подписке профиля	/api/v1.1/subscriptions/get
Удалить подписку профиля	/api/v1.1/subscriptions/delete
Восстановить удаленную подписку профиля	/api/v1.1/subscriptions/undelete
Приостановить все подписки	/api/v1.1/subscriptions/suspend_all
Восстановить все приостановленные подписки	/api/v1.1/subscriptions/unsuspend_all

Базы данных

Получить список баз данных	/api/v1.1/databases/list
Получить информацию о базе данных	/api/v1.1/databases/get
Получить информацию о полях базы данных	/api/v1.1/databases/fields_get
Очистка базы данных для тестирования	/api/v1.1/databases/wipe

Ресурсы

Получить список ресурсов	/api/v1.1/resources/list
Получить информацию о ресурсе	/api/v1.1/resources/get/
Получить информацию о полях подписки ресурса	/api/v1.1/resources/fields_get/

Сегменты

Управление сегментами	
Добавить сегмент	/api/v1.1/segments/add
Обновить сегмент	/api/v1.1/segments/update
Получить информацию о сегменте	/api/v1.1/segments/get
Получить список сегментов	/api/v1.1/segments/list
Удалить сегмент	/api/v1.1/segments/delete
Обновить статистику подписок в сегменте	/api/v1.1/segments/refresh_counters
Получить статистику подписок профилей в сегменте	/api/v1.1/segments/get_counters
Управление профилями в статическом сегменте	
Добавить профиль в сегмент	/api/v1.1/profiles/segstatic/add
Удалить профиль из сегмент	/api/v1.1/profiles/segstatic/remove
Получить данные профилей статического сегмента	/api/v1.1/profiles/segstatic/list

Стоп-списки

Управление стоп-списками	
Добавить стоп-список	/api/v1.1/suppresses/add
Переименовать стоп-список	/api/v1.1/suppresses/update
Получить информацию о стоп-списке	/api/v1.1/suppresses/get
Получить информацию о нескольких стоп-списках	/api/v1.1/suppresses/list
Удалить стоп-список	/api/v1.1/suppresses/delete
Выгрузить данные из стоп-списка в файл	/api/v1.1/suppresses/export
Добавление, проверка и удаление email-адреса	
Проверить email-адрес в стоп-списке	/api/v1.1/suppresses/email_check
Добавить email-адрес в стоп-список	/api/v1.1/suppresses/email_add
Добавить один или несколько email-адресов в стоп-список	/api/v1.1/suppresses/emails_add
Удалить email-адрес из стоп-списка	/api/v1.1/suppresses/email_remove
Удалить все email-адреса из стоп-списка	/api/v1.1/suppresses/emails_remove
Добавление, проверка и удаление email-домена	
Проверить домен в стоп-списке	/api/v1.1/suppresses/domain_check
Добавить домен в стоп-список	/api/v1.1/suppresses/domain_add
Добавить один или несколько доменов в стоп-список	/api/v1.1/suppresses/domains_add
Удалить домен из стоп-списка	/api/v1.1/suppresses/domain_remove
Удалить все домены из стоп-списка	/api/v1.1/suppresses/domains_remove
Добавление, проверка и удаление номера телефона	
Проверить номер телефона в стоп-списке	/api/v1.1/suppresses/phone_check
Добавить номер телефона в стоп-список	/api/v1.1/suppresses/phone_add
Добавить один или несколько номеров в стоп-список	/api/v1.1/suppresses/phones_add
Удалить номер из стоп-списка	/api/v1.1/suppresses/phone_remove
Удалить все номера из стоп-списка	/api/v1.1/suppresses/phones_remove

Шаблоны

Добавить шаблон сообщения	/api/v1.1/templates/add
Обновить шаблон сообщения	/api/v1.1/templates/update
Получить информацию о шаблоне	/api/v1.1/templates/get
Получить список шаблонов	/api/v1.1/templates/list
Удалить шаблон	/api/v1.1/templates/delete

Рассылкии

Управление рассылками	
Активировать рассылку	/api/v1.1/campaigns/activate
Деактивировать рассылку	/api/v1.1/campaigns/deactivate
Получить информацию о рассылке	/api/v1.1/campaigns/get
Получить статус рассылки	/api/v1.1/campaigns/get_status
Получить лог рассылки	/api/v1.1/campaigns/log_v2
Получить список рассылок	/api/v1.1/campaigns/list
Клонировать рассылку	/api/v1.1/campaigns/clone
Удалить рассылку	/api/v1.1/campaigns/delete
Броадкаст рассылки	
Добавить броадкаст рассылку	/api/v1.1/campaigns/broadcasts/add
Обновить броадкаст рассылку	/api/v1.1/campaigns/broadcasts/update
Запустить броадкаст рассылку	/api/v1.1/campaigns/broadcasts/start
Получить информацию о броадкаст рассылке	/api/v1.1/campaigns/broadcasts/get
Получить список броадкаст рассылок	/api/v1.1/campaigns/broadcasts/list
Регулярные рассылки	
Добавить регулярную рассылку	/api/v1.1/campaigns/regulars/add/
Обновить регулярную рассылку	/api/v1.1/campaigns/regulars/update
Запустить регулярную рассылку	/api/v1.1/campaigns/regulars/start
Получить информацию о регулярной рассылке	/api/v1.1/campaigns/regulars/get
Получить список регулярных рассылок	/api/v1.1/campaigns/regulars/list
Триггеры	
Добавить триггер рассылку	/api/v1.1/campaigns/triggers/add
Обновить триггер рассылку	/api/v1.1/campaigns/triggers/update
Клонировать триггер рассылку	/api/v1.1/campaigns/triggers/clone
Запуск триггерной рассылки (API call)	/api/v1.1/campaigns/triggers/start
Массовая отправка триггера	/api/v1.1/campaigns/triggers/start_batch
Импорт профиля + Отправка триггера	/api/v1.1/campaigns/triggers/import_and_start
Массовый импорт профилей + отправка триггера	/api/v1.1/campaigns/triggers/import_and_start_batch
Задание на массовую отправку триггера	/api/v1.1/campaigns/triggers/start_batch_task

Задание на массовый импорт профилей + отправка триггера	/api/v1.1/campaigns/triggers/import_and_start_batch_task
Получить информацию о триггерной рассылке	/api/v1.1/campaigns/triggers/get
Получить список триггерных рассылок	/api/v1.1/campaigns/triggers/list

Кампании

Управление кампаниями	
Получить информацию о кампании	/api/v1.1/scampaigns/get
Получить список кампаний	/api/v1.1/scampaigns/list
Активировать кампанию	/api/v1.1/scampaigns/activate
Деактивировать кампанию	/api/v1.1/scampaigns/deactivate
Завершить кампанию	/api/v1.1/scampaigns/completed
Получить статус кампании	/api/v1.1/scampaigns/get_status

Сценарии

Отправить профиль клиента в сценарий	/api/v1.1/workflows/start
Одновременный импорт и запуск профиля в сценарий	/api/v1.1/workflows/import_and_start
Массовый импорт и запуск профилей в сценарий	/api/v1.1/workflows/import_and_start_batch
Задание на массовый импорт и запуск профилей в сценарий	/api/v1.1/workflows/import_and_start_batch_task
Получить список сценариев	/api/v1.1/workflows/list

Промокоды

Импортировать промокоды	/api/v1.1/promocodes/import
Получить информацию о промокоде	/api/v1.1/promocodes/get
Активировать промокод	/api/v1.1/promocodes/activate
Обновить промокод	api/v1.1/promocodes/update
Привязать промокод к профилю	/api/v1.1/promocodes/attach
Отвязать промокод от профиля	/api/v1.1/promocodes/detach
Получить все промокоды	/api/v1.1/promocodes/export_all

Программа лояльности

Управление баллами и транзакциями	
Получить баланс балльного счёта	/api/v1.1/loyalty/account/balance
Получить транзакции по счёту профиля	/api/v1.1/loyalty/account/transactions

Сгораемые баллы за период	/api/v1.1/loyalty/account/expiring_points
Получить уровень профиля в программе лояльности	/api/v1.1/loyalty/account/tiers
Начисление баллов участнику	/api/v1.1/loyalty/accrue_points
Списание баллов участника	/api/v1.1/loyalty/redeeme_points
Отмена балльной транзакции	/api/v1.1/loyalty/cancel_transaction
Экспорт транзакций баллов	/api/v1.1/loyalty/transactions_export
Управление участниками	
Регистрация участника в программе лояльности	/api/v1.1/loyalty/register_member
Удаление участника из программы лояльности	/api/v1.1/loyalty/delete_member
Временные транзакции и предзаказы	
Получение списка триггерных промоакций	/api/v1.1/loyalty/trigger_promotions
Предзаказ	/api/v1.1/market/orders/preorder
Подтверждение заказа	/api/v1.1/market/orders/preorder_commit
Подтверждение временной транзакции	/api/v1.1/loyalty/commit_transaction
Отмена временной транзакции	/api/v1.1/loyalty/rollback_transaction

Цели

Регистрация события достижения цели	/api/v1.1/goals/register

Пуши

Добавить события с app push	/api/v1.1/pushes/post_events

Маркет

Заказы	
Импорт заказа и статусов позиций	/api/v1.1/market/orders/import
Получить список заказов	/api/v1.1/market/orders/list
Удалить заказ	/api/v1.1/market/orders/delete
Изменение статуса позиции заказа	/api/v1.1/market/orders/update_order_line_status
Продукты и SKU	
Импорт продуктов, SKU и категорий	/api/v1.1/market/products/import
Импорт SKU и категорий	/api/v1.1/market/sku/import
Получение списка продуктов	/api/v1.1/market/products/list
Получение списка SKU	/api/v1.1/market/sku/list

Отчет и статистика

Получить сводный отчет	/api/v1.1/reports/summary
Получить отчет о возвратах	/api/v1.1/reports/bounces
Получить отчет о недостатках	/api/v1.1/reports/undelivered

Сендеры

Сендеры	
Получить список сендеров	/api/v1.1/senders/list
Виртуальные сендеры	
Добавить виртуальный сендер	/api/v1.1/vsenders/add
Обновить виртуальный сендер	/api/v1.1/vsenders/update
Получить информацию о виртуальном сендере	/api/v1.1/vsenders/get
Получить список виртуальных сендеров	/api/v1.1/vsenders/list
Клонировать виртуальный сендер	/api/v1.1/vsenders/clone
Удалить виртуальный сендер	/api/v1.1/vsenders/delete

Запросы к внешним таблицам данных

Запросы сегментации	
Добавить запрос сегментации	/api/v1.1/datatables/query_segments/add
Обновить запрос сегментации	/api/v1.1/datatables/query_segments/update
Удалить запрос сегментации	/api/v1.1/datatables/query_segments/delete
Получить список запросов сегментации	/api/v1.1/datatables/query_segments/list
Получить информацию о запросе сегментации	/api/v1.1/datatables/query_segments/get
Запросы для шаблонов	
Добавить запрос для шаблонов	/api/v1.1/datatables/query_templates/add
Обновить запрос для шаблонов	/api/v1.1/datatables/query_templates/update
Удалить запрос для шаблонов	/api/v1.1/datatables/query_templates/delete
Получить список запросов для шаблонов	/api/v1.1/datatables/query_templates/list
Получить информацию о запросе для шаблонов	/api/v1.1/datatables/query_templates/get

Прочее

Загрузить файл	/api/v1.1/file_upload
Получить веб-версию сообщения	/api/v1.1/webver/get

Импорт и настройка коллекции API-методов в Postman

Инструкция поможет вам импортировать и настроить коллекцию API-методов для работы с Altcraft Platform через Postman. Следуя шагам ниже, вы сможете быстро загрузить API-методы, настроить переменные среды и начать отправлять тестовые запросы для взаимодействия с платформой.

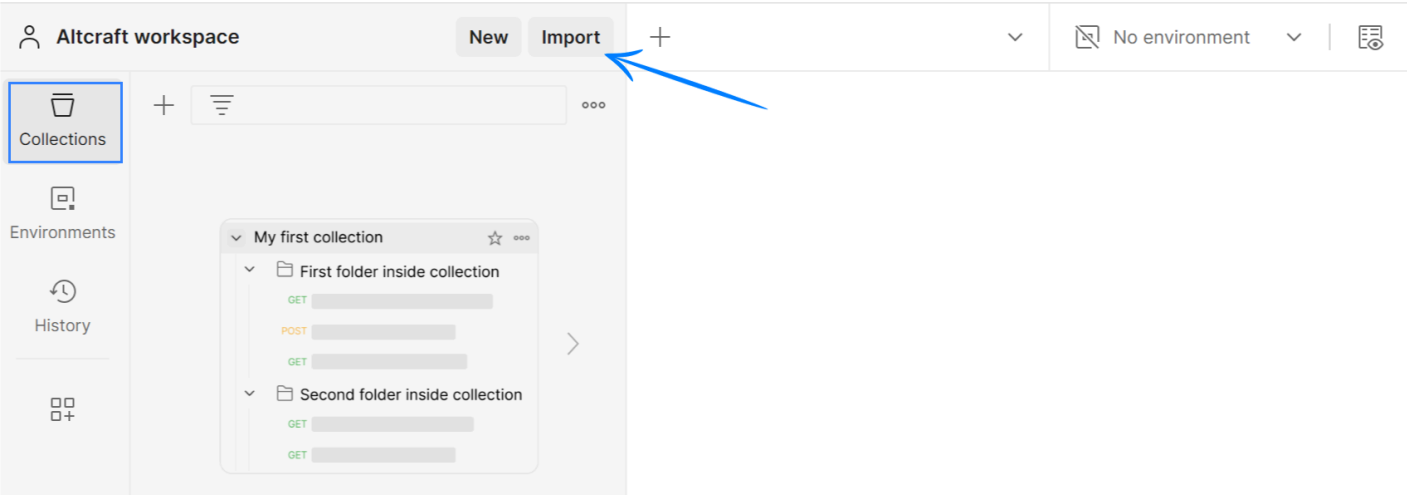
Импорт коллекции

1. Сначала убедитесь, что у вас установлен **Postman**. Вы можете скачать его на **официальном сайте**.

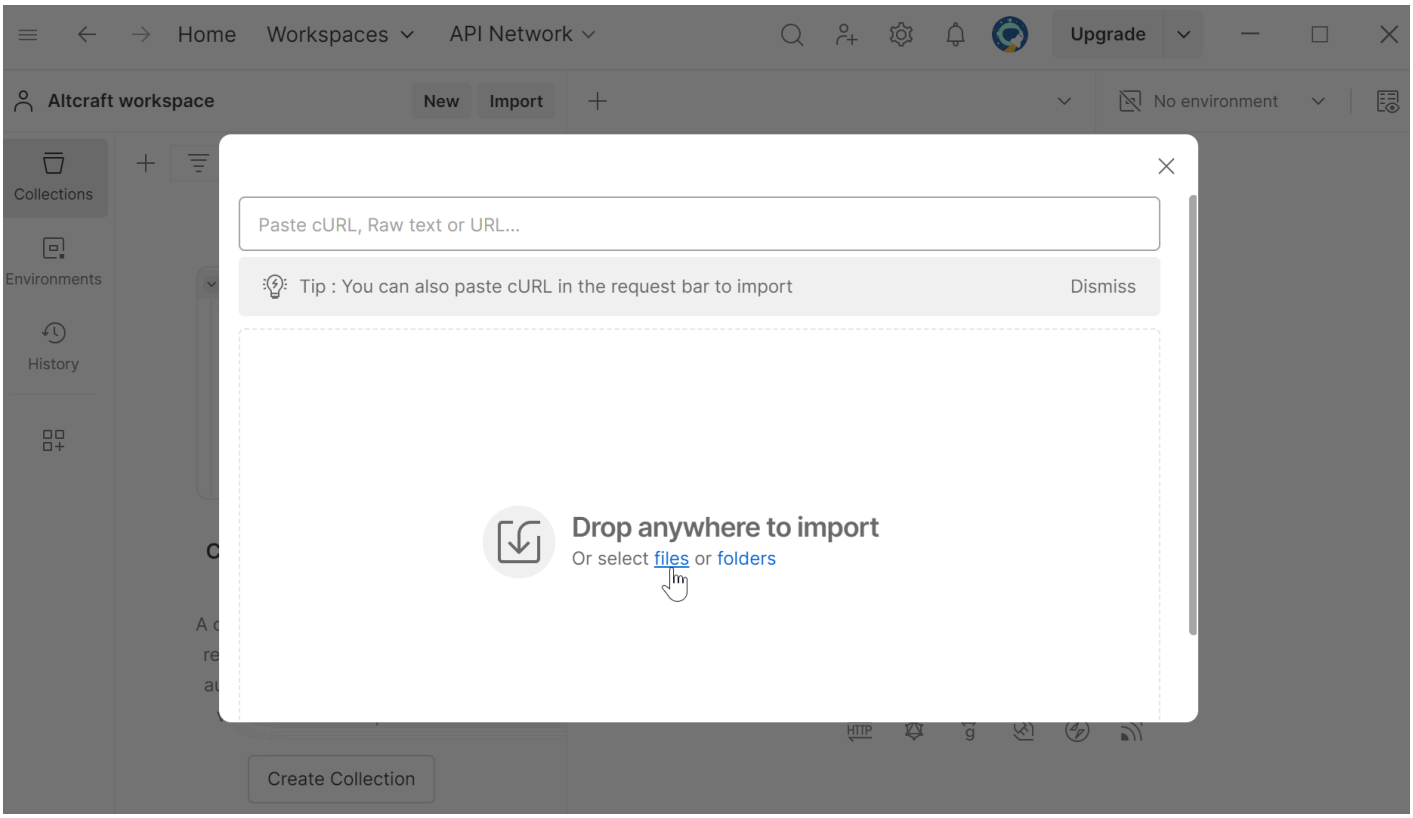
2. **Скачайте** коллекцию API-методов на ваше устройство.

> **TIP**
>
> Если загрузка файла не началась, кликните правой кнопкой мыши по ссылке и выберите **“Сохранить ссылку как”**.

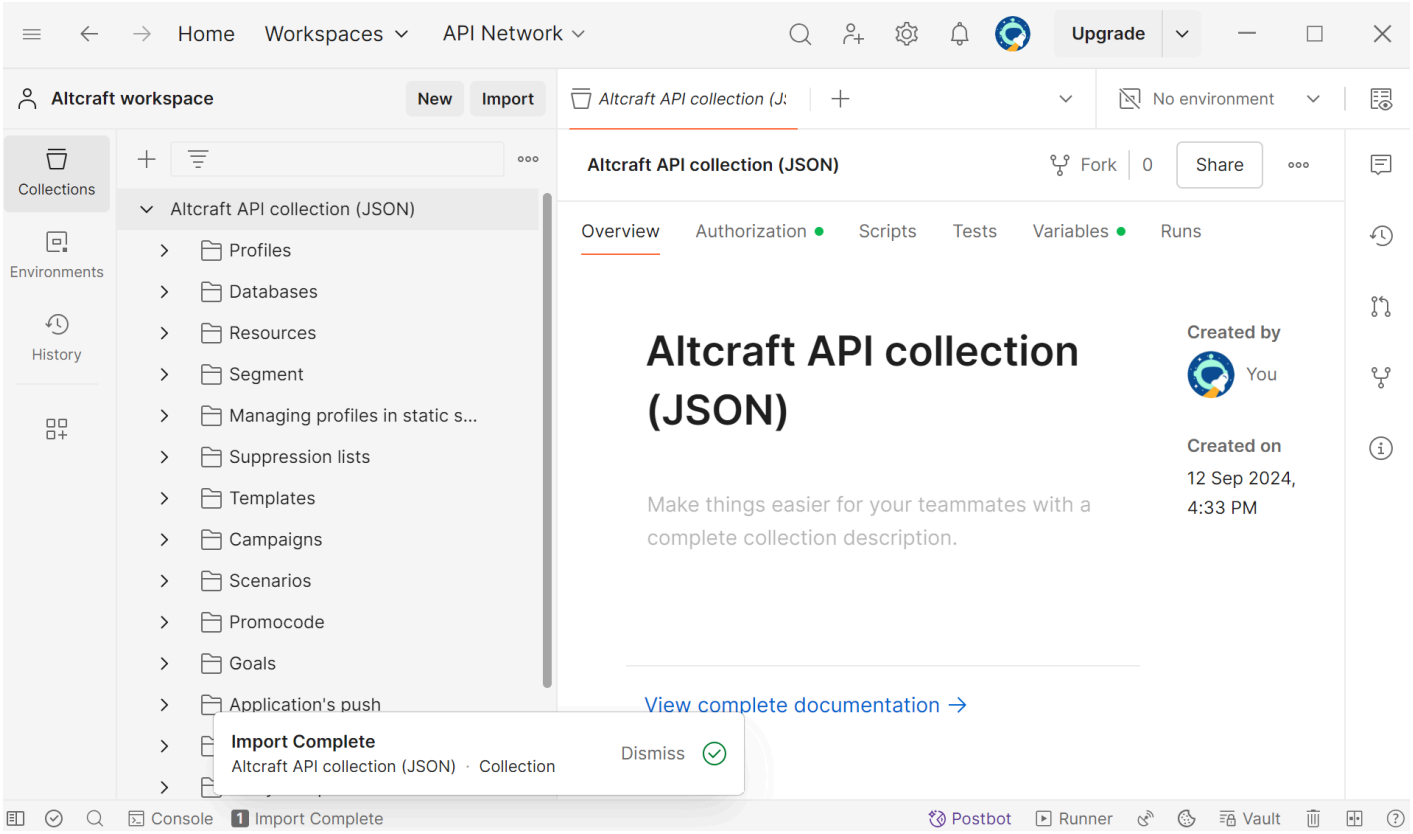
3. В Postman перейдите в раздел **“Collections”** и нажмите кнопку **“Import”**:



4. Импортируйте файл с коллекцией через окно загрузки:

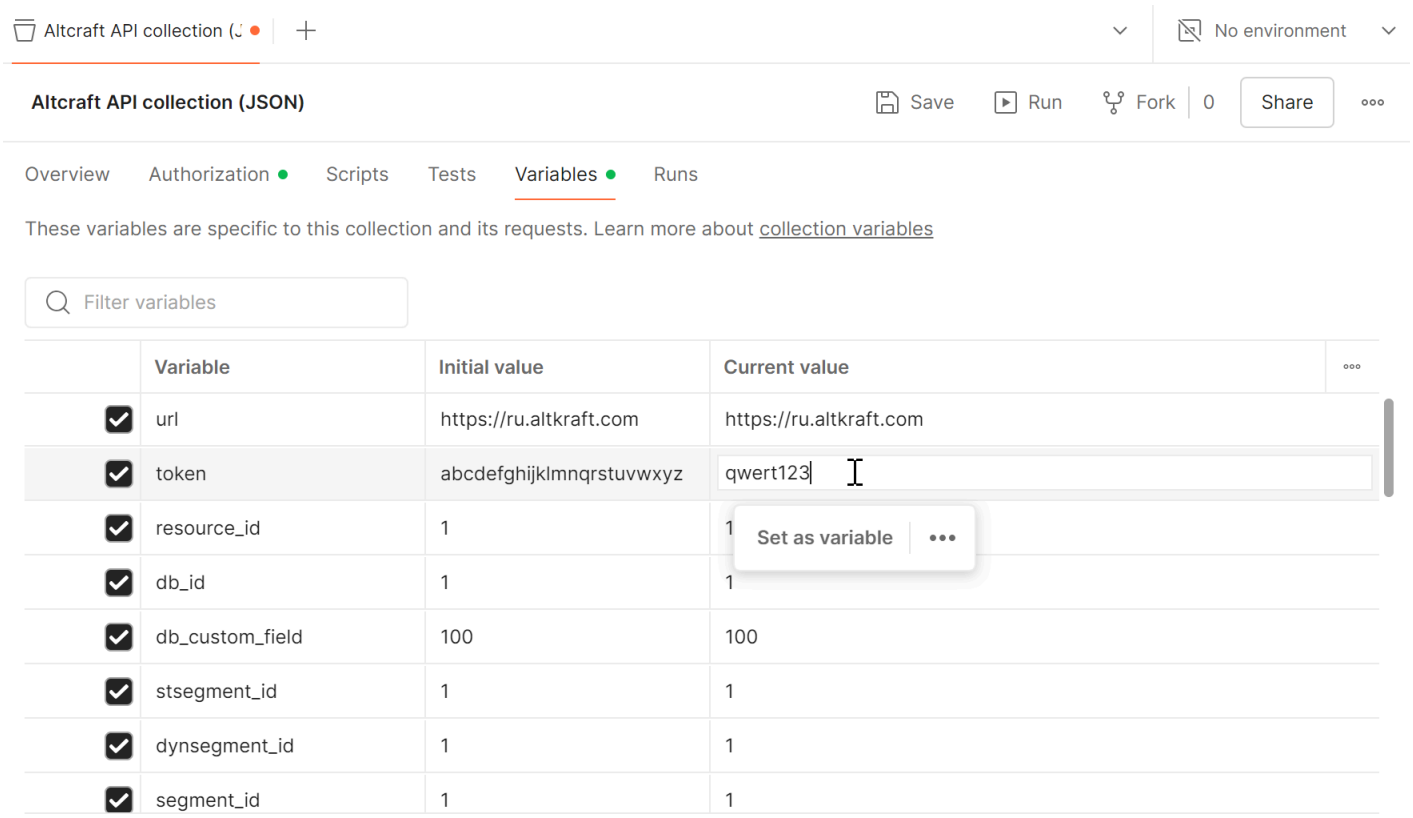


После завершения импорта все API-методы появятся в боковой панели:



5. Коллекция содержит переменные среды Postman. Перейдите в раздел **“Variables”**, чтобы просмотреть список переменных.

6. В колонке **“Initial value”** записаны исходные значения для каждой переменной. Они выступают в качестве примера. Замените эти значения на актуальные для вашего аккаунта в колонке **“Current value”**.



7. Сохраните изменения с помощью кнопки **“Save”** или Ctrl + S.

> **CAUTION**
>
> Обязательно сохраняйте изменения в списке переменных после их редактирования.

Список переменных коллекции

Переменная Postman	Описание
url	URL-адрес панели управления платформы Если вы работаете в облачном аккаунте Altcraft Platform, используйте `ru.altkraft.com`. При установке платформы на ваше оборудование (on-Premise) — используйте ваш собственный URL.
token	Токен платформы (создается в разделе "Настройки" => "Токены")
Основные объекты платформы	
resource_id	Идентификатор ресурса
db_id	Идентификатор базы профилей
db_custom_field	Имя дополнительного поля в базе профилей Обратите внимание, во всех запросах коллекции используется дополнительное поле с типом "Целое число". Если вы работаете с другими типами полей, убедитесь, что подставляете значение в корректном формате.

Переменная Postman	Описание
stsegment_id	Идентификатор статического сегмента
dynsegment_id	Идентификатор динамического сегмента
segment_id	Идентификатор статического или динамического сегмента
suplist_id	Идентификатор стоп-списка
template_id	Идентификатор шаблона сообщения
campaign_id	Идентификатор рассылки
workflow_id	Идентификатор сценария
pixel_id	Идентификатор пикселя
loyalty_id	Идентификатор программы лояльности
vsender_id	Идентификатор виртуального сендера
Системные параметры	
profile_id	Идентификатор профиля
smid	Идентификатор отправки сообщения (можно найти в заголовке письма `X-AKMTA-FSMID`)
subscription_id	Идентификатор push-подписки профиля (push-токен)
rtoken	Токен ресурса (можно найти в окне редактирования ресурса на вкладке "Детали" — кнопка i в правом верхнем углу)
launch_id	Внутренний идентификатор отправки, уникальный в рамках рассылки
Маркет	
product_id	Дополнительный идентификатор продукта
sku_id	Дополнительный идентификатор SKU
order_external_id	Дополнительный идентификатор заказа (строка)
order_internal_id	Идентификатор заказа (число)
line_id	Дополнительный идентификатор позиции заказа
market_endpoint	Дополнительный идентификатор точки контакта

Переменная Postman	Описание
market_region	Дополнительный идентификатор региона
market_category	Дополнительный идентификатор категории продукта или SKU
market_currency	Идентификатор валюты

Импортировать профиль

Описание

Запрос добавляет профиль клиента в базу. Также при помощи этого запроса можно обновить профиль, который уже есть в базе.

TIP

Чтобы одновременно с импортом отправлять триггерную рассылку, используйте этот запрос.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/import`

Параметры запроса

CAUTION

- При использовании JSON в теле запроса используйте заголовок **Content-Type: application/json**.
- При использовании XML в теле запроса используйте заголовок **Content-Type: application/xml**.

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных

Параметр	Тип	Пример	Обязательный	Описание
matching	string	"email" – поиск по email из профиля или подписок "email_profile" – по email из профиля "email_sub" – по email из подписок "phone" – по телефону из профиля или подписок "phone_sub" – по телефону из подписок "profile_id" – по идентификатору профиля (только при обновлении!) "push_sub" – по пуш подписке "custom" – по кастомному полю "custom_sub" – по подписке на кастомный канал "email_phone" – по email или телефону из профиля "email_phone_sub" – по email или телефону из подписок	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию – email Для каждого типа матчинга в теле запроса необходимо передавать определенные поля. **Подробнее об этих полях можно узнать здесь.** Обратите внимание, что при режиме "email" или "phone" поиск происходит одновременно в контактных данных и в подписках.
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false)
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки (по умолчанию – false)
detect_geo	bool	true	Нет	Включает автоопределение geo-данных по полю `_regip` или `_ip` в data

Параметр	Тип	Пример	Обязательный	Описание
add_to_segments	JSON список	`[1,2]`	Нет	Добавление профиля в сегмент
remove_from_segments	JSON список	`[1,2]`	Нет	Удаление профиля из сегмента
data	JSON object	{ "_fname": "John", "_lname": "Doe" }	Да	Данные о профиле, в т. ч. данные о подписках на ресурсы (`subscriptions`)

Массив subscriptions

Массив `subscriptions` хранит данные о подписках профиля на ресурсы. Один объект — одна подписка.

Параметры	Тип	Пример	Описание
resource_id	int	1	Идентификатор ресурса
status	string	"subscribed"	Статус подписки (опциональное поле). Рекомендуется передавать только в тех случаях, когда необходимо обновить статус подписки.
priority	int	1	Приоритет подписки (опциональное поле). В случае если приоритет подписки передается в запросе, опция "Повышать приоритет для новых подписок" в [ресурсе](/user-guide/profiles-and-databases) не работает.
custom_fields	object	{ "_browser_name": "Chrome", "_device_type": "web" }	Стандартные и дополнительные поля подписки (опциональное поле). Получить информацию о доступных полях подписки в ресурсе можно с помощью метода [/v1.1/resources/fields_get](/developer-guide/resources/2195591) </td>

cats	array	["category_1", "category_2"]	Категории ресурса, на которые подписать профиль (опциональное поле).
Email-канал			
channel	string	"email"	Тип канала
email	string	"john@example.com"	Email-адрес
SMS-канал			
channel	string	"sms"	Тип канала
phone	string	"+7900000000 00"	Номер телефона
Push-канал			
channel	string	"push"	Тип канала
provider	string	"Firefox"	Тип провайдера
subscription_id	string	"abcdefghijklmnqrstuvwxyz"	Идентификатор подписки
Telegram Bot канал			
channel	string	"telegram_bot"	Тип канала

cc_data	object	{
		"id": 1
		}
			ID чата в Telegram-бот

WhatsApp*-канал			
channel	string	"whatsapp"	Тип канала

cc_data	object	{
		"phone": "+7900000000 00"
		}
			Телефон профиля в WhatsApp*

Viber-канал			

channel	string	"viber"	Тип канала
cc_data	object	{ "phone": "+79000000000" }	Телефон профиля в Viber
Notify-канал			
channel	string	"notify"	Тип канала
cc_data	object	{ "phone": "+79000000000" }	Телефон профиля в VK/OK/SMS

Примеры запросов

Импорт профиля с добавлением подписки на email-канал

JSON

```
{
     "token": "abcdefghijklmnqrstuvwxyz",
     "db_id": 1,
     "matching": "email",
     "email": "example@example.com",
     "detect_geo": true,
     "data": {
        "_fname": "Olly",
        "_lname": "Lambert",
        "email": "example@example.com",
        "phones": ["+79000000000"],
        "subscriptions": [
            {
                "resource_id": 1,
                "channel": "email",
                "email": "example@example.com"
            },
            {
                "resource_id": 1,
                "channel": "sms",
                "phone": "+79000000000"
            }
        ]
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <email>example@example.com</email>
    <matching>email</matching>
    <detect_geo>true</detect_geo>
    <data>
        <_fname>Olly</_fname>
        <_lname>Lambert</_lname>
        <email>example@example.com</email>
        <phones>+79511059000</phones>
        <subscriptions array='true'>
            <resource_id>1</resource_id>
            <channel>email</channel>
            <email>example@example.com</email>
        </subscriptions>
    </data>
</xml>
```

Импорт профиля с передачей его дополнительных данных

JSON

```
{
     "token": "abcdefghijklmnqrstuvwxyz",
     "db_id": 1,
     "matching": "email",
     "email": "example@example.com",
     "detect_geo": true,
     "data": {
        "_fname": "Olly",
        "_lname": "Lambert",

        "email": "example@example.com",
        "phones": ["+79000000000"],

        "_bdate": "1990-02-22T21:00:00Z",
        "_sex": 0,

        "_regdate": "2019-03-14T22:00:00Z",
        "_regip": "94.231.119.122",
        "_ip": "94.231.119.122",

        "_tz": "Europe/Moscow",
        "_postal_code": "390000",

        "_os": "Windows 10",
        "_browser": "Firefox",

        "_vendor": "form_#31",

        "custom_field": "custom_value",

        "subscriptions": [
            {
                "resource_id": 1,
                "channel": "email",
                "email": "example@example.com"
            },
            {
                "resource_id": 1,
                "channel": "sms",
                "phone": "+79000000000"
            }
        ]
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <detect_geo>true</detect_geo>
    <email>example@example.com</email>
    <matching>email</matching>
    <data>
        <_fname>Olly</_fname>
        <_lname>Lambert</_lname>
        <email>example@example.com</email>
        <phones>+79511059000</phones>
        <_bdate>1990-07-06</_bdate>
        <_sex>0</_sex>
        <_regdate>2023-07-06</_regdate>
        <_regip>192.168.0.179</_regip>
        <_ip>192.168.0.179</_ip>
        <_tz>Europe/Moscow</_tz>
        <_postal_code>390000</_postal_code>
        <_os>Windows 10</_os>
        <_browser>Firefox</_browser>
        <_vendor>form_#31</_vendor>
        <custom_field>custom_value</custom_field>
        <subscriptions array='true'>
            <resource_id>1</resource_id>
            <channel>email</channel>
            <email>example@example.com</email>
        </subscriptions>
        <subscription>
            <resource_id>1</resource_id>
            <channel>sms</channel>
            <phone>+79000000000</phone>
        </subscription>
      </subscription>
    </data>
</xml>
```

Импорт профиля с добавлением подписок на различные каналы

JSON

```
{
     "token": "abcdefghijklmnqrstuvwxyz",
     "db_id": 1,
     "matching": "email",
     "email": "example@example.com",
     "detect_geo": true,
     "data": {
        "_fname": "Olly",
        "_lname": "Lambert",
        "email": "example@example.com",
        "subscriptions": [
            {
                "resource_id": 1,
                "channel": "email",
                "email": "example@example.com"
            },
            {
                "resource_id": 2,
                "channel": "sms",
                "phone": "+79000000000"
            },
            {
                "resource_id": 1,
                "channel": "push",
                "subscription_id": "a81c264a938b475",
                "provider": "android-firebase",
                "custom_fields": {
                    "_browser_name": "Firefox",
                    "_device_type": "web",
                    "custom_field_1": "test value"
                },
                "cats": [
                    "category_1",
                    "category_2"
                ]
            },
            {
                "resource_id": 1,
                "channel": "whatsapp",
                "cc_data": {
                    "phone": "+79000000000"
                }
            },
            {
                "resource_id": 1,
                "channel": "viber",
                "cc_data": {
                    "phone": "+79000000000"
                }
```

```
                }
            },
            {
                "resource_id": 1,
                "channel": "telegram_bot",
                "cc_data": {
                    "id": 12345
                }
            },
            {
                "resource_id": 1,
                "channel": "notify",
                "cc_data": {
                    "phone": "+79000000000"
                }
            }
        ]
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <detect_geo>true</detect_geo>
    <email>example@example.com</email>
    <matching>email</matching>
    <data>
        <_fname>Olly</_fname>
        <_lname>Lambert</_lname>
        <email>example@example.com</email>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>email</channel>
            <email>example@example.com</email>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>sms</channel>
            <phone>+79000000000</phone>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>push</channel>
            <subscription_id>a81c264a938b475</subscription_id>
            <provider>android-firebase</provider>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>whatsapp</channel>
            <cc_data>
              <phone>+79000000000</phone>
            </cc_data>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>viber</channel>
            <cc_data>
              <phone>+79000000000</phone>
            </cc_data>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>telegram_bot</channel>
            <cc_data>
              <id>12345</id>
            </cc_data>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
```

```
            <resource_id>1</resource_id>
            <channel>notify</channel>
            <cc_data>
              <phone>+79000000000</phone>
            </cc_data>
        </subscriptions>
    </data>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "profile_id": "54759eb3c090d83494e2d804"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <profile_id>c1are24fu213lwith21that2axe9<profile_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_id	string	Идентификатор профиля

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Обновить профиль

Описание

Запрос обновляет данные профиля, которые уже есть в базе (повторяет структуру и параметры запроса /profiles/import).

URL-адрес

Метод: POST

```
https://example.com/api/v1.1/profiles/update
```

Параметры запроса

CAUTION

- При использовании JSON в теле запроса используйте заголовок **Content-Type: application/json**.
- При использовании XML в теле запроса используйте заголовок **Content-Type: application/xml**.

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных

Параметр	Тип	Пример	Обязательный	Описание
matching	string	“email” – поиск по email из профиля или подписок “email_profile” – по email из профиля “email_sub” – по email из подписок “phone” – по телефону из профиля или подписок “phone_sub” – по телефону из подписок “profile_id” – по идентификатору профиля (только при обновлении!) “push_sub” – по пуш подписке “custom” – по кастомному полю “custom_sub” – по подписке на кастомный канал “email_phone” – по email или телефону из профиля “email_phone_sub” – по email или телефону из подписок	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. по умолчанию – email Для каждого типа матчинга в теле запроса необходимо передавать определенные поля. **Подробнее об этих полях можно узнать здесь.** Обратите внимание, что при режиме “email” или “phone” поиск происходит одновременно в контактных данных и в подписках.
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false)
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки (по умолчанию – false)
detect_geo	bool	true	Нет	Включает автоопределение geo данных по полю _regip или _ip в data
add_to_segments	JSON список	`[1,2]`	Нет	Добавление профиля в сегмент

Параметр	Тип	Пример	Обязательный	Описание
remove_from_segments	JSON список	`[1,2]`	Нет	Удаление профиля из сегмента

data	JSON object	{
		“_fname”: “John”,
		“_lname”: “Doe”
		}
			Да	Данные о профиле,
				в т. ч. данные о подписках на ресурсы
				(`subscriptions`)

Массив subscriptions

Массив `subscriptions` хранит данные о подписках профиля на ресурсы. Один объект — одна подписка.

Параметры	Тип	Пример	Описание
resource_id	int	1	Идентификатор ресурса
status	string	"subscribed"	Статус подписки (опциональное поле). Рекомендуется передавать только в тех случаях, когда необходимо обновить статус подписки.
priority	int	1	Приоритет подписки (опциональное поле). В случае если приоритет подписки передается в запросе, опция "Повышать приоритет для новых подписок" в [ресурсе](/user-guide/profiles-and-databases) не работает.

custom_fields	object	{
		"_browser_name": "Chrome",
		
		"_device_type": "web"
		}
			Стандартные и дополнительные поля подписки (опциональное поле).
			Получить информацию о доступных полях подписки в ресурсе можно с помощью метода [/v1.1/resources/fields_get](/developer-guide/resources/2195591)
			</td>

cats	array	["category_1", "category_2"]	Категории ресурса, на которые подписать профиль (опциональное поле).
Email-канал			

channel	string	"email"	Тип канала
email	string	"john@example.com"	Email-адрес
SMS-канал			
channel	string	"sms"	Тип канала
phone	string	"+7900000000 00"	Номер телефона
Push-канал			
channel	string	"push"	Тип канала
provider	string	"Firefox"	Тип провайдера
subscription_id	string	"abcdefghijkl mnqrstuvwxyz "	Идентификатор подписки
Telegram Bot канал			
channel	string	"telegram_bot "	Тип канала
cc_data	object	{ "id": 1 }	ID чата в Telegram-бот
WhatsApp*-канал			
channel	string	"whatsapp"	Тип канала
cc_data	object	{ "phone": "+790000000 00" }	Телефон профиля в WhatsApp*
Viber-канал			
channel	string	"viber"	Тип канала
cc_data	object	{ "phone": "+790000000 00" }	Телефон профиля в Viber
Notify-канал			

channel	string	"notify"	Тип канала
cc_data	object	{ "phone": "+79000000000" }	Телефон профиля в VK/OK/SMS

Примеры запросов

Обновление основных данных профиля и добавление подписки на email-канал

JSON

```
{
     "token": "abcdefghijklmnqrstuvwxyz",
     "db_id": 1,
     "matching": "email",
     "email": "example@example.com",
     "detect_geo": true,
     "data": {
        "_fname": "Olly",
        "_lname": "Lambert",
        "email": "example@example.com",
        "phones": ["+79000000000"],
        "subscriptions": [
            {
                "resource_id": 1,
                "channel": "email",
                "email": "example@example.com"
            }
        ]
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <email>example@example.com</email>
    <matching>email</matching>
    <detect_geo>true</detect_geo>
    <data>
        <_fname>Olly</_fname>
        <_lname>Lambert</_lname>
        <email>example@example.com</email>
        <phones>+79511059000</phones>
        <subscriptions array='true'>
            <resource_id>1</resource_id>
            <channel>email</channel>
            <email>example@example.com</email>
        </subscriptions>
    </data>
</xml>
```

Обновление дополнительных данных профиля

JSON

```
{
     "token": "abcdefghijklmnqrstuvwxyz",
     "db_id": 1,
     "matching": "email",
     "email": "example@example.com",
     "detect_geo": true,
     "data": {
        "_fname": "Olly",
        "_lname": "Lambert",

        "email": "example@example.com",
        "phones": ["+79000000000"],

        "_bdate": "1990-02-22T21:00:00Z",
        "_sex": 0,

        "_regdate": "2019-03-14T22:00:00Z",
        "_regip": "94.231.119.122",
        "_ip": "94.231.119.122",

        "_tz": "Europe/Moscow",
        "_postal_code": "390000",

        "_os": "Windows 10",
        "_browser": "Firefox",

        "_vendor": "form_#31",

        "custom_field": "custom_value"
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <detect_geo>true</detect_geo>
    <email>example@example.com</email>
    <matching>email</matching>
    <data>
        <_fname>Olly</_fname>
        <_lname>Lambert</_lname>
        <email>example@example.com</email>
        <phones>+79511059000</phones>
        <_bdate>1990-07-06</_bdate>
        <_sex>0</_sex>
        <_regdate>2023-07-06</_regdate>
        <_regip>192.168.0.179</_regip>
        <_ip>192.168.0.179</_ip>
        <_tz>Europe/Moscow</_tz>
        <_postal_code>390000</_postal_code>
        <_os>Windows 10</_os>
        <_browser>Firefox</_browser>
        <_vendor>form_#31</_vendor>
        <custom_field>custom_value</custom_field>
    </data>
</xml>
```

Добавление профилю подписок на различные каналы

JSON

```
{
     "token": "abcdefghijklmnqrstuvwxyz",
     "db_id": 1,
     "matching": "email",
     "email": "example@example.com",
     "detect_geo": true,
     "data": {
        "_fname": "Olly",
        "_lname": "Lambert",
        "email": "example@example.com",
        "subscriptions": [
            {
                "resource_id": 1,
                "channel": "email",
                "email": "example@example.com"
            },
            {
                "resource_id": 2,
                "channel": "sms",
                "phone": "+79000000000"
            },
            {
                "resource_id": 1,
                "channel": "push",
                "subscription_id": "a81c264a938b475",
                "provider": "android-firebase",
                "custom_fields": {
                    "_browser_name": "Firefox",
                    "_device_type": "web",
                    "custom_field_1": "test value"
                },
                "cats": [
                    "category_1",
                    "category_2"
                ]
            },
            {
                "resource_id": 1,
                "channel": "whatsapp",
                "cc_data": {
                    "phone": "+79000000000"
                }
            },
            {
                "resource_id": 1,
                "channel": "viber",
                "cc_data": {
                    "phone": "+79000000000"
                }
```

```
                }
            },
            {
                "resource_id": 1,
                "channel": "telegram_bot",
                "cc_data": {
                    "id": 12345
                }
            },
            {
                "resource_id": 1,
                "channel": "notify",
                "cc_data": {
                    "phone": "+79000000000"
                }
            }
        ]
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <detect_geo>true</detect_geo>
    <email>example@example.com</email>
    <matching>email</matching>
    <data>
        <_fname>Olly</_fname>
        <_lname>Lambert</_lname>
        <email>example@example.com</email>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>email</channel>
            <email>example@example.com</email>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>sms</channel>
            <phone>+79000000000</phone>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>push</channel>
            <subscription_id>a81c264a938b475</subscription_id>
            <provider>android-firebase</provider>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>whatsapp</channel>
            <cc_data>
              <phone>+79000000000</phone>
            </cc_data>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>viber</channel>
            <cc_data>
              <phone>+79000000000</phone>
            </cc_data>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
            <channel>telegram_bot</channel>
            <cc_data>
              <id>12345</id>
            </cc_data>
        </subscriptions>
        <subscriptions>
            <resource_id>1</resource_id>
```

```
            <resource_id>1</resource_id>
            <channel>notify</channel>
            <cc_data>
              <phone>+79000000000</phone>
            </cc_data>
        </subscriptions>
    </data>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "profile_id": "54759eb3c090d83494e2d804"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
        <profile_id>c1are24fu213lwith21that2axe9<profile_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_id	string	Идентификатор профиля

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Добавить профиль в базу данных

TIP

Запрос добавления профиля не ищет существующие записи и добавляет любые данные, кроме тех, которые дублируют уникальное поле профиля (email). **Рекомендуем использовать Импорт профилей в большинстве ситуаций.**

Описание

Добавить новый профиль в базу данных.

CAUTION

Запрос принимает любые данные. Если не задать структуру, то будет создан пустой профиль.

URL-адрес

`https://example.com/api/v1.1/profiles/add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных
data	JSON object	`{ “_fname”: “John”, “_lname”: “Doe” }`	Да	Данные о профиле
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false)
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки (по умолчанию – false)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "data": {
        "_fname": "Fillipp",
        "_lname": "Doe"
    },
    "db_id": 1,
    "skip_triggers": true
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <skip_triggers>true</skip_triggers>
    <data>
        <_fname>Fillipp</_fname>
        <_lname>Doe</_lname>
    </data>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "",
  "profile_id": "6638e1e8ee44bb50a67115c4"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <profile_id>6638e1e8ee44bb50a67115c4</profile_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить информацию о профиле

Описание

Получить информацию о профиле.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/get/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных
Поиск профиля подписчика				
[matching](/developer-guide/37848489/)	string	"email" "email_profile" "email_sub" "phone" "phone_sub" "profile_id" "push_sub" "custom" "custom_sub" "email_phone" "email_phone_sub"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию - email.
email	string	"example@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес

phone	string	"+7900000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. **TIP** Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "email": "example@example.com"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <email>example@example.com</email>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "LOYALTY_SCORE": 500,
        "New_field_7": "00016",
        "_acid": "FBFLR5Z8BxgvNPxBJZgL",
        "_bdate": "1990-07-06T11:15:25.526Z",
        "_browser": "Chrome",
        "_city": "Beijing",
        "_click_exist": false,
        "_click_push_exist": false,
        "_click_sms_exist": false,
        "_confirm_exist": false,
        "_country": "CN",
        "_createtime": "2024-04-08T10:48:05.418Z",
        "_deliv_date": "2024-04-18T10:18:29.809Z",
        "_deliv_exist": true,
        "_deliv_push_exist": false,
        "_deliv_sms_date": "2024-05-06T10:00:05.548Z",
        "_deliv_sms_exist": true,
        "_domaingroup": "other",
        "_emaildomain": "example.com",
        "_fname": "Christopher",
        "_ip": "106.12.21.70",
        "_is_desktop": true,
        "_is_test": true,
        "_lat": 34.7732,
        "_lname": "Lambert",
        "_lon": 113.722,
        "_md5": "1118e73b34d3f9a9030b9e4da6dcfb29",
        "_open_date": "2024-04-18T10:18:29.727Z",
        "_open_exist": true,
        "_open_push_exist": false,
        "_os": "Windows 10",
        "_pixel_open_exist": false,
        "_platform": "Other",
        "_postal_code": "17",
        "_ps_wf": [
            12,
            11,
            10
        ],
        "_ps_wfs": [
            {
                "di": "2024-04-08T10:48:07.093Z",
                "do": "2024-04-13T10:48:22.359Z",
                "et": "",
                "ln": 8,
                "s": 1,
                "wId": 12
```

```
            "wId": 12
        },
        {
            "di": "2024-04-08T19:35:04.15Z",
            "do": "2024-04-12T19:37:19.821Z",
            "et": "",
            "ln": 8,
            "s": 1,
            "wId": 11
        },
        {
            "di": "2024-04-16T14:42:49.063Z",
            "do": "2024-04-17T14:42:52.438Z",
            "et": "",
            "ln": 8,
            "s": 1,
            "wId": 10
        }
    ],
    "_read_date": "2024-04-18T10:18:29.727Z",
    "_read_exist": true,
    "_regdate": "2024-04-08T10:48:05.418Z",
    "_region": "Beijing",
    "_reply_exist": false,
    "_send_date": "2024-04-10T09:32:06.275960407Z",
    "_send_exist": true,
    "_send_push_exist": false,
    "_send_sms_date": "2024-04-12T12:17:05.995339508Z",
    "_send_sms_exist": true,
    "_sex": 1,
    "_status": 0,
    "_subscribetime": "2024-04-08T10:48:05.418Z",
    "_tz": "Asia/Shanghai",
    "_undeliv_date": "2024-04-17T11:49:42.098Z",
    "_undeliv_exist": true,
    "_updatetime": "2024-04-19T14:25:22.185Z",
    "_xxh": "5ce324d8407a8e2b",
    "_xxm": 8507,
    "db_id": 1,
    "email": "example@example.com",
    "id": "6613cb65dc4594c2934db63f",
    "phones": [
        "+79568150396"
    ],
    "subscriptions": [
        {
            "resource_id": 24,
            "channel": "email",
            "email_data": {
                "email": "example@example.com",
```

```
                    "domain": "example.com",
                    "domain_group": "other",
                    "md5": "1118e73b34d3f9a9030b9e4da6dcfb29"
                },
                "priority": 0,
                "status": "unsubscribed",
                "reg_info": {
                    "date": "2024-04-10T09:28:55.743Z"
                },
                "hash_id": "4a3c5813",
                "is_delete": false
            },
            {
                "resource_id": 25,
                "channel": "sms",
                "phone_data": {
                    "phone": "+79568150396",
                    "md5": "06ec7f059dae6f002d06a2f62a4320d8"
                },
                "priority": 0,
                "status": "subscribed",
                "reg_info": {
                    "date": "2024-04-12T12:16:44.086Z"
                },
                "hash_id": "f67a97c1",
                "is_delete": false
            }
        ],
        "unsub": {
            "c": 115,
            "dt": "2024-04-18T10:18:29.727Z",
            "m": 15
        }
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
    <data>
        <CRM_ID>66</CRM_ID>
        <LOYALTY_SCORE>500</LOYALTY_SCORE>
        <_acid>HWbGhUN1jKAdY8iNjdYf</_acid>
        <_browser>Chrome</_browser>
        <_city>St Petersburg</_city>
        <_click_exist>false</_click_exist>
        <_click_push_exist>false</_click_push_exist>
        <_click_sms_exist>false</_click_sms_exist>
        <_confirm_exist>false</_confirm_exist>
        <_country>RU</_country>
        <_createtime>2024-04-08T10:24:09.087Z</_createtime>
        <_deliv_date>2024-05-06T10:00:06.623Z</_deliv_date>
        <_deliv_exist>true</_deliv_exist>
        <_deliv_push_exist>false</_deliv_push_exist>
        <_deliv_sms_date>2024-05-06T10:00:05.274Z</_deliv_sms_date>
        <_deliv_sms_exist>true</_deliv_sms_exist>
        <_domaingroup>other</_domaingroup>
        <_emaildomain>example.com</_emaildomain>
        <_fname>Alina</_fname>
        <_ip>92.53.90.8</_ip>
        <_is_desktop>true</_is_desktop>
        <_is_mobile>true</_is_mobile>
        <_is_test>true</_is_test>
        <_lat>59.939095</_lat>
        <_lname>Lambert</_lname>
        <_lon>30.315868</_lon>
        <_md5>c5278382ce8972b4cd23b447c0e58989</_md5>
        <_open_date>2024-05-06T10:00:06.481Z</_open_date>
        <_open_exist>true</_open_exist>
        <_open_push_exist>false</_open_push_exist>
        <_os>Windows 10</_os>
        <_pixel_open_exist>false</_pixel_open_exist>
        <_platform>Other</_platform>
        <_postal_code>1042</_postal_code>
        <_ps_wf>12</_ps_wf>
        <_ps_wf>11</_ps_wf>
        <_ps_wfs>
            <di>2024-04-08T10:24:10.069Z</di>
            <do>2024-04-19T10:24:43.381Z</do>
            <ln>8</ln>
            <s>1</s>
            <wId>12</wId>
        </_ps_wfs>
        <_ps_wfs>
            <di>2024-04-08T19:35:04.091Z</di>
            <do>2024-04-18T19:37:37.428Z</do>
            <ln>8</ln>
```

```
            <in>8</in>
            <s>1</s>
            <wId>11</wId>
        </_ps_wfs>
        <_read_date>2024-05-06T10:00:06.481Z</_read_date>
        <_read_exist>true</_read_exist>
        <_regdate>2024-04-08T10:24:09.087Z</_regdate>
        <_region>Saint Petersburg</_region>
        <_reply_exist>false</_reply_exist>
        <_sbounce_date>2024-04-28T10:00:03.452Z</_sbounce_date>
        <_sbounce_exist>true</_sbounce_exist>
        <_send_date>2024-04-10T09:32:05.531223661Z</_send_date>
        <_send_exist>true</_send_exist>
        <_send_push_exist>false</_send_push_exist>
        <_send_sms_date>2024-04-18T11:14:02.172438444Z</_send_sms_date>
        <_send_sms_exist>true</_send_sms_exist>
        <_status>0</_status>
        <_subscribetime>2024-04-08T10:24:09.087Z</_subscribetime>
        <_tz>Europe/Moscow</_tz>
        <_undeliv_date>2024-04-17T11:49:39.091Z</_undeliv_date>
        <_undeliv_exist>true</_undeliv_exist>
        <_undeliv_sms_date>2024-04-22T10:00:02.62Z</_undeliv_sms_date>
        <_undeliv_sms_exist>true</_undeliv_sms_exist>
        <_updatetime>2024-04-18T12:36:31.7Z</_updatetime>
        <_xxh>507a01069df7046e</_xxh>
        <_xxm>7902</_xxm>
        <db_id>1</db_id>
        <email>example@example.com</email>
        <id>6613c5c94a5d12bd6cc09171</id>
        <phones>+79006574081</phones>
        <subscriptions>
            <channel>email</channel>
            <email_data>
                <domain>example.com</domain>
                <domain_group>other</domain_group>
                <email>example@example.com</email>
                <md5>c5278382ce8972b4cd23b447c0e58989</md5>
            </email_data>
            <hash_id>fcc69fe5</hash_id>
            <is_delete>false</is_delete>
            <priority>0</priority>
            <reg_info>
                <date>2024-04-10T09:28:55.706Z</date>
            </reg_info>
            <resource_id>24</resource_id>
            <status>subscribed</status>
        </subscriptions>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
```

```
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	JSON object	Информация о профиле
error	int	Код ошибки
error_text	string	Текст ошибки

Импортировать профиль в RabbitMQ

Описание

Добавление или обновление профиля клиента через очередь RabbitMQ для больших объемов данных.

Очередь, куда должны попадать сообщения для импорта, – называется “*database_import*”. Она должна быть доступна в виртуальном того же брокера сообщений, что и вся система.

Результаты будут отправляться в другую очередь – “*database_import_result*”. По умолчанию устанавливается TTL сообщений на очереди - 24 часа. Можно указать свой TTL в конфигурации main.json: `RABBITMQ_X_MESSAGE_TTL`, в секундах.

Для идентификации запросов можно использовать дополнительное поле – “*feedback_id*”.
Оно появится в ответе, и в случае ошибки, будет отображено в логе – “*job_server.log*”.

Параметры сообщения

Параметр	Тип	Пример	Обязательный	Описание
account_id	int	1	Да	Идентификатор аккаунта
data	JSON object	`{ "_fname": "John", "_lname": "Doe" }`	Да	Данные о профиле
db_id	int	1	Да	Идентификатор базы данных
feedback_id	string	"abcdefghijklmnqrstuvwxyz"	Нет	Идентификатор базы данных
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false)
update_only	bool	true	Нет	Только обновление (по умолчанию – false)
Поиск профиля подписчика				

[matching](/developer-guide/37848489)	string	"email" "email_profile" "email_sub" "phone" "phone_sub" "profile_id" "push_sub" "custom" "custom_sub" "email_phone" "email_phone_sub"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию - email.
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+7900000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. **TIP** Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример сообщения

```
{
    "account_id": 1,
    "data": {
        "_fname": "John",
        "_lname": "Doe"
    },
    "db_id": 1,
    "email": "john@example.com",
    "feedback_id": "abcdefghijklmnqrstuvwxyz",
    "update_only": true
}
```

Пример ответа

```
{
    "error": 0,
    "error_text": "",
    "feedback_id": "abcdefghijklmnqrstuvwxyz",
    "matching_field": "email",
    "profile_id": "abcdefghijklmnqrstuvwxyz"
}
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
feedback_id	string	Идентификатор запроса
matching_field	string	Поле для поиска
profile_id	string	Идентификатор профиля

Массовое добавление профилей в базу данных

Описание

Массовое добавление новых профилей в базу данных.

Список профилей необходимо передать в параметр `"data"`. Ответ сервера будет содержать параметр `"result"` с результатом добавления, ошибками и идентификаторами добавленных профилей.

CAUTION

В одном запросе не рекомендуется:

- передавать более 10 000 профилей;
- передавать более 10 Мб данных. Можно передавать больший объем в несколько потоков.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/add_batch/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
data	JSON object	[{ “_fname”: “John”, “_lname”: “Doe”, “subscriptions”: [{ “channel”: “email”, “email_data”: { “email”: “john@example.com” }, “resource_id”: 1 }] }]	Да	Данные о профилях
db_id	int	1	Да	Идентификатор базы данных

Параметр	Тип	Пример	Обязательный	Описание
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false)
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки (по умолчанию – false)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "skip_triggers": true,
    "data": [
        {
            "_fname": "John",
            "_lname": "Doe",
            "subscriptions": [
                {
                    "channel": "email",
                    "email_data": {
                        "email": "example@example.com"
                    },
                    "resource_id": 27
                }
            ]
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <skip_triggers>true</skip_triggers>
    <data array='true'>
        <_fname>Fillipp</_fname>
        <_lname>Doe</_lname>
        <subscriptions array='true'>
            <channel>email</channel>
            <email_data>
                <email>example@example.com</email>
            </email_data>
            <resource_id>1</resource_id>
        </subscriptions>
    </data>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "result": [
    {
      "error": 0,
      "error_text": "Successful operation",
      "profile_id": "663a138cee44bb50a67115cd"
    }
  ]
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <profile_id>663a138cee44bb50a67115cd</profile_id>
  </xml>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_id	string	Идентификатор профиля

Массовое обновление профилей

Описание

Массовое обновление данных профилей.

NOTE

Если передать параметры объекта **subscriptions** `status` и `priority` со значением `0`, то значение учитывается.
Чтобы не изменять и не обновлять значения этих параметров – не передавайте их, либо установите значения в `-1`.

CAUTION

В одном запросе не рекомендуется:

- передавать более 10 000 профилей;
- передавать более 10 Мб данных. Можно передавать больший объем в несколько потоков.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/update_batch`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен

data	JSON array	[
		{
		"_fname": "John",
		"_lname": "Doe",
		"custom_id": "abcdefghijklmnqrstuvwxyz",
		"email": "john@example.com",
		"subscriptions": [
		{
		"channel": "email",
		"email_data": {
		"email": "john@example.com"
		},
		"resource_id": 1
		}
		]
		}
		]
			Да	Данные о профилях

db_id	int	1	Да	Идентификатор базы данных
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки (по умолчанию – false)
Поиск профиля подписчика для обновления				

matching	string	"email" – поиск по email из профиля или подписок	

```
        "email_profile" – по email из профиля

        "email_sub" – по email из подписок

        "phone" – по телефону из профиля или подписок

        "phone_sub" – по телефону из подписок

        "profile_id" – по идентификатору профиля (только
при обновлении!)

        "push_sub" – по пуш подписке

        "custom" – по кастомному полю

        "custom_sub" – по подписке на кастомный канал

        "email_phone" – по email или телефону из профиля

        "email_phone_sub" – по email или телефону из
подписок
    </td>
    <td>Нет, если поиск по email из профиля или
подписок</td>
    <td>
        Режим поиска подписчика.
        По умолчанию - email.

        Для каждого типа матчинга в теле запроса
необходимо передавать определенные поля. **Подробнее об
этих полях
        можно узнать <a href="/developer-
guide/37848489">здесь</a>.**

        Обратите внимание, что при режиме "email" или
"phone" поиск происходит одновременно в
        контактных данных и в подписках.
    </td>
</tr>
<tr>
    <td>email</td>
    <td>string</td>
    <td>"john@example.com"</td>
    <td>
        "matching":"email" - по профилю и подпискам

        "matching":"email_profile" - только по профилю
    </td>
    <td>Email-адрес. Должен быть передан в элементе
массива data.</td>
</tr>
<tr>
    <td>phone</td>
```

<td>string</td>
<td>"+79000000000"</td>
<td>"matching":"phone"</td>
<td>Номер телефона</td>
</tr>
<tr>
<td>profile_id</td>
<td>string</td>
<td>"abcdefghijklmnqrstuvwxyz"</td>
<td>"matching":"profile_id"</td>
<td>Идентификатор профиля</td>
</tr>
<tr>
<td>field_name</td>
<td>string</td>
<td>"CRM_ID"</td>
<td>"matching":"custom"</td>
<td>
Название кастомного поля профиля для поиска. Данное поле должно присутствовать в элементе
массива data.
</td>
</tr>
<tr>
<td>resource_id</td>
<td>int</td>
<td>24</td>
<td>Нет</td>
<td>Идентификатор ресурса для поиска по email и phone</td>
</tr>

Массив subscriptions

Массив `subscriptions` хранит данные о подписках профиля на ресурсы. Один объект — одна подписка.

Параметры	Тип	Пример	Описание
resource_id	int	1	Идентификатор ресурса
status	string	"subscribed"	Статус подписки. Рекомендуется передавать только в тех случаях, когда необходимо обновить статус подписки.
priority	int	1	Приоритет подписки

custom_fields	object	{
		"_browser_name": "Chrome",
		"_device_type": "web",
		}
			Стандартные и дополнительные поля подписки. Получить информацию о доступных полях подписки в ресурсе можно с помощью метода [/v1.1/resources/fields_get](/developer-guide/resources/2195591)

cats	array	["category_1", "category_2"]	Категории ресурса, на которые подписать профиль.
Email канал			
channel	string	"email"	Тип канала
email	string	"john@example.com"	Email-адрес
SMS канал			
channel	string	"sms"	Тип канала
phone	string	"+79000000000"	Номер телефона
Push канал			
channel	string	"push"	Тип канала
provider	string	"Firefox"	Тип провайдера
subscription_id	string	"abcdefghijklmnqrstuvwxyz"	Идентификатор подписки
Telegram Bot канал			
channel	string	"telegram_bot"	Тип канала

cc_data	object	{
		"id": 1
		}
			id чата в Telegram-бот

WhatsApp*-канал			

channel	string	"whatsapp"	Тип канала
cc_data	object	{ "phone": "+79000000000" }	Телефон профиля в WhatsApp*
Viber канал			
channel	string	"viber"	Тип канала
cc_data	object	{ "phone": "+79000000000" }	Телефон профиля в Viber

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "matching": "custom",
    "field_name": "Custom_ID",
    "skip_triggers": true,
    "data": [
        {
            "_fname": "Lala",
            "_lname": "Doe",
            "Custom_ID": 77,
            "email": "example@example.com",
            "subscriptions": [
                {
                    "channel": "email",
                    "email_data": {
                        "email": "example@example.com"
                    },
                    "resource_id": 28,
                    "custom_fields": {
                        "custom_field": "test value"
                    },
                    "cats": [
                        "category_1",
                        "category_2"
                    ]
                }
            ]
        },
        {
            "_fname": "Victor",
            "Custom_ID": 88,
            "email": "example@example.com"
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <matching>custom</matching>
    <field_name>Custom_ID</field_name>
    <skip_triggers>true</skip_triggers>
    <data>
        <_fname>Lala</_fname>
        <_lname>Doe</_lname>
        <Custom_ID>77</Custom_ID>
        <email>example@example.com</email>
        <subscriptions array='true'>
            <channel>email</channel>
            <email_data>
                <email>example@example.com</email>
            </email_data>
            <resource_id>28</resource_id>
            <custom_fields>
                <custom_field>test value</custom_field>
            </custom_fields>
            <cats>category_1</cats>
            <cats>category_2</cats>
        </subscriptions>
    </data>
    <data>
        <_fname>Alex</_fname>
        <Custom_ID>345</Custom_ID>
        <email>example@example.com</email>
    </data>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "result": [
        {
            "error": 0,
            "error_text": "Successful operation",
            "profile_id": "abcdefghijklmnqrstuvwxyz"
        },
        {
            "error": 0,
            "error_text": "Successful operation",
            "profile_id": "abcdefghijklmnqrstuvwzzz"
        }
    ]
}
```

XML

```
<xml>
<error>0</error>
  <error_text>Successful operation</error_text>
  <result>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
  </result>
  <result>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <profile_id>abcdefghijklmnqrstuvwzzz</profile_id>
  </result>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки, для всего запроса
error_text	string	Текст ошибки, для всего запроса
data	array of object	Результаты обновления по профилям
data.error	int	Код ошибки, для профиля
data.error_text	string	Текст ошибки, для для профиля

Параметр	Тип	Описание
data.profile_id	string	Идентификатор профиля

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Массовый импорт профилей

Описание

Массовое добавление или обновление профилей клиента.

> **NOTE**
>
> Если передать параметры объекта **subscriptions** `status` и `priority` со значением `0`, то значение учитывается.
> Чтобы не изменять и не обновлять значения этих параметров – не передавайте их, либо установите значения в `-1`.

> **CAUTION**
>
> В одном запросе не рекомендуется:
>
> - передавать более 10 000 профилей;
> - передавать более 10 Мб данных. Можно передавать больший объем в несколько потоков.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/import_batch`

Параметры запроса

> **CAUTION**
>
> - При использовании JSON в теле запроса используйте заголовок **Content-Type: application/json**.
> - При использовании XML в теле запроса используйте заголовок **Content-Type: application/xml**.

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен

data	JSON array	[
		{
		"_fname": "John",
		"_lname": "Doe",
		"subscriptions": [
		{
		"channel": "email",
		"email_data": {
		"email": "example@example.com"
		},
		"resource_id": 1
		}
		]
		}
		]
			Да	Данные о профилях

db_id	int	1	Да	Идентификатор базы данных
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки (по умолчанию – false)
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false)
detect_geo	bool	true	Нет	Заполняет геоданные по IP
Поиск профиля подписчика для обновления				

matching	string	"email" – поиск по email из профиля или подписок	

```
        "email_profile" – по email из профиля

        "email_sub" – по email из подписок

        "phone" – по телефону из профиля или подписок

        "phone_sub" – по телефону из подписок

        "profile_id" – по идентификатору профиля
(только при обновлении!)

        "push_sub" – по пуш подписке

        "custom" – по кастомному полю

        "custom_sub" – по подписке на кастомный канал

        "email_phone" – по email или телефону из
профиля

        "email_phone_sub" – по email или телефону из
подписок
    </td>
    <td>Нет, если поиск по email из профиля или
подписок</td>
    <td>
        Режим поиска подписчика.
        По умолчанию - email.

        Для каждого типа матчинга в теле запроса
необходимо передавать определенные поля. **Подробнее
об
        этих полях можно узнать <a href="/developer-
guide/37848489">здесь</a>.**

        Обратите внимание, что при режиме "email" или
"phone" поиск происходит одновременно в
        контактных данных и в подписках.
    </td>
</tr>
<tr>
    <td>email</td>
    <td>string</td>
    <td>"example@example.com"</td>
    <td>
        "matching":"email" - по профилю и подпискам

        "matching":"email_profile" - только по профилю
    </td>
    <td>Email-адрес. Должен быть передан в элементе
массива data.</td>
</tr>
<tr>
```

		<td>phone</td> <td>string</td> <td>"+79000000000"</td> <td>"matching":"phone"</td> <td>Номер телефона. Телефон должен присутствовать в элементе массива data.</td> </tr> <tr> <td>field_name</td> <td>string</td> <td>"CRM_ID"</td> <td>"matching":"custom"</td> <td> Название кастомного поля профиля для поиска. Данное поле должно присутствовать в элементе массива data. </td> </tr> <tr> <td>resource_id</td> <td>int</td> <td>24</td> <td>Нет</td> <td>Идентификатор ресурса для поиска по email и phone</td> </tr>	

Массив subscriptions

Массив `subscriptions` хранит данные о подписках профиля на ресурсы. Один объект — одна подписка.

Параметры	Тип	Пример	Описание
resource_id	int	1	Идентификатор ресурса
status	string	"subscribed"	Статус подписки. Рекомендуется передавать только в тех случаях, когда необходимо обновить статус подписки.
priority	int	1	Приоритет подписки

custom_fields	object	{
		"_browser_name": "Chrome",
		"_device_type": "web",
		}
			Стандартные и дополнительные поля подписки.
			Получить информацию о доступных полях подписки в ресурсе можно с помощью метода [/v1.1/resources/fields_get
			](/developer-guide/resources/2195591)
			</td>
			</tr>
			<tr>
			<td>cats</td>
			<td>array</td>
			<td>
			[
			"category_1",
			"category_2"
			]

```
    </td>
    <td>Категории ресурса, на которые подписать профиль.</td>
</tr>
<tr>
    <td colspan="4">
        <strong>Email канал</strong>
    </td>
</tr>
<tr>
    <td>channel</td>
    <td>string</td>
    <td>"email"</td>
    <td>Тип канала</td>
</tr>
<tr>
    <td>email</td>
    <td>string</td>
    <td>"example@example.com"</td>
    <td>Email-адрес</td>
</tr>
<tr>
    <td colspan="4">
        <strong>SMS канал</strong>
    </td>
</tr>
<tr>
    <td>channel</td>
    <td>string</td>
    <td>"sms"</td>
    <td>Тип канала</td>
</tr>
<tr>
    <td>phone</td>
    <td>string</td>
    <td>"+79000000000"</td>
    <td>Номер телефона</td>
</tr>
<tr>
    <td colspan="4">
        <strong>Push канал</strong>
    </td>
</tr>
<tr>
    <td>channel</td>
    <td>string</td>
    <td>"push"</td>
    <td>Тип канала</td>
</tr>
<tr>
    <td>
        <a href="/developer-guide/miscellaneous/37848490">provider</a>
    </td>
    <td>string</td>
```

```
    <td>"Firefox"</td>
    <td>Тип провайдера</td>
</tr>
<tr>
    <td>subscription_id</td>
    <td>string</td>
    <td>"abcdefghijklmnqrstuvwxyz"</td>
    <td>Идентификатор подписки</td>
</tr>
<tr>
    <td colspan="4">
        <strong>Telegram Bot канал</strong>
    </td>
</tr>
<tr>
    <td>channel</td>
    <td>string</td>
    <td>"telegram_bot"</td>
    <td>Тип канала</td>
</tr>
<tr>
    <td>cc_data</td>
    <td>object</td>
    <td>
{
    "id": 1
}
    </td>
    <td>id чата в Telegram-бот</td>
</tr>
<tr>
    <td colspan="4">
        <strong>WhatsApp*-канал</strong>
    </td>
</tr>
<tr>
    <td>channel</td>
    <td>string</td>
    <td>"whatsapp"</td>
    <td>Тип канала</td>
</tr>
<tr>
    <td>cc_data</td>
    <td>object</td>
    <td>
{
    "phone": "+79000000000"
}
```

```
    </td>
    <td>Телефон профиля в WhatsApp*</td>
</tr>
<tr>
    <td colspan="4">
        <strong>Viber канал</strong>
    </td>
</tr>
<tr>
    <td>channel</td>
    <td>string</td>
    <td>"viber"</td>
    <td>Тип канала</td>
</tr>
<tr>
    <td>cc_data</td>
    <td>object</td>
    <td>
{
    "phone": "+79000000000"
}
    </td>
    <td>Телефон профиля в Viber</td>
</tr>
```

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "matching": "custom",
    "field_name": "Custom_ID",
    "skip_triggers": true,
    "data": [
        {
            "_fname": "Lala",
            "_lname": "Doe",
            "Custom_ID": 77,
            "email": "example@example.com",
            "subscriptions": [
                {
                    "channel": "email",
                    "email_data": {
                        "email": "example@example.com"
                    },
                    "resource_id": 28,
                    "custom_fields": {
                        "custom_field": "test value"
                    },
                    "cats": [
                        "category_1",
                        "category_2"
                    ]
                }
            ]
        },
        {
            "_fname": "Victor",
            "Custom_ID": 88,
            "email": "example@example.com"
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <matching>custom</matching>
    <field_name>Custom_ID</field_name>
    <skip_triggers>true</skip_triggers>
    <data>
        <_fname>Lala</_fname>
        <_lname>Doe</_lname>
        <Custom_ID>77</Custom_ID>
        <email>example@example.com</email>
        <subscriptions array='true'>
            <channel>email</channel>
            <email_data>
                <email>example@example.com</email>
            </email_data>
            <resource_id>28</resource_id>
            <custom_fields>
                <custom_field>test value</custom_field>
            </custom_fields>
            <cats>category_1</cats>
            <cats>category_2</cats>
        </subscriptions>
    </data>
    <data>
        <_fname>Alex</_fname>
        <Custom_ID>345</Custom_ID>
        <email>example@example.com</email>
    </data>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "result": [
    {
      "error": 0,
      "error_text": "Successful operation",
      "profile_id": "abcdefghijklmnqrstuvwxyz"
    }
  ]
}
```

XML

```
<xml>
 <error>0</error>
  <error_text>Successful operation</error_text>
  <result>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
  </result>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки, для всего запроса
error_text	string	Текст ошибки, для всего запроса
data	array of objects	Результаты обновления по профилям
data.error	int	Код ошибки, для профиля
data.error_text	string	Текст ошибки, для для профиля
data.profile_id	string	Идентификатор профиля

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Приостановить все профили в базе данных

> **NOTE**
>
> Этот API считается устаревшим и находится здесь только для информации. Пожалуйста, не настраивайте новые интеграции на этот метод.

Описание

Метод изменяет глобальный статус всех профилей в базе данных с “подписан” на “приостановлен”.

URL-адрес

Метод: POST

`https://example.com/api/v1.0/lists/{list_id}/suspend_all?token={token}`

`{list_id}` – идентификатор базы данных.

`{token}` – API-токен.

Пример ответа

```
{
        "Status":  "Ok",
        "Suspended": 4
}
```

Возвращаемые параметры

Параметр	Тип	Описание
Status	string	Статус запроса
Suspended	int	Количество профилей

Восстановить все приостановленные профили в базе данных

> **NOTE**
>
> Этот API считается устаревшим и находится здесь только для информации. Пожалуйста, не настраивайте новые интеграции на этот метод.

Описание

Изменяет глобальный статус всех профилей базы данных на “подписан”, если у них был статус “приостановлен”.

URL-адрес

Метод: POST.

`https://example.com/api/v1.0/lists/{list_id}/unsuspend_all?token={token}`

`{list_id}` – идентификатор базы данных.

`{token}` – API-токен.

Удалить профиль

Описание

Полное удаление профиля.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/delete`

> **CAUTION**
>
> Данный запрос, помимо самого профиля, синхронно удаляет всю его историю и связи.

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных
Поиск профиля подписчика				
matching	string	"email" "email_profile" "phone" "profile_id" "custom" </td> <td>Нет, если поиск по email из профиля или подписок</td> <td> Режим поиска подписчика. По умолчанию - email. </td>		

email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю</td> <td>Email-адрес</td>	
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. **TIP** Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

CAUTION

Для удаления профиля необходимы соответствующие полномочия, которые настраиваются в роли, после чего соответствующая роль может быть назначена токену. Подробнее о ролях можно узнать здесь.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "matching": "email",
    "email": "example@example.com"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
  <matching>email</matching>
  <email>example@example.com</email>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "profile_id": "5f33d33ece94481aa9382110"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <profile_id>5f33d33ece94481aa9382110</profile_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	JSON object	Информация о профиле
error	int	Код ошибки
error_text	string	Текст ошибки
profile_id	string	Идентификатор профиля

Упрощенный импорт профиля

Описание

Упрощенный импорт профиля имеет линейную структуру и предназначен для простейших интеграций.

Для получения доступа к большему функционалу воспользуйтесь методом `/api/v1.1/profiles/import`.

Импортирует или обновляет профиль, добавляет подписки, если передать номер ресурса и есть контактные данные.

URL-адрес

Метод: POST. Данные могут быть переданы как в параметрах URL, так и в теле запроса, если это POST.

`https://example.com/api/integrations/any/profile_import`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных
matching	string	“email”	Нет	Способ поиска профиля для обновления: • email - поиск по email из профиля или подписок • email_profile - поиск по email из профиля • phone - поиск по телефону из профиля или подписок • custom - свое поле, указанное в `field_name` В это поле также можно передать имя поля для поиска (`field_name` вместо `custom`). По умолчанию – email.
resource_id	int	1	Нет	Сформировать подписку на указанный ресурс, на базе импортируемых данных.
email	string	“example@example.com”	Нет	Адрес email подписки. Также требуется при поиске по email.
phone	string	“+75551111444”	Нет	Номер телефона. Также требуется при поиске по телефону.

Параметр	Тип	Пример	Обязательный	Описание
_regip	string	“172.16.0.1”	Нет	IPv4 или IPv6 адрес регистрации. В случае передачи IPv4 адреса будет произведена попытка геолокации.
_regurl	string	“https://example.com/?”	Нет	Если не передан — будет установлен из HTTP заголовка Referer.
*				Можно передавать любые поля, имеющиеся в БД. Хотя бы одно поле требуется для успешного импорта. Для получения информации о полях базы данных, используйте /api/v1.1/databases/fields_get
trigger_id	int	13	Нет	Идентификатор триггерной рассылки для запуска. Если `trigger_id` задан, то произойдет одновременный импорт профиля и отправка триггера, если нет — обычный импорт профиля.
workflow_id	int	1	Нет	Идентификатор цепочки. Если `workflow_id` задан, то произойдет одновременный импорт профиля и запуск профиля в сценарий, если нет — обычный импорт профиля.

Ответы сервиса

Упрощенный импорт отвечает HTTP кодом и текстом ошибки в теле сообщения.

HTTP Код	Пояснение
200	Успешный импорт.
400	Ошибки в формировании запроса, проверьте правильность передаваемых полей.
402	Достигнуты ограничения аккаунта по тарифу.
403	API ключ не передан или не правильный, у него нет прав на запись в базы данных.
404	Не найдена база данных для импорта или ресурс для подписки.
409	Для заданного поиска найден более чем один профиль, это значит что поиск недостаточно точен.
435	Невозможно соблюсти уникальность поля при сохранении данных, обычно это поле email, когда поиск осуществляется не по email, а, например, по кастомному полю.
500	Внутренняя ошибка сервиса.

Пример запроса

```
curl "https://example.com/api/integrations/any/profile_import" \
    -d token="abcdefghijklmnqrstuvwxyz" \
    -d db_id=1 \
    -d resource_id=1 \
    -d email="john.doe@example.com" \
    -d _fname="John" \
    -d _lname="Doe"
```

Пример ответа

```
Successfully added 5beeea0ad861602a078a6f34
```

Функциональное обновление полей базы

При использовании запросов для обновления полей профиля можно передавать дополнительные инструкции прямо в поле.

Инструкция имеет определенную структуру:

```
{
    "action": "set",
    "value": "x",
}
```

Поле `action` может принимать одно из следущих значений:

action	Назначение	Тип поля	value
set	Работает так же, как и простое обновление поля, устанавливает значение	все типы: date, string, enum, phone, int, float, tags, object, arr_object	Зависит от типа поля в бд
unset	Удаляет поле из структуры профиля (для всех полей, кроме `email`)	все типы: date, string, enum, phone, int, float, tags, object, arr_object	Нет
incr	Увеличивает значение поля	int, float	Число
add	Добавляет элементы в поле профиля	tags, arr_object	Массив строк
delete	Удаляет элементы из поля профиля	tags, arr_object	Массив строк
upsert	Устанавливает значение из поля *value*в свойство, указанное в поле *submatch*	object	Зависит от типа поля

NOTE

Для типов данных `object`, `arr_object` необходимо передавать поля `value` и `submatch` (путь к свойству, значение которого необходимо обновить).

Пример запроса `profiles/update`, уменьшающий значение одного поля и увеличивающий другое:

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 24,
    "profile_id": "5a5b64a969d4265b958b9ac3",
    "data": {
        "checks": {
            "action": "incr",
            "value": 1000
        },
        "returns": {
            "action": "incr",
            "value": -1000
        }
    }
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
  <profile_id>6613c4754a5d12bd6cc0915f</profile_id>
  <data>
    <checks>
      <action>incr</action>
      <value>1000</value>
    </checks>
    <returns>
      <action>incr</action>
      <value>-1000</value>
    </returns>
  </data>
</xml>
```

Пример запроса `profiles/update`, манипулирующий тегами:

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "db_id": 24,
  "profile_id": "5a5b64a969d4265b958b9ac3",
  "data": {
    "category": {
       "action": "add",
       "value": ["cars", "cats"]
    }
  }
}
```

XML

```
<xml>
 <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
  <profile_id>6613c4bf4a5d12bd6cc0916b</profile_id>
  <data>
    <category>
      <action>add</action>
      <value array='true'>cars</value>
      <value array='true'>cats</value>
    </category>
  </data>
</xml>
```

Пример запроса `profiles/update`, который обновит поле *“x”* в объекте:

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "db_id": 1,
  "profile_id": "6613c4bf4a5d12bd6cc0916b",
  "data": {
    "myObject": {
       "action": "upsert",
       "value": "y",
       "submatch": "x"
    }
  }
}
```

XML

```
<xml>
    <token>2fefb577533d4cae919350c450755239</token>
    <db_id>1</db_id>
    <profile_id>66102b984a5d12bd6cc0912f</profile_id>
    <data>
        <myObject>
            <action>upsert</action>
            <value>y</value>
            <submatch>x</submatch>
        </myObject>
    </data>
</xml>
```

Пример запроса `profiles/update`, который установит новое значение (`value`) для всех элементов объекта, содержащих указанное свойство (`submatch`):

Запрос

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "profile_id": "66102b984a5d12bd6cc0912f",
    "data": {
        "myObject": {
            "action": "upsert",
            "value": {
                "x": "y",
                "z": 1
            },
            "submatch": "x"
        }
    }
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
  <profile_id>66102b984a5d12bd6cc09132</profile_id>
  <data>
    <myObject>
      <action>upsert</action>
      <value>
        <x>y</x>
        <z>1</z>
      </value>
      <submatch>x</submatch>
    </myObject>
  </data>
</xml>
```

Объект

JSON

```
{
    {
       "A": 1,
       "x": false,
    },
    {
       "B": "test"
    },
    {
       "x": "test",
       "C": true
    }
 }
```

XML

```
<xml>
    <value>
        <A>1</A>
        <x>false</x>
    </value>
    <value>
        <B>test</a>
    </value>
    <value>
        <x>test</x>
        <c>true</c>
    </value>
</xml>
```

Обновленный объект

JSON

```
{
    {
       "A": 1,
       "x": {"x": "y", "z": 1},
    },
    {
       "B": "test"
    },
    {
       "x": {"x": "y", "z": 1},
       "C": true
    }
}
```

XML

```
<xml>
    <value>
        <A>1</A>
        <x>
            <m>y</m>
            <z>1</z>
        </x>
    </value>
    <value>
        <B>test</B>
    </value>
    <value>
        <x>
            <m>y</m>
            <z>1</z>
        </x>
        <c>true</c>
    </value>
</xml>
```

NOTE

Если у полей с типом `object` и `object_arr` отсутствует свойство, которое вы хотите изменить (`action`), то это свойство создается автоматически.

Функциональное обновление полей подписки

При использовании запросов для обновления полей подписки профиля на ресурс можно передавать дополнительные инструкции прямо в поле.

Инструкция имеет определенную структуру:

```
{
    "action": "set",
    "value": "x",
}
```

Поле `action` может принимать одно из следущих значений:

action	Назначение	Тип поля	value
set	Работает так же, как и простое обновление поля, устанавливает значение	все типы: string, int, float, boolean	Зависит от типа поля
delete	Удаляет поле из структуры профиля, только для Custom полей	все типы: string, int, float, boolean	Нет
incr	Увеличивает значение поля	int, float	Число

TIP

Описание полей подписки на ресурс вы найдёте по этой ссылке.

Пример запроса /v1.1/subscriptions/set для обновления системных полей подписки

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "channel": "email",
    "db_id": 1,
    "email": "example@example.com",
    "profile_id": "6527e4e2097c54377ee0e41d",
    "resource_id": 1,
    "status": "subscribe",
     "cats": [
        "category_1",
        "category_2"
    ],
    "custom_fields": {
        "_ad_id": {
            "action":"set",
            "value":"qwerty123"
        },
        "_ad_track":{
            "action":"set",
            "value":true
        },
        "_app_id": {
            "action":"delete"
        },
        "_app_iid": {
            "action":"set",
            "value":"example iid"
        },
        "_app_ver": {
            "action":"set",
            "value": "v1.2.3.4"
        },
        "_browser_name":{
            "action":"set",
            "value":"Chrome"
        },
        "_browser_ver": {
            "action":"set",
            "value":"4.5.6.7.8"
        },
        "_device_model": {
            "action":"set",
            "value":"Device model"
        },
        "_device_name":{
            "action":"set",
            "value":"Device name"
        },
        "_device_type": {
```

```
        "_device_type": {
            "action":"set",
            "value":"Mobile"
        },
        "_os": {
            "action":"set",
            "value":"Android"
        },
        "_os_language": {
            "action":"set",
            "value":"ru"
        },
        "_os_tz": {
            "action":"set",
            "value":"Europe/Moscow"
        },
        "_os_ver": {
            "action":"set",
            "value": "v1.2.3.4"
        }
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <channel>email</channel>
    <db_id>1</db_id>
    <email>example@example.com</email>
    <profile_id>66102b984a5d12bd6cc09132</profile_id>
    <resource_id>29</resource_id>
    <status>subscribe</status>
    <cats>category_1</cats>
    <cats>category_2</cats>
    <custom_fields>
        <_ad_id>
            <action>set</action>
            <value>qwerty123</value>
        </_ad_id>
        <_ad_track>
            <action>set</action>
            <value>true</value>
        </_ad_track>
        <_app_id>
            <action>delete</action>
        </_app_id>
        <_app_iid>
            <action>set</action>
            <value>example iid</value>
        </_app_iid>
        <_app_ver>
            <action>set</action>
            <value>v1.2.3.4</value>
        </_app_ver>
        <_browser_name>
            <action>set</action>
            <value>Chrome</value>
        </_browser_name>
        <_browser_ver>
            <action>set</action>
            <value>4.5.6.7.8</value>
        </_browser_ver>
        <_device_model>
            <action>set</action>
            <value>Device model</value>
        </_device_model>
        <_device_name>
            <action>set</action>
            <value>Device name</value>
        </_device_name>
        <_device_type>
            <action>set</action>
            <value>Mobile</value>
```

```
            <value>Mobile</value>
        </_device_type>
        <_os>
            <action>set</action>
            <value>Android</value>
        </_os>
        <_os_language>
            <action>set</action>
            <value>ru</value>
        </_os_language>
        <_os_tz>
            <action>set</action>
            <value>Europe/Moscow</value>
        </_os_tz>
        <_os_ver>
            <action>set</action>
            <value>v1.2.3.4</value>
        </_os_ver>
    </custom_fields>
</xml>
```

Пример запроса /v1.1/subscriptions/set для обновления кастомных полей подписки

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "channel": "email",
    "db_id": 1,
    "email": "example@example.com",
    "profile_id": "6527e4e2097c54377ee0e41d",
    "resource_id": 1,
    "status": "subscribe",
    "cats": [
        "category_1",
        "category_2"
    ],
    "custom_fields": {
        "custom_str_field": {
            "action": "set",
            "value": "example value"
        },
        "custom_int_field_1": {
            "action": "incr",
            "value": 1000
        },
        "custom_int_field_2": {
            "action": "incr",
            "value": -1000
        },
        "custom_bool_field": {
            "action": "delete"
        }
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <channel>email</channel>
    <db_id>1</db_id>
    <email>example@example.com</email>
    <profile_id>6613c4bf4a5d12bd6cc0916b</profile_id>
    <resource_id>29</resource_id>
    <status>subscribe</status>
    <cats>category_1</cats>
    <cats>category_2</cats>
    <custom_fields>
        <custom_str_field>
            <action>set</action>
            <value>example value</value>
        </custom_str_field>
        <custom_int_field_1>
            <action>incr</action>
            <value>1000</value>
        </custom_int_field_1>
        <custom_int_field_2>
            <action>incr</action>
            <value>-1000</value>
        </custom_int_field_2>
        <custom_bool_field>
            <action>delete</action>
        </custom_bool_field>
    </custom_fields>
</xml>
```

Пример запроса /v1.1/profiles/import для обновления системных полей подписки

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "matching": "email",
    "email": "example@example.com",
    "data": {
        "_fname": "John",
        "_lname": "Doe",
        "email": "example@example.com",
        "subscriptions": [
            {
                "channel": "push",
                "subscription_id": "9f135a38-aa7e-4bf3-8d75-de971d597cf0",
                "provider": "MR",
                "resource_id": 1,
                "custom_fields": {
                    "_ad_id": {
                        "action": "set",
                        "value": "qwerty123"
                    },
                    "_ad_track": {
                        "action": "set",
                        "value": true
                    },
                    "_app_id": {
                        "action": "set",
                        "value": "123"
                    },
                    "_app_iid": {
                        "action": "set",
                        "value": "123"
                    },
                    "_app_ver": {
                        "action": "set",
                        "value": "v1.2.3.4"
                    },
                    "_browser_name": {
                        "action": "set",
                        "value": "Chrome"
                    },
                    "_browser_ver": {
                        "action": "set",
                        "value": "4.5.6.7.8"
                    },
                    "_device_model": {
                        "action": "set",
                        "value": "Device model"
                    },
                    "_device_name": {
```

```
                    "_device_name": {
                        "action": "set",
                        "value": "Device name"
                    },
                    "_device_type": {
                        "action": "set",
                        "value": "Mobile"
                    },
                    "_os": {
                        "action": "set",
                        "value": "Android"
                    },
                    "_os_language": {
                        "action": "set",
                        "value": "ru"
                    },
                    "_os_tz": {
                        "action": "set",
                        "value": "Europe/Moscow"
                    },
                    "_os_ver": {
                        "action": "set",
                        "value": "v1.2.3.4"
                    }
                }
            }
        ]
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <matching>email</matching>
    <email>example@example.com</email>
    <data>
        <_fname>Fillippina</_fname>
        <_lname>Doe</_lname>
        <email>example@example.com</email>
        <subscriptions array='true'>
            <channel>push</channel>
            <subscription_id>9f135a38-aa7e-4bf3-8d75-de971d597cf0</subscription_id>
            <provider>MR</provider>
            <resource_id>26</resource_id>
            <custom_fields>
                <_ad_id>
                    <action>set</action>
                    <value>qwerty123</value>
                </_ad_id>
                <_ad_track>
                    <action>set</action>
                    <value>true</value>
                </_ad_track>
                <_app_id>
                    <action>delete</action>
                </_app_id>
                <_app_iid>
                    <action>set</action>
                    <value>example iid</value>
                </_app_iid>
                <_app_ver>
                    <action>set</action>
                    <value>v1.2.3.4</value>
                </_app_ver>
                <_browser_name>
                    <action>set</action>
                    <value>Chrome</value>
                </_browser_name>
                <_browser_ver>
                    <action>set</action>
                    <value>4.5.6.7.8</value>
                </_browser_ver>
                <_device_model>
                    <action>set</action>
                    <value>Device model</value>
                </_device_model>
                <_device_name>
                    <action>set</action>
                    <value>Device name</value>
```

```
                    <value>Device name</value>
                </_device_name>
                <_device_type>
                    <action>set</action>
                    <value>Mobile</value>
                </_device_type>
                <_os>
                    <action>set</action>
                    <value>Android</value>
                </_os>
                <_os_language>
                    <action>set</action>
                    <value>ru</value>
                </_os_language>
                <_os_tz>
                    <action>set</action>
                    <value>Europe/Moscow</value>
                </_os_tz>
                <_os_ver>
                    <action>set</action>
                    <value>v1.2.3.4</value>
                </_os_ver>
            </custom_fields>
        </subscriptions>
    </data>
</xml>
```

Пример запроса /v1.1/profiles/import для обновления кастомных полей подписки

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "matching": "email",
    "email": "example@example.com",
    "data": {
        "_fname": "John",
        "_lname": "Doe",
        "email": "example@example.com",
        "subscriptions": [
            {
                "channel": "email",
                "channel_sid": "email",
                "email": "example@example.com",
                "resource_id": 1
            },
            {
                "channel": "sms",
                "phone": "+79203214111",
                "resource_id": 1
            },
            {
                "channel": "push",
                "subscription_id": "9f135a38-aa7e-4bf3-8d75-de971d597cf0",
                "provider": "MR",
                "resource_id": 1,
                "custom_fields": {
                    "custom_str_field": {
                        "action": "set",
                        "value": "example value"
                    },
                    "custom_int_field": {
                        "action": "incr",
                        "value": 1000
                    },
                    "custom_bool_field": {
                        "action": "delete"
                    }
                }
            }
        ]
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <matching>email</matching>
    <email>example@example.com</email>
    <data>
        <_fname>Fillippina</_fname>
        <_lname>Doe</_lname>
        <email>example@example.com</email>
        <subscriptions>
            <channel>email</channel>
            <channel_sid>email</channel_sid>
            <email>example@example.com</email>
            <resource_id>1</resource_id>
        </subscriptions>
        <subscriptions>
            <channel>sms</channel>
            <phone>+79203214111</phone>
            <resource_id>1</resource_id>
        </subscriptions>
        <subscriptions>
            <channel>push</channel>
            <subscription_id>9f135a38-aa7e-4bf3-8d75-de971d597cf0</subscription_id>
            <provider>MR</provider>
            <resource_id>1</resource_id>
            <custom_fields>
                <custom_str_field>
                    <action>set</action>
                    <value>example value</value>
                </custom_str_field>
                <custom_int_field>
                    <action>incr</action>
                    <value>1000</value>
                </custom_int_field>
                <custom_bool_field>
                    <action>delete</action>
                </custom_bool_field>
            </custom_fields>
        </subscriptions>
    </data>
</xml>
```

Выгрузка профилей в файл

Описание

Получить информацию о всех профилях подписчиков базы данных или ресурса.

Для получения информации необходимо выполнить несколько запросов: **выгрузку данных в файл, проверку статуса задания и загрузку файла**.

Шаг 1. Выгрузка данных в файл

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/get_all`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
source_type	string	“database” “segment”	Да	Тип источника данных
source_id	int	1	Да	Идентификатор источника данных
file_format	string	“csv”	Нет	Формат файла для экспорта. По умолчанию - csv.
custom_request	JSON object	`{"occupation":"doctor"}`	Нет	Поля профиля для уточнения поиска

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "source_type": "database",
    "source_id": 1,
    "file_format": "csv"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <source_type>database</source_type>
  <source_id>1</source_id>
  <file_format>csv</file_format>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "task_id": "task_b2f9fd6a-2499-4a63-b592-f15d4e3b69c5"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <task_id>task_b2f9fd6a-2499-4a63-b592-f15d4e3b69c5</task_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
task_id	string	Идентификатор задания на экспорт
error	int	Код ошибки
error_text	string	Текст ошибки

Шаг 2. Проверка статуса экспорта и получение ссылки

URL-адрес

Метод: POST

`https://example.com/api/v1.1/tasks/status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
task_id	string	“task_b2f9fk6a-2479-4a63-b592-f15d4e3b69c5”	Да	Идентификатор задания

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "task_id": "task_b2f9fk6a-2479-4a63-b592-f15d4e3b69c5"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <task_id>task_b2f9fk6a-2479-4a63-b592-f15d4e3b69c5</task_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "status": "success",
        "progress": 100,
        "result": {
            "source_id": 1,
            "total": 10,
            "type": "database",
            "urls": [
                "/api/download/6b248de2-39b9-4715-9720-5f4e23f4a296/1_database_export.csv"
            ]
        }
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
<data>
    <status>success</status>
    <progress>100</progress>
    <result>
      <source_id>1</source_id>
      <total>10</total>
      <type>database</type>
      <urls>/api/download/6b248de2-39b9-4715-9720-5f4e23f4a296/1_database_export.csv</urls>
    </result>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
status	string	Статус задания
progress	int	Прогресс выполнения
total	int	Количество профилей
urls	array	Ссылка на скачивание
error	int	Код ошибки
error_text	string	Текст ошибки

Шаг 3. Загрузка файла

URL-адрес

Ссылка на загрузку файла выглядит подобным образом:

`https://example.com/api/download/6b248de2-33b9-4715-9720-5f7e23f4a296/1_database_export.csv?token=abcdefghijklmnqrstuvwxyz`, где:

- **`https://example.com`** — ссылка на панель управления платформой
- **`api/download/6b248de2-33b9-4715-9720-5f7e23f4a296/1_database_export.csv`** — ссылка из параметра `urls` прошлого запроса
- **`?token=abcdefghijklmnqrstuvwxyz`** — API-токен пользователя (после знака равенства).

Для загрузки файла с профилями необходимо либо открыть получившуюся ссылку в браузере, либо отправить пустой **GET** запрос с указанием данной ссылки в URL.

Получение данных по нескольким профилям

Описание

Получение данных по нескольким профилям.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
db_id	int	20	Да	Идентификатор базы данных
from_id	string	“0014a2feb331ace2.20200124134127”	Нет	Указатель на следующую страницу, берется из предыдущего запроса или передается пустой для первого.
limit	int	3	Нет	Лимит количества записей. По умолчанию - 1000.

query	JSON обект	“query”: {
		“$or”:[
		{
		“_lname”: “Black”
		},
		{
		“_city”: “Bay City”
		}
		]
		}
			Нет	Уточнение поиска профилей по параметрам в виде запроса к базе данных в формате JSON. Запрос формируется согласно документации к MongoDB.

Параметр	Тип	Пример	Обязательный	Описание
fields	Array of string	“fields”: [“_id”, “_lname”, “_fname”, “_city”, “subscriptions”],	Нет	Содержит поля для фильтрации отображаемых данных о профиле.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "from_id": "65117f62736a0e15d273fae9",
    "limit": 1,
    "query": {
        "$or": [
            {
                "_lname": "Doe"
            },
            {
                "_city": "Moscow"
            }
        ]
    },
    "fields": [
        "_id",
        "_lname",
        "_fname",
        "_city",
        "subscriptions"
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <from_id>65117f62736a0e15d273fae9</from_id>>
    <limit>1</limit>
    <fields>_id</fields>
    <fields>_lname</fields>
    <fields>_fname</fields>
    <fields>_city</fields>
    <fields>subscriptions</fields>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "_city": "Moscow",
            "_fname": "Anna",
            "_lname": "Murphy",
            "id": "65117f62736a0e15d273faee",
            "subscriptions": [
                {
                    "resource_id": 5,
                    "channel": "email",
                    "email_data": {
                        "email": "example@example.edu",
                        "domain": "example.edu",
                        "domain_group": "other",
                        "md5": "7dbeb0466131e377e58988a97196d18e"
                    },
                    "priority": 0,
                    "status": "unsubscribed",
                    "reg_info": {
                        "date": "2024-01-09T12:12:18.172Z"
                    },
                    "hash_id": "e70dbabe",
                    "is_delete": false
                },
                {
                    "resource_id": 26,
                    "channel": "push",
                    "push_data": {
                        "subscription_id": "example@example.edu",
                        "provider": "Firefox"
                    },
                    "priority": 0,
                    "status": "subscribed",
                    "reg_info": {
                        "date": "2024-04-18T12:30:44.592Z"
                    },
                    "hash_id": "e70dbabe",
                    "is_delete": false
                }
            ]
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "next_from_id": "65117f62736a0e15d273faf4"
}
```

XML

```
<xml>
    <data>
        <_city>Toronto</_city>
        <_fname>Juan</_fname>
        <_lname>Wallace</_lname>
        <id>65117f62736a0e15d273fae9</id>
        <subscriptions>
            <channel>email</channel>
            <email_data>
                <domain>@example.edu</domain>
                <domain_group>other</domain_group>
                <email>example@example.edu</email>
                <md5>18450b1edfef75279fc4f4f60c78d97e</md5>
            </email_data>
            <hash_id>a0cc2672</hash_id>
            <is_delete>false</is_delete>
            <priority>0</priority>
            <reg_info>
                <date>2024-01-09T12:12:18.172Z</date>
            </reg_info>
            <resource_id>5</resource_id>
            <status>hardbounced</status>
        </subscriptions>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <next_from_id>65117f62736a0e15d273faea</next_from_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	string	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории.
data	array of objects	Массив полученных данных по профилям

Объединение нескольких профилей

Описание

Объединение нескольких профилей в новый или уже существующий профиль.

CAUTION

- Профили, созданные объединением, — это не импорт. Импорт триггер по ним не срабатывает.
- `Email: localhost` как отсутствующее поле, создается при необходимости и не копируется в другой профиль.
- Не localhost email копируется, как любое другое поле, но чтобы не было коллизий — пропадает из source-профиля, меняется на пустое (то есть `localhost`).

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/merge`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
data	JSON object	“data”: { “_fname”: “John”, “_lname”: “Doe” }	Нет	Данные о профиле, которые изменятся после объединения.
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false)

Параметр	Тип	Пример	Обязательный	Описание

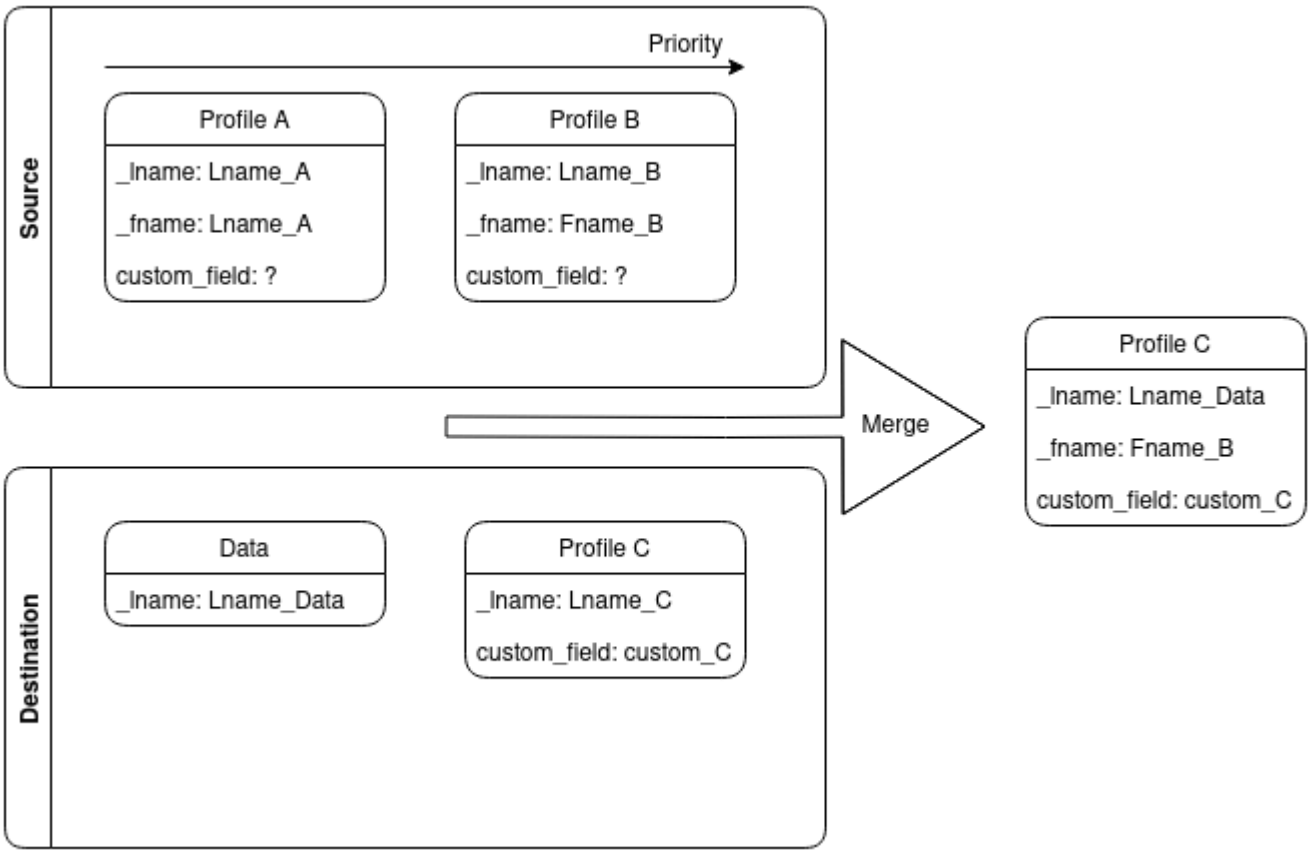
source	array	“source”: [
		{
		“db_id”: 1,
		“email”: “test1@mail.ru”
		},
		{
		“db_id”: 2,
		“email”: “test2@mail.ru”
		}
		]
			Да	Содержит данные матчинга о профилях, которые нужно объединить.

destination	JSON object	“destination”: {
		“db_id”: 3,
		“email”: “test3@mail.ru”
		}
			Да	Содержит данные матчинга профиля, который получится в результате объединения.

matching	string	“email” “email_profile” “email_sub” “phone” “phone_sub” “profile_id” “push_sub” “custom” “custom_sub” “email_phone” “email_phone_sub”	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию - email. Для каждого типа матчинга в теле запроса необходимо передавать определенные поля. Подробнее об этих полях можно узнать здесь.
field_name	string	custom_ID	Да, если “matching”: “custom”	Имя кастомного поля БД Altcraft при **matching : custom**. В форме сбора данных для поиска нужно поле с таким же именем.
field_value	int/string	custom_string	Да, если “matching”: “custom”	Значение кастомного поля при **matching : custom**.
subscriptions	bool	true	Нет	Копировать и объединить подписки. Подписки на категории ресурса и данные из полей подписки также скопируются. (по умолчанию – true)

Параметр	Тип	Пример	Обязательный	Описание
history	bool	true	Нет	Копировать историю и флаги активности (по умолчанию – true)
static_segments	bool	true	Нет	Вносить ли изменения в статические сегменты (source-профили выйдут из сегментов, а destination-профиль - войдет) (по умолчанию – false)
overwrite	bool	true	Нет	Копировать ли поля source-профилей (по умолчанию – false) Если у destination-профиля есть пустые поля, они в любом случае будут заполнены полями профиля с наибольшим приоритетом.
delete	bool	true	Нет	Удалить ли source-профили после обьединения (по умолчанию – false)

Копирование полей профилей



Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "matching": "custom",
    "field_name": "field_1",
    "source": [
        {
            "db_id": 1,
            "field_value": "1",
            "data": {
                "_fname": "Profile1",
                "_lname": "LnameProfile1"
            }
        },
        {
            "db_id": 1,
            "matching": "email",
            "email": "test2@mail.ru",
            "data": {
                "_fname": "Profile2",
                "_lname": "LnameProfile2"
            }
        }
    ],
    "destination": {
        "db_id": 2,
        "matching": "email",
        "email": "test3@mail.ru",
        "data": {
            "_fname": "Profile3",
            "_lname": "LnameProfile3"
        }
    },
    "subscriptions": true,
    "static_segments": true,
    "history": false
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <matching>custom</matching>
    <field_name>field_1</field_name>
    <source>
        <db_id>1</db_id>
        <field_value>1</field_value>
        <data>
            <_fname>Profile1</_fname>
            <_lname>LnameProfile1</_lname>
        </data>
    </source>
    <source>
        <db_id>1</db_id>
        <matching>email</matching>
        <email>test2@mail.ru</email>
        <data>
            <_fname>Profile2</_fname>
            <_lname>LnameProfile2</_lname>
        </data>
    </source>
    <destination>
        <db_id>2</db_id>
        <matching>email</matching>
        <email>test3@mail.ru</email>
        <data>
            <_fname>Profile2</_fname>
            <_lname>LnameProfile2</_lname>
        </data>
    </destination>
    <subscriptions>true</subscriptions>
    <static_segments>true</static_segments>
    <history>false</history>
</xml>
```

Пример ответа

JSON

```
{
  "destination_id": {
    "db_id": 2,
    "profile_id": "60a7855930b8bc8535b2acfb"
  },
  "error": 0,
  "error_text": "",
  "source_ids": [
    {
      "db_id": 1,
      "profile_id": "60a7839330b8bc86a5bc5f62"
    },
    {
      "db_id": 1,
      "profile_id": "60a7845230b8bc8535b2acf7"
    }
  ]
}
```

XML

```
<xml>
    <destination_id>
        <db_id>2</db_id>
        <profile_id>60a7855930b8bc8535b2acfb</profile_id>
    </destination_id>
    <error>0</error>
    <error_text></error_text>
    <source_ids>
        <db_id>1</db_id>
        <profile_id>60a7839330b8bc86a5bc5f62</profile_id>
    </source_ids>
    <source_ids>
        <db_id>1</db_id>
        <profile_id>60a7845230b8bc8535b2acf7</profile_id>
    </source_ids>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
destination_id	JSON объект	Содержит данные о профиле, получившемся в результате объединения: `db_id` — идентификатор базы данных `profile_id` — идентификатор профиля

Параметр	Тип	Описание
source_ids	JSON объект	Содержит данные об объединенных профилях в виде списка: `db_id` — идентификатор базы данных `profile_id` — идентификатор профиля
error	int	Код ошибки
error_text	string	Текст ошибки

Отписать профиль от ресурса

Описание

Передает в платформу информацию об отписке профиля. Статус подписки профиля на ресурс меняется на “Отписан”. Также этот метод позволяет отписать от определенных категорий ресурса.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/message_unsub`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
rtoken	string	“abcdefghijklmnqrstuvwxyz”	Да	Токен ресурса
smid	string	“abcdefghijklmnqrstuvwxyz”	Да	Идентификатор отправляемого сообщения
categories	array	[“cat_to_unsub”]	Нет	Категория ресурса

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "rtoken":"2fHLS9YR9daS-a90b371eecc2952f",
  "smid": "fGmXiWU9Y4NSIpCwl4QG6QXAAAQEQsQQQEQEAABCRA",
  "categories": ["cat_to_unsub"]
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <rtoken>2fHLS9YR9daS-a90b371eecc2952f</rtoken>
  <smid>fGmXiWU9Y4NSIpCwl4QG6QXAAAQEQsQQQEQEAABCRA</smid>
  <categories>cat_to_unsub</categories>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "profile_id": "658997338e8f4032d92fd56d"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <profile_id>658997338e8f4032d92fd56d</profile_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_id	string	Идентификатор профиля

Разделение профиля

Описание

Разделение профиля на несколько новых или уже существующих профилей.

> **CAUTION**
>
> Профили, созданные разделением, — это не импорт. Импорт триггер по ним не срабатывает.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/split`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
data	JSON object	“data”: { “_fname”: “John”, “_lname”: “Doe” }	Нет	Данные о профиле, которые изменятся после разделения.
skip_triggers	bool	true	Нет	Пропустить запуск триггеров по умолчанию – false
source	JSON object	“source”: { “db_id”: 1, “email”: “test1@mail.ru” }	Да	Содержит данные матчинга о профилях, которые нужно разделить

Параметр	Тип	Пример	Обязательный	Описание
destination	array	“destination”: [{ “db_id”: 3, “email”: “test3@mail.ru” }]	Да	Содержит данные матчинга профиля, который получится в результате разделения
matching	string	“email” “email_profile” “email_sub” “phone” “phone_sub” “profile_id” “push_sub” “custom” “custom_sub” “email_phone” “email_phone_sub”	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию - email. Для каждого типа матчинга в теле запроса необходимо передавать определенные поля. Подробнее об этих полях можно узнать здесь.
field_name	string	custom_ID	Да, если “matching”: “custom”	Имя кастомного поля БД Altcraft при **matching : custom** . В форме сбора данных для поиска нужно поле с таким же именем.
field_value	int/string	custom_string	Да, если “matching”: “custom”	Значение кастомного поля при **matching : custom**.
subscriptions	bool	true	Нет	Копировать и разделить подписки. Подписки на категории ресурса и данные из полей подписки также скопируются. (по умолчанию – true)
history	bool	true	Нет	Копировать историю и флаги активности (по умолчанию – true)
static_segments	bool	true	Нет	Вносить ли изменения в статические сегменты (source-профили выйдут из сегментов, а destination-профиль - войдет) (по умолчанию – false)

Параметр	Тип	Пример	Обязательный	Описание
subscription_filters	JSON объект	“subscription_filters”: [{ “resource_id”: 1, “channel”: “push”, “provider”: “Safari” }]	Нет	Фильтр - для копирования конкретных подписок. Если не указывать - копируются все.
overwrite	bool	true	нет	Копировать ли поля source-профиля (по умолчанию – false) Если у destination-профилей есть пустые поля, они в любом случае будут заполнены полями source-профиля.
delete	bool	true	нет	Удалить ли source-профиль после разделения (по умолчанию – false)

Копирование полей профиля

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "matching": "custom",
    "field_name": "field_1",
    "source": {
        "db_id": 1,
        "field_value": 1,
        "data": {
            "_fname": "Profile1",
            "_lname": "LnameProfile1"
        }
    },
    "destination": [
        {
            "db_id": 2,
            "matching": "email",
            "email": "test2@mail.ru",
            "data": {
                "_fname": "Profile2",
                "_lname": "LnameProfile2"
            },
        },
        {
            "db_id": 2,
            "field_value": 3,
            "data": {
                "_fname": "Profile3",
                "_lname": "Lname3"
            },
            "subscription_filters": [
                {
                    "resource_id": 1,
                    "channel": "push",
                    "provider": "Safari"
                },
                {
                    "resource_id": 2,
                    "channel": "email",
                }
            ]
        }
    ],
    "subscriptions": true,
    "static_segments": true,
    "history": false
}
```

XML

```
<xml>
 <token>abcdefghijklmnqrstuvwxyz</token>
    <matching>custom</matching>
    <field_name>field_1</field_name>
    <delete>true</delete>
    <source>
        <db_id>1</db_id>
        <field_value>1</field_value>
        <data>
            <_fname>Profile1</_fname>
            <_lname>LnameProfile1</_lname>
        </data>
    </source>
    <destination>
        <db_id>2</db_id>
        <matching>email</matching>
        <email>test2@mail.ru</email>
        <data>
            <_fname>Profile2</_fname>
            <_lname>LnameProfile2</_lname>
        </data>
    </destination>
    <destination>
        <db_id>2</db_id>
        <field_value>3</field_value>
        <data>
            <_fname>Profile3</_fname>
            <_lname>LnameProfile3</_lname>
        </data>
        <subscription_filters>
            <resource_id>1</resource_id>
            <channel>push</channel>
            <provider>Safari</provider>
        </subscription_filters>
        <subscription_filters>
            <resource_id>2</resource_id>
            <channel>email</channel>
        </subscription_filters>
    </destination>
    <subscriptions>true</subscriptions>
    <static_segments>true</static_segments>
    <history>false</history>
</xml>
```

Пример ответа

JSON

```
{
    "destination_ids": [
        {
            "db_id": 2,
            "profile_id": "60a7b150fa3adb6f1179fe7f"
        },
        {
            "db_id": 2,
            "profile_id": "60a7b150fa3adb6f1179fe80"
        }
    ],
    "error": 0,
    "error_text": "",
    "source_id": {
        "db_id": 1,
        "profile_id": "60a7a45bfa3adb7132dcceaa"
    }
}
```

XML

```
<xml>
    <destination_ids>
        <db_id>1</db_id>
        <profile_id>66432f6cee44bb50a6711634</profile_id>
    </destination_ids>
    <destination_ids>
        <db_id>1</db_id>
        <profile_id>6613cb65dc4594c2934db63f</profile_id>
    </destination_ids>
    <error>0</error>
    <source_id>
        <db_id>1</db_id>
        <profile_id>6643405b41c3a674897dd9cc</profile_id>
    </source_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
destination_ids	JSON объект	Содержит данные о профилях, получившихся в результате разделения: `db_id` — идентификатор базы данных `profile_id` — идентификатор профиля

Параметр	Тип	Описание
source_id	JSON объект	Содержит данные о разделенном профиле в виде списка: `db_id` — идентификатор базы данных `profile_id` — идентификатор профиля
error	int	Код ошибки
error_text	string	Текст ошибки

Получить историю одного профиля

Описание

Получить историю действий одного профиля в сообщениях.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscribers/history_get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
list_id	Int	1	Да	Идентификатор базы данных
xxh	string	“0eb51aefd919f90a”	Нет, если есть matching	xxhash идентификатор профиля
matching	string	“email” “email_profile” “email_sub” “phone” “phone_sub” “profile_id” “push_sub” “custom” “custom_sub” “email_phone” “email_phone_sub”	Нет, если поиск по xxh	Режим поиска подписчика. По умолчанию – email. Для каждого типа матчинга в теле запроса необходимо передавать определенные поля. Подробнее об этих полях можно узнать здесь.
date_from	string	“2017-01-01”	Нет	Начало отчетного периода xxh.YYYYMMDD
date_to	string	“2017-12-31”	Нет	Конец отчетного периода xxh.YYYYMMDD

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "list_id": 20,
  "xxh": "0eb51aefd919f90a"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <list_id>20</list_id>
  <xxh>0eb51aefd919f90a</xxh>
</xml>
```

Пример ответа

JSON

```
{
"data": [
    {
        "xxh": "9202595093f237d4",
        "profile_id": "66f66973096b3b454bbbccec",
        "email": "example@example.com",
        "list_id": 66,
        "action": "send",
        "count": 1,
        "datetime": "2024-09-27T15:52:57.004Z",
        "message_id": 5,
        "campaign_id": 137,
        "event_id": "66f6aaa9da00a9db1c0b2683",
        "smid": "uBAhiRBRQhARg7y489hr5F2b2aXMJaztFS7vM7DE1NGJhZWM5",
        "campaign_name": "New astral campaign",
        "message_name": "New template",
        "subscription": {
            "resource_id": 5,
            "resource_name": "res1",
            "channel": "email",
            "email": "sub1@yandex.ru",
            "priority": 0,
            "status": "subscribed",
            "custom_fields": {
                "_device_type": "other"
            },
            "cats": []
        },
        "channel": "email",
        "subscriptions": [
            {
                "resource_id": 5,
                "resource_name": "res1",
                "channel": "email",
                "email": "sub1@yandex.ru",
                "priority": 0,
                "status": "subscribed",
                "custom_fields": {
                    "_device_type": "other"
                },
                "cats": []
            },
            {
                "resource_id": 5,
                "resource_name": "res1",
                "channel": "email",
                "email": "sub2@yandex.ru",
                "priority": 0,
                "status": "subscribed"
```

```
                "status": "subscribed",
                "custom_fields": {
                    "_device_type": "mob"
                },
                "cats": []
            }
        ]
    }
    ],
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

<xml>
  <data>
        <item>
            <xxh>9202595093f237d4</xxh>
            <profile_id>66f66973096b3b454bbbccec</profile_id>
            <email>example@example.ru</email>
            <list_id>66</list_id>
            <action>send</action>
            <count>1</count>
            <datetime>2024-09-27T15:52:57.004Z</datetime>
            <message_id>5</message_id>
            <campaign_id>137</campaign_id>
            <event_id>66f6aaa9da00a9db1c0b2683</event_id>
            <smid>uBAhiRBRQhARg7y489hr5F2b2aXMJaztFS7vM7DE1NGJhZWM5</smid>
            <campaign_name>New astral campaign</campaign_name>
            <message_name>New template</message_name>
            <subscription>
                <resource_id>5</resource_id>
                <resource_name>res1</resource_name>
                <channel>email</channel>
                <email>sub1@yandex.ru</email>
                <priority>0</priority>
                <status>subscribed</status>
                <custom_fields>
                    <_device_type>other</_device_type>
                </custom_fields>
            </subscription>
            <channel>email</channel>
            <subscriptions>
                <subscription>
                    <resource_id>5</resource_id>
                    <resource_name>res1</resource_name>
                    <channel>email</channel>
                    <email>sub1@yandex.ru</email>
                    <priority>0</priority>
                    <status>subscribed</status>
                    <custom_fields>
                        <_device_type>other</_device_type>
                    </custom_fields>
                </subscription>
                <subscription>
                    <resource_id>5</resource_id>
                    <resource_name>res1</resource_name>
                    <channel>email</channel>
                    <email>sub2@yandex.ru</email>
                    <priority>0</priority>
                    <status>subscribed</status>
                    <custom_fields>
                        <_device_type>mob</_device_type>

```
                    <_device_type>mob</_device_type>
                </custom_fields>
            </subscription>
        </subscriptions>
    </item>
</data>
<error>0</error>
<error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	string	Массив полученных данных по событиям в истории профиля
data.xxh	string	xxhash идентификатор профиля
data.profile_id	string	Основной идентификатор профиля
data.email	string	Email-адрес профиля
data.list_id (db_id)	int	Идентификатор базы данных
data.action	string	Действие подписчика (Подробнее)
data.count	int	Количество действий, совершенных в эту секунду
data.datetime	string	Дата и время действия в формате RFC 3339 (ISO 8601)
data.message_id	int	Идентификатор шаблона сообщения Возвращает 0, если действие не предполагает использование шаблона.
data.campaign_id	int	Идентификатор рассылки Возвращает 0, если действие не предполагает использование рассылки.
data.event_id	string	Идентификатор события
data.smid	string	Уникальной идентификатор отправки в рамках рассылки Возвращает пустую строку, если действие не предполагает отправку сообщения.
data.campaign_name	string	Название рассылки Возвращает пустую строку, если действие не предполагает использование рассылки.
data.message_name	string	Название шаблона сообщения Возвращает пустую строку, если действие не предполагает использование шаблона.
data.subscription	string	Объект с информацией о подписке, в рамках которой выполнялась отправка

Параметр	Тип	Описание
data.channel	string	Канал коммуникации, в рамках которого зарегистрировано событие. Возвращает пустую строку, если канал не задействован.
data.subscriptions	array of objects	Все подписки профиля. Для формата CSV будет представлены в виде JSON строки.
error	int	Код ошибки
error_text	string	Текст ошибки

События в истории профиля (action)

События Email канала	
send	Email отправлен
deliv	Email доставлен
undeliv	Email не доставлен
open	Email открыт
read	Email прочитан. Сообщение открыто пользователем в течении 8 и более секунд
click	Клик в Email. Событие регистрируется при переходе по ссылке в письме. Для каждой ссылки регистрируется свое событие клика, т.е. если профиль перейдет по двум разным ссылкам в письме, будут зарегистрированы 2 события клика.
confirm	Переход по ссылке подтверждения подписки в Email письме (confirm-link)
subscribe_email	Подписан на Email канал ресурса
unsubscribe_email	Отписан от Email канала ресурса
reply	Получен ответ на Email-письмо
complain	Получен статус Жалобщик
hbounce	Получен статус Hardbounced
События SMS канала	
send_sms	SMS сообщение отправлено в шлюз для доставки подписчику
deliv_sms	SMS сообщение доставлено
undeliv_sms	SMS сообщение не доставлено
click_sms	Неуникальный клик по ссылке в SMS сообщении
subscribe_sms	Подписан на SMS канал ресурса

unsubscribe_sms	Отписан от SMS канала ресурса
События Push канала	
send_push	Push отправлен
deliv_push	Push доставлен
open_push	Неуникальное открытие push сообщения
click_push	Клик в Push
undeliv_push	Push сообщение не доставлено
subscribe_push	Подписан на Push канал ресурса
unsubscribe_push	Отписан от Push канала ресурса
События Telegram Bot канала	
telegram_bot_send	В канале Telegram Bot произошло событие Telegram bot отправка
telegram_bot_deliv	В канале Telegram Bot произошло событие Telegram bot доставка
telegram_bot_undeliv	В канале Telegram Bot произошло событие Telegram bot недоставка
telegram_bot_click	В канале Telegram Bot произошло событие Telegram bot клик
telegram_bot_subscribe	В канале Telegram Bot произошло событие Telegram bot подписка
События WhatsApp*-канала	
whatsapp_send	Сообщение отправлено для доставки подписчику в whatsapp чат
whatsapp_deliv	Сообщение доставлено в whatsapp чат
whatsapp_undeliv	Сообщение не доставлено в whatsapp чат
whatsapp_click	Неуникальный клик по ссылке в сообщении в whatsapp чате
whatsapp_read	Сообщение прочитано пользователем
whatsapp_subscribe	В канале whatsapp произошло событие Whatsapp подписка

События Viber канала	
viber_send	Сообщение отправлено для доставки подписчику в Viber чат
viber_deliv	Сообщение доставлено получателю
viber_undeliv	Сообщение не доставлено, так как отклонено Viber или Devino.Online
viber_click	Переход по ссылке в сообщении. Если получатель несколько раз кликнул по одной и той же ссылке, фиксируется каждое событие клика.
viber_read	Сообщение прочитано пользователем
viber_subscribe	В канале Viber произошло событие Viber подписка
События для всех каналов	
offence	Сообщение не отправлено из-за ограничений политики отправки
suppress	Сообщение не было отправлено рассылкой, так как профиль находится в стоп-списке
События импорта профилей подписчиков	
import_manual	Профиль создан вручную
import_api	Профиль создан API-импортом
import_file	Профиль создан импортом из файла
import_form	Профиль создан импортом из формы
import_push	Профиль создан в результате push-импорта
import_popup	Профиль создан импортом через попап
События отписки	
unsub_api	Подписчик отписан от рассылок через API
unsub_manual	Отписан вручную (глобальный статус профиля изменен на "Отписан")
События пикселя	
pixel_open	Достигнута цель
События, связанные с промокодами	
promocode_attach	Привязан промокод
promocode_detach	Отвязан промокод

promocode_activate	Активирован промокод
События сегментов	
segs_add	Вошёл в статический сегмент
segs_remove	Вышел из статического сегмента
События связей между профилями	
rel_attach	Профиль получает связь с другим профилей
rel_detach	Профиль теряет связь с другим профилем
rel_strengthen	Увеличивается вес связи между профилями
События формы	
form_load	Форма загружена
form_page_show	Страница формы загружена
form_post	Форма заполнена и отправлена
form_abandon	Форма брошена. Профиль загружает страницу с формой и не закрывает её в течение часа, при этом не отправляя форму
form_bounce	Ошибка при заполнении формы (попытка повторного заполнения, технические проблемы и др.)

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Получить историю нескольких профилей

Описание

Получить историю нескольких профилей.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscribers/history_get_all`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
list_id	int	20	Да	Идентификатор базы данных
from_id	string	“0014a2feb331ace2.20200124134127”	Нет	Указатель на следующую страницу, берется из предыдущего запроса или передается пустой для первого.
limit	int	500	Нет	Лимит количества записей (по умолчанию – 1000)
date_from	string	“2016-11-25”	Да	Начало периода YYYY-MM-DD
date_to	string	“2016-11-29”	Да	Конец периода YYYY-MM-DD
fields	Array of string	[“CustomID”]	Нет	Добавит к результату поле fields с данными профиля из перечисленных полей. Для формата CSV будет представлен в виде JSON строки.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "list_id": 4,
    "limit": 3,
    "date_from": "2024-07-15",
    "date_to": "2024-10-08",
    "format": "json",
    "from_id": "1",
    "fields": [
        "CustomNumber",
        "CustomEnum"
    ]
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <list_id>1</list_id>
  <limit>3</limit>
  <date_from>2024-07-15</date_from>
  <date_to>2024-10-09</date_to>
  <format>xml</format>
  <from_id>1</from_id>
  <fields>CustomNumber</fields>
  <fields>CustomEnum</fields>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "xxh": "b8cdef2673dd18ff",
            "profile_id": "669e6966ebf4ac52b6239a9c",
            "email": "heatherhawkins@tambee.name",
            "list_id": 6,
            "action": "subscribe_push",
            "count": 1,
            "datetime": "2024-07-22T14:15:02.838Z",
            "message_id": 0,
            "campaign_id": 0,
            "event_id": "669e6966ebf4ac52b6239aa4",
            "smid": "",
            "campaign_name": "",
            "message_name": "",
            "subscription": {
                "resource_id": 3,
                "resource_name": "Push res",
                "channel": "push",
                "subscription_id": "2",
                "provider": "Chrome",
                "priority": 0,
                "status": "subscribed",
                "custom_fields": {},
                "cats": []
            },
            "channel": "push",
            "subscriptions": [
                {
                    "resource_id": 3,
                    "resource_name": "Push res",
                    "channel": "push",
                    "subscription_id": "2",
                    "provider": "Chrome",
                    "priority": 0,
                    "status": "subscribed",
                    "custom_fields": {},
                    "cats": []
                }
            ]
        },
        {
            "xxh": "2d15ffc902593f01",
            "profile_id": "669e6966ebf4ac52b6239a9e",
            "email": "howardowens@jaxspan.org",
            "list_id": 6,
            "action": "subscribe_push",
            "count": 1
```

```
            "count": 1,
            "datetime": "2024-07-22T14:15:02.839Z",
            "message_id": 0,
            "campaign_id": 0,
            "event_id": "669e6966ebf4ac52b6239aa5",
            "smid": "",
            "campaign_name": "",
            "message_name": "",
            "subscription": {
                "resource_id": 3,
                "resource_name": "Push res",
                "channel": "push",
                "subscription_id": "0",
                "provider": "Chrome",
                "priority": 0,
                "status": "subscribed",
                "custom_fields": {},
                "cats": []
            },
            "channel": "push",
            "subscriptions": [
                {
                    "resource_id": 3,
                    "resource_name": "Push res",
                    "channel": "push",
                    "subscription_id": "0",
                    "provider": "Chrome",
                    "priority": 0,
                    "status": "subscribed",
                    "custom_fields": {},
                    "cats": []
                }
            ]
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "next_from_id": "2"
}
```


XML

<xml>
  <data>
            <xxh>b8cdef2673dd18ff</xxh>
            <profile_id>669e6966ebf4ac52b6239a9c</profile_id>
            <email>heatherhawkins@tambee.name</email>
            <list_id>6</list_id>
            <action>subscribe_push</action>
            <count>1</count>
            <datetime>2024-07-22T14:15:02.838Z</datetime>
            <message_id>0</message_id>
            <campaign_id>0</campaign_id>
            <event_id>669e6966ebf4ac52b6239aa4</event_id>
            <smid></smid>
            <campaign_name></campaign_name>
            <message_name></message_name>
            <subscription>
                <resource_id>3</resource_id>
                <resource_name>Push res</resource_name>
                <channel>push</channel>
                <subscription_id>2</subscription_id>
                <provider>Chrome</provider>
                <priority>0</priority>
                <status>subscribed</status>
            </subscription>
            <channel>push</channel>
            <subscriptions>
                <subscription>
                    <resource_id>3</resource_id>
                    <resource_name>Push res</resource_name>
                    <channel>push</channel>
                    <subscription_id>2</subscription_id>
                    <provider>Chrome</provider>
                    <priority>0</priority>
                    <status>subscribed</status>
                </subscription>
            </subscriptions>
        </item>
        <item>
            <xxh>2d15ffc902593f01</xxh>
            <profile_id>669e6966ebf4ac52b6239a9e</profile_id>
            <email>howardowens@jaxspan.org</email>
            <list_id>6</list_id>
            <action>subscribe_push</action>
            <count>1</count>
            <datetime>2024-07-22T14:15:02.839Z</datetime>
            <message_id>0</message_id>
            <campaign_id>0</campaign_id>
            <event_id>669e6966ebf4ac52b6239aa5</event_id>
            <smid></smid>

```
            <smid></smid>
            <campaign_name></campaign_name>
            <message_name></message_name>
            <subscription>
                <resource_id>3</resource_id>
                <resource_name>Push res</resource_name>
                <channel>push</channel>
                <subscription_id>0</subscription_id>
                <provider>Chrome</provider>
                <priority>0</priority>
                <status>subscribed</status>
            </subscription>
            <channel>push</channel>
            <subscriptions>
                <subscription>
                    <resource_id>3</resource_id>
                    <resource_name>Push res</resource_name>
                    <channel>push</channel>
                    <subscription_id>0</subscription_id>
                    <provider>Chrome</provider>
                    <priority>0</priority>
                    <status>subscribed</status>
                </subscription>
            </subscriptions>
        </item>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <next_from_id>2</next_from_id>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
data	string	Массив полученных данных по событиям в истории профилей
data.xxh	string	xxhash идентификатор профиля
data.profile_id	string	Основной идентификатор профиля
data.email	string	Email-адрес профиля
data.list_id (db_id)	int	Идентификатор базы данных
data.action	string	Действие подписчика (Подробнее)
data.count	int	Количество действий, совершенных в эту секунду
data.datetime	string	Дата и время действия в формате RFC 3339 (ISO 8601)

Параметр	Тип	Описание
data.message_id	int	Идентификатор шаблона сообщения Возвращает 0, если действие не предполагает использование шаблона.
data.campaign_id	int	Идентификатор рассылки Возвращает 0, если действие не предполагает использование рассылки.
data.event_id	string	Идентификатор события
data.smid	string	Уникальной идентификатор отправки в рамках рассылки Возвращает пустую строку, если действие не предполагает отправку сообщения.
data.campaign_name	string	Название рассылки Возвращает пустую строку, если действие не предполагает использование рассылки.
data.message_name	string	Название шаблона сообщения Возвращает пустую строку, если действие не предполагает использование шаблона.
data.subscription	string	Объект с информацией о подписке, в рамках которой выполнялась отправка
data.channel	string	Канал коммуникации, в рамках которого зарегистрировано событие. Возвращает пустую строку, если канал не задействован.
data.subscriptions	array of objects	Все подписки профиля. Для формата CSV будет представлены в виде JSON строки.
data.fields	object	Поля с данными профиля, запрошенные в поле `fields`. Для формата CSV будут представлены в виде JSON строки.
next_from_id	string	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории.
error	int	Код ошибки
error_text	string	Текст ошибки

Добавить связь

Описание

Создаёт связь между двумя профилями. При наличии идентичной связи между профилями запрос завершится ошибкой дубликации.

URL-адрес

Метод: POST.

`https://example.com/api/v1.1/relations/attach`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
rel_id	int	3	Да	ID связи
profile_a	object	true	Да	Профиль базы данных A, **к которому** направлена связь
profile_b	object	true	Да	Профиль базы данных B, **от которого** направлена связь
weights	object	`{"user_like":2,"donate":343,"is_online":true}`	Да	Свойства связи профилей
Поиск профиля подписчика				
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля

field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. **TIP** Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "rel_id": 3,
    "profile_a": {
        "profile_id": "5d63dbdb439e7306b596c5d0"
    },
    "profile_b": {
        "profile_id": "5d441038439e7363f3c26539"
    },
    "weights": {
        "user_like": 2,
        "donate": 343,
        "is_online": true
    }
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <rel_id>3</rel_id>
    <profile_a>
        <profile_id>5d63dbdb439e7306b596c5d0</profile_id>
    </profile_a>
    <profile_b>
        <profile_id>5d441038439e7363f3c26539</profile_id>
    </profile_b>
    <weights>
        <user_like>2</user_like>
        <donate>343</donate>
        <is_online>true</is_online>
    </weights>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "",
    "profile_a": "5c910f6bc484d023b1309bee",
    "profile_b": "5c910938c484d023b1309bac",
    "relation_id": 1
}
```

XML

```
<xml>
  <error>0</error>
  <error_text></error_text>
  <profile_a>5c910f6bc484d023b1309bee</profile_a>
  <profile_b>5c910938c484d023b1309bac</profile_b>
  <relation_id>1</relation_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки

Параметр	Тип	Описание
error_text	string	Текст ошибки
profile_a	string	Идентификатор профиля A
profile_b	string	Идентификатор профиля B
relation_id	1	Идентификатор связи

Удалить связь

Описание

Удаляет связь между двумя профилями.

URL-адрес

Метод: POST.

`https://example.com/api/v1.1/relations/detach`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
rel_id	int	3	Да	ID связи
profile_a	object	true	Да	Профиль базы данных A, **к которому** направлена связь
profile_b	object	true	Да	Профиль базы данных B, **от которого** направлена связь
Поиск профиля подписчика				
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. :::tip Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2". :::

resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "rel_id": 3,
    "profile_a": {
        "profile_id": "5d63dbdb439e7306b596c5d0"
    },
    "profile_b": {
        "profile_id": "5d441038439e7363f3c26539"
    }
}
```

XML

```
<xml>
<token>abcdefghijklmnqrstuvwxyz</token>
  <rel_id>3</rel_id>
  <profile_a>
    <profile_id>5d63dbdb439e7306b596c5d0</profile_id>
  </profile_a>
  <profile_b>
    <profile_id>5d441038439e7363f3c26539</profile_id>
  </profile_b>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "",
    "profile_a": "5c910f6bc484d023b1309bee",
    "profile_b": "5c910938c484d023b1309bac",
    "relation_id": 1
}
```

XML

```
<xml>
    <error>0</error>
    <error_text></error_text>
    <profile_a>5c910f6bc484d023b1309bee</profile_a>
    <profile_b>5c910938c484d023b1309bac</profile_b>
    <relation_id>1</relation_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_a	string	Идентификатор профиля A
profile_b	string	Идентификатор профиля B
relation_id	1	Идентификатор связи

Усилить/ослабить связь

Описание

Изменяет (прибавляет или убавляет) значения свойств связи на числа, указанные в запросе. Если связи не найдено, создаётся новая связь между профилями.

URL-адрес

Метод: POST.

`https://example.com/api/v1.1/relations/strengthen`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
rel_id	int	3	Да	ID связи
profile_a	object	true	Да	Профиль базы данных A, **к которому** направлена связь
profile_b	object	true	Да	Профиль базы данных B, **от которого** направлена связь
weights	object	`{"user_like":2,"donate":343,"is_online":true}`	Да	Свойства связи профилей. Если необходимо ослабить свойство, то передайте отрицательное значение: -2 или -343
Поиск профиля подписчика				
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.

field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. :::tip Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2". :::
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "rel_id": 3,
    "profile_a": {
        "profile_id": "5d63dbdb439e7306b596c5d0"
    },
    "profile_b": {
        "profile_id": "5d441038439e7363f3c26539"
    },
    "weights": {
        "user_like": 2,
        "donate": 343,
        "is_online": true
    }
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <rel_id>3</rel_id>
  <profile_a>
    <profile_id>5d63dbdb439e7306b596c5d0</profile_id>
  </profile_a>
  <profile_b>
    <profile_id>5d441038439e7363f3c26539</profile_id>
  </profile_b>
  <weights>
    <user_like>2</user_like>
    <donate>343</donate>
    <is_online>true</is_online>
  </weights>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "",
    "profile_a": "5c910f6bc484d023b1309bee",
    "profile_b": "5c910938c484d023b1309bac",
    "relation_id": 1
}
```

XML

```
<xml>
  <error>0</error>
  <error_text></error_text>
  <profile_a>5c910f6bc484d023b1309bee</profile_a>
  <profile_b>5c910938c484d023b1309bac</profile_b>
  <relation_id>1</relation_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_a	string	Идентификатор профиля A
profile_b	string	Идентификатор профиля B
relation_id	1	Идентификатор связи

Перезаписать значения свойств связи

Описание

Изменяет значения свойств связи на указанные в запросе.

URL-адрес

Метод: POST.

`https://example.com/api/v1.1/relations/edit`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
rel_id	int	3	Да	ID связи
profile_a	object	true	Да	Профиль базы данных A, **к которому** направлена связь
profile_b	object	true	Да	Профиль базы данных B, **от которого** направлена связь
weights	object	`{"user_like":2,"donate":343,"is_online":true}`	Да	Свойства связи профилей.
Поиск профиля подписчика				
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.

field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. :::tip Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2". :::
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "rel_id": 3,
    "profile_a": {
        "profile_id": "5d63dbdb439e7306b596c5d0"
    },
    "profile_b": {
        "profile_id": "5d441038439e7363f3c26539"
    },
    "weights": {
        "user_like": 2,
        "donate": 343,
        "is_online": true
    }
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <rel_id>3</rel_id>
  <profile_a>
    <profile_id>5d63dbdb439e7306b596c5d0</profile_id>
  </profile_a>
  <profile_b>
    <profile_id>5d441038439e7363f3c26539</profile_id>
  </profile_b>
  <weights>
    <user_like>2</user_like>
    <donate>343</donate>
    <is_online>true</is_online>
  </weights>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "",
    "profile_a": "5c910f6bc484d023b1309bee",
    "profile_b": "5c910938c484d023b1309bac",
    "relation_id": 1
}
```

XML

```
<xml>
<error>0</error>
  <error_text></error_text>
  <profile_a>5c910f6bc484d023b1309bee</profile_a>
  <profile_b>5c910938c484d023b1309bac</profile_b>
  <relation_id>1</relation_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_a	string	Идентификатор профиля A
profile_b	string	Идентификатор профиля B
relation_id	1	Идентификатор связи

Получить информацию о связях профиля

Описание

Запрос отдаёт информацию о связях профиля из базы данных А, свойствах связей и значениях свойств.

URL-адрес

Метод: POST.

`https://example.com/api/v1.1/relations/get_counter`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
rel_id	int	3	Да	ID связи
profile_id		"abcdefghijklmnqrstuvwxyz"	Да	Профиль из базы данных А для поиска
Поиск профиля подписчика				
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. :::tip Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2". :::

resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "rel_id": 1,
    "profile_id": "67ef9942dc993311c9a067f5"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <rel_id>1</rel_id>
  <profile_id>67ef9942dc993311c9a067f5</profile_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "profile_id": "67ef9942dc993311c9a067f5",
        "profile_xxh": "2e6325012644fdf9",
        "props": {
            "donate": {
                "top": [
                    {
                        "profile_id": "67dd005523ea81dd52f98195",
                        "value": 1,
                        "email_b": "67dd005523ea81dd52f98195@localhost"
                    }
                ],
                "sum": 344
            },
            "user_like": {
                "top": [],
                "sum": 3
            }
        }
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
    <data>
        <profile_id>67ef9942dc993311c9a067f5</profile_id>
        <profile_xxh>2e6325012644fdf9</profile_xxh>
        <props>
            <donate>
                <sum>344</sum>
                <top>
                    <email_b>67dd005523ea81dd52f98195@localhost</email_b>
                    <profile_id>67dd005523ea81dd52f98195</profile_id>
                    <value>1</value>
                </top>
            </donate>
            <user_like>
                <sum>3</sum>
            </user_like>
        </props>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
data	object	Информация о связях профиля
data.profile_id	string	Идентификатор профиля
data.profile_xxh	string	Хешированый идентификатор профиля
data.props	object	Свойства связей

Свойства связей: числовые

Параметр	Тип	Описание
top	object	Топ свойства (связанные профили и значения)
sum	int/float	Сумма числового свойства

Получить список связей профиля

Описание

Запрос возвращает список всех существующих связей для указанного профиля из базы данных A, включая идентификаторы связанных профилей, типы связей и их свойства.

URL-адрес

Метод: POST.

`https://example.com/api/v1.1/relations/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
rel_id	int	3	Да	ID связи
profile_id	string	"669e6968ebf4ac52b623a997"	Да, если не указан другой тип поиска	Идентификатор профиля из базы данных A для поиска
Поиск профиля подписчика				
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+7900000000 00"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. :::tip Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2". :::

resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "rel_id": 2,
    "profile_id": "669e6968ebf4ac52b623a997"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <rel_id>2</rel_id>
  <profile_id>669e6968ebf4ac52b623a997</profile_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "profile_id_a": "669e6968ebf4ac52b623a997",
            "xxh_a": "5af1023c84f66195",
            "profile_id_b": "6751b8f1c5e3db9bcb75befa",
            "xxh_b": "879cebbb0e6b6723",
            "db_b": 9,
            "props": {
                "exr": {
                    "value": 11323123
                }
            }
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 1
}
```


XML


```
<xml>
    <data>
        <profile_id_a>abcdefghijklmnqrstuvwxyz</profile_id_a>
            <xxh_a>5af1023c84f66195</xxh_a>
            <profile_id_b>6751b8f1c5e3db9bcb75befa</profile_id_b>
            <xxh_b>879cebbb0e6b6723</xxh_b>
            <db_b>9</db_b>
            <props>
                <exr>
                    <value>11323123</value>
                </exr>
            </props>
        </relation>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>1</total_count>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
data	object	Информация о связях профиля
data.profile_id_a	string	Идентификатор профиля А (профиль, к которому направлена связь)
data.xxh_a	string	Хешированый идентификатор профиля А
data.profile_id_b	string	Идентификатор профиля Б (профиль, инициирующий связь)
data.xxh_b	string	Хешированый идентификатор профиля Б
data.db_b	int	Идентификатор базы, в которой находится профиль Б
data.props	object	Свойства связей
data.total_count	int	Количество связей с другими профилями

Добавить или редактировать подписку

Описание

Добавить или редактировать подписку (изменить статус, подписать или отписать).

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscriptions/set`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных (по умолчанию - "json") </td>
db_id	int	1	Да	Идентификатор базы данных
profile_id	string	"abcdefghijklmnqrstuvwxyz"	Да	Идентификатор профиля
resource_id	int	1	Да	Идентификатор ресурса
status	string	"subscribed"	Нет	Статус подписки. Рекомендуется передавать только в тех случаях, когда необходимо обновить статус подписки. </td>
priority	int		Нет	Приоритет подписки

custom_fields	object	{ "_browser_name": "Chrome", "_device_type": "web" }	Нет	Стандартные и дополнительные поля подписки. Получить информацию о доступных полях подписки в ресурсе можно с помощью метода [/v1.1/resources/fields_get](/developer-guide/resources/2195591) </td>
cats	array	["category_1", "category_2"]	Нет	Категории ресурса, на которые подписать профиль
date	string	2018-08-22T00:00:00.000Z	Нет	Дата регистрации (в формате RFC3339)
ip	string	10.250.13.5	Нет	IP регистрации
url	string	https://example.com/	Нет	URL регистрации
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false) </td>

Дополнительно для Email-канала:

Параметры	Тип	Пример	Описание
channel	string	“email”	Тип канала
email	string	“example@example.com”	Email-адрес

Дополнительно для PUSH-канала:

Параметры	Тип	Пример	Описание
channel	string	“push”	Тип канала
provider	string	“Firefox”	Тип провайдера
subscription_id	string	“abcdefghijklmnqrstuvwxyz”	Идентификатор подписки

Дополнительно для SMS-канала:

Параметры	Тип	Пример	Описание
channel	string	“sms”	Тип канала
phone	string	“+79000000000”	Номер телефона

Дополнительно для Telegram-канала:

Параметры	Тип	Пример	Описание
channel	string	“telegram_bot”	Тип канала

cc_data	object	{
		“id”: 1
		}
			id чата в Telegram-бот

Дополнительно для WhatsApp*-канала:

Параметры	Тип	Пример	Описание
channel	string	“whatsapp”	Тип канала

cc_data	object	{
		“phone”: “+79000000000”
		}
			Телефон профиля в WhatsApp*

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "db_id": 1,
  "resource_id": 1,
  "channel": "email",
  "email": "example@example.com",
  "priority": 10,
  "profile_id": "abcdefghijklmnqrstuvwxyz",
  "status": "suspended"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
  <resource_id>29</resource_id>
  <channel>email</channel>
  <email>example@example.com</email>
  <priority>10</priority>
  <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
  <status>suspended</status>
</xml>
```

CAUTION

Обратите внимание, опция “Повышать приоритет для новых подписок” в ресурсе работает только в случае, если при добавлении новой подписки явно не указан приоритет. Если приоритет новой подписки передается в API-запросе, то приоритеты существующих подписок профиля не изменятся.

Пример ответа

JSON

```
{
  "data": {
    "action": "subscribed",
    "is_new_channel": false,
    "is_new_contact": false,
    "is_new_subscription": true,
    "is_restored": false,
    "is_safe": true
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <action>subscribed</action>
    <is_new_channel>false</is_new_channel>
    <is_new_contact>false</is_new_contact>
    <is_new_subscription>true</is_new_subscription>
    <is_restored>false</is_restored>
    <is_safe>true</is_safe>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
action	string	Результат изменения подписки
is_safe	bool	Операция была безопасной (без hardbounce и т.д.)
is_new_subscription	bool	Не был подписан на этот ресурс ранее
is_new_channel	bool	Уже подписан на этот ресурс, но на новый канал
is_new_contact	bool	Новый контакт уже подписан на ресурс или канал
is_restored	bool	Восстановлена ранее удаленная подписка

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Получить все подписки профиля

Описание

Получить все подписки одного профиля.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscriptions/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
profile_id	string	“abcdefghijklmnqrstuvwxyz”	Да	Идентификатор профиля

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "profile_id": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "resource_id": 29,
            "channel": "email",
            "email": "example@example.com",
            "priority": 10,
            "status": "suspended",
            "custom_fields": {
                "custom_int_field_2": 20000
            }
        }
    ],
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <channel>email</channel>
        <custom_fields>
            <custom_int_field_2>20000</custom_int_field_2>
        </custom_fields>
        <email>example@example.com</email>
        <priority>10</priority>
        <resource_id>29</resource_id>
        <status>suspended</status>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
channel	string	Тип канала (email, push или sms)

Параметр	Тип	Описание
priority	int	Приоритет подписки (меньше число – больше приоритет)
resource_id	int	Идентификатор ресурса, связанного с подпиской
status	string	Статус подписки
custom_fields	object	Стандартные и дополнительные поля подписки. Получить информацию о полях подписки в ресурсе можно с помощью метода /v1.1/resources/fields_get
cats	array	Категории ресурса, на которые подписан профиль

Дополнительно для Email-канала:

Параметры	Тип	Описание
email	string	Email-адрес

Дополнительно для PUSH-канала:

Параметры	Тип	Описание
provider	string	Тип провайдера
subscription_id	string	Идентификатор подписки

Дополнительно для SMS-канала:

Параметры	Тип	Описание
phone	string	Номер телефона

Дополнительно для Telegram-канала:

Параметры	Тип	Описание
cc_data	object	id чата в Telegram-бот

Дополнительно для WhatsApp*-канала:

Параметры	Тип	Описание
cc_data	object	Телефон профиля в WhatsApp*

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Получить все подписки нескольких профилей

Описание

Получить все подписки нескольких профилей.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscriptions/list_all`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
resource_id	int	1	Да	Идентификатор ресурса
db_id	int	1	Нет	Идентификатор базы данных
channel	string	“email”	Нет	Тип канала
status	string	“subscribed”	Нет	Статус подписки
from_id	string	“60df2b6325bf0aa04e22e8f5”	Нет	Идентификатор первого профиля в списке
limit	int	1	Нет	Лимит на количество возвращаемых данных

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "resource_id": 29,
    "from_id": "abcdefghijklmnqrstuvwxyz",
    "limit": 1
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
  <resource_id>1</resource_id>
  <from_id>abcdefghijklmnqrstuvwxyz</from_id>
  <limit>1</limit>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "db_id": 1,
            "profile_id": "6643052fee44bb50a671161d",
            "_xxh": "026e82a364dd17e6",
            "hash_id": "fcf06ea1",
            "resource_id": 29,
            "channel": "email",
            "email": "example@example.com",
            "priority": 0,
            "status": "subscribed",
            "custom_fields": {
                "custom_int_field_2": 20000
            }
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "next_from_id": "cPydNpzRRGLhjKziMHRQ3awnGtyo8hjzn7Jar5mhd"
}
```

XML

```
<xml>
    <data>
        <_xxh>026e82a364dd17e6</_xxh>
        <channel>email</channel>
        <custom_fields>
            <custom_int_field_2>20000</custom_int_field_2>
        </custom_fields>
        <db_id>1</db_id>
        <email>example@example.com</email>
        <hash_id>fcf06ea1</hash_id>
        <priority>0</priority>
        <profile_id>6643052fee44bb50a671161d</profile_id>
        <resource_id>29</resource_id>
        <status>subscribed</status>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <next_from_id>cPydNpzRRGLhjKziMHRQ3awnGtyo8hjzn7Jar5mhdm</next_from_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	string	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории
resource_id	int	Идентификатор ресурса, связанного с подпиской
channel	string	Тип канала
custom_fields	object	Стандартные и дополнительные поля подписки. Получить информацию о полях подписки в ресурсе можно с помощью метода /v1.1/resources/fields_get
cats	array	Категории ресурса, на которые подписан профиль.

Дополнительно для Email-канала:

Параметры	Тип	Описание
email	string	Email-адрес

Дополнительно для PUSH-канала:

Параметры	Тип	Описание
provider	string	Тип провайдера
subscription_id	string	Идентификатор подписки

Дополнительно для SMS-канала:

Параметры	Тип	Описание
phone	string	Номер телефона

Дополнительно для Telegram-канала:

Параметры	Тип	Описание
cc_data	object	ID чата в Telegram-бот

Дополнительно для WhatsApp*-канала:

Параметры	Тип	Описание
cc_data	object	Телефон профиля в WhatsApp*

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Получить информацию о подписке профиля

Описание

Получить подробную информацию об одной из подписок профиля.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscriptions/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных
format	string	“json”		Формат возвращаемых данных (по умолчанию – “json”)
profile_id	string	“abcdefghijklmnqrstuvwxyz”	Да	Идентификатор профиля
resource_id	int	1	Да	Идентификатор ресурса

Дополнительно для Email-канала:

Параметры	Тип	Пример	Описание
channel	string	“email”	Тип канала
email	string	“john@example.com”	Email-адрес

Дополнительно для PUSH-канала:

Параметры	Тип	Пример	Описание
channel	string	“push”	Тип канала
provider	string	“Firefox”	Тип провайдера
subscription_id	string	“abcdefghijklmnqrstuvwxyz”	Идентификатор подписки

Дополнительно для SMS-канала:

Параметры	Тип	Пример	Описание
channel	string	“sms”	Тип канала
phone	string	“+79000000000”	Номер телефона

Дополнительно для Telegram-канала:

Параметры	Тип	Пример	Описание
channel	string	“telegram_bot”	Тип канала

cc_data	object	{
		“id”: 1
		}
			id чата в Telegram-бот

Дополнительно для WhatsApp*-канала:

Параметры	Тип	Пример	Описание
channel	string	“whatsapp”	Тип канала

cc_data	object	{
		“phone”: “+79000000000”
		}
			Телефон профиля в WhatsApp*

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "resource_id": 26,
    "channel": "push",
    "subscription_id": "abcdefghijklmnqrstuvwxyz",
    "profile_id": "abcdefghijklmnqrstuvwxyz",
    "provider": "Firefox"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
  <resource_id>26</resource_id>
  <channel>push</channel>
  <subscription_id>abcdefghijklmnqrstuvwxyz</subscription_id>
  <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
  <provider>Firefox</provider>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "channel": "push",
    "priority": 10,
    "provider": "Firefox",
    "resource_id": 1,
    "status": "subscribed",
    "subscription_id": "abcdefghijklmnqrstuvwxyz",
    "custom_fields": {
      "_device_type": "mob",
      "_os": "Android",
      "custom_field_1": "test value"
    },
    "cats": ["news_category", "transaction_category"]
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <channel>push</channel>
        <priority>0</priority>
        <provider>Firefox</provider>
        <resource_id>26</resource_id>
        <status>subscribed</status>
        <subscription_id>abcdefghijklmnqrstuvwxyz</subscription_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
channel	string	Тип канала (email, push или sms)

Параметр	Тип	Описание
priority	int	Приоритет подписки (меньше число – больше приоритет)
resource_id	int	Идентификатор ресурса, связанного с подпиской
status	string	Статус подписки
custom_fields	object	Стандартные и дополнительные поля подписки. Получить информацию о полях подписки в ресурсе можно с помощью метода /v1.1/resources/fields_get
cats	array	Категории ресурса, на которые подписан профиль.

Дополнительно для Email-канала:

Параметры	Тип	Описание
email	string	Email-адрес

Дополнительно для PUSH-канала:

Параметры	Тип	Описание
provider	string	Тип провайдера
subscription_id	string	Идентификатор подписки

Дополнительно для SMS-канала:

Параметры	Тип	Описание
phone	string	Номер телефона

Дополнительно для Telegram-канала:

Параметры	Тип	Описание
cc_data	object	id чата в Telegram-бот

Дополнительно для WhatsApp*-канала:

Параметры	Тип	Описание
cc_data	object	Телефон профиля в WhatsApp*

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Удалить подписку профиля

Описание

Удалить подписку профиля.

Подписка останется в профиле, но получит метку: `"is_delete" - true`.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscriptions/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
profile_id	string	“abcdefghijklmnqrstuvwxyz”	Да	Идентификатор профиля
resource_id	int	1	Да	Идентификатор ресурса
db_id	int	1	Да	Идентификатор базы данных

Дополнительно для Email-канала:

Параметры	Тип	Пример	Описание
channel	string	“email”	Тип канала
email	string	“john@example.com”	Email-адрес

Дополнительно для PUSH-канала:

Параметры	Тип	Пример	Описание
channel	string	“push”	Тип канала
provider	string	“Firefox”	Тип провайдера
subscription_id	string	“abcdefghijklmnqrstuvwxyz”	Идентификатор подписки

Дополнительно для SMS-канала:

Параметры	Тип	Пример	Описание
channel	string	“sms”	Тип канала

Параметры	Тип	Пример	Описание
phone	string	“+79000000000”	Номер телефона

Дополнительно для Telegram-канала:

Параметры	Тип	Пример	Описание
channel	string	“telegram_bot”	Тип канала

cc_data	object	{
		“id”: 1
		}
			id чата в Telegram-бот

Дополнительно для WhatsApp*-канала:

Параметры	Тип	Пример	Описание
channel	string	“whatsapp”	Тип канала

cc_data	object	{
		“phone”: “+79000000000”
		}
			Телефон профиля в WhatsApp*

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "resource_id": 26,
    "channel": "push",
    "subscription_id": "abcdefghijklmnqrstuvwxyz",
    "profile_id": "abcdefghijklmnqrstuvwxyz",
    "provider": "Firefox"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <resource_id>26</resource_id>
    <channel>push</channel>
    <subscription_id>abcdefghijklmnqrstuvwxyz</subscription_id>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
    <provider>Firefox</provider>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Восстановить удаленную подписку профиля

Описание

Восстановить удаленную подписку.

Изменить `"is_delete"` с `false` на `true`.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscriptions/undelete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”		Формат возвращаемых данных (по умолчанию – “json”)
profile_id	string	“abcdefghijklmnqrstuvwxyz”	Да	Идентификатор профиля
resource_id	int	1	Да	Идентификатор ресурса
db_id	int	1	Да	Идентификатор базы данных

Дополнительно для Email-канала:

Параметры	Тип	Пример	Описание
channel	string	“email”	Тип канала
email	string	“john@example.com”	Email-адрес

Дополнительно для PUSH-канала:

Параметры	Тип	Пример	Описание
channel	string	“push”	Тип канала
provider	string	“Firefox”	Тип провайдера
subscription_id	string	“abcdefghijklmnqrstuvwxyz”	Идентификатор подписки

Дополнительно для SMS-канала:

Параметры	Тип	Пример	Описание
channel	string	“sms”	Тип канала

Параметры	Тип	Пример	Описание
phone	string	“+79000000000”	Номер телефона

Дополнительно для Telegram-канала:

Параметры	Тип	Пример	Описание
channel	string	“telegram_bot”	Тип канала

cc_data	object	{
		“id”: 1
		}
			id чата в Telegram-бот

Дополнительно для WhatsApp*-канала:

Параметры	Тип	Пример	Описание
channel	string	“whatsapp”	Тип канала

cc_data	object	{
		“phone”: “+79000000000”
		}
			Телефон профиля в WhatsApp*

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "resource_id": 26,
    "channel": "push",
    "subscription_id": "abcdefghijklmnqrstuvwxyz",
    "profile_id": "abcdefghijklmnqrstuvwxyz",
    "provider": "Firefox"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <resource_id>26</resource_id>
    <channel>push</channel>
    <subscription_id>abcdefghijklmnqrstuvwxyz</subscription_id>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
    <provider>Firefox</provider>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*

Приостановить все подписки

Описание

Изменить статус подписок с `"subscribed"` на `"suspended"` для базы данных подписчиков.

Подписки с другими статусами не изменятся.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscriptions/suspend_all`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
db_id	int	1	Да	Идентификатор базы данных
channel	int	“email”	Нет	Тип канала
resource_id	int	1	Нет	Идентификатор ресурса

Если передать параметр `"channel"`, то приостановятся подписки на канал.

Если передать параметр `"resource_id"`, то приостановятся подписки на ресурс.

Если передать параметры `"channel"` и `"resource_id"`, то приостановятся подписки на канал и ресурс.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "channel": "email"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
  <channel>email</channel>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "updated": 4080
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <updated>1344</updated>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
updated	int	Количество приостановленных подписок

Восстановить все приостановленные подписки

Описание

Изменить статус подписок с `"suspended"` на `"subscribed"` для базы данных подписчиков.

Подписки с другими статусами не изменятся.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/subscriptions/unsuspend_all`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
db_id	int	1	Да	Идентификатор базы данных
channel	int	“email”	Нет	Тип канала
resource_id	int	1	Нет	Идентификатор ресурса

Если передать параметр `"channel"`, то восстановятся подписки на канал.

Если передать параметр `"resource_id"`, то восстановятся подписки на ресурс.

Если передать параметры `"channel"` и `"resource_id"`, то восстановятся подписки на канал и ресурс.

А если их не использовать, то все подписки будут восстановлены.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <db_id>1</db_id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "updated": 5428
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <updated>5428</updated>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
updated	int	Количество приостановленных подписок

Получить список баз данных

Описание

Получить список баз данных.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/databases/list/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	int	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
from_id	int	1	Нет	Идентификатор первой базы в списке по умолчанию – 1
limit	int	1	Нет	Лимит возвращаемых данных по умолчанию – 1000
ui_tags	[]String	[“tag1”, “tag2”]	Нет	Список тегов для поиска по умолчанию – []
groups	[]String	[“group1”, “group2”]	Нет	Список групп для поиска по умолчанию – []
groups_id	[]int	[0, 1]	Нет	Список Id групп для поиска по умолчанию – []

Пример запроса

JSON

```
{
  "limit": 2,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <limit>2</limit>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "id": 1,
      "name": "Database #1",
      "ui_tags": [],
      "Groups": [
        {
          "Id": 0,
          "Name": "main"
        }
      ]
    },
    {
      "id": 2,
      "name": "Database #2",
      "ui_tags": ["tag1", "tag2"],
      "Groups": [
        {
          "Id": 1,
          "Name": "subaccount"
        }
      ]
    }
  ],
  "error": 0,
  "error_text": "Successful operation",
  "next_from_id": 6,
  "total_count": 4
}
```

XML

```
<xml>
  <data>
    <id>1</id>
    <name>Database #1</name>
    <Groups>
      <Id>0</Id>
      <Name>main</Name>
    </Groups>
  </data>
  <data>
    <id>2</id>
    <name>Database #2</name>
    <ui_tags>tag1</ui_tags>
    <ui_tags>tag2</ui_tags>
    <Groups>
      <Id>1</Id>
      <Name>subaccount</Name>
    </Groups>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
  <next_from_id>6</next_from_id>
  <total_count>4</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	int	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории
total_count	int	Общее количество баз данных
id	int	Идентификатор базы данных
name	string	Название базы данных
ui_tags	[]String	Список тегов

Параметр	Тип	Описание
groups	[]String	Список групп

Получить информацию о базе данных

Описание

Получить подробную информацию о базе данных.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/databases/get/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор базы данных
format	int	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
  "id": 1,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>1</id>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "id": 1,
    "name": "B2C customers",
    "suppress_id": 1,
    "description": "Может содержать некоторую информацию о базе данных"
  },
  "error": 0,
  "error_text": "Successful operation"
}
```


XML


```
<xml>
  <data>
    <id>1</id>
    <name>B2C customers</name>
    <suppress_id>1</suppress_id>
    <description>Может содержать некоторую информацию о базе данных</description>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
suppress_id	int	Идентификатор стоп-списка базы
description	string	Описание базы данных

Получить информацию о полях базы данных

Описание

Получить информацию о полях, используемых в базе данных. Типы полей и допустимые значения описаны в **отдельной статье**.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/databases/fields_get/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор базы данных
format	int	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
  "id": 1,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>1</id>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "category": "system",
      "title": "Id",
      "name": "_id",
      "required": true,
      "submanager": false,
      "type": "unicode",
      "enum_values": null
    },
    {
      "category": "default",
      "title": "FirstName",
      "name": "_fname",
      "required": false,
      "submanager": false,
      "type": "unicode",
      "enum_values": null
    },
    {
      "category": "custom",
      "title": "Custom_String",
      "name": "Custom_String",
      "required": false,
      "submanager": false,
      "type": "unicode",
      "enum_values": null
    },
    {
      "category": "custom",
      "title": "Custom_Int",
      "name": "Custom_Int",
      "required": false,
      "submanager": false,
      "type": "int",
      "enum_values": null
    },
    {
      "category": "custom",
      "title": "Custom_Bool",
      "name": "Custom_Bool",
      "required": false,
      "submanager": false,
      "type": "bool",
      "enum_values": null
    },
    {
      "category": "custom"
```

```
      "category": "custom",
      "title": "Custom_Date",
      "name": "Custom_Date",
      "required": false,
      "submanager": false,
      "type": "date",
      "enum_values": null
    },
    {
      "category": "custom",
      "title": "Custom_Tags",
      "name": "Custom_Tags",
      "required": false,
      "submanager": false,
      "type": "tags",
      "enum_values": null
    },
    {
      "category": "custom",
      "title": "Custom_Enum",
      "name": "Custom_Enum",
      "required": false,
      "submanager": false,
      "type": "enum",
      "enum_values": [
        {
          "name": "GMail",
          "value": 2
        },
        {
          "name": "HotMail",
          "value": 3
        },
        {
          "name": "Yahoo!",
          "value": 1
        }
      ]
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```


XML

```
<xml>
  <data>
    <category>system</category>
    <title>Id</title>
    <name>_id</name>
    <required>true</required>
    <submanager>false</submanager>
    <type>unicode</type>
  </data>
  <data>
    <category>default</category>
    <title>FirstName</title>
    <name>_fname</name>
    <required>false</required>
    <submanager>false</submanager>
    <type>unicode</type>
  </data>
  <data>
    <category>custom</category>
    <title>Custom_String</title>
    <name>Custom_String</name>
    <required>false</required>
    <submanager>false</submanager>
    <type>unicode</type>
  </data>
  <data>
    <category>custom</category>
    <title>Custom_Int</title>
    <name>Custom_Int</name>
    <required>false</required>
    <submanager>false</submanager>
    <type>int</type>
  </data>
  <data>
    <category>custom</category>
    <title>Custom_Bool</title>
    <name>Custom_Bool</name>
    <required>false</required>
    <submanager>false</submanager>
    <type>bool</type>
  </data>
  <data>
    <category>custom</category>
    <title>Custom_Date</title>
    <name>Custom_Date</name>
    <required>false</required>
    <submanager>false</submanager>
    <type>date</type>
```

```
    <type>date</type>
  </data>
  <data>
    <category>custom</category>
    <title>Custom_Tags</title>
    <name>Custom_Tags</name>
    <required>false</required>
    <submanager>false</submanager>
    <type>tags</type>
  </data>
  <data>
    <category>custom</category>
    <title>Custom_Enum</title>
    <name>Custom_Enum</name>
    <required>false</required>
    <submanager>false</submanager>
    <type>enum</type>
    <enum_values>
      <name>GMail</name>
      <value>2</value>
    </enum_values>
    <enum_values>
      <name>HotMail</name>
      <value>3</value>
    </enum_values>
    <enum_values>
      <name>Yahoo!</name>
      <value>1</value>
    </enum_values>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
category	string	Категория поля базы: system, default, custom
title	string	Название поля для интерфейса
name	string	Название поля для запросов
required	bool	Обязательное поле (true)
submanager	bool	Поле добавлено в менеджер подписок

Параметр	Тип	Описание
type	string	Тип поля
enum_values	array	Значения полей со списком вариантов

Очистка базы данных для тестирования

Описание

Использование этого метода рассчитано для тестирования.

1. Удаляются все профили из базы данных.
2. Удаляются все данные, связанные с этой базой данных.

> **CAUTION**
>
> Внимание! Статистика по базе данных не очищается и не удаляется.

Метод будет работать только при условии, что в файле конфигурации `main.json` добавлен параметр:

```
{
    "ALLOW_DATA_WIPE": true
}
```

URL-адрес

Метод: POST

`https://example.com/api/v1.1/databases/wipe`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор базы данных
format	int	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
  "id": 1,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>1</id>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "count_action": 0,
    "count_profile": 0
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <count_action>0</count_action>
    <count_profile>0</count_profile>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	JSON object	Данные
data.count_action	int	Количество удаленных действий профилей
data.count_profile	int	Количество удаленных профилей
error	int	Код ошибки
error_text	string	Текст ошибки

Получить статистику по базе данных

Описание

Получить статистику подписок на определенный ресурс в рамках базы.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/databases/get_counters`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор базы данных
resource_id	int	1	Да	Идентификатор ресурса
format	int	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 5,
    "resource_id": 24,
    "format": "json"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>5</id>
    <resource_id>24</resource_id>
    <format>json</format>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "channel": "email",
            "status": {
                "all": 1336,
                "subscribed": 925,
                "unsubscribed": 0,
                "complained": 0,
                "hardbounced": 411,
                "unconfirmed": 0,
                "suspended": 0,
                "invalid": 0,
                "subscribed_sendable": 925,
                "unconfirmed_sendable": 0
            }
        },
        {
            "channel": "sms",
            "status": {
                "all": 1336,
                "subscribed": 0,
                "unsubscribed": 0,
                "complained": 0,
                "hardbounced": 0,
                "unconfirmed": 0,
                "suspended": 0,
                "invalid": 0,
                "subscribed_sendable": 0,
                "unconfirmed_sendable": 0
            }
        },
        {
            "channel": "push",
            "status": {
                "all": 1336,
                "subscribed": 0,
                "unsubscribed": 0,
                "complained": 0,
                "hardbounced": 0,
                "unconfirmed": 0,
                "suspended": 0,
                "invalid": 0,
                "subscribed_sendable": 0,
                "unconfirmed_sendable": 0
            }
        }
    ],
    "error": 0
```

```
    "error" : 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <channel>email</channel>
        <status>
            <all>1336</all>
            <subscribed>925</subscribed>
            <unsubscribed>0</unsubscribed>
            <complained>0</complained>
            <hardbounced>411</hardbounced>
            <unconfirmed>0</unconfirmed>
            <suspended>0</suspended>
            <invalid>0</invalid>
            <subscribed_sendable>925</subscribed_sendable>
            <unconfirmed_sendable>0</unconfirmed_sendable>
        </status>
    </data>
    <data>
        <channel>sms</channel>
        <status>
            <all>1336</all>
            <subscribed>0</subscribed>
            <unsubscribed>0</unsubscribed>
            <complained>0</complained>
            <hardbounced>0</hardbounced>
            <unconfirmed>0</unconfirmed>
            <suspended>0</suspended>
            <invalid>0</invalid>
            <subscribed_sendable>0</subscribed_sendable>
            <unconfirmed_sendable>0</unconfirmed_sendable>
        </status>
    </data>
    <data>
        <channel>push</channel>
        <status>
            <all>1336</all>
            <subscribed>0</subscribed>
            <unsubscribed>0</unsubscribed>
            <complained>0</complained>
            <hardbounced>0</hardbounced>
            <unconfirmed>0</unconfirmed>
            <suspended>0</suspended>
            <invalid>0</invalid>
            <subscribed_sendable>0</subscribed_sendable>
            <unconfirmed_sendable>0</unconfirmed_sendable>
        </status>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

```
</xml1>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	array of objects	Массив полученных данных по каналам ресурса
error	int	Код ошибки
error_text	string	Текст ошибки

Обновить статистику по базе данных

Описание

Обновить статистику подписок профилей в базе.

Возвращает ответ с ID задания, не дожидаясь выполнения.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/databases/refresh_counters`

Параметры запроса

```
    <tr>
        <td>format</td>
        <td>JSON object </td>
        <td>"json"</td>
        <td>Нет</td>
        <td> Формат возвращаемых данных по умолчанию – "json" </td>
    </tr>
    <tr>
        <td>feedback</td>
        <td>JSON object </td>
        <td>
```

```
"feedback": {
    "type": "JSON",
    "url": "http://example.com/receiver"
}
```

```
            </td>
        <td>Нет</td>
        <td>Параметр, опередеяющий способ отправки данных об обновлении статистики. <br/>
Параметр "type" – формат или метод, используемый для отправки данных.<br/> "url" – URL-
адрес, на который будут отправлены данные статистики.<br/></td>
    </tr>
</tbody>
```

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	1	Да	Идентификатор базы данных

resource_id	int	1	Да	Идентификатор ресурса

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 5,
    "resource_id": 24,
    "format": "json",
    "feedback": {
        "type": "JSON",
        "url": "http://example.com/receiver"
    }
}
```

XML

```
<xml>
     <token>abcdefghijklmnqrstuvwxyz</token>
     <id>5</id>
     <resource_id>24</resource_id>
     <format>json</format>
     <feedback>
         <type>JSON</type>
         <url>http://example.com/receiver</url>
     </feedback>
 </xml>
```

Пример ответа

JSON

```
{
    "data": {
        "task_id": "6908adeb-346e-4f5e-9f68-df062bf795cf"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
     <data>
         <task_id>6908adeb-346e-4f5e-9f68-df062bf795cf</task_id>
     </data>
     <error>0</error>
     <error_text>Successful operation</error_text>
 </xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	JSON objects	Содержит идентификатор задания – task_id
error	int	Код ошибки
error_text	string	Текст ошибки

Проверка статуса задания

URL-адрес

Метод: POST

`https://example.com/api/v1.1/tasks/status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
task_id	string	“518f4997-ec43-4df5-bfd0-093aaacf679f”	Да	Идентификатор задания

Пример запроса

JSON

```
{
    "task_id": "930bfdb0-a5a8-4207-a77d-6b989bdea326",
    "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
     <task_id>930bfdb0-a5a8-4207-a77d-6b989bdea326</task_id>
     <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "status": "completed",
        "progress": 100,
        "result": [
            {
                "channels": [
                    {
                        "channel": "email",
                        "status": {
                            "all": 1336,
                            "complained": 0,
                            "hardbounced": 411,
                            "invalid": 0,
                            "subscribed": 925,
                            "subscribed_sendable": 925,
                            "suspended": 0,
                            "unconfirmed": 0,
                            "unconfirmed_sendable": 0,
                            "unsubscribed": 0
                        }
                    },
                    {
                        "channel": "push",
                        "status": {
                            "all": 1336,
                            "complained": 0,
                            "hardbounced": 0,
                            "invalid": 0,
                            "subscribed": 0,
                            "subscribed_sendable": 0,
                            "suspended": 0,
                            "unconfirmed": 0,
                            "unconfirmed_sendable": 0,
                            "unsubscribed": 0
                        }
                    },
                    {
                        "channel": "sms",
                        "status": {
                            "all": 1336,
                            "complained": 0,
                            "hardbounced": 0,
                            "invalid": 0,
                            "subscribed": 0,
                            "subscribed_sendable": 0,
                            "suspended": 0,
                            "unconfirmed": 0,
                            "unconfirmed_sendable": 0
```

```
                        "unconfirmed_sendable": 0,
                        "unsubscribed": 0
                    }
                }
            ],
            "in_progress": false,
            "last_count_time": "counted at 2024-04-17T21:36:01Z in 82.746959ms",
            "refresh_task_id": "",
            "resource_id": 24,
            "total_count": 1336,
            "updated": "2024-04-17T21:36:01.449Z"
        }
    ]
},
"error": 0,
"error_text": "Successful operation"
}
```


XML

```
<xml>
     <data>
         <status>completed</status>
         <progress>100</progress>
         <result>
             <channels>
                 <channel>email</channel>
                 <status>
                     <all>1336</all>
                     <complained>0</complained>
                     <hardbounced>411</hardbounced>
                     <invalid>0</invalid>
                     <subscribed>925</subscribed>
                     <subscribed_sendable>925</subscribed_sendable>
                     <suspended>0</suspended>
                     <unconfirmed>0</unconfirmed>
                     <unconfirmed_sendable>0</unconfirmed_sendable>
                     <unsubscribed>0</unsubscribed>
                 </status>
             </channels>
             <channels>
                 <channel>push</channel>
                 <status>
                     <all>1336</all>
                     <complained>0</complained>
                     <hardbounced>0</hardbounced>
                     <invalid>0</invalid>
                     <subscribed>0</subscribed>
                     <subscribed_sendable>0</subscribed_sendable>
                     <suspended>0</suspended>
                     <unconfirmed>0</unconfirmed>
                     <unconfirmed_sendable>0</unconfirmed_sendable>
                     <unsubscribed>0</unsubscribed>
                 </status>
             </channels>
             <channels>
                 <channel>sms</channel>
                 <status>
                     <all>1336</all>
                     <complained>0</complained>
                     <hardbounced>0</hardbounced>
                     <invalid>0</invalid>
                     <subscribed>0</subscribed>
                     <subscribed_sendable>0</subscribed_sendable>
                     <suspended>0</suspended>
                     <unconfirmed>0</unconfirmed>
                     <unconfirmed_sendable>0</unconfirmed_sendable>
                     <unsubscribed>0</unsubscribed>
                 </status>
```

```
                </status>
            </channels>
            <in_progress>false</in_progress>
            <last_count_time>counted at 2024-04-17T21:36:01Z in
82.746959ms</last_count_time>
            <refresh_task_id></refresh_task_id>
            <resource_id>24</resource_id>
            <total_count>1336</total_count>
            <updated>2024-04-17T21:36:01.449Z</updated>
        </result>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Получить список ресурсов

Описание

Получить список ресурсов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/resources/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
from_id	int	1	Нет	Идентификатор первого ресурса в списке (по умолчанию – 1)
limit	int	1	Нет	Лимит на количество возвращаемых ресурсов (по умолчанию – 1000)
groups	[]String	`[ "group1", "group2" ]`	Нет	Список групп для поиска (по умолчанию – `[]`)
groups_id	[]int	`[ 0, 1 ]`	Нет	Список Id групп для поиска (по умолчанию – `[]`)

Пример запроса

JSON

```
{
  "from_id": 1,
  "limit": 1,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <from_id>1</from_id>
  <limit>1</limit>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "id": 1,
      "name": "Resource",
      "groups": [
        {
          "id": 1,
          "name": "subaccount"
        }
      ]
    }
  ],
  "error": 0,
  "error_text": "Successful operation",
  "next_from_id": 4,
  "total_count": 1
}
```

XML

```
<xml>
  <data>
    <id>1</id>
    <name>Resource</name>
    <groups>
      <id>1</id>
      <name>subaccount</name>
    </groups>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
  <next_from_id>4</next_from_id>
  <total_count>1</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	int	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории
id	int	Идентификатор ресурса
name	string	Название ресурса
total_count	int	Общее количество ресурса
groups	[]String	Список групп

Получить информацию о ресурсе

Описание

Получить подробную информацию о конкретном ресурсе.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/resources/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор ресурса

Пример запроса

JSON

```
{
  "id": 1,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>1</id>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "channels": ["email", "push"],
    "dbs_ids": [1],
    "id": 1,
    "name": "Resource",
    "trk_domain": "tracking.example.com",
    "url": "https://blog.example:8080"
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <channels>email</channels>
    <channels>push</channels>
    <dbs_ids>1</dbs_ids>
    <id>1</id>
    <name>Resource</name>
    <trk_domain>tracking.example.com</trk_domain>
    <url>https://blog.example:8080</url>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
channel	JSON array	Доступные каналы
dbs_ids	JSON array	Прикрепленные базы данных
id	int	Идентификатор ресурса
name	string	Название ресурса
trk_domain	string	Домен для отслеживания действий подписчиков

Параметр	Тип	Описание
url	string	URL-адрес, используемый в ресурсе
descr	string	Описание ресурса

Получить информацию о полях подписки ресурса

Описание

Получить информацию о стандартных и дополнительных полях подписки.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/resources/fields_get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор ресурса
channel	string	“email” “sms” “push” “telegram_bot” “whatsapp”	Да	Канал подписки на ресурс

Пример запроса

JSON

```
{
  "id": 1,
  "channel": "email",
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>1</id>
  <channel>email</channel>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "category": "custom",
      "title": "New_field_1",
      "name": "api_new_field_1",
      "required": false,
      "type": "int"
    },
    {
      "category": "system",
      "title": "Device type",
      "name": "_device_type",
      "required": false,
      "type": "unicode",
      "dictionary": "devices"
    },
    {
      "category": "system",
      "title": "Device model",
      "name": "_device_model",
      "required": false,
      "type": "unicode"
    },
    {
      "category": "system",
      "title": "Device name",
      "name": "_device_name",
      "required": false,
      "type": "unicode"
    },
    {
      "category": "system",
      "title": "Platform",
      "name": "_os",
      "required": false,
      "type": "unicode",
      "dictionary": "oses"
    },
    {
      "category": "system",
      "title": "Platform version",
      "name": "_os_ver",
      "required": false,
      "type": "version"
    },
    {
      "category": "system",
      "title": "Platform time zone"
```

```
    "title": "Platform time zone",
    "name": "_os_tz",
    "required": false,
    "type": "unicode",
    "dictionary": "tz"
  },
  {
    "category": "system",
    "title": "Platform language",
    "name": "_os_language",
    "required": false,
    "type": "unicode",
    "dictionary": "languages"
  },
  {
    "category": "system",
    "title": "Ad tracking allowed",
    "name": "_ad_track",
    "required": false,
    "type": "bool"
  },
  {
    "category": "system",
    "title": "Ad identifier",
    "name": "_ad_id",
    "required": false,
    "type": "unicode"
  },
  {
    "category": "system",
    "title": "Application ID",
    "name": "_app_id",
    "required": false,
    "type": "unicode"
  },
  {
    "category": "system",
    "title": "Application installation ID",
    "name": "_app_iid",
    "required": false,
    "type": "unicode"
  },
  {
    "category": "system",
    "title": "Application version",
    "name": "_app_ver",
    "required": false,
    "type": "version"
  },
  {
```

```
    {
      "category": "system",
      "title": "Browser",
      "name": "_browser_name",
      "required": false,
      "type": "unicode",
      "dictionary": "browsers"
    },
    {
      "category": "system",
      "title": "Browser version",
      "name": "_browser_ver",
      "required": false,
      "type": "version"
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```


XML

```
<xml>
  <data>
    <category>custom</category>
    <title>New_field_1</title>
    <name>api_new_field_1</name>
    <required>false</required>
    <type>int</type>
  </data>
  <data>
    <category>system</category>
    <title>Device type</title>
    <name>_device_type</name>
    <required>false</required>
    <type>unicode</type>
    <dictionary>devices</dictionary>
  </data>
  <data>
    <category>system</category>
    <title>Device model</title>
    <name>_device_model</name>
    <required>false</required>
    <type>unicode</type>
  </data>
  <data>
    <category>system</category>
    <title>Device name</title>
    <name>_device_name</name>
    <required>false</required>
    <type>unicode</type>
  </data>
  <data>
    <category>system</category>
    <title>Platform</title>
    <name>_os</name>
    <required>false</required>
    <type>unicode</type>
    <dictionary>oses</dictionary>
  </data>
  <data>
    <category>system</category>
    <title>Platform version</title>
    <name>_os_ver</name>
    <required>false</required>
    <type>version</type>
  </data>
  <data>
    <category>system</category>
    <title>Platform time zone</title>
```

```
  <title>Platform time zone</title>
  <name>_os_tz</name>
  <required>false</required>
  <type>unicode</type>
  <dictionary>tz</dictionary>
</data>
<data>
  <category>system</category>
  <title>Platform language</title>
  <name>_os_language</name>
  <required>false</required>
  <type>unicode</type>
  <dictionary>languages</dictionary>
</data>
<data>
  <category>system</category>
  <title>Ad tracking allowed</title>
  <name>_ad_track</name>
  <required>false</required>
  <type>bool</type>
</data>
<data>
  <category>system</category>
  <title>Ad identifier</title>
  <name>_ad_id</name>
  <required>false</required>
  <type>unicode</type>
</data>
<data>
  <category>system</category>
  <title>Application ID</title>
  <name>_app_id</name>
  <required>false</required>
  <type>unicode</type>
</data>
<data>
  <category>system</category>
  <title>Application installation ID</title>
  <name>_app_iid</name>
  <required>false</required>
  <type>unicode</type>
</data>
<data>
  <category>system</category>
  <title>Application version</title>
  <name>_app_ver</name>
  <required>false</required>
  <type>version</type>
</data>
<data>
```

```
    <category>system</category>
    <title>Browser</title>
    <name>_browser_name</name>
    <required>false</required>
    <type>unicode</type>
    <dictionary>browsers</dictionary>
  </data>
  <data>
    <category>system</category>
    <title>Browser version</title>
    <name>_browser_ver</name>
    <required>false</required>
    <type>version</type>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
category	string	Категория поля подписки `"system","custom"`
title	string	Название поля для интерфейса
name	string	Название поля для запросов
required	bool	Обязательное поле (true)
type	string	Тип поля
default_value	*	Значение поля по умолчанию

Параметр	Тип	Описание

dictionary	string	Название поля, используемое в запросе **/v1.1/dictionary** (запрос на получение значений системных полей, которые отображаются в интерфейсе в виде выпадающего списка)
		
		`"browsers"`
		`"devices"`
		`"tz"`
		`"oses"`
		`"languages"`

Системные поля подписки

Параметр	Тип	Пример	Описание
_device_type	string	“mob”	Тип устройства. В параметр рекомендуется передавать определенные значения. Получить список допустимых значений можно через запрос `/v1.1/dictionary ("devices")`
_device_model	string	“MHA-L29”	Модель устройства.
_device_name	string	“HUAWEI Mate 9”	Имя устройства.
_os	string	“Android”	Операционная система платформы. В параметр рекомендуется передавать определенные значения. Получить список допустимых значений можно через запрос `/v1.1/dictionary ("oses")`
_os_ver	string	“v11.0”	Версия ОС.
_os_tz	string	“+0300”	Часовой пояс платформы. В параметр рекомендуется передавать определенные значения. Получить список допустимых значений можно через запрос `/v1.1/dictionary ("tz")`
_os_language	string	“ru”	Язык платформы. В параметр рекомендуется передавать определенные значения. Получить список допустимых значений можно через запрос `/v1.1/dictionary ("languages")`
_ad_track	bool	true	Реклама разрешена.
_ad_id	string	“ad 123”	Рекламный идентификатор.
_app_id	string	“app 123”	Идентификатор приложения.

Параметр	Тип	Пример	Описание
_app_iid	string	“example iid”	Идентификатор установки приложения.
_app_ver	string	“v1.2.3.4”	Версия приложения.
_browser_name	string	“Chrome”	Браузер. В параметр рекомендуется передавать определенные значения. Получить список допустимых значений можно через запрос `/v1.1/dictionary ("browsers")`
_browser_ver	string	“4.5.6.7.8”	Версия браузера.

Получить статистику по ресурсам

Описание

Получить статистику по подпискам на конкретный ресурс.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/resources/get_counters`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
list_id	int	1	Да	Идентификатор базы данных
id	int	1	Да	Идентификатор ресурса
format	JSON object	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "list_id": 24,
    "id": 5,
    "format": "json"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <list_id>24</list_id>
    <id>5</id>
    <format>json</format>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "channel": "email",
            "status": {
                "all": 1336,
                "subscribed": 925,
                "unsubscribed": 0,
                "complained": 0,
                "hardbounced": 411,
                "unconfirmed": 0,
                "suspended": 0,
                "invalid": 0,
                "subscribed_sendable": 925,
                "unconfirmed_sendable": 0
            }
        },
        {
            "channel": "sms",
            "status": {
                "all": 1336,
                "subscribed": 0,
                "unsubscribed": 0,
                "complained": 0,
                "hardbounced": 0,
                "unconfirmed": 0,
                "suspended": 0,
                "invalid": 0,
                "subscribed_sendable": 0,
                "unconfirmed_sendable": 0
            }
        },
        {
            "channel": "push",
            "status": {
                "all": 1336,
                "subscribed": 0,
                "unsubscribed": 0,
                "complained": 0,
                "hardbounced": 0,
                "unconfirmed": 0,
                "suspended": 0,
                "invalid": 0,
                "subscribed_sendable": 0,
                "unconfirmed_sendable": 0
            }
        }
    ],
    "error": 0
```

```
    "error" : 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
     <data>
         <channel>email</channel>
         <status>
             <all>1336</all>
             <subscribed>925</subscribed>
             <unsubscribed>0</unsubscribed>
             <complained>0</complained>
             <hardbounced>411</hardbounced>
             <unconfirmed>0</unconfirmed>
             <suspended>0</suspended>
             <invalid>0</invalid>
             <subscribed_sendable>925</subscribed_sendable>
             <unconfirmed_sendable>0</unconfirmed_sendable>
         </status>
     </data>
     <data>
         <channel>sms</channel>
         <status>
             <all>1336</all>
             <subscribed>0</subscribed>
             <unsubscribed>0</unsubscribed>
             <complained>0</complained>
             <hardbounced>0</hardbounced>
             <unconfirmed>0</unconfirmed>
             <suspended>0</suspended>
             <invalid>0</invalid>
             <subscribed_sendable>0</subscribed_sendable>
             <unconfirmed_sendable>0</unconfirmed_sendable>
         </status>
     </data>
     <data>
         <channel>push</channel>
         <status>
             <all>1336</all>
             <subscribed>0</subscribed>
             <unsubscribed>0</unsubscribed>
             <complained>0</complained>
             <hardbounced>0</hardbounced>
             <unconfirmed>0</unconfirmed>
             <suspended>0</suspended>
             <invalid>0</invalid>
             <subscribed_sendable>0</subscribed_sendable>
             <unconfirmed_sendable>0</unconfirmed_sendable>
         </status>
     </data>
     <error>0</error>
     <error_text>Successful operation</error_text>
</xml>
```

</ xml1>
```

## Возвращаемые параметры

| Параметр | Тип | Описание |
|---|---|---|
| data | array of objects | Массив полученных данных по каналам ресурса |
| error | int | Код ошибки |
| error_text | string | Текст ошибки |
```

Обновить статистику по ресурсам

Описание

Обновить статистику подписок профилей на ресурс.

Возвращает ответ с ID задания, не дожидаясь выполнения.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/resources/refresh_counters`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	1	Да	Идентификатор ресурса
list_id	int	1	Да	Идентификатор базы данных
format	JSON object	"json"	Нет	Формат возвращаемых данных по умолчанию – "json"
feedback	JSON object	```json "feedback": { "type": "JSON", "url": "http://example.com/receiver" } ```	Нет	Параметр, опередеяющий способ отправки данных об обновлении статистики. Параметр "type" - формат или метод, используемый для отправки данных. "url" - URL-адрес, на который будут отправлены данные статистики.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 24,
    "list_id": 1,
    "format": "json",
    "feedback": {
        "type": "JSON",
        "url": "http://example.com/receiver"
    }
}
```

XML

```
<xml>
     <token>abcdefghijklmnqrstuvwxyz</token>
     <id>24</id>
     <list_id>1</list_id>
     <format>json</format>
     <feedback>
         <type>JSON</type>
         <url>http://example.com/receiver</url>
     </feedback>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "task_id": "c9c3ec48-1bef-4969-9c1a-df39f0c179bb"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
     <data>
         <task_id>c9c3ec48-1bef-4969-9c1a-df39f0c179bb</task_id>
     </data>
     <error>0</error>
     <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	JSON objects	Содержит идентификатор задания - task_id
error	int	Код ошибки
error_text	string	Текст ошибки

Проверка статуса задания

URL-адрес

Метод: POST

`https://example.com/api/v1.1/tasks/status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
task_id	string	“518f4997-ec43-4df5-bfd0-093aaacf679f”	Да	Идентификатор задания

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "task_id": "c9c3ec48-1bef-4969-9c1a-df39f0c179bb"
}
```

XML

```
<xml>
     <token>abcdefghijklmnqrstuvwxyz</token>
     <task_id>c9c3ec48-1bef-4969-9c1a-df39f0c179bb</task_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "status": "completed",
        "progress": 100,
        "result": [
            {
                "channels": [
                    {
                        "channel": "sms",
                        "status": {
                            "all": 1336,
                            "complained": 0,
                            "hardbounced": 0,
                            "invalid": 0,
                            "subscribed": 0,
                            "subscribed_sendable": 0,
                            "suspended": 0,
                            "unconfirmed": 0,
                            "unconfirmed_sendable": 0,
                            "unsubscribed": 0
                        }
                    },
                    {
                        "channel": "push",
                        "status": {
                            "all": 1336,
                            "complained": 0,
                            "hardbounced": 0,
                            "invalid": 0,
                            "subscribed": 0,
                            "subscribed_sendable": 0,
                            "suspended": 0,
                            "unconfirmed": 0,
                            "unconfirmed_sendable": 0,
                            "unsubscribed": 0
                        }
                    },
                    {
                        "channel": "email",
                        "status": {
                            "all": 1336,
                            "complained": 0,
                            "hardbounced": 434,
                            "invalid": 0,
                            "subscribed": 842,
                            "subscribed_sendable": 842,
                            "suspended": 0,
                            "unconfirmed": 0,
                            "unconfirmed_sendable": 0
```

```
                        "unconfirmed_sendable": 0,
                        "unsubscribed": 60
                    }
                }
            ],
            "in_progress": false,
            "last_count_time": "counted at 2024-04-19T13:31:08Z in 231.374251ms",
            "list_id": 1,
            "refresh_task_id": "",
            "updated": "2024-04-19T13:31:08.927Z"
        }
    ]
},
"error": 0,
"error_text": "Successful operation"
}
```


XML

```
<xml>
    <data>
        <status>completed</status>
        <progress>100</progress>
        <result>
            <channels>
                <channel>sms</channel>
                <status>
                    <all>1336</all>
                    <complained>0</complained>
                    <hardbounced>0</hardbounced>
                    <invalid>0</invalid>
                    <subscribed>0</subscribed>
                    <subscribed_sendable>0</subscribed_sendable>
                    <suspended>0</suspended>
                    <unconfirmed>0</unconfirmed>
                    <unconfirmed_sendable>0</unconfirmed_sendable>
                    <unsubscribed>0</unsubscribed>
                </status>
            </channels>
            <channels>
                <channel>push</channel>
                <status>
                    <all>1336</all>
                    <complained>0</complained>
                    <hardbounced>0</hardbounced>
                    <invalid>0</invalid>
                    <subscribed>0</subscribed>
                    <subscribed_sendable>0</subscribed_sendable>
                    <suspended>0</suspended>
                    <unconfirmed>0</unconfirmed>
                    <unconfirmed_sendable>0</unconfirmed_sendable>
                    <unsubscribed>0</unsubscribed>
                </status>
            </channels>
            <channels>
                <channel>email</channel>
                <status>
                    <all>1336</all>
                    <complained>0</complained>
                    <hardbounced>434</hardbounced>
                    <invalid>0</invalid>
                    <subscribed>842</subscribed>
                    <subscribed_sendable>842</subscribed_sendable>
                    <suspended>0</suspended>
                    <unconfirmed>0</unconfirmed>
                    <unconfirmed_sendable>0</unconfirmed_sendable>
                    <unsubscribed>60</unsubscribed>
                </status>
```

```
                </status>
            </channels>
            <in_progress>false</in_progress>
            <last_count_time>counted at 2024-04-19T13:31:08Z in
231.374251ms</last_count_time>
            <list_id>1</list_id>
            <refresh_task_id></refresh_task_id>
            <updated>2024-04-19T13:31:08.927Z</updated>
        </result>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
 </xml>
```

Добавить сегмент

Описание

Добавить новый сегмент.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/segments/add/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
name	string	“New segment”	Да	Название сегмента
list_id	int	5	Да, для статических сегментов	Идентификатор базы данных, из которой будут взяты профили
segment	JSON array		Да, для динамических и обновляемых сегментов	Условия сегментации. Массив должен содержать хотя бы один объект. Не используется для статических сегментов.

Тип сегмента определяется сочетанием параметров:

- `is_static` равен `false` — сегмент **динамический**
- `is_static` равен `true` и `is_refresh_daily` равен `false` — сегмент **статический**
- `is_static` равен `true` и `is_refresh_daily` равен `true` — сегмент **обновляемый**

Пример запроса

Для статического сегмента

JSON

```
{
    "name": "Segment 2",
    "is_static": true,
    "list_id": 5,
    "is_refresh_daily": false,
    "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <list_id>6</list_id>
  <name>New segment 2</name>
  <is_static>true</is_static>
  <is_refresh_daily>false</is_refresh_daily>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Для динамического сегмента

При создании динамического сегмента наличие правил сегментации в поле `segment` обязательно.

JSON

```
{
  "name": "Dynamic segment",
  "is_static": false,
  "segment": [
    {
      "condition": "and",
      "limit": 0,
      "list_id": 1,
      "percent_from": 0,
      "percent_to": 99,
      "queries": [
        {
          "compo": "default",
          "name": "statseg",
          "operator": "once",
          "type": "statseg",
          "value": [
            {
                "name": "segment_id",
                "value": 1
            }
          ]
        },
        {
          "compo": "default",
          "name": "statseg",
          "operator": "once",
          "type": "statseg",
          "value": [
            {
                "name": "segment_id",
                "value": 2
            }
          ]
        }
      ]
    }
  ],
  "token": "abcdefghijklmnqrstuvwxyz"
}
```


XML

```
<xml>
    <name>Dynamic segment</name>
    <is_static>false</is_static>
    <segment array="true">
        <condition>and</condition>
        <limit>0</limit>
        <list_id>1</list_id>
        <percent_from>0</percent_from>
        <percent_to>99</percent_to>
        <queries>
            <compo>default</compo>
            <name>statseg</name>
            <operator>once</operator>
            <type>statseg</type>
            <value array="true">
                <name>segment_id</name>
                <value>1</value>
            </value>
        </queries>
        <queries>
            <compo>default</compo>
            <name>statseg</name>
            <operator>once</operator>
            <type>statseg</type>
            <value array="true">
                <name>segment_id</name>
                <value>2</value>
            </value>
        </queries>
    </segment>
    <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Для обновляемого сегмента

При создании обновляемого сегмента наличие правил сегментации в поле `segment` обязательно.

JSON

```
{
  "name": "Updatable segment",
  "is_static": true,
  "is_refresh_daily": true,
  "segment": [
    {
      "condition": "and",
      "limit": 0,
      "list_id": 1,
      "percent_from": 0,
      "percent_to": 99,
      "queries": [
        {
          "compo": "default",
          "name": "statseg",
          "operator": "once",
          "type": "statseg",
          "value": [
            {
                "name": "segment_id",
                "value": 1
            }
          ]
        },
        {
          "compo": "default",
          "name": "statseg",
          "operator": "once",
          "type": "statseg",
          "value": [
            {
                "name": "segment_id",
                "value": 2
            }
          ]
        }
      ]
    }
  ],
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <name>New segment 2</name>
  <is_static>true</is_static>
  <is_refresh_daily>true</is_refresh_daily>
  <list_id>1</list_id>
  <segment array="">
    <condition>and</condition>
    <limit>0</limit>
    <list_id>1</list_id>
    <percent_from>0</percent_from>
    <percent_to>99</percent_to>
    <queries array="">
      <compo>domaingroup</compo>
      <name>email</name>
      <operator>equal</operator>
      <type>email</type>
      <value array="">
        <name>domaingroup</name>
        <value>Yahoo! (US)</value>
      </value>
    </queries>
  </segment>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "saved_id": 1
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <saved_id>1</saved_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор нового сегмента

Обновить сегмент

Описание

Изменить правила построения сегмента.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/segments/update/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Да	Формат возвращаемых данных (по умолчанию – “json”)
list_id	int	5	Да, для статических сегментов	Идентификатор базы данных, из которой будут взяты профили
id	int	1	Да	Идентификатор сегмента
name	string	“New segment”	Да	Название сегмента
segment	JSON array		Да, для динамических и обновляемых сегментов	Условия сегментации. Массив должен содержать хотя бы один объект. Не используется для статических сегментов.

Пример запроса

TIP

При обновлении условий сегментации заданную базу профилей изменить **невозможно**.

Для статического сегмента

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz"
    "id": 2,
    "list_id":5
    "name": "Static segment 2",
    "is_static": true,
    "is_refresh_daily": false
}
```

XML

```
<xml>
  <id>2</id>
  <list_id>5</list_id>
  <name>New segment 2</name>
  <is_static>true</is_static>
  <is_refresh_daily>false</is_refresh_daily>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Для динамического сегмента

При изменении правил построения динамического сегмента наличие правил сегментации в поле `segment` обязательно.

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 14,
    "list_id": 6,
    "name": "Static segment 2",
    "is_static": false,
    "segment": [
        {
            "condition": "and",
            "limit": 0,
            "percent_from": 1,
            "list_id": 6,
            "percent_to": 99,
            "queries": [
                {
                    "compo": "domaingroup",
                    "name": "email",
                    "operator": "equal",
                    "type": "email",
                    "value": [
                        {
                            "name": "domaingroup",
                            "value": "Yahoo! (US)"
                        }
                    ]
                }
            ]
        }
    ]
}
```


XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>14</id>
  <name>New segment 2</name>
  <is_static>false</is_static>
  <segment array="">
    <list_id>6</list_id>
    <condition>and</condition>
    <limit>0</limit>
    <percent_from>0</percent_from>
    <percent_to>99</percent_to>
    <queries array="">
      <compo>domaingroup</compo>
      <name>email</name>
      <operator>equal</operator>
      <type>email</type>
      <value array="">
        <name>domaingroup</name>
        <value>Yahoo! (US)</value>
      </value>
    </queries>
  </segment>
</xml>
```

Для обновляемого сегмента

При изменении правил построения обновляемого сегмента наличие правил сегментации в поле `segment` обязательно.

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 28,
    "list_id": 6,
    "name": "Static segment 2",
    "is_static": true,
    "is_refresh_daily": true,
    "segment": [
        {
            "condition": "and",
            "limit": 0,
            "percent_from": 1,
            "list_id": 6,
            "percent_to": 99,
            "queries": [
                {
                    "compo": "domaingroup",
                    "name": "email",
                    "operator": "equal",
                    "type": "email",
                    "value": [
                        {
                            "name": "domaingroup",
                            "value": "Yahoo! (US)"
                        }
                    ]
                }
            ]
        }
    ]
}
```


XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>28</id>
  <name>New segment 2</name>
  <is_static>true</is_static>
  <is_refresh_daily>true</is_refresh_daily>
  <segment array="">
    <list_id>6</list_id>
    <condition>and</condition>
    <limit>0</limit>
    <percent_from>0</percent_from>
    <percent_to>99</percent_to>
    <queries array="">
      <compo>domaingroup</compo>
      <name>email</name>
      <operator>equal</operator>
      <type>email</type>
      <value array="">
        <name>domaingroup</name>
        <value>Yahoo! (US)</value>
      </value>
    </queries>
  </segment>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 1
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>14</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор сегмента

Получить информацию о сегменте

Описание

Получить подробную информацию о сегменте.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/segments/get/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Нет	Идентификатор сегмента

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>10</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "count": 100,
        "id": 1,
        "name": "Segment A",
        "type": "dynamic_segment",
        "list_id": 1,
        "segment": [
            {
                "condition": "or",
                "limit": 0,
                "percent_from": 0,
                "percent_to": 99,
                "queries": [
                    {
                        "child": [],
                        "compo": "domaingroup",
                        "condition": "",
                        "name": "email",
                        "operator": "equal",
                        "type": "email",
                        "value": [
                            {
                                "name": "domaingroup",
                                "value": "Yahoo! (US)"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
    <data>
        <count>0</count>
        <description>""</description>
        <id>14</id>
        <name>"Test segment1"</name>
        <type>"dynamic_segment"</type>
        <list_id>1</list_id>
        <segment>
            <condition>"or"</condition>
            <limit>0</limit>
            <percent_from>0</percent_from>
            <percent_to>99</percent_to>
            <queries>
                <compo>"default"</compo>
                <condition>""</condition>
                <name>"custom_tags"</name>
                <object_path>""</object_path>
                <object_type>""</object_type>
                <operator>"in_data"</operator>
                <type>"subscription_email"</type>
                <value>
                    <name>"url"</name>
                    <value>"http:127.0.0.1:8080/json.email.subscriptions"</value>
                </value>
                <value>
                    <name>"field"</name>
                    <value>"subscriptions"</value>
                </value>
                <value>
                    <name>"resource_id"</name>
                    <value>"1"</value>
                </value>
            </queries>
        </segment>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Параметр	Тип	Описание
count	int	Количество профилей в сегменте
id	int	Идентификатор сегмента
list_id	int	Идентификатор базы профилей
name	string	Название сегмента
type	string	Тип сегмента
segment	JSON array	Правила сегментации
descr	string	Описание сегмента

Получить список сегментов

Описание

Получить список сегментов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/segments/list/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
from_id	int	1	Нет	Идентификатор первого сегмента в списке (по умолчанию – 1)
limit	int	1	Нет	Лимит на количество сегментов в списке (по умолчанию – 1000)
ui_tags	[string]	[“tag1”, “tag2”]	Нет	Список тегов для поиска (по умолчанию – `[]`)
groups	[string]	[“group1”, “group2”]	Нет	Список групп для поиска (по умолчанию – `[]`)
groups_id	[int]	[0, 1]	Нет	Список Id групп для поиска (по умолчанию – `[]`)

Пример запроса

JSON

```
{
   "token": "abcdefghijklmnqrstuvwxyz",
   "limit": 2
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
   <limit>1</limit>
</xml>
```

Пример ответа

JSON

```
{
   "data": [
      {
         "id": 1,
         "name": "Segment A",
         "ui_tags": [
            "tag1",
            "tag2"
         ],
         "list_id": 1,
         "type": "updatable_segment",
         "Groups": [
            {
               "Id": 1,
               "Name": "subaccount"
            }
         ]
      },
      {
         "id": 2,
         "name": "Segment B",
         "ui_tags": [

         ],
         "list_id": 2,
         "type": "static_segment",
         "Groups": [
            {
               "Id": 0,
               "Name": "main"
            }
         ]
      }
   ],
   "error": 0,
   "error_text": "Successful operation",
   "next_from_id": 2
   "total_count": 2
}
```


XML

```
<xml>
    <data>
        <groups>
            <id>1</id>
            <name>"subaccount"</name>
        </groups>
        <id>2</id>
        <name>"Женщины в возрасте 35-60"</name>
        <list_id>1</list_id>
        <type>"updatable_segment"</type>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <next_from_id>5</next_from_id>
    <total_count>9</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	int	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории
total_count	int	Общее количество сегментов
id	int	Идентификатор сегмента
list_id	int	Идентификатор базы профилей
name	string	Название сегмента
ui_tags	[string]	Список тегов
type	string	Тип сегмента
groups	[string]	Список групп

Удалить сегмент

Описание

Удалить сегмент.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/segments/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор сегмента

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>10</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить статистику по сегментам

Описание

Получить статистику подписок профилей в сегменте.

> **INFO: Обратите внимание**
>
> Методы `get_counters` и `refresh_counters` работают в связке. Статистика по сегменту **не обновляется автоматически** — сначала необходимо вызвать метод `refresh_counters`, который инициирует пересчет данных и возвращает ID задания. После успешного выполнения задания (статус можно проверить через `tasks/status`) обновленные данные станут доступны в методе `get_counters`.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/segments/get_counters`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
list_id	int	3	Да, для динамического сегмента	Идентификатор базы данных **Параметр используется только для динамического сегмента**
resource_id	int	2	Да	Идентификатор ресурса
id	int	1	Да	Идентификатор сегмента
format	int	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "list_id": 1,
    "resource_id": 24,
    "id": 7,
    "format": "json"
}
```

XML

```
<xml>
     <token>abcdefghijklmnqrstuvwxyz</token>
     <list_id>1</list_id>
     <resource_id>24</resource_id>
     <id>7</id>
     <format>json</format>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "channel": "email",
            "status": {
                "subscribed_sendable": 209,
                "unconfirmed_sendable": 0
            }
        },
        {
            "channel": "push",
            "status": {
                "subscribed_sendable": 0,
                "unconfirmed_sendable": 0
            }
        },
        {
            "channel": "sms",
            "status": {
                "subscribed_sendable": 108,
                "unconfirmed_sendable": 0
            }
        }
    ],
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
     <data>
         <channel>email</channel>
         <status>
             <subscribed_sendable>209</subscribed_sendable>
             <unconfirmed_sendable>0</unconfirmed_sendable>
         </status>
     </data>
     <data>
         <channel>push</channel>
         <status>
             <subscribed_sendable>0</subscribed_sendable>
             <unconfirmed_sendable>0</unconfirmed_sendable>
         </status>
     </data>
     <data>
         <channel>sms</channel>
         <status>
             <subscribed_sendable>108</subscribed_sendable>
             <unconfirmed_sendable>0</unconfirmed_sendable>
         </status>
     </data>
     <error>0</error>
     <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	array of objects	Массив полученных данных по подпискам сегмента
error	int	Код ошибки
error_text	string	Текст ошибки

Обновить статистику по сегментам

Описание

Обновить статистику подписок профилей в сегменте.

Возвращает ответ с ID задания, не дожидаясь выполнения.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/segments/refresh_counters`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
list_id	int	3	Да, для динамического сегмента	Идентификатор базы данных Параметр используется только для динамического сегмента
resource_id	int	2	Да	Идентификатор ресурса
id	int	1	Да	Идентификатор сегмента
format	int	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "list_id": 1,
    "resource_id": 24,
    "id": 7,
    "format": "json"
}
```

XML

```
<xml>
     <token>abcdefghijklmnqrstuvwxyz</token>
     <list_id>1</list_id>
     <resource_id>24</resource_id>
     <id>7</id>
     <format>json</format>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "task_id": "55a345c8-3d79-47ef-b03e-b84ce8e0c126"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
     <data>
         <task_id>55a345c8-3d79-47ef-b03e-b84ce8e0c126</task_id>
     </data>
     <error>0</error>
     <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	JSON objects	Содержит идентификатор задания – task_id
error	int	Код ошибки
error_text	string	Текст ошибки

Проверка статуса задания

URL-адрес

Метод: POST

`https://example.com/api/v1.1/tasks/status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
task_id	string	“518f4997-ec43-4df5-bfd0-093aaacf679f”	Да	Идентификатор задания

Пример запроса

JSON

```
{
    "task_id": "930bfdb0-a5a8-4207-a77d-6b989bdea326",
    "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
     <task_id>930bfdb0-a5a8-4207-a77d-6b989bdea326</task_id>
     <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "status": "completed",
        "progress": 100,
        "result": [
            {
                "channels": [
                    {
                        "channel": "email",
                        "status": {
                            "subscribed_sendable": 209,
                            "unconfirmed_sendable": 0
                        }
                    },
                    {
                        "channel": "push",
                        "status": {
                            "subscribed_sendable": 0,
                            "unconfirmed_sendable": 0
                        }
                    },
                    {
                        "channel": "sms",
                        "status": {
                            "subscribed_sendable": 0,
                            "unconfirmed_sendable": 0
                        }
                    }
                ],
                "in_progress": false,
                "list_id": 1,
                "refresh_task_id": "",
                "resource_id": 27,
                "updated": "2024-04-20T09:30:01.583Z"
            }
        ]
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
     <data>
         <status>completed</status>
         <progress>100</progress>
         <result>
             <channels>
                 <channel>email</channel>
                 <status>
                     <subscribed_sendable>209</subscribed_sendable>
                     <unconfirmed_sendable>0</unconfirmed_sendable>
                 </status>
             </channels>
             <channels>
                 <channel>push</channel>
                 <status>
                     <subscribed_sendable>0</subscribed_sendable>
                     <unconfirmed_sendable>0</unconfirmed_sendable>
                 </status>
             </channels>
             <channels>
                 <channel>sms</channel>
                 <status>
                     <subscribed_sendable>0</subscribed_sendable>
                     <unconfirmed_sendable>0</unconfirmed_sendable>
                 </status>
             </channels>
             <in_progress>false</in_progress>
             <list_id>1</list_id>
             <refresh_task_id></refresh_task_id>
             <resource_id>27</resource_id>
             <updated>2024-04-20T09:30:01.583Z</updated>
         </result>
     </data>
     <error>0</error>
     <error_text>Successful operation</error_text>
</xml>
```

Добавить или удалить профиль

Описание

Добавляет профиль в **статический сегмент** или удаляет профиль по идентификатору (`profile_id`).

URL-адрес

Метод: POST

Добавить профиль в сегмент: `https://example.com/api/v1.1/profiles/segstatic/add`

Удалить профиль из сегмента: `https://example.com/api/v1.1/profiles/segstatic/remove`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
segment_id	int	1	Да	Идентификатор статического сегмента
db_id	int	1	Да	Идентификатор Базы профилей
profile_id	String	“abcdefghijklmnqrstuvwxyz”	Да	Идентификатор профиля в Базе данных
skip_triggers	bool	true	Нет	Отключает настроенные на события триггеры и цепочки (по умолчанию – false)

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "segment_id": 2,
  "db_id": 8,
  "profile_id": "abcdefghijklmnqrstuvwxyz",
  "skip_triggers": true
}
```

XML

```
<xml>
  <token>0</token>
  <segment_id>2</segment_id>
  <db_id>8</db_id>
  <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
  <skip_triggers>true</skip_triggers>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить данные профилей статического или обновляемого сегмента

Описание

Запрос для получения данных профилей статического или обновлемого сегмента постранично.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/profiles/segstatic/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор статического сегмента
from_id	string	“abcdefghijklmnqrstuvwxyz”	Нет	Идентификатор первого профиля для выгрузки
from_db	int	1	Да для from_id	Идентификатор Базы профилей первого профиля
count	int	20	Нет	Количество записей на странице
fields	[string]	`[“custom_field_1”]`	Нет	Список полей для извлечения из профиля, если пусто — никакие поля не извлекаются

Пример запроса

JSON

```
{
   "token": "abcdefghijklmnqrstuvwxyz",
   "id": 13,
   "from_id": "abcdefghijklmnqrstuvwxyz",
   "from_db": 3,
   "count": 25,
   "fields": ["email"]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>33</id>
    <from_id>abcdefghijklmnqrstuvwxyz</from_id>
    <from_db>75</from_db>
    <count>25</count>
    <fields array="true">email</fields>
</xml>
```

Пример ответа

JSON

```
{
   "next_id": 26,
   "data": [  {
       "profile_id": "abcdefghijklmnqrstuvwxyz",
       "db_id": 3,
       "xxh": "abcdefghijklmxyz",
       "captured": "ISODateString",
       "profile": {profile Object}
   }
]}
```

XML

```
<xml>
  <next_id>26</next_id>
  <data>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
    <db_id>3</db_id>
    <xxh>abcdefghijklmxyz</xxh>
    <captured>ISODateString</captured>
    <profile></profile>
  </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_id	int	Следующий профиль в запросе
data	JSON object	Поля профиля клиента

Добавить стоп-список

Описание

Добавить стоп-список.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
name	string	“New suppression list”	Да	Название стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "name": "New suppression list"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <name>New suppression list</name>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 6
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>7</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор нового стоп-списка

Переименовать стоп-список

Описание

Переименовать стоп-список.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/update`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка
name	string	“Suppression list v.2”	Да	Название стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 2,
    "name": "New suppression list"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>2</id>
  <name>Suppression list v.2</name>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 2
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>2</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить информацию о стоп-списке

Описание

Получить информацию о стоп-списке.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 2
}
```

XML

```
<xml>
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "id": 3,
        "name": "New suppression list",
        "is_global": false,
        "description": ""
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <description>""</description>
        <id>1</id>
        <is_global>true</is_global>
        <name>"Global suppress list"</name>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор стоп-списка
name	string	Название стоп-списка
is_global	bool	true, если это глобальный список
descr	string	Описание стоп-списка

Получить информацию о нескольких стоп-списков

Описание

Получить информацию о нескольких стоп-списков.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
from_id	int	1	Нет	Идентификатор первого стоп-списка в ответе (по умолчанию – 1)
limit	int	5	Нет	Лимит на количество возвращаемых данных (по умолчанию – 1000)
ui_tags	[]String	[“tag1”, “tag2”]	Нет	Список тегов для поиска (по умолчанию – `[]`)
groups	[]String	[“group1”, “group2”]	Нет	Список групп для поиска (по умолчанию – `[]`)
groups_id	[]int	[0, 1]	Нет	Список Id групп для поиска (по умолчанию – `[]`)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "from_id": 1,
    "limit":2
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <from_id>1</from_id>
    <limit>2</limit>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "id": 1,
            "name": "Global suppress list",
            "ui_tags": [],
            "groups": [
                {
                    "id": 0,
                    "name": "main"
                }
            ]
        },
        {
            "id": 3,
            "name": "New suppression list",
            "ui_tags": [],
            "groups": [
                {
                    "id": 0,
                    "name": "main"
                }
            ]
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 5
}
```

XML

```
<xml>
    <data>
        <groups>
            <id>0</id>
            <name>"main"</name>
        </groups>
        <id>1</id>
        <name>"Global suppress list"</name>
    </data>
    <data>
        <groups>
            <id>0</id>
            <name>"main"</name>
        </groups>
        <id>3</id>
        <name>"New suppression list"</name>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>5</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор стоп-списка
name	string	Название стоп-списка
total_count	int	Общее количество стоп-списков
ui_tags	[]String	Список тегов
groups	[]String	Список групп

Удалить стоп-список

Описание

Удалить стоп-список.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 2
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>4</id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Выгрузить данные из стоп-списка в файл

Описание

Загрузка csv-файла с данными из стоп-списка в формате md5. Для получения информации необходимо выполнить несколько запросов: выгрузку данных в файл, проверку статуса задания и загрузку файла.

Шаг 1. Выгрузка данных в файл

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/export`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор стоп-списка
type	string	“emails” “phones” “domains” “all”	Нет	Тип контактных данных для экспорта По умолчанию — all

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1,
    "type": "emails"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>3</id>
  <type>emails</type>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "task_id": "c110f0-06d0-4695-9aee-a9bab742d"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <task_id>c110f0-06d0-4695-9aee-a9bab742d</task_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
task_id	string	Идентификатор задания на экспорт
error	int	Код ошибки
error_text	string	Текст ошибки

Шаг 2. Проверка статуса экспорта и получение ссылки

URL-адрес

Метод: POST

`https://example.com/api/v1.1/tasks/status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
task_id	string	“c110f0-06d0-4695-9aee-a9bab742d”	Да	Идентификатор задания

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "task_id": "c110f0-06d0-4695-9aee-a9bab742d"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <task_id>c110f0-06d0-4695-9aee-a9bab742d</task_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "status": "completed",
        "progress": 100,
        "result": {
            "hashed_file": "suppress/b5bb73d7-530d-4a64-89d4-
8ad22bad73/suppress_list_name.md5.zip",
            "is_api": true,
            "regular_file": "",
            "suppress_id": 1,
            "total": 32
        }
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <status>completed</status>
    <progress>100</progress>
    <xml>
      <hashed_file>suppress/b5bb73d7-530d-4a64-89d4-
8ad22bad73/suppress_list_name.md5.zip</hashed_file>
      <is_api>true</is_api>
      <regular_file></regular_file>
      <suppress_id>1</suppress_id>
      <total>32</total>
    </xml>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
status	string	Статус задания
progress	int	Прогресс выполнения
hashed_file	array	Ссылка на скачивание файла с данными в формате md5
is_api	bool	Экспорт через API
suppress_id	int	Идентификатор стоп-списка
total	int	Количество контактов в стоп-списке
error	int	Код ошибки
error_text	string	Текст ошибки

Шаг 3. Загрузка результата

Ссылка на загрузку файла выглядит подобным образом:

`https://example.com/api/download/suppress/b5bb73d7-530d-4a64-89d4-8ad22bad73/suppress_list_name.md5.zip?token=abcdefghijklmnqrstuvwxyz`

- **`https://example.com`** — ссылка на панель управления платформой
- **`api/download/suppress/b5bb73d7-530d-4a64-89d4-8ad22bad73/suppress_list_name.md5.zip`** — ссылка из параметра `urls` прошлого запроса
- **`?token=abcdefghijklmnqrstuvwxyz`** — API-токен пользователя (после знака равенства).

Для загрузки файла с профилями необходимо либо открыть получившуюся ссылку в браузере, либо отправить пустой **GET** запрос с указанием данной ссылки в URL.

Добавление и удаление из стоп-списка

Email-адрес

- Проверить email-адрес в стоп-списке
- Добавить email-адрес в стоп-список
- Добавить один или несколько email-адресов в стоп-список
- Удалить email-адрес из стоп-списка
- Удалить все email-адреса из стоп-списка

Email-домен

- Проверить домен в стоп-списке
- Добавить домен в стоп-список
- Добавить один или несколько доменов в стоп-список
- Удалить домен из стоп-списка
- Удалить все домены из стоп-списка

Номер телефона

- Проверить номер телефона в стоп-списке
- Добавить номер телефона в стоп-список
- Добавить один или несколько номеров в стоп-список
- Удалить номер из стоп-списка
- Удалить все номера из стоп-списка

Проверить email-адрес в стоп-списке

Описание

Проверить email-адрес в стоп-списке.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/email_check`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
email	string	“john@example.com”	Да	Email-адрес
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "email": "john@example.com",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <email>john@example.com</email>
    <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "domain_matched": false,
    "email_matched": true,
    "error": 0,
    "error_text": "Successful operation",
    "matched": true,
    "phone_matched": false
}
```

XML

```
<xml>
    <domain_matched>false</domain_matched>
    <email_matched>true</email_matched>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <matched>true</matched>
    <phone_matched>false</phone_matched>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
domain_matched	bool	true, если домен найден в списке доменов
email_matched	bool	true, если домен найден в списке email-адресов
matched	bool	true, если email найден в списке email-адресов или доменах

Добавить email-адрес в стоп-список

Описание

Добавить адрес электронной почты в стоп-список. Позволяет добавить адрес запросом без JSON структур.

URL-адрес

Методы: GET, POST

`https://example.com/api/v1.1/suppresses/email_add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
email	string	“john@example.com”	Да	Email-адрес
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "email": "john@example.com",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <email>john@example.com</email>
    <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "john@example.com": true
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <john@example.com>false</john@example.com>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Добавить один или несколько email-адресов в стоп-список

Описание

Добавить один или несколько email-адресов в стоп-список. Использует JSON структуры.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/emails_add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
emails	JSON array	`["john@example.com"]`	Да	Список из email-адресов
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "emails": [
        "example@example.com",
        "example1@example.com",
        "example2@example.com"
    ],
    "id": 3
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <emails>example@example.com</emails>
  <emails>example1@example.com</emails>
  <emails>example2@example.com</emails>
  <id>3</id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 3,
    "total_duplicated": 0,
    "total_imported": 3,
    "total_invalid": 0
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>3</total_count>
    <total_duplicated>0</total_duplicated>
    <total_imported>3</total_imported>
    <total_invalid>0</total_invalid>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
total_count	int	Всего адресов
total_duplicated	int	Адреса, добавленные в стоп-список ранее
total_imported	int	Адресов добавлено

Параметр	Тип	Описание
total_updated	int	Адресов обновлено (при совпадении)

Удалить email-адрес из стоп-списка

Описание

Удалить email-адрес из стоп-списка.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/email_remove`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	12	Да	Идентификатор стоп-списка
email	string	“example@example.com”	Да	Email-адрес

CAUTION

Вы можете использовать MD5-хеш email-адреса.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "email": "john@example.com",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <email>john@example.com</email>
    <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Удалить все email-адреса из стоп-списка

Описание

Удалить все email-адреса из стоп-списка.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/emails_remove`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "removed_count": 1000
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <removed_count>2</removed_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
remove_count	int	Количество удаленных email-адресов

Проверить домен в стоп-списке

Описание

Проверить домен в стоп-списке.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/domain_check`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка
domain	string	“example.com”	Да	Домен

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "domain": "example.com",
    "id": 1
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <domain>example.com</domain>
  <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "matched": true
}
```

XML

```
<xml>
    <domain_matched>true</domain_matched>
    <email_matched>false</email_matched>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <matched>true</matched>
    <phone_matched>false</phone_matched>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
matched	bool	true, если домен найден в стоп-списке

Добавить домен в стоп-список

Описание

Добавить домен в стоп-список.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/domain_add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
domain	string	“example.com”	Да	Домен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "domain": "example.com",
    "id": 1
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <domain>example.com</domain>
  <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "example.com": true
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <example.com>false</example.com>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Добавить один или несколько доменов в стоп-список

Описание

Добавить один или несколько доменов в стоп-список.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/domains_add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
domains	JSON array	`["example.com"]`	Да	Список доменов
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "domains": [
        "example.com",
        "example1.com"
    ],
    "id": 3
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <domains>example1.com</domains>
  <domains>example2.com</domains>
  <domains>example3.com</domains>
  <id>3</id>
</xml>
```

Пример ответа

JSON

```json
{
  "error": 0,
  "error_text": "Successful operation",
  "total_count": 3,
  "total_duplicated": 0,
  "total_imported": 3,
  "total_invalid": 0
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>3</total_count>
    <total_duplicated>0</total_duplicated>
    <total_imported>3</total_imported>
    <total_invalid>0</total_invalid>
</xml>
```

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <total_count>3</total_count>
  <total_duplicated>0</total_duplicated>
  <total_imported>3</total_imported>
  <total_invalid>0</total_invalid>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
total_count	int	Всего доменов
total_duplicated	int	Домены, добавленные в стоп-список ранее

Параметр	Тип	Описание
total_imported	int	Доменов добавлено
total_updated	int	Доменов обновлено при совпадении

Удалить домен из стоп-списка

Описание

Удалить домен из стоп-списка.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/domain_remove`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка
domain	string	“example.com”	Да	Домен

CAUTION

Вы можете использовать MD5-хеш домена.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "domain": "example.com",
    "id": 3
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <domain>test.com</domain>
  <id>3</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Удалить все домены из стоп-списка

Описание

Удалить все домены из стоп-списка.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/domains_remove`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 3
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>3</id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "removed_count": 100
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <removed_count>100</removed_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
removed_count	int	Количество удаленных доменов

Проверить номер телефона в стоп-списке

Описание

Проверить номер телефона в стоп-списке.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/phone_check`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
value	string	“+79000000001”	Да	Номер телефона
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 1,
  "value":"+79000000001"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>1</id>
    <value>+79000000001</value>
</xml>
```

Пример ответа

JSON

```
{
    "domain_matched": false,
    "email_matched": false,
    "error": 0,
    "error_text": "Successful operation",
    "matched": true,
    "phone_matched": true
}
```

XML

```
<xml>
    <domain_matched>false</domain_matched>
    <email_matched>false</email_matched>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <matched>true</matched>
    <phone_matched>true</phone_matched>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
domain_matched	bool	true, если значение найдено в списке доменов
email_matched	bool	true, если значение найдено в списке email-адресов
matched	bool	true, если значение найдено в списке email-адресов, доменов или телефонных номеров
phone_matched	bool	true, если значение найдено в списке телефонных номеров

Добавить номер телефона в стоп-список

Описание

Добавить номер телефона в стоп-список.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/phone_add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
value	string	“+79000000001”	Да	Номер телефона
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 1,
  "value":"+79000000001"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>1</id>
    <value>+79000000001</value>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "+79000000991": true
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data><+79000000001>false</+79000000001>
</data>
<error>0</error>
<error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
data	obj	Номер телефона, добавленный в стоп-список

Добавить один или несколько номеров в стоп-список

Описание

Добавить один или несколько телефонных номеров в стоп-список.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/phones_add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
values	JSON array	[“+79009009001”, “+79009009002”]	Да	Список номеров
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1,
    "values": [
        "+79000000001",
        "+79999999991"
    ]
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
  <values>+79000000001</values>
  <values>+79999999991</values>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "total_count": 2,
  "total_duplicated": 0,
  "total_imported": 2,
  "total_invalid": 0
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>2</total_count>
    <total_duplicated>1</total_duplicated>
    <total_imported>1</total_imported>
    <total_invalid>0</total_invalid>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
total_count	int	Всего номеров
total_duplicated	int	Номера, добавленные в стоп-список ранее
total_imported	int	Номеров добавлено

Параметр	Тип	Описание
total_invalid	int	Количество невалидных номеров

Удалить номер из стоп-списка

Описание

Удалить номер телефона из стоп-списка.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/phone_remove`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка
value	string	“+79009009001”	Да	Номер телефона

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1,
    "value": "+79999999991"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
  <value>+79999999991</value>
</xml>
```

Пример ответа

JSON

```
{
  "data": null,
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Удалить все номера из стоп-списка

Описание

Удалить все телефонные номера из стоп-списка.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/suppresses/phones_remove`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор стоп-списка

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "removed_count": 500
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <removed_count>4</removed_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
removed_count	int	Количество удаленных номеров

Получить список шаблонов

Описание

Получить список всех шаблонов сообщений.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/templates/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
from_id	int	1	Нет	Идентификатор первого шаблона в ответе (по умолчанию – 1)
limit	int	100	Нет	Лимит на количество возвращаемых данных (по умолчанию – 1000)
channel	string	“email” “sms” “push” “telegram_bot” “telegram_group” “custom_channel”	Нет	Канал, в котором используется шаблон (по умолчанию – все каналы)
ui_tags	[string]	`["tag1", "tag2"]`	Нет	Список тегов для поиска (по умолчанию – `[]`)
groups	[string]	`["group1", "group2"]`	Нет	Список групп для поиска (по умолчанию – `[]`)
groups_id	[int]	`[0, 1]`	Нет	Список Id групп для поиска (по умолчанию – `[]`)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
          "limit": 2
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <limit>2</limit>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "id": 1,
            "updated_user": "ivan_ivanov",
            "update_time": "2025-05-21T09:17:53.362Z",
            "name": "Message A",
                        "channels": [
                "email"
            ]
                },
        {
            "id": 2,
            "name": "Message B",
                        "channels": [
                "push"
            ]
                }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 10
}
```

XML

```
<xml>
  <data>
    <id>1</id>
    <updated_user>ivan_ivanov</updated_user>
    <update_time>2025-05-21T09:17:53.362Z</update_time>
    <name>Message A</name>
    <channels>email</channels>
  </data>
  <data>
    <id>2</id>
    <name>Message B</name>
    <channels>push</channels>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <total_count>10</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
total_count	int	Общее количество шаблонов
id	int	Идентификатор шаблона сообщения
name	string	Название шаблона сообщения
channels	array	Каналы, для которых настроен шаблон
is_draft	bool	Если это черновик, то шаблон нельзя отправить
is_confirm	bool	Имеется ли ссылка подтверждения в теле письма
ui_tags	[]String	Список тегов
groups	array of objects	Список групп (id группы + название группы)
updated_user	string	Кем был обновлён шаблон
update_time	string	Когда был обновлён шаблон

Получить информацию о шаблоне

Описание

Получить информацию о шаблоне сообщения.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/templates/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор шаблона сообщения
channel_sid	array of strings	[“email”, “sms”, “cc_channel”]	Нет	Фильтр по каналам связи. Если указан, возвращаются только шаблоны для указанных каналов. По умолчанию - пустой массив (все каналы)
get_variable	boolean	true	Нет	Если true, в ответ включается список переменных (params), используемых в шаблоне. По умолчанию - false

Доступные типы переменных

Данные профиля

`lead.fname`, `lead.lname`, `lead.email`, `lead.phone`

`lead.birthdate`, `lead.gender`, `lead._city`, `lead._country`

`lead._xxh`, `lead_custom`.

Параметры

`campid`, `workflow_id`, `workflow_name`, `msg_type`

`attribute_value`, `node_id`

Даты и время

`datenow`, `yearlong`, `hourse`, `minutes`, `ampm`

Переменные apicontent

`apicontent.field_1`, `apicontent.field_2`, `apicontent.field_N`

Сегментация

`segname`

Подробнее про переменные, используемые в шаблонах, можно узнать в этой статье.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1,
    "channel_sid": ["email", "sms", "cc_channel"],
    "get_variable": true
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
  <channel_sid>email</channel_sid>
  <channel_sid>sms</channel_sid>
  <channel_sid>cc_channel</channel_sid>
  <get_variable>true</get_variable>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "id": 1,
        "updated_user": "example_user",
        "update_time": "2025-05-21T09:17:53.362Z",
        "name": "example template name",
        "description": "example template description",
        "channels": [
            {
                "id": 1,
                "sid": "email",
                "fields": {
                    "text": "example text version",
                    "html": "example html content",
                    "subject": "example subject",
                    "from_email": "example@example.com",
                    "from_name": "example from name",
                    "replyto_email": "reply@example.com",
                    "replyto_name": "example reply name",
                    "is_not_inline": true,
                    "attach": []
                },
                "params": [
                    "lead.fname",
                    "datenow",
                    "segname",
                    "apicontent.field_name",
                    "workflow_name"
                ],
                "presets": null
            },
            {
                "id": 2,
                "sid": "sms",
                "fields": {
                    "sms": "example sms content",
                    "sms_max_count": 0,
                    "add_sms_url_scheme": false
                },
                "params": [
                    "lead.fname",
                    "datenow",
                    "segname",
                    "apicontent.field_name",
                    "workflow_name"
                ],
                "presets": null
            }
        ]
```

```
        ],
        "json_url": "https://example.com/examples/message.json",
        "is_draft": true,
        "is_confirm": true,
        "ui_tags": [
            "example_tag_1",
            "example_tag_2"
        ],
        "groups": [
            {
                "id": 2,
                "name": "example_group_1"
            },
            {
                "id": 3,
                "name": "example_group_2"
            }
        ],
        "suppress_id": 1
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <id>1</id>
    <updated_user>example_user</updated_user>
    <update_time>2025-05-21T09:17:53.362Z</update_time>
    <name>example template name</name>
    <description>example template description</description>
    <channels>
      <id>1</id>
      <sid>email</sid>
      <fields>
        <text>example text version</text>
        <html>example html content</html>
        <subject>example subject</subject>
        <from_email>example@example.com</from_email>
        <from_name>example from name</from_name>
        <replyto_email>reply@example.com</replyto_email>
        <replyto_name>example reply name</replyto_name>
        <is_not_inline>true</is_not_inline>
        <attach array='true'></attach>
      </fields>
      <params>lead.fname</params>
      <params>datenow</params>
      <params>segname</params>
      <params>apicontent.field_name</params>
      <params>workflow_name</params>
      <presets></presets>
    </channels>
    <channels>
      <id>2</id>
      <sid>sms</sid>
      <fields>
        <sms>example sms content</sms>
        <sms_max_count>0</sms_max_count>
        <add_sms_url_scheme>false</add_sms_url_scheme>
      </fields>
      <params>lead.fname</params>
      <params>datenow</params>
      <params>segname</params>
      <params>apicontent.field_name</params>
      <params>workflow_name</params>
    </channels>
    <json_url>https://example.com/examples/message.json</json_url>
    <is_draft>true</is_draft>
    <is_confirm>true</is_confirm>
    <ui_tags>example_tag_1</ui_tags>
    <ui_tags>example_tag_2</ui_tags>
    <groups>
      <id>2</id>
```

```
      <id>2</id>
      <name>example_group_1</name>
    </groups>
    <groups>
      <id>3</id>
      <name>example_group_2</name>
    </groups>
    <suppress_id>1</suppress_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор шаблона сообщения
name	string	Название шаблона сообщения
description	string	Описание шаблона
channels	array of objects	Данные о настройках шаблона для каждого канала
channels[].id	int	Идентификатор канала в шаблоне
channels[].sid	string	Системный идентификатор канала (email, sms, push и т.д.)
channels[].fields	object	Настройки содержимого шаблона для канала
channels[].params	array of strings	Список переменных, используемых в шаблоне. Возвращается только при `get_variable: true` в запросе.
channels[].presets	object|null	Предустановленные настройки канала
json_url	string	URL для динамического JSON-контента
is_draft	bool	Если это черновик, то шаблон нельзя отправить
is_confirm	bool	Имеется ли ссылка подтверждения в теле письма
ui_tags	[]String	Список тегов для поиска
groups	array of objects	Список групп для поиска
suppress_id	int	Идентификатор стоп-списка
updated_user	string	Кем был обновлён шаблон

Параметр	Тип	Описание
update_time	string	Когда был обновлён шаблон

Удалить шаблон

Описание

Удаление шаблона сообщения.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/templates/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор шаблона сообщения
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
        "id": 1
}
```

XML

```
<xml>
  <token>97</token>
  <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": 97,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>97</data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	int	Идентификатор удалённого шаблона
error	int	Код ошибки
error_text	string	Текст ошибки

Добавить шаблон сообщения

Описание

Добавить новый шаблон сообщения.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/templates/add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
name	string	“Шаблон для подтверждения подписки”	Да	Название шаблона
descr	string	“Шаблон используется в сценарии Double Opt-In”	Нет	Описание шаблона
channels	объект	“channels”: [{ “sid”: “email”, “fields”: { “text”: “Текстовая версия письма”, “html”: “Hello, world!”, “subject”: “Тема письма”, “from_email”: “”, “from_name”: “”, “replyto_email”: “”, “replyto_name”: “” } }]	Да	Настройки шаблона для конкретного канала Подробнее
json_url	string	`"https://altcraft.com/public/assets/example/test.json"`	Нет	Ссылка на динамический JSON-контент
ui_tags	[string]	`["tag_1","tag_2"]`	Нет	Список тегов

Параметр	Тип	Пример	Обязательный	Описание
groups	[int]	`[1,2]`	Нет	Список идентификаторов групп доступа
is_draft	bool	false	Нет	Если это черновик, то шаблон нельзя отправить
is_confirm	bool	true	Нет	Имеется ли ссылка подтверждения в теле сообщения
suppress_id	int	2	Нет	Идентификатор стоп-списка

Пример запроса

INFO

В любом шаблоне обязательно должна присутствовать ссылка отписки.

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "name": "Название шаблона",
  "descr": "Описание шаблона",
  "channels": [
    {
      "sid": "email",
      "fields": {
        "text": "Текстовая версия письма",
        "html": "<html>Hello, world!</html>",
        "subject": "Тема письма",
        "from_email": "",
        "from_name": "",
        "replyto_email": "",
        "replyto_name": "",
        "is_not_inline": true
      }
    }
  ],
  "json_url": "https://altcraft.com/example/message.json",
  "ui_tags": ["tag_1", "tag_2"],
  "groups": [1, 2],
  "is_draft": false,
  "is_confirm": false,
  "suppress_id": 2
}
```


XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <name>Template</name>
    <descr>Template description</descr>
    <channels array = 'true'>
        <fields>
            <body>Hello, world!</body>
            <click_url>https://example.com</click_url>
            <icon>https://example.com/images/picture.jpg</icon>
            <title>Welcome</title>
        </fields>
        <sid>push</sid>
    </channels>
    <is_confirm>false</is_confirm>
    <is_draft>false</is_draft>
    <json_url>https://altcraft.com/example/message.json</json_url>
    <suppress_id>2</suppress_id>
    <groups>1</groups>
    <groups>2</groups>
    <ui_tags>tag_1</ui_tags>
    <ui_tags>tag_2</ui_tags>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "saved_id": 10
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <saved_id>10</saved_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	string	Идентификатор созданного шаблона

Обновить шаблон сообщения

Описание

Добавить новый шаблон сообщения.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/templates/update`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор шаблона сообщения
name	string	“Шаблон для подтверждения подписки”	Да	Название шаблона
descr	string	“Шаблон используется в сценарии Double Opt-In”	Нет	Описание шаблона
channels	[]channels	“channels”: [{ “sid”: “email”, “fields”: { “text”: “Текстовая версия письма”, “html”: “Hello, world!”, “subject”: “Тема письма”, “from_email”: “”, “from_name”: “”, “replyto_email”: “”, “replyto_name”: “” } }]	Да	Настройки шаблона для конкретного канала Подробнее
json_url	string	`"https://altcraft.com/public/assets/example/test.json"`	Нет	Ссылка на динамический JSON-контент
ui_tags	[]string	`["tag_1","tag_2"]`	Нет	Список тегов
groups	[]int	`[1,2]`	Нет	Список идентификаторов групп доступа

Параметр	Тип	Пример	Обязательный	Описание
is_draft	bool	false	Нет	Если это черновик, то шаблон нельзя отправить
is_confirm	bool	true	Нет	Имеется ли ссылка подтверждения в теле письма
suppress_id	int	2	Нет	Идентификатор стоп-списка

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 10,
  "name": "Название шаблона",
  "descr": "Описание шаблона",
  "channels": [
    {
      "sid": "email",
      "fields": {
        "text": "Текстовая версия письма",
        "html": "<html>Hello, world!</html>",
        "subject": "Шаблон сообщения",
        "from_email": "",
        "from_name": "",
        "replyto_email": "",
        "replyto_name": "",
        "is_not_inline": true
      }
    }
  ],
  "json_url": "https://altcraft.com/public/assets/example/test.json",
  "ui_tags": ["tag_1", "tag_2"],
  "groups": [],
  "is_draft": false,
  "is_confirm": false,
  "suppress_id": 4
}
```

XML

```
<xml>
  <channels array = 'true'>
    <fields>
      <body>Hello, world!</body>
      <click_url>https://example.com</click_url>
      <icon>https://example.com/images/picture.jpg</icon>
      <title>Welcome</title>
    </fields>
    <sid>push</sid>
  </channels>
  <descr>Template description</descr>
  <groups>1</groups>
  <groups>2</groups>
  <is_confirm>false</is_confirm>
  <is_draft>false</is_draft>
  <json_url>https://altcraft.com/example/message.json</json_url>
  <id>10</id>
  <name>Template</name>
  <suppress_id>2</suppress_id>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <ui_tags>tag_1</ui_tags>
  <ui_tags>tag_2</ui_tags>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "saved_id": 10
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <saved_id>10</saved_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	string	Идентификатор обновленного шаблона

Channel object

Используется в API-запросах, связанных с добавлением и обновлением шаблонов сообщений.

JSON

```
{
        "sid": "email",
        "fields": {
                "text": "Hello, world!",
                "html": "<html>Hello, world!</html>",
                "subject": "Welcome"
        }
},
{
        "sid": "sms",
        "fields": {
                "sms": "Hello, world!"
        }
},
{
        "sid": "push",
        "fields": {
                "title": "Welcome",
                "body": "Hello, world!",
                "icon": "https://example.com/images/picture.jpg",
                "click_url": "https://example.com"
        }
}
```

XML

```
<channels>
  <sid>email</sid>
  <fields>
    <text>Hello, world!</text>
    <html>&lt;html&gt;Hello, world!&lt;/html&gt;</html>
    <subject>Welcome</subject>
  </fields>
</channels>
<channels>
  <sid>sms</sid>
  <fields>
    <sms>Hello, world!</sms>
  </fields>
</channels>
<channels>
  <sid>push</sid>
  <fields>
      <body>Hello, world!</body>
      <click_url>https://example.com</click_url>
      <icon>https://example.com/images/picture.jpg</icon>
      <title>Welcome</title>
  </fields>
</channels>
```

Описание

Параметр	Тип	Пример	Описание
sid	string	“email” “sms” “push” “telegram_bot” “telegram_group” “custom_channel”	Название канала

fields	object	“fields”: {
		“text”: “Hello, world”,
		“html”: “Hello, world!”,
		“subject”: “Welcome”
		}
			Настройки шаблона для конкретного канала

Fields object

Каждый объект `channel` содержит объект `fields`:

Email fields

JSON

```
{
  "sid": "email",
  "fields": {
    "text": "Текстовая версия письма",
    "html": "<html>HTML версия письма</html>",
    "subject": "Тема письма",
    "from_email": "test@example.com",
    "from_name": "Jenny's List",
    "replyto_email": "test_reply@example.com",
    "replyto_name": "Jenny",
    "is_not_inline": true
  }
}
```

XML

```
<xml>
  <sid>email</sid>
  <fields>
    <text>Текстовая версия письма</text>
    <html>&lt;html&gt;HTML версия письма&lt;/html&gt;</html>
    <subject>Тема письма</subject>
    <from_email>test@example.com</from_email>
    <from_name>Jenny&apos;s List</from_name>
    <replyto_email>test_reply@example.com</replyto_email>
    <replyto_name>Jenny</replyto_name>
    <is_not_inline>true</is_not_inline>
  </fields>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
text	string	“Hello, world!”	Текстовая версия письма
html	string	`"<html>Hello, world!</html>"`	HTML версия письма
is_amp	bool	false	Есть ли в шаблоне amp-версия
amp	string	`"<html>Hello, world!</html>"`	AMP HTML версия письма
subject	string	“Weather alert”	Тема сообщения
from_email	string	“test@example.com”	Перезаписать from-email
from_name	string	“Jenny’s List”	Перезаписать from-name
replyto_email	string	“test_reply@example.com”	Перезаписать replyto_email

Параметр	Тип	Пример	Описание
replyto_name	string	“Jenny”	Перезаписать replyto_name
is_not_inline	bool	true	Не конвертировать CSS в inline

SMS fields

JSON

```
{
  "sid": "sms",
  "fields": {
    "sms": "Содержание SMS сообщения",
    "sms_max_count": 0,
    "add_sms_url_scheme": true
  }
}
```

XML

```
<xml>
  <sid>sms</sid>
  <fields>
    <sms>Содержание SMS сообщения</sms>
    <sms_max_count>0</sms_max_count>
    <add_sms_url_scheme>true</add_sms_url_scheme>
  </fields>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
sms	string	“Hello, world!”	Текст сообщения
sms_max_count	int	0	Лимит на количество SMS
add_sms_url_scheme	bool	true	Добавлять “http://” в ссылки

Push fields

JSON

```
{
  "sid": "push",
  "fields": {
    "title": "Заголовок push уведомления",
    "body": "Содержание push уведомления",
    "icon": "https://example.com/images/icon.jpg",
    "click_url": "https://example.com",
    "custom_push_json_raw": "{}"
  }
}
```

XML

```
<xml>
  <sid>push</sid>
  <fields>
    <title>Заголовок push уведомления</title>
    <body>Содержание push уведомления</body>
    <icon>https://example.com/images/icon.jpg</icon>
    <click_url>https://example.com</click_url>
    <custom_push_json_raw>{}</custom_push_json_raw>
  </fields>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
title	string	“Welcome”	Заголовок сообщения
body	string	“Hello, world!”	Текст сообщения
icon	string	“https://example.com/images/test.jpg”	URL-адрес, по которому расположена иконка для уведомления
click_url	string	“https://example.com”	Ссылка для перехода по клику
custom_push_json_raw	string	“{}”	Custom JSON для передачи в мобильный push

Notify fields

JSON

```
{
  "sid": "notify",
  "fields": {
    "template": {
      "id": 1,
      "name": "message_after_purchase",
      "type": "NOTIFY",
      "status": "NOT_SENT",
      "is_draft": true,
      "edna_account_id": "",
      "notify_service_name": "altcraft_testing",
      "vk_data": {
        "body": "Содержание Notify сообщения",
        "buttons": [],
        "attachments": [],
        "enabled": true,
        "inline_keyboard": false,
        "hide_keyboard": false,
        "reply_enabled": false
      },
      "ok_data": {
        "enabled": true,
        "body": "Содержание Notify сообщения"
      },
      "sms_data": {
        "enabled": true,
        "body": "Содержание Notify сообщения",
        "type": "advert"
      },
      "routes": [
        {
          "route": "vk",
          "deliv_timeout": 86400,
          "read_timeout": 86400
        },
        {
          "route": "ok",
          "deliv_timeout": 86400,
          "read_timeout": 86400
        },
        {
          "route": "sms"
        }
      ]
    },
    "info": {
      "is_obsolete": false,
      "not_found": false
    }
```

```
    },
    "params": {
      "name": "{{$.Lead._fname}}",
      "order_number": "{{$.Lead.order_number}}"
    }
  }
}
```

XML

```
<xml>
  <sid>notify</sid>
  <fields>
                <info>
                    <is_obsolete>false</is_obsolete>
                    <not_found>false</not_found>
                </info>
                <params>
                    <name>{{$.Lead._fname}}</name>
                    <order_number>{{$.Lead.order_number}}</order_number>
                </params>
                <template>
                    <id>1</id>
                    <is_draft>true</is_draft>
                    <name>message_after_purchase</name>
                    <notify_service_name>altcraft_testing</notify_service_name>
                    <ok_data>
                        <body>Содержание Notify сообщения</body>
                        <enabled>true</enabled>
                    </ok_data>
                    <routes>
                        <deliv_timeout>86400</deliv_timeout>
                        <read_timeout>86400</read_timeout>
                        <route>vk</route>
                    </routes>
                    <routes>
                        <deliv_timeout>86400</deliv_timeout>
                        <read_timeout>86400</read_timeout>
                        <route>ok</route>
                    </routes>
                    <routes>
                        <route>sms</route>
                    </routes>
                    <sms_data>
                        <body>Содержание Notify сообщения</body>
                        <enabled>true</enabled>
                        <type>advert</type>
                    </sms_data>
                    <status>NOT_SENT</status>
                    <type>NOTIFY</type>
                    <vk_data>
                        <body>Содержание Notify сообщения</body>
                        <enabled>true</enabled>
                        <hide_keyboard>false</hide_keyboard>
                        <inline_keyboard>false</inline_keyboard>
                        <reply_enabled>false</reply_enabled>
                    </vk_data>
                </template>
            </fields>
```

```
            </fields>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
name	string	“message_after_purchase”	Название шаблона
type	string	“NOTIFY”	Тип шаблона
status	string	“NOT_SENT”	Статус отправки шаблона
is_draft	boolean	true	Является ли шаблон черновиком
edna_account_id	string	“123456”	ID аккаунта Edna
notify_service_name	string	“altcraft_testing”	Имя сервиса уведомлений
body	string	“Содержание Notify сообщения”	Текст сообщения для VK
buttons	array	[]	Кнопки в сообщении VK
attachments	array	[]	Вложения в сообщении VK
enabled	boolean	true	Включено ли сообщение VK
inline_keyboard	boolean	false	Используется ли инлайн-клавиатура VK
hide_keyboard	boolean	false	Скрыть ли клавиатуру VK после отправки
reply_enabled	boolean	false	Разрешён ли ответ VK пользователю
enabled	boolean	true	Включено ли сообщение в OK
body	string	“Содержание Notify сообщения”	Текст сообщения для OK
enabled	boolean	true	Включено ли SMS-сообщение
body	string	“Содержание Notify сообщения”	Текст SMS-сообщения
type	string	“advert”	Тип SMS-сообщения
route	string	“vk”	Канал доставки
deliv_timeout	integer	86400	Таймаут доставки (в секундах)
read_timeout	integer	86400	Таймаут ожидания прочтения (в секундах)
route	string	“ok”	Второй канал доставки
route	string	“sms”	Третий канал доставки
is_obsolete	boolean	false	Устарел ли шаблон
not_found	boolean	false	Был ли шаблон найден

Telegram fields

JSON

```
{
  "sid": "telegram_bot",
  "fields": {
    "image": {
      "attach": [
        {
          "content": "/9j/bcvb4234hgdffsd45",
          "filekey": "67863659182463338076 2bee",
          "name": "example.jpg",
          "size": 65101,
          "type": "image_path"
        }
      ]
    },
    "image_caption": "Описание Telegram bot сообщения",
    "markup_style": "HTML"
  }
}
```

XML

```
<xml>
  <sid>telegram_bot</sid>
  <fields>
    <image>
      <attach>
          <content>/9j/bcvb4234hgdffsd45</content>
          <filekey>67863659182463338076 2bee</filekey>
          <name>example.jpg</name>
          <size>65101</size>
          <type>image_path</type>
      </attach>
    </image>
    <image_caption>Описание Telegram bot сообщения</image_caption>
    <markup_style>HTML</markup_style>
  </fields>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
legacy_button_link	string	“https://example.com”	Ссылка, прикреплённая к кнопке
legacy_button_name	string	“button1”	Название кнопки
markdown_type	string	“HTML”	Тип форматирования текста (HTML/Markdown)

Параметр	Тип	Пример	Описание
text	string	“example”	Основной текст сообщения
content	string	“/9j/bcvb4234hgdffsd45”	Бинарное содержимое файла (base64)
filekey	string	“67863659182463338076 2bee”	Уникальный идентификатор файла
name	string	“example.jpg”	Название прикреплённого файла
size	integer	65101	Размер файла в байтах
type	string	“image_path”	Тип вложения (например, путь к файлу)
image_caption	string	“Описание изображения”	Подпись к изображению
audio_caption	string	“Описание аудиофайла”	Подпись к аудиофайлу
video_caption	string	“Описание видеофайла”	Подпись к видеофайлу
markup_style	string	“HTML”	Стиль разметки для Telegram (HTML/Markdown)

Whatsapp fields

JSON

```
{
  "id": 9,
  "sid": "whatsapp",
  "fields": {
    "type": "TEMPLATE",
    "template": {
      "type": "360DIALOG",
      "name": "initiate_message_ru",
      "hsm_external_id": "c466caf6_c340_4d85_8659_6c9685618a9f_ru_initiate_message_ru",
      "language": "ru",
      "wa_account_id": "+79999999999",
      "hsm_hash_sum": "0b7ac0a91b0ba52e961b87c2297d2bd06a3f5bc1",
      "is_invalid": false,
      "hsm_not_found": false,
      "account_not_found": false
    },
    "raw_hsm": {
      "dialog_template": {
        "components": [
          {
            "type": "BODY",
            "text": "Здравствуйте! Благодарю за ожидание, мне понадобилось немного времени, чтобы уточнить информацию по вашему запросу. Подскажите, когда Вам будет удобно обсудить детали? 😀"
          }
        ],
        "name": "initiate_message_ru",
        "namespace": "c466caf6_c340_4d85_8659_6c9685618a9f",
        "category": "ISSUE_RESOLUTION",
        "language": "ru",
        "rejected_reason": "NONE",
        "status": "approved"
      },
      "edna_template": null
    }
  }
}
```


XML

```
<xml>
  <id>9</id>
  <sid>whatsapp</sid>
  <fields>
    <type>TEMPLATE</type>
    <template>
      <type>360DIALOG</type>
      <name>initiate_message_ru</name>

<hsm_external_id>c466caf6_c340_4d85_8659_6c9685618a9f_ru_initiate_message_ru</hsm_external_
id>
      <language>ru</language>
      <wa_account_id>+79999999999</wa_account_id>
      <hsm_hash_sum>0b7ac0a91b0ba52e961b87c2297d2bd06a3f5bc1</hsm_hash_sum>
      <is_invalid>false</is_invalid>
      <hsm_not_found>false</hsm_not_found>
      <account_not_found>false</account_not_found>
    </template>
    <raw_hsm>
      <dialog_template>
        <components>
          <component>
            <type>BODY</type>
            <text>Здравствуйте! Благодарю за ожидание, мне понадобилось немного времени,
чтобы уточнить информацию по вашему запросу. Подскажите, когда Вам будет удобно обсудить
детали? 😃</text>
          </component>
        </components>
        <name>initiate_message_ru</name>
        <namespace>c466caf6_c340_4d85_8659_6c9685618a9f</namespace>
        <category>ISSUE_RESOLUTION</category>
        <language>ru</language>
        <rejected_reason>NONE</rejected_reason>
        <status>approved</status>
      </dialog_template>
    </raw_hsm>
  </fields>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
type	string	“360DIALOG”	Тип шаблона WhatsApp
name	string	“initiate_message_ru”	Название шаблона

Параметр	Тип	Пример	Описание
hsm_external_id	string	“c466caf6_c340_4d85_8659_6c9685618a9f_ru_initiate_message_ru”	Внешний ID шаблона
language	string	“ru”	Язык шаблона
wa_account_id	string	“+79999999999”	ID WhatsApp-аккаунта
hsm_hash_sum	string	“0b7ac0a91b0ba52e961b87c2297d2bd06a3f5bc1”	хеш шаблона
is_invalid	boolean	false	Является ли шаблон некорректным
hsm_not_found	boolean	false	Найден ли HSM шаблон
account_not_found	boolean	false	Найден ли аккаунт
name	string	“initiate_message_ru”	Название шаблона в raw_hsm
namespace	string	“c466caf6_c340_4d85_8659_6c9685618a9f”	Namespace шаблона
category	string	“ISSUE_RESOLUTION”	Категория шаблона
status	string	“approved”	Статус шаблона (например, approved, rejected)
rejected_reason	string	“NONE”	Причина отклонения шаблона (если есть)
components.type	string	“BODY”	Тип компонента шаблона
components.text	string	“Здравствуйте! Благодарю за ожидание…”	Текст тела сообщения
edna_template	string	“null”	Шаблон Edna, если присутствует (в данном случае — отсутствует)

Viber™ fields

JSON

```
"id": 10,
{
  "sid": "viber",
  "fields": {
    "series": [
      {
        "type": "TEXT",
        "text": "текст сообщения"
      },
      {
        "type": "IMAGE",
        "image": "/849074172/3/example.jpg"
      },
      {
        "type": "VIDEO",
        "thumbnail": "/848985842/3/6786323458246333807 62bee/example.jpg",
        "duration": 6,
        "file_size": 2,
        "action": "https://example.com/lib/preview/mp4/example.mp4"
      },
      {
        "type": "TEXT_BUTTON",
        "text": "Пример текста",
        "caption": "текст кнопки",
        "action": "https://example.com"
      },
      {
        "type": "FILE",
        "file_type": "pdf",
        "file_name": "example.pdf",
        "action": "https://www.example.org/testfiles/resources/pdf/example.pdf"
      }
    ]
  }
}
```

XML

```
<xml>
  <id>10</id>
  <sid>viber</sid>
  <fields>
    <series>
      <element>
        <type>TEXT</type>
        <text>текст сообщения</text>
      </element>
      <element>
        <type>IMAGE</type>
        <image>/849074172/3/example.jpg</image>
      </element>
      <element>
        <type>VIDEO</type>
        <thumbnail>/848985842/3/6786323458246333807 62bee/example.jpg</thumbnail>
        <duration>6</duration>
        <file_size>2</file_size>
        <action>https://example.com/lib/preview/mp4/example.mp4</action>
      </element>
      <element>
        <type>TEXT_BUTTON</type>
        <text>Пример текста</text>
        <caption>текст кнопки</caption>
        <action>https://example.com</action>
      </element>
      <element>
        <type>FILE</type>
        <file_type>pdf</file_type>
        <file_name>example.pdf</file_name>
        <action>https://www.example.org/testfiles/resources/pdf/example.pdf</action>
      </element>
    </series>
  </fields>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
type	string	“TEXT”	Тип элемента в серии сообщения
text	string	“текст сообщения”	Текстовое содержимое сообщения
image	string	“/849074172/3/example.jpg”	Путь к изображению
thumbnail	string	“/848985842/3/6786323458246333807 62bee/example.jpg”	Превью для видео

Параметр	Тип	Пример	Описание
duration	integer	6	Длительность видео в секундах
file_size	integer	2	Размер файла в мегабайтах
action	string	“https://example.com/lib/preview/mp4/example.mp4”	Ссылка действия (для видео, кнопки или файла)
caption	string	“текст кнопки”	Подпись для кнопки
file_type	string	“pdf”	Тип файла (например, PDF)
file_name	string	“example.pdf”	Название прикреплённого файла

---

* Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.

Получить список рассылок

Описание

Получить список рассылок.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных по умолчанию – "json"
from_id	int	1	Нет	Идентификатор первой рассылки в ответе по умолчанию – 1
limit	int	2	Нет	Лимит на количество возвращаемых данных по умолчанию – 1000
type	string	"trigger"	Нет	Тип рассылки
is_active	bool	true	Нет	Статус рассылки
ui_tags	[]String	["tag1", "tag2"]	Нет	Список тегов для поиска по умолчанию – []
groups	[]String	["group1", "group2"]	Нет	Список групп для поиска по умолчанию – []
groups_id	[]int	[0, 1]	Нет	Список ID групп для поиска по умолчанию – []
attributes	JSON object	`"attributes": \{ "season": \{ "options": \["spring_2024", "summer_2024"\] \} \}`	Нет	Значения атрибутов для поиска

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "limit": 2
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <limit>2</limit>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "id": 5,
            "name": "New meteoric campaign",
            "type": "broadcast",
            "is_active": false,
            "subid": "",
            "for_confirm": false,
            "ui_tags": [],
            "groups": [
                {
                    "id": 0,
                    "name": "main"
                }
            ],
            "attributes": {
                "season": "spring_2024",
                "branch": "marketing"
            }
        },
        {
            "id": 7,
            "name": "Copy of New meteoric campaign",
            "type": "broadcast",
            "is_active": false,
            "subid": "",
            "for_confirm": false,
            "ui_tags": [],
            "groups": [
                {
                    "id": 0,
                    "name": "main"
                }
            ]
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "next_from_id": 8,
    "total_count": 134
}
```


XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <data>
        <id>5</id>
        <name>New meteoric campaign</name>
        <type>broadcast</type>
        <is_active>false</is_active>
        <subid></subid>
        <for_confirm>false</for_confirm>
        <groups>
            <id>0</id>
            <name>main</name>
        </groups>
        <attributes>
            <season>spring_2024</season>
            <branch>marketing</branch>
        </attributes>
    </data>
    <data>
        <id>7</id>
        <name>Copy of New meteoric campaign</name>
        <type>broadcast</type>
        <is_active>false</is_active>
        <subid></subid>
        <for_confirm>false</for_confirm>
        <groups>
            <id>0</id>
            <name>main</name>
        </groups>
    </data>
    <next_from_id>8</next_from_id>
    <total_count>134</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	int	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории

Параметр	Тип	Описание
id	int	Идентификатор рассылки
is_active	bool	Статус рассылки
name	string	Название рассылки
type	string	Тип рассылки
total_count	int	Общее количество рассылок
for_confirm	bool	Имеет ли рассылка сообщения со ссылкой подтверждения
ui_tags	[]String	Список тегов
groups	[]String	Список групп
attributes	object	Атрибуты, назначенные рассылке, и их значения

Получить информацию о рассылке

Описание

Получить информацию о рассылке.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
id	int	1	Да	Идентификатор рассылки

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "id": 118,
        "name": "Daily weather notification",
        "description": "",
        "type": "regular",
        "updated_user": "roman",
        "update_time": "2024-05-20T09:17:53.362Z",
        "is_active": false,
        "is_completed": false,
        "data_id": 1,
        "data_type": "list",
        "content": [
            {
                "subject": "",
                "from_email": "",
                "from_name": "",
                "replyto_email": "",
                "replyto_name": "",
                "message_id": 21
            }
        ],
        "status": "Inactive",
        "subid": "",
        "virtual_sender_id": 2,
        "providers": [
            "google",
            "mozilla",
            "safari",
            "safari-native",
            "firefox-firebase",
            "chrome-firebase",
            "opera-firebase",
            "yandex-firebase",
            "samsung-firebase",
            "android-firebase",
            "ios-firebase",
            "ios-apns",
            "yandex-Appmetrica-device-id-ios",
            "yandex-Appmetrica-device-id-android",
            "yandex-Appmetrica-ios-ifa",
            "yandex-Appmetrica-google-aid",
            "yandex-Appmetrica-android-token",
            "yandex-Appmetrica-ios-token",
            "android-huawei",
            "ios-huawei",
            "android-rustore"
        ],
        "push_sending_mode": "fanout"
```

```
        "push_sending_mode": "fanout",
        "resource_id": 26,
        "channel": "push",
        "attributes": {
            "season": "spring_2024",
            "branch": "marketing"
        },
        "for_confirm": false
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <data>
        <id>118</id>
        <name>Daily weather notification</name>
        <description></description>
        <type>regular</type>
        <updated_user>roman</updated_user>
        <update_time>2024-05-20T09:17:53.362Z</update_time>
        <is_active>false</is_active>
        <is_completed>false</is_completed>
        <data_id>1</data_id>
        <data_type>list</data_type>
        <content>
            <subject></subject>
            <from_email></from_email>
            <from_name></from_name>
            <replyto_email></replyto_email>
            <replyto_name></replyto_name>
            <message_id>21</message_id>
        </content>
        <status>Inactive</status>
        <subid></subid>
        <virtual_sender_id>2</virtual_sender_id>
        <providers>google</providers>
        <providers>mozilla</providers>
        <providers>safari</providers>
        <providers>safari-native</providers>
        <providers>firefox-firebase</providers>
        <providers>chrome-firebase</providers>
        <providers>opera-firebase</providers>
        <providers>yandex-firebase</providers>
        <providers>samsung-firebase</providers>
        <providers>android-firebase</providers>
        <providers>ios-firebase</providers>
        <providers>ios-apns</providers>
        <providers>yandex-Appmetrica-device-id-ios</providers>
        <providers>yandex-Appmetrica-device-id-android</providers>
        <providers>yandex-Appmetrica-ios-ifa</providers>
        <providers>yandex-Appmetrica-google-aid</providers>
        <providers>yandex-Appmetrica-android-token</providers>
        <providers>yandex-Appmetrica-ios-token</providers>
        <providers>android-huawei</providers>
        <providers>ios-huawei</providers>
        <providers>android-rustore</providers>
        <push_sending_mode>fanout</push_sending_mode>
        <resource_id>26</resource_id>
        <channel>push</channel>
```

```
        <channel>push</channel>
        <for_confirm>false</for_confirm>
        <attributes>
            <season>spring_2024</season>
            <branch>marketing</branch>
        </attributes>
    </data>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
content	JSON array	Шаблоны, прикрепленные к рассылке
data_id	int	Идентификатор источника подписчиков
data_type	string	Тип источника подписчиков
id	int	Идентификатор рассылки
update_time	string	Дата последнего редактирования рассылки
updated_user	string	Имя пользователя, отредактировавшего рассылку
is_active	bool	Статус рассылки
is_completed	bool	Завершена ли рассылка
name	string	Название рассылки
resource_id	int	Идентификатор ресурса
send_email_content	bool	Включены ли email-сообщения
send_push_content	bool	Включены ли PUSH-сообщения
send_sms_content	bool	Включены ли SMS-сообщения
push_sending_mode	string	Режим отправки push-сообщений (только для push-каналов)
for_confirm	bool	Имеет ли рассылка сообщения со ссылкой подтверждения
status	string	Состояние рассылки
subid	string	SubID рассылки
type	string	Тип рассылки
virtual_sender_id	int	Идентификатор виртуального сендера
descr	string	Описание рассылки
attributes	object	Атрибуты, назначенные рассылке, и их значения

Получить лог рассылки

Описание

Получить лог рассылки.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/log_v2`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
from_time	string	“2018-05-25T00:00:00.000Z”	Нет	Получить логи начиная с этой отметки времени (в формате RFC 3339)
id	int	1	Да	Идентификатор рассылки
limit	int	2	Нет	Лимит на количество возвращаемых данных по умолчанию – 1000

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1,
    "limit": 2
}
```

XML

```
<xml>
   <token>abcdefghijklmnqrstuvwxyz</token>
   <id>1</id>
   <limit>2</limit>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "level": "INFO",
            "message": "Campaign finished",
            "time": "2024-04-02T17:57:01.504Z",
            "state": "end",
            "value": ""
        },
        {
            "level": "INFO",
            "message": "Total time spent",
            "time": "2024-04-02T17:57:01.505Z",
            "state": "",
            "value": "138.351985ms"
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 140
}
```

XML

```
<xml>
    <data>
        <level>"INFO"</level>
        <message>"Campaign finished"</message>
        <state>"end"</state>
        <time>"2024-05-28T16:40:59.207Z"</time>
        <value>""</value>
    </data>
    <data>
        <level>"INFO"</level>
        <message>"Total time spent"</message>
        <state>""</state>
        <time>"2024-05-28T16:40:59.208Z"</time>
        <value>"351.155547ms"</value>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>9</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
level	string	Уровень логирования
message	string	Сообщение журнала
state	string	Состояние события
time	date	Дата регистрации события
value	string	Значение, связанное с состоянием события

Уровни логирования:

- DEBUG
- INFO
- WARNING
- ERROR

Состояния:

- start – старт рассылки
- end – окончание рассылки

Клонировать рассылку

Описание

Клонировать рассылку.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/clone`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
id	int	1	Да	Идентификатор рассылки
name	string	“Clone”	Нет	Название клона рассылки по умолчанию – “Clone of …”

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1,
    "name": "Clone"
}
```

XML

```
<xml>
   <token>abcdefghijklmnqrstuvwxyz</token>
   <id>1</id>
   <name>Clone</name>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 87
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>88</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор клона рассылки

Удалить рассылку

Описание

Удалить рассылку.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
id	int	1	Да	Идентификатор рассылки

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>1</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Активировать рассылку

Описание

Активировать рассылку.

Также используется для того, чтобы мгновенно отправить броадкаст или регулярную рассылку. Для этой отправки можно указать другой источник подписчиков.

Если поле `run_now = true`, то рассылка должна запускаться немедленно.

Поля `override_source_type` и `override_source_id` используются, только когда `run_now = true`.

Если переданы параметры `override_source_type` и `override_source_id`, то рассылка выполняются по переданному источнику подписчиков.

Если их нет, то рассылка выполняется по своему источнику.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/activate/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	12	Да	Идентификатор рассылки
run_now	bool	true	Нет	Запустить рассылку
override_source_type	enum	“list” или “segment”	Нет	Перезаписать источник данных
override_source_id	int	5	Нет	Перезаписать идентификатор данных
override_limit	int	56	Нет	Перезаписать лимит на старт рассылки
content	JSON object	`{ “field”: “value” }`	Нет	Контент для переменных apicontent в сообщениях.
attach	JSON array	[]	Нет	File object

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 82,
  "override_source_id": 1,
  "override_source_type": "list",
  "run_now": true
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>82</id>
    <override_source_id>1</override_source_id>
    <override_source_type>list</override_source_type>
    <run_now>true</run_now>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Деактивировать рассылку

Описание

Деактивировать рассылку.

Также можно прервать отправку сообщений работающей рассылки.

`abort_if_running = true` прерывает отправку сообщений, если рассылка работает. Значение `false` только деактивирует, позволяя дослать последние оставшиеся сообщения.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/deactivate`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
abort_if_running	bool	true	Нет	Прервать, если рассылка запущена. По умолчанию – false.
id	int	12	Да	Идентификатор рассылки

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": "78",
    "abort_if_running": "true"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>78</id>
    <abort_if_running>true</abort_if_running>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить статус рассылки

Описание

Получить статус рассылки.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/get_status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	12	Да	Идентификатор рассылки

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 82
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>82</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "is_active": false,
        "is_completed": false,
        "status": "Inactive"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <is_active>false</is_active>
        <is_completed>false</is_completed>
        <status>"Inactive"</status>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
is_active	bool	Активирована ли рассылка
is_completed	bool	Завершена ли рассылка
status	string	Текущий статус рассылки

Получить список броадкаст рассылок

Описание

Получить список броадкастов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/broadcasts/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	`"abcdefghijklmnqrstuvwxyz"`	Да	API-токен
from_id	int	`11768`	Нет	Идентификатор первой рассылки в списке. По умолчанию – `1`.
limit	int	`10`	Нет	Лимит на количество возвращаемых данных. По умолчанию – `1000`.
format	string	`"json"`	Нет	Формат возвращаемых данных. По умолчанию – `"json"`.
is_active	bool	`true`	Нет	Статус рассылки
ui_tags	[]String	`[ "tag1", "tag2" ]`	Нет	Список тегов для поиска. По умолчанию – `[]`.
groups	[]String	`[ "group1", "group2" ]`	Нет	Список групп для поиска. По умолчанию – `[]`.
groups_id	[]int	`[ 0, 1 ]`	Нет	Список Id групп для поиска. По умолчанию – `[]`.
attributes	JSON object	`"attributes": \{ "season": \{ "options": \["spring_2024", "summer_2024"\] \} \}`	Нет	Значения атрибутов для поиска

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "limit": 2
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <limit>2</limit>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "id": 3,
            "name": "New campaign 1",
            "type": "broadcast",
            "is_active": false,
            "for_confirm": false,
            "ui_tags": [],
            "groups": [
                {
                    "id": 0,
                    "name": "main"
                }
            ],
            "attributes": {
                "season": "spring_2024",
                "branch": "marketing"
            }
        },
        {
            "id": 5,
            "name": "New campaign 2",
            "type": "broadcast",
            "is_active": false,
            "for_confirm": false,
            "ui_tags": [],
            "groups": [
                {
                    "id": 0,
                    "name": "main"
                }
            ]
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 17
}
```


XML

```
<xml>
    <data>
        <for_confirm>false</for_confirm>
        <groups>
            <id>0</id>
            <name>main</name>
        </groups>
        <attributes>
            <season>spring_2024</season>
            <branch>marketing</branch>
        </attributes>
        <id>3</id>
        <is_active>false</is_active>
        <name>New campaign 1</name>
        <type>broadcast</type>
    </data>
    <data>
        <for_confirm>false</for_confirm>
        <groups>
            <id>0</id>
            <name>main</name>
        </groups>
        <id>5</id>
        <is_active>false</is_active>
        <name>New campaign 2</name>
        <type>broadcast</type>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>17</total_count>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор рассылки
name	string	Название рассылки
type	string	Тип рассылки
is_active	bool	Статус рассылки
for_confirm	bool	Имеет ли рассылка сообщения со ссылкой подтверждения

Параметр	Тип	Описание
total_count	int	Общее количество рассылок
ui_tags	[]String	Список тегов
groups	[]String	Список групп
attributes	object	Атрибуты, назначенные рассылке, и их значения

Получить информацию о броадкаст рассылке

Описание

Получить информацию о броадкасте.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/broadcasts/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	11768	Да	Идентификатор рассылки
format	string	“json”	Нет	Формат возвращаемых данных. По умолчанию – “json”.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": "87"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>87</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "name": "sms ",
        "is_active": false,
        "data_id": 13,
        "data_type": "segment",
        "content": [
            {
                "subject": "",
                "from_email": "",
                "from_name": "",
                "replyto_email": "",
                "replyto_name": "",
                "message_id": 27
            }
        ],
        "send_channel": "sms",
        "channel": "sms",
        "push_sending_mode": "fanout",
        "resource_id": 36,
        "subid": "",
        "limit": 0,
        "aglimit_count": 0,
        "aglimit_percent": 0,
        "start_schedule": {
            "start_mins": [
                38
            ],
            "start_hours": [
                16
            ],
            "start_mdays": [],
            "start_wdays": [],
            "start_months": [],
            "start_years": [],
            "tz": "",
            "profile": false,
            "losers": false
        },
        "for_confirm": false,
        "attributes": {
            "season": "spring_2024",
            "branch": "marketing"
        },
        "policy_id": 0,
        "description": "",
        "virtual_sender_id": 0,
        "is_random_tracking_prefix": false,
        "custom_tracking_domains": []
```

```
        "custom_tracking_domains": [],
        "watcher_list_id": 0,
        "watcher_limit": 0,
        "blind_carbon_copy": [],
        "unconditional_sending": false,
        "skip_unique_checking": false,
        "sub_cats_filter": null,
        "sub_fields_filter": null,
        "id": 179,
        "updated_user": "roman",
        "update_time": "2024-05-28T13:40:57.3Z"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
    <data>
        <aglimit_count>0</aglimit_count>
        <aglimit_percent>0</aglimit_percent>
        <channel>"sms"</channel>
        <content>
            <from_email>""</from_email>
            <from_name>""</from_name>
            <message_id>27</message_id>
            <replyto_email>""</replyto_email>
            <replyto_name>""</replyto_name>
            <subject>""</subject>
        </content>
        <data_id>13</data_id>
        <data_type>"segment"</data_type>
        <description>""</description>
        <for_confirm>false</for_confirm>
        <id>179</id>
        <is_active>false</is_active>
        <is_random_tracking_prefix>false</is_random_tracking_prefix>
        <limit>0</limit>
        <name>"sms "</name>
        <policy_id>0</policy_id>
        <push_sending_mode>"fanout"</push_sending_mode>
        <attributes>
            <season>spring_2024</season>
            <branch>marketing</branch>
        </attributes>
        <resource_id>36</resource_id>
        <send_channel>"sms"</send_channel>
        <skip_unique_checking>false</skip_unique_checking>
        <start_schedule>
            <losers>false</losers>
            <profile>false</profile>
            <start_hours>16</start_hours>
            <start_mins>38</start_mins>
            <tz>""</tz>
        </start_schedule>
        <subid>""</subid>
        <unconditional_sending>false</unconditional_sending>
        <update_time>"2024-05-28T13:40:57.3Z"</update_time>
        <updated_user>"roman"</updated_user>
        <virtual_sender_id>0</virtual_sender_id>
        <watcher_limit>0</watcher_limit>
        <watcher_list_id>0</watcher_list_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

</xml1>

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор рассылки
name	string	Название рассылки
is_active	bool	Статус рассылки
resource_id	int	Идентификатор ресурса
data_id	int	Идентификатор базы данных или сегмента
data_type	string	Тип источника подписчиков
content	json array	Контент, присоединенный к рассылке
send_channel	string	Канал отправки
channel	string	Канал
for_confirm	bool	Использует ли рассылка сообщения со ссылкой подтверждения
subid	string	Subid рассылки
start_shedule	json object	Start schedule object
limit	int	Лимит на каждый запуск
aglimit_count	int	Приращивать лимит по количеству
agllimit_percent	float	Приращивать лимит по проценту
policy_id	int	Идентификатор политики отправки
push_sending_mode	string	Режим отправки push-сообщений (только для push-каналов)
descr	string	Описание рассылки
attributes	object	Атрибуты, назначенные рассылке, и их значения
Smart настройки (только для smart-аккаунтов)		
virtual_sender_id	int	Идентификатор виртуального сендера
is_random_tracking_prefix	bool	Случайный префикс трекинг-домена
custom_tracking_domains	json array	Список произвольных трекинг-доменов
blind_carbon_copy	json array	Адреса для скрытой копии
unconditional_sending	bool	Отправка вне зависимости от статуса подписки
watcher_list_id	int	Идентификатор базы наблюдателей

watcher_limit	int	Лимит наблюдателей

Добавить броадкаст рассылку

Описание

Создать новую броадкаст рассылку.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/broadcasts/add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных. По умолчанию – "json".
name	string	"Broadcast campaign name"	Да	Название рассылки
is_active	bool	true	Нет	Статус рассылки
data_type	string	"list"	Да	Тип источника подписчиков
data_id	int	12	Да	Идентификатор базы данных или сегмента
resource_id	int	24	Да	Идентификатор ресурса
content	JSON array		Да	Контент, присоединенный к рассылке
send_channel	string	"email" "sms" "push"	Да	Канал отправки
content_type	string	"email"	Нет	Устаревший параметр. При его передаче рассылка будет использовать email-канал для отправки. Рекомендуется использовать **`send_channel`**.
send_email_content	bool	true	Нет	Устаревший параметр. При его передаче рассылка будет использовать email-канал для отправки. Рекомендуется использовать **`send_channel`**.
send_push_content	bool	true	Нет	Устаревший параметр. При его передаче рассылка будет использовать email-канал для отправки. Рекомендуется использовать **`send_channel`**.

send_sms_content	bool	true	Нет	Устаревший параметр. При его передаче рассылка будет использовать email-канал для отправки. Рекомендуется использовать `**send_channel**`.
subid	string	"Adv_1"	Нет	subid рассылки
start_schedule	JSON-object		Нет	Start schedule object — объект расписания
limit	int	0	Нет	Лимит на каждый запуск
aglimit_count	int	0	Нет	Приращивать лимит по количеству
aglimit_percent	float	99.0	Нет	Приращивать лимит по проценту
policy_id	int	1	Нет	Идентификатор политики отправки
push_sending_mode	string	"fanout" (по умолчанию) "last"	Нет	Режим отправки push-сообщений (только для push-каналов)
attributes	object	`"season":"2024_summer"`	Нет	Атрибуты, назначенные рассылке, и их значения
Smart настройки				
virtual_sender_id	int	12	Нет	Идентификатор виртуального сендера **Только для smart-аккаунтов**
is_random_tracking_prefix	bool	false	Нет	Случайный префикс трекинг-домена **Только для smart-аккаунтов**
custom_tracking_domains	JSON array	`["trk.example.com", "trk2.example.com"]`	Нет	Произвольные трекинг-домены **Только для smart-аккаунтов**
blind_carbon_copy	JSON array	`["john@example.com", "george@example.com"]`	Нет	Адреса для скрытой копии **Только для smart-аккаунтов**
unconditional_sending	bool	true	Нет	Отправка вне зависимости от статуса подписки **Только для smart-аккаунтов**
watcher_list_id	int	12	Нет	Идентификатор базы наблюдателей **Только для smart-аккаунтов**
watcher_limit	int	100	Нет	Лимит наблюдателей **Только для smart-аккаунтов**

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "virtual_sender_id": 5,
    "content": [
        {
            "from_email": "example@example.com",
            "from_name": "Victor",
            "message_id": 15,
            "replyto_email": "example@example.com",
            "replyto_name": "Victoria",
            "subject": "Hello"
        }
    ],
    "custom_tracking_domains": [
        "trk.example.com",
        "trk.example.org"
    ],
    "data_id": 1,
    "data_type": "list",
    "is_active": true,
    "limit": 1,
    "name": "New broadcast",
    "resource_id": 21,
    "send_channel": "email",
    "start_schedule": {
        "start_hours": [
            1,
            2,
            3,
            4
        ],
        "start_mins": [
            1,
            2,
            3,
            4
        ],
        "start_mdays": [
            1
        ],
        "start_months": [
            12
        ],
        "start_wdays": [],
        "start_years": [
            2024
        ]
    }
}
```

```
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <virtual_sender_id>5</virtual_sender_id>
    <content array='true'>
        <from_email>example@example.com</from_email>
        <from_name></from_name>
        <message_id>14</message_id>
        <replyto_email>example@example.com</replyto_email>
        <replyto_name></replyto_name>
        <subject>Hello!</subject>
    </content>
    <custom_tracking_domains>trk.example.com</custom_tracking_domains>
    <custom_tracking_domains>trk.example.org</custom_tracking_domains>
    <data_id>1</data_id>
    <data_type>list</data_type>
    <is_active>true</is_active>
    <limit>1</limit>
    <name>Broadcast</name>
    <resource_id>21</resource_id>
    <send_channel>email</send_channel>
    <start_schedule>
        <start_hours>1</start_hours>
        <start_hours>2</start_hours>
        <start_hours>3</start_hours>
        <start_hours>4</start_hours>
        <start_mins>1</start_mins>
        <start_mins>2</start_mins>
        <start_mins>3</start_mins>
        <start_mins>4</start_mins>
        <start_mdays array='true'>1</start_mdays>
        <start_months array='true'>12</start_months>
        <start_wdays array='true'></start_wdays>
        <start_years array='true'>2024</start_years>
    </start_schedule>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 90
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>90</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор рассылки

Обновить броадкаст рассылку

Описание

Изменить настройки броадкаст рассылки.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/broadcasts/update`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор рассылки
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
name	string	“Broadcast campaign name”	Да	Название рассылки
is_active	bool	true	Нет	Статус рассылки
data_type	string	“list”	Да	Тип источника подписчиков
data_id	int	12	Да	Идентификатор базы данных или сегмента
resource_id	int	24	Да	Идентификатор ресурса
content	JSON array		Да	Контент, присоединенный к рассылке
subid	string	“Adv_1”	Нет	subid рассылки
start_schedule	JSON-object		Нет	Start schedule object
limit	int	0	Нет	Лимит на каждый запуск
aglimit_count	int	0	Нет	Приращивать лимит по количеству
aglimit_percent	float	99.0	Нет	Приращивать лимит по проценту
policy_id	int	1	Нет	Идентификатор политики отправки
push_sending_mode	string	“fanout” (по умолчанию) “last”	Нет	Режим отправки push-сообщений (только для push-каналов)

Параметр	Тип	Пример	Обязательный	Описание
attributes	object	`"season":"2024_summer"`	Нет	Атрибуты, назначенные рассылке, и их значения
Smart настройки				
virtual_sender_id	int	12	Нет	Идентификатор виртуального сендера **Только для smart-аккаунтов**
is_random_tracking_prefix	bool	false	Нет	Случайный префикс трекинг-домена **Только для smart-аккаунтов**
custom_tracking_domains	JSON array	`["trk.example.com", "trk2.example.com"]`	Нет	Список трекинг-доменов **Только для smart-аккаунтов**
blind_carbon_copy	JSON array	`["john@example.com", "george@example.com"]`	Нет	Адреса для скрытой копии **Только для smart-аккаунтов**
unconditional_sending	bool	true	Нет	Отправка вне зависимости от статуса подписки **Только для smart-аккаунтов**
watcher_list_id	int	12	Нет	Идентификатор базы наблюдателей **Только для smart-аккаунтов**
watcher_limit	int	100	Нет	Лимит наблюдателей **Только для smart-аккаунтов**

NOTE

Изменять канал в уже сохраненной рассылке нельзя. При передаче соответствующих параметров — `send_channel`, `content_type`, `send_email_content`, `send_sms_content`, `send_push_content` — запрос не вернет ошибку, но и не обновит канал отправки.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "virtual_sender_id": 5,
    "content": [
        {
            "from_email": "example@example.com",
            "from_name": "Alexandra",
            "message_id": 14,
            "replyto_email": "example@example.com",
            "replyto_name": "Victoria",
            "subject": "Hi!"
        }
    ],
    "custom_tracking_domains": [
        "trk.example.com",
        "trk.example.org"
    ],
    "data_id": 1,
    "data_type": "list",
    "id": 90,
    "is_active": true,
    "limit": 10,
    "name": "Broadcast",
    "resource_id": 20,
    "start_schedule": {
        "start_hours": [
            1,
            2,
            3,
            4
        ],
        "start_mdays": [
            1
        ],
        "start_mins": [
            1,
            2,
            3,
            4
        ],
        "start_months": [
            12
        ],
        "start_wdays": [],
        "start_years": [
            2024
        ]
    }
}
```

```
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <virtual_sender_id>5</virtual_sender_id>
    <content array='true'>
        <from_email>broadcast1@altcraft.com</from_email>
        <from_name>Margarita</from_name>
        <message_id>14</message_id>
        <replyto_email>example@example.com</replyto_email>
        <replyto_name>Victoria</replyto_name>
        <subject>Good morning!</subject>
    </content>
    <custom_tracking_domains>trk.example.com</custom_tracking_domains>
    <custom_tracking_domains>trk.example.org</custom_tracking_domains>
    <data_id>1</data_id>
    <data_type>list</data_type>
    <id>89</id>
    <is_active>true</is_active>
    <limit>10</limit>
    <name>Broadcast</name>
    <resource_id>20</resource_id>
    <start_schedule>
        <start_hours>1</start_hours>
        <start_hours>2</start_hours>
        <start_hours>3</start_hours>
        <start_hours>4</start_hours>
        <start_mins>1</start_mins>
        <start_mins>2</start_mins>
        <start_mins>3</start_mins>
        <start_mins>4</start_mins>
        <start_mdays array='true'>1</start_mdays>
        <start_months array='true'>12</start_months>
        <start_wdays array='true'></start_wdays>
        <start_years array='true'>2024</start_years>
    </start_schedule>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 90
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>90</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор рассылки

Запустить броадкаст рассылку

Описание

Мгновенно запускает броадкаст рассылку, но только если она активна.

Параметрами `override_source_type` и `override_source_id` можно изменить аудиторию рассылки для запуска.

Параметр `override_limit` определяет лимит на отправленные сообщения для текущего запуска рассылки.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/broadcasts/start`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	12	Да	Идентификатор рассылки
override_source_type	enum	“list” или “segment”	Нет	Перезаписать источник данных
override_source_id	int	5	Нет	Перезаписать идентификатор данных
override_limit	int	56	Нет	Перезаписать лимит на старт рассылки
content	JSON object	`{ "field": "value" }`	Нет	Контент для переменных apicontent в сообщениях.
attach	JSON array of File object	[]	Нет	File object

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 90,
    "override_source_id": 1,
    "override_source_type": "list"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>90</id>
  <override_source_id>1</override_source_id>
  <override_source_type>list</override_source_type>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить список регулярных рассылок

Описание

Получить список регулярных рассылок.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/regulars/list/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	`"abcdefghijklmnqrstuvwxyz"`	Да	API-токен
from_id	int	`11768`	Нет	Идентификатор первой рассылки в списке. По умолчанию – `1`.
limit	int	`10`	Нет	Лимит на количество возвращаемых данных. По умолчанию – `1000`.
format	string	`"json"`	Нет	Формат возвращаемых данных. По умолчанию – `"json"`.
is_active	bool	`true`	Нет	Статус рассылки
ui_tags	[]String	`[ "tag1", "tag2" ]`	Нет	Список тегов для поиска. По умолчанию – `[]`.
groups	[]String	`[ "group1", "group2" ]`	Нет	Список групп для поиска. По умолчанию – `[]`.
groups_id	[]int	`[ 0, 1 ]`	Нет	Список Id групп для поиска. По умолчанию – `[]`.
attributes	JSON object	`"attributes": \{ "season": \{ "options": \["spring_2024", "summer_2024"\] \} \}`	Нет	Значения атрибутов для поиска

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "limit": 2
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <limit>2</limit>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "id": 15,
            "name": "Regular campaign 1",
            "type": "regular",
            "is_active": true,
            "for_confirm": false,
            "ui_tags": [
                "B2C"
            ],
            "groups": [
                {
                    "id": 1,
                    "name": "subaccount"
                }
            ],
            "attributes": {
                "season": "spring_2024",
                "branch": "marketing"
            }
        },
        {
            "id": 78,
            "name": "Regular campaign 2",
            "type": "regular",
            "is_active": false,
            "for_confirm": false,
            "ui_tags": [
                "B2C"
            ],
            "groups": [
                {
                    "id": 1,
                    "name": "subaccount"
                }
            ]
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 3
}
```


XML

```
<xml>
    <data>
        <for_confirm>false</for_confirm>
        <groups>
            <id>1</id>
            <name>subaccount</name>
        </groups>
        <attributes>
            <season>spring_2024</season>
            <branch>marketing</branch>
        </attributes>
        <id>15</id>
        <is_active>true</is_active>
        <name>Regular campaign 1</name>
        <type>regular</type>
        <ui_tags>B2C</ui_tags>
    </data>
    <data>
        <for_confirm>false</for_confirm>
        <groups>
            <id>1</id>
            <name>subaccount</name>
        </groups>
        <id>78</id>
        <is_active>false</is_active>
        <name>Regular campaign 2</name>
        <type>regular</type>
        <ui_tags>B2C</ui_tags>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>3</total_count>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор рассылки
name	string	Название рассылки
type	string	Тип рассылки
is_active	bool	Статус рассылки

Параметр	Тип	Описание
for_confirm	bool	Имеет ли рассылка сообщения со ссылкой подтверждения
total_count	int	Общее количество рассылок
ui_tags	[]String	Список тегов
groups	[]String	Список групп
attributes	object	Атрибуты, назначенные рассылке, и их значения

Получить информацию о регулярной рассылке

Описание

Получить информацию о регулярной рассылке.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/regulars/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	11768	Да	Идентификатор рассылки
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 15
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>15</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "name": "Regular campaign name",
        "is_active": true,
        "data_id": 2,
        "data_type": "segment",
        "content": [
            {
                "subject": "Hi!",
                "from_email": "broadcast@altcraft.com",
                "from_name": "",
                "replyto_email": "broadcast@altcraft.com",
                "replyto_name": "",
                "message_id": 15
            }
        ],
        "send_channel": "email",
        "channel": "email",
        "resource_id": 20,
        "subid": "",
        "limit": 10,
        "aglimit_count": 0,
        "aglimit_percent": 0,
        "start_schedule": {
            "start_mins": [
                10
            ],
            "start_hours": [
                12
            ],
            "start_mdays": [],
            "start_wdays": [],
            "start_months": [],
            "start_years": [],
            "tz": "",
            "profile": false,
            "losers": false
        },
        "for_confirm": false,
        "attributes": {
            "season": "spring_2024",
            "branch": "marketing"
        },
        "policy_id": 0,
        "description": "",
        "virtual_sender_id": 1,
        "is_random_tracking_prefix": false,
        "custom_tracking_domains": [
            "trk.example.com"
```

```
            trk.example.com",
            "trk.example.org"
        ],
        "watcher_list_id": 0,
        "watcher_limit": 0,
        "blind_carbon_copy": [],
        "unconditional_sending": false,
        "skip_unique_checking": false,
        "sub_cats_filter": [],
        "sub_fields_filter": null,
        "id": 15,
        "updated_user": "roman",
        "update_time": "2024-04-03T15:35:54.024Z"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

<xml>
    <data>
        <aglimit_count>0</aglimit_count>
        <aglimit_percent>0</aglimit_percent>
        <channel>"email"</channel>
        <content>
            <from_email>"broadcast@altcraft.com"</from_email>
            <from_name>""</from_name>
            <message_id>15</message_id>
            <replyto_email>"broadcast@altcraft.com"</replyto_email>
            <replyto_name>""</replyto_name>
            <subject>"Hi!"</subject>
        </content>
        <custom_tracking_domains>"trk.example.com"</custom_tracking_domains>
        <custom_tracking_domains>"trk.example.org"</custom_tracking_domains>
        <data_id>2</data_id>
        <data_type>"segment"</data_type>
        <description>""</description>
        <for_confirm>false</for_confirm>
        <id>15</id>
        <is_active>false</is_active>
        <attributes>
            <season>spring_2024</season>
            <branch>marketing</branch>
        </attributes>
        <is_random_tracking_prefix>false</is_random_tracking_prefix>
        <limit>10</limit>
        <name>"Regular campaign name"</name>
        <policy_id>0</policy_id>
        <push_sending_mode>"fanout"</push_sending_mode>
        <resource_id>20</resource_id>
        <send_channel>"email"</send_channel>
        <skip_unique_checking>false</skip_unique_checking>
        <start_schedule>
            <losers>false</losers>
            <profile>false</profile>
            <start_hours>12</start_hours>
            <start_mins>10</start_mins>
            <tz>""</tz>
        </start_schedule>
        <subid>""</subid>
        <unconditional_sending>false</unconditional_sending>
        <update_time>"2024-05-20T09:17:53.326Z"</update_time>
        <updated_user>"roman"</updated_user>
        <virtual_sender_id>1</virtual_sender_id>
        <watcher_limit>0</watcher_limit>
        <watcher_list_id>0</watcher_list_id>
    </data>
    <error>0</error>
</xml>

```
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор
name	string	Название рассылки
is_active	bool	Статус рассылки
resource_id	int	Идентификатор ресурса
data_id	int	Идентификатор баз данных или сегмента
data_type	string	Тип источника подписчиков
content	JSON array	Контент, присоединенный к рассылке
send_channel	string	Канал отправки
channel	string	Канал
for_confirm	bool	Имеет ли рассылка сообщения со ссылкой подтверждения
subid	string	subid рассылки
start_shedule	JSON-object	Start schedule object
limit	int	Лимит на каждый запуск
aglimit_count	int	Приращивать лимит по количеству
agllimit_percent	float	Приращивать лимит по проценту
policy_id	int	Идентификатор политики отправки
push_sending_mode	string	Режим отправки push-сообщений (только для push-каналов)
descr	string	Описание рассылки
attributes	object	Атрибуты, назначенные рассылке, и их значения
Smart настройки		
virtual_sender_id	int	Идентификатор виртуального сендера **Только для smart-аккаунтов**
is_random_tracking_prefix	bool	Случайный префикс трекинг-домена **Только для smart-аккаунтов**
custom_tracking_domains	JSON array	Список произвольных трекинг-доменов **Только для smart-аккаунтов**

Параметр	Тип	Описание
blind_carbon_copy	JSON array	Адреса для скрытой копии **Только для smart-аккаунтов**
unconditional_sending	bool	Отправка вне зависимости от статуса подписки **Только для smart-аккаунтов**
watcher_list_id	int	Идентификатор базы наблюдателей **Только для smart-аккаунтов**
watcher_limit	int	Лимит наблюдателей **Только для smart-аккаунтов**

Добавить регулярную рассылку

Описание

Добавить регулярную рассылку.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/regulars/add/`

Параметры запроса

Атрибуты, назначенные рассылке, и их значения

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных по умолчанию – "json"
name	string	"Broadcast campaign name"	Да	Название рассылки
is_active	bool	true	Нет	Статус рассылки
data_type	string	"list"	Да	Тип источника подписчиков
data_id	int	12	Да	Идентификатор базы данных или сегмента
resource_id	int	24	Да	Идентификатор ресурса
content	JSON array		Да	Контент, присоединенный к рассылке
send_channel	string	"email" "sms" "push"	Да	Канал отправки
content_type	string	"email"	Нет	Устаревший параметр. При его передаче рассылка будет использовать email-канал для отправки. Рекомендуется использовать **`send_channel`**.
send_email_content	bool	true	Нет	Устаревший параметр. При его передаче рассылка будет использовать email-канал для отправки. Рекомендуется использовать **`send_channel`**.
send_push_content	bool	true	Нет	Устаревший параметр. При его передаче рассылка будет использовать email-канал для отправки. Рекомендуется использовать **`send_channel`**.

send_sms_content	bool	true	Нет	Устаревший параметр. При его передаче рассылка будет использовать email-канал для отправки. Рекомендуется использовать `**send_channel**`.
subid	string	"Adv_1"	Нет	subid рассылки
start_schedule	JSON object		Нет	Start schedule object
limit	int	0	Нет	Лимит на каждый запуск
aglimit_count	int	0	Нет	Приращивать лимит по количеству
aglimit_percent	float	99.0	Нет	Приращивать лимит по проценту
policy_id	int	1	Нет	Идентификатор политики отправки
push_sending_mode	string	"fanout" (по умолчанию) "last"	Нет	Режим отправки push-сообщений (только для push-каналов)
attributes	object	`"season":"2024_summer"`	Нет	attributes
Smart настройки				
virtual_sender_id	int	12	Нет	Идентификатор виртуального сендера **Только для smart-аккаунтов**
is_random_tracking_prefix	bool	false	Нет	Случайный префикс трекинг-домена **Только для smart-аккаунтов**
custom_tracking_domains	JSON array	["trk.example.com", "trk2.example.com"]	Нет	Список произвольных трекинг-доменов **Только для smart-аккаунтов**
blind_carbon_copy	JSON array	["john@example.com", "george@example.com"]	Нет	Адреса для скрытой копии **Только для smart-аккаунтов**
unconditional_sending	bool	true	Нет	Отправка вне зависимости от статуса подписки **Только для smart-аккаунтов**
watcher_list_id	int	12	Нет	Идентификатор базы наблюдателей **Только для smart-аккаунтов**
watcher_limit	int	100	Нет	Лимит наблюдателей **Только для smart-аккаунтов**

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "virtual_sender_id": 5,
    "content": [
        {
            "from_email": "regular@altcraft.com",
            "from_name": "Paul",
            "message_id": 15,
            "replyto_email": "regular@altcraft.com",
            "replyto_name": "Roman",
            "subject": "Hello"
        }
    ],
    "custom_tracking_domains": [
        "trk.example.com",
        "trk.example.org"
    ],
    "data_id": 1,
    "data_type": "list",
    "is_active": true,
    "limit": 1,
    "name": "Regular campaign #1",
    "resource_id": 20,
    "send_channel": "email",
    "start_schedule": {
         "start_hours": [
            1,
            2,
            3,
            4
        ],
        "start_mins": [
            1,
            2,
            3,
            4
        ],
        "start_mdays": [
            1
        ],
        "start_months": [
            12
        ],
        "start_wdays": [],
        "start_years": [
            2024
        ]
    }
}
```

```
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <virtual_sender_id>5</virtual_sender_id>
    <content array='true'>
        <from_email>example@example.com</from_email>
        <from_name>Paul</from_name>
        <message_id>15</message_id>
        <replyto_email>example@example.com</replyto_email>
        <replyto_name>Roman</replyto_name>
        <subject>Hello</subject>
    </content>
    <custom_tracking_domains>trk.example.com</custom_tracking_domains>
    <custom_tracking_domains>trk.example.org</custom_tracking_domains>
    <data_id>1</data_id>
    <data_type>list</data_type>
    <is_active>true</is_active>
    <limit>1</limit>
    <name>Regular campaign #2</name>
    <resource_id>20</resource_id>
    <send_channel>email</send_channel>
    <start_schedule>
        <start_hours>1</start_hours>
        <start_hours>2</start_hours>
        <start_hours>3</start_hours>
        <start_hours>4</start_hours>
        <start_mins>1</start_mins>
        <start_mins>2</start_mins>
        <start_mins>3</start_mins>
        <start_mins>4</start_mins>
        <start_mdays array='true'>1</start_mdays>
        <start_months array='true'>12</start_months>
        <start_wdays array='true'></start_wdays>
        <start_years array='true'>2024</start_years>
    </start_schedule>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 92
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>93</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор рассылки

Обновить регулярную рассылку

Описание

Обновить регулярную рассылку.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/regulars/update`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор рассылки
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
name	string	“Broadcast campaign name”	Да	Название рассылки
is_active	bool	true	Нет	Статус рассылки
data_type	string	“list”	Да	Тип источникаподписчиков
data_id	int	12	Да	Идентификатор базы данных или сегмента
resource_id	int	24	Да	Идентификатор ресурса
content	JSON array		Да	Контент, присоединенный к рассылке
subid	string	“Adv_1”	Нет	subid рассылки
start_schedule	JSON-object		Нет	Start schedule object
limit	int	0	Нет	Лимит на каждый запуск
aglimit_count	int	0	Нет	Приращивать лимит по количеству
aglimit_percent	float	99.0	Нет	Приращивать лимит по проценту
policy_id	int	1	Нет	Идентификатор политики отправки
push_sending_mode	string	“fanout” (по умолчанию) “last”	Нет	Режим отправки push-сообщений (только для push-каналов)

Параметр	Тип	Пример	Обязательный	Описание
attributes	object	`"season":"2024_summer"`	Нет	Атрибуты, назначенные рассылке, и их значения
Smart настройки				
virtual_sender_id	int	12	Нет	Идентификатор виртуального сендера **Только для smart-аккаунтов**
is_random_tracking_prefix	bool	false	Нет	Случайный префикс трекинг-домена **Только для smart-аккаунтов**
custom_tracking_domains	JSON array	`["trk.example.com", "trk2.example.com"]`	Нет	Список трекинг-доменов **Только для smart-аккаунтов**
blind_carbon_copy	JSON array	`["john@example.com", "george@example.com"]`	Нет	Адреса для скрытой копии **Только для smart-аккаунтов**
unconditional_sending	bool	true	Нет	Отправка вне зависимости от статуса подписки **Только для smart-аккаунтов**
watcher_list_id	int	12	Нет	Идентификатор базы наблюдателей **Только для smart-аккаунтов**
watcher_limit	int	100	Нет	Лимит наблюдателей **Только для smart-аккаунтов**

NOTE

Изменять канал в уже сохраненной рассылке нельзя. При передаче соответствующих параметров — `send_channel`, `content_type`, `send_email_content`, `send_sms_content`, `send_push_content` — запрос не вернет ошибку, но и не обновит канал отправки.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "virtual_sender_id": 5,
    "content": [
        {
            "from_email": "regular@altcraft.com",
            "from_name": "Roman",
            "message_id": 14,
            "replyto_email": "regular@altcraft.com",
            "replyto_name": "Romana",
            "subject": "Hi!"
        }
    ],
    "custom_tracking_domains": [
        "trk.example.com",
        "trk.example.org"
    ],
    "data_id": {{db_id}},
    "data_type": "list",
    "id": 93,
    "is_active": true,
    "limit": 1,
    "name": "Regular campaign 1",
    "resource_id": 20,
    "start_schedule": {
        "start_hours": [
            1,
            2,
            3,
            4
        ],
        "start_mins": [
            1,
            2,
            3,
            4
        ],
        "start_mdays": [
            1
        ],
        "start_months": [
            12
        ],
        "start_wdays": [],
        "start_years": [
            2024
        ]
    }
}
```

```
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <virtual_sender_id>5</virtual_sender_id>
    <content array='true'>
        <from_email>example@example.com</from_email>
        <from_name>Roman</from_name>
        <message_id>14</message_id>
        <replyto_email>example@example.com</replyto_email>
        <replyto_name>Romana</replyto_name>
        <subject>Hi!</subject>
    </content>
    <custom_tracking_domains>trk.example.com</custom_tracking_domains>
    <custom_tracking_domains>trk.example.org</custom_tracking_domains>
    <data_id>1</data_id>
    <data_type>list</data_type>
    <id>93</id>
    <is_active>true</is_active>
    <limit>1</limit>
    <name>Regular campaign 1</name>
    <resource_id>20</resource_id>
    <start_schedule>
        <start_hours>1</start_hours>
        <start_hours>2</start_hours>
        <start_hours>3</start_hours>
        <start_mins>1</start_mins>
        <start_mins>2</start_mins>
        <start_mins>3</start_mins>
        <start_mins>4</start_mins>
        <start_mdays array='true'>1</start_mdays>
        <start_months array='true'>12</start_months>
        <start_wdays array='true'></start_wdays>
        <start_years array='true'>2024</start_years>
    </start_schedule>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 93
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>93</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор рассылки

Запустить регулярную рассылку

Описание

Мгновенно запускает регулярную рассылку, если она активна. Производит однократную отправку.

Параметрами `override_source_type` и `override_source_id` можно изменить аудиторию рассылки для запуска.

Параметр `override_limit` определяет лимит на отправленные сообщения для текущего запуска рассылки.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/regulars/start`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	12	Да	Идентификатор рассылки
override_source_type	enum	“list” или “segment”	Нет	Перезаписать источник данных
override_source_id	int	5	Нет	Перезаписать идентификатор данных
override_limit	int	56	Нет	Перезаписать лимит на старт рассылки
content	JSON object	`{ "field": "value" }`	Нет	Контент для переменных apicontent в сообщениях.
attach	JSON array	[]	Нет	File object

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 94,
    "override_source_id": 1,
    "override_source_type": "list"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>93</id>
    <override_source_id>1</override_source_id>
    <override_source_type>list</override_source_type>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить список триггерных рассылок

Описание

Получить список триггерных рассылок.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/triggers/list/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	`"abcdefghijklmnqrstuvwxyz"`	Да	API-токен
from_id	int	`11768`	Нет	Идентификатор первой рассылки в списке. По умолчанию – `1`.
limit	int	`10`	Нет	Лимит на количество возвращаемых данных. По умолчанию – `1000`.
format	string	`"json"`	Нет	Формат возвращаемых данных. По умолчанию – `"json"`.
is_active	bool	`true`	Нет	Статус рассылки
ui_tags	[]String	`[ "tag1", "tag2" ]`	Нет	Список тегов для поиска. По умолчанию – `[]`.
groups	[]String	`[ "group1", "group2" ]`	Нет	Список групп для поиска. По умолчанию – `[]`.
groups_id	[]int	`[ 0, 1 ]`	Нет	Список Id групп для поиска. По умолчанию – `[]`.
attributes	JSON object	`"attributes": \{ "season": \{ "options": \["spring_2024", "summer_2024"\] \} \}`	Нет	Значения атрибутов для поиска

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "limit": 2
}
```

XML

```
<xml>
   <token>abcdefghijklmnqrstuvwxyz</token>
   <limit>2</limit>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "id": 12,
            "name": "New trigger campaign 3",
            "type": "api_call",
            "is_active": true,
            "for_confirm": false,
            "ui_tags": [],
            "groups": [
                {
                    "id": 0,
                    "name": "main"
                }
            ],
            "attributes": {
                "season": "spring_2024",
                "branch": "marketing"
            }
        },
        {
            "id": 14,
            "name": "New planetary campaign",
            "type": "api_call",
            "is_active": true,
            "for_confirm": false,
            "ui_tags": [],
            "groups": [
                {
                    "id": 1,
                    "name": "subaccount"
                }
            ],
            "providers": [
                "google",
                "mozilla",
                "safari",
                "safari-native",
                "firefox-firebase",
                "chrome-firebase",
                "opera-firebase",
                "yandex-firebase",
                "samsung-firebase",
                "android-firebase",
                "ios-firebase",
                "ios-apns",
                "yandex-Appmetrica-device-id-ios",
                "yandex-Appmetrica-device-id-android",
                "yandex-Appmetrica-ios-ifa"
```

```
                "yandex-Appmetrica-ios-ifa",
                "yandex-Appmetrica-google-aid",
                "yandex-Appmetrica-android-token",
                "yandex-Appmetrica-ios-token",
                "mr",
                "android-huawei",
                "ios-huawei",
                "android-rustore"
            ]
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 41
}
```


XML

```xml
<xml>
    <data>
        <for_confirm>false</for_confirm>
        <groups>
            <id>0</id>
            <name>main</name>
        </groups>
        <attributes>
            <season>spring_2024</season>
            <branch>marketing</branch>
        </attributes>
        <id>12</id>
        <is_active>true</is_active>
        <name>New trigger campaign 3</name>
        <type>api_call</type>
    </data>
    <data>
        <for_confirm>false</for_confirm>
        <groups>
            <id>1</id>
            <name>subaccount</name>
        </groups>
        <id>14</id>
        <is_active>true</is_active>
        <name>New planetary campaign</name>
        <providers>google</providers>
        <providers>mozilla</providers>
        <providers>safari</providers>
        <providers>safari-native</providers>
        <providers>firefox-firebase</providers>
        <providers>chrome-firebase</providers>
        <providers>opera-firebase</providers>
        <providers>yandex-firebase</providers>
        <providers>samsung-firebase</providers>
        <providers>android-firebase</providers>
        <providers>ios-firebase</providers>
        <providers>ios-apns</providers>
        <providers>yandex-Appmetrica-device-id-ios</providers>
        <providers>yandex-Appmetrica-device-id-android</providers>
        <providers>yandex-Appmetrica-ios-ifa</providers>
        <providers>yandex-Appmetrica-google-aid</providers>
        <providers>yandex-Appmetrica-android-token</providers>
        <providers>yandex-Appmetrica-ios-token</providers>
        <providers>mr</providers>
        <providers>android-huawei</providers>
        <providers>ios-huawei</providers>
        <providers>android-rustore</providers>
        <type>api_call</type>
    </data>
```

```
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <total_count>41</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор рассылки
name	string	Название рассылки
type	string	Тип триггера
for_confirm	bool	Имеет ли рассылка сообщения со ссылкой подтверждения
is_active	bool	Статус рассылки
ui_tags	[]String	Список тегов
groups	[]String	Список групп
attributes	object	Атрибуты, назначенные рассылке, и их значения

Получить информацию о триггерной рассылке

Описание

Получить информацию о триггерной рассылке.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/triggers/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор рассылки
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 12
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>12</id>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "call_url":
"trk.test.test.altcraft.com/trg/465ea6fedf0a75fafe1a413943aae3696d95939038311994/<email|md5
|xxh>",
    "content": [
      {
        "from_email": "",
        "from_name": "",
        "message_id": 41,
        "replyto_email": "",
        "replyto_name": "",
        "subject": ""
      }
    ],
    "send_channel": "email",
    "custom_tracking_domains": [],
    "id": 7,
    "is_active": true,
    "attributes": {
        "season": "spring_2024",
        "branch": "marketing"
    },
    "is_random_tracking_prefix": false,
    "list_id": 20,
    "name": "API call trigger",
    "resource_id": 44,
    "skip_minutes": 0,
    "subid": "API",
    "trigger_type": "api_call",
    "virtual_sender_id": 98
  },
  "error": 0,
  "error_text": "Successful operation"
}
```


XML

```
<xml>
    <data>

<call_url>http://trk.test.test.altcraft.com:8095/trg/65512f8ec196edd63e50585b4bfd377365512f
8ec196edd62f3de6b87e2969e3/
            <email|md5|xxh>
            </call_url>
            <channel>0</channel>
            <content>
                <message_id>8</message_id>
            </content>
            <for_confirm>false</for_confirm>
            <id>12</id>
            <is_active>true</is_active>
            <attributes>
                <season>spring_2024</season>
                <branch>marketing</branch>
            </attributes>
            <is_new_ch>false</is_new_ch>
            <is_new_contact>false</is_new_contact>
            <is_new_sub>false</is_new_sub>
            <is_random_tracking_prefix>false</is_random_tracking_prefix>
            <is_status_changed>false</is_status_changed>
            <list_id>1</list_id>
            <name>New trigger campaign 3</name>
            <policy_id>0</policy_id>
            <resource_id>4</resource_id>
            <send_channel>email</send_channel>
            <skip_minutes>0</skip_minutes>
            <skip_unique_checking>false</skip_unique_checking>
            <start_schedule>
                <losers>false</losers>
                <profile>false</profile>
            </start_schedule>
            <subscribe_resource_id>0</subscribe_resource_id>
            <trigger_type>api_call</trigger_type>
            <unconditional_sending>false</unconditional_sending>
            <update_time>2024-02-15T20:45:00.705Z</update_time>
            <updated_user>roman</updated_user>
            <virtual_sender_id>0</virtual_sender_id>
        </data>
        <error>0</error>
        <error_text>Successful operation</error_text>
    </xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор рассылки
name	string	Название рассылки
is_active	bool	Статус рассылки
content	JSON array	Content object
trigger_type	string	Тип триггера
send_channel	string	Канал отправки
skip_minutes	int	Подождать (минуты) перед активацией триггера
resource_id	int	Идентификатор ресурса
for_confirm	bool	Использует ли рассылка сообщения со ссылкой подтверждения
policy_id	int	Идентификатор политики отправки
descr	string	Описание рассылки
push_sending_mode	string	Режим отправки push-сообщений (только для push-каналов)
attributes	object	Атрибуты, назначенные рассылке, и их значения
Smart настройки (только для smart-аккаунтов)		
virtual_sender_id	int	Идентификатор виртуального сендера
is_random_tracking_prefix	bool	Случайный префикс трекинг-домена
custom_tracking_domains	JSON array	Список трекинг-доменов
blind_carbon_copy	JSON array	Адреса для скрытой копии
unconditional_sending	bool	Отправка вне зависимости от статуса подписки

Тип триггера в зависимости от полей ответа

Поля, меняющиеся в зависимости от типа триггера (`trigger_type`)

Поле	Тип	Описание	api_call	api_import	action	update	pixel	subscribe
action	string	Тип запуска, основанный на действии	Нет	Нет	Да	Нет	Нет	Нет

Поле	Тип	Описание	api_call	api_import	action	update	pixel	subscribe

update_query	JSON array	Обновление полей профиля:
		{
		“fieldname”: “custom_ID”,
		“fieldtype”: “int”,
		“operator”: “equal”,
		“value”: “2000”
		}
			Нет	Нет	Нет	Да	Нет	Нет

link_name	string	Имя ссылки в сообщении	Нет	Нет	Да	Нет	Нет	Нет
message_id	int	Идентификатор сообщения для событий захвата	Нет	Нет	Да	Нет	Нет	Нет
list_id	int	Идентификатор базы данных	Да	Да	Нет	Да	Нет	Нет
subscribe_resource_id	int	Идентификатор ресурса подписки, на которую реагирует триггер	Нет	Нет	Нет	Нет	Нет	Да
channel	string	Канал на который реагировать	Нет	Нет	Нет	Нет	Нет	Да
is_new_sub	bool	Реагировать на новый ресурс	Нет	Нет	Нет	Нет	Нет	Да
is_new_ch	bool	Реагировать на новый канал в ресурс	Нет	Нет	Нет	Нет	Нет	Да
is_new_contact	bool	Реагировать на новый контакт в канале	Нет	Нет	Нет	Нет	Нет	Да
is_status_changed	bool	Реагировать на смену статуса на перечисленные в `trigger_statuses`	Нет	Нет	Нет	Нет	Нет	Да
trigger_statuses	array(int)	Статусы новых/измененных подписок, на которые реагирует триггер	Нет	Нет	Нет	Нет	Нет	Да
pixel_id	int	Идентификатор пикселя для захвата событий	Нет	Нет	Нет	Нет	Да	Нет
pixel_target	string	Цель пикселя для захвата событий	Нет	Нет	Нет	Нет	Да	Нет

trigger_type: action

Его эквивалент в пользовательском интерфейсе “Тип триггера”: “При выполнении подписчиком выбранного действия”.

Его дополняют три поля.

Два основных: `"action": "<action name>"` и `"message_id": 123`.

И поле `"link_name": "String"`, которое появляется вместе с `"action": "click_sms"` и `"action": "clicklink_sms"`.

- `"action": "open"` — Открыт Email.
- `"action": "read"` — Прочитан Email.
- `"action": "click"` — Любой клик в Email.
- `"action": "confirm"` — Клик на Confirm ссылку.
- `"action": "click_sms"` — Любой клик в SMS.
- `"action": "clicklink"` — Клик по ссылке в Email (здесь также требуется поле `"link_name": "String"`).
- `"action": "clicklink_sms"` — Клик по ссылке в SMS (здесь также требуется поле `"link_name": "String"`).

trigger_type: api_import

Его эквивалент в пользовательском интерфейсе “Тип триггера”: “При импорте подписчика через API”.

Его дополняет поле `"list_id": 32` — ID листа, для которого создан триггер.

trigger_type: pixel

Его эквивалент в пользовательском интерфейсе “Тип триггера”: “Когда пиксель активирован”.

Его дополняют два поля:

- `"pixel_id": 64` — в интерфейсе “Выберите пиксель”.
- `"pixel_target": "Target-name"` — в интерфейсе “Цель”.

trigger_type: update

Его эквивалент в пользовательском интерфейсе “Тип триггера”: “Когда какое либо поле изменено”.

Его дополняют два поля:

- `"list_id": 32` — ID листа, для которого создан триггер.
- `"update_query": {"fieldname": "_postal_code", "fieldtype": "unicode", "operator": "equal", "value": "800028"}` — условие срабатывания триггера.

Запрос обновления

- `"fieldname": "_postal_code"` — имя стандартного или дополнительного поля в листе. Дополнительные поля отображаются под своим именем. Перевод стандартных полей находится на странице Внутренние названия стандартных полей.
- `"fieldtype": "unicode"` — тип данных в поле, зависит от `"fieldname"`. Запрос поля листа показывает список стандартных (default) и дополнительных (custom) полей. Также там есть system поля, но они здесь не используются.
- `"operator": "equal"` — логический оператор.
- `"value": "800028"` — значение, которое должно принять поле при обновлении подписчика.
- `"submatch": "object_key"` — путь к свойству, значение которого должно измениться, чтобы сработал триггер. Используется только с типами полей *object* и *arr_object*.

Типы полей

Тип поля можно узнать запросом поля листа.

- `unicode` — Строка, `"abc"`

- `int` — Число, `-1234`
- `bool` — Boolean, `true`
- `date` — Дата (в виде строки), `"2000-10-24T00:00:00.000Z"`
- `tags` — Теги (список строк), `["T1", "Tag2"]`
- `enum` — Список, представляет собой структуру key-value `{"item_one": 1, "item_two": 2}`
- `object` — Объект, `{"A": "B", "C": "D"}`
- `arr_object` — Массив объектов, `[{"A": 1},{"B": 2}]`

Операторы запроса

Оператор	Описание	Line	Number	IP	Enum	Date	Boolean	Tags	Object	Arr_object
“equal”	если он станет равным	+	+	+	+	+	+	-	-	-
“not_equal”	если он станет не равным	+	+	+	+	+	+	-	-	-
“not_exists_or_empty”	если он не существует или пуст	+	+	+	+	+	+	-	+	+
“not_empty_or_exists”	если не пустой или существует	+	+	+	+	+	+	-	+	+
“tag_added_any”	если один из тегов добавлен	-	-	-	-	-	-	+	-	-
“tag_removed_any”	если один из тегов удален	-	-	-	-	-	-	+	-	-
“equal_property”	если значение указанного свойство станет равным	-	-	-	-	-	-	-	+	+
“not_equal_property”	если значение указанного свойство станет не равным	-	-	-	-	-	-	-	+	+
“added_element”	если в массив добавлен новый элемент	-	-	-	-	-	-	-	-	+
“deleted_element”	если из массива удален один из элементов	-	-	-	-	-	-	-	-	+

Добавить триггерную рассылку

Описание

Добавить триггерную рассылку.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/triggers/add/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"csv"	Нет	Формат возвращаемых данных по умолчанию – "json"
name	string	"New Awesome Campaign"	Да	Название рассылки
is_active	bool	true	Нет	Статус рассылки

content	JSON array	[
		  \{
		
		"from_email"\: "",
		
		"from_name"\: "",
		
		"message_id"\: 12,
		
		"replyto_email"\: "",
		
		"replyto_name"\: "",
		
		"subject"\: "Weather alert"
		  \}
		]
			Да	Контент, присоединенный к рассылке

subid	string	"Adv_1"	Нет	Subid рассылки
trigger_type	string	"api_call"	Да	Тип триггера
send_channel	string	"email" "sms" "push"	Да	Канал отправки. Если в запросе также передаются другие параметры, связанные с каналом (`content_type`, `send_email_content`, `send_sms_content`, `send_push_content`), рассылка будет использовать канал, указанный в параметре `send_channel`.
content_type	string	"email"	Нет	Параметр устарел. При его передаче рассылка будет использовать email-канал для отправки.
send_email_content	bool	false	Нет	Параметр устарел. При его передаче рассылка будет использовать email-канал для отправки.
send_sms_content	bool	false	Нет	Параметр устарел. При его передаче рассылка будет использовать email-канал для отправки.
send_push_content	bool	true	Нет	Параметр устарел. При его передаче рассылка будет использовать email-канал для отправки.
skip_minutes	int	0	Нет	Подождать X минут перед активацией рассылки

resource_id	int	24	Да	Идентификатор ресурса
policy_id	int	1	Нет	Идентификатор политики отправки
push_sending_mode	string	"fanout" (по умолчанию) "last"	Нет	Режим отправки push-сообщений (только для push-каналов)
attributes	object	`"season":"2024_summer"`	Нет	Атрибуты, назначенные рассылке, и их значения
Smart настройки				
virtual_sender_id	int	12	Нет	Идентификатор виртуального сендера **Только для smart-аккаунтов**
is_random_trk_prefix	bool	false	Нет	Случайный префикс трекинг-домена **Только для smart-аккаунтов**
custom_trk_domain	JSON array	`["trk.example.com", "trk2.example.com"]`	Нет	Список произвольных трекинг-доменов **Только для smart-аккаунтов**
blind_carbon_copy	JSON array	`["john@example.com", "george@example.com"]`	Нет	Адреса для скрытой копии **Только для smart-аккаунтов**
unconditional_sending	bool	true	Нет	Отправка вне зависимости от статуса подписки **Только для smart-аккаунтов**

* Опция доступна только при использовании Altcraft MP по лицензии (On-Premise). Для активации этой возможности в файл конфигурации `main.json` в свойство `EXTRAS` нужно добавить значение `"unc_campaign"`.

Параметры, связанные с типом триггера

Поле	Тип	Описание	api_call	api_import	action	update	pixel	subscribe
action	string	Тип запуска, основанный на действии	Нет	Нет	Да	Нет	Нет	Нет

Поле	Тип	Описание	api_call	api_import	action	update	pixel	subscribe

update_query	JSON array	Обновление полей профиля:
		{
		“fieldname”: “custom_ID”,
		“fieldtype”: “int”,
		“operator”: “equal”,
		“value”: “2000”
		}
			Нет	Нет	Нет	Да	Нет	Нет

link_name	string	Имя ссылки в сообщении	Нет	Нет	Да	Нет	Нет	Нет
message_id	int	Идентификатор шаблона сообщения для событий захвата	Нет	Нет	Да	Нет	Нет	Нет
list_id	int	Идентификатор базы данных	Да	Да	Нет	Да	Нет	Нет
subscribe_resource_id	int	Идентификатор ресурса подписки, на которую реагирует триггер	Нет	Нет	Нет	Нет	Нет	Да
channel	string	Канал на который реагировать	Нет	Нет	Нет	Нет	Нет	Да
is_new_sub	bool	Реагировать на новый ресурс	Нет	Нет	Нет	Нет	Нет	Да
is_new_ch	bool	Реагировать на новый канал в ресурс	Нет	Нет	Нет	Нет	Нет	Да
is_new_contact	bool	Реагировать на новый контакт в канале	Нет	Нет	Нет	Нет	Нет	Да
is_status_changed	bool	Реагировать на смену статуса на перечисленные в `trigger_statuses`	Нет	Нет	Нет	Нет	Нет	Да
trigger_statuses	array(int)	Статусы новых/измененных подписок, на которые реагирует триггер	Нет	Нет	Нет	Нет	Нет	Да
pixel_id	int	Идентификатор пикселя для захвата событий	Нет	Нет	Нет	Нет	Да	Нет

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "name": "New trigger campaign №1",
  "content": [
    {
      "subject": "Maybe very important message",
      "from_email": "",
      "from_name": "James",
      "replyto_email": "",
      "replyto_name": "",
      "message_id": 14
    }
  ],
  "trigger_type": "api_call",
  "send_channel": "email",
  "list_id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <name>New trigger campaign №1</name>
    <content array='true'>
        <from_name>James</from_name>
        <message_id>15</message_id>
        <subject>Maybe very important message</subject>
    </content>
    <trigger_type>api_call</trigger_type>
    <send_channel>email</send_channel>
    <list_id>1</list_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 95,
        "trigger_link":
"http://trk.test.test.altcraft.com:8095/trg/65512f8ec196edd6ac71fa5d068d355f65512f8ec196edd
62f3de6b87e2969e3/<email|md5|xxh>"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>96</saved_id>

<trigger_link>http://trk.test.test.altcraft.com:8095/trg/65512f8ec196edd6fc63524eba02bff065
512f8ec196edd62f3de6b87e2969e3/
            <email|md5|xxh>
            </trigger_link>
        </data>
        <error>0</error>
        <error_text>Successful operation</error_text>
    </xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор новой рассылки
trigger_link	string	Только если `trigger_type = "api_call"`

Обновить триггерную рассылку

Описание

Изменить настройки существующей триггерной рассылки.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/triggers/update/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“csv”	Нет	Формат возвращаемых данных по умолчанию – “json”
id	int	1	Да	Идентификатор триггера
name	string	“New Awesome Campaign”	Да	Название рассылки
is_active	bool	true	Нет	Статус рассылки

content	JSON array	[
		{
		“from_email”: “”,
		“from_name”: “”,
		“message_id”: 12,
		“replyto_email”: “”,
		“replyto_name”: “”,
		“subject”: “Weather alert”
		}
		]
			Нет	Контент, присоединенный к рассылке

subid	string	“Adv_1”	Нет	Subid рассылки
trigger_type	string	“api_call”	Да	Тип триггера
skip_minutes	int	0	Нет	Подождать X минут перед активацией рассылки
resource_id	int	24	Да	Идентификатор ресурса
policy_id	int	1	Нет	Идентификатор политики отправки
push_sending_mode	string	“fanout” (по умолчанию) “last”	Нет	Режим отправки push-сообщений (только для push-каналов)

Параметр	Тип	Пример	Обязательный	Описание
attributes	object	`"season":"2024_summer"`	Нет	Атрибуты, назначенные рассылке, и их значения
Smart настройки				
virtual_sender_id	int	12	Нет	Идентификатор виртуального сендера **Только для smart-аккаунтов**
is_random_trk_prefix	bool	false	Нет	Случайный префикс трекинг-домена **Только для smart-аккаунтов**
custom_trk_domain	JSON array	`["trk.example.com", "trk2.example.com"]`	Нет	Список трекинг-доменов **Только для smart-аккаунтов**
blind_carbon_copy	JSON array	`["john@example.com", "george@example.com"]`	Нет	Адреса для скрытой копии **Только для smart-аккаунтов**
unconditional_sending	bool	true	Нет	Отправка вне зависимости от статуса подписки **Только для smart-аккаунтов**

NOTE

Изменять канал в уже сохраненной рассылке нельзя. При передаче соответствующих параметров — `send_channel`, `content_type`, `send_email_content`, `send_sms_content`, `send_push_content` — запрос не вернет ошибку, но и не обновит канал отправки.

Параметры, связанные с типом триггера

Поле	Тип	Описание	api_call	api_import	action	update	pixel	subscribe
action	string	Тип запуска, основанный на действии	Нет	Нет	Да	Нет	Нет	Нет

update_query	JSON array	Обновление полей профиля:
		`{`
		`“fieldname”: “custom_ID”,`
		`“fieldtype”: “int”,`
		`“operator”: “equal”,`
		`“value”: “2000”`
		`}`
			Нет	Нет	Нет	Да	Нет	Нет

link_name	string	Имя ссылки в сообщении	Нет	Нет	Да	Нет	Нет	Нет
message_id	int	Идентификатор шаблона сообщения для событий захвата	Нет	Нет	Да	Нет	Нет	Нет

Поле	Тип	Описание	api_call	api_import	action	update	pixel	subscribe
list_id	int	Идентификатор базы данных	Да	Да	Нет	Да	Нет	Нет
subscribe_resource_id	int	Идентификатор ресурса подписки, на которую реагирует триггер	Нет	Нет	Нет	Нет	Нет	Да
channel	string	Канал на который реагировать	Нет	Нет	Нет	Нет	Нет	Да
is_new_sub	bool	Реагировать на новый ресурс	Нет	Нет	Нет	Нет	Нет	Да
is_new_ch	bool	Реагировать на новый канал в ресурс	Нет	Нет	Нет	Нет	Нет	Да
is_new_contact	bool	Реагировать на новый контакт в канале	Нет	Нет	Нет	Нет	Нет	Да
is_status_changed	bool	Реагировать на смену статуса на перечисленные в `trigger_statuses`	Нет	Нет	Нет	Нет	Нет	Да
trigger_statuses	array(int)	Статусы новых/измененных подписок, на которые реагирует триггер	Нет	Нет	Нет	Нет	Нет	Да
pixel_id	int	Идентификатор пикселя для захвата событий	Нет	Нет	Нет	Нет	Да	Нет

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 42,
  "name": "New trigger campaign update",
  "content": [
    {
      "subject": "Maybe not very important message",
      "from_email": "",
      "from_name": "Alex",
      "replyto_email": "",
      "replyto_name": "",
      "message_id": 8
    }
  ],
  "trigger_type": "api_call",
  "list_id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>42</id>
    <name>New trigger campaign 3</name>
    <content array='true'>
        <from_name>James</from_name>
        <message_id>15</message_id>
        <subject>Maybe very important message</subject>
    </content>
    <trigger_type>api_call</trigger_type>
    <list_id>1</list_id>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "saved_id": 42,
    "trigger_link":
"http://trk.example.com:8095/trg/3a345a07f01657e7966566e657a34d023a345a07f01802e7/<email|md
5|xxh>"
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <data>
        <saved_id>42</saved_id>

<trigger_link>http://trk.example.com:8095/trg/3a345a07f01657e7966566e657a34d023a345a07f0180
2e7/&lt;email|md5|xxh&gt;</trigger_link>
    </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	12
trigger_link	string	Только если `trigger_type = "api_call"`

Запуск триггерной рассылки (API call)

Высокоприоритетные рассылки, мгновенно запускаемые с помощью API запроса.

API call trigger отправляет сообщения профилям из своей базы данных каждый раз, когда приходит соответствующий API запрос.

> **TIP**
>
> Чтобы отправлять триггер одновременно с импортом профиля клиента, используйте этот запрос.

Версия 1.1

Описание

Имеет несколько преимуществ перед версией 1.0:

- Использование API-контента позволяет менять содержание триггерного сообщения при каждой отправке.
- Больше вариантов выбора подписчиков: добавлен поиск по id профиля, дополнительному полю, а также запросом к базе данных.
- Использует API-токен, который передаётся в JSON POST запросе, что более безопасно, чем в ссылке.
- Использует статическую ссылку активации `/api/v1.1/campaigns/triggers/start/` вместо динамической, зависящей от базы данных.
- Добавлена возможность отправки нескольким подписчикам за один API запрос (только через запрос к базе данных).

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/triggers/start/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	5511	Да	Идентификатор триггера
content	JSON object	`{ "field": "value" }`	Нет	Контент для переменных apicontent в сообщениях.

attach	JSON array	`[]`	Нет	File object
custom_data	JSON object	`{ "x": "y"}`	Нет	Добавить некоторые данные, которые будут использоваться в "захвате событий".
Поиск профилей подписчиков				
matching	string	"email" "email_profile" "phone" "profile_id" "custom"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию — email. </td>
email	string	"john@example.com"	"matching":"email" — по профилю и подпискам "matching":"email_profile" — только по профилю </td> <td>Email-адрес</td>	
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. **TIP: Доступен поиск по сочетанию значений ** поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Да	Идентификатор ресурса для поиска по email и phone

md5	string	"d4c74594d841139328695756648b6bd6"	Если нет свойства "matching"	MD5-хеш подписчика
xxh	string	"d4c74594d841139328695756648b6bd6"	Если нет свойства "matching"	xxh-хеш подписчика

custom_request	JSON object	{
		"email": {
		"$in": [
		"asd@dsfsd.com",
		"zxcv@sdfsd.com"
		]
		}
		}
			Если нет свойства "matching"	MongoDB-запрос по полям профиля

subscription_filter	JSON object	Пример для push:
		"subscription_filter": {
		"email": "example@example.com",
		"phone": "+79106135133",
		"subscription_id": "CAREFULWITHTHATAXE",
		"provider": "Chrome",
		"not_strict": true
		}
		
		Пример для email:
		"subscription_filter": {
		"email": "example@example.com",
		"not_strict": true
		}
		
		Пример для sms:
		"subscription_filter": {
		"phone": "+79106135133",
		"not_strict": false
		}
			Нет	Выбирает конкретную подписку профиля для отправки.
				Не используется, если задан параметр **`custom_request`**
				
				`not_strict` определяет, что делать если подписка не найдена. В случае "true"
				будет использована любая подходящая подписка.
				</td>

Пример запроса

Отправить сообщение подписчику:

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 84,
  "matching": "profile_id",
  "profile_id": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>84</id>
  <matching>profile_id</matching>
  <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
</xml>
```

Отправить сообщения подписчикам, поле `Custom_ID` которых входит в список `["IpHicQquekzYkufl", "xzFKLXeaJkCYHuMb", "SRNOwHVZiWFItJhr"]`.

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 84,
  "custom_request":{"Custom_ID": {"$in": ["IpHicQquekzYkufl", "xzFKLXeaJkCYHuMb",
"SRNOwHVZiWFItJhr"]}}
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>84</id>
    <custom_request>{"Custom_ID": {"$in": ["IpHicQquekzYkufl", "xzFKLXeaJkCYHuMb",
"SRNOwHVZiWFItJhr"]}}</custom_request>
</xml>
```

Вставить текст `"New videos on your favorite channels"` на место переменной `{apicontent.email_title}`, если она используется в сообщении.

Приложить к сообщению файлы “日本.txt” и “US.txt”

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 96,
    "matching": "profile_id",
    "profile_id": "6606e08f10bcdd88297057a7",
    "attach": [
        {
            "data": "data:text/csv; base64,5LuK5pel44GvCg==",
            "name": "日本.txt"
        },
        {
            "data": "data:text/csv; base64,SEVMTE8K",
            "name": "US.txt"
        }
    ],
    "content": {
        "data_lines": [
            "Channel A",
            "Channel B"
        ],
        "email_title": "New videos on your favorite channels"
    }
}
```

XML

```
<xml>
<token>abcdefghijklmnqrstuvwxyz</token>
  <id>96</id>
  <matching>profile_id</matching>
  <profile_id>6606e08f10bcdd88297057a7</profile_id>
  <attach>
    <data>data:text/csv; base64,5LuK5pel44GvCg==</data>
    <name>日本.txt</name>
  </attach>
  <attach>
    <data>data:text/csv; base64,SEVMTE8K</data>
    <name>US.txt</name>
  </attach>
  <content>
    <data_lines>Channel A</data_lines>
    <data_lines>Channel B</data_lines>
    <email_title>New videos on your favorite channels</email_title>
  </content>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Версия 1.0

Описание

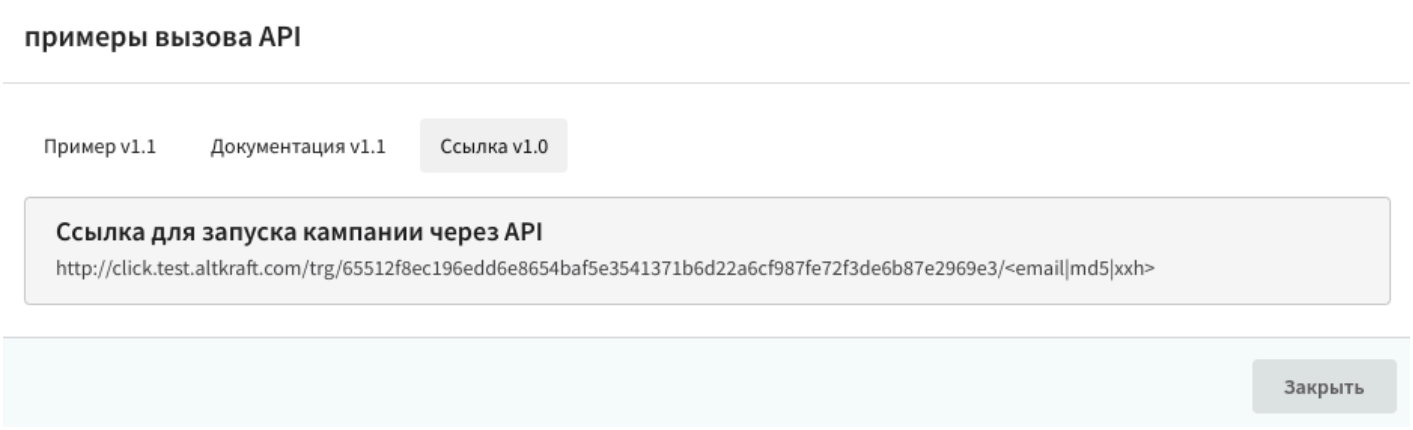
Методы: GET, POST

В этой версии каждый API триггер имеет уникальную ссылку, которая отображается после его создания.

`http://<your.tracking.domain>/trg/<base16_value_d1d93d9cca76a7c1742a2f23fbacd>/<email|md5|xxh>`

Она изменится, если поменять базу данных триггера.

Использование ссылки 1.0 позволяет отправлять триггер лишь одному профилю за запрос.



“email” является устаревшим параметром.

GET запрос

Просто отправляет сообщение.

Пример HTTP GET запроса по ссылке с XXH профиля.

`http://trk.example.org/trg/465ea6fedf0a75fa9ee0b5fb8f9d5fe89af34054909af8bb/dd78094ad0e36ca3`

POST JSON запрос

Используется та же самая ссылка, что и в GET.

`http://trk.example.org/trg/465ea6fedf0a75fa9ee0b5fb8f9d5fe89af34054909af8bb/dd78094ad0e36ca3`

Однако, в POST запрос можно добавить JSON данные, которые используются для того, чтобы сделать сообщение более персонифицированным и/или обновить данные профиля подписчика.

Например, если сообщение должно содержать Имя и Номер заказа подписчика, то в Request Payload нужно записать:

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 96,
    "matching": "profile_id",
    "profile_id": "6606e08f10bcdd88297057a7",
    "fields": [
        {
            "name": "FirstName",
            "value": "Jason",
            "option": 0
        },
        {
            "name": "Order",
            "value": "123abc456-78",
            "option": 1
        }
    ]
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>96</id>
  <matching>profile_id</matching>
  <profile_id>6606e08f10bcdd88297057a4</profile_id>
  <fields>
    <name>FirstName</name>
    <value>Jason</value>
    <option>0</option>
  </fields>
  <fields>
    <name>Order</name>
    <value>123abc456-78</value>
    <option>1</option>
  </fields>
</xml>
```

- `"name"` — название поля с данными профиля. Если такого поля в базе данных профиля нет, то в сообщение оно не попадёт.
- `"value"` — значение, которое нужно использовать в сообщении для этого поля. Формат *value* должен соответствовать формату поля в базе данных профиля.
- `"option"` — *(0 | 1 | 2)* определяет, как использовать данные из запроса.
- `"option": 0` — если в профиле заполнено поле (`"name"`), то в письмо попадёт значение из профиля, а значение из запроса *(“value”)* не используется.
- `"option": 1` — в письмо попадёт значение из запроса (`"value"`).
- `"option": 2` — в письмо попадёт значение из запроса (`"value"`). Обратите внимание, данные профиля в поле (`"name"`) будут заменены на значение (`"value"`) из запроса.

Импорт профиля + Отправка триггера

Описание

Добавление или обновление профиля с последующей отправкой триггерной рассылки.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/triggers/import_and_start/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
Импорт профиля клиента				
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
data	JSON object	`{ "_fname": "John", "_lname": "Doe" }`	Нет	Данные о профиле, в том числе подписки
skip_triggers	bool	true	Нет	Пропустить запуск триггеров. По умолчанию – false.
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки. По умолчанию – false.
detect_geo	bool	true	Нет	Включает автоопределение geo данных по полю _regip или _ip в data
Поиск профиля подписчика для обновления				
matching	string	"email" "email_profile" "phone" "profile_id" "custom"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию — email.

email	string	"john@example.com"	"matching":"email" — по профилю и подпискам "matching":"email_profile" — только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. **TIP** Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone
Запуск триггера				
trigger_id	int	13	Да	Идентификатор триггерной рассылкии для запуска
timeout	JSON object	`{ "days": 0, "hours": 0, "minutes": 5 }`	Нет	Задержка перед запуском триггера. Указываются дни, часы и минуты.
custom_data	JSON object	`{ "sending_time": "12:31:12" }`	Нет	Данные для захвата событий (action hooks)
content	JSON object	`{ "page": "registration_form" }`	Нет	Дополнительные данные для подстановки в шаблон сообщения переменной `{apicontent.fieldname}`

attach	JSON objects array	{
		"data": "data:text/csv; base64,SEVMTE8K",
		"name": "Attach.txt"
		}
			Нет	Файл для прикрепления к email сообщению (только для email триггеров)

Пример запроса

JSON

```
{
   "token": "abcdefghijklmnqrstuvwxyz",
   "matching": "email",
   "email": "example@example.com",
   "detect_geo": true,
   "data": {
      "_fname": "Olly",
      "_lname": "Lambert",
      "email": "example@example.com",
      "phones": [
         "+79000000000"
      ],
      "_bdate": "1990-02-22T21:00:00Z",
      "custom_field": "custom_value",
      "subscriptions": [
         {
            "channel": "email",
            "email": "example@example.com",
            "resource_id": 21
         },
         {
            "channel": "phone",
            "phone": "+79000000000",
            "resource_id": 21
         }
      ]
   },
   "trigger_id": 96,
   "timeout": {
         "days": 0,
         "hours": 0,
         "minutes": 5
   },
   "custom_data": {
      "sending_time": "12:31:12"
   },
     "attach": [
      {
         "data": "data:text/csv; base64,5LuK5pel44GvCg==",
         "name": "Attach.txt"
      }
   ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <matching>email</matching>
    <email>example@example.com</email>
    <detect_geo>true</detect_geo>
    <data>
        <_fname>Garry</_fname>
        <_lname>Sallivan</_lname>
        <_bdate>1990-02-22T21:00:00Z</_bdate>
        <email>example@example.com</email>
        <phones array='true'>+79000000027</phones>
        <custom_field>custom_value</custom_field>
        <subscriptions>
            <channel>email</channel>
            <email>testtriggers90@example.com</email>
            <resource_id>21</resource_id>
        </subscriptions>
        <subscriptions>
            <channel>phone</channel>
            <phone>+79000000780</phone>
            <resource_id>21</resource_id>
        </subscriptions>
    </data>
    <trigger_id>96</trigger_id>
    <timeout>
        <days>0</days>
        <hours>0</hours>
        <minutes>5</minutes>
    </timeout>
    <custom_data>
        <sending_time>12:31:12</sending_time>
    </custom_data>
    <attach array='true'>
        <data>data:text/csv; base64,5LuK5pel44GvCg==</data>
        <name>Attach.txt</name>
    </attach>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "",
    "profile_id": "54759eb3c090d83494e2d804"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <profile_id>660feee02984cf021df478f3</profile_id>
</xml>
```

CAUTION

Подписка на ресурс, по которой будет производиться отправка триггера, должна быть передана в объекте **data**.
Если нужной подписки не окажется среди переданных, API ответит следующей ошибкой:

```
{
    "error": 400,
    "error_text": "No active subscriptions passed for trigger resource"
}
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_id	string	Идентификатор профиля

Задание на массовую отправку триггера

Описание

Запуск триггерной рассылки для нескольких профилей.

Данный запрос является batch-версией запроса Запуск триггерной рассылки (API call).

Возвращает ответ с id таска, не дожидаясь выполнения.

> **NOTE**
>
> Тело запроса ограничено по умолчанию до 10мб, исключение: файлы — 100мб.

URL-адрес

Метод: POST

`http://example.com/api/v1.1/campaigns/triggers/start_batch_task`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных (по умолчанию – "json")
id	int	5511	Да	Идентификатор рассылки
attach	JSON array of File object	[]	Нет	File object
content	JSON object	`{ "H1": "Weather alert", "H2": "Welcome" }`	Нет	Контент для переменных apicontent в сообщениях.
custom_data	JSON object	`{ "field": "value" }`	Нет	Пользовательские параметры, возвращаются в хуках (Захват событий)

data	array	[{"email": "profile1@example.com"}, {"email": "profile2@example.com"}]	Да	Поле в базе данных и его значение, по которому будет произведен поиск нужного профиля.
Настройки поиска профиля				
matching	string	"email" "email_profile" "email_sub" "phone" "phone_sub" "profile_id" "push_sub" "custom" "custom_sub" "email_phone" "email_phone_sub"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию - email.
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю "matching":"email_sub" - только по подпискам	Email-адрес
phone	string	"+79000000000"	"matching":"phone" - по телефону из профиля и подписок "matching":"phone_sub" - по телефону из подписок"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска.

custom_request	JSON object	{
		"email": {
		"$in": [
		"asd@dsfsd.com",
		"zxcv@sdfsd.com"
		]
		}
		}
			Если нет свойства "matching"	MongoDB-запрос по полям профиля

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
task_id	string	Идентификатор задачи

Пример запроса с матчингом по email

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 96,
    "format": "json",
    "matching": "email",
    "data": [
        {
            "email": "example1@example.com"
        },
        {
            "email": "example2@example.com"
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>96</id>
    <format>xml</format>
    <matching>email</matching>
    <data>
        <email>example1@example.com</email>
    </data>
    <data>
        <email>example2@altcraft.com</email>
    </data>
</xml>
```

Пример запроса с матчингом по custom-полю

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 96,
    "format": "json",
    "skip_triggers": false,
    "matching": "custom",
    "field_name": "Custom_ID",
    "data": [
        {
            "field_value": "001"
        },
        {
            "field_value": "002"
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>96</id>
    <format>xml</format>
    <skip_triggers>false</skip_triggers>
    <matching>custom</matching>
    <field_name>Custom_ID</field_name>
    <data>
        <field_value>001</field_value>
    </data>
    <data>
        <field_value>002</field_value>
    </data>
</xml>
```

Пример ответа

JSON

```
{
        "error": 0,
        "error_text": "Successful operation",
        "task_id": "30421978-7f6d-4f39-913f-fb6cad8e2477"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <task_id>392b7e46-e4d9-4a07-959a-1e541138b684</task_id>
</xml>
```

Пример ответа при срабатывании ограничения запроса

JSON

```
{
    "error": 400,
    "error_debug": "http: request body too large",
    "error_text": "Invalid request"
}
```

XML

```
<xml>
  <error>400</error>
  <error_debug>http: request body too large</error_debug>
  <error_text>Invalid request</error_text>
</xml>
```

Задание на массовый импорт профилей + отправка триггера

Описание

Добавление или обновление нескольких профилей с последующей отправкой триггерной рассылки.

Данный запрос является batch-версией запроса Импорт профиля + Отправка триггера.

Возвращает ответ с id таска, не дожидаясь выполнения.

> **NOTE**
>
> Тело запроса ограничено по умолчанию до 10мб, исключение: файлы — 100мб.

URL-адрес

Метод: POST

`http://example.com/api/v1.1/campaigns/triggers/import_and_start_batch_task`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
Импорт профиля клиента				
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен

data	array	[
		{
		"data":
		{
		"_fna
		me": "John",
		"_lna
		me":
		"Lambert",
		"emai
		l":
		"profile4@exam
		ple.com",
		"phon
		es":
		"+79000000000"
		}
		},
		{
		"data":
		{
		"_fna
		me": "Tom",
		"_lna
		me": "Hard",
		"emai
		l":
		"profile3@exam
		ple.com",
		"phon
		es":
		"+790000000001
		"
		}
		}
		]
			нет	Данные о профиле, в том числе подписки.
				
				Каждый профиль — это отдельный JSON-объект "data".

skip_trig gers	bool	true	Нет	Пропустить запуск триггеров. По умолчанию – false.
skip_inv alid_sub scription s	bool	true	Нет	Пропустить невалидные подписки. По умолчанию – false.
detect_g eo	bool	true	Нет	Включает автоопределение geo данных по полю _regip или _ip в data
Настройки поиска профиля				

matching	string	"email" "email_profile" "phone" "profile_id" "custom"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. по умолчанию - email.
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. **TIP** Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Да	Идентификатор ресурса для поиска по email и phone
Запуск триггера				
trigger_id	int	13	Да	Идентификатор триггерной рассылки для запуска
timeout	JSON object	`{ "days": 0, "hours": 0, "minutes": 5 }`	Нет	Задержка перед запуском триггера. Указываются дни, часы и минуты.
custom_data	JSON object	`{ "sending_time": "12:31:12" }`	Нет	Данные для Захвата событий (Action hooks)

content	JSON object	`{ "page": "registration_form" }`	Нет	Дополнительные данные для подстановки в шаблон сообщения переменной `{apicontent.fieldname}`
attach	JSON objects array	`{ "data": "data:text/csv; base64,SEVMTE8K", "name": "Attach.txt" }`	Нет	Файл для прикрепления к email сообщению (только для email триггеров)

Возвращаемые параметры

Параметр	Тип	Описание
error_text	string	Текст ошибки
error	int	Код ошибки
task_id	string	Идентификатор задачи

Пример запроса с матчингом по email

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "matching": "email",
    "data": [
        {
            "data": {
                "_fname": "Victoria",
                "_lname": " Beckham",
                "email": "example1@yopmail.com",
                "phones": "+79000000000",
                "subscriptions": [
                    {
                        "channel": "email",
                        "email": "example2@yopmail.com",
                        "resource_id": 21
                    },
                    {
                        "channel": "phone",
                        "phone": "+79000005478",
                        "resource_id": 20
                    }
                ]
            }
        },
        {
            "data": {
                "_fname": "David",
                "_lname": " Beckham",
                "email": "example3@yopmail.com",
                "phones": "+79000000001",
                "subscriptions": [
                    {
                        "channel": "email",
                        "email": "example14@yopmail.com",
                        "resource_id": 21
                    },
                    {
                        "channel": "phone",
                        "phone": "+79000000000",
                        "resource_id": 20
                    }
                ]
            }
        }
    ],
    "trigger_id": 96,
    "timeout": {
        "days": 0,
        "hours": 0
```

```
        "hours": 0,
        "minutes": 5
    },
    "content": {
        "page": "registration_form"
    },
    "attach": [
        {
            "data": "data:text/csv; base64,5LuK5pel44GvCg==",
            "name": "Attach.txt"
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <matching>email</matching>
    <data>
        <_fname>Brooklyn</_fname>
        <_lname>Beckham</_lname>
        <email>example3@yopmail.com</email>
        <phones>+79000849376</phones>
        <subscriptions>
            <channel>email</channel>
            <email>example9@yopmail.com</email>
            <resource_id>21</resource_id>
        </subscriptions>
        <subscriptions>
            <channel>phone</channel>
            <phone>+79000005475</phone>
            <resource_id>20</resource_id>
        </subscriptions>
    </data>
    <data>
        <_fname>Romeo</_fname>
        <_lname>Beckham</_lname>
        <email>example32@yopmail.com</email>
        <phones>+79000000651</phones>
        <subscriptions>
            <channel>email</channel>
            <email>@yopmail.com</email>
            <resource_id>21</resource_id>
        </subscriptions>
        <subscriptions>
            <channel>phone</channel>
            <phone>+79000880000</phone>
            <resource_id>20</resource_id>
        </subscriptions>
    </data>
    <trigger_id>96</trigger_id>
    <timeout>
        <days>0</days>
        <hours>0</hours>
        <minutes>5</minutes>
    </timeout>
    <content>
        <page>registration_form</page>
    </content>
    <attach array='true'>
        <data>data:text/csv; base64,5LuK5pel44GvCg==</data>
        <name>Attach.txt</name>
    </attach>
```

```
    </attach>
</xml>
```

Пример запроса с матчингом по custom-полю

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "trigger_id": 96,
    "format": "json",
    "skip_triggers": false,
    "detect_geo": true,
    "matching": "custom",
    "field_name": "Custom_id",
    "data": [
        {
            "data": {
                "_fname": "Tom",
                "_lname": "Cruise",
                "email": "example@example.com",
                "phones": "+79000000000",
                "Custom_id": "0002"
            }
        },
        {
            "data": {
                "_fname": "Lili",
                "_lname": "Hard",
                "email": "example@example.com",
                "phones": "+79000000001",
                "Custom_id": "0001"
            }
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <trigger_id>96</trigger_id>
    <format>xml</format>
    <skip_triggers>false</skip_triggers>
    <detect_geo>true</detect_geo>
    <matching>custom</matching>
    <field_name>Custom_ID</field_name>
    <data>
        <_fname>Viola</_fname>
        <_lname>Lambert</_lname>
        <email>example1@example.com</email>
        <phones>+79000000000</phones>
        <Custom_ID>0002</Custom_ID>
    </data>
    <data>
        <_fname>Will</_fname>
        <_lname>Hard</_lname>
        <email>example2@example.com</email>
        <phones>+79000000001</phones>
        <Custom_ID>0001</Custom_ID>
    </data>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "task_id": "1c1a5861-6d4c-4d6f-bd6a-3ca4f46b217d"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <task_id>30421978-7f6d-4f39-913f-fb6cad8e2477</task_id>
</xml>
```

Пример ответа при срабатывании ограничения запроса

JSON

```
{
        "error":400,
        "error_debug":"http: request body too large",
        "error_text":"Invalid request"
}
```

XML

```
<xml>
    <error>400</error>
    <error_text>Invalid request</error_text>
</xml>
```

Проверка статуса задания

URL-адрес

Метод: POST

`https://example.com/api/v1.1/tasks/status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
task_id	string	“task_b2f9fk6a-2479-4a63-b592-f15d4e3b69c5”	Да	Идентификатор задания

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "task_id": "task_b2f9fk6a-2479-4a63-b592-f15d4e3b69c5"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <task_id>d521abe0-1abf-4e1d-b247-a790be66a424</task_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "status": "completed",
        "progress": 100,
        "result": {
            "error": 0,
            "error_text": "",
            "result": [
                {
                    "profile_id": "61e947333f62fc26d384452c"
                },
                {
                    "profile_id": "61e947333f62fc26d384452e"
                }
            ]
        }
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <progress>100</progress>
        <result>
            <captured>0</captured>
            <status>ok</status>
        </result>
        <status>completed</status>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Массовая отправка триггера

Описание

Запуск триггерной рассылки для нескольких профилей.

Данный запрос является batch-версией запроса Запуск триггерной рассылки (API call).

> **NOTE**
>
> Тело запроса ограничено по умолчанию до 10мб, исключение: файлы — 100мб.

URL-адрес

Метод: POST

`http://example.com/api/v1.1/campaigns/triggers/start_batch`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных (по умолчанию – "json")
trigger_id	int	1	Да	Идентификатор рассылки
attach	JSON array of File object	`[]`	Нет	File object
content	JSON object	`{ "H1": "Weather alert", "H2": "Welcome" }`	Нет	Контент для переменных apicontent в сообщениях.
custom_data	JSON object	`{ "field": "value"}`	Нет	Пользовательские параметры, возвращаются в хуках (Захват событий)

data	array	[{"email": "profile1@example.com"}, {"email": "profile2@example.com"}]	Да	Поле в базе данных и его значение, по которому будет произведен поиск нужного профиля. Подробнее о `data` здесь.
Настройки поиска профиля				
matching	string	"email" "email_profile" "email_sub" "phone" "phone_sub" "profile_id" "push_sub" "custom" "custom_sub" "email_phone" "email_phone_sub"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию - email. Подробнее о матчинге здесь.

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
result	array	Идентификаторы профилей, которым отправлена рассылка

Пример запроса с матчингом по email

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "trigger_id": 8,
    "format": "json",
    "matching": "email",
    "data": [
        {
            "email": "example1@example.com"
        },
        {
            "email": "example2@example.com"
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <trigger_id>85</trigger_id>
    <format>xml</format>
    <matching>email</matching>
    <data>
        <email>example3@example.com</email>
    </data>
    <data>
        <email>example4@example.com</email>
    </data>
</xml>
```

Пример запроса с матчингом по custom-полю

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "trigger_id": 96,
    "format": "json",
    "skip_triggers": false,
    "matching": "custom",
    "field_name": "Custom_ID",
    "data": [
        {
            "field_value": "003"
        },
        {
            "field_value": "002"
        }
    ]
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <trigger_id>85</trigger_id>
  <format>xml</format>
  <skip_triggers>false</skip_triggers>
  <matching>custom</matching>
  <field_name>Custom_ID</field_name>
  <data>
    <field_value>YY</field_value>
  </data>
    <data>
    <field_value>AA</field_value>
  </data>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "result": [
        {
            "profile_id": "6606e08f10bcdd88297057a7"
        },
        {
            "profile_id": "66101437dc4594c2934da6da"
        }
    ]
}
```


XML


```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <result>
        <profile_id>66101437dc4594c2934da6da</profile_id>
    </result>
    <result>
        <profile_id>6606e08f10bcdd88297057a7</profile_id>
    </result>
</xml>
```


JSON


```
{
        "error":400,
        "error_debug":"http: request body too large",
        "error_text":"Invalid request"
}
```


XML


```
<xml>
   <error>400</error>
   <error_debug>http: request body too large</error_debug>
   <error_text>Invalid request</error_text>
</xml>
```

Массовый импорт профилей + отправка триггера

Описание

Добавление или обновление нескольких профилей с последующей отправкой триггерной рассылки.

Данный запрос является batch-версией запроса Импорт профиля + Отправка триггера.

NOTE

Тело запроса ограничено по умолчанию до 10мб, исключение: файлы — 100мб.

URL-адрес

Метод: POST

`http://example.com/api/v1.1/campaigns/triggers/import_and_start_batch`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
Импорт профиля клиента				
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен

data	JSON object	[
		{
		"data": {
		"_fname": "John",
		"_lname": "Lambert",
		"email": "profile4@example.com",
		"phones": "+79000000000"
		}
		},
		{
		"data": {
		"_fname": "Tom",
		"_lname": "Hard",
		"email": "profile3@example.com",
		"phones": "+79000000001"
		}
		}
		]
			Нет	Данные о профиле, в том числе подписки.
				
				Каждый профиль — это отдельный JSON-объект "data".

skip_triggers	bool	true	Нет	Пропустить запуск триггеров. По умолчанию – false.
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки. По умолчанию – false.
detect_geo	bool	true	Нет	Включает автоопределение geo данных по полю _regip или _ip в data

subscription_filter	JSON object	Пример для push:
		`"subscription_filter": { "email": "example@example.com", "phone": "+79106135133", "subscription_id": "CAREFULWITHTHATAXE", "provider": "Chrome", "not_strict": true }`
		
		Пример для email:
		`"subscription_filter": { "email": "example@example.com", "not_strict": true }`
		
		Пример для sms:
		`"subscription_filter": { "phone": "+79106135133", "not_strict": false }`
			Нет	Выбирает конкретную подписку профиля для отправки.
				Если передан поиск профиля по <code>custom</code>, <code>profile_id</code> или <code>custom_request</code>.
				
				<code>not_strict</code> определяет, что делать, если подписка не найдена. В случае "true" будет использована любая подходящая подписка.
				</td>

Поиск профиля подписчика для обновления				
matching	string	"email" "email_profile" "phone" "profile_id" "custom"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. по умолчанию - email. Подробнее.

email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. **TIP** Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Да	Идентификатор ресурса для поиска по email и phone
Запуск триггера				
trigger_id	int	13	Да	Идентификатор триггерной рассылки для запуска

timeout	JSON object	{
		"days": 0,
		"hours": 0,
		"minutes": 5
		}
			Нет	Задержка перед запуском триггера. Указываются дни, часы и минуты.

custom_data	JSON object	{
		"sending_time": "12:31:12"
		}
			Нет	Данные для Захвата событий (Action hooks)

content	JSON object	{
		"page": "registration_form"
		}
			Нет	Дополнительные данные для подстановки в шаблон сообщения переменной `{apicontent.fieldname}`

attach	JSON objects array	{
		"data": "data:text/csv; base64,SEVMTE8K",
		"name": "Attach.txt"
		}
			Нет	Файл для прикрепления к email сообщению (только для email триггеров)

Возвращаемые параметры

Параметр	Тип	Описание
error_text	string	Текст ошибки
error	int	Код ошибки
result	array	Идентификаторы (profile_id) успешно импортированных профилей

Пример запроса с матчингом по email

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "trigger_id": 96,
    "format": "json",
    "skip_triggers": false,
    "detect_geo": true,
    "matching": "email",
    "data": [
        {
            "data": {
                "_fname": "Lili",
                "_lname": "Collins",
                "email": "example@example.com",
                "phones": "+79000000000",
                "subscriptions": [
                    {
                        "channel": "email",
                        "email": "example@example.com",
                        "priority": 0,
                        "resource_id": 21,
                        "status": 0
                    }
                ]
            }
        },
        {
            "data": {
                "_fname": "Eva",
                "_lname": "Collins",
                "email": "example@example.com",
                "phones": "+79000000001",
                "subscriptions": [
                    {
                        "channel": "email",
                        "email": "example@example.com",
                        "priority": 0,
                        "resource_id": 21,
                        "status": 0
                    }
                ]
            }
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <trigger_id>96</trigger_id>
    <format>xml</format>
    <skip_triggers>false</skip_triggers>
    <detect_geo>true</detect_geo>
    <matching>email</matching>
<data>
    <_fname>Lili</_fname>
    <_lname>Collins</_lname>
    <email>example@example.com</email>
    <phones>+79000000000</phones>
        <subscriptions>
            <channel>email</channel>
            <email>example@example.com</email>
            <priority>0</priority>
            <resource_id>21</resource_id>
            <status>0</status>
        </subscriptions>
</data>
<data>
    <_fname>Eva</_fname>
    <_lname>Collins</_lname>
    <email>example@example.com</email>
    <phones>+79000000001</phones>
    <subscriptions>
        <channel>email</channel>
        <email>example@example.com</email>
        <priority>0</priority>
        <resource_id>21</resource_id>
        <status>0</status>
    </subscriptions>
</data>
</xml>
```

Пример запроса с матчингом по custom-полю

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "trigger_id": 85,
    "format": "json",
    "skip_triggers": false,
    "detect_geo": true,
    "matching": "custom",
    "field_name": "Custom_ID",
    "data": [
        {
            "data": {
                "_fname": "Adama",
                "_lname": "Fuller",
                "email": "example@example.com,
                "phones": "+79000000000",
                "Custom_ID": "345",
                "subscriptions": [
                    {
                        "channel": "email",
                        "email": "example@example.com",
                        "priority": 0,
                        "resource_id": 21,
                        "status": 0
                    }
                ]
            }
        },
        {
            "data": {
                "_fname": "Eva",
                "_lname": "Fuller",
                "email": "example@example.com",
                "phones": "+79000000001",
                "Custom_ID": "567",
                "subscriptions": [
                    {
                        "channel": "email",
                        "email": "example@example.com",
                        "priority": 0,
                        "resource_id": 21,
                        "status": 0
                    }
                ]
            }
        }
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <trigger_id>85</trigger_id>
    <format>xml</format>
    <skip_triggers>false</skip_triggers>
    <detect_geo>true</detect_geo>
    <matching>custom</matching>
    <field_name>custom_field</field_name>
    <data>
        <_fname>Irina</_fname>
        <_lname>Depp</_lname>
        <email>example@example.com</email>
        <phones>+79000000000</phones>
        <custom_field>343</custom_field>
        <subscriptions array='true'>
            <channel>email</channel>
            <email>example@example.com</email>
            <priority>0</priority>
            <resource_id>21</resource_id>
            <status>0</status>
        </subscriptions>
    </data>
    <data>
        <_fname>Will</_fname>
        <_lname>Depp</_lname>
        <email>example@example.com</email>
        <phones>+79000000001</phones>
        <custom_field>560</custom_field>
        <subscriptions array='true'>
            <channel>email</channel>
            <email>example@example.com</email>
            <priority>0</priority>
            <resource_id>21</resource_id>
            <status>0</status>
        </subscriptions>
    </data>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "result": [
        {
            "profile_id": "661027b04a5d12bd6cc09125"
        },
        {
            "profile_id": "661027b04a5d12bd6cc09129"
        }
    ]
}
```


XML


```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <result>
        <profile_id>661027b04a5d12bd6cc09125</profile_id>
    </result>
    <result>
        <profile_id>661027b04a5d12bd6cc09129</profile_id>
    </result>
</xml>
```


Пример ответа при срабатывании ограничения запроса

JSON


```
{
    "error": 400,
    "error_debug": "http: request body too large",
    "error_text": "Invalid request"
}
```


XML

```
<xml>
   <error>400</error>
   <error_debug>http: request body too large</error_debug>
   <error_text>Invalid request</error_text>
</xml>
```

Клонировать триггер рассылку

Описание

Клонировать триггерную рассылку.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/campaigns/triggers/clone`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
id	int	1	Да	Идентификатор рассылки
name	string	“Clone trigger campaign”	Нет	Название клона рассылки по умолчанию – “Clone of …”
is_active	bool	false	Нет	Статус рассылки

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 96,
    "name": "Clone trigger campaign #1",
    "is_active": false
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <name>Clone trigger campaign #2</name>
    <id>95</id>
    <is_active>true</is_active>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 97
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <saved_id>95</saved_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор клона рассылки

Data array

Описание

Массив `data` используется в следующих запросах:

- задание на массовую отправку триггера
- задание на массовый импорт профиля + отправка триггера
- массовая отправка триггера
- массовый импорт профилей + отправка триггера

Массив состоит из объектов, каждый из которых содержит информацию об отдельном профиле. В качестве ключа и значения объекта необходимо указать данные, по которым будет происходить поиск профиля в базе.

Параметры

Параметр	Тип	Пример	Обязательный	Описание	
email	string	`"john@example.com"`	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю "matching":"email_sub" - только по подпискам	Email-адрес	
phone	string	"+79000000000"	"matching":"phone" - по телефону из профиля и подписок "matching":"phone_sub" - по телефону из подписок	Номер телефона	
profile_id	string	`"abcdefghijklmnqrstuvwxyz"`	"matching":"profile_id"	Идентификатор профиля	
field_value	int/string	"12345"	"matching":"custom"	Значение поля для поиска	
custom_request	JSON object	`{ "email": { "$in": [ "asd@dsfsd.com", "zxcv@sdfsd.com" ] } }`	Если нет свойства "matching"	MongoDB-запрос по полям профиля	

Примеры

email

JSON

```
 "data": [
                {"email": "profile1@example.com"},
                {"email": "profile2@example.com"}
        ]
```

XML

```
<data>
    <email>profile1@example.com</email>
</data>
<data>
    <email>profile2@example.com</email>
</data>
```

phone

JSON

```
 "data": [
                {"phone": "+79000000000"},
                {"phone": "+79000000001"}
        ]
```

XML

```
<data>
    <phone>+79000000000</phone>
</data>
<data>
    <phone>+79000000001</phone>
</data>
```

profile_id

JSON

```
 "data": [
                {"profile_id": "60ae529b3f62fcdebdb159bb"},
                {"profile_id": "843vfdkisfh934hfe834rhkk"}
        ]
```

XML

```
<data>
    <profile_id>+79000000000</profile_id>
</data>
<data>
    <profile_id>+79000000001</profile_id>
</data>
```

custom

JSON

```
"data": [
        {"field_value": "001"},
        {"field_value": "002"}
        ]
```

XML

```
<data>
    <field_value>001</field_value>
</data>
<data>
    <field_value>002</field_value>
</data>
```

Получить информацию о кампании

Описание

Получить детальную информацию о кампании, включая настройки аудитории, ЛКГ, маркетинговые активности и атрибуты.

> **CAUTION: Права доступа**
>
> Для использования этого метода API-токен должен иметь права:
>
> - **Кампании — Просмотр объекта**
> - Доступ к группам, в которых находятся кампании
>
> Настройте права в разделе **Настройки — Токены**.

URL-адрес

Метод: `POST`

```
https://example.com/api/v1.1/scampaigns/get
```

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	22	Да	Идентификатор кампании
format	string	"json"	Нет	Формат возвращаемых данных по умолчанию – "json"
marketing_activities	bool	false	Нет	Включить список маркетинговых активностей по умолчанию – false

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 22,
  "marketing_activities": true
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>22</id>
  <marketing_activities>true</marketing_activities>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "id": 22,
      "is_delete": false,
      "createtime": "2025-08-22T09:00:21.654Z",
      "updatetime": "2025-08-22T11:08:10.216Z",
      "updated_user": "api.user",
      "name": "Сезонная распродажа",
      "description": "Кампания по продвижению весенней коллекции",
      "tags": [],
      "state": "active",
      "audience": {
        "state": "ready",
        "seg_id": 4,
        "list_id": 2,
        "extension_conf": {
          "workflows": {"ids": []},
          "triggers": {"ids": []},
          "broadcasts": {"ids": [1]},
          "regulars": {"ids": []},
          "abns": {"ids": []},
          "placements": {"ids": []}
        },
        "lcg": {
          "stratification_field_settings": [
            {
              "field_name": "_sex",
              "field_type": "enum",
              "condition_type": "list",
              "conditions": [
                {"list": [1]},
                {"list": [2]}
              ]
            },
            {
              "field_name": "_region",
              "field_type": "unicode",
              "condition_type": "list",
              "conditions": [
                {"list": ["Москва", "Санкт-Петербург"]},
                {"list": ["Казань"]}
              ]
            }
          ],
          "ignore_entities": {
            "workflows": {"ids": []},
            "triggers": {"ids": []},
            "broadcasts": {"ids": [1]}
```

```
      "broadcasts": {"ids": [1]},
      "regulars": {"ids": []},
      "abns": {"ids": []},
      "placements": {"ids": []}
    },
    "size_formation_settings": {
      "limit": 133333,
      "percent": "16.00"
    }
  }
},
"active_period": {
  "settings": {
    "tz": "UTC",
    "end_at": "2025-08-24T12:03:21.93Z"
  },
  "history": {
    "start": "2025-08-22T09:00:47.786Z"
  }
},
"counters": {
  "segment": 631,
  "test_group": 530,
  "local_control_group": 101,
  "global_control_group": 0
},
"marketing_activities_counter": 2,
"groups": [
  {
    "id": 0,
    "name": "main"
  }
],
"attributes": {
  "season": "spring_2025",
  "branch": "marketing"
},
"marketing_activities": [
  {
    "workflows": [
      {
        "id": 9,
        "name": "Welcome Series",
        "is_active": true
      }
    ],
    "triggers": [
      {
        "id": 6,
        "name": "Thank You Purchase",
```

```
              "is_active": true
            }
          ],
          "broadcasts": [
            {
              "id": 1,
              "name": "Seasonal Sale Announcement",
              "is_active": true,
              "is_completed": false
            }
          ],
          "regulars": [
            {
              "id": 2,
              "name": "Weekly Newsletter",
              "is_active": true
            }
          ],
          "abns": [],
          "placements": []
        }
      ]
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```


XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <id>22</id>
    <is_delete>false</is_delete>
    <createtime>2025-08-22T09:00:21.654Z</createtime>
    <updatetime>2025-08-22T11:08:10.216Z</updatetime>
    <updated_user>api.user</updated_user>
    <name>Сезонная распродажа</name>
    <description>Кампания по продвижению весенней коллекции</description>
    <state>active</state>
    <audience>
      <state>ready</state>
      <seg_id>4</seg_id>
      <list_id>2</list_id>
      <extension_conf>
        <workflows>
        </workflows>
        <triggers>
        </triggers>
        <broadcasts>
          <ids>
            <id>1</id>
          </ids>
        </broadcasts>
        <regulars>
        </regulars>
        <abns>
        </abns>
        <placements>
        </placements>
      </extension_conf>
      <lcg>
        <stratification_field_settings>
          <field>
            <field_name>_sex</field_name>
            <field_type>enum</field_type>
            <condition_type>list</condition_type>
            <conditions>
              <condition>
                <list>
                  <value>1</value>
                </list>
              </condition>
              <condition>
                <list>
                  <value>2</value>
                </list>
```

```
              </list>
            </condition>
          </conditions>
        </field>
      </stratification_field_settings>
      <ignore_entities>
        <broadcasts>
          <ids>
            <id>1</id>
          </ids>
        </broadcasts>
      </ignore_entities>
      <size_formation_settings>
        <limit>133333</limit>
        <percent>16.00</percent>
      </size_formation_settings>
    </lcg>
  </audience>
  <active_period>
    <settings>
      <tz>UTC</tz>
      <end_at>2025-08-24T12:03:21.93Z</end_at>
    </settings>
    <history>
      <start>2025-08-22T09:00:47.786Z</start>
    </history>
  </active_period>
  <counters>
    <segment>631</segment>
    <test_group>530</test_group>
    <local_control_group>101</local_control_group>
    <global_control_group>0</global_control_group>
  </counters>
  <marketing_activities_counter>2</marketing_activities_counter>
  <groups>
    <group>
      <id>0</id>
      <name>main</name>
    </group>
  </groups>
  <attributes>
    <season>spring_2025</season>
    <branch>marketing</branch>
  </attributes>
 </data>
</xml>
```


Возвращаемые параметры

Основные поля

Параметр	Тип	Описание
`error`	int	Код ошибки. 0 - успешное выполнение

Параметр	Тип	Описание
`error_text`	string	Текст ошибки
`data`	array	Массив объектов кампаний

Объект data

Содержит основную информацию о кампании.

Параметр	Тип	Описание
`id`	int	Идентификатор кампании
`name`	string	Название кампании
`description`	string	Описание кампании
`list_id`	int	Идентификатор базы профилей
`seg_id`	int	Идентификатор сегмента
`state`	string	Текущий статус кампании
`createtime`	string	Дата и время создания кампании
`updatetime`	string	Дата и время последнего редактирования
`updated_user`	string	Имя пользователя, внесшего последние изменения
`is_delete`	boolean	Статус удаления кампании
`tags`	array	Теги, присвоенные кампаниии
`groups`	array	Группы доступа
`marketing_activities_counter`	int	Количество маркетинговых активностей (рассылок, сценариев, размещений)
`attributes`	object	Атрибуты кампании
`counters`	object	Количество профилей в аудитории, в Тестовой группе, Локальной и Глобальной контрольных группах

Объект counters

Содержит информацию о количестве профилей в различных группах кампании.

Параметр	Тип	Описание
`segment`	int	Количество профилей в сегменте
`test_group`	int	Количество профилей в тестовой группе (ТГ)
`local_control_group`	int	Количество профилей в локальной контрольной группе (ЛКГ)
`global_control_group`	int	Количество профилей в глобальной контрольной группе(ГКГ) базы профилей

Объект active_period

Содержит информацию о временных рамках кампании.

Параметр	Тип	Описание
`settings`	object	**Плановые** настройки периода активности
`history`	object	**Фактические** данные периода активности

settings

Параметр	Тип	Описание
`tz`	string	Часовой пояс кампании
`start_at`	string	**Запланированная** дата активации кампании
`end_at`	string	**Запланированная** дата завершения кампании

history

Параметр	Тип	Описание
`start`	string	**Фактическая** дата активации кампании

Объект audience

Содержит настройки и информацию об аудитории кампании.

Параметр	Тип	Описание
`state`	string	Статус разметки аудитории
`seg_id`	int	Идентификатор сегмента кампании
`list_id`	int	Идентификатор базы
`extension_conf`	object	Активности расширения аудитории
`lcg`	object	Настройки локальной контрольной группы

extension_conf

Активности, включенные в расширение аудитории.

Параметр	Тип	Описание
`workflows`	object	Сценарии, расширяющие аудиторию
`triggers`	object	Триггерные рассылки, расширяющие аудиторию
`broadcasts`	object	Бродкаст рассылки, расширяющие аудиторию
`regulars`	object	Регулярные рассылки, расширяющие аудиторию
`abns`	object	A/B-тесты, расширяющие аудиторию

Параметр	Тип	Описание
`placements`	object	Размещения, расширяющие аудиторию

Вложенные объекты extension_conf

Каждый объект содержит:

Параметр	Тип	Описание
`ids`	array	Массив идентификаторов активностей

lcg (Локальная контрольная группа)

Параметр	Тип	Описание
`stratification_field_settings`	array	Настройки стратификации
`ignore_entities`	object	Активности, игнорирующие разметку ЛКГ
`size_formation_settings`	object	Настройки размера ЛКГ

stratification_field_settings

Параметр	Тип	Описание
`field_name`	string	Название поля для стратификации
`field_type`	string	Тип поля
`condition_type`	string	Тип условия для стратификации
`conditions`	array	Условия стратификации

ignore_entities

Активности, игнорирующие из разметку ЛКГ

Параметр	Тип	Описание
`workflows`	object	Сценарии, игнорирующие разметку ЛКГ
`triggers`	object	Триггеры, игнорирующие разметку ЛКГ
`broadcasts`	object	Бродкаст рассылки, игнорирующие разметку ЛКГ
`regulars`	object	Регулярные рассылки, игнорирующие разметку ЛКГ
`abns`	object	A/B-тесты, игнорирующие разметку ЛКГ
`placements`	object	Размещения, игнорирующие разметку ЛКГ

size_formation_settings

Настройки размера ЛКГ

Параметр	Тип	Описание
`limit`	int	Лимит размера ЛКГ
`percent`	string	Процент от аудитории для ЛКГ

Объект marketing_activities

Содержит список всех маркетинговых активностей кампании.

Параметр	Тип	Описание
`workflows`	array	Сценарии
`triggers`	array	Триггеры
`broadcasts`	array	Бродкасты
`regulars`	array	Регулярные рассылки
`abns`	array	A/B-тесты
`placements`	array	Размещения

Общие поля маркетинговых активностей

Параметр	Тип	Описание
`id`	int	Идентификатор активности
`name`	string	Название маркетингово активности
`is_active`	boolean	Статус активности
`is_completed`	boolean	Статус завершения

Получить список кампаний

Описание

Получить список кампаний - объединений рассылок и сценариев в единые маркетинговые стратегии.

CAUTION: Права доступа

Для использования этого метода API-токен должен иметь права:

- **Кампании — Запуск**
- Доступ к группам, в которых находятся кампании

Настройте права в разделе **Настройки — Токены**.

URL-адрес

Метод: `POST`

```
https://example.com/api/v1.1/scampaigns/list
```

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных по умолчанию – "json"
from_id	int	1	Нет	Идентификатор первой кампании в ответе по умолчанию – 1
limit	int	2	Нет	Лимит на количество возвращаемых данных по умолчанию – 1000
state	string	"active"	Нет	Статус кампании. Один из: "cloning", "inactive", "waiting", "active", "completed", "deleted"
ui_tags	[]String	["tag1", "tag2"]	Нет	Список тегов для поиска по умолчанию – []
groups	[]String	["group1", "group2"]	Нет	Список групп для поиска по умолчанию – []
groups_id	[]int	[0, 1]	Нет	Список ID групп для поиска по умолчанию – []

attributes	JSON object	`"attributes": \{ "season": ["spring_2025", "summer_2025"] \}`	Нет	Значения атрибутов для поиска

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "limit": 1
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <limit>1</limit>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "id": 42,
      "is_delete": false,
      "createtime": "2025-08-22T07:15:29.711Z",
      "updatetime": "2025-08-22T09:13:24.264Z",
      "updated_user": "test",
      "name": "Сезонная распродажа",
      "description": "",
      "tags": [],
      "state": "inactive",
      "audience": {
        "state": "ready",
        "seg_id": 124,
        "list_id": 49,
        "extension_conf": {
          "workflows": {"ids": []},
          "triggers": {"ids": []},
          "broadcasts": {"ids": []},
          "regulars": {"ids": [556]},
          "abns": {"ids": []},
          "placements": {"ids": []}
        },
        "lcg": {
          "ignore_entities": {
            "workflows": {"ids": []},
            "triggers": {"ids": []},
            "broadcasts": {"ids": []},
            "regulars": {"ids": []},
            "abns": {"ids": []},
            "placements": {"ids": []}
          },
          "size_formation_settings": {
            "limit": 0,
            "percent": "10.00"
          }
        }
      },
      "active_period": {
        "settings": {
          "tz": "Europe/Moscow",
          "end_at": "2025-08-23T10:36:25.099+03:00"
        },
        "history": {
          "start": "2025-08-22T12:13:18.813+03:00",
          "stop": "2025-08-22T12:13:24.264+03:00"
        }
      }
```

```
      },
      "control_group_counter": 3002,
      "marketing_activities_counter": 1,
      "refresh_data": {
        "state": "completed",
        "progress": 100,
        "task_start_date": "2025-08-22T09:13:19.926Z",
        "task_end_date": "2025-08-22T09:13:20.048Z"
      },
      "group_id": 0,
      "groups_id": [0],
      "groups": [
        {
          "id": 0,
          "name": "main"
        }
      ],
      "attributes": {
        "season": "spring_2025",
        "branch": "marketing"
      }
    }
  ],
  "error": 0,
  "error_text": "",
  "total_count": 5
}
```


XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <id>42</id>
    <is_delete>false</is_delete>
    <createtime>2025-08-22T07:15:29.711Z</createtime>
    <updatetime>2025-08-22T09:13:24.264Z</updatetime>
    <updated_user>test</updated_user>
    <name>Сезонная распродажа</name>
    <state>inactive</state>
    <audience>
      <state>ready</state>
      <seg_id>124</seg_id>
      <list_id>49</list_id>
      <extension_conf>
        <workflows><ids/></workflows>
        <triggers><ids/></triggers>
        <broadcasts><ids/></broadcasts>
        <regulars><ids><id>556</id></ids></regulars>
        <abns><ids/></abns>
        <placements><ids/></placements>
      </extension_conf>
      <lcg>
        <ignore_entities>
          <workflows><ids/></workflows>
          <triggers><ids/></triggers>
          <broadcasts><ids/></broadcasts>
          <regulars><ids/></regulars>
          <abns><ids/></abns>
          <placements><ids/></placements>
        </ignore_entities>
        <size_formation_settings>
          <limit>0</limit>
          <percent>10.00</percent>
        </size_formation_settings>
      </lcg>
    </audience>
    <active_period>
      <settings>
        <tz>Europe/Moscow</tz>
        <end_at>2025-08-23T10:36:25.099+03:00</end_at>
      </settings>
      <history>
        <start>2025-08-22T12:13:18.813+03:00</start>
        <stop>2025-08-22T12:13:24.264+03:00</stop>
      </history>
    </active_period>
    <control_group_counter>3002</control_group_counter>
```

```
    <control_group_counter>3002</control_group_counter>
    <marketing_activities_counter>1</marketing_activities_counter>
    <refresh_data>
      <state>completed</state>
      <progress>100</progress>
      <task_start_date>2025-08-22T09:13:19.926Z</task_start_date>
      <task_end_date>2025-08-22T09:13:20.048Z</task_end_date>
    </refresh_data>
    <group_id>0</group_id>
    <groups_id>0</groups_id>
    <groups>
      <group>
        <id>0</id>
        <name>main</name>
      </group>
    </groups>
    <attributes>
      <season>spring_2025</season>
      <branch>marketing</branch>
    </attributes>
  </data>
  <total_count>5</total_count>
</xml>
```


Возвращаемые параметры

Основные поля

Параметр	Тип	Описание
`error`	int	Код ошибки. 0 - успешное выполнение
`error_text`	string	Текст ошибки
`total_count`	int	Общее количество кампаний
`next_from_id`	int	Указатель для получения следующей порции данных
`data`	array	Массив объектов кампаний

Объект data

Содержит основную информацию о кампании.

Параметр	Тип	Описание
`id`	int	Идентификатор кампании
`name`	string	Название кампании
`description`	string	Описание кампании
`list_id`	int	Идентификатор базы профилей

Параметр	Тип	Описание
`seg_id`	int	Идентификатор сегмента
`state`	string	Текущий статус кампании
`createtime`	string	Дата и время создания кампании
`updatetime`	string	Дата и время последнего обновления
`updated_user`	string	Имя пользователя, внесшего последние изменения
`is_delete`	boolean	Показывает, удалена ли кампания
`tags`	array	Теги, присвоенные кампании
`groups`	array	Группы пользователей, которые могут работать с кампанией
`group_id`	int	Идентификатор основной группы кампании
`groups_id`	array	Массив идентификаторов групп кампании
`marketing_activities_counter`	int	Количество маркетинговых активностей (рассылок, сценариев, размещений)
`control_group_counter`	int	Количество профилей в контрольной группе
`attributes`	object	Атрибуты кампании
`refresh_data`	object	Информация о процессе обновления данных кампании

Объект active_period

Содержит информацию о временных рамках кампании.

Параметр	Тип	Описание
`settings`	object	**Плановые** настройки периода активности
`history`	object	**Фактические** данные периода активности

settings - Запланированные временные параметры кампании

Параметр	Тип	Описание
`tz`	string	Часовой пояс кампании
`start_at`	string	Запланированная дата начала кампании
`end_at`	string	Запланированная дата окончания кампании

history - Фактические данные о выполнении кампании

Параметр	Тип	Описание
`start`	string	Фактическая дата начала кампании

Параметр	Тип	Описание
`stop`	string	Фактическая дата остановки кампании

Объект audience

Содержит настройки и информацию об аудитории кампании.

Параметр	Тип	Описание
`state`	string	Статус разметки аудитории
`seg_id`	int	Идентификатор сегмента
`list_id`	int	Идентификатор базы профилей
`extension_conf`	object	Настройки расширения аудитории через маркетинговые активности
`lcg`	object	Настройки локальной контрольной группы

extension_conf

Активности, которые расширяют аудиторию кампании

Параметр	Тип	Описание
`workflows`	object	Сценарии, расширяющие аудиторию
`triggers`	object	Триггерные рассылки, расширяющие аудиторию
`broadcasts`	object	Бродкаст рассылки, расширяющие аудиторию
`regulars`	object	Регулярные рассылки, расширяющие аудиторию
`abns`	object	A/B-тесты, расширяющие аудиторию
`placements`	object	Размещения, расширяющие аудиторию

Каждый из этих объектов содержит массив идентификаторов соответствующих активностей.

lcg - Настройки локальной контрольной группы

Параметр	Тип	Описание
`ignore_entities`	object	Активности, которые не учитывают разметку ЛКГ
`size_formation_settings`	object	Настройки размера локальной контрольной группы

ignore_entities - Активности, исключенные из разметки ЛКГ

Параметр	Тип	Описание
`workflows`	object	Сценарии, игнорирующие разметку ЛКГ
`triggers`	object	Триггеры, игнорирующие разметку ЛКГ
`broadcasts`	object	Бродкаст рассылки, игнорирующие разметку ЛКГ

Параметр	Тип	Описание
`regulars`	object	Регулярные рассылки, игнорирующие разметку ЛКГ
`abns`	object	A/B-тесты, игнорирующие разметку ЛКГ
`placements`	object	Размещения, игнорирующие разметку ЛКГ

size_formation_settings - Настройки размера ЛКГ

Параметр	Тип	Описание
`limit`	int	Максимальное количество профилей в ЛКГ
`percent`	string	Процент от общей аудитории для ЛКГ

Объект refresh_data

Содержит информацию о процессе обновления данных кампании.

Параметр	Тип	Описание
`state`	string	Статус задачи обновления данных
`progress`	int	Прогресс выполнения задачи в процентах
`task_start_date`	string	Дата и время начала задачи обновления
`task_end_date`	string	Дата и время завершения задачи обновления

Объект groups

Содержит информацию о группах доступа к кампании.

Параметр	Тип	Описание
`id`	int	Идентификатор группы пользователей
`name`	string	Название группы пользователей

Активация кампании

Описание

Запрос на активацию кампании

> **CAUTION: Права доступа**
>
> Для использования этого метода API-токен должен иметь права:
>
> - **Кампании - Запуск**
> - Доступ к группам, в которых находятся кампании
>
> Настройте права в разделе **Настройки** → **Токены**.

URL-адрес

Метод: `POST`

```
https://example.com/api/v1.1/scampaigns/activate
```

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	1	Да	Идентификатор кампании
activate_date	string	"2025-03-15T10:00:00Z"	Да	Дата активации в формате RFC 3339
deactivate_date	string	"2025-03-20T15:30:00Z"	Нет	Дата деактивации в формате RFC 3339
completed_date	string	"2025-04-01T18:45:00Z"	Нет	Дата завершения в формате RFC 3339

> **TIP**
>
> Если передать `start_date` без значения даты, кампания активируется немедленно.

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 1,
  "start_date": "2025-04-01T18:45:00Z",
  "deactivate_date": "2025-05-01T18:45:00Z",
  "completed_date": "2025-05-01T18:45:00Z"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
  <start_date>2025-04-01T18:45:00Z</start_date>
  <deactivate_date>2025-05-01T18:45:00Z</deactivate_date>
  <completed_date>2025-05-01T18:45:00Z</completed_date>
</xml>
```

Пример ответа

JSON

```
{
  "data": null,
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Параметр	Тип	Описание
data	null	Данные не возвращаются

Завершение кампании

Описание

Завершить кампанию прямо сейчас. Завершение прерывает отправку сообщений работающих рассылок и переводит все вложенные рассылки и сценарии в статус “неактивна”.

CAUTION: Права доступа

Для использования этого метода API-токен должен иметь права:

- **Кампании - Запуск**
- Доступ к группам, в которых находятся кампании

Настройте права в разделе **Настройки** → **Токены**.

URL-адрес

Метод: `POST`

```
https://example.com/api/v1.1/scampaigns/completed
```

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	18	Да	Идентификатор кампании

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 18
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>18</id>
</xml>
```

Пример ответа

JSON

```
{
  "data": null,
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
data	null	Данные не возвращаются

WARNING

Завершение кампании - необратимая операция. После завершения возобновить кампанию невозможно, только создать новую.

Деактивация кампании

Описание

Деактивировать кампанию прямо сейчас. Деактивация прерывает отправку сообщений работающих рассылок.

CAUTION: Права доступа

Для использования этого метода API-токен должен иметь права:

- **Кампании - Запуск**
- Доступ к группам, в которых находятся кампании

Настройте права в разделе **Настройки** → **Токены**.

URL-адрес

Метод: `POST`

```
https://example.com/api/v1.1/scampaigns/deactivate
```

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	18	Да	Идентификатор кампании

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 18
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>18</id>
</xml>
```

Пример ответа

JSON

```
{
  "data": null,
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
data	null	Данные не возвращаются

INFO

Деактивация временно останавливает кампанию. Все настройки и данные сохраняются для возможного последующего возобновления.

Получить статус кампании

Описание

Получить текущий статус кампании

> **CAUTION: Права доступа**
>
> Для использования этого метода API-токен должен иметь права:
>
> - **Кампании - Просмотр объекта**
> - Доступ к группам, в которых находятся кампании
>
> Настройте права в разделе **Настройки** → **Токены**.

URL-адрес

Метод: `POST`

```
https://example.com/api/v1.1/scampaigns/get_status
```

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	10	Да	Идентификатор кампании

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 10
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>10</id>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "status": "active"
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <status>active</status>
  </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
data	object	Объект с информацией о статусе

Параметры статуса

Параметр	Тип	Описание
status	string	Текущий статус кампании

Возможные статусы

Статус	Описание
cloning	В процессе клонирования
inactive	Неактивна
waiting	Ожидает активации
active	Активна
completed	Завершена
deleted	Удалена

Отправить профиль клиента в сценарий

Описание

Отправить профиль клиента в сценарий.

> **TIP**
>
> При запуске сценария через API профили будут попадать в стартовую ноду “По умолчанию”.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/workflows/start`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных (по умолчанию – "json")
id	int	12	Да	Идентификатор сценария
content	JSON object	`{ "field": "value" }`	Нет	Дополнительные данные для подстановки в шаблон сообщения переменной {apicontent.fieldname}. Также {apicontent} можно использовать в узлах "API вызов" и "Контент фильтр"
attach	JSON array	`[]`	Нет	Приложить файлы к сообщению
Поиск профиля подписчиков				

matching	string	"email" "email_profile" "phone" "profile_id" "custom" </td> <td>Нет, если поиск по email из профиля или подписок</td> <td> Режим поиска подписчика. По умолчанию – email. </td>		
email	string	"john@example.com"	"matching":"email" — по профилю и подпискам "matching":"email_profile" – только по профилю "matching":"email_sub" – только по подпискам </td> <td>Email-адрес</td>	
phone	string	"+79000000000"	"matching":"phone" — по телефону из профиля и подписок "matching":"phone_sub" – по телефону из подписок </td> <td>Номер телефона</td>	
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название пользовательского поля профиля для поиска.

field_value	int/string	"12345"	"matching":"custom"	Значение пользовательского поля для поиска. **TIP** Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

subscription_filter	JSON object	Пример для push:
		`"subscription_filter": { "email": "example@example.com", "phone": "+79106135133", "subscription_id": "CAREFULWITHTHATAXE", "provider": "Chrome", "not_strict": true }`
		
		Пример для email:
		`"subscription_filter": { "email": "example@example.com", "not_strict": true }`
		
		Пример для sms:
		`"subscription_filter": { "phone": "+79106135133", "not_strict": false }`
			Нет	Выбирает конкретную подписку профиля для отправки.
				Если передан поиск профиля по <code>custom</code> или <code>profile_id</code>.
				
				<code>not_strict</code> определяет, что делать, если подписка не найдена. В случае "true" будет
				использована любая подходящая подписка.
				</td>

Пример запроса

Отправить один профиль в сценарий.

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "id": 11,
  "matching": "profile_id",
  "profile_id": "5a74876369d42613b31947b8"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>11</id>
  <matching>profile_id</matching>
  <profile_id>5a74876369d42613b31947b8</profile_id>
</xml>
```

Добавить профиль в сценарий с дополнительным контентом, который можно подставлять в любые рассылки, запланированные в сценарии. По данным, которые переданы в дополнительном контенте, возможно сформировать условия выбора. Для этого используйте в сценарии ноду **“Контент фильтр”**.

Вставить текст “New videos on your favorite channels” на место переменной `"{apicontent.email_title}"`, если она используется в сообщении.

Приложить к сообщению файлы “日本.txt” и “US.txt”.

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 12,
    "matching": "profile_id",
    "profile_id": "66102b984a5d12bd6cc0912f",
    "attach": [
        {
            "data": "data:text/csv; base64,5LuK5pel44GvCg==",
            "name": "日本.txt"
        },
        {
            "data": "data:text/csv; base64,SEVMTE8K",
            "name": "US.txt"
        }
    ],
    "content": {
        "data_lines": [
            "Channel A",
            "Channel B"
        ],
        "email_title": "New videos on your favorite channels"
    }
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>12</id>
  <matching>profile_id</matching>
  <profile_id>661027b04a5d12bd6cc09125</profile_id>
  <attach>
    <data>data:text/csv; base64,5LuK5pel44GvCg==</data>
    <name>日本.txt</name>
  </attach>
  <attach>
    <data>data:text/csv; base64,SEVMTE8K</data>
    <name>US.txt</name>
  </attach>
  <content>
    <data_lines>Channel A</data_lines>
    <data_lines>Channel B</data_lines>
    <email_title>New videos on your favorite channels</email_title>
  </content>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Одновременный импорт и запуск профиля в сценарий

Описание

Данный запрос дает возможность импортировать/обновить профиль и сразу отправить его в сценарий (цепочку).

URL-адрес

Метод: POST

`https://example.com/api/v1.1/workflows/import_and_start`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию — “json”)
workflow_id	Int	1	Да	Идентификатор цепочки
matching	string	“email”	Да	Тип матчинга
profile_id	string	“5f19953a03e4227216278ba4”	*	Импорт: Поля матчига
email	string	“importStart9219@example.com”	*	Импорт: Поля матчига
phone	string	“+79001234567”	*	Импорт: Поля матчига
field_name	string	“CRM_ID”	*	Импорт: Поля матчига
field_value	any	“12345”	*	Импорт: Поля матчига
data	object	{ “_fname”: “John”, “_lname”: “Doe” }	Нет	Импорт: Обьект с полями профиля, в том числе подписки
skip_triggers	bool	true	Нет	Опции импорта: пропускать остальные триггеры и цепочки, настроенные на действие импорта и изменения в полях и подписках.

Параметр	Тип	Пример	Обязательный	Описание
skip_invalid_subscriptions	bool	true	Нет	Опции импорта: пропускать невалидные подписки
detect_geo	bool	true	Нет	Опции импорта: определение геолокации по IP
content	object	“content”: { “one”: “My email = importStart9219@example.com”, “two”: “Timestamp: 1595512477”, “three”: “Three” }	Нет	Дополнительные данные для подстановки в шаблон сообщения переменной {apicontent.fieldname}. Также {apicontent} можно использовать в узлах “API вызов” и “Контент фильтр”
attach	array of {name, flie_key, data, mimetype}		Нет	Опции триггера: прикрепить файлы к письму

Пример запроса

Импортировать один профиль и отправить его в сценарий.

JSON

```
{
  "token": "abcdefghkijlmnopqrstuv0123456789",
  "data": {
    "subscriptions": [
      {
        "email": "example001@example.com",
        "resource_id": 7
      }
    ]
  },
  "matching": "email",
  "email": "example001@example.com",
  "content": {
    "one": "My email = example001@example.com",
    "two": "Timestamp: 1595512477",
    "three": "Three"
  },
  "workflow_id": 12
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <data>
    <subscriptions array='true'>
      <email>example001@example.com</email>
      <resource_id>7</resource_id>
    </subscriptions>
  </data>
  <matching>email</matching>
  <email>example001@example.com</email>
  <content>
    <one>My email = example001@example.com</one>
    <two>Timestamp: 1595512477</two>
    <three>Three</three>
  </content>
  <workflow_id>12</workflow_id>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "profile_id": "5eeb54a2ce9448518e85fdad"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <profile_id>6613c13a4a5d12bd6cc09156</profile_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_id	string	ID профиля

Ответы сервиса

Код	Описание
0	Операция выполнена успешно
400,450	Некорректный запрос
401	Требуется API-токен
402	Достигнуто ограничение тарифа
403	Нет прав на действие
404	Обьект не найден в зоне видимости токена
409	Существует другая запись с такими же признаками уникальности
413	База данных недоступна в выбранном ресурсе
415	Запрашиваемый Content-Type не поддерживается
435	Неоднозначный поиск, есть несколько обьектов с заданными характеристиками
441	Обьект принадлежит другой группе
500	Внутренняя ошибка сервиса

Код	Описание
501	Такой метод отсуствует

Массовый импорт и запуск профилей в сценарий

Описание

Данный запрос позволяет импортировать или обновить несколько профилей и сразу отправить их в сценарий, он является batch-версией запроса Одновременный импорт и запуск профиля в сценарий.

> **NOTE**
>
> Тело запроса ограничено по умолчанию до 10мб, исключение: файлы — 100мб.
>
> В одном запросе не рекомендуется передавать более 10 000 профилей.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/workflows/import_and_start_batch`

Параметры запроса

Параметр	Тип	Пример	Условие обязательности	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Всегда	API токен
format	string	"json"	Нет (по умолчанию "json")	Формат возвращаемых данных
workflow_id	int	1	Всегда	Идентификатор цепочки
matching	string	"email"	Всегда	Режим поиска подписчика. Доступные значения: email, email_profile, phone, profile_id, custom
profile_id	string	"5f19953a03e4227216278ba4"	Только при matching = "profile_id"	Идентификатор профиля
email	string	"123@example.com"	Только при matching = "email" или "email_profile"	Email-адрес
phone	string	"+79001234567"	Только при matching = "phone"	Номер телефона
field_name	string	"CRM_ID"	Только при matching = "custom"	Название кастомного поля профиля для поиска
field_value	any	"12345"	Только при matching = "custom"	Значение кастомного поля для поиска. Для полей типа "теги" передавать через запятую: "тег_1, тег_2"

Параметр	Тип	Пример	Условие обязательности	Описание
data	object	{"_fname": "John", "_lname": "Doe"}	Нет	Данные о профиле, включая подписки
skip_triggers	bool	true	Нет (по умолчанию false)	Пропустить запуск триггеров
skip_invalid_subscriptions	bool	true	Нет (по умолчанию false)	Пропустить невалидные подписки
detect_geo	bool	true	Нет	Автоопределение geo данных по полям _regip или _ip в data
content	object	{"one": "My email value", "two": "Timestamp: 1595512477"}	Нет	Дополнительные данные для подстановки в шаблон через {apicontent.fieldname}
attach	array	[{"name": "file.txt", "file_key": "key123"}]	Нет	Файлы для прикрепления к Email сообщению (только для Email триггеров)

Возвращаемые параметры

Параметр	Тип	Описание
error_text	string	Текст ошибки
error	int	Код ошибки
task_id	string	Идентификатор задачи

Пример запроса с матчингом по email

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "workflow_id": 12,
  "matching": "email",
  "data": [
    {
      "data": {
        "_fname": "Alisa",
        "_lname": "Lambert",
        "email": "example@example.com",
        "phones": "+79000000000"
      }
    },
    {
      "data": {
        "_fname": "Alex",
        "_lname": "Lambert",
        "email": "example@example.com",
        "phones": "+79000000000"
      }
    }
  ]
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <workflow_id>10</workflow_id>
  <matching>email</matching>
  <data>
    <_fname>John</_fname>
    <_lname>Lambert</_lname>
    <email>example@example.com</email>
  </data>
  <data>
    <_fname>Bert</_fname>
    <_lname>Lambert</_lname>
    <email>example@example.com</email>
  </data>
</xml>
```

Пример запроса с матчингом по custom-полю

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "workflow_id": 12,
  "matching": "custom",
  "field_name": "CRM_ID",
  "data": [
    {
      "data": {
        "_fname": "Alina",
        "_lname": "Lambert",
        "email": "example@example.com",
        "phones": "+79006574081",
        "CRM_ID": "00066"
      }
    },
    {
      "data": {
        "_fname": "Victor",
        "_lname": "Lambert",
        "email": "example@example.com",
        "phones": "+79000005422",
        "CRM_ID": "00076"
      }
    }
  ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <workflow_id>12</workflow_id>
    <matching>custom</matching>
    <field_name>New_field_7</field_name>
    <data>
        <_fname>Polina</_fname>
        <_lname>Lambert</_lname>
        <email>example@example.com</email>
        <phones>+79006574082</phones>
        <New_field_7 string = "true" >00067</New_field_7>
    </data>
    <data>
        <_fname>Dariya</_fname>
        <_lname>Lambert</_lname>
        <email>example@example.com</email>
        <phones>+79000005423</phones>
        <New_field_7 string = "true">00077</New_field_7>
    </data>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation",
    "result": [
        {
            "profile_id": "6613c4754a5d12bd6cc0915f"
        },
        {
            "profile_id": "6613c4754a5d12bd6cc09163"
        }
    ]
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <result>
        <profile_id>6613c4bf4a5d12bd6cc09167</profile_id>
    </result>
    <result>
        <profile_id>6613c4bf4a5d12bd6cc0916b</profile_id>
    </result>
</xml>
```

Пример ответа при срабатывании ограничения запроса

JSON

```
{
  "error": 400,
  "error_debug": "http: request body too large",
  "error_text": "Invalid request"
}
```

XML

```
<xml>
   <error>400</error>
   <error_debug>http: request body too large</error_debug>
   <error_text>Invalid request</error_text>
</xml>
```

Задание на массовый импорт и запуск профилей в сценарий

Описание

Добавление или обновление нескольких профилей с последующей отправкой их в сценарий. Возвращает ответ с id таска, не дожидаясь выполнения.

> **NOTE**
>
> Тело запроса ограничено по умолчанию до 10мб, исключение: файлы — 100мб.
>
> В одном запросе не рекомендуется передавать более 10 000 профилей.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/workflows/import_and_start_batch_task`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API токен
format	string	"json"	Нет	Формат возвращаемых данных (по умолчанию — "json")
workflow_id	int	1	Да	Идентификатор цепочки
matching	string	"email"	Да	Режим поиска подписчика. По умолчанию — email.
profile_id	string	"5f19953a03e4227216278ba4"	"matching":"profile_id"	Идентификатор профиля
email	string	"importStart9219@example.com"	"matching":"email" — по профилю и подпискам "matching":"email_profile" — только по профилю	Email-адрес

phone	string	"+79001234567"	"matching":"phone"	Номер телефона
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.
field_value	any	"12345"	"matching":"custom"	Значение кастомного поля для поиска. Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".
data	object	\{ "_fname": "John", "_lname": "Doe" }	Нет	Данные о профиле, в том числе подписки
skip_triggers	bool	true	Нет	Пропустить запуск триггеров по умолчанию – false
skip_invalid_subscriptions	bool	true	Нет	Пропустить невалидные подписки по умолчанию – false
detect_geo	bool	true	Нет	Включает автоопределение geo данных по полю _regip или _ip в data
content	object	"content": \{ "one": "My email = importStart9219@example.com", "two": "Timestamp: 1595512477", "three": "Three" }	Нет	Дополнительные данные для подстановки в шаблон сообщения переменной \{apicontent.fieldname}. Также \{apicontent} можно использовать в узлах "API вызов" и "Контент фильтр"
custom_data	JSON object	"custom_data": \{ "field": "value", }	Нет	Пользовательские параметры, возвращаются в хуках ([Захват событий](/user-guide/integrations/action-hooks/37849833/)). Можно передавать в корне запроса или отдельно в каждом объекте `data`.
attach	array of \{name, flie_key, data, mimetype}		Нет	Файл для прикрепления к Email сообщению (только для Email триггеров)

Возвращаемые параметры

Параметр	Тип	Описание
error_text	string	Текст ошибки
error	int	Код ошибки
task_id	string	Идентификатор задачи

Пример запроса с матчингом по email

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "workflow_id": 12,
  "matching": "email",
  "data": [
    {
      "data": {
        "_fname": "Herbert",
        "_lname": "Lambert",
        "email": "example@example.com",
        "phones": "+79862014852"
      }
    },
    {
      "data": {
        "_fname": "Bert",
        "_lname": "Lambert",
        "email": "example@example.com",
        "phones": "+79862014853"
      }
    }
  ]
}
```

XML

```
<xml>
<token>abcdefghijklmnqrstuvwxyz</token>
  <workflow_id>12</workflow_id>
  <matching>email</matching>
  <data>
      <_fname>Herb</_fname>
      <_lname>Limbert</_lname>
      <email>example@example.com</email>
      <phones>+79847076617</phones>
  </data>
  <data>
      <_fname>Bert</_fname>
      <_lname>Limbert</_lname>
      <email>example@example.com</email>
      <phones>+79817076617</phones>
  </data>
</xml>
```

Пример запроса с матчингом по custom-полю

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "workflow_id": 12,
    "matching": "custom",
    "field_name": "New_field_7",
    "data": [
        {
            "data": {
                "_fname": "Kristina",
                "_lname": "Lambert",
                "email": "example@example.com",
                "phones": "+79568150395",
                "New_field_7": "00015"
            }
        },
        {
            "data": {
                "_fname": "Christopher",
                "_lname": "Lambert",
                "email": "example@example.com",
                "phones": "+79568150396",
                "New_field_7": "00016"
            }
        }
    ]
}
```


XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <workflow_id>12</workflow_id>
    <matching>custom</matching>
    <field_name>CRM_ID</field_name>
    <data>
        <_fname>Naomy</_fname>
        <_lname>Depp</_lname>
        <email>example@example.com</email>
        <phones>+798144890880</phones>
        <CRM_ID>7830</CRM_ID>
    </data>
    <data>
        <_fname>Nataly</_fname>
        <_lname>Depp</_lname>
        <email>example@example.com</email>
        <phones>+73755334351</phones>
        <CRM_ID>0934</CRM_ID>
    </data>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "task_id": "30421978-7f6d-4f39-913f-fb6cad8e2477"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <task_id>675231ce-63b4-4bca-a153-10d87b938d26</task_id>
</xml>
```

Пример ответа при срабатывании ограничения запроса

JSON

```
{
  "error": 400,
  "error_debug": "http: request body too large",
  "error_text": "Invalid request"
}
```

XML

```
<xml>
  <error>400</error>
  <error_debug>http: request body too large</error_debug>
  <error_text>Invalid request</error_text>
</xml>
```

Проверка статуса задания

URL-адрес

Метод: POST

`https://example.com/api/v1.1/tasks/status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
task_id	string	“task_b2f9fk6a-2479-4a63-b592-f15d4e3b69c5”	Да	Идентификатор задания

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "task_id": "task_b2f9fk6a-2479-4a63-b592-f15d4e3b69c5"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <task_id>task_b2f9fk6a-2479-4a63-b592-f15d4e3b69c5</task_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "status": "completed",
        "progress": 100,
        "result": {
            "error": 0,
            "error_text": "",
            "result": [
                {
                    "profile_id": "61e947333f62fc26d384452c"
                },
                {
                    "profile_id": "61e947333f62fc26d384452e"
                }
            ]
        }
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <status>completed</status>
        <progress>100</progress>
        <result>
            <error>0</error>
            <error_text></error_text>
            <result>
                <profile_id>61e947333f62fc26d384452c</profile_id>
            </result>
            <result>
                <profile_id>61e947333f62fc26d384452e</profile_id>
            </result>
        </result>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Получить список сценариев

Описание

Получить список сценариев.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/workflows/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
format	string	"json"	Нет	Формат возвращаемых данных. По умолчанию – "json".
from_id	int	1	Нет	Идентификатор первого сценария в списке По умолчанию – 1.
limit	int	1	Нет	Лимит на количество возвращаемых сценариев По умолчанию – 1000.
groups	[string]	["group1", "group2"]	Нет	Список групп для поиска По умолчанию – [].
groups_id	[int]	[0, 1]	Нет	Список ID групп для поиска По умолчанию – [].
ui_tags	[string]	["tag1", "tag2"]	Нет	Список тегов для поиска По умолчанию – [].
attributes	JSON object	`"attributes": \{ "season": \{ "options": \["spring_2024", "summer_2024"\] \} \}`	Нет	Значения атрибутов для поиска

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "from_id": 10,
    "limit": 1,
    "ui_tags": [
        "B2C"
    ]
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <from_id>10</from_id>
    <limit>1</limit>
    <ui_tags array='true'>B2C</ui_tags>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "id": 10,
            "updated_user": "ivan_ivanov",
            "update_time": "2025-05-21T09:17:53.362Z",
            "name": "Приветственная цепочка",
            "list_id": 1,
            "is_active": true,
            "ui_tags": [
                "B2C"
            ],
            "mode": "skip",
            "groups": [
                {
                    "id": 2,
                    "name": "Sales and promotion"
                }
            ],
            "attributes": {
                "season": "spring_2024",
                "branch": "marketing"
            },
            "nodes": [
                {
                    "name": "DEFAULT_START_NODE_TITLE",
                    "type": "start",
                    "node_id": 0
                },
                {
                    "name": "ANY_IMPORT_START_NODE_TITLE",
                    "type": "start",
                    "node_id": 2
                },
                {
                    "templates": [
                        1
                    ],
                    "name": "Приветственное письмо",
                    "type": "campaign",
                    "node_id": 3,
                    "campaign_id": 17
                },
                {
                    "name": "1 д.0 ч.0 м.",
                    "type": "delay",
                    "node_id": 4
                },
                {
```

```
{
    "name": "Подтвердил подписку?",
    "type": "condition",
    "node_id": 5
},
{
    "name": "Получил SMS-напоминание?",
    "type": "condition",
    "node_id": 6
},
{
    "templates": [
        5
    ],
    "name": "SMS-напоминание",
    "type": "campaign",
    "node_id": 7,
    "campaign_id": 30
},
{
    "name": "1 д.0 ч.0 м.",
    "type": "delay",
    "node_id": 8
},
{
    "name": "Заявка в колл-центр или отдел продаж ",
    "type": "feedback",
    "node_id": 9
},
{
    "name": "Приветственные бонусы",
    "type": "feedback",
    "node_id": 10
},
{
    "name": "Назначить промокод",
    "type": "promocode_attach",
    "node_id": 11
},
{
    "templates": [
        22
    ],
    "name": "Промо-предложение",
    "type": "campaign",
    "node_id": 12,
    "campaign_id": 39
}
{
    "name": "Переход в сценарий #29",
```

```
                    "type": "workflow_jump",
                    "node_id": 13,
                    "next_workflow_id": 29,
                    "next_node_id": 0
                }
            ]
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "next_from_id": 11,
    "total_count": 20
}
```


XML

```
<xml>
    <data>
        <groups>
            <id>2</id>
            <updated_user>ivan_ivanov</updated_user>
            <update_time>2025-05-21T09:17:53.362Z</update_time>
            <name>Sales and promotion</name>
        </groups>
        <id>10</id>
        <is_active>true</is_active>
        <list_id>1</list_id>
        <mode>skip</mode>
        <name>Приветственная цепочка</name>
        <attributes>
            <season>spring_2024</season>
            <branch>marketing</branch>
        </attributes>
        <nodes>
            <name>DEFAULT_START_NODE_TITLE</name>
            <node_id>0</node_id>
            <type>start</type>
        </nodes>
        <nodes>
            <name>ANY_IMPORT_START_NODE_TITLE</name>
            <node_id>2</node_id>
            <type>start</type>
        </nodes>
        <nodes>
            <campaign_id>17</campaign_id>
            <name>Приветственное письмо</name>
            <node_id>3</node_id>
            <templates>1</templates>
            <type>campaign</type>
        </nodes>
        <nodes>
            <name>1 д.0 ч.0 м.</name>
            <node_id>4</node_id>
            <type>delay</type>
        </nodes>
        <nodes>
            <name>Подтвердил подписку?</name>
            <node_id>5</node_id>
            <type>condition</type>
        </nodes>
        <nodes>
            <name>Получил SMS напоминание?</name>
            <node_id>6</node_id>
            <type>condition</type>
        </nodes>
```

```
        </nodes>
         <nodes>
            <campaign_id>30</campaign_id>
            <name>SMS-напоминание</name>
            <node_id>7</node_id>
            <templates>5</templates>
            <type>campaign</type>
        </nodes>
        <nodes>
            <name>1 д.0 ч.0 м.</name>
            <node_id>8</node_id>
            <type>delay</type>
        </nodes>
        <nodes>
            <name>Заявка в колл-центр или отдел продаж </name>
            <node_id>9</node_id>
            <type>feedback</type>
        </nodes>
        <nodes>
            <name>Приветственные бонусы</name>
            <node_id>10</node_id>
            <type>feedback</type>
        </nodes>
        <nodes>
            <name>Назначить промокод</name>
            <node_id>11</node_id>
            <type>promocode_attach</type>
        </nodes>
        <nodes>
            <campaign_id>39</campaign_id>
            <name>Промо-предложение</name>
            <node_id>12</node_id>
            <templates>22</templates>
            <type>campaign</type>
        </nodes>
        <nodes>
            <name>Переход в сценарий #29</name>
            <next_node_id>0</next_node_id>
            <next_workflow_id>29</next_workflow_id>
            <node_id>13</node_id>
            <type>workflow_jump</type>
        </nodes>
        <ui_tags>B2C</ui_tags>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <next_from_id>11</next_from_id>
    <total_count>20</total_count>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	int	Идентификатор следующего сценария. Указатель можно передать в следующем запросе для получения ещё одной порции данных.
total_count	int	Общее количество полученных сценариев
Объект data		
id	int	Идентификатор сценария
name	string	Название сценария
list_id	int	Идентификатор базы профилей
is_active	bool	Активный сценарий или нет
mode	string	Режим при неоднократном попадании профиля в сценарий
ui_tags	array	Список тегов
groups	array of objects	Список групп
attributes	object	Атрибуты, назначенные сценарию, и их значения
nodes	array of objects	Список элементов сценария
node_id	int	Идентификатор ноды
campaign_id	int	Идентификатор рассылки (для ноды "Каналы")
templates	int	Идентификатор шаблона (для ноды "Каналы")
next_workflow_id	int	Идентификатор следующего сценария (для ноды перехода в сценарий)
next_node_id	int	Идентификатор следующей ноды (для перехода в сценарий)
updated_user	string	Кем был обновлён сценарий
update_time	string	Когда был обновлён сценарий

Активировать сценарий

Описание

Запрос активирует сценарий.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/workflows/activate`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор сценария
skip_profiles	bool	true	Нет	Очистить сценарий перед повторной активацией

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1,
    "skip_profiles": true
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>1</id>
    <skip_profiles>true</skip_profiles>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Деактивировать сценарий

Описание

Запрос деактивирует сценарий.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/workflows/deactivate`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор сценария
skip_profiles	bool	true	Нет	Очистить сценарий перед повторной активацией

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1,
    "skip_profiles": true
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>1</id>
    <skip_profiles>true</skip_profiles>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Импортировать промокоды

Описание

Импортировать промокоды в программу лояльности.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/promocodes/import`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	int	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
codes	JSON array	`["CODE-1", "CODE-2"]`	Да	Список промокодов
loyalty_id	int	1	Да	Идентификатор программы лояльности

Пример запроса

JSON

```
{
    "codes": [
        "CODE-1",
        "CODE-2"
    ],
    "loyalty_id": 6,
    "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <codes>CODE-1</codes>
    <codes>CODE-2</codes>
    <loyalty_id>6</loyalty_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "code": "CODE-1",
            "error": 0,
            "error_text": ""
        },
        {
            "code": "CODE-2",
            "error": 0,
            "error_text": ""
        }
    ],
    "discarded_by_limit": 0,
    "duplicated": 0,
    "error": 0,
    "error_text": "Successful operation",
    "inserted": 2
}
```

XML

```
<xml>
    <data>
        <code>CODE-1</code>
        <error>0</error>
    </data>
    <data>
        <code>CODE-2</code>
        <error>0</error>
    </data>
    <discarded_by_limit>0</discarded_by_limit>
    <duplicated>0</duplicated>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <inserted>2</inserted>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
code	string	Промокод
discarded_by_limit	int	Превышение лимита программы
duplicated	int	Дублирующиеся промокоды
inserted	int	Промокодов импортировано

Получить информацию о промокоде

Описание

Получить подробную информацию о промокоде.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/promocodes/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
code	string	“CODE-12345”	Да	Промокод
loyalty_id	int	1	Да	Идентификатор программы лояльности

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "code": "CODE-1",
    "loyalty_id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <code>CODE-1</code>
    <loyalty_id>5</loyalty_id>
</xml>
```

Пример ответа

JSON


```
{
    "data": {
        "created": "2024-04-01T12:39:34Z",
        "is_attached": false,
        "is_activated": false,
        "is_expired": false,
        "attached": "",
        "activated": "",
        "expired": "",
        "expire_date": "",
        "loyalty_id": 5,
        "code": "CODE-1"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML


```
<xml>
    <data>
        <code>CODE-1</code>
        <created>2024-04-01T12:39:34Z</created>
        <is_activated>false</is_activated>
        <is_attached>false</is_attached>
        <is_expired>false</is_expired>
        <loyalty_id>5</loyalty_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
data.activated	string	Дата и время активации
data.attached	string	Дата и время назначения
data.code	string	Промокод

Параметр	Тип	Описание
data.created	string	Дата и время создания
data.expire_date	string	Дата и время, когда код истечёт
data.expired	string	Дата и время, когда код истёк
data.is_activated	bool	true если код активирован
data.is_attached	bool	true если код назначен
data.is_expired	bool	true если код истёк
data.loyalty_id	int	Идентификатор программы лояльности
data.database_id	int	Идентификатор базы данных, если промокод привязан к профилю
data.profile_id	string	Идентификатор профиля, если промокод привязан

Активировать промокод

Описание

Активировать промокод.

> **CAUTION**
>
> Нельзя активировать промокод, который не присоединен ни к одному из профилей.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/promocodes/activate`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
code	string	“CODE-12345”	Да	Промокод
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
loyalty_id	int	1	Да	Идентификатор программы лояльности

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "code": "Code-1",
    "loyalty_id": 5
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <code>Code-1</code>
    <loyalty_id>5</loyalty_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Обновить промокод

Описание

Обновить дату истечения промокода, активировать или деактивировать.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/promocodes/update`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
activated	bool	true	Нет	Активация/деактивация промокода
code	string	“CODE-12345”	Да	Промокод
expire_date	string	“2018-08-25T08:00:00Z”	Нет	Дата истечения промокода в формате RFC 3339
loyalty_id	int	1	Да	Идентификатор программы лояльности

CAUTION

Ожидается, что как минимум один из параметров — `"activated"` или `"expire_date"` — будет передан в запросе.

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "activated": true,
    "code": "CODE-12345",
    "expire_date": "2018-08-25T08:00:00Z",
    "loyalty_id": 1,
    "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <activated>true</activated>
    <code>Code-1</code>
    <expire_date>2024-08-10T08:00:00Z</expire_date>
    <loyalty_id>6</loyalty_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Привязать промокод к профилю

Описание

Назначить промокод подписчику.

Если в запрос передан параметр `"code"`, то выполняется поиск по данному коду.

Будет назначен:

- Если промокод найден (присоединён к множественным профилям или свободен).
- Если промокод не найден, и разрешено создание новых.

Если в запрос не передан параметр `"code"`, то будет назначен промокод из программы лояльности (если есть свободные или разрешено создание новых).

URL-адрес

Метод: POST

`https://example.com/api/v1.1/promocodes/attach`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
code	string	"CODE-12345"	Нет	Код
db_id	int	1	Да	Идентификатор базы данных
format	string	"json"	Нет	Формат возвращаемых данных (по умолчанию – "json")
loyalty_id	int	1	Да	Идентификатор программы лояльности
Поиск профиля подписчика				
matching	string	"email" "email_profile" "phone" "profile_id" "custom"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика (по умолчанию — email)
email	string	"john@example.com"	"matching":"email" — по профилю и подпискам "matching":"email_profile" — только по профилю	Email-адрес

phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string		"matching":"custom"	Название поля для поиска
field_value	string/int		"matching":"custom"	Значение поля для поиска
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 1,
    "matching": "profile_id",
    "profile_id": "abcdefghijklmnqrstuvwxyz",
    "loyalty_id": 5,
    "code": "CODE-1"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <matching>profile_id</matching>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
    <loyalty_id>5</loyalty_id>
    <code>CODE-1</code>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Отвязать промокод от профиля

Описание

Отвязать промокод от профиля.

Если для промокода не выбрано уникальное значение и по нему найдется более, чем один профиль, то API вернет ошибку.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/promocodes/detach`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
code	string	"CODE-12345"	Нет	Код
db_id	int	1	Да	Идентификатор базы данных
format	string	"json"	Нет	Формат возвращаемых данных (по умолчанию – "json")
loyalty_id	int	1	Да	Идентификатор программы лояльности
Поиск профиля подписчика				
matching	string	"email" "email_profile" "phone" "profile_id" "custom"	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика (по умолчанию — email)
email	string	"john@example.com"	"matching":"email" — по профилю и подпискам "matching":"email_profile" — только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string		"matching":"custom"	Название поля для поиска

field_value	string/int		"matching":"custom"	Значение поля для поиска
resource_id	int	24	Нет	Идентификатор ресурса для поиска по email и phone

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "code": "Code-1",
    "db_id": 1,
    "loyalty_id": 6,
    "matching": "profile_id",
    "profile_id": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <code>Code-1</code>
    <db_id>1</db_id>
    <loyalty_id>4</loyalty_id>
    <matching>profile_id</matching>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": null,
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data></data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить все промокоды

Описание

Позволяет выгрузить все промокоды и их текущий статус одним списком данных.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/promocodes/export_all`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
loyalty_id	int	1	Да	Идентификатор программы лояльности

Пример запроса

JSON

```
{
  "loyalty_id": 6,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
    <loyalty_id>6</loyalty_id>
    <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "code": "Code-1"
    },
    {
      "code": "Code-2",
      "attached": "2024-04-01T14:02:05Z",
      "activated": "2024-04-01T14:04:12Z"
    },
    {
      "code": "Code-3"
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```


XML


```
<xml>
    <data>
        <code>Code-1</code>
    </data>
    <data>
        <activated>2024-04-01T14:04:12Z</activated>
        <attached>2024-04-01T14:02:05Z</attached>
        <code>Code-2</code>
    </data>
    <data>
        <code>Code-3</code>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
data	array	Список промокодов
data.code	string	Идентификатор промокода и сам промокод
data.attached	string	Дата присоединения к профилю, нет поля если не присоединялось
data.activated	string	Дата активации промокода, нет поля если активации не было

Параметр	Тип	Описание
data.expire_date	string	Дата, когда промокод будет просрочен, может быть не установлено
data.expired	string	Дата, когда промокод был просрочен, нет поля если не просрочен
error	int	Код ошибки
error_text	string	Текст ошибки

Получить уровень профиля в программе лояльности

Описание

Позволяет получить текущий уровень участника программы лояльности в рамках запрашиваемой группы уровней.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/account/tiers`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
tiers_group_id	int	10	Да	Идентификатор группы уровней (например, 10)
matching	string	“email”	Да	Способ поиска профиля. Возможные значения: `"email"`, `"phone"`, `"profile_id"`, `"custom"`
field_name	string	“user_id”	Да, если `matching` = `custom`	Название пользовательского поля, используемого для поиска
field_value	string / int	123456	Да, если `matching` = `custom`	Значение пользовательского поля
email	string	“user@example.com”	Да, если `matching` = `email`	Email для поиска профиля
phone	string	“+1234567890”	Да, если `matching` = `phone`	Номер телефона в международном формате
profile_id	string	“profile123”	Да, если `matching` = `profile_id`	Идентификатор профиля
token	string	“abc123def456”	Да	API-токен для авторизации

Пример запроса

JSON

```
{
  "loyalty_program_id": 123,
  "tiers_group_id": 10,
  "matching": "email",
  "field_name": "user_id",
  "field_value": 123456,
  "email": "user@example.com",
  "phone": "+1234567890",
  "profile_id": "profile123",
  "token": "abc123def456"
}
```

XML

```
<xml>
    <loyalty_program_id>123</loyalty_program_id>
    <tiers_group_id>10</tiers_group_id>
    <matching>email</matching>
    <field_name>user_id</field_name>
    <field_value>123456</field_value>
    <email>user@example.com</email>
    <phone>+1234567890</phone>
    <profile_id>profile123</profile_id>
    <token>abc123def456</token>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": [
    {
      "tiers_group_type": "cumulative",
      "tier_eid": "silver1-2",
      "tier_id": 13,
      "tier_order": 1,
      "reached_at": "2024-06-15T09:12:34Z"
    },
    {
      "tiers_group_type": "cumulative",
      "tier_eid": "gold1-3",
      "tier_id": 14,
      "tier_order": 2,
      "reached_at": "2024-09-01T12:34:56Z"
    }
  ]
}
```


XML


```
<xml>
    <data>
        <tiers_group_type>cumulative</tiers_group_type>
        <tier_eid>silver1-2</tier_eid>
        <tier_id>13</tier_id>
        <tier_order>1</tier_order>
        <reached_at>2024-06-15T09:12:34Z</reached_at>
    </data>
    <data>
        <tiers_group_type>cumulative</tiers_group_type>
        <tier_eid>gold1-3</tier_eid>
        <tier_id>14</tier_id>
        <tier_order>2</tier_order>
        <reached_at>2024-09-01T12:34:56Z</reached_at>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
tiers_group_type	string	Тип группы уровней
tier_eid	string	Внешний идентификатор уровня
tier_id	int	Идентификатор уровня
tier_order	int	Порядковый номер уровня (от 0)
reached_at	string	Дата достижения уровня в формате RFC3339

Экспорт транзакций баллов

Описание

Создание задачи на формирование отчёта по транзакциям выбранных баллов за период. В ответе возвращается идентификатор задачи для отслеживания статуса.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/transactions_export`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
points_currency_eid	string	“bonus_points”	Да, если не передан `points_currency_iid`	Внешний идентификатор валюты баллов
points_currency_iid	int	1	Да, если не передан `points_currency_eid`	Внутренний идентификатор валюты баллов
from_id	string	“a1b2c3d4-e5f6-7890-g1h2-i3j4k5l6m7n8”	Нет	Идентификатор транзакции (UUID) для пагинации; если не указан, экспорт с начала периода
date_from	string (RFC3339)	“2023-09-01T00:00:00Z”	Да	Начало периода (включительно)
date_to	string (RFC3339)	“2023-09-30T23:59:59Z”	Да	Окончание периода (включительно)

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "points_currency_eid": "bonus_points",
  "date_from": "2023-09-01T00:00:00Z",
  "date_to": "2023-09-30T23:59:59Z",
  "from_id": "a1b2c3d4-e5f6-7890-g1h2-i3j4k5l6m7n8"
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <points_currency_eid>bonus_points</points_currency_eid>
  <date_from>2023-09-01T00:00:00Z</date_from>
  <date_to>2023-09-30T23:59:59Z</date_to>
  <from_id>a1b2c3d4-e5f6-7890-g1h2-i3j4k5l6m7n8</from_id>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "task_id": "task_b2f9fd6a-2499-4a63-b592-f15d4e3b69c5"
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <task_id>task_b2f9fd6a-2499-4a63-b592-f15d4e3b69c5</task_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
task_id	string (UUID)	Идентификатор задачи на экспорт

Сгораемые баллы за период

Описание

Возвращает количество баллов, подлежащих сгоранию за выбранный период по указанной валюте баллов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/account/expiring_points`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
points_currency_eid	string	“bonus_points”	Да, если не передан `points_currency_iid`	Внешний идентификатор валюты баллов
points_currency_iid	int	1	Да, если не передан `points_currency_eid`	Внутренний идентификатор валюты баллов
date_from	string (RFC3339)	“2023-09-01T00:00:00Z”	Да	Начало периода (включительно)
date_to	string (RFC3339)	“2023-09-30T23:59:59Z”	Да	Окончание периода (включительно)
matching	string	“email”	Да	Способ поиска профиля. Возможные значения: `"email"`, `"phone"`, `"profile_id"`, `"custom"`
field_name	string	“user_id”	Да, если `matching` = `custom`	Название пользовательского поля
field_value	string / int	“12345”	Да, если `matching` = `custom`	Значение пользовательского поля
email	string	“user@example.com”	Да, если `matching` = `email`	Email участника
phone	string	“+1234567890”	Да, если `matching` = `phone`	Телефон участника в международном формате
profile_id	string	“profile_123”	Да, если `matching` = `profile_id`	Идентификатор профиля

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "points_currency_eid": "bonus_points",
  "matching": "email",
  "email": "user@example.com",
  "date_from": "2023-09-01T00:00:00Z",
  "date_to": "2023-09-30T23:59:59Z"
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <points_currency_eid>bonus_points</points_currency_eid>
  <matching>email</matching>
  <email>user@example.com</email>
  <date_from>2023-09-01T00:00:00Z</date_from>
  <date_to>2023-09-30T23:59:59Z</date_to>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "expiring_points": "150.24"
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <expiring_points>150.24</expiring_points>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
expiring_points	string	Количество баллов, которые сгорят в указанный период

Получение транзакций по счёту профиля

Описание

Возвращает историю транзакций по балльному счёту участника программы лояльности. Поддерживает пагинацию и фильтрацию по валюте баллов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/account/transactions`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
points_currency_eid	string	“bonus_points”	Да, если не передан `points_currency_iid`	Внешний идентификатор валюты баллов
points_currency_iid	int	1	Да, если не передан `points_currency_eid`	Внутренний идентификатор валюты баллов
from_id	string	“a1b2c3d4-e5f6-7890-g1h2-i3j4k5l6m7n8”	Нет	UUID последней транзакции для пагинации (записи с ID > from_id)
limit	int	20	Нет	Количество записей на страницу (1–100, по умолчанию 20)
matching	string	“email”	Да	Способ поиска профиля. Возможные значения: `"email"`, `"phone"`, `"profile_id"`, `"custom"`
field_name	string	“user_id”	Да, если `matching` = `custom`	Название пользовательского поля
field_value	string / int	“12345”	Да, если `matching` = `custom`	Значение пользовательского поля
email	string	“user@example.com”	Да, если `matching` = `email`	Email участника
phone	string	“+1234567890”	Да, если `matching` = `phone`	Телефон участника в международном формате
profile_id	string	“profile_123”	Да, если `matching` = `profile_id`	Идентификатор профиля

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "points_currency_eid": "bonus_points",
  "matching": "email",
  "email": "user@example.com",
  "limit": 20
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <points_currency_eid>bonus_points</points_currency_eid>
  <matching>email</matching>
  <email>user@example.com</email>
  <limit>20</limit>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "transaction_id": "a1b2c3d4-e5f6-7890-g1h2-i3j4k5l6m7n8",
      "transaction_eid": "trans_12345",
      "created_at": "2023-09-01T12:34:56Z",
      "transaction_type": "accrual",
      "amount": "100.1",
      "balance_before": "500.5",
      "balance_after": "600.6",
      "promo_iid": 42,
      "promo_eid": "promo_2023",
      "description": "Начисление баллов за покупку",
      "activation_date": "2023-09-01T12:34:56Z",
      "expiration_date": "2023-12-31T23:59:59Z",
      "source_type": "order",
      "source_iid": 789,
      "source_sid": "order_123",
      "source_subiid": 1,
      "is_temp": false,
      "custom_json": {"key": "value"},
      "product_eid": "prod_123",
      "sku_eid": "sku_456",
      "canceled": false
    }
  ],
  "next_from_id": "b2c3d4e5-f6g7-8901-h2i3-j4k5l6m7n8o9",
  "total_count": 150,
  "error": 0,
  "error_text": "Successful operation"
}
```


XML

```
<xml>
  <data>
    <transaction_id>a1b2c3d4-e5f6-7890-g1h2-i3j4k5l6m7n8</transaction_id>
    <transaction_eid>trans_12345</transaction_eid>
    <created_at>2023-09-01T12:34:56Z</created_at>
    <transaction_type>accrual</transaction_type>
    <amount>100.1</amount>
    <balance_before>500.5</balance_before>
    <balance_after>600.6</balance_after>
    <promo_iid>42</promo_iid>
    <promo_eid>promo_2023</promo_eid>
    <description>Начисление баллов за покупку</description>
    <activation_date>2023-09-01T12:34:56Z</activation_date>
    <expiration_date>2023-12-31T23:59:59Z</expiration_date>
    <source_type>order</source_type>
    <source_iid>789</source_iid>
    <source_sid>order_123</source_sid>
    <source_subiid>1</source_subiid>
    <is_temp>false</is_temp>
    <custom_json>{"key": "value"}</custom_json>
    <product_eid>prod_123</product_eid>
    <sku_eid>sku_456</sku_eid>
    <canceled>false</canceled>
  </data>
  <next_from_id>b2c3d4e5-f6g7-8901-h2i3-j4k5l6m7n8o9</next_from_id>
  <total_count>150</total_count>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
transaction_id	string (UUID)	Уникальный идентификатор транзакции
transaction_eid	string	Внешний ID транзакции
created_at	string (RFC3339)	Дата и время создания транзакции
transaction_type	string	Тип транзакции (`accrual`, `redeem`, `expire`, и др.)
amount	string	Количество начисленных или списанных баллов

Параметр	Тип	Описание
balance_before	string	Баланс до проведения операции
balance_after	string	Баланс после проведения операции
promo_iid	int	Внутренний ID промоакции
promo_eid	string	Внешний ID промоакции
description	string	Описание транзакции
activation_date	string (RFC3339)	Дата активации баллов
expiration_date	string (RFC3339)	Дата истечения баллов
source_type	string	Источник транзакции (`api`, `manual`, `system`, `order`, `workflow`, `tier_group`)
product_eid	string	Внешний ID продукта
sku_eid	string	Внешний ID SKU
canceled	boolean	Флаг отмены транзакции
next_from_id	string (UUID)	UUID для следующей страницы
total_count	int	Общее количество транзакций

Получение списка триггерных промоакций

Описание

Возвращает список триггерных промоакций программы лояльности. Поддерживается пагинация и фильтрация по активности.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/trigger_promotions`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
from_id	int	0	Нет	ID промоакции для пагинации (записи с ID > `from_id`)
limit	int	20	Нет	Количество записей (1–100), по умолчанию 20
is_active	boolean	true	Нет	Фильтр по активности: `true` — активные, `false` — неактивные, не указан — все

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "limit": 20,
  "is_active": true
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <limit>20</limit>
  <is_active>true</is_active>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "id": 5,
      "name": "Триггерная акция: Бонус за покупку",
      "description": "Начисление бонусов за каждую покупку",
      "external_id": "trigger_promo_2023_q3",
      "start_date": "2023-09-01T00:00:00Z",
      "end_date": "2023-09-30T23:59:59Z",
      "is_active": true,
      "redeem_rules": [
        {
          "points_currency_iid": 1,
          "points_currency_eid": "curr1",
          "static_amount": "100.16",
          "min_amount": "50",
          "max_amount": "500"
        }
      ],
      "accrual_rules": [
        {
          "points_currency_iid": 1,
          "points_currency_eid": "curr1",
          "static_amount": "100",
          "min_amount": "50",
          "max_amount": "500"
        }
      ]
    }
  ],
  "next_from_id": 8,
  "total_count": 134,
  "error": 0,
  "error_text": "Successful operation"
}
```


XML

```
<xml>
  <data>
    <id>5</id>
    <name>Триггерная акция: Бонус за покупку</name>
    <description>Начисление бонусов за каждую покупку</description>
    <external_id>trigger_promo_2023_q3</external_id>
    <start_date>2023-09-01T00:00:00Z</start_date>
    <end_date>2023-09-30T23:59:59Z</end_date>
    <is_active>true</is_active>
  </data>
  <next_from_id>8</next_from_id>
  <total_count>134</total_count>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Уникальный ID промоакции
name	string	Название промоакции
description	string	Описание промоакции
external_id	string	Внешний идентификатор промоакции
start_date	string (RFC3339)	Дата начала
end_date	string (RFC3339)	Дата окончания
is_active	boolean	Флаг активности
redeem_rules	array	Правила списания (`тип баллов`, значения)
accrual_rules	array	Правила начисления (`тип баллов`, значения)
next_from_id	int	ID для следующей страницы
total_count	int	Общее количество промоакций

Начисление баллов участнику

Описание

Начисляет баллы на счёт участника программы лояльности с использованием триггерной промоакции.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/accrue_points`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
promotion_id	int	24	Нет	Идентификатор триггерной промоакции
points_currency_iid	int	1	Нет	Внутренний идентификатор типа баллов
points_currency_eid	string	“stars”	Нет	Внешний идентификатор типа баллов
amount	string	“100”	Да	Количество начисляемых баллов
is_temp	boolean	false	Нет	Признак временной транзакции. Может быть отменена по таймауту или отдельным запросом
temp_timeout_sec	int	123	Нет	Время в секундах, после которого временная транзакция будет отменена при отсутствии подтверждения
created_at	string	“2024-09-01T12:34:56Z”	Нет	Дата и время создания транзакции (если не указано — текущее время) в формате RFC3339
activation_date	string	“2024-09-01T12:34:56Z”	Нет	Дата активации баллов (если не указано — баллы сразу активны) в формате RFC3339
expiration_date	string	“2024-12-31T23:59:59Z”	Нет	Дата сгорания баллов (если не указано — берётся из настроек типа баллов) в формате RFC3339
transaction_eid	string	“uiuiuiiu-ouiihfs-sdfsdf”	Нет	Внешний идентификатор транзакции

Параметр	Тип	Пример	Обязательный	Описание
external_data	object	`{"order_id": "123"}`	Нет	Произвольные дополнительные данные транзакции (JSON-объект)
matching	string	“email”	Да	Способ поиска профиля: `email`, `phone`, `profile_id`, `custom`
field_name	string	“user_id”	Да, если `matching = custom`	Название пользовательского поля для поиска
field_value	string / int	123456	Да, если `matching = custom`	Значение пользовательского поля
email	string	“user@example.com”	Да, если `matching = email`	Email участника
phone	string	“+1234567890”	Да, если `matching = phone`	Телефон участника в международном формате
profile_id	string	“profile123”	Да, если `matching = profile_id`	Идентификатор профиля

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "promotion_id": "PROMO123",
  "points_currency_eid": "stars",
  "amount": "100",
  "matching": "email",
  "email": "user@example.com",
  "is_temp": false,
  "transaction_eid": "uiuiuiiu-ouiihfs-sdfsdf",
  "external_data": {
    "order_id": "ORDER-123"
  }
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <promotion_id>PROMO123</promotion_id>
  <points_currency_eid>stars</points_currency_eid>
  <amount>100</amount>
  <matching>email</matching>
  <email>user@example.com</email>
  <is_temp>false</is_temp>
  <transaction_eid>uiuiuiiu-ouiihfs-sdfsdf</transaction_eid>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": {
    "transaction_id": "TRANS12345",
    "balance_before": "500.00",
    "balance_after": "600.50"
  }
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <transaction_id>TRANS12345</transaction_id>
    <balance_before>500.00</balance_before>
    <balance_after>600.50</balance_after>
  </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки

Параметр	Тип	Описание
error_text	string	Текст ошибки
transaction_id	string	Идентификатор транзакции начисления баллов
balance_before	string	Баланс баллов до начисления (строка)
balance_after	string	Баланс баллов после начисления (строка)

Списание баллов участника

Описание

Списывает баллы со счёта участника программы лояльности с использованием триггерной промоакции.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/redeeme_points`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
promotion_id	int	123	Нет	Идентификатор триггерной промоакции
points_currency_iid	int	1	Нет	Внутренний идентификатор типа баллов
points_currency_eid	string	“stars”	Нет	Внешний идентификатор типа баллов (например, `"stars"`)
amount	string	“100”	Да	Количество списываемых баллов
is_temp	boolean	false	Нет	Признак временной транзакции. Может быть отменена по таймауту или отдельным запросом
temp_timeout_sec	int	123	Нет	Время в секундах, после которого временная транзакция будет отменена при отсутствии подтверждения
created_at	string	“2024-09-01T12:34:56Z”	Нет	Дата и время создания транзакции (если не указано — текущее время) в формате RFC3339
transaction_eid	string	“uiuiuiiu-ouiihfs-sdfsdf”	Нет	Внешний идентификатор транзакции
external_data	object	`{"order_id": "123"}`	Нет	Произвольные дополнительные данные транзакции (JSON-объект)
matching	string	“email”	Да	Способ поиска профиля: `email`, `phone`, `profile_id`, `custom`
field_name	string	“user_id”	Да, если `matching = custom`	Название пользовательского поля для поиска

Параметр	Тип	Пример	Обязательный	Описание
field_value	string / int	123456	Да, если `matching = custom`	Значение пользовательского поля
email	string	“user@example.com”	Да, если `matching = email`	Email участника
phone	string	“+1234567890”	Да, если `matching = phone`	Телефон участника в международном формате
profile_id	string	“profile123”	Да, если `matching = profile_id`	Идентификатор профиля

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "promotion_id": "PROMO123",
  "points_currency_eid": "stars",
  "amount": "100",
  "matching": "email",
  "email": "user@example.com",
  "is_temp": false,
  "transaction_eid": "uiuiuiiu-ouiihfs-sdfsdf",
  "external_data": {
    "order_id": "ORDER-123"
  }
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <promotion_id>PROMO123</promotion_id>
  <points_currency_eid>stars</points_currency_eid>
  <amount>100</amount>
  <matching>email</matching>
  <email>user@example.com</email>
  <is_temp>false</is_temp>
  <transaction_eid>uiuiuiiu-ouiihfs-sdfsdf</transaction_eid>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": {
    "transaction_id": "TRANS12345",
    "balance_before": "500.00",
    "balance_after": "399.50"
  }
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <transaction_id>TRANS12345</transaction_id>
    <balance_before>500.00</balance_before>
    <balance_after>399.50</balance_after>
  </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
transaction_id	string	Идентификатор транзакции списания баллов
balance_before	string	Баланс баллов до списания (строка)
balance_after	string	Баланс баллов после списания (строка)

Подтверждение временной транзакции

Описание

Подтверждает временную транзакцию, созданную с флагом `is_temp`. Если подтверждается временная транзакция начисления баллов, создаётся новая полноценная транзакция.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/commit_transaction`

Параметры запроса

Должен быть указан хотя бы один из идентификаторов:

Параметр	Тип	Пример	Обязательный	Описание
transaction_id	string	“TRANS12345”	Нет	Идентификатор временной транзакции
transaction_eid	string	“TRANS12345”	Нет	Внешний идентификатор временной транзакции

Пример запроса

JSON

```
{
  "transaction_id": "TRANS12345"
}
```

XML

```
<xml>
  <transaction_id>TRANS12345</transaction_id>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": {
    "transaction_id": "TRANS12345",
    "balance": "399.50"
  }
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <transaction_id>TRANS12345</transaction_id>
    <balance>399.50</balance>
  </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
transaction_id	string	Идентификатор подтверждённой транзакции
balance	string	Текущий баланс активных баллов пользователя

Предварительный расчет заказа

Описание

Метод `preorder` позволяет выполнить **предварительный расчет заказа** с учетом промоакций и баллов участника программы лояльности.

Результат запроса может использоваться для отображения итоговых сумм, скидок и списаний **до создания окончательного заказа**.

> **TIP**
>
> Метод не создает заказ в системе, а возвращает расчетное состояние заказа с примененными правилами маркета и лояльности. Для создания рассчитанного заказа после успешного выполнения preorder используйте метод `preorder_commit`.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/orders/preorder`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
external_id	string	"12"	Да	Внешний идентификатор заказа
matching	string	"email" "phone" "profile_id" "custom"	Да	Способ поиска профиля участника программы лояльности
email	string	"example@example.com"	Если `matching = email`	Email участника
phone	string	"+79000000000"	Если `matching = phone`	Телефон участника
profile_id	string	"696e304f547840a7286619e0"	Если `matching = profile_id`	Идентификатор профиля

data	JSON object	\{ "endpoint": "site", "status": "new", "currency": "RUB", "total_price": "139999.00", "lines": [...] \}	Да	Объект [`data`](#data-object-preorder) содержит данные заказа

TIP

Поле `lines` обязательно и должно содержать как минимум одну позицию заказа.

Объект data

Параметр	Тип	Пример	Обязательный	Описание
endpoint	string	"bot", "website"	**Да**	Источник заказа (точка контакта). Должен быть настроен в Маркете
status	string	"new", "confirmed", "paid"	**Да**	Статус заказа. Используются статусы, настроенные в платформе
currency	string	"RUB", "USD", "EUR"	**Да**	Валюта заказа (трехбуквенный код ISO)
total_price	string	"2500.00"	**Да**	Общая стоимость заказа до применения скидок
final_total_price	string	"2150.00"	**Да**	Итоговая стоимость заказа после применения всех скидок
delivery_cost	string	"300.00"	Нет	Стоимость доставки
create_time	string (ISO 8601)	"2026-02-18T13:01:05.141Z"	Нет	Дата и время создания заказа. Если не передано, используется текущее время сервера
region	string	"region_moscow"	Нет	Идентификатор региона для применения региональных цен и акций
lines	array	См. ниже раздел "Массив lines"	**Да**	Массив позиций заказа. Должен содержать минимум 1 элемент
promocodes	array	См. ниже раздел "Массив promocodes"	Нет	Массив промокодов, примененных к заказу
applied_promotions	array	См. ниже раздел "Массив applied_promotions"	Нет	Промоакции, примененные к данному заказу.

redeem_points	array	См. ниже раздел "Массив redeem_points"	**Да**, если указано `type="redeem_points"` в `applied_promotions`	Массив списаний баллов по типам валют
custom_fields	object	`{ "field1": "value1", "field2": "value2" }`	Нет	Дополнительные поля заказа
tags	array	`[ "summer_sale", "first_order" ]`	Нет	Теги для маркировки заказа

Массив lines

Массив содержит информацию о позициях заказа. Должен содержать минимум 1 позицию.

Каждая позиция вынесена в отдельный объект внутри массива со следующими параметрами:

Параметр	Тип	Пример	Обязательный	Описание
external_id	string	"line_1"	**Да**	Внешний идентификатор позиции в системе клиента
product	string	"iphone17pmb256"	**Да***	Идентификатор продукта
sku	string	"IPHONE17-256-BLACK"	**Да***	Идентификатор SKU
line_number	integer	1	Нет	Порядковый номер линии в заказе
count	integer	1	**Да**	Количество единиц товара
base_price_per_item	string	"139999.00"	**Да**	Базовая цена за единицу товара
min_price_per_item	string	"130000.00"	Нет	Минимально допустимая цена за единицу товара
final_price_per_item	string	"139500.00"	**Да**	Финальная цена за единицу после применения всех скидок
status	string	"new"	Нет	Статус позиции заказа
custom_fields	object	\{"color": "black", "size": "XL"\}	Нет	Дополнительные поля позиции
applied_promotions	array	См. ниже раздел "Массив applied_promotions"	Нет	Промоакции, примененные к данной позиции.

* — обязателен **один из параметров**: `product` или `sku`

Массив promocodes

Массив содержит объекты с информацией о промокодах.

Полное описание структуры промокода можно найти в этой статье.

Параметр	Тип	Пример	Обязательный	Описание
code	string	"SUMMER20"	**Да**	Код промокода
pool_id	integer	42	**Да**	Идентификатор пула промокодов

Массив redeem_points

Массив с инфомацией о списании баллов лояльности. Каждый объект описывает отдельное списание.

Полное описание структуры списания баллов можно найти в этой статье.

Параметр	Тип	Пример	Обязательный	Описание
points_currency_iid	integer	1	**Да***	Внутренний идентификатор типа баллов
points_currency_eid	string	"bonus1"	**Да***	Внешний идентификатор типа баллов
name	string	"Бонусные баллы"	Нет	Название типа баллов
amount	integer	499	**Да**	Количество баллов для списания

* — обязателен **один из параметров**: `points_currency_iid` или `points_currency_eid`

Массив applied_promotions

> **TIP**
>
> Массив `applied_promotions` может содержаться как в объекте `data`, так и в массиве `lines` внутри этого объекта. В первом случае в массиве будут переданы промоакции, применяемые **к продуктам**, во втором — акции, применяемые **к заказу**.

Массив применяемых к заказу промоакций. Каждый объект массива описывает отдельную промоакцию.

Параметр	Тип	Пример	Обязательный	Описание
type	string	"discount", "accrue_points", "redeem_points", "code_issue", "message"	**Да**	Тип промоакции

promotion	object	`"promotion": { "external_id": "promo_1", "name": "Summer Sale" }`	**Да**	Объект с идентификатором и названием промоакции
amount	string	"150.00"	**Да,** если `type` равен `discount`,`accrue_points` или `redeem_points`	Размер скидки или количество баллов
points_info	object	`"points_info": { "points_currency_iid": 1, "points_currency_eid": "bonus1", "name": "Бонусные баллы", "amount": "50" }`	**Да,** если `type` равен `accrue_points` или `redeem_points`	Объект с информацией о баллах

Структура объекта `points_info` выглядит следующим образом:

Параметр	Тип	Пример	Обязательный	Описание
points_currency_iid	integer	1	**Да***	Внутренний идентификатор типа баллов
points_currency_eid	string	"bonus1"	**Да***	Внешний идентификатор типа баллов
name	string	"Бонусные баллы"	Нет	Название типа баллов
amount	string	"6999"	**Да**	Количество баллов
expiration_date	string (ISO 8601)	"2027-02-18T00:00:00.000Z"	Нет	Дата сгорания начисленных баллов

* — обязателен один из параметров: `points_currency_iid` или `points_currency_eid`

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "external_id": "12",
  "matching": "email",
  "email": "example@example.com",
  "data": {
    "endpoint": "site",
    "status": "new",
    "delivery_cost": "0.00",
    "total_price": "139999.00",
    "currency": "RUB",
    "redeem_points": [
      {
        "points_currency_iid": 1,
        "amount": 499
      }
    ],
    "lines": [
      {
        "external_id": "line___ASTGRf",
        "product": "iphone17pmb256",
        "base_price_per_item": "139999.00",
        "min_price_per_item": "0.00",
        "final_price_per_item": "139999.00",
        "count": 1,
        "custom_fields": {},
        "currency": "RUB"
      }
    ]
  }
}
```


XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <external_id>12</external_id>
  <matching>email</matching>
  <email>example@example.com</email>
  <data>
    <endpoint>site</endpoint>
    <status>new</status>
    <delivery_cost>0.00</delivery_cost>
    <total_price>139999.00</total_price>
    <currency>RUB</currency>
    <redeem_points>
      <points_currency_iid>1</points_currency_iid>
      <amount>499</amount>
    </redeem_points>
    <lines>
      <external_id>line___ASTGRf</external_id>
      <product>iphone17pmb256</product>
      <base_price_per_item>139999.00</base_price_per_item>
      <min_price_per_item>0.00</min_price_per_item>
      <final_price_per_item>139999.00</final_price_per_item>
      <count>1</count>
      <currency>RUB</currency>
    </lines>
  </data>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "profile": {
            "profile_id": "696e304f547840a7286619e0"
        },
        "order": {
            "processing_target": "calculated",
            "total_price": "139999.00",
            "final_total_price": "139500.00",
            "delivery_cost": "0.00",
            "currency": "RUB",
            "lines": [
                {
                    "external_id": "line___ASTGRf",
                    "line_number": 0,
                    "count": 1,
                    "product": "iphone17pmb256",
                    "base_price_per_item": "139999.00",
                    "min_price_per_item": "0.00",
                    "final_price_per_item": "139500.00",
                    "custom_fields": {},
                    "applied_promotions": [
                        {
                            "type": "accrue_points",
                            "promotion": {
                                "external_id": "bonus",
                                "name": "Начисление баллов"
                            },
                            "points_info": {
                                "points_currency_iid": 1,
                                "points_currency_eid": "bonus1",
                                "name": "Основной балл ",
                                "amount": "6999"
                            }
                        },
                        {
                            "type": "redeem_points",
                            "promotion": {
                                "external_id": "bonus",
                                "name": "Списание баллов"
                            },
                            "points_info": {
                                "points_currency_iid": 1,
                                "points_currency_eid": "bonus1",
                                "name": "Основной балл ",
                                "amount": "499"
                            }
                        }
                    ]
```

```
                    ]
                }
            ],
            "custom_fields": {},
            "applied_promotions": [],
            "points_info": [
                {
                    "points_currency_iid": 1,
                    "points_currency_eid": "bonus1",
                    "name": "Основной балл ",
                    "status": "success",
                    "active_amount_for_current_order": "20999",
                    "accrue_points_amount_for_current_order": "6999",
                    "redeemed_points_amount_for_current_order": "499"
                }
            ],
            "promocodes_info": []
        }
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
  <data>
    <profile>
      <profile_id>696e304f547840a7286619e0</profile_id>
    </profile>
    <order>
      <processing_target>calculated</processing_target>
      <total_price>139999.00</total_price>
      <final_total_price>139500.00</final_total_price>
      <delivery_cost>0.00</delivery_cost>
      <currency>RUB</currency>
      <lines>
        <external_id>line___ASTGRf</external_id>
        <line_number>0</line_number>
        <count>1</count>
        <product>iphone17pmb256</product>
        <base_price_per_item>139999.00</base_price_per_item>
        <min_price_per_item>0.00</min_price_per_item>
        <final_price_per_item>139500.00</final_price_per_item>
        <applied_promotions>
          <type>accrue_points</type>
          <promotion>
            <external_id>bonus</external_id>
            <name>Начисление баллов</name>
          </promotion>
          <points_info>
            <points_currency_iid>1</points_currency_iid>
            <points_currency_eid>bonus1</points_currency_eid>
            <name>Основной балл </name>
            <amount>6999</amount>
          </points_info>
        </applied_promotions>
        <applied_promotions>
          <type>redeem_points</type>
          <promotion>
            <external_id>bonus</external_id>
            <name>Начисление баллов</name>
          </promotion>
          <points_info>
            <points_currency_iid>1</points_currency_iid>
            <points_currency_eid>bonus1</points_currency_eid>
            <name>Основной балл </name>
            <amount>499</amount>
          </points_info>
        </applied_promotions>
      </lines>
      <points_info>
        <points_currency_iid>1</points_currency_iid>
        <points_currency_eid>bonus1</points_currency_eid>
```

```
        <points_currency_eid>bonus1</points_currency_eid>
        <name>Основной балл </name>
        <status>success</status>
        <active_amount_for_current_order>20999</active_amount_for_current_order>

<accrue_points_amount_for_current_order>6999</accrue_points_amount_for_current_order>

<redeemed_points_amount_for_current_order>499</redeemed_points_amount_for_current_order>
      </points_info>
    </order>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
data	object	Данные результата расчета
error	int	Код ошибки
error_text	string	Текст ошибки

Объект `data` содержит следующую структуру.

Параметр	Тип	Описание
profile	object	Профиль участника. Содержит идентификатор профиля
order	object	Рассчитанный заказ. Подробнее

Подтверждение заказа

Описание

Метод `preorder_commit` выполняет **финальное подтверждение заказа** после успешного предварительного расчета через метод `preorder`. Метод создает заказ в системе, применяет все рассчитанные скидки, начисляет и списывает баллы, активирует промокоды.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/orders/preorder_commit`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	**Да**	API-токен
external_id	string	"order_12345"	**Да**	Внешний идентификатор заказа в системе клиента
profile_id	string	"696e304f547840a7286619e0"	**Да***	Идентификатор профиля. Поле, используемое для матчинга
email	string	"user@example.com"	**Да***	Email участника. Поле, используемое для матчинга
phone	string	"+79991234567"	**Да***	Телефон участника. Поле, используемое для матчинга
skip_triggers	boolean	false	Нет	Отключить выполнение триггеров при создании заказа. По умолчанию — false

data	object	"data": {
		"endpoint": "bot",
		"status": "confirmed",
		"currency": "RUB",
		"total_price": "2500.00",
		"final_total_price": "2150.00",
		"delivery_cost": "300.00",
		"lines": [...],
		"promocodes": [...],
		"redeem_points": [...],
		"region": "region_moscow",
		"custom_fields": ,
		"tags": ["tag1", "tag2"]
		}
			Да	Объект с данными заказа

* — для идентификации профиля необходимо передать **один из параметров**: `profile_id`, `email` или `phone`

Объект data

Параметр	Тип	Пример	Обязательный	Описание
endpoint	string	"bot", "website"	**Да**	Источник заказа (точка контакта). Должен быть настроен в Маркете
status	string	"new", "confirmed", "paid"	**Да**	Статус заказа. Используются статусы, настроенные в платформе
currency	string	"RUB", "USD", "EUR"	**Да**	Валюта заказа (трехбуквенный код ISO)
total_price	string	"2500.00"	**Да**	Общая стоимость заказа до применения скидок
final_total_price	string	"2150.00"	**Да**	Итоговая стоимость заказа после применения всех скидок
delivery_cost	string	"300.00"	Нет	Стоимость доставки
create_time	string (ISO 8601)	"2026-02-18T13:01:05.141Z"	Нет	Дата и время создания заказа. Если не передано, используется текущее время сервера
region	string	"region_moscow"	Нет	Идентификатор региона для применения региональных цен и акций

lines	array	См. ниже раздел "Массив lines"	**Да**	Массив позиций заказа. Должен содержать минимум 1 элемент
promocodes	array	См. ниже раздел "Массив promocodes"	Нет	Массив промокодов, примененных к заказу
requested_promotions	array	См. ниже раздел "Массив requested_promotions"	Нет	Промоакции, примененные к данному заказу. В это поле передается структура из поля `applied_promotions` ответа метода `/preorder`
redeem_points	array	См. ниже раздел "Массив redeem_points"	**Да**, если указано `type="redeem_points"` в `requested_promotions`	Массив списаний баллов по типам валют
custom_fields	object	`{ "field1": "value1", "field2": "value2" }`	Нет	Дополнительные поля заказа
tags	array	`[ "summer_sale", "first_order" ]`	Нет	Теги для маркировки заказа

Массив lines

Массив содержит информацию о позициях заказа. Должен содержать минимум 1 позицию.

Каждая позиция вынесена в отдельный объект внутри массива со следующими параметрами:

Параметр	Тип	Пример	Обязательный	Описание
external_id	string	"line_1"	**Да**	Внешний идентификатор позиции в системе клиента
product	string	"iphone17pmb256"	**Да***	Идентификатор продукта
sku	string	"IPHONE17-256-BLACK"	**Да***	Идентификатор SKU
line_number	integer	1	Нет	Порядковый номер линии в заказе
count	integer	1	**Да**	Количество единиц товара
base_price_per_item	string	"139999.00"	**Да**	Базовая цена за единицу товара

min_price_per_item	string	"130000.00"	Нет	Минимально допустимая цена за единицу товара
final_price_per_item	string	"139500.00"	**Да**	Финальная цена за единицу после применения всех скидок
status	string	"new"	Нет	Статус позиции заказа
custom_fields	object	\{"color": "black", "size": "XL"\}	Нет	Дополнительные поля позиции
requested_promotions	array	См. ниже раздел "Массив requested_promotions"	Нет	Промоакции, примененные к данной позиции. В это поле передается структура из поля `applied_promotions` ответа метода `/preorder` для соответствующей линии заказа.

* — обязателен **один из параметров**: `product` или `sku`

Массив promocodes

Массив содержит объекты с информацией о промокодах.

Полное описание структуры промокода можно найти в этой статье.

Параметр	Тип	Пример	Обязательный	Описание
code	string	"SUMMER20"	**Да**	Код промокода
pool_id	integer	42	**Да**	Идентификатор пула промокодов

Массив redeem_points

Массив с инфомацией о списании баллов лояльности. Каждый объект описывает отдельное списание.

Полное описание структуры списания баллов можно найти в этой статье.

Параметр	Тип	Пример	Обязательный	Описание
points_currency_iid	integer	1	**Да***	Внутренний идентификатор типа баллов
points_currency_eid	string	"bonus1"	**Да***	Внешний идентификатор типа баллов
name	string	"Бонусные баллы"	Нет	Название типа баллов
amount	integer	499	**Да**	Количество баллов для списания

* — обязателен **один из параметров**: `points_currency_iid` или `points_currency_eid`

Массив requested_promotions

TIP

Массив `requested_promotions` может содержаться как в объекте `data`, так и в массиве `lines` внутри этого объекта. В первом случае в массиве будут переданы промоакции, применяемые **к продуктам**, во втором — акции, применяемые **к заказу**.

Массив применяемых к заказу промоакций. Каждый объект массива описывает отдельную промоакцию.

Параметр	Тип	Пример	Обязательный	Описание
type	string	"discount", "accrue_points", "redeem_points", "code_issue", "message"	**Да**	Тип промоакции
promotion	object	`"promotion": { "external_id": "promo_1", "name": "Summer Sale" }`	**Да**	Объект с идентификатором и названием промоакции
amount	string	"150.00"	**Да,** если `type` равен `discount`,`accrue_points` или `redeem_points`	Размер скидки или количество баллов
points_info	object	`"points_info": { "points_currency_iid": 1, "points_currency_eid": "bonus1", "name": "Бонусные баллы", "amount": "50" }`	**Да,** если `type` равен `accrue_points` или `redeem_points`	Объект с информацией о баллах

Структура объекта `points_info` выглядит следующим образом:

Параметр	Тип	Пример	Обязательный	Описание
points_currency_iid	integer	1	**Да***	Внутренний идентификатор типа баллов
points_currency_eid	string	"bonus1"	**Да***	Внешний идентификатор типа баллов
name	string	"Бонусные баллы"	Нет	Название типа баллов
amount	string	"6999"	**Да**	Количество баллов
expiration_date	string (ISO 8601)	"2027-02-18T00:00:00.000Z"	Нет	Дата сгорания начисленных баллов

* — обязателен один из параметров: `points_currency_iid` или `points_currency_eid`

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "external_id": "order_12345",
  "profile_id": "696e304f547840a7286619e0",
  "skip_triggers": false,
  "data": {
    "endpoint": "bot",
    "status": "confirmed",
    "currency": "RUB",
    "total_price": "2500.00",
    "final_total_price": "2150.00",
    "delivery_cost": "300.00",
    "create_time": "2026-02-18T13:01:05.141Z",
    "region": "region_moscow",
    "lines": [
      {
        "external_id": "line_1",
        "product": "popit",
        "base_price_per_item": "1500.00",
        "final_price_per_item": "1350.00",
        "count": 1,
        "requested_promotions": [
          {
            "type": "discount",
            "promotion": {
              "external_id": "promo_1",
              "name": "Summer Sale"
            },
            "amount": "150.00"
          },
          {
            "type": "accrue_points",
            "promotion": {
              "external_id": "bonus_promo",
              "name": "Начисление баллов"
            },
            "points_info": {
              "points_currency_iid": 1,
              "points_currency_eid": "bonus1",
              "name": "Бонусные баллы",
              "amount": "75"
            }
          }
        ]
      },
      {
        "external_id": "line_2",
        "sku": "skuf",
        "base_price_per_item": "700.00"
```

```
        "base_price_per_item": "700.00",
        "min_price_per_item": "650.00",
        "final_price_per_item": "650.00",
        "count": 2,
        "requested_promotions": [
          {
            "type": "redeem_points",
            "promotion": {
              "external_id": "redeem_promo",
              "name": "Списание баллов"
            },
            "points_info": {
              "points_currency_iid": 1,
              "points_currency_eid": "bonus1",
              "name": "Бонусные баллы",
              "amount": "50"
            }
          }
        ]
      }
    ],
    "promocodes": [
      {
        "code": "HYDO-4728-DZGU",
        "pool_id": 42
      }
    ],
    "redeem_points": [
      {
        "points_currency_eid": "bonus1",
        "amount": 100
      }
    ],
    "requested_promotions": [
      {
        "type": "code_issue",
        "promotion": {
          "external_id": "welcome_promo",
          "name": "Welcome Gift"
        }
      }
    ],
    "custom_fields": {
      "source": "telegram_bot",
      "campaign": "spring_sale"
    },
    "tags": ["premium", "first_order"]
  }
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <external_id>order_12345</external_id>
  <profile_id>696e304f547840a7286619e0</profile_id>
  <skip_triggers>false</skip_triggers>
  <data>
    <endpoint>bot</endpoint>
    <status>confirmed</status>
    <currency>RUB</currency>
    <total_price>2500.00</total_price>
    <final_total_price>2150.00</final_total_price>
    <delivery_cost>300.00</delivery_cost>
    <create_time>2026-02-18T13:01:05.141Z</create_time>
    <region>region_moscow</region>
    <lines>
      <external_id>line_1</external_id>
      <product>popit</product>
      <base_price_per_item>1500.00</base_price_per_item>
      <final_price_per_item>1350.00</final_price_per_item>
      <count>1</count>
      <requested_promotions>
        <type>discount</type>
        <promotion>
          <external_id>promo_1</external_id>
          <name>Summer Sale</name>
        </promotion>
        <amount>150.00</amount>
      </requested_promotions>
      <requested_promotions>
        <type>accrue_points</type>
        <promotion>
          <external_id>bonus_promo</external_id>
          <name>Начисление баллов</name>
        </promotion>
        <points_info>
          <points_currency_iid>1</points_currency_iid>
          <points_currency_eid>bonus1</points_currency_eid>
          <name>Бонусные баллы</name>
          <amount>75</amount>
        </points_info>
      </requested_promotions>
    </lines>
    <lines>
      <external_id>line_2</external_id>
      <sku>skuf</sku>
      <base_price_per_item>700.00</base_price_per_item>
      <min_price_per_item>650.00</min_price_per_item>
      <final_price_per_item>650.00</final_price_per_item>
      <count>2</count>
```

```
      <count>2</count>
      <requested_promotions>
        <type>redeem_points</type>
        <promotion>
          <external_id>redeem_promo</external_id>
          <name>Списание баллов</name>
        </promotion>
        <points_info>
          <points_currency_iid>1</points_currency_iid>
          <points_currency_eid>bonus1</points_currency_eid>
          <name>Бонусные баллы</name>
          <amount>50</amount>
        </points_info>
      </requested_promotions>
    </lines>
    <promocodes>
      <code>HYDO-4728-DZGU</code>
      <pool_id>42</pool_id>
    </promocodes>
    <redeem_points>
      <points_currency_eid>bonus1</points_currency_eid>
      <amount>100</amount>
    </redeem_points>
    <requested_promotions>
      <type>code_issue</type>
      <promotion>
        <external_id>welcome_promo</external_id>
        <name>Welcome Gift</name>
      </promotion>
    </requested_promotions>
  </data>
</xml>
```

Пример ответа

JSON

```
{
  "data": {
    "status": "success",
    "processing_target": "immediate",
    "external_id": "order_12345",
    "issued_promocodes": [
      {
        "codes": ["HYDO-4728-DZGU"],
        "pool": {
          "pool_id": 100,
          "name": "Welcome Promocodes"
        },
        "promotion": {
          "external_id": "welcome_promo",
          "name": "Welcome Gift"
        }
      }
    ],
    "points_changes": [
      {
        "points_info": {
          "points_currency_iid": 1,
          "points_currency_eid": "bonus1",
          "name": "Бонусные баллы",
          "amount": "21500"
        },
        "accrue_points": 75,
        "redeem_points": 150
      }
    ],
    "contents": [
      {
        "promotion": {
          "external_id": "message_promo",
          "name": "Информационное сообщение"
        },
        "contents": [
          {
            "type": "text",
            "message": "Спасибо за покупку! Вы получили 75 бонусов."
          }
        ]
      }
    ]
  },
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <status>success</status>
    <processing_target>immediate</processing_target>
    <external_id>order_12345</external_id>
    <issued_promocodes>
      <codes>HYDO-4728-DZGU</codes>
      <pool>
        <pool_id>100</pool_id>
        <name>Welcome Promocodes</name>
      </pool>
      <promotion>
        <external_id>welcome_promo</external_id>
        <name>Welcome Gift</name>
      </promotion>
    </issued_promocodes>
    <points_changes>
      <points_info>
        <points_currency_iid>1</points_currency_iid>
        <points_currency_eid>bonus1</points_currency_eid>
        <name>Бонусные баллы</name>
        <amount>21500</amount>
      </points_info>
      <accrue_points>75</accrue_points>
      <redeem_points>150</redeem_points>
    </points_changes>
    <contents>
      <promotion>
        <external_id>message_promo</external_id>
        <name>Информационное сообщение</name>
      </promotion>
      <contents>
        <type>text</type>
        <message>Спасибо за покупку! Вы получили 75 бонусов.</message>
      </contents>
    </contents>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
data	object	Данные результата подтверждения
error	int	Код ошибки (0 при успехе)
error_text	string	Текст ошибки

Параметры, возвращаемые в объекте `data`:

Параметр	Тип	Описание
status	string	Статус обработки: • `success` — заказ успешно создан • `outdated` — цена изменилась с момента предварительного расчета (нужно повторить `/preorder`)
processing_target	string	Только для статуса `success`. Значение: `immediate` или `processed`
external_id	string	Внешний идентификатор заказа (из запроса)
issued_promocodes	array	Массив выданных промокодов (для акций типа `code_issue`)
points_changes	array	Массив изменений баланса баллов по типам валют
contents	array	Массив контента для отображения (текстовые сообщения из промоакций)

Массив `issued_promocodes` состоит из объектов с информацией о **примененных промокодах**:

Параметр	Тип	Описание
codes	array	Массив выданных кодов промокодов
pool	object	Информация о пуле промокодов
pool.pool_id	integer	Идентификатор пула
pool.name	string	Название пула
promotion	object	Объект с информацией о промоакции
promotion.external_id	string	Внешний идентификатор промоакции
promotion.name	string	Название промоакции

Массив `points_changes` состоит из объектов с информацией об **изменении баланса баллов**:

Параметр	Тип	Описание
points_info	object	Объект с информацией о типе баллов
points_info.points_currency_iid	integer	Внутренний идентификатор типа баллов

Параметр	Тип	Описание
points_info.points_currency_eid	string	Внешний идентификатор типа баллов
points_info.name	string	Название типа баллов
points_info.amount	string	Текущий баланс после операции
accrue_points	integer	Количество начисленных баллов
redeem_points	integer	Количество списанных баллов

Отмена временной транзакции

Описание

Отменяет **временную** транзакцию, созданную с флагом `is_temp`. После отмены баланс активных баллов пересчитывается.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/rollback_transaction`

Параметры запроса

Должен быть указан хотя бы один из идентификаторов:

Параметр	Тип	Пример	Обязательный	Описание
transaction_id	string	“TRANS12345”	Нет	Идентификатор временной транзакции
transaction_eid	string	“TRANS12345”	Нет	Внешний идентификатор временной транзакции

Пример запроса

JSON

```
{
  "transaction_id": "TRANS12345"
}
```

XML

```
<xml>
  <transaction_id>TRANS12345</transaction_id>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": {
    "transaction_id": "TRANS12345",
    "balance": "399.50"
  }
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <transaction_id>TRANS12345</transaction_id>
    <balance>399.50</balance>
  </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
transaction_id	string	Идентификатор отменённой транзакции
balance	string	Текущий баланс активных баллов пользователя после отмены

Отмена балльной транзакции

Описание

Отменяет любую балльную транзакцию — временную или постоянную. При отмене пересчитывается текущий баланс активных баллов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/cancel_transaction`

Параметры запроса

Должен быть указан хотя бы один идентификатор:

Параметр	Тип	Пример	Обязательный	Описание
transaction_id	string	“TRANS12345”	Нет	Идентификатор транзакции
transaction_eid	string	“TRANS12345”	Нет	Внешний идентификатор транзакции

Пример запроса

JSON

```
{
  "transaction_id": "TRANS12345"
}
```

XML

```
<xml>
  <transaction_id>TRANS12345</transaction_id>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": {
    "transaction_id": "TRANS12345",
    "balance": "399.50"
  }
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <transaction_id>TRANS12345</transaction_id>
    <balance>399.50</balance>
  </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
transaction_id	string	Идентификатор отменённой транзакции
balance	string	Баланс активных баллов после отмены

Получение баланса балльного счёта

Описание

Возвращает текущий баланс балльного счёта участника программы лояльности: общее количество баллов и количество активных баллов по выбранному типу баллов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/account/balance`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
points_currency_iid	int	1	Нет	Внутренний идентификатор типа баллов
points_currency_eid	string	“starts”	Нет	Внешний идентификатор типа баллов (например, `"starts"`)
matching	string	“email”	Да	Способ поиска профиля: `email`, `phone`, `profile_id`, `custom`
field_name	string	“user_id”	Да, если `matching = custom`	Название пользовательского поля для поиска
field_value	string / int	123456	Да, если `matching = custom`	Значение пользовательского поля
email	string	“user@example.com”	Да, если `matching = email`	Email участника
phone	string	“+1234567890”	Да, если `matching = phone`	Телефон участника в международном формате
profile_id	string	“profile123”	Да, если `matching = profile_id`	Идентификатор профиля

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "points_currency_eid": "starts",
  "matching": "email",
  "email": "user@example.com"
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <points_currency_eid>starts</points_currency_eid>
  <matching>email</matching>
  <email>user@example.com</email>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": {
    "total_balance": "1000.00",
    "active_balance": "800.50"
  }
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <data>
    <total_balance>1000.00</total_balance>
    <active_balance>800.50</active_balance>
  </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
total_balance	string	Общее количество баллов на балльном счёте (в строковом формате)
active_balance	string	Количество активных баллов на балльном счёте (в строковом формате)

Регистрация участника в программе лояльности

Описание

Регистрирует участника в программе лояльности. Перед регистрацией профиль ищется в базе по заданному способу сопоставления (email, телефон, идентификатор профиля или пользовательское поле).

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/register_member`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
matching	string	“email”	Да	Способ поиска профиля: `email`, `phone`, `profile_id`, `custom`
registration_date	string	“2024-09-01T12:34:56Z”	Нет	Дата регистрации в формате RFC3339; если не указана — используется текущее время
email	string	“user@example.com”	Да, если `matching = email`	Email участника для поиска профиля
phone	string	“+1234567890”	Да, если `matching = phone`	Телефон участника в международном формате
profile_id	string	“profile123”	Да, если `matching = profile_id`	Идентификатор профиля
field_name	string	“user_id”	Да, если `matching = custom`	Название пользовательского поля для поиска
field_value	string / int	“123456”	Да, если `matching = custom`	Значение пользовательского поля

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "matching": "email",
  "email": "user@example.com",
  "registration_date": "2024-09-01T12:34:56Z"
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <matching>email</matching>
  <email>user@example.com</email>
  <registration_date>2024-09-01T12:34:56Z</registration_date>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "profile_id": "123456"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <profile_id>123456</profile_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки

Параметр	Тип	Описание
error_text	string	Текст ошибки
profile_id	string	Идентификатор профиля, зарегистрированного в программе лояльности

Удаление участника из программы лояльности

Описание

Удаляет участника из программы лояльности. Перед удалением профиль ищется по выбранному способу сопоставления: email, телефон, идентификатор профиля или пользовательское поле.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/loyalty/delete_member`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abc123def456”	Да	API-токен
loyalty_program_id	int	123	Да	Идентификатор программы лояльности
matching	string	“email”	Да	Способ поиска профиля: `email`, `phone`, `profile_id`, `custom`
email	string	“user@example.com”	Да, если `matching = email`	Email для поиска профиля
phone	string	“+1234567890”	Да, если `matching = phone`	Телефон в международном формате
profile_id	string	“profile123”	Да, если `matching = profile_id`	Идентификатор профиля
field_name	string	“user_id”	Да, если `matching = custom`	Название пользовательского поля
field_value	string / int	“123456”	Да, если `matching = custom`	Значение пользовательского поля

Пример запроса

JSON

```
{
  "token": "abc123def456",
  "loyalty_program_id": 123,
  "matching": "email",
  "email": "user@example.com"
}
```

XML

```
<xml>
  <token>abc123def456</token>
  <loyalty_program_id>123</loyalty_program_id>
  <matching>email</matching>
  <email>user@example.com</email>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation",
  "profile_id": "123456"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <profile_id>123456</profile_id>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
profile_id	string	Идентификатор удалённого профиля

Регистрация события достижения цели

Описание

Запрос регистрирует событие достижения цели. В рамках регистрации события также можно передавать дополнительную информацию о профиле.

Данные о событии достижения цели можно использовать в следующих механиках платформы:

- сегментация профилей по истории действий;
- запуск триггерной рассылки или сценария;
- движение профиля по сценарию (узлы “Условие”, “JSON-фильтр” и “Вызов API”);
- подставновка динамического контента в шаблоны сообщений.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/goals/register`

Параметры запроса

Основные параметры

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
db_id	int	1	Да	Идентификатор базы данных
pixel_id	int	1	Да	Идентификатор пикселя
goal	string	"card"	Нет	Название цели
goals	JSON array	`[ { "goal": "card", "value": 10.0 } ]`	Нет	Несколько целей со значениями
value	float	-25.5	Нет	Значение цели
date	string	"2006-01-25T15:04:05+07:00"	Нет	Дата события в формате RFC 3339

Параметры матчинга

Эти параметры используются для поиска профиля в базе. Подробнее.

Параметр	Тип	Пример	Обязательный	Описание

matching	string	"email" – поиск по email из профиля или подписок "email_profile" – по email из профиля "email_sub" – по email из подписок "phone" – по телефону из профиля или подписок "phone_sub" – по телефону из подписок "profile_id" – по идентификатору профиля (только при обновлении!) "push_sub" – по пуш подписке "custom" – по кастомному полю "custom_sub" – по подписке на кастомный канал "email_phone" – по email или телефону из профиля "email_phone_sub" – по email или телефону из подписок	Нет, если поиск по email из профиля или подписок	Режим поиска подписчика. По умолчанию - email.
email	string	"john@example.com"	"matching":"email" - по профилю и подпискам "matching":"email_profile" - только по профилю	Email-адрес
phone	string	"+79000000000"	"matching":"phone"	Номер телефона
profile_id	string	"abcdefghijklmnqrstuvwxyz"	"matching":"profile_id"	Идентификатор профиля
field_name	string	"CRM_ID"	"matching":"custom"	Название кастомного поля профиля для поиска.

field_value	int/string	"12345"	"matching":"custom"	Значение кастомного поля для поиска. Доступен поиск по сочетанию значений поля с типом "теги". Теги передаются в строке черз запятую: "тег_1, тег_2".

Дополнительные параметры

Параметр	Тип	Пример	Обязательный	Описание
pixel_data	JSON object	`{ "key":"value", "array":[1,2,3] }`	Нет	JSON-объект с дополнительными данными события цели. Объект не должен содержать вложенные объекты или массивы. Размер объекта не больше 1 KB.
skip_triggers	bool	false	Нет	Отключает настроенные на событие триггеры и сценарии. По умолчанию — false.
enrich_profile	bool	true	Нет	Событие обновляет данные профиля. Используется для обновления данных через следующие поля: browser, os, device, country, region, city, zip, time_zone.
is_test	bool	false	Нет	Событие отмечается как тестовое для аналитики. По умолчанию — false.
vendor	string	"altcraft"	Нет	Поле профиля `_vendor` на момент события
external_type	string	"event_pixel_test"	Нет	Тип внешнего идентификатора
external_event_id	string	"qwerty123"	Нет	Внешний идентификатор
altcraft_client_id	string	"9f11d32d-2935-43d0-b8df-f4c3568095b2"	Нет	Уникальный сквозной идентификатор Altcraft

HTTP-параметры

Параметр	Тип	Пример	Обязательный	Описание
referer	string	"api.example.com"	Нет	Заголовок запроса Referer
user_agent	string	"Mozilla/5.0 (Linux; Android 6.0.1; SM-G935S Build/MMB29K; wv)"	Нет	Заголовок запроса User-Agent

accept_language	string	"fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5"	Нет	Язык операционной системы устройства клиента (Заголовок запроса Accept-Language)
client_ip	string	"10.0.0.0"	Нет	IP клиента из оригинального запроса
client_ip_v4	string	"198.16.74.224"	Нет	IPv4 адрес клиента
client_ip_v6	string	"fe80::5215:5556:d75f:806b"	Нет	IPv6 адрес клиента
browser	string	"Chrome"	Нет	Браузер клиента Получить список допустимых значений можно через запрос [`/v1.1/dictionary ("browsers")`](/developer-guide/miscellaneous/dictionary)
os	string	"Ubuntu"	Нет	Операционная система клиента Получить список допустимых значений можно через запрос [`/v1.1/dictionary ("oses")`](/developer-guide/miscellaneous/dictionary)
device	string	"Desktop"	Нет	Тип клиентского устройства Получить список допустимых значений можно через запрос [`/v1.1/dictionary ("devices")`](/developer-guide/miscellaneous/dictionary)

TIP

Поля **browser**, **os** и **device** автоматически заполняются платформой на основе содержимого поля **user_agent**.

Если в одном запросе вы передаёте и user_agent, и отдельные значения для browser, os и device, то данные из user_agent имеют приоритет и перезаписывают указанные значения.

Если вы передаёте поля browser, os и device без user_agent, для обновления профиля необходимо указать `"enrich_profile": true`, иначе платформа не обновит профиль.

Параметры геолокации

Платформа автоматически определит значения, если передать `client_ip`.

Параметр	Тип	Пример	Обязательный	Описание
lat	float	55.244075	Да, если передаётся lon	Широта клиента
lon	float	55.244075	Да, если передаётся lat	Долгота клиента

country	string	"RU"	Нет	Страна клиента (ISO-код или полное название)
region	string	"Ryazan"	Нет	Регион клиента
city	string	"Ryazan"	Нет	Город клиента
address	string	"Pochtovaya street"	Нет	Адрес клиента
zip	string	"390000"	Нет	Почтовый индекс клиента
time_zone	string	"Europe/Moscow"	Нет	Часовой пояс

TIP

Платформа автоматически определяет значения полей **country**, **region**, **city**, **zip** и **time_zone** на основе IP-адреса, переданного в параметрах **client_ip**, **client_ip_v4** или **client_ip_v6**. Если в запросе указаны как IP-адрес, так и отдельные значения для этих полей, данные, определённые по IP, будут иметь приоритет и перезапишут переданные вручную значения.

Для обновления профиля с использованием значений из полей country, region, city, zip и time_zone необходимо установить параметр `"enrich_profile": true`.

UTM-параметры

Параметр	Тип	Пример	Обязательный	Описание
utm_content	string	"utm_content"	Нет	UTM параметры
utm_medium	string	"utm_medium"	Нет	
utm_source	string	"utm_source"	Нет	
utm_campaign	string	"utm_campaign"	Нет	
utm_term	string	"utm_term"	Нет	

Параметры сети

Параметр	Тип	Пример	Обязательный	Описание
net_ssid	string	"10.200.0.0"	Нет	Имя сети клиента
net_mac_addr	string	"28:87:ba:c1:0d:5e"	Нет	MAC-адрес клиента
net_gateway_ipv4	string	"10.200.0.1"	Нет	IPv4 адрес шлюза клиента
net_gateway_ipv6	string	"::ff"	Нет	IPv6 адрес шлюза клиента

Параметры рассылки

Параметр	Тип	Пример	Обязательный	Описание
send_message_id	string	"abcdefghijklmnqrstuvwxyz"	Нет	Идентификатор отправленного сообщения в платформе. Позволяет связать событие пикселя с конкретной рассылкой.

Параметры программы лояльности

Параметр	Тип	Пример	Обязательный	Описание
loyalty_id	int	5	Нет	Идентификатор программы лояльности
promo_id	string	"6285183c3f62fcd4ac5ed36d"	Нет	Идентификатор привязанного промокода

Параметры маркета

Параметр	Тип	Пример	Обязательный	Описание
endpoint_eid	string	"example_endpoint"	Нет	Внешний идентификатор точки контакта. Если для пикселя выбрана определенная база профилей, то эта база должна совпадать с базой канала продаж.
order_eid	string	"12345"	Нет	Внешний идентификатор заказа
product_eid	string	"example_product"	Да, если передается sku_eid	Внешний идентификатор продукта
sku_eid	string	"example_sku"	Нет	Внешний идентификатор SKU. При использовании этого параметра в запросе также необходимо передавать product_eid.
region_eid	string	"region_eYFe2AE6s"	Нет	Внешний идентификатор региона
count_items	int	1	Нет	Количество товара в позиции заказа
categories	array	`[ "category_1", "category_2" ]`	Нет	Категории продукта или SKU

INFO

Максимальная длина любого строкового поля — 128 символов.

Примеры запроса

Пример запроса для регистрации одной цели и её значения

JSON

```
{
    "token": "91f1dfa81c264a938b475677c60ce967",
    "db_id": 1,
    "goal": "card",
    "pixel_id": 1,
    "referer": "api.example.com",
    "value": -25.5,
    "matching":"profile_id",
    "profile_id": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>1</db_id>
    <goal>card</goal>
    <pixel_id>1</pixel_id>
    <referer>api.example.com</referer>
    <value>-25.5</value>
    <matching>profile_id</matching>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
</xml>
```

Пример запроса для регистрации нескольких целей и их значений

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "client_ip": "10.20.30.40",
    "db_id": 24,
    "goals": [
        {
            "goal": "view product",
            "value": 10.99
        },
        {
            "goal": "add to cart",
            "value": -12.34
        }
    ],
    "pixel_id": 7,
    "matching": "profile_id",
    "profile_id": "abcdefghijklmnqrstuvwxyz",
    "referer": "api.req.from.my.site"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <client_ip>10.20.30.40</client_ip>
    <db_id>24</db_id>
    <goals>
        <goal>view product</goal>
        <value>10.99</value>
    </goals>
    <goals>
        <goal>add to cart</goal>
        <value>-12.34</value>
    </goals>
    <pixel_id>7</pixel_id>
    <matching>profile_id</matching>
    <profile_id>abcdefghijklmnqrstuvwxyz</profile_id>
    <referer>api.req.from.my.site</referer>
</xml>
```

Пример запроса для передачи дополнительной информации о событии

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "db_id": 24,
    "goals": [
        {
            "goal": "cart",
            "value": 10.99
        },
        {
            "goal": "two",
            "value": -12.34
        }
    ],
    "client_ip": "192.168.0.0",
    "pixel_id": 7,
    "referer": "api.req.from.my.site",
    "accept_language": "fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5",
    "user_agent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/51.0.2704.103 Safari/537.36",
    "matching": "profile_id",
    "profile_id": "abcdefghijklmnqrstuvwxyz",
    "pixel_data": {
        "hello": "hi",
        "arr": [
            1,
            2,
            3
        ],
        "int_val": 56893,
        "bool_val": true,
        "date_val": "2023-08-12"
    },
    "skip_triggers": false,
    "enrich_profile": true,
    "is_test": true,
    "vendor": "altcraft",
    "external_type": "event_pixel_test",
    "external_event_id": "qwerty123",
    "altcraft_client_id": "9f11d32d-2935-43d0-b8df-f4c3568095b2",
    "browser": "Chrome",
    "os": "Ubuntu",
    "device": "Desktop",
    "utm_content": "utm_content",
    "utm_medium": "utm_medium",
    "utm_source": "utm_source",
    "utm_campaign": "utm_campaign",
    "utm_term": "utm_term",
    "client_ip_v4": "198.16.74.224",
    "client_ip_v6": "fe80::5215:5556:d75f:806b"
```

```
    "client_ip_v6": "fe80::3215:5556:d75f:866b",
    "net_ssid": "10.200.0.0",
    "net_mac_addr": "28:87:ba:c1:0d:5e",
    "net_gateway_ipv4": "10.200.0.1",
    "net_gateway_ipv6": "::ff",
    "loyalty_id": 5,
    "promo_id": "abcdefghijklmnqrstuvwxyz",
    "endpoint_eid": "online_store",
    "order_eid": "12345",
    "product_eid": "hd_450_bt_earphones",
    "sku_eid": "hd_450bt_earphones_black",
    "region_eid": "region_eYFe2AE6s",
    "count_items": 1,
    "categories": [
        "category_KBZ1qurFP"
    ],
    "lat": 55.244075,
    "lon": 19.379932,
    "country": "RU",
    "region": "Ryazan",
    "city": "Ryazan",
    "address": "Pochtovaya street",
    "zip": "390000",
    "time_zone": "Europe/Moscow",
    "send_message_id": "w4Wq877djSG1_2_a1_6t_2p_U_2_g.2UCSmxz7zFnuCfCMK_5KmKH9"
}
```


XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <db_id>24</db_id>
    <goals>
        <goal>cart</goal>
        <value>10.99</value>
    </goals>
    <goals>
        <goal>two</goal>
        <value>-12.34</value>
    </goals>
    <client_ip>192.168.0.0</client_ip>
    <pixel_id>7</pixel_id>
    <referer>api.req.from.my.site</referer>
    <accept_language>fr-CH, fr;q=0.9, en;q=0.8, de;q=0.7, *;q=0.5</accept_language>
    <user_agent>Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/51.0.2704.103 Safari/537.36</user_agent>
    <matching>profile_id</matching>
    <profile_id>651516181a5c52f94532fb1d</profile_id>
    <pixel_data>
        <hello>hi</hello>
        <arr>1</arr>
        <arr>2</arr>
        <arr>3</arr>
        <int_val>56893</int_val>
        <bool_val>true</bool_val>
        <date_val>2023-08-12</date_val>
    </pixel_data>
    <skip_triggers>false</skip_triggers>
    <enrich_profile>true</enrich_profile>
    <is_test>true</is_test>
    <vendor>altcraft</vendor>
    <external_type>event_pixel_test</external_type>
    <external_event_id>qwerty123</external_event_id>
    <altcraft_client_id>9f11d32d-2935-43d0-b8df-f4c3568095b2</altcraft_client_id>
    <browser>Chrome</browser>
    <os>Ubuntu</os>
    <device>Desktop</device>
    <utm_content>utm_content</utm_content>
    <utm_medium>utm_medium</utm_medium>
    <utm_source>utm_source</utm_source>
    <utm_campaign>utm_campaign</utm_campaign>
    <utm_term>utm_term</utm_term>
    <client_ip_v4>198.16.74.224</client_ip_v4>
    <client_ip_v6>fe80::5215:5556:d75f:806b</client_ip_v6>
    <net_ssid>10.200.0.0</net_ssid>
    <net_mac_addr>28:87:ba:c1:0d:5e</net_mac_addr>
    <net_gateway_ipv4>10.200.0.1</net_gateway_ipv4>
    <net_gateway_ipv6>::ff</net_gateway_ipv6>
```

```
    <net_gateway_ipv6>::ff</net_gateway_ipv6>
    <loyalty_id>5</loyalty_id>
    <promo_id>6285183c3f62fcd4ac5ed36d</promo_id>
    <endpoint_eid>online_store</endpoint_eid>
    <order_eid>12345</order_eid>
    <product_eid>hd_450_bt_earphones</product_eid>
    <sku_eid>hd_450bt_earphones_black</sku_eid>
    <region_eid>region_eYFe2AE6s</region_eid>
    <count_items>1</count_items>
    <categories>category_KBZ1qurFP</categories>
    <lat>55.244075</lat>
    <lon>19.379932</lon>
    <country>RU</country>
    <region>Ryazan</region>
    <city>Ryazan</city>
    <address>Pochtovaya street</address>
    <zip>390000</zip>
    <time_zone>Europe/Moscow</time_zone>

<send_message_id>w4Wq877djSG1_2_a1_6t_2p_U_2_g.2UCSmxz7zFnuCfCMK_5KmKH9</send_message_id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Обработка и добавление подписки

API-запрос используется для передачи push-токена из мобильного приложения в Altcraft Platform для формирования подписки у профиля.

С помощью этого запроса можно:

- Добавить новые подписки.
- Поменять статус уже существующих подписок.

Режим поиска профиля можно задавать с помощью поля `matching` в запросе (см. спецификацию ниже). Если профиль не будет найден, то будет создан новый.

При этом, если в ресурсе указано несколько баз данных, то все они будут обрабатываться по отдельности. Запрос к такому ресурсу будет создавать и обновлять профили сразу в нескольких базах. Пример такого случая можно посмотреть ниже.

Если в ресурсе нет ни одной базы, то будет возвращена ошибка.

URL-адрес

Метод: POST

`https://<cookiesaver_host>/push_subscribe`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
id	string	“abcdefghijklmnqrstuvwxyz”	Нет	Идентификатор профиля
resource_token	string	“abcdefghijklmnqrstuvwxyz”	Да	Уникальный идентификатор ресурса
db_ids	[ ] int	[2,3]	Нет	Фильтр баз данных ресурса.
matching	string	“custom”	Нет	Режим поиска подписчика. Если поле не передано или пустое, то будет использован поиск профиля по id или, если id тоже не передано, по пуш подпискам. **Для push_subscribe метода это поле может принимать только следующие значения:** • custom • profile_id

Параметр	Тип	Пример	Обязательный	Описание
subscriptions	JSON array SubscriptionObject	`[{ “provider” : “android-firebase”, “subscription_id” : “%subscriptionid%” }]`	Да	Подписки на push, которые нужно добавить в профиль или обновить. “subscription_id” — уникален для каждого устройства.

SubscriptionObject

Параметр	Тип	Пример	Обязательный	Описание
provider	string	“android-firebase”	Да	Идентификатор провайдера в Altcraft MP. Возможны следующие значения: • android-firebase • ios-firebase • ios-apns • yandex-Appmetrica-device-id-ios • yandex-Appmetrica-device-id-android • yandex-Appmetrica-ios-ifa • yandex-Appmetrica-google-aid • yandex-Appmetrica-android-token • yandex-Appmetrica-ios-token • android_huawei • ios_huawei **Каждому типу токена в AppMetrica соответствует свой id провайдера.**
subscription_id	string	“abcdef-ghijklmn-qrstuvwxyz”	Да	Токен push подписки
unsubscribe	bool	false	Нет	Флаг отписки. Отменяет подписку, если true.

Возвращаемые параметры

Параметр	Тип	Описание
data	JSON array DataObject	Информация о созданных подписках. **В массиве может быть несколько объектов в следующих случаях:** • В запросе было передано несколько SubscriptionObject • В ресурсе Altcraft MP используется несколько баз данных. В этом случае для каждой базы из ресурса добавится новый профиль с подписками.
error	int	Код ошибки
error_text	string	Текст ошибки

DataObject

Параметр	Тип	Пример	Описание
subscription_id	string	“abcdef-ghijklmn-qrstuvwxyz”	Токен push подписки
unsubscribe	bool	false	Флаг, была ли подписка добавлена или наоборот отменена
profile_id	string	“5fb62e815ae47a4aa04dfadd”	id профиля (переданный или вновь созданный), для которого добавлена подписка.
provider	string	“android-firebase”	Идентификатор провайдера в Altcraft MP. Возможны следующие значения: • android-firebase • ios-firebase • ios-apns • yandex-Appmetrica-device-id-ios • yandex-Appmetrica-device-id-android • yandex-Appmetrica-ios-ifa • yandex-Appmetrica-google-aid • yandex-Appmetrica-android-token • yandex-Appmetrica-ios-token • android_huawei • ios_huawei Каждому типу токена в AppMetrica соответствует свой id провайдера
db_id	int	2	id базы данных в Altcraft MP, в которой находится профиль
resource_id	int	1	id ресурса в Altcraft MP
subscription_hash_id	string	“92807e89”	Хеш подписки

Параметр	Тип	Пример	Описание
status	string	“error”	Статус операции • ok • error
description	string	“Duplicate push subscription”	Описание ошибки

Примеры

Запрос с матчингом по системному id профиля

Запрос

```
{
  "id": "59db33d269d426509e9c4bf9",
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "data": [
    {
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false,
      "profile_id": "59db33d269d426509e9c4bf9",
      "provider": "android-firebase",
      "db_id": 17,
      "resource_id": 5,
      "subscription_hash_id": "adsfdfdffdf",
      "status": "ok",
      "description": ""
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```

Запрос с матчингом по дополнительному полю профиля.

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "matching": "custom",
  "field_name": "CustomNumber1",
  "field_value": 123,
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "data": [
    {
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false,
      "profile_id": "59db33d269d426509e9c4bfc",
      "provider": "android-firebase",
      "db_id": 17,
      "resource_id": 5,
      "subscription_hash_id": "adsfdfdffdf",
      "status": "ok",
      "description": ""
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```

Запрос с матчингом по пуш подпискам

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "data": [
    {
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false,
      "profile_id": "59db33d269d426509e9c4bfa",
      "provider": "android-firebase",
      "db_id": 17,
      "resource_id": 5,
      "subscription_hash_id": "adsfdfdffdf",
      "status": "ok",
      "description": ""
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```

Запрос к ресурсу с несколькими базами данных

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "matching": "custom",
  "field_name": "CustomNumber1",
  "field_value": 12345,
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "error": 200,
  "error_text": "Successful operation",
  "data": [
    {
      "db_id": 34,
      "description": "",
      "profile_id": "6156ed18eb7fec10a206e916",
      "provider": "android-firebase",
      "resource_id": 22,
      "status": "ok",
      "subscription_hash_id": "340ca49c",
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false
    },
    {
      "db_id": 2,
      "description": "matching error: unknown field",
      "resource_id": 22,
      "status": "error"
    }
  ]
}
```

Запрос с матчингом по дополнительному полю профиля в определенной базе

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "db_ids": [1],
  "matching": "custom",
  "field_name": "CustomNumber1",
  "field_value": 123,
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "error": 0,
  "error_text": "Successful operation",
  "data": [
    {
      "db_id": 1,
      "description": "",
      "profile_id": "637f1a284fda7c5c08be62f",
      "provider": "android-firebase",
      "resource_id": 31,
      "status": "ok",
      "subscription_hash_id": "f8af057de",
      "subscription_id": "%subscriptionid%",
      "unsubscribe": false
    }
  ]
}
```

Запрос не обновил и не добавил ни одной подписки профиля

Запрос

```
{
  "resource_token": "3LXP2FRcRxJ-7b14db88ec937f14",
  "matching": "custom",
  "field_name": "CustomNumber1",
  "field_value": 12345,
  "subscriptions": [
    {
      "provider": "android-firebase",
      "subscription_id": "%subscriptionid%"
    }
  ]
}
```

Ответ

```
{
  "error": 200,
  "error_text": "Successful operation",
  "data": [
    {
      "db_id": 34,
      "description": "",
      "profile_id": "6156ed18eb7fec10a206e916",
      "resource_id": 22,
      "status": "ok"
    }
  ]
}
```

Добавить события с app push

Описание

Добавить события с app push.

Дополнительный API-метод для создания событий.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/pushes/post_events`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
events	int	1 — доставлен 2 — открыт 3 — доставлен и открыт	Да	События
launch_id	int	1	Да (при передаче uid в специальном формат — необязательно)	Приходит вместе с push сообщением от Altcraft в Data payload.
provider	string	“android-firebase”	Да (при передаче uid в специальном формате — необязательно)	Провайдер
subscription_id	string	“subscriptionid”	Да (при передаче uid в специальном формате — необязательно)	Идентификатор подписки
uid	string	“uid”	Да	Приходит вместе с push сообщением от Altcraft в Data payload.

CAUTION

Если `uid` имеет формат **extra (e|<launch_id>|<legacy_uid>|<provider>)**, то параметры `launch_id`, `provider`, `subscription_id` становятся необязательными, и в запросе необходимо передавать только `token`, `uid` и `events`.

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "events": 1,
  "launch_id": "launch_id",
  "provider": "android-firebase",
  "subscription_id": "subscriptionid",
  "uid": "uid"
}
```

XML

```
<xml>
 <token>abcdefghijklmnqrstuvwxyz</token>
  <events>1</events>
  <launch_id>launch_id</launch_id>
  <provider>android-firebase</provider>
  <subscription_id>subscriptionid</subscription_id>
  <uid>uid</uid>
</xml>
```

Пример запроса с uid в специальном формате

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "events": 1,
  "uid": "e|2_4S_4HFn8RDhhmA|w4HFn8RPDFUi_2_4S_S_3___6.2NvyN8Hb67wmzJshM_6BuFTt|ios-apns"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <events>1</events>
    <uid>e|2_4S_4HFn8RDhhmA|w4HFn8RPDFUi_2_4S_S_3___6.2NvyN8Hb67wmzJshM_6BuFTt|ios-
apns</uid>
</xml>
```

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Структура заказа (order data object)

Описание

Объект `order` используется в API запросах, связанных с импортом заказов в маркет.

Параметр	Тип	Пример	Обязательный	Описание
external_id	string	“123456”	Да	Идентификатор заказа
matching	string	“email”	Нет	Режим поиска подписчика. По умолчанию – email. Для каждого типа матчинга в теле запроса необходимо передавать определенные поля. Подробнее об этих полях можно узнать здесь.
endpoint	string	“website”	Да	Дополнительный идентификатор точки продажи
tags	array	`[“one”,“two”]`	Нет	Список тегов
status	string	“new” “delivered” “paid” “canceled” “part_refunded”	Да	Идентификатор статуса заказа
delivery_cost	string	“200”	Нет	Стоимость доставки
total_price	string	“3000”	Да	Общая стоимость заказа. Складывается из суммы каждой позиции в заказе и стоимости доставки.
currency	string	“USD”	Да	Валюта, в которой указана стоимость заказа. Валюты настраиваются в интерфейсе платформы: **Маркет** —> **Настройки** —> **Другое**. Если в запросе передать валюту, которая не задана в платформе, произойдет ошибка.
create_time	string	“2006-01-23T15:04:05.141Z”	Нет	Дата создания заказа (формат RFC3339)
update_time	string	“2006-01-23T15:04:05.141Z”	Нет	Дата обновления заказа (формат RFC3339)

Параметр	Тип	Пример	Обязательный	Описание
custom_fields	object	{ “order_field_id_1”: “data”, “order_field_id_2”: 123 }	Нет	Значение дополнительных полей заказа. `"[id_доп_поля]": "[значение_доп_поля]"`
region	string	“region_external_id”	Нет	Дополнительный идентификатор региона. Создать регион и получить его идентификатор можно в интерфейсе платформы: **Маркет** —> **Настройки**.
lines	array	[{ “external_id”: “7777”, “product”: “product_external_id”, “base_price_per_item”: “600”, “min_price_per_item”: “300”, “final_price_per_item”: “500”, “count”: 5 }]	Да	Данные о позициях заказа. Минимум одна позиция. См. ниже.

JSON

JSON

```
[
  {
    "external_id": "123456",
    "matching": "email",
    "email": "johndoe@example.com",
    "endpoint": "endpoint_external_id",
    "tags": [
      "one",
      "two"
    ],
    "status": "new",
    "delivery_cost": "100",
    "total_price": "1000",
    "currency": "USD",
    "custom_fields": {
      "order_field_id_1": "data",
      "order_field_id_2": 123
    },
    "region": "region_external_id",
    "lines": [
      {
        "external_id": "1234000",
        "status": "new",
        "sku": "sku_external_id",
        "base_price_per_item": "900",
        "min_price_per_item": "900",
        "final_price_per_item": "900",
        "count": 5,
        "custom_fields": {
          "line_field_id_2": "data",
          "line_field_id_1": 123
        }
      }
    ]
  }
]
```

XML

```
<xml>
    <row>
        <external_id>123456</external_id>
        <matching>email</matching>
        <email>johndoe@example.com</email>
        <endpoint>endpoint_external_id</endpoint>
        <tags>one</tags>
        <tags>two</tags>
        <status>new</status>
        <delivery_cost>100</delivery_cost>
        <total_price>1000</total_price>
        <currency>USD</currency>
        <custom_fields>
            <order_field_id_1>data</order_field_id_1>
            <order_field_id_2>123</order_field_id_2>
        </custom_fields>
        <region>region_external_id</region>
        <lines>
            <external_id>1234000</external_id>
            <status>0001</status>
            <sku>sku_external_id</sku>
            <base_price_per_item>900</base_price_per_item>
            <min_price_per_item>900</min_price_per_item>
            <final_price_per_item>900</final_price_per_item>
            <count>5</count>
            <custom_fields>
                <line_field_id_2>data</line_field_id_2>
                <line_field_id_1>123</line_field_id_1>
            </custom_fields>
        </lines>
    </row>
</xml>
```

Lines object

Описание

Массив объектов `lines` содержит объекты с данными о позициях заказа. Один объект – одна позиция.

Параметр	Тип	Пример	Обязательный	Описание
external_id	string	“1235678”	Да	Дополнительный идентификатор позиции

Параметр	Тип	Пример	Обязательный	Описание
status	string	“new”	Нет	Идентификатор статуса позиции заказа: • `delivering` • `processing` • `new` • `refunded` • `canceled` • `delivered` • идентификатор кастомного статуса
sku	string	IPhone_Red_128Gb	Не должен присутствовать в запросе, если задан `product`	Дополнительный идентификатор SKU. Если товар представлен в виде продукта — укажите ID продукта, если же товар представлен в виде его разновидности — укажите **только** ID SKU.
product	string	MP3_player_BassPro	Не должен присутствовать в запросе, если задан `sku`	Дополнительный идентификатор продукта
base_price_per_item	string	“600”	Да	Базовая цена
min_price_per_item	string	“300”	Да	Минимальная цена
final_price_per_item	string	“500”	Да	Конечная цена
count	int	5	Да	Количество товара в позиции
custom_fields	object	`{ “imported”: true, “lot_number”: “39003” }`	Нет	Значение дополнительных полей в позиции заказа

JSON

JSON

```
[
    {
        "external_id": "1235678",
        "status": "0001",
        "sku": "sku_external_id",
        "base_price_per_item": "600",
        "min_price_per_item": "300",
        "final_price_per_item": "500",
        "count": 5,
        "custom_fields":
        {
            "line_field_id_2": 123,
            "line_field_id_2": "data"
        }
    }
]
```

XML

```
<xml>
    <external_id>1235678</external_id>
    <status>0001</status>
    <sku>sku_external_id</sku>
    <base_price_per_item>600</base_price_per_item>
    <min_price_per_item>300</min_price_per_item>
    <final_price_per_item>500</final_price_per_item>
    <count>5</count>
    <custom_fields>
        <line_field_id_2>data</line_field_id_2>
    </custom_fields>
</xml>
```

Product data object

Описание

Массив объектов `product` используется в API запросах, связанных с импортом продуктов в маркет.

Параметр	Тип	Пример	Обязательный	Описание
name	string	“Canon EOS Digital Camera”	Да	Название продукта.
external_id	string	“12345”	Да	Дополнительный идентификатор продукта, который используется при добавлении позиции в заказе.
price	string	“490”	Нет	Актуальная цена продукта.
old_price	string	“520”	Нет	Старая цена продукта.
currency	string	“USD”	Нет	Валюта, в которой указана цена продукта. Валюты настраиваются в интерфейсе платформы: **Маркет** —> **Настройки** —> **Другое**. Если в запросе передать валюту, которая не задана в платформе, произойдет ошибка.
tags	array	`[ “camera”, “Canon” ]`	Нет	Список тегов.
manufacturer_name	string	“Canon”	Нет	Название производителя.
description	string	“Test description”	Нет	Описание продукта. Длина текста — не более 1000 символов.
url	string	`"https://example.com"`	Нет	URL страницы продукта на сайте магазина.
pictures	array	`[ “https://example_1.com”, “https://example_2.com” ]`	Нет	Список URL-ссылок на изображения продукта.
count	int	60	Нет	Количество продуктов, доступное для продажи.

Параметр	Тип	Пример	Обязательный	Описание
expiration_time	string	“730”	Нет	Срок годности или службы (в сутках).
is_available	bool	true	Нет	Есть ли продукт в наличии у магазина.
delivery	bool	true	Нет	Возможность курьерской доставки (по всем регионам, в которые доставляет магазин).
barcode	array	`[ 5235235, 347347, 335578769 ]`	Нет	Список штрихкодов продукта от производителя.
manufacturer_warranty	bool	true	Нет	Официальная гарантия производителя.
custom_fields	object	`{ “color”: “black”, “weight”: 701, “dimensions”: “105*141*77” }`	Нет	Дополнительные поля продукта. Подробнее.

Параметр	Тип	Пример	Обязательный	Описание

regional_data	array	[
		{
		“external_id”: “region_YnCrJj8Y1”,
		“price”: “36000”,
		“old_price”: “38500”,
		“currency”: “RUB”,
		“custom_fields”: {
		“color”: “black”,
		“weight”: 701
		},
		“count”: 30,
		“is_available”: true,
		“delivery”: true
		}
		]
			Нет	Данные о продукте, которые могут различаться в зависимости от региона.
				
				`external_id` – это дополнительный идентификатор региона. Создать регион и получить его идентификатор можно в интерфейсе платформы: **Маркет** —> **Настройки**.

categories	array	[“category_1”, “category_2”]	Нет	Дополнительный идентификатор категории продукта. Подробнее.

sku	array	[
		{
		“name”: “Canon EOS 90D Body”,
		“external_id”: “12345678”,
		}
		]
			Нет	Данные о SKU продукта. Подробнее.

JSON

JSON

```
[
  {
    "name": "iPhone",
    "external_id": "123456",
    "price": "500",
    "old_price": "400",
    "currency": "USD",
    "tags": ["one", "two"],
    "manufacturer_name": "Apple",
    "description": "Test description",
    "url": "https://example.com",
    "pictures": ["https://example.com", "https://example.com"],
    "count": 60,
    "expiration_time": "730",
    "is_available": true,
    "delivery": true,
    "barcode": [5235235, 347347, 335578769],
    "manufacturer_warranty": false,
    "custom_fields": {
      "product_field_id_1": 123,
      "product_field_id_2": "data"
    },
    "regional_data": [
      {
        "external_id": "region_YnCrJj8Y1",
        "price": "864825",
        "old_price": "400",
        "currency": "USD",
        "custom_fields": {
          "product_field_id_1": 123,
          "product_field_id_2": "data"
        },
        "count": 30,
        "is_available": true,
        "delivery": true
      }
    ],
    "categories": ["category_1", "category_2"],
    "sku": [
      {
        "name": "iPhone",
        "external_id": "12345678"
      }
    ]
  }
]
```

XML

```
<xml>
 <row>
    <name>iPhone</name>
    <external_id>12345</external_id>
    <price>235</price>
    <old_price>400</old_price>
    <currency>USD</currency>
    <description></description>
    <product_external_id>12345678</product_external_id>
    <categories>1234</categories>
    <categories>12345</categories>
    <tags>one</tags>
    <tags>two</tags>
    <manufacturer_name>Apple</manufacturer_name>
    <url>https://example.com</url>
    <pictures>https://example.com</pictures>
    <pictures>https://example.com</pictures>
    <count>50</count>
    <expiration_time>730</expiration_time>
    <is_available>true</is_available>
    <delivery>true</delivery>
    <barcode>23346346</barcode>
    <barcode>23546362</barcode>
    <barcode>76834566</barcode>
    <manufacturer_warranty>true</manufacturer_warranty>
    <custom_fields>
      <sku_custom_field_1>123</sku_custom_field_1>
      <sku_custom_field_2>data</sku_custom_field_2>
    </custom_fields>
    <regional_data>
      <external_id>region_YnCrJj8Y1</external_id>
      <price>864825</price>
      <old_price>400</old_price>
      <currency>USD</currency>
      <custom_fields>
        <sku_custom_field_1>123</sku_custom_field_1>
        <sku_custom_field_2>data</sku_custom_field_2>
      </custom_fields>
      <count>30</count>
      <is_available>true</is_available>
      <delivery>true</delivery>
    </regional_data>
  </row>
</xml>
```

Структура SKU (SKU data object)

Описание

Массив объектов `sku` используется в API запросах, связанных с импортом SKU в маркет.

Параметр	Тип	Пример	Обязательный	Описание
name	string	“Canon EOS 90D Body”	Да	Название SKU.
external_id	string	“12345678”	Да	Дополнительный идентификатор SKU, который используется при добавлении позиции в заказе.
price	string	“510”	Нет	Актуальная цена SKU.
old_price	string	“600”	Нет	Старая цена SKU.
currency	string	“USD”	Нет	Валюта, в которой указана цена SKU. Валюты настраиваются в интерфейсе платформы: **Маркет** —> **Настройки** —> **Другое**. Если в запросе передать валюту, которая не задана в платформе, произойдет ошибка.
tags	array	`[ “camera”, “Canon” ]`	Нет	Список тегов.
manufacturer_name	string	“Canon”	Нет	Название производителя.
description	string	“test description”	Нет	Описание SKU. Длина текста — не более 1000 символов.
product_external_id	string	“12345”	Да	Дополнительный идентификатор продукта, к которому привязан SKU. Должен быть одинаковым для всех SKU одного продукта.
url	string	`"https://example.com"`	Нет	URL страницы SKU на сайте магазина.
pictures	array	`[ “https://example_1.com”, “https://example_2.com” ]`	Нет	URL-ссылка на картинку SKU.
count	int	10	Нет	Количество SKU, доступное для продажи.

Параметр	Тип	Пример	Обязательный	Описание
expiration_time	string	“730”	Нет	Срок годности или службы.
is_available	bool	true	Нет	Есть ли SKU в наличии у магазина.
delivery	bool	false	Нет	Возможность курьерской доставки (по всем регионам, в которые доставляет магазин).
barcode	array	[5235235, 347347, 335578769]	Нет	Список штрихкодов SKU.
manufacturer_warranty	bool	true	Нет	Официальная гарантия производителя.

custom_fields	object	{
		“color”: “black”,
		“weight”: 730,
		“dimensions”: “105*139*79”
		}
			Нет	Дополнительные поля SKU.

Параметр	Тип	Пример	Обязательный	Описание

regional_data	array	[
		{
		“external_id”: “region_YnCrJj8Y1”,
		“price”: “36000”,
		“old_price”: “38500”,
		“currency”: “RUB”,
		“custom_fields”: {
		“color”: “black”,
		“weight”: 701
		},
		“count”: 10,
		“is_available”: true,
		“delivery”: true
		}
		]
			Нет	Данные о SKU, которые могут различаться в зависимости от региона.
				
				`external_id` – это дополнительный идентификатор региона. Создать регион и получить его идентификатор можно в интерфейсе платформы: **Маркет** —> **Настройки**.

categories	array	[“category_1”, “category_2”]	Нет	Название категории SKU.

JSON

JSON

```
[
  {
    "name": "iPhone",
    "external_id": "12345",
    "price": "235",
    "old_price": "400",
    "currency": "USD",
    "description": "",
    "product_external_id": "12345678",
    "categories": ["1234", "12345"],
    "tags": ["one", "two"],
    "manufacturer_name": "Apple",
    "url": "https://example.com",
    "pictures": ["https://example.com", "https://example.com"],
    "count": 50,
    "expiration_time": "730",
    "is_available": true,
    "delivery": true,
    "barcode": [23346346, 23546362, 76834566],
    "manufacturer_warranty": true,
    "custom_fields": {
      "sku_custom_field_1": 123,
      "sku_custom_field_2": "data"
    },
    "regional_data": [
      {
        "external_id": "region_YnCrJj8Y1",
        "price": "864825",
        "old_price": "400",
        "currency": "USD",
        "custom_fields": {
          "sku_custom_field_1": 123,
          "sku_custom_field_2": "data"
        },
        "count": 30,
        "is_available": true,
        "delivery": true
      }
    ]
  }
]
```


XML

```
<xml>
  <row>
    <name>iPhone</name>
    <external_id>12345</external_id>
    <price>235</price>
    <old_price>400</old_price>
    <currency>USD</currency>
    <description></description>
    <product_external_id>12345678</product_external_id>
    <categories>1234</categories>
    <categories>12345</categories>
    <tags>one</tags>
    <tags>two</tags>
    <manufacturer_name>Apple</manufacturer_name>
    <url>https://example.com</url>
    <pictures>https://example.com</pictures>
    <pictures>https://example.com</pictures>
    <count>50</count>
    <expiration_time>730</expiration_time>
    <is_available>true</is_available>
    <delivery>true</delivery>
    <barcode>23346346</barcode>
    <barcode>23546362</barcode>
    <barcode>76834566</barcode>
    <manufacturer_warranty>true</manufacturer_warranty>
    <custom_fields>
      <sku_custom_field_1>123</sku_custom_field_1>
      <sku_custom_field_2>data</sku_custom_field_2>
    </custom_fields>
    <regional_data>
      <external_id>region_YnCrJj8Y1</external_id>
      <price>864825</price>
      <old_price>400</old_price>
      <currency>USD</currency>
      <custom_fields>
        <sku_custom_field_1>123</sku_custom_field_1>
        <sku_custom_field_2>data</sku_custom_field_2>
      </custom_fields>
      <count>30</count>
      <is_available>true</is_available>
      <delivery>true</delivery>
    </regional_data>
  </row>
</xml>
```

Категории (categories array)

Массив `categories` используется для создания категорий и подкатегорий в маркете. Каждый объект создает одну категорию. Передается в следующих запросах:

- Импорт продуктов, SKU и категорий
- Импорт SKU и категорий

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
external_id	string	“1234”	Да	Дополнительный идентификатор категории
name	string	“Cell phones and accessories”	Да	Название поля
parent_external_id	string	“123”	Нет	Дополнительный идентификатор главной категории. Передается при создании подкатегории.

JSON

JSON

```
[
        {
                "external_id": "123",
                "name": "Test category",
        },
        {
                "external_id": "12345",
                "name": "Test subcategory",
                "parent_external_id": "123"
        }
]
```

XML

```
<row>
  <external_id>123</external_id>
  <name>Test category</name>
</row>
<row>
  <external_id>12345</external_id>
  <name>Test subcategory</name>
  <parent_external_id>123</parent_external_id>
</row>
```

Custom fields array

Custom fields array

Массив `custom_fields` используется для создания дополнительных полей у различных сущностей маркета (заказ, позиция заказа, продукт, SKU). Содержит объекты, каждый из которых создает одно дополнительное поле.

Передается в следующих API-запросах:

- Импорт продуктов, SKU и категорий
- Импорт SKU и категорий
- Импорт заказа и статусов позиций

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
external_id	string	“1234”	Да	Дополнительный идентификатор пользовательского поля
name	string	“color”	Да	Название поля
entity_type	string	“product” “sku” “order” “line_order”	Да	Тип сущности
required	bool	false	Да	Является ли поле обязательным для заполнения
value_type	string	“unicode” “int” “float” “bool” “enum” “date”	Да	Тип данных, записываемых в поле
default_value	string	“black”	Нет	Значение дополнительного поля по умолчанию
enum_values	object	{“one”: 1, “two”: 2}	Да, если `value_type==enum`	Значение каждого свойства в дополнительном поле с типом "`enum`"

JSON

JSON

```
[
  {
    "external_id": "12",
    "name": "product_custom_field",
    "entity_type": "product",
    "required": true,
    "value_type": "unicode",
    "default_value": "data"
  },
  {
    "external_id": "123",
    "name": "sku_custom_field",
    "entity_type": "sku",
    "required": false,
    "value_type": "int",
    "default_value": 100
  },
  {
    "external_id": "1234",
    "name": "order_custom_field",
    "entity_type": "order",
    "required": false,
    "value_type": "enum",
    "default_value": { "one": 1, "two": 2 }
  },
  {
    "external_id": "1234",
    "name": "order_custom_field",
    "entity_type": "line_order",
    "required": false,
    "value_type": "bool",
    "default_value": false
  }
]
```


XML

```
<xml>
  <row>
    <external_id>12</external_id>
    <name>product_custom_field</name>
    <entity_type>product</entity_type>
    <required>true</required>
    <value_type>unicode</value_type>
    <default_value>data</default_value>
  </row>
  <row>
    <external_id>123</external_id>
    <name>sku_custom_field</name>
    <entity_type>sku</entity_type>
    <required>false</required>
    <value_type>int</value_type>
    <default_value>100</default_value>
  </row>
  <row>
    <external_id>1234</external_id>
    <name>order_custom_field</name>
    <entity_type>order</entity_type>
    <required>false</required>
    <value_type>enum</value_type>
    <default_value>
      <one>1</one>
      <two>2</two>
    </default_value>
  </row>
  <row>
    <external_id>1234</external_id>
    <name>order_custom_field</name>
    <entity_type>line_order</entity_type>
    <required>false</required>
    <value_type>bool</value_type>
    <default_value>false</default_value>
  </row>
</xml>
```

Импорт заказа и статусов позиций

Описание

Импортирует в маркет заказ, метод оплаты и статусы позиций.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/orders/import`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен Убедитесь, что для токена доступны роли, позволяющие работать с маркетом.
custom_fields	array of objects	“custom_fields”: [{ “external_id”: “1234”, “name”: “1234”, “entity_type”: “line_order”, “required”: false, “value_type”: “unicode”, “default_value”: “data” }]	Нет	Создание новых дополнительных полей для сущностей: • SKU • продукт • заказ • позиция заказа Поля нужно создать до их использования в заказе.
custom_line_statuses	array of objects	“custom_line_statuses”: [{ “external_id”: “1234”, “name”: “Поступил в пункт выдачи”, “type”: “in_process” }]	Нет	Кастомный статус позиции заказа. Необходимо передать идентификатор статуса (`external_id`), его название (`name`) и тип (`type`). Возможные значения поля **`type`**: • `delivered` (доставлен) • `canceled` (отменен) • `refunded` (возврат) • `in_process` (обрабатывается)
skip_triggers	bool	true	Нет	Пропустить запуск триггеров (по умолчанию – false)

Параметр	Тип	Пример	Обязательный	Описание
data	array of objects	“data”: [{ “external_id”: “123456”, “matching”:“email”, “email":"johndoe@example.com” “endpoint”: “endpoint_external_id”, “status”: “new”, “total_price”: “3000”, “currency”: “USD”, “create_time”: “2022-01-02T15:04:05.141Z”, “lines”: [{ “external_id”: “7777”, “status”: “new”, “product”: “product_external_id”, “base_price_per_item”: “600”, “min_price_per_item”: “300”, “final_price_per_item”: “500”, “count”: 5 }] }]	Да	Данные о заказе. В этом блоке можно указать значения для ранее созданных `custom_fields`

TIP

Если необходимо задать значение для созданного дополнительного поля, передайте в объект `custom_fields`, где ключом будет `external_id` этого поля, а значением — требуемые данные.

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "custom_fields": [
    {
      "external_id": "order_field_id_123",
      "name": "Дополнительное поле заказа",
      "entity_type": "order",
      "required": false,
      "value_type": "unicode",
      "default_value": "значение тестового поля"
    },
    {
      "external_id": "line_field_id_123",
      "name": "Дополнительное поле позиции заказа",
      "entity_type": "line_order",
      "required": false,
      "value_type": "unicode",
      "default_value": "значение тестового поля"
    }
  ],
  "custom_line_statuses": [
    {
      "external_id": "status_id_123",
      "name": "Доставлен в пункт выдачи",
      "type": "in_process"
    }
  ],
  "skip_triggers": true,
  "data": [
    {
      "external_id": "1234000",
      "matching": "email",
      "email": "johndoe@example.com",
      "endpoint": "website",
      "tags": ["one", "two"],
      "status": "delivered",
      "delivery_cost": "100",
      "total_price": "1000",
      "currency": "USD",
      "create_time": "2022-01-02T15:04:05.141Z",
      "custom_fields": {
        "order_field_id_123": "data"
      },
      "region": "region_YnCrJj8Y1",
      "lines": [
        {
          "external_id": "line_01",
          "status": "status_id_123"
```

```
          "status": "status_id_123",
          "product": "iphone_test_1",
          "base_price_per_item": "900",
          "min_price_per_item": "900",
          "final_price_per_item": "900",
          "count": 1,
          "custom_fields": {
            "line_field_id_123": "data"
          }
        }
      ]
    }
  ]
}
```


XML

```
<xml>
<token>abcdefghijklmnqrstuvwxyz</token>
  <custom_fields>
    <external_id>order_field_id_123</external_id>
    <name>Дополнительное поле заказа</name>
    <entity_type>order</entity_type>
    <required>false</required>
    <value_type>unicode</value_type>
    <default_value>значение тестового поля</default_value>
  </custom_fields>
  <custom_fields>
    <external_id>line_field_id_123</external_id>
    <name>Дополнительное поле позиции заказа</name>
    <entity_type>line_order</entity_type>
    <required>false</required>
    <value_type>unicode</value_type>
    <default_value>значение тестового поля</default_value>
  </custom_fields>
  <custom_line_statuses>
    <external_id>status_id_123</external_id>
    <name>Доставлен в пункт выдачи</name>
    <type>in_process</type>
  </custom_line_statuses>
  <skip_triggers>true</skip_triggers>
  <data>
    <external_id>1234000</external_id>
    <matching>email</matching>
    <email>johndoe@example.com</email>
    <endpoint>website</endpoint>
    <tags>one</tags>
    <tags>two</tags>
    <status>delivered</status>
    <delivery_cost>100</delivery_cost>
    <total_price>1000</total_price>
    <currency>USD</currency>
    <create_time>2022-01-02T15:04:05.141Z</create_time>
    <custom_fields>
      <order_field_id_123>data</order_field_id_123>
    </custom_fields>
    <region>region_YnCrJj8Y1</region>
    <lines>
      <external_id>line_01</external_id>
      <status>status_id_123</status>
      <product>iphone_test_1</product>
      <base_price_per_item>900</base_price_per_item>
      <min_price_per_item>900</min_price_per_item>
      <final_price_per_item>900</final_price_per_item>
      <count>1</count>
      <custom_fields>
```

```
      <custom_fields>
        <line_field_id_123>data</line_field_id_123>
      </custom_fields>
    </lines>
  </data>
</xml>
```

INFO

`custom_fields` в корне запроса создаёт новые поля (массив), а в `data` — заполняет их значениями (объект “id_поля”: “значение”)

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
result	array	Идентификаторы импортированных заказов

Пример ответа

JSON

```
{
  "error": 0,
  "error_text": "",
  "result": [
    {
      "error": 0,
      "external_id": "1234000"
    }
  ]
}
```

XML

```
<xml>
    <error>0</error>
    <error_text></error_text>
    <result>
        <error>0</error>
        <external_id>1234000</external_id>
    </result>
</xml>
```

Получить список заказов

Описание

Получить список всех заказов в аккаунте или заказов по конкретной точке контакта.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/orders/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
from_id	string	“2oDantiaiPtY-b7b41276360564d4”	Нет	Указатель на следующую страницу, берется из предыдущего запроса или передается пустой для первого.
limit	int	3	Нет	Лимит количества записей.
endpoint	string	“website”	Нет	Точка контакта. Указывается, если необходимо получить список заказов по конкретной точке контакта.

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "from_id": "",
  "limit": 1,
  "endpoint": "qwerty"
}
```

XML

```
<xml>
 <token>abcdefghijklmnqrstuvwxyz</token>
  <from_id></from_id>
  <limit>1</limit>
  <endpoint>qwerty</endpoint>
</xml>
```

Пример ответа JSON

JSON

```
{
    "error": 0,
    "error_text": "",
    "next_from_id": "2p2FXnb7zzoh-26167c2af5162ca4",
    "result": [
        {
            "profile": {
                "profile_id": "61addfc13f62fc8440i8b013",
                "xxh": "636gcaf5614536asf"
            },
            "external_id": "12345678-0001",
            "endpoint": "website",
            "status": "new",
            "delivery_cost": "50000",
            "total_price": "4549000",
            "currency": "RUB",
            "custom_fields": {},
            "lines": [
                {
                    "external_id": "line_DwWuNkAYV",
                    "product": "xiaomi_11T",
                    "base_price_per_item": "4499000",
                    "min_price_per_item": "0",
                    "final_price_per_item": "4499000",
                    "count": 1,
                    "custom_fields": {},
                    "currency": "RUB"
                }
            ],
            "create_time": "2021-12-27T09:38:39Z",
            "update_time": "2021-12-27T09:38:39Z"

        }
    ]
}
```


XML

```
<xml>
    <next_from_id>2p2FXnb7zzoh-26167c2af5162ca4</next_from_id>
    <result>
        <profile>
            <profile_id>61addfc13f62fc8440i8b013</profile_id>
            <xxh>636gcaf5614536asf</xxh>
        </profile>
        <external_id>12345678-0001</external_id>
        <endpoint>website</endpoint>
        <status>new</status>
        <delivery_cost>50000</delivery_cost>
        <total_price>4549000</total_price>
        <currency>RUB</currency>
        <lines>
            <external_id>line_DwWuNkAYV</external_id>
            <product>xiaomi_11T</product>
            <base_price_per_item>4499000</base_price_per_item>
            <min_price_per_item>0</min_price_per_item>
            <final_price_per_item>4499000</final_price_per_item>
            <count>1</count>
            <currency>RUB</currency>
        </lines>
        <create_time>2021-12-27T09:38:39Z</create_time>
        <update_time>2021-12-27T09:38:39Z</update_time>
    </result>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	string	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории.
result	array	Массив полученных данных по заказам

Удалить заказ

Описание

Удалить заказ в маркете, в том числе из истории профиля (запрос полностью удаляет заказ из базы данных).

Для удаления необходимо использовать токен, для которого назначена роль с соответствующими правилами доступа.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/orders/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Нет, если передается external_id	Внутренний идентификатор заказа
external_id	string	“A598t32D”	Нет, если передается id	Внешний идентификатор заказа

Пример запроса

JSON

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "id": 1
}

{
        "token": "abcdefghijklmnqrstuvwxyz",
        "external_id": "A598t32D"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
</xml>

<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <external_id>"A598t32D"</external_id>
</xml>
```

Пример ответа JSON

JSON

```
{
    "error": 0,
    "error_text": ""
}
```

XML

```
<xml>
   <error>0</error>
   <error_text></error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Получить статус заказа

Описание

Получить информацию о статусе заказа.

Для запроса необходимо использовать токен, для которого назначена роль с соответствующими правилами доступа.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/orders/get_status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Нет, если передается external_id	Внутренний идентификатор заказа
external_id	string	“A598t32D”	Нет, если передается id	Внешний идентификатор заказа

Пример запроса

JSON

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "id": 1
}

{
        "token": "abcdefghijklmnqrstuvwxyz",
        "external_id": "A598t32D"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
</xml>

<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <external_id>"A598t32D"</external_id>
</xml>
```

Пример ответа JSON

JSON

```
{
    "error": 0,
    "error_text": "",
    "result": {
        "endpoint": "marketplace",
        "status": "delivered"
    }
}
```

XML

```
<xml>
   <error>0</error>
   <error_text></error_text>
   <result>
      <endpoint>"marketplace"</endpoint>
      <status>"delivered"</status>
   </result>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
result	array	Массив полученных данных по заказам
error	int	Код ошибки
error_text	string	Текст ошибки

Изменение статуса позиции заказа

Описание

Обновляет статус позиции заказа в Маркете.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/orders/update_order_line_status`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
order_external_id	string	“order_1”	Да	Идентификатор заказа (external_id)
order_line_external_id	string	“line_DBSrmlVYD”	Да	Идентификатор позиции заказа
status_external_id	string	“new”	Нет	Идентификатор статуса позиции заказа: • `delivering` • `processing` • `new` • `refunded` • `canceled` • `delivered` • идентификатор кастомного статуса

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "format":"json",
    "order_external_id":"order_1",
    "order_line_external_id":"line_DBSrmlVYD",
    "status_external_id":"delivered"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <format>json</format>
  <order_external_id>order_1</order_external_id>
  <order_line_external_id>line_DBSrmlVYD</order_line_external_id>
  <status_external_id>delivered</status_external_id>
</xml>
```

Пример ответа JSON

JSON

```
{
    "data": {
        "order_external_id": "order_1",
        "order_line_external_id": "line_DBSrmlVYD",
        "status_external_id": "delivered"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <data>
        <order_external_id>order_1</order_external_id>
        <order_line_external_id>line_DBSrmlVYD</order_line_external_id>
        <status_external_id>delivered</status_external_id>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
data	object	Данные об измененной позиции заказа

Импорт продуктов, SKU и категорий

Описание

Позволяет импортировать 3 типа сущностей одновременно: продукты, SKU и категории.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/products/import`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен. Убедитесь, что для токена доступны роли, позволяющие работать с маркетом.

custom_fields	array of objects	[
		{
		“external_id”: “1234”,
		“name”: “product_custom_field”,
		“entity_type”: “product”,
		“required”: false,
		“value_type”: “unicode”,
		“default_value”: “data”
		}
		]
			Нет	Создание дополнительных полей для следующих сущностей:
				• продукт
				• SKU
				• заказ
				• позиция заказа
				Подробнее

categories	array of objects	[
		{
		“external_id”: “1234”,
		“name”: “Camera & Photo”,
		“parent_external_id”: “123”
		}
		]
			Нет	Создание категорий и подкатегорий
				Подробнее

Параметр	Тип	Пример	Обязательный	Описание

data	array of objects	[
		{
		“name”: “iPhone”,
		“product_external_id”: “12345678”,
		“external_id”: “123456”
		}
		]
			Да	Данные о продуктах и их SKU

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "custom_fields": [
    {
      "external_id": "test_field_1",
      "name": "Тестовое поле продукта",
      "entity_type": "product",
      "required": false,
      "value_type": "unicode",
      "default_value": "Значение тестового поля продукта по умолчанию"
    },
    {
      "external_id": "test_field_2",
      "name": "Тестовое поле sku",
      "entity_type": "sku",
      "required": false,
      "value_type": "unicode",
      "default_value": "Значение тестового поля sku по умолчанию"
    }
  ],
  "categories": [
    {
      "external_id": "test_category_1",
      "name": "Тестовая категория"
    }
  ],
  "data": [
    {
      "name": "Apple iPhone 13",
      "external_id": "iphone_test_1",
      "price": "900",
      "old_price": "950",
      "currency": "USD",
      "tags": ["phone", "Apple"],
      "manufacturer_name": "Apple",
      "description": "Описание продукта",
      "url": "https://example.com",
      "pictures": ["https://example.com", "https://example.com"],
      "count": 20,
      "expiration_time": "730",
      "is_available": true,
      "delivery": true,
      "barcode": [5235235, 347347, 335578769],
      "manufacturer_warranty": true,
      "custom_fields": {
        "test_field_1": "тестовое поле"
      },
      "regional_data": [
        {
```

```
        {
          "external_id": "region_YnCrJj8Y1",
          "price": "65000",
          "old_price": "70000",
          "currency": "RUB",
          "count": 5,
          "is_available": true,
          "delivery": false
        }
      ],
      "categories": ["test_category_1"],
      "sku": [
        {
          "name": "Apple iPhone 13 256GB",
          "external_id": "sku_test_1",
          "price": "1000",
          "old_price": "1100",
          "currency": "USD",
          "tags": ["phone", "Apple"],
          "manufacturer_name": "Apple",
          "description": "Описание SKU",
          "url": "https://example.com",
          "count": 10,
          "expiration_time": "730",
          "is_available": true,
          "delivery": true,
          "barcode": [5235235, 347347, 335578769],
          "manufacturer_warranty": true,
          "custom_fields": {
            "test_field_2": "тестовое поле"
          },
          "regional_data": [
            {
              "external_id": "region_Un7rdJoBT",
              "price": "75000",
              "old_price": "80000",
              "currency": "RUB",
              "count": 5,
              "is_available": true,
              "delivery": true
            }
          ]
        }
      ]
    }
  ]
}
```

XML

```
<xml>
   <token>abcdefghijklmnqrstuvwxyz</token>
  <custom_fields>
    <external_id>test_field_1</external_id>
    <name>Тестовое поле продукта</name>
    <entity_type>product</entity_type>
    <required>false</required>
    <value_type>unicode</value_type>
    <default_value>Значение тестового поля продукта по умолчанию</default_value>
  </custom_fields>
  <custom_fields>
    <external_id>test_field_2</external_id>
    <name>Тестовое поле sku</name>
    <entity_type>sku</entity_type>
    <required>false</required>
    <value_type>unicode</value_type>
    <default_value>Значение тестового поля sku по умолчанию</default_value>
  </custom_fields>
  <categories>
    <external_id>test_category_1</external_id>
    <name>Тестовая категория</name>
  </categories>
  <data>
    <name>Apple iPhone 13</name>
    <external_id>iphone_test_1</external_id>
    <price>900</price>
    <old_price>950</old_price>
    <currency>USD</currency>
    <tags>phone</tags>
    <tags>Apple</tags>
    <manufacturer_name>Apple</manufacturer_name>
    <description>Описание продукта</description>
    <url>https://example.com</url>
    <pictures>https://example.com</pictures>
    <pictures>https://example.com</pictures>
    <count>20</count>
    <expiration_time>730</expiration_time>
    <is_available>true</is_available>
    <delivery>true</delivery>
    <barcode>5235235</barcode>
    <barcode>347347</barcode>
    <barcode>335578769</barcode>
    <manufacturer_warranty>true</manufacturer_warranty>
    <custom_fields>
      <test_field_1>тестовое поле</test_field_1>
    </custom_fields>
    <regional_data>
      <external_id>region_YnCrJj8Y1</external_id>
      <price>65000</price>
```

```
      <price>65000</price>
      <old_price>70000</old_price>
      <currency>RUB</currency>
      <count>5</count>
      <is_available>true</is_available>
      <delivery>false</delivery>
    </regional_data>
    <categories>test_category_1</categories>
    <sku>
      <name>Apple iPhone 13 256GB</name>
      <external_id>sku_test_1</external_id>
      <price>1000</price>
      <old_price>1100</old_price>
      <currency>USD</currency>
      <tags>phone</tags>
      <tags>Apple</tags>
      <manufacturer_name>Apple</manufacturer_name>
      <description>Описание SKU</description>
      <url>https://example.com</url>
      <count>10</count>
      <expiration_time>730</expiration_time>
      <is_available>true</is_available>
      <delivery>true</delivery>
      <barcode>5235235</barcode>
      <barcode>347347</barcode>
      <barcode>335578769</barcode>
      <manufacturer_warranty>true</manufacturer_warranty>
      <custom_fields>
        <test_field_2>тестовое поле</test_field_2>
      </custom_fields>
      <regional_data>
        <external_id>region_Un7rdJoBT</external_id>
        <price>75000</price>
        <old_price>80000</old_price>
        <currency>RUB</currency>
        <count>5</count>
        <is_available>true</is_available>
        <delivery>true</delivery>
      </regional_data>
    </sku>
  </data>
</xml>
```

Пример ответа JSON

JSON

```
{
    "error": 0,
    "error_text": "",
    "result": [
        {
            "error": 0,
            "external_id": "iphone_test_1"
        }
    ]
}
```

XML

```
<xml>
    <error>0</error>
    <error_text></error_text>
    <result>
        <error>0</error>
        <external_id>iphone_test_1</external_id>
    </result>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
result	array	Идентификаторы продуктов, которые были успешно импортированы

Получение списка продуктов

Описание

Позволяет получить список продуктов и их SKU.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/products/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
from_id	string	“2oDantiaiPtY-b7b41276360564d4”	Нет	Указатель на следующую страницу, берется из предыдущего запроса или передается пустой для первого.
limit	int	3	Нет	Лимит количества записей.

Пример запроса

JSON

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "from_id": "0014a2feb331ace2.20200124134127",
        "limit": 3
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <from_id>0014a2feb331ace2.20200124134127</from_id>
    <limit>3</limit>
</xml>
```

Пример ответа JSON

JSON


```
{
    "error": 0,
    "error_text": "",
    "result": [
        {
            "name": "Xiaomi 11T",
            "external_id": "xiaomi_11T",
            "sku": [
                {
                    "name": "Xiaomi 11T 256ГБ",
                    "external_id": "xiaomi_11t_256gb"
                },
                {
                    "name": "Xiaomi 11T 128 ГБ",
                    "external_id": "xiaomi_11t_128gb"
                }
            ]
        },
        {
            "name": "Apple iPhone 13",
            "external_id": "iphone_13",
            "sku": [
                {
                    "name": "Apple iPhone 13 256GB",
                    "external_id": "iphone_13_256gb"
                },
                {
                    "name": "Apple iPhone 13 128GB",
                    "external_id": "iphone_13_128gb"
                }
            ]
        },
        {
            "name": "Wireless headphones Xiaomi Mi True Pro",
            "external_id": "xiaomi_mi_true_pro"
        }
    ]
}
```


XML

```
<xml>
    <error>0</error>
    <error_text></error_text>
    <result>
        <name>Xiaomi 11T</name>
        <external_id>xiaomi_11T</external_id>
        <sku>
            <name>Xiaomi 11T 256ГБ</name>
            <external_id>xiaomi_11t_256gb</external_id>
        </sku>
        <sku>
            <name>Xiaomi 11T 128 ГБ</name>
            <external_id>xiaomi_11t_128gb</external_id>
        </sku>
    </result>
    <result>
        <name>Apple iPhone 13</name>
        <external_id>iphone_13</external_id>
        <sku>
            <name>Apple iPhone 13 256GB</name>
            <external_id>iphone_13_256gb</external_id>
        </sku>
        <sku>
            <name>Apple iPhone 13 128GB</name>
            <external_id>iphone_13_128gb</external_id>
        </sku>
    </result>
    <result>
        <name>Wireless headphones Xiaomi Mi True Pro</name>
        <external_id>xiaomi_mi_true_pro</external_id>
    </result>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	string	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории.

Параметр	Тип	Описание
result	array	Массив полученных данных по продуктам и их SKU

Получение списка SKU

Описание

Позволяет получить данные о SKU.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/sku/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
from_id	string	“2oDantiaiPtY-b7b41276360564d4”	Нет	Указатель на следующую страницу, берется из предыдущего запроса или передается пустой для первого.
limit	int	3	Нет	Лимит количества записей.

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz",
  "from_id": "",
  "limit": 3
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <from_id></from_id>
    <limit>3</limit>
</xml>
```

Пример ответа JSON

JSON


```
{
    "error": 0,
    "error_text": "",
    "next_from_id": "L8PSdcsMmLq-6021b5621680598b",
    "result": [
        {
            "name": "Xiaomi 11T 256ГБ",
            "external_id": "xiaomi_11t_256gb"
        },
        {
            "name": "Xiaomi 11T 128 ГБ",
            "external_id": "xiaomi_11t_128gb"
        },
                {
            "name": "Apple iPhone 13 256GB",
            "external_id": "iphone_13_256gb"
        }
    ]
}
```


XML


```
<xml>
    <error>0</error>
    <error_text></error_text>
    <next_from_id>L8PSdcsMmLq-6021b5621680598b</next_from_id>
    <result>
        <name>Xiaomi 11T 256ГБ</name>
        <external_id>xiaomi_11t_256gb</external_id>
    </result>
    <result>
        <name>Xiaomi 11T 128 ГБ</name>
        <external_id>xiaomi_11t_128gb</external_id>
    </result>
    <result>
        <name>Apple iPhone 13 256GB</name>
        <external_id>iphone_13_256gb</external_id>
    </result>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	string	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории.
result	array	Массив полученных данных по SKU

Импорт SKU и категорий

Описание

Позволяет импортировать 2 типа сущностей одновременно: SKU и категории.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/sku/import`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен

custom_fields	array of objects	[
		{
		“external_id”: “1234”,
		“name”: “1234”,
		“entity_type”: “sku”,
		“required”: false,
		“value_type”: “unicode”,
		“default_value”: “data”
		}
		]
			Нет	Создание дополнительных полей для следующих сущностей:
				• продукт
				• SKU
				• заказ
				• позиция заказа

categories	array of objects	[
		{
		“external_id”: “1234”,
		“name”: “1234”,
		“parent_external_id”: “123”
		}
		]
			Нет	Создание категорий и подкатегорий

data	array of objects	[
		{
		“name”: “iPhone”,
		“external_id”: “123456”
		}
		]
			Да	Данные о SKU
				Подробнее

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "custom_fields": [
        {
            "external_id": "test_field_string",
            "name": "Строковое поле",
            "entity_type": "sku",
            "required": false,
            "value_type": "unicode",
            "default_value": "Значение тестового поля sku по умолчанию"
        }
    ],
    "categories": [
        {
            "external_id": "test_category_2",
            "name": "Тестовая категория 2"
        }
    ],
    "data": [
        {
            "name": "Apple iPhone 13 256GB",
            "external_id": "sku_test_1",
            "price": "1000",
            "old_price": "1100",
            "currency": "USD",
            "description": "Описание SKU",
            "product_external_id": "iphone_test_1",
            "categories": [
                "test_category_2"
            ],
            "tags": [
                "phone",
                "Apple"
            ],
            "manufacturer_name": "Apple",
            "url": "https://example.com",
            "pictures": [
                "https://example.com",
                "https://example.com"
            ],
            "count": 10,
            "expiration_time": "730",
            "is_available": true,
            "delivery": true,
            "barcode": [
                23346346,
                23546362,
                76834566
            ]
```

```
            ],
            "manufacturer_warranty": true,
            "custom_fields": {
                "test_field_string": "Тестовое поле"
            },
            "regional_data": [
                {
                    "external_id": "region_YnCrJj8Y1",
                    "price": "75000",
                    "old_price": "80000",
                    "currency": "RUB",
                    "custom_fields": {
                        "test_field_string": "Тестовое поле"
                    },
                    "count": 8,
                    "is_available": true,
                    "delivery": true
                }
            ]
        }
    ]
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <custom_fields>
    <external_id>test_field_string</external_id>
    <name>Строковое поле</name>
    <entity_type>sku</entity_type>
    <required>false</required>
    <value_type>unicode</value_type>
    <default_value>Значение тестового поля sku по умолчанию</default_value>
  </custom_fields>
  <categories>
    <external_id>test_category_2</external_id>
    <name>Тестовая категория 2</name>
  </categories>
  <data>
    <name>Apple iPhone 13 256GB</name>
    <external_id>sku_test_1</external_id>
    <price>1000</price>
    <old_price>1100</old_price>
    <currency>USD</currency>
    <description>Описание SKU</description>
    <product_external_id>iphone_test_1</product_external_id>
    <categories>test_category_2</categories>
    <tags>phone</tags>
    <tags>Apple</tags>
    <manufacturer_name>Apple</manufacturer_name>
    <url>https://example.com</url>
    <pictures>https://example.com</pictures>
    <pictures>https://example.com</pictures>
    <count>10</count>
    <expiration_time>730</expiration_time>
    <is_available>true</is_available>
    <delivery>true</delivery>
    <barcode>23346346</barcode>
    <barcode>23546362</barcode>
    <barcode>76834566</barcode>
    <manufacturer_warranty>true</manufacturer_warranty>
    <custom_fields>
      <test_field_string>Тестовое поле</test_field_string>
    </custom_fields>
    <regional_data>
      <external_id>region_YnCrJj8Y1</external_id>
      <price>75000</price>
      <old_price>80000</old_price>
      <currency>RUB</currency>
      <custom_fields>
        <test_field_string>Тестовое поле</test_field_string>
      </custom_fields>
```

```
      </custom_fields>
      <count>8</count>
      <is_available>true</is_available>
      <delivery>true</delivery>
    </regional_data>
  </data>
</xml>
```

Пример ответа JSON

JSON

```
{
  "error": 0,
  "error_text": "",
  "result": [
    {
      "error": 0,
      "external_id": "sku_test_1"
    }
  ]
}
```

XML

```
<xml>
    <error>0</error>
    <error_text></error_text>
    <result>
        <error>0</error>
        <external_id>sku_test_1</external_id>
    </result>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
result	array	Идентификаторы SKU, которые были успешно импортированы

Удалить продукты

Описание

Удалить продукты в маркете. Метод поддерживает как одиночное удаление, так и массовое удаление нескольких продуктов за один запрос.

Для удаления необходимо использовать токен, для которого назначена **роль с соответствующими правилами доступа**.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/products/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
ids	array of integers	[1, 2, 3]	Нет, если передается `external_ids`	Массив внутренних идентификаторов продуктов
external_ids	array of strings	[“product_1”, “product_2”]	Нет, если передается `ids`	Массив внешних идентификаторов продуктов

Пример запроса

Одиночный JSON

Идентификация по парметру: `"ids"`:

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "ids": [1]
}
```

Идентификация по парметру: `"external_ids"`:

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "external_ids": ["sku_123"]
}
```

Батч JSON

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "ids": [1, 2],
        "external_ids": ["sku_3"]
}
```

Одиночный XML

Идентификация по парметру: `"ids"`:

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <ids>1</ids>
</xml>
```

Идентификация по парметру: `"external_ids"`:

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <external_ids>product_123</external_ids>
</xml>
```

Батч XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <ids>1</ids>
  <ids>2</ids>
  <external_ids>product_3</external_ids>
</xml>
```

Пример ответа

JSON

```
{
    "deleted_count": 2,
    "error": 0,
    "error_text": ""
}
```

XML

```
<xml>
   <deleted_count>2</deleted_count>
   <error>0</error>
   <error_text></error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
deleted_count	int	Количество фактически удаленных продуктов
error	int	Код ошибки
error_text	string	Текст ошибки

Удалить SKU

Описание

Удалить SKU (товарные позиции) в маркете. Метод поддерживает как одиночное удаление, так и массовое удаление нескольких SKU за один запрос.

Для удаления необходимо использовать токен, для которого назначена **роль с соответствующими правилами доступа**.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/market/sku/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
ids	array of integers	[1, 2, 3]	Нет, если передается `external_ids`	Массив внутренних идентификаторов SKU
external_ids	array of strings	[“sku_1”, “sku_2”]	Нет, если передается `ids`	Массив внешних идентификаторов SKU

Пример запроса

Одиночный JSON

Идентификация по парметру: `"ids"`:

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "ids": [1]
}
```

Идентификация по парметру: `"external_ids"`:

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "external_ids": ["sku_123"]
}
```

Батч JSON

```
{
        "token": "abcdefghijklmnqrstuvwxyz",
        "ids": [1, 2],
        "external_ids": ["sku_3"]
}
```

Одиночный XML

Идентификация по парметру: `"ids"`:

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <ids>1</ids>
</xml>
```

Идентификация по парметру: `"external_ids"`:

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <external_ids>product_123</external_ids>
</xml>
```

Батч XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <ids>1</ids>
  <ids>2</ids>
  <external_ids>sku_3</external_ids>
</xml>
```

Пример ответа

JSON

```
{
    "deleted_count": 2,
    "error": 0,
    "error_text": ""
}
```

XML

```
<xml>
   <deleted_count>2</deleted_count>
   <error>0</error>
   <error_text></error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
deleted_count	int	Количество фактически удаленных SKU
error	int	Код ошибки
error_text	string	Текст ошибки

Получить сводный отчет

Описание

Получить сводный отчет.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/reports/summary`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – `"json"`)
resource_id	int	1	Нет	Идентификатор ресурса
date_from	string	“2018-08-10”	Да	Начало временного диапазона
date_to	string	“2018-08-15”	Да	Конец временного диапазона
data_type	string	“segment”	Да	Источник подписчиков для отчета
data_id	int/string	1	Нет	Идентификатор источника данных для отчета
data_group	string	“day”	Нет	Параметр группировки (“day” , “week” или “month”)
show_deleted	bool	false	Нет	Показывать ли статистику удаленных объектов
is_test_data	bool	false	Нет	Показывать статистику настоящих отправок/тестовых отправок и отправок на базы наблюдателей

Список объектов для отчета:

data_type	Описание	Можно передать параметр “data_id”	Тип параметра “data_id”
campaign	По рассылкам	Да	int
list	По базам	Да	int

data_type	Описание	Можно передать параметр “data_id”	Тип параметра “data_id”
segment	По сегментам	Да	int
message	По сообщениям	Да	int
sender	По сендерам	Да	int
domaingroup	По группе доменов	Да	string
domain	По доменам	Нет	—
vendor	По идентификатору формы импорта	Нет	—
day	Ежедневный отчет	Нет	—
week	Еженедельный отчет	Нет	—
month	Ежемесячный отчет	Нет	—
environment	По мобильному или десктопному трафику	Нет	—

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "data_group": "vendor",
    "data_id": 7,
    "data_type": "list",
    "date_from": "2024-04-04",
    "date_to": "2024-05-05",
    "format": "json"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <data_group>vendor</data_group>
  <data_id>7</data_id>
  <data_type>list</data_type>
  <date_from>2024-04-04</date_from>
  <date_to>2024-05-05</date_to>
  <format>xml</format>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "clicks": 22,
            "complains": 2,
            "confirms": 7,
            "delivs": 22,
            "hbounces": 3,
            "id": "Vendor #2",
            "opens": 44,
            "reads": 2,
            "sbounces": 15,
            "sends": 41,
            "suppresses": 1,
            "uclicks": 15,
            "unsubs": 3,
            "uopens": 22
        },
        {
            "clicks": 34,
            "complains": 4,
            "confirms": 13,
            "delivs": 24,
            "hbounces": 7,
            "id": "Vendor #1",
            "opens": 48,
            "reads": 3,
            "sbounces": 21,
            "sends": 31,
            "suppresses": 3,
            "uclicks": 21,
            "unsubs": 6,
            "uopens": 24
        }
    ],
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
  <data>
    <clicks>22</clicks>
    <complains>2</complains>
    <confirms>7</confirms>
    <confirmsubs>0</confirmsubs>
    <delivs>22</delivs>
    <hbounces>3</hbounces>
    <id>Vendor #2</id>
    <opens>44</opens>
    <reads>2</reads>
    <sbounces>15</sbounces>
    <sends>41</sends>
    <suppresses>1</suppresses>
    <uclicks>15</uclicks>
    <unsubs>3</unsubs>
    <uopens>22</uopens>
  </data>
  <data>
    <clicks>34</clicks>
    <complains>4</complains>
    <confirms>13</confirms>
    <confirmsubs>0</confirmsubs>
    <delivs>24</delivs>
    <hbounces>7</hbounces>
    <id>Vendor #1</id>
    <opens>48</opens>
    <reads>3</reads>
    <sbounces>21</sbounces>
    <sends>31</sends>
    <suppresses>3</suppresses>
    <uclicks>21</uclicks>
    <unsubs>6</unsubs>
    <uopens>24</uopens>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	string	Идентификатор источника данных

Параметр	Тип	Описание
clicks	int	Количество зафиксированных кликов
complains	int	Количество жалоб на спам (FBL)
opens	int	Количество зафиксированных открытий
reads	int	Количество прочитанных сообщений
replies	int	Количество ответов
sends	int	Количество отправленных сообщений
confirms	int	Количество подтвержденных сообщений
delivs	int	Количество доставленных сообщений
undelivs	int	Количество недоставленных сообщений
unsubs	int	Количество отписавшихся
uopens	int	Количество уникальных открытий
ureads	int	Уникальных прочтений
offence	int	Количество нарушений коммуникационной политики
glanceds	int	Количество просмотренных сообщений
hbounces	int	Hard bounces — некорректные адреса email
sbounces	int	Soft bounces — временные проблемы с доставкой
suppresses	int	Количество заблокированных стоп-списком сообщений
uclicks	int	Количество уникальных кликов
ulclicks	int	Количество уникальных кликов по ссылкам отписки
sends_push	int	Количество отправленных push
clicks_push	int	Количество кликов в push
uclicks_push	int	Уникальные клики в push
delivs_push	int	Количество доставленных push
ureads_push	int	Уникальные прочтения push
opens_push	int	Количество зафиксированных открытий push
uopens_push	int	Количество уникальных открытий push
push_ctr	int	Отношение кликов по ссылке ко всем отправленым push
sbounce_push	int	Soft bounce в push
hbounce_push	int	Hard bounce в push
sends_sms	int	Количество отправленных SMS

Параметр	Тип	Описание
clicks_sms	int	Количество кликов в SMS
uclicks_sms	int	Уникальные клики в SMS
delivs_sms	int	Количество доставленных SMS
undelivs_sms	int	Количество недоставленных SMS
sms_ctr	int	Отношение кликов по ссылке ко всем отправленым SMS
unsubs_sms	int	Отписок в SMS
open_rate	int	Процент открытий
push_deliv_rate	int	Процент доставок push
u_read_rate	int	Процент уникальных прочтений
u_open_click_rate	int	Соотношение уникальных открытий к уникальным кликам
sb_rate	int	Процент Soft bounce
email_read_rate	int	Процент прочтений email
delivery_rate	int	Процент доставок
email_unsub_rate	int	Процент отписок в email
email_u_read_rate	int	Процент уникальных прочтений email
open_click_rate	int	Соотношение открытий писем к кликам
suppresses_rate	int	Процент заблокированных стоп-списком сообщений
unsub_rate	int	Процент отписок
u_click_rate	int	Процент уникальных кликов
click_rate	int	Процент кликов
sms_deliv_rate	int	Процент доставок SMS
hb_rate	int	Процент Hard bounce
push_open_rate	int	Процент открытий push
undelivs_push	int	Недоставки push
complain_rate	int	Процент жалоб
u_open_rate	int	Процент уникальных открытий
email_reply_rate	int	Процентов ответов в email
sends_total	int	Всего отправлено
delivs_total	int	Всего доставлено
undelivs_total	int	Всего недоставок

Параметр	Тип	Описание
reads_total	int	Всего прочтений
clicks_total	int	Всего кликов
uclicks_total	int	Всего уникальных кликов
opens_total	int	Всего открытий
ureads_total	int	Всего уникальных прочтений
uopens_total	int	Всего уникальных открытий

Получить отчет о возвратах

Описание

Получить отчет о возвратах писем от email-провайдеров.

Провайдер выражает сигнал о возврате письма, отвечая на него ошибкой с кодом от 400 до 499.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/reports/bounces`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
date_from	string	“2018-08-10”	Да	Начало временного диапазона
date_to	string	“2018-08-15”	Да	Конец временного диапазона
campaign_id	int	1	Нет	Идентификатор рассылки
group_by	string	“from_domain”	Нет	Группировать по …
top	int	2	Нет	Выбрать X наиболее частых возвратов

Возможные значения параметра `group_by`:

- `ip` – группировать по IP-адресу сендера
- `from_domain` – группировать по from домену

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "campaign_id": 116,
    "date_from": "2024-04-26",
    "date_to": "2024-05-01",
    "format": "json",
    "top": 2
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <campaign_id>116</campaign_id>
    <date_from>2024-04-26</date_from>
    <date_to>2024-05-01</date_to>
    <format>xml</format>
    <top>2</top>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "data_id": "",
            "count": 3,
            "bounce_text": "Your message was not delivered because the other user mailbox
was not available",
            "to_domain": "example.com"
        },
        {
            "data_id": "",
            "count": 3,
            "bounce_text": "This response is sent when the message simply failed. Often
times this is not caused by you, but rather because of a far-end server problem",
            "to_domain": "jetwire.name"
        }
    ],
    "error": 0,
    "error_text": "Successful operation"
}
```


XML


```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <data>
        <data_id></data_id>
        <count>3</count>
        <bounce_text>Your message was not delivered because the other user mailbox was not
available</bounce_text>
        <to_domain>example.com</to_domain>
    </data>
    <data>
        <data_id></data_id>
        <count>3</count>
        <bounce_text>This response is sent when the message simply failed. Often times this
is not caused by you, but rather because of a far-end server problem</bounce_text>
        <to_domain>jetwire.name</to_domain>
    </data>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
bounce_text	string	Ответ почтового сервера
count	int	Количество ответов почтового сервера
data_id	string	Идентификатор данных
to_domain	string	Домен

Получить отчет о недоставках

Описание

Получить отчет о недоставках к email-провайдерам.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/reports/undelivered`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
date_from	string	“2018-08-10”	Да	Начало временного диапазона
date_to	string	“2018-08-15”	Да	Конец временного диапазона
campaign_id	int	1	Нет	Идентификатор рассылки
group_by	string	“from_domain”	Нет	Группировать по …
top	int	2	Нет	Выбрать X наиболее частых возвратов

Возможные значения параметра `group_by`:

- `ip` – группировать по IP-адресу сендера
- `from_domain` – группировать по from домену

Пример запроса

JSON

```
{
    "campaign_id": 116,
    "token": "abcdefghijklmnqrstuvwxyz",
    "date_from": "2024-04-26",
    "date_to": "2024-05-01",
    "format": "json",
    "top": 2
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <campaign_id>116</campaign_id>
    <date_from>2024-04-26</date_from>
    <date_to>2024-05-01</date_to>
    <format>json</format>
    <top>2</top>
</xml>
```

Пример ответа

JSON

```
{
    "data": [
        {
            "data_id": "",
            "count": 3,
            "bounce_text": "This response is sent when the message simply failed. Often
times this is not caused by you, but rather because of a far-end server problem",
            "to_domain": "example.com"
        },
        {
            "data_id": "",
            "count": 3,
            "bounce_text": "This response is sent when the message simply failed. Often
times this is not caused by you, but rather because of a far-end server problem",
            "to_domain": "jetwire.name"
        }
    ],
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <data>
        <data_id></data_id>
        <count>3</count>
        <bounce_text>This response is sent when the message simply failed. Often times this
is not caused by you, but rather because of a far-end server problem</bounce_text>
        <to_domain>example.com</to_domain>
    </data>
    <data>
        <data_id></data_id>
        <count>3</count>
        <bounce_text>This response is sent when the message simply failed. Often times this
is not caused by you, but rather because of a far-end server problem</bounce_text>
        <to_domain>jetwire.name</to_domain>
    </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
bounce_text	string	Ответ почтового сервера
count	int	Количество ответов почтового сервера
data_id	string	Идентификатор данных
to_domain	string	Домен

Получить список сендеров

Описание

Получить список сендеров, доступных аккаунту.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/senders/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
from_id	int	2	Нет	Идентификатор первого сендера в списке (по умолчанию – 1)
limit	int	10	Нет	Лимит на количество возвращаемых данных (по умолчанию – 1000)
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "limit": 2
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <limit>2</limit>
</xml>
```

Пример ответа

JSON


```
{
    "data": [
        {
            "id": 3,
            "name": "Email sender",
            "type": "akmta",
            "channel": "email"
        },
        {
            "channel": "sms",
            "id": 2,
            "name": "Bulk SMS sender",
            "type": "bulksms_com"
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "next_from_id": 5,
    "total_count": 4
}
```


XML


```
<xml>
    <data>
        <channel>email</channel>
        <id>3</id>
        <name>Email sender</name>
        <type>akmta</type>
    </data>
    <data>
        <channel>sms</channel>
        <id>2</id>
        <name>Bulk SMS sender</name>
        <type>bulksms_com</type>
    </data>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <next_from_id>5</next_from_id>
    <total_count>5</total_count>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
next_from_id	int	Указатель необходимо передать в следующем запросе для получения еще одной порции данных, до тех пор пока вывод результата не будет содержать 0 строчек истории.
total_count	int	Общее количество сендеров
id	int	Идентификатор сендера
name	string	Имя сендера
type	string	Тип сендера
channel	string	Канал

Получить список виртуальных сендеров

Описание

Получить список виртуальных сендеров.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/vsenders/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
from_id	int	1	Нет	Идентификатор первого сендера в списке (по умолчанию – 1)
limit	int	2	Нет	Лимит на количество возвращаемых данных (по умолчанию – 1000)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "limit": 2
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <limit>2</limit>
</xml>
```

Пример ответа

JSON


```
{
    "data": [
        {
            "id": 1,
            "name": "Virtual sender12",
            "sms_channel": false,
            "email_channel": true,
            "custom_channels": {}
        },
        {
            "id": 2,
            "name": "generator",
            "sms_channel": true,
            "email_channel": true,
            "custom_channels": {}
        }
    ],
    "error": 0,
    "error_text": "Successful operation",
    "total_count": 5
}
```


XML


```
<xml>
<data>
    <email_channel>true</email_channel>
    <id>1</id>
    <name>AKMTA virtual sender</name>
    <sms_channel>false</sms_channel>
  </data>
  <data>
    <email_channel>false</email_channel>
    <id>2</id>
    <name>Bulk SMS virtual sender</name>
    <sms_channel>true</sms_channel>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
  <total_count>2</total_count>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
name	string	Название виртуального сендера
sms_channel	bool	true, если есть SMS-канал
email_channel	bool	true, если есть email-канал

Получить информацию о виртуальном сендере

Описание

Получить информацию о виртуальном сендере.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/vsenders/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	Идентификатор виртуального сендера
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 1
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <id>2</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "id": 1,
        "name": "Virtual sender",
        "email_rules": [
            {
                "type": "default",
                "value": "",
                "senders": [
                    {
                        "from_email": "",
                        "replyto_email": "",
                        "from_name": "",
                        "sender_id": 3,
                        "percent": 100
                    }
                ]
            }
        ],
        "sms_rules": [],
        "custom_channels": {}
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML

```
<xml>
    <data>
    <id>1</id>
    <name>Virtual sender</name>
    <email_rules>
      <type>default</type>
      <value></value>
      <senders>
        <from_email></from_email>
        <replyto_email></replyto_email>
        <from_name></from_name>
        <sender_id>3</sender_id>
        <percent>100</percent>
      </senders>
    </email_rules>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	Идентификатор виртуального сендера
name	string	Название виртуального сендера
email_rules	JSON array	Email-правила
sms_rules	JSON array	SMS-правила

Клонировать виртуальный сендер

Описание

Клонировать виртуальный сендер.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/vsenders/clone`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор виртуального сендера
name	string	“Clone of …”	Нет	Название для нового виртуального сендера (по умолчанию – “Clone of …”)

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "name": "Copy of virtual sender #99",
    "id": 3
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <name>Copy of virtual sender #99</name>
  <id>3</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 6
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <data>
        <saved_id>7</saved_id>
    </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор нового виртуального сендера

Добавить виртуальный сендер

Описание

Добавить виртуальный сендер.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/vsenders/add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
name	string	“New virtual sender”	Да	Название виртуального сендера
email_rules	JSON array	`“email_rules”: []`	Да	Правила для email сендеров
sms_rules	JSON array	`“sms_rules”: []`	Да	Правила для SMS сендеров

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "name": "New virtual sender",
    "email_rules": [
        {
            "senders": [
                {
                    "from_email": "",
                    "from_name": "",
                    "percent": 100,
                    "replyto_email": "",
                    "sender_id": 3
                }
            ],
            "type": "default",
            "value": ""
        }
    ],

    "sms_rules": []
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <name>New virtual sender</name>
    <email_rules array ='true'>
        <senders array = 'true'>
            <percent>100</percent>
            <sender_id>3</sender_id>
        </senders>
        <type>default</type>
    </email_rules>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 8
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
    <error>0</error>
    <error_text>Successful operation</error_text>
    <data>
        <saved_id>9</saved_id>
    </data>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор нового виртуального сендера

Обновить виртуальный сендер

Описание

Обновить виртуальный сендер.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/vsenders/update`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор виртуального сендера
name	string	“New virtual sender”	Да	Название виртуального сендера
email_rules	JSON array	`“email_rules”: []`	Да	Правила для email сендеров
sms_rules	JSON array	`“sms_rules”: []`	Да	Правила для SMS сендеров

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "name": "Virtual sender",
    "email_rules": [
        {
            "senders": [
                {
                    "from_email": "",
                    "from_name": "",
                    "percent": 100,
                    "replyto_email": "",
                    "sender_id": 3
                }
            ],
            "type": "default",
            "value": ""
        }
    ],
    "id": 9,
    "sms_rules": []
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <name>Virtual sender</name>
  <email_rules array='true'>
    <senders array='true'>
      <from_email></from_email>
      <from_name></from_name>
      <percent>100</percent>
      <replyto_email></replyto_email>
      <sender_id>3</sender_id>
    </senders>
    <type>default</type>
    <value></value>
  </email_rules>
  <id>9</id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "saved_id": 1
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <saved_id>1</saved_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор виртуального сендера

Удалить виртуальный сендер

Описание

Удалить виртуальный сендер.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/vsenders/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
id	int	1	Да	Идентификатор виртуального сендера

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "id": 12
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>11</id>
</xml>
```

Пример ответа

JSON

```
{
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

AKMTA object

JSON

JSON

```
{
    "dkim_id": 4,
    "dkim_selector": "dkim_selector",
    "ips": [
        {
            "ip": "24.24.24.34",
            "node": 10
        }
    ],
    "lock_rules_obj": {
        "common_rules": [],
        "isp_rules": [
            {
                "isps": [
                    6
                ],
                "rules": [
                    {
                        "patterns": [
                            1004,
                            1005,
                            497
                        ],
                        "rule": 1001
                    }
                ]
            }
        ]
    },
    "redirect_email": "user@example.com",
    "retry_rules_obj": {
        "common_rules": [
            {
                "patterns": [
                    1003
                ],
                "rule": 1002
            }
        ],
        "isp_rules": [
            {
                "isps": [
                    1007,
                    1008,
                    1009,
                    1010,
                    1011,
                    1012
                ]
```

```
            ],
            "rules": [
                {
                    "patterns": [
                        1003
                    ],
                    "rule": 1002
                }
            ]
        }
    ]
},
"sender_dkim_id": 8,
"sender_dkim_selector": "sender_dkim_selector",
"sending_rules_obj": {
    "common_rules": [
        {
            "isps": [
                0
            ],
            "strategy": 1001
        }
    ],
    "ip_rules": [
        {
            "ips": [
                "24.24.24.34"
            ],
            "rules": [
                {
                    "isps": [
                        1007,
                        1009
                    ],
                    "strategy": 1002
                },
                {
                    "isps": [
                        1008,
                        1010
                    ],
                    "strategy": 1003
                }
            ]
        }
    ]
},
"users": [
    {
        "domain": "mx.localtest.ak",
```

```
            "pop3_pass": "nnr614g5",
            "redirect_email": "user1@example.com"
        },
        {
            "domain": "mx.localtest.ak",
            "pop3_pass": "nnr614g5",
            "redirect_email": "user2@example.com"
        }
    ]
}
```

XML

```xml
<xml>
  <dkim_id>4</dkim_id>
  <dkim_selector>dkim_selector</dkim_selector>
  <ips array='true'>
    <ip>24.24.24.34</ip>
    <node>10</node>
  </ips>
  <lock_rules_obj>
    <common_rules array='true'></common_rules>
    <isp_rules array='true'>
      <isps array='true'>6</isps>
      <rules array='true'>
        <patterns>1004</patterns>
        <patterns>1005</patterns>
        <patterns>497</patterns>
        <rule>1001</rule>
      </rules>
    </isp_rules>
  </lock_rules_obj>
  <redirect_email>user@example.com</redirect_email>
  <retry_rules_obj>
    <common_rules array='true'>
      <patterns array='true'>1003</patterns>
      <rule array='true'>1002</rule>
    </common_rules>
    <isp_rules array='true'>
      <isps>1007</isps>
      <isps>1008</isps>
      <isps>1009</isps>
      <isps>1010</isps>
      <isps>1011</isps>
      <isps>1012</isps>
      <rules array='true'>
        <patterns array='true'>1003</patterns>
        <rule>1002</rule>
      </rules>
    </isp_rules>
  </retry_rules_obj>
  <sender_dkim_id>8</sender_dkim_id>
  <sender_dkim_selector>sender_dkim_selector</sender_dkim_selector>
  <sending_rules_obj>
    <common_rules array='true'>
      <isps>0</isps>
      <strategy>1001</strategy>
    </common_rules>
    <ip_rules array='true'>
      <ips array='true'>24.24.24.34</ips>
      <rules array='true'>
        <isps>1007</isps>
```

```
        <isps>1007</isps>
        <isps>1009</isps>
        <strategy>1002</strategy>
      </rules>
      <rules array='true'>
        <isps>1008</isps>
        <isps>1010</isps>
        <strategy>1003</strategy>
      </rules>
    </ip_rules>
  </sending_rules_obj>
  <users>
    <domain>mx.localtest.ak</domain>
    <pop3_pass>nnr614g5</pop3_pass>
    <redirect_email>user1@example.com</redirect_email>
  </users>
  <users>
    <domain>mx.localtest.ak</domain>
    <pop3_pass>nnr614g5</pop3_pass>
    <redirect_email>user2@example.com</redirect_email>
  </users>
</xml>
```

Описание

Поле	Тип
ips	JSON array
dkim_id	int
dkim_selector	string
sender_dkim_id	int
sender_dkim_selector	string
redirect_email	string
sending_rules_obj	JSON Object
retry_rules_obj	JSON Object
lock_rules_obj	JSON Object
users	JSON array

Content object

Используется в API запросах, связанных с рассылкми.

JSON

JSON

```
{
  "subject": "Maybe very important message",
  "from_email": "",
  "from_name": "James",
  "replyto_email": "",
  "replyto_name": "",
  "message_id": 45
}
```

XML

```
<xml>
  <subject>Maybe very important message</subject>
  <from_email></from_email>
  <from_name>James</from_name>
  <replyto_email></replyto_email>
  <replyto_name></replyto_name>
  <message_id>45</message_id>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
message_id	int	12	Идентификатор сообщения
from_email	string	“test@example.com”	Перезаписать from-email
from_name	string	“Jenny’s List”	Перезаписать from-name
replyto_email	string	“test@example.com”	Перезаписать replyto_email
replyto_name	string	“Jenny”	Перезаписать replyto_name
subject	string	“Weather alert”	Перезаписать тему сообщения

Email rule object

JSON

JSON

```
{
  "type": "from_domain",
  "value": "altcraft.com",
  "senders": [
    {
      "sender_id": 287,
      "percent": 100,
      "from_email": "no-reply@altcraft.com",
      "replyto_email": "no-reply@altcraft.com",
      "from_name": ""
    }
  ]
}
```

XML

```
<xml>
  <type>from_domain</type>
  <value>altcraft.com</value>
  <senders array='true'>
    <sender_id>287</sender_id>
    <percent>100</percent>
    <from_email>no-reply@altcraft.com</from_email>
    <replyto_email>no-reply@altcraft.com</replyto_email>
    <from_name></from_name>
  </senders>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
type	string	“default”	Тип email-правила
value	string/int	“”	В зависимости от типа
senders	JSON array	`[]`	Список сендеров

Типы email-правилs

Тип	Тип значения	Пример	Описание
default	string	“”	Всегда применяется
group_id	int	12	Если идентификатор группы равен …
from_domain	string	“altcraft.com”	Если From-домен равен …
to_domain	string	“gmail.com”	Если To-домен равен …

File object

JSON

JSON

```
{
    "data": "data:text/csv; base64,SEVMTE8K",
    "name": "US.txt"
}
```

XML

```
<xml>
    <data>data:text/csv; base64,SEVMTE8K</data>
    <name>US.txt</name>
</xml>
```

Описание

Параметр	Тип	Описание
name	string	Имя прикреплённого файла
data	string	Файл полностью в кодировке base64
file_key	string	Ранее загруженный идентификатор файла, вместо данных
mimetype	string	Добавить информацию о MIME (не обязательно)

Profile data object

Используется в API запросах, связанных с импортом подписчиков в базы данных. В объекте передаются значения полей базы и подписки.

Информацию о всех полях конкретной базы данных можно получить по запросу к `api/v1.1/databases/fields_get/`.

Общая информация о базах данных и используемых полях доступна в пользовательской документации.

JSON

JSON

```
{
    "data": {
        "_regcountry": "US",
        "_regcity": "New York",
        "_regip": "94.231.119.122",
        "_regurl": "www.example.com",
        "_regdate": "2019-03-14T22:00:00Z",
        "_bdate": "1990-02-22T21:00:00Z",
        "_browser": "Thunderbird",
        "_city": "Ryazan",
        "_country": "RU",
        "_fname": "Olly",
        "_ip": "94.231.119.122",
        "_lname": "Lambert",
        "_postal_code": "390000",
        "_region": "Ryazan Oblast",
        "_vendor": "form_#31",
        "_sex": 0,
        "_status": 0,
        "_tz": "Europe/Moscow",
        "email": "example@altcraft.com",
        "phones": [
            "+79000000000"
        ],
        "custom_field": "custom_value",
        "subscriptions": [
            {
                "channel": "email",
                "email": "boris.yudaev@altcraft.com",
                "priority": 0,
                "resource_id": 1,
                "status": 0
            }
        ]
    }
}
```


XML

```
<xml>
    <data>
        <regcountry>US</regcountry>
        <regcity>New York</regcity>
        <regip>94.231.119.122</regip>
        <regurl>www.example.com</regurl>
        <regdate>2019-03-14T22:00:00Z</regdate>
        <bdate>1990-02-22T21:00:00Z</bdate>
        <browser>Thunderbird</browser>
        <city>Ryazan</city>
        <country>RU</country>
        <fname>Olly</fname>
        <ip>94.231.119.122</ip>
        <lname>Lambert</lname>
        <postal_code>390000</postal_code>
        <region>Ryazan Oblast</region>
        <vendor>form_#31</vendor>
        <sex>0</sex>
        <status>0</status>
        <tz>Europe/Moscow</tz>
        <email>example@altcraft.com</email>
        <phones array='true'>+79000000000</phones>
        <custom_field>custom_value</custom_field>
        <subscriptions array='true'>
            <channel>email</channel>
            <email>boris.yudaev@altcraft.com</email>
            <priority>0</priority>
            <resource_id>1</resource_id>
            <status>0</status>
        </subscriptions>
    </data>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
`bdate`	string	`1990-02-22T21:00:00Z`	Дата рождения в формате RFC 3339
`browser`	string	`Firefox`	Браузер
`_city`	string	`Ryazan`	Город
`_country`	string	`RU`	Страна в формате TLD
`_fname`	string	`Olly`	Имя

Параметр	Тип	Пример	Описание
`_ip`	string	`94.231.119.122`	IP адрес *
`_lname`	string	`Lambert`	Фамилия
`_postal_code`	int	`390000`	Почтовый индекс
`_region`	string	`Ryazan Oblast`	Регион
`_status`	int	`0`	Глобальный статус подписки
`_tz`	string	`Europe/Moscow`	Часовой пояс
`email`	string	`example@altcraft.com`	Адрес электронной почты
`phones`	JSON array	`["+79000000000"]`	Номера телефонов
`subscriptions`	JSON array	`["",""]`	Подписки профиля

`custom_field`	*	`custom_value`	Дополнительные поля базы данных.
			
			Возможно функциональное обновление полей. Для этого необходимо указать действие (`action`) и значение (`value`).
			
			Пример запроса для увеличения значения поля:
			“custom_field”: {
			“action”: “incr”,
			“value”: 1000
			}

`_regcountry`	string	`US`	Страна регистрации в формате TLD
`_regcity`	string	`New York`	Город регистрации
`_regip`	string	`94.231.119.122`	IP регистрации
`_regurl`	string	`www.example.org`	URL регистрации
`_regdate`	string	`2018-02-22T21:00:00Z`	Дата регистрации в формате RFC 3339
`_sex`	int	`0`	Идентификатор Пола, указывается согласно Список гендерных идентификаций
`_vendor`	string	`form#33`	Форма сбора данных клиентов

* При передаче в запросе на импорт IP адреса в формате IPv4 платформа автоматически определяет местоположение подписчика и записывает следующие системные поля данных:

JSON

```
{
        "_ip": "94.231.119.122", // уже добавляется само по себе
        "_lat": 54.6197, // географическая широта
        "_lon": 39.74, // географическая долгота
        "_postal_code": "390000",
        "_region": "Ryazanskaya Oblast",
        "_country": "RU",
        "_city": "Ryazan",
        "_tz": "Europe/Moscow"
}
```

XML

```
<xml>
    <ip>94.231.119.122</ip>// уже добавляется само по себе
    <lat>54.6197</lat>// географическая широта
    <lon>39.74</lon>// географическая долгота
    <postal_code>390000</postal_code>
    <region>Ryazanskaya Oblast</region>
    <country>RU</country>
    <city>Ryazan</city>
    <tz>Europe/Moscow</tz>
</xml>
```

Sender object

JSON

JSON

```
{
  "sender_id": 287,
  "percent": 100,
  "from_email": "no-reply@altcraft.com",
  "replyto_email": "no-reply@altcraft.com",
  "from_name": "James",
}
```

XML

```
<xml>
  <sender_id>287</sender_id>
  <percent>100</percent>
  <from_email>no-reply@altcraft.com</from_email>
  <replyto_email>no-reply@altcraft.com</replyto_email>
  <from_name>James</from_name>
</xml>
```

Описание

Параметр	Тип	Пример	Описание	SMS	Email
sender_id	int	123	Идентификатор сендера	+	+
percent	int	100	Процент распределения сообщений	+	+
from_email	string	“no-reply@altcraft.com”	Перезаписать from-email	-	+
replyto_email	string	“no-reply@altcraft.com”	Перезаписать replyto-email	-	+
from_name	string	“Altcraft”	Перезаписать from-name	+	+

Sender types

Сендеры создаются в интерфейсе администратора и используются для отправки SMS и email-сообщений.

Тип	Канал	Описание
“akmta”	email	Агент пересылки сообщений по умолчанию
“dmta”	email	Альтернативный отправщик (устарело)
“eshark”	email	Отправщик EmailShark (устарело)
“webzo”	email	Отправщик WebZo.net (устарело)
“mandrill”	email	Интеграция с API MailChimp Mandrill
“emaildirect”	email	Итеграция с EmailDirect (устарело)
“amazon_ses”	email	Интеграция с Amazon SES
“smsc_ru”	sms	Интеграция с SMSc.ru API
“bulksms_com”	sms	Интеграция с BulkSMS.com API

SMS rule object

JSON

JSON

```
{
    "senders": [
        {
            "from_email": "",
            "from_name": "Example",
            "percent": 100,
            "replyto_email": "",
            "sender_id": 798
        }
    ],
    "type": "default",
    "value": ""
}
```

XML

```
<xml>
  <senders array='true'>
    <from_email></from_email>
    <from_name>Example</from_name>
    <percent>100</percent>
    <replyto_email></replyto_email>
    <sender_id>798</sender_id>
  </senders>
  <type>default</type>
  <value></value>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
type	string	“default”	Тип SMS-правила
value	string/int	“”	В зависимости от типа
senders	JSON array	`[]`	Список сендеров

Типы SMS правил

Тип	Тип значения	Пример	Описание
default	string	“”	Всегда применяется
group_id	int	12	Если идентификатор группы равен …
from_domain	string	“altcraft.com”	Если From-домен равен …

Start schedule object

JSON

JSON

```
{
    "start_hours": [
        2,
        4
    ],
    "start_mdays": [
        1
    ],
    "start_mins": [
        0,
        10,
        20,
        30,
        40,
        50
    ],
    "start_months": [
        12
    ],
    "start_wdays": [],
    "start_years": [
        2016
    ]
}
```

XML

```
<xml>
    <start_hours>2</start_hours>
    <start_hours>4</start_hours>
    <start_mdays array='true'>1</start_mdays>
    <start_mins>0</start_mins>
    <start_mins>10</start_mins>
    <start_mins>20</start_mins>
    <start_mins>30</start_mins>
    <start_mins>40</start_mins>
    <start_mins>50</start_mins>
    <start_months array='true'>12</start_months>
    <start_wdays></start_wdays>
    <start_years array='true'>2024</start_years>
</xml>
```

Описание

Параметр	Тип	Пример	Описание
start_mins	JSON array	`[1, 31]`	Старт по минутам, 0-59
start_hours	JSON array	`[22]`	Старт по часам, 0-23
start_mdays	JSON array	`[]`	Старт по дням месяца, 1-31
start_wdays	JSON array	`[]`	Старт по дням недели, …
start_months	JSON array	`[]`	Старт по месяцам, 1-12
start_years	JSON array	`[]`	Старт по годам
tz	string	Europe/Moscow	Часовой пояс расписания рассылок
profile	bool	true	Отправка по часовому поясу подписчиков
losers	bool	true	Отправить сразу тем, кто пропустил отправку по местному времени.

Subscription object

Описание

Объект `subscription` содержит данные о подписке профиля на коммуникационные каналы. Структура объекта варьируется в зависимости от типа канала и контекста API.

Основная структура

Параметры	Тип	Пример	Описание
"channel"	string	"email"	Тип канала ("email" / "sms" / "push" / "notify" / "viber" / "whatsapp" / "telegram_bot")
"db_id"	int	1	Идентификатор базы данных
"email"	string	"john@example.com"	Email-адрес
"phone"	string	"+79000000000"	Номер телефона **(Для SMS-подписки)**
"provider"	string	"Firefox"	Тип провайдера **(Для Push-подписки)**
"subscription_id"	string	"abcdefghijklmnqrstuvwxyz"	Идентификатор подписки **(Для Push-подписки)**
"cc_data"	object	\{ "id": 1 \}	ID чата **в Telegram-бот**
		\{ "phone": "+79000000000" \}	Телефон профиля в **WhatsApp*/ Viber™ / Notify**
"priority"	int	1	Приоритет подписки (опциональное поле). В случае, если приоритет подписки передается в запросе, опция "Повышать приоритет для новых подписок" в ресурсе не работает.
"profile_id"	string	"54759eb3c090d83494e2d804"	Идентификатор профиля
"resource_id"	int	1	Идентификатор ресурса
"custom_fields"	object	"subscribed"	Стандартные и дополнительные поля подписки (опциональное поле). Получить информацию о доступных полях подписки в ресурсе можно с помощью метода /v1.1/resources/fields_get
"status"	string	"subscribed"	Статус подписки Подробнее

Параметры	Тип	Пример	Описание
"cats"	array	`[ "category_1", "category_2" ]`	Категории ресурса, на которые подписан профиль (опциональное поле).

Расширенная структура при запросе профиля через хендлер `/api/v1.1/profiles/get/`

При запросе профиля через хендлер `/api/v1.1/profiles/get/` объект `subscription` содержит дополнительные метаданные:

- Данные канала хранятся в отдельных объектах:
  - `"email_data"` - для Email-подписок
  - `"phone_data"` - для SMS/Viber/WhatsApp
  - `"cc_data"` - для Telegram
- Технические поля
  - `"hash_id"` - уникальный хеш подписки
  - `"is_delete"` - флаг удаления (soft delete)
- История:
  - `"reg_info"` - дата регистрации подписки

Пример структуры

JSON

```
{
  "resource_id": 4,
  "channel": "email",
  "email_data": {
        "email": "user@example.com",
        "domain": "example.com",
        "md5": "d922b1da12193df553aad2bee61da46b"
  },
  "status": "subscribed",
  "reg_info": {
    "date": "2025-07-23T11:27:05.487Z"
  },
  "hash_id": "7641c5c5",
  "is_delete": false
}
```

XML

```
<xml>
<subscription>
    <resource_id>4</resource_id>
    <channel>email</channel>
    <email_data>
        <email>user@example.com</email>
        <domain>example.com</domain>
        <md5>d922b1da12193df553aad2bee61da46b</md5>
    </email_data>
    <status>subscribed</status>
    <reg_info>
        <date>2025-07-23T11:27:05.487Z</date>
    </reg_info>
    <hash_id>7641c5c5</hash_id>
    <is_delete>false</is_delete>
</subscription>
</xml>
```

Пример объекта

JSON

```
{
  "subscriptions": [
    {
      "resource_id": 1,
      "channel": "email",
      "email": "example@example.com"
    },
    {
      "resource_id": 2,
      "channel": "sms",
      "phone": "+79000000000"
    },
    {
      "resource_id": 1,
      "channel": "push",
      "subscription_id": "a81c264a938b475",
      "provider": "android-firebase",
      "custom_fields": {
        "_browser_name": "Firefox",
        "_device_type": "web",
        "custom_field_1": "test value"
      },
      "cats": [
        "category_1",
        "category_2"
      ]
    },
    {
      "resource_id": 1,
      "channel": "whatsapp",
      "cc_data": {
        "phone": "+79000000000"
      }
    },
    {
      "resource_id": 1,
      "channel": "viber",
      "cc_data": {
        "phone": "+79000000000"
      }
    },
    {
      "resource_id": 1,
      "channel": "telegram_bot",
      "cc_data": {
        "id": 12345
      }
    },
    {
```

```
    {
      "resource_id": 1,
      "channel": "notify",
      "cc_data": {
        "phone": "+79000000000"
      }
    }
  ]
}
```

XML

```
<xml>
<subscriptions>
  <subscription>
    <resource_id>1</resource_id>
    <channel>email</channel>
    <email>example@example.com</email>
  </subscription>

  <subscription>
    <resource_id>2</resource_id>
    <channel>sms</channel>
    <phone>+79000000000</phone>
  </subscription>

  <subscription>
    <resource_id>1</resource_id>
    <channel>push</channel>
    <subscription_id>a81c264a938b475</subscription_id>
    <provider>android-firebase</provider>
    <custom_fields>
      <_browser_name>Firefox</_browser_name>
      <_device_type>web</_device_type>
      <custom_field_1>test value</custom_field_1>
    </custom_fields>
    <cats>
      <category>category_1</category>
      <category>category_2</category>
    </cats>
  </subscription>

  <subscription>
    <resource_id>1</resource_id>
    <channel>whatsapp</channel>
    <cc_data>
      <phone>+79000000000</phone>
    </cc_data>
  </subscription>

  <subscription>
    <resource_id>1</resource_id>
    <channel>viber</channel>
    <cc_data>
      <phone>+79000000000</phone>
    </cc_data>
  </subscription>

  <subscription>
    <resource_id>1</resource_id>
    <channel>telegram_bot</channel>
```

```
    <channel>telegram_bot</channel>
    <cc_data>
      <id>12345</id>
    </cc_data>
  </subscription>

  <subscription>
    <resource_id>1</resource_id>
    <channel>notify</channel>
    <cc_data>
      <phone>+79000000000</phone>
    </cc_data>
  </subscription>
</subscriptions>
```

Subscription status

Описание

Параметр `status` в объекте `subscription` перезаписывает статус подписки. Рекомендуется передавать его только в тех случаях, когда необходимо обновить статус.

Использовать можно как строчные, так и численные обозначения статуса подписки.

Статус	Описание	Число
subscribed	Коммуникация возможна, подписан	0
unsubscribed	Отписался сам или отписан	1
complained	Пожаловался на сообщение (FBL, CFL)	2
hardbounced	Недействительный адрес	3
unconfirmed	Не подтвердил подписку, возможна отправка писем подтверждения	4
suspended	Временно приостановленная подписка	5
invalid	Для других причин, когда коммуникация невозможна	6

**Организация Meta, которой принадлежат продукты Instagram, Facebook и WhatsApp, признана экстремистской и запрещена на территории РФ.*
*** Viber™ является товарным знаком Rakuten Group, Inc. Заблокирован на территории РФ.*

Trigger types

Тип	Описание
api_call	По вызову API
api_import	При импорте профиля по API
action	При совершении подписчиков какого-либо действия
update	При обновлении полей профиля
pixel	При активации пикселя

Добавить запрос сегментации

Описание

Добавление нового запроса сегментации к внешней базе данных SQL.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_segments/add/`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
name	string	"Main query"	Да	Название запроса
shortname	string	"query1"	Да	Короткое название запроса
description	string	"query to the DB"	Нет	Описание запроса
connector_id	int	2	Да	ID коннектора к внешней ДБ
column	string	"author"	Да	Колонка, используемая для поиска профилей
sql	string	"SELECT * FROM table"	Нет	SQL-запрос
cache_time_min	int	35	Нет	Время кэширования результата
groups_id	\[\]int	\[0, 1 \]	Нет	Список ID групп для поиска по умолчанию – \[\]
parameters	array	`[{ "name": "CITY", "type": "string", "title": "head", "options_type": "query" }]`	Нет	Опции пользовательских переменных, используемых в SQL-запросах

Пример запроса

JSON

```
{
   "token": "abcdefghijklmnqrstuvwxyz",
   "name": "Main Query",
   "shortname": "query1",
   "sql": "SELECT * FROM table",
   "connector_id": 1,
   "column": "price"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <name>Main Query</name>
  <shortname>query1uery</shortname>
  <sql>SELECT * FROM table</sql>
  <connector_id>1</connector_id>
  <column>price</column>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
          "saved_id": 1
     },
     "error": 0,
     "error_text": "Successful operation",
}
```

XML

```
<xml>
  <data>
    <saved_id>1</saved_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор созданного запроса

Обновить запрос сегментации

Описание

Обновить запрос сегментации.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_segments/update`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	"abcdefghijklmnqrstuvwxyz"	Да	API-токен
id	int	1	Да	ID запроса
name	string	"Main query"	Да	Название запроса
shortname	string	"query1"	Да	Короткое название запроса
description	string	"query to the DB"	Нет	Описание запроса
connector_id	int	2	Да	ID коннектора к внешней ДБ
column	string	"author"	Да	Колонка, используемая для поиска профилей
sql	string	"SELECT * FROM table"	Нет	SQL-запрос
cache_time_min	int	35	Нет	Время кэширования результата
groups_id	\[\]int	\[0, 1 \]	Нет	Список ID групп для поиска по умолчанию – \[\]
parameters	object	{ "name": "CITY", "type": "string", "title": "head", "options_type": "query" }	Нет	Опции пользовательских переменных, используемых в SQL-запросах

Пример запроса

JSON

```
{
   "token": "abcdefghijklmnqrstuvwxyz",
   "id": 1,
   "name": "Main Query",
   "shortname": "query1",
   "sql": "SELECT * FROM table",
   "connector_id": 1,
   "column": "price"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
  <name>Main Query</name>
  <shortname>query1uery</shortname>
  <sql>SELECT * FROM table</sql>
  <connector_id>1</connector_id>
  <column>price</column>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
          "saved_id": 1
     },
     "error": 0,
     "error_text": "Successful operation",
}
```

XML

```
<xml>
  <data>
    <saved_id>1</saved_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор обновленного запроса

Получить информацию о запросе сегментации

Описание

Получить информацию о запросе сегментации.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_segments/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да, если не указан `shortname`	ID запроса
shortname	string	“query1”	Да, если не указан `id`	Короткое название запроса

Пример запроса

JSON

```
{
  "id": 2,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>2</id>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "id": 2,
        "name": "QueryforextDB",
        "shortname": "query",
        "description": "External DB segment",
        "sql": "SELECT * FROM table",
        "connector_id": 1,
        "parametrs": {
           "name": "variable1",
           "title": "var",
           "default_value": "low"
        },
        "cache_time_min": 35
     },
     "error": 0,
     "error_text": "Successful operation",
}
```

XML

```
<xml>
   <data>
      <cache_time_min>35</cache_time_min>
      <connector_id>1</connector_id>
      <description>External DB segment</description>
      <id>2</id>
      <name>QueryforextDB</name>
      <parametrs>
         <default_value>low</default_value>
         <name>variable1</name>
         <title>var</title>
      </parametrs>
      <shortname>query</shortname>
      <sql>SELECT * FROM table</sql>
   </data>
   <error>0</error>
   <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки

Параметр	Тип	Описание
error_text	string	Текст ошибки
id	int	ID запроса
name	string	Название запроса
shortname	string	Короткое название запроса
description	string	Описание запроса
connector_id	int	ID коннектора к внешней ДБ
column	string	Колонка, используемая для поиска профилей
sql	string	SQL-запрос
cache_time_min	int	Время кэширования результата
groups_id	[]int	Список ID групп для поиска по умолчанию – []
parameters	object	Опции пользовательских переменных, используемых в SQL-запросах

Получить список запросов на сегментацию

Описание

Получить список запросов на сегментацию.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_segments/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
from_id	int	1	Нет	Идентификатор первой рассылки в ответе по умолчанию – 1
limit	int	2	Нет	Лимит на количество возвращаемых данных по умолчанию – 1000
groups	[]String	[“group1”, “group2”]	Нет	Список групп для поиска по умолчанию – []
groups_id	[]int	[0, 1]	Нет	Список Id групп для поиска по умолчанию – []

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "id": 2,
        "name": "QueryforextDB",
        "shortname": "query",
        "description": "External DB segment",
        "sql": "SELECT * FROM table",
        "connector_id": 1,
        "parametrs": {
           "name": "variable1",
           "title": "var",
           "default_value": "low"
        },
        "cache_time_min": 35
     },
     "error": 0,
     "error_text": "Successful operation",
}
```

XML

```
<xml>
   <data>
      <cache_time_min>35</cache_time_min>
      <connector_id>1</connector_id>
      <description>External DB segment</description>
      <id>2</id>
      <name>QueryforextDB</name>
      <parametrs>
         <default_value>low</default_value>
         <name>variable1</name>
         <title>var</title>
      </parametrs>
      <shortname>query</shortname>
      <sql>SELECT * FROM table</sql>
   </data>
   <error>0</error>
   <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	ID запроса
name	string	Название запроса
shortname	string	Короткое название запроса
description	string	Описание запроса
connector_id	int	ID коннектора к внешней ДБ
column	string	Колонка, используемая для поиска профилей
sql	string	SQL-запрос
cache_time_min	int	Время кэширования результата
groups_id	[]int	Список ID групп для поиска по умолчанию – []
parameters	object	Опции пользовательских переменных, используемых в SQL-запросах

Удалить запрос сегментации

Описание

Удалить запрос сегментации.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_segments/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	ID запроса

Пример запроса

JSON

```
{
  "id": 2,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>2</id>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": null,
  "error": 0,
  "error_text": "Successful operation",
}
```

XML

```
<xml>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Добавить запрос для шаблонов

Описание

Добавление нового запроса для шаблонов к внешней базе данных SQL.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_templates/add`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
name	string	“Main query”	Да	Название запроса
shortname	string	“query1”	Да	Короткое название запроса
description	string	“query to the DB”	Нет	Описание запроса
connector_id	int	2	Да	ID коннектора к внешней ДБ
sql	string	“SELECT * FROM table”	Нет	SQL-запрос
groups_id	[]int	[0, 1]	Нет	Список ID групп для поиска по умолчанию – []

Пример запроса

JSON

```
{
   "token": "abcdefghijklmnqrstuvwxyz",
   "name": "Main Query",
   "shortname": "query1",
   "sql": "SELECT * FROM table",
   "connector_id": 1,
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <name>Main Query</name>
  <shortname>query1uery</shortname>
  <sql>SELECT * FROM table</sql>
  <connector_id>1</connector_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
          "saved_id": 1
     },
     "error": 0,
     "error_text": "Successful operation",
}
```

XML

```
<xml>
  <data>
    <saved_id>1</saved_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор созданного запроса

Обновить запрос для шаблонов

Описание

Обновить запрос для шаблонов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_templates/update`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	ID запроса
name	string	“Main query”	Да	Название запроса
shortname	string	“query1”	Да	Короткое название запроса
description	string	“query to the DB”	Нет	Описание запроса
connector_id	int	2	Да	ID коннектора к внешней ДБ
sql	string	“SELECT * FROM table”	Нет	SQL-запрос
groups_id	[]int	[0, 1]	Нет	Список ID групп для поиска по умолчанию – []

Пример запроса

JSON

```
{
   "token": "abcdefghijklmnqrstuvwxyz",
   "id": 1,
   "name": "Main Query",
   "shortname": "query1",
   "sql": "SELECT * FROM table",
   "connector_id": 1
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
  <id>1</id>
  <name>Main Query</name>
  <shortname>query1uery</shortname>
  <sql>SELECT * FROM table</sql>
  <connector_id>1</connector_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
          "saved_id": 1
     },
     "error": 0,
     "error_text": "Successful operation",
}
```

XML

```
<xml>
  <data>
    <saved_id>1</saved_id>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
saved_id	int	Идентификатор обновленного запроса

Получить информацию о запросе для шаблонов

Описание

Получить информацию о запросе для шаблонов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_templates/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да, если не указан `shortname`	ID запроса
shortname	string	“query1”	Да, если не указан `id`	Короткое название запроса

Пример запроса

JSON

```
{
  "id": 2,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>2</id>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "id": 2,
        "name": "QueryforextDB",
        "shortname": "query",
        "description": "External DB segment",
        "sql": "SELECT * FROM table",
        "connector_id": 1
     },
     "error": 0,
     "error_text": "Successful operation",
}
```

XML

```
<xml>
   <data>
      <connector_id>1</connector_id>
      <description>External DB segment</description>
      <id>2</id>
      <name>QueryforextDB</name>
      <shortname>query</shortname>
      <sql>SELECT * FROM table</sql>
   </data>
   <error>0</error>
   <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	ID запроса
name	string	Название запроса
shortname	string	Короткое название запроса
description	string	Описание запроса
connector_id	int	ID коннектора к внешней ДБ
sql	string	SQL-запрос

Параметр	Тип	Описание
groups_id	[]int	Список ID групп для поиска по умолчанию – []

Получить список запросов для шаблонов

Описание

Получить список запросов для шаблонов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_templates/list`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных по умолчанию – “json”
from_id	int	1	Нет	Идентификатор первой рассылки в ответе по умолчанию – 1
limit	int	2	Нет	Лимит на количество возвращаемых данных по умолчанию – 1000
groups	[]String	[“group1”, “group2”]	Нет	Список групп для поиска по умолчанию – []
groups_id	[]int	[0, 1]	Нет	Список Id групп для поиска по умолчанию – []

Пример запроса

JSON

```
{
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "id": 2,
        "name": "QueryforextDB",
        "shortname": "query",
        "description": "External DB segment",
        "sql": "SELECT * FROM table",
        "connector_id": 1,
        "parametrs": {
           "name": "variable1",
           "title": "var",
           "default_value": "low"
        },
        "cache_time_min": 35
     },
     "error": 0,
     "error_text": "Successful operation",
}
```

XML

```
<xml>
   <data>
      <cache_time_min>35</cache_time_min>
      <connector_id>1</connector_id>
      <description>External DB segment</description>
      <id>2</id>
      <name>QueryforextDB</name>
      <parametrs>
         <default_value>low</default_value>
         <name>variable1</name>
         <title>var</title>
      </parametrs>
      <shortname>query</shortname>
      <sql>SELECT * FROM table</sql>
   </data>
   <error>0</error>
   <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
id	int	ID запроса
name	string	Название запроса
shortname	string	Короткое название запроса
description	string	Описание запроса
connector_id	int	ID коннектора к внешней ДБ
sql	string	SQL-запрос
groups_id	[]int	Список ID групп для поиска по умолчанию – []

Удалить запрос для шаблонов

Описание

Удалить запрос для шаблонов.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/datatables/query_templates/delete`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
id	int	1	Да	ID запроса

Пример запроса

JSON

```
{
  "id": 2,
  "token": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
  <id>2</id>
  <token>abcdefghijklmnqrstuvwxyz</token>
</xml>
```

Пример ответа

JSON

```
{
  "data": null,
  "error": 0,
  "error_text": "Successful operation",
}
```

XML

```
<xml>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>t>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Загрузить файл

Описание

Загрузка файла и получение его идентификатора.

URL-адрес

Метод: POST

`https://example.com/api/v1.1/file_upload`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“json”	Нет	Формат возвращаемых данных (по умолчанию – “json”)
header	string	“emails.csv”	Да	
data	string	“data:text/csv; base64,YQpi…”	Да	

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "data": "data:text/csv; base64,SEVMTE8K",
    "header": "US.csv"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <data>data:text/csv; base64,SEVMTE8K</data>
    <header>US.csv</header>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "file": "fd7006e5-5c42-4eba-aa6a-f6cb7f5cb6f2"
    },
    "error": 0,
    "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <file>fd7006e5-5c42-4eba-aa6a-f6cb7f5cb6f2</file>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки
file	string	Уникальный идентификатор файла

Получить веб-версию сообщения

Описание

Получить веб-версию сообщения.

URL-адрес

`https://example.com/api/v1.1/webver/get`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
format	string	“scv”	Нет	Формат возвращаемых данных (по умолчанию - “json”)
send_message_id	string	“abcdefghijklmnqrstuvwxyz”	Да	Идентификатор веб-версии сообщения

Пример запроса

JSON

```
{
    "token": "abcdefghijklmnqrstuvwxyz",
    "send_message_id": "abcdefghijklmnqrstuvwxyz"
}
```

XML

```
<xml>
    <token>abcdefghijklmnqrstuvwxyz</token>
    <send_message_id>abcdefghijklmnqrstuvwxyz</send_message_id>
</xml>
```

Пример ответа

JSON

```
{
    "data": {
        "createtime": "2018-08-24T09:26:23.292Z",
        "email": {
            "from_email": "example@example.com",
            "from_name": "example.com",
            "html": "HTML content ...",
            "replyto_email": "example@example.com",
            "replyto_name": "example.com",
            "subject": "Welcome to Altcraft",
            "text": "Text version also important\n\n--\nUnsubscribe:
http://trk.gmail.com/37cfe6oTGQF4LYqk6oMi8eCtZ?dummy=1"
        }
    },
    "error": 0,
    "error_text": "Successful operation"
}
```


XML


```
<xml>
  <data>
    <createtime>2018-08-24T09:26:23.292Z</createtime>
    <email>
      <from_email>example@example.com</from_email>
      <from_name>example.com</from_name>
      <html>HTML content ...</html>
      <replyto_email>example@example.com</replyto_email>
      <replyto_name>example.com</replyto_name>
      <subject>Welcome to Altcraft</subject>
      <text>Text version also important --
Unsubscribe: http://trk.gmail.com/37cfe6oTGQF4LYqk6oMi8eCtZ?dummy=1</text>
    </email>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```


Возвращаемые параметры

Параметр	Тип	Описание
error	int	Код ошибки
error_text	string	Текст ошибки

Параметр	Тип	Описание
createtime	date	Дата создания сообщения
email	JSON object	Шаблон сообщения

Push провайдеры

Push провайдер	Web / App	Имя провайдера в API запросе	
		Идентификатор 1	Идентификатор 2
Google Chrome (устаревший)	Web	"google"	"Chrome"
Mozilla Firefox	Web	"mozilla"	"Firefox"
Safari	Web	"safari"	"Safari"
Android Firebase	App	"android-firebase"	"AndroidFirebase"
IOS Firebase	App	"ios-firebase"	"IOSFirebase"
Firefox Firebase	Web	"firefox-firebase"	"FirefoxFirebase"
Chrome Firebase	Web	"chrome-firebase"	"ChromeFirebase"
Opera Firebase	Web	"opera-firebase"	"OperaFirebase"
Yandex Firebase	Web	"yandex-firebase"	"YandexBrowserFirebase"
Samsung Firebase	Web	"samsung-firebase"	"SamsungInternetforAndroidFirebase"
IOS APNs	App	"ios-apns"	"IOSAPNs"
AppMetrica with iOS device id	App	"yandex-Appmetrica-device-id-ios"	"YandexAppMetricaDeviceIDiOS"
AppMetrica with Android device id	App	"yandex-Appmetrica-device-id-android"	"YandexAppMetricaDeviceIDAndroid"
AppMetrica with iOS IFA	App	"yandex-Appmetrica-ios-ifa"	"YandexAppMetricaIOSIFA"
AppMetrica with Google AID	App	"yandex-Appmetrica-google-aid"	"YandexAppMetricaGoogleAID"
AppMetrica with Android push token	App	"yandex-Appmetrica-android-token"	"YandexAppMetricaAndroidToken"
AppMetrica with iOS push token	App	"yandex-Appmetrica-ios-token"	"YandexAppMetricaIOSToken"
MailRu (устаревший)	App	"mr"	"MR"
Android Huawei Mobile Services	App	"android-huawei"	"AndroidHuawei"
IOS Huawei Mobile Services	Web	"ios-huawei"	"IOSHuawei"
Android RuStore	App	"android-rustore"	"AndroidRuStore"

Дедупликация запросов

Описание

Повторный запрос — это запрос к API, который содержит тот же заголовок и те же данные, что и первичный запрос. Повторный запрос может отправляться в том случае, если в момент получения данных произошёл сбой соединения. Платформа не будет воспринимать повторный запрос, если он приводит к изменению данных, чтобы исключить дублирование событий. Эта схема работает по умолчанию и не требует дополнительной настройки. По умолчанию платформа осуществляет дедупликацию в течение 1 дня с момента первого запроса.

Платформа осуществляет дедупликацию по следующему алгоритму:

- Проверка наличия заголовка **Request-ID**
- Если Request-ID не указан, то проверяется контрольная сумма тела и заголовков запроса

Следует отметить, что не все запросы поддерживают дедупликацию по контрольной сумме, и если в таких запросах не указан Request-ID, то проверка на дублирование **проводиться не будет**. Список таких запросов вы можете найти в таблице ниже.

> **INFO**
>
> Различные клиенты отправки запросов могут добавлять в запрос свои собственные системные заголовки или другую информацию. Это может привести к **несрабатыванию** дедупликации запроса.

Как понять, что запрос был воспринят как дублирующий?

Ответ на повторный запрос будет такой же, как и на первичный запрос. Однако повторный запрос будет содержать в себе дополнительные заголовки:

```
Duplicate-Key: NeOScRqBgsqj67CR5dYcqvcmPZA=
Duplicate-Request: true
```

Заголовок **Duplicate-Key** состоит из Request-ID запроса и токена отправителя. Если же Request-ID не передан, то заголовок будет хранить в себе хешированные данные запроса.

Как послать запрос, чтобы он не воспринимался как дублирующий?

Рекомендуется маркировать каждый запрос уникальным идентификатором, тогда, изменяя этот идентификатор, вы сможете выполнять запросы, даже если они содержат одинаковые данные.

Сделать это можно одним из способов:

- Измените данные в теле запроса или URL, например, `&request_id=12345`
- Добавьте заголовок **Request-ID** с идентификатором запроса

API запросы, у которых работает дедупликация

На повторы проверяются все запросы, которые выполняют модификацию данных или активацию/запуск сущностей:

Категория	Запрос
Профили (устаревшие API-методы)	/v1.0/lists/:list_id/lead/unconfirm /v1.0/lists/:list_id/lead/unsubscribe /v1.0/lists/:list_id/lead/tags_add /v1.0/lists/:list_id/lead/tags_remove /v1.0/lists/:list_id/lead/subscribe /v1.0/lists/:list_id/lead/complain /v1.0/lists/:list_id/suspend_all /v1.0/lists/:list_id/unsuspend_all /v1.0/lists/:list_id/import
Интеграции	/integrations/lpgenerator/profile_import /integrations/tilda/profile_import /integrations/any/profile_import
Стоп-список	/v1.1/suppresses/add /v1.1/suppresses/update /v1.1/suppresses/domain_add /v1.1/suppresses/domains_add /v1.1/suppresses/domain_remove /v1.1/suppresses/domains_remove /v1.1/suppresses/email_add /v1.1/suppresses/emails_add /v1.1/suppresses/email_remove /v1.1/suppresses/emails_remove /v1.1/suppresses/phone_add /v1.1/suppresses/phones_add /v1.1/suppresses/phone_remove /v1.1/suppresses/phones_remove
Подписки	/v1.1/subscriptions/set /v1.1/subscriptions/suspend_all /v1.1/subscriptions/unsuspend_all /v1.1/subscribers/import /v1.1/subscribers/update
Сегменты	/v1.1/segments/add /v1.1/segments/update /v1.1/segments/refresh_counters
Рассылки	/v1.1/campaigns/deactivate /v1.1/campaigns/activate /v1.1/campaigns/clone

Категория	Запрос
Триггерные рассылки	/v1.1/campaigns/triggers/add /v1.1/campaigns/triggers/update /v1.1/campaigns/triggers/start /v1.1/campaigns/triggers/import_and_start /v1.1/campaigns/triggers/clone /v1.1/triggers/start_batch (только с Request-ID) /v1.1/triggers/start_batch_task (только с Request-ID) /v1.1/triggers/import_and_start_batch (только с Request-ID) /v1.1/triggers/import_and_start_batch_task (только с Request-ID)
Броадкасты	/v1.1/campaigns/broadcasts/add /v1.1/campaigns/broadcasts/update /v1.1/campaigns/broadcasts/start
Регулярные рассылки	/v1.1/campaigns/regulars/add /v1.1/campaigns/regulars/update /v1.1/campaigns/regulars/start
Сендеры	/v1.1/admin/senders/queue/hold /v1.1/admin/senders/queue/unhold /v1.1/admin/senders/queue/purge /v1.1/admin/senders/update /v1.1/admin/senders/add_to_account /v1.1/admin/senders/add /v1.1/admin/vsenders/add /v1.1/admin/vsenders/update /v1.1/admin/vsenders/clone
Ноды	/v1.1/admin/nodes/mta/start /v1.1/admin/nodes/mta/stop
Статический сегмент	/v1.1/profiles/segstatic/add /v1.1/profiles/segstatic/remove
Профили	/v1.1/profiles/add /v1.1/profiles/update /v1.1/profiles/import /v1.1/profiles/add_batch /v1.1/profiles/update_batch /v1.1/profiles/import_batch
Промокоды	/v1.1/promocodes/attach /v1.1/promocodes/activate /v1.1/promocodes/import /v1.1/promocodes/detach /v1.1/promocodes/update
Связи	/v1.1/relations/attach /v1.1/relations/detach /v1.1/relations/edit /v1.1/relations/strengthen

Категория	Запрос
Базы данных	/v1.1/databases/wipe /v1.1/databases/refresh_counters
Вирутальные сендеры	/v1.1/vsenders/add /v1.1/vsenders/update /v1.1/vsenders/clone
Сценарии	/v1.1/workflows/start /v1.1/workflows/import_and_start /v1.1/workflows/import_and_start_batch (только с Request-ID) /v1.1/workflows/import_and_start_batch_task (только с Request-ID)
Регистрация событий пушей	/v1.1/pushes/post_events /v1.1/pushes/crypto_post_events
Шаблоны	/v1.1/templates/add /v1.1/templates/update /v1.1/templates/delete
Ресурсы	/v1.1/resources/push_subscribe /v1.1/resources/refresh_counters
Таблицы данных	/v1.1/datatables/query_templates/update /v1.1/datatables/query_templates/delete /v1.1/datatables/query_templates/add /v1.1/datatables/query_segments/update /v1.1/datatables/query_segments/delete /v1.1/datatables/query_segments/add

Работа с API через RabbitMQ

Описание

В данной статье рассматриваются способы отправки API запросов через RabbitMQ.

> **NOTE**
>
> Для идентификации запросов посылается уникальный параметр **request_id.**

Очереди

Во время работы с API в RabbitMQ используются две очереди:

1. **api_req** — очередь для входящих запросов;
2. **api_resp** — очередь, в которую попадают ответы на запросы (только если указан `request_id`)

Структура отправляемого JSON сообщения

Параметр	Тип	Пример	Обязательный	Описание
request_id	string	“abcd1234”	Да, если нужен ответ на запрос	Идентификатор запроса
request	string	api/v1.1/campaigns/triggers/import_and_start_batch	Да	Путь API запроса

body	JSON обьект	“body”: {
		“data”:
		{
		“_fname”: “Fname1”,
		“_lname”: “Lname1”,
		“email”:
		“profile1@example.com”
		},
		“email”:
		“profile1@example.com”,
		“db_id”: 1,
		“detect_geo”: true,
		“token”:
		“abcdefghijklmnopqrstuvwxyzABCD”
		}
			Да	Тело API запроса

Структура принимаемого JSON сообщения

Параметр	Тип	Пример	Описание

body	JSON обьект	“body”:
		{
		“error”:0,
		“error_text”:“Successful operation”,
		“profile_id”:“60f039e830b8bcb28392f8eb”
		}
			Тело ответа на запрос

request_id	string	“abcd1234”	Идентификатор запроса

Пример работы с API на Go

Послать API запрос можно с помощью скрипта, обращающегося к RabbitMQ:

1. Написать и выполнить скрипт

Пример

```
package main

        "encoding/json"
        "log"

        "github.com/streadway/amqp"
)

const accID = 1
const resourceID = 3
const dbID = 23
const msgID = 17
const segmentID = 85
const amountOfPushMsgs = 1
const emailDomain = "example.com"

const req = `{
        "account_id": 1,
        "request_id": "db1894e4-1a2c-4021-8233-9cca0b96b79e",
        "request": "api/v1.1/campaigns/triggers/import_and_start_batch",
    "body": {
                "token": "abcdefghijklmnopqrstuvwxyzABC"
                "format": "json",
                "trigger_id": 240,
                "skip_triggers": false,
                "detect_geo": true,
                "matching": "custom",
                "field_name":"CustomF",

                "custom_data":{
                        "some":"some0"
                },
                "content":{
                        "someCont":"someCont0"
                },
                "data": [
                        {
                                "data":{
                                        "_fname":"NUMBER13",
                                        "_lname":"Lambert",
                                        "phones":"790000000013",
                                        "email": "profile1@example.com",
                                        "CustomF":18
                                },
                                "custom_data":{
                                        "some":"some1"
                                },
                                "content":{
```

```
                                "content":{
                                        "someCont":"someCont1"
                                }
                        },
                        {
                                "data":{
                                        "_fname":"NUMBER14",
                                        "_lname":"Hard",
                                        "phones":"790000000014",
                                        "email": "profile2@example.com",
                                        "CustomF":14
                                }
                        }
                ]
        }
}`

func failOnError(err error, msg string) {
        if err != nil {
                log.Fatalf("%s: %s", msg, err)
        }
}

func main() {
        var err error

        var conn *amqp.Connection
        conn, err = amqp.Dial("amqp://example:abcdefghijklmnopqrstuvwxyz127.0.0.1:5672/")
        failOnError(err, "Failed to connect to RabbitMQ")
        defer conn.Close()

        var ch *amqp.Channel
        ch, err = conn.Channel()
        failOnError(err, "Failed to open RabbitMQ channel")
        defer ch.Close()

        qUeueImport, err := ch.QueueDeclare(
                "api_req", // name
                true,      // durable
                false,     // delete when unused
                false,     // exclusive
                false,     // no-wait
                nil,       // arguments
        )
        failOnError(err, "Failed to declare RabbitMQ queue")

        for i := 0; i < amountOfPushMsgs; i++ {
                var bodyMap = make(map[string]interface{})
                err = json.Unmarshal([]byte(req), &bodyMap)
                failOnError(err, "Failed json.Unmarshal to bodyMap")
```

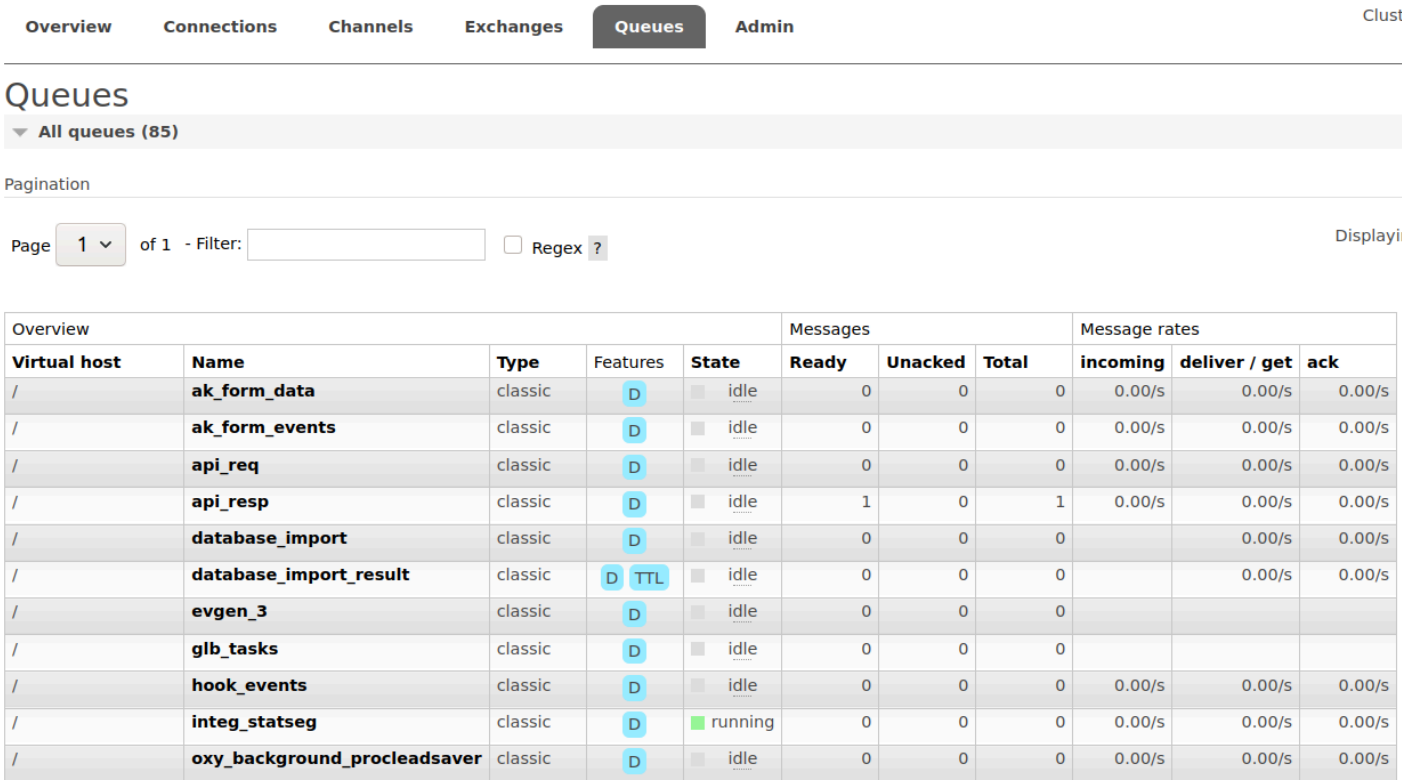
```
                for k, v := range bodyMap {
                        log.Println(k, v)
                }

                body, err := json.Marshal(bodyMap)
                failOnError(err, "Failed to json.Marshal bodyMap")

                err = ch.Publish(
                        "",               // exchange
                        qUeueImport.Name, // routing key
                        false,            // mandatory
                        false,            // immediate
                        amqp.Publishing{
                                ContentType: "text/plain",
                                Body:        []byte(body),
                        })
                failOnError(err, "Failed to publish a message")
        }

        log.Printf("[v] Sended %d requests to %s", amountOfPushMsgs, qUeueImport.Name)
}
```

2. После выполнения скрипта, зайти в RabbitMQ во вкладку **Queues**:



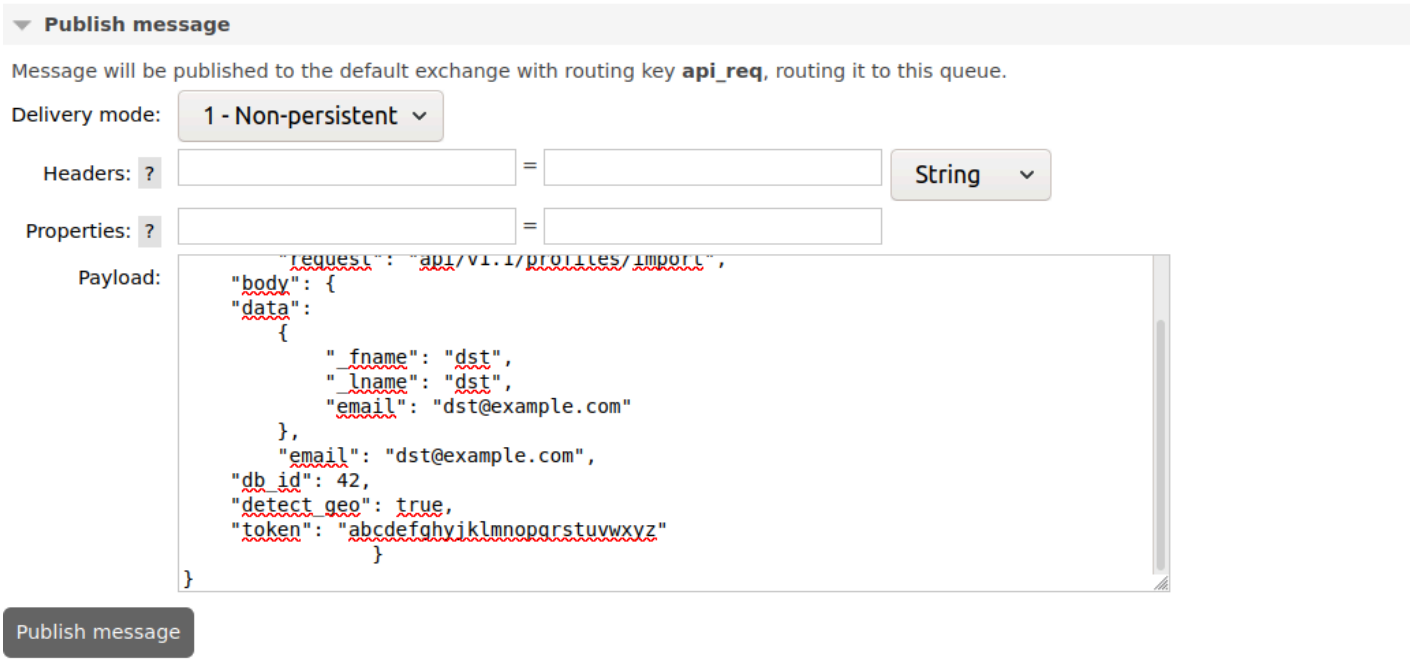
Overview					Messages			Message rates		
Virtual host	**Name**	**Type**	Features	**State**	**Ready**	**Unacked**	**Total**	**incoming**	**deliver / get**	**ack**
/	**ak_form_data**	classic	D	idle	0	0	0	0.00/s	0.00/s	0.00/s
/	**ak_form_events**	classic	D	idle	0	0	0	0.00/s	0.00/s	0.00/s
/	**api_req**	classic	D	idle	0	0	0	0.00/s	0.00/s	0.00/s
/	**api_resp**	classic	D	idle	1	0	1	0.00/s	0.00/s	0.00/s
/	**database_import**	classic	D	idle	0	0	0		0.00/s	0.00/s
/	**database_import_result**	classic	D TTL	idle	0	0	0		0.00/s	0.00/s
/	**evgen_3**	classic	D	idle	0	0	0			
/	**glb_tasks**	classic	D	idle	0	0	0			
/	**hook_events**	classic	D	idle	0	0	0	0.00/s	0.00/s	0.00/s
/	**integ_statseg**	classic	D	running	0	0	0	0.00/s	0.00/s	0.00/s
/	**oxy_background_procleadsaver**	classic	D	idle	0	0	0	0.00/s	0.00/s	0.00/s

3. Выбрать **api_resp** и получить ответ на API запрос:

Пример работы с API через интерфейс RabbitMQ Management Plugin

Послать API запрос можно прямо в RabbitMQ через **api_req**:

1. В RabbitMQ необходимо выбрать вкладку **Queues**. Далее - **api_req** и **Publish message**.



2. После того как запрос будет отправлен, посмотреть его результат можно будет так же, как и в способе №1.

Список гендерных идентификаций

В API-запросах, различных отчетах, а также системных параметрах половая принадлежность может быть отображена разными способами.

Мужской пол может обозначаться на платформе одним из параметров нижеприведенного объекта:

```
{
        "id": 1,
        "title": "Male",
        "values": ["male", "1", "Male", "MASCULINE",
            "M",
            "MAN",
            "♂",
            "М",
            "МУЖ",
            "МУЖСК",
            "МУЖСКОЙ",
            "МУЖЧИНА"
        ]
    }
```

За **женский пол** отвечает любой из параметров следующего кода:

```
    {
        "id": 2,
        "title": "Female",
        "values": ["female", "2", "Female", "FEM",
            "FEMININE",
            "F",
            "WOMAN",
            "W",
            "♀",
            "Ж",
            "ЖЕН",
            "ЖЕНСК",
            "ЖЕНСКИЙ",
            "ЖЕНЩИНА"
        ]
    }
```

TIP

Если вам потребовалось добавить другие обозначения, вы можете сделать это, добавив нужные значения в справочник гендерных идентификаторов. Подробнее об этом можно узнать в данной статье.

Получить допустимые значения полей browsers, devices, tz, oses, languages

Описание

API-метод позволяет получить допустимые значения для полей:

- `browser` (браузер)
- `device` (устройство)
- `tz` (часовой пояс)
- `os` (операционная система)
- `language` (язык)

Эти поля передаются в следующих запросах:

- `/v1.1/resources/fields_get` (получение информации о полях подписки ресурса)
- `/api/v1.1/goals/register` (регистрация события достижения цели)

URL-адрес

Метод: POST

URL адрес: `https://example.com/api/v1.1/dictionary`

Параметры запроса

Параметр	Тип	Пример	Обязательный	Описание
token	string	“abcdefghijklmnqrstuvwxyz”	Да	API-токен
name	string	“browsers” “devices” “tz” “oses” “languages”	Да	Название системного поля, для которого необходимо получить список допустимых значений.

Пример запроса

JSON

```
{
   "token": "abcdefghijklmnqrstuvwxyz",
   "name": "devices"
}
```

XML

```
<xml>
  <token>1</token>
  <name>devices</name>
</xml>
```

Пример ответа

JSON

```
{
  "data": [
    {
      "title": "Desktop",
      "value": "web"
    },
    {
      "title": "Other",
      "value": "other"
    },
    {
      "title": "Mobile",
      "value": "mob"
    }
  ],
  "error": 0,
  "error_text": "Successful operation"
}
```

XML

```
<xml>
  <data>
    <value>web</value>
    <title>Desktop</title>
  </data>
  <data>
    <value>other</value>
    <title>Other</title>
  </data>
  <data>
    <value>mob</value>
    <title>Mobile</title>
  </data>
  <error>0</error>
  <error_text>Successful operation</error_text>
</xml>
```

mSDK

Описание

Altcraft mSDK — это мобильный SDK, который обеспечивает надёжную работу с push-уведомлениями и взаимодействие с платформой Altcraft. Он разработан для Android и iOS, поддерживает Firebase, Huawei, RuStore и APNS, а также предоставляет расширенные возможности по управлению подписками, обработке событий и работе с несколькими пользователями на одном устройстве.

SDK может использовать **JWT-токены** или **rToken**, что позволяет гибко настраивать уровень безопасности и типы матчинга данных профилей. Автоматическая обработка уведомлений и событий доставки обеспечивает стабильность и высокую точность передачи данных, а встроенный WorkManager гарантирует доставку событий даже в случае сетевых сбоев.

Возможности

- Совместим с Kotlin, Java (Android), Swift, Objective-C (iOS), JavaScript (React Native).
- Управление подпиской на push-уведомления: добавление, приостановка, отписка.
- Автоматический показ уведомлений с содержимым, настраиваемым в платформе, с возможностью кастомизации отображения.
- Автоматическое обновление push-токена при его изменении.
- Автоматическая передача событий доставки и открытия уведомлений.
- Поддержка анонимных и зарегистрированных пользователей, а также нескольких пользователей на одном устройстве (при использовании JWT).
- Регистрация мобильных событий.
- Автоматический повтор запросов через WorkManager для максимальной доставки событий.
- Поддержка push-провайдеров: Firebase, Huawei, RuStore (Android), APNS (iOS).
- Использование JWT-токенов для обеспечения безопасности и неограниченных вариантов матчинга профилей.
- Возможность работы через rToken с ограниченным доступом по идентификатору push-подписки.
- Поддержка переопределения формата push-уведомлений и доступа ко всем data-полям сообщений.

GitHub

Репозиторий Altcraft mSDK доступен на **GitHub**:

- **iOS**
- **Android**
- **React Native**

Менеджеры зависимостей

Cocoapods:

```
pod 'Altcraft', '1.0.3'
```

Swift Package Manager:

`1.0.3`

Maven Central:

`com.altcraft:android-sdk:1.0.3`

npm:

`@altcraft/react-native-sdk`

Работа с ролевым и JWT-токеном

Варианты авторизации

JWT-токен

Данный тип авторизации использует JWT-токен, который приложение передаёт в SDK. Токен добавляется в заголовок каждого запроса.

JWT (JSON Web Token) — это строка в формате JSON, содержащая claims (набор данных), подписанных для проверки подлинности и целостности.

Токен формируется и подписывается ключом шифрования на стороне серверной части клиента (ключи шифрования не хранятся в приложении). По запросу SDK, приложение обязано передать полученный с сервера JWT-токен.

Преимущества:

- Повышенная безопасность API-запросов.
- Возможность поиска профилей по любым идентификаторам (email, телефон, custom ID).
- Поддержка нескольких пользователей на одном устройстве.
- Восстановление доступа к профилю после переустановки приложения.
- Идентификация конкретного профиля на разных устройствах.

rToken

Альтернативный способ авторизации — использование ролевого токена (**rToken**), переданного в параметры конфигурации SDK. При этом методе авторизации запросы содержат заголовок с ролевым токеном.

Особенности:

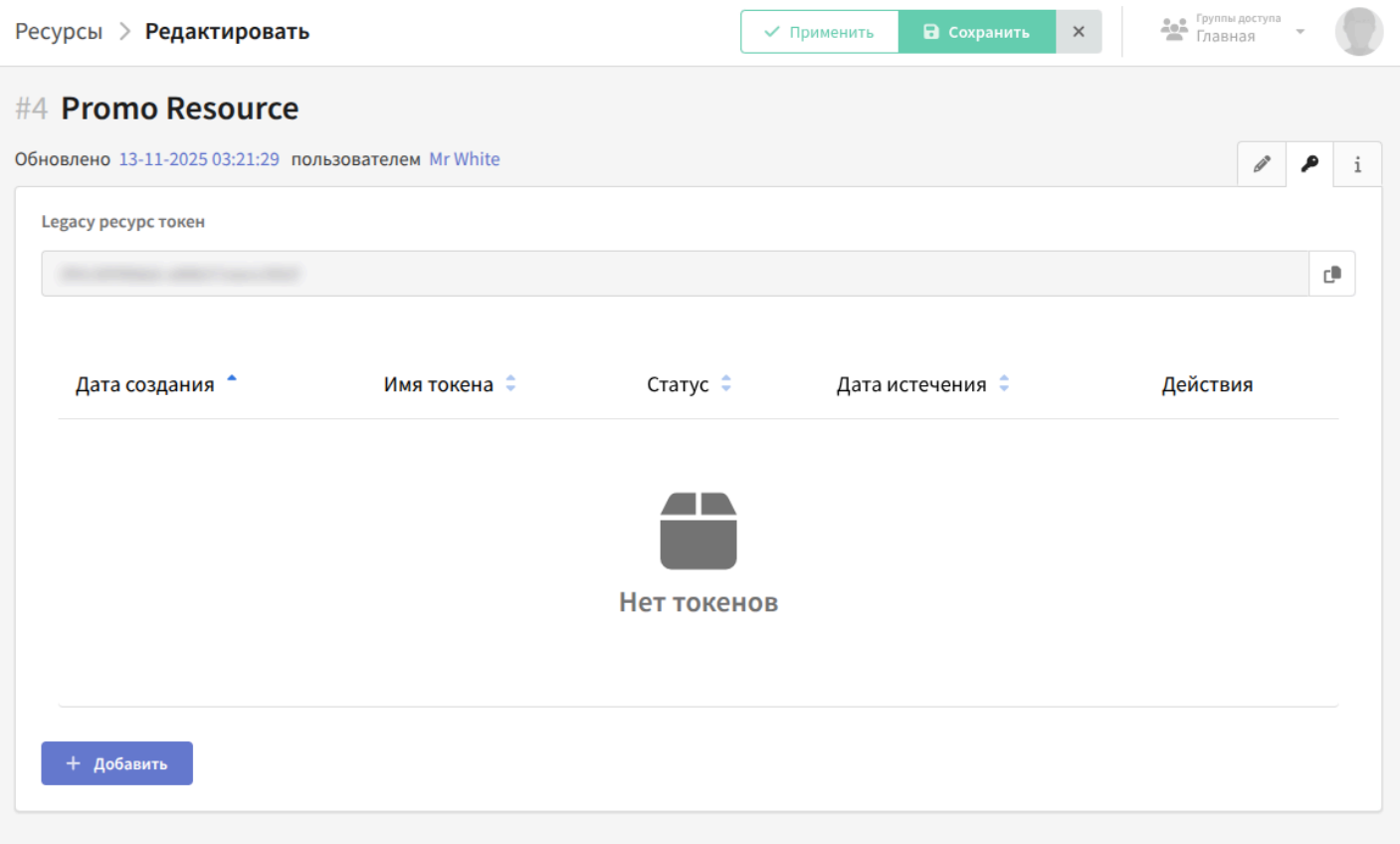
- Поиск профилей возможен только по push-токену устройства (например, FCM).
- Если push-токен изменился и не был передан на сервер (например, после удаления и переустановки приложения), связь с профилем будет потеряна, в результате будет создан новый профиль.

Ограничения:

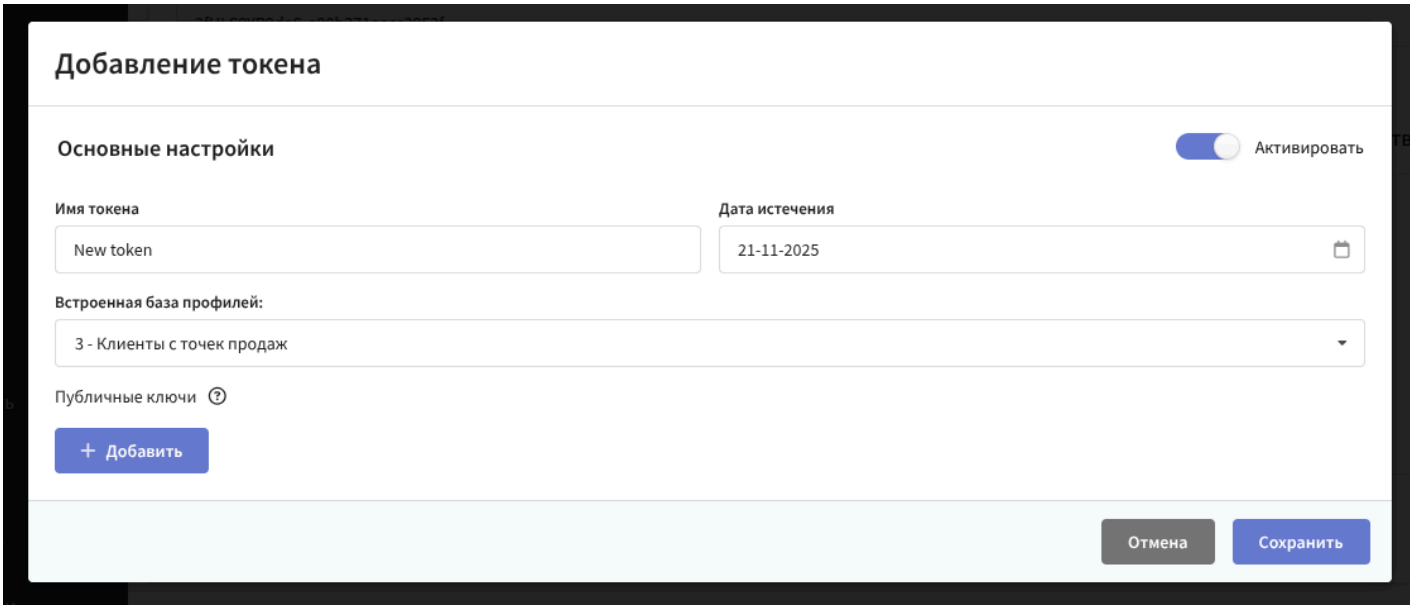
- Потеря связи с профилем при изменении push-токена, которое не было зафиксировано на сервере Altcraft.
- Отсутствие возможности использовать приложение для разных профилей на одном устройстве.
- Невозможность регистрации одного пользователя на другом устройстве.

Настройка ролевого токена и сервиса предоставления JWT

После создания ресурса в его настройках будет доступен раздел с управлением токенами:



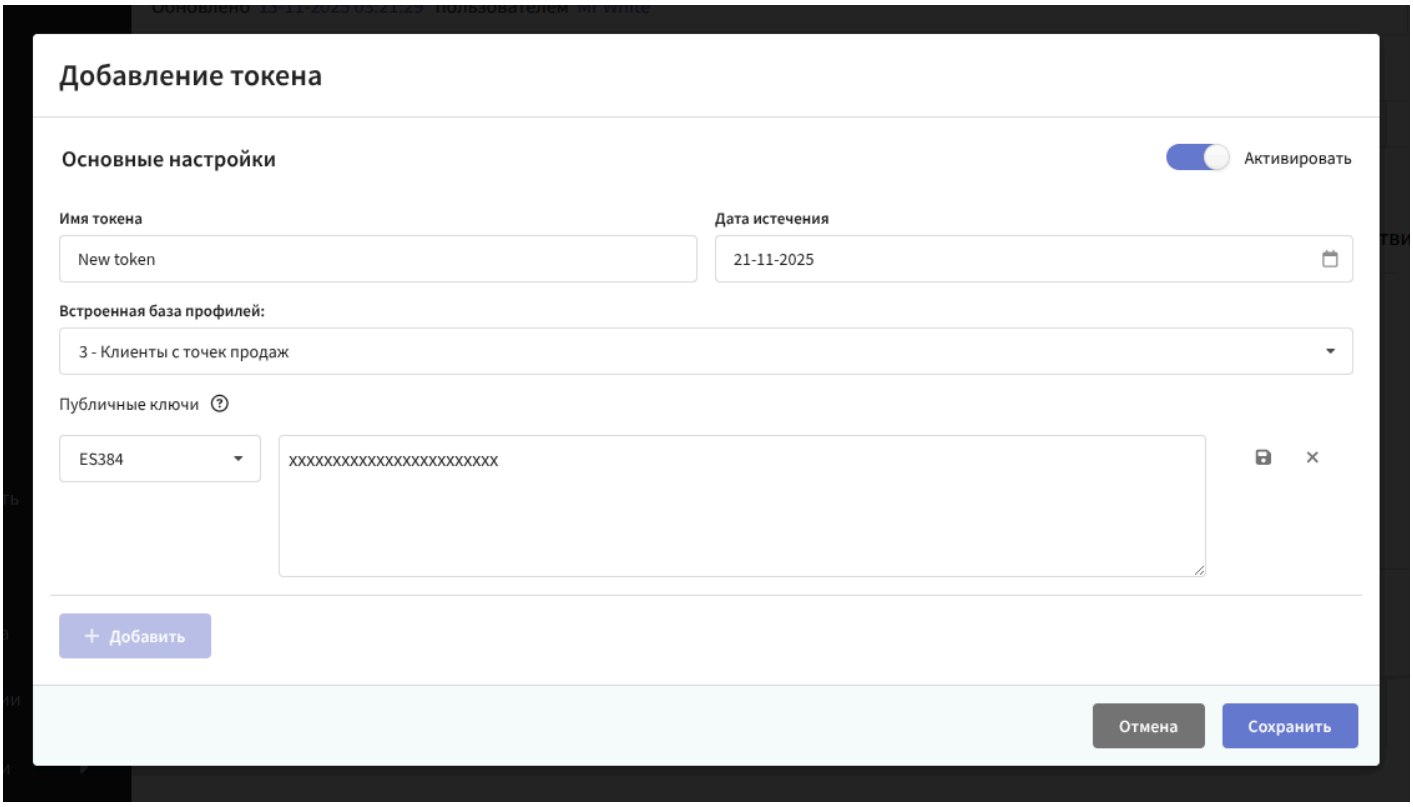
Для создания **ролевого** токена не требуется добавлять публичный ключ, просто укажите его название, дату истечения и привязанную к нему базу профилей:



Ролевой токен выступает ключом-доступа со стороны mSDK к определённой связке “ресурс-база”. В рамках авторизации с помощью данного токена будут доступны следующие действия в платформе:

- Регистрация событий
- Обновление полей профиля
- Импорт профилей

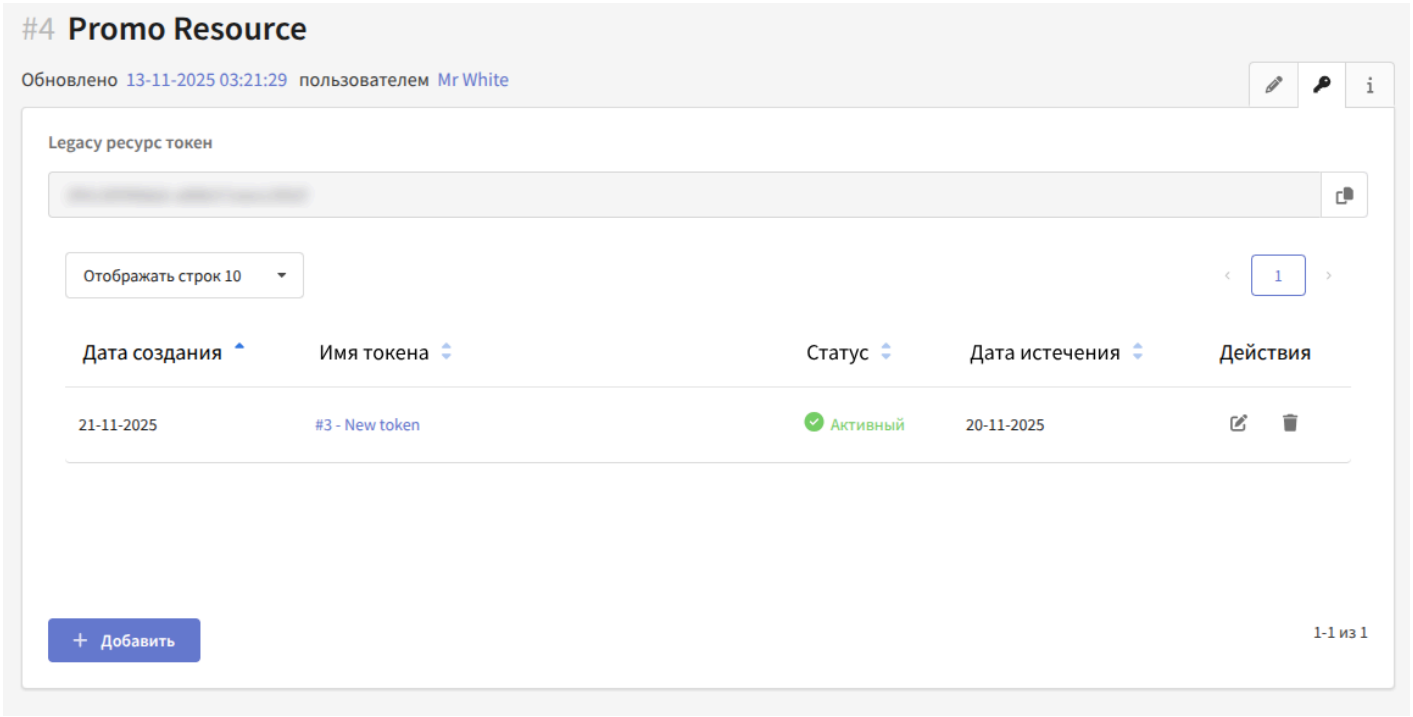
Для создания **JWT-токена** понадобится предоставить публичный ключ. Платформа допускает использование алгоритма ES384 (ECDSA, как наиболее надежный), но также можно использовать RS256, ES256, ES512 для совместимости с библиотеками разных приложений.



TIP: Генерация ключа

Для примера сгенерируем ключ алгоритмом ES384: `openssl ecparam -name secp384r1 -genkey -noout -out private.ec.key` `openssl ec -in private.ec.key -pubout -out public.pem` Будут сгенерированы файлы с приватным и публичным ключом. Для вставки в платформу нужен **публичный** ключ.

После создания токена, его можно скопировать и использовать в дальнейшем:



Авторизация запросов mSDK с помощью ролевого токена

rToken, при использовании его в авторизации, предоставляет доступ к операциям в рамках конкретного ресурса. Однако, из-за открытого формата и неизменяемого значения, вводится ограничение на используемый матчинг профилей.

Так, при использовании ролевого токена, профили ищутся по набору query-параметров. В текущей реализации backend ожидает следующие query-параметры при использовании rToken:

- **provider** — провайдер мобильных уведомлений
- **subscription_id** — токен устройства, предоставляемый провайдером

Именно по идентификатору подписки будет производиться поиск профиля в рамках БД, закреплённых за ресурсом.

Особенности

- Поиск профилей возможен только по push-токену устройства (например, FCM).
- При импорте профиля через push-подписку, он помечается как “временный”.
- При регистрации мобильных событий, привязка к профилю (запись в историю событий профиля) невозможна.
- Если push-токен изменился и не был передан на сервер (например, после удаления и переустановки приложения), связь с профилем будет потеряна и будет создан новый профиль.

Использование

При использовании библиотеки `Altcraft SDK` токен указывается как параметр `rToken` класса конфигурации `AltcraftConfiguration`.

В дальнейшем, библиотека предоставляет его в качестве заголовка авторизации.

Авторизация запросов mSDK с помощью JWT-токена

Для безопасной авторизации в платформе рекомендуется использовать JWT-токен для авторизации запросов. Токен должен быть подписан парным ключом, который добавлялся ранее, на этапе создания токена.

Особенности

- Повышенная безопасность API-запросов. JWT выступает защитной оберткой ролевого токена, позволяя избежать неавторизованных действий. Так, ролевой токен зашивается в payload JWT-токена и лишь по предоставленным ранее (в настройках ресурса) ключам, платформа авторизует действия mSDK.
- Возможность поиска профилей по любым идентификаторам (email, телефон, custom ID), в соответствии с матчингом.
- Поддержка нескольких пользователей на одном устройстве.
- Идентификация конкретного профиля на разных устройствах.
- Восстановление доступа к профилю после переустановки приложения.

Использование

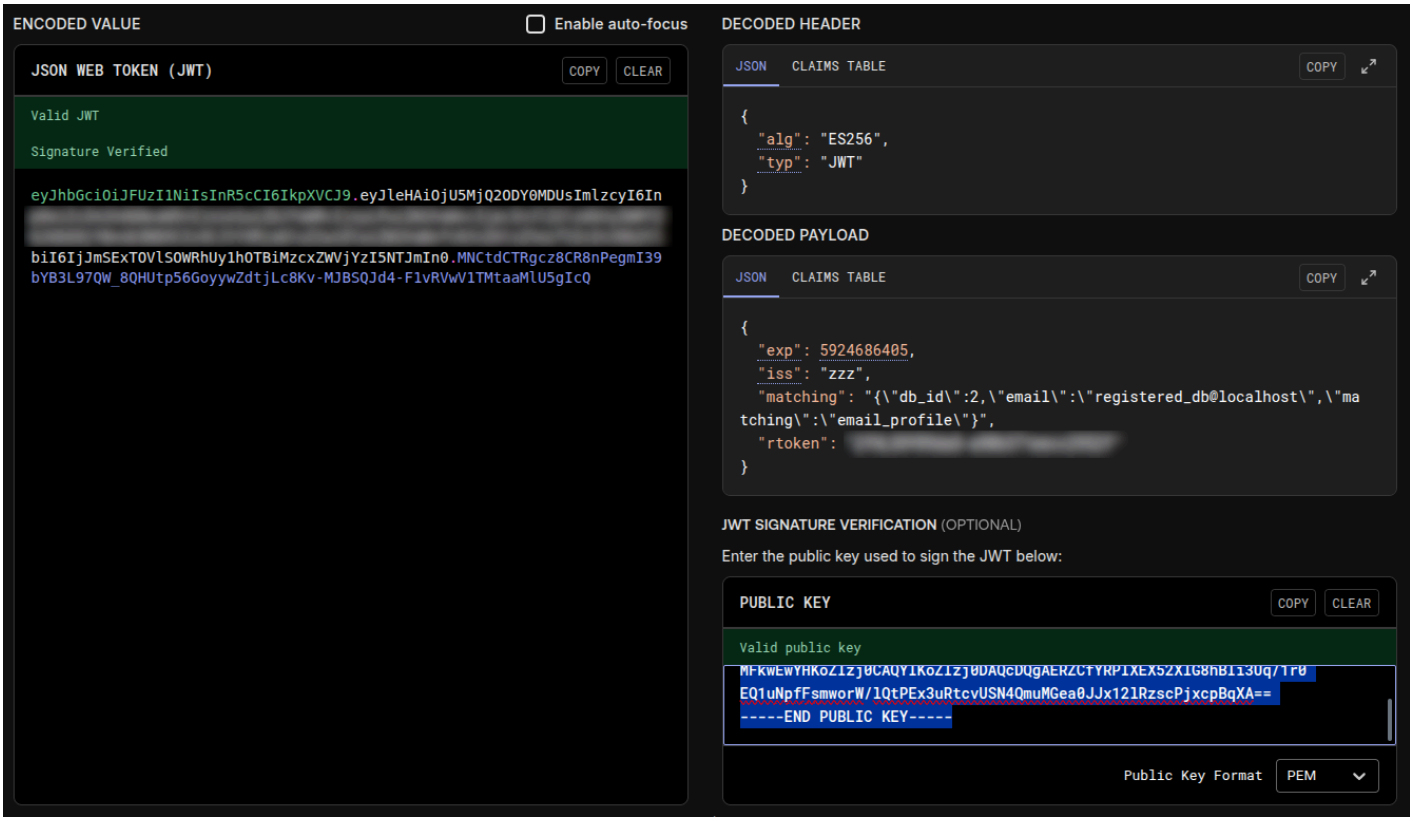
В качестве полезной нагрузки JWT, платформа ожидает следующую структуру:

```
{
"iss": "<App Name>",
"exp": <UnixTimeUTC>,
"rtoken": "<RoleToken>",
"matching": "JSONString",
}
```

- `iss` — `issuer` — уникальный идентификатор создателя токена
- `exp` — `expiration time` — время истечения токена в формате UNIX timestamp **в секундах**
- `rtoken` — ролевой токен, полученный при настройке ресурса в платформе
- `matching` — `JSONString` — сериализованный в строку объект, составленный в соответствии с документацией. Например: `{"db_id":2,"email":"registered_db@localhost","matching":"email_profile"}`.

При использовании библиотеки `Altcraft SDK` передаётся как имплементация интерфейса авторизации `JWTInterface`. В дальнейшем, библиотека также предоставляет его в качестве заголовка авторизации.

Пример того, как может выглядеть итоговый JWT:



Быстрый старт

В статье приведен пример быстрой интеграции Altcraft mSDK с Firebase Cloud Messaging (FCM). После выполнения указанных шагов ваше приложение сможет получать push-уведомления Altcraft.

Для использования Huawei Mobile Services или RuStore необходимо добавить соответствующие SDK в проект и реализовать их интерфейсы, используя тот же подход, что показан для Firebase.

Шаг 0. Предварительные условия

- SDK push-провайдеров интегрированы в проект приложения (см. инструкции по интеграции push провайдеров);
- В приложение добавлен класс, расширяющий `Application()`;
- Все необходимые сущности (ресурс, рассылка, шаблон, база профилей) заранее созданы в платформе Altcraft;
- Для отправки push-уведомлений на устройствах с Android 13 (Tiramisu) требуется явное разрешение пользователя.

Шаг 1. Подключение Altcraft mSDK к приложению

- Добавьте зависимость библиотеки в файл `build.gradle.kts` уровня приложения (app level).

```
    dependencies {
        implementation("com.altcraft:android-sdk:1.0.3")
    }
```

- Выполните синхронизацию изменений Gradle.

Шаг 2. Реализация JWTInterface

Выполните реализацию интерфейса SDK, предназначенного для предоставления **JWT-токена**.

В модуле приложения создайте класс, реализующий `JWTInterface`, и переопределите `getJWT()`:

```
class JWTProvider(private val context: Context /** добавьте свойство context если это
необходимо*/) : JWTInterface {
    override fun getJWT(): String?{
      // ваш код, возвращающий актуальный JWT-токен
    }
}
```

TIP: Обратите внимание

- `getJWT()` — синхронная функция. Поток выполнения SDK будет приостановлен до получения JWT;

- Рекомендуется, чтобы `getJWT()` возвращал значение немедленно из кэша (`in-memory`, `SharedPreferences` или `EncryptedSharedPreferences`). Это ускорит выполнение запросов;
- Желательно подготовить актуальный JWT как можно раньше, например, на старте приложения, и сохранить его в кэш, чтобы при обращении SDK токен был доступен без задержек. При отсутствии значения допустимо вернуть `null`.

Пример реализации с использованием suspend-функции получения токена

```
class JWTProvider(private val context: Context /** добавьте свойство context если это
необходимо*/) : JWTInterface {
    private suspend fun fetchJwt(context: Context): String? {
        // ваш код, возвращающий актуальный JWT-токен
        return jwt
    }

    override fun getJWT(): String? = runBlocking {
        fetchJwt(context)
    }
}
```

Шаг 3. Регистрация провайдера JWT

Зарегистрируйте провайдера JWT в `Application.onCreate()`:

```
class App : Application() {
    override fun onCreate() {
        super.onCreate()
        AltcraftSDK.setJWTProvider(JWTProvider(/** добавьте свойство context если это
необходимо*/))
    }
}
```

Шаг 4. Реализация интерфейса FCM

В модуле приложения создайте класс, реализующий `FCMInterface`:

Пример реализации

```
/**
 * FCM implementation for managing Firebase Cloud Messaging tokens.
 *
 * Provides methods to retrieve and delete FCM tokens via the Firebase SDK.
 */
class FCMProvider : FCMInterface {

    /**
     * Retrieves the current FCM token.
     *
     * Returns `null` if an error occurs during retrieval.
     *
     * @return The FCM token or `null` on failure.
     */
    override suspend fun getToken(): String? {
        return try {
            Firebase.messaging.token.await()
        } catch (e: Exception) {
            null
        }
    }

    /**
     * Deletes the current FCM token.
     *
     * Invokes [completion] with `true` if successful, `false` otherwise.
     *
     * @param completion Callback with the result of the deletion.
     */
    override suspend fun deleteToken(completion: (Boolean) -> Unit) {
        try {
            FirebaseMessaging.getInstance().deleteToken().addOnCompleteListener {
                completion(it.isSuccessful)
            }
        } catch (e: Exception) {
            completion(false)
        }
    }
}
```

Шаг 5. Создание сервиса, расширяющего `FirebaseMessagingService()`

В модуле приложения создайте класс, расширяющий `FirebaseMessagingService`, и реализуйте в нём метод `onMessageReceived()` с вызовом `AltcraftSDK.PushReceiver.takePush()` для передачи данных push-уведомления в SDK:

Пример создания сервиса

```
/**
 * FCM service for handling push tokens and messages.
 */
class FCMService : FirebaseMessagingService() {

    /**
     * Called when a new FCM token is generated.
     *
     * @param token The new FCM token.
     */
    override fun onNewToken(token: String) {
        super.onNewToken(token)
    }

    /**
     * Called when a push message is received.
     *
     * Forwards the message to all receivers with additional metadata.
     *
     * @param message The received [RemoteMessage].
     */
    override fun onMessageReceived(message: RemoteMessage) {
        super.onMessageReceived(message)

       AltcraftSDK.PushReceiver.takePush(this@FCMService, message.data)
    }
}
```

Шаг 6. Регистрация сервиса FCM в манифесте приложения

Зарегистрируйте сервис Firebase Cloud Messaging в `AndroidManifest.xml` приложения:

```
<!-- FCM service -->
<service
    android:name="<your_package_name>.FCMService"
    android:exported="false">
    <intent-filter>
    </intent-filter>
</service>
```

Шаг 7. Регистрация провайдера FCM в `Application.onCreate()`

Зарегистрируйте провайдера Firebase Cloud Messaging в `Application.onCreate()`

```
class App : Application() {
    override fun onCreate() {
        super.onCreate()
        AltcraftSDK.pushTokenFunctions.setFCMTokenProvider(FCMProvider())
    }
}
```

Шаг 8. Инициализация SDK

В `Application.onCreate()` создайте переменную конфигурации и выполните инициализацию с помощью функции `initialization()`.

Обратите внимание — функция `initialization()` должна выполняться **после регистрации** всех используемых провайдеров (и JWT-провайдера, и провайдеров push-токенов).

Подробнее об инициализации

```
AltcraftSDK
 // Инициализация SDK и установка конфигурации
  └── fun initialization(context: Context, configuration: AltcraftConfiguration, complete:
((Result<Unit>) -> Unit)? = null): Unit
```

`fun initialization(context: Context, configuration: AltcraftConfiguration, complete: ((Result<Unit>) -> Unit)? = null): Unit`

Пример инициализации:

```
class App : Application() {
    override fun onCreate() {
        super.onCreate()
        AltcraftSDK.setJWTProvider(JWTProvider(applicationContext))
        AltcraftSDK.pushTokenFunctions.setFCMTokenProvider(FCMProvider())

        val config = AltcraftConfiguration.Builder(
            apiUrl = "https://pxl-your-instance.altcraft.com",
            icon = R.drawable.your_icon
        ).build()

        // установите необходимые параметры конфигурации (см. README Android MSDKv2)

        AltcraftSDK.initialization(context = this, configuration = config)
    }
}
```

Шаг 9. Выполнение подписки на push-уведомления

С помощью функции `pushSubscribe()` выполните подписку на push-уведомления в любом удобном для вас месте кода, например, после предоставления разрешения на отправку уведомлений:

```
AltcraftSDK
└──  val pushSubscriptionFunctions: PublicPushSubscriptionFunctions
    // Подписка на push-уведомления (status = SUBSCRIBED)
    ├─ fun pushSubscribe(
         context: Context,
         sync: Boolean = true,
         profileFields: Map<String, Any?>? = null,
         customFields: Map<String, Any?>? = null,
         cats: List<DataClasses.CategoryData>? = null,
         replace: Boolean? = null,
         skipTriggers: Boolean? = null
       ): Unit
```

Пример выполнения подписки:

```
class MainActivity : ComponentActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        NotificationPermissionHandler.isGranted(
            activity = this,
            onGranted = {
                if(isUnsubscribed()){
                   AltcraftSDK.pushSubscriptionFunctions.pushSubscribe(context = this)
                }
            }
        )
    }
}
```

Конфигурация SDK

Предварительные условия

- SDK провайдеров push-уведомлений интегрированы в проект приложения (см. инструкции по интеграции push-провайдеров).
- В приложении добавлен класс, расширяющий Application.
- В проекте должен быть подключён репозиторий Maven Central.
- Добавлена зависимость библиотеки в файл `build.gradle.kts` уровня приложения (app level):

```
    dependencies {
        implementation("com.altcraft:android-sdk:1.0.3")
    }
```

Подготовка приложения

Настройка JWT-интерфейса (опционально)

JWTInterface — интерфейс запроса JWT-токена. Предоставляет актуальный JWT-токен из приложения по запросу SDK. Реализация данного интерфейса требуется, если используется JWT-аутентификация API-запросов. JWT подтверждает, что пользовательские идентификаторы аутентифицированы приложением.

Реализация JWT-аутентификации обязательна, если используется тип матчинга отличный от push-данных из подписки (например, идентификатор пользователя — email или телефон).

Интерфейс SDK:

```
interface JWTInterface {
    fun getJWT(): String?
}
```

Реализация на стороне приложения

```
class JWTProvider(): JWTInterface {
    override fun getJWT(): String? {
        // код, возвращающий JWT-токен
    }
}
```

Регистрация провайдера в `Application.onCreate()`

```
class App : Application() {
    override fun onCreate() {
        super.onCreate()
        AltcraftSDK.setJWTProvider(JWTProvider(applicationContext))
    }
}
```

TIP: К сведению

`getJWT()` — синхронная функция. Поток выполнения SDK будет приостановлен до получения JWT. Рекомендуется, чтобы `getJWT()` возвращал значение немедленно — из кэша (in-memory, `SharedPreferences` или `EncryptedSharedPreferences`) — это ускорит выполнение запросов. Желательно подготовить актуальный JWT как можно раньше (на старте приложения) и сохранить его в кэш, чтобы при обращении SDK токен был доступен без задержек. При отсутствии значения допустимо вернуть `null`.

Подготовка к работе с push-провайдерами

Для предварительной подготовки приложения к работе в провайдерами push, необходимо:

- Интегрировать SDK push-провайдеров в проект;
- Реализовать интерфейсы SDK для передачи и удаления push-токена;
- Расширить сервисы push-провайдеров;
- Зарегистрировать сервис push-провайдера в `AndroidManifest.xml`.

Firebase Cloud Messaging

Интеграция FCM SDK в проект

Статья по интеграции FCM доступна здесь.

Реализация интерфейса Altcraft SDK

FCMInterface — интерфейс запроса и удаления push-токена FCM:

- Передает токен FCM в SDK Altcraft
- Удаляет токен FCM по запросу SDK Altcraft

Интерфейс SDK:

```
interface FCMInterface {
    suspend fun getToken(): String?
    fun deleteToken(completion: (Boolean) -> Unit)
}
```

▶ Рекомендуемая реализация на стороне приложения

Реализация сервиса провайдера

Рекомендуемая реализация `FCMService` с передачей уведомления в функцию `takePush()`:

```
/**
 * FCM service for handling push tokens and messages.
 */
class FCMService : FirebaseMessagingService() {

    /**
     * Called when a new FCM token is generated.
     *
     * @param token The new FCM token.
     */
    override fun onNewToken(token: String) {
        super.onNewToken(token)
    }

    /**
     * Called when a push message is received.
     *
     * @param message The received [RemoteMessage].
     */
    override fun onMessageReceived(message: RemoteMessage) {
        super.onMessageReceived(message)

       AltcraftSDK.PushReceiver.takePush(this@FCMService, message.data)
    }
}
```

Регистрация сервиса push-провайдера в `AndroidManifest.xml`

```
<service
    android:name="<your_package_name>.FCMService"
    android:exported="false">
    <intent-filter>
    </intent-filter>
</service>
```

Huawei Mobile Services

Интеграция HMS SDK в проект

Статья по интеграции HMS доступна здесь.

Реализация интерфейса Altcraft SDK

HMSInterface — интерфейс запроса и удаления push-токена HMS.

Интерфейс SDK:

```
interface HMSInterface {
    suspend fun getToken(context: Context): String?
    fun deleteToken(context: Context, complete: (Boolean) -> Unit)
}
```

▶ Рекомендуемая реализация на стороне приложения

Реализация сервиса провайдера

Рекомендуемая реализация `HMSService` с передачей уведомления в функцию `takePush()`:

```
/**
 * HMS service for handling push tokens and incoming notifications.
 *
 * Extends [HmsMessageService] and overrides key HMS callback methods.
 */
class HMSService : HmsMessageService() {
    /**
     * Called when a new HMS token is generated.
     *
     * @param token The new HMS token.
     */
    override fun onNewToken(token: String) {
        super.onNewToken(token)
    }

    /**
     * Called when a push message is received from HMS.
     *
     * Forwards the message with additional metadata to all receivers.
     *
     * @param message The received [RemoteMessage].
     */
    override fun onMessageReceived(message: RemoteMessage) {
        AltcraftSDK.PushReceiver.takePush(this@HMSService, message.dataOfMap)
    }
}
```

Регистрация сервиса push-провайдера в `AndroidManifest.xml`

```
<service
    android:name="<your_package_name>.HMSService"
    android:exported="false">
    <intent-filter>
    </intent-filter>
</service>
```

RuStore

Интеграция RuStore SDK в проект

Статья по интеграции RuStore доступна здесь.

Реализация интерфейса Altcraft SDK

RustoreInterface — интерфейс запроса и удаления push-токена RuStore.

Интерфейс SDK:

```
interface RustoreInterface {
    suspend fun getToken(): String?
    fun deleteToken(complete: (Boolean) -> Unit)
}
```

▶ Рекомендуемая реализация на стороне приложения

Реализация сервиса провайдера

Рекомендуемая реализация `RuStoreService` с передачей уведомления в функцию `takePush()`:

```
/**
 * RuStore service for handling push notifications.
 *
 * Extends [RuStoreMessagingService] and overrides key callbacks.
 */
class RuStoreService : RuStoreMessagingService() {
    override fun onNewToken(token: String) {
        super.onNewToken(token)
    }

    /**
     * Called when a push message is received.
     *
     * Forwards the message to all receivers with added metadata.
     *
     * @param message The received RemoteMessage.
     */
    override fun onMessageReceived(message: RemoteMessage) {
        AltcraftSDK.PushReceiver.takePush(this, message.data)
    }
}
```

Регистрация сервиса push-провайдера в `AndroidManifest.xml`

```
<service
    android:name="<your_package_name>.RuStoreService"
    android:exported="true"
    tools:ignore="ExportedService">
    <intent-filter>
    </intent-filter>
</service>
```

Инициализация SDK

Параметры инициализации

Для передачи параметров конфигурации используется класс `AltcraftConfiguration`:

```
class AltcraftConfiguration private constructor(
    private val apiUrl: String,
    private val icon: Int? = null,
    private val rToken: String? = null,
    private val appInfo: DataClasses.AppInfo? = null,
    private val providerPriorityList: List<String>? = null,
    private val enableLogging: Boolean? = null,
    private val pushReceiverModules: List<String>? = null,
    private val pushChannelName: String? = null,
    private val pushChannelDescription: String? = null
)
```

Описание параметров:

apiUrl `String`

Обязательный: **Да**
Описание: URL конечной точки Altcraft API

---

icon `Int?`

По умолчанию: `null`
Обязательный: Нет
Описание: Идентификатор ресурса `drawable`, используемого как иконка уведомлений

---

rToken `String?`

По умолчанию: `null`
Обязательный: Нет
Описание: Ролевой токен Altcraft (идентифицирует ресурс, БД или аккаунт). Используется, если единственный тип матчинга — push-токен устройства, выданный провайдером.

---

appInfo `DataClasses.AppInfo?`

По умолчанию: `null`
Обязательный: Нет
Описание: Базовые метаданные приложения для Firebase Analytics. Для установки используйте публичный data-класс SDK `AppInfo` (пакет `com.altcraft.sdk.data.DataClasses`):

```
  data class AppInfo(
      /** Firebase app_id */
      val appID: String,
      /** Firebase app_instance_id */
      val appIID: String,
      /** Firebase app_version */
      val appVer: String
  )
```

---

enableLogging `Boolean`

По умолчанию: `null`
Обязательный: Нет
Описание: Включает/отключает ведение логов. Логи можно найти по logCat-тегу `altcraft_lib`.

---

providerPriorityList `List<String>?`

По умолчанию: `null`
Обязательный: Нет
Описание: Список приоритетов провайдеров. Используется для автоматического обновления push-токена подписки, если токен более приоритетного провайдера недоступен. Приоритет определяется индексом в списке: элемент с индексом **0** — самый приоритетный.

Пример использования:

```
providerPriorityList = listOf(
    "android-firebase",
    "android-huawei",
    "android-rustore"
)
```

При такой настройке:

- SDK сначала запросит токен **FCM**; если FCM недоступен — **HMS**; если HMS недоступен — **RuStore**;
- Работает при условии, если в приложении реализованы интерфейсы соответствующих провайдеров.

Для удобства установки значения параметра SDK содержит публичные константы:

```
const val FCM_PROVIDER: String = "android-firebase"
const val HMS_PROVIDER: String = "android-huawei"
const val RUS_PROVIDER: String = "android-rustore"
```

Если значение не задано, приоритет провайдеров следующий:

```
FCM_PROVIDER –> HMS_PROVIDER –> RUS_PROVIDER
```

Значение параметра может содержать **один элемент** — в этом случае будет использоваться только один провайдер, независимо от доступности push-токена.

Параметр можно не указывать, если:

- в проекте используется только один провайдер;
- приоритет по умолчанию соответствует требованиям.

TIP: Кейсы использования

Кейс 1. Приоритет `RUS_PROVIDER -> FCM_PROVIDER`. Пользователь удаляет RuStore — уведомления RuStore недоступны. При рекомендованной реализации `RustoreInterface.getToken()` вернёт `null`, SDK автоматически переключится на **FCM**, обновит токен подписки, коммуникации сохранятся.

Кейс 2. Параметр не задан, действуют значения по умолчанию `FCM_PROVIDER -> HMS_PROVIDER -> RUS_PROVIDER`. На устройстве Huawei без Google-сервисов SDK автоматически перейдёт на **HMS** без дополнительного кода.

---

pushReceiverModules `List<String>?`

По умолчанию: `null`
Обязательный: Нет
Описание: Список имён пакетов, в которых находятся реализации `AltcraftPushReceiver : AltcraftSDK.PushReceiver()`.

Если указан, SDK обнаружит эти классы и передаст им входящее уведомление (через `pushHandler(context: Context, message: Map<String, String>)`). Если классы не найдены ни в одном из пакетов, уведомление будет показано средствами SDK.

Пример создания класса `AltcraftPushReceiver`

```
package com.altcraft.altcraftmobile.test

@Keep
class AltcraftPushReceiver: AltcraftSDK.PushReceiver() {
    override fun pushHandler(context: Context, message: Map<String, String>) {

        // обработать уведомление с помощью функций SDK
        super.pushHandler(context, message)

        // дополнительная логика обработки уведомлений
    }
}
```

TIP: Кейс использования

Кейс: Приложение содержит несколько модулей, в которых необходимы данные входящих push-уведомлений Altcraft. В каждом из этих модулей можно создать класс `AltcraftPushReceiver: AltcraftSDK.PushReceiver()` и получить входящее push-уведомление как `message: Map<String, String>`.

Обратите внимание, что класс обязательно должен называться `AltcraftPushReceiver`.

Если в проекте есть только один класс `AltcraftPushReceiver`, внутри метода `pushHandler` необходимо вызвать `super.pushHandler(context, message)`, чтобы уведомление корректно обработалось средствами SDK. При наличии нескольких таких классов вызов `super.pushHandler` нужно оставить лишь в одном из

них — иначе каждое уведомление будет продублировано. Если же вы хотите полностью контролировать обработку push-уведомлений самостоятельно, вызов `super.pushHandler` можно опустить.

---

pushChannelName `String?`

По умолчанию: `null`
Обязательный: Нет
Описание: Базовое имя канала push-уведомлений (видно в настройках Android). В зависимости от настроек звука и вибрации push-шаблона в платформе Altcraft к имени добавляется суффикс:

- `allSignal` — звук и вибрация включены;
- `soundless` — бесшумный канал;
- `onlySound` — только звук (без вибрации).
  Пример: при `pushChannelName = "Altcraft"` и режиме `allSignal` видимое имя канала — `"Altcraft_allSignal"`. Если параметр не указан — SDK использует имена по умолчанию: `"allSignal"`, `"soundless"`, `"onlySound"`.

---

pushChannelDescription `String?`

По умолчанию: `null`
Обязательный: Нет
Описание: Описание push-канала (видно в настройках Android). К описанию добавляется режим уведомления:

- `Vibration and sound enabled`;
- `Vibration and sound disabled`;
- `Sound enabled, vibration disabled`. **Пример:** `"Altcraft notification channel. (vibration and sound enabled)"`. Если параметр не указан — SDK использует значения по умолчанию:
- **icon** — `altcraft-sdk/src/main/res/drawable/icon.xml` (если icon = null);
- **providerPriorityList** — `FCM_PROVIDER -> HMS_PROVIDER -> RUS_PROVIDER` (если null);
- **pushChannelName** — `"allSignal", "soundless", "onlySound"` (если null);
- **pushChannelDescription** — `"Vibration and sound enabled", "Vibration and sound disabled", "Sound enabled, vibration disabled"` (если null).

Выполнение инициализации

> **TIP: **Обратите внимание****
>
> Вызывайте `AltcraftSDK.initialization(...)` тогда, когда это необходимо, но только **после** регистрации всех провайдеров (JWT-провайдера и провайдеров push-токенов). Запросы следует выполнять **после** установки конфигурации.

Для инициализации SDK используется функция `fun initialization(context: Context, configuration: AltcraftConfiguration, complete: ((Result<Unit>) -> Unit)? = null)`:

```
// Инициализация SDK и настройка конфигурации

AltcraftSDK.initialization(
    context = context,
    configuration = config,
    complete = null // опционально
)
```

Пример правильного порядка инициализации в `Application.onCreate()` (после регистрации провайдеров)

```
class App : Application() {
    override fun onCreate() {
        super.onCreate()

        // Инициализация SDK поставщика Rustore (если используется)
        RuStorePushClient.init(this, "rustore-project-id-1234")

        // Регистрация поставщика перед инициализацией SDK
        AltcraftSDK.setJWTProvider(JWTProvider(applicationContext))
        AltcraftSDK.pushTokenFunctions.setFCMTokenProvider(FCMProvider())
        AltcraftSDK.pushTokenFunctions.setHMSTokenProvider(HMSProvider())
        AltcraftSDK.pushTokenFunctions.setRuStoreTokenProvider(RuStoreProvider())

        // Конфигурация SDK
        val config = AltcraftConfiguration.Builder(
            apiUrl = "https://pxl-example.altcraft.com",
            icon   = R.drawable.ic_notification
        ).build()

        // Инициализация
        AltcraftSDK.initialization(this, config)
    }
}
```

Пример минимальной рабочей настройки

```
val config = AltcraftConfiguration.Builder(
    apiUrl = "https://pxl-example.altcraft.com"
).build()

AltcraftSDK.initialization(context, config)
```

Пример настройки всех параметров

```
val config = AltcraftConfiguration.Builder(
    apiUrl = "https://pxl-example.altcraft.com",
    icon   = R.drawable.ic_notification,
    rToken = null,
    appInfo = DataClasses.AppInfo(
        appID  = "com.example.app",
        appIID = "8b91f3a0-1111-2222-3333-c1a2c1a2c1a2",
        appVer = "1.0.0"
    ),
    providerPriorityList = listOf(
        FCM_PROVIDER, // "android-firebase"
        HMS_PROVIDER, // "android-huawei"
        RUS_PROVIDER  // "android-rustore"
    ),
    pushReceiverModules = listOf(
        context.packageName,
        "com.example.push_receiver",
        "com.example.feature.test"
    ),
    pushChannelName        = "Altcraft",
    pushChannelDescription = "Altcraft notifications channel"
).build()

AltcraftSDK.initialization(context, config)
```

Пример инициализации с callback завершения

```
AltcraftSDK.initialization(context, config) { result ->
    when {
        result.isSuccess -> {
            // действия в случае успешной инициализации
        }
        result.isFailure -> {
            // обработка ошибок инициализации
        }
    }
}
```

Функционал SDK

TIP

Предварительно вам необходимо настроить SDK для работы с вашим приложением. Подробная инструкция находится здесь

Работа со статусами подписки

Изменение статуса подписки

```
AltcraftSDK
└─ val pushSubscriptionFunctions: PublicPushSubscriptionFunctions
   // Подписка (статус = SUBSCRIBED)
   ├─ fun pushSubscribe(
   │     context: Context,
   │     sync: Boolean = true,
   │     profileFields: Map<String, Any?>? = null,
   │     customFields: Map<String, Any?>? = null,
   │     cats: List<DataClasses.CategoryData>? = null,
   │     replace: Boolean? = null,
   │     skipTriggers: Boolean? = null
   │   ): Unit
   // Приостановка (статус = SUSPENDED)
   ├─ fun pushSuspend(
   │     context: Context,
   │     sync: Boolean = true,
   │     profileFields: Map<String, Any?>? = null,
   │     customFields: Map<String, Any?>? = null,
   │     cats: List<DataClasses.CategoryData>? = null,
   │     replace: Boolean? = null,
   │     skipTriggers: Boolean? = null
   │   ): Unit
   // Отписка (статус = UNSUBSCRIBED)
   └─ fun pushUnSubscribe(
         context: Context,
         sync: Boolean = true,
         profileFields: Map<String, Any?>? = null,
         customFields: Map<String, Any?>? = null,
         cats: List<DataClasses.CategoryData>? = null,
         replace: Boolean? = null,
         skipTriggers: Boolean? = null
       ): Unit
```

Функции изменения статуса подписки:

- `fun pushSubscribe()` — выполняет подписку на push-уведомления;
- `fun pushUnSubscribe()` — отменяет подписку на push-уведомления;
- `fun pushSuspend()` — приостанавливает подписку на push-уведомления (уведомления не приходят, но при этом не создается событие отписки в профиле пользователя);
- `fun unSuspendPushSubscription()` — используется для создания `LogIn`-, `LogOut`-переходов.

Функции этой группы имеют одинаковую сигнатуру, содержащую следующие параметры:

---

context: `Context`

Обязательный: **Да**
Описание: Android Context.

---

sync: `Boolean`

По умолчанию: `true`
Обязательный: Нет
Описание: Флаг, устанавливающий синхронность выполнения запроса.

Успешное выполнение запроса:

В случае успешного выполнения запроса данной группы функций будет создано событие с кодом **230**, содержащее значение `event.value`, определяемое в зависимости от флага синхронизации:

Если `sync == true`

```
ResponseWithHttpCode
  ├─ httpCode: 200
  └─ response
     ├─ error: 0
     ├─ errorText: ""
     └─ profile
        ├─ id: "your id"
        ├─ status: "subscribed"
        ├─ isTest: false
        └─ subscription
           ├─ subscriptionId: "your subscriptionId"
           ├─ hashId: "7f31a9c4"
           ├─ provider: "android-firebase"
           ├─ status: "subscribed"
           ├─ fields
           │  ├─ _os_ver: {"raw":"14","ver":"[\"14.0\", \"0.0\", \"0.0\"]"}
           │  ├─ _device_type: "mob"
           │  ├─ _ad_track: true
           │  ├─ _device_name: "Pixel"
           │  ├─ _os_language: "en"
           │  ├─ _os_tz: "+0100"
           │  ├─ _os: "Android"
           │  ├─ _device_model: "Pixel 7"
           │  └─ _app_ver: "1.0.0"
           └─ cats
              └─ [ { name: "developer_news", title: "dev_news", steady: false, active:
false } ]
```

При **синхронном** запросе в значении события `event.value` по ключу `"response_with_http_code"` доступны:

- **httpCode** – транспортный код ответа;
- **Response** – data class, содержащий:
  - `error: Int?` — внутренний код ошибки сервера *(0, если ошибок нет)*,
  - `errorText: String?` — текст ошибки *(пустая строка, если ошибок нет)*,
  - `profile: ProfileData?` — данные профиля, если запрос успешный:
    - информация о профиле (**ProfileData**)
    - подписка (**SubscriptionData**)
    - категории подписки (**CategoryData**)
    - если запрос завершился с ошибкой, то вернется только `profile = null`

▶ Структуры данных

Если `sync == false`

```
ResponseWithHttpCode
  ├─ httpCode: Int?
  └─ response: Response?
      ├─ error: Int?
      ├─ errorText: String?
      └─ profile: ProfileData? = null
```

При **асинхронном** запросе в значении события `event.value` по ключу `"response_with_http_code"` доступны:

- **httpCode** – транспортный код ответа;
- **Response** – data class, содержащий:
  - `error: Int?` — внутренний код ошибки сервера *(0, если ошибок нет)*,
  - `errorText: String?` — текст ошибки *(пустая строка, если ошибок нет)*,
  - `profile: ProfileData?` — данные профиля, для асинхронного запроса всегда равен `null`.

Выполнение запроса с ошибкой:

Если запрос данной группы функций завершился ошибкой, будет создано событие со следующими кодами:

Операции без автоматического повтора попытки на стороне SDK:

- **430** – подписка на уведомления;
- **431** — приостановка подписки;
- **432** — отписка.

Операции с автоматическим повтором попытки на стороне SDK:

- **530** – подписка на уведомления;
- **531** — приостановка подписки;
- **532** — отписка.

Содержимое события:

- только `httpCode`, если сервер Altcraft был **недоступен**;
- `error` и `errorText`, если сервер **вернул ошибку**.

Получение значений событий

```
AltcraftSDK.eventSDKFunctions.subscribe { event ->
    val codes = setOf(
        230, 231, 232,
        430, 431, 432,
        531, 532, 533
    )

    val code = event.eventCode ?: return@subscribe
    if (code !in codes) return@subscribe

    (event.eventValue?.get("response_with_http_code") as?
DataClasses.ResponseWithHttpCode)?.let { responseWithHttp ->
        val httpCode = responseWithHttp.httpCode

        val response = responseWithHttp.response
        val error = response?.error
        val errorText = response?.errorText

        val profile = response?.profile
        val profileId = profile?.id
        val profileStatus = profile?.status
        val profileIsTest = profile?.isTest

        val subscription = profile?.subscription
        val subscriptionId = subscription?.subscriptionId
        val hashId = subscription?.hashId
        val provider = subscription?.provider
        val subscriptionStatus = subscription?.status

        val fields = subscription?.fields

        val cats = subscription?.cats
        val firstCat = cats?.firstOrNull()
        val catName = firstCat?.name
        val catTitle = firstCat?.title
        val catSteady = firstCat?.steady
        val catActive = firstCat?.active
    }
}
```

---

profileFields: `Map<String, Any?>?`

По умолчанию: `null`
Обязательный: Нет
Описание: Словарь, содержащий поля профиля.

Параметр может принимать как **системные поля** (например, `_fname` — имя или `_lname` — фамилия), так и **опциональные** (заранее создаются вручную в интерфейсе платформы). Допустимые структуры (JSON-

совместимые):

- **Скалярные значения**:
  - String
  - Boolean
  - Int
  - Long
  - Float
  - Double
  - `null`
- **Объекты**: `Map<String, *>`
- **Списки**: `List<*>`
- **Массивы карт**: `Array<Map<String, *>>`

Если передано невалидное опциональное поле, запрос завершится с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: with field "название_поля": Incorrect field
```

---

customFields: `Map<String, Any?>?`

По умолчанию: `null`
Обязательный: Нет
Описание: Словарь, содержащий поля подписки.

Параметр может принимать как **системные поля** (например, `_device_model` — модель устройства или `_os` — операционная система), так и **опциональные** (заранее создаются вручную в интерфейсе платформы).
Допустимые типы значений (JSON-совместимые, **только скаляры**):

- String
- Boolean
- Int
- Long
- Float
- Double
- `null`

Если передано невалидное опциональное поле, запрос завершится с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: field "название_поля" is not valid: failed
convert custom field
```

TIP: Обратите внимание

Большая часть системных полей подписки автоматически собирается SDK и добавляется к push-запросам. К таким системным полям относятся: `"_os"`, `"_os_tz"`, `"_os_language"`, `"_device_type"`, `"_device_model"`, `"_device_name"`, `"_os_ver"`, `"_ad_track"`, `"_ad_id"`.

---

cats: `listOf(CategoryData)`

По умолчанию: `null`
Обязательный: Нет
Описание: Категории подписок.

Структура категории:

```
data class CategoryData(
    val name: String?,
    val title: String? = null,
    val steady: Boolean? = null,
    val active: Boolean?
)
```

При отправке push-запроса с указанием категорий используйте только поля `name` (название категории) и `active` (статус активности категории), другие поля не используется в обработке запроса. Поля `title` и `steady` заполняются при получении информации о подписке.

Пример запроса:

```
val cats = listOf(
    DataClasses.CategoryData(name = "football", active = true),
    DataClasses.CategoryData(name = "hockey",   active = true)
)
```

Категории используемые в запросе должны быть предварительно созданы и добавлены к ресурсу в платформе Altcraft. Если в запросе будет использовано поле, которое не добавлено в ресурс — запрос вернется с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: field "subscriptions.cats" is not valid:
category not found in resource
```

---

replace: `Boolean?`

По умолчанию: `null`
Обязательный: Нет

Описание: При активации флага все подписки других профилей с тем же push-токеном в текущей базе данных переводятся в статус `unsubscribed` после успешного запроса.

---

skipTriggers: `Boolean?`

По умолчанию: `null`
Обязательный: Нет
Описание: При активации флага профиль, содержащий данную подписку, будет игнорироваться в триггерах рассылок и сценариев.

Примеры реализации запроса

Пример выполнения запроса подписки на push-уведомления

Минимальная рабочая настройка:

```
AltcraftSDK.pushSubscriptionFunctions.pushSubscribe(context)
```

Передача всех доступных параметров:

```
AltcraftSDK.pushSubscriptionFunctions.pushSubscribe(
    context = this,
    sync = true,
    profileFields = mapOf("_fname" to "Andrey", "_lname" to "Pogodin"),
    customFields  = mapOf("developer" to true),
    cats = listOf(DataClasses.CategoryData(name = "developer_news", active = true)),
    replace = false,
    skipTriggers = false
)
```

TIP: Обратите внимание

Для `pushSubscribe`, `pushSuspend`, `pushUnSubscribe` предусмотрен автоматический повтор запроса со стороны SDK если http-код ответа находится в диапазоне **500…599**. Запрос не повторяется, если код ответа в этот диапазон не входит.

Функция `unSuspendPushSubscription()`

Функция `fun unSuspendPushSubscription()` предназначена для создания `LogIn`-, `LogOut`-переходов. Она работает следующим образом:

- проводит поиск подписок с тем же push-токеном, что и текущий, не относящихся к профилю, на который указывает текущий JWT-токен;
- меняет статус для найденных подписок с `subscribed` на `suspended`;
- меняет статус в подписках профиля, на который указывает текущий JWT, с `suspended` на `subscribed` (если профиль на который указывает JWT существует и в нем содержатся подписки);

- возвращает data class `ResponseWithHttpCode?`, в котором `response.profile` — текущий профиль, на который указывает JWT (если профиля не существует, то вместо `response.profile` вернется `null`).

Рекомендуемая реализация `LogIn`-, `LogOut`-переходов

`LogIn`-переход:

- Анонимный пользователь входит в приложение. Данному пользователю присвоен `JWT_1`, указывающий на базу данных **#1Anonymous**;
- Выполнена подписка на push-уведомления, профиль создан в базе данных **#1Anonymous**;
- Пользователь регистрируется, ему присваивается `JWT_2`, указывающий на базу данных **#2Registered**;
- Вызывается функция `unSuspendPushSubscription()` — подписка анонимного пользователя в базе данных **#1Anonymous** приостанавливается;
- Выполняется поиск профиля в базе данных **#2Registered** для восстановления подписки;
- Так как подписки с таким push-токеном в базе данных **#2Registered** не существует, функция `unSuspendPushSubscription()` вернет `null`;
- После получения значения `null` можно выполнить запрос на подписку `pushSubscribe()`, который создаст новый профиль в базе **#2Registered**.

`LogOut`-переход:

- Пользователь выполнил выход из профиля на стороне приложения (`LogOut`);
- Пользователю присваивается `JTW_1`, указывающий на базу данных **#1Anonymous**;
- Вызывается функция `unSuspendPushSubscription()`, которая приостановит подписку базе данных в **#2Registered** и сменит статус подписки в базе **#1Anonymous** на `subscribed`;
- Запрос вернет `#1Anonymous != null` — подписка уже существует, новая не требуется.

Пример реализации:

```
private suspend fun unSuspend(context: Context, logIn: Boolean) {

    // Изменение JWT до запроса
    setAuth(context, logIn)

    AltcraftSDK.pushSubscriptionFunctions
        .unSuspendPushSubscription(context)
        ?.let { result ->
            if (result.httpCode == 200 && result.response?.profile?.subscription == null) {
                AltcraftSDK.pushSubscriptionFunctions.pushSubscribe(
                    context = context
                    // Укажите необходимые параметры
                )
            }
        }
}

fun logIn(context: Context)  = CoroutineScope(Dispatchers.IO).launch { unSuspend(context,
true) }
fun logOut(context: Context) = CoroutineScope(Dispatchers.IO).launch { unSuspend(context,
false) }
```

Запрос статуса подписки

```
AltcraftSDK
└── val pushSubscriptionFunctions: PublicPushSubscriptionFunctions
    // Статус последней подписки профиля
    ├── suspend fun getStatusOfLatestSubscription(
    │       context: Context
    │   ): DataClasses.ResponseWithHttpCode?
    // Статус подписки по текущему токену/провайдеру
    ├── suspend fun getStatusForCurrentSubscription(
    │       context: Context
    │   ): DataClasses.ResponseWithHttpCode?
    // Статус последней подписки по указанному провайдеру (если null — используется
текущий)
    └── suspend fun getStatusOfLatestSubscriptionForProvider(
            context: Context,
            provider: String? = null
        ): DataClasses.ResponseWithHttpCode?
```

Функции запроса статуса подписки:

- `fun getStatusOfLatestSubscription()` — статус последней подписки профиля;
- `fun getStatusOfLatestSubscriptionForProvider()` — статус подписки для текущего токена/провайдера;

- `fun getStatusForCurrentSubscription()` — статус последней подписки по провайдеру. Если указан `null`, то используется текущий.

`suspend fun getStatusOfLatestSubscription(context: Context): DataClasses.ResponseWithHttpCode?`

Функция получения статуса последней подписки профиля. Возвращает объект `ResponseWithHttp?`, содержащий `response?.profile?.subscription` — последнюю созданную подписку в профиле. Если такой подписки не существует, будет передан `null`.

Пример использования:

```
AltcraftSDK.pushSubscriptionFunctions.getStatusOfLatestSubscription(context)
```

`suspend fun getStatusForCurrentSubscription(context: Context): DataClasses.ResponseWithHttpCode?`

Функция получения статуса подписки для текущего токена/провайдера. Возвращает объект `ResponseWithHttp?`, содержащий `response?.profile?.subscription` — подписку, найденную по текущему push-токену и провайдеру. Если такой подписки не существует, будет передан `null`.

Пример использования:

```
AltcraftSDK.pushSubscriptionFunctions.getStatusForCurrentSubscription(context)
```

`suspend fun getStatusOfLatestSubscriptionForProvider(context: Context, provider: String? = null): DataClasses.ResponseWithHttpCode?`

Функция получения статуса последней подписки по провайдеру. Возвращает объект `ResponseWithHttp?`, содержащий `response?.profile?.subscription` — последнюю созданную подписку с указанным провайдером push-уведомлений. Если провайдер не указан (`provider = null`), используется провайдер текущего токена. Если такой подписки не существует, будет передан `null`.

```
AltcraftSDK.pushSubscriptionFunctions.getStatusOfLatestSubscriptionForProvider(context,
provider = null)
```

Ниже представлен пример извлечения данных о профиле, подписке и категориях из ответа функций получения статуса. Данный подход актуален для всех функций получения статуса:

Данные из функций получения статуса

```
CoroutineScope(Dispatchers.IO).launch {
    AltcraftSDK.pushSubscriptionFunctions
        .getStatusForCurrentSubscription(this@App)
        ?.let { it ->
            val httpCode = it.httpCode
            val response = it.response
            val error = response?.error
            val errorText = response?.errorText
            val profile = response?.profile
            val subscription = profile?.subscription
            val cats = subscription?.cats
        }
}
```

Управление push-токенами провайдеров

```
AltcraftSDK
└── val pushTokenFunctions: PublicPushTokenFunctions
    |
    // Сохранить токен вручную (onNewToken)
    ├── fun setPushToken(context: Context, provider: String, token: String): Unit
    |
    // Получить текущие данные токена устройства
    ├── suspend fun getPushToken(context: Context): DataClasses.TokenData?
    |
    // Установить провайдер Firebase Cloud Messaging (null – удалить)
    ├── fun setFCMTokenProvider(provider: FCMInterface?): Unit
    |
    // Установить провайдер Huawei Mobile Services (null – удалить)
    ├── fun setHMSTokenProvider(provider: HMSInterface?): Unit
    |
    // Установить провайдер RuStore (null – удалить)
    ├── fun setRuStoreTokenProvider(provider: RustoreInterface?): Unit
    |
    // Удалить токен push указанного провайдера
    ├── suspend fun deleteDeviceToken(context: Context, provider: String, complete: () ->
Unit): Unit
    |
    // Принудительное обновление токена (удалить –> обновить)
    ├── fun forcedTokenUpdate(context: Context, complete: () -> Unit): Unit
    |
    // Изменить список приоритетов провайдеров и инициировать обновление токена
    └── suspend fun changePushProviderPriorityList(context: Context, priorityList:
List<String>): Unit
```

Функции для работы с токеном провайдера в SDK:

- `fun setPushToken()` — ручная установка push-токена устройства и провайдера в `UserDefaults`;
- `fun getPushToken()` — получение текущего push-токена;
- `fun setFCMTokenProvider()` — установка и снятие push-токена FCM;
- `fun setHMSTokenProvider()` — установка и снятие push-токена HSM;
- `fun setRuStoreTokenProvider()` — установка и снятие push-токена RuStore;
- `fun changePushProviderPriorityList()` — функция, позволяющая выполнить динамическую смену провайдера с обновлением push-токена подписки;
- `fun deleteDeviceToken()` — удаление push-токена указанного провайдера;
- `fun forcedTokenUpdate()` — удаление текущего push-токена с последующим обновлением.

`fun setPushToken(context: Context, provider: String, token: String): Unit`

Функция предназначена для ручной установки push-токена устройства и провайдера. Должна выполняться в функции `onNewToken()` сервиса push-провайдера. Используется как упрощённый вариант передачи токена в SDK без реализации интерфейсов провайдеров.

Не рекомендуется использовать эту функцию для передачи токена. Рекомендуемый подход передачи push-токена в SDK — реализация `FCMInterface`, `HMSInterface` или `APNSInterface`.

Пример использования:

```
AltcraftSDK.pushTokenFunctions.setPushToken(context, provider, token)
```

Пример передачи токена в `FCMService.onNewToken()`:

```
class FCMService : FirebaseMessagingService() {
    override fun onNewToken(token: String) {
        super.onNewToken(token)

        // Передать новый токен вручную
        AltcraftSDK.pushTokenFunctions.setPushToken(this, FCM_PROVIDER, token)
    }

    override fun onDeletedMessages() {}

    override fun onMessageReceived(message: RemoteMessage) {
        super.onMessageReceived(message)
        AltcraftSDK.PushReceiver.takePush(this@FCMService, message.data)
    }
}
```

`suspend fun getPushToken(context: Context): DataClasses.TokenData?`

Функция возвращает текущие данные push-токена устройства и провайдера в виде data class `TokenData(val provider: String, val token: String)`. Если токен недоступен — будет передано `null`.

Пример использования:

```
AltcraftSDK.pushTokenFunctions.getPushToken(context)
```

Пример запроса **получения данных push-токена**:

```
CoroutineScope(Dispatchers.IO).launch {
    AltcraftSDK.pushTokenFunctions.getPushToken(context).let {
        val provider = it?.provider
        val token = it?.token
    }
}
```

`fun setFCMTokenProvider(provider: FCMInterface?): Unit`

Функция устанавливает или снимает push-токен провайдера Firebase Cloud Messaging. Чтобы отключить провайдера, передайте `null`.

Пример использования:

```
AltcraftSDK.pushTokenFunctions.setFCMTokenProvider(FCMProvider())
```

TIP: Важно

Вызывайте `setFCMTokenProvider()` в `Application.onCreate()` до вызова `AltcraftSDK.initialization()`. Это гарантирует регистрацию на старте процесса приложения, независимо от состояния жизненного цикла других компонентов и того, запущено приложение в foreground или background.

`fun setHMSTokenProvider(provider: HMSInterface?): Unit`

Функция устанавливает или снимает push-токен провайдера Huawei Mobile Services. Чтобы отключить провайдера, передайте `null`.

Пример использования:

```
AltcraftSDK.pushTokenFunctions.setHMSTokenProvider(HMSProvider())
```

TIP: Важно

Вызывайте `setHMSTokenProvider()` в `Application.onCreate()` до вызова `AltcraftSDK.initialization()`. Это гарантирует регистрацию на старте процесса приложения,

независимо от состояния жизненного цикла других компонентов и того, запущено приложение в foreground или background.

`fun setRuStoreTokenProvider(provider: RustoreInterface?): Unit`

Функция устанавливает или снимает push-токен провайдера RuStore. Чтобы отключить провайдера, передайте `null`.

Пример использования:

```
AltcraftSDK.pushTokenFunctions.setRustoreTokenProvider(RustoreProvider())
```

TIP: Важно

Вызывайте `setRustoreTokenProvider()` в `Application.onCreate()` до вызова `AltcraftSDK.initialization()`. Это гарантирует регистрацию на старте процесса приложения, независимо от состояния жизненного цикла других компонентов и того, запущено приложение в foreground или background.

`suspend fun changePushProviderPriorityList(context: Context, priorityList: List<String>): Unit`

Функция, позволяющая выполнить динамическую смену push-провайдера с обновлением токена подписки. Для этого необходимо передать новый массив с другим порядком провайдеров. Например: `listOf(HMS_PROVIDER, RUSTORE_PROVIDER, FCM_PROVIDER)`.

```
AltcraftSDK.pushTokenFunctions.changePushProviderPriorityList(context, listOf(HMS_PROVIDER, RUSTORE_PROVIDER, FCM_PROVIDER))
```

`suspend fun deleteDeviceToken(context: Context, provider: String, complete: () -> Unit): Unit`

Функция удаления push-токена указанного провайдера. Токен инвалидируется и удаляется из локального кеша на устройстве и с сервера push-провайдера. После удаления можно запросить новый токен.

```
AltcraftSDK.pushTokenFunctions.deleteDeviceToken(context, provider) {
    // callback после удаления
}
```

`fun forcedTokenUpdate(context: Context, complete: () -> Unit): Unit`

Функция удаления текущего push-токена с его последующим обновлением.

Пример использования:

```
AltcraftSDK.pushTokenFunctions.forcedTokenUpdate(context) {
    // callback после удаления
}
```

Пример регистрации провайдеров

Мы **не рекомендуем** использовать функцию `setPushToken` для установки push-токена. Вместо этого отдельно настройте функции получения токена для каждого используемого провайдера. Ниже указан пример реализации этого метода:

Рекомендованный способ регистрации провайдеров

```
class App : Application() {
    override fun onCreate() {
        super.onCreate()

        // установить провайдер RuStore
        RuStorePushClient.init(this, "rustore project id")

        // установить провайдер JWT
        AltcraftSDK.setJWTProvider(JWTProvider(applicationContext))

        // установить провайдер FCM
        AltcraftSDK.pushTokenFunctions.setFCMTokenProvider(FCMProvider())

        // установить провайдер HMS
        AltcraftSDK.pushTokenFunctions.setHMSTokenProvider(HMSProvider())

        // установить провайдер RuStore
        AltcraftSDK.pushTokenFunctions.setRuStoreTokenProvider(RuStoreProvider())

        // создать AltcraftConfiguration
        val config = AltcraftConfiguration.Builder(
            apiUrl = "your api url",
            R.drawable.ic_altcraft_label
        ).build()

        // Инициализация SDK
        AltcraftSDK.initialization(context = this@App, configuration = config)
    }
}
```

Передача push-уведомлений в SDK

`PushReceiver` — публичный класс, в котором содержится функция, принимающая уведомления.

```
AltcraftSDK
└── open class PushReceiver
    // Обработка входящего push
    ├── open fun pushHandler(
    │       context: Context,
    │       message: Map<String, String>
    │   ): Unit
    // Точка входа в SDK для доставки push
    └── companion object
        └── fun takePush(
                context: Context,
                message: Map<String, String>
            ): Unit
```

Функции передачи push-уведомлений в SDK:

- `fun takePush()` — принимает уведомления в сервисе push-провайдеров для их дальнейшей обработки на стороне SDK;
- `fun pushHandler()` — запускает стандартный механизм обработки push-уведомления Altcraft в SDK.

Прием уведомления

`fun takePush(context: Context, message: Map<String, String>): Unit`

Функция, принимающая push-уведомления для их дальнейшей обработки на стороне SDK.

Пример использования:

```
  override fun onMessageReceived(message: RemoteMessage) {
       super.onMessageReceived(message)

      AltcraftSDK.PushReceiver.takePush(this@FCMService, message.data)
   }
   // message.data – payload сообщения
```

Обработка уведомления

`open fun pushHandler(context: Context, message: Map<String, String>): Unit`

Функция запускает стандартный механизм обработки push-уведомления Altcraft в SDK.

Получение уведомлений в любом пакете приложения (опционально)

Для того чтобы получить уведомления в любом пакете приложения, выполните следующие действия:

Шаг 1. Создайте класс `AltcraftPushReceiver`, расширяющий `PushReceiver`. В этом классе будет переопределена функция `pushHandler()`:

```
@Keep
class AltcraftPushReceiver : AltcraftSDK.PushReceiver() {
    override fun pushHandler(context: Context, message: Map<String, String>) {
        // стандартная обработка push-сообщений и отображение уведомлений
        super.pushHandler(context, message)
    }
}
```

TIP: Обратите внимание

Класс должен называться **AltcraftPushReceiver**. Если вы назовете его иначе, SDK не сможет его найти для передачи уведомления.

Шаг 2. В зависимости от ваших бизнес-целей, настройте логику работы класса AltcraftPushReceiver:

- Если вы хотите, чтобы обработку и показ уведомлений выполнял **SDK**, то:
    - используйте `super.pushHandler(context, message)`.
- Если вы хотите **самостоятельно** обрабатывать уведомления, то:
    - не вызывайте функцию `super.pushHandler()`;
    - вручную регистрируйте события открытия с помощью функции `openEvent()`, иначе оно не зарегистрируется платформой Altcraft.

События доставки “deliveryEvent” регистрируются автоматически. Создание `AltcraftPushReceiver` классов на регистрацию этого события **не влияет**.

TIP: Обратите внимание

Если вы создали несколько классов `AltcraftPushReceiver`, то вызов `super.pushHandler()` в каждом из этих классов будет показывать сообщение пользователю. Чтобы избежать дубликации уведомлений, вызывайте `super.pushHandler()` только в **одном классе**.

Шаг 3. Добавьте имена пакетов, которые содержат классы `AltcraftPushReceiver`, в параметр конфигурации `pushReceiverModules`. После этого SDK автоматически определит наличие классов `AltcraftPushReceiver` в указанных пакетах с помощью механизма рефлексии.

Если код приложения будет обфусцироваться, то класс должен быть **помечен аннотацией** `@Keep` или **добавлен в правила** `R8/ProGuard`, иначе SDK не сможет его обнаружить.

Пример добавления пакета в параметр `pushReceiverModules`:

```
pushReceiverModules = listOf(
   context.packageName, // пакет приложения
   "com.altcraft.altcraftmobile.test"
)
```

Мобильные события

```
// Функция объекта PublicMobileEventFunctions

AltcraftSDK
└── val mobileEventFunctions: PublicMobileEventFunctions

    // Отправить мобильное событие (mobile event) на сервер
    └── fun mobileEvent(
        context: Context,
        sid: String,
        eventName: String,
        sendMessageId: String? = null,
        payload: Map<String, Any?>? = null,
        matching: Map<String, Any?>? = null,
        matchingType: String? = null,
        profileFields: Map<String, Any?>? = null,
        subscription: DataClasses.Subscription? = null,
        utm: DataClasses.UTM? = null
    ): Unit
```

Кейс: Передача информации о рекламной кампании

Информация о рекламной кампании приложения, которая привела к установке, может быть передана в платформу как значение параметров `profileFields` / `customFields` функции `pushSubscribe`, а также параметров `utm` или `payload` функции `mobileEvent()`. Получив UTM-метку в приложении как строку, необходимо вызвать функцию `mobileEvent()`:

Передача через payload

```
AltcraftSDK.mobileEventFunctions.mobileEvent(
            this,
            sid = "your sid",
            eventName = "app_install",
            payload = mapOf("utm" to "your utm tag")
        )
```

Передача в поля UTM

```
AltcraftSDK.mobileEventFunctions.mobileEvent(
            this,
            sid = "your sid",
            eventName = "app_install",
            utm = DataClasses.UTM(
                campaign = "your campaign utm",
                content =  "your content utm",
                keyword = "your keyword utm",
                medium = "your medium utm",
                source = "your source utm",
                temp = "your temp utm"
            )
        )
```

Передача в пользовательское поле профиля

```
//передача utm-метки в поле профиля
//поля профиля должны быть предварительно добавлены в платформе
AltcraftSDK.shared.pushSubscriptionFunctions.pushSubscribe(
     profileFields: ["utm": "your utm tag"]
)
```

Переданные любым из этих способов данные можно использовать для создания сегмента в платформе.

Для регистрации мобильного события используйте функцию `mobileEvent()`. Она имеет следующие параметры:

---

context: `Context`

Обязательный: **Да**

Описание: Android Context.

---

sid: `String`

Обязательный: **Да**

Описание: Строковый идентификатор пикселя, к которому привязываются мобильные события.

---

eventName: `String`

Обязательный: **Да**
Описание: Имя мобильного события.

---

sendMessageId: `String?`

Обязательный: Нет
Описание: SMID-идентификатор отправленного сообщения (если событие связано с конкретной рассылкой).

---

payload: `String`

Обязательный: Нет
Описание: Данные события — карта со строковыми ключами и значениями, для которых допускаются только **скалярные** типы данных:

- String
- Boolean
- Int
- Long
- Float
- Double
- `null`

---

matching: `Map<String, Any?>?`

Обязательный: Нет
Описание: Карта, в которую можно передавать значения с типами и идентификаторами матчинга.

---

matchingType: `String?`

Обязательный: Нет
Описание: Тип матчинга.

---

profileFields: `Map<String, Any?>?`

Обязательный: Нет
Описание: Поля профиля — карта со строковыми ключами и значениями (JSON-совместимые типы):

- **Скалярные значения**:
  - String
  - Boolean
  - Int
  - Long
  - Float
  - Double
  - `null`
- **Объекты**: `Map<String, *>`
- **Списки**: `List<*>`
- **Массивы карт**: `Array<Map<String, *>>`

> **TIP: Обратите внимание**
>
> Параметр используется только при работе с JWT-авторизацией.

---

utm: `DataClasses.UTM?`

Обязательный: Нет
Описание: UTM-метки. Добавляются с помощью data class UTM, где каждый вид UTM — отдельный параметр.

```
data class UTM (
    val campaign: String? = null,
    val content: String? = null,
    val keyword: String? = null,
    val medium: String? = null,
    val source: String? = null,
    val temp: String? = null
)
```

---

subscription: `DataClasses.Subscription?`

Обязательный: Нет
Описание: Параметр добавления подписки для выбранного канала.

Значения параметра — реализации (подтипы) sealed-интерфейса Subscription:

- `EmailSubscription` — email-подписка
- `SmsSubscription` — SMS-подписка
- `PushSubscription` — push-подписка
- `CcDataSubscription` — подписка в Telegram, Whatsapp, Viber, Notify.

> **TIP: Обратите внимание**
>
> Используется только при работе с JWT-авторизацией.

Реализации интерфейса `Subscription`

Общая модель: `Subscription`

Назначение — базовый (sealed) интерфейс для всех видов подписок.
Серилизация полиморфная, поле-дискриминатор — `channel`.

Общие поля (для всех реализаций):

Поле	Тип	Обязательный	Описание
`resource_id`	`Int`	Да	Идентификатор ресурса/источника подписки
`status`	`String?`	Нет	Статус подписки (например, активна/приостановлена)
`priority`	`Int?`	Нет	Приоритет доставки для данной подписки
`custom_fields`	`Map<String, Any?>?` *	Нет	Пользовательские поля (ключ-значение) для расширенной сегментации
`cats`	`List<String>?`	Нет	Категории подписки
`channel`	`String`	Да	Тип канала; фиксируется реализацией.

Варианты подписки:

EmailSubscription (`channel = "email"`)

Основные поля

Поле	Тип	Обязательный	Описание
`resourceId`	`Int`	Да	ID ресурса Altcraft
`email`	`String`	Да	Адрес электронной почты получателя

Дополнительные поля

Поле	Тип	Обязательный	Описание
`status`	`String`	Нет	Статус подписки
`priority`	`Int`	Нет	Приоритет подписки
`customFields`	`Map<String, Any?>`	Нет	Стандартные и кастомные поля подписки
`cats`	`List<String>`	Нет	Категории подписки

SmsSubscription (`channel = "sms"`)

Основные поля

Поле	Тип	Обязательный	Описание
`resourceId`	`Int`	Да	ID ресурса Altcraft
`phone`	`String`	Да	Номер телефона в международном формате

Дополнительные поля

Поле	Тип	Обязательный	Описание
`status`	`String`	Нет	Статус подписки
`priority`	`Int`	Нет	Приоритет подписки
`customFields`	`Map<String, Any?>`	Нет	Стандартные и кастомные поля подписки
`cats`	`List<String>`	Нет	Категории подписки

PushSubscription (`channel = "push"`)

Основные поля

Поле	Тип	Обязательный	Описание
`resourceId`	`Int`	Да	ID ресурса Altcraft
`provider`	`String`	Да	Провайдер (например, `"android-firebase"`)
`subscriptionId`	`String`	Да	Уникальный идентификатор подписки у провайдера

Дополнительные поля

Поле	Тип	Обязательный	Описание
`status`	`String`	Нет	Статус подписки
`priority`	`Int`	Нет	Приоритет подписки
`customFields`	`Map<String, Any?>`	Нет	Стандартные и кастомные поля подписки
`cats`	`List<String>`	Нет	Категории подписки

CcDataSubscription (`channel ∈ {"telegram_bot","whatsapp","viber","notify"}`)

Основные поля

Поле	Тип	Обязательный	Описание
`resourceId`	`Int`	Да	ID ресурса Altcraft
`channel`	`String`	Да	Один из: `"telegram_bot"` , `"whatsapp"` , `"viber"` , `"notify"`
`ccData`	`JsonObject`	Да	Канал-специфичные данные (например, chat ID, номер, токены)

Дополнительные поля

Поле	Тип	Обязательный	Описание
`status`	`String`	Нет	Статус подписки
`priority`	`Int`	Нет	Приоритет подписки
`customFields`	`Map<String, Any?>`	Нет	Стандартные и кастомные поля подписки
`cats`	`List<String>`	Нет	Категории подписки

Обновление полей профиля

INFO: Важно

Обновление полей профиля с помощью функции `updateProfileFields()` работает только при использовании JWT-авторизации API-запросов SDK.

```
AltcraftSDK
└── val profileFunctions: PublicProfileFunctions
    // Обновляет поля профиля Altcraft
    └── fun updateProfileFields(
        context: Context,
        profileFields: Map<String, Any?>? = null,
        skipTriggers: Boolean? = null
    ): Unit
```

Для обновление полей профиля используйте функцию `updateProfileFields()`. Она имеет следующие параметры:

---

context: `Context`

Обязательный: **Да**
Описание: Android Context.

---

profileFields: `Map<String, Any?>?`

Обязательный: Нет
Описание: Поля профиля — карта содержащая значение для полей, которые необходимо изменить (JSON-совместимые типы):

- **Скалярные значения**:
  - String
  - Boolean
  - Int
  - Long
  - Float
  - Double

    - `null`
- **Объекты**: `Map<String, *>`
- **Списки**: `List<*>`
- **Массивы карт**: `Array<Map<String, *>>`

> **INFO: Обратите внимание**
>
> Для того чтобы обновление прошло успешно, поля должны быть предварительно добавлены в профиль подписчика.

---

skipTriggers: `Boolean?`

По умолчанию: `null`
Обязательный: Нет
Описание: При активации флага профиль, содержащий данную подписку, будет игнорироваться в триггерах рассылок и сценариев.

Получение событий SDK в приложении

```
// Функции объекта Events

AltcraftSDK
└── val eventSDKFunctions: Events
    // Подписка на события SDK (заменяет существующего подписчика)
    ├── fun subscribe(
    │       newSubscriber: (DataClasses.Event) -> Unit
    │   ): Unit
    // Отписка от событий
    └── fun unsubscribe(): Unit
```

В приложении может быть **только один** активный подписчик на события SDK.

Подписка на события

`fun subscribe(newSubscriber: (DataClasses.Event) -> Unit): Unit`

Функция используется для того, чтобы подписаться и получать события SDK в приложении. При возникновении события SDK она вызывает `callback` и передаёт в него экземпляр класса `Event` (или его наследника).

Пример использования:

```
AltcraftSDK.eventSDKFunctions.subscribe { event ->
    // Обработка события
}
```

Все события, передаваемые SDK, являются экземплярами `Event` или его наследников:

- **Event** — общее событие (информация, успешные запросы);
- **Error** — событие об ошибке;
- **RetryError** — событие об ошибке при выполнения запроса для которого предусмотрен автоматический повтор выполнения на стороне SDK.

Каждое событие содержит поля:

- **function** — имя функции, вызывавшей событие;
- **eventCode** — внутренний код события SDK (список событий SDK представлен в пункте “События SDK”);
- **eventMessage** — сообщение события;
- **eventValue** — произвольные данные [String: Any?]? которые добавляются к некоторым событиям как полезная нагрузка;
- **date** — время события.

Классы событий SDK

```
open class Event(
    val function: String,
    val eventCode: Int? = null,
    val eventMessage: String? = null,
    val eventValue: Map<String, Any?>? = null,
    val date: Date = Date(),
)

open class Error(
    function: String,
    eventCode: Int? = 0,
    eventMessage: String? = null,
    eventValue: Map<String, Any?>? = null,
    date: Date = Date(),
) : Event(function, eventCode, eventMessage, eventValue, date)

class RetryError(
    function: String,
    eventCode: Int? = 0,
    eventMessage: String? = null,
    eventValue: Map<String, Any?>? = null,
    date: Date = Date(),
) : Error(function, eventCode, eventMessage, eventValue, date)
```

Пример содержания события успешной подписки на push-уведомления

```
├ function: processResponseprocessResponse
├ eventCode: 230
├ eventMessage: "successful request: push/subscribe"
├ eventValue
│  ├ http code: 200
│  └ response
│     ├ error: 0
│     ├ errorText: ""
│     └ profile
│        ├ id: "your id"
│        ├ status: "subscribed"
│        ├ isTest: false
│        └ subscription
│           ├ subscriptionId: "your subscriptionId"
│           ├ hashId: "c52b28d2"
│           ├ provider: "android-firebase"
│           ├ status: "subscribed"
│           ├ fields
│           │  ├ _device_name: "Pixel 7"
│           │  ├ _device_model: "Google Pixel 7"
│           │  ├ _os_tz: "+0300"
│           │  ├ _os_language: "ru"
│           │  ├ _os_ver: {"raw":"14","ver":[14]}
│           │  ├ _ad_track: true
│           │  ├ _os: "Android"
│           │  └ _device_type: "mob"
│           └ cats
│              └ [ { name: "developer_news", title: "dev_news", steady: false, active:
false } ]
└ date: 2025-09-03 09:01:44 +0000
```

Отписка от событий

`fun unsubscribe(): Unit`

При помощи данной функции можно отписаться от событий SDK. Она отменяет передачу событий SDK, при этом `callback` остаётся назначенным, но доставка событий прекращается.

Пример использования:

```
AltcraftSDK.eventSDKFunctions.unsubscribe()
```

Список всех событий SDK

Список событий

Код	Описание
200	SDK configuration is installed
201	push provider set
202	received a notification unrelated to the Altcraft Platform
203	received Altcraft push notification
204	push is posted
205	SDK data has been cleared
206	waiting for SDK initialization to complete
207	waiting for push token update to complete
220	receiver override event
230	subscribe request succeeded
231	push suspend request succeeded
232	push unsubscribed request succeeded
233	push update request succeeded
234	push unsuspend request succeeded
235	profile status request succeeded
236	push event delivered successfully
237	mobile event delivered successfully
401	the configuration is not set
402	userTag is null. It is impossible to identify the user
404	SDK initialization timeout has expired
405	mobile event parts is null
422	unsuspend request data is null
423	profile request data is null
430	subscribe request failed
431	suspend request failed
432	unsubscribed request failed
433	update request failed
434	unsuspend request failed
435	status request failed
436	push event delivery failed

Код	Описание
437	mobile event request failed
450	push data is null
451	uid in the push data is null or empty, it is impossible to send a push event to the server
452	error uploading the notification image
453	couldn’t create notification
454	foreground info is null
455	the notification channel has not been created
471	invalid provider. Available - android-firebase, android-huawei, android-rustore
472	invalid customFields: not all values are primitives
480	subscribe retry limit reached (request removed from DB)
484	push event retry limit reached (request removed from DB)
485	mobile event retry limit reached (request removed from DB)
501	config data is null
502	current push token is null
503	userTag is null. It is impossible to identify the user
504	no permission to send notifications
505	no internet connection, retry when connection is restored
506	failed to update the push token. The subscription request was rejected
520	push subscribe request data is null
521	token update request data is null
524	push event request data is null
525	mobile event request data is null
529	common data is null
530	subscribe request failed (retryable)
531	token update request failed (retryable)
534	push event delivery failed (retryable)
535	mobile event delivery failed (retryable)
540	JWT token is null
541	matching mode is null
542	JWT payload exceeds allowed size (16 KB limit): input rejected to prevent DoS

Код	Описание
543	JWT does not contain a payload
544	auth data is null
545	matching claim does not contain a matching ID
560	response data is null

Дополнительные функции SDK

Ручная регистрация push-событий

> **TIP: Обратите внимание**
>
> Эти функции применяются только в том случае, если вы реализуете собственную логику обработки уведомлений и не передаёте их в Altcraft SDK. По умолчанию Altcraft SDK обрабатывает уведомления автоматически, поэтому использование данных методов требуется лишь при необходимости кастомной обработки на стороне клиента.

```
// Функции объекта PublicPushEventFunctions

AltcraftSDK
└── val pushEventFunctions: PublicPushEventFunctions

    // Зафиксировать доставку Altcraft-push (вызывает delivery-ивент)
    ├── fun deliveryEvent(
    │       context: Context,
    │       message: Map<String, String>? = null,
    │       uid: String? = null
    │   ): Unit

    // Зафиксировать открытие Altcraft-push (вызывает open-ивент)
    └── fun openEvent(
            context: Context,
            message: Map<String, String>? = null,
            uid: String? = null
        ): Unit
```

Функции ручной регистрации событий:

- `fun deliveryEvent()` — регистрация доставки уведомления;
- `fun openEvent()` — регистрация открытия уведомления.

`fun deliveryEvent(context: Context, message: Map<String, String>? = null, uid: String? = null): Unit`

Функция ручной регистрации события доставки уведомления в платформе Altcraft. Для регистрации события на сервере передайте в параметр `message` или `uid` полезные данные уведомления.

Пример использования:

```
AltcraftSDK.pushEventFunctions.deliveryEvent(context, message, uid)
```

`fun openEvent(context: Context, message: Map<String, String>? = null, uid: String? = null): Unit`

Функция ручной регистрации события открытия уведомления в платформе Altcraft. Для регистрации события на сервере передайте в параметр `message` или `uid` полезные данные уведомления.

Пример использования:

```
AltcraftSDK.pushEventFunctions.openEvent(context, message, uid)
```

Передача произвольных данных в Intent extras при открытии push-уведомления

SDK поддерживает передачу произвольных данных из push-уведомления в приложение через специальный ключ `"_extra"`. Значение `_extra` должно быть строкой, содержащей JSON-объект.

При клике по уведомлению SDK автоматически:

1. Извлекает значение `_extra` из push payload.
2. Добавляет его в `Intent`.
3. Передаёт в Activity, обрабатывающую открытие уведомления или deep link.

Таким образом, содержимое `_extra` становится обычным `Intent` extra и доступно в приложении через:

```
val extraJson = intent?.getStringExtra("_extra")
```

Поле `_extra` предназначено для передачи дополнительных параметров, которые не должны быть частью URL, но необходимы при обработке открытия уведомления.

Значение поля `_extra` может быть добавлено с помощью Custom JSON в шаблоне push-уведомления.

Очистка данных SDK

`fun clear(context: Context, onComplete: (() -> Unit)? = null)`

Функция позволяет выполнить очистку данных SDK и отменить работу всех фоновых задач, которые ожидают выполнения. Она выполняет отмену задач `WorkManager`, удаляет записи БД `Room` и очищает `SharedPreferences`.

Пример использования:

```
AltcraftSDK
// Полная очистка данных SDK (БД, SharedPreferences, фоновые задачи)
└── fun clear(
        context: Context,
        onComplete: (() -> Unit)? = null
    ): Unit
```

Функция содержит необязательный параметр `completion`, который вызывается после завершения очистки и отмены задач.

Функциональное обновление полей профиля

```
AltcraftSDK
└ val pushSubscriptionFunctions: PublicPushSubscriptionFunctions
   └ fun actionField(key: String): ActionFieldBuilder
```

`fun actionField(key: String): ActionFieldBuilder`

Функция, облегчающая процесс функционального обновления полей профиля. Поддерживает следующие команды:

```
    .set(value)
    .unset(value)
    .incr(value)
    .add(value)
    .delete(value)
    .upsert(value)
```

Пример использования:

```
AltcraftSDK.pushSubscriptionFunctions.pushSubscribe(
    context = context,
    // "_fname" – изменяемое поле
    // .set("Andrey") – команда, которая установить значение "Andrey" для этого поля
    profileFields = AltcraftSDK.pushSubscriptionFunctions
        .actionField("_fname").set("Andrey")
)
```

Сброс запрета на retry-операции при инициализации SDK

TIP: Обратите внимание

При инициализации SDK выполняется запуск фоновых задач, выполняющих контроль и повторную отправку запросов, связанных с push-уведомлениями, мобильными событиями, а также проверку и

выполнение запроса на обновление push-токена устройства. Выполнение данной функции ограничено одним запуском в пределах одного жизненного цикла процесса приложения.

Иногда может возникнуть необходимость сброса этого ограничения. В таких ситуациях используйте функцию:

```
fun reinitializeRetryControlInThisSession(): Unit
```

Функция сбрасывает флаг запрета на повторный запуск фоновых задач, выполняющих контроль и повторную отправку запросов.

Запрос разрешения на отправку уведомлений

TIP: Обратите внимание

Начиная с Android 13 (API 33, Tiramisu), приложения должны явно запрашивать разрешение `POST_NOTIFICATIONS`, прежде чем отправлять уведомления. На более ранних версиях Android вызов функции не требуется — разрешение предоставляется автоматически.

```
fun requestNotificationPermission(context: Context, activity: ComponentActivity)
```

Функция выполняет проверку текущего статуса разрешения и, при необходимости, показывает пользователю системный диалог запроса разрешения. Если пользователь уже выбрал запрет на показ уведомлений, то повторный вызов функции снова выведет диалог запроса.

Публичный API SDK

object AltcraftSDK

```
AltcraftSDK
// Инициализация SDK и установка конфигурации
├─ fun initialization(context: Context, configuration: AltcraftConfiguration, complete:
((Result<Unit>) -> Unit)? = null): Unit
// Полная очистка данных SDK (БД, SP, фоновые задачи)
├─ fun clear(context: Context, onComplete: (() -> Unit)? = null): Unit
// Регистрация провайдера JWT
├─ fun setJWTProvider(provider: JWTInterface?): Unit
// Разрешить переинициализацию push-модуля в текущей сессии
├─ fun unlockInitialOperationsInThisSession(): Unit
// Базовый получатель Altcraft push (можно переопределить)
├─ open class PushReceiver
│  // Обработка входящего push-сообщения
│  ├─ open fun pushHandler(context: Context, message: Map<String, String>): Unit
│  // Точка входа доставки push в SDK
│  └─ companion object
│     └─ fun takePush(context: Context, message: Map<String, String>): Unit
// Публичные функции подписки
├─ val pushSubscriptionFunctions: PublicPushSubscriptionFunctions
│  // Подписка на пуш уведомление(status = SUBSCRIBED)
│  ├─ fun pushSubscribe(
│  │     context: Context,
│  │     sync: Boolean = true,
│  │     profileFields: Map<String, Any?>? = null,
│  │     customFields: Map<String, Any?>? = null,
│  │     cats: List<DataClasses.CategoryData>? = null,
│  │     replace: Boolean? = null,
│  │     skipTriggers: Boolean? = null
│  │   ): Unit
│  // Приостановка подписки на push уведомления(status = SUSPENDED)
│  ├─ fun pushSuspend(
│  │     context: Context,
│  │     sync: Boolean = true,
│  │     profileFields: Map<String, Any?>? = null,
│  │     customFields: Map<String, Any?>? = null,
│  │     cats: List<DataClasses.CategoryData>? = null,
│  │     replace: Boolean? = null,
│  │     skipTriggers: Boolean? = null
│  │   ): Unit
│  // Отписка от push уведомлегний (status = UNSUBSCRIBED)
│  ├─ fun pushUnSubscribe(
│  │     context: Context,
│  │     sync: Boolean = true,
│  │     profileFields: Map<String, Any?>? = null,
│  │     customFields: Map<String, Any?>? = null,
│  │     cats: List<DataClasses.CategoryData>? = null,
│  │     replace: Boolean? = null,
│  │     skipTriggers: Boolean? = null
│  │   ): Unit
```

```
│  │    ): Unit
│  // смена статуса подписки указанной в JWT с suspended на subscribed, остальные подписки
содержащие указанный push токен сменят статус с subscribed на suspended.
│  ├─ suspend fun unSuspendPushSubscription(context: Context):
DataClasses.ResponseWithHttpCode?
│  // Статус последней подписки профиля
│  ├─ suspend fun getStatusOfLatestSubscription(context: Context):
DataClasses.ResponseWithHttpCode?
│  // Статус последней подписки профиля по указанному провайдеру push уведомлений
│  ├─ suspend fun getStatusOfLatestSubscriptionForProvider(context: Context, provider:
String? = null): DataClasses.ResponseWithHttpCode?
│  // Статус подписки с текущим токеном устройсва.
│  ├─ suspend fun getStatusForCurrentSubscription(context: Context):
DataClasses.ResponseWithHttpCode?
│  // добавить функциональное поле профиля(set/incr/...)
│  └─ fun actionField(key: String): ActionFieldBuilder
// Публичные функции управления токенами
├─ val pushTokenFunctions: PublicPushTokenFunctions
│  // Сохранить токен провайдера вручную (onNewToken)
│  ├─ fun setPushToken(context: Context, provider: String, token: String): Unit
│  // Получить текущий токен устройства
│  ├─ suspend fun getPushToken(context: Context): DataClasses.TokenData?
│  // Зарегистрировать провайдера FCM
│  ├─ fun setFCMTokenProvider(provider: FCMInterface?): Unit
│  // Зарегистрировать провайдера HMS
│  ├─ fun setHMSTokenProvider(provider: HMSInterface?): Unit
│  // Зарегистрировать провайдера RuStore
│  ├─ fun setRuStoreTokenProvider(provider: RustoreInterface?): Unit
│  // Удалить токен у выбранного провайдера
│  ├─ suspend fun deleteDeviceToken(context: Context, provider: String, complete: () ->
Unit): Unit
│  // Форс-обновление токена (удалить –> обновить)
│  ├─ fun forcedTokenUpdate(context: Context, complete: () -> Unit): Unit
│  // Изменить приоритет провайдеров и обновить токен
│  └─ suspend fun changePushProviderPriorityList(context: Context, priorityList:
List<String>): Unit
// Публичные функции отправки событий по пушам
├─ val pushEventFunction: PublicPushEventFunctions
│  // Зафиксировать доставку Altcraft-push (вызывает delivery-ивент)
│  ├─ fun deliveryEvent(context: Context, message: Map<String, String>? = null, uid:
String? = null): Unit
│  // Зафиксировать открытие Altcraft-push (вызывает open-ивент)
│  └─ fun openEvent(context: Context, message: Map<String, String>? = null, uid: String? =
null): Unit
//
// Регистрация мобильных событий
├─ val mobileEventFunctions: PublicMobileEventFunctions
│  // Регистрирует мобильное событие Altcraft
│  └─ fun mobileEvent(
│        context: Context,
```

```
│        sid: String,
│        eventName: String,
│        sendMessageId: String? = null,
│        payload: Map<String, Any?>? = null,
│        matching: Map<String, Any?>? = null,
│        matchingType: String? = null,
│        profileFields: Map<String, Any?>? = null,
│        subscription: DataClasses.Subscription? = null,
│        utm: DataClasses.UTM? = null
│    ): Unit
//
// Обновление полей профиля
├─ val profileFunctions: PublicProfileFunctions
│  // Обновляет поля профиля Altcraft
│  └─ fun updateProfileFields(
│        context: Context,
│        profileFields: Map<String, Any?>? = null,
│        skipTriggers: Boolean? = null
│    ): Unit
//
// События SDK (один подписчик)
└─ val eventSDKFunctions: Events
   // Подписаться на события SDK
   ├─ fun subscribe(newSubscriber: (DataClasses.Event) -> Unit): Unit
   // Отписаться от событий SDK
   └─ fun unsubscribe(): Unit
```

class AltcraftConfiguration

```
com.altcraft.sdk.config
└─ class AltcraftConfiguration private constructor(...)
   // Класс инициализации конфигурации Altcraft SDK:
   // URL API, ресурсный токен, сведения о приложении, флаг использования foreground
service
   // и настройки push-уведомлений (канал, модули получателя и приоритет провайдеров).

   ├─ class Builder(
   │     apiUrl: String,                          // Базовый URL Altcraft API (обязательный)
   │     icon: Int? = null,                       // ID ресурса иконки уведомлений
(опционально)
   │     rToken: String? = null,                  // Ролевой токен
   │     appInfo: DataClasses.AppInfo? = null,    // Метаданные Firebase Analytics
(опционально)
   │     enableLogging: Boolean = false,          // Включает/отключает ведение логов
(опционально)
   │     providerPriorityList: List<String>? = null, // Приоритет провайдеров push
уведомлений(опционально)
   │     pushReceiverModules: List<String>? = null,  // Пакеты модулей, где может быть
переопределён PushReceiver (опционально)
   │     pushChannelName: String? = null,         // Имя канала push-уведомлений
(опционально)
   │     pushChannelDescription: String? = null  // Описание канала push-уведомлений
(опционально)
   │   )
   │   └─ fun build(): AltcraftConfiguration     // Построить валидную конфигурацию
   │
   ├─ fun getApiUrl(): String                     // Возвращает базовый URL Altcraft API
   ├─ fun getIcon(): Int?                         // Возвращает ID иконки для уведомлений
(опционально)
   ├─ fun getRToken(): String?                    // Возвращает ресурсный токен
(опционально)
   ├─ fun getAppInfo(): DataClasses.AppInfo?      // Сведения о приложении (ID/IID/версия)
(опционально)
   ├─ fun getEnableLogging(): Boolean             // Включает/отключает ведение логов
(опционально)
   ├─ fun getProviderPriorityList(): List<String>? // Приоритет провайдеров push-
уведомлений (опционально)
   ├─ fun getPushReceiverModules(): List<String>? // Модули с переопределениями
PushReceiver (опционально)
   ├─ fun getPushChannelName(): String?           // Имя канала уведомлений (опционально)
   └─ fun getPushChannelDescription(): String?    // Описание канала уведомлений
(опционально)
```

Event Classes

```
Event
// Базовое событие SDK (универсальная телеметрия).
function: String
eventCode: Int?
eventMessage: String?
eventValue: Map<String, Any?>?
date: Date

Error
// Ошибка SDK, наследует Event.
function: String
eventCode: Int?
eventMessage: String?
eventValue: Map<String, Any?>?
date: Date

RetryError
// Ошибка запроса, для которого предусмотрен автоматический повтор и выполнение
function: String
eventCode: Int?
eventMessage: String?
eventValue: Map<String, Any?>?
date: Date
```

Data Classes

```
// Токен push-провайдера устройства
data class TokenData(
    provider: String,
    token: String
)

// Опциональные данные для Firebase Analytics (если используется)
data class AppInfo(
    appID: String,
    appIID: String,
    appVer: String
)
asMap(): Map<String, String>

// Обёртка ответа API вместе с HTTP-кодом
data class ResponseWithHttpCode(
    httpCode: Int?,
    response: Response?
)

// Ответ запроса
data class Response(
    error: Int? = null,
    @SerialName("error_text") errorText: String? = null,
    profile: ProfileData? = null
)

// Данные профиля пользователя
data class ProfileData(
    id: String? = null,
    status: String? = null,
    @SerialName("is_test") isTest: Boolean? = null,
    subscription: SubscriptionData? = null
)

// Текущая подписка профиля
data class SubscriptionData(
    @SerialName("subscription_id") subscriptionId: String? = null,
    @SerialName("hash_id") hashId: String? = null,
    provider: String? = null,
    status: String? = null,
    fields: Map<String, JsonElement>? = null,
    cats: List<CategoryData>? = null
)

// Категория подписки (имя, заголовок, флаги)
data class CategoryData(
    name: String?,
    title: String? = null
```

```
    title: String? = null,
    steady: Boolean? = null,
    active: Boolean?
)

// Подписка на канал Email (добавляется мобильным событием)
data class EmailSubscription(
    @SerialName("resource_id") resourceId: Int,
    email: String,
    status: String? = null,
    priority: Int? = null,
    @SerialName("custom_fields") customFields: Map<String, @Contextual Any?>? = null,
    cats: List<String>? = null,
    @SerialName("channel") channel: String = "email"
)

// Подписка на канал SMS (добавляется мобильным событием)
data class SmsSubscription(
    @SerialName("resource_id") resourceId: Int,
    phone: String,
    status: String? = null,
    priority: Int? = null,
    @SerialName("custom_fields") customFields: Map<String, @Contextual Any?>? = null,
    cats: List<String>? = null,
    @SerialName("channel") channel: String = "sms"
)

// Подписка на push-канал (добавляется мобильным событием)
data class PushSubscription(
    @SerialName("resource_id") resourceId: Int,
    provider: String,
    @SerialName("subscription_id") subscriptionId: String,
    status: String? = null,
    priority: Int? = null,
    @SerialName("custom_fields") customFields: Map<String, @Contextual Any?>? = null,
    cats: List<String>? = null,
    @SerialName("channel") channel: String = "push"
)

// Подписка с cc_data (Telegram, WhatsApp, Viber, Notify)
data class CcDataSubscription(
    @SerialName("resource_id") resourceId: Int,
    @SerialName("channel") channel: String,
    @SerialName("cc_data") ccData: JsonObject,
    status: String? = null,
    priority: Int? = null,
    @SerialName("custom_fields") customFields: Map<String, @Contextual Any?>? = null,
    cats: List<String>? = null
)
```

```
// Контейнер UTM-параметров для атрибуции мобильных событий (все поля опциональны)
data class UTM(
    campaign: String? = null,
    content: String? = null,
    keyword: String? = null,
    medium: String? = null,
    source: String? = null,
    temp: String? = null
)
```

Interfaces

JWTInterface:

```
JWTInterface
// Доступ к текущему JWT
└─ fun getJWT(): String?
   // Возвращает JWT или null
```

FCMInterface:

```
FCMInterface
// Контракт операций с FCM-токеном
├─ suspend fun getToken(): String?
│  // Возвращает токен FCM или null
└─ suspend fun deleteToken(completion: (Boolean) -> Unit)
   // Удаляет токен FCM
```

HMSInterface:

```
HMSInterface
// Контракт операций с HMS-токеном
├─ suspend fun getToken(context: Context): String?
│  // Возвращает токен HMS или null
└─ suspend fun deleteToken(context: Context, complete: (Boolean) -> Unit)
   // Удаляет токен HMS
```

RuStoreInterface:

```
RustoreInterface
// Контракт операций с RuStore-токеном
├─ suspend fun getToken(): String?
│  // Возвращает токен RuStore или null
└─ suspend fun deleteToken(complete: (Boolean) -> Unit)
   // Удаляет токен RuStore
```

Subscription:

```
Subscription (sealed interface)
// Базовый контракт модели подписки для всех каналов
├─ val resourceId: Int
│  // Обязательный ID ресурса Altcraft (JSON: "resource_id")
├─ val status: String?
│  // Необязательный статус подписки
├─ val priority: Int?
│  // Необязательный приоритет
├─ val customFields: Map<String, Any?>?
│  // Необязательные стандартные и кастомные поля (JSON: "custom_fields")
├─ val cats: List<String>?
│  // Необязательные категории подписки
└─ val channel: String
   // Обязательный тип канала (например, "email", "sms", "push", "telegram_bot", ...)
```

Firebase Cloud Messaging

Настройки проекта Firebase Cloud Messaging

Создайте проект Firebase:

Добавьте приложение к проекту:

Зарегистрируйте приложение, заполнив необходимые поля с информацией о нем:

Выполните загрузку `google-services.json`:

Настройка ресурса Altcraft

В платформе Altcraft создайте ресурс или добавьте к существующему новый канал коммуникации — Push. Затем в настройках ресурса выберите платформу **Android** и активируйте переключатель Firebase Cloud Messaging:

Чуть ниже перейдите в настройки Firebase и укажите **идентификатор вашего проекта** (`messagingSenderId`) и **идентификатор отправителя** (`projectId`). Также необходимо загрузить Firebase-файл с приватным ключом (.json):

Создать такой Firebase-файл можно в Firebase в разделе **Project setting** -> **Service account**.

После внесения изменений не забудьте **сохранить** ресурс.

Интеграция FCM в проект приложения

Ручная интеграция

Переместите загруженный файл `google-services.json` в в корневой каталог модуля приложения (app):

Затем добавьте плагины и зависимости в Gradle-файлы:

- **`build.gradle.kts (project level)`**

Добавьте плагин `google-services` в файл `build.gradle.kts` уровня проекта (project-level):

```
plugins {

    id("com.google.gms.google-services") version "4.4.3" apply false

}
```

TIP: Обратите внимание

Информация об актуальной версии плагина доступна здесь.

- **`build.gradle.kts (applevel)`**

Добавьте плагин `google-services` в файл `build.gradle.kts` уровня приложения (app-level)

```
plugins {

    id("com.google.gms.google-services")

}
```

Также добавьте зависимости:

С использованием firebase-bom (позволяет автоматически указывает версии firebase-библиотек)

```
dependencies {
 implementation(platform("com.google.firebase:firebase-bom:34.0.0"))
 implementation("com.google.firebase:firebase-messaging")
}
```

Без использования firebase-bom

```
dependencies {
 implementation("com.google.firebase:firebase-messaging:25.0.0")
}
```

> **TIP**
>
> Информация об актуальных версии библиотек доступна на этой странице Firebase.

После добавления плагинов и зависимостей выполните **синхронизацию** изменений Gradle.

Автоматическая настройка Android Studio

Перейдите в **Tools** -> **Firebase** -> **CloudMessaging** -> **Set up Firebase Cloud Messaging**:

Создание класса, расширяющего `FirebaseMessagingService()`

Создайте расширяющий `FirebaseMessagingService()` класс:

```
/**
 * FCM service for handling push tokens and messages.
 */
class FCMService : FirebaseMessagingService() {

    /**
     * Called when a new FCM token is generated.
     *
     * @param token The new FCM token.
     */
    override fun onNewToken(token: String) {
        super.onNewToken(token)
    }

    /**
     * Called when a push message is received.
     *
     * Forwards the message to all receivers with additional metadata.
     *
     * @param message The received [RemoteMessage].
     */
    override fun onMessageReceived(message: RemoteMessage) {
        super.onMessageReceived(message)

       // Сообщение
    }
}
```

Обновление `manifest` приложения

Добавьте информацию о `FCMService` в `AndroidManifest.xml` приложения:

```
        <service
            android:name="com.altcraft.fcm.FCMService"
            android:exported="false">
            <intent-filter>
            </intent-filter>
        </service>
```

Замените `com.altcraft.fcm.FCMService` на ваш пакет и класс.

Huawei Mobile Services

Настройки проекта Huawei Mobile Services

Создайте проект AppGalleryConnect. Для этого перейдите в раздел **Разработка и сервисы** -> **Добавить проект**:

Добавьте новое приложение в проект:

Укажите данные о вашем приложении:

Выполните загрузку файла `agconnect-services.json`:

Вернитесь к настройкам проекта и добавьте **Отпечаток подписи SHA-256**. Получить его можно, выполнив консольную команду `./gradlew signingReport` в корневом каталоге проекта приложения:

В левой боковой панели выберите **Push Kit** в разделе **Рост**:

Нажмите **Включить** -> **Выберите место хранения данных**:

Настройка ресурса Altcraft

В платформе Altcraft создайте ресурс или добавьте к существующему новый канал коммуникации — Push. Затем в настройках ресурса выберите платформу **Android** и активируйте переключатель Huawei Mobile Services:

Чуть ниже перейдите в настройки Huawei и укажите **ID приложения**, **ID клиента OAuth 2.0** и **секрет клиента OAuth 2.0**:

> **TIP**
>
> Все эти данные можно найти в разделе **Данные приложения** в Huawei Mobile Services (там же, куда на прошлом этапе был добавлен Отпечаток подписи SHA-256):

После внесения изменений не забудьте **сохранить** ресурс.

Интеграция Huawei Push Kit в проект приложения

Переместите загруженный файл `agconnect-services.json` в корневой каталог модуля приложения (app):

Добавьте репозитории и зависимости в Gradle-файлы:

`settings.gradle.kts (project setting)`

```
    dependencyResolutionManagement {
    repositories {

    // Прочие репозитории..

    maven("https://developer.huawei.com/repo/")

   }
  }
```

---

build.gradle.kts (project level)

```
   buildscript {
    repositories {
        maven("https://developer.huawei.com/repo/")
    }
    dependencies {
        classpath("com.android.tools.build:gradle:8.10.1") // Важно! укажите Android Gradle
Plugin (AGP) той версии, которую вы используете.
        classpath("com.huawei.agconnect:agcp:1.9.1.301")
    }
  }
```

---

build.gradle.kts (applevel)

```
  plugins {
    id("com.huawei.agconnect")
  }

  dependencies {
   implementation("com.huawei.hms:push:6.11.0.300")
   implementation("com.huawei.agconnect:agconnect-core:1.7.3.302")
  }
```

TIP: Обратите внимание

Информация об актуальных версиях зависимостей доступна на страницах Huawei Developer:

- AppGalleryConnect
- HMS Push Core

После добавления репозитория и зависимостей выполните **синхронизацию** изменений Gradle.

Создание класса, расширяющего `HmsMessageService()`

Создайте расширяющий `HmsMessageService()` класс:

```
/**
 * HMS service for handling push tokens and incoming notifications.
 *
 * Extends [HmsMessageService] and overrides key HMS callback methods.
 */
class HMSService : HmsMessageService() {
    /**
     * Called when a new HMS token is generated.
     *
     * @param token The new HMS token.
     */
    override fun onNewToken(token: String) {
        super.onNewToken(token)
    }

    /**
     * Called when a push message is received from HMS.
     *
     * Forwards the message with additional metadata to all receivers.
     *
     * @param message The received [RemoteMessage].
     */
    override fun onMessageReceived(message: RemoteMessage) {
         // Сообщение
    }
}
```

Обновление `manifest` приложения

Добавьте информацию о `HMSService` в `AndroidManifest.xml` приложения:

```
        <service
            android:name="com.altcraft.hms.HMSService"
            android:exported="false">
            <intent-filter>
            </intent-filter>
        </service>
```

Замените `com.altcraft.hms.HMSService` на ваш пакет и класс.

RuStore

> **INFO: Обратите внимание**
>
> Приложение RuStore должно быть **установлено** на устройство.

Настройки проекта RuStore dev

Перейдите в консоль разработчика:

Добавьте приложение в разделе **Приложение**:

Перейдите в раздел **Инструменты** консоли разработчика:

Перейдите к созданному приложению. Выберите пункт **Push-уведомления**:

Создайте проект:

В нем укажите укажите:

- название проекта;
- пакет приложения;
- отпечаток подписи SHA-256 приложения. Получить его можно, выполнив консольную команду `./gradlew signingReport` в корневом каталоге проекта приложения.

Внутри созданного проекта создайте сервисный токен:

Настройка ресурса Altcraft

В платформе Altcraft создайте ресурс или добавьте к существующему новый канал коммуникации — Push. Затем в настройках ресурса выберите платформу **Android** и активируйте переключатель RuStore:

Чуть ниже перейдите в настройки RuStore и укажите **идентификатор проекта** и **пароль**:

После внесения изменений не забудьте **сохранить** ресурс.

Интеграция RuStore Push Messaging в проект приложения

Добавьте репозитории и зависимости в Gradle-файлы:

`settings.gradle.kts (project setting)`

```
    dependencyResolutionManagement {
    repositories {

    // Прочие репозитории..

    maven("https://artifactory-external.vkpartner.ru/artifactory/maven")

   }
  }
```

`build.gradle.kts (applevel)`

```
  dependencies {

   implementation("ru.rustore.sdk:pushclient:6.10.0")
  }
```

TIP: Обратите внимание

Информация об актуальных версиях зависимостей доступна здесь.

После добавления репозитория и зависимостей выполните **синхронизацию** изменений Gradle.

Инициализация `RuStorePushClient`

Выполните инициализацию `RuStorePushClient` в `Application.onCreate()`:

```
class App : Application() {
    override fun onCreate() {
        super.onCreate()
        RuStorePushClient.init(this, "project_ID")

    }
}
```

Создание класса, расширяющего `RuStoreMessagingService()`

Создайте расширяющий `RuStoreMessagingService()` класс:

```
/**
 * RuStore service for handling push notifications.
 *
 * Extends [RuStoreMessagingService] and overrides key callbacks.
 */
class RuStoreService : RuStoreMessagingService() {
    override fun onNewToken(token: String) {
        super.onNewToken(token)
    }

    /**
     * Called when a push message is received.
     *
     * Forwards the message to all receivers with added metadata.
     *
     * @param message The received RemoteMessage.
     */
    override fun onMessageReceived(message: RemoteMessage) {
      // Сообщение
    }
}
```

Обновление `manifest` приложения

Добавьте информацию о `RuStoreService` в `AndroidManifest.xml` приложения:

```
        <service
            android:name="com.altcraft.rustore.RuStoreService"
            android:exported="true"
            tools:ignore="ExportedService">
            <intent-filter>
            </intent-filter>
        </service>
```

Замените `com.altcraft.rustore.RuStoreService` на ваш пакет и класс. Обратите внимание — в манифесте обязательно должен быть указан параметр `android:exported="true"`.

TIP: Условия корректной работы уведомлений RuStore

- Приложение RuStore установлено на устройство;
- Приложению RuStore предоставлено разрешение на показ уведомлений;
- Приложению Rustore предоставлено разрешение на прекращение экономии заряда батареи в фоновом режиме.

Быстрый старт

В статье приведен пример быстрой интеграции Altcraft mSDK с Apple Push Notification System (APNS) и Firebase Cloud Messaging (FCM). После выполнения указанных шагов ваше приложение сможет получать push-уведомления Altcraft.

Для использования Huawei Mobile Services необходимо добавить соответствующие SDK в проект и реализовать их интерфейсы, используя тот же подход, что показан для APNS и FCM.

Шаг 0. Предварительные условия

- Получен доступ к APNS получен на сайте;
- Приложение настроено в проекте Firebase, `GoogleServiceInfo.plist` загружен.

Шаг 1. Подключение Altcraft mSDK к приложению

Добавьте пакеты из репозиториев с помощью Swift Package Manager:

- Репозиторий Altcraft SDK;
- Репозиторий Firebase.

Шаг 2. Добавление `GoogleServiceInfo` в приложение

Добавьте `GoogleServiceInfo.plist` в папку вашего приложения:

Шаг 3. Подготовка приложения

Настройте следующие параметры таргета приложения:

- **General:**
    - Проверьте, что библиотека **Altcraft** добавлена в **Frameworks, Libraries, and Embedded Content** таргета приложения:
- Проверьте, что библиотека **FirebaseMessaging** добавлена в **Frameworks, Libraries, and Embedded Content** таргета приложения:
- **Signing & Capabilities:**
    - **PushNotifications;**
    - **AppGroups** — укажите идентификатор для группы (добавление идентификатора App Group необходимо обмена информацией с Notification Service Extension);
    - `Background Modes` — выберите **Background fetch**, **Remoute notification**, **Background processing**.
- **Info:**
    - Добавьте ключи и значения:

Ключ	Тип	Значение
Permitted background task scheduler identifiers	Array	“lib.Altcraft.bgTask.systemControl”
FirebaseAppDelegateProxyEnabled	Boolean	NO

Первый ключ необходим для регистрации `bgTask`-задачи, которая будет выполнять повтор неудачных запросов к серверу в background-режиме. Второй ключ — для отключения автоматической подмены (method swizzling) методов `AppDelegate`, которую Firebase Messaging выполняет по умолчанию.

Шаг 4. Реализация JWTInterface

Выполните реализацию протокола SDK, предназначенного для предоставления JWT-токена. Для этого создайте класс, реализующий `JWTInterface`, и переопределите `getJWT()`:

```
class JWTProvider: JWTInterface {
    func getToken() -> String? {
         //ваш код, возвращающий JWT
      }
}
```

TIP: К сведению

`getJWT()` — синхронная функция. Поток выполнения SDK будет приостановлен до получения JWT. Рекомендуется, чтобы `getJWT()` возвращал значение немедленно — из кэша (in-memory, `SharedPreferences` или `EncryptedSharedPreferences`) — это ускорит выполнение запросов. Желательно подготовить актуальный JWT как можно раньше (на старте приложения) и сохранить его в кэш, чтобы при обращении SDK токен был доступен без задержек. При отсутствии значения допустимо вернуть `nil`.

Шаг 5. Реализация FCMInterface

Выполните реализацию протокола SDK, предназначенного для предоставления и удаления FCM-токена. Для этого создайте класс, реализующий `FCMInterface`, и переопределите `getToken()`, `deleteToken()`.

```
class FCMProvider: FCMInterface {

    /// Возвращает текущий FCM-токен
   func getToken(completion: @escaping (String?) -> Void) {

        ///токен apns, полученный из userDefault
        let apnsToken = getAPNsTokenDataFromUserDefaults()

        ///установить токен apns для fcm перед запросом токена fcm
        Messaging.messaging().apnsToken = apnsToken

        ///функция запроса токена fcm
        Messaging.messaging().token { token, error in
            if error != nil {
                completion(nil)
            } else {
                completion(token)
            }
        }
    }

    /// Удаляет текущий токен FCM
    func deleteToken(completion: @escaping (Bool) -> Void) {
        Messaging.messaging().deleteToken { error in
            if error != nil {
                completion(false)
            } else {
                completion(true)
            }
        }
    }
}
```

Шаг 6. Реализация APNSInterface

Выполните реализацию протокола SDK, предназначенного для предоставления APNS-токена. Для этого создайте класс, реализующий `APNSInterface`, и переопределите `getToken()`:

```
class APNSProvider: APNSInterface {
    /// Извлекает текущий токен APNs из локального хранилища
    func getToken(completion: @escaping (String?) -> Void) {
        let token = getAPNsTokenFromUserDefault()  - передайте токен apns
        completion(token)
    }
}
```

Шаг 7. Настройка Altcraft SDK и Firebase SDK в `AppDelegate`

Altcraft SDK:

```
- выполните передачу идентификатора `AppGroups` в Altcraft SDK;
- выполните регистрацию `backgroundTasks` Altcraft SDK;
- установите JWT-, APNS-, FCM-провайдеров;
- выполните регистрацию в центре уведомлений.
```

FirebaseSDK:

```
- выполните установку конфигурации;
```

Пример правильной настройки Altcraft SDK и Firebase SDK в `AppDelegate`

```
class AppDelegate: UIResponder, UIApplicationDelegate,  UNUserNotificationCenterDelegate {
    func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        //установка конфигурации FirebaseSDK
        FirebaseApp.configure()

        //AltcraftSDK
        ///установка идентификатора AppGroup
        AltcraftSDK.shared.setAppGroup(groupName: "group.your_company.name")
        ///регистрации BGTAsk
        AltcraftSDK.shared.backgroundTasks.registerBackgroundTask()
        ///установка JWTProvider
        AltcraftSDK.shared.setJWTProvider(provider: JWTProvider())
        ///установка APNSProvider
        AltcraftSDK.shared.pushTokenFunctions.setAPNSTokenProvider(APNSProvider())
        ///установка FCMProvider
        AltcraftSDK.shared.pushTokenFunctions.setFCMTokenProvider(FCMProvider())
        ///регистрация в центре уведомлений
        AltcraftSDK.shared.notificationManager.registerForPushNotifications(for:
application)

        return true
    }

    func application(
        _ application: UIApplication,
        didRegisterForRemoteNotificationsWithDeviceToken apnsToken: Data
    ) {
        //сохраните apns токен в userDefault
        setAPNsTokenInUserDefault(apnsToken)
    }
}
```

```
- выполните установку `delegate`;
- передайте APNS-токен в Firebase в функции
`AppDelegate.application(_:didFinishLaunchingWithOptions:)`. Это гарантирует правильную
инициализацию Firebase и доступность FCM-токена в background-режиме, например, во время
выполнения `BGTask`. Так как APNS-токен не доступен в
`AppDelegate.application(_:didFinishLaunchingWithOptions:)`, используете сохраненное
значение, например, в `UserDefaults`.
```

Шаг 8. Создание Notification Service Extension

Создайте расширение приложения Notification Service Extension:

- Выполните **File** — **New** — **Target** — **Notification Service Extension**;
- Выберите название (Product Name) для таргета расширения;
- Активируйте его.

Выполните в Notification Service Extension:

- **General:**
    - Укажите **Minimum Deployments** — это параметр сборки в Xcode, который определяет минимальную версию операционной системы, на которой будет работать Notification Service Extension;
    - Добавьте библиотеку Altcraft в разделе **Frameworks, Libraries and Embedded Content**.
- **Signing & Capabilities:** - **AppGroups** — укажите идентификатор `AppGroup`.

Затем настройте SDK в `UNNotificationServiceExtension`. Для этого замените весь автоматически сгенерированный код NSE на код из примера ниже:

Пример реализации `UNNotificationServiceExtension`

```
class NotificationService: UNNotificationServiceExtension {

    /// - important! Set app groups identifier.
    let appGroupID =  "group.your.id"

    /// - important! Set the provider to JWT if you are using JWT authentication.
    let jwtProvider = JWTProvider()

    override func didReceive(_ request: UNNotificationRequest, withContentHandler
contentHandler: @escaping (UNNotificationContent) -> Void) {

        AltcraftNSE.shared.setAppGroup(groupName: appGroupID)
        AltcraftNSE.shared.setJWTProvider(provider: jwtProvider)

        contentHandler(request.content)

        if AltcraftNSE.shared.isAltcraftPush(request) {
            AltcraftNSE.shared.handleNotificationRequest(request: request, contentHandler:
contentHandler)
        } else {
            contentHandler(request.content)
        }
    }
    override func serviceExtensionTimeWillExpire() {
        AltcraftNSE.shared.serviceExtensionTimeWillExpire()
    }
}
```

TIP: Обратите внимание

Если вы не используете JWT-аутентификацию, исключите функцию установки JWT-провайдера `AltcraftSDK.shared.setJWTProvider(provider: jwtProvider)`.

Шаг 9. Инициализация SDK

Создайте переменную конфигурации и выполните инициализацию с помощью функции `initialization()`, где `ios-firebase` выбран приоритетным провайдером:

```
let config = AltcraftConfiguration.Builder()
.setApiUrl("your_api_url")
.setProviderPriorityList([Constants.ProviderName.firebase])
.build()

AltcraftSDK.shared.initialization(configuration: config)
```

В данном примере в качестве приоритетного провайдера установлен `ios-firebase`. Вы также можете установить APNS в качестве приоритетного провайдера:

```
let config = AltcraftConfiguration.Builder()
.setApiUrl("your_api_url")
.setProviderPriorityList([Constants.ProviderName.apns])
.build()

AltcraftSDK.shared.initialization(configuration: config)
```

Вы можете не устанавливать параметр `setProviderPriorityList`, тогда будет использоваться приоритет по умолчанию — `ios-apns -> ios-firebase -> ios-huawei`:

```
let config = AltcraftConfiguration.Builder()
.setApiUrl("your_api_url")
.build()

AltcraftSDK.shared.initialization(configuration: config)
```

Шаг 10. Выполнение подписки на push-уведомления

Выполните подписку на push-уведомления с помощью функции `pushSubscribe()`, передав необходимые параметры:

```
AltcraftSDK.shared.pushSubscriptionFunctions.pushSubscribe(
 profileFields: ["_fname": "user_first_name", "_lname": "user_last_name"]
)
```

Конфигурация SDK

Предварительные условия

- SDK провайдеров push-уведомлений интегрированы в проект приложения (см. инструкции по интеграции push-провайдеров).
- Выполнена загрузка пакета Altcraft с помощью Swift Package Manager или CocoaPods.

Подготовка приложения

Настройка таргета

Настройте следующие параметры таргета приложения:

- ☑ **General**

При использовании Swift Package Manager проверьте, что библиотека Altcraft добавлена в **Frameworks, Libraries, and Embedded Content** таргета приложения.

- ☑ **Signing & Capabilities**

Добавьте следующие Capabilities:

- **PushNotifications;**
- **AppGroups**;
- **Background Modes** (Background fetch, Background processing).

- ☑ **Info**

Добавьте key **Permitted background task scheduler identifiers** и value **“lib.Altcraft.bgTask.systemControl”**. Это необходимо для регистрации `bgTask`-задачи, которая будет выполнять повтор неудачных запросов к серверу в background-режиме.

- ☑ **Опционально:**

При использовании Firebase Cloud Messaging добавьте ключ `FirebaseAppDelegateProxyEnabled (Boolean)` с значением `NO`. Это отключит автоматическую подмену (method swizzling) методов `AppDelegate`, которую Firebase Messaging выполняет по умолчанию.

Настройка AppDelegate

Далее необходимо выполнить в `AppDelegate.application(_:didFinishLaunchingWithOptions:)` следующие действия:

- передать идентификатор `AppGroup` в SDK
- зарегистрировать `BGTask` (фоновые задачи SDK)
- (опционально) инициализировать функции `UNUserNotificationCenter` на стороне SDK

Установка идентификатора AppGroup

Добавление идентификатора `AppGroup` необходимо для обмена информацией с Notification Service Extension и требуется для корректной работы SDK. Выполняется с помощью функции `setAppGroup()`:

```
AltcraftSDK.shared.setAppGroup(groupName: String)
```

Регистрация BGTask

Регистрация фоновых задач SDK необходима для повторной отправки неуспешных запросов в фоновом режиме. Выпоняется с помощью функции `registerBackgroundTask()`:

```
AltcraftSDK.shared.backgroundTasks.registerBackgroundTask()
```

Инициализация функций UNUserNotificationCenter (опционально)

SDK содержит класс `NotificationManager` который выполняет следующие задачи:

- настраивает `UNUserNotificationCenter` и запрашивает разрешение на уведомление (alert/sound/badge);
- обеспечивает вызов `UIApplication.registerForRemoteNotifications()` и гарантирует корректную регистрацию в APNs;
- отображает уведомления в foreground;
- обрабатывает клики по уведомлениям и кнопкам — выполняет переход на указанный URL или открывает приложение;
- регистрирует push-событие открытия (“open”).

Для использования класса SDK `NotificationManager` вызовите функцию в `AppDelegate.application(_:didFinishLaunchingWithOptions:)`:

```
func registerForPushNotifications(for application: UIApplication, completion: ((_ granted: Bool, _ error: Error?) -> Void)? = nil): Void
```

Данная функция назначает делегата, запрашивает разрешение у пользователя и регистрирует приложение для получения push-уведомлений на стороне SDK.

Пример корректного заполнения `AppDelegate.application`

```
class AppDelegate: UIResponder, UIApplicationDelegate, UNUserNotificationCenterDelegate {
    func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        let appGroup = "group.your.id"

        //ID App Group Setup
        AltcraftSDK.shared.setAppGroup(groupName: appGroup)

        //Backgroung SDK task registration
        AltcraftSDK.shared.backgroundTasks.registerBackgroundTask()

        //SDK UNUserNotificationCenter functions initialization
        AltcraftSDK.shared.notificationManager.registerForPushNotifications(for:
application)

        //Other functions

        return true
    }
    //Other functions AppDelegate
}
```

Использование этого класса **опционально**, вы можете заменить его своей реализацией. Для этого:

- назначьте делегата центра уведомлений: `UNUserNotificationCenter.current().delegate = self`;
- запросите разрешения на уведомления и вызовите `UIApplication.shared.registerForRemoteNotifications()` строго на главном потоке;
- реализуйте метод `userNotificationCenter(_:willPresent:withCompletionHandler:)` для отображения уведомлений в foreground (banner/alert/sound/badge);
- вызовите `AltcraftSDK.shared.pushEventFunctions.openEvent(from:)` в `userNotificationCenter(_:didReceive:withCompletionHandler:)` при обработке нажатия на уведомление (регистрация события “open”);
- реализуйте собственную логику обработки кликов по уведомлению (URL/Deep Link/кнопки), не забудьте при этом продумать fallback-сценарий (навигацию по умолчанию при отсутствии валидной ссылки).

Настройка JWT-протокола (опционально)

JWTInterface — протокол запроса JWT-токена. Предоставляет актуальный JWT-токен из приложения по запросу SDK. Реализация данного протокола требуется, если используется JWT-аутентификация API-запросов. JWT подтверждает, что пользовательские идентификаторы аутентифицированы приложением.

Реализация JWT-аутентификации обязательна, если используется тип матчинга отличный от push-данных из подписки (например, идентификатор пользователя — email или телефон).

Протокол SDK:

```
public protocol JWTInterface {
    func getToken() -> String?
}
```

Реализация на стороне приложения

```
class JWTProvider: JWTInterface {
    func getToken() -> String? {
         //ваш код, который возвращает JWT
      }
}
```

Регистрация провайдера в `application(_:didFinishLaunchingWithOptions:)`

В `AppDelegate` укажите:

```
class AppDelegate: UIResponder, UIApplicationDelegate, MessagingDelegate,
UNUserNotificationCenterDelegate {
  func application(_ application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    //Другой код

    AltcraftSDK.setJWTProvider(provider: JWTProvider())
  }
  //Другие функции
}
```

TIP: К сведению

`getJWT()` — синхронная функция. Поток выполнения SDK будет приостановлен до получения JWT. Рекомендуется, чтобы `getJWT()` возвращал значение немедленно — из кэша (in-memory, `SharedPreferences` или `EncryptedSharedPreferences`) — это ускорит выполнение запросов. Желательно подготовить актуальный JWT как можно раньше (на старте приложения) и сохранить его в кэш, чтобы при обращении SDK токен был доступен без задержек. При отсутствии значения допустимо вернуть `nil`.

Протоколы запроса и удаления push-токена провайдера

Данная реализация протоколов запроса и удаления push-токена гарантирует использование актуального токена и обеспечивает возможность динамической смены провайдеров по требованию клиента. Реализуйте протоколы только для тех push-провайдеров, которые используются в вашем проекте.

Регистрируйте провайдеры в `application(_:didFinishLaunchingWithOptions:) (AppDelegate)`. Данная точка регистрации гарантирует раннюю, однократную и детерминированную регистрацию при старте процесса,

в том числе в background.

Apple Push Notifcation service

APNSInterface — протокол запроса push-токена APNs. В данном протоколе отсутствует функция удаления токена, в отличии от других протоколов push-провайдеров.

Протокол SDK:

```
public protocol APNSInterface {

    func getToken(completion: @escaping (String?) -> Void)
}
```

Рекомендуемая реализация на стороне приложения

```
class APNSProvider: APNSInterface {
    func getToken(completion: @escaping (String?) -> Void) {
        //ваш код, который возвращает токен APNs
    }
}
```

Регистрация провайдера в `application(_:didFinishLaunchingWithOptions:)`

```
class AppDelegate: UIResponder, UIApplicationDelegate, MessagingDelegate,
UNUserNotificationCenterDelegate {
  func application(
  _ application: UIApplication,
  didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) ->
Bool {
    //другой код

    AltcraftSDK.pushTokenFunctions.setAPNSTokenProvider(APNSProvider())
  }
  //другие функции
}
```

Firebase Cloud Messaging

FCMInterface — протокол запроса и удаления push-токена Firebase Cloud Messaging.

Протокол SDK:

```
public protocol FCMInterface {
    func getToken(completion: @escaping (String?) -> Void)

    func deleteToken(completion: @escaping (Bool) -> Void)
}
```

TIP: Обратите внимание

Рекомендуемая реализация на стороне приложения

```
class FCMProvider: FCMInterface {

    /// Извлекает текущий токен FCM
   func getToken(completion: @escaping (String?) -> Void) {

        ///Токен APNs (в виде данных), извлеченный из userDefault
        let apnsToken = getAPNsTokenDataFromUserDefaults()

        ///Установите токен APNs для FCM перед запросом токена FCM
        Messaging.messaging().apnsToken = apnsToken

        ///Функция запроса токена FCM
        Messaging.messaging().token { token, error in
            if error != nil {
                completion(nil)
            } else {
                completion(token)
            }
        }
    }

    /// Удаляет текущий токен FCM
    func deleteToken(completion: @escaping (Bool) -> Void) {
        Messaging.messaging().deleteToken { error in
            if error != nil {
                completion(false)
            } else {
                completion(true)
            }
        }
    }
}
```

Регистрация провайдера в `application(_:didFinishLaunchingWithOptions:)`

```
class AppDelegate: UIResponder, UIApplicationDelegate, MessagingDelegate,
UNUserNotificationCenterDelegate {
  func application(
  _ application: UIApplication,
  didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) ->
Bool {
    //Другой код

    AltcraftSDK.pushTokenFunctions.setFCMTokenProvider(FCMProvider())
  }
  //Другие функции
}
```

Huawei Mobile Services

HMSInterface — протокол запроса и удаления push-токена Huawei Mobile Services.

Протокол SDK:

```
public protocol HMSInterface {

    func getToken(completion: @escaping (String?) -> Void)

    func deleteToken(completion: @escaping (Bool) -> Void)
}
```

Рекомендуемая реализация на стороне приложения

```
class HMSProvider: HMSInterface {

    func getToken(completion: @escaping (String?) -> Void) {

        //переменная, содержащая токен APNs (в виде строки), который позже
        //передается в HmsInstanceId.getInstance().getToken(apnsToken)
        guard let apnsToken = getAPNsTokenFromUserDefault() else {
            completion(nil)
            return
        }

        let token = HmsInstanceId.getInstance().getToken(apnsToken)
        completion(token)
    }


    func deleteToken(completion: @escaping (Bool) -> Void) {
        HmsInstanceId.getInstance().deleteToken()
        completion(true)
    }
}
```

Регистрация провайдера в `application(_:didFinishLaunchingWithOptions:)`

```
class AppDelegate: UIResponder, UIApplicationDelegate, UNUserNotificationCenterDelegate {
  func application(
  _ application: UIApplication,
  didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) ->
Bool {
    //Другой код

    AltcraftSDK.pushTokenFunctions.setHMSTokenProvider(HMSProvider())
  }
  //Другие функции
}
```

Подготовка Notification Service Extension

Добавление расширения NSE необходимо для поддержки Rich Push и гарантированной регистрации событий доставки.

Создайте расширение приложения Notification Service Extension:

```
- Выполните **File** — **New** — **Target** — **Notification Service Extension**;
- Выберите название (Product Name) для таргета расширения;
- Активируйте его.
```

Выполните в Notification Service Extension:

- ☑ **General:**
- Укажите **Minimum Deployments** — это параметр сборки в Xcode, который определяет минимальную версию операционной системы, на которой будет работать Notification Service Extension;
- Добавьте библиотеку Altcraft в разделе **Frameworks, Libraries and Embedded Content**.
- ☑ **Signing & Capabilities:**
- Укажите идентификатор `AppGroup`.

Затем настройте SDK в `UNNotificationServiceExtension`:

- Импортируйте библиотеку `Altcraft`;
- В функции `didReceive(_ request: UNNotificationRequest, withContentHandler contentHandler: @escaping (UNNotificationContent) -> Void)`:
- Передайте идентификатор `appGroup` в функцию `AltcraftNSE.shared.setAppGroup(groupName: appGroupsName)`. Данный идентификатор должен соответствовать идентификатору переданному в SDK в таргете приложения;
- Установите провайдер JWT (если используется JWT-аутентификация запросов): `AltcraftNSE.shared.setJWTProvider(provider: jwtProvider)`;
- Выполните проверку источника уведомления с помощью функции `AltcraftNSE.shared.isAltcraftPush(request)`;
- После проверки источника уведомления, передайте `UNNotificationRequest` и `contentHandler` в функцию `AltcraftNSE.shared.handleNotificationRequest(request: request, contentHandler: contentHandler)`, если источник уведомления — Altcraft.

Приведенный ниже пример реализации `UNNotificationServiceExtension` можно использовать как готовый класс. Замените весь автоматически сгенерированный код на эту реализацию, если вам не требуется дополнительная логика. Если вы уже используете `UNNotificationServiceExtension`, интегрируйте функции Altcraft в свою реализацию.

Пример реализации `UNNotificationServiceExtension`

```
class NotificationService: UNNotificationServiceExtension {

    /// - важно! Укажите идентификатор App Group.
    let appGroupID =  "group.your.id"

    /// - важно! Установите провайдер JWT, если используется JWT-аутентификация.
    let jwtProvider = JWTProvider()

    override func didReceive(_ request: UNNotificationRequest, withContentHandler
contentHandler: @escaping (UNNotificationContent) -> Void) {

        AltcraftNSE.shared.setAppGroup(groupName: appGroupID)
        AltcraftNSE.shared.setJWTProvider(provider: jwtProvider)

        if AltcraftNSE.shared.isAltcraftPush(request) {
            AltcraftNSE.shared.handleNotificationRequest(request: request, contentHandler:
contentHandler)
        } else {
            contentHandler(request.content)
        }
    }
    override func serviceExtensionTimeWillExpire() {
        AltcraftNSE.shared.serviceExtensionTimeWillExpire()
    }
}
```

TIP: Обратите внимание

Таргет приложения и таргет Notification Service Extension работают в отдельных процессах и не разделяют общие объекты. Поэтому для корректной обработки уведомлений SDK необходимо зарегистрировать `JWTProvider` и идентификатор App Group также в расширении NSE — например, в методе `didReceive(_ request: UNNotificationRequest, withContentHandler contentHandler: @escaping (UNNotificationContent) -> Void)`. Это обеспечивает доступ SDK к токену авторизации при отправке push-события доставки (“delivery_event”).

Инициализация SDK

Параметры инициализации

Для передачи параметров конфигурации испоользуется класс `AltcraftConfiguration`:

```
public final class AltcraftConfiguration {
    private let apiUrl: String
    private let rToken: String?
    private var appInfo: AppInfo?
    private var providerPriorityList: [String]?
    private let enableLogging: Bool?
}
```

Сборка файла конфигурации происходит после вызова функции `build()` класса `AltcraftConfiguration.Builder(): AltcraftConfiguration.Builder().build()`.

Описание параметров:

apiUrl `String`

Обязательный: **Да**
Описание: URL конечной точки Altcraft API

---

rToken `String?`

По умолчанию: `nil`
Обязательный: Нет
Описание: Ролевой токен Altcraft (идентифицирует ресурс, БД или аккаунт). Используется, если единственный тип матчинга — push-токен устройства, выданный провайдером.

---

appInfo `AppInfo?`

По умолчанию: `nil`
Обязательный: Нет
Описание: Базовые метаданные приложения для Firebase Analytics. Для установки значения этого параметра используйте публичную структуру SDK `AppInfo`:

```
public struct AppInfo: Codable {
    /// Firebase app_id
    public var appID: String
    /// Firebase app_instance_id
    public var appIID: String
    /// Firebase app_version
    public var appVer: String
}
```

---

enableLogging `Bool?`

По умолчанию: `nil`
Обязательный: Нет
Описание: Включает/отключает ведение логов. Логи можно найти по префиксу `[Altcraft SDK]`.

---

providerPriorityList `[String]?`

По умолчанию: `nil`
Обязательный: Нет
Описание: Список приоритетов провайдеров. Используется для автоматического обновления push-токена подписки, если токен более приоритетного провайдера недоступен. Приоритет определяется индексом в списке: элемент с индексом **0** — самый приоритетный.

Пример использования:

```
providerPriorityList = [
    Constants.ProviderName.apns,
    Constants.ProviderName.firebase,
    Constants.ProviderName.huawei
]
```

При такой настройке:

- SDK сначала запросит токен **APNs**; если APNs недоступен — **FCM**; если FCM недоступен — **HMS**;
- Работает при условии, если в приложении реализованы протоколы соответствующих провайдеров.

Для удобства установки значения параметра SDK содержит публичные константы. Все константы находятся в `enum Constants`:

```
public enum Constants {
    public enum ProviderName {
        /// Provider name for Firebase
        public static let firebase = "ios-firebase"
        /// Provider name for APNs
        public static let apns = "ios-apns"
        /// Provider name for HMS
        public static let huawei = "ios-huawei"
    }
}
```

Если значение не задано, приоритет провайдеров следующий:

```
apns –> firebase –> huawei
```

Значение параметра может содержать **один элемент** — в этом случае будет использоваться только один провайдер, независимо от доступности push-токена.

Параметр можно не указывать, если:

- в проекте используется только один провайдер;
- приоритет по умолчанию соответствует требованиям.

Выполнение инициализации

TIP: Обратите внимание

Вызывайте `AltcraftSDK.shared.initialization()` тогда, когда это необходимо, но только **после** установки идентификатора `AppGroup` и провайдеров, реализующих протоколы SDK. Запросы следует выполнять **после** установки конфигурации.

Для инициализации SDK используется функция `public func initialization(configuration: AltcraftConfiguration?, completion: ((Bool) -> Void)? = nil)`:

```
AltcraftSDK.shared.initialization(configuration: AltcraftConfiguration?, completion:
((Bool) -> Void)? = nil)
```

Функция `initialization(configuration: AltcraftConfiguration?, completion: ((Bool) -> Void)? = nil)` ожидает файл конфигурации `AltcraftConfiguration`, для которого установлены необходимые значения и выполнена сборка:

```
 let config = AltcraftConfiguration.Builder()
        .setApiUrl("your apiUrl")
        .setRToken("your rToken")
        .setAppInfo(AppInfo(appID: "your appID", appIID: "your appIID", appVer: "your
appVer"))
        .setProviderPriorityList([Constants.ProviderName.firebase])
        .build()
```

Пример правильного порядка инициализации SDK в `AppDelegate.application(_:didFinishLaunchingWithOptions:)` (после установки идентификатора `AppGroup` и регистрации провайдеров)

```
class AppDelegate: UIResponder, UIApplicationDelegate, UNUserNotificationCenterDelegate {
    func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        // Firebase configuration initialization (if used)

        let appGroup = "group.your.id"

        AltcraftSDK.shared.setAppGroup(groupName: appGroup)
        AltcraftSDK.shared.backgroundTasks.registerBackgroundTask()
        AltcraftSDK.shared.setJWTProvider(provider: JWTProvider())
        AltcraftSDK.shared.pushTokenFunctions.setAPNSTokenProvider(APNSProvider())
        AltcraftSDK.shared.pushTokenFunctions.setFCMTokenProvider(FCMProvider())
        AltcraftSDK.shared.pushTokenFunctions.setHMSTokenProvider(HMSProvider())
        AltcraftSDK.shared.notificationManager.registerForPushNotifications(for:
application)

        let config = AltcraftConfiguration.Builder()
            .setApiUrl("your apiUrl")
            .setRToken("your rToken")
            .setAppInfo(AppInfo(appID: "your appID", appIID: "your appIID", appVer: "your
appVer"))
            .setProviderPriorityList([Constants.ProviderName.firebase])
            .build()

        AltcraftSDK.shared.initialization(configuration: config) { success in
            if success {
               // Actions after successful initialization
            }
        }

        // Other functions

        return true
    }
    // Other functions AppDelegate
}
```

Функционал SDK

TIP

Предварительно вам необходимо настроить SDK для работы с вашим приложением. Подробная инструкция находится здесь

Работа со статусами подписки

Изменение статуса подписки

```
AltcraftSDK
//   Singleton, the entry point to the SDK
└── static let shared: AltcraftSDK
    // Push subscription managment
    └── pushSubscriptionFunctions: PublicPushSubscriptionFunctions
        // Subscribe (status = SUBSCRIBED)
        ├── pushSubscribe(
        │       sync: Bool = true,
        │       profileFields: [String: Any?]? = nil,
        │       customFields: [String: Any?]? = nil,
        │       cats: [CategoryData]? = nil,
        │       replace: Bool? = nil,
        │       skipTriggers: Bool? = nil
        │   ): Void
        // Unsubscribe (status = UNSUBSCRIBED)
        ├── pushUnSubscribe(
        │       sync: Bool = true,
        │       profileFields: [String: Any?]? = nil,
        │       customFields: [String: Any?]? = nil,
        │       cats: [CategoryData]? = nil,
        │       replace: Bool? = nil,
        │       skipTriggers: Bool? = nil
        │   ): Void
        // Suspend (status = SUSPENDED)
        └── pushSuspend(
                sync: Bool = true,
                profileFields: [String: Any?]? = nil,
                customFields: [String: Any?]? = nil,
                cats: [CategoryData]? = nil,
                replace: Bool? = nil,
                skipTriggers: Bool? = nil
            ): Void
```

Функции изменения статуса подписки:

- `func pushSubscribe()` — выполняет подписку на push-уведомления;
- `func pushUnSubscribe()` — отменяет подписку на push-уведомления;
- `func pushSuspend()` — приостанавливает подписку на push-уведомления (уведомления не приходят, но при этом не создается событие отписки в профиле пользователя);
- `func unSuspendPushSubscription()` — используется для создания `LogIn`-, `LogOut`-переходов.

Функции этой группы имеют одинаковую сигнатуру, содержащую следующие параметры:

---

sync: `Bool`

По умолчанию: `true`
Обязательный: Нет
Описание: Флаг, устанавливающий синхронность выполнения запроса.

Успешное выполнение запроса:

В случае успешного выполнения запроса данной группы функций будет создано событие с кодом **230, 231 или 232**, содержащее значение `event.value`, определяемое в зависимости от флага синхронизации:

Если `sync == true`

```
ResponseWithHttpCode
  ├ httpCode: 200
  └ response
     ├ error: 0
     ├ errorText: ""
     └ profile
        ├ id: "your id"
        ├ status: "subscribed"
        ├ isTest: false
        └ subscription
           ├ subscriptionId: "your subscriptionId"
           ├ hashId: "c52b28d2"
           ├ provider: "ios-apns"
           ├ status: "subscribed"
           ├ fields
           │  ├ _os_ver: {"raw":"18.6.2","ver":"[\"18.0\", \"6.0\", \"2.0\"]"}
           │  ├ _device_type: "mob"
           │  ├ _ad_track: false
           │  ├ _device_name: "iPhone"
           │  ├ _os_language: "en"
           │  ├ _os_tz: "+0300"
           │  ├ _os: "IOS"
           │  └ _device_model: "iPhone14,7"
           └ cats
              └ [ { name: "developer_news", title: "dev_news", steady: false, active:
false } ]
```

При **синхронном** запросе в значении события `event.value` по ключу `"response_with_http_code"` доступны:

- **httpCode** – транспортный код ответа;
- **Response** – public struct, содержащий:
  - `error: Int?` — внутренний код ошибки сервера *(0, если ошибок нет)*,
  - `errorText: String?` — текст ошибки *(пустая строка, если ошибок нет)*,
  - `profile: ProfileData?` — данные профиля, если запрос успешный:
    - информация о профиле (**ProfileData**)
    - подписка (**SubscriptionData**)
    - категории подписки (**CategoryData**)
    - если запрос завершился с ошибкой, то вернется только `profile = nil`

▶ Структуры данных

Если `sync == false`

```
ResponseWithHttpCode
  ├─ httpCode: Int?
  └─ response: Response?
      ├─ error: Int?
      ├─ errorText: String?
      └─ profile: ProfileData? = nil
```

При **асинхронном** запросе в значении события `event.value` по ключу `"response_with_http_code"` доступны:

- **httpCode** – транспортный код ответа;
- **Response** – public struct, содержащий:
  - `error: Int?` — внутренний код ошибки сервера *(0, если ошибок нет)*,
  - `errorText: String?` — текст ошибки *(пустая строка, если ошибок нет)*,
  - `profile: ProfileData?` — данные профиля, для асинхронного запроса всегда равен `nil`.

Выполнение запроса с ошибкой:

Если запрос данной группы функций завершился ошибкой, будет создано событие со следующими кодами:

Операции без автоматического повтора попытки на стороне SDK:

- **430** – подписка на уведомления;
- **431** — приостановка подписки;
- **432** — отписка.

Операции с автоматическим повтором попытки на стороне SDK:

- **530** – подписка на уведомления;
- **531** — приостановка подписки;
- **532** — отписка.

Содержимое события:

- только `httpCode`, если сервер Altcraft был **недоступен**;
- `error` и `errorText`, если сервер **вернул ошибку**.

Получение значений событий

```
AltcraftSDK.shared.eventSDKFunctions.subscribe { event in
    let codes: Set<Int> = [
        230, 231, 232,
        430, 431, 432,
        531, 532, 533
    ]

    guard let code = event.eventCode, codes.contains(code) else { return }
    guard let responseWithHttp = event.value?["response_with_http_code"] as?
ResponseWithHttp else { return }

    let httpCode = responseWithHttp.httpCode

    let response = responseWithHttp.response
    let error = response?.error
    let errorText = response?.errorText

    let profile = response?.profile
    let profileId = profile?.id
    let profileStatus = profile?.status
    let profileIsTest = profile?.isTest

    let subscription = profile?.subscription
    let subscriptionId = subscription?.subscriptionId
    let hashId = subscription?.hashId
    let provider = subscription?.provider
    let subscriptionStatus = subscription?.status

    let fields = subscription?.fields

    let cats = subscription?.cats
    let firstCat = cats?.first
    let catName = firstCat?.name
    let catTitle = firstCat?.title
    let catSteady = firstCat?.steady
    let catActive = firstCat?.active
}
```

Поля `fields`, содержащиеся в `subscription` могут содержать ваши пользовательские поля подписки. Они имеют тип `[String: JSONValue]?`. `public enum JSONValue` содержит вспомогательные функции позволяющие упростить получение значений fields:

```
    public var stringValue: String? {
        if case let .string(value) = self { return value }
        return nil
    }

    /// Returns the numeric value if case is `.number`, otherwise `nil`.
    public var numberValue: Double? {
        if case let .number(value) = self { return value }
        return nil
    }

    /// Returns the bool value if case is `.bool`, otherwise `nil`.
    public var boolValue: Bool? {
        if case let .bool(value) = self { return value }
        return nil
    }

    /// Returns the object dictionary if case is `.object`, otherwise `nil`.
    public var objectValue: [String: JSONValue]? {
        if case let .object(value) = self { return value }
        return nil
    }

    /// Returns the array if case is `.array`, otherwise `nil`.
    public var arrayValue: [JSONValue]? {
        if case let .array(value) = self { return value }
        return nil
    }
```

Пример получения значения поля `field "_device_name"`:

```
AltcraftSDK.shared.eventSDKFunctions.subscribe { event ->
    if event.eventCode == 230 {
        if let responseWithHttp = event.value?["response_with_http_code"] as?
ResponseWithHttp {
            let response = responseWithHttp.response
            let profile = response?.profile
            let subscription = profile?.subscription

            // extract fields?["_device_name"] as a string value
            if let deviceName = subscription?.fields?["_device_name"]?.stringValue {
                    print(deviceName)
            }
        }
    }
}
```

---

profileFields: `[String: Any?]?`

По умолчанию: `nil`
Обязательный: Нет
Описание: Словарь, содержащий поля профиля.

Параметр может принимать как **системные поля** (например, `_fname` — имя или `_lname` — фамилия), так и **опциональные** (заранее создаются вручную в интерфейсе платформы). Допустимые типы значений (JSON-совместимые):

- String
- Bool
- Числа
- `nil`
- Объекты `[String: Any]`
- Массивы

Если передано невалидное опциональное поле, запрос завершится с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: with field "название_поля": Incorrect field
```

---

customFields: `[String: Any?]?`

По умолчанию: `nil`
Обязательный: Нет
Описание: Словарь, содержащий поля подписки.

Параметр может принимать как **системные поля** (например, `_device_model` — модель устройства или `_os` — операционная система), так и **опциональные** (заранее создаются вручную в интерфейсе платформы). Допустимые типы значений:

- String
- Bool
- Int
- Float
- Double
- `nil`

Если передано невалидное опциональное поле, запрос завершится с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: field "название_поля" is not valid: failed
convert custom field
```

Вложенные объекты, массивы и коллекции **не допускаются**.

> **TIP: Обратите внимание**
>
> Большая часть системных полей подписки автоматически собирается SDK и добавляется к push-запросам. К таким системным полям относятся: `"_os"`, `"_os_tz"`, `"_os_language"`, `"_device_type"`, `"_device_model"`, `"_device_name"`, `"_os_ver"`, `"_ad_track"`, `"_ad_id"`.

---

cats: `[CategoryData]?`

По умолчанию: `nil`
Обязательный: Нет
Описание: Категории подписок.

Структура категории:

```
public struct CategoryData: Codable {
    public var name: String?
    public var title: String?
    public var steady: Bool?
    public var active: Bool?
}
```

При отправке push-запроса с указанием категорий используйте только поля `name` (название категории) и `active` (статус активности категории), другие поля не используется в обработке запроса. Поля `title` и `steady` заполняются при получении информации о подписке.

Пример запроса:

```
let cats: [CategoryData] = [CategoryData(name: "football", title: nil, steady: nil, active: true)]
```

Пример ответа:

```
[ { name: "developer_news", title: "dev_news", steady: false, active: false } ]
```

Категории используемые в запросе должны быть предварительно созданы и добавлены к ресурсу в платформе Altcraft. Если в запросе будет использовано поле, которое не добавлено в ресурс — запрос вернется с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: field "subscriptions.cats" is not valid:
category not found in resource
```

---

replace: `Bool?`

По умолчанию: `nil`
Обязательный: Нет
Описание: При активации флага все подписки других профилей с тем же push-токеном в текущей базе данных переводятся в статус `unsubscribed` после успешного запроса.

---

skipTriggers: `Bool?`

По умолчанию: `nil`
Обязательный: Нет
Описание: При активации флага профиль, содержащий данную подписку, будет игнорироваться в триггерах рассылок и сценариев.

Примеры реализации запроса

Пример выполнения запроса подписки на push-уведомления

Минимальная рабочая настройка:

```
AltcraftSDK.shared.pushSubscriptionFunctions.pushSubscribe()
```

Передача всех доступных параметров:

```
AltcraftSDK.shared.pushSubscriptionFunctions.pushSubscribe(
    sync: true,
    profileFields: ["_fname":"Andrey", "_lname":"Pogodin"],
    customFields:  ["developer":true],
    cats: [CategoryData(name: "developer_news", active: true)],
    replace: false,
    skipTriggers:  false
)
```

TIP: Обратите внимание

Для `pushSubscribe`, `pushSuspend`, `pushUnSubscribe` предусмотрен автоматический повтор запроса со стороны SDK если http-код ответа находится в диапазоне **500…599**. Запрос не повторяется, если код ответа в этот диапазон не входит.

Функция `unSuspendPushSubscription()`

Функция `func unSuspendPushSubscription()` предназначена для создания `LogIn`-, `LogOut`-переходов. Она работает следующим образом:

- проводит поиск подписок с тем же push-токеном, что и текущий, не относящихся к профилю, на который указывает текущий JWT-токен;
- меняет статус для найденных подписок с `subscribed` на `suspended`;
- меняет статус в подписках профиля, на который указывает текущий JWT, с `suspended` на `subscribed` (если профиль на который указывает JWT существует и в нем содержатся подписки);
- возвращает public struct `ResponseWithHttpCode?`, в котором `response.profile` — текущий профиль, на который указывает JWT (если профиля не существует, то вместо `response.profile` вернется `nil`).

Пример использования:

```
AltcraftSDK.shared.pushSubscriptionFunctions.unSuspendPushSubscription(completion:
@escaping (ResponseWithHttp?) -> Void)
```

Рекомендуемая реализация `LogIn`-, `LogOut`-переходов

`LogIn`-переход:

- Анонимный пользователь входит в приложение. Данному пользователю присвоен `JWT_1`, указывающий на базу данных **#1Anonymous**;
- Выполнена подписка на push-уведомления, профиль создан в базе данных **#1Anonymous**;
- Пользователь регистрируется, ему присваивается `JWT_2`, указывающий на базу данных **#2Registered**;
- Вызывается функция `unSuspendPushSubscription()` — подписка анонимного пользователя в базе данных **#1Anonymous** приостанавливается;
- Выполняется поиск профиля в базе данных **#2Registered** для восстановления подписки;
- Так как подписки с таким push-токеном в базе данных **#2Registered** не существует, функция `unSuspendPushSubscription()` вернет `nil`;
- После получения значения `nil` можно выполнить запрос на подписку `pushSubscribe()`, который создаст новый профиль в базе **#2Registered**.

`LogOut`-переход:

- Пользователь выполнил выход из профиля на стороне приложения (`LogOut`);
- Пользователю присваивается `JTW_1`, указывающий на базу данных **#1Anonymous**;
- Вызывается функция `unSuspendPushSubscription()`, которая приостановит подписку базе данных в **#2Registered** и сменит статус подписки в базе **#1Anonymous** на `subscribed`;
- Запрос вернет `#1Anonymous != nil` — подписка уже существует, новая не требуется.

Пример реализации:

```
func logIn() {
    JWTManager.shared.setRegJWT()
    // JWT is set for an authorized user
    AltcraftSDK.shared.pushSubscriptionFunctions.unSuspendPushSubscription { result in
        if result?.httpCode == 200, result?.response?.profile?.subscription == nil {
            AltcraftSDK.shared.pushSubscriptionFunctions.pushSubscribe(
                // specify the required parameters
            )
        }
    }
}

func logOut() {
    JWTManager.shared.setAnonJWT()
    // JWT is set for an anonymous user
    AltcraftSDK.shared.pushSubscriptionFunctions.unSuspendPushSubscription { result in
        if result?.httpCode == 200, result?.response?.profile?.subscription == nil {
            AltcraftSDK.shared.pushSubscriptionFunctions.pushSubscribe(
                // specify the required parameters
            )
        }
    }
}
```

Запрос статуса подписки

```
AltcraftSDK
//   Singleton, the entry point to the SDK
└── static let shared: AltcraftSDK
    // Subscription managment
    └── pushSubscriptionFunctions: PublicPushSubscriptionFunctions
        // Last profile subscription status
        ├── getStatusOfLatestSubscription(
        │       completion: @escaping (ResponseWithHttp?) -> Void
        │   ): Void
        // Subscription status corresponding to the current token/provider
        ├── getStatusForCurrentSubscription(
        │       completion: @escaping (ResponseWithHttp?) -> Void
        │   ): Void
        // Status of the latest subscription for the provider (if nil — the current one is
used)
        └── getStatusOfLatestSubscriptionForProvider(
                provider: String? = nil,
                completion: @escaping (ResponseWithHttp?) -> Void
            ): Void
```

Функции запроса статуса подписки:

- `func getStatusOfLatestSubscription()` — статус последней подписки профиля;
- `func getStatusOfLatestSubscriptionForProvider()` — статус подписки для текущего токена/провайдера;
- `func getStatusForCurrentSubscription()` — статус последней подписки по провайдеру. Если указан `nil`, то используется текущий.

`public func getStatusOfLatestSubscription(completion: @escaping (ResponseWithHttp?) -> Void): Void`

Функция получения статуса последней подписки профиля. В `completion` передаётся объект `ResponseWithHttp?`, содержащий `response?.profile?.subscription` — последнюю созданную подписку в профиле. Если такой подписки не существует, будет передан `nil`.

Пример использования:

```
AltcraftSDK.shared.pushSubscriptionFunctions.getStatusOfLatestSubscription(completion:
@escaping (ResponseWithHttp?) -> Void)
```

`public func getStatusForCurrentSubscription(completion: @escaping (ResponseWithHttp?) -> Void): Void`

Функция получения статуса подписки для текущего токена/провайдера. В `completion` передаётся объект `ResponseWithHttp?`, содержащий `response?.profile?.subscription` — подписку, найденную по текущему push-токену и провайдеру. Если такой подписки не существует, будет передан `nil`.

Пример использования:

```
AltcraftSDK.shared.pushSubscriptionFunctions.getStatusForCurrentSubscription(completion:
@escaping (ResponseWithHttp?) -> Void)
```

`public func getStatusOfLatestSubscriptionForProvider(provider: String? = nil, completion: @escaping (ResponseWithHttp?) -> Void): Void`

Функция получения статуса последней подписки по провайдеру. В `completion` передаётся объект `ResponseWithHttp?`, содержащий `response?.profile?.subscription` — последнюю созданную подписку с указанным провайдером push-уведомлений. Если провайдер не указан, используется провайдер текущего токена. Если такой подписки не существует, будет передан `nil`.

Пример использования:

```
AltcraftSDK.shared.pushSubscriptionFunctions.getStatusOfLatestSubscriptionForProvider(provi
der: String? = nil, completion: @escaping (ResponseWithHttp?) -> Void)
```

Ниже представлен пример извлечения данных о профиле, подписке и категориях из ответа функций получения статуса. Данный подход актуален для всех функций получения статуса:

Данные из функций получения статуса

```
AltcraftSDK.shared.pushSubscriptionFunctions.getStatusForCurrentSubscription{
responseWithHttp in
    // HTTP code
    let httpCode = responseWithHttp?.httpCode

    // Response
    let response = responseWithHttp?.response
    let error = response?.error
    let errorText = response?.errorText

    // Profile
    let profile = response?.profile
    let profileId = profile?.id
    let profileStatus = profile?.status
    let profileIsTest = profile?.isTest

    // Subscription
    let subscription = profile?.subscription
    let subscriptionId = subscription?.subscriptionId
    let hashId = subscription?.hashId
    let provider = subscription?.provider
    let subscriptionStatus = subscription?.status

    // Fields (dictionary [String: JSONValue])
    let fields = subscription?.fields

    // Cats (array of CategoryData)
    let cats = subscription?.cats

    // CategoryData (every element of cats array)
    let firstCat = cats?.first
    let catName = firstCat?.name
    let catTitle = firstCat?.title
    let catSteady = firstCat?.steady
    let catActive = firstCat?.active
}
```

В случае успешно выполненного запроса будет создано событие с кодом 234. В случае ошибки — событие с кодом 434.

Управление push-токенами провайдеров

```
AltcraftSDK
//   Singleton, the entry point to the SDK
└── static let shared: AltcraftSDK
    // Managing push tokens and providers (FCM / HMS / APNS)
    └── pushTokenFunctions: PublicPushTokenFunctions
        // Manually save the token (String for FCM/HMS, Data for APNS)
        ├── setPushToken(
        │       provider: String,
        │       pushToken: Any?
        │   ): Void
        // Asynchronously retrieve the current push token (completion is optional)
        ├── getPushToken(
        │       completion: ((TokenData?) -> Void)? = nil
        │   ): Void
        // Set the Firebase Cloud Messaging provider (nil — remove)
        ├── setFCMTokenProvider(
        │       _ provider: FCMInterface?
        │   ): Void
        // Set the Huawei Mobile Services provider (nil — remove)
        ├── setHMSTokenProvider(
        │       _ provider: HMSInterface?
        │   ): Void
        // Set the Apple Push Notification service provider (nil — remove)
        ├── setAPNSTokenProvider(
        │       _ provider: APNSInterface?
        │   ): Void
        // Apply a new provider priority list and initiate token update
        ├── changePushProviderPriorityList(
        │       _ list: [String]
        │   ): Void
        // Delete the push token of the specified provider and call completion
        ├── deleteDeviceToken(
        │       provider: String,
        │       completion: @escaping () -> Void
        │   ): Void
        // Force token update
        └── forcedTokenUpdate(
                completion: (() -> Void)? = nil
            ): Void
```

Функции для работы с токеном провайдера в SDK:

- `func setPushToken()` — ручная установка push-токена устройства и провайдера в `UserDefaults`;
- `func getPushToken()` — получение текущего push-токена;
- `func setAPNSTokenProvider()` — установка и снятие push-токена APNS;
- `func setFCMTokenProvider()` — установка и снятие push-токена FCM;
- `func setHMSTokenProvider()` — установка и снятие push-токена HSM;

- `func changePushProviderPriorityList()` — функция, позволяющая выполнить динамическую смену провайдера с обновлением push-токена подписки;
- `func deleteDeviceToken()` — удаление push-токена указанного провайдера;
- `func forcedTokenUpdate()` — удаление текущего push-токена с последующим обновлением.

`public func setPushToken(provider: String, pushToken: Any?): Void`

Функция предназначена для ручной установки push-токена устройства и провайдера в `UserDefaults`. Используется как упрощенный вариант передачи push-токена в SDK без реализации протоколов провайдеров.

Не рекомендуется использовать эту функцию для передачи токена. Рекомендуемый подход передачи push-токена в SDK — реализация `FCMInterface`, `HMSInterface` или `APNSInterface`.

Пример использования:

```
AltcraftSDK.shared.pushTokenFunctions.setPushToken(provider: String, pushToken: Any?)
```

`public func getPushToken(completion: ((TokenData?) -> Void)? = nil): Void`

Функция передает в `completion` объект `TokenData(provider: String, token: String)`, содержащий текущий push-токен устройства и его провайдера. Если push-токен недоступен, в `completion` будет передано `nil`.

Пример использования:

```
AltcraftSDK.shared.pushTokenFunctions.getPushToken(completion: ((TokenData?) -> Void)? =
nil)
```

Пример запроса **получения данных push-токена**:

```
AltcraftSDK.shared.pushTokenFunctions.getPushToken{ data in
    let provider = data?.provider
    let token = data?.token
}
```

`public func setFCMTokenProvider(_ provider: FCMInterface?): Void`

Функция устанавливает или снимает push-токен провайдера Firebase Cloud Messaging. Чтобы отключить провайдера, передайте `nil`.

Пример использования:

```
AltcraftSDK.shared.pushTokenFunctions.setFCMTokenProvider(_ provider: FCMInterface?)
```

TIP: Важно

Вызывайте `setFCMTokenProvider()` в `AppDelegate.application(_:didFinishLaunchingWithOptions:)` до вызова `AltcraftSDK.initialization()`. Это гарантирует регистрацию на старте процесса приложения, независимо от состояния жизненного цикла других компонентов и того, запущено приложение в foreground или background.

`public func setHMSTokenProvider(_ provider: HMSInterface?): Void`

Функция устанавливает или снимает push-токен провайдера Huawei Mobile Services. Чтобы отключить провайдера, передайте `nil`.

```
AltcraftSDK.shared.pushTokenFunctions.setHMSTokenProvider(_ provider: HMSInterface?)
```

TIP: Важно

Вызывайте `setHMSTokenProvider()` в `AppDelegate.application(_:didFinishLaunchingWithOptions:)` до вызова `AltcraftSDK.initialization()`. Это гарантирует регистрацию на старте процесса приложения, независимо от состояния жизненного цикла других компонентов и того, запущено приложение в foreground или background.

`public func setAPNSTokenProvider(_ provider: APNSInterface?): Void`

Функция устанавливает или снимает push-токен провайдера Apple Push Notification service. Чтобы отключить провайдера, передайте `nil`.

```
AltcraftSDK.shared.pushTokenFunctions.setHMSTokenProvider(_ provider: APNSInterface?)
```

TIP: Важно

Вызывайте `setAPNSTokenProvider()` в `AppDelegate.application(_:didFinishLaunchingWithOptions:)` до вызова `AltcraftSDK.initialization()`. Это гарантирует регистрацию на старте процесса приложения, независимо от состояния жизненного цикла других компонентов и того, запущено приложение в foreground или background.

`public func changePushProviderPriorityList(_ list: [String]): Void`

Функция, позволяющая выполнить динамическую смену push-провайдера с обновлением токена подписки. Для этого необходимо передать новый массив с другим порядком провайдеров. Например: `[Constants.ProviderName.firebase, Constants.ProviderName.apns, Constants.ProviderName.huawei]`.

Пример использования:

```
AltcraftSDK.shared.pushTokenFunctions.changePushProviderPriorityList(_ list: [String])
```

`public func deleteDeviceToken(provider: String, completion: @escaping () -> Void): Void`

Функция удаления push-токена для указанного провайдера. Токен инвалидируется и удаляется из локального кеша на устройстве и с сервера push-провайдера. После удаления можно запросить новый токен.

Пример использования:

```
AltcraftSDK.shared.pushTokenFunctions.deleteDeviceToken(provider: String, completion:
@escaping () -> Void)
```

`public func forcedTokenUpdate(completion: (() -> Void)? = nil): Void`

Функция удаления текущего push-токена с его последующим обновлением.

Пример использования:

```
AltcraftSDK.shared.pushTokenFunctions.forcedTokenUpdate(completion: (() -> Void)? = nil)
```

Пример регистрации провайдеров

Мы **не рекомендуем** использовать функцию `setPushToken` для установки push-токена. Вместо этого отдельно настройте функции получения токена для каждого используемого провайдера. Ниже указан пример реализации этого метода:

Рекомендованный способ регистрации провайдеров в `AppDelegate`

```
class AppDelegate: UIResponder, UIApplicationDelegate, MessagingDelegate,
UNUserNotificationCenterDelegate {
    func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        let appGroup = "your appGroup id"

        AltcraftSDK.shared.setAppGroup(groupName: appGroup)
        AltcraftSDK.shared.backgroundTasks.registerBackgroundTask()
        AltcraftSDK.shared.setJWTProvider(provider: JWTProvider())

        //apns provider
        AltcraftSDK.shared.pushTokenFunctions.setAPNSTokenProvider(APNSProvider())

        //fcm provider
        AltcraftSDK.shared.pushTokenFunctions.setFCMTokenProvider(FCMProvider())

        //hms provider
        AltcraftSDK.shared.pushTokenFunctions.setHMSTokenProvider(HMSProvider())

        AltcraftSDK.shared.notificationManager.registerForPushNotifications(for:
application)

        let config = AltcraftConfiguration.Builder().setApiUrl("your api url").build()
        AltcraftSDK.shared.initialization(configuration: config)

        //other functions

        return true
    }
}
```

Работа с push-уведомлениями

Инициализация push-уведомлений

Для инициализации push-уведомления в `application(_:didFinishLaunchingWithOptions:)` вызовите:

`AltcraftSDK.shared.notificationManager.registerForPushNotifications(for: application)`

Метод назначает `UNUserNotificationCenter.current().delegate` на `NotificationManager` со стороны Altcraft SDK, запрашивает разрешение на показ уведомлений (alert/sound/badge) и регистрирует приложение в APNS через `registerForRemoteNotifications()`.

TIP: Важно

Системный диалог разрешений отображается только при первом запросе (или если статус ещё не определён). Если пользователь уже давал ответ ранее, диалог повторно **не появится.**

Передача push-уведомлений в SDK

SDK содержит класс AltcraftNSE, который предназначен для использования внутри Notification Service Extension. Он содержит функции для обработки входящие push-уведомления (`UNNotificationRequest`) и обеспечивает регистрацию мобильных событий в NSE.

```
// Used inside Notification Service Extension (NSE) — facade for handling Altcraft push notifications,
// detecting Altcraft-specific requests, and registering mobile events without direct dependency on UIApplication
AltcraftNSE
// Shared singleton instance for centralized NSE logic
├─ public static let shared: AltcraftNSE
// Provides access to mobile event registration API
├─ public let mobileEventFunctions: PublicMobileEventFunctions
// Check whether the notification request belongs to Altcraft
├─ public func isAltcraftPush(_ request: UNNotificationRequest) -> Bool
// Handle an incoming Altcraft notification request and deliver the modified content
├─ public func handleNotificationRequest(
│     request: UNNotificationRequest,
│     contentHandler: @escaping (UNNotificationContent) -> Void
│  ): Void
// Called when NSE is about to time out — returns the best available notification content
├─ public func serviceExtensionTimeWillExpire(): Void
// Set the App Group identifier and initialize Core Data in the shared container
├─ public func setAppGroup(groupName: String): Void
// Register a JWT provider for secure token management in Altcraft SDK
└─ public func setJWTProvider(provider: JWTInterface): Void
```

Функции класса `AltcraftNSE`:

- `func isAltcraftPush()` — функция проверки источника уведомления;
- `func handleNotificationRequest()` — функция, принимающая `UNNotificationRequest` из Notification Service Extension для его дальнейшей обработки на стороне SDK;
- `func serviceExtensionTimeWillExpire()` — функция, вызывающаяся по истечении времени работы Notification Service Extension (~30 секунд).

`public func isAltcraftPush(_ request: UNNotificationRequest) -> Bool`

Функция SDK, проверяющая, является ли источником уведомления Altcraft по маркеру в `userInfo`.

Пример использования:

```
AltcraftNSE.shared.isAltcraftPush(_ request: UNNotificationRequest) -> Bool
```

`public func handleNotificationRequest(_ request: UNNotificationRequest, withContentHandler contentHandler: @escaping (UNNotificationContent) -> Void): Void`

Функция SDK, принимающая `UNNotificationRequest` из Notification Service Extension для его обработки на стороне SDK и дальнейшего показа уведомления.

Пример использования:

```
AltcraftNSE.shared.handleNotificationRequest(_ request: UNNotificationRequest,
withContentHandler contentHandler: @escaping (UNNotificationContent) -> Void)
```

`public func serviceExtensionTimeWillExpire()`

Функция, которая вызывается системой по истечении времени работы Notification Service Extension (~30 секунд).

Пример использования:

```
AltcraftNSE.shared.serviceExtensionTimeWillExpire()
```

`public func setAppGroup(groupName: String): Void`

Функция для добавления идентификатора `AppGroup` (обязательная установка).

Пример использования:

```
AltcraftNSE.shared.setAppGroup(groupName: "group.your.app.id")
```

`public func setJWTProvider(provider: JWTInterface): Void`

Функция для установки JWT-провайдера:

```
AltcraftNSE.shared.setJWTProvider(provider: jwtProvider)
```

Получение foreground-уведомления в приложении (опционально)

По умолчанию этот функционал **можно не настраивать**: если вы делегировали обработку push-уведомлений SDK (через `registerForPushNotifications`), то `NotificationManager` сам:

- будет установлен как `UNUserNotificationCenterDelegate`,
- обработает входящие уведомления,
- и покажет уведомление в foreground со стандартными параметрами (banner/alert + badge/sound).

Настройка, описанная ниже, требуется **только если**:

- вам нужен доступ к содержимому уведомления в foreground (например, для аналитики/кастомного UI),
- или вы хотите **самостоятельно управлять показом** уведомления (показывать/скрывать, менять presentation options).

Чтобы настроить поведение **push-уведомлений в foreground**, используйте `NotificationManager`:

```
// Доступ к менеджеру push-уведомлений SDK
let manager = AltcraftSDK.shared.notificationManager

// Управление обработкой push-уведомлений в foreground.
// true  – обработка и выбор способа показа передаётся приложению.
// false – SDK использует стандартную логику показа уведомлений.
manager.customPushProcessing = false

// Колбэк для обработки push-уведомлений в foreground.
// Вызывается при получении уведомления, когда приложение активно.
// completion должен быть вызван только при customPushProcessing = true.
manager.onForegroundNotification = { notification, complete in
    // Пользовательская логика обработки уведомления

    // Важно: вызывать complete(...) нужно только если manager.customPushProcessing = true
    if #available(iOS 14.0, *) {
        complete([.banner, .badge, .sound])
    } else {
        complete([.alert, .badge, .sound])
    }

    // или [] чтобы не показывать
}
```

Если `customPushProcessing = false`, SDK применяет стандартные параметры показа уведомления, а callback (если задан) вызывается только для дополнительной логики приложения. Если `customPushProcessing = true`, приложение должно самостоятельно вызвать `complete(...)`, иначе уведомление в foreground может быть не показано.

Получение события клика по уведомлению в приложении (опционально)

По умолчанию этот функционал **не требует дополнительной настройки**. Если обработка push-уведомлений делегирована SDK, `NotificationManager` автоматически:

- обработает нажатие на уведомление или action,
- отправит push-event типа `open` на сторону Altcraft,
- выполнит стандартную логику обработки действий уведомления.

Настройка, описанная ниже, требуется **только если**:

- приложению необходимо получить доступ к данным уведомления при клике,
- требуется собственная логика обработки переходов или действий,

- либо нужно полностью перехватить обработку клика на стороне приложения.

Чтобы передать обработку клика приложению, используйте `NotificationManager`:

```
let manager = AltcraftSDK.shared.notificationManager

// Управление обработкой кликов по уведомлениям.
// true  — обработка клика полностью передаётся приложению.
// false — SDK использует стандартную логику обработки клика.
manager.customClickProcessing = true

// Колбэк обработки клика по уведомлению.
// Вызывается при нажатии на уведомление или выполнении action.
// completion должен быть вызван обязательно при customClickProcessing = true.
manager.onNotificationClick = { response, complete in
    // Данные уведомления
    let userInfo = response.notification.request.content.userInfo

    // Пользовательская логика обработки клика
    print("push click")

    // Завершение обработки
    complete()
}
```

Если `customClickProcessing = false`, SDK самостоятельно обрабатывает клик по уведомлению, отправляет соответствующие события и завершает обработку. Callback `onNotificationClick` в этом случае (если задан) вызывается только для дополнительной логики приложения.

Если `customClickProcessing = true`, приложение **обязано вызвать** `complete()`. Без вызова `complete()` система iOS будет считать обработку клика незавершённой, что может привести к некорректному поведению приложения.

Мобильные события

```
// Функции объекта PublicMobileEventFunctions

AltcraftSDK.shared
└── let mobileEventFunctions: PublicMobileEventFunctions

    // Отправить мобильное событие (mobile event) на сервер
    └── func mobileEvent(
            sid: String,
            eventName: String,
            sendMessageId: String? = nil,
            payload: [String: Any?]? = nil,
            matching: [String: Any?]? = nil,
            matchingType: String? = nil,
            profileFields: [String: Any?]? = nil,
            subscription: (any Subscription)? = nil
            utm: UTM? = nil
        )
```

Кейс: Передача информации о рекламной кампании

Информация о рекламной кампании приложения, которая привела к установке, может быть передана в платформу как значение параметров `profileFields` / `customFields` функции `pushSubscribe`, а также параметров `utm` или `payload` функции `mobileEvent()`. Получив UTM-метку в приложении как строку, необходимо вызвать функцию `mobileEvent()`:

Передача через payload

```
    AltcraftSDK.shared.mobileEventFunctions.mobileEvent(
            sid: "your sid",
            eventName: "app_install",
            payload: ["utm": "your utm tag"]
        )
```

Передача в поля UTM

```
       AltcraftSDK.shared.mobileEventFunctions.mobileEvent(
            sid: "your sid",
            eventName: "app_install",
            utm: UTM (
                campaign : "your campaign utm",
                content : "your content utm",
                keyword : "your keyword utm",
                medium : "your medium utm",
                source : "your source utm",
                temp : "your temp utm"
            )
        )
```

Передача в пользовательское поле профиля

```
//передача utm-метки в поле профиля
//поля профиля должны быть предварительно добавлены в платформе
AltcraftSDK.shared.pushSubscriptionFunctions.pushSubscribe(
     profileFields: ["utm": "your utm tag"]
)
```

Переданные любым из этих способов данные можно использовать для создания сегмента в платформе.

Обратите внимание, что при настройке на iOS приложение должно использовать **AppsFlyer**, иначе UTM-метка не будет передана в приложение.

Для регистрации мобильного события используйте функцию `mobileEvent()` из `PublicMobileEventFunctions`. Она имеет следующие параметры:

---

sid: `String`

Обязательный: **Да**

Описание: Строковый идентификатор пикселя, к которому привязываются мобильные события.

---

eventName: `String`

Обязательный: **Да**

Описание: Имя мобильного события.

---

sendMessageId: `String?`

Обязательный: Нет

Описание: SMID-идентификатор отправленного сообщения (если событие связано с конкретной рассылкой).

---

payload: `[String: Any?]?`

Обязательный: Нет

Описание: Данные события — карта со строковыми ключами и значениями, для которых допускаются только **скалярные** типы данных:

- `String`
- `Bool`
- `Int`
- `Int64` / `UInt64` (или `NSNumber` эквиваленты)
- `Float`
- `Double`
- `nil` / `NSNull`

> **TIP: Обратите внимание**
>
> Несериализуемые объекты (например, `Date` без преобразования, кастомные классы) приведут к ошибке кодирования.

---

matching: `[String: Any?]?`

Обязательный: Нет
Описание: Карта, в которую можно передавать значения с типами и идентификаторами матчинга (например, `["email" : "user@example.com"]`).

---

matchingType: `String?`

Обязательный: Нет
Описание: Тип матчинга.

---

profileFields: `[String: Any?]?`

Обязательный: Нет
Описание: Поля профиля — карта со строковыми ключами и значениями (JSON-совместимые типы):

- **Скалярные значения**:

    - `String`
    - `Bool`
    - `Int`
    - `Int64` / `UInt64` (или эквиваленты `NSNumber`)
    - `Float`
    - `Double`
    - `nil` / `NSNull`
- **Объекты**: `[String: Any?]`
- **Списки**: `[Any?]`
- **Массивы карт**: `[[String: Any?]]`

> **TIP: Обратите внимание**
>
> Параметр `profileFields` используется только при работе с JWT-авторизацией.

---

utm: `UTM?`

Обязательный: Нет
Описание: UTM-метки. Добавляются с помощью struct UTM, где каждый вид UTM — свойство структуры.

```
public struct UTM{
    /// UTM Campaign
    public var campaign: String?
    /// UTM Content
    public var content: String?
    /// UTM Keyword/Term
    public var keyword: String?
    /// UTM Medium
    public var medium: String?
    /// UTM Source
    public var source: String?
    /// UTM Temp
}
```

---

subscription: `(any Subscription)?`

Обязательный: Нет
Описание: Добавляемая к профилю подписка одного из типов: `EmailSubscription`, `SmsSubscription`, `PushSubscription`, `CcDataSubscription`.

> **TIP: Обратите внимание**
>
> Используется только при работе с JWT-авторизацией.

Реализации интерфейса `Subscription`

Назначение — общий протокол для всех типов подписок.
Сериализация — через `Codable`, поле-дискриминатор — `channel`.

Общие поля (для всех реализаций):

Поле	Тип	Обязательный	Описание
`resource_id`	`Int`	Да	Идентификатор ресурса/источника подписки
`status`	`String?`	Нет	Статус подписки (например, активна/приостановлена)
`priority`	`Int?`	Нет	Приоритет доставки для данной подписки
`custom_fields`	`[String: JSONValue]?`	Нет	Пользовательские поля (ключ-значение) для расширенной сегментации
`cats`	`[String]?`	Нет	Категории подписки
`channel`	`String`	Да	Тип канала; фиксируется реализацией.

TIP: Обратите внимание

`JSONValue` — сериализуемый тип/перечень JSON-значений в вашем SDK. Все значения должны быть совместимы с JSON.

Варианты подписки:

EmailSubscription (`channel = "email"`)

Основные поля

Поле	Тип	Обязательный	Описание
`resourceId`	`Int`	Да	ID ресурса Altcraft
`email`	`String`	Да	Адрес электронной почты получателя

Дополнительные поля

Поле	Тип	Обязательный	Описание
`status`	`String?`	Нет	Статус подписки
`priority`	`Int?`	Нет	Приоритет подписки
`customFields`	`[String: JSONValue]?`	Нет	Стандартные и кастомные поля подписки

Поле	Тип	Обязательный	Описание
`cats`	`[String]?`	Нет	Категории подписки

SmsSubscription (`channel = "sms"`)

Основные поля

Поле	Тип	Обязательный	Описание
`resourceId`	`Int`	Да	ID ресурса Altcraft
`phone`	`String`	Да	Номер телефона в международном формате

Дополнительные поля

Поле	Тип	Обязательный	Описание
`status`	`String?`	Нет	Статус подписки
`priority`	`Int?`	Нет	Приоритет подписки
`customFields`	`[String: JSONValue]?`	Нет	Стандартные и кастомные поля подписки
`cats`	`[String]?`	Нет	Категории подписки

PushSubscription (`channel = "push"`)

Основные поля

Поле	Тип	Обязательный	Описание
`resourceId`	`Int`	Да	ID ресурса Altcraft
`provider`	`String`	Да	Провайдер (например, `"ios-apns"`)
`subscriptionId`	`String`	Да	Уникальный идентификатор подписки у провайдера

Дополнительные поля

Поле	Тип	Обязательный	Описание
`status`	`String?`	Нет	Статус подписки
`priority`	`Int?`	Нет	Приоритет подписки
`customFields`	`[String: JSONValue]?`	Нет	Стандартные и кастомные поля подписки
`cats`	`[String]?`	Нет	Категории подписки

CcDataSubscription (`channel ∈ {"telegram_bot","whatsapp","viber","notify"}`)

Основные поля

Поле	Тип	Обязательный	Описание
`resourceId`	`Int`	Да	ID ресурса Altcraft
`channel`	`String`	Да	Один из: `"telegram_bot"`, `"whatsapp"`, `"viber"`, `"notify"`
`ccData`	`[String: JSONValue]`	Да	Канал-специфичные данные (например, chat ID, номер, токены)

Дополнительные поля

Поле	Тип	Обязательный	Описание
`status`	`String?`	Нет	Статус подписки
`priority`	`Int?`	Нет	Приоритет подписки
`customFields`	`[String: JSONValue]?`	Нет	Стандартные и кастомные поля подписки
`cats`	`[String]?`	Нет	Категории подписки

Обновление полей профиля

INFO: Важно

Обновление полей профиля с помощью функции `updateProfileFields()` работает только при использовании JWT-авторизации API-запросов SDK.

```
AltcraftSDK.shared
└── let profileFunctions: PublicProfileFunctions
    // Обновляет поля профиля Altcraft
    └── func updateProfileFields(
        profileFields: [String: Any?]? = nil,
        skipTriggers: Bool? = nil
    ): Void
```

Для обновление полей профиля используйте функцию `updateProfileFields()`. Она имеет следующие параметры:

---

profileFields: `[String: Any?]?`

Обязательный: Нет
Описание: Поля профиля — карта содержащая значение для полей, которые необходимо изменить (JSON-совместимые типы):

- **Скалярные значения**:

    - `String`
    - `Bool`
    - `Int`
    - `Int64` / `UInt64` (или эквиваленты `NSNumber`)
    - `Float`
    - `Double`
    - `nil` / `NSNull`
- **Объекты**: `[String: Any?]`
- **Списки**: `[Any?]`
- **Массивы карт**: `[[String: Any?]]`

> **INFO: Обратите внимание**
>
> Для того чтобы обновление прошло успешно, поля должны быть предварительно добавлены в профиль подписчика.

---

skipTriggers: `Bool?`

По умолчанию: `nil`
Обязательный: Нет
Описание: При активации флага профиль, содержащий данную подписку, будет игнорироваться в триггерах рассылок и сценариев.

Получение событий SDK в приложении

```
// Функции объекта Events

AltcraftSDK
// Singleton, точка входа в SDK
└── static let shared: AltcraftSDK
    // Поток событий SDK (один активный подписчик)
    └── eventSDKFunctions: SDKEvents
        // Подписка на события SDK (заменяет существующего подписчика)
        ├── subscribe(
        │       callback: @escaping (Event) -> Void
        │   ): Void
        // Отписка от событий (callback остается назначенным, доставка прекращается)
        └── unsubscribe(): Void
```

В приложении может быть **только один** активный подписчик на события SDK.

Подписка на события

`public func subscribe(callback: @escaping (Event) -> Void): Void`

Функция используется для того, чтобы подписаться и получать события SDK в приложении. При возникновении события SDK она вызывает `callback` и передаёт в него экземпляр класса `Event` (или его наследника).

Пример использования:

```
AltcraftSDK.shared.eventSDKFunctions.subscribe { event in
  // событие
}
```

Все события, передаваемые SDK, являются экземплярами `Event` или его наследников:

- **Event** — общее событие (информация, успешные запросы);
- **Error** — событие об ошибке;
- **RetryError** — событие об ошибке при выполнения запроса для которого предусмотрен автоматический повтор выполнения на стороне SDK.

Каждое событие содержит поля:

- **function** — имя функции, вызывавшей событие;
- **eventCode** — внутренний код события SDK (список событий SDK представлен в пункте “События SDK”);
- **eventMessage** — сообщение события;
- **eventValue** — произвольные данные [String: Any?]? которые добавляются к некоторым событиям как полезная нагрузка;
- **date** — время события.

Классы событий SDK

```
open class Event: Hashable {
    public let id = UUID()
    public let function: String
    public let message: String?
    public let eventCode: Int?
    public let value: [String: Any?]?
    public let date: Date
}

open class ErrorEvent: Event {
    public override init(
        function: String,
        message: String? = nil,
        eventCode: Int? = nil,
        value: [String: Any?]? = nil,
        date: Date = Date()
    ) {
        super.init(
            function: function,
            message: message,
            eventCode: eventCode,
            value: value,
            date: date
        )
    }
}

public class RetryEvent: ErrorEvent {
    public override init(
        function: String,
        message: String? = nil,
        eventCode: Int? = 0,
        value: [String: Any?]? = nil,
        date: Date = Date()
    ) {
        super.init(
            function: function,
            message: message,
            eventCode: eventCode,
            value: value,
            date: date
        )
    }
}
```

Пример содержания события успешной подписки на push-уведомления

```
├ function: processResponse
├ eventCode: 230
├ message: "successful request: push/subscribe"
├ value
│  ├ http code: 200
│  └ response
│     ├ error: 0
│     ├ errorText: ""
│     └ profile
│        ├ id: // ID профиля
│        ├ status: subscribed
│        ├ isTest: false
│        └ subscription
│           ├ subscriptionId: // push-токен подписки
│           ├ hashId: // хеш ID
│           ├ provider: "ios-apns"
│           ├ status: subscribed
│           ├ fields
│           │  ├ _ad_id: // google service ad id
│           │  ├ _ad_track: false
│           │  ├ _app_id: "AltcraftMobile"
│           │  ├ _app_iid: "1.0.0"
│           │  ├ _app_ver: {"raw":"1.0.0","ver":[1,0]}
│           │  ├ _device_model: "iPhone14,7"
│           │  ├ _device_name: "iPhone"
│           │  ├ _device_type: "mob"
│           │  ├ _os: "Android"
│           │  ├ _os_language: "ru"
│           │  ├ _os_tz: "+0300"
│           │  └ _os_ver: {"raw":"18.6.2","ver":"[\"18.0\", \"6.0\", \"2.0\"]"}
│           └ cats
│              └ [ { name: "cats_1", title: "cats_1", steady: false, active: false } ]
└ date: Tue Aug 12 15:49:20 GMT+03:00 2025
```

Отписка от событий

`public func unsubscribe(): Void`

При помощи данной функции можно отписаться от событий SDK. Она отменяет передачу событий SDK, при этом `callback` остаётся назначенным, но доставка событий прекращается.

Пример использования:

```
AltcraftSDK.shared.eventSDKFunctions.unsubscribe()
```

Список всех событий SDK

Список событий

Код	Описание
200	SDK configuration is installed
201	push provider set
203	received Altcraft push notification
204	push is posted
205	SDK data has been cleared
206	SDK background task is registered
207	SDK background task completed
208	SDK background task is registered
209	SDK background task completed
230	subscribe request succeeded
231	token update request succeeded
232	unsuspend request succeeded
233	status request succeeded
234	push event delivered successfully
235	mobile event delivered successfully
401	the configuration is not set
402	userTag is null. It is impossible to identify the user
403	error in CoreData. Re-initialize the SDK
404	failed to load model from framework
405	App Group name was not set
406	forcing a push token update is not possible: the operation is not supported for APNS
407	the CoreData entity name is not valid
408	entity not found by ID
409	SDK initialization timeout has expire
422	unsuspend request data is nil
423	profile request data is nil
430	subscribe request failed
431	token update request failed
432	unsuspend request failed

Код	Описание
433	status request failed
434	push event delivery failed
435	mobile event delivery failed
450	invalid userInfo format
451	uid in the push data is null; it is impossible to send a push event to the server
452	could not download image data from URL
453	‘media’ key is missing or URL string is invalid
454	‘buttons’ key is missing or is invalid
455	invalid button identifier (see event value)
456	out of range for identifier (see event value)
457	unknown button identifier (see event value)
471	invalid apiUrl value — empty or null
472	invalid resource token value — resource token is empty
473	invalid provider. Available — ios-apns, ios-firebase, ios-huawei
474	non-JSON object has been provided
475	invalid values: not all values are primitives
476	unsupported subscription type. Available: EmailSubscription, SmsSubscription, PushSubscription, CcDataSubscription
501	config data is nil
502	current token is nil
503	userTag is null. It is impossible to identify the user
504	notification permission denied
505	SDK background task expired
520	subscribe request data is nil
521	update request data is nil
524	push event request data is nil
525	mobile event request data is nil
529	common data is nil
530	subscribe request failed (retryable)
531	token update request failed (retryable)
534	push event delivery failed (retryable)

Код	Описание
535	mobile event delivery failed (retryable)
540	JWT token is nil
541	JWT parsing error
542	invalid matching
543	auth data is nil
561	failed to create request
562	failed processing the response
571	invalid request URL
572	invalid response format
573	invalid push subscribe request data
574	invalid push event request data

Дополнительные функции SDK

Функциональное обновление полей профиля

`public func actionField(key: String) -> ActionFieldBuilder`

Функция, облегчающая процесс функционального обновления полей профиля. Поддерживает следующие команды:

```
    .set(value)
    .unset(value)
    .incr(value)
    .add(value)
    .delete(value)
    .upsert(value)
```

Пример использования:

```
// "_fname" — изменяемое поле
// .set("Andrey") — команда, которая установит для поля значение "Andrey"
AltcraftSDK.shared.pushSubscriptionFunctions.pushSubscribe(
    profileFields: AltcraftSDK.shared.pushSubscriptionFunctions.actionField(key:
"_fname").set(value: "Andrey")
)
```

Очистка данных SDK

`public func clear(completion: (() -> Void)? = nil): Void`

Функция позволяет выполнить очистку данных SDK и отменить работу всех фоновых задач, которые ожидают выполнения. Она удаляет записи `CoreData` и очищает `UserDefaults`.

Пример использования:

```
// Полная очистка данных SDK (кэш/база данных/настройки), затем вызов completion
AltcraftSDK.shared. clear(completion: (() -> Void)? = nil)
```

Функция принимает необязательный параметр `completion`, который вызывается после завершения очистки и отмены задач.

Ручная регистрация push-событий

> **TIP: Обратите внимание**
>
> Эти функции применяются только в том случае, если вы реализуете собственную логику обработки уведомлений и не передаёте их в Altcraft SDK. По умолчанию Altcraft SDK обрабатывает уведомления автоматически, поэтому использование данных методов требуется лишь при необходимости кастомной обработки на стороне клиента.

```
// Функции объекта PublicPushEventFunctions

AltcraftSDK
//   Singleton, точка входа в SDK
└── public static let shared: AltcraftSDK
    // Ручная отправка push-событий (delivery / open)
    └── public let pushEventFunctions: PublicPushEventFunctions
        // Запись доставки push-уведомления Altcraft (запускает событие доставки)
        ├── public func deliveryEvent(from request: UNNotificationRequest): Void
        // Запись открытия push-уведомления Altcraft (запускает событие открытия)
        └── public func openEvent(from request: UNNotificationRequest): Void
```

Функции ручной регистрации событий:

- `func deliveryEvent()` — регистрация доставки уведомления;
- `func openEvent()` — регистрация открытия уведомления.

`public func deliveryEvent(from request: UNNotificationRequest): Void`

Функция ручной регистрации события доставки уведомления в платформе Altcraft. Для регистрации события на сервере передайте `UNNotificationRequest` в параметр `request`.

Пример использования:

```
AltcraftSDK.shared.pushEventFunctions.deliveryEvent(from: UNNotificationRequest)
```

`public func openEvent(from request: UNNotificationRequest): Void`

Функция ручной регистрации события открытия уведомления в платформе Altcraft. Для регистрации события открытия на сервере передайте `UNNotificationRequest` в параметр `request`.

```
AltcraftSDK.shared.pushEventFunctions.openEvent(from: UNNotificationRequest)
```

Публичный API SDK

class AltcraftSDK

```
AltcraftSDK
// Синглтон точка входа в SDK
├─ public static let shared: AltcraftSDK
// Инициализация SDK конфигурацией (completion вызывается на main)
├─ public func initialization(configuration: AltcraftConfiguration?, completion: ((Bool) ->
Void)? = nil): Void
// Установка App Group и инициализация Core Data под shared-контейнер
├─ public func setAppGroup(groupName: String): Void
// Регистрация JWT-провайдера для получения токенов
├─ public func setJWTProvider(provider: JWTInterface): Void
// Полная очистка данных SDK (кэш/БД/настройки), затем вызов completion
├─ public func clear(completion: (() -> Void)? = nil): Void
|
// Поток SDK-событий (один активный подписчик)
├─ public let eventSDKFunctions: SDKEvents
│  // Подписаться на события SDK (заменяет существующего подписчика)
│  ├─ func subscribe(callback: @escaping (Event) -> Void): Void
│  // Отписаться от событий (колбэк остаётся назначенным, доставка останавливается)
│  ├─ func unsubscribe(): Void
│  // Модель события (базовый тип для всех событий/ошибок)
│  ├─ Event
│  │   // Базовая модель события
│  │   id: UUID
│  │   function: String
│  │   message: String?
│  │   eventCode: Int?
│  │   value: [String: Any?]?
│  │   date: Date
│  │   // Конструктор (нормализация function, фильтрация nil в value)
│  │   init(function: String, message: String? = nil, eventCode: Int? = nil, value:
[String: Any?]? = nil, date: Date = Date())
│  │   // Сравнение по уникальному id
│  │   static func ==(lhs: Event, rhs: Event) -> Bool
│  │   // Хеширование по id
│  │   func hash(into hasher: inout Hasher): Void
│  |
│  │   // Ошибки без повторов (4xx-класс)
│  │   ErrorEvent : Event
│  │   init(function: String, message: String? = nil, eventCode: Int? = nil, value:
[String: Any?]? = nil, date: Date = Date())
│  |
│  │   // Повторяемые ошибки (обычно 5xx)
│  │   RetryEvent : ErrorEvent
│  │   init(function: String, message: String? = nil, eventCode: Int? = 0, value: [String:
Any?]? = nil, date: Date = Date())
|
// Управление push-токенами и провайдерами (FCM/HMS/APNs)
├─  public let pushTokenFunction: PublicPushTokenFunctions
│  // Установить провайдера Firebase Cloud Messaging (nil — снять)
```

```
│  // Установить провайдера Firebase Cloud Messaging (nil — снять)
│  ├─ public func setFCMTokenProvider(_ provider: FCMInterface?): Void
│  // Установить провайдера Huawei Mobile Services (nil — снять)
│  ├─ public func setHMSTokenProvider(_ provider: HMSInterface?): Void
│  // Установить провайдера Apple Push Notification service (nil — снять)
│  ├─ public func setAPNSTokenProvider(_ provider: APNSInterface?): Void
│  // Асинхронно получить текущий push-токен (completion опционален)
│  ├─ public func getPushToken(completion: ((TokenData?) -> Void)? = nil): Void
│  // Сохранить токен вручную (String для FCM/HMS, Data для APNs); provider: "ios-firebase"
| "ios-huawei" | "ios-apns"
│  ├─ public func setPushToken(provider: String, pushToken: Any?): Void
│  // Применить новый приоритет провайдеров и инициировать обновление токена
│  ├─ public func changePushProviderPriorityList(_ list: [String]): Void
│  // Удалить токен выбранного провайдера и вызвать completion
│  ├─ public func deleteDeviceToken(provider: String, completion: @escaping () -> Void):
Void
│  // Форсировать обновление токена (определить провайдера –> удалить (кроме APNs) –>
запросить новый)
│  └─ public func forcedTokenUpdate(completion: (() -> Void)? = nil): Void
│
// Управление подпиской на push (изменение статуса, запросы статуса)
├─  public let pushSubscriptionFunctions: PublicPushSubscriptionFunctions
│  // Оформить подписку (status = SUBSCRIBED)
│  ├─ public func pushSubscribe(
│  │     sync: Bool = true,
│  │     profileFields: [String: Any?]? = nil,
│  │     customFields: [String: Any?]? = nil,
│  │     cats: [CategoryData]? = nil,
│  │     replace: Bool? = nil,
│  │     skipTriggers: Bool? = nil
│  │   ): Void
│  // Отписаться (status = UNSUBSCRIBED)
│  ├─ public func pushUnSubscribe(
│  │     sync: Bool = true,
│  │     profileFields: [String: Any?]? = nil,
│  │     customFields: [String: Any?]? = nil,
│  │     cats: [CategoryData]? = nil,
│  │     replace: Bool? = nil,
│  │     skipTriggers: Bool? = nil
│  │   ): Void
│  // Приостановить (status = SUSPENDED)
│  ├─ public func pushSuspend(
│  │     sync: Bool = true,
│  │     profileFields: [String: Any?]? = nil,
│  │     customFields: [String: Any?]? = nil,
│  │     cats: [CategoryData]? = nil,
│  │     replace: Bool? = nil,
│  │     skipTriggers: Bool? = nil
│  │   ): Void
│  // UnSuspend подписки (один запрос, без ретраев/персиста)
```

```
│  ├─ public func unSuspendPushSubscription(
│  │     completion: @escaping (ResponseWithHttp?) -> Void
│  │   ): Void
│  // Статус последней подписки профиля
│  ├─ public func getStatusOfLatestSubscription(
│  │     completion: @escaping (ResponseWithHttp?) -> Void
│  │   ): Void
│  // Статус подписки, соответствующий текущим токену/провайдеру
│  ├─ public func getStatusForCurrentSubscription(
│  │     completion: @escaping (ResponseWithHttp?) -> Void
│  │   ): Void
│  // Статус последней подписки по провайдеру (если nil — используется текущий)
│  └─ public func getStatusOfLatestSubscriptionForProvider(
│  │       provider: String? = nil,
│  │       completion: @escaping (ResponseWithHttp?) -> Void
│  │     ): Void
│   // добавить функциональное поле профиля(set/incr/...)
│  └─ public func actionField(key: String) -> ActionFieldBuilder
│
│ // Регистрация мобильных событий
├─ public let mobileEventFunctions: PublicMobileEventFunctions
│  // Регистрирует мобильное событие Altcraft
│  └─ public func mobileEvent(
│        sid: String,
│        altcraftClientID: String = "",
│        eventName: String,
│        sendMessageId: String? = nil,
│        payload: [String: Any?]? = nil,
│        matching: [String: Any?]? = nil,
│        matchingType: String? = nil,
│        profileFields: [String: Any?]? = nil,
│        subscription: (any Subscription)? = nil,
│        utm: UTM? = nil
│     ): Void
│
│ // Обновление полей профиля
├─ public let profileFunctions: PublicProfileFunctions
│  // Обновляет поля профиля Altcraft
│  └─ public func updateProfileFields(
│        profileFields: [String: Any?]? = nil,
│        skipTriggers: Bool? = nil
│     ): Void
│
// ручная отправка пуш-событий (delivery/open)
├─  public let  pushEventFunctions: PublicPushEventFunctions
│  // Зафиксировать доставку Altcraft-push (вызывает delivery-ивент)
│  ├─ public func deliveryEvent(from request: UNNotificationRequest): Void
│  // Зафиксировать открытие Altcraft-push (вызывает open-ивент)
│  └─ public func openEvent(from request: UNNotificationRequest): Void
```

```
|
// Регистрация периодических фоновых задач (BGAppRefreshTask)
├─ public let backgroundTasks: BackgroundTasks
│  // Зарегистрировать задачу и запланировать периодический запуск (~каждые 3 часа)
│  └─ public func registerBackgroundTask(): Void
|
// Управление нотификациями в приложении (делегат центра уведомлений)
├─  public let notificationManager: NotificationManager
   // Синглтон NotificationManager (точка доступа к менеджеру уведомлений)
   ├─ public static let shared: NotificationManager
   // ObjC-friendly аксессор (эквивалент shared)
   ├─ public class func sharedInstance() -> NotificationManager
   //
   // NotificationCenter events (SDK posts these)
   ├─ public extension Notification.Name
   │  // Fired when a notification is about to be presented in foreground.
   │  // userInfo: ["notification": UNNotification]
   │  ├─ static let altcraftPushWillPresent: Notification.Name
   │  // Fired when user taps on a delivered notification or performs an action.
   │  // userInfo: ["response": UNNotificationResponse]
   │  └─ static let altcraftPushDidReceive: Notification.Name
   //
   // Режимы делегирования обработки приложению
   ├─ public var customPushProcessing: Bool
   ├─ public var customClickProcessing: Bool
   //
   // Колбэки для кастомной обработки
   │  // Called when a foreground push arrives.
   │  // App must call completion when customPushProcessing = true.
   ├─ public var onForegroundNotification: ((UNNotification, @escaping
(UNNotificationPresentationOptions) -> Void) -> Void)?
   │  // Called when the user taps a notification (or performs an action).
   │  // App must call completion when customClickProcessing = true.
   ├─ public var onNotificationClick: ((UNNotificationResponse, @escaping () -> Void) ->
Void)?
   //
   // Регистрация на push (delegate + requestAuthorization +
registerForRemoteNotifications)
   ├─ func registerForPushNotifications(for application: UIApplication, completion: ((_
granted: Bool, _ error: Error?) -> Void)? = nil): Void
   //
   // UNUserNotificationCenterDelegate
   │  // Постит altcraftPushWillPresent; при customPushProcessing отдаёт решение
приложению,
   │  // иначе показывает banner/alert (+ badge/sound)
   ├─ public func userNotificationCenter(
   │     _ center: UNUserNotificationCenter,
   │     willPresent notification: UNNotification,
   │     withCompletionHandler completionHandler: @escaping
(UNNotificationPresentationOptions) -> Void
```

```
(UNNotificationPresentationOptions) -> Void
   │   ): Void
   │  // Постит altcraftPushDidReceive; при customClickProcessing отдаёт обработку
приложению,
   │  // иначе парсит userInfo, шлёт open event и выполняет action; затем вызывает
completionHandler
   └─ public func userNotificationCenter(
        _ center: UNUserNotificationCenter,
        didReceive response: UNNotificationResponse,
        withCompletionHandler completionHandler: @escaping () -> Void
      ): Void
```

class AltcraftNSE

```
// Используется внутри Notification Service Extension (NSE) — отвечает за обработку push-уведомлений Altcraft,
// определение уведомлений Altcraft и регистрацию мобильных событий без зависимости от UIApplication
AltcraftNSE
// Общий (singleton) экземпляр для централизованной логики NSE
├─ public static let shared: AltcraftNSE
// Предоставляет доступ к API регистрации мобильных событий
├─ public let mobileEventFunctions: PublicMobileEventFunctions
// Регистрирует мобильное событие Altcraft
├─ public func mobileEvent(
│     sid: String,
│     altcraftClientID: String = "",
│     eventName: String,
│     sendMessageId: String? = nil,
│     payload: [String: Any?]? = nil,
│     matching: [String: Any?]? = nil,
│     matchingType: String? = nil,
│     profileFields: [String: Any?]? = nil,
│     subscription: (any Subscription)? = nil,
│     utm: UTM? = nil
│  ): Void
// Обновляет поля профиля Altcraft
├─ public func updateProfileFields(
│     profileFields: [String: Any?]? = nil,
│     skipTriggers: Bool? = nil
│  ): Void
// Проверяет, относится ли уведомление к Altcraft
├─ public func isAltcraftPush(_ request: UNNotificationRequest) -> Bool
// Обрабатывает входящее уведомление Altcraft и передаёт изменённое содержимое в contentHandler
├─ public func handleNotificationRequest(
│     request: UNNotificationRequest,
│     contentHandler: @escaping (UNNotificationContent) -> Void
│  ): Void
// Вызывается системой при истечении времени выполнения NSE
├─ public func serviceExtensionTimeWillExpire(): Void
// Устанавливает App Group и инициализирует общий контейнер данных
├─ public func setAppGroup(groupName: String): Void
// Регистрирует JWT-провайдер для безопасной работы с токенами
└─ public func setJWTProvider(provider: JWTInterface): Void
```

class AltcraftConfiguration

```
AltcraftConfiguration
// Класс конфигурации Altcraft SDK.
// Хранит базовый API URL и опциональные параметры (rToken, AppInfo, providerPriorityList,
enableLogging).
// Создаётся только через Builder. ObjC-совместим: AltcraftConfiguration /
AltcraftConfiguration_Builder.
│
├─ @objc(AltcraftConfiguration_Builder) public final class Builder
│  // Builder для формирования конфигурации с валидацией.
│  // Позволяет задать обязательные и опциональные поля и собрать объект конфигурации.
│  // Конструктор билдера
│  ├─ public override init()
│  // Задать API URL (обязательный параметр)
│  ├─ @discardableResult public func setApiUrl(_ url: String) -> Builder
│  // Задать ресурсный токен (опционально)
│  ├─ @discardableResult public func setRToken(_ rToken: String?) -> Builder
│  // Задать метаданные приложения (Swift-модель AppInfo) (опционально)
│  ├─ @discardableResult public func setAppInfo(_ info: AppInfo?) -> Builder
│  // Задать метаданные приложения (ObjC DTO AppInfoObjC) (опционально, Swift недоступно)
│  ├─ @available(swift, obsoleted: 1) @objc(setAppInfo:) @discardableResult public func
setAppInfo(_ dto: AppInfoObjC?) -> Builder
│  // Задать приоритет провайдеров пуш-уведомлений (опционально)
│  ├─ @discardableResult public func setProviderPriorityList(_ list: [String]?) -> Builder
│  // Управление внутренним логированием SDK (опционально)
│  // true — включить, false — выключить, nil — оставить поведение по умолчанию
│  ├─ @discardableResult public func setEnableLogging(_ enabled: Bool?) -> Builder
│  // Построить валидную конфигурацию (nil, если валидация не пройдена)
│  // Валидация: apiUrl не пустой; providerPriorityList валиден; appInfo резолвится (Swift
AppInfo приоритетнее, иначе AppInfoObjC -> AppInfo)
│  └─ public func build() -> AltcraftConfiguration?
// Доступ к параметрам конфигурации
├─ public func getApiUrl() -> String
├─ public func getRToken() -> String?
├─ public func getAppInfo() -> AppInfo?
├─ public func getEnableLogging() -> Bool?
└─ public func getProviderPriorityList() -> [String]?
```

Structures

```
/// Хранит push-токен и имя провайдера.
public struct TokenData: Codable {
    public let provider: String
    public let token: String
}

/// Метаданные приложения для Firebase Analytics (если используется).
public struct AppInfo: Codable {
    public var appID: String
    public var appIID: String
    public var appVer: String
}

/// Обёртка ответа API с HTTP-кодом.
public struct ResponseWithHttp {
    public let httpCode: Int?
    public let response: Response?
}

/// Ответ синхронного запроса подписки.
public struct Response: Codable {
    public let error: Int?
    public let errorText: String?
    public let profile: ProfileData?
}

/// Данные профиля пользователя.
public struct ProfileData: Codable {
    public let id: String?
    public let status: String?
    public let isTest: Bool?
    public let subscription: SubscriptionData?
}

/// Текущая подписка профиля.
public struct SubscriptionData: Codable {
    public let subscriptionId: String?
    public let hashId: String?
    public let provider: String?
    public let status: String?
    public let fields: [String: JSONValue]?
    public let cats: [CategoryData]?
}

/// Категория подписки (имя/заголовок/флаги).
public struct CategoryData: Codable {
    public var name: String?
    public var title: String?
    public var steady: Bool?
```

```
    public var steady: Bool?
    public var active: Bool?
}

/// Подписка канала Email (мобильные события).
public struct EmailSubscription: Subscription, Codable {
    public let resourceId: Int
    public let email: String
    public let status: String?
    public let priority: Int?
    public let customFields: [String: JSONValue]?
    public let cats: [String]?
    public let channel: String = "email"
}

/// Подписка канала SMS (мобильные события).
public struct SmsSubscription: Subscription, Codable {
    public let resourceId: Int
    public let phone: String
    public let status: String?
    public let priority: Int?
    public let customFields: [String: JSONValue]?
    public let cats: [String]?
    public let channel: String = "sms"
}

/// Подписка push-канала (мобильные события).
public struct PushSubscription: Subscription, Codable {
    public let resourceId: Int
    public let provider: String
    public let subscriptionId: String
    public let status: String?
    public let priority: Int?
    public let customFields: [String: JSONValue]?
    public let cats: [String]?
    public let channel: String = "push"
}

/// Подписка с cc_data (Telegram, WhatsApp, Viber, Notify).
public struct CcDataSubscription: Subscription, Codable {
    public let resourceId: Int
    public let channel: String
    public let ccData: [String: JSONValue]
    public let status: String?
    public let priority: Int?
    public let customFields: [String: JSONValue]?
    public let cats: [String]?
}

/// UTM-параметры для мобильных событий.
```

```
public struct UTM: Codable {
    public let campaign: String?
    public let content: String?
    public let keyword: String?
    public let medium: String?
    public let source: String?
    public let temp: String?
}
```

Protocols

JWTInterface:

```
JWTInterface
// Доступ к текущему JWT
└─ func getToken() -> String?
   // Возвращает текущий JWT или nil
```

APNSInterface:

```
APNSInterface
// Контракт получения APNs-токена
└─ func getToken(completion: @escaping (String?) -> Void)
   // Возвращает APNs-токен через completion или nil
```

FCMInterface:

```
FCMInterface
// Контракт операций с FCM-токеном (получение/отзыв)
├─ func getToken(completion: @escaping (String?) -> Void)
│  // Возвращает FCM-токен через completion или nil
└─ func deleteToken(completion: @escaping (Bool) -> Void)
   // Удаляет FCM-токен; completion(true|false)
```

HMSInterface:

```
HMSInterface
// Контракт операций с HMS-токеном
├─ func getToken(completion: @escaping (String?) -> Void)
│  // Возвращает HMS-токен через completion или nil
└─ func deleteToken(completion: @escaping (Bool) -> Void)
   // Удаляет HMS-токен; completion(true|false)
```

Subscription:

```
Subscription (protocol)
// Базовый протокол для всех типов подписок передаваемых с мобильным событием
```

Apple Push Notification Service

Вы можете использовать для настройки APNS **авторизационный ключ** или **сертификат**. В зависимости от этого, способ настройки будет отличаться.

Опция 1: использование Auth Key

Шаг 1. Настройка APNS

Зайдите на сайт Apple Developer в раздел **Account**. Перейдите в **Certificates, Identifiers & Profiles — Keys**. Создайте новый **Key**:

Выберите **Apple Push Notification service**:

Настройте конфигурацию APNs. Для этого нажмите **Configure**. Выберите Environment в зависимости от окружения:

Зарегистрируйте ключ:

Apple Developer предоставляет возможность **однократно** сохранить созданный ключ. **Обязательно** сохраните его в надежном месте! После загрузки его невозможно будет получить повторно:

Шаг 2. Настройка ресурса Altcraft

Настройте ресурс в платформе. Добавьте канал Push, активируйте переключатель Apple Push Notification service:

Активируйте переключатель **Использовать APNs токен**. Укажите следующие данные в полях ниже:

- **apns-topic** — Bundle ID приложения
- **kid** — ID ключа, который вы создали на предыдущем шаге. Он хранится в разделе **Certificates, Identifiers & Profiles — Keys** сайта Apple Developer
- **iss** — TeamID с сайта Apple Developer
- **Ключ подписи** — созданный вами Auth Key. Чтобы добавить его, откройте загруженный p8-файл с помощью текстового редактора и скопируйте его содержимое

Сохраните ресурс. Настройка провайдера завершена. Далее следуйте инструкции по инициализации SDK.

Опция 2: использование сертификата

Шаг 1: Настройка APNS

Запросите сертификат в бюро сертификации приложения Связка ключей:

Заполните необходимые данные, обязательно сохраните файл на диске:

Сохраненный файл в формате **.certSigningRequest** понадобится на следующем этапе настройки.

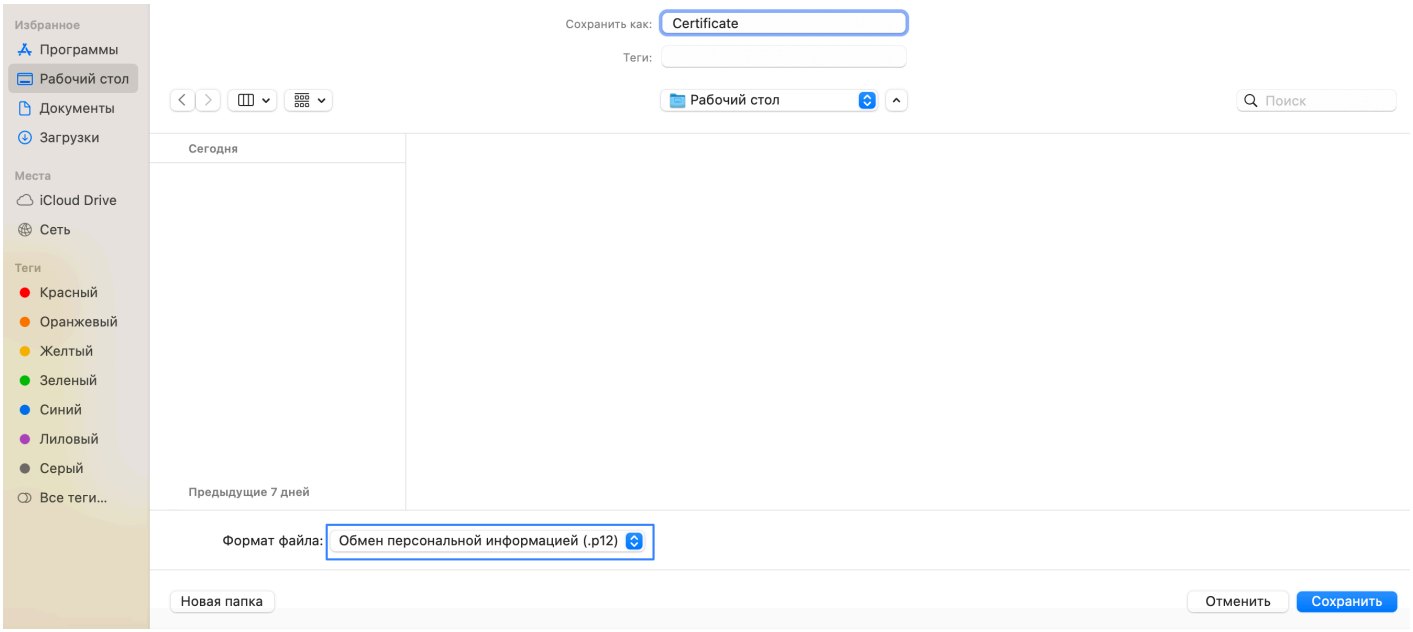
Зайдите на сайт Apple Developer в раздел **Account**. Перейдите в **Certificates, Identifiers & Profiles — Certificates**. Создайте новый **сертификат**:

В Services укажите **Apple Push Notification service SSL (Sandbox)** или **(Sandbox and Production)** в зависимости от окружения. Затем выберите ваше приложение из списка:

Загрузите заранее созданный **Certificate Signing Request** (файл в формате .certSigningRequest):

Скачайте созданный сертификат. Откройте скачанный файл в приложении Связка ключей. Дважды нажмите на сертификат в приложении и задайте уровень доверия **Всегда доверять**:

Нажмите правой кнопкой мыши на сертфикат и экспортируйте его как файл **формата .p12**. При необходимости установите для сертификата пароль:



Используя **терминал** перейдите в папку, в которую вы экспортировали сертификат. Создайте **pem-ключ** с помощью команды `openssl pkcs12 -legacy -in filename.p12 -out filename.key.pem -nocerts -nodes`, где `filename.p12` — имя экспортированного файла, а `filename.key.pem` — название создаваемого pem-файла с ключом.

Не закрывая терминал и не переходя в другие папки, создайте **pem-сертификат** с помощью команды `openssl pkcs12 -legacy -in filename.p12 -out filename.crt.pem -clcerts -nokeys` по аналогии с командой выше (`filename.p12` — исходный файл, `filename.crt.pem` — конечный файл).

Откройте оба файла в терминале при помощи **nano**. Их содержимое необходимо будет вставить в поля платформы на следующем этапе настройки:

Шаг 2. Настройка ресурса Altcraft

Настройте ресурс в платформе. Добавьте канал Push, активируйте переключатель Apple Push Notification service:

Активируйте переключатель **Использовать APNs сертификат**. Укажите следующие данные в полях ниже:

- **apns-topic** — Bundle ID приложения
- **Сертификат** — содержимое файла сертификата
- **Приватный ключ** — значение ключа сертификата

Сохраните ресурс. Настройка провайдера завершена. Далее следуйте инструкции по инициализации SDK.

Firebase Cloud Messaging

Шаг 1. Настройки проекта Firebase Cloud Messaging

- Выполните создание проекта Firebase;
- Зарегистрируйте новое приложение в проекте:
- Выполните загрузку `GoogleSrvice-Info.plist`;
- Перейдите в настройки проекта -> Cloud Messaging -> Apple app configuration;
- Добавьте необходимые данные APNs Authentication Key (необходимо предварительно создать ключ на Apple Developer): Укажите следующие данные:
  - загрузите загруженный файл-.p8
  - укажите `key ID`
  - укажите `team ID`

Шаг 2. Настройка ресурса Altcraft

- укажите `messagingSenderId`;
- укажите `projectId`;
- загрузите Firebase-файл с приватным ключом (.json). Файл создается в разделе Project setting -> Service account.

Шаг 3. Интеграция Firebase Cloud Messaging в проект приложения

- Переместите загруженные файл `GoogleSrvice-Info.plist` в папку приложения (app);
- Добавьте пакет Firebase с помощью Swift Package Manager — репозиторий Firebase. Версия пакета Firebase должна быть совместима с версией XCODE.
- **General:**
  - Проверьте, что библиотека **FirebaseMessaging** добавлена в **Frameworks, Libraries, and Embedded Content** таргета приложения.
- **Signing & Capabilities:**
  - **PushNotifications;**

Опция 1: FirebaseAppDelegateProxyEnabled — YES (По умолчанию для Firebase Messaging)

1. Добавьте протокол `MessagingDelegate` к `AppDelegate`:

```
class AppDelegate: UIResponder, UIApplicationDelegate, MessagingDelegate {}
```

Это позволит получать callback с актуальным FCM-токеном через метод `messaging(_:didReceiveRegistrationToken:)`.

2. Инициализируйте Firebase в `didFinishLaunchingWithOptions`:

```
FirebaseApp.configure()
Messaging.messaging().delegate = self
application.registerForRemoteNotifications()
```

3. Реализуйте метод делегата для получения токена:

```
func messaging(_ messaging: Messaging, didReceiveRegistrationToken fcmToken: String?)
{
    guard let token = fcmToken else { return }
    print(" FCM Token: \(token)")
}
```

Полный пример инициализации Firebase и получения FCM-токена

```
@main
struct IOSQuickStartApp: App {
    @UIApplicationDelegateAdaptor(AppDelegate.self) var delegate

    var body: some Scene {
        WindowGroup {
            ContentView()
        }
    }
}

class AppDelegate: UIResponder, UIApplicationDelegate, MessagingDelegate,
UNUserNotificationCenterDelegate {
    func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        ///Инициализация Firebase
        FirebaseApp.configure()
        ///Установка делегата
        Messaging.messaging().delegate = self
        application.registerForRemoteNotifications()

        return true
    }

    func messaging(_ messaging: Messaging, didReceiveRegistrationToken fcmToken: String?) {
        if let token = fcmToken {
           //FCM-токен
        }
    }
}
```

Опция 2: FirebaseAppDelegateProxyEnabled - NO

1. В настройках таргета приложения:
    - **Info:**
        - Добавьте ключ `FirebaseAppDelegateProxyEnabled (Boolean)` с значением `NO` в Info приложения.
2. Протокол `MessagingDelegate` в таком случае не требуется. При отключённом `FirebaseAppDelegateProxyEnabled` Firebase **не выполняет** автоматический swizzling `AppDelegate`, поэтому SDK сам управляет установкой и обработкой APNs- и FCM-токенов. Делегат `MessagingDelegate` в этом случае **не нужен**.
3. Инициализация Firebase в `didFinishLaunchingWithOptions`:

```
    func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        FirebaseApp.configure()
        return true
    }
```

4. Получение и сохранение APNs-токена в `didRegisterForRemoteNotificationsWithDeviceToken`:

```
class AppDelegate: UIResponder, UIApplicationDelegate, UNUserNotificationCenterDelegate {
    func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        FirebaseApp.configure()
        return true
    }

    func application(
        _ application: UIApplication,
        didRegisterForRemoteNotificationsWithDeviceToken apnsToken: Data
    ) {
        // Сохраните токен для последующего использования при запросе FCM-токена
        setAPNsTokenInUserDefault(apnsToken)
    }
}
```

Шаг 4. Ручная установка APNs-токена в Firebase Messaging перед запросом FCM-токена

Пример внутри провайдера SDK

```
class FCMProvider: FCMInterface {

    /// Извлекает текущий токен FCM
    func getToken(completion: @escaping (String?) -> Void) {

        /// APNs-токен, извлечённый из UserDefaults
        let apnsToken = getAPNsTokenDataFromUserDefaults()

        /// Устанавливаем APNs-токен вручную перед запросом FCM-токена
        Messaging.messaging().apnsToken = apnsToken

        /// Запрашиваем FCM-токен
        Messaging.messaging().token { token, error in
            if error != nil {
                completion(nil)
            } else {
                completion(token)
            }
        }
    }

    /// Удаляет текущий токен FCM
    func deleteToken(completion: @escaping (Bool) -> Void) {
        Messaging.messaging().deleteToken { error in
            if error != nil {
                completion(false)
            } else {
                completion(true)
            }
        }
    }
}
```

Huawei Mobile Services

Шаг 1. Настройки проекта Huawei Mobile Services

Создайте проект AppGalleryConnect. Для этого перейдите в раздел **Разработка и сервисы** -> **Добавить проект**:

Добавьте новое приложение в проект:

Укажите данные о вашем приложении:

Выполните загрузку файла `agconnect-services.plist`:

В левой боковой панели выберите **Push Kit** в разделе **Рост**:

Нажмите **Включить** — **Выберите место хранения данных**:

Далее перейдите в раздел **PushKit — Настройки — Приложение — Конфигурация IOS — Включить**. В открывшейся конфигурации IOS укажите:

```
 - **Ключ аутентификации APNs** – данные из файла закрытого .p8-ключа открытого в текстовом редакторе;
 - **ID ключа** – `keyID` файла закрытого .p8-ключа (обычно совпадает с названием самого файла  – значение после `AuthKey_`);
 - **ID разработчика** – `teamID` из Apple Developer.
```

Шаг 2. Настройка ресурса Altcraft

В платформе Altcraft создайте ресурс или добавьте к существующему новый канал коммуникации — Push. Затем в настройках ресурса выберите платформу **iOS** и активируйте переключатель Huawei Mobile Services:

Чуть ниже перейдите в настройки Huawei и укажите **ID приложения**, **ID клиента OAuth 2.0** и **секрет клиента OAuth 2.0**:

Обратите внимание, что ID приложения — **ID клиента** из раздела **Данные приложения**.

> **TIP**
>
> Все эти данные можно найти в разделе **Данные приложения** в Huawei Mobile Services:

После внесения изменений не забудьте **сохранить** ресурс.

Шаг 3. Интеграция Huawei Push Kit в проект приложения

Переместите загруженный файл `agconnect-services.plist` в корневой каталог модуля приложения (app). Затем подключите pod-файл `HmsPushSDK`.

Настройте таргет приложения:

- **General — Frameworks, Libraries, and Embedded Content**

Проверьте, что необходимые фреймворки добавлены и установлены как **Do Not Embed**:

- `AGConnectCore.xcframework`;
- `AGConnectCredential.xcframework`;
- `HiAnalytics.xcframework`;
- `HMFoundation.xcframework`.
- **Signing & Capabilities**

Добавьте **Push Notifications**.

Шаг 4. Получение HMS-токена

Получите HMS-токен с помощью функции `HmsInstanceId.getInstance().getToken(apnsToken)`, передав APNs-токен в качестве параметра:

Пример получения и удаления токена HMS в реализации интерфейса SDK

```
// обязательно импортируйте HmsPushSdk в файлы в которых используете функции HMS

class HMSProvider: HMSInterface {

    /// Извлекает текущий токен HMS с помощью токена APNs
    func getToken(completion: @escaping (String?) -> Void) {
        guard let apnsToken = getAPNsTokenFromUserDefault() else {
            completion(nil)
            return
        }

        let token = HmsInstanceId.getInstance().getToken(apnsToken)
        completion(token)
    }


    func deleteToken(completion: @escaping (Bool) -> Void) {
        HmsInstanceId.getInstance().deleteToken()
        completion(true)
    }
}
```

Быстрый старт

В статье приведён пример быстрой интеграции Altcraft mSDK для **React Native**.

Шаг 0. Предварительные условия

- Проект React Native создан и запускается на iOS/Android.
- Нативные push-провайдеры интегрированы в проект:
  - Android: FCM, HMS, RuStore
  - iOS: APNS, FCM, HMS
- На нативной стороне выполнена настройка Altcraft mSDK:
  - зарегистрированы провайдеры (JWT/APNS/FCM/HMS/Rustore)
  - настроен перехват входящих push-уведомлений и передача в SDK

Шаг 1. Установка пакета

```
npm i @altcraft/react-native-sdk
# или
yarn add @altcraft/react-native-sdk
```

Шаг 2. Конфигурация и инициализация SDK

Импортируйте SDK и выполните инициализацию на старте приложения (например, в `App.tsx`):

```
AltcraftSDK.initialize({
  apiUrl: 'https://pxl-your-instance.altcraft.com',
  rToken: null,
  appInfo: {
    appID: 'your-app-id',
    appIID: 'your-installation-id',
    appVer: '1.0.0',
  },
  providerPriorityList: null,
  enableLogging: true,

  // -------- ANDROID-ONLY (опционально) --------
  android_icon: null,
  android_pushReceiverModules: null,
  android_pushChannelName: null,
  android_pushChannelDescription: null,
});
```

Шаг 3. Выполнение подписки на push-уведомления

Подписка на push выполняется вызовом `pushSubscribe()`:

```
AltcraftSDK.pushSubscribe(
  true,
  { _fname: 'user_first_name', _lname: 'user_last_name' }, // profileFields
  null,                                                    // customFields
  null,                                                    // cats
  null,                                                    // replace
  null                                                     // skipTriggers
);
```

- `sync: true` — выполняет запрос синхронно (результат можно получить через событие SDK, если платформа его возвращает).
- `sync: false` — выполняет запрос асинхронно.

(Опционально) Подписка на события SDK

Для получения результатов вызовов и внутренних событий SDK используйте `subscribeToEvents()`:

```
AltcraftSDK.subscribeToEvents((event: SdkEvent) => {
  // event.function — имя функции SDK, которая вызвала событие
  // event.type — 'event' | 'error' | 'retryError'
  // event.code / event.message — код и сообщение
  // event.value — дополнительные данные (если есть)
  console.log('[AltcraftSDK event]', event);
});
```

Чтобы отписаться:

```
AltcraftSDK.unsubscribeFromEvent();
```

Конфигурация SDK

Предварительные условия

- SDK провайдеров push-уведомлений интегрированы в проект приложения (см. инструкции по интеграции push-провайдеров).
- Выполнена загрузка пакета npm с помощью одной из команд:

```
yarn add altcraft-react-native-sdk
# or
npm i altcraft-react-native-sdk
```

Подготовка приложения

Передача JWT-токена в SDK (опционально)

Реализация JWT требуется, если используется JWT-аутентификация API-запросов. JWT подтверждает, что пользовательские идентификаторы аутентифицированы приложением.

Реализация JWT-аутентификации обязательна, если используется тип матчинга отличный от push-данных из подписки (например, идентификатор пользователя — email или телефон).

Настройка JWT осуществляется аналогично нативным SDK платформ (Android/iOS).

Подготовка к работе с push-провайдерами

Для **iOS** подготовка к работе с push-провайдерами производится **нативно**. Подробная инструкция описана в этом разделе статьи по настройке mSDK.

Инициализация SDK

Параметры инициализации

Для передачи параметров конфигурации используется объект `AltcraftConfig`:

```
  apiUrl: string;

  rToken?: string | null;
  appInfo?: {
    appID: string;
    appIID: string;
    appVer: string;
  } | null;

  providerPriorityList?: string[] | null;
  enableLogging?: boolean | null;

  // -------- ANDROID-ONLY --------
  android_icon?: number | null;
  android_pushReceiverModules?: string[] | null;
  android_pushChannelName?: string | null;
  android_pushChannelDescription?: string | null;
};
```

Описание параметров:

apiUrl `string`
Обязательный: **Да**
Описание: URL конечной точки Altcraft API

---

rToken `string | null`
По умолчанию: `null`
Обязательный: Нет
Описание: Ролевой токен Altcraft (идентифицирует ресурс/БД/аккаунт). Используется, если единственный тип матчинга — push-токен устройства.

---

appInfo `{ appID: string; appIID: string; appVer: string } | null`
По умолчанию: `null`
Обязательный: Нет
Описание: Базовые метаданные приложения.

---

enableLogging `boolean | null`
По умолчанию: `null`
Обязательный: Нет
Описание: Включает/отключает ведение логов.

---

providerPriorityList `string[] | null`
По умолчанию: `null`
Обязательный: Нет
Описание: Список приоритетов провайдеров (по индексу: **0** — самый приоритетный).

Пример использования:

Android

```
providerPriorityList: ['android-firebase', 'android-huawei', 'android-rustore']
```

iOS

```
providerPriorityList: ['ios-apns', 'ios-firebase', 'ios-huawei']
```

---

android_icon `number | null` *(только Android)*
По умолчанию: `null`
Обязательный: Нет
Описание: ID ресурса `drawable`, используемого как иконка уведомлений

---

android_pushReceiverModules `string[] | null` *(только Android)*
По умолчанию: `null`
Обязательный: Нет
Описание: Список имён пакетов, в которых находятся реализации нативного `AltcraftPushReceiver`.

---

android_pushChannelName `string | null` *(только Android)*
По умолчанию: `null`
Обязательный: Нет
Описание: Базовое имя канала push-уведомлений (видно в настройках Android).

---

android_pushChannelDescription `string | null` *(только Android)*
По умолчанию: `null`
Обязательный: Нет
Описание: Описание push-канала (видно в настройках Android).

Выполнение инициализации

Для инициализации SDK используйте функцию `initialize(config)`:

```
await AltcraftSDK.initialize({
  apiUrl: 'https://pxl-example.altcraft.com',

  // optional
  rToken: null,
  enableLogging: true,
  providerPriorityList: ['android-firebase', 'android-huawei', 'android-rustore'],

  // android-only (optional)
  android_icon: null,
  android_pushReceiverModules: null,
  android_pushChannelName: null,
  android_pushChannelDescription: null,
});
```

Функционал SDK

TIP

Предварительно вам необходимо настроить SDK для работы с вашим приложением. Подробная инструкция находится здесь.

Работа со статусами подписки

Изменение статуса подписки

```
AltcraftSDK (React Native)
└─ pushSubscriptionFunctions
   // Подписка (статус = SUBSCRIBED)
   ├─ function pushSubscribe(
   │     sync: boolean | null = true,
   │     profileFields: Record<string, unknown> | null = null,
   │     customFields: Record<string, unknown> | null = null,
   │     cats: CategoryData[] | null = null,
   │     replace: boolean | null = null,
   │     skipTriggers: boolean | null = null
   │   ): void
   // Приостановка (статус = SUSPENDED)
   ├─ function pushSuspend(
   │     sync: boolean | null = true,
   │     profileFields: Record<string, unknown> | null = null,
   │     customFields: Record<string, unknown> | null = null,
   │     cats: CategoryData[] | null = null,
   │     replace: boolean | null = null,
   │     skipTriggers: boolean | null = null
   │   ): void
   // Отписка (статус = UNSUBSCRIBED)
   └─ function pushUnSubscribe(
         sync: boolean | null = true,
         profileFields: Record<string, unknown> | null = null,
         customFields: Record<string, unknown> | null = null,
         cats: CategoryData[] | null = null,
         replace: boolean | null = null,
         skipTriggers: boolean | null = null
      ): void
```

- `function pushSubscribe()` — выполняет подписку на push-уведомления;
- `function pushUnSubscribe()` — отменяет подписку на push-уведомления;

- `function pushSuspend()` — приостанавливает подписку на push-уведомления (уведомления не приходят, но при этом не создается событие отписки в профиле пользователя);
- `function unSuspendPushSubscription()` — используется для создания `LogIn`-, `LogOut`-переходов.

Функции этой группы имеют одинаковую сигнатуру, содержащую следующие параметры:

---

sync: `Boolean`

По умолчанию: `true`
Обязательный: Нет
Описание: Флаг, устанавливающий синхронность выполнения запроса.

Успешное выполнение запроса:

В случае успешного выполнения запроса данной группы функций будет создано событие с кодом **230, 231 или 232**, содержащее значение `event.value`, определяемое в зависимости от флага синхронизации:

Если `sync == true`

```
ResponseWithHttpCode
  ├─ httpCode: 200
  └─ response
     ├─ error: 0
     ├─ errorText: ""
     └─ profile
        ├─ id: "your id"
        ├─ status: "subscribed"
        ├─ isTest: false
        └─ subscription
           ├─ subscriptionId: "your subscriptionId"
           ├─ hashId: "7f31a9c4"
           ├─ provider: "android-firebase"
           ├─ status: "subscribed"
           ├─ fields
           │  ├─ _os_ver: "{\"raw\":\"14\",\"ver\":\"[\\\"14.0\\\", \\\"0.0\\\",
\\\"0.0\\\"]\"}"
           │  ├─ _device_type: "mob"
           │  ├─ _ad_track: "true"
           │  ├─ _device_name: "Pixel"
           │  ├─ _os_language: "en"
           │  ├─ _os_tz: "+0100"
           │  ├─ _os: "Android"
           │  ├─ _device_model: "Pixel 7"
           │  └─ _app_ver: "1.0.0"
           └─ cats
              └─ [ { name: "developer_news", title: "dev_news", steady: false, active:
false } ]
```

При **синхронном** запросе в значении события `event.value` по ключу `"response_with_http_code"` доступны:

- **httpCode** — транспортный код ответа;
- **response** — export type, содержащий:
  - `error: number | null` — внутренний код ошибки сервера *(0, если ошибок нет)*,
  - `errorText: string | null` — текст ошибки *(пустая строка, если ошибок нет)*,
  - `profile: ProfileData | null` — данные профиля, если запрос успешный:
    - информация о профиле (**ProfileData**)
    - подписка (**Subscription**)
    - если запрос завершился с ошибкой, то вернется только `profile = null`

▶ export type

Если sync == false

```
ResponseWithHttpCode
  ├─ httpCode: number | null
  └─ response: Response | null
      ├─ error?: number | null
      ├─ errorText?: string | null
      └─ profile?: ProfileData | null // всегда null
```

При **асинхронном** запросе в значении события `event.value` по ключу `"response_with_http_code"` доступны:

- **httpCode** — транспортный код ответа;
- **response** — export type, содержащий:
  - `error: number | null` — внутренний код ошибки сервера *(0, если ошибок нет)*,
  - `errorText: string | null` — текст ошибки *(пустая строка, если ошибок нет)*,
  - `profile: ProfileData | null` — для асинхронного запроса всегда равен `null`.

Выполнение запроса с ошибкой:

Если запрос данной группы функций завершился ошибкой, будет создано событие со следующими кодами:

Операции без автоматического повтора попытки на стороне SDK:

- **430** – подписка на уведомления;
- **431** — приостановка подписки;
- **432** — отписка.

Операции с автоматическим повтором попытки на стороне SDK:

- **530** – подписка на уведомления;
- **531** — приостановка подписки;
- **532** — отписка.

Содержимое события:

- только `httpCode`, если сервер Altcraft был **недоступен**;
- `error` и `errorText`, если сервер **вернул ошибку**.

Получение значений событий

```
const RESPONSE_CODES = new Set<number>([
  230, 231, 232,
  430, 431, 432,
  531, 532, 533,
]);

subscribeToEvents((event: SdkEvent) => {
  if (event.code == null) return;
  if (!RESPONSE_CODES.has(event.code)) return;

  const raw = event.value?.['response_with_http_code'] as unknown;
  const responseWithHttp = raw as ResponseWithHttpCode | null;
  if (!responseWithHttp) return;

  const httpCode = responseWithHttp.httpCode;

  const response = responseWithHttp.response;
  const error = response?.error ?? null;
  const errorText = response?.errorText ?? null;

  const profile = response?.profile ?? null;
  const profileId = profile?.id ?? null;
  const profileStatus = profile?.status ?? null;
  const profileIsTest = profile?.isTest ?? null;

  const subscription = profile?.subscription ?? null;
  const subscriptionId = subscription?.subscriptionId ?? null;
  const hashId = subscription?.hashId ?? null;
  const provider = subscription?.provider ?? null;
  const subscriptionStatus = subscription?.status ?? null;

  const fields = subscription?.fields ?? null;

  const cats = subscription?.cats ?? null;
  const firstCat = cats?.[0] ?? null;
  const catName = firstCat?.name ?? null;
  const catTitle = firstCat?.title ?? null;
  const catSteady = firstCat?.steady ?? null;
  const catActive = firstCat?.active ?? null;
});
```

---

profileFields: `Record<string, unknown> | null`

По умолчанию: `null`
Обязательный: Нет
Описание: Объект, содержащий поля профиля.

Параметр может принимать как **системные поля** (например, `_fname` — имя или `_lname` — фамилия), так и **опциональные** (заранее создаются вручную в интерфейсе платформы). Допустимые структуры (JSON-совместимые):

- **Скалярные значения**:
  - String
  - Boolean
  - Number
  - `null`
- **Объекты**: `Record<string, unknown>`
- **Списки**: `unknown[]`

Если передано невалидное опциональное поле, запрос завершится с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: with field "название_поля": Incorrect field
```

---

customFields: `Record<string, unknown> | null`

По умолчанию: `null`
Обязательный: Нет
Описание: Объект, содержащий поля подписки.

Параметр может принимать как **системные поля** (например, `_device_model` — модель устройства или `_os` — операционная система), так и **опциональные** (заранее создаются вручную в интерфейсе платформы). Допустимые типы значений (JSON-совместимые, **только скаляры**):

- String
- Boolean
- Number
- `null`

Если передано невалидное опциональное поле, запрос завершится с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: field "название_поля" is not valid: failed
convert custom field
```

> **TIP: Обратите внимание**
>
> Большая часть системных полей подписки автоматически собирается SDK и добавляется к push-запросам. К таким системным полям относятся: `"_os"`, `"_os_tz"`, `"_os_language"`, `"_device_type"`, `"_device_model"`, `"_device_name"`, `"_os_ver"`, `"_ad_track"`, `"_ad_id"`.

---

cats: `CategoryData[] | null`

По умолчанию: `null`
Обязательный: Нет
Описание: Категории подписок.

Структура категории:

```
  name?: string | null;
  title?: string | null;
  steady?: boolean | null;
  active?: boolean | null;
};
```

При отправке push-запроса с указанием категорий используйте только поля `name` (название категории) и `active` (статус активности категории), другие поля не используются в обработке запроса. Поля `title` и `steady` заполняются при получении информации о подписке.

Пример запроса:

```
const cats: CategoryData[] = [
  { name: 'football', active: true },
  { name: 'hockey', active: true },
];
```

Категории используемые в запросе должны быть предварительно созданы и добавлены к ресурсу в платформе Altcraft. Если в запросе будет использовано поле, которое не добавлено в ресурс — запрос вернется с ошибкой:

```
SDK error: 430
http code: 400
error: 400
errorText: Platform profile processing error: field "subscriptions.cats" is not valid:
category not found in resource
```

---

replace: `boolean | null`

По умолчанию: `null`
Обязательный: Нет
Описание: При активации флага все подписки других профилей с тем же push-токеном в текущей базе данных переводятся в статус `unsubscribed` после успешного запроса.

---

skipTriggers: `boolean | null`

По умолчанию: `null`
Обязательный: Нет

Описание: При активации флага профиль, содержащий данную подписку, будет игнорироваться в триггерах рассылок и сценариев.

Примеры реализации запроса

Пример выполнения запроса подписки на push-уведомления

Минимальная рабочая настройка:

```
AltcraftSDK.pushSubscribe();
```

Передача всех доступных параметров:

```
AltcraftSDK.pushSubscribe(
  true,
  { _fname: 'Andrey', _lname: 'Pogodin' },
  { developer: true },
  [{ name: 'developer_news', active: true }],
  false,
  false
);
```

TIP: Обратите внимание

Для `pushSubscribe`, `pushSuspend`, `pushUnSubscribe` предусмотрен автоматический повтор запроса со стороны SDK если http-код ответа находится в диапазоне **500…599**. Запрос не повторяется, если код ответа в этот диапазон не входит.

Функция `unSuspendPushSubscription()`

Функция `async function unSuspendPushSubscription(): Promise<ResponseWithHttpCode | null>` предназначена для создания `LogIn`-, `LogOut`-переходов. Она работает следующим образом:

- проводит поиск подписок с тем же push-токеном, что и текущий, не относящихся к профилю, на который указывает текущий JWT-токен;
- меняет статус для найденных подписок с `subscribed` на `suspended`;
- меняет статус в подписках профиля, на который указывает текущий JWT, с `suspended` на `subscribed` (если профиль, на который указывает JWT, существует и в нем содержатся подписки);
- возвращает `ResponseWithHttpCode | null`, где `response.profile` — текущий профиль, на который указывает JWT (если профиля не существует, то вместо `response.profile` вернется `null`).

Рекомендуемая реализация LogIn-, LogOut-переходов

`LogIn`-переход

- Анонимный пользователь входит в приложение. Данному пользователю присвоен `JWT_1`, указывающий на базу данных **#1Anonymous**;

- Выполнена подписка на push-уведомления, профиль создан в базе данных **#1Anonymous**;
- Пользователь регистрируется, ему присваивается `JWT_2`, указывающий на базу данных **#2Registered**;
- Вызывается функция `unSuspendPushSubscription()` — подписка анонимного пользователя в базе данных **#1Anonymous** приостанавливается;
- Выполняется поиск профиля в базе данных **#2Registered** для восстановления подписки;
- Так как подписки с таким push-токеном в базе данных **#2Registered** не существует, функция `unSuspendPushSubscription()` вернет `null`;
- После получения значения `null` можно выполнить запрос `pushSubscribe()`, который создаст новый профиль в базе **#2Registered**.

`LogOut`-переход

- Пользователь выполнил выход из профиля на стороне приложения (`LogOut`);
- Пользователю присваивается `JWT_1`, указывающий на базу данных **#1Anonymous**;
- Вызывается функция `unSuspendPushSubscription()`, которая приостановит подписку в базе данных **#2Registered** и сменит статус подписки в базе **#1Anonymous** на `subscribed`;
- Запрос вернет профиль **#1Anonymous != null** — подписка уже существует, новая не требуется.

Пример реализации:

```
async function unSuspend(logIn: boolean): Promise<void> {
  setAuth(logIn);

  const result: ResponseWithHttpCode | null =
    await AltcraftSDK.unSuspendPushSubscription();

  if (!result) {
    AltcraftSDK.pushSubscribe();
    return;
  }

  const httpCode = result.httpCode ?? null;
  const subscription = result.response?.profile?.subscription ?? null;

  if (httpCode === 200 && subscription == null) {
    AltcraftSDK.pushSubscribe();
  }
}

  void unSuspend(true);
}

  void unSuspend(false);
}
```

Запрос статуса подписки

```
AltcraftSDK
└── function getStatusOfLatestSubscription(): Promise<ResponseWithHttpCode | null>
└── function getStatusForCurrentSubscription(): Promise<ResponseWithHttpCode | null>
└── function getStatusOfLatestSubscriptionForProvider(
       provider: string | null
    ): Promise<ResponseWithHttpCode | null>
```

Функции запроса статуса подписки:

- `function getStatusOfLatestSubscription()` — статус последней подписки профиля;
- `function getStatusForCurrentSubscription()` — статус подписки для текущего токена/провайдера;
- `function getStatusOfLatestSubscriptionForProvider()` — статус последней подписки по провайдеру. Если указан `null`, то используется провайдер текущего токена.

`function getStatusOfLatestSubscription(): Promise<ResponseWithHttpCode | null>`

Функция получения статуса последней подписки профиля. Возвращает `ResponseWithHttpCode | null`, содержащий `response?.profile?.subscription` — последнюю созданную подписку в профиле. Если такой подписки не существует, будет передан `null`.

Пример использования:

```
const result = await AltcraftSDK.getStatusOfLatestSubscription();
```

`function getStatusForCurrentSubscription(): Promise<ResponseWithHttpCode | null>`

Функция получения статуса подписки для текущего токена/провайдера. Возвращает `ResponseWithHttpCode | null`, содержащий `response?.profile?.subscription` — подписку, найденную по текущему push-токену и провайдеру. Если такой подписки не существует, будет передан `null`.

Пример использования:

```
const result = await AltcraftSDK.getStatusForCurrentSubscription();
```

`function getStatusOfLatestSubscriptionForProvider(provider: string | null): Promise<ResponseWithHttpCode | null>`

Функция получения статуса последней подписки по провайдеру. Возвращает `ResponseWithHttpCode | null`, содержащий `response?.profile?.subscription` — последнюю созданную подписку с указанным провайдером. Если провайдер не указан (`provider = null`), используется провайдер текущего токена. Если такой подписки не существует, будет передан `null`.

Пример использования:

```
const result = await AltcraftSDK.getStatusOfLatestSubscriptionForProvider(null);
```

Ниже представлен пример извлечения данных о профиле, подписке и категориях из ответа функций получения статуса. Данный подход актуален для всех функций получения статуса:

Данные из функций получения статуса

```
async function readStatus(): Promise<void> {
  const result = await AltcraftSDK.getStatusForCurrentSubscription();
  if (!result) return;

  const httpCode = result.httpCode ?? null;

  const response = result.response ?? null;
  const error = response?.error ?? null;
  const errorText = response?.errorText ?? null;

  const profile = response?.profile ?? null;
  const subscription = profile?.subscription ?? null;

  const cats = subscription?.cats ?? null;
}
```

Управление push-токенами провайдеров

TIP

На нативной стороне приложения можно использовать функции, указанные в статьях для соответствующих платформ (для Android, для iOS)

```
AltcraftSDK
|
└   // Получить текущие данные токена устройства
  ├──   function getPushToken(): Promise<TokenData | null>
  |
  |  // Сохранить/обновить токен вручную для конкретного провайдера
  └──   function setPushToken(provider: string, token: string | null): Promise<void>
```

Функции для работы с токеном провайдера в SDK:

- `function setPushToken()` — установка push-токена устройства и провайдера;
- `function getPushToken()` — получение текущего push-токена.

`setPushToken(provider, token)`

Функция предназначена для установки push-токена устройства и провайдера.

- В React Native **не передаётся** `context` — сохранение и хранение токена выполняется на нативной стороне SDK.
- `token: null` — означает очистку токена для указанного провайдера.

Пример использования:

```
// или
// import AltcraftSDK from '@altcraft-react-native-sdk/...';

await AltcraftSDK.setPushToken('FCM', token);
```

Пример передачи токена при получении нового токена:

```
// или
// import AltcraftSDK from '@altcraft-react-native-sdk/...';

async function onNewToken(provider: string, token: string) {
  await AltcraftSDK.setPushToken(provider, token);
}
```

`getPushToken()`

Функция возвращает текущие данные push-токена устройства и провайдера в виде `TokenData`.

Формат возвращаемых данных:

- `TokenData` — объект с полями:
  - `provider: string`
  - `token: string`

Если токен недоступен — будет возвращено `null`.

Пример использования:

```
// или
// import AltcraftSDK from '@altcraft-react-native-sdk/...';

const tokenData = await AltcraftSDK.getPushToken();
```

Пример запроса получения данных push-токена:

```
// или
// import AltcraftSDK from '@altcraft-react-native-sdk/...';

async function readPushToken() {
  const data = await AltcraftSDK.getPushToken();

  const provider = data?.provider ?? null;
  const token = data?.token ?? null;

  return { provider, token };
}
```

Регистрация и передача push-уведомлений в SDK

В нативном коде

Для настройки работы с push-уведомлениями (получение и передача в SDK) в React Native воспользуйтесь инструкциями **нативных SDK**:

Android:

- Подключение push-провайдеров
- Подготовка SDK к работе с push-провайдерами
- Передача push-уведомлений в SDK

Обратите внимание, в RN-проекте на Android точка входа приложения — `MainApplication : Application(), ReactApplication`. Используйте `MainApplication` для регистрации провайдеров , **а не отдельный** `Application` **-класс**.

iOS:

- Подключение push-провайдеров
- Настройка AppDelegate приложения
- Подготовка NSE
- Работа с push-уведомлениями

> **INFO: Важно**
>
> **npm**-библиотека Altcraft `@altcraft/react-native-sdk` уже включает нативные SDK. Все нативные классы доступны в коде iOS- и Android-приложения сразу после установки пакета. **Не подключайте** SDK дополнительно через Gradle, CocoaPods или Swift Package Manager.

В React Native (только для Android)

В Android входящие push-уведомления можно получать в React-части приложения, используя одного из провайдеров:

- **FCM (Firebase Cloud Messaging)**: `@react-native-firebase/app` и `@react-native-firebase/messaging`
- **HMS (Huawei Mobile Services)**: `@hmscore/react-native-hms-push`

FCM

Установите необходимые зависимости:

```
npm i @react-native-firebase/app @react-native-firebase/messaging
# или
yarn add @react-native-firebase/app @react-native-firebase/messaging
```

Обработка push-уведомлений через FCM происходит следующим образом:

Background push

```
messaging().setBackgroundMessageHandler(async (rm) => {
  console.log('altcraft_lib push background');

  const payload = RemoteMessagetoStringMap(rm?.data);
  if (payload) {
    AltcraftSDK.takePush(payload);
  }
});
```

Foreground push

```
function registerFCMForegroundPush(): void {
  if (fcmFg || Platform.OS !== 'android') return;
  fcmFg = true;

  messaging().onMessage(async (rm) => {
    console.log('altcraft_lib push foreground');

    const payload = RemoteMessagetoStringMap(rm?.data);
    if (payload) {
      AltcraftSDK.takePush(payload);
    }
  });
}
```

Для передачи push-уведомления в SDK используется метод:

```
AltcraftSDK.takePush(message: Record<string, string>): void
```

где `message` — данные push-уведомления, предварительно преобразованные из `RemoteMessage` в формат `Record<string, string>`.

Данный подход можно объединить с использованием нативного push-сервиса `FirebaseMessagingService`: уведомления могут обрабатываться нативно и параллельно передаваться в React Native/Altcraft SDK для бизнес-логики.

HMS

Установите необходимые зависимости:

```
npm i @hmscore/react-native-hms-push
# или
yarn add @hmscore/react-native-hms-push
```

Обработка push-уведомлений через HMS происходит следующим образом:

Background push

```
HmsPushMessaging.setBackgroundMessageHandler(async (dataMessage) => {
  console.log('altcraft_lib push background');

  const payload = new RNRemoteMessage(dataMessage).getDataOfMap();
  if (payload) {
    AltcraftSDK.takePush(payload);
  }
});
```

Foreground push

```
let hmsFg = false;

function registerHMSForegroundPush(): void {
  if (hmsFg || Platform.OS !== 'android') return;
  hmsFg = true;

  HmsPushEvent.onRemoteMessageReceived((dataMessage) => {
    console.log('altcraft_lib push foreground');

    const payload = new RNRemoteMessage(dataMessage).getDataOfMap();
    if (payload) {
      AltcraftSDK.takePush(payload);
    }
  });
}
```

Для передачи push-уведомления в SDK используется метод:

```
AltcraftSDK.takePush(message: Record<string, string>): void
```

где `message` — данные push-уведомления, предварительно преобразованные из `RemoteMessage` в формат `Record<string, string>`.

Данный подход можно объединить с использованием нативного push-сервиса `HmsMessageService`: уведомления могут обрабатываться нативно и параллельно передаваться в React Native/Altcraft SDK для бизнес-логики.

Мобильные события

```
// Функция объекта AltcraftSDK (React Native)

AltcraftSDK
└── function mobileEvent(
        sid: string,
        eventName: string,
        sendMessageId?: string | null,
        payload?: Record<string, unknown> | null,
        matching?: Record<string, unknown> | null,
        matchingType?: string | null,
        profileFields?: Record<string, unknown> | null,
        subscription?: Subscription | null,
        utm?: UTM | null
    ): void
```

Для регистрации мобильного события используйте функцию `mobileEvent()` из `AltcraftSDK`. Она имеет следующие параметры:

---

sid: `string`

Обязательный: **Да**
Описание: Строковый идентификатор пикселя, к которому привязываются мобильные события.

---

eventName: `string`

Обязательный: **Да**
Описание: Имя мобильного события.

---

sendMessageId: `string | null`

Обязательный: Нет
Описание: SMID-идентификатор отправленного сообщения (если событие связано с конкретной рассылкой).

---

payload: `Record<string, unknown> | null`

Обязательный: Нет
Описание: Данные события — карта со строковыми ключами. В RN-бридже значения сериализуются в `string`:

- `string` —> как есть
- `number` / `boolean` / `bigint` —> `String(value)`
- `object` / `array` —> `JSON.stringify(value)`
- `null` / `undefined` —> ключ не передаётся (поле удаляется из payload)

> **TIP: Обратите внимание**
>
> Несериализуемые объекты (например, `Date` без преобразования, кастомные классы, функции) будут переданы как строка через `String(value)` или могут привести к некорректным данным. Рекомендуется передавать JSON-совместимые значения.

---

matching: `Record<string, unknown> | null`

Обязательный: Нет
Описание: Карта, в которую можно передавать значения с типами и идентификаторами матчинга (например, `{"email": "user@example.com"}`). В RN-бридже значения сериализуются в `string` по тем же правилам, что и `payload`.

---

matchingType: `string | null`

Обязательный: Нет
Описание: Тип матчинга.

---

profileFields: `Record<string, unknown> | null`

Обязательный: Нет
Описание: Поля профиля — карта со строковыми ключами. В RN-бридже значения сериализуются в `string` по тем же правилам, что и `payload`.

> **TIP: Обратите внимание**
>
> Параметр `profileFields` используется только при работе с JWT-авторизацией.

---

subscription: `Subscription | null`

Обязательный: Нет
Описание: Параметр добавления подписки для выбранного канала.

Значение параметра — одна из реализаций union-типа `Subscription`:

- `EmailSubscription` — email-подписка
- `SmsSubscription` — SMS-подписка
- `PushSubscription` — push-подписка
- `CcDataSubscription` — подписка в Telegram, Whatsapp, Viber, Notify.

> **TIP: Обратите внимание**
>
> Параметр `subscription` используется только при работе с JWT-авторизацией.

Общие поля подписки (для всех реализаций)

Поле	Тип	Обязательный	Описание
`type`	`string`	Да	Тип канала (дискриминатор).
`resource_id`	`number`	Да	Идентификатор ресурса/источника подписки.
`status`	`string | null`	Нет	Статус подписки (например, активна/приостановлена).
`priority`	`number | null`	Нет	Приоритет доставки.
`custom_fields`	`{ [key: string]: string } | null`	Нет	Пользовательские поля подписки.
`cats`	`string[] | null`	Нет	Категории подписки.

Варианты подписки

EmailSubscription (`type = "email"`)

Поле	Тип	Обязательный	Описание
`resource_id`	`number`	Да	ID ресурса Altcraft
`email`	`string`	Да	Адрес электронной почты

SmsSubscription (`type = "sms"`)

Поле	Тип	Обязательный	Описание
`resource_id`	`number`	Да	ID ресурса Altcraft
`phone`	`string`	Да	Номер телефона в международном формате

PushSubscription (`type = "push"`)

Поле	Тип	Обязательный	Описание
`resource_id`	`number`	Да	ID ресурса Altcraft
`provider`	`string`	Да	Провайдер (например, `"android-firebase"`)
`subscription_id`	`string`	Да	Уникальный идентификатор подписки у провайдера

CcDataSubscription (`type = "cc_data"`)

Поле	Тип	Обязательный	Описание
`resource_id`	`number`	Да	ID ресурса Altcraft
`channel`	`string`	Да	Один из: `"telegram_bot"` , `"whatsapp"` , `"viber"` , `"notify"`
`cc_data`	`{ [key: string]: string }`	Да	Канал-специфичные данные (например, chat ID, номер, токены)

---

utm: `UTM | null`

Обязательный: Нет
Описание: UTM-метки. Передаются структурой `UTM` , где каждый вид UTM — отдельное поле:

```
    campaign?: string | null;
    content?: string | null;
    keyword?: string | null;
    medium?: string | null;
    source?: string | null;
    temp?: string | null;
};
```

Примеры использования

```
AltcraftSDK.mobileEvent('your sid', 'app_open');
```

Передача данных через payload

```
AltcraftSDK.mobileEvent(
  'your sid',
  'purchase',
  null,
  {
    orderId: 'A-1001',
    amount: 3990,
    currency: 'RUB',
    items: [{ sku: 'sku-1', qty: 2 }],
  }
);
```

Передача UTM-меток

```
AltcraftSDK.mobileEvent(
  'your sid',
  'app_install',
  null,
  null,
  null,
  null,
  null,
  null,
  {
    campaign: 'your campaign utm',
    content: 'your content utm',
    keyword: 'your keyword utm',
    medium: 'your medium utm',
    source: 'your source utm',
    temp: 'your temp utm',
  }
);
```

Передача `matching` / `matchingType`

```
AltcraftSDK.mobileEvent(
  'your sid',
  'profile_update',
  null,
  { source: 'settings' },
  { email: 'user@example.com' },
  'email'
);
```

*Передача `subscription` (JWT)

```
    const sub: PushSubscription = {
      type: 'push',
      resource_id: 10,
      provider: 'android-firebase',
      subscription_id: 'provider-sub-id',
      status: 'active',
      priority: 1,
      custom_fields: { utm: 'your utm tag' },
      cats: ['promo', 'news'],
    };

    AltcraftSDK.mobileEvent(
      'your sid',
      'jwt_bind_subscription',
      null,
      null,
      null,
      null,
      null,
      sub,
      null
    );
```

TIP: Обратите внимание

При передаче `subscription` в RN-бридже часть полей сериализуется в строковый формат (например, `custom_fields` и `cats` передаются как JSON-строки). Передавайте JSON-совместимые значения.

Получение событий SDK в приложении

В React Native события SDK передаются из нативного слоя через `NativeEventEmitter`. В приложении может быть **только один активный подписчик** на события SDK.

События эмитятся под именем **`AltcraftSdkEvent`**. Каждое событие, получаемое в JS, имеет следующий формат:

```
  function: string;                 // имя функции SDK
  code: number | null;              // код события
  message: string;                  // сообщение события
  type: 'event' | 'error' | 'retryError';
  value?: Record<string, unknown> | null; // дополнительная нагрузка
};
```

Типы событий:

- `event` — информационные события и успешные операции;
- `error` — ошибки выполнения;
- `retryError` — ошибки операций, для которых SDK выполняет автоматический повтор.

Подписка на события

Для подписки используйте функцию `subscribeToEvents`.

При повторной подписке предыдущий подписчик автоматически удаляется.

```
subscribeToEvents((event) => {
  switch (event.type) {
    case 'event':
      // обработка успешного события
      break;

    case 'error':
      // обработка ошибки
      break;

    case 'retryError':
      // обработка ошибки с автоповтором
      break;
  }
});
```

Отписка от событий

Для прекращения получения событий используйте `unsubscribeFromEvent`.

```
unsubscribeFromEvent();
```

После вызова подписчик удаляется, и события SDK больше не доставляются в JavaScript.

Список всех событий SDK

Список событий

Код	Описание
200	SDK configuration is installed
201	push provider set

Код	Описание
202	received a notification unrelated to the Altcraft Platform
203	received Altcraft push notification
204	push is posted
205	SDK data has been cleared
206	waiting for SDK initialization to complete
207	waiting for push token update to complete
220	receiver override event
230	subscribe request succeeded
231	push suspend request succeeded
232	push unsubscribed request succeeded
233	push update request succeeded
234	push unsuspend request succeeded
235	profile status request succeeded
236	push event delivered successfully
237	mobile event delivered successfully
401	the configuration is not set
402	userTag is null. It is impossible to identify the user
404	SDK initialization timeout has expired
405	mobile event parts is null
422	unsuspend request data is null
423	profile request data is null
430	subscribe request failed
431	suspend request failed
432	unsubscribed request failed
433	update request failed
434	unsuspend request failed
435	status request failed
436	push event delivery failed
437	mobile event request failed
450	push data is null

Код	Описание
451	uid in the push data is null or empty, it is impossible to send a push event to the server
452	error uploading the notification image
453	couldn’t create notification
454	foreground info is null
455	the notification channel has not been created
471	invalid provider. Available - android-firebase, android-huawei, android-rustore
472	invalid customFields: not all values are primitives
480	subscribe retry limit reached (request removed from DB)
484	push event retry limit reached (request removed from DB)
485	mobile event retry limit reached (request removed from DB)
501	config data is null
502	current push token is null
503	userTag is null. It is impossible to identify the user
504	no permission to send notifications
505	no internet connection, retry when connection is restored
506	failed to update the push token. The subscription request was rejected
520	push subscribe request data is null
521	token update request data is null
524	push event request data is null
525	mobile event request data is null
529	common data is null
530	subscribe request failed (retryable)
531	token update request failed (retryable)
534	push event delivery failed (retryable)
535	mobile event delivery failed (retryable)
540	JWT token is null
541	matching mode is null
542	JWT payload exceeds allowed size (16 KB limit): input rejected to prevent DoS
543	JWT does not contain a payload
544	auth data is null

Код	Описание
545	matching claim does not contain a matching ID
560	response data is null

Дополнительные функции SDK

Очистка данных SDK

`AltcraftSDK.clear();`

Функция позволяет выполнить очистку данных SDK и отменить работу всех фоновых задач, которые ожидают выполнения.

Ручная регистрация push-событий

> **TIP: Обратите внимание**
>
> Эти функции применяются только в том случае, если вы реализуете собственную логику обработки уведомлений и не передаёте их в SDK автоматически.

Android

В React Native доступны функции ручной регистрации push-событий на стороне JS:

- `deliveryEvent(message?: Record<string, string> | null, messageUID?: string | null)` — регистрация доставки уведомления;
- `openEvent(message?: Record<string, string> | null, messageUID?: string | null)` — регистрация открытия уведомления.

Параметры:

- `message: Record<string, string> | null` — data payload уведомления в виде словаря строк.
- `messageUID: string | null` — идентификатор сообщения.

Правила передачи:

- можно передать **только** `message` (SDK возьмёт идентификатор сообщения из payload, если он в нём присутствует);
- можно передать **только** `messageUID` (если payload недоступен, но идентификатор известен);
- если оба параметра `null` — событие не будет зарегистрировано корректно (нет данных для идентификации сообщения).

iOS

Ручная регистрация push-событий SDK на стороне React Native **не реализуется**. Если требуется ручная регистрация событий, выполните её в нативной части приложения.

См. инструкцию для iOS (ручная регистрация push-событий).

Публичный API SDK

default export AltcraftSDK

```
AltcraftSDK
// Инициализация SDK и установка конфигурации
├─ function initialize(config: AltcraftConfig): Promise<void>
// Полная очистка данных SDK
├─ function clear(): Promise<void>
// Запрос разрешений на уведомления (платформозависимо)
├─ function requestNotificationPermission(): void

// Подписка на push (fire-and-forget)
├─ function pushSubscribe(
│    sync?: boolean | null,
│    profileFields?: Record<string, unknown> | null,
│    customFields?: Record<string, unknown> | null,
│    cats?: CategoryData[] | null,
│    replace?: boolean | null,
│    skipTriggers?: boolean | null
│  ): void
├─ function pushSuspend(...): void
└─ function pushUnSubscribe(...): void

// Статусы подписки (async)
├─ function unSuspendPushSubscription(): Promise<ResponseWithHttpCode | null>
├─ function getStatusOfLatestSubscription(): Promise<ResponseWithHttpCode | null>
├─ function getStatusOfLatestSubscriptionForProvider(provider?: string | null):
Promise<ResponseWithHttpCode | null>
└─ function getStatusForCurrentSubscription(): Promise<ResponseWithHttpCode | null>

// События SDK (один подписчик на стороне JS)
├─ function subscribeToEvents(handler: (event: SdkEvent) => void): void
└─ function unsubscribeFromEvent(): void

// Отправка мобильных событий
└─ function mobileEvent(
     sid: string,
     eventName: string,
     sendMessageId?: string | null,
     payload?: Record<string, unknown> | null,
     matching?: Record<string, unknown> | null,
     matchingType?: string | null,
     profileFields?: Record<string, unknown> | null,
     utm?: UTM | null
   ): void

// Токен push-провайдера
├─ function setPushToken(provider: string, token: string | null): Promise<void>
└─ function getPushToken(): Promise<TokenData | null>

// Android: входящий push payload в SDK
└─ function takePush(message: Record<string, string>): void
```

```
├─ function takePush(message: Record<string, string>): void

// Ручная регистрация push-событий (delivery/open) на стороне RN
├─ function deliveryEvent(message?: Record<string, string> | null, messageUID?: string |
null): void
└─ function openEvent(message?: Record<string, string> | null, messageUID?: string | null):
void

// iOS: запись значения в UserDefaults / App Group
└─ function setUserDefaultsValue(suiteName: string | null, key: string, value: unknown):
void
```

Event Classes

```
SdkEvent
// Универсальное событие SDK, доставляется в JS через NativeEventEmitter.
function: string
code: number | null
message: string
type: 'event' | 'error' | 'retryError'
value?: Record<string, unknown> | null
```

Exported Types

```
// Категория подписки
  name?: string | null;
  title?: string | null;
  steady?: boolean | null;
  active?: boolean | null;
};

// Токен push-провайдера устройства
  provider: string | null;
  token: string | null;
};

// Метаданные приложения
  appID: string;
  appIID: string;
  appVer: string;
};

// Ответ API вместе с HTTP-кодом
  httpCode: number | null;
  response: Response | null;
};

  error?: number | null;
  errorText?: string | null;
  profile?: ProfileData | null;
};

  id?: string | null;
  status?: string | null;
  isTest?: boolean | null;
  subscription?: SubscriptionData | null;
};

  subscriptionId?: string | null;
  hashId?: string | null;
  provider?: string | null;
  status?: string | null;
  fields?: { [key: string]: string } | null;
  cats?: CategoryData[] | null;
};

// Контейнер UTM-параметров
  campaign?: string | null;
  content?: string | null;
  keyword?: string | null;
  medium?: string | null;
  source?: string | null;
  term?: string | null;
```

```
  temp?: string | null;
};
```

AltcraftConfig

```
  apiUrl: string;

  rToken?: string | null;
  appInfo?: AppInfo | null;
  providerPriorityList?: string[] | null;
  enableLogging?: boolean | null;

  // -------- ANDROID-ONLY --------
  android_icon?: number | null;
  android_pushReceiverModules?: string[] | null;
  android_pushChannelName?: string | null;
  android_pushChannelDescription?: string | null;
};
```

Настройка провайдеров

Настройка провайдеров происходит согласно инструкции к **нативным SDK** каждой платформы. Более подробно об этом можно прочесть в этой части инструкции по использованию SDK, а ниже находятся ссылки на документацию по нативной настройке провайдеров:

Android:

- Firebase Cloud Messaging
- Huawei Mobile Services
- RuStore

iOS:

- Apple Push Notification Service
- Firebase Cloud Messaging
- Huawei Mobile Services