TGTrack service API

API сервиса позволяет интегрировать сервис «Откуда Подпсики» с другими вашими системами: телеграм боты, CRM системы, сайты и т.д.

Разные конструкторы ботов имеют разные возможности по интеграции, поэтому при работе с API есть 2 режима работы: полный и ограниченный.

- **Полная интеграция**. Работает в тех конструкторах, которые способны переслать данные, полученные от телеграма о старте бота. Например, salebot.
- **Ограниченная интеграция**. Для конструкторов ботов, которые не позволяют получить данные по пользователю телеграм, который их запустил, возможна ограниченная интеграция.

Сравнение доступных функций в разных видах интеграций:

	Полная ○ salebot ○ sambot	Средняя ○ botHelp ○ waBot ○ winwin	Минимальная smartsender
Передача активаций бота в рекламные системы Яндекс/ФБ/Гугл/ВК, оптимизация на старты бота	+	+	+
Глубокие цели: передача в рекламные системы Яндекс/ФБ/Гугл/ВК глубоких целей воронки. Например, просмотр урока №2	+	-	-
Трекинг отписок (блокировок) бота	+	-	-
Аналитическая отчетность по аудитории: источники трафика, срок жизни в боте, полный список подписчиков, ютм метки откуда пришли	+	только подписки	-
Аналитическая отчетность по подпискам: переходы в бота, конверсии в подписку	+	+	-
Ежедневные отчеты бота “подписки 24ч” “отписки 24 ч”	 + +	 + -	 - -
Досье подписчика	+	+	-

Полная интеграция для ботов доступна, если конструктор позволяет транслировать полученный от Телеграм webhook запроса без какой либо его обработки на стороне платформы бота.

См метод on_telegram_webhook

Запрос к API

При настройке доступа к API вы получаете ссылку вида

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/МЕТОД_API

Макс: https://max.tgtrack.ru/API/bot-api/v1/API_KEY/МЕТОД_API

- API_КЛЮЧ – уникальный для каждого канала или подключенного бота ключ

- МЕТОД_API – вызываемый метод API.

Данные должны передаваться методом POST в формате JSON.

Ответ API

В ответ на запрос, API возвращает статус обработки запроса. Если запрос был обработан успешно, возвращается статус "OK" и, опционально, данные. Если произошла ошибка, возвращается статус "error" с кодом ошибки и описанием.

Успешный ответ

Если запрос выполнен успешно, возвращается следующий формат ответа:

Название	Тип	Описание
status	string	Статус запроса, всегда "OK"
data	object	(optional) Данные, возвращаемые API

Пример успешного ответа в формате JSON:

```
{
  "status": "OK",
  "data": {
    "id": "123456789",
    "username": "john_doe",
    "first_name": "John",
    "last_name": "Doe"
  }
}
```

Если данные отсутствуют:

```
{
  "status": "OK",
  "mode": "production"
}
```

Если была ошибка

Если произошла ошибка при выполнении запроса, возвращается следующий формат ответа:

Название	Тип	Описание
status	string	Статус запроса, всегда "error"
error_code	int	Код ошибки
error_description	string	Описание ошибки
error_details	string	(optional) Дополнительные подробности об ошибке, если есть

Пример ошибочного ответа в формате JSON:


```
{
 "status": "error",
 "mode": "sandbox",
 "error_code": 401,
 "error_description": "Invalid token"
 "error_details": "Token was invaildated"
}
```


Проверка запросов к API

Если вы используете конструктор ботов, то протестировать интеграцию со стороны конструктора бывает непросто.

В этом случае воспользуйтесь страничкой проверки ваших запросов к API сервиса:

телеграм: https://bot-api.tgtrack.ru/last_events/

макс: https://max.tgtrack.ru/API/last_events/

Укажите на странице токен подключения к API и страничка покажет последние 100 полученных по этому токену запросов, статус их обработки и описание ошибки, если такая была.

Описание объектов API

Объекты – это набор данных, описывающих единую сущность. Например, объект «пользователь», который состоит из ID, имени, ника и т.д.

Ниже описание поддерживаемых объектов

User

Объект **User** представляет собой пользователя в Telegram. Он содержит информацию о пользователе, такую как ID, имя пользователя (ник), имя и фамилия.

Название	Тип	Описание
id	string	ID пользователя в Telegram
username	string	(optional) Имя пользователя (ник)
first_name	string	(optional) Имя пользователя
last_name	string	(optional) Фамилия пользователя
is_premium	bool	(optional) Есть ли премиум у пользователя. Если не задано, то считается, что нет

Пример:

```
{
  "id": "123456789",
  "username": "john_doe",
  "first_name": "John",
  "last_name": "Doe"
}
```

Label

Краткое описание: Объект **Label** используется в качестве дополнительного аналитического среза для данных. Он позволяет передавать дополнительные метки вместе с событиями, чтобы улучшить отчетность. Например, при передаче события "продажа" можно указать, кто из менеджеров совершил продажу и какой продукт был продан

Название	Тип	Описание
name	string	Название метки
value	string	Значение метки

Пример:

```
{
  "name": "manager",
  "value": "John Doe"
}
```

Методы API - запись

on_telegram_webhook

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/on_telegram_webhook

Макс: не применимо

Используется в полной интеграции

Метод **on_telegram_webhook** обрабатывает сообщение от телеграма, которое получил ваш бот. Это самый простой способ интеграции - просто пересылайте полученный от телеграма вебхук при старте или блокировке вашего бота. Мы сами его обработаем как нужно.

(!) никак не меняйте вебхук от телеграма (!)

просто перешлите его 1 в 1 как есть в теле запроса в методе POST

my_bot_was_started

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/my_bot_was_started

Макс: не применимо

Метод **bot_was_started** сообщает, что ваш бот был запущен пользователем.

Используется в ограниченной интеграции. Должен передать только параметр, с которым бота стартанули.

Если конструктор, на котором сделан ваш бот не позволяет получить параметр start, с которым ваш бот был запущен, можно использовать ключевое слово “auto_detect”. В этом случае сервис Откуда Подписки будет стараться определить

Название	Тип	Описание
start_value	string	Значение параметра “start” при старте бота. Если значения нет, то пустая строка.

Пример:

```
{
"start_value":"PJaf5568712cd"
}
```

my_bot_was_stopped

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/my_bot_was_stopped

Макс: не применимо

Используется в полной интеграции

Метод **bot_was_stopped** сообщает, что ваш бот был заблокирован пользователем.

Название	Тип	Описание
user_id	string	ID пользователя, который заблокировал вашего бота
date	int	(optional) Дата события в формате unixtime. Если не задана, то применяется текущая

Пример запроса:

```
{
  "user_id":"123456789"
}
```

my_bothelp_was_started

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/my_bothelp_was_started

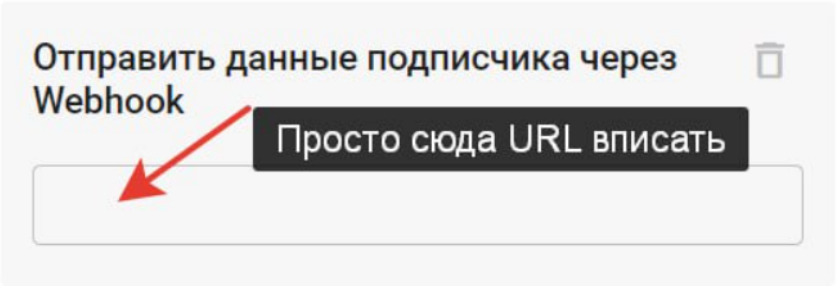
Макс: не применимо

Используется в интеграции бота, сделанного на ботхелпе

1. В самом боте после пункта старт, создайте действие.
2. Выберите там пункт "Отправить данные подписчика через Webhook" на url

https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/my_bothelp_was_started

---



Это будет работать, **только на ботах, сделанных на ботхелпе**. Просто выберите пункт о пересылке данных пользователя и отошлите его на наш url методом POST

win_win_was_started

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/win_win_was_started

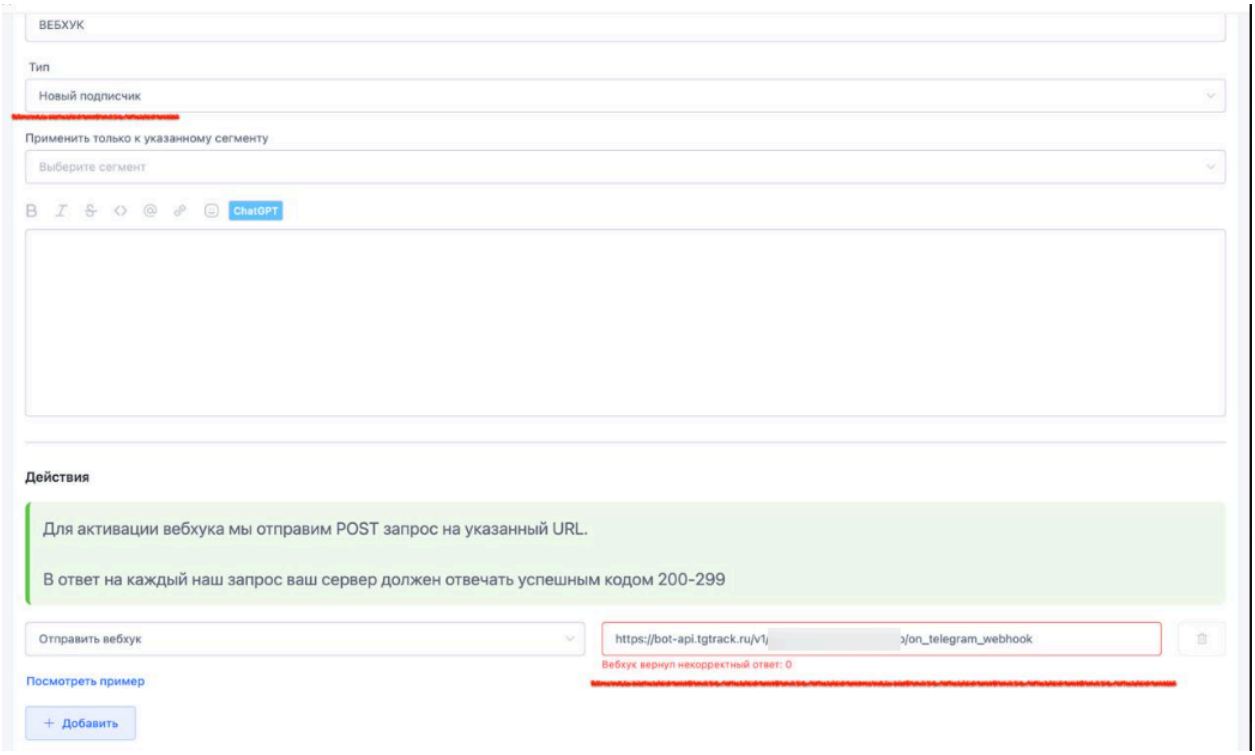
Макс: не применимо

Используется в интеграции бота, сделанного на win win bot

В самом боте создайте отправку данных подписчика при типе событий “новый подписчик”

Выберите там пункт "Отправить вебхук" на url

https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/win_win_was_started



Это будет работать, **только на ботах, сделанных на win win bot**. Просто выберите пункт о пересылке данных пользователя и отошлите его на наш url методом POST

sambot_on_tg_webhook

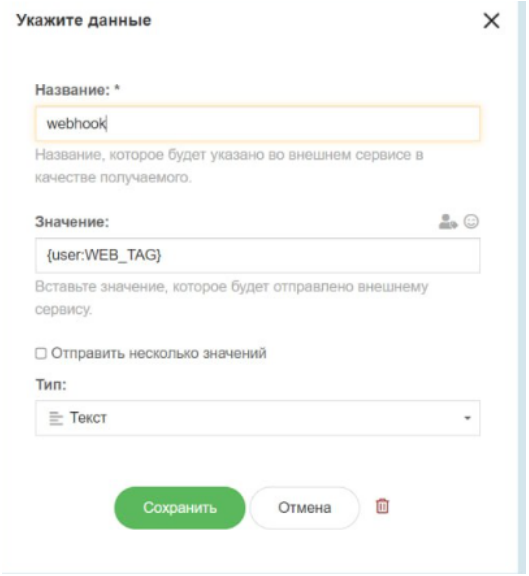
Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/sambot_on_tg_webhook

Макс: не применимо

Используйте этот метод, если ваш бот собран на конструкторе SamBot.

При старте или блокировке вашего бота отправьте вебхук, полученный от телеграм в поле “webhook” на этот метод.

Вот так выглядит настройка отправки в sambot:



Что нужно передать

Название	Тип	Описание
webhook	Json	webhook, полученный от телеграм

Пример:

```
{
  "webhook": {<полный вебхук, полученный от телеграм>}
}
```

---

on_toolsy_event

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/on_toolsy_event

Макс: не применимо

Используйте, если ваш бот подключен к сервису toolsy (прием платежей).

Кроме старта бота, в качестве глубоких целей, будут передаваться события активации подписки, обновление подписки, создание платежа и создание возврата.

В самом боте настройте отправку событий на этот адрес.

https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/on_toolsy_event

on_send_pulse_event

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/on_send_pulse_event

Макс: не применимо

Используйте, если ваш бот подключен к сервису SendPulse

Метод обрабатывает события старта и стопа бота

В самом боте настройте отправку событий на этот адрес.

https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/on_send_pulse_event

user_did_start_bot
user_did_subscribe (синоним)

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/user_did_start_bot

Макс: https://max.tgtrack.ru/API/bot-api/v1/API_KEY/user_did_start_bot

Используйте этот метод, если ваш бот собран на конструкторе, который не позволяет получить вебхук от телеграма, но позволяет получить данные пользователя, запустившего бота

При старте или блокировке вашего бота отправьте вебхук, полученный от телеграм в поле “webhook” на этот метод.

Что нужно передать

Название	Тип	Описание
user_id	string	ID пользователя в Телеграм, с которым связано событие
first_name	string	Имя пользователя
last_name	string	(optional). Фамилия, если задана
username	string	(optional) Username в телеграм, если известен
full_name	string	(optional) полное имя пользователя телеграм
start_value	string	(optional) параметр start, с которым запустили бота, если возможно получит

Пример:


```
{
  "user_id":"987654321",
  "first_name": "John",
  "last_name":  "Doe",
  "username":  "johnDoe5672",
  "start_value":  "TGTrack-PJ123456"
}
```

Методы API - глубокие цели

send_reach_goal

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/send_reach_goal

Макс: https://max.tgtrack.ru/API/bot-api/v1/API_KEY/send_reach_goal

Метод **send_reach_goal** пересылает достижение цели в рекламную систему, откуда пришел пользователь.

Например, в вашей воронке есть шаг получения номера телефона у пользователя. И вы хотите передавать в рекламную систему Яндекс/ФБ/ВК/Гугл тех, кто оставил номер телефона, чтобы реклама обучалась на эти события.

В этом случае метод **send_reach_goal** пробросит достижение цели в ту рекламную платформу, откуда пришел пользователь.

Цель можно передавать в течение 21 дня с момента подписки пользователя на канал или старта вашего бота.

Название	Тип	Описание
user_id	string	ID пользователя в телеграм, с которым связано событие
target	string	Идентификатор события в рекламной системе. Достижение этой цели будет передано в Яндекс/ФБ/ВК/Гугл

Пример:

```
{
  "user_id":"987654321",
  "target": "userDidSharePhone"
}
```

add_event

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/add_event

Макс: https://max.tgtrack.ru/API/bot-api/v1/API_KEY/add_event

Используется в полной интеграции

Метод **add_event** записывает новое событие в канал или бота. Используется для передачи событий жизненного цикла подписчика. Например: прохождение этапов воронки, продажа, квалификация лида и т.п.

Название	Тип	Описание
user_id	string	ID пользователя в телеграм, с которым связано событие

Название	Тип	Описание
event_type	string	Тип события. Показывается в отчетах как шаг в воронке и считается кол-во событий.
event_result	string	(optional). Одно из 3: in_progress/success/failure. Результат шага по отношению к целям лидогенерации. in_progress – (по умолчанию) лид продолжает двигаться по воронке success – достигнута цель воронки (продажа, запись на вебинар, заявка и т.п.) failure – цель не достигнута, лид вышел из воронки и больше с ним не работаем Если не задано, то применяется in_progress
date	int	(optional) Дата события в формате unixtime. Если не задана, то применяется текущая
conversion_target	string	(optional) Если задано, то эта цель будет передана в Яндекс Директ/ФБ/ВК/Гугл, если пользователь пришел из соответствующей рекламной системы. Работает только 21 день после подписки/старта вашего бота.
amount	float	(optional) Сумма. Положительная для выручки, отрицательная для затрат. Позволяет задавать стоимостное выражение достижения цели. Например, выручку по продажам.
labels	Label[]	(optional) Массив дополнительных аналитических меток для события

Пример:

```
{
  "user_id":"987654321",
  "event_type": "sale",
  "date": 1622520000,
  "amount": 1500.00,
  "conversion_target": "lead",
  "labels": [
    {
      "name": "manager",
      "value": "Jane Doe"
    },
    {
      "name": "product",
      "value": "Smartphone"
    }
  ]
}
```

Методы API - получение данных

get_user_info

Телеграм: https://bot-api.tgtrack.ru/v1/API_КЛЮЧ/get_user_info

Макс: https://max.tgtrack.ru/API/bot-api/v1/API_KEY/get_user_info

Используется для получения данных пользователя по user id телеграма.

Метод **get_user_info** получает на вход user id пользователя, возвращает в ответе его данные. Используете метод POST для передачи данных в запросе.

Параметры запроса:

Название	Тип	Описание
user_id	string	user id пользователя телеграма.

Пример:

```
{
  "user_id": "123456789"
}
```

Возвращаемый ответ:

Название	Тип	Описание
user_id	string	ID пользователя в телеграм, с которым связано событие
first_join_date	int	дата последней подписки пользователя в формате unix time. (если нет то 0)
last_join_date	int	дата последней подписки пользователя в формате unix time. (если нет то 0)
invite_link	string	Источник, откуда пользователь подписался в первый раз
first_name	string	имя пользователя в телеграме.
username	string	юзернейм пользователя в телеграме (если нет то пустая строка).
left_date	int	дата отписки (если нет то 0)
utm_source	string	(optional) UTM метка source.
utm_medium	string	(optional) UTM метка medium.
utm_content	string	(optional) UTM метка content.
utm_campaign	string	(optional) UTM метка campaign.
utm_term	string	(optional) UTM метка term.

Пример:

```
{
   "status": "OK",
   "data": {
      "user_id": "123456789",
      "first_join_date": 1680387078,
      "last_join_date": 1680387078,
```

```
    "invite_link": "Яндекс.Директ",
    "left_date": 0,
    "first_name": "Вася",
    "last_name": "",
    "username": "vasya",
    "utm_source": "test_source",
    "utm_medium": "test_medium",
    "utm_campaign": "test_campaign"
  }
}
```


get_user_last_transition

Телегам: не применимо

max: https://max.tgtrack.ru/API/bot-api/v1/API_KEY/get_user_last_transition

Используется для получения последнего перехода пользователя в канал за последние 7 дней.

Метод get_user_last_transition получает на вход user_id пользователя и возвращает информацию о его последнем переходе, а также текущий статус подписки в канале.

Для передачи данных используйте метод POST.

Параметры запроса:

Название	Тип	Описание
user_id	string	user id пользователя телеграма / макс.

Пример:


```
{
  "user_id": "123456789"
}
```


Возвращаемый ответ:

Название	Тип	Описание
transition_date	int	Дата последнего перехода в канал в формате unix time
subscribed_by_transition	string	Признак подписки именно по этому переходу: yes / no
status	string	Текущий статус пользователя в канале: member / not_member
source	string	Название источника (название ссылки, по которой был переход)

Название	Тип	Описание
source_type	string	Тип источника/ссылки
click_id	string	click_id перехода (если нет, то пустая строка), для Яндекс - yandex client id
utm_source	string	(optional) UTM метка source
utm_medium	string	(optional) UTM метка medium
utm_content	string	(optional) UTM метка content
utm_campaign	string	(optional) UTM метка campaign
utm_term	string	(optional) UTM метка term

Пример:


```
{
 "status": "OK",
 "data": {
  "transition_date": 1716991000,
  "subscribed_by_transition": "yes",
  "status": "member",
  "source": "Яндекс.Директ",
  "source_type": "tracking_link",
  "click_id": "abc123xyz",
  "utm_source": "test_source",
  "utm_medium": "test_medium",
  "utm_content": "test_content",
  "utm_campaign": "test_campaign",
  "utm_term": "test_term"
 }
}
```


Если за последние 7 дней переходов не найдено, метод возвращает ошибку:

No transition found in the last 7 days.

get_token_info

Телегам: не применимо
max: https://max.tgtrack.ru/API/bot-api/v1/API_KEY/get_token_info

Используется для получения информации о текущем API-токене и чате, к которому он привязан.

Метод get_token_info возвращает дату создания токена, его актуальность, ID чата, название чата и текущую ссылку чата (если установлена).

Для вызова используйте метод POST (тело можно не передавать).

Параметры запроса:

Нет обязательных параметров.

Пример:

{}

Возвращаемый ответ:

Название	Тип	Описание
token_date	int	Дата создания токена в формате unix time
is_valid	bool	Признак валидности токена (true / false)
chat_id	int	ID чата в MAX
chat_title	string	Название чата
url	string	Текущая ссылка на чат (если не задана — пустая строка)

Пример:


```
{
 "status": "OK",
 "data": {
  "token_date": 1716991000,
  "is_valid": true,
  "chat_id": 1234567890,
  "chat_title": "Test Chat",
  "url": "https://max.ru/my_chat"
 }
}
```

Методы API - управление

set_default_chat_link

Телегам: не применимо
max: https://max.tgtrack.ru/API/bot-api/v1/API_KEY/set_default_chat_link

Используется для установки ссылки чата по умолчанию.

Метод set_default_chat_link принимает ссылку на чат MAX, извлекает из нее username и сохраняет его как дефолтную ссылку чата.

Для передачи данных используйте метод POST.

Параметры запроса:

Название	Тип	Описание
url	string	Ссылка на чат MAX

Пример:

```
{
  "url": "https://max.ru/my_chat"
}
```

Возвращаемый ответ:

При успешной установке возвращается стандартный успешный ответ.

Пример:

```
{
  "status": "OK"
}
```

Возможные ошибки:

- url must be specified - не задана новая ссылка
- invalid chat link, url must be a valid max link - ссылка не является корректной ссылкой на ресурс в максе
- chat username not found in link - не удалось распарсить ссылку и выделить куда она ведет
- chat not found - чат был удален из сервиса

Список ошибок

Таблица возможных ошибок и их расшифровок

Код ошибки	Короткое описание (англ)	Полное описание (рус)
400	Bad request	Запрос содержит синтаксические ошибки или недостаточные данные.
401	Invalid token	Токен аутентификации не верен или отсутствует. Доступ к ресурсу запрещен.
402	Token expired	Токен аутентификации истек. Необходимо повторно аутентифицироваться.
403	Forbidden	У вас нет достаточных прав для выполнения этого действия.
404	Method not found	Запрашиваемый метод не найден. Проверьте URL и метод запроса.
405	Resource not found	Запрашиваемый ресурс не существует.
409	Data conflict	Произошел конфликт данных, например, попытка создать дублирующую запись.
415	Unsupported media type	Запрашиваемый тип данных не поддерживается сервером.
422	Unprocessable entity	Данные запроса не прошли валидацию. Проверьте корректность передаваемых данных.
429	Too many requests	Превышен лимит запросов. Попробуйте повторить запрос позже.
500	Internal server error	Произошла внутренняя ошибка сервера. Повторите попытку позже.
503	Service unavailable	Сервис временно недоступен. Попробуйте повторить запрос позже.

Лимиты на использование API

При использовании API бота применяются лимиты, при превышении которых API возвращает ошибку 429 Too many requests.

По умолчанию действуют следующие лимиты:

- 3 запроса в секунду
- 500 запросов в час
- 5000 запросов в день

Счетчик использования сбрасывается каждую секунду, час или день соответственно.

Если для вашего проекта требуется больше запросов, напишите нам в поддержку @varivan.

Как подключать своих ботов

Все современные конструкторы ботов поддерживают интеграцию с другими системами по API.

Для подключения своего бота добавьте шаги вызова стороннего API (webhook) при следующих событиях в вашем боте и настройте вызов методов нашего API:

1. старт бота пользователем -> my_bot_was_started
2. (опционально) блокировка бота -> my_bot_was_stopped
3. (опционально) промежуточные важные события прохождения по воронке -> add_event

SaleBot

Документация по настройке webhook:

https://docs.salebot.pro/api-v-konstruktore-salebot.pro/integraciya-bota-so-storonnimi-api

Базовый алгоритм:

1. установите значение переменной save_webhook в значение 1
2. перешлите содержимое переменной tg_request в метод on_telegram_webhook, когда:
    a. пользователь запускает вашего бота
    b. пользователь блокирует вашего бота
3. Убедитесь, что бот пересылает событие старта как при старте с параметром ?start=...., так и без него
4. Тут подробное видео, как настроить интеграцию с salebot: https://www.youtube.com/watch?v=RcB0YyrciT4