

CalendarBot API — Документация

Умный календарь через Telegram-бот

Хотя бот работает через Telegram, его логику можно представить как REST API. Все операции выполняются от имени пользователя (`telegram_id`).

Примечание: В реальности запросы поступают через Telegram, но эта документация помогает понять внутреннюю структуру.

1. POST /events — Создание события

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
<code>event_title</code>	string	body	Название события	Обязательно
<code>event_description</code>	string	body	Описание события	Необязательно
<code>start_datetime</code>	string	body	Дата и время начала (YYYY-MM-DD HH:MM)	Обязательно
<code>end_datetime</code>	string	body	Дата и время окончания	Необязательно
<code>event_place</code>	string	body	Место проведения	Необязательно
<code>telegram_id</code>	string	header	ID пользователя в Telegram	Обязательно

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность
<code>id</code>	string (UUID)	body	Уникальный ID события	Обязательно
<code>event_title</code>	string	body	Название события	Обязательно
<code>event_description</code>	string	body	Описание	Необязательно
<code>start_datetime</code>	string	body	Дата и время начала	Обязательно
<code>end_datetime</code>	string	body	Дата и время окончания	Необязательно
<code>event_place</code>	string	body	Место проведения	Необязательно
<code>created_at</code>	string (ISO)	body	Время создания	Обязательно

Пример запроса

```
POST /events HTTP/1.1
Host: api.eventbot.example
Content-Type: application/json
telegram_id: 7894615329

{
  "event_title": "Встреча с командой",
  "event_description": "Обсудим релиз",
  "start_datetime": "2025-09-14T18:00:00",
  "end_datetime": "2025-09-14T19:30:00",
  "event_place": "Zoom"
}
```

Пример успешного ответа

```
HTTP/1.1 201 Created

{
  "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "event_title": "Встреча с командой",
  "event_description": "Обсудим релиз",
  "start_datetime": "2025-09-14T18:00:00",
  "end_datetime": "2025-09-14T19:30:00",
  "event_place": "Zoom",
  "created_at": "2025-09-13T15:30:00Z"
}
```

2. GET /events — Получение событий по периоду

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
start_date	string	query	Начальная дата (YYYY-MM-DD)	Необязательно
end_date	string	query	Конечная дата (YYYY-MM-DD)	Необязательно
start_time	string	query	Время начала (HH:MM)	Необязательно
end_time	string	query	Время окончания (HH:MM)	Необязательно
telegram_id	string	header	ID пользователя	Обязательно

Response

Массив объектов событий (структура как в POST /events)

Пример запроса

```
GET /events?start_date=2025-09-14&end_date=2025-09-14 HTTP/1.1
Host: api.eventbot.example
telegram_id: 7894615329
```

Пример успешного ответа

HTTP/1.1 200 OK

```
[
  {
    "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
    "event_title": "Встреча с командой",
    "event_description": "Обсудим релиз",
    "start_datetime": "2025-09-14T18:00:00",
    "end_datetime": "2025-09-14T19:30:00",
    "event_place": "Zoom",
    "created_at": "2025-09-13T15:30:00Z"
  }
]
```

3. PUT /events/{id} — Изменение события

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
event_title	string	body	Новое название	Необязательно
event_description	string	body	Новое описание	Необязательно
start_datetime	string	body	Новая дата-время начала	Необязательно
end_datetime	string	body	Новая дата-время окончания	Необязательно
event_place	string	body	Новое место	Необязательно
telegram_id	string	header	ID пользователя	Обязательно

Response

Структура как в POST , но с полем updated_at .

Пример запроса

```
PUT /events/a1b2c3d4-e5f6-7890-abcd-ef1234567890 HTTP/1.1
Host: api.eventbot.example
Content-Type: application/json
telegram_id: 7894615329

{
  "start_datetime": "2025-09-14T19:00:00",
  "event_place": "Google Meet"
}
```

Пример успешного ответа

HTTP/1.1 200 OK

```
{
  "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "event_title": "Встреча с командой",
  "event_description": "Обсудим релиз",
  "start_datetime": "2025-09-14T19:00:00",
  "end_datetime": "2025-09-14T19:30:00",
  "event_place": "Google Meet",
  "created_at": "2025-09-13T15:30:00Z",
  "updated_at": "2025-09-14T10:00:00Z"
}
```

4. DELETE /events/{id} — Удаление события

Request

Название параметра	Тип данных	Находится в	Описание	Обязательность
telegram_id	string	header	ID пользователя	Обязательно

Response

Название параметра	Тип данных	Находится в	Описание	Обязательность
id	string	body	ID удалённого события	Обязательно
deleted_at	string (ISO)	body	Время удаления	Обязательно

Пример запроса

```
DELETE /events/a1b2c3d4-e5f6-7890-abcd-ef1234567890 HTTP/1.1
Host: api.eventbot.example
telegram_id: 7894615329
```

Пример успешного ответа

```
HTTP/1.1 200 OK

{
  "id": "a1b2c3d4-e5f6-7890-abcd-ef1234567890",
  "deleted_at": "2025-09-14T10:15:00Z"
}
```

⚠ Коды ошибок

Код	Сообщение	Описание
400 Bad Request	Invalid input	Неверный формат даты, времени или отсутствуют обязательные поля

401 Unauthorized	Missing or invalid telegram_id	Не передан telegram_id или он не существует
404 Not Found	Event not found	Событие с таким ID не найдено
500 Internal Server Error	Server error	Ошибка на стороне Supabase или бэкенда

Пример ошибки

```
HTTP/1.1 400 Bad Request

{
  "error": "Invalid datetime format",
  "details": "Field 'start_datetime' must be in format YYYY-MM-DDTHH:MM:SS"
}
```