

#### **Developer Notes: Event Scheduling API**

##### **1. Authentication & Token Handling**

- The system provides a Register API and a Login API to authenticate users.

###### **- Register API:**

- Requires: username, password, email
- Response: access token and refresh token

###### **- Login API:**

- Requires: username, password
- Response: access token and refresh token (same format as register)

###### **- Usage:**

- For all authenticated requests, add the following header:

Authorization: Bearer <access\_token>

##### **2. Bulk Event Creation (No Conflict Detection)**

- Bulk creation uses Django's bulk\_create() for efficiency.
- All event objects are prepared in memory and inserted in one DB call.
- This avoids multiple DB connections in a loop.

###### **- Trade-off:**

- Conflict detection is not available for bulk creation
- Since objects are not saved individually, they can't be validated one by one

##### **3. Optional: Extend Access Token Lifetime**

- You can increase the access token lifetime in `settings.py` to avoid frequent refreshes:

###### **Example:**

```
from datetime import timedelta
```

```
SIMPLE_JWT = {
    "ACCESS_TOKEN_LIFETIME": timedelta(hours=1)
}
```

##### **4. Event History & Versioning**

- The app uses django-simple-history to keep track of event versions.
- Each time an event is updated, a new historical version is stored.
- Key difference between id and history\_id:

Field	Description
-------	-------------

id ----- -- ----- - The original primary key of the event (remains constant)

history\_id ----- ----- ----- A globally unique ID that increases with each historical change

- You can retrieve version history, compare diffs, and rollback to previous versions using history\_id.

## **API Endpoint: POST /api/auth/register**

Summary

Registers a new user in the system.

URL

**POST /api/auth/register**

Request Headers

Content-Type: application/json

Request Body

```
{  
  "username": "ateebtest",  
  "email": "hamsad@yopamail.com",  
  "password": "Tesst123"  
}
```

Response

```
{  
  "refresh":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYmcmVzaC1sImV4cCI6MTc0ODY5NTkzNSviaWF0IjoxNzQ4MDkxMTM1LCJqdGkiOiI2YWY1MWM0ODdiNzI0YjAzOTE4ZDk5MzY2OWRiMzJhMSIsInVzZXJfaWQiOjEyfQ.I2y2O_-V6AJlZXJwnhLnOR6IVfOIG-1zPZzTEa9LwRk",  
  "access":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzQ4MDk0NzM1LCJpYXQiOjE3NDgwOTEzMzUsImp0aSI6ImYxODFiNmE0YTBjMDQ5Zjc5NTQ4M2E0NmE5YjZkMWMzIiwidXNlcI9pZCI6MTJ9.jI3XMMSiZA16-YLjx-muTG07dvnlaHr28U70JXj3ns",  
  "user": {  
    "id": 12,  
    "username": "ateebtest"  
  }  
}
```

## Example cURL

```
curl --location 'http://127.0.0.1:8000/api/auth/register/' \
--header 'Content-Type: application/json' \
--data-raw '{
    "username": "ateebtest",
    "email": "hamsad@yopamail.com",
    "password": "Tesst123"
}''
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections like Collections, Environments, Flows, History, and a plus sign for creating new items. The main area shows a collection named 'event\_management'. Under 'event management', there's a 'auth' folder containing a POST request to 'http://127.0.0.1:8000/api/auth/register/'. The request has a status of '201 Created', a response time of '284 ms', and a size of '842 B'. The response body is a JSON object:

```
1 {
2     "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
3         eyJ0b2tlbl90eXB1IjoicmVmcmVzaC1sImV4cCI6Mtc0ODY5NTkzNSwiaWF0IjoxNzQ4MDkxMTM1LCJqdGkiOiI2YWY1MnM0ODdiNzI0Yj
4             AzOTE4ZDk5MzY20WR1MzJhMSIsInVzZXJfaWQidjEyf0.IIy20_-V6AJ1ZXJwnhLnORsIVf0IG-1zPz2Tea9LwRk",
5     "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
6         eyJ0b2tlbl90eXB1IjoicmVmcmVzaC1sImV4cCI6IkpXVCJ9.
7             c5NTQ4M2E0NmE5YjZkMWMzIiwidXNlcl9pZCI6MT39.j13XMMSiZA16-YLjx-muTg97dvn1ajHr28U70JXj3ns",
8     "user": {
9         "id": 12,
10        "username": "ateebtest"
11    }
12}
```

At the bottom, there are buttons for Postbot, Runner, Start Proxy, Cookies, Vault, and Trash.

# API Endpoint: POST /api/auth/login

## Summary

Authenticates a user and returns a JWT bearer token.

## URL

`POST /api/auth/login`

## Request Headers

Content-Type application/json

## Request Body

```
{  
  "username": "testuser",  
  "password": "TestPass123"  
}
```

## Response

```
{  
  "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBIIjoicmVmcnVzaCIsImV4cCI6MTc0ODY5NDk3NiwiaWF0IjoxNzQ4MDkwMTc2LCJqdGkiOiI5MzI1MWViYzI0ZWE0MTc1YjU1NDM0ZWYyNDI5YTZhMyIsInVzZXJfaWQiOjF9.kOWPTGgYzqItm7pRHrN84QO5iN8pFFH-tyAdRJ_f9FQ",  
  "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBIIjoiYWNjZXNzIwiZXhwIjoxNzQ4MDkzNzc2LCJpYXQiOjE3NDgwOTAxNzYsImp0aSI6IjZmY2YxOWM3OWE5YzRmMThhNjZiM2NiMzRhNjdhNTZkIiwidXNlcI9pZCI6MX0.uN12lMnIT9kKA  
KoAfnReHysduaVaKbmJQlaaY1ajDWM",  
  "user": {  
    "id": 1,  
    "username": "testuser"  
  }  
}
```

## Example cURL

```
curl --location 'http://127.0.0.1:8000/api/auth/login/' \  
--header 'Content-Type: application/json' \  
--data '{  
  "username": "testuser",  
  "password": "TestPass123"  
}'
```

Home Workspaces API Network

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan.](#)

**event\_management**

New Import

auth / http://127.0.0.1:8000/api/auth/login/

POST http://127.0.0.1:8000/api/auth/login/

Params Authorization Headers (10) Body Scripts Settings

Body (raw JSON)

```

1 {
2   "username": "testuser",
3   "password": "TestPass123"
4 }
```

Send Save Share

Auth / http://127.0.0.1:8000/api/auth/login/

200 OK 262 ms 832 B Save Response

Body Cookies Headers (10) Test Results

{ JSON Preview Visualize }

```

1 {
2   ...
3     "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
4       eyJ0b2tlb190eXB1IjoiemVmcmVzaC1sImV4cCI6MTc0ODY5NDk3NiwiaWF0IjoxNzQ4MDkwMTC2LCJqdGkiOiI5MzI1MwViYzI0ZWE0MT
5       c1YjU1NDM0ZWYyNDI5YTZhMyIsInVzZXJfaWQiOjF9.k0WPTGgYzqItm7pRHRN84Q05iN8pFFH-tyAdRJ_f9FQ",
6     "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
7       eyJ0b2tlb190eXB1IjoiYWNjZXNzIiwiZXhwIjoxNzQ4MDkzNzc2LCJpYXQiOjE3NDgwOTAxNzYsImp0aSI6IjZmY2Yx0WM30WE5YzRmMT
8       hhnJzIM2NiMzRhNjdhNTZkIiwidXNlc1pZCI6MX0.uN12lMnIT9kKAKoAfneHysduaVaKbmJQlaaY1ajDW",
9     "user": {
10       ...
11       "id": 1,
12       ...
13       "username": "testuser"
14     }
15 }
```

Postbot Runner Start Proxy Cookies Vault Trash

## **API Endpoint: POST /api/auth/logout**

### Summary

Logs out a user by blacklisting their refresh token.

### URL

**POST /api/auth/logout**

### Request Headers

Content-Type: application/json

### Request Body

```
{  
  "refresh":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaC1sImV4cCI6MTc0ODY5NTkzNSwiaWF0IjoxNzQ4MDkxMTM1LCJqdGkiOiI2YWY1MWM0ODdiNzI0YjAzOTE4ZDk5MzY2OWRiMzJhMSIsInVzZXJfaWQiOjEyfQ.I2y2O_-V6AJlZXJwnhLnOR6IVfOIG-1zPZzTEa9LwRk"  
}
```

### Response

- Status 205 Reset Content

### **Example cURL**

```
curl --location 'http:/  
/127.0.0.1:8000/api/auth/logout/' \  
--header 'Content-Type: application/json' \  
--data '{  
  "refresh":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaC1sImV4cCI6MTc0ODY5NTkzNSwiaWF0IjoxNzQ4MDkxMTM1LCJqdGkiOiI2YWY1MWM0ODdiNzI0YjAzOTE4ZDk5MzY2OWRiMzJhMSIsInVzZXJfaWQiOjEyfQ.I2y2O_-V6AJlZXJwnhLnOR6IVfOIG-1zPZzTEa9LwRk"  
}'
```

Home Workspaces API Network

Search Postman

Upgrade

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan.](#)

event\_management

New Import Overview POST http POST http POST http [CONFLICT] POST http GET http PUT http > + No environment

Collections

auth

- POST http://127.0.0.1:8000/api/auth/register/
- POST http://127.0.0.1:8000/api/auth/login/
- POST http://127.0.0.1:8000/api/auth/logout/
- POST http://127.0.0.1:8000/api/auth/refresh/

changelog/diffcheck

- GET http://localhost:8000/api/events/3/changelog/
- GET http://localhost:8000/api/events/3/diff/5/7/

event crud and bulk create

- POST http://localhost:8000/api/events/
- GET http://localhost:8000/api/events/?page=1&page\_...
- POST http://localhost:8000/api/events/batch/

event crud wrt id

- PUT http://localhost:8000/api/events/5/
- GET http://localhost:8000/api/events/4/
- DEL http://localhost:8000/api/events/2/

history/rollback event

- GET http://localhost:8000/api/events/8/history/1/
- GET http://localhost:8000/api/events/3/history/
- POST http://localhost:8000/api/events/3/rollback/5/

share event/permissions

- DEL http://localhost:8000/api/events/3/permissions/2/
- POST http://localhost:8000/api/events/3/share/
- GET http://localhost:8000/api/events/3/permissions/
- PUT http://localhost:8000/api/events/3/permissions/3/

auth / http://127.0.0.1:8000/api/auth/logout/

POST http://127.0.0.1:8000/api/auth/logout/

Params Authorization Headers (10) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON

Beautify

```
1 {  
2   ... "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
eyJ0b2tlb190eXBlIjoicmVmcmVzaC1sImV4cCI6MTc0ODY5NTkzNSwiaWF0IjoxNzQ4MDkxMTM1LCJqdGkiOiI2YWY1MW00d...  
OTE4ZDk5MzY20WR1MzJhMSIsInVzZXJfaWQiOjEyfQ.I2y20_-VeAJlZXJwnhLnOR6IVf0IG-1zPZzTEa9LwRk"  
3 }
```

Body Cookies Headers (9) Test Results

205 Reset Content 6.08 s 288 B Save Response

Raw Preview Visualize

Postbot Runner Start Proxy Cookies Vault Trash

**API Endpoint: POST /api/auth/refresh**

**Summary**

Generates a new access token using the provided refresh token.

**URL**

**POST /api/auth/refresh**

**Request Headers**

Content-Type: application/json

**Request Body**

```
{  
  "refresh": "your_refresh_token_here"  
}  
  
{  
  
  "refresh":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaC1sImV4cCI6MTc0ODA3MDQ1NCwiaWF0IjoxNz  
    Q3OTg0MDU0LCJqdGkiOiJmYzYwZTljOTRkNzk0MTNhYmEzYWE4NGM3NTlhMTQ5ZSIslnVzZXJfaWQiOjF9.wMYR1ne  
    mKDsi4S36XsRQ-nWD_6DvaG0hB2nRfcpscaU"  
}
```

**Response**

```
{  
  "access": "new_access_token_here"  
}  
  
{  
  
  "access":  
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoiYWNjZXNzIiwiZXhwIjoxNzQ4MDk4NTk0LCJpYXQiOjE3ND  
    gwOTQ5NzYsImp0aSI6ImY3MGEzMjA4MTA1MjQ1OGZhOTY0NmY4NzgxZjI1MDk3IiwidXNlcl9pZCI6MX0.TVR4x0iCdk9  
    Wt0kBGTGTwG8tKcmITCqwL8K8ZIKc8x1Y"  
}
```

## Example cURL

```
curl --location 'http://127.0.0.1:8000/api/auth/refresh/' \
--header 'Content-Type: application/json' \
--data '{
    "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXB1IjoicmVmcmVzaC1sImV4cCI6MTc0ODY5OTc3NiwiawF0IjoxNzQ4MDk0OTc2LCJqdGkiOiI0Mj1mZjZhNTVmNTg0MjNjOGVkMDAzOD1lZmQzZTQ1ZiIsInVzZXJfaWQiOjF9.LuEDdw1Lftoq_SQf8KoWvDgYvHu55WksYPMylWn9MT0"
}'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' (selected), 'Environments', 'Flows', 'History', and 'APIs'. The main area displays a collection named 'event\_management'. Under this collection, there are several categories: 'auth', 'changelog/diffcheck', 'event crud and bulk create', 'event crud wrt id', 'history/rollback event', 'share event/permissions', and 'other'. A specific POST request is selected under the 'auth' category, with the URL 'http://127.0.0.1:8000/api/auth/refresh/'. The 'Body' tab is active, showing a JSON payload:

```
1 {
2     "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ0b2tlbl90eXB1IjoicmVmcmVzaC1sImV4cCI6MTc0ODY5OTc3NiwiawF0IjoxNzQ4MDk0OTc2LCJqdGkiOiI0Mj1mZjZhNTVmNTg0MjNjOGVkMDAzOD1lZmQzZTQ1ZiIsInVzZXJfaWQiOjF9.LuEDdw1Lftoq_SQf8KoWvDgYvHu55WksYPMylWn9MT0"
3 }
```

Below the body, the response status is shown as '200 OK' with a duration of '19 ms' and a size of '552 B'. The response body is also displayed in JSON format:

```
1 {
2     "access": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
eyJ0b2tlbl90eXB1IjoicmVmcmVzaC1sImV4cCI6MTc0ODY5OTc3NiwiawF0IjoxNzQ4MDk0OTc2LCJqdGkiOiI0Mj1mZjZhNTVmNTg0MjNjOGVkMDAzOD1lZmQzZTQ1ZiIsInVzZXJfaWQiOjF9.LuEDdw1Lftoq_SQf8KoWvDgYvHu55WksYPMylWn9MT0"
3 }
```

At the bottom of the interface, there are various navigation and utility buttons like 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a help icon.

**API Endpoint: GET /api/events/**

## Summary

Retrieve a list of events. Supports pagination.

---->URL

**GET /api/events/**

## Query Parameters

- page (optional): Page number (default: 1)
- page\_size (optional): Number of items per page (default: 10)

## Request Headers

Accept: application/json

Response (200 OK)

{

```
"results": [
{
  "id": 3,
  "title": "testing history",
  "description": "history description",
  "start_time": "2025-06-01 11:23:00",
  "end_time": "2025-06-01 11:23:00",
  "location": "srinagar",
  "is_recurring": true,
  "recurrence_pattern": "weekly"
},
{
  "id": 4,
  "title": "for event 4 history lets see os",
  "description": "lwts description",
  "start_time": "2025-06-01 11:23:00",
  "end_time": "2025-06-01 11:23:00",
  "location": "srinagar",
  "is_recurring": true,
  "recurrence_pattern": "weekly"
}
],
"count": 18,
"num_pages": 2,
"current_page": 1
}
```

## Example cURL

```

curl --location 'http://localhost:8000/api/events/?page=1&page_size=10' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlb190eXB1IjoiYWNjZXNzIiwidXhwIjoxNzQ
4MDkzNzc2LCJpYXQiOjE3NDgwOTAxNzYsImp0aSI6IjZmY2YxOWM3OWE5YzRmMThhNjZiM2NiMzRhNjd
hNTZkIiwidXNlc19pZCI6MX0.uN121MnIT9kKAKoAfnReHysduaVaKbmJQlaaY1ajDWM' \
--header 'Content-Type: application/json' \
--data ''

```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'event\_management' selected. Under 'event crud and bulk create', there are two entries: a POST request to http://localhost:8000/api/events/ and a GET request to http://localhost:8000/api/events/?page=1&page\_size=10. The GET request is highlighted. The main panel shows the request details: method 'GET', URL 'http://localhost:8000/api/events/?page=1&page\_size=10', and a note 'This request does not have a body'. Below the URL, there are tabs for Body, Cookies, Headers (10), Test Results, and a preview of the response. The response body is displayed as JSON:

```

1 {
2   "results": [
3     {
4       "id": 3,
5       "title": "testing history",
6       "description": "history description",
7       "start_time": "2025-06-01 11:23:00",
8       "end_time": "2025-06-01 11:23:00",
9       "location": "srinagar",
10      "is_recurring": true,
11      "recurrence_pattern": "weekly"
12    },
13    {
14      "id": 4,
15      "title": "for event 4 history lets see os",
16      "description": "lets description",
17      "start_time": "2025-06-01 11:23:00",
18      "end_time": "2025-06-01 11:23:00",
19      "location": "srinagar",
20      "is_recurring": true,
21      "recurrence_pattern": "weekly"
22    }
]

```

Home Workspaces API Network

Search Postman

Upgrade

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan.](#)

**event\_management**

New Import

[CONFLICT • POST http: GET http: PUT http: POST http: DEL http:/] [GET http: POST http:] No environment

Collections Environments Flows History

auth

- POST http://127.0.0.1:8000/api/auth/register/
- POST http://127.0.0.1:8000/api/auth/login/
- POST http://127.0.0.1:8000/api/auth/logout/
- POST http://127.0.0.1:8000/api/auth/refresh/

changelog/diffcheck

- GET http://localhost:8000/api/events/3/changelog/
- GET http://localhost:8000/api/events/3/diff/5/7/

event crud and bulk create

- POST http://localhost:8000/api/events/
- GET http://localhost:8000/api/events/?page=1&page\_size=10
- POST http://localhost:8000/api/events/batch/

event crud wrt id

- PUT http://localhost:8000/api/events/5/
- GET http://localhost:8000/api/events/4/
- DEL http://localhost:8000/api/events/2/

history/rollback event

- GET http://localhost:8000/api/events/8/history/1/
- GET http://localhost:8000/api/events/3/history/
- POST http://localhost:8000/api/events/3/rollback/5/

share event/permissions

- DEL http://localhost:8000/api/events/3/permissions/...
- POST http://localhost:8000/api/events/3/share/
- GET http://localhost:8000/api/events/3/permissions/
- PUT http://localhost:8000/api/events/3/permissions/...

HTTP event crud and bulk create / http://localhost:8000/api/events/?page=1&page\_size=10

GET http://localhost:8000/api/events/?page=1&page\_size=10

Params Authorization Headers (9) Body Scripts Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

Body Cookies Headers (10) Test Results

200 OK 3.81 s 2.44 KB Save Response

{ } JSON Preview Visualize

```

86     "description": "aas si event 1",
87     "start_time": "2025-05-27 08:05:00",
88     "end_time": "2025-05-27 08:05:00",
89     "location": "Boarcc Room",
90     "is_recurring": false,
91     "recurrence_pattern": null
92   },
93   {
94     "id": 15,
95     "title": "ateeb is jj",
96     "description": "yayaya product design meeting",
97     "start_time": "2025-05-29 09:00:00",
98     "end_time": "2025-05-29 09:00:00",
99     "location": "Zoom",
100    "is_recurring": false,
101    "recurrence_pattern": null
102  }
103 ],
104 "count": 18,
105 "num_pages": 2,
106 "current_page": 1
107 }
```

Online Find and replace Console Postbot Runner Start Proxy Cookies Vault Trash ?

**API Endpoint: POST /api/events/**

## Summary

Creates a new event.

## URL

**POST /api/events/**

## Request Headers

Authorization: Bearer <your-access-token> (which we get from register/ and login/ api)

Content-Type: application/json

## Request Body

```
{  
  "title": "lets ssss ds",  
  "description": "aas si event 1",  
  "start_time": "2025-04-27 8:05",  
  "end_time": "2025-04-27 8:09",  
  "location": "Boarcdd Room",  
  "is_recurring": false  
}
```

## Response (201 Created)

```
{  
  "message": "Event created successfully",  
  "event": {  
    "id": 24,  
    "title": "lets ssss ds",  
    "description": "aas si event 1",  
    "start_time": "2025-04-27 08:05:00",  
    "end_time": "2025-04-27 08:09:00",  
    "location": "Boarcdd Room",  
    "is_recurring": false,  
    "recurrence_pattern": null,  
    "owner_id": 1  
  }  
}
```

## Example cURL

```
curl --location 'http://localhost:8000/api/events/' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXB1IjoiYWNjZXNzIiwiZXhwIjoxNzQ
4MTAwNjMzLCJpYXQiOjE3NDgwOTcwMzM5NTlmMzVlNjA2YTQzY2ViYWM4OTBjNzY1MGE
2Yzc1IiwidXNlc19pZCI6MX0.FMyknBs5GLeRyw_tF5tVTmVTXNbNUhkc2IFXGkCtbVY' \
--header 'Content-Type: application/json' \
--data '{
    "title": "lets ssss ds",
    "description": "aas si event 1",
    "start_time": "2025-04-27 8:05",
    "end_time": "2025-04-27 8:09",
    "location": "Boarcdd Room",
    "is_recurring": false
}'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'event\_management' selected, showing various API endpoints like auth, changelog/diffcheck, event crud and bulk create, event crud wrt id, history/rollback event, share event/permissions, and others. The main area shows a POST request to 'http://localhost:8000/api/events/'. The 'Body' tab is selected, displaying the JSON payload from the cURL example. The 'Headers' tab shows 'Content-Type: application/json'. Below the request, the response status is '201 Created' with a response time of '8.01 s' and a size of '594 B'. The response body is also shown in JSON format, indicating success and providing the event ID (24) and other details.

**API Endpoint: POST /api/events/batch/**

## Summary

Create multiple events in a single request.

## URL

POST `/api/events/batch/`

## Request Headers

Content-Type: application/json

## Request Body

```
[  
  {  
    "title": "sss is jj",  
    "description": "dd product design meeting",  
    "start_time": "2025-04-29T09:00:00Z",  
    "end_time": "2025-04-29T10:00:00Z",  
    "location": "Zoom",  
    "is_recurring": false  
  },  
  {  
    "title": "last saaas",  
    "description": "Monthly marketing planning session",  
    "start_time": "2025-04-01T11:00:00Z",  
    "end_time": "2025-04-01T12:00:00Z",  
    "location": "Conference Room A",  
    "is_recurring": true,  
    "recurrence_pattern": "monthly"  
  }  
]
```

## Response (201 Created)

```
{  
  "created_events": [  
    {  
      "id": 22,  
      "title": "sss is jj",  
      "start_time": "2025-04-29 09:00:00",  
      "end_time": "2025-04-29 10:00:00"  
    },  
    {  
      "id": 23,  
      "title": "last saaas",  
      "start_time": "2025-04-01 11:00:00",  
      "end_time": "2025-04-01 12:00:00",  
      "location": "Conference Room A",  
      "is_recurring": true,  
      "recurrence_pattern": "monthly"  
    }  
  ]  
}
```

```
"title": "last saas",
"start_time": "2025-04-01 11:00:00",
"end_time": "2025-04-01 12:00:00"
}
],
"errors": []
}
```

#### Example cURL

```
curl --location 'http://localhost:8000/api/events/batch/' \
--header 'Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXB1IjoiYWNjZXNzIiwidXhwIjoxNzQ
4MDkzNzc2LCJpYXQiOjE3NDgwOTAxNzYsImp0aSI6IjZmY2YxOWM3OWE5YzRmMThhNjZiM2NiMzRhNjd
hNTZkIiwidXNlcl9pZCI6MX0.uN12lMnIT9kKAKoAfnReHysduaVaKbmJQlaaY1ajDWM' \
--header 'Content-Type: application/json' \
--data '[

{
    "title": "sss is jj",
    "description": "dd product design meeting",
    "start_time": "2025-04-29T09:00:00Z",
    "end_time": "2025-04-29T10:00:00Z",
    "location": "Zoom",
    "is_recurring": false
},
{
    "title": "last saas",
    "description": "Monthly marketing planning session",
    "start_time": "2025-04-01T11:00:00Z",
    "end_time": "2025-04-01T12:00:00Z",
    "location": "Conference Room A",
    "is_recurring": true,
    "recurrence_pattern": "monthly"
}
]
```

] '

] '

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home' and 'Workspaces' dropdowns, a search bar, and various status icons. Below the header, a yellow banner says 'Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan](#)'.

The main area displays a collection named 'event\_management'. On the left, a sidebar lists 'Collections', 'Environments', 'Flows', and 'History'. The 'event\_management' collection contains several sub-folders and their associated API endpoints:

- auth**:
  - POST http://127.0.0.1:8000/api/auth/register/
  - POST http://127.0.0.1:8000/api/auth/login/
  - POST http://127.0.0.1:8000/api/auth/logout/
  - POST http://127.0.0.1:8000/api/auth/refresh/
- changelog/diffcheck**:
  - GET http://localhost:8000/api/events/3/changelog/
  - GET http://localhost:8000/api/events/3/diff/5/7/
- event crud and bulk create**:
  - POST http://localhost:8000/api/events/
  - GET http://localhost:8000/api/events/?page=1&page...
  - POST http://localhost:8000/api/events/batch/
- event crud wrt id**:
  - PUT http://localhost:8000/api/events/5/
  - GET http://localhost:8000/api/events/4/
  - DEL http://localhost:8000/api/events/2/
- history/rollback event**:
  - GET http://localhost:8000/api/events/8/history/1/
  - GET http://localhost:8000/api/events/3/history/
  - POST http://localhost:8000/api/events/3/rollback/5/
- share event/permissions**:
  - DEL http://localhost:8000/api/events/3/permissions/...
  - POST http://localhost:8000/api/events/3/share/
  - GET http://localhost:8000/api/events/3/permissions/...
  - PUT http://localhost:8000/api/events/3/permissions/...

In the center, a specific POST request is selected for 'event crud and bulk create' to 'http://localhost:8000/api/events/batch/'. The 'Body' tab is active, showing a JSON payload:

```
1 [  
2 {  
3   "title": "sss is jj",  
4   "description": "dd product design meeting",  
5   "start_time": "2025-04-29T09:00:00Z",  
6   "end_time": "2025-04-29T10:00:00Z",  
7   "location": "Zoom",  
8   "is_recurring": false  
9 },  
10 {  
11   "title": "last saas",  
12   "description": "Monthly marketing planning session",  
13   "start_time": "2025-04-01T11:00:00Z",  
14 }]
```

The response status is '201 Created' with a response time of 6.79 s and a size of 545 B. The 'Body' tab also shows the JSON response:

```
{ } JSON ▾ Preview ⚡ Visualize ▾  
1 {  
2   "created_events": [  
3     {  
4       "id": 22,  
5       "title": "sss is jj",  
6       "start_time": "2025-04-29 09:00:00",  
7       "end_time": "2025-04-29 10:00:00"  
8     },  
9     {  
10       "id": 23,  
11       "title": "last saas",  
12       "start_time": "2025-04-01 11:00:00",  
13       "end_time": "2025-04-01 12:00:00"  
14     }]
```

At the bottom, there are various navigation and utility buttons: 'Postbot', 'Runner', 'Start Proxy', 'Cookies', 'Vault', 'Trash', and a help icon.

API Endpoint: GET /api/events/{id} /

## Summary

Retrieves details of a specific event by ID.

## URL

GET `/api/events/<id>/`

### Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

### Response:

{

```
"id": 3,  
"title": "testing history",  
"description": "history description",  
"start_time": "2025-06-01 11:23:00",  
"end_time": "2025-06-01 18:32:00",  
"location": "srinagar",  
"is_recurring": true,  
"recurrence_pattern": "weekly"  
}
```

## Example cURL

```
curl --location 'http://localhost:8000/api/events/3/' \  
--header 'Authorization: Bearer <your-access-token>' \  
--header 'Content-Type: application/json'
```

The screenshot shows the Postman interface with the following details:

- API Endpoint:** `/api/events/3/`
- Method:** GET
- URL:** `http://localhost:8000/api/events/3/`
- Headers:** Authorization (Bearer <your-access-token>), Content-Type (application/json)
- Body:** JSON response (shown below)
- Test Results:** 200 OK, 3.19 s, 539 B
- JSON Response:**

```
1 {  
2     "id": 3,  
3     "title": "testing history",  
4     "description": "history description",  
5     "start_time": "2025-06-01 11:23:00",  
6     "end_time": "2025-06-01 18:32:00",  
7     "location": "srinagar",  
8     "is_recurring": true,  
9     "recurrence_pattern": "weekly"  
10 }
```

On the right side of the interface, there is a sidebar with the following text:

**API Endpoints:**  
PUT  
`/api/events/{id}`  
Updates an existing event

**Summary:**

URL

```
PUT /api/events/5/
```

Request Headers

```
Authorization: Bearer <your-access-token>
```

```
Content-Type: application/json
```

Request Body

```
{  
  "title": "fodibbo didi ",  
  "description": "lwts dajajjaj",  
  "start_time": "2025-06-01 20:23",  
  "end_time": "2025-06-01 21:32",  
  "location": "srinagar",  
  "is_recurring": false,  
  "recurrence_pattern": ""  
}
```

Response--

```
{  
  
  "message": "Event updated successfully",  
  "event": {  
    "id": 5,  
    "title": "fodibbo didi ",  
    "description": "lwts dajajjaj",  
    "start_time": "2025-06-01 20:23:00",  
    "end_time": "2025-06-01 21:32:00",  
    "location": "srinagar",  
    "is_recurring": false,  
    "recurrence_pattern": "",  
    "owner_id": 1  
  }  
}
```

#### Example cURL

```
curl --location --request PUT 'http://localhost:8000/api/events/5/' \  
--header 'Authorization: Bearer <your-access-token>' \
```

```
--header 'Content-Type: application/json' \
--data '{
    "title": "fodibbo didi ",
    "description": "lwts dajajjaj",
    "start_time": "2025-06-01 20:23",
    "end_time": "2025-06-01 21:32",
    "location": "srinagar",
    "is_recurring": false,
    "recurrence_pattern": ""
}'
```

The screenshot shows the Postman application interface. On the left, the sidebar displays the 'event\_management' workspace with various collections like 'auth', 'changelog/diffcheck', 'event crud and bulk create', and 'event crud wrt id'. The 'event crud wrt id' collection is currently selected. In the main area, a request is being made to the endpoint `http://localhost:8000/api/events/5/` using the `PUT` method. The request body contains the following JSON:

```
{
    "title": "fodibbo didi ",
    "description": "lwts dajajjaj",
    "start_time": "2025-06-01 20:23",
    "end_time": "2025-06-01 21:32",
    "location": "srinagar",
    "is_recurring": false,
    "recurrence_pattern": ""
}
```

Below the request, the response is shown as a `200 OK` status with a response time of 5.28 seconds and a response size of 588 B. The response body is also a JSON object:

```
{
    "message": "Event updated successfully",
    "event": {
        "id": 5,
        "title": "fodibbo didi ",
        "description": "lwts dajajjaj",
        "start_time": "2025-06-01 20:23:00",
        "end_time": "2025-06-01 21:32:00",
        "location": "srinagar",
        "is_recurring": false,
        "recurrence_pattern": "",
        "owner_id": 1
    }
}
```

## API Endpoint: DELETE /api/events/{id}/

### Summary

Deletes a specific event by ID.

## URL

**DELETE /api/events/2/**

### Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

### Response:

{

"message": "Event deleted successfully"

}

### Example cURL

```
curl --location --request DELETE 'http://localhost:8000/api/events/2/' \
--header 'Authorization: Bearer <your-access-token>' \
--header 'Content-Type: application/json'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'event\_management' collections expanded, showing various API endpoints like auth, changelog/diffcheck, event crud and bulk create, event crud wrt id, history/rollback event, share event/permissions, and others. The main workspace shows a 'DELETE' request for the URL 'http://localhost:8000/api/events/2/'. The 'Headers' tab is selected, showing two headers: 'Authorization' with the value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ...' and 'Content-Type' with the value 'application/json'. Below the request, the 'Test Results' section shows a successful response with status 200 OK, duration 3.83 s, and size 368 B. The response body is displayed as JSON: { "message": "Event deleted successfully" }.

**API Endpoint: POST /api/events/{id}/share/**

### Summary

Share an event with multiple users and assign roles.

URL

**POST /api/events/3/share/**

Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

Request Body

```
{  
  "users": [  
    {  
      "user_id": 3,  
      "role": "EDITOR"  
    },  
    {  
      "user_id": 4,  
      "role": "VIEWER"  
    }  
  ]  
}
```

Response

```
{  
  "message": "Event shared successfully",  
  "permissions": [  
    {  
      "user_id": 1,  
      "username": "testuser",  
      "role": "OWNER"  
    },  
    {  
      "user_id": 3,  
      "username": "hzhhz",  
      "role": "EDITOR"  
    },  
    {  
      "user_id": 6,  
      "username": "saoood",  
      "role": "EDITOR"  
    }  
  ]  
}
```

```
},
{
"user_id": 9,
"username": "fahad1",
"role": "VIEWER"
},
{
"user_id": 4,
"username": "fahad",
"role": "VIEWER"
}
]
```

#### Example cURL

```
curl --location 'http://localhost:8000/api/events/3/share/' \
--header 'Authorization: Bearer <your-access-token>' \
--header 'Content-Type: application/json' \
--data '{
  "users": [
    { "user_id": 3, "role": "EDITOR" },
    { "user_id": 4, "role": "VIEWER" }
  ]
}'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'event\_management' selected, containing collections like 'auth', 'changelog/diffcheck', 'event crud and bulk create', 'event crud wrt id', 'history/rollback event', and 'share event/permissions'. The main area shows a POST request to 'http://localhost:8000/api/events/3/share/' with the following JSON body:

```

1 {
2   "users": [
3     {
4       "user_id": 3,
5       "role": "EDITOR"
6     },
7     {
8       "user_id": 4,
9       "role": "VIEWER"
10    }
11  ]
12 }

```

The response status is 200 OK, and the response body is:

```

1 {
2   "message": "Event shared successfully",
3   "permissions": [
4     {
5       "user_id": 1,
6       "username": "testuser",
7       "role": "OWNER"
8     },
9     {
10      "user_id": 3,
11      "username": "hzhhz",
12      "role": "EDITOR"
13    }
14  ]
15 }

```

## API Endpoint: GET /api/events/{id}/permissions/

### Summary

Retrieve a list of users with permissions on the event.

### URL

**GET /api/events/3/permissions/**

### Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

### RESPONSE-

{

```

"participants": [
{
"user_id": 1,
"username": "testuser",

```

```

"role": "OWNER"
},
{
"user_id": 3,
"username": "hzhhz",
"role": "EDITOR"
},
]

```

### Example cURL

```

curl --location 'http://localhost:8000/api/events/3/permissions/' \
--header 'Authorization: Bearer <your-access-token>' \
--header 'Content-Type: application/json'

```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'event\_management' selected, containing various API endpoints categorized under 'auth', 'changelog/diffcheck', 'event crud and bulk create', 'event crud wrt id', 'history/rollback event', and 'share event/permissions'. The main area shows a 'share event/permissions' endpoint at `http://localhost:8000/api/events/3/permissions/`. A GET request is being prepared with the following headers:

Key	Value
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...
Key	Value

The response body is displayed as JSON:

```

1 {
2   "participants": [
3     {
4       "user_id": 1,
5       "username": "testuser",
6       "role": "OWNER"
7     }
8     {
9       "user_id": 3,
10      "username": "hzhhz",
11      "role": "EDITOR"
12    }
13  ]

```

**API Endpoint:** `PUT /api/events/{id}/permissions/{user_id}/`

#### Summary

Update the role of a specific user for an event.

#### URL

`PUT /api/events/3/permissions/3/`

#### Request Headers

`Authorization: Bearer <your-access-token>`

`Content-Type: application/json`

#### Request Body

```
{  
  "role": "VIEWER"  
}
```

#### RESPONSE-

```
{  
  "message": "User permission updated",  
  "user_id": 3,  
  "role": "VIEWER"  
}
```

#### Example cURL

```
curl --location --request PUT  
'http://localhost:8000/api/events/3/permissions/3/' \  
--header 'Authorization: Bearer <your-access-token>' \  
--header 'Content-Type: application/json' \  
--data '{"role": "VIEWER"}'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'event\_management' selected. The main area shows a 'share event/permissions' collection with a single PUT request to `http://localhost:8000/api/events/3/permissions/3`. The request body is set to `{"role": "VIEWER"}`. The response status is 200 OK with a message: "User permission updated", "user\_id": 3, "role": "VIEWER".

**API Endpoint: DELETE /api/events/{id}/permissions/{user\_id}/**

## Summary

Remove a user's permission for an event.

## URL

**DELETE /api/events/3/permissions/3/**

## Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

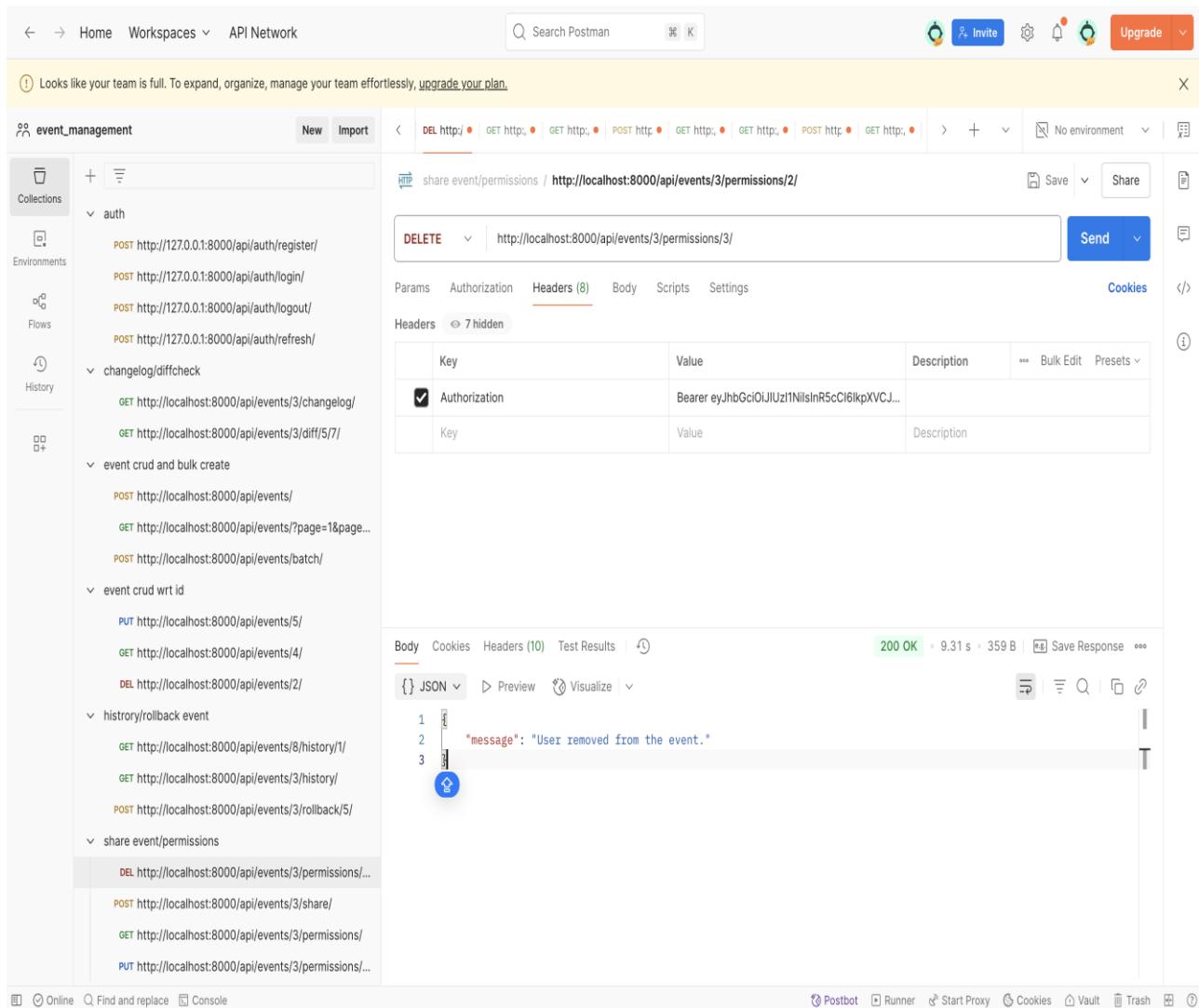
REPSONE-

{

```
"message": "User removed from the event."
}
```

### Example cURL

```
curl --location --request DELETE
'http://localhost:8000/api/events/3/permissions/3/' \
--header 'Authorization: Bearer <your-access-token>' \
--header 'Content-Type: application/json'
```



The screenshot shows the Postman application interface. On the left, there's a sidebar with 'event\_management' selected, showing various API endpoints categorized under 'auth', 'changelog/diffcheck', 'event crud and bulk create', 'event crud wrt id', 'history/rollback event', and 'share event/permissions'. The main panel shows a DELETE request to `http://localhost:8000/api/events/3/permissions/3/`. The 'Headers' tab is active, containing the following table:

Key	Value	Description	Bulk Edit	Presets
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...			
Key	Value	Description		

Below the request, the response section shows a green '200 OK' status with a response time of 9.31 s and a size of 359 B. The 'Body' tab displays the JSON response:

```

1   {
2     "message": "User removed from the event."
3   }

```

**API Endpoint: GET /api/events/{id}/history/**

Summary

Retrieve the edit history of an event respective history id

URL

**GET /api/events/3/history/**

Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

RESPONSE-

```
[  
  {  
    "version_id": 22,  
    "event_id": 3,  
    "history_type": "~",  
    "history_date": "2025-05-24 12:09:12"  
  },  
  {  
    "version_id": 9,  
    "event_id": 3,  
    "history_type": "~",  
    "history_date": "2025-05-23 18:04:39"  
  },  
]
```

Example cURL

```
curl --location 'http://localhost:8000/api/events/3/history/' \  
--header 'Authorization: Bearer <your-access-token>' \  
--header 'Content-Type: application/json'
```

The screenshot shows the Postman application interface. The left sidebar contains collections, environments, flows, and history. The main area displays the 'event\_management' collection with several API endpoints listed under categories like 'auth', 'changelog/diffcheck', 'event crud and bulk create', 'event crud wrt id', and 'history/rollback event'. A specific POST request for 'history/rollback event' is selected, showing its details in the center panel. The request URL is `http://localhost:8000/api/events/3/history/`. The 'Headers' tab is active, showing 'Authorization' and 'Content-Type' headers. The 'Body' tab shows a JSON response with two history entries. The bottom right corner shows the response status as '200 OK'.

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan](#).

**event\_management**

New Import

PUT http://localhost:8000/api/events/3/history/

GET http://localhost:8000/api/events/3/history/

Params Authorization Headers (9) Body Scripts Settings

Headers (7 hidden)

Key	Value	Description	Bulk Edit	Presets
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...			
Content-Type	application/json			
Key	Value	Description		

Body Cookies Headers (10) Test Results

200 OK 2.46 s 920 B Save Response

{ } JSON Preview Visualize

```
[{"version_id": 22, "event_id": 3, "history_type": "-", "history_date": "2025-05-24 12:09:12"}, {"version_id": 9, "event_id": 3, "history_type": "-", "history_date": "2025-05-23 18:04:39"}]
```

Online Find and replace Console

Prophet Runner Start Proxy Cookies Vault Trash

API Endpoint: GET /api/events/{id}/history/{history\_id}/

## Summary

Retrieve a specific version from the event's edit history.

## URL

**GET /api/events/3/history/4/**

## Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

## RESPONSE-

```
{
  "id": 3,
  "title": "for case 4 history",
  "description": "lwts description",
  "start_time": "2025-06-01 11:23:00",
  "end_time": "2025-06-01 18:32:00",
  "location": "srinagar",
  "is_recurring": true,
  "recurrence_pattern": "weekly",
  "history_date": "2025-05-23T12:43:04.761531Z",
  "history_type": "~"
}
```

### Example cURL

```
curl --location 'http://localhost:8000/api/events/3/history/4/' \
--header 'Authorization: Bearer <your-access-token>' \
--header 'Content-Type: application/json'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with 'Collections' expanded, showing a tree structure of API endpoints under 'event\_management'. The 'History' section is also visible. In the main area, a specific request is selected:

- Request URL:** `http://localhost:8000/api/events/3/history/4/`
- Method:** `GET`
- Headers:**
  - `Authorization`: `Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...`
  - `Content-Type`: `application/json`
- Body:** JSON preview showing the JSON object defined at the top of the page.

**API Endpoint:** POST /api/events/{id}/rollback/{history\_id}/

#### Summary

Rollback the event to a specific historical version.

#### URL

**POST /api/events/3/rollback/4/**

#### Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

#### RESPONSE-

{

```
"message": "Event rolled back successfully",
"event": {
  "id": 3,
  "title": "for case 4 history",
  "description": "lwts description",
  "start_time": "2025-06-01 11:23:00",
  "end_time": "2025-06-01 18:32:00",
  "location": "srinagar",
  "is_recurring": true,
  "recurrence_pattern": "weekly",
  "owner_id": 1
}
```

#### Example cURL

```
curl --location --request POST 'http://localhost:8000/api/events/3/rollback/4/' \
\

--header 'Authorization: Bearer <your-access-token>' \
--header 'Content-Type: application/json'
```

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan](#).

## event\_management

New Import

POST http://localhost:8000/api/events/3/rollback/5/

Save Share

Send

Collections

Environments

Flows

History

+

+

auth

POST http://127.0.0.1:8000/api/auth/register/

POST http://127.0.0.1:8000/api/auth/login/

POST http://127.0.0.1:8000/api/auth/logout/

POST http://127.0.0.1:8000/api/auth/refresh/

changelog/diffcheck

GET http://localhost:8000/api/events/3/changelog/

GET http://localhost:8000/api/events/3/diff/5/7/

event crud and bulk create

POST http://localhost:8000/api/events/

GET http://localhost:8000/api/events/?page=1&amp;page...

POST http://localhost:8000/api/events/batch/

event crud wrt id

PUT http://localhost:8000/api/events/5/

GET http://localhost:8000/api/events/4/

DEL http://localhost:8000/api/events/2/

history/rollback event

GET http://localhost:8000/api/events/8/history/1/

GET http://localhost:8000/api/events/3/history/

POST http://localhost:8000/api/events/3/rollback/5/

share event/permissions

DEL http://localhost:8000/api/events/3/permissions/...

POST http://localhost:8000/api/events/3/share/

GET http://localhost:8000/api/events/3/permissions/

PUT http://localhost:8000/api/events/3/permissions/...

HTTP history/rollback event / http://localhost:8000/api/events/3/rollback/5/

POST http://localhost:8000/api/events/3/rollback/4/

Save Share

Send

Params Authorization Headers (10) Body Scripts Settings

Cookies

Headers 8 hidden

	Key	Value	Description	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ...			
<input checked="" type="checkbox"/>	Content-Type	application/json			
	Key	Value	Description		

Body Cookies Headers (10) Test Results

200 OK 1.82 s 587 B Save Response

{ JSON Preview Visualize

```
1 i
2   "message": "Event rolled back successfully",
3   "event": {
4     "id": 3,
5     "title": "for case 4 history",
6     "description": "lwts description",
7     "start_time": "2025-06-01 11:23:00",
8     "end_time": "2025-06-01 18:32:00",
9     "location": "srinagar",
10    "is_recurring": true,
11    "recurrence_pattern": "weekly",
12    "owner_id": 1
13  }
```

Online Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash

**API Endpoint: GET /api/events/{id}/changelog/**

Summary

Retrieve a list of all changes made to a specific event.

URL

**GET /api/events/3/changelog/**

Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

RESPONSE-

```
[  
  {  
    "history_id": 3,  
    "history_date": "2025-05-23 12:36:37",  
    "history_type": "~",  
    "changed_by": null,  
    "change_reason": null,  
    "changed_fields": {}  
  },  
  {  
    "history_id": 4,  
    "history_date": "2025-05-23 12:43:04",  
    "history_type": "~",  
    "changed_by": null,  
    "change_reason": null,  
    "changed_fields": {  
      "title": {  
        "from": "testing history",  
        "to": "for case 4 history"  
      },  
      "description": {  
        "from": "history description",  
        "to": "lwts description"  
      }  
    }  
  },  
]
```

**Example cURL**

```
curl --location 'http://localhost:8000/api/events/3/changelog/' \
--header 'Authorization: Bearer <your-access-token>' \
--header 'Content-Type: application/json'
```

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Collections, Environments, Flows, and History. The 'Collections' section is expanded, showing a tree structure with categories like 'auth', 'changelog/diffcheck', 'event crud and bulk create', 'event crud wrt id', 'history/rollback event', and 'share event/permissions'. Under each category, there are several API requests listed with their URLs and methods (e.g., POST, GET, PUT, DEL). The main workspace on the right displays a specific request for 'changelog/diffcheck' at 'http://localhost:8000/api/events/3/changelog'. This request is a GET method. Below the request details, there are tabs for Body, Cookies, Headers (10), and Test Results. The Headers tab is active, showing two entries: 'Authorization' with value 'Bearer eyJhbGciOiJIUzI1NiIsInR5CjI6IkpXVCJ...' and 'Content-Type' with value 'application/json'. The Body tab shows a JSON response with a code editor view. The status bar at the bottom includes links for Online, Find and replace, Console, Postbot, Runner, Start Proxy, Cookies, Vault, Trash, and Help.

**API Endpoint:** GET /api/events/{id}/diff/{old\_version\_id}/{new\_version\_id}/

#### Summary

View the difference between two historical versions of an event.

#### URL

**GET /api/events/3/diff/3/4/**

#### Request Headers

Authorization: Bearer <your-access-token>

Content-Type: application/json

#### RESPONSE-

```
{  
  "title": {  
    "version_1": "testing history",  
    "version_2": "for case 4 history"  
  },  
  "description": {  
    "version_1": "history description",  
    "version_2": "lwts description"  
  }  
}
```

#### Example cURL

```
curl --location 'http://localhost:8000/api/events/3/diff/3/4/' \  
--header 'Authorization: Bearer <your-access-token>' \  
--header 'Content-Type: application/json'
```

Home Workspaces API Network

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan.](#)

**event\_management**

New Import

Collections Environments Flows History

auth

changelog/diffcheck / http://localhost:8000/api/events/3/diff/5/7

GET http://localhost:8000/api/events/3/diff/3/4

Params Authorization Headers (9) Body Scripts Settings Cookies

Headers (7 hidden)

Key	Value	Description	Bulk Edit	Presets
Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ...			
Content-Type	application/json			
Key	Value	Description		

Body Cookies Headers (10) Test Results 200 OK 2.00 s 471 B Save Response

{ JSON Preview Visualize }

```
1 {
2   "title": {
3     "version_1": "testing history",
4     "version_2": "for case 4 history"
5   },
6   "description": {
7     "version_1": "history description",
8     "version_2": "lwts description"
9   }
10 }
```

Online Find and replace Console

Postbot Runner Start Proxy Cookies Vault Trash