

GitHub Link - <https://github.com/JyotsnaJahanaviDevi/dt-events-api>

POST - create a new event

created 3 events with titles - Deep Thought, Netflix, The Dark Knight which responded with 3 ids

POST http://localhost:3000/api/v3/app/events

Key	Value
uid	18
name	The Dark Knight
tagline	I am Batman
schedule	2025-12-06T23:43:00
description	Directed by Christopher Nolan
image	Select files
moderator	Bob
category	Movie
sub_category	Sci-fi
rigor_rank	1

Body Cookies Headers (7) Test Results

{ } JSON D Preview Visualize

```
1 {
2   "id": "69347db09b5b401362fb312c"
3 }
```

GET - all the events upto limit 5 (fetches the created 3 events)

event 1

GET http://localhost:3000/api/v3/app/events?type=latest&limit=5&page=1

Params

Key	Value
type	latest
limit	5
page	1

Body Cookies Headers (7) Test Results

{ } JSON D Preview Visualize

```
1 [
2   {
3     "_id": "69347db09b5b401362fb312a",
4     "type": "event",
5     "uid": 18,
6     "name": "DeepThought",
7     "tagline": "Think deeply",
8     "schedule": "2025-12-06T18:13:00.000Z",
9     "description": "A software company",
10    "files": [
11      {
12        "image": "uploads\\1765047288901-no face.jpg"
13      }
14    ],
15    "moderator": "Adam",
16    "category": "Technology",
17    "sub_category": "Software Development",
18    "rigor_rank": 1,
19    "attendees": []
20  }
21 ]
```

event 2

The screenshot shows the Postman application interface. The URL in the address bar is `http://localhost:3000/api/v3/app/events`. The request method is set to `GET`, and the query parameters are `type=latest&limit=5&page=1`. The response status is `200 OK` with a response time of `67 ms` and a size of `1.15 KB`. The response body is displayed in JSON format, showing two event documents. The first event has an _id of `69347db09b5b401362fb312b`, a type of `event`, uid `18`, name `Netflix`, tagline `"Your streaming partner"`, schedule `"2025-12-06T18:13:00.000Z"`, description `"An entertainment platform"`, files `null`, moderator `"Bob"`, category `"Social Media"`, sub_category `"OTT"`, rigor_rank `1`, and attendees `[]`. The second event has an _id of `69347db09b5b401362fb312c`, a type of `event`, uid `18`, name `The Dark Knight`, tagline `"I am Batman"`, schedule `"2025-12-06T18:13:00.000Z"`, description `"Directed by Christopher Nolan"`, files `null`, moderator `"Bob"`, category `"Movie"`, sub_category `"Sci-fi"`, rigor_rank `1`, and attendees `[]`.

event 3

The screenshot shows the Postman application interface. The URL in the address bar is `http://localhost:3000/api/v3/app/events`. The request method is set to `GET`, and the query parameters are `type=latest&limit=5&page=1`. The response status is `200 OK` with a response time of `67 ms` and a size of `1.15 KB`. The response body is displayed in JSON format, showing two event documents. The first event has an _id of `69347db09b5b401362fb312b`, a type of `event`, uid `18`, name `Netflix`, tagline `"Your streaming partner"`, schedule `"2025-12-06T18:13:00.000Z"`, description `"An entertainment platform"`, files `null`, moderator `"Bob"`, category `"Social Media"`, sub_category `"OTT"`, rigor_rank `1`, and attendees `[]`. The second event has an _id of `69347db09b5b401362fb312c`, a type of `event`, uid `18`, name `The Dark Knight`, tagline `"I am Batman"`, schedule `"2025-12-06T18:13:00.000Z"`, description `"Directed by Christopher Nolan"`, files `null`, moderator `"Bob"`, category `"Movie"`, sub_category `"Sci-fi"`, rigor_rank `1`, and attendees `[]`.

GET - fetch event by ID

3rd event(the dark knight) fetched by ID

The screenshot shows the Postman interface with a successful GET request. The URL is `http://localhost:3000/api/v3/app/events?_id=69347db09b5b401362fb312c`. The response body is a JSON object representing the event "The Dark Knight".

```
1 {  
2   "_id": "69347db09b5b401362fb312c",  
3   "type": "event",  
4   "uid": 18,  
5   "name": "The Dark Knight",  
6   "tagline": "I am Batman",  
7   "schedule": "2025-12-06T18:13:00.000Z",  
8   "description": "Directed by Christopher Nolan",  
9   "files": null,  
10  "moderator": "Bob",  
11  "category": "Movie",  
12  "sub_category": "Sci-fi",  
13  "rigor_rank": 1,  
14  "attendees": []  
15 }
```

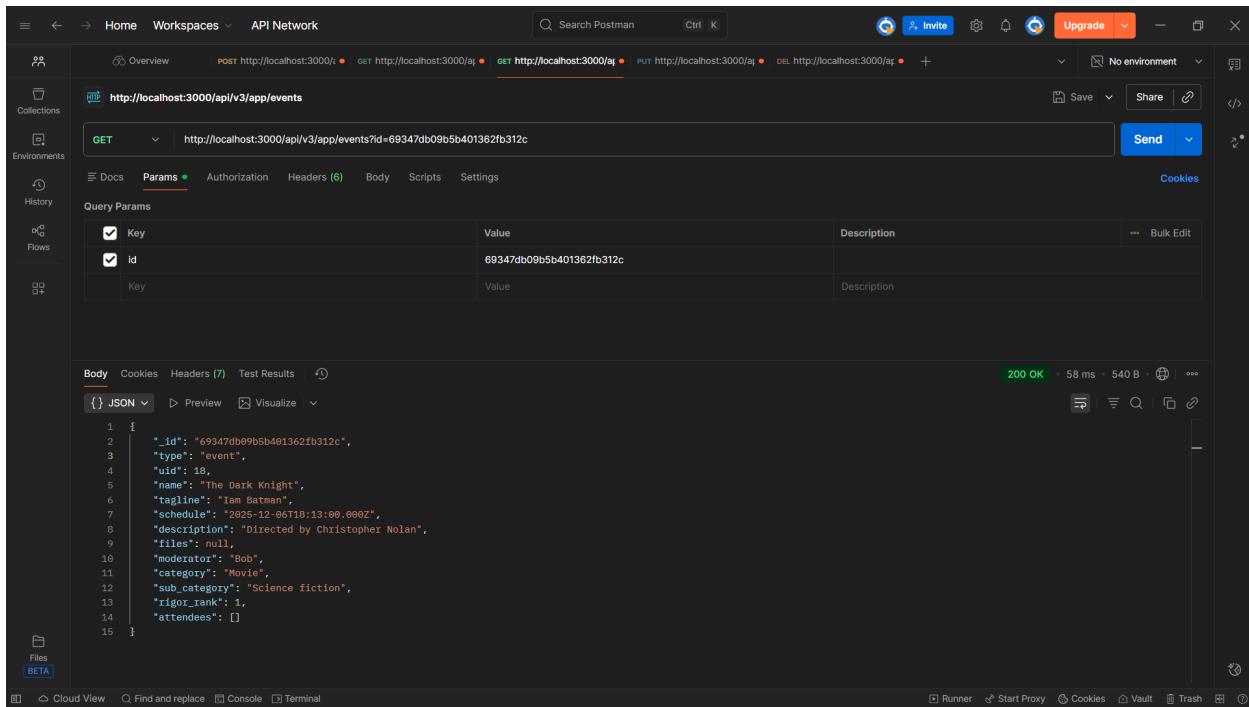
PUT - updating content of existing event by ID

updated sub_category of 3rd event (the dark knight) from sci-fi to Science fiction
shows “Event updated successfully” message

The screenshot shows the Postman interface with a successful PUT request. The URL is `http://localhost:3000/api/v3/app/events/69347db09b5b401362fb312c`. The request body contains a key-value pair: `sub_category: Science fiction`. The response body is a JSON object with a single key: `message: "Event updated successfully"`.

```
1 {  
2   "message": "Event updated successfully"  
3 }
```

GET - fetching the updated event to check the updated changes
 sub_category updated from sci-fi to Science fiction



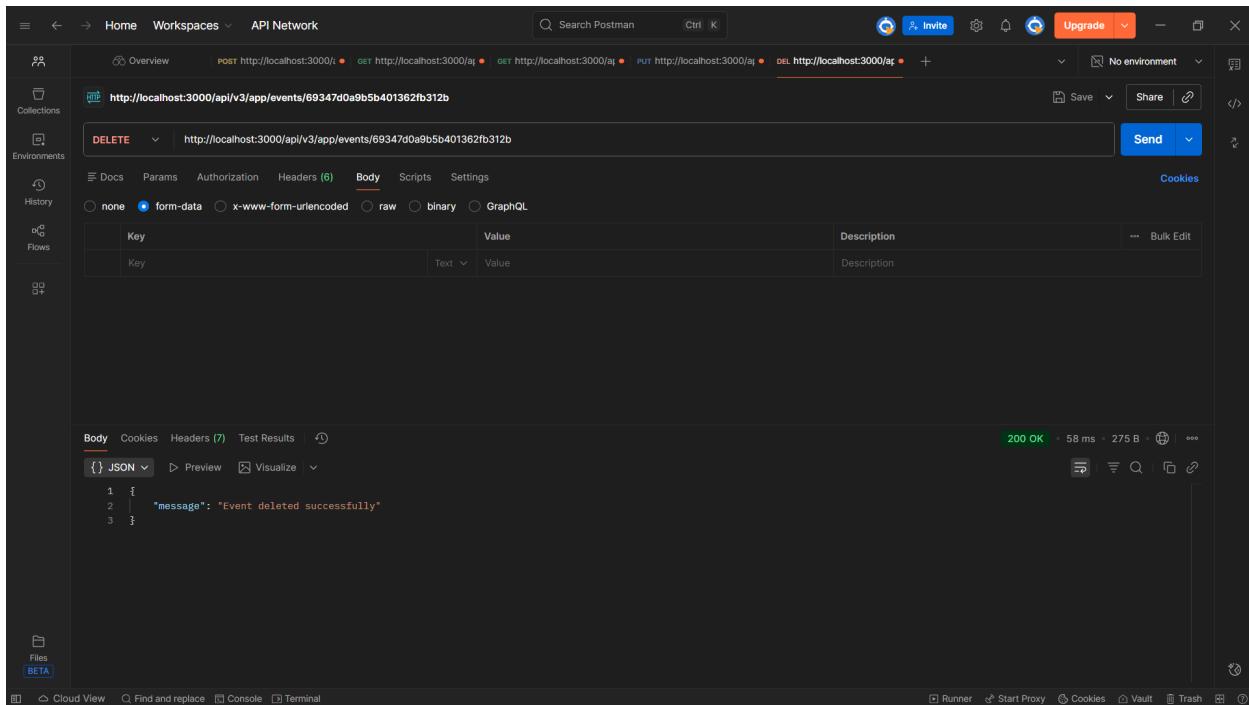
The screenshot shows the Postman interface with a successful GET request. The URL is `http://localhost:3000/api/v3/app/events?d=69347db09b5b401362fb312c`. The response status is 200 OK, and the response body is a JSON object representing an event:

```

1  {
2     "_id": "69347db09b5b401362fb312c",
3     "type": "event",
4     "uid": 18,
5     "name": "The Dark Knight",
6     "tagline": "I am Batman",
7     "schedule": "2025-12-06T18:13:00.000Z",
8     "description": "Directed by Christopher Nolan",
9     "files": null,
10    "moderator": "Bob",
11    "category": "Movie",
12    "sub_category": "Science fiction",
13    "rigo_rank": 1,
14    "attendees": []
15 }

```

DELETE - delete event by ID
 deleted 2nd (Netflix) event
 shows “Event deleted successfully” message



The screenshot shows the Postman interface with a successful DELETE request. The URL is `http://localhost:3000/api/v3/app/events/69347d0a9b5b401362fb312b`. The response status is 200 OK, and the response body is a JSON object with a single key-value pair:

```

1  {
2     "message": "Event deleted successfully"
3 }

```

GET - fetching the deleted event to check if its deleted
fetching 2nd(Netflix) event by ID
shows “Event not found” error since it is a deleted event

The screenshot shows the Postman application interface. In the top navigation bar, there are tabs for Home, Workspaces, and API Network. The main workspace shows a collection named "http://localhost:3000/api/v3/app/events". A specific GET request is selected, with the URL being `http://localhost:3000/api/v3/app/events?id=69347d0a9b5b401362fb312b`. The "Params" tab is active, showing two parameters: "Key" and "id", both with the value `69347d0a9b5b401362fb312b`. Below the request details, the "Body" tab is open, showing a JSON response with a single key-value pair: `{ "error": "Event not found" }`. The status bar at the bottom indicates a 404 Not Found error, 68 ms duration, and 269 B size.