

## 1. Register user

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8081/api/users/register`
- Method:** `POST`
- Body (raw):**

```
1 {
2   "name": "Alice Smith",
3   "email": "alice@gmail.com",
4   "password": "password123",
5   "role": "Attendee"
6 }
```
- Response:** `200 OK`, 20 ms, 266 B. The response body (JSON) is:

```
1 {
2   "name": "Alice Smith",
3   "email": "alice@gmail.com",
4   "password": "password123",
5   "role": "Attendee",
6   "userId": 4
7 }
```

## 2. Create event

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8081/api/events/create?organizerId=3&venueId=1`
- Method:** `POST`
- Body (raw):**

```
1 {
2   "eventId": 4,
3   "name": "skating",
4   "description": "learn how to skating",
5   "startDateTime": "2025-06-15T09:00:00",
6   "endDateTime": "2025-06-15T17:00:00",
7   "price": 50.00,
8   "organiser": {
9     "name": "Ankit",
10    "email": "Ankit@gmail.com",
11    "password": "my123",
12    "role": "Organizer",
13    "userId": 3
14  },
15   "venue": {
16     "venueId": 1,
17     "name": "Esplanade Triantille",
18     "location": "1442 Rue Clark Montreal",
19     "capacity": 150,
20     "description": "a spacious venue for large event",
21     "events": [
```
- Response:** `200 OK`, 317 ms, 137.88 KB.

### 3. Get user by id

The screenshot shows a REST client interface with the following details:

- Request:** GET `http://localhost:8081/api/users/3`
- Response:** 200 OK, 14 ms, 261 B
- Body (JSON):**

```
1 {
2   "name": "ankit",
3   "email": "ankit@gmail.com",
4   "password": "password123",
5   "role": "Organizer",
6   "userId": 3
7 }
```

### 4. Get event by id

The screenshot shows a REST client interface with the following details:

- Request:** GET `http://localhost:8081/api/events/4`
- Response:** 200 OK, 13 ms, 142.12 KB
- Body (JSON):**

```
1 {
2   "eventId": 4,
3   "name": "skating",
4   "description": "learn how to skating and have fun",
5   "startDateTime": "2025-05-15T09:00:00",
6   "endDateTime": "2025-06-15T17:00:00",
7   "price": 50.00,
8   "organiser": {
9     "name": "Ankit",
10    "email": "Ankit@gmail.com",
11    "password": "my123",
12    "role": "Organizer",
13    "userId": 3
14  },
15  "venue": {
16    "venueId": 1,
17    "name": "Esplanade Triantille",
18    "location": "1442 Rue Clark Montreal",
19    "capacity": 150,
20    "description": "a spacious venue for large event",
21    "events": [
```

## 5. Update event

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** `http://localhost:8081/api/events/4?userId=3`
- Body:** JSON format, raw view. The JSON payload is:

```
{  "eventId": 4,  "name": "skating",  "description": "learn how to skating and have fun",  "startDateTime": "2025-05-15T09:00:00",  "endDateTime": "2025-06-15T17:00:00",  "price": 50.00,  "organiser": {    "name": "Ankit",    "email": "Ankit@gmail.com",    "password": "my123",    "role": "Organizer",    "userId": 3  },  "venue": {    "venueId": 1,    "name": "Esplanade Trantille",    "location": "1442 Rue Clark Montreal",    "capacity": 150,    "description": "a spacious venue for large event",    "venueId": 1  }}
```
- Response:** 200 OK, 20 ms, 142.12 KB. The response body is in JSON format, preview view, showing the same event data as the request body.

## 6. Update user

The screenshot shows a REST client interface with the following details:

- Method:** PUT
- URL:** `http://localhost:8081/api/users/3`
- Body:** JSON format, raw view. The JSON payload is:

```
{  "name": "Alice Updated",  "email": "alice.updated@example.com",  "password": "my123",  "role": "Organizer"}
```
- Response:** 200 OK, 65 ms, 273 B. The response body is in JSON format, preview view, showing the updated user data:

```
{  "name": "Alice Updated",  "email": "alice.updated@example.com",  "password": "my123",  "role": "Organizer",  "userId": 3}
```

## 7. User register for event

The screenshot shows a web browser interface for testing HTTP requests. The address bar displays the URL: `http://localhost:8081/api/registrations/register?eventId=4&attendeeld=4`. The request method is set to **POST**. The response status is **200 OK**, with a response time of 686 ms and a size of 141.96 KB.

The response body is displayed in JSON format:

```
1 {
2   "event": {
3     "eventId": 4,
4     "name": "skating",
5     "description": "learn how to skating and have fun",
6     "startDateTime": "2025-05-15T09:00:00",
7     "endDateTime": "2025-06-15T17:00:00",
8     "price": 50.00,
9     "organiser": {
10      "name": "Ankit",
11      "email": "Ankit@gmail.com",
12      "password": "my123",
13      "role": "Organizer",
14      "userId": 3
15    },
16    "venue": {
17      "venueId": 1,
18      "name": "Esplanade Trantille",
19      "location": "1442 Rue Clark Montreal",
20      "capacity": 150,
21      "description": "a spacious venue for large event",
```

## 8. Get user who registered for event

Workspaces ▾ More ▾

🔍

👤

⚙️

🔔

🌐

Upgrade ▾

—

👤 Overview

GET http://localhost:8081/ap

+

▾

🗑️ No environment

🔗 HTTP

http://localhost:8081/api/registrations/event/4/attendees

💾 Save ▾

Str

GET ▾

http://localhost:8081/api/registrations/event/4/attendees

Send

Params

Auth

Headers (8)

Body ●

Scripts

Settings

Co

<input checked="" type="checkbox"/>	Accept	ⓘ	*/*	
<input checked="" type="checkbox"/>	Accept-Encoding	ⓘ	gzip, deflate, br	
<input checked="" type="checkbox"/>	Connection	ⓘ	keep-alive	
	Key		Value	Description

Body ▾

🔄

200 OK • 403 ms • 268 B • 🌐

{ } JSON ▾

▶ Preview

🔗 Visualize ▾

🔄 📄 🔍

```
1  [  
2    {  
3      "name": "Alice Smith",  
4      "email": "alice@gmail.com",  
5      "password": "password123",  
6      "role": "Attendee",  
7      "userId": 4  
8    }  
9  ]
```

9. User tried to delet event without permission

Workspaces ▾ More ▾

Overview **DEL** http://localhost:8081/ap +

No environment ▾

Save ▾ Share

**DELETE** ▾ http://localhost:8081/api/events/4?userId=4 **Send** ▾

Params ● Auth Headers (8) **Body** ● Scripts Settings Cookies

raw ▾ **JSON** ▾ Beautify

1 {

Body ▾ ↺

**500 Internal Server Error** • 29 ms • 267 B • 🌐 ⋮

{ } JSON ▾ ▶ Preview 🔄 Visu

1 {

2 "timestamp": "2025-02

3 "status": 500,

4 "error": "Internal Se

5 "path": "/api/events/4"

6 }

**500 Internal Server Error**

The server has encountered a situation it does not know how to handle.

10. Admin delete event



<http://localhost:8081/api/events/4?userId=5>



Save

Share



DELETE

<http://localhost:8081/api/events/4?userId=5>

Send



[Params](#)
[Auth](#)
[Headers \(8\)](#)
[Body](#)
[Scripts](#)
[Settings](#)

## Cookies

raw JSON 

## Beautify



```
1 {
2   "name": "admin me",
3   "email": "admin@gmail.com",
4   "password": "me123",
5   "role": "Admin"
6 }
7
```

Body  

200 OK • 80 ms • 123 B •  | 



Raw 



▶ Preview



 Visualize 



1