API Specification for Event Management System

Base URL

http://localhost:5000/api

Authentication

All endpoints, except user registration and login, require a **Bearer Token** in the Authorization header.

Endpoints

1. User Registration

POST /auth/register
Description: Create a new user account.

Request Body:

```
{
  "username": "johndoe",
  "email": "john.doe@example.com",
  "password": "password123"
}
```

Response:

- **201 Created**

```
{
  "message": "User registered successfully.",
  "userId": 1
}
```

- **400 Bad Request**

```
{
  "error": "Email already exists."
}
```

---

2. User Login

POST /auth/login
Description: Authenticate a user and return a JWT.

Request Body:

```
{
  "email": "john.doe@example.com",
  "password": "password123"
}
```

Response:

- **200 OK**

```
{
  "token": "<JWT_TOKEN>"
}
```

- **401 Unauthorized**

```
{
  "error": "Invalid email or password."
}
```

---

3. Get All Events

GET /events
Description: Retrieve a list of all events.

Headers:
Authorization: Bearer <JWT_TOKEN>

Response:

- **200 OK**

```
[
  {
    "id": 1,
    "name": "Music Festival",
    "description": "A grand music festival featuring various artists.",
    "date": "2024-12-15"
  },
  {
    "id": 2,
    "name": "Tech Conference",
    "description": "A conference to showcase the latest technology trends.",
    "date": "2024-12-20"
  }
]
```

---

4. Create an Event (Admin Only)

POST /events
Description: Create a new event.

Headers:
Authorization: Bearer <JWT_TOKEN>

Request Body:

```
{
 "name": "Art Exhibition",
 "description": "An exhibition of contemporary art pieces.",
 "date": "2025-01-10"
}
```

Response:

- **201 Created**

```
{
 "id": 3,
 "name": "Art Exhibition",
 "description": "An exhibition of contemporary art pieces.",
 "date": "2025-01-10"
}
```

- **403 Forbidden**

```
{
 "error": "Admin access required."
}
```

---

5. RSVP to an Event

POST /events/:eventId/rsvps
Description: RSVP to a specific event.

Headers:
Authorization: Bearer <JWT_TOKEN>

Path Parameter:

- eventId (integer): The ID of the event to RSVP to.

Response:

- **200 OK**

```
{
 "message": "RSVP confirmed for event.",
 "eventId": 1
}
```

- **404 Not Found**

```
{
 "error": "Event not found."
}
```

---

6. Get RSVPs for an Event (Admin Only)

GET /events/:eventId/rsvps
Description: Retrieve a list of users who RSVPed for a specific event.

Headers:
Authorization: Bearer <JWT_TOKEN>

Path Parameter:

- eventId (integer): The ID of the event.

Response:

- **200 OK**

```
[
 {
  "userId": 2,
  "username": "johndoe",
  "email": "john.doe@example.com"
 },
 {
  "userId": 3,
  "username": "janedoe",
  "email": "jane.doe@example.com"
 }
]
```

- **403 Forbidden**

```
{
 "error": "Admin access required."
}
```

---

7. Delete an Event (Admin Only)

DELETE /events/:eventId
Description: Delete a specific event.

Headers:
Authorization: Bearer <JWT_TOKEN>

Path Parameter:

- eventId (integer): The ID of the event.

Response:

- **200 OK**

```
{
  "message": "Event deleted successfully."
}
```

- **404 Not Found**

```
{
  "error": "Event not found."
}
```

---

8. Get User Profile
GET /users/profile
Description: Retrieve the logged-in user's profile.

Headers:
Authorization: Bearer <JWT_TOKEN>

Response:

- **200 OK**

```
{
  "id": 2,
  "username": "johndoe",
  "email": "john.doe@example.com",
  "isAdmin": false
}
```

Thank You
Edges For Training Team