

Event Management System API

Documentation

Base URL

<https://EGO/api/v1>

Authentication

All protected routes require a JWT token in the cookie header. The token is obtained by logging in via the `/login` endpoint.

User Management

Register a new user

- **Endpoint:** POST `/register`
- **Description:** Creates a new user account
- **Request Body:**

```
{  "name": "string",  "email": "string",  "password": "string",  "role": "string" // Optional, defaults to "Standard User"}
```
- **Success Response:** 201 Created
- **Error Responses:**
 - 409 Conflict - User already exists
 - 500 Internal Server Error - Server error

Login

- **Endpoint:** POST /login
- **Description:** Authenticates user and returns JWT token in cookie
- **Request Body:**

```
{  
  "email": "string",  
  "password": "string"  
}
```
- **Success Response:** 200 OK with user details
- **Error Responses:**
 - 404 Not Found - Email not found
 - 405 Method Not Allowed - Incorrect password
 - 500 Internal Server Error - Server error

Password Reset

1. Request OTP

- a. **Endpoint:** POST /sendOtp
- b. **Description:** Sends OTP to user's email for password reset
- c. **Request Body:**

```
{  
  "email": "string"  
}
```
- d. **Success Response:** 200 OK
- e. **Error Responses:**
 - i. 404 Not Found - User not found
 - ii. 500 Internal Server Error - Server error

2. Reset Password

- a. **Endpoint:** PUT /forgetPassword
- b. **Description:** Resets user password with valid OTP
- c. **Request Body:**

```
{  
  "email": "string",  
  "otp": "string",  
  "password": "string"  
}
```
- d. **Success Response:** 200 OK

e. Error Responses:

- i. 400 Bad Request - Incorrect OTP or same password
- ii. 404 Not Found - User not found
- iii. 500 Internal Server Error - Server error

Get Current User Profile

- **Endpoint:** GET /users/profile
- **Authentication:** Required (All roles)
- **Description:** Returns current user's profile
- **Success Response:** 200 OK with user details
- **Error Response:** 500 Internal Server Error

Update Current User Profile

- **Endpoint:** PUT /users/profile
- **Authentication:** Required (All roles)
- **Description:** Updates current user's profile
- **Request Body:** User fields to update
- **Success Response:** 200 OK
- **Error Responses:**
 - 400 Bad Request - Trying to use same password
 - 404 Not Found - User not found
 - 500 Internal Server Error - Server error

Admin: Get All Users

- **Endpoint:** GET /users
- **Authentication:** Required (System Admin only)
- **Description:** Returns list of all users
- **Success Response:** 200 OK with array of users
- **Error Response:** 500 Internal Server Error

Admin: Get User by ID

- **Endpoint:** GET /users/:id
- **Authentication:** Required (System Admin only)
- **Description:** Returns details of a specific user

- **Success Response:** 200 OK with user details
- **Error Response:** 500 Internal Server Error

Admin: Update User Role

- **Endpoint:** PUT /users/:id
- **Authentication:** Required (System Admin only)
- **Description:** Updates a user's role
- **Request Body:**

```
{
  "role": "string" // "System Admin", "Organizer", or "Standard User"
}
```
- **Success Response:** 200 OK
- **Error Response:** 500 Internal Server Error

Admin: Delete User

- **Endpoint:** DELETE /users/:id
- **Authentication:** Required (System Admin only)
- **Description:** Deletes a user
- **Success Response:** 200 OK
- **Error Response:** 500 Internal Server Error

Event Management

Get Approved Events

- **Endpoint:** GET /events
- **Description:** Returns list of all approved events (public)
- **Success Response:** 200 OK with array of events
- **Error Response:** 500 Internal Server Error

Admin: Get All Events

- **Endpoint:** GET /events/all

- **Authentication:** Required (System Admin only)
- **Description:** Returns list of all events regardless of status
- **Success Response:** 200 OK with array of events
- **Error Response:** 500 Internal Server Error

Get Event by ID

- **Endpoint:** GET /events/:id
- **Description:** Returns details of a specific event (public)
- **Success Response:** 200 OK with event details
- **Error Responses:**
 - 404 Not Found - Event not found
 - 500 Internal Server Error

Create Event

- **Endpoint:** POST /events
- **Authentication:** Required (Organizer only)
- **Description:** Creates a new event
- **Request Body:**

```
{
  "title": "string",
  "description": "string",
  "date": "ISO Date",
  "location": "string",
  "category": "string",
  "ticketPrice": number,
  "ticketAvailable": number
}
```
- **Success Response:** 201 Created with event details
- **Error Response:** 400 Bad Request or 500 Internal Server Error

Update Event

- **Endpoint:** PUT /events/:id
- **Authentication:** Required (Organizer or System Admin)
- **Description:** Updates an existing event
 - Organizers can only update their own events (cannot change status)

- Admins can update any event and change status
- **Request Body:** Event fields to update
- **Success Response:** 200 OK with updated event
- **Error Responses:**
 - 404 Not Found - Event not found or unauthorized
 - 400 Bad Request or 500 Internal Server Error

Delete Event

- **Endpoint:** DELETE /events/:id
- **Authentication:** Required (Organizer or System Admin)
- **Description:** Deletes an event
 - Organizers can only delete their own events
 - Admins can delete any event
- **Success Response:** 200 OK
- **Error Responses:**
 - 404 Not Found - Event not found or unauthorized
 - 500 Internal Server Error

Organizer: Get My Events

- **Endpoint:** GET /users/events
- **Authentication:** Required (Organizer only)
- **Description:** Returns list of events created by current organizer
- **Success Response:** 200 OK with array of events
- **Error Response:** 500 Internal Server Error

Organizer: Get Event Analytics

- **Endpoint:** GET /users/events/analytics
- **Authentication:** Required (Organizer only)
- **Description:** Returns booking analytics for organizer's events
- **Success Response:** 200 OK with analytics data
- **Error Response:** 500 Internal Server Error

Booking Management

Book Tickets

- **Endpoint:** POST /bookings
- **Authentication:** Required (Standard User only)
- **Description:** Books tickets for an event
- **Request Body:**

```
{  
  "eventId": "string",  
  "numberOfTickets": number  
}
```
- **Success Response:** 201 Created with booking details
- **Error Responses:**
 - 400 Bad Request - Not enough tickets available
 - 404 Not Found - Event not found
 - 500 Internal Server Error

Get Booking by ID

- **Endpoint:** GET /bookings/:id
- **Authentication:** Required (Standard User only)
- **Description:** Returns details of a specific booking (must belong to current user)
- **Success Response:** 200 OK with booking details
- **Error Responses:**
 - 404 Not Found - Booking not found or unauthorized
 - 500 Internal Server Error

Cancel Booking

- **Endpoint:** DELETE /bookings/:id
- **Authentication:** Required (Standard User only)
- **Description:** Cancels a booking (must belong to current user)
- **Success Response:** 204 No Content
- **Error Responses:**
 - 404 Not Found - Booking not found
 - 500 Internal Server Error

User: Get My Bookings

- **Endpoint:** GET /users/bookings
- **Authentication:** Required (Standard User only)
- **Description:** Returns list of bookings for current user
- **Success Response:** 200 OK with array of bookings
- **Error Response:** 500 Internal Server Error

Models

User

```
interface User {  
  _id: string;  
  name: string;  
  email: string;  
  password: string;  
  role: "System Admin" | "Organizer" | "Standard User";  
  createdAt: Date;  
  updatedAt: Date;  
}
```

Event

```
interface Event {  
  _id: string;  
  title: string;  
  description: string;  
  date: Date;  
  location: string;  
  category: string;  
  ticketPrice: number;  
  ticketAvailable: number;  
  ticketSold: number;  
  organizer: User | string; // Reference to User  
  status: "pending" | "approved" | "rejected";  
}
```



```
    createdAt: Date;  
    updatedAt: Date;  
}
```

Booking

```
interface Booking {  
  _id: string;  
  user: User | string; // Reference to User  
  event: Event | string; // Reference to Event  
  numberOfTickets: number;  
  totalPrice: number;  
  status: "confirmed" | "cancelled";  
  createdAt: Date;  
  updatedAt: Date;  
}
```