

# Event Management API Documentation

## Overview

The Event Management API is a RESTful service built with Node.js and Express, connected to a MongoDB database for handling events. The API provides various endpoints to create, read, update, and delete events, as well as manage event-specific data like attendees, schedules, and categories.

## API Endpoints

### 1. POST /api/v3/app/events

**Description:** Create a new event.

**Response:** Returns a success message and the newly created event.

### 2. GET /api/v3/app/events

**Description:** Get all events.

**Query Parameters (optional):**

- type: Filter by event type.
- limit: Limit the number of events returned.
- page: The page number for pagination.

**Response:** Returns an array of events.

### 3. GET /api/v3/app/events/:id

**Description:** Get a specific event by its ID.

**URL Parameter:**

- id: The ID of the event.

**Response:** Returns the event details.

### 4. PUT /api/v3/app/events/:id

**Description:** Update an event by its ID.

**URL Parameter:**

- id: The ID of the event to be updated.

**Request Body (JSON):** Any updated field(s) for the event.

**Response:** Returns the updated event.

## 5. DELETE /api/v3/app/events/:id

**Description:** Delete an event by its ID.

**URL Parameter:**

- id: The ID of the event to be deleted.

**Response:** Returns a success message after the event is deleted.

### Data Model

The following object data model is used when creating or updating events:

```
{  
  "type": "event",  
  "uid": 18,  
  "name": "Name of the event",  
  "tagline": "A proper tag-line for the event",  
  "schedule": "Timestamp (Date + time)",  
  "description": "String",  
  "files": {  
    "image": "Image file (File upload)"  
  },  
  "moderator": "A user who is going to host",  
  "category": "Category of the event",  
  "sub_category": "Sub category",  
  "rigor_rank": "Integer value",  
  "attendees": ["Array of user Id's who are attending the event"]  
}
```