1. **Admin Event API**

Endpoint: /api/admin/events/

Allowed Methods: GET, POST, PUT, PATCH, DELETE

Description: This API allows administrators to perform CRUD operations on events.

- **GET /api/admin/events/** - Retrieve a list of events.
    - Parameters:
        - **search** (optional): Search in title and event type.
        - **ordering** (optional): Order by price in ascending or descending order.
    - Headers:
        - **Authorization**: Token <access_token>
    - Response:
        - 200: List of events with details.
- **POST /api/admin/events /** - Create a new event.
    - Headers:
        - **Authorization**: Token <access_token>
    - Body:
        - Event details (title, event_type, price, etc.)
    - Response:
        - 200: Created event details.
        - 400: **Fields required validation**
- **GET /api/admin/events/{pk}/** - Retrieve details of a specific event and ticket summary.
    - Parameters:
        - **pk**: ID of the event.
    - Headers:
        - **Authorization**: Token <access_token>
    - Response: Event details.
- **PUT /api/admin/events /{pk}/** - Update details of a specific event.
    - Parameters:
        - **pk**: ID of the event.
    - Headers:

- **Authorization**: Token <access_token>
- Body:
    - Updated event details (title, event_type, price, etc.)
- Response: Updated event details.
    - 200: Created event details.
    - 400: **Fields required validation**
- **PATCH /api/admin/events/{pk}/** - Partially update details of a specific event.
    - Parameters:
        - **pk**: ID of the event.
    - Headers:
        - **Authorization**: Token <access_token>
    - Body:
        - Updated event details (title, event_type, price, etc.)
    - Response: Updated event details.
        - 200: Created event details.
        - 400: **Fields options validation**
- **DELETE /api/admin/events/{pk}/** - Delete a specific event.
    - Parameters:
        - **pk**: ID of the event.
    - Headers:
        - **Authorization**: Token <access_token>
    - Response: No content.

2. **Event List and single object for User**

   Endpoint: /api/events/

   Allowed Methods: GET

   Description: This API allows customers to view a list of events.

   - **GET /api/events/** - Retrieve a list of viewable events.
       - Parameters:
           - **search** (optional): Search in title and event type.
           - **ordering** (optional): Order by price in ascending or descending order.
       - Response:

- 200: List of viewable events with details.

3. **Tickets API for User**

Endpoint: /api/tickets/

Allowed Methods: GET, POST, DELETE

Description: This API allows customers to manage their event tickets.

- **GET /api/tickets/** - Retrieve a list of tickets owned by the authenticated user.
  - Headers:
    - **Authorization**: Token <access_token>
  - Response: List of tickets with details.
- **GET /api/tickets/{pk}/** - Retrieve details of a specific ticket owned by the authenticated user.
  - Parameters:
    - **pk**: ID of the ticket.
  - Headers:
    - **Authorization**: Token <access_token>
  - Response: Ticket details.
- **POST /api/tickets/** - Purchase a ticket for an event.
  - Headers:
    - **Authorization**: Token <access_token>
  - Body:
    - Event ID (event)
  - Response:
    - 200: Created ticket details.
    - 400: Validation Error
      1. Booking window error
      2. Seat unavailable error
- **DELETE /api/tickets/{pk}/** - Cancel a specific ticket owned by the authenticated user.
  - Parameters:
    - **pk**: ID of the ticket.
  - Headers:
    - **Authorization**: Token <access_token>

- Response:
  - 200: No content.