



```
# YAML file encompasses three endpoints: Movie Information, User Account
Management, Booking
#This YAML file includes:
#- Basic information: Includes version, "info" section with a description, and
"paths" section with relevant endpoints.
#- GET: Both movie information and user profile endpoints follow the format.
#- POST: The user creation endpoint demonstrates a POST request with a request
body and various responses.
#- DELETE: Both user account deletion and booking cancellation endpoints are
included, showcasing DELETE requests with appropriate responses.
```

```
openapi: 3.0.0
```

```
info:
```

```
  title: Ensemble Cinemas API
```

```
  version: 1.0.0
```

```
  description: This API provides access to movie information, user account
management, and booking functionalities.
```

```
servers:
```

```
  - url: https://ensemble-cinemas-api.com/api/v1
```

```
# Movie information endpoints
```

```
  /movies:
```

```
    get:
```

```
      summary: View movie catalog in application
```

```
      responses:
```

```
        '200':
```

```
          description: List of available movies currently playing in the theater
          content:
```

```
            application/json:
```

```
              schema:
```

```
                type: array
```

```
                items:
```

```
                  type: object
```

```
                  properties:
```

```
                    title:
```

```
                      type: string
```

```
                      description: Movie title
```

```
                    description:
```

```
                      type: string
```

```
                      description: Movie description
```

```
                    genre:
```

```
                      type: string
```

```
                      description: Movie genre
```

```
                    releaseDate:
```

```
                      type: string
```

```

        format: date
        description: Movie release date
    '404':
        description: Not Found

# User account management endpoints

/users:
  post:
    summary: Create a new user account
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              username:
                type: string
                description: Username
              email:
                type: string
                format: email
                description: User email address
              password:
                type: string
                description: User password
    responses:
      '201':
        description: User created successfully
      '400':
        description: Bad request (invalid data)
      '409':
        description: Conflict (username already exists)

  delete:
    summary: Delete user account (requires authentication)
    responses:
      '204':
        description: User account deleted successfully
      '401':
        description: Unauthorized (requires authentication)
      '403':
        description: Forbidden (user cannot delete other accounts)

/users/login:
  post:

```

```
summary: User login
requestBody:
  required: true
  content:
    application/json:
      schema:
        type: object
        properties:
          username:
            type: string
            description: Username
          password:
            type: string
            description: User password
responses:
  '200':
    description: Login successful (returns an authentication token)
    content:
      application/json:
        schema:
          type: object
          properties:
            token:
              type: string
              description: Authentication token
  '401':
    description: Unauthorized (invalid credentials)
```

```
/users/me:
  get:
    summary: Get user profile information
    responses:
      '200':
        description: User profile information
        content:
          application/json:
            schema:
              type: object
              $ref: '#/components/responses/user_profile'
      '401':
        description: Unauthorized (requires authentication)
      '404':
        description: User not found
```

Booking endpoints

```
/bookings/{bookingId}:
```

```
delete:
  summary: Cancel a booking (requires authentication)
  responses:
    '200':
      description: Booking cancelled successfully
      content:
        application/json:
          schema:
            type: object
            properties:
              message:
                type: string
                description: Cancellation confirmation message
    '401':
      description: Unauthorized (requires authentication)
    '403':
      description: Forbidden (user cannot cancel other bookings)
    '404':
      description: Booking not found
```

components:

schemas:

```
user_profile:
  type: object
  required:
    - username
    - password
  properties:
    username:
      type: string
    location:
      type: string
    phone:
      type: string
    membership:
      type: string
```

responses:

```
Unauthorized:
  description: Invalid credentials
```

```
BadRequest:
  description: Invalid request
```