

# **Book API**

This API provides endpoints to manage books.

Base URL: **/books**

---

GET /

**Description:** Retrieve a list of books.

- **Request:**
    - Method: **GET**
    - Authentication: None required
    - Query Parameters: (Optional) Parameters for filtering books
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Array of books
- 

POST /

**Description:** Add a new book.

- **Request:**
  - Method: **POST**
  - Authentication: Required (Admin)
  - Request Body: Book details
- **Response:**
  - Status Code: **200 OK**

- Content Type: **application/json**
  - Body: New book details
- 

PUT /

**Description:** (Not Supported)

- **Response:**
    - Status Code: **403 Forbidden**
    - Body: Message indicating PUT operation not supported
- 

DELETE /

**Description:** (Not Supported)

- **Response:**
    - Status Code: **403 Forbidden**
    - Body: Message indicating DELETE operation not supported
- 

GET /:bookId

**Description:** Retrieve details of a specific book.

- **Request:**
  - Method: **GET**
  - Authentication: None required
  - Path Parameter: **bookId** - ID of the book
- **Response:**
  - Status Code: **200 OK**

- Content Type: **application/json**
  - Body: Details of the requested book
- 

## POST **/:bookId**

**Description:** (Not Supported)

- **Response:**
    - Status Code: **403 Forbidden**
    - Body: Message indicating POST operation not supported
- 

## PUT **/:bookId**

**Description:** Update details of a specific book.

- **Request:**
    - Method: **PUT**
    - Authentication: Required (Admin)
    - Path Parameter: **bookId** - ID of the book
    - Request Body: Updated book details
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Updated details of the book
- 

## DELETE **/:bookId**

**Description:** Delete a specific book.

- **Request:**

- Method: **DELETE**
- Authentication: Required (Admin)
- Path Parameter: **bookId** - ID of the book
- **Response:**
  - Status Code: **200 OK**
  - Content Type: **application/json**
  - Body: { **"\_id": "<bookId>", "success": true** }

## Issue API

This API provides endpoints to manage book issues.

Base URL: **/issues**

---

GET /

**Description:** Retrieve a list of book issues.

- **Request:**
    - Method: **GET**
    - Authentication: Required (User/Admin)
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Array of book issues
-

## POST /

**Description:** Issue a book to a student.

- **Request:**
    - Method: **POST**
    - Authentication: Required (Admin)
    - Request Body: Issue details (book, student)
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Details of the issued book
- 

## PUT /

**Description:** (Not Supported)

- **Response:**
    - Status Code: **403 Forbidden**
    - Body: Message indicating PUT operation not supported
- 

## DELETE /

**Description:** Remove all book issues (for testing).

- **Request:**
  - Method: **DELETE**
  - Authentication: Required (Admin)
- **Response:**
  - Status Code: **200 OK**

- Content Type: **application/json**
  - Body: Success message
- 

## GET **/student/**

**Description:** Retrieve a list of issues for the currently authenticated student.

- **Request:**
    - Method: **GET**
    - Authentication: Required (User)
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Array of book issues for the student
- 

## GET **/:issuelId**

**Description:** Retrieve details of a specific book issue.

- **Request:**
  - Method: **GET**
  - Authentication: Required (User/Admin)
  - Path Parameter: **issuelId** - ID of the issue
- **Response:**
  - Status Code: **200 OK**
  - Content Type: **application/json**
  - Body: Details of the requested book issue

---

## POST **/:issueId**

**Description:** (Not Supported)

- **Response:**
  - Status Code: **403 Forbidden**
  - Body: Message indicating POST operation not supported

---

## PUT **/:issueId**

**Description:** Mark a book issue as returned.

- **Request:**
  - Method: **PUT**
  - Authentication: Required (Admin)
  - Path Parameter: **issueId** - ID of the issue
- **Response:**
  - Status Code: **200 OK**
  - Content Type: **application/json**
  - Body: Updated details of the returned book issue

---

## DELETE **/:issueId**

**Description:** (Not Supported)

- **Response:**
  - Status Code: **403 Forbidden**
  - Body: Message indicating DELETE operation not supported

# User API

This API provides endpoints for user authentication and management.

Base URL: **/users**

---

## OPTIONS /

**Description:** Enable pre-flight CORS request.

- **Request:**
    - Method: **OPTIONS**
  - **Response:**
    - Status Code: **200 OK**
    - Body: Empty
- 

## GET /

**Description:** Retrieve a list of all users (Admin access).

- **Request:**
    - Method: **GET**
    - Authentication: Required (User/Admin)
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Array of user details
- 

## PUT /:userId



**Description:** Update user details (Admin access).

- **Request:**
    - Method: **PUT**
    - Authentication: Required (User)
    - Path Parameter: **userId** - ID of the user to update
    - Request Body: Updated user details
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Updated user details
- 

**PUT /password/:userId**

**Description:** Change user password.

- **Request:**
    - Method: **PUT**
    - Authentication: Required (User)
    - Path Parameter: **userId** - ID of the user to update password
    - Request Body: New password
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Success message
-

## POST /signup

**Description:** Register a new user.

- **Request:**
    - Method: **POST**
    - Request Body: User details for registration
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Registration success message
- 

## POST /login

**Description:** Login and generate an authentication token.

- **Request:**
    - Method: **POST**
    - Authentication: Local authentication (username and password)
    - Request Body: User login credentials
  - **Response:**
    - Status Code: **200 OK**
    - Content Type: **application/json**
    - Body: Authentication token and user information
- 

## GET /logout

**Description:** Logout and destroy the session.

- **Request:**
    - Method: **GET**
    - Authentication: Required (User)
  - **Response:**
    - Redirect to /
- 

## GET /checkJWTtoken

**Description:** Check the validity of a JWT token.

- **Request:**
  - Method: **GET**
  - Authentication: Required (User)
- **Response:**
  - Status Code: **200 OK**
  - Content Type: **application/json**
  - Body: Token validation status and user information