Library Management System

Table of Contents

---

1. Project Overview

The **Library Management System** is a Spring Boot-based RESTful API that allows users to manage books, patrons, and borrowing records within a library. The application implements robust features such as authentication using JWT, logging with Aspect-Oriented Programming (AOP), caching for performance optimization, and declarative transaction management to ensure data integrity.

---

2. Running the Application:

- **API Base URL:** `http://localhost:5545`

---

3. API Endpoints

The API follows RESTful principles and is secured using JWT-based authentication. Below is a comprehensive list of available endpoints categorized by their functionality.

Authentication

Method	Endpoint	Description
POST	/api/auth/login	Authenticate user and obtain JWT token
POST	/api/auth/register	Register a new user

Books

Method	Endpoint	Description
GET	/api/books	Retrieve a list of all books
GET	/api/books/{id}	Retrieve a book by its ID
POST	/api/books	Add a new book
PUT	/api/books/{id}	Update an existing book
DELETE	/api/books/{id}	Delete a book by its ID

Patrons

Method	Endpoint	Description
GET	/api/patrons	Retrieve a list of all patrons
GET	/api/patrons/{id}	Retrieve a patron by its ID
POST	/api/patrons	Add a new patron
PUT	/api/patrons/{id}	Update an existing patron
DELETE	/api/patrons/{id}	Delete a patron by its ID

Borrowing

Method	Endpoint	Description
POST	/api/borrow/{bookId}/patron/{patronId}	Borrow a book for a patron
PUT	/api/return/{bookId}/patron/{patronId}	Return a borrowed book

---

4. Authentication

The API uses JWT (JSON Web Tokens) for securing endpoints. Users must authenticate to obtain a JWT token, which must be included in the `Authorization` header for protected endpoints.

Obtaining a JWT Token

1. **Register a New User**

   Endpoint: POST http://localhost:5545/api/auth/register

   Request Body:

   ```
   {
       "username": "john_doe",
       "password": "SecureP@ssw0rd"
   }
   ```

   Response:

   ```
   {
       "message": "User registered successfully"
   }
   ```

2. **Login**

   Endpoint: POST http://localhost:5545/api/auth/login

   Request Body:

   ```
   {
       "username": "john_doe",
       "password": "SecureP@ssw0rd"
   }
   ```

   Response:

   ```
   {
       "jwt": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
   }
   ```

   Save the `jwt` token returned in the response for authenticating subsequent requests.

Using the JWT Token

Include the JWT token in the `Authorization` header of your HTTP requests to access protected endpoints.

Header Example:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Note: Replace `eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...` with your actual JWT token.