

# Application Documentation

## Introduction

This system works to reserve books and record the time for returning them, in addition to recording information about books and patrons we use springboot framework to develop the hole system with aspect log to trace all request in system also we use caching to improve the performance for the system

## Prerequisites

First of all we use java 17 with Spring Boot framework

Also we use maven build tool for managing dependencies and building our Spring Boot project.

Database : we use MySQL database for storing information

Testing Frameworks :we use Mockito and SpringBootTest for testing.

## API Endpoints

The application exposes the following API endpoints:

Book management endpoints:

- GET /api/books: Retrieve a list of all books.
- GET /api/books/{id}: Retrieve details of a specific book by ID.
- POST /api/books: Add a new book to the library.
- PUT /api/books/{id}: Update an existing book's information.
- DELETE /api/books/{id}: Remove a book from the library.

Patron management endpoints:

- GET /api/patrons: Retrieve a list of all patrons.
- GET /api/patrons/{id}: Retrieve details of a specific patron by ID.
- POST /api/patrons: Add a new patron to the system.
- PUT /api/patrons/{id}: Update an existing patron's information.
- DELETE /api/patrons/{id}: Remove a patron from the system.

Borrowing endpoints:

- POST /api/borrow/{bookId}/patron/{patronId}: Allow a patron to borrow a book.
- PUT /api/return/{bookId}/patron/{patronId}: Record the return of a borrowed book by a patron

- Auth endpoints:
- POST /register:to make new user .
- POST /login:to get the token for this user.

## Authentication

follow these steps:

### Registration:

- Use the `/register` endpoint to create a new user account.
- Provide the required information, username and password.

### Login:

- Access the `/login` endpoint with your credentials(username,password).
- Retrieve the authentication token from the response.

### Authenticated Requests:

- Include the obtained token in the `Authorization` header for authenticated requests.
- Example: `Authorization: Bearer [your_token]`

## System Package

1-Aspect Package: contain the whole aspect log classes such as Aopconfig and LoggingAspect

2-Controller Package: contain the controller classes in system such as bookController , Patron controller and BorrowingRecordController

3-Entity Package: contain the entity classes such as book ,patron, BorrowingRecord and user

4-Exception: contain classes to handle exception such as

GlobalExceptionHandler :to handle the exception in the system

5-Repo Package : contain all Repository interfaces

6-security Package : contain all security classes

7-service Package: contain all services classes

8-serviceInterface: contain all services interfaces