

Documentation

Ahmed Emad Fathy Mohamed Elhofy

August 1, 2024

Introduction

In this project "library_management_system" I have built a Library Management System API using Spring Boot. The system allows librarians to manage books, patrons, and borrowing records. Also, the system has validation as the user can't add a book without the title or the ISBN or the author name as if the user enter the author name as numbers the system will throw a message and it will show a message that this is required.

In this project, I have applied aspects to log method calls, exceptions, and performance metrics of certain operations like book additions and updates.

All the Endpoints are tested and handled to any thing a user would enter

To run the Project:

first clone the repo, then go to the main file "LibrarySystemApplication.java" and run then go to postman as i am explaining in this documentation

Note that o am working on Mysql workbench and the password as it shown in the code is ahmed123456

Contents

1	BookController	3
2	BorrowingRecordController	4
3	PatronController	5

1 BookController

- Get All Books
 - Endpoint: /api/books
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to GET.
 - * Enter the URL: `http://localhost:8080/api/books`
 - * Click Send.
- Get Book by ID
 - Endpoint: /api/books/{id}
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to GET.
 - * Enter the URL: `http://localhost:8080/api/books/1`
 - * Replace 1 with the actual id of the book you want to retrieve.
 - * Click Send.
- Create a Book
 - Endpoint: /api/books
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to POST.
 - * Enter the URL: `http://localhost:8080/api/books`
 - * Go to the Body tab.
 - * Select raw and JSON from the dropdown.
 - * Enter the JSON body:

```
{  
  "title": "Book Title",  
  "author": "Author Name",  
  "isbn": "1234567890"  
}
```
 - * Click Send.
- Update a Book
 - Endpoint: /api/books/{id}
 - Example Postman Request

- * Open Postman.
 - * Set the request method to PUT.
 - * Enter the URL: `http://localhost:8080/api/books/1`
 - * Replace 1 with the actual id of the book you want to update.
 - * Go to the Body tab.
 - * Select raw and JSON from the dropdown.
 - * Enter the JSON body:


```
{
  "title": "Updated Title",
  "author": "Updated Author",
  "isbn": "0987654321"
}
```
 - * Click Send.
- Delete a Book
 - Endpoint: `/api/books/{id}`
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to DELETE.
 - * Enter the URL: `http://localhost:8080/api/books/1`
 - * Replace 1 with the actual id of the book you want to delete.
 - * Click Send.

2 BorrowingRecordController

- Borrow a Book
 - Endpoint: `/api/borrow/{bookId}/patron/{patronId}`
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to POST.
 - * Enter the URL: `http://localhost:8080/api/borrow/1/patron/1`
 - * Replace 1 with the actual bookId and patronId.
 - * Click Send.
- Return a Book
 - Endpoint: `/api/borrow/return/{bookId}/patron/{patronId}`
 - Example Postman Request
 - * Open Postman.

- * Set the request method to PUT.
- * Enter the URL: `http://localhost:8080/api/borrow/return/1/patron/1`
- * Replace 1 with the actual bookId and patronId.
- * Click Send.

3 PatronController

- Get All Patrons
 - Endpoint: `/api/patrons`
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to GET.
 - * Enter the URL: `http://localhost:8080/api/patrons`
 - * Click Send.
- Get Patron by ID
 - Endpoint: `/api/patrons/{id}`
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to GET.
 - * Enter the URL: `http://localhost:8080/api/patrons/1`
 - * Replace 1 with the actual id of the patron you want to retrieve.
 - * Click Send.
- Create a Patron
 - Endpoint: `/api/patrons`
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to POST.
 - * Enter the URL: `http://localhost:8080/api/patrons`
 - * Go to the Body tab.
 - * Select raw and JSON from the dropdown.
 - * Enter the JSON body:


```
{
    "name": "Patron Name",
    "email": "patron@example.com"
}
```
 - * Click Send.

- Update a Patron
 - Endpoint: `/api/patrons/{id}`
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to PUT.
 - * Enter the URL: `http://localhost:8080/api/patrons/1`
 - * Replace 1 with the actual id of the patron you want to update.
 - * Go to the Body tab.
 - * Select raw and JSON from the dropdown.
 - * Enter the JSON body:

```
{
  "name": "Updated Patron Name",
  "email": "updated@example.com"
}
```
 - * Click Send.
- Delete a Patron
 - Endpoint: `/api/patrons/{id}`
 - Example Postman Request
 - * Open Postman.
 - * Set the request method to DELETE.
 - * Enter the URL: `http://localhost:8080/api/patrons/1`
 - * Replace 1 with the actual id of the patron you want to delete.
 - * Click Send.