Library Management System Documentation

Introduction

The Library Management System (LMS) is a software application built on Spring Boot, designed to automate library management tasks such as book cataloging, patron management, and borrowing records.

Requirements

- Java Development Kit (JDK) 8 or higher

-IntelliJ editor.

- install workbench or xampp for MySQL database

-in my .rar file you will find file called "SQL script", you must run this file in workbench or xampp server.

-the name of database schema is LMS.

Database connection properties

```
spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/LMS
spring.datasource.username=root4
spring.datasource.password=root4
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

The Library Management System provides the following API endpoints:

 -Books 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.
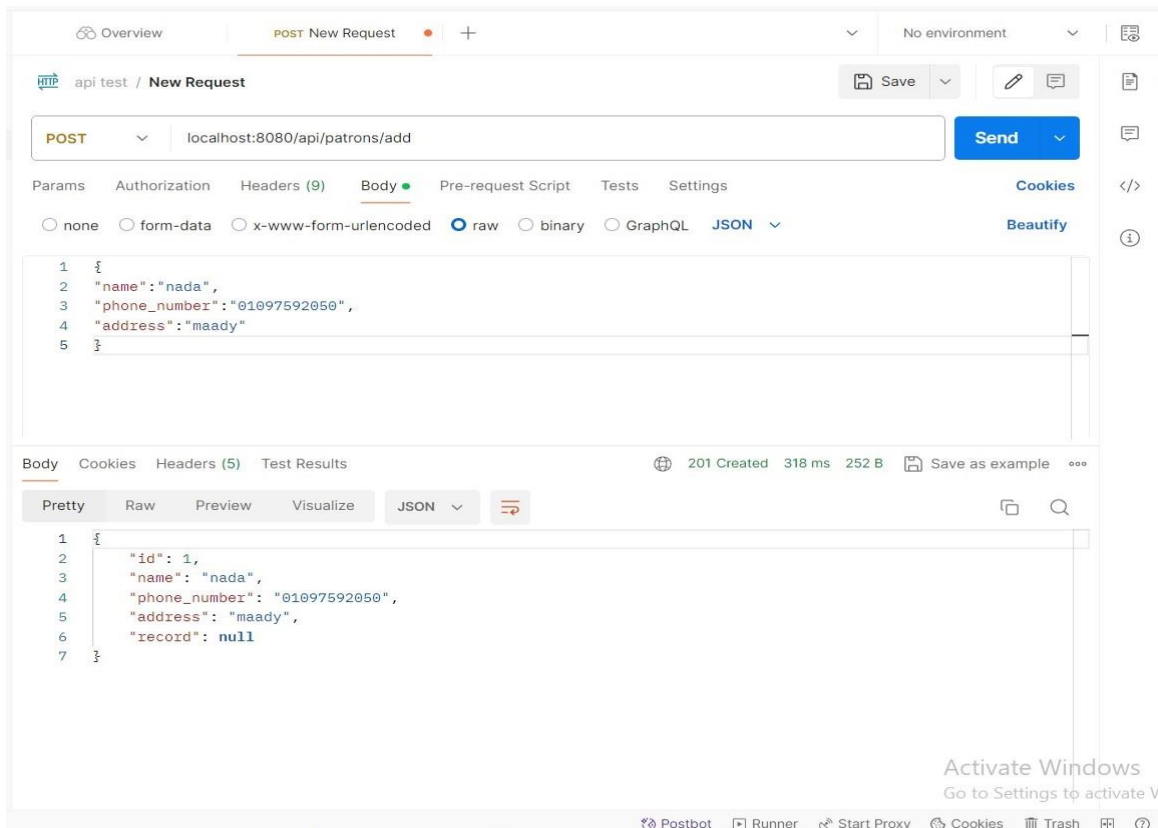
-Patron Management Endpoints:

  - 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.

Some snapshot for testing API:

No environment ∨

api test / **New Request**   🖫 Save ∨   ✎ 🗩

| PUT ∨ | localhost:8080/api/patrons/1 | **Send** ∨ |

Params   Authorization   Headers (9)   Body ●   Pre-request Script   Tests   Settings   **Cookies**

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL   JSON ∨   **Beautify**

```
1  {
2  "name":"noor",
3  "address":"dar al salam",
4  "phone_number":"01097592080"
5  }
6
```

Body   Cookies   Headers (5)   Test Results   🌐 200 OK   144 ms   254 B   🖫 Save as example   •••

Pretty   Raw   Preview   Visualize   JSON ∨

```
1  {
2      "id": 1,
3      "name": "noor",
4      "phone_number": "01097592080",
5      "address": "dar al salam",
6      "record": null
7  }
```

No environment ∨

api test / **New Request**   🖫 Save ∨   ✎ 🗩

| GET ∨ | localhost:8080/api/patrons/1 | **Send** ∨ |

Params   Authorization   Headers (7)   Body   Pre-request Script   Tests   Settings   **Cookies**

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

Body   Cookies   Headers (5)   Test Results   🌐 200 OK   15 ms   252 B   🖫 Save as example   •••

Pretty   Raw   Preview   Visualize   JSON ∨

```
1  {
2      "id": 1,
3      "name": "noor",
4      "phone_number": "01097592080",
5      "address": "dar al salam",
6      "record": []
7  }
```

HTTP api test / **New Request**   💾 Save ∨   ✏️ 💬

PUT ∨   localhost:8080/api/borrow/1/patron/4   **Send** ∨

Params   Authorization   Headers (9)   Body ●   Pre-request Script   Tests   Settings   **Cookies**

○ none   ○ form-data   ○ x-www-form-urlencoded   ● raw   ○ binary   ○ GraphQL   JSON ∨   **Beautify**

```
1  {
2  "return_date":"2024-04-05"
3  }
```

Body   Cookies   Headers (5)   Test Results   🌐 404 Not Found   487 ms   227 B   💾 Save as example   ⋯

Pretty   Raw   Preview   Visualize   JSON ∨

```
1  {
2      "message": "not found id of patron:4",
3      "status_code": 500
4  }
```

HTTP api test / **New Request**   💾 Save ∨   ✏️ 💬

PUT ∨   localhost:8080/api/borrow/1/patron/1   **Send** ∨

Params   Authorization   Headers (9)   Body ●   Pre-request Script   Tests   Settings   **Cookies**

○ none   ○ form-data   ○ x-www-form-urlencoded   ● raw   ○ binary   ○ GraphQL   JSON ∨   **Beautify**

```
1  {
2  "return_date":"2024-04-09"
3  }
```

Body   Cookies   Headers (5)   Test Results   🌐 200 OK   295 ms   196 B   💾 Save as example   ⋯

Pretty   Raw   Preview   Visualize   Text ∨

```
1  Return date updated successfully
```

Overview | PUT New Request | + | No environment

api test / New Request — Save

PUT `localhost:8080/api/books/1` — **Send**

Params | Authorization | Headers (9) | Body • | Pre-request Script | Tests | Settings — Cookies

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON ∨ — Beautify

```
1  {
2  "isbn":"123-234-5",
3  "author":"mostafa",
4  "name":"science",
5  "publication_year":2008,
6  "title":"title2"
7  }
```

Body | Cookies | Headers (5) | Test Results — 🌐 200 OK 1171 ms 282 B 💾 Save as example ⋯

Pretty | Raw | Preview | Visualize — JSON ∨

```
1  {
2      "id": 1,
3      "name": "science",
4      "title": "title2",
5      "author": "mostafa",
6      "publication_year": 2008,
7      "record": null,
8      "isbn": "123-234-5"
9  }
```

----

api test / New Request — Save

GET `localhost:8080/api/books/1` — **Send**

Params | Authorization | Headers (7) | Body | Pre-request Script | Tests | Settings — Cookies

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

Body | Cookies | Headers (5) | Test Results — 🌐 200 OK 177 ms 280 B 💾 Save as example ⋯

Pretty | Raw | Preview | Visualize — JSON ∨

```
1  {
2      "id": 1,
3      "name": "science",
4      "title": "title2",
5      "author": "mostafa",
6      "publication_year": 2008,
7      "record": [],
8      "isbn": "123-234-5"
9  }
```