# Library Management System using Spring Boot Documentation

[Library Management System using Spring Boot Documentation 1](#_Toc174630244)

[Prerequisites : 1](#_Toc174630245)

[Setting Up the Database : 2](#_Toc174630246)

[Create a MySQL Database**:** 2](#_Toc174630247)

[Create the necessary tables**:** 2](#_Toc174630248)

[Configuration : 2](#_Toc174630249)

[Building the Application: 2](#_Toc174630250)

[Open the Project**:** 2](#_Toc174630251)

[Build the Project**:** 2](#_Toc174630252)

[Running the Application: 3](#_Toc174630253)

[Using Your IDE**:** 3](#_Toc174630254)

[Interacting with API Endpoints: 3](#_Toc174630255)

[Base URL**:** 3](#_Toc174630256)

[API Endpoints**:** 3](#_Toc174630257)

[Testing API Endpoints: 4](#_Toc174630258)

[**Postman**: 4](#_Toc174630259)

[Stopping the Application: 4](#_Toc174630260)

[**Using IDE**: 4](#_Toc174630261)

[**Using Command Line**: 4](#_Toc174630262)

[Logs & Troubleshooting: 4](#_Toc174630263)

[**Application Logs**: 4](#_Toc174630264)

[**Common Issues:** 4](#_Toc174630265)

## 

## Prerequisites :

Before running the application, ensure you have the following installed on your machine:

* Java Development Kit (JDK) 8 or higher
* Maven (for building the project)
* MySQL (for the database)
* Spring Tool Suite (STS) or any IDE that supports Spring Boot (e.g., IntelliJ IDEA)

## Setting Up the Database :

### **Create a MySQL Database:**

* Open MySQL Workbench or your preferred MySQL client.
* Run the following command to create a new database:

CREATE DATABASE library\_management\_system;

### **Create the necessary tables:**

* This is my file sql for creation tables, there columns and relations between them.



* This is my ERD Structure :



## Configuration :

Update application.properties:

In the src/main/resources/application.properties file, configure the database connection:

spring.datasource.url=jdbc:mysql://localhost:3305/library\_management\_system

spring.datasource.username=your-username

spring.datasource.password=your-password

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update

spring.jpa.show-sql=true

Replace your-username and your-password with your MySQL credentials.

## Building the Application:

### **Open the Project:**

Import the project into your IDE (e.g., Spring Tool Suite or IntelliJ IDEA).

### **Build the Project:**

* Open a terminal or use the IDE terminal.
* Navigate to the root directory of your project.
* Run the following command to build the project:

mvn clean install

## Running the Application:

### **Using Your IDE:**

* Locate the main application class (usually in src/main/java/com/yourpackage/LibraryManagementSystemApplication.java).
* Right-click the class and select "Run as Spring Boot Application" or use the IDE's run option.

## Interacting with API Endpoints:

### **Base URL:**

The base URL for your API will be <http://localhost:8080>.

### **API Endpoints:**

Here are the available endpoints:

#### **Books:**

* **GET** /api/books: Retrieve a list of all books.
* **POST** /api/books: Add a new book.
* **GET** /api/books/{id}: Retrieve a specific book by its ID.
* **PUT** /api/books/{id}: Update a specific book by its ID.
* **DELETE** /api/books/{id}: Delete a specific book by its ID.

#### **Patrons:**

* **GET** /api/patrons: Retrieve a list of all patrons.
* **POST** /api/patrons: Add a new patron.
* **GET** /api/patrons/{id}: Retrieve a specific patron by their ID.
* **PUT** /api/patrons/{id}: Update a specific patron by their ID.
* **DELETE** /api/patrons/{id}: Delete a specific patron by their ID.

#### **Borrowing Records:**

* **GET** /api/borrowing-records: Retrieve a list of all borrowing records.
* **POST** /api/borrowing-records: Add a new borrowing record.
* **GET** /api/borrowing-records/{id}: Retrieve a specific borrowing record by its ID.
* **PUT** /api/borrowing-records/{id}: Update a specific borrowing record by its ID.
* **DELETE** /api/borrowing-records/{id}: Delete a specific borrowing record by its ID.

## Testing API Endpoints:

### **Postman**:

* Download and install [Postman](https://www.postman.com/).
* Use Postman to test the API endpoints by sending HTTP requests to the above URLs.
* This is my collection that tested by it all endpoints:



## **Stopping the Application:**

### **Using IDE**:

Simply stop the application from the IDE.

### **Using Command Line**:

Press Ctrl+C in the terminal where the application is running.

## **Logs & Troubleshooting:**

### **Application Logs**:

The application logs are printed in the terminal or console where the application is running. Look out for any errors or warnings.

### **Common Issues:**

* **Database Connection Issues**: Ensure the MySQL service is running and the credentials are correct in application.properties.
* **Port Already in Use**: If you get an error about port 8080 being in use, change the port by adding server.port=8080 (or any other available port) in the application.properties file.