

Case Study
On
Library Management System

INTRODUCTION

- The Project aims to automate the process of **Library Management System** and Make the process of recruitment process.
- It provides a platform where the **librarian can view, add, update, and delete books** available in the library.
- It allows the **librarian to manage student details**, including adding and viewing registered students.
- It provides information about **book availability**, helping in proper issue and return of books.

ABSTRAT

Library Management System (LMS). The main objective of the **Library Management System** is to develop software that manages library activities in a college or institution.

It provides an interactive platform where the **librarian can manage all library operations** such as book records, student details, and book issue/return activities from a single console.

The system helps in maintaining **complete details of all books** in a digital format. It allows the librarian to easily **add, update, view, and delete books**, as well as manage student information efficiently.

What we are going to build: Client Requirement

The client wants a **Library Management System** where the **admin (librarian)** can manage all library-related activities such as **books, students, and book issue/return records**.

The system should provide a centralized platform to efficiently control and track library operations.

We have to build a **simple Library Management System** with the following requirements:

- **Admin (Librarian)** should be able to **create, update, delete, and view books** in the library.

- **Students** should be able to **borrow more than one book** (as per library rules).
- The system should **track book availability** and update quantity automatically after issue and return.
- **Admin** should be able to **create, update, delete, and view student records**.
- The system should maintain **complete issue and return history** of books.
- The admin should be able to **view all issued and returned book records** at any time.

What we are going to build: Some technical terms.

- Login and Register APIs for Admin (Librarian)
- Book APIs with pagination and sorting
- Proper user input validation
- Exception handling for errors like book not found or out of stock
- Role-based security (Admin access)
- JWT-based authentication for secure APIs

What are the technologies and tools we are going use ?

- Framework: **Spring Boot** Java Framework.
 - Java 8+
 - Maven
 - STS
 - Apache Tomcat
 - Spring core, Spring security(jwt) , Spring data JPA(Hibernate) et.c
- MySQL Database.
- Postman

SYSTEM REQUIREMENTS

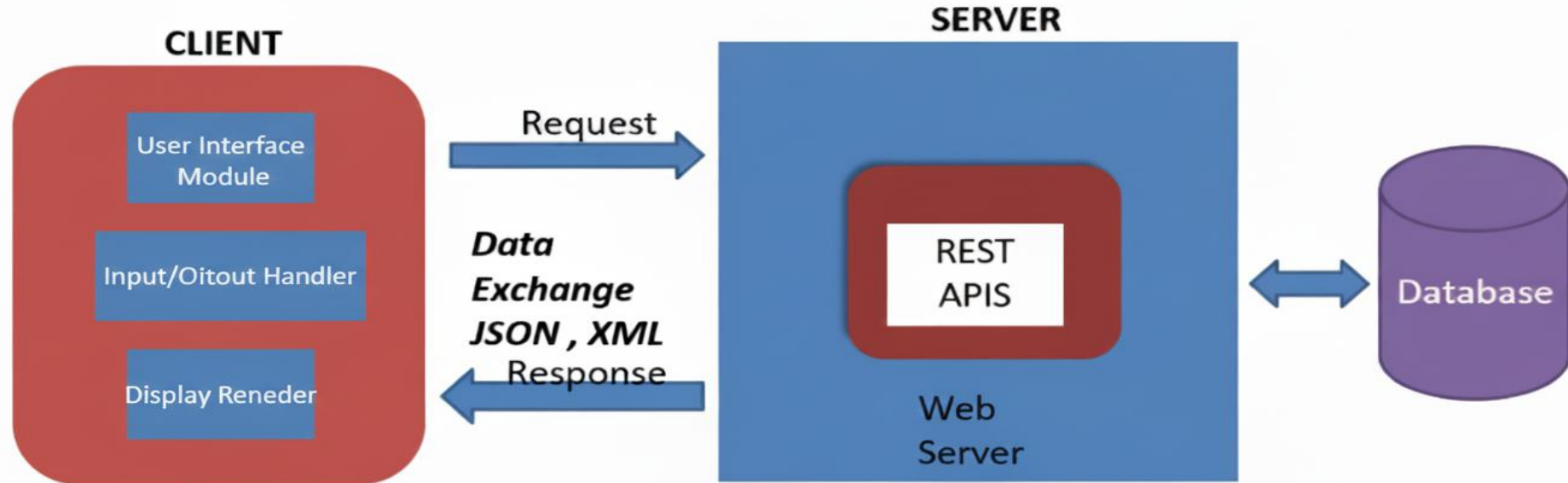
SOFTWARE REQUIREMENT :

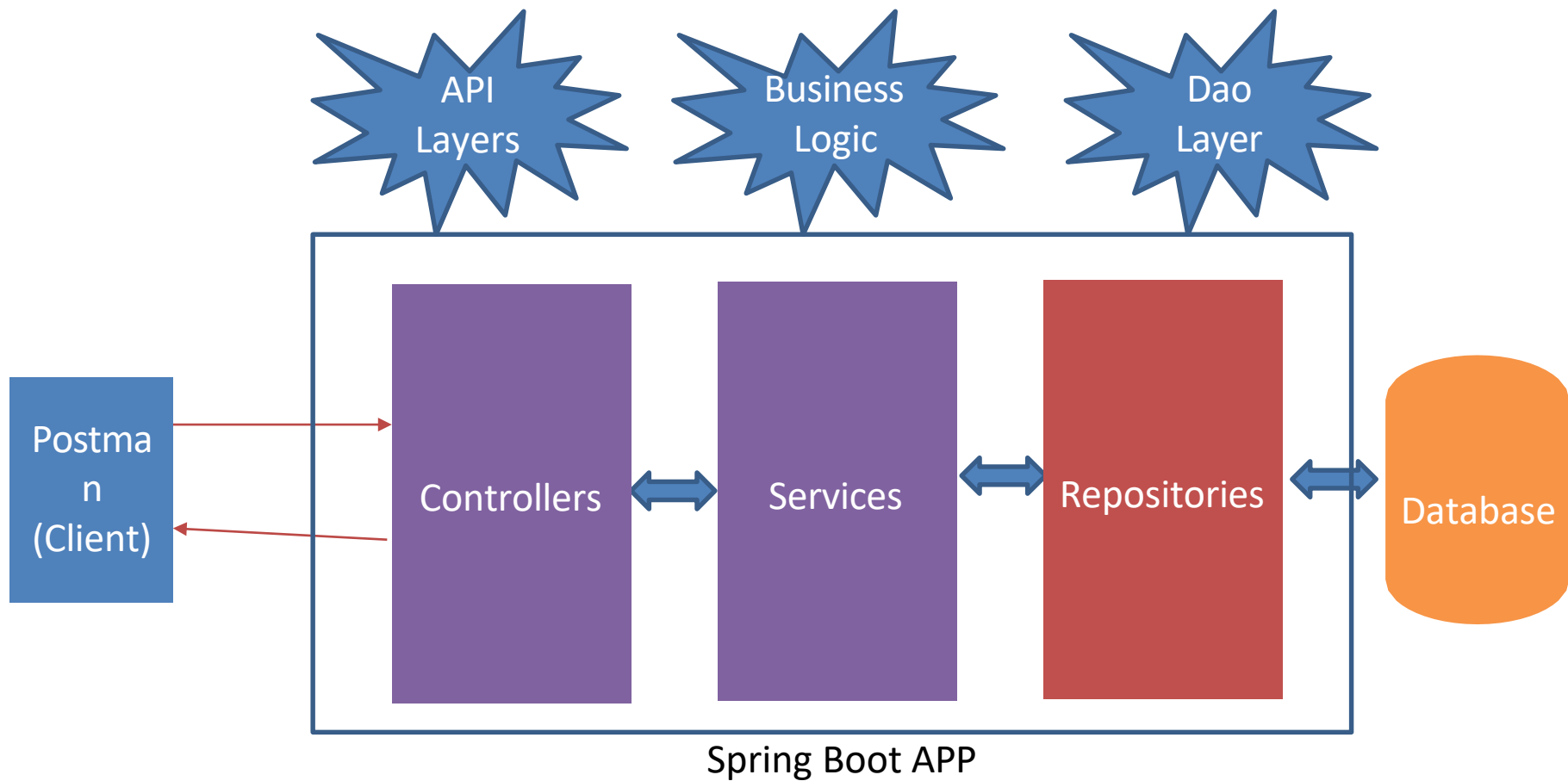
- ❑ **Back End:** Spring Boot , JPA Hibernate and Postman Tool
- ❑ **DB Server:** MySQL Server 8.0.CE
- ❑ **Web Server:** Apache Tomcat
- ❑ **IDE:** Spring Tool Suite(STS)
- ❑ **Browser:** Google Chrome

PROJECT MODULE

- ❖ **Admin (Librarian) Module**
- ❖ **Book Management Module**
- ❖ **Book Delete Module**
- ❖ **Book Addition Module**

Client – Server Architecture





ADMIN MODULE

- In Admin Module Can move after login successfully. Admin Can save their details by using Postman Tool.
- Admin module can save admin details by using post mapping, Admin can get all details by using Get Mapping , Admin can Update details by using Put Mapping , Admin can delete their details By using Delete Mapping.
- In Admin can contain Test methods we mention Save , List of Admins , update and Delete methods.
- After Completion Admin we can add the books of record to the database .

Http Request Methods

HTTP defines a set of **request methods** to indicate the desired action to be performed for a given resource

GET	http://localhost:9080/login	Return to the admin login page
GET	http://localhost:9080/books	Return to the books section
GET	http://localhost:9080/book/1	Return to the book of id 1
PUT	http://localhost:9080/update/1	Update a book of id 20
DELETE	http://localhost:9080/delete/2	Delete a book of id 2

Http
Methods



URI (Uniform Resource Identifier)

- ✓ URI is used to identify resource.

GET	http://localhost:9080/login	Return the login page of admin
GET	http://localhost:9080/book/1	Return the book of id 1
POST	http://localhost:9080/addbook/	Create a new book
PUT	http://localhost:9080/update/2	Update a book of id 2
DELETE	http://localhost:9080/category/2	Delete a category of id 2

BOOK MANAGEMENT MODULE

- After successful login, the Admin can save, retrieve, update, and delete book details by using the Postman Tool via REST API mappings (Post, Get, Put, and Delete).
- Testing and Validation:** The module contains dedicated Test methods to verify the functionality of Save, List of Books, Update, and Delete operations.
- Hierarchical Structure:** The Book Module is created as a child table after the Admin module is complete.

BOOK DELETE MODULE

- Safety Confirmation:** Implement a confirmation dialog (e.g., "Are you sure?") to prevent accidental clicks.
- Status Validation:** Ensure the book isn't currently checked out or reserved before allowing deletion.
- Soft Delete:** Instead of permanent erasure, mark the book as inactive in the database so it can be recovered if needed.
- Audit Trail:** Automatically log the date, time, and user ID of the person who performed the deletion for accountability.

BOOK UPDATE MODULE

DATA DICTIONARY

TABLES OF DATABASE

The screenshot displays the MySQL Workbench interface. The left sidebar contains navigation panels for **MANAGEMENT** (Server Status, Client Connections, Users and Privileges, Status and System Variables, Data Export, Data Import/Restore), **INSTANCE** (Startup / Shutdown, Server Logs, Options File), **PERFORMANCE** (Dashboard, Performance Reports, Performance Schema Setup), **Administration** (Schemas), and **Information** (No object selected).

The main workspace shows a query in the **Query 1** tab:

```
1 • create database library_management;
2 • use library_management;
3 • show tables from library_management;
4 • select * from author_details;
5 • select * from user;
6 • select * from authors;
7 • select * from author_details;
```

The **Result Grid** displays the results of the query:

id	email	first_name	last_name	password
13	admin@admin.in	admin	admin	\$2a\$10\$KcGMXvFYAgMsCOC9MD6sN.NO895sxq...

The **Output** panel shows the execution log:

#	Time	Action	Message	Duration / Fetch
1	22:38:23	select * from publishers LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
2	22:38:25	use library_management	0 row(s) affected	0.000 sec
3	22:38:27	show tables from library_management	13 row(s) returned	0.016 sec / 0.000 sec
4	22:38:32	select * from user LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The bottom status bar shows the system clock as 10:38 PM on 12/18/2025.

BOOKS DATABASE

MySQL Workbench

new connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information

No object selected

Object Info Session

Query 1 java* SQL File 4* SQL File 6* SQL File 7* SQL File 7* SQL File 8* SQL File 9* x SQL File 10*

Limit to 1000 rows

```
3 show tables from library_management;
4 select * from author_details;
5 select * from user;
6 select * from authors;
7 select * from author_details;
8 select * from book_details;
9 select * from books;
```

Result Grid

	id	description	isbn	name	serial_name
▶	1	Book description	AP 1287	Spring in Action	CXEF12389
	5	Description1	BP567#R	Spring Microservices	KCXEF12389
	9	description2	GH67F#	Spring Boot	UV#JH
•	books	books	books	books	books

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

books 2 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	23:07:46	select * from books LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
2	23:07:49	use library_management	0 row(s) affected	0.000 sec
3	23:07:51	select * from authors LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
4	23:08:03	select * from books LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Humid Now

Search

11:08 PM 12/18/2025

CATEGORY DATABASE

MySQL Workbench

new connection x

File Edit View Query Database Server Tools Scripting Help

Navigator: MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information

Query 1 java* SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10*

Limit to 1000 rows

```
7 • select * from author_details;
8 • select * from book_details;
9 • select * from books;
10 • select * from books_authors;
11 • select * from books_categories;
12 • select * from books_publishers;
13 • select * from categories;
```

Result Grid

id	name
3	Dummy category
7	New category
11	Spring category

categories 3 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	23:07:46	select * from books LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
2	23:07:49	use library_management	0 row(s) affected	0.000 sec
3	23:07:51	select * from authors LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
4	23:08:03	select * from books LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
5	23:08:12	select * from categories LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Humid Now

Search

11:08 PM 12/18/2025

PUBLISHER DATABASE

MySQL Workbench

new connection x

File Edit View Query Database Server Tools Scripting Help

Navigator

- MANAGEMENT
 - Server Status
 - Client Connections
 - Users and Privileges
 - Status and System Variables
 - Data Export
 - Data Import/Restore
- INSTANCE
 - Startup / Shutdown
 - Server Logs
 - Options File
- PERFORMANCE
 - Dashboard
 - Performance Reports
 - Performance Schema Setup

Administration Schemas

Information

No object selected

Object Info Session

Query 1 java* SQL File 6* SQL File 7* SQL File 7* SQL File 8* SQL File 9* SQL File 10*

Limit to 1000 rows

```
9 • select * from books;
10 • select * from books_authors;
11 • select * from books_categories;
12 • select * from books_publishers;
13 • select * from categories;
14 • select * from hibernate_sequence;
15 • select * from publishers;
```

Result Grid

	id	name
▶	4	Dummy publisher
	8	publisher2
	12	publisher3
•	HULL	HULL

publishers+ x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
❌ 1	23:07:46	select * from books LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...	0.000 sec
✅ 2	23:07:49	use library_management	0 row(s) affected	0.000 sec
✅ 3	23:07:51	select * from authors LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
✅ 4	23:08:03	select * from books LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
✅ 5	23:08:12	select * from categories LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
✅ 6	23:08:18	select * from publishers LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Humid Now

Search

11:08 PM 12/18/2025

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

POST METHOD FOR ADD DATA

The screenshot displays the Postman API client interface. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The workspace is named 'triveni garimella9990's Workspace'. The left sidebar shows 'Collections' with 'My Collection' containing 'GET Get data' and 'POST Post data'. The main panel shows a POST request to 'http://localhost:9080/category/2'. The 'Params' tab is active, showing a table with 'Key' and 'Value' columns. The 'Body' tab is active, showing an HTML response. The response status is '200 OK' with a response time of '16 ms' and a size of '2.57 KB'. The HTML content is displayed in a code editor with line numbers 64 to 81.

POST Get data • POST Post data • +

My Collection / Post data

POST http://localhost:9080/category/2

Send

Docs Params Authorization Headers (9) Body Scripts Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies (1) Headers (12) Test Results (1/1)

200 OK • 16 ms • 2.57 KB

HTML Preview Visualize

```
64 <div class="form-group">
65 <div class="row">
66 <div class="col-sm-6 col-sm-offset-3">
67 <input type="submit" name="login-submit" id="login-submit"
68 class="form-control btn btn-success" value="Log In" />
69 </div>
70 </div>
71 </div>
72 </form>
73
74 </div>
75 </div>
76 </div>
77 <div><footer class="page-footer font-small blue pt-4">
78 <!-- Copyright -->
79 <div class="footer-copyright text-center py-3">
80 © 2022 Copyright: <a href="https://www.knowledgefactory.net/">
81 www.knowledgefactory.net</a>
```

Humid Now

Search

10:57 PM 12/18/2025

GET METHOD FOR GET ALL DATA

The screenshot displays the Postman API client interface. The workspace is named "triveni garimella9990's Workspace". The selected collection is "My Collection" with the environment set to "POST Post data". The request method is "GET" and the URL is "http://localhost:9080/books". The response status is "200 OK" with a response time of 375 ms and a size of 2.57 KB. The response body is shown in HTML format, displaying a login page for a "Library Management System".

Query Params

Key	Value	Description
Key	Value	Description

Body

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <meta charset="utf-8">
6 <meta http-equiv="x-ua-compatible" content="ie=edge">
7 <title>Login</title>
8 <meta name="viewport" content="width=device-width, initial-scale=1">
9 <link rel="stylesheet"
10 | href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
11 <link rel="stylesheet"
12 | href="https://use.fontawesome.com/releases/v5.4.1/css/all.css">
13 </head>
14
15 <body>
16 <div><nav class="navbar navbar-expand-sm bg-info navbar-light">
17
18 <a class="navbar-brand" href="#">Library Management System</a>
```


PUT METHOD FOR UPDATE DATA

The screenshot displays the Postman interface for a PUT request. The top navigation bar includes 'Home', 'Workspaces', and 'API Network'. The left sidebar shows 'Collections' with 'My Collection' containing 'GET Get data' and 'POST Post data'. The main area shows the 'PUT' method with the URL 'http://localhost:9080/update/2'. The 'Params' tab is active, showing a table with one parameter: 'Key' with value 'Value'. The 'Body' tab is active, showing the response status '200 OK' and the HTML body content.

Query Params

Key	Value	Description
Key	Value	Description

Body

200 OK • 15 ms • 2.57 KB

```
54 class= "form-control" id= "username" name= "username"
55 placeholder= "Enter Email ID" autofocus= "autofocus">
56
57
58 <div class= "form-group">
59   <label for= "password">Password</label>: <input type= "password"
60   id= "password" name= "password" class= "form-control"
61   placeholder= "Enter Password" />
62 </div>
63
64 <div class= "form-group">
65   <div class= "row">
66     <div class= "col-sm-6 col-sm-offset-3">
67       <input type= "submit" name= "login-submit" id= "login-submit"
68       class= "form-control btn btn-success" value= "log In" />
69     </div>
70   </div>
71 </div>
```

STS Configuration

workspace-spring-tools-for-eclipse-4.32.2.RELEASE - lbm/src/main/java/com/verinon/lbm/jpa/BookRepository.java - Spring Tools for Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- eshop [boot]
- lbm [boot]
- librarymanagementsystem [boot] [devtools]
 - src/main/java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-17]
 - Maven Dependencies
 - src
 - target
 - LICENSE
 - pom.xml
 - README.md

BookRepository.java

```
1 package com.verinon.lbm.jpa;
2
3 import com.verinon.lbm.pojos.BookPojo;
4
5
6
7 public interface BookRepository extends JpaRepository<BookPojo,Integer>
8 {
9     //List<BookPojo> findByBook_name(String book_name);
10 }
11
```

Outline

- com.verinon.lbm.jpa
 - BookRepository

Console

librarymanagementsystem - Application [Spring Boot App] C:\Program Files\Java\jdk-17.0.5\bin\javaw.exe (Dec 18, 2025, 10:32:38 PM elapsed: 0:30:52) [pid: 22424]

```
2025-12-18 22:32:41.882 INFO 22424 --- [ restartedMain] o.hibernate.jpa.internal.util.LogHelper : org.hibernate.Version : HH0000204: Processing PersistenceUnitInfo
2025-12-18 22:32:41.927 INFO 22424 --- [ restartedMain] o.hibernate.annotations.common.Version : org.hibernate.annotations.common.Version : HH0000412: Hibernate ORM core version : HH000000001: Hibernate Commons Annotations : HH0000400: Using dialect: org.hibernate.dialect.Dialect
2025-12-18 22:32:42.081 INFO 22424 --- [ restartedMain] o.h.e.t.j.p.i.JtaPlatformInitiator : o.h.e.t.j.p.i.JtaPlatformInitiator : HH0000490: Using JtaPlatform implementation : HH0000490: Using JtaPlatform implementation
2025-12-18 22:32:42.786 INFO 22424 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-12-18 22:32:42.848 WARN 22424 --- [ restartedMain] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. This will not secure any request.
2025-12-18 22:32:43.398 INFO 22424 --- [ restartedMain] o.s.s.web.DefaultSecurityFilterChain : Will not secure any request
2025-12-18 22:32:43.718 INFO 22424 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2025-12-18 22:32:43.773 INFO 22424 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 9080 (http)
2025-12-18 22:32:43.783 INFO 22424 --- [ restartedMain] c.k.d.l.Application : Started Application in 3.919 seconds (Catalina version: 9.0.6)
2025-12-18 22:52:27.071 WARN 22424 --- [1-1 housekeeper] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Thread starvation or clock jump detected after shutdown
2025-12-18 22:56:27.670 INFO 22424 --- [nio-9080-exec-3] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-12-18 22:56:27.670 INFO 22424 --- [nio-9080-exec-3] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-12-18 22:56:27.672 INFO 22424 --- [nio-9080-exec-3] o.s.web.servlet.DispatcherServlet : Completed initialization in 2 ms
2025-12-18 23:03:20.999 WARN 22424 --- [1-1 housekeeper] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Thread starvation or clock jump detected after shutdown
```