

Spring Boot Library API – Assignment by ALI DURRANI

1. Project Overview

This project is a RESTful Library Management API developed using Java 21 and Spring Boot. It allows clients to manage books, borrowers, and borrow/return actions. The application is containerized with Docker and supports CI/CD via GitHub Actions.

2. Technologies Used

- Java 21
- Spring Boot 3.5.x
- Spring Web, Spring Data JPA
- H2 Database (in-memory)
- Maven
- Docker
- GitHub Actions (CI/CD)

3. Setup Instructions

To run locally without Docker:

1. Clone the repo: `git clone https://github.com/AliDurrani2015/library-api.git`
 2. Navigate to folder: `cd library-api`
 3. Build the app: `mvn clean package`
 4. Run the app: `java -jar target/library-0.0.1-SNAPSHOT.jar`
- App runs on `http://localhost:8080/`

4. Docker Instructions

To run with Docker:

1. Build: `docker build -t library-api .`
 2. Run: `docker run -p 8080:8080 library-api`
- Open in browser: `http://localhost:8080/api/books`

5. API Endpoints

Books:

- GET /api/books
- POST /api/books
- PUT /api/books/{id}
- DELETE /api/books/{id}

Borrowers:

- GET /api/borrowers
- POST /api/borrowers

Borrow/Return:

- POST /api/borrow/{borrowerId}/book/{bookId}
- POST /api/return/{borrowerId}/book/{bookId}

6. Assumptions

- Each borrower can borrow only one book at a time.
- A book cannot be borrowed if already borrowed.
- Basic validation is performed on input data.
- The database is in-memory and resets on restart.

7. Unit Testing

Unit tests are written using JUnit and Mockito. They cover both controller and service layers.

Run them with: mvn test

8. CI/CD with GitHub Actions

A CI workflow is defined in .github/workflows/ci.yml. It triggers on every push to the main branch.

The workflow performs:

- Checkout
- Java setup (Java 21)
- Build and test using Maven

9. Final Status

- ✓API development completed
- ✓Unit tests passing
- ✓Docker build successful
- ✓CI/CD pipeline green
- ✓README and documentation complete