

✅ Project Planning Steps

1. Define Requirements

You've already done this — a REST API for a Library System with three core resources:

- /books
- /users
- /loans

Each endpoint should support **CRUD** operations.

2. Entity Relationship Diagram (ERD)

Let's define basic relationships:

- A User can have many Loans
- A Book can have many Loans
- A Loan links one User to one Book

So your entities are:

- **User** (id, name, email, etc.)
- **Book** (id, title, author, availableCopies, etc.)
- **Loan** (id, user_id, book_id, loanDate, returnDate)

I can generate an ERD diagram for you if you'd like — just let me know.

📋 Development Checklist

🔧 Setup

- ☐ Install Java (17+ recommended)
- ☐ Install Maven or Gradle
- ☐ Set up Spring Boot project (via Spring Initializr)
- ☐ Add dependencies: Spring Web, Spring Data JPA, H2 Database
- ☐ Configure application.properties or application.yml

🗄 Database

- ☐ Decide on H2 or PostgreSQL
 - H2: good for quick testing, no setup required
 - PostgreSQL: better for production/realistic use cases

Given you're learning, **start with H2** — I can walk you through it.

Given you're learning, **start with H2** — I can walk you through it.

Build Your Application

- ☐ Define JPA entities (`@Entity`)
- ☐ Set up repositories (`JpaRepository`)
- ☐ Create REST controllers (`@RestController`)
- ☐ Add service layer (optional but recommended)
- ☐ Write CRUD endpoints
- ☐ Test using Postman or Curl

Testing

- ☐ Manual testing of each endpoint
- ☐ Write basic unit and integration tests (if comfortable)

H2 Database Overview

What is H2?

H2 is an in-memory relational database. It's fast, doesn't require installation, and resets on every application restart (unless file-based).

How to Use in Spring Boot:

1. Add this dependency:

```
<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
```

2. In `application.properties`:

```
spring.datasource.url=jdbc:h2:mem:librarydb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.h2.console.enabled=true
```

`spring.jpa.hibernate.ddl-auto=update`

Access H2 console at `http://localhost:8080/h2-console`

- JDBC URL: `jdbc:h2:mem:librarydb`