**Exercise 9: Creating a Spring Boot Application**

Scenario:

You need to create a Spring Boot application for the library management system to simplify configuration and deployment.

**1. Project Setup**

1.1 Creating a Spring Boot Project

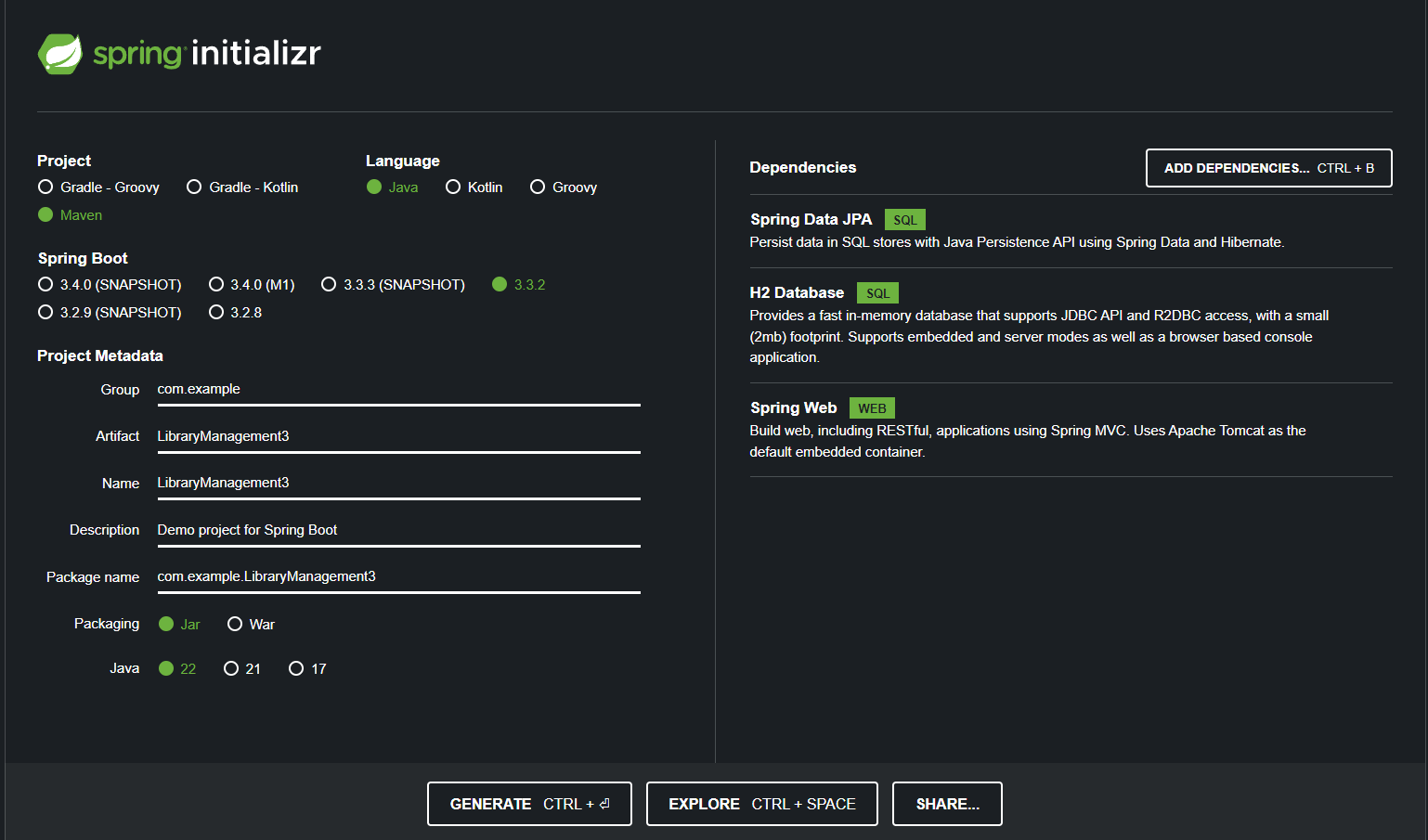
The Spring Boot project was created using Spring Initializr with the following specifications:

* Project Name: LibraryManagement

1.2 Adding Dependencies

The pom.xml file was configured to include the necessary dependencies:

* spring-boot-starter-web: For building web applications, including RESTful services.
* spring-boot-starter-data-jpa: For JPA (Java Persistence API) integration.
* h2: In-memory database for testing and development.



**2. Application Configuration**

2.1 Application Properties

The application.properties file was configured to set up the H2 database and JPA. This configuration enables the H2 console for viewing the database and sets up the JPA for managing database operations.

**3. Entity and Repository Setup**

3.1 Book Entity

The Book entity was created to represent books in the library. This class is annotated with @Entity to define it as a JPA entity

3.2 BookRepository Interface

The BookRepository interface was created by extending JpaRepository to provide CRUD operations for Book entities:

**4. REST Controller**

4.1 BookController

The BookController class was implemented to handle RESTful API requests for Book entities. This class includes the following endpoints:

* GET /books: Retrieve all books.
* GET /books/{id}: Retrieve a book by ID.
* POST /books: Create a new book.
* PUT /books/{id}: Update a book.
* DELETE /books/{id}: Delete a book by ID.

5. Running the Application

The Spring Boot application can be run using the main method in LibraryManagementApplication3 class. The application will start on the default port, and I changed the port to 8098 and then tested the REST endpoints using Postman.

**Output:**

