WEEK – 3

Exercise 9: Creating a Spring Boot Application

Nipuna.A

Superset id:6432842

**Scenario:**

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

**Steps:**

1. **Create a Spring Boot Project:**
   * Use **Spring Initializr** to create a new Spring Boot project named **LibraryManagement**.
2. **Add Dependencies:**
   * Include dependencies for **Spring Web, Spring Data JPA, and H2 Database**.
3. **Create Application Properties:**
   * Configure database connection properties in **application.properties**.
4. **Define Entities and Repositories:**
   * Create **Book** entity and **BookRepository** interface.
5. **Create a REST Controller:**
   * Create a **BookController** class to handle CRUD operations.
6. **Run the Application:**
   * Run the Spring Boot application and test the REST endpoints.

Solution:

1)Book.java

package com.library.model;

public class Book {

    private String title;

    public Book(String title) {

        this.title = title;

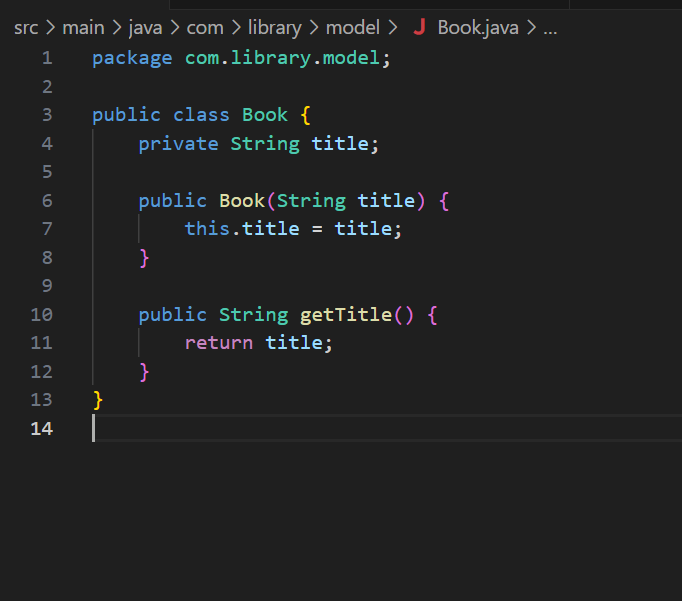
    }

    public String getTitle() {

        return title;

    }

}



2)BookRepository.java

package com.library.repository;

import com.library.model.Book;

import org.springframework.stereotype.Repository;

@Repository

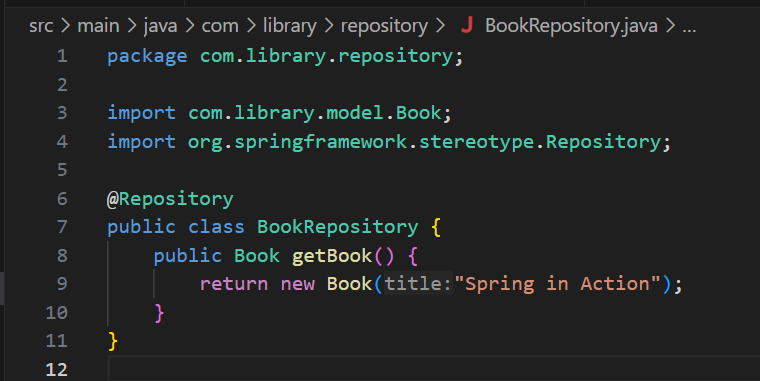
public class BookRepository {

    public Book getBook() {

        return new Book("Spring in Action");

    }

}



3)BookService.java

package com.library.service;

import com.library.model.Book;

import com.library.repository.BookRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service

public class BookService {

    private BookRepository bookRepository;

    @Autowired

    public BookService(BookRepository bookRepository) {

        System.out.println("Constructor Injection used");

        this.bookRepository = bookRepository;

    }

    @Autowired

    public void setBookRepository(BookRepository bookRepository) {

        System.out.println("Setter Injection used");

        this.bookRepository = bookRepository;

    }

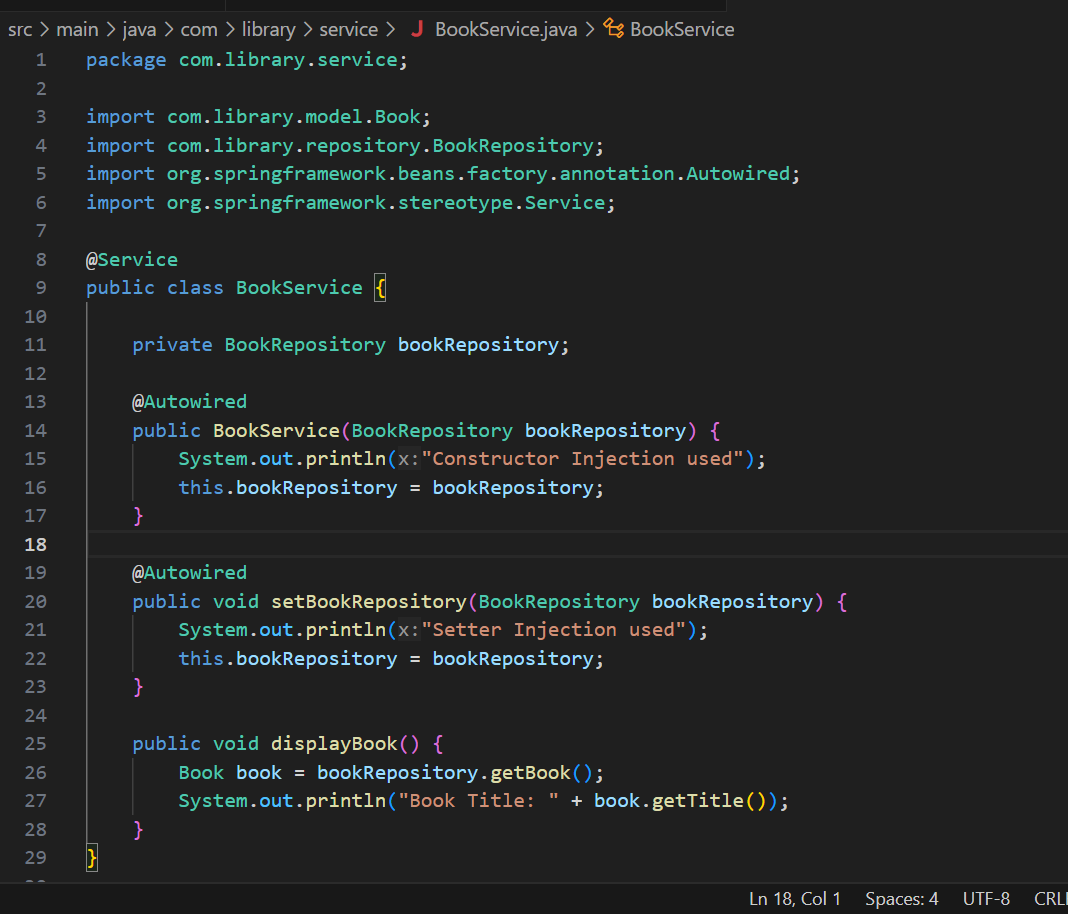
    public void displayBook() {

        Book book = bookRepository.getBook();

        System.out.println("Book Title: " + book.getTitle());

    }

}



4)LibraryManagementApplication.java

package com.library;

import com.library.service.BookService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class LibraryManagementApplication implements CommandLineRunner {

    @Autowired

    private BookService bookService;

    public static void main(String[] args) {

        SpringApplication.run(LibraryManagementApplication.class, args);

    }

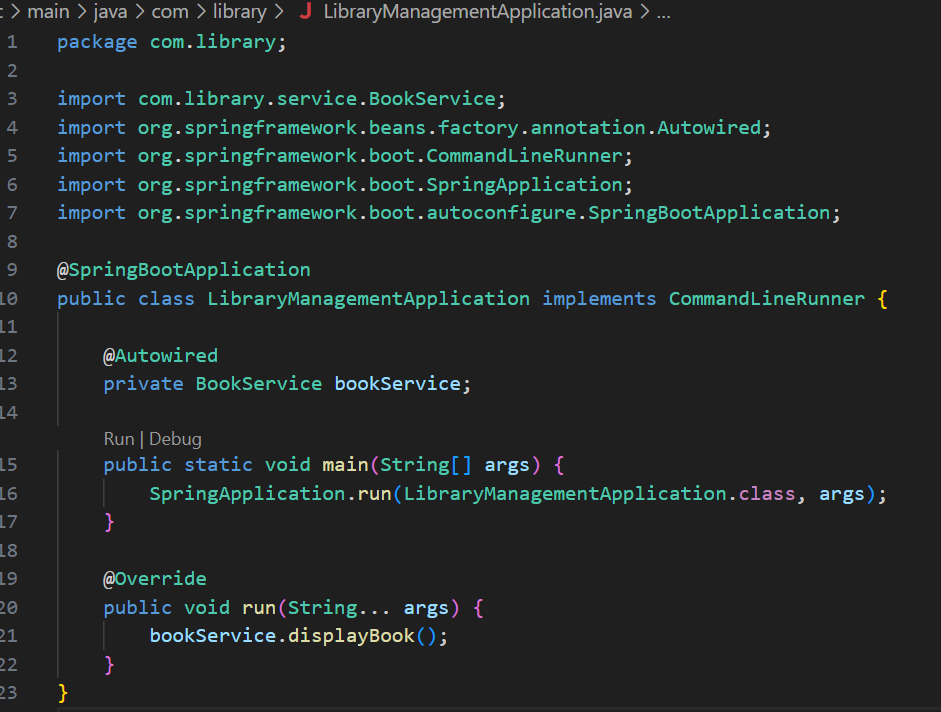
    @Override

    public void run(String... args) {

        bookService.displayBook();

    }

}



5)Application properties

server.port=8081

spring.datasource.url=jdbc:h2:mem:librarydb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

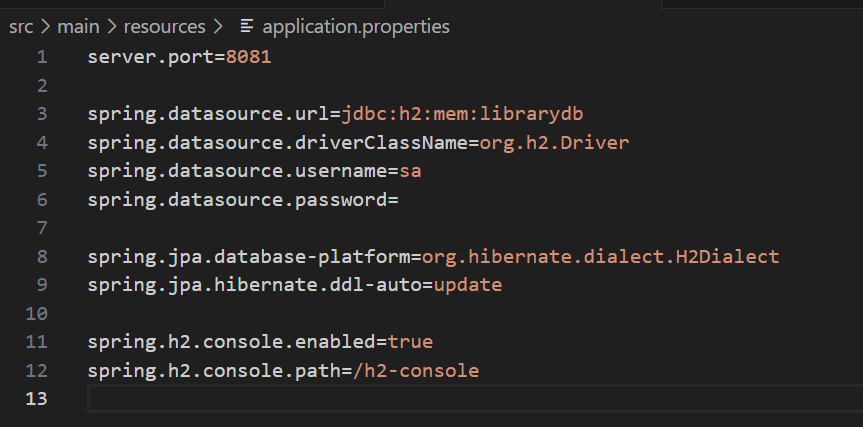
spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console



6)Pom.xml

<project xmlns="http://maven.apache.org/POM/4.0.0"

         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0

                             http://maven.apache.org/xsd/maven-4.0.0.xsd">

    <modelVersion>4.0.0</modelVersion>

    <groupId>com.library</groupId>

    <artifactId>LibraryManagement</artifactId>

    <version>1.0.0</version>

    <packaging>jar</packaging>

    <name>LibraryManagement</name>

    <description>Spring Boot Library Management System</description>

    <parent>

        <groupId>org.springframework.boot</groupId>

        <artifactId>spring-boot-starter-parent</artifactId>

        <version>3.2.0</version>

        <relativePath/> <!-- lookup parent from repository -->

    </parent>

    <properties>

        <java.version>17</java.version>

    </properties>

    <dependencies>

        <!-- Spring Web -->

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-web</artifactId>

        </dependency>

        <!-- Spring Data JPA -->

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-data-jpa</artifactId>

        </dependency>

        <!-- H2 Database -->

        <dependency>

            <groupId>com.h2database</groupId>

            <artifactId>h2</artifactId>

            <scope>runtime</scope>

        </dependency>

        <!-- Spring Boot Test (Optional, for unit testing) -->

        <dependency>

            <groupId>org.springframework.boot</groupId>

            <artifactId>spring-boot-starter-test</artifactId>

            <scope>test</scope>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <!-- Spring Boot Maven Plugin -->

            <plugin>

                <groupId>org.springframework.boot</groupId>

                <artifactId>spring-boot-maven-plugin</artifactId>

                <version>3.2.0</version>

                <executions>

                    <execution>

                        <goals>

                            <goal>repackage</goal>

                        </goals>

                    </execution>

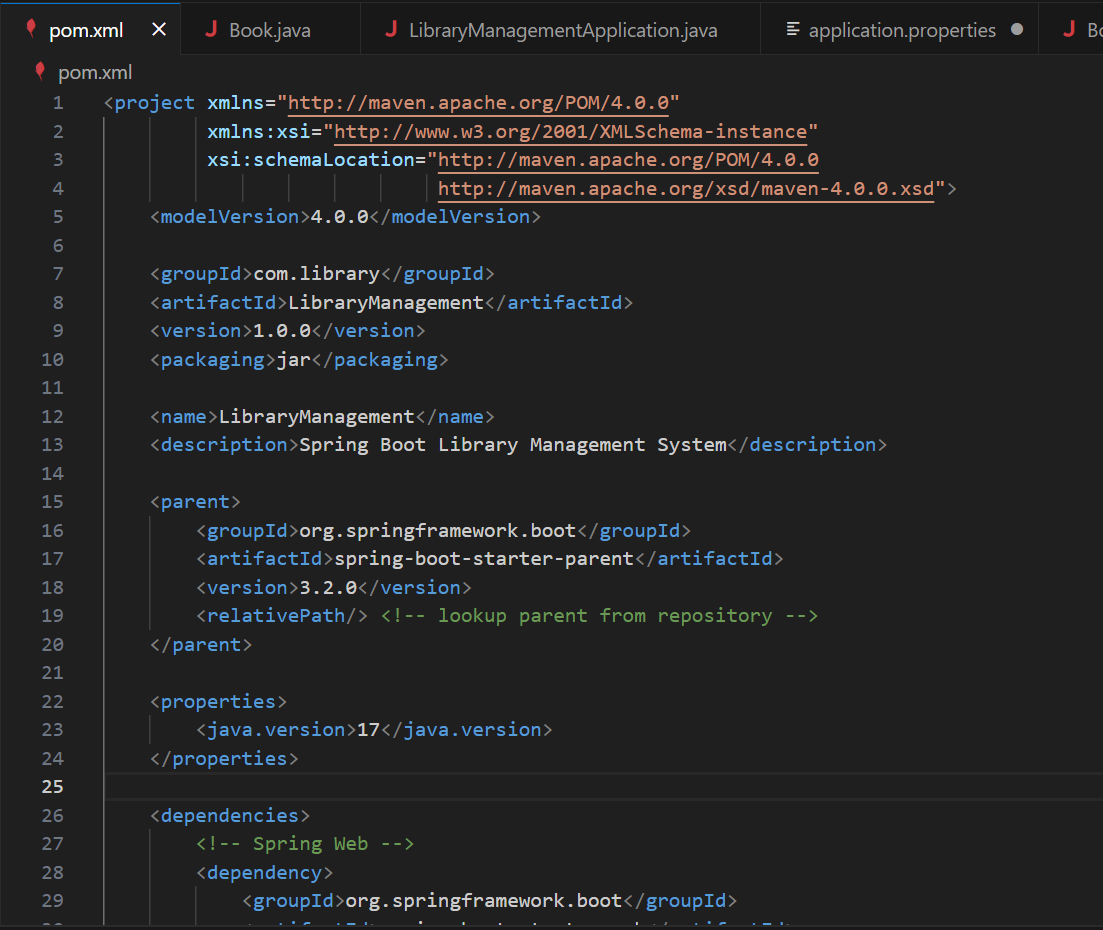
                </executions>

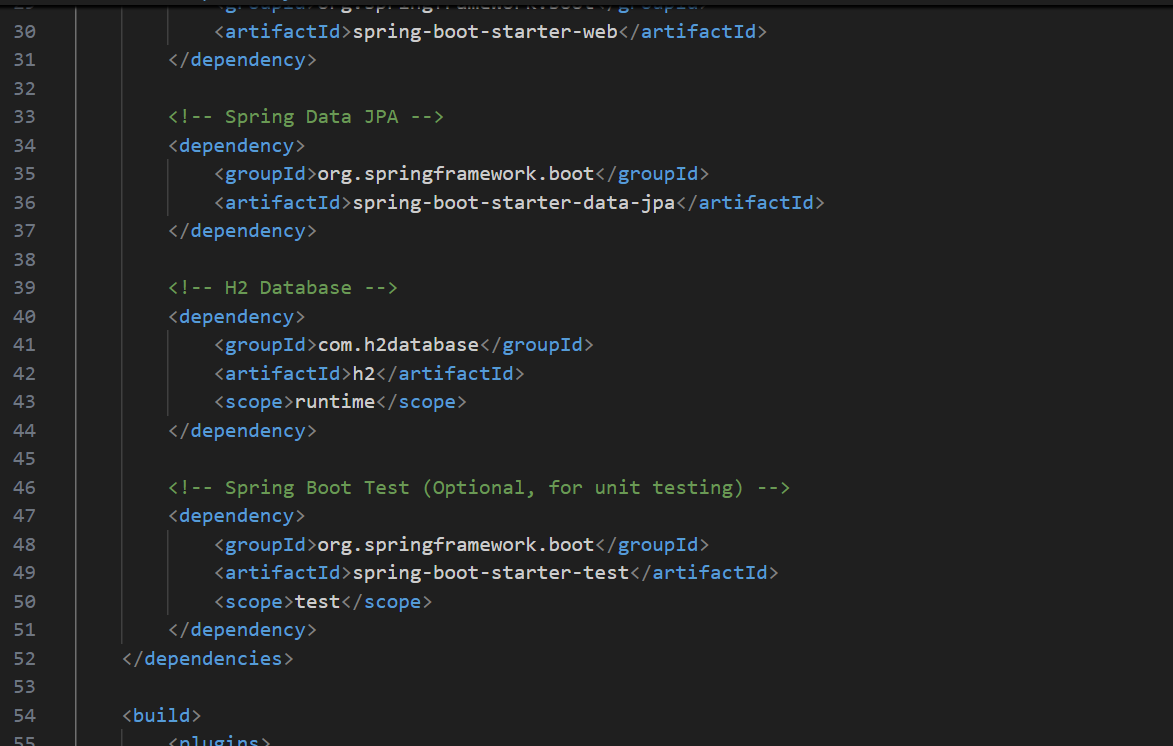
            </plugin>

        </plugins>

    </build>

</project>





Output:

