**Spring Core and Maven**

**Exercise 1: Configuring a Basic Spring Application**

**Code:**

**<!--pom.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<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.example.librarymanagement</groupId>

<artifactId>librarymanagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.31</version>

</dependency>

</dependencies>

</project>

**<!--applicationContext.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

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

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="bookRepository" class="com.example.librarymanagement.repository.BookRepository"/>

<bean id="bookService" class="com.example.librarymanagement.service.BookService">

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

**//BookRepository.java**

package com.example.librarymanagement.repository;

public class BookRepository {

public String getBookInfo(String isbn) {

return "Book details for ISBN: " + isbn;

}

}

**//BookService.java**

package com.example.librarymanagement.service;

import com.example.librarymanagement.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public String getBookDetails(String isbn) {

return bookRepository.getBookInfo(isbn);

}

}

**//Main.java**

package com.example.librarymanagement;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.librarymanagement.service.BookService;

public class Main {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = context.getBean("bookService", BookService.class);

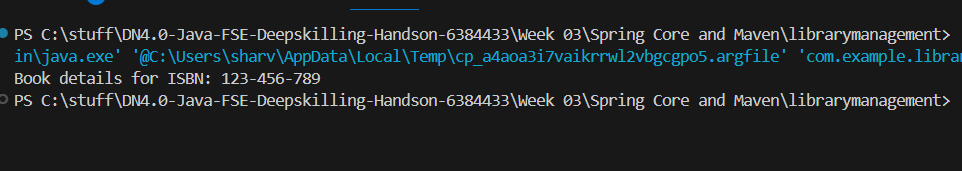
String bookDetails = bookService.getBookDetails("123-456-789");

System.out.println(bookDetails);

((ClassPathXmlApplicationContext) context).close();

}

}

**Output:**

**Exercise 2: Implementing Dependency Injection**

**Code:**

**<!--applicationContext.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

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

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="bookRepository" class="com.example.librarymanagement.repository.BookRepository"/>

<bean id="bookService" class="com.example.librarymanagement.service.BookService">

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

**//BookService.java**

package com.example.librarymanagement.service;

import com.example.librarymanagement.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public BookRepository getBookRepository() {

return this.bookRepository;

}

public void setBookRepository(BookRepository bookRepository) {

System.out.println("Dependency injection: Setting BookRepository");

this.bookRepository = bookRepository;

}

public String getBookDetails(String isbn) {

return bookRepository.getBookInfo(isbn);

}

}

**//Main.java**

package com.example.librarymanagement;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.librarymanagement.service.BookService;

public class Main {

public static void main(String[] args) {

System.out.println("Initializing Spring context...");

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

System.out.println("Getting BookService bean...");

BookService bookService = context.getBean("bookService", BookService.class);

System.out.println("Testing book service...");

String bookDetails = bookService.getBookDetails("978-3-16-148410-0");

System.out.println(bookDetails);

if (bookService.getBookRepository() != null) {

System.out.println("Dependency injection successful!");

} else {

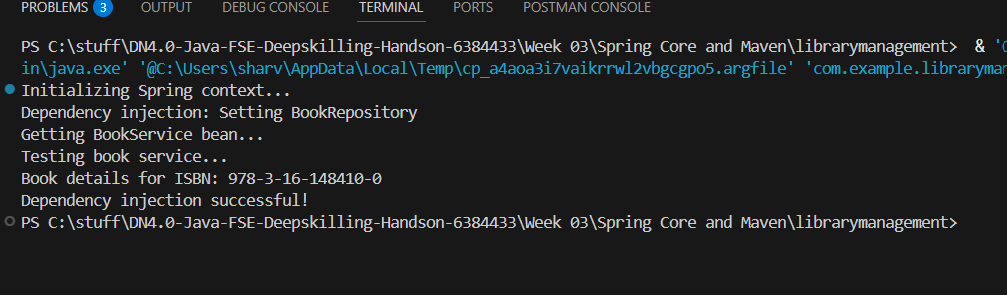
System.out.println("Dependency injection failed!");

}

((ClassPathXmlApplicationContext) context).close();

}

}

**Output:**

**Exercise 3: Implementing Logging with Spring AOP**

**Code:**

**<!--pom.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<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.example.librarymanagement</groupId>

<artifactId>librarymanagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.31</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.31</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.7</version>

</dependency>

</dependencies>

</project>

**//LoggingAspect.java**

package com.example.librarymanagement.aspect;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class LoggingAspect {

@Around("execution(\* com.example.librarymanagement.service.\*.\*(..)) || " +

"execution(\* com.example.librarymanagement.repository.\*.\*(..))")

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long startTime = System.currentTimeMillis();

String className = joinPoint.getTarget().getClass().getSimpleName();

String methodName = joinPoint.getSignature().getName();

System.out.println(String.format("Entering %s.%s()", className, methodName));

try {

Object result = joinPoint.proceed();

long executionTime = System.currentTimeMillis() - startTime;

System.out.println(String.format(

"Exiting %s.%s() - execution time: %d ms",

className, methodName, executionTime));

return result;

} catch (Exception e) {

long executionTime = System.currentTimeMillis() - startTime;

System.out.println(String.format(

"Exception in %s.%s() - execution time: %d ms - Exception: %s",

className, methodName, executionTime, e.getMessage()));

throw e;

}

}

}

**<!--applicationContext.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

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

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:component-scan base-package="com.example.librarymanagement"/>

<aop:aspectj-autoproxy/>

<bean id="bookRepository" class="com.example.librarymanagement.repository.BookRepository"/>

<bean id="bookService" class="com.example.librarymanagement.service.BookService">

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

**//LibraryManagementApplication.java**

package com.example.librarymanagement;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.librarymanagement.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = context.getBean("bookService", BookService.class);

System.out.println("Testing with valid ISBN:");

String validResult = bookService.getBookDetails("978-3-16-148410-0");

System.out.println("Result: " + validResult);

System.out.println("\nTesting with invalid ISBN:");

String invalidResult = bookService.getBookDetails("000-0-00-000000-0");

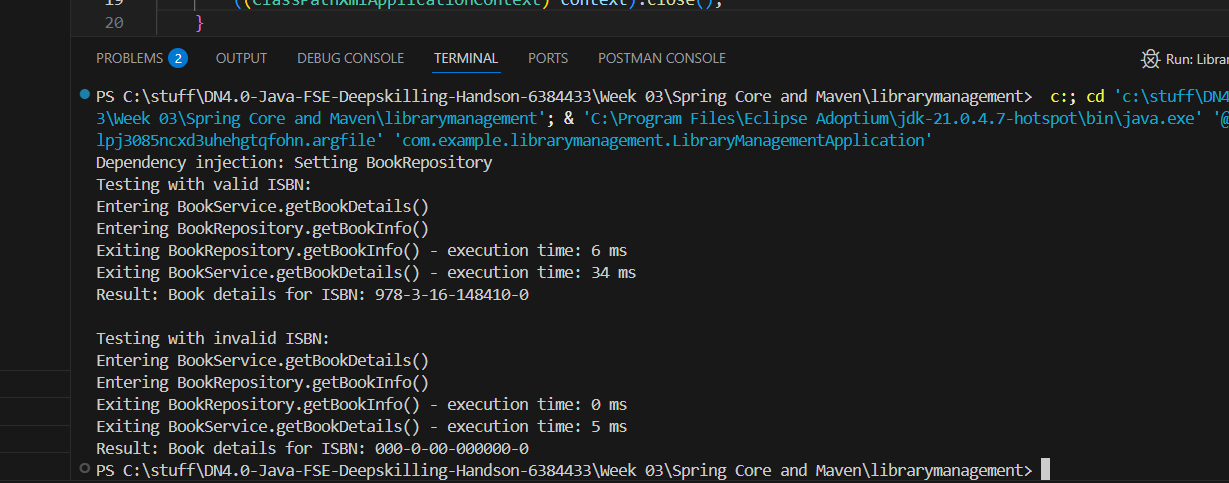
System.out.println("Result: " + invalidResult);

((ClassPathXmlApplicationContext) context).close();

}

}

**Output:**



**Exercise 4: Creating and Configuring a Maven Project**

**Code:**

**<!--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-SNAPSHOT</version>

<packaging>jar</packaging>

<properties>

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

<spring.version>5.3.32</spring.version>

<slf4j.version>1.7.36</slf4j.version>

<maven.compiler.plugin.version>3.11.0</maven.compiler.plugin.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-simple</artifactId>

<version>${slf4j.version}</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>1.9.21</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>${maven.compiler.plugin.version}</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Exercise 5: Configuring the Spring IoC Container’**

**Code:**

**<!--applicationContext.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

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

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="bookRepository" class="com.library.repository.BookRepository" />

<bean id="bookService" class="com.library.service.BookService">

<property name="bookRepository" ref="bookRepository" />

</bean>

</beans>

**//BookRepository.java**

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Book saved: " + bookName);

}

}

**//BookService.java**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("Adding book: " + bookName);

bookRepository.saveBook(bookName);

}

}

**//LibraryManagementApplication.java**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

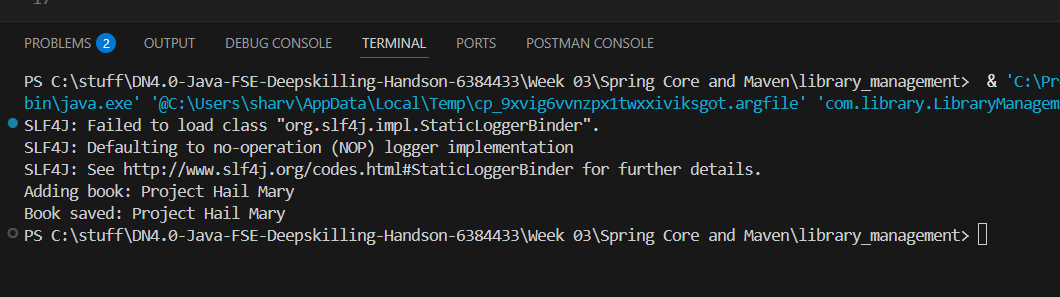
BookService bookService = context.getBean("bookService", BookService.class);

bookService.addBook("Project Hail Mary");

((org.springframework.context.ConfigurableApplicationContext) context).close();

}

}

**Output:**

**Exercise 6: Configuring Beans with Annotations**

**Code:**

**<!--applicationContext.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:context="http://www.springframework.org/schema/context"

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

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:component-scan base-package="com.library" />

</beans>

**//BookRepository.java**

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Book saved: " + bookName);

}

}

**//BookService.java**

package com.library.service;

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

import org.springframework.stereotype.Service;

import com.library.repository.BookRepository;

@Service

public class BookService {

private BookRepository bookRepository;

@Autowired

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("Adding book: " + bookName);

bookRepository.saveBook(bookName);

}

}

**//LibraryManagementApplication.java**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = context.getBean(BookService.class);

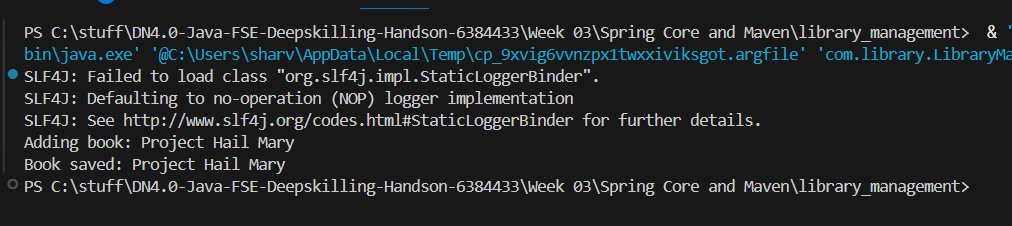
bookService.addBook("Project Hail Mary");

((org.springframework.context.ConfigurableApplicationContext) context).close();

}

}

**Output:**



**Exercise 7: Implementing Constructor and Setter Injection**

**Code:**

**<!--applicationContext.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

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

xsi:schemaLocation="http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd">

<bean id="bookRepository" class="com.library.repository.BookRepository" />

<bean id="bookService" class="com.library.service.BookService">

<constructor-arg value="Central City Library" />

<property name="bookRepository" ref="bookRepository" />

</bean>

</beans>

**//BookService.java**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

private String libraryName;

public BookService(String libraryName) {

this.libraryName = libraryName;

System.out.println("Constructor Injection: Library name = " + libraryName);

}

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("Adding book to " + libraryName + ": " + bookName);

bookRepository.saveBook(bookName);

}

}

**//LibraryManagementApplication.java**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = context.getBean(BookService.class);

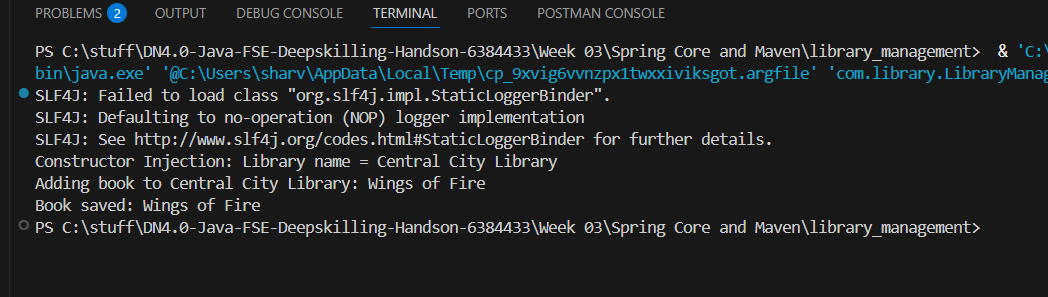
bookService.addBook("Wings of Fire");

((org.springframework.context.ConfigurableApplicationContext) context).close();

}

}

**Output:**



**Exercise 8: Implementing Basic AOP with Spring**

**Code:**

**<!--applicationContext.xml-->**

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:aop="http://www.springframework.org/schema/aop"

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

xsi:schemaLocation="

http://www.springframework.org/schema/beans

http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd">

<context:component-scan base-package="com.library" />

<aop:aspectj-autoproxy />

</beans>

**//BookRepository.java**

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Book saved: " + bookName);

}

}

**//BookService.java**

package com.library.service;

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

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

import org.springframework.stereotype.Service;

import com.library.repository.BookRepository;

@Service

public class BookService {

private BookRepository bookRepository;

private String libraryName;

public BookService(@Value("Central Library") String libraryName) {

this.libraryName = libraryName;

System.out.println("Constructor Injection: Library name = " + libraryName);

}

@Autowired

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("Adding book to " + libraryName + ": " + bookName);

bookRepository.saveBook(bookName);

}

}

**//LibraryManagementApplication.java**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

BookService bookService = context.getBean(BookService.class);

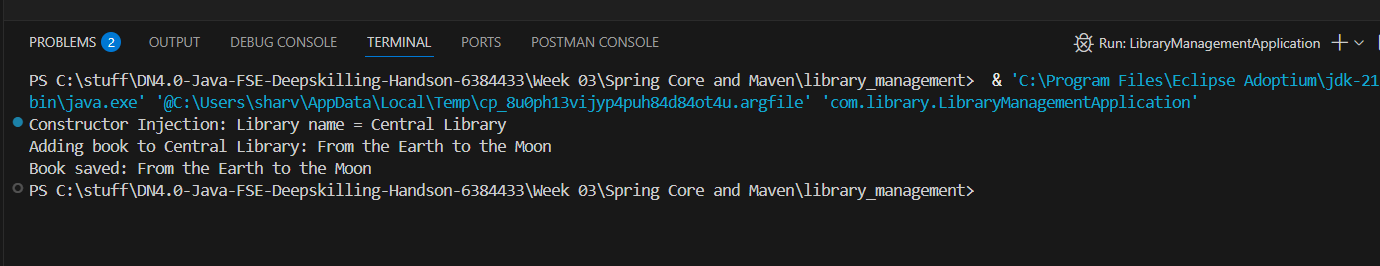
bookService.addBook("From the Earth to the Moon");

((org.springframework.context.ConfigurableApplicationContext) context).close();

}

}

**Output:**



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

**Code:**

**#application.properties**

spring.application.name=Library-Management

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

**//Book.java**

package com.library.Library\_Management.entity;

import jakarta.persistence.\*;

@Entity

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String author;

public Book() {}

public Book(String title, String author) {

this.title = title;

this.author = author;

}

public Long getId() { return id; }

public void setId(Long id) { this.id = id; }

public String getTitle() { return title; }

public void setTitle(String title) { this.title = title; }

public String getAuthor() { return author; }

public void setAuthor(String author) { this.author = author; }

}

**//BookRepository.java**

package com.library.Library\_Management.repository;

import com.library.Library\_Management.entity.Book;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Long> {

}

**//BookController.java**

package com.library.Library\_Management.controller;

import com.library.Library\_Management.entity.Book;

import com.library.Library\_Management.repository.BookRepository;

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

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/books")

public class BookController {

@Autowired

private BookRepository bookRepository;

@GetMapping

public List<Book> getAllBooks() {

return bookRepository.findAll();

}

@PostMapping

public Book addBook(@RequestBody Book book) {

return bookRepository.save(book);

}

@GetMapping("/{id}")

public Book getBookById(@PathVariable Long id) {

return bookRepository.findById(id).orElse(null);

}

@PutMapping("/{id}")

public Book updateBook(@PathVariable Long id, @RequestBody Book updatedBook) {

Book book = bookRepository.findById(id).orElse(null);

if (book != null) {

book.setTitle(updatedBook.getTitle());

book.setAuthor(updatedBook.getAuthor());

return bookRepository.save(book);

}

return null;

}

@DeleteMapping("/{id}")

public void deleteBook(@PathVariable Long id) {

bookRepository.deleteById(id);

}

}

**//LibraryManagementApplication.java**

package com.library.Library\_Management;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

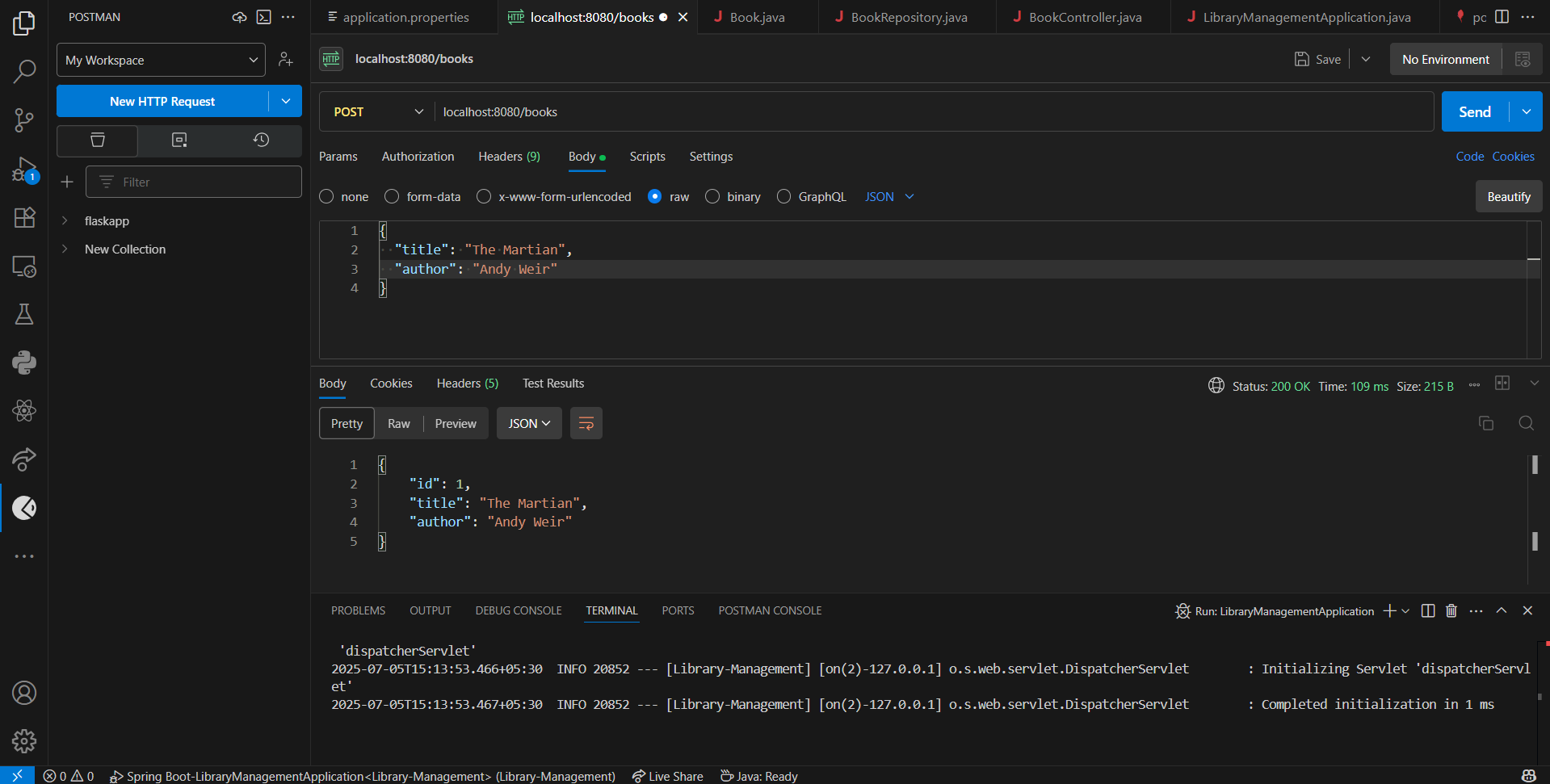
public class LibraryManagementApplication {

public static void main(String[] args) {

SpringApplication.run(LibraryManagementApplication.class, args);

}

}

**Output:**

