**(6428317 Mukesh P)**

**WEEK 3: SPRING CORE AND MAVEN:**

**MANDATORY HANDS – ON:**

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

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() {

System.***out***.println("BookService: Adding a book...");

bookRepository.saveBook();

}

}

BookRepository.java:

**package** com.library.repository;

**public** **class** BookRepository {

**public** **void** saveBook() {

System.***out***.println("Book saved to repo");

}

}

MainApp.java:

**package** com.library;

**import** com.library.service.BookService;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** MainApp {

**public** **static** **void** main(String[] args) {

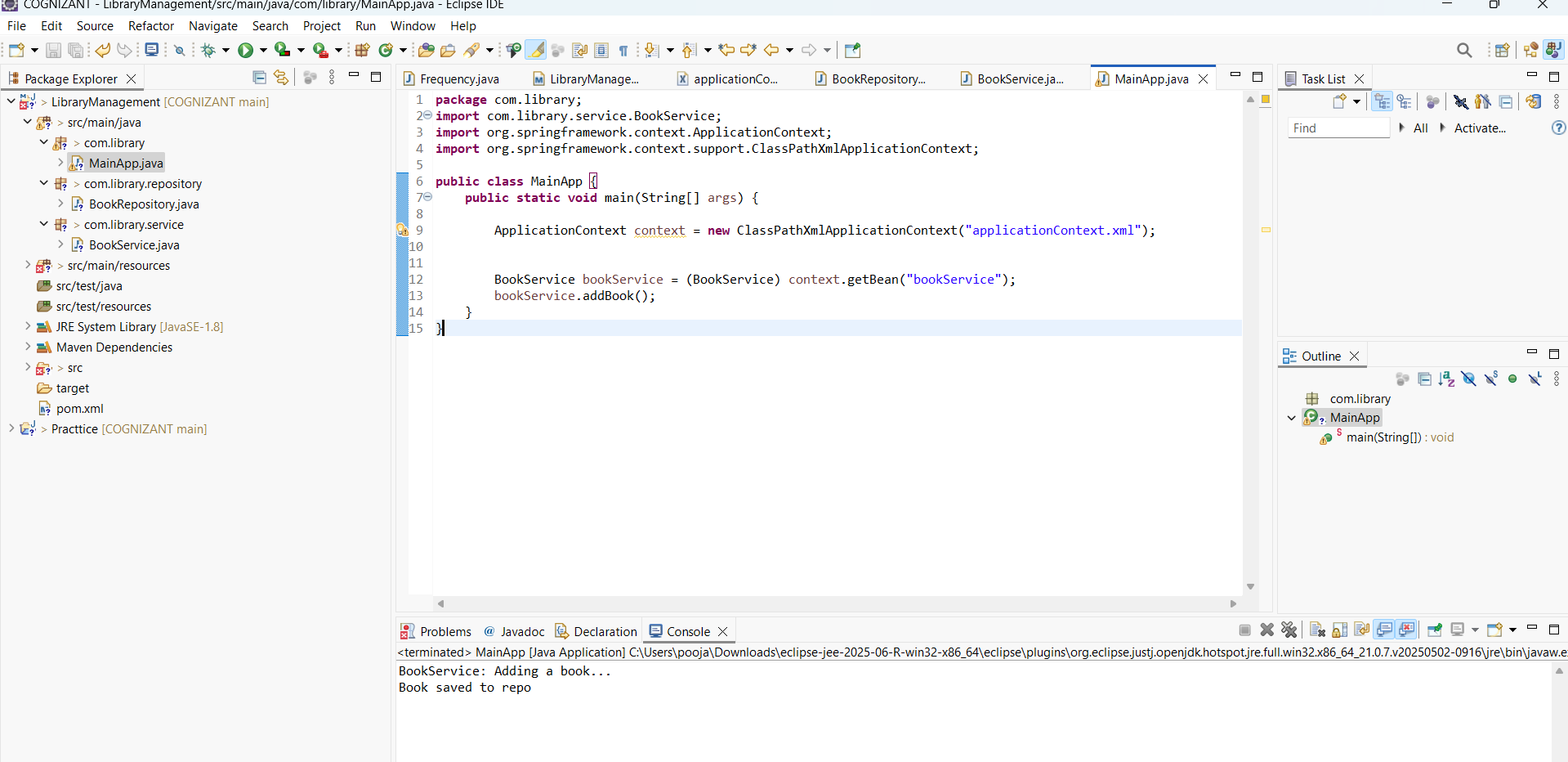
ApplicationContext context = **new** ClassPathXmlApplicationContext("applicationContext.xml");

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

bookService.addBook();

}

}



**Exercise 2: Implementing Dependency Injection**

**Modify the XML Configuration:**

<?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"*>

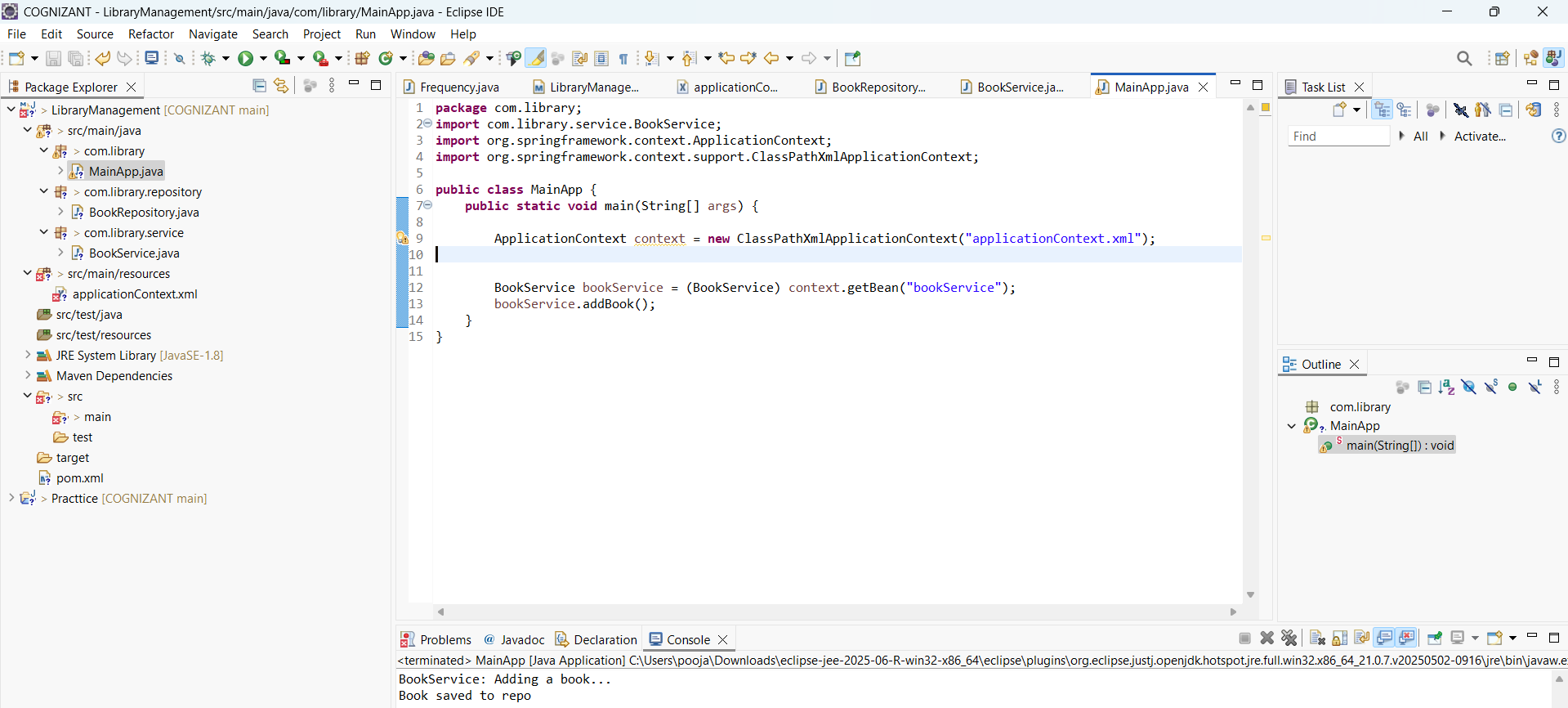
<bean id=*"bookRepository"* class=*"com.library.repository.BookRepository"* />

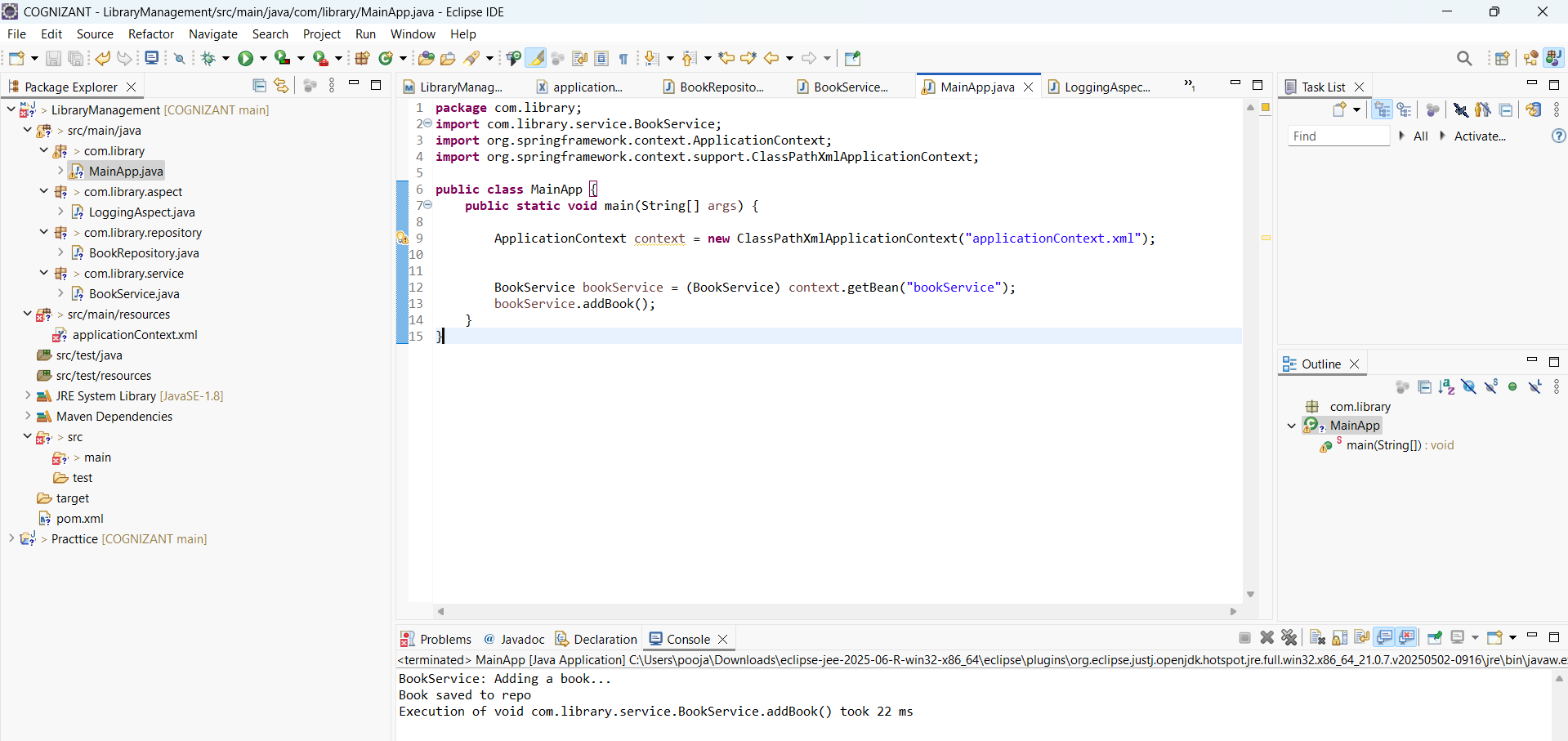
<bean id=*"bookService"* class=*"com.library.service.BookService"*>

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

</bean>

</beans>



}

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

<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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.library</groupId>

<artifactId>LibraryManagement2</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.33</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.33</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.33</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

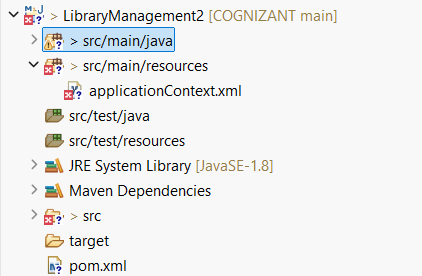
</configuration>

</plugin>

</plugins>

</build>

</project>

****

**ADDITIONAL HANDS-ON:**

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

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"*

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"*>

<!-- Enable AOP Support -->

<aop:aspectj-autoproxy />

<!-- Beans -->

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

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

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

</bean>

<bean id=*"logginAspect"* class=*"com.library.aspect.LogginAspect"* />

</beans>

**BookRepository.java:**

**package** com.library.repository;

**public** **class** BookRepository {

**public** **void** saveBook() {

System.***out***.println("Book saved to the repository.");

}

}

**BookService.java:**

**package** com.library.service;

**import** com.library.repository.BookRepository;

**public** **class** BookService {

**private** BookRepository bookRepository;

// Setter

**public** **void** setBookRepository(BookRepository bookRepository) {

**this**.bookRepository = bookRepository;

}

**public** **void** addBook() {

System.***out***.println("BookService: Adding a book...");

bookRepository.saveBook();

}

}

**LogginAspect.java:**

**package** com.library.aspect;

**import** org.aspectj.lang.ProceedingJoinPoint;

**import** org.aspectj.lang.annotation.Around;

**import** org.aspectj.lang.annotation.Aspect;

@Aspect

**public** **class** LogginAspect {

@Around("execution(\* com.library.service.\*.\*(..))")

**public** Object logExecutionTime(ProceedingJoinPoint joinPoint) **throws** Throwable {

**long** start = System.*currentTimeMillis*();

Object result = joinPoint.proceed();

**long** timeTaken = System.*currentTimeMillis*() - start;

System.***out***.println("Execution of " + joinPoint.getSignature() + " took " + timeTaken + " ms");

**return** result;

}

}

**MainApp.java**

**package** com.library;

**import** com.library.service.BookService;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** MainApp {

**public** **static** **void** main(String[] args) {

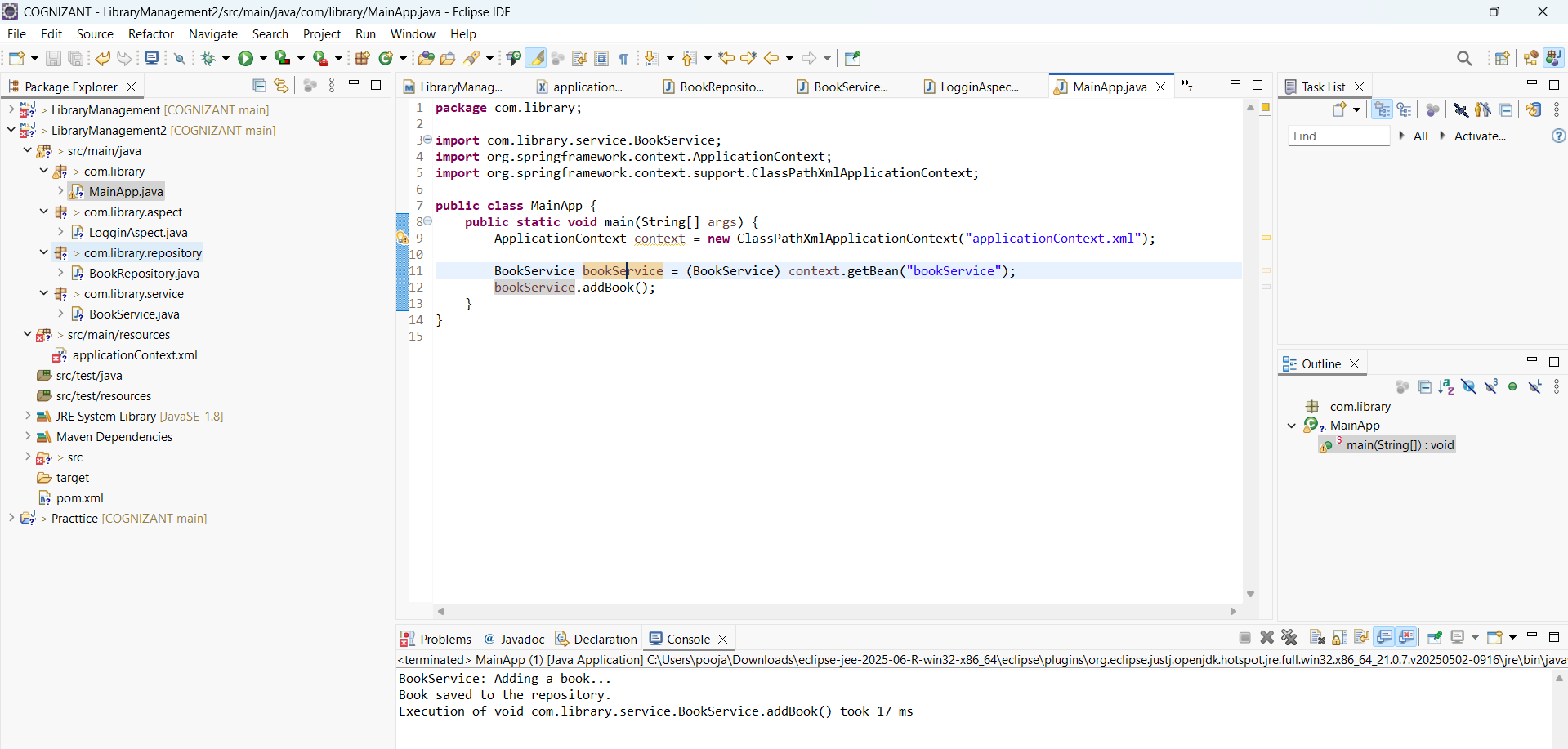
ApplicationContext context = **new** ClassPathXmlApplicationContext("applicationContext.xml");

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

bookService.addBook();

}

}



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

<?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=*"City Central Library"* />

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

</bean>

</beans>

**BookService.java:**

**package** com.library.service;

**import** com.library.repository.BookRepository;

**public** **class** BookService {

**private** String libraryName;

**private** BookRepository bookRepository;

**public** BookService(String libraryName) {

**this**.libraryName = libraryName;

}

**public** **void** setBookRepository(BookRepository bookRepository) {

**this**.bookRepository = bookRepository;

}

**public** **void** addBook() {

System.***out***.println("[" + libraryName + "] BookService: Adding a book...");

bookRepository.saveBook();

}

}

**BookRepositoy.java:**

**package** com.library.repository;

**public** **class** BookRepository {

**public** **void** saveBook() {

System.***out***.println("Book saved to the repository.");

}

}

**MainApp.java:**

**package** com.library;

**import** com.library.service.BookService;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**public** **class** MainApp {

**public** **static** **void** main(String[] args) {

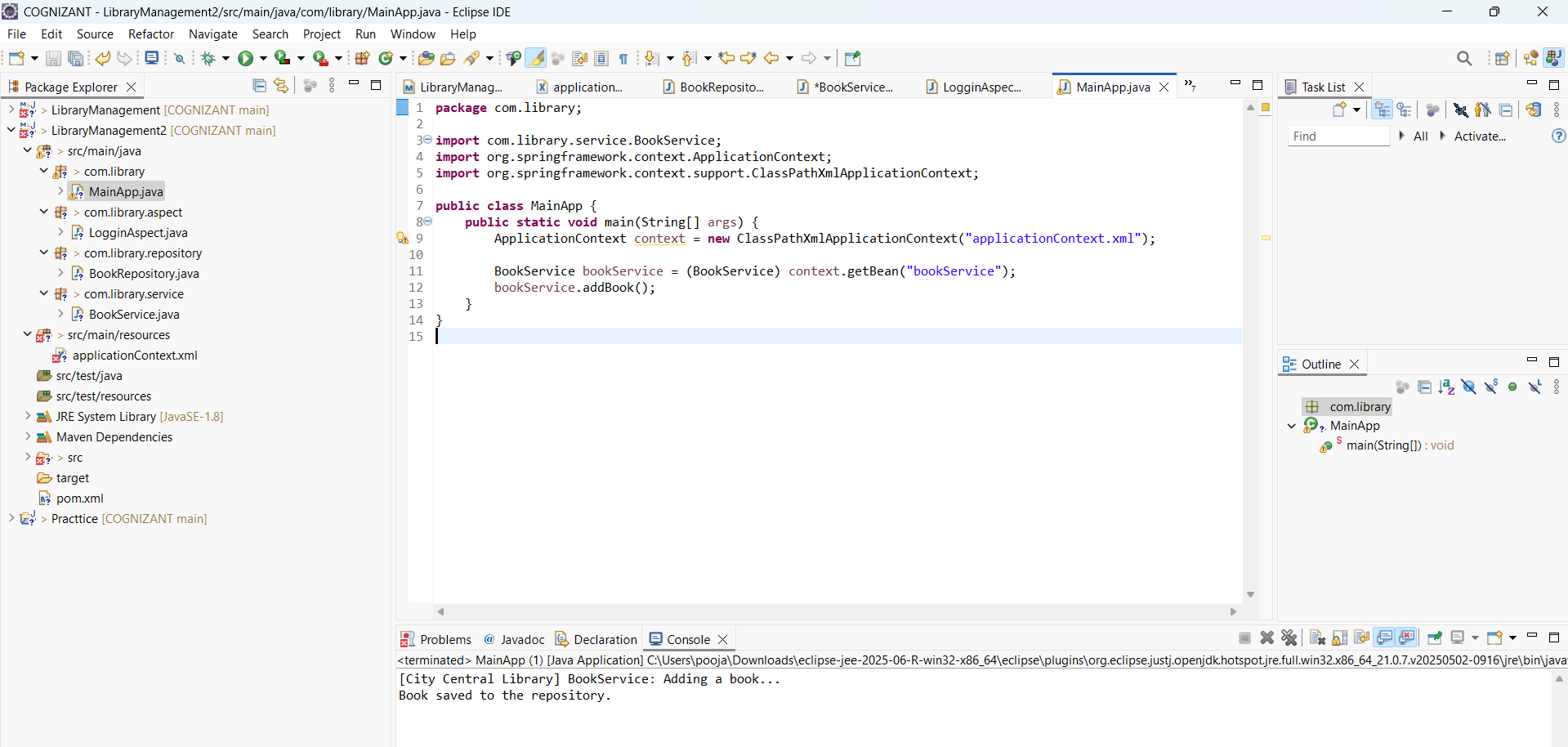
ApplicationContext context = **new** ClassPathXmlApplicationContext("applicationContext.xml");

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

bookService.addBook();

}

}



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

**Add Dependencies:**

spring.application.name=LibraryManagement2

spring.h2.console.enabled=true

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

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

**Book.java:**

**package** com.library.model;

**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.repository;

**import** com.library.model.Book;

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

**public** **interface** BookRepository **extends** JpaRepository<Book, Long> {

}

**BookController.java:**

**package** com.library.controller;

**import** com.library.model.Book;

**import** com.library.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 createBook(@RequestBody Book book) {

**return** bookRepository.save(book);

}

@GetMapping("/{id}")

**public** Book getBook(@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);

}

}

**Main:**

**package** com.library;

**import** org.springframework.boot.SpringApplication;

**import** org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

**public** **class** LibraryManagement2Application {

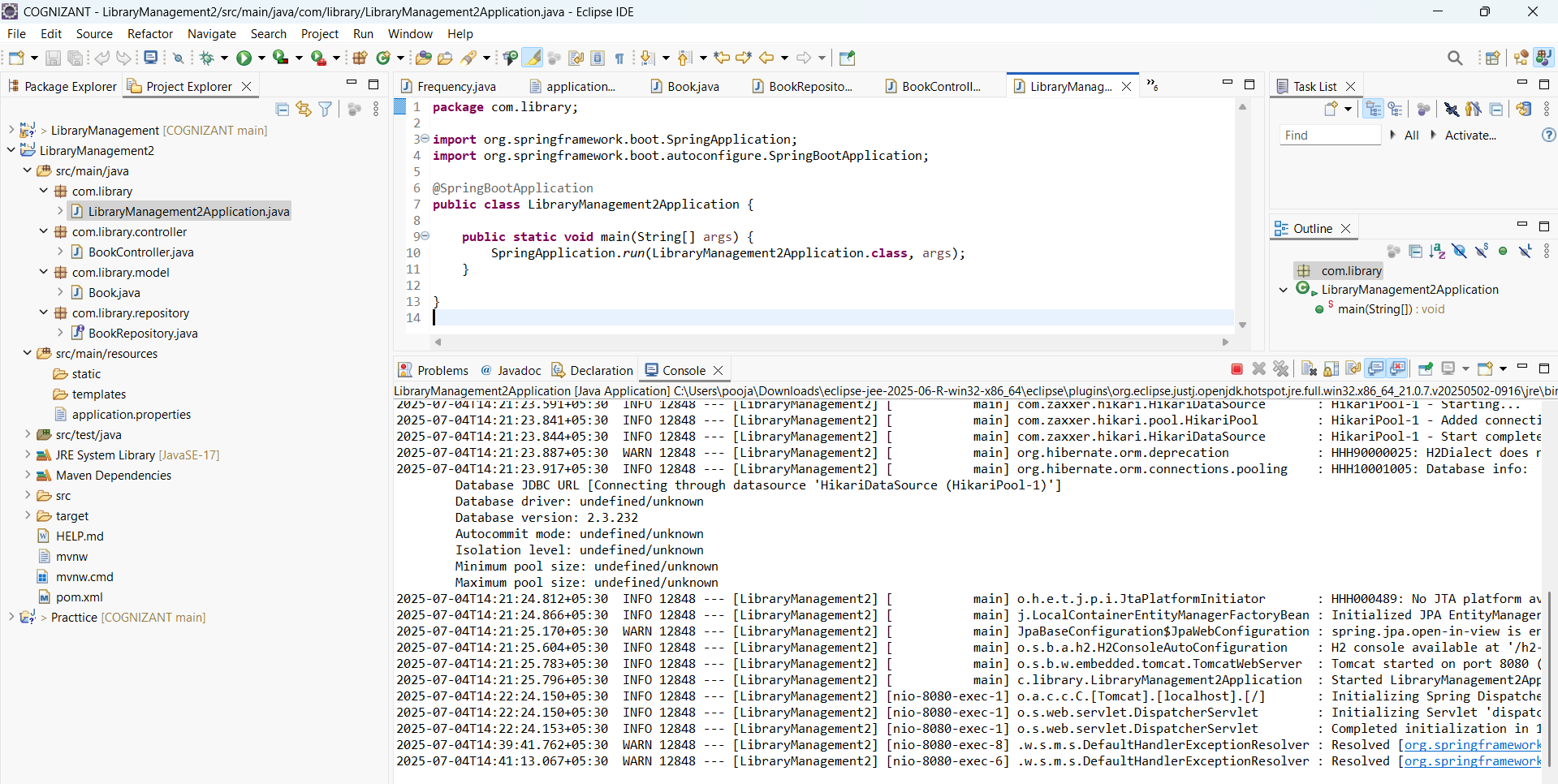
**public** **static** **void** main(String[] args) {

SpringApplication.*run*(LibraryManagement2Application.**class**, args);

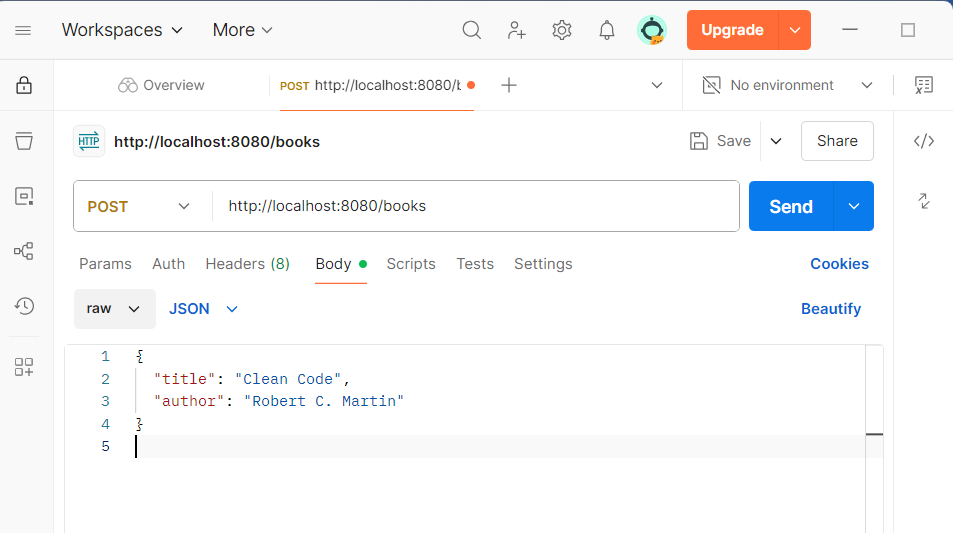
}

}

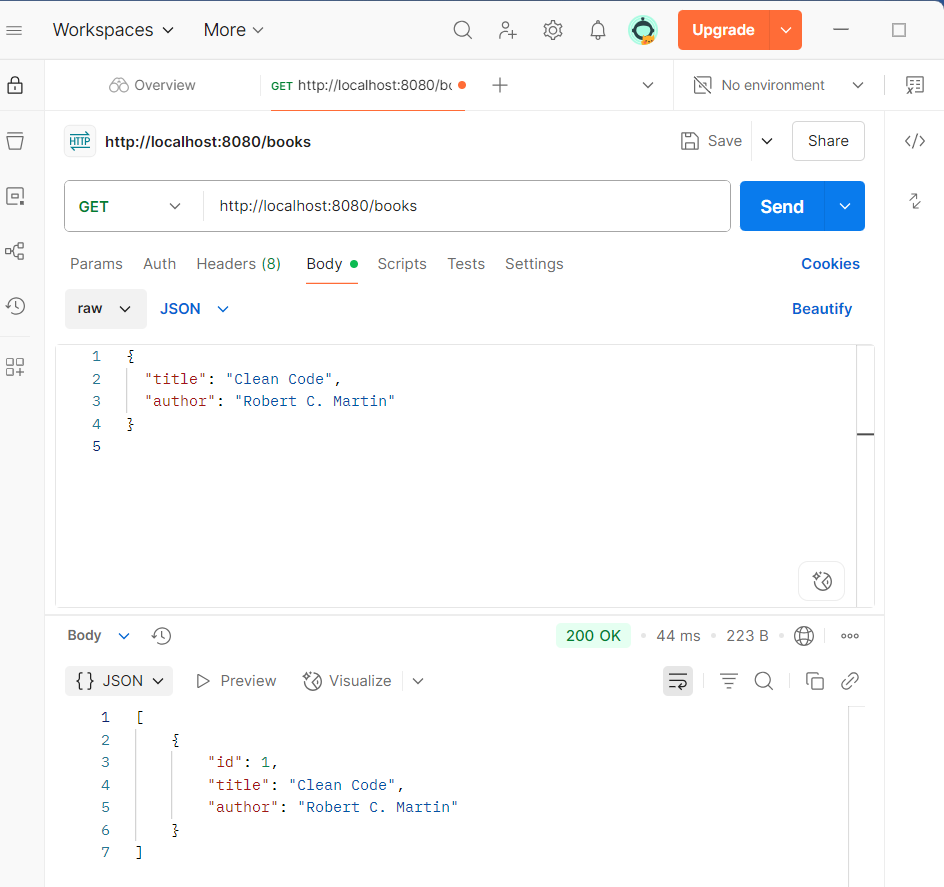
**OUTPUT:**

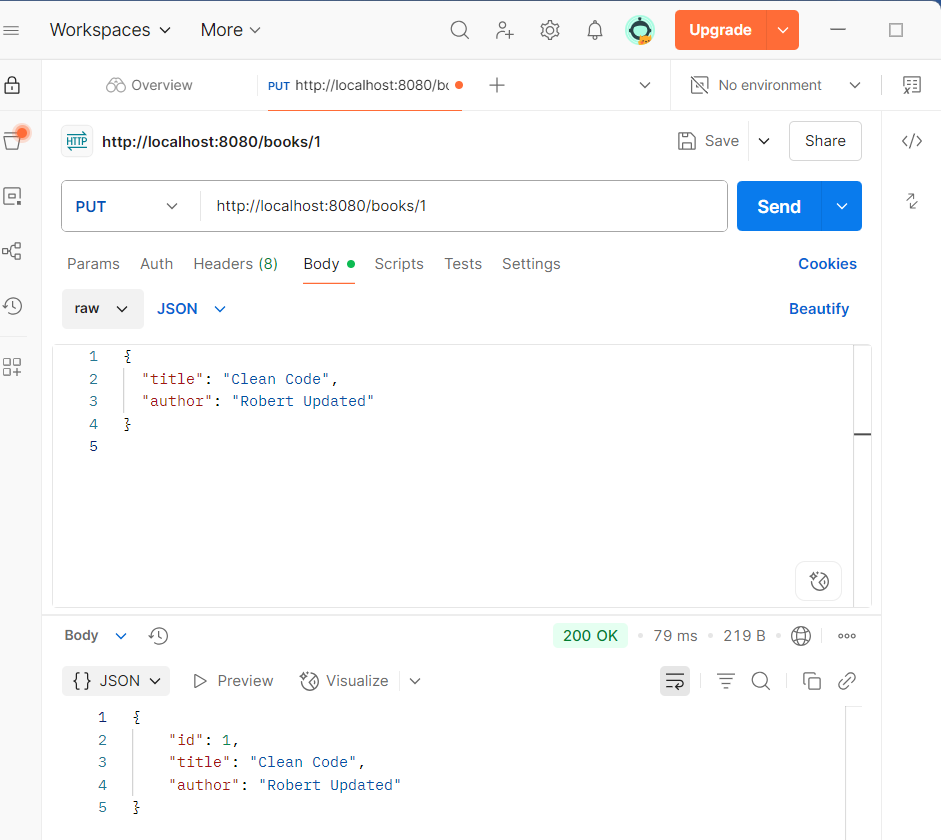


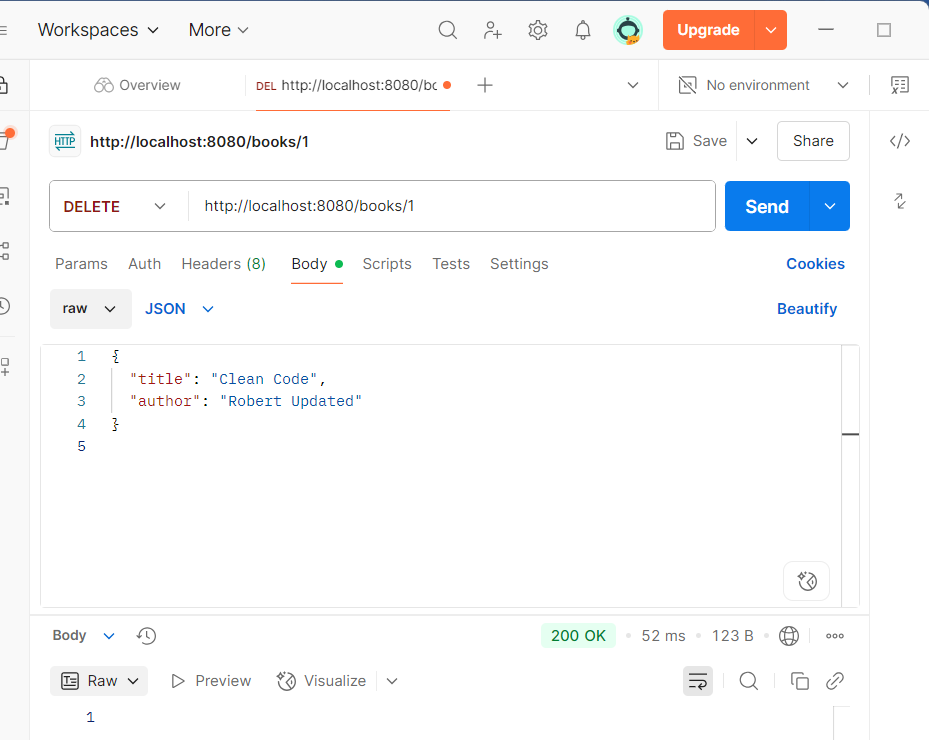
**TESTING WITH POSTMAN:**

**POST METHOD**

**GET METHOD**



**UPDATING WITH PUT METHOD:**



**DELETE METHOD**

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

**Enable AspectJ Support:**

<?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"*

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"*>

<aop:aspectj-autoproxy />

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

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

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

</bean>

<bean id=*"loggingAspect"* class=*"com.library.aspect.LoggingAspect"* />

</beans>

**LoggingAspect.java:**

**package** com.library.aspect;

**import** org.aspectj.lang.ProceedingJoinPoint;

**import** org.aspectj.lang.annotation.Around;

**import** org.aspectj.lang.annotation.Aspect;

@Aspect

**public** **class** LoggingAspect {

@Around("execution(\* com.library.service.\*.\*(..))")

**public** Object logExecutionTime(ProceedingJoinPoint joinPoint) **throws** Throwable {

**long** start = System.*currentTimeMillis*();

Object result = joinPoint.proceed(); // Proceed with the actual method call

**long** duration = System.*currentTimeMillis*() - start;

System.***out***.println("Execution of " + joinPoint.getSignature() + " took " + duration + " ms");

**return** result;

}

**Exercise 6: Configuring Beans with Annotations**

**Enable Component Scanning:**

<?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: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/context*

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

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

</beans>

**Annotate the Classes:**

**BookRepository.java:**

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

public void saveBook() {

System.out.println("Book saved to the repository.");

}

}

**BookService.java:**

package com.library.service;

import com.library.repository.BookRepository;

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

import org.springframework.stereotype.Service;

@Service

public class BookService {

@Autowired // Inject BookRepository automatically

private BookRepository bookRepository;

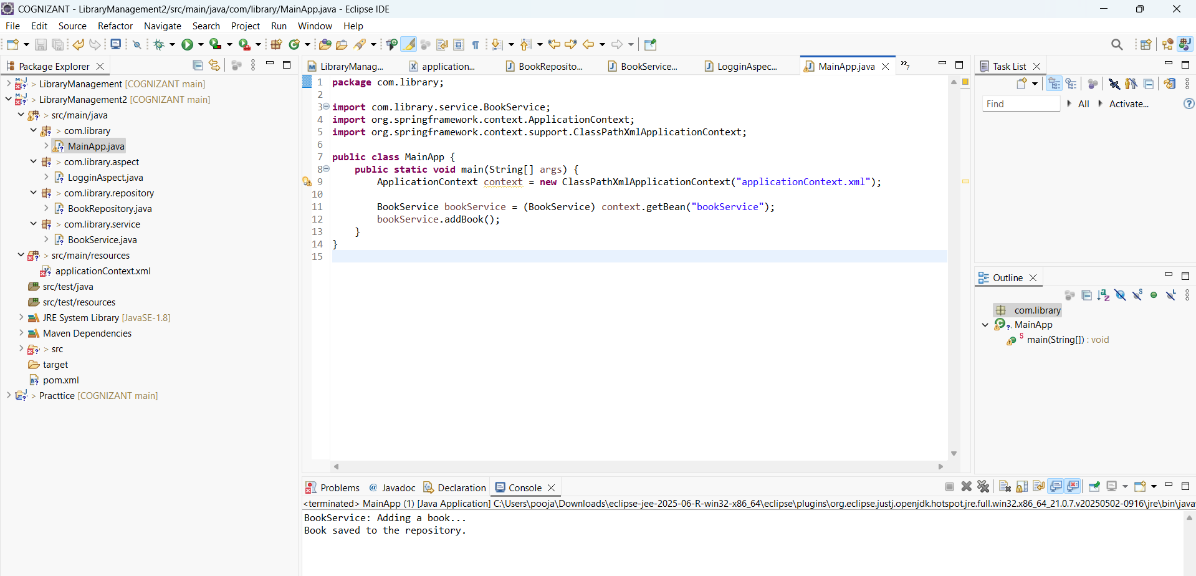
public void addBook() {

System.out.println("BookService: Adding a book...");

bookRepository.saveBook();

}

}



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

**LoggingAspect.java:**

package com.library.aspect;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

@Aspect

public class LogginAspect {

@Before("execution(\* com.library.service.\*.\*(..))")

public void beforeMethod(JoinPoint joinPoint) {

System.out.println("→ [Before] Calling method: " + joinPoint.getSignature().getName());

}

@After("execution(\* com.library.service.\*.\*(..))")

public void afterMethod(JoinPoint joinPoint) {

System.out.println("← [After] Method finished: " + joinPoint.getSignature().getName());

}

}

**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"*

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"*>

<aop:aspectj-autoproxy />

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

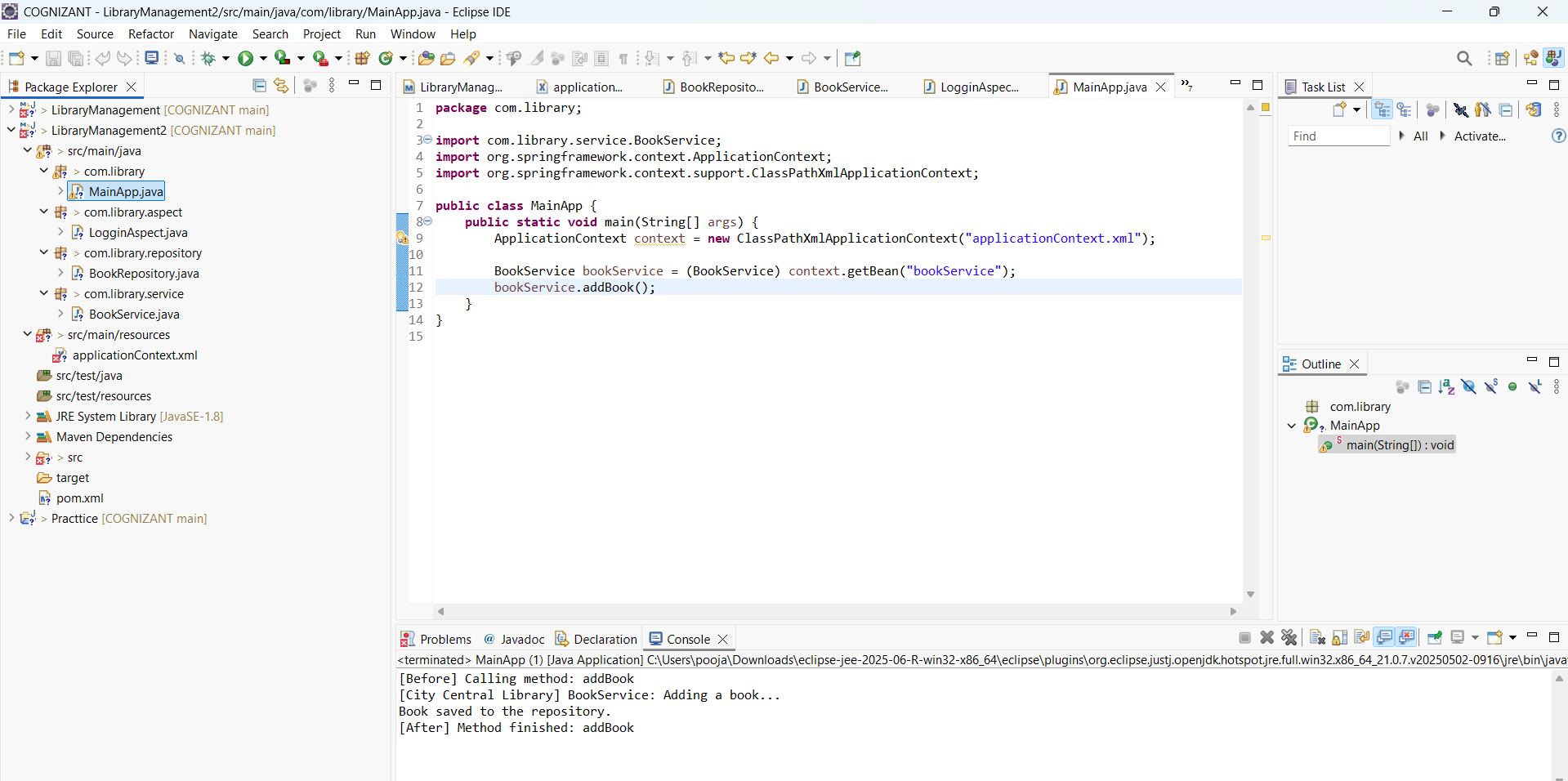
<bean id=*"bookService"* class=*"com.library.service.BookService"*>

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

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

</bean>

<bean id=*"logginAspect"* class=*"com.library.aspect.LogginAspect"* />

</beans>