**Spring Core Maven**

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

package com.library.librarymanagement.repository;

public class BookRepository {

public void save() {

System.out.println("Book saved to database.");

}

}

package com.library.librarymanagement.service;

import com.library.librarymanagement.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook() {

System.out.println("Adding book...");

bookRepository.save();

}

}

package com.library.librarymanagement;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.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);

bookService.addBook();

}

}

<?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.librarymanagement.repository.BookRepository"/>

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

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

</bean>

</beans>

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

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>3.5.3</version>

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

</parent>

<groupId>com.library</groupId>

<artifactId>librarymanagement</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>librarymanagement</name>

<description>Demo project for Spring Boot</description>

<url/>

<licenses>

<license/>

</licenses>

<developers>

<developer/>

</developers>

<scm>

<connection/>

<developerConnection/>

<tag/>

<url/>

</scm>

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.33</version>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

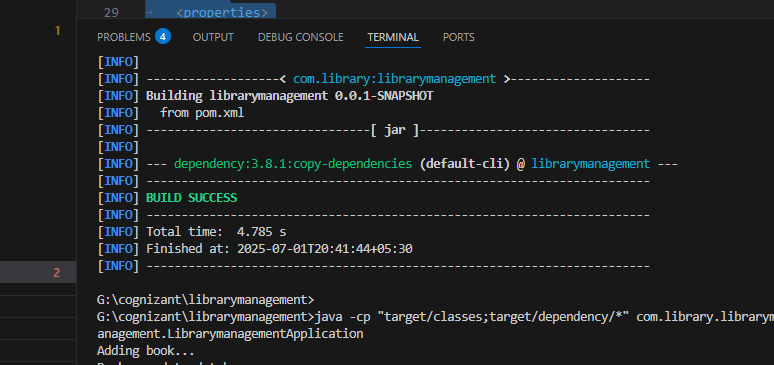
<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>



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

package com.library.librarymanagement.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.librarymanagement.service.\*.\*(..))")

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long start = System.currentTimeMillis();

Object result = joinPoint.proceed(); // Execute the method

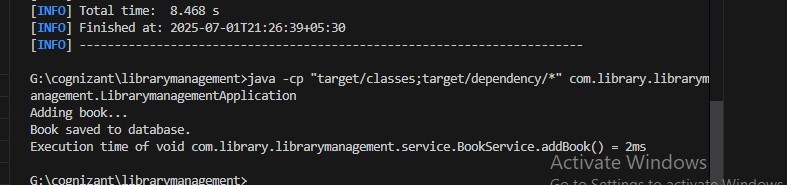
long end = System.currentTimeMillis();

System.out.println("Execution time of " + joinPoint.getSignature() + " = " + (end - start) + "ms");

return result;

}

}



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

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

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>3.5.3</version>

<relativePath/>

</parent>

<groupId>com.library</groupId>

<artifactId>librarymanagement</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>librarymanagement</name>

<description>Spring AOP Logging Project</description>

<properties>

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

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

</plugin>

<plugin>

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

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

<version>3.10.1</version>

<configuration>

<source>17</source>

<target>17</target>

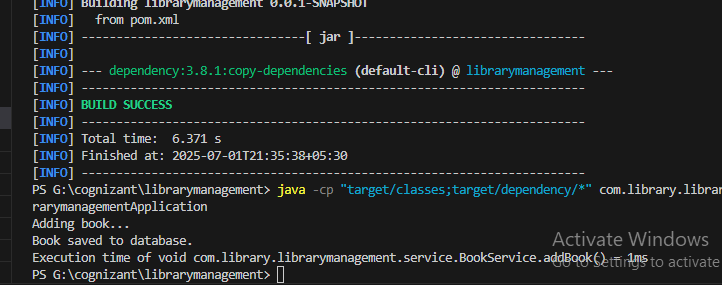
</configuration>

</plugin>

</plugins>

</build>

</project>



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

package com.library.librarymanagement;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.librarymanagement.service.BookService;

public class LibrarymanagementApplication {

public static void main(String[] args) {

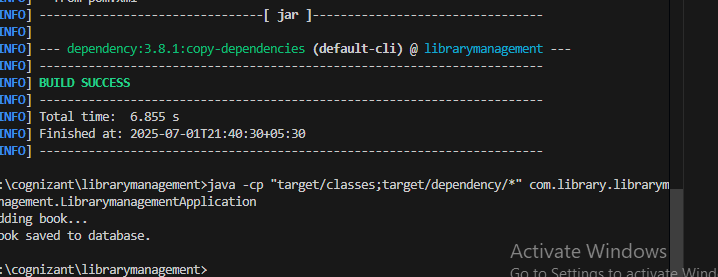
ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

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

bookService.addBook();

}

}



**Exercise 6: Configuring Beans with Annotations**

package com.library.librarymanagement.service;

import com.library.librarymanagement.repository.BookRepository;

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

import org.springframework.stereotype.Service;

@Service

public class BookService {

@Autowired

private BookRepository bookRepository;

public void addBook() {

System.out.println("Adding book...");

bookRepository.save();

}

}

<?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

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

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

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

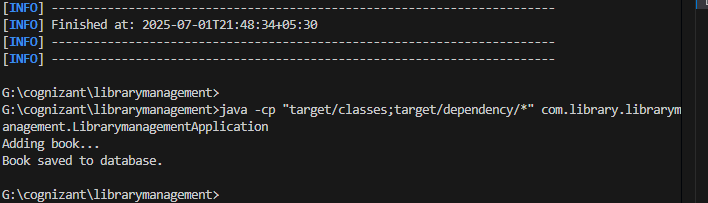
http://www.springframework.org/schema/aop

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

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

<aop:aspectj-autoproxy />

</beans>



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

package com.library.librarymanagement.service;

import com.library.librarymanagement.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public BookService() {

System.out.println("Default constructor called");

}

public BookService(BookRepository bookRepository) {

this.bookRepository = bookRepository;

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

}

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

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

}

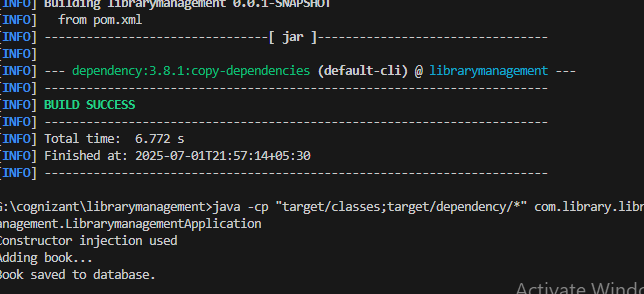
public void addBook() {

System.out.println("Adding book...");

bookRepository.save();

}

}



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

package com.library.librarymanagement.aspect;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class LoggingAspect {

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

public void logBefore(JoinPoint joinPoint) {

System.out.println("👉 BEFORE: " + joinPoint.getSignature());

}

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

public void logAfter(JoinPoint joinPoint) {

System.out.println("✅ AFTER: " + joinPoint.getSignature());

}

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

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long start = System.currentTimeMillis();

Object result = joinPoint.proceed();

long end = System.currentTimeMillis();

System.out.println("⏱ Execution time of " + joinPoint.getSignature() + " = " + (end - start) + " ms");

return result;

}

}

package com.library.librarymanagement.service;

import org.springframework.stereotype.Service;

@Service

public class BookService {

public void addBook() {

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

}

}

package com.library.librarymanagement;

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

import org.springframework.boot.CommandLineRunner;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import com.library.librarymanagement.service.BookService;

@SpringBootApplication

public class LibrarymanagementApplication implements CommandLineRunner {

@Autowired

private BookService bookService;

public static void main(String[] args) {

SpringApplication.run(LibrarymanagementApplication.class, args);

}

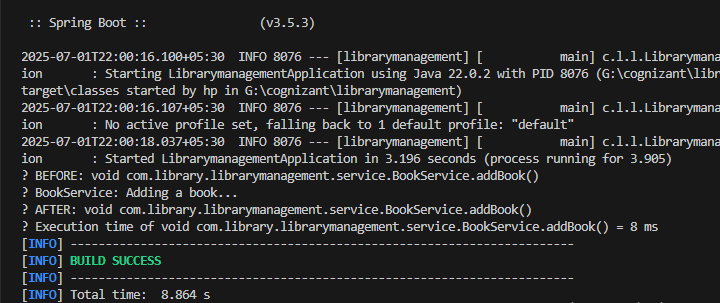
@Override

public void run(String... args) {

bookService.addBook();

}

}



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

package com.library.librarymanagement.controller;

import com.library.librarymanagement.service.BookService;

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

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

@RestController

@RequestMapping("/books")

public class BookController {

@Autowired

private BookService bookService;

@PostMapping

public String addBook() {

return bookService.addBook();

}

@GetMapping

public String getBooks() {

return "📖 List of books";

}

}

package com.library.librarymanagement.service;

import org.springframework.stereotype.Service;

@Service

public class BookService {

public String addBook() {

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

return "✅ Book saved";

}

}

package com.library.librarymanagement;

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);

}

}

