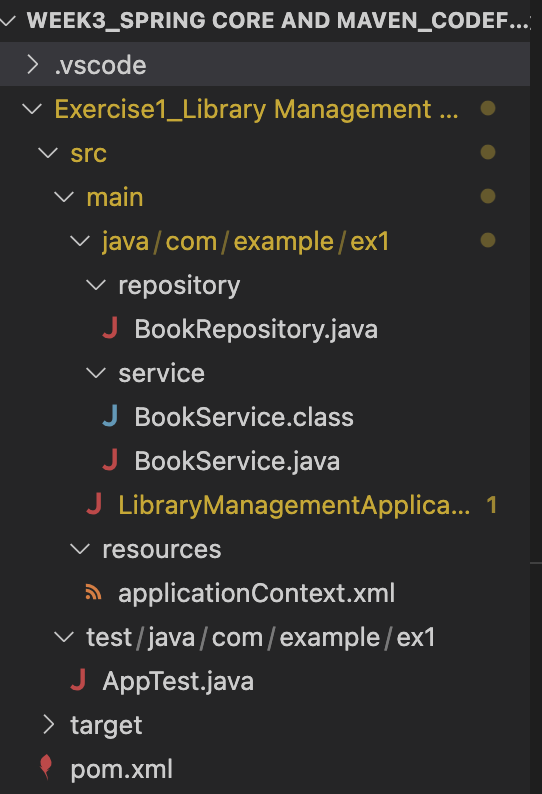
**Exercise 1: Configuring a Basic Spring Application (MANDATORY)**

**Folder Structure:**

****

**BookRepository.java:**

package com.example.ex1.repository;

public class BookRepository {

public void fetchBooks() {

System.out.println("BookRepository: Fetching books from the database.");

}

}

**BookService.java:**

package com.example.ex1.service;

public class BookService {

public void displayBookInfo() {

System.out.println("BookService: Displaying book information.");

}

}

**LibraryManagementApplication.java:**

package com.example.ex1;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.ex1.repository.BookRepository;

import com.example.ex1.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

BookService bs = c.getBean("bookService", BookService.class);

BookRepository br = c.getBean("bookRepository", BookRepository.class);

bs.displayBookInfo();

br.fetchBooks();

}

}

**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="bookService" class="com.example.ex1.service.BookService"/>

<bean id="bookRepository" class="com.example.ex1.repository.BookRepository"/>

</beans>

**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.ex1</groupId>

<artifactId>library\_management</artifactId>

<version>1.0-SNAPSHOT</version>

<name>library\_management</name>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<version>3.8.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

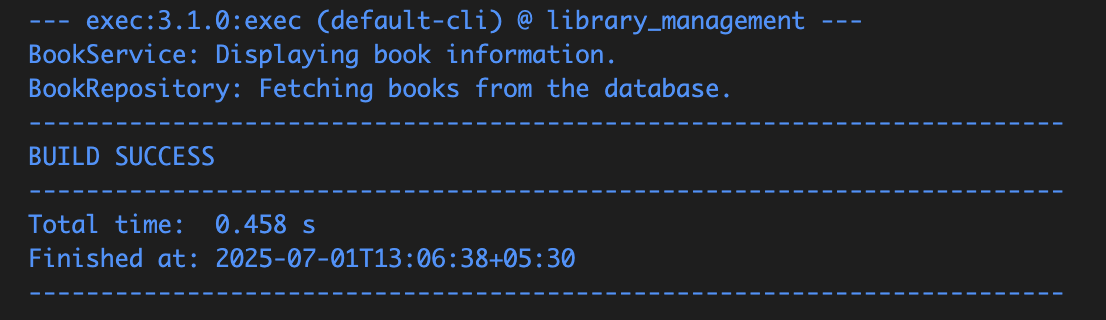
</plugin>

</plugins>

</build>

</project>

**Output:**

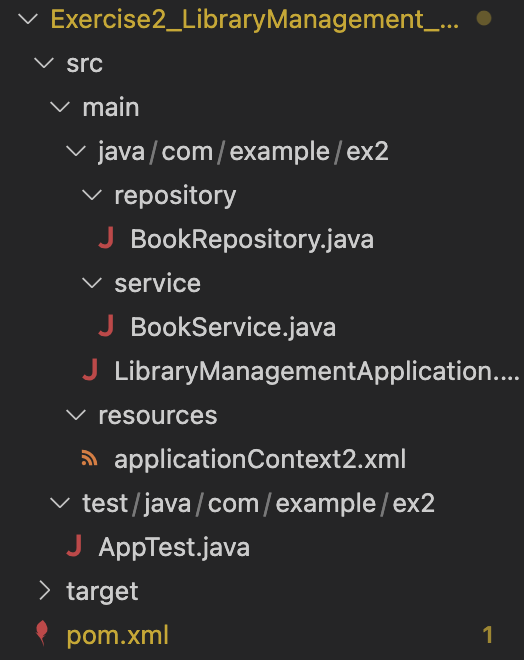


BookService: Displaying book information.

BookRepository: Fetching books from the database.

**Exercise 2: Implementing Dependency Injection (MANDATORY)**

**Folder Structure:**



**BookRepository.java:**

package com.example.ex2.repository;

public class BookRepository {

public void fetchBooks() {

System.out.println("BookRepository: Fetching books from the database.");

}

}

**BookService.java:**

package com.example.ex2.service;

import com.example.ex2.repository.BookRepository;

public class BookService {

private BookRepository br;

public void setBookRepository(BookRepository br) {

this.br = br;

}

public void displayBookInfo() {

System.out.println("BookService: Displaying book information.");

br.fetchBooks();

}

}

**LibraryManagementApplication:**

package com.example.ex2;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.ex2.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

ApplicationContext c = new ClassPathXmlApplicationContext("applicationContext2.xml");

BookService bs = c.getBean("bookService", BookService.class);

bs.displayBookInfo();

}

}

**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.ex2.repository.BookRepository" /> <bean id="bookService" class="com.example.ex2.service.BookService">

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

</bean>

</beans>

**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.ex2</groupId>

<artifactId>library\_management\_di</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<version>3.8.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>3.1.0</version>

<configuration>

<mainClass>com.example.ex2.LibraryManagementApplication</mainClass>

</configuration>

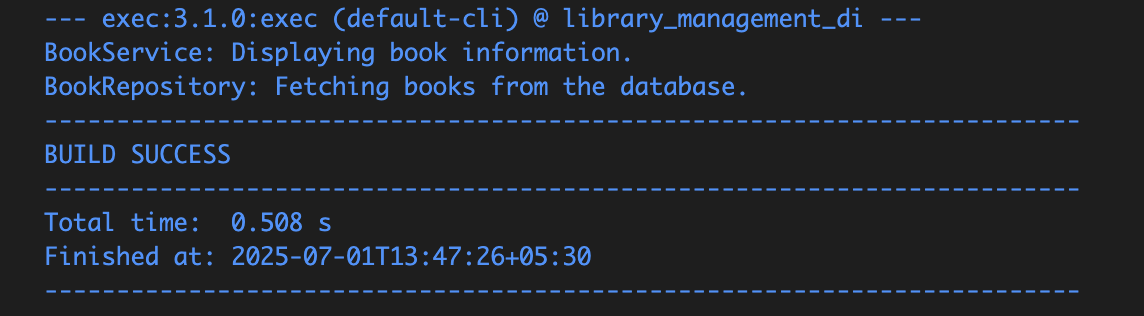
</plugin>

</plugins>

</build>

</project>

**Output:**

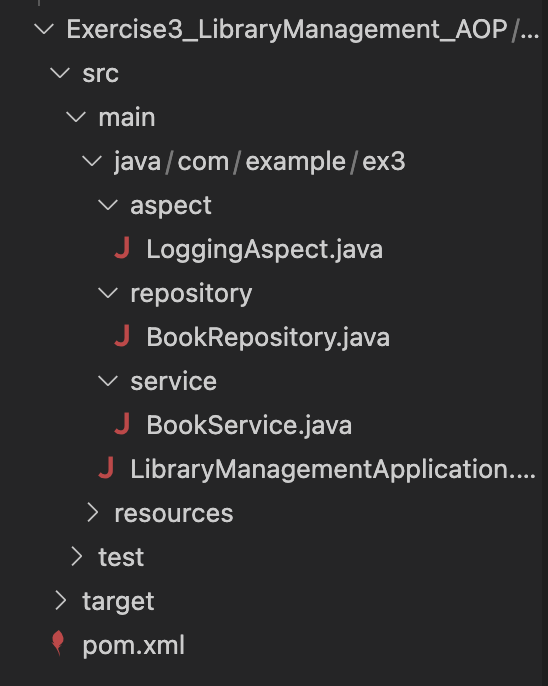


BookService: Displaying book information.

BookRepository: Fetching books from the database.

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

**Folder Structure**



**LoggingAspect.java:**

package com.example.ex3.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.example.ex3.service.\*.\*(..))")

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long s = System.currentTimeMillis();

Object res = joinPoint.proceed();

long e = System.currentTimeMillis();

System.out.println("Method " + joinPoint.getSignature().getName() +" executed in " + (e - s) + "ms");

return res;

}

}

**BookRepository.java:**

package com.example.ex3.repository;

public class BookRepository {

public void fetchBooks() {

System.out.println(" BookRepository: Fetching books from DB...");

}

}

**BookService.java:**

package com.example.ex3.service;

import com.example.ex3.repository.BookRepository;

public class BookService {

private BookRepository br;

public void setBookRepository(BookRepository br) {

this.br = br;

}

public void displayBookInfo() {

System.out.println("BookService: Displaying book information...");

br.fetchBooks();

}

}

**LibraryManagementApplication:**

package com.example.ex3;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.ex3.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

try (ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml")) {

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

bs.displayBookInfo();

}

}

}

**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.example.ex3.repository.BookRepository" />

<bean id="bookService" class="com.example.ex3.service.BookService">

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

</bean>

<bean id="loggingAspect" class="com.example.ex3.aspect.LoggingAspect" />

</beans>

**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.example.ex3</groupId>

<artifactId>library\_management\_aop</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.release>21</maven.compiler.release>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>3.1.0</version>

<configuration>

<mainClass>com.example.ex3.LibraryManagementApplication</mainClass>

</configuration>

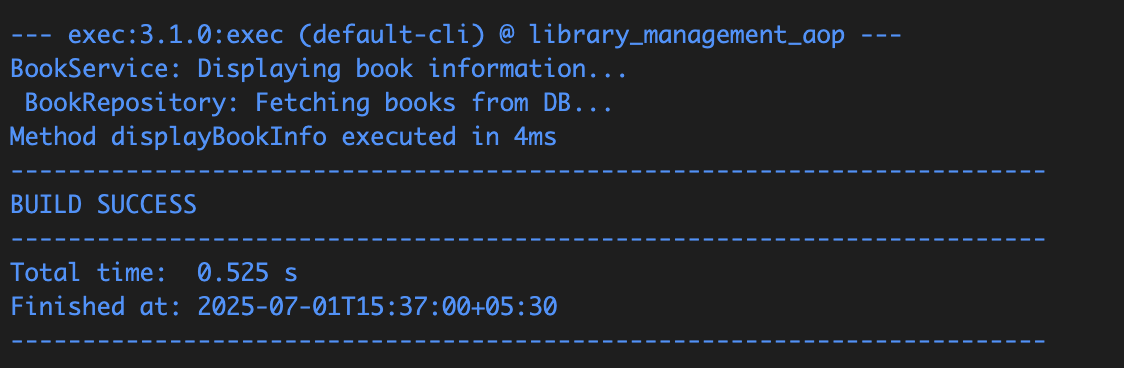
</plugin>

</plugins>

</build>

</project>

**Output:**



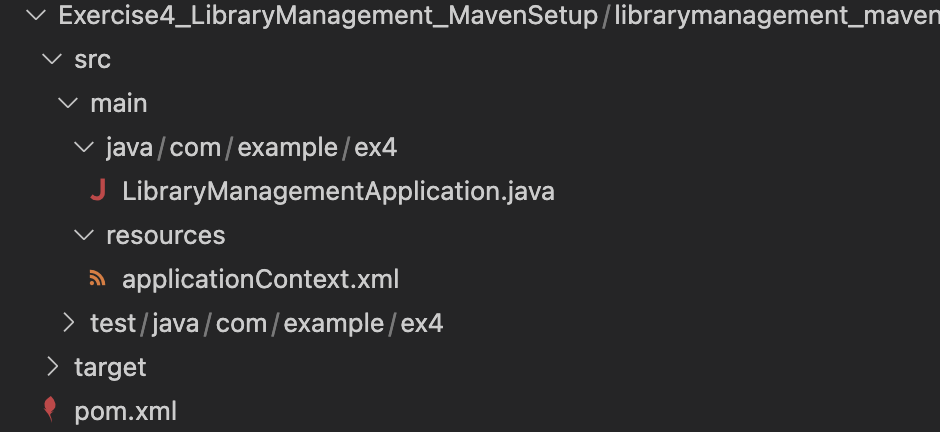
BookService: Displaying book information...

BookRepository: Fetching books from DB...

Method displayBookInfo executed in 4ms

**Exercise 4: Creating and Configuring a Maven Project (MANDATORY)**

**Folder Structure:**



**LibraryManagementApplication.java:**

package com.example.ex4;

public class LibraryManagementApplication {

public static void main(String[] args) {

System.out.println("Library Management Project Setup Completed!");

}

}

**applicationContext.xml:**

<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="exampleBean" class="java.lang.String">

<constructor-arg value="Spring Bean Initialized" />

</bean>

</beans>

**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.example.ex4</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>5.3.36</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.36</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<version>3.8.0</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

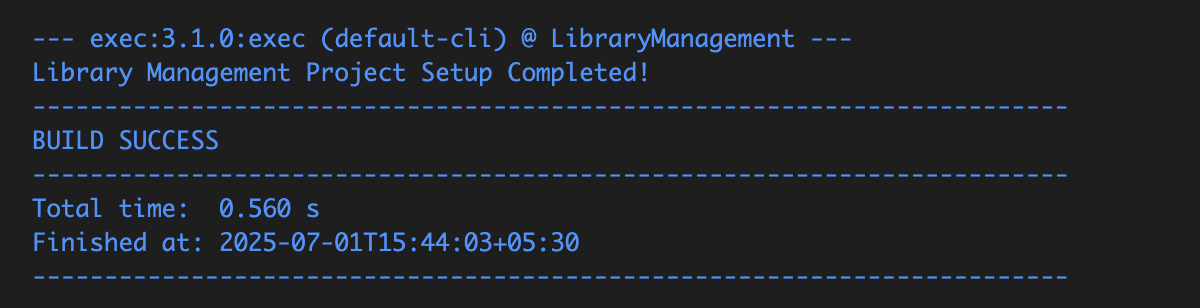
</plugin>

</plugins>

</build>

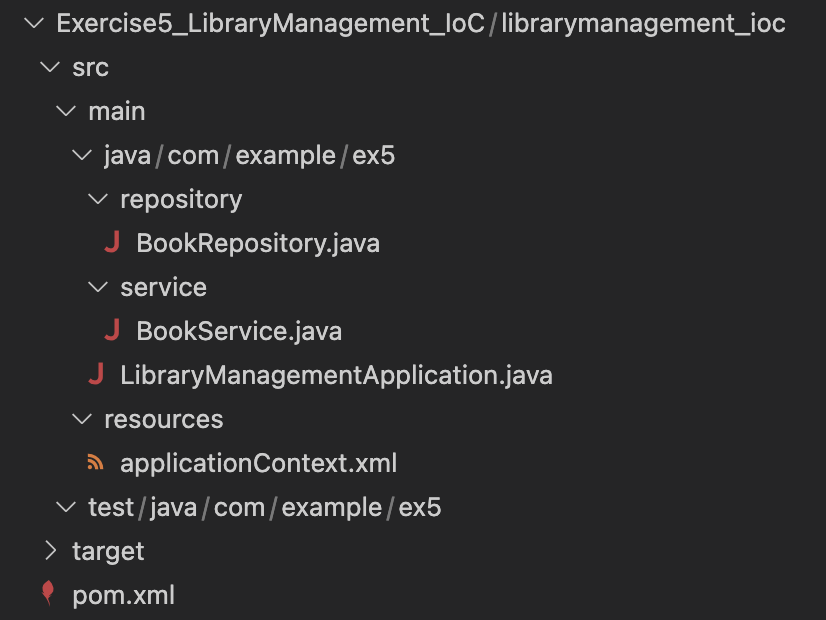
</project>

**Output:**

****

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

**Folder Structure:**



**BookRepository.java:**

package com.example.ex5.repository;

public class BookRepository {

public void fetchBooks() {

System.out.println(" BookRepository: Fetching books from the database...");

}

}

**BookService.java:**

package com.example.ex5.service;

import com.example.ex5.repository.BookRepository;

public class BookService {

private BookRepository br;

public void setBookRepository(BookRepository br) {

this.br = br;

}

public void displayBookInfo() {

System.out.println(" BookService: Displaying book information...");

br.fetchBooks();

}

}

**LibraryManagementApplication.java:**

package com.example.ex5;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.ex5.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

try (ClassPathXmlApplicationContext c =

new ClassPathXmlApplicationContext("applicationContext.xml")) {

BookService bs = c.getBean("bookService", BookService.class);

bs.displayBookInfo();

}

}

}

**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.ex5.repository.BookRepository" />

<bean id="bookService" class="com.example.ex5.service.BookService">

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

</bean>

</beans>

**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.example.ex5</groupId>

<artifactId>LibraryManagementIoC</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

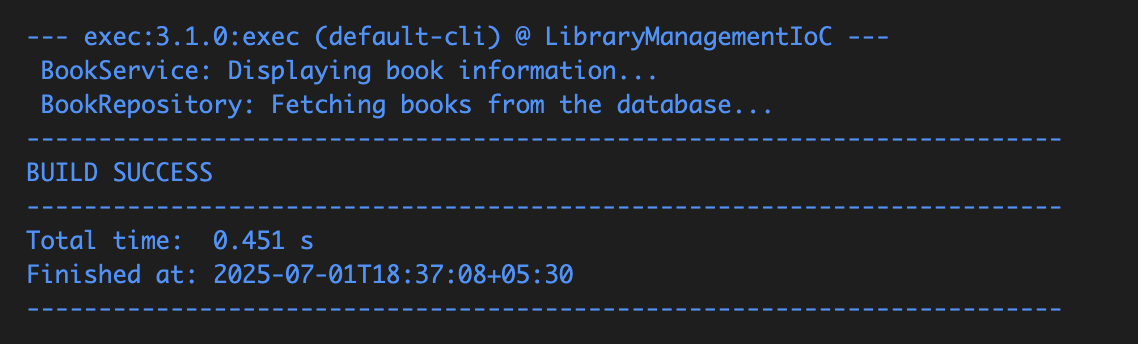
<version>5.3.36</version>

</dependency>

</dependencies>

</project>

**Output:**

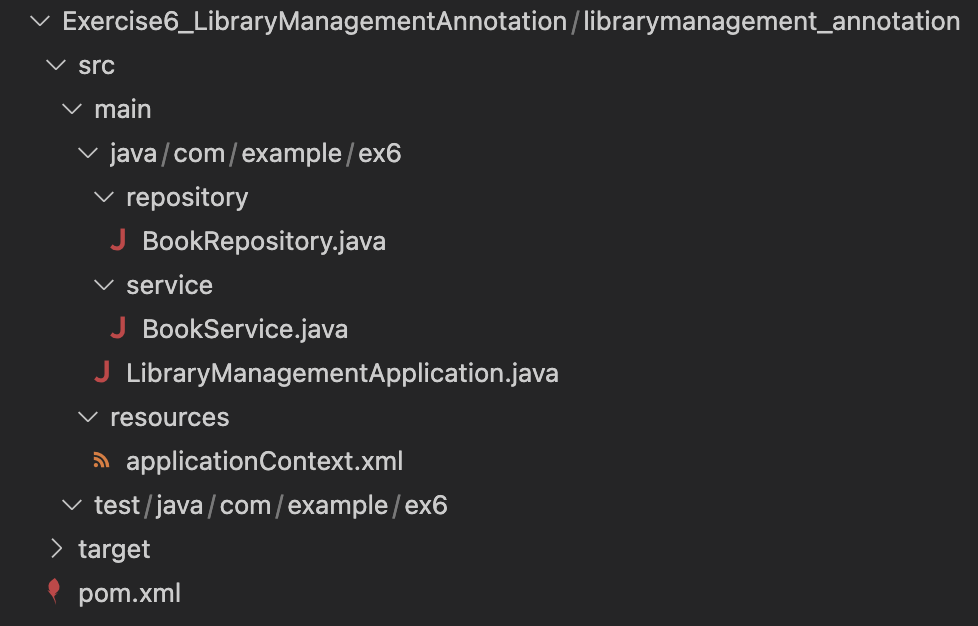
****

BookService: Displaying book information...

BookRepository: Fetching books from the database...

**Exercise 6: Configuring Beans with Annotations**

**Folder Structure:**



BookRepository:

package com.example.ex6.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

public void fetchBooks() {

System.out.println("📘 BookRepository: Fetching books from the database...");

}

}

BookService.java:

package com.example.ex6.service;

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

import org.springframework.stereotype.Service;

import com.example.ex6.repository.BookRepository;

@Service

public class BookService {

private BookRepository br;

@Autowired

public void setBookRepository(BookRepository br) {

this.br = br;

}

public void displayBookInfo() {

System.out.println("📚 BookService: Displaying book info...");

br.fetchBooks();

}

}

**LibraryManagementApplication.java:**

package com.example.ex6;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.ex6.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

try (ClassPathXmlApplicationContext c =

new ClassPathXmlApplicationContext("applicationContext.xml")) {

BookService bs = c.getBean(BookService.class);

bs.displayBookInfo();

}

}

}

**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: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.example.ex6" />

</beans>

**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.example.ex6</groupId>

<artifactId>library\_management\_annotations</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.release>21</maven.compiler.release>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>3.1.0</version>

<configuration>

<mainClass>com.example.ex6.LibraryManagementApplication</mainClass>

</configuration>

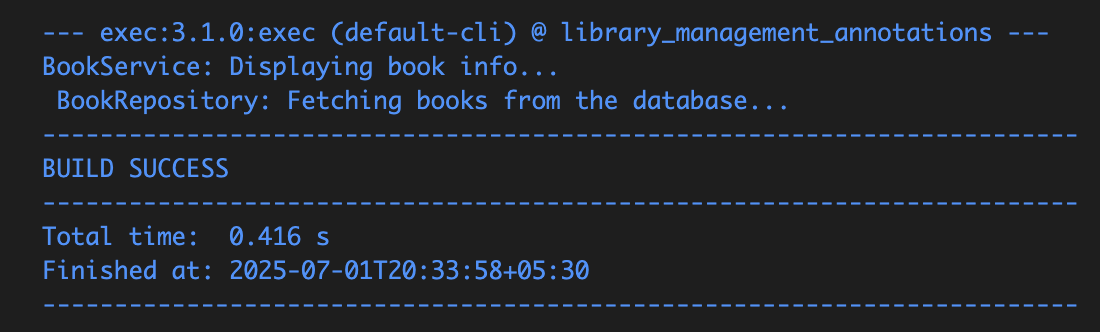
</plugin>

</plugins>

</build>

</project>

**Output:**



BookService: Displaying book info...

BookRepository: Fetching books from the database…

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

**Folder Structure:**



**BookRepository.java:**

package com.example.ex7.repository;

public class BookRepository {

public void printRepositoryInfo() {

System.out.println(" BookRepository: Accessing book data...");

}

}

**BookService.java:**

package com.example.ex7.service;

import com.example.ex7.repository.BookRepository;

public class BookService {

private BookRepository br;

public BookService(BookRepository br) {

this.br = br;

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

}

public void setBookRepository(BookRepository br) {

this.br = br;

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

}

public void displayBookInfo() {

System.out.println(" BookService: Displaying book information.");

br.printRepositoryInfo();

}

}

**LibraryManagementApplication.java:**

package com.example.ex7;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.example.ex7.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

BookService bs = (BookService) c.getBean("bookService");

bs.displayBookInfo();

}

}

**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.ex7.repository.BookRepository" />

<bean id="bookService" class="com.example.ex7.service.BookService">

<constructor-arg ref="bookRepository" />

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

</bean>

</beans>

**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.ex7</groupId>

<artifactId>constructor\_setter\_injection</artifactId>

<version>1.0-SNAPSHOT</version>

<name>constructor\_setter\_injection</name>

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.8</maven.compiler.source>

<maven.compiler.target>1.8</maven.compiler.target>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.32</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-beans</artifactId>

<version>5.3.32</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<pluginManagement>

<plugins>

<plugin>

<artifactId>maven-clean-plugin</artifactId>

<version>3.1.0</version>

</plugin>

<plugin>

<artifactId>maven-resources-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

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

<version>3.8.0</version>

</plugin>

<plugin>

<artifactId>maven-surefire-plugin</artifactId>

<version>2.22.1</version>

</plugin>

<plugin>

<artifactId>maven-jar-plugin</artifactId>

<version>3.0.2</version>

</plugin>

<plugin>

<artifactId>maven-install-plugin</artifactId>

<version>2.5.2</version>

</plugin>

<plugin>

<artifactId>maven-deploy-plugin</artifactId>

<version>2.8.2</version>

</plugin>

<plugin>

<artifactId>maven-site-plugin</artifactId>

<version>3.7.1</version>

</plugin>

<plugin>

<artifactId>maven-project-info-reports-plugin</artifactId>

<version>3.0.0</version>

</plugin>

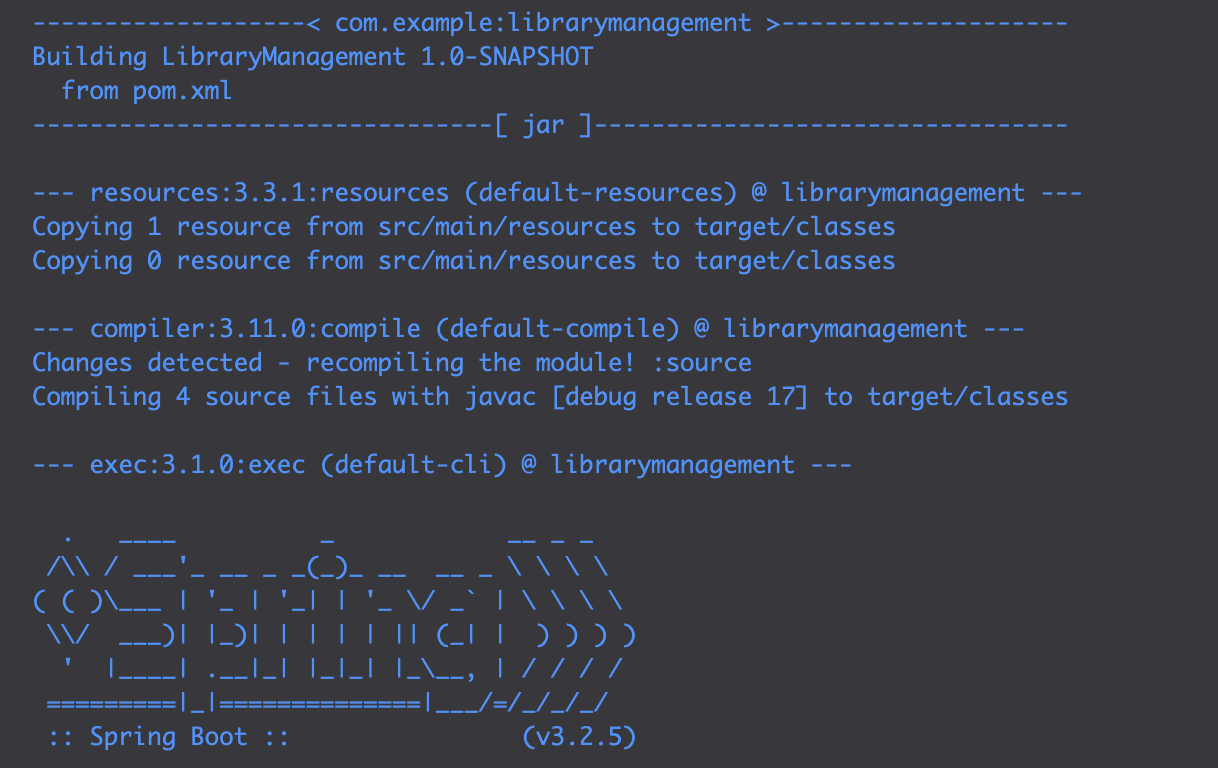
</plugins>

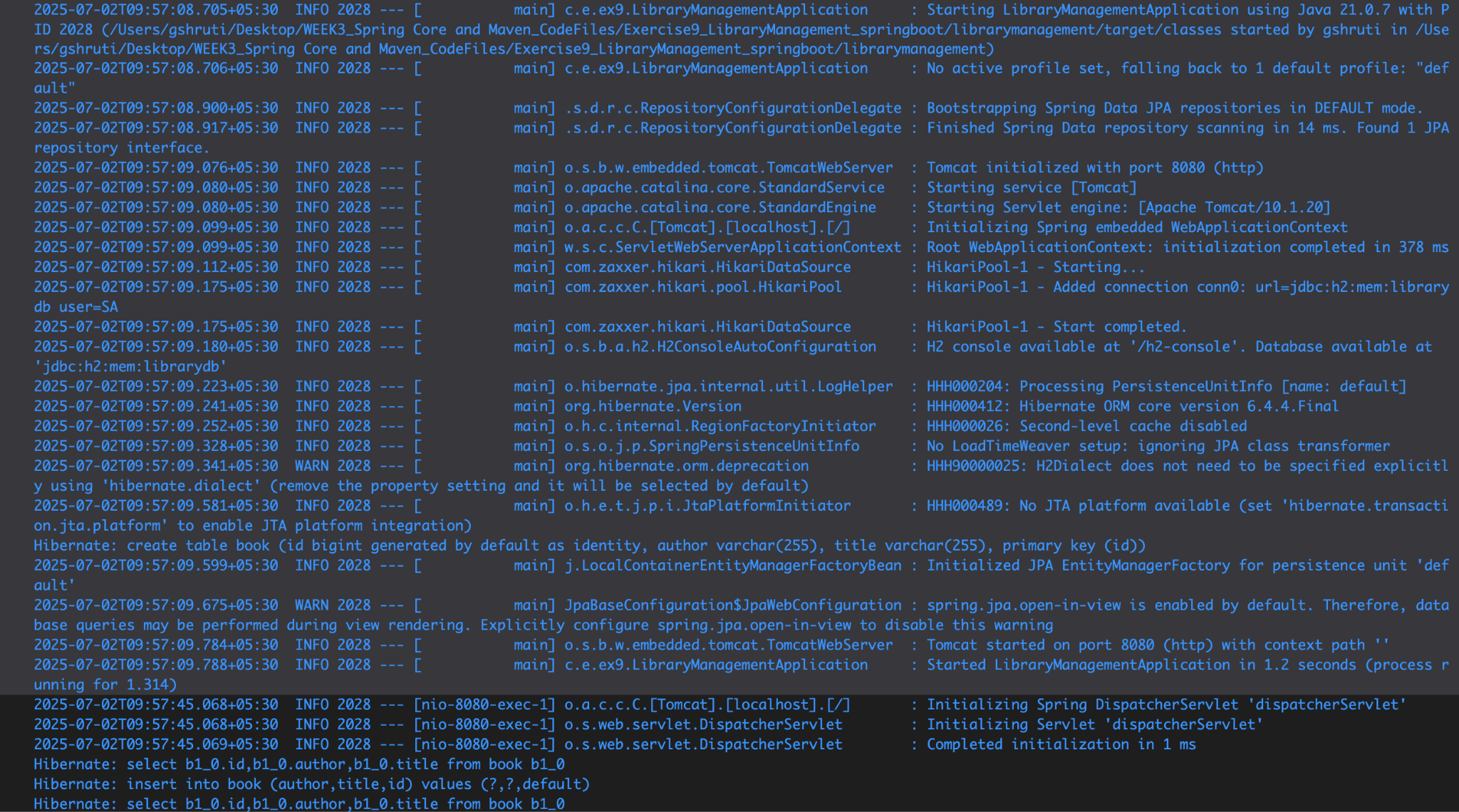
</pluginManagement>

</build>

</project>

**Output:**





**Run and Test the Application**

