**Spring Core Maven**

**EXERCISE 2: Implementing Dependency Injection**

**Source Code**

**Steps:**

Modify the XML Configuration:

1. <beans xmlns="http://www.springframework.org/schema/beans"
2. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3. xsi:schemaLocation="http://www.springframework.org/schema/beans
4. http://www.springframework.org/schema/beans/spring-beans.xsd">
5. <bean id="bookRepository" class="com.library.repository.BookRepository"/>
6. <bean id="bookService" class="com.library.service.BookService">
7. <property name="bookRepository" ref="bookRepository"/>
8. </bean>
9. </beans>

Update the BookService Class:}

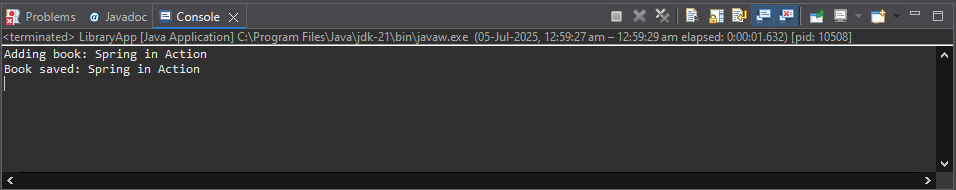
1. package com.library.service;
2. import com.library.repository.BookRepository;
3. public class BookService {
4. private BookRepository bookRepository;
5. // Setter for Dependency Injection
6. public void setBookRepository(BookRepository bookRepository) {
7. this.bookRepository = bookRepository;
8. }
9. public void addBook(String bookName) {
10. System.***out***.println("Adding book: " + bookName);
11. bookRepository.saveBook(bookName);
12. }
13. }

Test the Configuration:

1. package com.library.main;
2. import org.springframework.context.ApplicationContext;
3. import org.springframework.context.support.ClassPathXmlApplicationContext;
4. import com.library.service.BookService;
5. public class LibraryApp {
6. public static void main(String[] args) {
7. ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");
8. BookService bookService = (BookService) context.getBean("bookService");
9. bookService.addBook("Spring in Action");
10. }
11. }

* **Right click on LibraryApp.java -> Run As -> Java Application**

**Output:**

****