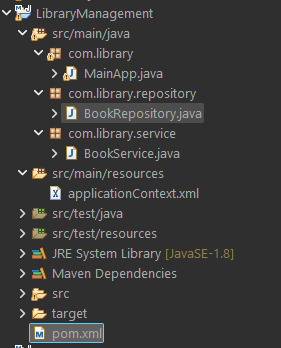
**Exercise 1 : Configuring a Basic Spring Application**

**Scenario**: Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

**Steps :**

1. Set Up a Spring Project **:**
2. A Maven project named **LibraryManagement** is created.
3. Spring Core dependencies is added in the **pom.xml** file.
4. <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">
5. <modelVersion>4.0.0</modelVersion>
6. <groupId>com.library</groupId>
7. <artifactId>LibraryManagement</artifactId>
8. <version>0.0.1-SNAPSHOT</version>
9. <dependencies>
10. <dependency>
11. <groupId>org.springframework</groupId>
12. <artifactId>spring-context</artifactId>
13. <version>5.3.31</version>
14. </dependency>
15. </dependencies>
16. </project>
17. Configure the Application Context:
18. An XML configuration file named **applicationContext.xml** in the **src/main/resources** directory is created.



1. Beans for **BookService** and **BookRepository** in the XML file.
2. <?xml version="1.0" encoding="UTF-8"?>
3. <beans xmlns="http://www.springframework.org/schema/beans"
4. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5. xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
6. <!-- BookRepository Bean -->
7. <bean id="bookRepository" class="com.library.repository.BookRepository" />
8. <!-- BookService Bean with dependency injection -->
9. <bean id="bookService" class="com.library.service.BookService">
10. <property name="bookRepository" ref="bookRepository"/>
11. </bean>
12. </beans>
13. Define Service and Repository Classes:
14. A package named **com.library.service** and a class **BookService** is added.
15. package com.library.service;
16. import com.library.repository.BookRepository;
17. public class BookService {
18. private BookRepository bookRepository;
19. public void setBookRepository(BookRepository bookRepository) {
20. this.bookRepository = bookRepository;
21. }
22. public void addBook(String bookName) {
23. System.***out***.println("Service: Adding book...");
24. bookRepository.saveBook(bookName);
25. }
26. }

b. A package named **com.library.repository** and a class **BookRepositor** is added.

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

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

}

}

1. Run the Application:

A main class **MainApp** is created.

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("The Spring Mastery");

}

}

**Output :**

A screenshot of a computer

AI-generated content may be incorrect.