**Module 1 : Spring Core and Maven**

**Spring Core\_Maven Exercises**

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

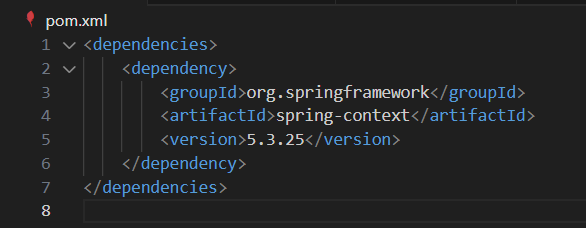
**Scenario:** Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

**Description**: The Spring Framework is an open-source application framework and inversion of control container for the Java platform. It provides comprehensive infrastructure support for developing robust, enterprise-level Java applications.

**Solution:**

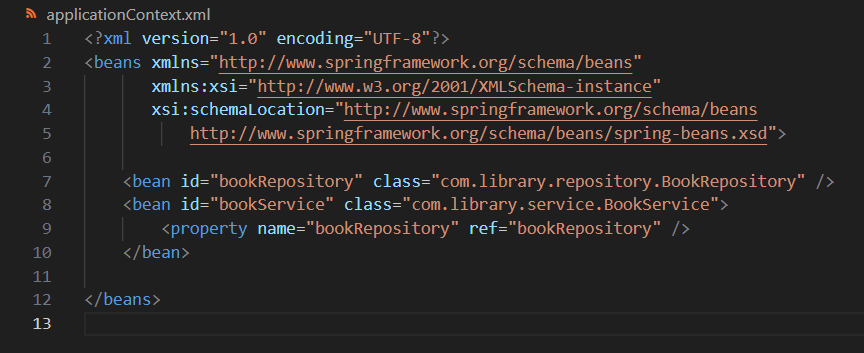
**Step 1: Set Up a Spring Project**

* Create a Maven project named **LibraryManagement**.
* Add the following Spring Core dependency to the pom.xml file:



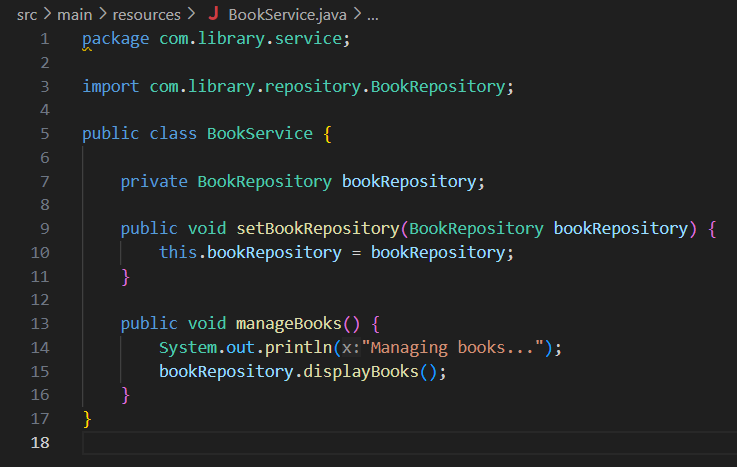
**Step 2: Configure the Application Context**

* **Create an XML file named applicationContext.xml in the src/main/resources directory.**
* **Define beans for BookService and BookRepository:**

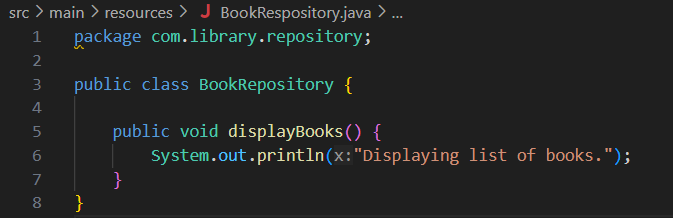
****

**Step 3: Define Service and Repository Classes**

* **BookService.java** (package: com.library.service)

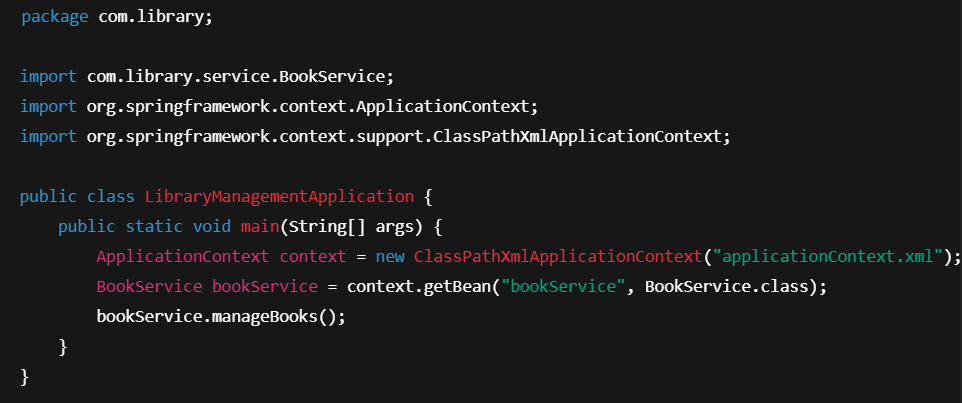


* **BookRepository.java** (package: com.library.service)



**Step 4: Run the Application**

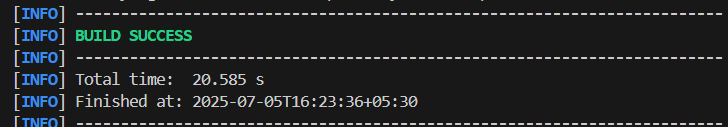
* **LibraryManagementApplication.java (package: com.library)**

****

**Step 5: Run the Following commands to test and run**

* **mvn clean install**
* **mvn clean test**

**It will show the build success message to confirm the testing.**

****

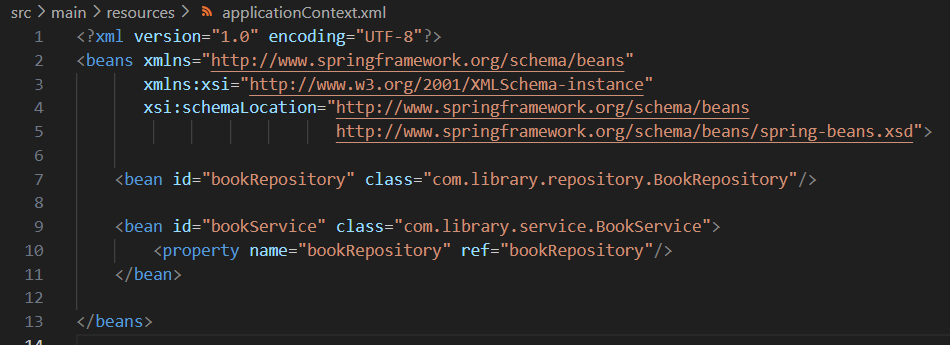
**Exercise 2: Implementing Dependency Injection**

**Scenario:** In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.

**Solution:**

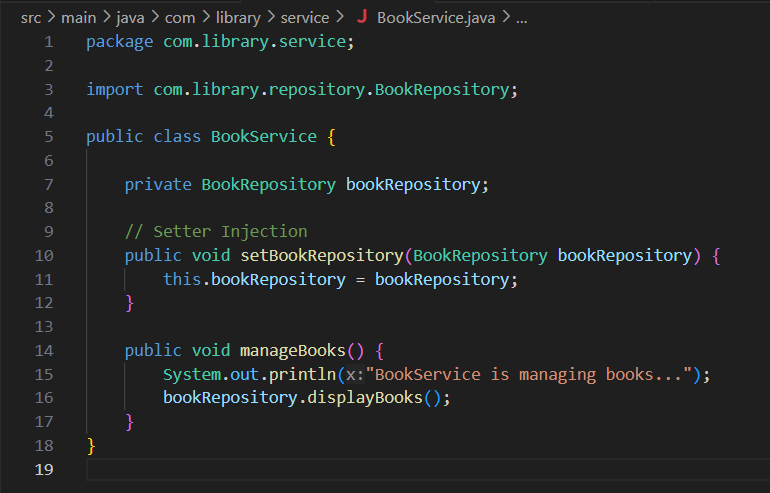
**Step 1: Modify the XML Configuration**

* Update the **applicationContext.xml** if not already done in Exercise 1 to wire the **BookRepository** into **BookService**:



**Step 2:** **Update the BookService Class**

* Ensure **BookService** contains a **setter method** to receive the **BookRepository** bean:



**Step 3:** **Test the Configuration**

* Run the **LibraryManagementApplication** class and ensure it loads the beans and calls the methods as expected.
* Expected output:



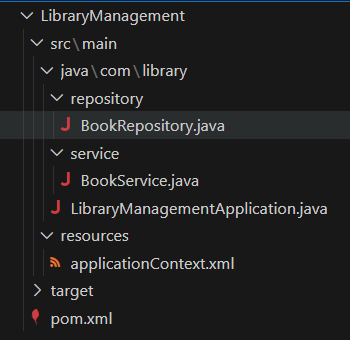
**Exercise 4: Creating and Configuring a Maven Project**

**Scenario:** You need to set up a new Maven project for the library management application and add Spring dependencies.

**Solution:**

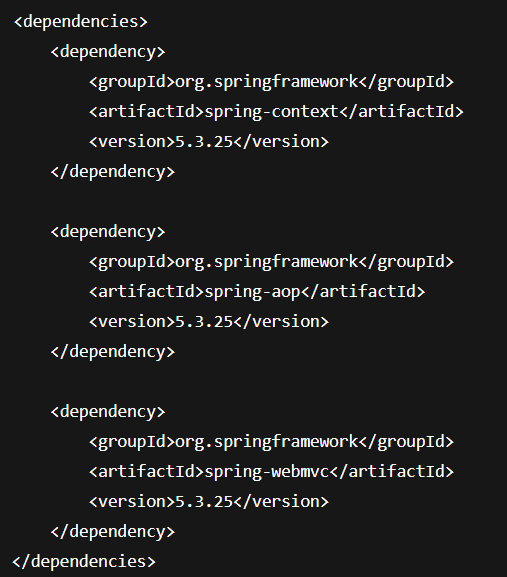
**Step 1:** Create a New Maven Project:

* Create a new Maven project named **LibraryManagement**.



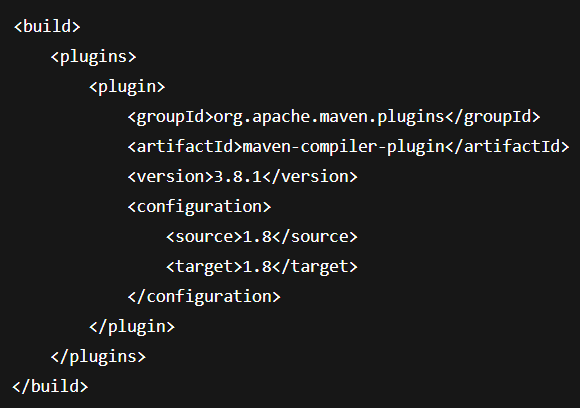
**Step 2:** Add Spring Dependencies in pom.xml:

* Include dependencies for **Spring Context, Spring AOP, and Spring WebMVC**.



**Step 3: Configure Maven Plugins:**

* Configure the Maven Compiler Plugin for Java version 1.8 in the pom.xml file.



**Step 4: Build and verify the Project by running the following command:**

* **mvn clean install**
* **Run java file by right clicking on the editor.**
* **Expected output:**

