

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:

- Create a Maven project named **LibraryManagement**.
- Add Spring Core dependencies in the **pom.xml** file.

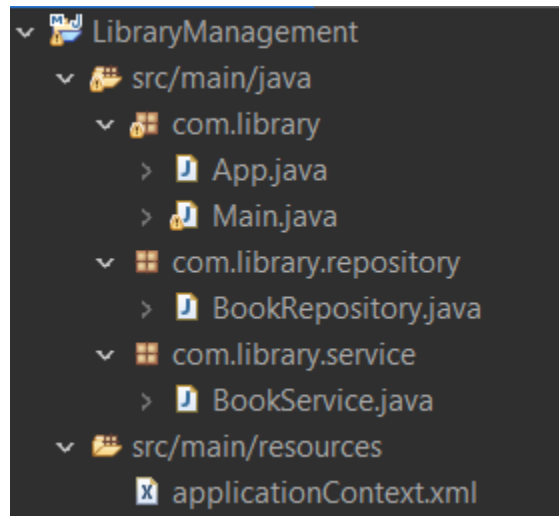
Output:

```
1  http://maven.apache.org/xsd/maven-4.0.0.xsd (xsi:schemaLocation with catalog)
2  1● <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <groupId>com.library</groupId>
6     <artifactId>LibraryManagement</artifactId>
7     <version>0.0.1-SNAPSHOT</version>
8     <packaging>jar</packaging>
9
10    <name>LibraryManagement</name>
11    <url>http://maven.apache.org</url>
12
13    13● <properties>
14        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15    </properties>
16
17    17● <dependencies>
18    18●   <dependency>
19        <groupId>junit</groupId>
20        <artifactId>junit</artifactId>
21        <version>3.8.1</version>
22        <scope>test</scope>
23    </dependency>
24    24●   <dependency>
25        <groupId>org.springframework</groupId>
26        <artifactId>spring-context</artifactId>
27        <version>5.3.34</version>
28    </dependency>
29    </dependencies>
30 </project>
```

2. Configure the Application Context:

- Create an XML configuration file named **applicationContext.xml** in the **src/main/resources** directory.

Output:



- Define beans for **BookService** and **BookRepository** in the XML file.

Output:

```
http://www.springframework.org/schema/beans/spring-beans.xsd (xsi:schemaLocation)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xsi:schemaLocation="http://www.springframework.org/schema/beans
5                           http://www.springframework.org/schema/beans/spring-beans.xsd">
6
7     <!-- Repository Bean -->
8     <bean id="bookRepository" class="com.library.repository.BookRepository" />
9
10    <!-- Service Bean -->
11    <bean id="bookService" class="com.library.service.BookService">
12        <property name="bookRepository" ref="bookRepository"/>
13    </bean>
14
15 </beans>
```

3. Define Service and Repository Classes:

- Create a package **com.library.service** and add a class **BookService**.

Output:

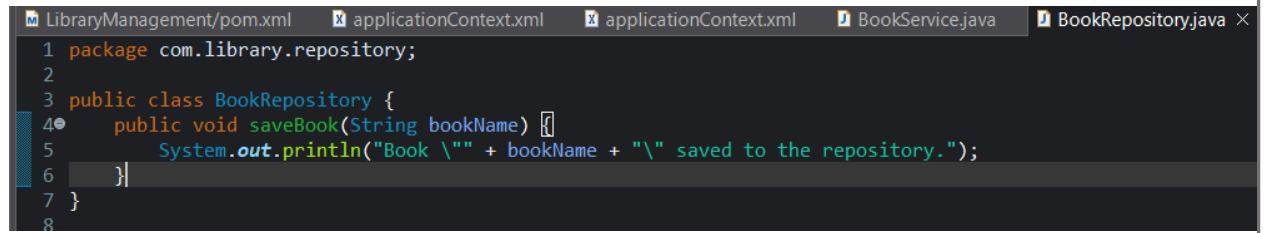
```

1 package com.library.service;
2 import com.library.repository.BookRepository;
3
4 public class BookService {
5     private BookRepository bookRepository;
6
7     // Setter for Spring injection
8     public void setBookRepository(BookRepository bookRepository) {
9         this.bookRepository = bookRepository;
10    }
11
12    public void addBook(String bookName) {
13        bookRepository.saveBook(bookName);
14    }
15 }
16

```

- Create a package **com.library.repository** and add a class **BookRepository**.

Output:



```

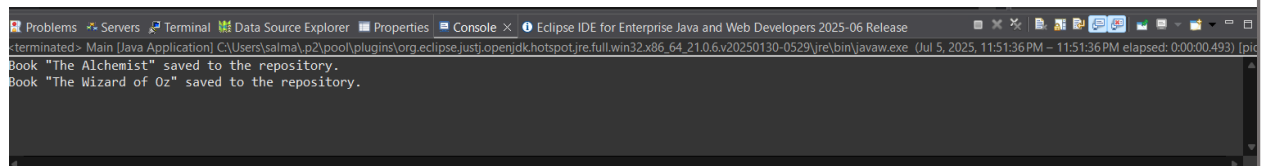
LibraryManagement/pom.xml  applicationContext.xml  applicationContext.xml  BookService.java  BookRepository.java X
1 package com.library.repository;
2
3 public class BookRepository {
4     public void saveBook(String bookName) {
5         System.out.println("Book \"" + bookName + "\" saved to the repository.");
6     }
7 }
8

```

4. Run the Application:

- Create a main class to load the Spring context and test the configuration.

Output:



```

Eclipse IDE for Enterprise Java and Web Developers 2025-06 Release
terminated> Main [Java Application] C:\Users\salma\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.6.v20250130-0529\jre\bin\javaw.exe (Jul 5, 2025, 11:51:36 PM - 11:51:36 PM elapsed: 0:00:00.493) [pic
Book "The Alchemist" saved to the repository.
Book "The Wizard of Oz" saved to the repository.

```