**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.
2. **Configure the Application Context:**
   * Create an XML configuration file named **applicationContext.xml** in the **src/main/resources** directory.
   * Define beans for **BookService** and **BookRepository** in the XML file.
3. **Define Service and Repository Classes:**
   * Create a package **com.library.service** and add a class **BookService**.
   * Create a package **com.library.repository** and add a class **BookRepository**.
4. **Run the Application:**
   * Create a main class to load the Spring context and test the configuration.

**Solution:**

**Set Up a Spring Project**:

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

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.20</version>

</dependency>

<!-- Add other dependencies if needed -->

</dependencies>

**Configure the Application Context**:

<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">

<!-- Define BookRepository bean -->

<bean id="bookRepository" class="com.library.repository.BookRepository">

<!-- You can add properties or constructor arguments here if needed -->

</bean>

<!-- Define BookService bean -->

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

<!-- You can add properties or constructor arguments here if needed -->

</bean>

</beans>

**Define Service and Repository Classes**:

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

package com.library.service;

public class BookService {

// Service methods

}

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

package com.library.repository;

public class BookRepository {

// Repository methods

}

// BookService.java

package com.library.service;

public class BookService {

private final BookRepository bookRepository;

// Constructor injection

public BookService(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

// Service methods

}

// BookRepository.java

package com.library.repository;

public class BookRepository {

// Repository methods

}

**Run the Application**:

// LibraryManagementApplication.java

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

BookService bookService = (BookService) context.getBean("bookService");

// Test the configuration

if (bookService != null) {

System.out.println("BookService bean successfully loaded.");

} else {

System.out.println("Failed to load BookService bean.");

}

}

}