Step 1: Create Spring Configuration File

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

<?xml version="1.0" encoding="UTF-8"?>

<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.BookRepositoryImpl"/>

<!-- Define BookService bean and inject BookRepository -->

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

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

In the above configuration file, we have defined two beans: bookRepository and bookService. The bookService bean has a dependency on bookRepository, which is injected using the property element.

Step 2: Update the BookService Class

Ensure that the BookService class has a setter method for BookRepository:

package com.library.service;

import com.library.repository.BookRepository;

public class BookServiceImpl implements BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

@Override

public void addBook(String title, String author) {

// implementation

}

}

In the above code, we have added a setter method setBookRepository to the BookServiceImpl class, which allows Spring to inject the BookRepository instance.

Step 3: Run the Application

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

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class LibraryManagementApplication {

public static void main(String[] args) {

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

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

bookService.addBook("Java Programming", "John Doe");

}

}

In the above code, we have created a LibraryManagementApplication class with a main method that loads the Spring context using the ClassPathXmlApplicationContext class. We then retrieve the BookService bean from the context and call the addBook method to test the configuration.