# **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.

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

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

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

<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">

<modelVersion>4.0.0</modelVersion>

<groupId>com.Library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0.0</version>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>21</maven.compiler.source>

<maven.compiler.target>21</maven.compiler.target>

<spring.version>6.1.0</spring.version> <!-- Spring 6.x for Java 21 -->

</properties>

<dependencies>

<!-- Spring Context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

</dependencies>

</project>

**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.

*applicationContext.xml:*

<?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

https://www.springframework.org/schema/beans/spring-beans.xsd">

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

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

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

</bean>

</beans>

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

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

*BookService.java:*

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public String getBookDetails(String isbn) {

return bookRepository.getBookInfo(isbn);

}

}

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

*BookRepository.java:*

package com.library.repository;

public class BookRepository {

public String getBookInfo(String isbn) {

return "Book details for ISBN: %s (Java 21)".formatted(isbn);

}

}

**4. Run the Application:**

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

*LibraryApplication.java:*

package com.library.main;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class LibraryApplication {

public static void main(String[] args) {

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

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

String bookInfo = bookService.getBookDetails("123-456-789");

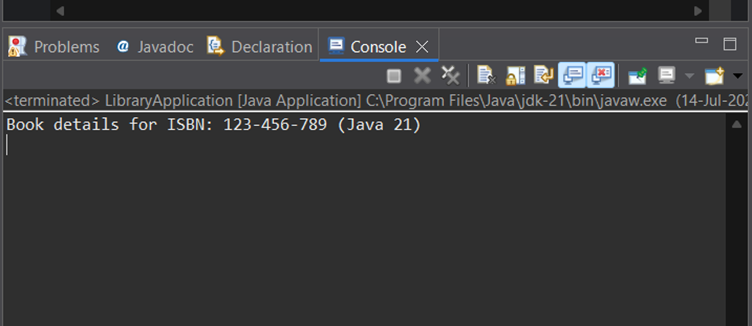
System.out.println(bookInfo);

((ClassPathXmlApplicationContext) context).close();

}

}

**OUTPUT:**

****