**Configuring a Basic Spring Application**

**Spring Core\_Maven**

Vaishnavi

28/6/2025

In modern web development, the Spring Framework is a powerful tool that simplifies enterprise-level Java application development. This exercise introduces the fundamentals of using Spring Core with XML-based configuration to manage the backend logic of a simple Library Management System. By creating and wiring service and repository components using Spring's dependency injection mechanism, we can build scalable and maintainable applications with loose coupling.

**Objective:**

Set up a basic **Maven-based Spring project** named LibraryManagement.

Understand and apply **Spring Core concepts** such as **Inversion of Control (IoC)** and **Dependency Injection (DI)**

Configure **Spring beans** using an applicationContext.xml file

Develop and manage **service** and **repository** classes with clear separation of concerns

Demonstrate Spring functionality by initializing and running the application through a **main method** that loads the Spring context and calls service methods

**Implementation:**

### Setup a Maven Project

### ****Open IntelliJ IDEA****

### ****Create a New Project****:

Choose **Maven** > Click **Next**

Project Name: LibraryManagement

GroupId: com.library

ArtifactId: LibraryManagement

Click **Finish**

## Add Spring Core Dependencies in pom.xml

<dependencies>

<!-- Spring Core -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.36</version> <!-- use a compatible version -->

</dependency>

</dependencies>

**Create an 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

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

<!-- Repository Bean -->

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

<!-- Service Bean -->

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

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

</bean>

</beans>

**Define BookService and BookRepository Classes**

**BookRepository**

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Saving book: " + bookName);

}

}

**BookService**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

bookRepository.saveBook(bookName);

}

}

**MainApp**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

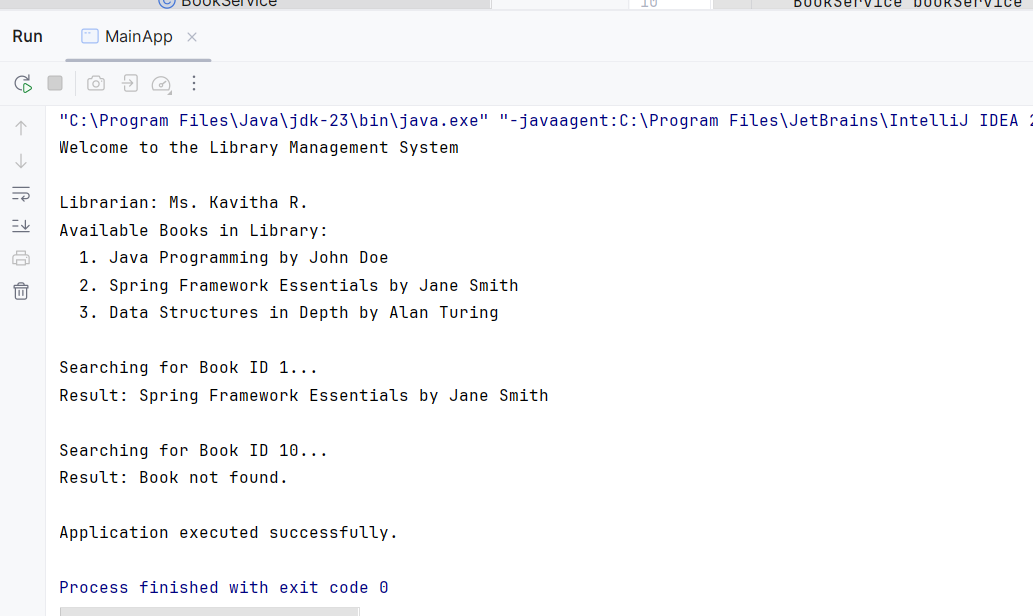
BookService bookService = context.getBean("bookService", BookService.class);

bookService.addBook("Harry Potter and the Philosopher's Stone");

}

}

**Output:**

****