**Exercise 6: Configuring Beans with Annotations**

**Step 1: Enable Component Scanning**

Update applicationContext.xml to include component scanning for the com.library package:

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

<beans xmlns="(link unavailable)"

xmlns:xsi="(link unavailable)"

xmlns:context="(link unavailable)"

xsi:schemaLocation="(link unavailable)

(link unavailable)

(link unavailable)">

<context:component-scan base-package="com.library"/>

</beans>

**Step 2: Annotate Classes**

Use @Service annotation for the BookService class:

package com.library.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service

public class BookService {

private BookRepository bookRepository;

@Autowired

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

// Service methods

}

Use @Repository annotation for the BookRepository class:

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

// Repository methods

}

**Step 3: Test the Configuration**

Run the LibraryManagementApplication main class to verify the annotation-based 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 = context.getBean(BookService.class);

System.out.println("Book Service: " + bookService);

System.out.println("Book Repository: " + bookService.getBookRepository());

}

}

**Output:**

Book Service: com.library.service.BookService@12345678

Book Repository: com.library.repository.BookRepository@90123456