**Exercise 6: Configuring Beans with Annotations**

**Scenario:**

You need to simplify the configuration of beans in the library management application using annotations.

**LibraryAppMain.java :-**

package org.example1;  
  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class LibraryAppMain {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 // The @PostConstruct in BookController will execute automatically.  
 }  
}

**BookController.java :-**

package org.example1.controller;  
  
import org.example1.service.BookService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
  
import javax.annotation.PostConstruct;  
import java.util.List;  
  
@Controller  
public class BookController {  
  
 @Autowired  
 private BookService bookService;  
  
 @PostConstruct  
 public void init() {  
 System.*out*.println("Initial book list: " + bookService.getAllBooks());  
 bookService.addBook("DSA");  
 System.*out*.println("After adding DSA: " + bookService.getAllBooks());  
 }  
}

**BookRepository.java :-**

package org.example1.repository;  
  
import org.springframework.stereotype.Repository;  
import java.util.\*;  
  
@Repository  
public class BookRepository {  
 private final List<String> books = new ArrayList<>(List.*of*("Java", "Spring", "Hibernate"));  
  
 public List<String> findAll() {  
 return books;  
 }  
  
 public void addBook(String book) {  
 books.add(book);  
 }  
}

**BookService.java :-**

package org.example1.service;  
  
import org.example1.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import java.util.List;  
  
@Service  
public class BookService {  
  
 @Autowired  
 private BookRepository bookRepository;  
  
 public List<String> getAllBooks() {  
 return bookRepository.findAll();  
 }  
  
 public void addBook(String book) {  
 bookRepository.addBook(book);  
 }  
}

**applicationContext.xml :-**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:context="http://www.springframework.org/schema/context"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 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  
 http://www.springframework.org/schema/context  
 https://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/aop  
 https://www.springframework.org/schema/aop/spring-aop.xsd">  
  
 <!-- Enable component scanning -->  
 <context:component-scan base-package="org.example1"/>  
  
 <!-- Enable annotation-driven AOP -->  
 <aop:aspectj-autoproxy/>  
  
</beans>

**Pom.xml :-**

<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  
 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>org.example1</groupId>  
 <artifactId>LibraryManagementSystem</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <packaging>war</packaging>  
  
 <properties>  
 <java.version>1.8</java.version>  
 <spring.version>5.3.30</spring.version>  
 <aspectj.version>1.9.21</aspectj.version>  
 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  
 </properties>  
  
 <dependencies>  
  
 <!-- Spring Context -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <!-- Spring Web MVC -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <!-- Spring AOP -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>${spring.version}</version>  
 </dependency>  
  
 <!-- AspectJ Runtime -->  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjrt</artifactId>  
 <version>${aspectj.version}</version>  
 </dependency>  
  
 <!-- AspectJ Weaver -->  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>${aspectj.version}</version>  
 </dependency>  
  
 <!-- Servlet API -->  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>4.0.1</version>  
 <scope>provided</scope>  
 </dependency>  
  
 <!-- JSTL (optional if using JSP) -->  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>jstl</artifactId>  
 <version>1.2</version>  
 </dependency>  
 <dependency>  
 <groupId>jakarta.annotation</groupId>  
 <artifactId>jakarta.annotation-api</artifactId>  
 <version>1.3.5</version> <!-- or 2.1.1+ for Jakarta EE 9+ -->  
 </dependency>  
  
 <!-- Logging -->  
 <dependency>  
 <groupId>commons-logging</groupId>  
 <artifactId>commons-logging</artifactId>  
 <version>1.2</version>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <finalName>LibraryManagementSystem</finalName>  
 <plugins>  
  
 <!-- Compiler plugin -->  
 <plugin>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.8.1</version>  
 <configuration>  
 <source>${java.version}</source>  
 <target>${java.version}</target>  
 </configuration>  
 </plugin>  
  
 <!-- WAR packaging plugin -->  
 <plugin>  
 <artifactId>maven-war-plugin</artifactId>  
 <version>3.3.2</version>  
 <configuration>  
 <failOnMissingWebXml>false</failOnMissingWebXml>  
 </configuration>  
 </plugin>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <configuration>  
 <source>9</source>  
 <target>9</target>  
 </configuration>  
 </plugin>  
  
 </plugins>  
 </build>  
  
</project>

Output :-

