**Exercise 4: Creating and Configuring a Maven Project**

**Scenario:**

You need to set up a new Maven project for the library management application and add Spring dependencies.

**App.java :-**

package org.example1;  
  
import org.example1.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;  
  
public class App {  
 public static void main(String[] args) {  
 ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");  
 BookService service = (BookService) context.getBean("bookService");  
 service.listBooks().forEach(book -> System.*out*.println(book.getTitle()));  
 }  
}

**LoggingAspect.java :-**

package org.example1.aop;  
  
import org.aspectj.lang.annotation.Aspect;  
import org.aspectj.lang.annotation.Before;  
import org.springframework.stereotype.Component;  
  
@Aspect  
@Component  
public class LoggingAspect {  
  
 @Before("execution(\* org.example1.service.BookService.\*(..))")  
 public void logBefore() {  
 System.*out*.println("[AOP] BookService method is about to be called...");  
 }  
}

**LibraryController.java :-**

package org.example1.controller;  
  
import org.example1.model.Book;  
import org.example1.service.BookService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.web.bind.annotation.RequestMapping;  
  
import java.util.List;  
  
@Controller  
public class LibraryController {  
  
 @Autowired  
 private BookService bookService;  
  
 @RequestMapping("/books")  
 public String showBooks(Model model) {  
 List<Book> books = bookService.listBooks();  
 model.addAttribute("books", books);  
 return "book";  
 }  
}

**Book.java :-**

package org.example1.model;  
  
public class Book {  
 private int id;  
 private String title;  
  
 public Book() {}  
 public Book(int id, String title) {  
 this.id = id;  
 this.title = title;  
 }  
  
 public int getId() { return id; }  
 public void setId(int id) { this.id = id; }  
  
 public String getTitle() { return title; }  
 public void setTitle(String title) { this.title = title; }  
}

**BookRepository.java :-**

package org.example1.repository;  
  
import org.example1.model.Book;  
import java.util.Arrays;  
import java.util.List;  
  
public class BookRepository {  
 public List<Book> getAllBooks() {  
 return Arrays.*asList*(  
 new Book(1, "Java Basics"),  
 new Book(2, "Spring in Action")  
 );  
 }  
}

BookService :-

package org.example1.service;  
  
import org.example1.model.Book;  
import org.example1.repository.BookRepository;  
import java.util.List;  
  
public class BookService {  
  
 private BookRepository bookRepo;  
  
 // Constructor-based DI  
 public BookService(BookRepository bookRepo) {  
 this.bookRepo = bookRepo;  
 }  
  
 // Setter-based DI (optional)  
 public void setBookRepo(BookRepository bookRepo) {  
 this.bookRepo = bookRepo;  
 }  
  
 public List<Book> listBooks() {  
 return bookRepo.getAllBooks();  
 }  
}

**applicationContext.xml :-**

<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
 xmlns:aop="http://www.springframework.org/schema/aop"  
 xmlns:context="http://www.springframework.org/schema/context"  
 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  
 http://www.springframework.org/schema/context  
 http://www.springframework.org/schema/context/spring-context.xsd  
 http://www.springframework.org/schema/aop  
 http://www.springframework.org/schema/aop/spring-aop.xsd">  
  
 <!-- Component Scanning -->  
 <context:component-scan base-package="org.example1"/>  
  
 <!-- WebMVC Annotation Support -->  
 <context:annotation-config/>  
  
 <!-- Bean Definitions -->  
 <bean id="bookRepo" class="org.example1.repository.BookRepository"/>  
  
 <bean id="bookService" class="org.example1.service.BookService">  
 <constructor-arg ref="bookRepo"/>  
 </bean>  
  
 <!-- AOP -->  
 <bean id="loggingAspect" class="org.example1.aop.LoggingAspect"/>  
 <aop:config>  
 <aop:aspect ref="loggingAspect">  
 <aop:pointcut id="allServiceMethods" expression="execution(\* org.example1.service.\*.\*(..))"/>  
 <aop:before pointcut-ref="allServiceMethods" method="logBefore"/>  
 </aop:aspect>  
 </aop:config>  
  
</beans>

**Book.jsp :-**

<%@ page contentType="text/html;charset=UTF-8" %>  
<html>  
<head><title>Books</title></head>  
<body>  
<h2>Books List:</h2>  
<ul>  
 <c:forEach

items="${books}" var="book">  
 <li>${book.id} - ${book.title}</li>  
 </c:forEach>  
</ul>  
</body>  
</html>

**web.xml :-**

<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
 version="3.1">  
  
 <display-name>LibraryManagement</display-name>  
  
 <servlet>  
 <servlet-name>spring</servlet-name>  
 <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>  
 <init-param>  
 <param-name>contextConfigLocation</param-name>  
 <param-value>/WEB-INF/applicationContext.xml</param-value>  
 </init-param>  
 <load-on-startup>1</load-on-startup>  
 </servlet>  
  
 <servlet-mapping>  
 <servlet-name>spring</servlet-name>  
 <url-pattern>/</url-pattern>  
 </servlet-mapping>  
  
</web-app>

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>LibraryManagement</artifactId>  
 <version>1.0</version>  
 <packaging>war</packaging>  
  
 <properties>  
 <maven.compiler.source>1.8</maven.compiler.source>  
 <maven.compiler.target>1.8</maven.compiler.target>  
 </properties>  
  
 <dependencies>  
 <!-- Spring Core, Context, WebMVC, AOP -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>5.3.33</version>  
 </dependency>  
 <dependency>  
 <groupId>javax.servlet</groupId>  
 <artifactId>javax.servlet-api</artifactId>  
 <version>4.0.1</version>  
 <scope>provided</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.21</version>  
 </dependency>  
  
 </dependencies>  
  
 <build>  
 <finalName>LibraryManagement</finalName>  
 <plugins>  
 <!-- Compiler Plugin -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.11.0</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

**Output :-**

