Exercise 3: Implementing Logging with Spring AOP

**Scenario:**

The library management application requires logging capabilities to track method execution times.

**1. Add Spring AOP Dependency:**

o Update *pom.xml* to include Spring AOP dependency.

<?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>

<!-- Spring AOP -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

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

</dependency>

<!-- AspectJ Weaver (required for @Aspect) -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.20</version>

</dependency>

</dependencies>

</project>

**2. Create an Aspect for Logging:**

o Create a package com.library.aspect and add a class LoggingAspect with a method to log execution times.

*LoggingAspect.java:*

package com.library.aspect;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.springframework.stereotype.Component;

*@Aspect*

*@Component*

public class LoggingAspect {

*@Around*("execution(\* com.library.service.\*.\*(..))")

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long startTime = System.*currentTimeMillis*();

// Proceed with the original method execution

Object result = joinPoint.proceed();

long executionTime = System.*currentTimeMillis*() - startTime;

System.***out***.println("[AOP LOG] " + joinPoint.getSignature() +

" executed in " + executionTime + "ms");

return result;

}

}

**3. Enable AspectJ Support:**

o Update *applicationContext.xml* to enable AspectJ support and register the aspect.

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="http://www.springframework.org/schema/beans

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

http://www.springframework.org/schema/aop

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

<!-- Enable AspectJ auto-proxying -->

<aop:aspectj-autoproxy/>

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

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

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

</bean>

<!-- Register Aspect as a bean -->

<bean class="com.library.aspect.LoggingAspect"/>

</beans>

**4. Test the Aspect:**

o Run the *LibraryManagementApplication* main class and observe the console for log messages indicating method execution times.

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("978-3-16-148410-0");

System.***out***.println(bookInfo);

}

}

**OUTPUT:**

