Step 1: Define an Aspect

Create a package com.library.aspect and add a class LoggingAspect:

package com.library.aspect;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

@Aspect

public class LoggingAspect {

private static final Logger LOGGER = LoggerFactory.getLogger(LoggingAspect.class);

@Before("execution(\* \*(..))")

public void logBeforeMethodExecution(JoinPoint joinPoint) {

LOGGER.info("Before executing method: {}", joinPoint.getSignature());

}

@After("execution(\* \*(..))")

public void logAfterMethodExecution(JoinPoint joinPoint) {

LOGGER.info("After executing method: {}", joinPoint.getSignature());

}

}

In the above code, we have defined an aspect class LoggingAspect with two advice methods: logBeforeMethodExecution and logAfterMethodExecution. These methods will log messages before and after the execution of any method in the application.

Step 2: Create Advice Methods

The advice methods are already defined in the LoggingAspect class.

Step 3: Configure the Aspect

Update applicationContext.xml to register the aspect and enable AspectJ auto-proxying:

<?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

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

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

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

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

<aop:aspectj-autoproxy/>

<!-- Define the aspect -->

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

</beans>

<?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

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

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

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

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

<aop:aspectj-autoproxy/>

<!-- Define the aspect -->

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

</beans>

In the above configuration file, we have enabled AspectJ auto-proxying using the <aop:aspectj-autoproxy/> element and defined the loggingAspect bean.

Step 4: Test the Aspect

Run the LibraryManagementApplication main class to verify the AOP functionality:

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

bookService.addBook("Java Programming", "John Doe");

}

}