**Exercise 3: Implementing Logging with Spring AOP**

**applicationContext.xml**

<?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"

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

       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

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

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

    <!-- Bean Definitions -->

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

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

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

    </bean>

    <!-- Aspect Definition -->

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

    <!-- Enable AOP -->

    <aop:aspectj-autoproxy />

    <!-- AOP Configuration -->

    <aop:config>

        <aop:aspect ref="loggingAspect">

            <aop:around method="logExecutionTime"

                        pointcut="execution(\* com.library.service.\*.\*(..))"/>

        </aop:aspect>

    </aop:config>

</beans>

**LibraryManagementApplication.java**

package com.library;

import com.library.service.BookService;

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.printBookTitle(); // Should trigger logging aspect

}

}

**LoginAspect.java**

//it has to be created int a separate folder aspect in java/com/library/aspect

package com.library.aspect;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.\*;

import org.aspectj.lang.reflect.MethodSignature;

public class LoggingAspect {

// Around advice to log execution time

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long start = System.currentTimeMillis();

// Proceed with the method

Object result = joinPoint.proceed();

long end = System.currentTimeMillis();

MethodSignature signature = (MethodSignature) joinPoint.getSignature();

String methodName = signature.getDeclaringTypeName() + "." + signature.getName();

System.out.println("Execution time of " + methodName + ": " + (end - start) + " ms");

return result;

}

}

**BookRepository.java**

package com.library.repository;

public class BookRepository {

    public String getBookTitle() {

        return "Effective Java";

    }

}

**BookService.java**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    // ✅ Setter method required for XML-based DI

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void printBookTitle() {

        System.out.println("Book: " + bookRepository.getBookTitle());

    }

}

**AppTest.java**

package com;

import static org.junit.Assert.assertTrue;

import org.junit.Test;

/\*\*

 \* Unit test for simple App.

 \*/

public class AppTest

{

    /\*\*

     \* Rigorous Test :-)

     \*/

    @Test

    public void shouldAnswerWithTrue()

    {

        assertTrue( true );

    }

}

**Pom.xml**

<?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 http://maven.apache.org/xsd/maven-4.0.0.xsd">

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.library</groupId>

  <artifactId>library-management-spring-aop</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>library-management-spring-aop</name>

  <!-- FIXME change it to the project's website -->

  <url>http://www.example.com</url>

   <properties>

    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

    <maven.compiler.source>8</maven.compiler.source>

    <maven.compiler.target>8</maven.compiler.target>

  </properties>

  <dependencies>

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>4.11</version>

      <scope>test</scope>

    </dependency>

     <dependency>

            <groupId>org.springframework</groupId>

            <artifactId>spring-context</artifactId>

            <version>5.3.30</version> <!-- or latest compatible -->

        </dependency>

        <!-- Spring AOP -->

<dependency>

    <groupId>org.springframework</groupId>

    <artifactId>spring-aop</artifactId>

    <version>5.3.30</version>

</dependency>

 <dependency>

     <groupId>org.aspectj</groupId>

     <artifactId>aspectjrt</artifactId>

     <version>1.9.19</version>

</dependency>

<!-- AspectJ -->

<dependency>

    <groupId>org.aspectj</groupId>

    <artifactId>aspectjweaver</artifactId>

    <version>1.9.21</version>

</dependency>

  </dependencies>

  <build>

    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

      <plugins>

        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

        <plugin>

          <artifactId>maven-clean-plugin</artifactId>

          <version>3.1.0</version>

        </plugin>

        <plugin>

            <groupId>org.codehaus.mojo</groupId>

            <artifactId>exec-maven-plugin</artifactId>

            <version>3.1.0</version>

            <configuration>

                <mainClass>com.library.LibraryManagementApplication</mainClass>

            </configuration>

        </plugin>

        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

        <plugin>

          <artifactId>maven-resources-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-compiler-plugin</artifactId>

          <version>3.8.0</version>

        </plugin>

        <plugin>

          <artifactId>maven-surefire-plugin</artifactId>

          <version>2.22.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-jar-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-install-plugin</artifactId>

          <version>2.5.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-deploy-plugin</artifactId>

          <version>2.8.2</version>

        </plugin>

        <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

        <plugin>

          <artifactId>maven-site-plugin</artifactId>

          <version>3.7.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-project-info-reports-plugin</artifactId>

          <version>3.0.0</version>

        </plugin>

      </plugins>

    </pluginManagement>

  </build>

</project>