**Exercise – 8**

**Main.java**

package com.lib;

import com.lib.model.Book;

import com.lib.service.bookServe;

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");

    bookServe obj = (bookServe) context.getBean("bookServe ");

        obj.getBooks();

        obj.getBookById(123);

        obj.addBook(new Book(130,"Wings of Fire","Abdul Kalam"));

        obj.updateBook(new Book(130,"Wings of Fire","A.P.J. Abdul Kalam"));

        obj.deleteBook(130);

    }

}

**bookServe.java**

package com.lib.service;

import com.lib.model.Book;

import com.lib.repository.bookRepo;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class bookServe {

    private bookRepo bookRepository;

    @Autowired

    public bookServe (bookRepo bookRepository) {

        this.bookRepository = bookRepository;

    }

    public List<Book> getBooks() {

        return bookRepository.getBooks();

    }

    public Book getBookById(int id){

        return bookRepository.getBookById(id);

    }

    public void addBook(Book book){

        bookRepository.addBook(book);

    }

    public void updateBook(Book book){

        bookRepository.updateBook(book);

    }

    public void deleteBook(int id){

        bookRepository.deleteBook(id);

    }

}

**bookRepo.java**

package com.lib.repository;

import com.lib.model.Book;

import org.springframework.stereotype.Repository;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;

@Repository

public class bookRepo {

    List<Book> books = new ArrayList<>(Arrays.asList(new Book(123, "Adventures of Tom Sawyer", "Mark Twain"),

            new Book(124, "Alice in Wonderland", "Lewis Carrol"),

            new Book(126, "Gulliver’s Travels", "Jonathan Swift"),

            new Book(125, "A passage to India", "E.M.Forster"),

            new Book(128, "Discovery of India", "Pandit Jawaharlal Nehru")));

    public List<Book> getBooks() {

        return books;

    }

    public Book getBookById(int id) {

        for (Book book : books) {

            if (book.getBookId() == id) {

                return book;

            }

        }

        return null;

    }

    public void addBook(Book book) {

        books.add(book);

    }

    public void updateBook(Book book) {

        int index = 0;

        for (int i = 0; i < books.size(); i++) {

            if (books.get(i).getBookId() == book.getBookId()) {

                index = i;

                break;

            }

        }

        books.set(index, book);

    }

    public void deleteBook(int id) {

        int index = 0;

        for (int i = 0; i < books.size(); i++) {

            if (books.get(i).getBookId() == id) {

                index = i;

                break;

            }

        }

        books.remove(index);

    }

}

**Book.java**

package com.lib.model;

public class Book {

    private int bookId;

    private String bookName;

    private String author;

    public Book(int bookId, String bookName, String author) {

        this.bookId = bookId;

        this.bookName = bookName;

        this.author = author;

    }

    public int getBookId() {

        return bookId;

    }

    public void setBookId(int bookId) {

        this.bookId = bookId;

    }

    public String getBookName() {

        return bookName;

    }

    public void setBookName(String bookName) {

        this.bookName = bookName;

    }

    public String getAuthor() {

        return author;

    }

    public void setAuthor(String author) {

        this.author = author;

    }

}

**LoggingAspect.java**

package com.lib.aspect;

import org.aspectj.lang.JoinPoint;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.aspectj.lang.annotation.Before;

import org.springframework.stereotype.Component;

@Aspect

@Component

public class LoggingAspect {

    @Before("execution(\* com.library.repository.BookRepository.\*(..))")

    public void beforeLogging(JoinPoint jp)  {

        System.out.println("Method "+jp.getSignature().getName()+" is invoked");

    }

    @After("execution(\* com.library.repository.BookRepository.\*(..))")

    public void afterLogging(JoinPoint jp){

        System.out.println("Method "+jp.getSignature().getName()+" is executed");

    }

}

**AppTest.java**

package com.lib;

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 );

    }

}

**applicationContext.xml**

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

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

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

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/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">

<!-- Enabling component scanning -->

<context:component-scan base-package=" com.lib" />

<aop:aspectj-autoproxy/>

</beans>

**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>LibraryManagement-8</artifactId>

<version>1.0-SNAPSHOT</version>

<name>LibraryManagement-8</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>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<dependencies>

<!-- Spring AOP -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.29</version>

</dependency>

<!-- AspectJ Weaver -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.22.1</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.29</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</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>

<!-- 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>