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

**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>com.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.34</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>5.3.34</version>

</dependency>

</dependencies>

</project>

**applicationContext.xml**

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

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

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/aop

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

<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>

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

<aop:config>

<aop:aspect ref="loggingAspect">

<aop:around method="logExecutionTime"

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

</aop:aspect>

</aop:config>

</beans>

**LoggingAspect.java**

package com.library.aspect;

import org.aspectj.lang.ProceedingJoinPoint;

public class LoggingAspect {

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long start = System.currentTimeMillis();

Object result = joinPoint.proceed();

long end = System.currentTimeMillis();

System.out.println(joinPoint.getSignature().toShortString() + " executed in " + (end - start) + " ms");

return result;

}

}

**BookRepository.java**

package com.library.repository;

import java.util.ArrayList;

import java.util.List;

public class BookRepository {

private final List<String> books = new ArrayList<>();

public void saveBook(String title) {

books.add(title);

System.out.println("Book saved: " + title);

}

public boolean updateBook(String oldTitle, String newTitle) {

int index = books.indexOf(oldTitle);

if (index >= 0) {

books.set(index, newTitle);

System.out.println("Book updated from " + oldTitle + " to " + newTitle);

return true;

}

return false;

}

public boolean deleteBook(String title) {

if (books.remove(title)) {

System.out.println("Book deleted: " + title);

return true;

}

return false;

}

public List<String> getAllBooks() {

return books;

}

}

**BookService.java**

package com.library.service;

import com.library.repository.BookRepository;

import java.util.List;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String title) {

bookRepository.saveBook(title);

}

public void updateBook(String oldTitle, String newTitle) {

if (!bookRepository.updateBook(oldTitle, newTitle)) {

System.out.println("Book not found: " + oldTitle);

}

}

public void deleteBook(String title) {

if (!bookRepository.deleteBook(title)) {

System.out.println("Book not found: " + title);

}

}

public void displayBooks() {

List<String> books = bookRepository.getAllBooks();

if (books.isEmpty()) {

System.out.println("No books in the library.");

} else {

System.out.println("Books in the library:");

for (String book : books) {

System.out.println("- " + book);

}

}

}

}

**LibraryMainApp.java**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.Scanner;

public class LibraryMainApp {

public static void main(String[] args) {

ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

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

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("1. Add Book");

System.out.println("2. Display Books");

System.out.println("3. Update Book");

System.out.println("4. Delete Book");

System.out.println("5. Exit");

System.out.print("Enter choice: ");

choice = scanner.nextInt();

scanner.nextLine();

switch (choice) {

case 1:

System.out.print("Enter book title: ");

String title = scanner.nextLine();

bookService.addBook(title);

break;

case 2:

bookService.displayBooks();

break;

case 3:

System.out.print("Enter old title: ");

String oldTitle = scanner.nextLine();

System.out.print("Enter new title: ");

String newTitle = scanner.nextLine();

bookService.updateBook(oldTitle, newTitle);

break;

case 4:

System.out.print("Enter title to delete: ");

String deleteTitle = scanner.nextLine();

bookService.deleteBook(deleteTitle);

break;

case 5:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice.");

}

} while (choice != 5);

scanner.close();

}

}









