Library Management System (Spring Core Exercises 4 to 8)

| **Exercise** | **Description** |
| --- | --- |
| 4 | Create and configure Maven project with Spring |
| 5 | Configure Spring IoC container using XML |
| 6 | Use annotation-based bean definitions |
| 7 | Implement constructor and setter injection |
| 8 | Implement AOP for logging method execution times |

**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</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>library-management</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>

        <!-- Spring Core and Context -->

        <dependency>

            <groupId>org.springframework</groupId>

            <artifactId>spring-context</artifactId>

            <version>5.3.30</version>

        </dependency>

        <!-- Spring AOP -->

        <dependency>

            <groupId>org.springframework</groupId>

            <artifactId>spring-aop</artifactId>

            <version>5.3.30</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.MainApp</mainClass>

            </configuration>

        </plugin>

        <plugin>

                <groupId>org.apache.maven.plugins</groupId>

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

                <version>3.10.1</version>

                <configuration>

                    <source>1.8</source>

                    <target>1.8</target>

                </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>

**AppTest.java**

package com.library;

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">

    <!-- Enable annotation-based components -->

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

    <!-- Enable AOP -->

    <aop:aspectj-autoproxy/>

    <!-- Register the LoggingAspect bean -->

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

</beans>

**MainApp.java**

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

    public static void main(String[] args) {

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

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

        bookService.displayBook();

    }

}

**BookService.java**

package com.library.service;

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

import org.springframework.stereotype.Service;

import com.library.repository.BookRepository;

@Service

public class BookService {

    private BookRepository bookRepository;

    public BookService() {

        System.out.println("BookService: Default Constructor Called");

    }

    // Constructor Injection

    public BookService(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    // Setter Injection

    @Autowired

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void displayBook() {

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

    }

}

**BookRepository.java**

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

    public String getBookTitle() {

        return "Java FSE";

    }

}

**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 {

        System.out.println("Before method: " + joinPoint.getSignature());

        long start = System.currentTimeMillis();

        Object result = joinPoint.proceed();

        long elapsed = System.currentTimeMillis() - start;

        System.out.println("After method: " + joinPoint.getSignature());

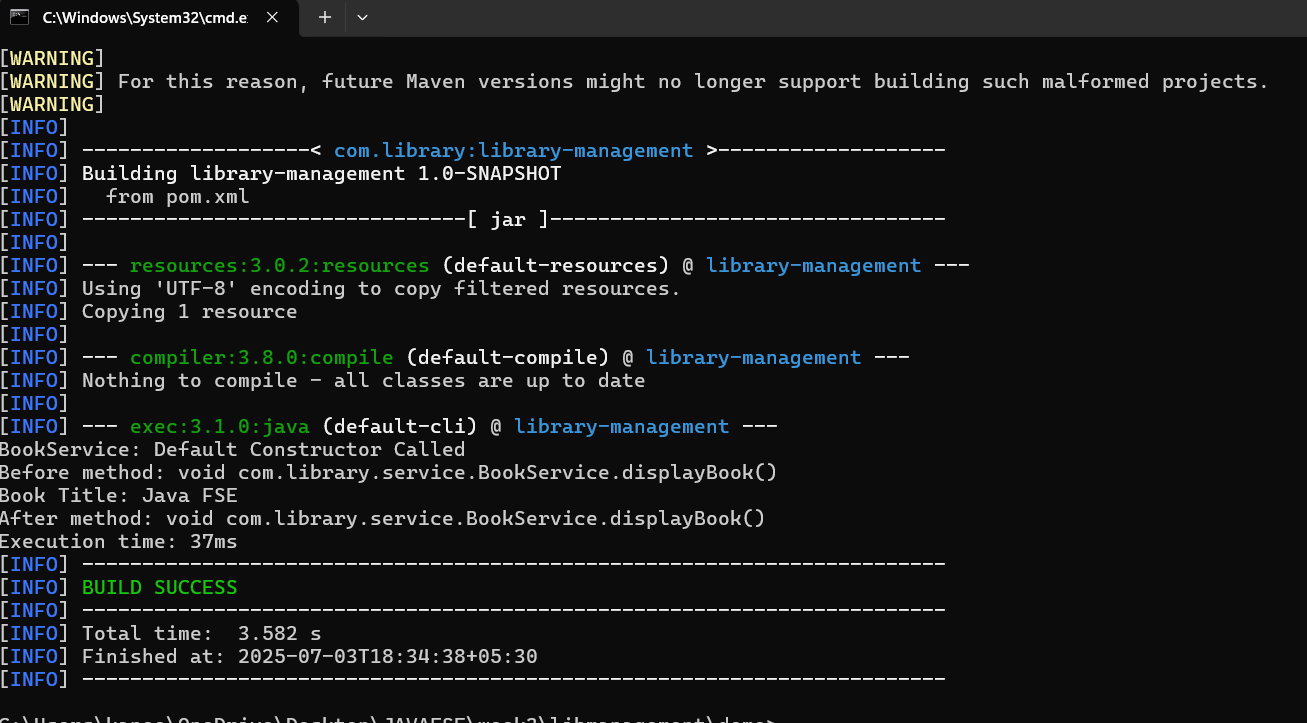
        System.out.println("Execution time: " + elapsed + "ms");

        return result;

    }

}

**Output:**



**Package Structure:**

com.library

├── MainApp.java

├── service/

│ └── BookService.java

├── repository/

│ └── BookRepository.java

└── aspect/

└── LoggingAspect.java