Exercise 1 and 2:

**1. Set Up a Spring Project**

**pom.xml**

<project xmlns="http://maven.apache.org/POM/4.0.0" ...>

<modelVersion>4.0.0</modelVersion>

<groupId>com.library</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<dependencies>

<!-- Spring Core -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.20</version>

</dependency>

</dependencies>

</project>

**2. Configure the Application Context**

**src/main/resources/applicationContext.xml**

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

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">

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

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

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

</bean>

</beans>

**3. Define Service and Repository Classes**

**com/library/repository/BookRepository.java**

package com.library.repository;

public class BookRepository {

public void save() {

System.out.println("Book saved to repository.");

}

}

com/library/service/BookService.java

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook() {

bookRepository.save();

System.out.println("BookService: addBook called.");

}

}

**4. Run the Application**

**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.addBook();

}

}

Outputs:

Book saved to repository.

BookService: addBook called.

Book saved to repository.

BookService: addBook called.

Exercise 3;

1. **Add Spring AOP Dependency**

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.20</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.7</version>

</dependency>

**2. Create Logging Aspect**

**com/library/aspect/LoggingAspect.java**

package com.library.aspect;

import org.aspectj.lang.ProceedingJoinPoint;

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

@Aspect

public class LoggingAspect {

@Around("execution(\* com.library.service.\*.\*(..))")

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {

long start = System.currentTimeMillis();

Object result = joinPoint.proceed();

long duration = System.currentTimeMillis() - start;

System.out.println(joinPoint.getSignature() + " executed in " + duration + "ms");

return result;

}

}

**3. Enable AspectJ Support**

Update applicationContext.xml:

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

xsi:schemaLocation="

...

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

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

<aop:aspectj-autoproxy/>

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

Output:

Book saved to repository.

BookService: addBook called.

void com.library.service.BookService.addBook() executed in 3ms

Exercise 4:

<build>

<plugins>

<plugin>

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

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

<version>3.8.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

</plugins>

</build>

Output:

[INFO] Scanning for projects...

[INFO]

[INFO] ----------------< com.library:LibraryManagement >----------------

[INFO] Building LibraryManagement 1.0-SNAPSHOT

[INFO] ...

[INFO] BUILD SUCCESS

SPRING DATA JPA HANDSON

Exercise 1:

Console output when running OrmLearnApplication.java:

06-07-25 19:30:45.123 main INFO OrmLearnApplication main 27 Inside main

MySQL Schema & Table Creation:

create schema ormlearn;

create table country(co\_code varchar(2) primary key, co\_name varchar(50));

insert into country values ('IN', 'India');

insert into country values ('US', 'United States of America');

Application Logs for Country Fetch:

06-07-25 19:31:02.456 main INFO OrmLearnApplication testGetAllCountries 47 Start

06-07-25 19:31:02.462 main DEBUG com.cognizant.ormlearn.OrmLearnApplication testGetAllCountries 49 countries=[Country{code='IN', name='India'}, Country{code='US', name='United States of America'}]

06-07-25 19:31:02.463 main INFO OrmLearnApplication testGetAllCountries 50 End

Exercise 2:

Difference between JPA, Hibernate and Spring Data JPA

A screenshot of a computer

AI-generated content may be incorrect.