**Exercise 1: Configuring a Basic Spring Application**

AdminService.java

package com.smartlib.service;

import com.smartlib.model.Book;

import java.util.\*;

public class AdminService {

private Map<Integer, Book> books = new HashMap<>();

private int counter = 1;

public void addBook(Book book) {

book.setId(counter++);

books.put(book.getId(), book);

System.out.println("[INFO] Book '" + book.getTitle() + "' added successfully.");

}

public Book getBook(int id) {

return books.get(id);

}

public List<Book> listBooks() {

return new ArrayList<>(books.values());

}

}

Book.java

package com.smartlib.model;

public class Book {

private int id;

private String title;

private String author;

private boolean issued;

public Book() {}

public Book(String title, String author) {

this.title = title;

this.author = author;

this.issued = false;

}

public int getId() { return id; }

public void setId(int id) { this.id = id; }

public String getTitle() { return title; }

public void setTitle(String title) { this.title = title; }

public String getAuthor() { return author; }

public void setAuthor(String author) { this.author = author; }

public boolean isIssued() { return issued; }

public void setIssued(boolean issued) { this.issued = issued; }

}

# IssueService.java

package com.smartlib.service;

import com.smartlib.model.Book;

import com.smartlib.model.Member;

import java.time.LocalDate;

import java.time.temporal.ChronoUnit;

import java.util.\*;

public class IssueService {

private Map<Integer, LocalDate> issueDates = new HashMap<>();

private AdminService adminService;

public void setAdminService(AdminService adminService) {

this.adminService = adminService;

}

public void issueBook(Book book, Member member) {

if (!book.isIssued()) {

book.setIssued(true);

issueDates.put(book.getId(), LocalDate.now().minusDays(10)); // Simulate overdue

System.out.println("[INFO] Book issued to member: " + member.getName());

}

}

public void checkOverdue(Book book) {

LocalDate issueDate = issueDates.get(book.getId());

if (issueDate != null) {

long days = ChronoUnit.DAYS.between(issueDate, LocalDate.now());

if (days > 7) {

System.out.println("[WARNING] Book ID " + book.getId() + " is overdue!");

}

}

}

public void returnBook(Book book, Member member) {

if (book.isIssued()) {

book.setIssued(false);

issueDates.remove(book.getId());

System.out.println("[INFO] Book '" + book.getTitle() + "' returned by member: " + member.getName());

}

}

}

LibraryServiceTest.java

package com.smartlib;

import com.smartlib.model.Book;

import com.smartlib.model.Member;

import com.smartlib.service.AdminService;

import com.smartlib.service.IssueService;

import com.smartlib.service.MemberService;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class LibraryServiceTest {

private AdminService adminService;

private MemberService memberService;

private IssueService issueService;

@Before

public void setup() {

adminService = new AdminService();

memberService = new MemberService();

issueService = new IssueService();

issueService.setAdminService(adminService);

}

@Test

public void testIssueAndReturnBook() {

Book book = new Book("JUnit in Action", "Craig Walls");

adminService.addBook(book);

Member member = new Member(1, "Tester");

memberService.registerMember(member);

issueService.issueBook(book, member);

assertTrue(book.isIssued());

issueService.returnBook(book, member);

assertFalse(book.isIssued());

}

}

# MainApp.java

package com.smartlib.main;

import com.smartlib.model.Book;

import com.smartlib.model.Member;

import com.smartlib.service.AdminService;

import com.smartlib.service.MemberService;

import com.smartlib.service.IssueService;

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");

AdminService admin = context.getBean("adminService", AdminService.class);

MemberService memberService = context.getBean("memberService", MemberService.class);

IssueService issueService = context.getBean("issueService", IssueService.class);

Book book = new Book("Spring for Professionals", "John Doe");

admin.addBook(book);

Member member = new Member(101, "Ravi");

memberService.registerMember(member);

issueService.issueBook(book, member);

issueService.checkOverdue(book);

issueService.returnBook(book, member);

((ClassPathXmlApplicationContext) context).close();

}

}

Member.java

package com.smartlib.model;

public class Member {

private int id;

private String name;

public Member() {}

public Member(int id, String name) {

this.id = id;

this.name = name;

}

public int getId() { return id; }

public void setId(int id) { this.id = id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

}

MemberService.java

package com.smartlib.service;

import com.smartlib.model.Member;

import java.util.\*;

public class MemberService {

private Map<Integer, Member> members = new HashMap<>();

public void registerMember(Member member) {

members.put(member.getId(), member);

System.out.println("[INFO] Member '" + member.getName() + "' registered.");

}

public Member getMember(int id) {

return members.get(id);

}

}

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"

xsi:schemaLocation="http://www.springframework.org/schema/beans

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

<bean id="adminService" class="com.smartlib.service.AdminService"/>

<bean id="memberService" class="com.smartlib.service.MemberService"/>

<bean id="issueService" class="com.smartlib.service.IssueService">

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

</bean>

</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>org.example</groupId>

<artifactId>SmartLibraryManager</artifactId>

<version>1.0-SNAPSHOT</version>

<properties>

<maven.compiler.source>22</maven.compiler.source>

<maven.compiler.target>22</maven.compiler.target>

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

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.34</version>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

**OUTPUT :**

