**Spring Core and Maven**

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

Pox.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 https://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>

<name>LibraryManagement</name>

<properties>

<java.version>11</java.version>

<spring.version>5.3.20</spring.version>

</properties>

<dependencies>

<!-- Spring Core -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<!-- For testing -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>${spring.version}</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.8.1</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

</configuration>

</plugin>

</plugins>

</build>

</project>

src/main/resources/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

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

<!-- BookRepository bean -->

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

<!-- BookService bean with dependency injection -->

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

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

</bean>

</beans>

BookRepository (src/main/java/com/library/repository/BookRepository.java)

package com.library.repository;

import java.util.HashMap;

import java.util.Map;

public class BookRepository {

private final Map<String, String> books = new HashMap<>();

public BookRepository() {

// Initialize with some sample data

books.put("ISBN-001", "Spring in Action");

books.put("ISBN-002", "Clean Code");

books.put("ISBN-003", "Effective Java");

}

public String findBookByIsbn(String isbn) {

return books.get(isbn);

}

public void addBook(String isbn, String title) {

books.put(isbn, title);

}

public Map<String, String> getAllBooks() {

return new HashMap<>(books);

}

}

BookService (src/main/java/com/library/service/BookService.java)

package com.library.service;

import com.library.repository.BookRepository;

import java.util.Map;

public class BookService {

private BookRepository bookRepository;

// Setter for dependency injection

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public String getBookTitle(String isbn) {

return bookRepository.findBookByIsbn(isbn);

}

public void addBook(String isbn, String title) {

bookRepository.addBook(isbn, title);

}

public Map<String, String> getAllBooks() {

return bookRepository.getAllBooks();

}

}

Main Application Class (src/main/java/com/library/LibraryApplication.java)

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import java.util.Map;

public class LibraryApplication {

public static void main(String[] args) {

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

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

System.out.println("Book with ISBN-001: " + bookService.getBookTitle("ISBN-001"));

bookService.addBook("ISBN-004", "Design Patterns");

System.out.println("Added new book with ISBN-004");

System.out.println("\nAll Books in Library:");

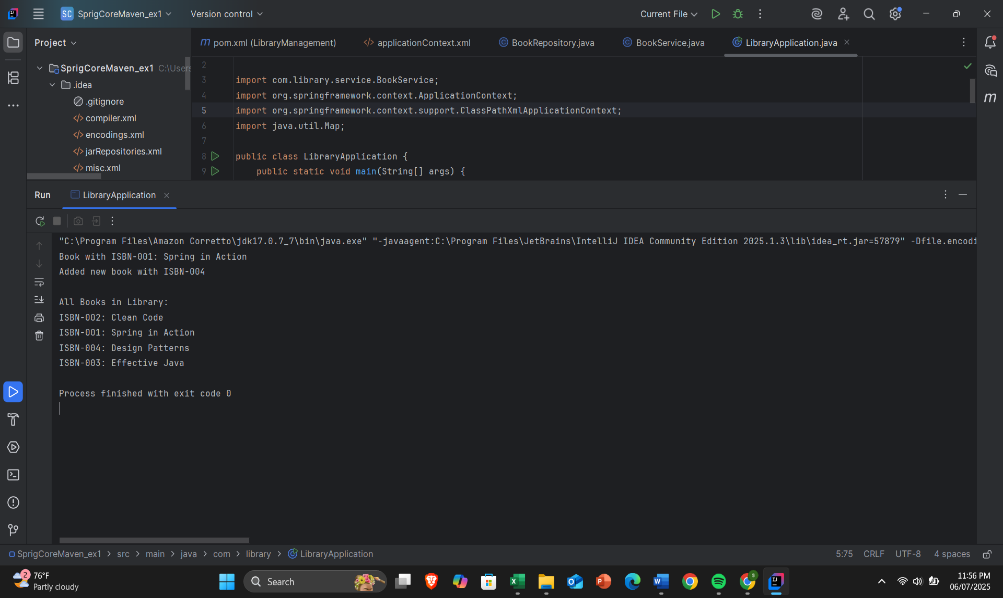
Map<String, String> allBooks = bookService.getAllBooks();

allBooks.forEach((isbn, title) -> System.out.println(isbn + ": " + title));

}

}

OUTPUT:



Exercise 2: Implementing Dependency Injection

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

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

\

</bean>

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

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

</bean>

</beans>

BookService.java

package com.library.service;

import com.library.repository.BookRepository;

import java.util.Map;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public String getBookTitle(String isbn) {

if (bookRepository == null) {

throw new IllegalStateException("BookRepository dependency not injected");

}

return bookRepository.findBookByIsbn(isbn);

}

public void addBook(String isbn, String title) {

bookRepository.addBook(isbn, title);

}

public Map<String, String> getAllBooks() {

return bookRepository.getAllBooks();

}

}

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) {

// Load the Spring application context

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

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

System.out.println("=== Testing Dependency Injection ===");

String isbn = "ISBN-001";

String title = bookService.getBookTitle(isbn);

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

String newIsbn = "ISBN-004";

String newTitle = "Spring Dependency Injection";

bookService.addBook(newIsbn, newTitle);

System.out.println("Added book: " + newTitle);

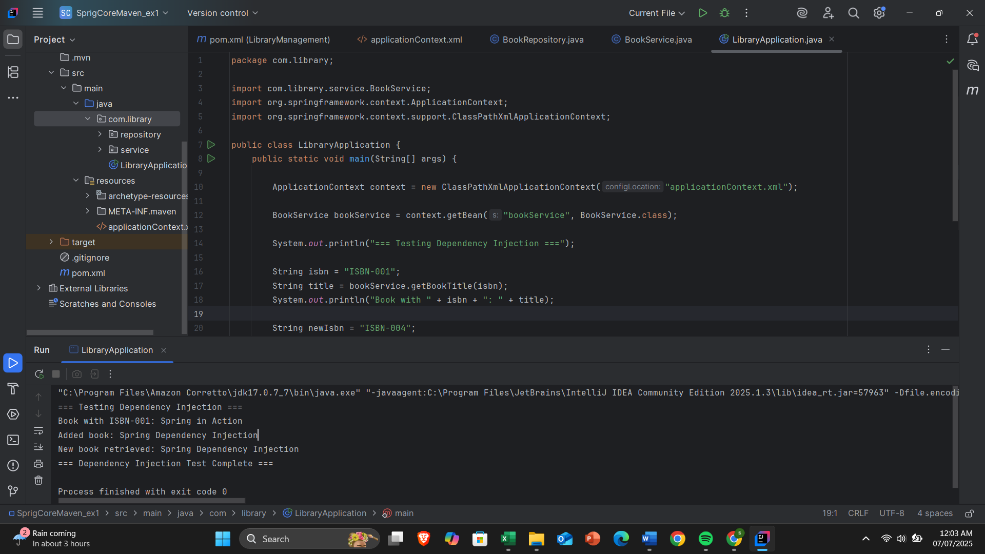
System.out.println("New book retrieved: " + bookService.getBookTitle(newIsbn));

System.out.println("=== Dependency Injection Test Complete ===");

}

}

OUTPUT:



**Exercise 4: Creating and Configuring a Maven Project**

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.example</groupId>

<artifactId>LibraryManagement</artifactId>

<version>1.0-SNAPSHOT</version>

<packaging>jar</packaging>

<name>LibraryManagement</name>

<properties>

<java.version>1.8</java.version>

<spring.version>5.3.33</spring.version>

</properties>

<!-- Spring Context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

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

<version>3.8.1</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

</configuration>

</plugin>

</plugins>

</build>

</project>