**Week 3**

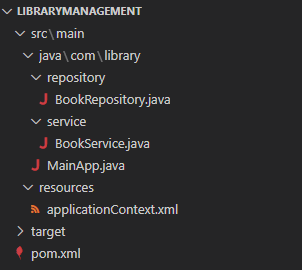
**Spring Core and Maven**

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

**Scenario:**

**Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.**

**FILE STRUCTURE**

****

**CODE**

*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>5.3.36</version>

        </dependency>

    </dependencies>

</project>

*src/main/java/com/library/Main.java*

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class MainApp {

    public static void main(String[] args) {

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

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

        bookService.addBook("The Adventures of Tom Sawyer");

        ((ClassPathXmlApplicationContext) context).close();

    }

}

*src/main/java/com/library/service/BookService.java*

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    // Setter for Dependency Injection

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void addBook(String bookName) {

        System.out.println("Adding book: " + bookName);

        bookRepository.saveBook(bookName);

    }

}

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

package com.library.repository;

public class BookRepository {

    public void saveBook(String bookName) {

        System.out.println("Book '" + bookName + "' added successfully.");

    }

}

*src/main/java/com/library/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

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

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

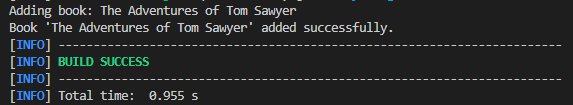
    <bean id="bookService" class="com.library.service.BookService">

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

    </bean>

</beans>

**OUTPUT**

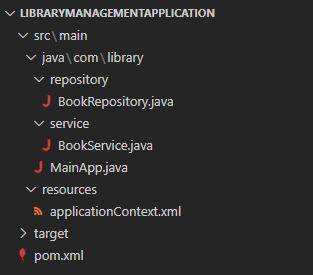
****

**Exercise 2: Implementing Dependency Injection**

**Scenario:**

**In the library management application, you need to manage the dependencies between the BookService and BookRepository classes using Spring's IoC and DI.**

**FILE STRUCTURE**

****

**CODE**

*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>5.3.36</version>

        </dependency>

    </dependencies>

</project>

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

package com.library.repository;

public class BookRepository {

    public void saveBook(String bookName) {

        System.out.println("Book '" + bookName + "' added successfully.");

    }

}

*src/main/java/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) {

        System.out.println("Dependency  Injection: BookRepository has been injected into BookService.");

        this.bookRepository = bookRepository;

    }

    public void addBook(String bookName) {

        System.out.println("Adding book: " + bookName);

        bookRepository.saveBook(bookName);

    }

}

*src/main/java/com/library/MainApp.java*

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class MainApp {

    public static void main(String[] args) {

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

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

        bookService.addBook("Mark Twain");

        ((ClassPathXmlApplicationContext) context).close();

    }

}

*src/main/java/com/library/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

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

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

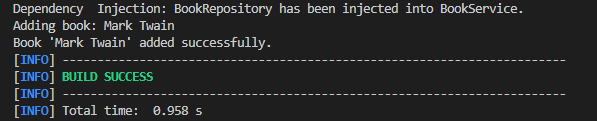
    <bean id="bookService" class="com.library.service.BookService">

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

    </bean>

</beans>

**OUTPUT**

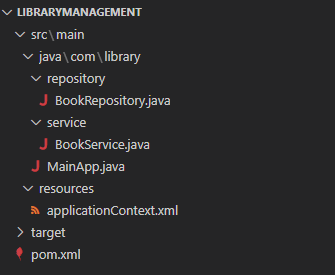
****

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

**Scenario:**

**You need to set up a new Maven project for the library management application and add Spring dependencies.**

**FILE STRUCTURE**

****

**CODE**

*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>

    <!-- Spring Core Context -->

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-context</artifactId>

        <version>5.3.34</version>

    </dependency>

    <!-- Spring AOP -->

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-aop</artifactId>

        <version>5.3.34</version>

    </dependency>

    <!-- Spring Web MVC -->

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-webmvc</artifactId>

        <version>5.3.34</version>

    </dependency>

    <!-- Servlet API (required for WebMVC) -->

    <dependency>

        <groupId>javax.servlet</groupId>

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

        <version>4.0.1</version>

        <scope>provided</scope>

    </dependency>

</dependencies>

<build>

    <plugins>

        <!-- Maven Compiler Plugin -->

        <plugin>

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

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

            <version>3.11.0</version>

            <configuration>

                <source>1.8</source>

                <target>1.8</target>

            </configuration>

        </plugin>

    </plugins>

</build>

</project>

*src/main/java/com/library/BookRepository.java*

package com.library.repository;

public class BookRepository {

    public void saveBook(String bookName) {

        System.out.println("Book '" + bookName + "' added successfully.");

    }

}

*src/main/java/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(String bookName) {

        System.out.println("Adding book: " + bookName);

        bookRepository.saveBook(bookName);

    }

}

*src/main/java/com/library/MainApp.java*

package com.library;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

import com.library.service.BookService;

public class MainApp {

    public static void main(String[] args) {

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

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

        bookService.addBook("Mark Twain");

        ((ClassPathXmlApplicationContext) context).close();

    }

}

*src/main/java/com/library/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

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

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

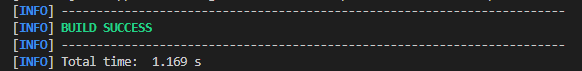
    <bean id="bookService" class="com.library.service.BookService">

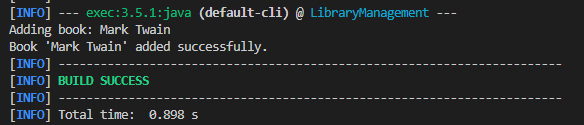
        <property name="bookRepository" ref="bookRepository" />

    </bean>

</beans>

**OUTPUT**

****

****