**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.

**Code:**

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

    }

}

**BookService.java:**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository = new BookRepository();

    public void displayBook() {

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

    } }

**BookRepository.java:**

package com.library.repository;

public class BookRepository {

    public String getBookTitle() {

        return "Spring in Action";

    }

}

**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">

    <!-- Define BookRepository Bean -->

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

    <!-- Define BookService Bean -->

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

</beans>

**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>

  <packaging>jar</packaging>

  <name>LibraryManagement</name>

  <url>http://maven.apache.org</url>

  <dependencies>

    <!-- Spring Context -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-context</artifactId>

      <version>5.3.29</version>

    </dependency>

    <!-- JUnit for testing -->

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>3.8.1</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <plugin>

        <groupId>org.codehaus.mojo</groupId>

        <artifactId>exec-maven-plugin</artifactId>

        <version>3.1.0</version>

        <configuration>

          <mainClass>com.library.LibraryManagementApplication</mainClass>

        </configuration>

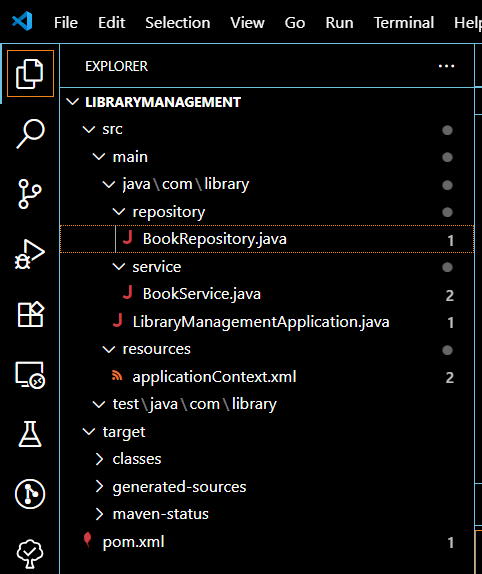
      </plugin>

    </plugins>

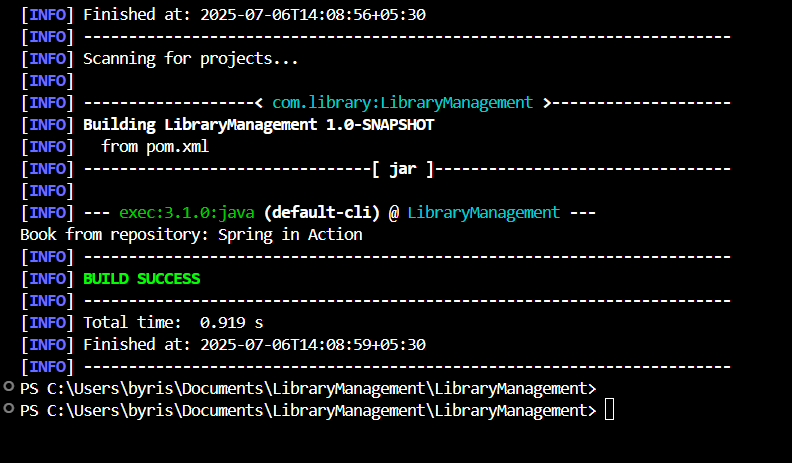
  </build>

</project>

**Structure:**

****

**Output:**

****

The ouput:

Book from repository: Spring in Action

Can be seen in the above ss.

**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.

**Code:**

**BookRepository.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 displayBook() {

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

    }

}

**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">

    <!-- Repository Bean -->

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

    <!-- Service Bean with dependency injected -->

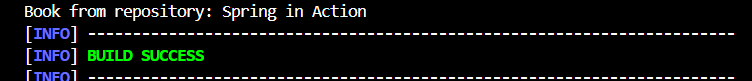
    <bean id="bookService" class="com.library.service.BookService">

        <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.

**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>

  <packaging>jar</packaging>

  <name>LibraryManagement</name>

  <dependencies>

    <!-- Spring Context Dependency -->

    <dependency>

      <groupId>org.springframework</groupId>

      <artifactId>spring-context</artifactId>

      <version>5.3.29</version>

    </dependency>

    <!-- JUnit for testing (optional) -->

    <dependency>

      <groupId>junit</groupId>

      <artifactId>junit</artifactId>

      <version>3.8.1</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <!-- Compiler Plugin for Java 8 -->

      <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>

      <!-- Exec Plugin to run Java main class -->

      <plugin>

        <groupId>org.codehaus.mojo</groupId>

        <artifactId>exec-maven-plugin</artifactId>

        <version>3.1.0</version>

        <configuration>

          <mainClass>com.library.LibraryManagementApplication</mainClass>

        </configuration>

      </plugin>

    </plugins>

  </build>

</project>

**Output:**

