**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.

**BookRepositoryEx2.java**

package com.library.ex2;

public class BookRepositoryEx2 {

    public String getBookTitle() {

        return "Spring Framework Advanced Concepts";

    }

}

**BookServiceEx2.java**

package com.library.ex2;

public class BookServiceEx2 {

    private BookRepositoryEx2 bookRepository;

    // Setter for DI

    public void setBookRepository(BookRepositoryEx2 bookRepository) {

        this.bookRepository = bookRepository;

    }

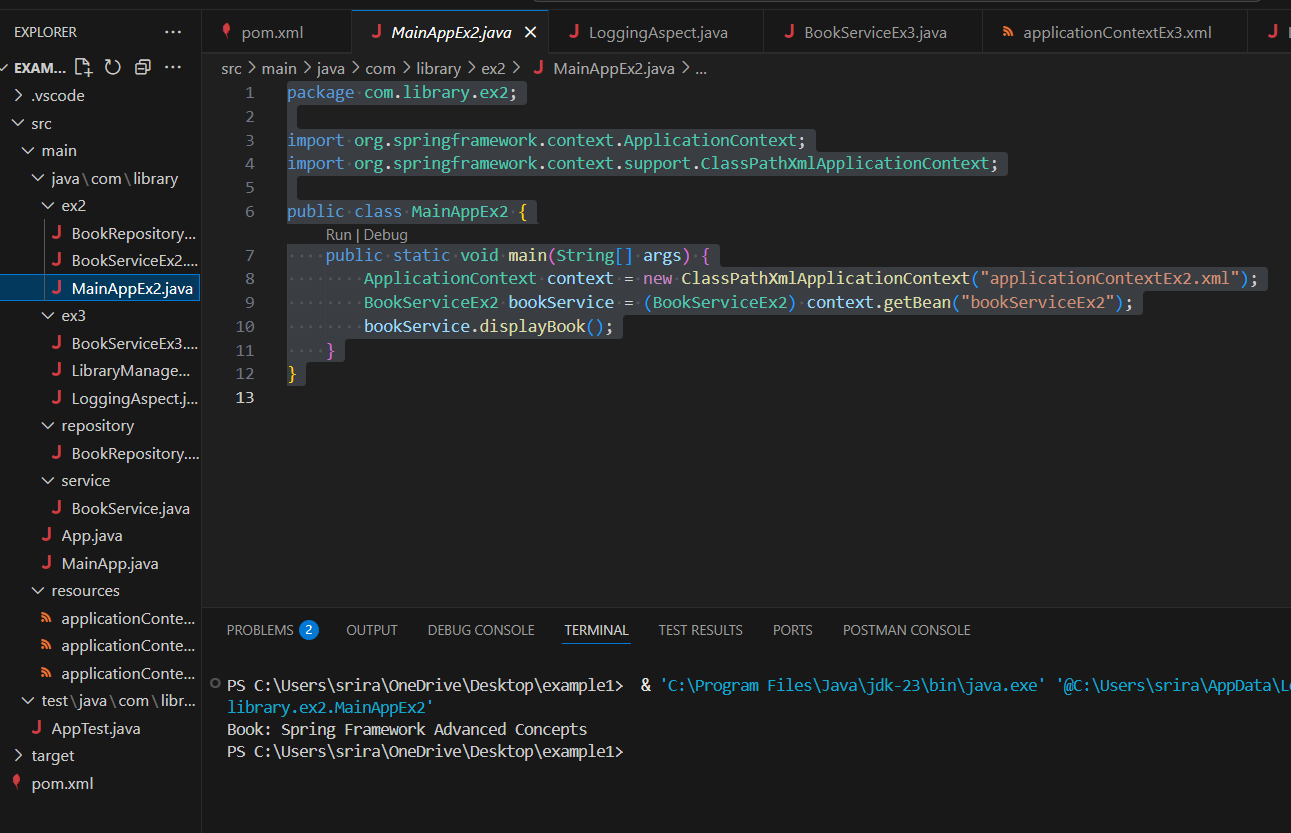
    public void displayBook() {

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

    }

}

**Output:**

****

**MainAppEx2.java**

package com.library.ex2;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainAppEx2 {

    public static void main(String[] args) {

        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContextEx2.xml");

        BookServiceEx2 bookService = (BookServiceEx2) context.getBean("bookServiceEx2");

        bookService.displayBook();

    }

}

**applicationContextEx2.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 BookRepositoryEx2 bean -->

    <bean id="bookRepositoryEx2" class="com.library.ex2.BookRepositoryEx2" />

    <!-- Define BookServiceEx2 bean and inject BookRepositoryEx2 -->

    <bean id="bookServiceEx2" class="com.library.ex2.BookServiceEx2">

        <property name="bookRepository" ref="bookRepositoryEx2" />

    </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>com.spring</groupId>

  <artifactId>example1</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>example1</name>

  <!-- FIXME change it to the project's website -->

  <url>http://www.example.com</url>

  <properties>

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

    <maven.compiler.source>17</maven.compiler.source>

    <maven.compiler.target>17</maven.compiler.target>

  </properties>

  <dependencies>

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-context</artifactId>

        <version>5.3.33</version>

    </dependency>

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-beans</artifactId>

        <version>5.3.33</version>

    </dependency>

    <dependency>

        <groupId>junit</groupId>

        <artifactId>junit</artifactId>

        <version>4.11</version>

        <scope>test</scope>

    </dependency>

    <!-- Spring AOP -->

    <dependency>

        <groupId>org.springframework</groupId>

        <artifactId>spring-aspects</artifactId>

        <version>5.3.30</version>

    </dependency>

    <!-- AspectJ Runtime -->

    <dependency>

        <groupId>org.aspectj</groupId>

        <artifactId>aspectjweaver</artifactId>

        <version>1.9.20</version>

    </dependency>

</dependencies>

  <build>

    <pluginManagement><!-- lock down plugins versions to avoid using Maven defaults (may be moved to parent pom) -->

      <plugins>

        <!-- clean lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#clean\_Lifecycle -->

        <plugin>

          <artifactId>maven-clean-plugin</artifactId>

          <version>3.1.0</version>

        </plugin>

        <!-- default lifecycle, jar packaging: see https://maven.apache.org/ref/current/maven-core/default-bindings.html#Plugin\_bindings\_for\_jar\_packaging -->

        <plugin>

          <artifactId>maven-resources-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

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

          <version>3.8.0</version>

        </plugin>

        <plugin>

          <artifactId>maven-surefire-plugin</artifactId>

          <version>2.22.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-jar-plugin</artifactId>

          <version>3.0.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-install-plugin</artifactId>

          <version>2.5.2</version>

        </plugin>

        <plugin>

          <artifactId>maven-deploy-plugin</artifactId>

          <version>2.8.2</version>

        </plugin>

        <!-- site lifecycle, see https://maven.apache.org/ref/current/maven-core/lifecycles.html#site\_Lifecycle -->

        <plugin>

          <artifactId>maven-site-plugin</artifactId>

          <version>3.7.1</version>

        </plugin>

        <plugin>

          <artifactId>maven-project-info-reports-plugin</artifactId>

          <version>3.0.0</version>

        </plugin>

      </plugins>

    </pluginManagement>

  </build>

</project>