WEEK – 3

Exercise 2: Implementing Dependency Injection

Nipuna,A

Superset id: 6432842

**Scenario:**

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

**Steps:**

1. **Modify the XML Configuration:**
   * Update **applicationContext.xml** to wire **BookRepository** into **BookService**.
2. **Update the BookService Class:**
   * Ensure that **BookService** class has a setter method for **BookRepository**.
3. **Test the Configuration:**
   * Run the **LibraryManagementApplication** main class to verify the dependency injection.

Solution:

1)BookRepository.java

package com.library.repository;

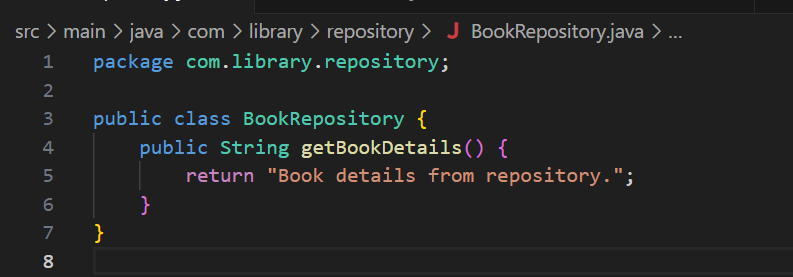
public class BookRepository {

    public String getBookDetails() {

        return "Book details from repository.";

    }

}



2) 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;

    }

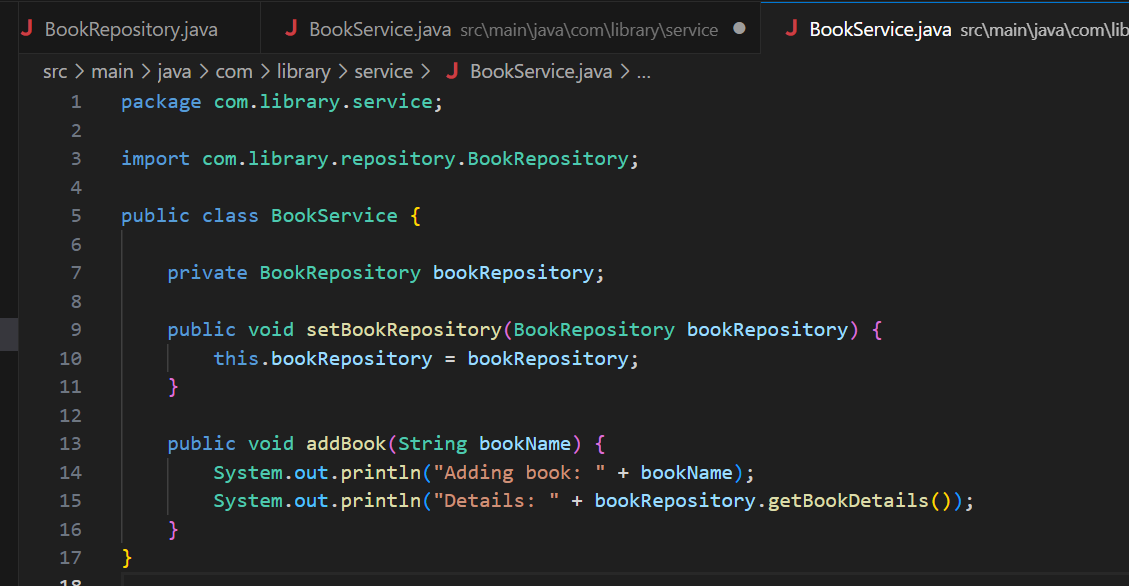
    public void addBook(String bookName) {

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

        System.out.println("Details: " + bookRepository.getBookDetails());

    }

}



3)App.java

package com.library;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App {

    public static void main(String[] args) {

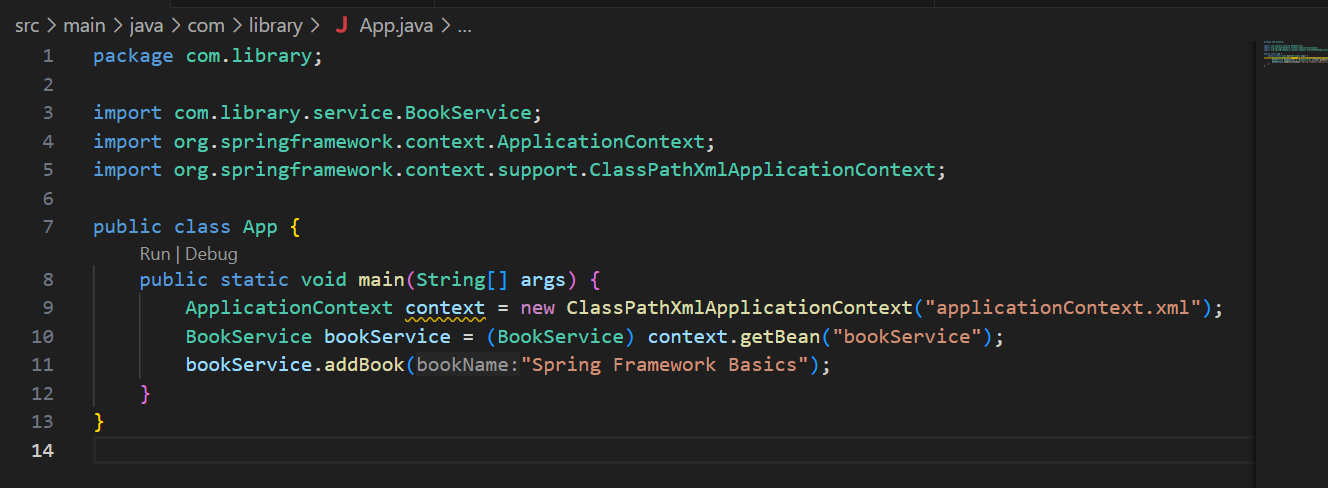
        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

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

        bookService.addBook("Spring Framework Basics");

    }

}



4)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 the repository bean -->

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

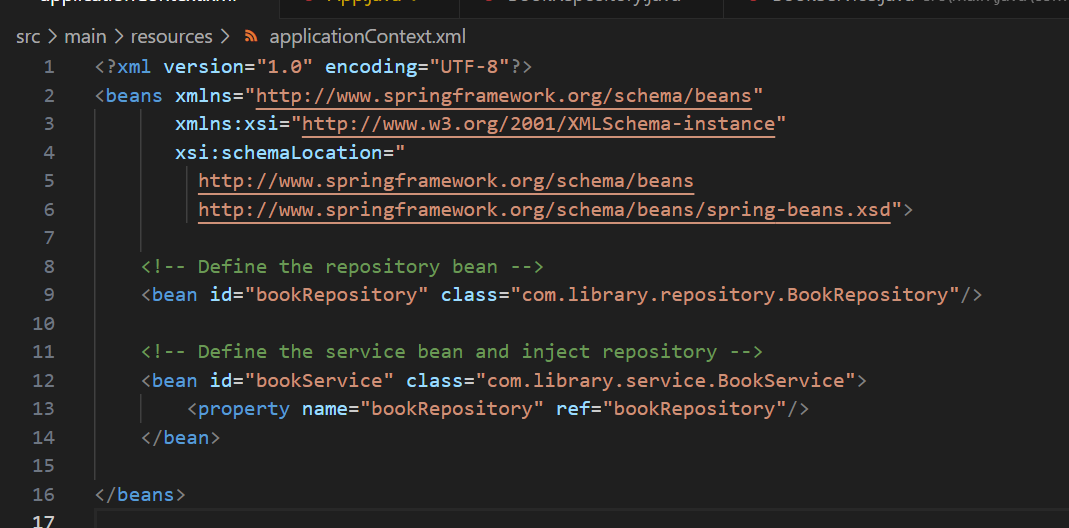
    <!-- Define the service bean and inject repository -->

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

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

    </bean>

</beans>



5)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 Context Dependency -->

        <dependency>

            <groupId>org.springframework</groupId>

            <artifactId>spring-context</artifactId>

            <version>5.3.34</version>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <!-- Plugin to allow running Java main classes -->

            <plugin>

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

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

                <version>3.1.0</version>

                <executions>

                    <execution>

                        <goals>

                            <goal>java</goal>

                        </goals>

                    </execution>

                </executions>

                <configuration>

                    <mainClass>com.library.App</mainClass>

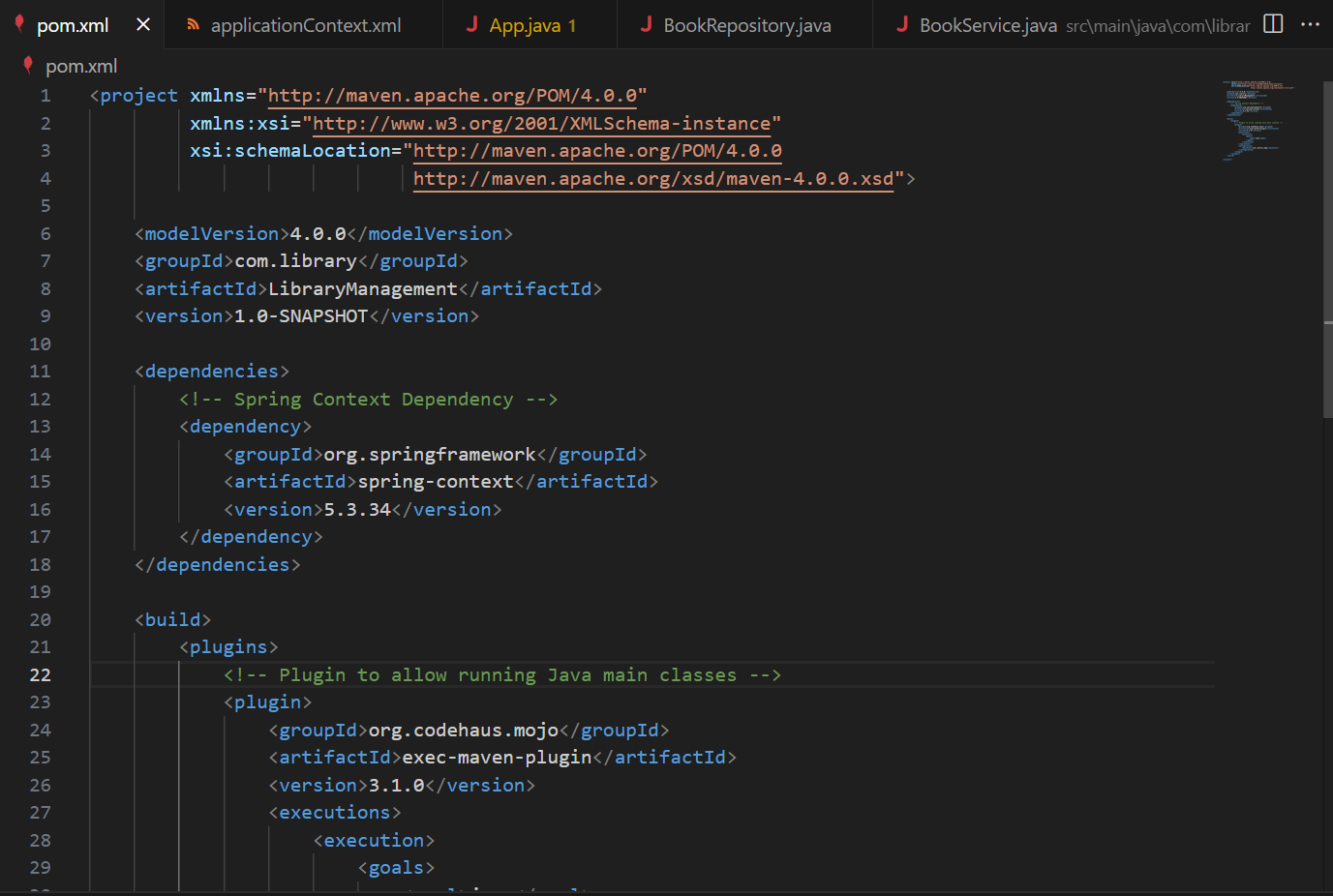
                </configuration>

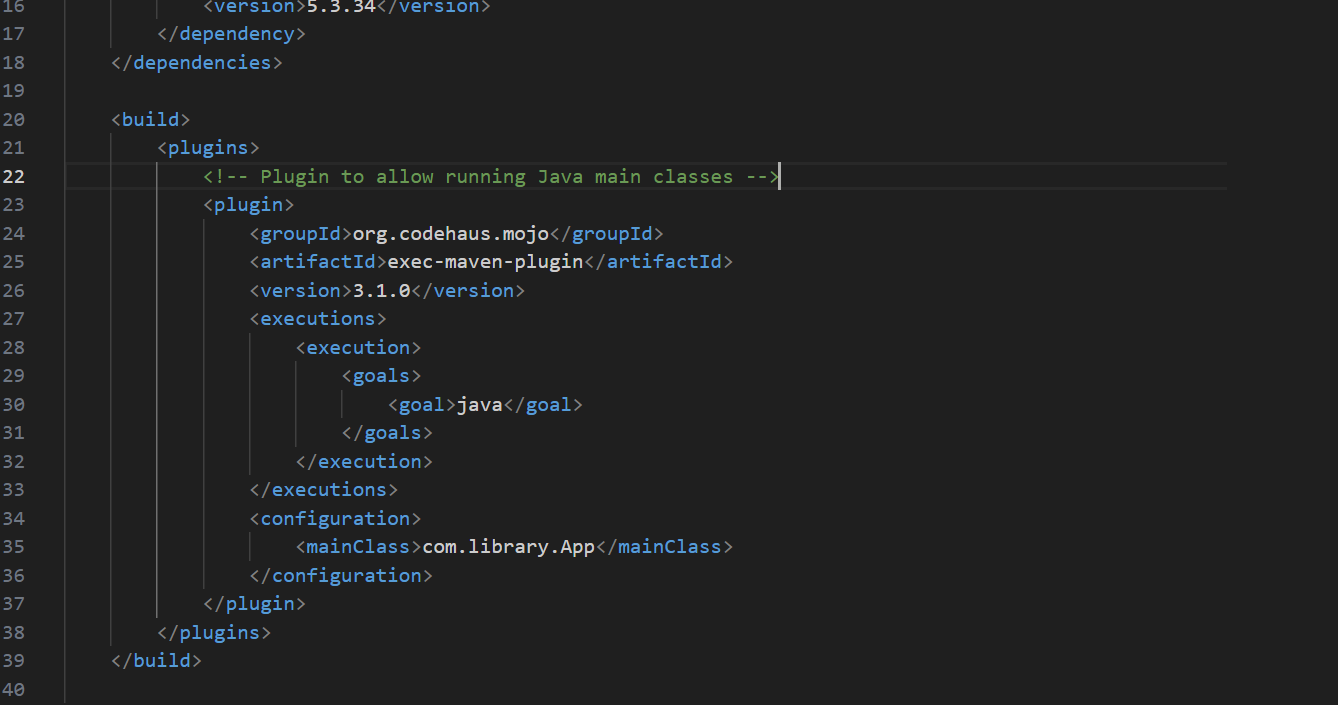
            </plugin>

        </plugins>

    </build>

</project>





Output:

