WEEK – 3

Exercise 5: Configuring the Spring IoC Container

Nipuna.A

Superset id: 6432842

**Scenario:**

The library management application requires a central configuration for beans and dependencies.

**Steps:**

1. **Create Spring Configuration File:**
   * Create an XML configuration file named **applicationContext.xml** in the **src/main/resources** directory.
   * Define beans for **BookService** and **BookRepository** in the XML file.
2. **Update the BookService Class:**
   * Ensure that the **BookService** class has a setter method for **BookRepository**.
3. **Run the Application:**
   * Create a main class to load the Spring context and test the configuration.

Solution:

1)Book.java

package com.library.model;

public class Book {

    private String title;

    public Book(String title) {

        this.title = title;

    }

    public String getTitle() {

        return title;

    }

}



2)BookRepository.java

package com.library.repository;

import com.library.model.Book;

public class BookRepository {

    public Book getBook() {

        return new Book("Spring in Action");

    }

}



3)BookService.java

package com.library.service;

import com.library.model.Book;

import com.library.repository.BookRepository;

public class BookService {

    private BookRepository bookRepository;

    public void setBookRepository(BookRepository bookRepository) {

        this.bookRepository = bookRepository;

    }

    public void printBookTitle() {

        Book book = bookRepository.getBook();

        System.out.println("Book title: " + book.getTitle());

    }

}

4)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) {

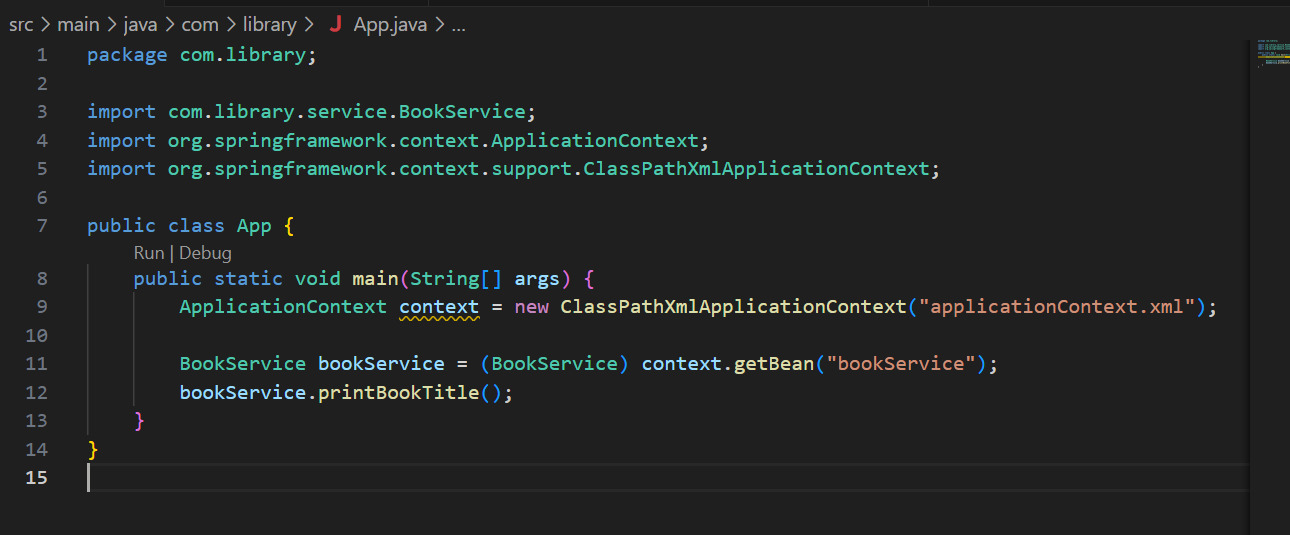
        ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");

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

        bookService.printBookTitle();

    }

}



5)ApplicationContext.xml

<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" />

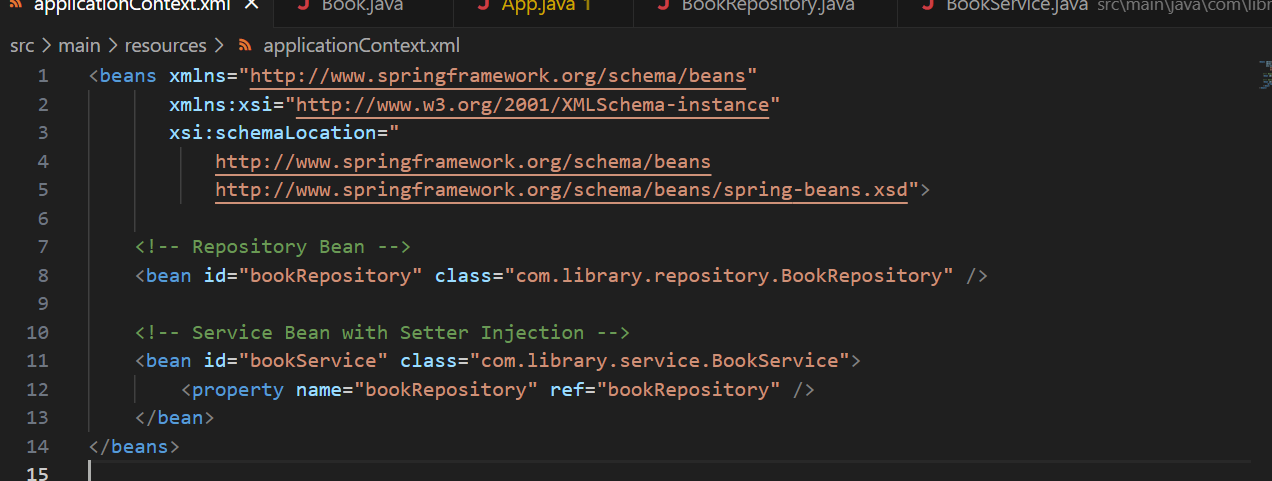
    <!-- Service Bean with Setter Injection -->

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

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

    </bean>

</beans>



6)Ppom.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

                             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>

    <properties>

        <maven.compiler.source>1.8</maven.compiler.source>

        <maven.compiler.target>1.8</maven.compiler.target>

    </properties>

    <dependencies>

        <!-- Spring Core Context for XML-based IoC -->

        <dependency>

            <groupId>org.springframework</groupId>

            <artifactId>spring-context</artifactId>

            <version>5.3.34</version>

        </dependency>

    </dependencies>

    <build>

        <plugins>

            <!-- Plugin to run Java class with main() -->

            <plugin>

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

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

                <version>3.1.0</version>

                <configuration>

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

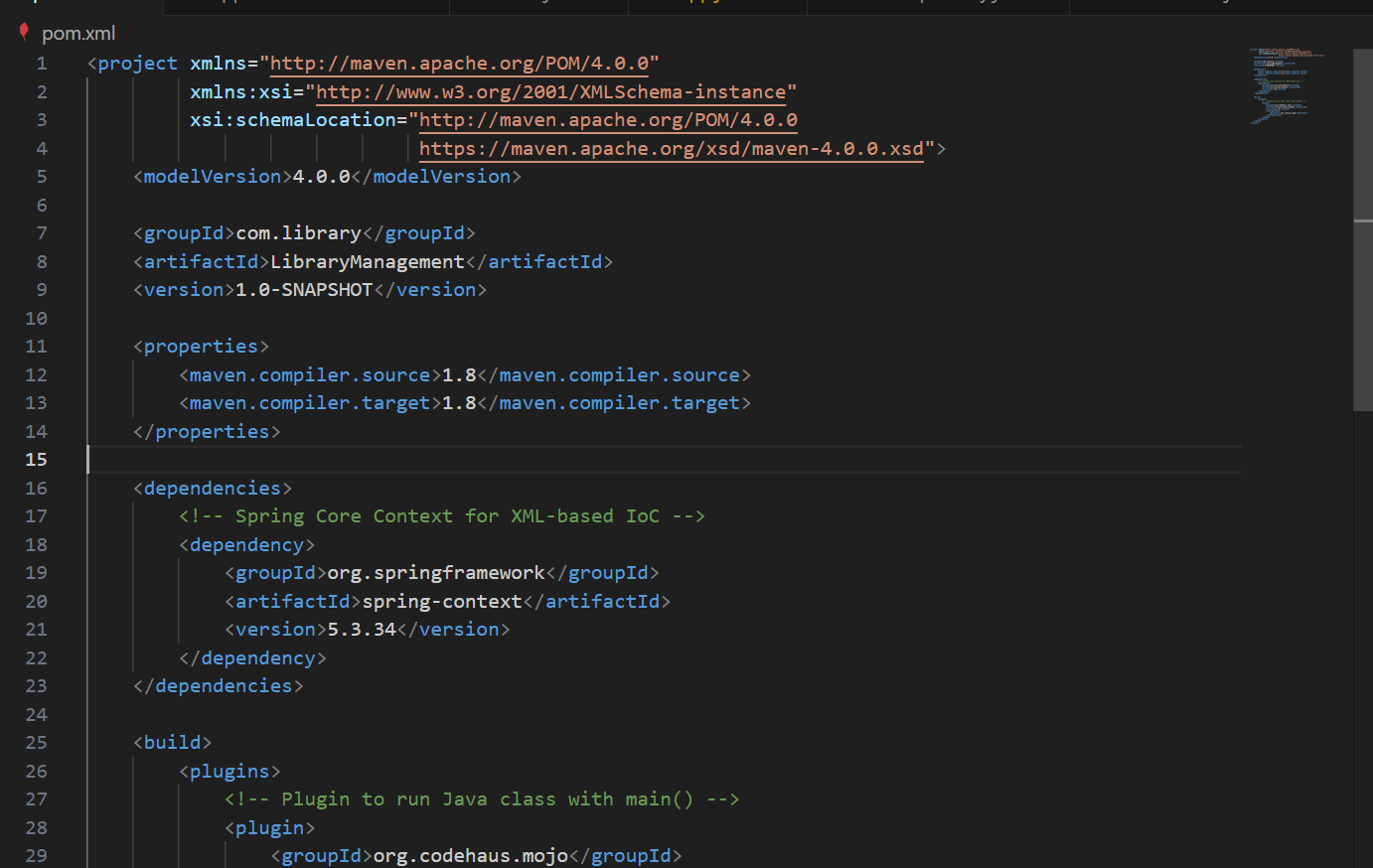
                </configuration>

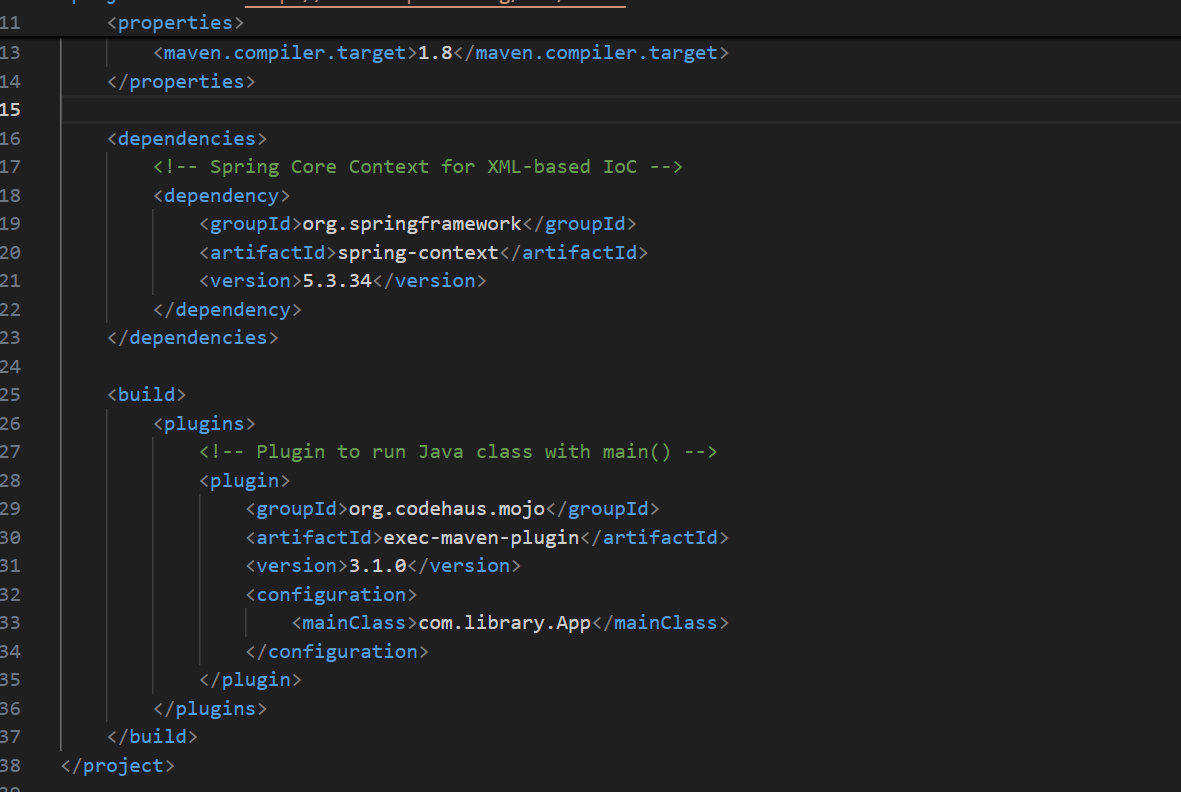
            </plugin>

        </plugins>

    </build>

</project>





Output:

