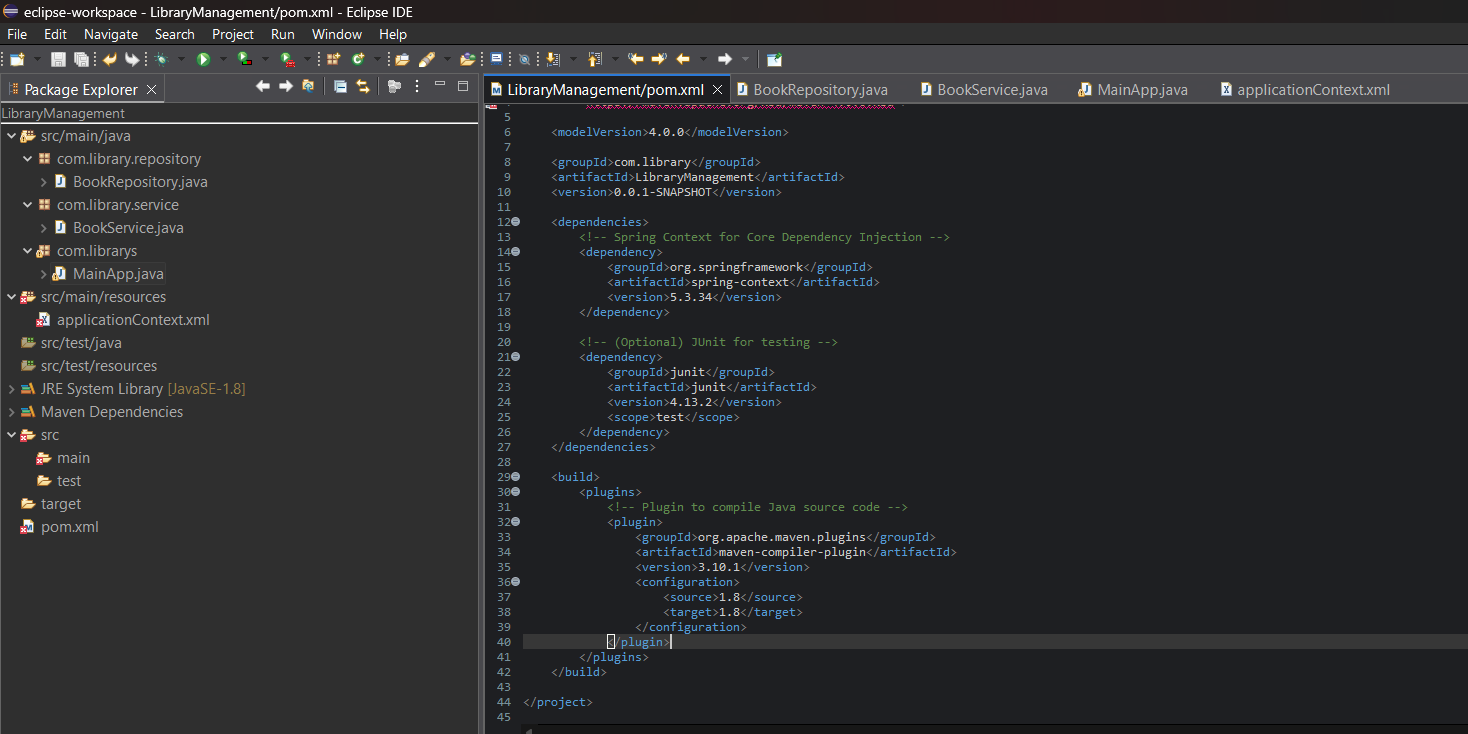
Spring Core and Maven

## Exercise 1: Configuring a Basic Spring Application



**Dependencies to be added on pom.xml :**

<dependencies>

<!-- Spring Core Dependency -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.34</version> <!-- Latest stable version -->

</dependency>

</dependencies>

**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>

**BookRepository.java :**

package com.library.repository;

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Book saved: " + bookName);

}

}

**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 in service layer: " + bookName);

bookRepository.saveBook(bookName);

}

}

**MainApp.java :**

package com.librarys;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

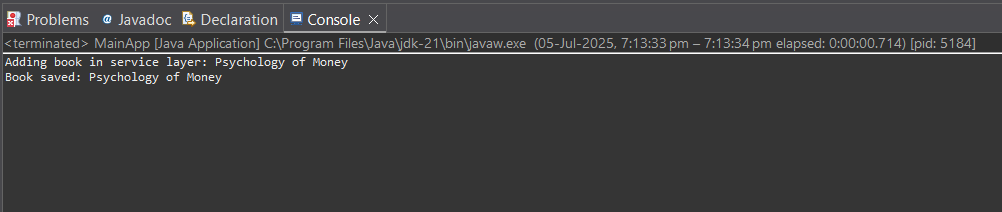
BookService bookService = context.getBean("bookService", BookService.class);

bookService.addBook("Psychology of Money");

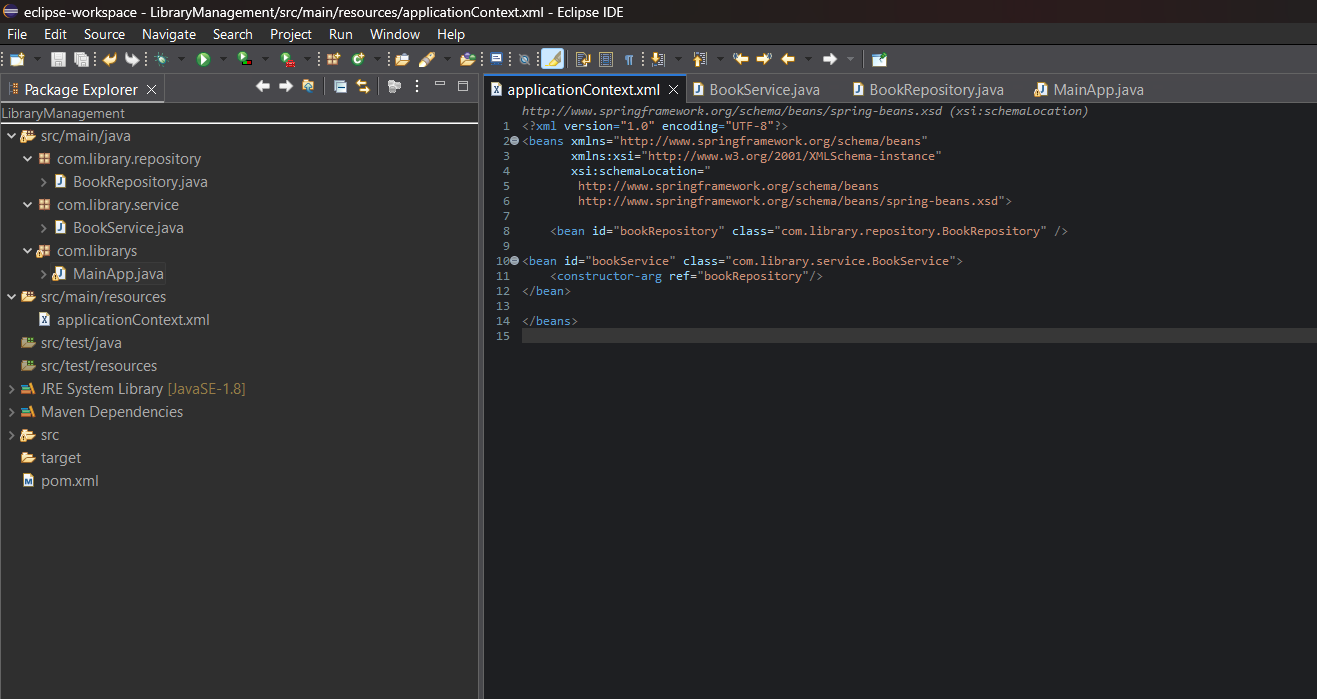
}

}

**OUTPUT :**

****

## Exercise 2: Implementing Dependency Injection



**BookService.java :**

package com.library.service;

import com.library.repository.BookRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service

public class BookService {

private final BookRepository bookRepository;

@Autowired

public BookService(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String bookName) {

System.out.println("Starting book registration process...");

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

if (bookName.length() < 3) {

System.out.println("Book name is too short. Cannot save.");

return;

}

bookRepository.saveBook(bookName);

System.out.println("Book \"" + bookName + "\" successfully added to the catalog.");

System.out.println("Process complete.\n");

}

}

**BookRepository.java :**

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Saving book to repository: " + bookName);

}

}

**MainApp.java :**

package com.librarys;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

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

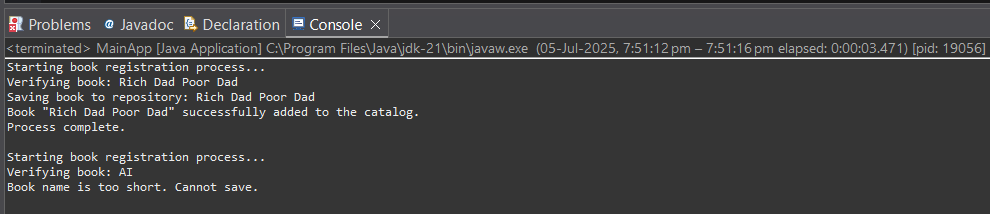
bookService.addBook("Rich Dad Poor Dad");

bookService.addBook("AI");

}

}

**OUTPUT :**



## Exercise 4: Creating and Configuring a Maven Project

**Dependencies to be added on pom.xml :**

<dependencies>

<!-- Spring Context (core DI container) -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.34</version>

</dependency>

<!-- Spring AOP (optional, for cross-cutting concerns like logging) -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.34</version>

</dependency>

<!-- Spring WebMVC (for controllers, DispatcherServlet, etc.) -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>5.3.34</version>

</dependency>

<!-- Servlet API (required for Spring WebMVC apps, marked as provided) -->

<dependency>

<groupId>javax.servlet</groupId>

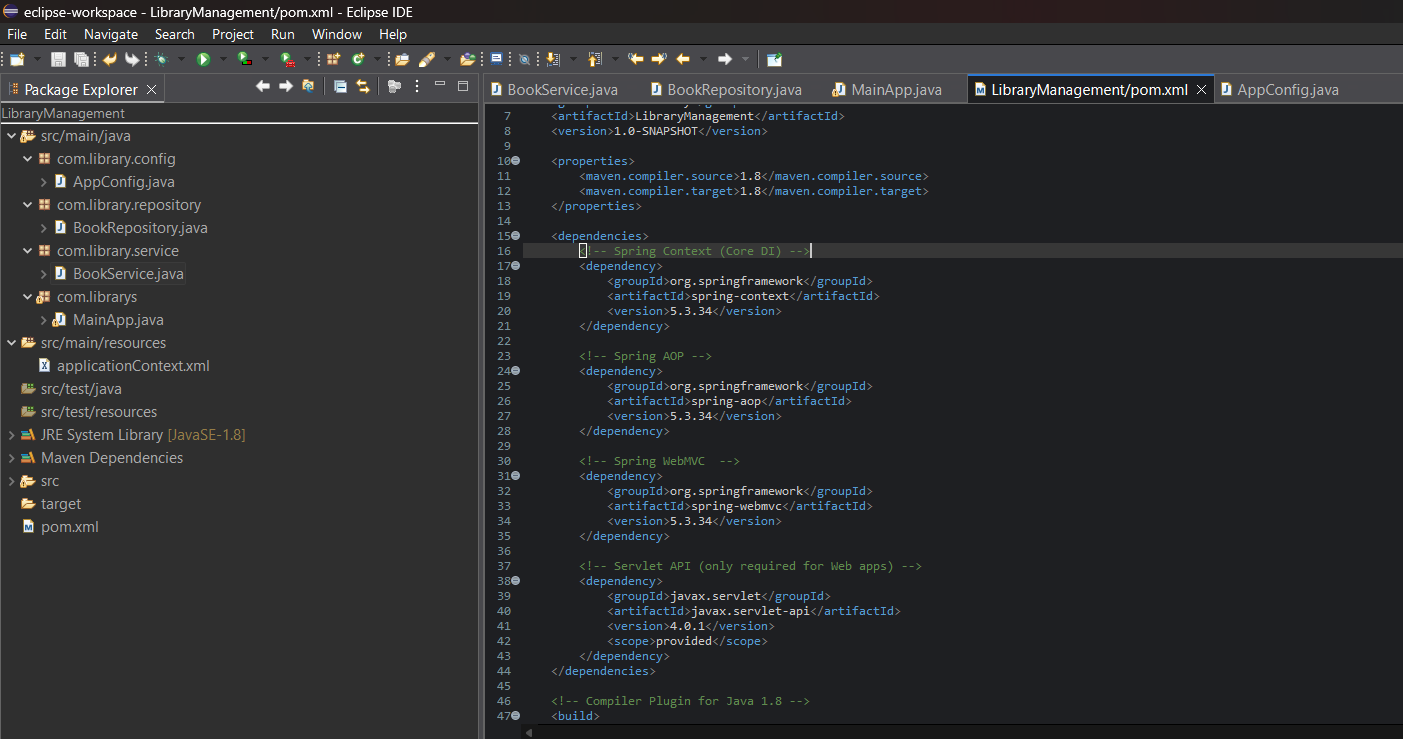
<artifactId>javax.servlet-api</artifactId>

<version>4.0.1</version>

<scope>provided</scope>

</dependency>

</dependencies>



**AppConfig.java :**

package com.library.config;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan(basePackages = "com.library")

public class AppConfig {

}

**BookRepository.java :**

package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository

public class BookRepository {

public void saveBook(String bookName) {

System.out.println("Book saved to repository: " + bookName);

}

}

**BookService.java :**

package com.library.service;

import com.library.repository.BookRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service

public class BookService {

@Autowired

private BookRepository bookRepository;

public void addBook(String bookName) {

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

bookRepository.saveBook(bookName);

}

}

**MainApp.java :**

package com.librarys;

import com.library.config.AppConfig;

import com.library.service.BookService;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class MainApp {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

BookService service = context.getBean(BookService.class);

service.addBook("The Psychology of Money");

}

}

**OUTPUT :**

