

Exercise 2: Implementing Dependency Injection

FROM FILE : Spring Core_Maven

1. Modifying the XML Configuration:

.Update applicationContext.xml to wire BookRepository into BookService.

Code:

```
<?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 for BookRepository -->

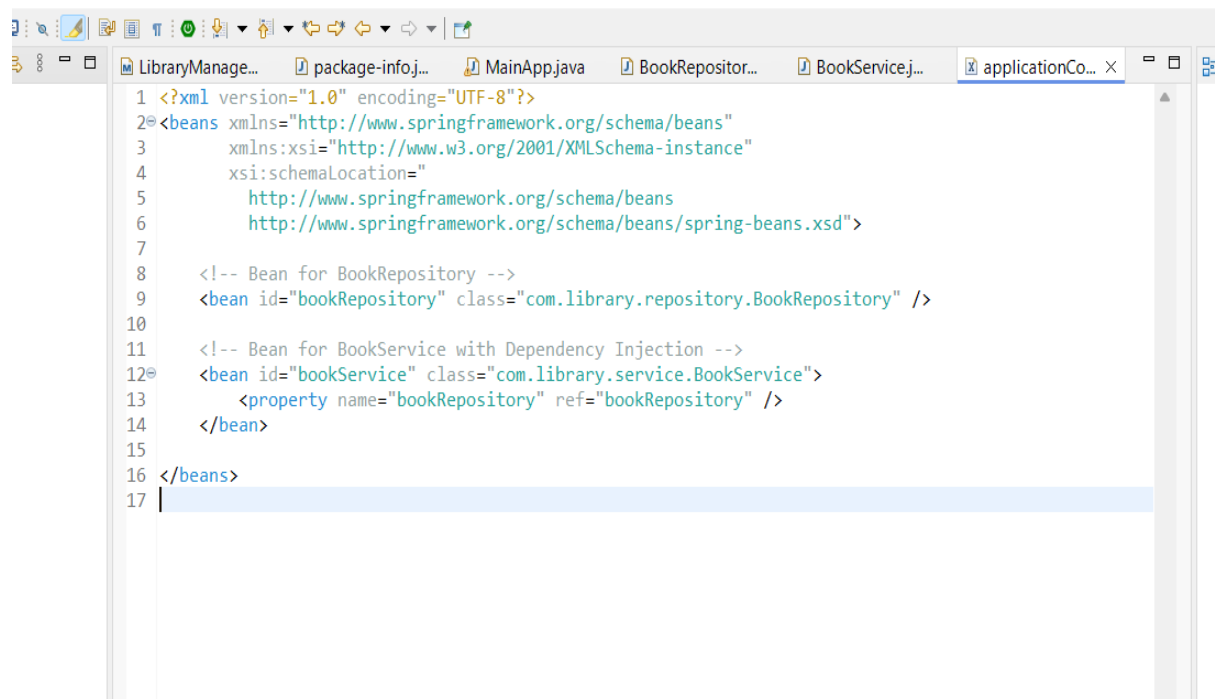
    <bean id="bookRepository"
class="com.library.repository.BookRepository" />

    <!-- Bean for BookService with Dependency Injection -->

    <bean id="bookService" class="com.library.service.BookService">
        <property name="bookRepository" ref="bookRepository" />
    </bean>

</beans>
```

PIC From SpringTool:



Updating the BookService Class:

.Ensure that BookService class has a setter method for BookRepository.

Code:

```
package com.library.service;
```

```
import com.library.repository.BookRepository;
```

```
public class BookService {
```

```
    private BookRepository bookRepository;
```

```
    // Setter method for Dependency Injection
```

```
    public void setBookRepository(BookRepository bookRepository) {
```

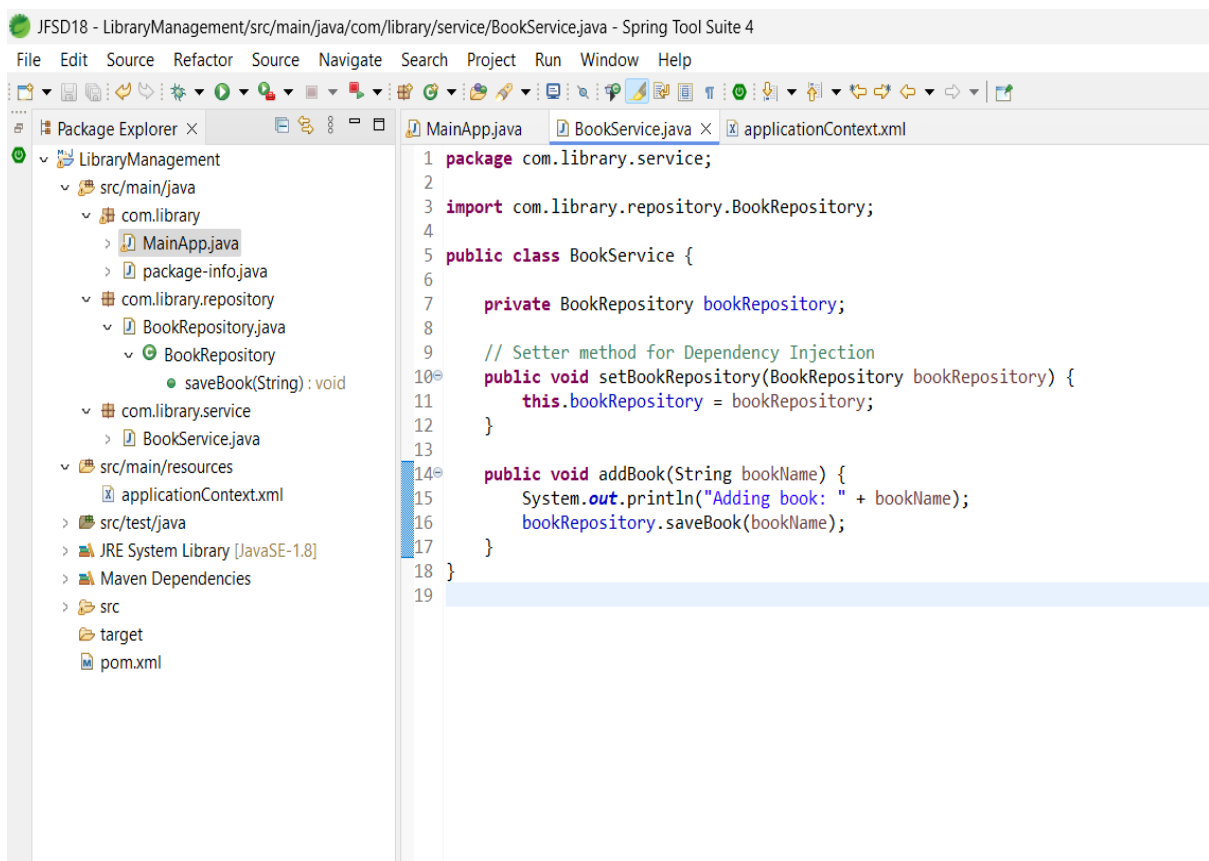
```
        this.bookRepository = bookRepository;
```

```
}
```

```
public void addBook(String bookName) {  
    System.out.println("Adding book: " + bookName);  
    bookRepository.saveBook(bookName);  
}
```

```
}
```

PIC From SpringTool:



MainApp.java class:

Code:

```
package com.library;

import org.springframework.context.ApplicationContext;

import
org.springframework.context.support.ClassPathXmlApplicationConte
xt;

import com.library.service.BookService;

public class MainApp {

    public static void main(String[] args) {

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

        // Retrieve the BookService bean

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

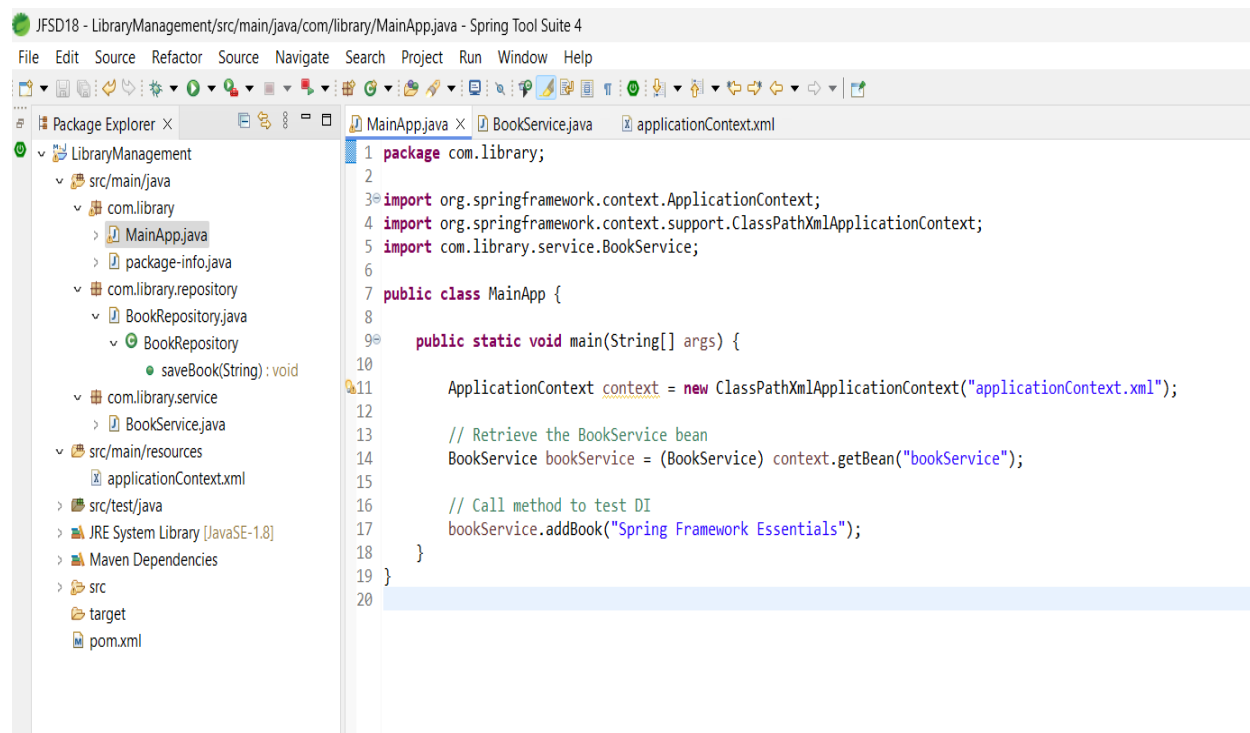
        // Call method to test DI

        bookService.addBook("Spring Framework Essentials");

    }

}
```

PIC FROM SpringTool:



Test the Configuration:

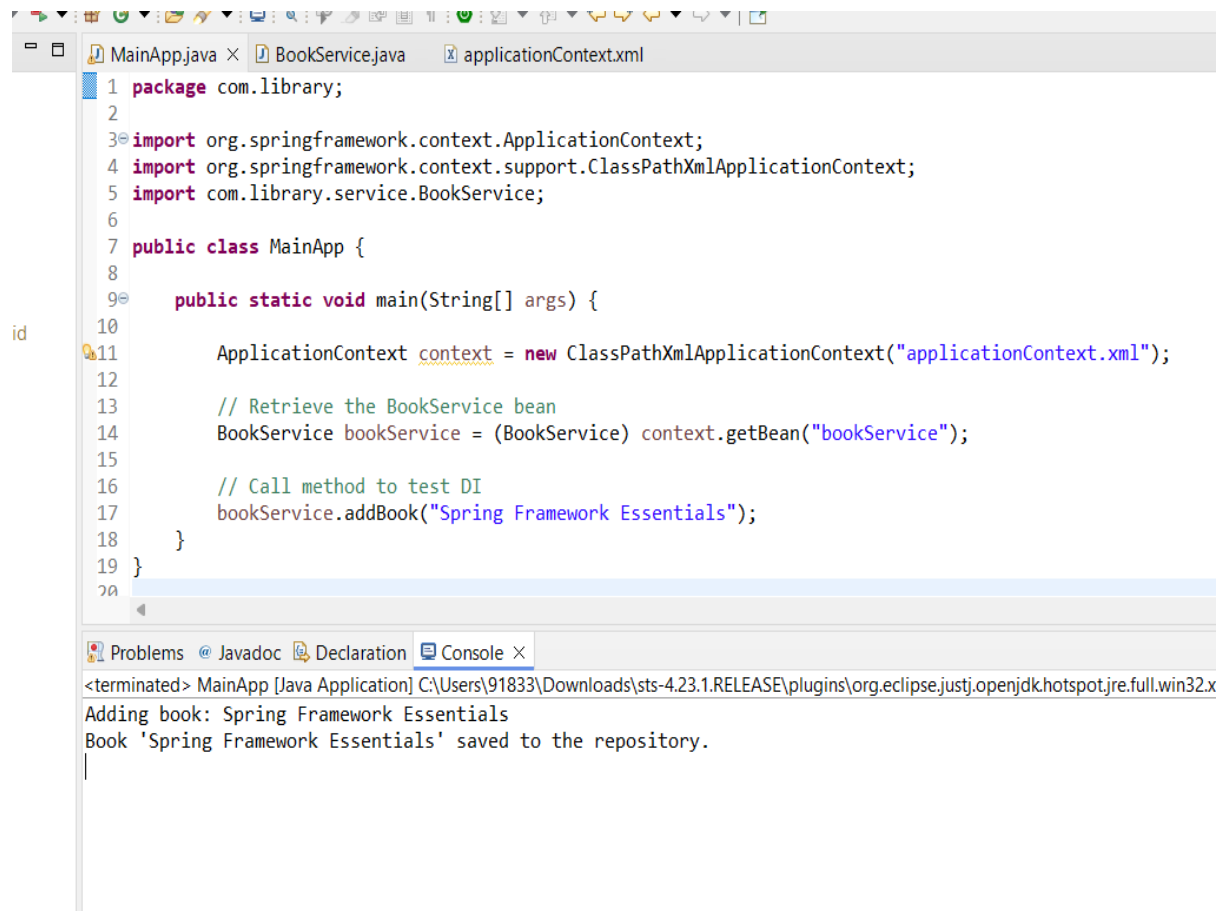
Run the LibraryManagementApplication main class to verify the dependency injection.

OUTPUT :

Adding book: Spring Framework Essentials

Book 'Spring Framework Essentials' saved to the repository.

PIC FROM SpringTool:



```
1 package com.library;
2
3 import org.springframework.context.ApplicationContext;
4 import org.springframework.context.support.ClassPathXmlApplicationContext;
5 import com.library.service.BookService;
6
7 public class MainApp {
8
9     public static void main(String[] args) {
10
11         ApplicationContext context = new ClassPathXmlApplicationContext("applicationContext.xml");
12
13         // Retrieve the BookService bean
14         BookService bookService = (BookService) context.getBean("bookService");
15
16         // Call method to test DI
17         bookService.addBook("Spring Framework Essentials");
18     }
19 }
20
```

Problems @ Javadoc Declaration Console X

<terminated> MainApp [Java Application] C:\Users\91833\Downloads\sts-4.23.1.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x
Adding book: Spring Framework Essentials
Book 'Spring Framework Essentials' saved to the repository.