**Spring Core and Maven**

**Exercise 1: Configuring a Basic Spring Application**

**//Adding Dependencies**

<dependencies>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

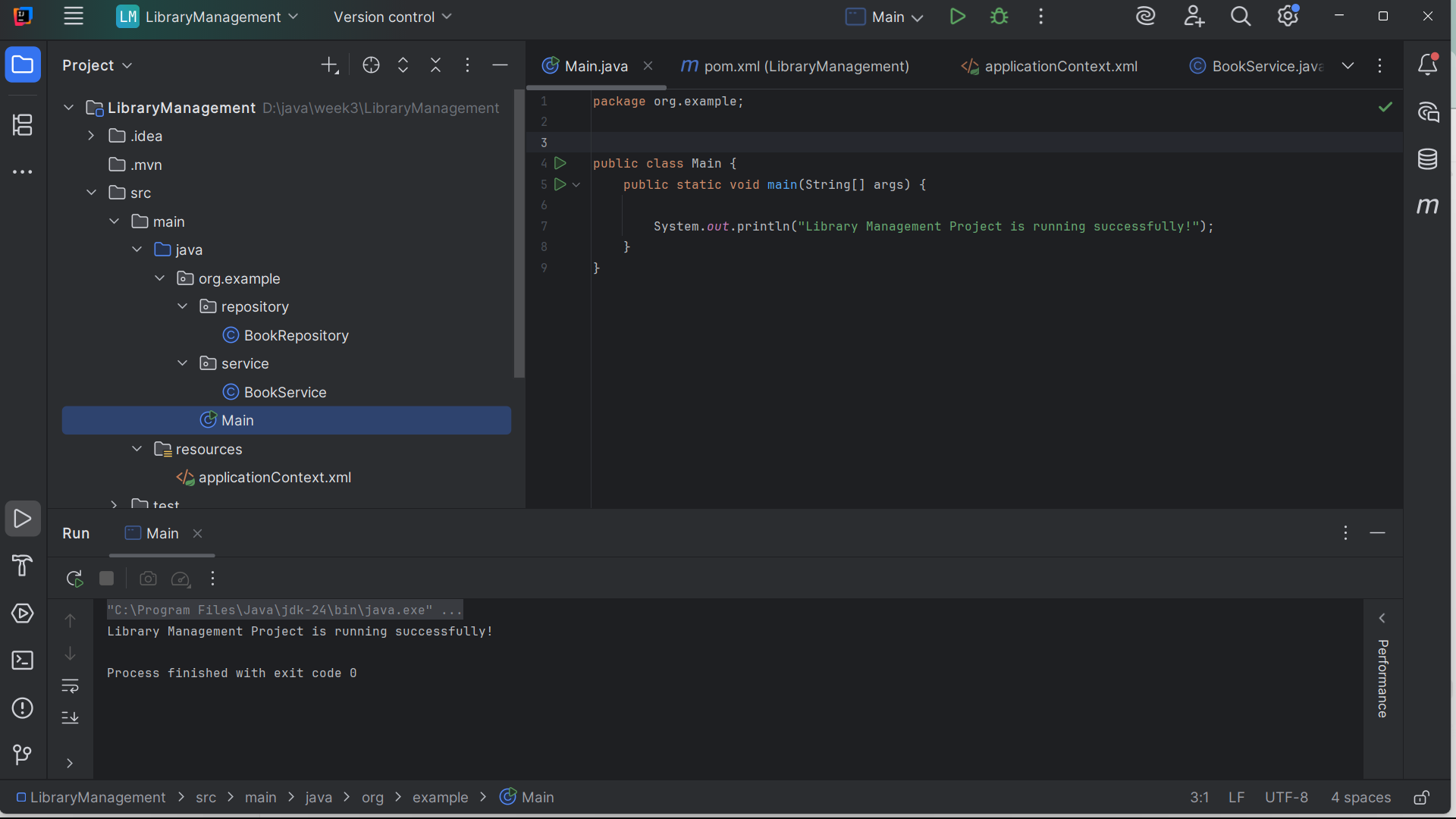
<version>6.2.8</version>

</dependency>

</dependencies>

**//Defining beans for BookService and BookRepository in the 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="org.example.repository.BookRepository"/>  
 <bean id="bookService" class="org.example.service.BookService">  
 </bean>  
</beans>



**Exercise 2: Implementing Dependency Injection**

**//applicationContext.xml file**

<?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="org.example.repository.BookRepository"/>  
  
  
 <bean id="bookService" class="org.example.service.BookService">  
 <property name="bookRepository" ref="bookRepository"/>  
 </bean>  
</beans>

**//BookService Clas**s

package org.example.service;  
  
import org.example.repository.BookRepository;

public class BookService {  
 private BookRepository bookRepository;  
 // Setter for Spring XML injection  
 public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
 }  
  
 public void displayBooks() {  
 System.*out*.println("Inside BookService...");  
 bookRepository.fetchBooks();  
 }  
}

**//BookRepoistory Class**

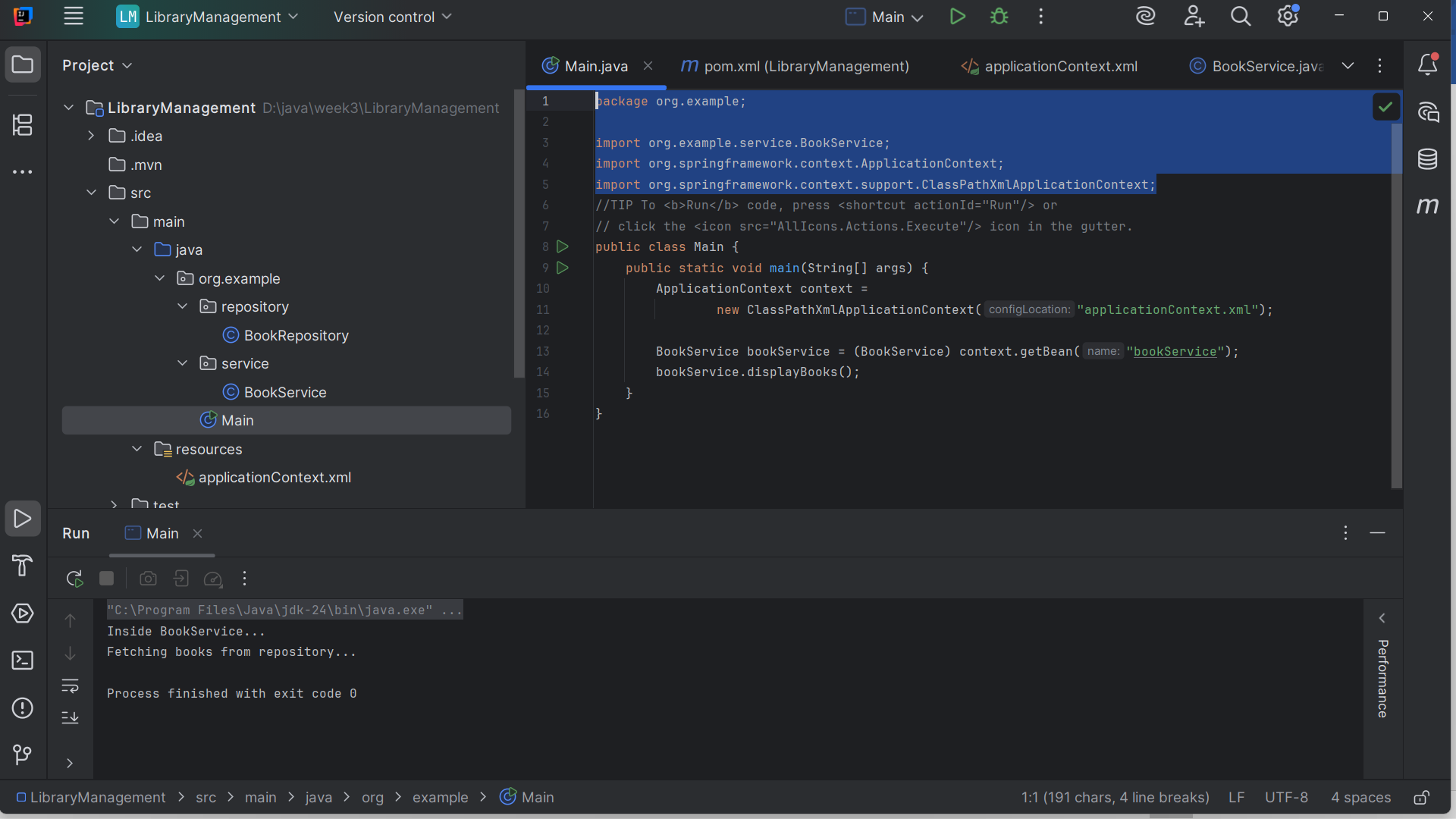
package org.example.repository;

public class BookRepository {  
 public void fetchBooks() {  
 System.*out*.println("Fetching books from repository...");  
 }  
}

**//Main Class**

package org.example;  
  
import org.example.service.BookService;  
import org.springframework.context.ApplicationContext;  
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class Main {  
 public static void main(String[] args) {  
 ApplicationContext context =  
 new ClassPathXmlApplicationContext("applicationContext.xml");  
  
 BookService bookService = (BookService) context.getBean("bookService");  
 bookService.displayBooks();  
 }  
}

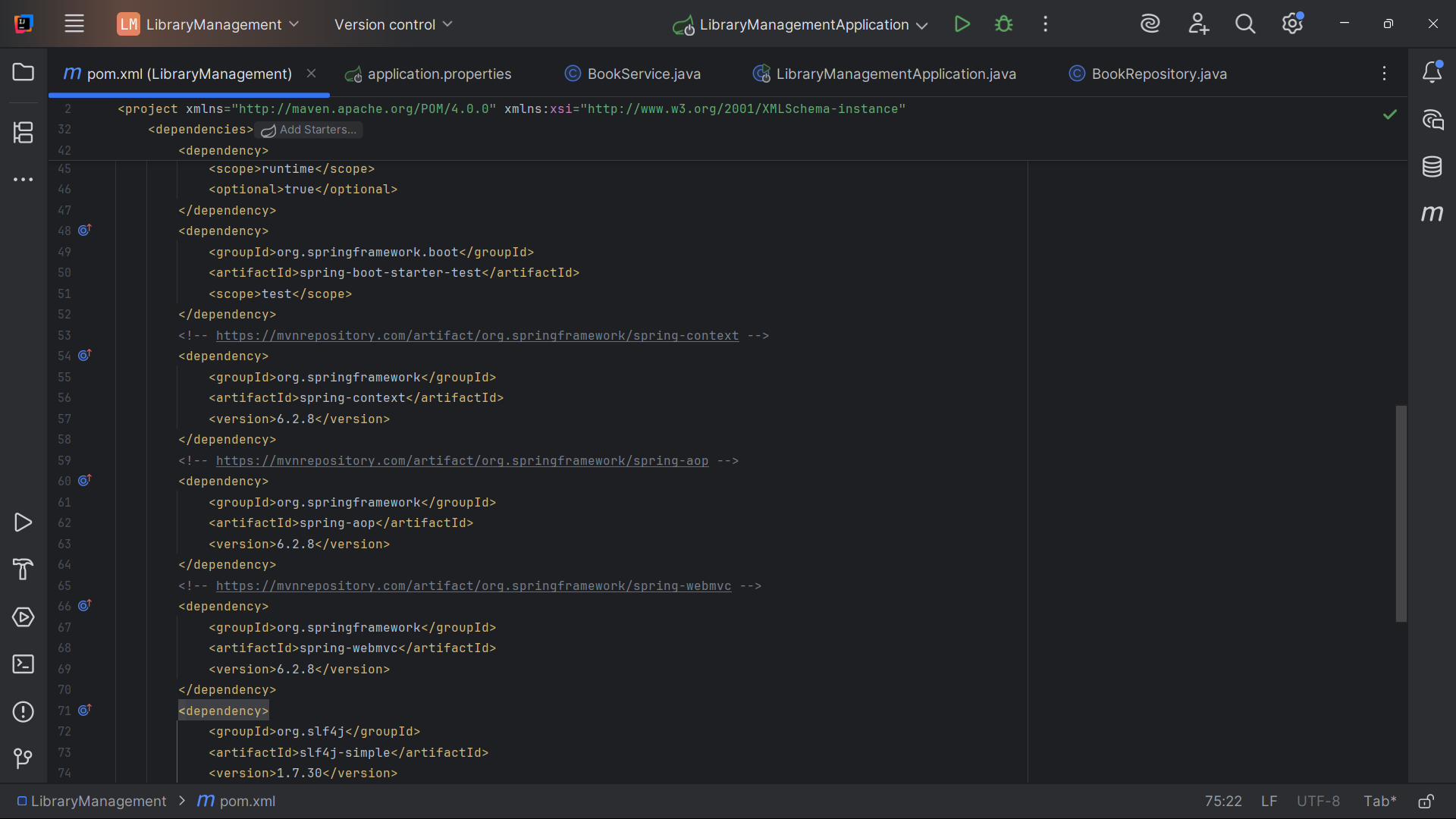


**Exercise 4: Creating and Configuring a Maven Project**

**Add Spring Dependencies in pom.xml**

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
</dependency>  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-web</artifactId>  
</dependency>  
<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-devtools</artifactId>  
 <scope>runtime</scope>  
 <optional>true</optional>  
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-context -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>6.2.8</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/org.springframework/spring-aop -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>6.2.8</version>  
 </dependency>  
 <!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-webmvc</artifactId>  
 <version>6.2.8</version>  
 </dependency>  
</dependencies>  
<build>  
 <plugins>  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.10.1</version>  
 <configuration>  
 <source>1.8</source>  
 <target>1.8</target>  
 </configuration>  
 </plugin>  
 </plugins>  
</build>



**//BookService class**

package com.example.LibraryManagement.service;  
  
import com.example.LibraryManagement.repository.BookRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class BookService {  
  
 @Autowired  
 private BookRepository bookRepository;  
  
 public void displayBooks() {  
 System.*out*.println("Inside BookService...");  
 bookRepository.fetchBooks();  
 }

public BookRepository getBookRepository() {  
 return bookRepository;  
}  
  
public void setBookRepository(BookRepository bookRepository) {  
 this.bookRepository = bookRepository;  
}

}

**//BookRepository**

package com.example.LibraryManagement.repository;  
  
import org.springframework.stereotype.Repository;  
  
@Repository  
public class BookRepository {  
 public void fetchBooks() {  
 System.*out*.println("Fetching books from repository...");  
 }  
}

**//Main class**

package com.example.LibraryManagement;  
  
import com.example.LibraryManagement.service.BookService;  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
import org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration;  
import org.springframework.context.ApplicationContext;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import static org.springframework.boot.autoconfigure.liquibase.LiquibaseProperties.UiService.*LOGGER*;  
@SpringBootApplication(exclude = {DataSourceAutoConfiguration.class})  
public class LibraryManagementApplication {  
 private static final Logger *LOGGER* = LoggerFactory.*getLogger*(LibraryManagementApplication.class);  
 public static void main(String[] args) {  
 *LOGGER*.info("Starting Library Management Application");  
  
 ApplicationContext context =SpringApplication.*run*(LibraryManagementApplication.class, args);  
 BookService bookService = context.getBean(BookService.class);  
  
 *LOGGER*.debug("Calling BookService.displayBooks()");  
 bookService.displayBooks();  
  
 *LOGGER*.info("Library Management Application Ended");  
 } }

