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

**Question:** Your company is developing a web application for managing a library. You need to use the Spring Framework to handle the backend operations.

**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>

**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() {  
 System.*out*.println("Adding book...");  
 bookRepository.saveBook();  
 }  
}

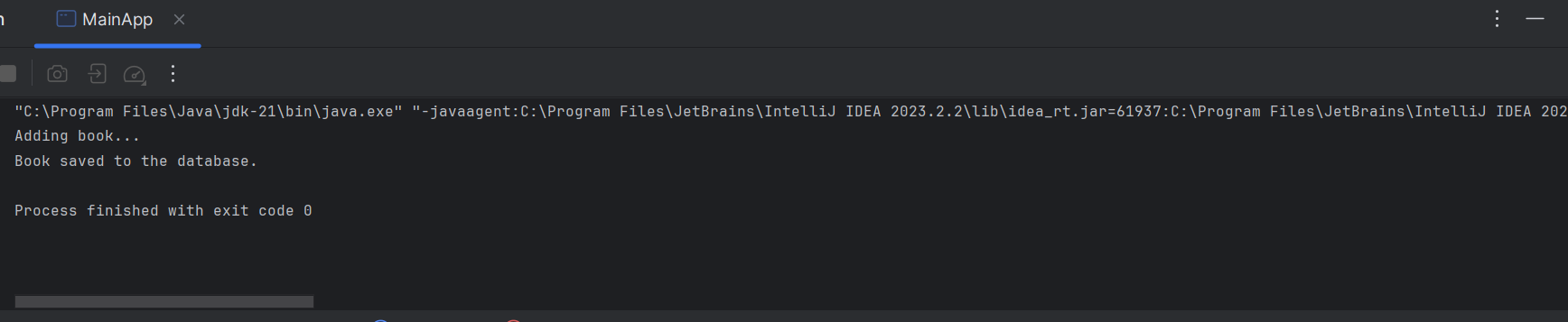
**BookRepository.java:**

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook() {  
 System.*out*.println("Book saved to the database.");  
 }  
}

**MainApp.java:**

package com.library;  
  
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();  
 }  
}

**OUTPUT:**



**Exercise 2: Implementing Dependency Injection**

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

**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>

**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() {  
 System.*out*.println("Adding Marvel Comics...");  
 bookRepository.saveBook();  
 }  
}

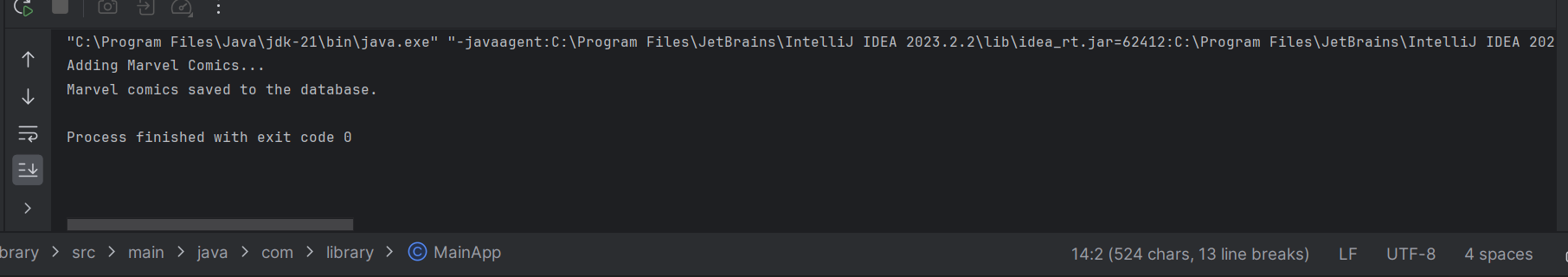
**BookRepository.java:**

package com.library.repository;  
  
public class BookRepository {  
 public void saveBook() {  
 System.*out*.println("Marvel comics saved to the database.");  
 }  
}

**MainApp.java:**

package com.library;  
  
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(); // should trigger BookRepository method  
 }  
}

**OUTPUT:**



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

**Question:** You need to set up a new Maven project for the library management application and add Spring dependencies.

**Pom.xml:**

<?xml version="1.0" encoding="UTF-8"?>  
<project xmlns="http://maven.apache.org/POM/4.0.0" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-context</artifactId>  
 <version>5.3.32</version>  
 <name>Spring Context</name>  
 <description>Spring Context</description>  
 <url>https://github.com/spring-projects/spring-framework</url>  
 <organization>  
 <name>Spring IO</name>  
 <url>https://spring.io/projects/spring-framework</url>  
 </organization>  
 <licenses>  
 <license>  
 <name>Apache License, Version 2.0</name>  
 <url>https://www.apache.org/licenses/LICENSE-2.0</url>  
 <distribution>repo</distribution>  
 </license>  
 </licenses>  
 <developers>  
 <developer>  
 <id>jhoeller</id>  
 <name>Juergen Hoeller</name>  
 <email>jhoeller@pivotal.io</email>  
 </developer>  
 </developers>  
 <scm>  
 <connection>scm:git:git://github.com/spring-projects/spring-framework</connection>  
 <developerConnection>scm:git:git://github.com/spring-projects/spring-framework</developerConnection>  
 <url>https://github.com/spring-projects/spring-framework</url>  
 </scm>  
 <issueManagement>  
 <system>GitHub</system>  
 <url>https://github.com/spring-projects/spring-framework/issues</url>  
 </issueManagement>  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-aop</artifactId>  
 <version>5.3.32</version>  
 <scope>compile</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-beans</artifactId>  
 <version>5.3.32</version>  
 <scope>compile</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-core</artifactId>  
 <version>5.3.32</version>  
 <scope>compile</scope>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework</groupId>  
 <artifactId>spring-expression</artifactId>  
 <version>5.3.32</version>  
 <scope>compile</scope>  
 </dependency>  
 </dependencies>  
</project>