Spring Core Maven

Exercise 1: Configuring a Basic Spring Application

Code:

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-3.1.xsd"*>

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

</bean>

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

<constructor-arg ref=*"bookRepository"*/>

</bean>

</beans>

BookRepository.java:

**package** com.library.repository;

**import** java.util.List;

**public** **class** BookRepository {

**public** BookRepository() {

System.***out***.println("BookRepository instance created.");

}

**public** List<String> findAllBooks() {

System.***out***.println("BookRepository: Finding all books...");

**return** List.*of*("The Great Gatsby", "1984", "To Kill a Mockingbird");

}

}

BookService.java:

**package** com.library.service;

**import** com.library.repository.BookRepository;

**import** java.util.List;

**public** **class** BookService {

**private** **final** BookRepository bookRepository;

**public** BookService(BookRepository bookRepository) {

**this**.bookRepository = bookRepository;

System.***out***.println("BookService instance created with BookRepository: " + bookRepository);

}

**public** List<String> getAllBooks() {

System.***out***.println("BookService: Getting all books...");

**return** bookRepository.findAllBooks();

}

}

Main.java:

**package** com.library;

**import** com.library.service.BookService;

**import** org.springframework.context.ApplicationContext;

**import** org.springframework.context.support.ClassPathXmlApplicationContext;

**import** java.util.List; // <--- ADD THIS IMPORT STATEMENT

**public** **class** Main {

**public** **static** **void** main(String[] args) {

System.***out***.println("Starting Spring Application Context loading...");

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

System.***out***.println("Spring Application Context loaded successfully.");

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

System.***out***.println("BookService bean retrieved from context: " + bookService);

System.***out***.println("\n--- Listing Books ---");

List<String> books = bookService.getAllBooks();

books.forEach(System.***out***::println);

System.***out***.println("--- End of Book List ---\n");

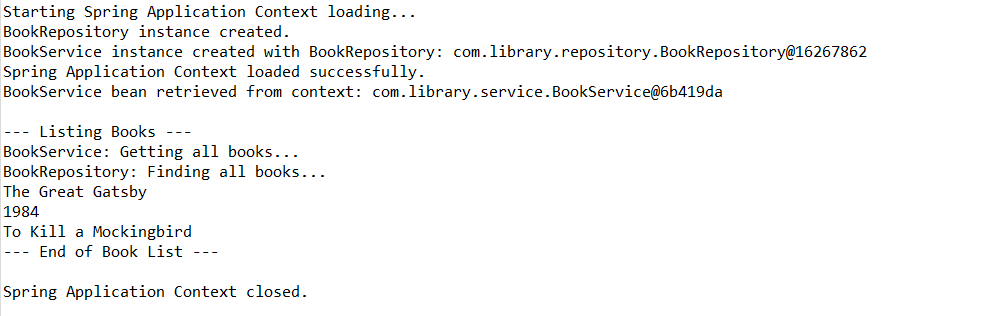
((ClassPathXmlApplicationContext) context).close();

System.***out***.println("Spring Application Context closed.");

}

}

Output:



Exercise 2: Implementing Dependency Injection

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-3.1.xsd"*>

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

<!-- No dependencies for BookRepository itself -->

</bean>

<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;

**import** java.util.List;

**public** **class** BookService {

**private** BookRepository bookRepository;

**public** BookService() {

System.***out***.println("BookService instance created (via default constructor).");

}

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

**this**.bookRepository = bookRepository;

System.***out***.println("BookRepository injected into BookService via setter: " + bookRepository);

}

**public** List<String> getAllBooks() {

**if** (bookRepository == **null**) {

System.***err***.println("Error: BookRepository not injected into BookService!");

**return** List.*of*();

}

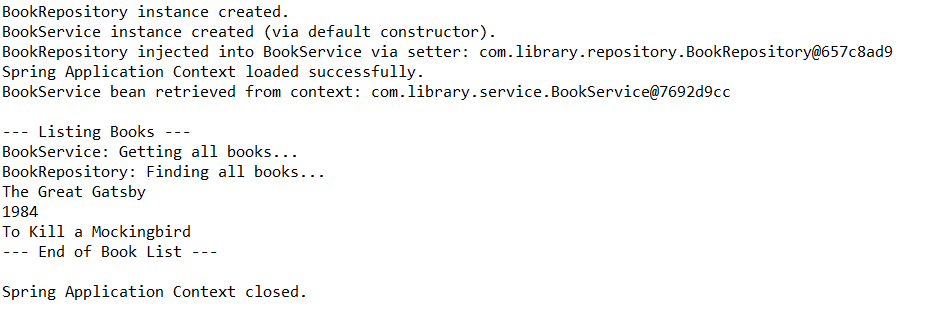
System.***out***.println("BookService: Getting all books...");

**return** bookRepository.findAllBooks();

}

}

Output:



Exercise 4: Creating and Configuring a Maven Project

Pom.xml:

<?xml version="1.0" encoding="UTF-8"?>

<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 http://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>

<spring.version>5.3.27</spring.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<source>${maven.compiler.source}</source>

<target>${maven.compiler.target}</target>

</configuration>

</plugin>

</plugins>

</build>

</project>