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

**pom.xml**

<dependencies>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.3.34</version>

</dependency>

</dependencies>

**BookRepository.java**

package org.library.repository;

public class BookRepository {

public void saveBook(String title) {

System.out.println("Repository Action: Book '" + title + "' has been archived.");

}

}

**BookService.java**

package org.library.service;

import org.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String title) {

System.out.println("BookService Log: Preparing to archive new book.");

bookRepository.saveBook(title);

}

}

**Main.java**

package org.library;

import org.library.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.addBook("Spring in Action");

}

}

**resources/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.library.repository.BookRepository"/>

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

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

**Output :**



**Exercise 2: Implementing Dependency Injection**

**resources/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.library.repository.BookRepository"/>

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

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

**BookRepository.java**

package org.library.repository;

public class BookRepository {

public void saveBook(String title) {

System.out.println("Repository Info: Book '" + title + "' successfully saved in the database.");

}

}

**BookService.java**

package org.library.service;

import org.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String title) {

System.out.println("Service Alert: Initiating save process for the new book entry.");

bookRepository.saveBook(title);

}

}

**Main.java**

package org.library;

import org.library.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");

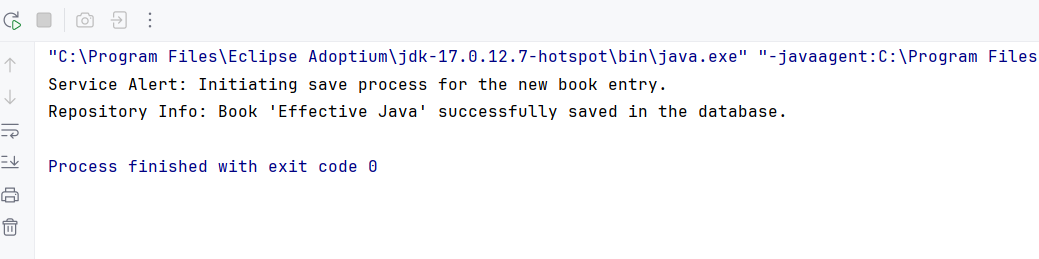
BookService bookService = (BookService) context.getBean("bookService");

bookService.addBook("Effective Java");

}

}

**Output :**



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

**pom.xml**

<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>

<java.version>1.8</java.version>

<spring.version>5.3.34</spring.version>

<servlet.version>4.0.1</servlet.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>

<dependency>

<groupId>javax.servlet</groupId>

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

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

<scope>provided</scope>

</dependency>

</dependencies>

<build>

<pluginManagement>

<plugins>

<plugin>

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

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

<version>3.10.1</version>

<configuration>

<source>${java.version}</source>

<target>${java.version}</target>

</configuration>

</plugin>

</plugins>

</pluginManagement>

</build>

</project>

**Exercise 5: Configuring the Spring IoC Container**

**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.library.repository.BookRepository"/>

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

<property name="bookRepository" ref="bookRepository"/>

</bean>

</beans>

**BookRepository.java**

package org.library.repository;

public class BookRepository {

public void saveBook(String title) {

System.out.println("Repository Log: Book '" + title + "' has been recorded successfully.");

}

}

**BookService.java**

package org.library.service;

import org.library.repository.BookRepository;

public class BookService {

private BookRepository bookRepository;

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String title) {

System.out.println("Service Notice: Processing request to add a book...");

bookRepository.saveBook(title);

}

}

**Main.java**

package org.library;

import org.library.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");

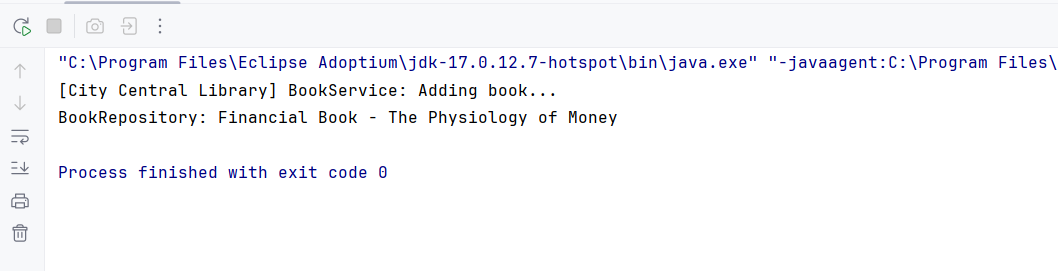
BookService bookService = (BookService) context.getBean("bookService");

bookService.addBook("Clean Code");

}

}

**Output :**



**Exercise 7: Implementing Constructor and Setter Injection**

**resources/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.library.repository.BookRepository"/>

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

<constructor-arg ref="bookRepository"/>

<property name="libraryName" value="Central Knowledge Hub"/>

</bean>

</beans>

**BookService.java**

package org.library.service;

import org.library.repository.BookRepository;

public class BookService {

private BookRepository repo;

private String libraryName;

public BookService(BookRepository repo) {

this.repo = repo;

}

public void setLibraryName(String libraryName) {

this.libraryName = libraryName;

}

public void registerBook(String bookTitle) {

System.out.println("Library [" + libraryName + "] is registering a new book...");

repo.saveBook(bookTitle);

}

}

**BookRepository.java**

package org.library.repository;

public class BookRepository {

public void saveBook(String title) {

System.*out*.println("BookRepository: Financial Book - " + title);

}

}

**Main.java**

package org.library;

import org.library.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");

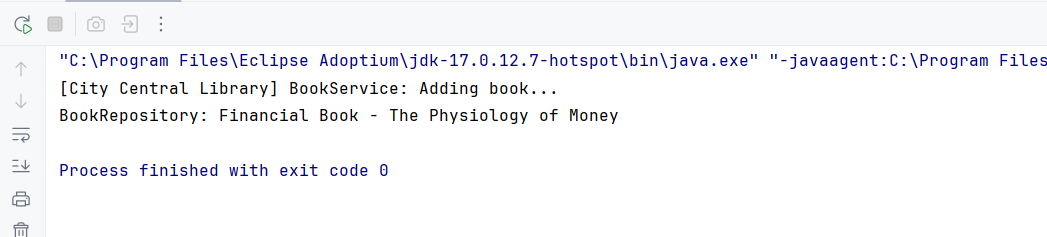
BookService service = (BookService) context.getBean("bookService");

service.registerBook("The Physiology of Money");

}

}

**Output :**



**Exercise 9: Creating a Spring Boot Application**

**LibraryManagementApplication.java**

package com.library.LibraryManagement;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class LibraryManagementApplication {

public static void main(String[] args) {

SpringApplication.run(LibraryManagementApplication.class, args);

}

}

**application.properties**

spring.application.name=LibraryManagement

spring.datasource.url=jdbc:h2:mem:librarydb

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.database-platform=org.hibernate.dialect.H2Dialect

spring.jpa.hibernate.ddl-auto=update

spring.h2.console.enabled=true

spring.h2.console.path=/h2-console

**BookController.java**

package com.library.LibraryManagement.controller;

import com.library.LibraryManagement.entity.Book;

import com.library.LibraryManagement.repository.BookRepository;

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

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/api/books")

public class BookController {

@Autowired

private BookRepository bookRepo;

@GetMapping

public List<Book> fetchAllBooks() {

return bookRepo.findAll();

}

@GetMapping("/{id}")

public Book fetchBook(@PathVariable("id") Long bookId) {

return bookRepo.findById(bookId).orElse(new Book("Not Found", "Unknown"));

}

@PostMapping

public String registerBook(@RequestBody Book book) {

Book saved = bookRepo.save(book);

return "Book '" + saved.getTitle() + "' by " + saved.getAuthor() + " added successfully.";

}

@PutMapping("/{id}")

public String modifyBook(@PathVariable("id") Long id, @RequestBody Book newDetails) {

return bookRepo.findById(id).map(b -> {

b.setTitle(newDetails.getTitle());

b.setAuthor(newDetails.getAuthor());

bookRepo.save(b);

return "Book with ID " + id + " updated.";

}).orElse("Book not found.");

}

@DeleteMapping("/{id}")

public String removeBook(@PathVariable("id") Long id) {

if (bookRepo.existsById(id)) {

bookRepo.deleteById(id);

return "Book with ID " + id + " deleted.";

} else {

return "Book not found.";

}

}

}

**Book.java**

package com.library.LibraryManagement.entity;

import jakarta.persistence.\*;

@Entity

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String author;

public Book() {}

public Book(String title, String author) {

this.title = title;

this.author = author;

}

public Long getId() { return id; }

public String getTitle() { return title; }

public void setTitle(String title) { this.title = title; }

public String getAuthor() { return author; }

public void setAuthor(String author) { this.author = author; }

}

**BookRepository.java**

package com.library.LibraryManagement.repository;

import com.library.LibraryManagement.entity.Book;

import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Long> {

}

**Output :**

