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

**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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>org.springframework.boot</groupId>

<artifactId>LibraryManagement</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter </artifactId>

<version>3.5.3</version>

</dependency>

</dependencies>

</project>

**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"/>

</beans>

**BookService.java:**

package com.library.service;

public class BookService {

public void printBookService() {

System.***out***.println("Book Service is ready.");

}

}

**BookRepository.java:**

package com.library.repository;

public class BookRepository {

public void printBookRepository() {

System.***out***.println("Book Repository is ready.");

}

}

**MainApp.java:**

package com.library;

import com.library.service.BookService;

import com.library.repository.BookRepository;

import org.springframework.context.ApplicationContext;

import org.springframework.context.support.ClassPathXmlApplicationContext;

public class MainApp {

public static void main(String[] args) {

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

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

BookRepository bookRepository=(BookRepository)applicationContext.getBean("bookRepository");

bookService.printBookService();

bookRepository.printBookRepository();

}

}

**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.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=new BookRepository();

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository=bookRepository;

}

public void addBook(String name) {

System.***out***.println("Adding Book "+name);

bookRepository.saveBook(name);

}

}

**BookRepository.java:**

package com.library.repository;

public class BookRepository {

public void saveBook(String name) {

System.***out***.println("Book "+name+" saved");

}

}

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

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

bookService.addBook("Harry Potter");

}

}

**Exercise 3: Implementing Logging with Spring AOP**

**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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>org.springframework.boot</groupId>

<artifactId>LibraryManagement</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter </artifactId>

<version>3.5.3</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>5.3.34</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

</dependencies>

</project>

**LoggingAspect:**

package com.library.aspect;

import org.aspectj.lang.ProceedingJoinPoint;

import org.aspectj.lang.annotation.Around;

import org.aspectj.lang.annotation.Aspect;

import org.springframework.stereotype.Component;

*@Aspect*

*@Component*

public class LoggingAspect {

*@Around*("execution(public void com.library.service.\*.\*(..))")

public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable{

long start=System.*currentTimeMillis*();

Object result=joinPoint.proceed();

long end=System.*currentTimeMillis*();

System.***out***.println("Execution of "+joinPoint.getSignature()+" took "+(end-start)+"ms");

return result;

}

}

**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"

xmlns:aop="http://www.springframework.org/schema/aop"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd">

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

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

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

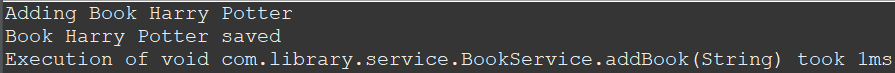
</bean>

<aop:aspectj-autoproxy/>

<bean id="loggingAspect" class="com.library.aspect.LoggingAspect"/>

</beans>

**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 https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>org.springframework.boot</groupId>

<artifactId>LibraryManagement</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter</artifactId>

<version>3.5.3</version>

</dependency>

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjweaver</artifactId>

<version>1.9.21</version>

</dependency>

<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>

</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="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=new BookRepository();

// Setter Method

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository=bookRepository;

}

public void addBook(String name) {

System.***out***.println("Adding Book "+name);

bookRepository.saveBook(name);

}

}

**BookRepository.java:**

package com.library.repository;

public class BookRepository {

public void saveBook(String name) {

System.***out***.println("Book "+name+" saved");

}

}

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

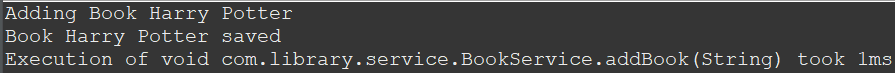
BookService bookService=(BookService)applicationContext.getBean("bookService");

bookService.addBook("Harry Potter");

}

}

**Output:**



**Exercise 6: Configuring Beans with Annotations**

**applicationContext.xml:**

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context

http://www.springframework.org/schema/context/spring-context.xsd">

<context:component-scan base-package="com.library" />

</beans>

**BookService.java:**

package com.library.service;

import org.springframework.stereotype.Service;

import com.library.repository.BookRepository;

*@Service*

public class BookService {

private BookRepository bookRepository=new BookRepository();

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository=bookRepository;

}

public void addBook(String name) {

System.***out***.println("Adding Book "+name);

bookRepository.saveBook(name);

}

}

**BookRepository.java:**

package com.library.repository;

import org.springframework.stereotype.Repository;

*@Repository*

public class BookRepository {

public void saveBook(String name) {

System.***out***.println("Book "+name+" saved");

}

}

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

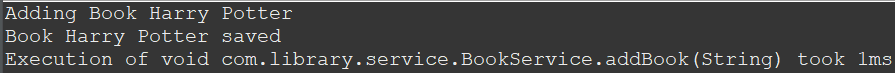
BookService bookService=(BookService)applicationContext.getBean("bookService");

bookService.addBook("Harry Potter");

}

}

**Output:**



**Exercise 7: Implementing Constructor and Setter 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.xsd">

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

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

<constructor-arg value="Central Library" />

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

</bean>

</beans>

**BookService:**

package com.library.service;

import com.library.repository.BookRepository;

public class BookService {

private String libraryName;

private BookRepository bookRepository;

public BookService(String libraryName) {

this.libraryName = libraryName;

}

public void setBookRepository(BookRepository bookRepository) {

this.bookRepository = bookRepository;

}

public void addBook(String name) {

System.***out***.println("Library: " + libraryName);

bookRepository.saveBook(name);

}

}

**BookRepository.java**

package com.library.repository;

public class BookRepository {

public void saveBook(String name) {

System.***out***.println("Saving Book "+name);

}

}

**Output:**



**Exercise 8: Implementing Basic AOP with Spring**

**LoggingAspect.java**

package com.library.aspect;

import org.aspectj.lang.annotation.Before;

import org.aspectj.lang.annotation.After;

import org.aspectj.lang.annotation.Aspect;

import org.springframework.stereotype.Component;

*@Aspect*

*@Component*

public class LoggingAspect {

*@Before*("execution(public void com.library.service.\*.\*(..))")

public void logBefore() {

System.***out***.println(">>> [LoggingAspect] Method is about to execute.");

}

*@After*("execution(public void com.library.service.\*.\*(..))")

public void logAfter() {

System.***out***.println("<<< [LoggingAspect] Method execution completed.");

}

}

**applicationContext.java:**

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="

http://www.springframework.org/schema/beans

https://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/aop

http://www.springframework.org/schema/aop/spring-aop.xsd

http://www.springframework.org/schema/context

https://www.springframework.org/schema/context/spring-context.xsd">

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

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

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

</bean>

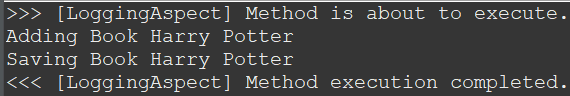
<context:component-scan base-package="com.library.aspect"/>

<aop:aspectj-autoproxy/>

<bean id="loggingAspect" class="com.library.aspect.LoggingAspect"/>

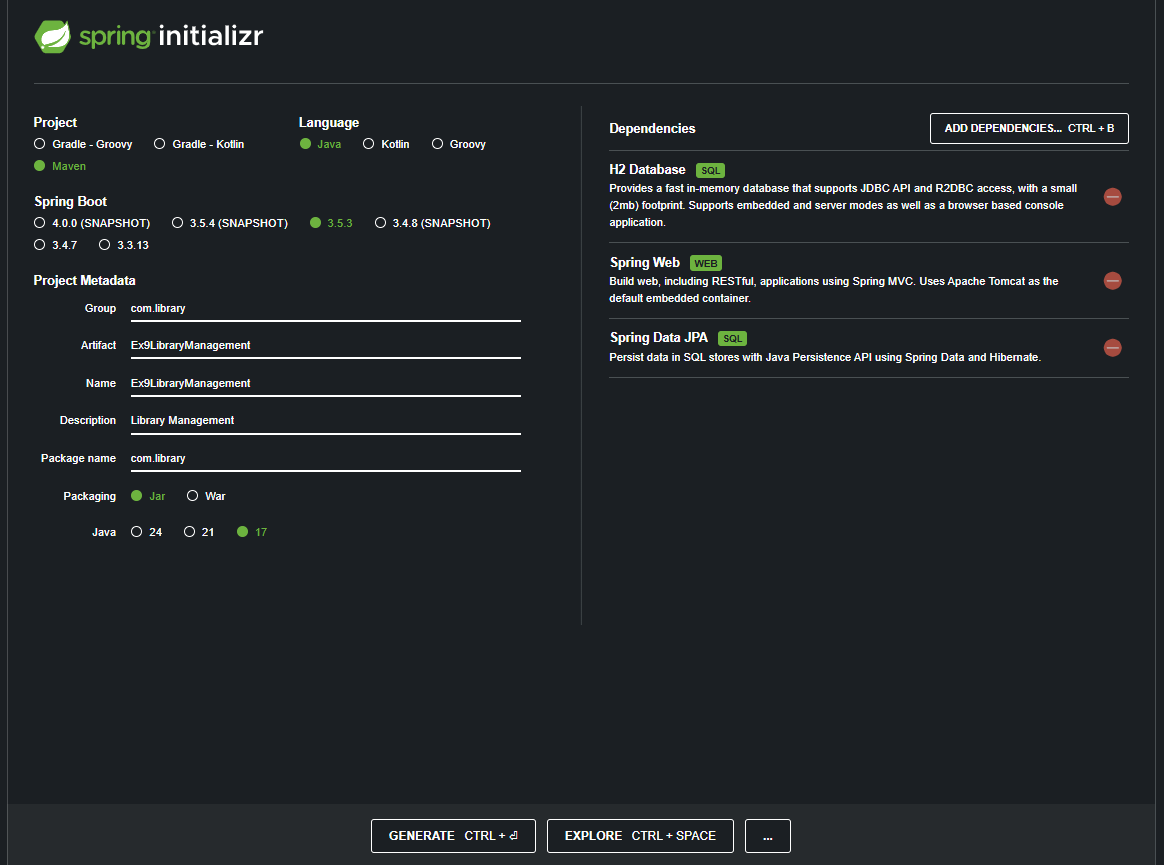
</beans>

**Output:**



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

**Spring Initializr:**



**Application.properties:**

spring.application.name=Ex9LibraryManagement

# H2 Database Configuration

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

spring.datasource.driverClassName=org.h2.Driver

spring.datasource.username=sa

spring.datasource.password=

# JPA Settings

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

spring.jpa.hibernate.ddl-auto=update

# H2 Console

spring.h2.console.enabled=true

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

**Book.java:**

package com.library.entity;

import jakarta.persistence.\*;

*@Entity*

*@Table*(name = "books")

public class Book {

*@Id*

*@GeneratedValue*(strategy = *GenerationType*.***IDENTITY***)

private Long id;

private String title;

private String author;

private int yearPublished;

public Book() {

}

public Book(String title, String author, int yearPublished) {

this.title = title;

this.author = author;

this.yearPublished = yearPublished;

}

public Long getId() {

return id;

}

public void setId(Long id) {

this.id = 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;

}

public int getYearPublished() {

return yearPublished;

}

public void setYearPublished(int yearPublished) {

this.yearPublished = yearPublished;

}

*@Override*

public String toString() {

return "Book{" +

"id=" + id +

", title='" + title + '\'' +

", author='" + author + '\'' +

", yearPublished=" + yearPublished +

'}';

}

}

**BookRepository.java:**

package com.library.repository;

import com.library.entity.Book;

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

import org.springframework.stereotype.Repository;

*@Repository*

public interface BookRepository extends JpaRepository<Book, Long> {

}

**BookController.java**

package com.library.controller;

import com.library.entity.Book;

import com.library.repository.BookRepository;

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

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

import java.util.List;

import java.util.Optional;

*@RestController*

*@RequestMapping*("/api/books")

public class BookController {

*@Autowired*

private BookRepository bookRepository;

// Get all books

*@GetMapping*

public List<Book> getAllBooks() {

return bookRepository.findAll();

}

// Get book by ID

*@GetMapping*("/{id}")

public Book getBookById(*@PathVariable* Long id) {

return bookRepository.findById(id).orElse(null);

}

// Add a new book

*@PostMapping*

public Book createBook(*@RequestBody* Book book) {

return bookRepository.save(book);

}

// Update a book

*@PutMapping*("/{id}")

public Book updateBook(*@PathVariable* Long id, *@RequestBody* Book updatedBook) {

Optional<Book> optionalBook = bookRepository.findById(id);

if (optionalBook.isPresent()) {

Book book = optionalBook.get();

book.setTitle(updatedBook.getTitle());

book.setAuthor(updatedBook.getAuthor());

book.setYearPublished(updatedBook.getYearPublished());

return bookRepository.save(book);

} else {

return null;

}

}

// Delete a book

*@DeleteMapping*("/{id}")

public void deleteBook(*@PathVariable* Long id) {

bookRepository.deleteById(id);

}

}

**Ex9LibraryManagementApplication.java:**

package com.library;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

*@SpringBootApplication*

public class Ex9LibraryManagementApplication {

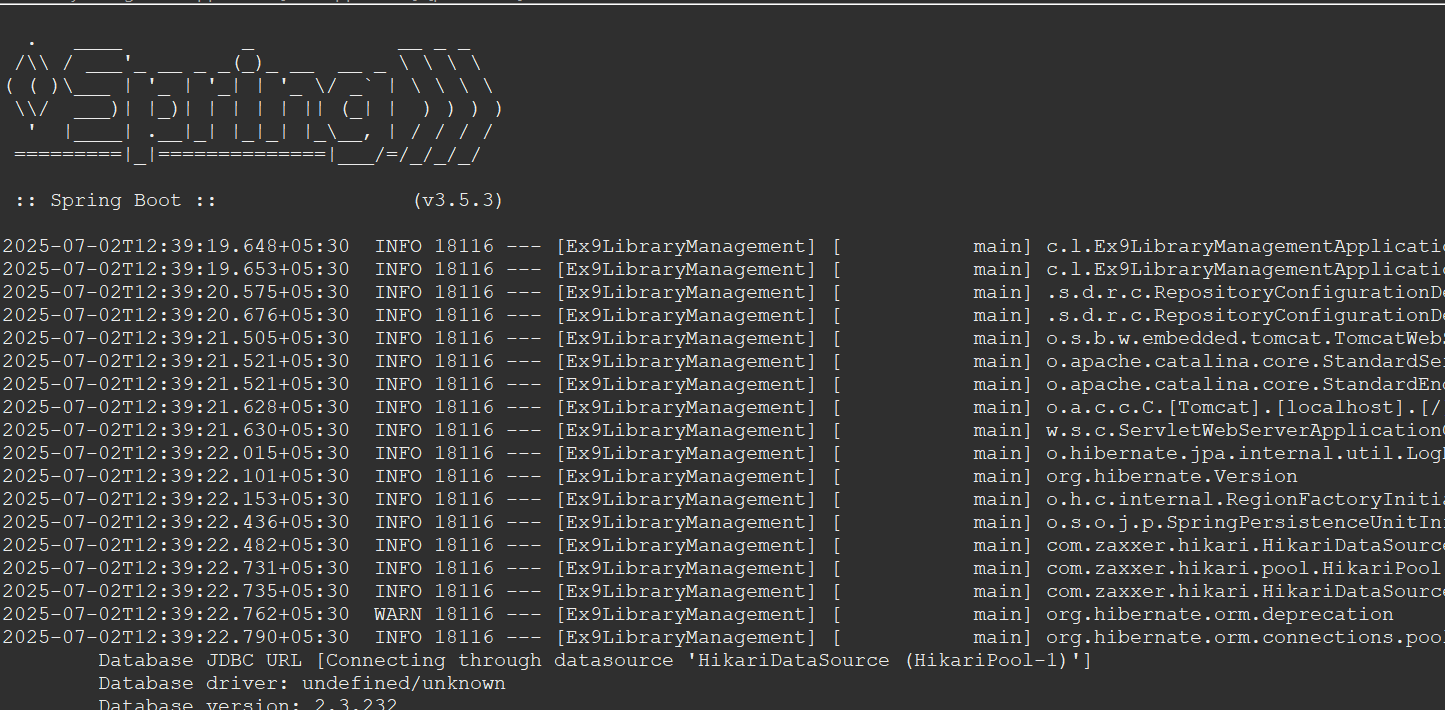
public static void main(String[] args) {

SpringApplication.*run*(Ex9LibraryManagementApplication.class, args);

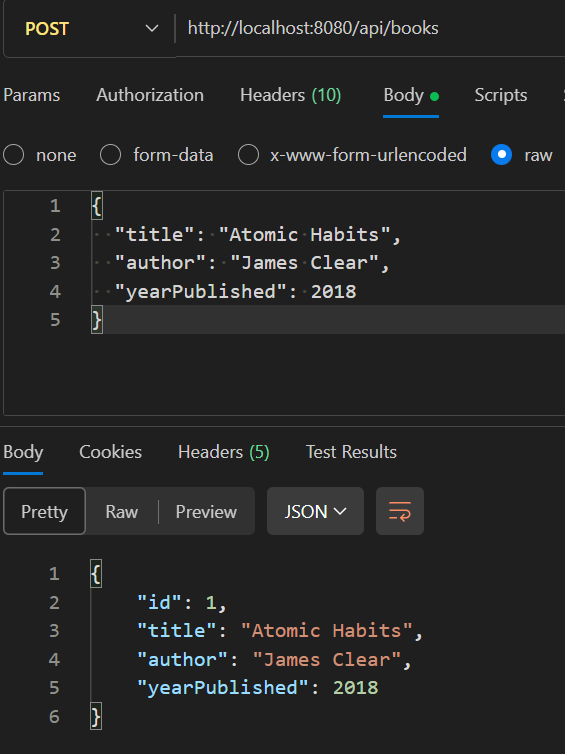
}

}

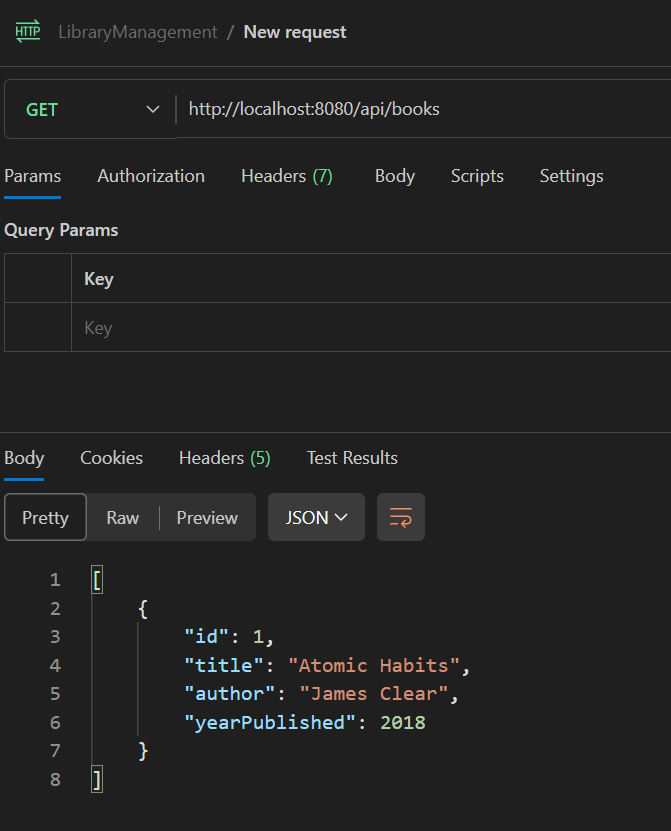
**Output for Main Application:**



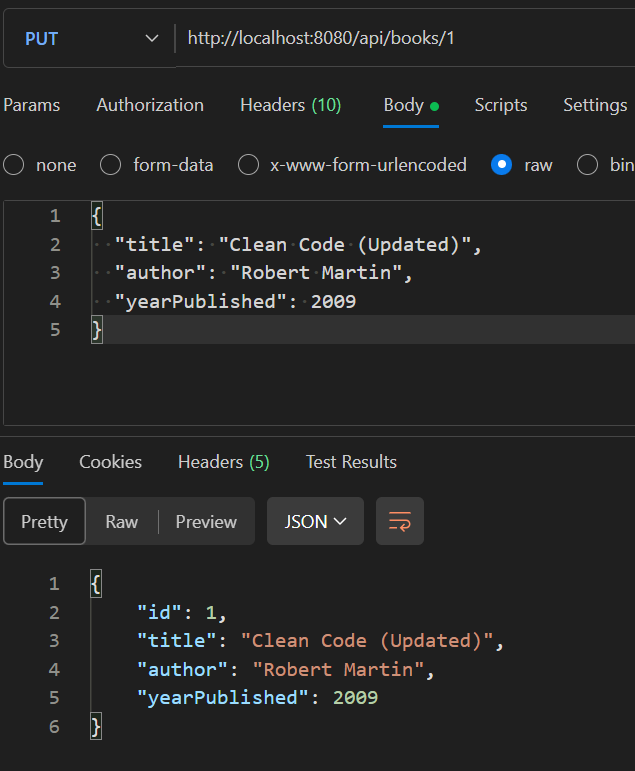
**POST Request:**



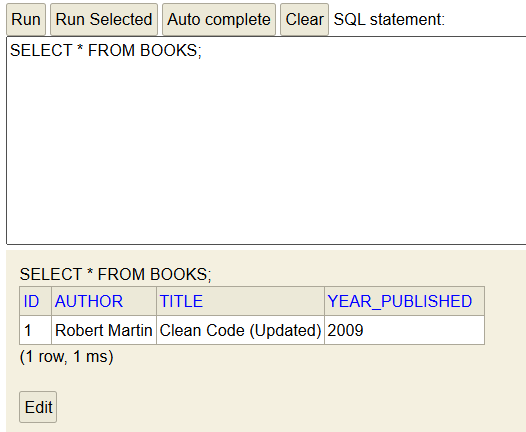
**GET Request(Also check for POST Request):**



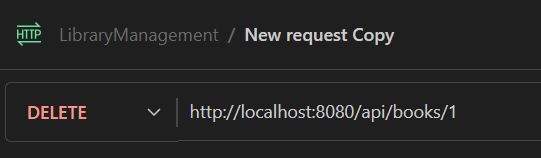
**PUT Request:**



**Checking in Database:**



**DELETE Request:**



**Checking in Database:**

