SuperSet ID : 6314413　　　　　　**Shaik Balaji Mahammad Rafi**

HANDSON EXERCISES - WEEK 3

Skill : Spring Core and Maven

Exercise 1 : Configuring a Basic Spring Application

CODE :

src/main/resources : applicationContent.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>
```

com.library.repository : BookRepository.java -

```
package com.library.repository;

public class BookRepository {
    public String getBookInfo() {
        return "Spring in Action by Craig Walls";
    }
}
```

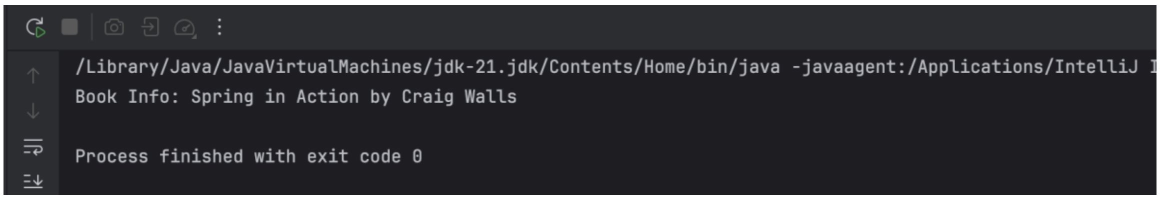
com.library.service : BookService.java -

```
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 displayBook() {
        System.out.println("Book Info: " + bookRepository.getBookInfo());
    }
}
```

src/main/java : Main.java -

```
import com.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 = context.getBean("bookService", BookService.class);
        bookService.displayBook();
    }
}
```

OUTPUT :



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Applications/IntelliJ I
Book Info: Spring in Action by Craig Walls

Process finished with exit code 0
```

Exercise 2 : Implementing Dependency Injection

CODE :

src/main/resources : applicationContent.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>
```

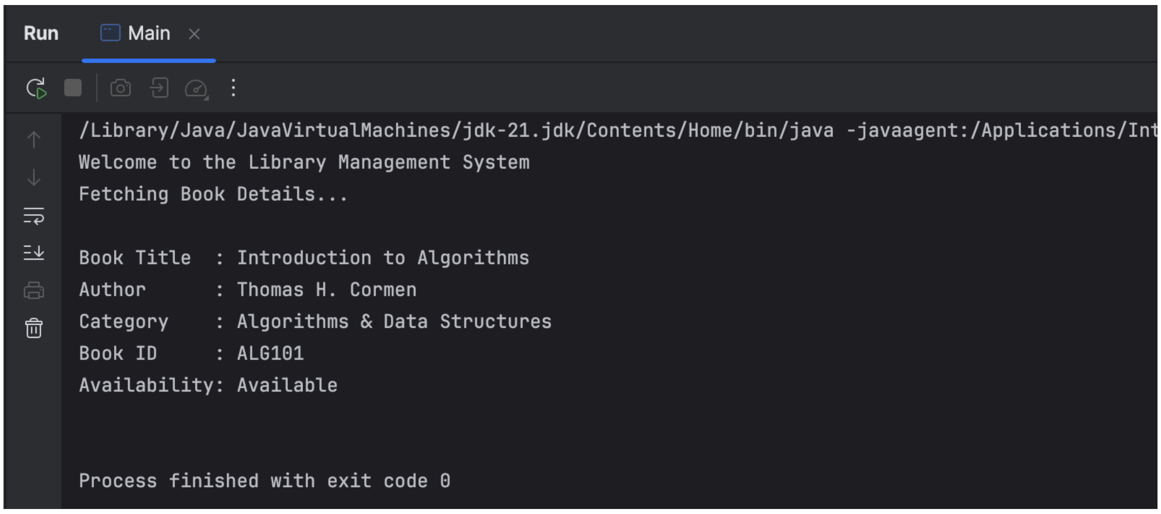
BookService.java :

```
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 displayBook() {
        System.out.println("Welcome to the Library Management System");
        System.out.println("Fetching Book Details...\n");
        System.out.println(bookRepository.getBookInfo());
    }
}
```

BookRepository.java :

```
package com.library.repository;

public class BookRepository {
    public String getBookInfo() {
        return """
                Book Title  : Introduction to Algorithms
                Author      : Thomas H. Cormen
                Category    : Algorithms & Data Structures
                Book ID     : ALG101
                Availability: Available
                """;
    }
}
```

OUTPUT :



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Applications/Int
Welcome to the Library Management System
Fetching Book Details...

Book Title  : Introduction to Algorithms
Author      : Thomas H. Cormen
Category    : Algorithms & Data Structures
Book ID     : ALG101
Availability: Available


Process finished with exit code 0
```

Exercise 3 : Implementing Logging with Spring AOP

CODE:

pom.xml :

```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>21</maven.compiler.source>
        <maven.compiler.target>21</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.3.38</version>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-aop</artifactId>
            <version>5.3.38</version>
        </dependency>

        <dependency>
            <groupId>org.aspectj</groupId>
            <artifactId>aspectjweaver</artifactId>
            <version>1.9.21</version>
        </dependency>

    </dependencies>

</project>
```

com.library.aspect : LoggingAspect.java :

```java
package com.library.aspect;

import org.aspectj.lang.ProceedingJoinPoint;
import org.aspectj.lang.annotation.Around;
import org.aspectj.lang.annotation.Aspect;

@Aspect
public class LoggingAspect {

    @Around("execution(* com.library.service.*.*(..))")
    public Object logExecutionTime(ProceedingJoinPoint joinPoint) throws Throwable {
        long start = System.currentTimeMillis();

        Object result = joinPoint.proceed(); // proceed with the original method

        long end = System.currentTimeMillis();
        System.out.println("Execution time of " + joinPoint.getSignature().getName() + ": "
+ (end - start) + " ms");

        return result;
    }
}
```

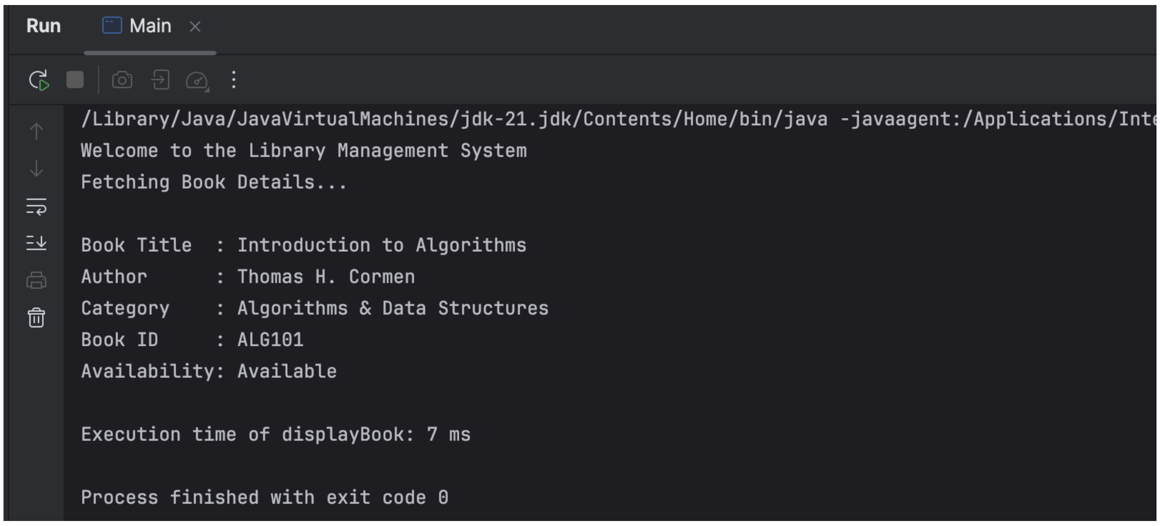
applicationContext.xml :

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

    <aop:aspectj-autoproxy />

    <bean id="bookRepository" class="com.library.repository.BookRepository"/>
    <bean id="bookService" class="com.library.service.BookService">
        <property name="bookRepository" ref="bookRepository"/>
    </bean>


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

Main.java :

```java
import com.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 = context.getBean("bookService", BookService.class);
      bookService.displayBook();
}
```

OUTPUT :



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Applications/Inte
Welcome to the Library Management System
Fetching Book Details...

Book Title  : Introduction to Algorithms
Author      : Thomas H. Cormen
Category    : Algorithms & Data Structures
Book ID     : ALG101
Availability: Available

Execution time of displayBook: 7 ms

Process finished with exit code 0
```

Exercise 4 : Creating and Configuring a Maven Project

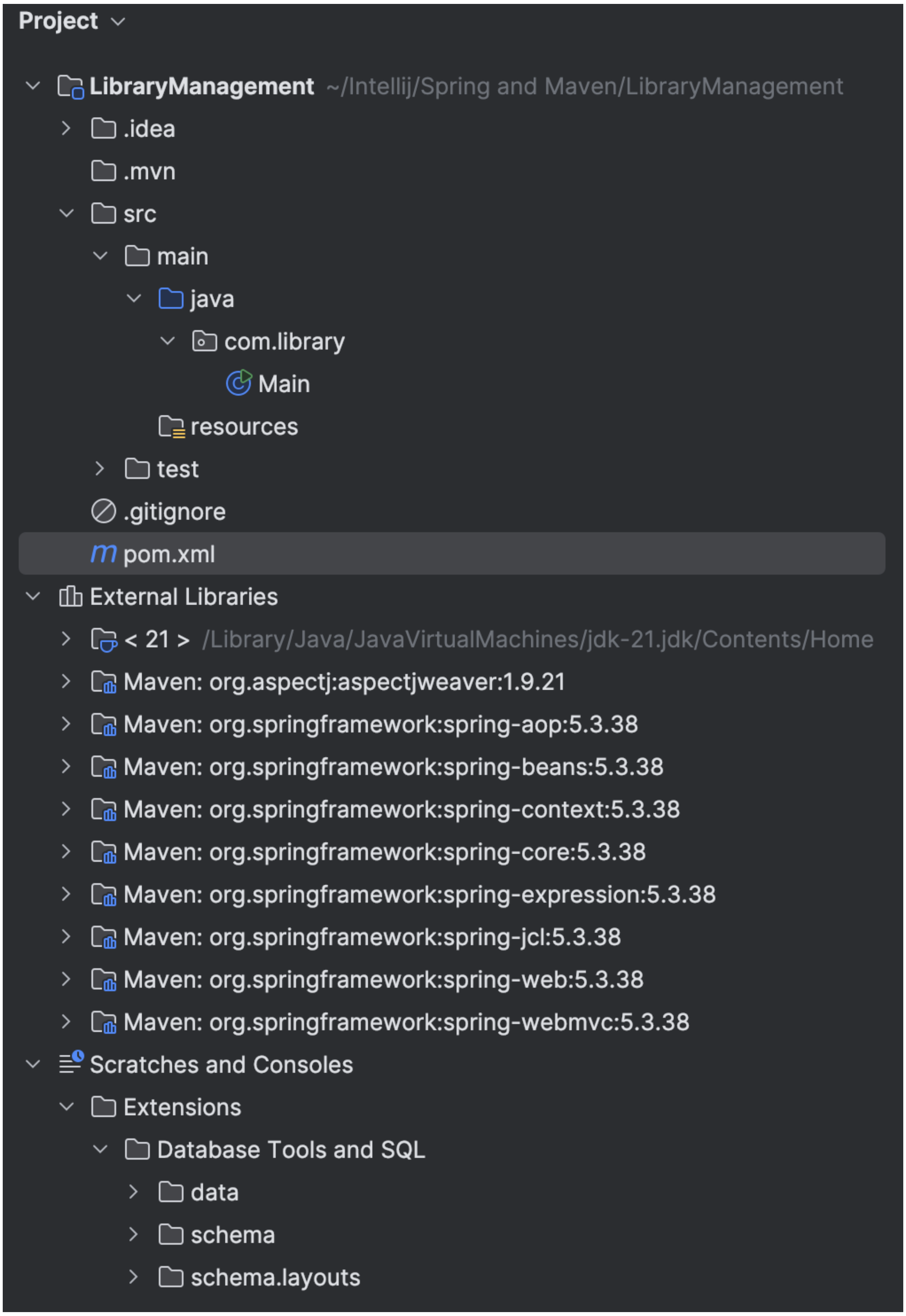
CODE :

pom.xml :

```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>
```

```
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>

    <dependencies>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.3.38</version>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-aop</artifactId>
            <version>5.3.38</version>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-webmvc</artifactId>
            <version>5.3.38</version>
        </dependency>

        <dependency>
            <groupId>org.aspectj</groupId>
            <artifactId>aspectjweaver</artifactId>
            <version>1.9.21</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.11.0</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
        </plugins>
    </build>
</project>
```

OUTPUT :



```
Project
LibraryManagement ~/Intellij/Spring and Maven/LibraryManagement
  .idea
  .mvn
  src
    main
      java
        com.library
          Main
      resources
    test
  .gitignore
  pom.xml
External Libraries
  < 21 > /Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home
  Maven: org.aspectj:aspectjweaver:1.9.21
  Maven: org.springframework:spring-aop:5.3.38
  Maven: org.springframework:spring-beans:5.3.38
  Maven: org.springframework:spring-context:5.3.38
  Maven: org.springframework:spring-core:5.3.38
  Maven: org.springframework:spring-expression:5.3.38
  Maven: org.springframework:spring-jcl:5.3.38
  Maven: org.springframework:spring-web:5.3.38
  Maven: org.springframework:spring-webmvc:5.3.38
Scratches and Consoles
  Extensions
    Database Tools and SQL
      data
      schema
      schema.layouts
```

Exercise 5 : Configuring the Spring IoC Container

CODE:

applicationContent.xml :

```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>
```

BookRepository.java :

```java
package com.library.repository;

public class BookRepository {
    public String getBookTitle() {
        return "Java: The Complete Reference by Herbert Schildt";
    }
}
```

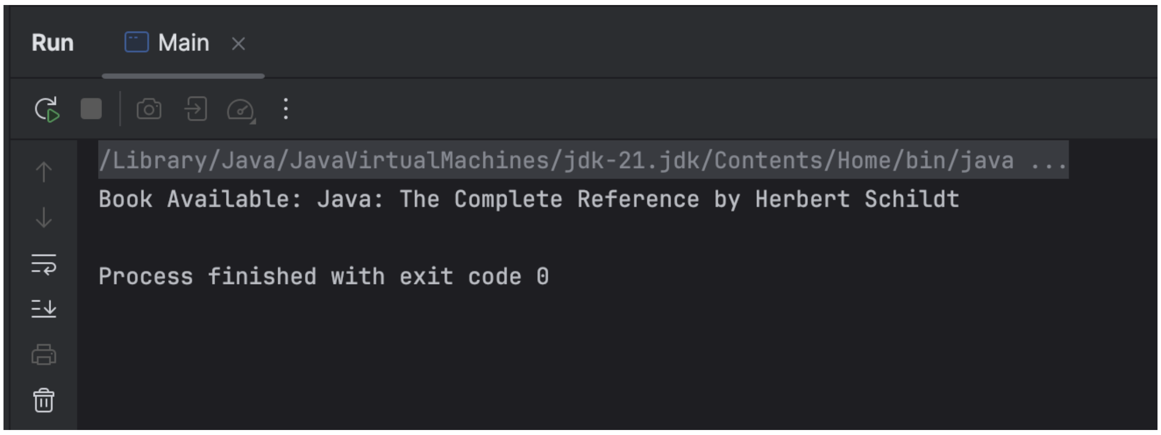
BookService.java :

```java
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;

    // setter for Spring to inject
    public void setBookRepository(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void displayBook() {
        System.out.println("Book Available: " + bookRepository.getBookTitle());
    }
}
```

Main.java :

```java
import com.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 = context.getBean("bookService", BookService.class);
        bookService.displayBook();
    }
}
```

OUTPUT :



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...
Book Available: Java: The Complete Reference by Herbert Schildt

Process finished with exit code 0
```

Exercise 6 : Configuring Beans with Annotations

CODE:

applicationContext.xml :

```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:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="
          http://www.springframework.org/schema/beans
          http://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>
```

BookRepository.java :

```
package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository
public class BookRepository {
    public String getBookTitle() {
        return "Java: The Complete Reference by Herbert Schildt";
    }
}
```

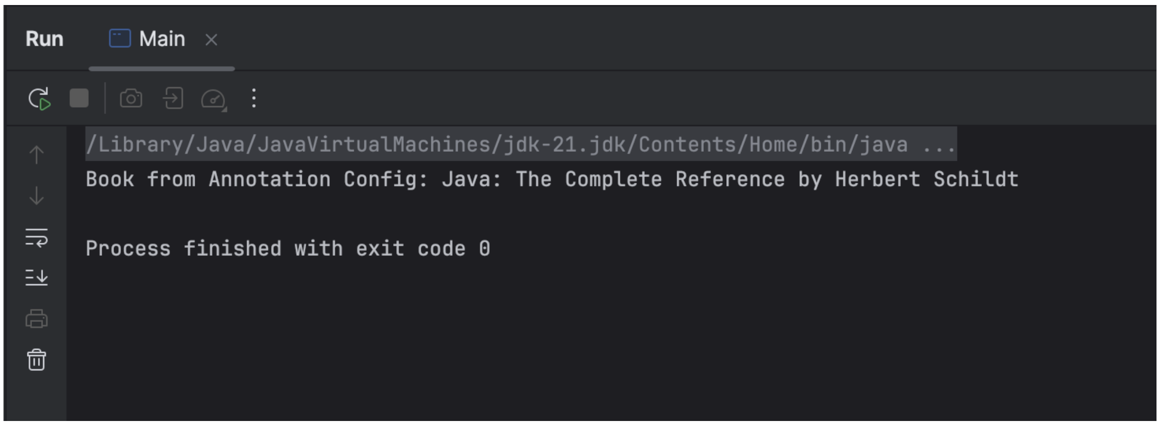
BookService.java :

```
package com.library.service;

import com.library.repository.BookRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class BookService {

    private BookRepository bookRepository;

    @Autowired
    public BookService(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void displayBook() {
        System.out.println("Book from Annotation Config: " +
bookRepository.getBookTitle());
    }
}
```

Main.java :

```
import com.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 = context.getBean(BookService.class);
        bookService.displayBook();
    }
}
```

OUTPUT :



Run Main

```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...
Book from Annotation Config: Java: The Complete Reference by Herbert Schildt

Process finished with exit code 0
```

Exercise 7 : Implementing Constructor and Setter Injection

CODE:

applicationContext.xml :

```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 ref="bookRepository"/>
        <property name="libraryName" value="Public Library"/>
    </bean>

</beans>
```

BookService.java :

```java
package com.library.service;

import com.library.repository.BookRepository;

public class BookService {
    private BookRepository bookRepository;
    private String libraryName;

    public BookService(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void setLibraryName(String libraryName) {
        this.libraryName = libraryName;
    }
    public void displayBook() {

        System.out.println("Library Name: " + libraryName);
        System.out.println("Book Title  : " + bookRepository.getBookTitle());
    }

}
```

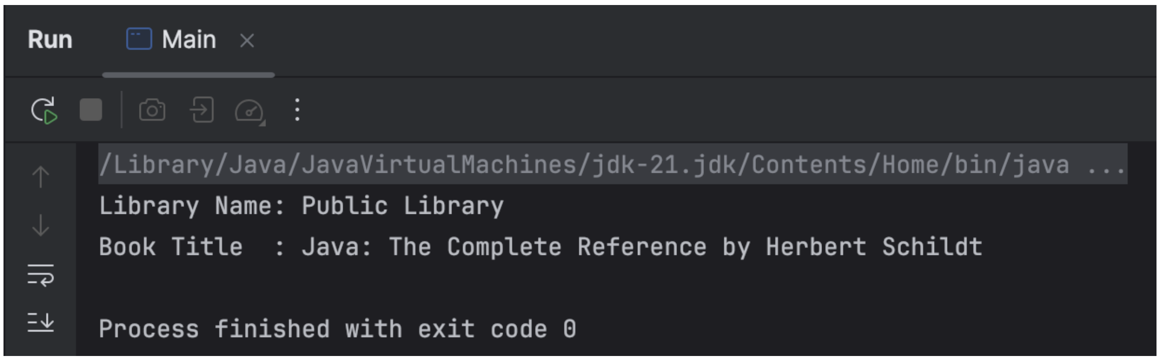
BookRepository.java :

```
package com.library.repository;

public class BookRepository {

    public String getBookTitle() {
        return "Java: The Complete Reference by Herbert Schildt";
    }

}
```

Main.java :

```
import com.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 = context.getBean("bookService", BookService.class);
        bookService.displayBook();
    }
}
```

OUTPUT:



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...
Library Name: Public Library
Book Title  : Java: The Complete Reference by Herbert Schildt

Process finished with exit code 0
```

Exercise 8 : Implementing Basic AOP with Spring

CODE:

com.library.aspect - LoggingAspect.java :

```java
package com.library.aspect;

import org.aspectj.lang.JoinPoint;
import org.aspectj.lang.annotation.After;
import org.aspectj.lang.annotation.Aspect;
import org.aspectj.lang.annotation.Before;

@Aspect
public class LoggingAspect {

    @Before("execution(* com.library.service.*.*(..))")
    public void beforeMethod(JoinPoint joinPoint) {
        System.out.println("[BEFORE] Executing: " + joinPoint.getSignature().getName());
    }

    @After("execution(* com.library.service.*.*(..))")
    public void afterMethod(JoinPoint joinPoint) {
        System.out.println("[AFTER] Completed: " + joinPoint.getSignature().getName());
    }
}
```

applicationContext.xml :

```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:context="http://www.springframework.org/schema/context"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xsi:schemaLocation="
           http://www.springframework.org/schema/beans
           http://www.springframework.org/schema/beans/spring-beans.xsd
           http://www.springframework.org/schema/context
           http://www.springframework.org/schema/context/spring-context.xsd
           http://www.springframework.org/schema/aop
           http://www.springframework.org/schema/aop/spring-aop.xsd">

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

    <aop:aspectj-autoproxy/>

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

</beans>
```

BookService.java :

```
package com.library.service;

import com.library.repository.BookRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Service
public class BookService {

    private BookRepository bookRepository;

    @Autowired
    public BookService(BookRepository bookRepository) {
        this.bookRepository = bookRepository;
    }

    public void displayBook() {
        System.out.println("Book: " + bookRepository.getBookTitle());
    }
}
```

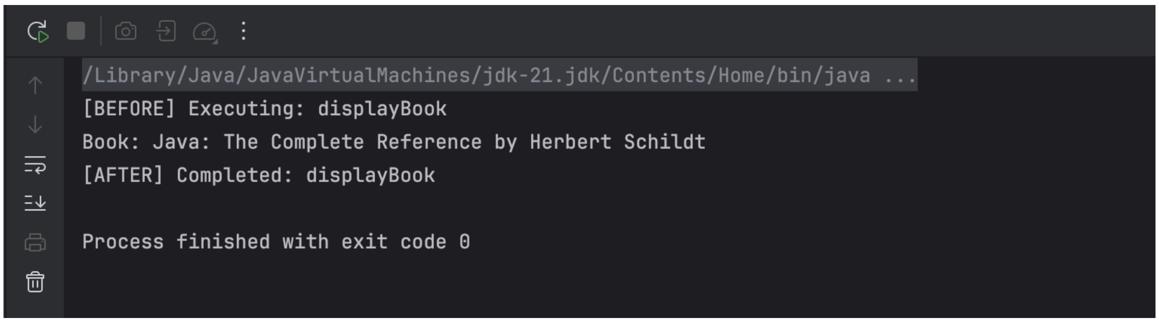
BookRepository.java :

```
package com.library.repository;

import org.springframework.stereotype.Repository;

@Repository
public class BookRepository {
    public String getBookTitle() {
        return "Java: The Complete Reference by Herbert Schildt";
    }
}
```

Main.java :

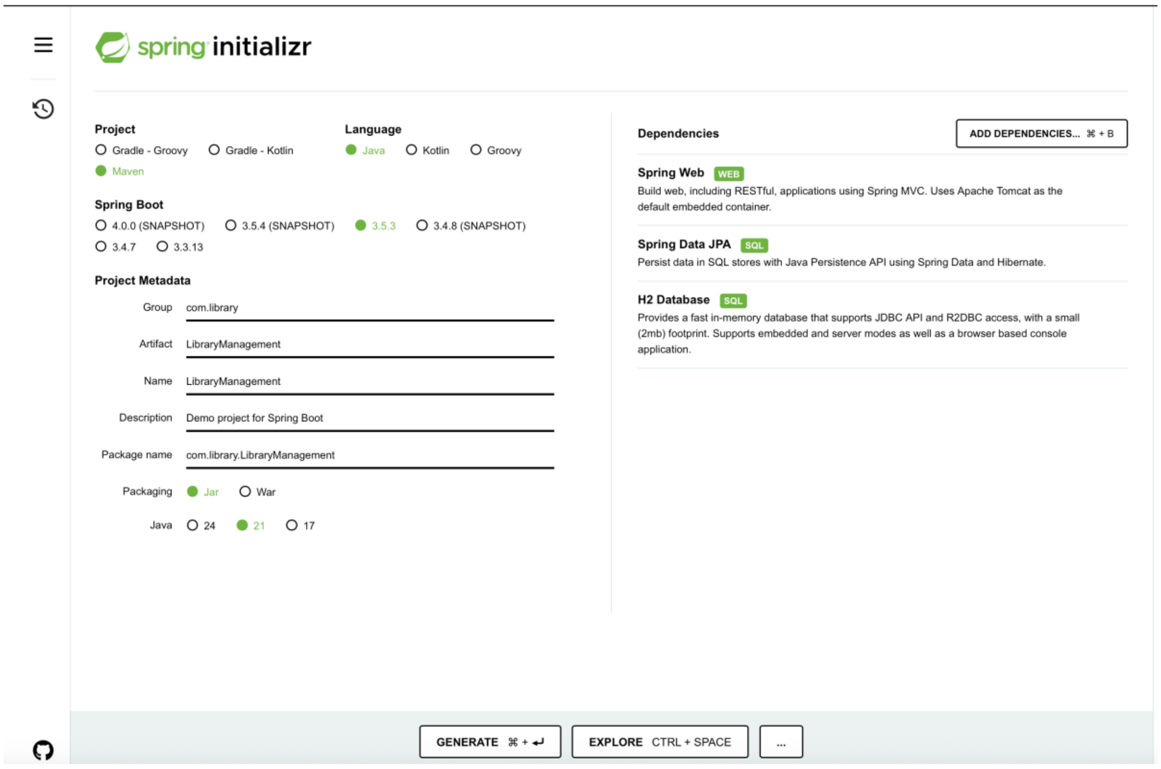
```
import com.library.service.BookService;
import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

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

        BookService bookService = context.getBean(BookService.class);
        bookService.displayBook();
    }
}
```

OUTPUT:



```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java ...
[BEFORE] Executing: displayBook
Book: Java: The Complete Reference by Herbert Schildt
[AFTER] Completed: displayBook

Process finished with exit code 0
```

Exercise 9 : Implementing Basic AOP with Spring



spring initializr

Project
Gradle - Groovy · Gradle - Kotlin · [✓ Maven]

Language
[✓ Java] · Kotlin · Groovy

Spring Boot
4.0.0 (SNAPSHOT) · 3.5.4 (SNAPSHOT) · [✓ 3.5.3] · 3.4.8 (SNAPSHOT) · 3.4.7 · 3.3.13

Project Metadata

Group	com.library
Artifact	LibraryManagement
Name	LibraryManagement
Description	Demo project for Spring Boot
Package name	com.library.LibraryManagement
Packaging	[✓ Jar] · War
Java	24 · [✓ 21] · 17

Dependencies

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

Spring Data JPA SQL
Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

H2 Database SQL
Provides a fast in-memory database that supports JDBC API and R2DBC access, with a small (2mb) footprint. Supports embedded and server modes as well as a browser based console application.

application.properties :

```
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
```

com.library.model : Book.java -

```java
package com.library.model;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.Id;

@Entity
public class Book {
    @Id
    @GeneratedValue
    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 String getAuthor() { return author; }

    public void setId(Long id) { this.id = id; }
    public void setTitle(String title) { this.title = title; }
    public void setAuthor(String author) { this.author = author; }
}
```

com.library.repository : BookRepository.java -

```java
package com.library.repository;

import com.library.model.Book;
import org.springframework.data.jpa.repository.JpaRepository;

public interface BookRepository extends JpaRepository<Book, Long> {
}
```

com.library.controller : BookController.java -

```java
package com.library.controller;

import com.library.model.Book;
import com.library.repository.BookRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;
```

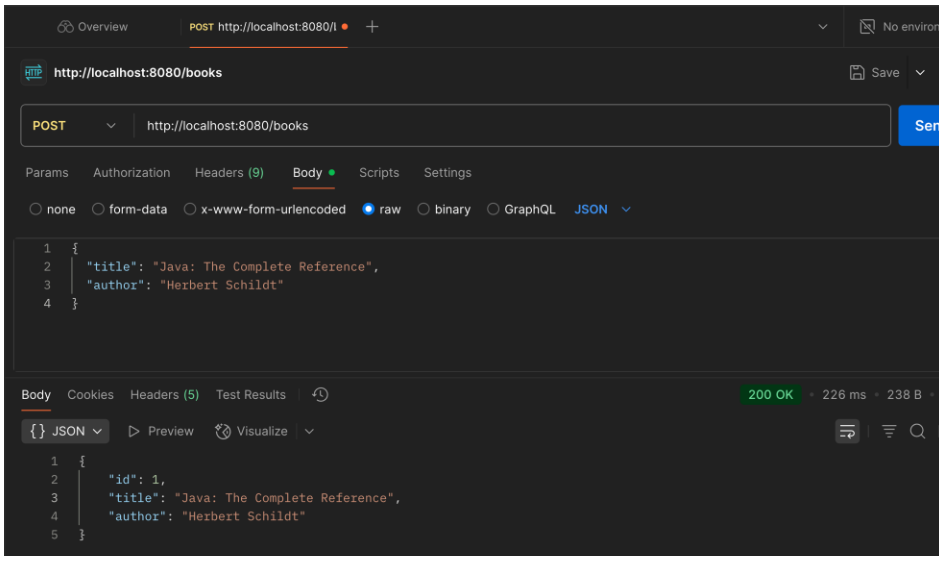
```
@RestController
@RequestMapping("/books")
public class BookController {

    @Autowired
    private BookRepository bookRepository;

    @PostMapping
    public Book addBook(@RequestBody Book book) {
        return bookRepository.save(book);
    }

    @PutMapping("/{id}")
    public Book updateBook(@PathVariable Long id, @RequestBody Book updatedBook)
{
        return bookRepository.findById(id)
                .map(book -> {
                    book.setTitle(updatedBook.getTitle());
                    book.setAuthor(updatedBook.getAuthor());
                    return bookRepository.save(book);
                })
                .orElseThrow(() -> new RuntimeException("Book not found with id: " + id));
    }

    @GetMapping
    public List<Book> getAllBooks() {
        return bookRepository.findAll();
    }

    @GetMapping("/{id}")
    public Book getBook(@PathVariable Long id) {
        return bookRepository.findById(id).orElse(null);
    }

    @DeleteMapping("/{id}")
    public void deleteBook(@PathVariable Long id) {
        bookRepository.deleteById(id);
    }
}
```

LibraryManagementApplication.java :

```
package com.library;

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);
    }
}
```

OUTPUT : (used Postman for testing)

POST Request :



http://localhost:8080/books

POST http://localhost:8080/books

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

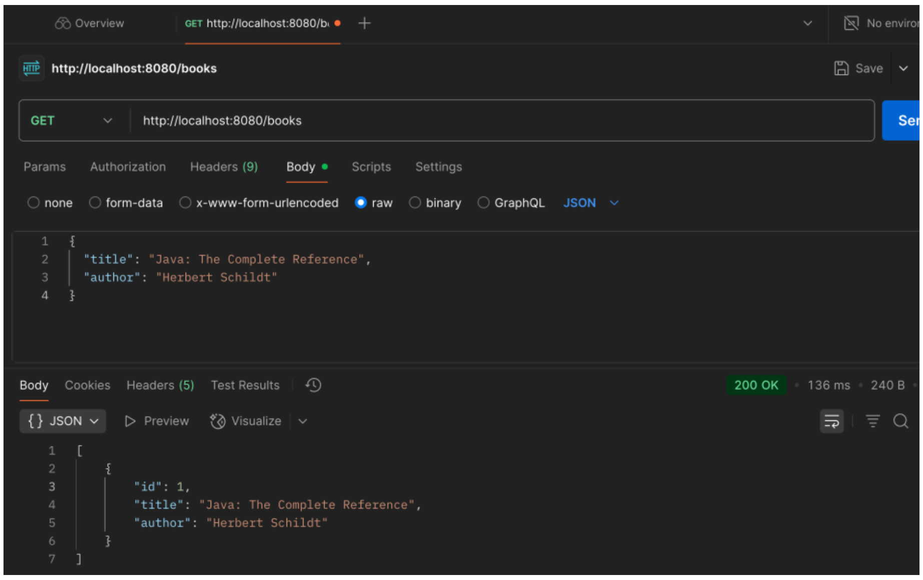
```
{
  "title": "Java: The Complete Reference",
  "author": "Herbert Schildt"
}
```

Body 200 OK 226 ms 238 B

{} JSON

```
{
    "id": 1,
    "title": "Java: The Complete Reference",
    "author": "Herbert Schildt"
}
```

GET All Request :



http://localhost:8080/books

GET http://localhost:8080/books

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

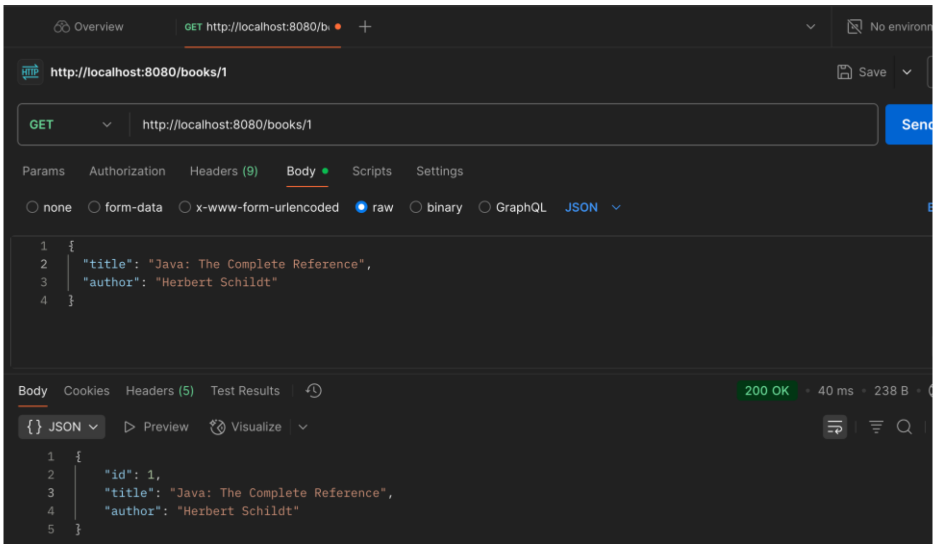
```
{
  "title": "Java: The Complete Reference",
  "author": "Herbert Schildt"
}
```

Body 200 OK 136 ms 240 B

{} JSON

```
[
    {
        "id": 1,
        "title": "Java: The Complete Reference",
        "author": "Herbert Schildt"
    }
]
```

GET by ID :



http://localhost:8080/books/1

GET http://localhost:8080/books/1

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

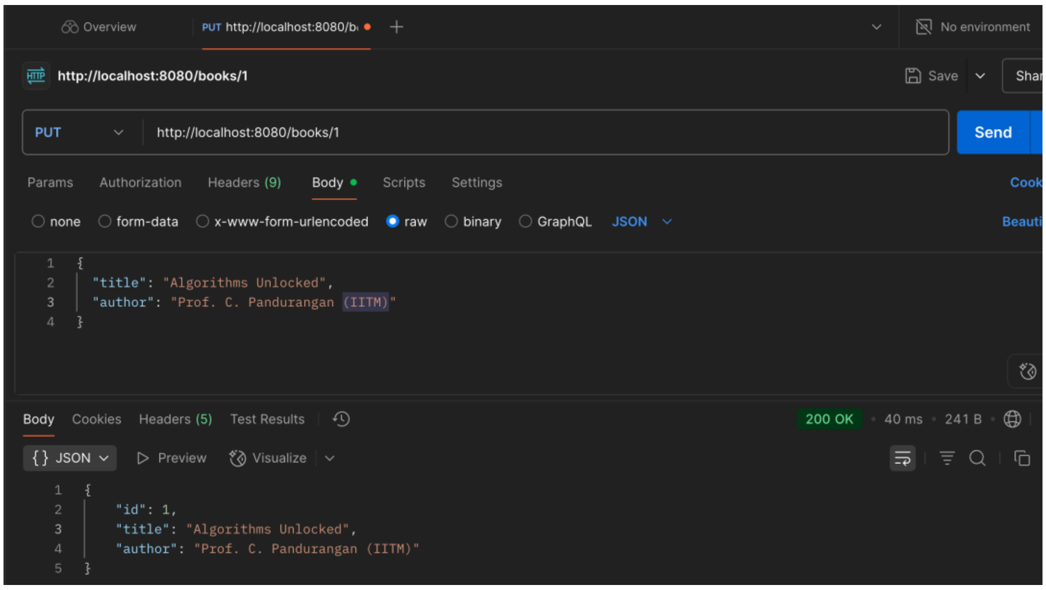
```
{
  "title": "Java: The Complete Reference",
  "author": "Herbert Schildt"
}
```

Body 200 OK 40 ms 238 B

{} JSON

```
{
    "id": 1,
    "title": "Java: The Complete Reference",
    "author": "Herbert Schildt"
}
```

PUT Request :



http://localhost:8080/books/1

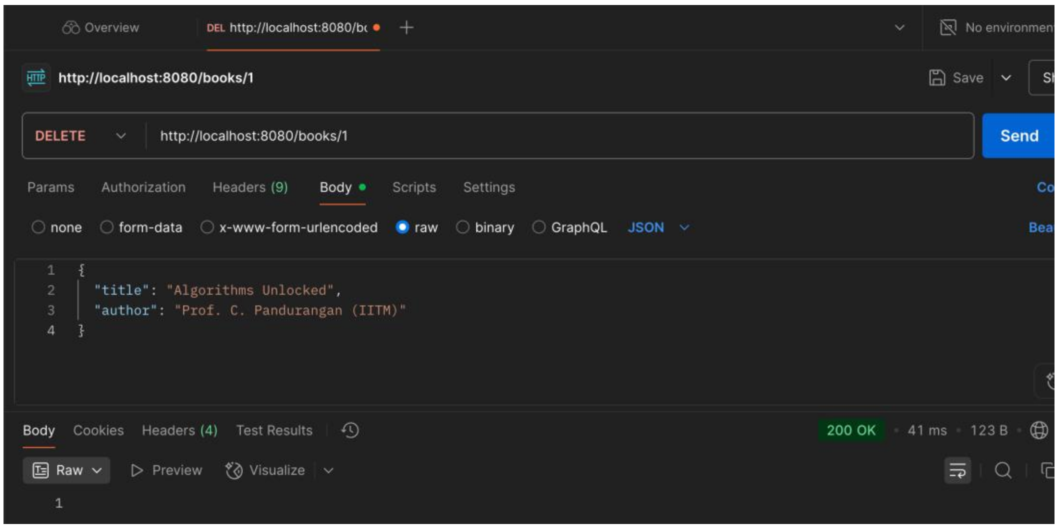
PUT http://localhost:8080/books/1

```
{
  "title": "Algorithms Unlocked",
  "author": "Prof. C. Pandurangan (IITM)"
}
```

200 OK · 40 ms · 241 B

```
{
    "id": 1,
    "title": "Algorithms Unlocked",
    "author": "Prof. C. Pandurangan (IITM)"
}
```

DELETE Request :



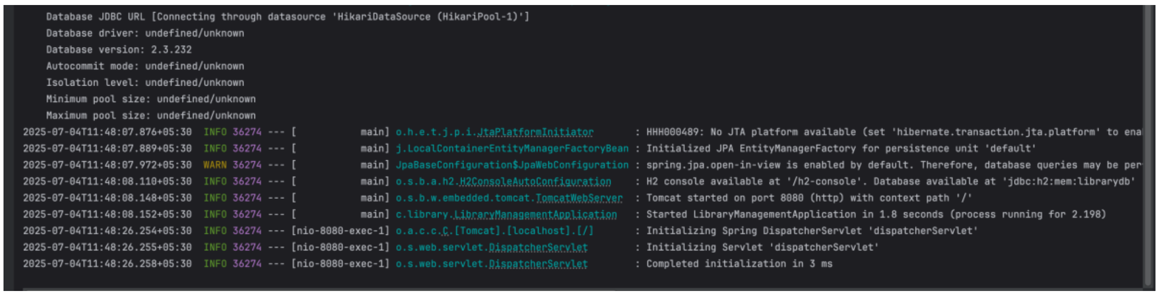
http://localhost:8080/books/1

DELETE http://localhost:8080/books/1

```
{
  "title": "Algorithms Unlocked",
  "author": "Prof. C. Pandurangan (IITM)"
}
```

200 OK · 41 ms · 123 B

Console Output in IntelliJ :



```
      Database JDBC URL [Connecting through datasource 'HikariDataSource (HikariPool-1)']
      Database driver: undefined/unknown
      Database version: 2.3.232
      Autocommit mode: undefined/unknown
      Isolation level: undefined/unknown
      Minimum pool size: undefined/unknown
      Maximum pool size: undefined/unknown
2025-07-04T11:48:07.876+05:30  INFO 36274 --- [           main] o.h.e.t.j.p.i.JtaPlatformInitiator       : HHH000489: No JTA platform available (set 'hibernate.transaction.jta.platform' to ena
2025-07-04T11:48:07.889+05:30  INFO 36274 --- [           main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2025-07-04T11:48:07.972+05:30  WARN 36274 --- [           main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be per
2025-07-04T11:48:08.110+05:30  INFO 36274 --- [           main] o.s.b.a.h2.H2ConsoleAutoConfiguration    : H2 console available at '/h2-console'. Database available at 'jdbc:h2:mem:librarydb'
2025-07-04T11:48:08.148+05:30  INFO 36274 --- [           main] o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port 8080 (http) with context path '/'
2025-07-04T11:48:08.152+05:30  INFO 36274 --- [           main] c.library.LibraryManagementApplication   : Started LibraryManagementApplication in 1.8 seconds (process running for 2.198)
2025-07-04T11:48:26.254+05:30  INFO 36274 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]       : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-07-04T11:48:26.255+05:30  INFO 36274 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Initializing Servlet 'dispatcherServlet'
2025-07-04T11:48:26.258+05:30  INFO 36274 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet        : Completed initialization in 3 ms
```