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

<modelVersion>4.0.0</modelVersion>

<parent>

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

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

<version>2.7.10</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<groupId>com.cdac</groupId>

<artifactId>SB-One2One-Mapping-Project</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>SB-One2One-Mapping-Project</name>

<description>Spring Boot with one to one mapping</description>

<properties>

<java.version>11</java.version>

</properties>

<dependencies>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

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

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

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

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

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

<scope>test</scope>

</dependency>

<dependency>

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

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

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

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

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

**Application.properties**

# JDBC PROPERTIES

spring.datasource.url=jdbc:mysql://localhost:3306/pune?useSSL=false&allowPublicKeyRetrieval=true

spring.datasource.username=root

spring.datasource.password=cdac

# HIBERNATE PROPERTIES

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.jpa.hibernate.ddl-auto=update

## Show sql query.

spring.jpa.show-sql=true

**Book.java**

package com.cdac.model;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToOne;

import javax.persistence.Table;

@Entity

@Table(name = "Table\_Book1")

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String title;

@OneToOne(cascade = CascadeType.ALL)

private Author author;

public Book() {

}

public Book(int id) {

super();

this.id = id;

}

public Book(int id, String title, Author author) {

super();

this.id = id;

this.title = title;

this.author = author;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public Author getAuthor() {

return author;

}

public void setAuthor(Author author) {

this.author = author;

}

@Override

public String toString() {

return "Book [id=" + id + ", title=" + title + ", author=" + author + "]";

}

}

**Author.java**

package com.cdac.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name = "Table\_Author1")

public class Author {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String first\_name;

private String last\_name;

private String Address;

public Author() {

}

public Author(int id) {

super();

this.id = id;

}

public Author(int id, String first\_name, String last\_name, String address) {

super();

this.id = id;

this.first\_name = first\_name;

this.last\_name = last\_name;

Address = address;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getFirst\_name() {

return first\_name;

}

public void setFirst\_name(String first\_name) {

this.first\_name = first\_name;

}

public String getLast\_name() {

return last\_name;

}

public void setLast\_name(String last\_name) {

this.last\_name = last\_name;

}

public String getAddress() {

return Address;

}

public void setAddress(String address) {

Address = address;

}

@Override

public String toString() {

return "Author [id=" + id + ", first\_name=" + first\_name + ", last\_name=" + last\_name + ", Address=" + Address+ "]";

}

}

**BookDao.java**

package com.cdac.dao;

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

import org.springframework.stereotype.Repository;

import com.cdac.model.Book;

@Repository

public interface BookDao extends JpaRepository<Book, Integer>{

}

**BookService.java**

package com.cdac.service;

import java.util.List;

import com.cdac.model.Book;

public interface BookService {

void add\_Book(Book book);

void modify\_book(Book book);

void remove\_Book\_By\_id(int id);

Book get\_Book\_By\_Id(int id);

List<Book> get\_All\_Books();

}

**BookServiceImpl.java**

package com.cdac.service;

import java.util.ArrayList;

import java.util.List;

import java.util.Optional;

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

import org.springframework.stereotype.Service;

import com.cdac.dao.BookDao;

import com.cdac.model.Book;

@Service

public class BookServiceImpl implements BookService {

@Autowired

private BookDao bookDao;

@Override

public void add\_Book(Book book) {

bookDao.save(book);

}

@Override

public void modify\_book(Book book) {

bookDao.save(book);

}

@Override

public void remove\_Book\_By\_id(int id) {

bookDao.deleteById(id);

}

@Override

public Book get\_Book\_By\_Id(int id) {

Optional<Book> opt = bookDao.findById(id);

if (opt.isPresent()) {

return opt.get();

}

return null;

}

@Override

public List<Book> get\_All\_Books() {

Iterable<Book> itr = bookDao.findAll();

List<Book> lst = new ArrayList<Book>();

itr.forEach(ele -> lst.add(ele));

return lst;

}

}

**BookController.java**

package com.cdac.controller;

import java.util.List;

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

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import com.cdac.model.Book;

import com.cdac.service.BookService;

@RestController

public class BookController {

@Autowired

private BookService bookService;

@PostMapping(value = { "/book" })

public String add\_Book(@RequestBody Book book) {

bookService.add\_Book(book);

return "Book added successfully";

}

@PutMapping(value = { "/book" })

public String update\_Book(@RequestBody Book book) {

bookService.modify\_book(book);

return "Book updated successfully";

}

@DeleteMapping(value = { "/book" })

public String delete\_Book(@RequestParam int id) {

System.out.println("first delete");

bookService.remove\_Book\_By\_id(id);

return "Book deleted successfully";

}

@DeleteMapping(value = { "/book/{id}" })

public String delete\_Book1(@PathVariable int id) {

System.out.println("second delete");

bookService.remove\_Book\_By\_id(id);

return "Book deleted successfully";

}

@GetMapping(value = { "/book/{id}" })

public Book get\_Book\_By\_Id(@PathVariable int id) {

return bookService.get\_Book\_By\_Id(id);

}

@GetMapping(value = { "/book" })

public List<Book> get\_All\_Books() {

return bookService.get\_All\_Books();

}

}

**SBOne2OneMappingProject.java**

package com.cdac;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SBOne2OneMappingProjectApplication {

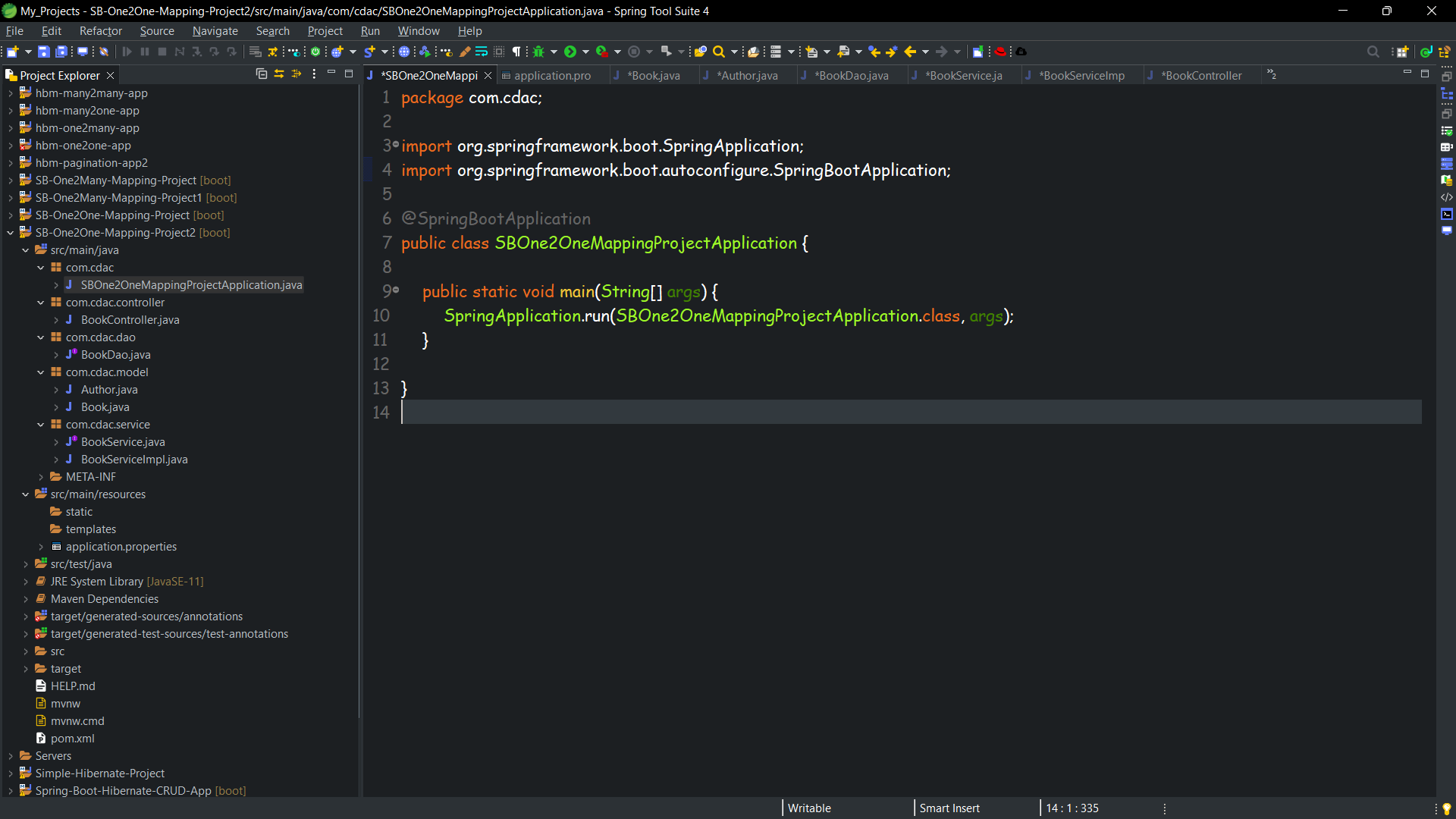
public static void main(String[] args) {

SpringApplication.run(SBOne2OneMappingProjectApplication.class, args);

}

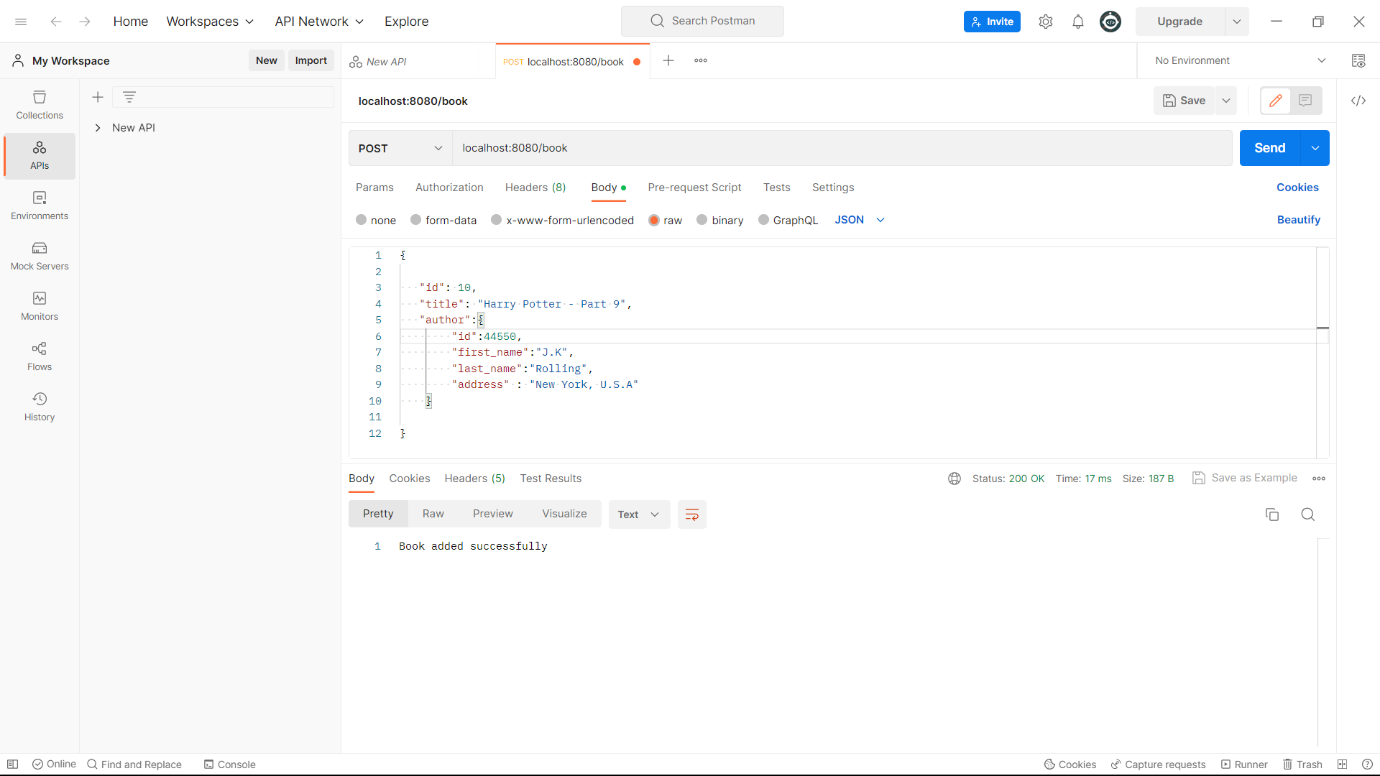
}

**Project Structrure**

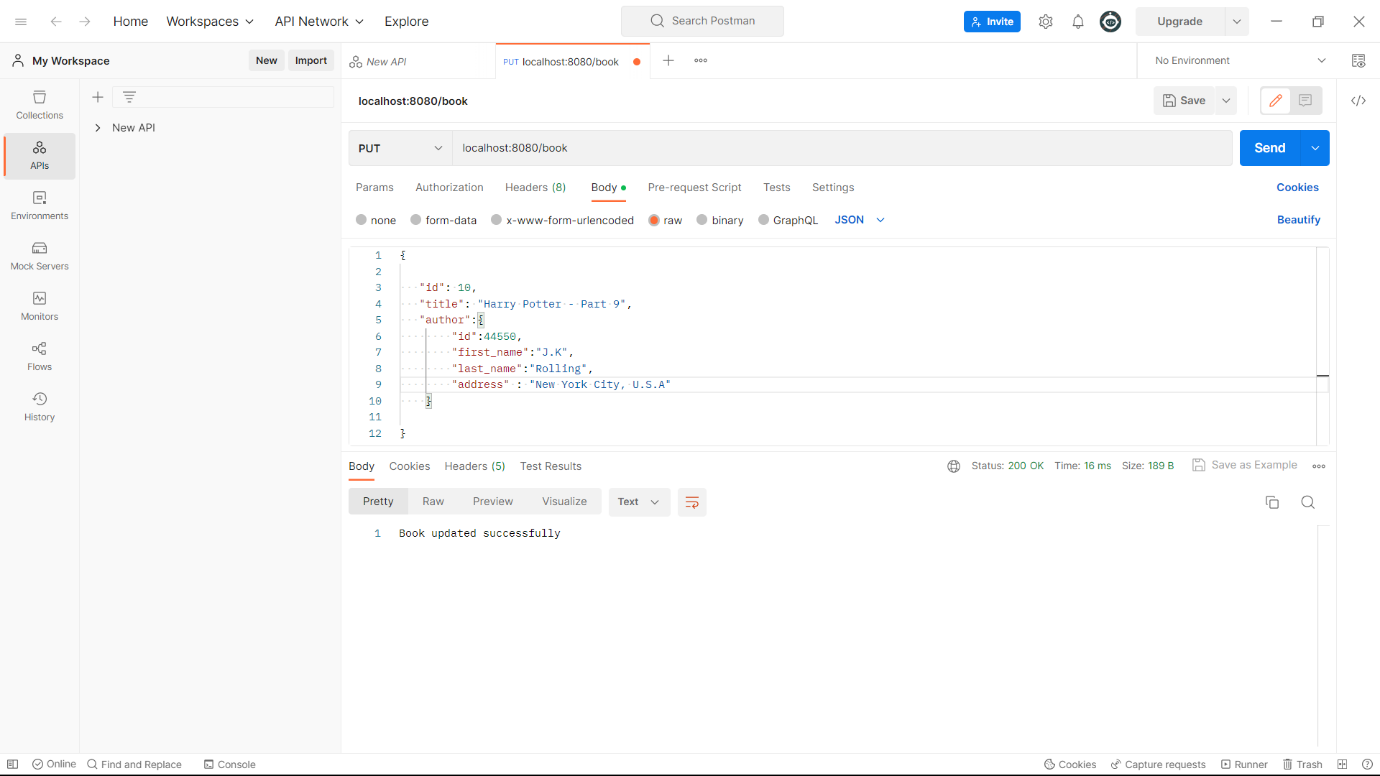


**PostMan and Database ScreenShots**

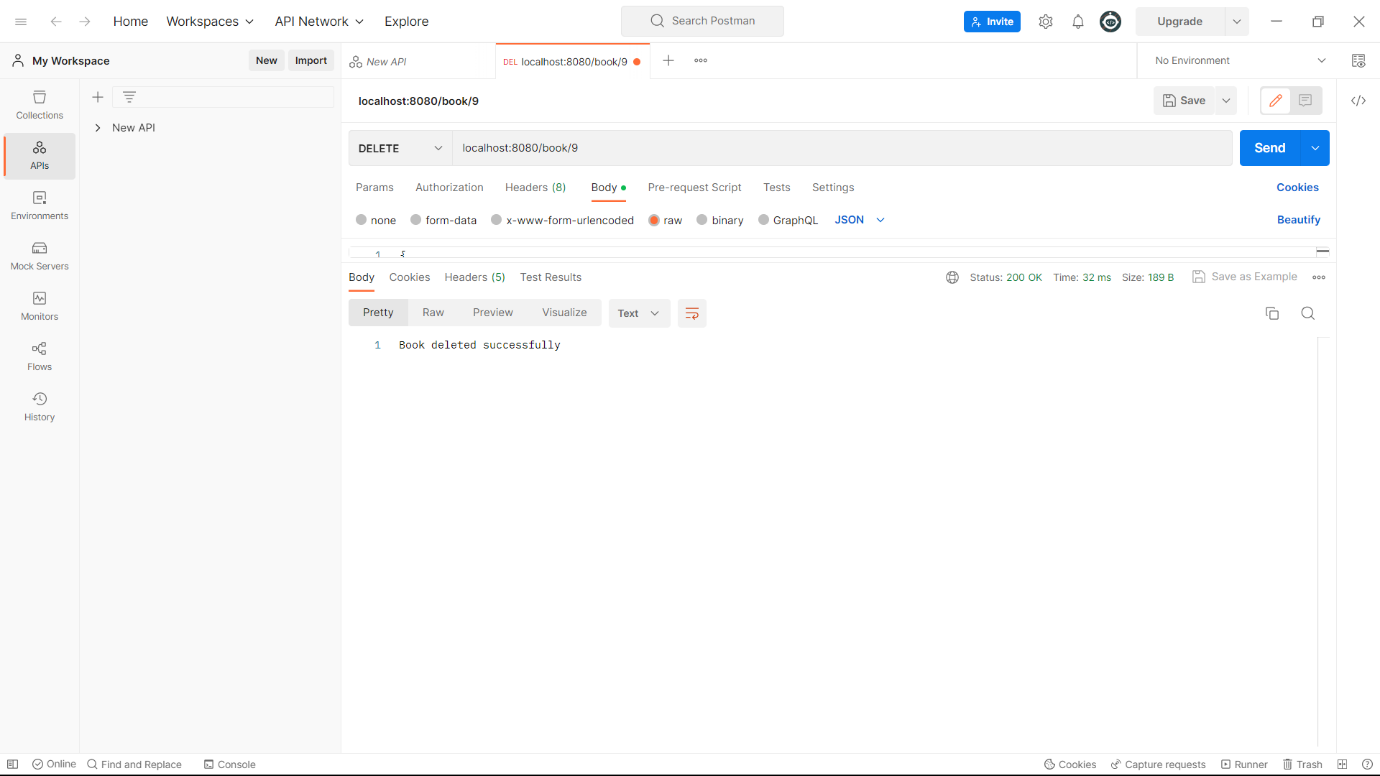
**Insert**



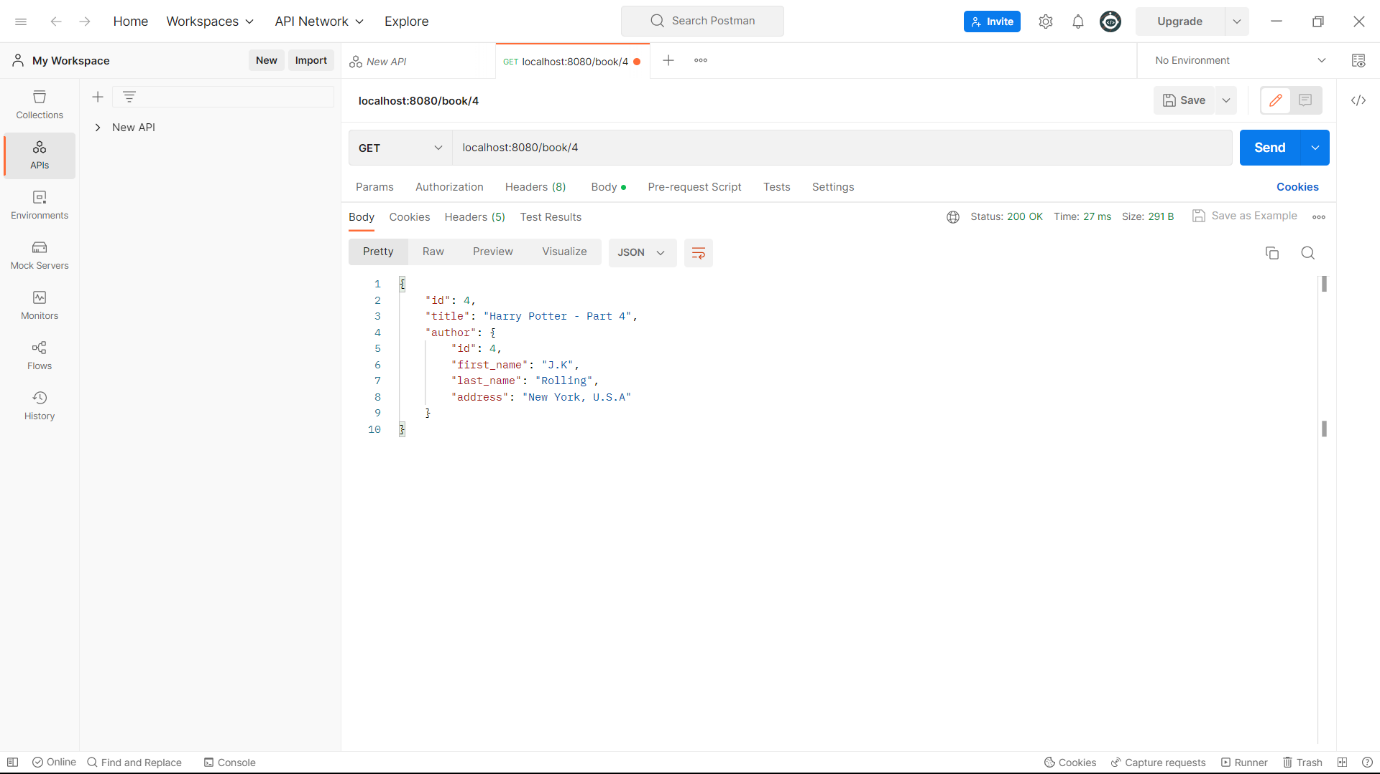
**Update**

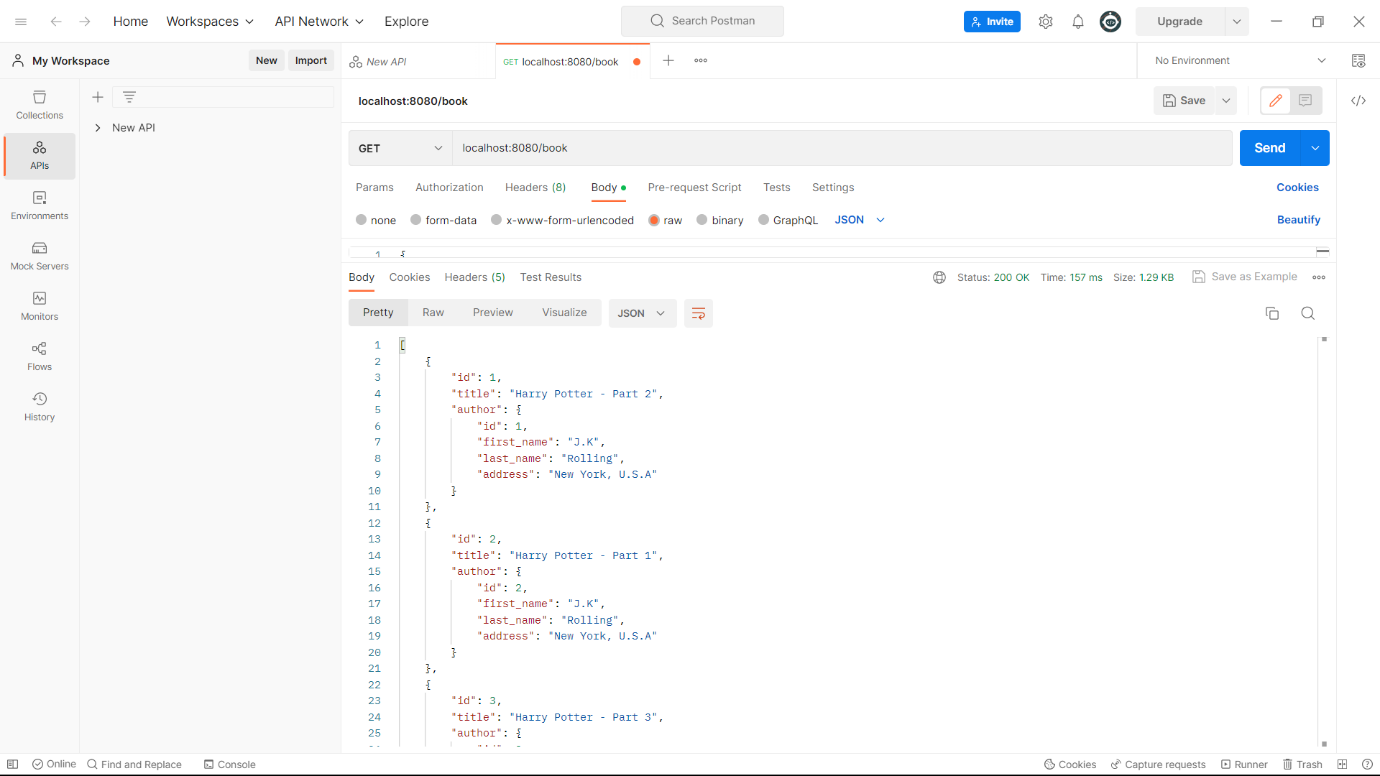


**delete**

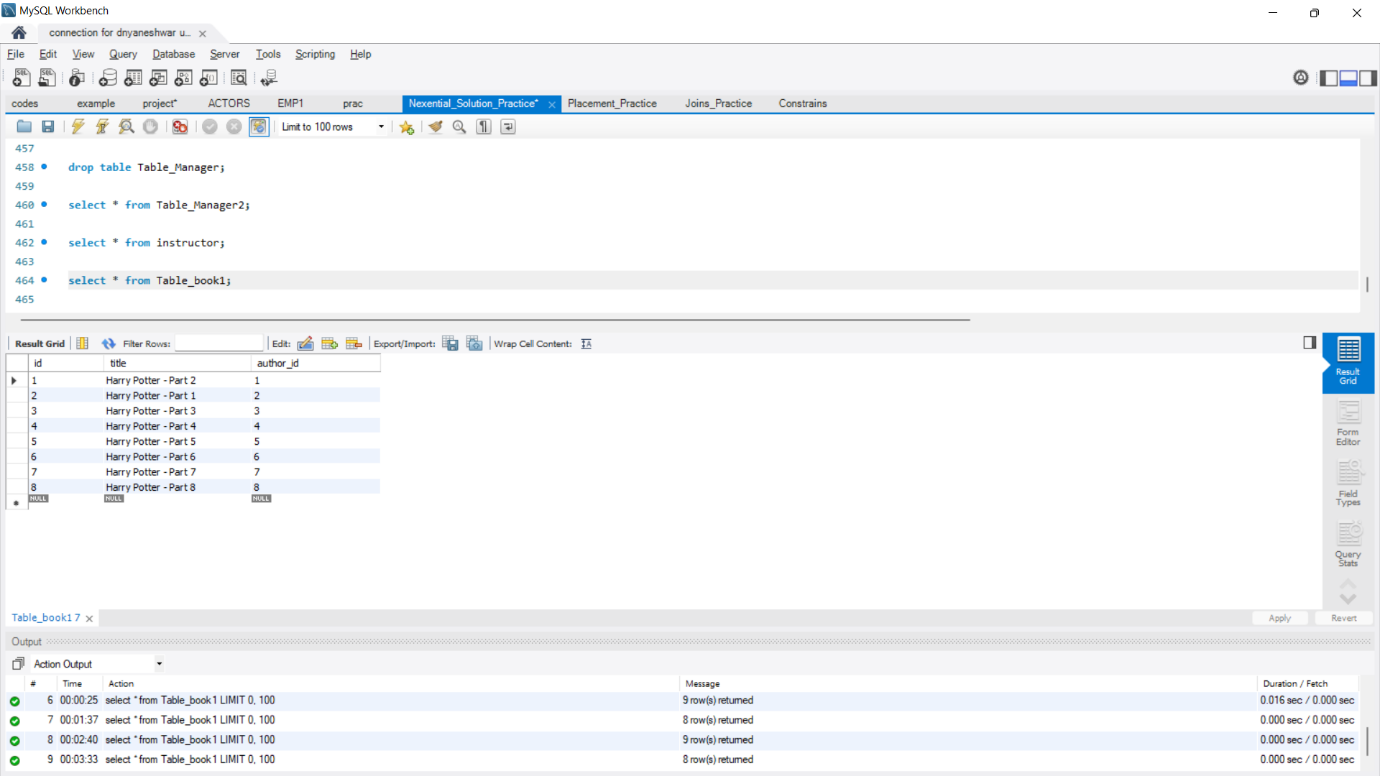


**select by id**



**select all**

**Table Book**



**Table Author**

