**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-One2Many-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

**Post.java**

package com.cdac.model;

import java.util.ArrayList;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Column;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.Table;

@Entity

@Table(name = "Table\_posts")

public class Post {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(name = "title")

private String title;

@Column(name = "description")

private String description;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name = "pc\_fid", referencedColumnName = "id")

List<Comment> comments = new ArrayList<>();

public Post() {

}

public Post(String title, String description) {

super();

this.title = title;

this.description = description;

}

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 String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public List<Comment> getComments() {

return comments;

}

public void setComments(List<Comment> comments) {

this.comments = comments;

}

}

**Comment.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\_comments")

public class Comment {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

private String text;

public Comment() {

}

public Comment(String text) {

super();

this.text = text;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getText() {

return text;

}

public void setText(String text) {

this.text = text;

}

}

**PostDao.java**

package com.cdac.dao;

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

import org.springframework.stereotype.Repository;

import com.cdac.model.Post;

@Repository

public interface PostDao extends JpaRepository<Post, Integer>{

}

**CommentDao.java**

package com.cdac.dao;

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

import org.springframework.stereotype.Repository;

import com.cdac.model.Comment;

@Repository

public interface CommentDao extends JpaRepository<Comment, Integer>{

}

**PostService.java**

package com.cdac.service;

import java.util.List;

import com.cdac.model.Post;

public interface PostService {

void add(Post post);

void modify(Post post);

void remove\_By\_Id(int id);

Post get\_By\_Id(int id);

List<Post> get\_All();

}

**PostServiceImpl.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.PostDao;

import com.cdac.model.Post;

@Service

public class PostServiceImpl implements PostService {

@Autowired

private PostDao postDao;

@Override

public void add(Post post) {

postDao.save(post);

}

@Override

public void modify(Post post) {

postDao.save(post);

}

@Override

public void remove\_By\_Id(int id) {

postDao.deleteById(id);

}

@Override

public Post get\_By\_Id(int id) {

Optional<Post> opt = postDao.findById(id);

if (opt.isPresent()) {

return opt.get();

}

return null;

}

@Override

public List<Post> get\_All() {

Iterable<Post> itr = postDao.findAll();

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

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

return lst;

}

}

**PostController.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.Post;

import com.cdac.service.PostService;

@RestController

public class PostController {

@Autowired

private PostService postService;

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

public String add\_Post(@RequestBody Post post) {

postService.add(post);

return "Post added successfully";

}

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

public String update\_Post(@RequestBody Post post) {

postService.modify(post);

return "Post updated successfully";

}

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

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

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

postService.remove\_By\_Id(id);

return "Post deleted successfully";

}

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

public String delete\_Post1(@PathVariable int id) {

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

postService.remove\_By\_Id(id);

return "Post deleted successfully";

}

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

public Post get\_Post\_By\_Id(@PathVariable int id) {

return postService.get\_By\_Id(id);

}

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

public List<Post> get\_All\_Posts() {

return postService.get\_All();

}

}

**SBOne2ManyMappingProject.java**

package com.cdac;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class SBOne2ManyMappingProjectApplication {

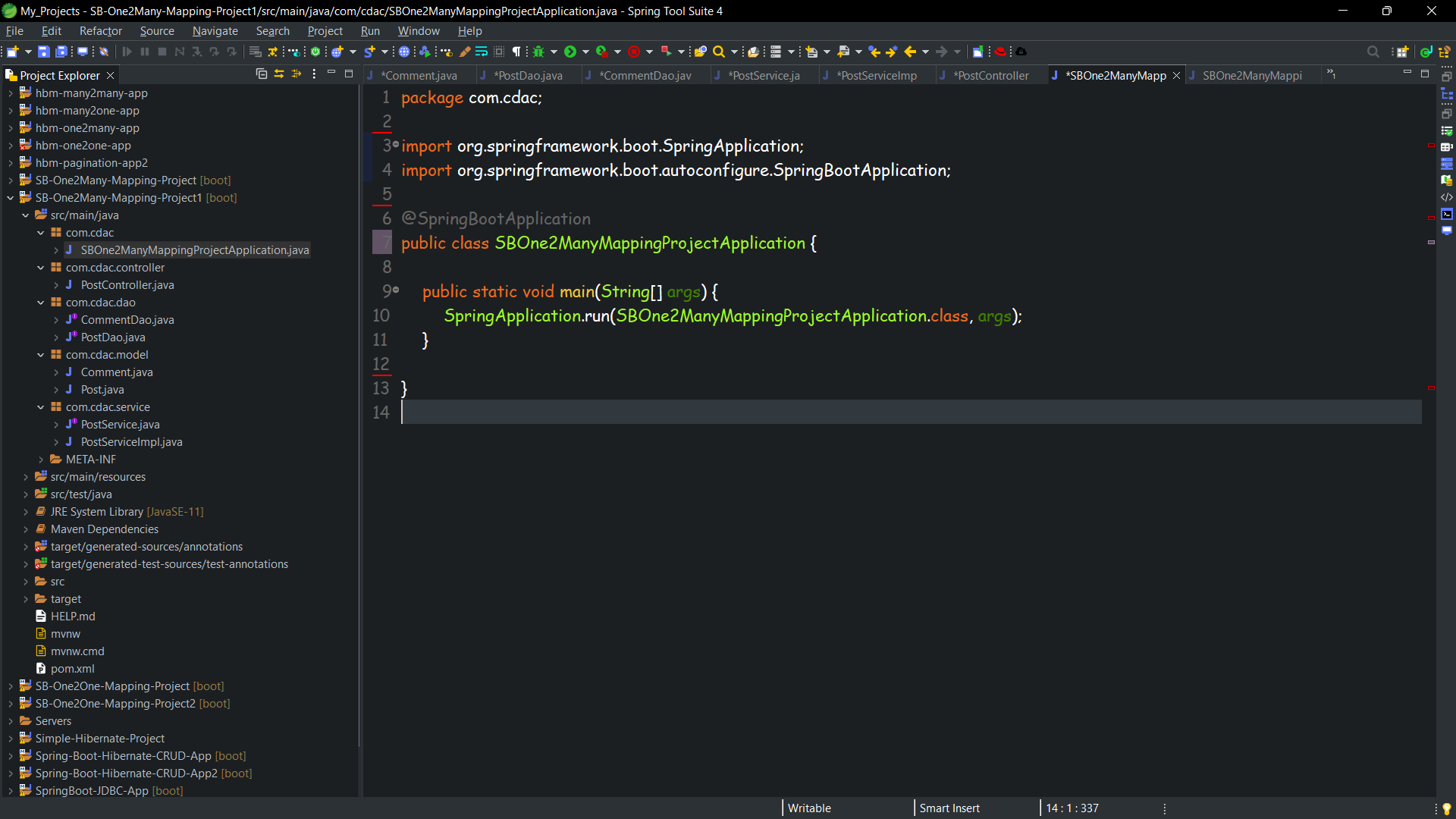
public static void main(String[] args) {

SpringApplication.run(SBOne2ManyMappingProjectApplication.class, args);

}

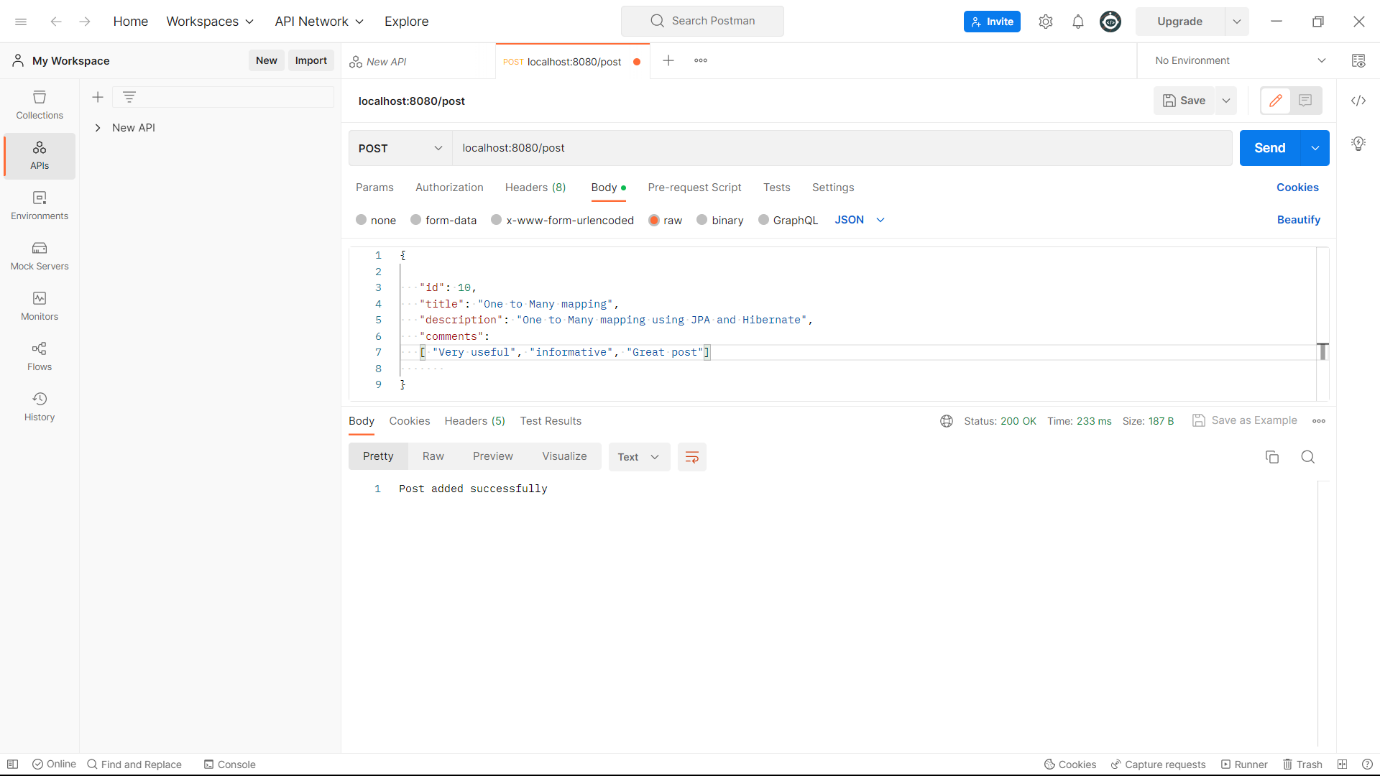
}

**Project Structrure**

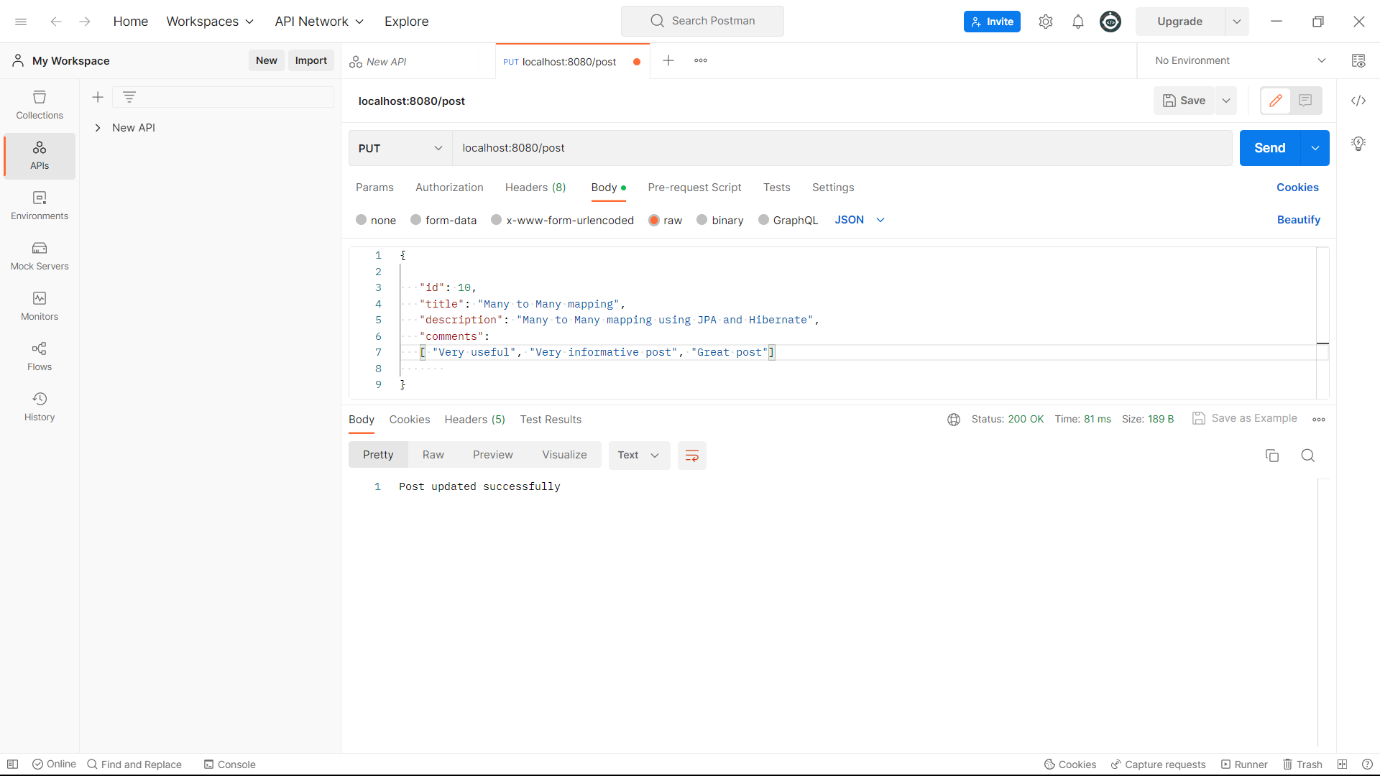


**PostMan and Database ScreenShots**

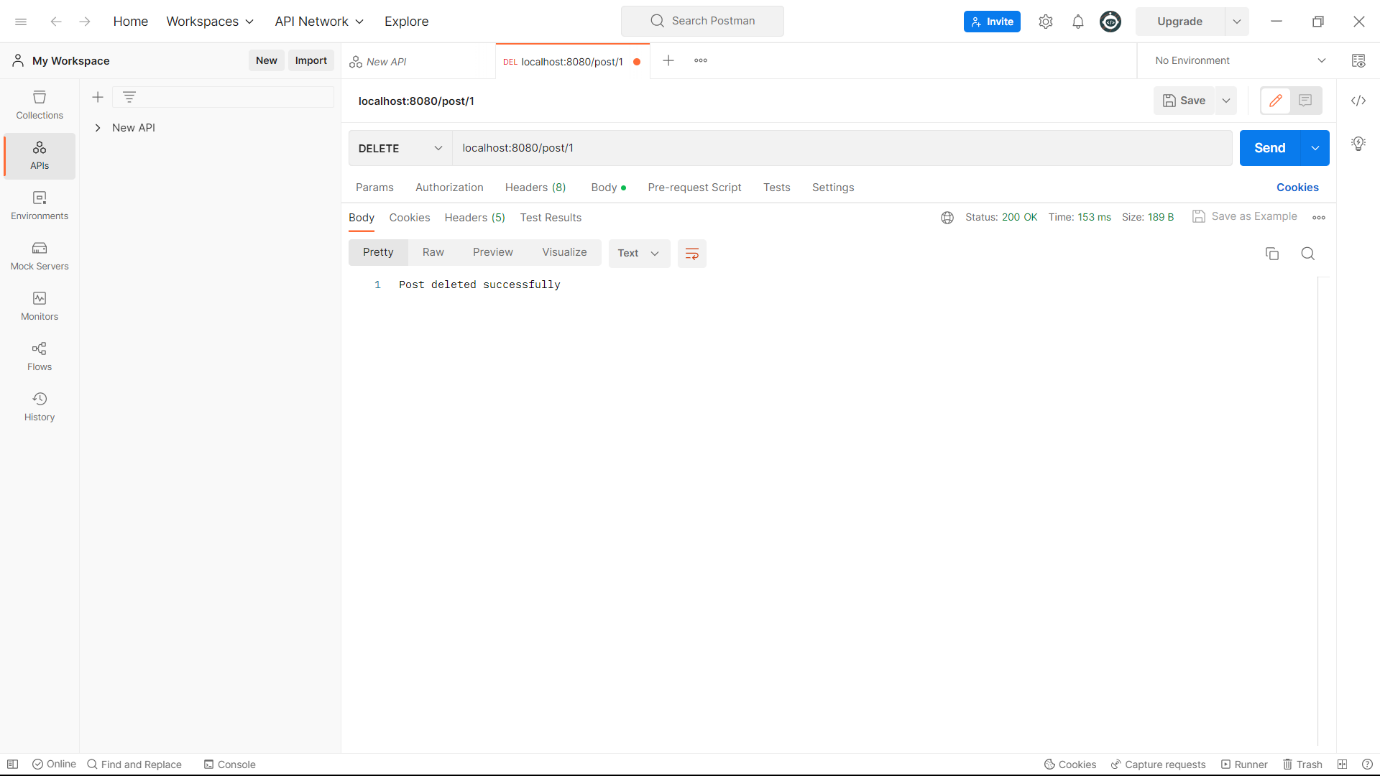
**Insert**



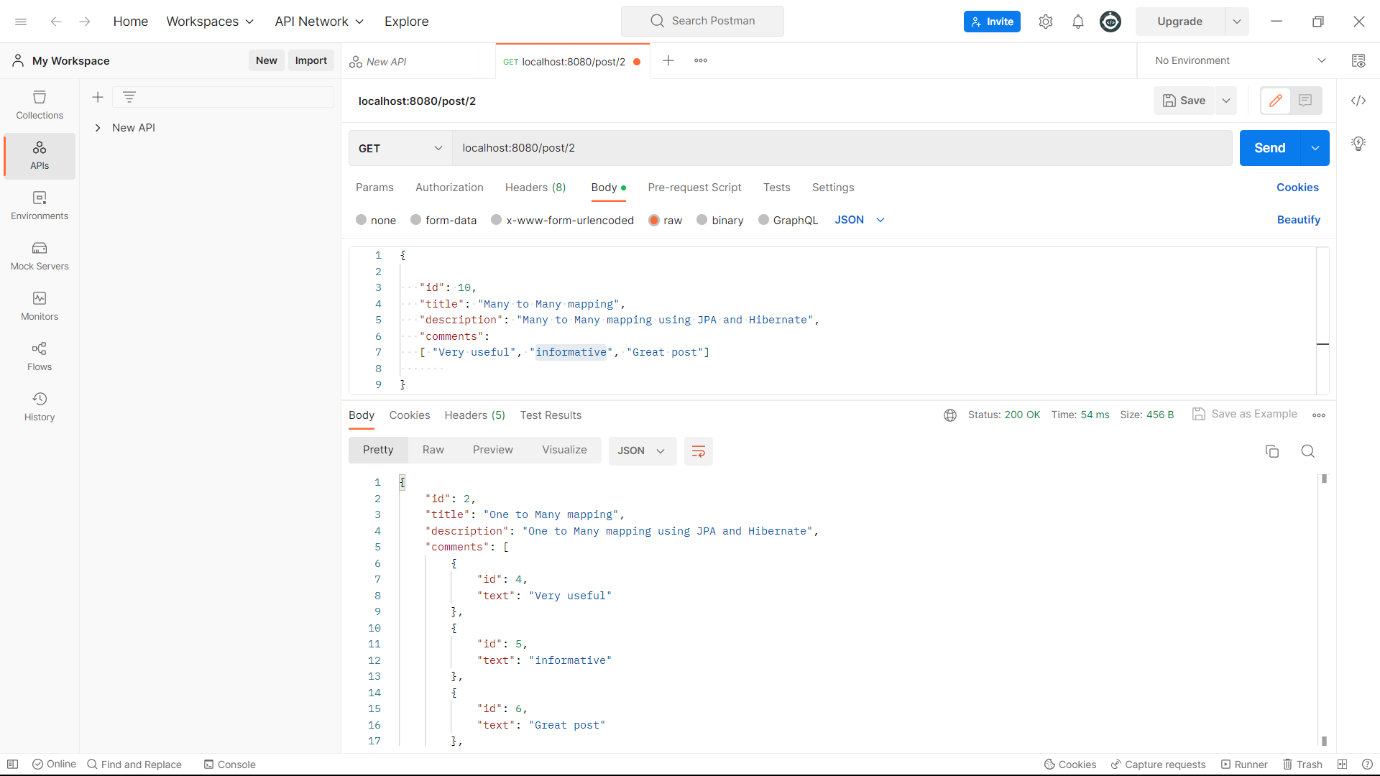
**Update**

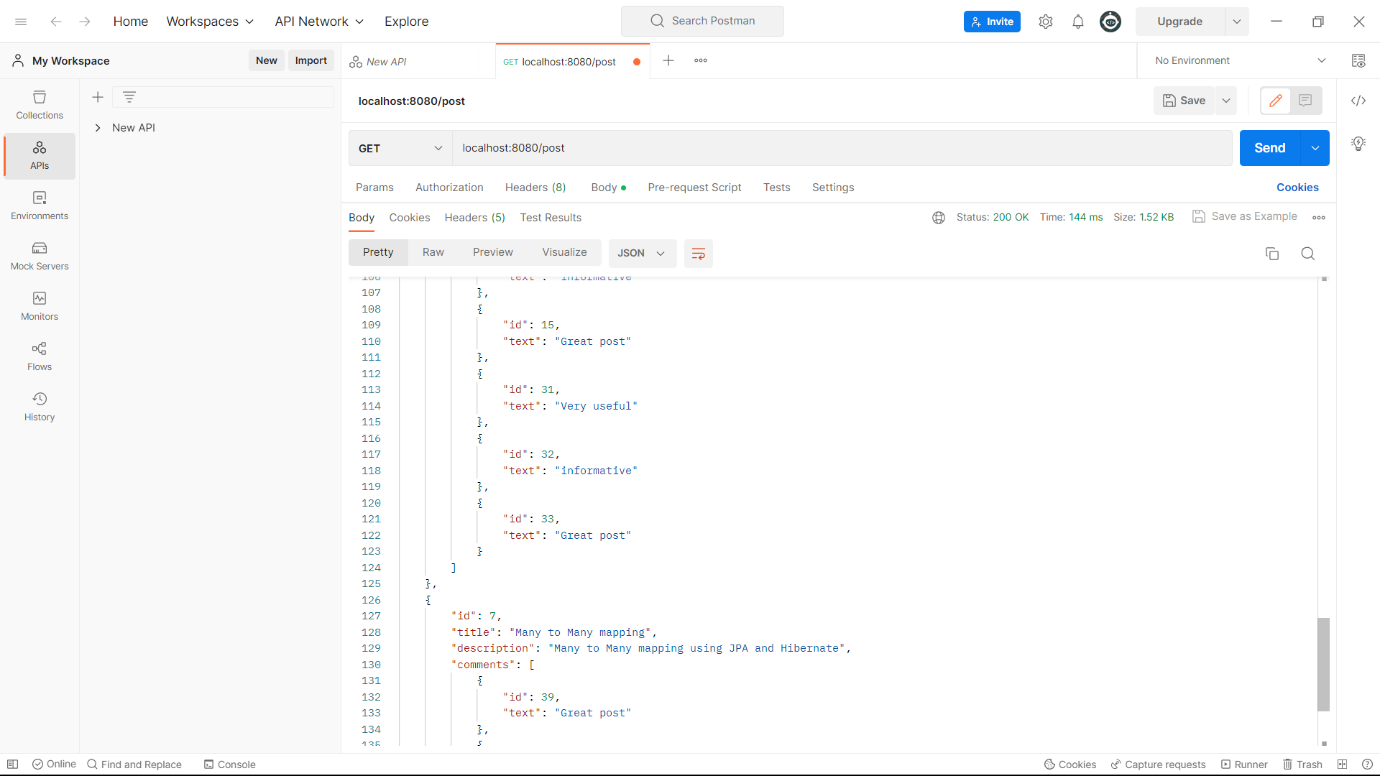


**delete**

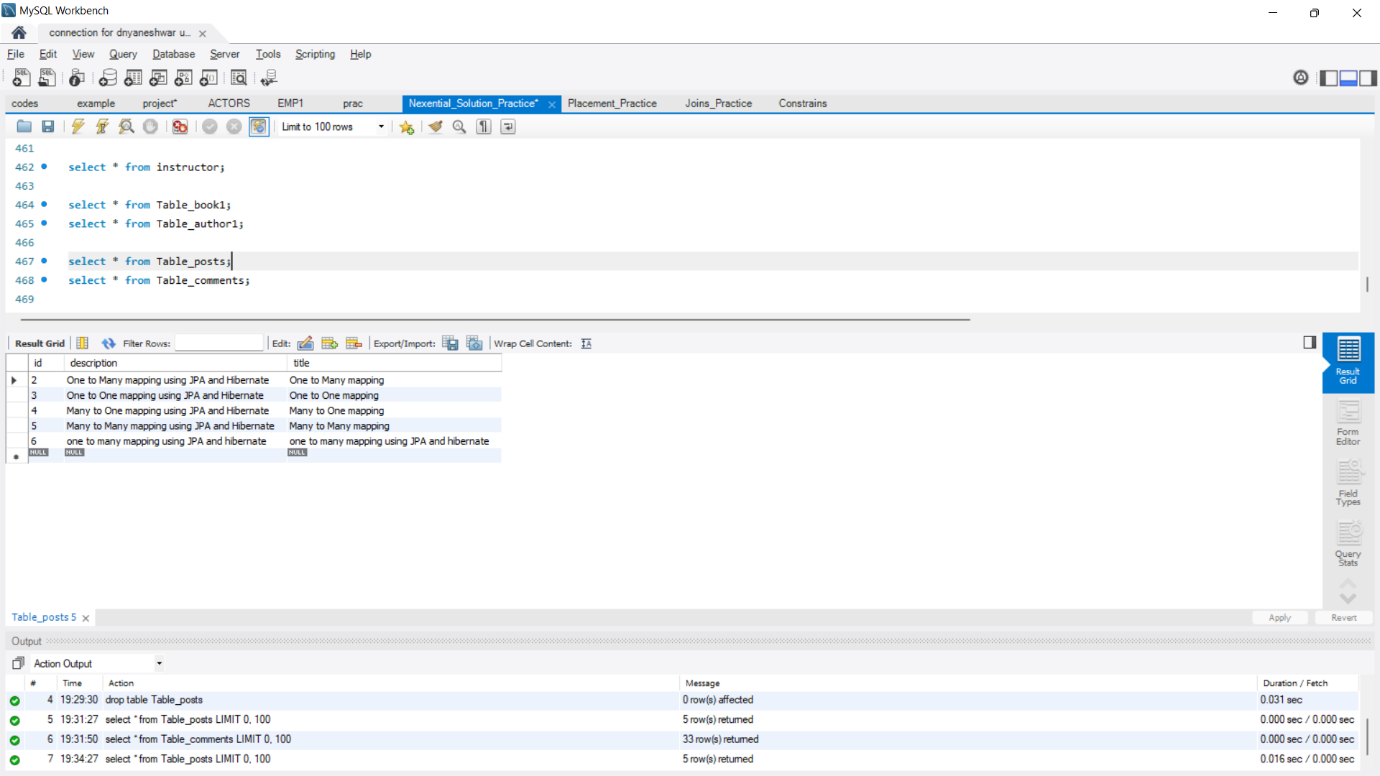


**select by id**



**select all** 

**Table Post**



**Table Comment**