**Courses.java**

**package com.springrest.entities;**

**public class Courses**

**{**

**private long id;**

**private String title;**

**private String description;**

**public Courses(long id, String title, String description) {**

**super();**

**this.id = id;**

**this.title = title;**

**this.description = description;**

**}**

**public Courses() {**

**super();**

**// TODO Auto-generated constructor stub**

**}**

**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 getDescription() {**

**return description;**

**}**

**public void setDescription(String description) {**

**this.description = description;**

**}**

**@Override**

**public String toString() {**

**return "Courses [id=" + id + ", title=" + title + ", description=" + description + "]";**

**}**

**}**

**CourseServiceImpl.java**

package com.springrest.Services;

import java.util.ArrayList;

import java.util.List;

import java.util.stream.Collectors;

import org.springframework.stereotype.Service;

import com.springrest.entities.Courses;

@Service

public class CourseServiceImpl implements courseService

{

//Temporary list to store data

List<Courses> list;

public CourseServiceImpl()

{

list = new ArrayList();

list.add(new Courses(123,"Java","Basic core java Course"));

list.add(new Courses(345,"DBMS","MySql Course"));

list.add(new Courses(101,"Data Analysis","basic Course"));

}

@Override

public List<Courses> getCourses()

{

return list;

}

@Override

public Courses getCourse(long courseId)

{

Courses c = null;

for(Courses course:list)

{

if(course.getId()==courseId)

{

c=course;

break;

}

}

return c;

}

@Override

public Courses addCourse(Courses course)

{

list.add(course);

return course;

}

}

**CourseService.java**

package com.springrest.Services;

import java.util.List;

import com.springrest.entities.Courses;

**public interface courseService**

**{**

**public List<Courses> getCourses();**

**public Courses getCourse(long courseId);**

**public Courses addCourse(Courses course);**

**public Courses updateCourse(Courses course);**

**public void deleteCourse(long parseLong);**

**}**

**MyController.java**

**package com.springrest.Controller;**

**import java.util.List;**

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

**import org.springframework.http.HttpStatus;**

**import org.springframework.http.ResponseEntity;**

**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.RestController;**

**import com.springrest.Services.courseService;**

**import com.springrest.entities.Courses;**

**@RestController**

**public class MyController {**

**@Autowired**

**private courseService cservice;**

**//get all courses**

**@GetMapping("/courses")**

**public List<Courses> getCourses()**

**{**

**return this.cservice.getCourses();**

**}**

**//get specific course with id**

**@GetMapping("courses/{courseId}")**

**public Courses getCourse(@PathVariable String courseId)**

**{**

**return this.cservice.getCourse(Long.*parseLong*(courseId));**

**}**

**//add course to list**

**@PostMapping("/courses")**

**public Courses addCourse(@RequestBody Courses course)**

**{**

**return this.cservice.addCourse(course);**

**}**

**//update course using PUT**

**@PutMapping("/courses")**

**public Courses updateCourse(@RequestBody Courses course)**

**{**

**return this.cservice.updateCourse(course);**

**}**

**//delete course using DELETE**

**@DeleteMapping("/courses/{courseId}")**

**public ResponseEntity deleteCourse(@PathVariable String courseId)**

**{**

**try {**

**this.cservice.deleteCourse(Long.*parseLong*(courseId));**

**return new ResponseEntity(HttpStatus.*OK*);**

**}catch(Exception e)**

**{**

**return new ResponseEntity(HttpStatus.*INTERNAL\_SERVER\_ERROR*);**

**}**

**}**

**}**

**AFTER ADDING DATA BASE REPOSITORY**

**CourseDAO.java**

package com.springrest.DAO;

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

import com.springrest.entities.Courses;

public interface CourseDAO extends JpaRepository<Courses,Long>

{

}

**MyController.java**

package com.springrest.Controller;

import java.util.List;

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

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

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.RestController;

import com.springrest.Services.courseService;

import com.springrest.entities.Courses;

@RestController

public class MyController {

@Autowired

private courseService cservice;

//get all courses

@GetMapping("/courses")

public List<Courses> getCourses()

{

return this.cservice.getCourses();

}

//get specific course with id

@GetMapping("courses/{courseId}")

public Courses getCourse(@PathVariable String courseId)

{

return this.cservice.getCourse(Long.parseLong(courseId));

}

//add course to list

@PostMapping("/courses")

public Courses addCourse(@RequestBody Courses course)

{

return this.cservice.addCourse(course);

}

//update course using PUT

@PutMapping("/courses")

public Courses updateCourse(@RequestBody Courses course)

{

return this.cservice.updateCourse(course);

}

//delete course using DELETE

@DeleteMapping("/courses/{courseId}")

public ResponseEntity deleteCourse(@PathVariable String courseId)

{

try

{

this.cservice.deleteCourse(Long.parseLong(courseId));

return new ResponseEntity(HttpStatus.OK);

}catch(Exception e)

{

return new ResponseEntity(HttpStatus.INTERNAL\_SERVER\_ERROR);

}

}

}

**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>3.1.0</version>**

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

**</parent>**

**<groupId>com.springrest</groupId>**

**<artifactId>SpringRestProject</artifactId>**

**<version>0.0.1-SNAPSHOT</version>**

**<name>Demo\_Project-1</name>**

**<description> project for Spring Boot</description>**

**<properties>**

**<java.version>17</java.version>**

**</properties>**

**<dependencies>**

**<dependency>**

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

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

**</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>**

**<version>3.0.3</version>**

**</dependency>**

**<dependency>**

**<groupId>org.springframework.data</groupId>**

**<artifactId>spring-data-commons</artifactId>**

**<version>3.0.2</version>**

**</dependency>**

**<dependency>**

**<groupId>org.springframework.data</groupId>**

**<artifactId>spring-data-jpa</artifactId>**

**<version>3.0.2</version>**

**</dependency>**

**<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->**

**<dependency>**

**<groupId>mysql</groupId>**

**<artifactId>mysql-connector-java</artifactId>**

**<version>8.0.30</version>**

**</dependency>**

**</dependencies>**

**<build>**

**<plugins>**

**<plugin>**

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

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

**</plugin>**

**</plugins>**

**</build>**

**</project>**

**CourseServiceImpl.java**

**package com.springrest.Services;**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Optional;**

**import java.util.stream.Collectors;**

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

**import org.springframework.stereotype.Service;**

**import com.springrest.DAO.CourseDAO;**

**import com.springrest.entities.Courses;**

**@Service**

**public class CourseServiceImpl implements courseService**

**{**

**//Temporary list to store data**

**//List<Courses> list;**

**@Autowired**

**private CourseDAO coursedao;**

**public CourseServiceImpl()**

**{**

**// list = new ArrayList();**

**// list.add(new Courses(123,"Java","Basic core java Course"));**

**// list.add(new Courses(345,"DBMS","MySql Course"));**

**// list.add(new Courses(101,"Data Analysis","basic Course"));**

**}**

**@Override**

**public List<Courses> getCourses()**

**{**

**return coursedao.findAll();**

**}**

**@Override**

**public Courses getCourse(long courseId)**

**{**

**// Courses c = null;**

**// for(Courses course:list)**

**// {**

**// if(course.getId()==courseId)**

**// {**

**// c=course;**

**// break;**

**// }**

**// }**

**final Courses c1 = coursedao.getOne(courseId);**

**return c1;**

**}**

**@Override**

**public Courses addCourse(Courses course)**

**{**

**//list.add(course);**

**coursedao.save(course);**

**return course;**

**}**

**@Override**

**public Courses updateCourse(Courses course)**

**{**

**// list.forEach(e->**

**// {**

**// if(e.getId()==course.getId())**

**// {**

**// e.setTitle(course.getTitle());**

**// e.setDescription(course.getDescription());**

**//**

**// }**

**// });**

**coursedao.save(course);**

**return course;**

**}**

**@Override**

**public void deleteCourse(long parseLong)**

**{**

**//list=this.list.stream().filter(e-//>e.getId()!=parseLong).collect(Collectors.toList());**

**Courses entity =coursedao.getOne(parseLong);**

**coursedao.delete(entity);**

**}**

**}**