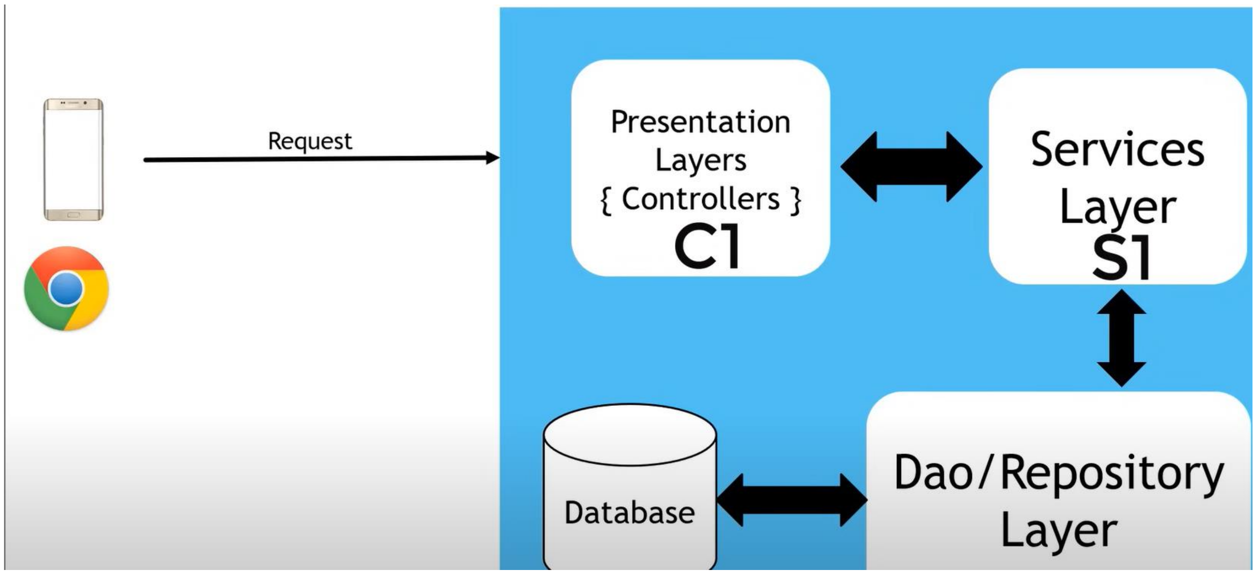
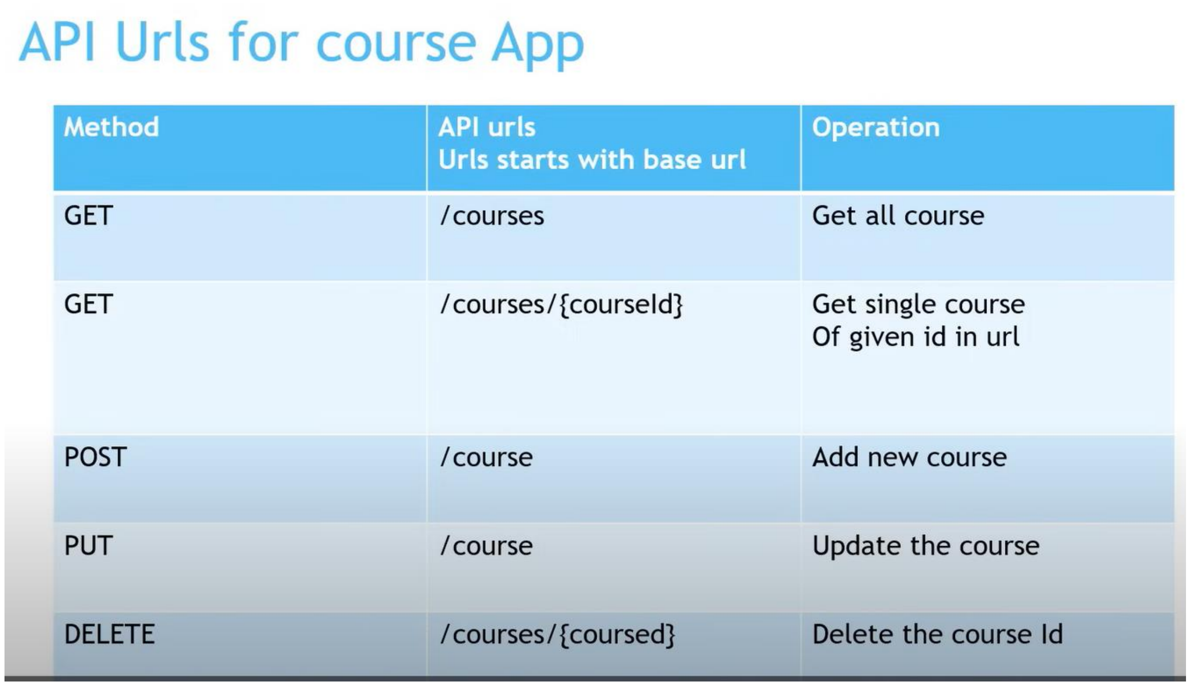
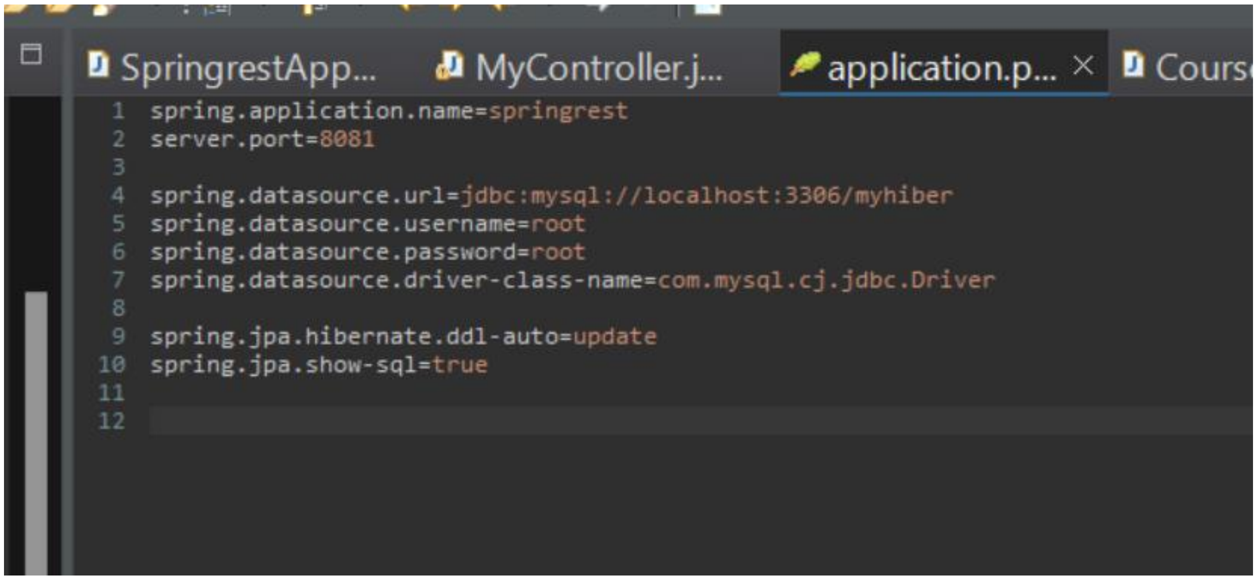
Rest API Using spring boot





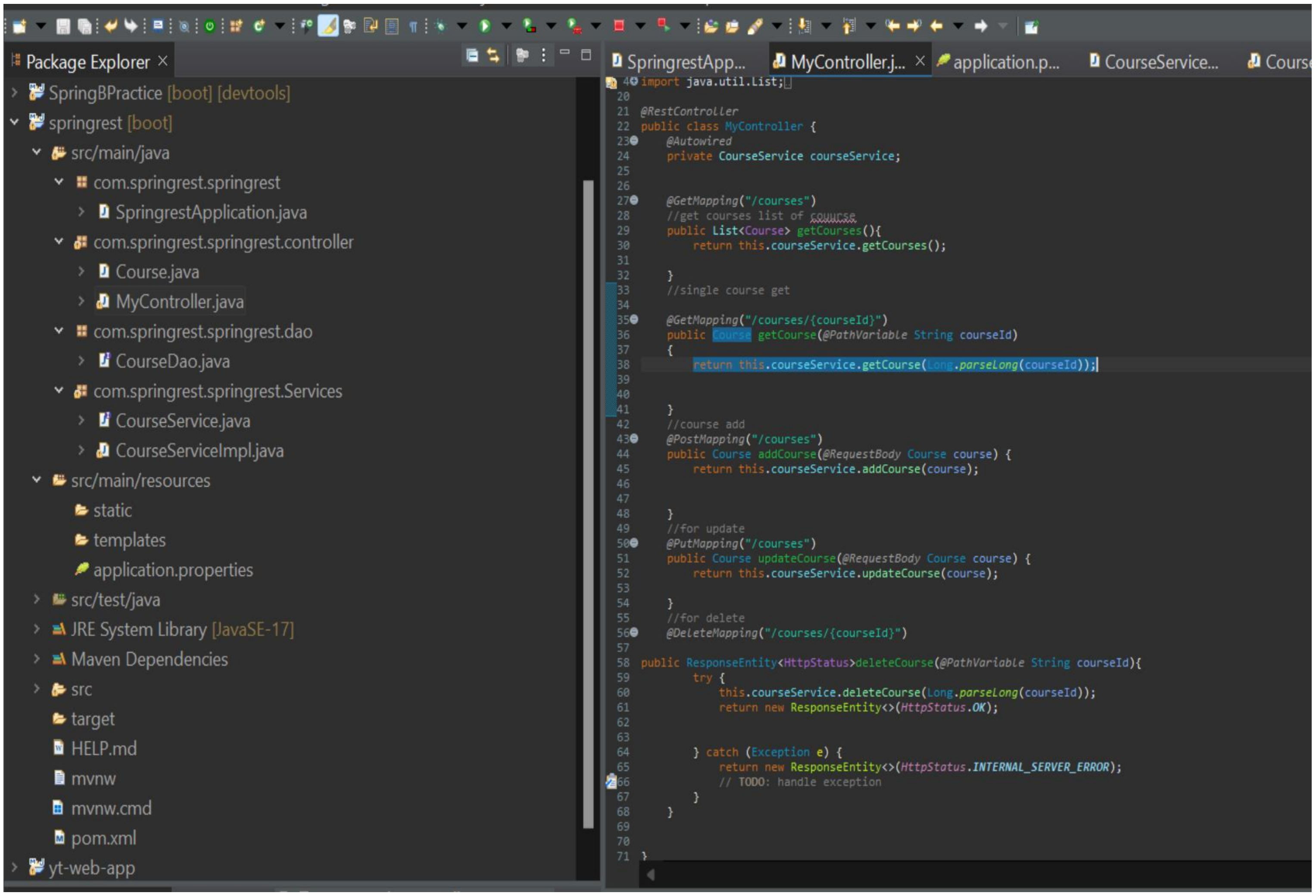
API Urls for course App

Method	API urls Urls starts with base url	Operation
GET	/courses	Get all course
GET	/courses/{courseId}	Get single course Of given id in url
POST	/course	Add new course
PUT	/course	Update the course
DELETE	/courses/{coursed}	Delete the course Id



```
spring.application.name=springrest
server.port=8081

spring.datasource.url=jdbc:mysql://localhost:3306/myhiber
spring.datasource.username=root
spring.datasource.password=root
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

```

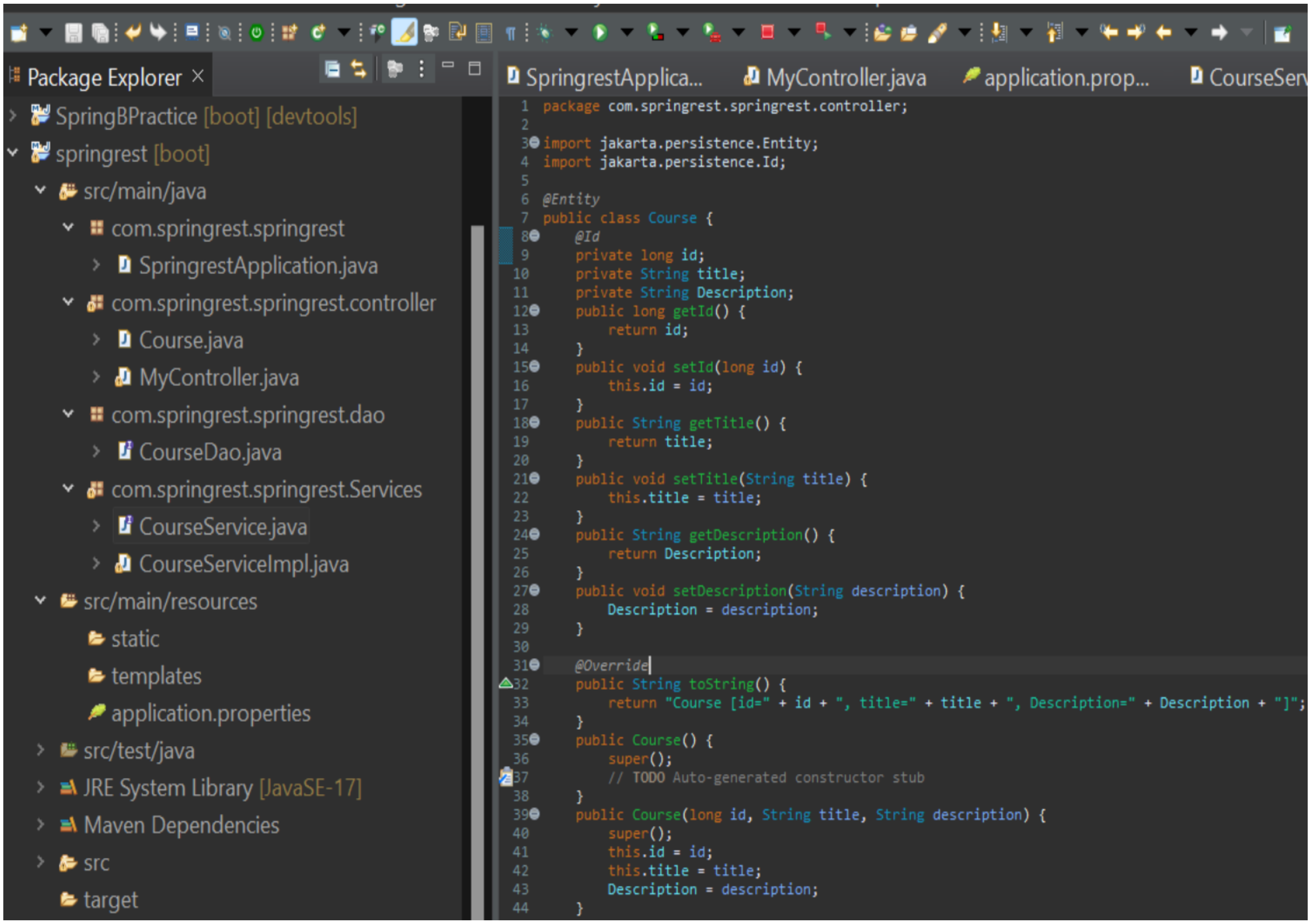
My Controller →controll all request and All Api are designed here



```
SpringBPractice [boot] [devtools]
springrest [boot]
    src/main/java
        com.springrest.springrest
            SpringrestApplication.java
        com.springrest.springrest.controller
            Course.java
            MyController.java
        com.springrest.springrest.dao
            CourseDao.java
        com.springrest.springrest.Services
            CourseService.java
            CourseServiceImpl.java
    src/main/resources
        static
        templates
        application.properties
    src/test/java
    JRE System Library [JavaSE-17]
    Maven Dependencies
    src
    target
    HELP.md
    mvnw
    mvnw.cmd
    pom.xml
yt-web-app
```

```java
import java.util.List;

@RestController
public class MyController {
    @Autowired
    private CourseService courseService;


    @GetMapping("/courses")
    //get courses list of couurse
    public List<Course> getCourses(){
        return this.courseService.getCourses();

    }
    //single course get

    @GetMapping("/courses/{courseId}")
    public Course getCourse(@PathVariable String courseId)
    {
        return this.courseService.getCourse(Long.parseLong(courseId));


    }
    //course add
    @PostMapping("/courses")
    public Course addCourse(@RequestBody Course course) {
        return this.courseService.addCourse(course);


    }
    //for update
    @PutMapping("/courses")
    public Course updateCourse(@RequestBody Course course) {
        return this.courseService.updateCourse(course);

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

public ResponseEntity<HttpStatus>deleteCourse(@PathVariable String courseId){
        try {
            this.courseService.deleteCourse(Long.parseLong(courseId));
            return new ResponseEntity<>(HttpStatus.OK);


        } catch (Exception e) {
            return new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
            // TODO: handle exception
        }
    }


}
```

This is course.java entity and column is here only

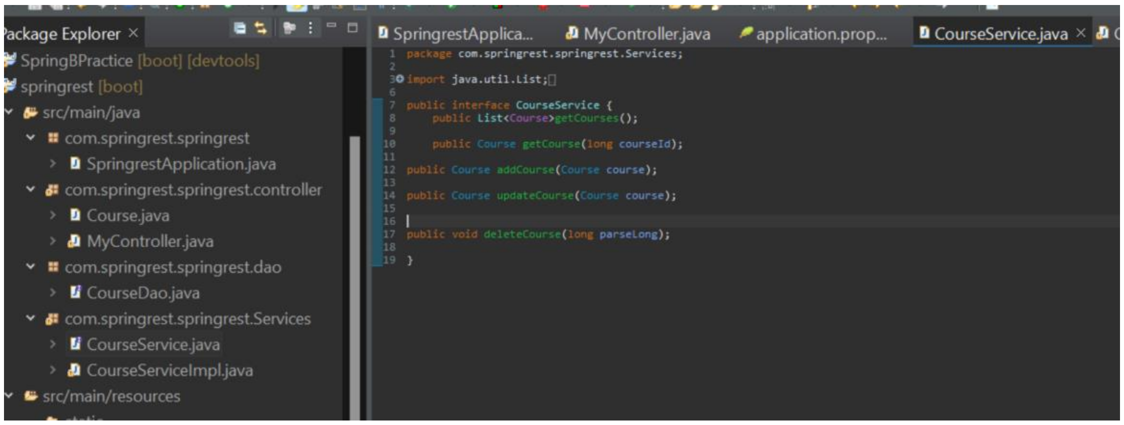


Package Explorer

- SpringBPractice [boot] [devtools]
- springrest [boot]
  - src/main/java
    - com.springrest.springrest
      - SpringrestApplication.java
    - com.springrest.springrest.controller
      - Course.java
      - MyController.java
    - com.springrest.springrest.dao
      - CourseDao.java
    - com.springrest.springrest.Services
      - CourseService.java
      - CourseServiceImpl.java
  - src/main/resources
    - static
    - templates
    - application.properties
  - src/test/java
  - JRE System Library [JavaSE-17]
  - Maven Dependencies
  - src
  - target

SpringrestApplica... MyController.java application.prop... CourseServ

```java
package com.springrest.springrest.controller;

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

@Entity
public class Course {
    @Id
    private long id;
    private String title;
    private String Description;
    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) {
        Description = description;
    }

    @Override
    public String toString() {
        return "Course [id=" + id + ", title=" + title + ", Description=" + Description + "]";
    }
    public Course() {
        super();
        // TODO Auto-generated constructor stub
    }
    public Course(long id, String title, String description) {
        super();
        this.id = id;
        this.title = title;
        Description = description;
    }
```

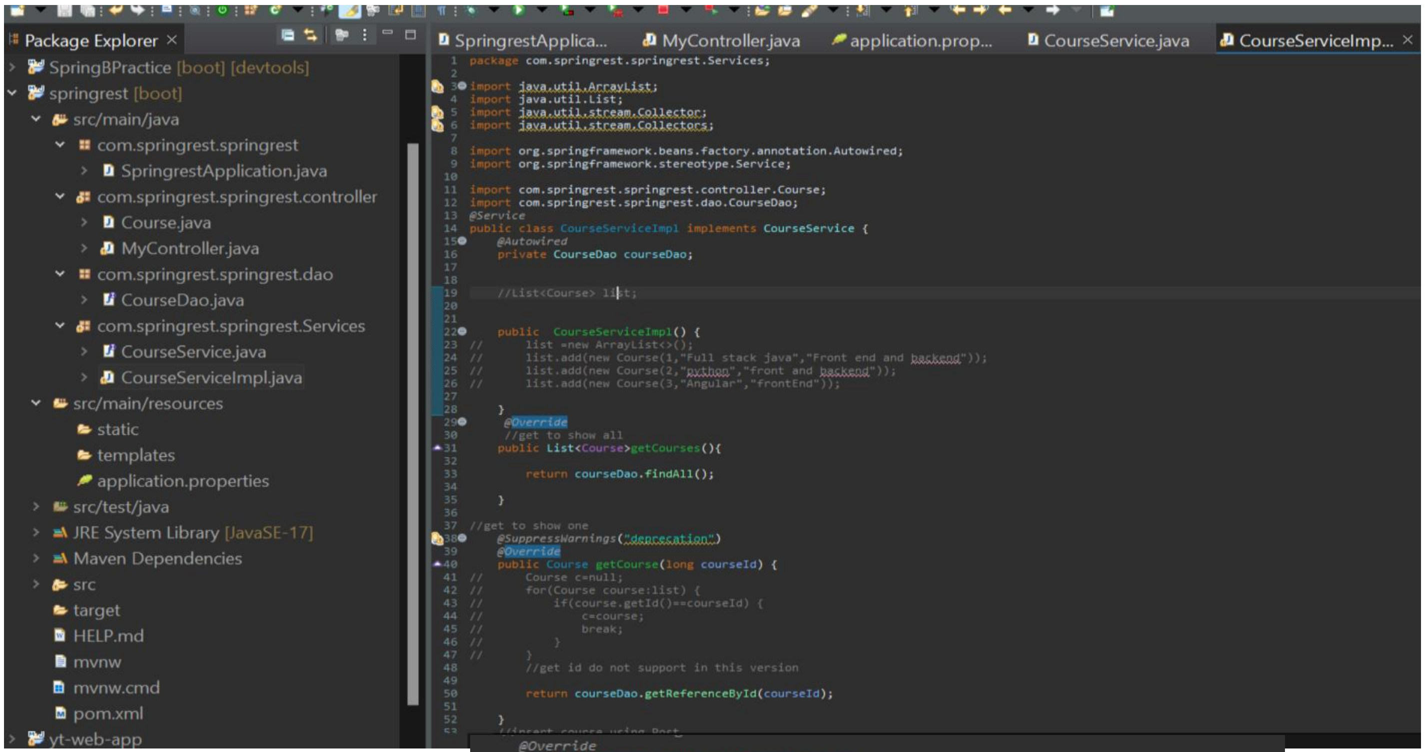


```java
package com.springrest.springrest.Services;

import java.util.List;

public interface CourseService {
    public List<Course>getCourses();

    public Course getCourse(long courseId);

public Course addCourse(Course course);

public Course updateCourse(Course course);


public void deleteCourse(long parseLong);

}
```

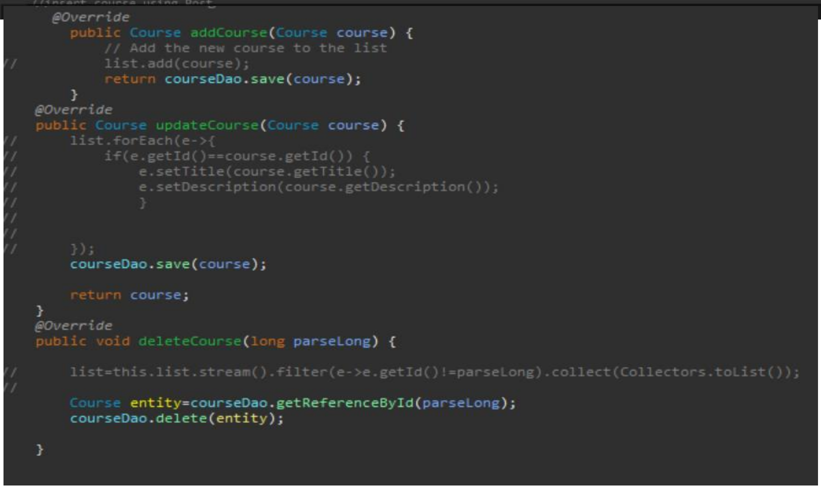
Then create Service Layer→

in service layer Business logic is written this in an interface and it is 100% incomplete so we need body to write logic so create a class and implements properties

---

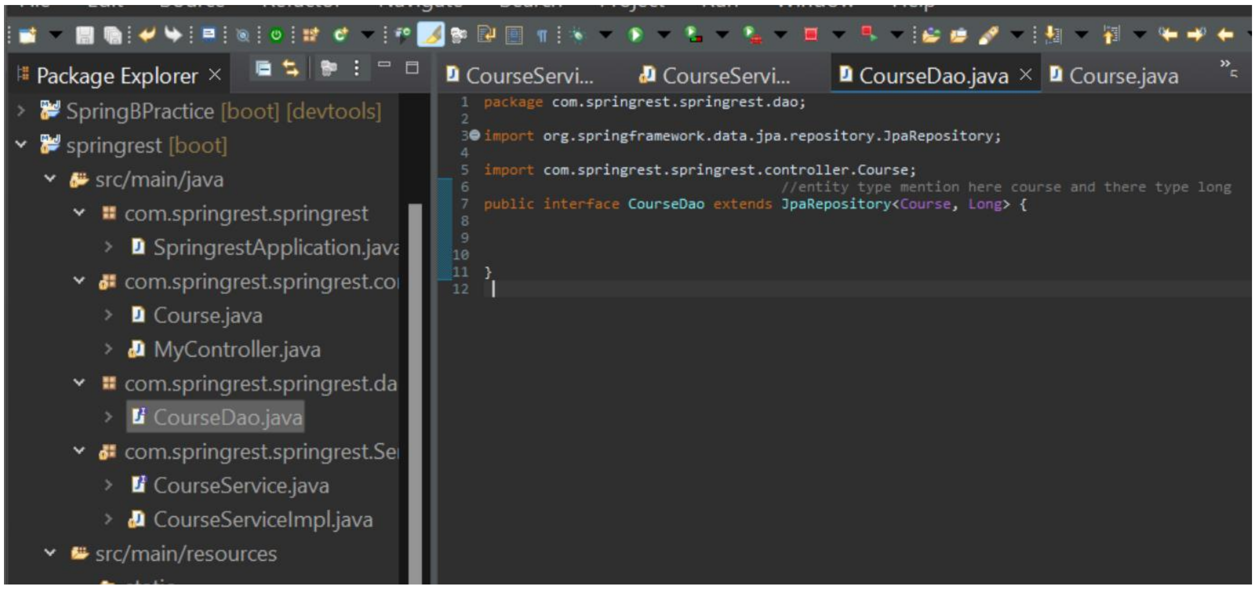


```java
package com.springrest.springrest.Services;

import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collector;
import java.util.stream.Collectors;

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

import com.springrest.springrest.controller.Course;
import com.springrest.springrest.dao.CourseDao;
@Service
public class CourseServiceImpl implements CourseService {
    @Autowired
    private CourseDao courseDao;


    //List<Course> list;


    public  CourseServiceImpl() {
//      list =new ArrayList<>();
//      list.add(new Course(1,"Full stack java","Front end and backend"));
//      list.add(new Course(2,"python","front and backend"));
//      list.add(new Course(3,"Angular","frontEnd"));

    }
     @Override
     //get to show all
    public List<Course>getCourses(){

        return courseDao.findAll();

    }

//get to show one
    @SuppressWarnings("deprecation")
    @Override
    public Course getCourse(long courseId) {
//      Course c=null;
//      for(Course course:list) {
//          if(course.getId()==courseId) {
//              c=course;
//              break;
//          }
//      }
        //get id do not support in this version

        return courseDao.getReferenceById(courseId);

    }
    //insert course using Post
```

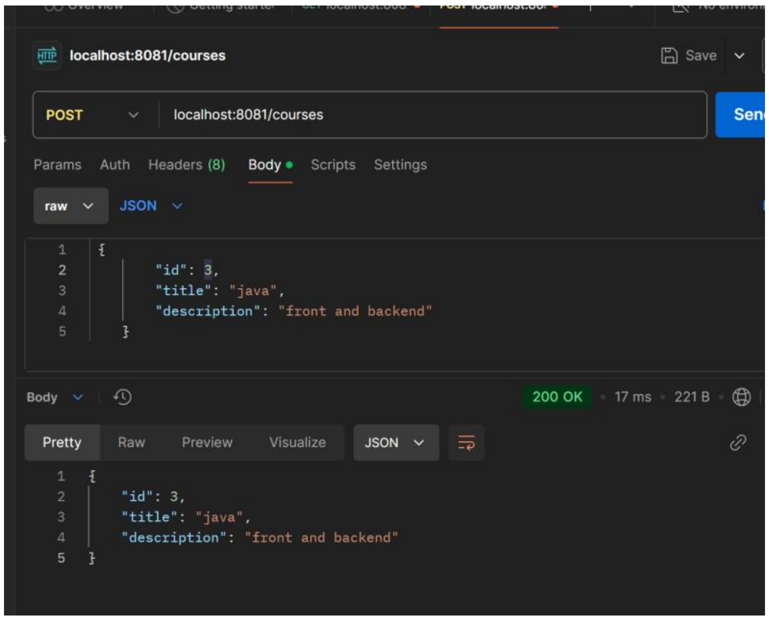


```java
      @Override
        public Course addCourse(Course course) {
            // Add the new course to the list
//          list.add(course);
            return courseDao.save(course);
        }
    @Override
    public Course updateCourse(Course course) {
//      list.forEach(e->{
//          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());
//
        Course entity=courseDao.getReferenceById(parseLong);
        courseDao.delete(entity);

    }
```

This is dao class which is directly interact with database so we just created one class and extends to JPA repository



```java
package com.springrest.springrest.dao;

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

import com.springrest.springrest.controller.Course;
                                        //entity type mention here course and there type long
public interface CourseDao extends JpaRepository<Course, Long> {



}
```



```json
{
    "id": 3,
    "title": "java",
    "description": "front and backend"
}
```

```json
{
    "id": 3,
    "title": "java",
    "description": "front and backend"
}
```