Muhammad Farhan Mardadi
1506757743

Write-up Tutorial 7

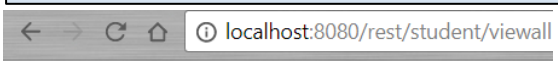
Yang dipelajari

Pada tutorial ini saya yang pelajari adalah bagaimana mengimplementasi web service sederhana. Dimana terdapat dua service, yang pertama sebagai producer yaitu bertugas untuk menyediakan data, yang kedua sebagai consumer yaitu aplikasi yang berinteraksi dengan pengguna yang dapat mengambil data dari producer sehingga tidak perlu mengambil data dari database.

Latihan

1. Untuk membuat service ini saya membuat method yang mengembalikan List of Student Model dengan cara mengambil student melalui service yang telah dibuat sebelumnya.

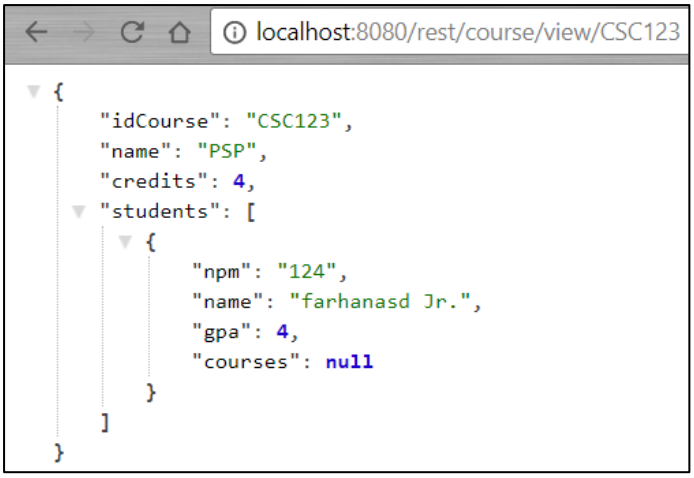
```
@RequestMapping("/student/viewall")
public List<StudentModel> viewAll() {
    List<StudentModel> students = studentService.selectAllStudents();
    return students;
}
```




```
[
  {
    "npm": "123",
    "name": "farhanasd",
    "gpa": 3.2,
    "courses": [
      {
        "idCourse": "CSC126",
        "name": "MPKT",
        "credits": 6,
        "students": null
      }
    ]
  },
  {
    "npm": "123123123",
    "name": "mardadi",
    "gpa": 4,
    "courses": []
  },
  {
    "npm": "1234",
    "name": "farhanasd",
    "gpa": 3.2,
    "courses": []
  },
  {
    "npm": "124",
    "name": "farhanasd Jr.",
    "gpa": 4,
    "courses": [
      {
        "idCourse": "CSC123",
        "name": "PSP",
        "credits": 4,
        "students": null
      },
```

2. Untuk membuat service pada kelas Course saya membuat class CourseRestController yang berisikin 2 mapping yakni untuk view course by id dan view all course, implementasinya mirip dengan yang ada pada class StudentRestController.

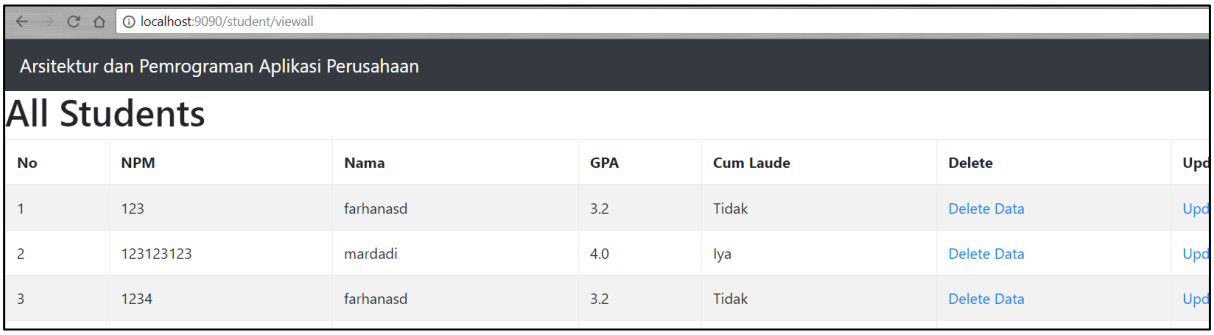
```java
package com.example.tutorial7.rest;

import java.util.List;

@RestController
@RequestMapping("/rest")
public class CourseRestController {

    @Autowired
    CourseService courseService;

    @RequestMapping("/course/view/{id}")
    public CourseModel view(@PathVariable(value = "id") String id) {
        CourseModel course = courseService.selectCourse(id);
        return course;
    }

    @RequestMapping("/course/viewall")
    public List<CourseModel> viewAll() {
        List<CourseModel> courses = courseService.selectAllCourses();
        return courses;
    }
}
```



3. Untuk membuat service consumer yang dapat menampilkan semua student. Pertama saya harus melengkapi method yang ada di class StudentDAOImpl yang mengambil object dari service producer. Lalu melengkapi method yang ada di class StudentServiceRest yang mengembalikan list dari semua student.

```
@SuppressWarnings("unchecked")
@Override
public List<StudentModel> selectAllStudents() {
    List<StudentModel> students = restTemplate.getForObject("http://localhost:8080/rest/student/viewall", List.class);
    return students;
}
```

```
@Override
public List<StudentModel> selectAllStudents() {
    Log.info("REST - select all students");
    return studentDAO.selectAllStudents();
}
```



4. Untuk membuat service consumer untuk class Course. Pertama saya membuat interface CourseDAO yang berisikan method-method yang akan dibutuhkan untuk class Course. Lalu membuat class CourseDAOImpl untuk mengimplementasikan method-method yang ada pada interface CourseDAO. Dan yang terakhir yaitu dengan membuat class CourseServiceRest yang mengimplementasikan method yang ada pada inteface CourseService.

```
package com.example.tutorial7.dao;

import java.util.List;

import com.example.tutorial7.model.CourseModel;

public interface CourseDAO {
    CourseModel selectCourse(String id);
    List<CourseModel> selectAllCourses();
}
```

```java
package com.example.tutorial7.dao;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

import com.example.tutorial7.model.CourseModel;

@Service
public class CourseDAOImpl implements CourseDAO{

    @Autowired
    private RestTemplate restTemplate;

    @Override
    public CourseModel selectCourse(String id) {
        CourseModel course = restTemplate.getForObject("http://localhost:8080/rest/course/view/" + id, CourseModel.class);
        return course;
    }

    @SuppressWarnings("unchecked")
    @Override
    public List<CourseModel> selectAllCourses() {
        List<CourseModel> courses = restTemplate.getForObject("http://localhost:8080/rest/course/viewall", List.class);
        return courses;
    }
}
```

```java
package com.example.tutorial7.service;

import java.util.List;

@Slf4j
@Service
@Primary
public class CourseServiceRest implements CourseService {

    @Lazy
    @Autowired
    private CourseDAO courseDAO;

    @Override
    public CourseModel selectCourse(String id) {
        log.info("REST - select course with id {}", id);
        return courseDAO.selectCourse(id);
    }

    @Override
    public List<CourseModel> selectAllCourses() {
        log.info("REST - select all courses");
        return courseDAO.selectAllCourses();
    }

}
```