

Pada tutorial kali ini saya mempelajari fungsi dari REST dan pembagian kode menjadi Producer dan juga Consumer serta bagaimana keduanya saling berhubungan satu sama lainnya.

Latihan 1:

Kode pada CourseRestController

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

Method ini berguna seperti method viewall pada studentController namun pada method ini, return valuenya adalah objek(List of StudentModel) yang akan ditunjukkan dalam bentuk JSON.

Latihan 2:

Kode pada class CourseRestController

```
@RestController
@RequestMapping("/rest")
public class CourseRestController {
    @Autowired
    StudentService studentService;
    @RequestMapping("/course/view/{id}")
    public CourseModel view (@PathVariable("id") String id) {
        CourseModel course = studentService.selectCourse(id);
        return course;
    }

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

Pada latihan ke-2 ini saya membuat class baru dengan nama CourseRestController yang berisi method method yang berhubungan dengan course dengan rest. @RequestMapping("/rest") berfungsi untuk menandakan bahwa akan path yang akan masuk ke controller adalah path yang diawali dengan /rest. Terdapat 2 method pada controller ini yaitu view untuk melihat course dan viewall untuk melihat seluruh course yang ada. Cara kerjanya kurang lebih sama seperti view course dan viewall course pada controller biasa namun perbedaannya adalah return value dari method ini adalah objek yang akan ditampilkan dalam bentuk JSON.

Latihan 3:

Method code pada class StudentDAOImpl

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

Method ini akan meng-*consume* dari path "<http://localhost:8080/rest/student/viewall>" yang mana akan menghasilkan objek list of student sebagai return valuenya dan akan disimpan di variable students. Setelah itu return dari method ini adalah variable students yang sebelumnya sudah dijelaskan.

Method code pada class StudentServiceRest

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

Method ini akan memanggil method selectAllStudents() yang ada di StudentDAO untuk mendapatkan object list of student model. Return valuenya adalah students yang isinya sama dengan students pada method sebelum ini.

Latihan 4:

Course DAO

```
public interface CourseDAO {
    CourseModel selectCourse (String id);
}
```

Digunakan sebagai interface untuk CourseDAOImpl

CourseDAOImpl

```
package com.example.dao;

import com.example.model.CourseModel;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

@Service
public class CourseDAOImpl implements CourseDAO {
    @Autowired
    private RestTemplate restTemplate;

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

Pada class ini terdapat method selectCourse yang akan mencari course sesuai dengan id yang diminta. Terdapat variable course yang akan menyimpan hasil dari path "http://localhost:8080/rest/course/view/" +id sebagai object dengan menggunakan method getForObject dari restTemplate.

Interface CourseService

```
package com.example.service;

import com.example.model.CourseModel;

public interface CourseService {
    CourseModel selectCourse (String id);
    |
}
```

Pembuatan interface yang akan digunakan CourseServiceRest

Kode pada CourseServiceRest

```
package com.example.service;

import com.example.dao.CourseDAO;
import com.example.model.CourseModel;
import lombok.extern.slf4j.Slf4j;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

@Slf4j
@Service
public class CourseServiceRest implements CourseService{
    @Autowired
    private CourseDAO courseDAO;

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

Pada class ini akan dipanggil method selectCourse pada courseDAO dan akan disimpan dalam variabel course. Return valuenya adalah course yang dicari sesuai dengan id yang diminta