

## LATIHAN

**1. Service Producer untuk lihat semua student dengan mapping: `"/rest/student/viewall"`.**

Menambahkan method viewall seperti di bawah ini di kelas StudentRestController yang fungsinya mengambil semua data pada tabel student

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

JSON	Raw Data	Headers
Save	Copy	
▼ 0:		
npm:	"123"	
name:	"Abin"	
gpa:	3.09	
▼ courses:		
▼ 0:		
idCourse:	null	
name:	"MPKT"	
credits:	6	
students:	null	
▼ 1:		
npm:	"1234"	
name:	"Sisi"	
gpa:	2.45	
▼ courses:		
▼ 0:		
idCourse:	null	
name:	"DDP 1"	
credits:	4	
students:	null	

**2. Service Producer untuk lihat course dari idnya dengan mapping: `"/rest/course/view/{id}"`. Dan service lihat semua course dengan mapping: `"/rest/course/viewall"`.**

Menambahkan kelas `CourseRestController` dalam package `rest`, lalu membuat method yang memiliki fungsi yang sama seperti di kelas `StudentRestController`. Karena di kelas `CourseService` belum terdapat method `selectAllCourses()`, maka terjadi error.

```
1 package com.example.rest;|
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.web.bind.annotation.PathVariable;
7 import org.springframework.web.bind.annotation.RequestMapping;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import com.example.model.CourseModel;
11 import com.example.service.CourseService;
12
13 @RestController
14 @RequestMapping("/rest")
15 public class CourseRestController {
16     @Autowired
17     CourseService courseService;
18
19     @RequestMapping("/course/view/{id_course}")
20     public CourseModel view(@PathVariable (value="id_course") String id) {
21         CourseModel course = courseService.selectCourse(id);
22         return course;
23     }
24
25     @RequestMapping("/course/viewall")
26     public List<CourseModel> viewall() {
27         return courseService.selectAllCourses();
28     }
29 }
```

Membuat method `selectAllCourses()`.

```
1 package com.example.service;
2
3 import java.util.List;
4
5 import com.example.model.CourseModel;
6
7 public interface CourseService
8 {
9     CourseModel selectCourse (String id);
10
11     List<CourseModel> selectAllCourses();|
12 }
```

```

1  package com.example.service;
2
3  import java.util.List;
12
13  @Slf4j
14  @Service
15  public class CourseServiceDatabase implements CourseService
16  {
17      @Autowired
18      private CourseMapper courseMapper;
19
20
21      @Override
22      public CourseModel selectCourse (String id)
23      {
24          log.info ("select course with id_course {}", id);
25          return courseMapper.selectCourse (id);
26      }
27
28
29      @Override
30      public List<CourseModel> selectAllCourses() {
31          log.info ("select all courses");
32          return courseMapper.selectAllCourses ();
33      }
34
35  }

```

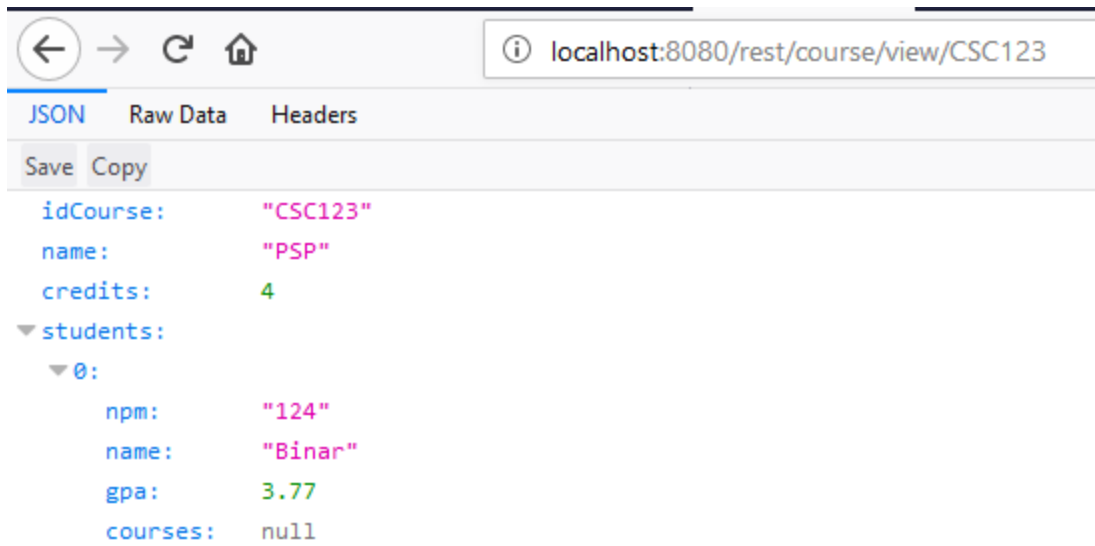
Menambahkan query untuk mengisi method selectAllCourses().

```

@Select("select id_course, name, credits from course")
@Results(value = {
    @Result(property="idCourse", column="id_course"),
    @Result(property="name", column="name"),
    @Result(property="credits", column="credits"),
    @Result(property="students", column="id_course",
        javaType = List.class,
        many=@Many(select="selectStudents"))
})
List<CourseModel> selectAllCourses();

```

"/rest/course/view/{id}"



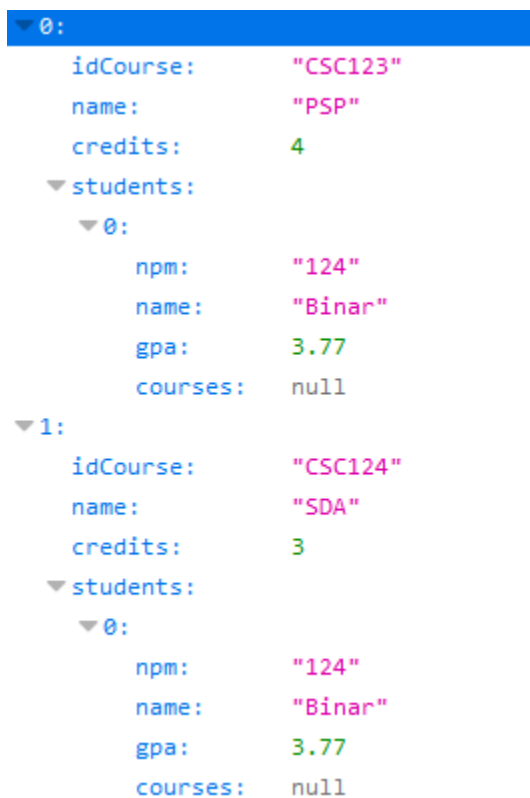
localhost:8080/rest/course/view/CSC123

JSON Raw Data Headers

Save Copy

```
{
  "idCourse": "CSC123",
  "name": "PSP",
  "credits": 4,
  "students": [
    {
      "npm": "124",
      "name": "Binar",
      "gpa": 3.77,
      "courses": null
    }
  ]
}
```

“/rest/course/viewall”



```
[
  {
    "idCourse": "CSC123",
    "name": "PSP",
    "credits": 4,
    "students": [
      {
        "npm": "124",
        "name": "Binar",
        "gpa": 3.77,
        "courses": null
      }
    ]
  },
  {
    "idCourse": "CSC124",
    "name": "SDA",
    "credits": 3,
    "students": [
      {
        "npm": "124",
        "name": "Binar",
        "gpa": 3.77,
        "courses": null
      }
    ]
  }
]
```

### 3. Service Consumer untuk lihat semua student dengan mapping: "localhost:9090/student/viewall".

Mengubah isi method selectAllStudents pada kelas StudentDAOImpl yang menggunakan kelas RestTemplate untuk meng-consume object REST *web service* menggunakan method getForObject yang menerima parameter berupa URL producer web service dan tipe class dari object yang didapatkan

```
@Service
public class StudentDAOImpl implements StudentDAO {
    @Autowired
    private RestTemplate restTemplate;

    @Override
    public StudentModel selectStudent(String npm) {
        StudentModel student = restTemplate.getForObject("http://localhost:8080/rest/student/view/" + npm,
            StudentModel.class);
        return student;
    }

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

Menambahkan return studentDAO.selectAllStudents() pada kelas StudentServiceRest di package service

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

No	NPM	Name	GPA	Cum laude	Delete	Update
1	123	Abin	3.09	Sangat Memuaskan	<a href="#">Delete</a>	<a href="#">Update</a>
2	1234	Sisi	2.45	Sangat Memuaskan	<a href="#">Delete</a>	<a href="#">Update</a>
3	124	Binar	3.77	Cum Laude!	<a href="#">Delete</a>	<a href="#">Update</a>

#### 4. Service Consumer untuk lihat course dari idnya dengan mapping:

"localhost:9090/course/view/{id}". Dan service lihat semua course dengan mapping:

"localhost:9090/course/viewall".

Menambahkan kelas CourseDAO yang berisi method yang memiliki fungsi yang sama seperti di kelas StudentDAO.

```
package com.example.dao;

import java.util.List;

public interface CourseDAO {
    CourseModel selectCourse (String id);

    List<CourseModel> selectAllCourses ();
}
```

Menambahkan kelas CourseDAOImpl yang mengimplementasi kelas CourseDAO.

```
package com.example.dao;

import java.util.List;

@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;
    }

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

Menambahkan kelas CourseServiceRest yang mengimplementasi CourseService. Karena di kelas CourseService belum ada method selectAllCourse, maka terjadi error.

```

package com.example.service;

import java.util.List;

@Slf4j
@Service
@Primary
public class CourseServiceRest implements CourseService {
    @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();
    }
}

```

Menambahkan method selectAllCourse() pada kelas CourseService.

```

package com.example.service;

import java.util.List;

import com.example.model.CourseModel;

public interface CourseService
{
    CourseModel selectCourse (String id);

    List<CourseModel> selectAllCourses();
}

```

Lalu karena menyebabkan error pada kelas **CourseServiceDatabase** dan kelas **CourseMapper**, maka kedua kelas tersebut dihapus.

Setelah itu menambahkan method viewall pada CourseController untuk tampilan lihat semua course.

```

@RequestMapping("/course/viewall")
public String viewall (Model model)
{
    List<CourseModel> courses = courseDAO.selectAllCourses ();
    model.addAttribute ("courses", courses);

    return "viewall-course";
}

```

Lalu membuat tampilan viewall-course.html.

“localhost:9090/course/view/{id}”



Navbar Home Daftar Mahasiswa Menambah Mahasiswa

ID = CSC123

Nama = PSP

SKS = 4.0

Mahasiswa yang mengambil

- 124-Binar

“localhost:9090/course/viewall”



Navbar Home Daftar Mahasiswa Menambah Mahasiswa

## All Courses

Show 10 entries

Search:

No	Name	SKS	Delete	Update
1	PSP	4.0	<a href="#">Delete</a>	<a href="#">Update</a>
2	SDA	3.0	<a href="#">Delete</a>	<a href="#">Update</a>
3	DDP 1	4.0	<a href="#">Delete</a>	<a href="#">Update</a>
4	MPKT	6.0	<a href="#">Delete</a>	<a href="#">Update</a>

Showing 1 to 4 of 4 entries

Previous **1** Next



Salah satu yang baru saya ketahui dari tutorial 7 ini adalah bagaimana pembagian pekerjaan antara frontend dan backend. Setelahnya, tentu saja saya belajar memahami bagaimana pembuatan web service ke dalam bentuk json. Saya juga mempelajari bagaimana menjadi consumer yang menggunakan data API dari producer.