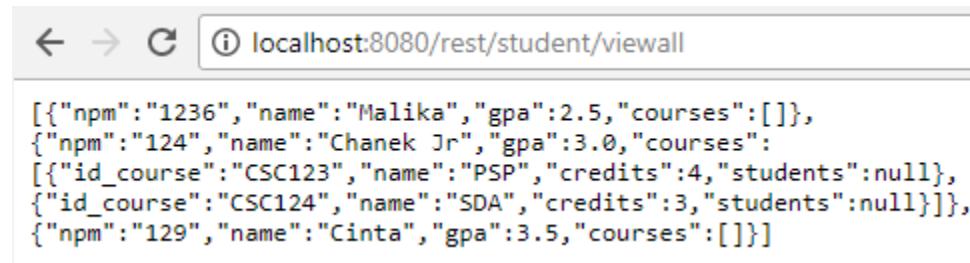


Write up 07

Latihan 1 : Membuat service untuk mengembalikan seluruh student yang ada di basis data. Method ini mengembalikan objek *List of students* yang didapat dari studentService.

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

Tampilan :



```
String parse
[
  {
    "npm": "1236",
    "name": "Malika",
    "gpa": 2.5,
    "courses": [
    ]
  },
  {
    "npm": "124",
    "name": "Chanek Jr",
    "gpa": 3.0,
    "courses": [
      {
        "id_course": "CSC123",
        "name": "PSP",
        "credits": 4,
        "students": null
      },
      {
        "id_course": "CSC124",
        "name": "SDA",
        "credits": 3,
        "students": null
      }
    ]
  },
  {
    "npm": "129",
    "name": "Cinta",
    "gpa": 3.5,
    "courses": [
    ]
  }
]
```

Latihan 2 : Buatlah service untuk class Course. Buatlah controller baru yang terdapat service untuk melihat suatu course dengan masukan ID Course (view by ID) dan service untuk melihat semua course (view all).

```

@RestController
@RequestMapping("/rest")
public class CourseRestController {
    @Autowired
    CourseService courseService;

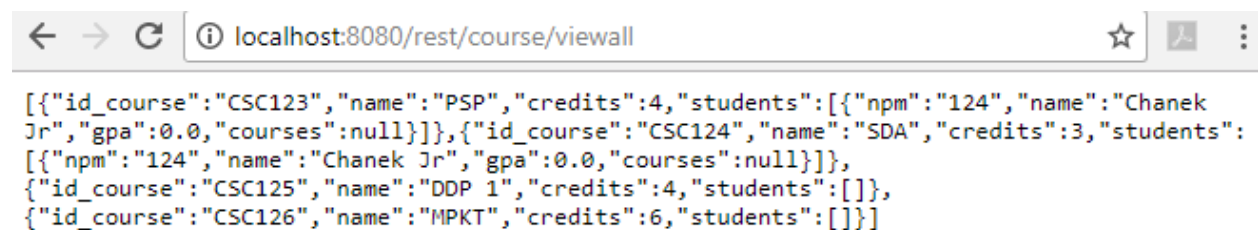
    @RequestMapping("/course/view/{id}")
    public CourseModel selectCourse (@PathVariable(value = "id") String id)
    {
        CourseModel course = courseService.selectCourse(id);
        return course;
    }

    @RequestMapping("/course/viewall")
    public List<CourseModel> selectAllCourse(){
        List<CourseModel> courses = courseService.selectAllCourse();

        return courses;
    }
}

```

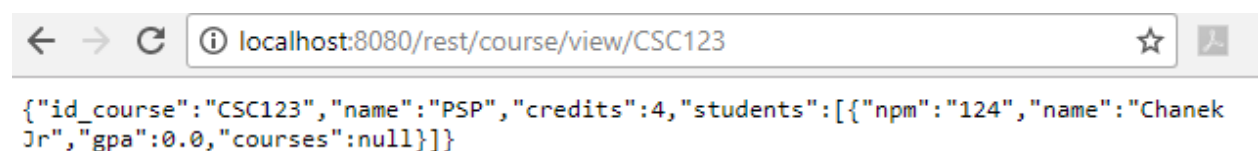
Tampilan ketika dijalankan :



```

[{"id_course":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek Jr","gpa":0.0,"courses":null}]},{ "id_course":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"Chanek Jr","gpa":0.0,"courses":null}]},{ "id_course":"CSC125","name":"DDP 1","credits":4,"students":[]},{ "id_course":"CSC126","name":"MPKT","credits":6,"students":[]}]

```



```

{"id_course":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek Jr","gpa":0.0,"courses":null}]}

```

Latihan 3 : Implementasikan service consumer untuk view all Students dengan melengkapi method selectAllStudents yang ada di kelas StudentServiceRest.

Pada consumer yang dipetakan bukan objek melainkan melalui url dari producer.

```

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

```

@SuppressWarnings digunakan untuk menghandle warning 'unchecked'

Method selectAllStudents akan mengambil dan mengembalikan daftar mahasiswa

```
@RestController
@RequestMapping("/rest")
public class StudentRestController {

    @Autowired
    StudentService studentService;

    @RequestMapping("/student/view/{npm}")
    public StudentModel view (@PathVariable(value = "npm") String npm) {
        StudentModel student = studentService.selectStudent(npm);
        return student;
    }

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

Latihan 4 : Implementasikan service consumer untuk class CourseModel dengan membuat class-class DAO dan service baru.

Membuat CourseServiceRest sebagai berikut

```
1 package com.example.tutorial07consumer.service;
2
3+ import java.util.List;
13 @Slf4j
14 @Service
15 @Primary
16 public class CourseServiceRest implements CourseService{
17     @Autowired
18     private CourseDAO courseDAO;
19
20     @Override
21     public CourseModel selectCourse(String id) {
22         log.info("select course with id {}", id);
23         return courseDAO.selectCourse(id);
24     }
25
26     @Override
27     public List<CourseModel> selectAllCourse() {
28         return courseDAO.selectAllCourses();
29     }
30
31 }
```

Menambahkan CourseDAOImpl sebagai berikut

```
1 package com.example.tutorial07consumer.service;
2
3 import java.util.List;
4
5
6
7
8
9
10
11 public class CourseDAOImpl implements CourseDAO {
12     @Autowired
13     private RestTemplate restTemplate;
14
15     @Override
16     public CourseModel selectCourse(String id) {
17         CourseModel course = restTemplate.getForObject(
18             "http://localhost:8080/rest/course/view/"+id, CourseModel.class);
19         return course;
20     }
21
22     @SuppressWarnings("unchecked")
23     @Override
24     public List<CourseModel> selectAllCourses() {
25         List<CourseModel> courses =
26             restTemplate.getForObject("http://localhost://8080/rest/course/viewall/", List.class);
27         return courses;
28     }
29 }
30
31
```

Lesson learned : Pada tutorial 7 ini saya mempelajari mengenai REST Web Service, serta bagaimana menghubungkan *backend* dengan *frontend*.