

Tutorial Membuat *Service Producer* (view)



Latihan Membuat *Service Producer*

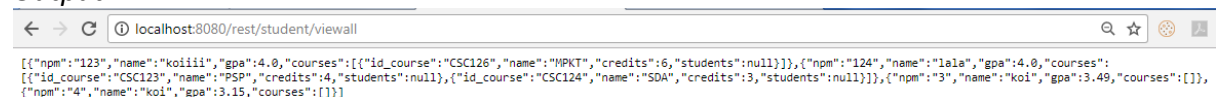
1. *ViewAll*

Saya menambahkan method `viewAll` di `StudentRestController`

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

Karena yang diminta adalah mengeluarkan semua data *student* maka tentunya *output* dalam bentuk *List* sehingga *method* yang saya buat tipenya *List* dengan *return* `List<StudentModel>` yang berisi data dari seluruh *student*.

Output :



String parse

```
{
  {
    "npm": "123",
    "name": "koiiii",
    "gpa": 4.0,
    "courses": [
      {
        "id_course": "CSC126",
        "name": "MPKI",
        "credits": 6,
        "students": null
      }
    ]
  },
  {
    "npm": "124",
    "name": "lala",
    "gpa": 4.0,
    "courses": [
      {
        "id_course": "CSC123",
        "name": "PSP",
        "credits": 4,
        "students": null
      },
      {
        "id_course": "CSC124",
        "name": "SDA",
        "credits": 3,
        "students": null
      }
    ]
  },
  {
    "npm": "3",
    "name": "koi",
    "gpa": 3.49,
    "courses": [
    ]
  },
  {
    "npm": "4",
    "name": "koi",
    "gpa": 3.15,
    "courses": [
    ]
  }
}
```

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).

a. *View by ID*

Output :



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



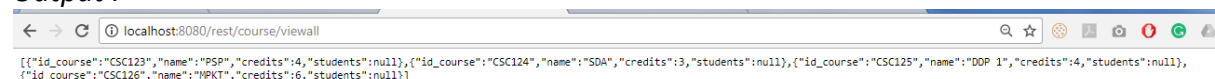
```
{
  "id_course": "CSC123",
  "name": "PSP",
  "credits": 4,
  "students": [
    {
      "npm": "124",
      "name": "lala",
      "gpa": 0.0,
      "courses": null
    }
  ]
}
```

Method yang saya buat untuk *view course by ID* sama seperti method *student* sebelumnya hanya saja ini dilakukan pada *course*. Ganti parameter menjadi *id_course*, ganti model menjadi *CourseModel* lalu pilih method *selectCourse* dengan menerima parameter *id_course*.

```
@RequestMapping("course/view/{id_course}")
public CourseModel viewCourseById(@PathVariable(value = "id_course") String id_course) {
    CourseModel course = studentService.selectCourse(id_course);
    return course;
}
```

b. *Viewall*

Output :



```
[ { "id_course": "CSC123", "name": "PSP", "credits": 4, "students": null }, { "id_course": "CSC124", "name": "SDA", "credits": 3, "students": null }, { "id_course": "CSC125", "name": "DOP 1", "credits": 4, "students": null }, { "id_course": "CSC126", "name": "MPKT", "credits": 6, "students": null } ]
```

String parse

```
[
  {
    "id_course": "CSC123",
    "name": "PSP",
    "credits": 4,
    "students": null
  },
  {
    "id_course": "CSC124",
    "name": "SDA",
    "credits": 3,
    "students": null
  },
  {
    "id_course": "CSC125",
    "name": "DDP 1",
    "credits": 4,
    "students": null
  },
  {
    "id_course": "CSC126",
    "name": "MPKT",
    "credits": 6,
    "students": null
  }
]
```

- Method yang saya buat di *Controller*, logic nya sama seperti *method viewall* di *student*, hanya tinggal saja menjadi *course*.

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

- Buat *method* ini di *Mapper*

```
@Select("select id_course, name, credits from course")
List<CourseModel> selectCourses ();
```

- Buat *method* ini di *StudentService*

```
List<CourseModel> selectCourses();
```

- Buat *method* ini di *StudentServiceDatabase*

```
@Override
public List<CourseModel> selectCourses ()
{
    log.info ("select all courses");
    return studentMapper.selectCourses ();
}
```