

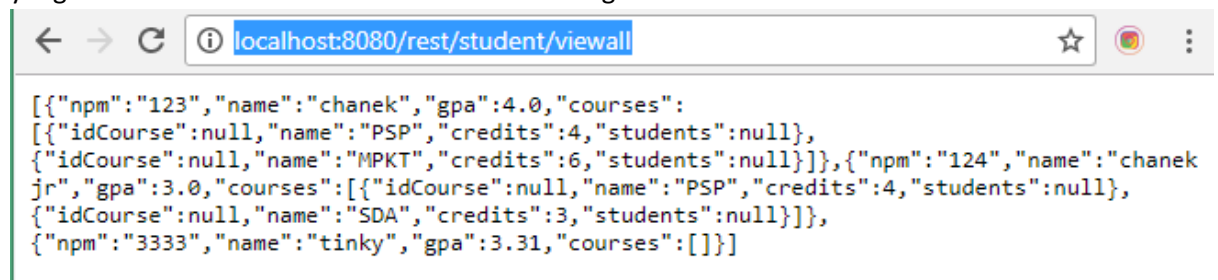
LATIHAN

1. Service untuk mengembalikan seluruh *student* yang ada di basis data

Untuk mengembalikan seluruh *student* yang ada di basis data, saya membuat *method* `viewAll()` yang mengembalikan *list of student* sebagai berikut :

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

Method tersebut memanfaatkan *method* `selectAllStudents()` pada *class* `StudentService` yang sudah dibuat sebelumnya untuk melakukan *read* seluruh *student* pada basis data. Tampilan yang dihasilkan dari *method* tersebut adalah sebagai berikut:



```
[{"npm": "123", "name": "chanek", "gpa": 4.0, "courses": [{"idCourse": null, "name": "PSP", "credits": 4, "students": null}, {"idCourse": null, "name": "MPKT", "credits": 6, "students": null}]}, {"npm": "124", "name": "chanek jr", "gpa": 3.0, "courses": [{"idCourse": null, "name": "PSP", "credits": 4, "students": null}, {"idCourse": null, "name": "SDA", "credits": 3, "students": null}]}, {"npm": "3333", "name": "tinky", "gpa": 3.31, "courses": []}]
```

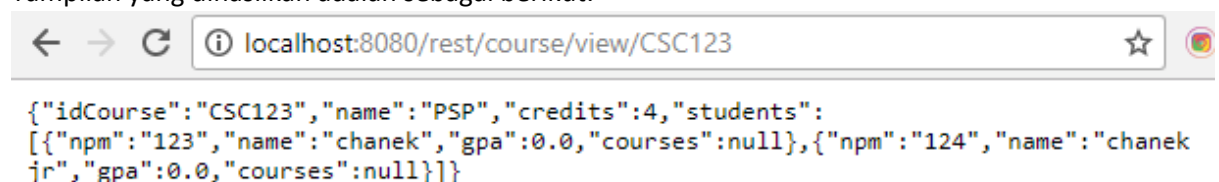
2. Service untuk *class* `Course`

- `view(id)`

Dalam *method* `view(id)`, saya mengimplementasikannya hampir mirip dengan `view(npm)` pada *class* `Student` sebelumnya. *Course* yang ingin ditampilkan akan di-*retrieve* menggunakan *method* `selectCourse(id)` yang telah diimplementasikan sebelumnya pada *class* `CourseService`. Berikut potongan kodenya:

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

Tampilan yang dihasilkan adalah sebagai berikut:



```
{"idCourse": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "123", "name": "chanek", "gpa": 0.0, "courses": null}, {"npm": "124", "name": "chanek jr", "gpa": 0.0, "courses": null}]}
```

- `viewAll()`

Sebelumnya pada tutorial 5, saya belum mengimplementasikan *service* untuk me-*retrieve* seluruh *course* dari *database*. Karena itu, pertama-tama yang saya lakukan adalah membuat akses ke *database* untuk membaca seluruh *course* yang ada di *database*.

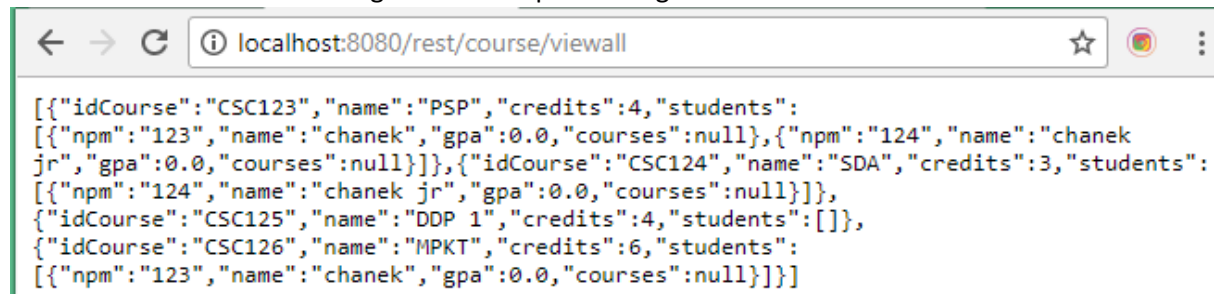
```
@Select("select id_course, name, credits from course")
@Results(value = {
    @Result(property="idCourse", column="id_course"),
    @Result(property="name", column="name"),
    @Result(property="credits", column="credits"),
})
```

```
@Result(property="students", column="id_course",
        javaType = List.class,
        many=@Many(select="selectStudents"))
})
List<CourseModel> selectAllCourses();
```

Setelah itu saya membuat *method* pengakses di *class* CourseService dan CourseServiceDatabase. Kemudian, pada CourseRestController saya membuat *method* viewAll() yang memanfaatkan *method* selectAllCourses pada CourseService yang baru saya buat tersebut. Berikut adalah *method* pada *class* CourseRestService:

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

Kode-kode tersebut akan menghasilkan tampilan sebagai berikut:




3. Implementasi *service consumer* untuk *view all* Students

Untuk memperoleh *viewAll* Students diperlukan *entity* untuk menampung *list of* StudentModel. Saya menggunakan tipe data ResponseEntity<T> untuk melakukan ini. *Generic type* T diisi dengan List<StudentModel>. Setelah menerima data dari *webservice* Tutorial07Producer, data akan ditampilkan. Potongan kode untuk melakukan *request* ke *webservice* adalah sebagai berikut:

```
@Override
public List<StudentModel> selectAllStudents() {
    ResponseEntity<List<StudentModel>> responseEntity =

        restTemplate.exchange("http://localhost:8080/rest/s
tudent/viewall",
        HttpMethod.GET,
        null,
        new ParameterizedTypeReference <List<StudentModel>>
        () {});
    List<StudentModel> students = responseEntity.getBody();
    return students;
}
```

Tampilan web akan menjadi sebagai berikut:

[←](#) [→](#) [↻](#) [localhost:9090/student/viewall](#) [☆](#)  [⋮](#)

- [Home](#)
- [Daftar Mahasiswa](#)
- [Menambah Mahasiswa](#)

All Students

No.	NPM	Name	GPA	Cum laude	Delete	Update
1	123	chanek	4.0	Cum Laude	Delete	Update
2	124	chanek jr	3.0	Tidak Cum Laude	Delete	Update
3	3333	tinky	3.31	Tidak Cum Laude	Delete	Update

Mata Kuliah APAP

4.