

- Yang saya pelajari adalah, bagaimana kerja antara **backend** dan **frontend**. Selama ini yang saya tahu kerja antara **backend** dan **frontend** hanya sebatas garis besarnya saja. **Backend** mengurus bagian servernya dan **frontend** mengurus bagian tampilannya, tapi saya tidak tahu bagaimana cara kerjanya. Tetapi di tutorial 7 ini saya jadi tahu bagaimana kerja antara **backend** dan **frontend**, bagaimana menghubungkan antara **backend** dan **frontend**

➤ Latihan 1

```
localhost:8080/rest/student/viewall

[{"npm":"123","name":"debian","gpa":3.7,"courses":
[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]}],{"npm":"124","name":"Chanek
alham","gpa":3.44,"courses":[{"idCourse":null,"name":"PSP","credits":4,"students":null},
{"idCourse":null,"name":"SDA","credits":3,"students":null}]}
```

Penjelasan : method ini mengembalikan object yang di inginkan dari *service*. Pada *case* ini *student service*

➤ Latihan 2

```
localhost:8080/rest/course/view/csc123

{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek alham","gpa":3.44,"courses":null}]}

localhost:8080/rest/course/viewall

[{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek alham","gpa":3.44,"courses":null}]}],{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"123","name":"Chanek
alham","gpa":3.44,"courses":null}]}],{"idCourse":"CSC125","name":"MPKT","credits":6,"students":
[{"npm":"123","name":"debian","gpa":3.7,"courses":null}]}
```


Penjelasan : method ini sama seperti latihan 1, tetapi object yang di kembalikan dari *course service*

➤ Latihan 3

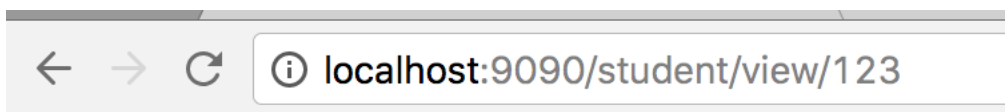
Penjelasan : pada latihan ini, selectAllStudent pada StudentServiceRest memanggil selectAllStudent() pada studentDAO, method selectAllStudent() pada class StudentDAOImpl adalah seperti berikut

```
@Override
public List<StudentModel> selectAllStudent() {
    ResponseEntity<List<StudentModel>> response = restTemplate.exchange(
        uri: "http://localhost:8080/rest/student/viewall/", HttpMethod.GET,
        new ParameterizedTypeReference<List<StudentModel>>() {
        });
    List<StudentModel> students = response.getBody();
    return students;
}
```

Result :



```
{"npm":123,"name":"debian","gpa":3.7,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]}
```



```
localhost:9090/student/view/123
```

[Navbar](#) ☐

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

NPM = 123

Name = debian

GPA = 3.7

Kuliah yang diambil

- MPKT - 6 sks

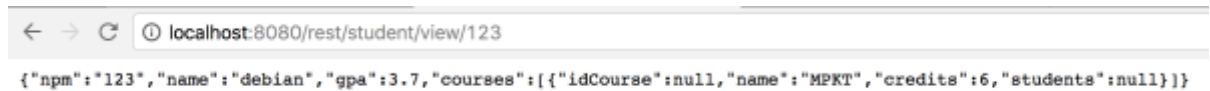
➤ Latihan 4

Penjelasan : method ini mirip dengan latihan 3, tetapi method ini di tujukan untuk *course*

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

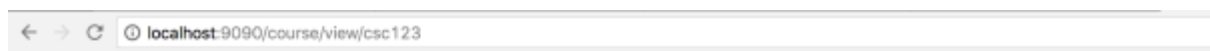
@Override
public List<CourseModel> selectAllCourse() {
    ResponseEntity<List<CourseModel>> response = restTemplate.exchange( url: "http://localhost:8080/rest/course/viewall/", HttpMethod.GET, requestEntity: null,
        new ParameterizedTypeReference<List<CourseModel>>() {
    });
    List<CourseModel> courses = response.getBody();
    return courses;
}
```

Result :



← → ↻ ⓘ localhost:8080/rest/student/view/123

```
{"npm":"123","name":"debian","gpa":3.7,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]}
```



← → ↻ ⓘ localhost:9090/course/view/csc123

[Navbar](#)

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

ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang Mengambil:

- 124 - Chanek alham

Welcome To Tutorial 7 APAP