

Latihan

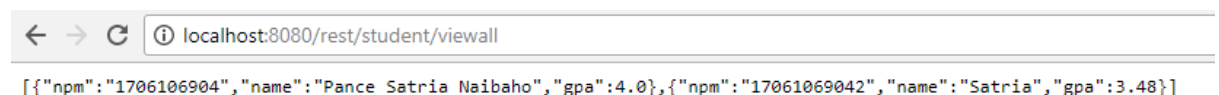
1. *Service* untuk mengembalikan seluruh *student*, *mapping* yang digunakan: "rest/student/viewall"

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

    return students;
}
```

Class **StudentService** memanggil method *selectAllStudents()* dan ditampung oleh *list* objek **StudentModel** dan kemudian objek tersebut di-*return*. Contoh tampilan ketika mengakses <http://localhost:8080/rest/student/viewall>:

*disini saya menggunakan database lokal



```
[{"npm": "1706106904", "name": "Pance Satria Naibaho", "gpa": 4.0}, {"npm": "17061069042", "name": "Satria", "gpa": 3.48}]
```

2. Mengimplementasikan *service consumer* untuk view all Students

```
@Override
public List<StudentModel> selectAllStudents() {
    ResponseEntity<List<StudentModel>> res =
        restTemplate.exchange("http://localhost:8080/rest/student/viewall",
            HttpMethod.GET, null,
            new ParameterizedTypeReference<List<StudentModel>>() {});

    HttpStatus statusCode = res.getStatusCode();
    HttpHeaders headers = res.getHeaders();

    log.info("REST - select all student - code status: "+statusCode);
    log.info("REST - select all student - content type: "+headers.getContentType());

    List<StudentModel> selectAllStudents = res.getBody();
    return selectAllStudents;
}
```

Penjelasan: **RestTemplate** digunakan untuk meng-*consume* object REST web service. Method yang digunakan adalah **exchange** yang menerima parameter berupa URL producer web service, Http Method yaitu **GET**, parameter null, dan tipe class dari objek yang didapatkan.

ResponseEntity adalah *extension* dari **HttpEntity** yang menambahkan **HttpStatus** kode status. Jadi kita bisa mendapatkan *status code* yang merupakan kembalian dari web service yang dituju (misal 201, 200, dll), juga kita bisa mendapatkan headers dari web service yang dituju. Untuk mendapatkan objek yang dikembalikan oleh web service maka digunakan method **getBody()**.

Dalam RestTemplate, class ResponseEntity dikembalikan dengan method `getForEntity()` dan `exchange()`.

Berikut contoh tampilan *log* yang mencetak *code status* dan *content type* web service yang dituju

```
\ntServiceRest      : REST - select all student
\OImpl              : REST - select all student - code status: 200
\OImpl              : REST - select all student - content type: application/json;charset=UTF-8
```

Rangkuman Hasil Pembelajaran:

Suatu sistem informasi berskala besar tidak hanya terdiri dari satu aplikasi saja. Sistem informasi tersebut terdiri dari beberapa aplikasi yang tersebar pada beberapa mesin server. Biasanya pengimplementasiannya adalah dengan memisahkan *layer backend (service producer)* dan *frontend (service consumer)*. *Service consumer* merupakan aplikasi yang berinteraksi dengan pengguna. Sementara *service producer* merupakan aplikasi yang memberikan data kepada service consumer berdasarkan apa yang service consumer minta.