

Tutorial 7 Web Service Menggunakan Spring Boot

Ringkasan dari materi / hal yang dipelajari

- Web service merupakan sebuah URL yang akan mengembalikan data dalam representasi yang telah disepakati kedua belah pihak. Namun dalam global use biasanya menggunakan JSON (JavaScript Object Notation). Biasanya web service menganut REST.
- Pada sisi penyedia web service, untuk membuat controller menggunakan REST web service dipakai *spring java annotation* `@RestController`, tidak seperti controller biasanya yang menggunakan anotasi `@Controller`. Untuk membedakan request mapping dengan controller biasa, kita bisa menambahkan path-nya contoh "REST" sehingga ketika diakses harus memasukkan awalan "REST" pada URL di browser ketika mengakses web service
- Anotasi `@Primary` dapat digunakan untuk menentukan implementasi dari sebuah interface mana yang akan dipakai.
- Pada sisi pengguna service, tidak perlu lagi menggunakan mapper, kita bisa memanfaatkan implementasi REST Template yang consume service melalui URL web service.

Latihan 1

Method baru yaitu `viewall` ditambahkan pada `StudentRestController.java` seperti berikut.

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

Method ini mempunyai return type `List` karena data student yang akan dikembalikan dari hasil pemanggilan method `selectAllStudents()` pada class `StudentService` lebih dari satu sehingga ditampung di dalam `List`. Ketika dijalankan di browser dengan URL "`http://localhost:8080/rest/student/viewall`" akan menampilkan JSON berisi seluruh student yang ada.

Latihan 2

Method baru yaitu `selectAllStudents` ditambahkan pada `StudentDAOImpl.java` seperti berikut.

```
@Override
public List selectAllStudents() {
    ResponseEntity response = restTemplate.getForEntity(
        "http://localhost:8080/rest/student/viewall",
        StudentModel[].class);
    return Arrays.asList(response.getBody());
}
```

Satrio Nugroho
1706106993

Method ini mempunyai return type `List` karena data seluruh student yang akan ditampilkan ditampung di dalam `List`. Kita tidak bisa menggunakan method `getForObject` untuk object yang berisi `List` karna yang akan di-*return* adalah `List of LinkedHashMaps` yang mewakili property data student padahal yang diharapkan `List of student`. Spring menyediakan method yang lebih generic yang dapat digunakan yaitu `getForEntity` tapi parameternya harus berupa array dalam hal ini berarti kita gunakan `Array of Student`, sehingga kita perlu convert ke list terlebih dahulu sebelum di-*return*.

Method ini menerima data dari *producer*, sehingga method ini adalah consumer dari JSON yang telah dibuat tadi. Pada metode seperti ini akan mempercepat *development speed* namun menambah stack dari suatu *system*, jika bertambah stack memungkinkan untuk bertambah lama suatu *system*. Karena metode ini tergantung pada *producer*.