Aisyah Sadiyah 1606954703

Tutorial 7
Membuat Web Service Menggunakan Spring Boot Framework

Review

Sistem informasi berskala besar tidak hanya terdiri dari satu aplikasi saja. Sistem informasi tersebut terdiri dari beberapa aplikasi yang tersebar pada beberapa mesin server. Dengan menyebar aplikasi tersebut ke mesin server yang berbeda, pengembang dapat memaksimalkan performa hardware masing-masing mesin server untuk aplikasi yang hanya di-deploy di server tersebut. Contohnya adalah pemisahan layer backend (service producer) dan frontend (service consumer). Service consumer merupakan aplikasi yang berinteraksi dengan pengguna. Sedangkan service producer merupakan aplikasi yang memberikan data kepada service consumer berdasarkan apa yang service consumer minta. Dengan pemisahan tanggung jawab ini, aplikasi service consumer dapat fokus untuk menyediakan dan mengolah data saja ke pengguna tanpa perlu memiliki database. Sementara service producer hanya bertanggung jawab untuk menyediakan data dari database dan biasanya tidak memiliki view yang dapat dilihat pengguna. Untuk dapat berkomunikasi dengan service consumer, service producer menyediakan web service yang dapat dikonsumsi oleh service consumer. Web service merupakan sebuah URL yang akan mengembalikan data dalam representasi yang telah disepakati seperti JSON atau XML.

Latihan

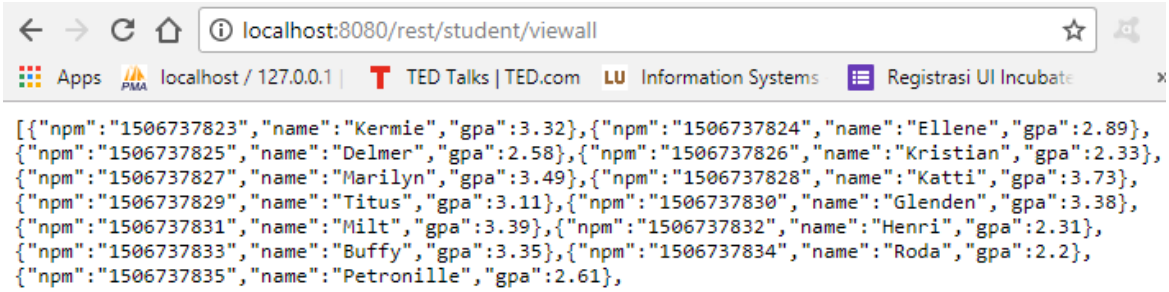
1. Latihan 1: Buatlah service untuk mengembalikan seluruh student yang ada di basis data. Service ini mirip seperti method viewAll di Web Controller. Service tersebut di-mapping ke "/rest/student/viewall".

   Pada class StudentRestController.java tambahkan method seperti di bawah ini. @RequestMapping("/rest") pada class header tersebut artinya semua request mapping di dalam kelas tersebut harus selalu diawali dengan "/rest". Pemanggilan service viewall Student di-mapping ke /rest/student/viewall. Method tersebut berfungsi mengambil data dari list student dan mengembalikan ke students.

   StudentRestController.java

```
@RequestMapping("/student/viewall")
public List<StudentModel> view (Model model)  //mengembalikan ke list student
{
    List<StudentModel> students = studentService.selectAllStudents (); //mengambil dari students
    return students;
}
```

Ini adalah tampilan outputnya



2. Latihan 2: Implementasikan service consumer untuk view all Students dengan melengkapi method selectAllStudents yang ada di kelas StudentServiceRest.
   Method di bawah ini akan mengembalikan kepada studentDAO selectAllStudents().
   StudentServiceRest.java

```
@Override
public List<StudentModel> selectAllStudents ()
{
    log.info("REST - select all students");
    return studentDAO.selectAllStudents();
}
```

Method di bawah ini menggunakan ResponseEntity untuk mengambil nilai objek yang dalam bentuk list dan akan dikembalikan dalam bentuk list.
StudentDAOImpl.java

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

Ini adalah tampilan outputnya