

Rangkuman Tutorial 7

Pada tutorial 7 ini mempelajari pengertian tentang web service dan bagaimana membangun web service dengan Spring. Web service memungkinkan integrasi antar aplikasi yang tersebar di banyak server. Dengan adanya web service antar aplikasi satu dengan yang lainnya dapat saling komunikasi atau bertukar data.

Dalam pengembangan aplikasi web service ada dua konsep pemisahan level aplikasi yaitu service consumer dan service producer. Hal ini seperti pengembangan pada aplikasi biasa dimana ada backend dan frontend untuk memudahkan bagi pengembang dalam mengembangkan aplikasi dan fokus pada layanan masing-masing level. Service consumer (frontend) ini merupakan aplikasi yang berinteraksi dengan pengguna. Service producer (backend) ini merupakan sebuah aplikasi yang menyediakan data untuk service consumer.

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"`

```
*StudentRestController.java
package com.example.rest;

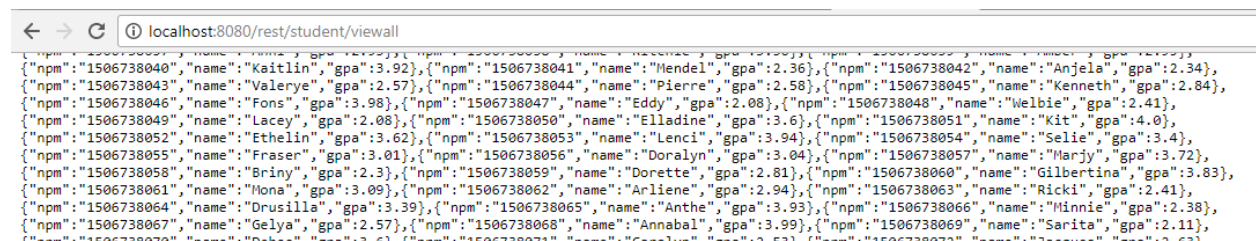
import java.util.List;

@RestController
@RequestMapping("/rest")
public class StudentRestController {

    @Autowired
    StudentService studentService;

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

Pada method viewall() ini sama saja seperti sebelumnya hanya saja berbeda pada method ini me-return data object List<StudentModel> .



Latihan 2

Implementasikan service consumer untuk view all Students dengan melengkapi method `selectAllStudents()` yang ada di kelas `StudentServiceRest`.

```
package com.example.service;

import java.util.List;

@Slf4j
@Service
@Primary
public class StudentServiceRest implements StudentService {

    @Autowired
    private StudentDAO studentDAO;

    @Override
    public StudentModel selectStudent(String npm) {
        log.info("REST - select student with npm {}", npm);
        return studentDAO.selectStudent(npm);
    }

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

Sebelum mengimplementasikan method `selectAllStudents()` di class `StudentServiceRest`, pertama adalah implementasi kode pada method `selectAllStudents()` tersebut yang ada pada class `StudentDAOImpl`. Di dalam method tersebut dimana akan me-return objek berupa `List<StudentModel>` hal ini berbeda dengan implementasi kode pada method `selectStudent()`. Untuk mendapatkan objek berupa list maka disini akan menggunakan class `ResponseEntity` dengan tipe data generiknya `StudentModel[]` (Array). Class ini berfungsi untuk membuat dan menampung respon data dari rest template. Method pada restTemplate yang digunakan adalah `getForEntity()` dimana method tersebut akan me-return objek `ResponseEntity`. Setelah itu re dari objek `ResponseEntity` akan menyimpan array dengan data `StudentModel`. Variable re akan diiterasikan untuk dimasukkan ke dalam List untuk dijadikan return method `selectAllStudents()`.

```
Tutorial07Consumer - ApapTutorial04Application [Spring Boot App] C:\Program Files\Java\jdk-8.0.144\bin\javaw.exe (Apr 14, 2018, 11:28:40 PM)
2018-04-14 23:28:47.731 INFO 5652 --- [main] com.example.ApapTutorial04Application : Started ApapTutorial04Application in 5.016 sec
2018-04-14 23:29:34.096 INFO 5652 --- [nio-9090-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring FrameworkServlet 'dispatcherServlet'
2018-04-14 23:29:34.101 INFO 5652 --- [nio-9090-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initiali
2018-04-14 23:29:34.180 INFO 5652 --- [nio-9090-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initiali
2018-04-14 23:29:34.357 INFO 5652 --- [nio-9090-exec-1] com.example.service.StudentServiceRest : REST - select student with npm 1506737824
2018-04-14 23:29:42.546 INFO 5652 --- [nio-9090-exec-2] com.example.service.StudentServiceRest : REST - select all students
2018-04-14 23:49:28.787 INFO 5652 --- [nio-9090-exec-8] com.example.service.StudentServiceRest : REST - select all students
```

Pada hasil di console berikut adalah hasil dimana “REST – select all student” merupakan log yang mepresentasikan pengambilan data semua student dari aplikasi service producer dengan URL `localhost:9090/rest/student/viewall`.

← → ↻

localhost:9090/student/viewall

☆

Navbar

Home

Daftar Mahasiswa

Menambah Mahasiswa

Search

Search

View All Student

No	NPM	Name		Cum Laude	Delete	Update
1	1506737823	Kermie	3.99	Cum Laude!	Update Data	Delete Data
2	1506737824	Ellene	2.89	Sangat Memuaskan!	Update Data	Delete Data
3	1506737825	Delmer	2.58	Sangat Memuaskan!	Update Data	Delete Data
4	1506737827	Marilyn	3.49	Cum Laude!	Update Data	Delete Data
5	1506737828	Katti	3.73	Cum Laude!	Update Data	Delete Data
6	1506737829	Titus	3.11	Sangat Memuaskan!	Update Data	Delete Data
7	1506737830	Glenden	3.38	Sangat Memuaskan!	Update Data	Delete Data
8	1506737831	Milt	3.39	Sangat Memuaskan!	Update Data	Delete Data
9	1506737832	Henri	2.31	Sangat Memuaskan!	Update Data	Delete Data

Berikut merupakan hasil tampilan yang berhasil mengambil data semua student di aplikasi service producer.