



Pada Tutorial ini, akan membahas mengenai penerapan web *service*, Web *service* merupakan sebuah URL yang akan mengembalikan data dalam representasi yang telah disepakati seperti JSON atau XML. Dalam tutorial ini Anda akan menggunakan JSON sebagai representasi data tersebut. Pada tutorial ini terdapat 2 *service* yaitu *service consumer* dan *service producer*. Ini di gunakan untuk memisahkan layer backend (*service producer*) dan frontend (*service consumer*). Dimana untuk *service provider* adalah backend yang memberikan data ataupun informasi kepada *service consumer*. *Service consumer* merupakan interaksi langsung kepada user. Ini juga merupakan implementasi nyata dari *service producer*. *Service producer* hanya menyediakan data-data dari database. Selain itu pada tutorial ini kita juga mengetahui bagaimana penerapan *service* antar backend dan front end pada pembuatan aplikasi. Disini akan membuat suatu fitur dari aplikasi yaitu view dan view all. Dan akan terlihat jelas bagaimana tampilan untuk kedua fitur ini dari sisi backend dan front end.

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"`

Program

```
StudentRestController.java application.properties StudentController.java
public class StudentRestController {
    @Autowired
    StudentService studentService;

    @RequestMapping("/student/view/{npm}")
    public StudentModel view (@PathVariable(value = "npm") String npm){
        StudentModel student = studentService.selectStudent(npm);
        return student;
    }

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

Output



```
← → ↺ 🏠 ⓘ localhost:8080/rest/student/viewall

▼ [
  ▼ {
    "npm": "1506737823",
    "name": "Kermie",
    "gpa": 3.32
  },
  ▼ {
    "npm": "1506737824",
    "name": "Ellene",
    "gpa": 2.89
  },
  ▼ {
    "npm": "1506737825",
    "name": "Delmer",
    "gpa": 2.58
  },
  ▼ {
    "npm": "1506737826",
    "name": "Kristian",
    "gpa": 2.33
  },
  ▼ {
    "npm": "1506737827",
    "name": "Marilyn",
    "gpa": 3.49
  },
  ▼ {
    "npm": "1506737828",
    "name": "Katti",
    "gpa": 3.73
  },
  ▼ {
    "npm": "1506737829",
    "name": "Titus",
    "gpa": 3.11
  },
  ▼ {
    "npm": "1506737830",
    "name": "Glenden",
    "gpa": 3.38
  },
],
```

Membuat *method* viewall dari sisi tampilan backend. Untuk membuat *method* ini dengan menerapkan web *service* maka sebelumnya membuat suatu class baru yaitu **StudentRestController** pada package **com.example.rest** sesuai dengan langkah-langkah tutorial ini. Kemudian untuk implementasikan viewall, menambahkan suatu *method* untuk request mapping `/student/viewall`, *method* ini akan berupa suatu list yang akan menampilkan seluruh data mahasiswa dan mengembalikan suatu object student.



Latihan 2.

Program

```
StudentDAO.java StudentDAOImpl.java application.properties application.properties StudentServiceRest.java

@Service
public class StudentDAOImpl implements StudentDAO{

    @Autowired
    private RestTemplate restTemplate;
    // @Lazy

    // @Bean
    // public RestTemplate restTemplate(RestTemplateBuilder builder) {
    //     return builder.build();
    // }

    @Override
    public StudentModel selectStudent (String npm)
    {
        StudentModel student = restTemplate.getForObject("http://localhost:8080/rest/student/view/"
+ npm, StudentModel.class);
        return student;
    }

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

Output



Tutorial 7
CSIM603026 – Arsitektur dan Pemrograman
Aplikasi Perusahaan
Semester Genap 2017/2018

Navbar Home Daftar Mahasiswa Menambah Mahasiswa						Search	Search
All Students							
Show 10 entries		Search:					
No	NPM	Name	GPA	Cum Laude			
1	1506737823	Kermie	3.32	Sangat Memuaskan	Update	Delete	
2	1506737824	Ellene	2.89	Sangat Memuaskan	Update	Delete	
3	1506737825	Delmer	2.58	Sangat Memuaskan	Update	Delete	
4	1506737826	Kristian	2.33	Sangat Memuaskan	Update	Delete	
5	1506737827	Marilyn	3.49	Sangat Memuaskan	Update	Delete	
6	1506737828	Katti	3.73	Cum Laude!	Update	Delete	
7	1506737829	Titus	3.11	Sangat Memuaskan	Update	Delete	
8	1506737830	Glenden	3.38	Sangat Memuaskan	Update	Delete	
9	1506737831	Milt	3.39	Sangat Memuaskan	Update	Delete	
10	1506737832	Henri	2.31	Sangat Memuaskan	Update	Delete	

Pada latihan 2 ini, adalah menerapkan web *service* untuk melakukan komunikasi antara *service producer* dan *service consumer*. Sebelumnya telah di buat *service producer* untuk *method* view all kemudian akan di implementasikan *method* tersebut pada *service producer* untuk dapat membuat tampilan interaksi yang akan digunakan user seperti gambar diatas. Dengan menambahkan *method* view all, yang berisi suatu list dari data-data student dan akan menggunakan *restTemplate* untuk dapat berkomunikasi dengan *service producer* dengan menambahkan link dari *service producer* ke *service consumer* untuk *method* view all, ini berguna untuk mengambil data-data dari database pada *service producer*. Dan akan mengembalikan object dari list student.