

Membuat Web Producer

Ringkasan: pada tutorial membuat web producer ini kita belajar mengenai bagaimana suatu rest controller dapat mengembalikan data dengan format json. Untuk membuat *rest controller* kita hanya perlu menambahkan kode program seperti berikut ini sebelum nama class controller.

```
@RestController
@RequestMapping("/rest")
```

Dalam tutorial ini kita membuat dua buah method yaitu method untuk menampilkan mahasiswa berdasarkan npm dan pada latihan kita menambahkan method untuk menampilkan seluruh data mahasiswa. Untuk melakukan kedua hal itu pada rest controller kita perlu menambahkan kode program seperti berikut ini.

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

Terlihat bahwa tidak ada perbedaan yang berarti dengan view/viewall yang kita pelajari pada tutorial 5. Perbedaan yang signifikan adalah jenis data yang dikembalikan yaitu json.

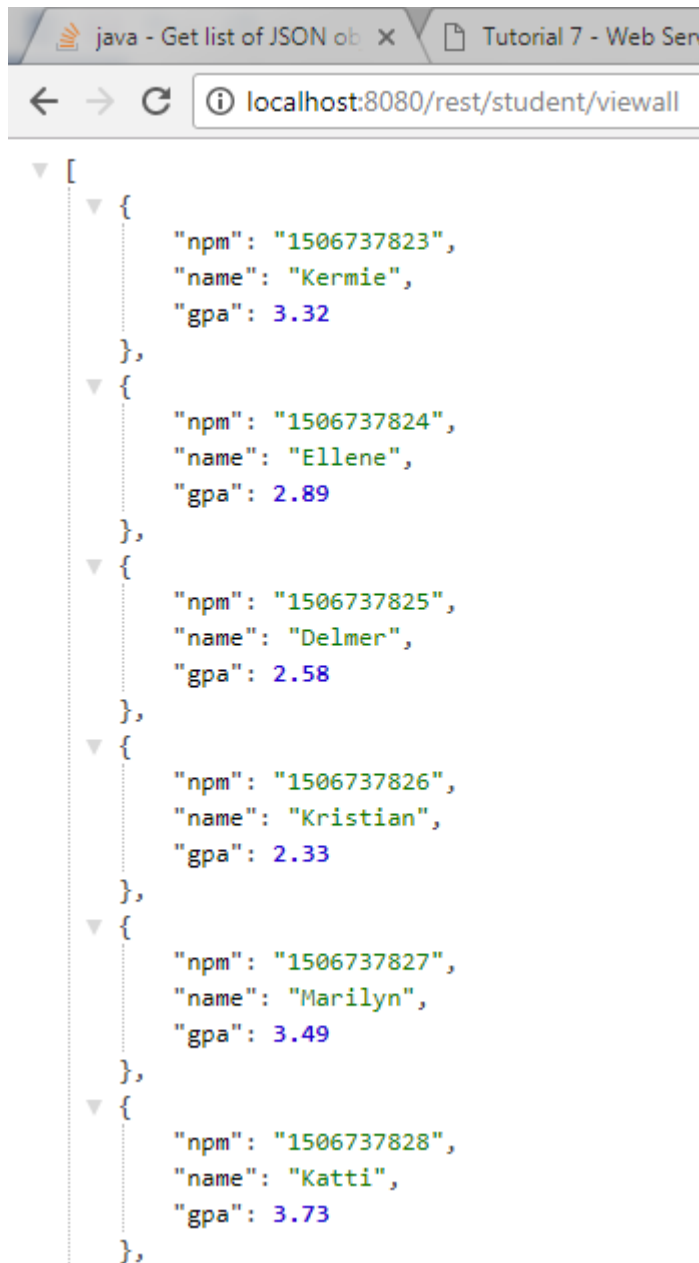
Latihan:

1. Buatlah method pada controller **studentRestController** yang terdapat pada package **com.example.rest**.

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

kita menggunakan mapper dan service yang sebelumnya dipakai untuk tutorial 4 dan 5 yaitu **selectAllStudents**.

2. Lakukan pengetesan pada web browser

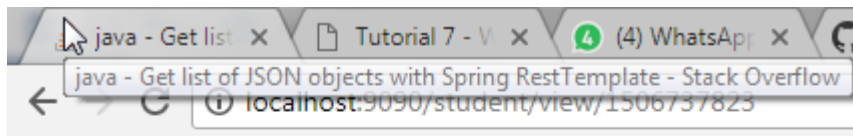


Membuat Web Consumer

Ringkasan: Service consumer adalah bagaimana kita memanfaatkan data yang telah dibuat pada latihan sebelumnya service producer. Meskipun menggunakan file dari tutorial 5 namun kita tidak menggunakan **StudentMapper**. Fungsi dari studentMapper akan digantikan oleh studentDAOImpl.

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

Pada method `selectStudent` di `StudentDAOImpl` kita memanggil url dari service producer yang telah kita buat sebelumnya. Kemudian kita return sebagai `StudentModel`, pada view akan terlihat seperti ini.



NPM = 1506737823

Name = Kermie

GPA = 3.32

Latihan:

1. Tambahkan method pada controller StudentDAOImpl

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

Method ini memanggil rest controller viewall yang ada pada serviceProducer. Karena tipe data yang dikembalikan oleh rest controller adalah berupa list maka perlu dilakukan konversi agar return yang dihasilkan sesuai.

2. Tes pada web browser

Navbar
Home
Daftar Mahasiswa
Menambah Mahasiswa
Search
Search

All Students

Show 10 entries

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	Cum Laude!	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