

## Tutorial 7

Pada kesempatan kali ini kita mempelajari bagaimana cara penggunaan sebuah Web Service dengan menggunakan 2 Project yang berbeda. Project yang dimaksud adalah Producer sebagai penyedia data & Customer yang akan mengambil data menggunakan Rest Template pada ServiceRest yang diajarkan pada tutorial tersebut. Hal yang harus diperhatikan bahwa anotasi `@Primary` yang digunakan pada Service Rest akan me autowired otomatis menginstansiasi `StudentService` menggunakan `StudentServiceRest` karena dia primary.

### 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"`

Tambahkan Method dibawah ini pada service yang akan mapping pada `/student/viewall` :

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

    return students;
}
```

Maka akan muncul Output sbb :

The screenshot shows two browser windows. The top window is a Chrome browser at the address `localhost:8080/rest/student/viewall`. The address bar shows a star icon, a globe icon, and a menu icon. The page content displays a JSON array: `[{"npm": "123", "name": "Yogine", "gpa": 3.5}, {"npm": "2", "name": "Adjie", "gpa": 2.2}]`.

The bottom window is the 'Json Parser Online' website (`json.parser.online.fr`). It has a header with 'Json Parser Online', 'You like it? Support it! Donate', and a 'Try out Beta!' button. The page is divided into two main sections: 'String parse' and 'JS eval'. The 'String parse' section contains the same JSON array as the top browser window. The 'JS eval' section shows the JSON array converted into a JavaScript object structure: `{ "npm": "123", "name": "Yogine", "gpa": 3.5 }, { "npm": "2", "name": "Adjie", "gpa": 2.2 }`. The footer of the website includes 'Author: Oliver Corneil' and links for 'FAQ', 'Privacy Policy', 'Changelog', 'Donation via PayPal', and 'Donation in Bitcoins'.

## Latihan 2

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

Tambahkan method dibawah pada `StudentServiceRest` :

```
@Override
public List<StudentModel> selectAllStudents() {
    // TODO Auto-generated method stub
    Log.info("REST - select all students");
    return studentDAO.selectAllStudent();
}
```

Dimana method tersebut akan mengembalikan kepada `StudentDAO.SelectAllStudent()`;

Buat lah method `selectAllStudent()` pada `StudentDaoImpl.java` :

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

Berikut hasil output menggunakan Port:9090



No.	NPM	Name	GPA	Status	Delete Data	Update Data
1	123	Yogine	3.5	Cum Laude!	<a href="#">Delete Data</a>	<a href="#">Update Data</a>
2	2	Adjie	2.2	Sangat memuaskan!	<a href="#">Delete Data</a>	<a href="#">Update Data</a>

Mata Kuliah APAP