

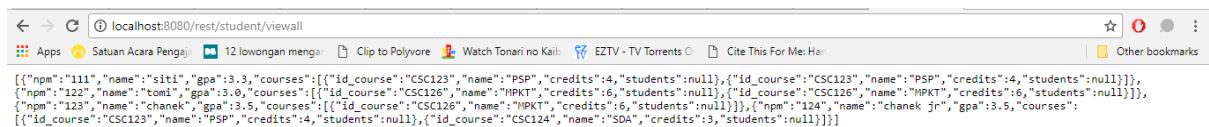
Hal yang telah dipelajari

Pada tutorial kali ini, saya mempelajari bagaimana memisah antara layer backend (service producer) dan frontend (service consumer). Dimana pada backend hanya khusus untuk mengelola data-data dari database saja, sedangkan untuk frontend hanya perlu membuat tampilan tanpa harus berurusan dengan pengohan data dari database. Saya juga mempelajari bagaimana cara menghubungkan antara backend dengan frontend dengan web service pada spring boot.

Latihan 1

Menambahkan method viewall pada clas StudentRestController dengan requestmapping student/viewall.

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

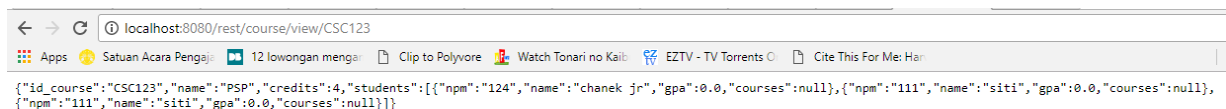


Method ini mengembalikan object yang diinginkan oleh student service dari Producer. Bentuk yang dikembalikan dari method ini dengan format json.

Latihan 2

Menambahkan method viewCourse pada clas StudentRestController dengan requestmapping course/view/{id}.

```
@RequestMapping ( "/course/view/{id}")
public CourseModel viewCourse ( @PathVariable ( value = "id" ) String id) {
    CourseModel course = courseService.selectCourse(id);
    return course;
}
```



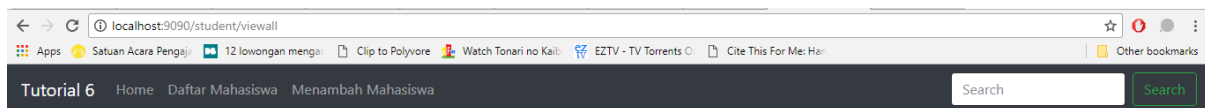
Method ini mengembalikan object yang diinginkan oleh course service dari Producer. Bentuk yang dikembalikan dari method ini dengan format json.

Latihan 3

```
@Override
public List <StudentModel> selectAllStudents ()
{
    List<StudentModel> students =
    restTemplate.getForObject ("http://localhost:8080/rest/student/viewall", List.class );
    return students;
}
```

Pada latihan 3 ini berasal dari service consumer sehingga pada latihan ini akan memberi tampilan terhadap object yang dihasilkan pada latihan 1. Saya menambahkan method selectAllStudents() membuat list object yang memanggil link yang berasal dari service producer pada class StudentDAOImpl dimana method tersebut telah diinisiasi lebih dulu di interface StudentDAO.

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



All Students

Show entries

Search:

No	NPM	Name	GPA	Cumlaude	Action
1	111	siti	3.3	Sangat Memuaskan!	<button>Delete Data</button> <button>Update Data</button>
2	122	tomi	3.0	Sangat Memuaskan!	<button>Delete Data</button> <button>Update Data</button>
3	123	chanek	3.5	Cumlaude!	<button>Delete Data</button> <button>Update Data</button>
4	124	chanek jr	3.5	Cumlaude!	<button>Delete Data</button> <button>Update Data</button>

Showing 1 to 4 of 4 entries

Previous **1** Next

Crafted with by Dayana Sheilla Agathy © 2017. Tutorial 6 Arsitektur dan Perancangan Aplikasi Perusahaan.

Berikut hasil keluaran dari method tersebut.

Latihan 4

Pada latihan 4 diminta untuk menampilkan semua course yang ada di database. Pertama saya membuat interface CourseDAO dan class CourseDAOImpl. Class CourseDAOImpl mengimplementasikan interface CourseDAO. Kemudian menambahkan method selectAllCourses().

```
@Override
public List <CourseModel> selectAllCourses ()
{
    List<CourseModel> courses = restTemplate.getForObject ("http://localhost:8080/rest/course/viewall", List.class );
    return courses;
}
```

Kemudian menambahkan method `selectAllCourses()` pada class `CourseService`.

```
public List < CourseModel > selectAllCourses ()
{
    Log.info ( "REST - select all courses" );
    return courseDAO.selectAllCourses();
}
```

Kemudian menambahkan method `viewallCourses` pada `courseController`

```
@RequestMapping("/course/viewall")
public String viewallCourse (Model model)
{
    List<CourseModel> courses = courseDAO.selectAllCourses ();
    model.addAttribute ("courses", courses);
    model.addAttribute("title", "Lihat Semua Course");
    return "courses";
}
```

Kemudian menambahkan file html untuk view `viewall courses`

```
<title>Daftar Course</title>
<div th:replace="fragments/header"></div>
<br />
<div class="container">
    <h3>All Courses</h3>
    <br />

    <div th:each="course, iterationStatus: ${courses}">
        <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
        <h3 th:text="'Id course = ' + ${course.id_course}">Student NPM</h3>
        <h3 th:text="'Name = ' + ${course.name}">Student Name</h3>
        <h3 th:text="'Credit = ' + ${course.credits}">Student GPA</h3>

        <h3>Mahasiswa yang mengambil</h3>
        <ul th:each="student, iterationStatus: ${course.students}">
            <li th:text="${student.npm} + '-' + ${student.name} " >
                Nama kuliah-X sks
            </li>
        </ul>
    </div>
</div>
<div th:replace="fragments/footer"></div>
```