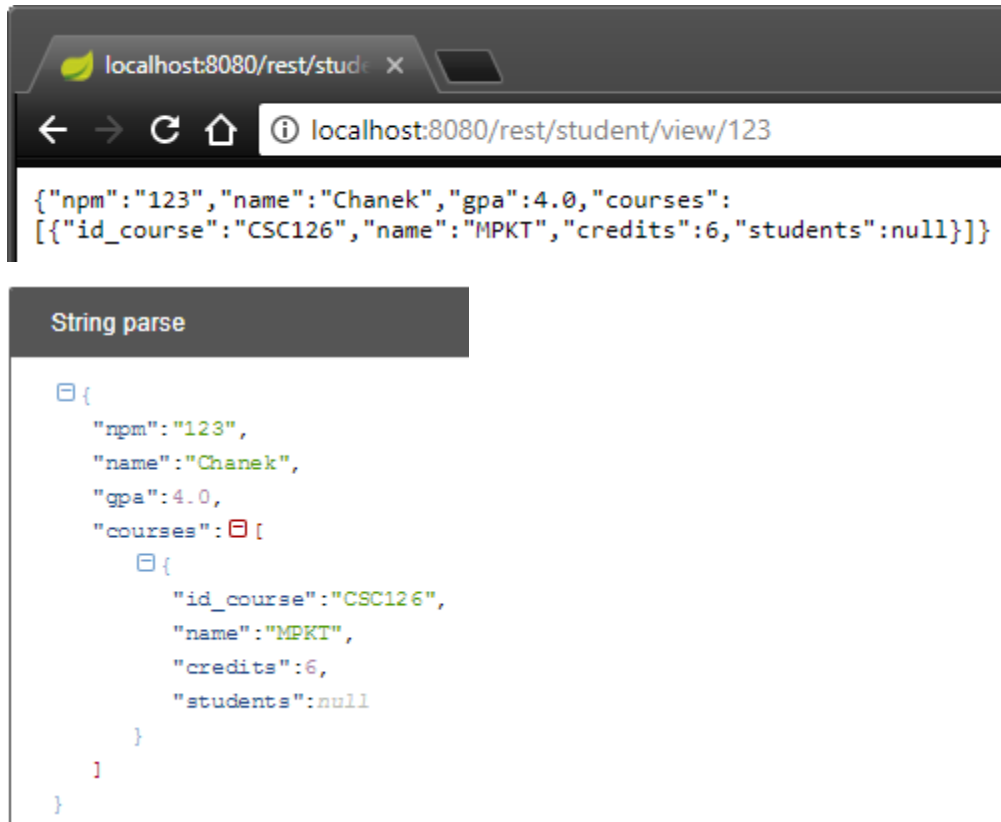


Membuat Web Service Menggunakan Spring Boot Framework

Membuat Service Producer

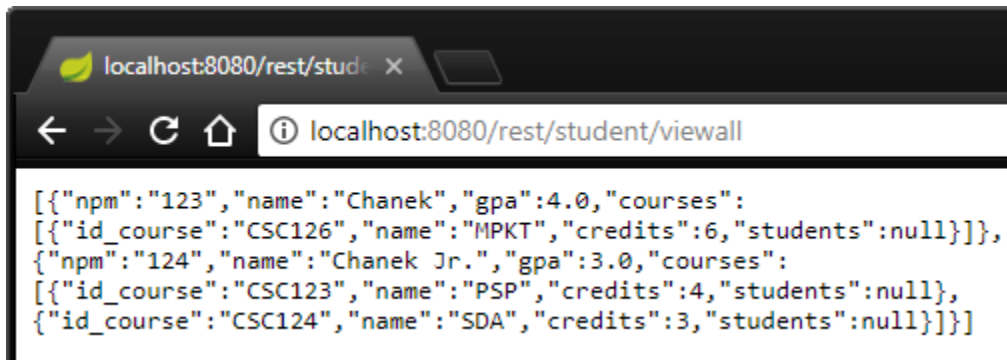
Hasil tampilan “localhost:8080/rest/student/view/123”



Latihan

1. **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”.

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



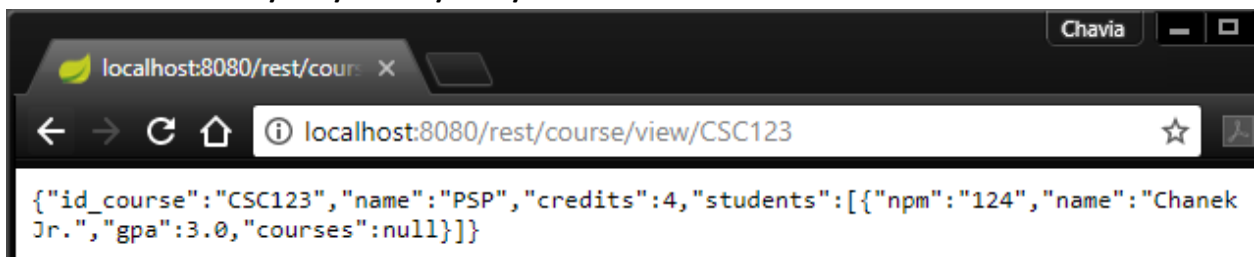
2. **Latihan 2:** Buatlah service untuk class Course. Buatlah controller baru yang terdapat service untuk melihat suatu course dengan masukan ID Course (view by ID) dan service untuk melihat semua course (view all).

```
@RestController
@RequestMapping("/rest")
public class CourseRestController
{
    @Autowired
    CourseService courseService;

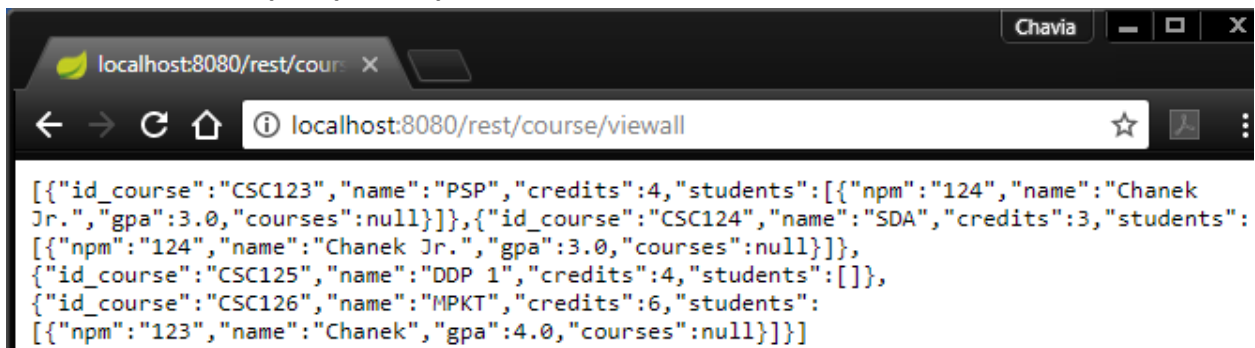
    @RequestMapping("/course/view/{id_course}")
    public CourseModel view(@PathVariable(value = "id_course") String id_course) {
        CourseModel course = courseService.selectCourse(id_course);
        return course;
    }

    @RequestMapping("/course/viewall")
    public List<CourseModel> viewCourse (Model model) {
        List<CourseModel> courses = courseService.selectAllCourses();
        return courses;
    }
}
```

Hasil localhost:8080/rest/course/view/CSC123

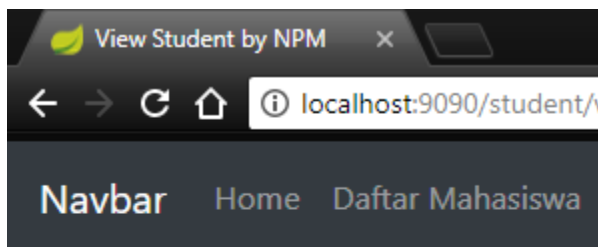


Hasil localhost:8080/rest/course/viewall



Membuat Service Consumer

Hasil tampilan dan *console* “localhost:9090/student/view/123”



NPM = 123

Name = Chanek

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

2017-11-04 12:35:51.837 INFO 7732 --- [nio-9090-exec-1] com.example.service.StudentServiceRest : REST - select student with npm 123

Latihan

1. **Latihan 3:** Implementasikan service consumer untuk view all Students dengan melengkapi method `selectAllStudents` yang ada di kelas `StudentServiceRest`.

Pada file `StudentDAOImpl.java`:

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

Pada file `StudentServiceRest.java`:

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

Hasil tampilan dan *console* **"localhost:9090/student/viewall"**

No	NPM	Name	GPA	Cum Laude	Delete	Update
1	123	Chanek	4.0	Yes	Delete Data	Update Data
2	124	Chanek Jr.	3.0	No	Delete Data	Update Data

```
2017-11-04 12:48:40.554 INFO 7732 --- [nio-9090-exec-5] com.example.service.StudentServiceRest : REST - select all students
```

2. **Latihan 4:** Implementasikan service consumer untuk class CourseModel dengan membuat class-class DAO dan service baru.

Pada file CourseDAOImpl.java:

```
@Override
public CourseModel selectCourse(String idCourse) {
    CourseModel course =
restTemplate.getForObject("http://localhost:8080/rest/course/view/" + idCourse,
        CourseModel.class);
    return course;
}

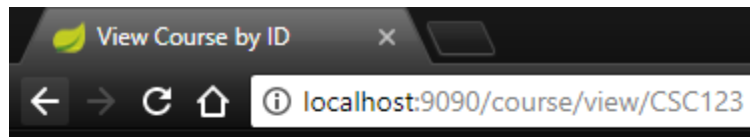
@Override
public List<CourseModel> selectAllCourses() {
    List<CourseModel> courses =
restTemplate.getForObject("http://localhost:8080/rest/course/viewall",
        List.class);
    return courses;
}
```

Pada file CourseServiceRest.java:

```
@Override
public CourseModel selectCourse (String id_course)
{
    Log.info ("REST - select course with id {}", id_course);
    return courseDAO.selectCourse (id_course);
}

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

Hasil tampilan dan *console* “localhost:9090/course/view/CSC123”



ID = CSC123

Name = PSP

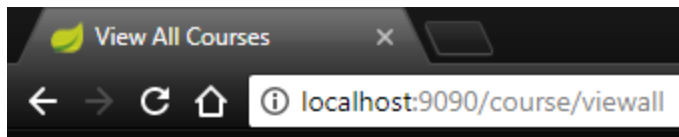
GPA = 4

Mahasiswa yang mengambil

- 124 - Chanek Jr.

2017-11-04 12:49:18.945 INFO 7732 --- [io-9090-exec-10] com.example.service.CourseServiceRest : REST - select course with id CSC123

Hasil tampilan dan *console* “localhost:9090/course/viewall”



All Courses

No. 1

id course = CSC123

Name = PSP

Credits = 4

Mahasiswa yang mengambil

- 124 - Chanek Jr.
-

No. 2

id course = CSC124

Name = SDA

Credits = 3

Mahasiswa yang mengambil

- 124 - Chanek Jr.
-

No. 3

Lesson Learned

Hal yang saya pelajari pada Tutorial 7 adalah beberapa hal, yaitu implementasi pemisahan *layer backend (service producer)* dan *layer frontend (service consumer)*. Dengan adanya pemisahan tanggung jawab ini, sebuah aplikasi *service consumer* dapat fokus untuk menyediakan dan mengolah data saja ke pengguna tanpa perlu memiliki database. Sementara *service producer* hanya bertanggung jawab untuk menyediakan data dari *database* dan biasanya tidak memiliki view yang dapat dilihat pengguna. Untuk dapat berkomunikasi dengan *service consumer*, *service producer* menyediakan web service yang dapat dikonsumsi oleh *service consumer*.