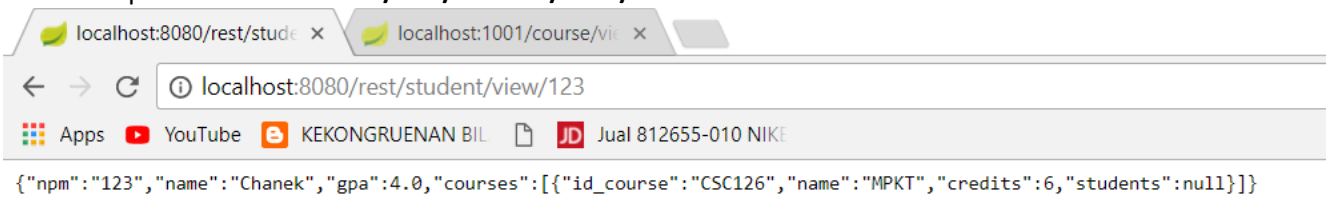
Galuh Octavia Chrisdianti
1506757844
APAP – A

Membuat Web Service Menggunakan Spring Boot Framework

Membuat Service Producer

Hasil tampilan **"localhost:8080/rest/student/view/123"**

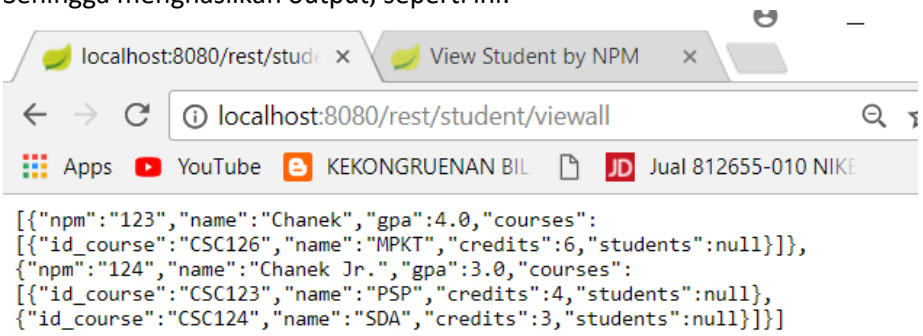


Latihan

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

Sehingga menghasilkan output, seperti ini:

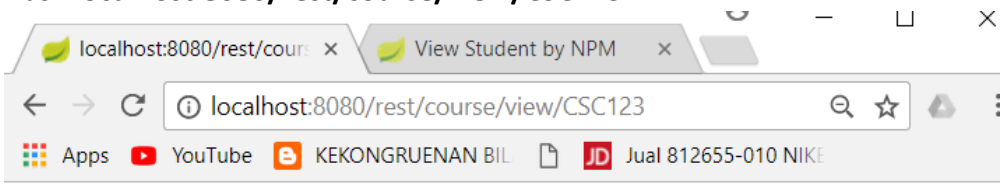


Galuh Octavia Chrisdianti
1506757844
APAP – A

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).

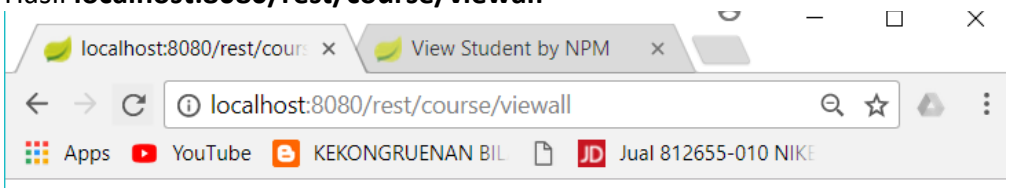
```
@RestController
@RequestMapping("/rest")
public class CourseRestController
{
    @Autowired
    CourseService courseService;

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

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



{"id_course":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek Jr.","gpa":3.0,"courses":null}]}

Hasil **localhost:8080/rest/course/viewall**

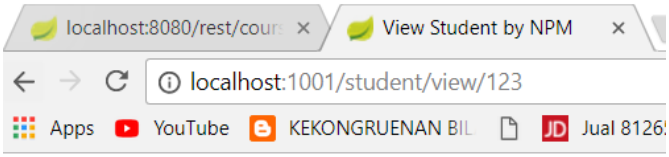


[{"id_course":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek Jr.","gpa":3.0,"courses":null}]},
{"id_course":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"Chanek Jr.","gpa":3.0,"courses":null}]},{"id_course":"CSC125","name":"DDP 1","credits":4,"students":[]},{"id_course":"CSC126","name":"MPKT","credits":6,"students":[{"npm":"123","name":"Chanek","gpa":4.0,"courses":null}]}]

Galuh Octavia Chrisdianti
1506757844
APAP – A

Membuat Service Consumer

Hasil tampilan dan *console* **"localhost:1001/student/view/123"**



NPM = 123

Name = Chanek

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

Galuh Octavia Chrisdianti
1506757844
APAP – A

Latihan

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

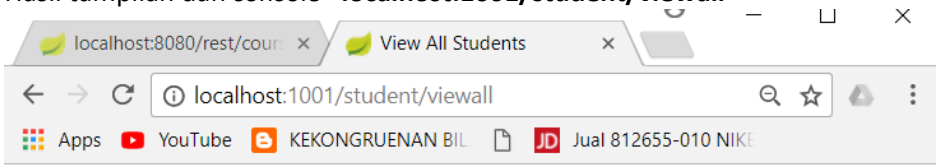
   Pada file StudentDAOImpl.java:

```
@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:1001/student/viewall"**



All Students

No	NPM	Name	GPA	Cumlaude	Delete	Update	
1	123	Chanek	4.0	Yes	Cum Laude!	Delete Data	Updated Data
2	124	Chanek Jr.	3.0	No	Sangat Memuaskan!	Delete Data	Updated Data

Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 7

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

   Pada file CourseDAOImpl.java:

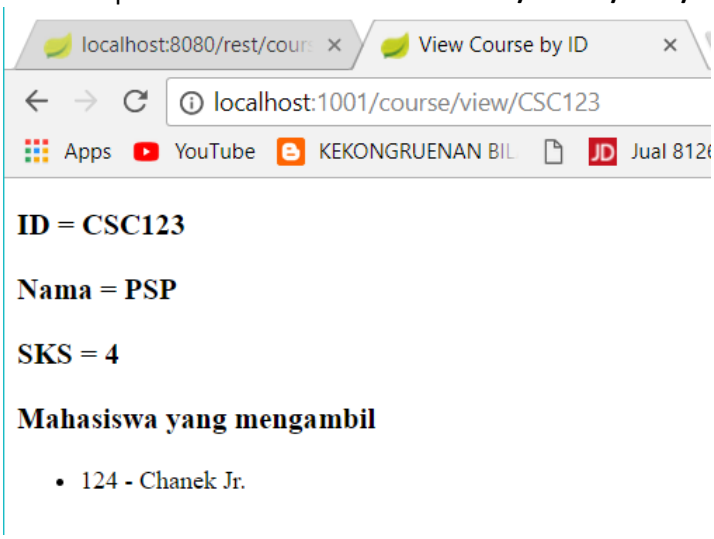
```
@Service
public class CourseDAOImpl implements CourseDAO {
    @Autowired
    private RestTemplate restTemplate;

    @Override
    public CourseModel selectCourse(String id_course) {
        CourseModel course = restTemplate.getForObject(
                "http://localhost:8080/rest/course/view/" + id_course,
                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:

```
public class CourseServiceRest implements CourseService {
    @Autowired
    private CourseDAO courseDAO;

    @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();
    }
```

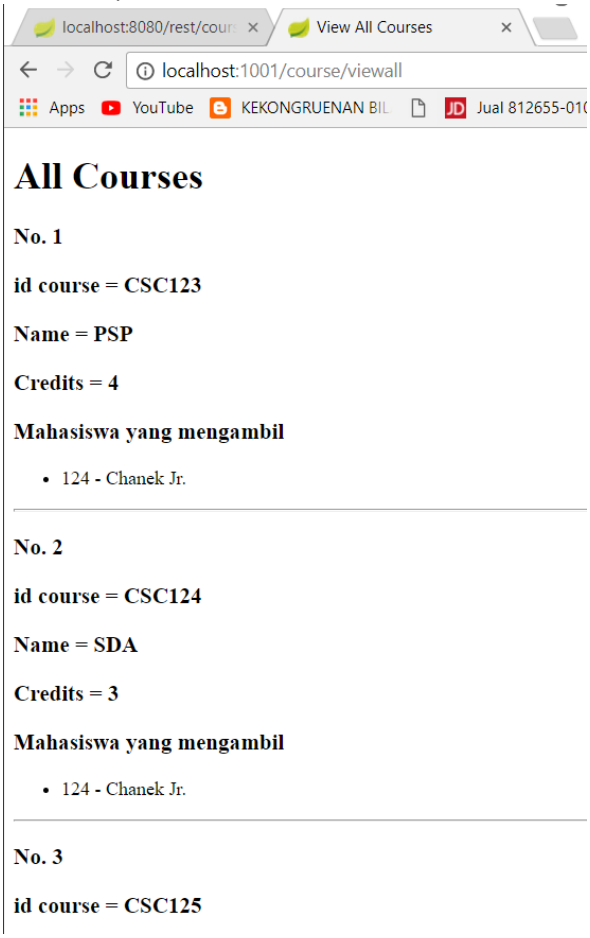
Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 7

Hasil tampilan dan *console* **"localhost:1001/course/view/CSC123"**



Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 7

Hasil tampilan dan *console* **"localhost:1001/course/viewall"**



Hal yang dipelajari:

Yang saya pelajari dari *tutotial* 7 kali adalah dimulainya implementasi pemisahan anatar *layer backend* (*service producer*) dan *layer frontend (service consumer).* Dengan adanya pemisahan *layering* ini maka untuk *service consumer* dapat berfokus hanya dalam penyajian data ke *user* saja tanpa harus melakuka pengolahan data langsung menghubungi database. Sedangkan *service producer* akan berfokus pada pengambilan dan pengolahan data dari *database.* Untuk dapat berkomunikasi dengan *service consumer*, *service producer* menyediakan web service yang dapat dikonsumsi oleh *service consumer*.