

Nama : Alberto
NPM: 1506757756

Latihan 1:

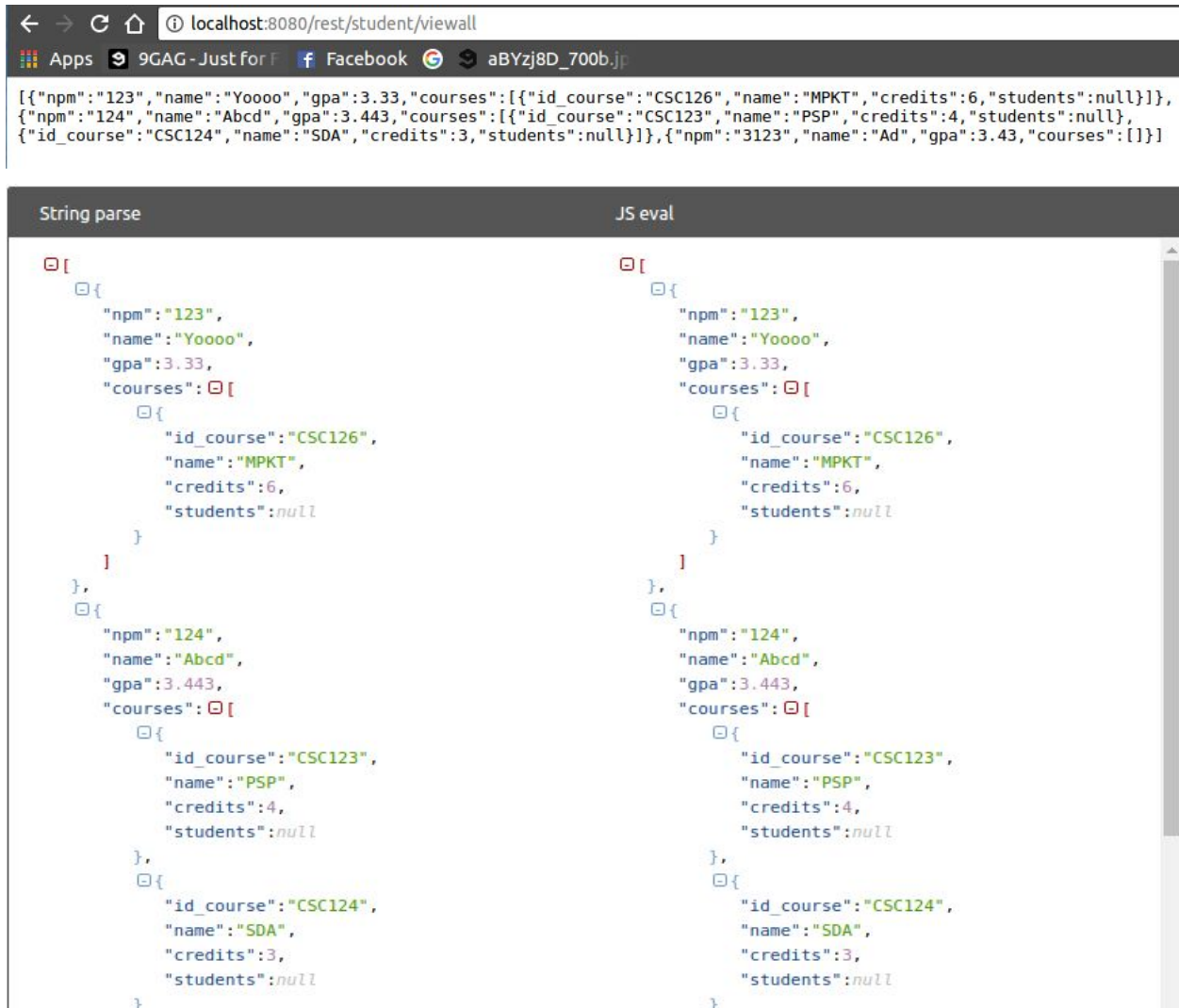
Buatlah service untuk mengembalikan seluruh student yang ada di basis data. Service ini mirip seperti method viewAll di Web Controller. Service tersebut di-mapping ke "/rest/student/viewall".

Buat Method viewall di StudentRestController yang akan memanggil selectAllStudents yang ada di StudentService dan mengembalikan objek List tersebut.

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

    return students;
}
```

Tampilan:



localhost:8080/rest/student/viewall

```
[{"npm":"123","name":"Yoooo","gpa":3.33,"courses":[{"id_course":"CSC126","name":"MPKT","credits":6,"students":null}]}, {"npm":"124","name":"Abcd","gpa":3.443,"courses":[{"id_course":"CSC123","name":"PSP","credits":4,"students":null}, {"id_course":"CSC124","name":"SDA","credits":3,"students":null}]}, {"npm":"3123","name":"Ad","gpa":3.43,"courses":[]}]
```

String parse

```
[{"npm":"123","name":"Yoooo","gpa":3.33,"courses":[{"id_course":"CSC126","name":"MPKT","credits":6,"students":null}]}, {"npm":"124","name":"Abcd","gpa":3.443,"courses":[{"id_course":"CSC123","name":"PSP","credits":4,"students":null}, {"id_course":"CSC124","name":"SDA","credits":3,"students":null}]}, {"npm":"3123","name":"Ad","gpa":3.43,"courses":[]}]
```

JS eval

```
[{"npm":"123","name":"Yoooo","gpa":3.33,"courses":[{"id_course":"CSC126","name":"MPKT","credits":6,"students":null}]}, {"npm":"124","name":"Abcd","gpa":3.443,"courses":[{"id_course":"CSC123","name":"PSP","credits":4,"students":null}, {"id_course":"CSC124","name":"SDA","credits":3,"students":null}]}, {"npm":"3123","name":"Ad","gpa":3.43,"courses":[]}]
```

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).

Dengan cara membuat CourseRestController yang berisi kode berikut:

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

    return course;
}

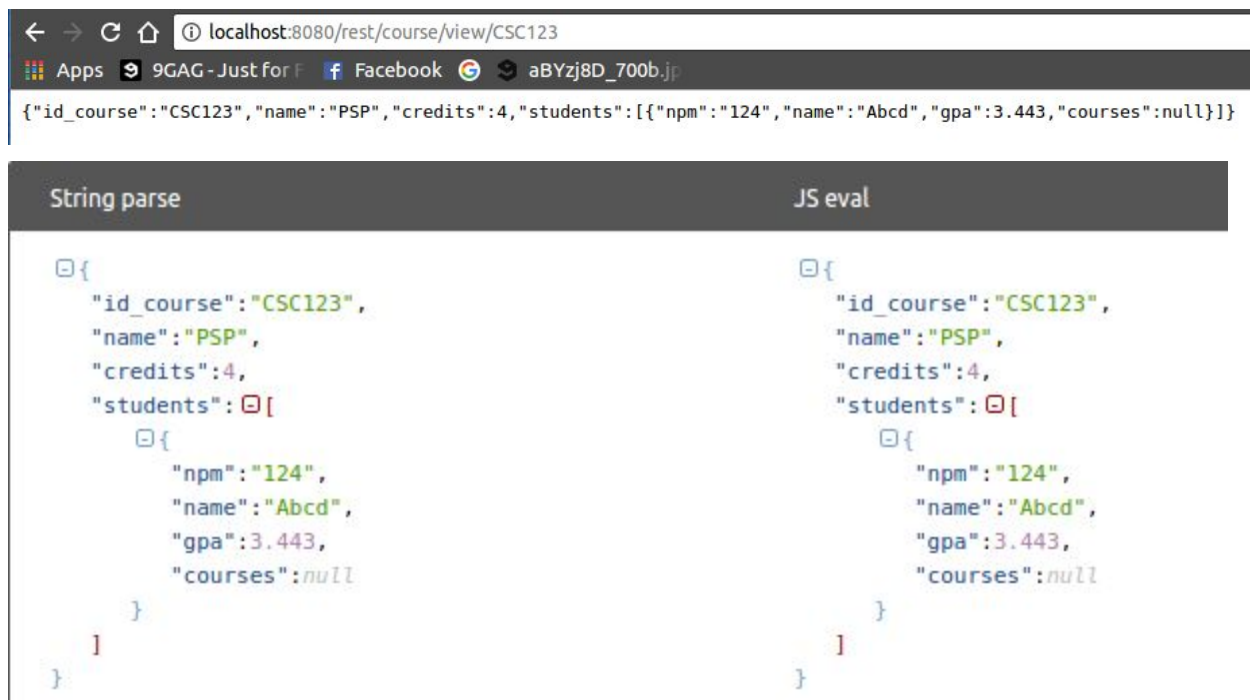
@RequestMapping("/course/viewall")
public List<CourseModel> viewall ()
{
    List<CourseModel> courses = studentService.selectAllCourses();

    return courses;
}
```

Method viewCourse digunakan untuk menampilkan course berdasarkan idCourse yang diinput dan mengembalikannya dalam bentuk objek. Sementara method viewall digunakan untuk mengembalikan semua course yang ada dan dikembalikan dalam bentuk objek.

Tampilan:

View Course



The screenshot shows a web browser at the URL `localhost:8080/rest/course/view/CSC123`. The response is a JSON object: `{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null}]}`. Below the browser, there are two panels for parsing the JSON: "String parse" and "JS eval". Both panels show the same JSON structure with syntax highlighting.

String parse	JS eval
<pre>{ "id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{ "npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null }] }</pre>	<pre>{ "id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{ "npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null }] }</pre>

View All Course:

localhost:8080/rest/course/viewall

Apps 9GAG - Just for Facebook aBYzj8D_700b.jp

```
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null}]}, {"id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null}]}, {"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students": []}, {"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": [{"npm": "123", "name": "Yoooo", "gpa": 3.33, "courses": null}]}]
```

String parse

JS eval

```
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null}]}, {"id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null}]}, {"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students": []}, {"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": [{"npm": "123", "name": "Yoooo", "gpa": 3.33, "courses": null}]}]
```

```
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null}]}, {"id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "Abcd", "gpa": 3.443, "courses": null}]}, {"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students": []}, {"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": [{"npm": "123", "name": "Yoooo", "gpa": 3.33, "courses": null}]}]
```

Latihan 3:

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


Buat method ini di dalam `StudentServiceRest` yang akan memanggil `selectAllStudents` yang ada di `StudentDAO`

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

Method yang ada di `StudentDAOImpl` adalah seperti di bawah ini. Method tersebut menggunakan `RestTemplate` dan `getForObject` yang menerima parameter link <http://localhost:8080/rest/course/viewall> dan menampilkan list dari student.

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

Tampilan:



All Students

Show 10 entries

Search:

No	NPM	Name	GPA	Cum Laude	Delete	Update
No data available in table						
No. 1	NPM = 123	Name = Yoooo	GPA = 3.33	Tidak	Delete Data	Update Data
No. 2	NPM = 124	Name = Abcd	GPA = 3.443	Tidak	Delete Data	Update Data
No. 3	NPM = 3123	Name = Ad	GPA = 3.43	Tidak	Delete Data	Update Data

Showing 0 to 0 of 0 entries

Previous Next

Maka di console akan muncul tulisan seperti berikut yang berarti method di Service terpanggil

```
INFO 3987 --- [nio-9090-exec-1] o.a.c.c.â.â.â.[Tomcat].â.[localhost].â.[/] : Initializing Spring FrameworkServlet
INFO 3987 --- [nio-9090-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet'
INFO 3987 --- [nio-9090-exec-1] o.s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet'
INFO 3987 --- [nio-9090-exec-4] com.example.service.StudentServiceRest : REST - select all students
```

Latihan 4:

Implementasikan service consumer untuk class CourseModel dengan membuat class-class DAO dan service baru.

Buat method ini di dalam CourseServiceRest yang akan memanggil method-method yang ada di CourseDAO

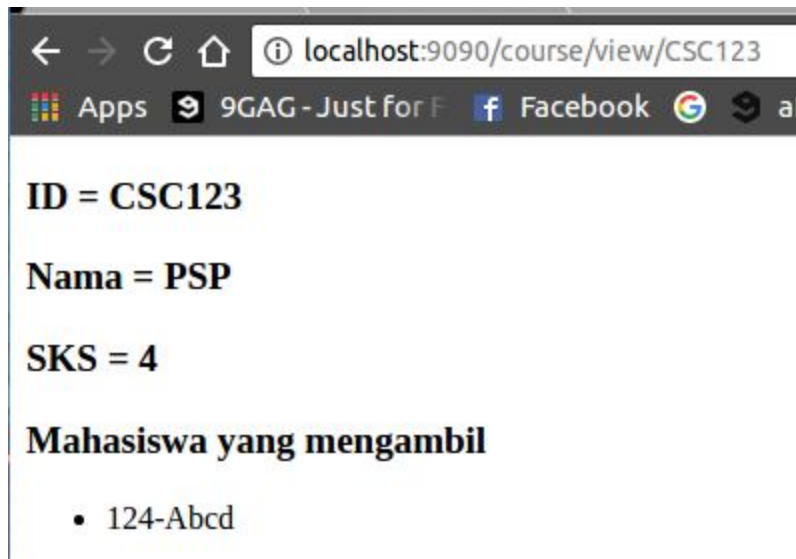
```
.  
  
@Override  
public CourseModel selectCourse(String idCourse) {  
    log.info("REST - select course with id {}", idCourse);  
    return courseDAO.selectCourse(idCourse);  
}  
  
@Override  
public List<CourseModel> selectAllCourse() {  
    log.info("REST - select all courses");  
    return courseDAO.selectAllCourse();  
}
```

Method yang ada di CourseDAOImpl adalah seperti di bawah ini. Method tersebut menggunakan RestTemplate dan getForObject yang menerima parameter link dan mengembalikan objek course dari producer.

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

Tampilan

View Course:



View All Course:

The screenshot shows a web browser window with the address bar displaying 'localhost:9090/course/viewall'. The page content includes:

All Courses

Show 10 entries

Search:

No	Id	Name	Credits
No data available in table			
1	CSC123	PSP	4
2	CSC124	SDA	3
3	CSC125	DDP 1	4
4	CSC126	MPKT	6

Lesson Learned:

Pada tutorial kali ini, saya mempelajari tentang 2 layer webservice yaitu Producer dan Consumer. Producer lebih mengatur kepada akses ke database dan apa yang harus dilakukan sementara untuk Consumer digunakan untuk menampilkan apa yang telah dikerjakan oleh Producer.

Pada tutorial kali ini saya juga belajar tentang RestTemplate yang digunakan untuk meng-consume object REST web service. Sementara method getObject menerima parameter berupa URL producer web service dan tipe class dari object yang didapatkan.