1

#### Producer

localhost:8080/rest/student/view/123 apap-2017/tutorial7\_150 × / Joan localhost:8080/rest/stude × Json Parser Online X Tutorial 7 - Web Service X ← → C (i) localhost:8080/rest/student/view/123 {"npm":"123","r gpa":4.0,"courses":[{"id\_course":"CSC126","name":"MPKT","credits":6,"students":null}]} Parse online napap-2017/tutorial7\_150 × / localhost8080/rest/stud × / Json Parser Online × Tutorial 7 - Web Service × ← → C ① Tidak aman | json.parser.online.fr Json Parser Online You like it? Support it! Donate Try out ("npm":"128", "name":"Namnung", "gpa":4.0, "courses": [{"id\_course":"CSC126", "name":"MPKI", "credits":6, "students":null}]) String parse JS eval "npm": "123", "npm": "123", "name": "Namnung",
"gpa": 4.0, "name":"Namr "courses": 🖯 [ "courses": 🖯 [ "id\_course":"CSC126", "id\_course":"CSC126", "credits":6, "credits":6, "students":null "students":null Latihan 1: localhost:8080/rest/student/viewall 🕥 apap-2017/tutorial7\_150 x 0 - 0 Q 🕁 😘 🔝 🚳  $\leftarrow$   $\rightarrow$   $\mathbf{C}$  (i) localhost:8080/rest/student/viewall [("npm":"122","name": "Namnung", "gpa":4.0, "Courses": [("id\_course":"CSC128", "name":"MPMT", "credits":6, "students":null}), ("npm":"124", "name": "Budi", "gpa":3.5, "courses": [("id\_course":"CSC128", "name":"PSP", "credits":4, "students":null}, ("id\_course":"CSC124", "name":"SDM", "credits":3, "students":null}), ("npm":"125", "name":"Ari", "gpa":3.8, "courses":[])] Parse online 🔾 apap-2017/tutorial7\_150 x 🥦 localhost:8080/rest/stud x 🗸 Json Parser Online x 🕻 🗋 Tutorial 7 - Web Service x ø ← → C ① Tidak aman | json.parser.online.fr © ☆ 😉 🔡 🐵 : Json Parser Online You like it? Support it! Donate Try out Beta! Samples ▼ Options ▼ [("npm":"128", "name":"Namnung", "opa"(4.0, "courses":
[("id\_ourse":"CSC126", "name":"MSFTT, "credita":6, "sudents":null}]],
("npm":"128", "name:""MSFT, "opa":6.8, "nourses":
[("id\_ourse":"CSC128", "name":"MSFT, "credita":4, "sudents":null),
("id\_ourse":"CSC124", "name":"SEAT", "credita":4, "sudents":null})],
("npm":"128", "name":"%ar", "opa":3.0, "courses":[])] | Topm":"125",
 "name":"Namuning",
 "goal",
 "courses":"C50126",
 "name":"NSETT",
 "codits":"6,
 "students":null
} "npm":"123",
"name":"Nammung",
"gpa":4.0,
"courses": 🖯 [ "name":"MPKT",
"credits":6,
"students":null "npm":"124",
"name":"Budi",
"gpa":3.5,
"courses": 🗎 [ "id\_course":"CSC123",
"name":"PSP",
"credits":4, "name": "PSP",
"credits":4,
"students": null "id\_course":"CSC124",
"name":"SDA",
"credits":3,
"students":null
-tivate Wi "name":"SDA", "credits":3, "students":null "students"; null Activate Windows FAQ • Privacy Policy • Changelog • Donation via PayPal • Donation in Bitcoins Author: Olivier Cuenot }, 1. 01 01 "npm": "125", "npm": "125", "name": "Ari", "name": "Ari", "gpa":3.0, "gpa":3, "courses": 0 [ "courses": 🖯 { 1 Ac

1

#### Code:

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

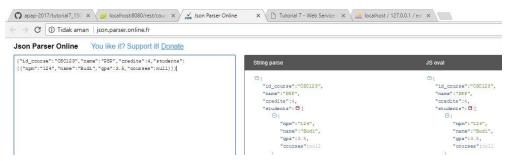
## Penjelasan latihan 1:

Method tersebut agar mengembalikan List of StudentModel dalam bentuk objek dengan format JSON yang kemudian akan diparse agar kita dapat melihat outputnya (seperti API).

Latihan 2: localhost:8080/rest/course/view/CSC123



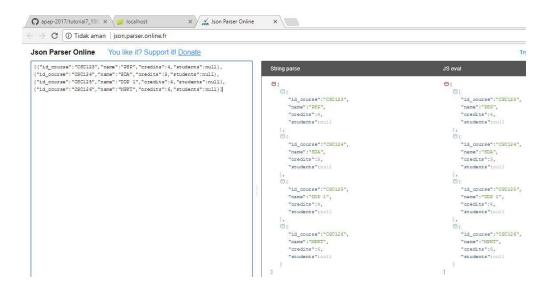
Parse online



Latihan 2: localhost:8080/rest/course/viewall



Parse online



#### Code:

```
package com.example.rest;
3⊕ import java.util.List;
 14
 15 @RestController
 16 @RequestMapping("/rest")
 17 public class CourseRestController {
 180
         @Autowired
 19
         CourseService courseService;
 20
 210
         @RequestMapping("/course/view/{id_course}")
         public CourseModel view (Model model,
 22
 23
                 @PathVariable(value = "id course") String id course)
 24
             CourseModel course = courseService.selectCourse (id course);
 25
 26
             return course;
 27
         }
 28
 29
 30⊜
         @RequestMapping("course/viewall")
         public List<CourseModel> viewall (Model model) {
 31
 32
             List<CourseModel> courses = courseService.selectAllCourse();
 33
 34
             return courses;
 35
         }
 36 }
```

# Penjelasan latihan 2:

Terdapat @RequestMapping("/rest") pada *class header* artinya semua *request mapping* pada kelas tersebut diawali dengan "/rest". Kedua method tersebut agar mengembalikan CourseModel dan List of CourseModel dalam bentuk objek dengan format JSON yang kemudian akan diparse agar kita dapat melihat outputnya (seperti API).

## Consumer

- Uji coba running
  - o localhost:8080/rest/student/view/123



localhost:9090/student/view/123



MPKT-6 sks

2017-11-02 00:21:13.049 INFO 7444 --- [nio-9090-exec-1] com.example.service.StudentServiceRest : REST - select student with npm123

Latihan 3: localhost:9090/student/viewall



### Penjelasan latihan 3:

Method tersebut diimplementasikan pada kelas StudentServiceRest, jadi kelas ini menggunakan web service bukan lagi dari database. Mungkin kelas ini fungsinya sama seperti service database dimana nanti method akan digunakan pada StudentController.

• Latihan 4:

localhost:9090/course/view/CSC123



124-Budi

localhost:9090/course/viewall



#### Code:

### Class CourseServiceRest

```
package com.example.service;
  3⊕ import java.util.List;
 13
 14 @Slf4j
 15 @Service
 16 @Primary
 17 public class CourseServiceRest implements CourseService {
 189
        @Autowired
 19
        private CourseDAO courseDAO;
 20
 210
         @Override
        public CourseModel selectCourse(String id course) {
₩22
 23
            log.info("REST - select course with id course{}", id course);
 24
             return courseDAO.selectCourse(id course);
 25
         }
 26
 279
        @Override
        public List<CourseModel> selectAllCourse() {
⇔28
 29
            log.info("REST - select all course");
 30
            return courseDAO.selectAllCourse();
 31
        }
 32
33 }
```

### Class Course DAOImpl

```
package com.example.dao;
  3@ import java.util.List;
5 import org.springframework.beans.factory.annotation.Autowired;
   6 import org.springframework.stereotype.Service;
   7 import org.springframework.web.client.RestTemplate;
   9 import com.example.model.CourseModel;
 12 public class CourseDAOImpl implements CourseDAO {
 140
          MAutowired
  15
         private RestTemplate restTemplate;
 16
  170
         public CourseModel selectCourse(String id_course) {
   CourseModel course = restTemplate.getForObject("http://localhost:8080/rest/course/view/" + id_course, CourseModel.class);
△18
 19
 20
21
 230
         public List<CourseModel> selectAllCourse() {
Q<sub>25</sub>
26
              List<CourseModel> courses = restTemplate.getForObject("http://localhost:8080/rest/course/viewall", List.class);
              return courses;
28 }
```

## Interface CourseDAO

```
package com.example.dao;

package com.example.dao;

minort java.util.List;

public interface CourseDAO {
    CourseModel selectCourse(String id_course);
    List<CourseModel> selectAllCourse();
}
```

# Penjelasan latihan 4:

Pada latihan 4 saya membuat 2 kelas baru dan 1 interface yaitu pada package DAO (Interface CourseDAO dan Class CourseDAOImplyang mengimplements CourseDAO) dan pada package service (Class CourseServiceRest yang mengimplements CourseService). Pada interface CourseDAO saya mendefine method apa saja yang akan digunakan di CourseController dari CourseMapper, kemudian pada Class CourseDAOImpl akan dipanggil object url yang dibutuhkan, sementara CourseServiceRest adalah hal yang sama seperti CourseServiceDatabase (tetapi dalam hal ini menggunakan web service).