

Tutorial07Produser

Student/view/123



Setelah pengaplikasian Json Parser online



LATIHAN 1

Dalam latihan 1 yang menampilkan seluruh data student ini saya membuat requestMapping viewall di dalam requestMapping rest melanjutkan requestMapping view student berdasarkan id yang sebelumnya dicontohkan. Perbedaannya terletak pada pengembalian berupa sebuah List<StudentModel> karena menampilkan seluruh data student. Berikut cuplikan kodenya:

```
@RestController
@RequestMapping("/rest")
public class StudentRestController {

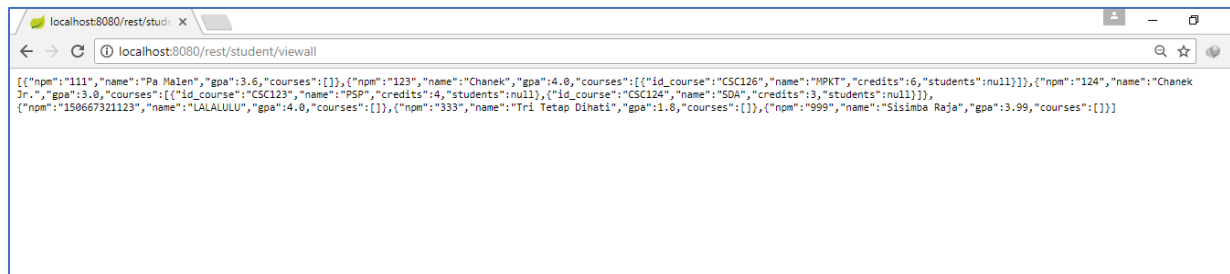
    @Autowired
    StudentService studentService;

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

Ervina Puspita Sari | 1506689414

APAP B

student/viewall



Hasil file json yang keluar.

LATIHAN 2

Dalam membuat service baru untuk Kelas Course saya membuat CourseRestController.java dalam package com.exaple.rest dalam Tutorial07Produser yang isinya hampir sama dengan StudentRestController.java yang dicontohkan sebelumnya, hanya nama kelas dan methodnya yang saya ganti. Berikut cuplikan kodenya:

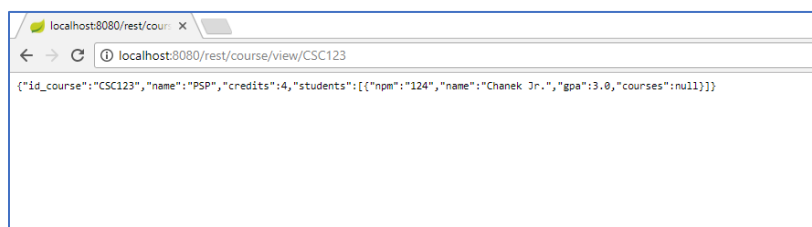
```
@RestController
@RequestMapping("/rest")
public class CourseRestController {

    @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> viewall() {
        List<CourseModel> courses = courseService.selectAllCourses();
        return courses;
    }
}
```

course/view/CSC123

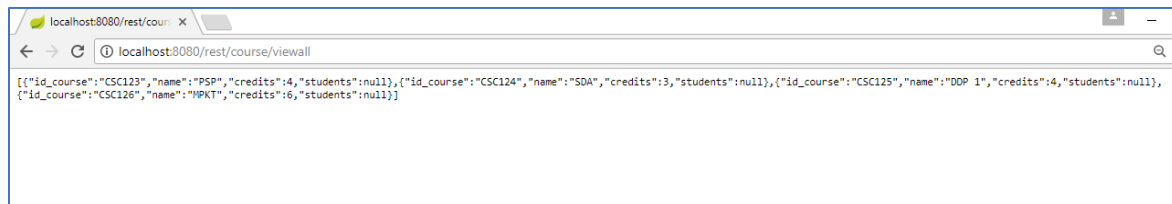


Ervina Puspita Sari | 1506689414

APAP B

Hasil file json yang keluar

[course/viewall](#)



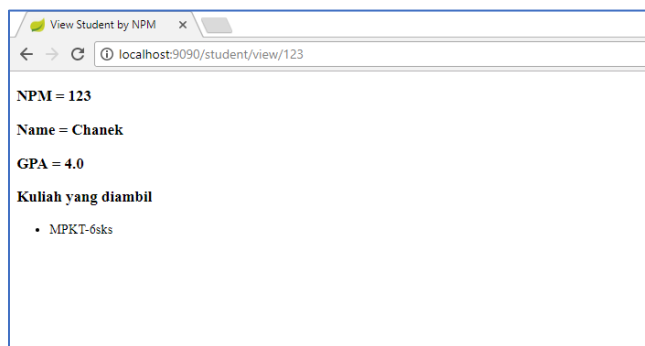
Hasil file json yang keluar

Tutorial07Consumer

[student/view/123](#)



[9090/student/view/123](#)



LATIHAN 3

Saat saya melengkapi metode `selectAllStudents`, saya tetap menggunakan `restTemplate` seperti metode `selectStudent()` sebelumnya, hanya saya untuk dapat mengembalikan `List<StudentModel>` saya juga menggunakan `ResponseEntity` yang mereturn array as List. Berikut cuplikan kodenya:

```
@Override
public List<StudentModel> selectAllStudents () {

    ResponseEntity<StudentModel[]> response =
        restTemplate.getForEntity("http://localhost:8080/rest/student/viewall", StudentModel[].class);

    return Arrays.asList(response.getBody());
}
```

Ervina Puspita Sari | 1506689414

APAP B

student/viewall

```
localhost:8080/rest/stud: X
localhost:8080/rest/student/viewall

[{"npm": "111", "name": "Pa Malen", "gpa": 3.6, "courses": []}, {"npm": "123", "name": "Chanek", "gpa": 4.0, "courses": [{"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": null}], {"npm": "124", "name": "Chanek Jr.", "gpa": 3.0, "courses": [{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": null}], {"npm": "150667321123", "name": "LALALULU", "gpa": 4.0, "courses": []}, {"npm": "333", "name": "Tri Tetap Dihati", "gpa": 1.8, "courses": []}, {"npm": "999", "name": "Sisimba Raja", "gpa": 3.99, "courses": []}]
```

9090/student/viewall

View All Students

- [Home](#)
- [Daftar Mahasiswa](#)
- [Menambah Mahasiswa](#)
- [Update Mahasiswa](#)

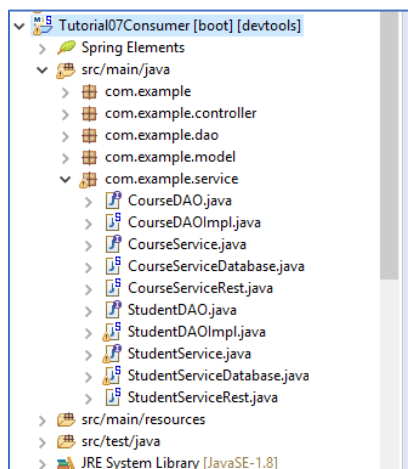
Show 10 entries

NO	NPM	Name	GPA	Keterangan	Delete	Update
1	111	Pa Malen	3.6	Cumlaude	Delete Data	Update Data
2	123	Chanek	4.0	Cumlaude	Delete Data	Update Data
3	124	Chanek Jr.	3.0	Sangat Memuaskan!	Delete Data	Update Data
4	150667321123	LALALULU	4.0	Cumlaude	Delete Data	Update Data
5	333	Tri Tetap Dihati	1.8	Sangat Memuaskan!	Delete Data	Update Data
6	999	Sisimba Raja	3.99	Cumlaude	Delete Data	Update Data

Showing 1 to 6 of 6 entries

Mata Kuliah APAP

LATIHAN 4

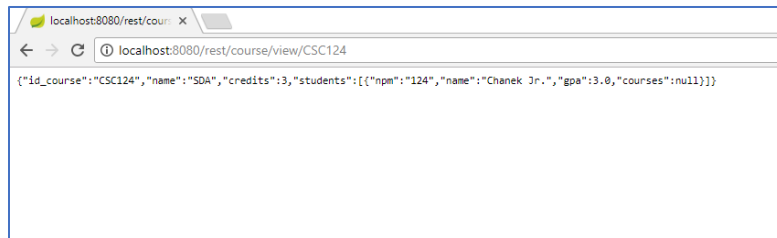


Dalam mengimplementasikan service consumer untuk Kelas CourseModel, saya mengikuti langkah-langkah yang sama dengan tutorial service consumer sebelumnya. Saya membuat interface CourseDAO, dan CourseDAOImpl yang implements interface CourseDAO. Isinya pun hampir mirip hanya berbeda nama kelas dan methodnya saja. Saya membuat view course berdasarkan id_course dan viewall course, hanya saja saat tutorial06 ternyata tidak dibuat controller untuk menampilkan viewall course jadi saat ini yang bisa ditampilkan hanya view course berdasarkan id_course saja melalui dua host 8080 dan 9090.

Ervina Puspita Sari | 1506689414

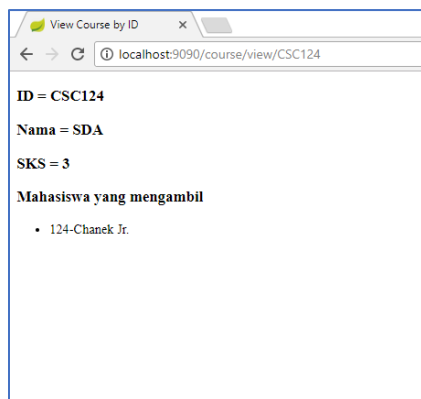
APAP B

[course/view/CSC124](#)



Hasil file json yang keluar

[9090/course/view/CSC124](#)



Hasil file json yang keluar

*P.S : yang saya pelajari dari tutorial ini adalah bagaimana cara service bekerja, ada produser dan consumer menggambarkan adanya dua pihak nanti yang akan saling bertukar informasi berupa informasi-informasi berwujud json.