

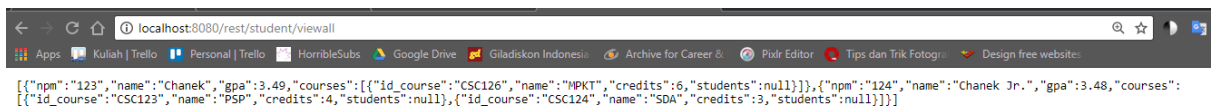
Ringkasan

Pada tutorial kali ini, saya mempelajari bagaimana cara menggunakan web service. Web service dilakukan dengan melakukan pemisahan layer antara backend dengan frontend dari sebuah web. Frontend merupakan aplikasi yang berinteraksi langsung dengan pengguna sedangkan backend merupakan pengolahan data dari database.

Latihan

1. Pada method `viewall()` yang berada pada `StudentRestController`, akan dilakukan mapping ke `"rest/student/viewall"` dan method akan memanggil method `selectAllStudents()` yang ada pada `StudentService` dan akan mengembalikan list dari object `StudentModel` dengan format JSON.

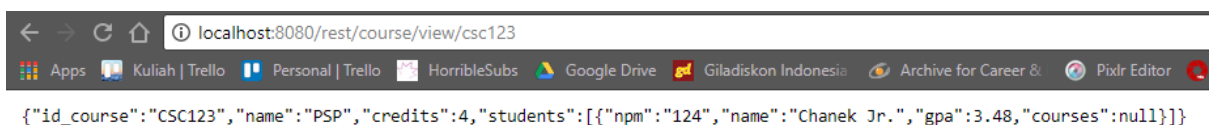
```
@RequestMapping("/student/viewall")
public List<StudentModel> viewall () {
    List<StudentModel> students = studentService.selectAllStudents ();
    return students;
}
```



```
{["npm":"123","name":"Chanek","gpa":3.49,"courses":[{"id_course":"CSC126","name":"MPKT","credits":6,"students":null}],["npm":"124","name":"Chanek Jr.","gpa":3.48,"courses":[{"id_course":"CSC123","name":"PSP","credits":4,"students":null},{"id_course":"CSC124","name":"SDA","credits":3,"students":null}]}
```

2. Saya membuat `CourseRestController` yang akan menggunakan REST web service untuk course. Untuk melihat suatu course, lakukan mapping ke `"rest/course/view/id_course"`. `id_course` merupakan parameter untuk melihat course yang diinginkan, selanjutnya panggil method `selectCourse` dengan parameter `id_course` yang diinginkan dari `CourseService`. Method ini akan mengembalikan object `CourseModel` dalam format JSON.

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



```
{"id_course":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek Jr.","gpa":3.48,"courses":null}]}
```

Pada method `viewall()` yang berada pada `CourseRestController`, akan dilakukan mapping ke `"rest/course/viewall"` dan method akan memanggil method `selectAllCourses()` yang ada pada `CourseService` dan akan mengembalikan list dari object `CourseModel` dengan format JSON.

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

```
localhost:8080/rest/course/viewall
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "Chanek Jr.", "gpa": 3.48, "courses": null}], {"id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "Chanek Jr.", "gpa": 3.48, "courses": null}], {"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": [{"npm": "123", "name": "Chanek", "gpa": 3.49, "courses": null}]}
```

- Method ini menggunakan `ResponseEntity<T>` yang akan menyimpan hasil yang diinginkan dari rest dengan url yang diinginkan dalam bentuk JSON. Pada method ini digunakan juga `ParameterizedTypeReference` untuk melakukan passing data dengan tipe generic. Selanjutnya akan menggunakan `getBody()` untuk mengambil data dari JSON yang dihasilkan pada response yang dihasilkan.

```
@Override
public List<StudentModel> selectAllStudents ()
{
    ResponseEntity<List<StudentModel>> response = restTemplate.exchange("http://localhost:8080/rest/student/viewall",
        HttpMethod.GET, null, new ParameterizedTypeReference<List<StudentModel>>() {});
    List<StudentModel> students = response.getBody();

    return students;
}
```

localhost:9090/student/viewall

Home Daftar Mahasiswa Menambah Mahasiswa

All Students

Show 10 entries

No	NPM	Name	GPA	Predikat	Update	Delete
1	123	Chanek	3.49	Cum Laude!	Update	Delete
2	124	Chanek Jr	3.48	Sangat Memuaskan!	Update	Delete

Showing 1 to 2 of 2 entries

Previous 1 Next

Mata Kuliah APAP

- Method untuk view course menggunakan `getForObject` untuk mendapatkan data sebuah object dari sebuah url dengan tipe class yang telah ditentukan. Setelah membuat method pada `CourseDAOImpl`, panggil method `selectCourse` dengan parameter `id_course` dari `CourseDAO` pada `CourseServiceRest` untuk menjalankan `CourseService`

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

localhost:9090/course/view/csc123

Apps Kuliah | Trello Personal | Trello HorribleSubs Google Drive

Home Daftar Mahasiswa Menambah Mahasiswa

ID = CSC123

Name = PSP

Credit = 4

Mahasiswa yang mengambil

- 124 - Chanek Jr.

Mata Kuliah APAP

Method untuk view all course menggunakan `ResponseEntity<T>` yang akan menyimpan hasil yang diinginkan dari rest dengan url yang diinginkan dalam bentuk JSON. Pada method ini digunakan juga `ParameterizedTypeReference` untuk melakukan passing data dengan tipe generic. Selanjutnya akan menggunakan `getBody()` untuk mengambil data dari JSON yang dihasilkan pada response yang dihasilkan. Setelah membuat method pada `CourseDAOImpl`, panggil method `selectAllCourses` dengan parameter `id_course` dari `CourseDAO` pada `CourseServiceRest` untuk menjalankan `CourseService`

```
@Override
public List<CourseModel> selectAllCourses ()
{
    ResponseEntity<List<CourseModel>> response = restTemplate.exchange("http://localhost:8080/rest/course/viewall",
        HttpMethod.GET, null, new ParameterizedTypeReference<List<CourseModel>>() {});
    List<CourseModel> courses = response.getBody();

    return courses;
}
```

All Courses

No. 1
ID = CSC123
Name = PSP
Credit = 4
Mahasiswa yang mengambil
• 124 - Chanak Jr.

No. 2
ID = CSC124
Name = SDA
Credit = 3
Mahasiswa yang mengambil
• 124 - Chanak Jr.

No. 3
ID = CSC126
Name = MPKT
Credit = 6
Mahasiswa yang mengambil
• 123 - Chanak