

Nama : Nabilla Andini

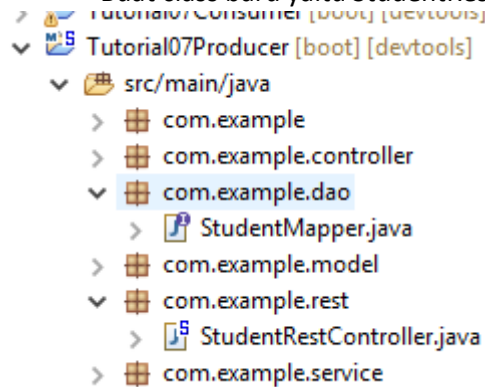
Kelas : C

NPM : 1506689465

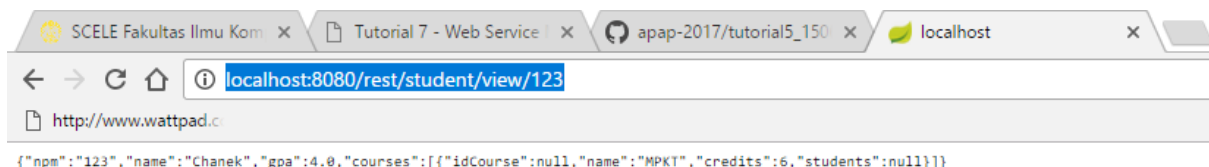
Tutorial 7 : Membuat Web Service Menggunakan Spring Boot Framework

A. Membuat Service producer

- Buat package baru com.example.rest
- Buat class baru yaitu StudentRestController.java pada package com.example.rest



- Buka halaman "localhost:8080/rest/student/view/123"



- JSON View



Nama : Nabilla Andini

Kelas : C

NPM : 1506689465

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("/student/viewall")
public List <StudentModel> viewAll () {

    List<StudentModel> students = studentService.selectAllStudents ();

    return students;
}
```

Students merupakan List of StudentModel yang berisi semua student yang ada di db dipanggil dengan method selectAllStudents dari StudentService. Kemudian list of StudentModel tsb direturn sebagai hasilnya.

localhost:8080/rest/student/viewall



JSON View



Nama : Nabilla Andini

Kelas : C

NPM : 1506689465

```
"courses": [
  {
    "idCourse": null,
    "name": "PSP",
    "credits": 4,
    "students": null
  },
  {
    "idCourse": null,
    "name": "SDA",
    "credits": 3,
    "students": null
  }
],
{
  "npm": "444",
  "name": "qwbjbj",
  "gpa": 2.2,
  "courses": [
  ]
},
{
  "npm": "6666",
  "name": "aaaaa",
  "gpa": 3.9,
  "courses": [
  ]
}
```

```
"courses": [
  {
    "idCourse": null,
    "name": "PSP",
    "credits": 4,
    "students": null
  },
  {
    "idCourse": null,
    "name": "SDA",
    "credits": 3,
    "students": null
  }
],
{
  "npm": "444",
  "name": "qwbjbj",
  "gpa": 2.2,
  "courses": [
  ]
},
{
  "npm": "6666",
  "name": "aaaaa",
  "gpa": 3.9,
  "courses": [
  ]
}
```

Activate Windows
Go to Settings to activate Windows

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).

- View Course by ID

```
@RequestMapping("/course/view/{idCourse}")
public CourseModel viewCourse(Model model, @PathVariable(value =
"idCourse") String id_course){
    CourseModel course = studentService.selectCourse(id_course);
    return course;
}
```

Sama seperti contoh view student by npm. Hanya select course berdasarkan id yang diperoleh dengan menjalankan method selectCourse(id_course) dari StudentService. Lalu course tsb direturn sebagai hasilnya untuk ditampilkan.

localhost:8080/rest/course/view/csc124

http://www.wattpad.com

```
{ "idCourse": "CSC124", "name": "SDA", "credits": 3, "students": [ { "npm": "124", "name": "Chanek Jr.", "gpa": 3.0, "courses": null } ] }
```

Nama : Nabilla Andini

Kelas : C

NPM : 1506689465

JSON View

String parse	JS eval
<pre>{ "idCourse": "CSC124", "name": "SDA", "credits": 3, "students": [{ "npm": "124", "name": "Chanek Jr.", "gpa": 3.0, "courses": null }] }</pre>	<pre>{ "idCourse": "CSC124", "name": "SDA", "credits": 3, "students": [{ "npm": "124", "name": "Chanek Jr.", "gpa": 3, "courses": null }] }</pre>

- View All Course

1. Menambahkan method viewAllCourse() di StudentRestController

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

2. Menambahkan List<CourseModel> selectAllCourses(); pada StudentService
3. Menambahkan method di StudentServiceDatabase

```
@Override
public List<CourseModel> selectAllCourses(){
    return studentMapper.selectAllCourses();
}
```

4. Menambahkan method di StudentMapper

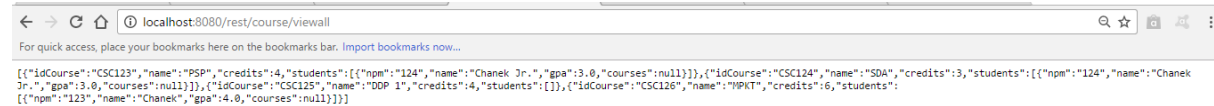
```
@Results(value = {
    @Result(property = "idCourse", column = "id_course"),
    @Result(property = "name", column = "name"),
    @Result(property = "credits", column = "credits"),
    @Result(property = "students", column = "id_course" ,
        javaType = List.class,
        many = @Many(select = "selectStudents"))
})
List<CourseModel> selectAllCourses();
```

5. Hasil :

Nama : Nabilla Andini

Kelas : C

NPM : 1506689465



JSON View



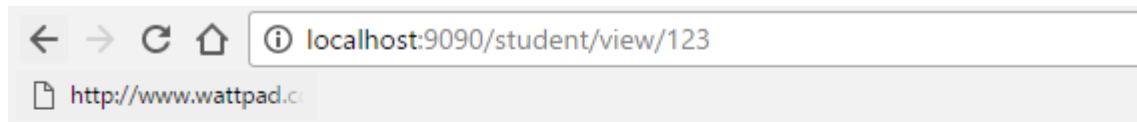
Nama : Nabilla Andini

Kelas : C

NPM : 1506689465

B. Membuat Service Consumer

- Pastikan service producer sudah berjalan dengan menjalankan localhost:8080/rest/student/view/123.
- Untuk menguji service consumer buka localhost:9090/student/view/123



NPM = 123

Name = Chanek

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

Mata Kuliah APAP

LATIHAN

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

Cantumkan dan jelaskan method Anda pada write-up

Commit dan Push pekerjaan Anda.

1. Menambahkan method selectAllStudents() di StudentServiceRest

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

2. Menambahkan method List<StudentModel> selectAllStudents (); di StudentDAO
3. Di StudentDAOImpl ditambahkan:

```
@Override
public List<StudentModel> selectAllStudents (){
    ResponseEntity<StudentModel[]> response =
        restTemplate.getForEntity("http://localhost:8080/rest/student/viewall",
        StudentModel[].class);
    return Arrays.asList(response.getBody());
}
```

Nama : Nabilla Andini

Kelas : C

NPM : 1506689465

Untuk mendapatkan list of objects tidak dapat menggunakan getForObject, tetapi menggunakan

ResponseEntity<T> getForEntity(java.lang.String url, java.lang.Object). Yang akan mengembalikan hasil ke ResponseEntity yang merepresentasikan keseluruhan http response

<div><div><div>←</div><div>→</div><div>↺</div><div>🏠</div></div><div>📘 localhost:9090/student/viewall</div></div> <div>For quick access, place your bookmarks here on the bookmarks bar. Import bookmarks now...</div>						
<h2>All Students</h2>						
No	NPM	Name	GPA	Cum laude	Delete	Update
No. 1	NPM = 123	Name = Chanek	GPA = 4.0	Cum Laude!	Delete Data	Update Data
No. 2	NPM = 124	Name = Chanek Jr.	GPA = 3.0	Sangat Memuaskan	Delete Data	Update Data
No. 3	NPM = 444	Name = qwbjbj	GPA = 2.2	Sangat Memuaskan	Delete Data	Update Data
No. 4	NPM = 6666	Name = aaaaa	GPA = 3.9	Cum Laude!	Delete Data	Update Data

Mata Kuliah APAP

Latihan 4: Implementasikan service consumer untuk class CourseModel dengan

membuat class-class DAO dan service baru.

Cantumkan dan jelaskan method Anda pada write-up

Commit dan Push pekerjaan Anda.

- **View Course By id course**

1. Membuat Interface CourseDAO di com.example.dao
CourseModel selectCourse (String id_course);
2. Membuat class CourseDAOImpl pada com.example.service

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

3. Implementasi method di StudentServiceRest

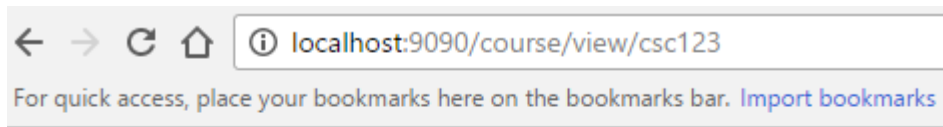
```
@Override
    public CourseModel selectCourse(String id) {
        Log.info("REST - select course with id {}", id);
        return courseDAO.selectCourse(id);
    }
```

Nama : Nabilla Andini

Kelas : C

NPM : 1506689465

4. Hasil:



ID = CSC123

Name = PSP

Credits = 4

Mahasiswa yang mengambil

- 124-Chanek Jr.

Mata Kuliah APAP

Nama : Nabilla Andini

Kelas : C

NPM : 1506689465

- **View All Courses**

1. Tambahkan method `List<CourseModel> selectAllCourses();` di `CourseDAO`
2. Tambahkan Method di `CourseDAOImpl`

```
@Override
public List<CourseModel> selectAllCourses ()
{
    ResponseEntity<CourseModel[]> responseEntity =
restTemplate.getForEntity("http://localhost:8080/rest/course/viewall",
CourseModel[].class);
    return Arrays.asList(responseEntity.getBody());
}
```

Sama seperti cara mengambil list of student yang dijelaskan di latihan 3

3. Tambahkan method di `StudentServiceRest`

```
@Override
public List<CourseModel> selectAllCourses() {
    Log.info("REST - select all courses");
    return courseDAO.selectAllCourses();
}
```

4. Hasil:

Nama : Nabilla Andini

Kelas : C

NPM : 1506689465



localhost:9090/course/viewall

For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

All Courses

No 1

ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang mengambil

- 124 - Chanek Jr.
-

No 2

ID = CSC124

Nama = SDA

SKS = 3

Mahasiswa yang mengambil

- 124 - Chanek Jr.
-

Nama : Nabilla Andini

Kelas : C

NPM : 1506689465

← → ↺ 🏠 ⓘ localhost:9090/course/viewall

For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

ID = CSC124

Nama = SDA

SKS = 3

Mahasiswa yang mengambil

- 124 - Chanek Jr.

No 3

ID = CSC125

Nama = DDP 1

SKS = 4

Mahasiswa yang mengambil

No 4

ID = CSC126

Nama = MPKT

SKS = 6

Mahasiswa yang mengambil

- 123 - Chanek

Lesson Learned:

1. Web Services connect services together. While, A service is the endpoint of a connection. Also, a service has some type of underlying computer system that supports the connection offered.
2. The application of a producer and a consumer
3. RESTful web service will return a JSON response.
4. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web.
5. The usage of @Primary annotation which indicates that a particular bean definition should be given preference when multiple beans are candidate to be autowired
6. Get an object and a list of object from RestTemplate