

Tutorial 7

CSIM603026 – Arsitektur dan Pemrograman Aplikasi Perusahaan

Semester Gasal 2017/2018

Membuat Web Service Menggunakan Spring Boot Framework

Membuat Service producer

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”**

Dengan mencopy method selectStudent sebelumnya, kemudian method diubah dengan mengembalikan arraylist dari students.

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



Abdalla Chair

1506757831

APAP - C

String parse

JS eval

```
[{"nm": "123", "name": "Nabilah", "gpa": 3.5, "courses": [{"idCourse": null, "name": "MPRT", "credits": 6, "students": null}], }, {"nm": "124", "name": "Bila", "gpa": 3.5, "courses": [{"idCourse": null, "name": "PSP", "credits": 4, "students": null}, {"idCourse": null, "name": "SDA", "credits": 3, "students": null}], }, {"nm": "125", "name": "Aan ajen ajen", "gpa": 3.2, "courses": [{"idCourse": null, "name": "MPRT", "credits": 6, "students": null}, {"idCourse": null, "name": "PSP", "credits": 4, "students": null}, {"idCourse": null, "name": "SDA", "credits": 3, "students": null}], }]
```

```
[{"nm": "123", "name": "Nabilah", "gpa": 3.5, "courses": [{"idCourse": null, "name": "MPRT", "credits": 6, "students": null}], }, {"nm": "124", "name": "Bila", "gpa": 3.5, "courses": [{"idCourse": null, "name": "PSP", "credits": 4, "students": null}, {"idCourse": null, "name": "SDA", "credits": 3, "students": null}], }, {"nm": "125", "name": "Aan ajen ajen", "gpa": 3.2, "courses": [{"idCourse": null, "name": "MPRT", "credits": 6, "students": null}, {"idCourse": null, "name": "PSP", "credits": 4, "students": null}, {"idCourse": null, "name": "SDA", "credits": 3, "students": null}], }]
```

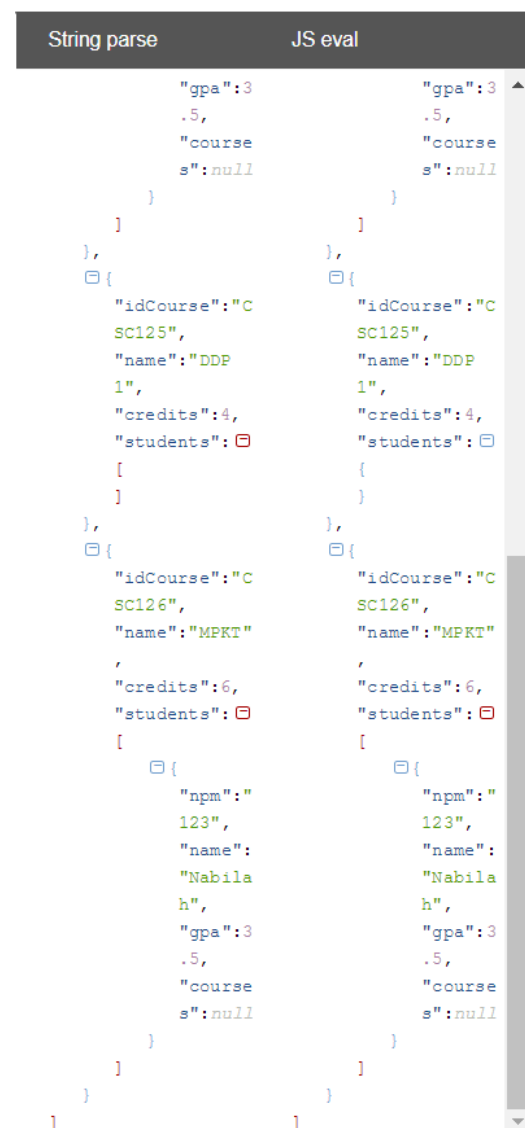
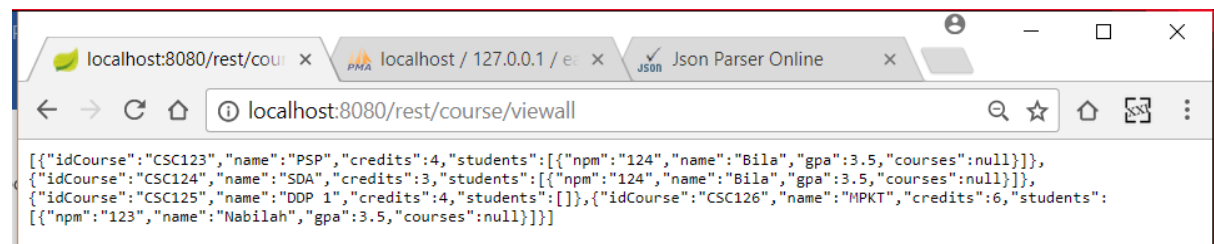
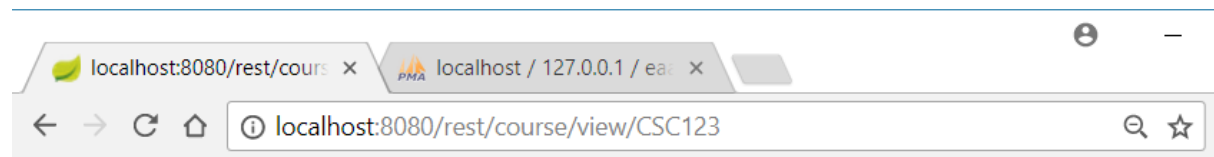
Latihan 2:

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 elihat semua course (view all)

Method – method yang dibuat kurang lebih sama dengan method student untuk view dan viewall, hanya saja setiap “student” diubah menjadi course.

```
StudentRestController... application.properties application.properties CourseRestController.ja...
2
3+ import java.util.List;
14
15 @RestController
16 @RequestMapping("/rest")
17 public class CourseRestController
18 {
19
20     @Autowired
21     CourseService courseService;
22
23     @RequestMapping("/course/view/{id}")
24     public CourseModel view (@PathVariable(value = "id") String id) {
25         CourseModel course = courseService.selectCourse(id);
26         return course;
27     }
28
29     @Autowired
30     @RequestMapping("/course/viewall")
31     public List<CourseModel> viewAll () {
32         List<CourseModel> courses = courseService.selectAllCourse();
33         return courses;
34     }
35
36 }
37
```

String parse	JS eval
<pre>{ "idCourse": "CSC1 23", "name": "PSP", "credits": 4, "students": [{ "npm": "124 ", "name": "Bi la", "gpa": 3.5, "courses": null }] }</pre>	<pre>{ "idCourse": "CSC1 23", "name": "PSP", "credits": 4, "students": [{ "npm": "124 ", "name": "Bi la", "gpa": 3.5, "courses": null }] }</pre>



Abdalla Chair
1506757831
APAP - C

Membuat Service Consumer

Latihan 3:

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

Method dibuat pertama-tama pada section berikut:

`StudentServiceRest`:

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

`StudentDAOImpl`:

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

Kemudian method dapat menampilkan view dibawah ini, dengan menggunakan “`getForObject`” dari



`RestTemplate`, yang merujuk ke `localhost:8080/rest/student/viewall`

Abdalla Chair

1506757831

APAP - C

String parse

JS eval

```
[
  {
    "npm": "123",
    "name": "Nabilah",
    "gpa": 3.5,
    "courses": [
      {
        "idCourse": null,
        1,
        "name": "MPKT",
        "credits": 6,
        "students": null
        1
      }
    ]
  },
  {
    "npm": "124",
    "name": "Bila",
    ,
    "gpa": 3.5,
    "courses": [
      {
        "idCourse": null,
        1,
        "name": "PSP",
        "credits": 4,
        "students": null
        1
      }
    ]
  },
  {
    "npm": "125",
    "name": "Aanajen ajen",
    "gpa": 3.2,
    "courses": [
      {
        "idCourse": null,
        1,
        "name": "SDA",
        "credits": 3,
        "students": null
        1
      }
    ]
  }
]
```

String parse

JS eval

```
    "name": "Bila",
    ,
    "gpa": 3.5,
    "courses": [
      {
        "idCourse": null,
        1,
        "name": "PSP",
        "credits": 4,
        "students": null
        1
      },
      {
        "idCourse": null,
        1,
        "name": "SDA",
        "credits": 3,
        "students": null
        1
      }
    ]
  },
  {
    "npm": "125",
    "name": "Aanajen ajen",
    "gpa": 3.2,
    "courses": [
      {
        "idCourse": null,
        1,
        "name": "PSP",
        "credits": 4,
        "students": null
        1
      }
    ]
  }
]
```

Navbar

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

All Students

Show entries

Search:

No	NPM	Name	GPA	Cum laude	Delete	Update
1	123	Nabilah	3.5	Cum Laude!	Delete Data	Update Data
2	124	Bila	3.5	Cum Laude!	Delete Data	Update Data
3	125	Aan ajen ajen	3.2	Sangat Memuaskan	Delete Data	Update Data

Showing 1 to 3 of 3 entries

Previous 1 Next

Mata Kuliah APAP

Latihan 4:

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

Method kurang lebih sama dengan Student untuk ServiceRest, DAO, dan DAOImpl nya. Kemudian membuat course/viewall pada course ccontroller untuk mengambil view yang dituju

```
CourseService... CourseDAO.java CourseService... CourseMappe... CourseDAOImp
1 package com.example.service;
2
3 import java.util.List;
16
17 @Slf4j
18 @Service
19 @Primary
20 public class CourseServiceRest implements CourseService
21 {
22     @Autowired
23     private CourseDAO courseDAO;
24
25     @Override
26     public CourseModel selectCourse(String id_course) {
27         log.info("REST - select course with id_course {}", id_course);
28         return courseDAO.selectCourse(id_course);
29     }
30
31     @Override
32     public List<CourseModel> selectAllCourse() {
33         log.info("REST - select all course");
34         return courseDAO.selectAllCourse();
35     }
36
37 }
```

```
CourseService... CourseDAO.java CourseService...
1 package com.example.dao;
2
3 import java.util.List;
4
5 import com.example.model.CourseModel;
6
7 public interface CourseDAO {
8     CourseModel selectCourse (String id);
9     List<CourseModel> selectAllCourse();
10 }
11 }
```

(viewall-course), setelah itu method dapat mengembalikan view dan viewall dari localhost:8080/rest/course/view dan localhost:8080/rest/course/viewall yang ada pada Producer (CourseRestController)

Abdalla Chair
1506757831
APAP - C

CourseDAOImpl:

```
CourseServic... CourseDAO.java CourseDAOImpl... CourseServic... CourseMappe... viewall.html CourseContro... viewall-cou...
2
3 import java.util.List;
14
15 @Service
16 public class CourseDAOImpl implements CourseDAO
17 {
18     @Autowired
19     private RestTemplate restTemplate;
20
21     @Override
22     public CourseModel selectCourse (String id_course)
23     {
24         CourseModel course =
25             restTemplate.getForObject("http://localhost:8080/rest/course/view/"+id_course,CourseModel.class);
26         return course;
27     }
28
29     @Override
30     public List<CourseModel> selectAllCourse()
31     {
32         List<CourseModel> courses = restTemplate.getForObject("http://localhost:8080/rest/course/viewall", List.class);
33         return courses;
34     }
35 }
36
```

Viewall-course:

```
CourseServic... CourseDAO.java CourseDAOImpl... viewall-cou... CourseServic... CourseM...
5     <div th:replace="fragments/fragment :: links"></div>
6 </head>
7 <body>
8     <div th:replace="fragments/fragment :: header"></div>
9     <h1>All Courses</h1>
10     <table id="table_id" class="display">
11         <thead>
12             <tr>
13                 <th>No</th>
14                 <th>ID</th>
15                 <th>Name</th>
16                 <th>Credits</th>
17             </tr>
18         </thead>
19         <tbody>
20             <div th:each="course,iterationStatus: ${courses}">
21                 <tr>
22                     <td><h3 th:text="${iterationStatus.count}">No. 1</h3></td>
23                     <td><h3 th:text="${course.idCourse}">Student NPM</h3></td>
24                     <td><h3 th:text="${course.name}">Student Name</h3></td>
25                     <td><h3 th:text="${course.credits}">Student GPA</h3></td>
26                 </tr>
27             </div>
28         </tbody>
29     </table>
```


localhost:8080/rest/cour x

localhost / 127.0.0.1 / e x

Json Parser Online x

localhost:8080/rest/course/view/CSC123

```
{"idCourse": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "Bila", "gpa": 3.5, "courses": null}]}
```

View Course x

localhost / 127.0.0.1 / e x

localhost:9090/course/view/CSC123

[Navbar](#)

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

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

ID = CSC123

Name = PSP

Credits = 4

Mahasiswa yang mengambil

- 124-Bila

Mata Kuliah APAP

The screenshot displays two browser windows. The top window shows a REST client at `localhost:8080/rest/course/viewall` with a JSON response containing course and student data. The bottom window shows a web application at `localhost:9090/course/viewall` with a navigation bar, a search bar, and a table of courses.

REST API Response (Top Window):

```
[{"idCourse": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "Bila", "gpa": 3.5, "courses": null}]}, {"idCourse": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "Bila", "gpa": 3.5, "courses": null}]}, {"idCourse": "CSC125", "name": "DDP 1", "credits": 4, "students": []}, {"idCourse": "CSC126", "name": "MPKT", "credits": 6, "students": [{"npm": "123", "name": "Nabilah", "gpa": 3.5, "courses": null}]}
```

Web Application (Bottom Window):

Navbar

- Home
- Daftar Mahasiswa
- Menambah Mahasiswa

All Courses

Show entries

Search:

No	ID	Name	Credits
1	CSC123	PSP	4
2	CSC124	SDA	3
3	CSC125	DDP 1	4
4	CSC126	MPKT	6

Showing 1 to 4 of 4 entries

Previous 1 Next

Mata Kuliah APAP

LESSON LEARNED:

Pada tutorial kali ini, saya belajar mengenai penggunaan web service dalam framework springboot, dengan membagikan aplikasi menjadi 2 layer, yaitu layer consumer dan layer producer, dimana bertujuan untuk memisahkan backend dan front end. Untuk producer merupakan bagian backend dan consumer bagian front end. Hal ini bisa diwujudkan dengan memahami *RestTemplate* yang difasilitasi oleh springboot.