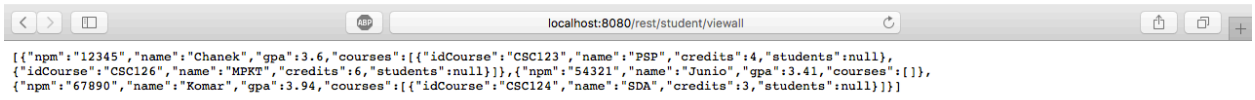
LESSON LEARNED

Hal yang saya pelajari dari tutorial kali ini adalah saya mempelajari penggunaan webservice untuk pengembangan sebuah sistem berskala enterprise. Pada tutorial ini dijelaskan bahwa terdapat dua layer, yaitu backend layer (service producer) yang menyediakan data dan frontend layer (service consumer) yang menampilkan data. Komunikasi antara kedua layer tersebut bisa dilakukan dengan menggunakan webservice dengan mengembalikan data dalam bentuk JSON.

LATIHAN

Latihan 1:



String parse


```
[
  {
    "npm":"12345",
    "name":"Chanek",
    "gpa":3.6,
    "courses":[2]
  },
  {
    "npm":"54321",
    "name":"Junio",
    "gpa":3.41,
    "courses":[0]
  },
  {
    "npm":"67890",
    "name":"Komar",
    "gpa":3.94,
    "courses":[
      {
        "idCourse":"CSC124",
        "name":"SDA",
        "credits":3,
        "students":null
      }
    ]
  }
]
```

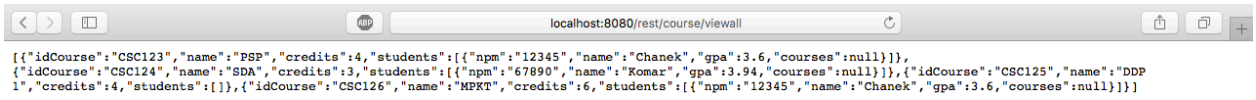

Pada StudentRestController.java di service producer, tambahkan method baru seperti berikut:

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

Method tersebut akan mengembalikan data seluruh student dalam bentuk JSON jika mengakses http://localhost:8080/rest/student/viewall.

Latihan 2:

localhost:8080/rest/course/viewall



String parse


```
[
  {
    "idCourse":"CSC123",
    "name":"PSP",
    "credits":4,
    "students":[1]
  },
  {
    "idCourse":"CSC124",
    "name":"SDA",
    "credits":3,
    "students":[1]
  },
  {
    "idCourse":"CSC125",
    "name":"DDP 1",
    "credits":4,
    "students":[0]
  },
  {
    "idCourse":"CSC126",
    "name":"MPKT",
    "credits":6,
    "students":[1]
  }
]
```


Langkah pertama yang dilakukan adalah membuat CourseRestController.java pada package com.tutorial7.rest dengan konten seperti berikut:

```
package com.tutorial7.rest;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;
import com.tutorial7.model.CourseModel;
import com.tutorial7.service.CourseService;

@RestController
@RequestMapping("/rest")
public class CourseRestController {
  @Autowired
  CourseService courseDAO;

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

Method viewall() tersebut akan mengembalikan data seluruh course dalam bentuk JSON jika mengakses http://localhost:8080/rest/course/viewall.

localhost:8080/rest/course/view/{id}

String parse


```
{
  "idCourse":"CSC123",
  "name":"PSP",
  "credits":4,
  "students":[
    {
      "npm":"12345",
      "name":"Chanek",
      "gpa":3.6,
      "courses":null
    }
  ]
}
```


Pada file yang sama, tambahkan juga method berikut untuk mengembalikan data terkait suatu course ketika mengakses http://localhost:8080/rest/course/view/{id} :

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

Latihan 3:

Pada StudentDAOImpl.java di service consumer, lengkapi method selectAllStudents() dengan kode berikut:

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

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

Kemudian lengkapi kode method selectAllStudents() pada StudentServiceRest.java dengan kode berikut:

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

Untuk mengakses fitur ini gunakan alamat berikut: http://localhost:9090/course/viewall

Latihan 4:

Langkah pertama adalah membuat CourseDAO.java dan CourseDAOImpl.java pada package com.tutorial7.mapper dengan kode seperti berikut:

CourseDAO.java

```
package com.tutorial7.mapper;

import java.util.List;

import com.tutorial7.model.CourseModel;

public interface CourseDAO {
  CourseModel selectCourse(String id);
  List<CourseModel> selectAllCourses();
}
```

CourseDAOImpl.java

```
package com.tutorial7.mapper;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Primary;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

import com.tutorial7.model.CourseModel;

import lombok.extern.slf4j.Slf4j;

@Slf4j
@Service
@Primary
public class CourseDAOImpl implements CourseDAO {
  @Autowired
  private RestTemplate restTemplate;

  @Bean
```

```
  public RestTemplate restTemplate() {
    return new RestTemplate();
  }

  @Override
  public CourseModel selectCourse(String id) {
    log.info("REST - select course with id {}", id);
    CourseModel course =
restTemplate.getForObject("http://localhost:8080/rest/course/view/"+id, CourseModel.class);
    return course;
  }

  @Override
  public List<CourseModel> selectAllCourses() {
    log.info("REST - select all courses");
    List<CourseModel> courses =
restTemplate.getForObject("http://localhost:8080/rest/course/viewall", List.class);
    return courses;
  }
}
```

Kemudian buat CourseServiceRest.java untuk mengakses data-data tersebut agar dapat digunakan di controller dengan kode berikut:

```
package com.tutorial7.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Primary;
import org.springframework.stereotype.Service;

import com.tutorial7.mapper.CourseDAO;
import com.tutorial7.model.CourseModel;

import lombok.extern.slf4j.Slf4j;

@Slf4j
@Service
@Primary
public class CourseServiceRest implements CourseService {
  @Autowired
  private CourseDAO courseDAO;

  @Override
  public CourseModel selectCourse(String id) {
    log.info("select course with npm {}", id);
    return courseDAO.selectCourse(id);
  }

  @Override
  public List<CourseModel> selectAllCourses() {
    log.info("select all students");
    return courseDAO.selectAllCourses();
  }
}
```

Sebelum membuat method selectAllCourses() pada CourseServiceRest, saya harus menambahkan baris berikut ke dalam CourseService.java agar dapat mengakses seluruh data course:

```
List<CourseModel> selectAllCourses();
```

Selanjutnya tambahkan mapping untuk melihat daftar seluruh course pada CourseController.java di service consumer dengan kode berikut:

```
@RequestMapping("/course/viewall")
public String viewall(Model model) {
  List<CourseModel> courses = courseDAO.selectAllCourses();

  model.addAttribute("courses", courses);
  model.addAttribute("title", "View All Courses");

  return "viewall-courses";
}
```

Terakhir, tambahkan halaman viewall-courses untuk menampilkan seluruh data course:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>View All Courses</title>
</head>

<body>
  <h1>All Courses</h1>

  <div th:each="course,iterationStatus: ${courses}">
    <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
    <h3 th:text="'ID = ' + ${course.idCourse}">Course ID</h3>
    <h3 th:text="'Name = ' + ${course.name}">Course Name</h3>
    <h3 th:text="'Credits = ' + ${course.credits}">Course Credits</h3>
    <h3 th:text="'Mahasiswa yang mengambil:'">List Students</h3>
    <ul th:each="student, iterationStatus: ${course.students}">
      <li th:text="${student.npm} + ' - ' + ${student.name}">Student NPM - Student
        Name</li>
    </ul>
    <hr/>
  </div>

</body>
</html>
```

Untuk mengakses fitur ini gunakan alamat berikut: http://localhost:9090/course/viewall dan http://localhost:9090/course/view/{id}