Julio Jaya Handoyo
1506757440
Arsitektur dan Pemrograman Aplikasi Perusahaan – A

Tutorial 7 *Web Service* Menggunakan Spring Boot

Hal yang saya pelajari dari tutorial ini adalah melakukan pemisahan *layer backend* (*service producer*) dan *frontedn* (*service consumer*). *Service consumer* merupakan aplikasi yang berinteraksi dengan pengguna, sedangkan *service producer* merupakan aplikasi yang memberikan data kepada *service consumer* berdasarkan permintaan *service consumer*. Melalui pemisahan ini, aplikasi *service consumer* lebih fokus untuk menyediakan dan mengolah data ke pengguna tanpa menggunakan *database*, sedangkan *service producer* hanya bertanggung jawab menyediakan data dari *database* dan biasanya tidak memiliki view yang dapat dilihat pengguna. Agar dapat berkomunikasi dengan *service consumer*, *service producer* menyediakan *web service* yang dapat dikonsumsi oleh *service consumer*. *Web service* merupakan URL yang akan mengembalikkan data dalam representasi JSON atau XML. Dalam tutorial ini, saya juga mempelajari penggunaan *web service* yang mengembalikkan data dalam representasi JSON.

Latihan Service Producer

1. Buatlah *service* untuk mengembalikkan seluruh *student* yang ada di basis data. *Service* ini mirip seperti *method* viewAll di Web Controller. *Service* tersebut di-*mapping* ke "/rest/student/viewall".

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

    @Autowired
    StudentService studentService;

    @RequestMapping("/student/view/{npm}")
    public StudentModel view(@PathVariable(value = "npm") String npm) {
        StudentModel student = studentService.selectStudent(npm);
        return student;
    }

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

        return students;
    }
}
```

```
@Select("select npm, name, gpa from student where npm = #{npm}")
@Results(value = {
        @Result(property="npm", column="npm"),
        @Result(property="name", column="name"),
        @Result(property="gpa", column="gpa"),
        @Result(property="courses", column="npm",
        javaType = List.class,
        many= @Many(select="selectCourses"))
})
StudentModel selectStudent (@Param("npm") String npm);

@Select("select npm,name,gpa from student")
@Results(value = {
        @Result(property="npm", column="npm"),
        @Result(property="name", column="name"),
        @Result(property="gpa", column="gpa"),
        @Result(property="courses", column="npm",
        javaType = List.class,
        many = @Many(select="selectCourses"))
})
List<StudentModel> selectAllStudents ();

@Select("select course.id_course, name, credits from studentcourse join course on studentcourse.
@Results(value = {
        @Result(property="idCourse", column="id_course"),
        @Result(property="name", column="name"),
        @Result(property="credits",column="credits")
})
List<CourseModel> selectCourses(@Param("npm") String npm);
```

Saya membuat method view yang mengembalikkan list berisi semua mahasiswa yang ada di database pada RestController. Method ini akan meyimpan semua mahasiswa yang ada di database ke students list dan mengembalikkan students list tersebut. Saya juga menambahkan method selectCourses pada StudentMapper yang hanya mengambil id, nama, dan credit course agar setiap course yang berada dalam mahasiswa tidak memunculkan student pada web service

localhost:8080/rest/student/viewall


```
[{"npm":"122","name":"Julio","gpa":3.8,"courses":[]},
{"npm":"123","name":"Chanek","gpa":3.6,"courses":
[{"idCourse":"CSC126","name":"MPKT","credits":6,"students":null}]},
{"npm":"124","name":"Chanek Junior","gpa":3.3,"courses":
[{"idCourse":"CSC123","name":"PSP","credits":4,"students":null},
{"idCourse":"CSC124","name":"SDA","credits":3,"students":null}]},
{"npm":"127","name":"David","gpa":3.4,"courses":[]}]
```

String parse


```
        "npm":"122",
        "name":"Julio",
        "gpa":3.8,
        "courses":[
        ]
    },
    {
        "npm":"123",
        "name":"Chanek",
        "gpa":3.6,
        "courses":[
            {
                "idCourse":"CSC126",
                "name":"MPKT",
                "credits":6,
                "students":null
            }
        ]
    },
    {
        "npm":"124",
        "name":"Chanek Junior",
        "gpa":3.3,
        "courses":[
            {
                "idCourse":"CSC123",
                "name":"PSP",
                "credits":4,
                "students":null
            },
            {
                "idCourse":"CSC124",
                "name":"SDA",
                "credits":3,
                "students":null
            }
        ]
    },
    {
        "npm":"127",
        "name":"David",
        "gpa":3.4,
        "courses":[
        ]
```


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).

```
    StudentModel selectStudent (String npm);

    List<StudentModel> selectAllStudents ();

    void addStudent (StudentModel student);

    void deleteStudent (String npm);

    void updateStudent(StudentModel student);

    CourseModel selectCourse (String idCourse);

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

```java
    @Select("select student.npm, name, gpa from studentcourse join student on studentcourse
    List<StudentModel> selectStudentCourse(@Param("id_course") String id_course);

    @Select("select id_course, name, credits from course where id_course = #{id_course}")
    @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="selectStudentCourse"))
    })
    CourseModel selectCourse(@Param("id_course") String id_course);

    @Select("SELECT * FROM course")
    @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="selectStudentsCourse"))
    })
    List<CourseModel> selectAllCourses();

    @Select("select student.npm, name, gpa from studentcourse join student on studentcourse
    @Results(value = {
            @Result(property="npm", column="npm"),
            @Result(property="name", column="name"),
            @Result(property="gpa", column="gpa")
    })
    List<StudentModel> selectStudentsCourse();
```

```java
package com.example.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.example.model.CourseModel;
import com.example.service.StudentService;

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

    @Autowired
    StudentService studentService;

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

        return course;
    }

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

        return courses;
    }
}
```

Saya menambahkan method selectAllCourses pada *Service* (method selectCourse sudah ada). Lalu, pada StudentMapper saya menambahkan method selectAllCourses(untuk mengambil semua course yang ada pada database) dan SelectStudentsCourse(mengambil data mahasiswa(npm, nama, gpa) yang mengikuti course tersebut tanpa mengambil data course yang diambil mahasiswa tersebut). Setelah itu, saya juga membuat RestController untuk Course. Di RestController tersebut, terdapat 2 method yang mengembalikkan course by id dan list semua course.

localhost:8080/rest/course/view/csc124


```
{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"Chanek
Junior","gpa":3.3,"courses":null}]}
```


String parse


```
{
    "idCourse":"CSC124",
    "name":"SDA",
    "credits":3,
    "students": [
        {
            "npm":"124",
            "name":"Chanek Junior",
            "gpa":3.3,
            "courses":null
        }
    ]
}
```


localhost:8080/rest/course/viewall


```
[{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek
Junior","gpa":3.3,"courses":null}]},{"idCourse":"CSC124","name":"SDA","credits":3,"students":
[{"npm":"124","name":"Chanek Junior","gpa":3.3,"courses":null}]},
{"idCourse":"CSC125","name":"DDP 1","credits":4,"students":[]},
{"idCourse":"CSC126","name":"MPKT","credits":6,"students":
[{"npm":"123","name":"Chanek","gpa":3.6,"courses":null}]}]
```

String parse


```
        "idCourse":"CSC123",
        "name":"PSP",
        "credits":4,
        "students":[
            {
                "npm":"124",
                "name":"Chanek Junior",
                "gpa":3.3,
                "courses":null
            }
        ]
    },
    {
        "idCourse":"CSC124",
        "name":"SDA",
        "credits":3,
        "students":[
            {
                "npm":"124",
                "name":"Chanek Junior",
                "gpa":3.3,
                "courses":null
            }
        ]
    },
    {
        "idCourse":"CSC125",
        "name":"DDP 1",
        "credits":4,
        "students":[
        ]
    },
    {
        "idCourse":"CSC126",
        "name":"MPKT",
        "credits":6,
        "students":[
            {
                "npm":"123",
                "name":"Chanek",
                "gpa":3.6,
                "courses":null
            }
        ]
```


Latihan Service Consumer

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

```java
package com.example.dao;

import java.util.List;

@Service
public class StudentDAOImpl implements StudentDAO {

    @Autowired
    private RestTemplate restTemplate;

    @Override
    public StudentModel selectStudent(String npm) {
        // TODO Auto-generated method stub
        StudentModel student = restTemplate.getForObject("http://localhost:8080/rest/student/view/" + npm, StudentModel.class);
        return student;
    }

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

```java
import java.util.List;

@Slf4j
@Service
@Primary
public class StudentServiceRest implements StudentService {

    @Autowired
    private StudentDAO studentDAO;

    @Autowired
    private CourseDAO courseDAO;

    @Override
    public StudentModel selectStudent(String npm) {
        // TODO Auto-generated method stub
        log.info ("REST - select student with npm {}", npm);
        return studentDAO.selectStudent(npm);
    }

    @Override
    public List<StudentModel> selectAllStudents() {
        // TODO Auto-generated method stub
        log.info("REST - select all students");
        return studentDAO.selectAllStudents();
    }
```

Pad kelas StudentDAOImpl, saya menambahkan method selectAllStudents() yang berisi list students yang mengambil data JSON dari http://localhost:8080/rest/student/viewall dan simpan ke dalam list dan mengembalikkan list tersebut
Saya mengimplementasi method selectAllStudents() pada StudentServiceRest dengan mengembalikkan studentDAO yang memanggil method selectAllStudents yang ada pada kelas StudentDAO.

localhost:8080/rest/student/viewall


```
[{"npm":"122","name":"Julio","gpa":3.8,"courses":[]},
{"npm":"123","name":"Chanek","gpa":3.6,"courses":
[{"idCourse":"CSC126","name":"MPKT","credits":6,"students":null}]},
{"npm":"124","name":"Chanek Junior","gpa":3.3,"courses":
[{"idCourse":"CSC123","name":"PSP","credits":4,"students":null},
{"idCourse":"CSC124","name":"SDA","credits":3,"students":null}]},
{"npm":"127","name":"David","gpa":3.4,"courses":[]}]
```


localhost:9090/student/viewall

All Students

Show 10 entries Search:

No	NPM	Name	GPA	Cum Laude	Delete	Update
1	122	Julio	3.8	Ya	Delete Data	Update Data
2	123	Chanek	3.6	Ya	Delete Data	Update Data
3	124	Chanek Junior	3.3	Tidak	Delete Data	Update Data
4	127	David	3.4	Tidak	Delete Data	Update Data

Showing 1 to 4 of 4 entries

Previous 1 Next

4. Implemetasikan *service consumer* untuk *class* CourseModel dengan membuat *class-class* DAO dan *service* baru.

```
package com.example.dao;

import java.util.Lis
Tutorial07Consumer/src/mai

public interface CourseDAO {
    CourseModel selectCourse(String id);

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

Saya membuat interface baru yang bernama CourseDAO dan berisi 2 method, yaitu selectCourse(id) dan selectAllCourses()

```java
package com.example.dao;

import java.util.List;

@Service
public class CourseDAOImpl implements CourseDAO {

    @Autowired
    private RestTemplate restTemplate;

    @Override
    public CourseModel selectCourse(String id) {
        // TODO Auto-generated method stub
        CourseModel course = restTemplate.getForObject("http://localhost:8080/rest/course/view/" + id, CourseModel.class);
        return course;
    }

    @Override
    public List<CourseModel> selectAllCourses() {
        // TODO Auto-generated method stub
        List<CourseModel> courses = restTemplate.getForObject("http://localhost:8080/rest/course/viewall", List.class);
        return courses;
    }
}
```

Kemudian saya membuat implementasi dari CourseDAO pada kelas CourseDAOImpl. Kelas ini berisi dua method selectCourse(id) dan selectAllCourses(). Kedua method ini mengambil data JSON dari web service dan menyimpannnya masing-masing ke course (“http://localhost:8080/rest/course/view/” + id) dan courses list(“http://localhost:8080/rest/course/viewall”). Method-method ini mengembalikkan course dan courses.

```java
package com.example.service;

import java.util.List;

public interface StudentService
{
    StudentModel selectStudent (String npm);

    List<StudentModel> selectAllStudents ();

    void addStudent (StudentModel student);

    void deleteStudent (String npm);

    void updateStudent(StudentModel student);

    CourseModel selectCourse (String idCourse);

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

Lalu, saya menambahkan method selectAllCourses() pada interface StudentService(method selectCourse sudah ada).

```java
package com.example.service;

import java.util.List;

@Slf4j
@Service
@Primary
public class StudentServiceRest implements StudentService {

    @Autowired
    private StudentDAO studentDAO;

    @Autowired
    private CourseDAO courseDAO;

    public StudentModel selectStudent(String npm) {

    public List<StudentModel> selectAllStudents() {

    @Override
    public CourseModel selectCourse(String idCourse) {
        // TODO Auto-generated method stub
        log.info ("REST - select course with id {}", idCourse);
        return courseDAO.selectCourse(idCourse);
    }

    @Override
    public List<CourseModel> selectAllCourses() {
        // TODO Auto-generated method stub
        log.info ("REST - select all courses");
        return courseDAO.selectAllCourses();
    }
```

Lalu, saya mengimplementasikan method selectCourse dan selectAllCourses pada StudentServiceRest. Kedua method ini masing-masing mengembalikkan courseDAO yang memanggil method selectCourse(id) dan courseDAO yang memanggil method selectAllCourses()

```java
    @RequestMapping("/course/viewall")
    public String viewAllCourses(Model model) {
        List<CourseModel> courses = studentDAO.selectAllCourses();
        model.addAttribute("courses", courses);
        return "viewall-course";
    }
}
```

Kemudian, saya menambahkan method viewAllCourses pada controller.

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head th:replace="fragments/fragment :: headerFragment(pageTitle='View All Courses')">
    </head>
    <link rel="stylesheet" href="/css/datatables.min.css"/>
    <link rel = "stylesheet" href ="/css/bootstrap.min.css"/>
    <body>
        <h1>All Courses</h1>
        <table id="coursesTable" class="display" width="100%">
            <thead>
                <tr>
                    <th>No</th>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Credit</th>
                </tr>
            </thead>
            <tbody>
                <tr th:each="course, iterationStatus: ${courses}">
                    <td th:text="${iterationStatus.count}">No</td>
                    <td th:text="${course.idCourse}"></td>
                    <td th:text="${course.name}"></td>
                    <td th:text="${course.credits}"></td>
                </tr>
            </tbody>
        </table>
        <script type="text/javascript" src="/js/jquery-3.2.1.min.js"></script>
        <script type="text/javascript" src="/js/datatables.min.js"></script>
        <script type="text/javascript" src="/js/bootstrap.min.js"></script>
        <script>
            $(document).ready(function(){
                $('#coursesTable').DataTable();
            });
        </script>
```

Saya menambahkan file baru bernama viewall-course yang digunakan untuk menampikan data kepada pengguna

localhost:8080/rest/course/view/csc124


```
{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"Chanek Junior","gpa":3.3,"courses":null}]}
```


localhost:9090/course/view/csc124

ID = CSC124

Name = SDA

Credits = 3

Mahasiswa yang mengambil

- 124 - Chanek Junior

localhost:8080/rest/course/viewall


```
[{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Chanek
Junior","gpa":3.3,"courses":null}]},{"idCourse":"CSC124","name":"SDA","credits":3,"students":
[{"npm":"124","name":"Chanek Junior","gpa":3.3,"courses":null}]},
{"idCourse":"CSC125","name":"DDP 1","credits":4,"students":[]},
{"idCourse":"CSC126","name":"MPKT","credits":6,"students":
[{"npm":"123","name":"Chanek","gpa":3.6,"courses":null}]}]
```


localhost:9090/course/viewall

All Courses

Show 10 entries

Search:

No	ID	Name	Credit
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