MUHAMMAD AKBAR SETIADI
APAP A
1506737810

Yang Saya Pelajari

Saya mempelajari bagaimana cara membuat Rest API pada Java Spring. Dimana pada tutorial ini dilakukan di Project Tutorial07Producer. Kemudian saya juga belajar bagaimana menangkap dan menggunakan Rest yang dibuat oleh project lain, dimana pada tutorial ini dilakukan di project Tutorial07Consumer.

Latihan 1

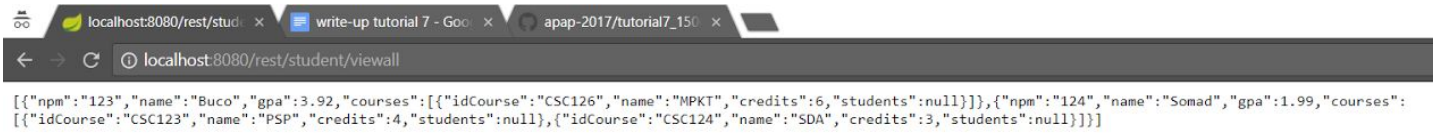
1. Pada class StudentRestController.java saya menambahkan method di bawah ini

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

method selectAllStudents() sebelumnya sudah saya buat untuk melakukan query select seluruh tuple yang ada di dalam table student

```
@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 ();
```

2. Kemudian menjalankan program di browser, dan di bawah ini tampilannya



[{"npm":"123","name":"Buco","gpa":3.92,"courses":[{"idCourse":"CSC126","name":"MPKT","credits":6,"students":null}]},{"npm":"124","name":"Somad","gpa":1.99,"courses":
[{"idCourse":"CSC123","name":"PSP","credits":4,"students":null},{"idCourse":"CSC124","name":"SDA","credits":3,"students":null}]}]

Latihan 2

1. Sebelumnya saya sudah membuat method selectCourse() untuk melakukan query select pada table course berdasarkan id course.

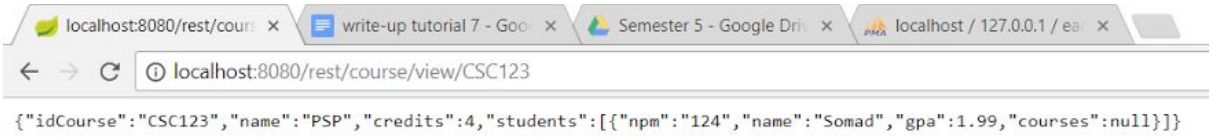
```
@Select ("select id_course,name,credits from course where id_course = #{id_course}")
@Results (value = {
        @Result (property = "idCourse" , column = "id_course" ),
        @Result (property = "name" , column = "name" ),
        @Result (property = "credits" , column = "credits" ),
        @Result (property = "students" , column = "id_course", javaType = List.class,
        many=@Many (select = "selectStudents"))
})
CourseModel selectCourse (@Param("id_course") String id_course);
```

2. Pada class StudentRestController.java saya menambahkan method di bawah ini

MUHAMMAD AKBAR SETIADI
APAP A
1506737810

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

3. Menjalankan program di website



4. Untuk viewall, saya menambahkan method selectAllCourses() pada interface StudentMapper.java

```
@Select("select id_course, name, credits 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 = "selectStudents"))
})
List<CourseModel> selectAllCourses ();
```

5. Kemudian menambahkan method tersebut di dalam interface StudentService.java dan class StudentServiceDatabase.java

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

6. Pada class StudentRestController.java saya menambahkan method di bawah ini

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

7. Menjalankan program di browser (http://localhost:8080/rest/course/viewall)

Latihan 3

MUHAMMAD AKBAR SETIADI
APAP A
1506737810

1. Di dalam class StudentDAOImpl.java, saya melakukan implementasi method hampir mirip dengan method selectStudent(). Yang berbeda adalah alamat url REST dan response yang diterima.

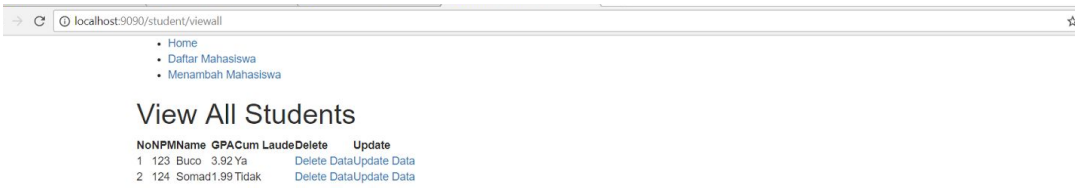
```
@Override
public List<StudentModel> selectAllStudents() {
    List<StudentModel> allStudents = restTemplate.getForObject("http://localhost:8080/rest/student/viewall", List.class);
    return allStudents;
}
```

2. Dan yang terakhir di dalam class StudentServiceRest.java, saya melakukan implementasi method selectAllStudents() seperti di bawah ini.

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

Saya cukup me *return* method yang ada di class StudentDAOImpl.java yang ada pada nomor 1

3. Kemudian menjalankan program di browser dan berikut hasilnya.



Latihan 4

Pada latihan ini sauya membuat 2 service, yaitu melakukan select Course berdasarkan id Course dan melakukan select semua Course yang ada. Berikut langkah-langkah yang saya lakukan.

1. Menambahkan interface CourseDAO.java di dalam package DAO dan membuat 2 method di interface tersebut, yaitu selectCourse() untuk mengambil course berdasarkan id course dan selectAllCourses() untuk mengambil semua course yang ada.

MUHAMMAD AKBAR SETIADI
APAP A
1506737810

```java
package com.example.dao;

import java.util.List;

import com.example.model.CourseModel;

public interface CourseDAO {
    //Implementasi pada class Course
        CourseModel selectCourse(String idCourse);
        List<CourseModel> selectAllCourses();
}
```

2. Membuat class CourseDAOImpl.java dengan mengimplementasikan interface CourseDAO.java di atas.

```java
@Service
public class CourseDAOImpl implements CourseDAO {
    @Autowired
    private RestTemplate restTemplate;

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

    @Override
    public List<CourseModel> selectAllCourses() {
        List<CourseModel> allCourses = restTemplate.getForObject("http://localhost:8080/rest/course/viewall", List.class);
        return allCourses;
    }
}
```

3. Pada interface StudentService.java yang sudah dibuat sebelumnya, belum terdapat method selectAllCourses(), sehingga saya menambahkan method tersebut di interface StudentService.java.

```java
import java.util.List;

public interface StudentService
{
    StudentModel selectStudent (String npm);

    List<StudentModel> selectAllStudents ();

    CourseModel selectCourse (String idCourse);

    List<CourseModel> selectAllCourses();

    void addStudent (StudentModel student);

    void deleteStudent (String npm);

    void updateStudent(StudentModel student);
}
```

4. Mengimplementasikan method di atas pada class StudentServiceDatabase.java dan StudentServiceRest.java. Karena yang digunakan adalah class StudentServiceRest.java, maka method tersebut yang ada di class StudentServiceDatabase.java saya *return null*. Sedangkan pada class StudentServiceRest.java, saya lakukan implementasi seperti di bawah ini.

MUHAMMAD AKBAR SETIADI
APAP A
1506737810

```java
@Slf4j
@Service
@Primary
public class StudentServiceRest implements StudentService {
    @Autowired
    private StudentDAO studentDAO;

    @Autowired
    private CourseDAO courseDAO;

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

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

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

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


    @Override
    public void addStudent(StudentModel student) {
        // TODO Auto-generated method stub

    }

    @Override
    public void deleteStudent(String npm) {
        // TODO Auto-generated method stub

    }

    @Override
    public void updateStudent(StudentModel student) {
        // TODO Auto-generated method stub

    }
}
```

5. Di dalam controller, belum terdapat service untuk select semua course. Untuk itu, saya menambahkan method untuk itu yaitu method viewAllCourse().

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

    if (course != null) {
        model.addAttribute ("course", course);
        model.addAttribute("title","View Course");
        return "view-course";
    } else {
        model.addAttribute ("id_course", id_course);
        model.addAttribute("title","Not Found!");
        return "not-found";
    }
}

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

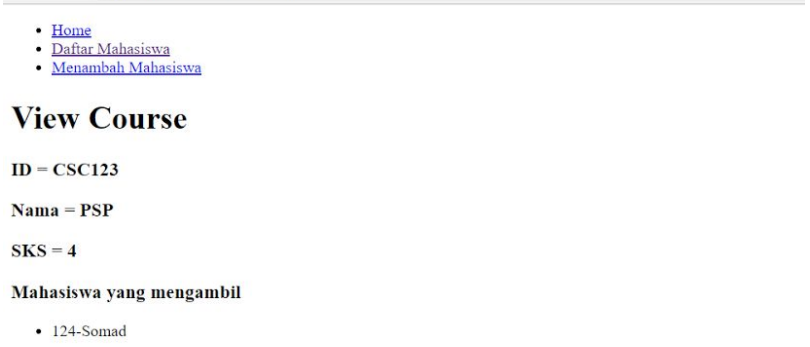
6. Pada resources, saya membuat file course-viewall.html yang berisi di bawah ini.

MUHAMMAD AKBAR SETIADI
APAP A
1506737810

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View All Students</title>
        <meta charset="utf-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1" />
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" />
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
        <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
        <link rel="stylesheet" type="text/css" href="//cdn.datatables.net/1.10.16/css/jquery.dataTables.css" />
        <script type="text/javascript" charset="utf8" src="//cdn.datatables.net/1.10.16/js/jquery.dataTables.js"></script>
    </head>
    <body>
        <div class="container">
            <div th:replace = "fragments/fragment :: header" ></div>
            <table id="table_id" class="display">
                <thead>
                    <tr>
                        <th>No</th>
                        <th>ID Course</th>
                        <th>Name</th>
                        <th>Credits</th>
                    </tr>
                </thead>
                <tbody>
                    <tr th:each="course,iterationStatus: ${courses}">
                        <td th:text="${iterationStatus.index +1}">No</td>
                        <td th:text="${course.idCourse}"></td>
                        <td th:text="${course.name}"></td>
                        <td th:text="${course.credits}"></td>
                    </tr>
                </tbody>
            </table>
        </div>
    </body>
</html>
```

7. Kemudian menjalankan kedua service tersebut.

a. Select course dengan id CSC123



b. Select semua course yang ada