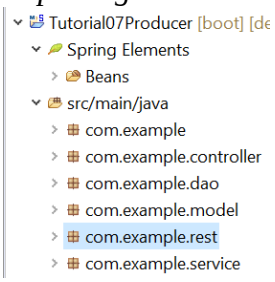
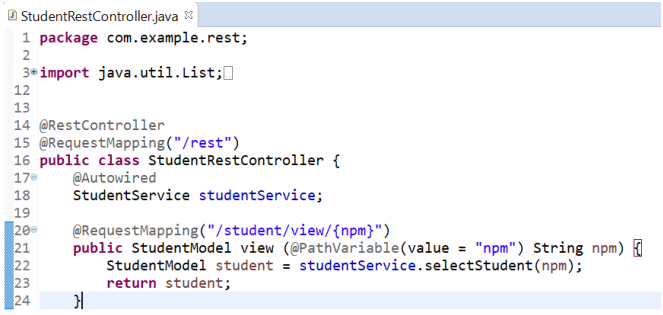
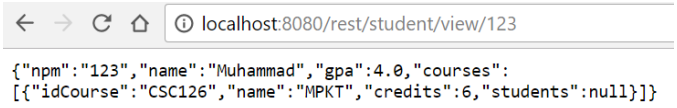
Tutorial 7 Write Up

Muhammad Alvin Abyan – 1506721844 – APAP [A]

MEMBUAT *SERVICE PRODUCER*

1. *Copy* berkas dari Tutorial 5 ke *project* Tutorial07Producer
2. Membuat *package* baru **com.example.rest**
   a. 

3. Membuat *class* baru **StudentRestController.java** pada *package* **com.example.rest** dengan method untuk view student
   a. 

4. Membuka **localhost:8080/rest/student/view/123**
   a. 

5. Membaca dengan **JSON View**
   a. 

6. *Commit* dan *Push* DONE

LATIHAN

1. Buat *service* untuk melihat seluruh student yang berada di basis data dan diakses melalui **/rest/student/viewall**

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


```
[{"npm":"123","name":"Muhammad","gpa":4.0,"courses":
[{"idCourse":"CSC126","name":"MPKT","credits":6,"students":null}]}
{"npm":"124","name":"Alvin","gpa":4.0,"courses":
[{"idCourse":"CSC123","name":"PSP","credits":4,"students":null},
{"idCourse":"CSC124","name":"SDA","credits":3,"students":null}]}]
```


- 

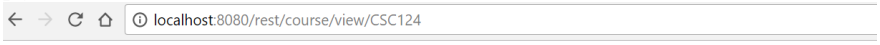
- Penjelasan: Membuat method *viewAll* dengan membuat inisiasi List of Student Model pada method tersebut yang berisi hasil kueri dari selectAllStudents() pada studentService

2. Buat *service* untuk melihat suatu course view by ID dan *service* untuk view all course

a. course view by ID

- CourseRestController.java

```
package com.example.rest;

import java.util.List;

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

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

- 

- {"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"Alvin","gpa":4.0,"courses":null}]}

String parse


```
{
    "idCourse":"CSC124",
    "name":"SDA",
    "credits":3,
    "students":[
        {
            "npm":"124",
            "name":"Alvin",
            "gpa":4.0,
            "courses":null
        }
    ]
}
```


- Penjelasan: Membuat object CourseModel yang berisi hasil dari selectCourse pada courseService

b. view all course

-
```
@RequestMapping("/course/viewall")
public List<CourseModel> viewAll () {
    List<CourseModel> allCourse = courseService.selectAllCourse();
    return allCourse;
}
```

-
```
@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="selectCourseStudent"))
    }) List<CourseModel> selectAllCourse();
```

-

- [{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Alvin","gpa":4.0,"courses":null}]},{"idCourse":"CSC124","name":"SDA","credits":3,"students"
[{"npm":"124","name":"Alvin","gpa":4.0,"courses":null}]},{"idCourse":"CSC125","name":"DDP 1","credits":4,"students":[]},
{"idCourse":"CSC126","name":"MPKT","credits":6,"students":[{"npm":"123","name":"Muhammad","gpa":4.0,"courses":null}]}]

String parse


```
[
    {
        "idCourse":"CSC123",
        "name":"PSP",
        "credits":4,
        "students":[
            {
                "npm":"124",
                "name":"Alvin",
                "gpa":4.0,
                "courses":null
            }
        ]
    },
    {
        "idCourse":"CSC124",
        "name":"SDA",
        "credits":3,
        "students":[
            {
                "npm":"124",
                "name":"Alvin",
                "gpa":4.0,
                "courses":null
            }
        ]
    },
    {
        "idCourse":"CSC125",
        "name":"DDP 1",
```

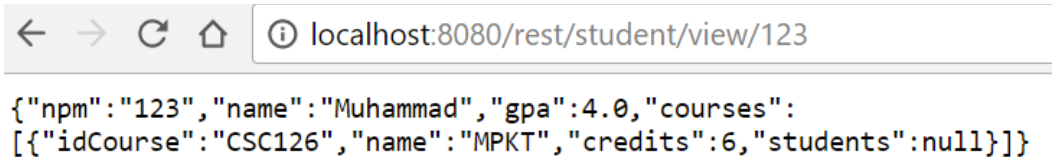
- Penjelasan: Membuat List of CourseModel yang menampung hasil kueri dari selectAllCourse() pada courseService

MEMBUAT *SERVICE CONSUMER*

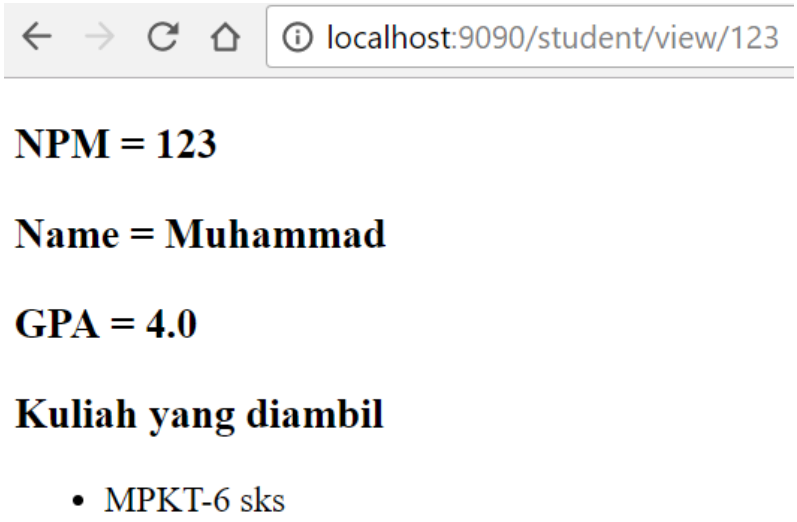
1. *Import* berkas-berkas Tutorial 6 untuk *project* Tutorial07Consumer DONE
2. Atur **application.properties** dengan menambahkan baris **server.port=9090** DONE
3. Tambahkan *interface* dengan nama **StudentDAO.java**
   a.
```
public interface StudentDAO {
    StudentModel selectStudent (String npm);
    List<StudentModel> selectAllStudent ();
}
```
4. Tambakan *class* dengan nama **StudentDAOImpl.java**
   a.
```
@Service
public class StudentDAOImpl implements StudentDAO {

    @Autowired
    private RestTemplate restTemplate;

    @Bean
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }

    @Override
    public StudentModel selectStudent(String npm) {
        StudentModel student = restTemplate.getForObject("http://localhost:8080/rest/student/view/"+npm, StudentModel.clas
        return student;
    }

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

}
```
5. Tambahkan *class* dengan **StudentServiceRest.java** di *package* Service
   a.
```
@Slf4j
@Service
@Primary
public class StudentServiceRest implements StudentService {

    @Autowired
    private StudentDAO studentDAO;

    @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 student");
        return studentDAO.selectAllStudent();
    }
```

6. Jalankan kedua *project* untuk *Service Producer* dan *Service Consumer*
DONE

7. Buka **localhost:8080/rest/student/view/123**

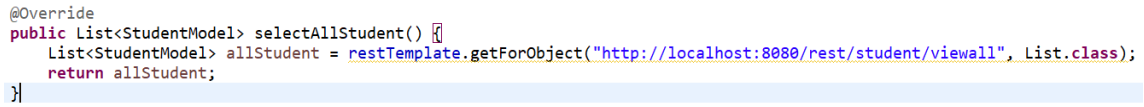
a. 

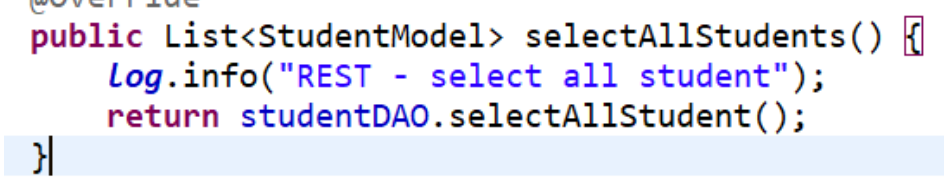
8. Buka **localhost:9090/student/view/123**

a. 

LATIHAN

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

a. 

b. 

c. Penjelasan: Membuat List of Student Model yang memperoleh data dari object yang berasal dari Rest Template **localhost:8080/rest/student/viewall** dan objectnya berupa List

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

a.
```
public interface CourseDAO {
    CourseModel selectCourse (String id_course);
    List<CourseModel> selectAllCourse ();
}
```

b.
```
@Service
public class CourseDAOImpl implements CourseDAO {

    @Autowired
    private RestTemplate restTemplate;

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

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

}
```

c.
```
@Slf4j
@Service
@Primary
public class CourseServiceRest implements CourseService {

    @Autowired
    private CourseDAO courseDAO;

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

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

d.

```
@Controller
public class CourseController {

    @Autowired
    CourseService courseDAO;
```

e.

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

    if (course != null) {
        model.addAttribute ("course", course);
        return "viewcourse";
    } else {
        model.addAttribute ("id_course", id_course);
        return "not-found";
    }
}

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

    if(courses != null) {
        model.addAttribute("courses", courses);
        return "viewallcourses";
    } else {
        return "not-found";
    }
}
```

f. Penjelasan: Membuat *interface* terlebih dahulu, kemudian membuat CourseDAOImpl yang mengimplementasikan CourseDAO yang menjadi tempat untuk mengambil object dari operasi REST menuju localhost:8080. Kemudian di kelas CourseServiceRest mengimplement CourseService agar bisa digunakan pada Controller. Terakhir, menambahkan method viewAll pada CourseController.

Lesson Learned: pada tutorial 7 ini saya mempelajari bagaimana cara menggunakan REST pada Spring dan juga cara bagaimana REST memiliki pembagian *Service Producer* dan *Service Consumer* untuk bertukar data