

## TUTORIAL 7 APAP

**Latihan 1: *Producer Service – View All Students***

Saya mengimplementasikan *method view all* untuk *student* dengan menambahkan *method* pada *class* *StudentRestController* yang memanggil *class service* dan menampung daftar mahasiswa tersebut pada *list of students*.

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

    return students;
}
```

**Latihan 2: *Producer Service – Course Service***

Saya mengimplementasikan *course service (view course by id & view all courses)* dengan membuat *class* *CourseRestController* yang *logic*-nya sama dengan *class* *StudentRestController*, namun berbeda pada tipe data, yakni menjadi *CourseModel*.

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

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

        return course;
    }

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

        return courses;
    }
}
```

**Latihan 3: Consumer Service – View All Students**

Saya melengkapi *method view all students* pada *class StudentServiceRest* dengan memanggil *method selectAllStudents* pada *class StudentDAOImpl*.

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

*Method selectAllStudents* tersebut akan mengambil daftar mahasiswa pada *producer service* dengan tipe *StudentModel[].class* yang kemudian akan dikonversi menjadi *List*.

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

    List<StudentModel> studentList;
    studentList = Arrays.asList(students);

    return studentList;
}
```

#### Latihan 4: Consumer Service – Course Service

Saya mengimplementasikan *course service* (*view course by id & view all courses*) dengan membuat *class* *CourseServiceRest* dan *CourseDAOImpl* yang *logic*-nya sama dengan *class* *StduentServiceRest* dan *StudentDAOImpl*, namun berbeda pada tipe data, yakni menjadi *CourseModel*.

##### *CourseServiceRest*

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

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

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

        return courseDAO.selectAllCourses();
    }
}
```

**CourseDAOImpl**

```

@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, CourseModel.class);

        return course;
    }

    @Override
    public List<CourseModel> selectAllCourses() {
        CourseModel[] courses =
            restTemplate.getForObject("http://localhost:8080/rest/course/viewa
                ll/", CourseModel[].class);

        List<CourseModel> courseList;
        courseList = Arrays.asList(courses);

        return courseList;
    }
}

```

**Lesson Learned**

Pada tutorial kali ini saya belajar hal baru mengenai REST Web Service; bagaimana cara membuat *consumer service* dan *producer service*, cara menghubungkan keduanya, dan lain-lain. Selain itu, tutorial ini juga semakin membuka pikiran saya bahwa skalabilitas dalam pemrograman itu sangat penting.