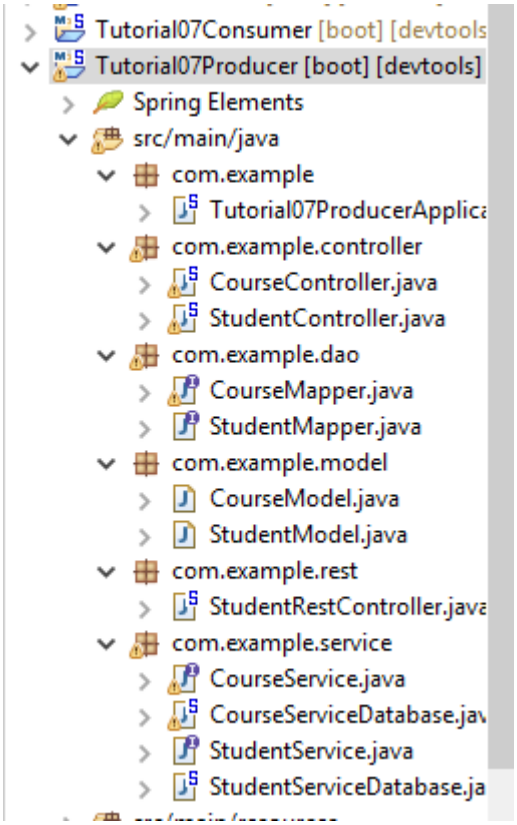
Anisha Eva Farida
1506689572
ADPAP – B

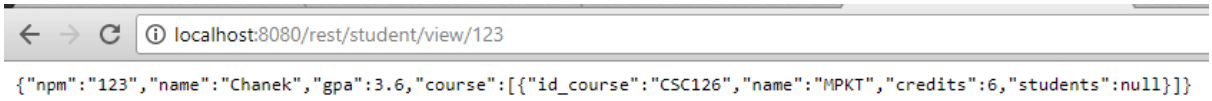
1. Membuat Service Producer



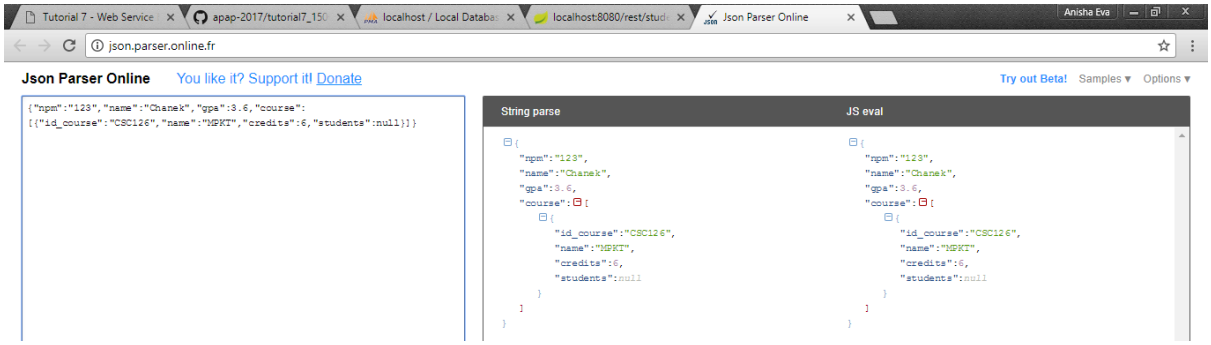
StudentRestController.java

```java
package com.example.rest;

import org.springframework.beans.factory.annotation.Autowired;

@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;
    }
}
```

Tampilan ketika membuka **localhost:8080/rest/student/view/123**



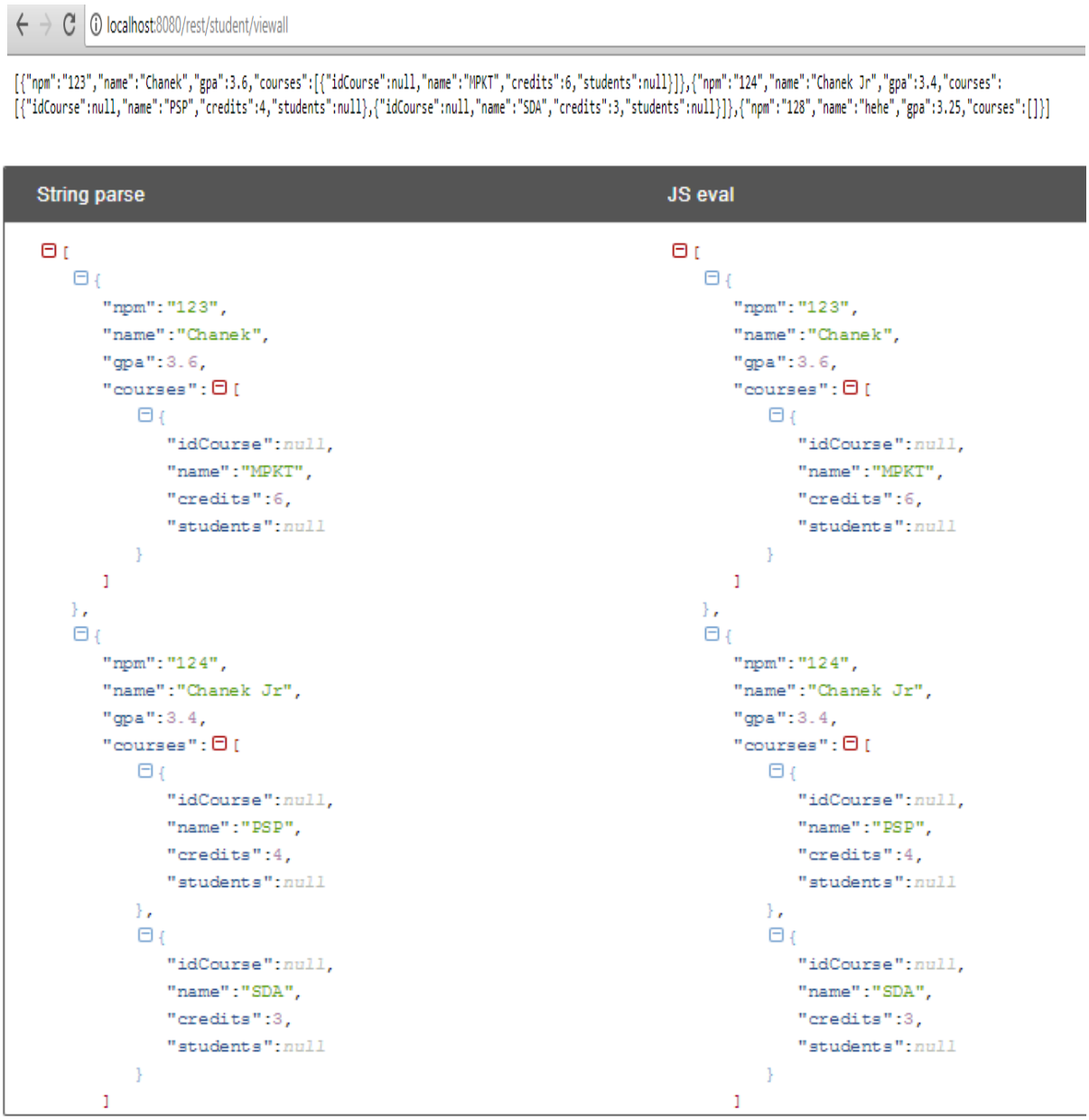
Tampilan menggunakan **JSON View**



Anisha Eva Farida
1506689572
ADPAP – B

LATIHAN

1. Viewall students dengan course yang diambil



Anisha Eva Farida
1506689572
ADPAP – B

StudentRestController.java

```java
package com.example.rest;

import java.util.List;

@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;
    }

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

viewall.html

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View All Students</title>
    </head>
    <body>
        <h1>All Students</h1>

        <div th:each="student,iterationStatus: ${students}">
            <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
            <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
            <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
            <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
            <a th:href="'/student/delete/' + ${student.npm}"> Delete Data </a> <br/>
            <a th:href="'/student/update/' + ${student.npm}"> Update Data </a> <br/>

            <h3>Kuliah yang diambil</h3>
                <ul th:each="course,iterationStatus: ${student.courses}">
                    <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
                        Nama Mata Kuliah - Jumlah SKS
                    </li>
                </ul>
```

Dnegan mengguankan method selectStudent yang telah digunakan pada tutorial sebelumnya kemudian kembaliannya berupa arraylist dari mahasiswa itu sendiri.

Anisha Eva Farida
1506689572
ADPAP – B

2. Service untuk kelas course dengan controller baru

view-course.html | CourseRestController.java

```java
package com.example.rest;

import java.util.List;

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

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

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

Anisha Eva Farida
1506689572
ADPAP – B

view-course.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title> View Course by ID </title>
    </head>
    <body>
        <h3 th:text="'ID = ' + ${course.idCourse}"> Course ID </h3>
        <h3 th:text="'Nama = ' + ${course.name}"> Course Name </h3>
        <h3 th:text="'SKS = ' + ${course.credits}"> Course Credit </h3>

        <h3> Mahasiswa yang mengambil </h3>
        <ul th:each="student,iterationStatus: ${course.students}">
            <li th:text="${student.npm} + '-' + ${student.name}">
            NPM - Nama Mahasiswa
            </li>
        </ul>
```

CourseController.java

```
    @Autowired
    CourseService courseDAO;

    @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-course";
        }
    }

    @RequestMapping("/course/viewall")
    public String view (Model model)
    {
        List<CourseModel> courses = courseDAO.selectAllCourse ();
        model.addAttribute ("courses", courses);

        return "view-course";
    }
```

Sama seperti method pada latihan 1, saya menggunakan method dari latihan sebelumnya kemudian untuk mahasiswa yang dimaksud di latihan 1 saya ubah menjadi coursenya.

Anisha Eva Farida
1506689572
ADPAP – B

Membuat Service Consumer

3.

StudentServiceRest.java | StudentDAOImpl.java

```
1  package com.example.service;
2
3  import java.util.List;
12
13 @Service
14 public class StudentDAOImpl implements StudentDAO
15 {
16     @Autowired
17     private RestTemplate restTemplate;
18
19     @Override
20     public StudentModel selectStudent (String npm)
21     {
22         StudentModel student =
23         restTemplate.getForObject("http://localhost:8080/rest/student/view/"+npm,StudentModel.class);
24         return student;
25     }
26
27     @Override
28     public List<StudentModel> selectAllStudents ()
29     {
30         List<StudentModel> students = restTemplate.getForObject("http://localhost:8080/rest/student/viewall", List.c
31         return students;
32     }
33 }
```

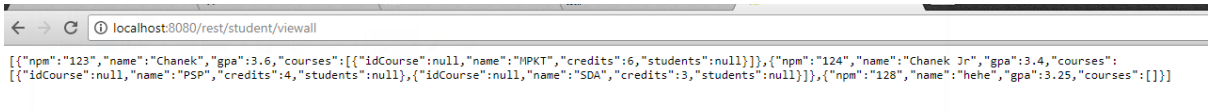
StudentServiceRest.java

```
12
13 import lombok.extern.slf4j.Slf4j;
14
15 @Slf4j
16 @Service
17 @Primary
18 public class StudentServiceRest implements StudentService
19 {
20     @Autowired
21     private StudentDAO studentDAO;
22
23     @Override
24     public StudentModel selectStudent (String npm)
25     {
26         log.info ("REST - select student with npm {}", npm);
27         return studentDAO.selectStudent (npm);
28     }
29
30     @Override
31     public List<StudentModel> selectAllStudents ()
32     {
33         log.info ("REST - select all students");
34         return studentDAO.selectAllStudents ();
35     }
36
```

localhost:8080/rest/student/view/123

{"npm":"123","name":"Chanek","gpa":3.6,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]}

Anisha Eva Farida
1506689572
ADPAP – B



Dengan melengkapi method selectAllStudents pada kelas StudentServiceRest kemudian data dapat ditampilkan.

4.

CourseService.java | CourseDAO.java

```java
package com.example.dao;

import java.util.List;

public interface CourseDAO {
    CourseModel selectCourse (String id);

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

CourseService.java

```java
package com.example.service;

import java.util.List;

public interface CourseService
{
    CourseModel selectCourse (String id_course);

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

Anisha Eva Farida
1506689572
ADPAP – B

CourseDAOImpl.java

```
1  package com.example.service;
2
3  import java.util.List;
12
13 @Service
14 public class CourseDAOImpl implements CourseDAO
15 {
16     @Autowired
17     private RestTemplate restTemplate;
18
19     @Override
20     public CourseModel selectCourse (String id_course)
21     {
22         CourseModel course =
23         restTemplate.getForObject("http://localhost:808
24         return course;
25     }
26
27     @Override
28     public List<CourseModel> selectAllCourse()
29     {
30         List<CourseModel> courses = restTemplate.getFor
31         return courses;
32     }
33 }
```

Anisha Eva Farida
1506689572
ADPAP – B

CourseDAOImpl.java | view-all-course.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View All Courses</title>
        <div th:replace="fragments/fragment :: links"></div>
    </head>
    <body>
        <div th:replace="fragments/fragment :: header"></div>
        <h1>All Courses</h1>
            <table class="display">
                <thead>
                    <tr>
                        <th>No</th>
                        <th>ID</th>
                        <th>Name</th>
                        <th>Credits</th>
                    </tr>
                </thead>
                <tbody>
                    <div th:each="course,iterationStatus: ${courses}">
                        <tr>
                            <td><h3 th:text="${iterationStatus.count}">No. 1</h3></td>
                            <td><h3 th:text="${course.idCourse}">Student NPM</h3></td>
                            <td><h3 th:text="${course.name}">Student Name</h3></td>
                            <td><h3 th:text="${course.credits}">Student GPA</h3></td>
                        </tr>
                    </div>
                </tbody>
            </table>
        <div th:replace="fragments/fragment :: footer"></div>
    </body>
</html>
```

Sama seperti latihan pada nomor 3 dan membuat file baru berupa view-al-course.html untuk tampilannya.

LESSON LEARNED

Saya belajar mengenai pembagian 2 layer yaitu consumer dan produser untuk memisahkan front dan back end dengan menggunakan resttemplate yang disediakan springboot.

Anisha Eva Farida
1506689572
ADPAP – B