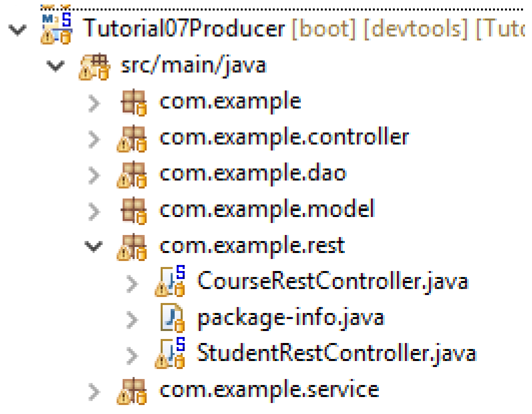
Nabilah Zhafira Viderisa Tutorial 7
1506721806
ADPAP – C

WRITE-UP TUTORIAL 7

Membuat Service producer

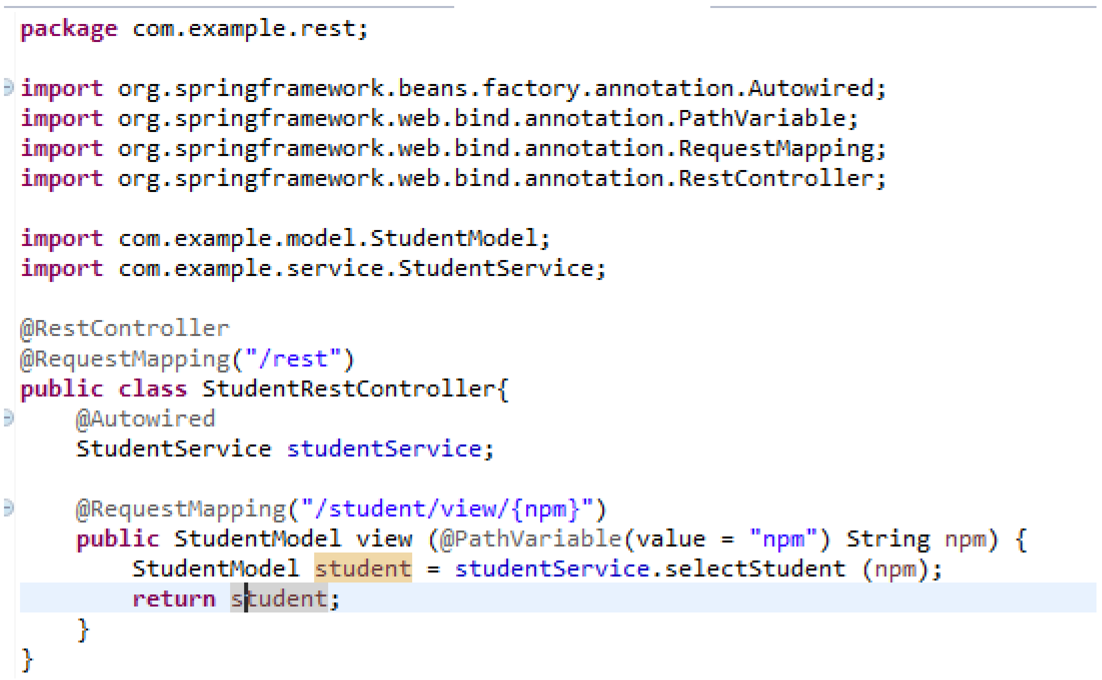
1. Import berkas-berkas yang ada di tutorial 5 untuk Service producer ini. Buatlah project 'Tutorial07Producer' dan silakan import satu folder atau import file satu per satu dari Tutorial05. Jika Anda langsung import satu project tutorial 5 jangan lupa untuk menghapus folder .git, namun jangan hapus .gitignore, dalam folder tersebut setelah dipindahkan menjadi folder Tutorial07Producer karena Anda tidak akan push dari direktori Tutorial07Producer.

2. Buat package baru com.example.rest, atau sesuaikan dengan package Anda masing-masing REST Controller Web Service diletakkan pada package terpisah, bukan pada package com.example.controller



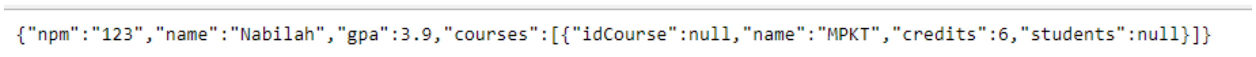
```
Tutorial07Producer [boot] [devtools] [Tut
  src/main/java
    com.example
    com.example.controller
    com.example.dao
    com.example.model
    com.example.rest
      CourseRestController.java
      package-info.java
      StudentRestController.java
    com.example.service
```

3. Buat class baru yaitu StudentRestController.java pada package com.example.rest

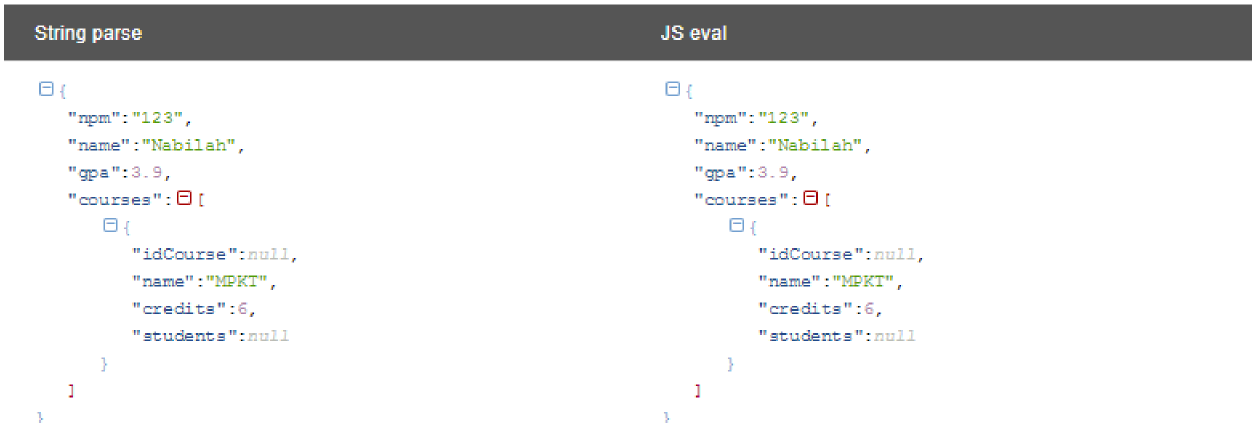


```java
package com.example.rest;

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.StudentModel;
import com.example.service.StudentService;

@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;
    }
}
```

http://localhost:8080/rest/student/view/123



```
{"npm":"123","name":"Nabilah","gpa":3.9,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]}
```



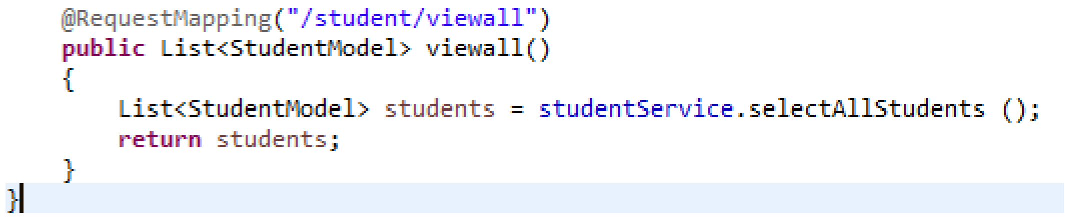
```
{
   "npm":"123",
   "name":"Nabilah",
   "gpa":3.9,
   "courses":[
      {
         "idCourse":null,
         "name":"MPKT",
         "credits":6,
         "students":null
      }
   ]
}
```

```
{
   "npm":"123",
   "name":"Nabilah",
   "gpa":3.9,
   "courses":[
      {
         "idCourse":null,
         "name":"MPKT",
         "credits":6,
         "students":null
      }
   ]
}
```

LATIHAN

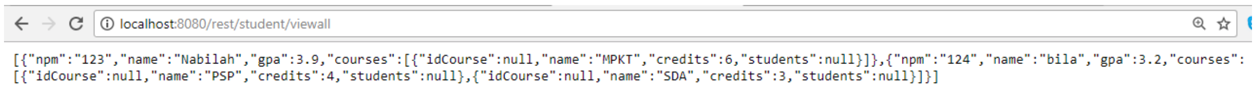
1. Latihan 1: Buatlah service untuk mengembalikan seluruh student yang ada di basis data. Service ini mirip seperti method viewAll di Web Controller. Service tersebut di-mapping ke “/rest/student/viewall”.

➔ Untuk membuat latihan nomor 1, saya menggunakan *method* selectAllStudents() yang sebelumnya sudah dibuat pada tutorial sebelumnya, namun hasil yang dikeluarkan merupakan arraylist dari students.



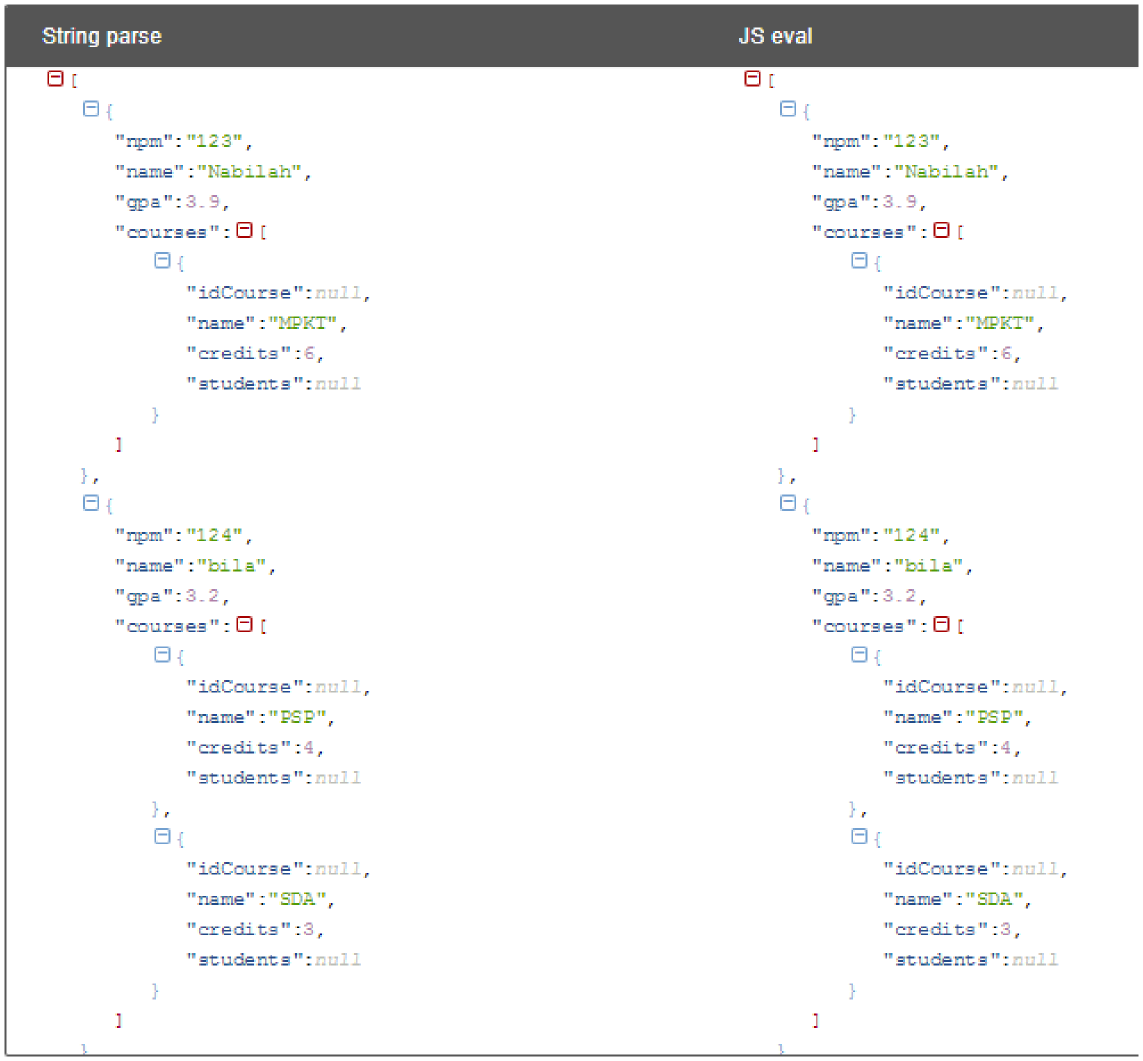
```
    @RequestMapping("/student/viewall")
    public List<StudentModel> viewall()
    {
        List<StudentModel> students = studentService.selectAllStudents ();
        return students;
    }
}
```

Hasil keluaran:



localhost:8080/rest/student/viewall

```
[{"npm":"123","name":"Nabilah","gpa":3.9,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]},{"npm":"124","name":"bila","gpa":3.2,"courses":
[{"idCourse":null,"name":"PSP","credits":4,"students":null},{"idCourse":null,"name":"SDA","credits":3,"students":null}]}]
```



String parse

```
[
   {
      "npm":"123",
      "name":"Nabilah",
      "gpa":3.9,
      "courses":[
         {
            "idCourse":null,
            "name":"MPKT",
            "credits":6,
            "students":null
         }
      ]
   },
   {
      "npm":"124",
      "name":"bila",
      "gpa":3.2,
      "courses":[
         {
            "idCourse":null,
            "name":"PSP",
            "credits":4,
            "students":null
         },
         {
            "idCourse":null,
            "name":"SDA",
            "credits":3,
            "students":null
         }
      ]
   }
```

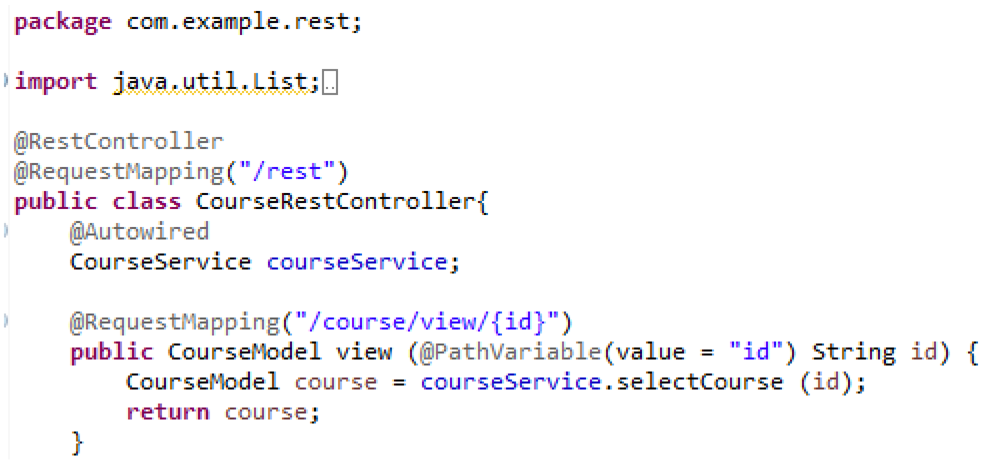
JS eval

```
[
   {
      "npm":"123",
      "name":"Nabilah",
      "gpa":3.9,
      "courses":[
         {
            "idCourse":null,
            "name":"MPKT",
            "credits":6,
            "students":null
         }
      ]
   },
   {
      "npm":"124",
      "name":"bila",
      "gpa":3.2,
      "courses":[
         {
            "idCourse":null,
            "name":"PSP",
            "credits":4,
            "students":null
         },
         {
            "idCourse":null,
            "name":"SDA",
            "credits":3,
            "students":null
         }
      ]
   }
```

2. Latihan 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).

➔ Untuk menampilkan *view by ID* saya membuat controller baru yaitu CourseRestController.java dengan isi yang hampir sama dengan StudentRestController.java. Hanya method untuk menampilkan courses ini mengembalikan courses dengan parameter *ID* courses tersebut

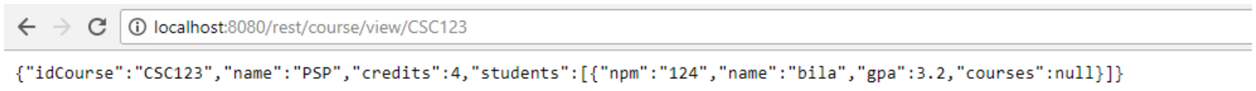
Nabilah Zhafira Viderisa Tutorial 7
1506721806
ADPAP – C



```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;
    }
```

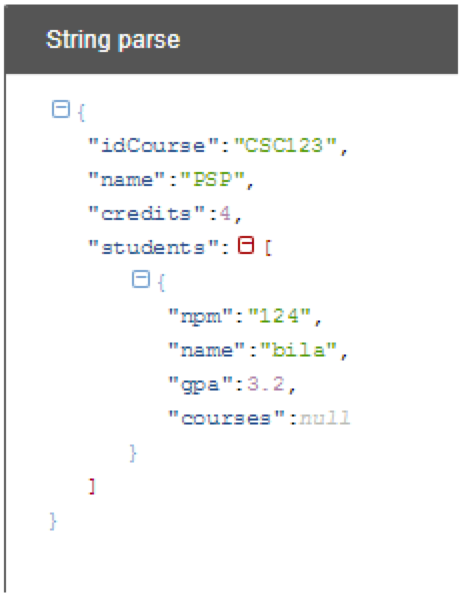
Hasil keluaran:

http://localhost:8080/rest/course/view/CSC123



localhost:8080/rest/course/view/CSC123

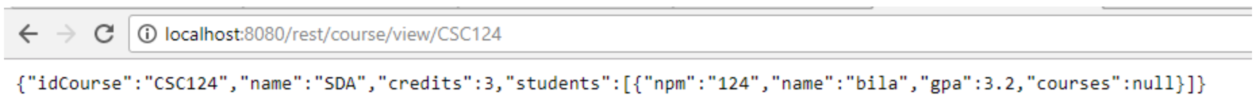
```
{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"bila","gpa":3.2,"courses":null}]}
```



String parse

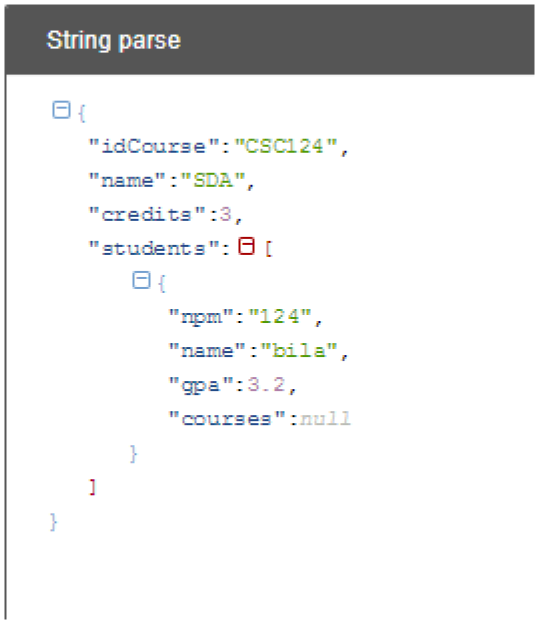
```
{
   "idCourse":"CSC123",
   "name":"PSP",
   "credits":4,
   "students":[
      {
         "npm":"124",
         "name":"bila",
         "gpa":3.2,
         "courses":null
      }
   ]
}
```

http://localhost:8080/rest/course/view/CSC124



localhost:8080/rest/course/view/CSC124

```
{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"bila","gpa":3.2,"courses":null}]}
```



String parse

```
{
    "idCourse":"CSC124",
    "name":"SDA",
    "credits":3,
    "students":[
        {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
        }
    ]
}
```

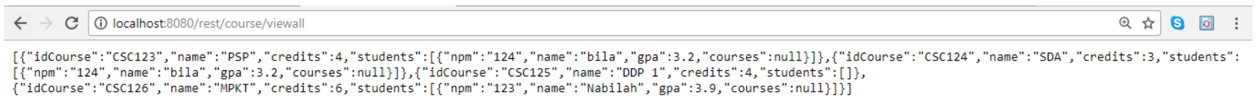
➔ Untuk menampilkan seluruh courses yang ada yaitu viewall courses, saya membuat method pada CourseMapper dan pada CourseService untuk menampilkan semua courses yang ada. Lalu saya menggunakannya pada CourseRestController.java untuk memanggil seluruh courses dengan cara menggunakan method selectAllCourses dari service.

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

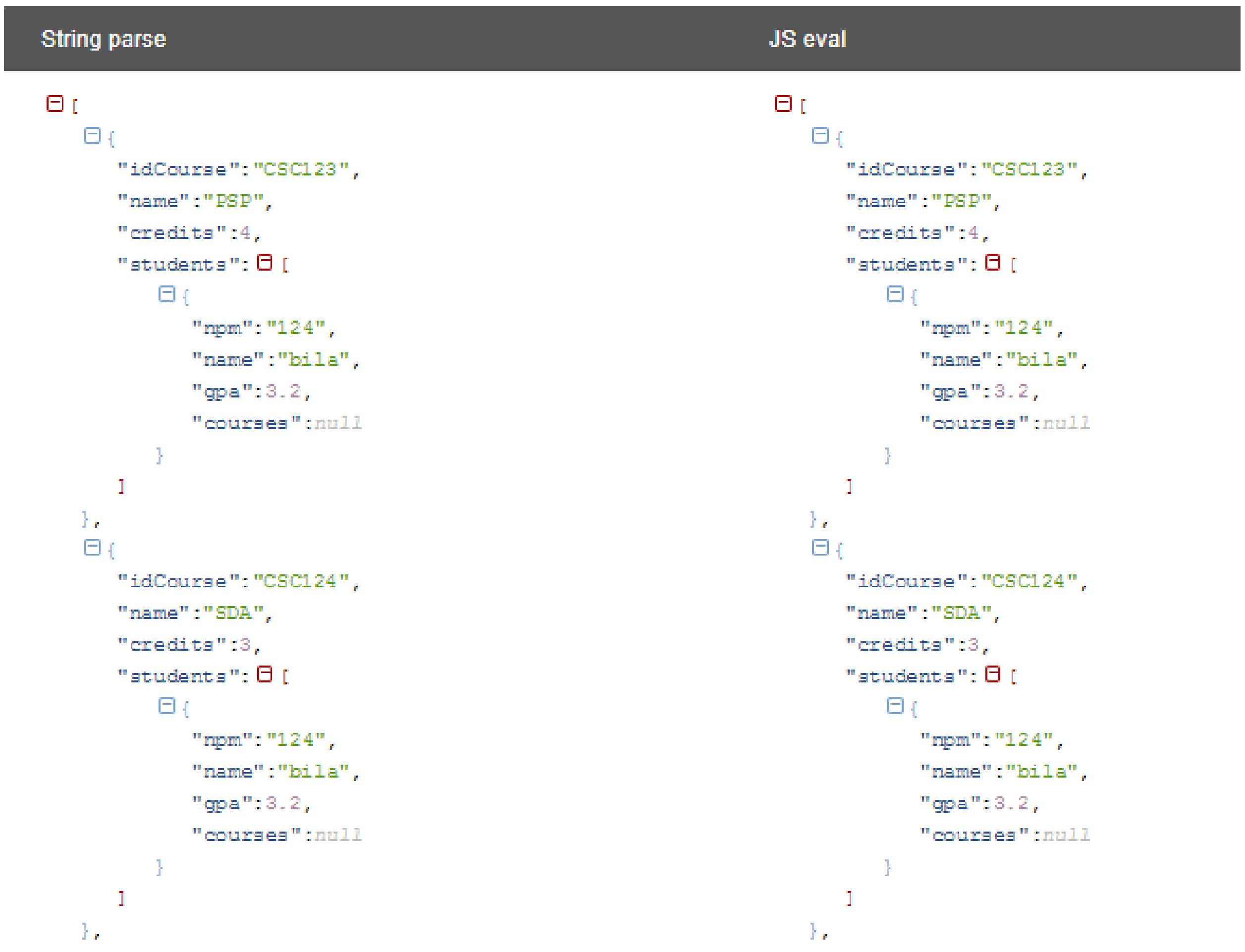
```
@RequestMapping("/course/viewall")
public List<CourseModel> viewall()
{
    List<CourseModel> courses = courseService.selectAllCourses ();
    return courses;
}
```

Hasil keluaran:

http://localhost:8080/rest/course/viewall



```
[{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"bila","gpa":3.2,"courses":null}]},{"idCourse":"CSC124","name":"SDA","credits":3,"students":
[{"npm":"124","name":"bila","gpa":3.2,"courses":null}]},{"idCourse":"CSC125","name":"DDP 1","credits":4,"students":[]},
{"idCourse":"CSC126","name":"MPKT","credits":6,"students":[{"npm":"123","name":"Nabilah","gpa":3.9,"courses":null}]}]
```

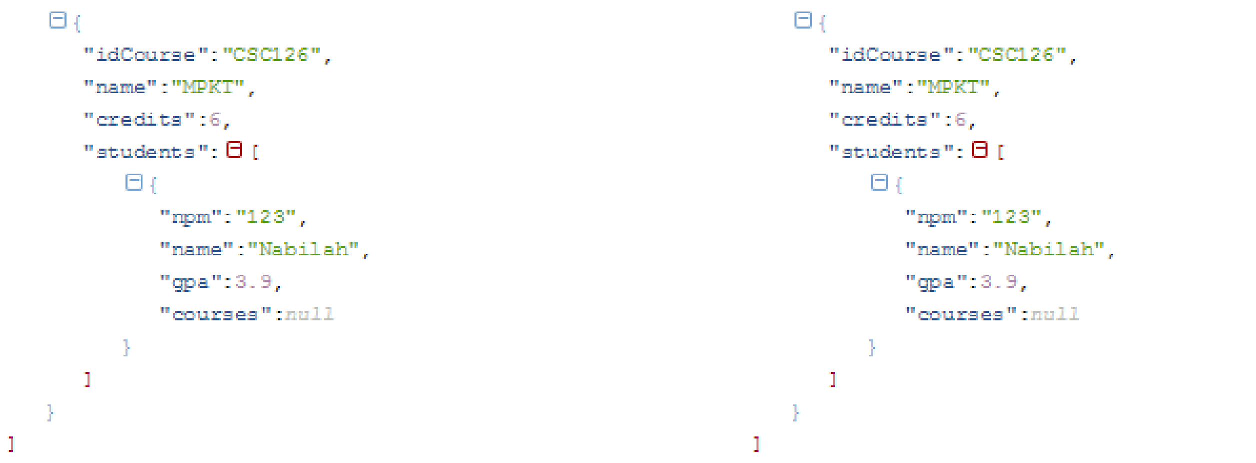


String parse

```
⊟ [
   ⊟ {
      "idCourse":"CSC123",
      "name":"PSP",
      "credits":4,
      "students": ⊟ [
         ⊟ {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
         }
      ]
   },
   ⊟ {
      "idCourse":"CSC124",
      "name":"SDA",
      "credits":3,
      "students": ⊟ [
         ⊟ {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
         }
      ]
   },
```

JS eval

```
⊟ [
   ⊟ {
      "idCourse":"CSC123",
      "name":"PSP",
      "credits":4,
      "students": ⊟ [
         ⊟ {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
         }
      ]
   },
   ⊟ {
      "idCourse":"CSC124",
      "name":"SDA",
      "credits":3,
      "students": ⊟ [
         ⊟ {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
         }
      ]
   },
```



```
   ⊟ {
      "idCourse":"CSC126",
      "name":"MPKT",
      "credits":6,
      "students": ⊟ [
         ⊟ {
            "npm":"123",
            "name":"Nabilah",
            "gpa":3.9,
            "courses":null
         }
      ]
   }
]
```

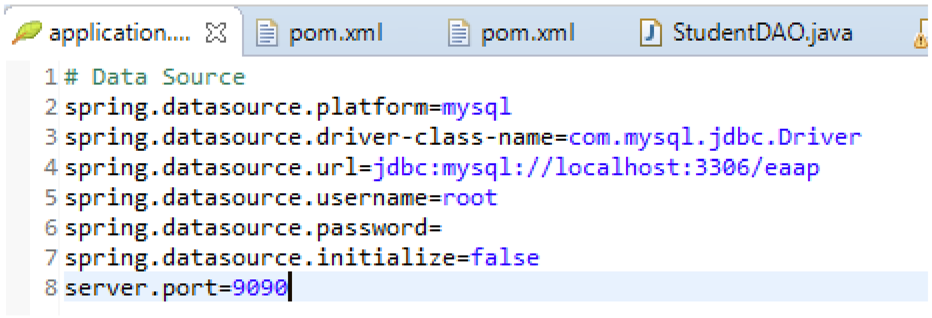
```
   ⊟ {
      "idCourse":"CSC126",
      "name":"MPKT",
      "credits":6,
      "students": ⊟ [
         ⊟ {
            "npm":"123",
            "name":"Nabilah",
            "gpa":3.9,
            "courses":null
         }
      ]
   }
]
```

Membuat Service Consumer

Nabilah Zhafira Viderisa
1506721806
ADPAP – C

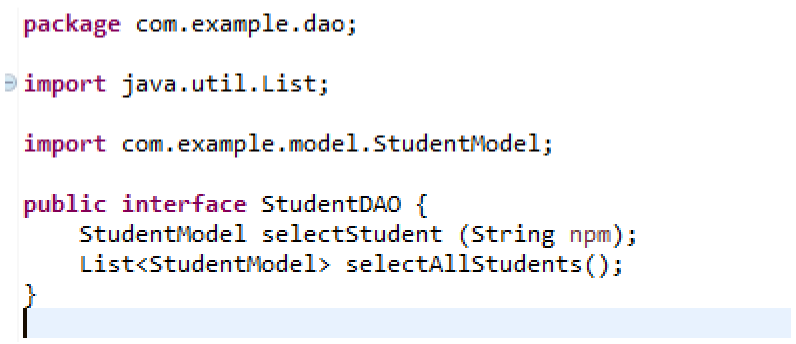
1. Import berkas-berkas yang ada di tutorial 6 untuk Service Consumer ini. Buatlah project 'Tutorial07COnsumerdan silakan import satu folder atau import file satu per satu dari Tutorial06. Jika Anda langsung import satu project tutorial 6 jangan lupa untuk menghapus folder .git, namun jangan hapus .gitignore, dalam folder tersebut setelah dipindahkan menjadi folder Tutorial07Consumer karena Anda tidak akan push dari direktori Tutorial07Consumer.

2. Karena Service Producer dan Service Consumer harus dijalankan secara bersamaan, ubah berkas application.properties yang ada di folder resources dengan menambahkan baris



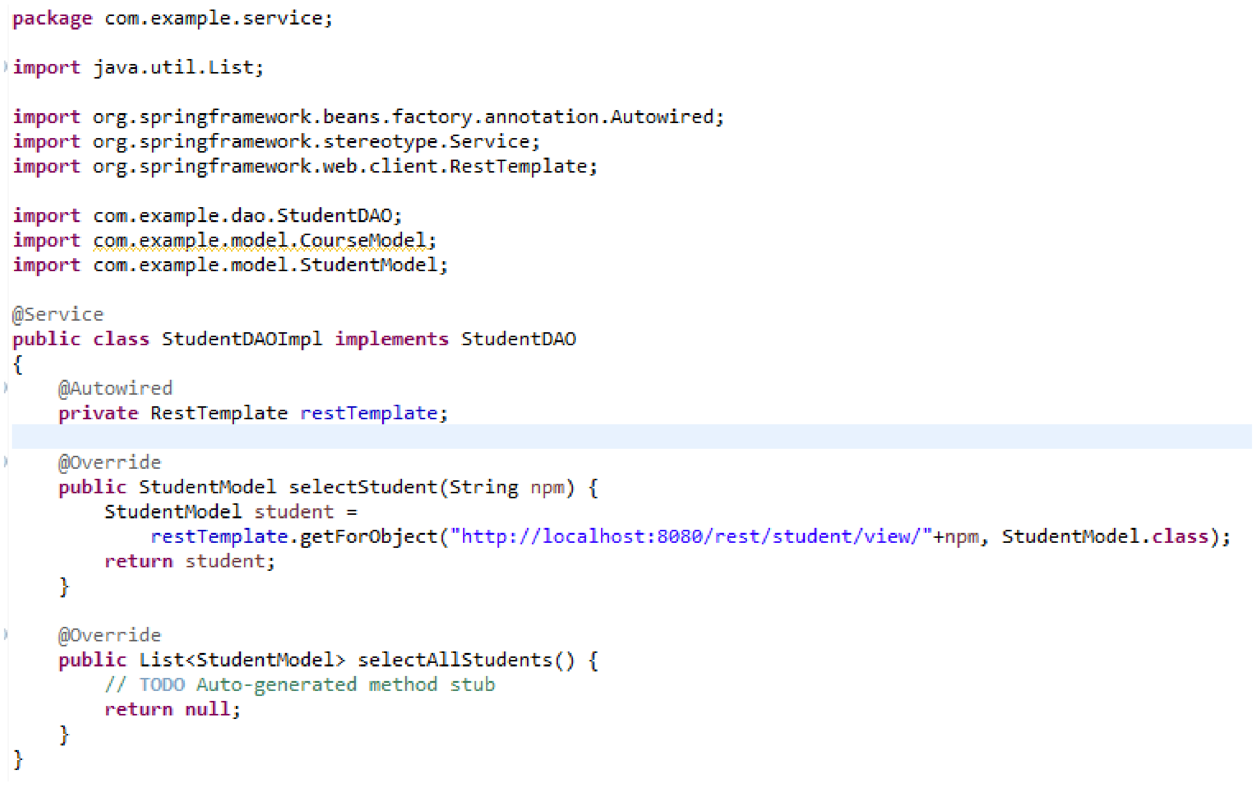
```
# Data Source
spring.datasource.platform=mysql
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/eaap
spring.datasource.username=root
spring.datasource.password=
spring.datasource.initialize=false
server.port=9090
```

4. Isilah StudentDAO dengan kode sebagai berikut:



```
package com.example.dao;

import java.util.List;

import com.example.model.StudentModel;

public interface StudentDAO {
    StudentModel selectStudent (String npm);
    List<StudentModel> selectAllStudents();
}
```

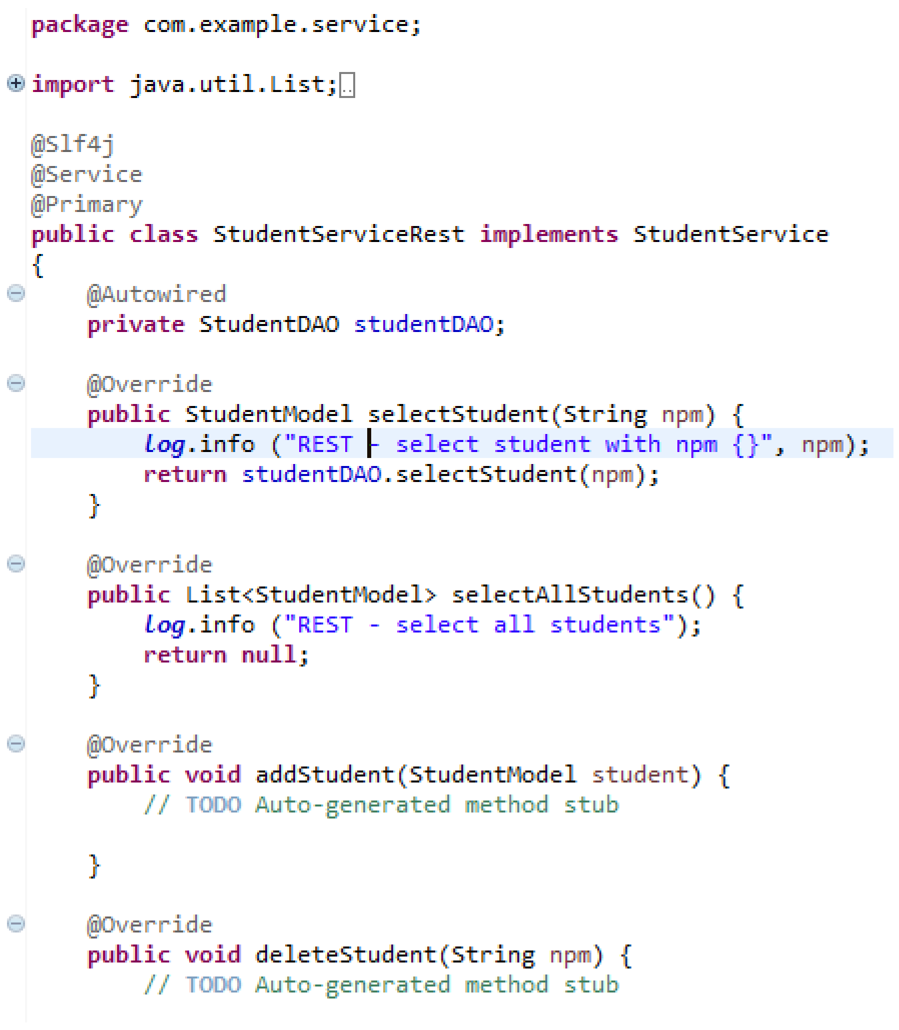
5. Selanjutnya kita akan membuat implementasi kelas StudentDAO tersebut dengan nama kelas StudentDAOImpl.java. Isinya adalah sebagai berikut



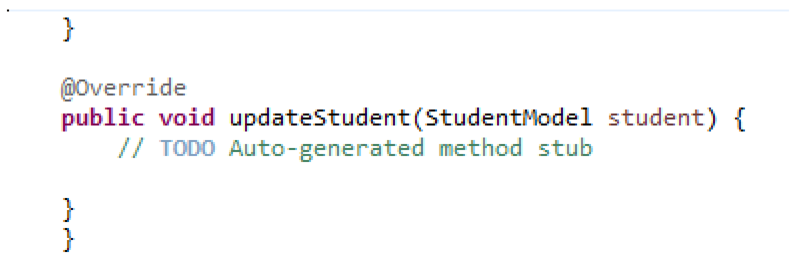
```
package com.example.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

import com.example.dao.StudentDAO;
import com.example.model.CourseModel;
import com.example.model.StudentModel;

@Service
public class StudentDAOImpl implements StudentDAO
{
    @Autowired
    private RestTemplate restTemplate;

    @Override
    public StudentModel selectStudent(String npm) {
        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
        return null;
    }
}
```

6. Selanjutnya, kita ingin mengubah agar StudentService mengambil data dari web service bukan dari database. Kita tidak perlu menghapus class StudentServiceDatabase. Karena scalable system, kita cukup menambahkan class baru yaitu StudentServiceRest yang mengimplement StudentService di package service.

7. Isinya adalah sebagai berikut:



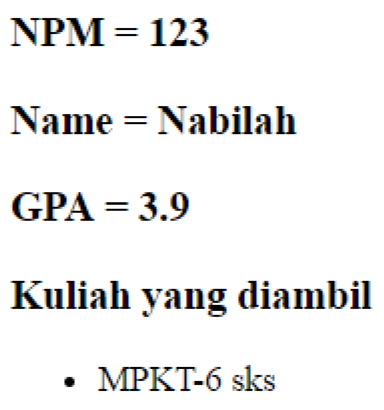
```java
package com.example.service;

import java.util.List;

@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 students");
        return null;
    }

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

    }

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



```java
    }

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

    }
}
```

8. Jalankan kedua project Spring Boot untuk Service Producer dan Service Consumer tersebut

9. Pastikan service producer sudah berjalan dengan menjalankan localhost:8080/rest/student/view/123.
10. Untuk menguji service consumer buka localhost:9090/student/view/123



NPM = 123

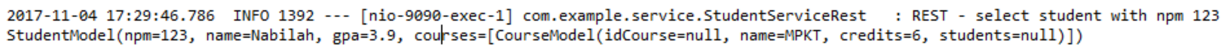
Name = Nabilah

GPA = 3.9

Kuliah yang diambil

- MPKT-6 sks

Di console:

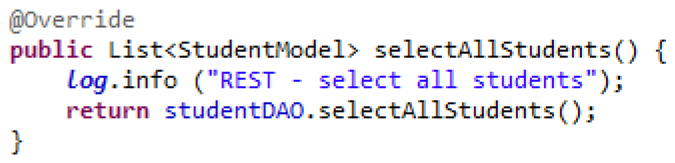


```
2017-11-04 17:29:46.786  INFO 1392 --- [nio-9090-exec-1] com.example.service.StudentServiceRest   : REST - select student with npm 123
StudentModel(npm=123, name=Nabilah, gpa=3.9, courses=[CourseModel(idCourse=null, name=MPKT, credits=6, students=null)])
```

LATIHAN

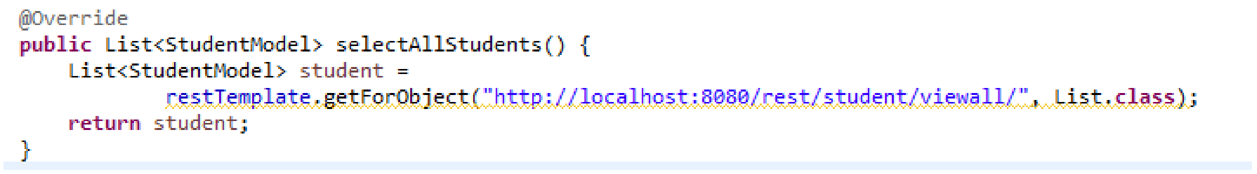
1. **Latihan 3**: Implementasikan service consumer untuk view all Students dengan melengkapi method selectAllStudents yang ada di kelas StudentServiceRest.

➔ Untuk mengimplementasikan view all students pada service customer hal yang pertama dilakukan adalah menambahkan return pada method selectAllStudents() pada StudentServiceRest menjadi seperti berikut:



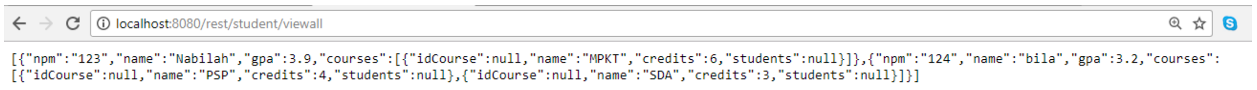
```
@Override
public List<StudentModel> selectAllStudents() {
    log.info ("REST - select all students");
    return studentDAO.selectAllStudents();
}
```

Selanjutnya di StudentDAOImpl.java saya melengkapi method selectAllStudents() dengan menambahkan getForObject



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

Saat dibuka localhost:8080/rest/student/viewall

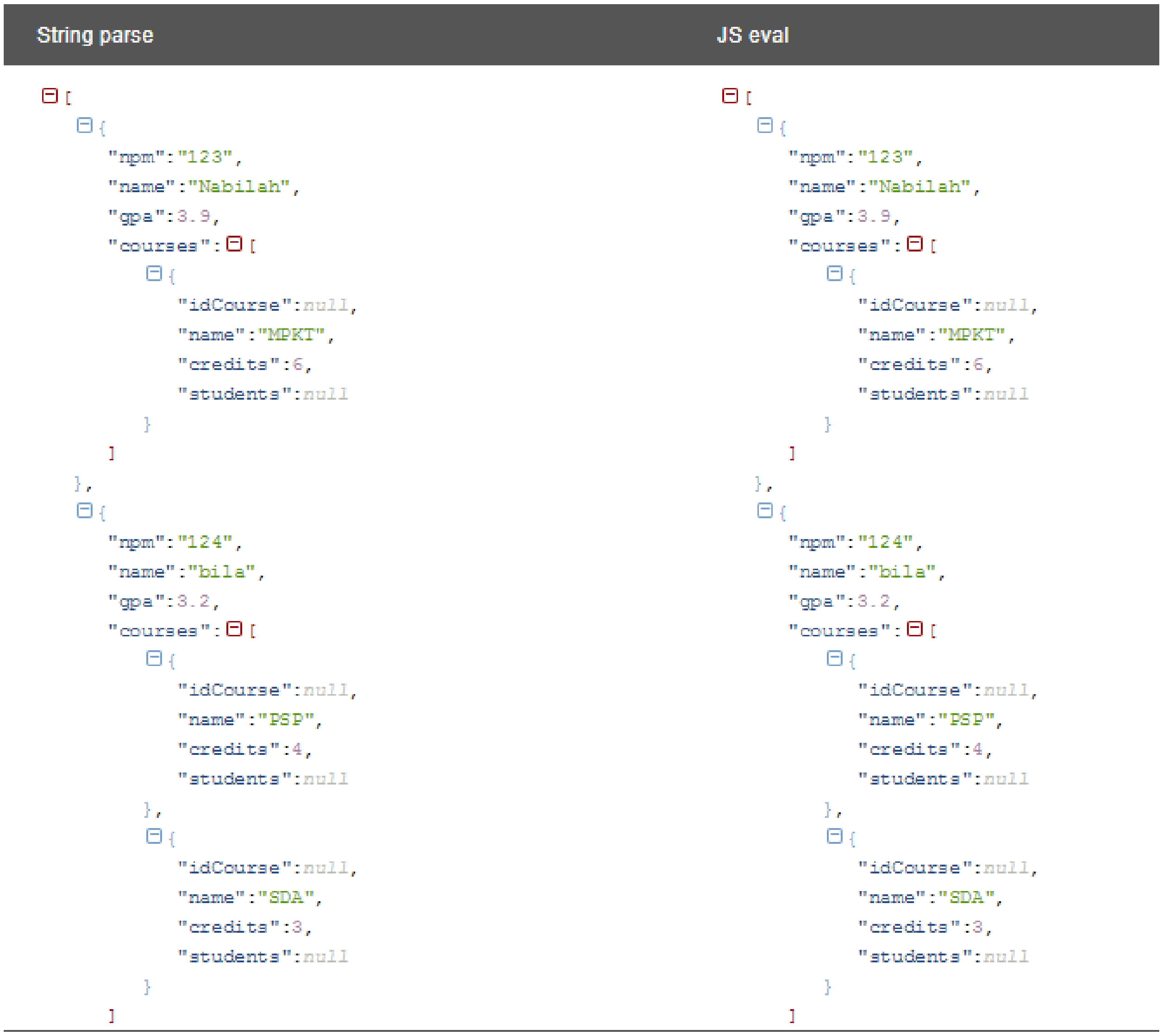


localhost:8080/rest/student/viewall

```
[{"npm":"123","name":"Nabilah","gpa":3.9,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]},{"npm":"124","name":"bila","gpa":3.2,"courses":
[{"idCourse":null,"name":"PSP","credits":4,"students":null},{"idCourse":null,"name":"SDA","credits":3,"students":null}]}]
```

Nabilah Zhafira Viderisa
1506721806
ADPAP – C

Tutorial 7

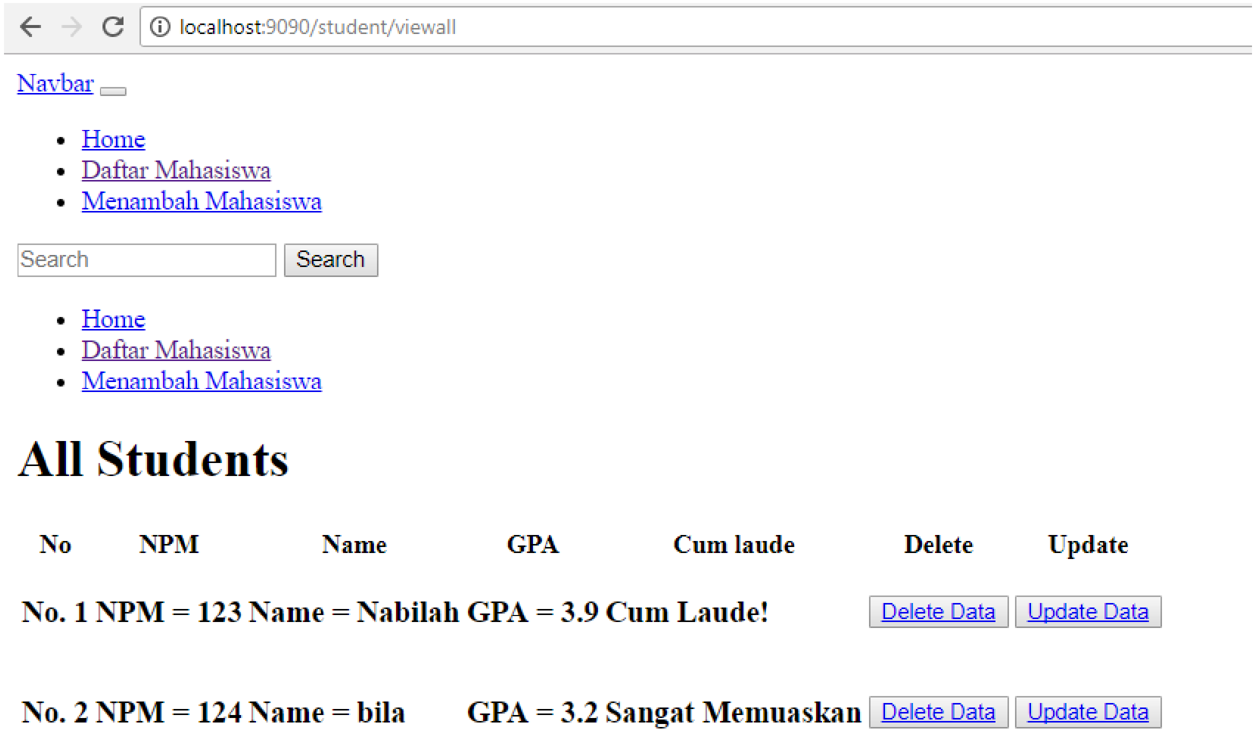


String parse

```
[
  {
    "npm":"123",
    "name":"Nabilah",
    "gpa":3.9,
    "courses":[
      {
        "idCourse":null,
        "name":"MPKT",
        "credits":6,
        "students":null
      }
    ]
  },
  {
    "npm":"124",
    "name":"bila",
    "gpa":3.2,
    "courses":[
      {
        "idCourse":null,
        "name":"PSP",
        "credits":4,
        "students":null
      },
      {
        "idCourse":null,
        "name":"SDA",
        "credits":3,
        "students":null
      }
    ]
```

JS eval

```
[
  {
    "npm":"123",
    "name":"Nabilah",
    "gpa":3.9,
    "courses":[
      {
        "idCourse":null,
        "name":"MPKT",
        "credits":6,
        "students":null
      }
    ]
  },
  {
    "npm":"124",
    "name":"bila",
    "gpa":3.2,
    "courses":[
      {
        "idCourse":null,
        "name":"PSP",
        "credits":4,
        "students":null
      },
      {
        "idCourse":null,
        "name":"SDA",
        "credits":3,
        "students":null
      }
    ]
```



localhost:9090/student/viewall

Navbar

- Home
- Daftar Mahasiswa
- Menambah Mahasiswa

Search | Search

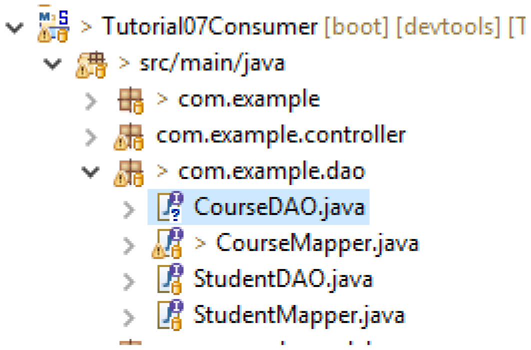
- Home
- Daftar Mahasiswa
- Menambah Mahasiswa

All Students

No	NPM	Name	GPA	Cum laude	Delete	Update
No. 1	NPM = 123	Name = Nabilah	GPA = 3.9	Cum Laude!	Delete Data	Update Data
No. 2	NPM = 124	Name = bila	GPA = 3.2	Sangat Memuaskan	Delete Data	Update Data

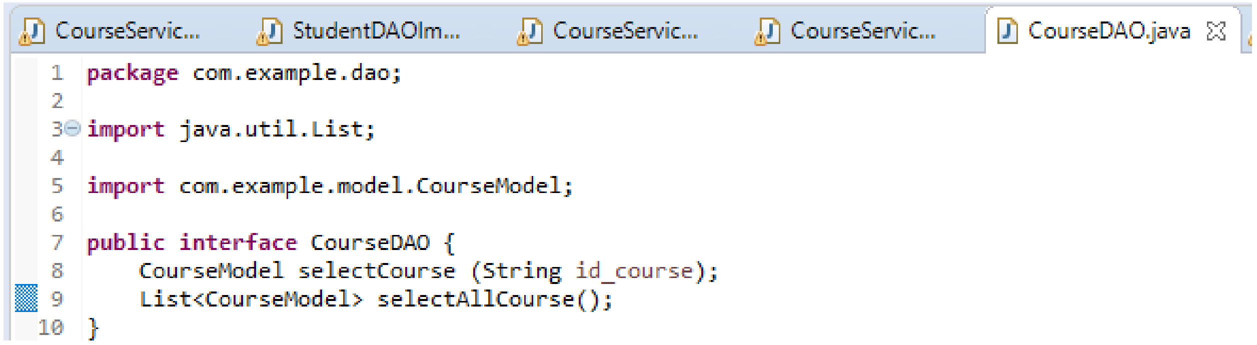
2. **Latihan 4:** Implementasikan service consumer untuk class CourseModel dengan membuat class-class DAO dan service baru.

➔ Untuk meng-implementasikan service consumer untuk class CourseModel, hal yang pertama saya lakukan adalah membuat CourseDAO.java



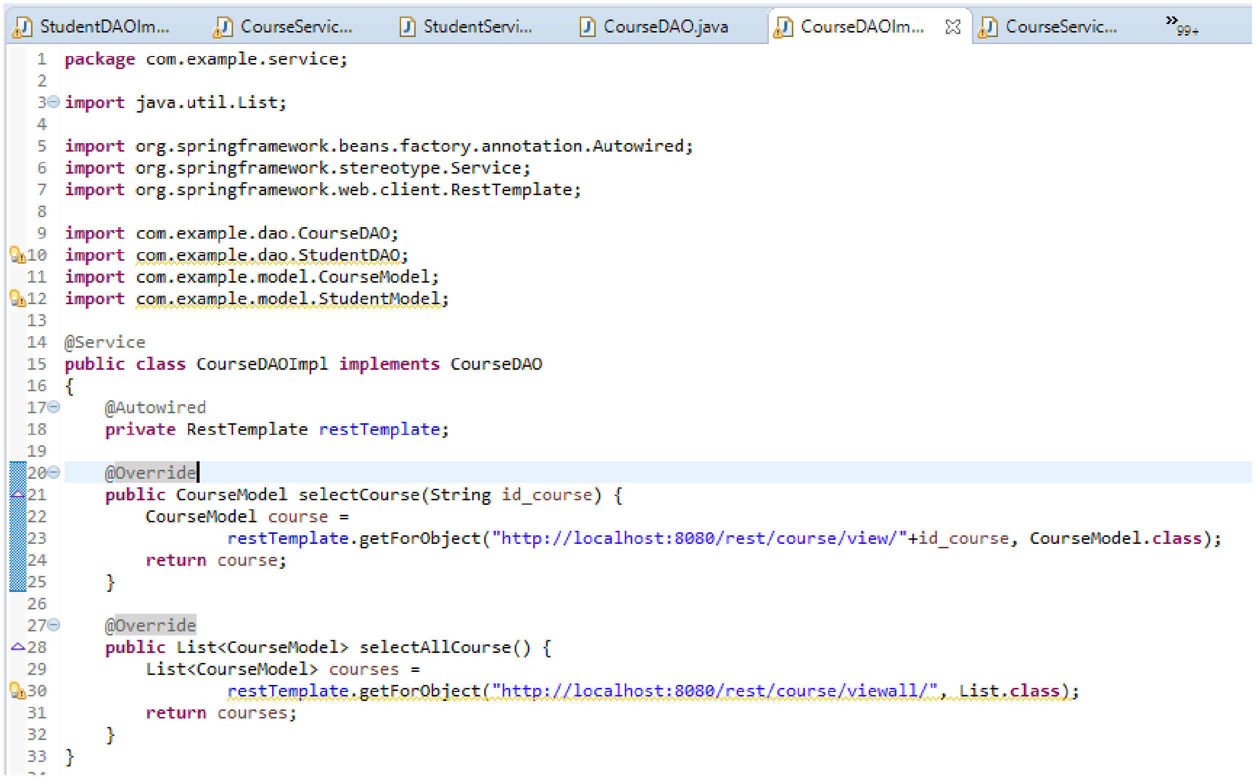
```
Tutorial07Consumer [boot] [devtools] [T
  src/main/java
    com.example
    com.example.controller
    com.example.dao
      CourseDAO.java
      CourseMapper.java
      StudentDAO.java
      StudentMapper.java
```

Yang berisi:



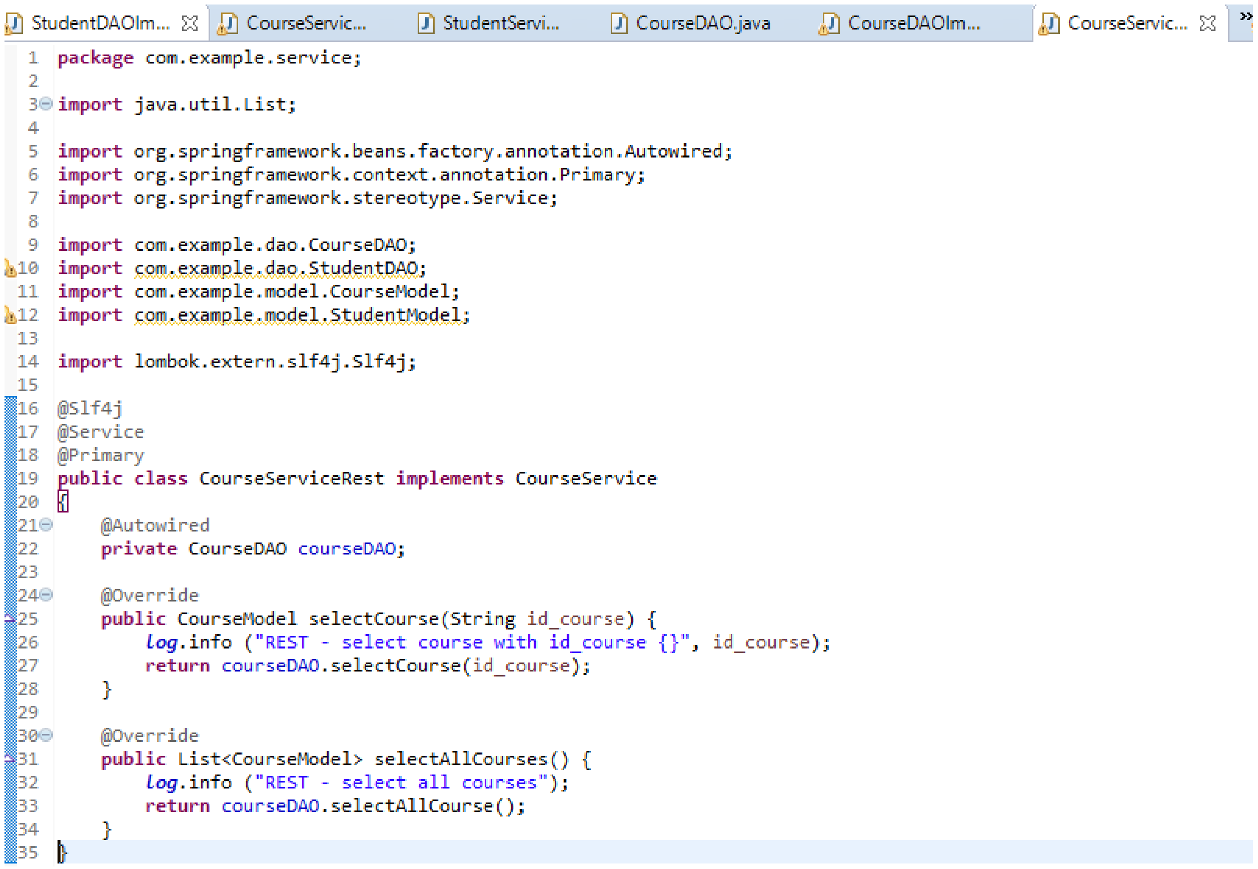
```
package com.example.dao;

import java.util.List;

import com.example.model.CourseModel;

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

Selanjutnya saya membuat CourseDAOImpl.java yang berisi



```java
package com.example.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.web.client.RestTemplate;

import com.example.dao.CourseDAO;
import com.example.dao.StudentDAO;
import com.example.model.CourseModel;
import com.example.model.StudentModel;

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

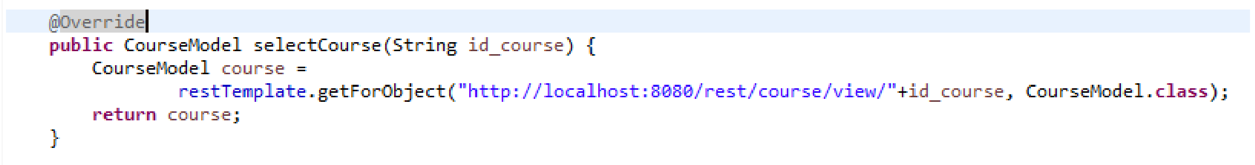
Setelah itu, saya membuat CourseServiceRest.java yang berisi:



StudentDAOIm... CourseServic... StudentServi... CourseDAO.java CourseDAOIm... CourseServic...

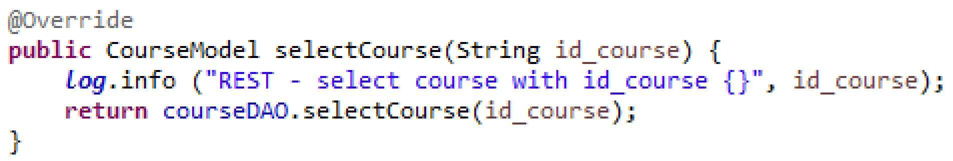
```java
package com.example.service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Primary;
import org.springframework.stereotype.Service;

import com.example.dao.CourseDAO;
import com.example.dao.StudentDAO;
import com.example.model.CourseModel;
import com.example.model.StudentModel;

import lombok.extern.slf4j.Slf4j;

@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 courses");
        return courseDAO.selectAllCourse();
    }
}
```

➔ Untuk meng-view berdasarkan id_course, pada CourseDAOImpl.java terdapat method:



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

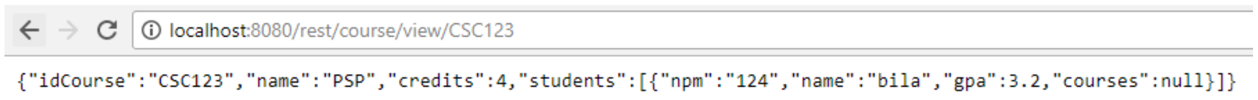
Dan pada CourseServiceRest.java terdapat:



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

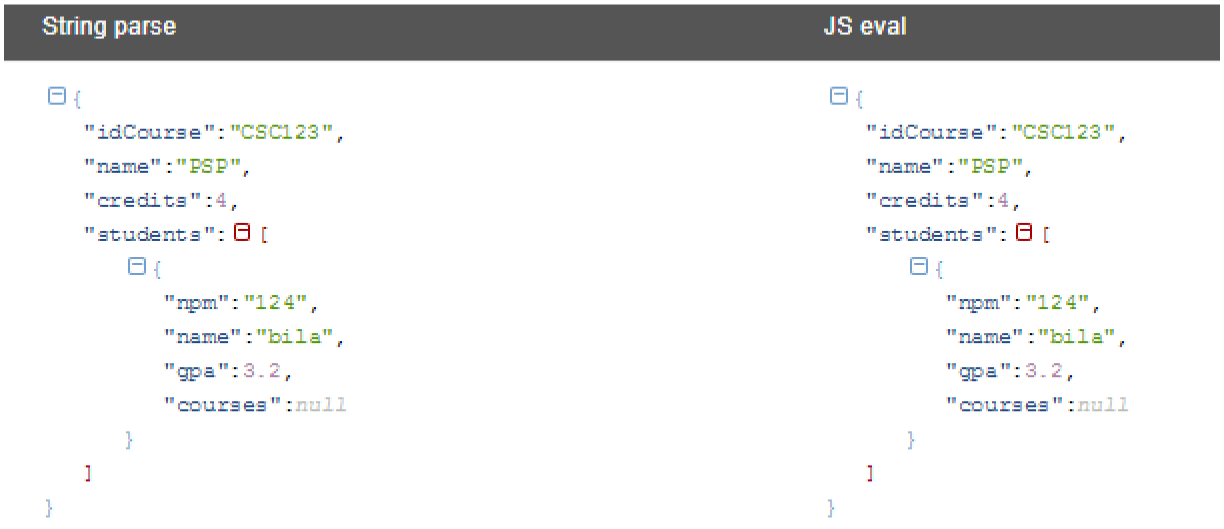
Hasil keluaran:

http://localhost:8080/rest/course/view/CSC123



localhost:8080/rest/course/view/CSC123

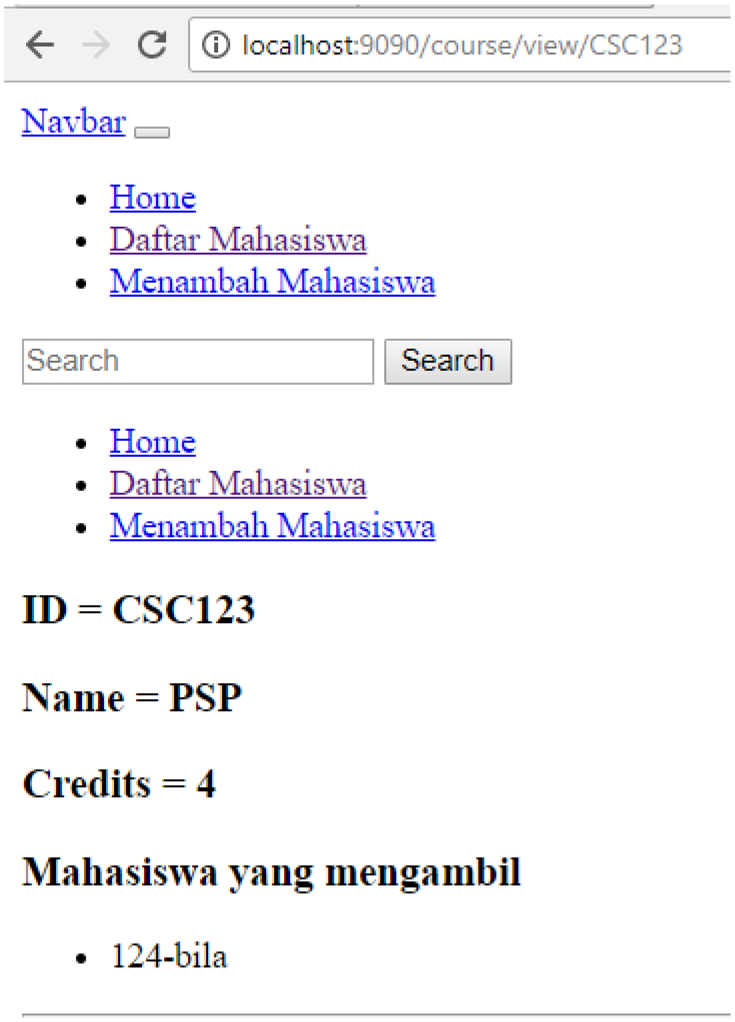
```
{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"bila","gpa":3.2,"courses":null}]}
```



String parse | JS eval

```
{
   "idCourse":"CSC123",
   "name":"PSP",
   "credits":4,
   "students":[
      {
         "npm":"124",
         "name":"bila",
         "gpa":3.2,
         "courses":null
      }
   ]
}
```

http://localhost:9090/course/view/CSC123



localhost:9090/course/view/CSC123

Navbar

- Home
- Daftar Mahasiswa
- Menambah Mahasiswa

Search | Search

- Home
- Daftar Mahasiswa
- Menambah Mahasiswa

ID = CSC123

Name = PSP

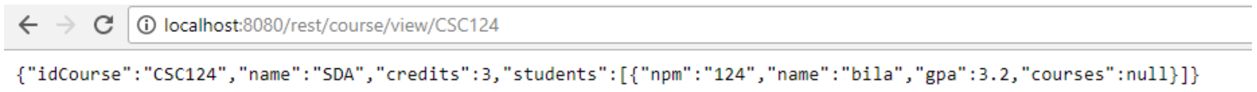
Credits = 4

Mahasiswa yang mengambil

- 124-bila

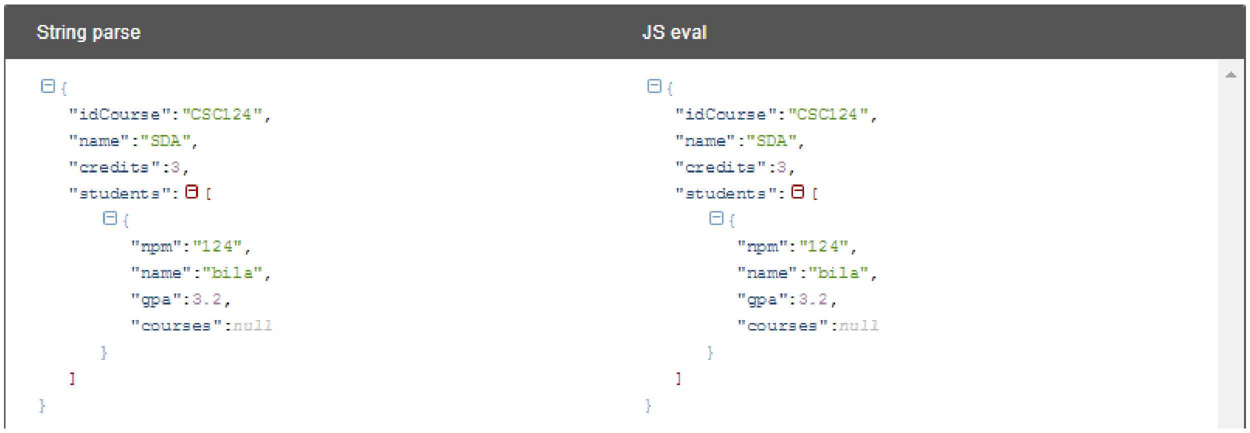
Nabilah Zhafira Viderisa Tutorial 7
1506721806
ADPAP – C

http://localhost:8080/rest/course/view/CSC124



localhost:8080/rest/course/view/CSC124

{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"bila","gpa":3.2,"courses":null}]}

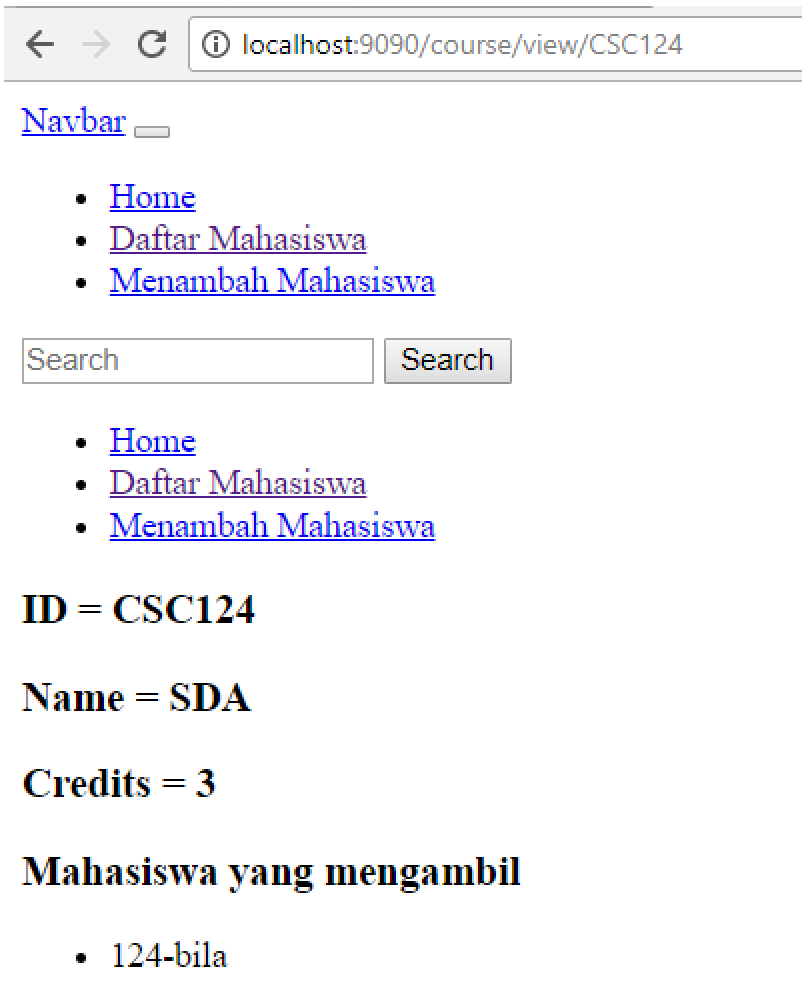


String parse JS eval

```
{
  "idCourse":"CSC124",
  "name":"SDA",
  "credits":3,
  "students":[
    {
      "npm":"124",
      "name":"bila",
      "gpa":3.2,
      "courses":null
    }
  ]
}
```

```
{
  "idCourse":"CSC124",
  "name":"SDA",
  "credits":3,
  "students":[
    {
      "npm":"124",
      "name":"bila",
      "gpa":3.2,
      "courses":null
    }
  ]
}
```

http://localhost:9090/course/view/CSC124


localhost:9090/course/view/CSC124

Navbar

- Home
- Daftar Mahasiswa
- Menambah Mahasiswa

Search Search

- Home
- Daftar Mahasiswa
- Menambah Mahasiswa

ID = CSC124

Name = SDA

Credits = 3

Mahasiswa yang mengambil

- 124-bila

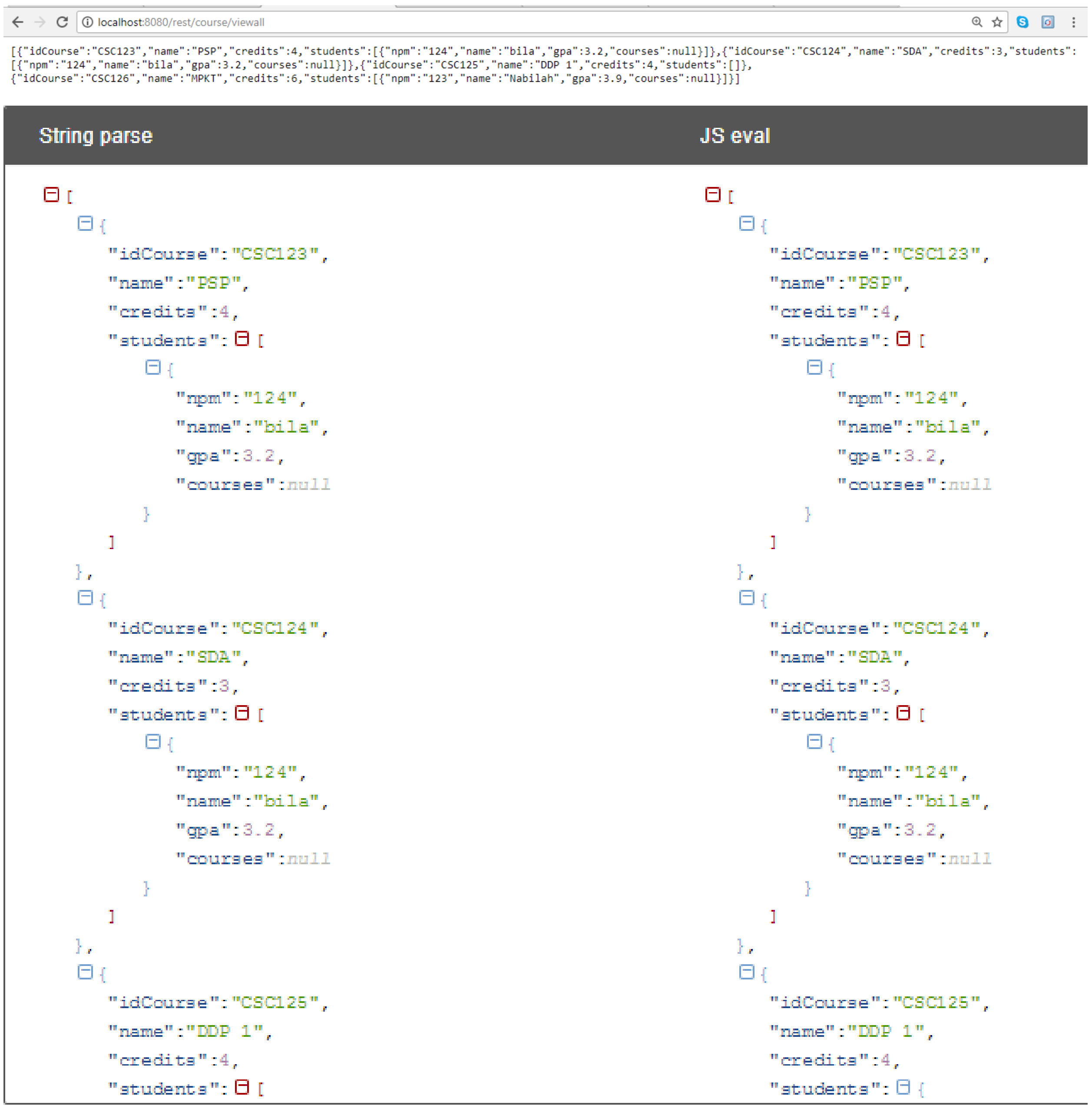
➔ Untuk meng-viewall courses, pada CourseDAOImpl.java terdapat method:

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

Dan pada CourseServiceRest.java terdapat:

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

http://localhost:8080/rest/course/viewall



localhost:8080/rest/course/viewall

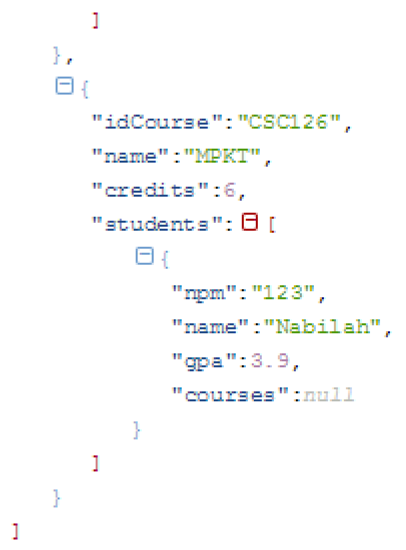
```
[{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"bila","gpa":3.2,"courses":null}]},{"idCourse":"CSC124","name":"SDA","credits":3,"students":
[{"npm":"124","name":"bila","gpa":3.2,"courses":null}]},{"idCourse":"CSC125","name":"DDP 1","credits":4,"students":[]},
{"idCourse":"CSC126","name":"MPKT","credits":6,"students":[{"npm":"123","name":"Nabilah","gpa":3.9,"courses":null}]}]
```

String parse

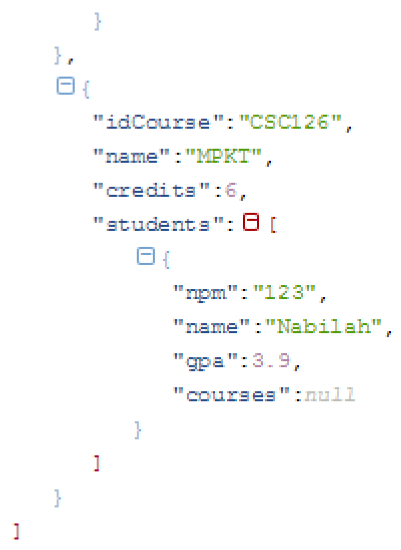
```
[
   {
      "idCourse":"CSC123",
      "name":"PSP",
      "credits":4,
      "students": [
         {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
         }
      ]
   },
   {
      "idCourse":"CSC124",
      "name":"SDA",
      "credits":3,
      "students": [
         {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
         }
      ]
   },
   {
      "idCourse":"CSC125",
      "name":"DDP 1",
      "credits":4,
      "students": [
```

JS eval

```
[
   {
      "idCourse":"CSC123",
      "name":"PSP",
      "credits":4,
      "students": [
         {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
         }
      ]
   },
   {
      "idCourse":"CSC124",
      "name":"SDA",
      "credits":3,
      "students": [
         {
            "npm":"124",
            "name":"bila",
            "gpa":3.2,
            "courses":null
         }
      ]
   },
   {
      "idCourse":"CSC125",
      "name":"DDP 1",
      "credits":4,
      "students": {
```

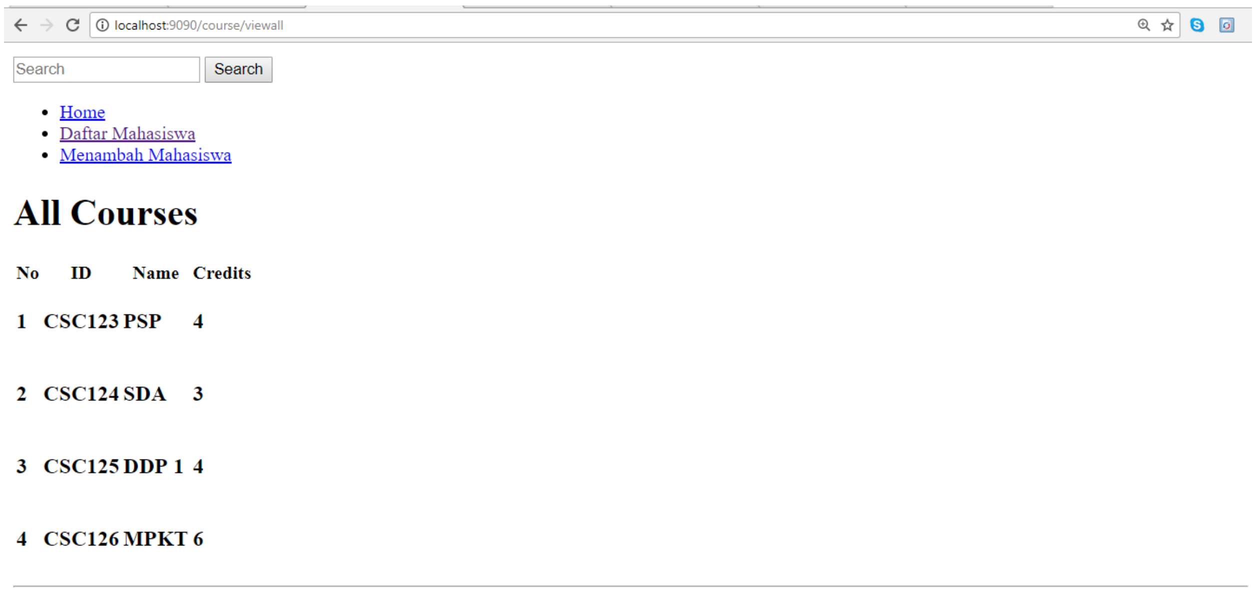


```
        ]
    },
    {
        "idCourse":"CSC126",
        "name":"MPKT",
        "credits":6,
        "students": [
            {
                "npm":"123",
                "name":"Nabilah",
                "gpa":3.9,
                "courses":null
            }
        ]
    }
]
```



```
        }
    },
    {
        "idCourse":"CSC126",
        "name":"MPKT",
        "credits":6,
        "students": [
            {
                "npm":"123",
                "name":"Nabilah",
                "gpa":3.9,
                "courses":null
            }
        ]
    }
]
```

http://localhost:9090/course/viewall



localhost:9090/course/viewall

Search Search

- Home
- Daftar Mahasiswa
- Menambah Mahasiswa

All Courses

No	ID	Name	Credits
1	CSC123	PSP	4
2	CSC124	SDA	3
3	CSC125	DDP 1	4
4	CSC126	MPKT	6

LESSON LEARNED:

Pada tutorial kali ini, saya belajar mengenai web service dalam framework springboot. Dimana pada tutorial kali ini saya belajar untuk membuat web service dengan menggunakan 2 layer yaitu consumer dan producer. Tujuan dari pemisahan ini adalah untuk memisahkan antara front-end (consumer) dan back-end (producer). Selain itu saya juga jadi mengetahui istilah-istilah dan method-method baru seperti Rest Template dan method getForObject.