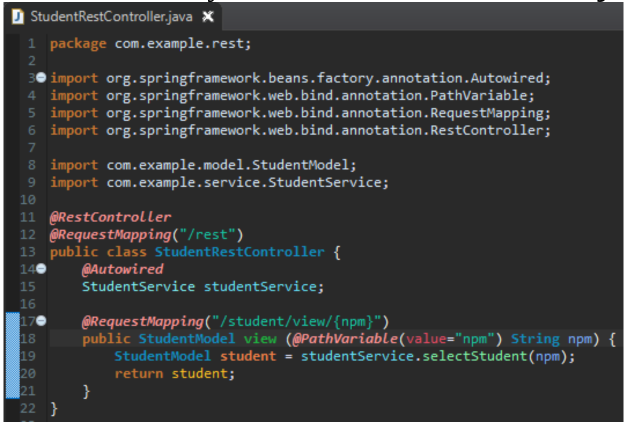
Fajar Bardan Salam | 1506689401
APAP-A

Tutorial 7

Membuat Web Service Menggunakan Spring Boot Framework

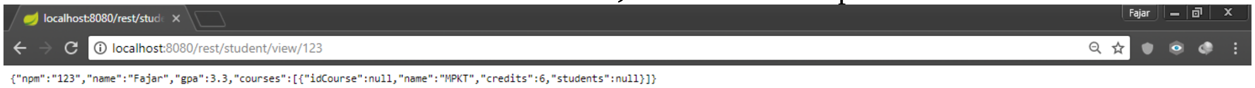
Membuat Service *Producer*

1. Import berkas-berkas yang ada di tutorial 5 untuk Service producer ini.
2. Buat *package* baru com.example.rest, atau sesuaikan dengan *package* Anda masing-masing
3. Buat class baru yaitu **StudentRestController.java pada package com.example.rest**



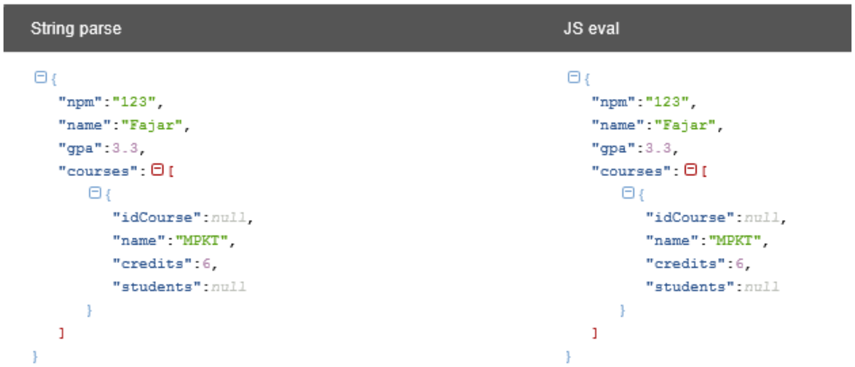
```java
package com.example.rest;

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;
    }
}
```

Jalankan **localhost:8080/rest/student/view/123,** akan keluar tampilan :



```
{"npm":"123","name":"Fajar","gpa":3.3,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]}
```

Gunakan JSON View untuk merapihkan tampilan



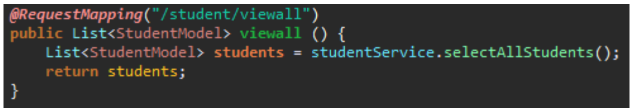
```
{
  "npm":"123",
  "name":"Fajar",
  "gpa":3.3,
  "courses":[
    {
      "idCourse":null,
      "name":"MPKT",
      "credits":6,
      "students":null
    }
  ]
}
```

```
{
  "npm":"123",
  "name":"Fajar",
  "gpa":3.3,
  "courses":[
    {
      "idCourse":null,
      "name":"MPKT",
      "credits":6,
      "students":null
    }
  ]
}
```

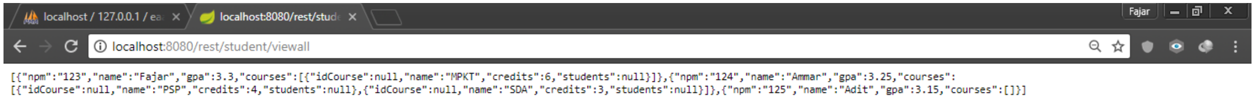
Fajar Bardan Salam / 1506689401
APAP-A

Latihan

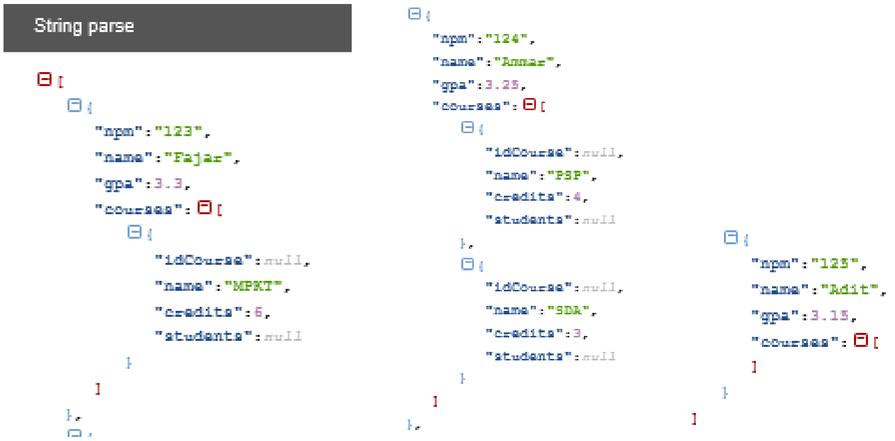
1. **Latihan 1:** Buatlah *service* untuk mengembalikan seluruh student yang ada di basis data. *Service* ini mirip seperti *method* viewAll di Web Controller. *Service* tersebut di-*mapping* ke **“/rest/student/viewall”.**



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

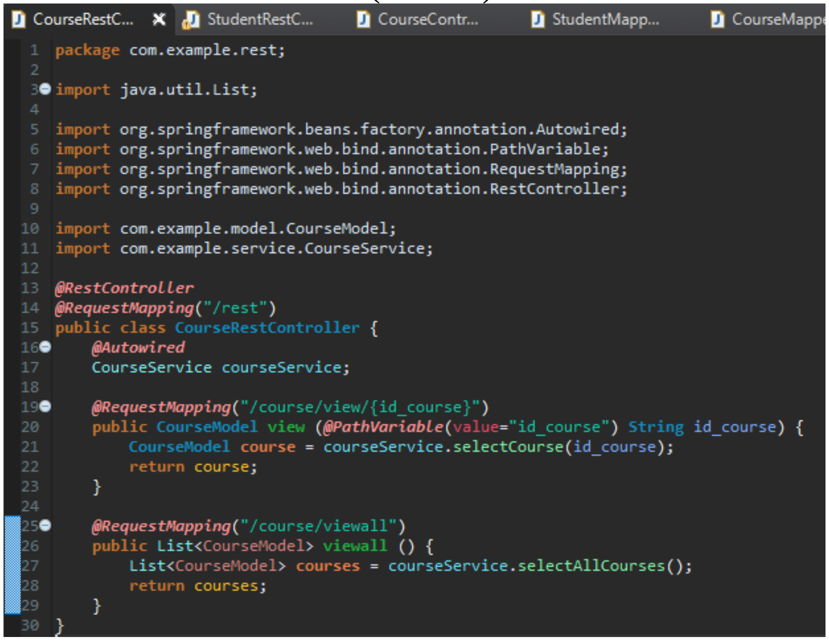

localhost / 127.0.0.1 / ea
localhost:8080/rest/stude
Fajar
localhost:8080/rest/student/viewall

```
[{"npm":"123","name":"Fajar","gpa":3.3,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]},{"npm":"124","name":"Ammar","gpa":3.25,"courses":
[{"idCourse":null,"name":"PSP","credits":4,"students":null},{"idCourse":null,"name":"SDA","credits":3,"students":null}]},{"npm":"125","name":"Adit","gpa":3.15,"courses":[]}]
```



```
⊟[
   ⊟{
      "npm":"123",
      "name":"Fajar",
      "gpa":3.3,
      "courses":⊟[
         ⊟{
            "idCourse":null,
            "name":"MPKT",
            "credits":6,
            "students":null
         }
      ]
   },
   ⊟{
      "npm":"124",
      "name":"Ammar",
      "gpa":3.25,
      "courses":⊟[
         ⊟{
            "idCourse":null,
            "name":"PSP",
            "credits":4,
            "students":null
         },
         ⊟{
            "idCourse":null,
            "name":"SDA",
            "credits":3,
            "students":null
         }
      ]
   },
   ⊟{
      "npm":"125",
      "name":"Adit",
      "gpa":3.15,
      "courses":⊟[
      ]
   }
]
```

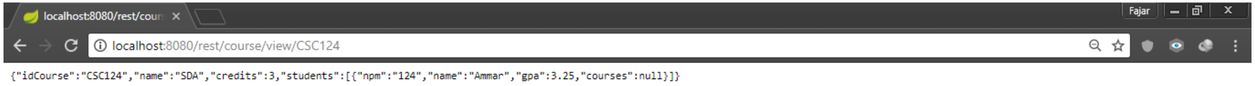
2. **Latihan 2:** Buatlah *service* untuk class Course. Buatlah *controller* baru yang terdapat *service* untuk melihat suatu course dengan masukan ID Course (view by ID) dan *service* untuk melihat semua course (view all)



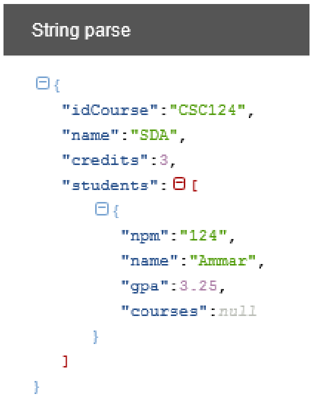
```
package com.example.rest;

import java.util.List;

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.CourseModel;
import com.example.service.CourseService;

@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;
    }

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

Fajar Bardan Salam | 1506689401
APAP-A

“/rest/course/view/{id_course}”


localhost:8080/rest/cour ×
localhost:8080/rest/course/view/CSC124
Fajar

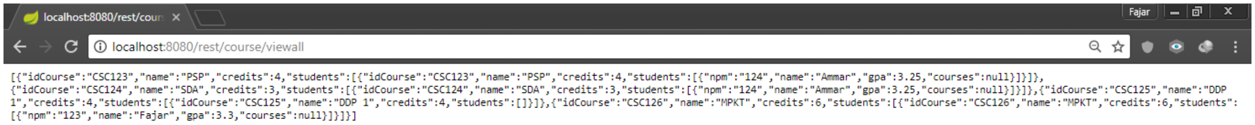
{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"Ammar","gpa":3.25,"courses":null}]}



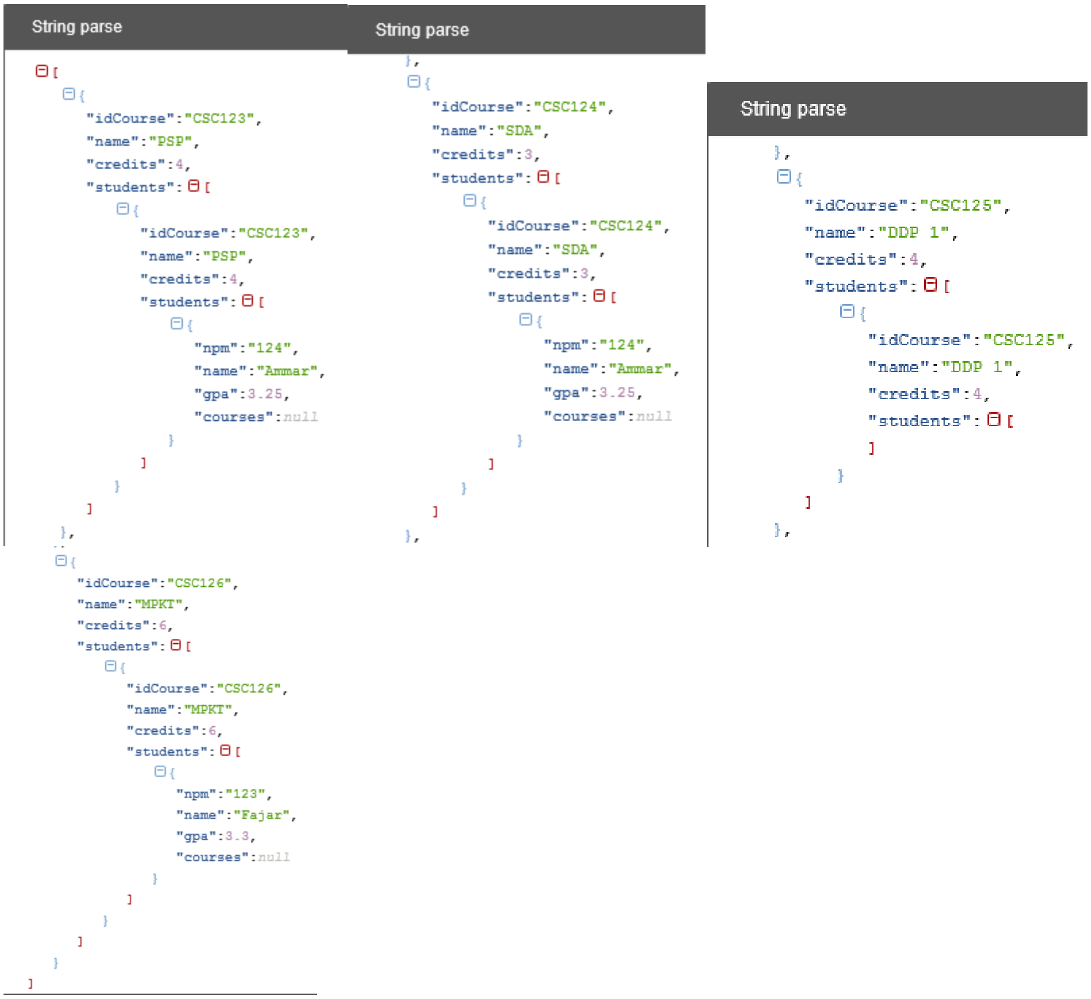
String parse

```
⊟{
   "idCourse":"CSC124",
   "name":"SDA",
   "credits":3,
   "students":⊟[
      ⊟{
         "npm":"124",
         "name":"Ammar",
         "gpa":3.25,
         "courses":null
      }
   ]
}
```

“/rest/course/viewall”


localhost:8080/rest/cour ×
localhost:8080/rest/course/viewall
Fajar

[{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"idCourse":"CSC123","name":"PSP","credits":4,"students":[{"npm":"124","name":"Ammar","gpa":3.25,"courses":null}]}]},
{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"Ammar","gpa":3.25,"courses":null}]}]},{"idCourse":"CSC125","name":"DDP
1","credits":4,"students":[{"idCourse":"CSC125","name":"DDP 1","credits":4,"students":[]}]},{"idCourse":"CSC126","name":"MPKT","credits":6,"students":[{"idCourse":"CSC126","name":"MPKT","credits":6,"students":
[{"npm":"123","name":"Fajar","gpa":3.3,"courses":null}]}]}]



String parse

```
⊟[
   ⊟{
      "idCourse":"CSC123",
      "name":"PSP",
      "credits":4,
      "students":⊟[
         ⊟{
            "idCourse":"CSC123",
            "name":"PSP",
            "credits":4,
            "students":⊟[
               ⊟{
                  "npm":"124",
                  "name":"Ammar",
                  "gpa":3.25,
                  "courses":null
               }
            ]
         }
      ]
   },
```

String parse

```
   },
   ⊟{
      "idCourse":"CSC124",
      "name":"SDA",
      "credits":3,
      "students":⊟[
         ⊟{
            "idCourse":"CSC124",
            "name":"SDA",
            "credits":3,
            "students":⊟[
               ⊟{
                  "npm":"124",
                  "name":"Ammar",
                  "gpa":3.25,
                  "courses":null
               }
            ]
         }
      ]
   },
```

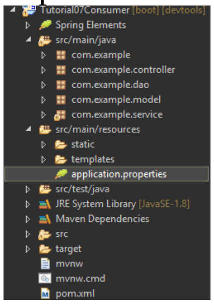
String parse

```
   },
   ⊟{
      "idCourse":"CSC125",
      "name":"DDP 1",
      "credits":4,
      "students":⊟[
         ⊟{
            "idCourse":"CSC125",
            "name":"DDP 1",
            "credits":4,
            "students":⊟[
            ]
         }
      ]
   },
```

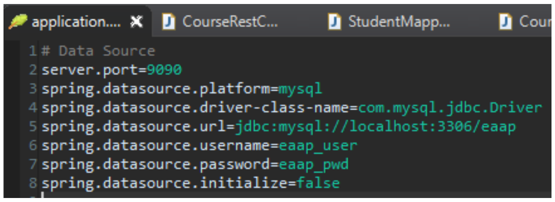
```
   ⊟{
      "idCourse":"CSC126",
      "name":"MPKT",
      "credits":6,
      "students":⊟[
         ⊟{
            "idCourse":"CSC126",
            "name":"MPKT",
            "credits":6,
            "students":⊟[
               ⊟{
                  "npm":"123",
                  "name":"Fajar",
                  "gpa":3.3,
                  "courses":null
               }
            ]
         }
      ]
   }
]
```

Membuat Service *Consumer*

1. Import berkas-berkas yang ada di tutorial 6 untuk Service Consumer ini.



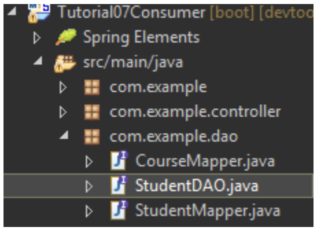
2. Karena *Service Producer* dan *Service Consumer* harus dijalankan secara bersamaan, ubah berkas **application.properties** yang ada di folder resources dengan menambahkan baris



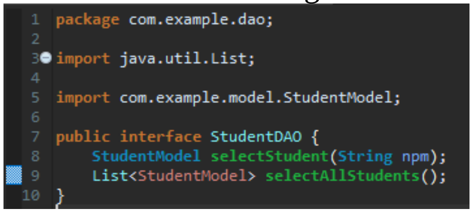
```
# Data Source
server.port=9090
spring.datasource.platform=mysql
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/eaap
spring.datasource.username=eaap_user
spring.datasource.password=eaap_pwd
spring.datasource.initialize=false
```

Nantinya Aplikasi *Service Producer* akan dijalankan di localhost:8080 sedangkan *Service Consumer* akan dijalankan di localhost:9090

3. Di DAO tambahkan ***interface*** dengan nama StudentDAO

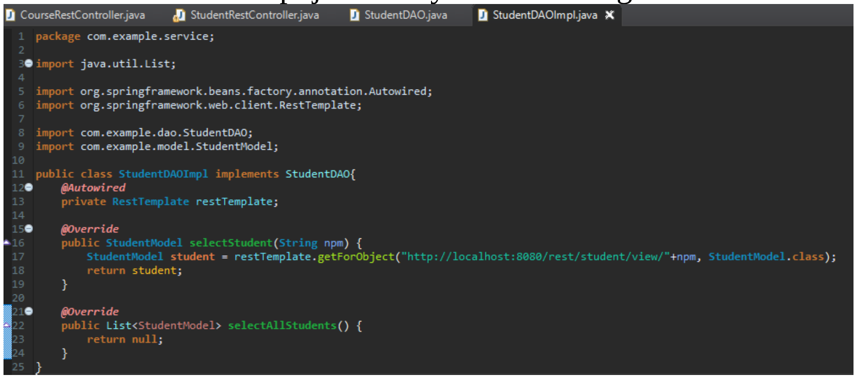


4. Isilah StudentDAO dengan kode sebagai berikut:



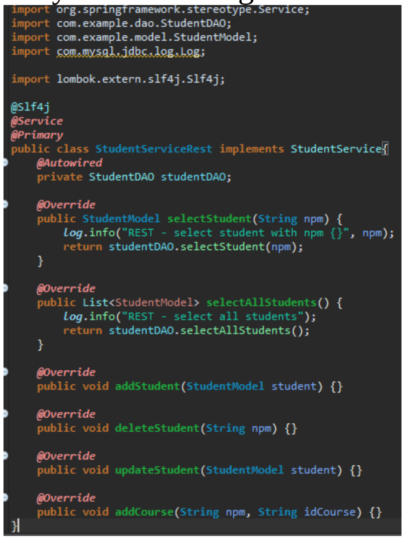
```java
package com.example.dao;

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



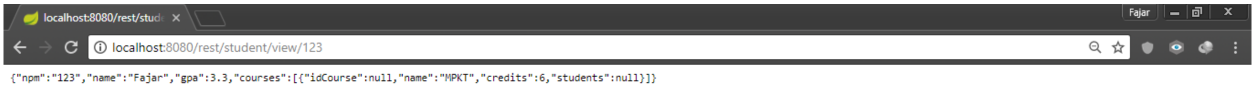
```java
package com.example.service;

import java.util.List;

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

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

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() {
        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
import org.springframework.stereotype.Service;
import com.example.dao.StudentDAO;
import com.example.model.StudentModel;
import com.mysql.jdbc.log.Log;

import lombok.extern.slf4j.Slf4j;

@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 studentDAO.selectAllStudents();
    }

    @Override
    public void addStudent(StudentModel student) {}

    @Override
    public void deleteStudent(String npm) {}

    @Override
    public void updateStudent(StudentModel student) {}

    @Override
    public void addCourse(String npm, String idCourse) {}
}
```

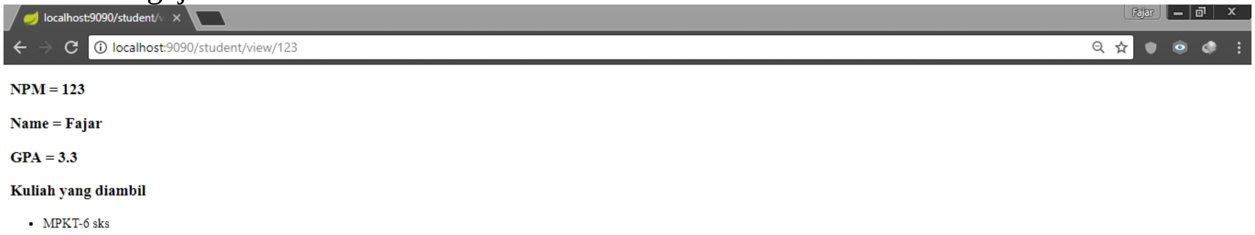
8. Jalankan kedua project Spring Boot untuk *Service Producer* dan *Service Consumer* tersebut
9. Pastikan *service producer* sudah berjalan dengan menjalankan **localhost:8080/rest/student/view/123**



localhost:8080/rest/student/view/123

```
{"npm":"123","name":"Fajar","gpa":3.3,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]}
```

10. Untuk menguji *service consumer* buka **localhost:9090/student/view/123**



localhost:9090/student/view/123

NPM = 123

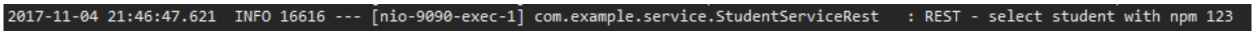
Name = Fajar

GPA = 3.3

Kuliah yang diambil

- MPKT-6 sks

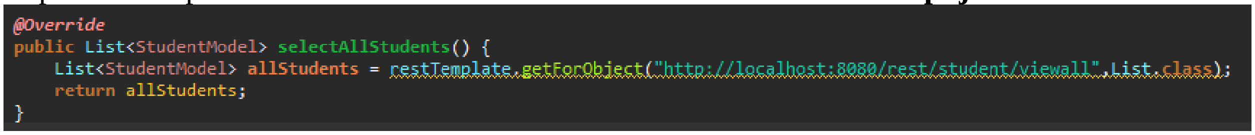
Dan di console tertulis



```
2017-11-04 21:46:47.621  INFO 16616 --- [nio-9090-exec-1] com.example.service.StudentServiceRest   : REST - select student with npm 123
```

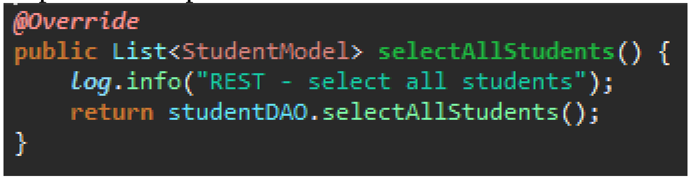
Latihan

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



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

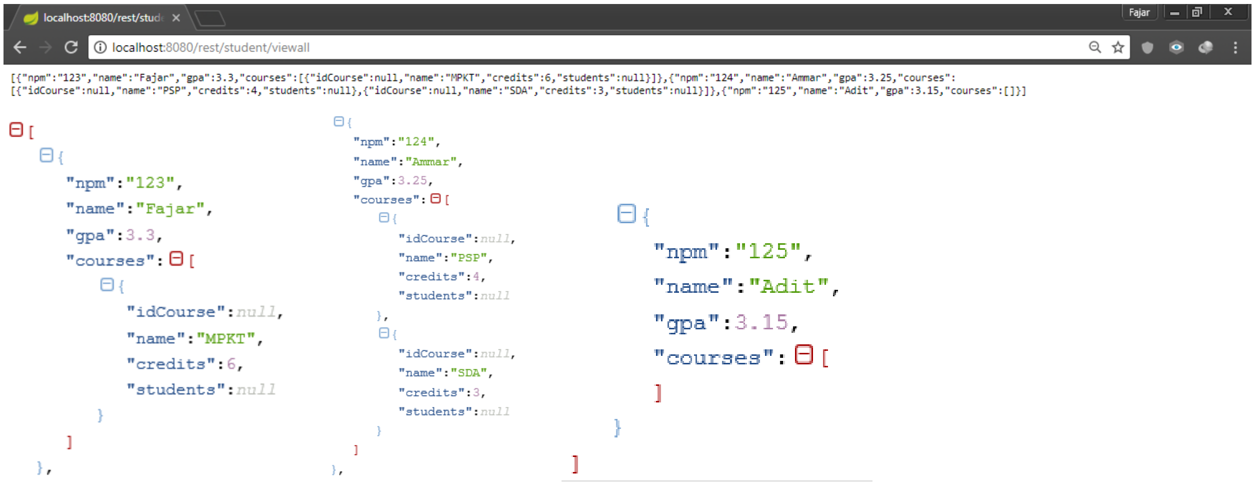
- Implementasi pada method **selectAllStudents** di kelas **StudentServiceRest**



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

Penjelasan : Implementasikan method tersebut dengan membuat List of Student Model yang mengambil object dari Rest Template **localhost:8080/rest/student/viewall** dengan objectnya yang berupa list

localhost:8080/rest/student/viewall

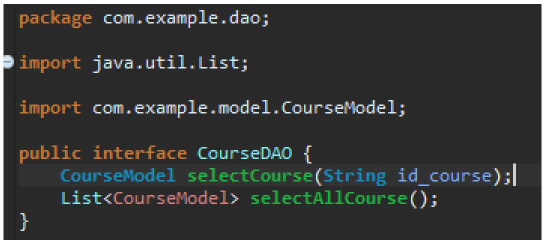


localhost:8080/rest/stud×
Fajar
localhost:8080/rest/student/viewall

[{"npm":"123","name":"Fajar","gpa":3.3,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]},{"npm":"124","name":"Ammar","gpa":3.25,"courses":
[{"idCourse":null,"name":"PSP","credits":4,"students":null},{"idCourse":null,"name":"SDA","credits":3,"students":null}]},{"npm":"125","name":"Adit","gpa":3.15,"courses":[]}]

```
⊟[
    ⊟{
        "npm":"123",
        "name":"Fajar",
        "gpa":3.3,
        "courses":⊟[
            ⊟{
                "idCourse":null,
                "name":"MPKT",
                "credits":6,
                "students":null
            }
        ]
    },
    ⊟{
        "npm":"124",
        "name":"Ammar",
        "gpa":3.25,
        "courses":⊟[
            ⊟{
                "idCourse":null,
                "name":"PSP",
                "credits":4,
                "students":null
            },
            ⊟{
                "idCourse":null,
                "name":"SDA",
                "credits":3,
                "students":null
            }
        ]
    },
    ⊟{
        "npm":"125",
        "name":"Adit",
        "gpa":3.15,
        "courses":⊟[
        ]
    }
]
```

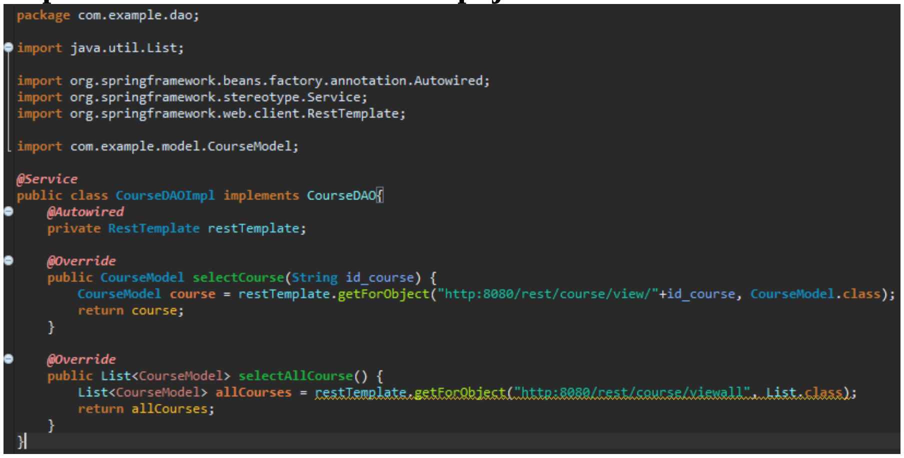
2. **Latihan 4:** Implementasikan *service consumer* untuk *class* CourseModel dengan membuat *class-class* DAO dan *service* baru
    - **Buat interface CourseDAO**



```
package com.example.dao;

import java.util.List;

import com.example.model.CourseModel;

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

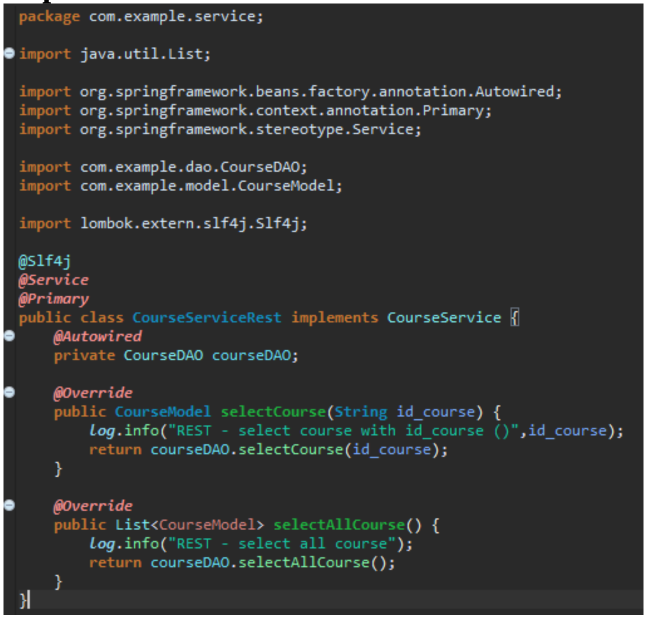
Fajar Bardan Salam | 1506689401
APAP-A

- **Implementasikan CourseDAOImpl.java**



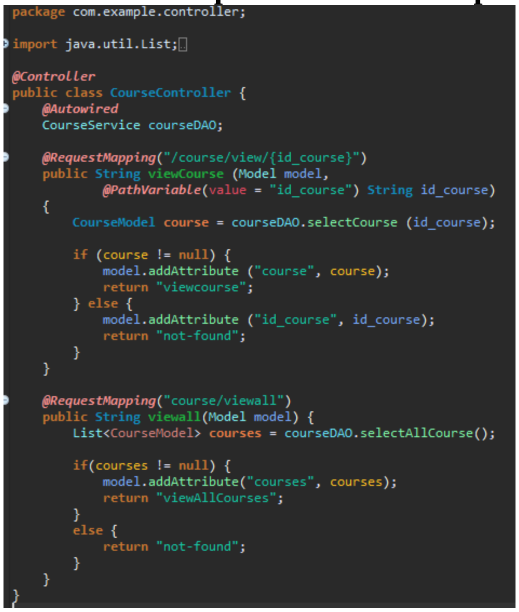
```
package com.example.dao;

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.model.CourseModel;

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

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

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

- **Implementasikan CourseServiceRest**



```
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.model.CourseModel;

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> selectAllCourse() {
        log.info("REST - select all course");
        return courseDAO.selectAllCourse();
    }
}
```

- **Tambahkan implementasi viewall pada CourseController**



```
package com.example.controller;

import java.util.List;

@Controller
public class CourseController {
    @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";
        }
    }

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

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

Penjelasan : Langkah implementasi yaitu pertama, buat interface CourseDAO, kemudian buat CourseDAOImpl yang mengimplementasikan CourseDAO. Dimana pad akelas ini kita akan mengambil object dari RestTemplate ke url localhost:8080. Kemudian buat kelas CourseServiceRest yang mengimplement CourseService agar dapat digunakan pada Controller. Kemudian, lengkapi method pada kelas CourseController dan coba jalankan program.

Lesson learned :
Pada tutorial ke-7 ini saya mempelajari bagaimana membuat web service menggunakan spring boot framework. Saya juga mempelajari bagaimana menggunakan REST pada Spring. Selain itu, saya juga mempelajari adanya pembagian pada *Service Consumer* dan *Service Producer* untuk saling bertukar data.