Aisyah Husna
1506689686
APAP - C

Tutorial 7 - ADPAP

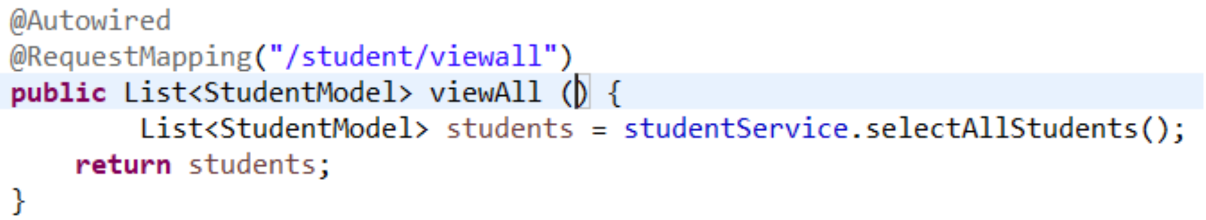
Membuat Web Service Menggunakan Spring Boot Framework

Membuat Service producer

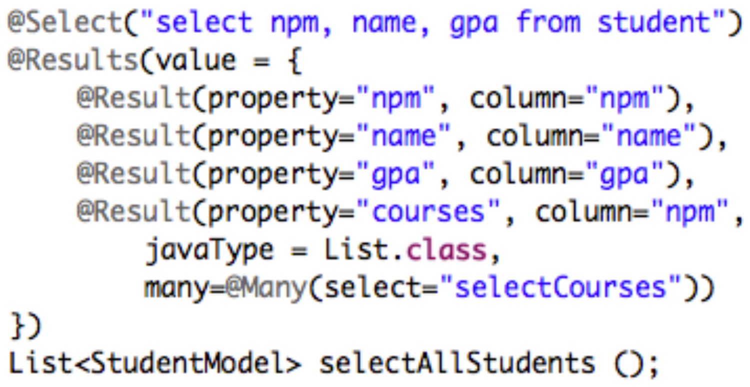
Latihan 1:

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"**

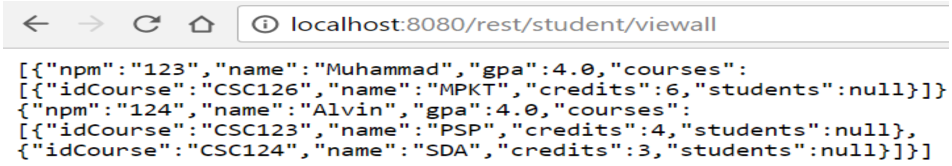
Langkahnya yaitu menyalin method selectStudent sebelumnya, kemudian method diubah dengan mengembalikan *arraylist of students*.



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

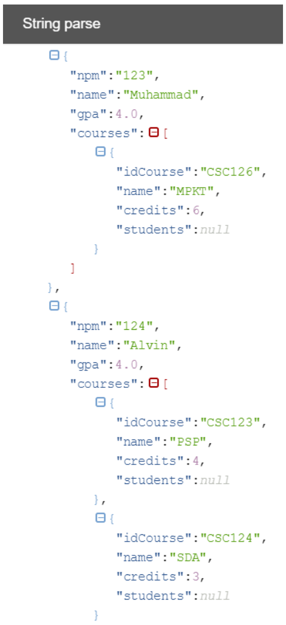


```
@Select("select npm, name, gpa from student")
@Results(value = {
    @Result(property="npm", column="npm"),
    @Result(property="name", column="name"),
    @Result(property="gpa", column="gpa"),
    @Result(property="courses", column="npm",
        javaType = List.class,
        many=@Many(select="selectCourses"))
})
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}]}]
```

Aisyah Husna
1506689686
APAP - C



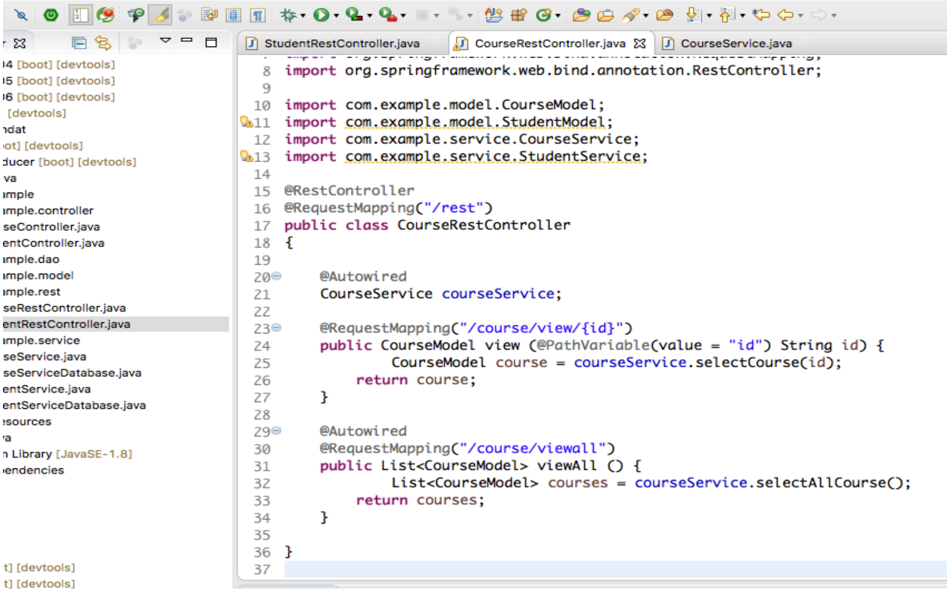
```
{
   "npm":"123",
   "name":"Muhammad",
   "gpa":4.0,
   "courses":[
      {
         "idCourse":"CSC126",
         "name":"MPKT",
         "credits":6,
         "students":null
      }
   ]
},
{
   "npm":"124",
   "name":"Alvin",
   "gpa":4.0,
   "courses":[
      {
         "idCourse":"CSC123",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

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

Latihan 2:

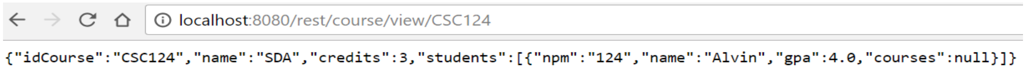
Buatlah service untuk class Course. Buatlah controller baru yang terdapat service untuk melihat suatu course dengan masukan ID Course (view by ID) dan service untuk elihat semua course (view all)

Method – method yang dibuat kurang lebih sama dengan method student untuk view dan viewall, hanya saja setiap "student" diubah menjadi course.

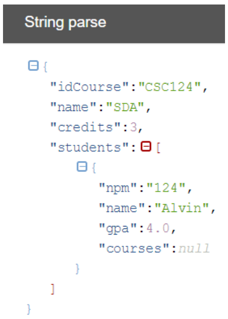


```
import org.springframework.web.bind.annotation.RestController;

import com.example.model.CourseModel;
import com.example.model.StudentModel;
import com.example.service.CourseService;
import com.example.service.StudentService;

@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;
    }

}
```

Aisyah Husna
1506689686
APAP - C



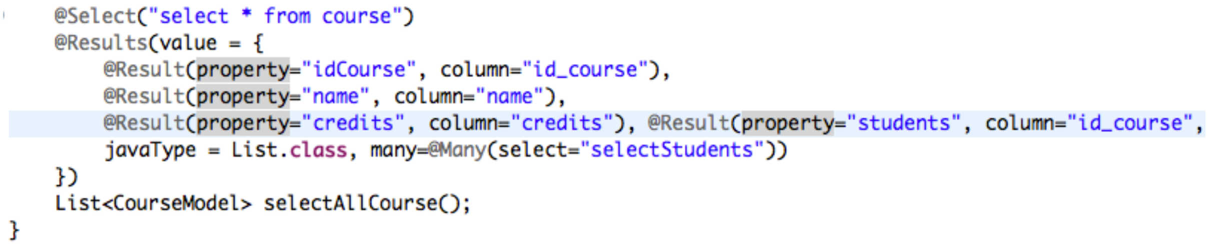
```
{"idCourse":"CSC124","name":"SDA","credits":3,"students":[{"npm":"124","name":"Alvin","gpa":4.0,"courses":null}]}
```



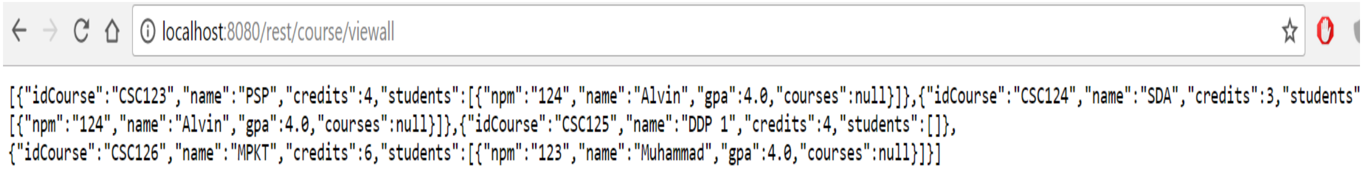
```
⊟{
   "idCourse":"CSC124",
   "name":"SDA",
   "credits":3,
   "students":⊟[
      ⊟{
         "npm":"124",
         "name":"Alvin",
         "gpa":4.0,
         "courses":null
      }
   ]
}
```

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

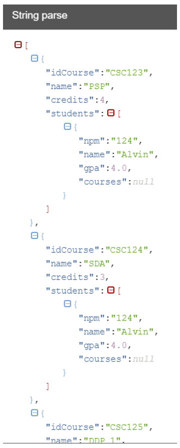
View all course



```
    @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="selectStudents"))
    })
    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}]}]
```



```
⊟[
   ⊟{
      "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*.**

Aisyah Husna
1506689686
APAP - C

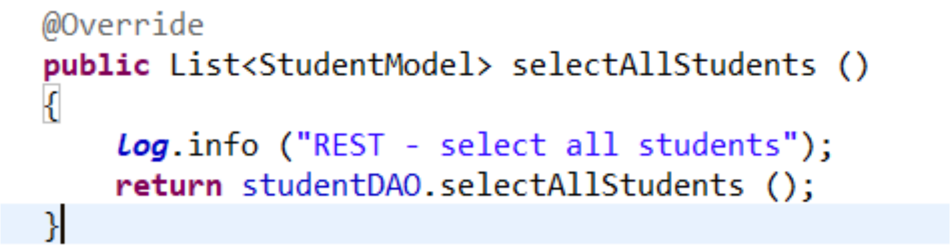
Membuat Service Consumer

Latihan 3:

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

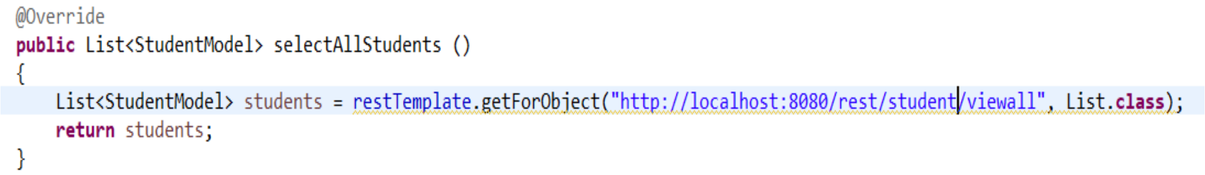
Method dibuat pertama-tama pada section berikut:

StudentServiceRest:

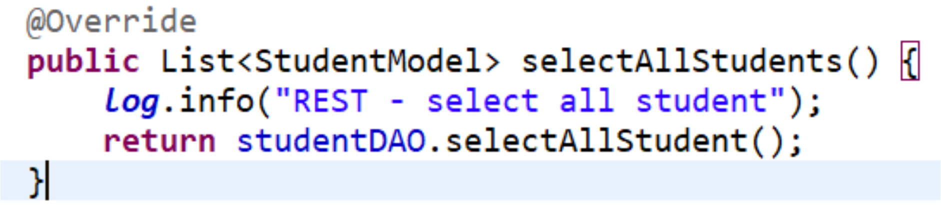


```
@Override
public List<StudentModel> selectAllStudents ()
{
    log.info ("REST - select all students");
    return studentDAO.selectAllStudents ();
}
```

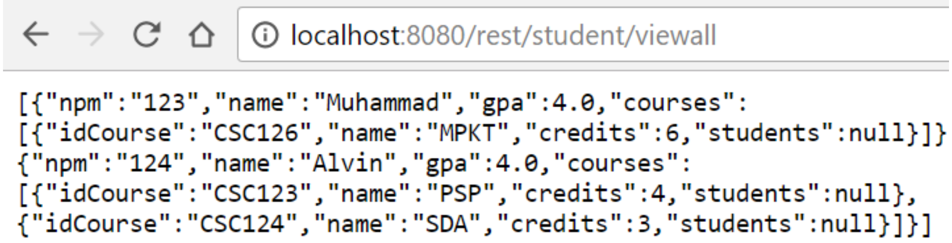
StudentDAOImpl:



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



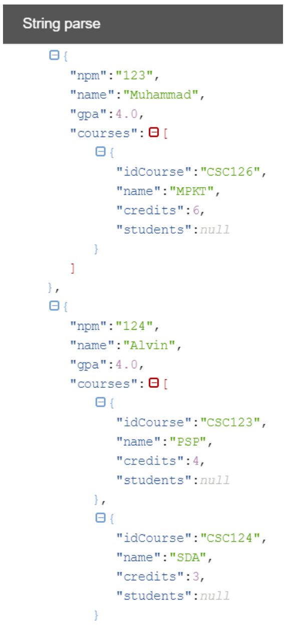
```
@Override
public List<StudentModel> selectAllStudents() {
    log.info("REST - select all student");
    return studentDAO.selectAllStudent();
}
```



```
[{"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}]}]
```

Kemudian method dapat menampilkan view dibawah ini, dengan menggunakan “getForObject” dari RestTemplate, yang merujuk ke localhost:8080/rest/student/viewall

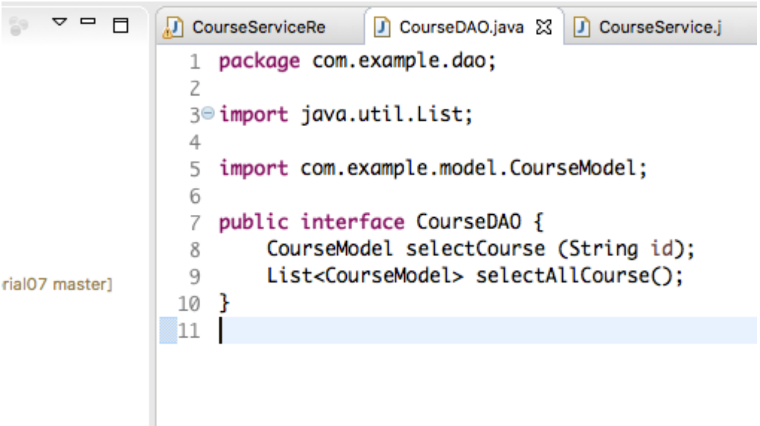
Aisyah Husna
1506689686
APAP - C



```
{
   "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
      }
```

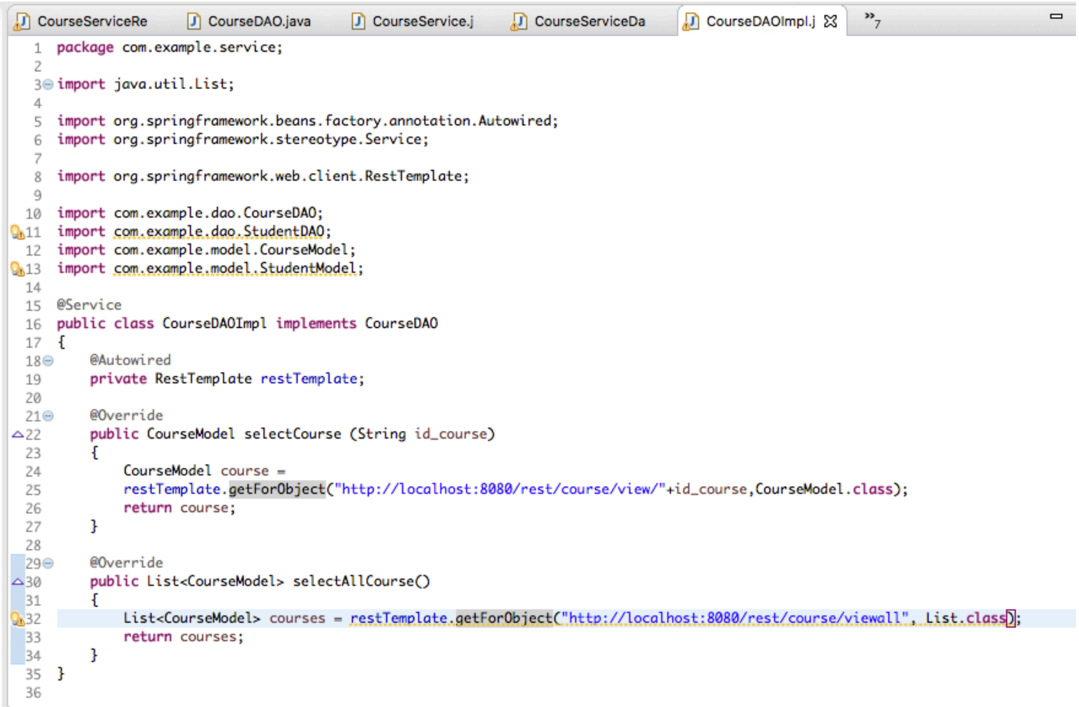
Latihan 4:

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

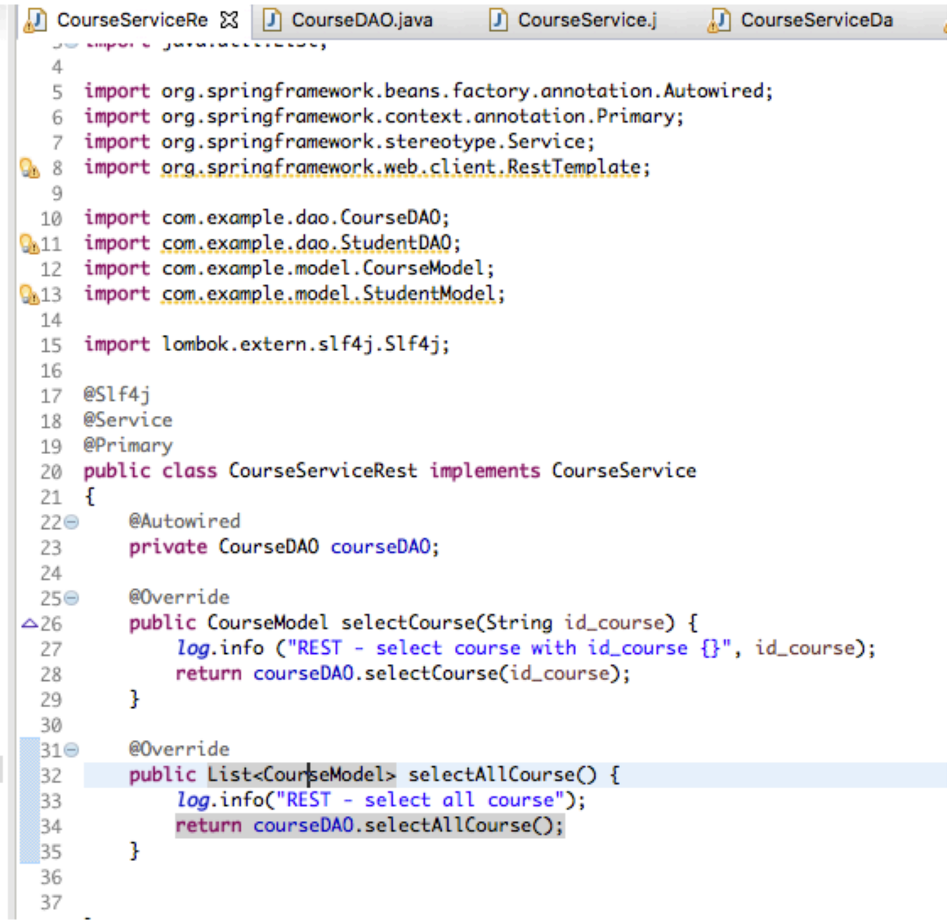


```java
package com.example.dao;

import java.util.List;

import com.example.model.CourseModel;

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

Aisyah Husna
1506689686
APAP - C

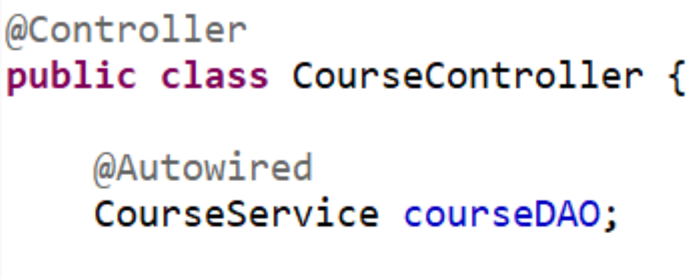


```java
package com.example.service;

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;
    }
}
```

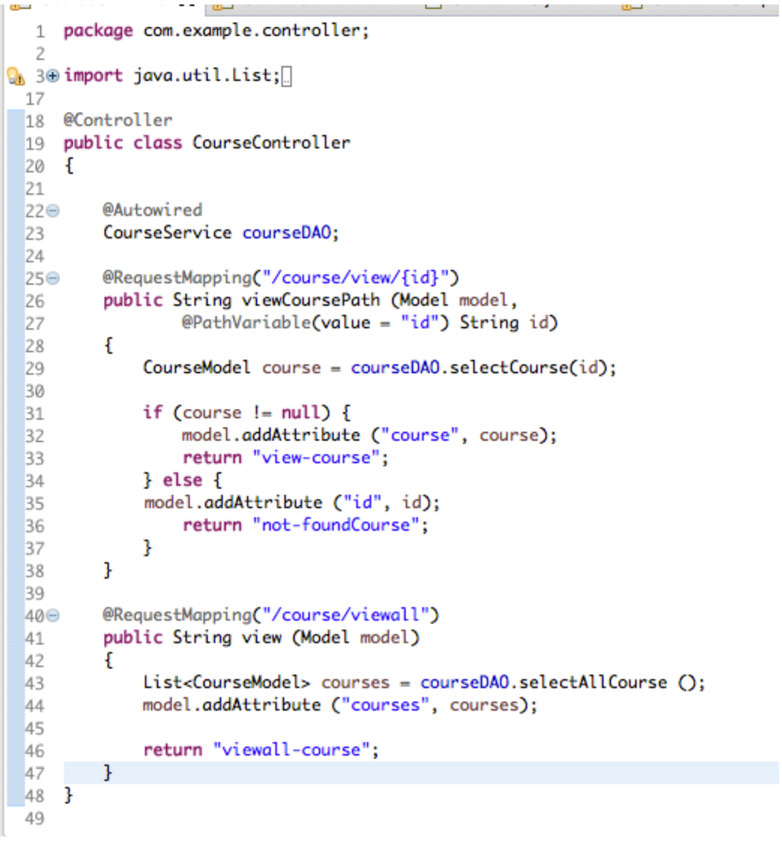


```java

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

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();
    }


```

Aisyah Husna
1506689686
APAP - C



```java
@Controller
public class CourseController {

    @Autowired
    CourseService courseDAO;
```



```java
package com.example.controller;

import java.util.List;

@Controller
public class CourseController
{

    @Autowired
    CourseService courseDAO;

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

        if (course != null) {
            model.addAttribute ("course", course);
            return "view-course";
        } else {
        model.addAttribute ("id", id);
            return "not-foundCourse";
        }
    }

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

        return "viewall-course";
    }
}
```

Penjelasan: **Sebelum membuat CourseDAOImpl, kita membuat *interface* terlebih dahulu. Kemudian CourseDaoImpl 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.**

Aisyah Husna
1506689686
APAP - C

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. Saya belajar bagaimana cara menggunakan web service dalam framework springboot, dengan membagikan aplikasi menjadi 2 layer, yaitu layer consumer dan layer producer. Tujuannya yaitu untuk memisahkan *backend* dan *frontend*. Untuk consumer bagian *frontend,* sementara *producer* merupakan bagian backend. Untuk itu, dengan mempelajari bagaimana cara menggunakan REST kita dapat memahami *RestTemplate* yang berguna untuk memisahkan *backend dan frontend* tersebut.