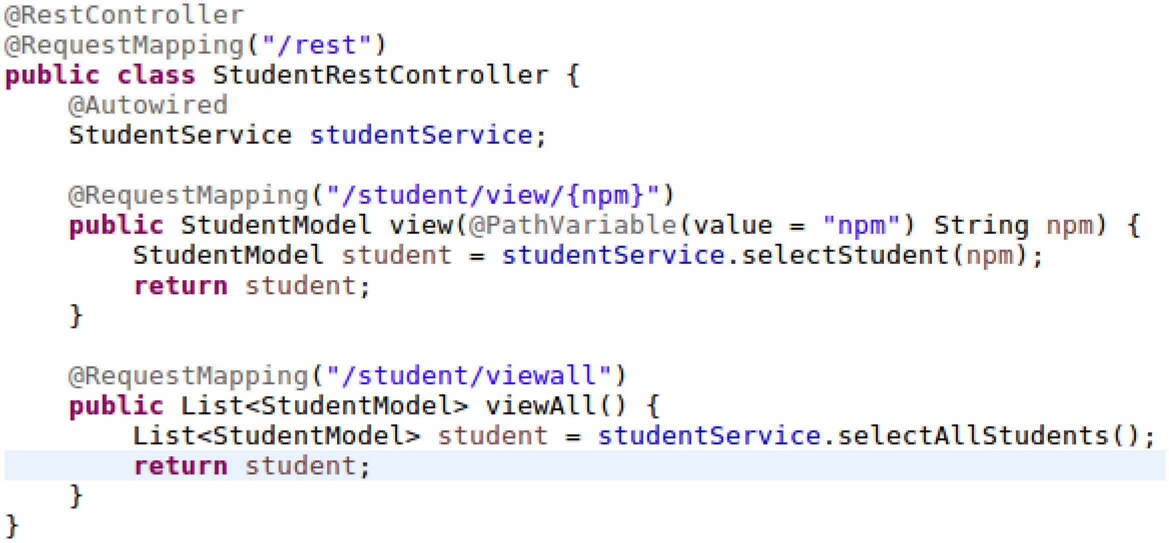
Latihan

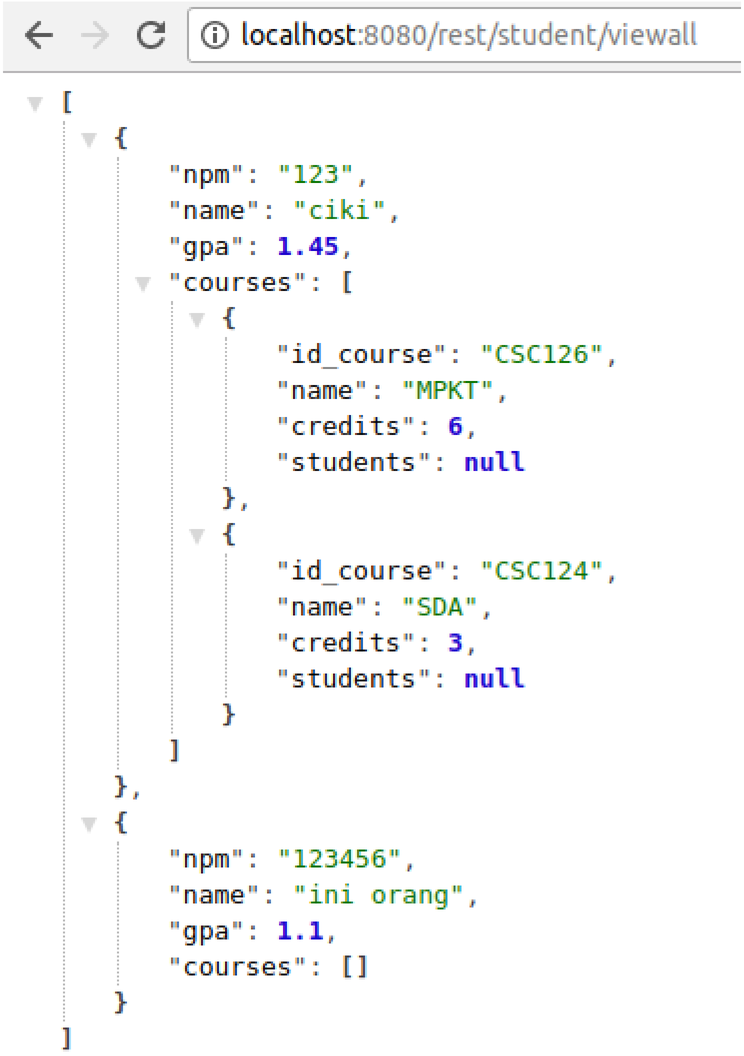
1. REST View All Students
   a) Pada StudentRestController tambahkan routing untuk /student/viewall



```
@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;
    }

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

List<StudentModel> diisi dengan menggunakan service database yang telah dibuat pada tutorial sebelumnya.

   b) Hasil

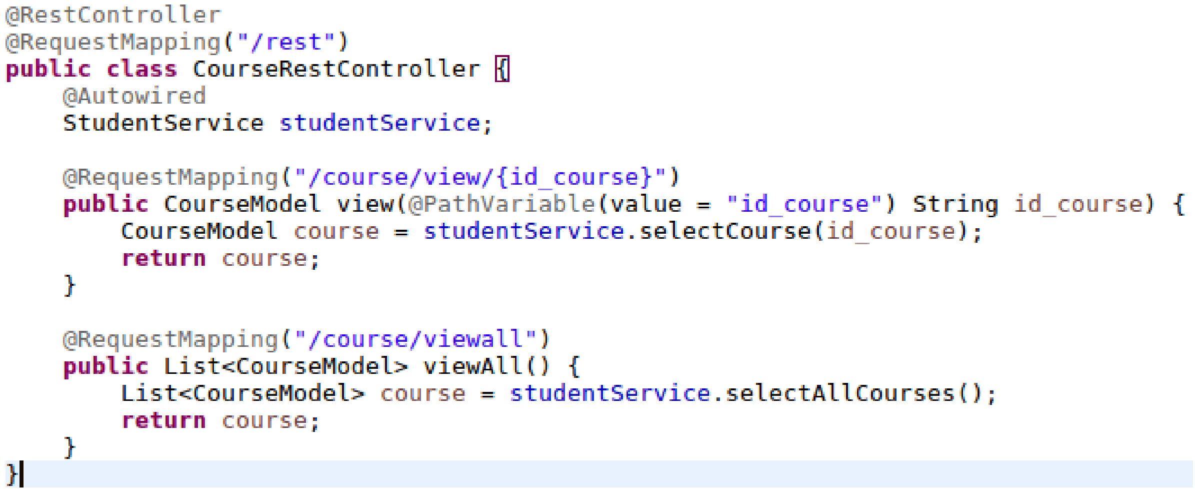


localhost:8080/rest/student/viewall

```
[
    {
        "npm": "123",
        "name": "ciki",
        "gpa": 1.45,
        "courses": [
            {
                "id_course": "CSC126",
                "name": "MPKT",
                "credits": 6,
                "students": null
            },
            {
                "id_course": "CSC124",
                "name": "SDA",
                "credits": 3,
                "students": null
            }
        ]
    },
    {
        "npm": "123456",
        "name": "ini orang",
        "gpa": 1.1,
        "courses": []
    }
]
```

2. REST untuk view Course by Id dan view all Courses

   a) Membuat class CourseRestController

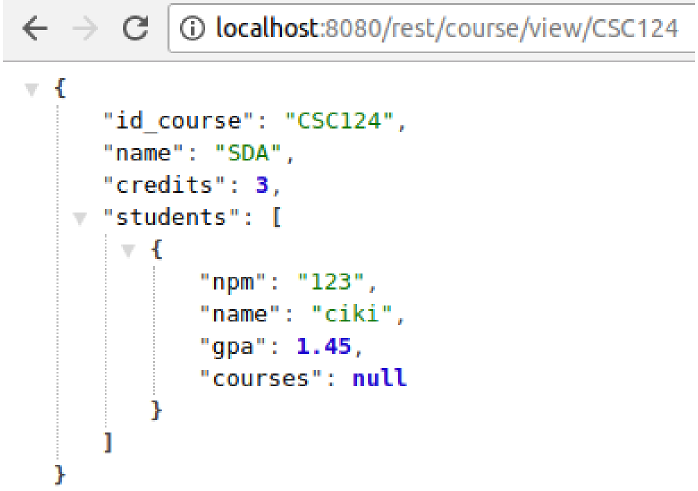


```
@RestController
@RequestMapping("/rest")
public class CourseRestController {
    @Autowired
    StudentService studentService;

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

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

Implementasi dilakukan sama seperti pada REST Students, menggunakan service database yang telah dibuat pada tutorial sebelumnya.

   b) Hasil

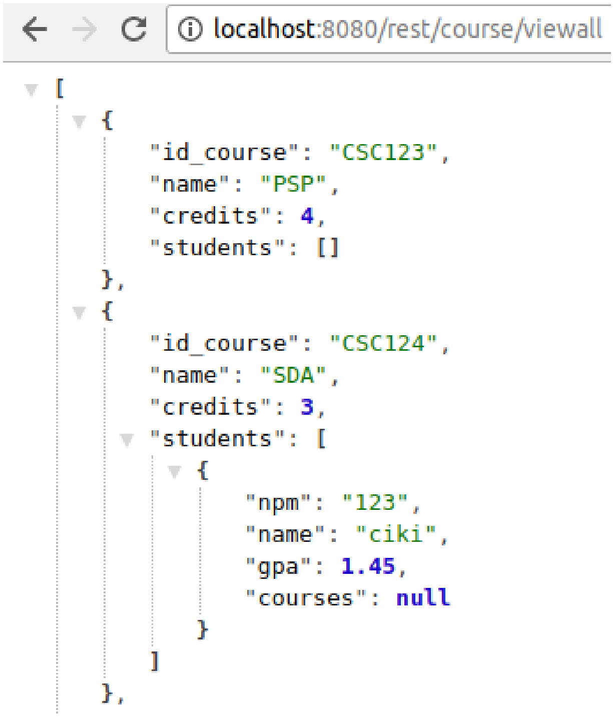
      i. Select by Id



localhost:8080/rest/course/view/CSC124

```
{
    "id_course": "CSC124",
    "name": "SDA",
    "credits": 3,
    "students": [
        {
            "npm": "123",
            "name": "ciki",
            "gpa": 1.45,
            "courses": null
        }
    ]
}
```

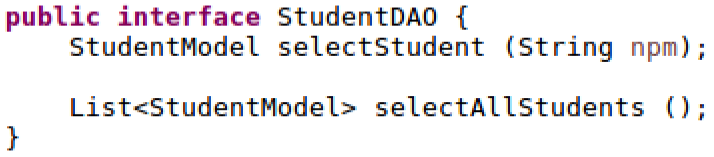
      ii. Select all



localhost:8080/rest/course/viewall

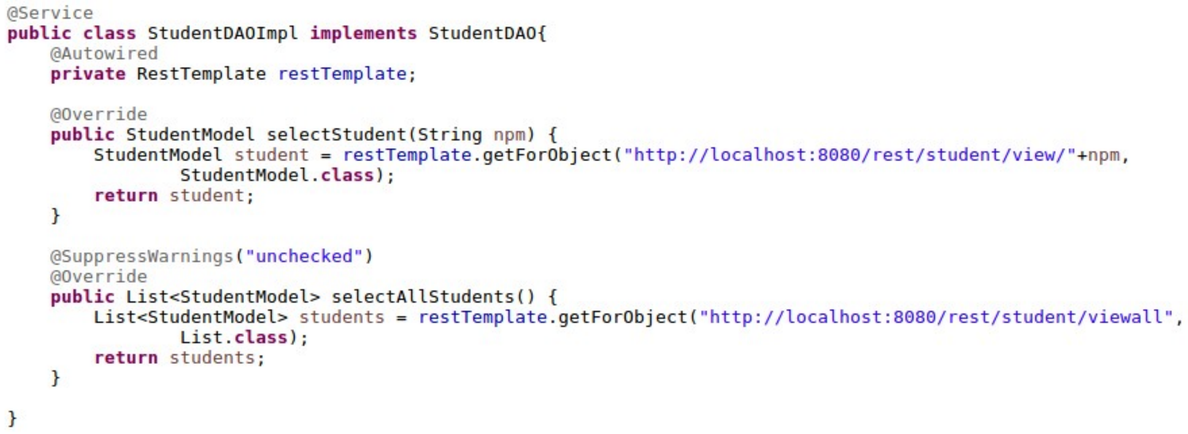
```
[
    {
        "id_course": "CSC123",
        "name": "PSP",
        "credits": 4,
        "students": []
    },
    {
        "id_course": "CSC124",
        "name": "SDA",
        "credits": 3,
        "students": [
            {
                "npm": "123",
                "name": "ciki",
                "gpa": 1.45,
                "courses": null
            }
        ]
    },
```

3. View All Students menggunakan REST
   a) Pada StudentDAO, tambahkan method selectAllStudents



```
public interface StudentDAO {
    StudentModel selectStudent (String npm);

    List<StudentModel> selectAllStudents ();
}
```

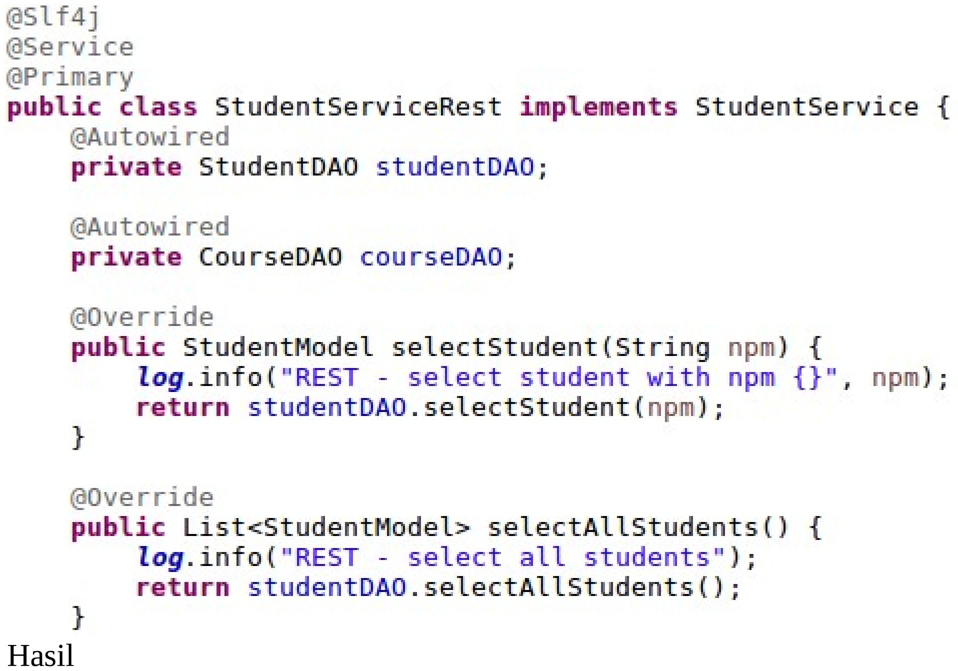
   b) Pada StudentDAOImpl, tambahkan implementasi method selectAllStudents



```
@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;
    }

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

}
```

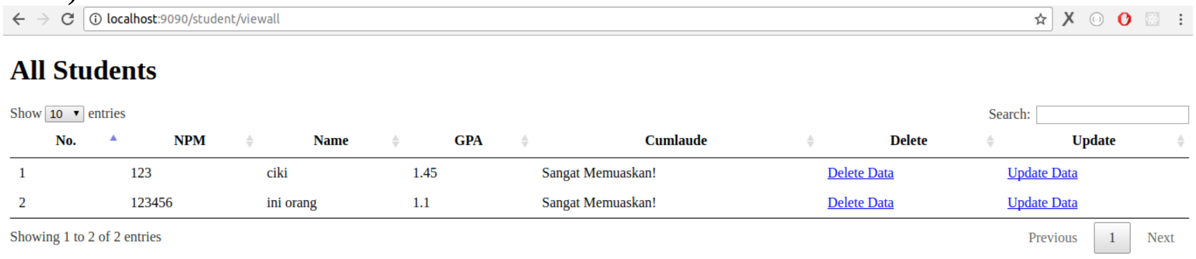
Type class dari restTemplate.getForObject menggunakan List, bukan StudentModel karena data Json tersebut adalah List of StudentModel.

   c) Pada StudentServiceRest, tambahkan method selectAllStudents yang mengambil data melalui StudentDAOImpl.



```
@Slf4j
@Service
@Primary
public class StudentServiceRest implements StudentService {
    @Autowired
    private StudentDAO studentDAO;

    @Autowired
    private CourseDAO courseDAO;

    @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();
    }
```

   d) Hasil



All Students

Show 10 entries

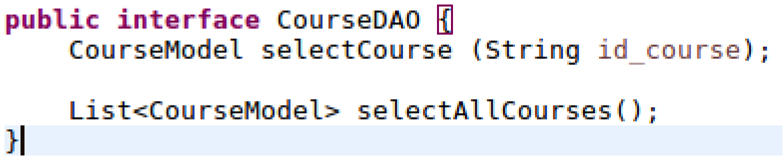
Search:

No.	NPM	Name	GPA	Cumlaude	Delete	Update
1	123	ciki	1.45	Sangat Memuaskan!	Delete Data	Update Data
2	123456	ini orang	1.1	Sangat Memuaskan!	Delete Data	Update Data

Showing 1 to 2 of 2 entries

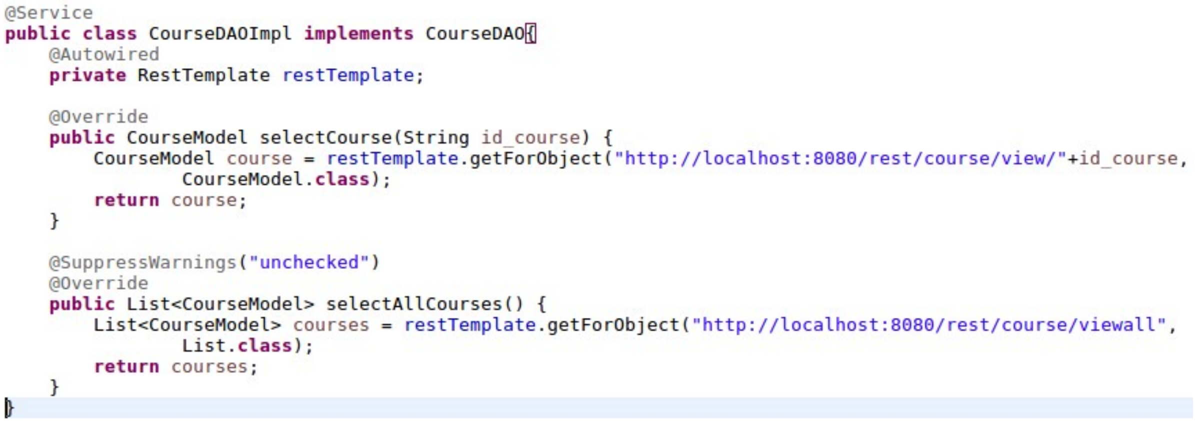
4. View Course by Id dan view all Courses menggunakan REST

   a) Membuat interface StudentDAO



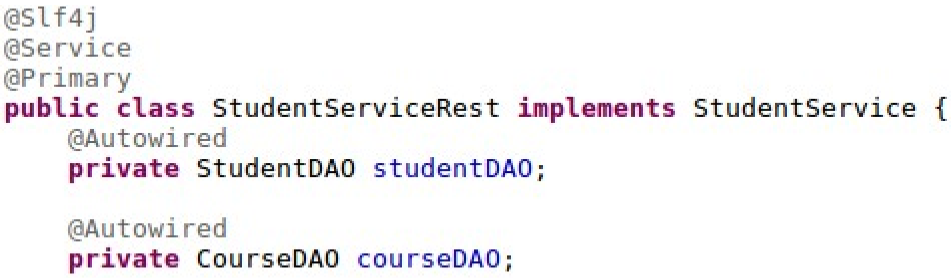
```
public interface CourseDAO {
    CourseModel selectCourse (String id_course);

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

   b) Membuat class StudentDAOImpl



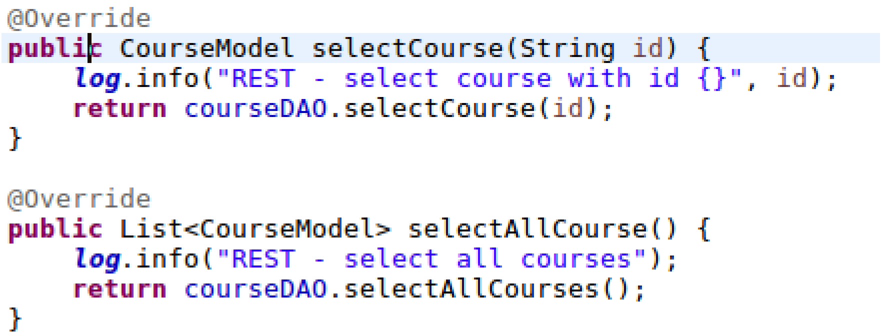
```
@Service
public class CourseDAOImpl implements CourseDAO{
    @Autowired
    private RestTemplate restTemplate;

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

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

   c) Pada StudentServiceRest, tambahkan autowired CourseDAO



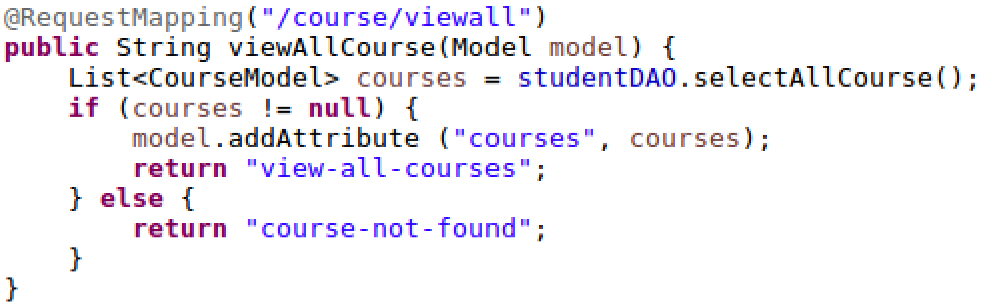
```
@Slf4j
@Service
@Primary
public class StudentServiceRest implements StudentService {
    @Autowired
    private StudentDAO studentDAO;

    @Autowired
    private CourseDAO courseDAO;
```

   d) Pada StudentServiceRest, tambahkan implementasi selectCourse dan selectAllCourses



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

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

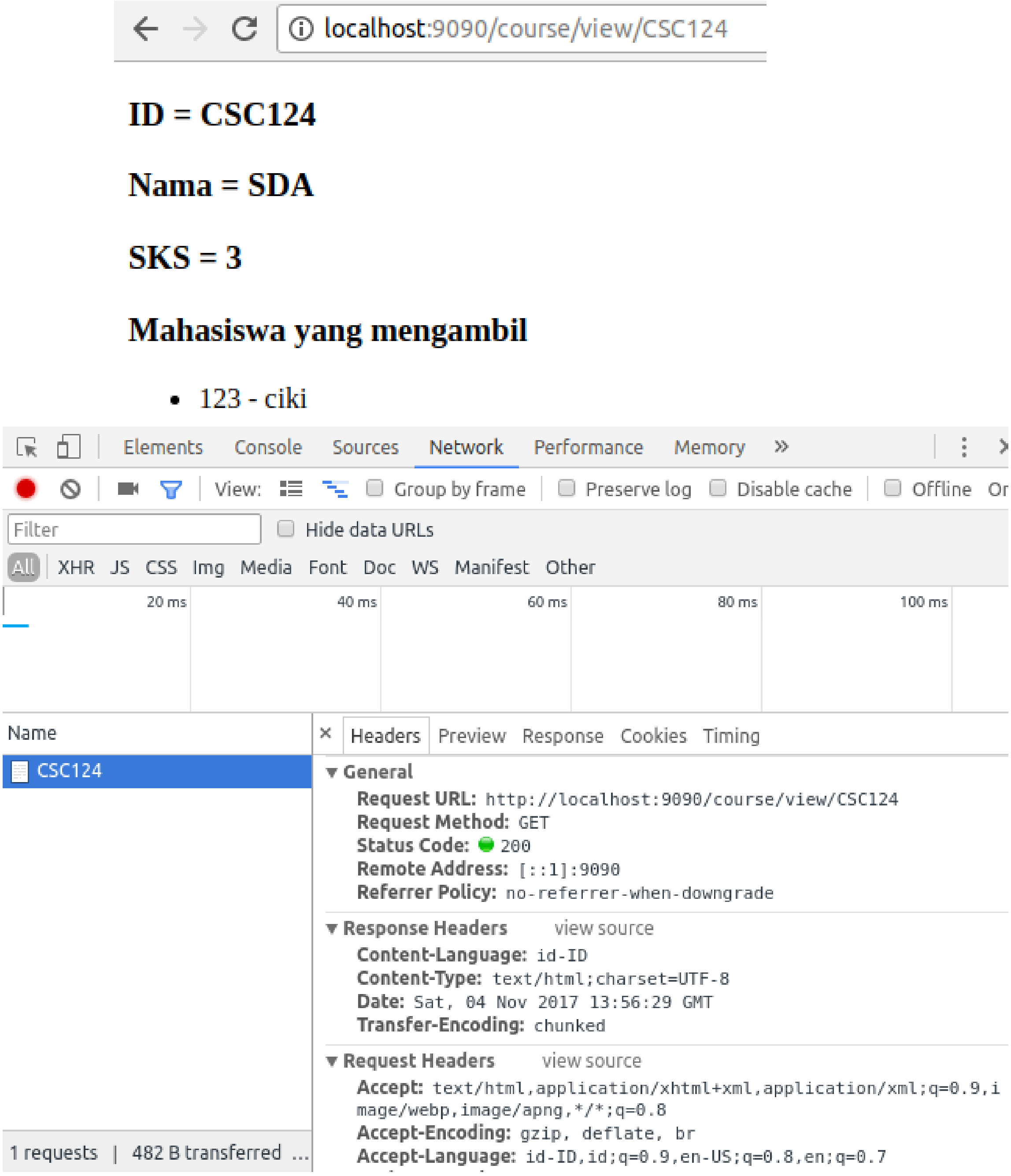
   e) Pada Controller, tambahkan routing untuk course/viewall karena pada tutorial sebelumnya tidak ada fungsi tersebut



```
@RequestMapping("/course/viewall")
public String viewAllCourse(Model model) {
    List<CourseModel> courses = studentDAO.selectAllCourse();
    if (courses != null) {
        model.addAttribute ("courses", courses);
        return "view-all-courses";
    } else {
        return "course-not-found";
    }
}
```

f) Hasil

i. Select by Id



localhost:9090/course/view/CSC124

ID = CSC124

Nama = SDA

SKS = 3

Mahasiswa yang mengambil

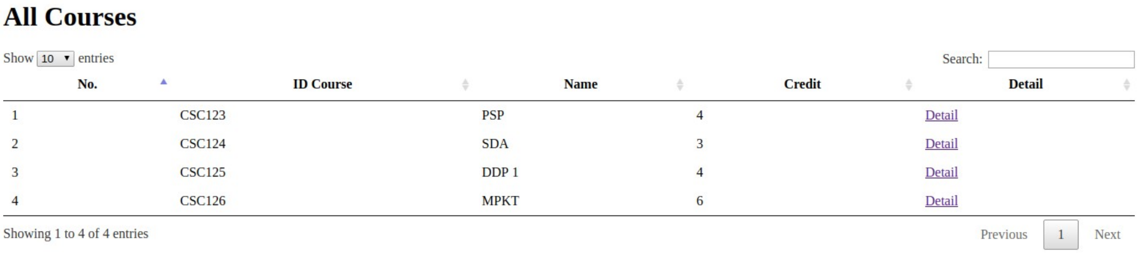
- 123 - ciki

CSC124

```
▼ General
   Request URL: http://localhost:9090/course/view/CSC124
   Request Method: GET
   Status Code: ● 200
   Remote Address: [::1]:9090
   Referrer Policy: no-referrer-when-downgrade
▼ Response Headers     view source
   Content-Language: id-ID
   Content-Type: text/html;charset=UTF-8
   Date: Sat, 04 Nov 2017 13:56:29 GMT
   Transfer-Encoding: chunked
▼ Request Headers     view source
   Accept: text/html,application/xhtml+xml,application/xml;q=0.9,i
   mage/webp,image/apng,*/*;q=0.8
   Accept-Encoding: gzip, deflate, br
   Accept-Language: id-ID,id;q=0.9,en-US;q=0.8,en;q=0.7
```

1 requests | 482 B transferred ...

ii. Select All



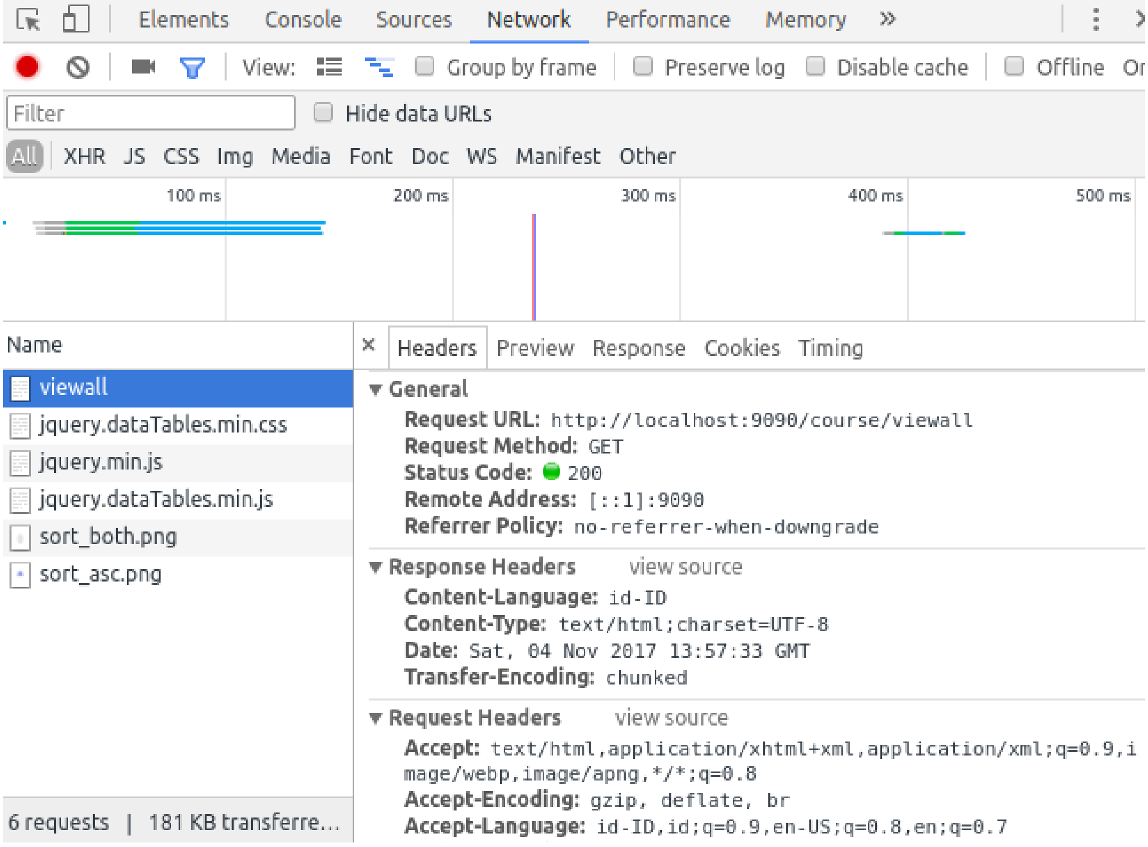
All Courses

Show 10 entries

Search:

No.	ID Course	Name	Credit	Detail
1	CSC123	PSP	4	Detail
2	CSC124	SDA	3	Detail
3	CSC125	DDP 1	4	Detail
4	CSC126	MPKT	6	Detail

Showing 1 to 4 of 4 entries



viewall
jquery.dataTables.min.css
jquery.min.js
jquery.dataTables.min.js
sort_both.png
sort_asc.png

6 requests | 181 KB transferre…

▼ General

Request URL: http://localhost:9090/course/viewall
Request Method: GET
Status Code: ● 200
Remote Address: [::1]:9090
Referrer Policy: no-referrer-when-downgrade

▼ Response Headers view source

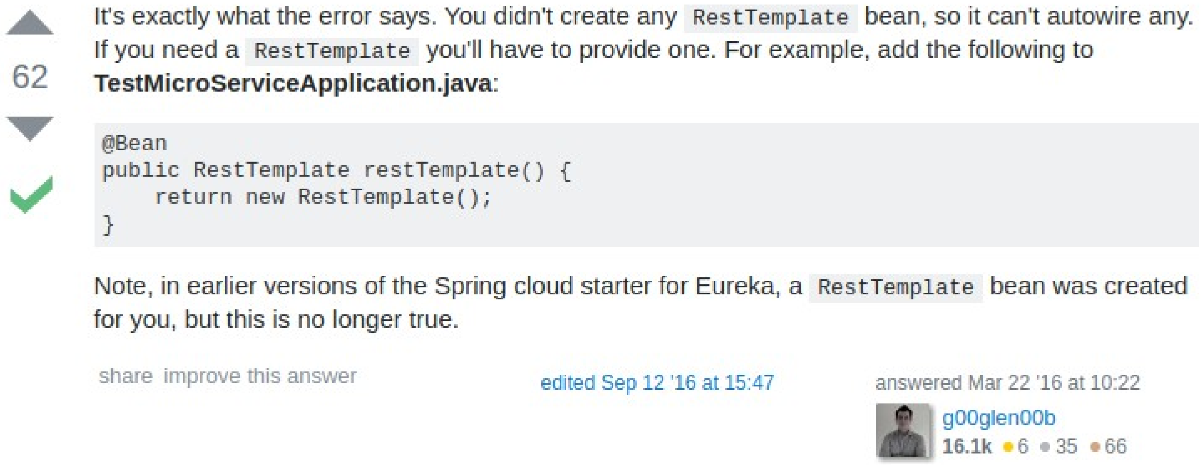
Content-Language: id-ID
Content-Type: text/html;charset=UTF-8
Date: Sat, 04 Nov 2017 13:57:33 GMT
Transfer-Encoding: chunked

▼ Request Headers view source

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate, br
Accept-Language: id-ID,id;q=0.9,en-US;q=0.8,en;q=0.7

Apa Yang Dipelajari

1. RestTemplate tidak lagi menjadi library default untuk SpringBoot.



It's exactly what the error says. You didn't create any `RestTemplate` bean, so it can't autowire any. If you need a `RestTemplate` you'll have to provide one. For example, add the following to **TestMicroServiceApplication.java**:

```
@Bean
public RestTemplate restTemplate() {
    return new RestTemplate();
}
```

Note, in earlier versions of the Spring cloud starter for Eureka, a `RestTemplate` bean was created for you, but this is no longer true.

share improve this answer

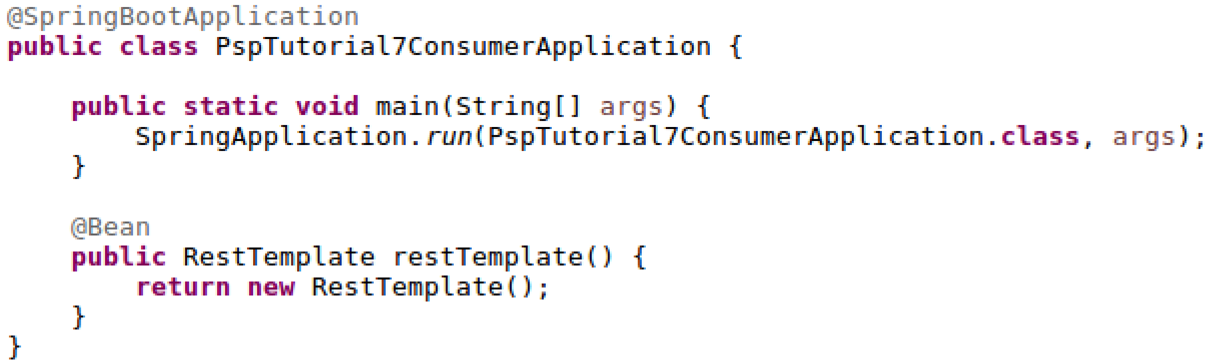
edited Sep 12 '16 at 15:47

answered Mar 22 '16 at 10:22

g00glen00b

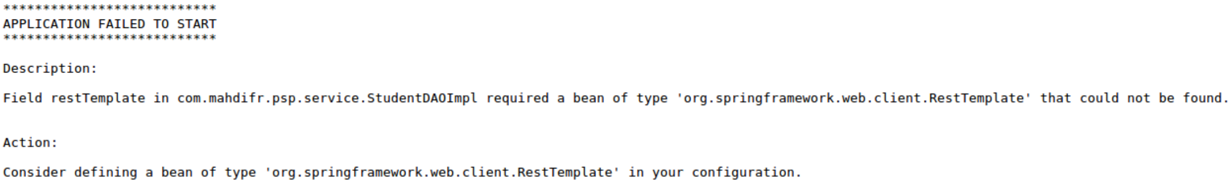
16.1k ●6 ●35 ●66

Sehingga pada implementasi tutorial ini harus ditambahkan @bean pada application



```
@SpringBootApplication
public class PspTutorial7ConsumerApplication {

    public static void main(String[] args) {
        SpringApplication.run(PspTutorial7ConsumerApplication.class, args);
    }

    @Bean
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }
}
```

Jika @bean tersebut tidak ada, maka akan muncul error sebagai berikut



```
***************************
APPLICATION FAILED TO START
***************************

Description:

Field restTemplate in com.mahdifr.psp.service.StudentDAOImpl required a bean of type 'org.springframework.web.client.RestTemplate' that could not be found.


Action:

Consider defining a bean of type 'org.springframework.web.client.RestTemplate' in your configuration.
```