

Latihan 1

StudentRestController.java

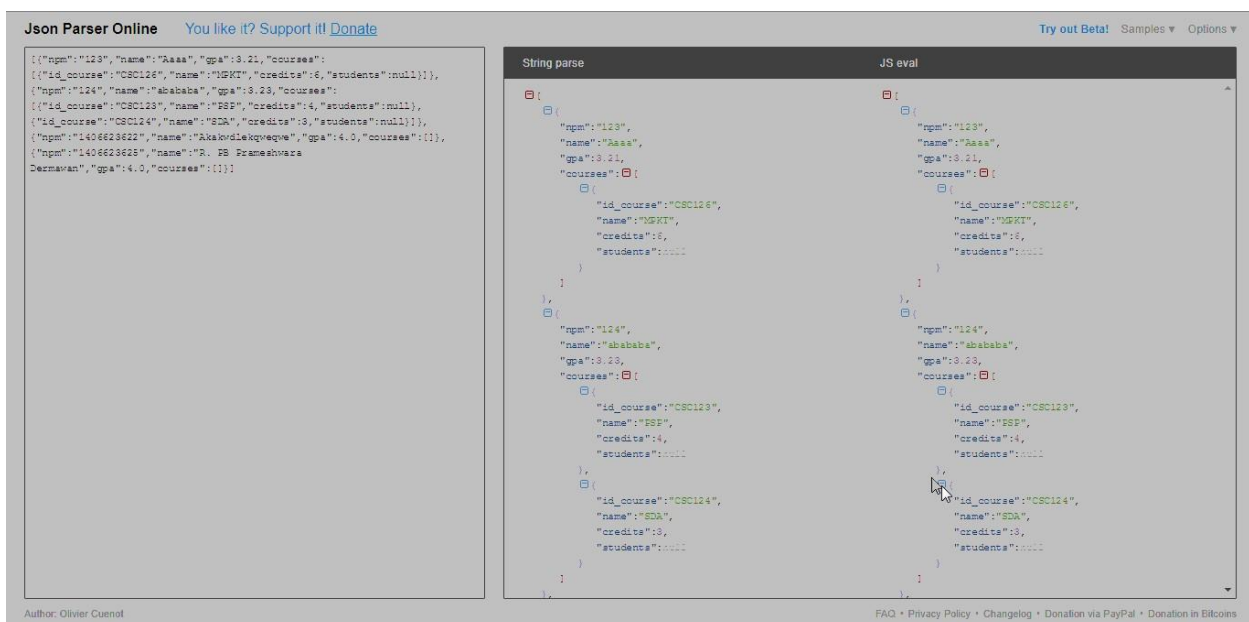
```
StudentRestController.java
1 package com.example.rest;
2
3 import java.util.List;
4
5
6
7
8
9
10
11
12
13 @RestController
14 @RequestMapping("/rest")
15 public class StudentRestController {
16     @Autowired
17     StudentService studentService;
18
19     @RequestMapping("/student/view/{npm}")
20     public StudentModel view(@PathVariable(value = "npm") String npm) {
21         StudentModel student = studentService.selectStudent(npm);
22         return student;
23     }
24
25     @RequestMapping("/student/viewall")
26     public List<StudentModel> view() {
27         List<StudentModel> students = studentService.selectAllStudents();
28         return students;
29     }
30 }
```

Adanya tambahan kode dari baris 25 hingga 30, yaitu sebuah request mapping ke /rest/student/viewall yang akan mengembalikan list semua student yang ada [ada database sesuai mapping yang ada di Mapper melalui service.

Berikut adalah screenshot dari percobaan melalui browser :



```
[[{"npm":"123","name":"Aaaa","gpa":3.21,"courses":[{"id_course":"CSC126","name":"HUKT","credits":6,"students":null}]}],{"npm":"124","name":"Abababa","gpa":3.23,"courses":[{"id_course":"CSC123","name":"PSP","credits":4,"students":null}]}],{"id_course":"CSC124","name":"SDA","credits":3,"students":null}]]],{"npm":"1406623622","name":"Akakudlekqueve","gpa":4.0,"courses":[]}],{"npm":"1406623625","name":"R. PB Prameshwara Dermawan","gpa":4.0,"courses":[]}]
```



Json Parser Online You like it? Support it! Donate

String parse JS eval

```
{
  "npm": "123",
  "name": "Aaaa",
  "gpa": 3.21,
  "courses": [
    {
      "id_course": "CSC126",
      "name": "HUKT",
      "credits": 6,
      "students": null
    }
  ]
},
{
  "npm": "124",
  "name": "Abababa",
  "gpa": 3.23,
  "courses": [
    {
      "id_course": "CSC123",
      "name": "PSP",
      "credits": 4,
      "students": null
    }
  ]
},
{
  "id_course": "CSC124",
  "name": "SDA",
  "credits": 3,
  "students": null
}],
{
  "npm": "1406623622",
  "name": "Akakudlekqueve",
  "gpa": 4.0,
  "courses": []
},
{
  "npm": "1406623625",
  "name": "R. PB Prameshwara Dermawan",
  "gpa": 4.0,
  "courses": []
}]
```

Author: Olivier Cuenot

FAQ • Privacy Policy • Changelog • Donation via PayPal • Donation in Bitcoins

Latihan 2

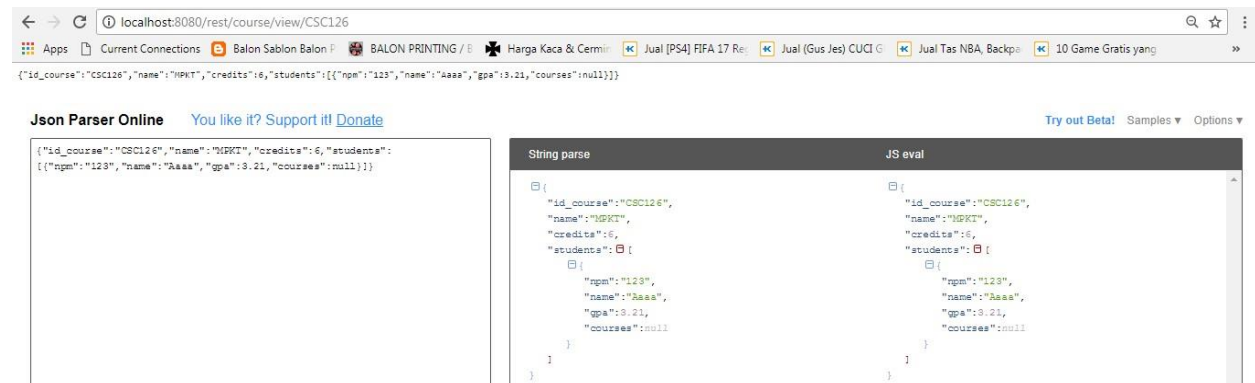
CourseRestController.java

```
CourseRestController.java
1 package com.example.rest;
2
3 import java.util.List;
4
5 @RestController
6 @RequestMapping("/rest")
7 public class CourseRestController {
8     @Autowired
9     CourseService courseService;
10
11     @RequestMapping("/course/view/{id_course}")
12     public CourseModel view(@PathVariable(value = "id_course") String id_course) {
13         CourseModel course = courseService.selectCourse(id_course);
14         return course;
15     }
16
17     @RequestMapping("/course/viewall")
18     public List<CourseModel> view() {
19         List<CourseModel> courses = courseService.selectAllCourses();
20         return courses;
21     }
22 }
```

Terdapat 2 Method yang ada, yaitu view course sesuai id_course dan view all course untuk melihat seluruh course yang ada.

Berikut adalah screenshot percobaan melalui browser :

Method 1 (View Course by ID)



The screenshot shows a web browser window with the address bar displaying `localhost:8080/rest/course/view/CSC126`. Below the browser window, there is a JSON Parser Online interface. The input field contains the following JSON:

```
{
  "id_course": "CSC126",
  "name": "MPKI",
  "credits": 6,
  "students": [
    {
      "npm": "123",
      "name": "Aaaa",
      "gpa": 3.21,
      "courses": null
    }
  ]
}
```

The output field shows the parsed JSON structure:

```
{
  "id_course": "CSC126",
  "name": "MPKI",
  "credits": 6,
  "students": [
    {
      "npm": "123",
      "name": "Aaaa",
      "gpa": 3.21,
      "courses": null
    }
  ]
}
```

Method 2 (View All Course)

localhost:8080/rest/course/viewall

Apps Current Connections Balon Sablon Balon P BALON PRINTING / B Harga Kaca & Cermir Jual [PS4] FIFA 17 Re Jual (Gus Jes) CUCI G Jual Tas NBA, Ba

```
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "abababa", "gpa": 3.23, "courses": null}], {"id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "abababa", "gpa": 3.23, "courses": null}], {"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students": []}, {"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": [{"npm": "123", "name": "Aaaa", "gpa": 3.21, "courses": null}]}
```

Json Parser Online You like it? Support it! Donate

Try out Beta! Samples Options

String parse JS eval

```
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "abababa", "gpa": 3.23, "courses": null}], {"id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "abababa", "gpa": 3.23, "courses": null}], {"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students": []}, {"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": [{"npm": "123", "name": "Aaaa", "gpa": 3.21, "courses": null}]}
```

Author: Olivier Cuenot

FAQ • Privacy Policy • Changelog • Donation via PayPal • Donation in Bitcoins

Latihan 3

StudentDAO.java

```
StudentDAO.java
1 package com.example.dao;
2
3 import java.util.List;
4
5
6 public interface StudentDAO {
7     StudentModel selectStudent (String npm);
8     List<StudentModel> selectAllStudents ();
9 }
10
11
```

Ditambahkan sebuah method selectAllStudents();

StudentDAOImpl.java

```
StudentDAOImpl.java
1 package com.example.dao;
2
3 import java.util.List;
4
5
6 @Service
7 public class StudentDAOImpl implements StudentDAO {
8     @Autowired
9     private RestTemplate restTemplate;
10
11     @Override
12     public StudentModel selectStudent(String npm) {
13         StudentModel student = restTemplate.getForObject("http://localhost:8080/rest/student/view/" + npm,
14             StudentModel.class);
15         return student;
16     }
17
18     @Override
19     public List<StudentModel> selectAllStudents() {
20         List<StudentModel> students = restTemplate.getForObject("http://localhost:8080/rest/student/viewall",
21             List.class);
22         return students;
23     }
24 }
25
26
```

Mengoverride method selectAllStudents() yang ada pada StudentDAO.java. Pada method tersebut akan mengembalikan sebuah list yang berisi semua student yang objectnya didapatkan melalui restTemplate dari REST yang sudah disediakan oleh server Producer sehingga StudentDAOImpl ini akan menggantikan fungsi dari mapper yang mengambil dari database. Pada consumer tidak perlu mengambil dari database, hanya perlu dari REST yang disediakan karena REST di producer sudah mengambil dari database seperlunya terlebih dahulu.

StudentServiceRest.java

```
StudentServiceRest.java
1 package com.example.service;
2
3 import java.util.List;
4
5 @Slf4j
6 @Service
7 @Primary
8 public class StudentServiceRest implements StudentService {
9     @Autowired
10     private StudentDAO studentDAO;
11
12     @Override
13     public StudentModel selectStudent(String npm) {
14         log.info("REST - select student with npm {}", npm);
15         return studentDAO.selectStudent(npm);
16     }
17
18     @Override
19     public List<StudentModel> selectAllStudents() {
20         log.info("REST - select all students");
21         return studentDAO.selectAllStudents();
22     }
23
24     @Override
25     public void addStudent(StudentModel student) {
26     }
27 }
```

Pada StudentServiceRest.java ditambahkan method dari baris 27 sampai 31 yaitu akan mengoverride method yang sama pada StudentService.java namun menggunakan StudentDAO yang dimodifikasi secara manual.

Berikut adalah screenshot percobaan melalui browser :

Tampilan Server Producer untuk view all students



The screenshot shows a web browser window with the address bar displaying `localhost:8080/rest/student/viewall`. The browser tabs include "Apps", "Current Connections", "Balon Sablon Balon P", and "BALON PRINTING / E". The main content area displays a JSON response from the REST API, which is a list of student records. Each record contains fields for `npm`, `name`, `gpa`, and `courses`. The response includes data for two students: one with `npm: 123` and `name: Aaaa`, and another with `npm: 1406623622` and `name: Akakwdlekqweqwe`.

```
[{"npm": "123", "name": "Aaaa", "gpa": 3.21, "courses": [{"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": null}]}, {"npm": "1406623622", "name": "Akakwdlekqweqwe", "gpa": 4.0, "courses": []}]
```

[Tampilan Server Consumer untuk view all students](#)

- [Home](#)
- [Daftar Mahasiswa](#)
- [Menambah Mahasiswa](#)

All Students

Show entries

Search:

No	NPM	Name	GPA	Predikat	Delete	Update
No. 1	NPM = 123	Name = Aaaa	GPA = 3.21	Sangat Memuaskan!	Delete Data	Update Data
No. 2	NPM = 124	Name = abababa	GPA = 3.23	Sangat Memuaskan!	Delete Data	Update Data
No. 3	NPM = 1406623622	Name = Akakwdlekqweqwe	GPA = 4.0	Cum Laude!	Delete Data	Update Data
No. 4	NPM = 1406623625	Name = R. PB Prameshwara Dermawan	GPA = 4.0	Cum Laude!	Delete Data	Update Data

Showing 1 to 4 of 4 entries

[Previous](#) [Next](#)

Mata Kuliah APAP

Latihan 4

CourseDAO.java

```
CourseDAO.java
1 package com.example.dao;
2
3 import java.util.List;
4
5
6
7 public interface CourseDAO {
8     CourseModel selectCourse(String id_course);
9
10    List<CourseModel> selectAllCourses();
11 }
```

Memasukkan 2 buat method kerangka yaitu view course by id dan view all course

CourseDAOImpl.java

```
CourseDAOImpl.java
1 package com.example.dao;
2
3 import java.util.List;
4
5
6
7
8
9
10
11 @Service
12 public class CourseDAOImpl implements CourseDAO{
13     @Autowired
14     private RestTemplate restTemplate;
15
16
17
18     @Override
19     public CourseModel selectCourse(String id_course) {
20         CourseModel course = restTemplate.getForObject("http://localhost:8080/rest/course/view/" + id_course,
21             CourseModel.class);
22         return course;
23     }
24
25     @Override
26     public List<CourseModel> selectAllCourses() {
27         List<CourseModel> courses = restTemplate.getForObject("http://localhost:8080/rest/course/viewall", List.class);
28         return courses;
29     }
30
31 }
```

Implementasi dari CourseDAO.java. Mengoverride method selectCourse(String id_course) dan selectAllCourses() yang ada pada CourseDAO.java. Pada masing-masing method tersebut akan mengembalikan sebuah course sesuai ID dan sebuah list yang berisi semua course. Object coursanya didapatkan melalui restTemplate dari REST yang sudah disediakan oleh server Producer sehingga CourseDAOImpl ini akan menggantikan fungsi dari mapper yang mengambil dari database. Pada consumer tidak perlu mengambil dari database, hanya perlu dari REST yang disediakan karena REST di producer sudah mengambil dari database seperlunya terlebih dahulu.

CourseServiceRest.java

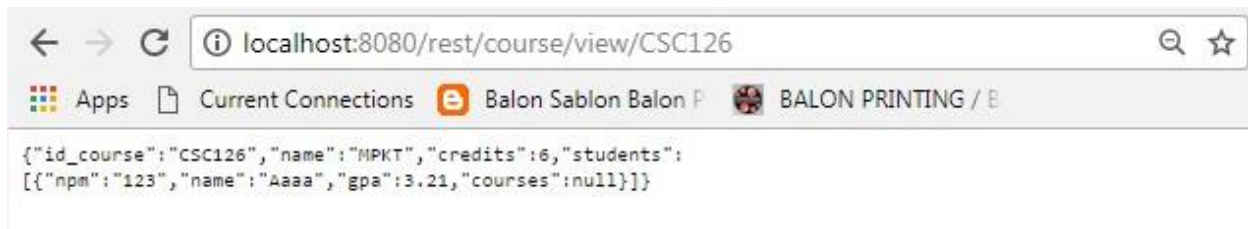
```
CourseServiceRest.java
1 package com.example.service;
2
3 import java.util.List;
4
5 @Slf4j
6 @Service
7 @Primary
8 public class CourseServiceRest implements CourseService{
9     @Autowired
10     private CourseDAO courseDAO;
11
12     @Override
13     public CourseModel selectCourse(String id_course) {
14         log.info("REST-select course with id_course {}", id_course);
15         return courseDAO.selectCourse(id_course);
16     }
17
18     @Override
19     public List<CourseModel> selectAllCourses() {
20         log.info("REST - select all courses");
21         return courseDAO.selectAllCourses();
22     }
23 }
24
25 }
```

Mengoverride method yang ada pada CourseService yaitu selectCourse by id dan selectAllCourses, namun pada class ini menggunakan courseDAO yang telah dimodifikasi secara manual.

Berikut adalah screenshot dari percobaan melalui browser :

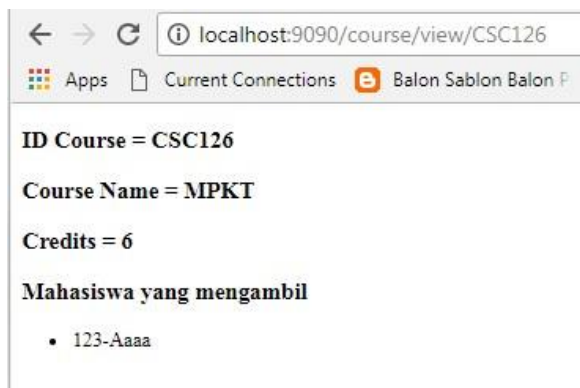
Method 1 (View Course by ID)

Tampilan Producer



The screenshot shows a web browser window with the address bar displaying `localhost:8080/rest/course/view/CSC126`. The browser's developer tools or console shows a JSON response: `{ "id_course": "CSC126", "name": "MPKT", "credits": 6, "students": [{ "npm": "123", "name": "Aaaa", "gpa": 3.21, "courses": null }] }`. The browser's taskbar at the bottom shows icons for 'Apps', 'Current Connections', and 'Balon Sablon Balon P'.

Tampilan Consumer



The screenshot shows a web browser window with the address bar displaying `localhost:9090/course/view/CSC126`. The browser's taskbar at the bottom shows icons for 'Apps', 'Current Connections', and 'Balon Sablon Balon P'. The main content area of the browser displays the following information:

- ID Course = CSC126**
- Course Name = MPKT**
- Credits = 6**
- Mahasiswa yang mengambil**
 - 123-Aaaa

Method 2 (View All Courses)

Tampilan Producer

```
localhost:8080/rest/course/viewall

[{"id_course":"CSC123","name":"PSP","credits":4,"students":
[{"npm":"124","name":"abababa","gpa":3.23,"courses":null}]},
{"id_course":"CSC124","name":"SDA","credits":3,"students":
[{"npm":"124","name":"abababa","gpa":3.23,"courses":null}]}, {"id_course":"CSC125","name":"DDP
1","credits":4,"students":[]}, {"id_course":"CSC126","name":"MPKT","credits":6,"students":
[{"npm":"123","name":"Aaaa","gpa":3.21,"courses":null}]}
```

Tampilan Consumer

localhost:9090/course/viewall

[Home](#)
[Daftar Mahasiswa](#)
[Menambah Mahasiswa](#)

Mata Kuliah APAP

List All Courses

No.	ID Course	Course Name	SKS	Mahasiswa Yang Mengambil
1	CSC123	PSP	4	• 124-abababa
2	CSC124	SDA	3	• 124-abababa
3	CSC125	DDP 1	4	
4	CSC126	MPKT	6	• 123-Aaaa

[Home](#)
[Daftar Mahasiswa](#)
[Menambah Mahasiswa](#)

Mata Kuliah APAP

Lesson Learned

Dalam Tutorial ini saya belajar mengenai web service dimana ada dua sisi server yang bekerja dalam project ini, yaitu sisi producer dan consumer. Pada sisi server producer akan mempersiapkan REST dimana database akan diambil seperlunya sesuai kebutuhan consumer, sehingga saat consumer ingin mengakses data yang diperlukan dalam viewnya, producer hanya perlu mengirimkan REST yang berisi data seperlunya tanpa memperbolehkan consumer mengakses database secara penuh. Hal tersebut tentunya sangat mendukung privasi data dan keamanannya.