

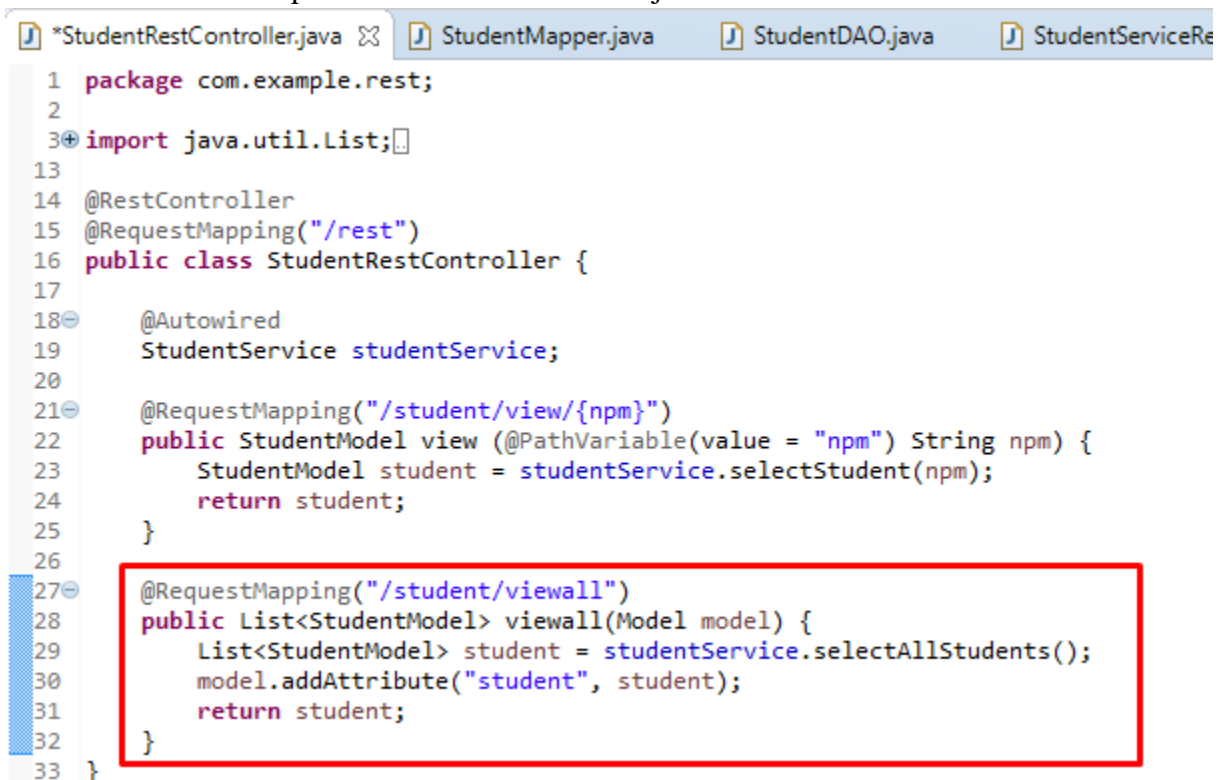
- **Hal-hal yang dipelajari dalam tutorial 7**

Dalam tutorial 7 ini dipelajari mengenai *Web Service*. Pengaplikasian dari *web service* ini adalah dengan melakukan pemisahan layer *backend* dengan *frontend*, hal ini bertujuan supaya performa *hardware* dari masing-masing mesin server dapat dapat dimaksimalkan dengan sebaik mungkin. Layer backend dan frontend ini direpresentasikan dengan *service producer* untuk layer *backend* dan *service consumer* untuk layer *frontend*. *Service consumer* merupakan aplikasi yang berinteraksi dengan users, sedangkan *service producer* merupakan aplikasi yang memberikan data kepada *service consumer* sesuai dengan data yang diminta. Untuk data yang dikembalikan oleh *service producer* ke *service consumer* direpresentasikan menggunakan JSON. Dalam tutorial ini menggunakan anotasi **@Primary** yang berguna untuk meng-*handle* data yang diambil pada *StudentService* agar data yang terambil tersebut merupakan data yang berasal dari *web service* pada *StudentServiceRest*.

- **Latihan**

- **Latihan 1**

Method *viewall* pada *StudentRestController.java*



```
1 package com.example.rest;
2
3 import java.util.List;
4
13
14 @RestController
15 @RequestMapping("/rest")
16 public class StudentRestController {
17
18     @Autowired
19     StudentService studentService;
20
21     @RequestMapping("/student/view/{npm}")
22     public StudentModel view (@PathVariable(value = "npm") String npm) {
23         StudentModel student = studentService.selectStudent(npm);
24         return student;
25     }
26
27     @RequestMapping("/student/viewall")
28     public List<StudentModel> viewall(Model model) {
29         List<StudentModel> student = studentService.selectAllStudents();
30         model.addAttribute("student", student);
31         return student;
32     }
33 }
```

Pada *method* diatas data *student* akan diambil menggunakan *method* *selectAllStudents()* dan akan mengembalikan semua data *student* dalam bentuk *list* dan ditampilkan pada *browser* seperti pada gambar dibawah ini.

➤ Latihan 2

Method `selectAllStudents()` pada `StudentServiceRest.java`

```
1 package com.example.service;
2
3 import java.util.List;
4
13
14 @Slf4j
15 @Service
16 @Primary
17 public class StudentServiceRest implements StudentService {
18
19     @Autowired
20     private StudentDAO studentDAO;
21
22     @Override
23     public StudentModel selectStudent(String npm) {
24         // TODO Auto-generated method stub
25         log.info("REST - select student with npm {}", npm);
26         return studentDAO.selectStudent(npm);
27     }
28
29     @Override
30     public List<StudentModel> selectAllStudents() {
31         // TODO Auto-generated method stub
32         log.info("REST - select all students");
33         return studentDAO.selectAllStudent();
34     }
35 }
```

Pada method diatas akan mengembalikan `selectAllStudent()` di `StudentDAO`.

Method `selectAllStudent()` pada `StudentDAOImpl.java`

```
1 package com.example.dao;
2
3 import java.util.Arrays;
4
16
17 @Service
18 public class StudentDAOImpl implements StudentDAO {
19
20     @Autowired
21     private RestTemplate restTemplate;
22
23     @Override
24     public StudentModel selectStudent(String npm) {
25         // TODO Auto-generated method stub
26         StudentModel student = restTemplate.getForObject("http://localhost/rest/student/view/"+npm, StudentModel.class);
27         return student;
28     }
29
30     @Override
31     public List<StudentModel> selectAllStudent() {
32         // TODO Auto-generated method stub
33         ResponseEntity<StudentModel[]> studentEntity = restTemplate.getForEntity("http://localhost/rest/student/viewall", StudentModel[].class);
34
35         return Arrays.asList(studentEntity.getBody());
36     }
37
38 }
```

Pada *method* diatas menggunakan `ResponEntity` untuk menangkap objek dalam bentuk list yang dihasilkan oleh *service producer*. Hasilnya akan dikembalikan dalam bentuk list dan ditampilkan pada URL seperti gambar dibawah ini.

All Data Students

Show 10 entriesSearch:

NO	NPM	Name	GPA	Cum Laude	Delete	Update
1.	1406623083	Dwi Arfi Faradina	3.75	Cum Laude!	Delete	Update
2.	1506737823	Kermie	3.32	Sangat Memuaskan!	Delete	Update
3.	1506737824	Ellene	2.89	Sangat Memuaskan!	Delete	Update
4.	1506737825	Delmer	2.58	Sangat Memuaskan!	Delete	Update
5.	1506737826	Kristian	2.33	Sangat Memuaskan!	Delete	Update
6.	1506737827	Marilyn	3.49	Cum Laude!	Delete	Update
7.	1506737828	Katti	3.73	Cum Laude!	Delete	Update