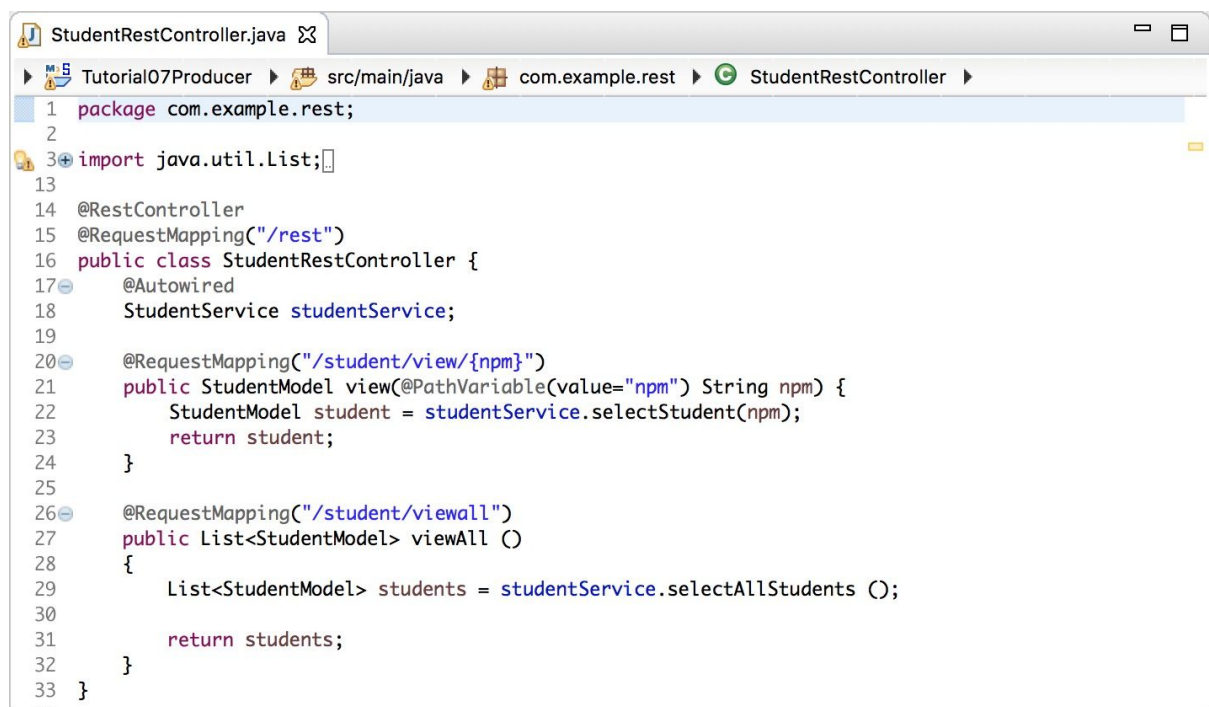
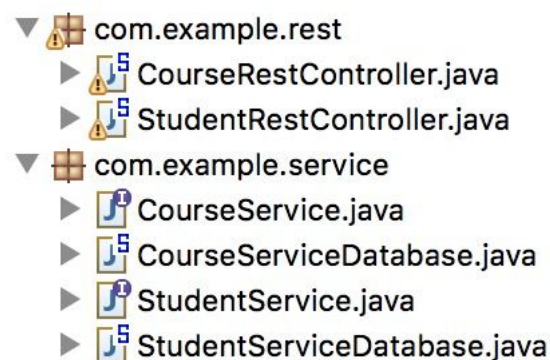


Yang dipelajari pada tutorial ini

Pada tutorial kali ini saya mempelajari bahwa front-end dan back-end dapat dipisah dengan adanya rest template yang disediakan oleh spring. Dengan dipisahannya front-end dan back-end, maka fokus pekerjaan menjadi lebih mudah. Saya juga mempelajari cara menggunakan request mapping di atas sebuah class. Saya juga harus mencari solusi dari error yang dihasilkan ketika menggunakan restTemplate di mana jika versi Spring Boot App saya di atas versi 1.3, saya harus menambahkan sebuah method restTemplate dan anotasi @Bean pada class yang saya sertakan restTemplate.

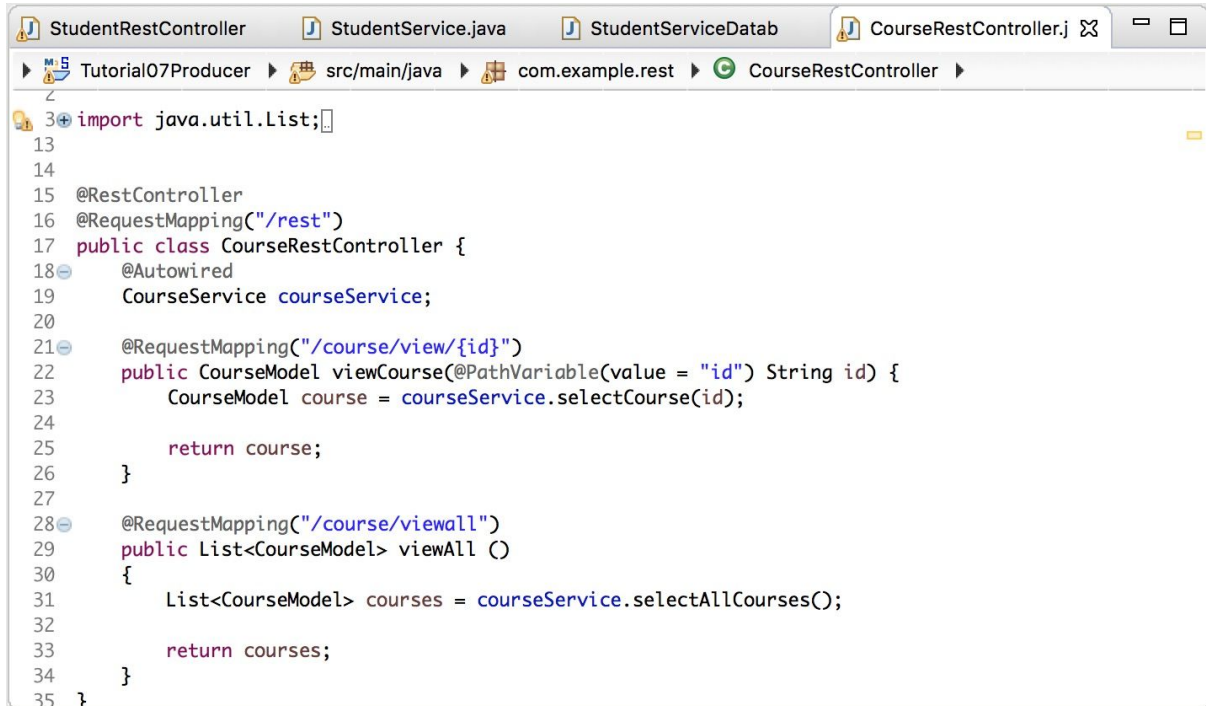
Latihan

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”.

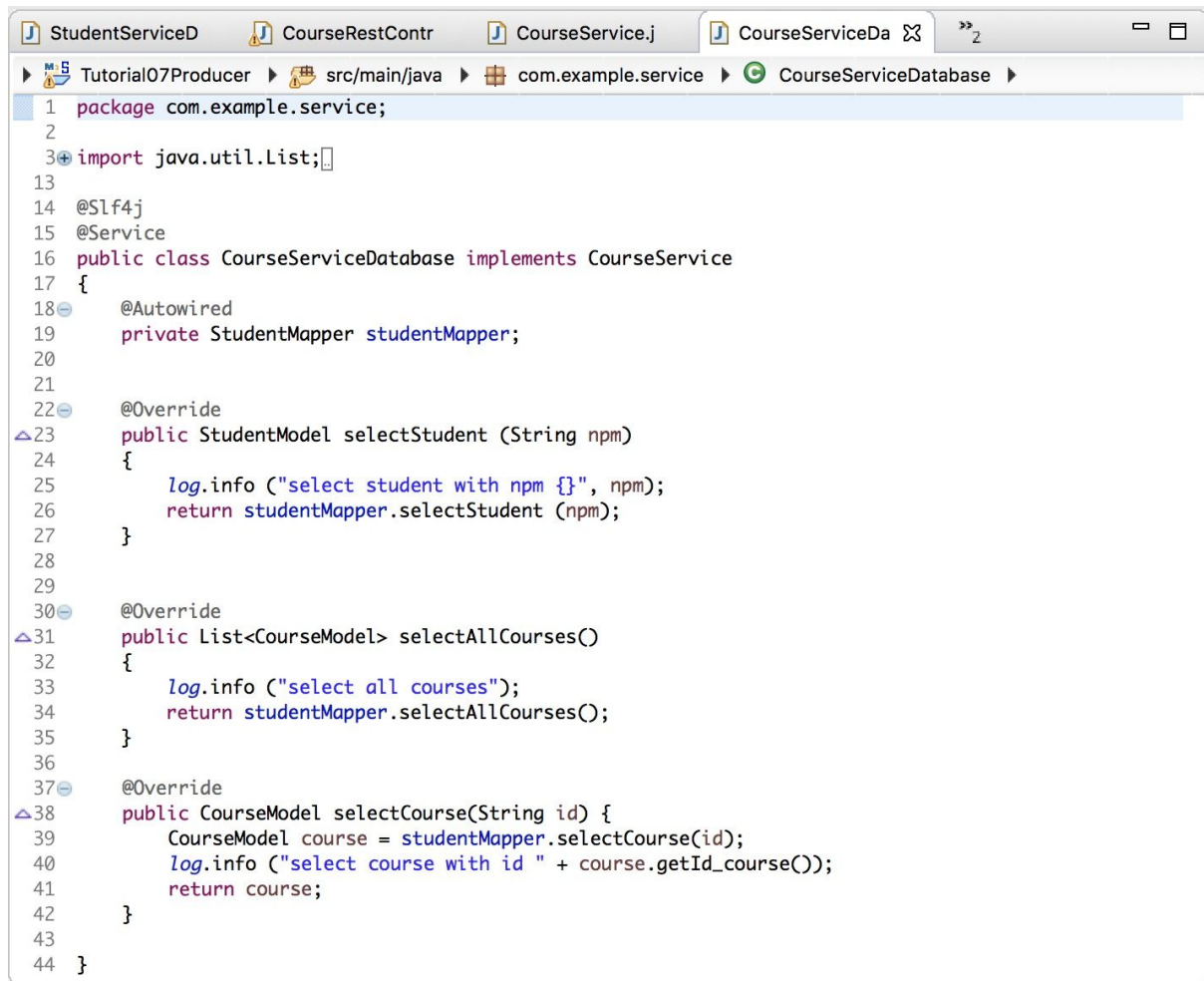


2. Buatlah service untuk class Course. Buatlah controller baru yang terdapat service untuk melihat suatu course dengan masukan ID Course (view by ID) dan service untuk melihat semua course (view all).

- ▼ com.example.rest
 - ▶ CourseRestController.java
 - ▶ StudentRestController.java
- ▼ com.example.service
 - ▶ CourseService.java
 - ▶ CourseServiceDatabase.java
 - ▶ StudentService.java
 - ▶ StudentServiceDatabase.java

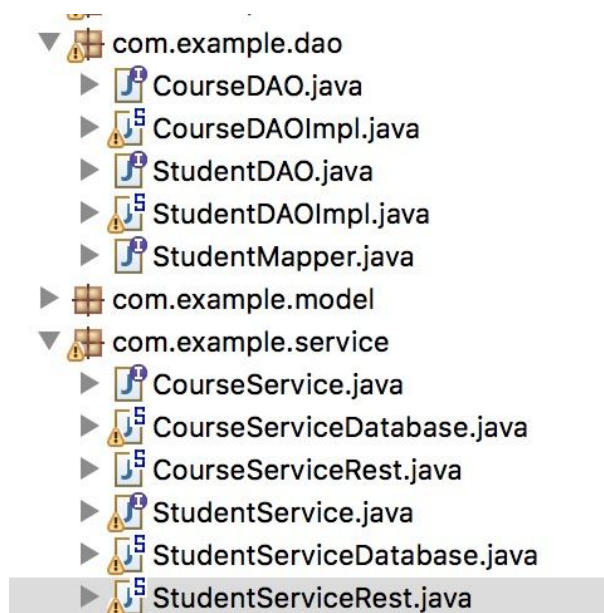


```
13 import java.util.List;
14
15 @RestController
16 @RequestMapping("/rest")
17 public class CourseRestController {
18     @Autowired
19     CourseService courseService;
20
21     @RequestMapping("/course/view/{id}")
22     public CourseModel viewCourse(@PathVariable(value = "id") String id) {
23         CourseModel course = courseService.selectCourse(id);
24
25         return course;
26     }
27
28     @RequestMapping("/course/viewall")
29     public List<CourseModel> viewAll ()
30     {
31         List<CourseModel> courses = courseService.selectAllCourses();
32
33         return courses;
34     }
35 }
```



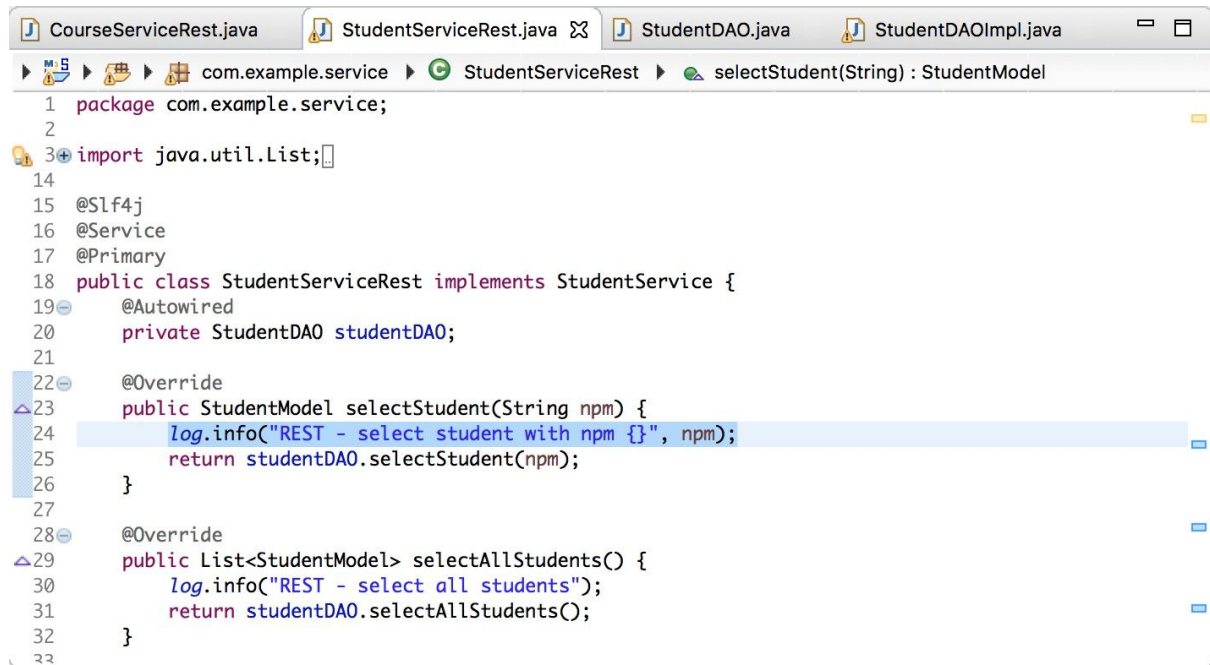
```
1 package com.example.service;
2
3 import java.util.List;
4
5 @Slf4j
6 @Service
7 public class CourseServiceDatabase implements CourseService
8 {
9     @Autowired
10     private StudentMapper studentMapper;
11
12     @Override
13     public StudentModel selectStudent (String npm)
14     {
15         log.info ("select student with npm {}", npm);
16         return studentMapper.selectStudent (npm);
17     }
18
19     @Override
20     public List<CourseModel> selectAllCourses()
21     {
22         log.info ("select all courses");
23         return studentMapper.selectAllCourses();
24     }
25
26     @Override
27     public CourseModel selectCourse(String id) {
28         CourseModel course = studentMapper.selectCourse(id);
29         log.info ("select course with id " + course.getId_course());
30         return course;
31     }
32 }
```

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



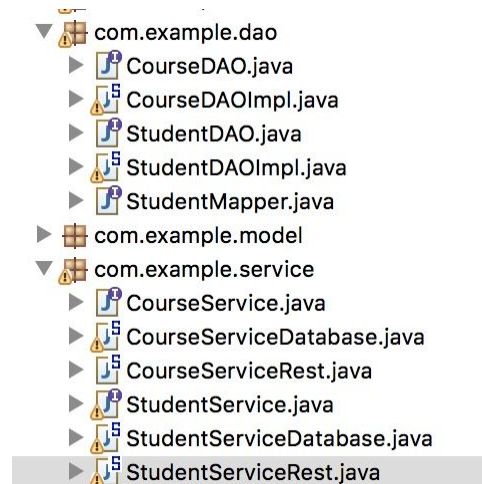
```
CourseServiceRest.java StudentServiceRest.java StudentDAO.java StudentDAOImpl.java
Tutorial07Consumer > src/main/java > com.example.dao > StudentDAO
1 package com.example.dao;
2
3 import java.util.List;
4
5
6
7 public interface StudentDAO {
8     StudentModel selectStudent(String npm);
9     List<StudentModel> selectAllStudents();
10 }
11
```

```
CourseServiceRest.java StudentServiceRest.java StudentDAO.java StudentDAOImpl.java
Tutorial07Consumer > src/main/java > com.example.dao > StudentDAOImpl
1 package com.example.dao;
2
3 import java.util.List;
4
5
6
7
8
9
10
11
12
13
14
15
16 @Slf4j
17 @Service
18 public class StudentDAOImpl implements StudentDAO {
19     @Autowired
20     @Lazy
21     private RestTemplate restTemplate;
22
23     @Bean
24     public RestTemplate restTemplate(RestTemplateBuilder builder) {
25         // Do any additional configuration here
26         return builder.build();
27     }
28
29     @Override
30     public StudentModel selectStudent(String npm) {
31         StudentModel student = restTemplate.getForObject(
32             "http://localhost:8080/rest/student/view/"+npm,
33             StudentModel.class);
34         return student;
35     }
36
37     @Override
38     public List<StudentModel> selectAllStudents() {
39         List<StudentModel> students = restTemplate.getForObject(
40             "http://localhost:8080/rest/student/viewall",
41             List.class);
42
43         return students;
44     }
45
46
```



```
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 }
```

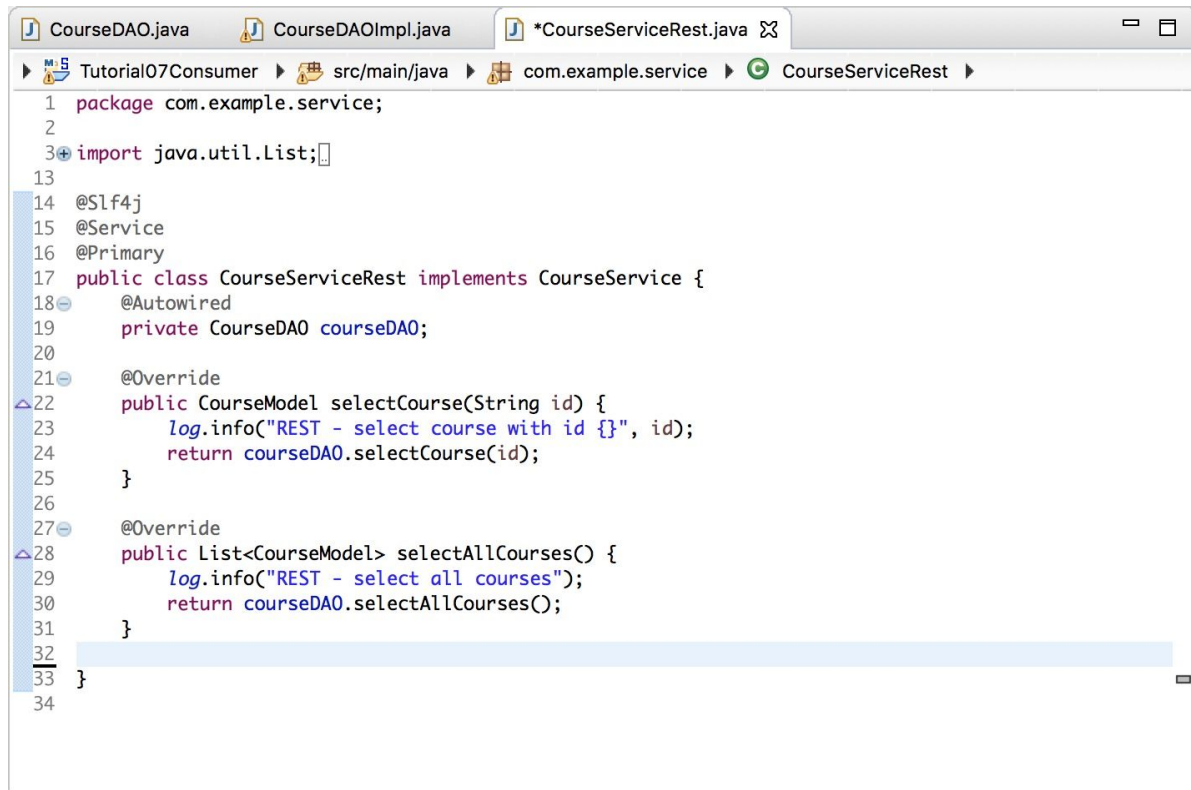
4. Implementasikan service consumer untuk class CourseModel dengan membuat class-class DAO dan service baru.




```
CourseDAO.java
Tutorial07Consumer > src/main/java > com.example.dao > CourseDAO
1 package com.example.dao;
2
3 import java.util.List;
4
5
6
7 public interface CourseDAO {
8     CourseModel selectCourse(String id);
9     List<CourseModel> selectAllCourses();
10 }
11
```

```
CourseDAO.java CourseDAOImpl.java
Tutorial07Consumer > src/main/java > com.example.dao > CourseDAOImpl > restTemplate(RestTemplateBuilder) : RestTemplate
1 import java.util.List;
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17 @Slf4j
18 @Service
19 public class CourseDAOImpl implements CourseDAO {
20     @Autowired
21     @Lazy
22     private RestTemplate restTemplate;
23
24     @Bean
25     public RestTemplate restTemplate(RestTemplateBuilder builder) {
26         // Do any additional configuration here
27         return builder.build();
28     }
29
30     @Override
31     public CourseModel selectCourse(String id) {
32         CourseModel course = restTemplate.getForObject(
33             "http://localhost:8080/rest/course/view/"+id,
34             CourseModel.class);
35         return course;
36     }
37
38     @Override
39     public List<CourseModel> selectAllCourses() {
40         List<CourseModel> courses = restTemplate.getForObject(
41             "http://localhost:8080/rest/course/viewall",
42             List.class);
43
44         return courses;
45     }
46
```

Teuku Amrullah Faisal
1506723231
APAP - C



```
1 package com.example.service;
2
3 import java.util.List;
4
5
6
7
8
9
10
11
12
13
14 @Slf4j
15 @Service
16 @Primary
17 public class CourseServiceRest implements CourseService {
18     @Autowired
19     private CourseDAO courseDAO;
20
21     @Override
22     public CourseModel selectCourse(String id) {
23         log.info("REST - select course with id {}", id);
24         return courseDAO.selectCourse(id);
25     }
26
27     @Override
28     public List<CourseModel> selectAllCourses() {
29         log.info("REST - select all courses");
30         return courseDAO.selectAllCourses();
31     }
32
33 }
34
```