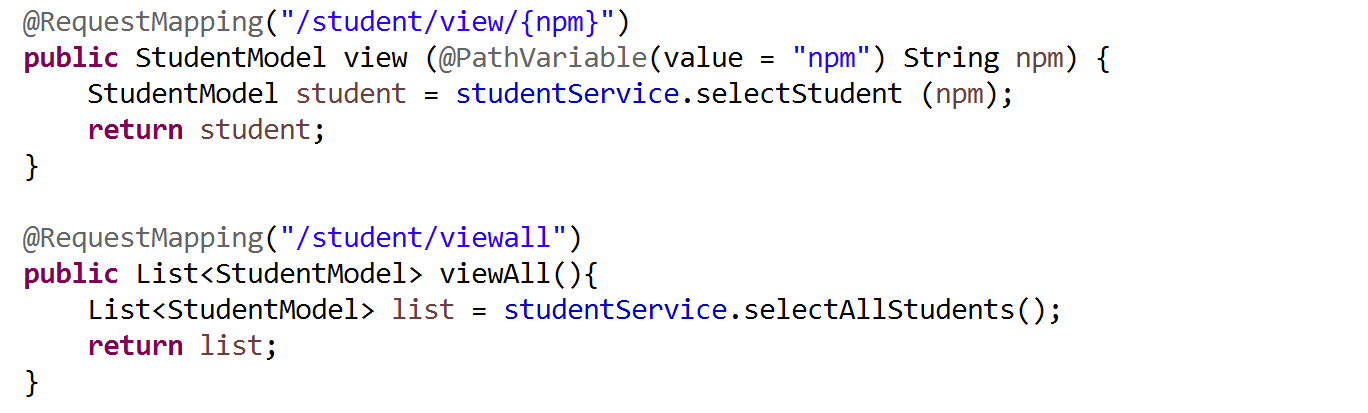
**Lesson Learned:**

Pada tutorial kali ini saya mempelajari penggunaan secara sederhana RESTful web service. ServiceProducer menyediakan web service yang dapat ‘dikonsumsi’ oleh service consumer. Resource yang diberiken oleh ServiceProducer direpresentasikan dalam bentuk JSON. Kemudian, ServiceConsumer akan menampilkan *resource* yang telah diperoleh dengan *view.* Oleh karena itu, dapat disimpulkan bahwa ServiceProducer adalah *layer backend* yang berinteraksi dengan *database* untuk meretrieve data (ServiceProducer tidak memiliki view), sedangkan ServiceConsumer bertugas untuk mengambil data yang diberikan ServiceProducer, menampilkannya, dan berinteraksi dengan pengguna.

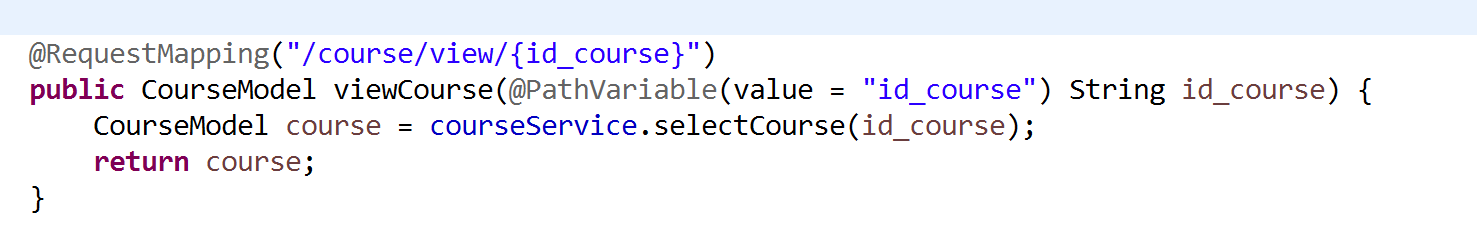
**Latihan 1:**

Pada latihan nomor 1, saya membuat *web service* dengan request mapping /rest/student/viewall pada class StudentRestController. Kemudian akan di retrieve data semua siswa yang ada di database, dengan bantuan interface StudentService yang akan mengeksekusi method selectStudent, yang akan mengekekusi method selectStudent yang ada di *class* StudentServiceDatabase (karena StudentServiceDatabase mengimplementaiskan interface StudentService). Kemudian, method selectStudent yang ada di class StudentServiceDatabase akan mengakses database dan meretrieve data yang diminta. Kemudian, data dikembalikan dalam bentuk JSON.

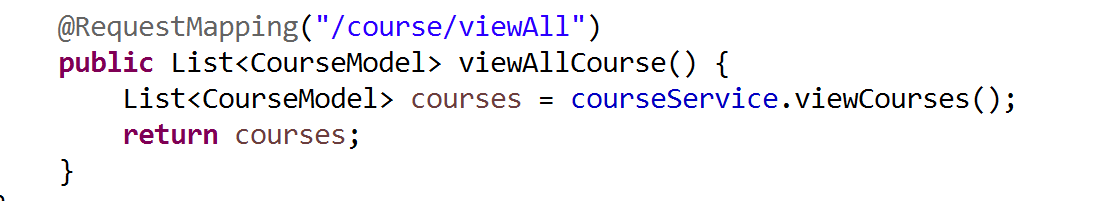


**Latihan 2:**

Pertama-tama saya membuat class CourseRestController. Lalu saya membuat method viewCourse yang akan mengembalikan object course dari database dengan bantuan interface CourseService yang memiliki method viewCourse dan class CourseServiceDatabase yang mengimplementasikan interface CourseService dan akan memanggil method viewCourse yang dimiliki oleh class StudentMapper. Method viewCourse yang dimiliki class StudentMapper akan mengeksekusi query yang akan mengembalikan course sesuai dengan id\_course yang di-input. Kemudian, object course dikembalikan dalam bentuk JSON oleh method viewCourse yang dimiliki oleh class CourseRestController.

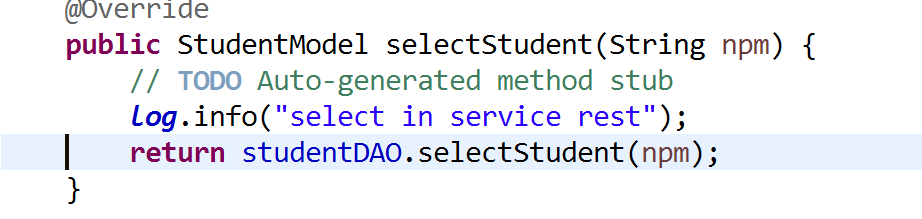


Kemudian saya juga membuat method viewAll dengan tahap-tahap yang sama, yaitu membuat method viewAll di interface CourseService, yang akan diimplementasikan oleh class CourseServiceDatabase. Di dalam class CourseServiceDatabase, method tersebut akan mengeksekusi method viewAll yang dimiliki oleh class StudentMapper, yang memiliki query yang akan mengembalikan semua course. Kemudian, semua course dikembalikan dalam bentuk JSON oleh class CourseRestController.



**Latihan 3**

Didalam class StudentServiceRest, terdapat method selectStudent yang akan mengeksekusi method selectStudent yang dimiliki oleh interface StudentDAO. Interface StudentDAO diimplementasikan oleh class StudentDAOImpl, sehingga method selectStudent yang ada di class StudentDAOImpl dieksekusi. Method ini akan me-retrieve data yang diberikan oleh producer dalam bentuk JSON, dan mengembalikannya dalam bentuk StudentModel. Data ini nantinya akan di-pass ke view oleh StudentController.



**Latihan 4**

Untuk memenuhi kebutuhan yang diminta soal, saya membuat interface CourseDAO dan class CourseDAOImpl. Interface CourseDAO memiliki method selectCourse dengan parameter id\_course, yang kemudian diimplementasikan oleh class CourseDAOImpl. Method ini akan me-retrieve data yang diberikan oleh producer dalam bentuk JSON, dan mengembalikannya dalam bentuk CourseModel. Data ini nantinya akan di-pass ke view oleh CourseController.

