

Write-Up Tutorial 7

Favian Kharisma Hazman - 1606886463

Pendahuluan

Pada tutorial kali ini dijelaskan tentang bagaimana membuat REST web service dengan framework spring boot dan database MySQL. Dengan pemisahan layer backend dan frontend sehingga layer backend independent terhadap user interface yang akan dibuat, aplikasi akan lebih mudah dikembangkan, memungkinkan pengembangan secara parallel dan skalabel.

Dengan controller yang langsung merender halaman, kita merubahnya menjadi endpoint masing masing yang melakukan operasi CRUD (create, read, update, delete) yang sudah didefinisikan sebelumnya.

Latihan

1. Buatlah FlightController menjadi RestController

Terdapat 5 method yang kita buat pada folder controller. Pattern atau cara pembuatannya sama seperti pada PilotController. Ada hal yang special dan berikut penanganannya.

- Kardinalitas Many-To-One yang Not Null
Karena model flight memiliki atribut pilot yang tidak bisa kosong, perlu anotasi untuk menambah pilot pada endpoint

POST ▾	{{url-tutorial7}}flight/add	Send ▾	Save ▾
--------	-----------------------------	--------	--------

Untuk mengidentifikasi nama dari atribut pilot tersebut yang merupakan sebuah objek, kita perlu memasukkan anotasi sebagai berikut

```
➤ @JsonIgnore
  public PilotModel getPilotFlight() {
      return pilotFlight;
  }
➤ @JsonProperty("pilotFlight")
  public void setPilotFlight(PilotModel pilotFlight) {
      this.pilotFlight = pilotFlight;
  } }
```

@JsonIgnore berfungsi untuk tidak mengambil pilot pada penerbangan tersebut pada view flight, sedangkan @JsonProperty berfungsi untuk attribute json yang menginisialisasi objek pilot pada json request body agar bisa diset dengan format json.

```
1 {
2     "flightNumber": "12345",
3     "origin": "Jakarta",
4     "destination": "Palembang",
5     "time": "2014-01-01",
6     "pilotFlight": { "id": 2,
7                     "licenseNumber": "1234" }
8 }
```

2. Men-consume web service dan membuat endpoint dari web service tersebut.

Pertama, API KEY didapatkan dengan cara mendaftarkan diri pada website vendor web service. Lalu, mendaftarkan initial URL dan API KEY dari web service tersebut pada package rest tepatnya class setting.

```
Setting.java  FlightControlle  PilotModel.java  FlightModel.jav  AirportControll  11
1 package com.apap.tutorial4.rest;
2
3 public class Setting {
4     final public static String pilotUrl = "https://995bed3e-f70f-49c7-90d3-9d4fabf4e70a.mock.pstmn.io";
5     final public static String airportUrl = "https://api.sandbox.amadeus.com/v1.2/airports/autocomplete";
6     final public static String airportApikey = "hb02VA0e8Kg72bgKHAZayxSkLGkSAfoc\n";
7 }
8
9
```

Dengan string path yang telah disimpan sebagai variable pada class setting, string tersebut dikonkatenasikan dengan request parameter yang dibutuhkan, yaitu keyword kota dan negara yang sudah tetap (ID)

```
27 @GetMapping(value= "/kota/{namaKota}")
28 public ResponseEntity<String> getStatus(@PathVariable("namaKota") String kota) {
29     String path = Setting.airportUrl+"?apikey="+Setting.airportApikey+"&term="+kota+"&country=ID";
30     // System.out.println(path);
31     ResponseEntity<String> response = restTemplate.getForEntity(path, String.class);
32     return response;
33 }
```

Dengan memanggil routing dan path variable yang kita buat, akan menampilkan response dari API tersebut

