

Bahy Helmi Hartoyo Putra
1606918124

Lesson Learned:

1. Membuat sebuah Rest Controller yang mana berfungsi untuk membuat aplikasi yang kita kembangkan dapat mengirimkan data ke luar/ke aplikasi lain.
2. Menggunakan Postman untuk melihat response data-data yang didapatkan dari tiap request yang kita buat. Ada 4 tipe yang digunakan pada tutorial ini, yaitu **GET**, **POST**, **PUT**, **DELETE**.
3. Untuk salah satu contoh **GET** di tutorial (View a Flight) yaitu akan mengembalikan sebuah FlightModel yang dikeluarkan dalam bentuk JSON dengan parameter request flightNumber-nya.
4. Untuk salah satu contoh **POST** di tutorial (Add a Flight) yaitu akan mengembalikan FlightModel juga dalam bentuk JSON.
5. Untuk salah satu contoh **PUT** di tutorial (Update a Flight) yaitu akan mengupdate flight bersangkutan (sesuai ID yang dimasukkan), mengembalikan sebuah String yang menyatakan sukses atau tidaknya data terupdate.
6. Untuk salah satu contoh **DELETE** di tutorial (Delete a Flight) yaitu akan menghapus flight dan mengembalikan sebuah String yang memberitahu sukses atau tidaknya deletion yang dilakukan.
7. Mock Server dibuat untuk menjadi *dummy* dari aplikasi produser yang bisa diakses oleh *user* dari luar, response-response dapat juga dibuat melalui Examples.

Latihan:

1. Membuat 5 Webservice baru di Postman untuk melakukan CRUD dari objek Flight. Setiap *output* kurang lebih sama seperti Webservice yang dibuat pada Pilot. Webservice sendiri merupakan sekumpulan fungsi program untuk melakukan pekerjaan tertentu yang dalam hal ini dapat melakukan manipulasi data, mengambil, menambahkan atau

menghapus.

```
@RestController
@RequestMapping("/flight")
public class FlightController {
    @Autowired
    private FlightService flightService;

    @PostMapping(value = "/add")
    public FlightModel addFlightSubmit(@RequestBody FlightModel flight) {
        return flightService.addFlight(flight);
    }

    @PutMapping(value = "/update/{flightId}")
    public String updateFlightSubmit(@PathVariable("flightId") long flightId,
        @RequestParam("destination") String destination,
        @RequestParam("origin") String origin,
        @RequestParam("time") Date time) {
        FlightModel flight = flightService.getFlightDetailById(flightId);
        if (flight.equals(null)) {
            return "Couldn't find your flight";
        }
        flight.setDestination(destination);
        flight.setOrigin(origin);
        flight.setTime(time);
        flightService.addFlight(flight);
        return "update";
    }

    @GetMapping(value = "/view/{flightNumber}")
    public FlightModel flightView(@PathVariable("flightNumber") String flightNumber) {
        FlightModel flight = flightService.getFlightDetailByFlightNumber(flightNumber).get();
        return flight;
    }

    @GetMapping(value = "/all")
    public List<FlightModel> flightViewAll() {
        return flightService.getAllFlight();
    }

    @DeleteMapping(value = "/{flightId}")
    public String deleteFlight(@PathVariable("flightId") long flightId) {
        FlightModel flight = flightService.getFlightDetailById(flightId);
        flightService.deleteFlight(flight);
        return "flight has been deleted";
    }
}
```

Add Flight Result:

► Add a Flight

POST ▼ `{{url-tutorial7}}/flight/add`

Params Authorization Headers (1) Body ● Pre-request Script Tests

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 433 ms Size: 373 B

Pretty Raw Preview JSON ▼

```

1 {
2   "id": 562,
3   "flightNumber": "1997",
4   "origin": "BYU - Bandar Udara Internasional Banyu, Depok",
5   "destination": "AMQ - Bandar Udara Internasional Pattimura, Ambon",
6   "time": "2018-11-01",
7   "pilot": {
8     "id": 1,
9     "licenseNumber": "1234",
10    "name": "Bahy",
11    "flyHour": 900
12  }
13 }
```

Update Flight Result:

► Update a Flight Ex

PUT ▼ `{{url-tutorial7}}/flight/update/2?destination=Bogor&origin=Depok&time=2018-11-01` Send ▼

Params ● Authorization Headers Body Pre-request Script Tests

KEY	VALUE	DESCRIPTION	**
<input checked="" type="checkbox"/> destination	Bogor		
<input checked="" type="checkbox"/> origin	Depok		
<input checked="" type="checkbox"/> time	2018-11-01		
Key	Value	Description	

Body Cookies Headers (3) Test Results Status: 200 OK Time: 10 ms Size: 121 B Save

Pretty Raw Preview Auto ▼

```

1 update|
```

Get Flight Result:


► Get a Flight

GET `{{url-tutorial7}}/flight/view/1151`

Params Authorization Headers Body Pre-request Script Tests

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 26 ms Size: 287 B

Pretty Raw Preview JSON 

```

1 {
2   "id": 2,
3   "flightNumber": "1151",
4   "origin": "Depok",
5   "destination": "Bogor",
6   "time": "2018-11-01",
7   "pilot": {
8     "id": 1,
9     "licenseNumber": "1234",
10    "name": "Bahy",
11    "flyHour": 900
12  }
13 }

```

Get All Flight Result:


► Get All Flight

GET `{{url-tutorial7}}/flight/all`

Params Authorization Headers Body Pre-request Script Tests

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 140 ms Size: 143.58 KB

Pretty Raw Preview JSON 

```

16 {
17   "id": 3,
18   "flightNumber": "1176",
19   "origin": "TIM - Bandar Udara Internasional Mozes Kilangin, Mimika",
20   "destination": "AMQ - Bandar Udara Internasional Pattimura, Ambon",
21   "time": "1986-10-17",
22   "pilot": {
23     "id": 1,
24     "licenseNumber": "1234",
25     "name": "Bahy",
26     "flyHour": 900
27   }
28 },
29 {
30   "id": 4,
31   "flightNumber": "1239",
32   "origin": "PNK - Bandar Udara Internasional Supadio, Pontianak",
33   "destination": "AMQ - Bandar Udara Internasional Pattimura, Ambon",
34   "time": "1986-10-17",
35   "pilot": {
36     "id": 1,
37     "licenseNumber": "1234",
38     "name": "Bahy",
39     "flyHour": 900
40   }
41 }

```

Delete Flight Result:

► Delete a Flight

DELETE `{{url-tutorial7}}/flight/561`

Params Authorization Headers Body Pre-request Script Tests

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 155 ms Size: 139 B

Pretty Raw Preview Auto

```
1 flight has been deleted
```

- Menggunakan API dari Amadeus Travel Innovation Sandbox untuk mengambil data Airport yang ada di kota yang dimasukkan oleh *user*. Cara pengerjaannya hampir sama seperti Get Status pada latihan, yaitu mengambil data menggunakan Service Consumer dan mengembalikan/menunjukkan data sebagai Service Producer. Yang dilakukan adalah mengambil url + apikey dari `api.sandbox.amadeus.com` lalu simpan menjadi variable (*airportUrl*) di `Setting.java`, lalu buat sebuah Rest Controller lagi bernama `AirportController`. Lalu dibuat sebuah `GetMapping` yang mengambil parameter *city* untuk di *pass* ke API sebagai *term*. Sehingga nantinya API akan mengembalikan semua Airport yang terdapat pada City yang kita input, dalam bentuk JSON.

URL API yang dimasukkan:

```
1 package com.apap.tutorial7.rest;
2
3 public class Setting {
4     final public static String pilotUrl = "https://53e6c667-c8c4-41ee-b31c-29ef8b2d6b56.mock.pstmn.io";
5     final public static String airportUrl = "https://api.sandbox.amadeus.com/v1.2/airports/autocomplete?apikey=0ZDTmaB1gyJxUg1xTW9nN9KCAPWnCdsH&term=";
6 }
7
```

Lakukan pemanggilan dan pengambilan data menggunakan API di Airport Controller:

```
1 package com.apap.tutorial6.controller;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @RestController
6 @RequestMapping("/airport")
7 public class AirportController {
8     @Autowired
9     RestTemplate restTemplate;
10
11     @Bean
12     public RestTemplate restAirport() {
13         return new RestTemplate();
14     }
15
16     @GetMapping(value =("/{city}")
17     public String getAirport(@PathVariable("city") String city) throws Exception {
18         String path = Setting.airportUrl + city + "&country=ID&all_airports=true";
19         return restTemplate.getForEntity(path, String.class).getBody();
20     }
21 }
22
```

Hasil pemanggilan GET melalui Postman (Kota: Makassar):

Get Airport by City Examples (0)

GET `{{url-tutorial7}}/airport/Makassar` Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies Code

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (3) Test Results Status: 200 OK Time: 806 ms Size: 232 B Save Download

Pretty Raw Preview JSON ≡

```
1 [
2   {
3     "value": "UPG",
4     "label": "Makassar - Sultan Hasanuddin International Airport [UPG]"
5   }
6 ]
```