

Tutorial 7

Syafiq Abdillah Umarghanis 1606890574

Yang saya pelajari

Pada tutorial ini, saya belajar tentang bagaimana caranya membuat web service dengan data yang sudah ada di tutorial sebelumnya.

Saya menggunakan aplikasi postman untuk membuat request.

Request yang bisa dilakukan adalah GET, POST, PUT, dan DELETE. Ilmu akan diterapkan pada latihan nomor 1

Kemudian, saya juga mempelajari tentang cara membuat mockup web service yang digunakan sebagai third party web service dimana ilmunya akan dipakai dalam pengerjaan latihan nomor 2

Latihan

RestController untuk Flight

a. POST add flight

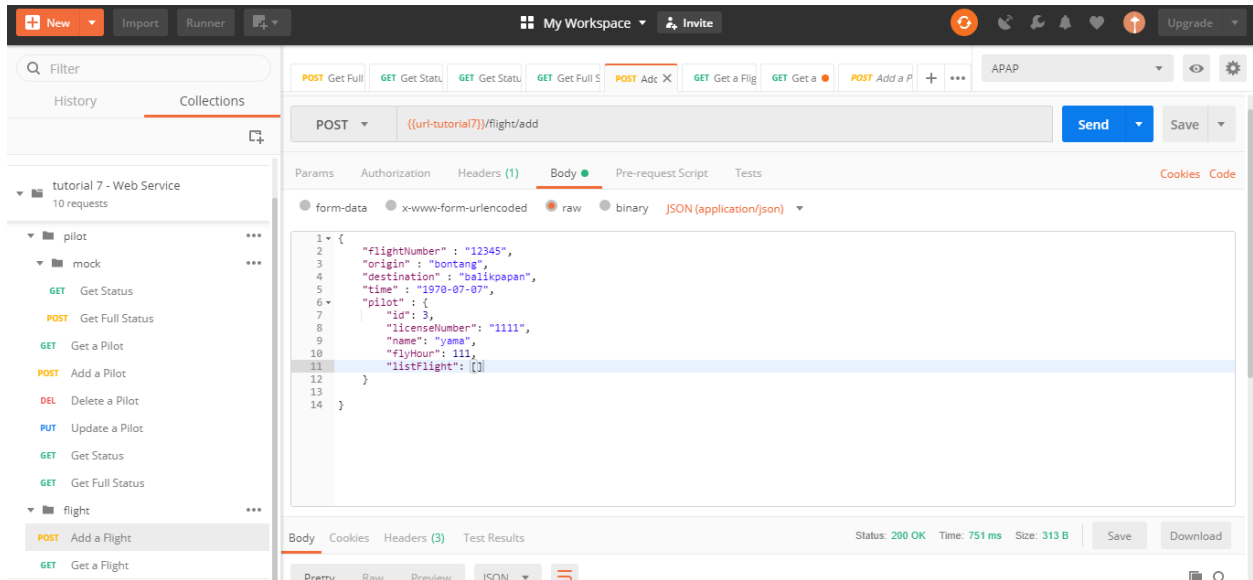
Di FlightController, saya membuat PostMapping seperti berikut yang menerima object FlightMode dan menyimpannya di database

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

    @Autowired
    private PilotService pilotService;

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

Di postman, saya mencoba menambahkan flight baru seperti berikut yang akan melakukan method POST dengan mapping “/flight/add” yang meminta webservice untuk menyimpan data di request body ke database. Parameter dari request ini saya buat di bagian body dalam bentuk raw dengan type JSON



Flight baru berhasil ditambahkan ke database

	id	destination	flight_number	origin	time	pilot_license_number
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	554	Internasional Sultan Hasanuddin...	1474	Internasional Husein Sastranega...	1974-04-04	1234
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	555	UPG - Bandar Udara Internasional Sultan Hasanuddin...	1521	AMQ - Bandar Udara Internasional Pattimura, Ambon	1974-04-04	1234
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	556	UPG - Bandar Udara Internasional Sultan Hasanuddin...	1533	KOE - Bandar Udara Internasional El Tari, Kupang	1977-08-16	1234
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	557	UPG - Bandar Udara Internasional Sultan Hasanuddin... Original length 57	1580	PDG - Bandar Udara Internasional Minangkabau, Pada...	1980-09-19	1234
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	558	UPG - Bandar Udara Internasional Sultan Hasanuddin...	1592	AMQ - Bandar Udara Internasional Pattimura, Ambon	1980-10-25	1234
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	559	UPG - Bandar Udara Internasional Sultan Hasanuddin...	1639	KOE - Bandar Udara Internasional El Tari, Kupang	1992-02-01	1234
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	560	UPG - Bandar Udara Internasional Sultan Hasanuddin...	1651	TNJ - Bandar Udara Internasional Raja Haji Fisabil	1992-02-01	1234
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	561	balikpapan	12345	bontang	1970-07-07	1111

b. PUT update flight

Di FlightController, saya membuat PutMapping seperti berikut untuk mengupdate data suatu flight dengan parameter yang diberikan. Untuk tipe data tiap parameter adalah `Optional<T>` karena param bisa jadi tidak diisi (bisa kosong)

```

@PutMapping(value = "/update/{flightID}")
public String updatePilotSubmit(@PathVariable("flightID") long flightID,
    @RequestParam("destination") Optional<String> destination,
    @RequestParam("origin") Optional<String> origin,
    @RequestParam("time") Optional<Date> time) {
    FlightModel flight = flightService.getFlightDetailById(flightID).get();
    if (flight.equals(null)) {
        return "Couldn't find your flight";
    }
    flight.setDestination(destination.get());
    flight.setOrigin(origin.get());
    flight.setTime(time.get());
    flightService.updateFlight(flightID, flight);
    return "update";
}

```

Di Postman, saya melakukan request PUT sebagai berikut yang meminta webservice untuk mengubah data flight dengan Id tertentu, menjadi flight dengan parameter yang telah ditentukan di params

Update a Flight

PUT `{{url-tutorial7}}/flight/update/562?destination=Lombok&origin=Bali&time=2012-01-01`

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> destination	Lombok	
<input checked="" type="checkbox"/> origin	Bali	
<input checked="" type="checkbox"/> time	2012-01-01	
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 470 ms

Pretty Raw Preview Auto

1 update

Di database, flight dummy yang saya buat dengan id 562 sudah terupdate seperti masukan yang saya berikan

<input type="checkbox"/>				560	UPG - Bandar Udara Internasional Sultan Hasanuddin...	1651	TNJ - Bandar Udara Internasional Raja Haji Fisabil...	1992-02-01	1234
<input type="checkbox"/>				561	balikpapan	12345	bontang	1970-07-07	1111
<input type="checkbox"/>				562	Lombok	9999	Bali	2012-01-01	1111

c. GET flight

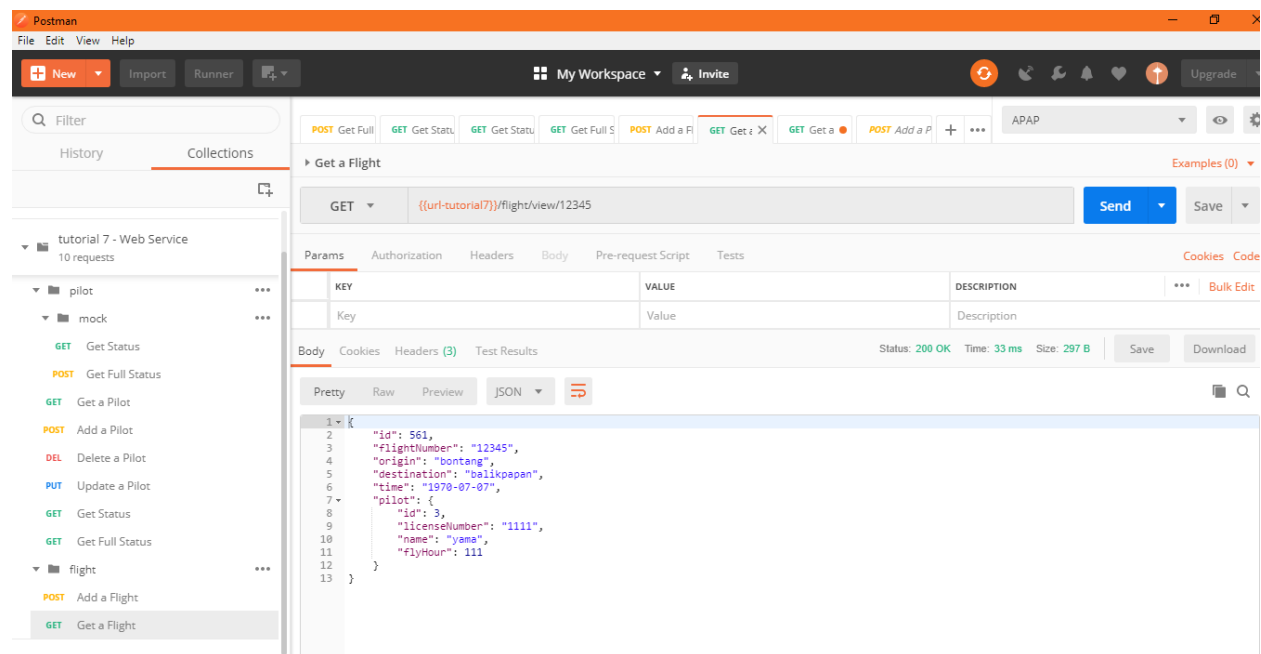
Di FlightModel, saya beri anotasi @JsonIgnore di atas hubungan Many-to-One antara flight dan pilot agar saat diview, tidak terjadi pemanggilan rekursif 'pilot memiliki flight yang memiliki pilot yang memiliki flight dst...'

```
@JsonIgnore
@OneToMany(mappedBy = "pilot", fetch = FetchType.LAZY)
private List<FlightModel> listFlight = new ArrayList<FlightModel>();
```

Di FlightController, saya membuat GetMapping yang meminta input flightNumber dari flight yang ingin ditampilkan. Output dari mthod ini adalah sebuah object FlightModel

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

Di postman, saya mencoba menampilkan flight yang saya tadi add dengan menggunakan request GET



d. GET all flight

Di FlightController, saya membuat GetMapping seperti berikut yang mengembalikan list of FlightModel

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

Di postman, saya membuat GET request sebagai berikut yang meminta list flight yang ada di database

Get All Flight Examples (

GET {{url-tutorial7}}/flight/all Send Save

Params Authorization Headers Body Pre-request Script Tests Cookies

KEY	VALUE	DESCRIPTION	...	Bull
Key	Value	Description		

Body Cookies Headers (3) Test Results Status: 200 OK Time: 1204 ms Size: 146.5 KB Save Downl

Pretty Raw Preview JSON ≡

```

1 [
2   {
3     "id": 1,
4     "flightNumber": "1131",
5     "origin": "BDO - Bandar Udara Internasional Husein Sastranegara, Bandung",
6     "destination": "AMQ - Bandar Udara Internasional Pattimura, Ambon",
7     "time": "1970-07-07",
8     "pilot": {
9       "id": 1,
10      "licenseNumber": "1234",
11      "name": "Sawadikap",
12      "flyHour": 900
13    }
14  },
15  {
16    "id": 2,
17    "flightNumber": "1151",
18    "origin": "SRG - Bandar Udara Internasional Achmad Yani, Semarang",
19    "destination": "AMQ - Bandar Udara Internasional Pattimura, Ambon",

```

e. DELETE flight

<code>{{url}}/flight/all</code>	
Delete flight	
<code>{{url}}/flight/{{flightID}}</code>	respon

Sebenarnya url mapping yang didefinisikan oleh soal **tidak menggambarkan bahwa akan terjadi suatu delete flight**. Namun karena begitu spesifikasinya, saya hanya mengikuti saja seperti yang diminta.

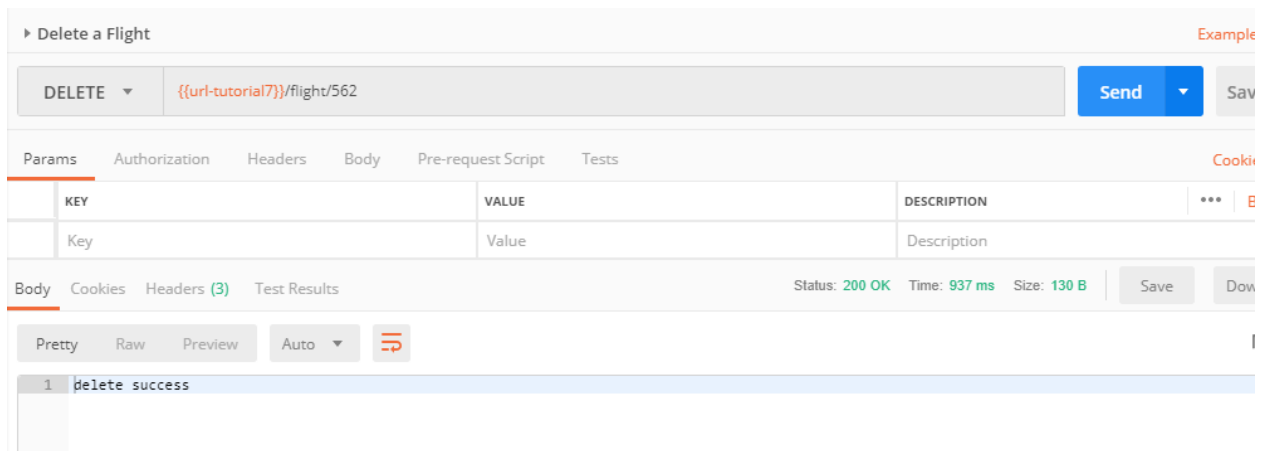
Di FlightController, saya membuat DeleteMapping seperti berikut yang akan menghapus flight yang idnya tercantum di path variable. Keluaran dari mthod ini adalah message "delete success" yang ditampilkan bila delete berhasil .

```

@DeleteMapping(value = "/{flightID}")
public String deleteFlight(@PathVariable("flightID") long flightID) {
    FlightModel flight = flightService.getFlightDetailById(flightID).get();
    flightService.deleteFlight(flight);
    return "delete success";
}

```

Di Postman, saya mencoba menghapus flight dengan ID 562 yang tadi saya buat, keluar notifikasi berhasil



Terlihat di DB bahwa flight dengan id 562 sudah tidak ada

<input type="checkbox"/>	Edit	Copy	Delete	560	UPG - Bandar Udara Internasional Sultan Hasanuddin...	1651	TNJ - Bandar Udara Internasional Raja Haji Fisabil...	1992-02-01	1234
<input type="checkbox"/>	Edit	Copy	Delete	561	balikpapan	12345	bontang	1970-07-07	1111

☐ Check all With selected: Edit Copy Delete Export

2. Mempelajari API <https://sandbox.amadeus.com/travel-innovation-sandbox/apis/get/airports/autocomplete>

- Saya telah mempelajari cara mengirimkan request dengan api tersebut.
- Saya membuat akun dengan menghubungkan dengan akun github, dan membuat app untuk mendapatkan APIkey
- Setelah mendapat APIkey, saya membuat Environment baru di postman dengan nama Amadeus dan membuat variable amadeus dengan nilai <https://api.sandbox.amadeus.com/v1.2/airports/autocomplete>
- Saya membuat sebuah Service Provider yang meminta data dari API amadeus dan menampilkannya
- Saya membuat Controller baru bernama AirportController yang isinya sebagai berikut. Saya menggunakan contoh saat membuat mockup.
- Method getAirport membutuhkan input namaKota yang bandaranya ingin dicari. Output dari method ini adalah list dari object airport yang diambil dari API amadeus

```

1 package com.apap.tutorial7.controller;
2 import org.springframework.web.bind.annotation.GetMapping;
3 import org.springframework.web.bind.annotation.PathVariable;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 import org.springframework.web.bind.annotation.RestController;
6 import org.springframework.web.client.RestTemplate;
7
8 import com.apap.tutorial7.rest.Setting;
9
10 @RestController
11 @RequestMapping("/airport")
12 public class AirportController {
13
14     @GetMapping(value="/{namaKota}")
15     public Object getAirport(@PathVariable("namaKota") String namaKota) throws Exception {
16         RestTemplate restTemplate = new RestTemplate();
17         String path = Setting.airportUrl + "?apikey=LK9M9TU1DZuztnXZW6Gkuj8wN0tZVQmz&term=" + namaKota + "&country=ID";
18         Object obj = restTemplate.getForObject(path, Object.class);
19         return obj;
20     }
21 }
22
23

```

- Kemudian di postman, saya membuat request GET yang menuju webservice yang saya buat untuk mendapatkan list airport sebagai berikut. Dan muncul list bandara yang dikehendaki

The screenshot shows the Postman interface with a GET request to the endpoint `{{url-tutorial7}}/airport/`. The response is a JSON array of airport data for Jakarta.

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body (JSON):

```

1 [
2   {
3     "value": "JKT",
4     "label": "Jakarta [JKT]"
5   },
6   {
7     "value": "CGK",
8     "label": "Jakarta - Soekarno - Hatta International Airport [CGK]"
9   },
10  {
11    "value": "SUB",
12    "label": "Surabaya - Juanda International Airport [SUB]"
13  },
14  {
15    "value": "HLP",
16    "label": "Jakarta - Halim Perdanakusuma Airport [HLP]"
17  },
18  {
19    "value": "YOG"

```

~Selesai~