

Latihan

1. Buatlah FlightController menjadi RestController sesuai spesifikasi sbb.:

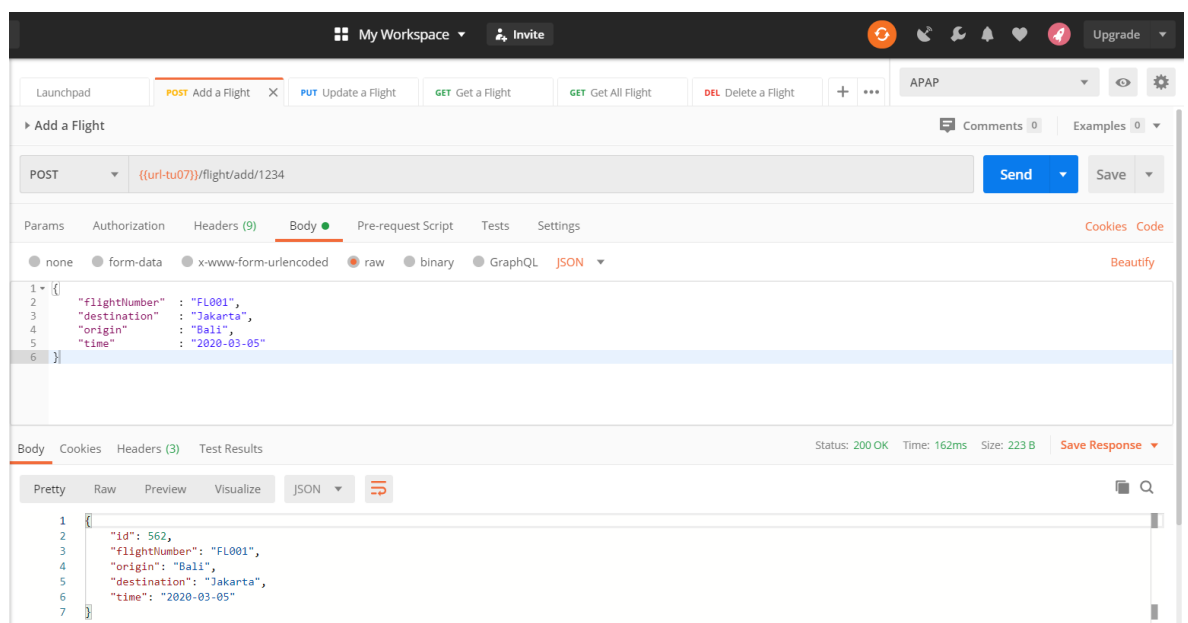
POST add flight <code>{url}/flight/add</code>	request body: new FlightModel response : new FlightModel
PUT update flight <code>{url}/flight/update/{flightID}?</code> <code>destination={destination}&origin={origin}</code> <code>&time={time}</code> parameter dapat tidak harus lengkap	response body: "flight update success"
GET flight <code>{url}/flight/view/{flightNumber}</code>	response : FlightModel
GET all flight <code>{url}/flight/all</code>	response : List<FlightModel>
Delete flight <code>{url}/flight/{flightID}</code>	response: "flight has been deleted"

Tambahkan pada Postman *request* dan save pada folder "Flight" di Collection "Tutorial-07_Web-Service"

a. Add a Flight

POST add flight <code>{url}/flight/add</code>	request body: new FlightModel response : new FlightModel
---	---

- Postman



- Database

Show query box

✓ Showing rows 0 - 0 (1 total, Query took 0.0005 seconds.)

```
SELECT * FROM `flight` WHERE id=562
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	id	destination	flight_number	origin	time	pilot_license_number
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	562	Jakarta	FL001	Bali	2020-03-05	1234

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table

b. Update a Flight

PUT update flight <code>{url}/flight/update/{flightID}?</code> <code>destination={destination}&origin={origin}&time={time}</code> parameter dapat tidak harus lengkap	response body: "flight update success"
---	--

- Postman

My Workspace | Invite | Upgrade

Launchpad | POST Add a Flight | PUT Update a Flight X | GET Get a Flight | GET Get All Flight | DEL Delete a Flight | + | ... | APAP

Update a Flight | Comments 0 | Examples 0

PUT | `{{url-tu07}}/flight/update/FL001?destination=Lampung&origin=Bali&time=2020-05-09` | Send | Save

Params | Authorization | Headers (8) | Body | Pre-request Script | Tests | Settings | Cookies | Code

Query Params

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> destination	Lampung	
<input checked="" type="checkbox"/> origin	Bali	
<input checked="" type="checkbox"/> time	2020-05-09	
Key	Value	Description

Body | Cookies | Headers (3) | Test Results | Status: 200 OK | Time: 41ms | Size: 137 B | Save Response

Pretty | Raw | Preview | Visualize | Text |

1 flight update success

- Database

Show query box

✓ Showing rows 0 - 0 (1 total, Query took 0.0006 seconds.)

```
SELECT * FROM `flight` WHERE id=562
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

	id	destination	flight_number	origin	time	pilot_license_number
<input type="checkbox"/>	562	Lampung	FL001	Bali	2020-05-09	1234

↑ ☐ Check all With selected: Edit Copy Delete Export

c. Get a Flight

GET flight <code>{url}/flight/view/{flightNumber}</code>	response : FlightModel
--	------------------------

- Postman

My Workspace Invite Upgrade

Launchpad POST Add a Flight PUT Update a Flight GET Get a Flight GET Get All Flight DEL Delete a Flight APAP

Get a Flight Comments 0 Examples 0

GET {{url-tu07}}/flight/view/FL001 Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 19ms Size: 223 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 562,
3   "flightNumber": "FL001",
4   "origin": "Bali",
5   "destination": "Lampung",
6   "time": "2020-05-09"
7 }
```

d. Get All Flight

GET all flight <i>{url}/flight/all</i>	response : List<FlightModel>
--	------------------------------

- Postman

Postman interface showing a GET request to `{{url-tu07}}/flight/all`. The response is a JSON array of 3 flight objects.

```
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   },
9   {
10    "id": 2,
11    "flightNumber": "1151",
12    "origin": "SRG - Bandar Udara Internasional Achmad Yani, Semarang",
13    "destination": "AMQ - Bandar Udara Internasional Pattimura, Ambon",
14    "time": "1961-11-10"
15  },
16  {
17    "id": 3,
18    "flightNumber": "1176",
19    "origin": "TIM - Bandar Udara Internasional Mozes Kilangin, Minika",
20    "destination": "AMQ - Bandar Udara Internasional Pattimura, Ambon",
```

e. Delete a Flight

Delete flight <i>{url}/flight/{flightID}</i>	response: "flight has been deleted"
--	-------------------------------------

- Postman

Postman interface showing a DELETE request to `{{url-tu07}}/flight/delete/FL001`. The response is the text "flight has been deleted".

```
1 flight has been deleted
```

- Database

Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)

```
SELECT * FROM `flight` WHERE id=562
```

2. Coba pelajari dokumentasi API yang ada pada link berikut: <https://developers.amadeus.com/> Dengan *parameter* pasti, bahwa *airport* yang dicari hanya yang ada di Indonesia, Buatlah sebuah *service producer*. Jika dipanggil, akan mengembalikan daftar *airport* yang ada di suatu kota di Indonesia. Masukan dari API tersebut hanyalah nama kota. Jangan lupa masukkan *request* untuk mengakses data tersebut pada Postman.

Hint 1: program yang Anda buat akan menggunakan *service consumer* untuk mendapatkan data dari *third party* lalu mengirimkannya lagi sebagai *service producer*.

Hint 2: Seringkali Anda perlu membuat akun terlebih dahulu untuk mendapatkan API Key.

Hint 3: Anda dapat menggunakan API dari provider lain selama sesuai dengan kebutuhan

- Airport Controller

```
import com.apap.tu07.rest.Setting;

@RestController
@RequestMapping("/airport")
public class AirportController {

    @Autowired
    private RestTemplate restTemplate;

    @Bean
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }

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

- Setting. Java

```
1 package com.apap.tu07.rest;|
2
3 public class Setting {
4     final public static String pilotUrl = "https://8b4459fc-3d49-4dc6-b9e5-f2d69c6ca67e.mock.pstmn.io";
5     final public static String airportUrl = "https://api.sandbox.amadeus.com/v1.2/airports/autocomplete";
6     final public static String airportApikey = "hBO2VAOe8Kg72bgKHAZayxSkLGkSAfoc\n";
7 }
```

- Postman

SAVE REQUEST

Requests in Postman are saved in collections (a group of requests).
[Learn more about creating collections](#)

Request name

Request description (Optional)

Make things easier for your teammates with a complete request description.

Descriptions support [Markdown](#)

Select a collection or folder to save to:

Q Search for a collection or folder

Tutorial-07_Mock-Server + Create Folder

GET Get a Status

Cancel

Save to Tutorial-07_Mock-Server