

## **Tutorial 6 - Unit Testing dan Load Testing Web App Sederhana**

### **Latihan :**

1. **Mengapa perlu menginisiasi object pilotModel, sedangkan yang di test hanya FlightModel ?**

Karena pada object FlightModel, terdapat atribut yang berisi objek PilotModel. Untuk dapat mengisi atribut tersebut, dibutuhkan suatu objek PilotModel. Jadi PilotModel juga harus diinisiasi.

2. **Jelaskan apa yang akan terjadi jika object PilotModel dihapus dan tidak dilakukan setPilot pada FlightModel ?**

Maka nilai Pilot pada FlightModel akan null dan tidak bisa dibuat. Karena FlightModel dan PilotModel mempunyai hubungan Many-To-One dimana Flight pasti akan memiliki sebuah Pilot. Jadi kalau tidak ada objek pilot, maka objek tersebut tidak bisa dibuat.

Seperti berikut (error yang tertulis jika PilotModel tidak diinisiasi dan tidak dilakukan setPilot)

```
Hibernate: insert into pilot (fly_hour, license_number, name) values (?, ?, ?)
Hibernate: insert into flight (destination, flight_number, origin, pilot_license_number, time) values (?, ?, ?, ?, ?)
```

3. **Jelaskan apa yang dilakukan oleh code**

```
Mockito.when(flightDb.findByFlightNumber(flight.get().getFlightNumber())).thenReturn(flight);
```

Mockito digunakan untuk mock interface sehingga fungsi dummy bisa ditambahkan ke mock interface yang digunakan dalam unit testing.

Method diatas adalah sebuah mock implementation dari method findByFlightNumber() di FlightDb. Yang mana akan mengembalikan nilai flight (yang telah dipilih inisiasi di baris sebelumnya) jika flightNumber nya ditemukan. Atau dengan kata lain menstimulasi sebuah method di FlightDb, dalam hal ini method findByFlightNumber() untuk mendapatkan sebuah nilai yang ingin di return (dalam hal ini objek flight)

4. **Jelaskan apa yang dilakukan oleh code**

```
Mockito.when(flightService.getFlightDetailByFlightNumber(flight.get().getFlightNumber())).thenReturn(flight);
```

Mockito digunakan untuk mock interface sehingga fungsi dummy bisa ditambahkan ke mock interface yang digunakan dalam unit testing.

Method diatas adalah sebuah mock implementation dari method getFlightDetailByFlightNumber() di interface FlightService. Yang mana akan mengembalikan nilai flight (yang telah dipilih inisiasi di baris sebelumnya) jika flightNumber nya ditemukan. Atau dengan kata lain menstimulasi sebuah method di FlightService, dalam hal ini method getFlightDetailByFlightNumber() untuk mendapatkan sebuah nilai yang ingin di return (dalam hal ini objek flight)

5. **Jelaskan apa yang di test oleh code**

```
.andExpect(MockMvcResultMatchers.status().isOk())
```

Melakukan verifikasi bahwa http response status adalah OK (200). Untuk memastikan request telah dieksekusi dengan sukses.

6. **Jelaskan apa yang di test oleh code**

```
.andExpect(MockMvcResultMatchers.jsonPath("$.flightNumber", Matchers.is(flight.get().getFlightNumber())));
```

Melakukan verifikasi bahwa respons yang diberikan sesuai dengan flightNumber yang menjadi parameter Matcher.is.

Di sini digunakan jsonPath yang melakukan ekstraksi konten respons dan memberikan nilai yang diminta.

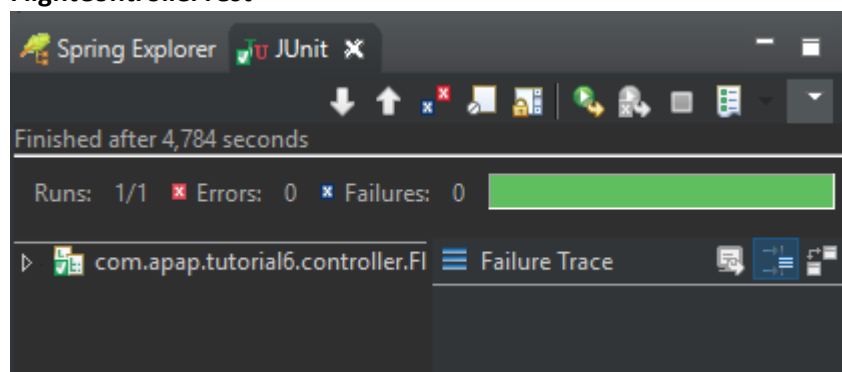
#### 7. Jelaskan anotasi `@ResponseBody` yang ada pada route `"/flight/view"`

```
@RequestMapping(value = "/flight/view", method = RequestMethod.GET)
private @ResponseBody FlightModel view(@RequestParam(value = "flightNumber") String flightNumber, Model model) {
    FlightModel archive = flightService.getFlightDetailByFlightNumber(flightNumber).get();
    return archive;
}
```

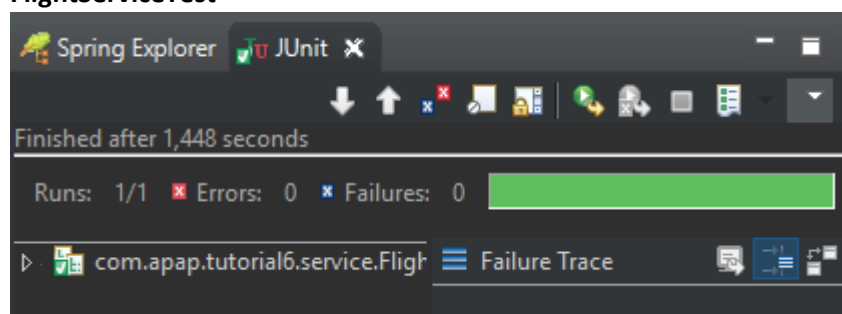
Jika memakai anotasi `ResponseBody`, maka spring akan mencoba mengkonversikan return value dari method tersebut dan menuliskannya secara langsung (secara otomatis) pada http response. Dapat diperhatikan dari nilai return yang bukan merupakan string seperti method pada controller yang biasa dipakai.

#### Run kelas test :

##### 1. FlightControllerTest

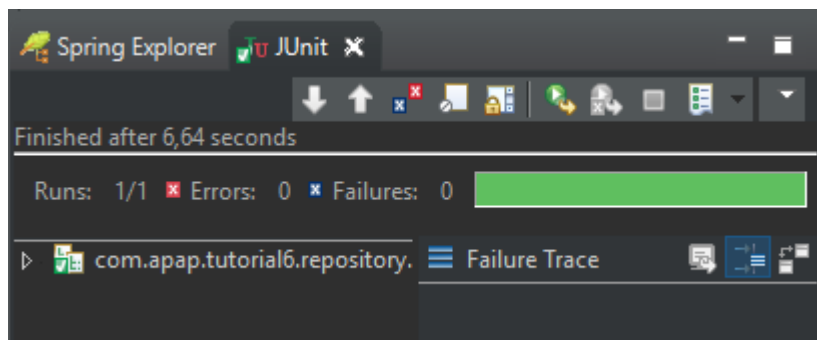


##### 2. FlightServiceTest



##### 3. FlightDbTest

RAFIQA AMINI MULIA  
1606917834  
APAP – C



## Hasil Load Testing :

### 1. Tabel

View Results in Table										
Name: View Results in Table										
Comments:										
Write results to file / Read from file										
Filename										
		Browse...		Log/Display Only:		<input type="checkbox"/> Errors <input type="checkbox"/> Successes		Configure		
Sample #	Start Time	Thread Name	Label	Sample Time(m.)	Status	Bytes	Sent Bytes	Latency	Connect Time...	
1	17:21:57.598	Thread Group 1...	HTTP Request	8808	✓	517182	147	121		2
2	17:21:57.998	Thread Group 1...	HTTP Request	8986	✓	517182	147	125		1
3	17:21:57.797	Thread Group 1...	HTTP Request	9526	✓	517182	147	77		1
4	17:21:58.403	Thread Group 1...	HTTP Request	9435	✓	517182	147	160		1
5	17:21:58.617	Thread Group 1...	HTTP Request	9802	✓	517182	147	181		1
6	17:21:58.818	Thread Group 1...	HTTP Request	9976	✓	517182	147	447		1
7	17:21:57.499	Thread Group 1...	HTTP Request	11420	✓	517182	147	216		37
8	17:21:59.017	Thread Group 1...	HTTP Request	10596	✓	517182	147	281		1
9	17:21:58.197	Thread Group 1...	HTTP Request	12002	✓	517182	147	214		1
10	17:21:59.208	Thread Group 1...	HTTP Request	13268	✓	517182	147	525		1
11	17:21:59.408	Thread Group 1...	HTTP Request	14417	✓	517182	147	7121		1
12	17:21:59.611	Thread Group 1...	HTTP Request	16693	✓	517182	147	7509		1
13	17:22:00.607	Thread Group 1...	HTTP Request	17226	✓	517182	147	8442		1
14	17:22:00.017	Thread Group 1...	HTTP Request	18622	✓	517182	147	8338		1
15	17:22:00.210	Thread Group 1...	HTTP Request	18750	✓	517182	147	8821		1
16	17:21:59.818	Thread Group 1...	HTTP Request	19409	✓	517182	147	8104		1
17	17:22:00.411	Thread Group 1...	HTTP Request	19529	✓	517182	147	8496		0
18	17:22:01.007	Thread Group 1...	HTTP Request	19533	✓	517182	147	9460		1
19	17:22:00.808	Thread Group 1...	HTTP Request	20774	✓	517182	147	9028		1
20	17:22:01.208	Thread Group 1...	HTTP Request	21186	✓	517182	147	11657		1
21	17:22:01.408	Thread Group 1...	HTTP Request	22893	✓	517182	147	12612		1
22	17:22:01.808	Thread Group 1...	HTTP Request	26371	✓	517182	147	16141		1
23	17:22:01.619	Thread Group 1...	HTTP Request	27060	✓	517182	147	14832		0
24	17:22:02.223	Thread Group 1...	HTTP Request	26852	✓	517182	147	16886		1
25	17:22:02.423	Thread Group 1...	HTTP Request	27819	✓	517182	147	16948		1
26	17:22:02.823	Thread Group 1...	HTTP Request	27770	✓	517182	147	17871		1
27	17:22:02.018	Thread Group 1...	HTTP Request	28945	✓	517182	147	16754		1
28	17:22:02.623	Thread Group 1...	HTTP Request	29173	✓	517182	147	17720		1
29	17:22:02.823	Thread Group 1...	HTTP Request	30040	✓	517182	147	18662		1

RAFIQA AMINI MULIA  
1606917834  
APAP – C

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename: Browse... Log/Display Only: ☐ Errors ☒ Successes ☐ Configure

Sample #	Start Time	Thread Name	Label	Sample Time(m...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
23	17:22:01.619	Thread Group 1...	HTTP Request	27060	✓	517182	147	14832	0
24	17:22:02.223	Thread Group 1...	HTTP Request	26852	✓	517182	147	16886	1
25	17:22:02.423	Thread Group 1...	HTTP Request	27819	✓	517182	147	16948	1
26	17:22:02.823	Thread Group 1...	HTTP Request	27770	✓	517182	147	17871	1
27	17:22:02.018	Thread Group 1...	HTTP Request	28945	✓	517182	147	16754	1
28	17:22:02.623	Thread Group 1...	HTTP Request	29173	✓	517182	147	17720	1
29	17:22:03.022	Thread Group 1...	HTTP Request	29049	✓	517182	147	18662	1
30	17:22:03.621	Thread Group 1...	HTTP Request	28870	✓	517182	147	18966	1
31	17:22:03.822	Thread Group 1...	HTTP Request	31299	✓	517182	147	20968	1
32	17:22:06.421	Thread Group 1...	HTTP Request	30419	✗	9307	147	30418	1
33	17:22:05.821	Thread Group 1...	HTTP Request	31021	✗	9307	147	31020	1
34	17:22:06.620	Thread Group 1...	HTTP Request	30224	✗	9307	147	30219	1
35	17:22:06.222	Thread Group 1...	HTTP Request	30624	✗	9307	147	30623	0
36	17:22:05.622	Thread Group 1...	HTTP Request	31226	✗	9307	147	31224	1
37	17:22:06.020	Thread Group 1...	HTTP Request	30828	✗	9307	147	30828	1
38	17:22:03.421	Thread Group 1...	HTTP Request	34679	✓	517182	147	25682	1
39	17:22:03.223	Thread Group 1...	HTTP Request	36219	✓	517182	147	25313	1
40	17:22:04.022	Thread Group 1...	HTTP Request	36608	✓	517182	147	25307	1
41	17:22:05.022	Thread Group 1...	HTTP Request	35635	✓	517182	147	27184	1
42	17:22:04.421	Thread Group 1...	HTTP Request	36783	✓	517182	147	27649	1
43	17:22:04.822	Thread Group 1...	HTTP Request	36500	✓	517182	147	26177	0
44	17:22:04.622	Thread Group 1...	HTTP Request	36944	✓	517182	147	26752	1
45	17:22:04.222	Thread Group 1...	HTTP Request	37445	✓	517182	147	26501	0
46	17:22:05.221	Thread Group 1...	HTTP Request	36947	✓	517182	147	27620	0
47	17:22:07.024	Thread Group 1...	HTTP Request	35170	✗	9307	147	35170	1
48	17:22:06.820	Thread Group 1...	HTTP Request	35385	✗	9307	147	35384	0
49	17:22:05.421	Thread Group 1...	HTTP Request	37888	✓	517182	147	29838	1
50	17:22:07.220	Thread Group 1...	HTTP Request	36724	✓	517182	147	31381	1

☐ Scroll automatically? ☐ Child samples? No of Samples 50 Latest Sample 36724 Average: 26008 Deviation: 9817

## 2. Kenapa ada yang merah?

Karena gagal. Karena waktu menerima request yang bawah, waktu telah melebihi waktu seharusnya sehingga request di terminate oleh sistem. Lalu kenapa dibawahnya ada yang jalan? Karena ketika akan diterminate oleh sistem, ternyata proses diatas telah selesai, jadi antrian tidak ada lagi dan bisa jalan. (scheduling)

```
java.sql.SQLTransientConnectionException: HikariPool-1 - Connection is not available, request timed out after 30256ms.
```

Pada console di spring boot terdapat error sebagai berikut, yang berarti proses di terminate setelah lewat 30256 ms.

Reference :

[https://www.tutorialspoint.com/mockito/mockito\\_overview.htm](https://www.tutorialspoint.com/mockito/mockito_overview.htm)

<https://www.baeldung.com/integration-testing-in-spring>

<https://www.javacodegeeks.com/2013/07/spring-mvc-requestbody-and-responsebody-demystified.html>