

TUTORIAL 6 APAP Unit Testing dan Load Testing Web App Sederhana

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

→ Karena di dalam instance FlightModel, terdapat atribut Pilot yang membutuhkan object dari PilotModel, sehingga perlu adanya object PilotModel untuk mengisi atribut tersebut.

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

→ Jika dihapus, code tidak error. Namun, saat test dijalankan, menghasilkan error, karena pada database terdapat constraint bahwa pilot_license_number tidak boleh bernilai *null*, sedangkan kita tidak membuat object PilotModel dan menghubungkannya ke FlightModel yang kita buat.

3. Jelaskan yang dilakukan code ini:

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

→ Code ini mengkonfigurasi agar mock melakukan return object flight, ketika memanggil method findByFlightNumber().

4. Jelaskan apa yang dilakukan oleh code:

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

→ Code ini mengkonfigurasi agar mock melakukan return object flight, ketika memanggil method getFlightDetailByFlightNumber().

5. Jelaskan apa yang dilakukan oleh code:

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

→ Code ini berfungsi untuk memeriksa bahwa response status/status http yang diterima ketika mengakses link terkait adalah OK (200).

6. Jelaskan apa yang dilakukan oleh code:

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

→ Code ini berfungsi untuk memeriksa bahwa ditemukan subset dari response body, dalam hal ini adalah flightNumber ketika kita melakukan view flight, yang cocok dengan yang dibandingkan, dalam hal ini adalah flightNumber dari object flight. JsonPath berfungsi untuk mengambil bagian yang kita perlukan saja dari dokumen JSON yang dikembalikan.

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;
}
```

→ Anotasi `@ResponseBody` berfungsi untuk memberitahu controller bahwa object yang direturn secara otomatis ditranslate menjadi JSON dan dikirim kembali menjadi object *HttpResponse*.

8. Berikut adalah hasil Load Testing dengan Apache JMeter:

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename:

Buttons: [Home](#) [Log/Display Only](#) [Errors](#) [Successes](#) [Configure](#)

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
1	15.33.11.460	Thread Group 1-1	HTTP Request	18859	Success	521106	147	1652	9
2	15.33.12.073	Thread Group 1-4	HTTP Request	18927	Success	521106	147	1177	1
3	15.33.12.275	Thread Group 1-5	HTTP Request	18766	Success	521106	147	142	1
4	15.33.11.873	Thread Group 1-3	HTTP Request	18669	Success	521106	147	275	1
5	15.33.12.872	Thread Group 1-7	HTTP Request	18601	Success	521106	147	198	1
6	15.33.12.872	Thread Group 1-6	HTTP Request	18451	Success	521106	147	809	1
7	15.33.11.873	Thread Group 1-2	HTTP Request	20922	Success	521106	147	175	1
8	15.33.13.273	Thread Group 1-10	HTTP Request	18854	Success	521106	147	250	1
9	15.33.13.380	Thread Group 1-8	HTTP Request	20210	Success	521106	147	138	1
10	15.33.12.472	Thread Group 1-6	HTTP Request	20983	Success	521106	147	3020	1
11	15.33.15.400	Thread Group 1-21	HTTP Request	30021	Success	9307	147	30217	2
12	15.33.16.077	Thread Group 1-24	HTTP Request	30221	Success	9307	147	30424	2
13	15.33.16.878	Thread Group 1-23	HTTP Request	30404	Success	9307	147	30588	2
14	15.33.16.575	Thread Group 1-22	HTTP Request	30590	Success	9307	147	30173	2
15	15.33.16.283	Thread Group 1-25	HTTP Request	30178	Success	9307	147	30233	2
16	15.33.16.483	Thread Group 1-26	HTTP Request	30223	Success	9307	147	30294	2
17	15.33.16.678	Thread Group 1-27	HTTP Request	30295	Success	9307	147	30250	2
18	15.33.16.883	Thread Group 1-28	HTTP Request	30251	Success	9307	147	30125	2
19	15.33.17.083	Thread Group 1-29	HTTP Request	30126	Success	9307	147	30305	2
20	15.33.17.284	Thread Group 1-30	HTTP Request	30307	Success	9307	147	30182	2
21	15.33.17.483	Thread Group 1-31	HTTP Request	30183	Success	9307	147	30162	1
22	15.33.17.682	Thread Group 1-32	HTTP Request	30162	Success	9307	147	30283	2
23	15.33.17.882	Thread Group 1-33	HTTP Request	30186	Success	9307	147	30304	2
24	15.33.18.082	Thread Group 1-34	HTTP Request	30286	Success	9307	147	30234	1
25	15.33.18.281	Thread Group 1-35	HTTP Request	30305	Success	9307	147	30112	2
26	15.33.18.482	Thread Group 1-36	HTTP Request	30182	Success	9307	147	17769	2
27	15.33.18.682	Thread Group 1-37	HTTP Request	30287	Success	9307	147	30246	2
28	15.33.18.882	Thread Group 1-38	HTTP Request	30225	Success	9307	147	30282	2
29	15.33.19.082	Thread Group 1-39	HTTP Request	30113	Success	9307	147	30257	2
30	15.33.13.877	Thread Group 1-32	HTTP Request	35570	Success	521106	147	30154	2
31	15.33.19.482	Thread Group 1-41	HTTP Request	30248	Success	9307	147	30305	2
32	15.33.19.682	Thread Group 1-42	HTTP Request	30263	Success	9307	147	30154	2
33	15.33.19.882	Thread Group 1-43	HTTP Request	30268	Success	9307	147	30305	2
34	15.33.20.082	Thread Group 1-44	HTTP Request	30104	Success	9307	147	17651	1
35	15.33.20.281	Thread Group 1-45	HTTP Request	30306	Success	9307	147	30119	1
36	15.33.20.481	Thread Group 1-46	HTTP Request	30211	Success	9307	147	30142	2
37	15.33.13.874	Thread Group 1-13	HTTP Request	36897	Success	521106	147	30196	2
38	15.33.20.881	Thread Group 1-48	HTTP Request	30219	Success	9307	147	17741	3
39	15.33.21.081	Thread Group 1-49	HTTP Request	30143	Success	9307	147	18422	2
40	15.33.21.280	Thread Group 1-50	HTTP Request	30187	Success	9307	147	18396	2
41	15.33.13.473	Thread Group 1-11	HTTP Request	38305	Success	521106	147	18379	2
42	15.33.14.275	Thread Group 1-15	HTTP Request	37564	Success	521106	147	17813	2
43	15.33.14.476	Thread Group 1-17	HTTP Request	37545	Success	521106	147	18492	2
44	15.33.14.674	Thread Group 1-18	HTTP Request	38172	Success	521106	147	18075	2
45	15.33.14.874	Thread Group 1-14	HTTP Request	38664	Success	521106	147	18952	2
46	15.33.15.280	Thread Group 1-20	HTTP Request	37558	Success	521106	147	31275	2
47	15.33.14.880	Thread Group 1-18	HTTP Request	38203	Success	521106	147	31015	2
48	15.33.15.079	Thread Group 1-19	HTTP Request	38303	Success	521106	147		
49	15.33.19.282	Thread Group 1-40	HTTP Request	37567	Success	521106	147		
50	15.33.20.681	Thread Group 1-47	HTTP Request	35403	Success	521106	147		

Scroll automatically? ☐ Child samples? ☐ No of Samples: 50 Latest Sample: 50453 [Refresh](#) [Download CSV](#)

Pada hasilnya, status yang hijau menyatakan bahwa hasil tes berhasil, dan yang merah menunjukkan bahwa hasil tes gagal. Pada 50 percobaan, terdapat yang merah karena performa komputer yang digunakan tidak dapat memadai seluruh tes. Pada tes dilakukan penambahan thread pada link terkait, namun terdapat thread yang gagal untuk dipenuhi karena kapasitas loadnya sudah penuh, sehingga tes menjadi gagal.