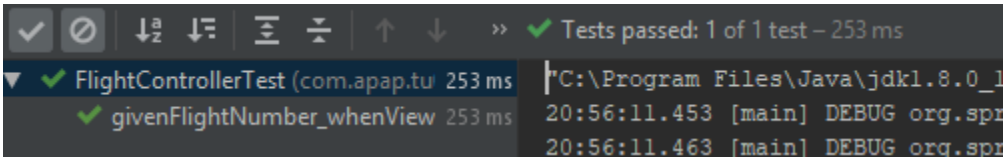
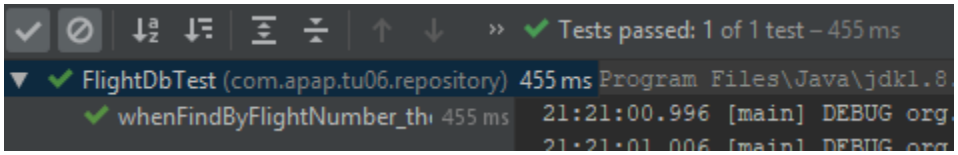
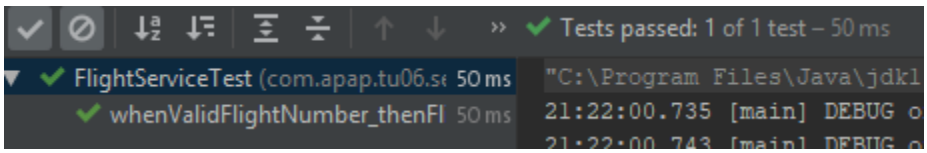
Usamah Nashirul Haq
1606917954

APAP TUTORIAL 6

Hasil JUnit Test







1. Mengapa perlu menginisiasi object PilotModel, sedangkan yang di test hanya FlightModel?
   Jawab: Karena pada database FlightModel terdapat *attribute* **pilot_licenseNumber**. *Attribute* ini adalah suatu **foreign key** yang tidak bisa bernilai **NULL**. Oleh karena itu, objek **Pilot** harus dibuat terlebih dahulu agar objek **Flight** dapat dibuat juga.

2. Jelaskan apa yang akan terjadi jika object PilotModel dihapus dan tidak dilakukan setPilot pada FlightModel?
   Jawab: Maka objek **FlightModel** akan terhapus juga. Karena pada database telah diatur **Delete On Cascade**, jika objek **Pilot** di-*delete*, maka semua objek **Flight** yang memiliki relasi atau hubungan dengan objek **Pilot** tersebut akan ter-*delete* juga.

3. Jelaskan apa yang dilakukan oleh code
   ```
   Mockito.when(flightDb.findByFlightNumber(flight.get().getFlightNumber())).thenReturn(flight);
   ```
   Jawab: *Code* itu mengembalikan objek yang telah dibuat. Jika objek yang telah dibuat pada test berhasil dibuat, code ini bertugas untuk mencari objek tersebut di database dan me-*return*-nya kembali. Objek yang telah di-*return* tersebut akan digunakan sebagai pembanding pada test yang dijalankan.

4. Jelaskan apa yang dilakukan oleh code
   ```
   Mockito.when(flightService.getFlightDetailByFlightNumber(flight.get().getFlightNumber())).thenReturn(flight);
   ```

Jawab: *Code* ini mengembalikan data atau info dari objek **Flight** yang telah kita masukan sebelumnya. Data atau info dari objek tersebut akan digunakan sebagai pembanding pada test yang dijalankan.

5. Jelaskan apa yang ditest oleh code

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

Jawab: *Code* ini melakukan pengecekan pada *path* atau url yang sedang kita test. Apakah *path* tersebut dapat ditemukan dan tidak mengembalikan error seperti 404, 405, dsb.

6. Jelaskan apa yang ditest oleh code

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

Jawab: *Code* ini melakukan pengecekan terhadap objek yang ada pada *path* yang kita berikan apakah sama dengan objek yang telah kita buat di *class test*-nya.

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;
}
```

Jawab: @ResponseBody digunakan untuk memberi tahu kepada *controller* bahwa objek yang ada pada class tersebut dikembalikan secara otomatis dan di-*convert* menjadi JSON, dan lalu dikembalikan sebagai *HttpResponse Object*.

8. Penjelasan 3 percobaan Apache JMeter

Jawab:

a. No Samples = 50, Ramp-Up = 10, Loop = 1

Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	20:28:45.585	Thread Group 1...	HTTP Request	88		1871	127	88	1
2	20:28:45.398	Thread Group 1...	HTTP Request	275		1871	127	275	0
3	20:28:45.197	Thread Group 1...	HTTP Request	476		1871	127	476	1
4	20:28:45.798	Thread Group 1...	HTTP Request	11		1871	127	11	1
5	20:28:45.997	Thread Group 1...	HTTP Request	11		1871	127	10	1
6	20:28:46.197	Thread Group 1...	HTTP Request	12		1871	127	12	1
7	20:28:46.397	Thread Group 1...	HTTP Request	11		1871	127	11	1
8	20:28:46.598	Thread Group 1...	HTTP Request	9		1871	127	9	1
9	20:28:46.798	Thread Group 1...	HTTP Request	9		1871	127	9	1
10	20:28:46.997	Thread Group 1...	HTTP Request	9		1871	127	9	0
11	20:28:47.198	Thread Group 1...	HTTP Request	10		1871	127	10	0
12	20:28:47.398	Thread Group 1...	HTTP Request	8		1871	127	8	1
13	20:28:47.597	Thread Group 1...	HTTP Request	9		1871	127	8	1
14	20:28:47.797	Thread Group 1...	HTTP Request	9		1871	127	9	2
15	20:28:48.000	Thread Group 1...	HTTP Request	9		1871	127	9	1
16	20:28:48.196	Thread Group 1...	HTTP Request	10		1871	127	10	1
17	20:28:48.395	Thread Group 1...	HTTP Request	8		1871	127	7	1
18	20:28:48.596	Thread Group 1...	HTTP Request	8		1871	127	7	1
19	20:28:48.795	Thread Group 1...	HTTP Request	8		1871	127	7	1
20	20:28:48.995	Thread Group 1...	HTTP Request	7		1871	127	7	1
21	20:28:49.196	Thread Group 1...	HTTP Request	8		1871	127	7	1
22	20:28:49.400	Thread Group 1...	HTTP Request	8		1871	127	7	0
23	20:28:49.600	Thread Group 1...	HTTP Request	6		1871	127	6	1
24	20:28:49.800	Thread Group 1...	HTTP Request	6		1871	127	6	0
25	20:28:50.000	Thread Group 1...	HTTP Request	7		1871	127	6	1
26	20:28:50.200	Thread Group 1...	HTTP Request	8		1871	127	8	1
27	20:28:50.399	Thread Group 1...	HTTP Request	7		1871	127	6	1

Scroll automatically? Child samples? No of Samples 50 Latest Sample 6 Average 24 Deviation 75

Usamah Nashirul Haq
1606917954

b. No Samples = 100, Ramp-Up = 10, Loop = 3

Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	21:04:18.983	Thread Group 1...	HTTP Request	16		1871	127	15	2
2	21:04:19.013	Thread Group 1...	HTTP Request	8		1871	127	8	1
3	21:04:19.022	Thread Group 1...	HTTP Request	9		1871	127	9	2
4	21:04:19.079	Thread Group 1...	HTTP Request	7		1871	127	6	1
5	21:04:19.086	Thread Group 1...	HTTP Request	6		1871	127	6	1
6	21:04:19.093	Thread Group 1...	HTTP Request	5		1871	127	5	0
7	21:04:19.184	Thread Group 1...	HTTP Request	6		1871	127	6	0
8	21:04:19.191	Thread Group 1...	HTTP Request	6		1871	127	6	1
9	21:04:19.198	Thread Group 1...	HTTP Request	5		1871	127	5	0
10	21:04:19.283	Thread Group 1...	HTTP Request	7		1871	127	7	1
11	21:04:19.291	Thread Group 1...	HTTP Request	6		1871	127	6	1
12	21:04:19.297	Thread Group 1...	HTTP Request	7		1871	127	7	1
13	21:04:19.385	Thread Group 1...	HTTP Request	6		1871	127	6	1
14	21:04:19.392	Thread Group 1...	HTTP Request	6		1871	127	6	0
15	21:04:19.399	Thread Group 1...	HTTP Request	7		1871	127	6	1
16	21:04:19.509	Thread Group 1...	HTTP Request	5		1871	127	5	1
17	21:04:19.515	Thread Group 1...	HTTP Request	6		1871	127	5	1
18	21:04:19.521	Thread Group 1...	HTTP Request	6		1871	127	5	1
19	21:04:19.586	Thread Group 1...	HTTP Request	6		1871	127	6	1
20	21:04:19.592	Thread Group 1...	HTTP Request	7		1871	127	7	1
21	21:04:19.600	Thread Group 1...	HTTP Request	6		1871	127	5	1
22	21:04:19.693	Thread Group 1...	HTTP Request	6		1871	127	6	1
23	21:04:19.700	Thread Group 1...	HTTP Request	5		1871	127	5	0
24	21:04:19.705	Thread Group 1...	HTTP Request	6		1871	127	6	1
25	21:04:19.793	Thread Group 1...	HTTP Request	6		1871	127	6	1
26	21:04:19.800	Thread Group 1...	HTTP Request	6		1871	127	6	0
27	21:04:19.806	Thread Group 1...	HTTP Request	7		1871	127	6	1

Scroll automatically? Child samples? No of Samples 300 Latest Sample 5 Average 6 Deviation 1

c. No Samples = 1000, Ramp-Up = 5, Loop = 5

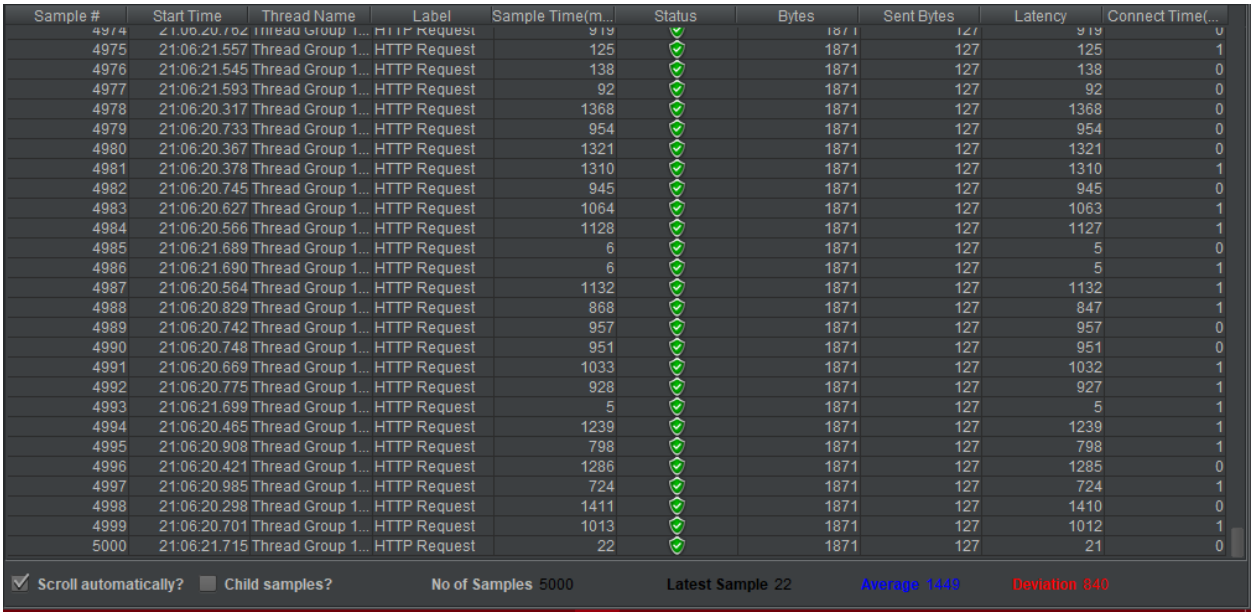
Table fase awal

Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	21:06:09.557	Thread Group 1...	HTTP Request	12		1871	127	12	1
2	21:06:09.567	Thread Group 1...	HTTP Request	8		1871	127	8	1
3	21:06:09.593	Thread Group 1...	HTTP Request	33		1871	127	32	1
4	21:06:09.570	Thread Group 1...	HTTP Request	64		1871	127	63	1
5	21:06:09.578	Thread Group 1...	HTTP Request	65		1871	127	64	1
6	21:06:09.576	Thread Group 1...	HTTP Request	75		1871	127	74	1
7	21:06:09.634	Thread Group 1...	HTTP Request	20		1871	127	20	1
8	21:06:09.651	Thread Group 1...	HTTP Request	11		1871	127	11	1
9	21:06:09.626	Thread Group 1...	HTTP Request	39		1871	127	38	1
10	21:06:09.643	Thread Group 1...	HTTP Request	25		1871	127	24	1
11	21:06:09.662	Thread Group 1...	HTTP Request	7		1871	127	7	1
12	21:06:09.665	Thread Group 1...	HTTP Request	8		1871	127	8	1
13	21:06:09.668	Thread Group 1...	HTTP Request	6		1871	127	6	1
14	21:06:09.670	Thread Group 1...	HTTP Request	7		1871	127	7	0
15	21:06:09.536	Thread Group 1...	HTTP Request	287		1871	127	287	2
16	21:06:09.535	Thread Group 1...	HTTP Request	288		1871	127	288	5
17	21:06:09.549	Thread Group 1...	HTTP Request	277		1871	127	277	1
18	21:06:09.673	Thread Group 1...	HTTP Request	154		1871	127	154	1
19	21:06:09.654	Thread Group 1...	HTTP Request	173		1871	127	173	1
20	21:06:09.675	Thread Group 1...	HTTP Request	152		1871	127	152	0
21	21:06:09.827	Thread Group 1...	HTTP Request	24		1871	127	24	1
22	21:06:09.827	Thread Group 1...	HTTP Request	24		1871	127	24	1
23	21:06:09.824	Thread Group 1...	HTTP Request	27		1871	127	27	0
24	21:06:09.827	Thread Group 1...	HTTP Request	24		1871	127	24	1
25	21:06:09.826	Thread Group 1...	HTTP Request	35		1871	127	35	1
26	21:06:09.829	Thread Group 1...	HTTP Request	54		1871	127	54	1
27	21:06:09.852	Thread Group 1...	HTTP Request	43		1871	127	42	3

Scroll automatically? Child samples? No of Samples 5000 Latest Sample 22 Average 1449 Deviation 840

Usamah Nashirul Haq
1606917954

Table fase akhir



Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
4974	21:06:20.762	Thread Group 1...	HTTP Request	919		1871	127	919	0
4975	21:06:21.557	Thread Group 1...	HTTP Request	125		1871	127	125	1
4976	21:06:21.545	Thread Group 1...	HTTP Request	138		1871	127	138	0
4977	21:06:21.593	Thread Group 1...	HTTP Request	92		1871	127	92	0
4978	21:06:20.317	Thread Group 1...	HTTP Request	1368		1871	127	1368	0
4979	21:06:20.733	Thread Group 1...	HTTP Request	954		1871	127	954	0
4980	21:06:20.367	Thread Group 1...	HTTP Request	1321		1871	127	1321	0
4981	21:06:20.378	Thread Group 1...	HTTP Request	1310		1871	127	1310	1
4982	21:06:20.745	Thread Group 1...	HTTP Request	945		1871	127	945	0
4983	21:06:20.627	Thread Group 1...	HTTP Request	1064		1871	127	1063	1
4984	21:06:20.566	Thread Group 1...	HTTP Request	1128		1871	127	1127	1
4985	21:06:21.689	Thread Group 1...	HTTP Request	6		1871	127	5	0
4986	21:06:21.690	Thread Group 1...	HTTP Request	6		1871	127	5	1
4987	21:06:20.564	Thread Group 1...	HTTP Request	1132		1871	127	1132	1
4988	21:06:20.829	Thread Group 1...	HTTP Request	868		1871	127	847	1
4989	21:06:20.742	Thread Group 1...	HTTP Request	957		1871	127	957	0
4990	21:06:20.748	Thread Group 1...	HTTP Request	951		1871	127	951	0
4991	21:06:20.669	Thread Group 1...	HTTP Request	1033		1871	127	1032	1
4992	21:06:20.775	Thread Group 1...	HTTP Request	928		1871	127	927	1
4993	21:06:21.699	Thread Group 1...	HTTP Request	5		1871	127	5	1
4994	21:06:20.465	Thread Group 1...	HTTP Request	1239		1871	127	1239	1
4995	21:06:20.908	Thread Group 1...	HTTP Request	798		1871	127	798	1
4996	21:06:20.421	Thread Group 1...	HTTP Request	1286		1871	127	1285	0
4997	21:06:20.985	Thread Group 1...	HTTP Request	724		1871	127	724	1
4998	21:06:20.298	Thread Group 1...	HTTP Request	1411		1871	127	1410	0
4999	21:06:20.701	Thread Group 1...	HTTP Request	1013		1871	127	1012	1
5000	21:06:21.715	Thread Group 1...	HTTP Request	22		1871	127	21	0

Scroll automatically? Child samples? No of Samples 5000 Latest Sample 22 Average 1449 Deviation 840

Penjelasan:

Pada 2 percobaan pertama digunakan parameter sebagai berikut **No Samples = 50, Ramp-Up = 10, Loop = 1** dan **No Samples = 50, Ramp-Up = 10, Loop = 1**. Terlihat pada table bahwa Sample Time dari percobaan 1 dan 2 tidak begitu jauh berbeda. *Device* saya masih kuat sebagai server untuk meng-*handle* sejumlah *user* seperti yang telah disebutkan. Namun ketika menjalankan percobaan ke-3, digunakan parameter sebagai berikut **No Samples = 1000, Ramp-Up = 5, Loop = 5**. Pada percobaan ini terlihat pada fase awal sudah berbeda dengan fase awal pada dua percobaan pertama. Terlihat lonjakan Sample Time pada percobaan ke-3. Percobaan ke-3 dilakukan hingga 5000x test dan terdata Sample Time hingga mencapai 2000ms. *Device* saya sudah cukup kesulitan untuk meng-*handle* kasus *user* sebanyak ini.