Nama: Stanley Sebastian
NPM: 1606893494
Kelas: C

TUTORIAL 6

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

   Karena pada FlightModel terdapat atribut Pilot yang perlu diisi. Oleh karena itu, dibutuhkan object dari PilotModel untuk mengisi atribut Pilot tersebut.

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

   Tidak terdapat error ketika object PilotModel dihapus, tetapi saat menjalankan *test* barulah error muncul. Hal ini dikarenakan terdapat *constraint* pada database yang menyebutkan kalau pilot_license_number tidak bisa bernilai null. Karena object PilotModel telah kita dihapus, maka kita tidak menghubungkan antara object PilotModel ke FlightModel.

3. Jelaskan apa yang dilakukan oleh code

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

   Code diatas berfungsi untuk mengkonfigurasi supaya 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 diatas berfungsi untuk mengkonfigurasi supaya Mock melakukan return object flight ketika memanggil method getFlightDetailByFlightNumber().

5. Jelaskan apa yang dilakukan oleh code

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

   Code diatas berfungsi untuk memeriksa apakah response status (http) yang diterima adalah OK (200) ketika mengakses link yang bersangkutan.

6. Jelaskan apa yang dilakukan oleh code

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

   Code diatas berfungsi untuk memeriksa apakah parameter flightNumber ketika melakukan view flight sama (cocok) dengan flightNumber dari object flight. Selanjutnya, JsonPath berfungsi untuk mengambil potongan/bagian yang hanya diperlukan dari JSON yang direturn.

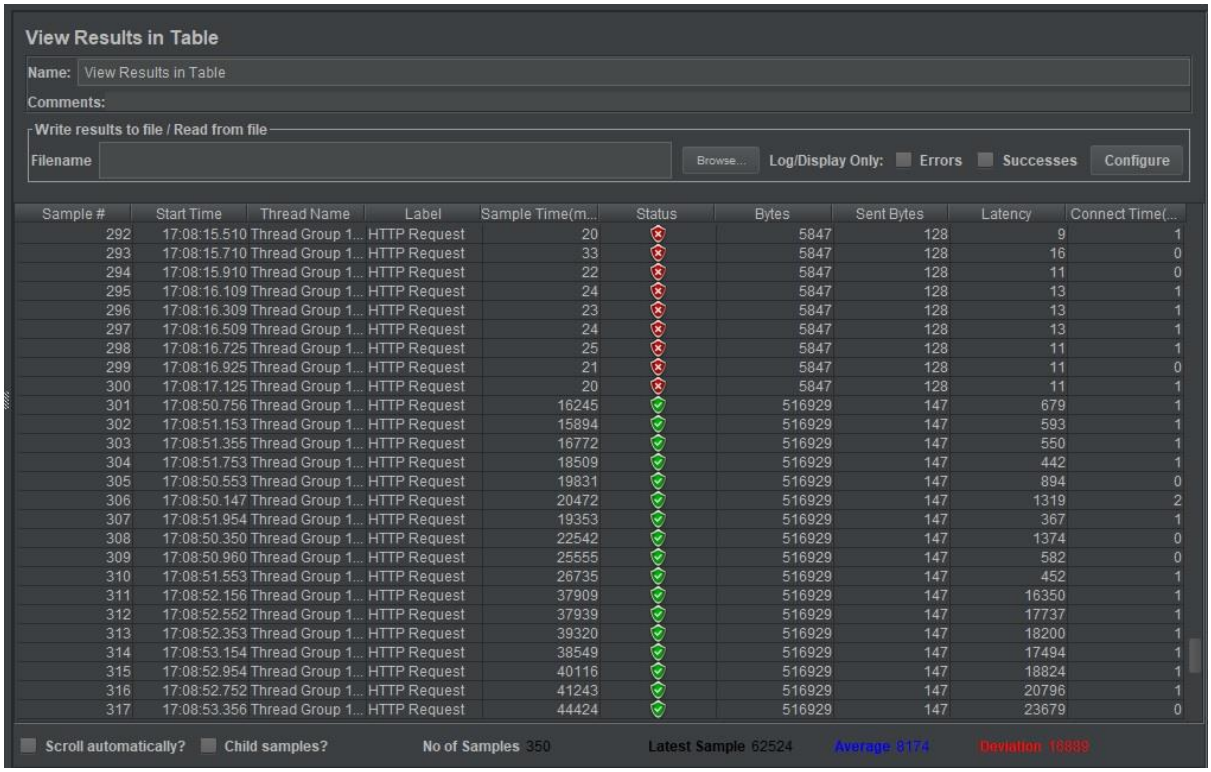
7. Jelaskan anotasi @ResponseBody yang ada pada route "/flight/view"

```
@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 pada code diatas berfungsi untuk mengabarkan Controller kalau object yang direturn secara otomatis telah diterjemahkan menjadi JSON dan dikirim kembali menjadi object HttpResponse.

Nama: Stanley Sebastian
NPM: 1606893494
Kelas: C

8. Hasil Load Testing Apache JMeter



Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
292	17:08:15.510	Thread Group 1...	HTTP Request	20		5847	128	9	1
293	17:08:15.710	Thread Group 1...	HTTP Request	33		5847	128	16	0
294	17:08:15.910	Thread Group 1...	HTTP Request	22		5847	128	11	0
295	17:08:16.109	Thread Group 1...	HTTP Request	24		5847	128	13	1
296	17:08:16.309	Thread Group 1...	HTTP Request	23		5847	128	13	1
297	17:08:16.509	Thread Group 1...	HTTP Request	24		5847	128	13	1
298	17:08:16.725	Thread Group 1...	HTTP Request	25		5847	128	11	1
299	17:08:16.925	Thread Group 1...	HTTP Request	21		5847	128	11	0
300	17:08:17.125	Thread Group 1...	HTTP Request	20		5847	128	11	1
301	17:08:50.756	Thread Group 1...	HTTP Request	16245		516929	147	679	1
302	17:08:51.153	Thread Group 1...	HTTP Request	15894		516929	147	593	1
303	17:08:51.355	Thread Group 1...	HTTP Request	16772		516929	147	550	1
304	17:08:51.753	Thread Group 1...	HTTP Request	18509		516929	147	442	1
305	17:08:50.553	Thread Group 1...	HTTP Request	19831		516929	147	894	0
306	17:08:50.147	Thread Group 1...	HTTP Request	20472		516929	147	1319	2
307	17:08:51.954	Thread Group 1...	HTTP Request	19353		516929	147	367	1
308	17:08:50.350	Thread Group 1...	HTTP Request	22542		516929	147	1374	0
309	17:08:50.960	Thread Group 1...	HTTP Request	25555		516929	147	582	0
310	17:08:51.553	Thread Group 1...	HTTP Request	26735		516929	147	452	1
311	17:08:52.156	Thread Group 1...	HTTP Request	37909		516929	147	16350	1
312	17:08:52.552	Thread Group 1...	HTTP Request	37939		516929	147	17737	1
313	17:08:52.353	Thread Group 1...	HTTP Request	39320		516929	147	18200	1
314	17:08:53.154	Thread Group 1...	HTTP Request	38549		516929	147	17494	1
315	17:08:52.954	Thread Group 1...	HTTP Request	40116		516929	147	18824	1
316	17:08:52.752	Thread Group 1...	HTTP Request	41243		516929	147	20796	1
317	17:08:53.356	Thread Group 1...	HTTP Request	44424		516929	147	23679	0

Hasil yang muncul pada saat saya melakukan Load Testing Apache JMeter adalah seperti diatas, terdapat status yang hijau (v) dan merah (*x*). Status yang hijau menandakan kalau hasil tes berhasil sedangkan yang merah menandakan kalau tes gagal. Jika diingat kembali, ketika melakukan tes terdapat penambahan thread pada link yang terkait, tetapi terdapat thread yang gagal untuk dipenuhi karena kapasitas load sudah penuh dan tes menjadi gagal.