

Soal Tutorial

1. Mengapa perlu menginisiasi object DealerModel, sedangkan yang di test hanya CarModel?

→ Karena pada saat melakukan inisiasi *object carModel*, diperlukan *object dealer* yang akan digunakan untuk di *set* menjadi *dealer* dari mobil tersebut.

2. Jelaskan apa yang akan terjadi jika object DealerModel dihapus dan tidak dilakukan setDealer pada CarModel?

→ Akan terjadi error, karena pada segmen *then* kita akan menyamakan *object carModel* yang kita inisiasi dengan *object carModel* yang diambil dari *database*. Jika kita tidak melakukan *setDealer* pada *object carModel* yang kita inisiasi, maka akan ada perbedaan nilai *attribute dealer* antara *object carModel* yang kita inisiasi dengan yang kita ambil dari *database*.

3. Jelaskan apa yang dilakukan oleh code?

```
Mockito.when(carDb.findByType(carModel.get().getType())).thenReturn(carModel);
```

→ Line ini akan memastikan pemanggilan *method findByType* pada kelas *carDb* untuk mengembalikan *object carModel* yang kita sudah inisiasi (*carModel*).

4. Jelaskan apa yang dilakukan oleh code

```
Mockito.when(carService.getCarDetailByType(carModel.get().getType())).thenReturn(carModel);
```

→ Line ini akan memastikan pemanggilan *method getCarDetailByType* pada kelas *carService* untuk mengembalikan *object carModel* yang kita sudah inisiasi (*carModel*).

5. Jelaskan apa yang ditest oleh code

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

→ Line ini mengecek apakah ketika mengakses url “/car/view” mendapatkan HTTP Response status 200

6. Jelaskan apa yang ditest oleh code

```
.andExpect(MockMvcResultMatchers.jsonPath("$.type", Matchers.is(car.getType())));
```

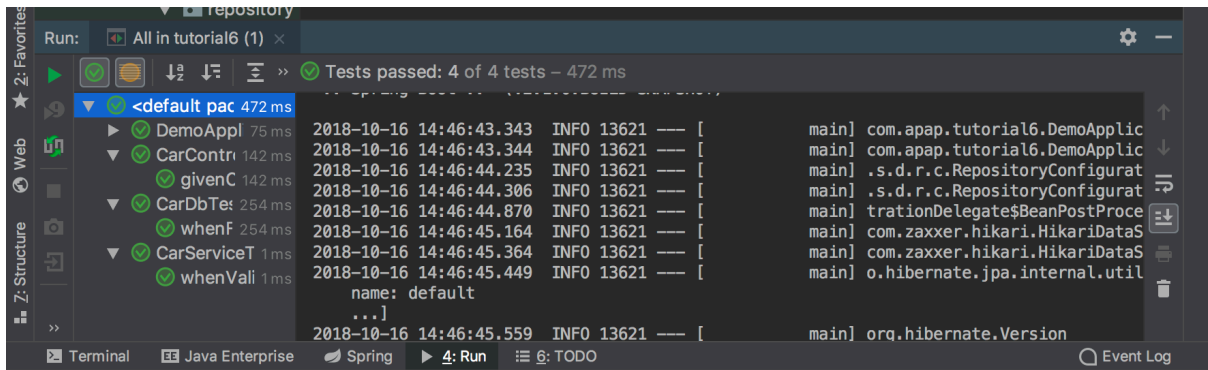
→ Line ini mengecek apakah *json* hasil *response* memiliki *value type* yang sama dengan *type* dari *object car* yang kita inisiasi.

7. Jelaskan anotasi @ResponseBody yang ada pada route “car/view”

```
@RequestMapping(value = "/car/view", method = RequestMethod.GET)
private @ResponseBody CarModel view(@RequestParam(value = "type") String type, Model model) {
    CarModel archive = carService.getCarDetailByType(type).get();
    return archive;
}
```

→ Return value pada method ini akan menghasilkan response HTTP. Respon HTTP tersebut akan di convert menjadi tipe data yang diminta oleh HTTP Request yang pada kasus ini adalah JSON.

Muhammad Faisal T.R
1606838666



Seluruh Test Berhasil Yeay!

Load Testing Dengan Apache JMeter

