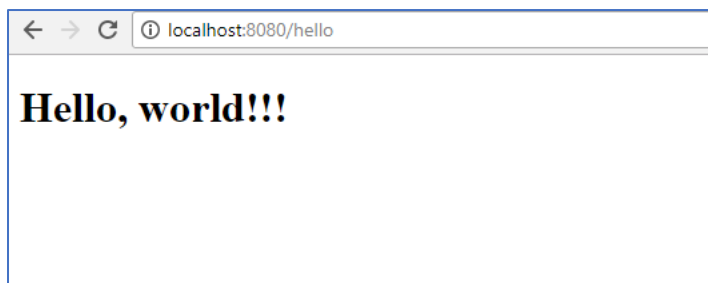


Latihan Project Hello World

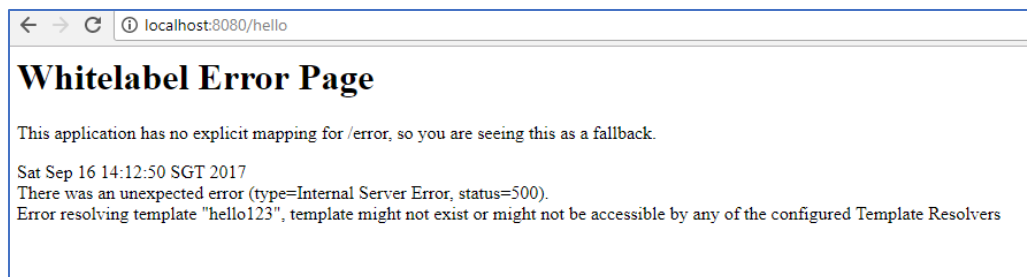
1. Tidak compile error, namun saat dibuka kembali localhost:8080/hello maka menampilkan Whitelable Error Page. Hal tersebut terjadi karena `@RequestMapping` telah diganti menjadi `"/hello123"`, sehingga seharusnya mengakses localhost:8080/hello123



2. Tidak compile error, page hello sebelumnya masih muncul. Hal tersebut membuktikan bahwa penggantian nama method tidak berpengaruh.

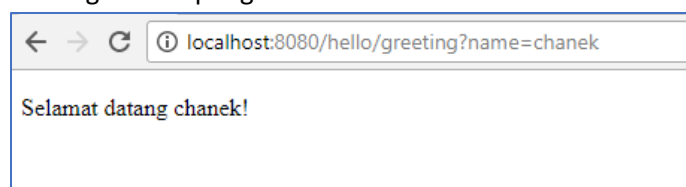


3. Tidak compile error, hanya saja menampilkan Whitelable Error Page, di mana file hello123 tidak eksis. Hal tersebut terjadi karena file html dalam templates bernama hello.html bukan hello123.html. File return "x" menandakan "x" adalah file html.

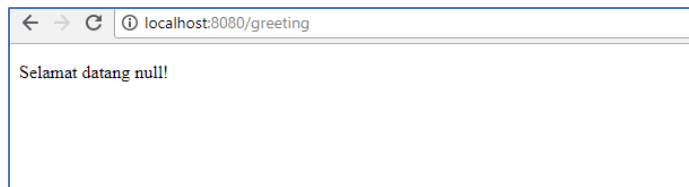


Latihan Request Parameter

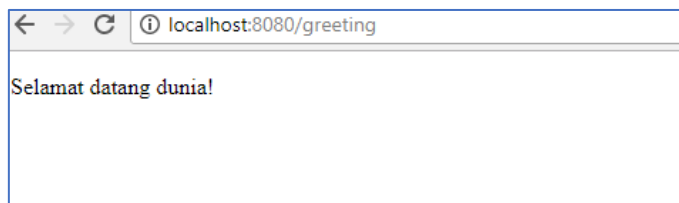
1. Hasilnya : "Selamat datang chanek" . Hal tersebut berarti bahwa `@RequestMapping` berfungsi untuk mengatur url pengaksesan.



2. Hasilnya : “Selamat datang null” . Hal tersebut terjadi karena param yang diambil dengan methode query string dalam url tidak ada. Itu berarti kita dapat menaruh required true/false pada @RequestParam.



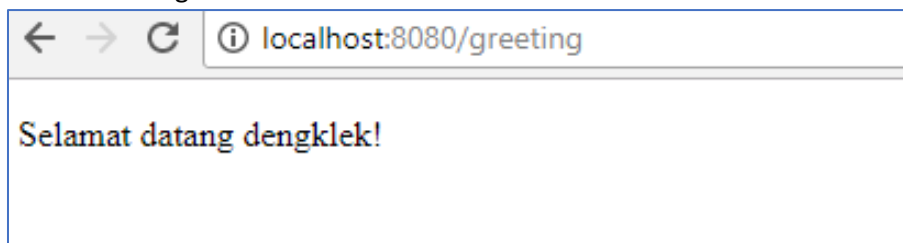
3. Hasilnya : “Selamat datang dunia” . Selain required pada @RequestParam, ada hal lain yaitu defaultValue, jadi saat kita tidak mengisi query string yang berupa param, maka nilai akan diambil dari defaultValue.



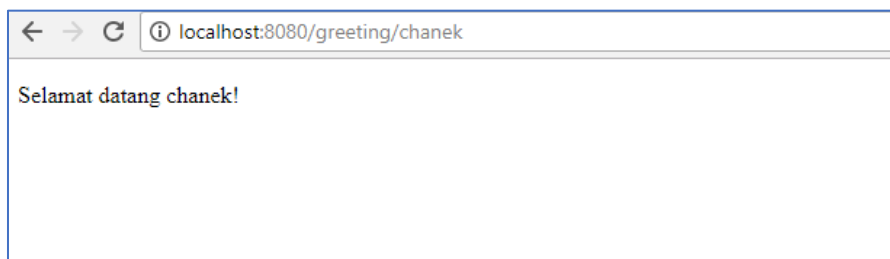
4. Karena tulisan “Sapaan untuk user” berada di luar tag <p>. Seharusnya dia berdiri sendiri menjadi <p> Sapaan untuk user</p> .

Latihan Path Variable

1. Hasilnya : “Selamat datang dengklek” . Hal ini berarti, nama dengklek dicetak karena query string “name” kosong atau tidak diisi.



2. Hasilnya : “Selamat datang chanek” . Hal ini berarti bahwa query string “name” ialah chanek, sehingga mencetak nama chanek.



Latihan

Saat mengerjakan soal latihan yang mengambil query string a dan b untuk dikalikan, saya menggunakan `@RequestMapping("/multiple")`. Di situ saya menggunakan `@RequestParam` sebanyak 2 kali untuk mengambil value "a" dan "b" serta membuat object model juga sebanyak dua kali (modela dan modelb). Kemudian, karena tipe param adalah String, maka saya melakukan `Integer.parseInt()` terhadap a dan b. Setelah itu saya membuat variable int perkalian di mana isinya mengalikan int a dan int b lalu kemudian mengadd variable-variable tersebut ke dalam model. Saya juga menambahkan `defaultValue="0"` karena jika nilai a / b tidak terisi maka otomatis akan dikalikan dengan 0.

```
//localhost:8080/multiple?a=1&b=2
@RequestMapping("/multiple")
public String multiple(@RequestParam(value="a", required = false, defaultValue="0" ) String a,
    Model modela, @RequestParam(value="b", required = false, defaultValue="0" ) String b, Model modelb)
{
    int aa = Integer.parseInt(a);
    int bb = Integer.parseInt(b);
    int perkalian = aa*bb;

    modela.addAttribute("a", aa);
    modelb.addAttribute("b", bb);
    modela.addAttribute("c", perkalian);

    return "multiple";
}
```

