

Pada tutorial kali ini saya belajar bagaimana cara memberi parameter dan memakai parameter tersebut di controller dan bagaimana mentransfer parameter tersebut agar bisa tampil di halaman html yang dibuat. Ada 2 cara yang dapat dipakai yaitu request parameter dan path variabel.

Untuk latihan Perkalian, saya menggunakan cara request parameter sebanyak 2. Lalu parameter di parse supaya bisa jadi int dan bisa dihitung perkaliannya. Dan data/parameter yang dikirim ke view nya (ke html nya) yaitu angka pertama, angka kedua dan hasil dari perkaliannya.

Di bawah ini adalah hasil jawaban dari setiap poin latihan.

Project Hello World

1. Ganti baris tersebut menjadi `@RequestMapping("/hello123")`

Pertanyaan:

- Apakah compile error?
Jawab = tidak terjadi compile error
- Jika tidak, stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:8080/hello apa yang terjadi?

Error Berikut :



Jawab :

Terjadi error karena request parameternya diganti jadi hello123 tetapi yang kita akses hello

- Kembalikan RequestMapping menjadi /hello

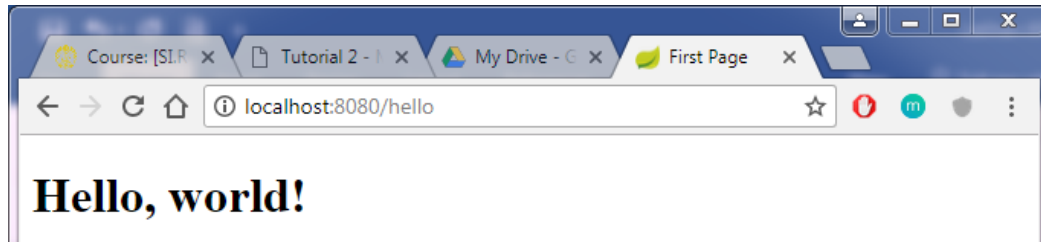
2. Ganti nama method index() dengan nama method hello()

Pertanyaan:

- Apakah compile error?
Jawab : tidak terjadi compile error
- Jika tidak, Stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:8080/hello apakah page hello sebelumnya masih muncul?

Jawab :

ya, page hello sebelumnya masih muncul



- Kembalikan nama method menjadi index()

3. Ganti string return type menjadi return "hello123";

Pertanyaan:

- Apakah compile error?

Jawab : Tidak terjadi compile error

- Jika tidak, Stop Spring Boot yang sedang berjalan, run kembali dan buka localhost: 8080/hello apakah page hello sebelumnya masih muncul?

Jawab :

Page hello sebelumnya tidak muncul dan malah muncul error sbb :



Kembalikan return type menjadi "hello"

Pertanyaan:

- Menandakan apakah String yang di-return tersebut?

Jawab : Menandakan menuju halaman template mana yang akan dituju.

Request Parameter (Query String)

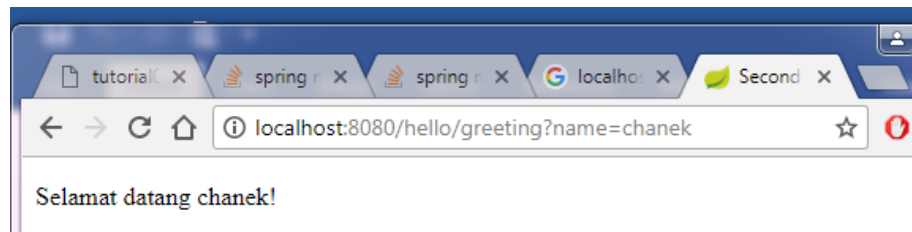
1. Ubah nilai anotasi RequestMapping dari "/greeting" menjadi "/hello/greeting"

Buka localhost: 8080/hello/greeting?name=chanek

Pertanyaan:

- apakah hasilnya?

Jawab : Tidak terjadi error dan berhasil memunculkan pesan "Selamat datang chanek!".



Kembalikan request map menjadi @RequestMapping("/greeting")

2. Akses localhost: 8080/greeting

Pertanyaan:

- Apakah hasilnya?

Jawab :

Terjadi Error dikarenakan tidak ada parameter, padahal seharusnya method menerima parameter berupa nama.



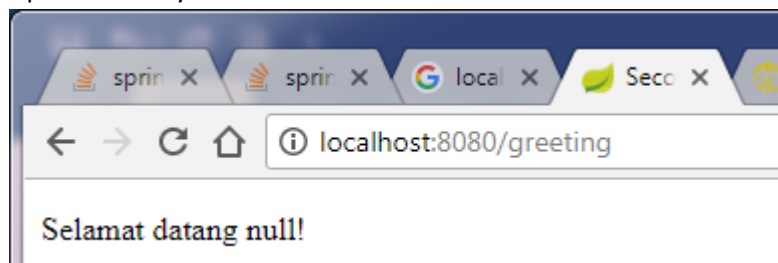
Ubah header method greeting menjadi seperti berikut

```
public String greeting (@RequestParam(value = "name", required = false) String name, Model model)
```

Stop Spring Boot yang sedang berjalan, run kembali, buka localhost: 8080/greeting

Pertanyaan:

- Apakah hasilnya?



Jawab :

Page berhasil dijalankan karena parameter sudah tidak wajib untuk diberikan, namun nama yang muncul null karena tidak diberi parameter nama.

3. Ubah header method greeting menjadi seperti berikut

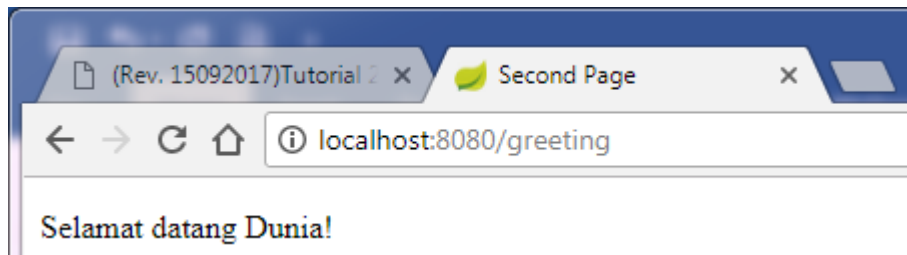
```
public String greeting (@RequestParam(value = "name", required = false, defaultValue = "dunia") String name, Model model)
```

Stop Spring Boot yang sedang berjalan, run kembali, dan buka localhost: 8080/greeting.

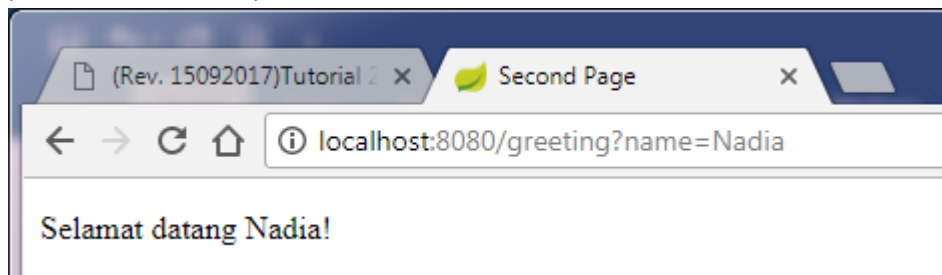
Pertanyaan:

- apakah hasilnya?

Jawab :



Nama yang muncul sudah tidak null lagi karena sudah diberi default value untuk ketika tidak diberi parameter nama, namun jika dicoba dimasukkan nama maka nama yang di parameter akan tetap muncul. Contoh :



4. Perhatikan bahwa pada berkas greeting.html, tag paragraf yang kita tambahkan adalah sebagai berikut :

```
<p th:text="'Selamat datang ' + ${name} + '!'">Sapaan untuk user</p>
```

Pertanyaan:

- Mengapa tulisan “Sapaan untuk user” tidak pernah muncul?

Jawab :

Karena terdapat atribut **th:text**, dimana ia menggantikan posisi si “Sapaan untuk user” dan hanya menampilkan apa yang ada di dalam atribut th:text tersebut.

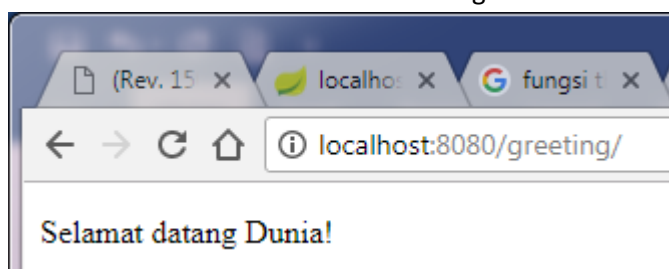
Path Variable

1. Akses localhost:8080/greeting/

Pertanyaan:

- Apa hasilnya?

Jawab : Muncul tulisan “Selamat datang Dunia!”



Ubah method greetingPath menjadi seperti berikut

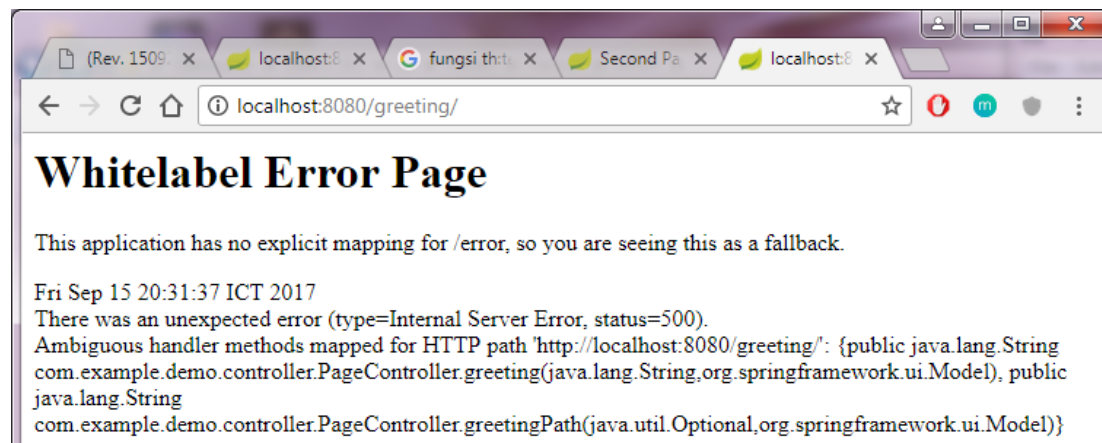
```
@RequestMapping(value = {"/greeting", "greeting/{name}"})
public String greetingPath(@PathVariable Optional<String> name, Model model) {
    if (name.isPresent()) {
        model.addAttribute("name", name.get());
    } else {
        model.addAttribute("name", "dengklek");
    }
    return "greeting";
}
```

Akses localhost:8080/greeting/

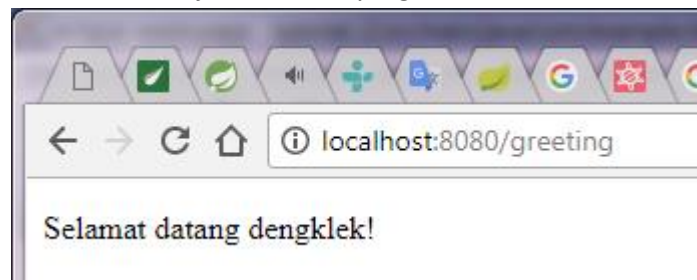
Pertanyaan:

- Apa hasilnya?

Jawab : Muncul error



Hal ini karena ada 2 method yg bisa dijalankan dengan /greeting. Namun ketika method public String greeting dihapus dan hanya ada method public String greetingPath maka halaman bisa dijalankan dan yang muncul sbb :



Akses localhost:8080/greeting/chanek

Pertanyaan:

- Apa hasilnya?

Jawab : tidak terjadi error dan pada halaman muncul tulisan "Selamat Datang chanek!"

