Ruth Nolytha Putri Sari

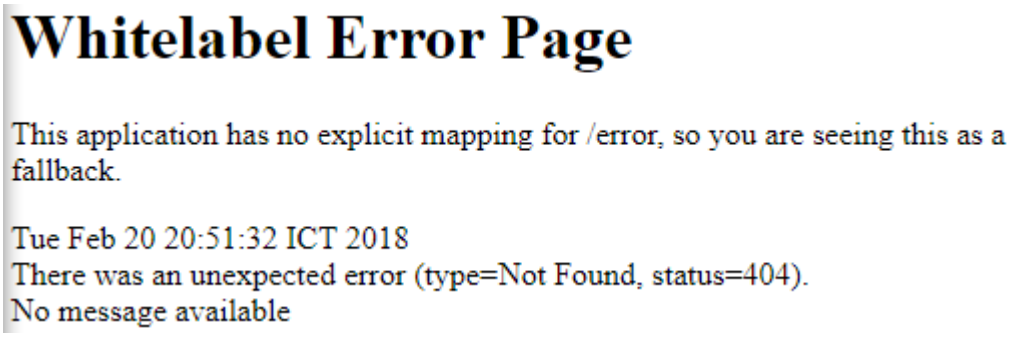
1606955006

Latihan Project Hello World

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

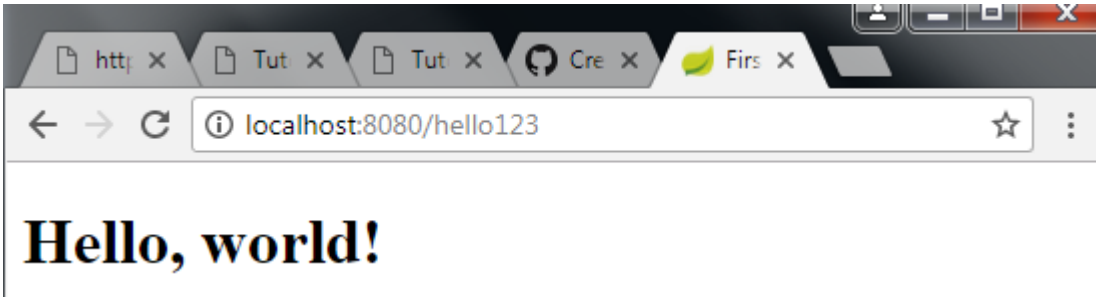
```
@Controller
public class PageController {
    @RequestMapping("/hello")
    public String index() {
        return "hello";
    }
}
@Controller
public class PageController {
    @RequestMapping("/hello123")
    public String index() {
        return "hello";
    }
}
```

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



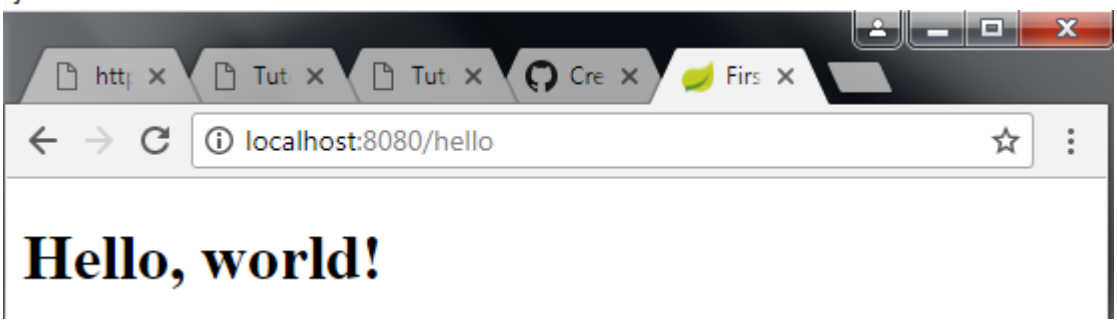
Seperti yang dapat dilihat pada gambar diatas, terjadi *compile error* karena alamat tempat @RequestMapping menuju tidak tersedia, karena yang kita panggil pada *browser* adalah "/hello". Tetapi ketika kita menggunakan @RequestMapping menuju "/hello123" dan memanggilnya menggunakan *url* yang sama, maka *browser* akan menampilkan *view* yang kita inginkan.

```
@Controller
public class PageController {
    @RequestMapping("/hello123")
```



2. Ganti nama method index() dengan nama method hello()
   Pertanyaan: Apakah compile error? Jika tidak, Stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:8080/hello apakah page hello sebelumnya masih muncul?

```
@Controller
public class PageController {
    @RequestMapping("/hello")
    public String hello() {
        return "hello";
    }
}
```



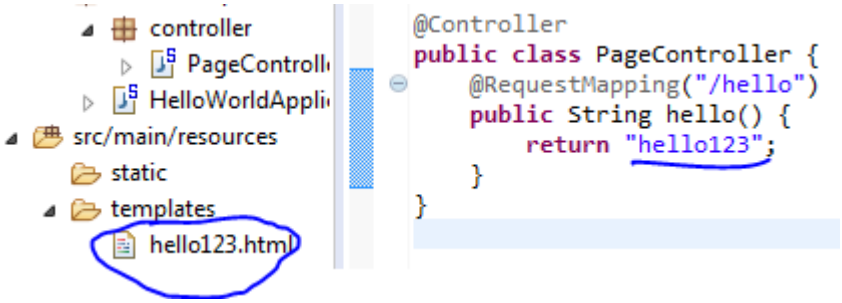
   Tidak akan terjadi *compile error* karena anotasi @RequestMapping("/hello") di atas method hello() menandakan bahwa jika ada request HTTP pada *path* /hello akan dipanggil *method* hello(). Apapun nama *method* yang memanggil @RequestMapping diatas deklarasi *method*-nya akan menggunakan @RequestMapping tersebut.

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

```
@Controller
public class PageController {
    @RequestMapping("/hello")
    public String hello() {
        return "hello123";
    }
}
```

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

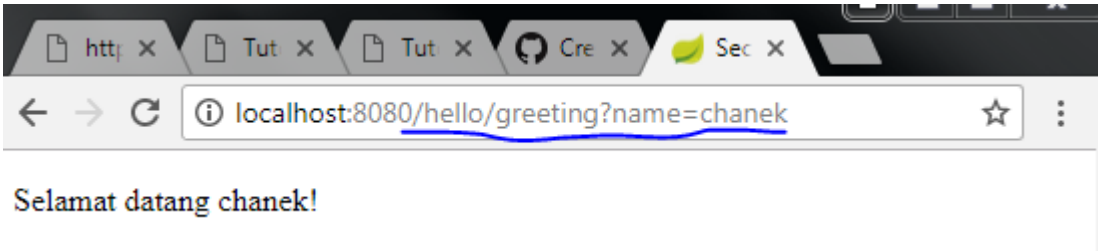
Akan terjadi *compile error*, karena *method* yang dijalankan mengharapkan nilai *return* "hello123". Sedangkan kita hanya memiliki hello.html pada *resources view* kita, bukan hello.html. Ketika mengganti nama *view* hello.html menjadi hello123.html sesuai *return* yang diharapkan dari *method* kita, maka browser akan menampilkan *view* yang kita inginkan.



Pertanyaan: Menandakan apakah String yang di-return tersebut?
Menandakan nama *view* yang diharapkan untuk ditampilkan ketika menjalankan method tersebut.

Request Parameter (Query String)

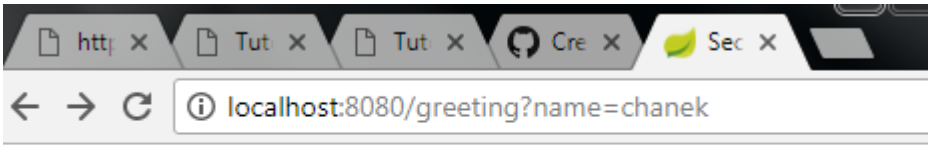
1. Ubah nilai anotasi RequestMapping dari "/greeting" menjadi "/hello/greeting"
   Buka localhost:8080/hello/greeting?name=chanek



Pertanyaan: apakah hasilnya?
Hasilnya akan menampilkan *view* greeting.html pada browser karena walau anotasi *RequestMapping*-nya diganti dan cara pemanggilan bpada *browser*-nya juga disesuaikan, tetapi *method*-nya tetap mengharapkan tampilan yang dikembalikan dari *view* greeting.html

Nilai parameter yang ditangkap melalui anotasi *RequestParam* dari *url* yang dijalankan pada *browser* kemudian digunakan pada *view* nilainya untuk ditampilkan kembali pada halaman *browser*.

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

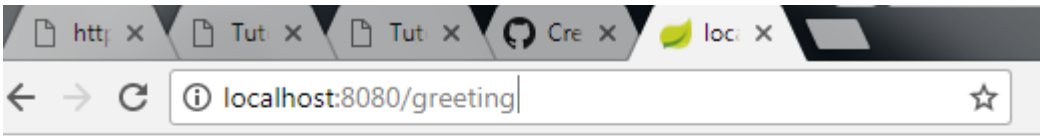


Selamat datang chanek!

Hal ini terjadi karena anotasi *RequestMapping* tersebut menjalankan *method index* yang berada pada baris tepat dibawah deklarasi anotasi tersebut. Maka yang dikembalikan adalah *view* greeting.html

```
@RequestMapping("/greeting")
public String index(@RequestParam(value = "name") String name, Model model) {
    model.addAttribute("name", name);
    return "greeting";
}
```

2. Akses localhost:8080/greeting
Pertanyaan: Apakah hasilnya?



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

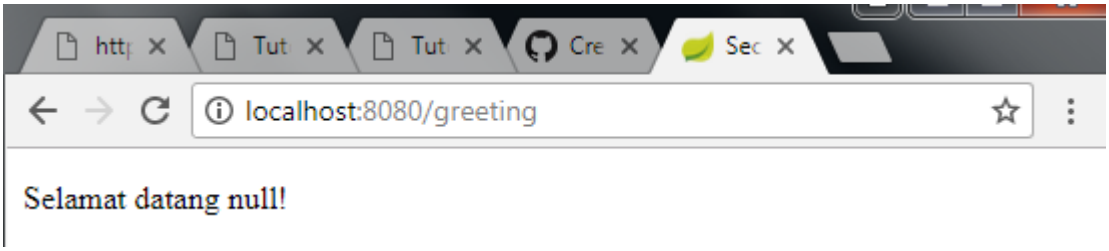
Tue Feb 20 21:22:32 ICT 2018
There was an unexpected error (type=Bad Request, status=400).
Required String parameter 'name' is not present

Karena String parameter yang dibutuhkam saat deklarasi anotasi RequestParam harus memiliki nilai.

3. Ubah header method greeting menjadi seperti berikut

```
@RequestMapping("/greeting")
public String index(@RequestParam(value = "name", required=false) String name, Model model) {
    model.addAttribute("name", name);
    return "greeting";
}
```

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



Penambahan atribut "*required=false*" berarti memungkinkan kita menjalankan *method* tersebut tanpa harus menerima parameter yang di-*request*, yang menyebabkan nilai parameter tersebut menjadi *null* ketika digunakan pada *view greeting.html*

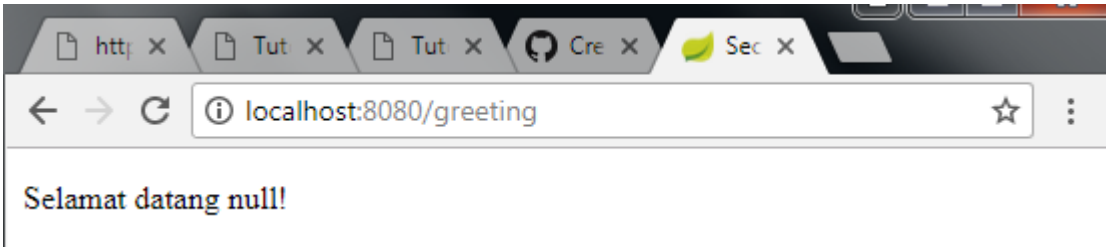
4. Pertanyaan: Mengapa tulisan "Sapaan untuk user" tidak pernah muncul?

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

Karena ada th:text yang merupakan salah satu atribut yang terdapat pada Thymeleaf. Atribut ini akan mengevaluasi ekspresi dan menetapkan nilai evaluasi pada tempat *tag*-nya melekat sebagai *body* dari *tag* tersebut. Maka nilai apapun yang ada pada *tag body* tersebut akan digantikan nilainya oleh isi dari atribut th:text tersebut.

Path Variable

1. Akses localhost:8080/greeting/
   Pertanyaan: Apa hasilnya?



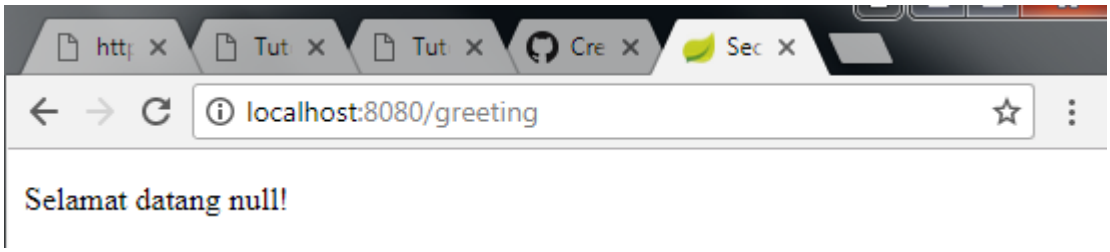
Pada pengeksekusian diatas yang diakses adalah method *index* pada anotasi *RequestMapping* yang terlihat seperti potongan *code* dibawah. *Method* tersebut tidak mengharuskan adanya nilai dari parameter yang diinginkan karena terdapat atribut *required=false* pada anotasi *RequestParam*-nya.

```
@RequestMapping("/greeting")
public String index(@RequestParam(value = "name", required=false) String name, Model model) {
    model.addAttribute("name", name);
    return "greeting";
}
```

Ubah method greetingPath menjadi seperti berikut

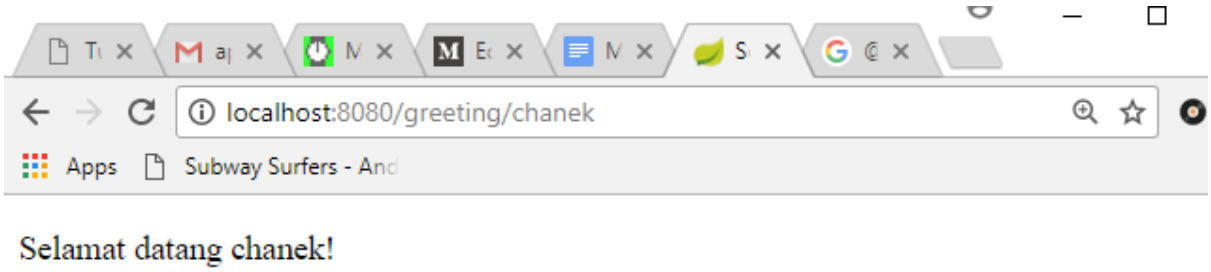
```
@RequestMapping({"/greeting/", "greeting/{name}"})
public String greetingPath(@PathVariable Optional<String> name, Model model) {
    if(name.isPresent()) {
        model.addAttribute("name", name.get());
    } else {
        model.addAttribute("name", "apap");
    }
    return "greeting";
}
```

Apa hasilnya?



Sama seperti method index sebelumya, anotasi PathVariable pada method ini memungkinkan nilainya tidak berisi(null) karena variable name yang digunakan merupakan tipe String yang optional.

Akses localhost:8080/greeting/chanek
Pertanyaan: Apa hasilnya?



Karena parameter yang dikirim melalui pemanggilan localhost:8080/greeting/chanek berisi nilai String=chanek, maka nilai string tersebut akan dilempar ke view greeting.html untuk ditampilkan pada halaman browser.

Latihan

1. Buatlah satu halaman baru untuk melakukan operasi perkalian

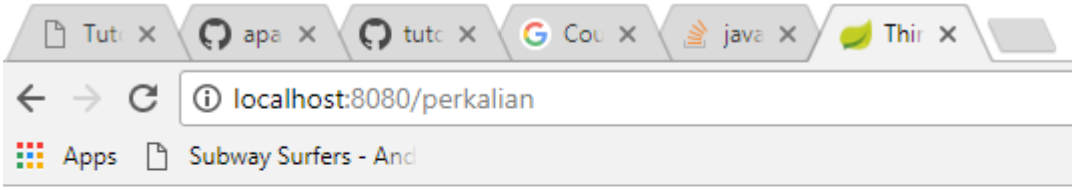
```
@Controller
public class PerkalianController {
      @RequestMapping("/perkalian")
      public String perkalian(@RequestParam(value = "a",
required=false, defaultValue = "0") Integer a, @RequestParam(value =
"b", required=false, defaultValue = "0") Integer b, Model model) {
            model.addAttribute("a", a);
            model.addAttribute("b", b);
            model.addAttribute("c", a*b);
```

```
9.              return "perkalian";
10.             }
11. }
```

2. View

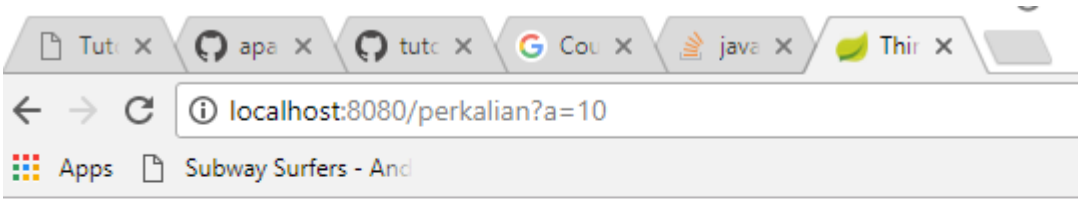
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Third Page</title>
    </head>
    <body>
        <h1>Perkalian</h1>
        <p th:text="${a} + ' x ' + ${b} + ' = ' + ${c}"></p>
    </body>
</html>
```

3. Hasil Tampilan



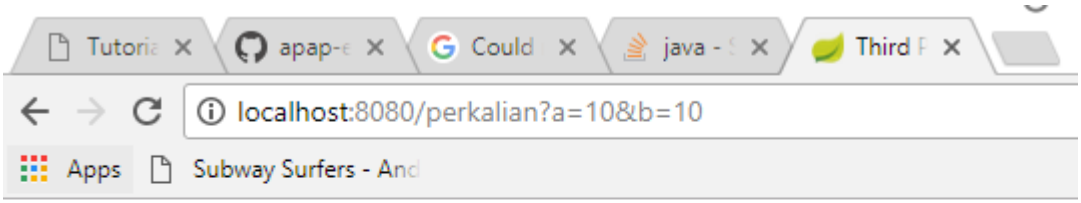
Perkalian

0 x 0 = 0



Perkalian

10 x 0 = 0



Perkalian

10 x 10 = 100