

## Latihan Project Hello World

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

```
@Controller
public class PageController {

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

**Pertanyaan:** Apakah *compile error*?

**Jawaban:** Tidak terjadi *compile error* karena tidak muncul *error log* saat *compile*. Karena `/hello` diganti menjadi `/hello123` maka saat `/hello` diakses *RequestMapping*-nya tidak ditemukan dalam *controller*. Whitelabel Error Page, 404 not found.



## Whitelabel Error Page

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

Wed Sep 13 11:59:18 ICT 2017

There was an unexpected error (type=Not Found, status=404).

No message available

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

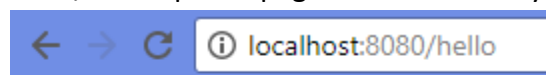
```
@Controller
public class PageController {

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

**Pertanyaan:** Apakah *compile error*?

**Jawaban:** Tidak terjadi *compile error* karena tidak muncul *error log* saat *compile*

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



## Hello, world!

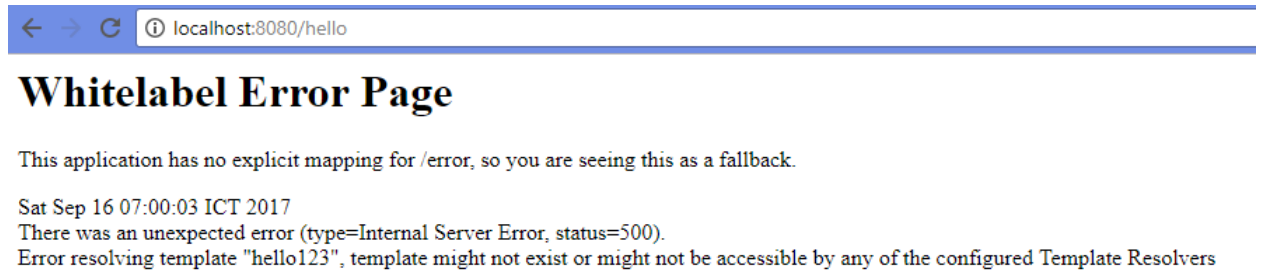
**Jawaban:** Ya, hal ini disebabkan karena penamaan pada method tidak akan mempengaruhi proses pencarian *RequestMapping* dalam *controller* di browser.

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

```
@Controller
public class PageController {

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

**Pertanyaan:** Apakah compile error?



**Jawaban:** Tidak terjadi *compile error* karena tidak muncul *error log* saat *dicompile* tapi Whitelable Error Page dengan error 500 – Internal Server Error

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

**Jawaban:** Tidak muncul karena saat masuk ke dalam method dengan RequestMapping /hello, method tersebut mereturn hello123 dan akan mencari file hello123.html yang terdapat di dalam resource/templates, jika file tersebut tidak ditemukan maka akan menyebabkan *error*.

**Pertanyaan:** Menandakan apakah String yang di-return tersebut?

**Jawaban:** String yang di *return* adalah nama file/dokumen html yang akan ditampilkan ke localhost dari folder resource/templates

## Latihan Request Parameter

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

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

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



Selamat datang chanek!

**Pertanyaan:** apakah hasilnya?

**Jawaban:** Karena route `/greeting` sudah diganti menjadi `/hello/greeting` dan ketika diakses di browser akan mencari requestmapping `/greeting` dan tidak menemukan requestmapping tersebut sehingga muncul Whitelable Error Page dengan error 404-Not Found

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

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

## 2. Akses `localhost: 8080/greeting`



**Pertanyaan:** Apakah hasilnya?

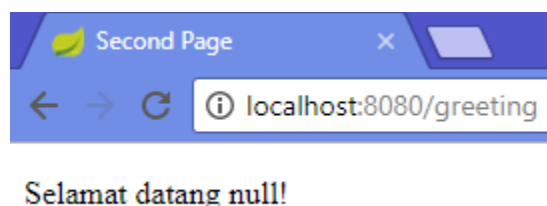
**Jawaban:** Whitelable Error Page dengan error 400-Bad Request

Ubah header method greeting menjadi seperti berikut :

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

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

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



**Pertanyaan:** Apakah hasilnya?

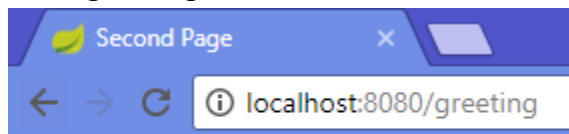
**Jawaban:** Menampilkan null pada variable setelah teks “Selamat datang” karena parameter *name* tidak dimasukkan *value* sehingga saat masuk kedalam method akan menjadi null, hal ini tidak menimbulkan *error* karena sudah di set parameter *required=false* di dalam RequestMapping sehingga parameter *name* diperbolehkan untuk tidak diisi di URL

3. Ubah header method greeting menjadi seperti berikut :

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

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

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



Selamat datang dunia!

**Pertanyaan:** apakah hasilnya?

**Jawaban:** Menampilkan nama sebagai default setelah kata “Selamat datang” yaitu “dunia”, hal ini terjadi karena parameter *name* tidak terdapat dalam URL tersebut sehingga saat masuk ke dalam method akan menjadi null, namun karena sudah di set *defaultValue=dunia*, maka nilai null tersebut akan ditimpa dengan kata “dunia”.

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>
```

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

**Pertanyaan:** Mengapa tulisan “Sapaan untuk user” tidak pernah muncul?

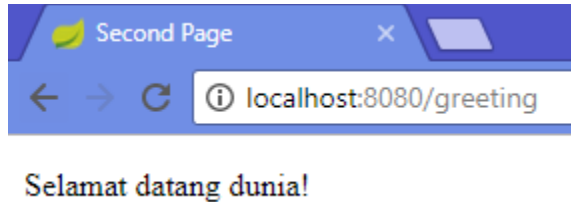
**Jawaban:** Penggunaan *thymeleaf* yang disimbolkan dengan *th:text* lah yang menyebabkan “Sapaan untuk user” ter-replace/ di *overwrite* dengan text yang ada pada atribut *th:text* di dalam tag *<p>*

## Latihan Path Variable:

1. Akses **localhost:8080/greeting/**

**Pertanyaan:** Apa hasilnya?

**Jawaban:** menghasilkan text selamat datang dengan default variable nama adalah 'dunia'

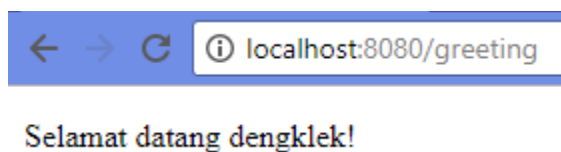


Ubah method `greetingPath` menjadi seperti berikut

```
@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";
}
```

```
@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";
}
```

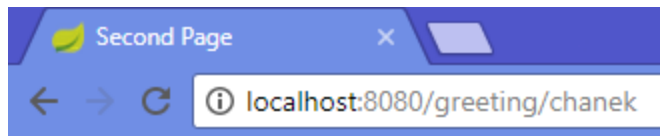
Akses **localhost:8080/greeting/**



**Pertanyaan:** Apa hasilnya?

**Jawaban:** Menampilkan “Selamat datang dengklek!” karena path variable name tidak terdapat dalam URL, sehingga valuenya menjadi null atau tidak present, jadi akan masuk ke else yaitu men-set attribute name dalam model menjadi kata “dengklek”.

Akses **localhost:8080/greeting/chanek**



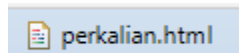
Selamat datang chanek!

**Pertanyaan:** Apa hasilnya?

**Jawaban:** Menampilkan text Selamat datang dengan variable nama 'chanek' karena ketika diakses /greeting/chanek, maka variable name akan terisi dengan value chanek sehingga name.isPresent() akan mengembalikan nilai true dan akan men-set attribute name dalam model menjadi "chanek"

## Latihan

1. Buatlah satu halaman baru untuk melakukan operasi perkalian.



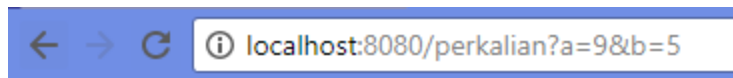
2. Halaman ini menerima 2 buah parameter dari query string yaitu a dan b.

```
@RequestMapping("/perkalian")
public String multiplication(@RequestParam(value = "a", required = false, defaultValue = "0") int a,
    @RequestParam(value = "b", required = false, defaultValue = "0") int b, Model model) {
    model.addAttribute("a_display", a);
    model.addAttribute("b_display", b);
    int c = a * b;
    model.addAttribute("c_display", c);

    return "perkalian";
}
```

3. Pada view tampilkan hasil perkalian a dan b.

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>Third Page</title>
5   </head>
6   <body>
7     <h1>Perkalian</h1>
8     <h2 th:text="${a_display} + ' x ' + ${b_display} + ' = ' + ${c_display}"></h2>
9   </body>
10 </html>
```



## Perkalian

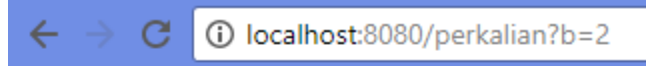
$$9 \times 5 = 45$$

4. Jika a atau b tidak ada, nilai defaultnya 0.

- Jika a tidak ada

```
@RequestParam(value = "a", required = false, defaultValue = "0") int a,
```

Output:



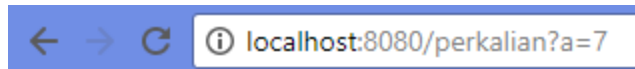
## Perkalian

$$0 \times 2 = 0$$

- Jika b tidak ada

```
@RequestParam(value = "b", required = false, defaultValue = "0") int b,
```

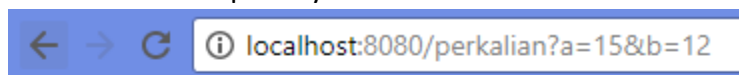
Output:



## Perkalian

$$7 \times 0 = 0$$

5. Contoh hasil tampilannya



## Perkalian

$$15 \times 12 = 180$$