

**Latihan Project Hello World**

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

**Pertanyaan:**

Apakah compile error?

Ya, Terdapat **"Whitelabel Error Page"**. Request HTTP pada path `/hello`, sedangkan `@RequestMapping("/hello123")` jadi tidak ada method apapun yang dipanggil.



## Whitelabel Error Page

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

Wed Sep 13 23:08:12 ICT 2017

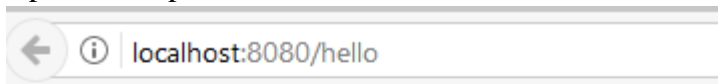
There was an unexpected error (type=Not Found, status=404).

No message available

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

**Pertanyaan:**

Apakah compile error? Tidak. Muncul **"Hello, world!"**



## Hello, world!

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

Masih muncul karena method `hello()` yang dipanggil jika ada request path `/hello`

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

**Pertanyaan:**

Apakah compile error? Ya, Terdapat **"Whitelabel Error Page"** karena `resquestmapping hello123` tidak ada.



## Whitelabel Error Page

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

Wed Sep 13 23:18:30 ICT 2017

There was an unexpected error (type=Internal Server Error, status=500).

Error resolving template `"hello123"`, template might not exist or might not be accessible by any of the configured Template Resolvers

**Pertanyaan:**

Menandakan apakah String yang di-return tersebut?  
Menandakan method yang dijalankan dan direturn hasilnya.

**Latihan Request Parameter**

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

**Pertanyaan:** apakah hasilnya? Terdapat **“Whitelabel Error Page”** karena tidak ada `@RequestMapping("/hello/greeting")`, tidak ada method yang dijalankan.



## Whitelabel Error Page

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

Wed Sep 13 23:18:30 ICT 2017

There was an unexpected error (type=Internal Server Error, status=500).

Error resolving template "hello123", template might not exist or might not be accessible by any of the configured Template Resolvers

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

**Pertanyaan:** Apakah hasilnya? Terdapat **“Whitelabel Error Page”** karena merequest parameter sedangkan tidak ada nilai yang dimasukkan di http request.



## Whitelabel Error Page

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

Wed Sep 13 23:25:05 ICT 2017

There was an unexpected error (type=Bad Request, status=400).

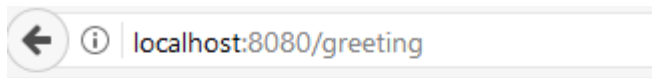
Required String parameter 'name' is not present

Ubah header method greeting menjadi seperti berikut

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

**Pertanyaan:** Apakah hasilnya? **“Selamat datang null!”**

Tidak error karena requirednya diubah false, walaupun tidak ada nilai yang dimasukkan sebagai parameter tidak akan error tapi nilainya dianggap null.



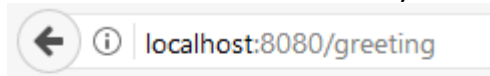
Selamat datang null!

- Ubah header method greeting menjadi seperti berikut

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

buka **localhost:8080/greeting**.

**Pertanyaan:** apakah hasilnya? **"Selamat datang dunia"** Karena tidak ada parameter yang dimasukkan dan nilai defaultnya "dunia" maka yang ditampilkan sesuai nilai default.



Selamat datang dunia!

- 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?

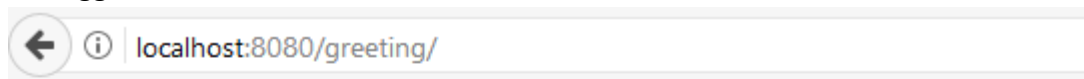
Karena th:text dari Thymeleaf akan overwrite seluruh teks yang ada pada tag dimana atribut tersebut ditempatkan.

### Latihan Path Variable:

- Akses **localhost:8080/greeting/**

**Pertanyaan:** Apa hasilnya? Terdapat **"Whitelabel Error Page"**

ketika request path /greeting/ maka yang dipanggil adalah @RequestMapping ( value = { "/greeting" , "greeting/{name}" }) yang mengharuskan ada parameter, sehingga error.



## Whitelabel Error Page

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

Wed Sep 13 23:43:08 ICT 2017

There was an unexpected error (type=Bad Request, status=400).

Required String parameter 'name' is not present

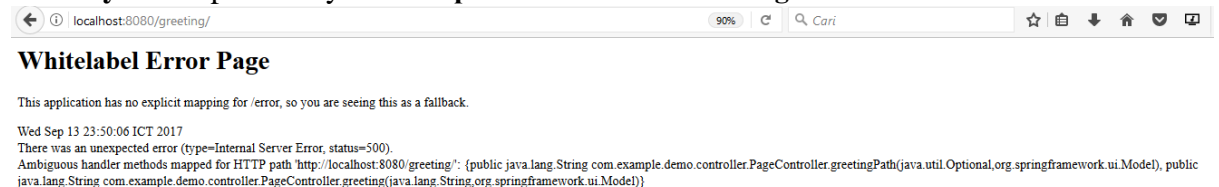
- Ubah method greetingPath menjadi seperti berikut

```
@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/**

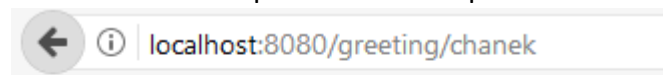
**Pertanyaan:** Apa hasilnya? Terdapat “Whitelabel Error Page”



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

**Pertanyaan:** Apa hasilnya? “Selamat datang chanek!”

Nama chanek dari parameter disimpan di model dan dapat ditampilkan di html



Selamat datang chanek!

## Latihan

1. Buatlah satu halaman baru untuk melakukan operasi perkalian.  
Nama file : perkalian.html
2. Halaman ini menerima **2 buah** parameter dari query string yaitu a dan b.

```
public String perkalian(@RequestParam(value= "a", required = false ,
defaultValue = "0") String a, @RequestParam(value= "b", required = false ,
defaultValue = "0") String b,Model model)
```

Menggunakan @RequestParam untuk mengambil parameter 2 string a dan b.  
Nilai a dan b disimpan ke dalam model a dan b agar dapat diambil nilainya di html untuk ditampilkan.

```
model.addAttribute("a", a);
model.addAttribute("b", b);
```

Variabel a dan b merupakan String, untuk memperoleh hasil perkalian keduanya harus diubah menjadi integer.

```
int numA = Integer.parseInt(a);
int numB = Integer.parseInt(b);
```

Lalu hasil perkalian disimpan di variabel dan model result

```
int result = numA * numB;
model.addAttribute("result", result);
```

3. Pada view tampilkan hasil perkalian a dan b.

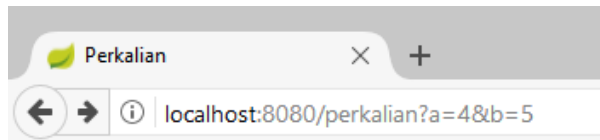
```
<p th:text="${a} + ' X ' + ${b} + ' = ' + ${result}"> </p>
```

Variabel result merupakan hasil operasi perkalian yang telah di return dari method perkalian di Controller.

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

Karena a dan b bisa tidak terisi, maka required = false agar tidak error ketika tidak ada parameter yang dimasukkan. Jika a dan b tidak terisi, maka nilai keduanya 0. defaultValue = "0"

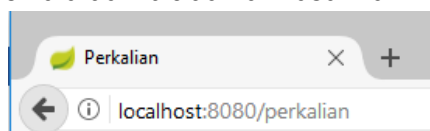
5. Contoh hasil tampilannya



## Perkalian

4 X 5 = 20

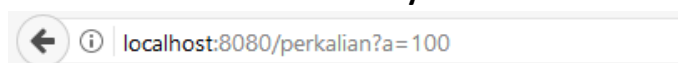
Jika a dan b tidak dimasukkan nilainya



## Perkalian

0 X 0 = 0

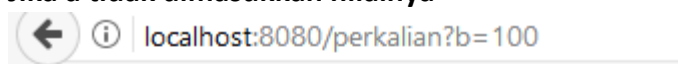
Jika b tidak dimasukkan nilainya



## Perkalian

100 X 0 = 0

Jika a tidak dimasukkan nilainya



## Perkalian

0 X 100 = 0

**Apa yang dipelajari:**

- Bagaimana SpringBoot bekerja
- Bagaimana cara menggunakan SpringBoot
- Mengerti konsep MVC
- Model menangani query atau database. Isi dari model merupakan bagian (fungsi-fungsi) yang berhubungan langsung dengan database untuk memanipulasi data seperti memasukkan data, pembaruan data, hapus data, dan lain-lain, tetapi model tidak dapat berhubungan langsung dengan bagian view.
- View yang mengatur tampilan website. Bagian view biasanya berupa file template HTML, yang diatur oleh *controller*. Bagian ini tidak memiliki akses langsung terhadap bagian model, tetapi berhubungan langsung dengan controller. View berfungsi untuk menerima dan merepresentasikan data kepada pengguna. Jadi bisa di katakan bahwa view merupakan halaman website tersebut.
- Controller merupakan bagian yang menjembatani model dan view. Controller berfungsi untuk menerima request dan data dari user kemudian menentukan apa yang akan diproses atau dirun oleh aplikasi. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu data dan mengirimkannya ke halaman web.