

## Tutorial 2

### Menggunakan Routing dan Controller dalam Project Spring Boot

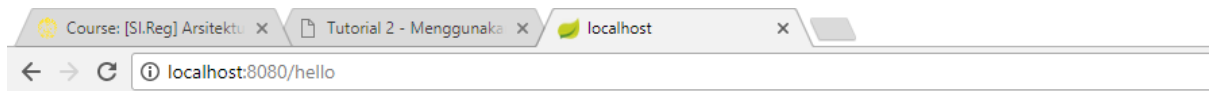
#### Latihan Project Hello World

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

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

**Jawab:** Tidak terjadi compile error, tetapi terjadi error not found ketika program dirun kembali, karena tidak terdapat method yang menangani request HTTP pada path /hello karena RequestMapping sudah diganti menjadi /hello123, dan page hello pun tidak dapat ditampilkan.

**Screenshot:**



### Whitelabel Error Page

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

Wed Sep 13 08:31:07 ICT 2017

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

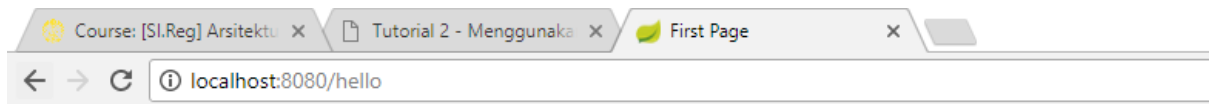
No message available

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

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

**Jawab:** Tidak terjadi compile error dan ketika dirun kembali page hello kembali muncul / dapat ditampilkan.

**Screenshot:**



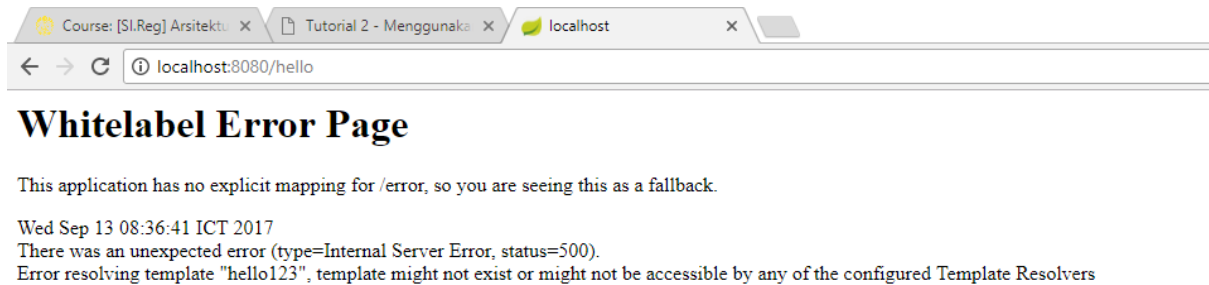
### Hello World

3. Ganti string return type menjadi 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?

**Jawab:** Tidak terjadi compile error tetapi page hello tidak muncul, melainkan terdapat error internal server karena template page hello123 yang dipanggil method yang menangani request HTTP path /hello tidak ada.

**Screenshot:**



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

**Jawab:** Page template yang ingin ditampilkan ke layar.

### Latihan Request Parameter

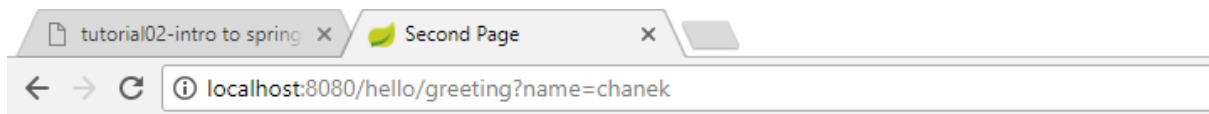
1. Ubah nilai anotasi RequestMapping dari "/greeting" menjadi "/hello/greeting"

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

**Pertanyaan:** apakah hasilnya?

**Jawab:** Page greeting ditampilkan dengan parameter name yang diambil dari url.

**Screenshot:**



Selamat datang chanek!

2. Akses localhost: 8080/greeting

**Pertanyaan:** Apakah hasilnya?

**Jawab:** Terjadi error bad request karena tidak terdapat parameter name pada url yang harus diambil oleh method yang menangani RequestMapping /greeting.

**Screenshot:**



Ubah header method greeting menjadi seperti berikut

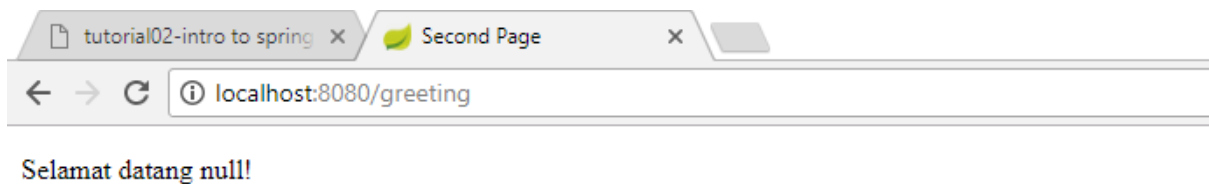
```
public String greeting (@RequestParam(value = "name", required = false) String  
name, Model model)
```

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

**Pertanyaan:** Apakah hasilnya?

**Jawab:** Ditampilkan page greeting dengan parameter name = null karena tidak terdapat parameter name pada url, tetapi tidak error karena parameter name pada header method greeting sudah diubah menjadi not required.

**Screenshot:**



3. Ubah header method greeting menjadi seperti berikut

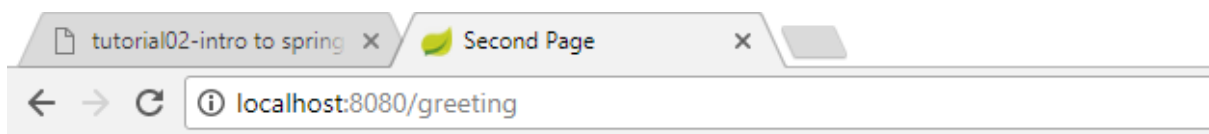
```
public String greeting (@RequestParam(value = "name", required = false,  
defaultValue = "dunia") String name, Model model)
```

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

**Pertanyaan:** apakah hasilnya?

**Jawab:** Ditampilkan page greeting dengan parameter name = dunia karena ketika program tidak menemukan parameter name pada url maka nilai default parameter name tersebut adalah dunia.

**Screenshot:**



Selamat datang dunia!

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

**Jawaban:** Karena pada code greeting.html tag p kita menggunakan atribut thymeleaf yaitu th:text. Atribut tersebut akan men-set body dari tag p tersebut dengan yang didefinisikan pada atribut th:text tersebut dan mensubstitusi teks “Sapaan untuk user” yang ada di body tag p tersebut.

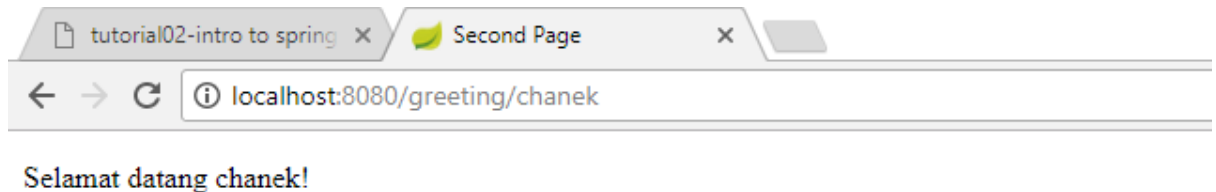
(Sumber: <http://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html>)

### Path Variables

1. Tambahkan method greetingPath pada PageController
2. Stop Spring Boot yang sedang berjalan, run kembali, dan buka localhost:8080/greeting/chanek apakah hasilnya?

**Jawab:** Ditampilkan page greeting dengan paramater name yang diambil dari path variable pada url.

**Screenshot:**



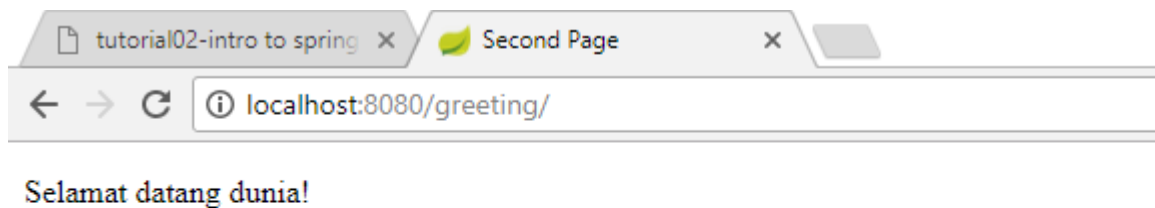
### Latihan Path Variable

1. Akses localhost:8080/greeting/

**Pertanyaan:** Apa hasilnya?

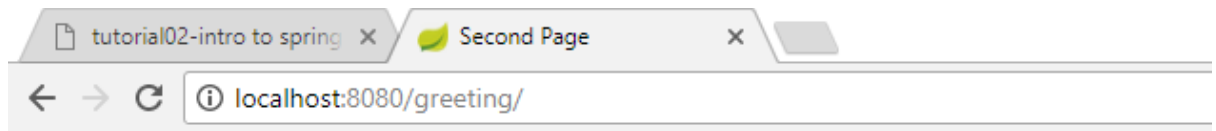
**Jawab1:** Jika method greeting yang sama-sama menangani RequestMapping /greeting tidak dihapus, maka ditampilkan page greeting dengan default parameter name yaitu dunia.

**Screenshot1:**



**Jawab2:** Jika method greeting yang sama-sama menangani RequestMapping /greeting dihapus, maka terjadi error not found karena tidak ada RequestMapping yang menangani url /greeting/

**Screenshot2:**



## Whitelabel Error Page

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

Fri Sep 15 19:10:44 ICT 2017

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

No message available

Ubah method greetingPath menjadi seperti berikut

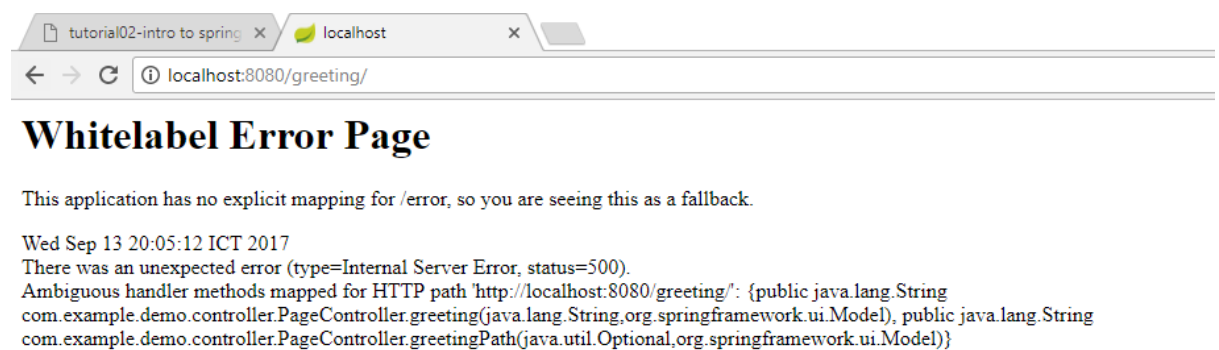
```
@RequestMapping(value = {"/greeting", "greeting/{name}"})
public String greetingPath(@PathVariable Optional<String> name, Model model) {
    if (name.isPresent()) {
        model.addAttribute("name", name.get());
    } else {
        model.addAttribute("name", "dengklek");
    }
    return "greeting";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

Akses localhost:8080/greeting/

**Pertanyaan:** Apa hasilnya?

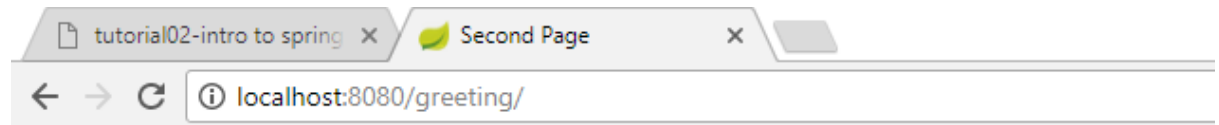
**Jawab1:** Jika method greeting yang sama-sama menangani RequestMapping /greeting tidak dihapus, maka akan ditampilkan error internal server karena program tidak dapat menentukan untuk menggunakan method yang mana.

**Screenshot1:**



**Jawab2:** Jika method greeting yang sama-sama menangani RequestMapping /greeting dihapus, maka akan ditampilkan page greeting dengan parameter name default yaitu dengklek.

**Screenshot2:**

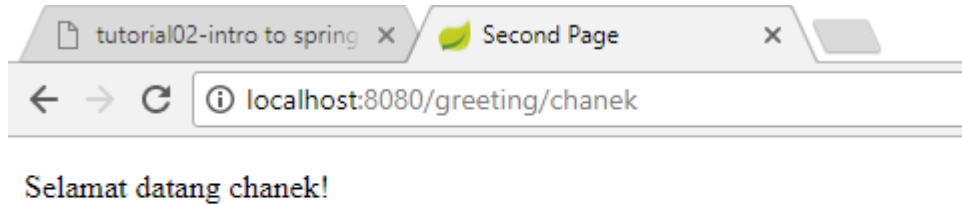


Selamat datang dengklek!

Akses localhost:8080/greeting/chanek

**Pertanyaan:** Apa hasilnya?

**Jawab:** Ditampilkan page greeting dengan parameter name yang diambil dari path variable pada url.



**Yang saya pelajari pada tutorial ini :**

Pada tutorial ini saya belajar tentang bagaimana page controller merespon pada HTTP request dan mengembalikan page template yang diinginkan. @RequestMapping yang didefinisikan di atas method menandakan bahwa jika ada request HTTP pada path yang didefinisikan pada RequestMapping, maka method tersebut akan terpanggil. Kemudian untuk menampilkan page template yang diinginkan, method tersebut akan mereturn sebuah string yang mana merupakan page template yang ingin ditampilkan.

Saya juga belajar tentang passing suatu variable dari url ke view menggunakan request parameter dan path variable. Untuk passing sebuah parameter ke view kita dapat menggunakan query string yang diambil dari url dengan menambahkan RequestParam pada header method dan passing ke view dengan Objek Model yang didefinisikan di header method juga.

Kemudian, cara lainnya untuk passing data dari url ke view adalah menggunakan path variable. Dengan menggunakan path variable, kita perlu memodifikasi RequestMapping sesuai dengan variable apa dari url yang ingin dipass, kemudian pada header method tetap harus didefinisikan objek yang ingin dipass.