

Penjelasan apa saja yang telah dipelajari.

Pada tutorial kali ini saya telah mempelajari bagaimana cara melakukan routing menggunakan spring dan juga bagaimana meng GET variable yang telah dipassing pada url address.

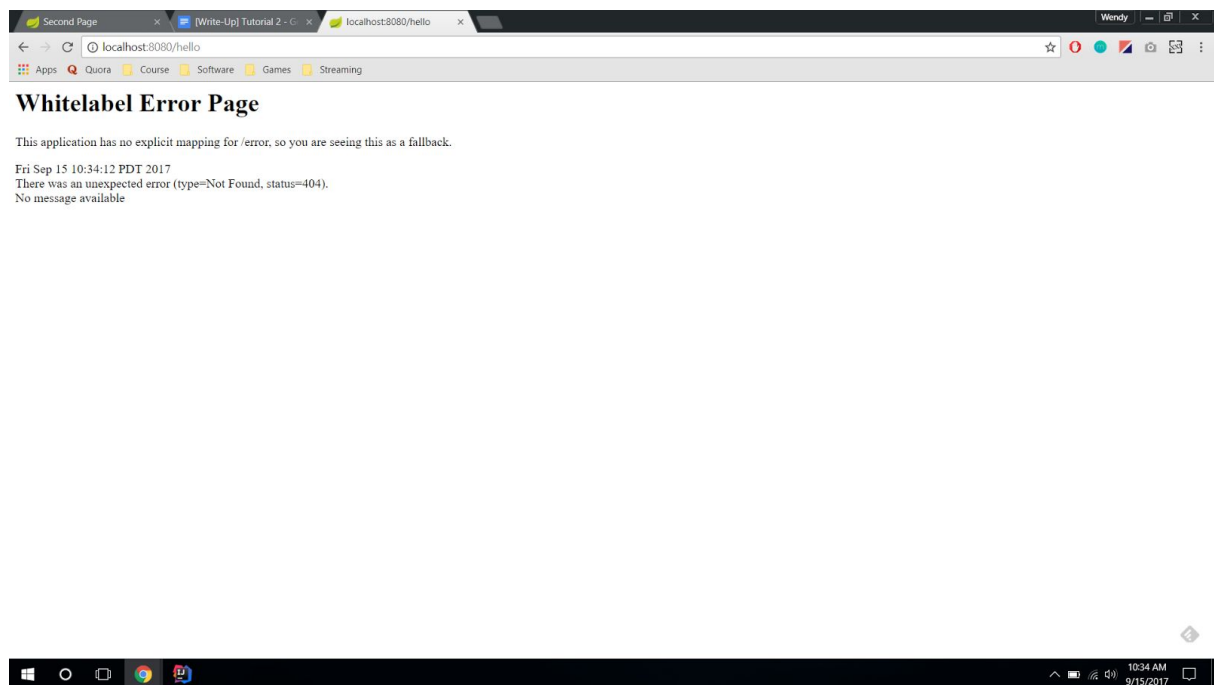
Jawaban Pertanyaan

1. Ganti baris tersebut 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?

Kembalikan RequestMapping menjadi /hello

**Tidak *compile error*. Namun ketika localhost:8080/hello dijalankan maka akan menampilkan halaman *white label* yang menunjukkan bahwa routing yang dituju tidak ada tujuannya (tidak kemana-mana/tidak menuju fungsi pada controller). Dan ketika *RequestMapping* telah dikembalikan ke /hello maka akan menuju halaman yang telah disediakan sesuai dengan functionnya.**

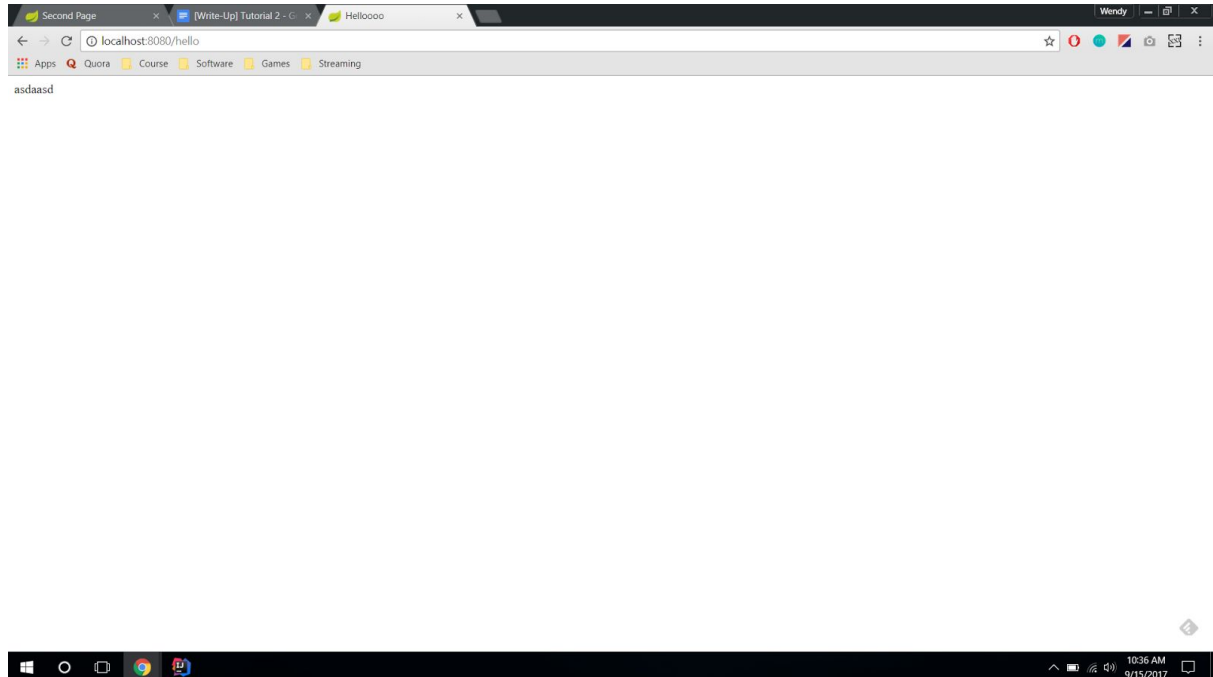


2. Ganti nama method `index()` dengan nama method `hello()`

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?

Kembalikan nama method menjadi `index()`

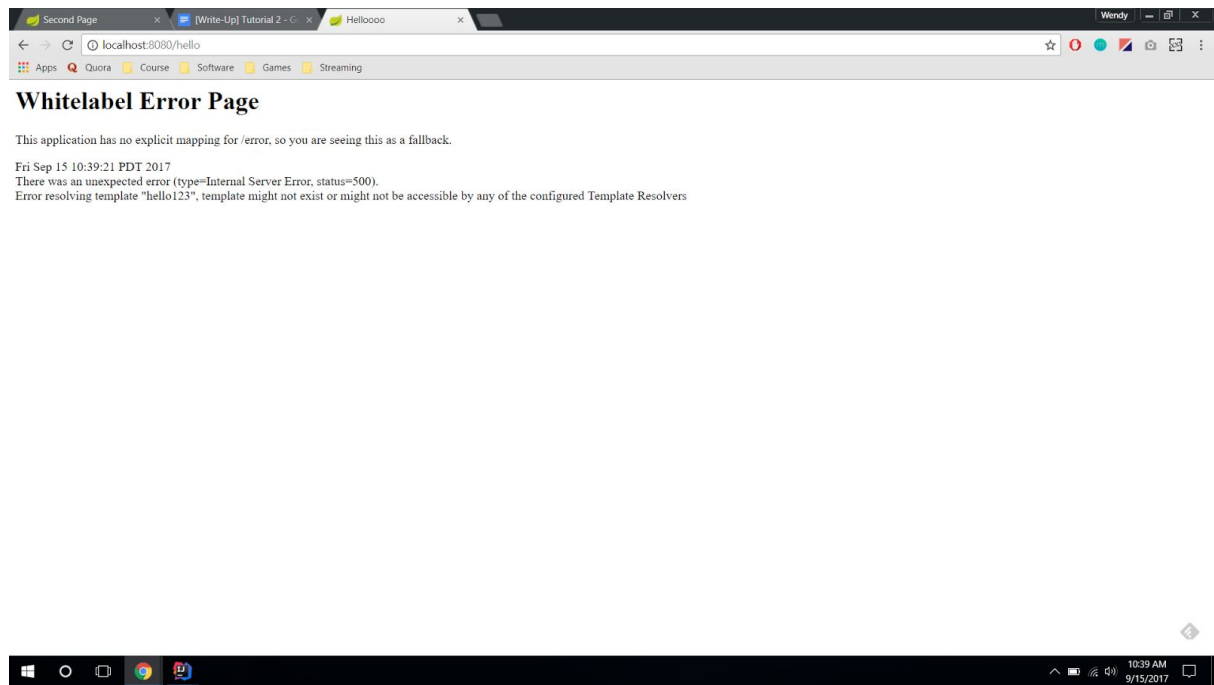
Tidak *compile error*. Dan tidak berubah apa-apa.



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?

**Tidak *compile error*, namun ketika berjalan ia menampilkan halaman *whitelabel*. Namun kondisinya berbeda dari sebelumnya dikarenakan walau ia telah menemukan alamat *mapping* nya (controller) namun ia tidak merujuk ke view yang ia tuju karena tidak ada yang namanya hello123.html. Jika diubah menjadi**



Kembalikan return type menjadi “hello”

Pertanyaan: Menandakan apakah String yang di-return tersebut?

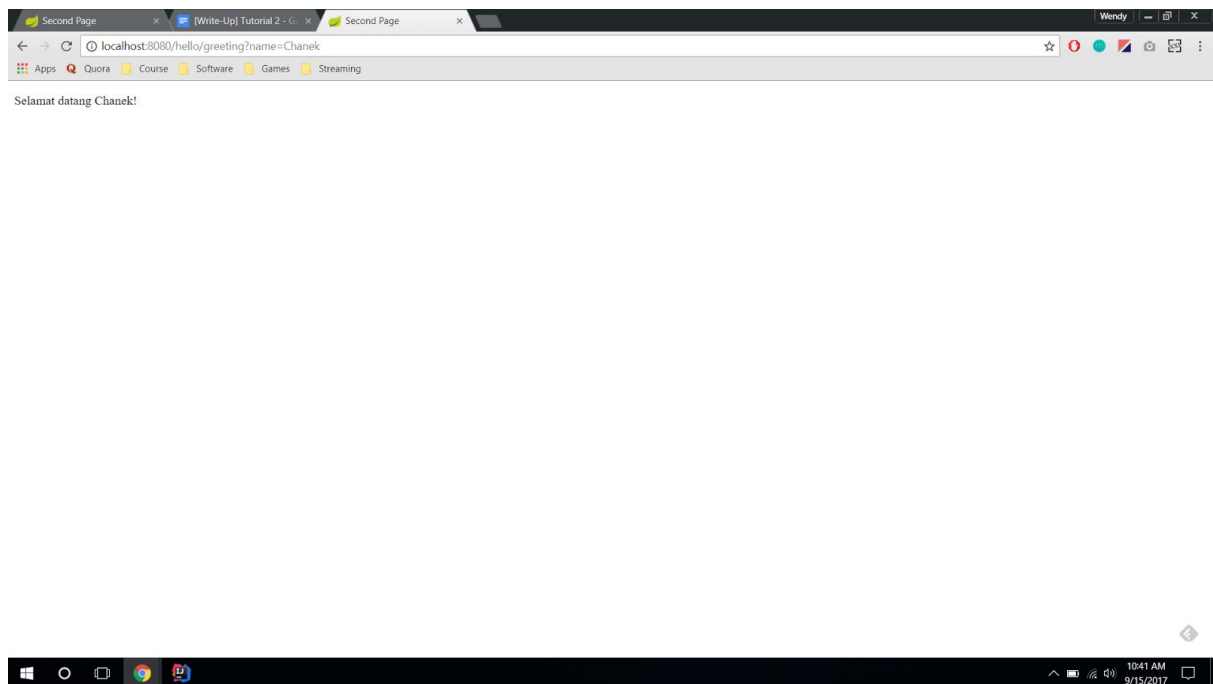
**String yang di return merupakan alamat html yang dituju menuju template .html nya**

## Request Parameter

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

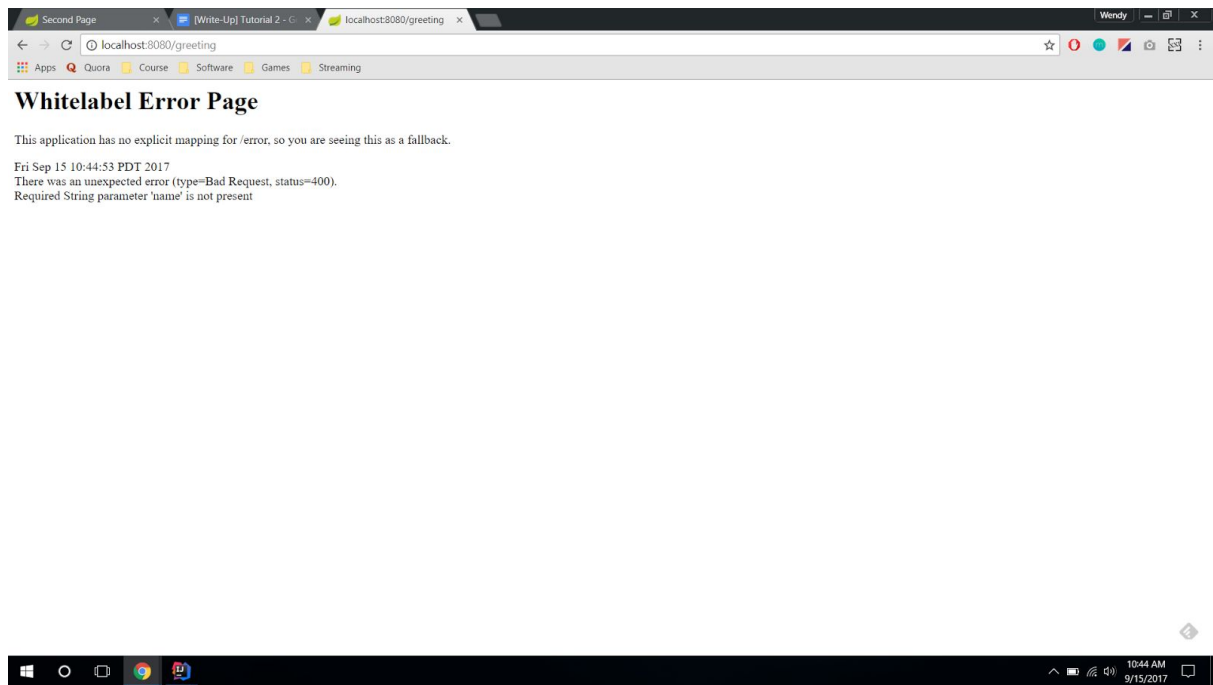
Pertanyaan: apakah hasilnya? Kembalikan request map menjadi @RequestMapping("/greeting")

**Hasilnya tetap seperti sebelumnya dimana akan menghasilkan Selamat datang Chanek. Namun jika request map diubah menjadi @RequestMapping("/greeting") maka hasilnya tidak akan muncul**



2. Akses localhost: 8080/greeting

Pertanyaan: Apakah hasilnya? **Hasilnya error dikarenakan ia tidak menemukan type nya ("name") atau bisa dibilang GET dengan type name pada url.**

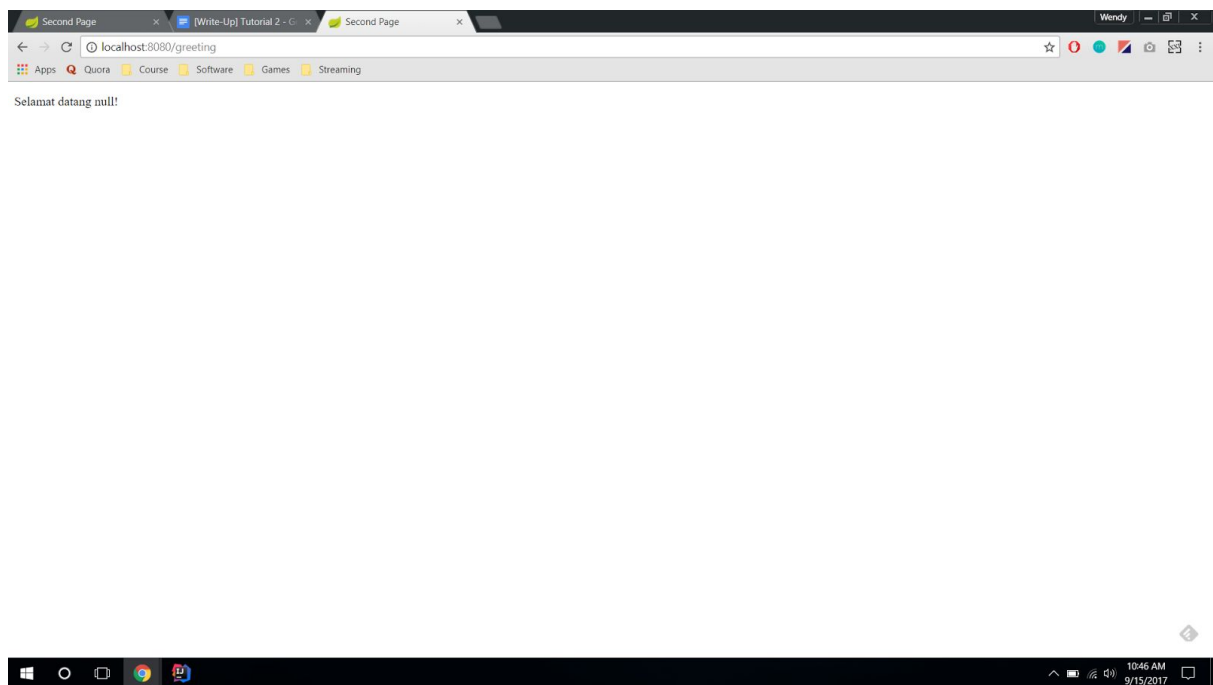


Ubah header method greeting menjadi seperti berikut:

```
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?

**Maka program akan tetap berjalan namun dengan default namanya null.**

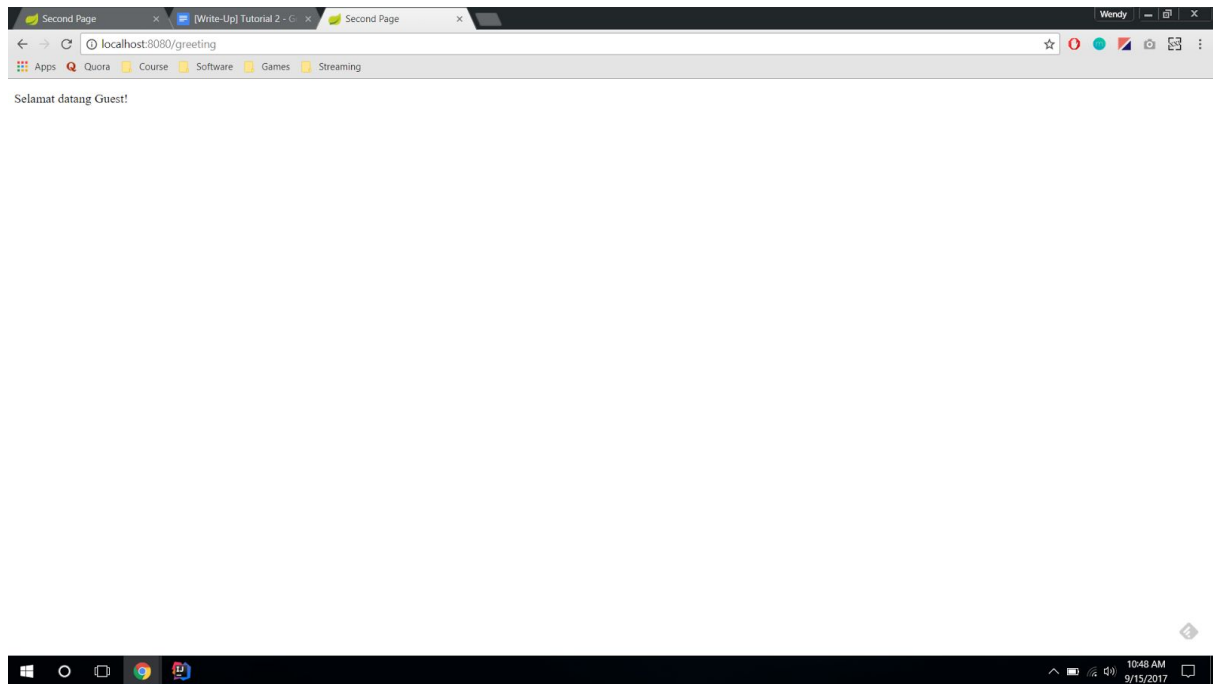


3. Ubah header method greeting menjadi seperti berikut

```
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?

**Maka program akan tetap berjalan namun dengan default namanya dunia.**



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

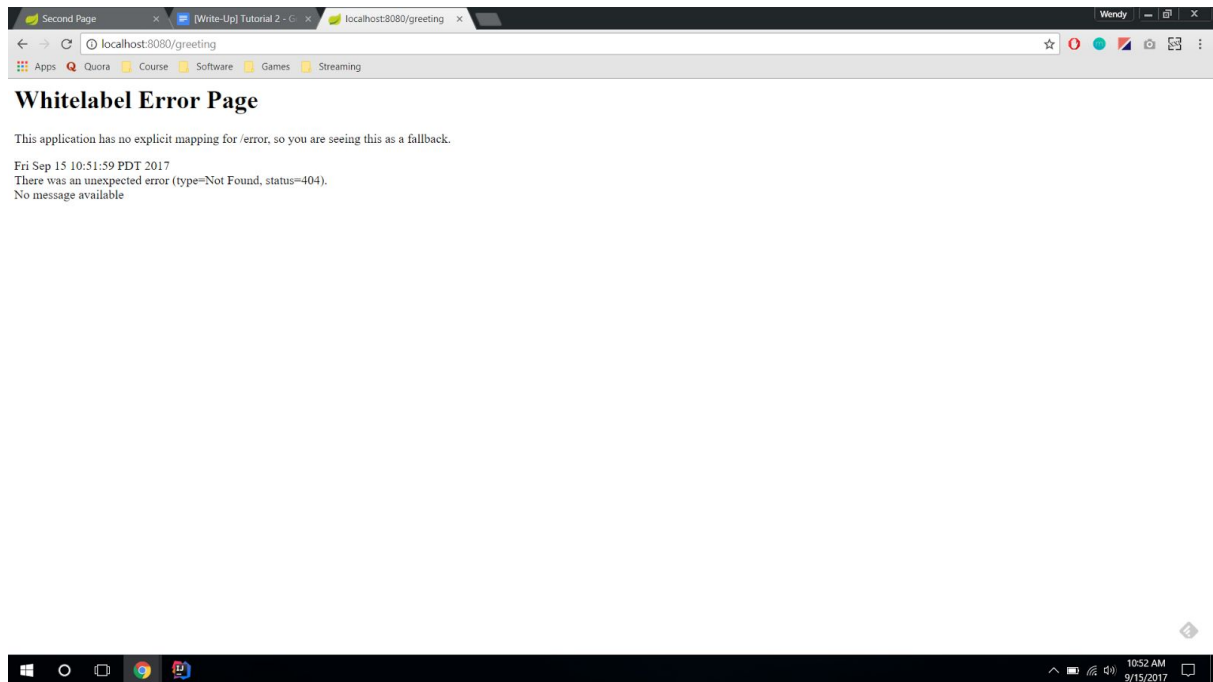
Pertanyaan: Mengapa tulisan “Sapaan untuk user” tidak pernah muncul?

**Hal ini dikarenakan memang sistemika Thymeleaf yang meng override String didalamnya.**

## Path Variable

1. Akses localhost:8080/greeting/ Pertanyaan: Apa hasilnya? Ubah method greetingPath menjadi seperti berikut

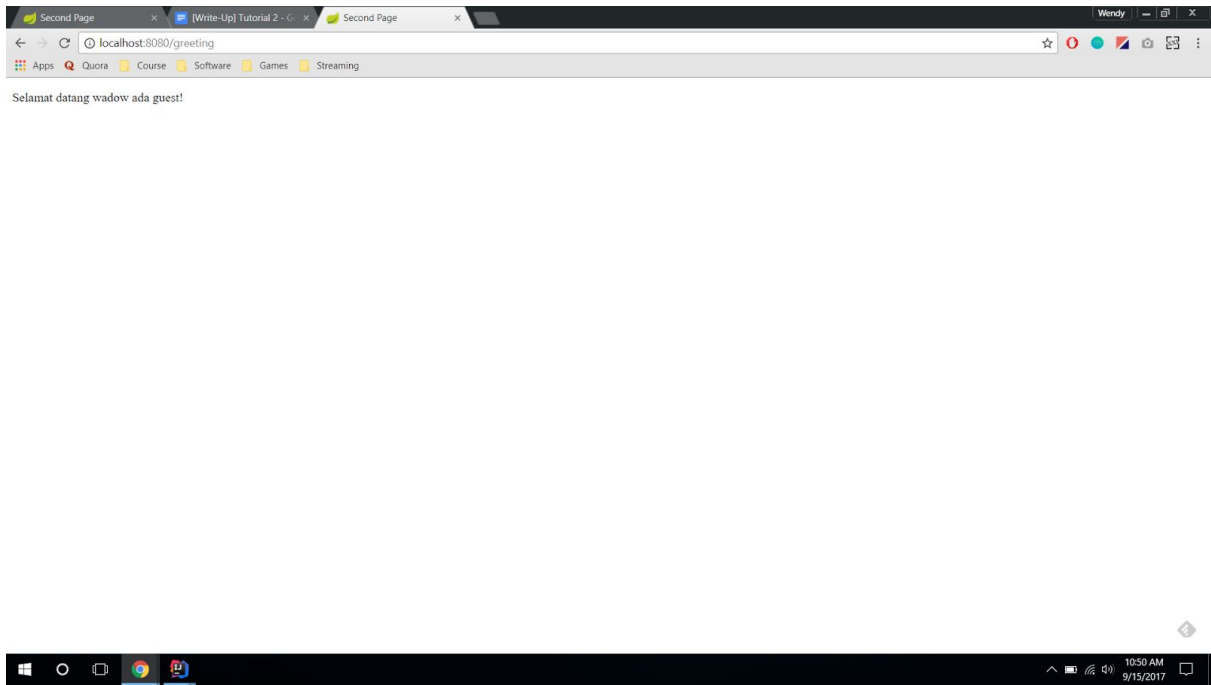
**Hasilnya Error dikarenakan tidak ada yang handle apabila tidak ada parameternya**



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

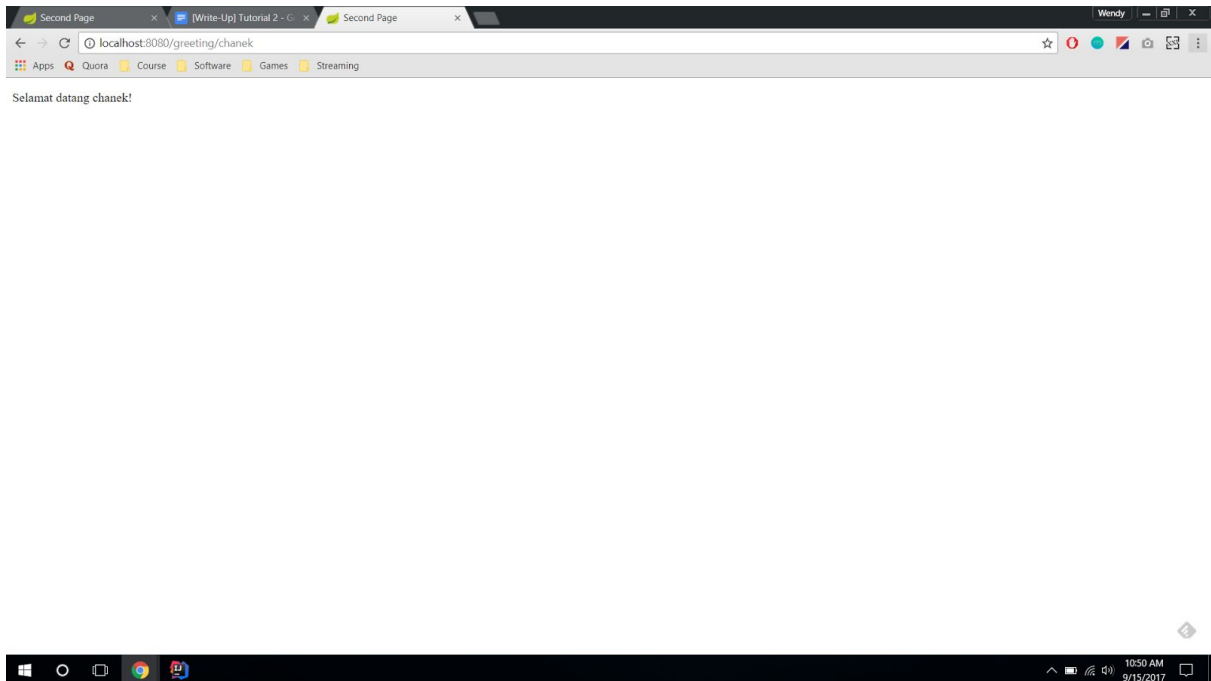
Akses localhost:8080/greeting/ Pertanyaan: Apa hasilnya?

**Program akan menghasilkan halaman dimana menyantumkan default dari valuenya. misalkan jika dilihat dari tutorial maka hasilnya selamat datang dengklek**



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

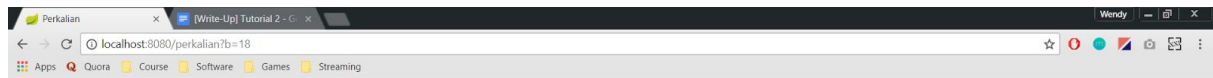
**Program akan menghasilkan halaman dimana menyantumkan sesuai dengan GET valuenya**





## Perkalian

1. Buatlah satu halaman baru untuk melakukan operasi perkalian.
2. Halaman ini menerima 2 buah parameter dari query string yaitu a dan b.
3. Pada view tampilkan hasil perkalian a dan b.
4. Jika a atau b tidak ada, nilai defaultnya 0.
5. Contoh hasil tampilannya

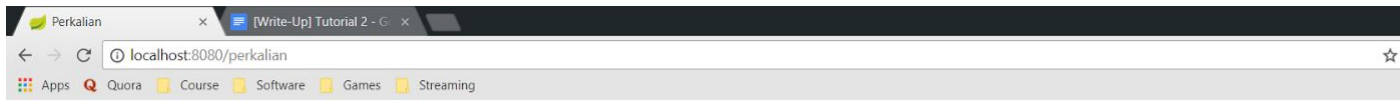


**0 X 18 = 0**



**18 X 0 = 0**





$$0 \times 0 = 0$$



$$2 \times 8 = 16$$

