

## Yang dipelajari pada tutorial ini

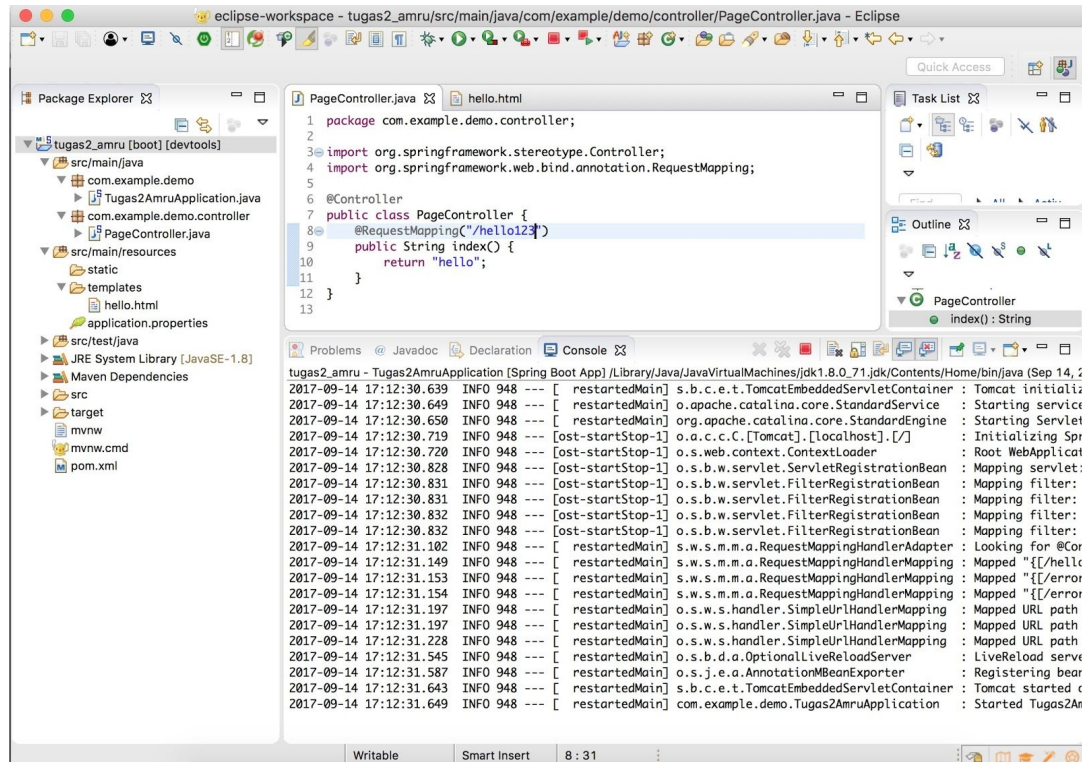
Saya mempelajari bagaimana cara membuat web application menggunakan MVC application yang mana dalam perkuliahan ini aplikasi yang dipakai adalah Spring. Dalam tutorial ini, saya mempelajari cara mengambil parameter query dan variable path.

## Latihan Project Hello World

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

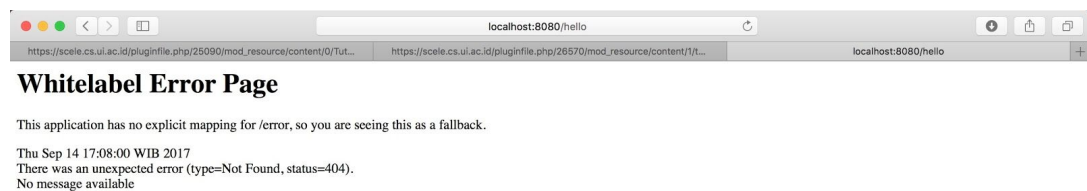
### Pertanyaan:

Apakah compile error?



Tidak terdapat compile error.

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

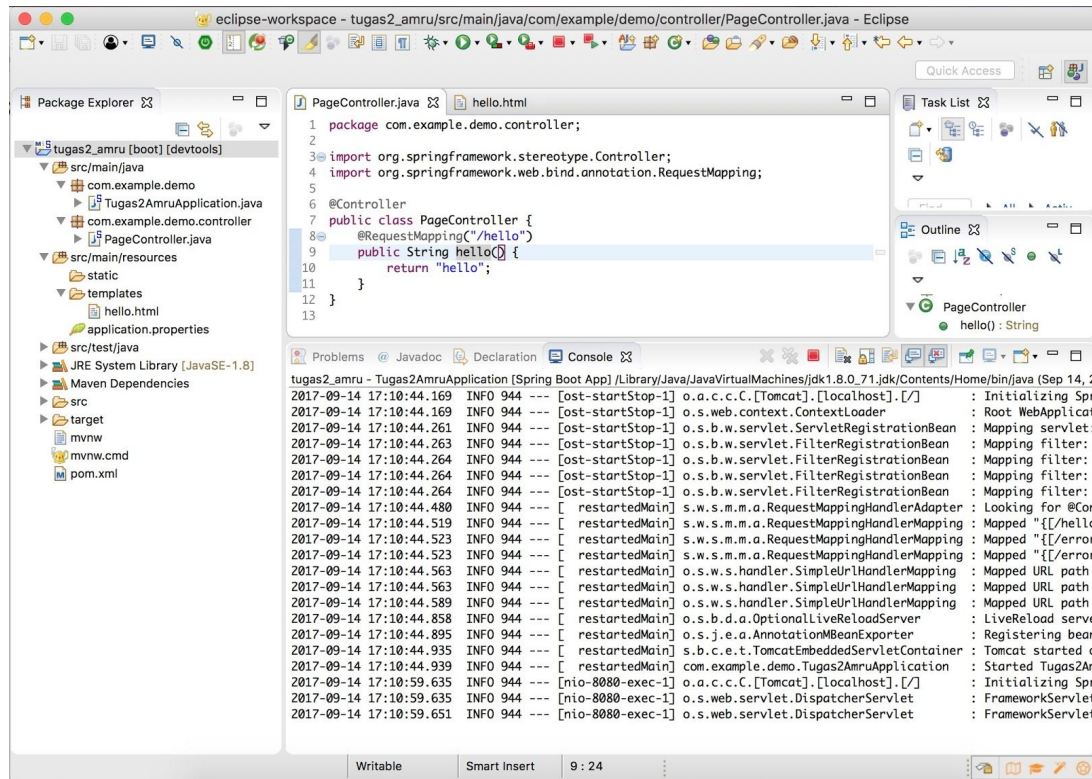


Terdapat error page.

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

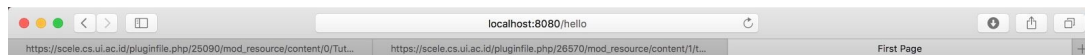
### Pertanyaan:

Apakah compile error?



Tidak terdapat compile error.

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



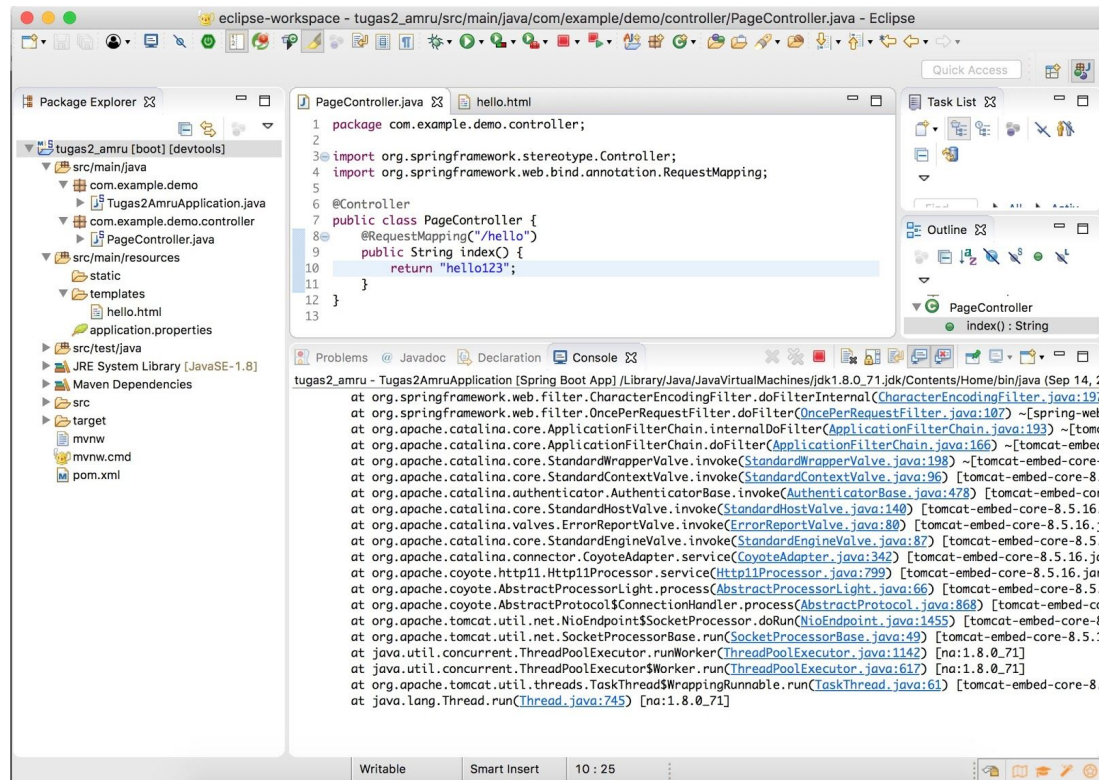
**Hello World!**

Page hello tetap muncul.

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

**Pertanyaan:**

Apakah compile error?



Tidak terjadi compile error.

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

Tidak muncul karena tidak sesuai dengan nama file html yang berada di src/main/resources/templates

Menandakan apakah String yang di-return tersebut?

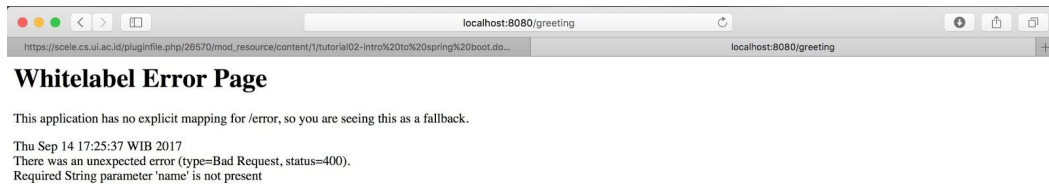
File html yang berada di src/main/resources/templates

## Latihan Request Parameter

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



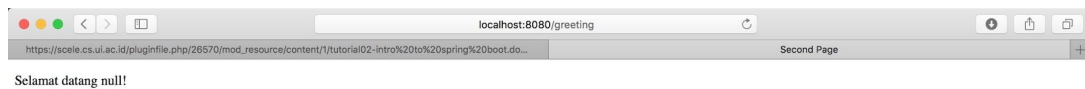
2. Akses localhost: 8080/greeting  
**Pertanyaan:** Apakah hasilnya?



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?

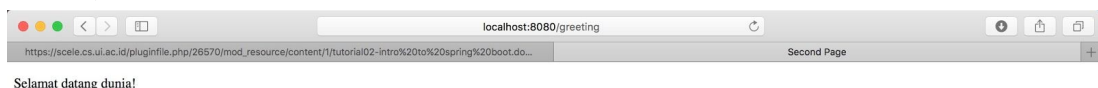


3. Ubah header method greeting menjadi seperti berikut

```
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?



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>
```

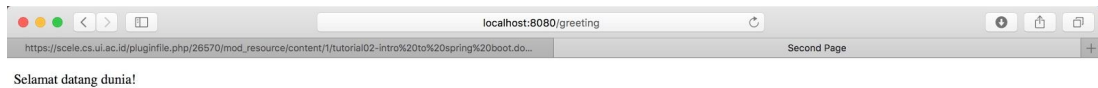
**Pertanyaan:** Mengapa tulisan “Sapaan untuk user” tidak pernah muncul?

Berdasarkan dokumentasi Thymeleaf, th:text adalah text yang akan muncul pada suatu tag. Apapun yang berada di antara opening tag dan closing tag (dalam hal ini yaitu “Sapaan untuk user”), tidak akan dimunculkan karena hanya berfungsi sebagai deskripsi tujuan tag yang digunakan.

### Latihan Path Variable:

1. Akses localhost:8080/greeting/

**Pertanyaan:** Apa hasilnya?

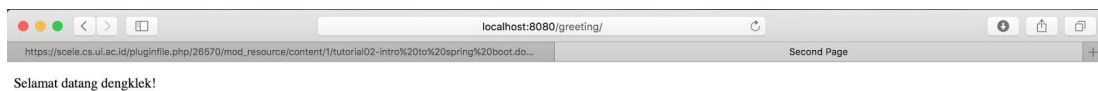


Ubah method greetingPath menjadi seperti berikut

```
@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?



Akses localhost:8080/greeting/chanek

**Pertanyaan:** Apa hasilnya?

