

Latihan Project Hello World

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

Pertanyaan:

Apakah compile error? **Tidak.**

Jika tidak, stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:

8080/hello apa yang terjadi? **Whitelabel Error Page, tipe=Not Found, status=404.**

Kembalikan RequestMapping menjadi `/hello`

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

Pertanyaan:

Apakah compile error? **Tidak.**

Jika tidak, Stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:

8080/hello apakah page hello sebelumnya masih muncul? **Ya, hasilnya "Hello, world!".**

Kembalikan nama method menjadi `index()`

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

Pertanyaan:

Apakah compile error? **Tidak.**

Jika tidak, Stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:

8080/hello apakah page hello sebelumnya masih muncul? **Tidak, hasilnya Whitelabel Error Page, tipe=Internal Server Error, status=500.**

Kembalikan return type menjadi `"hello"`

Pertanyaan:

Menandakan apakah String yang di-return tersebut? **Menandakan panggilan untuk file html yang ada di folder `/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? **"Selamat datang chanek!"**.

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

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

Pertanyaan: Apakah hasilnya? **Whitelabel Error Page, tipe=Bad Request, status=400.**

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? **"Selamat datang 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? **"Selamat datang 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>
```

Pertanyaan: Mengapa tulisan **"Sapaan untuk user"** tidak pernah muncul? **Karena dengan menggunakan atribut `th:text`, "Sapaan untuk user" akan diganti dengan isi atribut tersebut.**

Stop Spring Boot yang sedang berjalan, run kembali, dan buka **localhost:8080/greeting/chanek** apakah hasilnya? **"Selamat datang chanek!"**.

Latihan Path Variable

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

Pertanyaan: Apa hasilnya? **Whitelabel Error Page, tipe=Not Found, status=404.**

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? **"Selamat datang dengklek!"**.

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

Pertanyaan: Apa hasilnya? **"Selamat datang chanek!"**.