

Dwi Arfi Faradina Mawardi

1406623083

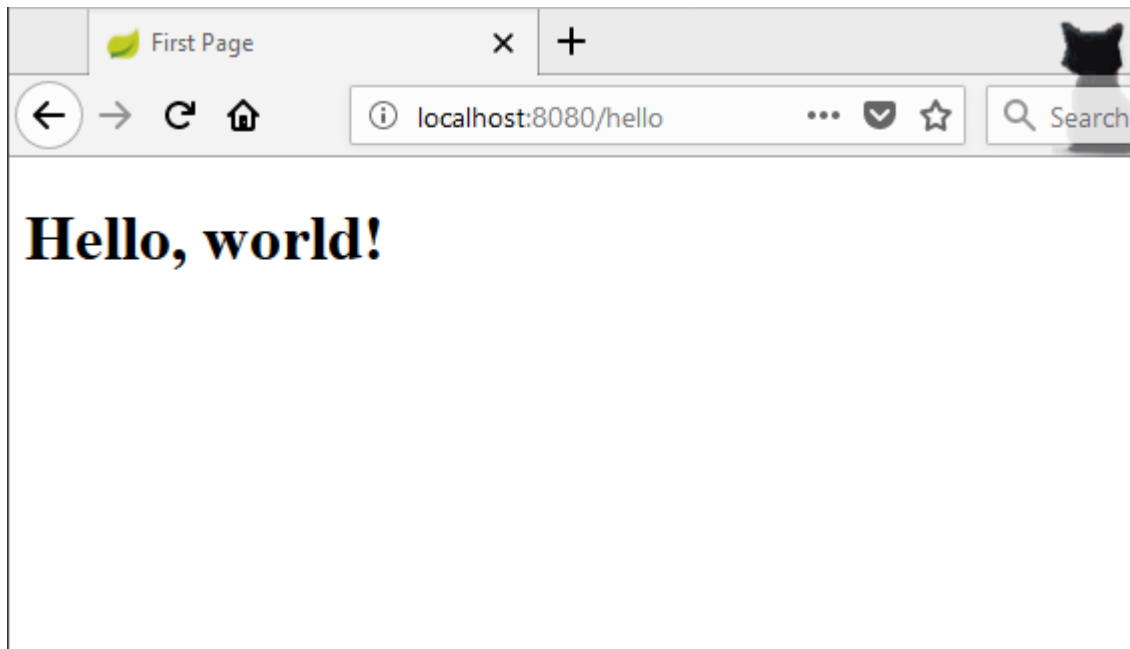
Tutorial 2 - APAP

Spring Boot merupakan sebuah web framework menggunakan Java. Framework menggunakan Java banyak digunakan untuk membuat aplikasi berskala besar. Beberapa perusahaan besar seperti Traveloka, Go-Jek, dan Blibli menggunakan Java sebagai back-end language mereka. Salah satu web framework Java yang paling populer adalah Spring.

Project Hello World

Langkah :

1. Pada Eclipse klik File > New > Spring Starter Project, tekan Next
2. Beri nama project, description, package dll.
Pada kolom name isi dengan nama project yang akan dibuat. Pada kolom group, artifact, demo, version, dan description, dan package tidak perlu diganti.
3. Untuk memudahkan pengerjaan project Spring boot, maka membutuhkan beberapa external library dalam bentuk dependency. Pilih Dependency Web, Thymeleaf, dan DevTools. Anda bisa menggunakan kolom pencarian untuk mencari dependency yang dibutuhkan.
4. Package merupakan suatu cara pengelompokkan/pengorganisasian kelas-kelas dan interface yang saling terkait menjadi suatu unit tunggal dalam library. Untuk menambahkan package Controller dapat dilakukan dengan cara klik kanan pada package com.example.demo > New > Package. Pada package com.example.demo.controller buatlah sebuah file Class dengan nama class PageController.java
5. Pada folder resources/templates tambahkan berkas hello.html
6. Klik kanan pada nama project di window Project Explorer > Run As... > Spring Boot App
7. Pada browser Anda buka localhost:8080/hello
8. Tampilan yang seharusnya Anda dapatkan:



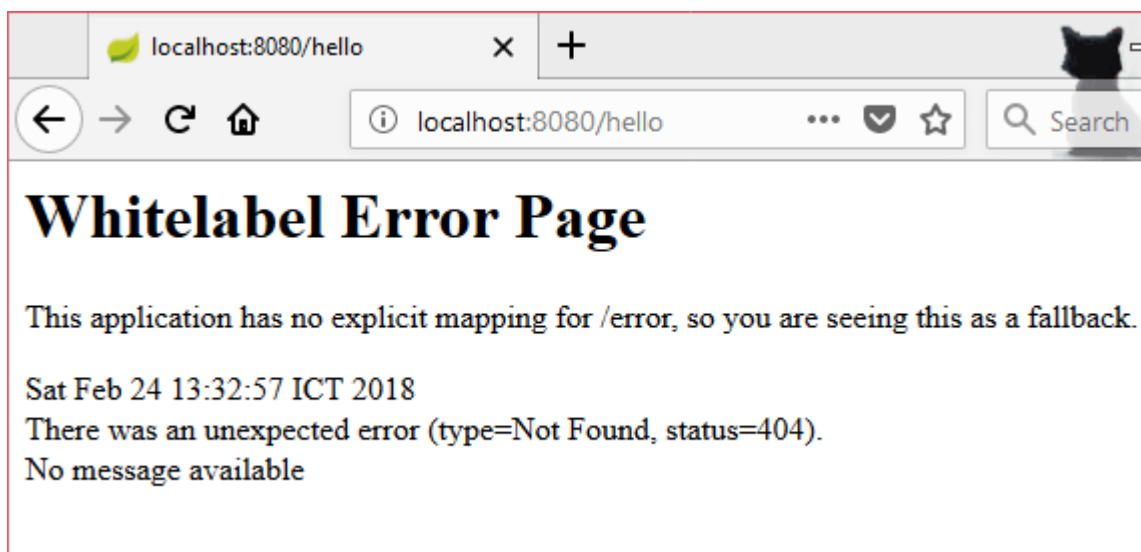
Latihan Project Hello World

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

Pertanyaan:

Apakah compile error?

Terjadi compile error

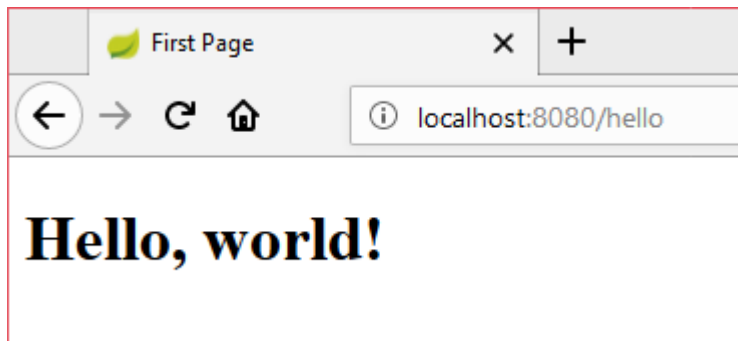


2. Ganti nama method `index()` dengan nama method `hello()`

Pertanyaan:

Apakah compile error?

tidak



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

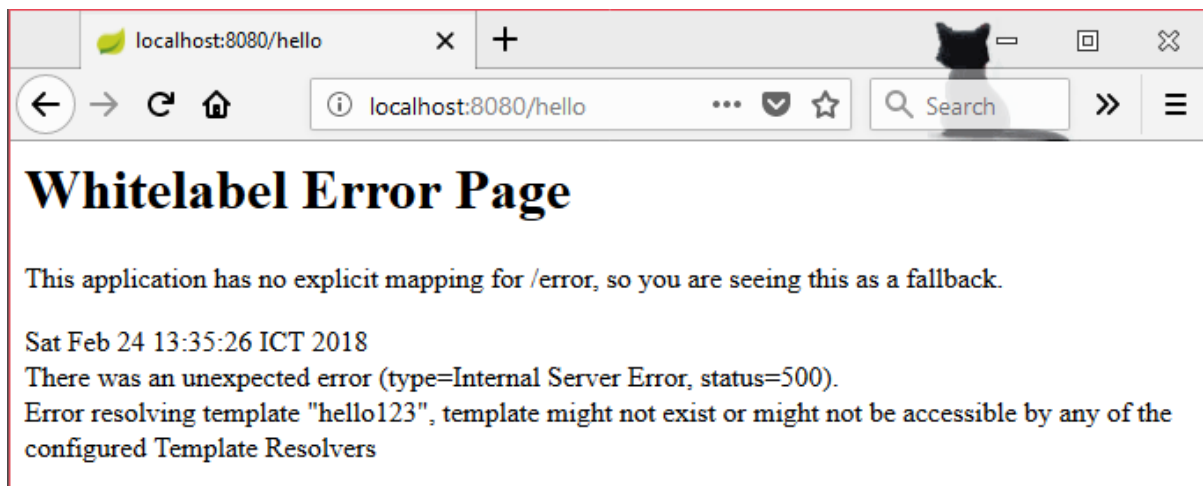
Pertanyaan:

Apakah compile error?

Terjadi error

Pertanyaan:

Menandakan apakah String yang di-return tersebut?



Request Parameter (Query String)

Langkah :

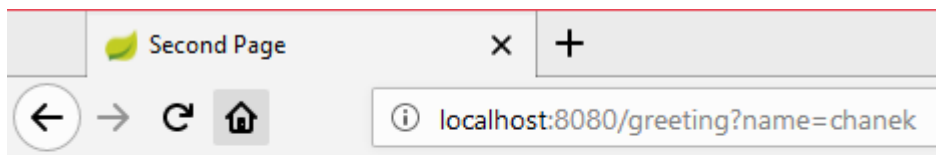
1. buat page baru dengan nama **greeting.html** pada folder templates
2. Selanjutnya pada PageController tambahkan import berikut:
`import org.springframework.web.bind.annotation.RequestMapping;`

```
import org.springframework.web.bind.annotation.RequestParam;
```

Dan tambahkan method berikut:

```
@RequestMapping("/greeting")
public String greeting (@RequestParam(value = "name") String name, Model model)
{
    model.addAttribute ("name", name);
    return "greeting";
}
```

3. Stop Spring Boot yang sedang berjalan dan run kembali (Tidak perlu melakukan ini kalau sudah menginstall dev tools dan live reload)
4. Buka **localhost:8080/greeting?name=chanek**



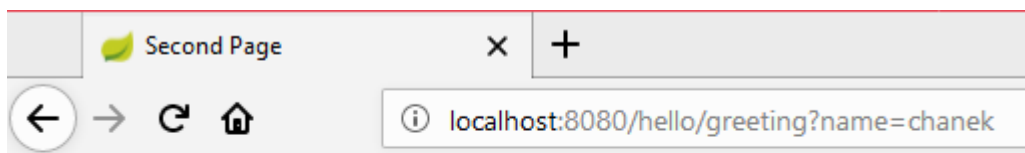
Selamat datang chanek!

Latihan Request Parameter

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

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

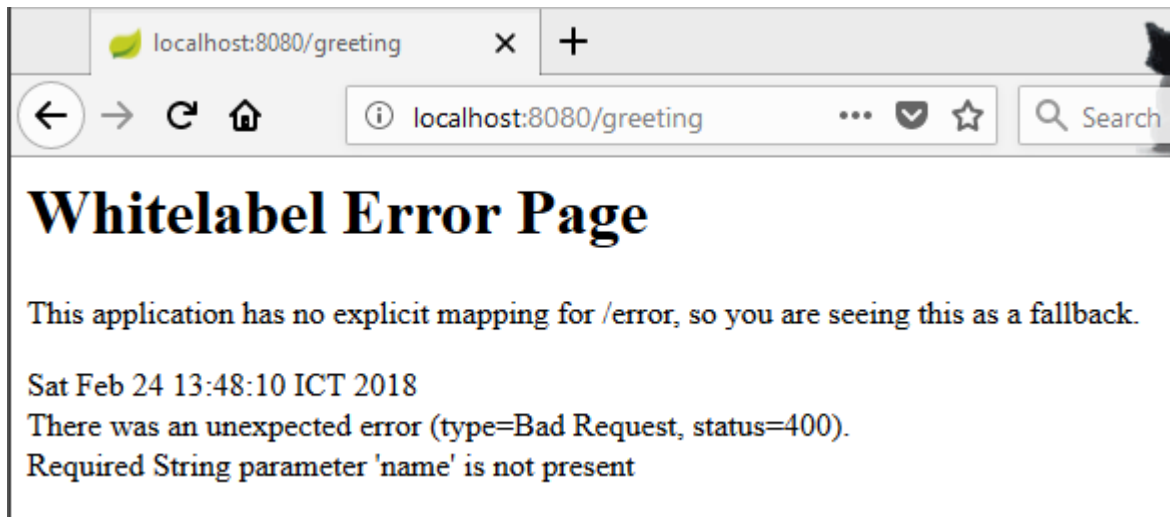
Pertanyaan: apakah hasilnya?



Selamat datang chanek!

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

Pertanyaan: Apakah hasilnya?

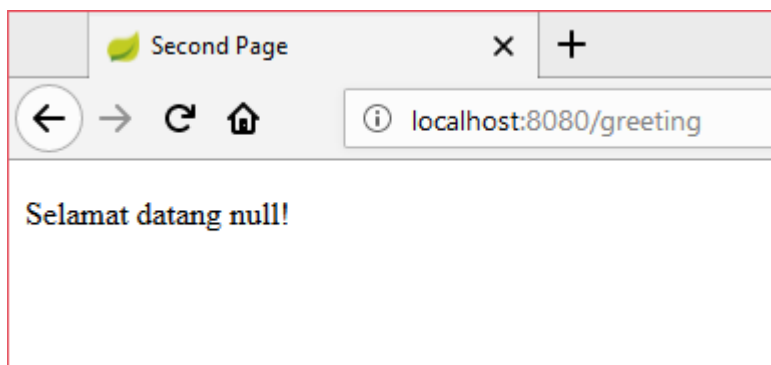


Ubah header method greeting menjadi seperti berikut

```
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?

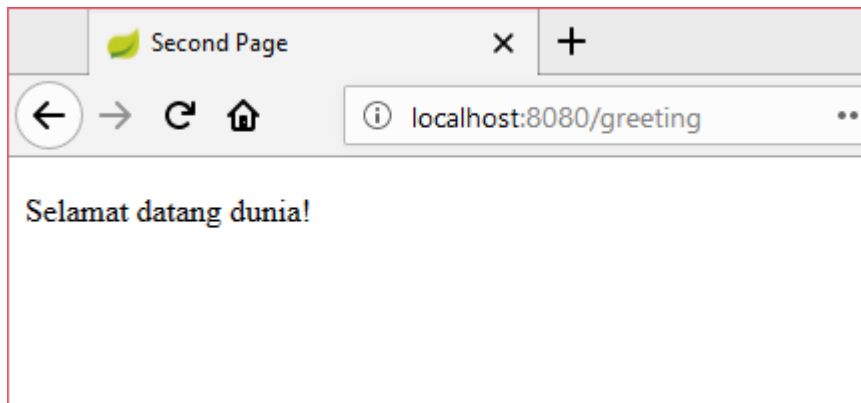


3. Ubah header method greeting menjadi seperti berikut

```
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?

Path Variable

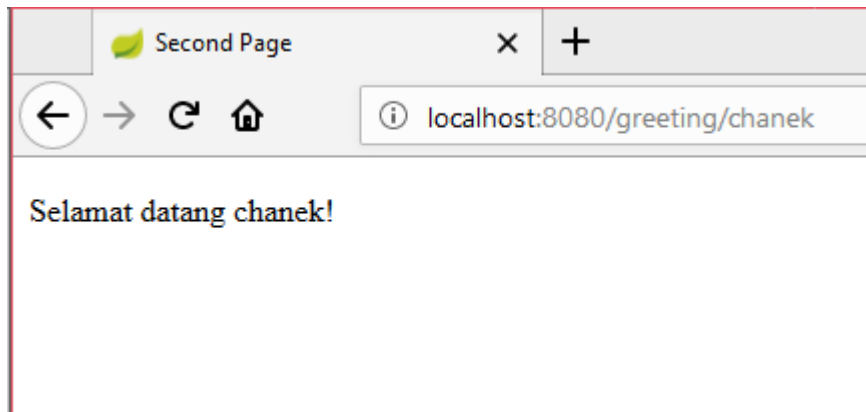
Langkah :

1. PageController tambahkan import berikut:
`import org.springframework.web.bind.annotation.PathVariable;`

Dan tambahkan method `greetingPath` pada **PageController**

```
@RequestMapping("/greeting/{name}")
public String greetingPath (@PathVariable String name, Model model)
{
    model.addAttribute("name", name);
    return "greeting";
}
```

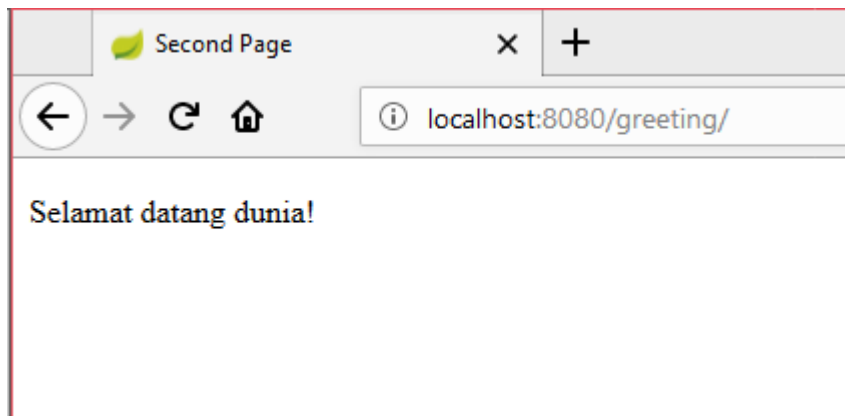
2. Stop Spring Boot yang sedang berjalan, run kembali, dan buka **localhost:8080/greeting/chanek** apakah hasilnya?



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", "apap");
    }
    return "greeting";
}
```

dan tambahkan import

```
import java.util.Optional;
```

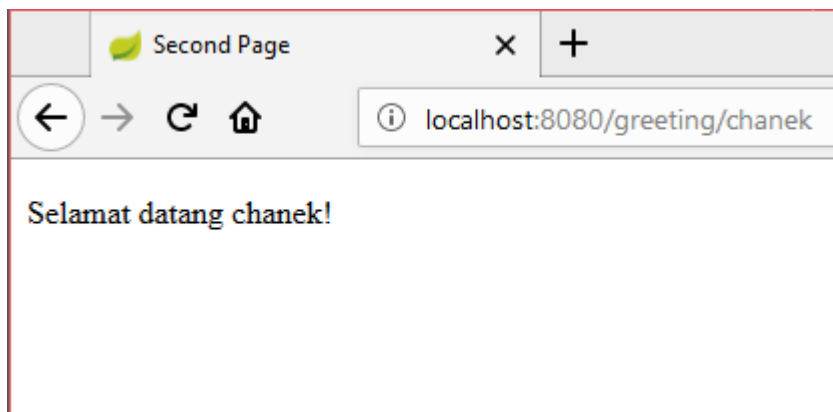
Akses **localhost:8080/greeting/**

Pertanyaan: Apa hasilnya?



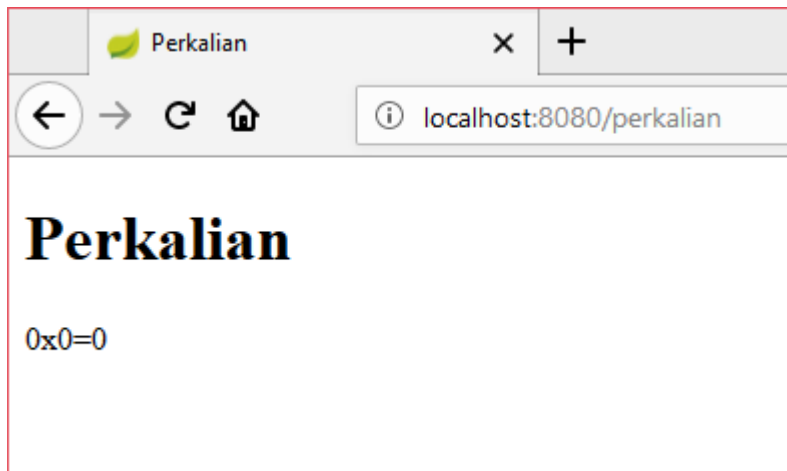
Akses **localhost:8080/greeting/chanek**

Pertanyaan: Apa hasilnya?

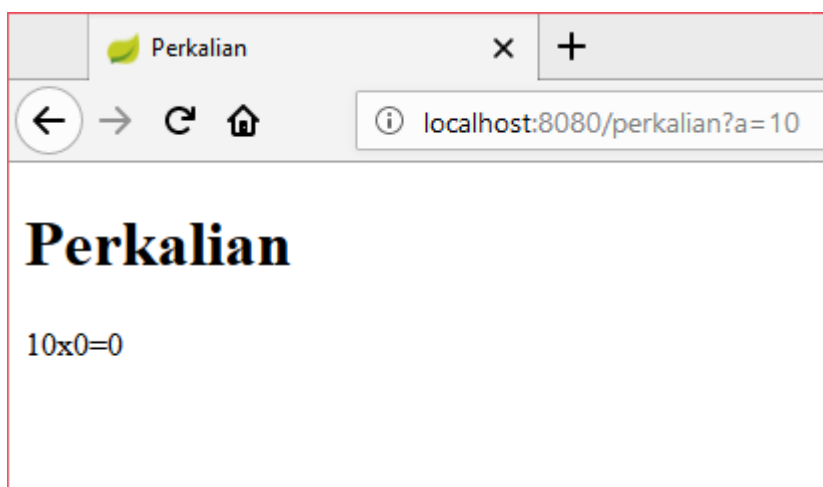


Latihan

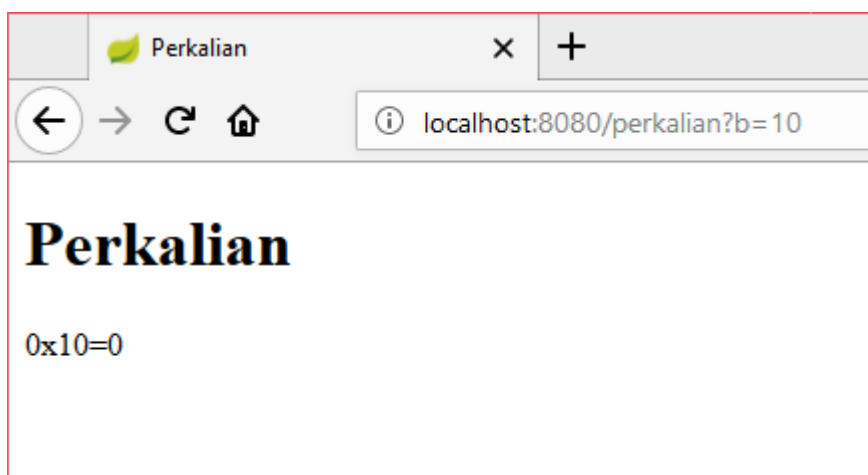
- **localhost:8080/perkalian**



- localhost:8080/perkalian?a=10



- localhost:8080/perkalian?b=10



- localhost:8080/perkalian?a=10&b=10

