



UNIVERSITAS INDONESIA

Arsitektur & Pemograman Aplikasi Perusahaan (APAP)

Tutorial 02-Spring

Tsalis Restu Puspita Asri

1706107056

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI S1 SISTEM INFORMASI

DEPOK

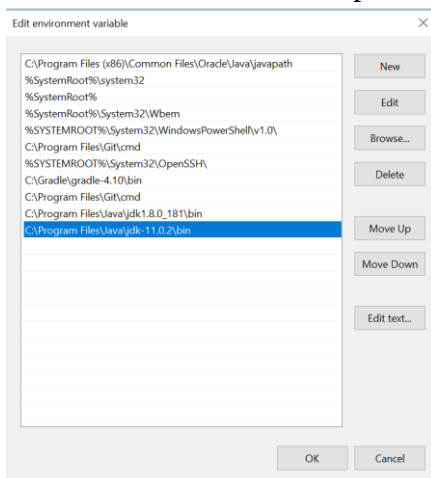
2019

1. Java SE (JDK)

- a. Unduh JDK sesuai dengan operating system yang berjalan di laptop pengguna.

Java SE Development Kit 11.0.2		
You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.		
Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux	147.28 MB	jdk-11.0.2_linux-x64_bin.deb
Linux	154.01 MB	jdk-11.0.2_linux-x64_bin.rpm
Linux	171.32 MB	jdk-11.0.2_linux-x64_bin.tar.gz
macOS	166.13 MB	jdk-11.0.2_osx-x64_bin.dmg
macOS	166.49 MB	jdk-11.0.2_osx-x64_bin.tar.gz
Solaris SPARC	186.78 MB	jdk-11.0.2_solaris-sparcv9_bin.tar.gz
Windows	150.94 MB	jdk-11.0.2_windows-x64_bin.exe
Windows	170.96 MB	jdk-11.0.2_windows-x64_bin.zip

- b. Atur environment variable path di windows setting



2. Spring Tool Suite (STS) for Eclipse

- a. Unduh Spring Tool Suite Eclipse pada <https://spring.io/tools>



Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4.
Free. Open source.

Download STS4
Linux 64-bit

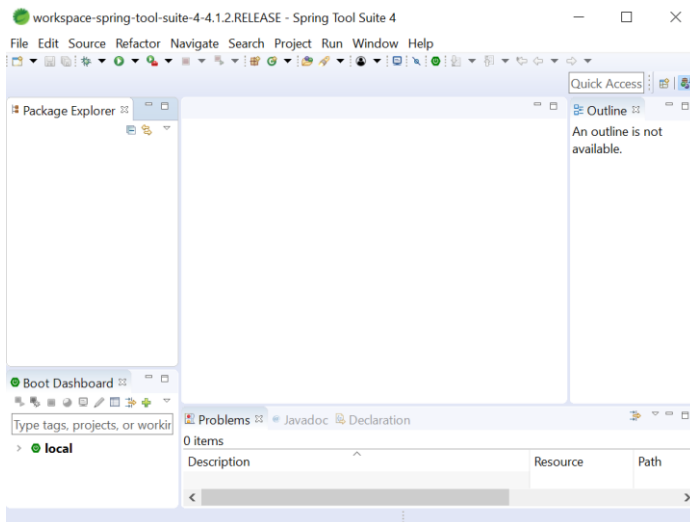
Download STS4
macOS 64-bit

Download STS4
Windows 64-bit

- b. Install Spring Tools Suite (STS) untuk eclipse

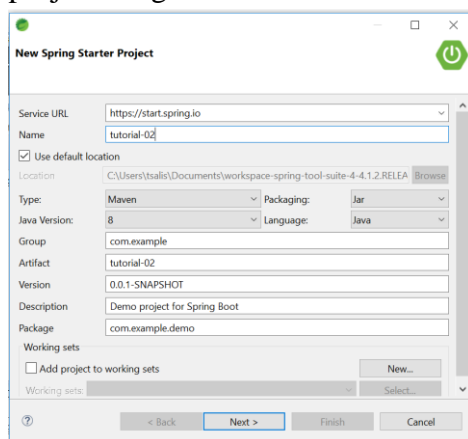


- c. Jalankan aplikasi Spring Tools Suite, akan muncul tampilan project seperti pada gambar berikut.

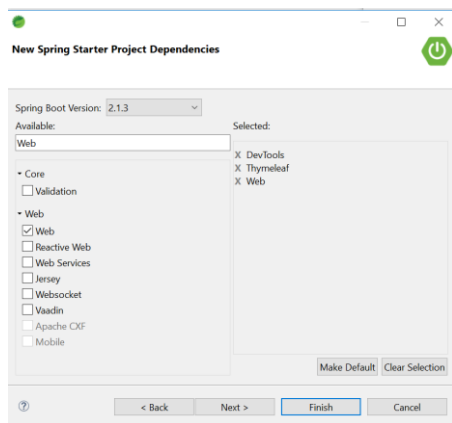


3. Membuat project baru

- a. Untuk membuat project baru, maka klik File -> New -> Spring Starter Project, isi nama project dengan nama tutorial-02. Kemudian klik Next.



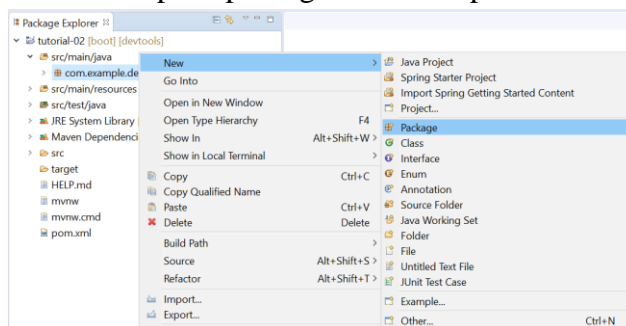
- b. Pilih Dependencies DevTool, Thymeleaf, Web dan klik Finish



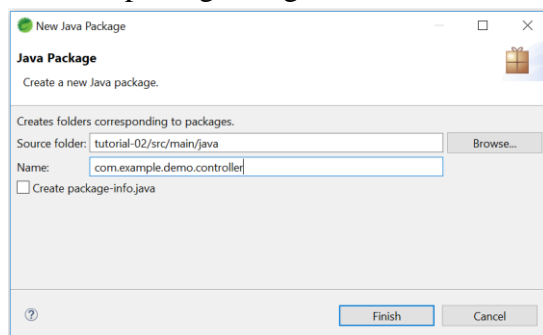
4. Project “Hello”

a. Membuat package Controller

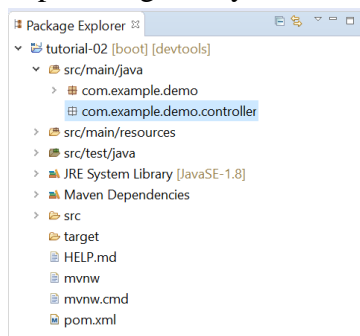
- Klik kanan pada package.com.example.demo -> New -> Package



- Isi nama package dengan nama com.example.demo.controller, kemudian klik Finish.

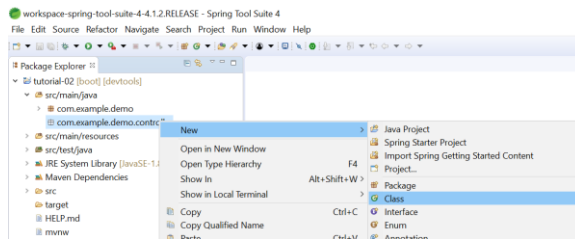


- Untuk mengetahui apakah package yang sudah kita buat sudah berhasil, maka kita dapat mengeceknya di src/main/java seperti gambar berikut.

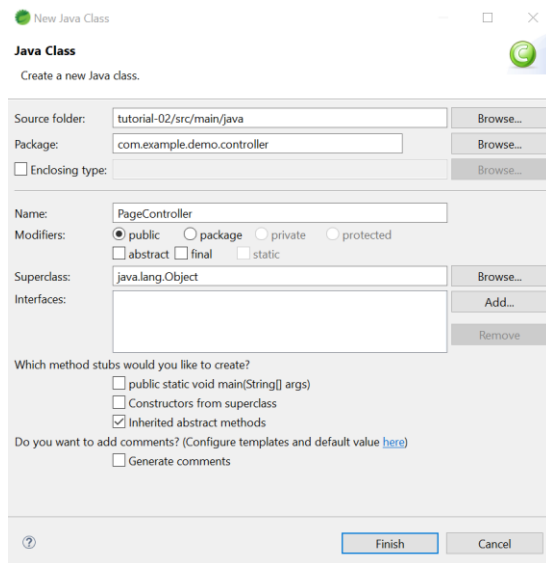


b. Membuat Class PageController

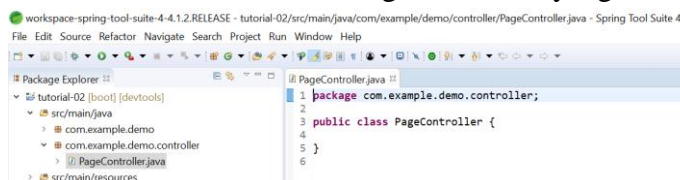
- Klik kanan pada package `com.example.demo.controller` -> New -> Class



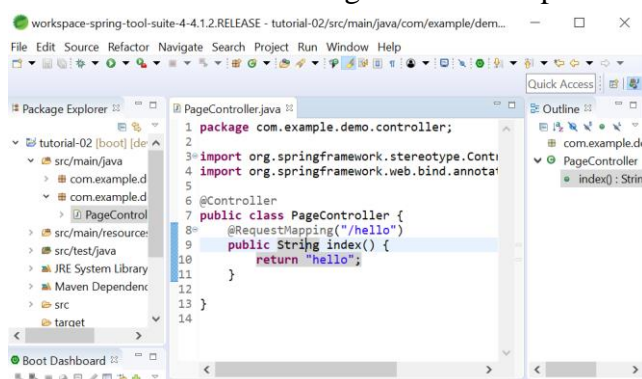
- Isi nama class dengan nama Page Controller



- Berikut ini hasil dari class PageController yang telah terbentuk

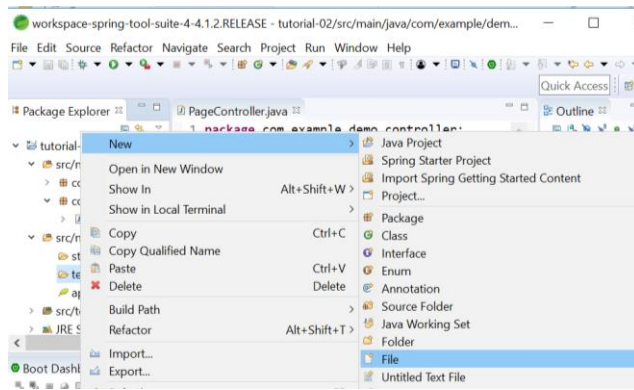


- Isi *code* di dalam Class PageController seperti tampilan di bawah ini.

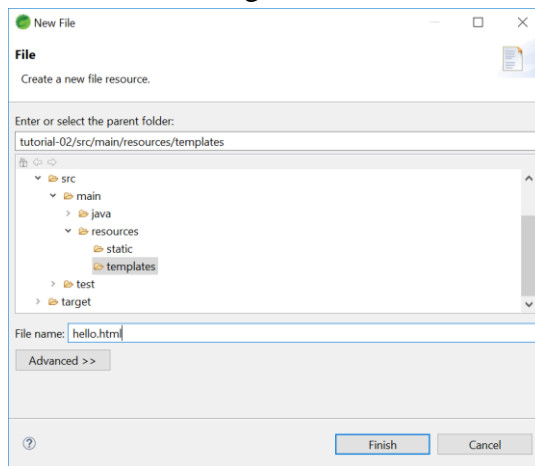


c. Membuat view hello.html

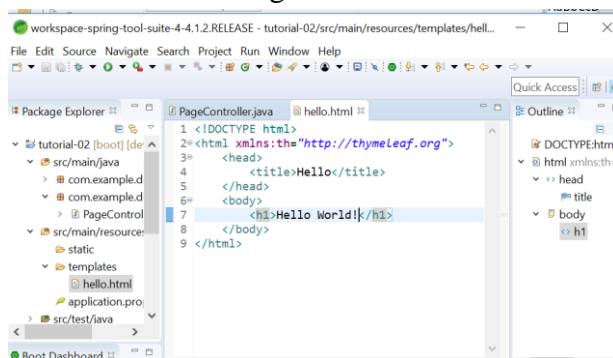
- Klik kanan pada template `src/main/resources` -> New -> File



- Beri nama file dengan nama hello.html

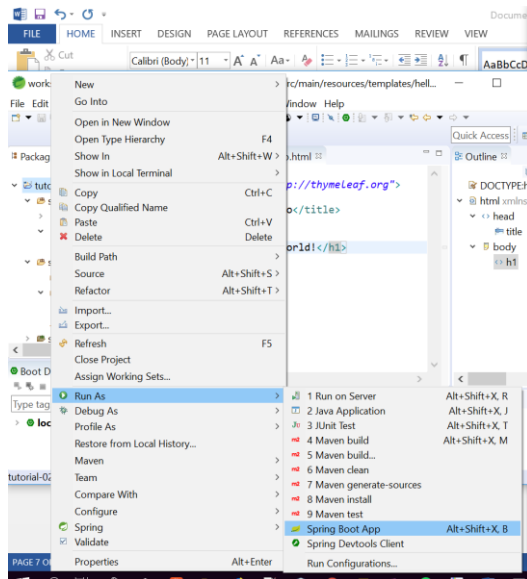


- Isi html tersebut dengan kode berikut ini.



d. Run project “Hello”

- Klik kanan pada folder tutorial-02 -> Run As -> SpringBoot App



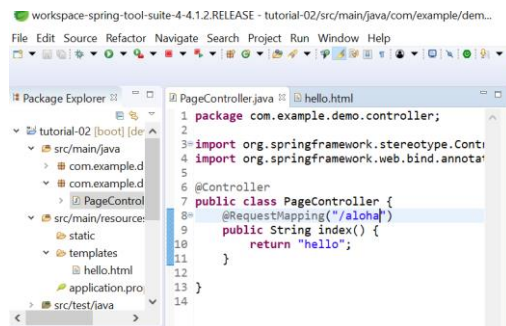
- Buka browser dan akses localhost:8080/hello, maka akan muncul tampilan seperti berikut.



Hello World!

Latihan Project “Hello”

1. Ganti `@RequestMapping("/hello")` menjadi `@RequestMapping("/aloha")` kemudian run.



- Apakah terjadi compile error?

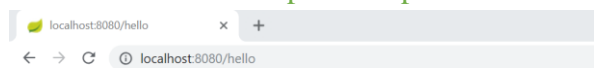
Jawab:

Tidak terjadi compile error pada saat di save running.

- Akses localhost:8080/hello, jelaskan apa yang terjadi

Jawab:

Akan muncul error seperti tampilan berikut.



Whitelabel Error Page

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

Tue Mar 05 20:27:11 ICT 2019

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

No message available

Pada tampilan tersebut muncul pesan error berupa whitelabel error page. Ini disebabkan adanya kesalahan dalam memetakan url sehingga terjadi ketidaksinkronan antara nama url dengan value yang direturn.

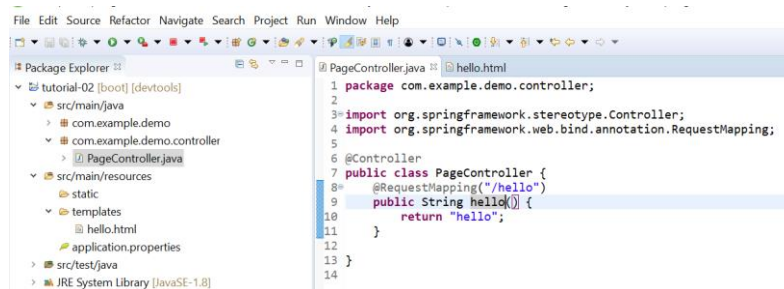
- Apakah kegunaan `@RequestMapping("/hello")` ?

Jawab:

`@RequestMapping` berguna untuk menandakan url yang akan digunakan dalam mengakses, dan method yang digunakan untuk mengakses url tersebut.

- Kembalikan seperti semula.

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



- Apakah terjadi compile error?

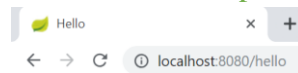
Jawab:

Tidak terjadi compile error saat di save running.

- Akses `localhost:8080/hello`, jelaskan apa yang terjadi

Jawab:

Akan muncul tampilan seperti berikut.

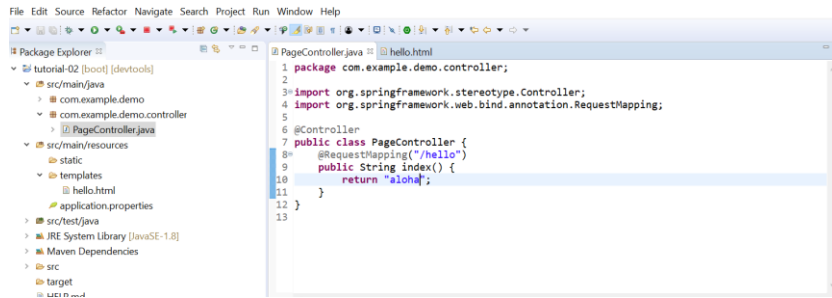


Hello World!

Tampilan tersebut menunjukkan pesan 'Hello World!' dan tidak muncul error. Method `index()` yang dirubah menjadi method `hello()` tidak memberikan pengaruh apapun, karena method `hello()` langsung memanggil variabel nama halaman yang ada di template, yaitu `hello.html`.

- Kembalikan seperti semula.

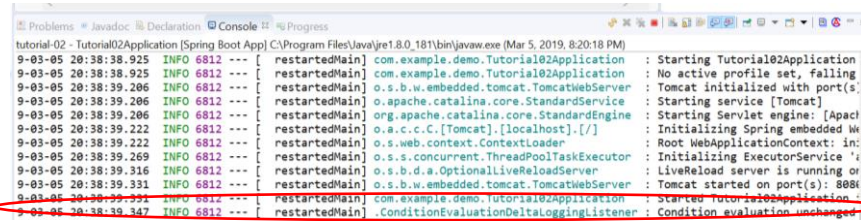
3. Ganti return "hello" menjadi return "aloha"



- Apakah terjadi compile error?

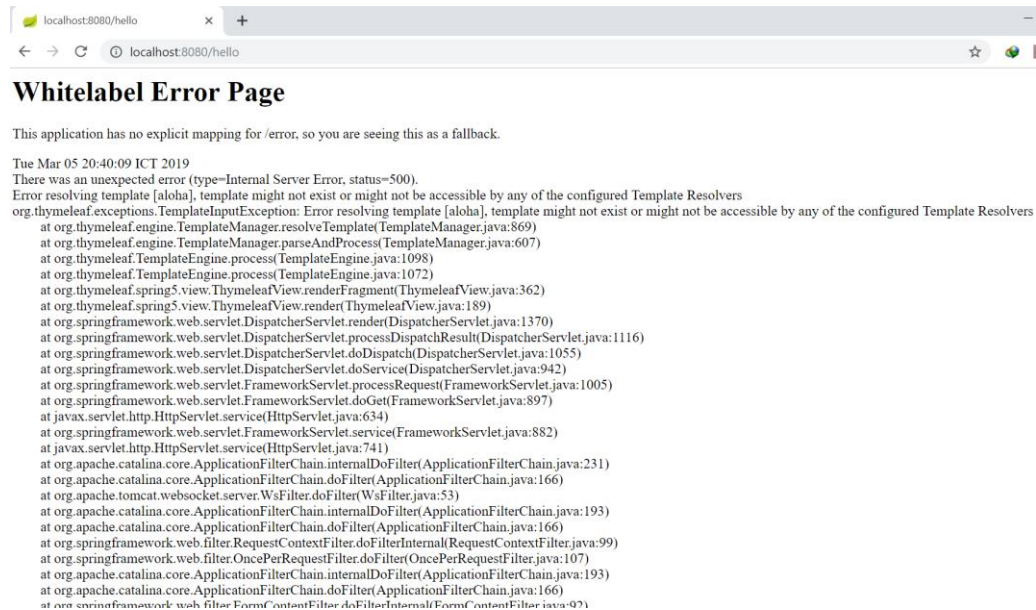
Jawab:

Tidak terjadi compile error, seperti yang terlihat pada console berikut.



- Akses localhost:8080/hello, jelaskan apa yang terjadi

Jawab :



Muncul pesan error whitelabel error page seperti tampilan diatas, karena pada saat mengakses, URL yang dipanggil tidak sesuai dengan template hello.html sehingga akan menampilkan error. Jadi URL yang ditandai untuk dipanggil seharusnya sesuai dengan value yang di-return dalam PageController.

- Apakah kegunaan return pada method controller tersebut?

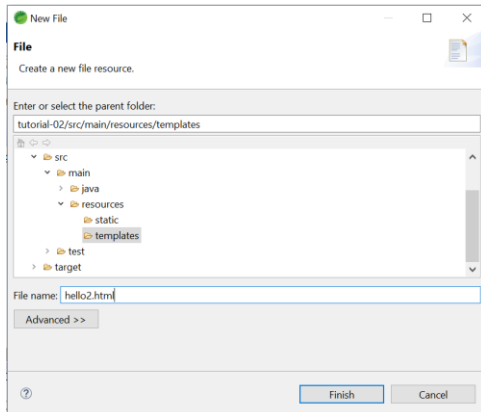
Jawab:

Dalam case ini, return pada method controller yang secara langsung akan menghubungkan ke template bernama hello.html

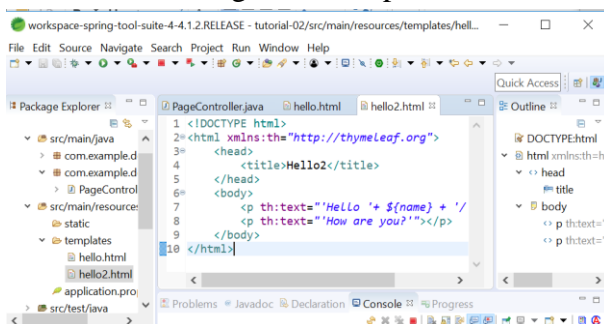
- **Kembalikan seperti semula.**

5. Membuat Request Parameter (Query String)

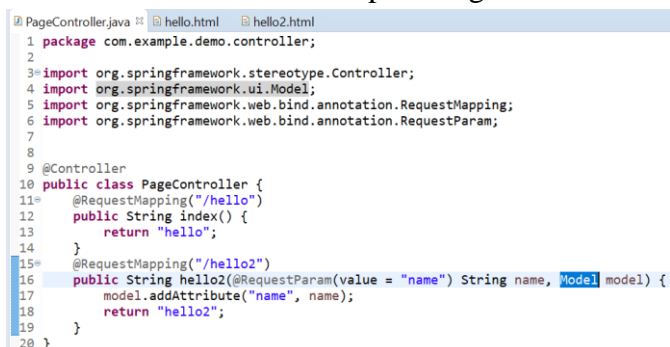
- Buat file bertipe html pada folder template dengan nama hello2.html



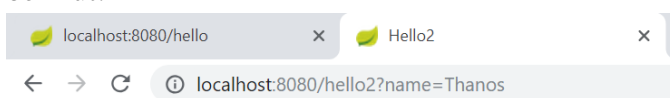
- Kemudian isi dengan kode seperti berikut ini.



- Tambahkan method berikut pada PageController



- Akses localhost:8080/hello2?name=Thanos, sehingga akan muncul tampilan seperti berikut.



Hello Thanos/

How are you?

- e. Pada halaman Hello2, tampilan tersebut tidak muncul pesan error. Nama Thanos muncul karena sudah di-set pada path.

Latihan Request Parameter

1. Mengapa tulisan “Marvel” pada Thymeleaf hello2 tidak pernah muncul di browser?

Jawab:

Karena tulisan ‘Marvel’ di hello2 menggunakan format thymeLeaf yang ditunjukkan dengan source code `<p th:text="Hello ' + ${name} + '">Marvel</p>`, berbeda halnya jika tidak menggunakan format thymeLeaf maka akan muncul tulisan Marvel.

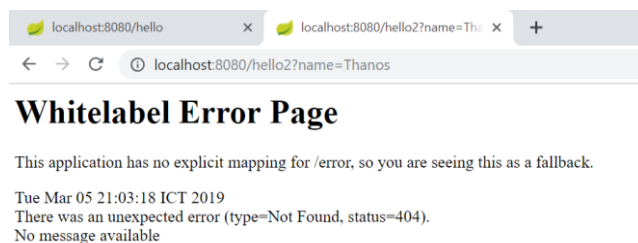
2. Ubah anotasi Request Mapping “/hello2” menjadi “/hello/hello2”.

```
@PageController.java | hello.html | hello2.html
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestParam;
7
8 @Controller
9 public class PageController {
10     @RequestMapping("/hello")
11     public String index() {
12         return "hello";
13     }
14
15     @RequestMapping("/hello/hello2")
16     public String hello2(@RequestParam(value = "name") String name, Model model) {
17         model.addAttribute("name", name);
18         return "hello2";
19     }
20 }
```

- Coba akses pada browser, jelaskan apa yang terjadi. Kembalikan seperti semula.

Jawab:

Ketika di save, tidak muncul compile error. Selanjutnya saat dijalankan pada browser akan muncul tampilan berikut.



Error tersebut muncul karena url yang dipanggil pada request mapping tidak sesuai.

3. Akses localhost:8080/hello2

- Jelaskan apa yang terjadi

Jawab:

Setelah dikembalikan menjadi /hello2, maka akan muncul tampilan seperti berikut pada browser

Whitelabel Error Page

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

Tue Mar 05 21:07:29 ICT 2019

There was an unexpected error (type=Bad Request, status=400).

Required String parameter 'name' is not present

org.springframework.web.bind.MissingServletRequestParameterException: Required String parameter 'name' is not present

```
at org.springframework.web.method.annotation.RequestParamMethodArgumentResolver.handleMissingValue(RequestParamMethodArgumentResolver.java:202)
at org.springframework.web.method.annotation.AbstractNamedValueMethodArgumentResolver.resolveArgument(AbstractNamedValueMethodArgumentResolver.java:113)
at org.springframework.web.method.support.HandlerMethodArgumentResolverComposite.resolveArgument(HandlerMethodArgumentResolverComposite.java:126)
at org.springframework.web.method.support.InvocableHandlerMethod.getMethodArgumentValues(InvocableHandlerMethod.java:166)
at org.springframework.web.method.support.InvocableHandlerMethod.invokeForRequest(InvocableHandlerMethod.java:134)
at org.springframework.web.servlet.mvc.method.annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:102)
at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.invokeHandlerMethod(RequestMappingHandlerAdapter.java:895)
at org.springframework.web.servlet.mvc.method.annotation.RequestMappingHandlerAdapter.handleInternal(RequestMappingHandlerAdapter.java:800)
at org.springframework.web.servlet.mvc.method.AbstractHandlerMethodAdapter.handle(AbstractHandlerMethodAdapter.java:87)
at org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:1038)
at org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:942)
at org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:1005)
at org.springframework.web.servlet.FrameworkServlet.doGet(FrameworkServlet.java:897)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:634)
at org.springframework.web.servlet.FrameworkServlet.service(FrameworkServlet.java:882)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:741)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:231)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:53)
at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:193)
at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:166)
```

Saat dijalankan akan muncul pesan error whitelabel error page. Hal ini karena pada `@RequestParam` baru mendeklarasikan variabel `name` dan belum disimpan di parameter, sehingga saat dipassing nilainya ke view, maka akan muncul error.

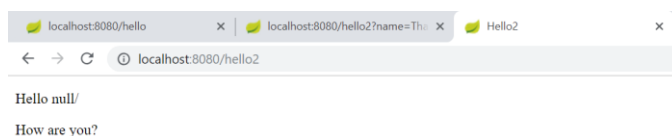
- Ubah method tersebut menjadi seperti dibawah ini.

```
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestParam;
7
8
9 @Controller
10 public class PageController {
11     @RequestMapping("/hello")
12     public String index() {
13         return "hello";
14     }
15     @RequestMapping("/hello2")
16     public String hello2(@RequestParam(value = "name", required = false) String name, Model model) {
17         model.addAttribute("name", name);
18         return "hello2";
19     }
20 }
21
```

- Akses kembali, jelaskan apa yang terjadi.

Jawab:

Saat diakses kembali melalui `http://localhost:8080/hello2`, akan muncul tampilan seperti berikut.



Tampilan tersebut tidak error. Hanya saja pada nama terisi null, karena pada `PageController`, variabel `name` berupa `String` belum diset nilainya atau masih kosong sehingga pada browser akan otomatis secara default terisi null.

- Kembalikan seperti semula.

```

*PageController.java | hello.html | hello2.html | calculator.html
1 package com.example.demo.controller;
2
3 import java.util.Optional;
4
5 import org.springframework.stereotype.Controller;
6 import org.springframework.ui.Model;
7 import org.springframework.web.bind.annotation.PathVariable;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.RequestParam;
10
11 @Controller
12 public class PageController {
13     @RequestMapping("/hello")
14     public String index() {
15         return "hello";
16     }
17     @RequestMapping("/hello2")
18     public String hello2(@RequestParam(value = "name") String name, Model model) {
19         model.addAttribute("name", name);
20         return "hello2";
21     }

```

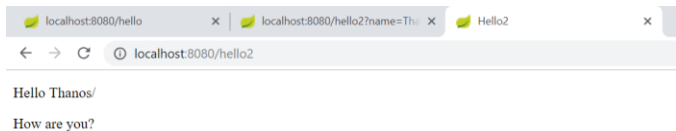
- Ubah method tersebut kembali menjadi seperti berikut.

```

*PageController.java | hello.html | hello2.html
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestParam;
7
8
9 @Controller
10 public class PageController {
11     @RequestMapping("/hello")
12     public String index() {
13         return "hello";
14     }
15     @RequestMapping("/hello2")
16     public String hello2(@RequestParam(value = "name", required = false, defaultValue = "Thanos") String name, Model model) {
17         model.addAttribute("name", name);
18         return "hello2";
19     }
20 }
21

```

- Akses kembali, jelaskan apa yang terjadiJawab:
Ketika diakses kembali melalui localhost:8080/hello2, sehingga akan muncul tampilan seperti berikut.



Pada tampilan tersebut tidak muncul error dan tertulis nama pengguna, dalam hal ini Thanos. Hal ini karena pada PageController sudah ditetapkan default nilai namanya, yaitu Thanos.

- Kembalikan seperti semula.

```

*PageController.java | hello.html | hello2.html | calculator.html
1 package com.example.demo.controller;
2
3 import java.util.Optional;
4
5 import org.springframework.stereotype.Controller;
6 import org.springframework.ui.Model;
7 import org.springframework.web.bind.annotation.PathVariable;
8 import org.springframework.web.bind.annotation.RequestMapping;
9 import org.springframework.web.bind.annotation.RequestParam;
10
11 @Controller
12 public class PageController {
13     @RequestMapping("/hello")
14     public String index() {
15         return "hello";
16     }
17     @RequestMapping("/hello2")
18     public String hello2(@RequestParam(value = "name") String name, Model model) {
19         model.addAttribute("name", name);
20         return "hello2";
21     }

```

6. Membuat Path Variable

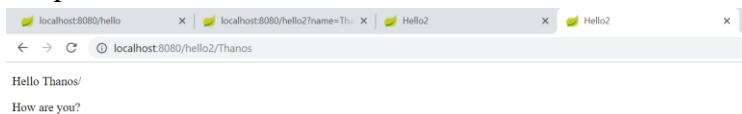
- Menambahkan method helloPath pada PageController

```

PageController.java hello.html hello2.html calculator.html
12 public class PageController {
13     @RequestMapping("/hello")
14     public String index() {
15         return "hello";
16     }
17     @RequestMapping("/hello2")
18     public String hello2(@RequestParam(value = "name") String name, Model model) {
19         model.addAttribute("name", name);
20         return "hello2";
21     }
22
23     @RequestMapping("/hello2/{name}")
24     public String helloPath(@PathVariable String name, Model model) {
25         model.addAttribute("name", name);
26         return "hello2";
27     }
28 }

```

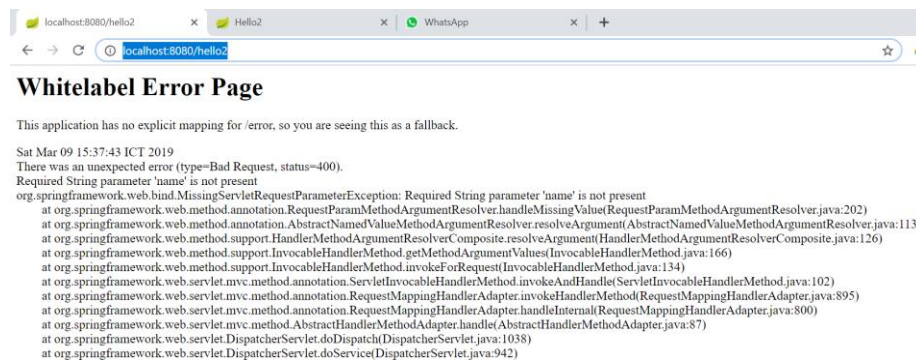
- b. Kemudian akses melalui <http://localhost:8080/hello2/Thanos> dan akan muncul seperti tampilan berikut.



Latihan Path Variable

1. Akses localhost:8080/hello2, jelaskan apa yang terjadi.

Jawab:



Tampilan yang diakses melalui localhost:8080/hello2 berupa pesan whitelabel error page, karena pada `@PathVariable` harus ditentukan parameter Stringnya sehingga saat mengakses nilai dari URI template tidak akan muncul error.

2. Ubah method helloPath menjadi seperti di bawah ini.

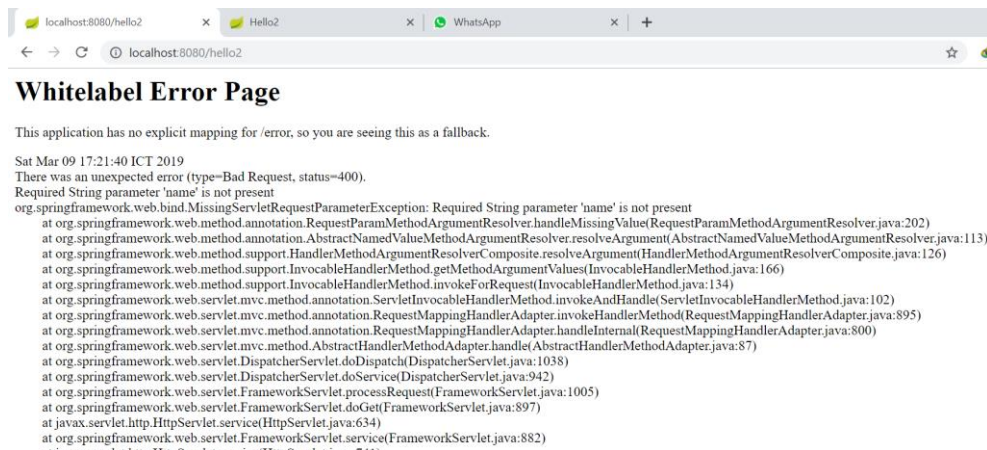
```

PageController.java hello.html hello2.html calculator.html
1 package com.example.demo.controller;
2
3 import java.util.Optional;
4
5 @Controller
6 public class PageController {
7     @RequestMapping("/hello")
8     public String index() {
9         return "hello";
10    }
11    @RequestMapping("/hello2")
12    public String hello2(@RequestParam(value = "name") String name, Model model) {
13        model.addAttribute("name", name);
14        return "hello2";
15    }
16
17    @RequestMapping("/hello2/{name}")
18    public String helloPath(@PathVariable Optional <String> name, Model model) {
19        if (name.isPresent()) {
20            model.addAttribute("name", name.get());
21        } else {
22            model.addAttribute("name", "Phoenix");
23        }
24        return "hello2";
25    }
26 }

```


- Akses localhost:8080/hello2, jelaskan apa yang terjadi.

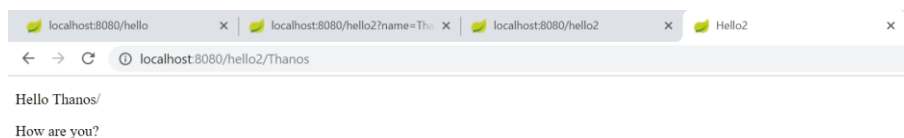
Jawab:



Akan menampilkan pesan error berupa whitelabel error page. Error ini muncul karena alamat hello2 sudah dideklarasikan pada method helloPath sebelumnya.

- Akses localhost:8080/hello2/Thanos, jelaskan apa yang terjadi.

Jawab:



Ketika diakses melalui localhost:8080/hello2/Thanos, maka tidak akan menampilkan pesan error dan muncul nama Thanos seperti yang diperlihatkan pada gambar diatas. Hal ini terjadi karena alamat url yang dipanggil spesifik, yaitu /hello2/{name} saat dilakukan pengecekan method isPresent() maka akan menghasilkan nilai true, sehingga akan ditampilkan output seperti gambar diatas

Latihan Calcu-Convert

- Buatlah sebuah halaman baru sebagai number calculator.
- Halaman ini menerima parameter berupa dua angka.
- Pada view calculator menampilkan formula penjumlahan kedua angka tersebut beserta
- hasilnya. Tampilkan pula bentuk pengubahan angka hasil penjumlahan menjadi kata/kalimat. Contoh: Jika parameternya adalah 2 dan 9 maka pada view harus tampil minimal “2 + 9 = 11 (Sebelas)”.

Jawab :

Berikut ini merupakan source code calcu-convert pada PageController

```

33 //LATIHAN CALCULATOR
34 @RequestMapping("/calculator/{angka1}/{angka2}")
35 //method merequest parameter angka1 dan angka2 dan mengubahnya ke bentuk integer
36 public String variabel(@PathVariable int angka1, @PathVariable int angka2, Model model) {
37
38     //perhitungan
39     int hasil = angka1+angka2;
40
41     //deklarasi array untuk mengubah bilangan menjadi kata
42     String[] kata = {"Nol", "Satu", "Dua", "Tiga", "Empat", "Lima",
43                     "Enam", "Tujuh", "Delapan", "Sembilan", "Sepuluh",
44                     "Sebelas", "Dua belas", "Tiga belas", "Empat belas", "Lima belas",
45                     "Enam belas", "Tujuh belas", "Delapan belas", "Sembilan belas", "Dua puluh"};
46
47     //bentuk untuk ditampilkan ke view
48     String angka = angka1 + "+" + angka2 + "=" + hasil + " (" + kata[hasil] + ")";
49
50     model.addAttribute("angka", angka);
51     return "calculator";
52 }

```

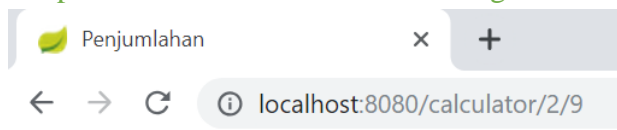
Sedangkan kode untuk menampilkan ke browser adalah sebagai berikut.

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>Penjumlahan</title>
5 </head>
6 <body>
7 <p th:text="${angka}"></p>
8 </body>
9 </html>

```

Output dari kode tersebut adalah sebagai berikut.



2+9=11 (Sebelas)