

## Tutorial 02 - Spring

Tutorial ini mengasumsikan penggunaan pada *Windows*, tanpa melarang penggunaan OS lain, silakan menyesuaikan.

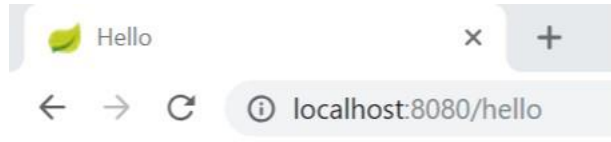
1. Java SE (JDK)
  - Unduh *installer* di <https://www.oracle.com/technetwork/java/javase/downloads/index.html>
  - Jalankan *installer*, ikuti petunjuk yang diberikan
  - Atur *environment variable* (Java path) pada *Windows setting*
2. Spring Tool Suite (STS) for Eclipse
  - Unduh berkas zip di <https://spring.io/tools>
  - Ekstrak berkas zip tersebut
  - Jalankan STS
3. Membuat Project Baru
  - Klik File >> New >> Spring Starter Project
  - Isi nama *project* dengan tutorial-02 >> Next
  - Pilih *dependencies*: DevTools, Thymeleaf, dan Web >> Finish
4. Project "Hello"
  - Buat *package controller*
    - o *Expand* tutorial-02 pada Project Explorer (klik pada ikon >)
    - o *Expand* src/main/java
    - o Klik kanan pada package com.example.demo >> New >> Package
    - o Isi nama dengan com.example.demo.controller
  - Buat Class PageController.java
    - o Klik kanan pada package com.example.demo.controller >> New >> Class
    - o Isi nama dengan PageController >> Finish
    - o Isi class tersebut dengan kode berikut:

```
PageController.java
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.RequestMapping;
5
6 @Controller
7 public class PageController {
8     @RequestMapping("/hello")
9     public String index() {
10         return "hello";
11     }
12 }
```

**Tips:** Ketika menggunakan anotasi `@Controller` dan `@RequestMapping` Anda akan menemukan *error* bahwa anotasi belum ter-*import*. Untuk dapat melakukan *import* secara otomatis dapat menggunakan shortcut `Ctrl+Shift+O`.

- Buat *view* hello.html
    - o *Expand* src/main/resources
    - o Klik kanan pada templates >> New >> File
    - o Isi nama dengan hello.html >> Finish
    - o Isi html tersebut dengan kode berikut:
- ```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3     <head>
4         <title>Hello</title>
5     </head>
6     <body>
7         <h1>Hello World!</h1>
8     </body>
9 </html>
```

- Run project "Hello"
  - o Klik kanan pada tutorial-02 >> Run As >> Spring Boot App
  - o Buka Browser
  - o Akses **localhost:8080/hello**
  - o Tampilan yang didapatkan:



# Hello World!

## Latihan Project "Hello"

1. Ganti `@RequestMapping("/hello")` menjadi `@RequestMapping("/aloha")` kemudian *run*.
    - Apakah terjadi *compile error*?
    - Akses `localhost:8080/hello`, jelaskan apa yang terjadi
    - Apakah Kegunaan `@RequestMapping("/hello")`
 Kembalikan seperti semula.
  2. Ganti nama *method* `index()` menjadi nama *method* `hello()`
    - Apakah terjadi *compile error*?
    - Akses `localhost:8080/hello`, jelaskan apa yang terjadi
 Kembalikan seperti semula.
  3. Ganti `return "hello"` menjadi `return "aloha"`
    - Apakah terjadi *compile error*?
    - Akses `localhost:8080/hello`, jelaskan apa yang terjadi
    - Apakah kegunaan `return` pada *method controller* tersebut?
 Kembalikan seperti semula.
5. Request Parameter (Query String)
- Buat `html file` baru pada *folder* `templates` dengan nama `hello2.html`

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>Hello2</title>
5   </head>
6   <body>
7     <p th:text="'Hello ' + ${name} + '!'">Marvel</p>
8     <p th:text="'How are you?'"></p>
9   </body>
10 </html>

```
  - Tambahkan *method* berikut pada `PageController`

```

@RequestMapping("/hello2")
public String hello2(@RequestParam(value = "name") String name, Model model) {
    model.addAttribute("name", name);
    return "hello2";
}

```
  - Matikan Spring Boot lalu jalankan kembali (tidak perlu dilakukan apabila menggunakan DevTools).
  - Akses `localhost:8080/hello2?name=Thanos`

Anda baru saja mengirim parameter melalui GET *request*. Method `hello2` kemudian menangkap *parameter name* dari *query string* dengan menambahkan *parameter* `@RequestParam(value = "[namaparameter]") String name` sebagai *parameter method*. Objek `Model` digunakan untuk *passing* nilai ke *view*, dalam contoh ini `hello2.html`.

### Latihan Request Parameter

1. Mengapa tulisan "Marvel" pada Thymeleaf `hello2` tidak pernah muncul di browser?
2. Ubah anotasi Request Mapping `"/hello2"` menjadi `"/hello/hello2"`. Coba akses pada browser, jelaskan apa yang terjadi. Kembalikan seperti semula.
3. Akses `localhost:8080/hello2`
  - Jelaskan apa yang terjadi.
  - Ubah *method*-nya menjadi seperti di bawah ini:

```
@RequestMapping("/hello2")
public String hello2(@RequestParam(value = "name", required = false) String name, Model model) {
    model.addAttribute("name", name);
    return "hello2";
}
```

Akses kembali, jelaskan apa yang terjadi. Kembalikan seperti semula.

- Ubah *method*-nya menjadi seperti di bawah ini:

```
@RequestMapping("/hello2")
public String hello2(@RequestParam(value = "name", required = false, defaultValue = "Thanos") String name, Model model) {
    model.addAttribute("name", name);
    return "hello2";
}
```

Akses kembali, jelaskan apa yang terjadi. Kembalikan seperti semula.

### 6. Path Variable

- Tambahkan method `helloPath` pada `PageController`

```
@RequestMapping("/hello2/{name}")
public String helloPath(@PathVariable String name, Model model) {
    model.addAttribute("name", name);
    return "hello2";
}
```

- Matikan Spring Boot lalu jalankan kembali
- Akses <http://localhost:8080/hello2/Thanos>

### Latihan Path Variable

1. Akses `localhost:8080/hello2`, jelaskan apa yang terjadi.
2. Ubah method `helloPath` menjadi seperti di bawah ini:

```
@RequestMapping(value={"/hello2", "/hello2/{name}"})
public String helloPath(@PathVariable Optional<String> name, Model model) {
    if (name.isPresent()) {
        model.addAttribute("name", name.get());
    } else {
        model.addAttribute("name", "Phoenix");
    }
    return "hello2";
}
```

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

### Latihan Calcu-Convert

1. Buatlah sebuah halaman baru sebagai *number calculator*.
2. Halaman ini menerima parameter berupa dua angka.

3. 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)".

### Pengumpulan

1. Latihan Project "Hello", Latihan Request Parameter, dan Latihan Path Variable tuliskan jawaban anda dalam satu *file* dengan format **npm\_nama-lengkap-anda.pdf**. Dalam menjawab, lengkapi *screen capture* tampilan/output pada browser sesuai tahapan yang diminta. Unggah ke submission slot yang disediakan di Scele.
2. Latihan Calcu-Convert beserta seluruh isi folder tutorial-02 dikumpulkan dengan cara push ke GitHub (<https://github.com/achmad-f-abka/apap>).