

I. Latihan Project Viral



1. Ganti baris `@RequestMapping("/viral")` menjadi `@RequestMapping("/viral2jari")`.
Kemudian run kembali.

a. Apakah terjadi compile error?

Jawab:

Tidak

b. Coba akses localhost:8080/viral, berikan alasan dan jelaskan apa yang terjadi.

Jawab:

Akses localhost:8080/viral menampilkan halaman *error*. Hal ini terjadi karena `@RequestMapping("/viral")` sudah tidak menunjuk pada controller manapun, sehingga yang ditampilkan adalah halaman error.



c. Apakah Kegunaan `@RequestMapping("/viral")`

Jawab:

Kegunaan `@RequestMapping("/viral")` adalah untuk menyediakan link/alamat yang dapat digunakan untuk memanggil controller tertentu, sehingga ketika melakukan pemanggilan sebuah *Request Mapping*, controller yang di-assign dengannya akan dijalankan.

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

- a. Apakah terjadi compile error?

Jawab:

Tidak

- b. Coba akses localhost:8080/viral, berikan alasan dan jelaskan apa yang terjadi.

Jawab:

Akses localhost:8080/viral menampilkan halaman viral.html yang terdapat pada komponen *resource*. Hal ini terjadi karena perubahan yang dilakukan hanya merupakan perubahan nama controller, sedangkan yang dipanggil tetap *Request Mapping* dengan nama yang sama, dan me-refer pada *controller* yang sama dengan nama yang sudah diganti.



3. Ganti string return type menjadi menjadi return "viral2jari"

- a. Apakah terjadi compile error?

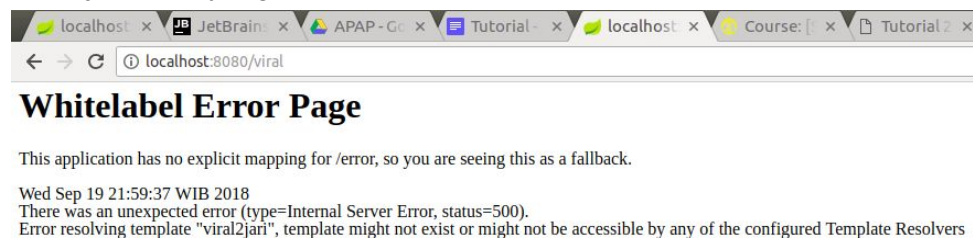
Jawab:

Tidak

- b. Coba akses localhost:8080/viral, berikan alasan dan jelaskan apa yang terjadi.

Jawab:

Akses localhost:8080/viral menampilkan halaman *error*. Hal ini terjadi karena perubahan yang dilakukan pada *String return* tidak merujuk pada file manapun pada komponen *resource*, sehingga tidak ada halaman dengan "viral2jari.html" yang akan ditampilkan.

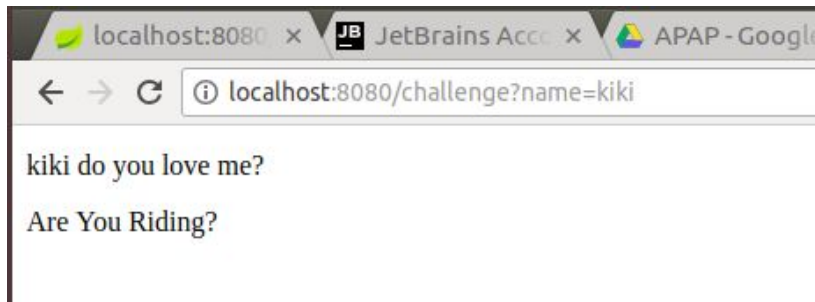


- c. Apakah kegunaan return type pada method controller tersebut?

Jawab:

Untuk menampilkan file .html dengan nama yang sesuai dengan *String* yang tertulis pada *return type* tersebut.

II. Latihan Request Parameter



1. Mengapa tulisan "*In My feeling*" pada *Thymeleaf challenge* tidak pernah muncul di browser?

Jawab:

Karena tulisan "*In My feeling*" sudah ditimpa oleh tulisan baru yang di-assign menggunakan *tag text* sebagai salah satu metode penulisan teks pada *Thymeleaf*.

2. Ubah nilai anotasi Request Mapping `"/challenge"` menjadi `"/viral/challenge"`. Deskripsikan hasil pada browser dan kembalikan ke bentuk sebelumnya!

Jawab:

Hasil pada browser setelah anotasi *Request Mapping* diubah menampilkan halaman *challenge.html*. Perubahan tersebut dilakukan untuk membuat *link* terlihat lebih spesifik menggambarkan navigasi *link* tersebut.

3. Akses localhost:8080/challenge.

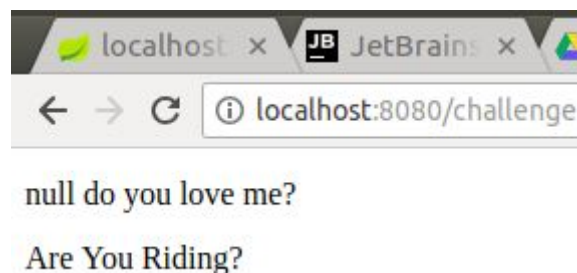
- a. Deskripsikan hasilnya

Jawab:

Akses pada *link* tersebut menampilkan halaman *challenge.html* pada browser. Halaman tersebut menerima atribut *name* sebagai parameter yang dicantumkan pada *link* yang bernilai "*kiki*". Atribut tersebut diterima oleh *controller* dalam bentuk *parameter*. Kemudian dilanjutkan ke halaman *challenge.html* sebagai atribut yang dapat di akses oleh *Thymeleaf*.

- b. Ubah methodnya menjadi seperti di bawah ini, deskripsikan lalu kembalikan ke bentuk awal!

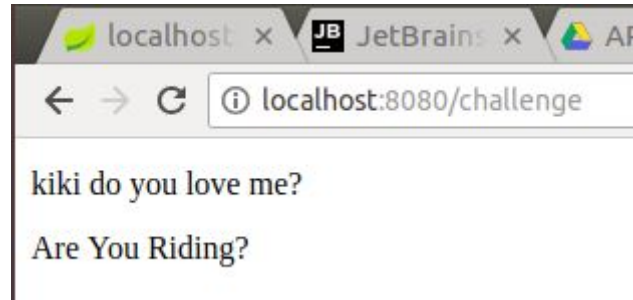
Jawab:



Setelah menambahkan atribut *required* dengan nilai *false*, penulisan parameter menjadi opsional pada *link* tersebut. Sehingga atribut yang diterima pada halaman *challenge.html* bernilai *null*.

- c. Ubah methodnya menjadi seperti di bawah ini, deskripsikan lalu kembalikan ke bentuk awal!

Jawab:



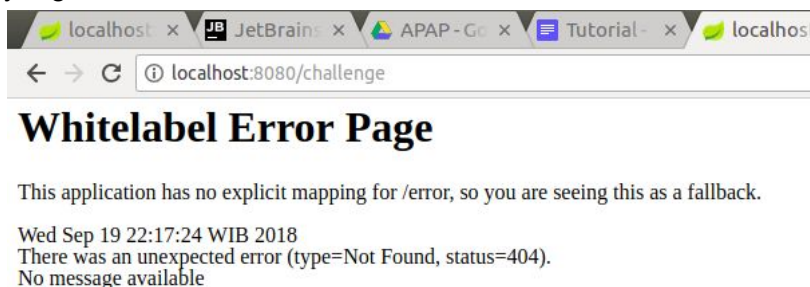
Penambahan atribut *defaultValue* setelah penambahan atribut *required* yang bernilai false membuat controller tersebut akan secara otomatis meng-assign atribut *defaultValue* sebagai *parameter* jika pada penulisan *link* atributnya dikosongkan.

III. Latihan Path Variable

1. Akses localhost:8080/challenge/
Apakah hasil dari halaman tersebut?

Jawab:

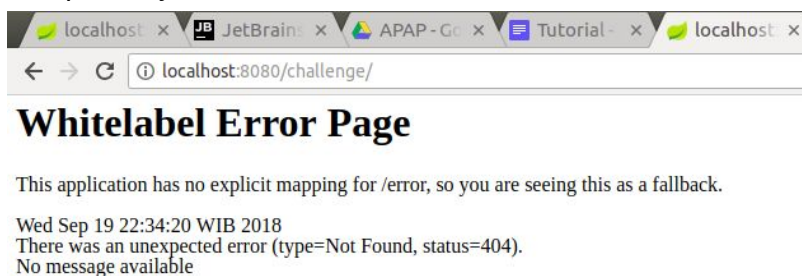
Akses pada *link* tersebut menampilkan halaman error karena penulisan belum lengkap. Penulisan *link* yang benar sudah ditambahkan *Path Variable* sebagai data yang akan dikirimkan ke *controller*.



2. Ubah method challengePath menjadi seperti berikut
Akses localhost:8080/challenge/
Apakah hasil dari halaman tersebut?

Jawab:

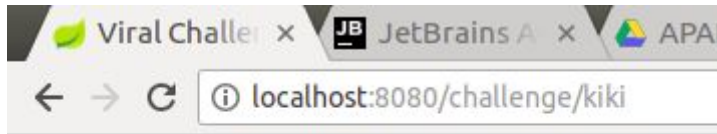
Akses pada *link* tersebut menampilkan halaman *error* karena tidak dapat dikenali oleh *controller*. Penggunaan *Path Variable* mensyaratkan penulisan variabel pada setiap *link*-nya



3. Akses `localhost:8080/challenge/kiki`
Apakah hasil dari halaman tersebut?

Jawab:

Akses pada link tersebut menampilkan halaman *challenge.html* yang telah menerima parameter *name* berupa *Path Variable* yang diterima melalui *controller* dari link yang dituliskan pada URL.



kiki do you love me?

Are You Riding?