



Back End Development 1 ○

+

Sesi 16



Springboot -⁺ Bootstrapping + Code Structures

Springboot - Bootstrapping + Code Structures - Sesi 16

Spring Web + Thymeleaf

Spring Web adalah salah satu proyek yang membantu kita untuk membuat sebuah aplikasi web (yang bisa berjalan menggunakan browser) dan Spring Web ini pasti menggunakan anotasi `@Controller` pada level Controller nya. Sedangkan Thymeleaf adalah sebuah template engine khusus Java tempat kita menuliskan file-file HTML nya sama seperti JSP tapi Thymeleaf adalah template engine yang paling baru didunia Java.

Buat sebuah proyek dengan spesifikasi sebagai berikut:

Project	Maven Project
Versi Spring Boot	2.1.12
Group	id.belajar
Artifact	Spring-web
Dependencies	Spring-boot-starter-thymeleaf
	Spring-boot-starter-web
	Spring-boot-devtools
	Org.webjar Bootstrap



```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.1.12.RELEASE</version>
    <relativePath/> <!-- Lookup parent from repository -->
  </parent>
  <groupId>id.belajar</groupId>
  <artifactId>spring-web</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>spring-web</name>
  <description>Demo project for Spring Boot</description>

  <properties>
    <java.version>1.8</java.version>
    <bootstrap.version>4.2.1</bootstrap.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
  </dependencies>
```



```
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <optional>true</optional>
    </dependency>

    <dependency>
        <groupId>org.webjars</groupId>
        <artifactId>bootstrap</artifactId>
        <version>${bootstrap.version}</version>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>
```



Buat sebuah class WelcomeController pada layer Controller.

id.belajar.springweb.controller.WelcomeController

```
package id.belajar.springweb.controller;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;

import java.util.Arrays;
import java.util.List;

@Controller
public class WelcomeController {

    private String message = "Hello from Spring Web";

    private List<String> tasks = Arrays.asList("a", "b", "c", "d", "e", "f",
"g");
```



HACKTIV8

```

@GetMapping("/")
public String main(Model model) {
    model.addAttribute("message", message);
    model.addAttribute("tasks", tasks);

    return "welcome"; //view
}

// /hello?name=Jhon
@GetMapping("/hello")
public String mainWithParam(
    @RequestParam(name = "name", required = false, defaultValue = "")
    String name, Model model) {

    model.addAttribute("message", name);

    return "welcome"; //view
}
}

```

Penjelasan kode diatas :

- @Controller adalah anotasi yang menandakan bahwa kita akan membuat sebuah aplikasi Web.
- @GetMapping adalah akses web menggunakan method GET
- Model adalah class yang digunakan untuk mengirimkan data ke View dengan nama file welcome.html
- @RequestParam adalah anotasi untuk membuat sebuah parameter dalam suatu URL.



```
<!DOCTYPE HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">

  <title>Spring Boot Thymeleaf Hello World Example</title>

  <link rel="stylesheet"
th:href="@{webjars/bootstrap/4.2.1/css/bootstrap.min.css}"/>
  <link rel="stylesheet" th:href="@{/css/main.css}"/>
</head>

<body>

<nav class="navbar navbar-expand-md navbar-dark bg-dark fixed-top">
  <a class="navbar-brand" href="#">Spring Boot Web</a>
</nav>
```




```
<main role="main" class="container">

  <div class="starter-template">
    <h1>Spring Boot Web Thymeleaf Example</h1>
    <h2>
      <span th:text="'Hello, ' + ${message}"></span>
    </h2>
  </div>

  <ol>
    <li th:each="task : ${tasks}" th:text="${task}"></li>
  </ol>

</main>
<!-- /.container -->

<script type="text/javascript"
th:src="@{webjars/bootstrap/4.2.1/js/bootstrap.min.js}"></script>
</body>
</html>
```



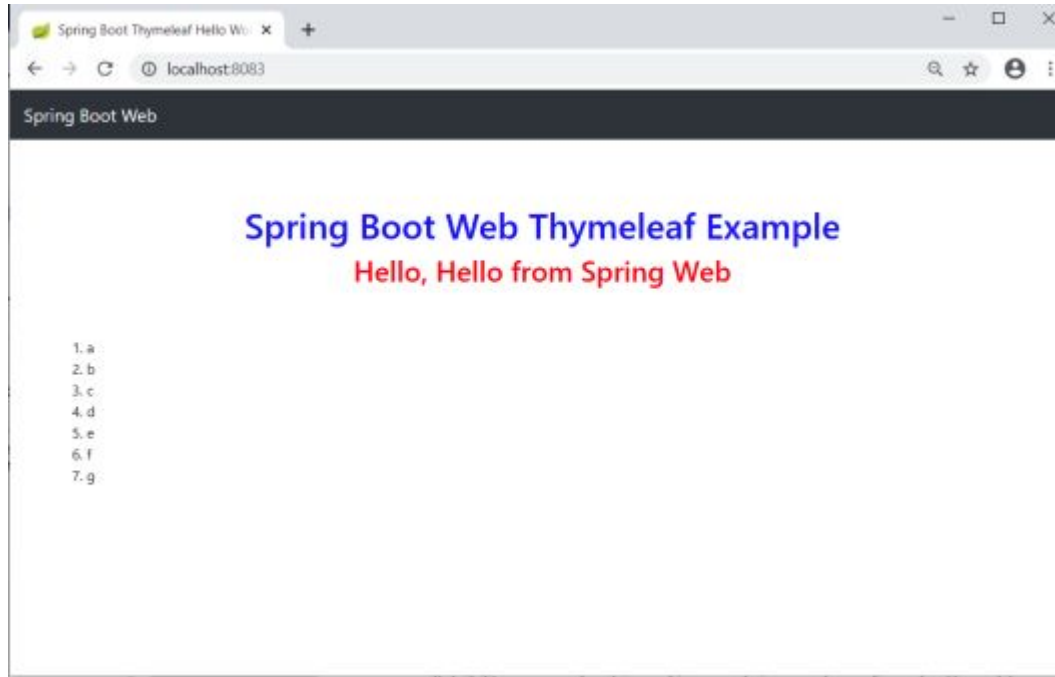
Dan buat file css nya dengan nama file main.css

```
body {  
  padding-top: 5rem;  
}  
.starter-template {  
  padding: 3rem 1.5rem;  
  text-align: center;  
}  
  
h1{  
  color:#0000FF;  
}  
  
h2{  
  color:#FF0000;  
}
```

Aplikasi ini akan kita jalankan menggunakan port 8083 untuk itu tambahkan konfigurasinya pada file application.properties.

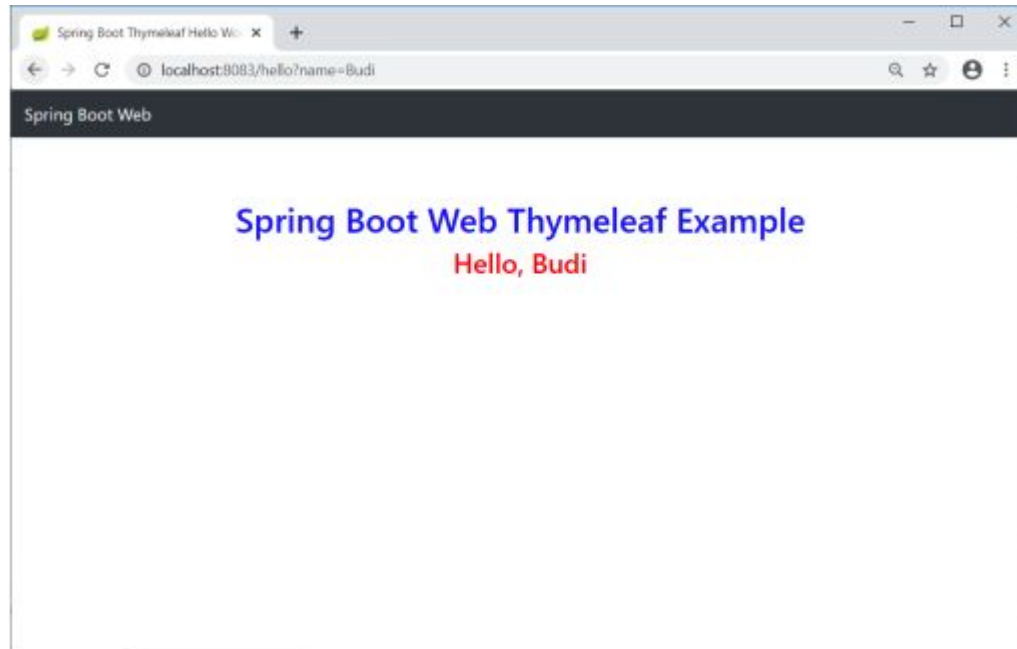
```
server.port=8083
```

jalankan aplikasi, kemudian akses menggunakan webbrowser dengan port 8083.



HACKTIV8

Apabila kita mengakses URL `localhost:8083/hello?name=Budi`, maka output nya akan seperti ini :



Springboot - Bootstrapping + Code Structures - Sesi 16

Spring Web + Bootstrap + Thymeleaf

Dependency spring-boot-starter-web dan terdapat dependency lagi yaitu **spring-boot-starter-thymeleaf**.

spring-boot-starter-thymeleaf digunakan sebagai view aplikasi web yang akan kita buat,
Other Ways bisa menggunakan jsp atau html.

```
<dependencies>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-web</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-test</artifactId>
    <scope>test</scope>
  </dependency>
</dependencies>
```



HTML Files

Selanjutnya kita akan coba buat file html sebagai view, silakan buat file html sederhana seperti ini. Misalkan saya beri nama index.html, file tersebut secara otomatis akan dicari oleh Spring Boot sebagai home screen aplikasi kita.

File tersebut dibuat di dalam folder resources/templates.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Selamat Datang Spring Boot</title>
</head>
<body>
<h2>Belajar Web Menggunakan Spring Boot</h2>
</body>
</html>
```



Silakan masukan alamat <http://localhost:8081/> seharusnya akan menampilkan html yang telah dibuat sebelumnya, port 8081 silakan disesuaikan pada file application.properties pada folder resources.

Dynamic Web

Sampai langkah ini sebenarnya sudah berhasil untuk membuat aplikasi web menggunakan Spring Boot, ya aplikasi web statis. Gimana jika aplikasi web dinamis? Sebelum membuat web dinamis, tambahkan terlebih dahulu dependensi berikut :

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <scope>runtime</scope>
</dependency>
<dependency>
    <groupId>org.projectlombok</groupId>
    <artifactId>lombok</artifactId>
</dependency>
```

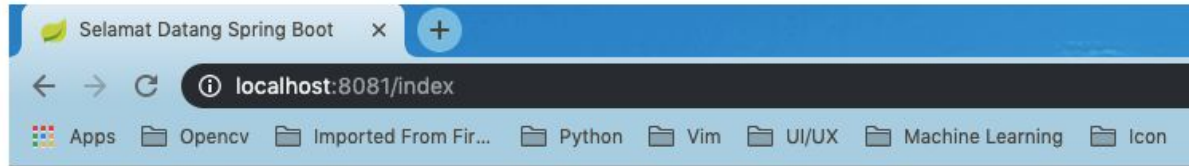




Latihan

+

TAMPILKAN DATA LIST MAHASISWA



Belajar Web Template Engine Thymeleaf Menggunakan Spring Boot

List Mahasiswa

Nim	Nama	IPK	Jurusan	Action
075410099	Noprianto	4.0	Teknologi Informasi	Edit Hapus Detail
075410101	Uwais	3.89	Teknologi Informasi	Edit Hapus Detail
075410100	Noureen Akhleema Shannum	3.99	Teknologi Informasi	Edit Hapus Detail

Step by Step

1. **Add Entity**
2. **Migrasi Database**
3. **Controller**
4. **Implement Thymeleaf**



Next : Menambahkan Entitas

Contoh project ini untuk melakukan manipulasi data mahasiswa sehingga yang dibutuhkan adalah entitas Mahasiswa, isinya seperti di bawah ini

```
@Entity
@Data
public class Mahasiswa {
    @Id
    @GeneratedValue(generator = "uuid2")
    @GenericGenerator(name = "uuid2", strategy = "uuid2")
    private String id;
    @NotNull
    @NotBlank
    @Column(unique = true)
    private String nim;
    @NotNull
    @NotBlank
    private String nama;
    private float ipk;
    @NotNull
    @NotBlank
    private String jurusan;
}
```



- @Entity, mendeklarasikan bahwa class Mahasiswa adalah berupa entitas yang akan di-mapping ke dalam sebuah tabel dalam database.
- @Data, menandai class Mahasiswa akan diberikan method getter dan setter
- @Id, @GeneratedValue(generator = "uuid2"), @GenericGenerator(name = "uuid2", strategy = "uuid2") digunakan untuk field id digunakan sebagai primary di database dengan nilai diisi menggunakan UUID.
- @Column(unique = true), maksudnya adalah agar field nim dibuat unik pada database
- @NotNull dan @NotBlank, field yang mendapatkan anotasi itu tidak boleh kosong ataupun NULL.

Next : Migrasi DB

Spring Boot dapat melakukan database migrasi menggunakan dependency Flyway. Terlebih dahulu kita tambahkan dependency tersebut pada file pom.xml seperti berikut ini

```
<dependency>
    <groupId>org.flywaydb</groupId>
    <artifactId>flyway-core</artifactId>
</dependency>
```



Pada file `application.properties` juga perlu ditambahkan lokasi file sql yang akan dieksekusi, konfigurasi seperti berikut ini :

```
spring.flyway.locations=classpath:db/migration
```

default pengecekan file sql terdapat di dalam folder `resources/db/migration`, tetapi kita bisa mengganti lokasi file sql tersebut dengan mengubah nilai pada `spring.flyway.locations=[lokasi file sql]`.

Jangan lupa menambahkan file sql tersebut pada direktori tersebut, selanjutnya untuk penamaan juga harus mengikuti flyway yaitu `V<VERSION>__<NAME>.sql`. Misalkan `V1.0__init_tables.sql`.

Selanjutnya silakan dijalankan seharusnya aplikasi akan membuatkan schema sesuai dengan file sql dan akan terbentuk tabel `flyway_schema_history` dalam database Anda.



Next : Controller

Controller dalam aplikasi web merupakan pintu masuk dari request client, tugasnya adalah meneruskan ke model atau mengembalikan request ke client.

Berikut ini adalah implementasi kode untuk controller tersebut

```
@Controller
public class MahasiswaController {
    @Autowired
    private MahasiswaDao mahasiswaDao;

    @GetMapping("/index")
    public ModelMap getAll(Pageable pageable) {
        return new ModelMap().addAttribute("mahasiswas", mahasiswaDao.findAll(pageable));
    }

    @GetMapping("/")
    public String index() {
        return "redirect:/index";
    }
}
```



Penjelasan kode di atas adalah sebagai berikut

- `@Controller`, menandai bahwa class `MahasiswaController` adalah sebuah controller yang akan dilakukan scan oleh Spring
- `@Autowired`, untuk meng-inisiasi objek `mahasiswaDao`.
- `@GetMapping("/index")`, method di bawahnya berarti menggunakan http request GET. Sedangkan parameter objek `pageable` agar support paging.
- `redirect:/index`, berarti ketika memanggil pada path "/" maka akan diteruskan ke path `"/index"` pada browser.



Next : Template Thymeleaf

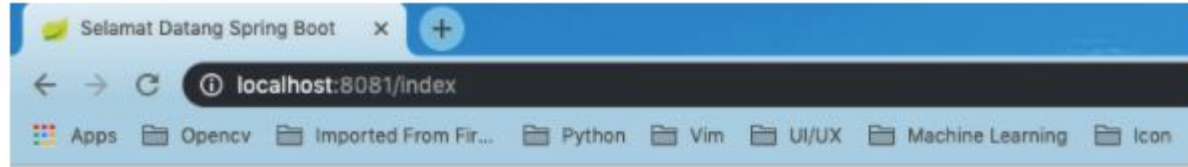
Template Thymeleaf merupakan salah satu template engine yang tersedia pada Spring Boot, karena ketika menggunakan jsp dianggap kurang menarik sehingga menggunakan template engine. File index.html yang sebelumnya perlu disesuaikan seperti berikut ini

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>Selamat Datang Spring Boot</title>
</head>
<body>
<h2>Belajar Web Template Engine Thymeleaf Menggunakan Spring Boot</h2>
<h2>List Mahasiswa</h2>

<table>
  <tr>
    <th>Nim</th>
    <th>Nama</th>
    <th>IPK</th>
    <th>Jurusan</th>
    <th>Action</th>
  </tr>
  <tr th:each="mhs : ${mahasiswas}">
    <td th:text="${mhs.nim}"> </td>
    <td th:text="${mhs.nama}"> </td>
    <td th:text="${mhs.ipk}"> </td>
    <td th:text="${mhs.jurusan}"> </td>
    <td>
      <a href="#">Edit</a> |
      <a href="#">Hapus</a> |
      <a href="#">Detail</a>
    </td>
  </tr>
</table>
</body>
</html>
```



Silakan dijalankan aplikasi yang telah dibuat, kemudian ketikan pada browser <http://localhost:8081>



Belajar Web Template Engine Thymeleaf Menggunakan Spring Boot

List Mahasiswa

Nim	Nama	IPK	Jurusan	Action
075410099	Noprianto	4.0	Teknologi Informasi	Edit Hapus Detail
075410101	Uwais	3.89	Teknologi Informasi	Edit Hapus Detail
075410100	Noureen Akhleema Shannum	3.99	Teknologi Informasi	Edit Hapus Detail

Thymeleaf Spring Boot