



链滴

# SpringBoot 页面展示 Thymeleaf

作者: [woyehua](#)

原文链接: <https://ld246.com/article/1543415588790>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<p></p>

<p>开发传统 Java WEB 工程时，我们可以使用 JSP 页面模板语言，但是在 SpringBoot 中已经不推荐使用了。SpringBoot 支持如下页面模板语言</p>

<p>Thymeleaf<br>

FreeMarker<br>

Velocity<br>

Groovy<br>

JSP<br>

上面并没有列举所有 SpringBoot 支持的页面模板技术。其中 Thymeleaf 是 SpringBoot 官方所推使用的，下面来谈谈 Thymeleaf 一些常用的语法规则。</p>

<p>添加 Thymeleaf 依赖</p>

<p>要想使用 Thymeleaf，首先要在 pom.xml 文件中单独添加 Thymeleaf 依赖。</p>

```
org.springframework.boot  
spring-boot-starter-thymeleaf
```

Spring Boot默认存放模板页面的路径在src/main/resources/templates或者src/main/view/templates，这个无论是使用什么模板语言都一样，当然默认路径是可以自定义的，不过一般不推荐这样做。外Thymeleaf默认的页面文件后缀是.html。

<p>数据显示</p>

<p>在 MVC 的开发过程中，我们经常需要通过 Controller 将数据传递到页面中，让页面进行动态示。</p>

<p>显示普通文本</p>

<p>创建一个 Controller 对象，在其中进行参数的传递</p>

<p>@Controller<br>

```
public class ThymeleafController {</p>
```

```
<pre> <code class="highlight-chroma"> <span class="highlight-line"> <span class="highlight-cl"> @RequestMapping(value = "show", method = RequestMethod.GET)
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> public String show  
Model model){
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     model.addAttribute("uid", "123456789");
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     model.addAttribute("name", "Jerry");
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl">     return "show";
```

```
</span> </span> <span class="highlight-line"> <span class="highlight-cl"> }</pre>
```

```
</span> </span> </code> </pre>
```

<p><br>

在 SpringBoot 默认的页面路径下创建 show.html 文件，内容如下</p>

### SpringBoot模版渲染

<p></p>

<p></p>

可以看到在p标签中有th:text属性，这个就是thymeleaf的语法，它表示显示一个普通的文本信息。

<p>如果我们要显示的信息是存在资源文件中的，同样可以在页面中显示，例如资源文件中定义了内  
welcome.msg=欢迎{0}光临！。可以在页面中将其显示</p>  
<h2></h2>  
另外，在th:utext中还能做一些基础的数学运算  
<p></p>  
显示带有样式的普通文本  
<p>如果我们想要传递到的页面的信息，它本身是带有 CSS 样式的，这个时候如何在页面中将携带  
样式信息也显示出来？此时我们的控制器方法这样写。</p>  
<p>@RequestMapping(value = "showStyle", method = RequestMethod.GET)<br>  
public String showStyle(Model model){<br>  
model.addAttribute("uid","123456789");<br>  
model.addAttribute("name","<span>Jerry</span>");<br>  
return "show\_style";<br>  
}<br>  
此时页面中需要借助 th:utext 属性进行显示</p>  
<p></p>  
通过浏览器查看页面源码可以看出th:utext和th:text的区别是：th:text会对&lt;和&gt;进行转义，而th  
utext不会转义。  
<p>显示对象</p>  
<p>我们常常需要将一个 bean 信息展示在前端页面当中。</p>  
<p>用于前端展示的 VO 类<br>  
public class User implements Serializable {<br>  
private Long uid ;<br>  
private String name ;<br>  
private Integer age ;<br>  
private Date birthday ;<br>  
private Double salary ;<br>  
//省略 get/set 方法<br>  
}<br>  
控制器方法<br>  
@RequestMapping(value = "/message/member\_show", method = RequestMethod.GET)<br>  
public String memberShow(Model model) {<br>  
User vo = new User();<br>  
vo.setUid(12345678L);<br>  
vo.setName("尼古拉丁.赵四");<br>  
vo.setAge(59);<br>  
vo.setSalary(1000.00);<br>  
vo.setBirthday(new Date());<br>  
model.addAttribute("member", vo);<br>  
return "message/member\_show";<br>  
}<br>  
页面展示</p>  
<div>  
    <p></p>  
    <p></p>  
    <p></p>  
    <p></p>  
    <p></p>  
    <p></p>  
</div>  
<hr>  
<div>  
    <p></p>  
    <p></p>

```
<p></p>
<p></p>
<p></p>
<p></p>
</div>
```

上面给出了两种展现方式，一种是通过\${属性}，另外一种是通过{属性}。

关于“\${属性}”和“{属性}”的区别？

访问完整信息，而访问指定对象中的属性内容，如果访问的只是普通的内容两者没有区别；

数据处理

在 thymeleaf 之中提供有相应的集合的处理方法，例如：在使用 List 集合的时候可以考虑采用 get()方法获取指定索引的数据，那么在使用 Set 集合的时候会考虑使用 contains()来判断某个数据是否存在，使用 Map 集合的时候也希望可以使用 containsKey()判断某个 key 是否存在，以及使用 get()根据 key 获取对应的 value，而这些功能在之前并不具备，下面来观察如何在页面中使用此类操作

控制器方法向页面传递一些数据，以供操作

```
@RequestMapping(value = "/user/set", method = RequestMethod.GET)
```

```
public String set(Model model) {
```

```
Set allNames = new HashSet();
```

```
List allIds = new ArrayList();
```

```
for (int x = 0 ; x < 5 ; x ++ ) {
```

```
allNames.add("boot-" + x);
```

```
allIds.add(x);
```

```
}
```

```
model.addAttribute("names", allNames);
```

```
model.addAttribute("ids", allIds);
```

```
model.addAttribute("mydate", new java.util.Date());
```

```
return "user_set";
```

```
}
```

数据处理