[Thymeleaf使用步骤 2](#_Toc712)

[Spring整合Thymeleaf 2](#_Toc2364)

[th标签 2](#_Toc21085)

[简单表达式 2](#_Toc15539)

[常用th标签 2](#_Toc23286)

[th标签内联 3](#_Toc7435)

[不常用 4](#_Toc8735)

[th属性 5](#_Toc2847)

## Thymeleaf使用步骤

1. 导包：thymeleaf
2. 增加头文件，<html xmlns="http://www.w3.org/1999/xhtml"

xmlns:th="http://www.thymeleaf.org">

3，使用th标签动态替换静态数据。<th th:text=”${message}”>hello</td>

## Spring整合Thymeleaf

导包spring-boot-starter-thymeleaf；更改头文件，替换静态数据。

## th标签

### 简单表达式

1，变量表达式，${...}

th:value = ${user.name}

2，选择表达式，\*{...}；一般跟在th:object后，读取其属性；

<div th:object=”{session.user}”> <span th:text = “\*{name}”></span></div>

3，[Url表达式，@{...}；](mailto:Url表达式，@{...}；th:href)

[th:href](mailto:Url表达式，@{...}；th:href) = [“@{/user/userDel(userId=${user.id})}”](mailto:\“@{/user/userDel(userId=${user.id})}\”)

4，文字国际化表达式，#{...}；国际化语句。

### **常用th标签**

1，简单数据转换(数字，日期)

th:text = “${#numbers.formatDecimal(user.weight,1,2)}”

th:text = “${#dates.format(user.birthday,”yyyy-MM-dd”)}”

2，字符串拼接

<span th:text = “${‘addr: ’+user.address}”> </span>

3，转义和非转义文本，th:text/th:utext；对文本数据进行转义/不转义处理。

This is an &lt;em&gt;HTML&lt;/em&gt；转义后为This is an <em>HTML</em>

4，表单中标签，th:action,th:method,th:field

<form th:action="@{/bb}" th:object="${user}" method="post" >

<input type="text" th:field="\*{name}"/>

<input type="text" th:field="\*{msg}"/>

<input type="submit"/>

</form>

5，显示页面的数据迭代；th:remove，th:each

//用 th:remove 移除除了第一个外的静态数据

<tbody th:remove="all-but-first">

//将后台传出的 userList 的集合进行迭代

<tr th:each="user:${userList}">

//用count进行统计，有顺序的显示

<td th:text="${user.count}">1</td>

<td th:text="${user.description}">Red Chair</td>

<td th:text="${'$'+ #numbers.formatDecimal(user.price, 1, 2}"></td>

</tr>

<tr>...</tr>

</tbody>

6，条件判断，{ }内不能使用<>，使用lt/gt替代；?运算符；th:if,th:switch/th:case

th:if = “${user.age lt 30}”

<!-- gender存在，选择对应的选项；gender不存在或为null，取name-->

<td th:switch="${customer.gender?.name()}">

<img th:case="'MALE'" src="..male.png" th:src="@{/male.png}" />

<img th:case="'FEMALE'" src="../female.png" th:src="@{/female.png}" />

<span th:case="\*">Unknown</span>

</td>

<!-- 增加class="enhanced"当balance大于10000 -->

<td th:class="${customer.balance gt 10000} ? 'enhanced'"

th:text="${customer.balance}">350</td>

7，根据后台数据选中select的选项

<select th:field="${resume.gender}" th:switch="${resumes.gender.toString()}">

<option value="男" th:case="'男'" th:selected="selected" >男</option>

<option value="女" th:case="'女'" th:selected="selected" >女</option>

<option value="">请选择</option>

</select>

8，spring表达式语言

th:text="${4 \* -6 \* -2 % 7}；算数表达式

th:text="${customerList[2].paymentMethod.description}"；Object navigation

th:text="${new java.util.Date().getTime()}"；Object instantiation

th:text="${T(java.lang.Math).random()}"；T operator

### th标签内联

1，文本内联

<textarea id="body" name="body" th:inline="text">

Dear [[${customerName}]]

</textarea>

1. 内联JS

--js起止加入如下代码，否则引号嵌套或者"<"">""&&”等不能用

/\*<![CDATA[\*/ …… /\*]]>\*/

　　--js附加代码：

/\*[+ var msg = 'This is a working application'; +]\*/

　　--js移除代码：

/\*[- \*/ var msg = 'This is a non-working template';/\* -]\*/

### 不常用

#### 表达式

1，文字

a)文本文字；'one text','Another one',……

b)数字文字；0,34,3.0,12.3，……

c)布尔文字；true，flase

d)空文字；null

e)文字标记；one，sometext，main，……

2，文本处理

a)字符串连接；+

b)文字替换；| The name id ${name} |

3，算术表达式

a)基本表达式；+，-，\*，/，%

b)减号（一元运算符）； -

4，布尔表达式

a)基本表达式；and，or

b)布尔否定（一元运算符）；！,not

5，比较和相等

a)比较；>，<，>=，<=(gt，lt，ge，le)

b)相等表达式；==，！=(eq，ne)

6，条件表达式

a)If-then；(if) ? (then)

b)If-then-else；(if) ? (then) : (else)

c)Default；(value) ? : (defaltvalue)

7，所有这些标签能够结合和嵌套：User is of type ' + (${user.isAdmin()} ? 'Administrator' : (${user.type} ?: 'Unknown'))

#### **表达式基本对象**

在上下文变量评估OGNL表达式时，一些对象表达式可获得更高的灵活性。这些对象将由#号开始引用。

#ctx: 上下文对象.

#vars: 上下文变量.

#locale: 上下文语言环境.

#httpServletRequest: (仅在web上文)HttpServletRequest对象.

#httpSession: (仅在web上文)  HttpSession 对象.

**表达式功能对象：**

#dates:java.util.Date对象的实用方法。

#calendars:和dates类似, 但是 java.util.Calendar 对象.

#numbers: 格式化数字对象的实用方法。

#strings: 字符创对象的实用方法：contains, startsWith, prepending/appending等.

#objects: 对objects操作的实用方法。

#bools: 对布尔值求值的实用方法。

#arrays: 数组的实用方法。

#lists: list的实用方法。

#sets: set的实用方法。

#maps: map的实用方法。

#aggregates: 对数组或集合创建聚合的实用方法。

#messages: 在表达式中获取外部信息的实用方法。

#ids: 处理可能重复的id属性的实用方法 (比如：迭代的结果)。

### th属性

给特定的属性设值，下面是用th:action给action设值。

<form action="subscribe.html" th:action="@{/subscribe}">

还有很多这样的属性，它们每一个都针对一个特定的XHTML或者HTML5属性。

th:text="${data}"；将data的值替换该属性所在标签的body。字符常量要用引号，比如th:text="'hello world'",th:text="2011+3",th:text="'my name is '+${user.name}"  
th:utext；和th:text的区别是"unescaped text"。  
th:with；定义变量，th:with="isEven=${prodStat.count}%2==0"，定义多个变量可以用逗号分隔。  
th:attr；设置标签属性，多个属性可以用逗号分隔，比如th:attr="src=@{/image/aa.jpg},title=#{logo}"，此标签不太优雅，一般用的比较少。  
th:[tagAttr]；设置标签的各个属性，比如th:value,th:action等可以一次设置两个属性，比如：th:alt-title="#{logo}"  
th:attrappend，th:attrprepend；对属性增加前缀和后缀，比如：th:attrappend="class=${' '+cssStyle}"  
对于属性是特定值的，比如checked属性，thymeleaf都采用bool值，比如th:checked=${user.isActive}  
th:each；循环，<tr th:each="user,userStat:${users}">,userStat是状态变量，有 index,count,size,current,even,odd,first,last等属性，如果没有显示设置状态变量，thymeleaf会默 认给个“变量名+Stat"的状态变量。  
th:if or th:unless；条件判断，支持布尔值，数字（非零为true)，字符，字符串等。  
th:switch，th:case；选择语句。 th:case="\*"表示default case。