

BraveWangDev

有分享,才有进步

☰ 目录视图

☰ 摘要视图

RSS 订阅

个人资料



BraveWangDev

关注

发私信

☰ 恒

访问：121965次

积分：2264

等级：BLOG > 5

排名：第18136名

原创：94篇

转载：1篇

译文：1篇

评论：43条

文章搜索

博客专栏



Java设计模式

文章：14篇

阅读：1756



Sass基础

文章：12篇

阅读：8170



SpringBoot从0开始

文章：11篇

阅读：36807



STM32学习笔记

文章：29篇

阅读：57079

文章分类

Java (1)

Android-自定义View (0)

ABAP (0)

SpringBoot-thymeleaf模板语法简介

标签：thymeleaf SpringBoot

2016-11-02 16:24

5233人阅读

评论(0)

☰ 分类：SpringBoot ( 11 ) ▼

目录(?)

[+]

本想简单说一下thymeleaf模板语法,因为毕竟后边SpringSecurity用到的语法很少,结果总结起来有点儿多...

关于SpringBoot-thymeleaf模板集成,请跳转：[SpringBoot-thymeleaf模板集成](#)

先说句有用的废话:  
thymeleaf模板语法,都以th属性开头,如:

1

<span th:text="...">

一,thymeleaf-简单表达式

1. 变量表达式

2. 选择或星号表达式

3. 文字国际化表达式

4. URL表达式

1,变量表达式

Thymeleaf模板引擎在进行模板渲染时，还会附带一个Context存放进行模板渲染的变量，在模板中定义的表达式本质上就是从Context中获取对应的变量的值

1

<p>Today is: <span th:text="\${day}">2 November 2016</span>.</p>

假设day的值为2016年11月2日，那么渲染结果为：<p>Today is: 2016年11月2日.</p>。  
注意：渲染后，模板中span值2 November 2016将被覆盖

2,选择(星号)表达式

可以简单理解为内层是对外层对象的引用

1

<div th:object="\${session.user}">

2

<p>Name: <span th:text="\*{firstName}">Sebastian</span>.</p>

3

<p>Surname: <span th:text="\*{lastName}">Pepper</span>.</p>

Web Dynpro for ABAP	(0)
Web Dynpro for Java	(0)
GOF_23种设计模式	(14)
51单片机	(0)
光立方	(0)
Html	(0)
CSS	(1)
JavaScript	(3)
AngularJs	(0)
Ionic	(0)
Cordova	(4)
FrameWork7	(0)
ReactNative	(0)
J2EE	(0)
STM32	(29)
MySql	(2)
Android	(2)
SpringBoot	(12)
Maven	(2)
redis	(1)
ActiveMQ	(5)
sass	(12)
Mybatis	(1)
Altium Designer	(2)
Vue	(5)
支付宝	(1)

文章存档	
2017年11月	(2)
2017年10月	(5)
2017年09月	(15)
2017年07月	(2)
2017年05月	(4)
展开	

阅读排行	
TFTLCD原理与驱动与指令介绍	(7071)
Maven构建SpringBoot项目	(6309)
SpringBoot-thymeleaf模板...	(5224)
SpringBoot-SpringSecurity...	(5059)
SpringBoot-SpringData-JPA...	(4938)
@Controller和@RestContro...	(4914)
SpringBoot-JUnit	(4278)
OLED原理,时序和操作	(4082)
通用定时器-输入捕获实验	(3247)
OLED显示-字库码表原理	(3029)

评论排行	
独立看门狗	(5)
TFTLCD原理与驱动与指令介绍	(5)
OLED显示-字库码表原理	(5)
3.单例模式	(4)
SpringBoot-SpringSecurity...	(4)
SpringBoot-SpringData-JPA...	(3)
@Controller和@RestContro...	(2)

```
4      <p>Nationality: <span th:text="*{nationality}">Saturn</span>. </p>
5  </div>
```

等同于以下方式:

```
1  <div>
2      <p>Name: <span th:text="${session.user.firstName}">Sebastian</span>. </p>
3      <p>Surname: <span th:text="${session.user.lastName}">Pepper</span>. </p>
4      <p>Nationality: <span th:text="${session.user.nationality}">Saturn</span>. </p>
5  </div>
```

也可以混用,如下:

```
1  <div th:object="${session.user}">
2      <p>Name: <span th:text="*{firstName}">Sebastian</span>. </p>
3      <p>Surname: <span th:text="${session.user.lastName}">Pepper</span>. </p>
4      <p>Nationality: <span th:text="*{nationality}">Saturn</span>. </p>
5  </div>
```

如何没有与th:object结合使用,\*{}与\${}效果一样,因为其范围自动扩展到context。

3,URL表达式

URL表达式指的是把一个有用的上下文或会话信息添加到URL,这个过程经常被叫做URL重写。

Thymeleaf对于URL的处理是通过语法@{...}来处理的

```
1  <!-- 绝对路径 -->
2  <!-- Will produce 'http://localhost:8080/gtvg/order/details?orderId=3' (plus rewriting) -->
3  <a href="details.html" th:href="@{http://localhost:8080/gtvg/order/details(orderId=${o.id})}">v
4
5  <!-- 相对路径 带参数-->
6  <!-- Will produce '/gtvg/order/details?orderId=3' (plus rewriting) -->
7  <a href="details.html" th:href="@{/order/details(orderId=${o.id})}">view</a>
8
9  <!-- Will produce '/gtvg/order/3/details' (plus rewriting) -->
10 <a href="details.html" th:href="@{/order/{orderId}/details(orderId=${o.id})}">view</a>
```

Thymeleaf支持相对路径和绝对路径  
(orderId=\${o.id})表示将括号内的内容作为URL参数处理  
@{...}表达式中可以通过{orderId}访问Context中的orderId变量  
@{/order}是Context相关的相对路径,在渲染时会自动添加上当前Web应用的Context名字,假设context名字为app,那么

4,文字国际化表达式

文字国际化表达式允许我们从一个外部文件获取区域文字信息(.properties)  
使用Key-Value方式,还可以提供一组参数(可选).

.properties

```
1  #{main.title}
2  #{message.entrycreated(${entryId})}
```

模板引用:

通用定时器PWM输出实验	(2)
通用定时器-输入捕获实验	(2)
FSMC驱动TFTLCD原理,时序...	(2)

推荐文章

- \* 【GitChat】每日精选20171109——双 11 大前端工程师读书清单
- \* 改做人工智能之前，90%的人都没能给自己定位
- \* 想入行 AI，别让那些技术培训坑了你...
- \* 大型技术组织 DevOps 转型经验总结
- \* 两款敏捷工具，治好你碎片化交付硬伤
- \* 低学历又如何？这样的程序员照样可以逆袭

最新评论

3,单例模式  
BraveWangDev : @wyjbird5:相互学习,共同进步

MySql-moji表情引发的存储异常-微信昵称wyjbird5 : 唉 我用的笨办法 我直接给替换成空字符串儿了...

3,单例模式  
BraveWangDev : @dulei294948:多谢大神点评,受益匪浅

TFTLCD原理与驱动与指令介绍  
BraveWangDev : @weixin\_40143470:多谢支持!

TFTLCD原理与驱动与指令介绍  
weixin\_40143470 : @ABAP\_Brave:支持博主

独立看门狗  
BraveWangDev : @qq\_37130151:你的理解是正确的,IWDG\_RLR计数器重装载值,每经过一个看门狗时钟周期...

SpringBoot-SpringData-JPA集成  
BraveWangDev : @Spring5945:DD的这个例子本来写的就不错,简单易懂,也没想过要改Test,就是拿这个例子...

Sass的嵌套  
BraveWangDev : @qq\_15071769:你好,工作原因多日没有登陆CSDN,如按照我的教程不能正常实现,请详细说明...

SpringBoot-SpringSecurity集成  
BraveWangDev : @qq\_29720067:多谢关注,互相学习

TFTLCD原理与驱动与指令介绍  
BraveWangDev : @yangxin\_no:谢谢你的留言,我没有不尊重原创,这是我个人的学习记录,需要的时候可以查阅,毕...

```
1 <table>
2   <th th:text="#{header.address.city}">...</th>
3   <th th:text="#{header.address.country}">...</th>
4 </table>
```

## 二.thymeleaf-字面值

- 1.文本文字：‘ one text’ , ‘Another one!’ ,...
- 2.文字数量：0, 34, 3.0, 12.3,...
- 3.布尔型常量：true, false
- 4.空的文字：null
- 5.文字标记：one, sometext, main,...

## 三：thymeleaf-文本处理

### 1.字符串拼接：+

```
1 <span th:text="'Welcome to our application, ' + ${user.name} + '!' ">
```

### 2.文字替换：|The name is \${name}|

```
1 <span th:text="'Welcome to our application, ${user.name}!' ">
```

相比以上两种方式都可以实现字符串合并,但是,|...|中只能包含变量表达式\${...}, 不能包含其他常量、条件表达式等。

## 四.表达基本对象

- 1.#ctx:上下文对象
- 2.#vars:上下文变量
- 3.#locale:上下文语言环境
- 4.#HttpServletRequest:(只有在Web上下文)HttpServletRequest对象
- 5.#HttpSession:(只有在Web上下文)HttpSession对象。

例如:

```
1 <span th:text="#{#locale.country}">US</span>.
2
3 th:text="#{#calendars.format(today,' dd MMMM yyyy')}"
```

# 五,表达式预处理

表达式预处理，它被定义在\_之间：

```
1 | #{selection.__${sel.code}__}
```

`${sel.code}` 将先被执行，结果(假如是AAA)将被看做表达式的一部分被执行  
结果#(为selection.AAA)。

# 六,thymeleaf运算符

在表达式中可以使用各类算术运算符，例如+, -, \*, /, %

```
1 | th:with="isEven=(${prodStat.count} % 2 == 0)"
```

逻辑运算符>, <, <=, >=, ==, !=都可以使用  
需要注意的是使用 >, <, >=, <=时需要用它的HTML转义符(> gt; < lt; >= ge; gte; <= le; lte; == eq; != ne; neq;)

```
1 | th:if="${prodStat.count} > 1"
2 | th:text="' Execution mode is ' + ( ${execMode} == 'dev')? 'Development' : 'Production' )"
```

布尔运算符 and,or

# 七,thymeleaf循环

数据集必须是可以遍历的，使用th:each标签：

```
1 | <body>
2 |   <h1>Product list</h1>
3 |
4 |   <table>
5 |     <tr>
6 |       <th>NAME</th>
7 |       <th>PRICE</th>
8 |       <th>IN STOCK</th>
9 |     </tr>
10 |     <tr th:each="prod : ${prods}">
11 |       <td th:text="${prod.name}">Onions</td>
12 |       <td th:text="${prod.price}">2.41</td>
13 |       <td th:text="${prod.inStock}? #{true} : #{false}">yes</td>
14 |     </tr>
15 |   </table>
16 |
17 |   <p>
18 |     <a href="../home.html" th:href="@{/}">Return to home</a>
19 |   </p>
20 | </body>
```

被循环渲染的元素<tr>中加入th:each标签  
th:each="prod : \${prods}"对集合变量prods进行遍历，对象prod在循环体中可通过表达式访问

## 八,thymeleaf条件求值

### 1,If/Unless

Thymeleaf中使用th:if和th:unless属性进行条件判断

设置标签只有在th:if中条件成立时才显示:

```
1 <a th:href="@{/login}" th:unless="${session.user != null}">Login</a>
```

th:unless与th:if相反, 表达式条件不成立时显示内容。

### 2,Switch

多路选择Switch结构,默认属性default,用\*表示

```
1 <div th:switch="${user.role}">
2   <p th:case="'admin'">User is an administrator</p>
3   <p th:case="#{roles.manager}">User is a manager</p>
4   <p th:case="*">User is some other thing</p>
5 </div>
```

### 3.If-then-else: (if)?(then):else 三元运算符

三元运算控制class属性选择

```
1 <tr th:class="${row.even}? 'even' : 'odd'">
```

三元运算嵌套

```
1 <tr th:class="${row.even}? (${row.first}? 'first' : 'even') : 'odd'">
```

还可以省略else部分, 当表达式结果为false, 返回null, 否则返回' alt'

```
1 <tr th:class="${row.even}? 'alt'">
2   ...
3 </tr>
```

### 4.If-then: (if) ? (then), 省略了else部分, 如果条件不成立, 返回null

如果第一个表达式的计算结果为null, 则取第二个表达式的结果

```
1 <div th:object="${session.user}">
2   <p>Age: <span th:text="{age}?: ' (no age specified)'">27</span>.</p>
3 </div>
```

等效于:

```
1 <p>Age: <span th:text="{age != null}? {age} : ' (no age specified)'">27</span>.</p>
```

条件表达式嵌套:

```
1 <p>Name: <span th:text="*{firstName} ?: (*{admin} ? 'Admin' : #{default.username})">Sebastian</p>
```

# 九,Thymeleaf-Utilities

Thymeleaf提供了套Utility对象,内置于Context中,可通过#直接访问：

- #dates: java.util的实用方法。对象:日期格式、组件提取等。
- #calendars: 类似于#日期,但对于java.util。日历对象
- #numbers: 格式化数字对象的实用方法。
- #strings: 字符串对象的实用方法:包含startsWith, 将/附加等。
- #objects: 实用方法的对象。
- #bools: 布尔评价的实用方法。
- #arrays: 数组的实用方法。
- #lists: list集合。
- #sets:set集合。
- #maps: map集合。
- #aggregates: 实用程序方法用于创建聚集在数组或集合。
- #messages: 实用程序方法获取外部信息内部变量表达式,以同样的方式,因为他们将获得使用# {...} 语法
- #ids: 实用程序方法来处理可能重复的id属性(例如,由于迭代)。

## Dates

```
1 #dates : utility methods for java.util.Date objects:
2 /*
3  * =====
4  * See javadoc API for class org.thymeleaf.expression.Dates
5  * =====
6  */
7
8 /*
9  * Null-safe toString()
10 */
11 ${#strings.toString(obj)} // also array*, list* and set*
12
13 /*
14  * Format date with the standard locale format
15  * Also works with arrays, lists or sets
16 */
17 ${#dates.format(date)}
18 ${#dates.arrayFormat(datesArray)}
19 ${#dates.listFormat(datesList)}
20 ${#dates.setFormat(datesSet)}
21
22 /*
23  * Format date with the specified pattern
24  * Also works with arrays, lists or sets
25 */
26 ${#dates.format(date, 'dd/MM/yyyy HH:mm')}
27 ${#dates.arrayFormat(datesArray, 'dd/MM/yyyy HH:mm')}
28 ${#dates.listFormat(datesList, 'dd/MM/yyyy HH:mm')}
29 ${#dates.setFormat(datesSet, 'dd/MM/yyyy HH:mm')}
30
31 /*
32  * Obtain date properties
33  * Also works with arrays, lists or sets
34 */
35 ${#dates.day(date)} // also arrayDay(...), listDay(...), etc.
36 ${#dates.month(date)} // also arrayMonth(...), listMonth(...), etc.
37 ${#dates.monthName(date)} // also arrayMonthName(...), listMonthName(...), etc.
```

```

38 | ${#dates.monthNameShort(date)}           // also arrayMonthNameShort(...), listMonthNameShort(...)
39 | ${#dates.year(date)}                     // also arrayYear(...), listYear(...), etc.
40 | ${#dates.dayOfWeek(date)}                // also arrayDayOfWeek(...), listDayOfWeek(...), etc.
41 | ${#dates.dayOfWeekName(date)}            // also arrayDayOfWeekName(...), listDayOfWeekName(...),
42 | ${#dates.dayOfWeekNameShort(date)}       // also arrayDayOfWeekNameShort(...), listDayOfWeekNameS
43 | ${#dates.hour(date)}                     // also arrayHour(...), listHour(...), etc.
44 | ${#dates.minute(date)}                   // also arrayMinute(...), listMinute(...), etc.
45 | ${#dates.second(date)}                   // also arraySecond(...), listSecond(...), etc.
46 | ${#dates.millisecond(date)}              // also arrayMillisecond(...), listMillisecond(...), etc
47 |
48 | /*
49 |  * Create date (java.util.Date) objects from its components
50 | */
51 | ${#dates.create(year,month,day)}
52 | ${#dates.create(year,month,day,hour,minute)}
53 | ${#dates.create(year,month,day,hour,minute,second)}
54 | ${#dates.create(year,month,day,hour,minute,second,millisecond)}
55 |
56 | /*
57 |  * Create a date (java.util.Date) object for the current date and time
58 | */
59 | ${#dates.createNow()}
60 |
61 | /*
62 |  * Create a date (java.util.Date) object for the current date (time set to 00:00)
63 | */
64 | ${#dates.createToday()}

```

## Calendars

```

1 | #calendars : analogous to #dates, but for java.util.Calendar objects:
2 | /*
3 | * =====
4 | * See javadoc API for class org.thymeleaf.expression.Calendars
5 | * =====
6 | */
7 |
8 | /*
9 | * Format calendar with the standard locale format
10 | * Also works with arrays, lists or sets
11 | */
12 | ${#calendars.format(cal)}
13 | ${#calendars.arrayFormat(calArray)}
14 | ${#calendars.listFormat(callList)}
15 | ${#calendars.setFormat(calSet)}
16 |
17 | /*
18 | * Format calendar with the specified pattern
19 | * Also works with arrays, lists or sets
20 | */
21 | ${#calendars.format(cal, 'dd/MMM/yyyy HH:mm')}
22 | ${#calendars.arrayFormat(calArray, 'dd/MMM/yyyy HH:mm')}
23 | ${#calendars.listFormat(callList, 'dd/MMM/yyyy HH:mm')}
24 | ${#calendars.setFormat(calSet, 'dd/MMM/yyyy HH:mm')}
25 |
26 | /*
27 | * Obtain calendar properties
28 | * Also works with arrays, lists or sets
29 | */
30 | ${#calendars.day(date)}                 // also arrayDay(...), listDay(...), etc.

```

```

31 | ${#calendars.month(date)}           // also arrayMonth(...), listMonth(...), etc.
32 | ${#calendars.monthName(date)}       // also arrayMonthName(...), listMonthName(...), etc.
33 | ${#calendars.monthNameShort(date)}  // also arrayMonthNameShort(...), listMonthNameShort(...)
34 | ${#calendars.year(date)}            // also arrayYear(...), listYear(...), etc.
35 | ${#calendars.dayOfWeek(date)}       // also arrayDayOfWeek(...), listDayOfWeek(...), etc.
36 | ${#calendars.dayOfWeekName(date)}   // also arrayDayOfWeekName(...), listDayOfWeekName(...),
37 | ${#calendars.dayOfWeekNameShort(date)} // also arrayDayOfWeekNameShort(...), listDayOfWeekNameShort(...)
38 | ${#calendars.hour(date)}            // also arrayHour(...), listHour(...), etc.
39 | ${#calendars.minute(date)}          // also arrayMinute(...), listMinute(...), etc.
40 | ${#calendars.second(date)}          // also arraySecond(...), listSecond(...), etc.
41 | ${#calendars.millisecond(date)}     // also arrayMillisecond(...), listMillisecond(...) etc
42 |
43 | /*
44 |  * Create calendar (java.util.Calendar) objects from its components
45 |  */
46 | ${#calendars.create(year, month, day)}
47 | ${#calendars.create(year, month, day, hour, minute)}
48 | ${#calendars.create(year, month, day, hour, minute, second)}
49 | ${#calendars.create(year, month, day, hour, minute, second, millisecond)}
50 |
51 | /*
52 |  * Create a calendar (java.util.Calendar) object for the current date and time
53 |  */
54 | ${#calendars.createNow()}
55 |
56 | /*
57 |  * Create a calendar (java.util.Calendar) object for the current date (time set to 00:00)
58 |  */
59 | ${#calendars.createToday()}

```

## Numbers

```

1 | #numbers : utility methods for number objects:
2 | /*
3 |  * =====
4 |  * See javadoc API for class org.thymeleaf.expression.Numbers
5 |  * =====
6 |  */
7 |
8 | /*
9 |  * =====
10 |  * Formatting integer numbers
11 |  * =====
12 |  */
13 |
14 | /*
15 |  * Set minimum integer digits.
16 |  * Also works with arrays, lists or sets
17 |  */
18 | ${#numbers.formatInteger(num, 3)}
19 | ${#numbers.arrayFormatInteger(numArray, 3)}
20 | ${#numbers.listFormatInteger(numList, 3)}
21 | ${#numbers.setFormatInteger(numSet, 3)}
22 |
23 | /*
24 |  * Set minimum integer digits and thousands separator:
25 |  * 'POINT', 'COMMA', 'NONE' or 'DEFAULT' (by locale).
26 |  * Also works with arrays, lists or sets
27 |  */
28 | ${#numbers.formatInteger(num, 3, 'POINT')}

```



```

29  ${#numbers.arrayFormatInteger(numArray, 3, 'POINT')}
30  ${#numbers.listFormatInteger(numList, 3, 'POINT')}
31  ${#numbers.setFormatInteger(numSet, 3, 'POINT')}
32
33  /*
34   * =====
35   * Formatting decimal numbers
36   * =====
37   */
38
39  /*
40   * Set minimum integer digits and (exact) decimal digits.
41   * Also works with arrays, lists or sets
42   */
43  ${#numbers.formatDecimal(num, 3, 2)}
44  ${#numbers.arrayFormatDecimal(numArray, 3, 2)}
45  ${#numbers.listFormatDecimal(numList, 3, 2)}
46  ${#numbers.setFormatDecimal(numSet, 3, 2)}
47
48  /*
49   * Set minimum integer digits and (exact) decimal digits, and also decimal separator.
50   * Also works with arrays, lists or sets
51   */
52  ${#numbers.formatDecimal(num, 3, 2, 'COMMA')}
53  ${#numbers.arrayFormatDecimal(numArray, 3, 2, 'COMMA')}
54  ${#numbers.listFormatDecimal(numList, 3, 2, 'COMMA')}
55  ${#numbers.setFormatDecimal(numSet, 3, 2, 'COMMA')}
56
57  /*
58   * Set minimum integer digits and (exact) decimal digits, and also thousands and
59   * decimal separator.
60   * Also works with arrays, lists or sets
61   */
62  ${#numbers.formatDecimal(num, 3, 'POINT', 2, 'COMMA')}
63  ${#numbers.arrayFormatDecimal(numArray, 3, 'POINT', 2, 'COMMA')}
64  ${#numbers.listFormatDecimal(numList, 3, 'POINT', 2, 'COMMA')}
65  ${#numbers.setFormatDecimal(numSet, 3, 'POINT', 2, 'COMMA')}
66  /*
67   * =====
68   * Utility methods
69   * =====
70   */
71
72  /*
73   * Create a sequence (array) of integer numbers going
74   * from x to y
75   */
76  ${#numbers.sequence(from, to)}
77  ${#numbers.sequence(from, to, step)}

```

## Strings

```

1  #strings : utility methods for String objects:
2  /*
3   * =====
4   * See javadoc API for class org.thymeleaf.expression.Strings
5   * =====
6   */
7
8  /*
9   * Check whether a String is empty (or null). Performs a trim() operation before check

```

```
10  * Also works with arrays, lists or sets
11  */
12  ${#strings.isEmpty(name)}
13  ${#strings.arrayIsEmpty(nameArr)}
14  ${#strings.listIsEmpty(nameList)}
15  ${#strings.setIsEmpty(nameSet)}
16
17  /*
18  * Perform an 'isEmpty()' check on a string and return it if false, defaulting to
19  * another specified string if true.
20  * Also works with arrays, lists or sets
21  */
22  ${#strings.defaultString(text, default)}
23  ${#strings.arrayDefaultString(textArr, default)}
24  ${#strings.listDefaultString(textList, default)}
25  ${#strings.setDefaultString(textSet, default)}
26
27  /*
28  * Check whether a fragment is contained in a String
29  * Also works with arrays, lists or sets
30  */
31  ${#strings.contains(name, 'ez')} // also array*, list* and set*
32  ${#strings.containsIgnoreCase(name, 'ez')} // also array*, list* and set*
33
34  /*
35  * Check whether a String starts or ends with a fragment
36  * Also works with arrays, lists or sets
37  */
38  ${#strings.startsWith(name, 'Don')} // also array*, list* and set*
39  ${#strings.endsWith(name, endingFragment)} // also array*, list* and set*
40
41  /*
42  * Substring-related operations
43  * Also works with arrays, lists or sets
44  */
45  ${#strings.indexOf(name, frag)} // also array*, list* and set*
46  ${#strings.substring(name, 3, 5)} // also array*, list* and set*
47  ${#strings.substringAfter(name, prefix)} // also array*, list* and set*
48  ${#strings.substringBefore(name, suffix)} // also array*, list* and set*
49  ${#strings.replace(name, 'las', 'ler')} // also array*, list* and set*
50
51  /*
52  * Append and prepend
53  * Also works with arrays, lists or sets
54  */
55  ${#strings.prepend(str, prefix)} // also array*, list* and set*
56  ${#strings.append(str, suffix)} // also array*, list* and set*
57
58  /*
59  * Change case
60  * Also works with arrays, lists or sets
61  */
62  ${#strings.toUpperCase(name)} // also array*, list* and set*
63  ${#strings.toLowerCase(name)} // also array*, list* and set*
64
65  /*
66  * Split and join
67  */
68  ${#strings.arrayJoin(namesArray, ',')}
69  ${#strings.listJoin(namesList, ',')}
70  ${#strings.setJoin(namesSet, ',')}
71  ${#strings.arraySplit(namesStr, ',')} // returns String[]
```

```

72  ${#strings.listSplit(namesStr,','')}           // returns List<String>
73  ${#strings.setSplit(namesStr,','')}           // returns Set<String>
74
75  /*
76   * Trim
77   * Also works with arrays, lists or sets
78   */
79  ${#strings.trim(str)}                         // also array*, list* and set*
80
81  /*
82   * Compute length
83   * Also works with arrays, lists or sets
84   */
85  ${#strings.length(str)}                      // also array*, list* and set*
86
87  /*
88   * Abbreviate text making it have a maximum size of n. If text is bigger, it
89   * will be clipped and finished in "...".
90   * Also works with arrays, lists or sets
91   */
92  ${#strings.abbreviate(str,10)}               // also array*, list* and set*
93
94  /*
95   * Convert the first character to upper-case (and vice-versa)
96   */
97  ${#strings.capitalize(str)}                  // also array*, list* and set*
98  ${#strings.unCapitalize(str)}                // also array*, list* and set*
99
100 /*
101  * Convert the first character of every word to upper-case
102  */
103  ${#strings.capitalizeWords(str)}             // also array*, list* and set*
104  ${#strings.capitalizeWords(str,delimiters)}  // also array*, list* and set*
105
106 /*
107  * Escape the string
108  */
109  ${#strings.escapeXml(str)}                   // also array*, list* and set*
110  ${#strings.escapeJava(str)}                  // also array*, list* and set*
111  ${#strings.escapeJavaScript(str)}            // also array*, list* and set*
112  ${#strings.unescapeJava(str)}                // also array*, list* and set*
113  ${#strings.unescapeJavaScript(str)}          // also array*, list* and set*
114
115 /*
116  * Null-safe comparison and concatenation
117  */
118  ${#strings.equals(str)}
119  ${#strings.equalsIgnoreCase(str)}
120  ${#strings.concat(str)}
121  ${#strings.concatReplaceNulls(str)}
122
123 /*
124  * Random
125  */
126  ${#strings.randomAlphanumeric(count)}

```

## Objects

```

1  #objects : utility methods for objects in general
2  /*
3  * =====

```

```
4  * See javadoc API for class org.thymeleaf.expression.Objects
5  * =====
6  */
7
8  /*
9  * Return obj if it is not null, and default otherwise
10 * Also works with arrays, lists or sets
11 */
12 ${#objects.nullSafe(obj,default)}
13 ${#objects.arrayNullSafe(objArray,default)}
14 ${#objects.listNullSafe(objList,default)}
15 ${#objects.setNullSafe(objSet,default)}
```

## Booleans

```
1  #bools : utility methods for boolean evaluation
2  /*
3  * =====
4  * See javadoc API for class org.thymeleaf.expression.Bools
5  * =====
6  */
7
8  /*
9  * Evaluate a condition in the same way that it would be evaluated in a th:if tag
10 * (see conditional evaluation chapter afterwards).
11 * Also works with arrays, lists or sets
12 */
13 ${#bools.isTrue(obj)}
14 ${#bools.arrayIsTrue(objArray)}
15 ${#bools.listIsTrue(objList)}
16 ${#bools.setIsTrue(objSet)}
17
18 /*
19 * Evaluate with negation
20 * Also works with arrays, lists or sets
21 */
22 ${#bools.isFalse(cond)}
23 ${#bools.arrayIsFalse(condArray)}
24 ${#bools.listIsFalse(condList)}
25 ${#bools.setIsFalse(condSet)}
26
27 /*
28 * Evaluate and apply AND operator
29 * Receive an array, a list or a set as parameter
30 */
31 ${#bools.arrayAnd(condArray)}
32 ${#bools.listAnd(condList)}
33 ${#bools.setAnd(condSet)}
34
35 /*
36 * Evaluate and apply OR operator
37 * Receive an array, a list or a set as parameter
38 */
39 ${#bools.arrayOr(condArray)}
40 ${#bools.listOr(condList)}
41 ${#bools.setOr(condSet)}
```

## Arrays

```
1 #arrays : utility methods for arrays
2 /*
3  * =====
4  * See javadoc API for class org.thymeleaf.expression.Arrays
5  * =====
6  */
7
8 /*
9  * Converts to array, trying to infer array component class.
10  * Note that if resulting array is empty, or if the elements
11  * of the target object are not all of the same class,
12  * this method will return Object[].
13  */
14 ${#arrays.toArray(object)}
15
16 /*
17  * Convert to arrays of the specified component class.
18  */
19 ${#arrays.toStringArray(object)}
20 ${#arrays.toIntegerArray(object)}
21 ${#arrays.toLongArray(object)}
22 ${#arrays.toDoubleArray(object)}
23 ${#arrays.toFloatArray(object)}
24 ${#arrays.toBooleanArray(object)}
25
26 /*
27  * Compute length
28  */
29 ${#arrays.length(array)}
30
31 /*
32  * Check whether array is empty
33  */
34 ${#arrays.isEmpty(array)}
35
36 /*
37  * Check if element or elements are contained in array
38  */
39 ${#arrays.contains(array, element)}
40 ${#arrays.containsAll(array, elements)}
```

## Lists

```
1 #lists : utility methods for lists
2 /*
3  * =====
4  * See javadoc API for class org.thymeleaf.expression.Lists
5  * =====
6  */
7
8 /*
9  * Converts to list
10  */
11 ${#lists.toList(object)}
12
13 /*
14  * Compute size
15  */
16 ${#lists.size(list)}
17
18 /*
```

```
19      * Check whether list is empty
20      */
21      ${#lists.isEmpty(list)}
22
23      /*
24      * Check if element or elements are contained in list
25      */
26      ${#lists.contains(list, element)}
27      ${#lists.containsAll(list, elements)}
28
29      /*
30      * Sort a copy of the given list. The members of the list must implement
31      * comparable or you must define a comparator.
32      */
33      ${#lists.sort(list)}
34      ${#lists.sort(list, comparator)}
```

Sets

```
1 #sets : utility methods for sets
2 /*
3  * =====
4  * See javadoc API for class org.thymeleaf.expression.Sets
5  * =====
6  */
7
8  /*
9  * Converts to set
10 */
11 ${#sets.toSet(object)}
12
13 /*
14 * Compute size
15 */
16 ${#sets.size(set)}
17
18 /*
19 * Check whether set is empty
20 */
21 ${#sets.isEmpty(set)}
22
23 /*
24 * Check if element or elements are contained in set
25 */
26 ${#sets.contains(set, element)}
27 ${#sets.containsAll(set, elements)}
```

Maps

```
1 #maps : utility methods for maps
2 /*
3  * =====
4  * See javadoc API for class org.thymeleaf.expression.Maps
5  * =====
6  */
7
8  /*
9  * Compute size
10 */
11 ${#maps.size(map)}
```

```

12
13  /*
14   * Check whether map is empty
15   */
16  ${#maps.isEmpty(map)}
17
18  /*
19   * Check if key/s or value/s are contained in maps
20   */
21  ${#maps.containsKey(map, key)}
22  ${#maps.containsAllKeys(map, keys)}
23  ${#maps.containsValue(map, value)}
24  ${#maps.containsAllValues(map, value)}

```

## Aggregates

```

1  #aggregates : utility methods for creating aggregates on arrays or collections
2  /*
3   * =====
4   * See javadoc API for class org.thymeleaf.expression.Aggregates
5   * =====
6   */
7
8  /*
9   * Compute sum. Returns null if array or collection is empty
10  */
11  ${#aggregates.sum(array)}
12  ${#aggregates.sum(collection)}
13
14  /*
15   * Compute average. Returns null if array or collection is empty
16  */
17  ${#aggregates.avg(array)}
18  ${#aggregates.avg(collection)}

```

## Messages

```

1  #messages : utility methods for obtaining externalized messages inside variables expressions, i
2  /*
3   * =====
4   * See javadoc API for class org.thymeleaf.expression.Messages
5   * =====
6   */
7
8  /*
9   * Obtain externalized messages. Can receive a single key, a key plus arguments,
10  * or an array/list/set of keys (in which case it will return an array/list/set of
11  * externalized messages).
12  * If a message is not found, a default message (like '??msgKey??') is returned.
13  */
14  ${#messages.msg('msgKey')}
15  ${#messages.msg('msgKey', param1)}
16  ${#messages.msg('msgKey', param1, param2)}
17  ${#messages.msg('msgKey', param1, param2, param3)}
18  ${#messages.msgWithParams('msgKey', new Object[] {param1, param2, param3, param4})}
19  ${#messages.arrayMsg(messageKeyArray)}
20  ${#messages.listMsg(messageKeyList)}
21  ${#messages.setMsg(messageKeySet)}
22
23  /*

```

```
24 * Obtain externalized messages or null. Null is returned instead of a default
25 * message if a message for the specified key is not found.
26 */
27 ${#messages.msgOrNull('msgKey')}
28 ${#messages.msgOrNull('msgKey', param1)}
29 ${#messages.msgOrNull('msgKey', param1, param2)}
30 ${#messages.msgOrNull('msgKey', param1, param2, param3)}
31 ${#messages.msgOrNullWithParams('msgKey', new Object[] {param1, param2, param3, param4})}
32 ${#messages.arrayMsgOrNull(messageKeyArray)}
33 ${#messages.listMsgOrNull(messageKeyList)}
34 ${#messages.setMsgOrNull(messageKeySet)}
```

IDs

```
1 #ids : utility methods for dealing with id attributes that might be repeated (for example, as a
2 /*
3 * =====
4 * See javadoc API for class org.thymeleaf.expression.Ids
5 * =====
6 */
7
8 /*
9 * Normally used in th:id attributes, for appending a counter to the id attribute value
10 * so that it remains unique even when involved in an iteration process.
11 */
12 ${#ids.seq('someId')}
13
14 /*
15 * Normally used in th:for attributes in <label> tags, so that these labels can refer to Ids
16 * generated by means if the #ids.seq(...) function.
17 *
18 * Depending on whether the <label> goes before or after the element with the #ids.seq(...)
19 * function, the "next" (label goes before "seq") or the "prev" function (label goes after
20 * "seq") function should be called.
21 */
22 ${#ids.next('someId')}
23 ${#ids.prev('someId')}
```

顶 8      踩 0

- 上一篇 [SpringBoot-thymeleaf模板集成](#)
- 下一篇 [OLED原理,时序和操作](#)

相关文章推荐

- using thymeleaf之二简单表达式
  - MVVM在美团点评酒旅移动端的最佳实践--王禹华
  - Thymeleaf模板的使用
  - C语言大型软件设计的面向对象--宋宝华
  - 5分钟了解Thymeleaf的标准方言(Standard dialect)
  - Retrofit 从入门封装到源码解析
- 设置属性值-Thymeleaf常见用法（三）
  - Spring Boot 2小时入门基础教程
  - Spring Boot—— Thymeleaf（ gradle ）的简单...
  - Shell脚本编程
  - Spring Boot学习记录(二)--thymeleaf模板
  - Thymeleaf系列六 模板 template



- Thymeleaf教程（四）Thymeleaf标准表达式语...
  - 跳过Java开发的各种坑
- Thymeleaf模板表达式
  - spring-boot--使用thymeleaf模板

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#)   [杂志客服](#)   [微博客服](#)   [webmaster@csdn.net](#)   400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 |

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved 