BraveWangDev
有分享,才有进步

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:

```
<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

```
<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： Today is: 2016年11月2日. 。
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

```
<div th:object="${session.user}">
    <p>Name: <span th:text="*{firstName}">Sebastian</span>.</p>
    <p>Surname: <span th:text="*{lastName}">Pepper</span>.</p>
```

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

等同于以下方式:

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

也可以混用,如下:

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

```
<!— 绝对路径 —>
<!— Will produce 'http://localhost:8080/gtvg/order/details?orderId=3' (plus rewriting) —>
<a href="details.html" th:href="@{http://localhost:8080/gtvg/order/details(orderId=${o.id})}">v

<!— 相对路径 带参数—>
<!— Will produce '/gtvg/order/details?orderId=3' (plus rewriting) —>
<a href="details.html" th:href="@{/order/details(orderId=${o.id})}">view</a>

<!— Will produce '/gtvg/order/3/details' (plus rewriting) —>
<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

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

模板引用:

关闭

```
<table>
    <th th:text="#{header.address.city}">...</th>
    <th th:text="#{header.address.country}">...</th>
</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：+

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

2.文字替换：|The name is ${name}|

```
<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:

```
<span th:text="${#locale.country}">US</span>.

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：

```
#{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+, -, *, /, %

```
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;)

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

布尔运算符 and,or

七,thymeleaf循环

数据集合必须是可以遍历的，使用th:each标签：

```
<body>
  <h1>Product list</h1>

  <table>
    <tr>
      <th>NAME</th>
      <th>PRICE</th>
      <th>IN STOCK</th>
    </tr>
    <tr th:each="prod : ${prods}">
      <td th:text="${prod.name}">Onions</td>
      <td th:text="${prod.price}">2.41</td>
      <td th:text="${prod.inStock}? #{true} : #{false}">yes</td>
    </tr>
  </table>

  <p>
    <a href="../home.html" th:href="@{/}">Return to home</a>
  </p>
</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:

```
<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

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

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

三元运算控制class属性选择

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

三元运算嵌套

```
<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’

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

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

等效于：

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

条件表达式嵌套：

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

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

```
#dates : utility methods for java.util.Date objects:
/*
 * ==========================================================================
 * See javadoc API for class org.thymeleaf.expression.Dates
 * ==========================================================================
 */

/*
 * Null-safe toString()
 */
${#strings.toString(obj)}                           // also array*, list* and set*

/*
 * Format date with the standard locale format
 * Also works with arrays, lists or sets
 */
${#dates.format(date)}
${#dates.arrayFormat(datesArray)}
${#dates.listFormat(datesList)}
${#dates.setFormat(datesSet)}

/*
 * Format date with the specified pattern
 * Also works with arrays, lists or sets
 */
${#dates.format(date, 'dd/MMM/yyyy HH:mm')}
${#dates.arrayFormat(datesArray, 'dd/MMM/yyyy HH:mm')}
${#dates.listFormat(datesList, 'dd/MMM/yyyy HH:mm')}
${#dates.setFormat(datesSet, 'dd/MMM/yyyy HH:mm')}

/*
 * Obtain date properties
 * Also works with arrays, lists or sets
 */
${#dates.day(date)}                     // also arrayDay(...), listDay(...), etc.
${#dates.month(date)}                   // also arrayMonth(...), listMonth(...), etc.
${#dates.monthName(date)}               // also arrayMonthName(...), listMonthName(...), etc.
```

关闭

```
${#dates.monthNameShort(date)}          // also arrayMonthNameShort(...), listMonthNameShort(...
${#dates.year(date)}                    // also arrayYear(...), listYear(...), etc.
${#dates.dayOfWeek(date)}               // also arrayDayOfWeek(...), listDayOfWeek(...), etc.
${#dates.dayOfWeekName(date)}           // also arrayDayOfWeekName(...), listDayOfWeekName(...),
${#dates.dayOfWeekNameShort(date)}      // also arrayDayOfWeekNameShort(...), listDayOfWeekNameS
${#dates.hour(date)}                    // also arrayHour(...), listHour(...), etc.
${#dates.minute(date)}                  // also arrayMinute(...), listMinute(...), etc.
${#dates.second(date)}                  // also arraySecond(...), listSecond(...), etc.
${#dates.millisecond(date)}             // also arrayMillisecond(...), listMillisecond(...), etc

/*
 * Create date (java.util.Date) objects from its components
 */
${#dates.create(year,month,day)}
${#dates.create(year,month,day,hour,minute)}
${#dates.create(year,month,day,hour,minute,second)}
${#dates.create(year,month,day,hour,minute,second,millisecond)}

/*
 * Create a date (java.util.Date) object for the current date and time
 */
${#dates.createNow()}

/*
 * Create a date (java.util.Date) object for the current date (time set to 00:00)
 */
${#dates.createToday()}
```

Calendars

```
#calendars : analogous to #dates, but for java.util.Calendar objects:
/*
 * ============================================================================
 * See javadoc API for class org.thymeleaf.expression.Calendars
 * ============================================================================
 */

/*
 * Format calendar with the standard locale format
 * Also works with arrays, lists or sets
 */
${#calendars.format(cal)}
${#calendars.arrayFormat(calArray)}
${#calendars.listFormat(calList)}
${#calendars.setFormat(calSet)}

/*
 * Format calendar with the specified pattern
 * Also works with arrays, lists or sets
 */
${#calendars.format(cal, 'dd/MMM/yyyy HH:mm')}
${#calendars.arrayFormat(calArray, 'dd/MMM/yyyy HH:mm')}
${#calendars.listFormat(calList, 'dd/MMM/yyyy HH:mm')}
${#calendars.setFormat(calSet, 'dd/MMM/yyyy HH:mm')}

/*
 * Obtain calendar properties
 * Also works with arrays, lists or sets
 */
${#calendars.day(date)}                 // also arrayDay(...), listDay(...), etc.
```

```
${#calendars.month(date)}              // also arrayMonth(...), listMonth(...), etc.
${#calendars.monthName(date)}          // also arrayMonthName(...), listMonthName(...), etc.
${#calendars.monthNameShort(date)}     // also arrayMonthNameShort(...), listMonthNameShort(...
${#calendars.year(date)}               // also arrayYear(...), listYear(...), etc.
${#calendars.dayOfWeek(date)}          // also arrayDayOfWeek(...), listDayOfWeek(...), etc.
${#calendars.dayOfWeekName(date)}      // also arrayDayOfWeekName(...), listDayOfWeekName(...),
${#calendars.dayOfWeekNameShort(date)} // also arrayDayOfWeekNameShort(...), listDayOfWeekNameS
${#calendars.hour(date)}               // also arrayHour(...), listHour(...), etc.
${#calendars.minute(date)}             // also arrayMinute(...), listMinute(...), etc.
${#calendars.second(date)}             // also arraySecond(...), listSecond(...), etc.
${#calendars.millisecond(date)}        // also arrayMillisecond(...), listMillisecond(...), etc

/*
 * Create calendar (java.util.Calendar) objects from its components
 */
${#calendars.create(year,month,day)}
${#calendars.create(year,month,day,hour,minute)}
${#calendars.create(year,month,day,hour,minute,second)}
${#calendars.create(year,month,day,hour,minute,second,millisecond)}

/*
 * Create a calendar (java.util.Calendar) object for the current date and time
 */
${#calendars.createNow()}

/*
 * Create a calendar (java.util.Calendar) object for the current date (time set to 00:00)
 */
${#calendars.createToday()}
```

Numbers

```
#numbers : utility methods for number objects:
/*
 * ====================================================================
 * See javadoc API for class org.thymeleaf.expression.Numbers
 * ====================================================================
 */

/*
 * ========================
 * Formatting integer numbers
 * ========================
 */

/*
 * Set minimum integer digits.
 * Also works with arrays, lists or sets
 */
${#numbers.formatInteger(num,3)}
${#numbers.arrayFormatInteger(numArray,3)}
${#numbers.listFormatInteger(numList,3)}
${#numbers.setFormatInteger(numSet,3)}

/*
 * Set minimum integer digits and thousands separator:
 * 'POINT', 'COMMA', 'NONE' or 'DEFAULT' (by locale).
 * Also works with arrays, lists or sets
 */
${#numbers.formatInteger(num,3,'POINT')}
```

关闭

```
${#numbers.arrayFormatInteger(numArray,3,'POINT')}
${#numbers.listFormatInteger(numList,3,'POINT')}
${#numbers.setFormatInteger(numSet,3,'POINT')}

/*
 * ========================
 * Formatting decimal numbers
 * ========================
 */

/*
 * Set minimum integer digits and (exact) decimal digits.
 * Also works with arrays, lists or sets
 */
${#numbers.formatDecimal(num,3,2)}
${#numbers.arrayFormatDecimal(numArray,3,2)}
${#numbers.listFormatDecimal(numList,3,2)}
${#numbers.setFormatDecimal(numSet,3,2)}

/*
 * Set minimum integer digits and (exact) decimal digits, and also decimal separator.
 * Also works with arrays, lists or sets
 */
${#numbers.formatDecimal(num,3,2,'COMMA')}
${#numbers.arrayFormatDecimal(numArray,3,2,'COMMA')}
${#numbers.listFormatDecimal(numList,3,2,'COMMA')}
${#numbers.setFormatDecimal(numSet,3,2,'COMMA')}

/*
 * Set minimum integer digits and (exact) decimal digits, and also thousands and
 * decimal separator.
 * Also works with arrays, lists or sets
 */
${#numbers.formatDecimal(num,3,'POINT',2,'COMMA')}
${#numbers.arrayFormatDecimal(numArray,3,'POINT',2,'COMMA')}
${#numbers.listFormatDecimal(numList,3,'POINT',2,'COMMA')}
${#numbers.setFormatDecimal(numSet,3,'POINT',2,'COMMA')}
/*
 * ========================
 * Utility methods
 * ========================
 */

/*
 * Create a sequence (array) of integer numbers going
 * from x to y
 */
${#numbers.sequence(from,to)}
${#numbers.sequence(from,to,step)}
```

Strings

```
#strings : utility methods for String objects:
/*
 * =============================================================
 * See javadoc API for class org.thymeleaf.expression.Strings
 * =============================================================
 */

/*
 * Check whether a String is empty (or null). Performs a trim() operation before check
```

```
 * Also works with arrays, lists or sets
 */
${#strings.isEmpty(name)}
${#strings.arrayIsEmpty(nameArr)}
${#strings.listIsEmpty(nameList)}
${#strings.setIsEmpty(nameSet)}

/*
 * Perform an 'isEmpty()' check on a string and return it if false, defaulting to
 * another specified string if true.
 * Also works with arrays, lists or sets
 */
${#strings.defaultString(text,default)}
${#strings.arrayDefaultString(textArr,default)}
${#strings.listDefaultString(textList,default)}
${#strings.setDefaultString(textSet,default)}

/*
 * Check whether a fragment is contained in a String
 * Also works with arrays, lists or sets
 */
${#strings.contains(name,'ez')}                     // also array*, list* and set*
${#strings.containsIgnoreCase(name,'ez')}           // also array*, list* and set*

/*
 * Check whether a String starts or ends with a fragment
 * Also works with arrays, lists or sets
 */
${#strings.startsWith(name,'Don')}                  // also array*, list* and set*
${#strings.endsWith(name,endingFragment)}           // also array*, list* and set*

/*
 * Substring-related operations
 * Also works with arrays, lists or sets
 */
${#strings.indexOf(name,frag)}                      // also array*, list* and set*
${#strings.substring(name,3,5)}                     // also array*, list* and set*
${#strings.substringAfter(name,prefix)}             // also array*, list* and set*
${#strings.substringBefore(name,suffix)}            // also array*, list* and set*
${#strings.replace(name,'las','ler')}               // also array*, list* and set*

/*
 * Append and prepend
 * Also works with arrays, lists or sets
 */
${#strings.prepend(str,prefix)}                     // also array*, list* and set*
${#strings.append(str,suffix)}                      // also array*, list* and set*

/*
 * Change case
 * Also works with arrays, lists or sets
 */
${#strings.toUpperCase(name)}                       // also array*, list* and set*
${#strings.toLowerCase(name)}                       // also array*, list* and set*

/*
 * Split and join
 */
${#strings.arrayJoin(namesArray,',')}
${#strings.listJoin(namesList,',')}
${#strings.setJoin(namesSet,',')}
${#strings.arraySplit(namesStr,',')}                // returns String[]
```

关闭

```
${#strings.listSplit(namesStr,',')}                 // returns List<String>
${#strings.setSplit(namesStr,',')}                  // returns Set<String>

/*
 * Trim
 * Also works with arrays, lists or sets
 */
${#strings.trim(str)}                               // also array*, list* and set*

/*
 * Compute length
 * Also works with arrays, lists or sets
 */
${#strings.length(str)}                             // also array*, list* an

/*
 * Abbreviate text making it have a maximum size of n. If text is bigger, it
 * will be clipped and finished in "..."
 * Also works with arrays, lists or sets
 */
${#strings.abbreviate(str,10)}                      // also array*, list* and set*

/*
 * Convert the first character to upper-case (and vice-versa)
 */
${#strings.capitalize(str)}                         // also array*, list* and set*
${#strings.unCapitalize(str)}                       // also array*, list* and set*

/*
 * Convert the first character of every word to upper-case
 */
${#strings.capitalizeWords(str)}                    // also array*, list* and set*
${#strings.capitalizeWords(str,delimiters)}         // also array*, list* and set*

/*
 * Escape the string
 */
${#strings.escapeXml(str)}                          // also array*, list* and set*
${#strings.escapeJava(str)}                         // also array*, list* and set*
${#strings.escapeJavaScript(str)}                   // also array*, list* and set*
${#strings.unescapeJava(str)}                       // also array*, list* and set*
${#strings.unescapeJavaScript(str)}                 // also array*, list* and set*

/*
 * Null-safe comparison and concatenation
 */
${#strings.equals(str)}
${#strings.equalsIgnoreCase(str)}
${#strings.concat(str)}
${#strings.concatReplaceNulls(str)}

/*
 * Random
 */
${#strings.randomAlphanumeric(count)}
```

Objects

```
#objects : utility methods for objects in general
/*
 * ============================================================
```

```
 * See javadoc API for class org.thymeleaf.expression.Objects
 * ============================================================================
 */

/*
 * Return obj if it is not null, and default otherwise
 * Also works with arrays, lists or sets
 */
${#objects.nullSafe(obj,default)}
${#objects.arrayNullSafe(objArray,default)}
${#objects.listNullSafe(objList,default)}
${#objects.setNullSafe(objSet,default)}
```

Booleans

```
#bools : utility methods for boolean evaluation
/*
 * ============================================================================
 * See javadoc API for class org.thymeleaf.expression.Bools
 * ============================================================================
 */

/*
 * Evaluate a condition in the same way that it would be evaluated in a th:if tag
 * (see conditional evaluation chapter afterwards).
 * Also works with arrays, lists or sets
 */
${#bools.isTrue(obj)}
${#bools.arrayIsTrue(objArray)}
${#bools.listIsTrue(objList)}
${#bools.setIsTrue(objSet)}

/*
 * Evaluate with negation
 * Also works with arrays, lists or sets
 */
${#bools.isFalse(cond)}
${#bools.arrayIsFalse(condArray)}
${#bools.listIsFalse(condList)}
${#bools.setIsFalse(condSet)}

/*
 * Evaluate and apply AND operator
 * Receive an array, a list or a set as parameter
 */
${#bools.arrayAnd(condArray)}
${#bools.listAnd(condList)}
${#bools.setAnd(condSet)}

/*
 * Evaluate and apply OR operator
 * Receive an array, a list or a set as parameter
 */
${#bools.arrayOr(condArray)}
${#bools.listOr(condList)}
${#bools.setOr(condSet)}
```

Arrays

关闭

```
#arrays : utility methods for arrays
/*
 * ============================================================================
 * See javadoc API for class org.thymeleaf.expression.Arrays
 * ============================================================================
 */

/*
 * Converts to array, trying to infer array component class.
 * Note that if resulting array is empty, or if the elements
 * of the target object are not all of the same class,
 * this method will return Object[].
 */
${#arrays.toArray(object)}

/*
 * Convert to arrays of the specified component class.
 */
${#arrays.toStringArray(object)}
${#arrays.toIntegerArray(object)}
${#arrays.toLongArray(object)}
${#arrays.toDoubleArray(object)}
${#arrays.toFloatArray(object)}
${#arrays.toBooleanArray(object)}

/*
 * Compute length
 */
${#arrays.length(array)}

/*
 * Check whether array is empty
 */
${#arrays.isEmpty(array)}

/*
 * Check if element or elements are contained in array
 */
${#arrays.contains(array, element)}
${#arrays.containsAll(array, elements)}
```

Lists

```
#lists : utility methods for lists
/*
 * ============================================================================
 * See javadoc API for class org.thymeleaf.expression.Lists
 * ============================================================================
 */

/*
 * Converts to list
 */
${#lists.toList(object)}

/*
 * Compute size
 */
${#lists.size(list)}

/*
```

```
 * Check whether list is empty
 */
${#lists.isEmpty(list)}

/*
 * Check if element or elements are contained in list
 */
${#lists.contains(list, element)}
${#lists.containsAll(list, elements)}

/*
 * Sort a copy of the given list. The members of the list must implement
 * comparable or you must define a comparator.
 */
${#lists.sort(list)}
${#lists.sort(list, comparator)}
```

Sets

```
#sets : utility methods for sets
/*
 * ============================================================
 * See javadoc API for class org.thymeleaf.expression.Sets
 * ============================================================
 */

/*
 * Converts to set
 */
${#sets.toSet(object)}

/*
 * Compute size
 */
${#sets.size(set)}

/*
 * Check whether set is empty
 */
${#sets.isEmpty(set)}

/*
 * Check if element or elements are contained in set
 */
${#sets.contains(set, element)}
${#sets.containsAll(set, elements)}
```

Maps

```
#maps : utility methods for maps
/*
 * ============================================================
 * See javadoc API for class org.thymeleaf.expression.Maps
 * ============================================================
 */

/*
 * Compute size
 */
${#maps.size(map)}
```

关闭

```
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 /*
```

关闭

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

IDs

```
#ids : utility methods for dealing with id attributes that might be repeated (for example, as a
/*
 * ============================================================================
 * See javadoc API for class org.thymeleaf.expression.Ids
 * ============================================================================
 */

/*
 * Normally used in th:id attributes, for appending a counter to the id attribute value
 * so that it remains unique even when involved in an iteration process.
 */
${#ids.seq('someId')}

/*
 * Normally used in th:for attributes in <label> tags, so that these labels can refer to Ids
 * generated by means if the #ids.seq(...) function.
 *
 * Depending on whether the <label> goes before or after the element with the #ids.seq(...)
 * function, the "next" (label goes before "seq") or the "prev" function (label goes after
 * "seq") function should be called.
 */
${#ids.next('someId')}
${#ids.prev('someId')}
```


- 上一篇 SpringBoot-thymeleaf模板集成
- 下一篇 OLED原理,时序和操作

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

关闭