<http://blog.csdn.net/zrk1000/article/details/72667478>

# 1、创建 html

<!DOCTYPE html><html xmlns:th="http://www.thymeleaf.org"></html>

另外：xmlns:layout="http://www.ultraq.net.nz/web/thymeleaf/layout".

th:\* attributes ：

layout:\* attributes

* 1
* 2
* 3
* 4
* 5
* 6
* 7

注意：html 中的标签必须严格规范，标签必须闭合，即<div />技术或者</div>类似结束

* 1
* 2

# 2、使用文本

| **语法** | **说明** |
| --- | --- |
| {home.welcome} | 使用国际化文本,国际化传参直接追加(value…) |
| ${user.name} | 使用会话属性 |
| @{} | <link rel="stylesheet" type="text/css" media="all"href="../../css/gtvg.css" th:href="@{/css/gtvg.css}" /> |
| – | – |
| ${} 中预存对象（表达式中基本对象） |  |
| param | 获取请求参数，比如${param.name},[http://localhost:8080?name=jeff](http://localhost:8080/?name=jeff) |
| session | 获取 session 的属性 |
| application | 获取 application 的属性 |
| execInfo | 有两个属性 templateName和 now(是 java 的 Calendar 对象) |
| ctx |  |
| vars |  |
| locale |  |
| httpServletRequest |  |
| httpSession |  |
| – | – |
| th扩展标签 |  |
| th:text | 普通字符串 |
| th:utext | 转义文本 |
| th:href |  |
| th:attr | <img src="../../images/gtvglogo.png" th:attr="src=@{/images/gtvglogo.png},title=#{logo},alt=#{logo}" /> |
| th:with | 定义常量 |
| th:attrappend |  |
| th:classappend |  |
| th:styleappend |  |

其他th标签

| **\*** | **\*** | **\*** |
| --- | --- | --- |
| th:abbr | th:accept | th:accept-charset |
| th:abbr | th:accept | th:accept-charset |
| th:accesskey | th:action | th:align |
| th:alt | th:archive | th:audio |
| th:autocomplete | th:axis | th:background |
| th:bgcolor | th:border | th:cellpadding |
| th:cellspacing | th:challenge | th:charset |
| th:cite | th:class | th:classid |
| th:codebase | th:codetype | th:cols |
| th:colspan | th:compact | th:content |
| th:contenteditable | th:contextmenu | th:data |
| th:datetime | th:dir | th:draggable |
| th:dropzone | th:enctype | th:for |
| th:form | th:formaction | th:formenctype |
| th:formmethod | th:formtarget | th:frame |
| th:frameborder | th:headers | th:height |
| th:high | th:href | th:hreflang |
| th:hspace | th:http-equiv | th:icon |
| th:id | th:keytype | th:kind |
| th:label | th:lang | th:list |
| th:longdesc | th:low | th:manifest |
| th:marginheight | th:marginwidth | th:max |
| th:maxlength | th:media | th:method |
| th:min | th:name | th:optimum |
| th:pattern | th:placeholder | th:poster |
| th:preload | th:radiogroup | th:rel |
| th:rev | th:rows | th:rowspan |
| th:rules | th:sandbox | th:scheme |
| th:scope | th:scrolling | th:size |
| th:sizes | th:span | th:spellcheck |
| th:src | th:srclang | th:standby |
| th:start | th:step | th:style |
| th:summary | th:tabindex | th:target |
| th:title | th:type | th:usemap |
| th:value | th:valuetype | th:vspace |
| th:width | th:wrap | th:xmlbase |
| th:xmllang | th:xmlspace | th:alt-title 或th:lang-xmllang（如果其中两个属性值相同） |

对于 html5 元素名称的另一种友好写法

<table>

<tr data-th-each="user : ${users}">

<td data-th-text="${user.login}">...</td>

<td data-th-text="${user.name}">...</td> </tr>

</table>

* 1
* 2
* 3
* 4
* 5

# 3、表达式语法

## 1、简单表达式语法

* #{...} : Message 表达式

<p th:utext="#{home.welcome(${session.user.name})}"> Welcome to our grocery store, Sebastian Pepper!</p>

<p th:utext="#{${welcomeMsgKey}(${session.user.name})}"> Welcome to our grocery store, Sebastian Pepper!</p>

* 1
* 2
* ${}:变量表达式

ongl标准语法，方法也可以被调用

* 1
* \*{} ：选择变量表达式

<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>

等价于

<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>

当然了，这两者可以混合使用

还有一种方式

<div>

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

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

<p>Nationality: <span th:text="\*{session.user.nationality}">Saturn</span>.</p>

</div>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* @{}: 链接 URL 表达式

<!-- 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})}">view</a> <!-- 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>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

## 2、变量

| **分类** | **示例** |
| --- | --- |
| 文本 | ‘one text’ , ‘Another one!’ ,… |
| 数字 | 0 , 34 , 3.0 , 12.3 ,… |
| 真假 | true , false |
| 文字符号 | one , sometext , main ,… |

## 3、字符连接

| **分类** | **示例** |
| --- | --- |
| + | ‘The name is ‘+${name} |
| |…| | |The name is ${name}| |

## 4、 算数运算

| **语法** | **示例** |
| --- | --- |
| +, -, \*, /, % | 二元运算符 |
| - | 减号（一元运算符） |

## 5、 真假运算

| **分类** | **示例** |
| --- | --- |
| and , or | 二元运算符 |
| ! , not | 否定（一元运算符） |

## 6、比较运算

| **分类** | **示例** |
| --- | --- |
| >, <, >=, <= (gt, lt, ge, le) | 比较 |
| == , != ( eq , ne ) | 平等 |

## 7、 条件运算

| **分类** | **示例** |
| --- | --- |
| if-then | (if) ? (then) |
| if-then-else | (if) ? (then) : (else) |
| Default | (value) ?: (defaultvalue) |

综合示例：

'User is of type ' + (${user.isAdmin()} ? 'Administrator' : (${user.type} ?: 'Unknown'))

* 1

# 4、表达式中使用内置对象

#dates :

utility methods for java.util.Date objects: formatting, component extraction, etc. #calendars : analogous to #dates , but for java.util.Calendar objects.

#numbers :

utility methods for formatting numeric objects.

#strings :

utility methods for String objects: contains, startsWith, prepending/appending, etc. #objects : utility methods for objects in general.

#bools :

utility methods for boolean evaluation. #arrays : utility methods for arrays.

#lists :

utility methods for lists.

#sets :

utility methods for sets.

#maps :

utility methods for maps.

#aggregates :

utility methods for creating aggregates on arrays or collections.

#messages :

utility methods for obtaining externalized messages inside variables expressions, in the same way as they would be obtained using #{...} syntax.

#ids :

utility methods for dealing with id attributes that might be repeated (for example, as a result of an iteration).

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30

# 5、预处理

\_\_${expression}\_\_

* 1

# 6、循环

<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>

迭代器的状态

index: 当前的索引，从0开始

count: 当前的索引，从1开始

size：总数

current:

even/odd:

first

last

<table>

<tr>

<th>NAME</th>

<th>PRICE</th>

<th>IN STOCK</th>

</tr>

<tr th:each="prod,iterStat : ${prods}" th:class="${iterStat.odd}? 'odd'">

<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>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26

# 7、判断

if

<a href="comments.html" th:href="@{/product/comments(prodId=${prod.id})}" th:if="${not #lists.isEmpty(prod.comments)}">view</a>

unless

<a href="comments.html" th:href="@{/comments(prodId=${prod.id})}" th:unless="${#lists.isEmpty(prod.comments)}">view</a>

switch

<div th:switch="${user.role}">

<p th:case="'admin'">User is an administrator</p> <p th:case="#{roles.manager}">User is a manager</p>

</div>

<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>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18

# 8、模板布局

th:fragment

示例

templates/footer.html

<!DOCTYPE html SYSTEM "http://www.thymeleaf.org/dtd/xhtml1-strict-thymeleaf-4.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"

<body>

<div th:fragment="copy">

© 2011 The Good Thymes Virtual Grocery

</div>

</body>

</html>

templates/index.html中使用

<body> ...

<div th:include="footer :: copy"></div>

</body>

或者

...

<div id="copy-section">

© 2011 The Good Thymes Virtual Grocery

</div>

...

使用

<body> ...

<div th:include="footer :: #copy-section"></div>

</body>

th:include 和 th:replace 区别

th:include 加入代码

th:replace 替换代码

模板传参：参数传递顺序不强制

定义

<div th:fragment="frag (onevar,twovar)">

<p th:text="${onevar} + ' - ' + ${twovar}">...</p>

</div>

使用

<div th:include="::frag (${value1},${value2})">...</div>

<div th:include="::frag (onevar=${value1},twovar=${value2})">...</div>

等价于 <div th:include="::frag" th:with="onevar=${value1},twovar=${value2}">）

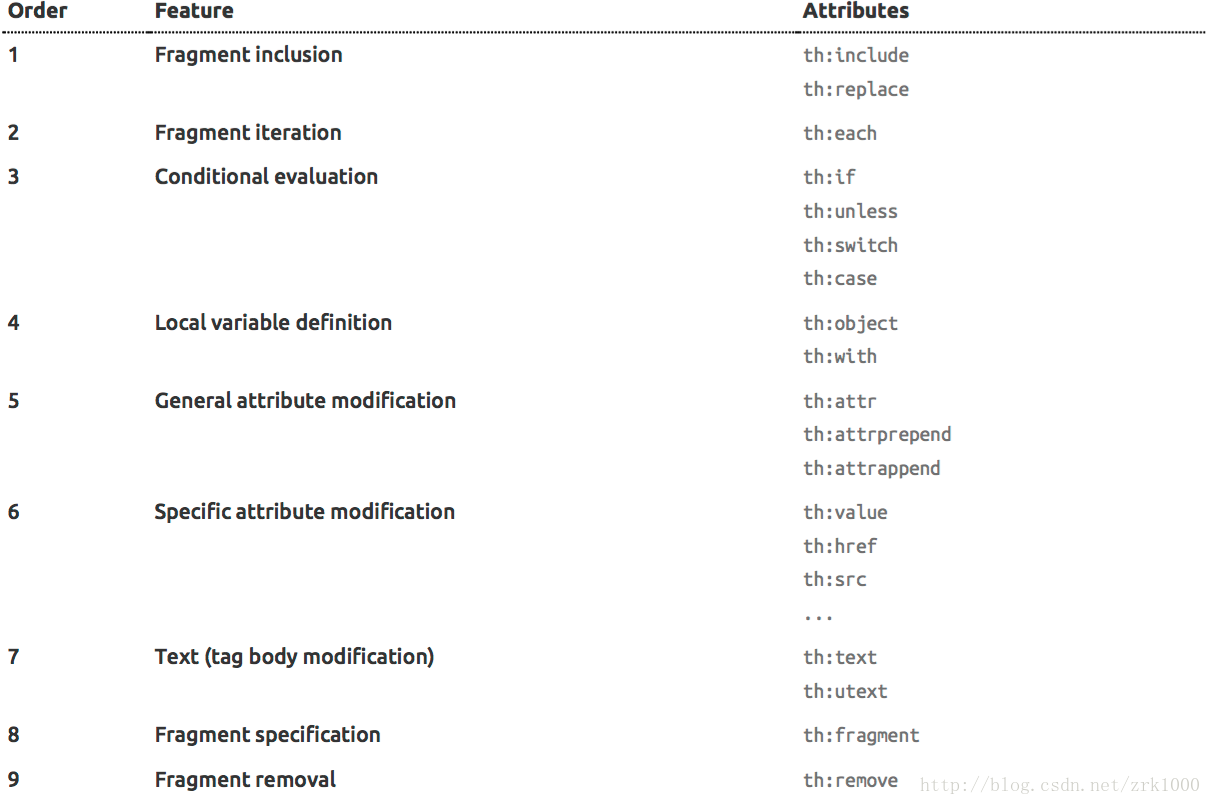
* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55

# 9、移除标签 th:remove

取值范围

* all：移除所有
* body：不移除自己，但移除他的子标签
* tag: 只移除自己，不移除他的子标签
* all-but-first：移除所有内容除第一个外
* none：啥都不做

# 10、执行顺序



# 11、thymeleaf注释语法

* html 看不到，并且 thymeleaf 不会执行

<!--/\* This code will be removed at thymeleaf parsing time! \*/-->

* 1
* and 未运行可以在 html 中看到，运行后就消失

<!--/\*-->

<div>you can see me only before thymeleaf processes me! </div>

<!--\*/-->

* 1
* 2
* 3
* 运行后才会看到

<span>hello!</span>

<!--/\*/

<div th:text="${true}">...</div>

/\*/-->

<span>goodbye!</span>

* 1
* 2
* 3
* 4
* 5

# 12、th:block 的使用

<table>

<th:block th:each="user : ${users}">

<tr>

<td th:text="${user.login}">...</td> <td th:text="${user.name}">...</td>

</tr>

<tr>

<td colspan="2" th:text="${user.address}">...</td>

</tr>

</th:block>

</table>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10

推荐下面写法（编译前看不见）

<table>

<tr>

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

<td th:text="${user.name}">...</td> </tr>

<tr>

<td colspan="2" th:text="${user.address}">...</td>

</tr>

<!--/\*/ </th:block> /\*/-->

</table>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9

# 13、文本内联th:inline

th:inline 可以等于 text , javascript(dart) , none

* text: [[…]]

<p th:inline="text">Hello, [[#{test}]]</p>

* 1
* javascript: /[[…]]/

<script th:inline="javascript">

var username = /\*[[

#{test}

]]\*/;

var name = /\*[[

${param.name[0]}+${execInfo.templateName}+'-'+${#dates.createNow()}+'-'+${#locale}

]]\*/;

</script>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8

<script th:inline="javascript">

/\*<![CDATA[\*/

var username = [[#{test}]];

var name = [[${param.name[0]}+${execInfo.templateName}+'-'+${#dates.createNow()}+'-'+${#locale}]];

/\*]]>\*/

</script>

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* adding code: /\* [+…+]\*/

var x = 23;

/\*[+

var msg = 'Hello, ' + [[${session.user.name}]]; +]\*/

var f = function() {

...

* 1
* 2
* 3
* 4
* 5
* removind code: /[- / and /\* -]\*/

var x = 23;

/\*[- \*/

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

var f = function() {

...

* 1
* 2
* 3
* 4
* 5

# 14、验证模板的正确性

<!DOCTYPE html SYSTEM "http://www.thymeleaf.org/dtd/xhtml1-strict-thymeleaf-4.dtd">

<!DOCTYPE html SYSTEM "http://www.thymeleaf.org/dtd/xhtml1-transitional-thymeleaf-4.dtd">

<!DOCTYPE html SYSTEM "http://www.thymeleaf.org/dtd/xhtml1-frameset-thymeleaf-4.dtd">

<!DOCTYPE html SYSTEM "http://www.thymeleaf.org/dtd/xhtml11-thymeleaf-4.dtd">

* 1
* 2
* 3
* 4

# 15、特殊用法展示

<td th:text="${#aggregates.sum(o.orderLines.{purchasePrice \* amount})}">23.32</td>

* 1

以上表示List orderLines的所有订单的总价

# 附件A: 基础对象

ctx：对应org.thymeleaf.spring[3|4].context.SpringWebContext

/\*

\* ======================================================================

\* See javadoc API for class org.thymeleaf.context.IContext

\* ====================================================================== \*/

${#ctx.locale} ${#ctx.variables}

/\*

\* ======================================================================

\* See javadoc API for class org.thymeleaf.context.IWebContext

\* ====================================================================== \*/

${#ctx.applicationAttributes}

${#ctx.httpServletRequest}

${#ctx.httpServletResponse}

${#ctx.httpSession}

${#ctx.requestAttributes}

${#ctx.requestParameters}

${#ctx.servletContext}

${#ctx.sessionAttributes}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17

locale: 对应java.util.Locale

vars: 对应 org.thymeleaf.context.VariablesMap

/\*

\* ====================================================================== \* See javadoc API for class org.thymeleaf.context.VariablesMap

\* ====================================================================== \*/

${#vars.get('foo')}

${#vars.containsKey('foo')}

${#vars.size()}

...

* 1
* 2
* 3
* 4
* 5
* 6
* 7

param

${param.foo} 是一个 String[] 如果要获取需要${param.foo[0]}

/\*

\* ============================================================================ \* See javadoc API for class org.thymeleaf.context.WebRequestParamsVariablesMap \* ============================================================================ \*/

${param.foo} // Retrieves a String[] with the values of request parameter 'foo'

${param.size()}

${param.isEmpty()}

${param.containsKey('foo')}

...

* 1
* 2
* 3
* 4
* 5
* 6
* 7

session

application

httpServletRequest

themes : as JSP tag spring:theme

Spring Beans 的访问

<div th:text="${@authService.getUserName()}">...</div>

* 1

# 附件B:工具对象

* dates See javadoc API for class org.thymeleaf.expression.Dates
* calendars See javadoc API for class org.thymeleaf.expression.Calendars
* numbers See javadoc API for class org.thymeleaf.expression.Numbers
* strings See javadoc API for class org.thymeleaf.expression.Strings
* objects See javadoc API for class org.thymeleaf.expression.Objects
* bools See javadoc API for class org.thymeleaf.expression.Bools
* arrays See javadoc API for class org.thymeleaf.expression.Arrays
* lists See javadoc API for class org.thymeleaf.expression.Lists
* sets See javadoc API for class org.thymeleaf.expression.Sets
* maps See javadoc API for class org.thymeleaf.expression.Maps
* aggregates See javadoc API for class org.thymeleaf.expression.Aggregates
* messages See javadoc API for class org.thymeleaf.expression.Messages
* ids See javadoc API for class org.thymeleaf.expression.Ids

# 附件C:DOM 选择器语法

DOM选择器借语法功能从XPath，CSS和jQuery，为了提供一个功能强大且易于使用的方法来指定模板片段

<div th:include="mytemplate :: [//div[@class='content']]">...</div>

* 1

<div th:include="mytemplate :: [div.content]">...</div>

* 1

版权声明：本文为博主原创文章，未经博主允许不得转载