**thymeleaf的简单用法-常用标签**

**一.常用标签的用法**

**1.1 th:text 用于文本的显示,并且可以进行简单的计算。**

<td  th:text="${username}">mlm</td> -> <td>maliming</td>

<td  th:text="9+7">2018</td> -> <td>16</td>

**1.2 th:utext 用于html的文本替换,常用于富文本编辑器编辑后的内容显示到前台页面上。**

<p th:utext="${article.content}"></p> -> <p>my name is <b>maliming</b></p>

**1.3 th:if 用于判断条件,还可以多条件 and,or(二元操作符)，!,not非(一元操作符)。**

<div th:if="${user} != null">show</div>如果user对象为空则不显示，不为空则显示 -> <div>show</div>

<div th:if="${user} != null and ${otherUser} != null">show</div>

运算符还可以写在${}里<div th:if="${user.number > 100}">show</div> -> <div>show</div> 运算符放在了 {} 内部, 此时整个 {} 内的表达式用 ognl/spel 引擎计算; 如果运算符放在外部, 那么 thymeleaf 引擎负责计算。

**1.4 th:unless 用于判断条件，与th:if作用相反。**

<div th:if="${user} != null">show</div>如果user对象不为空则不显示，为空则显示 -> <div>show</div>

**1.5 th:switch th:case 用于多个同等级相同目的判断,多选一时使用。**

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

    <p th:case="maliming">first</p>若${user.name}中的值为maliming则显示，否则不显示

    <p th:case="${otherUser.name}">second</p>

</div>

**1.6 th:action 用于定义后台控制器的路径，表单的提交地址，相当于<form>标签的action属性。**

<form th:action="@{user/login}" method="post"></form>

**1.7 th:each 用于遍历集合中的对象，相当于jstl中的<c:forEach>标签。**

List集合循环遍历

<tr th:each="user,userStat:${messages.list}">

    <td th:text="${user.name}"></td>

    <td th:text="userStat.index"></td>

</tr>

其中的user是临时变量，相当于for(User user : users)中的user，userStat称为状态变量，属性有index:当前迭代对象的index(从0开始计算),count: 当前迭代对象的index(从1开始计算), size:被迭代对象的大小,current:当前迭代变量,even/odd:布尔值，当前循环是否是偶数/奇数(从0开始计算), first:布尔值，当前循环是否是第一个,last:布尔值，当前循环是否是最后一个。

Map集合循环遍历

<div th:each="map:${maps}" th:text="${map}"></div>

数组循环遍历

<tr>

    <td th:each="array:${arrays}" th:text="${array}"></td>

</tr>

**1.8 th:value 用于属性赋值。**

<option th:value="${user.name}"></option> -> <option value="maliming"></option>

**1.9 th:src 用于外部资源的引入，例如图片，js文件。**

<img th:src="@{../images/myself.jpg}"/>或<script th:src="@{../static/login.js}"></script>

**1.10 th:href 用于定义超链接,相当于<a></a>标签的href属性。**

<a th:href="@{/user/selectUser?(currentPage=1,reTypeid=${reTypeid},inquiry=${inquiry})}"></a> 传参。

**1.11 th:remove 用于删除。可以表达式传参。**

<table border="1">    
    <thead>    
          <tr>    
             <th>编号</th>    
             <th>用户名</th>      
             <th>姓名</th>     
             <th>性别</th>    
          </tr>    
    </thead>    
    <tbody th:remove="all-but-first" 或者表达式th:remove="${choice}? all : all-but-first">    
          <tr>    
             <td>1</td>    
             <td>xxxxxxxx@qq.com</td>    
             <td>Tom</td>    
             <td>男</td>    
          </tr>    
          <tr>    
             <td>3</td>    
             <td>xxxxxxxx@qq.com</td>    
             <td>Lucy13</td>    
             <td>女</td>    
           </tr>  

     </tbody>

</table>

这里的<tbody></tbody>标签中有th:remove="all-but-first"，意思是只保留<tbody></tbody>标签中的第一个字标签，也就是name为Tom的所在<tr></tr>,其他的子标签则会删除，th:remove有五个属性值。

all：删除所在标签和内容及其所属的所有子标签。body：删除所在标签的内容及其所属的所有子标签。tag：删除所在标签，不删除任何所属的子标签。all-but-first：删除除第一个子标签外的其他子标签。none：不起作用，什么也不做。

**1.12 th:selected 用于选择框设置选中值。通常和th:each一起使用。**

<select>

    <option th:selected="${user.name} == ${otherUser.name}"></option> 若相等就默认选中此<option></option>

</select>

**1.13 th:object 用于表单数据对象绑定，后台controller中参数保持一致，和选择(星号)表达式。**

<form th:object="${user}">

    <input th:value="\*{name}"/> \*号代替了${user}

</form>

public ModelAndView addUser(@RequestParam(value = "user") User user，ModelMap model){}

**1.14 th:attr 用于设置任意属性**

<input th:attr="value=${user.name}"/> 设置单个属性。

<input th:attr="value=${user.username},name=username"/> 设置多个属性之间用逗号隔开。

**二.Thymeleaf中的内置对象**

#dates：日期格式化内置对象，具体方法可以参照java.util.Date；  
#calendars：类似于#dates，但是是java.util.Calendar类的方法；  
#numbers： 数字格式化；  
#strings：字符串格式化，具体方法可以参照java.lang.String，如startsWith、contains等；  
#objects：参照java.lang.Object；  
#bools：判断boolean类型的工具；  
#arrays：数组操作的工具；  
#lists：列表操作的工具，参照java.util.List；  
#sets：Set操作工具，参照java.util.Set；  
#maps：Map操作工具，参照java.util.Map；  
#aggregates：操作数组或集合的工具；

#messages：操作消息的工具。