# 平台改造方案（msBase）

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 版本 | 修改人 | 修改时间 |
| 1 | V0.1 | 谢永文 | 2016-11-24 |
| 2 |  |  |  |
| 3 |  |  |  |
| … |  |  |  |

* 项目背景

公司现有平台基于grails+Dojo架构，存在页面效果差、浏览器兼容性不好、Dojo开发资源少、前后台交互的二次开发量大等问题。为了解决以上问题，用grails+bootstrap+jQuery架构对平台进行改造。

* 平台改造内容

1. 数据列表交互

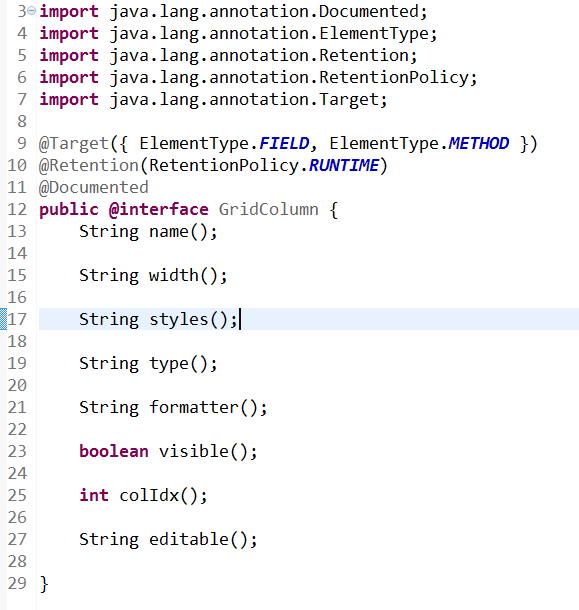
为了达到平台快速开发效果，把数据列表交互通过Taglib技术，前端数据列表使用jQuery的datatable技术。封装好后，以后直接调用数据列表表头，如：<ms:tableHead class=”domain类” id=”tableID” titles=”序号；姓名；性别；年龄…” >

数据列表数据，如：<ms:tableData class=”domain类” id=”tableID” type=”数据列表类型” colData=”seq；name；sex；age…” where=” flag=’0’ and sex=’1’ ” sort=”name desc age asc ” >

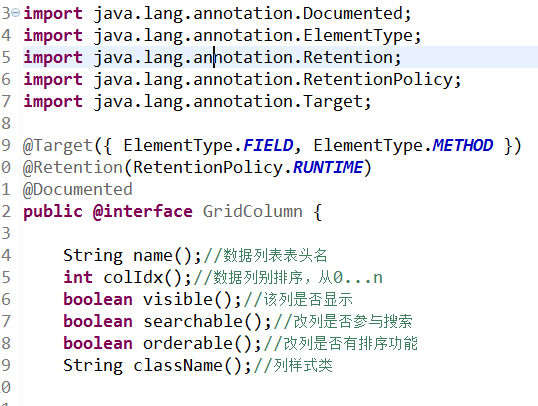
* 1. 标签属性表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 数据列表表头：tableHead | | | | | |
| 序号 | 属性 | 类型 | 是否必须 | 默认值 | 描述 |
| 1 | class | String | Y | 无 | Domain类，包括包名 |
| 2 | titles | String | N | Domain类加了@GridColum(name="姓名")的属性或方法 | Titles值优先domain类加了@GridColumn(name="姓名")的属性或方法；string以分号解析 |
| … |  |  |  |  |  |
| 数据列表数据tableData | | | | | |
| 序号 | 属性 | 类型 | 是否必须 | 默认值 | 描述 |
| 1 | class | String | Y | 无 | Domain类，包括包名 |
| 2 | type | String | Y |  | 数据列表类型，根据jQuery的datatable和业务定义类型 |
| 3 | colData | String | N | Domain类加了@GridColum的属性或方法 | colData值优先domain类加了@GridColumn的属性或方法；string以分号解析 |
| … |  |  |  |  |  |
| … |  |  |  |  |  |

* 1. GridColumn改造



原类



改造类，根据以下表做修改

<http://blog.csdn.net/u013380777/article/details/52067483>

|  |  |  |  |
| --- | --- | --- | --- |
| 用户参数名 | 源码参数名 | 英文解释 | 中文解释 |
| cellType | sCellType | Cell type to be created for a column | 设置列标签的类型（ex：th，td） |
| className | sClass | Class to assign to each cell in the column | 设置列的class属性值 |
| contentPadding | sContentPadding | Add padding to the text content used when calculating the optimal with for a table. | 设置填充内容，以计算与优化为一个表时所使用的文本内容，一般不需要设置 |
| createdCell | fnCreatedCell | Cell created callback to allow DOM manipulation | 设置cell创建完后的回调函数，设置背景色或者添加行 |
| data | mData | Set the data source for the column from the rows data object / array | 设置单元格里的值 |
| defaultContent | sDefaultContent | Set default, static, content for a column | 设置列的默认值 |
| name | sName | Set a descriptive name for a column | 设置列的描述性名称 |
| orderable | bSortable | Enable or disable ordering on this column | 设置列是否可以排序 |
| orderData | aDataSort | Define multiple column ordering as the default order for a column | 设置多列排序时列的默认顺序 |
| orderDataType | sSortDataType | Live DOM sorting type assignment |  |
| orderSequence | asSorting | Order direction application sequence | 设置列的默认排序，可以改变列排序的顺序处理 |
| render | mRender | Render (process) the data for use in the table |  |
| searchable | bSearchable | Enable or disable filtering on the data in this column | 设置列的数据是否过滤 |
| title | sTitle | Set the column title | 设置列的标题 |
| type | sType | Set the column type - used for filtering and sorting string processing.  Four types (string, numeric, date and html (which will strip HTML tags before ordering)) are currently available. | 设置列的类型，用于过滤和排序的字符串处理。 |
| visible | bVisible | Enable or disable the display of this column | 设置列是否显示 |
| width | sWidth | Column width assignment | 定义列的宽度 |

* 1. MsTaglib类
     1. 生成数据表头
     2. 生成数据列表数据

1. 前端布局
2. 前端控件
   1. msTree数据

3.1.1 msTree相关依赖文件

使用该插件请先引入jquery、bootstarp(js和css)、ztree(js和css)文件，同时根据需要选择引入不同tree插件，该封装插件考虑网络、等待时间相关因素，点击父节点加载子节点形式。Demo请下载cylw项目后启动，访问地址为/cylw/tree/testTree：具体请看以下介绍：

1. 如需使用下拉选形式树，请引用msInputTree.js文件。
2. 如需使用勾选形式树，请引用msCheckboxTree.js文件。
3. 如需进行增删改操作的树，请引用msOperateTree.js文件。
4. 注意：在引用msTree插件css时请将tree文件夹放入到web-app/css文件夹下，或将tree下的metroStyle.css文件和img文件夹统一放到web-app/css文件夹下

3.1.2 msTree下拉选(msInputTree.js)属性及方法

1. 属性：

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 属性值 | 属性值类型 | 应用对象 |
| data-ms-tree | msTree实例容器名称 | String | input |

1. 举例说明：

<input id="text1" type="text" data-ms-tree="treeDemo"/>，在此段html代码中，data-ms-tree属性后的容器名称必须是页面不存在的容器名称，如果存在则无法实例化msTree插件。

1. 方法：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 方法解释 |
| msTreeInput(holder,url) | holder：实例容器名称对应data-ms-tree中的容器名称、url：获取数据后台url | 无 | 调用该方法初始化下拉选树，需传入容器名称及获取数据url请求地址。双击msTree节点（地区名、机构名、部门名）将节点名称写入带有data-ms-tree属性input框。 |
| getInputCode(treeId) | treeId：实例容器名称 | String | 使用该方法能够获取到data-ms-tree属性input框中节点（地区名、机构名、部门名）对应的代码。 |

1. 举例：

树测试文本框1：<input id="text1" type="text" data-ms-tree="treeDemo"/>

<button type="button" id="get1">获取文本框1中名称对应的代码</button>

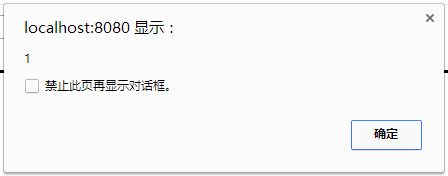
$("#get1").click(function(){

var value = getInputCode("treeDemo");

alert(value);

});





3.1.3 msTree勾选(msCheckboxTree.js)方法

1. 方法：

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 方法解释 |
| msTreeCheckbox(holder,url) | holder：实例容器名称，url：获取数据后台url | 无 | 初始化带有复选框的msTree实例。需要在页面提供容器。格式为div+ul格式 |
| getMsTreeCheck(treeId) | treeId：实例容器名称 | Json字符串 | 获取页面中已展开节点并勾选的节点。返回勾选节点的id、name、pId、pName属性。 |
| getMsTreeNoCheck(treeId) | treeId：实例容器名称 | Json字符串 | 获取页面中展开节点未勾选节点，返回勾选节点的id、name、pId、pName属性。 |

1. 详情请看/cylw/tree/testTree demo演示案例

3.1.4 msTree操作(msOperateTree.js)属性及方法

1. 属性

|  |  |  |
| --- | --- | --- |
| 属性名称 | 应用对象 | 属性解释 |
| data-ms-tree-operate-id | Input | 内嵌页面input应用此属性后，编辑时将所点击节点的id写入到该input，新增时将所点击节点id写入到内嵌页面input属性为data-ms-tree-operate-pId的元素中。 |
| data-ms-tree-operate-name | Input | 内嵌页面input应用此属性后，编辑时将所点击节点的name写入到该input，新增时将所点击节点name写入到内嵌页面input属性为data-ms-tree-operate-pName的元素中。 |
| data-ms-tree-operate-pid | Input | 内嵌页面input应用此属性后，编辑时将所点击节点的pId写入到该input，新增时将所点击节点id写入到内嵌页面input属性为data-ms-tree-operate-pId的元素中。 |
| data-ms-tree-operate-pname | Input | 内嵌页面input应用此属性后，编辑时将所点击节点的pName写入到该input，新增时将所点击节点name写入到内嵌页面input属性为data-ms-tree-operate-pName的元素中。 |
| newPageOperateHolder | Div | 内嵌页面容器divid名称 |
| editMsTreeNodeUrl |  | 此属性为后台javaBean必须提供属性，该属性用于保存点击编辑按钮时，向所点击节点获取后台返回数据地址 |
| delMsTreeNodeUrl |  | 此属性为后台javaBean必须提供属性，该属性用于保存点击删除按钮时，向所点击节点获取是否删除成功后台地址。 |
| loadMsTreeNodeUrl |  | 此属性为后台javaBean必须提供属性，该属性用于保存内嵌页面地址。 |

1. 方法:

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名 | 参数 | 返回值 | 方法解释 |
| msTreeOperate(holder,url) | holder：实例容器名称，url：获取数据后台url | 无 | 初始化操作的msTree实例。需要在页面提供容器。格式为div+ul格式 |
| msTreeOperateUpdateNode(treeId,isSilent) | treeId：需要更新的实例容器名称，isSilent：更新后是否展开节点 | 无 | 该方法如果只传入treeid则更新根节点，如传入treeId、true表示更新容器中选中节点把展开子节点，反之则展开子节点。 |

3.1.5 msTree后台工具类

1. 该工具类名称：MsTree.java

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 类型 | 是否必须 | 属性值 |
| id | string | Y | 节点id |
| name | string | Y | 节点名称 |
| pId | string | Y | 节点pid |
| parent | int | Y | 是否为父节点，0：父，1：子 |
| chked | int | 初始化复选框树时必须 | 是否勾选，0：勾选，1：不勾选 |
| loadMsTreeNodeUrl | string | 初始化为操作树时必须 | 存放内嵌页面到指定容器url地址，相对路径。"../abc.html" |
| editMsTreeNodeUrl | string | 初始化为操作树时必须 | 点击编辑时用于向后台获取数据的action地址，填写相对地址。"../boatArea/testEditaArea" |
| delMsTreeNodeUrl | string | 初始化为操作树时必须 | 点击删除时用于删除节点的后台action地址，请使用相对地址。"../boatArea/testDelArea" |
|  |  |  |  |
|  |  |  |  |

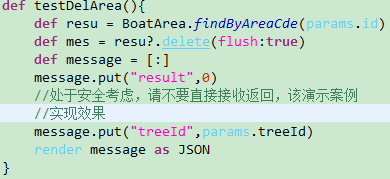
1. 初始化树举例(后台编码)：

初始化操作树，首先引入相关依赖文件后引入msOperateTree.js文件，前面有所介绍这里不一一介绍了。

初始化时，页面会向后台传入两个参数id、name。后台只需判断是否有参数传入，根据传入情况做相关操作，最后将从数据库中获得的数据集合用工具类转换为定义的格式即可。如下图所示：



1. editMsTreeNodeUrl、delMsTreeNodeUrl地址对应方法中，页面会向该方法中传入nodeId、treeId，返回数据时请将返回treeId返回，该插件中前端成功回调判断成功标志"result",0。如下图所示；



## 3.2 input数据校验

### 3.2.1 相关依赖文件

1. 引入jquery、bootstarp后，需要引入checks.js。
2. 属性介绍

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名称 | 属性值类型 | 应用对象 | 解释 |
| data-ms-validate | 无 | input | 该属性用于为元素绑定获得焦点失去失去焦点事件 |
| data-ms-validate-phone | 无 | input | 验证手机号 |
| data-ms-validate-idcard | 无 | input | 验证身份证号 |
| data-ms-validate-email | 无 | input | 验证邮箱 |
| data-ms-validate-max | String必须 | input | 验证最大长度 |
| data-ms-validate-min | String必须 | input | 验证最小长度 |
| data-ms-validate-leftblank | 无 | input | 去除左空格 |
| data-ms-validate-rightblank | 无 | input | 去除右空格 |
| data-ms-validate-allblank | 无 | input | 去除所有空格 |
| data-ms-validate-isnotnull | 无 | input | 非空验证 |
| data-ms-validate-integral | 无 | input | 整数验证 |
| data-ms-validate-small | 无 | input | 小数验证 |

1. 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法名称 | 参数 | 参数类型 | 解释 |
| validate（） | 无 | 无 | 为选择区域内的元素校验 |
| formSubmitCheck(formElem) | 有 | html元素 | 就该方法用于校验选定区域是否可以提交 |

3.2.1 相关依赖文件

3.2.1 相关依赖文件

3.2.1 相关依赖文件

1. 开发文档说明