内置组件的应用

# 常用组件

## 布局组件

**布局文件配置：**

/ numberone /src/main/webapp/WEB-INF/decorators.xml



**默认布局文件：**

/numberone/src/main/webapp/WEB-INF/views/layouts/default.jsp

**非公共，自己建立的布局文件：**

/ numberone /src/main/webapp/WEB-INF/views/模块路径/layouts/布局文件.jsp

**使用布局文件：**

JSP的head里添加：<meta name="decorator" content="default"/>

## 用户工具 UserUtils.java fns.tld

应用场景：在java文件或jsp页面上，获取当前用户相关信息

1. 获取当前用户：
   1. UserUtils.getUser();
   2. entity.currentUser()
   3. ${fns:getUser()}
2. 获取当前用户部门：
   1. UserUtils.getOfficeList()
   2. ${fns:getOfficeList()}
3. 获取当前用户区域：
   1. UserUtils.getAreaList()
   2. ${fns:getAreaList()}
4. 获取当前用户菜单：
   1. UserUtils.getMenuList()
   2. ${fns:getMenuList()}
5. 获取当前用户缓存：
   1. UserUtils.getCache(key);
   2. ${fns:getCache(cacheName, defaultValue)}
6. 设置当前用户缓存：
   1. UserUtils.putCache(key);

## 全局缓存 CacheUtils.java

应用场景：系统字典

1. 设置应用程序缓存：CacheUtils.put(key);
2. 获取应用程序缓存：CacheUtils.get(key);

## 字典工具 DictUtils.java

应用场景：系统全局固定的字典数据，java或jsp中获取字典相关数据。

1. 根据类型和值获取字典标签（列表取值）：
   1. DictUtils.getDictLabel(String value, String type, String defaultValue)
   2. ${ fns:getDictLabel (value, type, defaultValue)}
2. 根据类型和标签获取字典值（根据标签取值）：
   1. DictUtils.getDictValue(String label, String type, String defaultLabel)
   2. ${fns:getDictValue(label, type, defaultValue)}
3. 根据类型获取字典列表（下拉框，复选框，单选框）：
   1. DictUtils.getDictList(String type)
   2. ${fns:getDictList(type)}

## 功能权限控制

应用场景：访问功能授权，查看权限，编辑权限，导入权限，审核权限。

1. 给方法添加权限标志

@RequiresPermissions("sys:submodule:user:view")

@RequiresUser

1. 菜单中设置权限标志。
2. 判断权限：

SecurityUtils.getSubject().isPermitted("sys:user:edit")；

1. 视图中控制按钮（shiro.tld）：

<shiro:hasPermission name="sys:user:edit">

<input id="btnSubmit" class="btn btn-primary" type="submit" value="保 存"/>&nbsp;

</shiro:hasPermission>

<!-- 任何一个符合条件的权限 -->

<shiro:hasAnyPermissions name="sys:user:view, sys:user:edit,">

<input id="btnSubmit" class="btn btn-primary" type="submit" value="返 回"/>&nbsp;

</shiro: hasAnyPermissions>

## 数据权限

应用场景：某用户访问数据范围：公司及子公司，本公司，部门及子部门，本部门，当前用户，明细设置。

// 生成数据权限过滤条件（dsf为dataScopeFilter的简写，在xml中使用 ${sqlMap.dsf}调用权限SQL）

user.getSqlMap().put("dsf", *dataScopeFilter*(user.getCurrentUser(), "o", "u"));

<!-- 分页查询用户信息 -->

<select id=*"findList"* parameterType=*"User"* resultMap=*"userResult"*>

SELECT

<include refid=*"userColumns"*/>

FROM sys\_user a

<include refid=*"userJoins"*/>

WHERE a.del\_flag = '0'

<!-- 数据范围过滤 -->

${sqlMap.dsf}

</select>

/\*\*

\* 数据范围过滤

\* **@param** user 当前用户对象，通过“entity.getCurrentUser()”获取

\* **@param** officeAlias 机构表别名，多个用“,”逗号隔开。

\* **@param** userAlias 用户表别名，多个用“,”逗号隔开，传递空，忽略此参数

\* **@return** 标准连接条件对象

\*/

String dataScopeFilter (User user, String officeAlias, String userAlias)

## 智能分页组件

// 设置分页参数，则分页，如果不设置，则根据条件获取全部

user.setPage(page);

// 执行分页查询

page.setList(userDao.findPage(user));

## 树选择组件

标签文件：/ numberone /src/main/webapp/WEB-INF/tags/treeselect.tag

### 区域选择组件



<tags:treeselect id="area" name="area.id" value="${area.id}" labelName="area.name" labelValue="${area.name}" title="区域" url="/sys/area/treeData" cssClass="input-small" allowClear="true" notAllowSelectParent="true"/>

多选需要加 checked="true" 属性

### 公司选择组件



<tags:treeselect id="office" name="office.id" value="${user.office.id}" labelName="office.name" labelValue="${user.office.name}" title="部门" url="/sys/office/treeData?type=1" cssClass="input-small" allowClear="true" notAllowSelectParent="true"/>

多选需要加 checked="true" 属性

### 部门选择组件



<tags:treeselect id="office" name="office.id" value="${user.office.id}" labelName="office.name" labelValue="${user.office.name}" title="部门" url="/sys/office/treeData?type=2" cssClass="input-small" allowClear="true" notAllowSelectParent="true"/>

多选需要加 checked="true" 属性

### 人员选择组件



<tags:treeselect id="user" name="user.id" value="${user.id}" labelName="user.name" labelValue="${user.name}" title="用户" url="/sys/office/treeData?type=3" cssClass="input-small" allowClear="true" notAllowSelectParent="true"/>

多选需要加 checked="true" 属性

## 文件选择、文件上传组件

<form:hidden path="name" htmlEscape="false" maxlength="255" class="input-xlarge"/>

<tags:ckfinder input="name" type="files" uploadPath="/test "/>

## 富文本在线编辑器组件

<form:textarea id="name" htmlEscape="true" path="name" rows="4" maxlength="200" class="input-xxlarge"/>

<tags:ckeditor replace="name" uploadPath="/test " />

## 其它工具

/src/main/java/com/numberone / master/common/utils/ ...

Excel(Export/Import)、StringUtils、DateUtils、FileUtils、Encodes、CookieUtils、Collections3

# 封装及组件

## 工具类组件

1. UserUtils：用户工具类（获取当前用户的相关信息）。
2. CacheUtils：系统级别Cache工具类。
3. CookieUtils：Cookie操作工具类
4. DateUtils：日期时间工具类。
5. FileUtils：文件操作工具类。
6. StringUtils：字符串操作工具类。
7. ExcelExcel导入导出，参考用户导入导出例子。
8. Reflections：Java对象操作反射工具类。
9. BeanMapper：Bean与Bean，Bean与Conllection的互转
10. JaxbMapper：XML与Object互转。
11. JsonMapper：JSON与Object互转。
12. Cryptos、Digests：密钥工具类SHA1、MD5。
13. Collections3：集合对象工具类。
14. Encodes：各种编码转换工具类。
15. Exceptions：异常工具类。
16. FreeMarkers：FreeMarkers模板工具类。
17. Identities：唯一标识生成算法工具类（uuid、random）
18. PropertiesLoader：属性文件操作工具类
19. Threads：线程相关操作工具类。

## 自定义标签组件

1. ckeditor.tag：HTML在线编辑器。
2. ckfinder.tag：在线文件管理。
3. iconselect.tag：Icon图标选择。
4. message.tag：消息弹框。
5. validateCode.tag：验证码。

## EL 函数组件

1. fns.tld：系统相关EL函数。

## JavaScript 组件

1. jQuery：强大的js框架，动态特性，AJAX、插件扩展
2. jQuery Validate：基于jQuery的客户端校验插件。
3. jBox：基于jQuery的多功能对话框插件。
4. zTree：基于jQuery的树结构展示查看。
5. My7DatePicker：日期选择控件。
6. treeTable：基于jQuery的表格数显示插件。
7. CKEditor：富文本在线HTML编辑器。
8. CKFinder：在线文件资源管理器。
9. SuperSlide：基于jQuery的滑动门插件（滑动图片,Tab,横幅等）。

# 界面UI、CSS、JS

1. 主题CSS框架：<http://v2.bootcss.com/base-css.html>
2. 多功能下拉框：<http://ivaynberg.github.io/select2/>
3. 界面文档（查找class，查找css）：<http://www.bootcss.com/scaffolding.html>
4. 前端验证： /src/main/webapp/static/jquery-validation/1.11.0/demo/index.html
5. 弹出对话框： /src/main/webapp/static/jquery-jbox/2.3/docs/jbox-demo.html

top.$.jBox.open('iframe:'+href,'查看评论',$(top.document).width()-220,$(top.document).height()-180,{

buttons:{"关闭":true},

loaded:function(h){

$(".jbox-content", top.document).css("overflow-y","hidden");

$(".nav,.form-actions,[class=btn]", h.find("iframe").contents()).hide();

$("body", h.find("iframe").contents()).css("margin","10px");

}

});