

## 一、列表 js 使用说明

变量或方法	说明	参数说明
_viewer	列表的上下文句柄	
servId	当前服务 ID	
getParHandler()	获取当前页面的父句柄	
listBarTipLoad(msg)	列表显示加载的提示信息	msg:提示内容
listBarTip(msg)	列表显示成功的提示信息	msg:提示内容
listBarTipError(msg)	列表显示错误的提示信息	msg:提示内容
listClearTipLoad()	清除列表的提示	
doAct(act,reload,async)	列表页面与后台交互方法，调用此方法将自动将选中的记录主键值和关联信息传递到后台	act: 对应方法 reload:执行后是否刷新列表 async:true 异步，默认同步
doServAct(servId,act,reload)	列表页面与后台交互方法	served:服务 ID act: 对应方法 reload:执行后是否刷新列表
doActReload(act)	列表页面与后台交互方法，成功后刷新页面	act:对应方法，默认匹配当前服务的监听
getNavTreeObj(dictId)	获取导航树对象	dictId:树形对应的字典 ID
getBtn(actCode)	获取按钮对象	actCode: 按钮对应 ID
afterDelete()	列表删除方法后执行	
refresh(options)	刷新列表（不包括树形）	options: json 格式对象，不写默认刷新当前列表
refreshTreeAndGrid(options)	刷新列表(包括树形)	options: json 格式对象，不写默认刷新当前
onRefreshGridAndTree()	调用此方法后，修改卡片后刷新的列表包括树形	
getNavTreeObj(dictId)	获取对应字段编码的导航树对象	dictId:字典编码
beforeTreeNodeClickLoad(item,id,dictId)	在导航树的点击事件加载前	Item : 节点对象 Id : 节点 id dictid : 字典编码

afterTreeNodeClickLoad(item,id,dictId)	在导航树的点击事件加载后	Item：节点对象 Id：节点 id dictid：字典编码
resetSize()	重置当前页面大小	
shield()	当前页面处于透明屏蔽状态	
shieldHide()	取消列表屏蔽状态	
addRedHeader(text, style)	在列表增加红头文字	text: 文字 style: 文字样式
getParams()	获取传递的 params 参数	
grid.getSelectedPKCodes()	获取选中行的主键值 数组集合，如： [22,33,44]	
grid.getSelectedRowIndexes()	获取选中行的索引值 数组集合，如： [22,33,44]	
grid.getSelectedItemCodes(itemCode)	获取选中行的某字段 数组集合，如： [22,33,44]	itemCode:字段 ID
grid.getRowItemValue(pkCode, itemCode)	获取某行的某个字段 值	pkCode:主键值 itemCode:字段 ID
grid.getHeadCheckBox()	获取列表头部的复选 框的 jQuery 对象	
grid.getCheckBox()	获取列表的所有复选 框集合	
grid.getCheckBoxItem(pkCode)	根据主键获取某个复 选框对象	pkCode: 主键值
grid.getTable()	获取列表的表格对象	
grid.isEditable()	判断列表是否可编辑 状态	
grid.getColumnDef(itemCode)	取得指定字段(列)定 义	itemCode: 列的字段编码

## 二、卡片 js 使用说明

变量或方法	说明	参数说明
_viewer	卡片的上下文句柄	
servKeys	当前页面的主键编号	
servId	当前页面的服务 ID	
servName	当前页面的服务名称	

saveReturn	设置 saveReturn=true;则保存卡片后自动关闭当前卡片;	
getPKCode()	获取当前卡片主键值	
getParHandler()	获取当前卡片的父句柄	
cardBarTipLoad(msg)	卡片显示加载的提示信息	msg:提示内容
cardBarTip(msg)	卡片显示成功的提示信息	msg:提示内容
cardBarTipError(msg)	卡片显示错误的提示信息	msg:提示内容
cardClearTipLoad()	清除卡片的提示	
doAct(act,codes,reload,extendData,async)	卡片页面与后台交互方法	act: 对应方法 codes:需传递到后台的字段编号, 多个以逗号分隔; 没有 codes, 则默认传递所有字段; extendData: 扩展的 json 格式的数据; reload:执行后是否刷新卡表, 默认刷新 async:true 异步, 默认同步
doActReload(act,codes)	卡片页面与后台交互方法, 成功后刷新页面	act: 对应方法 codes:需传递到后台的字段编号, 多个以逗号分隔; 没有 codes, 则默认传递所有字段;
getBtn(actCode)	获取按钮对象	actCode: 按钮对应 ID
setBtnText(actCode,text)	修改指定按钮的名称	actCode: 按钮对应 id text: 按钮显示文字
refresh()	刷新列表 (不包括树形)	刷新卡片页面
resetSize()	重置当前页面大小	
shield()	当前页面处于透明屏蔽状态	
shieldHide()	取消列表屏蔽状态	
getItem(itemCode)	获取字段对象的方法	itemCode:字段编码
getItemConfig(itemCode)	获取字段的配置定义	itemCode:字段编码
getItem(itemCode).getValue()	获取字段值	itemCode:字段编码
getItem(itemCode).setValue(value)	设置字段值	value:设置值
getItem(itemCode).getContainer()	获取字段的容器对象	itemCode:字段编码 调用 hide()即隐藏 调用 show()即显示
getItem(itemCode).getLabel()	获取字段前面的 Label 字符	
itemValue(code)	获取卡片单个字段值的方法, 返回当前值	codes:字段 ID, 多个以, 分割
itemValues(codes)	获取卡片多个字段值的方法,	codes:字段 ID, 多个以,

	返回当前值	分割
getDefaultData(itemCode)	获取默认值数据	itemCode:字段 ID
getByIdData()		修改时获取的是业务数据，添加的时候获取的是默认值
backClick()	模拟点击返回按钮	
setMindDisabed()	意见只读	
setMindEnabled()	意见可写	
beforeSave()	可覆盖此方法,在保存之前执行；  如： _viewer.beforeSave = function() {alert(1);};	
afterSave()	可覆盖此方法,在保存之后执行；  如： _viewer.afterSave = function() {alert(1);};	
getChangeData()	获取卡片上修改的 json 格式数据对象； 可用 jQuery.isEmptyObject(data)判断是否为空	
saveForm()	保存 form	
refreshFile()	刷新文件	
getAllFileData()	获取所有未删除和新添加的文件数据	
disabledAllFile()	隐藏所有附件	
readonlyAllFile()	改变所有附件的权限，只可读	
hideGroup(id)	隐藏指定 ID 的分组框，包括里面内容	Id: 分组框 id
showGroup(id)	显示指定 ID 的分组框，包括里面内容	
expandGroup()	展开指定 ID 的分组框	
collapseGroup()	收缩指定 ID 的分组框	
setReadOnlyVar(value)	设置只读变量，用户设置后对点击进入关联功能起作用	
readCard()	设置当前页面的只读，包括按钮条、form、文件、意见、关联功能	
addRedHeader(text, style)	在列表增加红头文字	text: 文字 style: 文字样式
getPKCode()	获取当前主键值	

getParams()	获取传递的 params 参数	
getMindItem()	取得意见(自定义)字段	
getServSrcId()	取得 SERV_SRC_ID 的值	
tabHide(sid)	设置子 tab 的隐藏	sid: 子 tab 的对应服务
tabReadOnly(sid)	设置子 tab 的只读	sid: 子 tab 的对应服务
tabFocus(sid)	设置子 tab 的触发	sid: 子 tab 的对应服务
headMsg()	增加卡片页面的和工作流相关提示信息	

## 三、系统常量

系统提供一些常量的定义，可直接使用

常量编码	值	
UIConst.PK_KEY	"_PK_"	
UIConst.RTN_MSG	"_MSG_"	
UIConst.RTN_OK	"OK,"	
UIConst.RTN_ERR	"ERROR,"	
UIConst.RTN_WARN	"WARN,"	
UIConst.STR_YES	"1"	
UIConst.STR_NO	"2"	
UIConst.ACT_CARD_MODIFY	"cardModify"	
UIConst.ACT_CARD_ADD	"cardAdd"	
UIConst.CARD_STATUS	"_ADD_"	卡片状态

可参考文件：Constant.js

## 四、系统变量

系统级变量包括如下：

变量名称	值
日期	
@DATE@	当前日期，长 10 位，例如：2012-01-23
@DATE_YEAR@	当前年度，长 4 位：2012
@DATE_MONTH@	当前月度：3
@DATE_YEARMONTH@	当前年月，长 7 位：2012-03
@DATE_CN@	当前日期：2012 年 07 月 03 日
@DATE_WEEK_DAY_CN@	当前星期几，例如：星期一、星期二...
当前用户信息	

@USER_CODE@	用户编码
@USER_NAME@	用户名称
@LOGIN_NAME@	登录用户名
@USER_POST@	用户岗位
@USER_SEX@	性别
@USER_IMG@	截取后的头像文件
@USER_IMG_SRC@	头像原始文件
@CMPY_CODE@	公司编码
@CMPY_NAME@	公司名称（简称）
@DEPT_CODE@	部门编码
@DEPT_NAME@	部门名称
@TDEPT_CODE@	有效部门编码
@TDEPT_NAME@	有效部门名称
@DEPT_SRC_TYPE1@	处室扩展类型 1
@DEPT_SRC_TYPE2@	处室扩展类型 2
@OFFICE_PHONE@	办公电话
@USER_MOBILE@	手机
@USER_EMAIL@	邮箱
@ROLE_CODES@	当前用户角色列表，逗号分隔
@GROUP_CODES@	用户所有群组，单引号包含，逗号分隔
@ADMIN_GROUP_CODES@	用户所有管理群组
@DEPT_CODES@	用户所有部门，（当前、父、父的父...）
@ODEPT_CODE@	当前机构编码（部门类型为机构）
@ODEPT_NAME@	当前机构名称（部门类型为机构）
@ODEPT_SRC_TYPE1@	机构部门扩展类型 1
@ODEPT_SRC_TYPE2@	机构部门扩展类型 1
@ODEPT_CODE_PATH@	机构代码路径
@ODEPT_PCODE@	父机构代码
@ODEPT_LEVEL@	机构所在层级
@SUB_CODES@	所有工作委托给当期人的用户列表
@LEAD_CODES@	秘书对应领导编码列表
@JIAN_CODES@	所有设定跟当前人工作兼岗的用户列表
@AGT_FLAG@	当前用户是否出于委托别人办理状态
@USER_PT@	用户模版
@DEPT_PT@	处室模版
@TDEPT_PT@	部门模版
@ODEPT_PT@	机构模版
@CMPY_PT@	公司模版
其它	
@urlPath@	系统虚路径

获取系统变量在卡片 js 或列表 js 中可直接通过 `System.getVar(参数)` 获取。如获取用户编码：

`System.getVar("@USER_CODE@")` 即可获取当前登录用户编码，以此类推即可。

## 五、系统配置

1、在【系统配置】服务里，增加一条记录，设置[前端使用?]为是则可以在前端获取到该变量的值。

获取系统配置参数，需要增加前缀 `C_`，如：`System.getVar("C_SY_PAGE_TITLE");`

2、其中系统预置了一些系统配置，在开发的过程中如果需要，则需要配置其值，并启用即可。

系统预定义配置	用途	配置值说明
SY_BANNER_CONFIG	头部 banner 的配置	<code>{'banner':'/lw/page/img/head_logo.gif','bannerBack':'/lw/page/img/head_bg.gif','bannerTabBack':'/lw/page/img/rh-head-menu-blue.png'}</code> 说明： banner:banner 的含 login 的图片 bannerBack:整个 banner 的背景图片 bannerTabBack:banner 下方的 tab 条的背景
SY_DESK_MSG_INTERVAL	待办提醒更新时间	<code>300000</code> 说明： 待办提醒更新时间，单位毫秒，默认值为 5 分钟即 300000ms
SY_FONT_BIG	系统大字体显示	<code>true</code> 为 true，则系统将启用大字体风格
SY_HOME_CONFIG	登录后首页的配置	<code>{'tTitle':'系统首页','url':'SY_COMM_TEMPL.show.do','menuId':'hua xiaLW_Main','closeFlag':false}</code> 有此参数则直接进入相应系统的主页
SY_MB_LINK	系统页面是否显示进入手机系统链接	<code>true</code>
SY_PAGE_TITLE	登录后浏览器标题	<code>某某公司-管理系统</code>
SY_PJ_CSS	项目级扩展 css 文件	<code>/sy/test.css</code> 项目级公用 css 文件定义
SY_PJ_JS	项目级扩展 js 文件	<code>/sy/test.js</code>
SY_PORTAL_ICON	系统使用的左上角 LOGO 图标	<code>/sy/comm/page/img/logo_ruaho.gif</code>
SY_STYLE_DEF	系统默认风格定义	<code>{'SS_STYLE_MENU':'pageBody-lightBlue','SS_STYLE_BACK':'bodyBack-27','SS_STYLE_BLOCK':'conHeaderTitle-blue'}</code>

		SS_STYLE_MENU: 定义菜单样式 SS_STYLE_BACK: 内容列表区的背景 SS_STYLE_BLOCK: 内容列表区的头部
SY_STYLE_FEST	节日风格覆盖	<pre>{'SS_STYLE_MENU':'pageBody-orange','SS_STYLE_BACK':'bodyBack-gray','SS_STYLE_BLOCK':'conHeaderTitle-orange'}</pre> <p>参考上面默认风格设置， 此配置值优先级&gt;个人风格(优先级)&gt;系统默认风格(优先级)</p>
SY_TAB_COLOR	tab 颜色	<pre>{'1':{'li':'lightBlueLi','a':'lightBlueA'}}</pre> <p>一共 5 种，可选取其中几种作为指定的颜色：  <pre>{'1':{'li':'qingLi','a':'qingA'},'2':{'li':'yellowLi','a':'yellowA'},</pre>  <pre>'3':{'li':'grayLi','a':'grayA'},'4':{'li':'blueLi','a':'blueA'},</pre>  <pre>'5':{'li':'redLi','a':'redA'},'6':{'li':'lightLi','a':'lightA'},</pre> </p>
SY_TAB_MAX	系统标签最大个数	10 最外层系统的标签打开个数限制
SY_TOP_ALERT	是否启用顶部的提醒下拉面板	true
SY_WBIM_FLAG	即时通讯启用标志	true 即时通讯启用标志 true 为启用，默认不启用
SY_WFE_REMIND_CODE	催办单代字的默认值	催办
SY_PJ_BACKBTN	是否返回按钮显示在列表卡片	true 设置返回按钮的位置,默认显示在标题条

## 六、系统方法

注：红色参数为必填项

方法	说明	参数说明
FireFly.byId(sld,id)	获取某个服务的某条记录值	sld:服务编码 id:记录主键
FireFly.cardModify(sld,data)	修改某条记录	sld:服务编码 data:json 格式的数据或字符串格式的 json 字符串，如： <pre>{ "_PK_": "test", "USER_CODE": "test", "USER_NAME": "testName" }</pre> 只构造需要修改的字段即可，必须参数。



		其中 <b>_PK_</b> 为必填参数标识哪一条记录需要修改
FireFly.cardAdd( <b>sld</b> , <b>data</b> )	增加一条记录	sld:服务编码 data:json 格式的数据或字符格式的 json 字符串，如： {“USER_CODE”:“test”,“USER_NAME”:“testName”}。保存成功后将返回提示信息和主键信息
FireFly.batchSave( <b>sld</b> , <b>data</b> ,callback,tipFlag,async)	批量保存多条记录	sld:服务编码 data:json 格式的数据或字符格式的 json 字符串，此为两层的 json 数据，如： {'ADDFILE': [{'123': 'qwe'}, {'345': 'tyu'}], 'CODE': 'uiop'}。必须参数 Callback: 为成功执行后的回调方法。 TipFlag: 是否显示提示信息，默认为 true 显示 Async: 是否异步，默认为 false，非异步执行
FireFly.listDelete( <b>sld</b> , <b>data</b> ,tipFlag)	删除记录	sld:服务编码 data:json 格式的含主键信息的对象，如： {“_PK_”:“test1,test2,test3”} TipFlag: 是否显示提示信息，默认为 true 显示
FireFly.getListData( <b>sld</b> , <b>datas</b> )	获取列表	sld:服务编码 data:json 格式的对象，其中 data["_NOPAGE_"] = true; 标识查询全部数据; data["_searchWhere"] = " and S_USER='admin'"; 设置查询的条件; 调用方式
FireFly.getCardFile( <b>sld</b> , <b>dataId</b> , <b>itemCode</b> )	获取某服务的文件信息 其中: itemCode 有此参数则获取某字段的文件信息; 此参数为空则获取当前服务的关联文件信息	sld:服务编码 dataId:记录主键信息 itemCode:文件字段信息,
FireFly.getDict( <b>dictCode</b> ,pid,	获取字典信息	dictCode:字典编码

extWhere,level,params)		pid:某节点编码 extWhere:扩展条件 level:层级 params:任意参数提交给后台
FireFly.doAct(sId,ald,data,tipFlag,async,func)	执行某服务的某个方法和后台交互	sId:服务编码 ald:方法编码 data:json 格式的数据，用于传递给后台使用，如： {“USER_CODE”:”123”} tipFlag:是否提示信息，默认为 true，显示提示 async:异步标识，默认为 false，非异步执行 func:成功执行后的回调方法
JsonToStr(o)不推荐	将 json 对象转为字符串	O:json 对象 注：也可用 jQuery 的自带方法实现此功能
jQuery.toJSON(jsonObj)	将 json 转化成字符串	
StrToJson(strData)不推荐	字符串转换为 json 对象	注：简单的 json 对象转换，更复杂转换请用下面方法
jQuery.parseJSON(jsonStr)	将字符串转化成 json	
Tab.open(options)	打开一个新 tab 标签 调用方式如：  <pre>var params = {}; params["test"]="123"; params["hello"] = "world!"; var options= { "url": "SY_ORG_DEPT.list.do", "title": "test", "params": params, "menuFlag": 3}; openNewTab(options);</pre>	tTitle:必须 tab 显示标题 url:必须对应链接 打开列表： SY_USER.list.do?readOnly=true 打开卡片： SY_USER.card.do?pkCode=123 &readOnly=true 调后台服务： SY_USER.selfact.do?data={'params1': 'test'} 自定义页面： /sy/home.jsp?readOnly=true refreshFlag:每次都刷新当前页面 params:外部公共对象参数，供 iframe 内部调用 scrollFlag:是否有滚动条 menuFlag: 1、显示当前层级 2、全部显示 3、不显示 4、没有对应菜单，但是显示
Tab.close()	关闭当前页面的父 tab 标签	

以下方法为系统提供的工具类方法，具体调用可参考 tools.js 文件

#### 扩展的 jQuery 方法

- \* jQuery.parseJSON(jsonStr); //将字符串转化成 json
- \* jQuery.toJSON(jsonObj); //将 json 转化成字符串
- \* jQuery.contextmenu(); //右键菜单
- \* jQuery.textSearch(str, options); //高亮页面文本
- \* jQuery.evalJSON(); //-----
- \* jQuery.secureEvalJSON(); //-----
- \* jQuery.quoteString(str); //-----
- \* jQuery.extend(opts, default); //合并两个对象
- \* jQuery.timeago(timeago); //给定时间，返回字面显示，如返回“刚刚”，“2 分钟前”，“昨天”等
- \* JsonToStr(obj); //将 json 转化成字符串，不推荐使用
- \* StrToJson(str); //将字符串转化成对象，不推荐使用

#### rhDate 相关

- \* rhDate.getCurentTime(); //获取系统当前时间，返回结果：2013-01-23 16:28
- \* rhDate.stringToDate(); //将时间格式的字符串转化成时间
- \* rhDate.getTime(); //获取系统当前时间
- \* rhDate.getAreaDate(); //根据小时获取当前系统的中文区块时间
- \* rhDate.doDateDiff(); //求时间差
- \* rhDate.nextDate(); //某个日期减去天数得到另外一个日期
- \* rhDate.getTime(); //取得格式化事件 2012-03-01 07:06:02
- \* rhDate.pattern(fmt); //日期格式化，根据格式化类型，返回日期字符串 yyyy-MM-dd
- \* rhDate.dateDiff(strInterval, dtEnd); //获取传入时间与当前时间差 dateDiff("d","2013-1-28"),返回当前时间与2013-1-28相差天数，参数d表示天，也可以是y表示年，则返回相差年数

#### Cookie

- \* Cookie.get(cookName); //读cookie操作,参数: cookie名称返回值: 字符串
- \* Cookie.set(sName, sValue, oExpires, sPath, sDomain, bSecure); //写cookie操作 sName: cookie名称.sValue: cookie值,oExpires: 过期时间
- \* Cookie.del(sName); //删除cookie操作sName : cookie名称

#### Tip

- \* Tip.show(msg, parentFlag, scopeObj); //提示信息条，定时会隐藏, msg显示顶部提示信息, parentFlag 父层对象的标识
- \* Tip.showAttention(msg, parentFlag, scopeObj); //提示信息条，定时会隐藏, msg 显示顶部提示信息, parentFlag 父层对象的标识
- \* Tip.showError(msg, parentFlag, scopeObj); //提示信息条，定时会隐藏, msg显示顶部提示信息, parentFlag父层对象的标识
- \* Tip.showLoad(msg, parentFlag, appendObj, timeout, scopeObj); //加载信息提示, msg显示顶部提示信息, parentFlag父层对象的标识
- \* Tip.clear(); //外层清除提示信息
- \* Tip.clearLoad(); //删除加载提示信息条

#### Format

- \* Format.formatNum(str); //系统数据格式化方法
- \* Format.clearNum(strNum); //去除逗号


<ul style="list-style-type: none"> <li>* Format.(size, fix, value) ;//-----</li> <li>* Format.limit(size, value);//-----</li> <li>* Format.substr(start, end, value);//截串</li> <li>* Format.replaceDbQuotes(str);// 双引号替换为空串</li> <li>* Format.replaceSinQuotes(str);/// 单引号替换为空串</li> <li>* Format.RMBCapital(number);//人民币大写转换,numberValue 人民币小写</li> </ul>
Load
<ul style="list-style-type: none"> <li>* Load.js(pathUrl, viewer);//动态加载js, 并执行,pathUrl js的路径viewer 列表/卡片js中的viewer 对象</li> <li>* Load.scriptJS(pathUrl);//动态加载js,放入到&lt;script&gt;标签, 判断重复加载</li> </ul>
Tools
<ul style="list-style-type: none"> <li>* Tools.getFileSuffix(fileName);//取出文件扩展名</li> <li>* Tools.redirect(url);//重定位url</li> <li>* Tools.toMbIndex();//判断当前登录页面, 如果为手机登录页则跳转</li> <li>* Tools.lTrim(val);//-----</li> <li>* Tools.rTrim(val);//-----</li> <li>* Tools.itemVarReplace(str, data);//系统字段级变量替换</li> <li>* Tools.systemVarReplace(str);//系统级变量替换</li> <li>* Tools.parVarReplace(str);//父层级级变量替换</li> <li>* Tools.rhSetBodyBack();//根据个性化的数据, 设置背景的方法</li> <li>* Tools.xdocUrlReplace(url);//xdoc的url特殊处理替换</li> <li>* Tools.rhReplaceId(url);//替换菜单的info中的字符变量为id可用的字符等</li> <li>* Tools.rhReplaceHtmlTag(value);//html格式字符替换</li> <li>* Tools.getTreeLeafClass(dictId);//根据字典编号返回对应的叶子节点class</li> </ul>
Browser
<ul style="list-style-type: none"> <li>* Browser.versions();//rh封装的判断浏览器类别,Browser.versions.iPad</li> </ul>
Select
<ul style="list-style-type: none"> <li>* Select.usualContent(opts,viewer);//系统公用, 弹出查询选择的常用语, 可添加、修改、删除;viewer 页面上下文</li> <li>* Select.openSendSchem(target);//系统共用, 分发方案查询选择框,target 接受选择值的dom对象名称, 如: id~name</li> </ul>
Mouse
<ul style="list-style-type: none"> <li>* Mouse.getMousePoint(event);//获取鼠标在页面上的位置</li> <li>* Mouse.dialogCoordinate(x,y,width,height);//width 弹出框宽度, height 弹出框高度, x 鼠标x坐标, y鼠标y坐标</li> <li>* Mouse.diagonal(x1,y1,x2,y2);//面积, 即鼠标与当前窗口围成的上下左右四块面积, 用数学解决逻辑问题; //x1,y1弹出框的宽高, x2,y2鼠标和最外边框组成的最大区域坐标</li> <li>* Mouse.getScrollTop();//获取当前窗口滚动条高度</li> <li>* Mouse.dialogPosition(event,width,height)//获取子页面的弹出框位置坐标, 返回position{"x":x,"y":y}</li> </ul>
Todo
<ul style="list-style-type: none"> <li>* Todo.dbClickGrid(viewer,grid);//响应双击Grid事件, 打开待办,viewer 包含grid的Viewer对</li> </ul>

<p>象,grid 被双击的grid对象</p> <p>* Todo.open(sld, title, url, con, todold, objectID1);//打开待办</p> <p>* Todo.getCount ();//获取待办的总数量, 如果数量有变化则调用pageView内方法动态更新下拉面板</p> <p>* Todo.get(num,rowNum);//代办的获取, 系统顶部下拉面板内数据的获取</p> <p>* Todo.getData(rowNum);//代办的获取, 系统顶部下拉面板内数据的获取</p> <p>* Todo.openEntity(_viewer);//对列表上选中行双击查看绑定事件,双击打开该选中行的服务卡片页面</p>
Window
<p>* Window.getViewPort(target);//获取指定window的可视区域尺寸</p> <p>* Window.getScroll(target);//获取指定window的滚动条的信息</p> <p>* Window.searchScrollBegin();//搜索部分启用外部滚动条的监听事件</p>
RHWF
<p>* RHWF.showLeaderMind(grid);//判断列表页中是否存在“S_HAS_PS_MIND”列, 如果存在列并有值, 则展示“对钩”的小图片。当鼠标放到小图片上后, 显示领导意见。</p>
其他
<p>* showRHDialog(title,content,func,handler,widHeiArray);//显示提示信息框通用方法,title 标题, content 内容,func回调方法,handler 回调句柄</p> <p>* Stopwatch();//辅助调试计时器</p>

## 七、配置说明

输入类型	输入配置	说明
下拉框	字典编码, {'SEARCHHIDE':true,'SEARCHTYPE':multi} 例: SY_YESNO,{ 'SEARCHHIDE':true }	1 字典: 对应的字典编码; 2、SEARCHHIDE:可选, 只在高级查询时起作用, 高级查询里显示与否; 3、SEARCHTYPE: 可选, 只在高级查询起作用, 高级查询里显示多选;
查询选择	SY_SERV_ITEM,{ 'TARGET':SERV_NAV_ITEM~,'SOURCE':ITEM_CODE~ITEM_NAME,'EXTWHERE': and SERV_ID=^#SERV_ID#^ and DICT_ID is not null,'SEARCHEXTWHERE': and 1=1,'TYPE':multi,'HIDE':ITEM_CODE~ITEM_NAME} 例: 如上..	1 SY_SERV_ITEM: 对应的字典 ID; 2、'TARGET': 回写到页面上的字段 ID; 3、'SOURCE': 打开选择的页面的显示的字段列; 4、'EXTWHERE': 过滤列表显示数据的条件; 5、#SERV_ID#: 页面上的字段变量; 6、'TYPE': 单选, singel; 多选, multi; 7、SEARCHEXTWHERE: 高级查询里的查询选择的覆盖条件; 8、SEARCHHIDE:可选, 只在高级查询时

		<p>起作用，高级查询里显示与否；</p> <p>9、SEARCHTYPE：可选，只在高级查询起作用，高级查询里显示多选；</p> <p>10、DATAFLAG:false,打开选择页面默认不显示列表数据；</p> <p>11、HIDE:设定查询选择的某些列是否显示，此显示为虽隐藏但有数据；</p>
树形选择	<p>SY_ORG_DEPT,{ 'TYPE':'multi','EXTWHERE':' and 1=1','extendDicSetting':{},'rtnLeaf':true,'SEARCHHIDE':'true','SEARCHTYPE':'multi'}</p> <p>例：如上..</p>	<p>1、SY_SERV_ITEM：对应的字典 ID；</p> <p>2、TYPE:multi 多选； 默认单选</p> <p>3、EXTWHERE：过滤条件；</p> <p>4、extendDicSetting：扩展的树形设置条件；</p> <p>5、SEARCHHIDE:可选，只在高级查询时起作用，高级查询里显示与否；</p> <p>6、SEARCHTYPE：可选，只在高级查询起作用，高级查询里显示多选；</p> <p>7、rtnLeaf:只返回叶子节点；</p> <p>8、LEVEL:设置当前树的数据获取层级；</p> <p>9、PID:设置当前树从某个节点显示。</p> <p>10、extendDicSetting 对树的扩展设置，一般情况下不需要此项，为 json 对象格式，用于设置树的级联选中、显示层级等。如： {'rhexpand':false,'expandLevel':2,'cascadecheck':false,'checkParent':false,'childOnly':true}</p> <p>其中：rhexpand：是否展开全部节点 expandLevel：展开到第几层 cascadecheck：级联选中 checkParent：选中子时是否自动选中父 childOnly：只允许选择叶子</p> <p>11、MODEL:显示模式，和多选配合使用，此参数为 link 时，字典右侧显示预选择的节点容器。</p>
相关文件选择	<p>例：</p> <p>[[{'servId':'SY_COMM_KM','SOURCE':'KM_ID~KM_TITLE','SHOWITEM':'S_USER__NAME','servName':'知识库'},{'servId':'SY_COMM_TEST','SOURCE':'TEST_ID~TEST_NAME','SHOWITEM':'TEST_ID','servName':'测试服务','TARGET_SERV_ID_ITEM':'SERV_ID'}]]</p>	<p>servId：服务编码</p> <p>SOURCE：显示的列</p> <p>servName：显示名称</p> <p>SHOWITEM:设置某个字段显示在提示块</p> <p>TARGET_SERV_ID_ITEM:被选择数据的服务 ID</p>

嵌入服务 (自定义)	例：SY_ORG_USER	对应服务编码，此服务应为当前服务的关联服务。暂只支持显示方式为列表的嵌入。
Iframe (自定义)	例： <iframe width='600px' height='400px' src="@urlPath@/sy/comm/home/portaIComView.jsp?id=#PC_ID#" border=0 frameborder=1></iframe>	对应的的 iframe 写法，可以替换系统变量和当前的字段级变量值
静态文本 (label)	例： 我是一个静态文本  <span style='color:red'>Helloworld!</span>	<p>此字段对应自定义字段，用于在卡片显示一段静态文本。可通过默认值设定添加时的显示文本。</p> 
分组框	设置为【分组框自定义(自定义)】，需和自定义字段配合使用	对应自定义字段，用于分组卡片字段，从此字段开始往下分组，直到遇到下一个分组框字段。
文件上传	需要是表字段	有别于【附件(自定义)】，此字段为表字段，存储时会在表字段中存放相应数据。暂只支持一个文件上传。
超大文本	需要是表字段	用于显示一个文本编辑器。





工作流节点上配置某个节点上文件的权限

一般

组织/资源

操作权限

数据权限

文件权限

自定义变量

文件类型	查看	编辑	添加	删除
附件2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
附件	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

确定

取消

默认会取服务字段中附件(自定义)的字段作为类别  
其权限值，默认为字段配置中的 VALUE 值。

意见字段

服务的字段控制

基本设置

编码 \*

\_MIND\_LEADER

名称 \*

领导意见

类型 \*

自定义字段

移动类型 \*

卡片项（移动）

输入类型 \*

静态文本(label)

输入模式 \*

无

输入设置

{'MIND\_CODE': 'SP-0004', 'TMPL\_FTL': 'labelMindOneLine.ftl', 'SHOW\_TYPE': 'ALL'}

编码	即意见字段的编码，以 _MIND_ 开头。如 _MIND_LEADER
名称	即页面显示名称，如领导意见
类型	需要定义为自定义字段
输入类型	需要定义为静态文本(label)
输入设置	配置成类似 {'MIND_CODE': 'SP-0004', 'TMPL_FTL': '/sy/comm/mind/labelMindOneLine.ftl', 'SHOW_TYPE': 'ALL'}或者 SP-0004 MIND_CODE(必填)意见编码如 SP-0004 TMPL_FTL (可选)配置显示意见的模板，如'labelMindOneLine.ftl' 如果没有会用默认的模板/sy/comm/mind/labelMindDefault.ftl SHOW_TYPE(可选)FIRST/LAST/ALL 如果不填，默认为 ALL 同一种意见编码有可能有多条意见，如果配置成 FIRST , 将取第一条 配置成 LAST,将取最后一条显示 配置成 ALL,或不填，会将所有的都列出来

如图所示，{'MIND\_CODE': 'SP-0004', 'TMPL\_FTL': 'labelMindOneLine.ftl', 'SHOW\_TYPE': 'ALL'}

审批单名称:	测试上传
领导意见:	我是管理员 公有的数据h 袁安安 2013-01-05 09:40:06:362 我是管理员 公有的数据1234 袁安安 2013-01-04 17:51:04:357
主送:	

意见自定义字段

编码 *	MIND_ITEM
类型 *	自定义字段
输入类型 *	意见(自定义)
输入设置	{'MIND_LIST_ORDER': 'TIME'}

支持默认按时间(TIME) / 类型(TYPE)排序

组合多个字段的配置:

在字段项的扩展设置

里这么配置: {'COMBO':{'ITEM\_CODES':'TEST\_FLAG,S\_CMPY','SUFFIX':'号','SEP':'/'}}

该字段是一个多用途字段, 所以必须用标准 json 串来配置, COMBO 是组合字段配置, 里面的 ITEM\_CODES 表示要组合在一起的字段项, 逗号分隔, 不需要加上当前字段项。SUFFIX 是组合之后在最后面加的后缀。SEP 表示各个组合字段之间的分隔符, 不是必须的。各个字段显示宽度需要自己定义。

给组合字段增加用括号或者其它符号括起来的功能, 配置如下:

```
{'COMBO':{'ITEM_CODES':'CT_YEAR,CT_SERIAL','SUFFIX':'','SEP':'-','WRAP':{'CT_YEAR':{'BEFORE':'(','AFTER':')'}}}}
```

增加了'WRAP':{'CT\_YEAR':{'BEFORE':'(','AFTER':')'}}', 它的意思是给年份前后都分别加上左括号和右括号, 也可以给多个字段添加。

在字段的扩展设置里增加了重新设置位置的功能, 配置方法:

```
{'RESET_POSITION':{'POSITION':'LEFT_TOP','WIDTH':'100'}}
```

POSITION 可以为: LEFT\_TOP、CENTER\_TOP、RIGHT\_TOP

WIDTH 为: 该字段输入框的宽度, 在这配置宽度是为了计算离右边的距离

字段里的宽度不能修改, 使用默认的 0

字段的列表 URL 配置:

```
Tab.open({"url":"OA_GW_TMPL_QS_JGBM.card.do?pkCode=#GW_ID#","tTitle":"#GW_TITLE#","menuFlag":3})
```

这是一段 javascript 代码, 它会在列表上把该字段设置成超链接, 点击超链接的时候会执行该段代码, ##之间是该服务的字段项, 这些字段项必须是显示或者隐藏有数据的那些字段, OA\_GW\_TMPL\_QS\_JGBM 是要打开的服务。这里的引号必须是双引号。

其它

1. 访问卡片页面:  
[http://localhost:9009/core/view/stdCardView.jsp?sId=SY\\_ORG\\_USER&pkCode=admin](http://localhost:9009/core/view/stdCardView.jsp?sId=SY_ORG_USER&pkCode=admin)
2. 访问列表页面:  
[http://localhost:9009/core/view/stdListView.jsp?frameId=SY\\_SERV-tabsIframe&sId=SY\\_SERV&paramsFlag=false&readOnly=false&extWhere=&dataFlag=undefined&parWhere=undefined&parVar=undefined&type=undefined](http://localhost:9009/core/view/stdListView.jsp?frameId=SY_SERV-tabsIframe&sId=SY_SERV&paramsFlag=false&readOnly=false&extWhere=&dataFlag=undefined&parWhere=undefined&parVar=undefined&type=undefined)
3. 按钮图标:  
参考文件 Icon.css。如: *btn-add*
4. 修改桌面图标:  
sy\desk\css\images\app\_rh-icons 目录下, 和工作台设置里的图标设置
5. 左侧菜单图标, 通过在菜单里设定相应的图标:  
对应文件 common.css 里, 如: *.ui-state-default.leftMenu-bangong*

◆更多项目 js 示例, 请参考/sy/servjs/demo.js 示例文件。