版本：1.2.4修改日期：2013/07/26

# 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.getSelectPKCodes() | 获取选中行的主键值数组集合，如：[22,33,44] |  |
| grid.getSelectRowIndexs() | 获取选中行的索引值数组集合，如：[22,33,44] |  |
| grid.getSelectItemCodes(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 |

# 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，返回当前值 | codes:字段ID，多个以,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 |  |

# 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

# 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。

# 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 | {'banner':'/lw/page/img/head\_logo.gif','bannerBack':'/lw/page/img/head\_bg.gif','bannerTabBack':'/lw/page/img/rh-head-menu-blue.png'}  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 | 300000  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 | true  为true，则系统将启用大字体风格 |
| SY\_HOME\_CONFIG | 登录后首页的配置 | {'tTitle':'系统首页','url':'SY\_COMM\_TEMPL.show.do','menuId':'huaxiaLW\_Main','closeFlag':false}  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 | true |
| SY\_PAGE\_TITLE | 登录后浏览器标题 | 某某公司-管理系统 |
| SY\_PJ\_CSS | 项目级扩展css文件 | /sy/test.css  项目级公用css文件定义 |
| SY\_PJ\_JS | 项目级扩展js文件 | /sy/test.js |
| SY\_PORTAL\_ICON | 系统使用的左上角LOGO图标 | /sy/comm/page/img/logo\_ruaho.gif |
| SY\_STYLE\_DEF | 系统默认风格定义 | {'SS\_STYLE\_MENU':'pageBody-lightBlue','SS\_STYLE\_BACK':'bodyBack-27','SS\_STYLE\_BLOCK':'conHeaderTitle-blue'}  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 | {'SS\_STYLE\_MENU':'pageBody-orange','SS\_STYLE\_BACK':'bodyBack-gray','SS\_STYLE\_BLOCK':'conHeaderTitle-orange'}  参考上面默认风格设置，  此配置值优先级>个人风格(优先级) >系统默认风格(优先级) |
| SY\_TAB\_COLOR | tab颜色 | {'1':{'li':'lightBlueLi','a':'lightBlueA'}}  一共5种，可选取其中几种作为指定的颜色：{'1':{'li':'qingLi','a':'qingA'},'2':{'li':'yellowLi','a':'yellowA'}, '3':{'li':'grayLi','a':'grayA'},'4':{'li':'blueLi','a':'blueA'},'5':{'li':'redLi','a':'redA'},'6':{'li':'lightLi','a':'lightA'},}; |
| 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 |
| SY\_MENU\_FLOAT | 右侧浮动菜单启用标志 | true  浮动菜单启用标志 true为启用，默认不启用 |

# 系统方法

注：红色参数为必填项

|  |  |  |
| --- | --- | --- |
| 方法 | 说明 | 参数说明 |
| FireFly  .byId(sId,id) | 获取某个服务的某条记录值 | sId:服务编码  id:记录主键 |
| FireFly.cardModify(sId,data) | 修改某条记录 | sId:服务编码  data:json格式的数据或字符格式的json字符串，如：  {“\_PK\_”:”test”,“USER\_CODE”:”test”,”USER\_NAME”:”testName”}，只构造需要修改的字段即可，必须参数。  其中\_PK\_为必填参数标识哪一条记录需要修改 |
| FireFly.cardAdd(sId,data) | 增加一条记录 | sId: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(sId,data,callback,tipFlag,async) | 批量保存多条记录 | sId: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(sId,data,tipFlag) | 删除记录 | sId: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(sId,datas) | 获取列表 | sId: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(sId,dataId,itemCode) | 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 | sId:服务编码  dataId:记录主键信息  itemCode:文件字段信息， |
| FireFly.getDict(dictCode,pid,extWhere,level,params) | 获取字典信息 | dictCode:字典编码  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,aId,data,tipFlag,async,func) | 执行某服务的某个方法和后台交互 | sId:服务编码  aId: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：  **var** params = {};  params["test"]="123";  params["hello"] = "world!";  **var**options=  {"url":"SY\_ORG\_DEPT.list.do","tTitle":"test","params":params,"menuFlag":3};  openNewTab(options); | 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.getDateTime(); //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  \* Format.(size, fix, value) ;//------------------  \* Format.limit(size, value);//--------------  \* Format.substr(start, end, value);//截串  \* Format.replaceDblQuotes(str);// 双引号替换为空串  \* Format.replaceSinQuotes(str);//// 单引号替换为空串  \* Format.RMBCapital(number);//人民币大写转换,numberValue 人民币小写 | | |
| Load | | |
| \* 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  \* Load.scriptJS(pathUrl);//动态加载js,放入骑到<script>标签，判断重复加载 | | |
| Tools | | |
| \* Tools.getFileSuffix(fileName);//取出文件扩展名  \* Tools.redirect(url);//重定位url  \* Tools.toMbIndex();//判断当前登录页面，如果为手机登录页则跳转  \* Tools.lTrim(val);//-----------  \* Tools.rTrim(val);//-----------  \* Tools.itemVarReplace(str, data);//系统字段级变量替换  \* Tools.systemVarReplace(str);//系统级变量替换  \* Tools.parVarReplace(str);//父层级级变量替换  \* Tools.rhSetBodyBack();//根据个性化的数据，设置背景的方法  \* Tools.xdocUrlReplace(url);//xdoc的url特殊处理替换  \* Tools.rhReplaceId(url);//替换菜单的info中的字符变量为id可用的字符等  \* Tools.rhReplaceHtmlTag(value);//html格式字符替换  \* Tools.getTreeLeafClass(dictId);//根据字典编号返回对应的叶子节点class | | |
| Browser | | |
| \* Browser.versions();//rh封装的判断浏览器类别,Browser.versions.iPad | | |
| Select | | |
| \* 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  \* 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 | | |
| Mouse | | |
| \* Mouse.getMousePoint(event);//获取鼠标在页面上的位置  \* 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  \* 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  \* Mouse.getScrollTop();//获取当前窗口滚动条高度  \* 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} | | |
| Todo | | |
| \* Todo.dbClickGrid(viewer,grid);//响应双击Grid事件，打开待办,viewer 包含grid的Viewer对象,grid 被双击的grid对象  \* Todo.open(sId, title, url, con, todoId, objectID1);//打开待办  \* 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  \* Todo.get(num,rowNum);//代办的获取，系统顶部下拉面板内数据的获取  \* Todo.getData(rowNum);//代办的获取，系统顶部下拉面板内数据的获取  \* 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 | | |
| Window | | |
| \* Window.getViewPort(target);//获取指定window的可视区域尺寸  \* Window.getScroll(target);//获取指定window的滚动条的信息  \* Window.searchScrollBegin();//搜索部分启用外部滚动条的监听事件 | | |
| RHWF | | |
| \* 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。 | | |
| 其他 | | |
| \* 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  \* Stopwatch();//辅助调试计时器 | | |

# 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\_ITEMS~','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，只在高级查询时起作用，高级查询里显示与否；  9、SEARCHTYPE：可选，只在高级查询起作用，高级查询里显示多选；  10、DATAFLAG:false,打开选择页面默认不显示列表数据；  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； |
| 树形选择 | SY\_ORG\_DEPT,{'TYPE':'multi’,’EXTWHERE’:’ and 1=1’,’extendDicSetting’:{},’rtnLeaf’:true, ‘SEARCHHIDE’:’true’,’SEARCHTYPE’:’multi’}  例：如上.. | 1、SY\_SERV\_ITEM：对应的字典ID;  2、TYPE:multi多选；  默认单选  3、EXTWHERE：过滤条件；  4、extendDicSetting：扩展的树形设置条件；  5、SEARCHHIDE:可选，只在高级查询时起作用，高级查询里显示与否；  6、SEARCHTYPE：可选，只在高级查询起作用，高级查询里显示多选；  7、rtnLeaf:只返回叶子节点；  8、LEVEL:设置当前树的数据获取层级；  9、PID:设置当前树从某个节点显示。  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}  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  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。 |
| 相关文件选择 | 例：  [{‘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'}] | servId：服务编码  SOURCE：显示的列  servName：显示名称  SHOWITEM:设置某个字段显示在提示块  TARGET\_SERV\_ID\_ITEM:被选择数据的服务ID |
| 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/portalComView.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<br>  <span style='color:red'>Helloworld!</span> | 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。  C:\Users\ljinkai\AppData\Roaming\Tencent\Users\277641333\QQ\WinTemp\RichOle\S}HN(U6_P@N{D84]KBTCXM1.jpg |
| 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。 |
| 评论 | 设置为【评论(自定义)】，需和自定义字段配合使用 | 对应自定义字段，排序最好放到最下面，显示在卡片的下方评论，包括已发表评论和输入评论功能。 |
| 附件(自定义) | | |
| 服务的字段控制     |  |  | | --- | --- | | 编码 | 即文件类型的分类，如 ZHENGWEN | | 名称 | 即页面显示的文件类型，如正文 | | 类型 | 需要定义为自定义字段 | | 输入类型 | 需要定义为附件(自定义) | | 输入设置 | 配置成类似  {'VALUE':'3','TYPES':'\*.jpg;\*.jpeg;\*.png;\*.gif;\*.doc;  \*.docx;\*.wps;\*.xls;\*.xlsx;\*.ppt;\*.pptx;\*.txt;',  'UPLOAD\_BTN\_NAME':'上传附件','FILENUMBER':'13'}  图片上传还支持额外的配置：IMG\_WIDTH和IMG\_HEIGHT，表示图片的宽度和  高度 | | VALUE 对应该类文件的权限值 (四种类型的和)  只读 1 ，可编辑 2，可上传 4，可删除 8，可修改文件名和排序 16，32是可下载  如果为 3，则可查看，可编辑。  为15 则查看，编辑，上传，删除的权限都有。  多个权限为多个权限数字的累加值  还支持字母权限：D,M,X,U,W,R  分别表示：下载，修改，删除，上传，编辑，查看，配置的时候把数字换成字母  即可，例如：dmxuwr，就是具有所有权限，不区分大小写，不用加分隔符。 | | TYPES 配置可上传文件的后缀类型  如\*.jpg;\*.jpeg;\*.png;\*.gif;\*.doc;\*.docx;\*.wps;\*.xls;\*.xlsx;\*.ppt;\*.pptx;\*.txt; | | UPLOAD\_BTN\_NAME页面上显示的上传文件的按钮的名称 | | FILENUMBER该类文件能上传的个数  REVISION是否记住痕迹 |   工作流节点上配置某个节点上文件的权限    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     |  |  | | --- | --- | | 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'}    意见自定义字段    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>

1. 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>

1. 按钮图标：

参考文件Icon.css。如：*btn-add*

1. 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

1. 左侧菜单图标，通过在菜单里设定相应的图标：

对应在文件common.css里，如：*.ui-state-default.leftMenu-bangong*

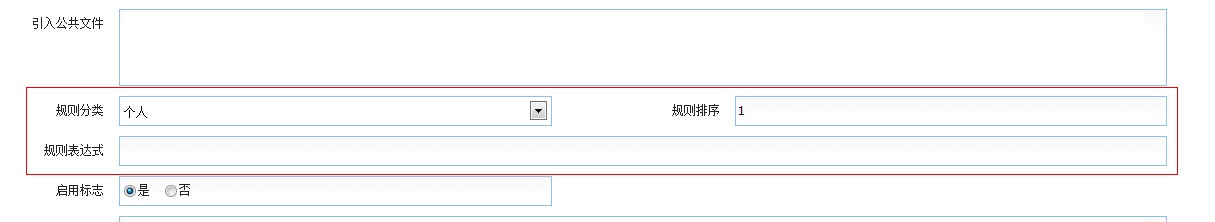
◆更多项目js示例，请参考/sy/servjs/demo.js示例文件。

# 其它配置说明

1 、门户权限配置说明

实现首页banner条上的门户权限配置。此些门户数据是根据门户模版里的配置而来。

其中：符合条件的 (某种规则分类 且 规则排序 且 符合规则表达式 的数据) 将会出现在首页banner上。



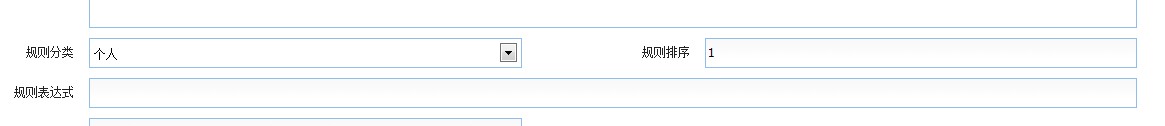
例如公司门户的配置为：



公司门户将出现在系统banner上，定义一个即可。

其它的部门门户类似。

个人门户有些不同的是，系统预定义一类默认的个人门户如：



此默认的个人门户在用户第一次进入时展示默认内容用，一旦做了修改，将系统自动生成一条与当前登录人相关的新记录，再次进入时将以新生成的模版展示，默认的个人门户将不再显示。

说明：像固定类的门户，如知识门户将通过系统配置的方式实现，不涉及权限的处理。所以banner上将会展示个人权限的门户+系统配置的门户。 暂时系统配置还未实现，下周一将实现此功能。

2 、 右侧浮动菜单配置说明

实现右侧浮动菜单上的快速导航配置。这些菜单数据是根据菜单里的配置而来。目前支持2级目录结构，如果配置的菜单项没有父节点而本身又不是节点，此菜单项将会显示在浮动菜单中，如果要显示其父菜单，那么其父菜单也要在菜单中进行配置，将其选为浮动图标。

其中：符合条件{"id":{"id":"","name":"","type":"","info","","dsname","","pid",{"id":"","name":"","type":"","info":"","dsname":"","pid":{...}}}} 将会出现在右侧浮动菜单上。

其中:

Id:菜单的id ; name:菜单名称 ; type:菜单类型 ; dsname:显示名称 ; pid :父菜单信息。

实现右侧浮动菜单上的快捷功能配置。这些菜单数据是根据模版组件权限和模版组件配置而来。如果用户想要此组件在快捷功能中显示，首先用户应该具有操作此组件的权限。其次管理员必须将此组件的显示位置设置为可以在浮动菜单中显示。例我分配的任务组件：



然后用户可以在快捷功能上点击设置进行设置想要出现的组件。

实现右侧浮动菜单上的设置皮肤配置：在菜单中对链接配置，第一个参数为floatMenu=rh\_skin 最后一个参数为pkCode,并且父菜单不能设置为浮动菜单上。样式弹出时的图片命名方式为class.png,如self\_blue.png.在配置服务时门户ID为pkCode，门户样式是一个字典,字典在配置的时候，字典的code为class如：self\_blue.

实现右侧浮动菜单上的设置布局配置: 在菜单中对链接配置，第一个参数为floatMenu=rh\_layout 最后一个参数为pkCode,并且父菜单不能设置为浮动菜单上。