目录

[1. JQ\_override.jsp说明 9](#_Toc418848364)

[方法介绍： 9](#_Toc418848365)

[window[CFG\_actionName(subffix)] (ui) 9](#_Toc418848366)

[\_override\_form(ui) 9](#_Toc418848367)

[\_override\_grid(ui) 10](#_Toc418848368)

[\_override\_tree(ui) 10](#_Toc418848369)

[\_override\_tbar(ui) 11](#_Toc418848370)

[2. 列表二次开发 11](#_Toc418848371)

[JS文件名称 11](#_Toc418848372)

[属性 11](#_Toc418848373)

[options 11](#_Toc418848374)

[timestamp 12](#_Toc418848375)

[selectedRowId 12](#_Toc418848376)

[gcfg 13](#_Toc418848377)

[scfg 13](#_Toc418848378)

[assembleData 13](#_Toc418848379)

[uiGrid 13](#_Toc418848380)

[uiIntgrt 13](#_Toc418848381)

[uiBase 13](#_Toc418848382)

[uiGreate 13](#_Toc418848383)

[uiSetting 13](#_Toc418848384)

[方法 14](#_Toc418848385)

[\_create() 14](#_Toc418848386)

[\_init() 14](#_Toc418848387)

[beforeInitGird(setting) 14](#_Toc418848388)

[eSelectRow(e, data) 14](#_Toc418848389)

[eDblClickRow(e, data) 14](#_Toc418848390)

[eOnSortableRows(e, data) 15](#_Toc418848391)

[eAfterSortableRows(e, data) 15](#_Toc418848392)

[component() 15](#_Toc418848393)

[refresh() 15](#_Toc418848394)

[reload() 15](#_Toc418848395)

[refreshPreviousGrid() 15](#_Toc418848396)

[reloadPreviousGrid() : 16](#_Toc418848397)

[getAction() : 16](#_Toc418848398)

[processFilter(filter) : 16](#_Toc418848399)

[addReturnButton(callback) 16](#_Toc418848400)

[getMasterId(type) 16](#_Toc418848401)

[getDetailTableIds() 16](#_Toc418848402)

[disableToolbar() 17](#_Toc418848403)

[enableToolbar() 17](#_Toc418848404)

[clickBackToGrid() 17](#_Toc418848405)

[clickCreate() 17](#_Toc418848406)

[clickUpdate() 17](#_Toc418848407)

[clickBatchUpdate() 17](#_Toc418848408)

[clickDelete() 18](#_Toc418848409)

[clickUpload() 18](#_Toc418848410)

[clickReport() 18](#_Toc418848411)

[clickCoflowTransact() 18](#_Toc418848412)

[clickCoflowTransactAndCheckout() 18](#_Toc418848413)

[clickCoflowCheckout() 18](#_Toc418848414)

[clickCoflowTrack() 19](#_Toc418848415)

[clickCoflowUntread() 19](#_Toc418848416)

[clickCoflowReassign() 19](#_Toc418848417)

[clickCoflowSuspend() 19](#_Toc418848418)

[clickCoflowTermination() 19](#_Toc418848419)

[addOutputValue(name, function(){}) 19](#_Toc418848420)

[addCallback(name, function(){}) 20](#_Toc418848421)

[clickSecondDev(id) 20](#_Toc418848422)

[rightTbarSecondDev(jq) 20](#_Toc418848423)

[后台控制器controller方法 20](#_Toc418848424)

[search() 20](#_Toc418848425)

[后台服务层service方法 20](#_Toc418848426)

[buildCustomerFilter(String tableId, String componentVersionId, String moduleId, String menuId, Map<String, String> paramMap) 20](#_Toc418848427)

[processPageData(List<Object> list) 21](#_Toc418848428)

[示例 21](#_Toc418848429)

[3. 表单二次开发 21](#_Toc418848430)

[JS文件名称 21](#_Toc418848431)

[属性 22](#_Toc418848432)

[options 22](#_Toc418848433)

[timestamp 22](#_Toc418848434)

[fcfg 22](#_Toc418848435)

[assembleData 23](#_Toc418848436)

[uiForm 23](#_Toc418848437)

[uiTbar 23](#_Toc418848438)

[uiSbar 23](#_Toc418848439)

[方法 23](#_Toc418848440)

[\_create() 23](#_Toc418848441)

[\_init() 23](#_Toc418848442)

[bindEvent() 23](#_Toc418848443)

[\_setDefaultValue() 24](#_Toc418848444)

[beforeSave(op) 24](#_Toc418848445)

[afterSave(entity, op) 24](#_Toc418848446)

[disableToolbarItem(id) 24](#_Toc418848447)

[enableToolbarItem(id) 24](#_Toc418848448)

[hideToobarItem(id) 24](#_Toc418848449)

[showToolbarItem(id) 25](#_Toc418848450)

[removeToolbarItem(id) 25](#_Toc418848451)

[getAction() 25](#_Toc418848452)

[load(dataId) 25](#_Toc418848453)

[reset() 25](#_Toc418848454)

[getMasterForm() 25](#_Toc418848455)

[getDetailForm() 26](#_Toc418848456)

[getSelfGrid() 26](#_Toc418848457)

[getMasterGrid() 26](#_Toc418848458)

[getDetailGrid() 26](#_Toc418848459)

[reloadSelfGrid() 26](#_Toc418848460)

[getFormData() 26](#_Toc418848461)

[setFormData(data) 27](#_Toc418848462)

[getItemId(columnName) 27](#_Toc418848463)

[getItemJQ(columnName) 27](#_Toc418848464)

[getItemValue(columnName) 27](#_Toc418848465)

[setItemValue(columnName) 27](#_Toc418848466)

[clickCreate() 27](#_Toc418848467)

[clickSave(op) 28](#_Toc418848468)

[clickAdd() 28](#_Toc418848469)

[clickSaveAll() 28](#_Toc418848470)

[clickReset() 28](#_Toc418848471)

[clickViewRecord(op) 28](#_Toc418848472)

[clickCoflowStart() 28](#_Toc418848473)

[clickCoflowStartAndComplete() 29](#_Toc418848474)

[clickCoflowComplete() 29](#_Toc418848475)

[clickCoflowCheckout() 29](#_Toc418848476)

[clickCoflowTrack() 29](#_Toc418848477)

[clickCoflowUntread() 29](#_Toc418848478)

[clickCoflowRecall() 29](#_Toc418848479)

[clickCoflowDeliver() 30](#_Toc418848480)

[clickCoflowHasread() 30](#_Toc418848481)

[clickBackToGrid() 30](#_Toc418848482)

[clickSecondDev(id) 30](#_Toc418848483)

[addCallback(name, function) 30](#_Toc418848484)

[addOutputValue(name, function) 30](#_Toc418848485)

[控制层controller方法 31](#_Toc418848486)

[save() 31](#_Toc418848487)

[saveAll() 31](#_Toc418848488)

[服务层service方法 31](#_Toc418848489)

[processBeforeSave(String tableName, Map<String, String> dataMap, Map<String, Object> paramMap) 31](#_Toc418848490)

[processAfterSave(String tableName, Map<String, String> dataMap, Map<String, Object> paramMap) 31](#_Toc418848491)

[processBeforeSaveOneDetail(String tableName, Map<String, String> dataMap, Map<String, String> masterDataMap, Map<String, Object> paramMap) 32](#_Toc418848492)

[processAfterSaveOneDetail(String tableName, Map<String, String> dataMap, Map<String, String> masterDataMap, Map<String, Object> paramMap) 32](#_Toc418848493)

[processMiddleSaveAll(String tableName, Map<String, String> dataMap, List<Map<String, String> > detailList, Map<String, Object> paramMap) 32](#_Toc418848494)

[processAfterSaveAll(String tableName, Map<String, String> dataMap, List<Map<String, String> > detailList, Map<String, Object> paramMap) 33](#_Toc418848495)

[示例 33](#_Toc418848496)

[4. 树二次开发 34](#_Toc418848497)

[JS文件名 34](#_Toc418848498)

[属性 34](#_Toc418848499)

[options 34](#_Toc418848500)

[timestamp 34](#_Toc418848501)

[zNodes 35](#_Toc418848502)

[uiTree 35](#_Toc418848503)

[方法介绍 35](#_Toc418848504)

[\_create() 35](#_Toc418848505)

[\_init() 35](#_Toc418848506)

[beforeInitTree(setting) 35](#_Toc418848507)

[treeClick (e, treeId, treeNode) 35](#_Toc418848508)

[\_initHelp(e, treeId, treeNode) 36](#_Toc418848509)

[toDefaultComponent(e, treeId, treeNode) 36](#_Toc418848510)

[控制器controller方法 36](#_Toc418848511)

[processTree () 36](#_Toc418848512)

[processTreeData(List<TreeDefine> list, Map<String, Object> paramMap) 36](#_Toc418848513)

[processNodeAuthority (List<TreeDefine> list, Map<String, Object> paramMap) 37](#_Toc418848514)

[示例 37](#_Toc418848515)

[5. 工具条二次开发 38](#_Toc418848516)

[JS文件名 38](#_Toc418848517)

[属性 38](#_Toc418848518)

[方法介绍 38](#_Toc418848519)

[示例 38](#_Toc418848520)

# JQ\_override.jsp说明

该文件是系统配置平台自定义构件二次开发JS规范模板，这个jsp里面的结构及方法不可随意删除，如果需要复写，请在指定方法中复写即可。

## **方法介绍：**

### window[CFG\_actionName(subffix)] (ui)

参数：ui 为对应区域列表或表单对象实例

说明：指定这个自定义构件对应的controller，构件中的所有区域都通过他来获取controller

### \_override\_form(ui)

参数：ui 为jquery.config.cform 对象实例

说明：复写自定义构件中的表单入口方法，在这个方法中可以复写jquery.config.cform类中的所有属性与方法

示例：如果该构件中有多个区域，复写时，请用ui.options.number来判断，到底要复写哪个区域上的表单。第一个区域ui.options.number=1，第二个区域ui.options.number=2，以此类推。比如，一个构件为上下结构的，要复写第一个区域的表单：

**示例代码**

**if** (1 === ui.options.number) { // 复写第一个区域的表单

// 给options.tableId 重新赋值

ui.options.tableId = "xxxx";

// 给表单上的元素绑定事件

ui.bindEvent = **function** () {

// 给USER\_NAME绑定一个onChange事件

ui.setItemOption("USER\_NAME", "onChange", **function**(e, data) {

CFG\_message(data.oldValue);

});

// 级联事件

ui.setItemOption ("option", "onChange", **function**(e, data) {

**var** text = ui.callItemMethod("USER\_CODE", "getText");

// 给USER\_NAME栏位设置

ui.setItemValue("USER\_NAME", text);

});

};

} **else** **if** (2 === ui.options.number) { // 复写第二个区域的表单

}

### \_override\_grid(ui)

参数：ui 为jquery.config.cgrid 对象实例

说明：复写自定义构件中的列表入口方法，在这个方法中可以复写jquery.config.cgrid类中的所有属性与方法

示例：如果该构件中有多个区域，复写时，请用ui.options.number来判断，到底要复写哪个区域上的表单。第一个区域ui.options.number=1，第二个区域ui.options.number=2，以此类推。

**示例代码**

**if** (1 === ui.options.number) { // 复写第一个区域的列表

// 给列表添加过滤条件 (IS\_DELETE=1)

ui.options.columns = "EQ" + CFG\_oc\_split + "IS\_DELETE" + CFG\_cv\_split + "1";

// 复写删除事件

ui.clickDelete = **function** () {

CFG\_message("删除！");

};

// 复写列表单击事件

ui.eSelectRow = **function** () {

};

} **else** **if** (2 === ui.options.number) {// 复写第二个区域的列表

}

### \_override\_tree(ui)

参数：ui 为jquery.config.ctree 对象实例

说明：复写自定义构件中的树结构入口方法

示例：树单击事件复写

**示例代码**

ui.treeClick = **function** (e, treeId, treeNode) {

// 只要是表节点都跳转对应组装的物理表构件

**if** (isNotEmpty(treeNode.tableId)) {

//列表过滤条件(PARRENT\_ID=xxx)

**var** params = "P\_columns=EQ" + CFG\_oc\_split + "PARENT\_ID" + CFG\_cv\_split + treeNode.value;

CFG\_clickTreeNode(ui.assembleData, "2", treeNode.tableId, treeNode.name, params, ui);

}

};

### \_override\_tbar(ui)

参数：ui 为jquery.config.ctbar 对象实例

说明：复写自定义构件中的工具条入口方法，注：复写时，需用ui.options.number来指定复写哪个区域的工具条，同时还需要用ui.options.type来判断，复写的是列表上的工具条还是表单上的工具条；同时，按钮ID的前缀必须是$.config.preButtonId常量值打头的，按钮的事件，请对应的\_override\_grid或\_override\_form方法中实现ui.clickSecondDev方法

建议：目前，工具条二次开发，可以在对应的列表或表单中实现：实现ui.options.toolbar=function(data, pos){return data;}这个接口即可

示例：给工具条添加按钮

**示例代码**

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* $.config.contentType.grid：表示列表

\* $.config.contentType.form：表示表单

\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**if** (1 === ui.options.number &&

$.config.contentType.grid === ui.options.type) {

ui.processItems = **function** (data, pos) {

data.push({"id": $.config.preButtonId + "view", "label": "浏览"})

**return** data;

};

}

# 列表二次开发

## JS文件名称

Jquery.config.cgrid.js

## 属性

### options

说明：初始化列表所需要传入参数

{

menuId : **null**, // 菜单ID

componentVersionId : **null**, // 自定义构件对应的组合构件ID

moduleId: **null**, // 自定义构件ID

tableId : **null**, // 列表对应的表ID

workflowId : **null**, // 工作流ID

box : **null**, // 工作箱

number: **null**,

model : 1, // 1-普通列表 2-缩略图列表

thumbnail : 0, // 0-页面打开时默认显示列表 1-页面打开时默认显示缩略图

sort : 0, // 0-不支持拖动排序 1-支持拖动排序

isAlone : **false**,// 是否独立使用：true-独立使用与主列表无关系，false-不是独立使用，如果有主列表，则主列表有关系

master : **null**, // {tableId: "xx", dataId: "yy"}

// 1. boolean: false则不显示工具条，否则显示

// 2. function(data) {return data;} 其中data为工具条toolbar的data

// 在这个方法中可以处理工具条中的按钮（添加、删除、禁用等）

toolbar : **null**,

global : **null**, // 每个区域共同拥有的一个全局最大DIV元素对象

// 列表过滤条件格式，如：EQ\_C\_NAME≡**XXX**;LIKE\_C\_TITLE≡**XXX**；多个用英文分号(;)分隔

// 三等于（≡）常量： CFG\_cv\_split； \_C\_常量：CFG\_oc\_split

// 复杂条件：如 (name like '%qiu%' or birth\_day=1983) and height > 168)

// 则格式为 (((LIKE\_C\_NAME≡1)OR(EQ\_C\_BIRTH\_DAY≡2012))AND(GT\_C\_HEIGHT≡168)) 注：每一项需要用括号括起来

columns : **null**,

// 过滤条件 function(filter) {...;return filter;}

processFilter : **null**

}

### globalTimestamp

说明：渲染构件时，生成了时间戳（通过该时间戳在后台可以从session中获取这个列表的过滤条件和排序方式：CommonUtil.getQueryFilter/CommonUtil.getQuerySort）

### timestamp

说明：渲染构件列表时，生成了时间戳

### selectedRowId

说明：选中一条记录ID，选中多条则为null

### gcfg

说明：列表的配置信息（JSON结构请查看GridPageModel.java）；通过gcfg.linkIndex > -1可以判断，该列表是否有配置“操作列”（gcfg.linkIndex的值就是所在列表options中所在索引值）

### assembleData

说明：构件组装数据（相当于在开发构件中的configInfo）

### uiGrid

说明：组件库列表所在的DIV Jquery对象

## 方法

### \_create()

参数：无

说明：创建列表构件

### \_init()

参数：无

说明：\_create方法执行后，初始化列表二次开发接口

### beforeInitGird(setting)

参数：setting 为组件库列表的options

说明：重构组件库列表setting二次开发接口，比如要添加列表列或是添加列表事件

### eSelectRow(e, data)

参数：e 为event事件

data 为单击行相关数据

说明：列表单击事件二次开发接口

### eDblClickRow(e, data)

参数：e 为event事件

data 为双击行相关数据

说明：列表双击事件二次开发接口

### eOnSortableRows(e, data)

参数：e 为event事件

data 为拖拽行相关数据

说明：列表拖动事件二次开发接口

### eAfterSortableRows(e, data)

参数：e 为event事件

data 为单击行相关数据

说明：结束列表拖动事件二次开发接口

### component()

参数：无

说明：获取自定义列表构件所在DIV Jquery对象

### refresh()

参数：无

说明：刷新自定义列表构件列表（重新渲染）

### load()

参数：无

说明：加载当前列表数据（回到第一页）

### reload()

参数：无

说明：重新加载当前页列表有数据

### refreshPreviousGrid()

参数：无

说明：刷新上一层自定义列表构件列表（前提是存在上一层自定义列表构件列表）

### reloadPreviousGrid() :

参数：无

说明：重新加载上一层自定义列表构件列表（前提是存在上一层自定义列表构件列表）

### getAction() :

参数：无

说明：获取当前列表对应controller

### processFilter(filter) :

参数：filter 为列表查询的过滤条件

说明：列表过滤条件处理二次开发接口

### addReturnButton(callback)

参数：callback 为回调方法

说明：向工具条上添加“返回”按钮，参数callback为回调方法

### getMasterId(type)

参数：type 为可选参数

说明：当type="DATA"时，获取的是对应主列表选中记录ID，否则为主列表对应的表ID

### getDetailTableIds()

参数：无

说明：获取所有关联从列表表ID，多个用英文“,”分隔

### disableToolbar()

参数：无

说明：禁用工具条

### enableToolbar()

参数：无

说明：启用工具条

### clickBackToGrid()

参数：无

说明：返回或关闭按钮事件

### clickCreate()

参数：无

说明：新增按钮事件

### clickUpdate()

参数：无

说明：修改按钮事件

### clickBatchUpdate()

参数：无

说明：批量修改按钮事件

### clickDelete()

参数：无

说明：删除按钮事件

### clickUpload()

参数：无

说明：电子文件上传按钮事件

### clickReport()

参数：无

说明：打印按钮事件

### clickCoflowTransact()

参数：无

说明：工作流办理按钮事件

### clickCoflowTransactAndCheckout()

参数：无

说明：工作流办理并签收按钮事件

### clickCoflowCheckout()

参数：无

说明：工作流签收按钮事件

### clickCoflowTrack()

参数：无

说明：工作流跟踪按钮事件

### clickCoflowUntread()

参数：无

说明：工作流退回按钮事件

### clickCoflowReassign()

参数：无

说明：工作流转办按钮事件

### clickCoflowSuspend()

参数：无

说明：工作流中止（删除）按钮事件

### clickCoflowTermination()

参数：无

说明：工作流终止按钮事件

### addOutputValue(name, function(){})

参数：name 为构件输出参数的方法名称

function 为构件输出参数的方法函数体

说明：添加件输出参数方法，用于构件组装时，绑定构件入参使用

### addCallback(name, function(){})

参数：name 为回调方法名称

function 为回调方法函数体

说明：添加回调方法，用于构件组装时，绑定回调方法使用

### clickSecondDev(id)

参数：id 为工具条上的按钮id

说明：工具条上的按钮事件二次开发接口，注意手动添加的工具条按钮ID必须是以 $.config.preButtonId打头的按钮才能进入这个方法

### getMasterForm()

参数：无

说明：获取当前列表对应主表的表单

### getSelfForm()

参数：无

说明：获取当前列表对应的表单

### getSelfBaseSearch()

参数：无

说明：获取当前列表对应的查询区（布局四中的查询区，非基本查询区）

### getMasterGrid()

参数：无

说明：获取当前列表对应主表的列表

### getDetailGrid()

参数：无

说明：获取当前列表对应从表的列表

### getSearchComboValue()

参数：无

说明：获取下拉检索框中的值

### disableToolbar()

参数：无

说明：禁用列表上的工具条

### enableToolbar()

参数：无

说明：启用列表上的工具条

### disableToolbarItem(id)

参数：id 工具条上的按钮id

说明：禁用工具条上指定id按钮

### enableToolbarItem(id)

参数：id – 工具条上的按钮id

说明：启用工具条上指定id按钮

### hideToolbarItem(id)

参数：id 工具条上按钮id

说明：隐藏工具条指定ID按钮

### showToolbarItem(id)

参数：id 工具条上按钮id

说明：显示工具条指定ID按钮

### removeToolbarItem(id)

参数：id 工具条上按钮ID

说明：移除工具条上指定ID按钮

### addRowData(data)

参数：data 为一行的列表数据

说明：如果为新数据，则向列表添加一行数据；如果是老数据，则更新列表中数据

### getRowData(rowId)

参数：rowId 为行ID（可选参数）

说明：如果rowId存在时，则根据指定rowId获取列表数据；如果不传入rowId时，则获取整个列表数据

### validGridData(rowId)

参数：rowId 为列表行ID（可选参数）

说明：校验列表中的可编辑数据；如果传入rowId则校验指定行的数据，否则校验整个列表数据

### setGridOption(key, value)

参数：key 为列表options中的属性

value 为属性key对应的值

说明：设置组件库grid中options属性值；如: ui.setGridOption(“url”, $.contextPath + "/action!method.json?P\_tableId=xxx")

### getGridOption (key)

参数：key 为列表options中的属性值

说明：获取组件库grid中options指定属性key的值；如：var url = ui.getGridOption(“url”);

### callGridMethod(method, args)

参数：method 为组件库grid中方法

args为方法(method)对应的参数列表（可选参数），如果method没有参数，则不需要传入

说明：调用组件库grid的方法；如ui.callGridMethod(“getDataIDs”)/ui.callGridMethod(“getRowData”, rowId)

### getParamValue(key)

参数：key 为参数名，如果参数名为”P\_”打头的，如”P\_rowId”，则可以直接用rowId

说明：获取URL中指定的参数值

### beforeDelete()

参数：无

说明：删除前二次开发接口；注：此方法需要有返回值（boolean），true表示继续执行/false表示终止执行

### afterDelete()

参数：无

说明：删除后二次开发接口

### beforeClickTbar(id)

参数：id 为工具条上的按钮ID

说明：单击工具条上的按钮前触发的事件的二次开发接口；注：此方法需要有返回值（boolean），true 表示继续执行/false表示终止执行

### beforeClickLink(id, rowId)

参数：id 为超链接ID

rowId 为行ID

说明：单击操作列中的超链接前触发的事件的二次开发接口；注：此方法需要有返回值（boolean），true表示继续执行/false表示终止执行

## 后台控制器controller方法

### search()

参数：无

说明：列表查询入口方法

## 后台服务层service方法

### buildCustomerFilter(String tableId, String componentVersionId, String moduleId, String menuId, Map<String, String> paramMap)

参数：tableId 为表ID

componentVersionId 为组合构件ID

moduleId 为自定义构件ID

menuId 为菜单ID

paramMap 为参数集合（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：列表过滤条件二次开发接口，注：返回过滤条件时，要以AppDefineUtil.RELATION\_AND、AppDefineUtil.RELATION\_OR开头，如return AppDefineUtil.RELATION\_AND + "DELETE\_FLAG=0";

### processPageData(List<Object> list)

参数：list 为查询结果集

说明：处理查询结果集的二次开发接口，比如某列加粗等

## 示例

添加列表过滤条件（当构件内有多个列表时， 在前后台得注意判断是要给哪个列表添加过滤条件，在前台可以用ui.options.number来判断，有后可用表名来判断TableUtil.getTableName(tableId)来判断）

方法一：在后台service实现，复写service中的buildCustomerFilter方法

方法二：在前台实现，在对应JQ\_override.jsp中找到 function \_override\_gird函数，直接给ui.options.columns设置自己过滤条件，过滤条件格式见属性介绍。

# 表单二次开发

## JS文件名称

Jquery.config.cgrid.js

## 属性

### options

说明：初始化列表所需要传入参数

{

menuId : **null**, // 菜单ID

componentVersionId : **null**, // 自定义构件对应的组合构件ID

moduleId: **null**, // 自定义构件ID

tableId : **null**, // 列表对应的表ID

workflowId : **null**, // 工作流ID

box : **null**, // 工作箱

number: **1**,

columns : **null**, // 目前格式与cgrid中columns格式一致，同时，如果是EQ或LIKE的条件时，这些条件将作为表单的默认值（前提：表单存在这些字段）

isNested : **true**, // 是否为嵌入式表单(如果为false，则根据表单自身设置为显示)

gDivId : **null**, // 组件库列表DIV ID，如果存在，返回时会重新加载列表数据

dataId : **null**, // 数据ID(修改/查看记录)

isView : **false**,// 是否查看

isAlone : **false**,// 是否独立使用：true-独立使用与主列表无关系，false-不是独立使用，如果有主列表，则主列表有关系

master : **null**, // 关联主表信息{tableId: "xx", dataId: "yy"}

// 1. boolean: false则不显示工具条，否则显示

// 2. function(data) {return data;} 其中data为工具条toolbar的data

// 在这个方法中可以处理工具条中的按钮（添加、删除、禁用等）

toolbar : **null**,

global : **null** // 每个区域共同拥有的一个全局最大DIV元素对象

}

### timestamp

说明：时间戳（所有表单元素的ID都是由 字段名+ “\_” + timestamp 组成的， 通过 $(“#” + 字段名 + “\_” + timestamp) 方法可以获取元素Jquery对象，建议用getItemJQ(columnName)方法来获取）

### fcfg

说明：表单配置信息（JSON结构请查看FormPageModel.java）

### assembleData

说明：构件组装数据（相当于在开发构件中的configInfo）

### uiForm

说明：组件库表单所在的DIV Jquery对象

### uiTbar

说明：表单工具条（顶部）的DIV Jquery对象

### uiSbar

说明：表单工具条（底部）的DIV Jquery对象（只有嵌入式表单中才有）

## 方法

### \_create()

参数：无

说明：创建表单构件

### \_init()

参数：无

说明：\_create方法执行后，初始化表单二次开发接口

### bindEvent()

参数：无

说明：绑定表单事件入口

### \_setDefaultValue()

参数：无

说明：设置表单默认值

### beforeSave(op)

参数：op为操作：save—保存， create—保存并新增，close—保存并关闭或保存并返回

说明：保存前回调方法二次开接口，如果return false; 则终止保存

### afterSave(entity, op)

参数：entity 为表单数据（json格式）

op 为操作：save—保存， create—保存并新增，close—保存并关闭或保存并返回

说明：保存后回调方法二次开发接口

### disableToolbarItem(id)

参数：id 为按钮ID

说明：禁用工具条上的按钮（如果顶部工具条与底部工具条都有，则同步禁用）

### enableToolbarItem(id)

参数：id 为按钮ID

说明：启用工具条上的按钮（如果顶部工具条与底部工具条都有，则同步启用）

### hideToobarItem(id)

参数：id 为按钮ID

说明：隐藏工具条上的按钮（如果顶部工具条与底部工具条都有，则同步隐藏）

### showToolbarItem(id)

参数：id 为按钮ID

说明：显示工具上的按钮（如果顶部工具条与底部工具条都有，则同步显示）

### removeToolbarItem(id)

参数：id 为按钮ID

说明：移除工具上的按钮（如果顶部工具条与底部工具条都有，则同步移除）

### getAction()

参数：无

说明：获取controller

### load(dataId)

参数：dataId 为表单数据ID

说明：加载表单数据

### reset()

参数：无

说明：重置表单数据

### getMasterForm()

参数：无

说明：获取当前表单对应的主表单（cform）对象

### getDetailForm()

参数：无

说明：获取当前表单对应的从表单（cform）对象

### getSelfGrid()

参数：无

说明：获取当前表单对应的列表（cgrid）对象

### getMasterGrid()

参数：无

说明：获取当前表单对应的主列表（cgrid）对象

### getDetailGrid()

参数：无

说明：获取当前表单对应的从列表（cgrid）对象

### reloadSelfGrid()

参数：无

说明：重新加载当前表单对应的列表数据

### getFormData()

参数：无

说明：获取当前表单数据

### setFormData(data)

参数：data 为表单json数据

说明：给表单设值

### getItemId(columnName)

参数：columnName 为字段名称

说明：获取该字段元素ID

### getItemJQ(columnName)

参数：columnName 为字段名称

说明：获取该字段元素对应的Jquery对象

### getItemBorderJQ(columnName)

参数：columnName 为字段名称

说明：获取该字段元素外围对应的DIV的Jquery对象（包括对应的label），比如要隐藏这个字段元素就可以用这个方法：ui.getItemBorderJQ(“NAME”).hide();

### getItemLabel(columnName)

参数：columnName 为字段名称

说明：获取该字段元素对应的label

### setItemLabel(columnName, label)

参数：columnName 为字段名称

label 为字段对应的label

说明：设置字段元素对应的label

### getItemValue(columnName)

参数：columnName 为字段名称

说明：获取该字段元素的值

### setItemValue(columnName, value)

参数：columnName 为字段名称

value 为字段对应的值

说明：给字段元素赋值

### callItemMethod(columnName, method, args)

参数：columnName 为字段名称

method为方法名称

args为方法(method)对应的参数列表（可选参数），如果method没有参数，则不需要传入

说明：调用columnName对应的组件方法，如调用enable/setValue方法：ui.callItemMethod(“NAME”, “enable”)/ui.setItemMethod(“NAME”, “setValue”, “qiucs”);

### getItemOption(columnName, key)

参数：columnName 为字段名称

key为option中的属性key

说明：获取该字段元素对应组件中options属性key值，如ui.getItemOption(“NAME”, “required”)

### setItemOption(columnName, key, value)

参数：columnName 为字段名称

key为options中的属性key

value为options中的属性key对应的值

说明：设置该字段元素对应组件中options属性key值，如添加onChange事件：ui.setItemOption(“NAME”, “onChange”, function(e, data) {alert(“onchange”);})

### clickCreate()

参数：无

说明：新增按钮

### clickSave(op)

参数：op为操作：save—保存， create—保存并新增，close—保存并关闭或保存并返回

说明：保存数据

### clickAdd()

参数：无

说明：添加，将表单数据添加到对应列表中（前台操作）

### clickSaveAll()

参数：无

说明：保存，将自身表单数据（主表）及对应的从列表数据集（从表）一次性提交到后台级联保存

### clickReset()

参数：无

说明：重置表单

### clickViewRecord(op)

参数：op 为操作：first—首条，next—下一条，previous—上一条，last—末条

说明：查看（首条，上一条，下一条，末条）

### clickCoflowStart()

参数：无

说明：启动（工作流）

### clickCoflowStartAndComplete()

参数：无

说明：启动并提交（工作流）

### clickCoflowComplete()

参数：无

说明：提交（工作流）

### clickCoflowCheckout()

参数：无

说明：工作流签收按钮事件

### clickCoflowTrack()

参数：无

说明：工作流跟踪按钮事件

### clickCoflowUntread()

参数：无

说明：工作流退回按钮事件

### clickCoflowRecall()

参数：无

说明：撤回（工作流）

### clickCoflowDeliver()

参数：无

说明：传阅（工作流）

### clickCoflowHasread()

参数：无

说明：阅毕（工作流）

### clickBackToGrid()

参数：无

说明：返回

### clickSecondDev(id)

参数：id 为工具条上按钮ID

说明：工具条二次开发按钮事件入口，注意手动添加的工具条按钮ID必须是以 $.config.preButtonId打头的按钮才能进入这个方法

### addCallback(name, function)

参数：name 为回调方法名

function 为回调方法函数体

说明：二次开发，添加回调方法入口

### addOutputValue(name, function)

参数：name 为输出参数方法名称

function 为输出参数方法体

说明：二次开发，添加输出参数方法入口

## 控制层controller方法

### save()

说明：表单保存入口

### saveAll()

说明：表单级联保存入口（保存自身表单及对应从列表数据集）

## 服务层service方法

### processBeforeSave(String tableName, Map<String, String> dataMap, Map<String, Object> paramMap)

参数：tableName 为表名称

dataMap 为表单数据（key—字段名，value—值）

paramMap 为参数集（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：表单数据保存前业务逻辑处理二次开发接口，如果要终止保存，请抛出BusinessException来终止执行，这样在前台可以捕获该异常，提示出来

### processAfterSave(String tableName, Map<String, String> dataMap, Map<String, Object> paramMap)

参数：tableName 为表名称

dataMap 为表单数据（key—字段名，value—值）

paramMap 为参数集（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：表单数据保存后业务逻辑处理二次开发接口

### processBeforeSaveOneDetail(String tableName, Map<String, String> dataMap, Map<String, String> masterDataMap, Map<String, Object> paramMap)

参数：tableName 为表名称

dataMap 为单条明细表单数据（key—字段名，value—值）

masterDataMap 为主表表单数据（key—字段名，value—值）

paramMap 为参数集（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：级联保存时，保存单条明细表表单数据保存前业务逻辑处理二次开发接口，如果要终止保存，请抛出BusinessException来终止执行，这样在前台可以捕获该异常，提示出来

### processAfterSaveOneDetail(String tableName, Map<String, String> dataMap, Map<String, String> masterDataMap, Map<String, Object> paramMap)

参数：tableName 为表名称

dataMap 为单条明细表单数据（key—字段名，value—值）

masterDataMap 为主表表单数据（key—字段名，value—值）

paramMap 为参数集（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：级联保存时，保存单条明细表表单数据保存后业务逻辑处理二次开发接口

### processMiddleSaveAll(String tableName, Map<String, String> dataMap, List<Map<String, String> > detailList, Map<String, Object> paramMap)

参数：tableName 为表名称

dataMap 为主表表单数据（key—字段名，value—值）

detailList 为明细表表单数据集

paramMap 为参数集（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：级联保存时，在保存明细后与保存主表数据之间，进行业务逻辑处理的二次开发接口

### processAfterSaveAll(String tableName, Map<String, String> dataMap, List<Map<String, String> > detailList, Map<String, Object> paramMap)

参数：tableName 为表名称

dataMap 为主表表单数据（key—字段名，value—值）

detailList 为明细表表单数据集

paramMap 为参数集（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：级联保存时，在保存明细后与保存主表数据之后，进行业务逻辑处理的二次开发接口

## 示例

1. 给表单元素绑定事件？

请在\_overried\_form方法复写ui.bindEvent，如下所示：

ui.bindEvent = **function** () {

// 给USER\_NAME绑定一个onChange事件

**var** jqUserName = ui.getItemJQ("USER\_NAME"/\*字段名称\*/);

**if** (jqUserName.length) {

jqUserName.textbox("option", "onChange", **function**(e, data) {

CFG\_message(data.oldValue);

});

}

};

2. 保存前校验？

前台校验：请在\_overried\_form方法复写ui.beforeSave，如下所示：

ui.beforeSave = **function** () {

// 给USER\_NAME绑定一个onChange事件

**var** userName = ui.getItemValue("USER\_NAME"/\*字段名称\*/);

**if** (isEmpty(userName)) {

CFG\_message("用户名不能为空！", "warning");

**return** **false**;

}

**return** **true**;

};

后台校验：请在服务层中对应的接口实现即可，如 processBeforeSave

3. 注意事件：表单中的单选框与复选框，请使用jq.radiolist和jq.checkboxlist来操作

# 树二次开发

## JS文件名

Jquery.config.ctree.js

## 属性

### options

说明：初始化列表所需要传入参数

{

menuId : **null**, // 菜单ID

componentVersionId : **null**, // 组装构件ID

moduleId : **null**, // 自定义构件ID

info : **null**, // 模块配置信息

rightPanel : **null**, // 右边区域DIV jquery对象

global : **null** //每个区域共同拥有的一个全局最大DIV元素对象

}

### timestamp

说明：时间戳

### zNodes

说明：根节点

### uiTree

说明：组件库树所在的DIV Jquery对象

## 方法介绍

### \_create()

参数：无

说明：创建树构件入口

### \_init()

参数：无

说明：初始化树构件，在\_create方法之后执行

### beforeInitTree(setting)

参数：setting – 组件库树options

说明：重构组件库树setting二次开发接口（可以改变树setting属性）

### treeClick (e, treeId, treeNode)

参数：e 为event事件

treeId 为树ID，使用$(“#”+treeId).tree(“method”)

treeNode 为树节点对象，使用：treeNode.id获取节点ID

说明：树节点单击事件入口，如果树节点单击事件有特殊需求，可以复写此接口

### \_initHelp(e, treeId, treeNode)

参数：e 为event事件

treeId 为树ID，使用$(“#”+treeId).tree(“method”)

treeNode 为树节点对象，使用：treeNode.id获取节点ID

说明：树右侧帮助信息页面入口

### toDefaultComponent(e, treeId, treeNode)

参数：e 为event事件

treeId 为树ID，使用$(“#”+treeId).tree(“method”)

treeNode 为树节点对象，使用：treeNode.id获取节点ID

说明：如果树节点没有绑定构件时，默认打开的页面；目前只有工作流节点用到此方法。如果树节点对应的右侧构件完全二次开发，则可以利用此接口来实现

## 控制器controller方法

### processTree ()

参数：无

说明：查找树节点的入口方法

### processTreeData(List<TreeDefine> list, Map<String, Object> paramMap)

参数：list 为树节点集合

paramMap 为参数集合（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：过滤树节点处理（处理业务相关的过滤，与人员或权限无关的）

### processNodeAuthority (List<TreeDefine> list, Map<String, Object> paramMap)

参数：list 为树节点集合

paramMap 为参数集合（以”P\_”打头的参数，并非所有参数，但Map中的key值不包含”P\_”）

说明：树节点权限处理（处理权限管理中授予树节点权限）

## 示例

当树结构在树定义中配置不出来时，同时还想用自定义树构件，这时就可以复写processTree方法来构造自己的业务树。比如组织树，步骤如下：

1. 在“树定义”中定义一个树节点（机构组织）
2. 在“构件定义”中定义一个树构件，选择“机构组织”树
3. 在生成的构件对应的controller中复写processTree方法，比如通过id的前缀来判断，如果id为32位的UUID则表示机构组织树的根节点，这时在组织表中查找parentId为-1的第一层节点并把结果集构造成树型结构，同时处理一下id，比如id添加一个前缀（如ORG\_ + id），这样就可以自己构造一棵树型构件的
4. 如果树型构件要绑定构件（比如机构组织物理表构件）的话，则在组装中绑定这个物理表构件
5. 如果要给“机构组织物理表构件”传递一些条件，则需要在JQ\_override.jsp中的 \_override\_tree方法中复写ui.treeClick方法，如下所示：

ui.treeClick = **function** (e, treeId, treeNode) {

**if** (**typeof** CFG\_clearComponentZone === "function") {

CFG\_clearComponentZone(**this**.assembleData);

}

**var** params, nodeArea, id;

id = treeNode.id

**if** (id.substring(0, 4) === "ORG\_") {

treeNode.type = "2"; // 表节点

// 传入参数

params = "P\_columns=EQ" + CFG\_oc\_split +"PARNET\_ID" + CFG\_cv\_split + id.substring(4);

CFG\_clickTreeNode(ui.assembleData, treeNode.type, treeNode.tableId, treeNode.name, params, ui);

} **else** {

ui.\_initHelp(e, treeId, treeNode);

}

};

# 工具条二次开发

## JS文件名

jquery.config.tbar.js

## 属性

## 方法介绍

## 示例