# AR.DataGrid 文档

## 用法：

<body>

<table id="dg"></table>

</body>

</html>

<script type="text/javascript">

var dg = new AR.DataGrid("Sys\_Role");

dg.isEditor = true;

dg.PKColumn.add("detail", "权限分配", 'roleAction', null, "权限分配", "edit");

dg.bind();

function roleAction(v, r, i) {

var rname = r.rolename;

AR.Utility.Window.open('RoleAction.html?id=' + v, '当前角色（' + rname + '）');

}

</script>

## 构造函数

|  |  |
| --- | --- |
| 名称 | 说明 |
| new AR.DataGrid  (objName,tableName,tableID,isTreeGrid) | objName：主对象名（视图名或表名）  用于获取列头和数据呈现表格  tableName：表名，用于编辑或删除操作  tableID：默认dg，否则需要指定  isTreeGrid：如果为true。  同时需要指定三个参数：  dg.options = {  idField: "ID列",  treeField: "名称列",  parentField: "父级ID列"  } |

## 1：对象：

|  |  |
| --- | --- |
| 名称 | 说明 |
| PKColumn | 主键操作区（编辑、删除、保存、取消） |
| Search | 搜索查询区 |
| ToolBar | 功能按钮区 |
| HeaderMenu | 配置表头的右键菜单 |
| ContextMenu | 行的右键菜单 |

## 2：属性：

|  |  |
| --- | --- |
| 名称 | 说明 |
| Internal | 内部参数，包括：  this.Internal = {  primarykey: 主键名称,  headerData: 列头数据,  isLoadCompleted: 表格数据是否已加载完成  } |
| ToolArea | 工具区（包含搜索区和按钮区）  this.ToolArea = {  id: div的ID,  $target: JQ对象,  Search: this.Search,  ToolBar: this.ToolBar,  isHidden: 是否隐藏（包括Search隐藏和ToolBar隐藏）  } |
| id | 表格的ID |
| isTreeGrid | 【默认false】TreeGrid或DataGrid属性 |
| tableName | 表名 |
| objName | 主对象名（视图名或表名） |
| isShowCheckBox | 【默认true】是否显示复选框 |
| isShowToolArea | 【默认true】是否显示工具区（等同ToolArea.isHidden属性） |
| isEditor | 【默认false】是否启用行内编辑 |
| options | Easyui的options参数（通过此参数，复盖EasyUI的参数 |
| $target | JQ对象 |
| defaultInsertData | 行内编辑时，添加数据的默认填充数据：  示例：  dg.defaultInsertData = {  objname: objName, align: 'center', width: 100, ordernum: 100, rowspan: 1, colspan: 1, mergeIndex: 1, datatype: 'string,50,0,0'  }; |

## 3：方法：

|  |  |  |
| --- | --- | --- |
| 名称 | 参数 | 说明 |
| addWhere | （name,  value,  pattern,  isOr） | 添加默认查询条件：  示例：  dg.addWhere("f1", "3", "like");  name:字段名  value：值  pattern：操作符  like、=、>、<、>=、<=、in、 <>、isnull  、likeor(这个为自定义条件)  isOr：and、or  likeor示例：  dg.addWhere("随意", "id>10 or id<10", "likeor"); |
| removeWhere | （name） | 根据字段名移除addWhere添加的条件 |
| datagrid | （key, value） | 统一treegrid和datagrid的EasyUI用法  示例：  dg.datagrid("getChecked"); |
| getChecked | （） | 【Array】获得打勾选中的行 |
| getCheckedId | （key） | 【Array】获得打勾选中的行ID。  key：若指定列名，则取指定的列。 |
| getSelected | （） | 返回第一个被选择的行记录或null |
| getData | （） | 返回已载入数据 |
| reload | （） | 重载记录，跟'load'方法一样但是重载的是当前页的记录而非第一页。 |
| bind | （） | 绑定表格（之前都是设置参数，这个函数才开始组合参数并获取表头及数据请求） |

# AR.DataGrid 的对象介绍：

#### 对象的共性：

|  |  |
| --- | --- |
| 名称 | 说明 |
| isHidden | 是否隐藏 |
| $target | JQ对象 |
| onBeforeExecute  onExecute  onAfterExecute | 执行前（参数可能各不相同）  执行（参数可能各不相同）  执行后（参数可能各不相同） |

## 1：PKColumn

### 属性

|  |  |  |
| --- | --- | --- |
| 名称 | 类型 | 说明 |
| Items | Aries.Dictionary | 存档每行的主键列信息，只能在数据呈现之后获取,如onLoadSuccess事件 |
| Editor | {  BtnEdit,  BtnDel,  BtnSave,  BtnCancel  } | 内部有四个系统的可控制的按钮。  其它通过add方法自定义添加的，只能通过Items获取。  按钮共性：  isHidden：是否隐藏  $target：JQ对象。  相关事件：  BtnEdit：onBeforeExecute = function (value, index)  BtnDel：onBeforeExecute = function (value, index)  BtnSave：onBeforeExecute = function (value, index, postData)  BtnCancel：onBeforeExecute = function (value, index) |

### 方法

|  |  |  |
| --- | --- | --- |
| 名称 | 参数 | 说明 |
| add | key, title, clickname, url, winTitle, lv2action | /\*\*  \*如果设置clickname url&winTitle则无效  \*@param{string} key 指向$Core.Common.js文件buttons\_temp对象中的key值  \*@param{string} title 鼠标划过显示的文字  \*@param{string} clickname 事件名  \*@param{string} url 打开页面的URL  \*@param{string} winTitle 打开窗口的标题  \*@param{string} lv2action 二级权限名称  \*/ |

## 2：Search 对象

#### 属性：

|  |  |  |
| --- | --- | --- |
| 名称 | 参数类型 | 说明 |
| Items | Aries.Dictionary | 存档所有Input的对象数组，在调用bind()方法后才能获取。 |
| BtnQuery |  | onBeforeExecute(searchJsonArray)：  可以对SearchJsonArray进行修改 |
| BtnReset |  | 清空值后，调用的BtnQuery进行查询。 |

## 3：ToolBar 对象

#### 属性：

|  |  |  |
| --- | --- | --- |
| 名称 | 参数类型 | 说明 |
| Items | Aries.Dictionary | 存档所有按钮的对象数组，在调用bind()方法后才能获取 |
| BtnAdd |  | function (index, isSameLevel)  isSaveLevel：TreeGrid时添加下级还是同级 |
| BtnDelBatch |  | onBeforeExecute = function (ids, index) |
| BtnImport |  | onBeforeExecute = function (param)  //导入之前执行事件，设置参数如：param.p1 = abc;param.p2 = 123 |
| BtnExport |  | onBeforeExecute = function (param)  \*@param{object} param 需要变更或者传递附加参数可对param进行修改 |
| BtnExportTemplate |  | onBeforeExecute = function (param)  @param{object} param 需要变更或者传递附加参数可对param进行修改 |

#### 方法：

|  |  |  |
| --- | --- | --- |
| 名称 | 参数类型 | 说明 |
| add | text, fname, index, css, lv2action | /\*\*  \*向工具条添加按钮  \*@param{string} text 按钮显示的文本  \*@param{string} fname 按钮注册的事件函数名称  \*@param{int} index 按钮的索引排序值从1开始,默认值最后  \*@param{string} css 样式名称，默认值'btn-sm'  \*@param{string} lv2action 二级权限控制，默认值0  \*/ |
| addHtml | HTMLString, index, lv2action | /\*\*  \*向工具条添加自定义的HTML元素  \*@param{string} HTMLString 一个字符串标签  \*@param{string} index 按钮的索引排序值从1开始,默认值最后  \*@param{int} lv2action 二级权限控制，默认值0  \*/ |

## 4：HeaderMenu、ContextMenu 对象

#### 属性：

|  |  |  |
| --- | --- | --- |
| 名称 | 参数类型 | 说明 |
| Items | Array | 存档右键菜单的内容  [{ "text": "名称", "onclick": "JS事件", "lv2action": "权限" },  { "text": "名称2", "onclick": "JS事件2", "lv2action": "权限2" }  ] |

#### 方法

|  |  |  |
| --- | --- | --- |
| 名称 | 参数类型 | 说明 |
| add | text, fname, lv2action | /\*\*  \*向工具条添加按钮  \*@param{string} text 按钮显示的文本  \*@param{string} fname 按钮注册的事件函数名称  \*@param{string} lv2action 二级权限控制，默认值0  \*/ |