ogrid

* 描述及最佳实践

ogrid的是一个支持Ajax的JavaScript控件，为呈现和操作数据在网格上提供解决方案。

ogrid使用最佳实践：

1.列模型中必须有id列，使用key=true选项指定。

2.使用formatterAlias格式化数据，使用editorAlias在直接编辑数据。

3.对齐方式：金额右对齐，时间居中对齐，其余左对齐。

1. 内置格式化函数别名：
   1. num1、num2、num3、num4、num5、num6、num7、num8、num9 ：分别保留 1 - 9位小数，整数部分使用“，”作为千分符。
   2. int ：无小数部分，整数部分使用“，”作为千分符。
   3. date、time、datetime ： 时间戳格式化成日期字符串、时间戳格式化成时间字符串、时间戳格式化成日期时间字符串。
   4. a ：添加超链接。
2. 内置编辑器别名：
   1. odatepicker ：日历
   2. otextfield ：普通输入域

继承：$.smile.obase

* 选项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 选项名 | 类型 | 默认值 | 是否可变 | 描述 |
| url | String |  | Y | 请求地址。 |
| datatype | String | json | Y | 数据类型  定义期望的数据格式。  可取xml(xml格式)、xmlstring(xml字符串格式)、json(json格式)、jsonstring(json字符串格式)、  local(数据定义在客户端的数组中)、function(自定义函数)、clientSide(相当于local)  如果希望ogrid初始化完成时不请求数据，可以设置为local，再次调用research方法时请求url选项地址 |
| bottomPager | Boolean | true | N | 是否显示下分页条。 |
| toppager | Boolean | true | N | 是否显示上分页条。 |
| editModel | String |  | N | 可取值：inline(行内编辑--可同时编辑多个单元格)、cell(单元格编辑)。 |
| gridview | Boolean | true | N | 设置为true，grid一次将插入所有数据，启用这个选项有一些限制，  如果设置为true将不能使用treeGrid、subGrid和afterInsertRow事件，  如果不是用这3个功能，就将gridview设置为true，会得到可观的速度。 |
| rowNum | Integer | 10 | Y | 显示在grid的中的记录数，如果rowNum为10，服务器返回15条记录，只有10条会被装载。 |
| rowList | Array | [ 15, 20, 30 ,50, 100 ] | N | 可选显示在grid的中的记录数。 |
| viewrecords | Boolean | true | N | 如果为true，jqGrid显示记录信息，显示内容可以调整emptyrecords、recordtext。 |
| multiselect | Boolean | false | N | 启用多选，会产生新列checkbox用于多选。 |
| rownumbers | Boolean | true | N | 会产生新列显示行号。 |
| hiddengrid | Boolean | false | N | 初始化隐藏grid，不请求数据，值显示caption层，当点击显示/隐藏按钮时，显示grid的并请求数据。 |
| hidegrid | Boolean | false | N | 当caption不为""时，隐藏grid。 |
| sortname | String |  | N | 排序列名。 |
| sortorder | String | asc | Y | 顺序，可设置：desc、asc。 |
| footerrow | Boolean | true | N | 如果设置为true，将在grid记录之下，分页条之上放置一行footer table，列的数量等于在colModel中定义的。 |
| shrinkToFit | Boolean | false | N | 此选项如果被设置，定义的列宽度可能将根据grid整个的宽度被重新计算，  如果设置为true，并且列的宽度也被设置,那时每一列是其宽度的比例。  例如，如果定义2列，宽度分别为80px和120px，但是grid的宽度为300px，那时两列将伸展占满grid  重新计算：如下：第一列，300 / 200 × 80 = 120，第二列：300 / 200 × 120 = 180，现在两列的总和为300px  如果shrinkToFit设置为false，无论什么列都不会被重新计算。 |
| autowidth | Boolean | true | N | grid的宽度会自动根据父元素重新计算，这步只有初始化grid的时候会做。 |
| width | Integer |  | Y | grid宽度 |
| height | Integer |  | Y | grid高度。 |
| caption | String | ogrid caption | Y | grid的标题，为""时无标题条。 |
| cellurl | String |  | Y | cell编辑请求路径URL。 |
| editurl | String |  | Y | inline编辑请求路径URL。 |
| treeGrid | Boolean | false | N | treeGrid是在grid中展现层次结构的一种形式。  treeGrid支持Nested和Adjacency两种模型，默认使用adjacency。  启用或者禁用treeGrid风格。 |
| expandColumn | String | null | N | 指示treeGrid用来展开的列，name来自colModel。 |
| levelField | String | level | N | 此域决定元素在层级中的水平。  通常root元素被设置为0，root的第一个孩子是1，等等。  grid需要此信息设置每个元素的缩进。 |
| parentIdField | String | pid | N | 指示父节点的id，如果父id设置为null，则元素为root节点。 |
| leafField | String | leaf | N | 指示是否为叶子，可取true或者false。  是否为叶子节点关系到元素的图片和是否能被展开。 |
| expandedField | String | expanded | N | 指示grid元素在加载时是否应该被展开，可取true或者false。  如果没有值，则设置为false。  注意：数据可以为null，但是不应该被移除。 |
| sortable | Boolean | true | N | 列可拖动排序。 |
| loadonce | Boolean | false | N | 如果这个选项设置为true，grid只从服务器加载一次数据(需要使用合适的datatype，这里使用json)。  第一次请求之后datatype选项自动设置成local，并且所有之后的操作都在客户端进行。 |
| rowTotal | Integer | null | N | 设置这个参数能指示服务器端需要加载记录的总数。  注意：rowNum决定显示在grid中的记录数，而rowTotal决定grid能操作的记录总数。  比如，一页显示15条记录，一共查1000条记录，这是rowNum为15，rowTotal为1000。  当这个参数被设置时，则发送额外的参数到服务器被默认命名为totalrows，参数名可以在prmNames选项被更改。  或者检查此参数，如果可用则替换掉rows参数。大部分这个选项需要结合loadonce选项设置为true。 |
| scroll | Boolean | false | N | 创建动态滚动grid  启用时，分页元素自动禁用并且能使用垂直滚动条加载数据。  设置为true时grid总是持有所有记录从开始到最新点。  当此参数为数字时，表格只控制可见的几行，所有数据都在这几行中加载 |

* 覆盖选项

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 选项名 | 类型 | 默认值 | 是否可变 | 描述 |
|  |  |  |  |  |
|  |  |  |  |  |

* 方法

|  |  |  |
| --- | --- | --- |
| 方法名 | 参数:类型 | 描述 |
| addRowData | rowid:和id列类型相同  data:Object | 添加一行记录。  rowid：行标识。  data：行数据。 |
| delRowData | rowid | 删除grid中id等于rowid的记录。 |
| clearGridData | clearfooter | /\*\*  \* 从grid中清除当前加载的数据。  \* @param clearfooter 设置为true，则同时清除footer row的数据。  \*/ |
| getRowData | rowid:行标识 | 返回请求行的数据对象，返回对象类型：键值对{name: value}，name来自colModel，value来自这行关联的列。  如果rowid不存在，则返回空对象。  1. 当编辑row或者cell时，不要使用这个方法，他将返回cell的内容，而不是input元素的实际值。  2. 不要在循环体中使用此方法，因为会有性能问题(当调用方法一次，将计算行数据一次)。  如果rowid没有被设置，则返回grid中的所有数据。 |
| getCell | rowid  iCol | 返回单元格的内容，rowid: 行的id，iCol: 列的索引或者colModel中列的name。  当编辑row或者cell时，不要使用这个方法，他将返回cell的内容，而不是input元素的实际值。 |
| getCellEle | rowid  colName | 返回单元格td，rowid: 行的id，colName: 列的colModel中列的name。 |
| getRowEle | rowid | 返回指定行：tr，rowid: 行的id。 |
| getColEle | colName | 返回指定列：td的数组，colName: 列的colModel中列的name。 |
| getGridParam | name | 返回请求的参数。name来自options数组，如果name没有设置，所有选项被返回。 |
| setGridParam | object:Object | 设置具体的选项，参数类型：键值对{name: value}。  注意：一些选项将会影响下一个trigger(“reloadGrid”)的执行。  注意：这个方法能覆盖选项。name来自options |
| setPostParam | object:Object | 设置请求参数，参数类型：键值对{name: value}。 |
| refresh |  | 刷新grid  重新装载grid的当前设置。  这意味着如果datatype是xml或者json，一个新的请求被发送到服务器。  这个方法应该应用到一个已经被构造的grid中。  注意：这个方法不能改变HEADER信息，这意味着colModel的改变将不会生效。  应该使用gridUnload重新加载不同的colModel。它只能工作在loadonce为false! |
| research |  | 重新查询数据。  将datatype设置为json，page设置为第一页，一个新的请求被发送到服务器。  这个方法应该应用到一个已经被构造的grid中。  注意：这个方法不能改变HEADER信息，这意味着colModel的改变将不会生效。  应该使用gridUnload重新加载不同的colModel。它只能工作在loadonce为false! |
| editCell | iRow  iCol  edit | 编辑行，索引iRow(也可以为行id)，列索引iCol的单元格  如果edit设置为false，cell仅仅被选择而不被编辑，设置为true cell被选择而且被编辑。 |
| restoreCell | iRow  iCol | 恢复cell编辑，行索引iRow(也可以为行id)，列索引iCol的单元格 |
| saveCell | iRow  iCol | 保存cell，行索引iRow(也可以为行id)，列索引iCol的单元格 |
| getChangedCells | method | 获取改变的cell，method为'all'时返回所有改变的行，为'dirty'时至返回改变的cell和行id，默认为'all'。  一般和cellsubmit='clientArray'配合使用。 |
| addButton | option:Object | /\*\*  \* 在下分页条上添加按钮  \* option结构：  \* {  \* id: button的id,  \* label: 按钮上的文字,  \* click: function() {  \* 点击后的动作  \* }  \* } |
| getSelectedRow |  | 当选项multiselect为true时，返回被选中行id的一维数组，否则返回最后选择的行id。 |
| setFrozenColumns |  | 冻结列,根据colModel中的frozen选项，ogrid会自己调用。 |
| destroyFrozenColumns |  | 取消冻结。 |
| setGroupHeaders | option:Object | /\*\*  \* 设置复杂列头  \* option结构：  \* {  \* useColSpanStyle: 是否跨列,  \* groupHeaders: [  \* {startColumnName: 需要合并列的开始列名, numberOfColumns: 合并列数, titleText: 合并列名},  \* ...  \* ]  \* }  \*/ |
| hideCol | colname:String|Array | /\*\*  \*  \* 传入需要隐藏的列名，可以为单个列名："columnName"，也可以为列名的数组：["columnName1", "columnName2"]。  \* NB：grid的宽度不会改变。  \*  \* @param colname 需要隐藏的列名。  \*/ |
| showCol | colname:String|Array | /\*\*  \*  \* 传入需要显示的列名，可以为单个列名："columnName"，也可以为列名的数组：["columnName1", "columnName2"]。  \* NB：grid的宽度不会改变。  \*  \* @param colname 需要隐藏的列名。  \*/ |

* 事件

|  |  |  |
| --- | --- | --- |
| 事件名 | 参数:类型 | 描述 |
| beforeRequest |  | 在请求任何数据之前调用，如果datatype为function此函数不会调用。  如果返回false，则不会请求服务器。 |
| loadBeforeSend |  | 回调参数: xhr, settings  在发送请求之前调用，在beforeRequest之后调用，可以修改XHR，设置用户http头等，返回false取消请求。  设置为同步请求数据settings.async = false; |
| serializeGridData |  | 回调参数: postData  发请求之前序列化数据，函数返回被序列化的数据。  常常用来传入用户数据 |
| loadError |  | 回调参数: xhr, status, error  如果请求失败，此函数被调用。 |
| beforeProcessing |  | 回调参数: data, status, xhr  从服务器端获取到数据返回，还未交给grid处理，数据格式依赖datatype参数  如果datatype为'json'，则数据为JS对象。  如果返回false，则不会渲染数据。 |
| gridComplete |  | 所有数据被grid处理完并且其他(分页、排序)也处理完之后调用。 |
| loadComplete |  | 回调参数: data  Grid请求最后一个调用 |
| onSelectRow |  | 回调参数: rowid, status, e  点击一行时执行  rowid: 行号,  status: 选择的状态, 能被multiselect为true是使用，true为选择，false为取消选择。  e: 事件对象。 |
| ondblClickRow |  | 回调参数: rowid, iRow, iCol, e  双击一行时执行 |
| formatCell |  | 回调参数：rowid, cellname, value, iRow, iCol  这个事件允许在编辑之前格式化cell内容，返回格式化之后的值 |
| afterEditCell |  | 回调参数：rowid, cellname, value, iRow, iCol  cell跳到编辑状态之前调用 |
| beforeSaveCell |  | 回调参数：rowid, cellname, value, iRow, iCol。  保存cell之前调用, 此处保存的值发送到服务器, 可以用于校验，可返回新值替换旧值。 |
| beforeSubmitCell |  | 回调参数：rowid, cellname, value, iRow, iCol。  发送到服务器之前调用, 可以添加额外的请求参数，返回对象，提交到服务器。 |
| afterSubmitCell |  | 回调参数：serverresponse, rowid, cellname, value, iRow, iCol。  cell和其他数据发送到服务器之后调用，返回值格式：[success(boolean),message]。  如：  [true,""]，cell值保存成功，  [false,"Error message"]，弹出错误对话框，cell内容未保存。  servereresponse是服务器响应。  应该使用serverresponse.responseText获取信息。 |
| afterSaveCell |  | 回调参数：rowid, cellname, value, iRow, iCol。  cell被保存之后调用，这是改变其他值最理想的地方。 |
| errorCell |  | 回调参数：serverresponse, status。  服务器错误时调用(例如：403, 404, 500等)，servereresponse是服务器响应。  使用serverresponse.responseText获取来自服务器的信息  status是错误状态，如果不使用，则出现模态对话框。 |
| onSelectCell |  | 回调参数：rowid, celname, value, iRow, iCol。  用于cell不可编辑，当cell被选择时调用。 |
| afterRestoreCell |  | 回调参数：rowid, value, iRow, iCol。  调用restoreCell或者用户按ESC离开编辑状态时调用。 |
| serializeCellData |  | 回调参数：postdata。  当保存单元格时，这个事件序列化数据传入ajax请求。  这个函数应该返回被序列化的数据。  当用户数据需要传到服务器时这个事件可以用到。  例如JSON字符串，XML字符串等。 |
| serializeRowData |  | 用于inline编辑，可参考serializeCellData。 |
| oneditfunc |  | 回调参数：rowid。  在成功进入编辑状态时调用，允许用户访问之前的input域。 |
| afterrestorefunc |  | 回调参数：rowid。  调用restoreRow之后被调用  传入rowid |
| successfunc |  | 回调参数：response。  请求返回之后立即调用  传入从服务器传回的信息  返回值结构：  服务器端响应，response参数基本结构：  {  success: true或false，用于判断是否更新成功。  row: {}修改行的数据对象，用于显示。  } |
| aftersavefunc |  | 回调参数：rowid, response。  successfunc返回true时调用  传入rowid(行id), response(服务器响应)  当url被设置为clientArray时也被调用 |
| errorfunc |  | 回调参数：rowid, response。  successfunc返回false时或者服务器错误时(例如：403, 404, 500等)调用  传入rowid(行id), response(服务器响应) |

* 静态方法。

命名空间：$.smile.ogrid，使用$.smile.ogrid.methodName(params)调用

例如：

$.smile.ogrid.createFormatterAlias(“example”, function(cellvalue, options, row) {

// TODO

}, function(cellvalue, options, cell) {

// TODO

});

|  |  |
| --- | --- |
| 方法名 | 描述 |
| createFormatterAlias | /\*\*  \* 创建列格式化函数  \* @param {String} alias 格式化函数名  \* @param {Function} formatter(cellvalue, options, row) 格式化函数  \* @param {Function} unformat(cellvalue, options, cell) 恢复格式化函数  \*/ |
| formatterAlias | /\*\*  \* 创建格式化别名  \* @param {String} alias 别名  \* @param {Object} formatter 格式化定义，结构如下：  \* {  \* type: '对应列模型中的formatter选项',  \* options: {  \* 对应列模型中的formatoptions选项  \* }  \* }  \*/ |
| createEditorAlias | /\*\*  \* 创建编辑器函数  \* @param {String} editorAlias 编辑器别名  \* @param {Function} customElement(value, options) 创建元素函数  \* @param {Function} customValue(elem, operation, value) 访问值函数  \*/ |
| editorAlias | /\*\*  \* 创建编辑器别名  \* @param {String} alias 别名  \* @param {Object} editor 编辑器定义，editor结构如下：  \* {  \* type: '编辑器的类型',  \* options: {  \* 编辑器的初始化选项  \* }  \* }  \*/ |