# 使用说明

opengrid是一个表格框架，支持宽度和高度的自适应，列宽支持固定宽度、自适应内容宽度、按百分比分配宽度、按表格宽度分配等设置宽度方式。支持分组，子列表等功能。

## 语法：

|  |
| --- |
| $(**function**(){  $('#test').opengrid({  height:400,  autoRowHeight: **false**,  striped: **true**,  url:'../getdata1.do',  pagination:{  pageSize : 10,  pageNo : 1,  orderBy:'id desc'  },  remoteSort: **false**,  idField:'id',  frozenColumns:[],  columns:[[  {field:'rownumber',title:'序号', width:'40', nowrap:**true**, type:'rownumber'},  {field:'id',title:'ID',width:'100',editor:'text',nowrap:**true**} ]] });  }); |

## 设计表头：

columns, frozenColumns值是一个二维数组，数组第一维表示有多少行表头，数组第二维表示每一行有多少列。

如：

|  |
| --- |
| *[[{field:'rownumber',title:'序号', nowrap:****true****, type:'rownumber'},*  *{field:'id',title:'ID',width:'100',editor:'text',nowrap:****true****} ]]* |

表示的是一个有一行两列的表头。

通过rowspan，colspan可以灵活的制作多行多列表头，rowspan表示当前单元竖跨多少行，colspan表示当前单元格横跨多少列，用法与html语法一样。

如：

|  |
| --- |
| [  [{field:'rownumber',title:'序号', width:'40', nowrap:**true**, type:'rownumber', rowspan:3 },{field:'',title:'基本信息',width:'40',nowrap:**true**,type:'',colspan:3}],  [{field:'',title:'基本信息1',width:'40',nowrap:**true**,type:'', colspan:2}, {field:'age',title:'年龄', width:'40', nowrap:**true**, editor:'text',type:'',rowspan:2, sortable:**true**}], [{field:'id',title:'ID',width:'40',editor:'text', nowrap:**true**},{field:'name',title:'Name',width:'100',editor:'text',nowrap:**true**}]  ] |

其效果和下列html书写的表格一样：

|  |
| --- |
| <table>  <tbody>  <tr>  <td rowspan=*"3"* field=*"rownumber"*>序号</td>  <td colspan=*"3"*>基本信息</td>  </tr>  <tr>  <td colspan=*"2"*>基本信息1</td>  <td rowspan=*"2"* field=*"age"*>年龄</td>  </tr>  <tr>  <td field=*"id"*>ID</td>  <td field=*"name"*>Name</td>  </tr>  </tbody>  </table> |

表示一个三行四列的表头，其效果如下：



## 控制列宽：

通过列属性的width，minWidth，maxWidth可以控制每列的宽度，minWidth，maxWidth是用来控制实际列宽的上限与下限的。

width的值可以是数字、’auto’,百分数，也可以为空。各种值意义如下：

|  |  |
| --- | --- |
| 数字 | 宽度单位为”px” |
| auto | 列宽随内容自适应宽度。 |
| 百分数 | 优先按数字、’auto’的列设置宽度后，将剩余的宽度按百分比分配到各宽度为百分数的列上。如：表格总宽度为1000px，  [{field:'',title:'基本信息1',width:'100'},  {field:'age',title:'年龄',width: '100'},  {field:'attr1',title:'列1',width: 'auto'},  {field:'attr2',title:'列2',width:'20%'},  {field:'attr3',title:'列3',width:'30%'}]  假设attr1按内容自适应宽度后宽度为200则  剩余宽度 A=1000-(100+100+200)=600;  那 attr2的宽度=A\*20% = 120;  attr3的宽度=A\*30%=210; |
| 空 | 当列宽为空时，将剩余宽度平均分给这些列,如果没有剩余宽度，则按minWidth设置 |

当实际上所有的列宽总和小于表格宽度时，想让剩余宽度平均分配到各列上时怎么办？

答：在opengrid中有个属性叫fitColumns(注意这不是列的属性)，如果fitColumns=true时，则会先按列实际宽度设置列宽，当所有列的宽度加起来小于表格宽度时，将会把剩余宽度再平均分配到各列上。如果我想让某列的宽度参数剩余宽度分配时，列有个属性叫fixed,如果fixed为true时，则不参与列宽分配。

# 属性

## opengrid属性:

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **类型** | **默认值** | **描述** |
| height | string | 'auto' | 高度，值为’auto’或数字。  ‘auto’:按内容自适应高度； |
| width | string | 'auto' | 宽度,值为’auto’或数字。  ‘auto’:按屏幕宽度。 |
| frozenColumns | array | **undefined** | 冻结列 |
| method | string | post | 到后台获取数据提交方法 |
| url | string | **null** | 从后台获取数据的url |
| loadMsg | string | "正在处理，请稍待。。。" | 加载时显示的提示信息 |
| emptyMsg | string | "没有查到数据!" | 没有查到信息时显示的提示信息 |
| fitColumns | boolean | **false** | 列自适应宽度，如果所有列没有填满表格宽度，则将剩余的表格宽度平均分配到各列上 |
| idField | string | **null** | 行数据主键 |
| singleSelect | boolean | **true** | 单选/多选 |
| selectOnCheck | boolean | **true** | Check框选中时行选中 |
| checkOnSelect | boolean | **true** | 选中行时，check框自动选中 |
| queryParams | string | **undefine** | 查询条件 |
| multiSort | boolean | **false** | 多列/单列排序 |
| remoteSort | boolean | **true** | 服务器端/客户端排序 |
| showHeader |  | **true** |  |
| scrollbarSize |  | 18 | 滚动条宽度 |
| bodyManager |  |  | 内容部分管理器，创建数据行等，可创建自己的数据行管理器 |
| editors |  |  | 编辑组件，启动编辑时，调用的编辑组件。 |
| rowStyler | **function**(rowIndex, row) {} |  | 行样式,rowIndex：行序号。Row：行数据 |
| columnOptions | object | {minWidth:40,maxWidth: 1000,resizable:true} | 列默认属性,参考列属性 |
| pagination | object |  | 分页信息，参见pagination属性 |

pagination属性：

|  |  |  |  |
| --- | --- | --- | --- |
| **属性** | **类型** | **默认值** | **描述** |
| pageNo | **int** |  | 当前页码 |
| pageSize | **int** |  | 每页记录数 |
| total | **int** |  | 数记录数 |
| pre | **function** |  | 上一页 |
| next | **function** |  | 下一页 |
| first | **function** |  | 第一页 |
| last | **function** |  | 最后一页 |
| goPage | **function** |  | 跳转到 |
| orderBy | **string** |  | 排序,与sql排序语法一样  如’id,age desc’ |

pagination默认值：

|  |
| --- |
| pagination : {  pageNo : 1,  pageSize : 10,  total : 10,  pre:**function**(){  **var** pagination = **this**.options.pagination;  pagination.goPage.call(**this**,pagination.pageNo-1);  },  next:**function**(){  **var** pagination = **this**.options.pagination;  pagination.goPage.call(**this**,pagination.pageNo+1);  },  first:**function**(){  **var** pagination = **this**.options.pagination;  pagination.goPage.call(**this**,1);  },  last:**function**(){  **var** pagination = **this**.options.pagination;  pagination.goPage.call(**this**,Math.ceil(pagination.total/pagination.pageSize));  },  goPage:**function**(page){  **var** pagination = **this**.options.pagination;  pagination.pageNo = page;  **this**.loadData();  }  } |

## Opengrid事件：

* 加载前

在加载表格前，主要设置查询参数，从后台获取数据。

|  |
| --- |
| onBeforeLoad : **function**(queryParams) {  **var** opts = **this**.options;  queryParams["pagination.pageNo"] = opts.pagination.pageNo;  queryParams["pagination.pageSize"] = opts.pagination.pageSize;  queryParams["pagination.offset"] = opts.pagination.offset;  queryParams["pagination.total"] = opts.pagination.total;  queryParams["pagination.orderBy"] = opts.pagination.orderBy;  } |

* 加载数据事件

|  |
| --- |
| loader : **function**(url,queryParams, successCallback, errorCallback) {  **var** opts = **this**.options;  **if** (!url) {  **return** **false**;  }  logger.startTimeRecord("loader");  $.ajax({  type : opts.method,  url : url,  data : queryParams,  dataType : "json",  success : **function**(data) {  logger.stopTimeRecord("loader");  successCallback(data);  },  error : **function**(a,b) {  logger.info("error:"+b);  errorCallback.apply(**this**, arguments);  }  });  } |

其它事件：

|  |  |  |  |
| --- | --- | --- | --- |
| 事件名 | 参数 | 返回值 | 描述 |
| onLoadSuccess | data |  |  |
| onLoadError |  |  |  |
| onClickRow | rowIndex, row |  |  |
| onDblClickRow | rowIndex, row |  |  |
| onClickCell | rowIndex, field, val |  |  |
| onDblClickCell | rowIndex, field, val |  |  |
| onSortColumn | sortNames, sortOrders |  |  |
| onResizeColumn | field,width |  |  |
| onSelect | rowIndex, row |  |  |
| onUnselect | rowIndex, row |  |  |
| onSelectAll | rows |  |  |
| onUnselectAll | rows |  |  |
| onCheck | rowIndex, row |  |  |
| onUncheck | rowIndex, row |  |  |
| onBeforeEdit | rowIndex, row |  |  |
| onAfterEdit | rowIndex, row, newrow |  |  |
| onCancelEdit | rowIndex, row |  |  |
| onCheckAll | rows |  |  |
| onUncheckAll | rows |  |  |

## 方法:

## 列属性

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 类型 | 默认值 | 描述 |
| field | string |  | 数据属性，对应data中的field |
| title | string |  | 表头名 |
| width | int |  | 宽度 |
| nowrap | boolean |  | 内容是否可换行 |
| editor | Editor对象 |  | 编辑控件 |
| type | string |  | 类型：  checkbox: checkbox列  rowlink: 子查询列  rownumber 序号列  xxx\_grownumber 按xxx分组时，xxx小组内序号列  xxx\_rownumber 按xxx分组时，xxx的序号 |
| rule | **function**(val,row,index) |  | type=’rowlink’时，rule返回值来控制显示子查询按钮 |
| rowspan |  |  | 列跨度 |
| colspan |  |  | 行跨度 |
| minWidth |  |  | 最小宽度 |
| maxWidth |  |  | 最大宽度 |
| resizable |  |  | 宽度是否可手动调整 |