qdTable 使用指南

一、引入

在head中引入js文件和css文件

<link rel="stylesheet" href="qd\_table.css">  
<script src="qd\_table.min.js"></script>

html中的标签标志

<table id="xxx"></table>

·若需要控制表格的几何位置时，最好使用一个父框对其进行承载，并控制父框的几何位置

<div class="father">  
 <table id="xxx"></table>  
</div>

※以下接口需在dom节点加载完毕后调用，否则将报错。简单的做法是将js写在body后

二、构造与格式

1.构建table类

**var** qdTable = **new** qd\_Table();

2.表格数据源格式

**var** dataSource = [{···}，{···}，···];

3.配置数据格式

**var** configData = {  
 xxx:{

……

},

yyy:{

……

},

……

};

三、API

※以下接口为单独调用

1.生成表格接口 <必要>

qdTable.setData(tableId,dataSource,configData);

功能：

主要方法，用于讲数据源写入table并生成最终表格。

参数：

tableId：指在html中用于做标志的table标签的id

dataSource：指要展示的表格数据源

configData：指用于初始化时调用插件功能的数据对象

用法：

Var tableId = xxx;

Var dataSource = xxx;

Var configData = xxx;

qdTable.setData(tableId,dataSource,configData);

2.重写表格接口

qdTable.resetData(tableId,dataSource,configData);

功能：重写表格数据，可修改‘数据源’/‘配置数据’

参数：（同上）

tableId：指在html中用于做标志的table标签的id

dataSource：指要展示的表格数据源

configData：指用于初始化时调用插件功能的数据对象

用法：

Var tableId = xxx;

Var dataSource = xxx;

Var configData = xxx;

qdTable.resetData(tableId,dataSource,configData);

3.设置添加行的回调函数

qdTable.setAddBtnCallBack(callback);

功能：

为“新增行”功能添加一个回调函数

参数：

Callback：回调函数。

用法：

qdTable.setAddBtnCallBack(function(itemArr){

//itemArr是添加函数的返回值，类型是array，内容是新增的数据项

});

4.设置删除行的回调函数

qdTable.setDelBtnCallBack(callback);

功能：

为“删除行”功能添加一个回调函数

参数：

Callback：回调函数。

用法：

qdTable.setDelBtnCallBack(function(indexArr){

//indexArr是删除函数的返回值，类型是array，内容是删除子项对应的下标

});

5.单独调用“添加行”

qdTable.addNewTableTr(tableId,addNum);

功能：

直接由js调用表格的“添加行”方法

参数：

tableId：需为上面setData/resetData方法中的tableId

addNum（>0）：指一次性要新增多少个数据子项，默认值为1，不符合规格会被置为1

用法：

Var tableId = xxx;

Var addNum = 5;

qdTable.addNewTableTr(tableId,addNum);

6.单独调用“删除行”

qdTable.delTableTr(tableId);

功能：

直接由js调用表格的“删除行”方法

参数：

tableId：需为上面setData/resetData方法中的tableId

用法：

Var tableId = xxx;

qdTable.delTableTr(tableId);

7.加载页面

qdTable.loadingShow(isShow,Tip);

功能：

调出加载页面

参数：

isShow：true/false，设置页面的展示与隐藏，类型Boolean。

Tip：用于修改页面上提示的内容。

用法：

展示

qdTable.loadingShow(**true**,'正在写入余下的数据，请稍后');

隐藏

qdTable.loadingShow(**false**);

※以下接口为写入配置数据中调用

8. writeLinkData

功能：对数据源中指定属性添加数据（下拉框的数据）

Value值：可单一对象/对象数组

对应的属性：

attrName：类型string，表格“数据源”中子项对象中的属性名

dataSource：类型array，关联数据的数组

isMoreSelect：类型boolean，是否为多选

用法：

//单一对象

var configData = {

……

writeLinkData:{

attrName:xxx,

dataSource:[aaa,bbb,ccc],

isMoreSelect:true/false

},

……

}

//对象数组

var configData = {

……

writeLinkData:[

{

attrName:xxx,

dataSource:[aaa,bbb,ccc],

isMoreSelect:true/false

},

{

attrName:yyy,

dataSource:[111,222,333],

isMoreSelect:true/false

},

……

],

……

}

9. rewriteTableTitle

功能：修改/自定义表头

Value值：对象

对应的属性：

dataSource：类型object，要修改表头名称的数据对象，当diyTitleData存在时，可不写

diyTitleData：类型array，要自定义表头的数据对象

用法：

var configData = {

……

rewriteTableTitle:{  
 dataSource:{  
 one:'周1',  
 two:'周2',  
 three:'周3',  
 four:'周4',  
 fix:'周5',  
 six:'周6',  
 seven:'周7',  
 eight:'周树人',  
 nine:'超出'  
 },  
 diyTitleData:[  
 {  
 name:'上半部分',//一级表头  
 children:[  
 {  
 name:'a',//二级表头  
 children:[  
 {  
 name:'1',//三级表头  
 attrName:'fix',

width:xx//设置准确宽度   
 },  
 {  
 name:'2',  
 attrName:'six',

width:xx//设置准确宽度  
 },  
 {  
 name:'3',  
 attrName:'four',

width:xx//设置准确宽度  
 },  
 ]  
 },  
 {  
 name:'b',  
 children:[  
 {  
 name:'4',  
 attrName:'eight',

width:xx//设置准确宽度  
 },  
 {  
 name:'5',  
 attrName:'one',

width:xx//设置准确宽度  
 }  
 ]  
 }  
 ]  
 },  
 {  
 name:'下半部分',  
 children:[  
 {  
 name:'6',  
 attrName:'two',

width:xx//设置准确宽度  
 },  
 {  
 name:'7',  
 attrName:'three',

width:xx//设置准确宽度  
 },  
 {  
 name:'8',  
 attrName:'seven',

width:xx//设置准确宽度  
 },  
 {  
 name:'9',  
 attrName:'nine',

width:xx//设置准确宽度  
 }  
 ]  
 }  
 ]  
 },

……

}

10. setTableWidth <必要>

功能：设置表格的宽度

Value值：数据对象

对应的属性：

tableWidth：类型number，表格的宽度

isEqual：类型boolean，是否均分内容部分的格子，建议true（均分）

用法：

var configData = {

……

setTableWidth:{

tableWidth:2000,

isEqual:true

},

……

}

11. setTableLineHeight

功能：设置表格中行的行高

Value值：数据对象

对应的属性：

lineHeight（>0）：类型number/string，表示行高，不符合规格时为30 (默认)

用法：

var configData = {

……

setTableLineHeight:{

lineHeight:’30px’/30

},

……

}

12. showColumn <必要>

功能：设置要展示的列

Value值：数据对象

对应的属性：

attrNameArr：类型array/string，表示要展示的列名称集合，all表示全显示，特定显示要传入数组

用法：

var configData = {

……

showColumn:{

attrNameArr:’all’/[……，four,……]

},

……

}

13. setEditColumn

功能：设置可编辑的列

Value值：数据对象

对应的属性：

editNameArr：类型array/string，表示要展示的列名称集合，all表示全编辑，特定可编辑要传入数组

用法：

var configData = {

……

showColumn:{

attrNameArr:’all’/[……，four,……]

},

……

}

14. setColumnOrder

功能：设置列的显示顺序

Value值：数据对象

对应的属性：

columnOrder：类型array，表示展示列的顺序，不用写全，没在数组中的属性将按照数据源中的顺序排序。

用法：

var configData = {

……

setColumnOrder:{

columnOrder:['fix','six','four','eight']

},

……

}

15. setPartRows

功能：设置表格是否要分页显示

Value值：数据对象

对应的属性：

partRows：类型number，表示一页展示的最大行数

isPaging：类型boolean，是否要分页显示

用法：

var configData = {

……

setPartRows:{

partRows:10,

isPaging:true

},

……

}

16. buttonRender

功能：对默认功能按钮区进行编辑

Value值：数据对象

对应的属性：

isCleanAll：类型boolean，表示是否要清空按钮区的默认按钮

htmlStr：类型array，存储你要添加进按钮区的节点html

用法：

var configData = {

……

buttonRender:{

isCleanAll:false,

htmlStr:[

'<button class="\_qd\_funBtnSize" style="background-color: #000000;color: #ffffff;" name="save" onclick="funBtnClick(this)">保存</button>',

'<div class="searchInputDiv"><input class="searchInput" type="text"><button class="\_qd\_funBtnSize searchBtn">搜索</button></div>'

]

},

……

}

17. tableLabelRender

功能：对特定列中的格子内容进行自定义编辑

Value值：可单一对象/对象数组

对应的属性：

attrName：类型string，要写入列在表格“数据源”中对应的属性名

htmlStr：类型string，要写入格子的html

用法：

//单一对象

var configData = {

……

tableLabelRender:{

attrName:'one',

htmlStr'<button class="\_qd\_funBtnSize" style="width:70px;height:30px;background-color: #000fff;color: #ffffff;margin: 0" name="setHeight" onclick="heightPanelShow(this)">设置高度</button>'

},

……

}

//对象数组

var configData = {

……

tableLabelRender:[

{

attrName:'one',

htmlStr:'<button class="\_qd\_funBtnSize" style="width:70px;height:30px;background-color: #000fff;color: #ffffff;margin: 0" name="setHeight" onclick="heightPanelShow(this)"

>设置高度</button>’

},

{

attrName:'five',

htmlStr'<button class="\_qd\_funBtnSize" style="width:70px;height:30px;background-color: #000fff;color: #ffffff;margin: 0" name="setHeight" onclick="heightPanelShow(this)"

>设置高度</button>'

},

……

],

……

}

18. openEditBox

功能：设置指定列的编辑模式为弹出框模式，设置了关联数据writeLinkData的列将自动设置为弹出框。

Value值：数组

用法：

var configData = {

……

openEditBox:['four','eight']

……

}