# Kendo UI

## Grid

### 设置界面表格高度

在界面中装载表格的地方，使用 style 指定高度。

<**div style="height**: 300**px**;**"**>  
 <**div id="grid"**></**div**>  
</**div**>

### saveChanges

grid的saveChanges方法会自动获取表格新增和修改的数据，然后根据type自动调用新增或修改方法。

注意：如果想要在保存之后刷新表格，最好不要使用saveChanges方法。因为保存操作没有完成时，就已经发出了查询请求，此时，新增的数据将查询不到。如果非要使用，可以在一定延迟后执行刷新操作。

**function** *toSave*() {  
 *$*(**'#grid'**).data(**'kendoGrid'**).saveChanges();  
 setTimeout(*refresh*, 500);  
}  
**function** *refresh*() {  
 **var** pageNo = *$*(**'#grid'**).data(**'kendoGrid'**).dataSource.**\_page**; *$*(**'#grid'**).data(**'kendoGrid'**).dataSource.read(pageNo);  
}

### 表格中刷新当前页

//刷新Grid表格  
function refushGrid() {  
//获取当前页面编号  
var pageNo = $('#grid').data('kendoGrid').dataSource.\_page;  
//刷新页面  
$('#grid').data('kendoGrid').dataSource.read(pageNo);  
}

### 设置表格的某列不可编辑

1. 在数据源读取数据时通过 editable 属性指定：

var crudServiceBaseUrl = '${base.contextPath}',  
dataSource = new kendo.data.DataSource({  
transport : {  
 read : {  
 url: crudServiceBaseUrl + "/shop/query",  
 type: "POST",  
 //contentType: "application/json",  
 dataType: "json"  
  
},  
create : {  
 url: crudServiceBaseUrl + "/shop/submit",  
 contentType: "application/json",  
 type : "POST"  
},  
update: {  
 url: crudServiceBaseUrl + "/shop/submit",  
 contentType: "application/json",  
 type: "POST"  
},  
destroy: {  
 url: crudServiceBaseUrl + "/shop/remove",  
 contentType: "application/json",  
 type: "POST"  
},  
parameterMap: function (options, type) {  
 if (type !== "read" && options.models) {  
 var datas = Hap.prepareSubmitParameter(options, type);  
 console.info("datas",datas);  
return kendo.stringify(datas);  
} else if (type === "read") {  
return Hap.prepareQueryParameter(viewModel.model.toJSON(), options);  
}  
}  
},  
batch : true,  
serverPaging: true,  
serverSorting: true,  
pageSize : 10,  
schema : {  
data : 'rows',  
total : 'total',  
model : {  
id : "shopId",  
fields: {  
shopId : {type: "long" },  
source : {type: "string",validation:{required:true},editable:false},  
shopName : {type: "string",validation:{required:true}},  
shopCode : {type: "string",validation:{required:true}},  
defaultDeliveryMode : {type: "string",validation:{required:true}},  
availableAutomaticSync : {type: "string"}  
}  
}  
}  
});

1. 在表格列中通过 editor 配置设置：

{  
field: "warehouseCode",  
title: '仓库代码',  
attributes: {style: "text-align:left"},  
headerAttributes: {  
"class": "table-header-cell",  
style : "text-align: center"  
},  
width: 50,  
editor: function(container, options){  
container.html(options.model.warehouseCode);  
container.removeClass('k-edit-cell');  
}  
  
},

1. 设置多列不可编辑

dataSource = new kendo.data.DataSource({  
transport: {  
read: {  
url: BaseUrl + "/mst/base/items/query",  
type: "POST",  
dataType: "json"  
},  
......  
parameterMap: function (options, type) {  
if (type !== "read" && options.models) {  
var datas = Hap.prepareSubmitParameter(options, type)  
return kendo.stringify(datas);  
} else if (type === "read") {  
return Hap.prepareQueryParameter(viewModel.model.toJSON(), options)  
}  
}  
},  
batch: true,  
serverPaging: true,  
pageSize: 20,  
schema: {  
data: 'rows',  
total: 'total',  
model: {  
id: "baseItemId",  
fields: {  
price : {  
validation:{  
checkPrice :function(input){  
if(input.is("[name='price']") ){  
if(input.val() < minPrice){  
input.attr("data-checkPrice-msg", "<@spring.message 'homs.pricerule'/>");  
return false;  
}  
return true;  
}  
}  
}  
}  
},  
editable : function(col){  
if(col =='skuId'||col == 'itemName'||col =='itemBarcode' ||col =='primaryUom'  
|| col == 'itemType'){  
return false;  
}  
return true;  
}  
}  
}  
});

### 设置表格列必须：

1. 通过数据源读取数据时设置：
2. schema : {  
   data : 'rows',  
   total : 'total',  
   model : {  
   id : "shopId",  
   fields: {  
   shopId : {type: "long" },  
   source : {type: "string",validation:{required:true},editable:false},  
   shopName : {type: "string",validation:{required:true}},  
   shopCode : {type: "string",validation:{required:true}},  
   defaultDeliveryMode : {type: "string",validation:{required:true}},  
   availableAutomaticSync : {type: "string"}  
   }  
   }  
   }

### 设置第一行为行序号

1. 设置行序号由1到数据源总体数量。第一页从1开始，第二页从pagesize+1开始。

columns: [{  
 **field** : **""**,  
 **title** : **'序号'**,  
 **attributes** : {  
 **style** : **"text-align:center"** },  
 **width** : 50,  
 **headerAttributes** : {  
 **"class"** : **"table-header-cell"**,  
 **style** : **"text-align: center"** },  
 **template** : **"<span class='row-number'></span>"** },

……  
 dataBound : **function**() {  
 **var** rows = **this**.items();  
 **var** page = **this**.pager.page() - 1;  
 **var** pagesize = **this**.pager.pageSize();  
 *$*(rows).each(  
 **function**() {  
 **var** index = *$*(**this**).index() + 1 + page \* pagesize;  
 **var** rowLabel = *$*(**this**).find(**".row-number"**);  
 *$*(rowLabel).html(index);  
 });  
 }  
}).data(**"kendoGrid"**);

1. 设置表格第一列为行序号

*//将rownumber中的#改成序号  
$*(**function**(){  
 **var** rownumList = *$*(**'[data-index="0"]'**);  
 **for** (**var** i=0;i<rownumList.**length**;i++)  
 {  
 **var** rownumText = *$*(rownumList[i]).html();  
 **if**(**"#"**==rownumText){  
 *$*(rownumList[i]).html(**"序号"**);  
*$*(rownumList[i]).eq(0).parent().parent().prev().children().eq(0).css(**"width"**,**"50px"**);  
*$*(rownumList[i]).eq(0).parent().parent().parent().parent().parent().next().children(**"table"**).children(**"colgroup"**).children().eq(0).css(**"width"**,**"50px"**);  
 }  
 }  
});

### 获取表格中鼠标焦点所在行的数据

$("#grid").on("click", "tr.k-state-current", function (e) {  
var data = $("#grid").data("kendoGrid").dataItem($(e.target).closest("tr")); //获取选中行对象  
});

### 获取表格中选中的数据

第一种方法

previewFunction:function(e){  
var grid = $('#grid').data('kendoGrid');  
//获取选中项  
var grid\_data = grid.selectedDataItems();  
  
data\_selected = new kendo.data.DataSource({  
data: grid\_data  
}) ;  
grid.setDataSource(data\_selected);  
},

第二种方法

previewFunction:function(e){  
var grid = $('#grid').data('kendoGrid');  
//获取选中项  
  
var selectedRows = grid.select();  
var selectedDataItems = [];  
for (var i = 0; i < selectedRows.length; i++) {  
var dataItem = grid.dataItem(selectedRows[i]);  
selectedDataItems.push(dataItem);  
}  
  
data\_selected = new kendo.data.DataSource({  
data: selectedDataItems  
}) ;  
grid.setDataSource(data\_selected);  
},

### 设置表格第一列为行序号

表格中有一个rownumber属性和一个selectable属性。其中，rownumber属性用于控制显示表格前的序号（此时表格序号对应的 field 显示的是“#”号，上面的函数就是将对应的 field 变成“序号”的），如果不想显示选择框，可以将表格的 selectable属性去掉。



*<!--将rownumber中的#改成序号 -->*$(function () {  
var rownumList = $('[data-index="0"]');  
for (var i = 0; i < rownumList.length; i++) {  
var rownumText = $(rownumList[i]).html();  
if ("#" == rownumText) {  
$(rownumList[i]).html("序号");  
$(rownumList[i]).eq(0).parent().parent().prev().children().eq(0).css("width", "50px");  
$(rownumList[i]).eq(0).parent().parent().parent().parent().parent().next().children("table").children("colgroup").children().eq(0).css("width", "50px");  
}  
}  
$("#pricerowGrid").kendoGrid({  
dataSource: priceRowDataSource,  
height: '300px',  
resizable: true,  
scrollable: true,  
navigatable: false,  
rownumber: true,  
selectable: 'multiple, rowbox',  
pageable: {  
pageSizes: [5, 10, 20, 50],  
refresh: true,  
buttonCount: 5  
},  
...  
});

### 双击表格中的某一行，跳转到对应界面

方法：$(selector).on(events,[selector],[data],fn)

//双击弹框，pricerowGrid 是一个选择  
pricerowGrid.on('dblclick', '.k-grid-content tr', function (event) {  
var data = $('#pricerowGrid').data("kendoGrid").dataItem($(event.target).closest("tr"));  
window.top.openTab(data.pricerowId, "价格行" + data.pricerowId, "${base.contextPath}/mst/productDetail/PriceRowDetail.html?pricerowId=" + data.pricerowId + "&productId=" + data.productId + "&code=" + data.code);  
});

也可以调用一个弹窗函数，传递参数过去

grid.on('dblclick', '.k-grid-content tr', function (event) {  
// console.log("点击",djiango);  
var data = $('#Grid').data("kendoGrid").dataItem($(event.target).closest("tr"));  
var id = data.id;  
var code = data.code;  
entryFunction(id, code); //跳转函数  
});  
  
//跳到详情界面  
function entryFunction(id, code){  
window.top.openTab(code,"短信模板","${base.contextPath}/om/om\_mailtemplate\_detail.html?id="+id);  
}

### 表格中行上移下移

实体类里设置一个字段，用这个字段排序。

{  
  
command: [  
{  
name: 'ding',  
template: '<**a class="k-button k-button-icontext k-grid-ding" title="上移"**><**span class="fa fa-flash"**></**span**></**a**>',  
click: function (e) {  
e.preventDefault();  
var grid=$('#grid').data("kendoGrid");  
var dataItem = grid.dataItem($(event.target).closest("tr"));  
  
var dataItems=grid.dataSource.data();  
for(var i in dataItems)  
{  
if(i!=0) {  
if (dataItems[i].id == dataItem.id) {  
var dataChange=dataItems[i-1];  
var priorityChange=dataItem.priority;  
dataItem.priority=dataChange.priority;  
dataChange.priority=priorityChange;  
var map=[];  
map.push(dataItem);  
map.push(dataChange);  
$.ajax({  
url: BaseUrl + "submit",  
type : "POST" ,  
contentType: "application/json",  
data:JSON.stringify(map),  
success:function (data) {  
if(data.success)  
{  
grid.dataSource.read();  
}else {  
kendo.ui.showErrorDialog({  
title:"提示",  
message:data.msg  
})  
}  
}  
})  
}  
}  
}  
  
}  
}  
],  
// locked: true,  
title: "上移",  
width: 120,  
headerAttributes: {  
style: "text-align:center"  
},  
attributes: {  
style: "text-align:center"  
}  
}

### 表格中声明组件

{  
field: "payMode",  
title: '退款方式',  
attributes: {style: "text-align:center"},  
headerAttributes: {  
"class": "table-header-cell",  
style: "text-align: center"  
},  
width: 100,  
template: function (dataItem) {  
var v = dataItem.payMode;  
$.each(PayTypeData, function (i, n) {  
if(n.value == v){  
v = n.meaning;  
return v;  
}  
})  
return v;  
},  
editor: function (container, options) {  
$('<**input name="' + options.field + '"**/>')  
.appendTo(container)  
.kendoDropDownList({  
dataTextField: "meaning",  
dataValueField: "value",  
dataSource: PayTypeData  
});  
}  
},

### 表格中添加编辑/删除列

{  
title: '<@spring.message "hap.delete"/>',  
width: 50,  
//如果该行id不为空，则添加一个编辑超链接  
template: function (rowdata) {  
if (!!rowdata.asRefundEntryId) {  
//点击链接后不会跳转，会调用 onClick 方法。  
return '<**a href="javascript:void(0);" onclick="***editFunctionResources*(\**''**+rowdata.**asRefundEntryId** +**'\')"**><@spring.message "hap.delete"/></**a**>'  
}  
return '';  
},  
attributes: {style: "text-align:center"},  
headerAttributes: { style: "text-align:center"}  
},

### 表格中设置选择一列的值，设置其他列的值随之改变

{  
field: "code",  
title: '<@spring.message "hmall.product.code"/>',  
width: 120,  
template:function(dataItem){  
if(dataItem.code !=null){  
return dataItem.code;  
}else{  
return '';  
}  
},  
editor : function(container, options){  
$('<**input required name="' + options.field + '"**/>').appendTo(container).kendoLov($.extend(${lovProvider.getLov(base.contextPath, base.locale, "LOV\_PRODUCT")},  
{  
select : function(e){  
options.model.set("productId",e.item.productId);  
options.model.set("code",e.item.code);  
options.model.set("productName",e.item.name);  
options.model.set("productType",e.item.productType);  
},  
query:function(e){  
  
}  
}))  
},

### 表格中声明时间组件

注意：要指定日期格式，否则表格中不显示

{  
field: "estimateConTime",  
title: '<@spring.message "hmall.OrderEntry.estimateConTime"/>',  
width: 180,  
format: "{0:yyyy-MM-dd HH:mm:ss}",  
editor: function (container, options) {  
$('<**input id="'+ options.field +'" name="' + options.field + '" data-value-primitive="true"**/>')  
.appendTo(container)  
.kendoDateTimePicker({  
  
}).data("kendoDateTimePicker");  
}  
}

### 表格中设置指定列背景色

attributes: {  
 style: **"#:data.approveStatus=='D'?'background-color:Tomato;color:white':(data.approveStatus=='Y'?'background-color:limegreen;color:white':(data.approveStatus=='N'?'background-color:yellow':''))#"** },

<http://blog.csdn.net/chelen_jak/article/list/20>

里面还有一些grid的操作，可学习