KENDO GRID TIPS AND TRICKS

OUR DEVELOPERS FOUND THESE SNIPPETS USEFUL IN MULTIPLE PROJECTS

PRESENTED BY: AIS TEAM

AI SOFTWARE LLC
WWW.AISOFTWARELLC.COM

1. HOW TO CLEAR FILTER WHILE ADDING NEW ROWS IN THE KENDO GRID

If you are using Kendo Grid with inline editing, one issue you will observe is if user filters the rows and then wants to add new records, the empty row will not display. To fix this, the filter should be cleared so that the empty row displays.

Here is a sample code to ensure that filter is cleared when [Add New Record] is clicked.

```
//this will attach event to Add
kendo.ui.Grid.fn.onAddClearFilter = function()
{
    var grid = this;
    grid.wrapper.find(".k-grid-add").on("click", function(e) {
        //send to first page if it's not already
        if (grid.pager) pager.page(1);
        //clear the filter
        if (grid.dataSource) grid.dataSource.filter([]);
    });
}
```

```
//the page where you have declared the grid and databinding, call above method as
following:

var grid = $(“#grid-id”).kendoGrid({
//all your options
}).data(“kendoGrid”);

//call as below
grid.onAddClearFilter();
```

Here is a working example on dojo: http://dojo.telerik.com/ofokE

2. HOW TO GET CURRENT ROW IN KENDO GRID

To implement various validation rules or business logic, it could be important to know which row in the grid user is currently editing. The available functions in grid are helpful but can be tweaked to get the behavior you may need during validation.

Here is a custom implementation of getting current row in the grid. Note that for this to work correctly, you need to have navigatable set to true and/or selectable property set to {“single”, “multiple”, true}.

```
//returns current row element
kendo.ui.Grid.fn.currentRow = function() {
    //this will only work if grid is navigatable
    var cell = this.current();
    if (cell) {
        return cell.closest('tr')[0];
    }
    //following will only work if grid is selectable, it will get the 1st row only for
multiple selection
    if (this.options.selectable != false)
        return this.select()[0];
    return null;
}
```

```
Configure the grid as following:
```

```
var grid = $("#grid").kendoGrid({
…
    selectable: true, //either one or
    navigatable: true  //this should be true
…
});
```

```
//How to use currentRow function

//in change or edit events, get the grid object and directly call
//    .currentRow() for the grid object. It will return null if
//    the selectable and navigatable both are false. Otherwise
//    you will get the row element. Calling grid.dataItem(row) gives
//    the data item associated with current row

change: function(e)
{
    var grid = this;
    var row = grid.currentRow();
    if(row)
        console.log(grid.dataItem(row));
},
edit: function(e)
{
    var grid = this;
    var row = grid.currentRow();
    if(row)
        console.log(grid.dataItem(row));
}
```

See a live working example on dojo here - http://dojo.telerik.com/IwoDA/2

3. HOW TO CHECK ALL CHECKBOXES IN ROWS USING HEADER CHECKBOX

It's a common requirement to have a checkbox in the one of the column and also have a checkbox in the column header. When user clicks on header checkbox, all the checkboxes in all rows should be checked or toggled. This can be achieved with a following approach.

```
function toggleAll(className) {
    e = window.event;
    var checked = $(e.target).context.checked;
    //var grid = $(e.target).closest('.k-grid').data("kendoGrid");

    $(e.target).closest("div .k-grid-
header").siblings().find(className).each(function(ndx) {
        $(this).prop('checked', checked);
    });
}
```

To use above function, here is how the grid configuration would looks like:

```
var grid = $("#grid").kendoGrid({
    toolbar: ["create"],
    columns: [
      { field: "",
          headerTemplate: '<input type="checkbox" onclick="toggleAll(\'.sel-
checkbox\')" />',
          template: "<input class='sel-checkbox' type=checkbox />"
      },
      { field: "name" },
      { field: "age" },
      { field: "percent"},
      { field: "",
          headerTemplate: '<input type="checkbox" onclick="toggleAll(\'.sel-
checkbox2\')" />',
          template: "<input class='sel-checkbox2' type=checkbox />"
      },
    ],
    //your rest of the grid configuration..
}
```

Please note that className is passed here to the toggleAll function to identify the column. You can employ other method such as passing field name and finding the field within each row.

Check out the live version of above code here at - http://dojo.telerik.com/uBUHi/2

Here is another version of above code that utilizes data model's field name to toggle the checkbox.

```
function toggleAll(fieldName) {
    e = window.event;
    var checked = $(e.target).context.checked;
    var grid = $(e.target).closest('.k-grid').data("kendoGrid");
    grid.dataSource.data().forEach(function(d) {
        d.set(fieldName, checked);
    });
}
```

Here is the grid configuration that calls above method:

```
var grid = $("#grid").kendoGrid({
    toolbar: ["create"],
    columns: [
      { field: "isSelected",
          headerTemplate: '<input type="checkbox" onclick="toggleAll(\'isSelected\')"
/>',
          template: '<input #=isSelected? \'checked\' : \'\' # type=checkbox />'
      },
      { field: "name" },
      { field: "age" },
      { field: "percent", template: "#=kendo.format('{0:p}', percent)#" },
      { field: "isActive",
          headerTemplate: '<input type="checkbox" onclick="toggleAll(\'isActive\')"
/>',
          template: '<input #=isActive? \'checked\' : \'\' # type=checkbox />'
      },
    ],
    //rest of the configuration
```

In above example, it's assumed that you have a field for the checkbox in your data model and you want to set the field value in all records within data source.

Check out the live version on dojo - http://dojo.telerik.com/ogAhe/2

4. HOW TO AUTOFIT ALL COLUMNS OF A KENDO GRID

When you need to autofit all columns of a kendo grid based on its content, kendo grid provides an 'autofitColumn' method. Here is an example of how to use this method effectively in following two types of scenarios:

1) When you have multiple columns under one title (grouped columns)
2) When the total width of all columns don't fill the entire width of the screen

Two of these issues are addressed in following approach:

```
//following function is used to fit all the columns per its content
kendo.ui.Grid.fn.fitColumns = function(parentColumn) {
    var grid = this;
    var columns = grid.columns;
    if (parentColumn && parentColumn.columns)
        columns = parentColumn.columns;
    columns.forEach(function(col) {
        //recursive call for the grouped columns
        if (col.columns)
            return grid.fitColumns(col);
        grid.autoFitColumn(col);
    });
    //this will expand columns if empty space is left
    grid.expandToFit();
}//fitColumns
```

```
//this will expand all the column sizes within kendo grid if
//  after autofit, there empty space left
//
kendo.ui.Grid.fn.expandToFit = function() {
    var $gridHeaderTable = this.thead.closest('table');
    var gridDataWidth = $gridHeaderTable.width();
    var gridWrapperWidth = $gridHeaderTable.closest('.k-grid-header-
wrap').innerWidth();
    // Since this is called after column auto-fit, reducing size
    // of columns would overflow data.
    if (gridDataWidth >= gridWrapperWidth) {
        return;
    }

    var $headerCols = $gridHeaderTable.find('colgroup > col');
    var $tableCols = this.table.find('colgroup > col');

    var sizeFactor = (gridWrapperWidth / gridDataWidth);
    $headerCols.add($tableCols).not('.k-group-col').each(function() {
        var currentWidth = $(this).width();
        var newWidth = (currentWidth * sizeFactor);
        $(this).css({
            width: newWidth
        });
    });
}//expandToFit
There are multiple places where you can call above function. Following are few places
where you could be best suited.

dataBound: function(e)
{
    this.fitColumns();
}

When the window resizes you can call grid resize as well as fit columns to refit the
columns within new width.

$(window).resize(function() {
    //assume that you have grid variable set correctly
    grid.resize();
    grid.fitColumns();
});
```

You can also modify above code to handle the case where, if a fix width is required for certain column then that column can be skipped in fitColumns() method.

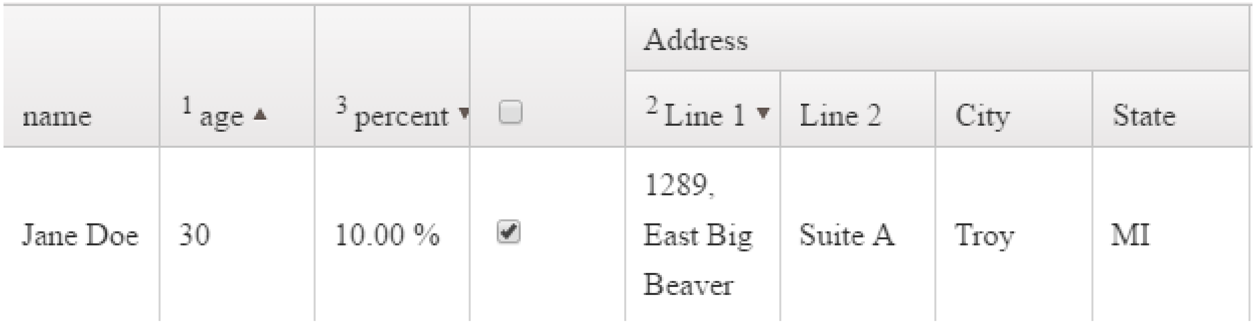
See a live example of this functions here - http://dojo.telerik.com/izExO

5. HOW TO SHOW SORT ORDINALS IN KENDO GRID COLUMNS

Kendo grid supports sorting multiple columns. Following is a documented configuration to support multi-column sort.

sortable: { mode: "multiple", allUnsort: true },

Users can click on column headers to sort the grid by that. Kendo by default will display up/down arrows to indicate whether the sorting is ascending or descending. But wouldn't it be nice to see the order of the columns in which the grid is sorted. For example, in following grid is sorted by Age Ascending, Line 1 Descending, Percent Descending – and it shows ordinals 1, 2 and 3 next to each column.



				Address			
name	^{1} age ▲	^{3} percent ▼	□	^{2} Line 1 ▼	Line 2	City	State
Jane Doe	30	10.00 %	☑	1289, East Big Beaver	Suite A	Troy	MI

Above behavior can be achieved using following code snippet. For a live demo visit - http://dojo.telerik.com/OzUbuc

```
//this is used to add sort orginals (1, 2, 3 etc.) to each sort columns
//to use this configure the multiple sorting in grid as follows and call this from
dataBound.
//example:
//  sort : { mode: "multiple", allowUnsort: true }
//  dataBound: function(e) { this.addSortOrdinals(); }
//
kendo.ui.Grid.fn.addSortOrdinals = function() {
    var self = this;
    this.thead.find('.k-link sup').remove();

    var sorted = this.dataSource.sort();
    if (!sorted) { return; }
    sorted.forEach(function(s, i) {
        var th = self.thead.find('th[data-field=' + s.field + '] .k-link');
        if (th.find('sup').length) {
            th.find('sup').text(i + 1);
        } else {
            th.prepend('<sup>' + (i + 1) + ' </sup>');
        }
    });
}//addSortOrdinals
```

6. HOW TO DISPLAY TEXT AREA EDITOR OR KENDO NUMERIC TEXT BOX WITHIN KENDO GRID

Kendo grid supports custom editors for in-cell editing within the grid. In enterprise applications, developer needs to use several types of custom editors such as following:

a. Kendo numeric text box for numeric columns
b. Kendo calendar or date time picker for date and datetime type of columns
c. Text area for a long description columns.

Here is an example of how to display text area within kendo grid.

```
function textAreaEditor(container, options) {
    $('<textarea class="k-textbox" name="' + options.field + '"
style="width:100%;height:100%;" />').appendTo(container);
}
```

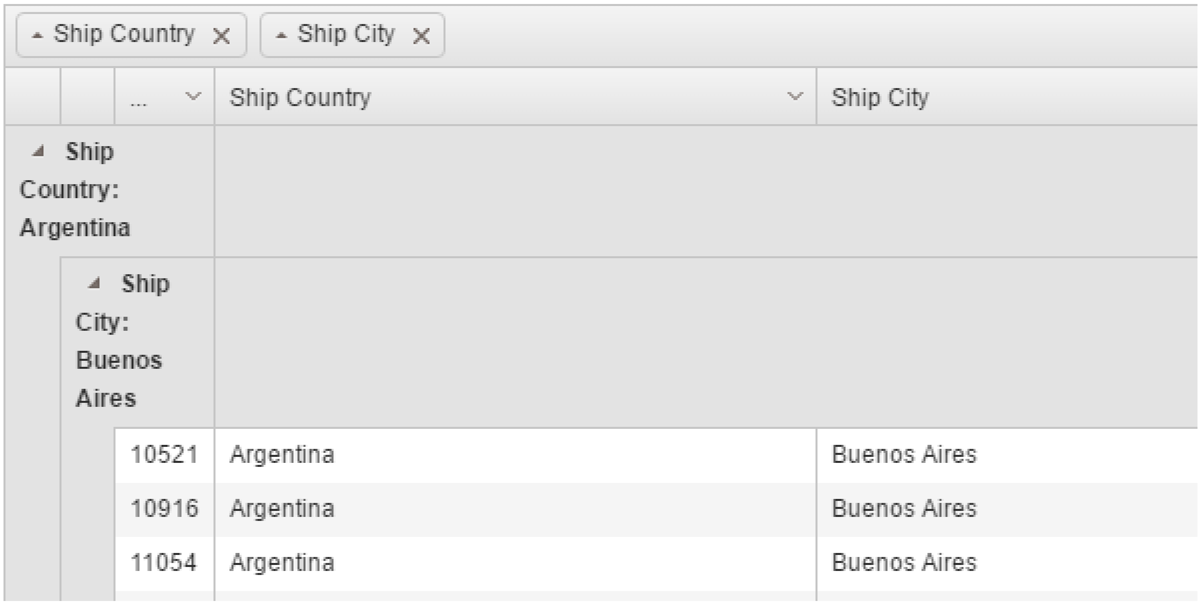
```
//use above function in grid configuration
columns: [
{ field: "description", width: "300px", editor: textAreaEditor }
]
```

Here is example to show numeric text box:

```
editor: function(container, options) {
    // create an input element
    var input = $("<input/>");
    // set its name to the field to which the column is bound ('name' in this case)
    input.attr("name", options.field);
    // append it to the container
    input.appendTo(container);
    // initialize a Kendo UI Numeric textbox
    input.kendoNumericTextBox({
        format: "p2",
        step: 0.01
    });
}
```

7. ISSUE OF GROUPING HEADER ROW WITH COLUMNS

If you have a locked column in the kendo grid and you have allowed grouping within this then you would have noticed following issue. When user groups any of the column, its title displays in the locked area and grid looks ugly. Here is the example:



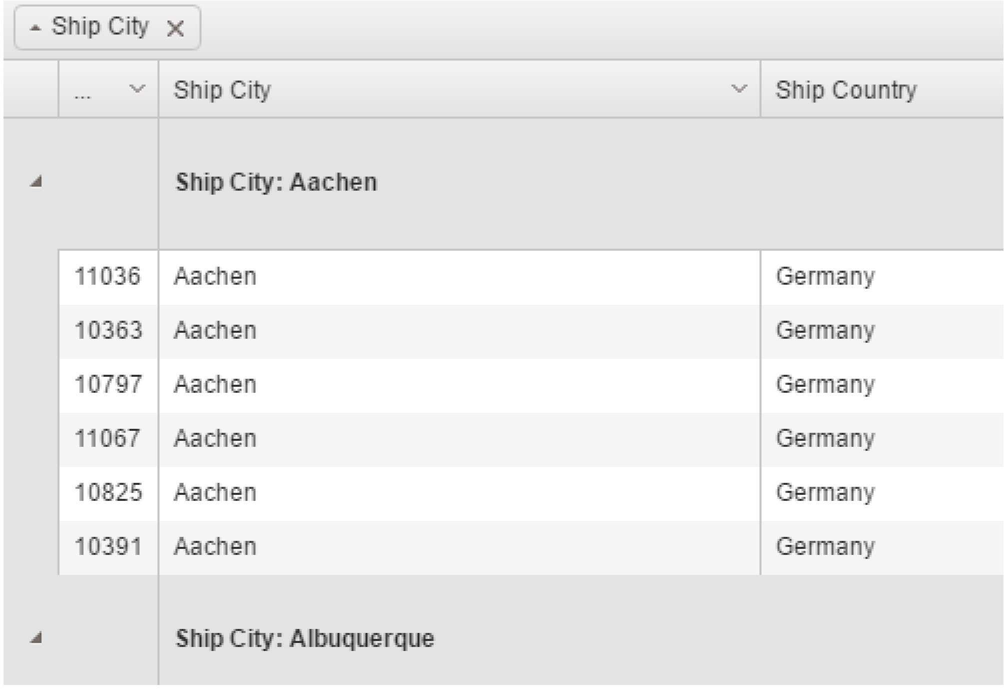
...	Ship Country	Ship City
Ship Country: Argentina		
Ship City: Buenos Aires		
10521	Argentina	Buenos Aires
10916	Argentina	Buenos Aires
11054	Argentina	Buenos Aires

To avoid above error we need to display the grouped header under the unlocked area. Here is an example of how to achieve that.

```
//fix group header in locked area
kendo.ui.Grid.fn.displayGroupHdrInUnlockedArea = function() {
    var grid = this;
    this.lockedTable.find(".k-grouping-row").each(function(index) {
        var arrow = $(this).find("a");
        grid.tbody.find(".k-grouping-row:eq(" + index + ") td").text($(this).text())
        $(this).find("p").text(" ").append(arrow);
    })
}//displayGroupHdrInUnlockedArea
```

```
dataBound: function(e){
    this.displayGroupHdrInUnlockedArea();
},
```

Here is how the grid will look after applying this fix. For a live demo visit - http://dojo.telerik.com/izaZ



...	Ship City	Ship Country
Ship City: Aachen		
11036	Aachen	Germany
10363	Aachen	Germany
10797	Aachen	Germany
11067	Aachen	Germany
10825	Aachen	Germany
10391	Aachen	Germany
Ship City: Albuquerque		

8. HOW TO SHOW CONTEXT MENU IN KENDO GRID

To display context menu within kendo grid, we can use kendoContextMenu. Here is a simple implementation of how to show context menu.

First, create an HTML for displaying context menu. You can decorate this HTML anyhow you like by adding icons and other styles of the text.

```
<ul id="ctxMenu">
        <li data-action="delete">Delete</li>
        <li data-action="duplicate">Duplicate</li>
</ul>
```

```
$("#ctxMenu").kendoContextMenu({
    orientation: "vertical",
    target: "#grid",
    filter: "tr[data-uid]",
    select: function(e) {
        //modify this function as needed. The action has the text for menu’s data-
action attribute
        //and row is the current grid row. You can get the dataItem(row) to get the
data model
        var action = $(e.item).attr("data-action");
        var row = e.target;
    }
});
```