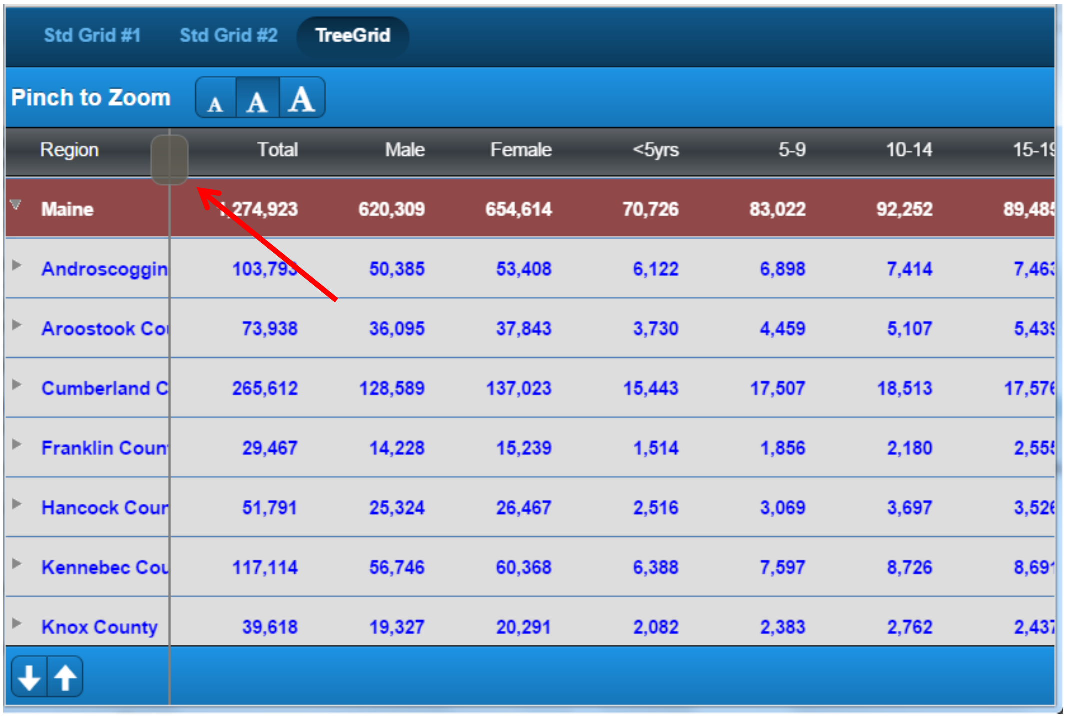
Locked Column Example - TreeGrid



Region	Total	Male	Female	<5yrs	5-9	10-14	15-19
Maine	1,274,923	620,309	654,614	70,726	83,022	92,252	89,485
Androscoggin	103,793	50,385	53,408	6,122	6,898	7,414	7,463
Aroostook Co	73,938	36,095	37,843	3,730	4,459	5,107	5,439
Cumberland C	265,612	128,589	137,023	15,443	17,507	18,513	17,576
Franklin Coun	29,467	14,228	15,239	1,514	1,856	2,180	2,555
Hancock Cour	51,791	25,324	26,467	2,516	3,069	3,697	3,526
Kennebec Cou	117,114	56,746	60,368	6,388	7,597	8,726	8,691
Knox County	39,618	19,327	20,291	2,082	2,383	2,762	2,437

a. Demonstrates

Horizontal scrolling with Locked Column for large 600 data row with 28 data elements per row Tree Grid example. Each column set references the same TreeStore, but with different column configurations. Multiple freeze and horizontal scrolling column sets can be implemented. Default thumb implemented via config “includeThumb : true” to allow user to resize locked column width. Thumb resizing is particularly useful for phone displays with limited real-estate as it allows user to resize the fixed column to see more of the other columns. See initialize() method for BigData example on how to implement for phones only. This example also demonstrates how to implement pinch event and buttons to reduce/expand grid size.

Thumbs can be represented by any supported Sencha object. Ext.Container is the default, but Ext.Button, Ext.Img, etc.. can be used. A veritical red line also appears when the thumb is being moved to indicate how the grid will be resized when the thumb drag stops. Grid minWidth/maxWidth configs also apply to stops for thumb (must be specified in pixels). Following thumbConfig object attributes can be overridden to customize the thumb:

- className: 'Ext.Container',
- height: '40px',
- width: '30px',
- offset: '1px', (used to offset the thumb from the grid border)
- top: '5px', (#pixels from top of grid container)
- style: 'background-color: #605d52; opacity:0.85; border-radius: .5em; border: 1px solid gray;',
- zIndex: 9,
- includeLine: true,
- lineColor: 'red' (color of temporary line displayed when thumb is being dragged)

All other Sencha configs supported by className can also be passed in thumbConfig.

b. Program Flow

Class: TouchTreeGrid.view.CensusFreezeCont
Controller: CommonController.js
View: CensusFreezeCont.js
Data and Columns Load:

- TouchTreeGrid.store.CensusFreezeStore (storeId = censusFreezeStore)
- Model: TouchTreeGrid.model.Census
- TreeStore (./data/censusmaine2000TREE.json)
- Data loaded via launch() method within CommonController.js:

  commonController.launch() =>
  commonController. loadCensusFreezeStore =>
  commonController. loadStore =>
  commonController.postLoadProcess =>
  commonController.loadColumnsCensusFreeze()

- censusController.loadColumnsCensusFreeze() loads columns[] array defined in pixels. Note pixels are preferred over em's in this case so that the column headers line up with the grid columns. This is necessary when font sizes are different between the two. CensusFreezeCont.js has initialize() method that overrides expand/collapse button image reducing amount of space used within locked grid.

  Column configurations are defined in this method for small, default and large profiles. Click on buttons or pinch to zoom smaller/larger on mobile devices. Scrolling grid (itemId: 'censusfreeze") defines activeProfile: 'default' which is updated within CommonController onCensusFreezeContHandlePinch() method. Both locked and scrolling grids define 'cls', 'width', 'minWidth' and 'itemHeight' for each profile via profileConfigArr[]. See onSimButtonTap() method for logic to resize via buttons. Treegriddemo.css defines default css to be overridden when small and large profiles are active.

c. Notable Configs

- CensusFreezeCont.js is outermost container for all scrolling grids. Unlimited combinations of freeze columns and horizontal scrolling column sets supported.
  - **itemId** = **'censusFreezeCont'**
  - **Layout = 'hbox'**

- **Freeze column**
  - xtype = 'touchtreegrid'
  - itemId = 'censusfreezeX'
  - listItemId = 'censusfreezelistX'
  - **linkedGridsParentItemId** = **'censusFreezeCont'**
    (ultimate parent container itemId for all scrolling grids)
  - **linkedGridsArr** = [{itemId: **'censusfreeze'** }]
    (itemId of <u>other</u> scrolling grid)
  - Columns[] array defines single dataIndex = 'CATEG' column (width = '90%').
  - categIndentPct = '0' (recommended)

- colNumberToTruncateForIndents = 0 (recommended)
- arrowPctWidth = ‘10’ (need to choose percent that allows arrow to remain visible for narrower panel width)
- Footer related configs
  - includeFooterLevels = true
(we want to include expand/collapse and footer levels in freeze column only)
  - includeFooter = true (default and not required)
- minWidth=’50px’. width = ‘130px’. maxWidth=’250px’.
- style = 'border-right: 2px solid gray;' (line border between panels)

- Custom Configs to support grid resizing and pinch event:
  - activeProfile: 'default'
  - profileConfigArr: [] (see code … defines configurations for small, default and large grid sizes)

- **Horizontal Scrolling columns**
  - Horizontal scrolling columns require a parent container with following configs:
    - Flex = 1 (optional, but at least one set of scrolling grids should have flex=1)
    - Layout = ‘hbox’
    - Scrollable = { direction: 'horizontal', directionLock: true }

  - Child container to this parent container is linked instance of TouchTreeGrid
  - xtype = ‘touchtreegrid’
  - itemId = 'censusfreeze'
  - listItemId = 'censusfreezelist'
  - **flex = 1** (required here)
  - **linkedGridsParentItemId** = **'censusFreezeCont'**
(ultimate parent container itemId for all scrolling grids)
  - **linkedGridsArr** = [{itemId: **'censusfreezeX'** }]
(itemId of other scrolling grid)
  - Columns[] array defines all scrolling columns (except frozen CATEG column and is loaded within loadColumnsCensusFreeze() method of CensusControler.js. All widths defined in pixels (vs. em’s) so that detail columns line up with header columns (which have different font-size).
  - categIndentPct = '0' (recommended)
  - colNumberToTruncateForIndents = 0 (recommended)
  - arrowPctWidth = '0' (recommended)
  - Footer related configs
    - includeFooter = true (default)
    - hideExpandCollapseBtns = true
(we want footer bar for consistency across both scrolling grids, but we don’t want expand/collapse and collapse-to button repeated on second panel)
    - includeFooterLevels = false
  - minWidth/width = **'2520px'** (includes 20px right padding)

        - Custom Configs to support grid resizing and pinch event:
            - activeProfile: 'default'
            - profileConfigArr: [] (see code … defines configurations for small, default and large grid sizes)

b. Pinch to Zoom Implementation Notes
    - Configs "activeProfile" and "profileConfigArr" are defined with grid instance and are used to switch configs based on profile
    - TouchTreeGrid.view.CensusFreezeCont contains listener on pinch event (i.e. for entire container which contains both locked and scrolling grids)

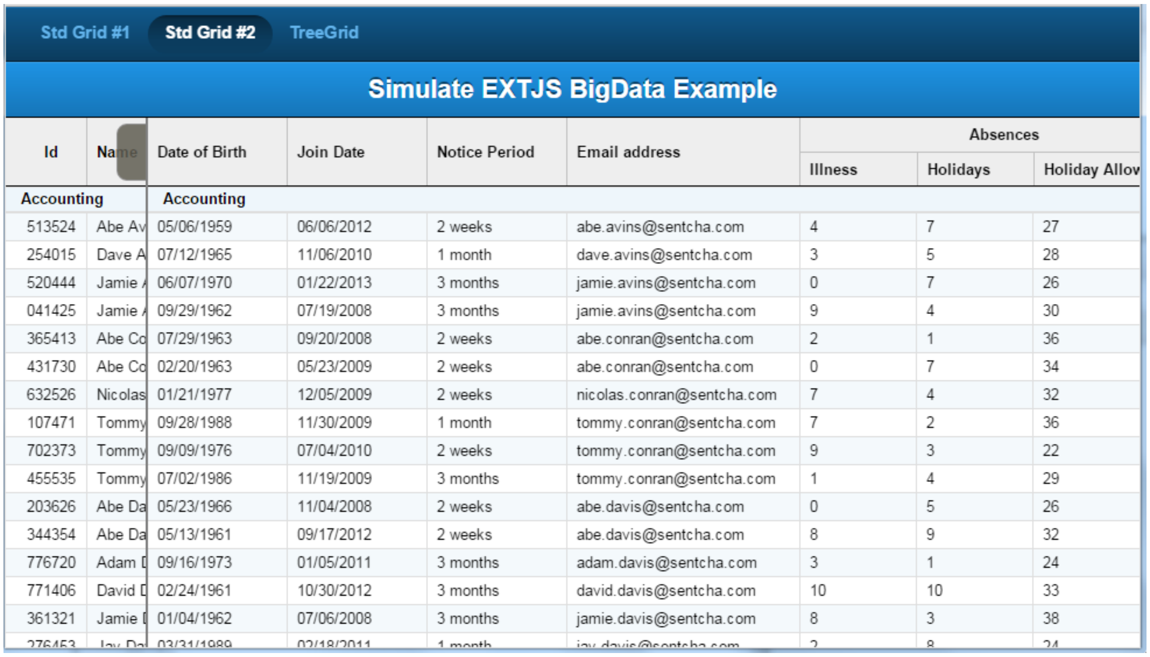
```
listeners: {
  pinch: {
    fn: function(e) {
  this.fireEvent('handlePinch', this, e);
  },
    buffer: 500,
    element: 'element'
  }
},
```

    - CommonController onCensusFreezeContHandlePinch() method processes pinch event by switching profile from small <-> default <-> large based on gesture.
    - Buttons are also included for simulation. onSimButtonTap() method handles button tap which calls onCensusFreezeContHandlePinch() method same as above

c. CSS – overrides default CSS defined in TouchTreeGrid.css via '.x-touchtreegrid-list-censusmaine' selector. Refer to treegriddemo.css:
    - .x-touchtreegrid-list-censusmaine .touchtreegrid-list-categ0 {…}
    - .x-touchtreegrid-list-censusmaine .touchtreegrid-list-categ1 {…}
    - .x-touchtreegrid-list-censusmaine .touchtreegrid-list-content{…}
    - .x-touchtreegrid-list-censusmaine .touchtreegrid-header {…}
    - Custom selectors for small and large grid configurations:
        - . x-touchtreegrid-list-censusmaine-small
        - . x-touchtreegrid-list-censusmaine-large

Standard Grid #2 – Locked Column Example

(Simulates EXTJS 4.2 KitchenSink Big Data grid example)



Simulate EXTJS BigData Example

Id	Name	Date of Birth	Join Date	Notice Period	Email address	Absences		
						Illness	Holidays	Holiday Allov
Accounting		Accounting						
513524	Abe Av	05/06/1959	06/06/2012	2 weeks	abe.avins@sentcha.com	4	7	27
254015	Dave A	07/12/1965	11/06/2010	1 month	dave.avins@sentcha.com	3	5	28
520444	Jamie A	06/07/1970	01/22/2013	3 months	jamie.avins@sentcha.com	0	7	26
041425	Jamie A	09/29/1962	07/19/2008	3 months	jamie.avins@sentcha.com	9	4	30
365413	Abe Co	07/29/1963	09/20/2008	2 weeks	abe.conran@sentcha.com	2	1	36
431730	Abe Co	02/20/1963	05/23/2009	2 weeks	abe.conran@sentcha.com	0	7	34
632526	Nicolas	01/21/1977	12/05/2009	2 weeks	nicolas.conran@sentcha.com	7	4	32
107471	Tommy	09/28/1988	11/30/2009	1 month	tommy.conran@sentcha.com	7	2	36
702373	Tommy	09/09/1976	07/04/2010	2 weeks	tommy.conran@sentcha.com	9	3	22
455535	Tommy	07/02/1986	11/19/2009	3 months	tommy.conran@sentcha.com	1	4	29
203626	Abe Da	05/23/1966	11/04/2008	2 weeks	abe.davis@sentcha.com	0	5	26
344354	Abe Da	05/13/1961	09/17/2012	2 weeks	abe.davis@sentcha.com	8	9	32
776720	Adam D	09/16/1973	01/05/2011	3 months	adam.davis@sentcha.com	3	1	24
771406	David D	02/24/1961	10/30/2012	3 months	david.davis@sentcha.com	10	10	33
361321	Jamie D	01/04/1962	07/06/2008	3 months	jamie.davis@sentcha.com	8	3	38
276453	Jay Da	03/31/1989	02/18/2011	1 month	jay.davis@sentcha.com	2	8	24

a. Demonstrates

Horizontal scrolling for Simple Lists with grouped Grids with synchronized vertical scrolling and expand/collapse functionality. This particular example simulates EXTJS 4.2 Kitchen Sink Big Data grid example. headerTplOverride config utilized to create custom grouped column headers.
Default thumb implemented to allow user to resize locked columns width (particularly useful for phones … see initialize() method for example on how to implement for phones only and to define smaller initial width). Style is modified to appear as a folder tab.

b. Program Flow

Class: TouchTreeGrid.view.BigDataCont
Data:

- TouchTreeGrid.store.BigDataStore (storeId = BigDataStore)
- Model: TouchTreeGrid.model.BigDataModel
- Store (./data/BigData.json) – dummied data provided in Sencha EXTJS 4.2.1 download (\ext-4.2.1.883\examples\kitchensink\app\data\BigData.js)
- Data loaded automatically via Store.

c. Notable Configs

- BigDataCont.js is outermost container for all scrolling grids. Unlimited combinations of freeze columns and horizontal scrolling column sets supported.
  - **itemId** = **'bigDataCont'**
  - **Layout = 'hbox'**

- **Freeze column**
  - xtype = 'touchtreegrid'
  - itemId = 'bigDataX'
  - listItemId = 'bigDataListX'

  - **linkedGridsParentItemId** = ‘**bigDataCont’**
    (ultimate parent container itemId for all scrolling grids)
  - **linkedGridsArr** = [{itemId: ' bigDataList' }]
    (itemId of <u>other</u> scrolling grid)
  - Columns[] array:

    ```
    columns: [
    {
        header: 'Id',
        dataIndex: 'employeeNo',
        width: '70px',
        style: 'text-align: center;',
        sortable: true
    },
    {
        header: 'Name',
        dataIndex: 'surname',
        width: '150px',
        style: 'text-align: left; padding-left: 8px;',
        renderer: 'this.fullName(values.forename, values.surname)',
        sortable: true
    }
    ]
    ```

  - header: {maxHeight: '59px', minHeight: '59px'}
  - itemHeight: 24
  - variableHeights: false
  - infinite: true
  - simpleList: true
  - columnSorting: true
  - additionalListConfigs: {grouped: true}
  - renderers: {fullName: function(forename, surname)
    {return forename + ' ' + surname;}}
  - headerTplOverride: (refer below)
  - width = ‘220px’ (includes 20px right padding)
  - style = 'border-right: 2px solid gray;' (line border between panels)

- **Horizontal Scrolling columns**
  Horizontal scrolling columns require a parent container with following configs:
  - Flex = 1 (optional, but at least one set of scrolling grids should have flex=1)
  - Layout = ‘hbox’
  - Scrollable = {direction: 'horizontal', directionLock: true}

  Child container to this parent container is linked instance of TouchTreeGrid
  - xtype = ‘touchtreegrid’
  - itemId = ' bigData'
  - listItemId = 'bigDataList'
  - additionalListConfigs: {grouped: true} (required for data grouping)
  - **flex = 1** (required here)
  - **linkedGridsParentItemId** = ‘**bigDataCont’**
    (ultimate parent container itemId for all scrolling grids)
  - **linkedGridsArr** = [{itemId: '**bigDataX**' }]
    (itemId of other scrolling grid)
  - Columns[] array

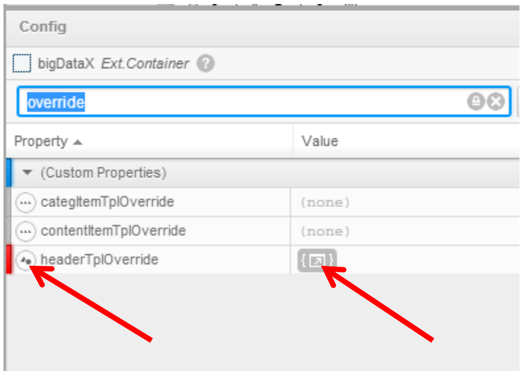
    ```
    [
        {
            header: 'Date of Birth',
    ```

```
        dataIndex: 'dob',
        width: '120px',
        style: 'text-align: left;',
        renderer: 'Ext.Date.format(values.dob, "m/d/Y")',
        sortable: true
    },
    {
        header: 'Join Date',
        dataIndex: 'joinDate',
        width: '120px',
        style: 'text-align: left;',
        renderer: 'Ext.Date.format(values.joinDate, "m/d/Y")',
        sortable: true
    },
    {
        header: 'Notice Period',
        dataIndex: 'noticePeriod',
        width: '120px',
        style: 'text-align: left;',
        sortable: true
    },
    {
        header: 'Email address',
        dataIndex: 'email',
        width: '200px',
        style: 'text-align: left;',
        renderer: 'this.emailFormat(values.email)',
        sortable: true
    },
    {
        header: 'Illness',
        dataIndex: 'sickDays',
        width: '100px',
        style: 'text-align: left;',
        sortable: false
    },
    {
        header: 'Holidays',
        dataIndex: 'holidayDays',
        width: '100px',
        style: 'text-align: left;',
        sortable: false
    },
    {
        header: 'Holiday Allowance',
        dataIndex: 'holidayAllowance',
        width: '150px',
        style: 'text-align: left;',
        sortable: false
    },
    {
        header: 'Rating',
        dataIndex: 'rating',
        width: '70px',
        style: 'text-align: left;',
        sortable: true
    },
    {
        header: 'Salary',
        dataIndex: 'salary',
        width: '110px',
        style: 'text-align: right; padding-right: 8px;',
        sortable: true,
        renderer: "this.formatNumbers(values.salary,  0, '$')"
    }
]
```

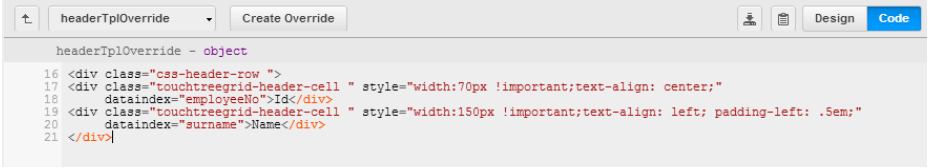
- renderers: {emailFormat: function(email) {return email;}}
- minWidth/width = '1200px' (includes right padding)
- style = 'border-right: 2px solid gray;' (line border between panels)
- other configs same as itemId = ‘bigData’

d. headerTplOverride Config
This config used to customize column headers including grouped headers. Plan is to support column headers same as Sencha grids do which is much more straight forward.

TIP: Architect allows long formatted text to be input into text string configs by changing type to "Object" for easier maintenance. Architect-generated code will be a single line of code however.



Property	Value
categItemTplOverride	(none)
contentItemTplOverride	(none)
headerTplOverride	{}



headerTplOverride - object

```
<div class="css-header-row ">
<div class="touchtreegrid-header-cell " style="width:70px !important;text-align: center;"
     dataindex="employeeNo">Id</div>
<div class="touchtreegrid-header-cell " style="width:150px !important;text-align: left; padding-left: .5em;"
     dataindex="surname">Name</div>
</div>
```

Header TPL for both Freeze and Scrolling columns:

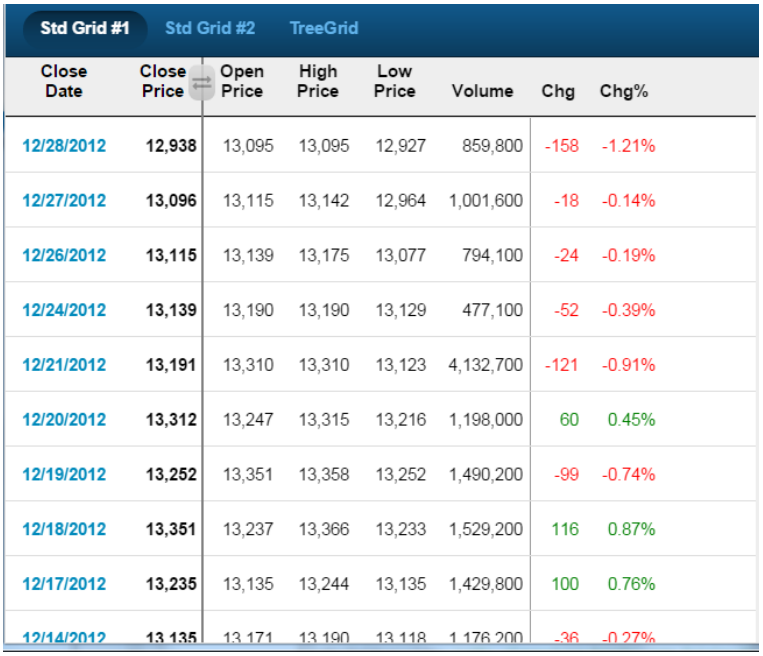
```
/* Freeze Columns (itemId = 'bigDataX') */
<div class="css-header-row ">
<div class="touchtreegrid-header-cell " style="width:70px !important;text-align: center;" dataindex="employeeNo">Id</div>
<div class="touchtreegrid-header-cell " style="width:150px !important;text-align: left; padding-left: .5em;"
dataindex="surname">Name</div>
</div>

/* Horiz Scrolling Columns (itemId = 'bigData'):
   - Column header grouping implemented using HTML table
   - Included dataindex attribute for column sorting (also requires columnSorting=true config in linked instance of
     TouchTreeGrid.
     (column sorting not currently supported for sub-grouped headers so dataindex attribute not included)
*/
<div class="css-header-row ">
<div class="touchtreegrid-header-cell " style="width:120px !important;text-align: left;" dataindex="dob">Date of Birth</div>
<div class="touchtreegrid-header-cell " style="width:120px !important;text-align: left;" dataindex="joinDate">Join Date</div>
<div class="touchtreegrid-header-cell " style="width:120px !important;text-align: left;" dataindex="noticePeriod">Notice
Period</div>
<div class="touchtreegrid-header-cell " style="width:200px !important;text-align: left;" dataindex="email">Email address</div>

<table cellpadding="0" cellspacing="0">
<tr class="touchtreegrid-header-cell " style="line-height: 29px; border-bottom: 1px solid #c0c0c0;
          border-right: 1px solid #c0c0c0; width:350px !important;text-align: center;"><td colspan="3">Absences</td></tr>
<tr>
<td class="touchtreegrid-header-cell " style="line-height: 30px; width:100px !important;text-align: left;">Illness</td>
<td class="touchtreegrid-header-cell " style="line-height: 30px; width:100px !important;text-align: left;">Holidays</td>
<td class="touchtreegrid-header-cell " style="line-height: 30px; width:150px !important;text-align: left;">Holiday Allowance</td>
</tr>
</table>

<div class="touchtreegrid-header-cell " style="width:70px !important;text-align: left;" dataindex="rating">Rating</div>
<div class="touchtreegrid-header-cell " style="width:110px !important;text-align: right;padding-right: 8px;"
dataindex="salary">Salary</div>
</div>
```

e. CSS – overrides default CSS defined in TouchTreeGrid.css via '.x-touchtreegrid-list-bigData' selector. Refer to treegriddemo.css:

- .x-touchtreegrid-list-bigData .touchtreegrid-header {...}

- .x-touchtreegrid-list-bigData .touchtreegrid-header-cell {...}
- .x-touchtreegrid-list-bigData .touchtreegrid-simplelist-cell{...}
- .x-touchtreegrid-list-bigData .x-touchtreegrid-item {...}
- .x-touchtreegrid-list-bigData .x-touchtreegrid-item:nth-child(even) {...}
  /* for alternate shading */
- .x-touchtreegrid-list-bigData .x-grid-sort-desc {...}
- .x-touchtreegrid-list-bigData .x-grid-sort-asc {...}
- .x-touchtreegrid-list-bigData .x-list-header {...} /* Grouped header rows */

Standard Grid #2 - Locked Column Example



Close Date	Close Price	Open Price	High Price	Low Price	Volume	Chg	Chg%
12/28/2012	12,938	13,095	13,095	12,927	859,800	-158	-1.21%
12/27/2012	13,096	13,115	13,142	12,964	1,001,600	-18	-0.14%
12/26/2012	13,115	13,139	13,175	13,077	794,100	-24	-0.19%
12/24/2012	13,139	13,190	13,190	13,129	477,100	-52	-0.39%
12/21/2012	13,191	13,310	13,310	13,123	4,132,700	-121	-0.91%
12/20/2012	13,312	13,247	13,315	13,216	1,198,000	60	0.45%
12/19/2012	13,252	13,351	13,358	13,252	1,490,200	-99	-0.74%
12/18/2012	13,351	13,237	13,366	13,233	1,529,200	116	0.87%
12/17/2012	13,235	13,135	13,244	13,135	1,429,800	100	0.76%
12/14/2012	13,135	13,171	13,190	13,118	1,176,200	-36	-0.27%

Demonstrates

Standard grid example with custom thumb with font image to resize locked columns width.

Program Flow

- Class: TouchTreeGrid.view. Dow2012LockedCont
- Data loaded via TouchTreeGrid.store.Dow2012 from local json file
- Significant Configs for parent container:
  - itemId: 'dow2012Cont',
  - layout: 'hbox'
- Significant Configs for Locked Grid:
  - simpleList: true,
  - itemId: 'dow2012X',
  - listItemId: 'dow2012listX',
  - linkedGridsParentItemId: 'Dow2012LockedCont',
  - linkedGridsArr: [{itemId: 'dow2012'}],
  - width: '170px'
  - includeThumb: true,
  - thumbConfig:
    - className: 'Ext.Button',
    - height: '30px',
    - width: '25px',
    - offset: '2px',
    - top: '5px',
    - text: '',
    - iconCls: 'fa-exchange',

- style: 'padding: 0;background-clr: #605d52; opacity:0.85; border-radius: .5em; border: 1px solid transparent;'
-
- Significant Configs for container parent to Scrolling Grid
  - xtype: 'container',
  - flex: 1,
  - layout: 'hbox',
  - scrollable: {direction: 'horizontal', directionLock: true}
- Significant Configs for Scrolling Grid:
  - simpleList: true,
  - itemId: 'dow2012',
  - listItemId: 'dow2012list',
  - linkedGridsParentItemId: 'Dow2012LockedCont',
  - linkedGridsArr: [{itemId: 'dow2012X'}]
  - minWidth: '26em',
  - width: '26em'

CSS – Refer to “DOW2012 HORIZ SCROLLING GRID OVERRIDES” section of treegriddemo.css