

## Basic data

<b>&lt;?xml version="1.0"?&gt;</b>	
<b>&lt;rows&gt;</b>	top xml tag, mandatory
<b>&lt;row id="11"&gt;</b>	each row must have unique ID
<b>&lt;cell&gt;</b> first column data <b>&lt;/cell&gt;</b>	
<b>&lt;cell&gt;</b> second column data <b>&lt;/cell&gt;</b>	
<b>&lt;/row&gt;</b>	
<b>&lt;row id="12"&gt;</b>	
<b>&lt;cell&gt;&lt;![CDATA[</b>	
<b>&lt;input type='button' value='any' /&gt;</b>	any hmtl can be stored inside CDATA
<b>]]&gt;&lt;/cell&gt;</b>	
<b>&lt;cell&gt;</b> second column data <b>&lt;/cell&gt;</b>	
<b>&lt;/row&gt;</b>	
<b>&lt;row id="13"&gt;</b>	
<b>&lt;cell&gt;</b> &lt;input type='button' value='any' /&gt; <b>&lt;/cell&gt;</b>	or it can be correctly escaped
<b>&lt;cell&gt;</b> second column data <b>&lt;/cell&gt;</b>	
<b>&lt;/row&gt;</b>	
<b>&lt;/rows&gt;</b>	

## Basic config

<b>&lt;?xml version="1.0"?&gt;</b>	
<b>&lt;rows&gt;</b>	
<b>&lt;head&gt;</b>	
<b>&lt;column type="edn" width="150" sort="str" format="0.00"&gt;</b>	column defined as readonly , 150px width
Column label	label of column
<b>&lt;/column&gt;</b>	
<b>&lt;column type="ro" width="150" sort="na" color="red"&gt;</b>	not sortable column
One more label	
<b>&lt;/column&gt;</b>	
<b>&lt;column type="co" width="150" sort="na" id="last"&gt;</b> Combo	combobox column
<b>&lt;option value="1"&gt;</b> One <b>&lt;/option&gt;</b>	
<b>&lt;option value="2"&gt;</b> Two <b>&lt;/option&gt;</b>	values labels of combobox values
<b>&lt;option value="3"&gt;</b> Three <b>&lt;/option&gt;</b>	
<b>&lt;/column&gt;</b>	
<b>&lt;/head&gt;</b>	
<b>&lt;/rows&gt;</b>	

## Advanced

<?xml version="1.0"?>	
<rows pos="123" total_count="2000">	smart rendering related attributes
<userdata name="some_1"></userdata>	global userdata
<row id="11">	
<userdata name="some_1">value_1</userdata>	row specific userdata
<cell> first column data </cell>	
<cell> second column data </cell>	
</row>	
<row id="12" class="css1" bgColor="red" style="color:blue;">	there are few ways to set styles of row
<cell> first column data </cell>	
<cell class="css2" style="font-weight:bold;"> second column data </cell>	same for cell
</row>	
<row id="13" locked="true">	not editable row
<cell colspan="2" rowspan="1"> first column data </cell>	colspan in grid
<cell></cell>	the second cell included in colspan still present as empty tag
</row>	
<row id="14" selected="true" call="true">	auto-selected after data loading
<cell type="ro"> first column data </cell>	cell will be rendered as "ro" without relation to column type
<cell></cell>	
</row>	
</rows>	

## Full

encoding can be important

<?xml version="1.0" encoding="iso-8859-1"?>	
<rows pos="123" total_count="2000">	
<head>	optional header configuration
<column type="edn" width="150" sort="str" format="0.00">	
Column label	
</column>	
<column type="ro" width="150" sort="na" color="red">	
One more label	
</column>	
<column type="co" width="150" sort="na" id="last">Combo	
<option value="1">One</option>	
<option value="2">Two</option>	
<option value="3">Three</option>	
</column>	
<settings>	optional, only used for forcing percent based width
<colwidth>px</colwidth>	
<splitat>2</splitat>	
</settings>	
<beforeInit>	

```

<call command="enableResizing">
  <param>false,false</param>
</call>
</beforeInit>
<afterInit>
  <call command="attachHeader">
    <param>Label 2-1,Label 2-2</param>
  </call>

  <call command="attachFooter">
    <param>Footer 1,Footer 2</param>
  </call>
</afterInit>
</head>
<userdata name="some_1"></userdata>
<row id="11">
  <userdata name="some_1">value_1</userdata>
  <cell> first column data </cell>
  <cell> second column data </cell>
</row>
<row id="12" class="css1" bgColor="red"
style="color:blue;">
  <cell> first column data </cell>
  <cell class="css2" style="font-weight:bold;"> second column
data </cell>
</row>
<row id="13" locked="true">
  <cell colspan="2" rowspan="1"> first column data </cell>
  <cell></cell>
</row>
<row id="14" selected="true" call="true">
  <cell type="ro"> first column data </cell>
  <cell></cell>
</row>
<row id="15">
  <cell><![CDATA[
    <input type='button' value='any' />
  ]]></cell>
  <cell> second column data </cell>
</row>
<row id="16">
  <cell> &lt;input type='button' value='any' /&gt; </cell>
  <cell> second column data </cell>
</row>
</rows>

```

there may be as many command calls as necessary

global userdata

row specific userdata

<b>afterInit</b>	group all command calls which will be executed after grid structure initialized command from this block will be executed even if no data structure tags ( column ) defined in header										
<b>beforeInit</b>	group all command calls which will be executed before grid structure initialized command from this block will be executed <b>only</b> if full grid structure defined in XML ( column tags )										
<b>call</b>	represent call of grid command, must be placed in <i>afterInit</i> or <i>beforeInit</i> section <table> <tr> <td><b>command</b></td><td>name of grid method which will be called.</td></tr> </table> <hr/>	<b>command</b>	name of grid method which will be called.								
<b>command</b>	name of grid method which will be called.										
<b>cell</b>	represent cell of grid, value of tag will be used as value of cell in grid <table> <tr> <td><b>class</b></td><td>css class which will be attached to cell</td></tr> <tr> <td><b>colspan</b></td><td>few cells can be grouped in rowspan, please beware that other cells, which will be included in colspan still need to be present in XML support of attribute must be enabled by API call</td></tr> <tr> <td><b>rowspan</b></td><td>few cells can be grouped in rowspan, feature is mutual exclusive with colspan ( cell can has colspan and rowspan in same time ) please beware that having multiple rowspan will slowdown grid _rowspan.js extension required to activate feature</td></tr> <tr> <td><b>style</b></td><td>css style string, value mapped to style attribute of HTML cell</td></tr> <tr> <td><b>type</b></td><td>type of used cell defined per column ( setColTypes ), but each cell in grid can have independent settings, so even if you have editable column you can define specified cells as readonly or checkboxes or some other control, by using this attribute</td></tr> </table> <hr/>	<b>class</b>	css class which will be attached to cell	<b>colspan</b>	few cells can be grouped in rowspan, please beware that other cells, which will be included in colspan still need to be present in XML support of attribute must be enabled by API call	<b>rowspan</b>	few cells can be grouped in rowspan, feature is mutual exclusive with colspan ( cell can has colspan and rowspan in same time ) please beware that having multiple rowspan will slowdown grid _rowspan.js extension required to activate feature	<b>style</b>	css style string, value mapped to style attribute of HTML cell	<b>type</b>	type of used cell defined per column ( setColTypes ), but each cell in grid can have independent settings, so even if you have editable column you can define specified cells as readonly or checkboxes or some other control, by using this attribute
<b>class</b>	css class which will be attached to cell										
<b>colspan</b>	few cells can be grouped in rowspan, please beware that other cells, which will be included in colspan still need to be present in XML support of attribute must be enabled by API call										
<b>rowspan</b>	few cells can be grouped in rowspan, feature is mutual exclusive with colspan ( cell can has colspan and rowspan in same time ) please beware that having multiple rowspan will slowdown grid _rowspan.js extension required to activate feature										
<b>style</b>	css style string, value mapped to style attribute of HTML cell										
<b>type</b>	type of used cell defined per column ( setColTypes ), but each cell in grid can have independent settings, so even if you have editable column you can define specified cells as readonly or checkboxes or some other control, by using this attribute										
<b>column</b>	in case of configuration from XML it represent column of grid value of tag used as column label <table> <tr> <td><b>align</b></td><td>Horizontal align of column ( left, right, center ; equal to setColAlign )</td></tr> <tr> <td><b>color</b></td><td>Color of column ( equal to setColumnColor )</td></tr> <tr> <td><b>format</b></td><td>For "edn" and "calendar" excells format can be specified ( equal to calling setNumberFormat or setDateFormat against related column )</td></tr> </table>	<b>align</b>	Horizontal align of column ( left, right, center ; equal to setColAlign )	<b>color</b>	Color of column ( equal to setColumnColor )	<b>format</b>	For "edn" and "calendar" excells format can be specified ( equal to calling setNumberFormat or setDateFormat against related column )				
<b>align</b>	Horizontal align of column ( left, right, center ; equal to setColAlign )										
<b>color</b>	Color of column ( equal to setColumnColor )										
<b>format</b>	For "edn" and "calendar" excells format can be specified ( equal to calling setNumberFormat or setDateFormat against related column )										

Will not affect any other column types.

<b>id</b>	Each column has ID, it has sense when you are using movable columns and column index not consistent enough While grid API uses indexes for all operations it provides API to convert IDs to indexes and back ( getColumnId, getColIndexById )
<b>sort</b>	Allows to set sorting type ( equal to setColSorting command - str,int,date) If you need to set custom routine just set name of javascript function here which will be used for such task. ( Its name must have more than 4 chars )
<b>type</b>	Default column type ( equal to setColTypes - ro,ed,co and etc. )
<b>width</b>	Width of column in percents or pixels, need to be specified as pure int, without ending px By default treated as width in pixels, can be switched to percents by using "setting/colwidth" tag

---

<b>colwidth</b>	Allow to specify how treat column width specified in "column@width". Value of tag can be <ul style="list-style-type: none"><li>• px - width will be set as pixels ( default value, tag may be skipped in such case )</li><li>• % - width will be set as percents</li></ul>
<b>head</b>	optional tag, used for grouping configuration tags ( necessary only if you want to define configuration in XML ) no attributes supported
<b>option</b>	Allow to define list of options for excell which can work with such conception ( co, coro, clist, in theory any custom excell which has inner combo property ) Content of tag used as option label "value" attribute is mandatory
<b>value</b>	value for related option inside combobox

---

**param** represent parameter of method call, there must be as much param tags as necessary

**row** tag represent row of grid, one tag for each row

**bgColor** the color of row, it is strongly recommended to use this parameter to set color of row ( all other ways

will override selection colors )

<b>class</b>	name of css class which will be attached to row
<b>id</b>	unique identifier, can be any combination of chars numbers
<b>locked</b>	row will be locked - switched to read-only state, can be reverted by JS API
<b>selected</b>	after data loading, row will be pre-selected
<b>selected</b>	has sense only with enabled "selected" attribute, if both attributes set to true - row will be selected and onSelectRow event will be called
<b>style</b>	string with css definitions, which will be set as style attribute of related row

---

## rows

just top xml tag, has no special meaning

### *Attributes*

<b>pos</b>	<p>In case of dynamic smart rendering XML fetched from server by chunks, the "pos" attribute shows in which place of grid data from incoming XML must be inserted.</p> <p>In most cases ( the exceptions are pretty specific ) this value will be equal to url parameter "posStart" ( which automatically added to all calls in SRND mode )</p> <p>Attribute has meaning only in SRND mode and will be ignored in any other.</p>
<b>total</b>	<p>Both dynamic paging and dynamic rendering modes require to specify maximum expected count of rows ( paging can work without such limit, but it will not show all available pages from the start in such case )</p> <p>Because hardcoding count or rows in javascript code is rare possible, it can be specified directly in XML.</p> <p>If value skipped in SRND mode - the total count set to size of incoming XML ( basically dynamic smart rendering switch to static smart rendering)</p> <p>If value skipped in paging mode - grid will request additional data from server until empty XML set will not be returned.</p> <p>Attribute has no meaning in non dynamic modes.</p>

---

## settings

Group grid configuration settings, has no individual meaning

**splitat**

Execute splitAt command in required position, functionality is kind of deprecated, and it recommended to use afterInit/call to initiate split

**userdata**

To store some additional data, which need not to be displayed, but may be useful for some client side operations - userdata sections can be used.

**name**

Because there may be many userdata tags per row|grid, they differ by name attribute. Value of name attribute used on client side to get related value.

---