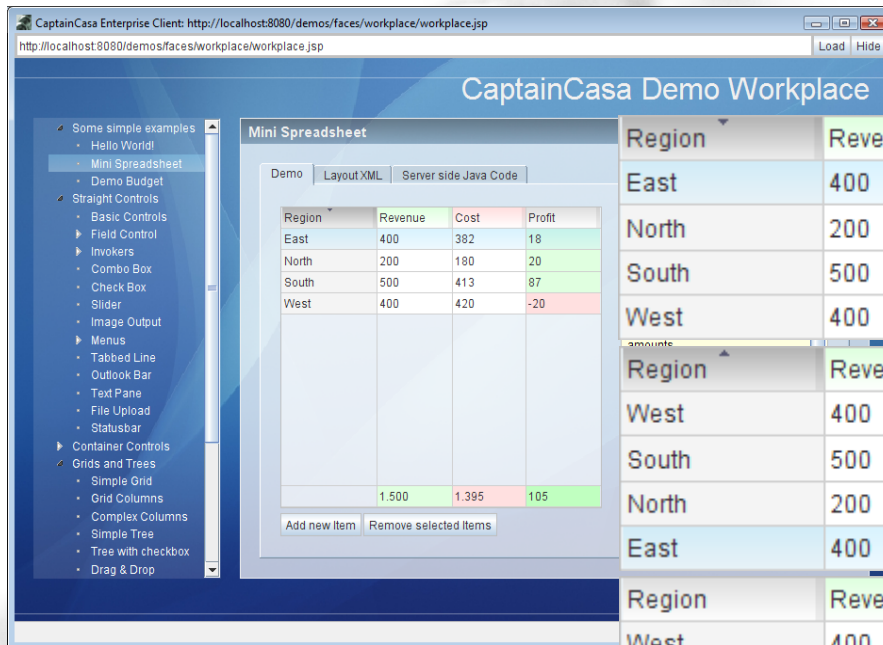


Grid Sorting



Example I - Grid



Region	Revenue	Cost	Profit
East	400	382	18
North	200	180	20
South	500	413	87
West	400	420	-20

Region	Revenue	Cost	Profit
West	400	420	-20
South	500	413	87
North	200	180	20
East	400	382	18

Region	Revenue	Cost	Profit
West	400	420	-20
East	400	382	18
North	200	180	20
South	500	413	87

Region	Revenue	Cost	Profit
South	500	413	87
North	200	180	20
East	400	382	18
West	400	420	-20

Sorting is data type aware

Example II - Tree

The screenshot shows a web application titled "CaptainCasa Demo Workplace" with a "Budget Calculation" section. A table lists various organizations and their associated values across different categories. A tree view is overlaid on the table, showing a hierarchy of nodes. The tree view is currently expanded to show the "Production" node, which is further expanded to show sub-nodes like "CC 1.0" through "CC 1.7".

Organization	Staff	Office	IT	Other	Leis...	Pleas...	Cars	Donuts
Board	0	0	0	0	0	0	0	0
Development	0	0	0	0	0	0	0	0
Marketing	0	0	0	0	0	0	0	0
Production	0	0	0	0	0	0	0	0
- CC 1.0	0	0	0	0	0	0	0	0
- CC 1.1	0	0	0	0	0	0	0	0
- CC 1.2	0	0	0	0	0	0	0	0
- CC 1.3	0	0	0	0	0	0	0	0
- CC 1.4	0	0	0	0	0	0	0	0
- CC 1.5	0	0	0	0	0	0	0	0
- CC 1.6	0	0	0	0	0	0	0	0
- CC 1.7	0	0	0	0	0	0	0	0
Sales	0	0	0	0	0	0	0	0

The tree view overlay shows the following structure:

- Organization
 - Board
 - Development
 - Marketing
 - Production
 - CC 1.0
 - CC 1.1
 - CC 1.2
 - CC 1.3
 - CC 1.4
 - CC 1.5
 - CC 1.6
 - CC 1.7
 - Sales

Sorting on each tree level

Default Sorting - Setup

Region	Revenue	Cost	Profit
East	400	382	18
North	200	180	20
South	500	413	87
West	400	420	-20

```
<t:fixgrid id="g_5" ...>
  <t:gridcol id="g_6" sortreference=".{region}" text="Region" width="100" >
    <t:field id="g_7" text=".{region}" />
  </t:gridcol>
  <t:gridcol id="g_8" sortreference=".{revenue}" text="Revenue" width="34%" >
    <t:formattedfield id="g_9" format="double" value=".{revenue}" .../>
  </t:gridcol>
  <t:gridcol id="g_10" sortreference=".{cost}" text="Cost" width="33%" >
    <t:formattedfield id="g_11" format="double" value=".{cost}" .../>
  </t:gridcol>
  <t:gridcol id="g_12" sortreference=".{profit}" text="Profit" width="33%" >
    <t:formattedfield id="g_13" format="double" value=".{profit}" .../>
  </t:gridcol>
  <t:gridfooter id="g_14" >
    <t:coldistance id="g_15" />
    <t:formattedfield id="g_16" .../>
    <t:formattedfield id="g_17" .../>
    <t:formattedfield id="g_18" .../>
  </t:gridfooter>
</t:fixgrid>
```

Just specify
the property
to use for
sorting within
the grid column
definition

Features

- Default - no explicit coding required, sorting is added to a grid „by definition“
- Sorting is done on server side - the client only knows a part of the whole grid
- Sorting is data type aware
- Sorting in a tree will sort every level of the tree
- Multiple extension points for adding own sorting rules (e.g. re-read grid data from database when sorting)