



BEAR TABLE

Custom widget

September 19 2016

SOMMAIRE

| | |
|--|----|
| 1. INTRODUCTION | 3 |
| 2. PROPERTIES | 4 |
| 3. VALUE | 5 |
| 4. INFORMATION IN THE RECORD | 6 |
| 5. HEADERLIST | 7 |
| 5.1. Field Control | 11 |
| 5.2. Control Select | 14 |
| 5.3. Chart | 14 |
| 5.4. how to set a list of values | 15 |
| 6. RULES | 16 |
| 6.5. Sumcol | 16 |
| 6.6. Uniquocol | 16 |
| 7. ACTIONS | 17 |
| 8. CHILDREN MANAGEMENT | 19 |
| 8.7. Data | 19 |
| 8.8. Control the display | 21 |

1. Introduction

The document explains how to use the widget.

2. Properties

The different properties are:

| properties | Description |
|-------------------|---|
| Value | Value of the table. A JSON variable |
| Table CSS Classes | CSS class to apply on the table. This information is use in the HTML in the tag "class=" |
| Table CSS Style | CSS Style to apply on the table, for local use. This information is use in the HTML in the tag "style=" |
| HeaderList | Control which columns are to be display, and control on different column |
| Pagination | Specify the pagination of the data |
| DynamicList | JSON variable. Columns control may be Select box. The value of the list can be giving as constant in the HeaderList, or can reference an external variable. This property groups all dynamic list |
| showDebug | Boolean. If yes, then information are displayed on the screen |
| Label | Label of the widget |
| ChecksRules | JSON variable. Edition rule can be specifying on the table, like one value must be unique in the table. This properties reference all rules. |

3. Value

JSON variable. Value of the widget. This is a list of Set, one set is a line in the table. The set contains the key (column key), and the value.

Example, to display 5 lines:

```
[
  {
    "country": "France",
    "browser": "FIREFOX",
    "market": 25,
    "system": "windows",
    "processor": "4"
  },
  {
    "country": "France",
    "browser": "IE",
    "market": 30,
    "system": "windows",
    "processor": "2"
  },
  {
    "country": "France",
    "browser": "CHROME",
    "market": 40,
    "system": "windows"
  },
  {
    "country": "Germany",
    "browser": "IE",
    "market": 40,
    "system": "windows"
  },
  {
    "country": "Germany",
    "browser": "Firefox",
    "market": 50,
    "system": "windows"
  }
]
```

4. Information in the record

Some action can be override in the record.

Then if for one specific record, when a specific cell has to be display, a value with the prefix exist, it is apply

| Attribut | Description | Example |
|-----------|---------------------------|---|
| _style | Apply a style | "country_style" : "background-color: #4CAF50" |
| _readonly | Set the field in readOnly | Country_readonly : true |

5. HeaderList

The header is a JSON information. This information controls the different columns to be display, and the control on the columns.

The list is a list of set. Each set is a column. In the example, we give the way to display the column Browser, which is a list of value.

| Attribut | Description | Example |
|---|--|--|
| Title | Title of the column | "Title" : "Main browser" |
| Name | Name. The name is use to find the value of the column in the Value list | "name" : "browser" |
| Filter | If true, then RichTable display a input, to let the user filter the value of the line | "Filter" : false |
| Order | If True, the user can order the lines by the value of this attribut | "order" : "true" |
| Control | Control on the field. See below the different available control | "control" : "select" |
| Styleheader | Style applies to the header. | "styleheader" : "background-color:yellow", |
| Stylecol | Style apply to the column, and only on the column, (note: the record can contains a _style too) | "stylecol" : "background-color:green", |
| Tips | Add a info bull (tooltips) on the widget | "tips" : "Give the first day of working day", |
| Size [content=input] [content=number] | Size of the field on screen. Value is not limited. | "size":10 |
| Footer [control=number] | If this property is set, then a new footer line is added, with the command. Available command are - "sum" (sum the column) | "footer":"sum" |
| Listoptionkey [control=select] | When the control is a select, this give the key use to save the value. List contains sets, then this information specify which attribute used in the list. | "listoptionkey":"key" |
| Listoptiondisplay [control=select] | This one which attributes used in the list for the value to the user | "listoptiondisplay":"display", |
| Listoptions [control=select] | In case of list static, the contents | listoptions": [{ "key": "IE", "display": "Internet Explorer" }, { "key": "FIREFOX", "display": "Firefox Mozilla" }, { "key": "CHROME", "display": "Chrome" }, { "key": "SMARTBROWSER", "display": "SmartBrowser" }, { "key": "SAFARI", "display": "Safari" }, { "key": "MOSAIC", "display": "Mosaic" }, { "key": "OPERA", "display": "Opera" }, { "key": "MOZILLA", "display": "Mozzila" }] |
| Listoptionsvariable [control=select] | In case of list dynamic, the name of the list in the DynamicList properties | "listoptionsvariable": "listprocessor" |
| Readonly [control=select] | Is the value equals true, then the select are in read only | |
| colspan | Number of columns. 1 per default | |



| | | |
|----------|--|--|
| subtitle | If set, then a subtitle is displayed in the cell | |
| parent | See children management | |
| isDebug | Widget send debug information to the console.log and in the html | |


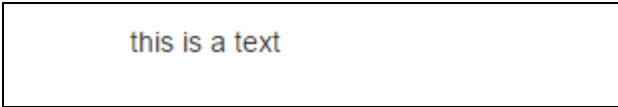
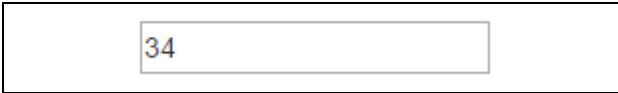
Example:

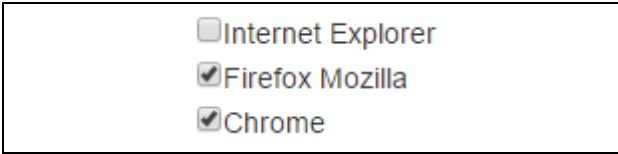
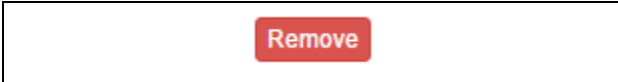

```
[
  {
    "title": "myNumber",
    "name": "mynumber",
    "filter": true,
    "order": true,
    "control": "number",
    "footer": "sum"
  },
  {
    "title": "Country",
    "name": "country",
    "filter": true,
    "order": true,
    "control": "input",
    "size": 10
  },
  {
    "title": "processor",
    "name": "processor",
    "filter": true,
    "order": true,
    "control": "select",
    "listoptionkey": "key",
    "listoptiondisplay": "value",
    "listoptionsvariable": "listprocessor"
  },
  {
    "title": "browser",
    "name": "browser",
    "filter": false,
    "order": false,
    "control": "select",
    "listoptionkey": "key",
    "listoptiondisplay": "display",
    "listoptions": [
      { "key": "IE", "display": "Internet Explorer" },
      { "key": "FIREFOX", "display": "Firefox Mozilla" },
      { "key": "CHROME", "display": "Chrome" },
      { "key": "SMARTBROWSER", "display": "SmartBrowser" },
      { "key": "SAFARI", "display": "Safari" },
      { "key": "MOSAIC", "display": "Mosaic" },
      { "key": "OPERA", "display": "Opera" },
      { "key": "MOZILLA", "display": "Mozzila" }
    ]
  },
  {
    "title": "market",
    "name": "market",
    "filter": true,
    "order": true,
    "control": "text"
  },
  {
    "title": "system",
    "name": "system",
    "filter": true,
    "order": true,
    "control": "button"
  },
  {
    "title": "mydatetime",
    "name": "mydatetime",
    "filter": true,
    "order": true,
    "control": "datetime"
  },
  {
    "title": "hour",
```

```
    "name": "myhour",  
    "filter": true,  
    "order": true,  
    "control": "hour"  
  },  
  {  
    "title": "password",  
    "name": "password",  
    "filter": true,  
    "order": true,  
    "control": "password"  
  }  
]
```

5.1. Field Control

| Control | Description | Example |
|------------|--|---|
| link |  <p>Display a link</p> <p>“title” : title of the Column</p> <p>“name” : content if the link is retrieved by the record[name]</p> <p>“labellink” : name if the link (constant)</p> <p>“labellinkdynamic” : get the content of [labellinkdynamic].</p> <p>“tips”: display when the mouse in on the link</p> <p>“target”: target of the link (use ‘blank’ to display in another window)</p> <p>“cssclass” : set up the class. If empty, a HREF link is display, use “btn btn-primary btn-xs” to display the link as a button</p> | <pre>{ "title": "Link (button class) WebSite", "name": "Vendor site", "tips" : "click here to access the WebSite", "namelink": "Vendor site", "control": "link", "target": "blank", "cssclass" : "btn btn- primary btn-xs" },</pre> |
| button | Reserved | |
| select |  <p>A select box. Only one value can be use. See the “how to set a list of value” to have an explanation on the list</p> | <pre>{ "title": "browser", "name": "browser", "control": "select", "listoptionkey": "key", "listoptiondisplay": "display", "listoptions": [{ "key": "IE", "display": "Internet Explorer" }, { "key": "FIREFOX", "display": "Firefox Mozilla" }, { "key": "CHROME", "display": "Chrome" }, { "key": "SMARTBROWSER", "display": "SmartBrowser" }, { "key": "SAFARI", "display": "Safari" }, { "key": "MOSAIC", "display": "Mosaic" }, { "key": "OPERA", "display": "Opera" }, { "key": "MOZILLA", "display": "Mozzila" }] }</pre> |
| Textselect | A select can be “disabled” or, to not have the select box | |

| | | |
|-------------|--|--|
| | display, this control displays the result as a Text. Then, record contains the key, display is the label. Exact same control as the select is used, only the display change. | |
| input |  <p>A text input, on one line</p> | <pre>{ "title": "City", "name": "mycity", "control": "input" }</pre> |
| textArea | A TextArea inputs. Cols and rows have to be used | <pre>{ "title": "conclusion", "name": "conclusion", "control": "textarea", "cols": 40, "rows": 5 }</pre> |
| hidden | The column is hidden (for a usage in a javascript for example). | |
| text |  <p>A text in read only</p> | <pre>{ "title": "city", "name": "mycity", "control": "text", }</pre> |
| number |  <p>A test input to select a number, an HTML5 format</p> | <pre>{ "title": "myNumber", "name": "mynumber", "control": "number", },</pre> |
| password | A text input, but the display is hide | |
| email | A text input, with a HTML5 format to Email. | |
| date | | |
| datetime | | |
| time | | |
| hour | | |
| contentjson | | |
| chart | Define a Chart. Different chart are possible, and value can be give directly or by an additional variable | <pre>{ "title": "ratio", "name": "chart", "control": "chart", "typechart": "Doughnut", "listdata": [65,59,80,81,54], "listlabels": ["January", "February", "March", "April", "May"] },</pre> |

| | | |
|--------------|---|---|
| checkbox | A checkbox, match to a Boolean | |
| checkboxlist |  <p>Like a select, but all the different options are visible, and user can select multiple value. So the result is a list of value.</p> <p>To set the list of value, see “how to set a list of values”</p> | <pre>{ "title": "checkList", "name": "MyCheckList", "filter": false, "order": false, "control": "checkboxlist", "listoptionkey": "key", "listoptiondisplay": "display", "listoptions": [{ "key": "IE", "display": "Internet Explorer" }, { "key": "FIREFOX", "display": "Firefox Mozilla" }, { "key": "CHROME", "display": "Chrome" }] }</pre> |
| children | See the children management | |
| remove |  <p>Remove the current line. Then, a button is display on the line, and when the user clicks on, the current line is removed.</p> <p>confirmation (default is false) : Boolean. If yes, then a confirmation is asked to the user</p> <p>confirmationmessage: change the default message</p> <p>Tips: info bubble on the button</p> <p>The readonly condition can be set in the record. If the button is “readonly”, then it’s disabled.</p> <p>Parent: if the record is part of children, then the parent has to be defined to explore the sub data.</p> | <pre>{ "control": "remove", "confirmation": true, "confirmationmessage": "Hey Guy, Do you want to delete the current line ?" }</pre> |
| insert |  <p>Insert a line just before this line. To insert after, use the Add control.</p> <p>Defaultvalue: the value to add.</p> <p>Tips: info bubble on the button</p> <p>The readonly condition can be set in the record. If the button is “readonly”, then it’s disabled.</p> | <pre>{ "control": "insert", "tips": "You add a new value BEFORE this line", "defaultvalue" : { "browser": "SMARTBROWSER", "mynumber": 12 } }</pre> |

| | | |
|-----|--|--|
| add | <div data-bbox="261 250 331 295">Add</div> <p>Insert a line just after this line. To insert before, use the insert control.</p> <p>Defaultvalue: the value to add. Example:</p> <pre>{ "country": "Italy", "city" : "Firenze" }</pre> <p>cssClass : class to apply to the button</p> <p>btnLabel : label of the button (else Add by default).</p> <p>addChild : if defined, the defaultValue is added as a child of the record, under the given name.</p> <p>Example: addChild: "actionsChild",</p> <p>Then, the JSON will be</p> <pre>{ "recordWhereisTheAdd" : "Hello", "actionChilds" : [{ "country": "Italy", "city" : "Firenze" }],</pre> <p>Tips: info bubble on the button</p> <p>The readonly condition can be set in the record. If the button is "readonly", then it's disabled.</p> | <pre>{ "control":"add", "tips" : "You add a new value AFTER this line", "defaultvalue" : { "browser": "MOSAIC", "mynumber": 15 } }</pre> |
|-----|--|--|

5.2. Control Select

A list of value in a select box. See 5.4 *how to set a list of values* to give the list of option in the widget.

5.3. Chart

The chart can be different type and are grouped on the parameter "typechart"

- “typechart” : type of chart, may be “doughnut”

List Data depends of the type of chart.

```
"typechart": "Doughnut",
"listdata": [65, 59, 80, 81, 54],
"listlabels": ["January", "February", "March", "April", "May"]
```

1. “showlegend” : true by default.

Reference: <http://itblin.github.io/angular-chart.js/>

5.4. how to set a list of values

Different widgets accept a list of values.

The different ways to build this list:

- static: give a attributes “listoptions”, a JSON array of record, like

```
"listoptions":
[ { "key": "IE",
  "display": "Internet Explorer" },
  { "key": "FIREFOX",
    "display": "Firefox Mozilla" },
  { "key": "CHROME",
    "display": "Chrome" }
],
"listoptionkey" : "key",
"listoptiondisplay" : "display"
```

According the widget, an attribute give the key and the display: for example, in the checkboxlist, the attribute “listoptionkey” return the JSON attribute to return as the internal value, and the attribute “listoptiondisplay” return the attribute to display.

- Dynamic
Use the listoptionvariable to give the name of the variable where the list is saved. Listoptionkey and listoptionvariable describe the information to use to display the option and the key to return.

```
"listoptionsvariable": "alllists.browser",
"listoptionkey" : "key",
"listoptiondisplay" : "display"
```

Where alllists is a different JSON / Javascript calculated, for example

```
"alllist":
"browser":
[ { "key": "IE",
  "display": "Internet Explorer" },
  { "key": "FIREFOX",
    "display": "Firefox Mozilla" },
  { "key": "CHROME",
    "display": "Chrome" }
],
```

6. Rules

Rules are verified immediately when user change a value. In case of error, then a message is displayed.

6.5. Sumcol

The column is summarize, and control to the value. If the value is not the same, an error is displayed.

6.6. Uniquecol

The column is check, and each line must have a null or a different value.

Example

```
[
{
  "name" : "SumOfNumber",
  "rule" : "sumcol",
  "colname" : "mynumber",
  "totalsum" : 100,
  "message": "The total must be 100"
},
{
  "name" : "Check Unique country",
  "rule" : "uniquecol",
  "colname" : "country",
  "message": "Each line should have a different country"
}
]
```


7. Actions

Actions are use to do some controles on the line, like set line in a color, or in read only.

Different actions actions, to apply style (styleline), set in readonly, or hide the line.

```
[
  { "name": "greenline",
    "action": "styleline",
    "attribut" : "status",
    "valueattribut": "close",
    "variable": "showstyleline",
    "valuevariable": true,
    "style": "background-color: #4CAF50;readonly: true"
  },
  { "name": "readonly",
    "action": "readonlyline",
    "attribut" : "status",
    "valueattribut": "close",
    "style": "background-color: #4CAF50;"
  },
  { "name": "showhistory",
    "action": "hideline",
    "variable" : "showhistory",
    "valuevariable" : true,
    "attribut" : "status",
    "valueattribut": "close"
  },
  { "name": "sort",
    "action": "sort",
    "sort1" : "mynumber",
    "sort2" : "country"
  }
]
```

Different actions are :

| Control | Description | Example |
|--------------|--|---|
| sort | Sort the data. This action is done only at the first time sort1 and sort2 are the two attributes name | <pre>{ "name": "sort", "action": "sort", "sort1" : "mynumber", "sort2" : "country" }</pre> |
| Styleline | <p>Apply a style on the line.</p> <p>Variable and valuevariable is use to get and information from the properties dynamicLists.</p> <p>Example : if valuevariable is “showstyleline”, then the dynamicLists. showstyleline is read, and compare to the valuevariable.</p> <p>Attribute and valueattributes reference attributes on the record.</p> | <pre>{ "name": "greenline", "action": "styleline", "attribut" : "status", "valueattribut": "close", "variable": "showstyleline", "valuevariable": true, "style": "background-color: #4CAF50; " },</pre> |
| readonlyline | Apply a read only on the line. Same controls “variable”, “variablevalue”, “attribute”, “attributevalue” are used | |
| hideline | Hide the line. Same controls “variable”, “variablevalue”, “attribute”, “attributevalue” are used | |

8. Children management

8.7. Data

The JSON data can have some children, like:

```
[
  {
    "firstname": "Walter",
    "lastname": "Bates",
    "details": { "ssn": "5553355" },
    "period": [
      { "type": "RTT", "nbday": 24 },
      { "type": "VAC", "nbday": 24 },
      { "type": "RECUP", "nbday": 24 }
    ]
  },
  {
    "firstname": "Helen",
    "lastname": "Kelly",
    "details": { "ssn": "2222" },
    "period": []
  },
  {
    "firstname": "Jan",
    "lastname": "Fisher",
    "details": { "ssn": "4444" },
    "period_bearwidget": true,
    "period": [
      { "type": "RTT", "nbday": 24 },
      { "type": "RECUP", "nbday": 24 }
    ]
  }
]
```

The Bear table can display the list of children, one line per children. Only one depth can be managed at this time.

To do that, in the Header, the attribute “**parent**” should be used. When the header contains :

```
{
  "title": "Type",
  "parent": "period",
  "name": "type",
  "control": "input",
  "stylecol": "background-color: #dff0d8"
},
```

The following management is done:

- In the data, for each record, children with the name “period” are search. For each line, the widget displays a new line in the array. In this example, the first record “Walter” contains 3 periods: 3 lines will be displayed

| | First name | last name |
|-------------------------------------|------------|-----------|
| <input checked="" type="checkbox"/> | Walter | Bates |
| | RTT | 24 |
| | VAC | 24 |
| | RECUP | 24 |
| <input type="checkbox"/> | Helen | Kelly |
| <input checked="" type="checkbox"/> | Jan | Fisher |
| | RTT | 24 |
| | RECUP | 24 |

- The list of headers for parent is used to display the line with this type. With the following header

```
[
  {
    "title": "",
    "name": "period",
    "control": "children",
    "styleheader": "background-color: #007DB3",
    "stylecol": "background-color: #F2DEDE",
    "tips": "Click here to display the period attached"
  },
  {
    "title": "First name",
    "name": "firstname",
    "control": "text",
    "styleheader": "background-color: #007DB3",
    "stylecol": "background-color: #F2DEDE"
  },
  {
    "title": "last name",
    "name": "lastname",
    "control": "text",
    "styleheader": "background-color: #007DB3",
    "stylecol": "background-color: #F2DEDE"
  },
  {
    "title": "",
    "parent": "period",
    "name": "margin",
    "control": "text"
  },
  {
    "title": "Type",
    "parent": "period",
    "name": "type",
    "control": "input",
    "stylecol": "background-color: #dff0d8"
  },
  {
    "title": "nb days",
    "parent": "period",
    "name": "nbday",
    "control": "text",
    "stylecol": "background-color: #dff0d8"
  }
]
```

The header “type”, “nbDays” and the first header (empty title) is used to display the second type of line. Note the header to give a margin.

8.8. Control the display

Using a header “children”, it’s possible to control by the user the visibility of the line.

The control must be “children” and the name is the name of the child in JSON.

In the example, the header

```
{  "title": "",
    "name": "period",
    "control": "children",
    "styleheader": "background-color: #007DB3",
    "stylecol": "background-color: #F2DEDE",
    "tips": "Click here to display the period attached"
},
```

Display a check box. When the user selects it, the periods attached to this line are displayed.

It’s possible to give a default values to the display: this is the “<name of data>_bearwidget”, and it must be a Boolean.

In the data,

```
{
  "firstname": "Jan",
  "lastname": "Fisher",
  "details": { "ssn": "4444" },
  "period_bearwidget": true,
  "period": [
    { "type": "RTT", "nbday": 24 },
    { "type": "RECUP", "nbday": 24 }
  ]
}
```

The value is set to true: by default, the line is displayed (if the value is not set, then sub-line are hidden)

HEADQUARTERS

GRENOBLE, FRANCE

32, rue Gustave Eiffel

38000 Grenoble

EMEA, ASIA & LATIN AMERICA

PARIS, FRANCE

73-77, rue de Sèvres

92100 Boulogne-Billancourt

NORTH AMERICA

SAN FRANCISCO, USA

51 Federal St. Suite 305

San Francisco, CA 94107



www.bonitasoft.com