



HPE Unified OSS Console

Integration Guide

Release 2.3.0

Part number: JP699AAE

Edition: 1.0



Hewlett Packard
Enterprise

Notices

Legal notice

© Copyright 2016 Hewlett Packard Enterprise Development LP

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the US

Trademarks

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Adobe®, Acrobat® and PostScript® are trademarks of Adobe Systems Incorporated.

Microsoft®, Internet Explorer, Windows®, Windows Server®, and Windows NT® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Firefox® is a registered trademark of the Mozilla Foundation.

Google Chrome® is a trademark of Google Inc.

Oracle® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

Red Hat® is a registered trademark of the Red Hat Company.

Linux® is a registered trademark of Linus Torvalds in the U.S. and other countries.

Apache CouchDB, CouchDB, and the project logo are trademarks of The Apache Software Foundation

Node.js project. Joyent® and Joyent's logo are registered trademarks of Joyent, Inc

Redis®, and the Redis logo are the trademarks of Salvatore Sanfilippo in the U.S. and other countries

Contents

Notices	1
Preface	6
About this guide	6
Audience	6
Software Versions.....	6
Associated Documents.....	6
Support.....	7
Chapter 1 – HPE Widgets Customization	8
1.1 HPE Highcharts	8
1.1.1 Overview.....	8
1.1.2 JSON schema	8
1.2 HPE Highmaps.....	9
1.2.1 Overview.....	9
1.2.2 JSON schema	10
1.3 HPE Knob Gauge.....	10
1.3.1 Overview.....	10
1.3.2 JSON schema	10
1.4 HPE Action.....	15
1.4.1 Overview.....	15
1.4.2 JSON schema.....	15
1.4.3 Example.....	15
1.4.3.1 Global configuration	15
1.4.3.2 Display Modes.....	20
1.5 HPE Table.....	23
1.5.1 Overview	23
1.5.2 JSON schema.....	23
1.5.3 Example.....	32
1.5.3.1 Global configuration.....	32
1.5.3.2 Column configuration.....	46
1.6 HPE Aggregation Table.....	65
1.6.1 Overview.....	65
1.6.2 JSON schema.....	65
1.7 HPE Top Table	69
1.7.1 Overview.....	69
1.7.2 JSON schema	69
1.8 HPE Breadcrumb.....	73
1.8.1 Overview.....	73
1.8.2 JSON schema.....	73
1.9 HPE Time Selector.....	74
1.9.1 Overview.....	74
1.9.2 JSON schema.....	74
1.10 HPE Drilldown.....	79
1.10.1 Overview	79
1.10.2 JSON schema	79
1.11 HPE Data Exchange Inspector.....	82
1.11.1 Overview	82
1.11.2 JSON schema.....	82
1.12 HPE Form	83
1.12.1 Overview.....	83

1.12.2 JSON schema.....	83
1.13 HPE Search.....	85
1.13.1 Overview.....	85
1.13.2 JSON schema.....	85
1.14 HPE Launch Tree.....	87
1.14.1 Overview.....	87
1.14.2 JSON schema.....	87
1.15 HPE Localization Inspector.....	89
1.15.1 Overview.....	89
1.15.2 JSON schema.....	89
1.16 HPE Meter.....	91
1.16.1 Overview.....	91
1.16.2 JSON schema.....	91
1.17 HPE Navigation Toolbar.....	98
1.17.1 Overview.....	98
1.17.2 JSON schema.....	98
1.18 HPE Web Content.....	101
1.18.1 Overview.....	101
1.18.2 JSON schema.....	101
1.19 HPE Widget Navigation.....	109
1.19.1 Overview.....	109
1.19.2 JSON schema.....	109
1.20 HPE Card.....	111
1.20.1 Overview.....	111
1.20.2 JSON schema.....	111
1.21 HPE Iframe.....	119
1.21.1 Overview.....	119
1.21.2 JSON schema.....	119
1.22 HPE Notifications.....	121
1.22.1 Overview.....	121
1.22.2 JSON schema.....	121
1.23 HPE Notifications Table.....	123
1.23.1 Overview.....	123
1.23.2 JSON schema.....	123
1.24 HPE Notifications Generator.....	130
1.24.1 Overview.....	130
1.24.2 JSON schema.....	130
1.25 HPE Notifications Keywords.....	132
1.25.1 Overview.....	132
1.25.2 JSON schema.....	132
1.26 HPE Chat.....	134
1.26.1 Overview.....	134
1.26.2 JSON schema.....	134
1.27 HPE Leaflet.....	136
1.27.1 Overview.....	136
1.27.2 JSON schema.....	136
1.28 HPE Widget Menu Bar.....	143
1.28.1 Overview.....	143
1.28.2 JSON schema.....	143
1.29 HPE Widget Tab.....	145
1.29.1 Overview.....	145
1.29.2 JSON schema.....	145

List of figures

Figure 1 – Example of a Highchart widget.....	8
Figure 2 – Example of hpe-ng-highmaps widget– Colored Map (with/without bubble chart option).....	9
Figure 3 – Widget knob gauge.....	10
Figure 4 - Example of aggregation table (multiple queries).....	65
Figure 5 - Example of top table (Top 10 of Network QOE score by Cell Name).....	69
Figure 6 – Example of widget breadcrumb.....	73
Figure 7 – Widget Time Selector.....	74
Figure 8 – Expanding QoE score with the Drill up and down widget.....	79
Figure 9 – Data Exchange Inspector.....	82
Figure 10 – Widget Form.....	83
Figure 11 - Widget Search.....	85
Figure 12 – Widget launch tree.....	87
Figure 13 - Widget Localization.....	89
Figure 14 – Example of meter with a reference fact (stacked on left).....	91
Figure 15 - Example of a Navigation Toolbar widget.....	98
Figure 16 – Widget Web Content – Edition in place.....	101
Figure 17 – Widget Navigation.....	109
Figure 18 - Widget card.....	111
Figure 19 - Example of iframe integrating www.hpe.com web site.....	119
Figure 20 – Widget notification.....	121
Figure 21 - Widget notification table.....	123
Figure 22 – Widget notifications generator.....	130
Figure 23 – Widget Chat.....	134
Figure 24- Widget Leaflet.....	136
Figure 25 - Widget Menu Bar.....	143
Figure 26 - Widget Tab.....	145

Preface

About this guide

This user guide document describes the HPE Unified Console product.

Audience

Here are some recommendations based on possible reader profiles:

- Administrator
- Operators
- Value Pack Designer
- Dashboards / Views Designer
- Integrator and delivery teams
- Unified OSS Console Add-ons Developers

Software Versions

The term UNIX is used as a generic reference to the operating system, unless otherwise specified.

Product Version	Supported Operating systems
HPE Unified OSS Console V2.3	Red Hat Enterprise Linux Server release 6.5 Red Hat Enterprise Linux Server release 7.2

Table 1 - Software Versions

Associated Documents

The following documents contain useful reference information:

- HPE Unified OSS Console V2.3 – Release Notes
- HPE Unified OSS Console V2.3 – Installation Guide
- HPE Unified OSS Console V2.3– User Guide
- HPE Unified OSS Console V2.3– Development Guide

Support

Please visit our HP and HPE Software Support Online Web site at <https://softwaresupport.hp.com/> for contact information, and details about HPE Software products, services, and support.

The Software support area of the Software Web site includes the following:

- Downloadable documentation.
- Troubleshooting information.
- Patches and updates.
- Problem reporting.
- Training information.
- Support program information.

Chapter 1 – HPE Widgets Customization

This chapter contains details about the customizations of the widgets delivered with the UOC product.

1.1 HPE Highcharts

1.1.1 Overview

Highcharts widgets allow to have a graphical representation of a set of data. UOC supports all types of charts highcharts can propose.



NOTE: It is highly recommended to refer to the Highcharts documentation: <http://api.highcharts.com/highcharts/>

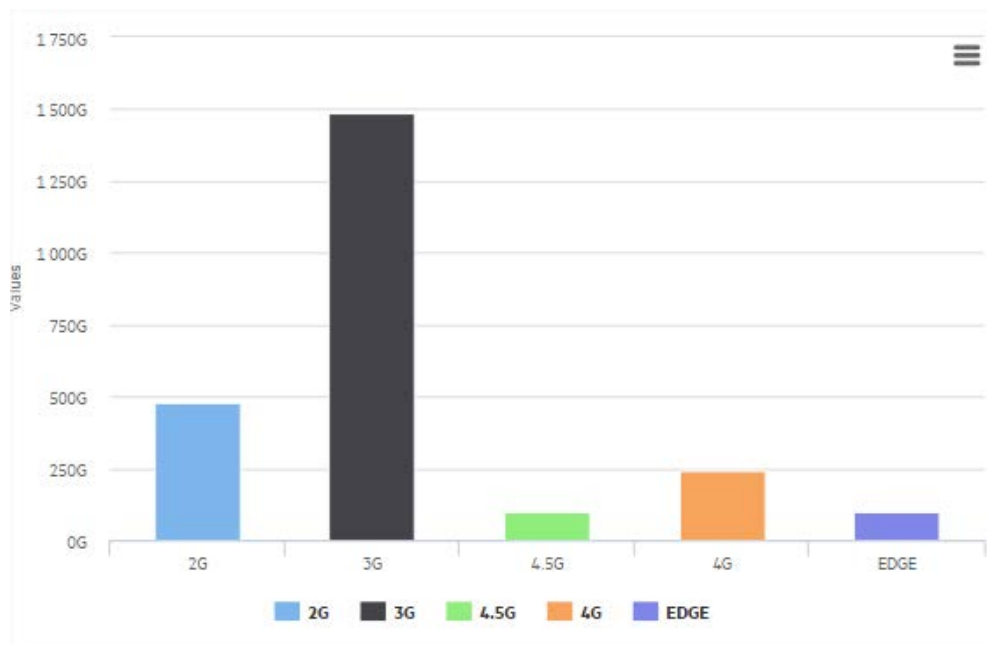


Figure 1 – Example of a Highchart widget

1.1.2 JSON schema

We didn't add the JSON schema of the hpe-ng-highcharts widgets to this documentation because of its length. However you can find the full schemas of the different types of highcharts in `<install_dir>/client/addons/hpe/widgets/hpe-ng-highcharts/hpe-chart-<chartType>.json`.

Here is the list of the different types of highcharts having a JSON schema definition:

- hpe-chart-area
- hpe-chart-area-range
- hpe-chart-area-spline
- hpe-chart-area-range-spline
- hpe-chart-bar
- hpe-chart-boxplot
- hpe-chart-bubble
- hpe-chart-column
- hpe-chart-column-range

- hpe-chart-errorbar
- hpe-chart-funnel
- hpe-chart-gauge
- hpe-chart-heatmap
- hpe-chart-line
- hpe-chart-pie
- hpe-chart-donut
- hpe-chart-pyramid
- hpe-chart-scatter
- hpe-chart-solidgauge
- hpe-chart-spline
- hpe-chart-treemap
- hpe-chart-waterfall

1.2 HPE Highmaps

1.2.1 Overview

Highmaps is a HTML5 mapping component optimized for creating schematic maps. It allows to build interactive maps to display any set of data or any other information linked to geography. This widget is able to execute a data request and use the first metric as a value to apply. This value can also optionally be used with threshold definition to display a given status or color.



NOTE: It is highly recommended to refer to the Highmaps documentation: <http://api.highcharts.com/highmaps>



IMPORTANT: Hpe-ng-highmaps widget rely on the Highmaps documentation, but we extended the configuration of the widget to support features the library does not support. You can refer to the JSON schema of the Hpe-ng-highmaps widget and look for the **geoJson** property.

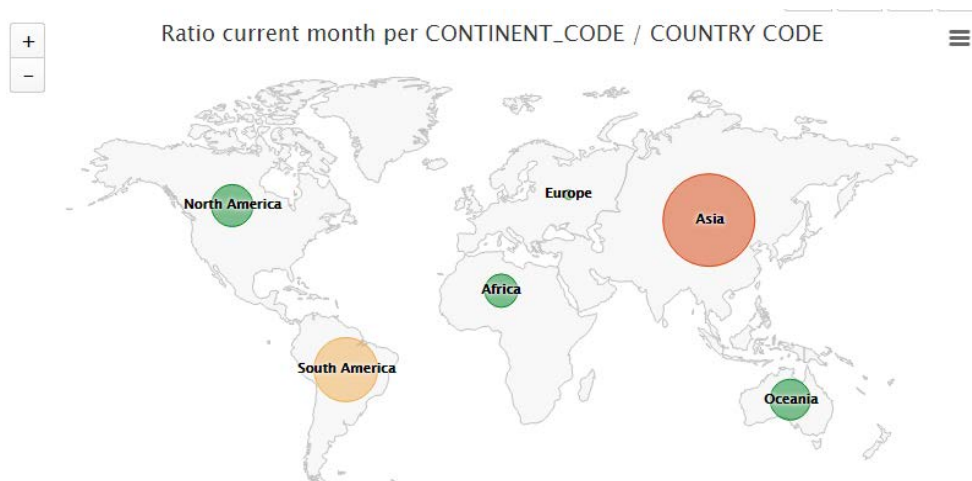


Figure 2 – Example of hpe-ng-highmaps widget– Colored Map (with/without bubble chart option)

1.2.2 JSON schema

We didn't add the JSON schema of the hpe-ng-highmaps widgets to this documentation because of its length. However you can find the full schema of the hpe-ng-highmaps widget in `<install_dir>/client/addons/hpe/widgets/hpe-ng-highmaps/hpe-ng-highmaps.json`.

1.3 HPE Knob Gauge

1.3.1 Overview

The HPE Knob Gauge is a widget allows to display a simple gauge. It is integrated with the UOC generic data fetching through data selection. As example an hpe knob gauge may be used to display the value of one or multiple facts.

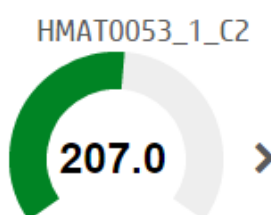


Figure 3 – Widget knob gauge

1.3.2 JSON schema

Below the JSON schema of HPE knob gauge widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-knob-gauge/hpe-knob-gauge.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Knob gauge",
  "description": "HPE knob gauge widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the knob gauge",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of hpe knob gauge widget"
        }
      },
      "options": {
        "type": "object",
```

```

"title": "Options",
"description": "Knob gauge options",
"properties": {
  "inputColor": {
    "type": "string",
    "title": "Input color",
    "description": "Input value color (red,
#FF0000...)"
  },
  "readOnly": {
    "type": "boolean",
    "title": "Readonly",
    "description": "Disable input value of the gauge",
    "default": "false"
  },
  "angleOffset": {
    "type": "integer",
    "title": "Angle offset",
    "description": "Starting angle in degrees",
    "default": "0"
  },
  "angleArc": {
    "type": "integer",
    "title": "Angle arc",
    "description": "Arc size in degrees",
    "default": "360"
  },
  "width": {
    "type": "integer",
    "title": "Width",
    "description": "Width of the gauge in pixels",
    "default": "200"
  },
  "height": {
    "type": "integer",
    "title": "Height",
    "description": "Height of the gauge in pixels",
    "default": "200"
  },
  "min": {
    "type": "string",
    "title": "Min",
    "description": "Minimum value of the gauge",
    "default": "0"
  },
}

```

```

        "max": {
            "type": "string",
            "title": "Max",
            "description": "Maximum value of the gauge",
            "default": "100"
        }
    }
},
"showValue": {
    "type": "boolean",
    "title": "Show value",
    "description": "Show input value"
},
"showUnit": {
    "type": "boolean",
    "title": "Show unit",
    "description": "Show fact unit"
},
"numberFormat": {
    "type": "string",
    "title": "Number format",
    "description": "Number of decimals"
},
"autoTitle": {
    "type": "boolean",
    "title": "Auto title",
    "description": "Automatically generate a title with facts and
dimensions"
},
"selector": {
    "type": "boolean",
    "title": "Selector",
    "description": "Display a gauge selector"
},
"multiple": {
    "type": "boolean",
    "title": "Multiple",
    "description": "Display multiple instance"
},
"thresholds": {
    "type": "object",
    "title": "Thresholds",
    "description": "Definition of thresholds per fact id",
    "properties": {

```

```

        "[fact_id]": {
            "type": "object",
            "title": "Fact id",
            "description": "Fact id is used as a key of thresholds
object ([fact_id] must be replaced by a fact id)",
            "properties": {
                "min": {
                    "type": "integer",
                    "title": "Min",
                    "description": "Minimum value for this
threshold (it overrides min value in options)"
                },
                "max": {
                    "type": "integer",
                    "title": "Max",
                    "description": "Maximum value for this
threshold (it overrides max value in options)"
                },
                "success": {
                    "type": "integer",
                    "title": "Success",
                    "description": "Success threshold"
                },
                "info": {
                    "type": "integer",
                    "title": "Info",
                    "description": "Info threshold"
                },
                "warning": {
                    "type": "integer",
                    "title": "Warning",
                    "description": "Warning threshold"
                },
                "danger": {
                    "type": "integer",
                    "title": "Danger",
                    "description": "Danger threshold"
                }
            }
        }
    },
    "newlinePerInstance": {
        "type": "boolean",
        "title": "New line per instance",

```

```
        "description": "Display an instance per line"
    },
    "validOnly": {
        "type": "boolean",
        "title": "Valid only",
        "description": "Display only valid values (valid numbers)"
    }
}
}
```

1.4 HPE Action

1.4.1 Overview

The HPE Action is a widget that provide to users a list of actions displayed in several modes .

As example we may use a hpe action as a top menu or a metro menu or even a simple set of buttons .

It contains 4 types of actions as following:

- Navigation Link : It's used as shortcut link for a view or a page .
- Modal Form : a popup that contains a form .
- Launch : it's a generic launch
- Parameters : it's a link that contain some parameters as filters or new configuration .

It contains 7 display modes :

- Dropdown
- Button
- list
- Radio button
- Tree list
- Accordion panels
- Grid menu
- Horizontal Menu

1.4.2 JSON schema

We didn't add the JSON schema of the hpe-action widget to this documentation because of its length. However you can find the full schemas of the different display modes of actions in `<install_dir>/client/addons/hpe/widgets/hpe-actions/hpe- action-<actionDisplayMode>.json`.

1.4.3 Example

1.4.3.1 Global configuration

The HPE Action has a common configuration structure and a special configuration for each mode .

This is an example of a sample configuration :

```
{
  "id": "myWidgetAction",
  "type": "hp-ng-dropdown",
  "configuration": {
    "init": {
      "showTitle": true,
      "showIcon": true,

```



```

    },
    "title": "Action title",
    "icon": "fa fa-sellsy",
    "groups": [
      {
        "id": "action-group-default",
        "name": "default action group",
        "items": [
          {
            "id": "action-item-default",
            "name": "Go to the view",
            "type": "navigation",
            "url": "/workspaces/Test-ActionWidget/views/Test-ActionHorizontalMenu",
          }
        ]
      }
    ]
  }
]
....

```

1.4.3.1.1 Identification and Type

The following table lists the properties that this widget supports. All parameters listed are common properties .

Name	Description	Default Value	Type
Id	The property specifies the id of the widget		String (required)
Type	this property specifies the type of the action widget .	hpe-action-dropdown hpe-action-button hpe-action-list hpe-action-radio-button hpe-action-tree hpe-action-accordion hpe-action-grid hpe-action-menu	String (required)
Configuration	Details in 3.4.2.2		(Array of Objects)

1.4.3.1.2 Configuration

The property configuration is an Array of Objects and required , it contains the following properties .

Name	Description	Default Value	Type
Title	The property specifies the title of the widget	empty	String (required)
Icons	The property specifies the icon of the widget (ex : fa fa- sellsy)	empty	String (required)
Init	Details in 3.4.2.3		Object (required)
Groups	Details in 3.4.2.4		Array of Objects (required)

1.4.3.1.3 Init

The property Init contains a list of a global options required for every action widget type .

Name	Description	Default Value	Type
showTitle	Display/Hide the group Title	true	Boolean (required)
showIcon	Display/Hide the icon Title	true	Boolean (required)
sizeBtn	Support 4 level and each number represent a size . 3 : small 4 : normal 6 : large 12 : full	12	Number (required for dropdown type)
showOnSide	Support 3 position 0 : left 0.5 : center 1 : right	0	Number (required for dropdown type)
openFirst	Contain the id of the group desired to be open first.	empty	String (optional for Accordion , Tree and List action type)

rowCellsNumber	Represent number of cells by line	3	Number (optional for Grid Type)
cellSpacing	Represent space between cell	5px	String (optional for Grid Type)
separator_menu	Display/Hide separation between menu elements.	false	Boolean (required for menu Type)
radioButtonArrangement	Select how we display the buttons radio . Column or Row	Column	String (required for radio Type)
showGroupsTitle	Display/hide the groups title .	false	Boolean (required for radio Type)

1.4.3.1.4 Groups

The property groups is a list of groups , each group contain one or many actions

The property groups is a required Array of Objects and contain the property listed below .

Name	Description	Default Value	Type
id	The property specifies the id of the group		String (required)
Name	The property specifies the name of the group		String (required)
Icon	The property specifies the icon of the group (ex : fa fa- sellsy)	empty	String (optional)
class	The property specifies a custom css class for the group	empty	String (optional)
tooltip	The property specifies a description text about the group	empty	String (optional)
items	Details in 3.4.2.5		Array of Objects (required)

1.4.3.1.5 Items

Items contain a list of actions within a group .

The property items is an Array of Objects and required , it contain the property listed below .

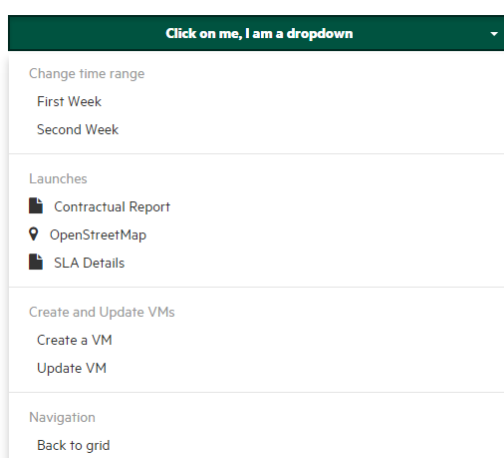
Name	Description	Default Value	Type
id	The property specifies the id of the item		String (required)
name	The property specifies the name of the item		String (required)
type		Predefined values : Parameters objectType navigation Launch	String (required)
icon	The property specifies the icon of the item (ex : fa fa-sellsy)	empty	String (optional)
Roles	List of roles authorized to run the action Ex ("roles":["Guest"])	empty	Array of string (optional)
Permissions	List of permissions needed to run the action	Ex ("permissions":["* launch"])	Array of string (optional)
tooltip	description text about the item	empty	String (optional)
outputs	The property contain our config json for parameters		Array of Objects (required for type : parameters)
tags	A single or a list of launch tags	Ex ("tags":["Demo"])	String (required for type : Launch)
dataSelection	The property contain a dataselection of our objectType	empty	Array of Objects (required for type : objectType)

dataFilters	The property contain a dataFilters config of our objectType	empty	Array of Objects (optional for type : objectType)
url	Link where we wish redirect	empty	String (required for type : Navigation)

1.4.3.2 Display Modes

1.4.3.2.1 Dropdown

A dropdown menu allows the user to choose one value from a list. When a drop-down list is inactive, it displays a single value (dropdown title). When activated, it displays (drops down) a list of items, from which the user may select one. When the user selects a new items, the control reverts to its inactive state.



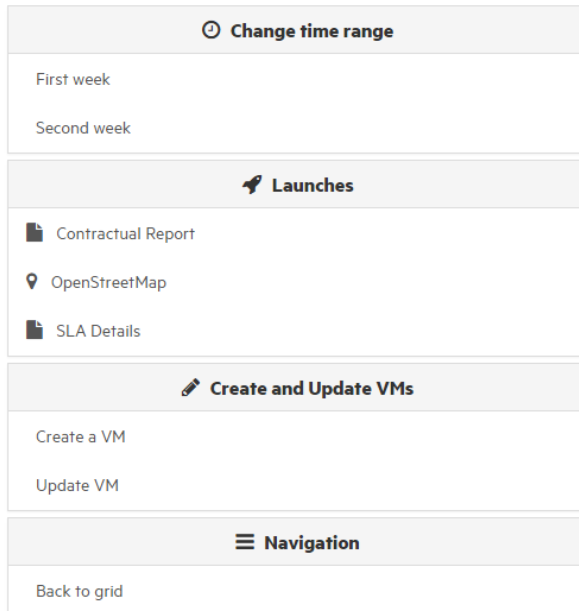
1.4.3.2.2 Button(s)

A menu button represent a single set of buttons or more, it is commonly used as a shortcut links or navigational controls .



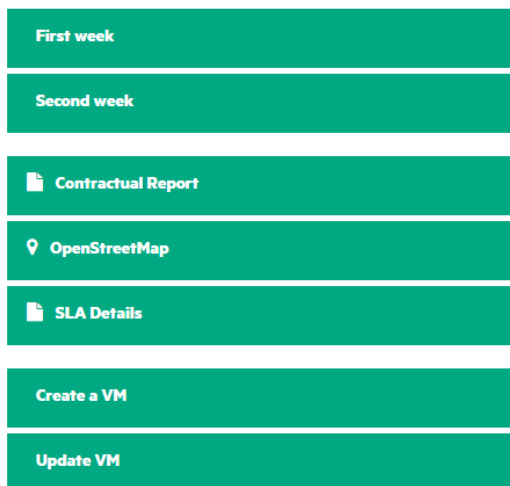
1.4.3.2.3 List

A list menu allows the user to select items from a list within panels .



1.4.3.2.4 Radio button

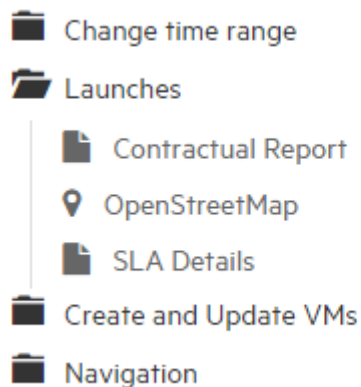
A radio button or option button allows the user to choose only one of a predefined set of options. Radio buttons are arranged in groups of two or more .



1.4.3.2.5 Tree list

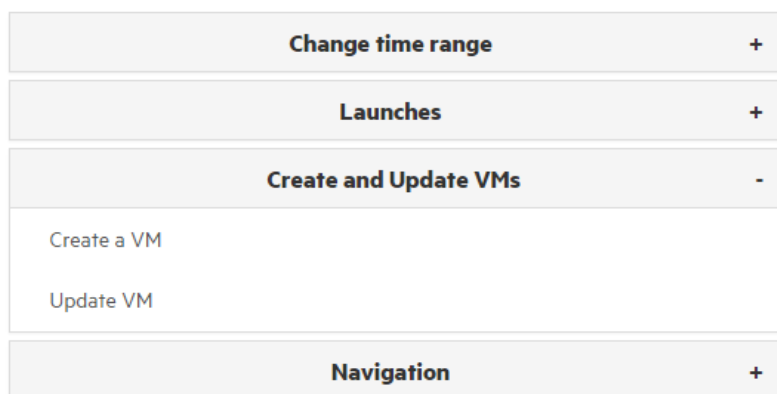
A tree List presents a hierarchical view of information. Each item can have a number of subitems. This is often visualized by indentation in a list.

An item can be *expanded* to reveal subitems, if any exist, and *collapsed* to hide subitems.



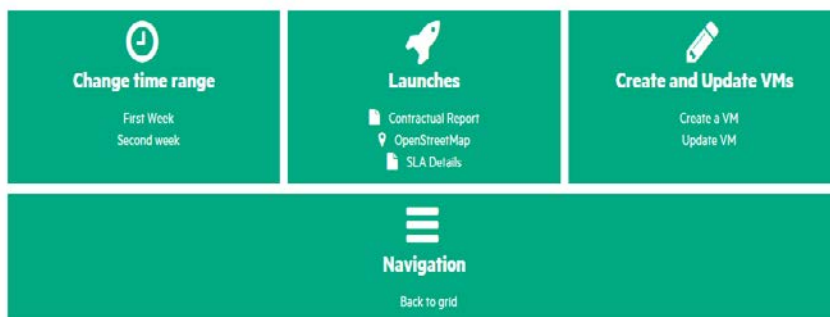
1.4.3.2.6 Accordion panels

An accordion is a vertically stacked list of items, Each item can be "expanded" or "stretched" to reveal the content associated with that item. We support only one item expanded at a time.



1.4.3.2.7 Grid menu

A Grid menu is a set of customizable buttons horizontally aligned and presented in one or many rows . each button contains a set of items .



1.4.3.2.8 Horizontal Menu

A Horizontal menu is a menu bar which contains a set of dropdown menus and buttons.



1.5 HPE Table

1.5.1 Overview

The HP Table Widget is a generic widget used to display data in tabular view.

It is integrated with the UOC generic data fetching through data selection.

For example, It can retrieve automatically columns presentation names and units, deduce cell formatting, by using any metadata associated with the current data selection.

In addition to this automatic behavior which requires no extra configuration, further customization or default behavior override is possible as detailed in the following sections.

The HP Table Widget is based on the Angular UI Grid library, which is a data grid library for the AngularJS framework.

More information about this library is available at:

<http://ui-grid.info/>

1.5.2 JSON schema

Below the JSON schema of HPE ng table configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-ng-table/hpe-ng-table.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Table",
  "description": "HPE ng table",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the ng table",
      "properties": {
        "maxRows": {
          "type": "integer",
          "title": "Max rows",
          "description": "Maximum number of rows used when vertical scrollbar is enabled",
          "default": "10"
        },
        "pagination": {
          "type": "boolean",
          "title": "Pagination",
          "description": "Enable pagination",

```



```

    "default": "false"
  },
  "paginationPageSize": {
    "type": "integer",
    "title": "Pagination page size",
    "description": "Maximum number of rows per page",
    "default": "10"
  },
  "paginationPageSizes": {
    "type": "array",
    "title": "Pagination page sizes",
    "description": "Allowed page sizes",
    "items": {
      "type": "integer",
      "title": "Page sizes",
      "description": "Items per page",
      "default": "[1,2,3,5,10,20,50,100]"
    }
  },
  "verticalScrollBar": {
    "type": "boolean",
    "title": "Vertical scrollbar",
    "description": "Enable vertical scrollbar"
  },
  "horizontalScrollBar": {
    "type": "boolean",
    "title": "Horizontal scrollbar",
    "description": "Enable horizontal scrollbar"
  },
  "enableFiltering": {
    "type": "boolean",
    "title": "Enable filtering",
    "description": "Enable filtering for all columns",
    "default": "false"
  },
  "dateFormat": {
    "type": "string",
    "title": "Date format",
    "description": "Format for all date formatted cells",
    "default": "yyyy-MM-dd HH:mm:ss"
  },
  "timeFormat": {
    "type": "string",
    "title": "TimeFormat schema.",

```

```

    "description": "Format for all time formatted cells",
    "default": "HH:mm:ss"
  },
  "numberFormat": {
    "type": "integer",
    "title": "NumberFormat schema.",
    "description": "Number of decimal places to round the number to",
    "default": "2"
  },
  "showGridFooter": {
    "type": "boolean",
    "title": "Show grid footer",
    "description": "Display a footer after the rows, containing
information about number of shown rows and total number of rows",
    "default": "false"
  },
  "enableRowSelection": {
    "type": "boolean",
    "title": "Enable row selection",
    "description": "Enable selection of rows",
    "default": "false"
  },
  "enableRowHeaderSelection": {
    "type": "boolean",
    "title": "Enable row header selection",
    "description": "Display a column at the left of the table, to check
or uncheck selected rows (RowHeader)",
    "default": "false"
  },
  "multiSelect": {
    "type": "boolean",
    "title": "Multi select",
    "description": "Enable multiple rows selection",
    "default": "false"
  },
  "enableSelectAll": {
    "type": "boolean",
    "title": "Enable select all",
    "description": "Enable button to select/unselect all rows (button is
in the RowHeader)"
  },
  "autoExport": {
    "type": "boolean",
    "title": "Auto export",
    "description": "Automatically export selection to output, display a
button to trigger export if false",

```

```

    "default": "true"
  },
  "allowCellFocus": {
    "type": "boolean",
    "title": "Allow cell focus",
    "description": "Enable selection of cells",
    "default": "false"
  },
  "showTitle": {
    "type": "boolean",
    "title": "Show title",
    "description": "Display the widget title"
  },
  "enableColumnResizing": {
    "type": "boolean",
    "title": "EnableColumnResizing schema.",
    "description": "Enable manual resizing of all columns of the table by
dragging a column header's right border",
    "default": "false"
  },
  "exporterMenuCsv": {
    "type": "boolean",
    "title": "Exporter menu csv",
    "description": "Enable the export csv entries in the table menu"
  },
  "resizeTimeout": {
    "type": "integer",
    "title": "Resize timeout",
    "description": "Resize timeout in in milliseconds"
  },
  "useExternalSorting": {
    "type": "boolean",
    "title": "Use external sorting",
    "description": "Prevents the internal sorting from executing. Events
will still be fired when the sort changes, and the sort information on the
columns will be updated, allowing an external sorter (for example, server
sorting) to be implemented",
    "default": "false"
  },
  "useExternalPagination": {
    "type": "boolean",
    "title": "UseExternalPagination schema.",
    "description": "Disables client side pagination. When true, handle
the paginationChanged event and set data and totalItems",
    "default": "false"
  },

```

```

"columns": {
  "type": "array",
  "title": "Columns",
  "description": "Columns definition",
  "items": {
    "type": "object",
    "title": "Column",
    "description": "Column definition",
    "required": [
      "field"
    ],
    "properties": {
      "displayName": {
        "type": "string",
        "title": "Display name",
        "description": "Column name that will be shown in the header"
      },
      "field": {
        "type": "string",
        "title": "Field",
        "description": "Field must be provided if you wish to bind to a
property in the data source"
      },
      "visible": {
        "type": "boolean",
        "title": "Visible",
        "description": "Column visibility"
      },
      "metadata": {
        "type": "object",
        "title": "Metadata",
        "description": "An explanation about the purpose of this
instance.",
        "properties": {
          "$ref": "#facts"
        }
      },
      "allowCellFocus": {
        "type": "boolean",
        "title": "Allow cell focus",
        "description": "Enable focus on a cell within this column",
        "default": "true"
      },
      "cellFilter": {
        "type": "string",

```

```

        "title": "Cell filter",
        "description": "Filter to apply to the content of each cell"
    },
    "cellTemplate": {
        "type": "string",
        "title": "Cell template",
        "description": "Custom template for each cell in this column"
    },
    "aggregationType": {
        "type": "integer",
        "title": "Aggregation type",
        "description": "The aggregation that you'd like to show in the
columnFooter for this column (sum: 2, count: 4, avg: 8, min: 16, max: 32)"
    },
    "aggregationHideLabel": {
        "type": "boolean",
        "title": "Aggregation hide label",
        "description": "if set to true hides the label text in the
aggregation footer, so only the value is displayed",
        "default": "false"
    },
    "sort": {
        "type": "object",
        "title": "Sort",
        "description": "Sort column",
        "properties": {
            "direction": {
                "type": "string",
                "title": "Direction",
                "description": "direction values are asc, desc"
            },
            "priority": {
                "type": "integer",
                "title": "Priority",
                "description": "says what order to sort the columns in
(lower priority gets sorted first)"
            }
        },
        "required": [
            "direction",
            "priority"
        ]
    },
    "formatter": {
        "type": "string",

```

```

        "title": "Formatter",
        "description": "Custom formatters (status, rawValue,
number...)"
    },
    "footerCellTemplate": {
        "type": "string",
        "title": "FooterCellTemplate schema.",
        "description": "Custom template for the footer for this column"
    },
    "rules": {
        "type": "array",
        "title": "Rules",
        "description": "List of rules (rules only apply for
cellFormatter)",
        "items": {
            "type": "object",
            "title": "rule",
            "description": "Rule definition",
            "properties": {
                "style": {
                    "type": "object",
                    "title": "Style",
                    "description": "Column style (can use CSS styles)",
                    "properties": {
                        "background-color": {
                            "type": "string",
                            "title": "Background color",
                            "description": "Set row background color (red,
#FF0000...)"
                        },
                        "color": {
                            "type": "string",
                            "title": "Color",
                            "description": "Set row text color (red, #FF0000...)"
                        },
                        "font-weight": {
                            "type": "string",
                            "title": "Font-weight",
                            "description": "Set row font weight (bold)"
                        }
                    }
                }
            }
        },
        "class": {
            "type": "string",
            "title": "Class",

```

```

        "description": "CSS class"
    },
    "conditions": {
        "type": "array",
        "title": "Conditions",
        "description": "Conditions list",
        "items": {
            "type": "object",
            "title": "Condition",
            "description": "Condition definition (list of possible
conditions: eq, not_eq, startsWith, not_startsWith, contains, not_contains,
endsWith, not_endsWith, gt, not_gt, ge, not_ge, lt, not_lt, le, not_le, isnull,
nonnull)",
            "properties": {
                "eq": {
                    "type": [
                        "string",
                        "integer"
                    ],
                    "title": "condition",
                    "description": "Equal condition"
                }
            }
        }
    },
    "required": [
        "conditions"
    ]
}
}
},
"rowFormatter": {
    "type": "object",
    "title": "Row formatter",
    "description": "Definition of a row formatter",
    "properties": {
        "formatter": {
            "type": "string",
            "title": "Formatter",
            "description": "Type of formatter (rowColor)"
        }
    },
    "rules": {

```

```

"type": "array",
"title": "Rules",
"description": "List of rules",
"items": {
  "type": "object",
  "title": "rule",
  "description": "Rule definition",
  "properties": {
    "style": {
      "type": "object",
      "title": "Style",
      "description": "Row style (CSS Style: see the w3c style
capabilities, the following properties are examples of use)",
      "properties": {
        "background-color": {
          "type": "string",
          "title": "Background color",
          "description": "Set row background color (red,
#FF0000...)"
        },
        "color": {
          "type": "string",
          "title": "Color",
          "description": "Set row text color (red, #FF0000...)"
        },
        "font-weight": {
          "type": "string",
          "title": "Font-weight",
          "description": "Set row font weight (bold)"
        }
      }
    }
  },
  "class": {
    "type": "string",
    "title": "Class",
    "description": "CSS class"
  },
  "conditions": {
    "type": "array",
    "title": "Conditions",
    "description": "Conditions list",
    "items": {
      "type": "object",
      "title": "Condition",

```


1.5.3.1.1 Identification and Title

Like all widget instances, a table widget has an **id** and a **type** property, along with a **title** to be displayed above the grid.

It is possible not to display the title. (see 1.5.3.1.10 about table parts visibility)

The type of widget for the generic table widget is “**hp-ng-table**”.

All graphical and behavioral configuration of a table widget is contained in the **configuration** object property of the widget.

Example of table widget declaration:

```
{  
  "id": "myTable",  
  "type": "hp-ng-table",  
  "title": "My Table",  
  "configuration": {  
    ...  
  }  
}
```

1.5.3.1.2 Inherited configuration properties

The global configuration properties are available in the **configuration** object property of the widget.

In the Angular UI Grid library, the equivalent properties are set in an object named *gridOptions* and set in the *options* attribute when used in a directive.

The documentation about these properties can be found at:

<http://ui-grid.info/docs/#/api/ui.grid.class:GridOptions>

Most of the properties of the configuration object are cascaded to the Angular UI Grid library grid options and so are available with the same behavior and default values.

Example of inherited properties:

```
...
"configuration": {
  "rowHeight": 60,
  "showGridFooter": true,
  ...
},
...
```

However, the following properties are not available directly and will not work as is:

- enablePagination
- enableHorizontalScrollbar
- enableVerticalScrollbar

To use pagination in a table, please see section 1.5.3.1.3

To use scrollbars in a table, please see section 1.5.3.1.4

Some default values of the inherited properties are also different from the original ones. These default values are detailed in the following sections.

1.5.3.1.3 Pagination

Pagination can be enabled to limit the table height when displaying many rows of data using the property **pagination**.

Example of pagination activation:

```
...
"configuration": {
  "pagination": true,
  ...
}
```

```
},
...
```

The pagination related properties are:

Name	Description	Default Value (in red when not inherited)	Type	Inherited Property
pagination	Enable pagination	false	boolean	no
paginationPageSize	Maximum number of rows per page	10	number	yes
paginationPageSizes	Allowed page sizes	[1,2,3,5,10,20,50,100]	Array of numbers	yes

1.5.3.1.4 Scrollbars

Vertical or horizontal scrollbars can be enabled instead of pagination to ease the display of many rows of data.

Even when enabled, the vertical scrollbar is disabled when the workspace containing the table is displayed in report mode. This enables the reporting of all data instead of the first page only.

Example of scrollbar activation:

```

...
"configuration": {
  "verticalScrollBar" : true,
  "maxRows" : 15,
  ...
},
...

```

The scrollbars related properties are:

Name	Description	Default Value	Type	Inherited Property
verticalScrollBar	Enable vertical scrollbar	false	boolean	no
horizontalScrollBar	Enable horizontal scrollbar	false	boolean	no
maxRows	Maximum number of rows used when vertical scrollbar is enabled	10	boolean	no

1.5.3.1.5 Sizing

Manual tuning the width for all columns of a table can be enabled by using this property:

Name	Description	Default Value	Type	Inherited Property
enableColumnResizing	Enable manual resizing of all columns of the table by dragging a column header's right border.	false	boolean	yes

Manual resizing can be disabled per column, and any column width can also be set or constrained. See 1.5.3.2.3

1.5.3.1.6 Rows Selection

The table widget support selection of one or many rows as needed.

The selected content is then formatted and set in a widget output whose id is WIDGET_SELECTIONS.

Example of selection configuration in a table widget:

```

...
"configuration": {
  "enableRowHeaderSelection": true,
  "enableRowSelection": true,
  "multiSelect": true,
  ...
},
"outputs": [{
  "name": "mySelection",
  "id": "WIDGET_SELECTIONS"
  },
  ...
],
...

```

The rows selection related properties are:

Name	Description	Default Value (in red when not inherited)	Type	Inherited Property
enableRowSelection	Enable selection of rows	false	boolean	yes
multiSelect	Enable multiple rows selection	false	boolean	no
enableRowHeaderSelection	Display a column at the left of the table, to check or uncheck selected rows. "RowHeader"	false	boolean	yes
autoExport	Automatically export selection to output, display a button to trigger export if false.	true	boolean	no
enableSelectAll	Enable button to select/unselect all rows. Button is in the "RowHeader" .	true	boolean	yes

1.5.3.1.7 Cell selection

The table widget support selection of a cell. Multiple cell selection is not currently supported.

The selected content is then formatted and set in a widget output whose id is WIDGET_SELECTIONS.

Please note that row selection cannot be used with cell selection at the same time.

Example of selection configuration in a table widget:

```
...
"configuration": {
  "allowCellFocus": true,
  ...
},
"outputs": [{
  "name": "mySelection",
  "id": "WIDGET_SELECTIONS"
},
...
],
...
```

The cell selection related properties are:

Name	Description	Default Value (in red when not inherited)	Type	Inherited Property
allowCellFocus	Enable selection of cells	false	boolean	yes

1.5.3.1.8 Filtering

The table widget support graphical filtering based on text or select input fields.

The property to enable or disable filtering for all columns is

Name	Description	Default Value	Type	Inherited Property
enableFiltering	Enable filtering for all columns	false	boolean	yes

To override this flag or to customize the filter per column, see 1.5.3.2.7

1.5.3.1.9 Global data formatting defaults

For the cells of each column of the table, data can be formatted using formatters which are equivalent to AngularJS filters.

The global behavior of formatting can be factorized for all columns by using these properties:

Name	Description	Default Value	Type	Inherited Property
dateFormat	Format for all date formatted cells	yyyy-MM-dd HH:mm:ss	string	no
timeFormat	Format for all time formatted cells	HH:mm:ss	string	no
numberFormat	Number of decimal places to round the number to	2	number	no

Example of configuration with custom date and number formats:

```
"configuration": {
  "dateFormat": "dd/MM/YYYY",
  "numberFormat": 1,
  ...
},
```

To override these global format templates per column, see 1.5.3.2.8

For more information, see:

<https://docs.angularjs.org/api/ng/filter/date>

<https://docs.angularjs.org/api/ng/filter/number>

1.5.3.1.10

1.5.3.1.11 Table parts visibility

Some parts of the table can be displayed or hidden as needed.

The configuration properties related to these features are:

Name	Description	Default Value	Type	Inherited Property
showTitle	display the widget title	true	boolean	no
showGridFooter	Display a footer after the rows, containing information about number of shown rows and total number of rows.	false	boolean	yes

Example of configuration of a table without title but with a footer:

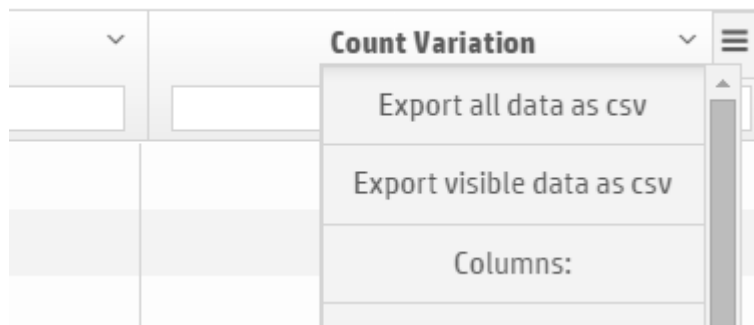
```
"configuration": {  
  "showTitle" : false,  
  "showGridFooter" : true,  
  ...  
},
```

1.5.3.1.12 Data export options

1.5.3.1.12.1 Export data as csv file

The table widget can export the data from displayed rows or from all rows as a csv file.

This feature is triggered by a menu option present in the table menu.



This feature can be disabled by using this property:

Name	Description	Default Value	Type	Inherited Property
exporterMenuCsv	Enable the export csv entries in the table menu	true	boolean	yes

1.5.3.1.12.2 Export data as pdf file

This feature is not implemented in the table widget.

Activating the “exporterMenuPdf” inherited property flag will lead to non functional entries in the menu.

The report of the view or the workspace will generate a pdf file.

1.5.3.1.13 Other Data source

In addition to using a data selection to drive data fetching, the table widget can use alternative source of raw data: inline data or data from widget input.

1.5.3.1.13.1 Inline data

Hardcoded data can be set in a **data** property directly in the **configuration** object.

Example of a table widget instance with inline data configuration:

```
{
  "id": "myTable",
  "type": "hp-ng-table",
  "title": "Sample table",
  "configuration": {
    "data": [
      {"site": "A1", "value": 0.4, "tKO": 0.9, "tWarn": 0.95},
      {"site": "B1", "value": 0.4, "tKO": 0.7, "tWarn": 0.9},
      {"site": "C1", "value": 0.4, "tKO": 0.5, "tWarn": 0.7},
      ...
    ],
    ...
  }
}
```

1.5.3.1.13.2 Data from widget input

Example configuration to use raw data from a widget input:

```
{
  "id": "myTable",
  ...
  "inputs": [{
    "name": "myData",
    "id": "TABLE_DATA",
    "listen": true
  }],
}
```

...

1.5.3.2 Column configuration

This section details the configuration per column.

1.5.3.2.1 Identification and Presentation Name

The columns configuration is defined in the **columns** property of the **configuration** object of the widget.

A column object contains at least a mandatory **field** attribute and an optional **displayName** attribute.

Example of columns declaration in a table widget configuration:

```
“configuration” : {  
...  
  “columns” : [  
    {  
      “field” : “site”,  
      ...  
    },  
    {  
      “field” : “tech”,  
      “displayName” : “Site Technology”  
      ...  
    },  
    {  
      “field” : “latency”,  
      ...  
    },  
    ...  
  ],  
  ...  
}
```

1.5.3.2.2 Inherited configuration properties

The columns are customized by defining the objects in the **columns** property of the configuration.

In the Angular UI Grid library, the equivalent properties are set in an object named *columnDefs* instead of columns.

The same properties used in a columnDef object can be used in a column object.

The documentation about these properties can be found at:

<http://ui-grid.info/docs/#/api/ui.grid.class:GridOptions.columnDef>

Example of inherited properties found in a column object:

Other inherited properties or additional properties that can be set in a column object are detailed in the following sections.

Name	Description	Default Value	Type
cellTooltip	tooltip string for the cells of this column	empty	string
headerTooltip	tooltip string for the header of this column	empty	string
cellClass	CSS classes to apply to the cells of this column	empty	string
name	Intermediate identifier between the field and the displayName, equal to the field if not set, can be used to duplicate columns based on the same field	field	string

1.5.3.2.3 Sizing

The width of a column can be customized through these properties:

Name	Description	Default Value	Type	Inherited Property
width	width of a column as : 1. a number of pixels as a number or string 2. a percentage of total table width e.g.: "20%" 3. "*" for auto computed width: use an equally divided width after columns with 1. or 2. have been added in the table	*	number or string	yes
minWidth	Minimum width of a column in pixels	None	number	yes
maxWidth	Maximum width of a column in pixels	None	number	yes
enableColumnResizing	Enable manual resizing by dragging the column header right border. Sets the width in pixels.	false	boolean	yes

Example of column width configuration with constraints:

```
{
  "field": "someValue",
  "width": "20%",
  "minWidth": 100,
  "maxWidth": 300
},
```

1.5.3.2.4 Visibility

A column visibility can be set by using this property:

Name	Description	Default Value	Type	Inherited Property
visible	Column visibility	true	boolean	yes

Example of configuration to hide a column by default in a table:

```
"configuration": {
...
  "columns": [
    {
      "field": "site",
      ...
    },
    {
      "field": "tech",
      "visible": false
      ...
    },
    ...
  ],
  ...
},
...
}
```

1.5.3.2.5 Position

Column positions from left to right is set by the order of column objects in the columns array.

Example of configuration and resulting table columns positions:

```
"configuration": {
...
  "columns": [
    {
      "field": "site",
      ...
    },
    {
      "field": "tech",
      "displayName": "Site Technology"
      ...
    },
    {
      "field": "latency",
      ...
    },
    ...
  ],
  ...
}
```

Site	Site Technology	Latency
A1	2G	43 ms
B1	3G	27 ms
C1	2G	29 ms

1.5.3.2.6 Sorting

Column sorting can be configured using the following properties:

Name	Description	Default Value	Type	Inherited Property
enableSorting	Enable or disable sorting for this column	true	boolean	yes
type	Type used for sorting, auto deduced when not set. Allowed values: 'string','boolean','number','date','object','numberStr'	auto	string	yes
sort	Object containing : a mandatory direction, either “asc” or “desc” an optional priority number starting from 0 for multisort.	none	object	yes

Example of multi sort by descending tech value and then by ascending site value:

```

“columns”: [
  {
    “field”:“site”,
    “sort”: {
      “direction”: “asc”,
      “priority”: 1
    }
  },
  {
    “field”:“tech”,
    “sort”: {
      “direction”: “desc”,
      “priority”: 0
    }
  },
  ...

```

1.5.3.2.7 Filtering

The filter or filters property of a column is set like in the original library. See examples here:

http://ui-grid.info/docs/#/tutorial/103_filtering

The difference is that in JSON, javascript constants used in the **condition** property of a filter object are not available.

They are parsed as string by the table widget.

Example of column definition using an input text and a specific condition:

```
{
  "field": "myValueIdentifier",
  "filter": {
    "condition": "uiGridConstants.filter.ENDS_WITH",
    "placeholder": "ends with"
  }
}
```

Example of column definition using a select filter:

```
{
  "field": "TECHNOLOGY",
  "filter": {
    "type": "select",
    "selectOptions": [ { "value": "2G", "label": "2nd Generation" },
                      { "value": "3G", "label": "3rd Generation" } ]
  }
}
```

More information in the filter section of this page:

<http://ui-grid.info/docs/#/api/ui.grid.class:GridOptions.columnDef>

1.5.3.2.8 Basic Formatters

Cell formatting for a column use the optional **formatter** property .

This property is not inherited and can be used to display the underlying data value in a cell by using one or many formats among:

1.5.3.2.8.1 “rawValue” : raw formatting, suitable for strings

1.5.3.2.8.2 “number” : use AngularJS number filter

- Can override a global numberFormat set in the table configuration
- Can use an optional property **unit** (displayed after the number).

1.5.3.2.8.3 “date” : use AngularJS date filter

- Can override a global dateFormat set in the table configuration

1.5.3.2.8.4 “time” : use AngularJS date filter

- Can override a global timeFormat set in the table configuration

1.5.3.2.8.5 “custom” : use the cellTemplate html template for formatting the cell (see1.5.3.2.11)

1.5.3.2.8.6 “status” : display a value as an icon. see next section for more details. (1.5.3.2.9)

Examples of formatter configuration:

Display the name of the day from a date instead of the default date format

```
{
  "field": "dayOfTheWeek",
  "formatter": "date : 'EEEE'",
```

Display an icon and then a number rounded to 1 decimal.

```
{
  "field": "variation",
  "formatter": ["status", "number:1"],
```

Display a number rounded to 1 decimal, translated into a currency and then going through a custom filter for AngularJS.

```
{
  "field": "variation",
  "formatter": "number : 1 | currency | sampleModuleFiltersCustom",
```

1.5.3.2.9 Status Formatter

1.5.3.2.9.1 Overview

The status formatter uses icons to display the underlying value of a cell.

This icon has a shape, set by an optional `classString` property referencing the CSS class of an icon. The shape will be a filled circle by default when no `classString` set.

This icon has a color, set by a mandatory property containing a CSS color representation (e.g.: `“rgb(255,0,0)”` `“#f3b237”` ...).

The color can be a name (e.g.: `“red”`, `“green”` ...), including `“transparent”`.

The icon choice is based on either:

- a mapping from the raw value if this value is a string or a number with fixed string representation (e.g.: small integers)
- thresholds if the value is a number.

1.5.3.2.9.2 Icons definition

Icon definitions are factored in a global property in the configuration object, with the following lookup structure:

```

“statusColorAliases” : {
  “colorName1” : { “color” : “colorValue1”, “classString” : “iconClassValue2” },
  “colorName2” : { “color” : “colorValue2”, “classString” : “iconClassValue2” },
  ...
}

```

For example:

```

“configuration”: {
  “statusColorAliases” : {
    “Down” : { “color” : “red”, “classString” : “fa fa-arrow-down” },
    “Up” : { “color” : “green”, “classString” : “fa fa-arrow-up” }
  },
}

```

1.5.3.2.9.3 Threshold based status formatting

The column property **statusColorFromThresholds** defines a list of interval objects, using following structure:

```

"statusColorFromThresholds" : {
  { "min" : "minValue1", "max" : "maxValue1", "color" : "colorNameOrValue1" },
  { "min" : "minValue2", "max" : "maxValue2", "color" : "colorNameOrValue2" },
  { "min" : "minValue3", "max" : "maxValue3", "color" : "colorNameOrValue3" },
  ...
}

```

Intervals must be given in increasing order

At least min or max must be given in an interval object

When not set, default min is - infinity and default max is + infinity

min and max values can be:

- a number (constant value)
- a string to lookup the number value from another column by giving its **field**

Example of threshold based status formatting:

green up arrow if value is positive, red arrow otherwise

<pre> "configuration": { "statusColorAliases": { "Down": { "color": "red", "classString": "fa fa-arrow-down" }, "Up": { "color": "green", "classString": "fa fa-arrow-up" }, }, "columns": [{ "field": "variation", "formatter": ["status", "number:1"], "unit": "%", "statusColorFromThresholds": [{ "max": 0, "color": "Down" }, { "min": 0, "color": "Up" }] }] } </pre>	
---	--

1.5.3.2.9.4 Value based status formatting:

The column property **statusColorFromValue** may be used to define a mapping from value to color. This color name will then be looked up in the **statusColorAliases** global property if there is one.

When not present, the **statusColorAliases** global property will directly be used to match a cell value with a color.

Example of value based status formatting:

Here, both direct lookup and indirect lookup through **statusColorAliases** is used.

No classString is given, so a filled circle is used as icon.

<pre> "configuration": { "statusColorAliases": { "ERROR": { "color": "rgb(255,0,0)" } }, "columns": [{ "field": "status", "formatter": "status", "statusColorFromValue": { "OK": "green", "KO": "ERROR" } }], ... </pre>	
--	--

Example of value based status formatting not using statusColorFromValue:

Here the value is directly looked up in the global property **statusColorAliases**

<pre> "configuration": { "statusColorAliases": { "3G": { "color": "black", "classString": "fa fa-tablet" }, "2G": { "color": "black", "classString": "fa fa-mobile" } }, "columns": [{ "field": "TECHNOLOGY", "formatter": ["status", "rawValue"] },], ... </pre>	<div style="text-align: center;"> □ 2G □ 2G □ 3G □ 3G □ 3G □ 3G □ 3G </div>
---	---

1.5.3.2.10 Style class override

CSS classes can be applied to a cell content for a column by setting the **cellClass** property.

Name	Description	Default Value	Type	Inherited Property
cellClass	CSS classes to be applied on all cells of this column	none	string	yes

Exam

Example of column with left justified content:

```
{  
  "field": "myValue",  
  "cellClass": "text-left",  
  ...  
}
```

1.5.3.2.11 HTML Templating

A cell can use a HTML template among its formatters or to redefine completely its content and style.

This html template is set in the **cellTemplate** column property

Name	Description	Default Value	Type	Inherited Property
cellTemplate	HTML template used by all cells of this column	none	string	yes

1.5.3.2.11.1 Conventions used in the HTML template

AngularJS expressions can be used in the template.

When using a dynamic style or class, ng-style or ng-class attributes are to be used instead of style and class.

The following variables are available inside AngularJS expressions:

`{{ row }}` : current row object (<http://ui-grid.info/docs/#/api/ui.grid.class:GridRow>)

`{{ row.entity }}` : current row data object associated with the current row

`{{ col }}` : current gridColumn object <http://ui-grid.info/docs/#/api/ui.grid.class:GridColumn>

In particular the following expressions can be used:

`{{ row.entity[col.field] }}` : value of the current cell

`{{ row.entity.fieldName }}` : value of another column whose **field** attribute is “fieldName”.

When using definition of status icon in the column or at table level, the color or classString associated to the current cell value are also available in

`{{ grid.appScope.getStatusColor(row.entity,col).color }}`

`{{ grid.appScope.getStatusColor(row.entity,col).classString }}`

Examples of HTML template :

Simple formatting of a value with the same content in tooltip.

```
{  
  "field": "someValue",  
  "cellTemplate": "<span title=\"v={{row.entity[col.field]}}\">v={{row.entity[col.field]}}</span>"  
},
```

Some Value
v=0.4
v=0.4
v=0.4
v=0.5
v=0.5
v=0.5
v=0.6
v=0.6
v=0.6
v=0.7

1.5.3.2.12 Formula based Extra column

Extra columns based on a number value computed using a formula can be added to a table.

The **formula** column property is used to define the formula.

A formula has the following structure:

[“operator” , input1 , input2 , input3 , ...]

Where **operator** is among:

"=" : affect an input (identity operator)

"+" : addition of all inputs

"*" : multiplication of all inputs

"/" : division of input1 by input 2

"-" : subtraction (input1 – input2)

"%" : modulo (input1 % input2)

"^" : Math pow (input1 ^ input2)

"deviation" : ((input1 / input 2) – 1.0)

"percent" : multiply input1 by 100

"count" : number of inputs defined and being of type number

"default" : input1 if it is defined and is a number, input2 otherwise

Any method of the Javascript Math library. for example:

"min" : minimum of the inputs

"max" : maximum of the inputs

"cos" : cosinus of input1

And where **input** may be:

- a constant number : e.g.: 123.456
- a string : to use the value of another column referenced by its **field** property or to use a value of one of the widget **inputs** referenced by its **name** property.
this other column may be based on a formula itself.
If not based on a formula, another column aggregation values based on all returned data are also available through suffix notations: avg, min, max, sum, median, count (e.g.: "latency.avg")
- a formula

Examples of formulas:

Offset between end of timerange and current row time in days.

Using a timestamp column and a widget input.

```

...
"configuration": {
  "columns": [
    {
      "displayName": "days ago",
      "field": "offset",
      "formatter": "number",
      "formula": [ "-", [ "/", [ "-", "dateTo3", "evtTime" ], 86400000 ], 1 ]
    },
    ...
  ]
}
...
"inputs": [
  {
    "name": "dateFrom3",
    "id": "TIMERANGE_FROM"
  }, {
    "name": "dateTo3",
    "id": "TIMERANGE_TO"
  }
]
...

```

Proportion of a count value over the total of all returned count values

```

{
  "field": "alarmCountPercent",
  "formula": [ "percent", [ "/", "alarmCount", "alarmCount.sum" ] ],
  "unit": "%",
  "formatter": "number"
}
...

```

1.5.3.2.13 Export formatters

Hpe-ng-table widget allows to export and download data (i.e. metrics based on dimensions) to a file (e.g. CSV).

Formatters for columns can be defined for this export context using the optional **exportOptions** property.

Name	Description	Default Value	Type	Inherited Property
exportOptions	Formatters for export context	none	object	no

A subset of the basic formatters are supported including **date** and **number**.

An additional formatter **duration** is also provided to format durations.

Chained formatters with pipes and custom formatters are currently not supported.

Examples of formatter configuration:

Display the timestamp using the default **date** formatter (ISO 8601 format)

```
{
  "field": "EVENT_TIMESTAMP",
  "exportOptions": {
    "formatter": "date"
  }
}
```

Display the timestamp using the **date** formatter with a specific format (Angular date filter format)

```
{
  "field": "EVENT_TIMESTAMP",
  "exportOptions": {
    "formatter": "date:'yyyy-MM-dd'"
  }
}
```

Display the volume rounded to 3 decimals using the **number** formatter

```
{
  "field": "volume_up_sum",
  "exportOptions": {
    "formatter": "number: 3"
  }
}
```

Display the duration using the **duration** formatter.

The first parameter is the unit of the source. Supported units are years, months, weeks, days, hours, minutes, seconds and ms.

The second parameter is the format (Angular date filter format).

Both parameters are mandatory.

In addition, an optional property **formatterOptions** enables to specify advanced formatting options.

```
{
  "field": "active_tot_dl_tp",
  "exportOptions": {
    "formatter": "duration:'ms':'HH:mm:ss'",
    "formatterOptions": {
      "trim": true, //Enable trimming of values. Default: false
      "precision": 2 //Number of digits to display for the final value. Default: 0
    }
  }
}
```



```
}
```

1.6 HPE Aggregation Table

1.6.1 Overview

An Aggregation table widget is a specific widget, dedicated to display aggregated data/values/metrics in a table. The operator can select several dimensions for analysis, several metrics to display, and the aggregated table will query aggregated result and display them in a colored table.

Aggregation Table Example (Multiple queries)

Fact	Mobile Handset	Tablet	Test	Test Device	USB Modem	Unknown	_UNKNOWN_	3G	_UNKNOWN_
Web/QoE									
Web Accessibility Q...	63	64	33	77	51	60	26	64	58
Web Retainability Q...	97	98	100	84	96	96	90	97	95
Web Quality QoE	30	16	50	20	8	35	24	31	21
Web QoE Score	63	60	58	62	52	63	45	64	58
Streaming/QoE									
streaming Accessibil...	67	53		43	65	67	48	68	52
Streaming Retainabi...	99	99		100	99	99	100	99	99
Streaming Quality Q...	100	100		100	99	100	100	100	99
Streaming QoE Score	86	81		77	86	86	79	87	81

Figure 4 - Example of aggregation table (multiple queries)

1.6.2 JSON schema

Below the JSON schema of HPE aggregation table configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-aggregation-table/hpe-aggregation-table.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Aggregation table",
  "description": "HPE aggregation table",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the aggregation table",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of the aggregation table"
        },
        "subtitle": {
          "type": "string",
          "title": "Subtitle",
          "description": "Subtitle of the aggregation table"
        },
        "autotitle": {
```

```

    "type": "boolean",
    "title": "Autotitle",
    "description": "Generate a title based on the data selection",
    "default": false
  },
  "ratioFormatter": {
    "type": "object",
    "title": "Ratio formatter",
    "description": "Ratio formatter configuration",
    "properties": {
      "displayValue": {
        "type": "boolean",
        "title": "Display value",
        "description": "Display fact values",
        "default": true
      },
      "numDecimal": {
        "type": "integer",
        "title": "Number of decimals",
        "description": "Number of decimal digits of fact values",
        "default": 0
      },
      "showUnit": {
        "type": "boolean",
        "title": "Show unit",
        "description": "Display unit of fact values",
        "default": false
      },
      "displayColor": {
        "type": "boolean",
        "title": "Display color",
        "description": "Display colored squares depending on fact
values",
        "default": true
      },
      "colors": {
        "type": "array",
        "title": "Colors",
        "description": "Color stops definition",
        "items": {
          "type": "object",
          "properties": {
            "pct": {
              "type": "integer",
              "title": "Percentage",

```

```

        "description": "Percentage value"
    },
    "color": {
        "type": "string",
        "title": "Color",
        "description": "Color to be applied (e.g. #FF0000)"
    }
},
"required": [
    "pct",
    "color"
]
}
}
},
"numberFormatter": {
    "type": "object",
    "title": "Number formatter",
    "description": "Number formatter configuration",
    "properties": {
        "numDecimal": {
            "type": "integer",
            "title": "Number of decimals",
            "description": "Number of decimal digits of fact values",
            "default": 0
        },
        "showUnit": {
            "type": "boolean",
            "title": "Show unit",
            "description": "Display unit of fact values"
        },
        "thousandSeparator": {
            "type": "string",
            "title": "Thousand separator",
            "description": "Thousands separator character",
            "default": ""
        }
    }
},
"showAsGroup": {
    "type": "boolean",
    "title": "Show as group",
    "description": "Groups facts by folder",

```

```

    "default": false
  },
  "events": {
    "type": "object",
    "title": "Events",
    "description": "Events configuration",
    "properties": {
      "click": {
        "type": "array",
        "title": "Click",
        "description": "Events on click configuration",
        "items": {
          "oneOf": [
            {
              "$ref": "#action-command"
            },
            {
              "$ref": "#action-launch"
            },
            {
              "$ref": "#action-module"
            },
            {
              "$ref": "#action-navigation"
            },
            {
              "$ref": "#action-object-type"
            },
            {
              "$ref": "#action-parameter"
            }
          ]
        }
      }
    }
  }
}

```

1.7 HPE Top Table

1.7.1 Overview

Similar to the aggregated table, a top table widget is available to display a sorted table keeping only the Top X results. The operator can select several dimensions for analysis, several metrics to display, define the top values he wants to display and the top table will query aggregated results and display them in a colored and filtered (top x result) way.

top 3 values of **DNS Resolution Attempts** and **DNS Resolution Failure Rate** and **DNS Resolution Failures** by **Device Brand / Technology** (best values first)

Rank	2/16/2015 9:15:00 AM	2/16/2015 9:30:00 AM	2/16/2015 9:45:00 AM	2/16/2015 10:00:00 AM	2/16/2015 10:15:00 AM	2/16/2015 10:30:00 AM	2/16/2015 10:45:00 AM	2/16/2015 11:00:00 AM	2/16/2015 11:15:00 AM	2/16/2015 11:30:00 AM	Rank
1	Motorola / 3G 27588	Motorola / 3G 138737	Motorola / 3G 135438	Motorola / 3G 123950	Motorola / 3G 118294	Motorola / 3G 107907	Motorola / 3G 101295	Motorola / 3G 91650	Motorola / 3G 86360	Motorola / 3G 87288	1
2	Huawei / 3G 16628	Huawei / 3G 83215	Huawei / 3G 83972	Huawei / 3G 76933	Huawei / 3G 69967	Huawei / 3G 65093	Huawei / 3G 59071	Huawei / 3G 56483	Huawei / 3G 54414	Huawei / 3G 54959	2
3	Unknown / 3G 8525	Unknown / 3G 44324	Unknown / 3G 46050	Unknown / 3G 43233	Unknown / 3G 40689	Unknown / 3G 38661	Unknown / 3G 37724	Unknown / 3G 34704	Unknown / 3G 34080	Unknown / 3G 37012	3

First Previous 1 2 3 4 5 ... Next Last

Figure 5 - Example of top table (Top 10 of Network QOE score by Cell Name)

1.7.2 JSON schema

Below the JSON schema of HPE top table configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-top-table/hpe-top-table.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Top table",
  "description": "HPE top table widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the top table",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of the top table widget"
        },
        "autotitle": {
          "type": "boolean",
          "title": "Autotitle",
          "description": "Generate a title based on the data selection",
          "default": false
        },
        "displayFactValue": {
          "type": "boolean",
```

```

    "title": "Display fact value",
    "description": "Display facts values in the cells of the top table",
    "default": true
  },
  "displayBar": {
    "type": "boolean",
    "title": "Display bar",
    "description": "Display a colored bar in the cells of the top table",
    "default": true
  },
  "barColors": {
    "type": "array",
    "title": "Bar colors",
    "description": "Color stops definition. Values are equally
distributed between the facts maximum values (from lower bound to higher
bound)",
    "items": {
      "type": "string",
      "description": "Color in the rgb format (e.g. rgb(255,0,0)) or the
hexadecimal format (e.g. #FF0000)"
    }
  },
  "numberDecimals": {
    "type": "string",
    "title": "Number of decimals",
    "description": "Number of decimal digits of fact values",
    "default": "auto"
  },
  "leftRankingColumn": {
    "type": "boolean",
    "title": "Left ranking column",
    "description": "Display the ranking column on the left",
    "default": true
  },
  "rightRankingColumn": {
    "type": "boolean",
    "title": "Right ranking column",
    "description": "Display the ranking column on the right",
    "default": false
  },
  "maxColumns": {
    "type": "integer",
    "title": "Max columns",
    "description": "Max number of columns per page, ranking column
excluded",
    "default": 10
  }

```

```
    },
    "nbSelectablePages": {
      "type": "integer",
      "title": "Number of selectable pages",
      "description": "Number of selectable pages before the ellipsis
appears",
      "default": 5
    },
    "events": {
      "type": "object",
      "title": "Events",
      "description": "Events configuration",
      "properties": {
        "click": {
          "type": "array",
          "title": "Click",
          "description": "Events on click configuration",
          "items": {
            "oneOf": [
              {
                "$ref": "#action-command"
              },
              {
                "$ref": "#action-launch"
              },
              {
                "$ref": "#action-module"
              },
              {
                "$ref": "#action-navigation"
              },
              {
                "$ref": "#action-object-type"
              },
              {
                "$ref": "#action-parameter"
              }
            ]
          }
        }
      }
    }
  }
}
```



```
}
```

1.8 HPE Breadcrumb

1.8.1 Overview

A Widget Breadcrumb is a widget dedicated to display information about the position of the end user in the workspaces.

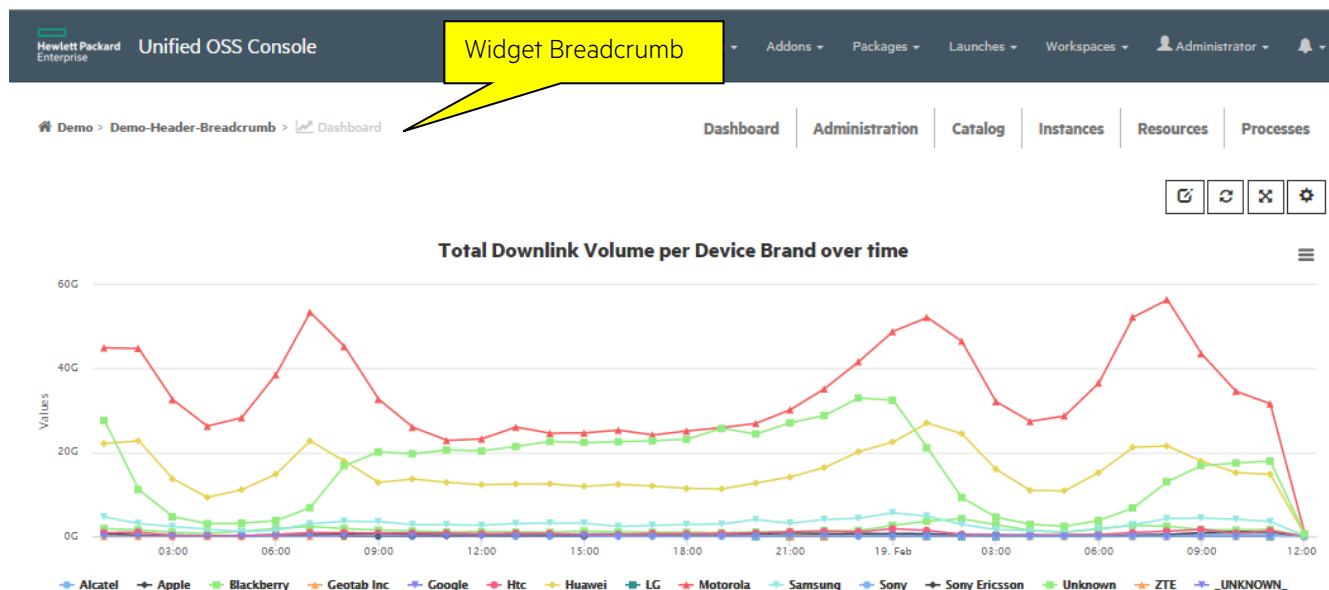


Figure 6 – Example of widget breadcrumb

1.8.2 JSON schema

Below the JSON schema of HPE breadcrumb widget configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-breadcrumb/hpe-breadcrumb.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "HPE Breadcrumb",
  "description": "The HPE Breadcrumb does not have any configuration. However, The widget must have a breadcrumb_def input. The view generally has the outputs defining the breadcrumb definition that allow the breadcrumb to initialize. You can see in the development guide the breadcrumb definition object",
  "properties": {}
}
```

1.9 HPE Time Selector

1.9.1 Overview

The widget time selector is very useful to specific one or multiple range of time (start date / end date) and optionally a granularity if you want to get metrics over the time (e.g. display a performance metric every 15 minutes).

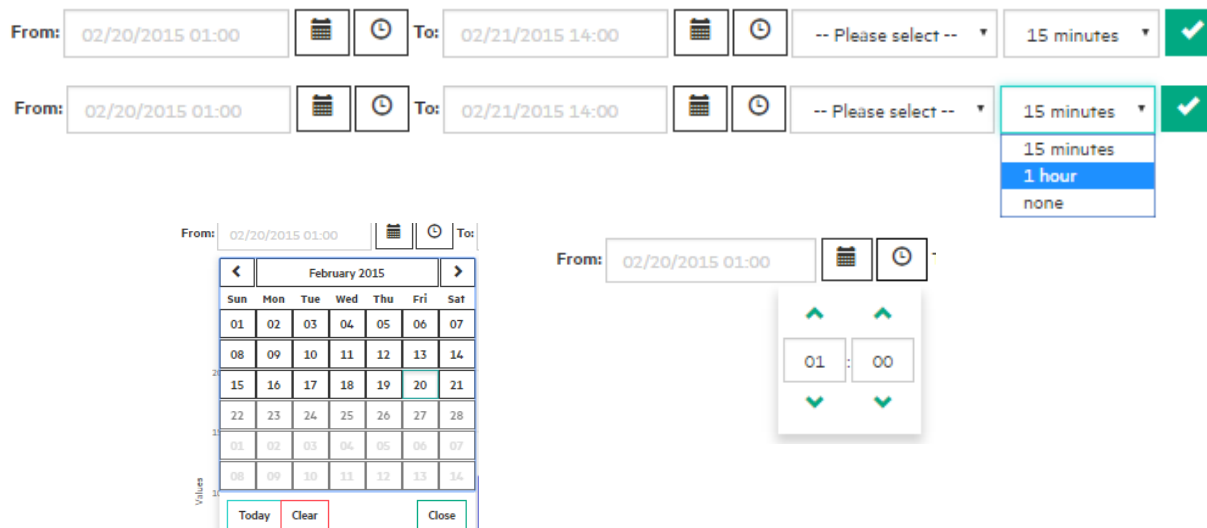


Figure 7 – Widget Time Selector

1.9.2 JSON schema

Below the JSON schema of HPE time selector configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-time-selector/hpe-time-selector.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Time selector",
  "description": "HPE time selector widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the time selector",
      "properties": {
        "showTitle": {
          "type": "boolean",
          "title": "Show title",
          "description": "Show the title of hpe time selector widget"
        },
        "title": {
```

```

    "type": "string",
    "title": "Title",
    "description": "Title of hpe time selector widget"
  },
  "init": {
    "type": "object",
    "title": "Initialization",
    "description": "Initialization of the hpe time selector",
    "properties": {
      "granularityChoices": {
        "type": "array",
        "title": "Granularity choices",
        "description": "List of granularity choices",
        "items": {
          "type": "object",
          "title": "Granularity",
          "description": "Definition of the granularity",
          "properties": {
            "labelKey": {
              "type": "string",
              "title": "Label key",
              "description": "The name that will be displayed in the
granularity dropdown. It supports custom labelKeys. The enum described below
defines granularity that are handled by the internationalization",
              "enum": [
                "GRANULARITY.NONE",
                "GRANULARITY.15MN",
                "GRANULARITY.1H",
                "GRANULARITY.24H",
                "GRANULARITY.1W",
                "GRANULARITY.2W",
                "GRANULARITY.1M"
              ]
            },
            "value": {
              "type": "integer",
              "title": "Value",
              "description": "Value of the granularity in seconds (0,
900, 3600...)"
            }
          }
        }
      },
      "timeRangeChoices": {
        "type": "array",

```

```

    "title": "Time range choices",
    "description": "List of time range choices",
    "items": {
        "type": "object",
        "title": "Time range",
        "description": "Definition of a time range. It is either an
object with an id property (enum) or an object describing a custom time range",
        "oneOf": [
            {
                "id": {
                    "type": "string",
                    "title": "time range id",
                    "description": "List of time range choices. Note: the
value \"\" is used to have a --Please Select-- choice in the dropdown",
                    "enum": [
                        "",
                        "01MN",
                        "02MN",
                        "03MN",
                        "05MN",
                        "10MN",
                        "15MN",
                        "30MN",
                        "LASTHOURL",
                        "LAST2HOURS",
                        "LAST4HOURS",
                        "LAST8HOURS",
                        "LAST12HOURS",
                        "LAST16HOURS",
                        "LAST24HOURS",
                        "TODAY",
                        "YESTERDAY",
                        "LASTMONTH",
                        "LAST2MONTHS",
                        "LASTWEEK",
                        "LAST2WEEKS",
                        "CURRENTMONTH"
                    ]
                }
            },
            {
                "label": {
                    "type": "string",
                    "title": "Label",
                    "description": "Display name of the timerange choice"
                }
            }
        ]
    }

```

```

        },
        "labelKey": {
            "type": "string",
            "title": "Label",
            "description": "Key of the timerange choice must be
unique"
        },
        "fromDifference": {
            "type": "string",
            "title": "From difference",
            "description": "Distance of the TIMERANGE_FROM value from
today (fromForcedTime) in seconds. Example: -86400 means 24h ago; 86400 means
24h in the future."
        },
        "toDifference": {
            "type": "string",
            "title": "To difference",
            "description": "Distance of the TIMERANGE_TO value from
today (toForcedTime) in seconds. Example: -86400 means 24h ago; 86400 means 24h
in the future."
        },
        "fromForcedTime": {
            "type": "string",
            "title": "From forced time",
            "description": "Time to use to calculate fromDifference
(default is 00:00:00)"
        },
        "toForcedTime": {
            "type": "string",
            "title": "From forced time",
            "description": "Time to use to calculate fromDifference
(default is 00:00:00)"
        }
    }
]
}
},
"showTime": {
    "type": "boolean",
    "title": "Show time",
    "description": "Show time (hours, minutes)"
},
"showSeconds": {
    "type": "boolean",
    "title": "Show seconds",
    "description": "Show time (seconds)"
},
}

```

```
    "showFrom": {
      "type": "boolean",
      "title": "Show from",
      "description": "Show from date calendar"
    },
    "showTo": {
      "type": "boolean",
      "title": "Show to",
      "description": "Show to date calendar"
    },
    "showGranularity": {
      "type": "boolean",
      "title": "Show granularity",
      "description": "Show granularity choices"
    },
    "showRelativeTimeRange": {
      "type": "boolean",
      "title": "Show relative timerange",
      "description": "Show relative timerange (Last minute, Last 2
minutes, Last hour...)"
    },
    "forceRefresh": {
      "type": "boolean",
      "title": "Force refresh",
      "description": "Force a refresh on the workspace when user clicks
on apply button"
    },
    "useMeridian": {
      "type": "boolean",
      "title": "Use meridian",
      "description": "Use meridian in time selection (AM, PM)"
    }
  }
}
}
}
}
}
```

1.10 HPE Drilldown

1.10.1 Overview

The Drill down and up widget allows to explore and drill down to metrics that have been used for the high level value computation. It explores a tree of facts and applies threshold definitions to provide a summarized presentation. It is usually linked to another aggregation widget (top table, aggregation table...) to select the right metric (fact) to explore and troubleshoot the eventual issue met by a customer.

Each line displays the name of the fact, its returned value, and a color that changes depending on the threshold.

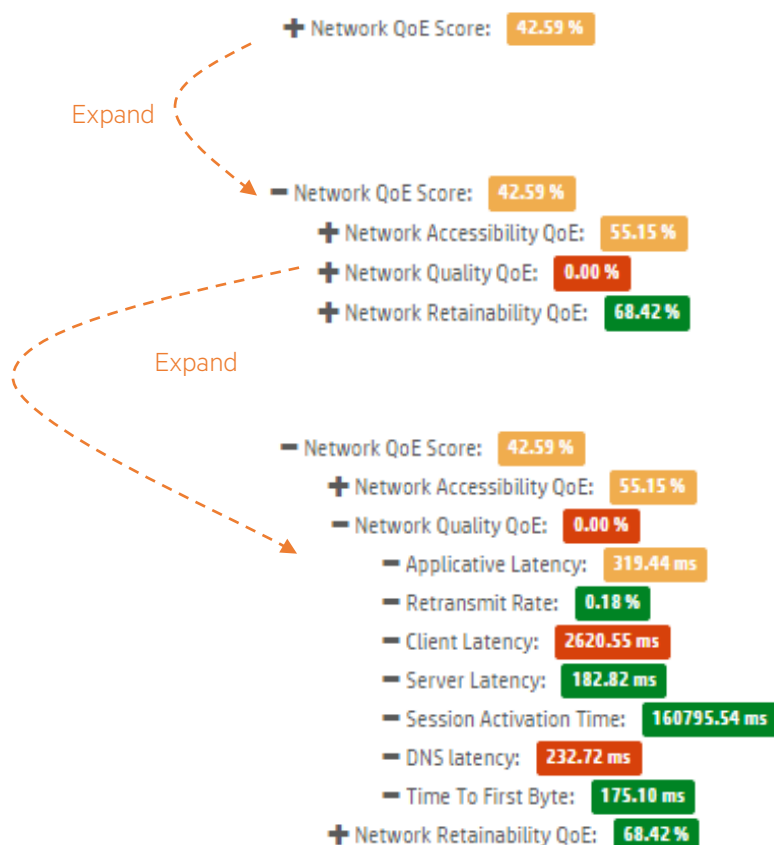


Figure 8 – Expanding QoE score with the Drill up and down widget

1.10.2 JSON schema

Below the JSON schema of HPE drilldown widget configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-drilldown/hpe-drilldown.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "HPE drilldown widget",
  "description": "HPE drilldown widget",
  "properties": {
```



```

"configuration": {
  "type": "object",
  "title": "Configuration",
  "description": "Configuration of the drilldown widget",
  "properties": {
    "autotitle": {
      "type": "boolean",
      "title": "Autotitle",
      "description": "Generate a title based on the data
selection"
    },
    "thresholds": {
      "type": "object",
      "title": "Thresholds",
      "description": "Thresholds is an object with <FACT_ID>
properties. It can set color thresholds on a specific fact. If no threshold are
set, the default color is INFO color",
      "properties": {
        "<FACT_ID>": {
          "type": "object",
          "title": "<FACT_ID>",
          "description": "The <FACT_ID> ",
          "properties": {
            "<LEVEL>": {
              "type": "integer",
              "title": "<LEVEL>",
              "description": "Each property is a level
(warning, danger) and their value is an integer."
            }
          }
        }
      }
    },
    "nbDecimals": {
      "type": "integer",
      "title": "NbDecimals",
      "description": "Number of decimals there will be behind the
fact value",
      "default": 0
    },
    "manageThresholds": {
      "type": "boolean",
      "title": "ManageThresholds",
      "description": "If you want to use the 'thresholds'
property to color the facts value. If set to false, the fact values will be
colored in gray",
      "default": true
    }
  }
}

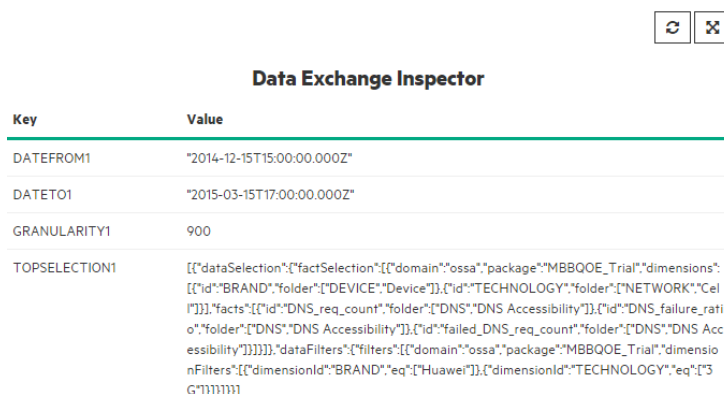
```

```
}  
  }  
    }  
      }  
        }
```

1.11 HPE Data Exchange Inspector

1.11.1 Overview

The data exchange inspector widget is a troubleshooting widget, very useful for View Designers, integrators and widget developers. It displays in real-time the contents of the global memory in charge of exchanging data between components. Everything set in the Data Exchange Service area is listed in an HTML table and displayed by this widget.



⌂ ✕

Data Exchange Inspector

Key	Value
DATEFROM1	"2014-12-15T15:00:00.000Z"
DATETO1	"2015-03-15T17:00:00.000Z"
GRANULARITY1	900
TOPSELECTION1	[[{"dataSelection":{"factSelection":[{"domain":"ossa","package":"MBBQOE_Trial","dimensions":[{"id":"BRAND","folder":{"DEVICE":"Device"}},{id":"TECHNOLOGY","folder":{"NETWORK":"Cell"}}],facts":[{"id":"DNS_req_count","folder":{"DNS":"DNS Accessibility"}},{id":"DNS_failure_ratio","folder":{"DNS":"DNS Accessibility"}},{id":"failed_DNS_req_count","folder":{"DNS":"DNS Accessibility"}]}]}]}],dataFilters":[{"filters":[{"domain":"ossa","package":"MBBQOE_Trial","dimensionFilters":[{"dimensionId":"BRAND","eq":["Huawei"]},{dimensionId":"TECHNOLOGY","eq":["3G"]}]}]}]]

Figure 9 – Data Exchange Inspector

1.11.2 JSON schema

Below the JSON schema of HPE data exchange inspector widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-data-exchange-inspector/hpe-data-exchange-inspector.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "HPE Data-Exchange-Inspector",
  "description": "The Data-Exchange-Inspector widget displays what is being exchanged between UI components. It does not need any configuration",
  "properties": {}
}
```

1.12 HPE Form

1.12.1 Overview

A Widget Form is a widget that can generate forms inside a widget and collect some end user inputs and execute an operation. The widget uses a definition provided by packages to get a description of the object type and the way to represent the information.

Figure 10 – Widget Form

1.12.2 JSON schema

Below the JSON schema of HPE form widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-form/hpe-form.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Form",
  "description": "HPE form widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the form",
      "properties": {
        "cssClass": {
          "type": "string",
```

```
"title": "CSS class",  
  "description": "CSS class to be applied to the form"  
}  
}  
}  
}  
}
```

1.13 HPE Search

1.13.1 Overview

A Widget Search is a widget that can generate forms dedicated to apply filters to another widget like a table. It is also possible to show/hide this search widget setting an input parameter (WIDGET_SHOW) to display an expand/collapse effect.

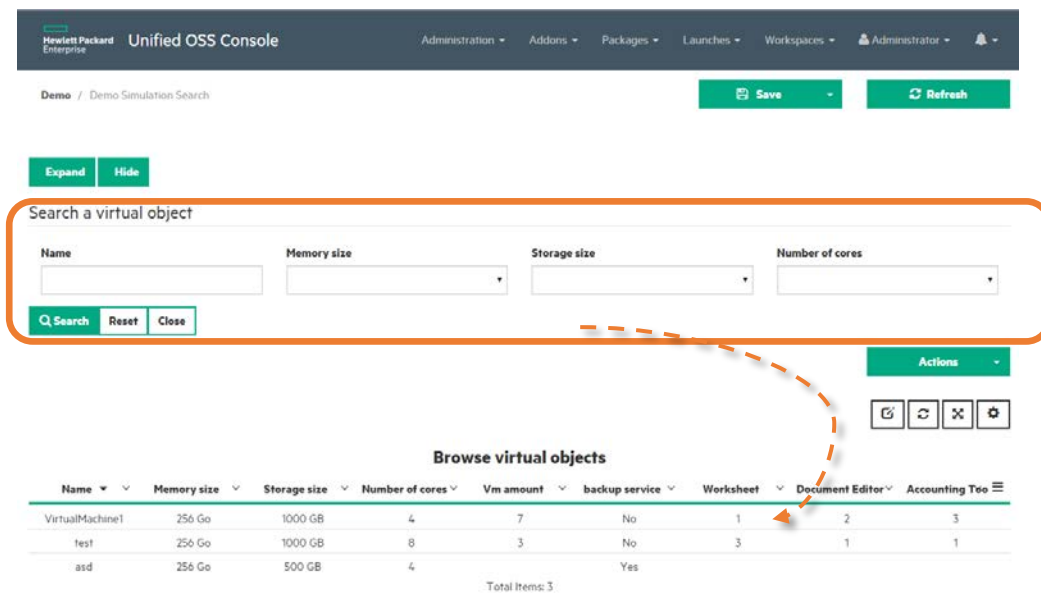


Figure 11 - Widget Search

1.13.2 JSON schema

Below the JSON schema of HPE search widget configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-search/hpe-search.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Search",
  "description": "HPE search widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the search",
      "properties": {
        "initialSearchValues": {
          "type": "object",
          "title": "Initial search values",
          "description": "Populate form with initial values",

```

```
    "properties": {
      "name": {
        "type": "string",
        "title": "Name",
        "description": "Name is an example of a form
property"
      }
    },
    "searchOperation": {
      "type": "object",
      "title": "Search operation",
      "description": "Operations supported by the plugin
(startswith, contains, like.i, not.like.i, eq, in, not.in)",
      "properties": {
        "[object_type_property]": {
          "type": "string",
          "title": " Object type property",
          "description": "The object type property on which
we will apply the filter",
        }
      }
    },
    "cssClass": {
      "type": "string",
      "title": "Css class",
      "description": "Css class used in the hpe search"
    }
  }
}
```

1.14 HPE Launch Tree

1.14.1 Overview

The Launch tree is a widget that displays in a tree representation a set of launches defined by categories. It is usually associated to a widget Iframe to display the launch next to the tree launch, and then provide a seamless integration of external systems.

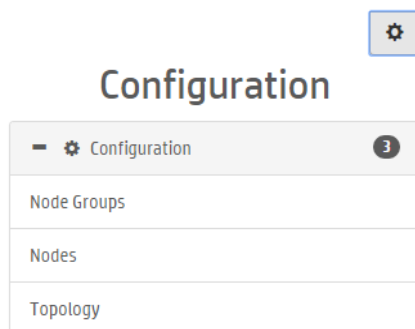


Figure 12 – Widget launch tree

1.14.2 JSON schema

Below the JSON schema of HPE launch tree widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-launch-tree/hpe-launch-tree.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Launch tree",
  "description": "HPE launch tree widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the launch tree widget",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of the launch tree widget"
        },
        "tags": {
          "type": "array",
          "title": "Tags",
          "description": "List of tags. Only launches having all the tags will be proposed",
          "items": {
            "type": "string"
          }
        }
      }
    }
  }
}
```



```
    }
  },
  "groupByCategory": {
    "type": "boolean",
    "title": "Group by category",
    "description": "Group launches by category",
    "default": false
  },
  "collapsible": {
    "type": "boolean",
    "title": "Collapsible",
    "description": "Allow panels to be collapsible",
    "default": false
  },
  "sortCategories": {
    "type": "boolean",
    "title": "Sort categories",
    "description": "Sort categories by name",
    "default": false
  },
  "sortLaunches": {
    "type": "boolean",
    "title": "Sort launches",
    "description": "Sort launches by name",
    "default": false
  }
},
"required": [
  "tags"
]
}
}
```

1.15 HPE Localization Inspector

1.15.1 Overview

The Widget Localization inspector is dedicated to SDK developer to identify quickly in all the add-ons and component, all the mistake related to the L10N support. Like missing localization, invalid localization ... scanning all the code and matching the defined locale in UOC.

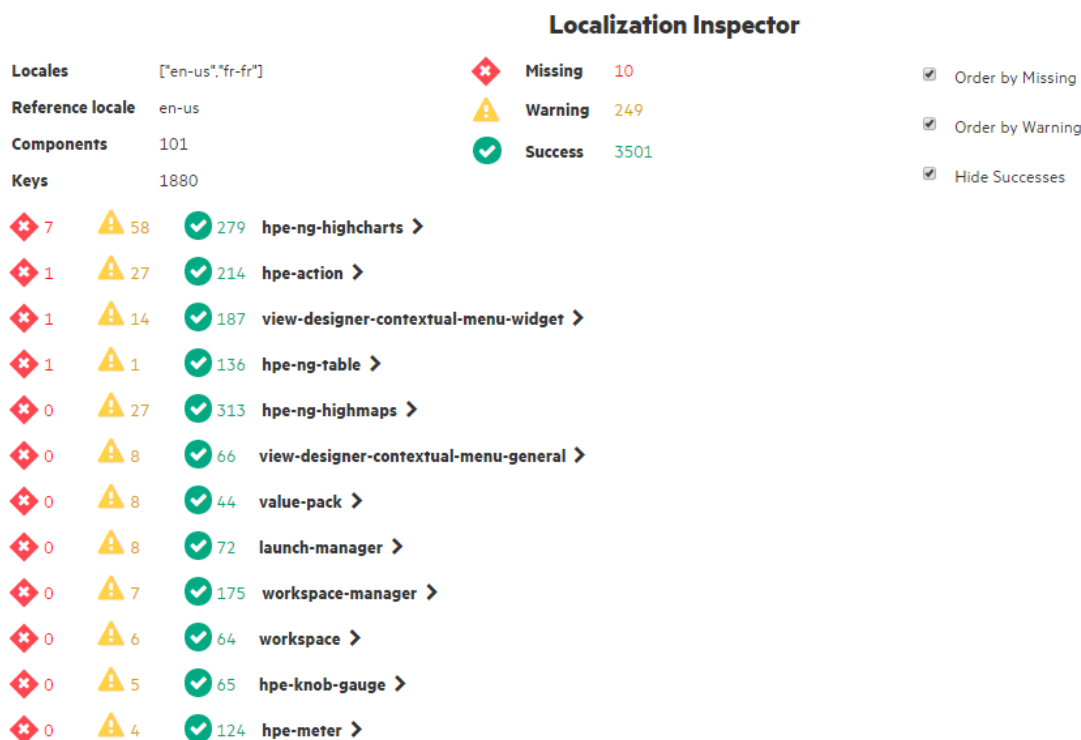


Figure 13 - Widget Localization

1.15.2 JSON schema

Below the JSON schema of HPE localization inspector widget configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-localization-inspector/hpe-localization-inspector.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Hpe localization inspector",
  "description": "HPE localization inspector widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the hpe localization inspector widget",

```

```
"properties": {
  "title": {
    "type": "string",
    "title": "Title",
    "description": "Title of hpe localization inspector widget"
  },
  "localeRef": {
    "type": "string",
    "title": "LocaleRef",
    "description": "Title of hpe localization inspector widget"
  },
  "orderByMissing": {
    "type": "boolean",
    "title": "Order by missing",
    "description": "Order widget by missing error"
  },
  "hideSuccesses": {
    "type": "boolean",
    "title": "Hide successes",
    "description": "Hide the widget that has no error"
  },
  "orderByWarning": {
    "type": "boolean",
    "title": "Order by warning",
    "description": "Order the widgets by warning error"
  }
}
}
```

1.16 HPE Meter

1.16.1 Overview

A Meter Widget is a metric widget that can be used for allocation, performances or progression use cases. It can handle a single query with multiple dimensions and facts and display a synthetic information about progress or state.

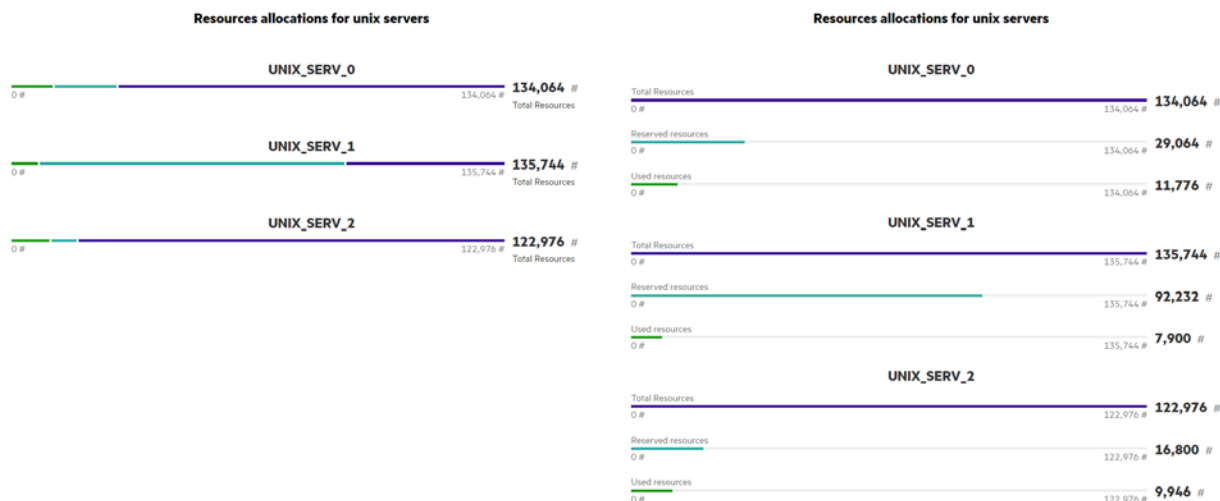


Figure 14 – Example of meter with a reference fact (stacked on left)

1.16.2 JSON schema

Below the JSON schema of HPE meter widget configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-meter/hpe-meter.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Meter",
  "description": "HPE meter widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the meter",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of the meter widget"
        }
      },
      "showTitle": {
        "type": "boolean",
        "title": "Show title",

```

```

    "description": "Show the title",
    "default": true
  },
  "titleColor": {
    "type": "string",
    "title": "Title color",
    "description": "Color of the title"
  },
  "autoTitle": {
    "type": "boolean",
    "title": "Autotitle",
    "description": "Generate a title based on the data selection",
    "default": true
  },
  "init": {
    "type": "object",
    "title": "Initialization",
    "description": "Initialization of the meter",
    "properties": {
      "general": {
        "type": "object",
        "title": "General",
        "description": "General properties",
        "properties": {
          "barsMod": {
            "type": "string",
            "title": "Bars mod",
            "description": "Display a bar per fact (classic) or a single
bar for all facts (stacked)",
            "enum": [
              "classic",
              "stacked"
            ],
            "default": "classic"
          },
          "sizeMod": {
            "type": "string",
            "title": "Size mod",
            "description": "Bars width",
            "enum": [
              "small",
              "medium",
              "large"
            ],
            "default": "medium"
          }
        }
      }
    }
  }
}

```

```

    },
    "showBarTitle": {
      "type": "boolean",
      "title": "Show bar title",
      "description": "Display a title for each bar",
      "default": true
    },
    "barTitleType": {
      "type": "string",
      "title": "Bar title type",
      "description": "Specify which information should be displayed
as title (dim: dimension value, fact: fact value, dimFact: dimension and fact
values, status: threshold status)",
      "enum": [
        "dim",
        "fact",
        "dimFact",
        "status"
      ],
      "default": "dimFact"
    },
    "maximumType": {
      "type": "string",
      "title": "Maximum type",
      "description": "Determine the maximum value of bars by
configuration (config) or using the value of a fact in the data selection
(factDependent)",
      "default": "config",
      "enum": [
        "config",
        "factDependent"
      ]
    },
    "referenceFact": {
      "type": "string",
      "title": "Reference fact",
      "description": "Specify which fact should be used as
reference for determining the maximum value of bars. Requires maximumType =
factDependent"
    },
    "thresholdsEnabled": {
      "type": "boolean",
      "title": "Thresholds enabled",
      "description": "Display thresholds",
      "default": false
    }
  },

```

```

        "thresholdsMod": {
            "type": "string",
            "title": "Thresholds mod",
            "description": "Specify how thresholds should be displayed
(onTop: horizontal bars, verticalBars: vertical bars, barColor: bar
coloration)",
            "enum": [
                "onTop",
                "verticalBars",
                "barColor"
            ]
        },
        "thresholdsVerticalColorEnabled": {
            "type": "boolean",
            "title": "Thresholds vertical color enabled",
            "description": "Color vertical thresholds indicators",
            "default": false
        },
        "thresholdsVerticalStatusEnabled": {
            "type": "boolean",
            "title": "Thresholds vertical status enabled",
            "description": "Display threshold status on vertical
threshold indicators",
            "default": false
        }
    },
    "legend": {
        "type": "object",
        "title": "Legend",
        "description": "Legend properties",
        "properties": {
            "showDimensionTitle": {
                "type": "boolean",
                "title": "Show dimension titles",
                "description": "Display dimension values as titles",
                "default": true
            },
            "showUnits": {
                "type": "boolean",
                "title": "Show unit",
                "description": "Display unit of fact values",
                "default": false
            },
            "showMinMax": {

```

```

        "type": "boolean",
        "title": "Show minimum / maximum",
        "description": "Display minimum and maximum values",
        "default": false
    },
    "precision": {
        "type": "integer",
        "title": "Precision",
        "description": "Number of decimal digits of fact values",
        "default": 0
    }
},
"valueLegend": {
    "type": "object",
    "title": "Value legend",
    "description": "Value legend properties",
    "properties": {
        "hidden": {
            "type": "boolean",
            "title": "Hidden",
            "description": "Hide the legend",
            "default": false
        },
        "top": {
            "type": "boolean",
            "title": "Top",
            "description": "Display legend on top of bars",
            "default": false
        },
        "width": {
            "type": "string",
            "title": "Width",
            "description": "Legend width",
            "default": "200px"
        },
        "showFact": {
            "type": "boolean",
            "title": "Show fact",
            "description": "Display fact value in the legend",
            "default": true
        },
        "showStatus": {
            "type": "boolean",

```



```

        "title": "Show status",
        "description": "Display threshold status in the legend",
        "default": false
    }
}
},
"factsConfiguration": {
    "type": "object",
    "title": "Facts configuration",
    "description": "Additional properties for facts (e.g. thresholds,
min/max values), indexed by fact id",
    "additionalProperties ": {
        "type": "object",
        "properties": {
            "thresholds": {
                "type": "array",
                "title": "Thresholds",
                "description": "Thresholds definitions. Warning: Thresholds
definitions are applied sequentially.",
                "items": {
                    "type": "object",
                    "properties": {
                        "name": {
                            "type": "string",
                            "title": "Name",
                            "description": "Display name of the threshold status"
                        },
                        "value": {
                            "type": "integer",
                            "title": "Value",
                            "description": "Threshold value (lower bound)"
                        },
                        "color": {
                            "type": "string",
                            "title": "Color",
                            "description": "Color associated to the threshold"
                        }
                    }
                }
            }
        }
    },
    "min": {
        "type": "integer",
        "title": "Minimum",
        "description": "Minimum value of the fact"
    },

```

```
        "max": {
            "type": "integer",
            "title": "Maximum",
            "description": "Maximum value of the fact"
        }
    }
}
}
}
}
}
}
}
}
}
}
```

1.17 HPE Navigation Toolbar

1.17.1 Overview

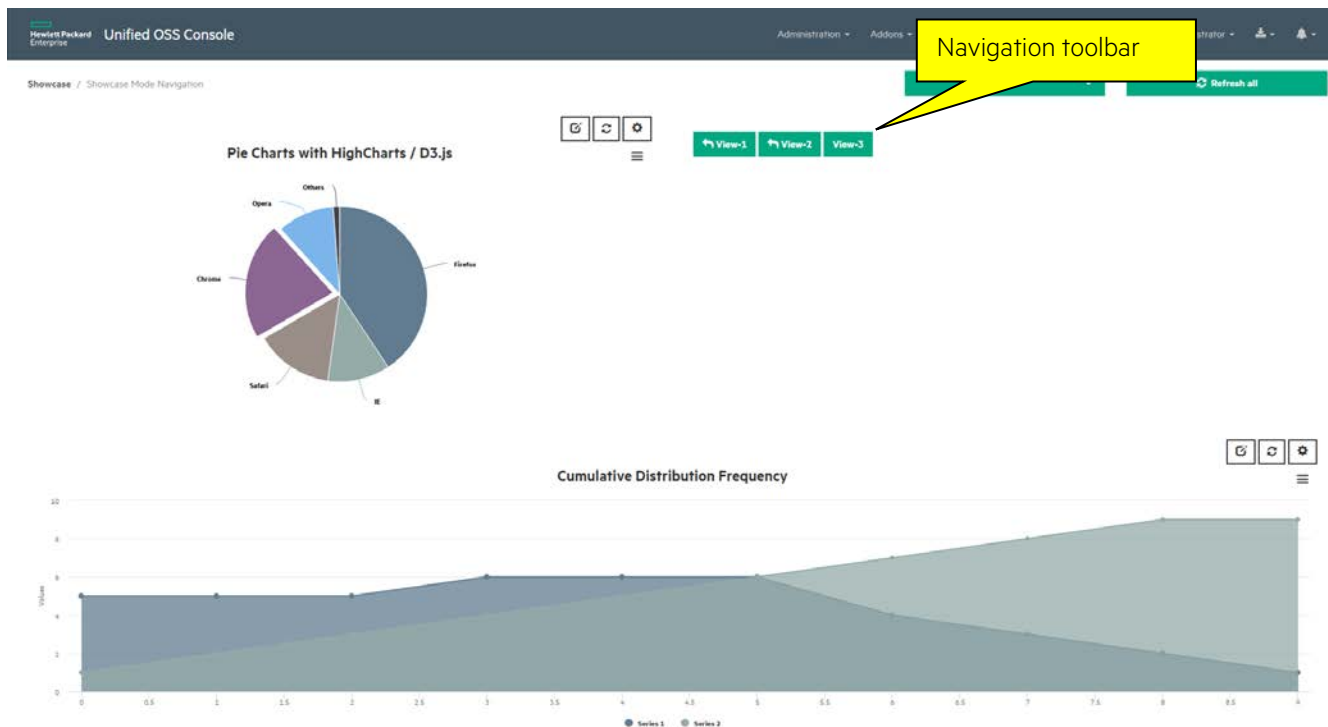


Figure 15 - Example of a Navigation Toolbar widget

1.17.2 JSON schema

Below the JSON schema of HPE navigation toolbar widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-navigation-toolbar/hpe-navigation-toolbar.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "HPE Navigation Toolbar",
  "description": "HPE Navigation Toolbar Configuration",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the Navigation Toolbar widget",
      "properties": {
        "navigationToolbar": {
          "type": "array",
          "title": "NavigationToolbar",
          "description": "NavigationToolbar: Array of navigation objects. Buttons that allow navigation",

```

```

        "items": {
            "type": "object",
            "title": "Navigation Object",
            "description": "An item of this set is represent a
button in the Navigation Toolbar",
            "properties": {
                "enabled": {
                    "type": "boolean",
                    "title": "Enabled",
                    "description": "To enable or disable the
button",
                    "default": true
                },
                "active": {
                    "type": "boolean",
                    "title": "Active",
                    "description": "When set to true, btn-info
class is applied to the button",
                    "default": false
                },
                "name": {
                    "type": "string",
                    "title": "Name",
                    "description": "Name that will be displayed
inside the button"
                },
                "message": {
                    "type": "string",
                    "title": "Message",
                    "description": "Name of the event that will be
emitted or broadcasted upon a click on the button. To navigate to a view use
'webgui.workspace.navigateView'"
                },
                "parameter": {
                    "type": "string",
                    "title": "Parameter",
                    "description": "parameter of the event. To
navigate to a view, set the view ID"
                },
                "icon": {
                    "type": "string",
                    "title": "Icon",
                    "description": "icon that will be displayed
inside and on the left of the button"
                },
                "direction": {
                    "type": "string",

```

```
        "title": "Direction",
        "description": "Whether the event will be
broadcasted or emitted. 'up' or 'parent' means emit and 'down' or 'child' means
broadcast",
        "default": "up"
    }
},
"required": [
    "name",
    "message",
    "parameter"
]
}
},
"required": [
    "navigationToolbar"
]
}
}
```

1.18 HPE Web Content

1.18.1 Overview

Web content widget is a very generic and useful widget that allow to add text, images, or links to an area of the view, like any simple widget. It is useful to present read only information near analysis or contact list in a troubleshooting views.

It could also be used as s in place WYSIWYG editor and ease knowledge sharing between users. An operator can edit notes on the screen and save it in the workspace to share with all users. It supports a lot of customization to select the feature you want to display to the users in toolbar.

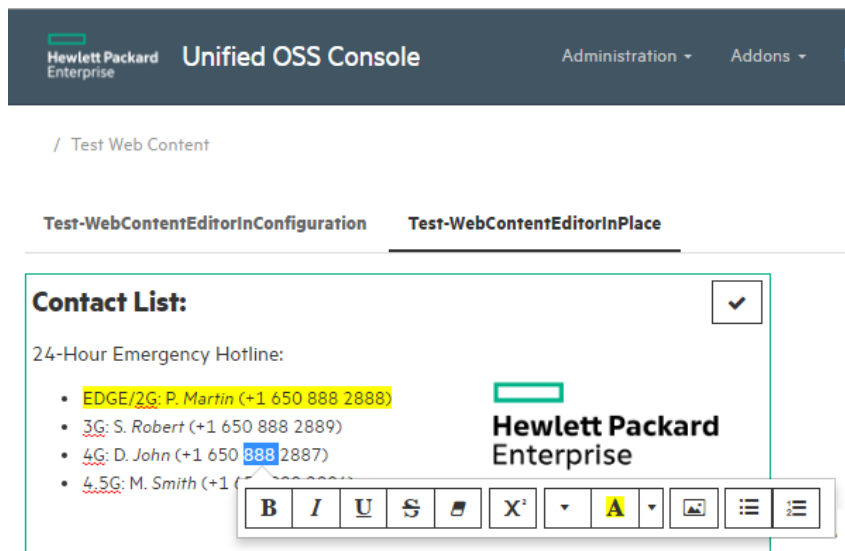


Figure 16 – Widget Web Content – Edition in place

1.18.2 JSON schema

Below the JSON schema of HPE web content widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-web-content/hpe-web-content.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "HPE web content",
  "description": "HPE web content",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the web content widget",
      "properties": {
        "content": {
          "type": "string",
          "title": "Content",

```

```

    "description": "Content of the web content"
  },
  "init": {
    "type": "object",
    "title": "Initialization",
    "description": "Initialization of the hpe web content",
    "properties": {
      "toolbar": {
        "type": "object",
        "title": "Toolbar",
        "description": "Toolbar object allow to customize the toolbar the
user will have using the web content",
        "properties": {
          "style": {
            "type": "object",
            "title": "Style",
            "description": "Toolbar buttons related to the text style",
            "properties": {
              "bold": {
                "type": "boolean",
                "title": "Bold",
                "description": "toggle font weight",
                "default": false
              },
              "italic": {
                "type": "boolean",
                "title": "Italic",
                "description": "toggle italic",
                "default": false
              },
              "underline": {
                "type": "boolean",
                "title": "Underline",
                "description": "toggle underline",
                "default": false
              },
              "strikethrough": {
                "type": "boolean",
                "title": "Strikethrough",
                "description": "toggle strikethrough",
                "default": false
              },
              "clear": {
                "type": "boolean",
                "title": "Clear",

```

```

        "description": "clear font style",
        "default": false
    }
}
},
"specialFont": {
    "type": "object",
    "title": "SpecialFont",
    "description": "Toolbar buttons related to the specialFont",
    "properties": {
        "superscript": {
            "type": "boolean",
            "title": "Superscript",
            "description": "toggle superscript",
            "default": false
        },
        "subscript": {
            "type": "boolean",
            "title": "Subscript",
            "description": "toggle subscript",
            "default": false
        }
    }
},
"fontSyle": {
    "type": "object",
    "title": "FontSyle",
    "description": "Toolbar buttons related to the text font",
    "properties": {
        "fontsize": {
            "type": "boolean",
            "title": "Fontsize",
            "description": "set font size",
            "default": false
        },
        "fontname": {
            "type": "boolean",
            "title": "Fontname",
            "description": "set font family",
            "default": false
        },
        "height": {
            "type": "boolean",
            "title": "Height",

```



```
"description": "set font height",
"default": false
},
"color": {
  "type": "boolean",
  "title": "Color",
  "description": "set foreground and background color",
  "default": false
},
"bold": {
  "type": "boolean",
  "title": "Bold",
  "description": "toggle font weight",
  "default": false
},
"italic": {
  "type": "boolean",
  "title": "Italic",
  "description": "toggle italic",
  "default": false
},
"underline": {
  "type": "boolean",
  "title": "Underline",
  "description": "toggle underline",
  "default": false
},
"strikethrough": {
  "type": "boolean",
  "title": "Strikethrough",
  "description": "toggle strikethrough",
  "default": false
},
"subscript": {
  "type": "boolean",
  "title": "Subscript",
  "description": "toggle subscript",
  "default": false
},
"clear": {
  "type": "boolean",
  "title": "Clear",
  "description": "clear font style",
  "default": false
```

```

    }
  }
},
"insert": {
  "type": "object",
  "title": "Insert",
  "description": "Toolbar buttons related to the media
insertion",
  "properties": {
    "picture": {
      "type": "boolean",
      "title": "Picture",
      "description": "open image dialog",
      "default": false
    },
    "video": {
      "type": "boolean",
      "title": "Video",
      "description": "open video dialog",
      "default": false
    },
    "link": {
      "type": "boolean",
      "title": "Link",
      "description": "open link dialog",
      "default": false
    },
    "table": {
      "type": "boolean",
      "title": "Table",
      "description": "insert a table",
      "default": false
    },
    "lineSeparator": {
      "type": "boolean",
      "title": "LineSeparator",
      "description": "insert a line separator",
      "default": false
    }
  }
},
"paragraphStyle": {
  "type": "object",
  "title": "ParagraphStyle",

```

```

style",
    "description": "Toolbar buttons related to the paragraph",
    "properties": {
        "ul": {
            "type": "boolean",
            "title": "Ul",
            "description": "toggle unordered list",
            "default": false
        },
        "ol": {
            "type": "boolean",
            "title": "Ol",
            "description": "toggle ordered list",
            "default": false
        },
        "paragraph": {
            "type": "boolean",
            "title": "Paragraph",
            "description": "dropdown for paragraph align",
            "default": false
        },
        "style": {
            "type": "boolean",
            "title": "Style",
            "description": "format selected block",
            "default": false
        },
        "height": {
            "type": "boolean",
            "title": "Height",
            "description": "set line height",
            "default": false
        }
    },
    "misc": {
        "type": "object",
        "title": "Misc",
        "description": "Toolbar buttons related to the
miscellaneous",
        "properties": {
            "undo": {
                "type": "boolean",
                "title": "Undo",
                "description": "Undo",

```

```

        "default": false
    },
    "redo": {
        "type": "boolean",
        "title": "Redo",
        "description": "Redo",
        "default": false
    },
    "codeview": {
        "type": "boolean",
        "title": "Codeview",
        "description": "toggle wysiwyg and html editing mode",
        "default": false
    },
    "fullscreen": {
        "type": "boolean",
        "title": "Fullscreen",
        "description": "toggle fullscreen editing mode",
        "default": false
    },
    "help": {
        "type": "boolean",
        "title": "Help",
        "description": "open help dialog",
        "default": false
    }
}
}
}
},
"editorInPlaceMode": {
    "type": "boolean",
    "title": "EditorInPlaceMode",
    "description": "When set to true, user can edit the web content
by double clicking on it",
    "default": false
},
"showBorder": {
    "type": "boolean",
    "title": "ShowBorder",
    "description": "Used to set a border around the web content
mandatory field to display the border",
    "default": false
},
"borderStyle": {

```

```
        "type": "string",
        "title": "BorderStyle",
        "description": "Css style of the border (solid, dotted...)
mandatory field to display the border"
    },
    "borderWidth": {
        "type": "integer",
        "title": "BorderWidth",
        "description": "Width of the border",
        "default": 3
    },
    "borderColor": {
        "type": "string",
        "title": "BorderColor",
        "description": "Color of the border",
        "default": "black"
    },
    "showBackgroundColor": {
        "type": "boolean",
        "title": "ShowBackgroundColor",
        "description": "Toggle the color of the background",
        "default": false
    },
    "backgroundColor": {
        "type": "string",
        "title": "BackgroundColor",
        "description": "Color of the background"
    }
}
}
```

1.19 HPE Widget Navigation

1.19.1 Overview

A Widget Navigation is a special widget, which acts like a widgets container to integrate several widgets and manage navigation between them. It defines a list of possible widgets. Only one widget is active at a time and they all share the same area in the view (this is different from a widget group that displays all of them and apply a layout).

A Widget Navigation is useful for example to display first a list of customers and allow the user to click on one row for to get more details. Another widget becomes active (customer details widget) and usually provides a navigation button to go back to the previous one.

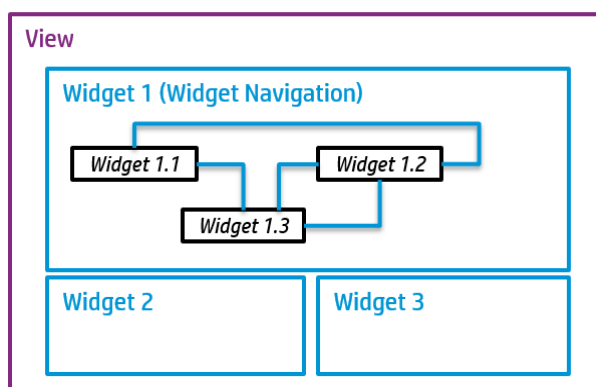


Figure 17 – Widget Navigation

1.19.2 JSON schema

Below the JSON schema of HPE widget navigation configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-widget-navigation/hpe-widget-navigation.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "HPE Widget Navigation",
  "description": "HPE Widget Navigation",
  "properties": {
    "navigationWidgets": {
      "type": "array",
      "title": "NavigationWidgets",
      "description": "This property is not part of the configuration, it is at the widget description level. It describes widgets among which the user will be able to navigate. The widgets described must emit (upon an action -- a row selection for instance) the webgui.widgetNavigate event with the target widget id as a parameter.",
      "$ref": "#widgets"
    }
  }
},
```

```
"required": [  
    "navigationWidgets"  
]  
}
```

1.20 HPE Card

1.20.1 Overview

A widget card is available to display summarized values (metric) for dashboards with a very simple representation. It also allow to apply thresholds to the panel.

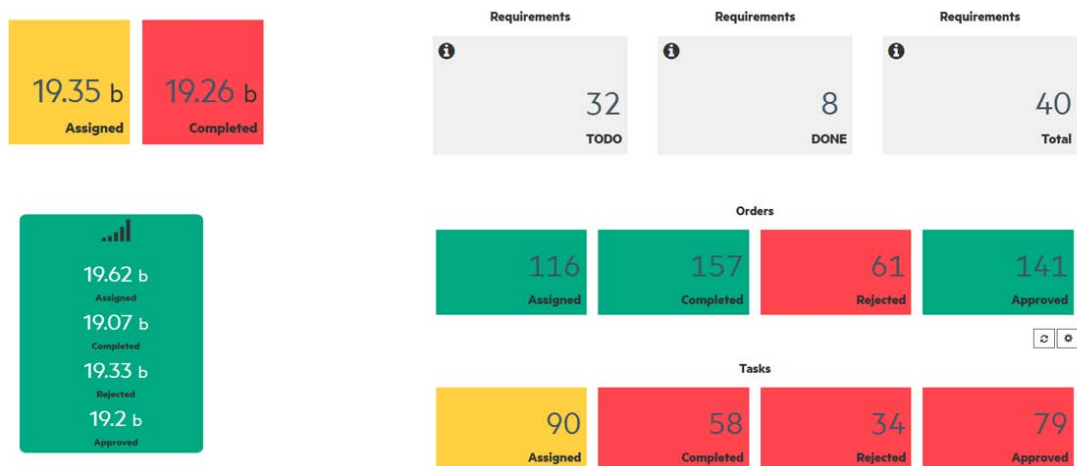


Figure 18 - Widget card

1.20.2 JSON schema

Below the JSON schema of HPE card configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-card/hpe-card.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Card",
  "description": "HPE card widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the card",
      "properties": {
        "display": {
          "type": "object",
          "title": "Display",
          "description": "Display options",
          "properties": {
            "multifact": {
```



```

        "type": "boolean",
        "title": "Multifactor",
        "description": "Display one card per dimension value"
    },
    "cardsPerRow": {
        "type": "integer",
        "title": "Cards per row",
        "description": "Number of cards per row"
    },
    "rowHeight": {
        "type": "integer",
        "title": "Row height",
        "description": "Height of the row in rem"
    }
}
},
"card": {
    "type": "object",
    "title": "Card",
    "description": "Definition of a card",
    "properties": {
        "title": {
            "type": "object",
            "title": "Title",
            "description": "Card title options",
            "properties": {
                "show": {
                    "type": "boolean",
                    "title": "Show",
                    "description": "Show title"
                },
                "size": {
                    "type": "integer",
                    "title": "Size",
                    "description": "Title size in em"
                },
                "x": {
                    "type": "integer",
                    "title": "X position",
                    "description": "X coordinates in pixels"
                },
                "y": {
                    "type": "integer",
                    "title": "Y position",

```

```

        "description": "Y coordinates in pixels"
    },
    "hAlign": {
        "type": "string",
        "title": "HAlign",
        "description": "Title horizontal alignment (left, center,
right)"
    },
    "vAlign": {
        "type": "string",
        "title": "VAlign",
        "description": "Title vertical alignment (top, middle,
bottom)"
    },
    "color": {
        "type": "string",
        "title": "Color",
        "description": "Title color (red, #FF0000...)"
    }
},
"icon": {
    "type": "object",
    "title": "Icon",
    "description": "Card icon options",
    "properties": {
        "show": {
            "type": "boolean",
            "title": "Show",
            "description": "Show icon"
        },
        "size": {
            "type": "integer",
            "title": "Size",
            "description": "Icon size in em"
        },
        "x": {
            "type": "integer",
            "title": "X position",
            "description": "X coordinates in pixels"
        },
        "y": {
            "type": "integer",
            "title": "Y position",
            "description": "Y coordinates in pixels"
        }
    }
}

```

```

    },
    "hAlign": {
      "type": "string",
      "title": "HAlign",
      "description": "Icon horizontal alignment (left, center,
right)"
    },
    "vAlign": {
      "type": "string",
      "title": "VAlign",
      "description": "Icon vertical alignment (top, middle,
bottom)"
    },
    "color": {
      "type": "string",
      "title": "Color",
      "description": "Icon color (red, #FF0000...)"
    }
  }
},
"value": {
  "type": "object",
  "title": "Value",
  "description": "Card value options",
  "properties": {
    "showValue": {
      "type": "boolean",
      "title": "Show value",
      "description": "Show card value"
    },
    "hAlign": {
      "type": "string",
      "title": "HAlign",
      "description": "Value horizontal alignment (left, center,
right)"
    },
    "vAlign": {
      "type": "string",
      "title": "VAlign",
      "description": "Value vertical alignment (top, middle,
bottom)"
    },
    "showUnit": {
      "type": "boolean",
      "title": "Show unit",
      "description": "Show value unit"
    }
  }
}

```

```

},
"showValueName": {
  "type": "boolean",
  "title": "Show value name",
  "description": "assigned, completed, approved, rejected"
},
"size": {
  "type": "integer",
  "title": "Size",
  "description": "Value size in em"
},
"numberFormat": {
  "type": "integer",
  "title": "Number format",
  "description": "Number of decimals"
},
"x": {
  "type": "integer",
  "title": "X position",
  "description": "X coordinates in pixels"
},
"y": {
  "type": "integer",
  "title": "Y position",
  "description": "Y coordinates in pixels"
},
"color": {
  "type": "string",
  "title": "Color",
  "description": "Value color (red, #FF0000...)"
},
"sizeUnit": {
  "type": "number",
  "title": "Size unit",
  "description": "Size unit in em"
},
"colorUnit": {
  "type": "string",
  "title": "Color unit",
  "description": ""
},
"sizeValueName": {
  "type": "number",
  "title": "Size value name",

```

```

        "description": "Size value name in em"
    },
    "colorValueName": {
        "type": "string",
        "title": "Color value name",
        "description": "Color value name (red, #FF0000...)"
    }
}
}
},
"customCard": {
    "type": "object",
    "title": "Custom card",
    "description": "Custom card options",
    "properties": {
        "enable": {
            "type": "boolean",
            "title": "Enable",
            "description": "Enable custom card"
        },
        "directiveId": {
            "type": "string",
            "title": "Directive id",
            "description": "Directive id of a custom card (hpe-custom-card)"
        }
    }
},
"thresholds": {
    "type": "object",
    "title": "Thresholds",
    "description": "Definition of thresholds per fact id",
    "properties": {
        "[fact_id]": {
            "type": "object",
            "title": "Fact id",
            "description": "Fact id is used as a key of thresholds object ([fact_id] must be replaced by a fact id)",
            "properties": {
                "[threshold_label]": {
                    "type": "object",
                    "title": "Threshold label",
                    "description": "Threshold label ([threshold_label] must be replaced by a label: Critical, Warning, Clear...)",
                    "properties": {

```

```

        "color": {
            "type": "string",
            "title": "Color",
            "description": "Threshold color (red, #FF0000...)"
        },
        "value": {
            "type": "integer",
            "title": "Value",
            "description": "Threshold value"
        },
        "icon": {
            "type": "string",
            "title": "Icon",
            "description": "Threshold icon font awesome class (fa fa-
tasks, fa fa-signal...)"
        },
        "display": {
            "type": "boolean",
            "title": "Banner mode",
            "description": "Display threshold color in banner"
        }
    }
}
}
}
},
"factsTitle": {
    "type": "object",
    "title": "Facts title",
    "description": "Definition of facts title",
    "properties": {
        "[fact_id]": {
            "type": "string",
            "title": "Title",
            "description": "Fact id is used as a key of factsTitle object
([fact_id] must be replaced by a fact id)"
        }
    }
},
"title": {
    "type": "string",
    "title": "Title",
    "description": "Title of hpe card widget"
},

```

```
"autoTitle": {
  "type": "boolean",
  "title": "Auto title",
  "description": "Automatically generate a title"
},
"titleSizeUnit": {
  "type": "integer",
  "title": "Title size unit",
  "description": "Title size in em"
},
"titlehAlign": {
  "type": "string",
  "title": "Title hAlign",
  "description": "Title horizontal alignment (left, center, right)"
},
"titleColor": {
  "type": "string",
  "title": "Title color",
  "description": "Title color (red, #FF0000...)"
}
}
}
}
```

1.21 HPE Iframe

1.21.1 Overview

The Iframe widget is dedicated to the integration of an external web application through its URL. It embeds the external web application in a widget making possible to integrate an external web page in UOC views alongside other UOC widgets.

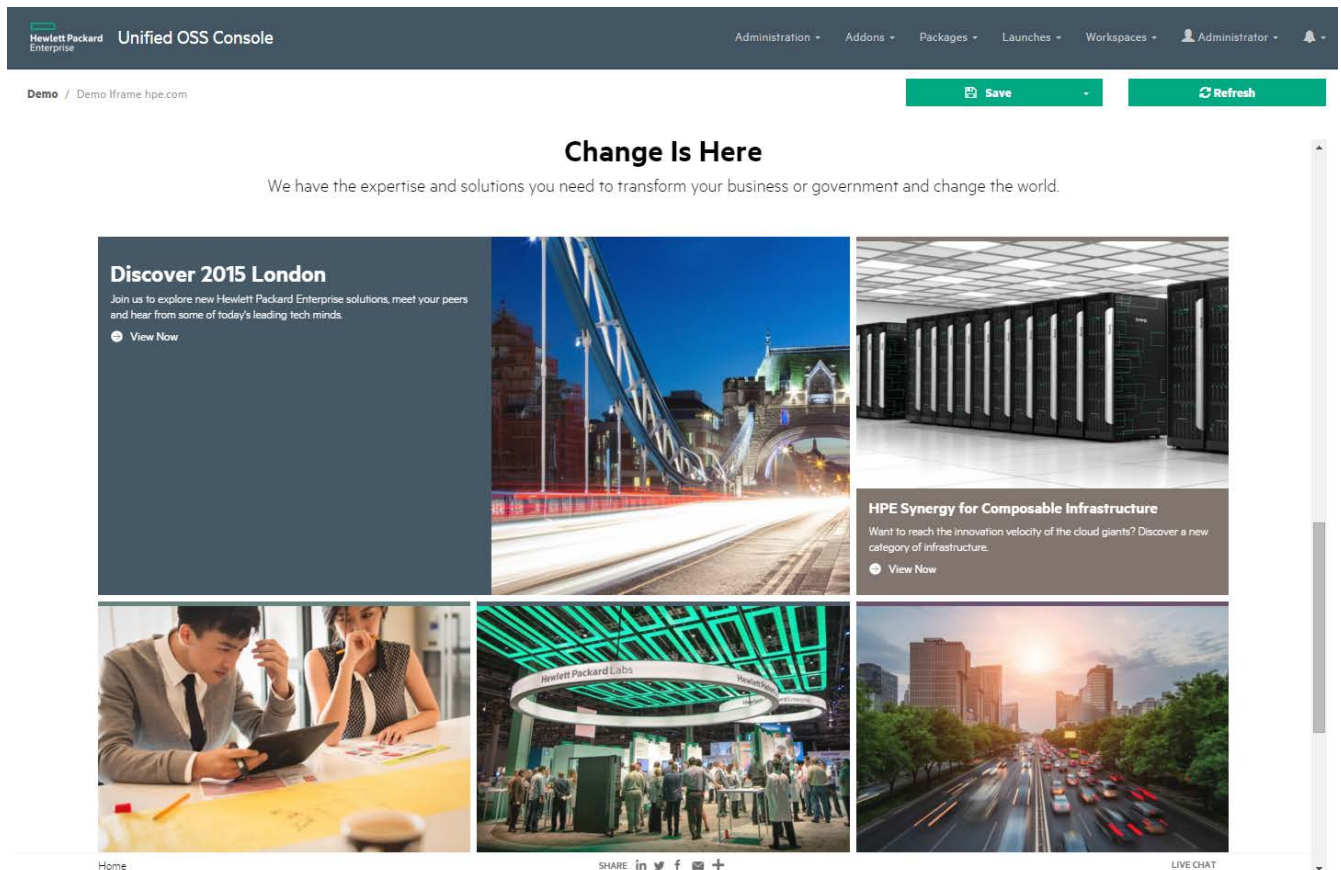


Figure 19 - Example of iframe integrating www.hpe.com web site

1.21.2 JSON schema

Below the JSON schema of HPE iframe configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-iframe/hpe-iframe.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Iframe",
  "description": "HPE iframe widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the iframe",
      "properties": {
```



```
"init": {
  "type": "object",
  "title": "Initialization",
  "description": "Initialization of the iframe",
  "properties": {
    "srcUrl": {
      "type": "string",
      "title": "Url",
      "description": "Url of the iframe (http://www.bing.com,
https://www.hpe.com...)"
    },
    "ratio": {
      "type": "string",
      "title": "Ratio",
      "description": "Ratio of the iframe: 4:3 or 16:9",
      "default": "4:3"
    }
  },
  "required": [
    "srcUrl"
  ]
}
}
```

1.22 HPE Notifications

1.22.1 Overview

The widget notifications displays one, several or all current notifications in a list. For each notification, it displays its title, level, details and associated keywords. It presents the same level of information as the notification message area.

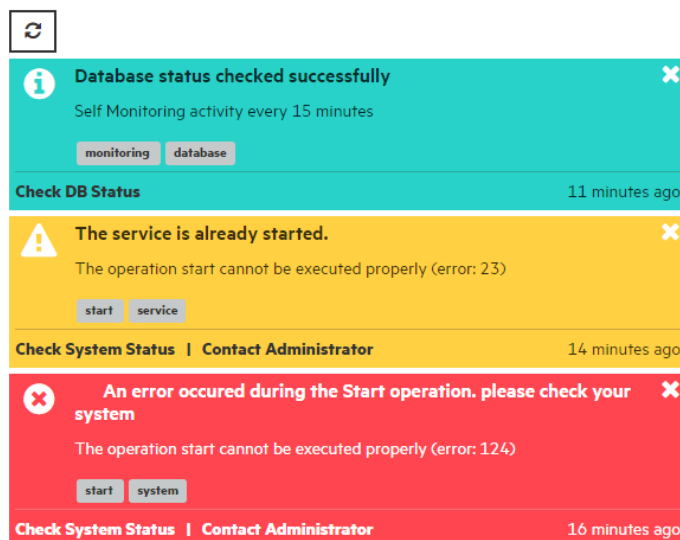


Figure 20 – Widget notification

1.22.2 JSON schema

Below the JSON schema of HPE notifications configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-notifications/hpe-notifications.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Notifications",
  "description": "HPE notifications widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the notifications",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of hpe notifications widget"
        },
        "showTitle": {
          "type": "boolean",

```

```
    "title": "Show title",
    "description": "Show the title of hpe notifications widget"
  },
  "summary": {
    "type": "boolean",
    "title": "Summary",
    "description": "Display notifications in summary mode (hide
description and actions from notifications)"
  },
  "expanded": {
    "type": "boolean",
    "title": "Expanded",
    "description": "Expand the description of notifications"
  },
  "titleColor": {
    "type": "string",
    "title": "Title color",
    "description": "Title color of hpe notifications widget"
  }
}
}
}
}
```

1.23 HPE Notifications Table

1.23.1 Overview

The notifications table is in charge of displaying all the current notification in a tabular way. This table can be customized by configuration to modify the columns visibility, order and presentation name. All the usual operations on table are also available (multiple sorting, filtering...).

Notifications Table

TimeStamp	Level	Title	Keywords	Actions
2016-06-24 16:53:58	Default	message have been sent to the op...	operator level1 message database	
2016-06-24 16:51:14	Info	Database status checked successfu...	start system	Check DB Status
2016-06-24 16:48:40	Warning	The service is already started.	start system	Check System Status Contact Adm
2016-06-24 16:46:48	Danger	An error ocurred during the Start o...	start system	Check System Status Contact Adm

1 - 4 of 4 items

Figure 21 - Widget notification table

1.23.2 JSON schema

Below the JSON schema of HPE notifications table configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-notifications-table/hpe-notifications-table.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Notifications table",
  "description": "HPE notifications table",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the ng table",
      "properties": {
        "maxRows": {
          "type": "integer",
          "title": "Max rows",
          "description": "Maximum number of rows used when vertical scrollbar is enabled",
          "default": "10"
        },
        "pagination": {
          "type": "boolean",
          "title": "Pagination",
          "description": "Enable pagination",

```

```

    "default": "false"
  },
  "paginationPageSize": {
    "type": "integer",
    "title": "Pagination page size",
    "description": "Maximum number of rows per page",
    "default": "10"
  },
  "paginationPageSizes": {
    "type": "array",
    "title": "Pagination page sizes",
    "description": "Allowed page sizes",
    "items": {
      "type": "integer",
      "title": "Page sizes",
      "description": "Items per page",
      "default": "[1,2,3,5,10,20,50,100]"
    }
  },
  "verticalScrollBar": {
    "type": "boolean",
    "title": "Vertical scrollbar",
    "description": "Enable vertical scrollbar"
  },
  "horizontalScrollBar": {
    "type": "boolean",
    "title": "Horizontal scrollbar",
    "description": "Enable horizontal scrollbar"
  },
  "enableFiltering": {
    "type": "boolean",
    "title": "Enable filtering",
    "description": "Enable filtering for all columns",
    "default": "false"
  },
  "dateFormat": {
    "type": "string",
    "title": "Date format",
    "description": "Format for all date formatted cells",
    "default": "yyyy-MM-dd HH:mm:ss"
  },
  "timeFormat": {
    "type": "string",
    "title": "TimeFormat schema.",

```

```

    "description": "Format for all time formatted cells",
    "default": "HH:mm:ss"
  },
  "numberFormat": {
    "type": "integer",
    "title": "NumberFormat schema.",
    "description": "Number of decimal places to round the number to",
    "default": "2"
  },
  "showGridFooter": {
    "type": "boolean",
    "title": "Show grid footer",
    "description": "Display a footer after the rows, containing
information about number of shown rows and total number of rows",
    "default": "false"
  },
  "enableRowSelection": {
    "type": "boolean",
    "title": "Enable row selection",
    "description": "Enable selection of rows",
    "default": "false"
  },
  "enableRowHeaderSelection": {
    "type": "boolean",
    "title": "Enable row header selection",
    "description": "Display a column at the left of the table, to check
or uncheck selected rows (RowHeader)",
    "default": "false"
  },
  "multiSelect": {
    "type": "boolean",
    "title": "Multi select",
    "description": "Enable multiple rows selection",
    "default": "false"
  },
  "enableSelectAll": {
    "type": "boolean",
    "title": "Enable select all",
    "description": "Enable button to select/unselect all rows (button is
in the RowHeader)"
  },
  "autoExport": {
    "type": "boolean",
    "title": "Auto export",
    "description": "Automatically export selection to output, display a
button to trigger export if false",

```

```

    "default": "true"
  },
  "allowCellFocus": {
    "type": "boolean",
    "title": "Allow cell focus",
    "description": "Enable selection of cells",
    "default": "false"
  },
  "showTitle": {
    "type": "boolean",
    "title": "Show title",
    "description": "Display the widget title"
  },
  "enableColumnResizing": {
    "type": "boolean",
    "title": "EnableColumnResizing schema.",
    "description": "Enable manual resizing of all columns of the table by
dragging a column header's right border",
    "default": "false"
  },
  "exporterMenuCsv": {
    "type": "boolean",
    "title": "Exporter menu csv",
    "description": "Enable the export csv entries in the table menu"
  },
  "resizeTimeout": {
    "type": "integer",
    "title": "Resize timeout",
    "description": "Resize timeout in in milliseconds"
  },
  "useExternalSorting": {
    "type": "boolean",
    "title": "Use external sorting",
    "description": "Prevents the internal sorting from executing. Events
will still be fired when the sort changes, and the sort information on the
columns will be updated, allowing an external sorter (for example, server
sorting) to be implemented",
    "default": "false"
  },
  "useExternalPagination": {
    "type": "boolean",
    "title": "UseExternalPagination schema.",
    "description": "Disables client side pagination. When true, handle
the paginationChanged event and set data and totalItems",
    "default": "false"
  },

```

```

"columns": {
  "type": "array",
  "title": "Columns",
  "description": "Columns definition",
  "items": {
    "type": "object",
    "title": "Column",
    "description": "Column definition",
    "required": [
      "displayName",
      "field"
    ],
    "properties": {
      "displayName": {
        "type": "string",
        "title": "Display name",
        "description": "Column name that will be shown in the header"
      },
      "field": {
        "type": "string",
        "title": "Field",
        "description": "Field must be provided if you wish to bind to a
property in the data source"
      },
      "visible": {
        "type": "boolean",
        "title": "Visible",
        "description": "Column visibility"
      },
      "metadata": {
        "type": "object",
        "title": "Metadata",
        "description": "An explanation about the purpose of this
instance.",
        "properties": {
          "$ref": "#facts"
        }
      },
      "allowCellFocus": {
        "type": "boolean",
        "title": "Allow cell focus",
        "description": "Enable focus on a cell within this column",
        "default": "true"
      },
      "cellFilter": {

```



```

    "type": "string",
    "title": "Cell filter",
    "description": "Filter to apply to the content of each cell"
  },
  "cellTemplate": {
    "type": "string",
    "title": "Cell template",
    "description": "Custom template for each cell in this column"
  },
  "aggregationType": {
    "type": "integer",
    "title": "Aggregation type",
    "description": "The aggregation that you'd like to show in the
columnFooter for this column (sum: 2, count: 4, avg: 8, min: 16, max: 32)"
  },
  "aggregationHideLabel": {
    "type": "boolean",
    "title": "Aggregation hide label",
    "description": "if set to true hides the label text in the
aggregation footer, so only the value is displayed",
    "default": "false"
  },
  "sort": {
    "type": "object",
    "title": "Sort",
    "description": "Sort column",
    "properties": {
      "direction": {
        "type": "string",
        "title": "Direction",
        "description": "direction values are asc, desc"
      },
      "priority": {
        "type": "integer",
        "title": "Priority",
        "description": "says what order to sort the columns in
(lower priority gets sorted first)"
      }
    }
  },
  "required": [
    "direction",
    "priority"
  ]
},
"formatter": {

```

```
        "type": "string",
        "title": "Formatter",
        "description": "Custom formatters (status, rawValue,
number...)"
    },
    "footerCellTemplate": {
        "type": "string",
        "title": "FooterCellTemplate schema.",
        "description": "Custom template for the footer for this column"
    }
}
}
}
}
}
}
```

1.24 HPE Notifications Generator

1.24.1 Overview

This widget is a simple SDK tool dedicated to add-ons developers to tune and customize their notifications. It displays the JSON definition of a notification and simulate a notification send operation.



```

1 {
2   "origin": "",
3   "domain": "",
4   "package": "",
5   "type": "",
6   "id": "",
7   "level": "danger",
8   "progress": false,
9   "title": "My title here...",
10  "data": {
11    "text": {
12      "en-us": "My description here",
13      "fr-fr": "Ma description ici"
14    },
15    "occurrences": 0
16  }

```

Send

Figure 22 – Widget notifications generator

1.24.2 JSON schema

Below the JSON schema of HPE notifications generator configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-notifications-generator/hpe-notifications-generator.json`.

```

{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Notifications generator",
  "description": "HPE notifications generator widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the notifications generator",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of hpe notifications widget"
        }
      }
    }
  }
}

```

```
    },  
    "showTitle": {  
      "type": "boolean",  
      "title": "Show title",  
      "description": "Show the title of hpe notifications widget"  
    },  
    "titleColor": {  
      "type": "string",  
      "title": "Title color",  
      "description": "Title color of hpe notifications widget"  
    }  
  }  
}  
}
```

1.25 HPE Notifications Keywords

1.25.1 Overview

The widget notifications keywords is able to display the list of all keywords available in the notification messages to ease filtering. This widget is strongly linked to a table and will drive the filtering of results based on keywords found in notification.

To help the search, it provides an auto-completion dropdown list to select.

1.25.2 JSON schema

Below the JSON schema of HPE notifications keywords configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-notifications-keywords/hpe-notifications-keywords.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Notifications keywords",
  "description": "HPE Notifications keywords",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the Notifications keywords",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of hpe notifications keywords widget"
        },
        "showTitle": {
          "type": "boolean",
          "title": "Show title",
          "description": "Show the title of hpe notifications keywords widget"
        },
        "titleColor": {
          "type": "string",
          "title": "Title color",
          "description": "Title color of hpe notifications keywords widget"
        },
        "init": {
          "type": "object",
          "title": "Initialization",
          "description": "Initialization of the Notifications keywords",

```

```
    "properties": {
      "maxResultsToShow": {
        "type": "integer",
        "title": "Max results to show",
        "description": "Maximum number of results to be displayed at a
time",
        "default": "10"
      },
      "addFromAutocompleteOnly": {
        "type": "boolean",
        "title": "Add from autocomplete only",
        "description": "Flag indicating that only tags coming from the
autocomplete list will be allowed"
      },
      "loadOnEmpty": {
        "type": "boolean",
        "title": "Load on empty",
        "description": "Flag indicating that the source option will be
evaluated when the input content becomes empty"
      },
      "loadOnFocus": {
        "type": "boolean",
        "title": "Load on focus",
        "description": "Flag indicating that the source option will be
evaluated when the input element gains focus"
      }
    }
  }
}
```

1.26 HPE Chat

1.26.1 Overview

The widget Chat is useful to quickly send messages to other users identified by their roles. It is an interactive tool for platform administrator. The message will be received in the notification area.

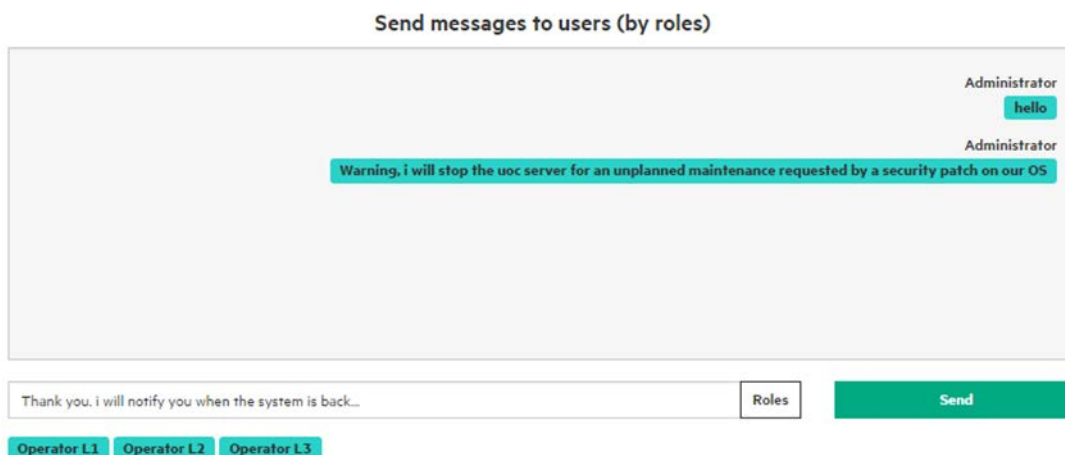


Figure 23 – Widget Chat

1.26.2 JSON schema

Below the JSON schema of HPE chat widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-chat/hpe-chat.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Chat",
  "description": "HPE chat widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the chat",
      "properties": {
        "title": {
          "type": "string",
          "title": "Title",
          "description": "Title of the chat widget"
        },
        "titleColor": {
          "type": "string",
          "title": "Title color",
          "description": "Color of the title"
        }
      }
    }
  }
}
```

```
    },
    "showTitle": {
      "type": "boolean",
      "title": "Show title",
      "description": "Show the title",
      "default": true
    },
    "enableSendToRoles": {
      "type": "boolean",
      "title": "Enable send to roles",
      "description": "Enable send to roles feature (i.e. send notification
to users having one or several roles)",
      "default": false
    },
    "notification": {
      "$ref": "#notification-composite-key"
    }
  },
  "required": [
    "notification"
  ]
}
}
```


1.27 HPE Leaflet

1.27.1 Overview

HPE Leaflet display geographical map based on latitude/longitude and rendered by external map providers. By default, the widget supports multiple map providers: including OpenStreetMap and Bing. It also supports thresholds color, custom popup dialog on click, clustering of markers, different layers and is fully integrated into the UOC communication mechanism.

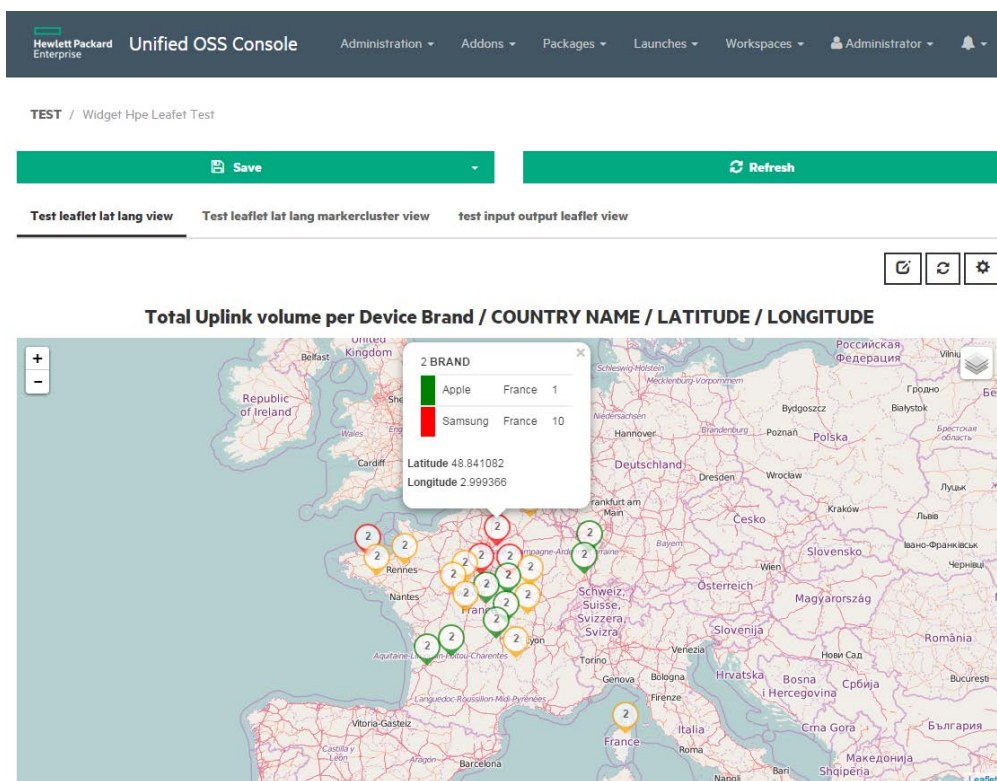


Figure 24- Widget Leaflet

1.27.2 JSON schema

Below the JSON schema of HPE leaflet configuration.

The full schema is also available in <install_dir>/client/addons/hpe/widgets/hpe-leaflet/hpe-leaflet.json.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Leaflet",
  "description": "HPE leaflet widget",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the HPE leaflet",
      "properties": {
```

```

"autotitle": {
    "type": "boolean",
    "title": "Auto title",
    "description": "Automatically generate a title"
},
"title": {
    "type": "string",
    "title": "Title",
    "description": "Title of hpe leaflet widget"
},
"width": {
    "type": "string",
    "title": "width",
    "description": "Width of hpe leaflet map"
},
"height": {
    "type": "string",
    "title": "height",
    "description": "Height of hpe leaflet map"
},
"showMarkerContent": {
    "type": "boolean",
    "title": "showMarkerContent",
    "description": "Show or hide the data markers"
},
"zoom": {
    "type": "integer",
    "title": "zoom",
    "description": "the zoom level to display the leaflet map",
    "default": "1"
},
"defaults": {
    "type": "object",
    "title": "defaults",
    "description": "the default options in hpe leaflet map",
    "properties": {
        "maxZoom": {
            "type": "integer",
            "title": "maxZoom",
            "description": "the max zoom level to display the
leaflet map",
            "default": "20"
        },
        "minZoom": {
            "type": "integer",

```

```

        "title": "minZoom",
        "description": "the min zoom level to display the
leaflet map",
        "default": "1"
    },
    "scrollWheelZoom": {
        "type": "boolean",
        "title": "scrollWheelZoom",
        "description": "the min zoom level to display the
leaflet map",
        "default": "1"
    }
},
"layersTypes": {
    "type": "array",
    "title": "layersTypes",
    "description": "the layers that can be used on the leaflet
map",
    "items": {
        "type": "string",
        "description": "the type of layer that can be used on
the leaflet map",
        "default": "openstreetmap"
    }
},
"events": {
    "type": "object",
    "title": "events",
    "description": "The event supported by the widget",
    "properties": {
        "click": {
            "type": "array",
            "title": "click",
            "description": "define click event supported by the
widget",
            "items": {
                "type": "object",
                "properties": {
                    "type": {
                        "type": "string",
                        "title": "type",
                        "description": "define the type used by
the click event",
                    }
                }
            }
        }
    }
},
"directiveId": {

```

```

        "type": "string",
        "title": "directiveId",
        "description": "define the directive
name of the module used by the click event",
    },
    "params": {
        "type": "object",
        "title": "directiveId",
        "description": "define the directive
name of the module used by the click event",
        "properties": {
            "selectDimension": {
                "type": "string",
                "title": "selectDimension",
                "description": "the dimension
Id that'll be displayed in the module",
            },
            "showLatLng": {
                "type": "boolean",
                "title": "showLatLng",
                "description": "show or hide
the latitude longitude value of each item displayed on the leaflet map",
            }
        }
    }
}

},
"mouseover": {
    "type": "array",
    "title": "mouseover",
    "description": "define mouseOver event supported by
the widget",
    "items": {
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "title": "type",
                "description": "define the type used by
the click event",
            },
            "directiveId": {
                "type": "string",
                "title": "directiveId",
                "description": "define the directive
name of the module used by the click event",
            }
        }
    }
}

```

```

    },
    "params": {
        "type": "object",
        "title": "directiveId",
        "description": "define the directive
name of the module used by the click event",
        "properties": {
            "selectDimension": {
                "type": "string",
                "title": "selectDimension",
                "description": "the dimension
Id that'll be displayed in the module",
            },
            "showLatLong": {
                "type": "boolean",
                "title": "showLatLong",
                "description": "show or hide
the latitude longitude value of each item displayed on the leaflet map",
            }
        }
    }
}
}
}
}
},
"leafletMap": {
    "type": "array",
    "title": "leafletMap",
    "description": "the layers that can be used on the leaflet
map",
    "items": {
        "type": "object",
        "properties": {
            "latLong": {
                "type": "array",
                "title": "latLong",
                "description": "Array of 2 items the couple of
dimension Id od latitude and longitude dimensions",
            },
            "clustering": {
                "type": "boolean",
                "title": "clustering",
                "description": "enable or disable the
clustering mode of markers",
            }
        }
    }
}

```

```

        }
    },
    "formula": {
        "type": "object",
        "title": "formula",
        "description": "Definition of calculated rules per fact
id",
        "properties": {
            "[fact_id]": {
                "type": "object",
                "title": "Fact id",
                "description": "Fact id is used as a key of
thresholds object ([fact_id] must be replaced by a fact id)",
            }
        },
        "thresholds": {
            "type": "object",
            "title": "Thresholds",
            "description": "Definition of thresholds per fact id",
            "properties": {
                "[fact_id]": {
                    "type": "object",
                    "title": "Fact id",
                    "description": "Fact id is used as a key of
thresholds object ([fact_id] must be replaced by a fact id)",
                    "properties": {
                        "[threshold_label]": {
                            "type": "object",
                            "title": "Threshold label",
                            "description": "Threshold label
([threshold_label] must e replaced by a label: Critical, Warning, Clear...)",
                            "properties": {
                                "color": {
                                    "type": "string",
                                    "title": "Color",
                                    "description": "Threshold color
(red, #FF0000...)"
                                },
                                "value": {
                                    "type": "integer",
                                    "title": "Value",
                                    "description": "Threshold
value"
                                }
                            },
                        },
                    },
                },
            },
        },
    },
    "icon": {

```

```

        "type": "string",
        "title": "Icon",
        "description": "Threshold icon
font awesome class (fa fa-tasks, fa fa-signal...)"
    },
    "display": {
        "type": "boolean",
        "title": "Banner mode",
        "description": "Display
threshold color in banner"
    }
}
}
}
}
}
},
}
}}
}

```

1.28 HPE Widget Menu Bar

1.28.1 Overview

A Widget Menu Bar is a widget dedicated to display menu bar provided by the Add-ons Menu Bar.

The view designer can easily integrate a menu inside a view. The Widget Menu Bar requires the Menu Bar Identifier to display a given Menu Bar.

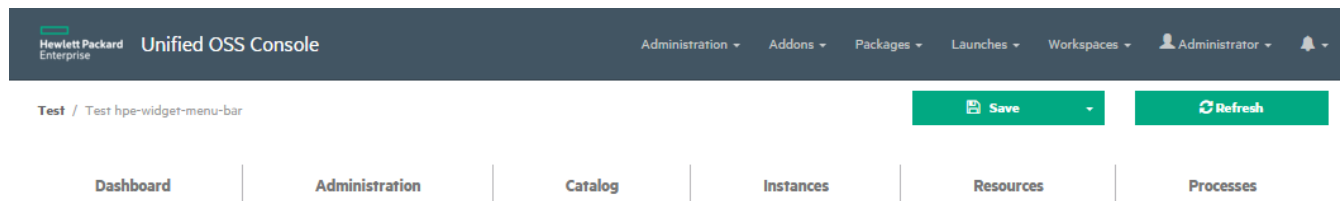


Figure 25 - Widget Menu Bar

1.28.2 JSON schema

Below the JSON schema of HPE menu bar widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-widget-menu-bar/hpe-widget-menu-bar.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Menu bar",
  "description": "HPE widget menu bar",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the menu bar",
      "properties": {
        "init": {
          "type": "object",
          "title": "Initialization",
          "description": "Initialization of hpe widget menu bar",
          "properties": {
            "menuBarId": {
              "type": "string",
              "title": "Menu bar id",
              "description": "Id of menu bar template (widget-
menu-bar-demo)"
            }
          }
        }
      }
    }
  }
}
```



```
        },  
        "separator": {  
            "type": "boolean",  
            "title": "Separator",  
            "description": "Display a vertical separator  
between menu items"  
        }  
    }  
}  
}
```

1.29 HPE Widget Tab

1.29.1 Overview

A widgets in tabular presentation. It defines a list of widgets that share the same area of the view. This widget is also able to dynamically clone an existing widget from its list and define a different configuration (filters, settings...). These changes can be saved in a workspace and restore automatically at the next reload.

A toolbar can be available by configuration to allow the end user to:

- Clone an existing widget (and create another tab)
- Delete a tab
- Move left or right
- Edit title

A widget tab is useful for example to provide a dynamic analysis cloning the same widget and applying different filters or settings.



Figure 26 - Widget Tab

1.29.2 JSON schema

Below the JSON schema of HPE tab widget configuration.

The full schema is also available in `<install_dir>/client/addons/hpe/widgets/hpe-widget-tab/hpe-widget-tab.json`.

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "type": "object",
  "title": "Tab",
  "description": "HPE widget tab",
  "properties": {
    "configuration": {
      "type": "object",
      "title": "Configuration",
      "description": "Configuration of the widget tab",
      "properties": {
        "dynamicTabEnabled": {
          "type": "boolean",
          "title": "Dynamic tab enabled",
          "description": "Show dynamic toolbar (actions available with the toolbar: delete, edit, clone, move left, move right)"
        },
        "tabWidgets": {
```

```
    "type": "array",
    "title": "Tab widgets",
    "description": "List of widgets",
    "items": {
      "type": "object",
      "title": "widget",
      "description": "Definition of a widget",
      "properties": {
        "$ref": "#widgets",
        "tabActive": {
          "type": "boolean",
          "title": "Tab active",
          "description": "Set widget tab as active"
        }
      }
    }
  },
  "title": {
    "type": "string",
    "title": "Title",
    "description": "Title of hpe widget tab"
  },
  "showTitle": {
    "type": "boolean",
    "title": "Show title",
    "description": "Show the title of hpe widget tab"
  },
  "titleColor": {
    "type": "string",
    "title": "Title color",
    "description": "Title color (red, #FF0000...)"
  }
}
}
```