

Chart

Definition

Chart (Formerly known as [SQL Chart](#)) allows you to define your own SQL query to determine the charts that you would like to generate, ranging from a number of graph types.

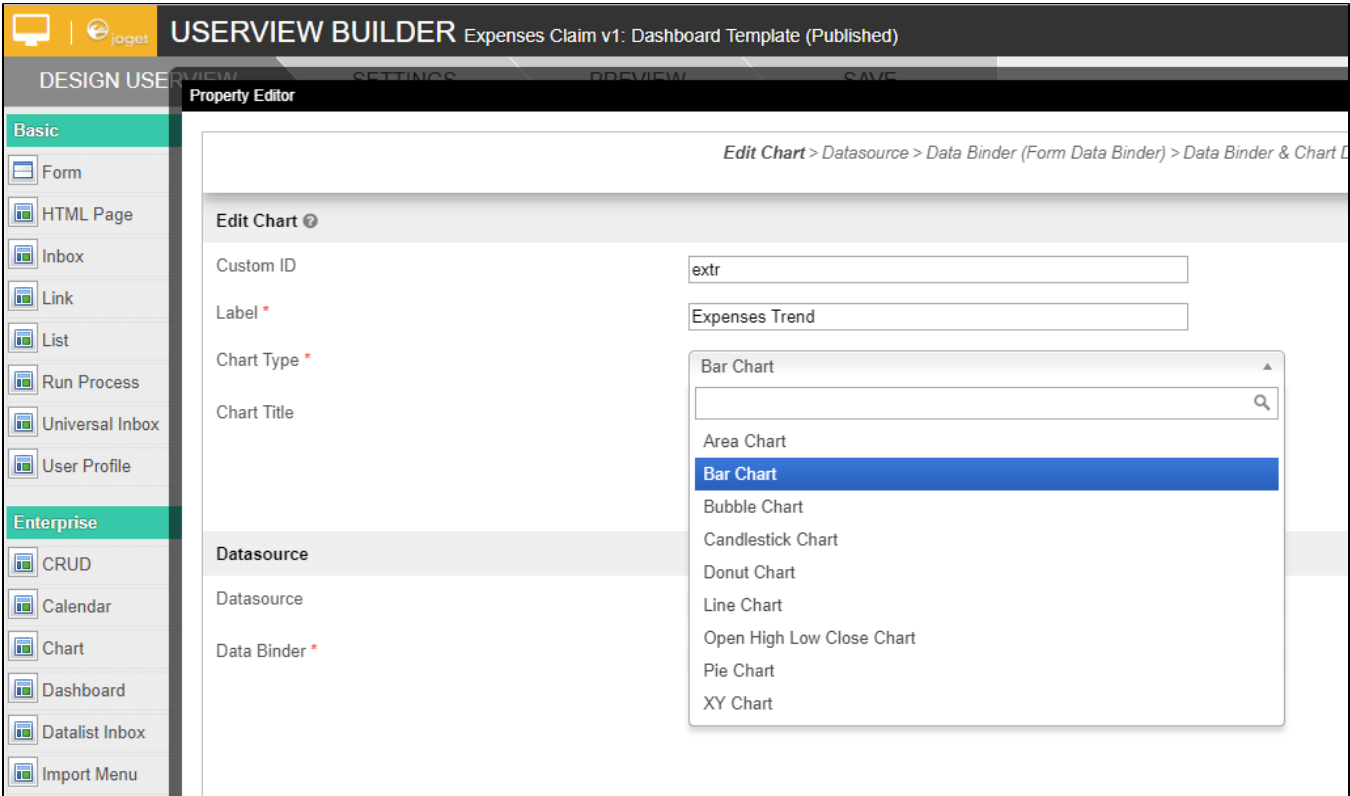


Figure 1: SQL Chart Properties

Name	Description
Custom ID	Item link slug. Optional field. Unique field Value defined here must be unique to the rest of the Userview Menus as the first matching name will be called upon.
Label	Menu label. Mandatory field.
Chart Type	- Area Chart - Bar Chart - Bubble Chart - Candlestick Chart - Donut Chart - Line Chart - Open High Low Close Chart (OHLC Chart) - Pie Chart - XY Chart

Chart Title	Chart Title to be displayed part of the generated graph. New Feature This feature has been enhanced in Joget Workflow v6 to support join, group and aggregate function.
-------------	---

Datasource

Datasource

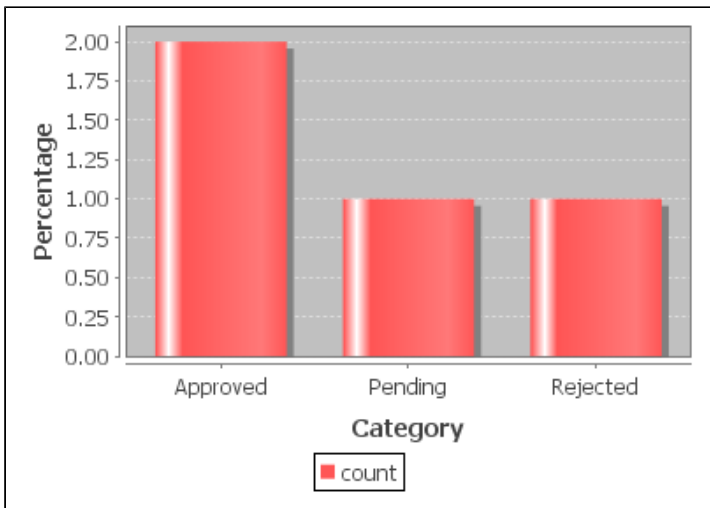
Using Data Binder

Data Binder *

Form Data Binder

Figure 2: SQL Chart Properties - Datasource

Name	Description
Datasource	- Using Data Binder - Default Datasource - Custom Datasource New Feature New feature in Joget Workflow for more flexible ways to build chart dataset using the existing Datalist Binders.
Data Binder	When Datasource is set to "Using Data Binder", this option will show up. Please see Datalist Binder for available binder to use.
SQL Query	When Datasource is set to use any of the "Datasource", this option will show up. SQL Query to produce the dataset required for the graph type. The first column in the dataset will be assumed for X-axis/label. The first column to be returned from the query must be a label (X-axis), followed by value columns for the Y-axis. Example 1: Example SELECT c_status as 'status', COUNT(c_status) as 'count' FROM app_fd_tix_tickets WHERE c_status IS NOT NULL GROUP BY c_status

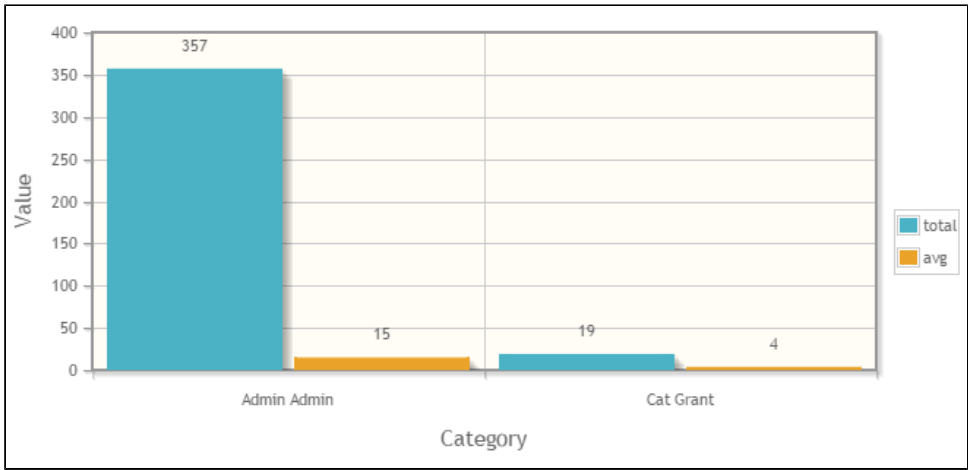


Example 2:

```
SQL

select c.c_claimant, sum( cast( replace(c.c_total,'$', '') as decimal(10,2)) ) as 'total', avg( cast(
replace(c.c_total,'$', '') as decimal(10,2)) ) as 'avg' from app_fd_hr_expense_claim c group by c.
c_claimant
```

app_fd_hr_expense_claim (3x2)		
c_claimant	total	avg
Admin Admin	357.00	14.875000
Cat Grant	19.00	3.800000



Data Binder & Chart Data Mapping

Order By

Date Created

Order

ASC

X-axis Value *

Title
(Example: China Travel, July Claims etc.)

Y-axis Values *

Number Value *

Total Amount \$

Legend Value *

Total

+

↑

↓

×

Figure 3: SQL Chart Properties - Data Binder & Chart Data Mapping (applicable for - using Data Binder)

Name	Description
Order By	Column to be sorted in the graph dataset. This would affect how the graph is plotted.
Order	• ASC • DESC
X-axis Value	X-axis label.
Y-axis Values	Y-axis dataset.

Chart Options

Edit SQL Chart > Chart Options > SQL > UI > Advanced

X-axis Label *

Category

X-axis display as

Category

Y-axis Label *

Value

Y-axis Prefix

Show Legend?

☒

Show Value Label in Chart?

☒

Stack Series?

☐

Display as Horizontal Chart?

☐

Width *

100%

Height *

80%

Colors

Figure 4: SQL Chart Properties - Chart Options (applicable for - using Datasource)

Name	Description
X-axis Label	X-axis Label

X-axis display as	• Category • Number • Date
Y-axis Label	Y-axis Label
Y-axis Prefix	Y-axis Prefix
Show Legend?	If checked, legend will be shown in the generated graph.
Show Value Label in Chart?	If checked, value label will be shown in the generated graph.
Stack Series?	If checked, this will affect the generated graph.
Display as Horizontal Chart?	Display as Horizontal Chart.
Width	Width in character. Example 100%
Height	Height in character. Example 300px

Colors

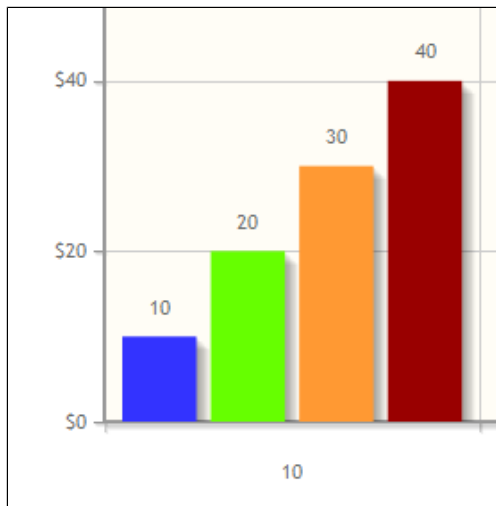
Series color. Optional field.

Comma separated values (CSV) of color codes.

Example:

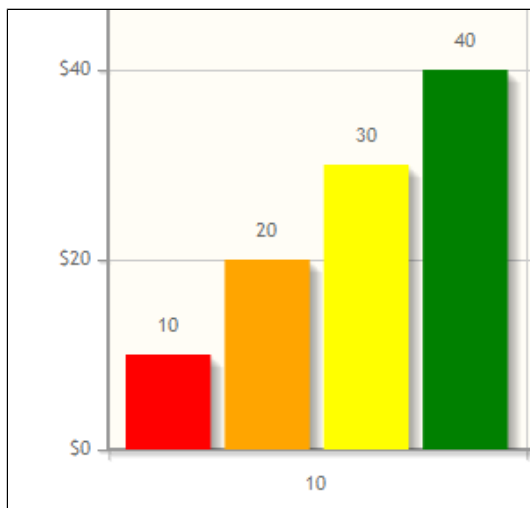
Example

#3333FF, #66FF00, #FF9933, #990000



Example

red, orange, yellow, green, blue, indigo



What are the default colors?

<http://stackoverflow.com/questions/18332440/what-are-the-jqplot-default-series-colors>

Advanced

Userview Key Data Filter

Userview Key Name

UI

Custom Header

1

Custom Footer

1

Figure 5: SQL Chart Properties - Advanced

Name	Description
Userview Key Name	When defined, additional condition will be appended using the value defined here as the parameter and the userview key value as the value. Example SQL: <code>SELECT category, count(category) FROM table1</code> Userview Key Name: <code>type</code> Userview Key Value: <code>val</code> Resultant SQL: <code>SELECT category, count(category) FROM table1 WHERE type = 'val'</code> When userview key value is defined, you may define #userviewKey# in your SQL query to have it replaced with the userview key value. Example SQL: <code>SELECT category, count(category) FROM table1 WHERE type = '#userviewKey#'</code> Userview Key Value: <code>val</code> Resultant SQL: <code>SELECT category, count(category) FROM table1 WHERE type = 'val'</code>
Custom Header	Custom Header in HTML.
Custom Footer	Custom Footer in HTML.

Interactive Chart

The following code can be modified and put in "Custom Footer" for interactive Chart.

```

<script>
$(document).ready(function(){
    $('#jq_plot_chart').bind('jqplotDataClick',
        function (event, seriesIndex, pointIndex, data) {
            console.log(event);
            console.log(seriesIndex);
            console.log(pointIndex);
            console.log(data);

            //for chart which used legend and x-axis,
            var xaxis = $(".jqplot-xaxis-tick:eq("+pointIndex+")");
            var series = $(".jqplot-table-legend-label:eq("+seriesIndex+")");

            console.log("x-axis :" + xaxis.text());
            console.log("series :" + series.text());
        }
    );

    //for double click event. Please note the arguments are different.
    $('#jq_plot_chart').bind('jqplotDbClick',
        function (event, coordinate, points, data) {
            console.log(event);
            console.log(coordinate);
            console.log(points);
            console.log(data);

            if (data) {
                var xaxis = $(".jqplot-xaxis-tick:eq("+data.pointIndex+")");
                var series = $(".jqplot-table-legend-label:eq("+data.seriesIndex+")");

                console.log("values :" + data.data);
                console.log("x-axis :" + xaxis.text());
                console.log("series :" + series.text());
            }
        }
    );
});
</script>

```



This code does not work with OHLC and candlestick chart.

Performance

You can configure the **Performance** settings in this Userview Element which allows one to cache existing content for improved performance and loading speed. Read more at [Performance Improvement with Userview Caching](#).