Chart Menu

Introduction

Chart menu allows you to select a form binder or define your own SQL query to display the chart data for the most common graph types. You can also include charts in your uservieW Dashboard Menu.

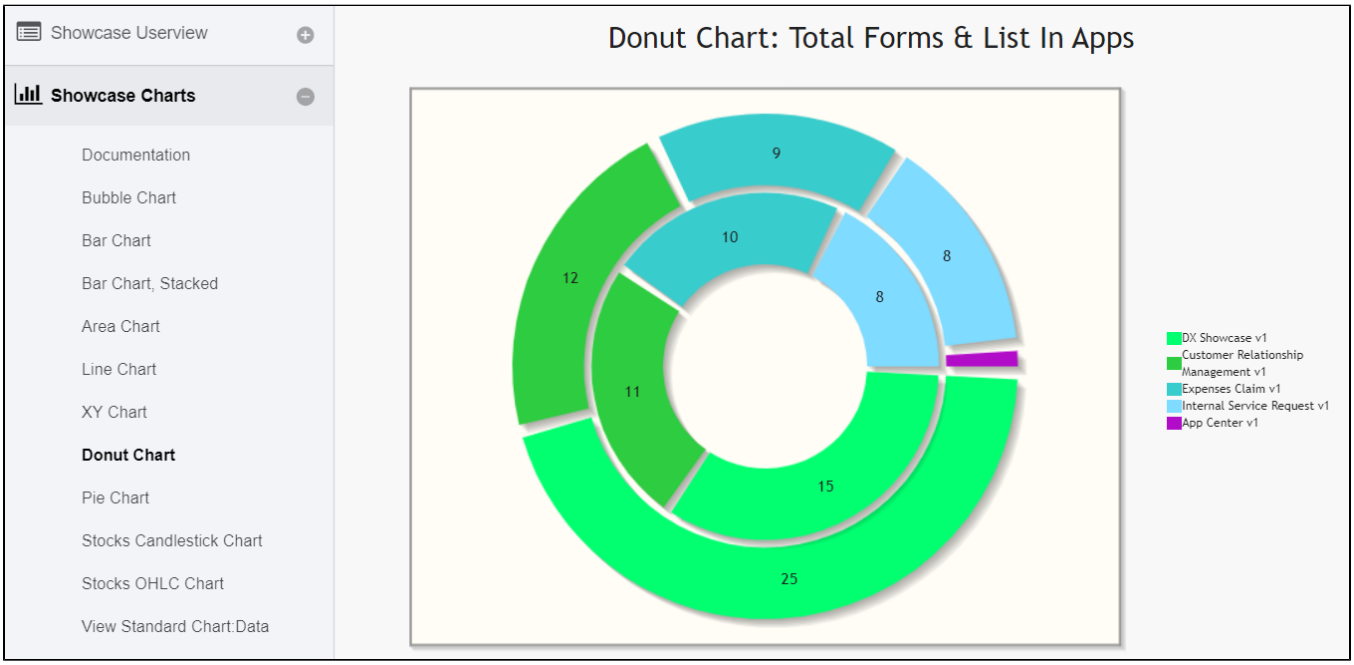


Chart Menu Properties

Configure Chart Menu

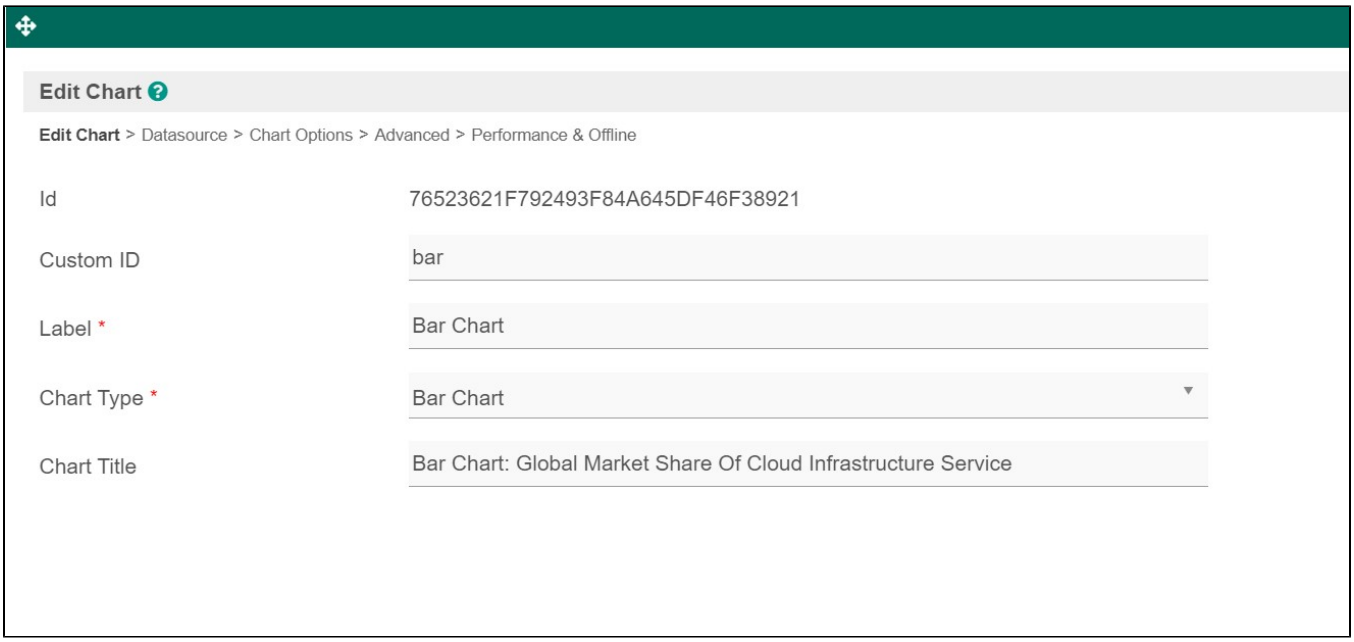


Figure 1: Edit Chart Properties

Name	Description
ID	Menu element unique id. Userview will use this id in the URL for the menu if the Custom ID is empty.
Custom ID	Item link slug. Optional 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 as part of the generated graph.

Configure Datasource

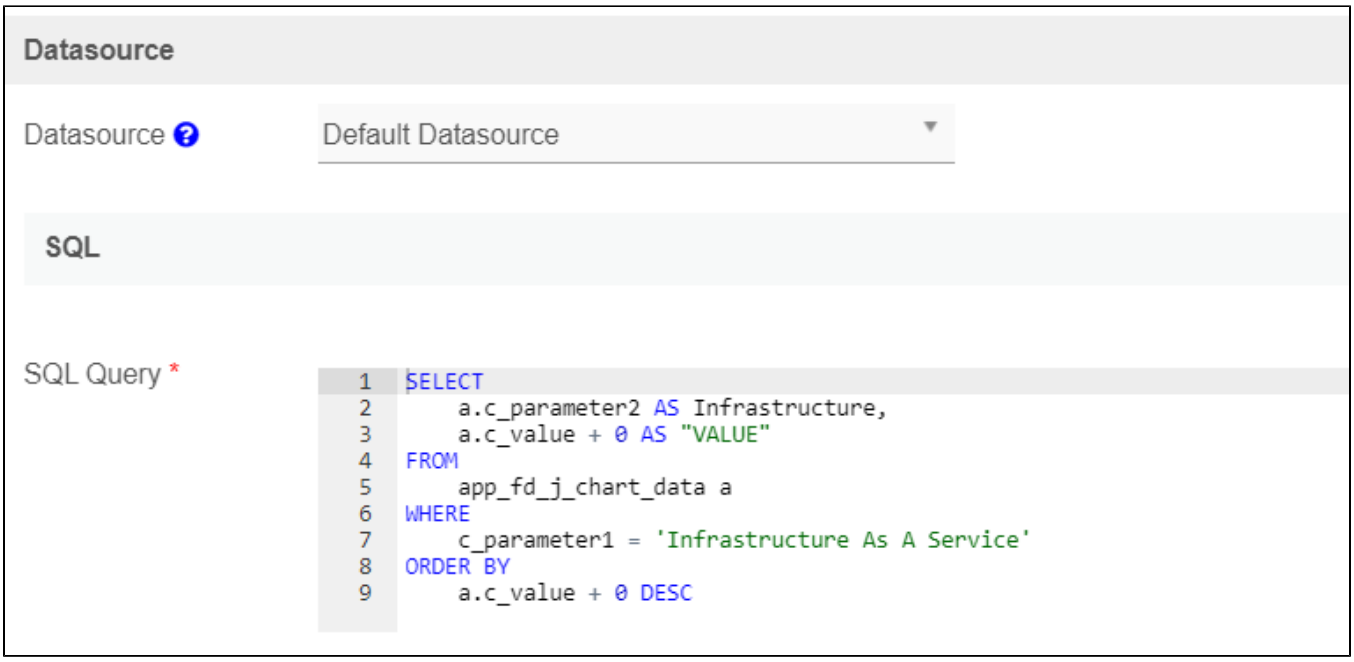
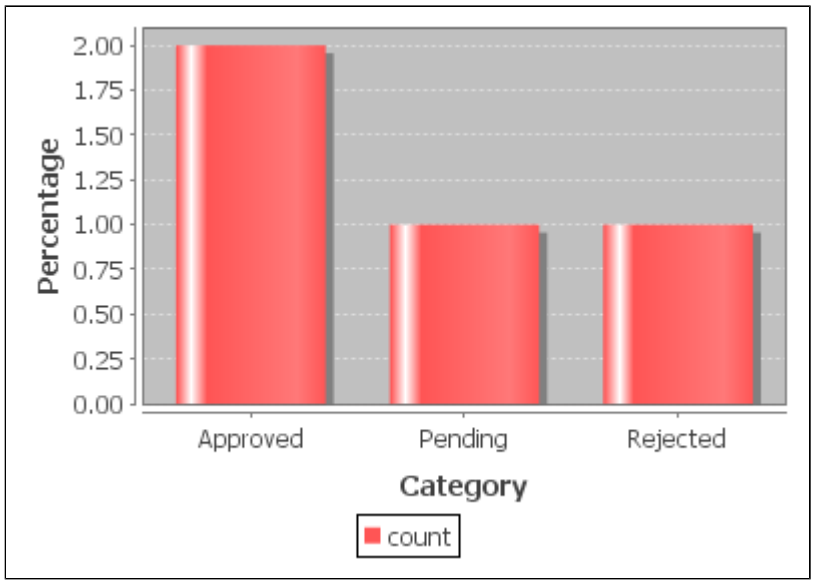


Figure 2: SQL Chart Properties - Datasource

Name	Description
Datasource	• Using Data Binder • Default Datasource • Custom Datasource
Data Binder	When Datasource is set to "Using Data Binder", this option will show up. Advance Form Data Binder has more flexible ways to build chart datasets using join, group, and aggregate function. Please see Datalist Binder for the available binder to use.

SQL Query	When Datasource is set to use any of the "Datasource", this option will show up. You use an SQL Query to produce the dataset required for the graph type. The first column in the dataset will be assumed for the X-axis/label.
	
	Example 1: The first column to be returned from the query must be a label (X-axis), followed by value columns for the Y-axis.
	
	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`



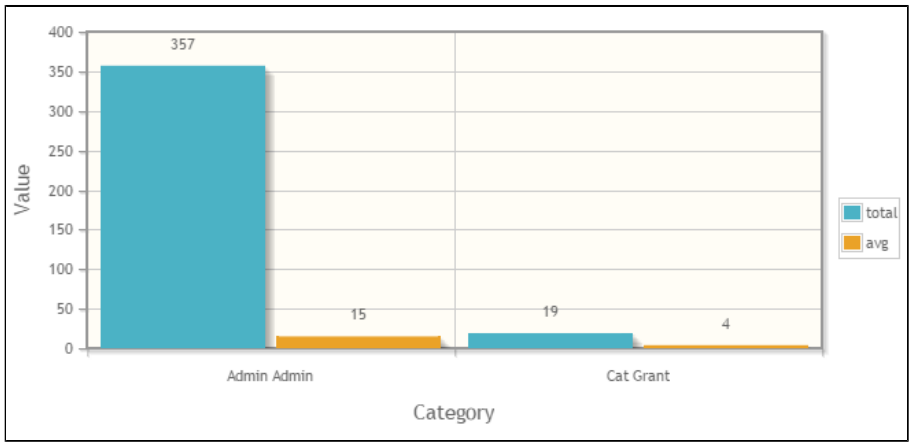
Example 2:

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_j_expense_claim c
GROUP BY
    c.c_claimant
```

app_fd_hr_expense_claim (3×2)

c_claimant	total	avg
Admin Admin	357.00	14.875000
Cat Grant	19.00	3.800000



Data Binder & Chart Data Mapping

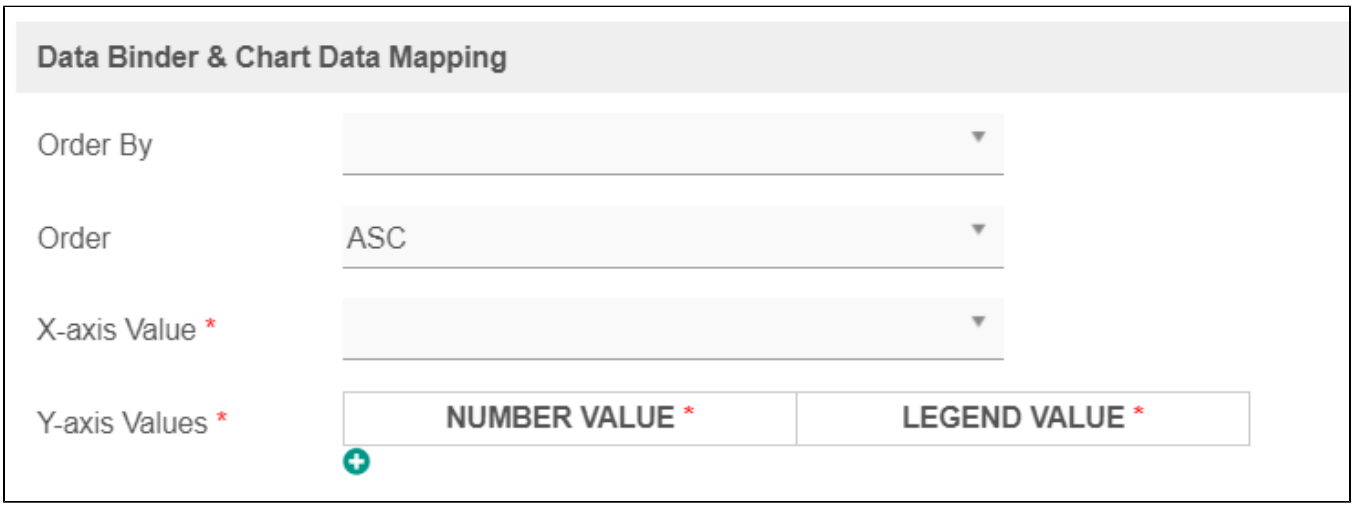


Figure 3: Data Binder & Chart Data Mapping Properties

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 Properties

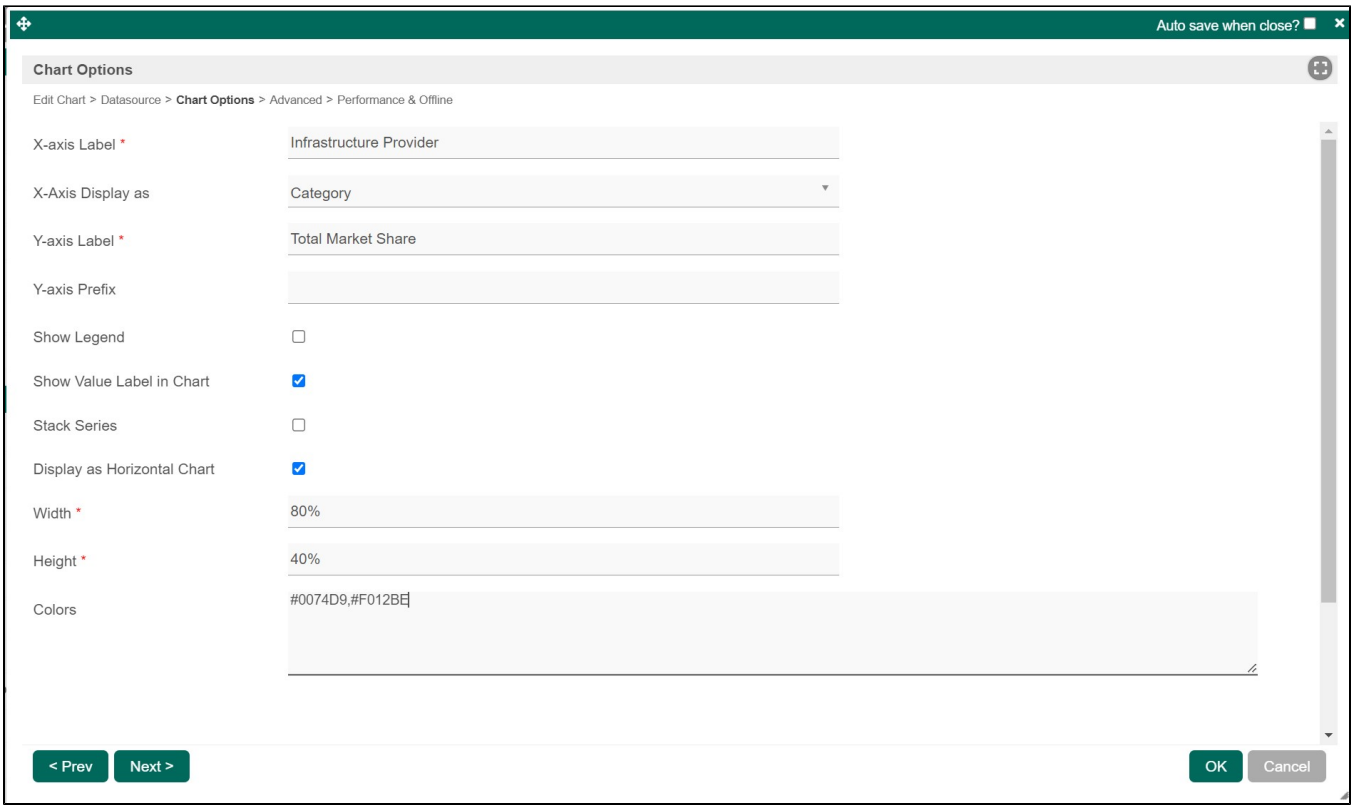
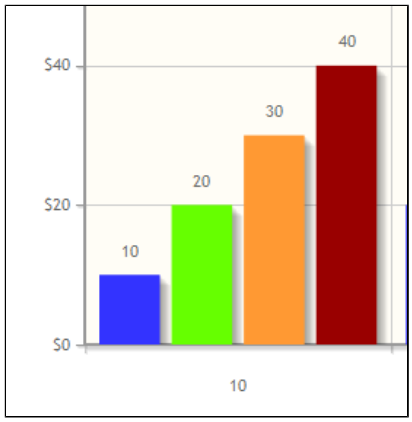
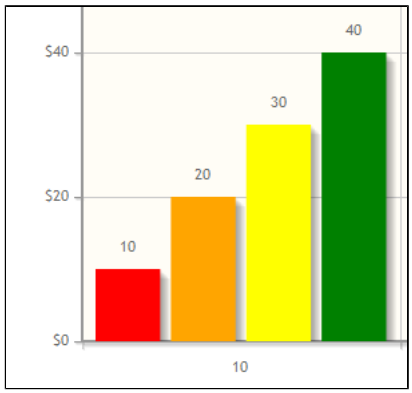


Figure 4: Chart Options

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, the legend will be shown in the generated graph.
Show Value Label in Chart?	If checked, the 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, for example 100%.
Height	Height in character, example 300px.

Colors	Series color. Optional field. Comma-separated values (CSV) of color codes. **Example 1**: #3333FF,#66FF00,#FF9933,#990000  **Example 2**: red,orange,yellow,green,blue,indigo  **What are the default colors?** http://stackoverflow.com/questions/18332440/what-are-the-jqplot-default-series-colors

Advanced Properties

Figure 5: Advanced Properties

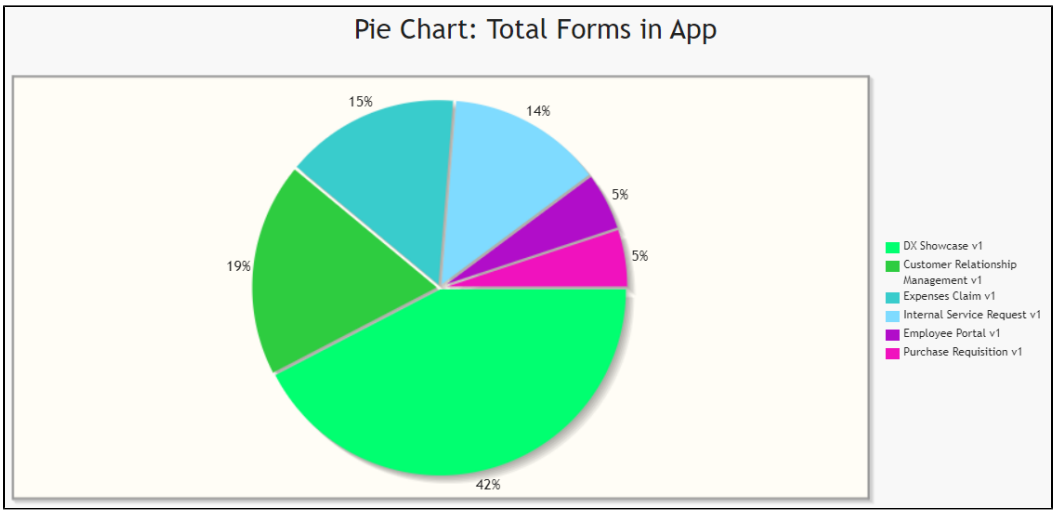
Name	Description
Userview Key Name	When defined, the additional conditions will be appended using the value defined here as the parameter and the userview key value as the value. **Example** **SQL**: SELECT category, count(category) FROM table1 **Userview Key Name**: type **Userview Key Value**: val **Resultant SQL**: SELECT category, count(category) FROM table1 WHERE type = 'val' 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**: SELECT category, count(category) FROM table1 WHERE type = '#userviewKey#' **Userview Key Value**: val **Resultant SQL**: SELECT category, count(category) FROM table1 WHERE type = 'val'
Custom Header	Custom Header in HTML.
Custom Footer	Custom Footer in HTML.

Performance & Offline Properties

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.

Additional Notes & Modifications

Display Labels Outside the Pie Chart



The following code can be modified and put in "Custom Header" for displaying the labels outside of the pie chart.

```
<script>
    $(function(){
        $.jqplot.preParseOptionsHooks.push(function(args){
            args.seriesDefaults.rendererOptions.dataLabelPositionFactor = 1.05;
        });
    });
</script>
```

The charts are plotted using jqPlot. Head over to their website here to see the full list of available hooks for customization.

Hide Gridlines from Chart Plot

The following code can be modified and put in "Custom Header" property, to hide gridlines from the chart plot.

```
<script>
    $(function(){
        $.jqplot.preParseOptionsHooks.push(function(args){
            args.axesDefaults.drawMajorGridlines = false;
        });
    });
</script>
```

Interactive Chart

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

```
<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('jqplotDblClick',
        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 charts.

Download Demo Apps

Available in Joget Marketplace

- Candlestick Charting
- Tutorial App: Joget Workflow Charts
- Joget DX Showcase APP_jshowcase_dx.jwa