

JavaScript

- -
 -
 - JavaScript
 - generateChart
 - JavaScript
- RequireShim
- Options JSON
 -
 - divSelector (options.divSelector)
 - chartId (options.chartId)
 - errorCallback (options.errorCallback)
 - dataset (options.dataset)
 - chart_information (options.dataset.chart_information)
 - metadata (options.dataset.metadata)
 - data (options.dataset.data)
 - attributes (options.dataset.attributes)
- PDF
 - blockRenderComplete
 - renderComplete
 -
- -
-

JavaScript 2JavaScriptCSSCSS

JavaScriptYellowfin/js/chartingLibraries/JavaScript

CSSYellowfinmycompanyname_chart_textcolour

JavaScript	JavaScript
CSS	CSS

JavaScript 2

- JavaScript
- JavaScript

JavaScript JavaScript

JavaScript

JavaScript

JavaScript

- JavaScript
- JavaScript JavaScriptCSS
- JavaScript

JavaScript JavaScript

generateChart

JavaScript **generateChart**YellowfinJavaScript **generateChart**

YellowfinOptions JSONgenerateChart

divOptions. divSelectorID

JavaScript

JavaScriptOptionsdiv

```
generateChart = function(options) {  
  
    // This is the div you draw your chart  
    into  
    var $chartDrawDiv = $(options.  
divSelector);  
  
    // This gets the height and width from  
    the dataset. Use these when creating your  
    chart so that  
    // it will fit the dashboard, canvas and  
    storyboard correctly  
    var height = options.dataset.chart_  
information.height;  
    var width = options.dataset.chart_  
information.width;  
  
    $chartDrawDiv.height(height);  
    $chartDrawDiv.width(width);  
  
    // This sets the background colour to  
    purple  
    $chartDrawDiv.css({  
        'background-color': 'purple'  
    });  
}
```

RequireShim

JavaScript **Require**JavaScript API

Require

```

require(['js/chartingLibraries/d3/d3'],
function(d3) {
    // Your code here which uses d3
});

```

AMD JavaScript Shim

ShimJS

ShimAMDVizJS

```

requirejs.config({
    shim: {
        'js/chartingLibraries/viz.
v1.1.2.min': {
            // once loaded, use this as the
            global module value
            exports: 'viz'
        }
    }
});

require(['js/chartingLibraries/d3_3.5.17/
d3_3.5.17', 'js/chartingLibraries/viz.
v1.1.2.min'], function(d3,viz) {

```

Options JSON

```



```

```

    },
    ...
  ],
  "demographic": [
    {
      "formatted_data": "Culture",
      "raw_data": "Culture"
    },
    {
      "formatted_
data": "Relaxation",
      "raw_data": "Relaxation"
    },
    ....
  ],
  "attributes": {
    "default_colours": [
      "#009eec",
      "#7cbb00",
      ....
    ],
    "orgrefcode_definitions":
      "Adventure": {
        "colour": "#009eec",
        "display_text": "Adventure",
        "sort_order": "2"
      },
      "Culture": {
        "colour": "#c63828",
        "display_text": "Culture",
        "sort_order": "3"
      },
      ....
    }
  }
}

```

divSelector (options.divSelector)

divIDCSSjQueryD3CSS

jQuerydiv

var \$chartDrawDiv = \$(options.divSelector);

chartId (options. chartId)

Yellowfin ID

errorCallback (options.errorCallback)

Yellowfin.JavaScript Yellowfin

dataset (options. dataset)

options.dataset [API](#) [API](#)

dataset

chart_information (options. dataset. chart_information)

- JavaScript APIJavaScript
-

metadata (options. dataset. metadata)

data_type	TEXTNUMERIC
field_name	
metric_colour	HEX#FFFFFFmetric_colour
org_refcode	JavaScript DEMOGRAPHIC orgrefcode_definitions

data (options. dataset. data)

raw_data	
formatted_data	options.dataset.data.camp_regionoptions.dataset.data.camp_rating

attributes (options. dataset. attributes)

default_colours	
orgrefcode_definitions	

PDF

JavaScript XHRCORSGoogleSheetsPDF

Yellowfin.JavaScript PDF

JavaScript YellowfinYellowfinblockRenderCompleterender Complete

blockRenderComplete

PDFJavaScript

blockRenderCompleteJS renderComplete

renderComplete

PDF

JS blockRenderCompleteblockRenderCompleterenderComplete

\$chartDrawDiv

```
generateChart = function(options) {
  options.blockRenderComplete(); //Prevents this report from being marked as renderComplete until
  "renderComplete" is also called;
  var $chartDrawDiv = $(options.divSelector);

  setTimeout(() => {
    $chartDrawDiv.text('Hello World');
    options.renderComplete();
  }, 5000);
}
```

5Hello WorldPDF

require

```

generateChart = function(options) {
    options.blockRenderComplete(); //Prevents this report being marked as renderComplete until the "options.
renderComplete" function is also called
    var $chartDrawDiv = $(options.divSelector);
    var processedData = processData(options.dataset.data);
    doDrawing(processedData, $chartDrawDiv, options.dataset.chart_information.height, options.dataset.
chart_information.width, options.renderComplete); //Pass through the options.renderComplete function to
doDrawing, so it can call the function when it finishes
},
processData = function(dataset) {
    var barDs = {
        labels : [],
        datasets : [
            {
                label : 'Camp Region',
                data : [],
                backgroundColor : ['red','green','purple','yellow','black']
            }
        ]
    };

    for (var i = 0; i < dataset.camp_region.length; i++) {
        barDs.labels.push(dataset.camp_region[i].formatted_data);

        barDs.datasets[0].data.push(dataset.camp_rating[i].raw_data);
        barDs.datasets[0].backgroundColor.push('#009EEC');
    }

    return barDs;
},

doDrawing = function(data, $chartDiv, height, width, renderComplete) {
    require(['js/chartingLibraries/chartjs/Chart.js'], function(chart) {
        var $canvas = $('<canvas height="500" width="500"></canvas>');

        $chartDiv.append($canvas);

        new Chart($canvas, {
            type: "polarArea",

            data: data,
            options : {
                responsive : false
            }
        });
        renderComplete(); //Once the chart drawing has completed, call renderComplete to notify the report this
chart is finished
    });
}

```

JavaScript

- JavaScript
- CSS
-
- JavaScript

JavaScript JavaScriptCSS

JavaScript generateChartJavaScript IDdivSelectorerrorCallback

CSS div

try/catch

Yellowfin

JavaScript

processDatademographiccamp rating

```
processData = function(dataset) {  
    var newDataset = {};  
    for(var i=0; i<dataset['camp_rating'].  
length; i++){  
  
        // Gets the demographic value from  
the dataset for this data row  
        var demographic =  
dataset['demographic'][i].formatted_data;  
  
        // Either adds the demographic to our  
dataset with the current  
        // camp rating, or retrieves an  
existing total for this  
        // demographic and adds the camp  
rating value  
        if(newDataset.  
hasOwnProperty(demographic)){  
            newDataset[demographic] +=  
dataset['camp_rating'][i].raw_data;  
        } else {  
            newDataset[demographic] =  
dataset['camp_rating'][i].raw_data;  
        }  
    }  
  
    return newDataset;  
}
```


processData

-
- /

D3.js SunburstBubble Cloudflare.json options. dataset. dataYellowfin

- 1.
- 2.

```
    var datas = options.dataset.data;

    // transform each field to contain
    it's name
    var fields = Object.keys(datas).
    map(function(fieldName) {
        return datas[fieldName].
    map(function(value) {
        let object = {};
        object[fieldName] = value;
        return object;
    });
    });

    // use d3.zip to transpose
    var zipped = d3.zip.apply(null,
    fields);

    // merge objects in array
    var result = zipped.
    map(function(objects) {
        return Object.assign.apply(null,
    objects);
    });

    return result;
```

GoogleCharts

```
[ 'Year', 'Sales', 'Expenses', 'Profit'],  
  
[ '2014', 1000, 400, 200],  
  
[ '2015', 1170, 460, 250],  
  
[ '2016', 660, 1120, 300],  
  
[ '2017', 1030, 540, 350]
```

options.dataset.datademographic invoiced amount

```
var ds = [[options.dataset.  
metadata.demographic.fieldname,  
options.dataset.  
metadata.invoiced_amount.fieldname]];  
  
var dataset = options.dataset.  
data;  
  
// push data into ds  
for (var i = 0; i < dataset.  
demographic.length; i++) {  
    ds.push([  
        dataset.  
demographic[i].formatted_data,  
parseFloat(dataset.  
invoiced_amount[i].raw_data)  
    ]);  
}  
  
return google.visualization.  
arrayToDataTable(ds);
```

JavaScript

- JavaScriptdebuggerJavaScript

- CSSjsChartNoOverflowYellowfin div

```
$(options.divSelector).  
addClass('jsChartNoOverflow');
```

try/catchYellowfinJavaScript

```
try {  
    // Your chart draw code  
    here  
} catch(err){  
    options.  
    errorCallback(err);  
}
```

- JavaScript

```
console.log(result);
```

- JavaScript Options JSONJSON

```
JSON.stringify(options);
```

JSONJSON

<https://jsonformatter.curiousconcept.com/>