

# JavaScript/HTML/Xpages

Fetch CSV records into Chart

# Google Chart API

## System Requirements:

Download Domino Designer **8.5.3** Environment (DDE)

<http://www.ibm.com/developerworks/downloads/ls/dominodesigner/>

Download /Make available Google Chart API

<https://developers.google.com/chart/?hl=uk>

Download jQuery-csv (jQuery Plugin) / jquery.csv-0.71 / jquery.csv-0.71.min

<https://code.google.com/p/jquery-csv/>

## Introduction:

Load CSV records to charts using Google Chart API. Create a JavaScript file in WebContent, load chart data to browser via Xpages. Build a connection to a Lotus Notes Domino Database located on C Drive to get started...

## Disclaimer:

Information contained in the following is presented as is. This tutorial assumes you have basic programming knowledge. All tutorials are based on an Eclipse/Eclipse-based software. Should you need to familiarize yourself with a certain Eclipse environment, prior to continuing this tutorial, please stop now and see our Tutorials page...

## Save JavaScript to WebContent

At this point we assume Domino Designer 8.5.3, Google Chart API, and the jQuery CSV plugin are available/installed in your environment. Create an Xpages form, plug a reference to the JavaScript and load Google Charts to user.

## myjquerybarchartsregdata.js

```
// @author: Dököll Solutions, Inc.
// Program: myjquerybarchartsregdata.js
// Created from copy: 2013.08.25.8.57.PM
// load the visualization library from Google and set a listener
google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);

function drawChart() {
    // grab the CSV
    $.get("TutsDataForJSCharts.csv", function(csvString) {
        // transform the CSV string into a 2-dimensional array
```

```

        var arrayData = $.csv.toArrays(csvString, {onParseValue:
$.csv.hooks.castToScalar});

        // use arrayData to load the select elements with the appropriate
options
        for (var i = 0; i < arrayData[0].length; i++) {
            // this adds the given option to both select elements
            $("select").append("<option value='" + i + "'>" + arrayData[0][i] +
"</option>");
        }
        // set the default selection
        $("#domain option[value='0']").attr("selected", "selected");
        $("#range option[value='1']").attr("selected", "selected");

        // this new DataTable object holds all the data
        var data = new google.visualization.arrayToDataTable(arrayData);

        // this view can select a subset of the data at a time
        var view = new google.visualization.DataView(data);
        view.setColumns([0,1]);

        var options = {
            title: "dokollolutionsinc.com Programming Tutorial Code Samples By
Type",
            hAxis: {title: data.getColumnLabel(0), minValue:
data.getColumnRange(0).min, maxValue: data.getColumnRange(0).max},
            vAxis: {title: data.getColumnLabel(1), minValue:
data.getColumnRange(1).min, maxValue: data.getColumnRange(1).max},
            legend: 'none'
        };

        var chart = new
google.visualization.BarChart(document.getElementById('chart'));
        chart.draw(view, options);

        // set listener for the update button
        $("select").change(function(){
            // determine selected domain and range
            var domain = +$("#domain option:selected").val();
            var range = +$("#range option:selected").val();

            // update the view
            view.setColumns([domain,range]);

            // update the options
            options.hAxis.title = data.getColumnLabel(domain);
            options.hAxis.minValue = data.getColumnRange(domain).min;
            options.hAxis.maxValue = data.getColumnRange(domain).max;
            options.vAxis.title = data.getColumnLabel(range);
            options.vAxis.minValue = data.getColumnRange(range).min;
            options.vAxis.maxValue = data.getColumnRange(range).max;

            // update the chart
            chart.draw(view, options);
        });
    });
}

```

## Create CSV file and place in WebContent

You will now create a CSV file with the following column coordinates: CodeName, Quantity, Downloads, Rating, Response, Inventory. You want to fill these columns with data to be read into Google charts. Firstly, add the following, or your choosing, to the CodeName column/rows (1) CSS, (2) LSAgent (3) Java (4) JavaAgent (5) XML, lastly add integer values to Quantity, Downloads, Rating, Response, Inventory respectively to column/rows.

CodeName	Quantity	Downloads	Rating	Response	Inventory
CSS	10	5	5	3	5
LSAgent	10	5	5	7	5
Java	30	10	5	7	10
JavaAgent	40	10	5	5	30
XML	20	5	5	5	15

## Create Xpages form/Embed HTML Code:

Make available the Google Chart API code via HTML code within Xpages form and show chart to user. Be sure to place the CSV file to the WebContent folder at this point or your choosing; simply change the code to locate the file if adding to a server or subfolders.

# xpjquerybarchartsregdata.xsp;

```
<?xml version="1.0" encoding="UTF-8"?>
<xp:view xmlns:xp="http://www.ibm.com/xsp/core">

<html>
<head>
  <title>Dököll Solutions, Inc. Tutorials (Bar)</title>
  <script src="https://www.google.com/jsapi"></script>
  <script src="http://code.jquery.com/jquery-1.10.1.min.js"></script>
  <script src="jquery.csv-0.71.js"></script>
  <script src="/sitefindermobile.nsf/myjquerybarchartsregdata.js"></script>
</head>
<body>
<center>
<h1>Dököll Solutions, Inc. Tutorials</h1>
<h3>Tutorials By Type</h3>
<hr></hr>
  <div id="chart" style="width:800px; height:500px;">
  </div>
  <h3>Choose: Quantity, Downloads, Rating, Response, Inventory</h3>
  <hr></hr>
  <br></br>
  <select id="range">
  <option value="" >Select Quantity</option>
  </select>
  <select id="domain" disabled="disabled">
  <option value="" >Select CodeName</option>
  </select>
</center>
</body>
```

```
</html>
</xp:view>
```

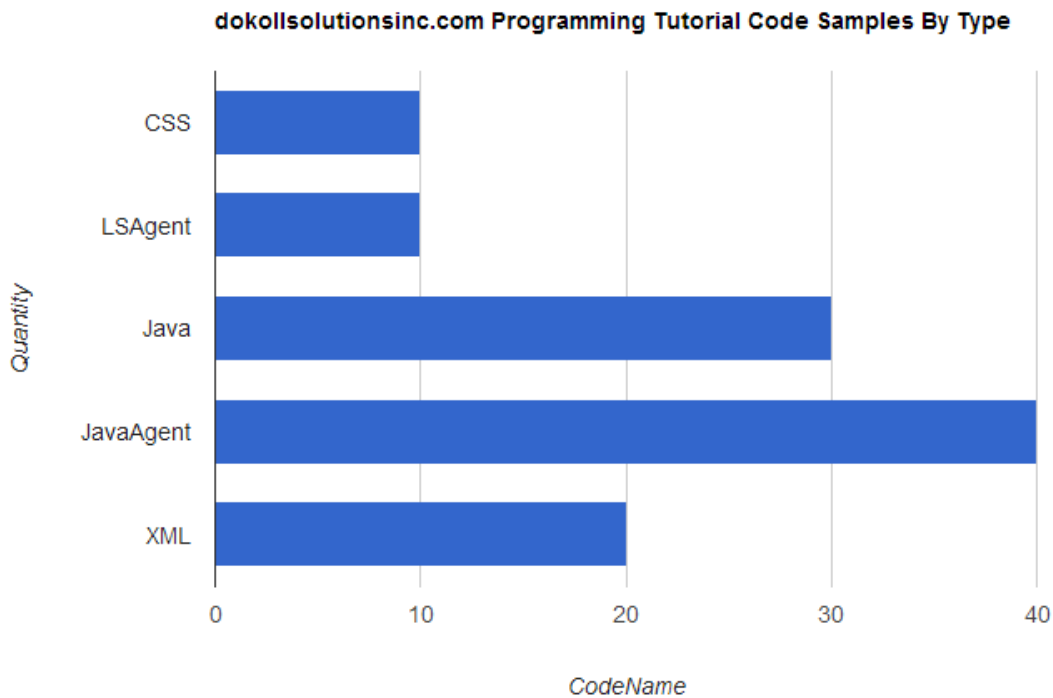
**Hover Charts to visualize data:**

See pop up references to the data included in the Google Charts by hovering on the respective Bar Chart column. If you are not see your chart, it may be because your CSV file cannot be found. Be sure to place the CSV file to the WebContent folder at this point or your choosing; simply change the code to locate the file if adding to a server or subfolders.

**Bar/Pie Chart Screenshot:**

**Dököll Solutions, Inc. Tutorials**  
Tutorials By Type

---



Choose: Quantity, Downloads, Rating, Response, Inventory

---

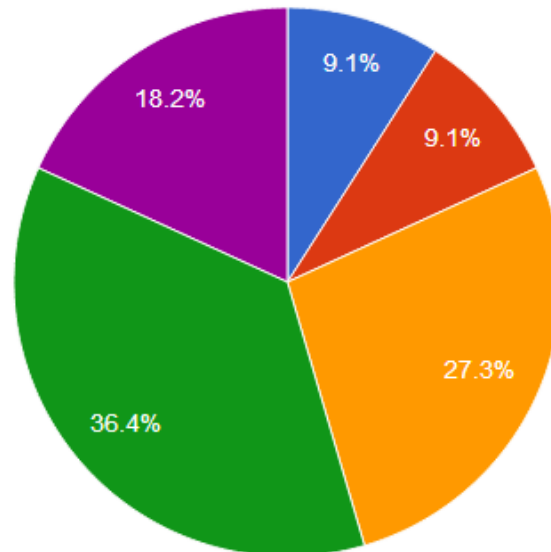
Quantity  CodeName

## Dököll Solutions, Inc. Tutorials

### Tutorials By Type

---

dokollsolutionsinc.com Programming Tutorial Code Samples By Type



---

Choose: Quantity, Downloads, Rating, Response, Inventory

### Conclusion:

While you can Repeat the above process to create a Pie Chart, using same CSV file, creating a scattered chart, as one example, will need additional programming on your part. Please visit the Google Chart API webpage for more info: <https://developers.google.com/chart/>

Questions, comments, please post a brief message on our [Contact](#) form on the main site.

Follow Us: [@DokollSolutions](#) | <https://twitter.com/DokollSolutions>

or Like Us: Dököll Solutions, Inc. | <https://www.facebook.com>

Thank you for coming...