Extending PI Vision

Agenda

- Day 1: Simple Value & Time Series symbols
- Day 2: Chart & Tool Pane symbols.
  Troubleshooting and Deployment.
- Day 3: Symbol of your choice

Prerequisites

Must Knows

Basic Javascript

Basic HTML

Basic CSS

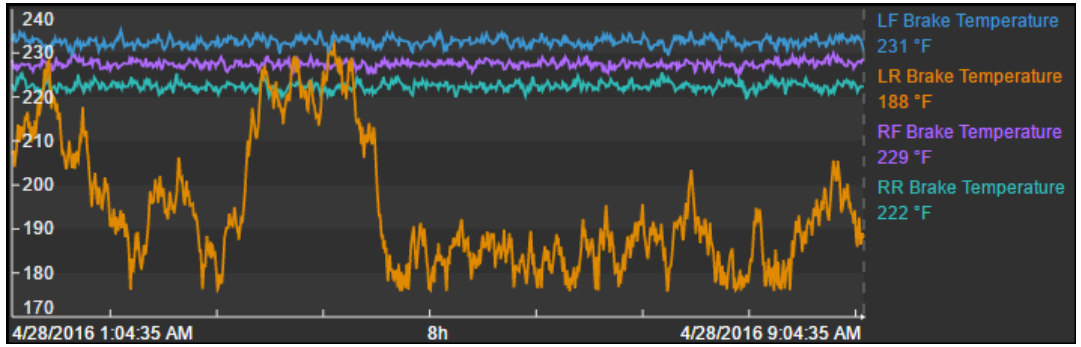
Nice to know

AngularJS

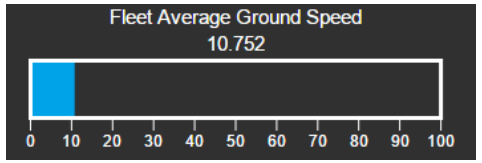
Git / GitHub

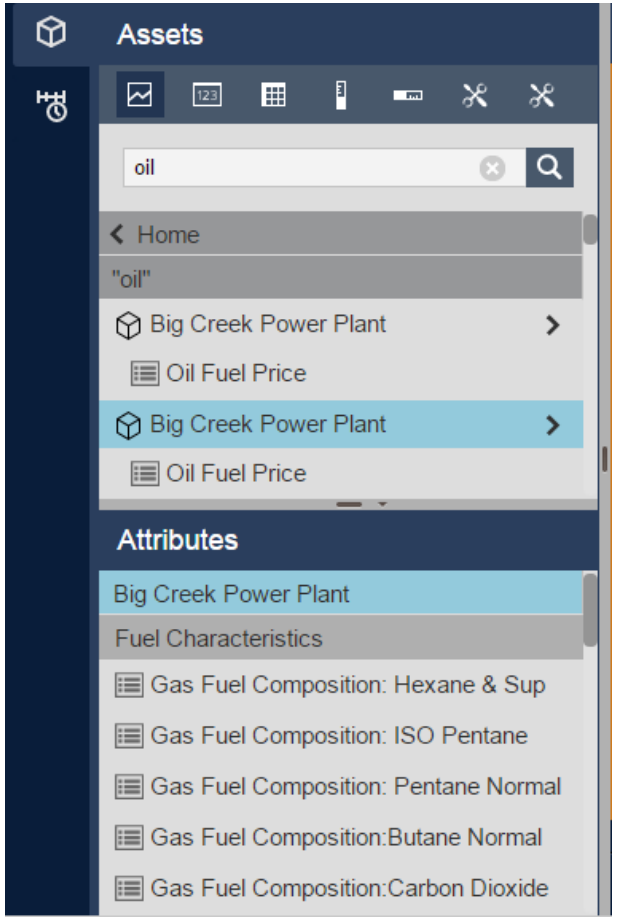
Introduction

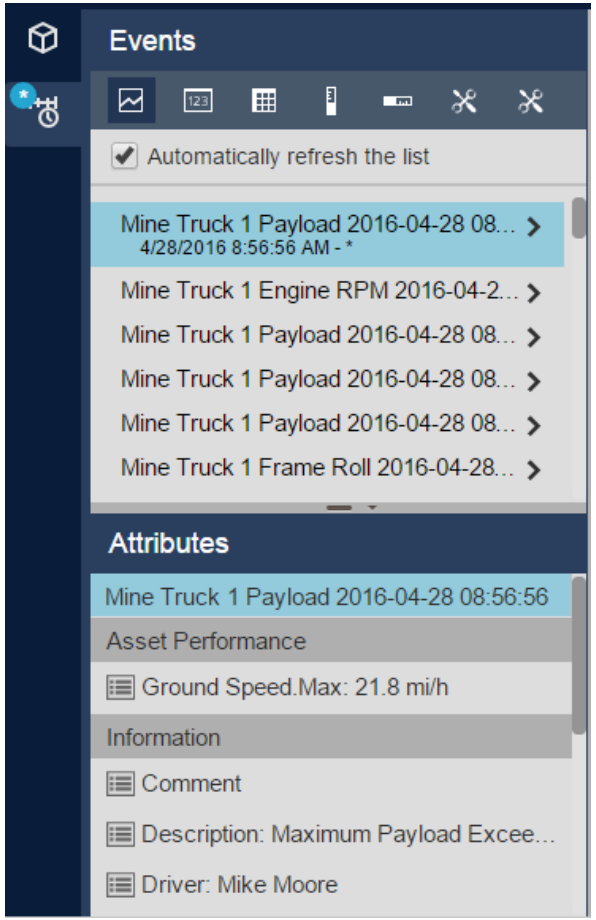
Extensibility examples



Name	Value	Trend	Average	Minimum	Maximum
Mine Truck 1|Left Front Accelerometer	6.10		6.92	2.26	11.23
Mine Truck 1|Left Rear Accelerometer	13.24		13.06	6.37	16.89
Mine Truck 1|Right Front Accelerometer	2.13		4.43	0.66	8.73
Mine Truck 1|Right Rear Accelerometer	7.84		9.70	2.64	17.84







Yes, native PI Vision symbols use the same framework as custom symbols

Examples from OSIsoft Demo Framework

Visit OSIsoft's GitHub at https://github.com/osisoft/PI-Coresight-Custom-Symbols/tree/master/Community%20Samples/OSIsoft

Winners of the Visualization Hackathon

Map

By Servelec Group

https://pisquare.osisoft.com/people/gemslie/blog/2017/03/10/hackathon-servelec-group-team-1

Scatterplot Matrix

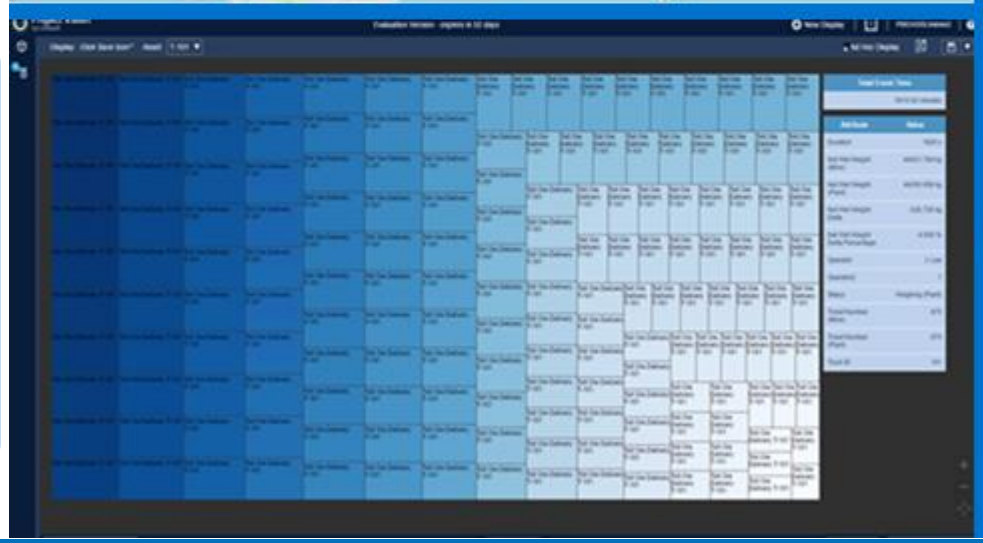
By Magion Industrial Software Solutions

https://pisquare.osisoft.com/people/DGolverdingen/blog/2017/03/07/pi-visualization-hackathon-magion-entry

Event Map

By Exele

https://pisquare.osisoft.com/people/DanFishman/blog/2017/03/12/exele-event-map-osisoft-visualization-virtual-hackathon-2017-entry



KPI Widgets

By Servelec Group

https://pisquare.osisoft.com/people/tim.maclaren/blog/2017/03/12/hackathon-servelec-group-team-2

Who benefits from extensions?

OSIsoft Teams

Partner EcoSphere

Partners

Customers

Getting started

Technologies

Microsoft

ASP.NET

PI Vision symbols are only made up of these

Workspace

PISRV01

PI Vision
PI Data Archive
PI AF Server
MS SQL Server

C:\Program Files\PIPC\PIVision\Scripts\app\editor\symbols

\ext ← Required. Where symbol files go

\ext\libraries ← Optional. Where 3rd party libraries and image files go

\ext\images

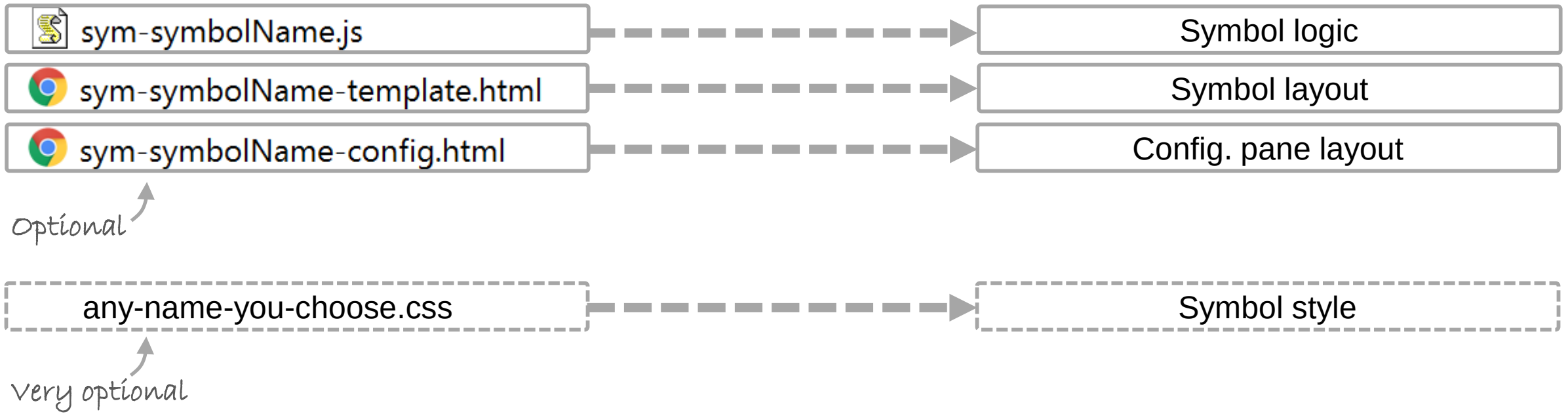
\ext\…

PICLIENT01

For testing and troubleshooting purposes only

Essential Symbol Files

C:\Program Files\PIPC\PIVision\Scripts\app\editor\symbols\ext



Symbol Template

```
(function (PV) {
  "use strict";

  function symbolVis() { };
  PV.deriveVisualizationFromBase(symbolVis);

  var definition = {
     typeName: "",
     visObjectType: symbolVis,
     datasourceBehavior: PV.Extensibility.Enums.DatasourceBehaviors.Single,
     getDefaultConfig: function(){
         return {
                  Height: 150,
                  Width: 150
         }
     }
  }

  symbolVis.prototype.init = function(scope, elem) { };

  PV.symbolCatalog.register(definition);
})(window.PIVisualization);
```

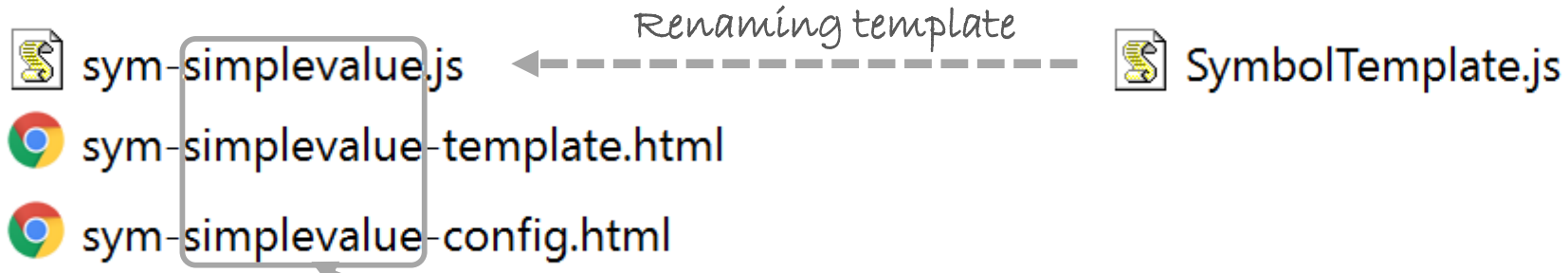
Minimum code to have the symbol registered in PI Vision.

Results in a blank symbol.

SymbolTemplate.js

Creating a symbol

Symbol definition



typeName should agree with the file name

```
...
var definition = {
    typeName: "simplevalue",
    ...

}
...
```

sym-simplevalue.js

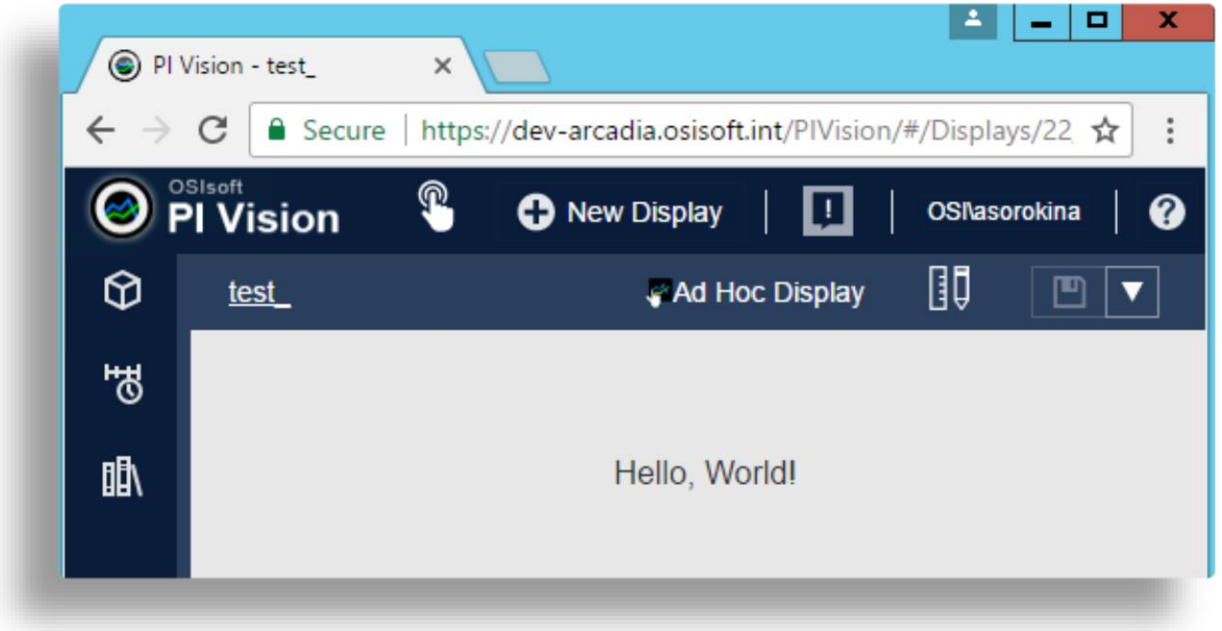
AngularJS Expressions

Allow you to insert dynamic values into your HTML

String operations

```
{{"Hello" + " ,World!"}}
```

sym-simplevalue-template.html



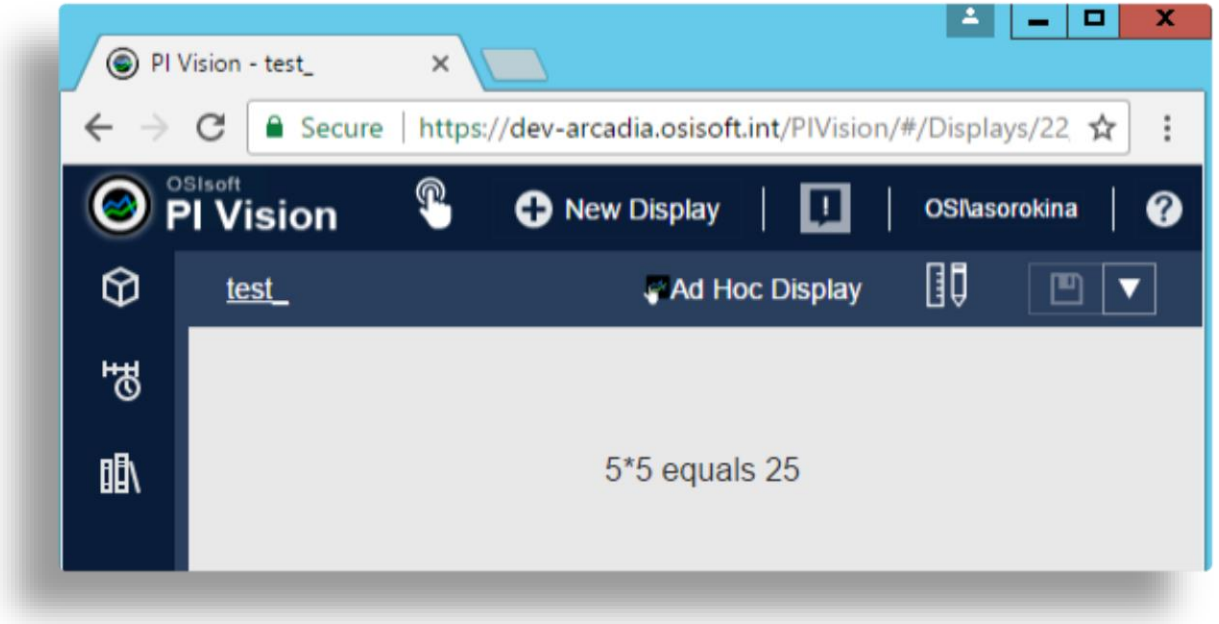
AngularJS Expressions

Allow you to insert dynamic values into your HTML

Numeric operations

```
5*5 equals {{5*5}}
```

sym-simplevalue-template.html



More operations:
https://docs.angularjs.org/guide/expression

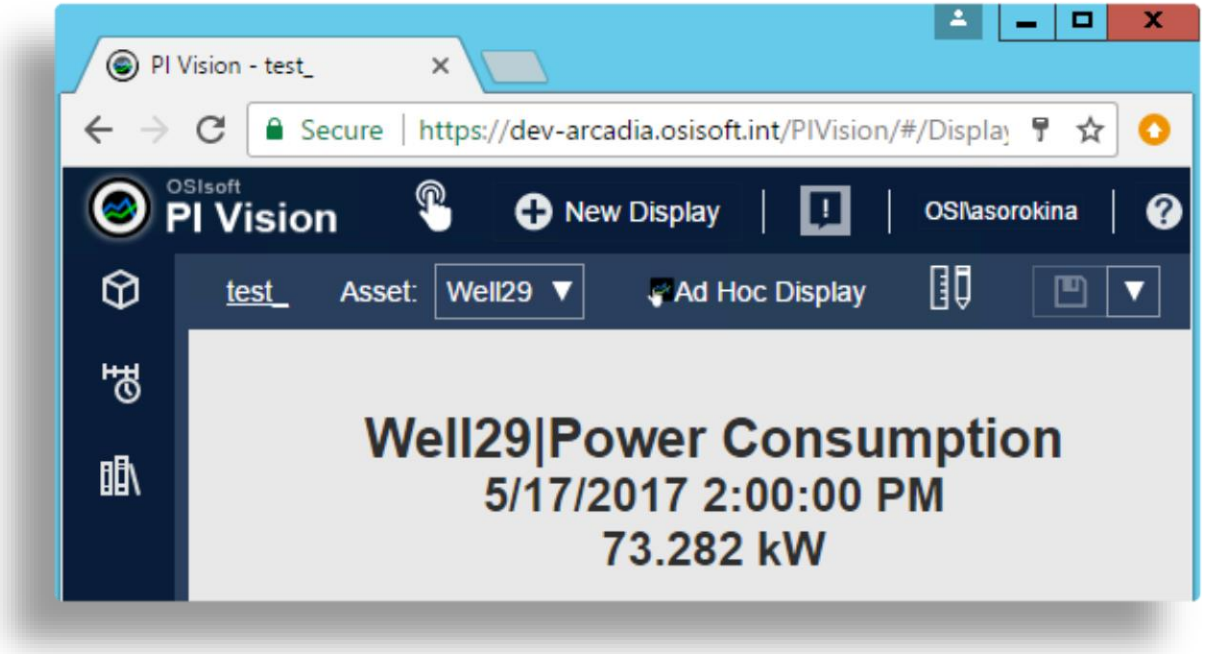
Challenges

Challenge 1 – Hello, World

1. Create files for a custom symbol named **basicvalue**

   In newly created **sym-basicvalue.js** add code from the SymbolTemplate.js file
2. In **sym-basicvalue-template.html**, create a simple Expression to display “Hello, <*your Name*>!”
3. In **sym-basicvalue-template.html**, create a simple Expression to display the result of (6*7)

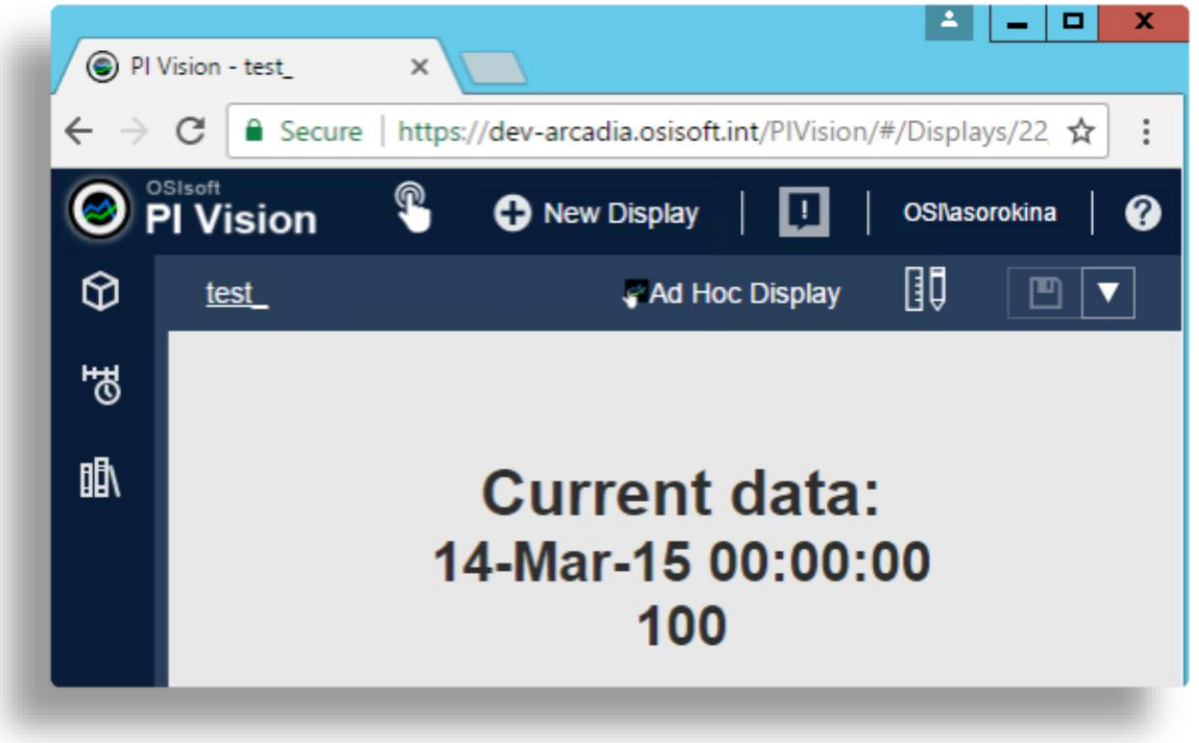
A Simple Value symbol



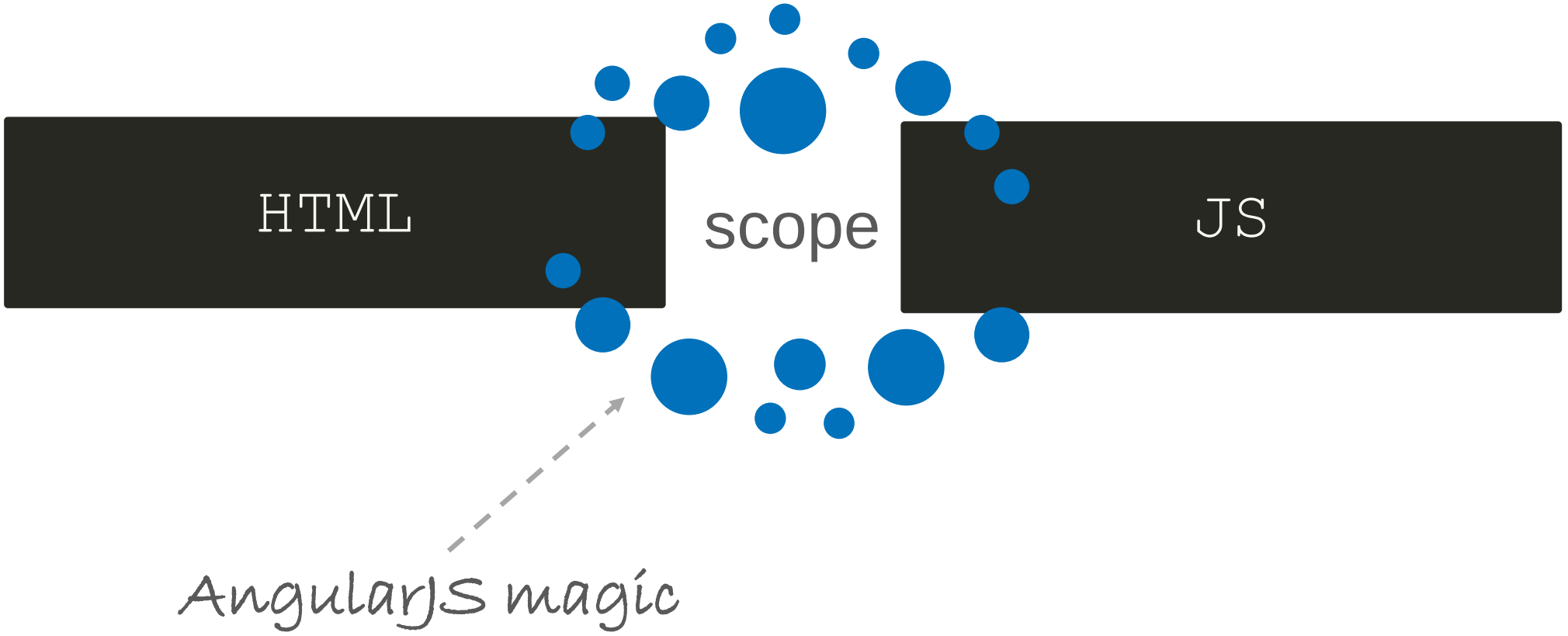
Working with mockup data

```
var dataItem = {
        Time: "14-Mar-15 00:00:00",
        Value: 100
}
```

Let's display this simple value in our custom symbol



Scope



Adding data to the Scope

Notice that our scope here doesn't have a "$" in front of it like AngularJS' $scope.

This is because PI Vision scope is in fact a wrapper around AngularJS scope.

Next stop: displaying this data on the custom symbol!

```
(function (PV) {
...
symbolVis.prototype.init = function(scope, elem) {
        scope.Time = dataItem.Time;
        scope.Value = dataItem.Value;
}
var dataItem = {
        Time: "14-Mar-15 00:00:00",
        Value: 100
}
...
})(window.PIVisualization);
```

sym-simplevalue.js

Our current -template.html

```
<div>
    <h1> Current data: </h1>
    <h2> Time  </h2>
    <h2> Value </h2>

</div>
```

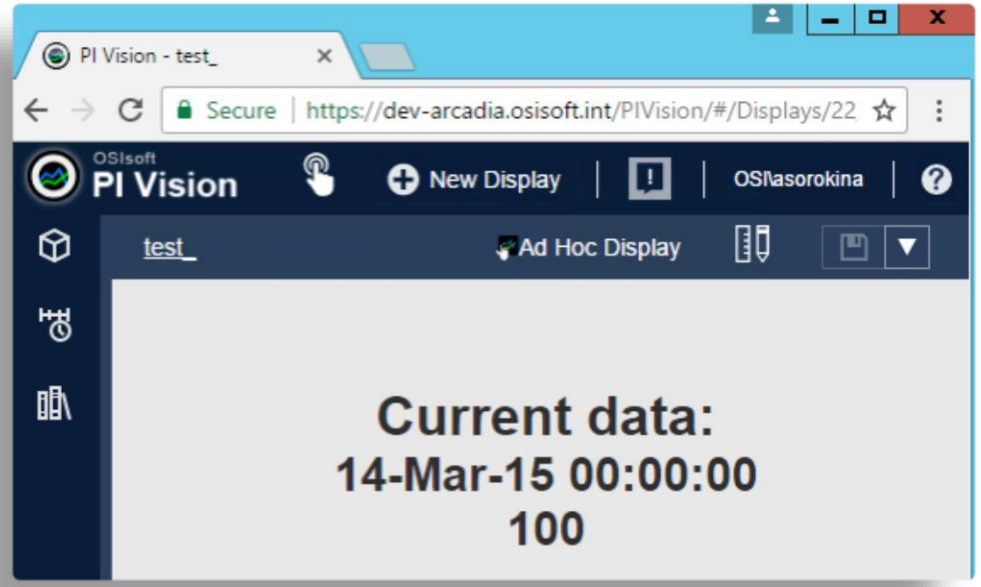
Data values go here

sym-simplevalue-template.html

Displaying our first data

```
<div>
    <h1> Current data: </h1>
    <h2> {{Time}} </h2>
    <h2> {{Value}} </h2>

</div>
```

sym-simplevalue-template.html




```
(function (PV) {
...
symbolVis.prototype.init = function(scope, elem) {
        scope.Time = dataItem.Time;
        scope.Value = dataItem.Value;
}
var dataItem = {
        Time: "14-Mar-15 00:00:00",
        Value: 100
}
...
})(window.PIVisualization);
```

sym-simplevalue.js

Challenges

Challenge 2 – Display a value

1. In **sym-basicvalue.js**, add a data object dataItem with Timestamp and Value properties.
2. Assign this data object to the Time and Value properties of scope.
3. Display the Timestamp of the data item inside <h3> tag.
4. Display the Value of the data item inside <h3> tag

Real-time data in PI Vision extensibility

Is automagically provided through the Data Update function.

```
(function (PV) {
...
symbolVis.prototype.init = function(scope, elem) {

        this.onDataUpdate = dataUpdate;
        function dataUpdate(data) {

                console.log(data);

        }
}
...
})(window.PIVisualization);
```

sym-simplevalue.js

Triggered every 5 seconds by default

But what do the data updates look like?

console.log(data) output for AF Attribute

Full data update with metadata (sporadic)

```
Object {…}
 Label:"Well29|Power Consumption"
 Path:"af:\\AFSERVER\Oil and Gas Well Downtime Tracking\West Texas\Clear Fork\Well29|Power
 Consumption"
 SymbolName:"Symbol0"
 Time:"5/10/2017 11:20:00 PM"
 Units:"kW"
 Value:"89.294"
```

console in browser (F12)

Data value update (typical)

```
Object {…}
 SymbolName:"Symbol0"
 Time:"5/10/2017 11:20:00 PM"
 Value:"89.294"
```

console in browser (F12)

console.log(data) output for PI Point

Full data update with metadata (sporadic)

```
Object {…}
 Label:"SINUSOID"
 Path:"pi:\\STARK\SINUSOID"
 SymbolName:"Symbol0"
 Time:"5/10/2017 11:20:00 PM"
 Value:"69.304"
```

console in browser (F12)

Data value update (typical)

```
Object {…}
 SymbolName:"Symbol0"
 Time:"5/10/2017 11:20:00 PM"
 Value:"69.304"
```

console in browser (F12)

Default Data Shape: Value

Full data update with metadata (sporadic)

```
Object {…}
  Label
  Path
  SymbolName
  Time
  Units
  Value
```

Data value update (typical)

```
Object {…}
  SymbolName
  Time
  Value
```

```
...

    var definition = {
        ...
        getDefaultConfig: function(){
            return {
                DataShape: "Value",
                Height: 150,
                Width: 150
            }
        }
    }
...
```

sym-simplevalue.js

More Shapes: [PI Vision Extensibility Documentation]

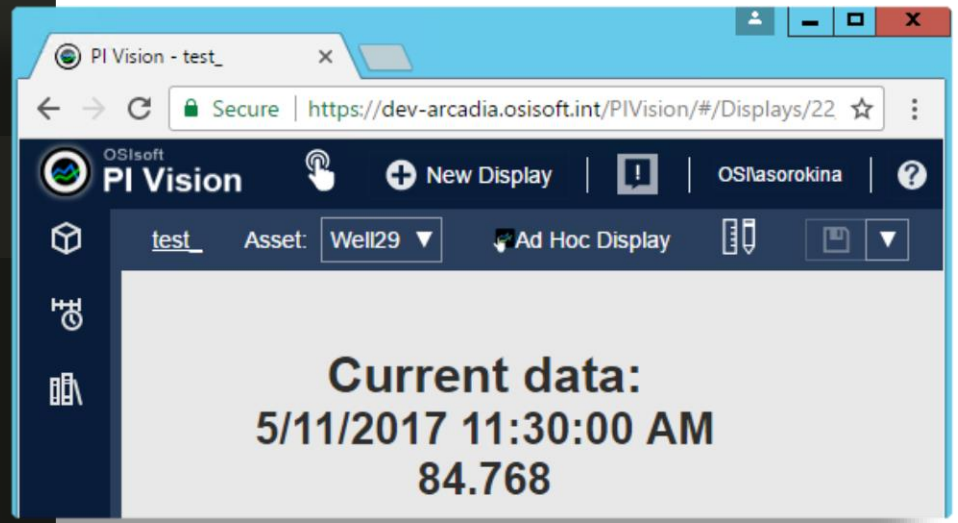
Real-time data in PI Vision extensibility

```
...
symbolVis.prototype.init = function(scope, elem) {

        this.onDataUpdate = dataUpdate;
        function dataUpdate(data) {
                scope.Time = data.Time;
                scope.Value = data.Value;




        }
}
...
```

sym-simplevalue.js

```
<div>
    <h1> Current data: </h1>
    <h2> {{Time}}  </h2>
    <h2> {{Value}} </h2>
</div>
```

sym-simplevalue-template.html



DataItem: *Object {…}*
- SymbolName
- Time
- Value
- …

Real-time data in PI Vision extensibility

```
...
symbolVis.prototype.init = function(scope, elem) {

        this.onDataUpdate = dataUpdate;
        function dataUpdate(data) {
                scope.Time = data.Time;
                scope.Value = data.Value;
                if(data.Label){
                    scope.Label = data.Label;
                    scope.UOM = data.Units;
                }
        }
}
...
```

sym-simplevalue.js

```
<div>
    <h1> {{Label}} </h1>
    <h2> {{Time}}  </h2>
    <h2> {{Value}} {{UOM}} </h2>
</div>
```

sym-simplevalue-template.html

DataItem: *Object {…}*
- SymbolName
- Time
- Value
- Units
- Label
- …

What we've learned so far

- Some custom symbols in the wild
- Core Symbol Files – follow a certain pattern
- Symbol Definition – typeName corresponds to file names
- AngularJS Expressions – binds data to view on page
- Symbol Data Shape – by default, 'Value'

Challenges

Challenge 3 – Display a PI Value

1. In **sym-basicvalue.js**, add a data update function and name it **dataUpdate**. Declare one parameter: **newdata**.
2. In the custom symbol, display the **newdata**'s **Label**, **Time**, **Value** and **Units** properties.

Troubleshooting

Debugging

1. Enable Debugging mode
   In **web.config** set
   <compilation debug="true" targetFramework="4.6"/>
2. Use Browser's Debugging Tools (F12)

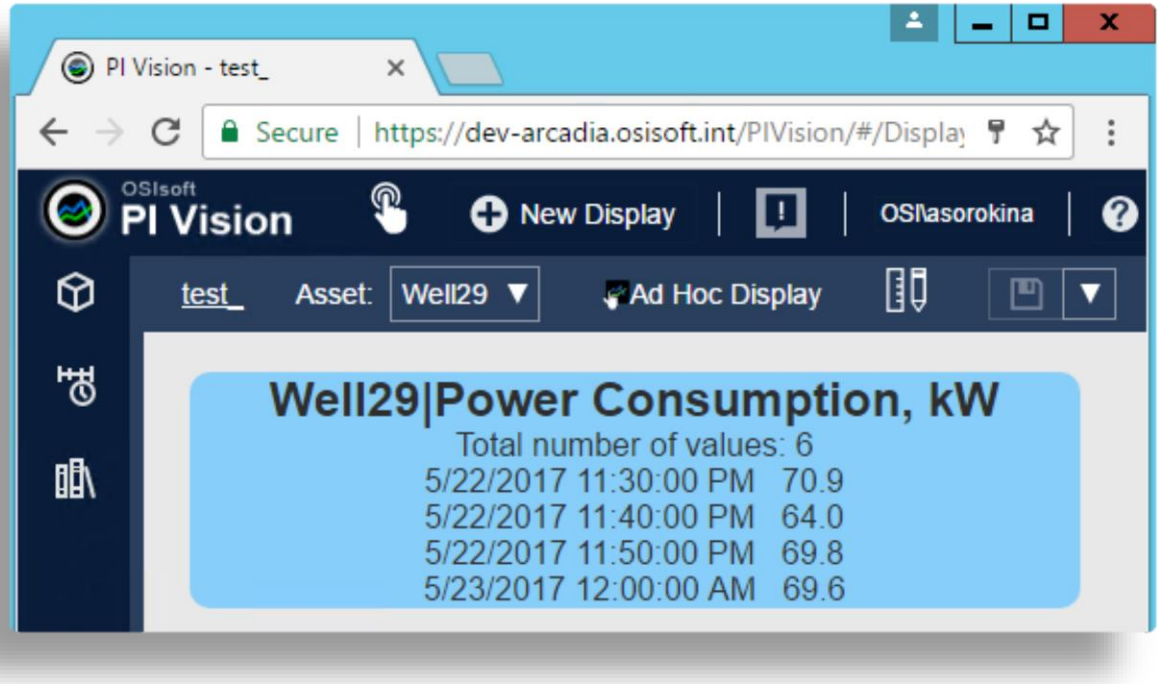
Chrome DevTools Overview:
https://developers.google.com/web/tools/chrome-devtools/

Debugging

3. …

Displaying historical data

A Timeseries symbol



Symbol definition


sym-timeseries.js


sym-timeseries-config.html


sym-timeseries-template.html

```
...
var definition = {
    typeName: "timeseries",
    ...

}
...
```

sym-timeseries.js

Mock data: Arrays

```
var dataItems = [ <---------------- Now we have an array
        {
         Time: "14-Mar-15 00:00:00",
         Value: 100
        },
        {
         Time: "4-May-15 00:00:00",
         Value: 50
        }
]
```

Let's display this simple array on the custom symbol

Adding data to the Scope

Next step: showing this data array on the display!

```
(function (PV) {
...
symbolVis.prototype.init = function(scope, elem) {
        scope.Values = dataItems;
}
var dataItems = [
        {
         Time: "14-Mar-15 00:00:00",
         Value: 100
        },
        {
         Time: "4-May-15 00:00:00",
         Value: 50
        }
]
...
})(window.PIVisualization);
```

sym-timeseries.js

Displaying array elements

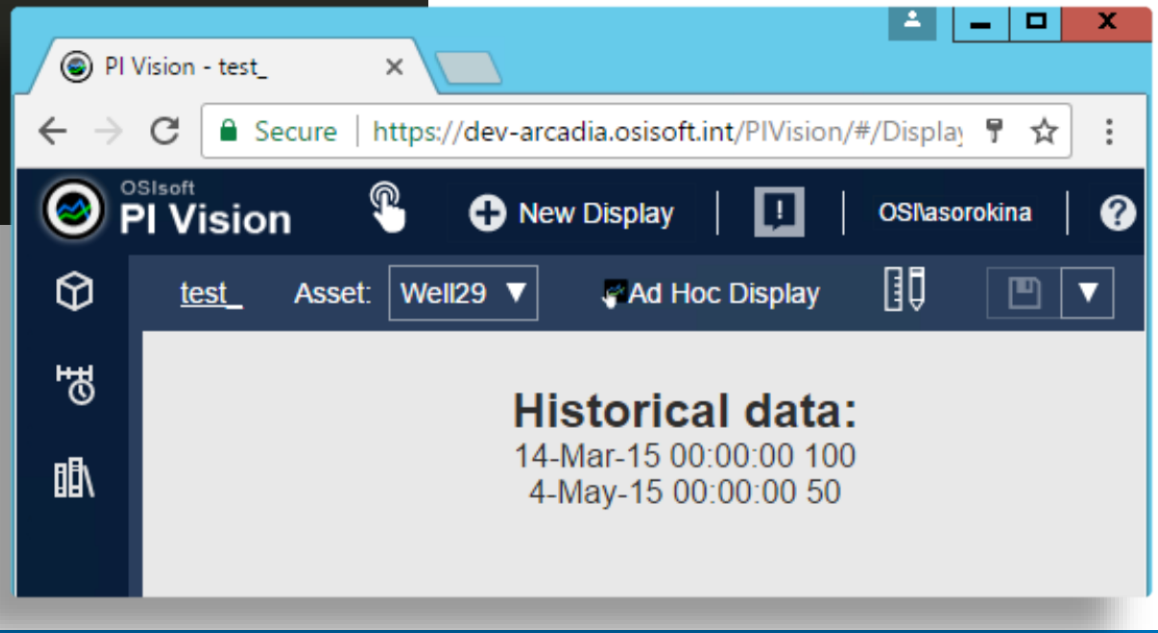
```
<div>

    <h1> Historical data: </h1>
    <div> {{Values[0].Time}} {{Values[0].Value}} </div>




</div>
```

sym-timeseries-template.html

Displaying array elements

```
<div>

    <h1> Historical data: </h1>
    <div> {{Values[0].Time}} {{Values[0].Value}} </div>
    <div> {{Values[1].Time}} {{Values[1].Value}} </div>




</div>
```

sym-timeseries-template.html



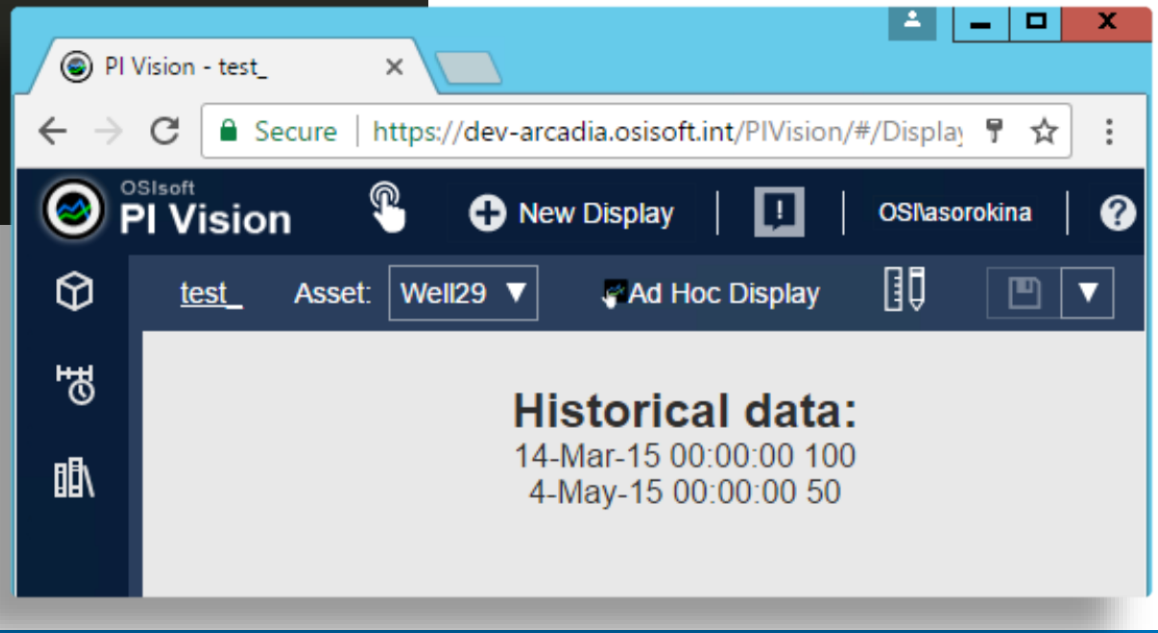
That's great and all, but what if we have 100 values?

NgRepeat Directive

```
<div>

    <h1> Historical data: </h1>
    <div ng-repeat="item in Values">
        {{item.Time}} {{item.Value}}
    </div>



</div>
```

sym-timeseries-template.html

Repeats this part for each item in the values array



Challenges

Challenge 4 – Display list of data

1. Create files for a custom symbol named **archivedata**
2. In **sym-archivedata.js**, add a data array dataItems with two objects with Time and Value properties. Yesterday at 6 AM, today at 9 AM.
3. Assign this data array to the DataItems property of scope.
4. Display all the data items of the array in a tag.

New Data Shape: Timeseries

Full data update with metadata (sporadic)

```
Object {…}
Data, array of:
    Label
    Path
    Units
    Values, array of:
        Time
        Value
    StartTime
    EndTime
SymbolName
```

Data value update (typical)

```
Object {…}
Data, array of:
    Values, array of:
        Time
        Value
    StartTime
    EndTime
SymbolName
```

```
...

    var definition = {
        ...
        getDefaultConfig: function(){
            return {
                DataShape: "Timeseries",
                Height: 150,
                Width: 150
            }
        }
    }
...
```

sym-simplevalue.js

DataType and Description fields are available on extra config tweak (see documentation)

More Shapes: PI Vision Extensibility Documentation

console.log(data) output for AF Attribute

Full data update with metadata (sporadic)

```
Object {…}
Data:Array(1)
       0:Object
           EndTime:"5/20/2017 4:32:59 PM"
           Label:"Well29|Power Consumption"
           Path:"\\ARCADIA\OSIDemo Oil and Gas Well Downtime Tracking\West Texas\Clear Fork\Well29|Power
           Consumption"
           StartTime:"5/20/2017 8:32:59 AM"
           Units:"kW"
           Values:Array(3)
               0:Object
                   Time:"5/20/2017 4:10:00 PM"
                   Value:"72.686"
               1:Object
                   Time:"5/20/2017 4:20:00 PM"
                   Value:"76.036"
               ...
               length:3
       length:1
SymbolName:"Symbol0"
```

Time range of the display

console in browser (F12)

console.log(data) output for AF Attribute

Data value update (typical)

```
Object {…}
Data:Array(1)
       0:Object
           EndTime:"5/20/2017 4:32:59 PM"
           StartTime:"5/20/2017 8:32:59 AM"
           Values:Array(3)
               0:Object
                   Time:"5/20/2017 4:10:00 PM"
                   Value:"72.686"
               1:Object
                   Time:"5/20/2017 4:20:00 PM"
                   Value:"76.036"
               ...
               length:3
       length:1
SymbolName:"Symbol0"
```

console in browser (F12)

console.log(data) output for PI Point

Full data update with metadata (sporadic)

```
Object {…}
Data:Array(1)
    0:Object
        EndTime:"5/20/2017 4:32:59 PM"
        StartTime:"5/20/2017 8:32:59 AM"
        Label:"SINUSOID"
        Path:"\\STARK\SINUSOID"
        Values:Array(4)
            0:Object
                Time:"5/20/2017 4:10:00 PM"
                Value:"72.686"
            1:Object
                Time:"5/20/2017 4:20:00 PM"
                Value:"76.036“
            ...
    length:1
SymbolName:"Symbol0"
```

console in browser (F12)

console.log(data) output for PI Point

Data value update (typical)

```
Object {…}
Data:Array(1)
    0:Object
        EndTime:"5/20/2017 4:32:59 PM"
        StartTime:"5/20/2017 8:32:59 AM"
        Values:Array(4)
            0:Object
                Time:"5/20/2017 4:10:00 PM"
                Value:"72.686"
            1:Object
                Time:"5/20/2017 4:20:00 PM"
                Value:"76.036“
            ...
    length:1
SymbolName:"Symbol0"
```

console in browser (F12)

Multiple data items

```
...
var definition = {
    ...
    datasourceBehavior:
      PV.Extensibility.Enums.DatasourceBehaviors.Multiple,
    getDefaultConfig: function(){
        return {
                DataShape: "Timeseries",
                Height: 150,
                Width: 150
        }
    }
}
...
```

sym-simplevalue.js

```
Object {…}
Data:Array(1)
    0:Object
        ...
        Label:"SINUSOID"
        Values: ...
        ...
    1:Object
        ...
        Label:"Well29|Line Pressure"
        Values: ...
        ...
    length:1
SymbolName:"Symbol0"
```

console in browser (F12)

Real-time data in PI Vision extensibility

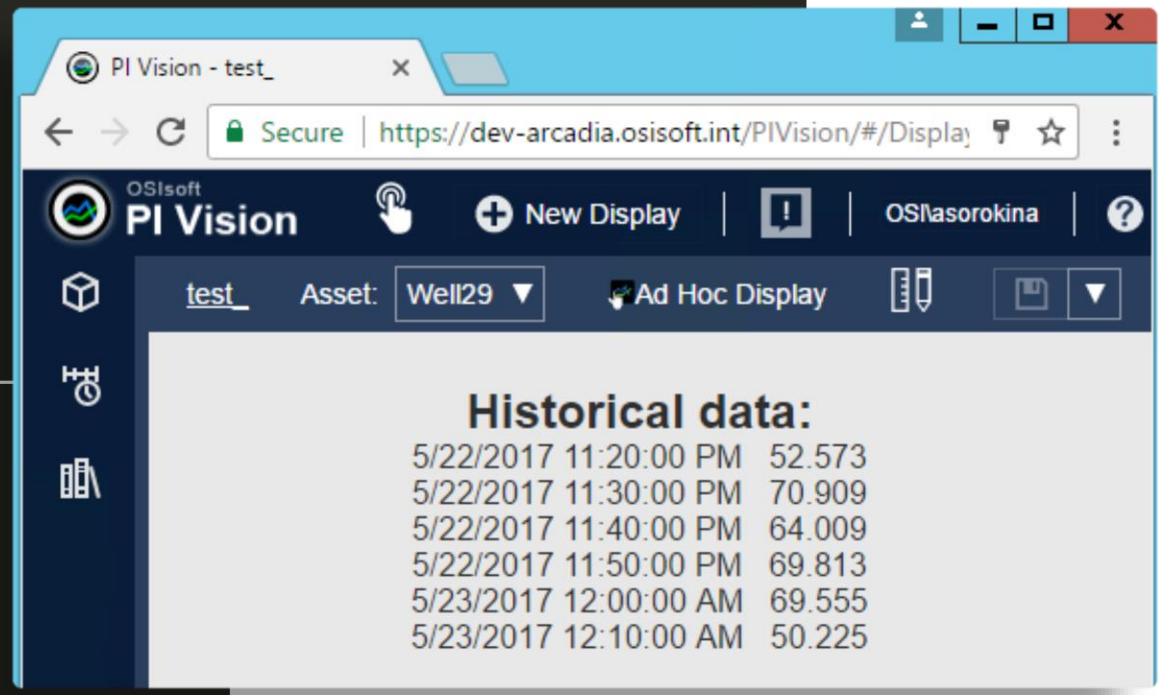
```
symbolVis.prototype.init = function(scope, elem) {
        this.onDataUpdate = dataUpdate;
        function dataUpdate(data) {
                if(!data)return;
                var firstAttribute = data.Data[0];
                scope.Values = firstAttribute.Values;




        }
}
...
```

sym-timeseries.js

```
<div>
    <h1> Historical data: </h1>
    <div ng-repeat="item in Values">
        {{item.Time}}   {{item.Value}}
    </div>
</div>
```

Non-Breaking SPace

sym-timeseries-template.html



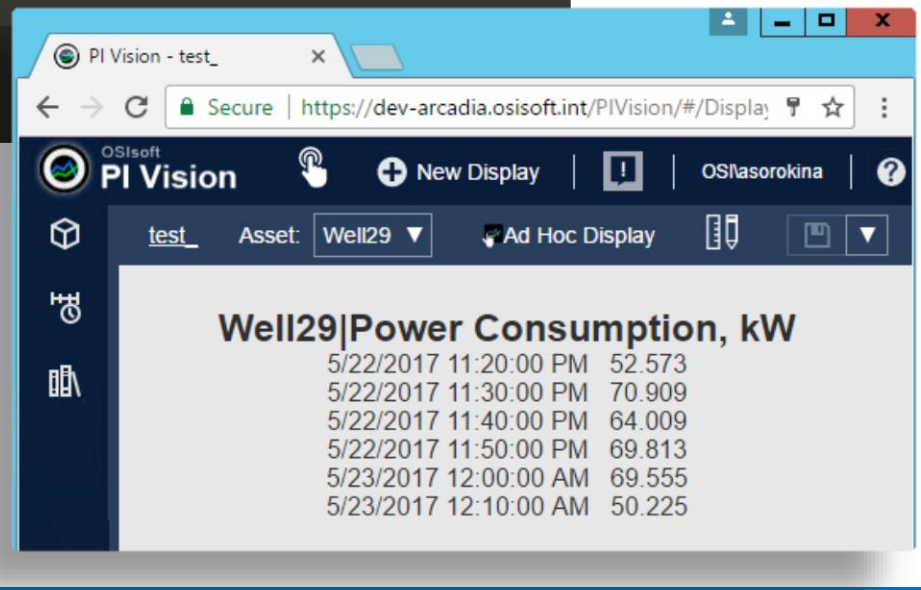
Real-time data in PI Vision extensibility

```
symbolVis.prototype.init = function(scope, elem) {
	this.onDataUpdate = dataUpdate;
	function dataUpdate(data) {
		if(!data)return;
		var firstAttribute = data.Data[0];
		scope.Values = firstAttribute.Values;
		if(firstAttribute.Label){
			scope.Label = firstAttribute.Label;
			scope.UOM = firstAttribute.Units;
		}
	}
}
```

sym-timeseries.js

```
<div>
    <h3> {{Label}}, {{UOM}} </h3>
    <div ng-repeat="item in Values">
        {{item.Time}}   {{item.Value}}
    </div>
</div>
```

sym-timeseries-template.html



Challenges

Challenge 5 – Display a list of PI Data

1. In sym-archivedata.js, add a data update function and name it dataUpdate, pass it newdata as a parameter
2. Change DataShape of the symbol to handle time series data
3. Display Label and UOM parameters inside <h3> tags
4. Display all the historical data (timestamps and values) inside tags.

NgShow Directive

Shows or hides element based on the expression provided in the ngShow attribute

```
<div>
        <h3> {{Label}}, {{UOM}} </h3>
        <div ng-repeat="item in Values" ng-show="{{item.Value > 60}}" >
                {{item.Time}}   {{item.Value}}
        </div>
</div>
```

sym-timeseries-template.html

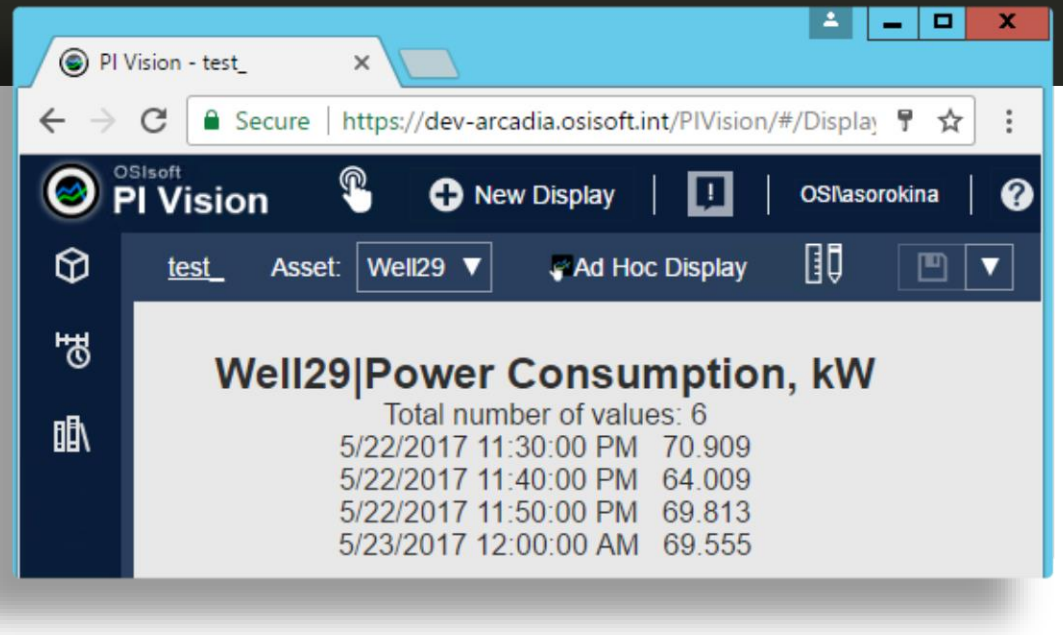
Will only show the element if the value of the Expression evaluates to true.

Check out the NgHide directive: https://docs.angularjs.org/api/ng/directive/ngHide

NgShow Directive

```
<div>
        <h3> {{Label}}, {{UOM}} </h3>
        <div>Total number of values: {{Values.length}}</div>
        <div ng-repeat="item in Values" ng-show="{{item.Value > 60}}">
                {{item.Time}}   {{item.Value}}
        </div>
</div>
```

sym-timeseries-template.html

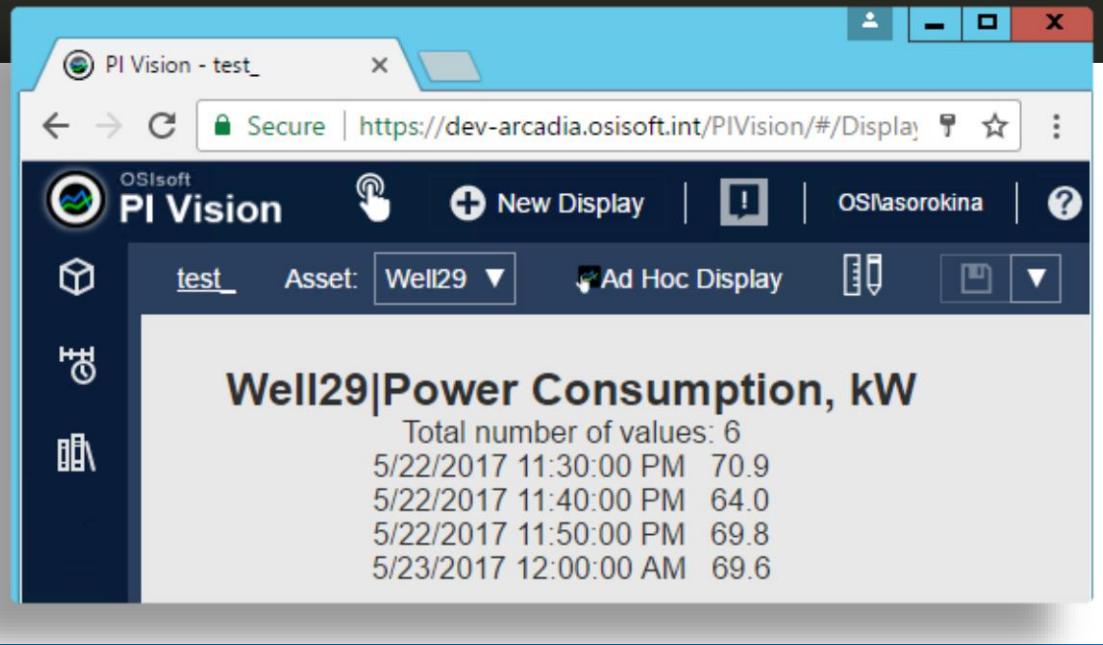


Displays the total number of values

Filters

```
<div>
        <h3> {{Label}}, {{UOM}} </h3>
        <div>Total number of values: {{Values.length}}</div>
        <div ng-repeat="item in Values" ng-show="{{item.Value > 60}}">
                {{item.Time}}   {{item.Value | number: 1}}
        </div>
</div>
```

sym-timeseries-template.html



Round the value to one decimal space

Pipe – "send the output to"

More Filters…

date

```
{{'2017-05-24T06:37:45Z' | date:'medium'}}
```

May 24, 2017 1:37:45 AM

```
{{'2017-05-24T06:37:45Z' | date:"MM/dd/yyyy 'at' h:mma"}}
```

05/24/2017 at 1:37AM

currency

```
{{3.50 | currency}}
```

$3.50

Defaults to the locale format

lowercase, uppercase

```
{{'A Møøse once bit my sister...' | uppercase}}
```

A MØØSE ONCE BIT MY SISTER...

limitTo

```
<div ng-repeat="item in Values | limitTo: 3">
```

Will print the first 3 values

```
{{'String is an array of characters' | limitTo: 9}}
```

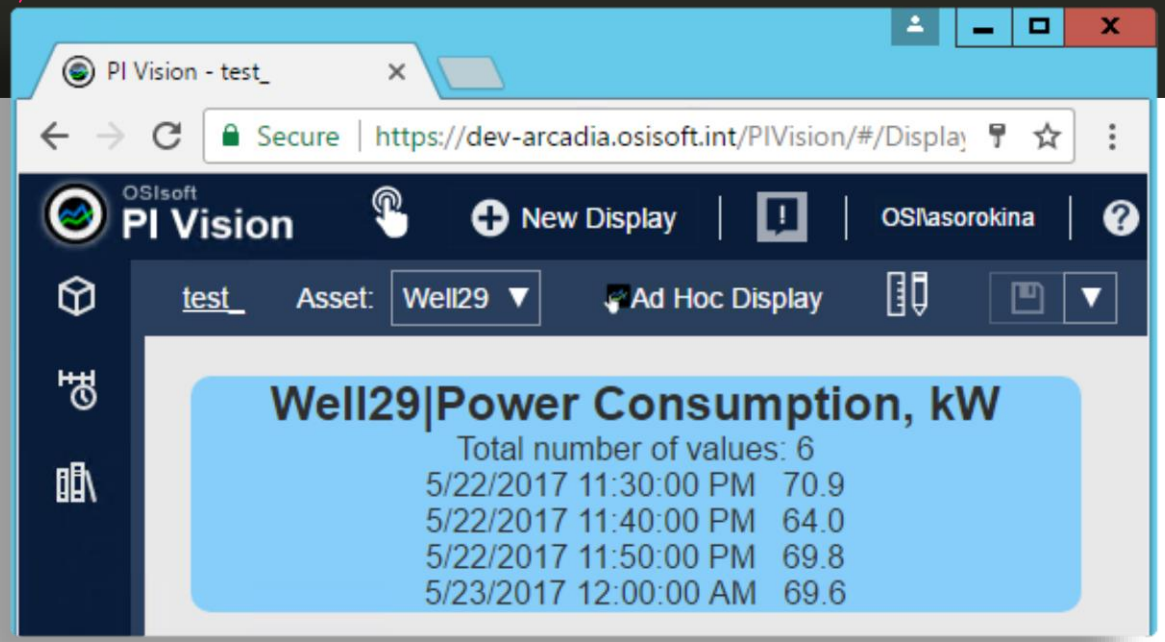
String is

Even more filters (orderBy, json, filter, etc.): https://docs.angularjs.org/api/ng/filter

Prettifying with CSS

```
<div style="background-color: lightskyblue; border-radius: 10px">
      <h3> {{Label}}, {{UOM}} </h3>
      <div>Total number of values: {{Values.length}}</div>
      <div ng-repeat="item in Values" ng-show="{{item.Value > 60}}">
            {{item.Time}}   {{item.Value | number: 1}}
      </div>
</div>
```

sym-timeseries-template.html

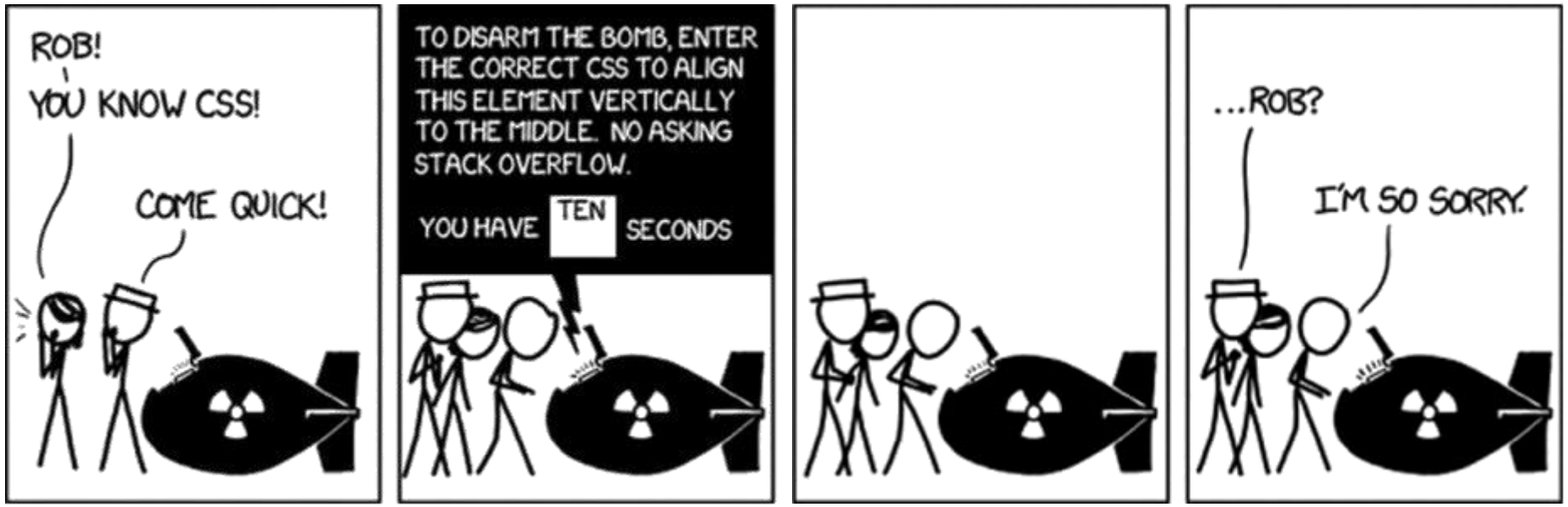


Rounds the boarders

Sets background color

More CSS properties:
https://developer.mozilla.org/en-US/docs/Web/CSS/Reference

Working with CSS…



Every. Single. Time.

https://m.9gag.com/gag/aYeXDyN

Original: https://xkcd.com/1168

Challenges

Challenge 6 – Filters and CSS

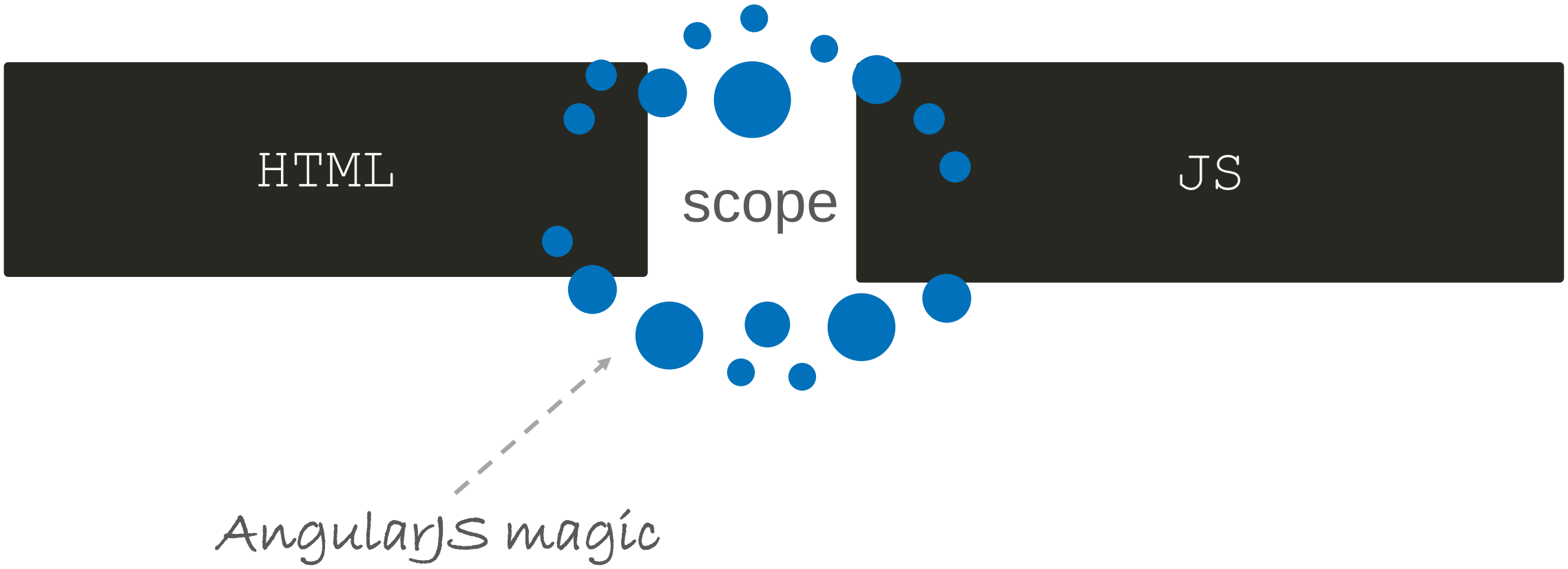
1. Show only values that exceed 70.
2. Display the total number of values.
3. Display each value rounded to two decimal places.
4. Apply some style to your custom symbol
   - background color of your choice
   - rounded borders

Configuration: Setting Defaults

```
...
  var definition = {
    ...
    getDefaultConfig: function(){
      return {
          DataShape: "Timeseries",
          Height: 150,
          Width: 150,
          BackgroundColor: "lightskyblue",
          BorderRadius: 10,
          DisplayDigits: 2
      }
    }
  }
...
```

sym-timeseries.js

Scope



Configuration: NgStyle

```
<div ng-style="{'background-color': config.BackgroundColor,
                'border-radius': config.BorderRadius + 'px'}">
  <h3> {{Label}}, {{UOM}} </h3>
  <div>Total number of values: {{Values.length}}</div>
  <div ng-repeat="item in Values" ng-show="{{item.Value > 60}}">
    {{item.Time}}   {{item.Value | number: config.DisplayDigits}}
  </div>

</div>
```

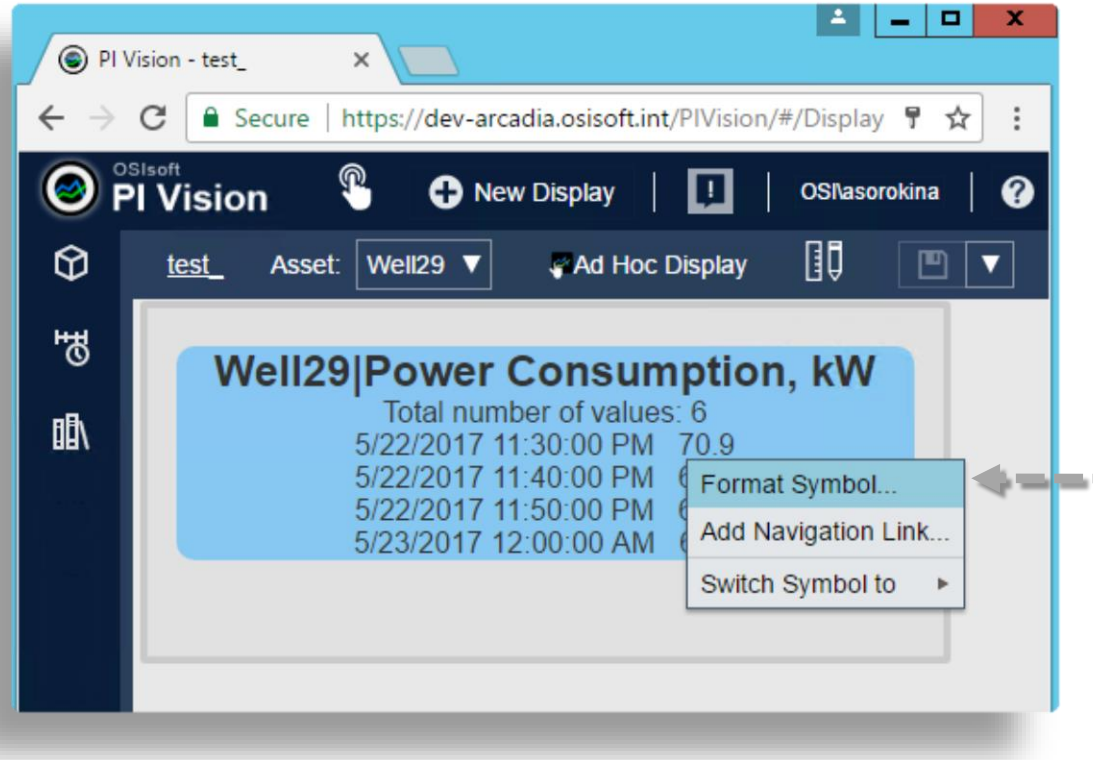
sym-timeseries-template.html

Challenges

Challenge 7 – CSS using NgStyle

1. In sym-archivedata.js, add default configuration options for decimal places, background color, and border radius
2. Set the default background color to **#ff5733**.
3. In sym-archivedata-template.html, add NgStyle directives to apply the styles specified in the js file.

Adding Configuration Pane



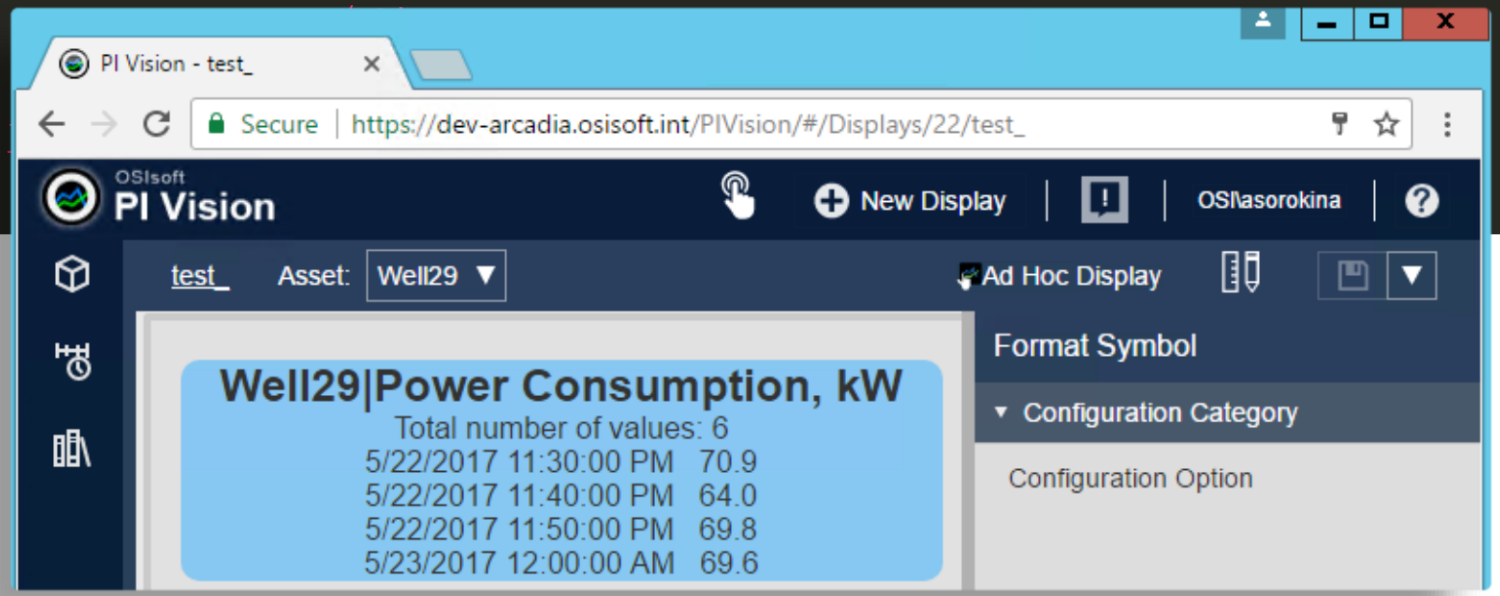

```
var definition = {
    ...
    getDefaultConfig: function(){
        ...
    },
    configOptions: function () {
            return [{
                title: "Format Symbol",
                mode: "format"
            }];
    }
}
```

sym-timeseries.js

Adding Configuration Pane

```
<ul kendo-panel-bar class="t-side-panel-bgcolor" style="font-size: 14px;">
      <li class="k-state-active">
            <span class="panel-bar-label-no-wrap">Configuration Category </span>
            <div class="config-option-single-row"
                 style="margin:10px 20px; font-size: 14px;">
                  <div class="config-option-ie11-overflow-fix">
                        <div class="config-label"> Configuration Option </div>
                  </div>
```

sym-timeseries-config.html

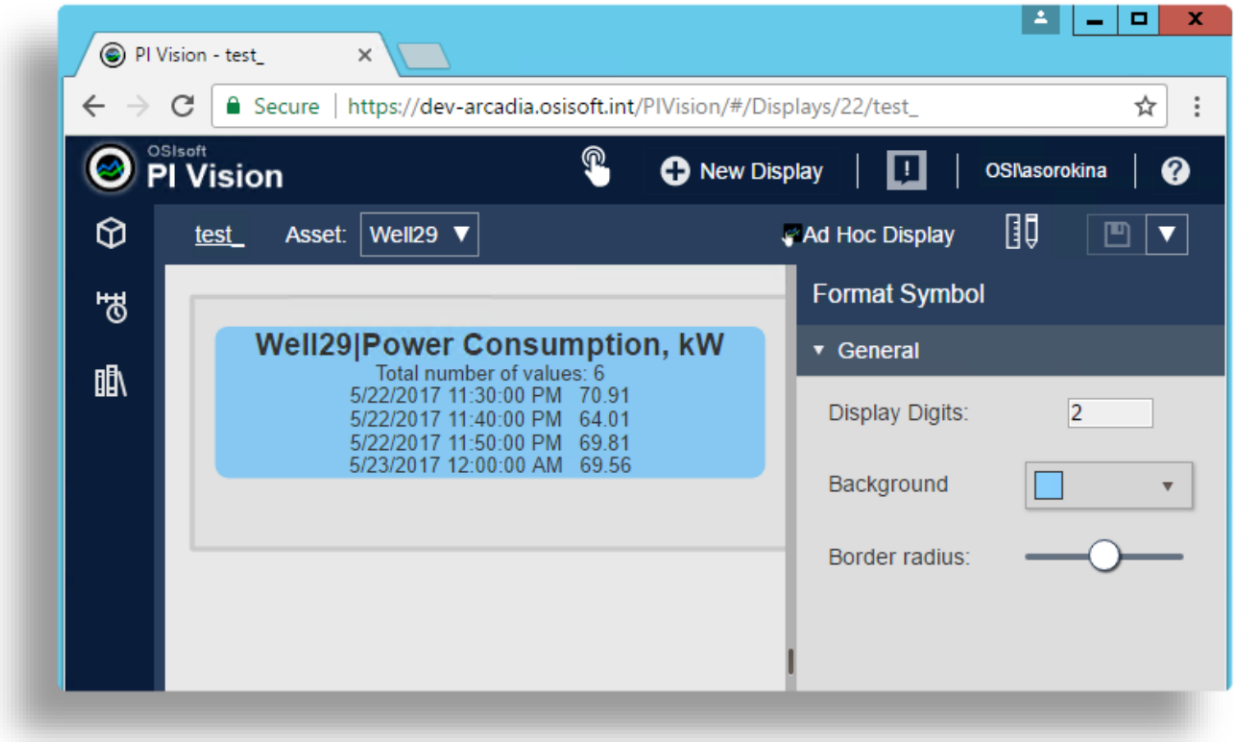


To start, use ConfigExample.html

Adding Configuration Pane

Configuration options for:

1. Display digits
2. Background color
3. Border radius



How do you ask for these in HTML?

Any good examples in the native symbols?

Adding Configuration: Number Input

```
<ul kendo-panel-bar class="t-side-panel-bgcolor" style="font-size: 14px;">
   <li class="k-state-active">
      <span class="panel-bar-label-no-wrap"> General </span>

      <div class="config-option-single-row"
           style="margin-left:20px; font-size: 14px;">
          <div class="config-option-ie11-overflow-fix">
              <div class="config-label">Display Digits:</div>
              <input type="number" min="0" max="10"
                 style="width:100px; height:100%; float: right; margin-
                        right:25px;">
          </div>
      </div>
   </li>
</ul>
```

sym-timeseries-config.html

Adding Configuration: Color Picker

```
<ul kendo-panel-bar class="t-side-panel-bgcolor" style="font-size: 14px;">
   <li class="k-state-active">

      ...

      <div style="width: 85%; margin: 10px 0 15px 20px;">
         <pv-color-picker colorkeyword="Background">
         </pv-color-picker>
      </div>

   </li>
</ul>
```

sym-timeseries-config.html

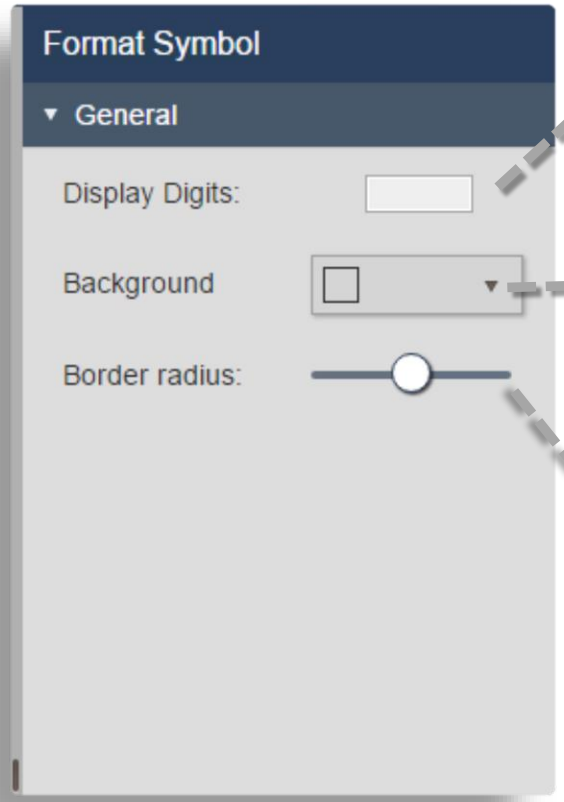
Adding Configuration: Range Input

```
<ul kendo-panel-bar class="t-side-panel-bgcolor" style="font-size: 14px;">
   <li class="k-state-active">
      ...
      <div class="config-option-single-row"
           style="margin-left:20px; font-size: 14px;">
        <div class="config-option-ie11-overflow-fix">
          <div class="config-label">Border radius:</div>
          <input class="config-option-single-row-input" type="range" min="0"
                 max="20" step="1"
                 style="width:100px; height:100%; float: right; margin-right:25px;"
                 title="{{config.BorderRadius}}">
        </div>
      </div>

   </li>
</ul>
```

sym-timeseries-config.html

Configuration Pane

sym-timeseries-config.html




```
...
 <input type="number" min="0" max="10"
  style="width:100px; height:100%; float: right;
         margin-right:25px;">
...
```

```
...
<pv-color-picker colorkeyword="Background">
</pv-color-picker>
...
```

```
...
<input class="config-option-single-row-input" type="range"
   min="0" max="20" step="1"
   style="width:100px; height:100%;
          float: right; margin-right:25px;"
   title="{{config.BorderRadius}}">
...
```

Now, how do we bind these options to the symbol properties?

NgModel directive

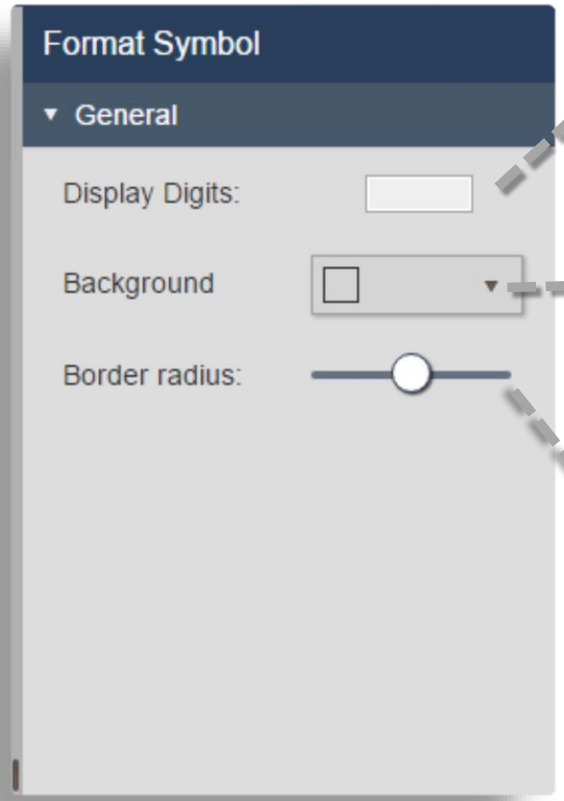
```
      ng-model="config.DisplayDigits"
                                        ng-model="config.BackgroundColor"

ng-model="config.BorderRadius"
```

sym-timeseries-config.html


```
<div ng-style="{'background-color': config.BackgroundColor,
                'border-radius': config.BorderRadius + 'px'}">
  <h3> {{Label}}, {{UOM}} </h3>
  <div>Total number of values: {{Values.length}}</div>
  <div ng-repeat="item in Values" ng-show="{{item.Value > 60}}">
    {{item.Time}}   {{item.Value | number: config.DisplayDigits}}
  </div>
</div>
```

sym-timeseries-template.html

NgModel directive



sym-timeseries-config.html

```
...
 <input type="number" min="0" max="10"
  style="width:100px; height:100%; float: right;
         margin-right:25px;" ng-model="config.DisplayDigits">
...
```

```
...
<pv-color-picker colorkeyword="Background"
     ng-model="config.BackgroundColor">
</pv-color-picker>
...
```

```
...
<input class="config-option-single-row-input" type="range"
   min="0" max="20" step="1" ng-model="config.BorderRadius"
   style="width:100px; height:100%;
          float: right; margin-right:25px;"
   title="{{config.BorderRadius}}">
...
```

Challenges

Challenge 8 – Style using Config Pane

1. In sym-archivedata-config.html, add configuration parameters for decimal places, background color and border radius
2. Bind these to the properties within sym-archivedata-template.html
3. Extra credit: Use the onConfigChange() callback function to find changed configuration parameter and output its old and new value to console.

What we've learned so far

- Displaying arrays – with NgRepeat directive
- New Symbol Data Shape - Timeseries
- Conditional visibility - NgShow directive
- Formatting output - Angular Filters
- Applying style with CSS
- Configuration – Number Input, Range Input, Color Picker
- Binding configuration values - NgModel
- Setting Default Configuration

Charts

Chart libraries

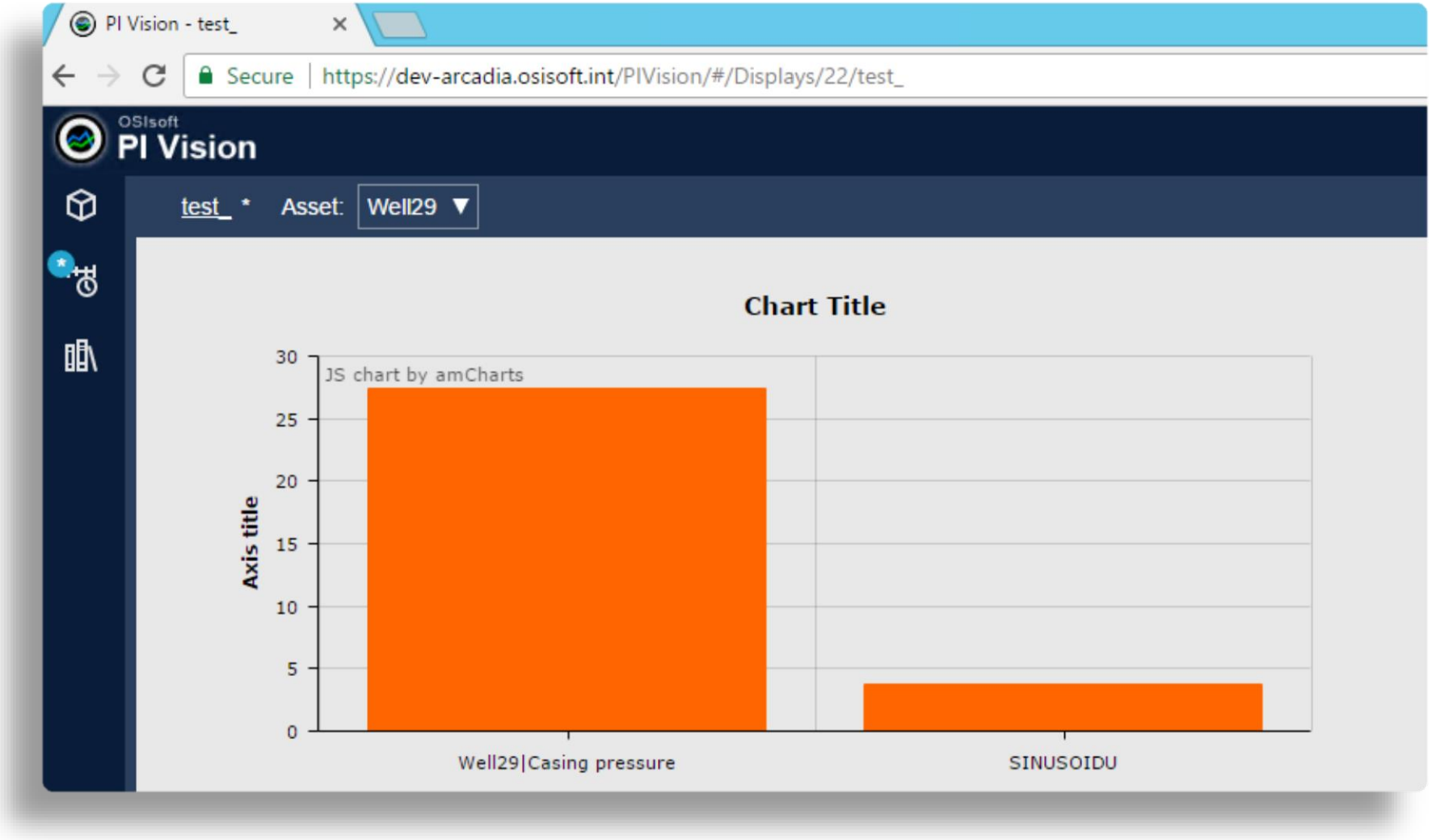
- AmCharts
  https://www.amcharts.com/demos/
- Google charts
  https://developers.google.com/chart/interactive/docs/gallery
- Highcharts
  http://www.highcharts.com/demo
- Plotly
  https://plot.ly/javascript/#basic-charts
- D3
  https://github.com/d3/d3/wiki/Gallery
- …

Remember about licensing!

Low level library – use only if you know what you're doing!

20 BEST JS CHART LIBRARIES: /s
https://thenextweb.com/dd/2015/06/12/20-best-javascript-chart-libraries/

A Bar Chart symbol



Symbol definition


sym-amcharts-bar.js

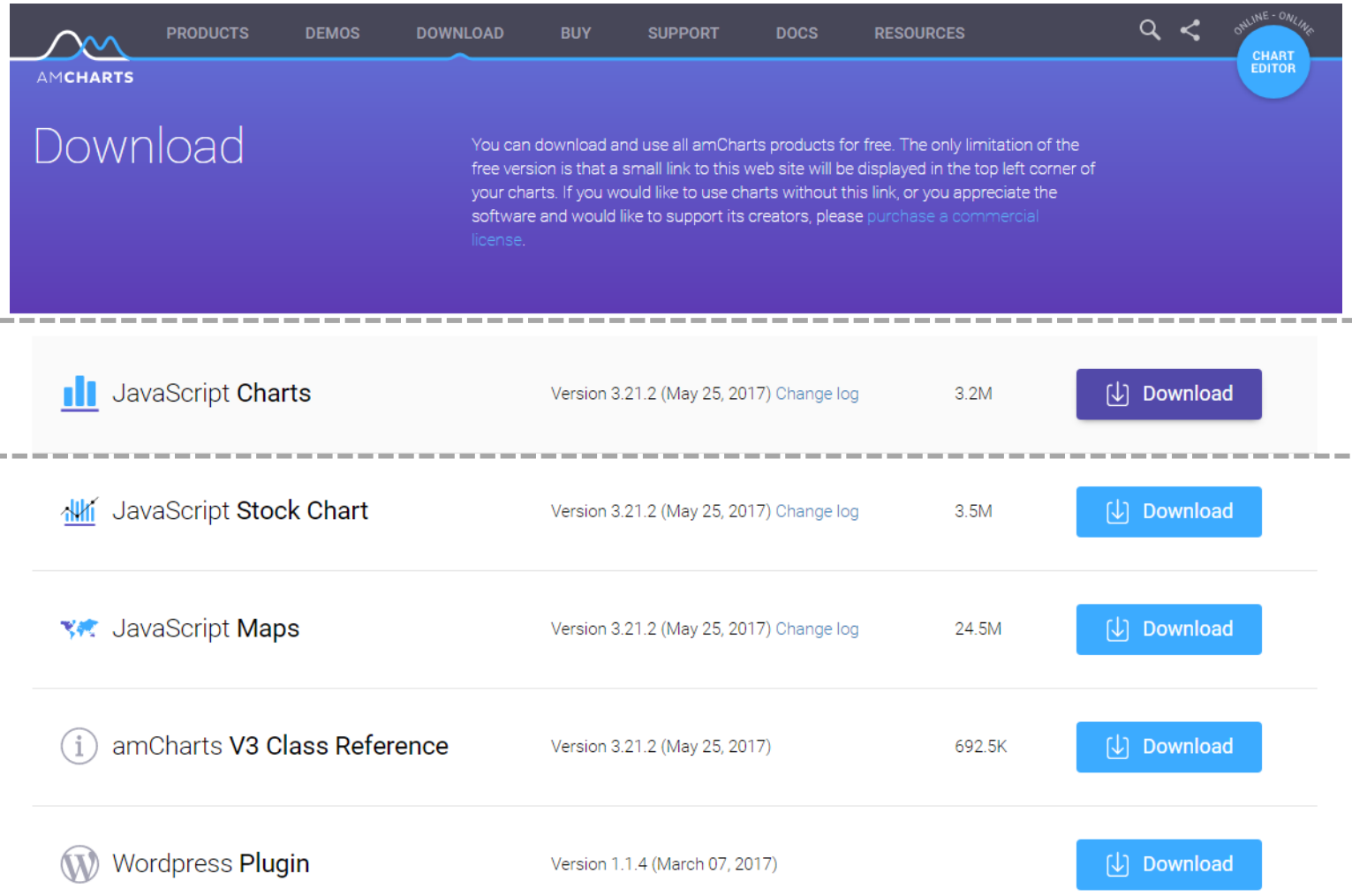

sym-amcharts-bar-config.html


sym-amcharts-bar-template.html

```
...
var definition = {
    typeName: "amcharts-bar",
    ...

}
...
```

sym-amcharts-bar.js

Downloading the AmCharts chart library



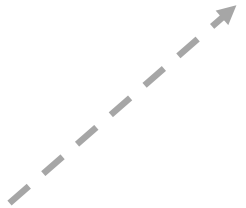
amcharts.js
serial.js

\ext\libraries

https://www.amcharts.com/download/

Working with AmCharts

AmCharts.makeChart(container, config)



An id of the HTML container element

JSON object of the chart configuration

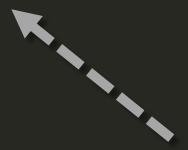
https://www.amcharts.com/kbase/your-first-chart-with-amcharts/

Working with AmCharts: Container Id

```
<div id="container" style="width:100%;height:100%"></div>
```

sym-amcharts-bar-template.html

```
...


symbolVis.prototype.init = function(scope, elem) {

  var symbolContainerDiv = elem.find("#container")[0];
  symbolContainerDiv.id = "barChart_" + Math.random().toString(36).substr(2, 16);


}

...
```



First argument for AmCharts.makeChart()

sym-amcharts-bar.js

Working with AmCharts: Chart Config

```
<div id="container" style="width:100%;height:100%"></div>
```

sym-amcharts-bar-template.html

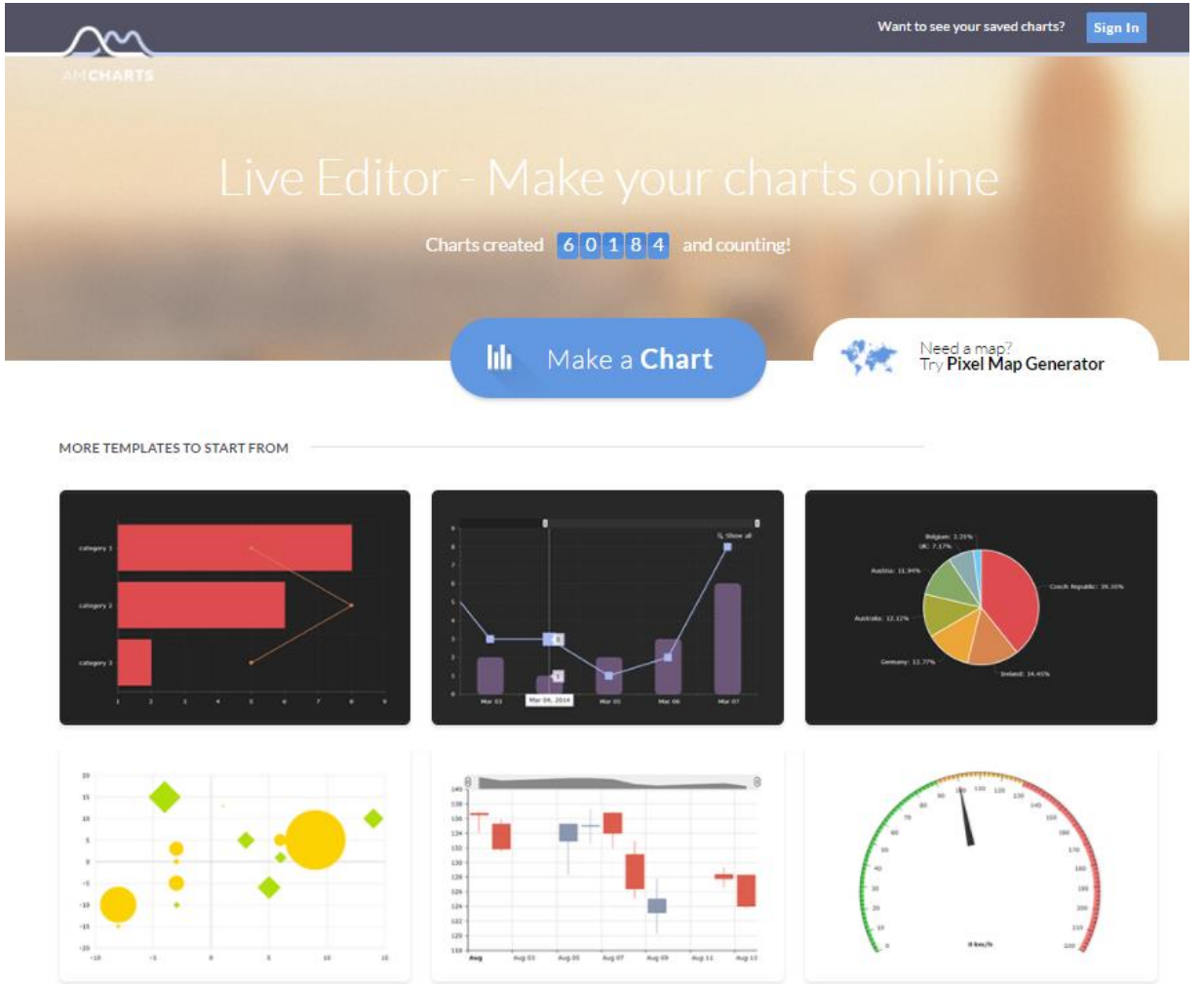
```
...
function getConfig(){
   return <...>
}

symbolVis.prototype.init = function(scope, elem) {

  var symbolContainerDiv = elem.find("#container")[0];
  symbolContainerDiv.id = "barChart_" + Math.random().toString(36).substr(2, 16);
  var chartConfig = getConfig();
  AmCharts.makeChart(symbolContainerDiv.id, chartConfig);

}
...
```

Here is where we'll return a chart configuration object

Where can we borrow an example of chart configuration?

sym-amcharts-bar.js

AmCharts Live Editor



https://live.amcharts.com/

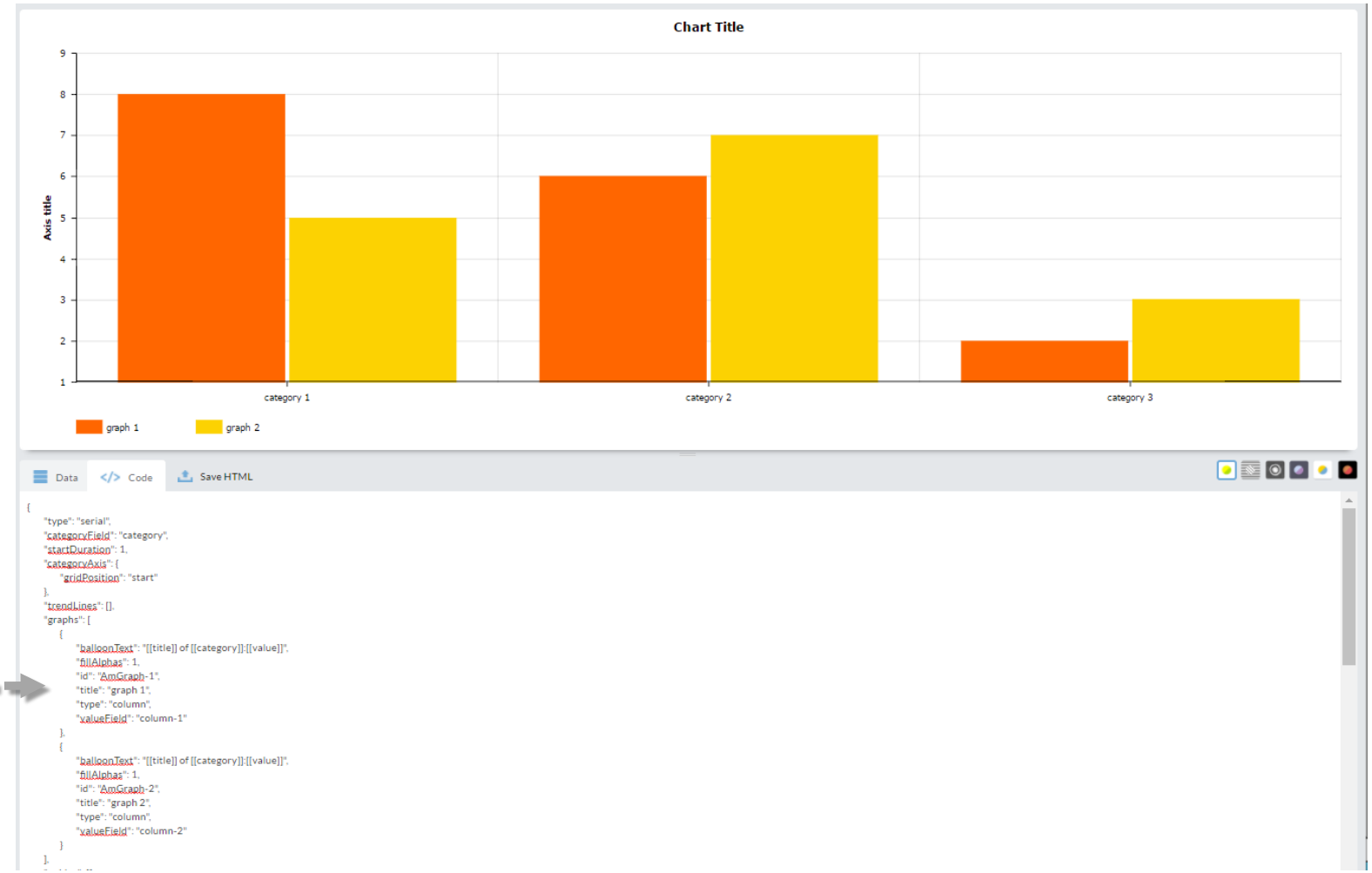
AmCharts Live Editor

We are going to use this as an example

https://live.amcharts.com/

AmCharts Live Editor: Copy Chart Config

Here is our Chart Config.
Ctrl+A. Ctrl+C.



https://live.amcharts.com/

Working with AmCharts: Chart Config

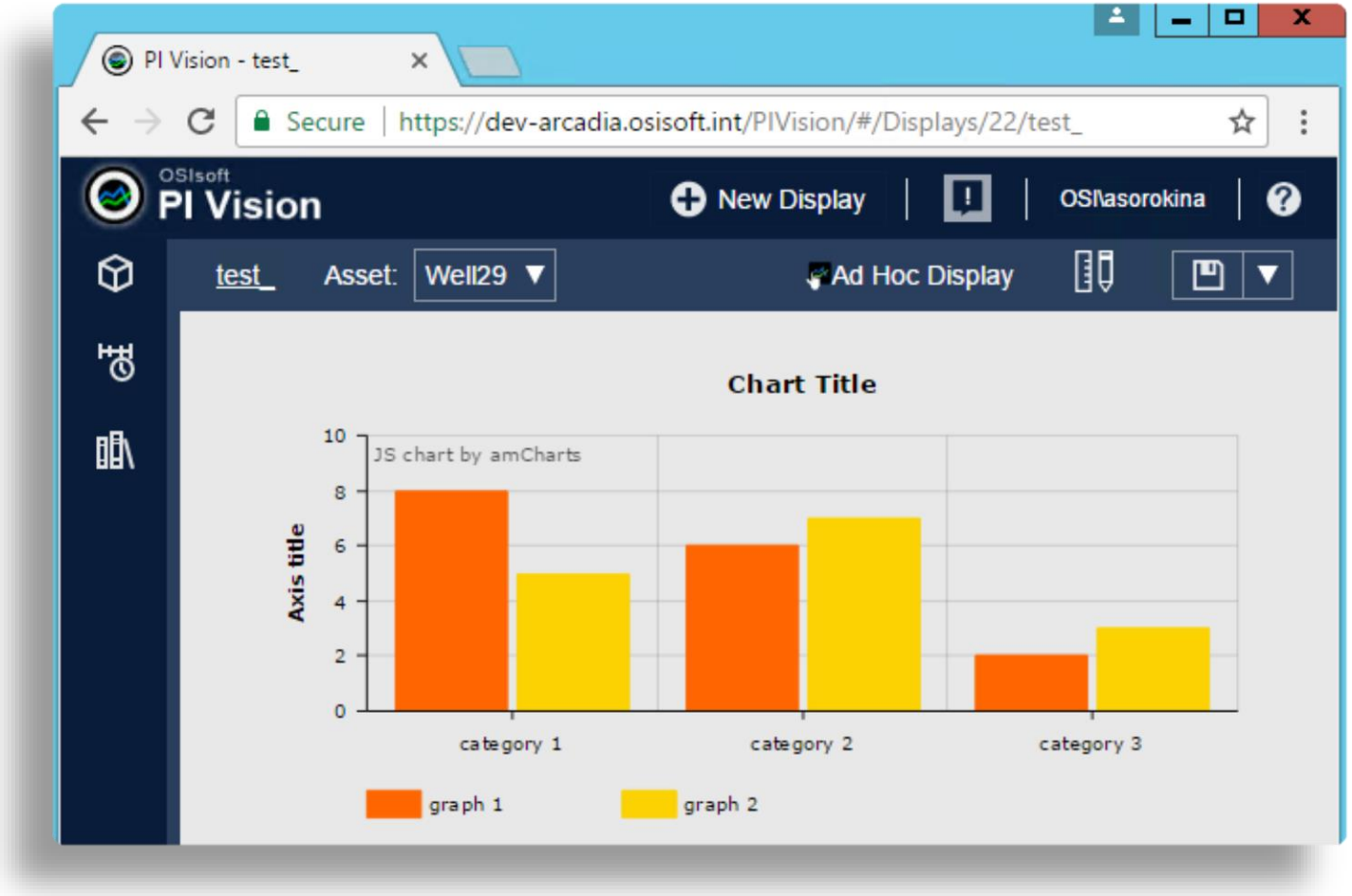
```
<div id="container" style="width:100%;height:100%"></div>
```

sym-amcharts-bar-template.html

```
function getConfig(){
   return < Replace this part with Chart Config from the AmCharts webpage >
}

symbolVis.prototype.init = function(scope, elem) {

  var symbolContainerDiv = elem.find("#container")[0];
  symbolContainerDiv.id = "barChart_" + Math.random().toString(36).substr(2, 16);

  var chartConfig = getConfig();
  AmCharts.makeChart(symbolContainerDiv.id, chartConfig);

}
...
```

sym-amcharts-bar.js

Working with AmCharts



Challenges

Challenge 9 – Working with AmCharts

1. Create files for a custom symbol named **barchart**
2. Download the AmCharts JavaScript library files and place amcharts.js and serial.js files in the **libraries** directory
3. Pick any **serial** chart from the AmCharts Live Editor and display it in the custom symbol

Working with AmCharts: Chart Config

```
...
var chartConfig = {
      "type": "serial",
      "categoryField": "category",
      "graphs": [
          { ...
            "valueField": "column-1"
          },
          { ...
            "valueField": "column-2"
          }
     ],
    ...
```

x-axis

y-axis

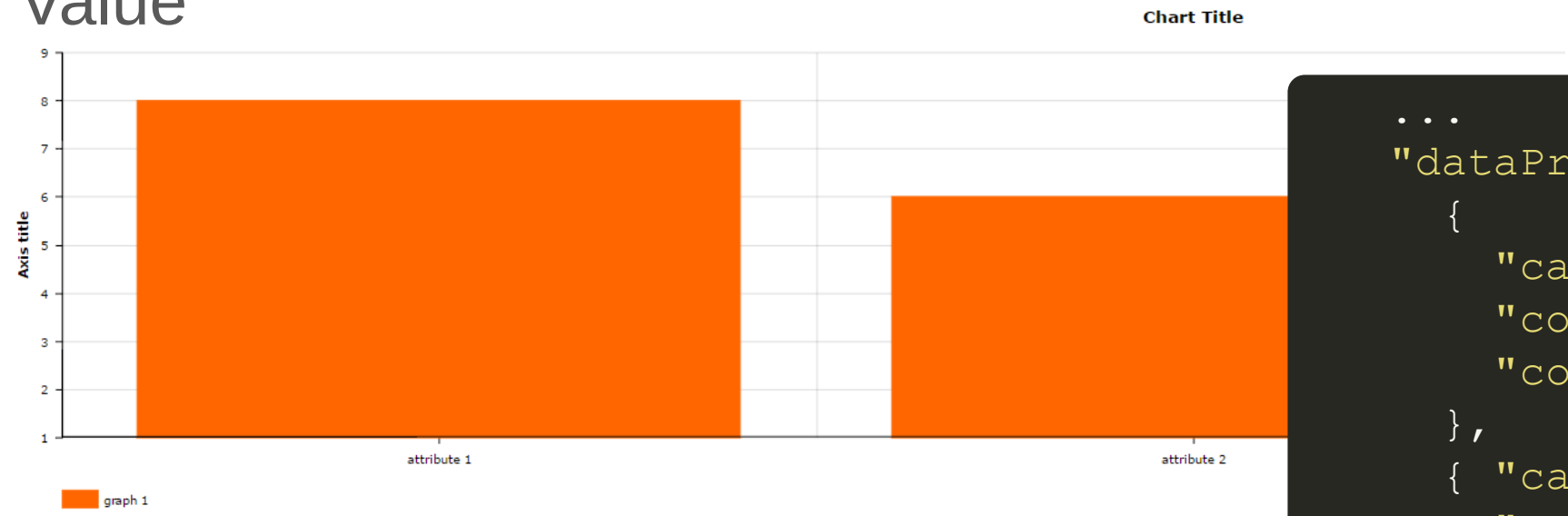
continues ->

```
    ...
    "dataProvider": [
        {
            "category": "category 1",
            "column-1": 8,
            "column-2": 5
        },
        { "category": "category 2",
            "column-1": 6,
            "column-2": 7
        },
        ...

}
...
```

sym-amcharts-bar.js

Naming Category and Value fields

Value



Attribute

```
    ...
    "dataProvider": [
      {
        "category": "category 1",
        "column-1": 8,
        "column-2": 5
      },
      { "category": "category 2",
        "column-1": 6,
        "column-2": 7
      },
      ...

}
...
```

sym-amcharts-bar.js

Naming Category and Value fields

Value

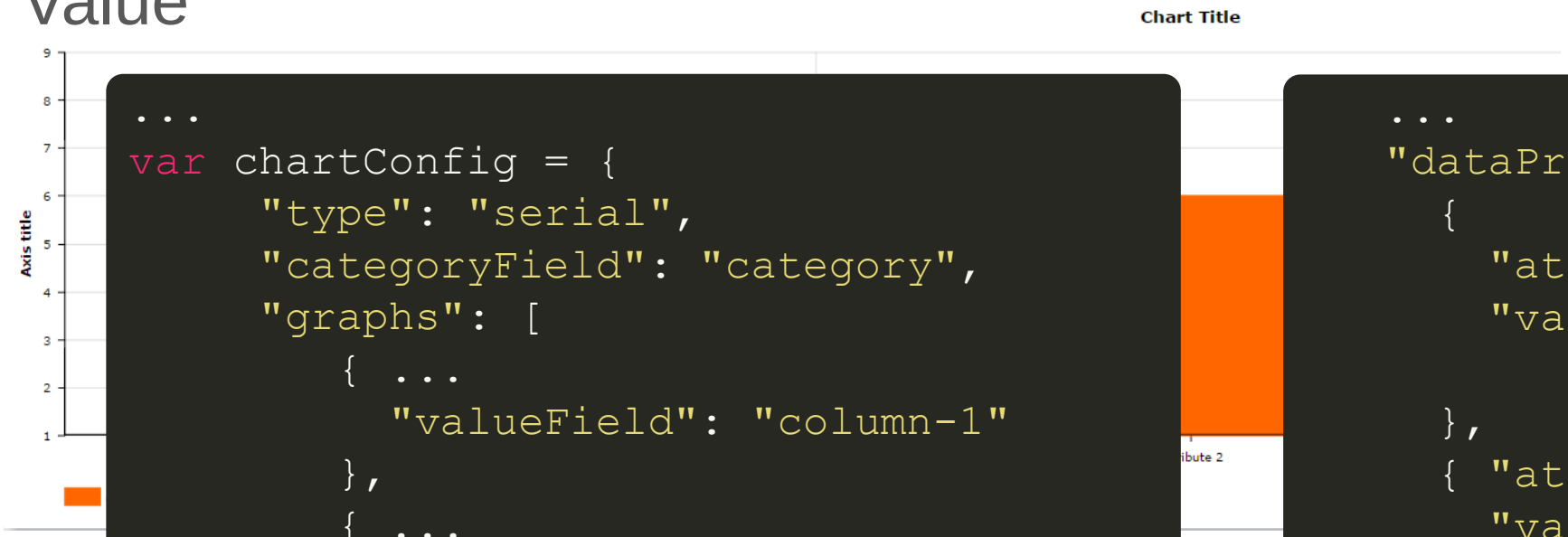
Attribute

```
    ...
    "dataProvider": [
      {
        "attribute": "attribute 1",
        "value": 8,

      },
      { "attribute": "attribute 2",
        "value": 6,

      },
      ...

}
...
```

sym-amcharts-bar.js

Naming Category and Value fields

Value



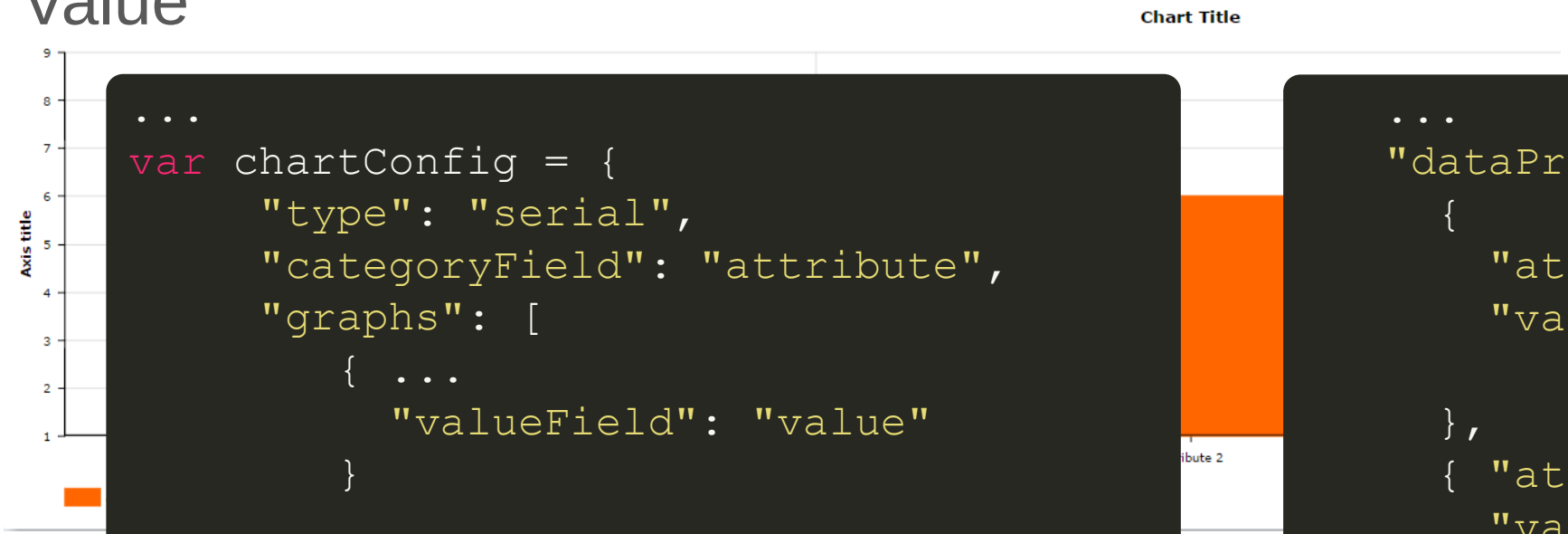

```
...
var chartConfig = {
      "type": "serial",
      "categoryField": "category",
      "graphs": [
          { ...
            "valueField": "column-1"
          },
          { ...
            "valueField": "column-2"
          }
      ],
    ...
```

```
    ...
    "dataProvider": [
      {
        "attribute": "attribute 1",
        "value": 8,

      },
      { "attribute": "attribute 2",
        "value": 6,

      },
      ...

}
...
```

sym-amcharts-bar.js

Naming Category and Value fields

Value




```
...
var chartConfig = {
      "type": "serial",
      "categoryField": "attribute",
      "graphs": [
          { ...
            "valueField": "value"
          }



    ],
   ...
```

```
    ...
    "dataProvider": [
      {
        "attribute": "attribute 1",
        "value": 8,

      },
      { "attribute": "attribute 2",
        "value": 6,

      },
      ...

}
...
```

sym-amcharts-bar.js

New Data Shape: Table

Full data update with metadata (sporadic)

```
Object {…}
Rows, array of:
    Label
    Path
    Units
    Time
    Value
SymbolName
```

Data value update (typical)

```
Object {…}
Rows, array of:
    Time
    Value

SymbolName
```

```
...

    var definition = {
        ...
        getDefaultConfig: function(){
            return {
                DataShape: "Table",
                Height: 150,
                Width: 150
            }
        }
    }
...
```

sym-simplevalue.js

More Shapes: [PI Vision Extensibility Documentation]

console.log(data) output for AF Attribute

Full data update with metadata (sporadic)

```
Object {…}
Rows:Array(1)
       0:Object
           Label:"Well29|Power Consumption"
           Path:"af:\\ARCADIA\OSIDemo Oil and Gas Well Downtime Tracking\West Texas\Clear Fork\Well29|Power
           Consumption"
           Time:"5/20/2017 4:10:00 PM"
           Units:"kW"
           Value: "72.686"
       length:1
SymbolName:"Symbol0"
```

console in browser (F12)

Data value update (typical)

```
Object {…}
Rows:Array(1)
       0:Object
           Time:"5/20/2017 4:10:00 PM"
           Value: "72.686"
       length:1
SymbolName:"Symbol0"
```

console in browser (F12)

console.log(data) output for PI Point

Full data update with metadata (sporadic)

```
Object {…}
Rows:Array(1)
       0:Object
           Description:"12 Hour Sine Wave"
           IsGood:false
           Label:"SINUSOID"
           Path:"pi:\\STARK\SINUSOID"
           Time:"5/20/2017 4:11:00 PM"
           Value: "Scan Off"
       length:1
SymbolName:"Symbol0"
```

console in browser (F12)

Data value update (typical)

```
Object {…}
Rows:Array(1)
       0:Object
           IsGood:false
           Time:"5/20/2017 4:10:00 PM"
           Value: "Scan Off"
       length:1
SymbolName:"Symbol0"
```

Scan Off is not a good value

console in browser (F12)

Conversion to the Chart format

Data format from PI Vision

```
Rows, array of:
      Label
      Path
      Units
      Time
      Value
SymbolName
```

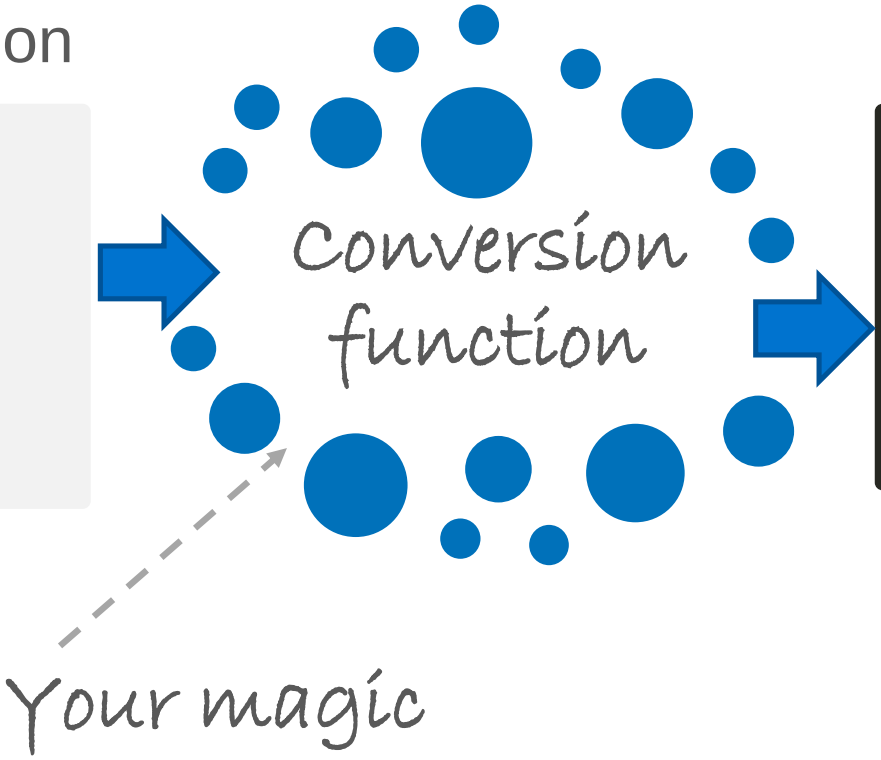


Chart input data format

```
"dataProvider", array of:
       "attribute"
       "value"
```

Conversion to the Chart format: Example

Data from PI Vision

```
Rows: [
      {
         Time:"5/20/2017
         4:10:00 PM"
         Value:"72.686"
         Label:"Sinusoid"
      }
]
```

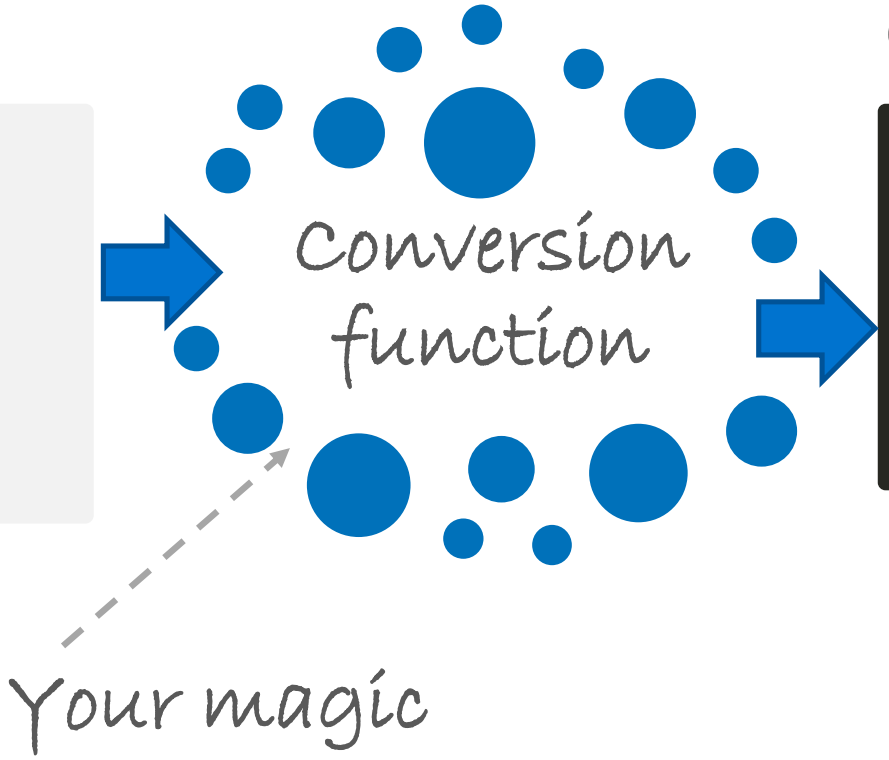


Chart input data

```
"dataProvider": [
        "attribute": "Sinusoid"
        "value": "72.686"
]
```

Working with AmCharts: updating Chart's data

```
...

symbolVis.prototype.init = function(scope, elem) {

 var chart = AmCharts.makeChart(symbolContainerDiv.id, chartConfig);

 function dataupdate(data) {


     var dataprovider = convertToChartDataFormat(data);
     chart.dataProvider = dataprovider;
     chart.validateData();
}


}
...
```

makeChart() returns a chart object that we can work with later in code

sym-amcharts-bar.js

Conversion function

```
...
symbolVis.prototype.init = function(scope, elem) {

 function dataupdate(data) {...}

 function convertToChartDataFormat(data){
   return data.Rows.map(function(item){
          return {
                   value: item.Value,
                   attribute: item.Label
                 }
           });
 }


}
...
```

sym-amcharts-bar.js

Array.map(): https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map

Challenge 10 – AmCharts and Table DataType

1. In sym-amcharts-bar.js, assign makeChart() result to an object and call it **chart**.
2. Add a function to convert data object given by the PI Vision data layer to the format of the chart.
3. Call this function from dataupdate() and assign its result to the chart's dataProvider property.
4. Run validateData() function to update the chart.

Conversion function: first attempt

```
...

symbolVis.prototype.init = function(scope, elem) {

 function dataupdate(data) {...}


 function convertToChartDataFormat(data){
   return data.Rows.map(function(item){
          return {
                   value: item.Value,
                   attribute: item.Label
                 }
           });
 }

}
...
```

Why does this work poorly?

Because Label is not supplied with every data update

sym-amcharts-bar.js

Conversion function: updating meta data (labels)

```
...
symbolVis.prototype.init = function(scope, elem) {
  var labels
  function updateLabel(data){
   labels = data.Rows.map(function(item){
     return item.Label;
   });
 }
 function convertToChartDataFormat(data, labels){
   return data.Rows.map(function(item , index ) {
         return {
                  value: item.Value,
                  attribute: labels[index]
                }
          });
 }
}
...
```

sym-amcharts-bar.js

Conversion function: updating meta data (labels)

```
...

symbolVis.prototype.init = function(scope, elem) {

function dataupdate(data) {
    ...

    if(data.Rows[0].Label) updateLabel(data);
    if( !labels || !chart) return;
    var dataprovider = convertToChartDataFormat(data , labels );

    ...
 }

}
...
```

sym-amcharts-bar.js

What we've learned so far

- Chart libraries overview
- How to work with the AmCharts library
- New Symbol Data Shape - Table
- How to write a data conversion function

Tool Panes

Workspace

PISRV01

PI Vision
PI Data Archive
PI AF Server
MS SQL Server

C:\Program Files\PIPC\PIVision\Scripts\app\editor**tools**

- \ext ← Required. Where symbol files go
- \ext\libraries ← Optional. Where 3rd party libraries and image files go
- \ext\images ← Optional. Where 3rd party libraries and image files go
- \ext\…

PICLIENT01

For testing and troubleshooting purposes only

Essential Tool Pane Files

C:\Program Files\PIPC\PIVision\Scripts\app\editor\tools\ext

Tool Pane Template

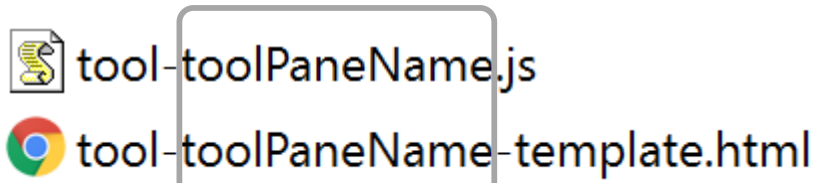
```
(function (PV){
"use strict";

var def = {
     typeName: "",
     inject: ['$rootScope'],
     init: init
   }

function init($rootScope) {

}

PV.toolCatalog.register(def);

})(window.PIVisualization)
```

Minimum code to have the tool pane registered in PI Vision.

Results in a blank pane.

ToolpaneTemplate.js

Tool Pane definition



typeName should agree with the file name

```
...
var def = {
    typeName: "toolPaneName",
    ...

}
...
```

sym-toolPaneName.js

Challenges

Challenge 11 – Hello, World from Tool Pane

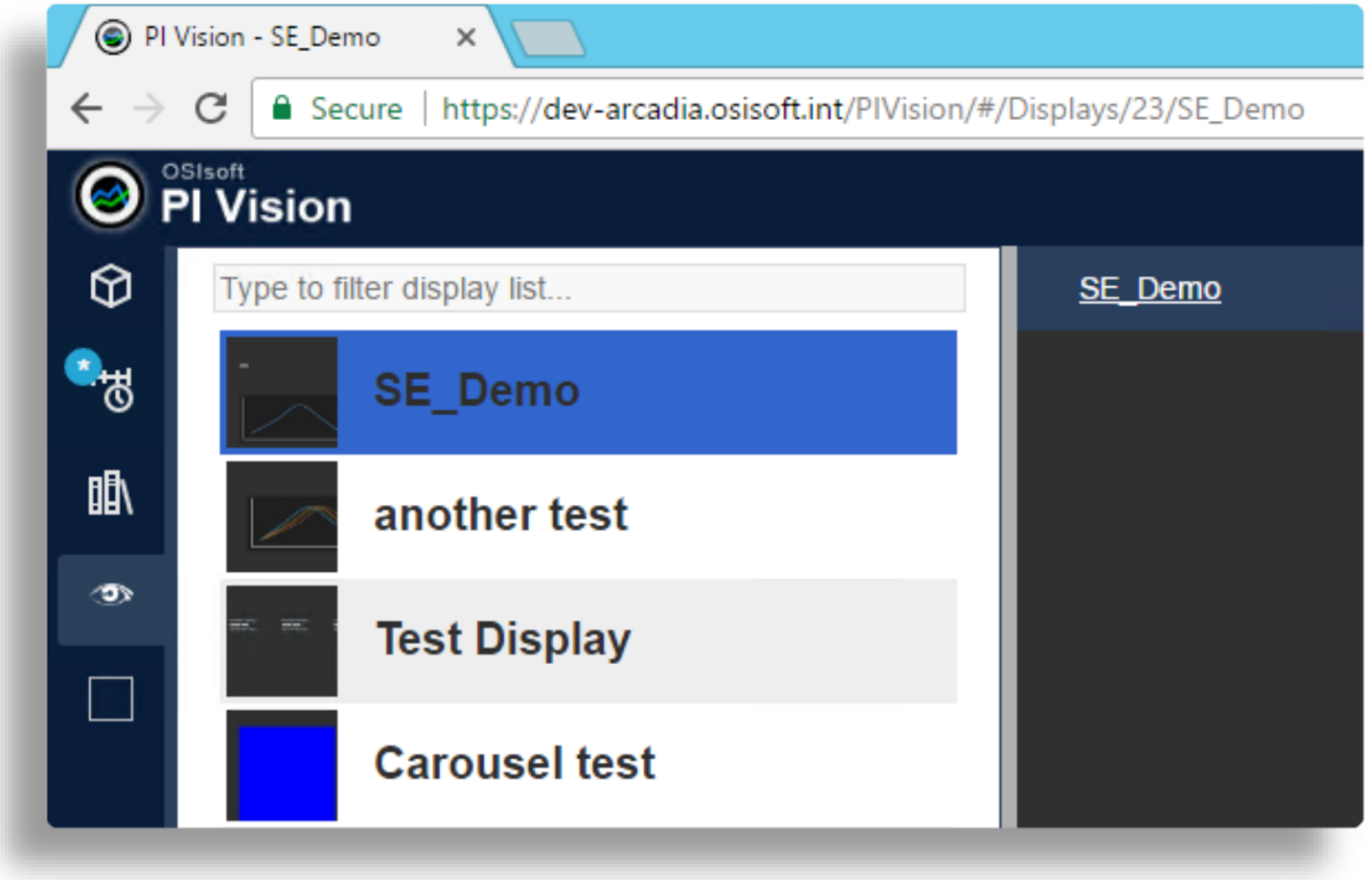
1. Create files for a custom tool pane named **helloworld**
2. Display a button on the tool pane
3. On a click of the button alert a popup message “Hello, world”

Tool Pane – Optional Challenge 12

How do you display this in HTML?

Where can we get a list of displays?

How can we update the display URL?



Deploying custom symbols

Download and Deploy

Community Samples from OSIsoft's GitHub

https://github.com/osisoft/PI-Coresight-Custom-Symbols

Upgrading custom symbols

Download, Deploy and Upgrade

Community Samples from Anna's GitHub from PI Coresight 2016 R2 to PI Vision 2017

https://github.com/AnnaPerry/PI-Coresight-Custom-Symbols/tree/master/Community%20Samples/OSIsoft/manual-data-entry

https://github.com/AnnaPerry/PI-Coresight-Custom-Symbols/tree/master/Community%20Samples/OSIsoft/amcharts-linechart

Follow the README.MD file or PI Vision Extensibility documentation

Miscellaneous

Exploring global objects

PV.ClientSettings.PIWebAPIUrl – PI Vision's PI Web API. For example:

```
["https://dev-arcadia.osisoft.int/piwebapi"]
```

scope.symbol.DataSources – Array of symbol's data sources in format:

```
["af:\\AFServer\ElementPath|Attribute","pi:\\PIDataArchive\Tag"]
```

Extending the app module with service – master class from Lonnie: https://github.com/LonnieBowling/coresight-stack-trend/blob/master/lb-data-service.js

Brainstorming Symbol ideas

Resources

Custom Symbols

- GitHub, https://github.com/osisoft/PI-Coresight-Custom-Symbols
- PI Square, https://pisquare.osisoft.com/community/developers-club/pi-visualization-development
- PI Vision native symbols
- PI Vision Extensibility documentation, https://techsupport.osisoft.com/Downloads/File/98bc271d-9fb6-4317-aa45-30f5c0077558

JS, HTML, CSS and AngularJS

- Google
- w3schools, https://www.w3schools.com/
- JavaScript reference, https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
- AngularJS documentation, https://docs.angularjs.org/api