

NS2024

POWER OF PARTNERSHIP



HTML Visualization

James Johnson

Tridium – Sales Engineer

NS2024
POWER OF PARTNERSHIP

Agenda

- Serving HTML from the station
- RequireJS config
- BajaScript
- Velocity templates
- Bajaux
- Embedding external web server content
- HTTP Header Providers – Content Security Policy
- Cloud dashboards and Niagara APIs

Serving HTML from the Station's File System

- Files can be edited by anyone with write access, without any source control or changelogs.
- Files are not versioned, so they cannot easily be upgraded or distributed to multiple stations.
- Files are not relativized, so not easily used as views on components.
- The HTML5 profile automatically provides the latest API improvements.
- HTML files bypass the profile, so they are more vulnerable to breaking when upgrading Niagara versions

Accessing HTML Files

- The file servlet provides access to files under the station's shared file space.

<https://bldg1f1.ns24.lan/file/index.html>

- The file ORD scheme with the File Download View.

<https://bldg1f1.ns24.lan/ord/file:^index.html|view:web:FileDownloadView>

noSnoop

- The Niagara web server snoops content being served and potentially modifies the HTML or CSS content such as replacing module ORD schemes with module servlet requests.
- Snooping can be disabled in HTML by adding syntax to the top of the file.

```
<!-- @noSnoop -->
```

- Snooping can be disabled in CSS by adding syntax to the top of the file.

```
/* @noSnoop */
```

Using JavaScript Libraries

- When serving HTML content from the station's file system it may be necessary to load JavaScript libraries such as BajaScript, d3, jQuery, underscore and others.
- RequireJS from <https://requirejs.org> uses the Asynchronous Module Definition (AMD) method of loading JavaScript code, which is a modular script loader and will likely improve the speed and quality of your code.
- RequireJS can be loaded in an HTML page using script tags.

```
<script src="/requirejs/config.js"></script>
```

```
<script  
src="/module/js/com/tridium/js/ext/require/require.js"></script>
```

```
local:|module://docDeveloper/doc/requirejs.html
```

HTML Example File

```
<!DOCTYPE html>
<!-- @noSnoop -->
<html>
<head>
  <title>This is a BajaScript test page</title>
  <script type='text/javascript' src='/requirejs/config.js'></script>
  <script type='text/javascript' src='/module/js/com/tridium/js/ext/require/require.min.js?'></script>
</head>
<body>
  <h1>BajaScript test page!</h1>
  <script>
    require(["baja"], function (baja) {
      baja.Ord.make('station:|slot:|').get()
        .then((station) => {
          alert('Station name: ' + station.get('stationName'));
        });
    });
  </script>
</body>
</html>
```

BajaScript

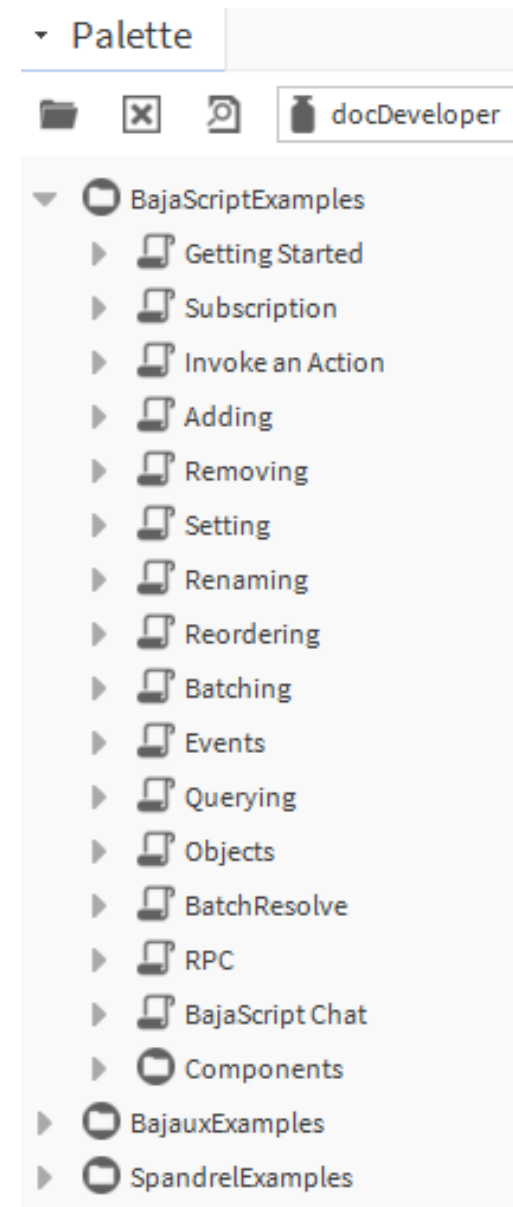
- A framework for building web applications which includes JavaScript APIs for getting data from the station

`local:|module://docDeveloper/doc/jsdoc/bajaScript-ux/index.html`

- Getting started with BajaScript tutorial

`local:|module://docDeveloper/doc/jsdoc/bajaScript-ux/tutorial-gettingStarted.html`

- The docDeveloper palette includes examples



Apache Velocity

- An open-source Java-based web template engine
<https://velocity.apache.org/engine/2.3>
- Uses markup statements in an HTML document which reference methods defined in Java code.
- The Velocity Template Language (VTL) is meant to provide the easiest, simplest, and cleanest way to incorporate dynamic content in a web page.
- VTL statements begin with the # character followed by a directive and use variables stored in the Velocity context that begin with the \$ character.
- `module://docDeveloper/doc/velocity/velocity.htm`

```
<html>
  <body>
    #set( $foo = "Velocity" )
    Hello $foo World!
  </body>
</html>
```

Velocity Servlet (axvelocity module)

- The servlet allows configuring the servlet name (URL path) and provides a manager view for Velocity documents.
- The Velocity Doc maps to a Velocity *.vm file.
- The Velocity Context elements define the variable names used in the *.vm file and bind the variable to a component in the station.

VelocityServlet (Velocity Servlet)	
Status	{ok}
Fault Cause	
Enabled	☒ true
Servlet Name	velocity
▼ AHU_01 Velocity Doc	
Template File	file:^velocity/ahu.vm
Mime Type	text/html
▼ Velocity Context Velocity Context	
▼ device Velocity Context Ord Element	
Ord	station: slot:/Drivers/BacnetNetwork/Floor/AHU_01
Use Profile	☐ false

AHU Velocity Example

```
<!DOCTYPE html>
<html>
<head>
  <title>$device.name Graphic</title>
  <link rel="stylesheet" href="/file/velocity/ahu.css">
</head>
<body>

  <h2>
    $device.name Graphic</br>
  </h2>

  #set( $pointsDeviceExt = $util.get("slot:points", $device))
  #set( $controlPoints = $util.getChildren($util.get("station:|$pointsDeviceExt.slotPath"), "control:ControlPoint") )

  <table border="1">
    <tr>
      <th>
        Display Name
      </th>
      <th>
        Value
      </th>
    </tr>
    #foreach($point in $controlPoints)
      <tr>
        <td>
          $point.getDisplayName($ax.cx)
        </td>
        <td>
          $util.getDisplay($point.out)
        </td>
      </tr>
    #end
  </table>

</body>
</html>
```

AHU_01 (Velocity Doc)

Template File	file:^velocity/ahu.vm
Mime Type	text/html
Velocity Context	Velocity Context
device	Velocity Context Ord Element
Ord	station: slot:/Drivers/BacnetNetwork/Floor/AHU_01
Use Profile	☒ false

Json Schema – History Query

{ } EnergyData (Json Schema)

Output

```
[
  [
    "timestamp",
    "kwh"
  ],
  [
    "2024-02-25 12:00 AM",
    84.2
  ],
  [
    "2024-02-25 12:15 AM",
    88.8
  ],
  [

```

Generate

Copy

Clear Output

Output History

Metrics

Indented Display

Enabled ☒ true

Status {ok}

Fault Cause

Last Updated 07-Mar-2024 09:49 AM EST

Config Json Schema Config Folder

Queries Json Schema Query Folder

Query Interval 000000h 10m 00s [0ms--inf]

Last Query Completed Timestamp 07-Mar-2024 11:07 AM EST

{q} HistoryQuery Json Schema Query

Query Ord history:/HtmlVisualization/MainKwh?period=lastWeek|bql:selec

Last Result Size 576

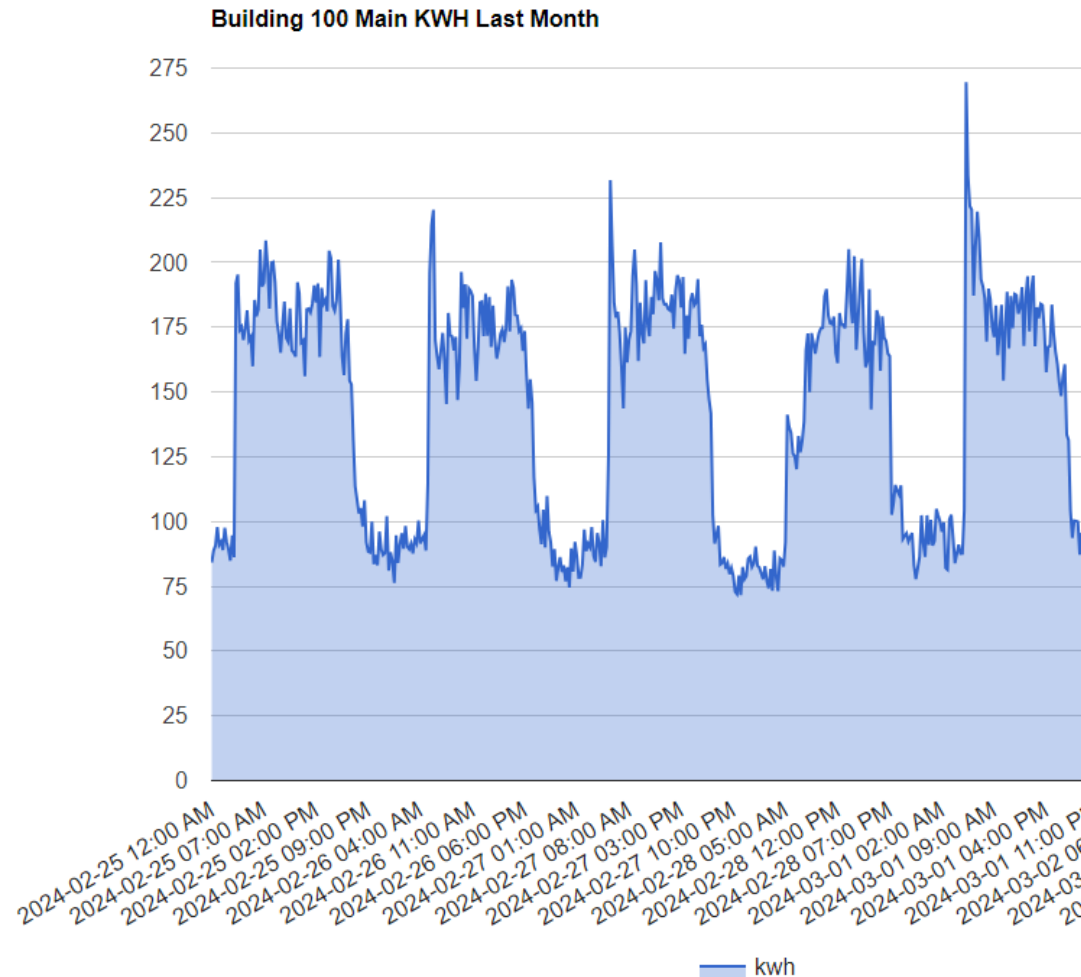
{q} BoundQueryResult Json Schema Bound Query Result

Query HistoryQuery

Output Style Row Array With Header Preview

Google Chart Velocity Example

Energy (Velocity Doc)	
Template File	file:^velocity/energy.vm
Mime Type	text/html
Velocity Context	Velocity Context
schema	Velocity Context Ord Element
Ord	station: slot:/GoogleChartExample/EnergyData
Use Profile	☐ false



```
<html>
<head>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript">
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);
function drawChart() {
  var data = google.visualization.arrayToDataTable( $schema.output );
  var options = {
    'title': 'Building 100 Main KWH Last Month',
    'legend': { position: 'bottom'},
    width: 1200 ,
    height: 800
  };

  var chart = new google.visualization.AreaChart(
    document.getElementById('chart_div'));

  chart.draw(data, options);
}
</script>
</head>
<body>
<div id="chart_div"></div>
</body>
</html>
```

Web Service – HTTP Header Providers

- Headers pass additional information with an HTTP request or response between client and server.
- Consists of a case-insensitive name followed by a colon, then by its value.
- Information ensures authenticity of the messages providing security against click-jacking and other threats.
- <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>

Http Header Providers	
Enabled	☒ true
Content-Security-Policy	Csp Header Provider
Cross-Origin-Opener-Policy	Generic Http Header Provider
X-Content-Type-Options	X Content Type Options Header Provider
X-Frame-Options	X Frame Options Header Provider
X-XSS-Protection	X Xss Protection Header Provider

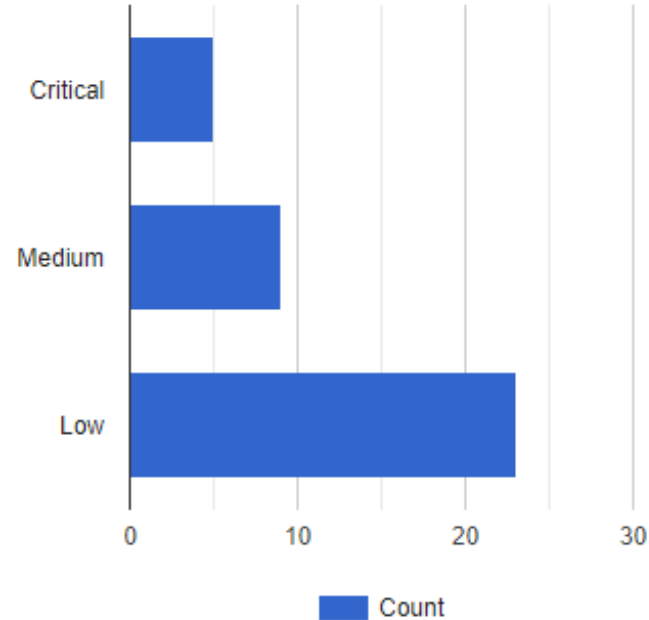
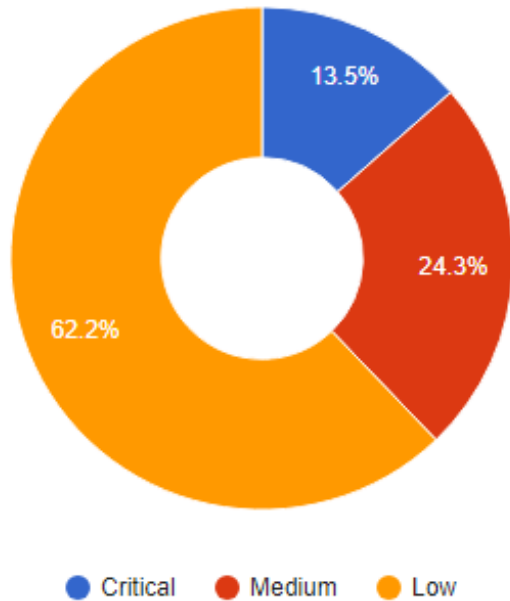
Content-Security-Policy (CSP)

- Response headers which control resources the user agent is allowed to load for a given web page.
- Policies mostly involve specifying origins and script endpoints.
- Helps protect against cross-site scripting (XSS) and data injection attacks.
- The default-src directive applies to other directives which are not explicitly configured.
- Violation Text displays information when a browser client reports as CSP violation to the station's web server.

report-uri	/csp-reports
script-src	'self' workbench 'unsafe-inline' 'unsafe-eval' https://www.gstatic.com
style-src	'self' workbench 'unsafe-inline' https://www.gstatic.com

Google Chart Example 2

Alarm Counts By Class



```
<html>
<head>
  <script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
  <link rel="stylesheet" href="/file/velocity/alarm.css">
  <script type="text/javascript">
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawPieChart);
    google.charts.setOnLoadCallback(drawBarChart);

    function drawPieChart() {
      var data = new google.visualization.DataTable();
      data.addColumn('string', 'Priority');
      data.addColumn('number', 'Count');
      data.addRows([
        [ '$critical.getDisplayName($ax.cx)', $critical.get("out").get("value") ],
        [ '$medium.getDisplayName($ax.cx)', $medium.get("out").get("value") ],
        [ '$low.getDisplayName($ax.cx)', $low.get("out").get("value") ]
      ]);

      var options = {
        'legend': { position: 'bottom'},
        width: 400,
        height: 400,
        pieHole: 0.4
      };

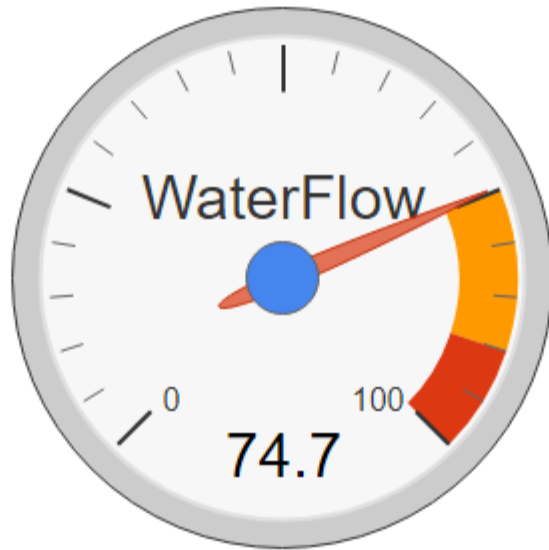
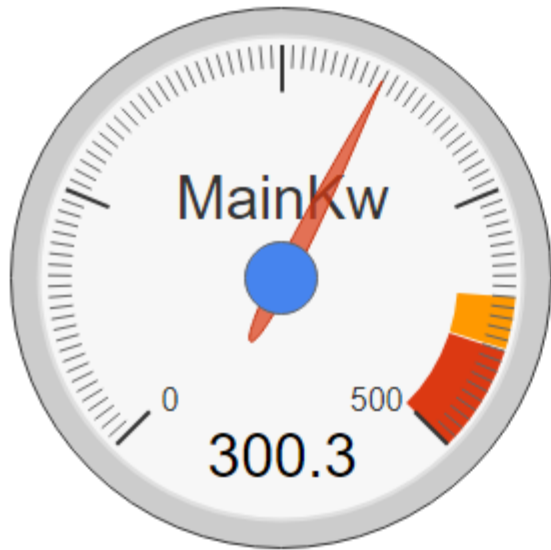
      var pchart = new google.visualization.PieChart(document.getElementById('chart_div'));
      pchart.draw(data, options);
    }

    function drawBarChart() {
      var data = new google.visualization.DataTable();
      data.addColumn('string', 'Priority');
      data.addColumn('number', 'Count');
      data.addRows([
        [ '$critical.getDisplayName($ax.cx)', $critical.get("out").get("value") ],
        [ '$medium.getDisplayName($ax.cx)', $medium.get("out").get("value") ],
        [ '$low.getDisplayName($ax.cx)', $low.get("out").get("value") ]
      ]);

      var options = {
        'legend': { position: 'bottom'},
        width: 400,
        height: 400,
        pieHole: 0.4
      };

      var bchart = new google.visualization.BarChart(document.getElementById('bar_div'));
      bchart.draw(data, options);
    }
  </script>
</head>
<body>
  <div id="alarm_div">
    <p id="chart_title">Alarm Counts By Class</p>
    <div id="chart_div"></div>
    <div id="bar_div"></div>
  </div>
</body>
</html>
```

Google Gauge Example



```
<html>
<head>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<script type="text/javascript" src="/requirejs/config.js"></script>
<script type="text/javascript" src="/module/js/com/tridium/js/ext/require/require.min.js?'"></script>
<link rel="stylesheet" href="/file/velocity/stats.css">
<script type="text/javascript">
  var power = 0, water = 0, gas = 0;

  require(["baja!", 'Promise'], function (baja, Promise) {
    'use strict';

    // A Subscriber is used to listen to Component events in Niagara.
    var sub = new baja.Subscriber();
    sub.attach({
      changed: function (prop, cx) {
        if(prop.getName() === 'out') {
          let objName = this.getName();
          if(objName === 'MainKw') { power = this.getOut().getValue().toFixed(1); }
          else if(objName === 'WaterFlow') { water = this.getOut().getValue().toFixed(1); }
          else if(objName === 'Gas') { gas = this.getOut().getValue(); }
        }
      }
    });

    addEventListener('unload', (event) => {
      if(sub !== null) {
        sub.unsubscribeAll();
        sub.detach();
      }
    });

    // Resolve the ORD to the power control point
    var promise1 = baja.Ord.make("station:|$power.slotPath").get({subscriber: sub})
    .then(()=> {
    });

    // Resolve the ORD to the water control point
    var promise2 = baja.Ord.make("station:|$water.slotPath").get({subscriber: sub})
    .then(()=> {
    });

    // Resolve the ORD to the gas control point
    var promise3 = baja.Ord.make("station:|$gas.slotPath").get({subscriber: sub})
    .then(()=> {
    });

    return Promise.all([ promise1, promise2 ]);
  });

  google.charts.load('current', {'packages':['gauge']});
  google.charts.setOnLoadCallback(drawPower);
  google.charts.setOnLoadCallback(drawWater);
  google.charts.setOnLoadCallback(drawGas);

  function drawPower() {

    var data = google.visualization.arrayToDataTable([
      ['Label', 'Value'],
      ['$power.getDisplayName($ax.cx)', power ]
    ]);

    var options = {
      width: 300, height: 300,
      redFrom: 450, redTo: 500,
      yellowFrom: 425, yellowTo: 449,
      minorTicks: 25,
      max: 500
    };
  }
}
```

Bajaux

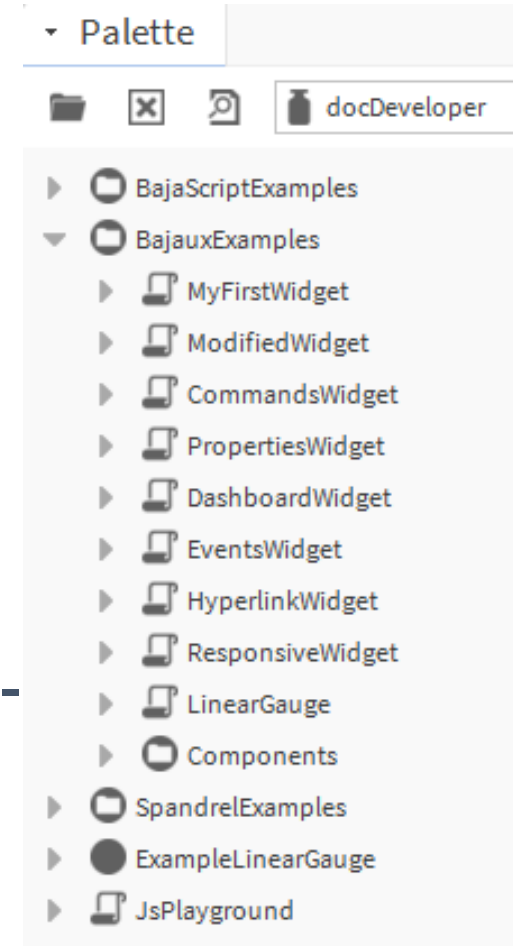
- A framework that enables developers to create widgets for Niagara using HTML5 and JavaScript

`local:|module://docDeveloper/doc/jsdoc/bajaux-ux/index.html`

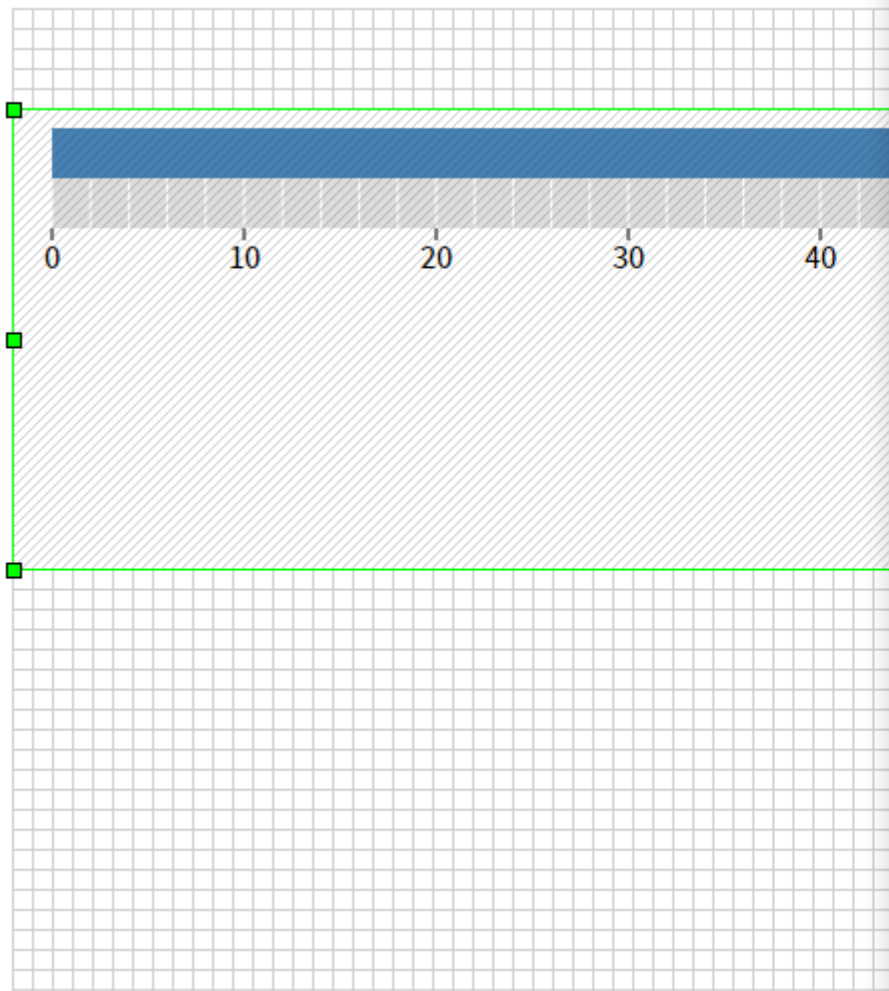
- Getting started with Bajaux tutorial

`local:|module://docDeveloper/doc/jsdoc/bajaux-ux/tutorial-10-mfw-gettingStarted.html`

- The docDeveloper palette includes examples



docDeveloper Linear Gauge Example



Properties

WebWidget

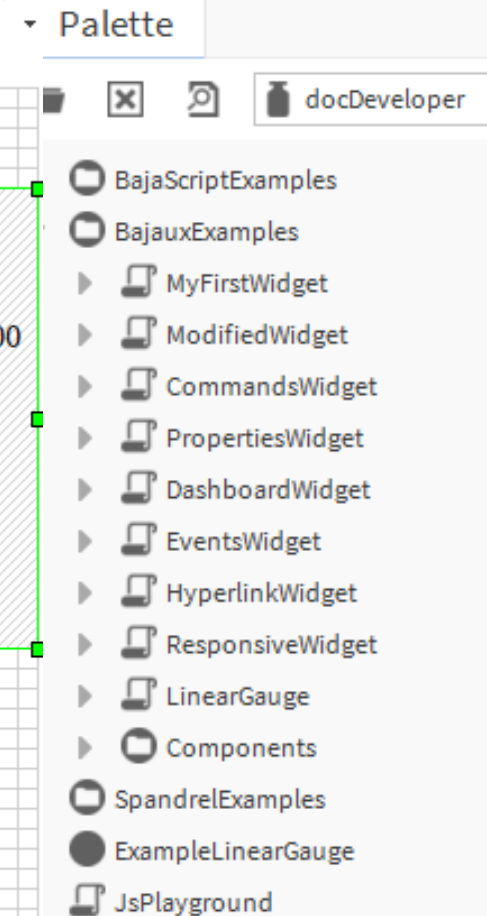
Web Widget

autoScale	true
autoScaleStep	10.00
barColor	
enabled	true
js	view:docDeveloper:ExampleLinearGauge
layer	
layout	0.0,50.0,1000.0,230.0
max	100.00
min	0.00
preferredSize	640.0,480.0
visible	true

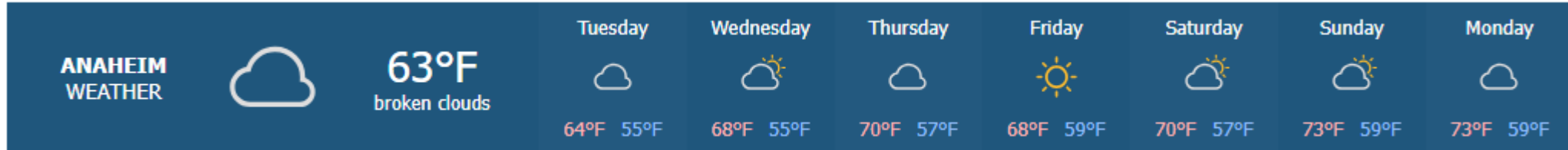
Wb View Binding

ord	station: slot:/GaugeExample/SineWave
degradeBehav	None
requiredPerm	r

OK Cancel

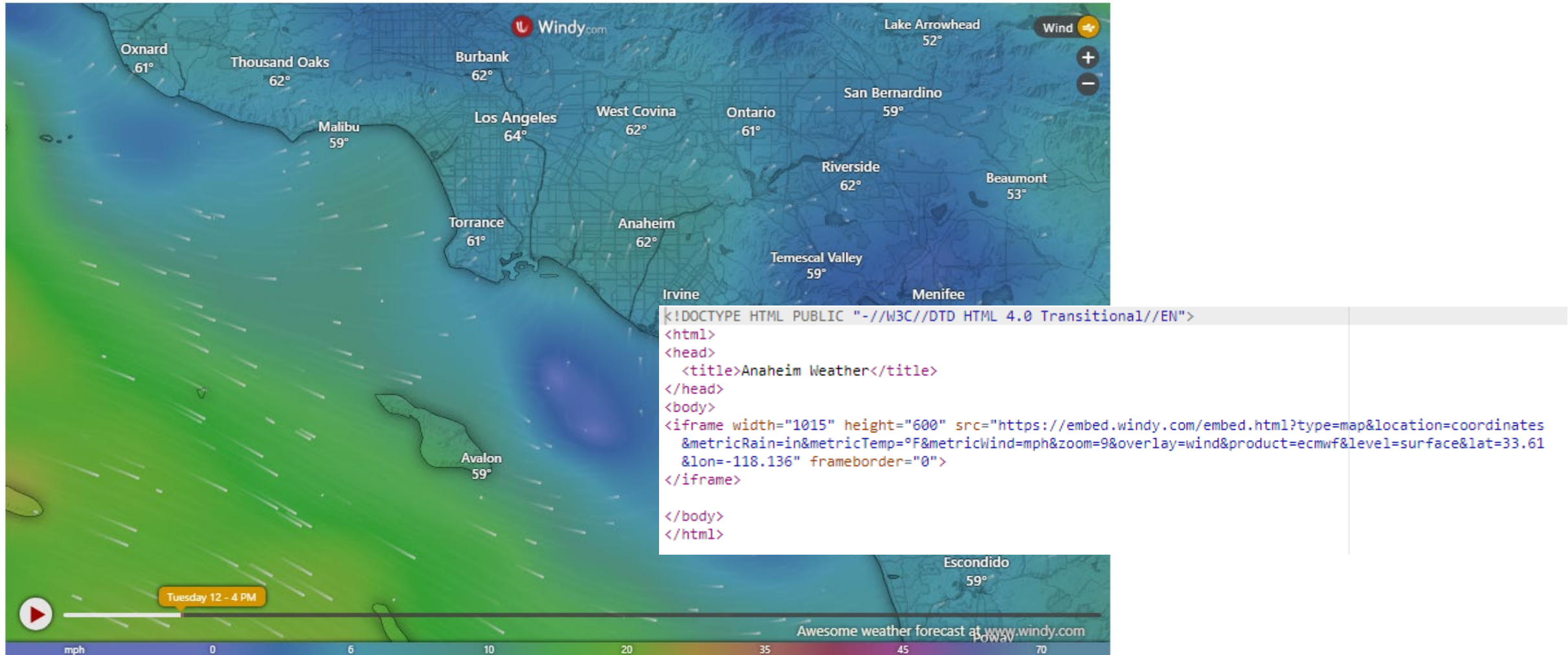


Weatherwidget.io Example

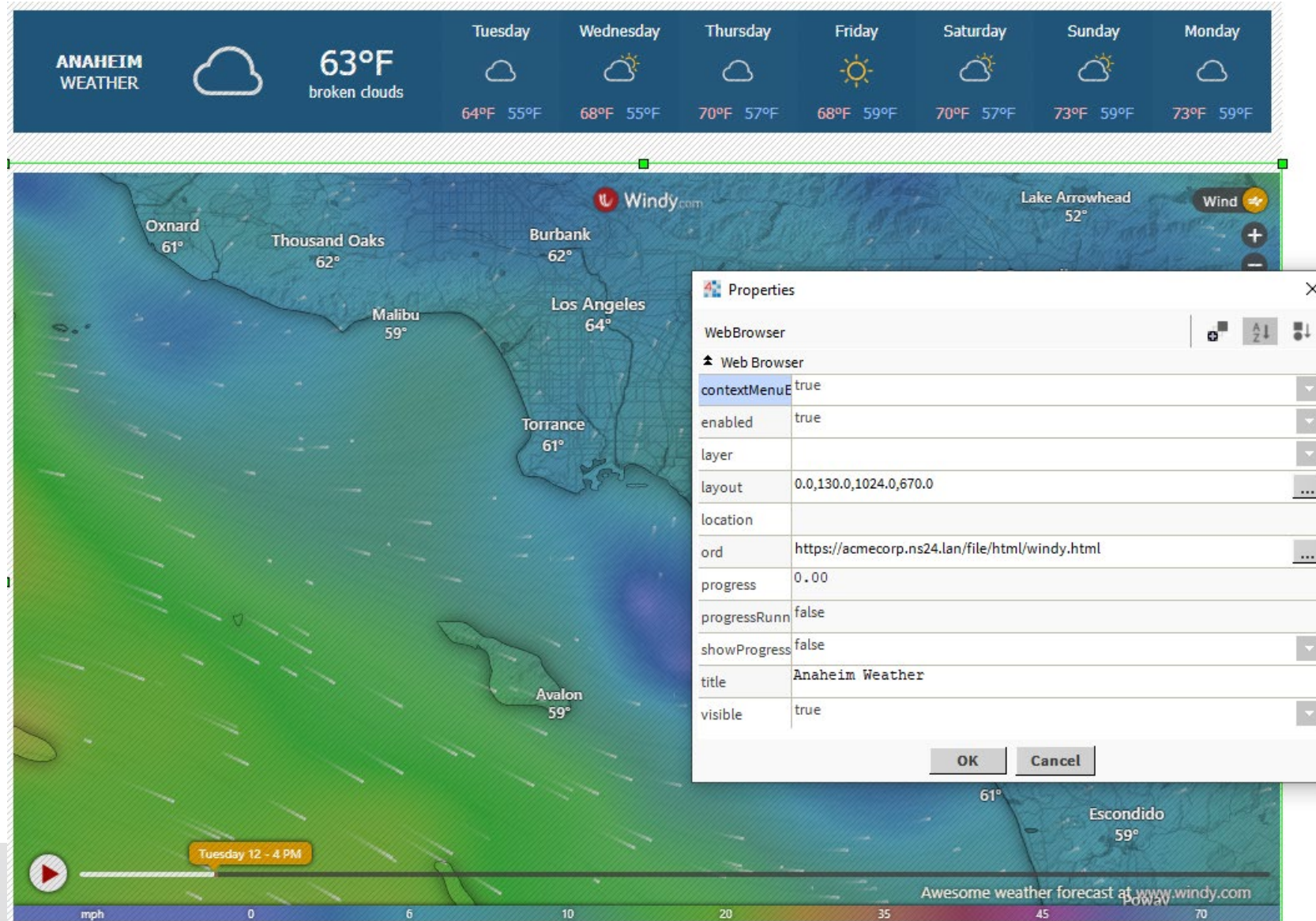


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<!-- @noSnoop -->
<html>
<head>
  <title></title>
</head>
<body>
  <a class="weatherwidget-io" href="https://forecast7.com/en/33d84nl17d9l/anaheim/?unit=us"
    data-label_1="ANAHEIM" data-label_2="WEATHER" data-theme="original" >ANAHEIM WEATHER</a>
  <script>
    !function(d,s,id) {
      var js,fjs=d.getElementsByTagName(s)[0];
      if(!d.getElementById(id)){
        js=d.createElement(s);
        js.id=id;
        js.src='https://weatherwidget.io/js/widget.min.js';
        fjs.parentNode.insertBefore(js,fjs);
      }
    }
    (document,'script','weatherwidget-io-js');
  </script>
</body>
</html>
```

Windy.com Example



Embedding External Web Page Using Px



API Data Access

- oBIX
- Niagara Analytics web API
- Json Toolkit Json Schema
- HTTP Client driver String Servlet
- Custom Servlet
- Niagara Cloud Suite (NCS) using Niagara Data Service (NDS)

oBIX History Query

GET

https://acmecorp.ns24.lan/obix/histories/HtmlDemo/MainKwh/~historyQuery?start=2024-03-12T00:00:00.000-04:00&end=2024-03-12T23:59:59.999-04:00

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Bulk Edit
☒	start	2024-03-12T00:00:00.000-04:00	
☒	end	2024-03-12T23:59:59.999-04:00	

Body Cookies (1) Headers (10) Test Results

Status: 200 OK Time: 58 ms Size: 15.11 KB Save Response

Pretty Raw Preview Visualize XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type='text/xsl' href='/obix/xsl'?>
3 <obj href="https://acmecorp.ns24.lan/obix/histories/HtmlDemo/MainKwh/~historyQuery/" is="obix:HistoryQueryOut" xmlns="http://obix.org/ns/schema/1.0"
  xsi:schemaLocation="http://obix.org/ns/schema/1.0 /obix/xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
4   <list name="data" of="#RecordDef obix:HistoryRecord">
5     <obj>
6       <abstime name="timestamp" val="2024-03-12T00:00:00.000-04:00" tz="America/New_York"/>
7       <real name="value" val="79.9000015258789"/>
8     </obj>
9     <obj>
10      <abstime name="timestamp" val="2024-03-12T00:15:00.000-04:00" tz="America/New_York"/>
11      <real name="value" val="80.0"/>
12    </obj>
13    <obj>
14      <abstime name="timestamp" val="2024-03-12T00:30:00.000-04:00" tz="America/New_York"/>
15      <real name="value" val="99.19999694824219"/>
16    </obj>
17    <obj>
18      <abstime name="timestamp" val="2024-03-12T00:45:00.000-04:00" tz="America/New_York"/>
19      <real name="value" val="85.19999694824219"/>
20    </obj>
```

Niagara Analytics GetTrend

POST

https://acmecorp.ns24.lan/na

Send

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none

form-data

x-www-form-urlencoded

raw

binary

Text

```
1 {"requests":
2   [
3     {
4       "message": "GetTrend",
5       "node": "slot:/AnalyticsPlayground/Bldg1",
6       "data": "h4:energy",
7       "timerange": "yesterday",
8       "interval": "hour"
9     }
10  ]
11 }
```

Body Cookies (1) Headers (11) Test Results

Status: 200 OK Time: 40 ms Size: 3.37 KB Save Response

Pretty

Raw

Preview

Visualize

Text

```
1 "responses" : [
2   {
3     "message": "GetTrend",
4     "node": "slot:/AnalyticsPlayground/Bldg1",
5     "data": "h4:energy",
6     "timerange": "yesterday",
7     "interval": "hour",
8     "rows": [
9       [
10        "2024-03-12T00:00:00.000-04:00",
11        "79.9000015258789",
12        "ok",
13        "{ }"
14      ],
15      [
16        "2024-03-12T01:00:00.000-04:00",
17        "89.5",
```

Json Toolkit Json Schema Exporter

GET

https://acmecorp.ns24.lan/ord/station:%7Cslot:/GoogleChartExample/EnergyData%7Cview:jsonToolkit:JsonExporter

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

Query Params

	Key	Value	Bulk Edit
	Key	Value	

body

Cookies (1)

Headers (11)

Test Results

Status: 200 OK

Time: 153 ms

Size: 19.9 KB

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1  [
2    [
3      "timestamp",
4      "kwh"
5    ],
6    [
7      "2024-03-03 12:00 AM",
8      78.8
9    ],
10   [
11     "2024-03-03 12:15 AM",
12     92
13   ],
14   [
15     "2024-03-03 12:30 AM",
16     89.2
17   ],
18   [
19     "2024-03-03 12:45 AM",
20     92.6
21   ],
22 ]
```

HTTP Client String Servlet

POST <https://acmecorp.ns24.lan/apidemo>

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded **raw** binary Text

```
1 {"pollEnergyData":"Building2"}
```

Body Cookies (1) Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1
2  "currentTime": "2024-03-13 15:28:03.772-0400",
3  "count": 5,
4  "power": 326.4,
5  "energy": 1578.9
6
```

Status: 200 OK

StringServlet (String Servlet)

Status [ok]

Fault Cause

Enabled ☒ true

Servlet Name apidemo

```
{"pollEnergyData":"Building2"}
```

Out

Last Received 13-Mar-2024 03:53 PM EDT

Csrf Protection ☐ false

Clear Between Duplicate Requests ☐ false

Response Body Response Folder

Default Response Response

Building1 Conditional Response

Enabled ☒ true

Conditions Conditions

And ☒ true

BodyContains Body Contains

Not ☐ false

Contains "pollEnergyData":"Building1"

Response Response

```
{"currentTime":"2024-03-13 15:27:54.363"
```

Data

Response Code 200 [100-599]

Building2 Conditional Response

Niagara Data Service

POST

https://www.niagara-cloud.com/api/v1/egress/telemetry

Send

Params

Authorization

Headers (11)

Body

Pre-request Script

Tests

Settings

Cookie:

Beautify

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "systemGuid": "{{systemGuid}}",
3   "cloudId": ["6ddcb4c7-59ef-403e-b0af-bef9591b1d03"],
4   "startTime": "{{startTime}}",
5   "endTime": "{{endTime}}",
6   "recordLimit": "{{totalRecords}}",
7   "includePreRecord": true,
8   "includePostRecord": true,
9   "sortAscending": true
10 }
```

body

Cookies

Headers (17)

Test Results

Status: 200 OK

Time: 1220 ms

Size: 939.6 KB

Save as example

Pretty

Raw

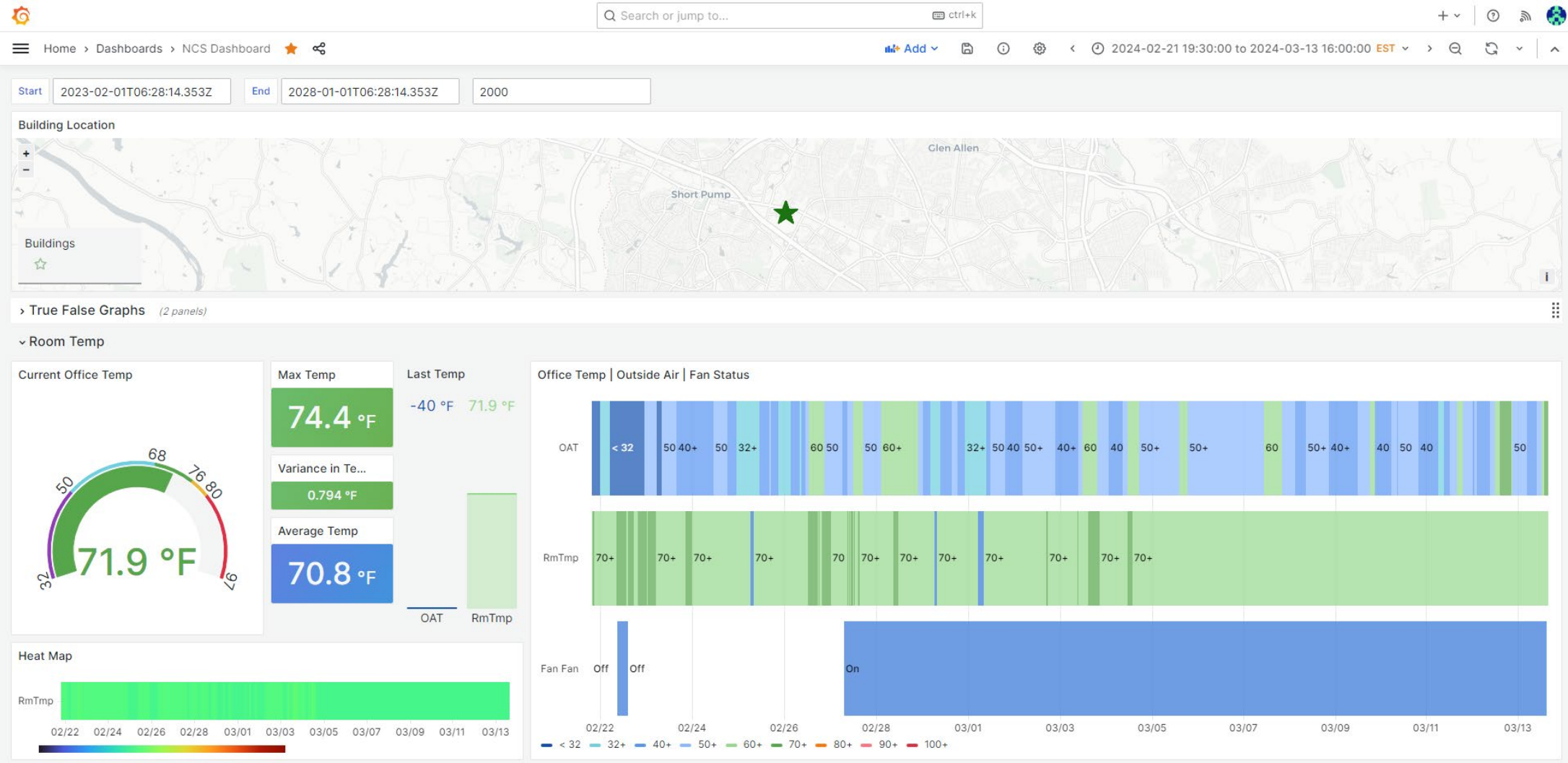
Preview

Visualize

JSON

```
1 {
2   "systemGuid": "68e471a9-7648-4f94-85a3-6df1daeaba75",
3   "recordLimit": 2000,
4   "pointDetails": [
5     {
6       "pointId": "6ddcb4c7-59ef-403e-b0af-bef9591b1d03",
7       "preRecord": {
8         "time": "2024-02-13T16:15:00.005Z",
9         "value": 71.9000015258789,
10        "properties": {"trendFlags.outOfOrder":false,"status.overridden":false,"status.fault":false,"status.ok":true,"trendFlags.modified":false,"status.null":false,"status.unackedAlarm":false,"status.disabled":false,"trendFlags.interpolated":false,"trendFlags.hidden":false,"status.alarm":false,"status.stale":false,"status.down":false,"trendFlags":"0","trendFlags.start":false,"status":"0"}
11      },
12      "postRecord": null,
13      "historyRecords": [
14        {
15          "time": "2024-02-13T16:30:00.098Z",
16          "value": 71.80000305175781,
17          "properties": {"trendFlags.outOfOrder":false,"status.overridden":false,"status.fault":false,"status.ok":true,"trendFlags.modified":false,"status.null":false,"status.unackedAlarm":false,"status.disabled":false,"trendFlags.interpolated":false,"trendFlags.hidden":false,"status.alarm":false,"status.stale":false,"status.down":false,"trendFlags":"0","trendFlags.start":false,"status":"0"}
18        },
19        {
20          "time": "2024-02-13T16:45:00.101Z",
21          "value": 71.70000457763672,
22          "properties": {"trendFlags.outOfOrder":false,"status.overridden":false,"status.fault":false,"status.ok":true,"trendFlags.modified":false,"status.null":false,"status.unackedAlarm":false,"status.disabled":false,"trendFlags.interpolated":false,"trendFlags.hidden":false,"status.alarm":false,"status.stale":false,
```

Grafana Dashboard Application Using API



Questions

