



NS2022

ACCELERATING INNOVATION

JSON Toolkit Demonstration

```
{  
  "released": true,  
  "current_version": 4.12,  
  "authors": [  
    "name": "Nick Dodd",  
    "name": "Jason Woollard",  
  ]  
}
```



When can I get hold of this?

- Available now:
 - To evaluate using Demo Licenses
 - Niagara 4.9 onwards included in images
- Production use:
 - Add DR-JSON license part
 - Not capacity based
 - Requires active SMA

Agenda

- Why might you need a JSON Toolkit?
- How to export JSON data
 - Then again, at scale
- A relatively ‘enlightening’ example
- Cloud use cases, with inbound data

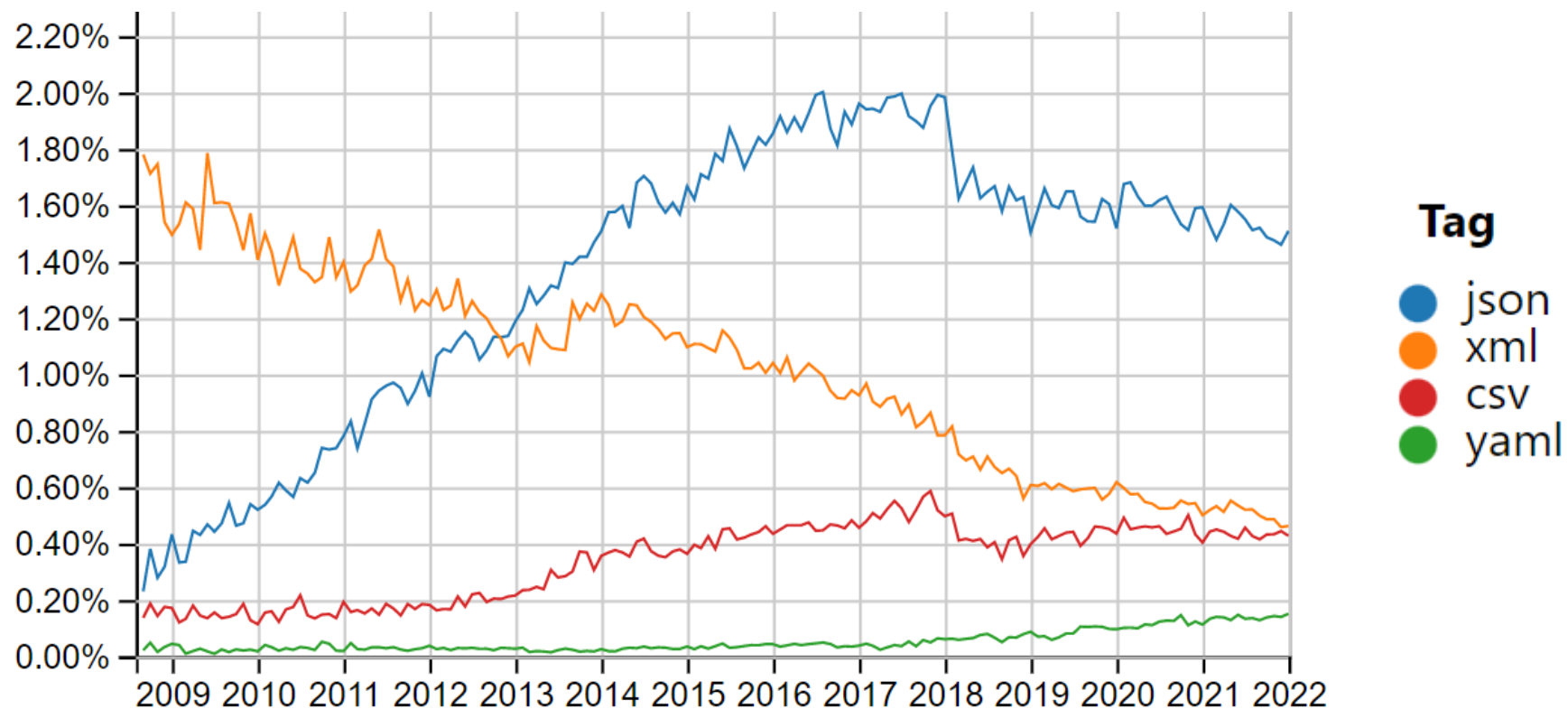
Again, What is JSON?

- “Key” : Value pairs
 - Key = String
 - Value = Numeric / Boolean / String
 - Objects { }
 - Arrays []
- Contain Key Value pairs
 - Contain lists of Values

```
{  
  "T-800": {  
    "manufacturer": "Cyberdyne",  
    "model": 101,  
    "cpu": "neural",  
    "self-aware": true  
    "requires": [  
      "clothes", "boots", "motorcycle"  
    ],  
  },  
  "T-1000": { ... }  
}
```

JSON took over the world

%
Stack
Overflow
Questions



<https://insights.stackoverflow.com/trends?tags=json%2Cxml%2Cyaml%2Ccsv>
[Accessed Feb 2022]

Where is JSON used?



UI / Charting / JavaScript



Web Service API's



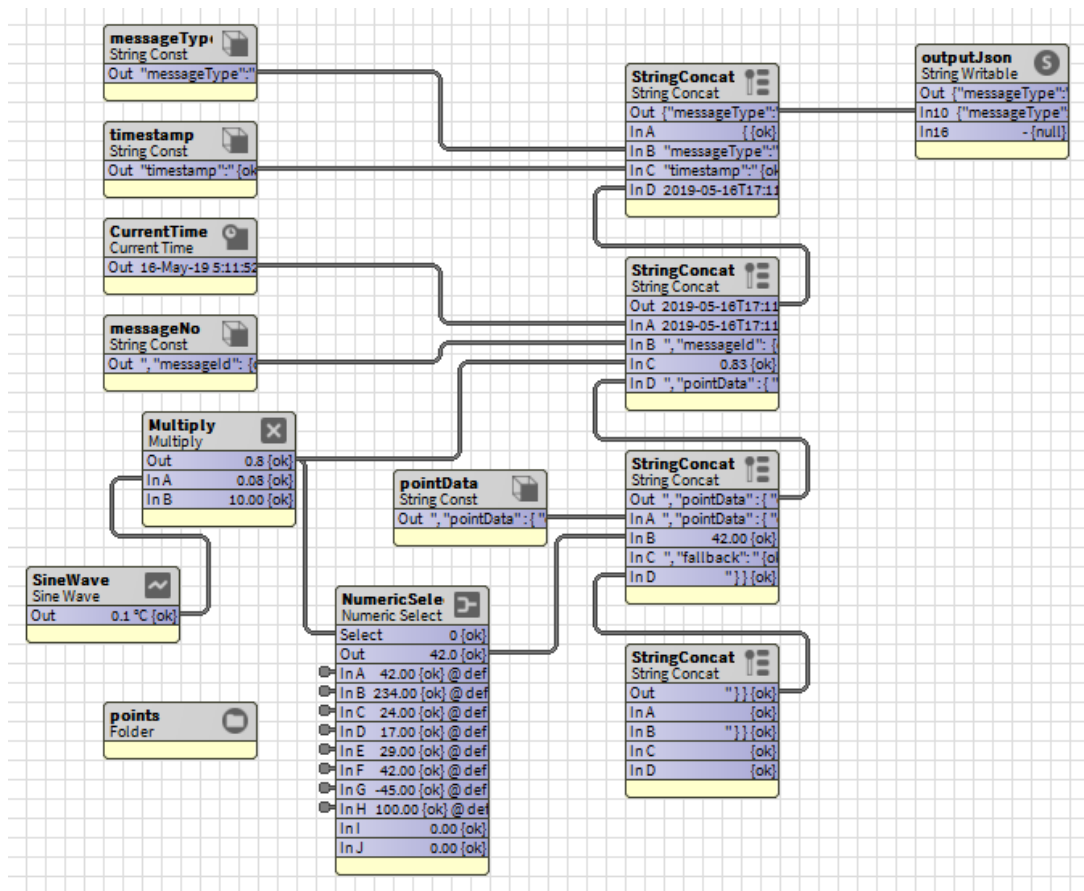
 mongoDB.

Storage - NoSQL



IoT - Sensor Data / Control / Config

The Wiresheet is not enough



The Downward Spiral:

- 1000s of points
- History Data
- Events Timestamps



Further Down The Spiral:

- Formatting Dates
- Numeric Precision
- Including Tags

Demand for a generic solution?

```
Watson
{
  "d": {
    "temp": 56.0467
  }
}
```

IAP

```
SNON 2.0
[
  {
    "messageID": "urn:uuid:15b4615a-b736-4e55-9d19-35b2ec807439",
    "messageTime": "2014-08-20T14:32:56.125Z",
    "message": {
      "id": "319e1420-fd4e-49e9-9d43-a24b2cf98486",
      "time": "2014-08-20T14:32:56.125Z"
    }
  }
]
```

```
SenML
[
  {"bn": "urn:dev:ow:10e2073a0108006:", "bt": "v", "bu": "A", "bver": 5, "n": "voltage", "u": "V", "v": 120.1},
  {"n": "current", "t": -5, "v": 1.2},
  {"n": "current", "t": -4, "v": 1.3},
  {"n": "current", "t": -3, "v": 1.4},
  {"n": "current", "t": -2, "v": 1.5},
  {"n": "current", "t": -1, "v": 1.6},
  {"n": "current", "v": 1.7}
]
```

Many APIs!
// Insert *anything* here ?

```
Sparkplug-A DData
{
  "timestamp": 1555602837059,
  "metric": {
    "seq": 0,
    "jsonExample/Points/Number/out": 82.6
  }
}
```

UDMI

```
Sparkplug-A Birth
{
  "timestamp": 1555599445478,
  "metric": {
    "seq": 0,
    "bdseq": 7,
    "node Control/Reboot": true,
    "node Control/Rebirth": false,
    "node Control/Next Server": false,
    "node Control/Scan Rate": 3000,
    "properties/Hardware Make": "Tridium",
    "properties/Hardware Model": "JACE-8000"
  }
}
```

JSON Schema 101

(Lots more docs and tutorials also available)



JSON Schema basics: Bindings

The screenshot shows the jsonToolkit interface. On the left is a tree view of schema types, with 'Object' and 'BoundObject' highlighted. On the right, a diagram shows a 'Ramp' object connected to a 'WindSpeed' object. The 'Ramp' object has an 'Out' property with the value '45.4 [ok]'. The 'WindSpeed' object has 'In10' and 'In16' properties, both with the value '45.4 [ok]'. A line connects the 'Out' property of 'Ramp' to the 'In10' property of 'WindSpeed'.

Object	Property	Value
Ramp	Ramp	
	Out	45.4 [ok]
WindSpeed	Out	45.4 [ok] @ 10
	In10	45.4 [ok]
	In16	- [null]
	WindSpeed	

JSON Schema basics

{ } BasicBindings (Json Schema)

Output

```
{  "stationName": "ns22",  "myApiVersion": 3.1415,  "messageId": 49,  "timestamp": "2022-03-02 17:04:38.115+0000",  "whatIsJson": "json is a lightweight data-interchange format. It is easy for humans to read and",  "numberWithHistory": {    "out": 73.02,    "in10": 73.02,    "in16": 0,    "nestedString": "Properties, Arrays and Objects may be nested within other Objects or Arrays!"  },  "selectedSlots": [
```

Generate
Copy
Clear Output
Output History
Metrics
Indented Display

Enabled true

Status {ok}

Fault Cause

Last Updated 02-Mar-2022 05:04 PM GMT

Config Json Schema Config Folder

Queries Json Schema Query Folder

{ } root Json Schema Object

- { } stationName Json Schema String Property
- { } myApiVersion Json Schema Numeric Property
- { } messageId Json Schema Count Property
- 🕒 timestamp Json Schema Current Time Property
- {*} whatIsJson Json Schema Bound Property
- {*} multipleSlots Json Schema Bound Object
- {*} Selected Slots Json Schema Bound Array

Binding with 'Regular' Schema

{ } JsonSchema (Json Schema)

Output

```
{  
  "messageNo" : 1,  
  "windSpeed" : 34  
}
```

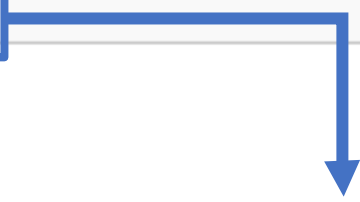
▼ { } root Json Schema Object

▶ { } messageCount Json Schema Count Property

▼ [*] sensorData Json Schema Bound Property

Binding

slot:/SchemaExample/WindSpeed/out



Absolute Ord - resolves against **base station**:

Scaling with 'Relative' Schema

The screenshot shows a software interface with the following components:

- RelativeJsonSchema**: A text editor containing a JSON schema:

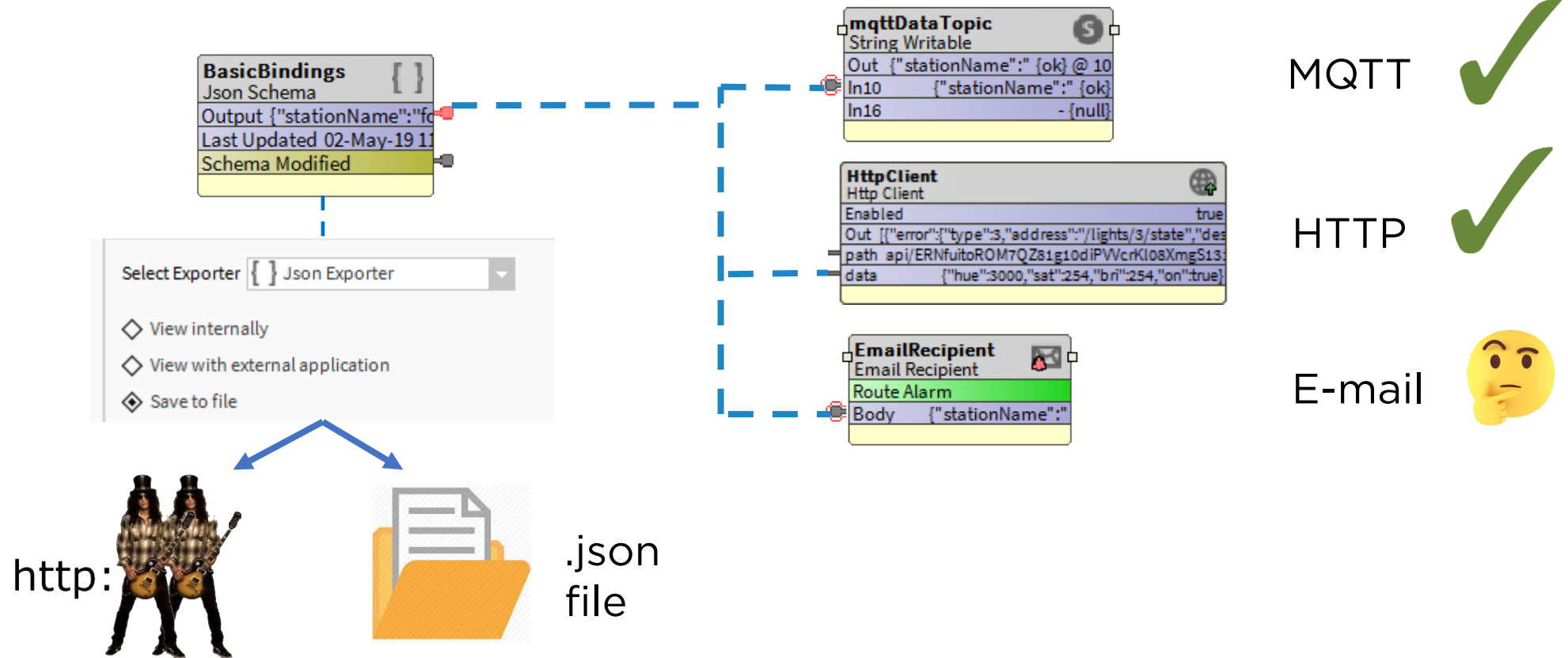
```
{  "messageNo" : 1,  "sensor1" : 1}
```
- Base Query**: A text field containing the query `station: | slot: / | neql: n: point`.
- Json Schema Object**: A tree view showing the schema structure:
 - root (Json Schema Object)
 - messageNo (Json Schema Count Property)
 - sensorData (Json Schema Bound Property)
- Binding**: A text field containing `slot: out`.

Arrows from the **Base Query** and **Binding** fields point to three resolved paths:

- slot:/path/sensor1
- slot:/path/sensor2
- slot:/path/sensor3

Relative Ord – resolves against base:

Transport Agnostic



Device Connectivity

Example Use Case 1



Device Connectivity

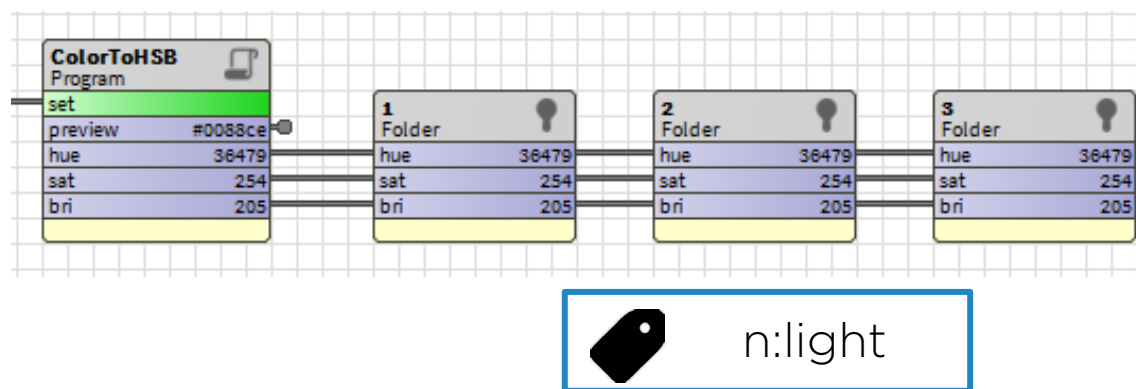
RelativeJsonSchema	[A]
Relative Json Schema	
Output {"hue":10068,"sat":254,"bri":254,"on":true}	
Last Updated	02-May-19 11:42 AM BST
Schema Modified	
Current Base And Output	



```
{  
  "hue": 10268,  
  "sat": 254,  
  "bri": 254,  
  "on": true  
}
```



Enlightening Example



RelativeJsonSchema (Relative Json Schema)

```
{  
  "hue": 10068,  
  "sat": 254,  
  "bri": 254,  
  "on": true  
}
```

Output

Base Query

slot:/Hue|neql:n:light

hueValues Json Schema Bound Object

Binding

slot:

Json Name Source

Display Name

Selected Slots

Slots To Include

- hue ✕
- sat ✕
- bri ✕
- on ✕

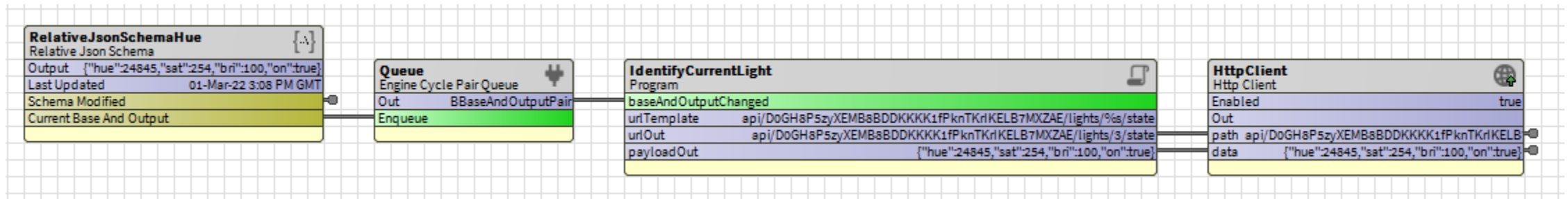
Linking to a Transport

JSON

Queue

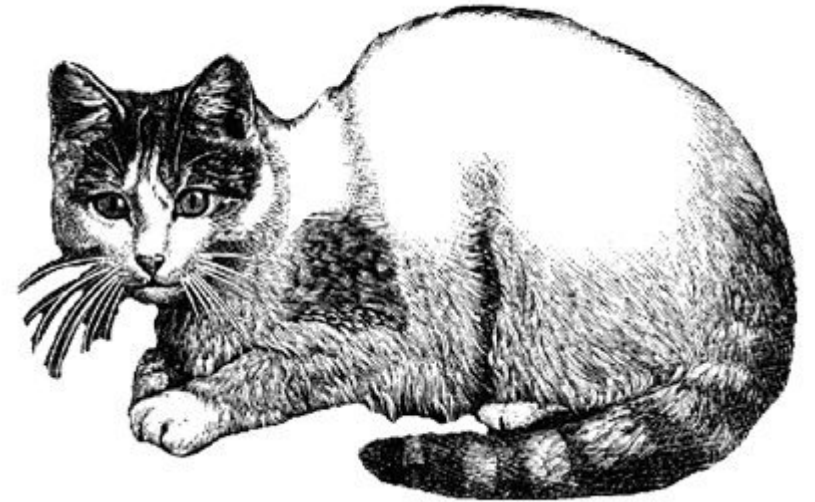
Change URL

Send



	BASE ITEM	MESSAGE	Update URL Path from base item	
<pre>{ "hue": 10068, "sat": 254, "bri": 254, "on": true }</pre>	Folder '1'	<code>{"hue":10068,"sat": 254,"bri": 254,"on": true}</code>	<code>api/lights/%s/state</code>	<pre>{ "hue": 10068, "sat": 254, "bri": 254, "on": true }</pre>
	Folder '2'	<code>{"hue":10068,"sat": 254,"bri": 254,"on": true}</code>	<code>api/lights/1/state</code>	
	Folder '3'	<code>{"hue":10068,"sat": 254,"bri": 254,"on": true}</code>		

Stop!
Demo time



Expert

Clicking Until
Stuff Happens

Visualization

Example Use Case 2



Queries

▼	Queries	Json Schema Query Folder
	Query Interval	000000h 01m 00s [0ms-+inf]
	Last Query Completed Timestamp	08-Jun-2019 03:31 PM BST
▼	{q} TransformQuery	Json Schema Query
	Query Ord	station: transform: slot: /VelocityServlet/hue/TransformGraph
	Last Result Size	5
▼	{q} BoundQueryResult	Json Schema Bound Query Result
	Query	TransformQuery
	Output Style	Row Array With Header Preview

Queries

{q} BoundQueryResult (Json Schema Bound Query Result)

Query TransformQuery

Output Style Row Array With Header Preview

- Key Value Pair Object
- Single Column Array
- Row Array With Header
- Row Array
- Objects Array
- Named Objects
- Column Array With Header
- Column Array

```
[  
  [  
    "Timestamp", "Light 1",  
    "Light 2", "Average"  
  ],  
  [  
    "55", 50.80,  
    43.70, 47.25  
  ],  
  [  
    "00", 49.10,  
    56.52, 52.13  
  ]  
]
```

Injecting schema output to HTML

```
<html>
  <script>
    ...
    function drawChart() {
      var data = google.visualization.arrayToDataTable(
        $schema.output // <-- this is the important bit, injects data from schema!
      );

      var chart = new google.visualization.LineChart(document.getElementById('chart'));
      chart.draw(data, options); }
    </script>

  <body>
    <div id="chart" style="width: 1040px; height: 780px"></div>
  </body>
</html>
```

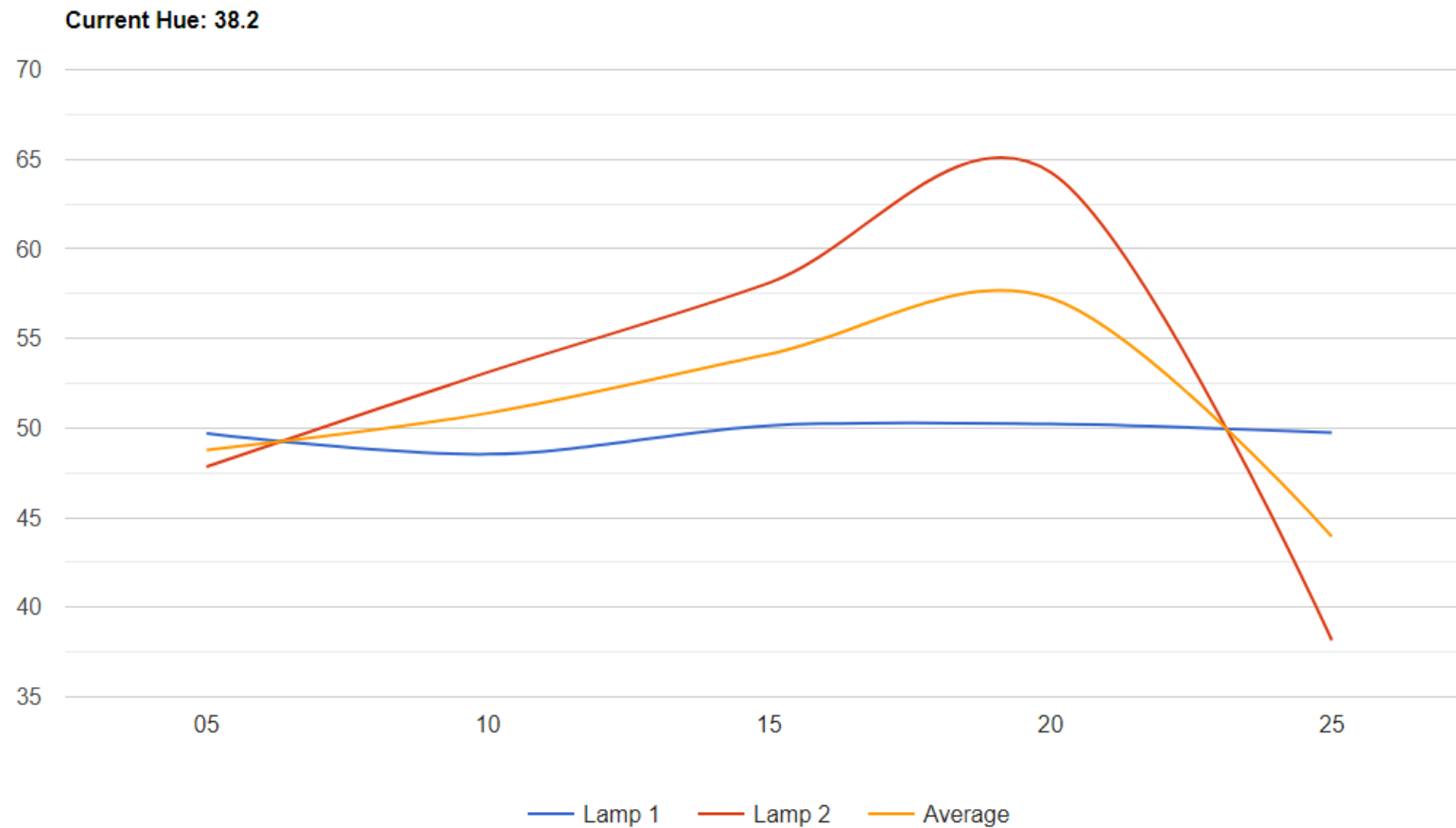

Injecting schema output to HTML

```
<html>
  <script>
    ...
    function drawChart() {
      var data = google.visualization.arrayToDataTable(
        $schema.output // <-- this is the important bit, injects data from schema!
      );

      var chart = new google.visualization.LineChart(document.getElementById('chart'));
      chart.draw(data, options); }
    </script>

  <body>
    <div id="chart" style="width: 1040px; height: 780px"></div>
  </body>
</html>
```

Viewing the result



jsonToolkit Concept Demonstration:

1. Bindings:

Brightness: 49.0 [ok] @ def

2. Inline Writer

Inline Writer: #40ffff

3. Type Override

Enabled

Save

Generate Json

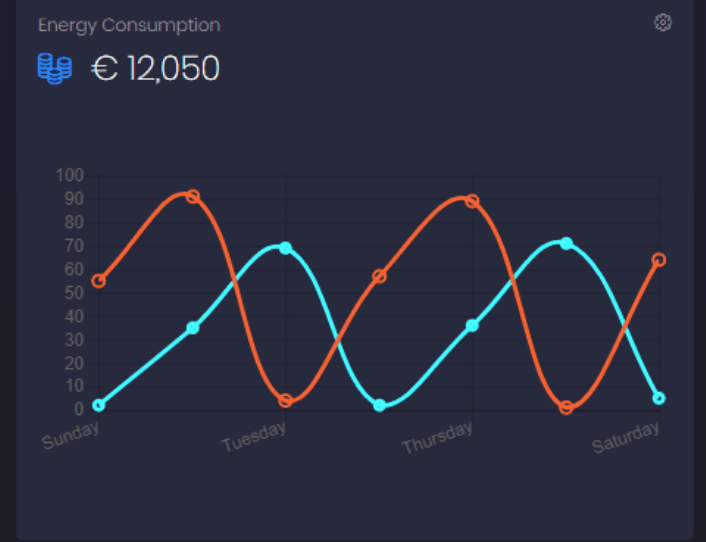
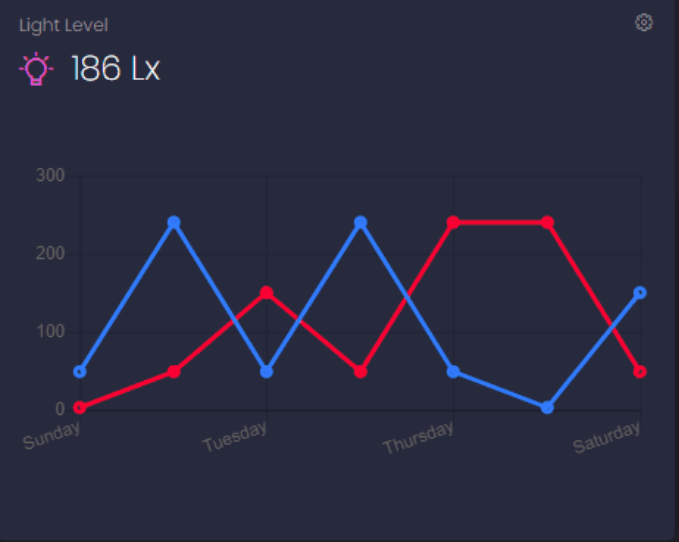
GROUND FLOOR OVERVIEW

JSON SITE NAVIGATION

- DASHBOARD
- WORK ORDERS
- CUSTOMERS
- REPORTS
- INTEGRATION

SAVED REPORTS

LAST MONTH



Ground Floor AHU ⚙️ [Inputs]

POINT	STATUS	HISTORY	VALUE
Outside Air Temperature	OK		3.2 °C
Frost Coil Temperature	OK		11.4 °C
Cooling Coil Temperature	FAULT		11.6 °C
Supply Air Temperature	OK		24.3 °C
Return Air Temperature	OK		24.7 °C

Alarms 3 ⚙️ [Previous 24 Hours]

- Zone Air Performance 0-50%**
Zone served by AHU2
- Clean Filter**
Supply Air filter dirty for AHU2
- Simultaneous Heating & Cooling**
Potential Energy Wastage: The heating and cooling coils on AHU2 are operating simultaneously, please investigate.

UX Ready

 Output

```
{
  "stationName": "dodd",
  "myApiVersion": 3.1415,
  "messageId": 2739,
  "timestamp": "2019-06-06 12:06:40.040+0100",
  "whatIsJson": "json is a lightweight data-interchange format. It is ea
  "numberWithHistory": {
    "out": 4.58,
    "in10": 4.58,
    "in16": 0,
    "nestedString": "Properties, Arrays and Objects may be nested within
  },
  "selectedSlots": [
    4.58,
    0
  ]
}
```

 Generate

 Clear Output

 Metrics

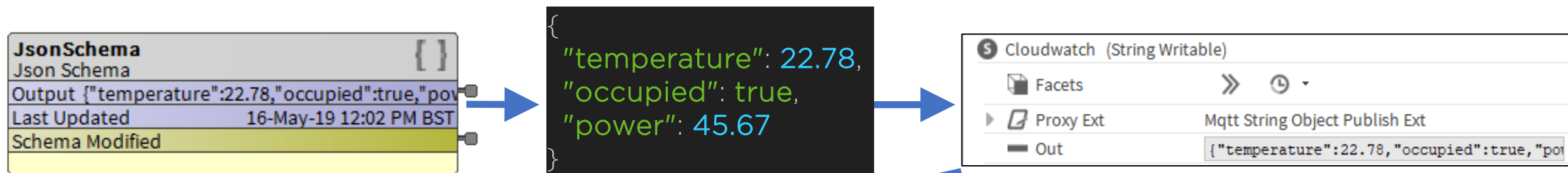
 Indented Display

Take me to the clouds above

Example Use Case 3



AWS: Cloudwatch Charting



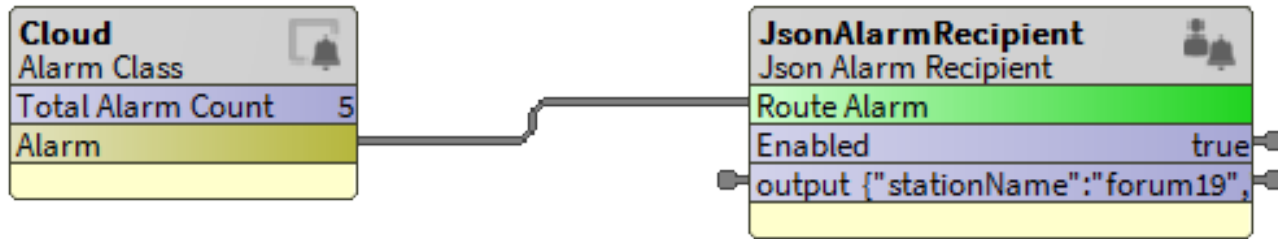

building/cloudwatch


Function


Cloudwatch

Launch!

JSON Alarm Recipient



JsonAlarmRecipient (Json Alarm Recipient)

- Time Range: 12:00 AM - 12:00 AM
- Days Of Week: Sun Mon Tue Wed Thu Fri Sat
- Transitions: toOffnormal toFault toNormal toAlert
- Route Acks: true
- Enabled: true
- Publish Point: null Select Source
- Json Schema: Json Schema

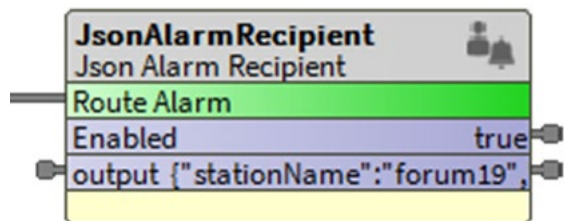
output

```
{
  "stationName": "forum19",
  "timestamp": "2019-05-02 13:52:45.224+0100",
  "uuid": "d318e44d-14e3-4fb6-ab66-a3aa4e415f4d",
  "alarmValue": 223.2487,
  "source": "local:|station:|slot:/Services/AlarmService/NumericWritable/OutOfRangeAlarmExt",
  "sourceState": "Offnormal",
  "offnormalValue": "223.2",
  "highLimit": "100.0",
  "lowLimit": "10.0"
}
```

- Generate
- Copy
- Clear Output
- Output History
- Metrics
- Indented Display

Json Schema Object	
{ } root	
{ } stationName	B Format String
{ } timestamp	Json Schema Alarm Record Property
Alarm Property	timestamp
{ } uuid	Json Schema Alarm Record Property
{ } alarmValue	Json Schema Alarm Record Property
{ } source	Json Schema Alarm Record Property
{ } sourceState	Json Schema Alarm Record Property
{ } offnormalValue	Json Schema Alarm Record Property
{ } highLimit	Json Schema Alarm Record Property

Alarm Demo - Route Outbound



MQTT



building/alarm



Function



DynamoDB

```
{  
  "stationName": "summit22",  
  "timestamp": "2022-05-16 12:24:50.002+0100",  
  "uuid": "3842f3f1-21b6-4...",  
  "alarmValue": 200,  
  "source": "local:|station:|slot:.....",  
  "sourceState": "Offnormal",  
  "offnormalValue": "200.0",  
  "highLimit": "100.0",  
  "lowLimit": "10.0"  
}
```

180 - forum19 Offnormal
Time: 2019-05-16 12:24:50.002+0100
Low Limit: 10.0
High Limit: 100.0
Breaching Value: 200.0
UUID: 291c3dc7-2b18-492c-8118-dde9ed745be4

176 - forum19 Offnormal
Time: 2019-05-15 14:40:30.014+0100
Low Limit: 10.0
High Limit: 100.0
Breaching Value: 200.0
UUID: 3842f3f1-21b6-4eb8-a6b6-bca1f30c8a48



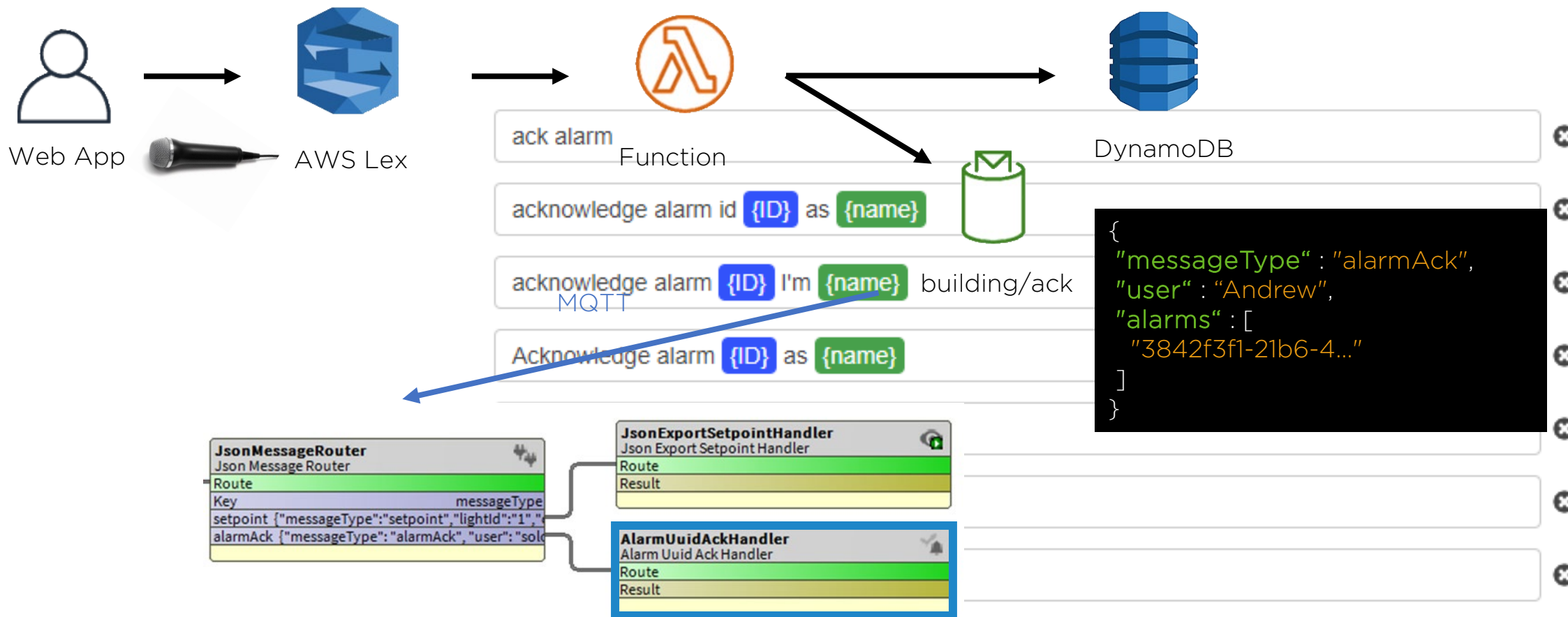
Web App
User

[Launch!](#)

Importing JSON

- ▼ Inbound
 - ▼ Routers
 - ▶ 🚦 JsonMessageRouter
 - ▶ 🚦 JsonDemuxRouter
 - ▼ Selectors
 - ▶ 🚦 JsonPath
 - ▶ 🚦 JsonAtIndex
 - ▶ 🚦 JsonContainsKey
 - ▶ 🚦 JsonIndexOf
 - ▶ 🚦 JsonSum
 - ▶ 🚦 JsonLength
 - ▶ 🚦 JsonFindAll
 - ▶ 🚦 JsonArrayForEach
 - ▼ Handlers
 - ▶ 🚦 AlarmUuidAckHandler
 - ▶ 🚦 SetPointHandler

Alarm Demo - Ack Inbound



Launch!

Demo: Change Point Values



AWS
Text & Object
Rekognition



Web App



building/lighting

```
{  
  "messageType": "setpoint",  
  "lightId": "1",  
  "colour": "red"  
}
```

MQTT

JsonMessageRouter
Json Message Router

Route
Key
messageType
setpoint {"messageType":"setpoint","lightId":"1",
alarmAck {"messageType":"alarmAck","user":"sol

JsonExportSetpointHandler
Json Export Setpoint Handler

Route
Result

AlarmUoidAckHandler
Alarm Uoid Ack Handler

Route
Result

[Launch!](#)

ColourInput (String Writable)

Facets	
Proxy Ext	null
Out	blue {ok} @ 10
In1	- {null}
ExportMarker	Json Export Marker
Id	1
Platform Writable	<input checked="" type="checkbox"/> true

JSON Path

JsonPath	
Json Path	
Route	
Out	["Vacant"]
Path	\$.housekeepingRooms.room[?(@.room

```
{
  "totalResults":31,
  "offset":20,
  "totalPages":2,
  "limit":20,
  "hasMore":false,
  "housekeepingRooms":{
    "hotelId":"FawltyTowers",
    "room":[
      {
        "smokingPreference":"NS",
        "floor":"02",
        "housekeeping":{
          "roomPersons":{
            "houseKeepingPersons":0,
            "frontOfficePersons":0
          },
          "housekeepingRoomStatus":{
            "frontOfficeStatus":"Vacant",
            "housekeepingRoomStatus":"Inspected",
            "housekeepingStatus":"Vacant",
            "reservationStatusList":[
              "Departed"
            ]
          }
        },
        "roomType":{
          "roomType":"KING",
          "pseudoRoom":false,
          "roomClass":"ALL"
        },
        "roomId":"11"
      },
      {
        "smokingPreference":"NS",
```

Oracle Hospitality Web Service

Hotel Occupancy Data in < 15 mins

Our existing “Micros” driver for this same task took months to develop

`$.housekeepingRooms.room[?(@.roomId='11')].housekeeping.housekeepingRoomStatus.frontOfficeStatus`

Developer Shout Out

- Docs and videos show how to customise JSON Toolkit:
 - Type Overrides - Defines BValue mapping to JSON
 - Custom query styles - Cowsay... BACnet/Bovine :)
 - Inline JSON Writer - Freestyle

Thank-you!

Any Questions?

Also welcome any time via
supportemea@tridium.com

