NF 23

CONNECTING THE WORLD

Niagara Data + the Cloud

A Choice of Capabilities

TRIDIUM

NF 23

CONNECTING
THE WORLD

History Archive Provider

Tony Hardwick

Tridium

Terminology

- Example Relational Databases:
  - MySQL
  - SQL Server
  - Oracle DB
- History Provider

Niagara History Database

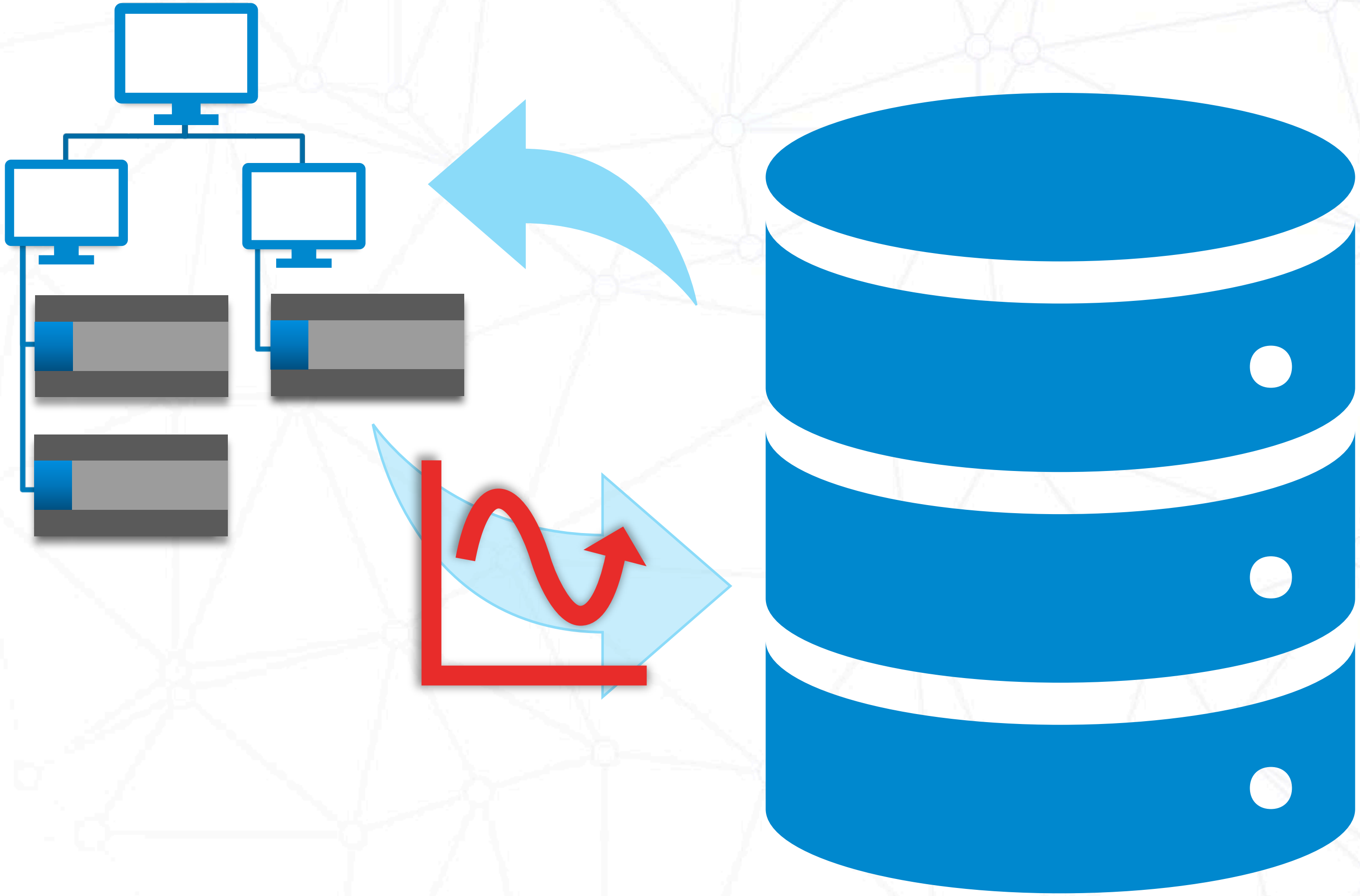
niagara4
supervisor

- Local storage limitations
  - File size limits
  - Total capacity
- Proprietary

Relational database benefits

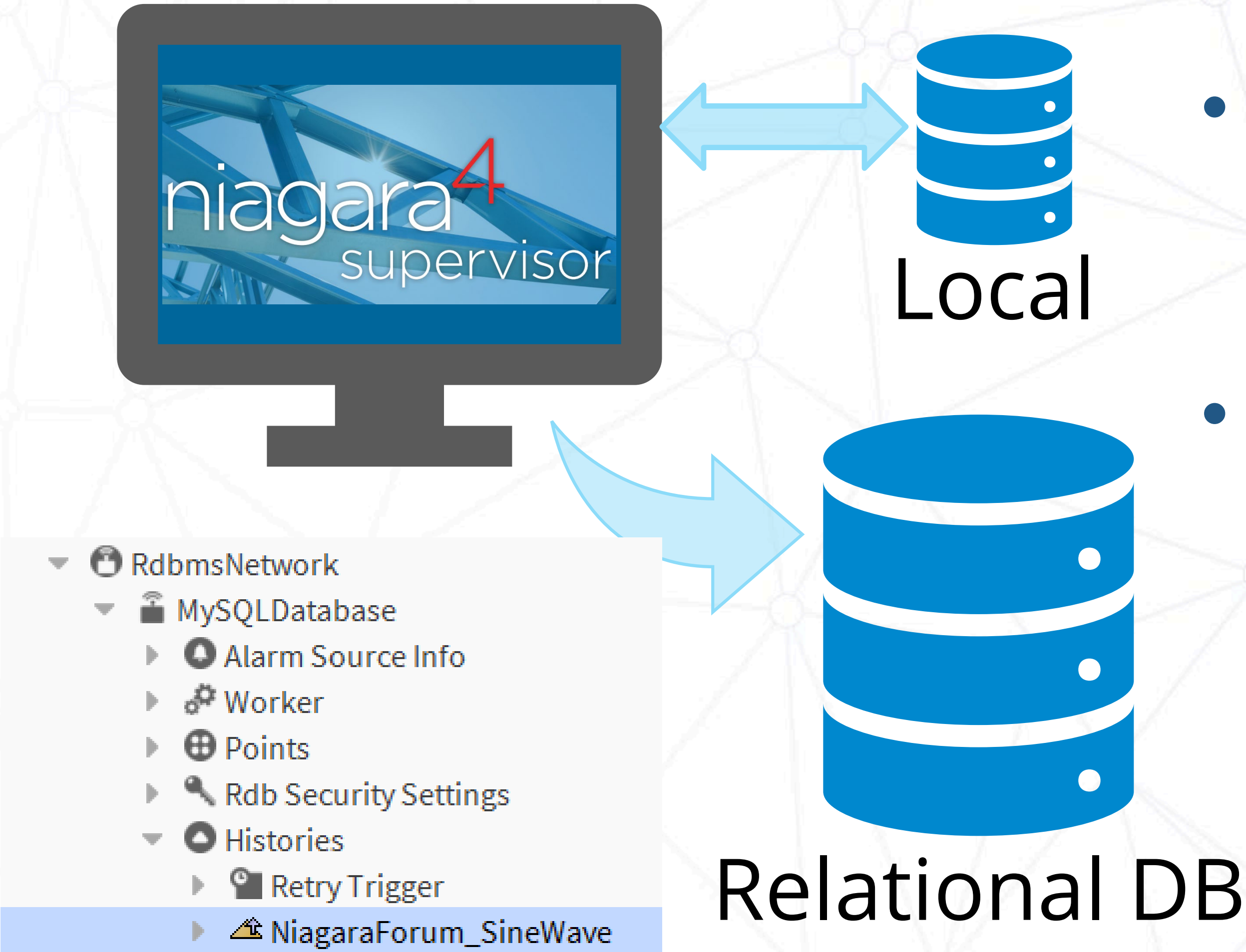
- Space!
- More options:
  - Backup
  - Data management
  - Presentation
  - Analysis
- Can be anywhere

You may remember me from such films…



- RDBMS Introduced in AX
- Primarily export
- Niagara can now access that data

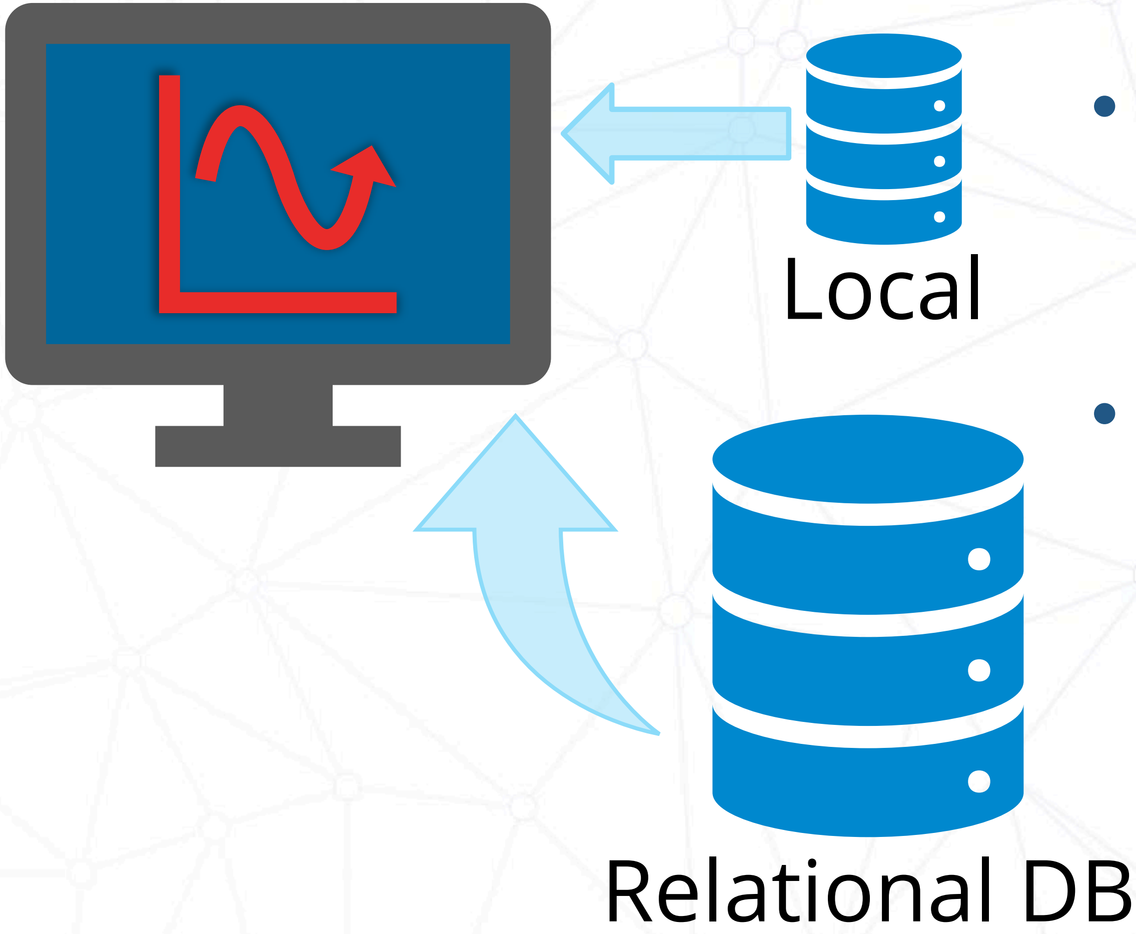
RDBMS Network



- RdbmsNetwork
  - MySQLDatabase
    - Alarm Source Info
    - Worker
    - Points
    - Rdb Security Settings
    - Histories
      - Retry Trigger
      - NiagaraForum_SineWave

- Local storage is used as normal
- Histories Exported to RDB as configured in the driver

Archive history provider



- Local store is loaded for recent records
- RDB accessed transparently if more records are required

What do I need?

- A supported relational database, e.g.:
  - MySQL
  - SQL Server
  - Oracle DB
- Rdbms module
- Rdbms<Database> module
- Licensed

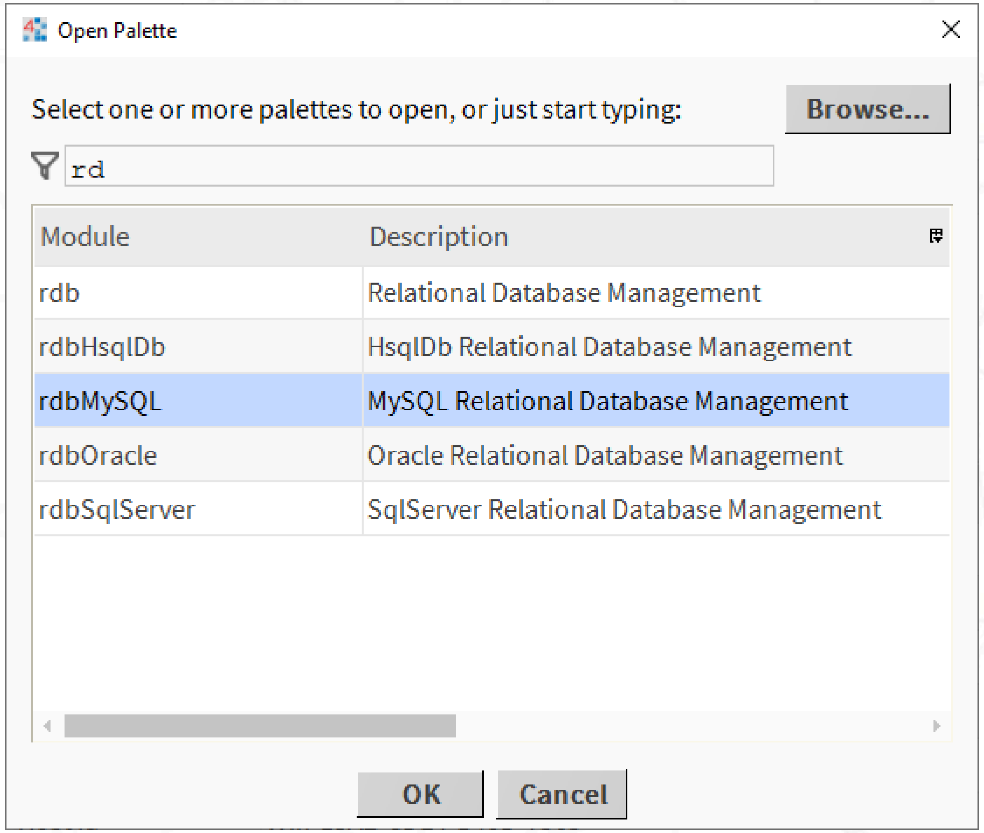
Configuration

Example installation

General setup

- RdbmsNetwork
  - Database device
- History service

Add your database



Open Palette

Select one or more palettes to open, or just start typing:

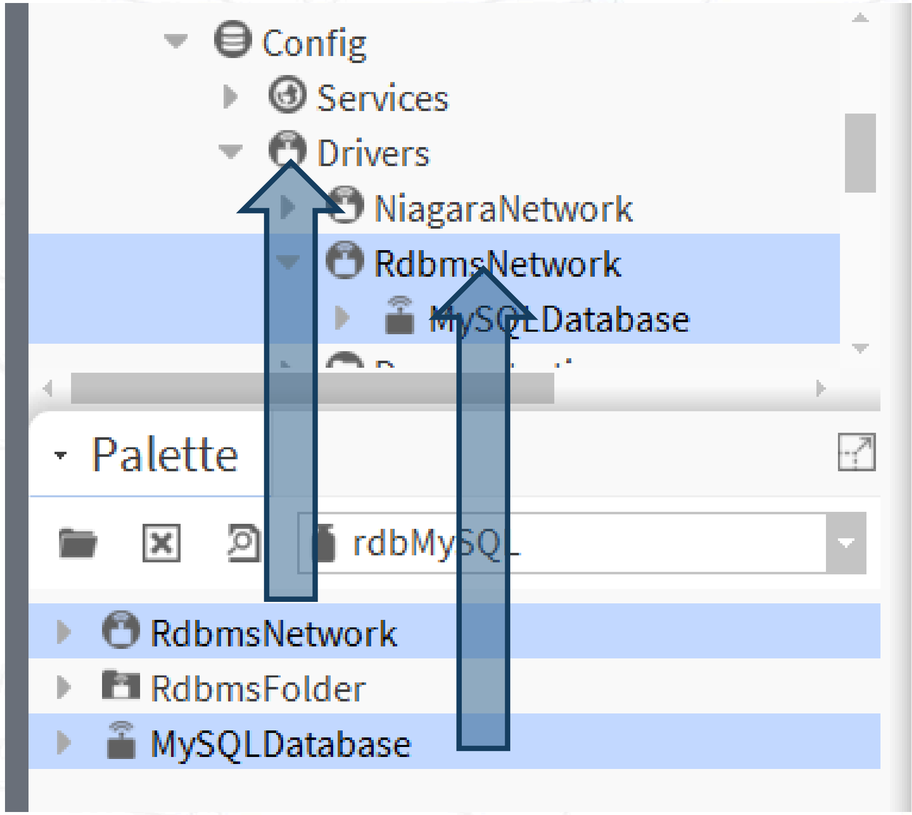
rd

Module	Description
rdb	Relational Database Management
rdbHsqlDb	HsqlDb Relational Database Management
rdbMySQL	MySQL Relational Database Management
rdbOracle	Oracle Relational Database Management
rdbSqlServer	SqlServer Relational Database Management

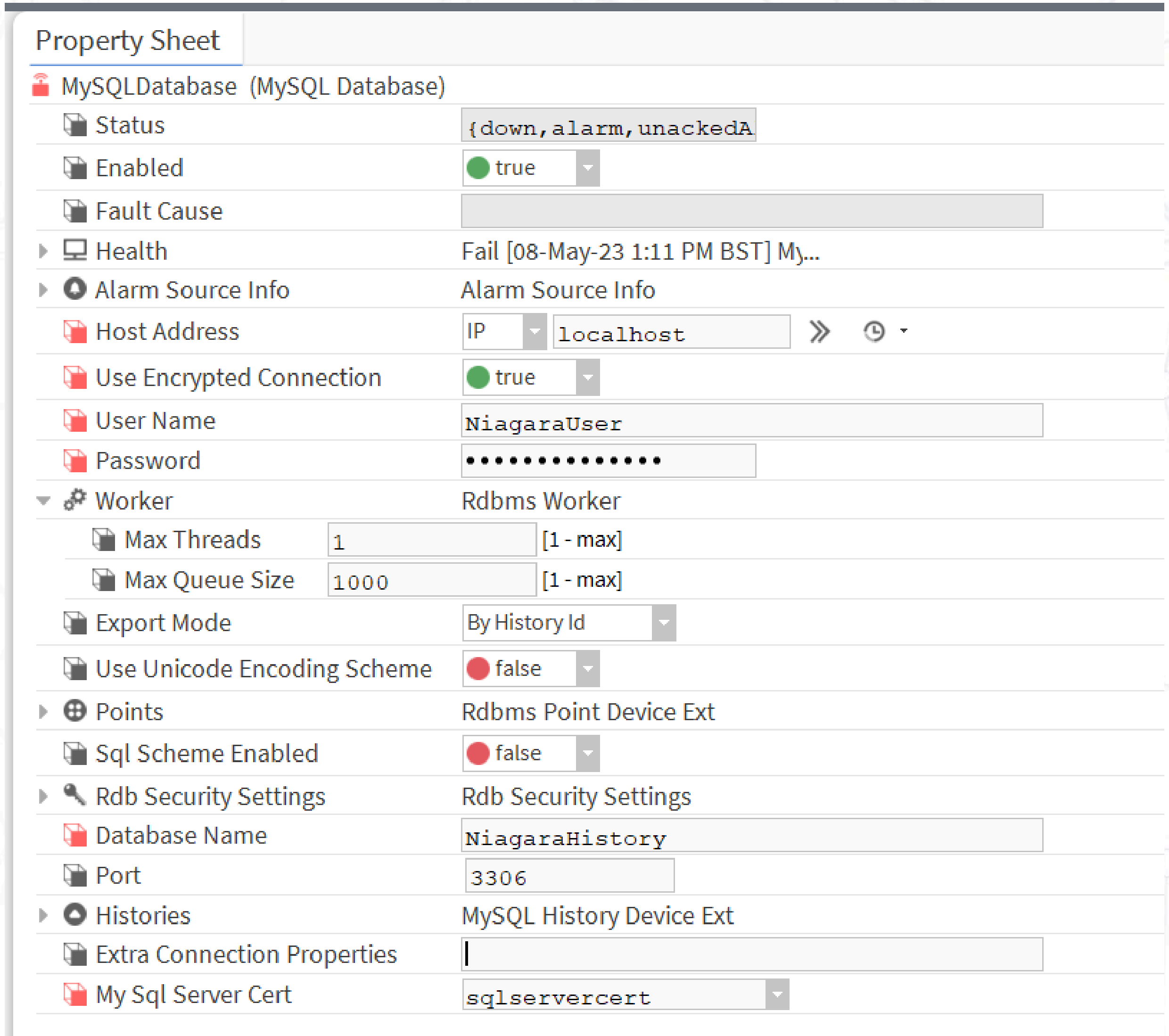
Choose your palette

Add RdbmsNetwork

Add Database



Configure Database connection

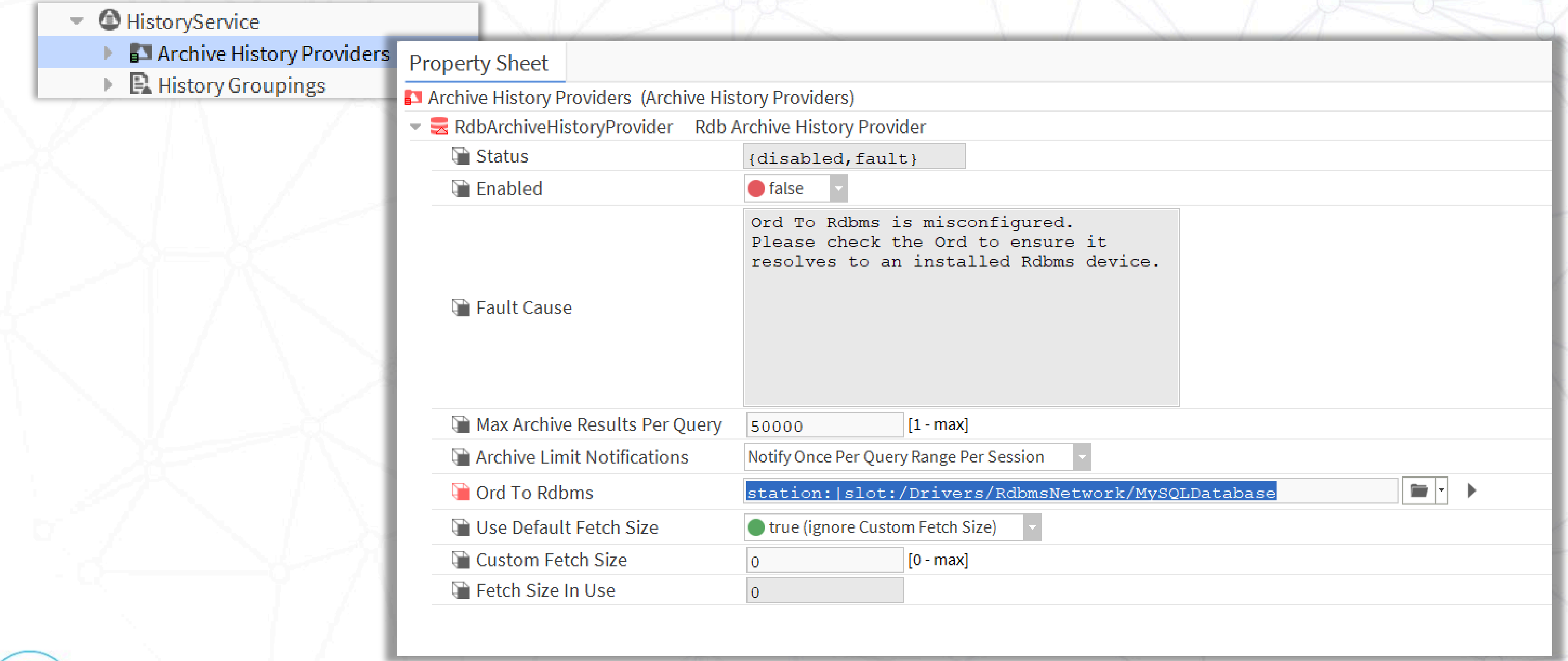


Property Sheet

MySQLDatabase (MySQL Database)

Status	{down,alarm,unackedA	
Enabled	true	
Fault Cause		
Health	Fail [08-May-23 1:11 PM BST] My...	
Alarm Source Info	Alarm Source Info	
Host Address	IP localhost	
Use Encrypted Connection	true	
User Name	NiagaraUser	
Password	••••••••••••••	
Worker	Rdbms Worker	
Max Threads	1	[1 - max]
Max Queue Size	1000	[1 - max]
Export Mode	By History Id	
Use Unicode Encoding Scheme	false	
Points	Rdbms Point Device Ext	
Sql Scheme Enabled	false	
Rdb Security Settings	Rdb Security Settings	
Database Name	NiagaraHistory	
Port	3306	
Histories	MySQL History Device Ext	
Extra Connection Properties		
My Sql Server Cert	sqlservercert	

Set up provider



- HistoryService
  - Archive History Providers
  - History Groupings

Property Sheet

Archive History Providers (Archive History Providers)

RdbArchiveHistoryProvider Rdb Archive History Provider

Status	{disabled,fault}
Enabled	● false
Fault Cause	Ord To Rdbms is misconfigured. Please check the Ord to ensure it resolves to an installed Rdbms device.
Max Archive Results Per Query	50000 [1 - max]
Archive Limit Notifications	Notify Once Per Query Range Per Session
Ord To Rdbms	station:|slot:/Drivers/RdbmsNetwork/MySQLDatabase
Use Default Fetch Size	● true (ignore Custom Fetch Size)
Custom Fetch Size	0 [0 - max]
Fetch Size In Use	0

Summary

NF 23

CONNECTING THE WORLD

Cloud/API Sourced Data Ingress

Curtis McKerlie

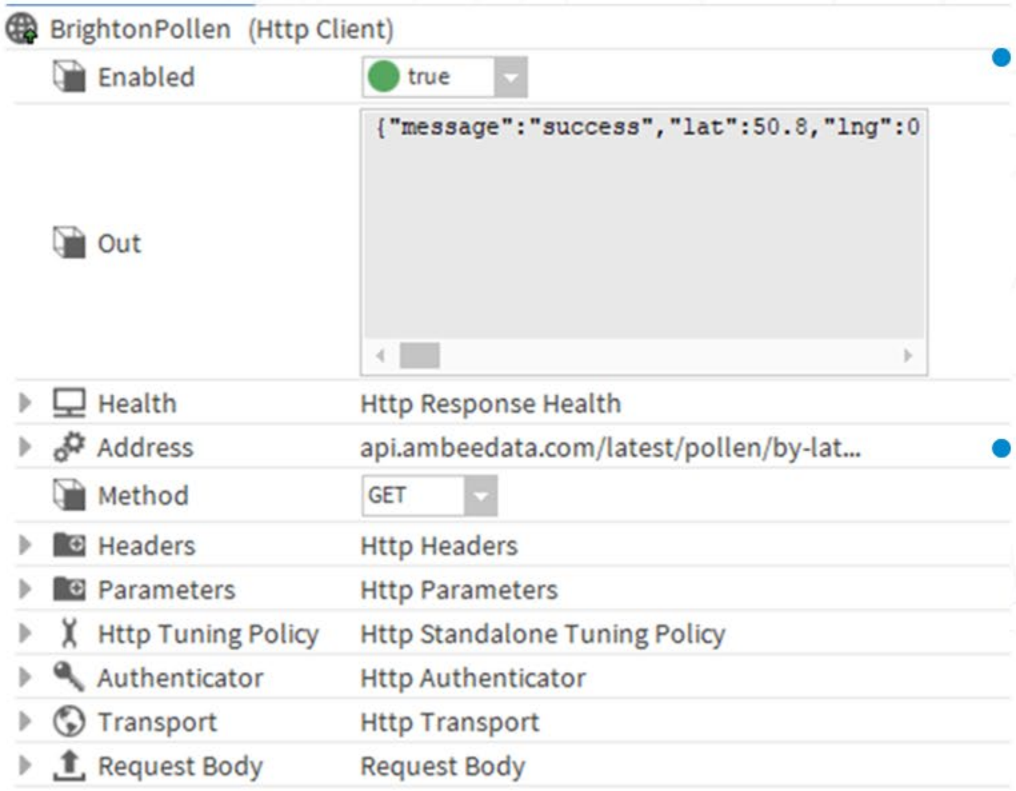
Tridium

Firstly a big thank you to,

AIRTHINGS
FOR BUSINESS

Optimised

The HTTP Client



BrightonPollen (Http Client)

Enabled	● true
Out	{"message":"success","lat":50.8,"lng":0
Health	Http Response Health
Address	api.ambeedata.com/latest/pollen/by-lat...
Method	GET
Headers	Http Headers
Parameters	Http Parameters
Http Tuning Policy	Http Standalone Tuning Policy
Authenticator	Http Authenticator
Transport	Http Transport
Request Body	Request Body

- The Niagara HTTP Client module allows Niagara Stations to interact with HTTP web services and APIs

- This allows data to be exchanged both in and out of a Niagara station

Terminology

- Web Services are online services which allow systems to exchange data, often in JSON or XML format, using HTTP as the protocol

- HTTP is also the protocol by which web pages are retrieved in web browsers. This allows data to be exchanged both in and out of a Niagara station

Terminology

- APIs are the interface contract between the client and server of the web service, defining;
  - which operations are available
  - the data format for exchange
  - the parameters which can be used
- RESTful web services/APIs obey a common set of rules and constraints to conform with modern architecture of scalability

Component Definitions

- HttpClient Component
  - This component is a standalone client, which you may use for making individual connections to single endpoints. You may use any type of request (GET/POST/PUT) with several configurations, such as parameters, headers and message body.

- HttpClient Network
  - This component offers the same functionality as a standalone client with the addition of several related endpoints. These endpoints serve as child StringPoint components with configurable proxy extensions per request. Each request can have a different address and a different set of parameters, headers and message body.

Component Definitions

- WebsocketClient Component
  - This component has similar functionality to the standalone http client component. A WebSocket is a persistent connection to an endpoint allowing full-duplex communications, where either the client or server side sends a message at any time.

- StringServlet Component
  - This component captures requests coming into the station from an external client allowing egress of data to 3rd party applications.

Use Cases

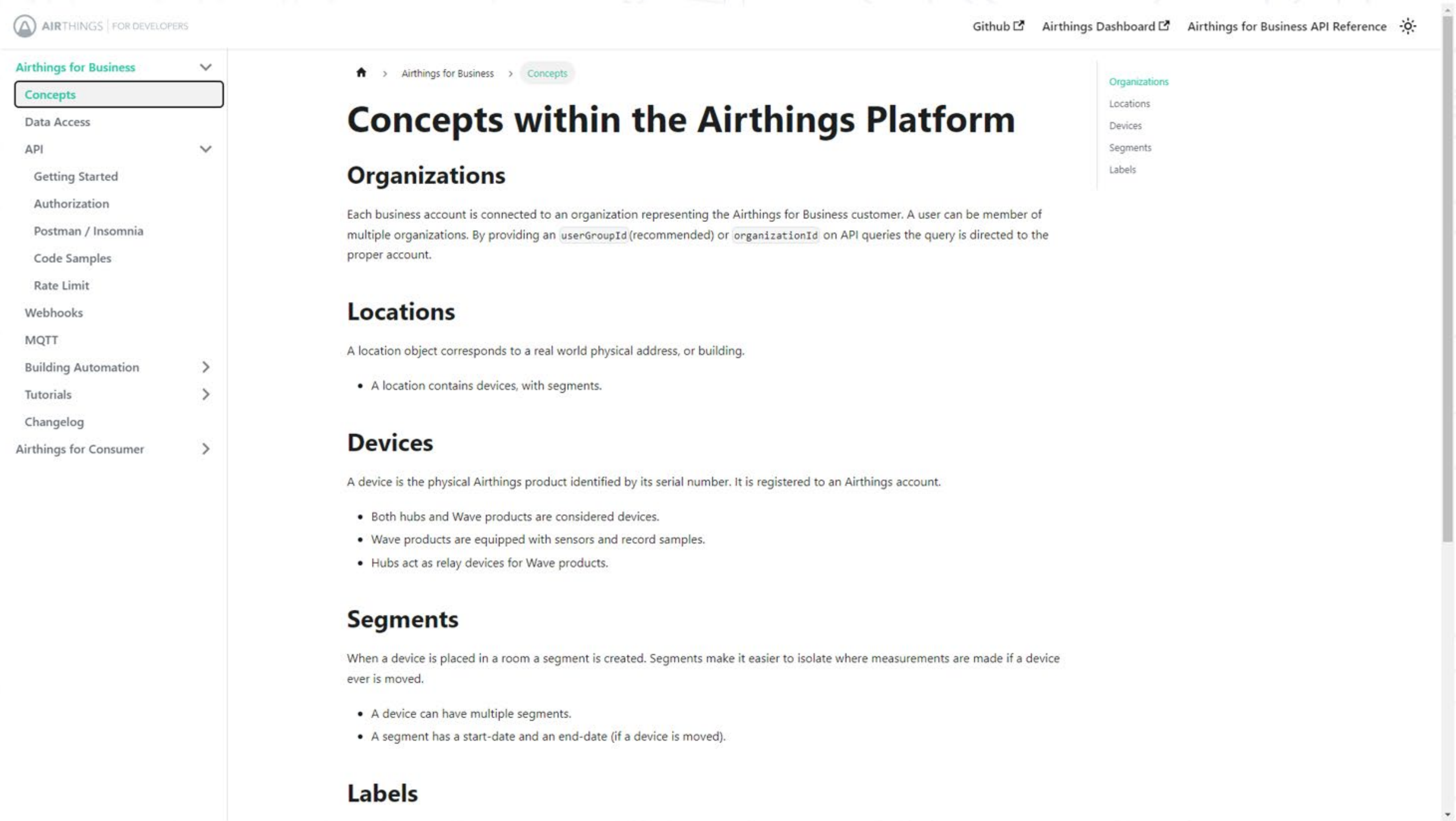
- Local/Remote device control
  - Integrate with APIs on IoT Devices / Gateways
- Bringing useful data into a building, from an API
  - Occupancy, Weather forecasts, Travel updates, Air quality data, parking…
- Exporting station data to an external services / cloud
  - REST APIs available for AWS IoT, Azure IoT, Google Cloud…

System Demo

NF 23

TRIDIUM

Documentation is Key



Concepts within the Airthings Platform

Organizations

Each business account is connected to an organization representing the Airthings for Business customer. A user can be member of multiple organizations. By providing an `userGroupId` (recommended) or `organizationId` on API queries the query is directed to the proper account.

Locations

A location object corresponds to a real world physical address, or building.

- A location contains devices, with segments.

Devices

A device is the physical Airthings product identified by its serial number. It is registered to an Airthings account.

- Both hubs and Wave products are considered devices.
- Wave products are equipped with sensors and record samples.
- Hubs act as relay devices for Wave products.

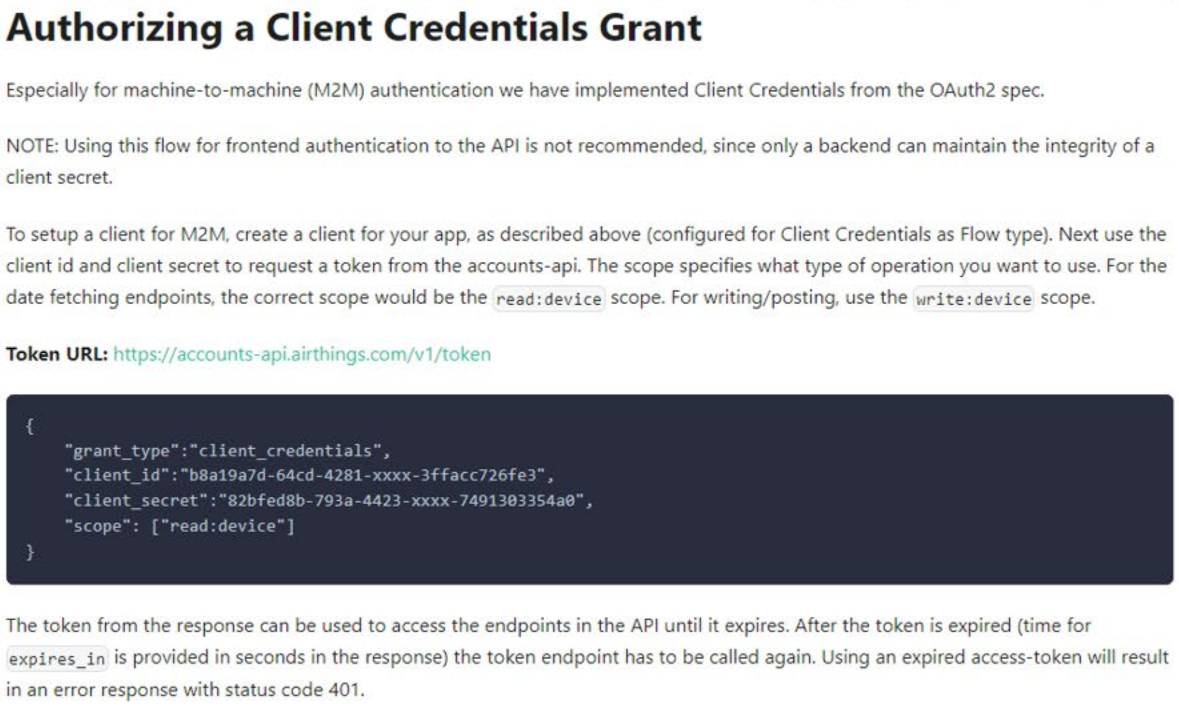
Segments

When a device is placed in a room a segment is created. Segments make it easier to isolate where measurements are made if a device ever is moved.

- A device can have multiple segments.
- A segment has a start-date and an end-date (if a device is moved).

Labels

Authorisation



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

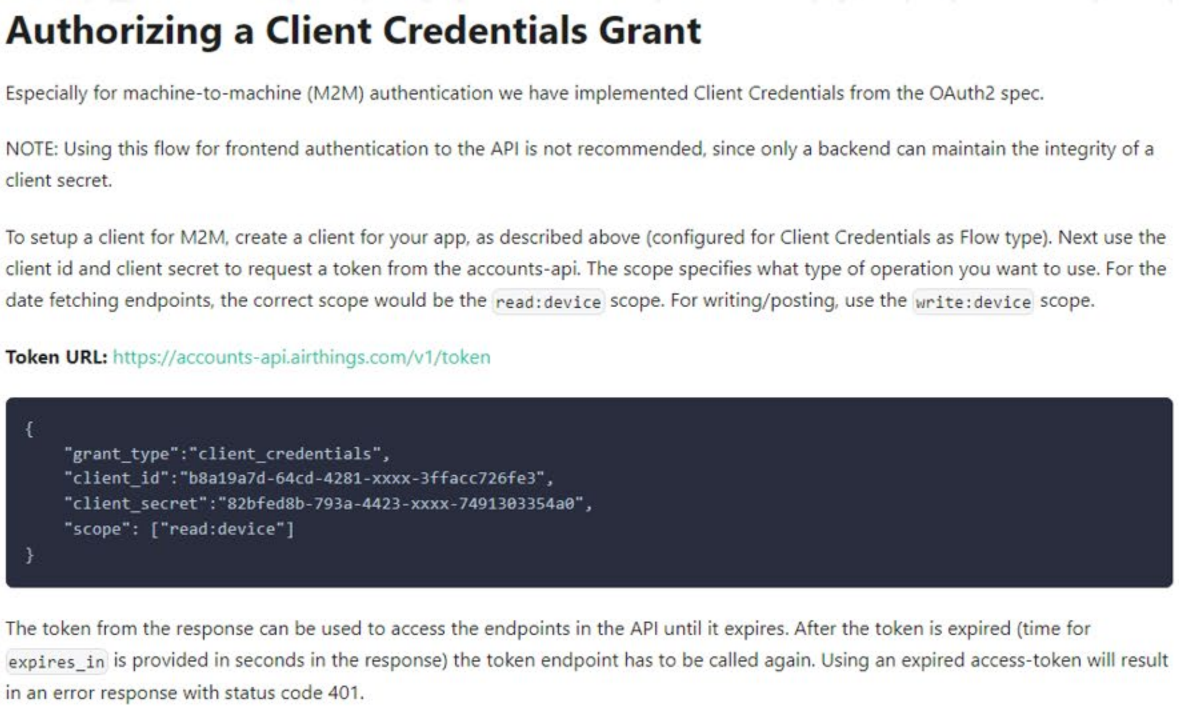
To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.

Authorisation



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

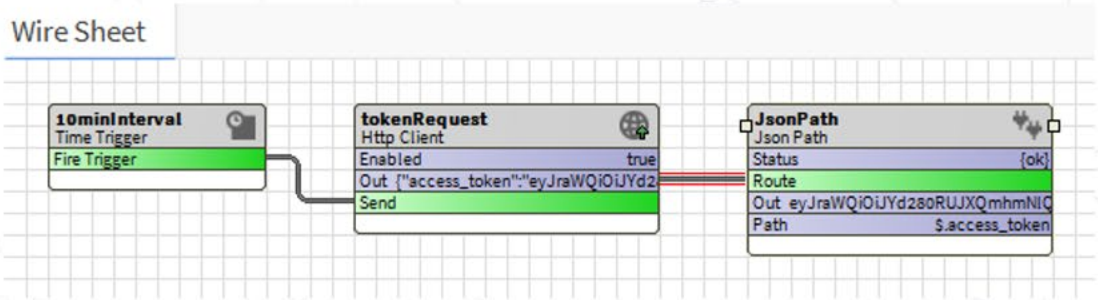
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.

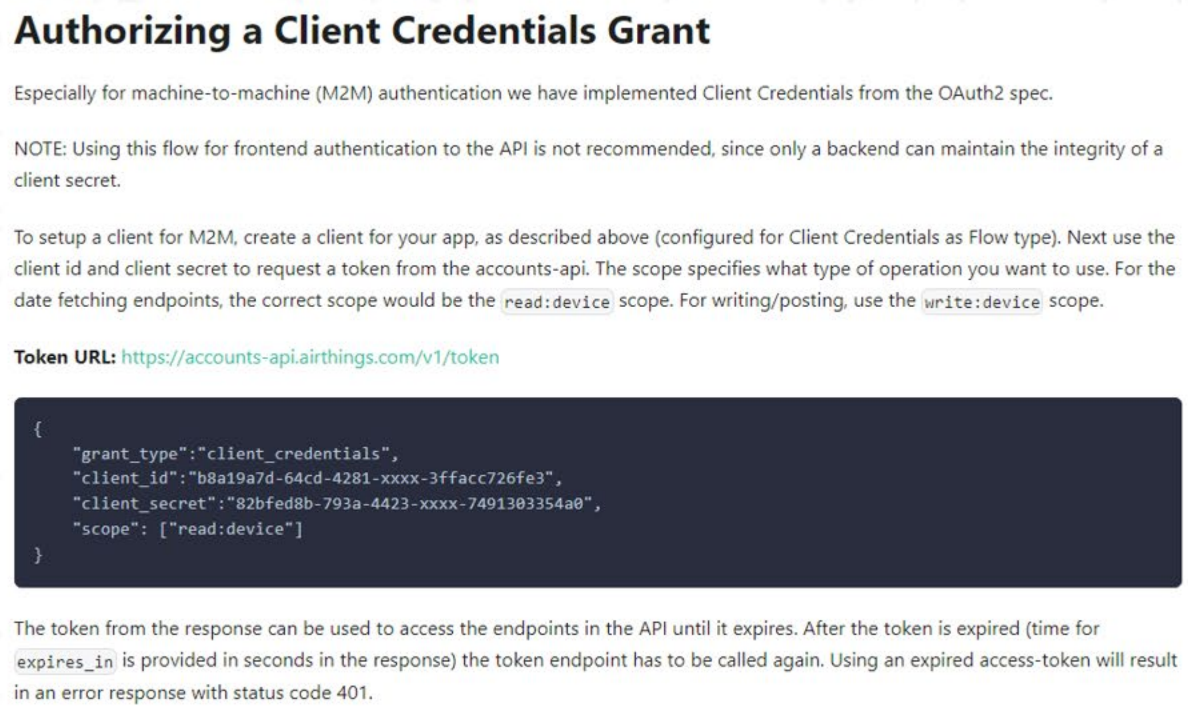


10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNIQ
Path $.access_token

Authorisation



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

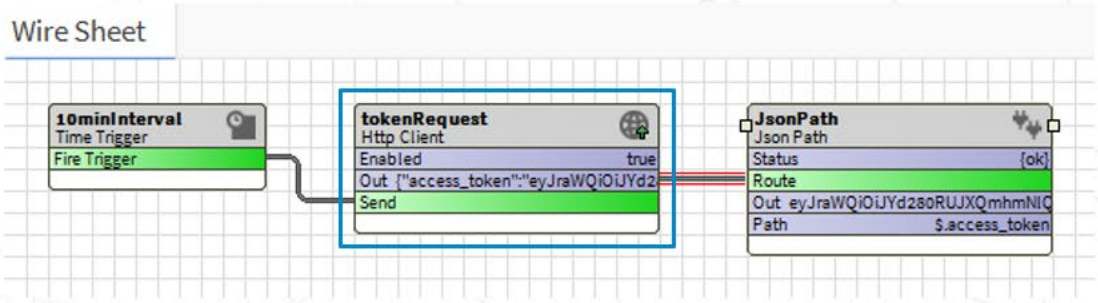
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.

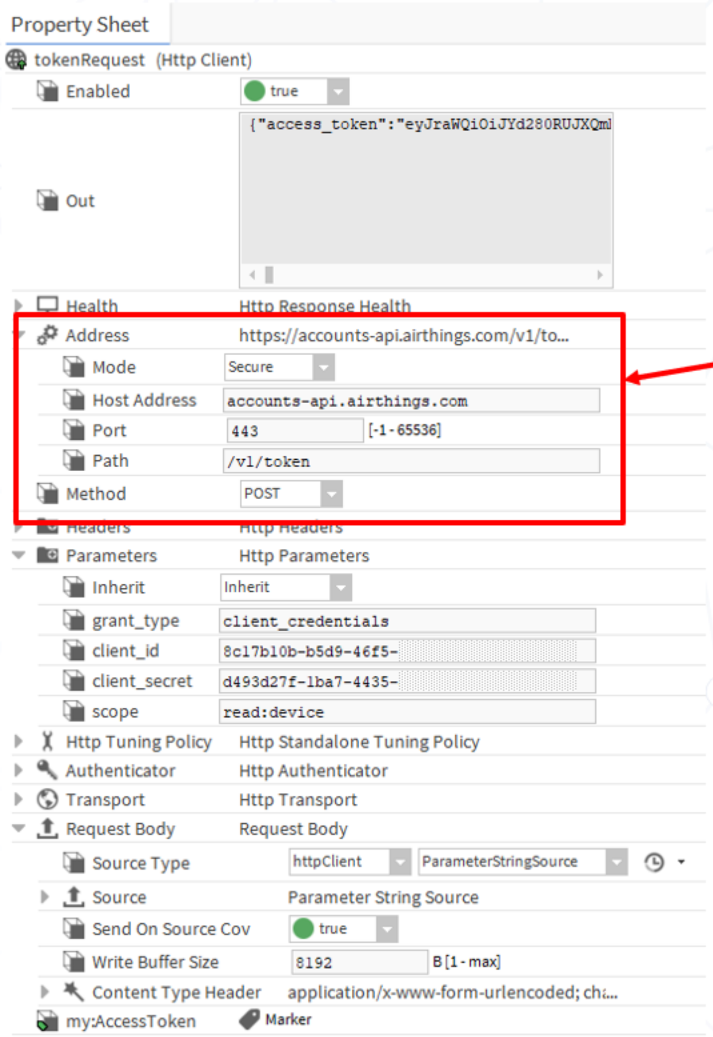


10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNIQ
Path $.access_token

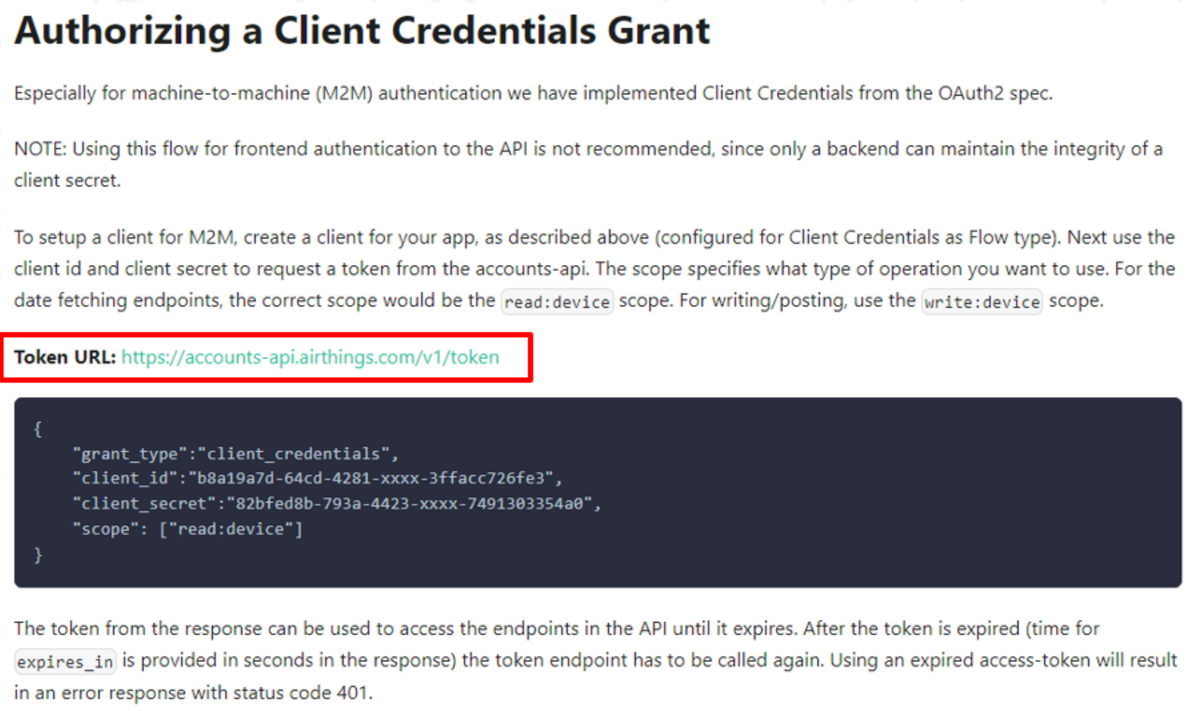
Authorisation



Property Sheet

tokenRequest (Http Client)

Enabled	true
Out	{"access_token":"eyJraWQiOiJYd28ORUJXQm
Health	Http Response Health
Address	https://accounts-api.airthings.com/v1/to...
Mode	Secure
Host Address	accounts-api.airthings.com
Port	443 [-1 - 65536]
Path	/v1/token
Method	POST
Headers	Http Headers
Parameters	Http Parameters
Inherit	Inherit
grant_type	client_credentials
client_id	8c17b10b-b5d9-46f5-
client_secret	d493d27f-1ba7-4435-
scope	read:device
Http Tuning Policy	Http Standalone Tuning Policy
Authenticator	Http Authenticator
Transport	Http Transport
Request Body	Request Body
Source Type	httpClient ParameterStringSource
Source	Parameter String Source
Send On Source Cov	true
Write Buffer Size	8192 B [1 - max]
Content Type Header	application/x-www-form-urlencoded; ch...
my:AccessToken	Marker



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

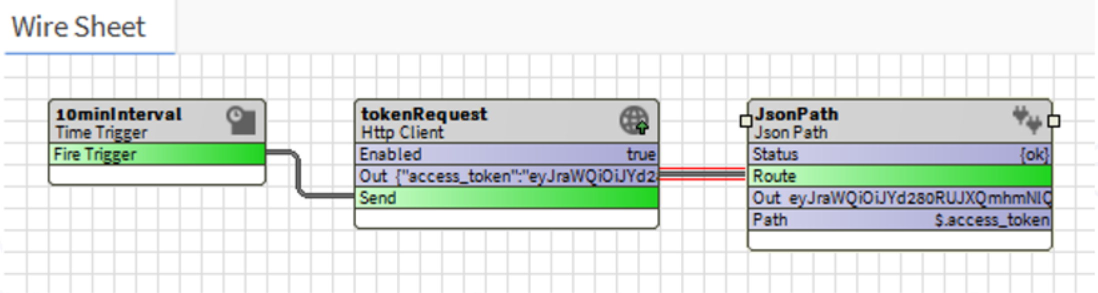
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.



Wire Sheet

10minInterval
Time Trigger
Fire Trigger

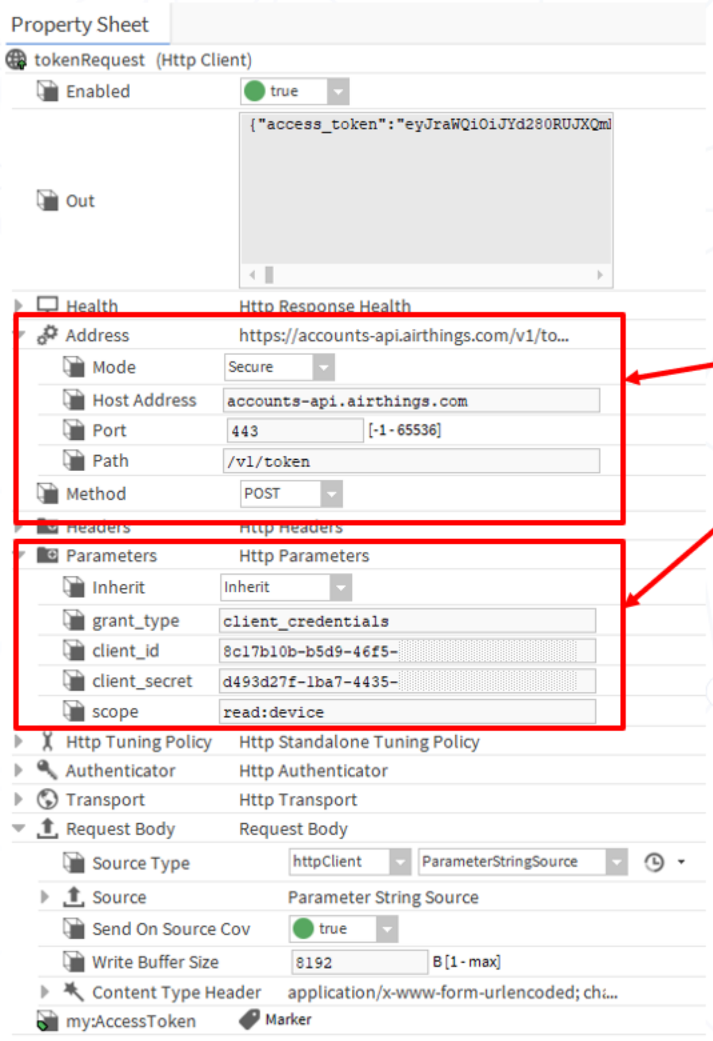
tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd28ORUJXQmhmNlQ
Path $.access_token

NF 23

TRIDIUM

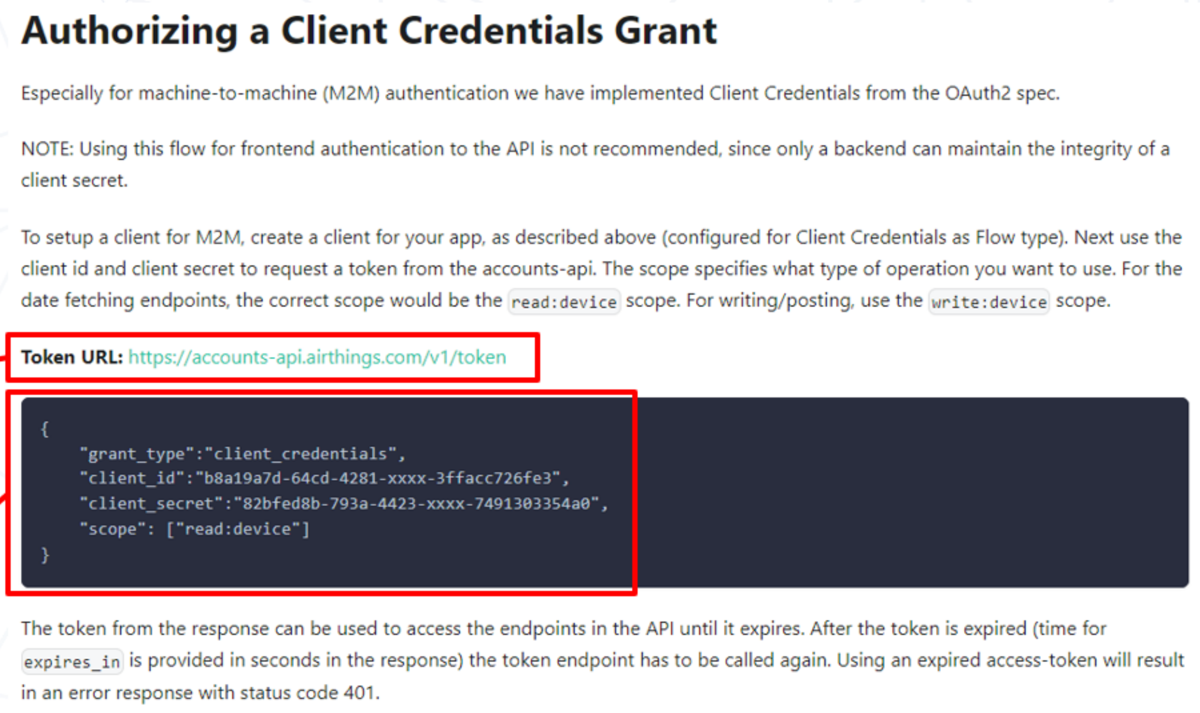
Authorisation



Property Sheet

tokenRequest (Http Client)

Enabled	true
Out	{"access_token":"eyJraWQiOiJYd28ORUJXQm
Health	Http Response Health
Address	https://accounts-api.airthings.com/v1/to...
Mode	Secure
Host Address	accounts-api.airthings.com
Port	443 [-1 - 65536]
Path	/v1/token
Method	POST
Headers	Http Headers
Parameters	Http Parameters
Inherit	Inherit
grant_type	client_credentials
client_id	8c17b10b-b5d9-46f5-[redacted]
client_secret	d493d27f-1ba7-4435-[redacted]
scope	read:device
Http Tuning Policy	Http Standalone Tuning Policy
Authenticator	Http Authenticator
Transport	Http Transport
Request Body	Request Body
Source Type	httpClient ParameterStringSource
Source	Parameter String Source
Send On Source Cov	true
Write Buffer Size	8192 B [1 - max]
Content Type Header	application/x-www-form-urlencoded; ch...
my:AccessToken	Marker



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

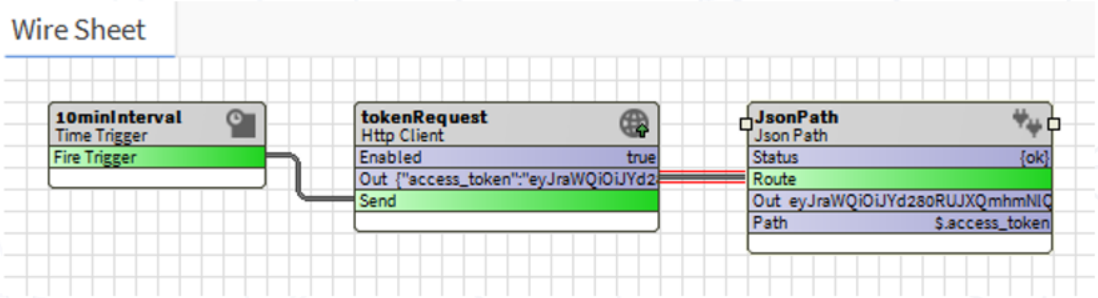
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.



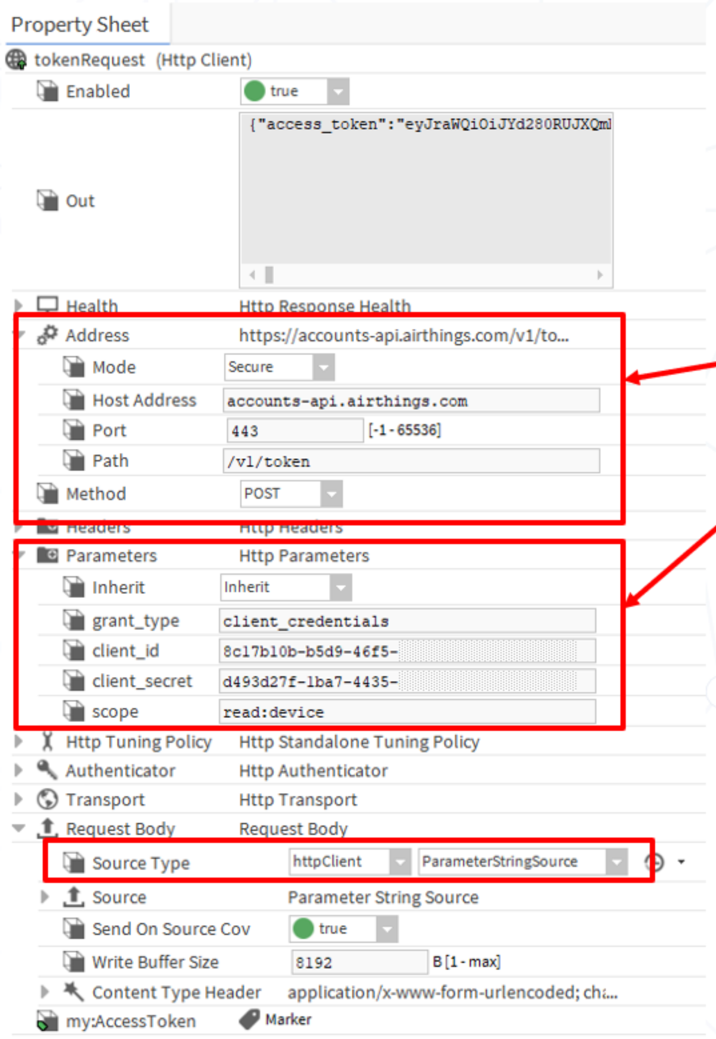
Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd28ORUJXQmhmNIC
Path $.access_token

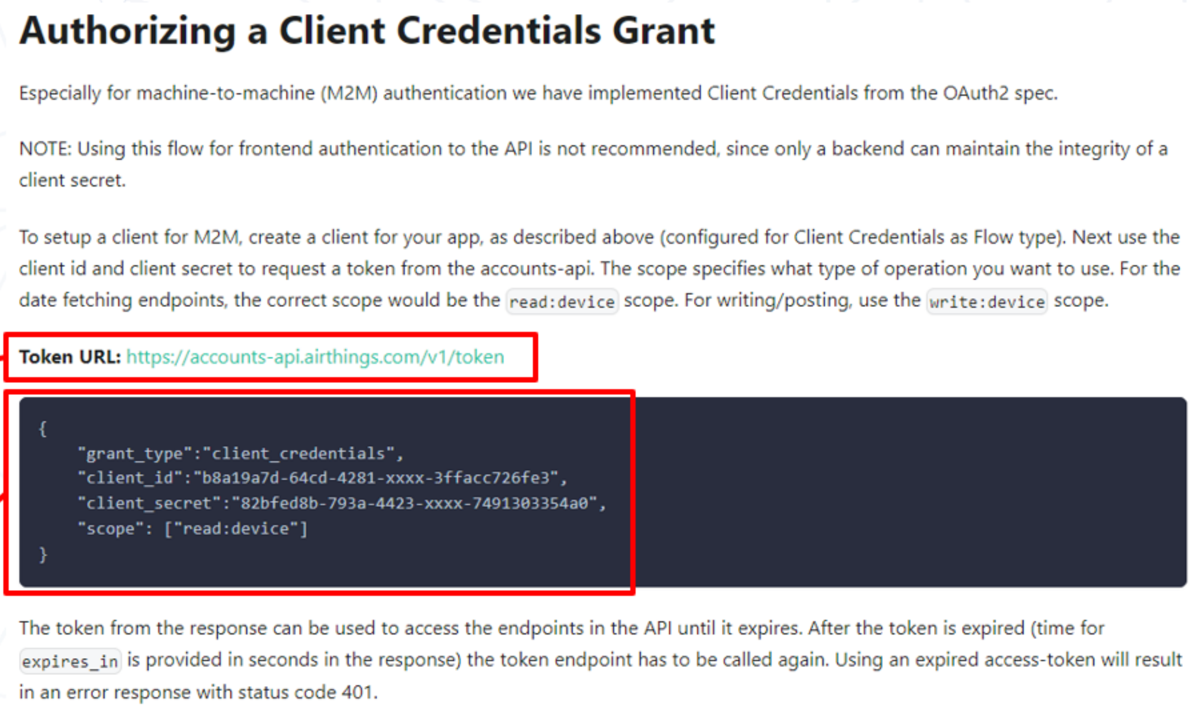
Authorisation



Property Sheet

tokenRequest (Http Client)

Enabled	true
Out	{"access_token":"eyJraWQiOiJYd280RUJXQml
Health	Http Response Health
Address	https://accounts-api.airthings.com/v1/to...
Mode	Secure
Host Address	accounts-api.airthings.com
Port	443 [-1 - 65536]
Path	/v1/token
Method	POST
Headers	Http Headers
Parameters	Http Parameters
Inherit	Inherit
grant_type	client_credentials
client_id	8c17b10b-b5d9-46f5-[redacted]
client_secret	d493d27f-1ba7-4435-[redacted]
scope	read:device
Http Tuning Policy	Http Standalone Tuning Policy
Authenticator	Http Authenticator
Transport	Http Transport
Request Body	Request Body
Source Type	httpClient ParameterStringSource
Source	Parameter String Source
Send On Source Cov	true
Write Buffer Size	8192 B [1 - max]
Content Type Header	application/x-www-form-urlencoded; ch...
my:AccessToken	Marker



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

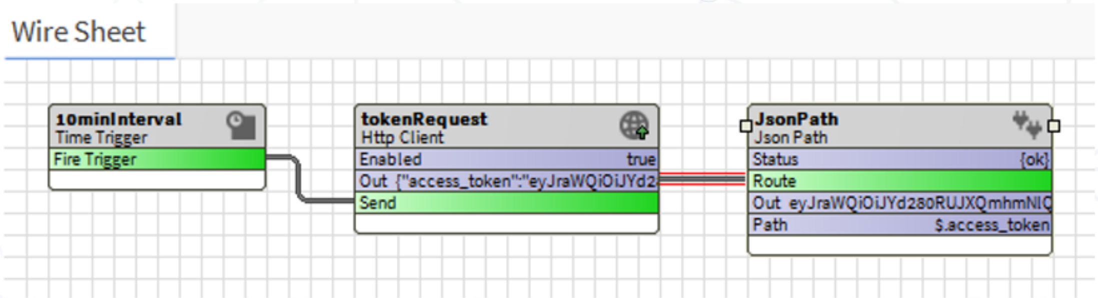
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

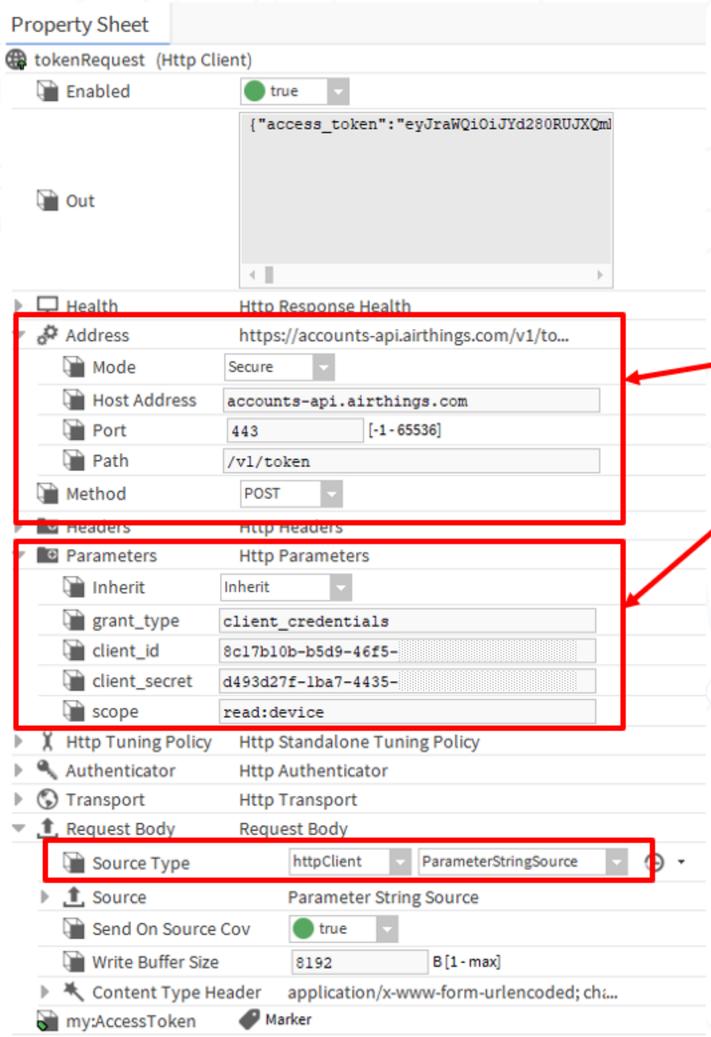
```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.



Wire Sheet

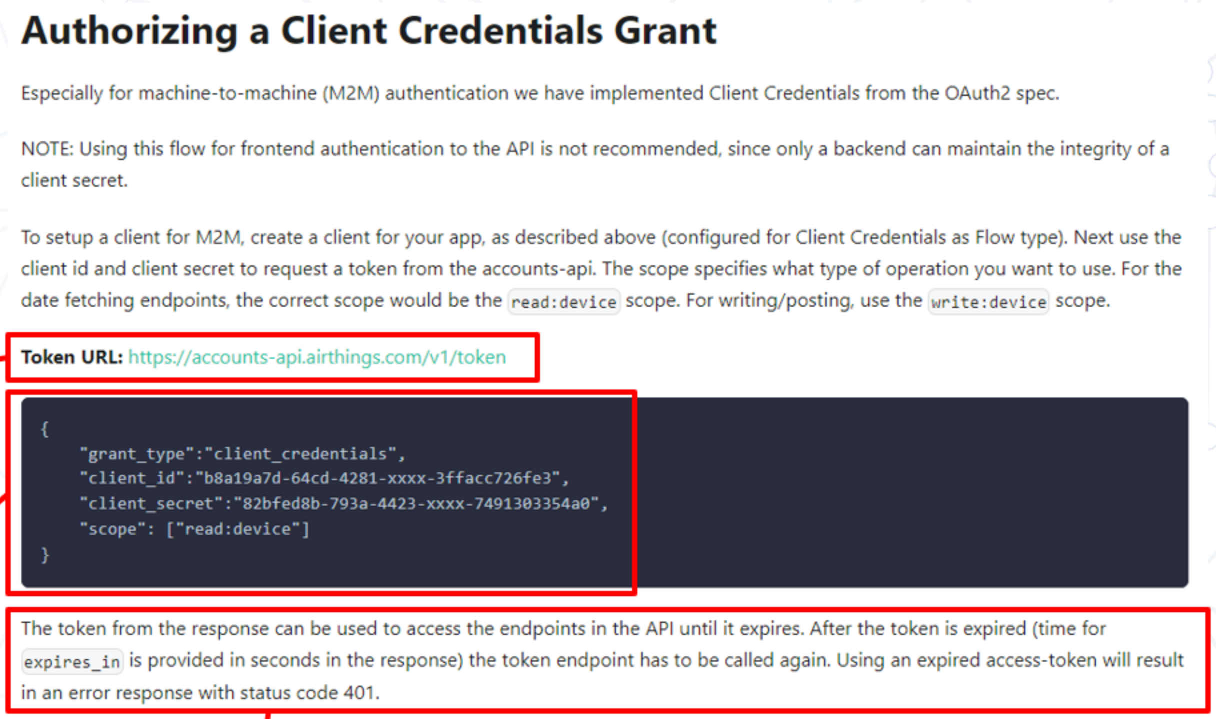
Authorisation



Property Sheet

tokenRequest (Http Client)

Enabled	true
Out	{"access_token":"eyJraWQiOiJYd280RUJXQml
Health	Http Response Health
Address	https://accounts-api.airthings.com/v1/to...
Mode	Secure
Host Address	accounts-api.airthings.com
Port	443 [-1 - 65536]
Path	/v1/token
Method	POST
Headers	Http Headers
Parameters	Http Parameters
Inherit	Inherit
grant_type	client_credentials
client_id	8c17b10b-b5d9-46f5-
client_secret	d493d27f-1ba7-4435-
scope	read:device
Http Tuning Policy	Http Standalone Tuning Policy
Authenticator	Http Authenticator
Transport	Http Transport
Request Body	Request Body
Source Type	httpClient ParameterStringSource
Source	Parameter String Source
Send On Source Cov	true
Write Buffer Size	8192 B [1 - max]
Content Type Header	application/x-www-form-urlencoded; ch...
my:AccessToken	Marker



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

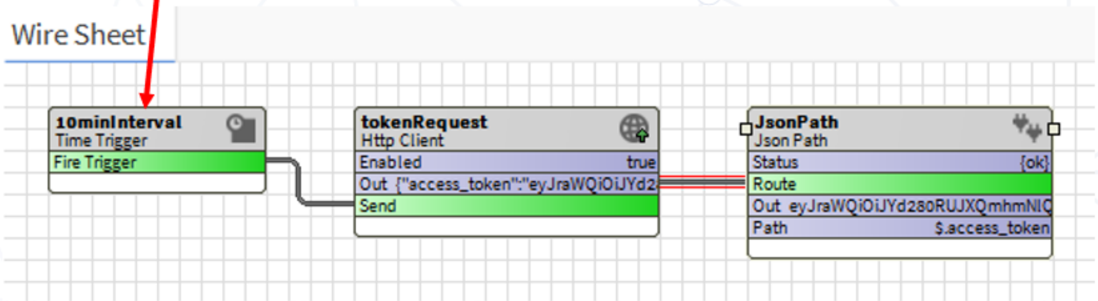
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.



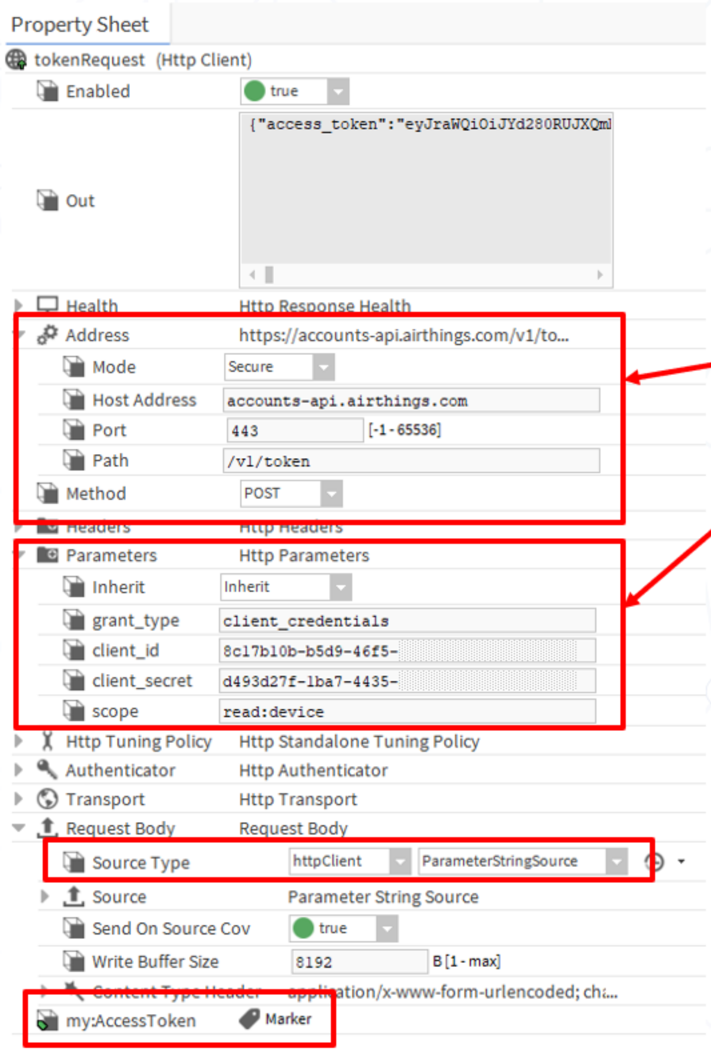
Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNlQ
Path $.access_token

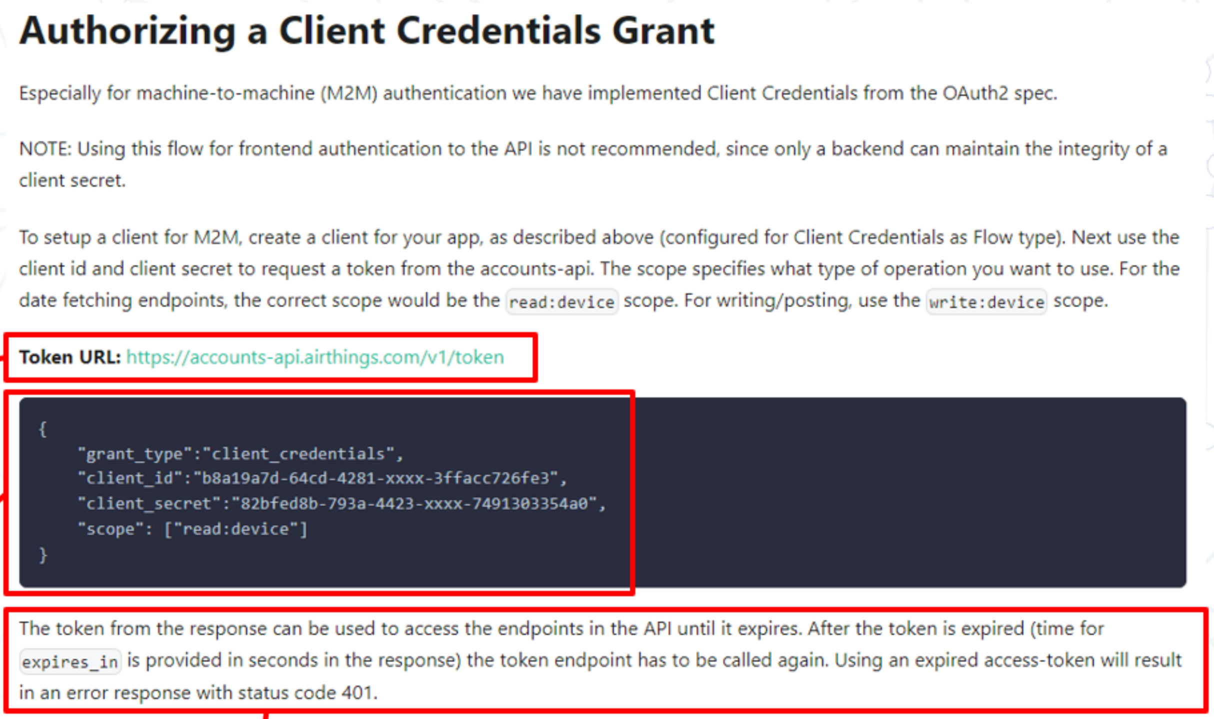
Authorisation



Property Sheet

tokenRequest (Http Client)

Enabled	● true
Out	{"access_token":"eyJraWQiOiJYd280RUJXQm
Health	Http Response Health
Address	https://accounts-api.airthings.com/v1/to...
Mode	Secure
Host Address	accounts-api.airthings.com
Port	443 [-1 - 65536]
Path	/v1/token
Method	POST
Headers	Http Headers
Parameters	Http Parameters
Inherit	Inherit
grant_type	client_credentials
client_id	8c17b10b-b5d9-46f5-
client_secret	d493d27f-1ba7-4435-
scope	read:device
Http Tuning Policy	Http Standalone Tuning Policy
Authenticator	Http Authenticator
Transport	Http Transport
Request Body	Request Body
Source Type	httpClient ParameterStringSource
Source	Parameter String Source
Send On Source Cov	● true
Write Buffer Size	8192 B [1 - max]
Content Type Header	application/x-www-form-urlencoded; ch...
my:AccessToken	Marker



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

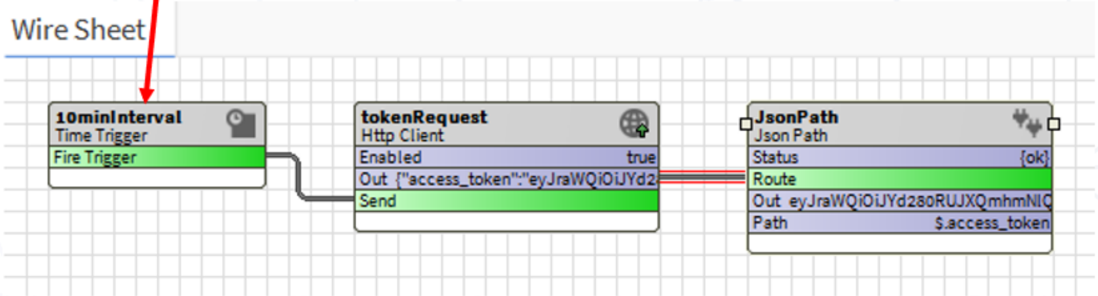
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.



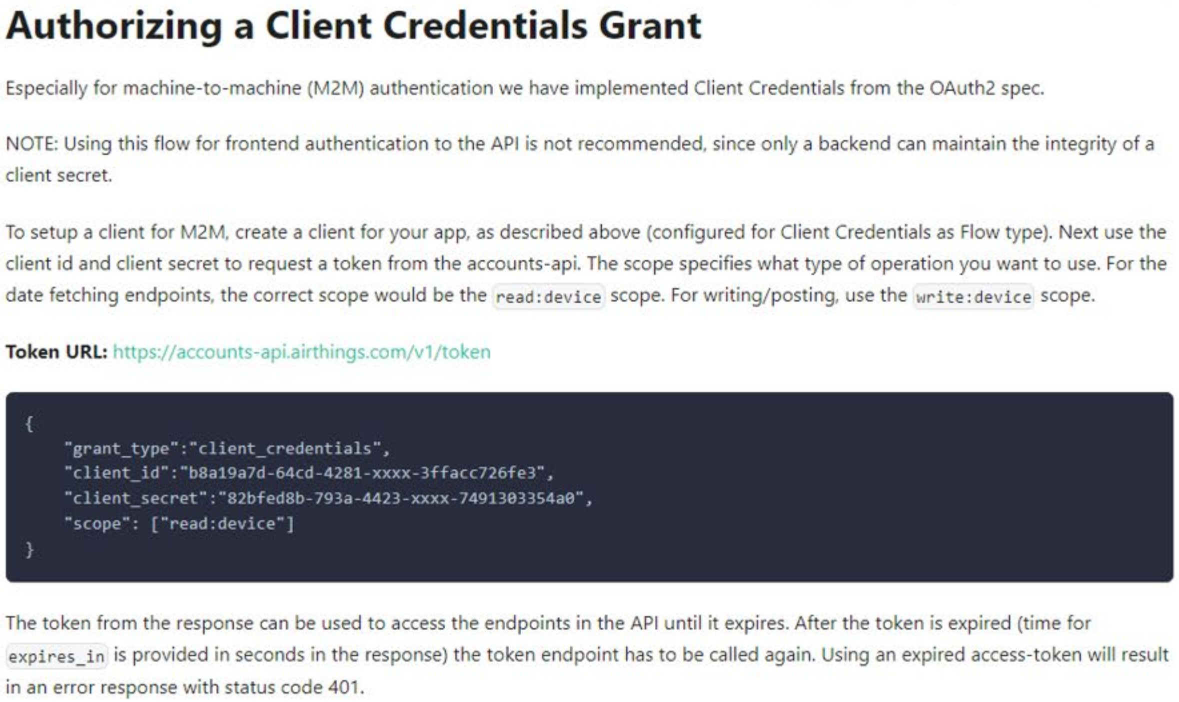
Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNIQ
Path $.access_token

Authorisation



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

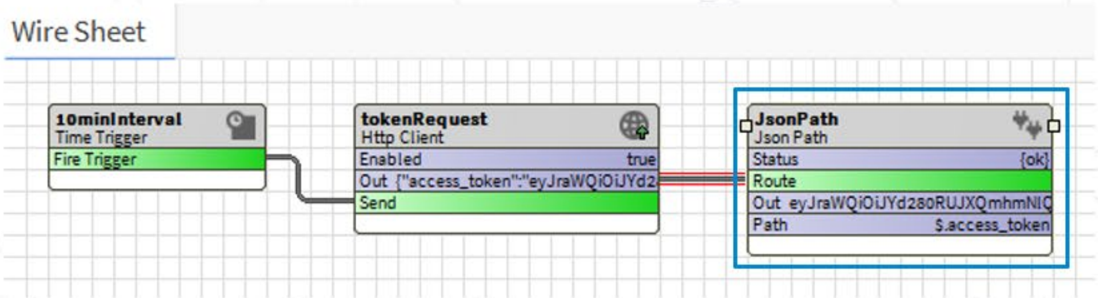
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.



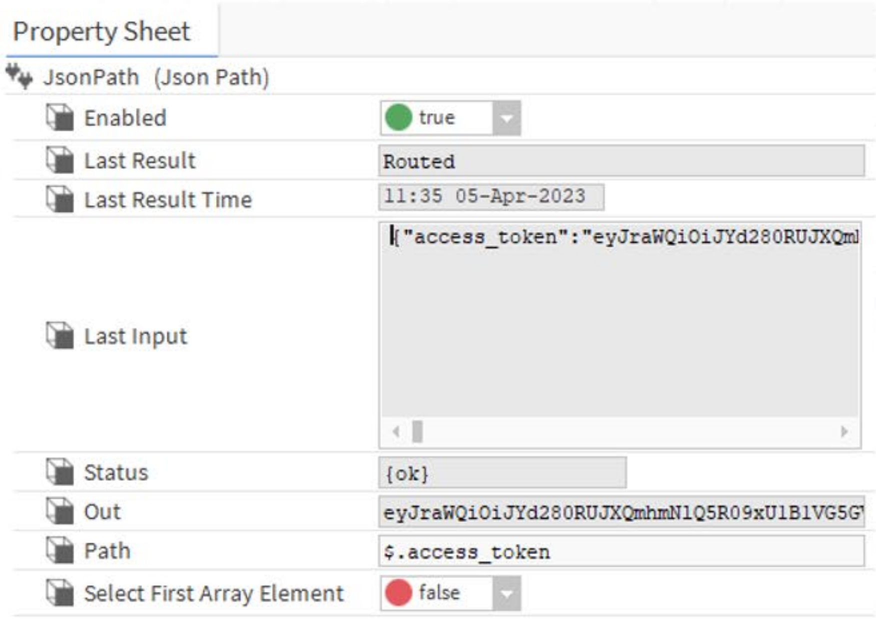
Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNIQ
Path $.access_token

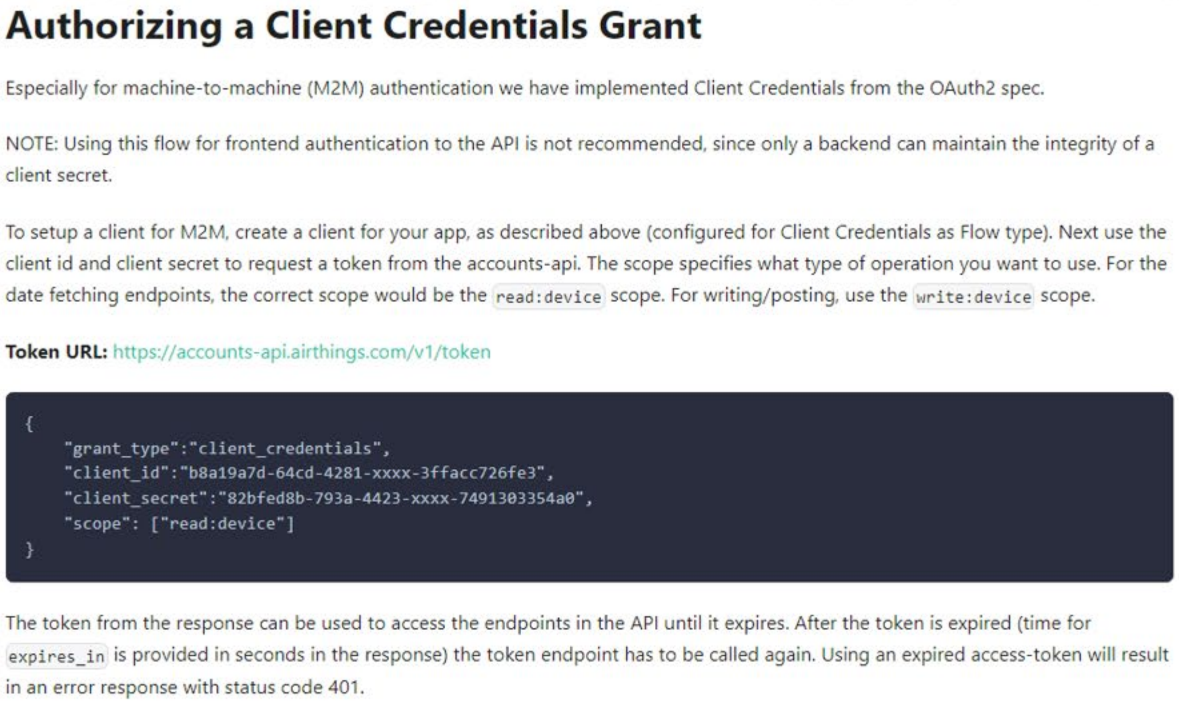
Authorisation



Property Sheet

JsonPath (Json Path)

Enabled	● true
Last Result	Routed
Last Result Time	11:35 05-Apr-2023
Last Input	{"access_token":"eyJraWQiOiJYd280RUJXQml
Status	{ok}
Out	eyJraWQiOiJYd280RUJXQmhmNlQ5R09xUlBlVG5G
Path	$.access_token
Select First Array Element	● false



Authorizing a Client Credentials Grant

Especially for machine-to-machine (M2M) authentication we have implemented Client Credentials from the OAuth2 spec.

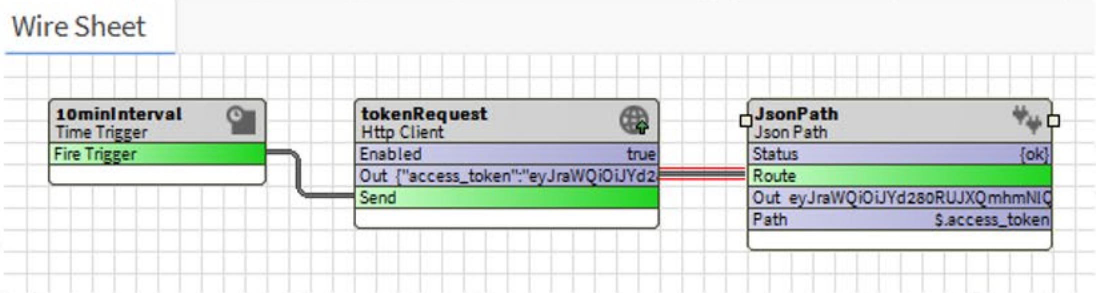
NOTE: Using this flow for frontend authentication to the API is not recommended, since only a backend can maintain the integrity of a client secret.

To setup a client for M2M, create a client for your app, as described above (configured for Client Credentials as Flow type). Next use the client id and client secret to request a token from the accounts-api. The scope specifies what type of operation you want to use. For the date fetching endpoints, the correct scope would be the `read:device` scope. For writing/posting, use the `write:device` scope.

Token URL: https://accounts-api.airthings.com/v1/token

```
{
    "grant_type":"client_credentials",
    "client_id":"b8a19a7d-64cd-4281-xxxx-3ffacc726fe3",
    "client_secret":"82bfed8b-793a-4423-xxxx-7491303354a0",
    "scope": ["read:device"]
}
```

The token from the response can be used to access the endpoints in the API until it expires. After the token is expired (time for `expires_in` is provided in seconds in the response) the token endpoint has to be called again. Using an expired access-token will result in an error response with status code 401.



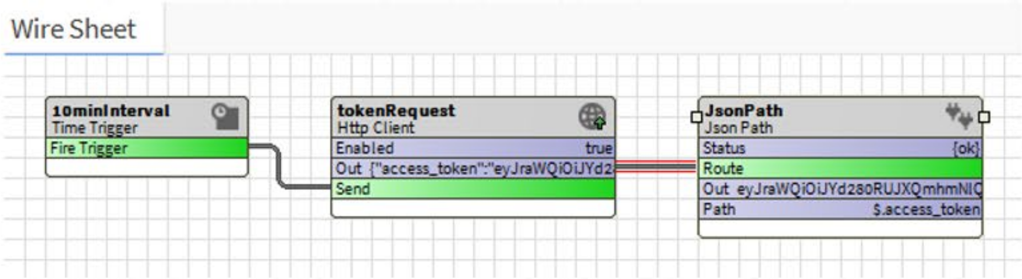
Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNlQ
Path $.access_token

Authorisation

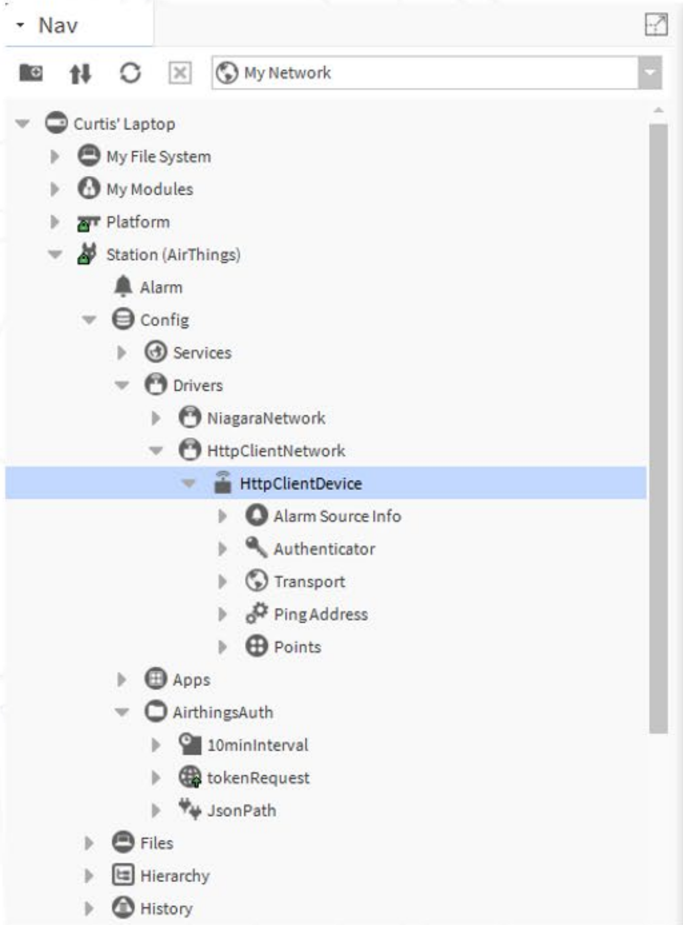


Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

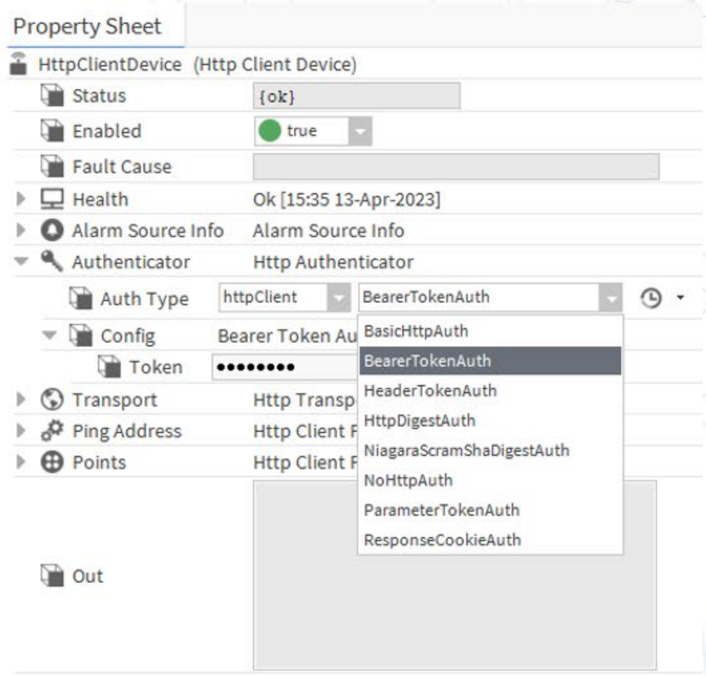
JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd28oRUJXQmhmNlC
Path $.access_token



Nav

My Network

- Curtis' Laptop
  - My File System
  - My Modules
  - Platform
  - Station (AirThings)
    - Alarm
    - Config
      - Services
      - Drivers
        - NiagaraNetwork
        - HttpClientNetwork
          - HttpClientDevice
            - Alarm Source Info
            - Authenticator
            - Transport
            - Ping Address
            - Points
      - Apps
      - AirthingsAuth
        - 10minInterval
        - tokenRequest
        - JsonPath
    - Files
    - Hierarchy
    - History

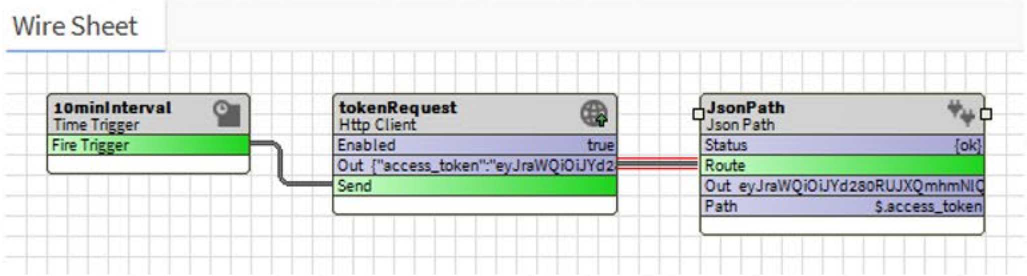


Property Sheet

HttpClientDevice (Http Client Device)

Status {ok}
Enabled true
Fault Cause
Health Ok [15:35 13-Apr-2023]
Alarm Source Info Alarm Source Info
Authenticator Http Authenticator
Auth Type httpClient BearerTokenAuth
Config Bearer Token Au
Token ••••••••
Transport Http Transp
Ping Address Http Client F
Points Http Client F
Out

Authorisation

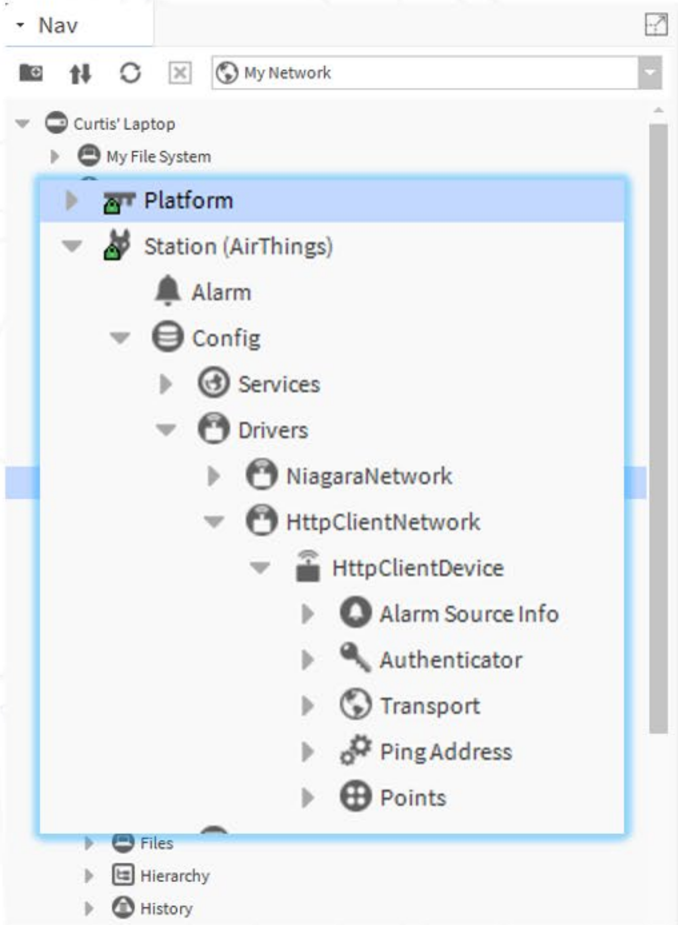


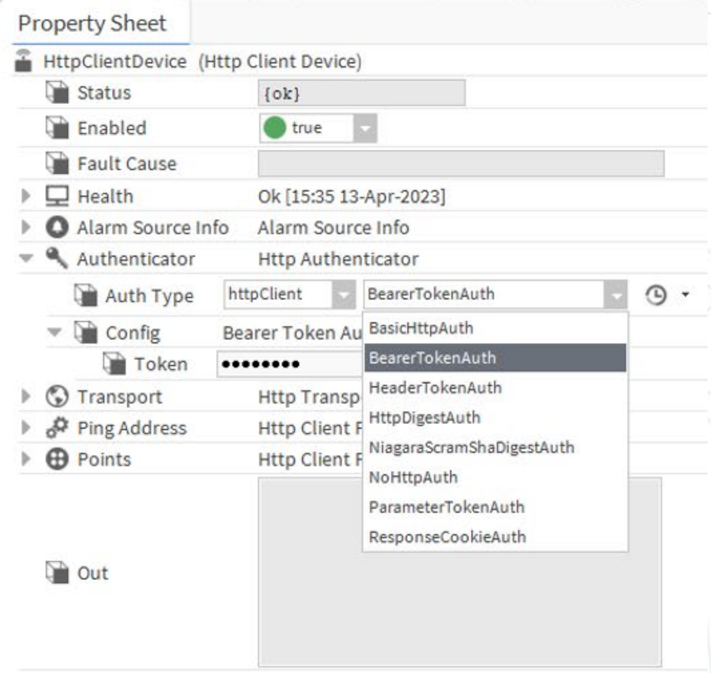
Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNIC
Path $.access_token



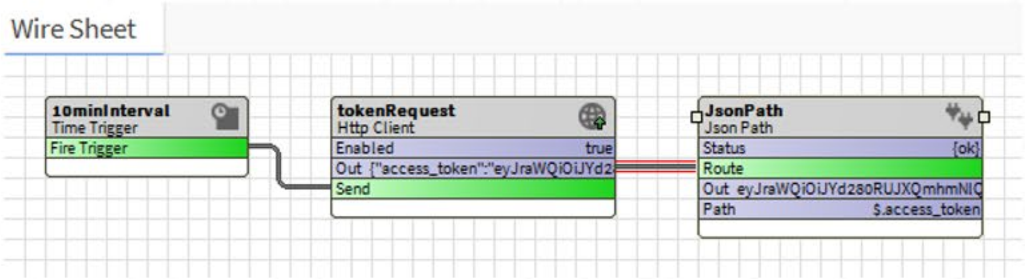


Property Sheet

HttpClientDevice (Http Client Device)

Status	{ok}
Enabled	true
Fault Cause	
Health	Ok [15:35 13-Apr-2023]
Alarm Source Info	Alarm Source Info
Authenticator	Http Authenticator
Auth Type	httpClient BearerTokenAuth
Config	Bearer Token Au
Token	••••••••
Transport	Http Transp
Ping Address	Http Client F
Points	Http Client F
Out	

Authorisation

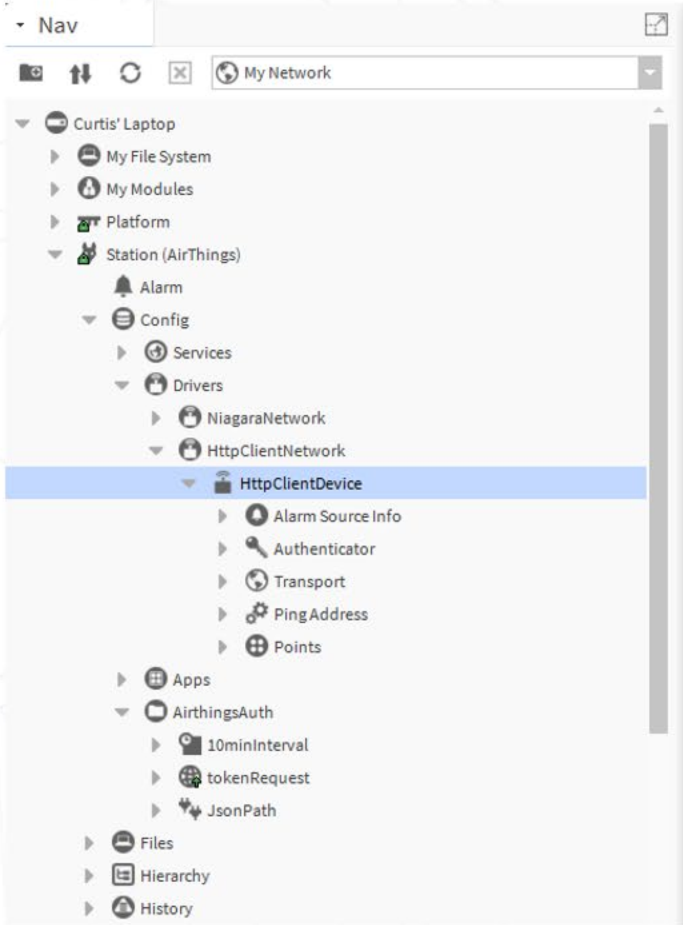


Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

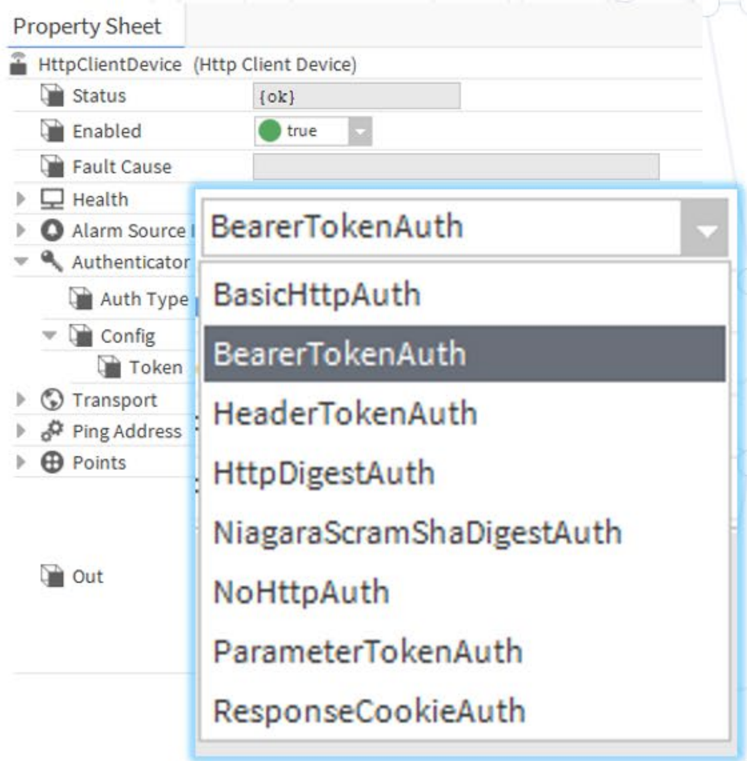
JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNlQ
Path $.access_token



Nav

My Network

- Curtis' Laptop
  - My File System
  - My Modules
  - Platform
  - Station (AirThings)
    - Alarm
    - Config
      - Services
      - Drivers
        - NiagaraNetwork
        - HttpClientNetwork
          - HttpClientDevice
            - Alarm Source Info
            - Authenticator
            - Transport
            - Ping Address
            - Points
      - Apps
      - AirthingsAuth
        - 10minInterval
        - tokenRequest
        - JsonPath
    - Files
    - Hierarchy
    - History



Property Sheet

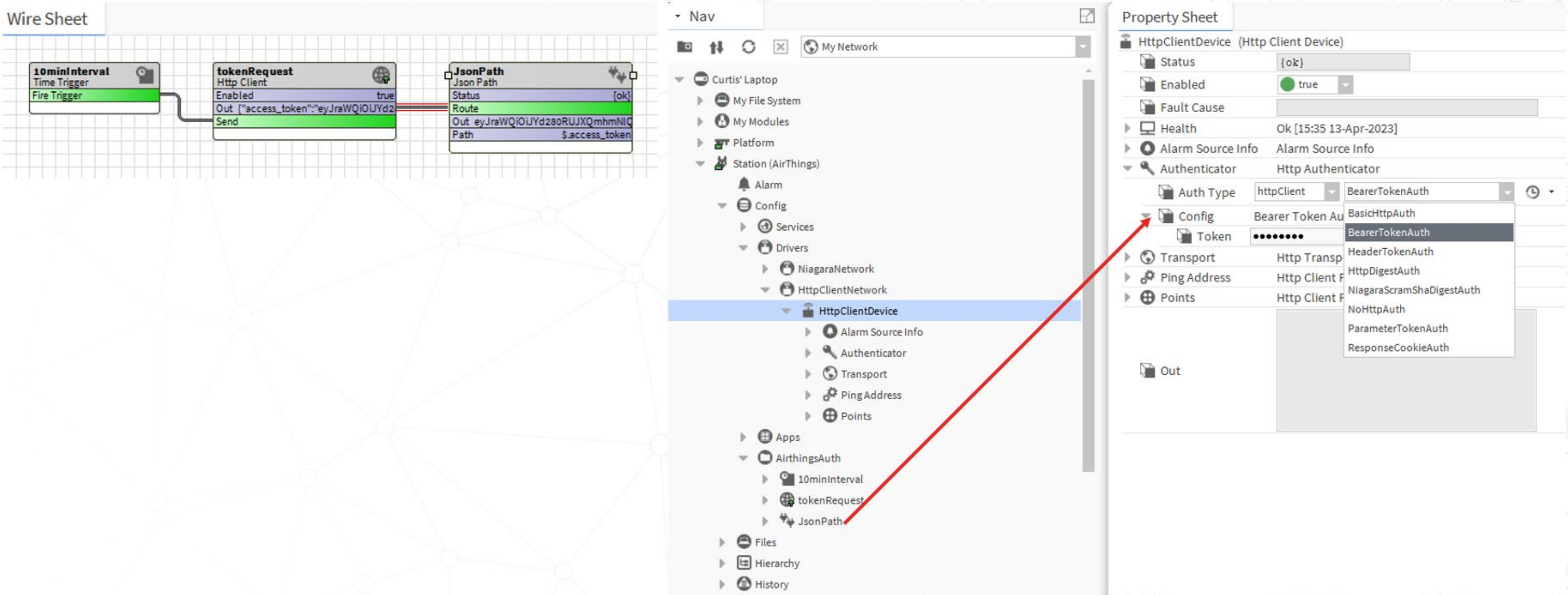
HttpClientDevice (Http Client Device)

- Status {ok}
- Enabled true
- Fault Cause
- Health
- Alarm Source I
- Authenticator
  - Auth Type BearerTokenAuth
  - Config
    - Token
- Transport
- Ping Address
- Points
- Out

BearerTokenAuth

- BasicHttpAuth
- BearerTokenAuth
- HeaderTokenAuth
- HttpDigestAuth
- NiagaraScramShaDigestAuth
- NoHttpAuth
- ParameterTokenAuth
- ResponseCookieAuth

Authorisation



Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

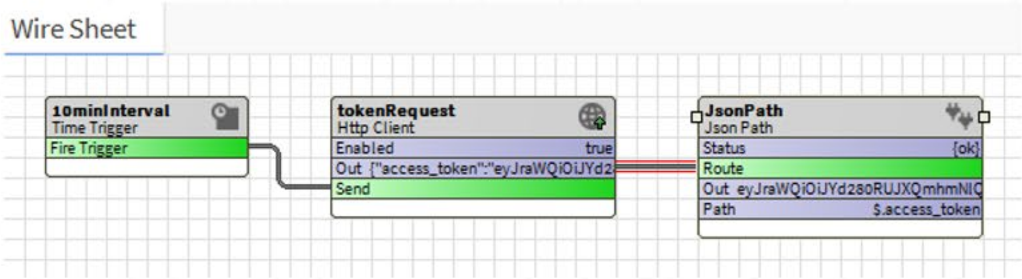
JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNIQ
Path $.access_token

Property Sheet

HttpClientDevice (Http Client Device)

Status	{ok}
Enabled	true
Fault Cause	
Health	Ok [15:35 13-Apr-2023]
Alarm Source Info	Alarm Source Info
Authenticator	Http Authenticator
Auth Type	httpClient BearerTokenAuth
Config	Bearer Token Au
Token	●●●●●●●●●
Transport	Http Transp
Ping Address	Http Client F
Points	Http Client F
Out	

Authorisation

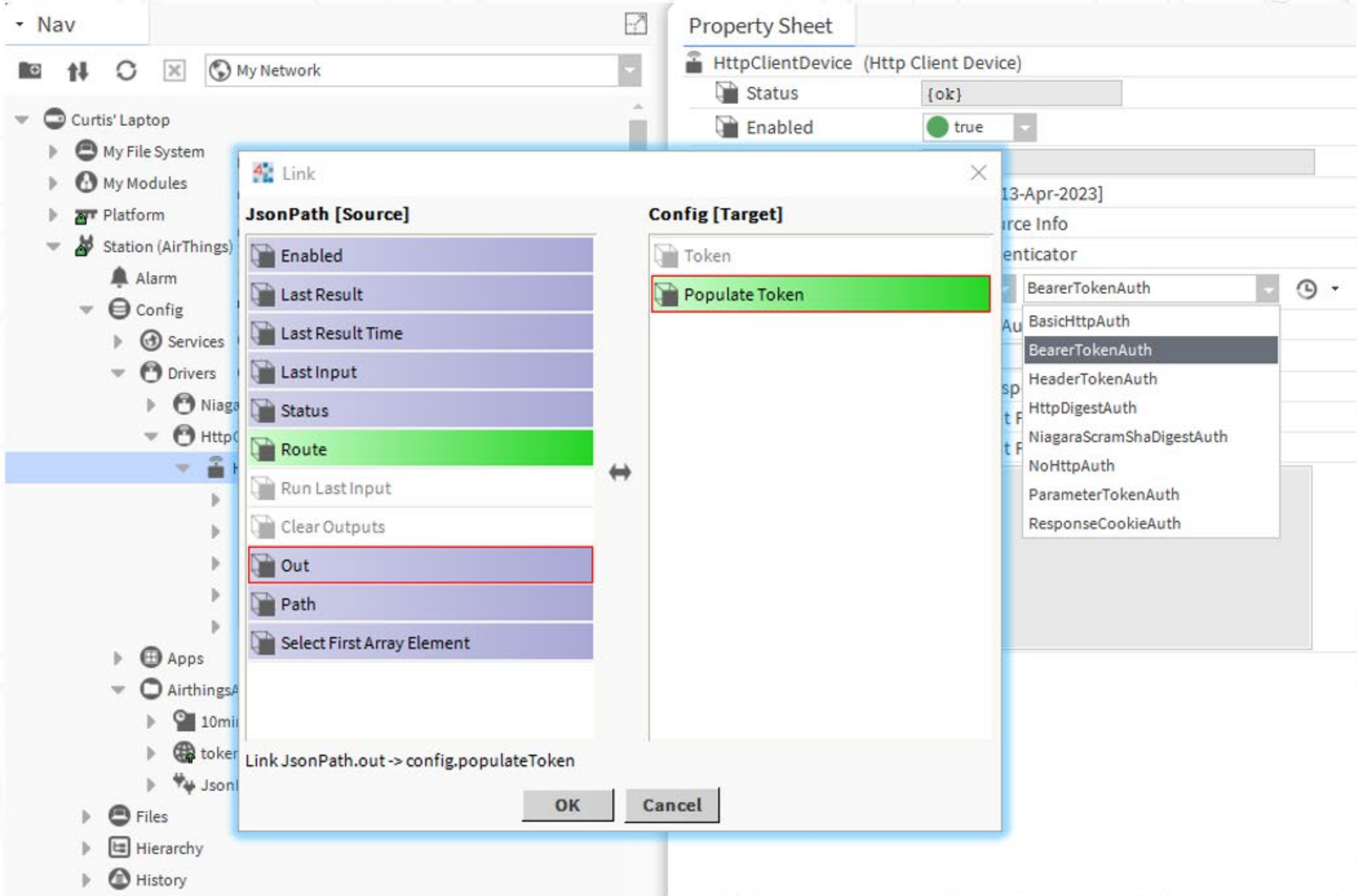


Wire Sheet

10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNIC
Path $.access_token



Link

JsonPath [Source]

- Enabled
- Last Result
- Last Result Time
- Last Input
- Status
- Route
- Run Last Input
- Clear Outputs
- Out
- Path
- Select First Array Element

Config [Target]

- Token
- Populate Token

Link JsonPath.out -> config.populateToken

OK Cancel

Property Sheet

HttpClientDevice (Http Client Device)

Status {ok}

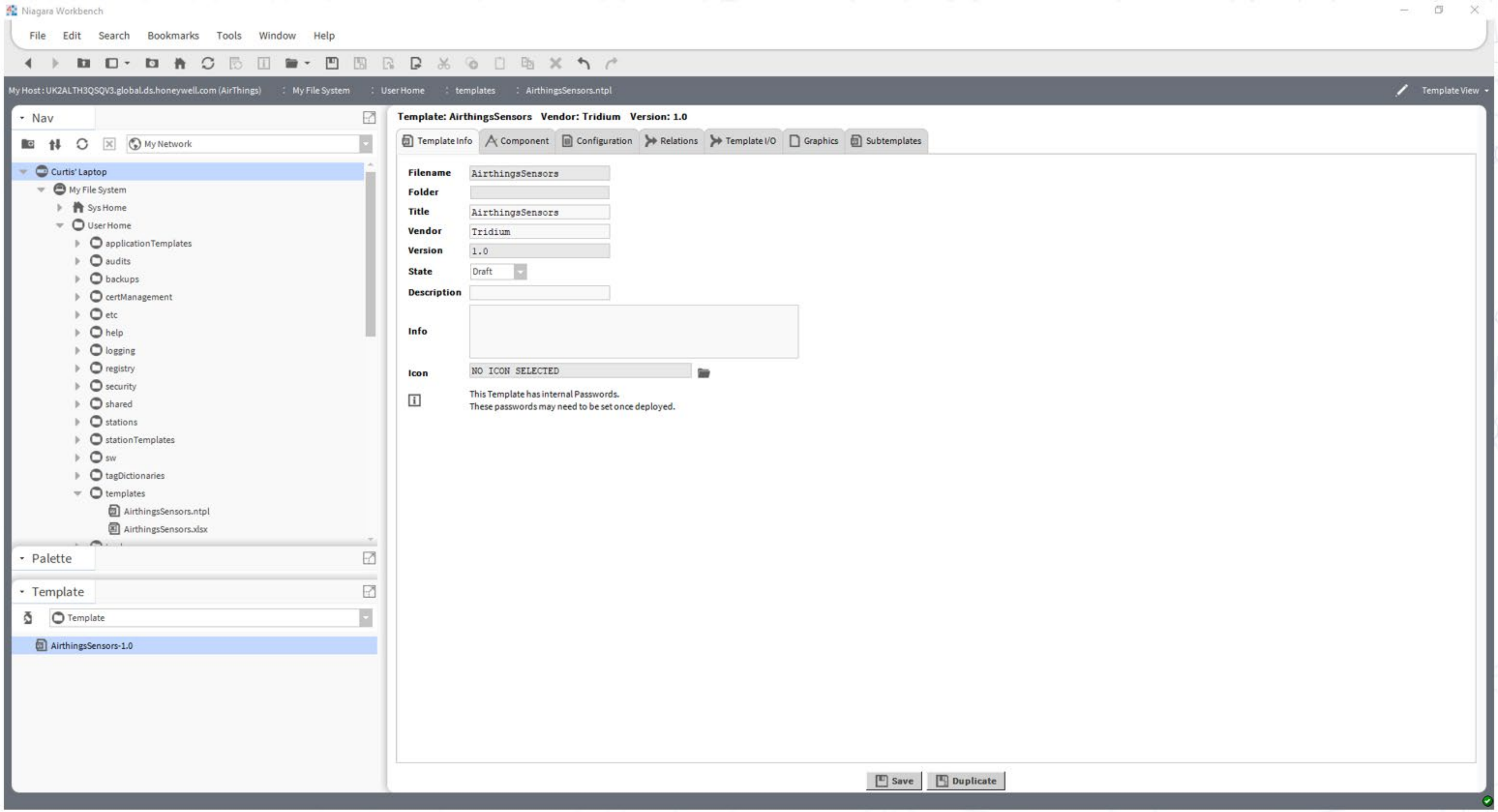
Enabled true

13-Apr-2023]

rce Info

enticator BearerTokenAuth

Creating a Template

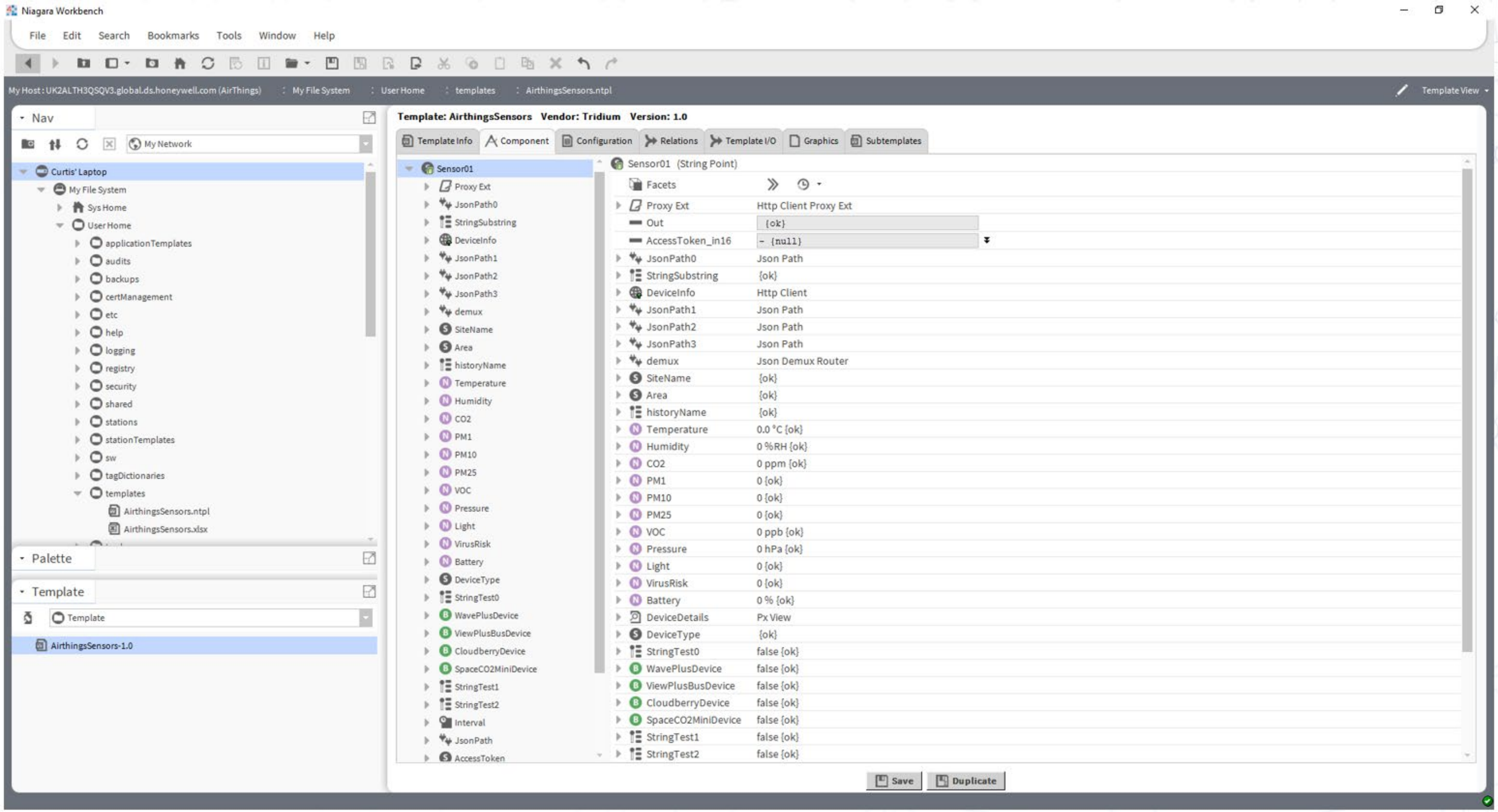


Template: AirthingsSensors Vendor: Tridium Version: 1.0

Filename	AirthingsSensors
Folder	
Title	AirthingsSensors
Vendor	Tridium
Version	1.0
State	Draft
Description	
Info	
Icon	NO ICON SELECTED

This Template has internal Passwords.
These passwords may need to be set once deployed.

Creating a Template

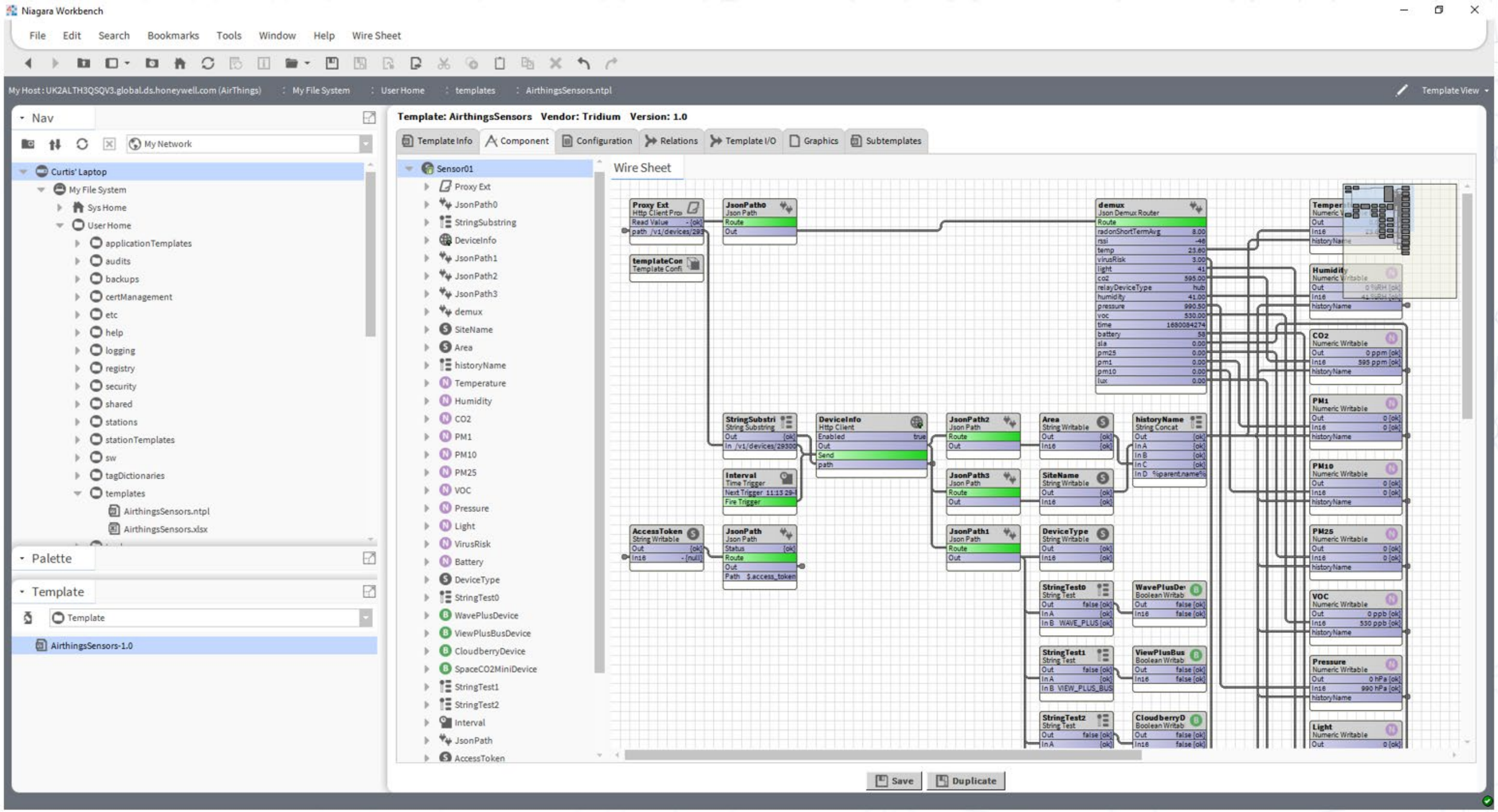


Template: AirthingsSensors Vendor: Tridium Version: 1.0

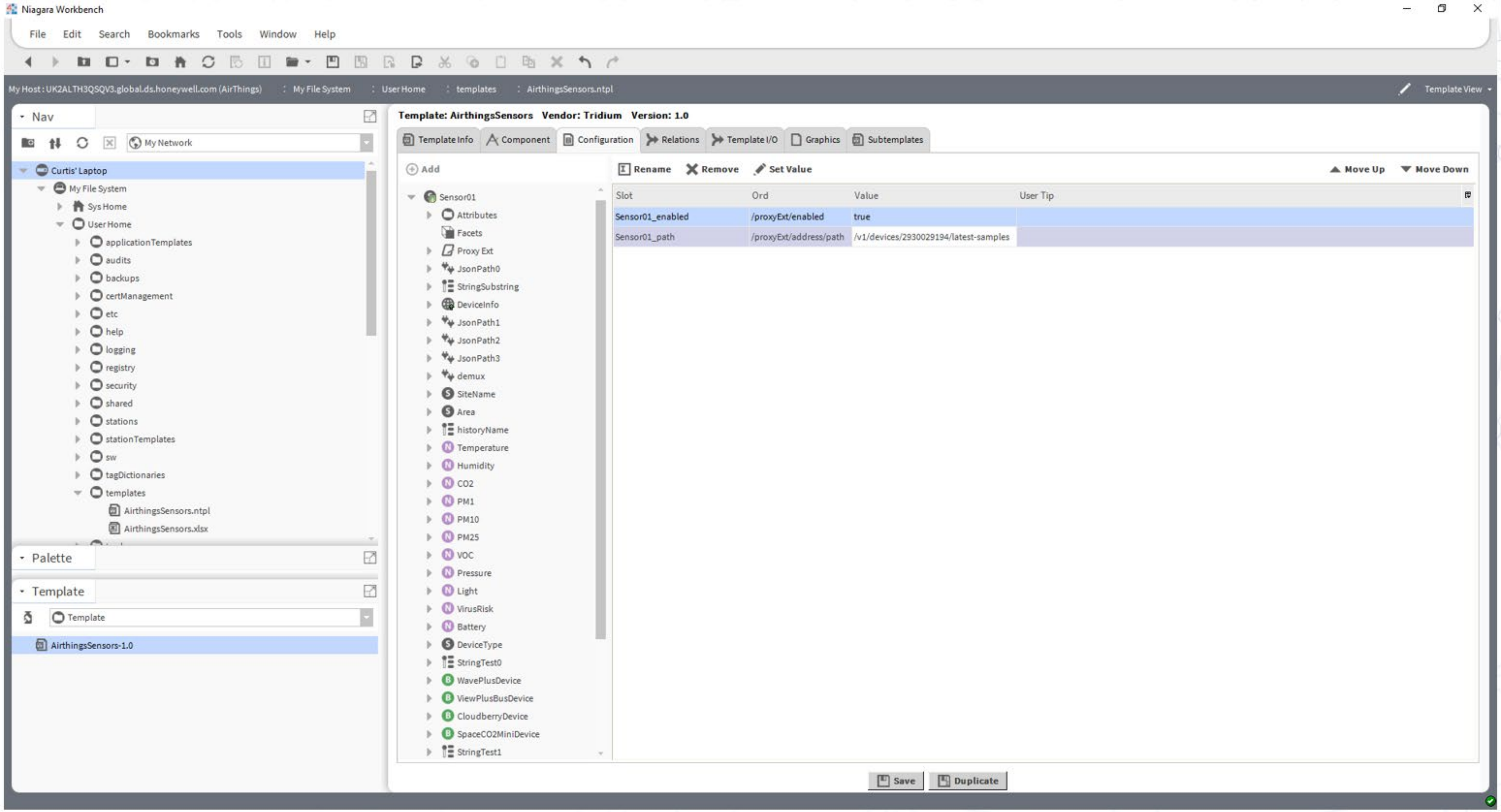
Sensor01 (String Point)

Facets	
Proxy Ext	Http Client Proxy Ext
Out	{ok}
AccessToken_in16	- {null}
JsonPath0	Json Path
StringSubstring	{ok}
DeviceInfo	Http Client
JsonPath1	Json Path
JsonPath2	Json Path
JsonPath3	Json Path
demux	Json Demux Router
SiteName	{ok}
Area	{ok}
historyName	{ok}
Temperature	0.0 °C {ok}
Humidity	0 %RH {ok}
CO2	0 ppm {ok}
PM1	0 {ok}
PM10	0 {ok}
PM25	0 {ok}
VOC	0 ppb {ok}
Pressure	0 hPa {ok}
Light	0 {ok}
VirusRisk	0 {ok}
Battery	0 % {ok}
DeviceDetails	Px View
DeviceType	{ok}
StringTest0	false {ok}
WavePlusDevice	false {ok}
ViewPlusBusDevice	false {ok}
CloudberryDevice	false {ok}
SpaceCO2MiniDevice	false {ok}
StringTest1	false {ok}
StringTest2	false {ok}

Creating a Template



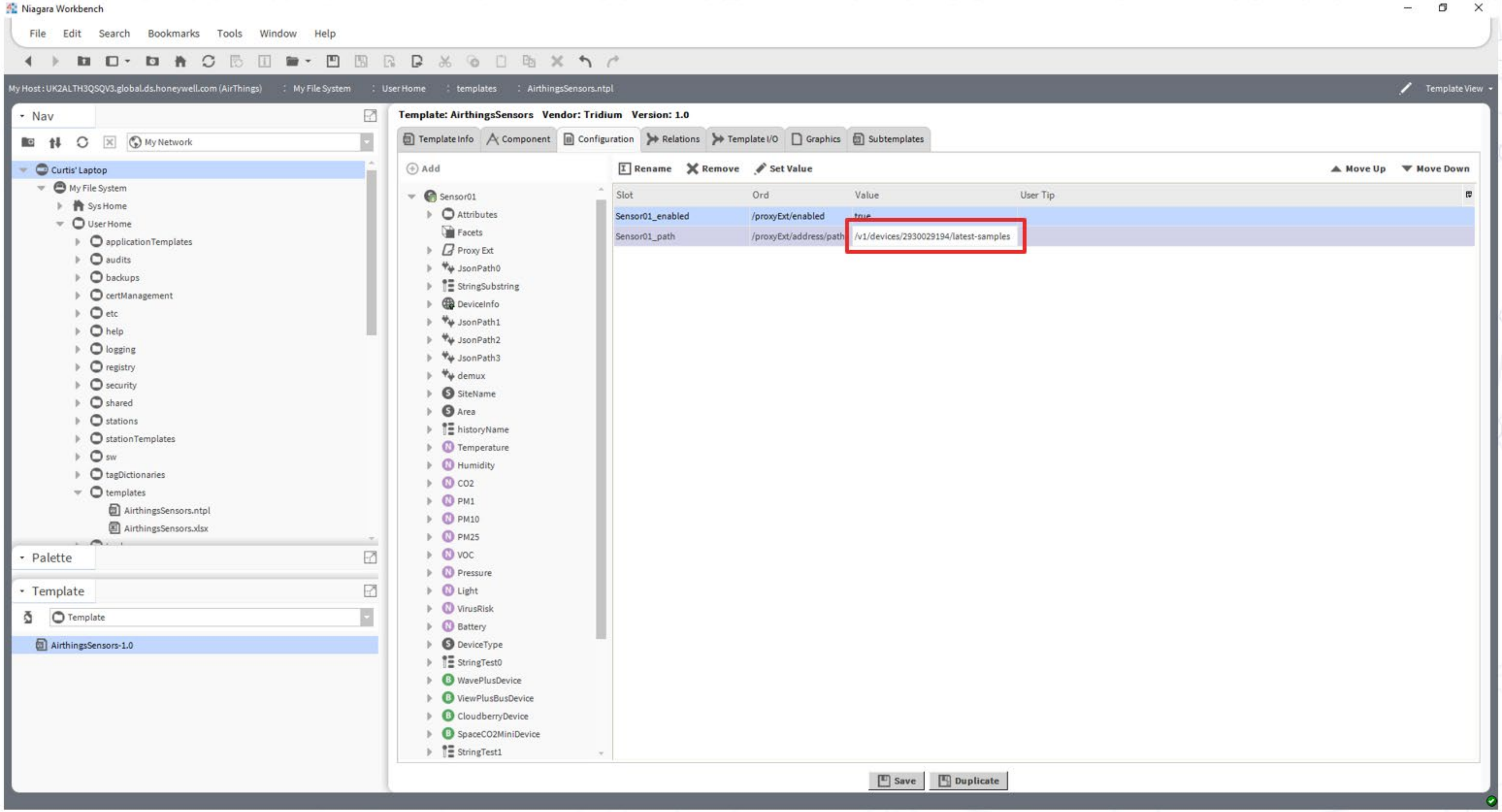
Creating a Template



Template: AirthingsSensors Vendor: Tridium Version: 1.0

Slot	Ord	Value	User Tip
Sensor01_enabled	/proxyExt/enabled	true	
Sensor01_path	/proxyExt/address/path	/v1/devices/2930029194/latest-samples	

Creating a Template

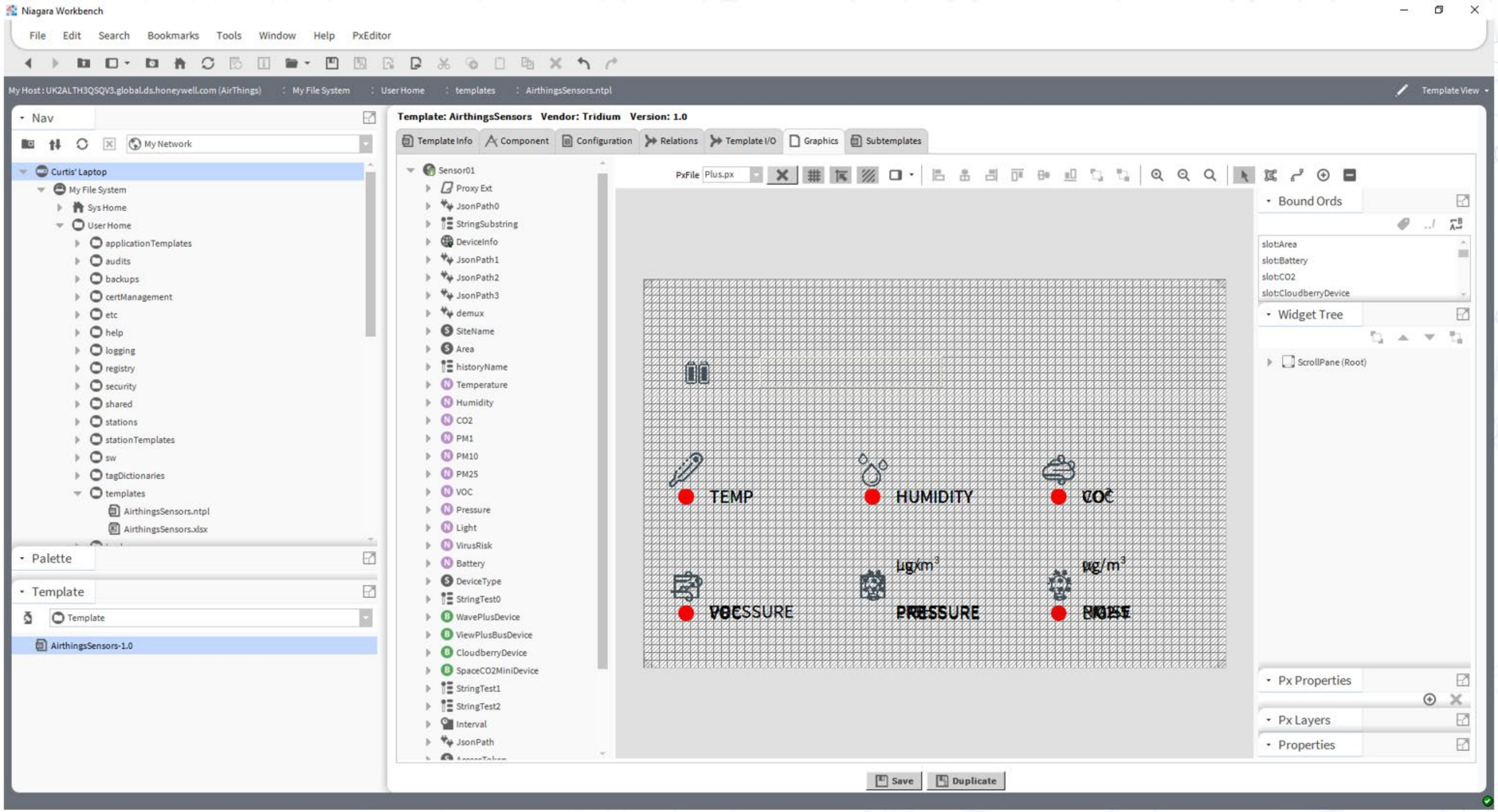


My Host : UK2ALTH3QSQV3.global.ds.honeywell.com (AirThings) : My File System : User Home : templates : AirthingsSensors.ntpl

Template: AirthingsSensors Vendor: Tridium Version: 1.0

Slot	Ord	Value	User Tip
Sensor01_enabled	/proxyExt/enabled	true	
Sensor01_path	/proxyExt/address/path	/v1/devices/2930029194/latest-samples	

Creating a Template



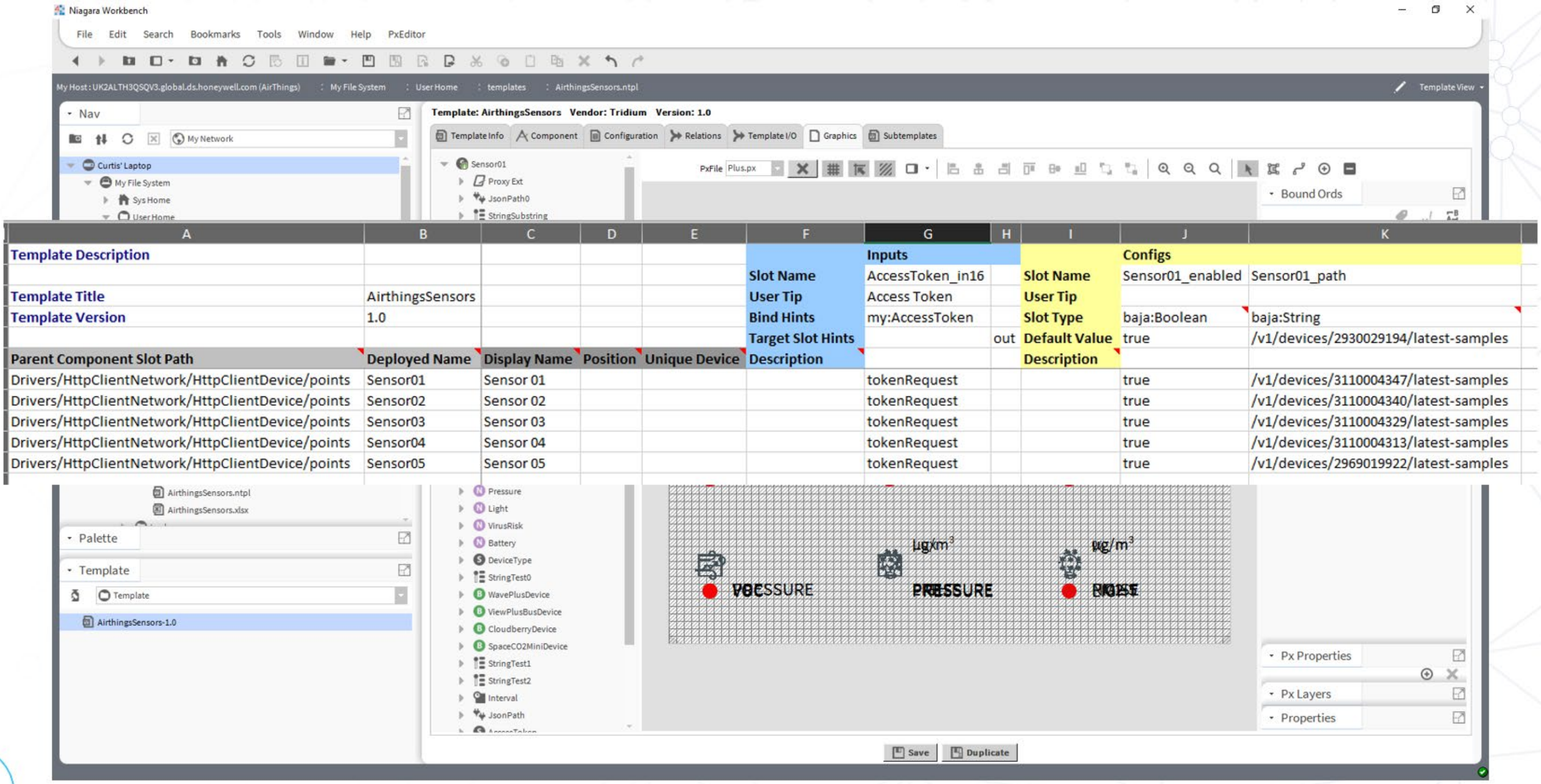
My Host : UK2ALTH3QSQV3.global.ds.honeywell.com (AirThings) : My File System : User Home : templates : AirthingsSensors.ntpl

Template: AirthingsSensors Vendor: Tridium Version: 1.0

NF 23

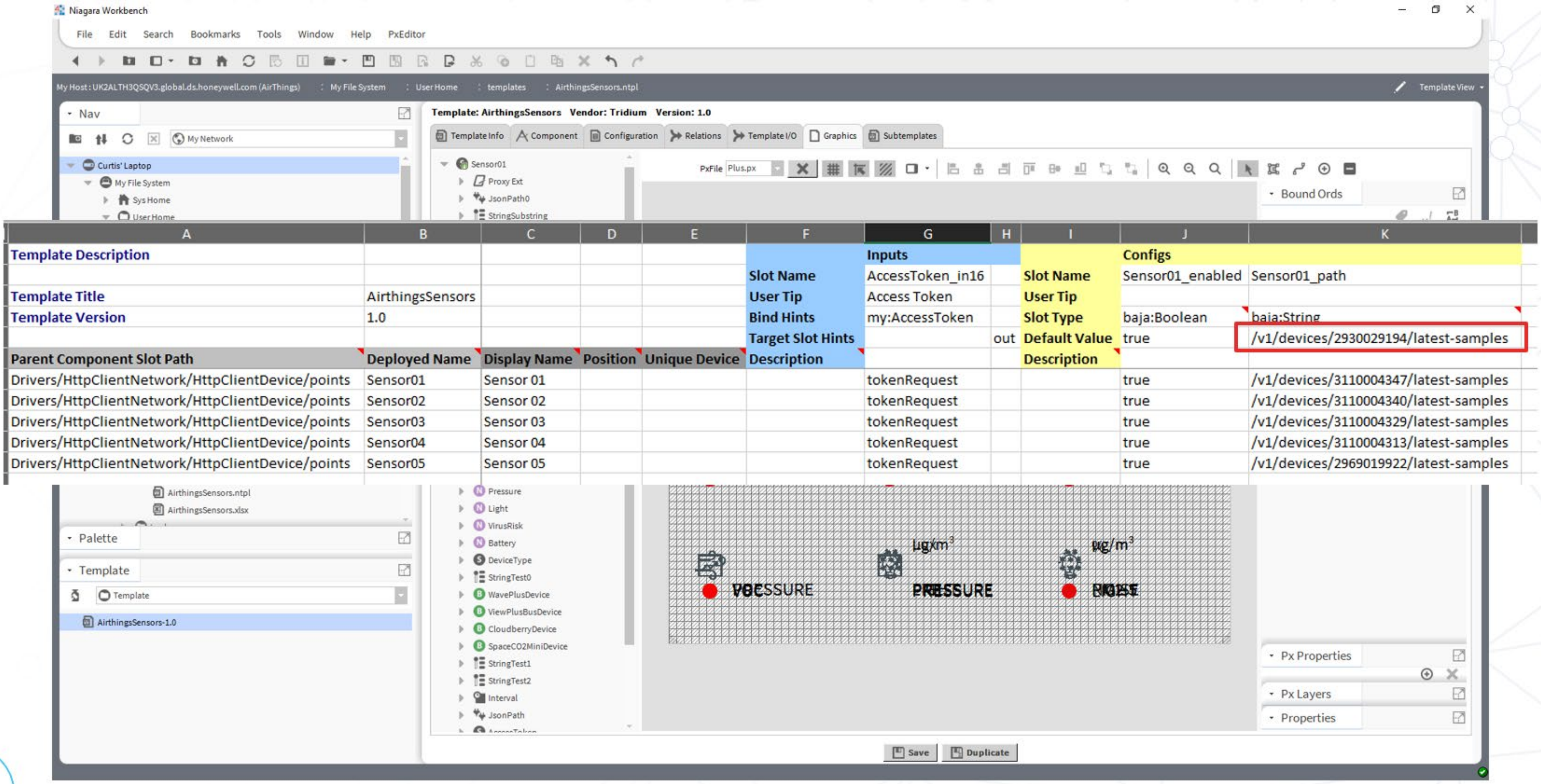
TRIDIUM

Creating a Template



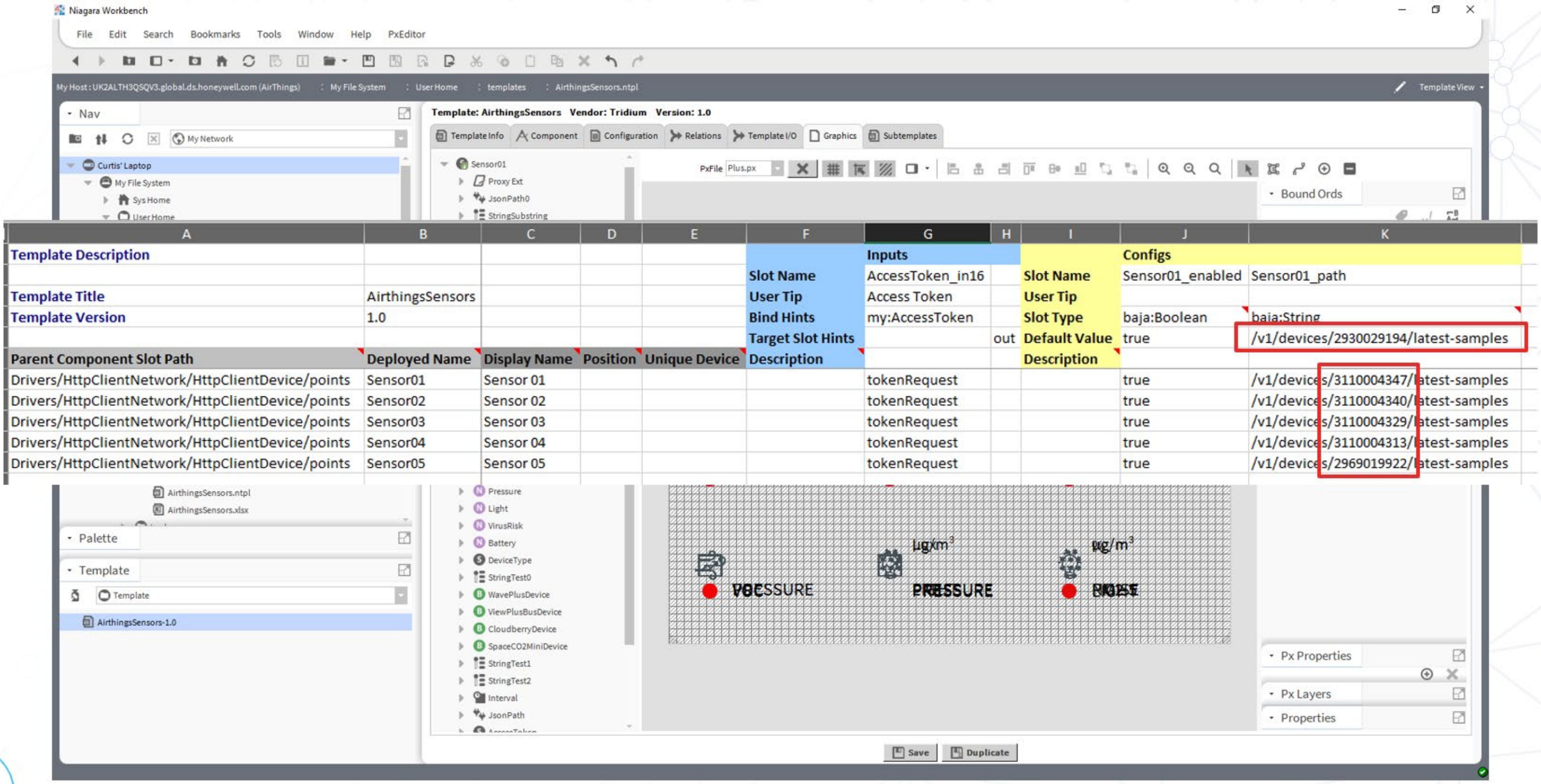
A	B	C	D	E	F	G	H	I	J	K
Template Description						Inputs			Configs	
					Slot Name	AccessToken_in16		Slot Name	Sensor01_enabled	Sensor01_path
Template Title	AirthingsSensors				User Tip	Access Token		User Tip		
Template Version	1.0				Bind Hints	my:AccessToken		Slot Type	baja:Boolean	baja:String
					Target Slot Hints		out	Default Value	true	/v1/devices/2930029194/latest-samples
Parent Component Slot Path	Deployed Name	Display Name	Position	Unique Device	Description			Description		
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor01	Sensor 01				tokenRequest			true	/v1/devices/3110004347/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor02	Sensor 02				tokenRequest			true	/v1/devices/3110004340/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor03	Sensor 03				tokenRequest			true	/v1/devices/3110004329/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor04	Sensor 04				tokenRequest			true	/v1/devices/3110004313/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor05	Sensor 05				tokenRequest			true	/v1/devices/2969019922/latest-samples

Creating a Template

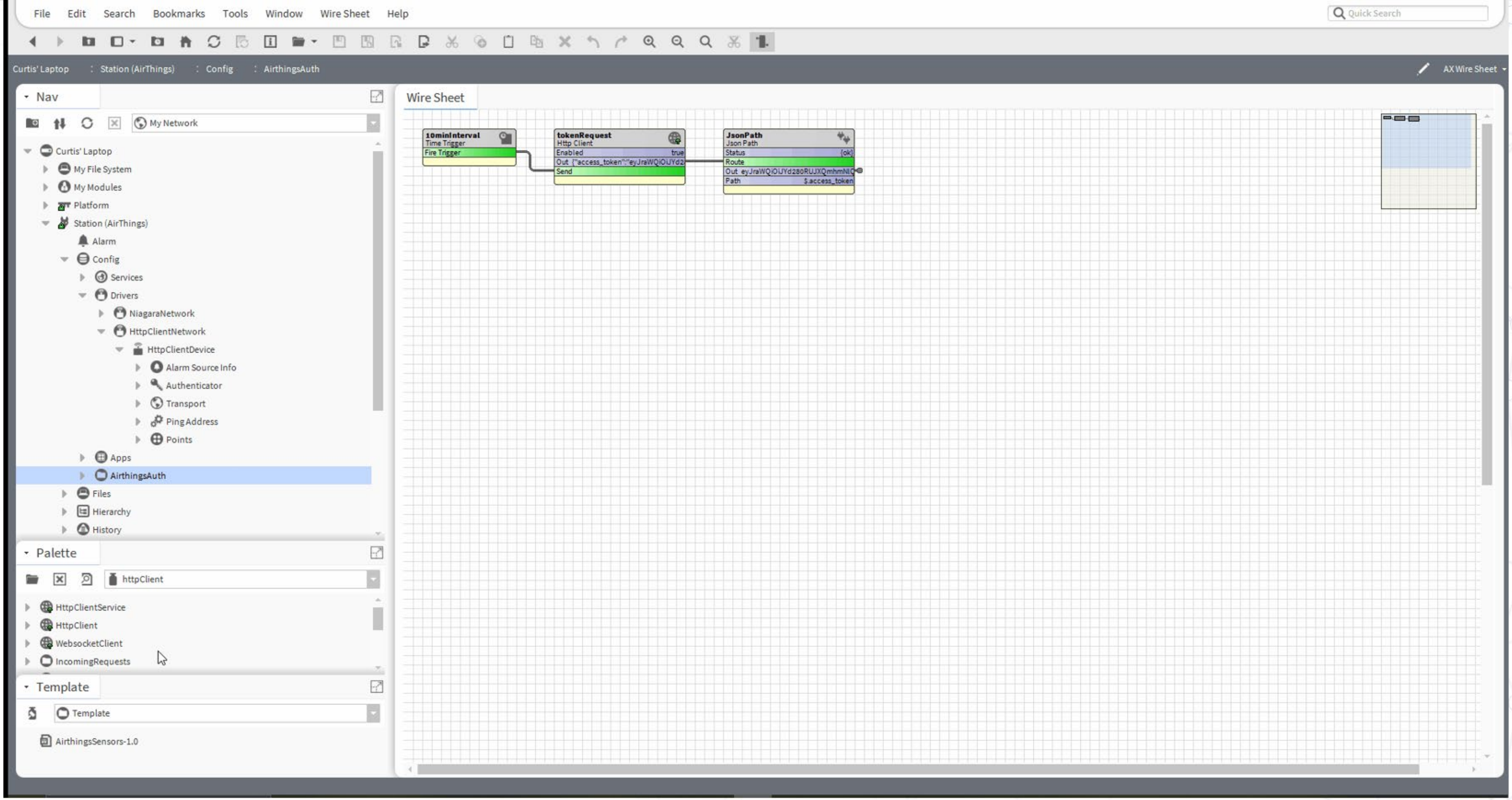


A	B	C	D	E	F	G	H	I	J	K
Template Description						Inputs			Configs	
					Slot Name	AccessToken_in16		Slot Name	Sensor01_enabled	Sensor01_path
Template Title	AirthingsSensors				User Tip	Access Token		User Tip		
Template Version	1.0				Bind Hints	my:AccessToken		Slot Type	baja:Boolean	baja:String
					Target Slot Hints		out	Default Value	true	/v1/devices/2930029194/latest-samples
Parent Component Slot Path	Deployed Name	Display Name	Position	Unique Device	Description			Description		
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor01	Sensor 01				tokenRequest			true	/v1/devices/3110004347/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor02	Sensor 02				tokenRequest			true	/v1/devices/3110004340/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor03	Sensor 03				tokenRequest			true	/v1/devices/3110004329/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor04	Sensor 04				tokenRequest			true	/v1/devices/3110004313/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor05	Sensor 05				tokenRequest			true	/v1/devices/2969019922/latest-samples

Creating a Template



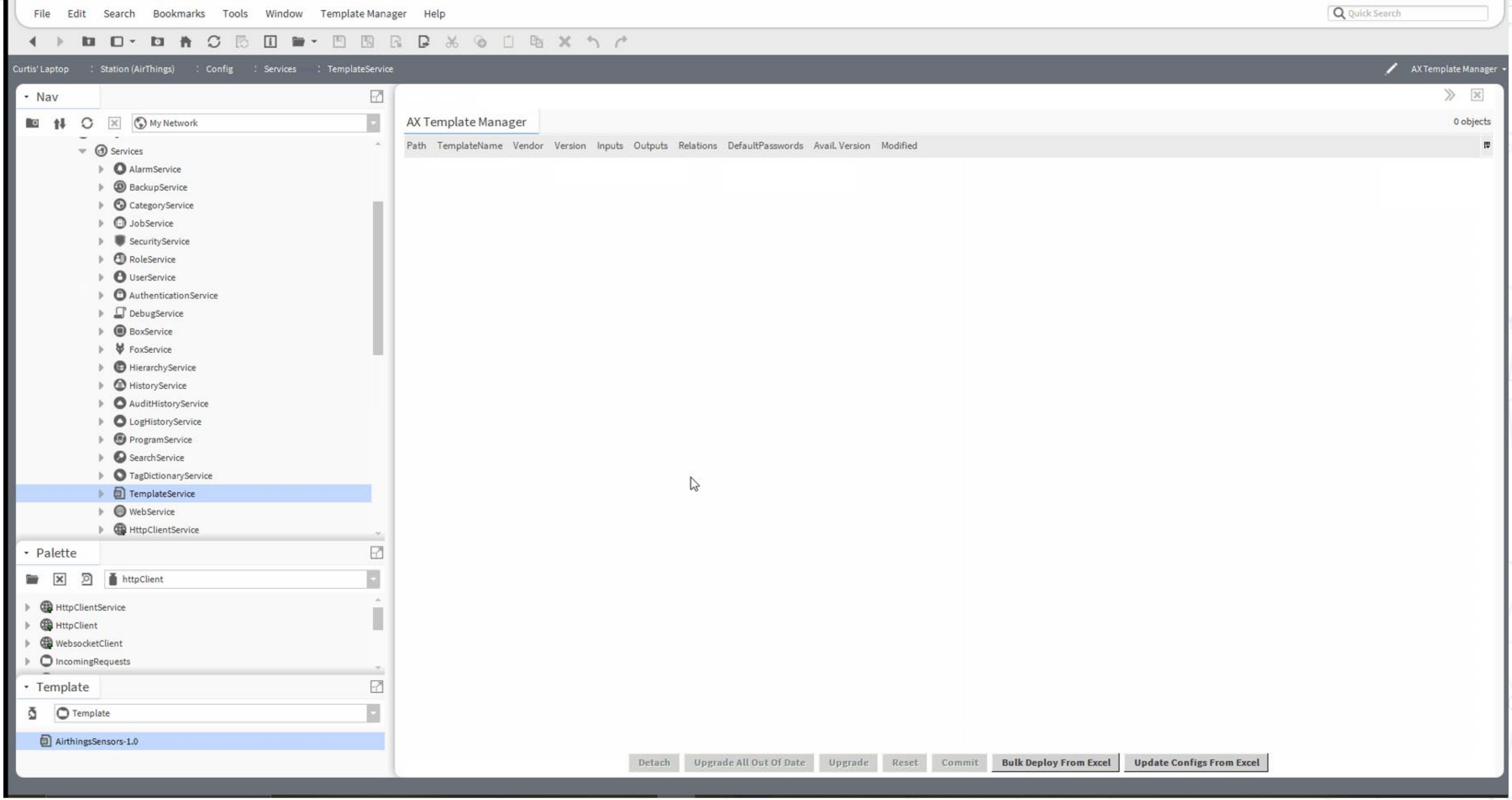
A	B	C	D	E	F	G	H	I	J	K
Template Description						Inputs			Configs	
					Slot Name	AccessToken_in16		Slot Name	Sensor01_enabled	Sensor01_path
Template Title	AirthingsSensors				User Tip	Access Token		User Tip		
Template Version	1.0				Bind Hints	my:AccessToken		Slot Type	baja:Boolean	baja:String
					Target Slot Hints		out	Default Value	true	/v1/devices/2930029194/latest-samples
Parent Component Slot Path	Deployed Name	Display Name	Position	Unique Device	Description			Description		
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor01	Sensor 01				tokenRequest			true	/v1/devices/3110004347/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor02	Sensor 02				tokenRequest			true	/v1/devices/3110004340/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor03	Sensor 03				tokenRequest			true	/v1/devices/3110004329/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor04	Sensor 04				tokenRequest			true	/v1/devices/3110004313/latest-samples
Drivers/HttpClientNetwork/HttpClientDevice/points	Sensor05	Sensor 05				tokenRequest			true	/v1/devices/2969019922/latest-samples

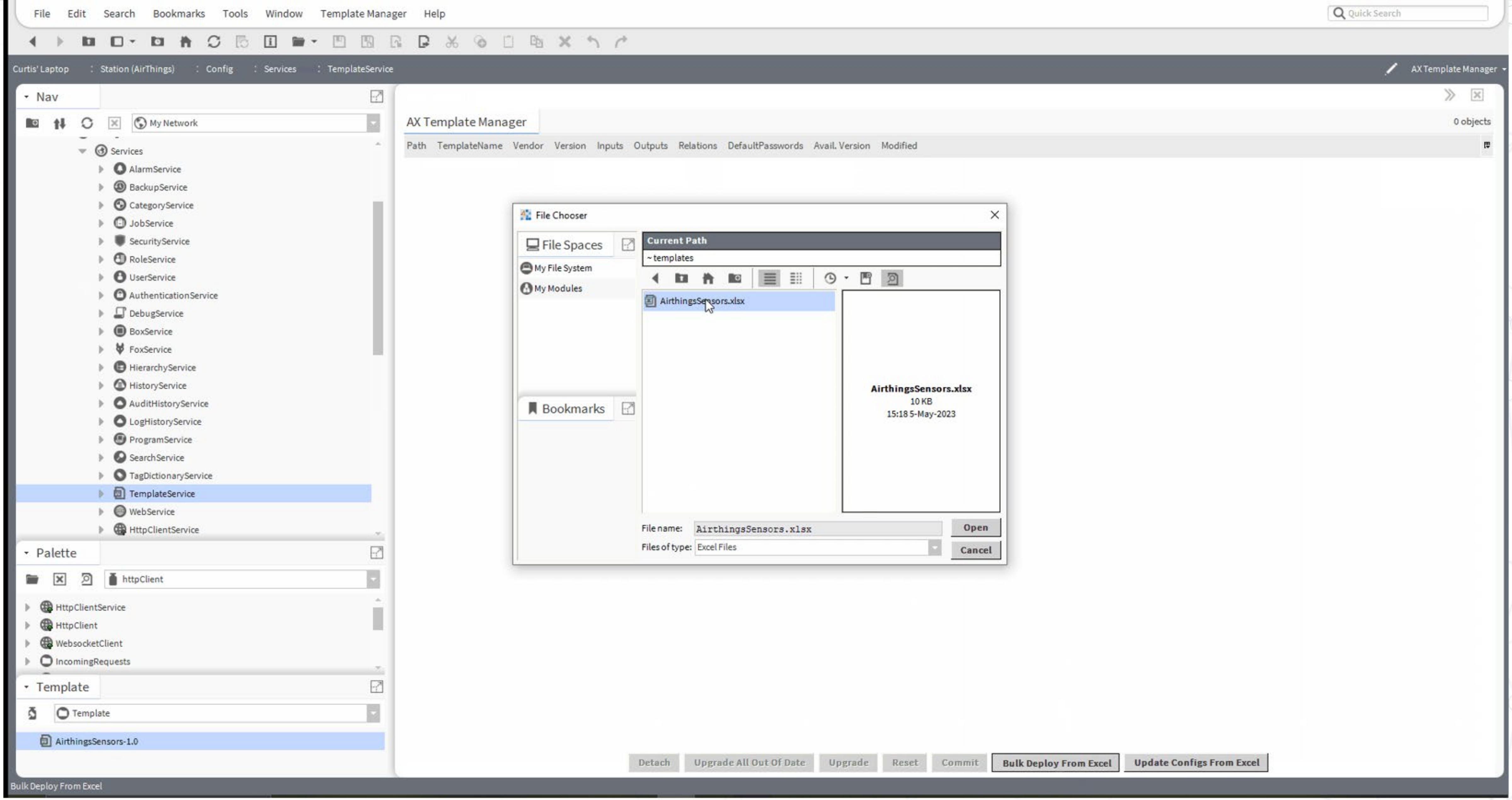


10minInterval
Time Trigger
Fire Trigger

tokenRequest
Http Client
Enabled true
Out {"access_token":"eyJraWQiOiJYd2
Send

JsonPath
Json Path
Status {ok}
Route
Out eyJraWQiOiJYd280RUJXQmhmNlC
Path $.access_token


File
Edit
Search
Bookmarks
Tools
Window
Template Manager
Help
Quick Search
Curtis' Laptop
Station (AirThings)
Config
Services
TemplateService
AX Template Manager
Nav
My Network
Services
AlarmService
BackupService
CategoryService
JobService
SecurityService
RoleService
UserService
AuthenticationService
DebugService
BoxService
FoxService
HierarchyService
HistoryService
AuditHistoryService
LogHistoryService
ProgramService
SearchService
TagDictionaryService
TemplateService
WebService
HttpClientService
Palette
httpClient
HttpClientService
HttpClient
WebsocketClient
IncomingRequests
Template
Template
AirthingsSensors-1.0
AX Template Manager
0 objects
Path
TemplateName
Vendor
Version
Inputs
Outputs
Relations
DefaultPasswords
Avail. Version
Modified
Detach
Upgrade All Out Of Date
Upgrade
Reset
Commit
Bulk Deploy From Excel
Update Configs From Excel

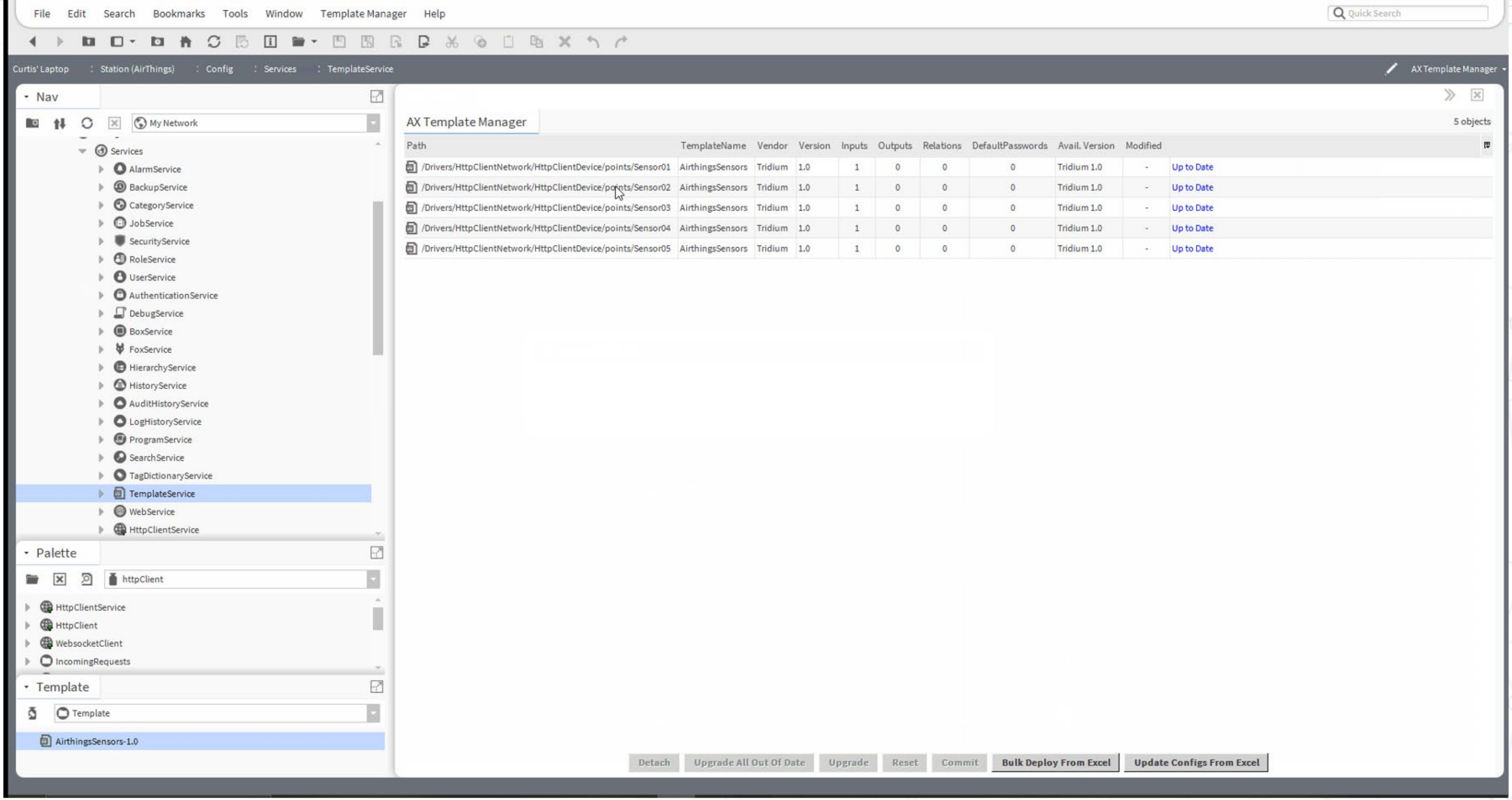


Current Path: ~ templates

AirthingsSensors.xlsx
10 KB
15:18 5-May-2023

File name: AirthingsSensors.xlsx

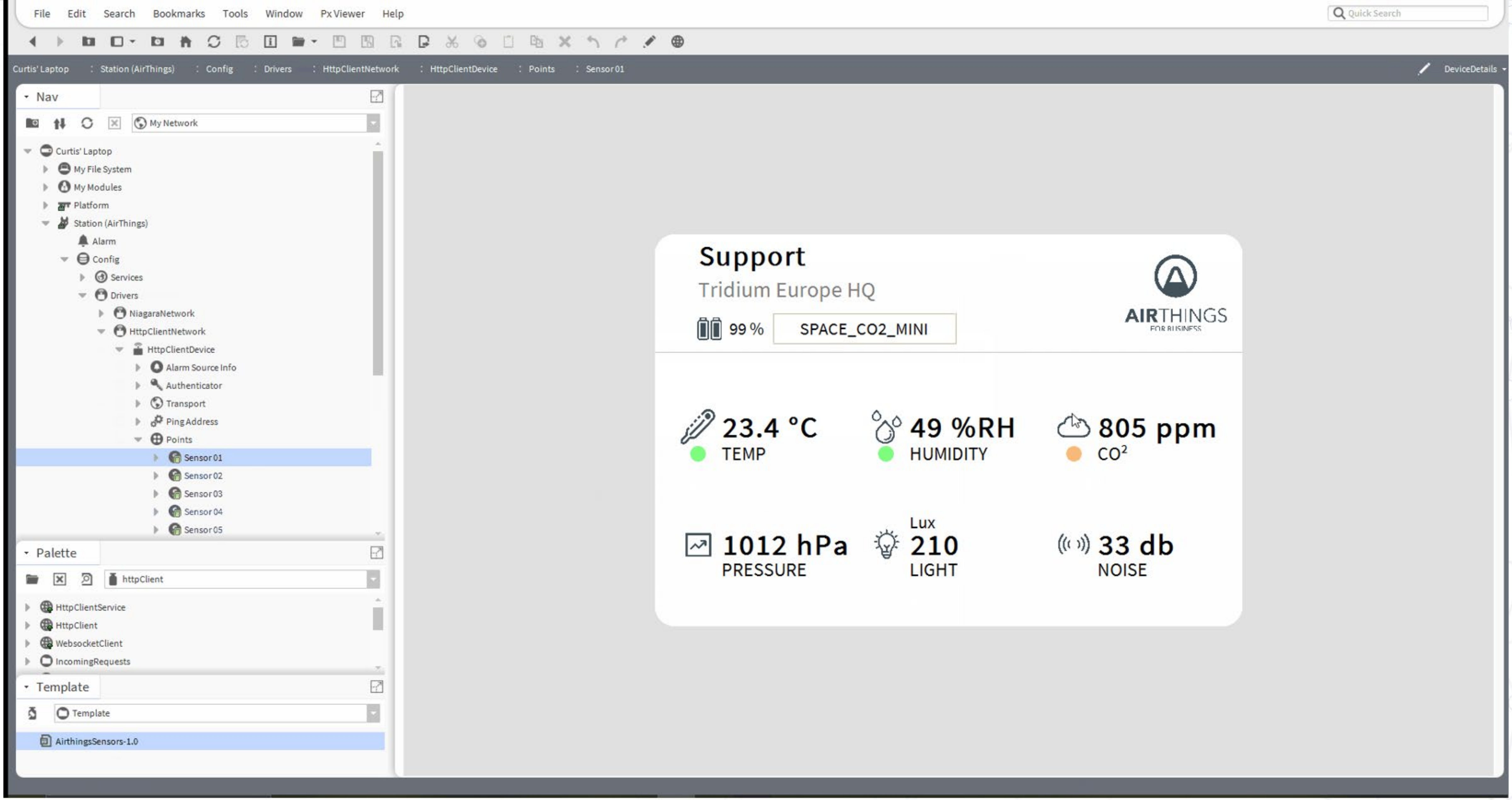
Files of type: Excel Files



AX Template Manager

5 objects

Path	TemplateName	Vendor	Version	Inputs	Outputs	Relations	DefaultPasswords	Avail. Version	Modified	
/Drivers/HttpClientNetwork/HttpClientDevice/points/Sensor01	AirthingsSensors	Tridium	1.0	1	0	0	0	Tridium 1.0	-	Up to Date
/Drivers/HttpClientNetwork/HttpClientDevice/points/Sensor02	AirthingsSensors	Tridium	1.0	1	0	0	0	Tridium 1.0	-	Up to Date
/Drivers/HttpClientNetwork/HttpClientDevice/points/Sensor03	AirthingsSensors	Tridium	1.0	1	0	0	0	Tridium 1.0	-	Up to Date
/Drivers/HttpClientNetwork/HttpClientDevice/points/Sensor04	AirthingsSensors	Tridium	1.0	1	0	0	0	Tridium 1.0	-	Up to Date
/Drivers/HttpClientNetwork/HttpClientDevice/points/Sensor05	AirthingsSensors	Tridium	1.0	1	0	0	0	Tridium 1.0	-	Up to Date



Support

Tridium Europe HQ

99 % SPACE_CO2_MINI

AIRTHINGS FOR BUSINESS

23.4 °C ● TEMP	49 %RH ● HUMIDITY	805 ppm ● CO²
1012 hPa PRESSURE	Lux 210 LIGHT	33 db NOISE

Demo Summary

- RTFM – Read The Full Manual (documentation is essential)

- Authorisation was achieved utilising a HttpClient Component with a single GET request

- Using the Bulk Deployment tool we were able to deploy 5 sensors in seconds, complete with points and Px Views

Licensing Requirements

- Free to evaluate with demo license
- Production use requires additional license and active SMA
- Each HTTP Client component counts as one global point
- Each HTTP Driver point counts as one global point

NF 23

CONNECTING THE WORLD

MQTT – Making Connections

Nick Dodd

Tridium

The Natural Order

1. Anything that is in the world when you're born is normal and ordinary and is just a natural part of the way the world works.

2. Anything that's invented between when you're fifteen and thirty-five is new and exciting and revolutionary and you can probably get a career in it.

3. **Anything invented after you're thirty-five is against the natural order of things.**

Douglas Adams - *How to Stop Worrying and Learn to Love the Internet*

The New Natural Order?

ChatGPT for some....

Transition from traditional Client-Server protocols, I/O, Connected Bus, to....

HTTP / AMQP / 6LoWPAN / **MQTT** / Mesh Topologies / Cloud / **Certificate Management**

MQTT

MQ Telemetry Transport

Knowledge Refresh

Central Broker

Clients (*Client ID*)

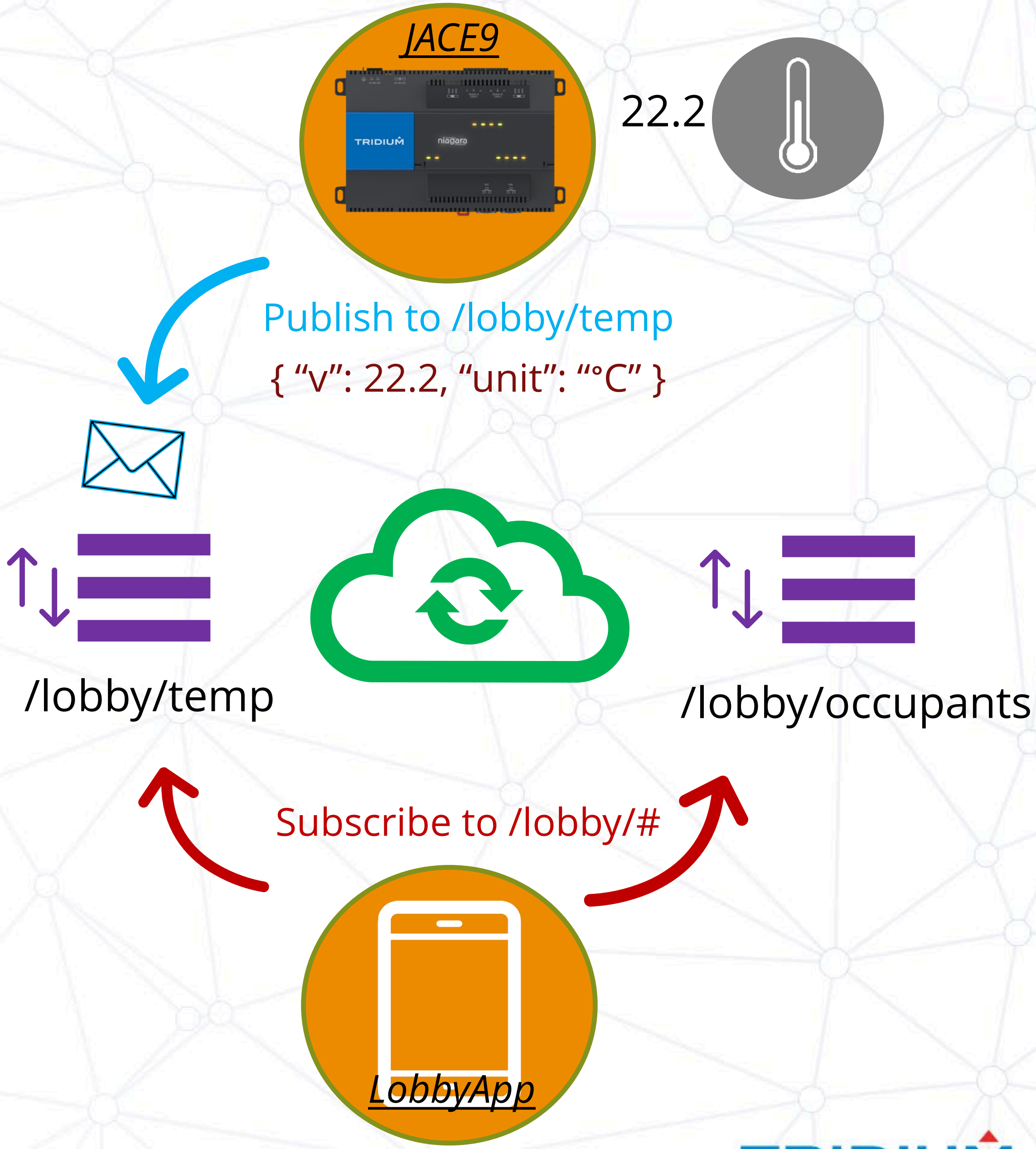
Message Queues (Topics)

Publish / Subscribe

Message payload / JSON

Retained messages

Quality of Service levels 0 - 2



Why this is a good thing

Small code footprint

Highly scalable

Clients never talk directly

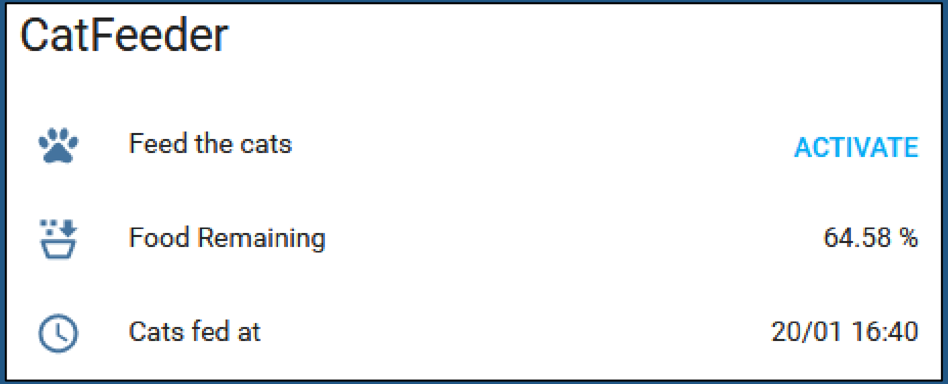
10001101
11

No message protocol

Use Case 1 – The Enthusiast

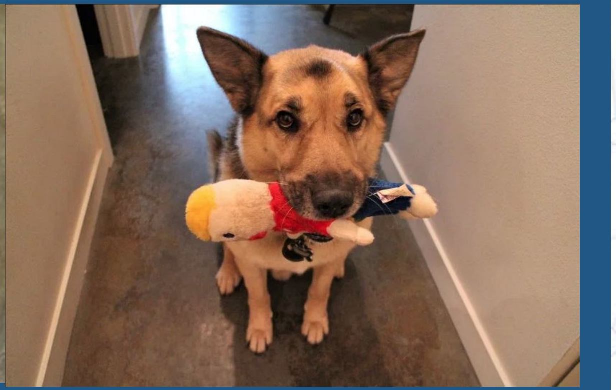
Homebrew HVAC anyone?

- Cat Feeding App https://github.com/lance36/catFeeder
- LiDAR Mailbox Monitor https://hackaday.com/2022/06/06/using-a-lidar-sensor-to-monitor-your-mailbox
- BarkBack Automatic Dog Sushing https://hackaday.com/2018/01/05/bark-back-iot-pet-monitor



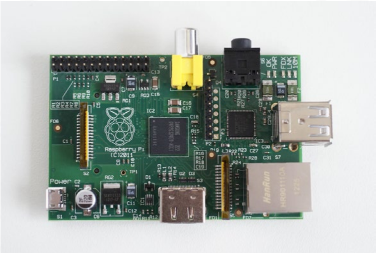
CatFeeder

Feed the cats	ACTIVATE
Food Remaining	64.58 %
Cats fed at	20/01 16:40



Needs

- Sensors, Arduinos, Raspberry Pi’s, Alexa, Soldering Iron
- Broker / IoT platform – NodeRed, Home Assistant, Dozens more
- Possibly: Coding Skills

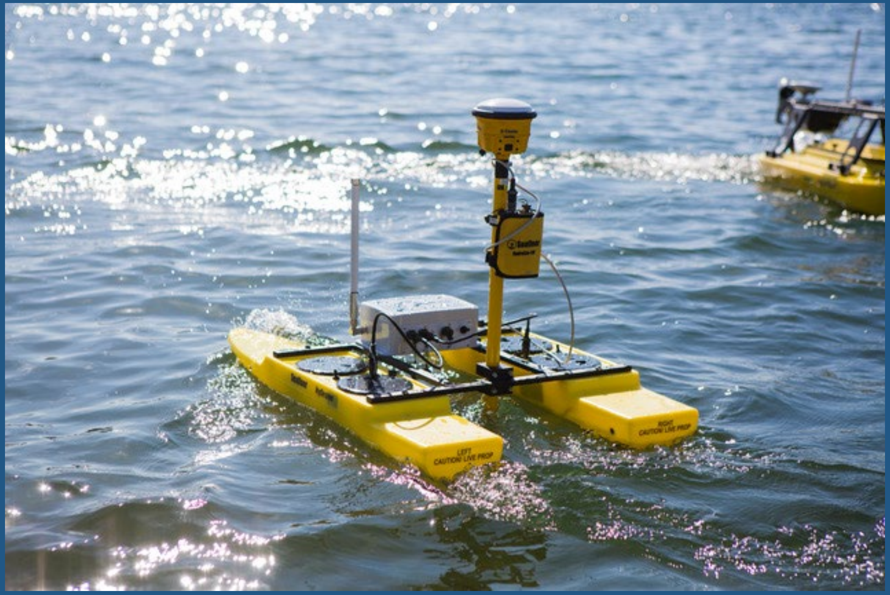


Use Case 2 – The IoT Professional

Seafloor Systems

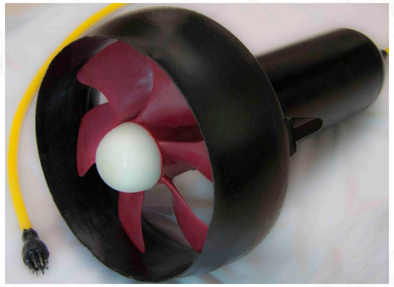
- Hydrographic Robot Boats
- Survey Seabed
- Crew = 0. Remotely operated
- Deploy updates to robots via AWS MQTT

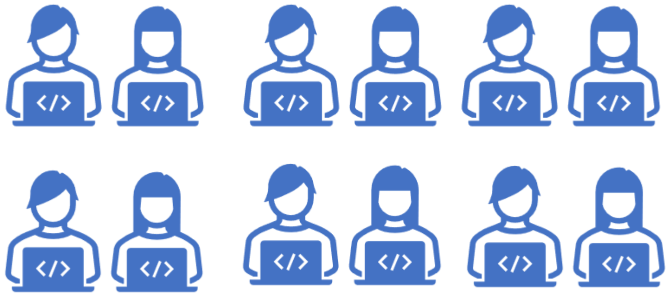
https://aws.amazon.com/solutions



Needs

- Sonar, thrusters, lamps, batteries
- Roboticists
- Soldering Irons
- Highly skilled engineers, roboticists, development team

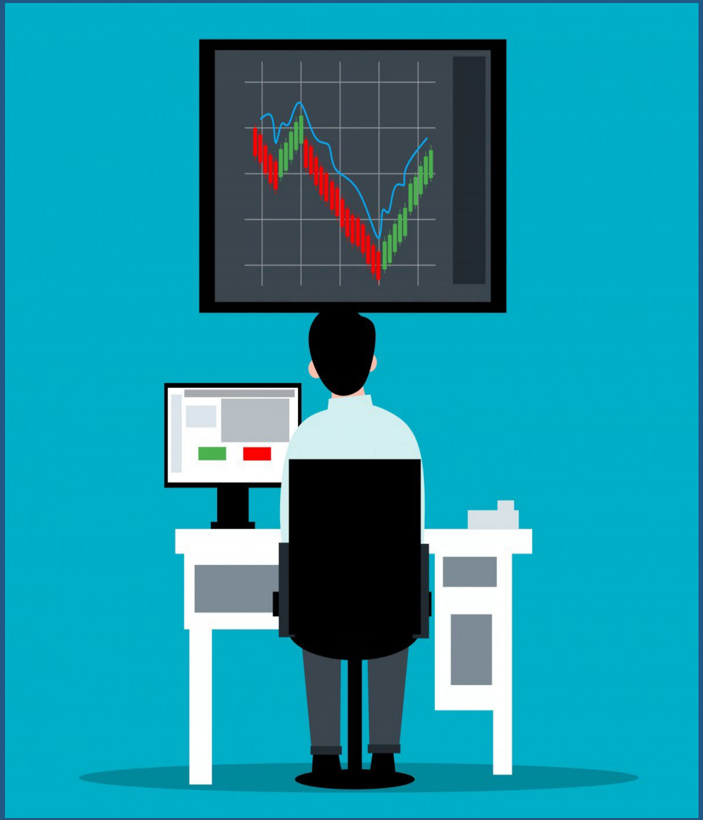




Use Case 3 – The BMS Engineer

Push Point / History / Alarm Data to a broker

Analytic Insights - *Splunk*
Archive – *Amazon S3*
Dashboards / Wall displays – *Grafana*
Performance Monitoring
Predictive Maintenance
SMS Events - *Twilio*

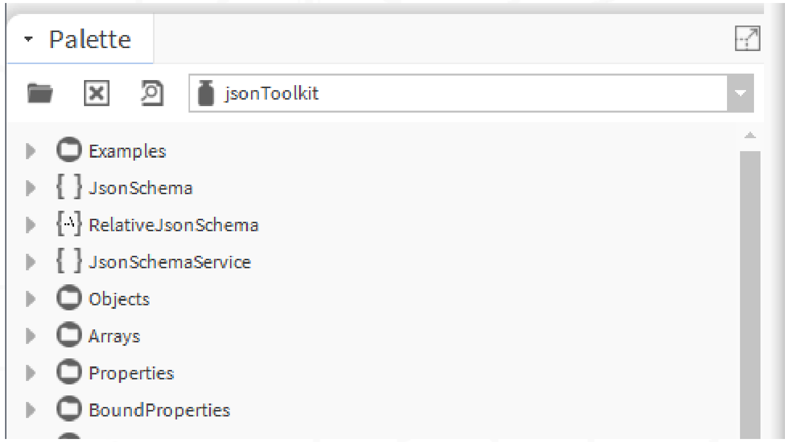


Bring data *into* the station

Sensor Telemetry – *EnOcean IoT*
Parking space data
Smart Meters
People Counters
Remote command and control

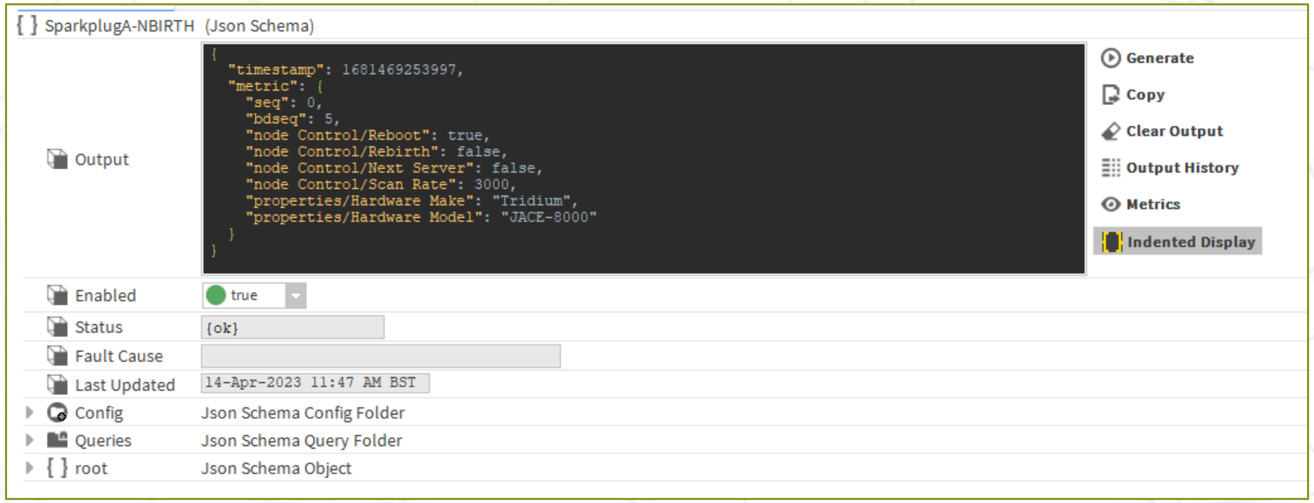
Needs

- Niagara Data Service + Developers or…
- A Niagara Station
- Workbench / Browser
- The MQTT / JSONToolkit modules



jsonToolkit

- Examples
- JsonSchema
- RelativeJsonSchema
- JsonSchemaService
- Objects
- Arrays
- Properties
- BoundProperties



{ } SparkplugA-NBIRTH (Json Schema)

Output

```
{
  "timestamp": 1681469253997,
  "metric": {
    "seq": 0,
    "bdseq": 5,
    "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"
  }
}
```

Enabled	● true
Status	{ok}
Fault Cause	
Last Updated	14-Apr-2023 11:47 AM BST
Config	Json Schema Config Folder
Queries	Json Schema Query Folder
{ } root	Json Schema Object

Connecting to the broker

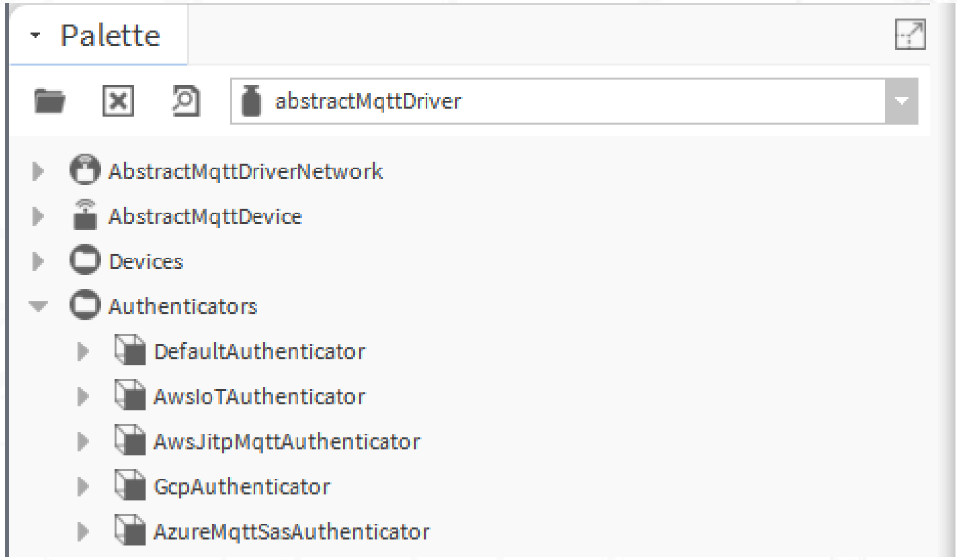
Taking the first step

“It's the job that's never started as takes longest to finish.”
JRR Tolkien

Taking the first step

Add MQTT Network…
Add MQTT Device…

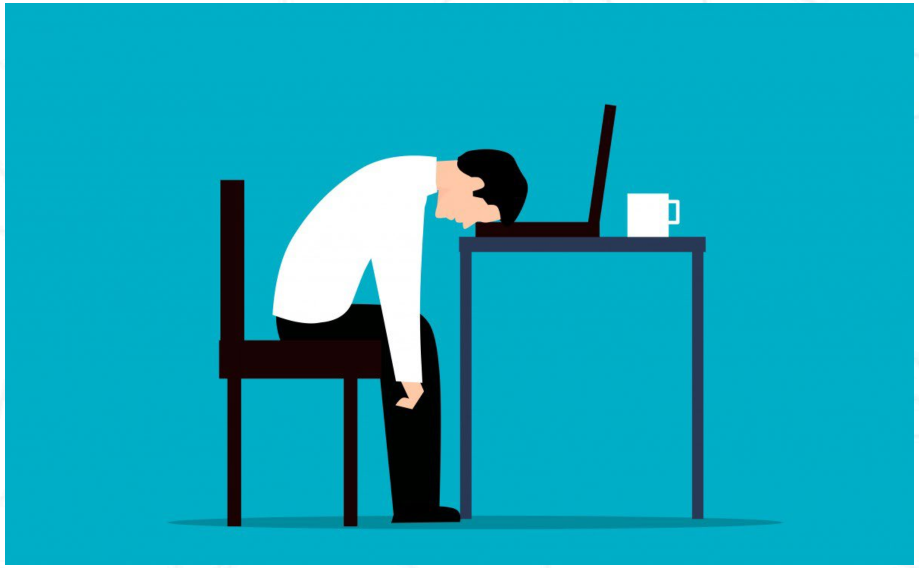
Now I need to authenticate with the broker.



Palette

abstractMqttDriver

- AbstractMqttDriverNetwork
- AbstractMqttDevice
- Devices
- Authenticators
  - DefaultAuthenticator
  - AwsIoTAuthenticator
  - AwsJitpMqttAuthenticator
  - GcpAuthenticator
  - AzureMqttSasAuthenticator



Different ways to authenticate

Anonymous
Testing Only!

Username/Password
Credentials

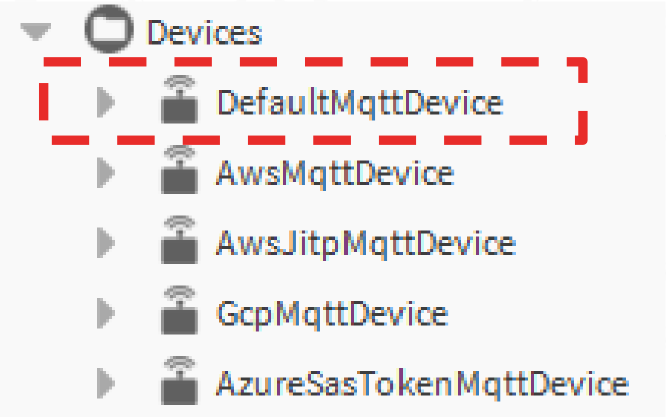
Certificates

Cloud
Specific

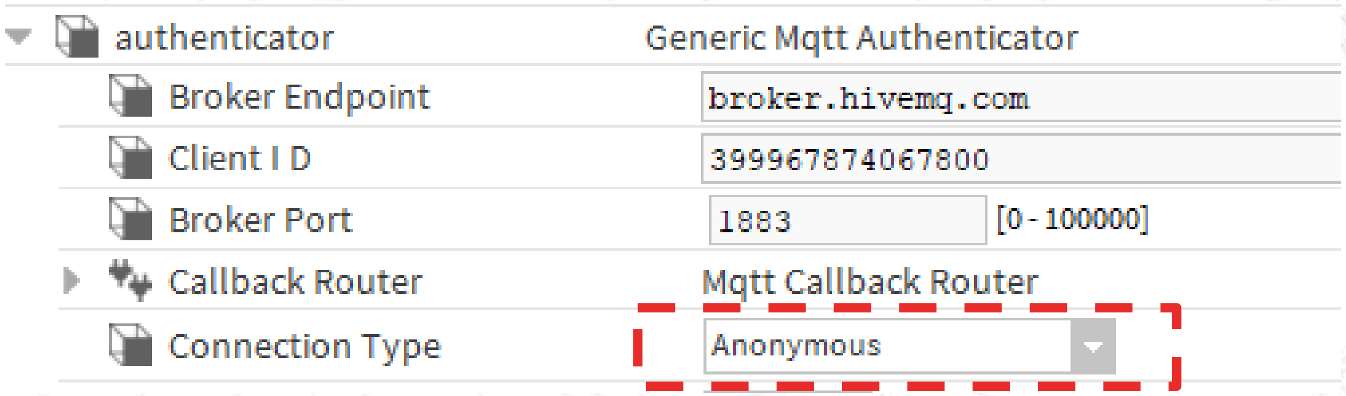
Anonymous

For testing

https://www.hivemq.com/public-mqtt-broker/



- Devices
  - DefaultMqttDevice
  - AwsMqttDevice
  - AwsJitpMqttDevice
  - GcpMqttDevice
  - AzureSasTokenMqttDevice

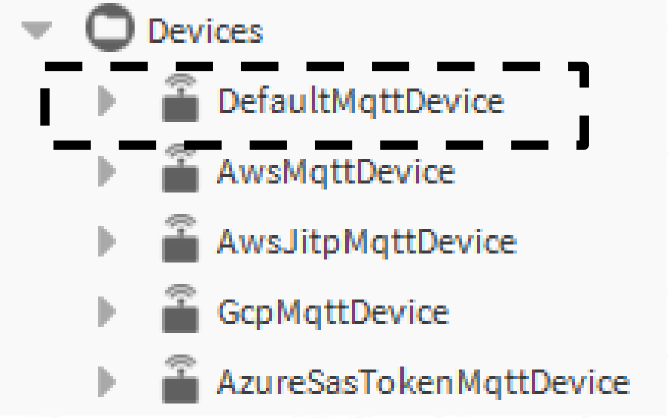


authenticator	Generic Mqtt Authenticator
Broker Endpoint	broker.hivemq.com
Client I D	399967874067800
Broker Port	1883 [0 - 100000]
Callback Router	Mqtt Callback Router
Connection Type	Anonymous

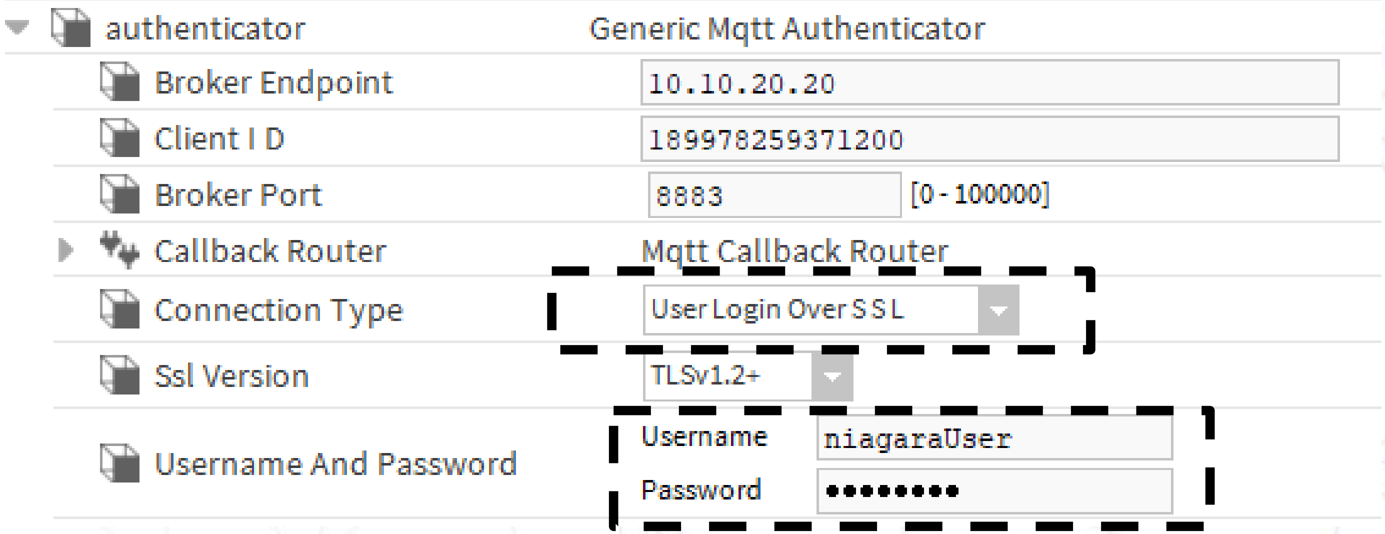
User Credentials

HiveMQ / Mosquitto / NanoMQ / VerneMQ / CloudMQTT

Credentials always sent over SSL



- Devices
  - DefaultMqttDevice
  - AwsMqttDevice
  - AwsJitpMqttDevice
  - GcpMqttDevice
  - AzureSasTokenMqttDevice

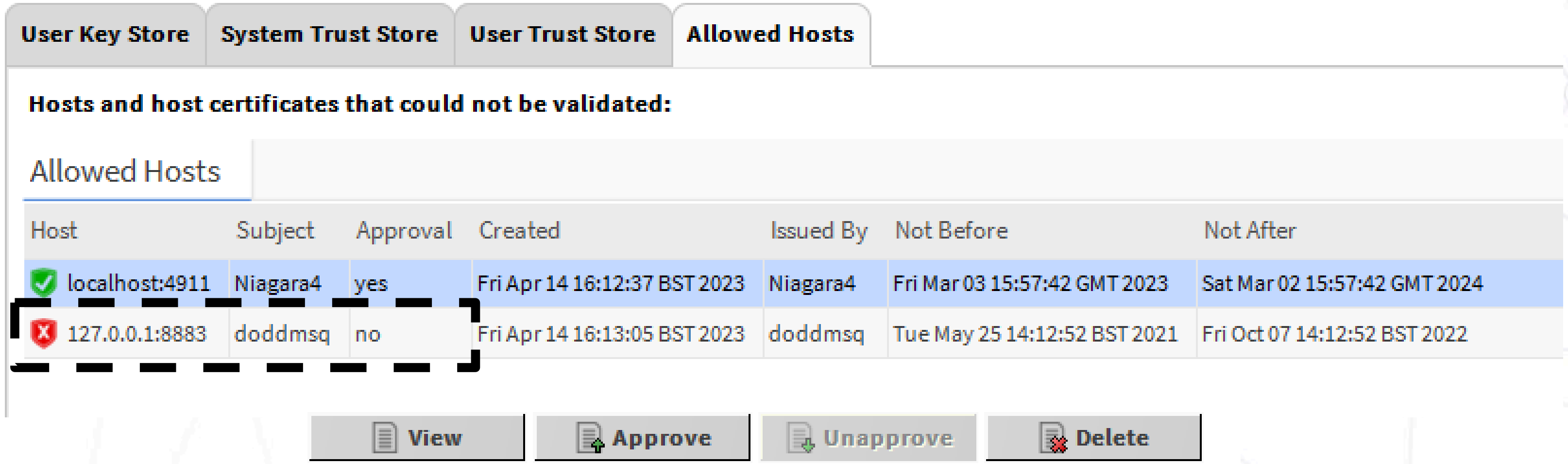


authenticator	Generic Mqtt Authenticator
Broker Endpoint	10.10.20.20
Client I D	189978259371200
Broker Port	8883 [0 - 100000]
Callback Router	Mqtt Callback Router
Connection Type	User Login Over SSL
Ssl Version	TLSv1.2+
Username And Password	Username niagaraUser Password •••••••••

Accept the Brokers certificate

Server certificate may not be automatically trusted.

Import the CA certificate or Approve in Allowed Hosts



Hosts and host certificates that could not be validated:

Allowed Hosts

Host	Subject	Approval	Created	Issued By	Not Before	Not After
✓ localhost:4911	Niagara4	yes	Fri Apr 14 16:12:37 BST 2023	Niagara4	Fri Mar 03 15:57:42 GMT 2023	Sat Mar 02 15:57:42 GMT 2024
✗ 127.0.0.1:8883	doddmsq	no	Fri Apr 14 16:13:05 BST 2023	doddmsq	Tue May 25 14:12:52 BST 2021	Fri Oct 07 14:12:52 BST 2022

Brief segue

SSL / Certificates

Knowledge Refresh 2: X509

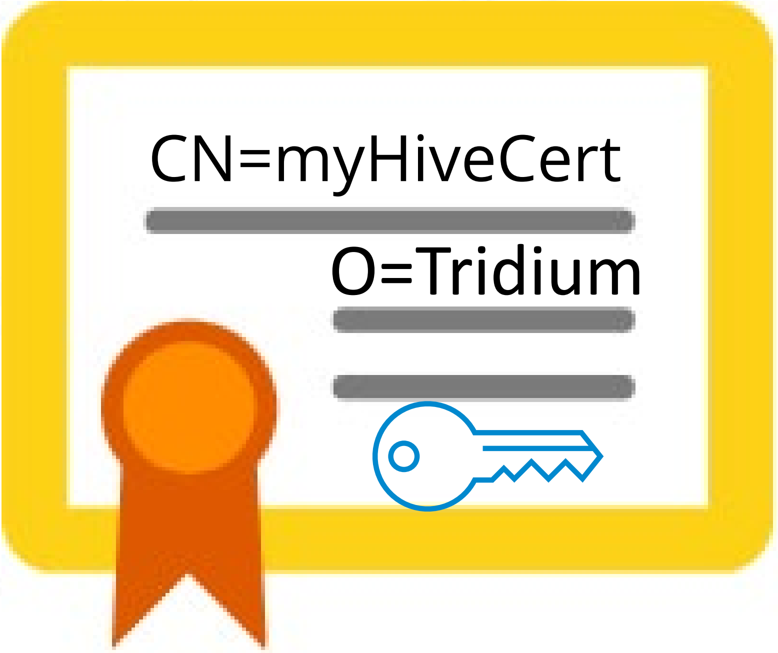
Public 🔑 🔑 Private

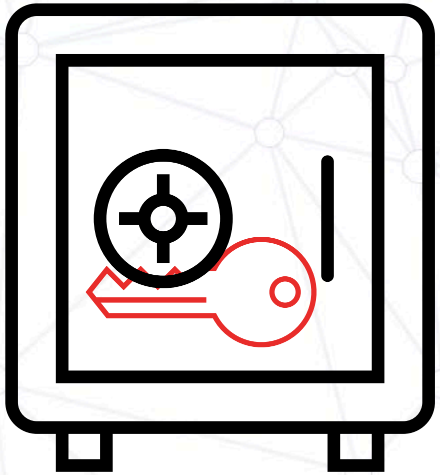
Certificates have a public and private key

Hello… 🔑 …4f7a22a!5fc… 🔑 …Hello

What one key encrypts, only the other can decrypt

Knowledge Refresh 2: X509

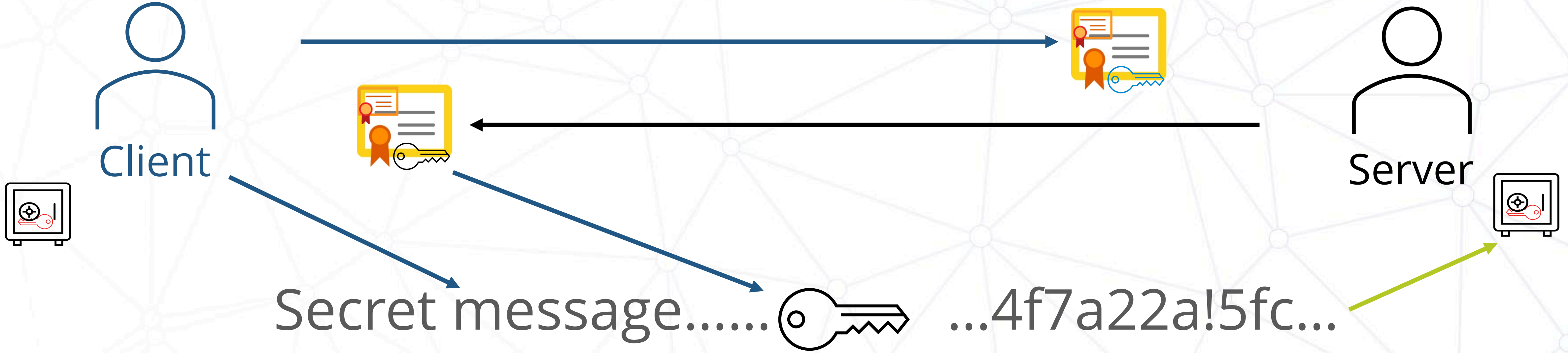




A public certificate typically contains some identifying fields + public key. Private key is kept secret

Knowledge Refresh 2: X509

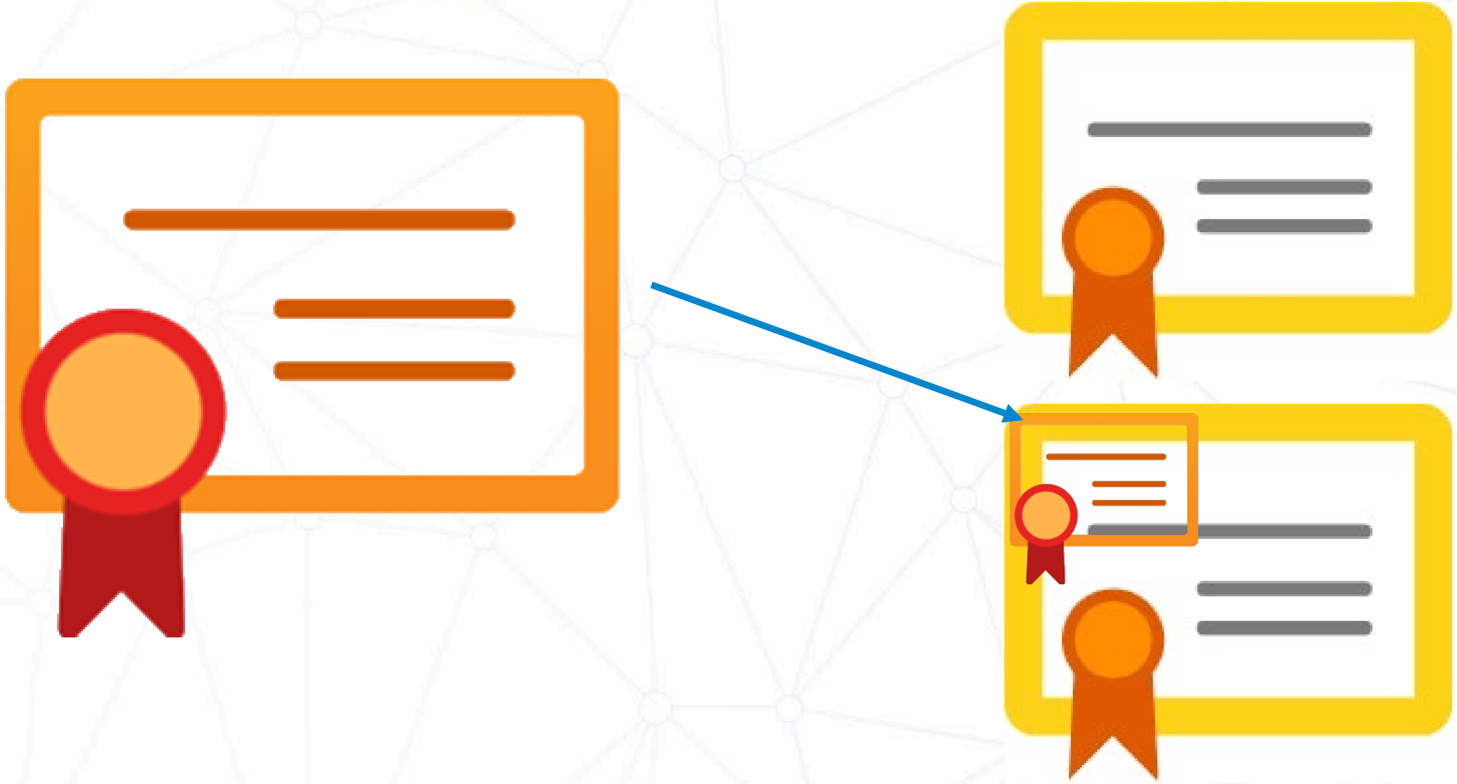
SSL is an exchange of certificates (handshake), establishing trust



Encrypt with the others public key, then decrypt with their private key

Knowledge Refresh 2: X509

A Certificate Authority (CA) can sign your certificate, proving you own that certificate



Self Signed

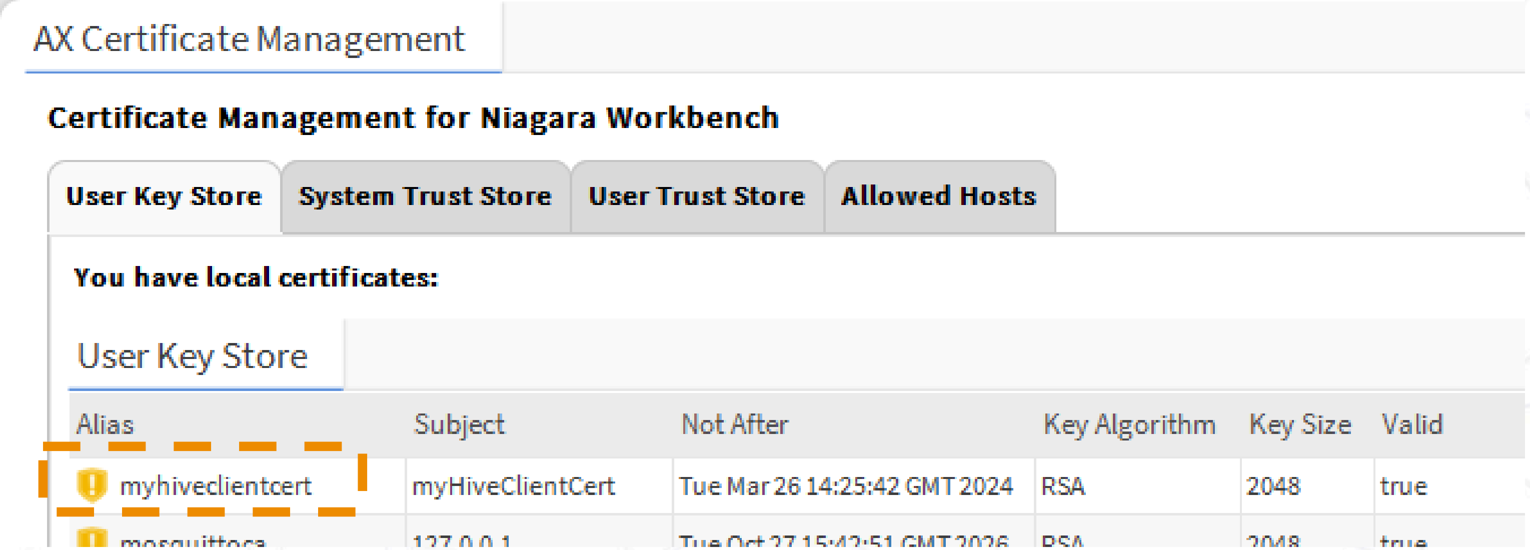
CA Signed

Client Certificate Auth

Coming in 4.14, Certificate is your credential (transport layer)

Use the Certificate Manager to generate a new certificate

Or Import Client Certificate and Private Key



AX Certificate Management

Certificate Management for Niagara Workbench

You have local certificates:

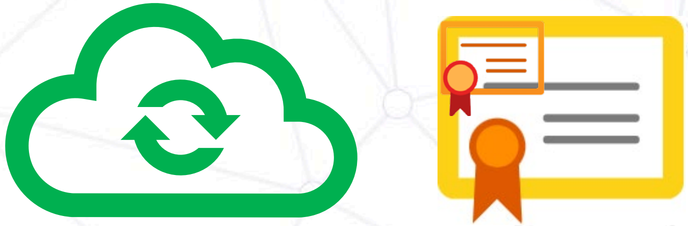
User Key Store

Alias	Subject	Not After	Key Algorithm	Key Size	Valid
myhiveclientcert	myHiveClientCert	Tue Mar 26 14:25:42 GMT 2024	RSA	2048	true
mosquittoca	127.0.0.1	Tue Oct 27 15:42:51 GMT 2026	RSA	2048	true

Broker setup

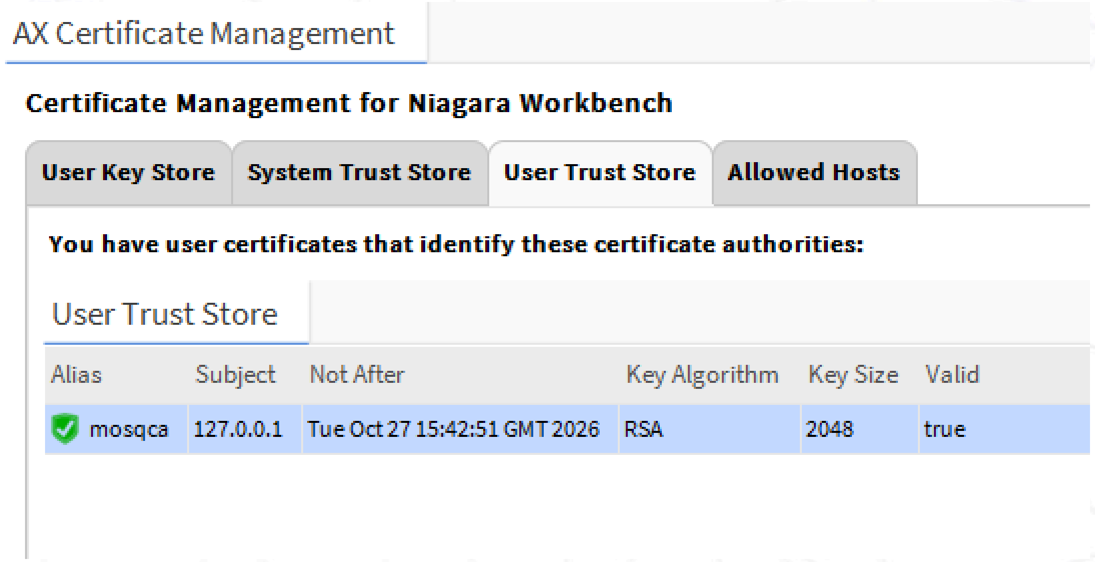
If self signed, broker will need to trust your certificate in advance

If CA signed, broker just needs to trust the CA



Client Certificate Auth

You may import the Servers Certificate or CA into Trust store



AX Certificate Management

Certificate Management for Niagara Workbench

You have user certificates that identify these certificate authorities:

User Trust Store

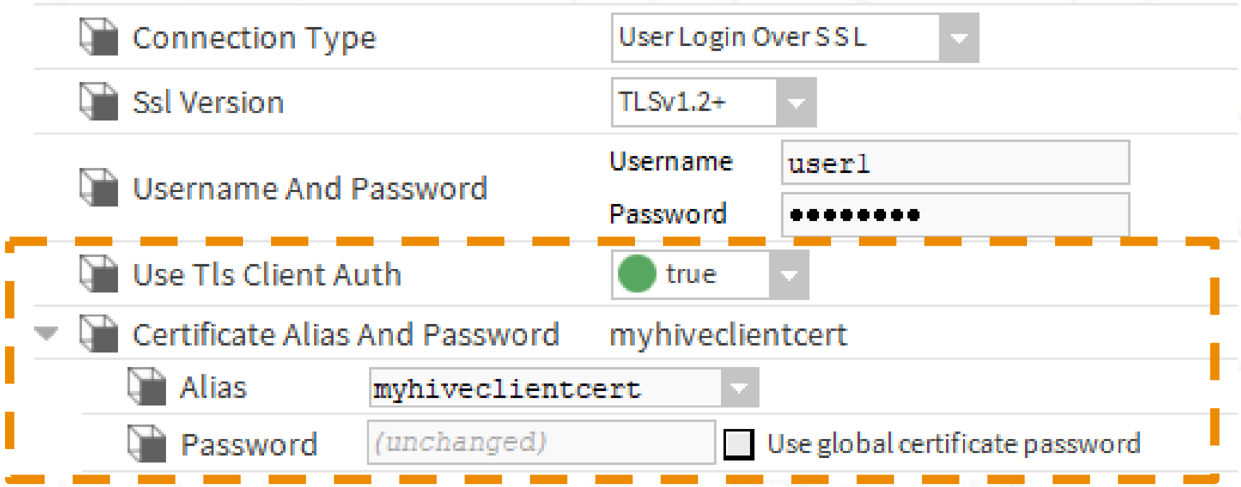
Alias	Subject	Not After	Key Algorithm	Key Size	Valid
✓ mosqca	127.0.0.1	Tue Oct 27 15:42:51 GMT 2026	RSA	2048	true

Client Certificate Auth

Select alias in default authenticator

May use Anonymous Over SSL

Or combine this with username/password (app layer)



Connection Type	User Login Over SSL
Ssl Version	TLSv1.2+
Username And Password	Username: user1 Password: ●●●●●●●●●
Use Tls Client Auth	● true
Certificate Alias And Password	myhiveclientcert
Alias	myhiveclientcert
Password	(unchanged) □ Use global certificate password

The Cloud

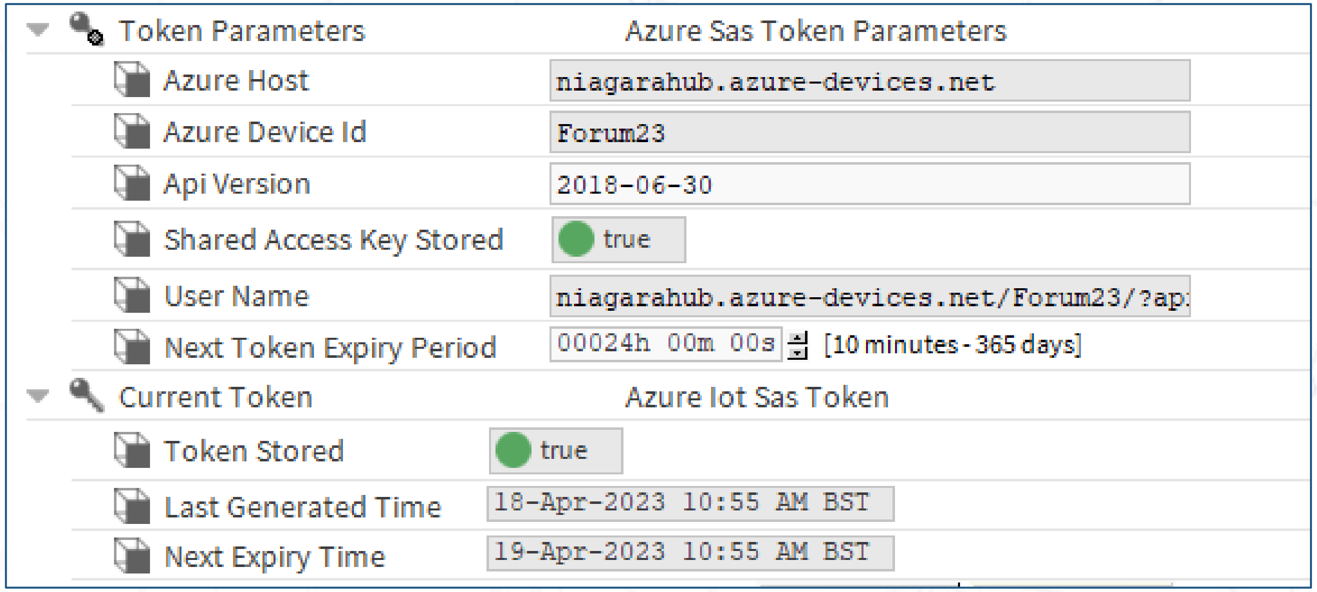
Azure

Azure – SAS Tokens (since 4.12)

Password replaced with generated token

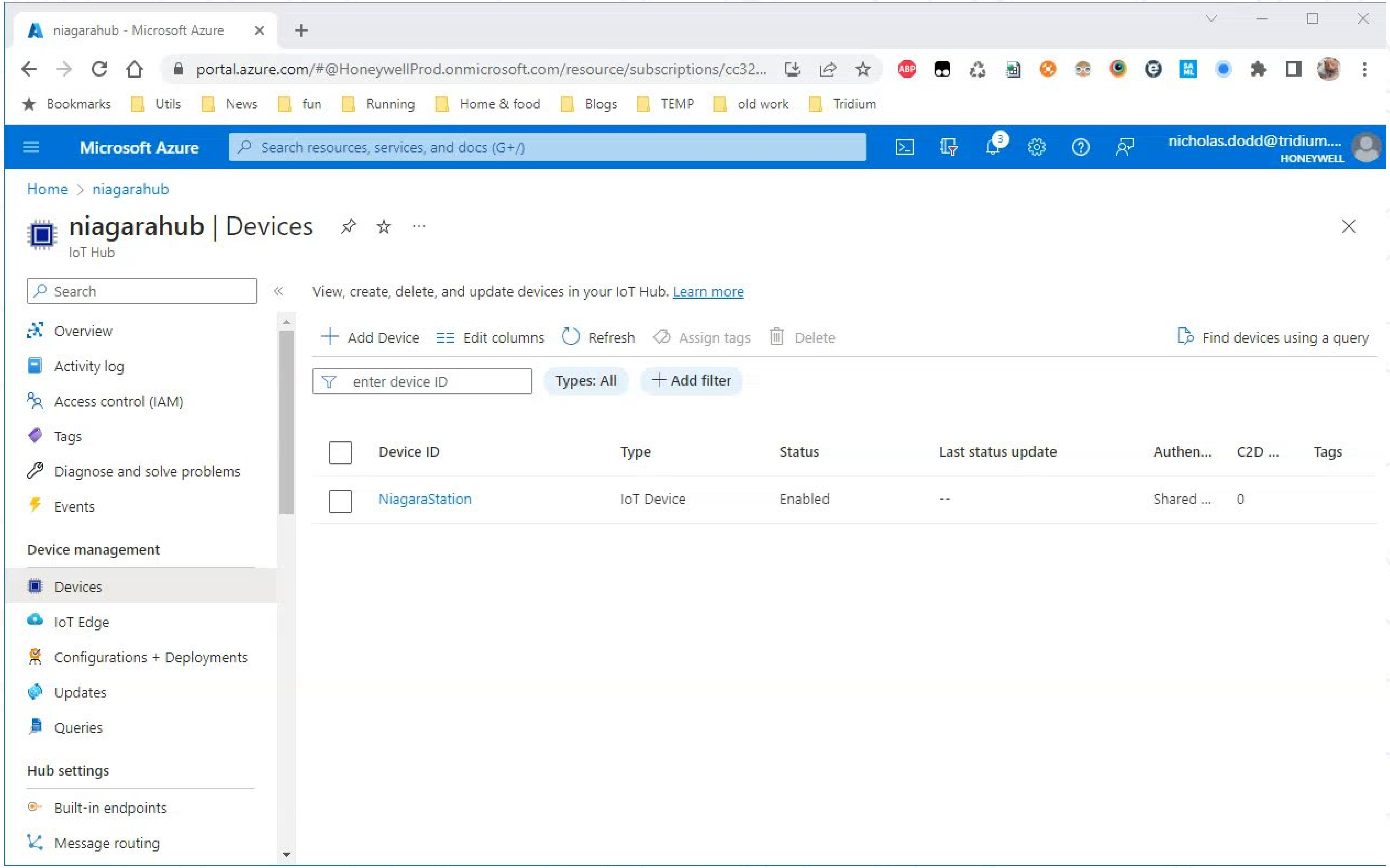
Securely stored encrypted in keyring file

Once first token generated, renews automatically every day



Token Parameters	Azure Sas Token Parameters
Azure Host	niagarahub.azure-devices.net
Azure Device Id	Forum23
Api Version	2018-06-30
Shared Access Key Stored	● true
User Name	niagarahub.azure-devices.net/Forum23/?ap:
Next Token Expiry Period	00024h 00m 00s [10 minutes - 365 days]
Current Token	Azure Iot Sas Token
Token Stored	● true
Last Generated Time	18-Apr-2023 10:55 AM BST
Next Expiry Time	19-Apr-2023 10:55 AM BST

On in 60 seconds?



View, create, delete, and update devices in your IoT Hub. Learn more

□	Device ID	Type	Status	Last status update	Authen...	C2D ...	Tags
□	NiagaraStation	IoT Device	Enabled	--	Shared ...	0	

60 secs

Takeaway points

Super Fast Setup ✓

Secure Credential Storage ✓

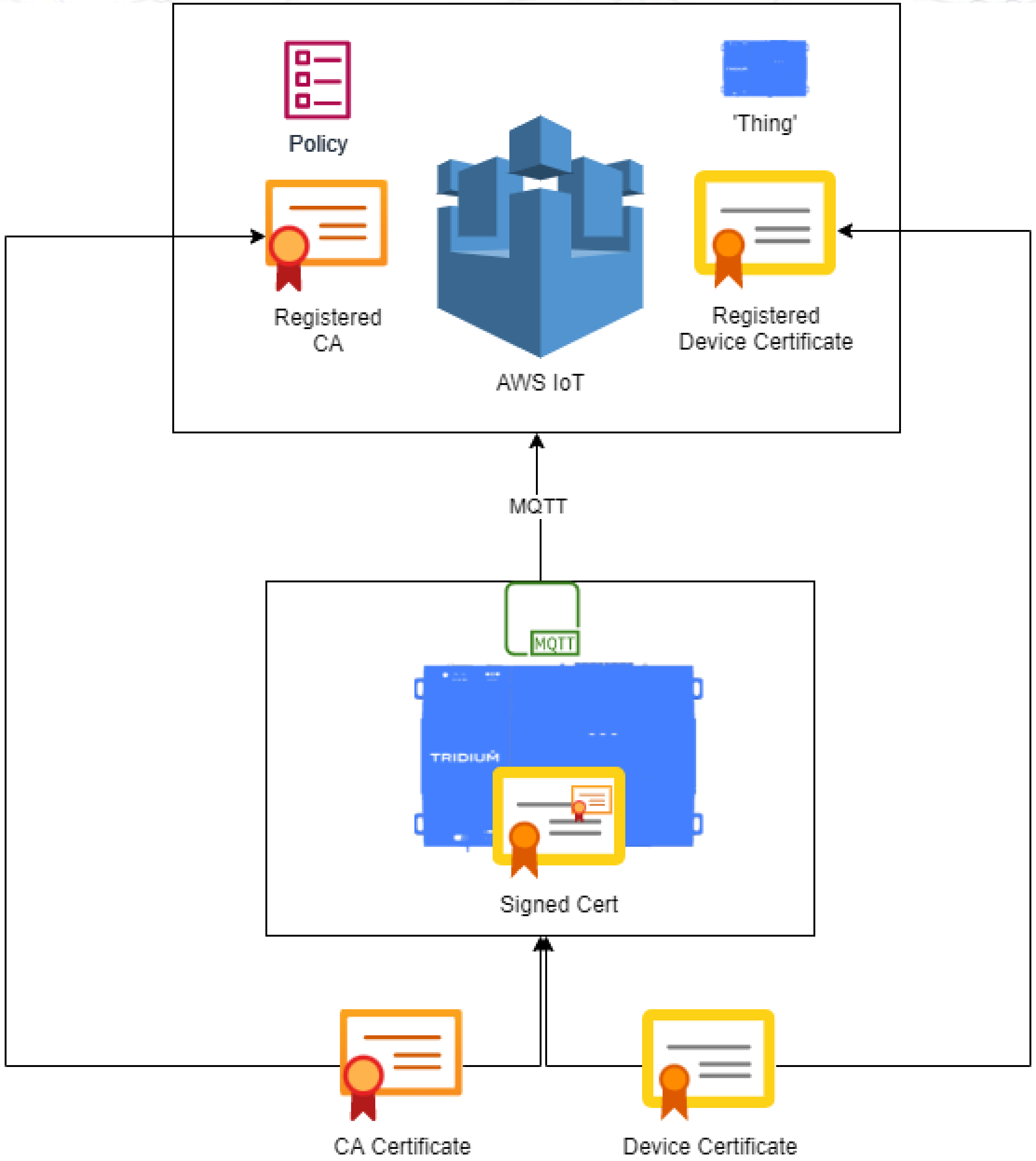
Auto Renewal (walk away) ✓

The Cloud

AWS

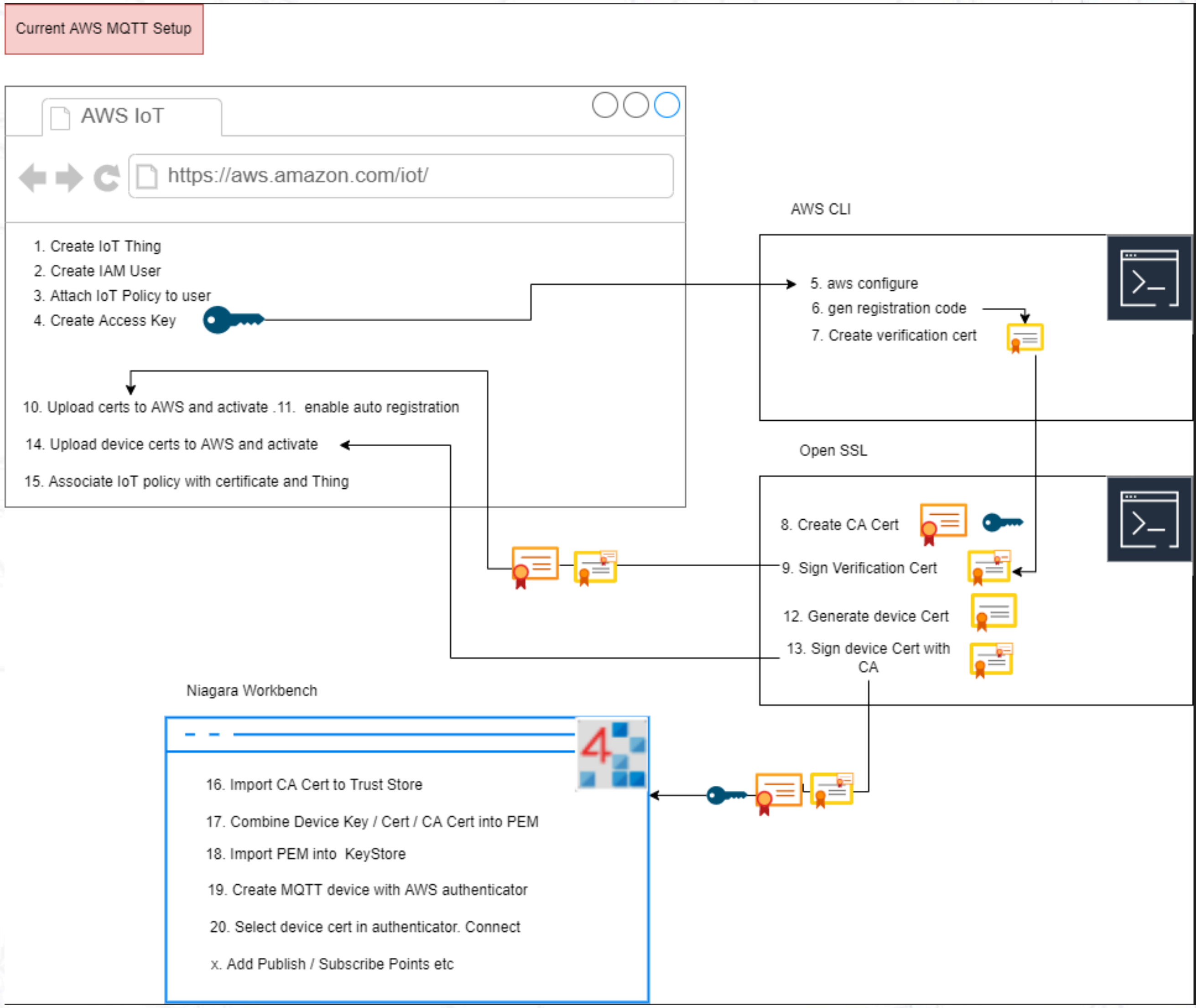
AWS - Certificates

- Create ‘Thing’
- Create policy
- Register CA
- Verify CA
- Generate Device Cert
- Sign with CA
- Upload signed cert
- Activate
- Import Certs to Niagara via workbench
- **Repeat** in x years



A lot of work

- 20 steps to onboard 1st device *
- ~10 per additional device
- Then the same pain again when they expire



Current AWS MQTT Setup

AWS IoT

https://aws.amazon.com/iot/

1. Create IoT Thing
2. Create IAM User
3. Attach IoT Policy to user
4. Create Access Key

10. Upload certs to AWS and activate .11. enable auto registration

14. Upload device certs to AWS and activate

15. Associate IoT policy with certificate and Thing

AWS CLI

5. aws configure
6. gen registration code
7. Create verification cert

Open SSL

8. Create CA Cert
9. Sign Verification Cert
12. Generate device Cert
13. Sign device Cert with CA

Niagara Workbench

16. Import CA Cert to Trust Store
17. Combine Device Key / Cert / CA Cert into PEM
18. Import PEM into KeyStore
19. Create MQTT device with AWS authenticator
20. Select device cert in authenticator. Connect

x. Add Publish / Subscribe Points etc

* Assuming use of CA Certificates

The Cloud

4.13 AWS – Zero Touch Commissioning

The Goal

“To be able to onboard a fleet of controllers to AWS with as close to ‘**zero touch**’ as possible on the controllers“

A 4.13 Solution in 3 parts

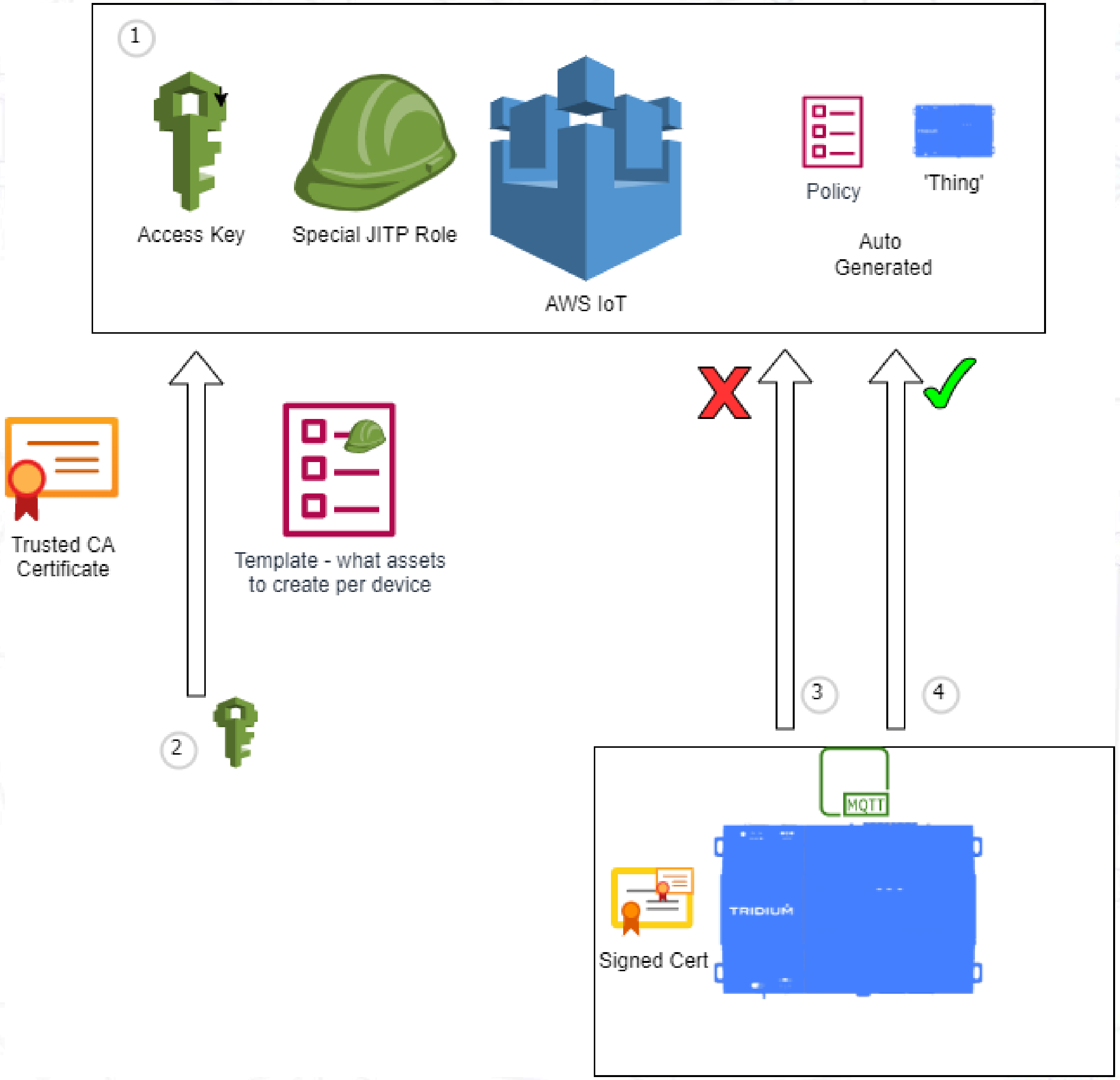
1- New AWS Authenticator / Service

1. Define credential and special JITP Role in AWS

2. Register a trusted CA and policy associated with special role.

3. New MQTT authenticator attempts connection with signed cert, fails while assets are auto generated

4. 2nd attempt succeeds

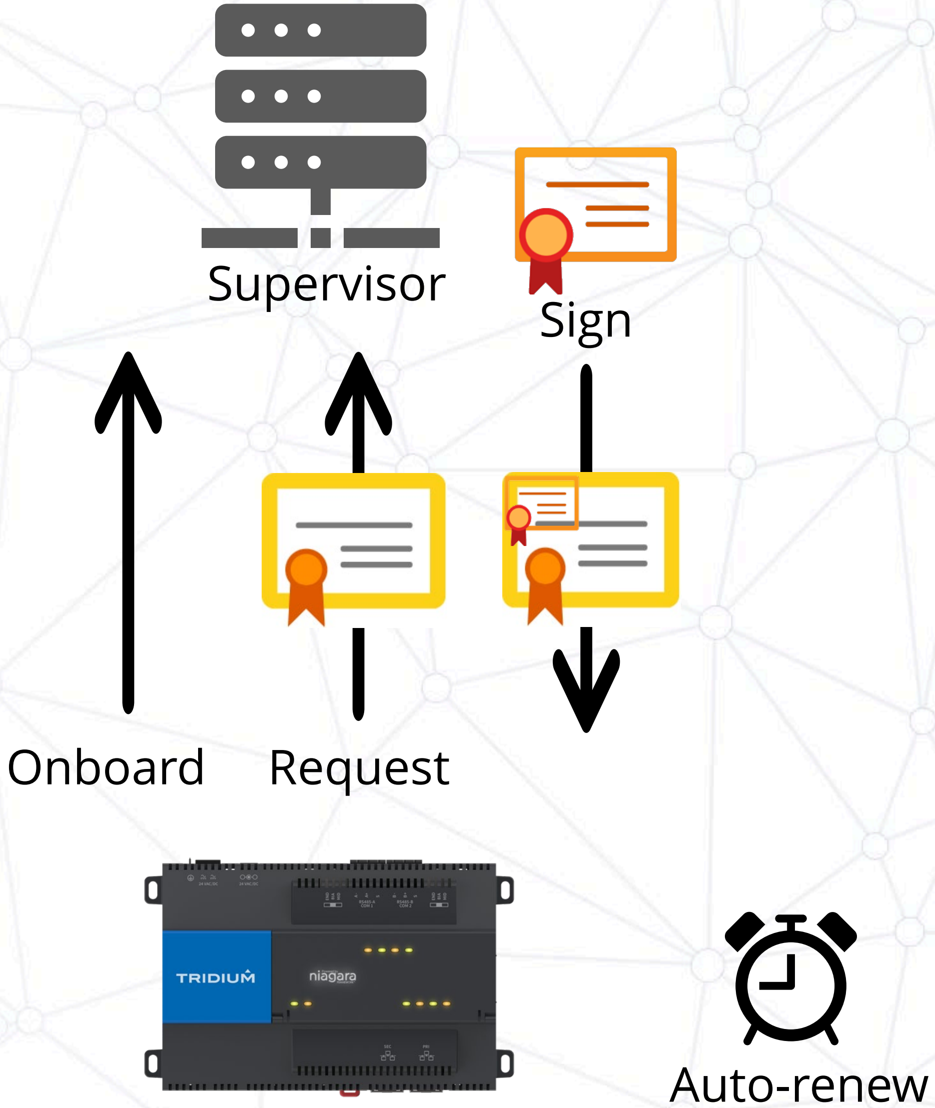


NF 23

TRIDIUM

2 – The Signing Service

- Components register and request signed certificates
- Supervisor signs with CA cert
- Certificates auto renewed with service
- BACnet-SC / Fox / Web to use this in later versions

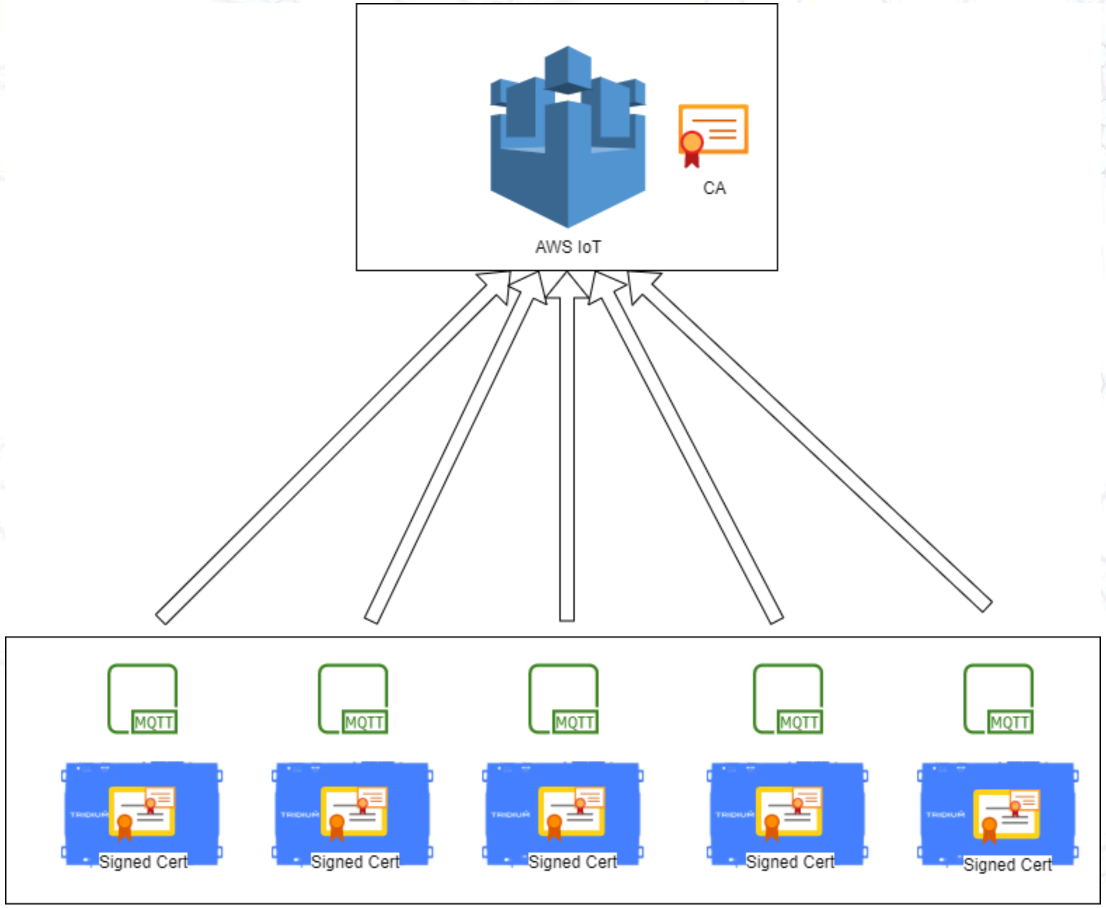


3 – New Provisioning Job

- Install the new MQTT device on each station

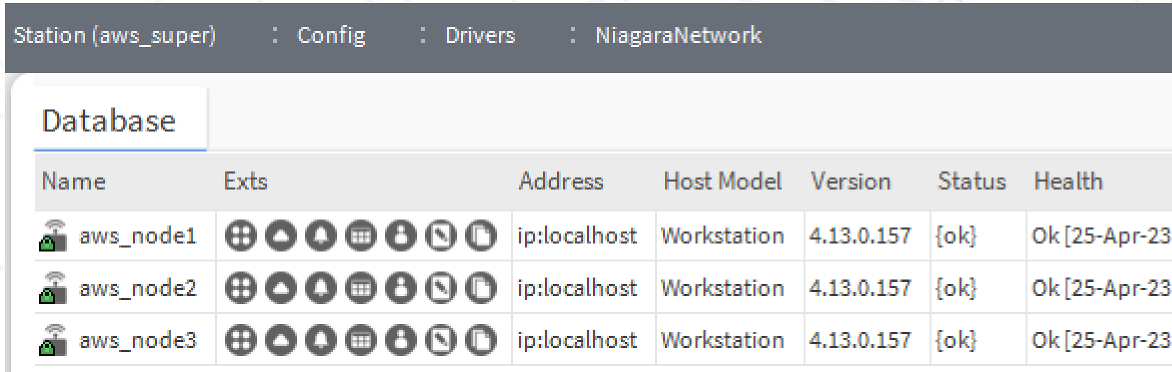
- Auto onboard with signing service

- Devices then auto connect to AWS



A fleet in 5 minutes?

- 3 blank stations on 3 platforms
- Empty supervisor except 3 connections to stations
- No certificates or Things setup in AWS instance, just the special role

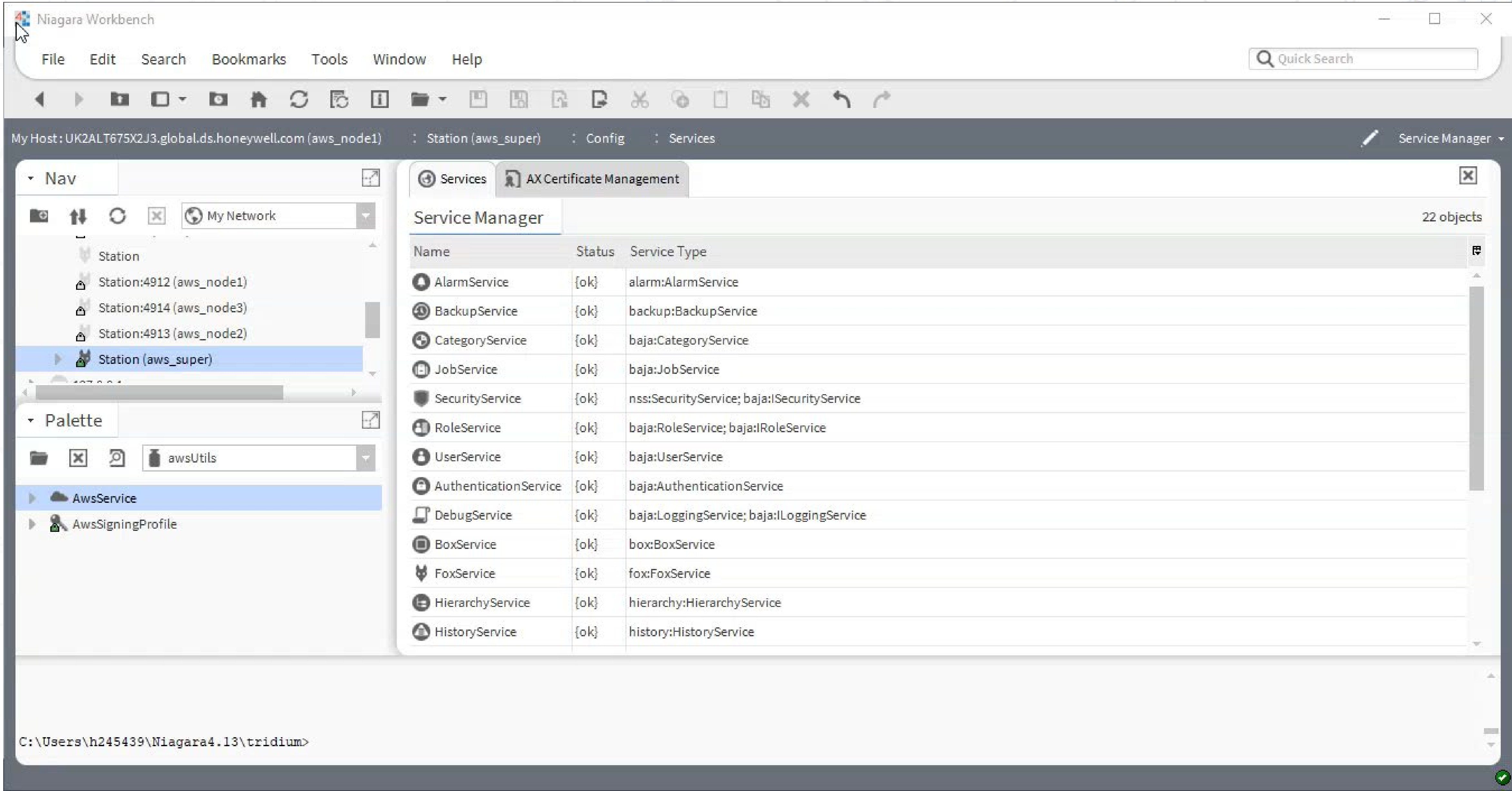


Station (aws_super) : Config : Drivers : NiagaraNetwork

Database

Name	Exts	Address	Host Model	Version	Status	Health
aws_node1		ip:localhost	Workstation	4.13.0.157	{ok}	Ok [25-Apr-23
aws_node2		ip:localhost	Workstation	4.13.0.157	{ok}	Ok [25-Apr-23
aws_node3		ip:localhost	Workstation	4.13.0.157	{ok}	Ok [25-Apr-23

1 Register CA Certificate



My Host : UK2ALT675X2J3.global.ds.honeywell.com (aws_node1) : Station (aws_super) : Config : Services

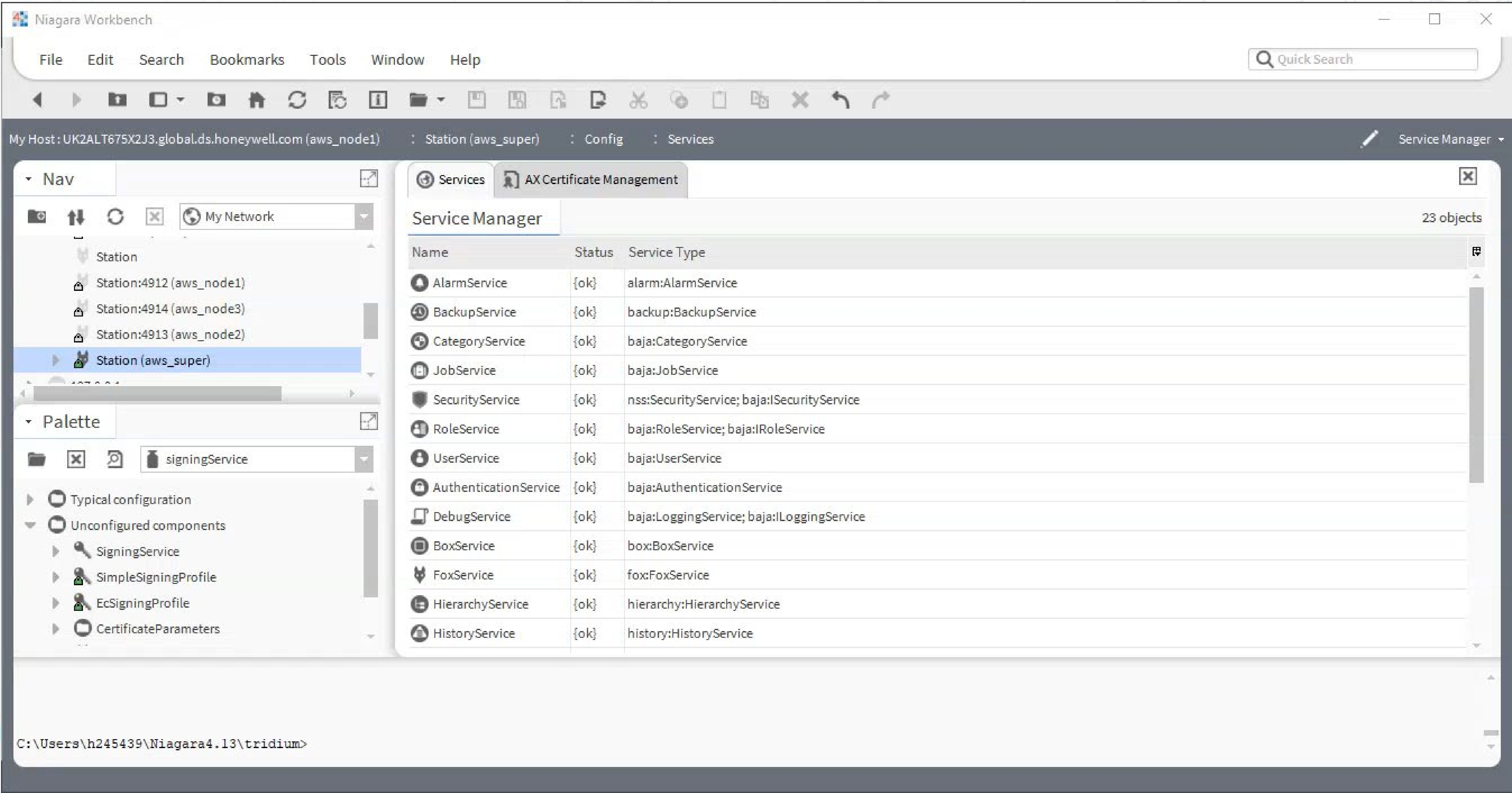
Service Manager

22 objects

Name	Status	Service Type
AlarmService	{ok}	alarm:AlarmService
BackupService	{ok}	backup:BackupService
CategoryService	{ok}	baja:CategoryService
JobService	{ok}	baja:JobService
SecurityService	{ok}	nss:SecurityService; baja:ISecurityService
RoleService	{ok}	baja:RoleService; baja:IRoleService
UserService	{ok}	baja:UserService
AuthenticationService	{ok}	baja:AuthenticationService
DebugService	{ok}	baja:LoggingService; baja:ILoggingService
BoxService	{ok}	box:BoxService
FoxService	{ok}	fox:FoxService
HierarchyService	{ok}	hierarchy:HierarchyService
HistoryService	{ok}	history:HistoryService

```
C:\Users\h245439\Niagara4.13\tridium>
```

2 Setup Signing Profile – 3:40 remaining



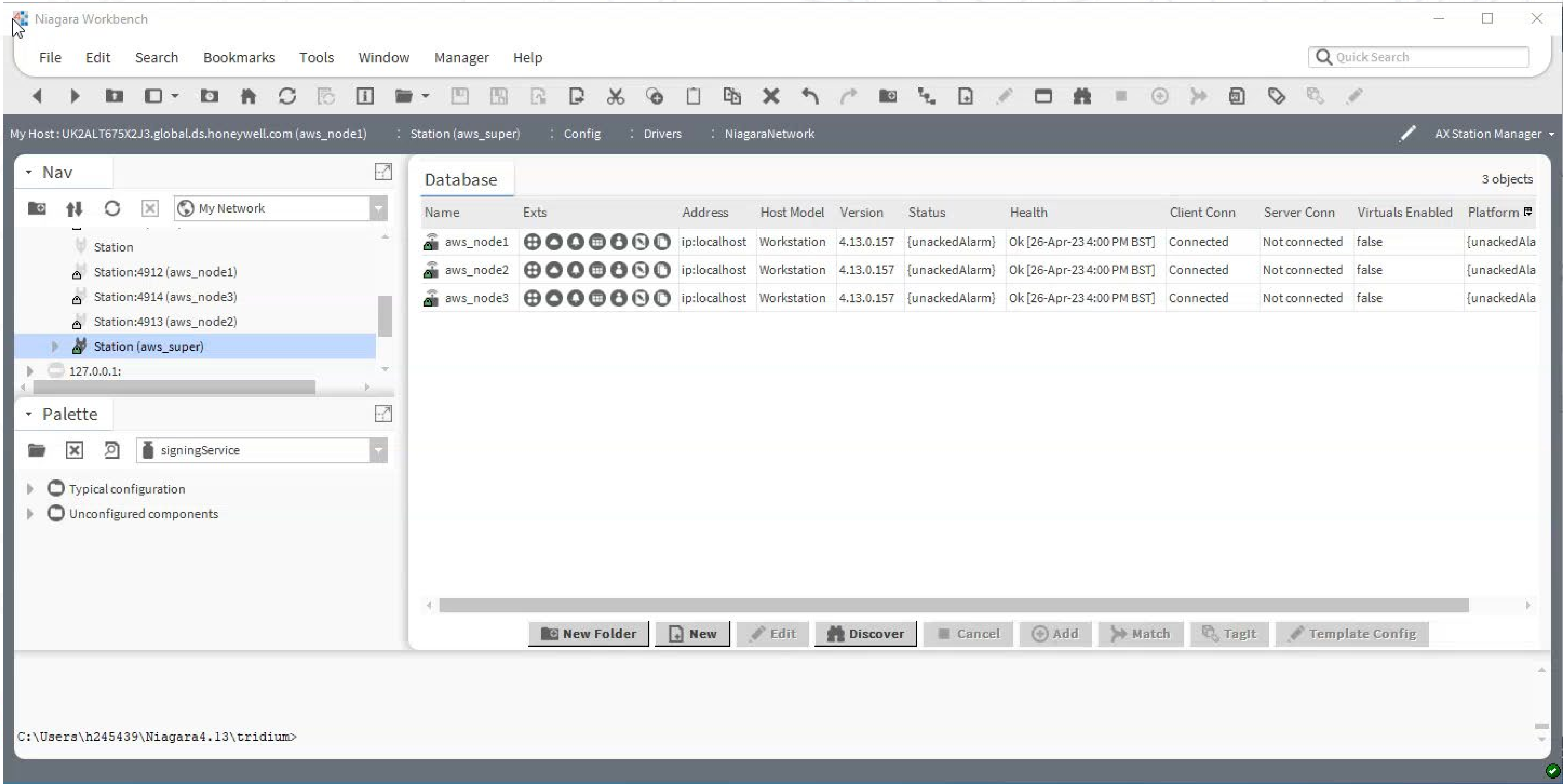
Service Manager

23 objects

Name	Status	Service Type
AlarmService	{ok}	alarm:AlarmService
BackupService	{ok}	backup:BackupService
CategoryService	{ok}	baja:CategoryService
JobService	{ok}	baja:JobService
SecurityService	{ok}	nss:SecurityService; baja:ISecurityService
RoleService	{ok}	baja:RoleService; baja:IRoleService
UserService	{ok}	baja:UserService
AuthenticationService	{ok}	baja:AuthenticationService
DebugService	{ok}	baja:LoggingService; baja:ILoggingService
BoxService	{ok}	box:BoxService
FoxService	{ok}	fox:FoxService
HierarchyService	{ok}	hierarchy:HierarchyService
HistoryService	{ok}	history:HistoryService

```
C:\Users\h245439\Niagara4.13\tridium>
```

3 Provision MQTT Devices – 2:40 remaining

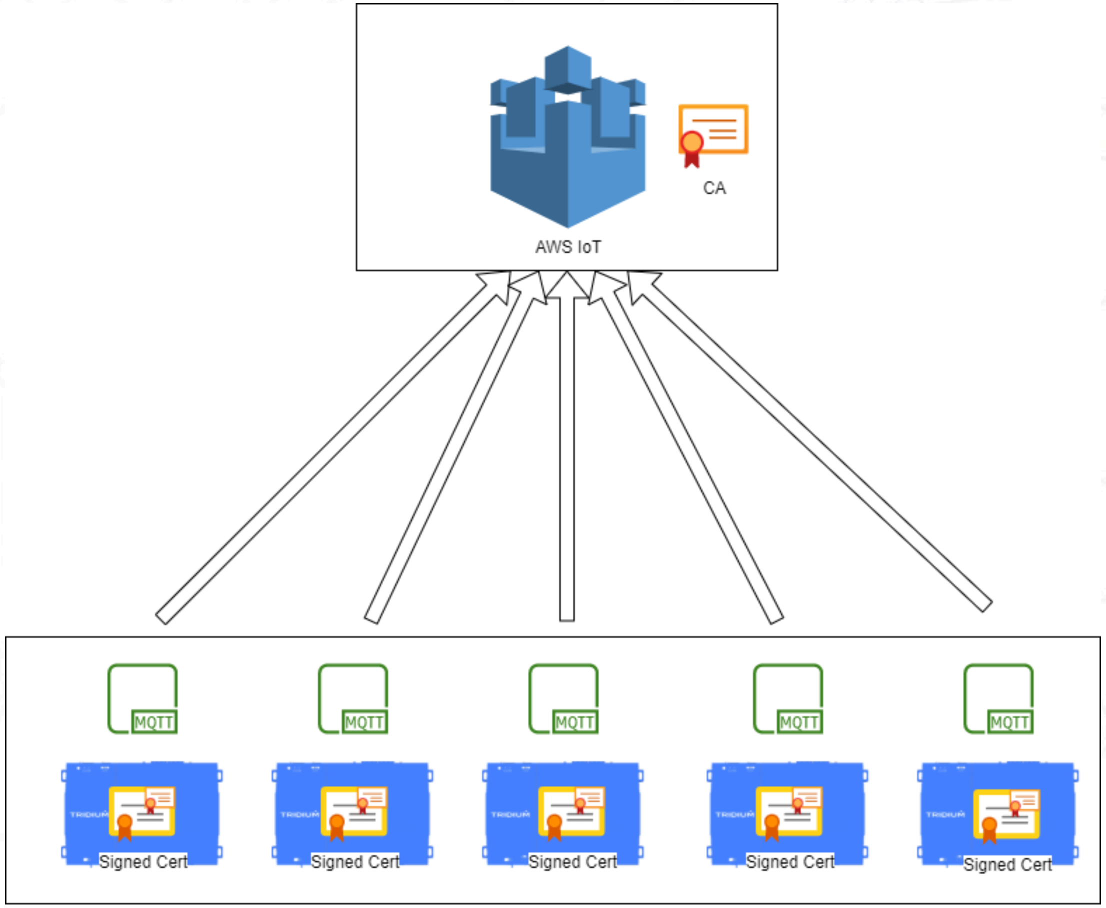


My Host : UK2ALT675X2J3.global.ds.honeywell.com (aws_node1) : Station (aws_super) : Config : Drivers : NiagaraNetwork

Name	Exts	Address	Host Model	Version	Status	Health	Client Conn	Server Conn	Virtuals Enabled	Platform
aws_node1		ip:localhost	Workstation	4.13.0.157	{unackedAlarm}	Ok [26-Apr-23 4:00 PM BST]	Connected	Not connected	false	{unackedAla
aws_node2		ip:localhost	Workstation	4.13.0.157	{unackedAlarm}	Ok [26-Apr-23 4:00 PM BST]	Connected	Not connected	false	{unackedAla
aws_node3		ip:localhost	Workstation	4.13.0.157	{unackedAlarm}	Ok [26-Apr-23 4:00 PM BST]	Connected	Not connected	false	{unackedAla

```
C:\Users\h245439\Niagara4.13\tridium>
```

Done



AWS IoT
CA
MQTT
MQTT
MQTT
MQTT
MQTT
Signed Cert
Signed Cert
Signed Cert
Signed Cert
Signed Cert

00:20

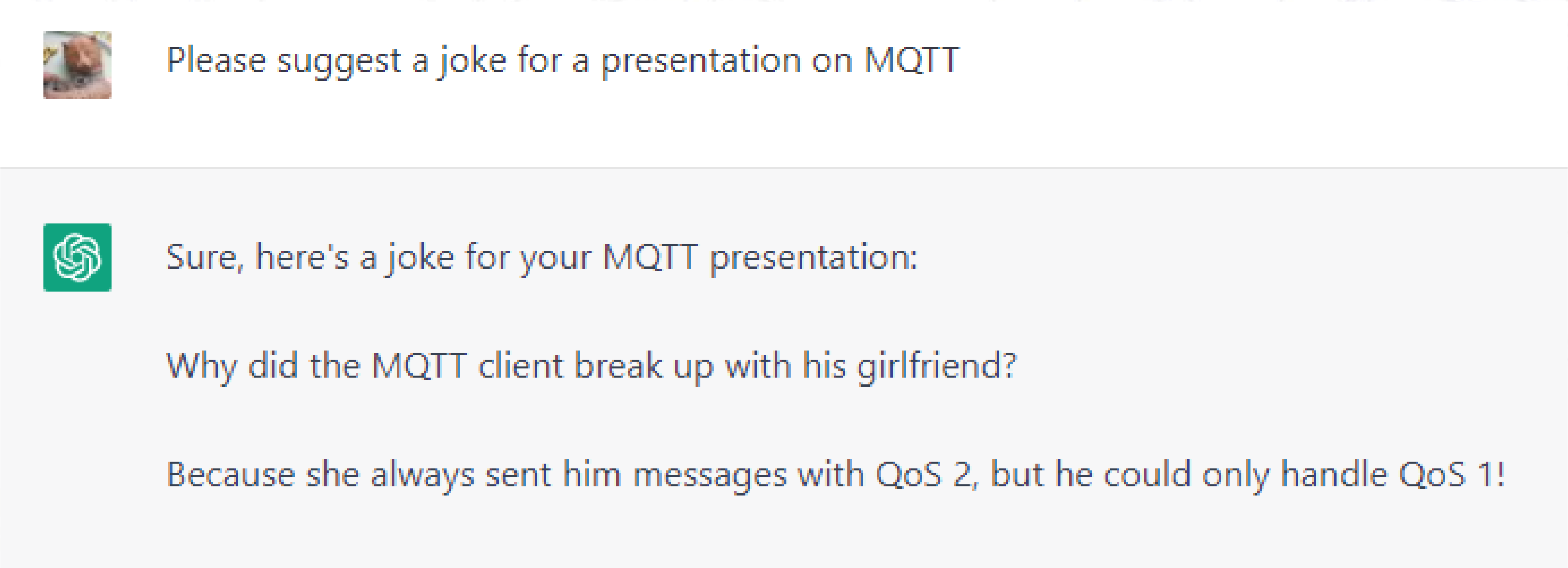
Takeaway points

All config performed from supervisor ✓

Auto Renewal (walk away) ✓

BACnet-SC and others coming soon ✓

Thank you



Please suggest a joke for a presentation on MQTT

Sure, here's a joke for your MQTT presentation:

Why did the MQTT client break up with his girlfriend?

Because she always sent him messages with QoS 2, but he could only handle QoS 1!

NF 23

CONNECTING

THE WORLD

JSON Toolkit Preview

Jason Woollard
Tridium

Where is JSON used?

UI / Charting / JavaScript

REST

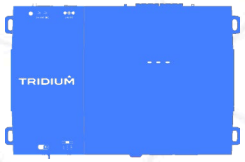
HTTP

Web Service API's

mongoDB

Storage – NoSQL

MQTT.ORG



IoT Device API's

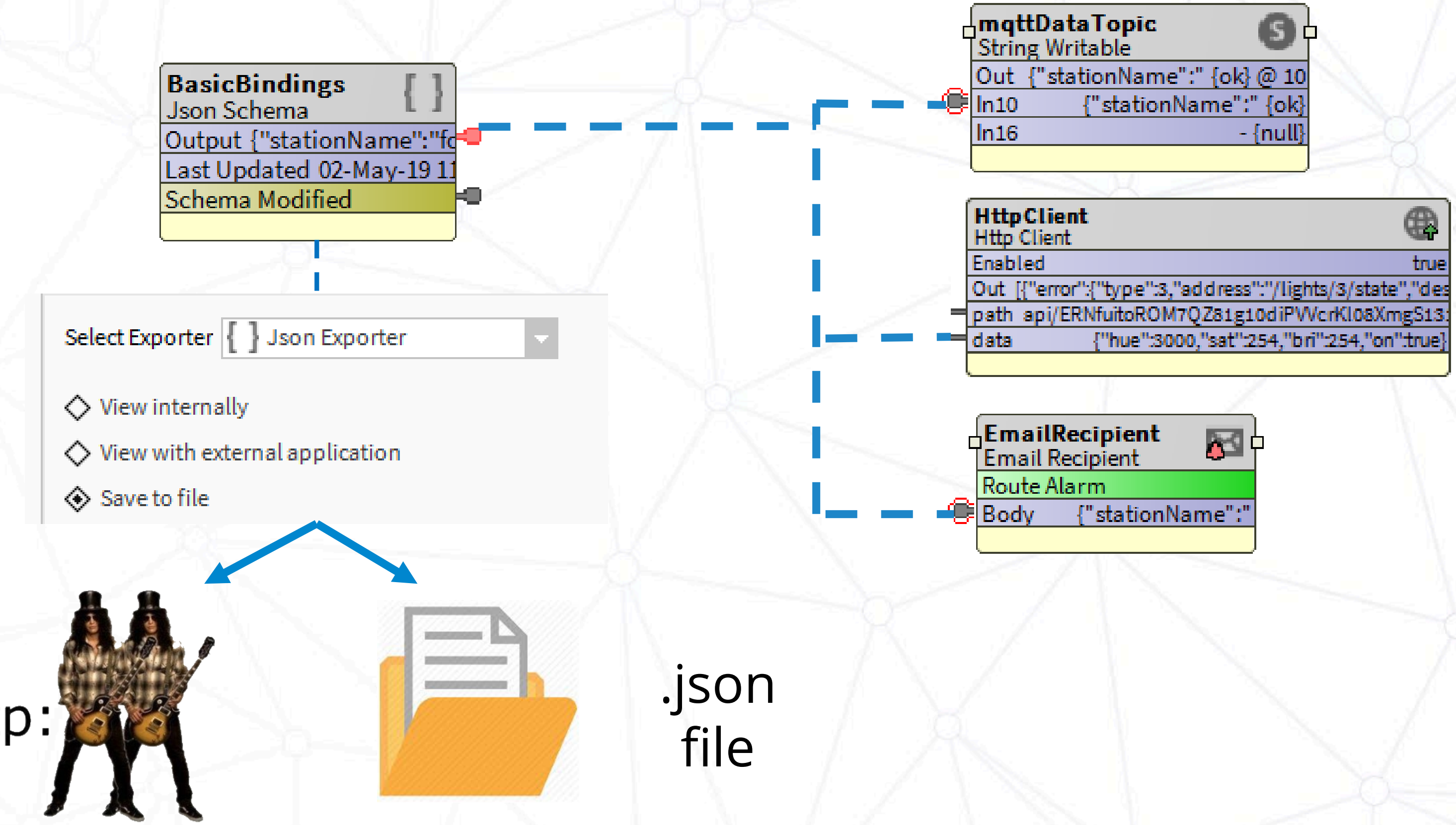
What is JSON?

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

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

Transport Agnostic

(Included with JSON Toolkit)



BasicBindings
Json Schema
Output {"stationName":"fo
Last Updated 02-May-19 11
Schema Modified

Select Exporter { } Json Exporter

◇ View internally
◇ View with external application
◈ Save to file

mqttDataTopic
String Writable
Out {"stationName":" {ok} @ 10
In10 {"stationName":" {ok}
In16 - {null}

HttpClient
Http Client
Enabled true
Out [{"error":{"type":3,"address":"/lights/3/state","des
path api/ERNfuitoROM7QZ81g10diPVVcrKI08XmgS13
data {"hue":3000,"sat":254,"bri":254,"on":true}

EmailRecipient
Email Recipient
Route Alarm
Body {"stationName":"

http:

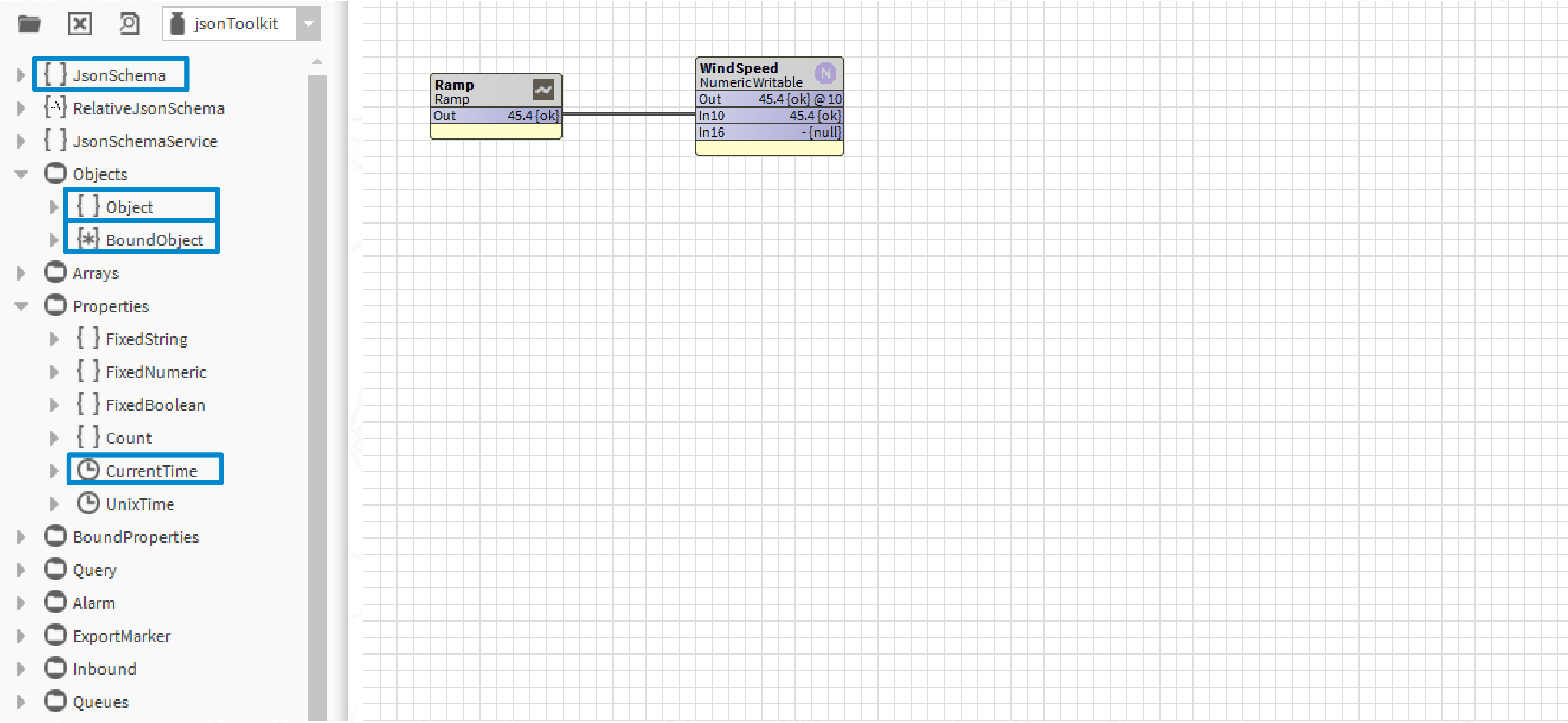
.json file

MQTT ✓

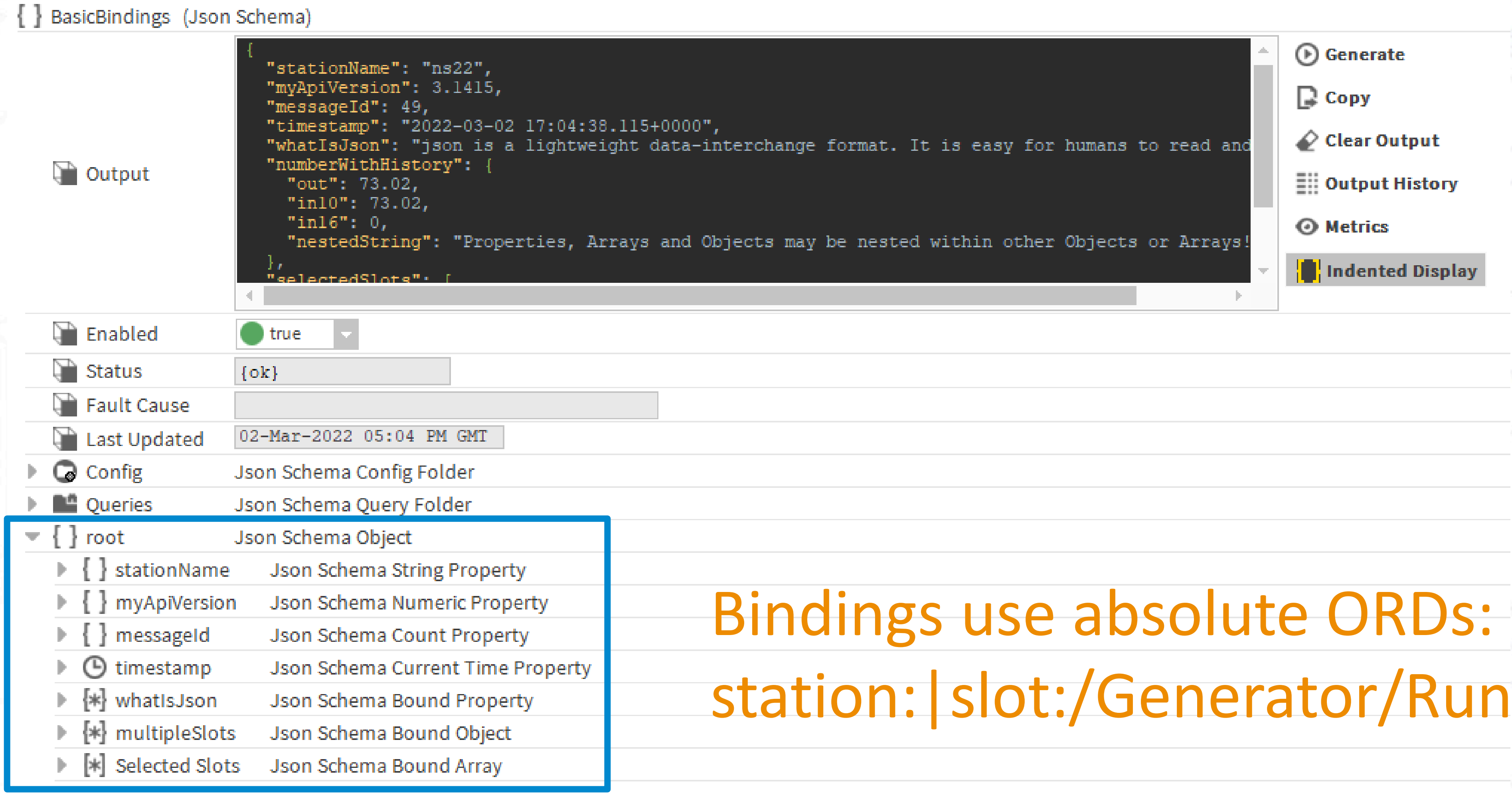
HTTP ✓

E-mail ⚠

JSON Schema basics: Bindings



JSON Schema basics



{ } BasicBindings (Json Schema)

Output

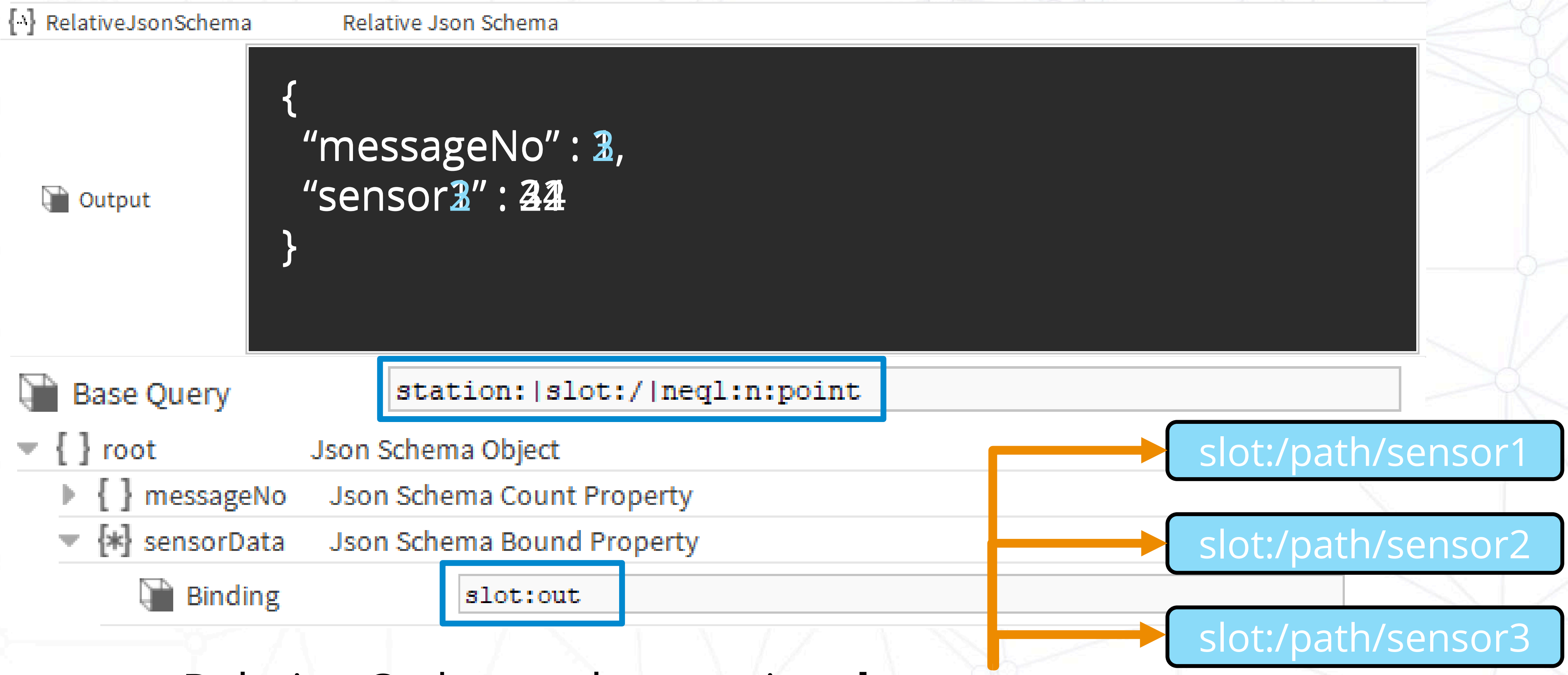
```
{
  "stationName": "ns22",
  "myApiVersion": 3.1415,
  "messageId": 49,
  "timestamp": "2022-03-02 17:04:38.115+0000",
  "whatIsJson": "json is a lightweight data-interchange format. It is easy for humans to read and
  "numberWithHistory": {
    "out": 73.02,
    "in10": 73.02,
    "in16": 0,
  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

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

Bindings use absolute ORDs:

station:|slot:/Generator/Runtime

Scaling with ‘Relative’ Schema



RelativeJsonSchema — Relative Json Schema

Output

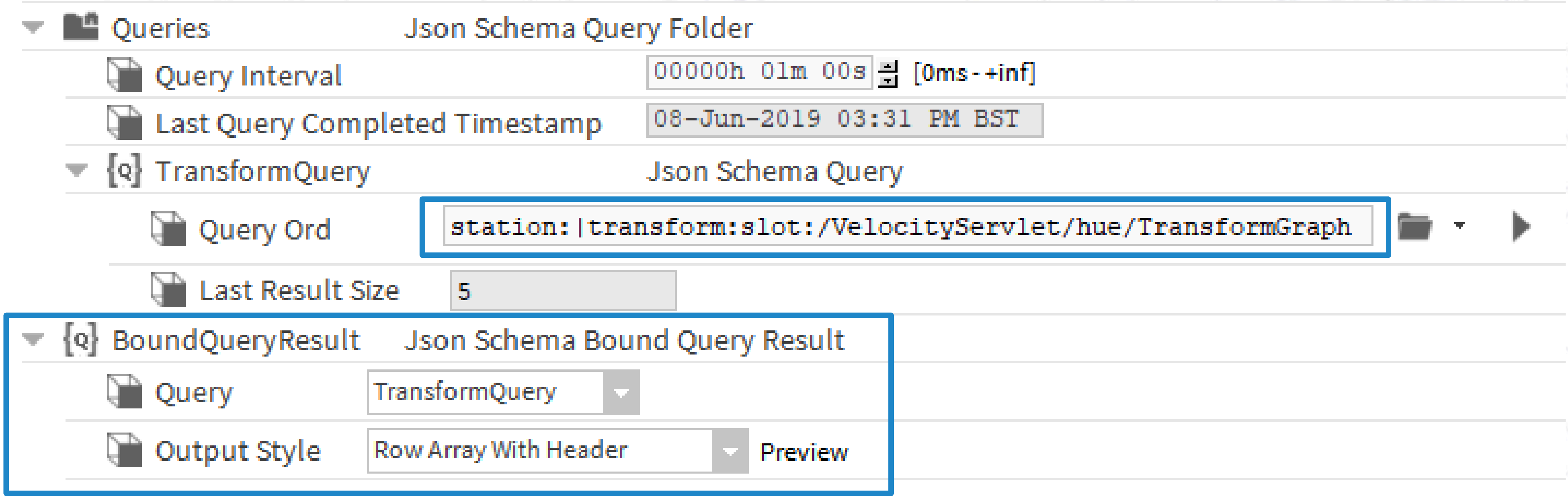
```
{
  "messageNo" : 1,
  "sensor1" : 41
}
```

Base Query: `station:|slot:/|neql:n:point`

- root — Json Schema Object
  - messageNo — Json Schema Count Property
  - sensorData — Json Schema Bound Property
    - Binding: `slot:out`

Relative Ord – resolves against **base:**

Queries



Queries Json Schema Query Folder

Query Interval 00000h 01m 00s [0ms - +inf]

Last Query Completed Timestamp 08-Jun-2019 03:31 PM BST

TransformQuery Json Schema Query

Query Ord `station:|transform:slot:/VelocityServlet/hue/TransformGraph`

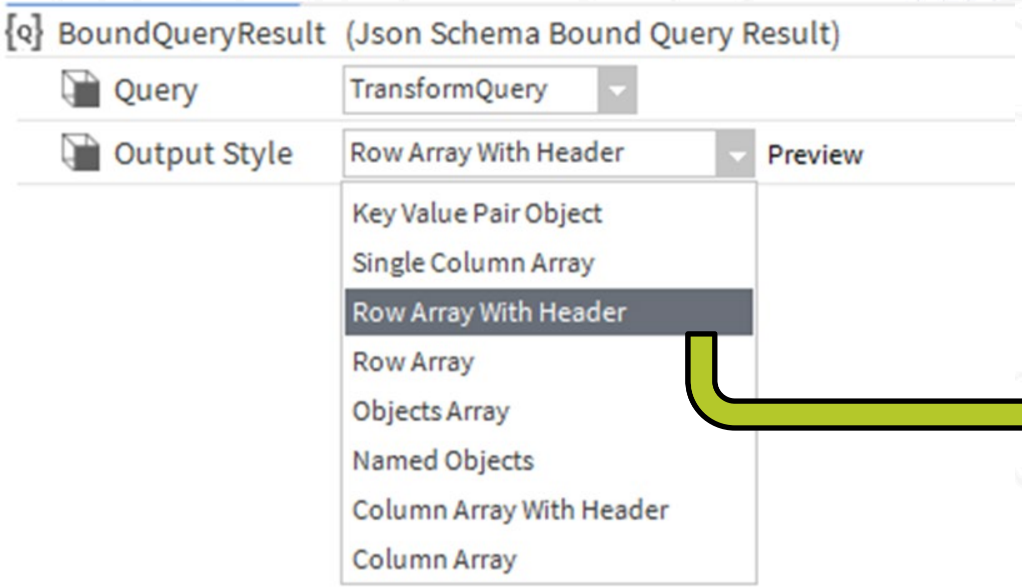
Last Result Size 5

BoundQueryResult Json Schema Bound Query Result

Query TransformQuery

Output Style Row Array With Header

Queries



{Q} BoundQueryResult (Json Schema Bound Query Result)

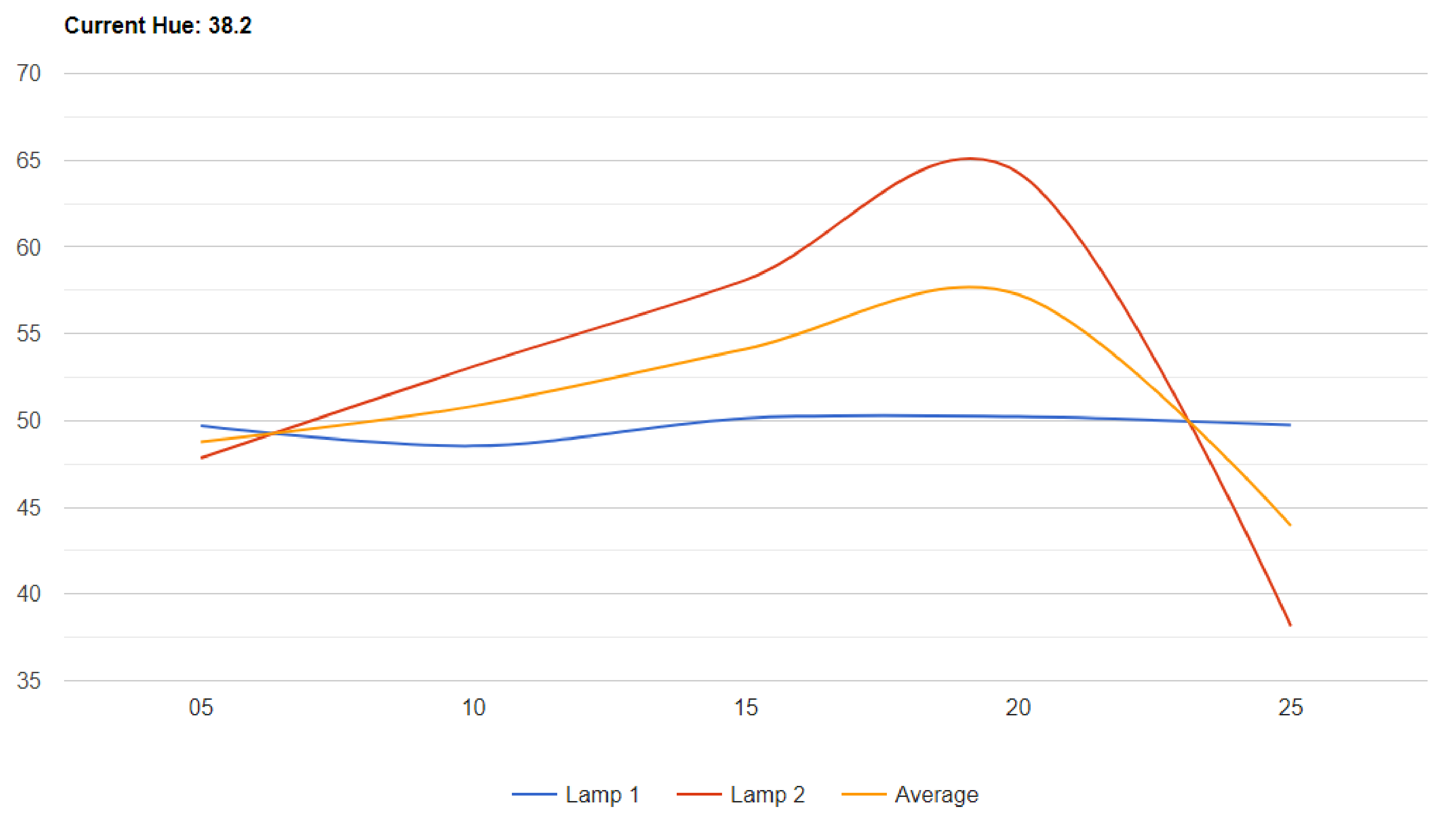
Query: TransformQuery

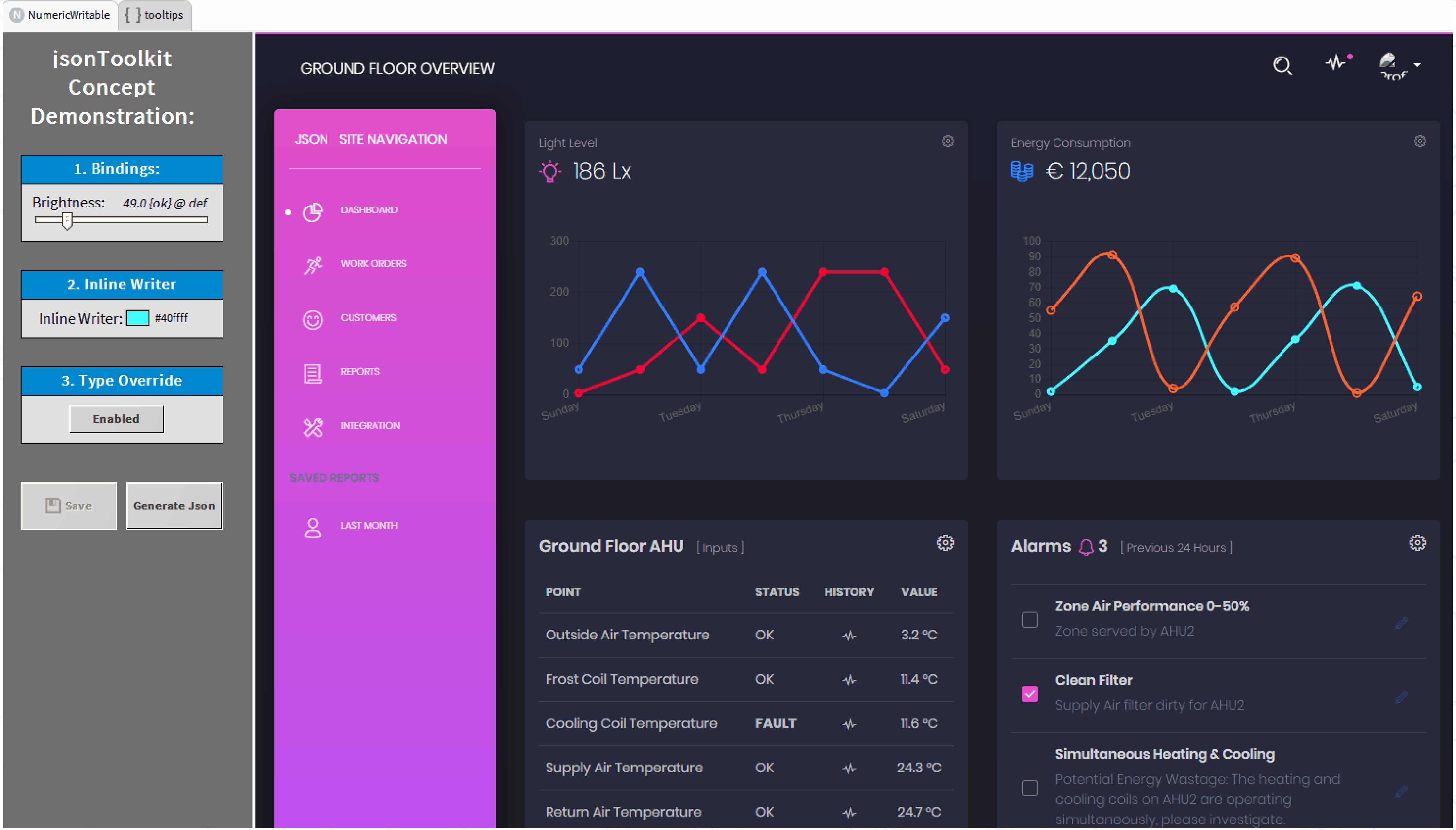
Output Style: Row Array With Header

Preview

```
[
  [
    "Timestamp", "Light 1",
    "Light 2", "Average"
  ],
  [
    "55", 50.8002,
    43.7057, 47.253
  ],
  [
    "00", 49.1097,
    56.5209.....
  ]
]
```

Viewing the result





jsonToolkit Concept Demonstration:

1. Bindings:

Brightness: *49.0 {ok} @ def*

2. Inline Writer

Inline Writer: 🟦 #40ffff

3. Type Override

GROUND FLOOR OVERVIEW

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

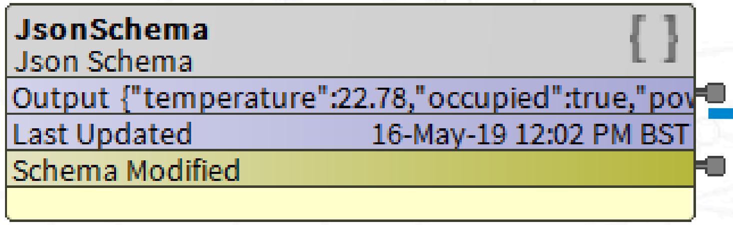
Alarms 🔔 **3** [Previous 24 Hours]

☐ **Zone Air Performance 0-50%**
Zone served by AHU2

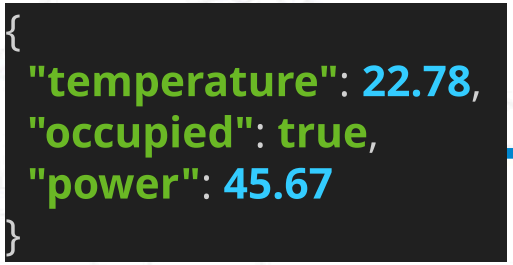
☑ **Clean Filter**
Supply Air filter dirty for AHU2

☐ **Simultaneous Heating & Cooling**
Potential Energy Wastage: The heating and cooling coils on AHU2 are operating simultaneously, please investigate.

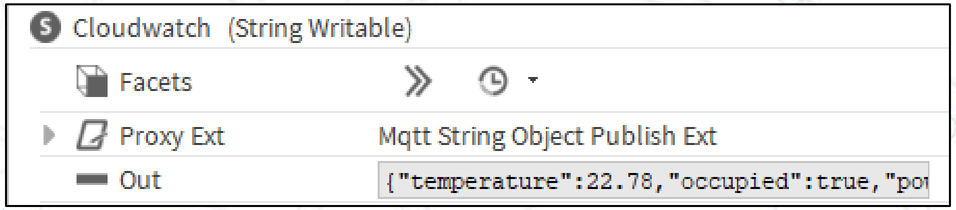
AWS: Cloudwatch Charting



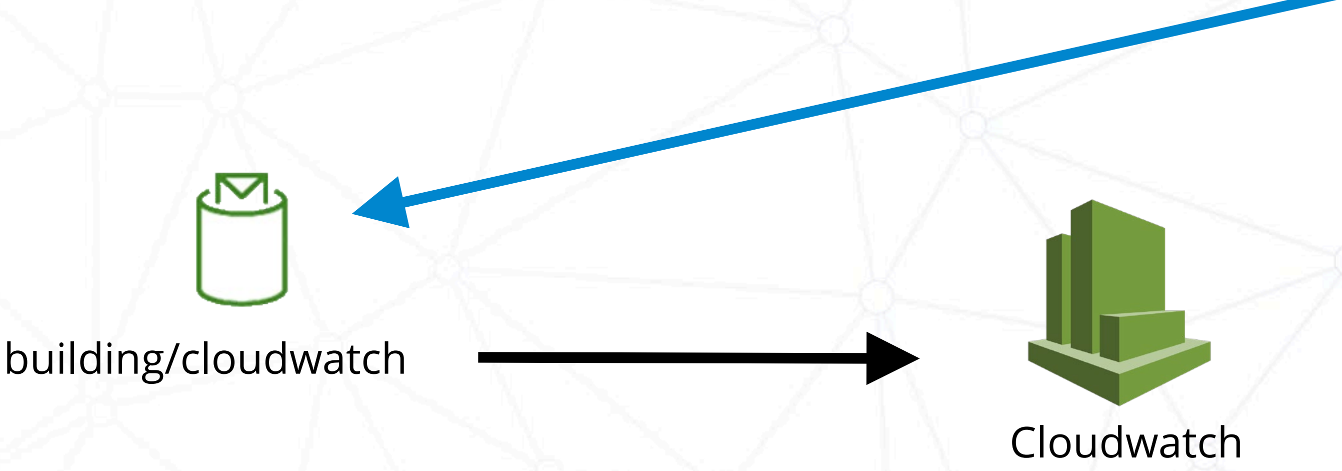
JsonSchema { }
Json Schema
Output {"temperature":22.78,"occupied":true,"pov
Last Updated 16-May-19 12:02 PM BST
Schema Modified

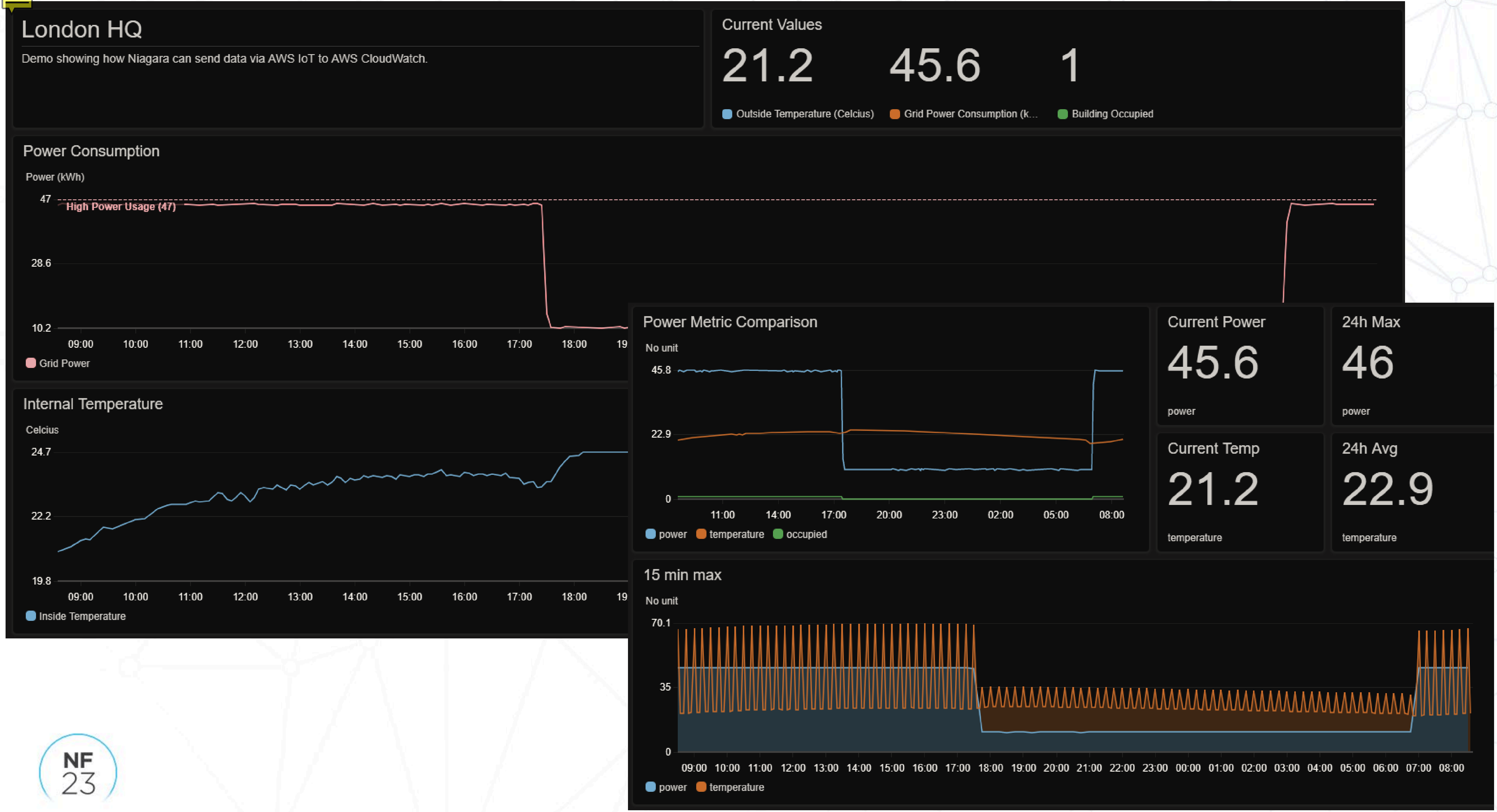


```
{
 "temperature": 22.78,
 "occupied": true,
 "power": 45.67
}
```



Cloudwatch (String Writable)
Facets
Proxy Ext Mqtt String Object Publish Ext
Out {"temperature":22.78,"occupied":true,"pov





London HQ

Demo showing how Niagara can send data via AWS IoT to AWS CloudWatch.

Current Values

21.2 | 45.6 | 1

● Outside Temperature (Celcius) ● Grid Power Consumption (k... ● Building Occupied

Power Consumption

Internal Temperature

Power Metric Comparison

Current Power

45.6

power

24h Max

46

power

Current Temp

21.2

temperature

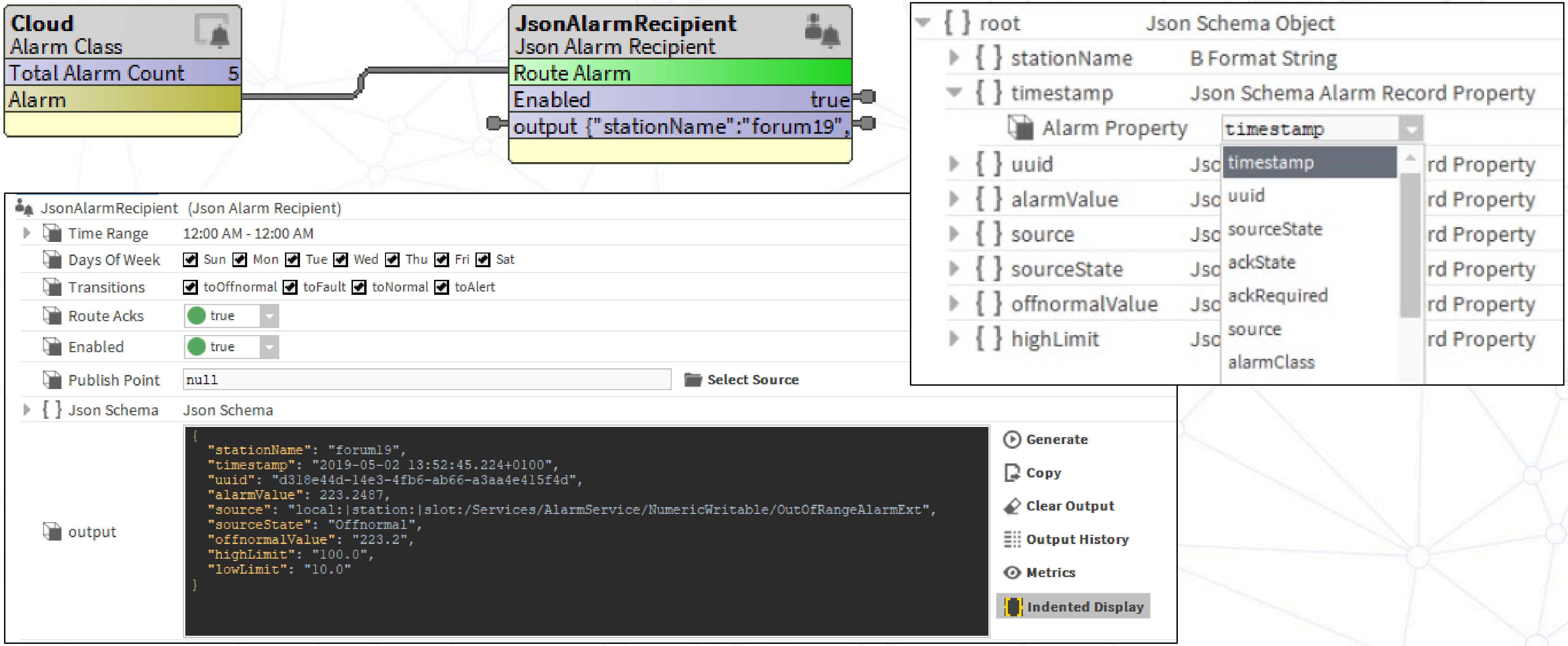
24h Avg

22.9

temperature

15 min max

JSON Alarm Recipient



Cloud
Alarm Class
Total Alarm Count 5
Alarm

JsonAlarmRecipient
Json Alarm Recipient
Route Alarm
Enabled true
output {"stationName":"forum19",

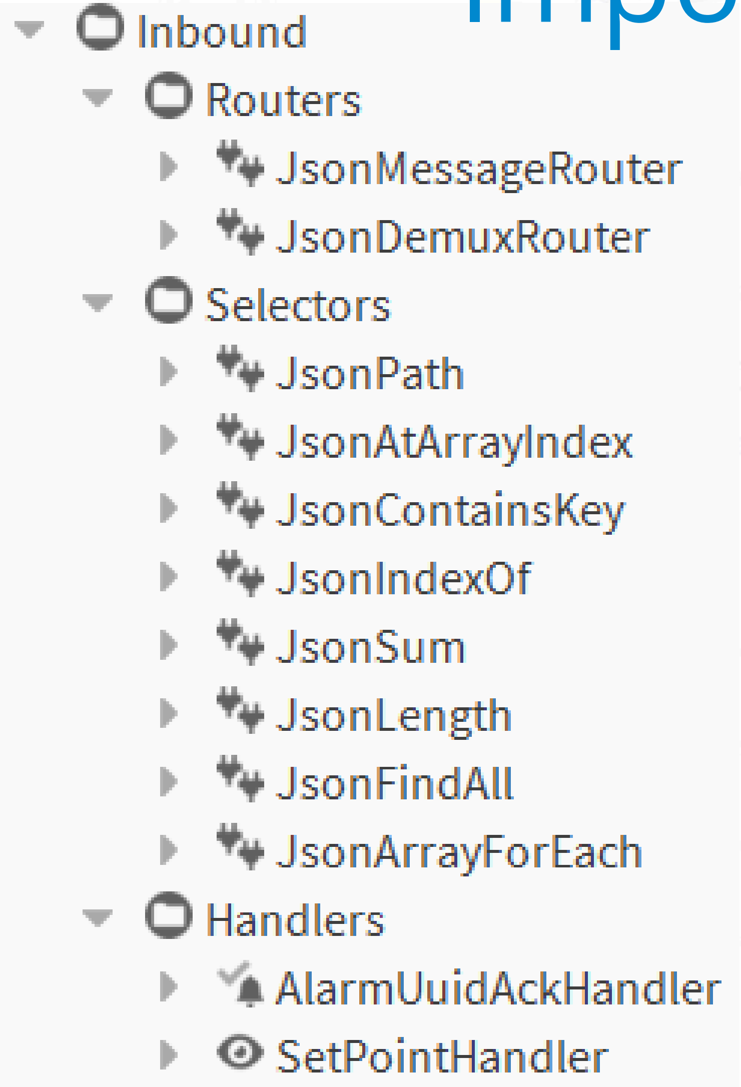
JsonAlarmRecipient (Json Alarm Recipient)

- 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"
}
```

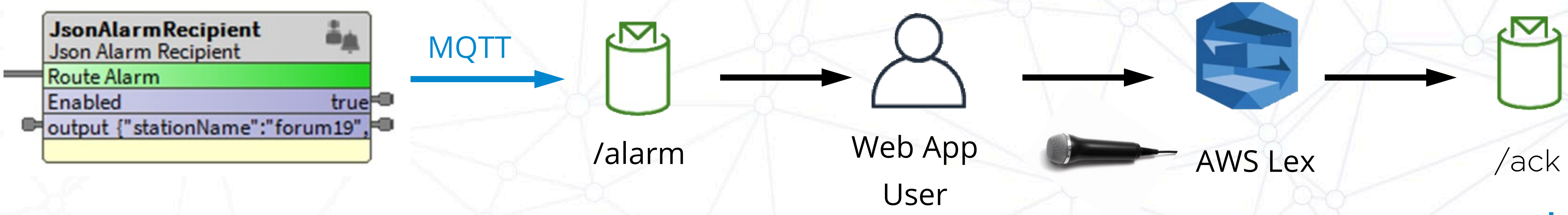
{ } root	Json Schema Object
{ } stationName	B Format String
{ } timestamp	Json Schema Alarm Record Property
Alarm Property	timestamp
{ } uuid	Jso…rd Property
{ } alarmValue	Jso…rd Property
{ } source	Jso…rd Property
{ } sourceState	Jso…rd Property
{ } offnormalValue	Jso…rd Property
{ } highLimit	Jso…rd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

Importing JSON



- Inbound
  - Routers
    - JsonMessageRouter
    - JsonDemuxRouter
  - Selectors
    - JsonPath
    - JsonAtArrayIndex
    - JsonContainsKey
    - JsonIndexOf
    - JsonSum
    - JsonLength
    - JsonFindAll
    - JsonArrayForEach
  - Handlers
    - AlarmUuidAckHandler
    - SetPointHandler

Alarm Demo

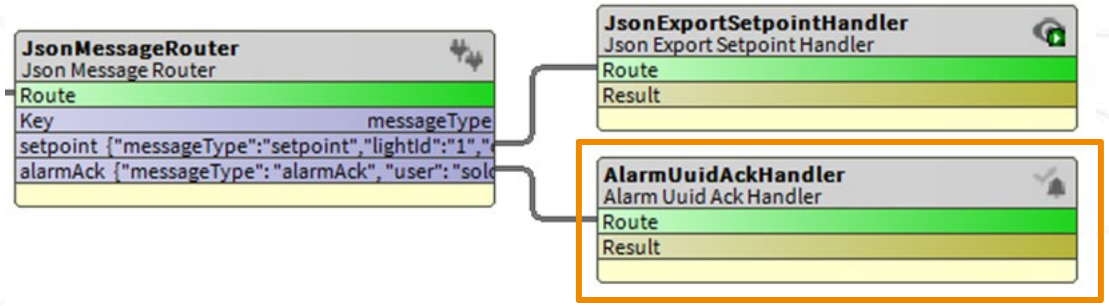


JsonAlarmRecipient
Json Alarm Recipient
Route Alarm
Enabled true
output {"stationName":"forum19",

```
{
 "stationName": “forum23",
 "timestamp": "2023-05-16 12:24:50",
 "uuid": "3842f3f1-21b6-4…",
 "alarmValue": 200,
 "source": "local:|station:|slot:…….",
 "sourceState": "Offnormal",
 "offnormalValue": "200.0",
 "highLimit": "100.0",
 "lowLimit": "10.0"
}
```

```
{
 "messageType“ : "alarmAck",
 "user“ : “Andrew",
 "alarms“ : ["3842f3f1-21b6…"]
}
```

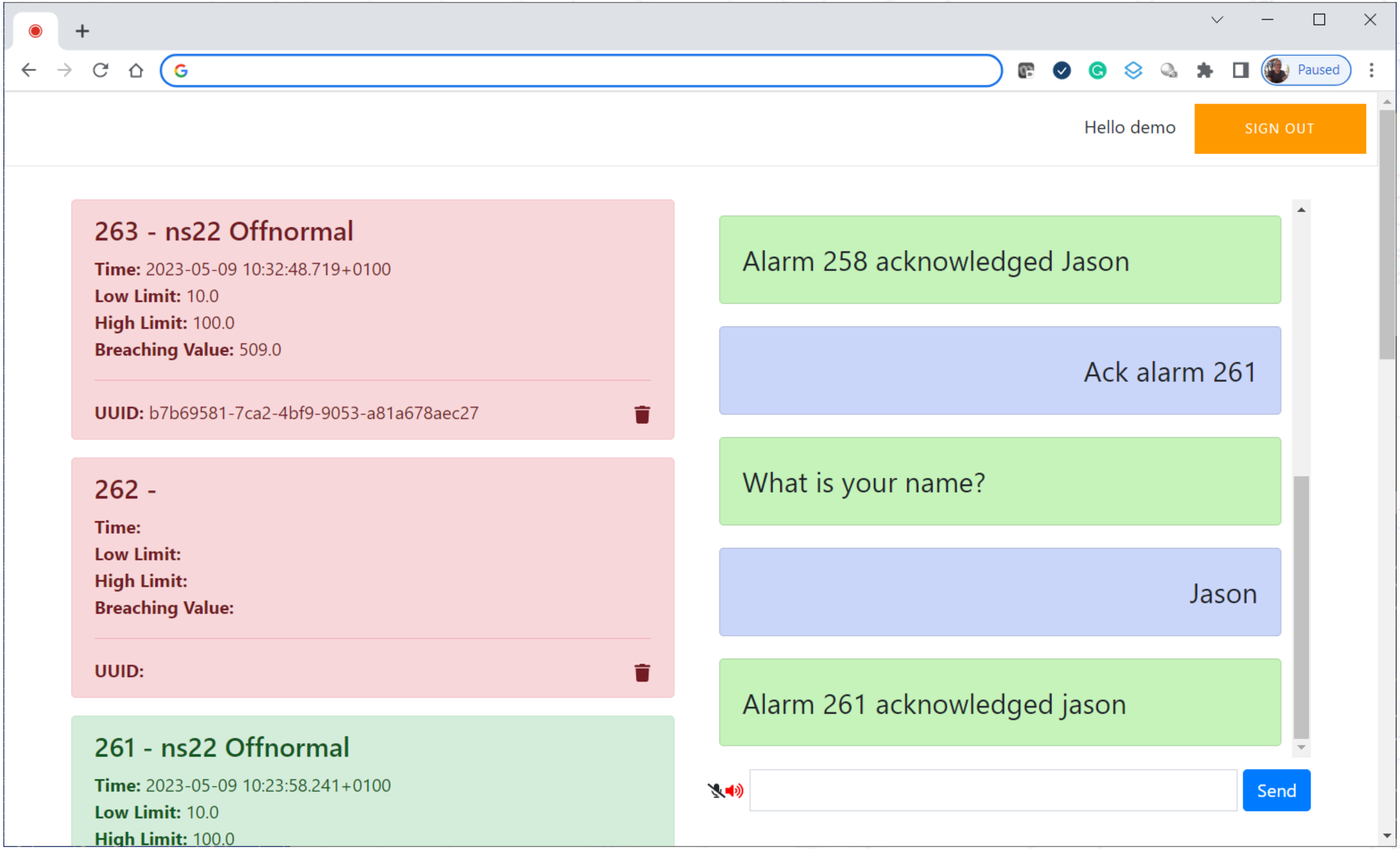
MQTT



JsonMessageRouter
Json Message Router
Route
Key messageType
setpoint {"messageType":"setpoint","lightId":"1","
alarmAck {"messageType":"alarmAck","user":"sol

JsonExportSetpointHandler
Json Export Setpoint Handler
Route
Result

AlarmUuidAckHandler
Alarm Uuid Ack Handler
Route
Result



263 - ns22 Offnormal

Time: 2023-05-09 10:32:48.719+0100
Low Limit: 10.0
High Limit: 100.0
Breaching Value: 509.0

UUID: b7b69581-7ca2-4bf9-9053-a81a678aec27

262 -

Time:
Low Limit:
High Limit:
Breaching Value:

UUID:

261 - ns22 Offnormal

Time: 2023-05-09 10:23:58.241+0100
Low Limit: 10.0
High Limit: 100.0

Alarm 258 acknowledged Jason

Ack alarm 261

What is your name?

Jason

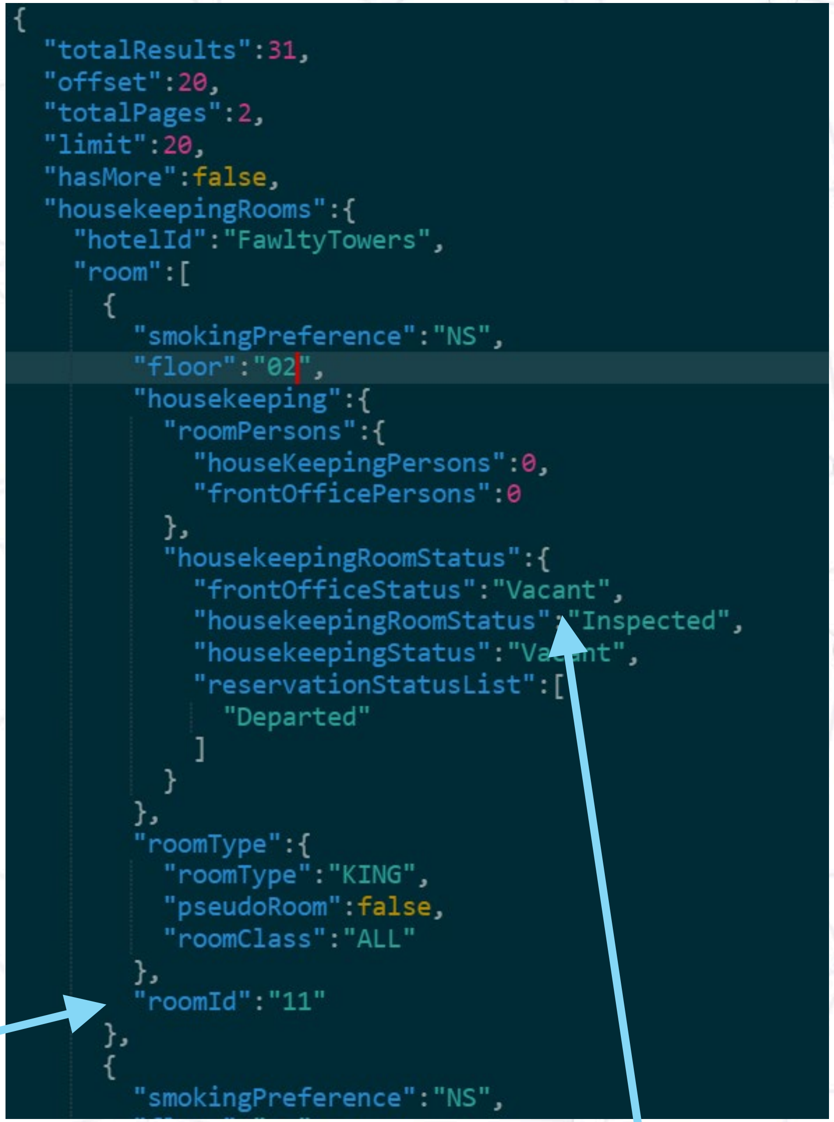
Alarm 261 acknowledged jason

JSON Path

Oracle Hospitality Web Services…

Hotel Room Occupancy Data
in < 15 Minutes

(The existing driver took over a month to develop)



```
{
  "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",
```

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

Recap: JSON Schema Feature

- Data Selection
  - Binding, Base Query, or Export Markers
- Query Data
  - Any BITable (***history:*** ***alarm:*** *transform:* *bql:* *neql:*)
- JSON Alarm Recipient
- Timestamp / Message Count Properties
- Tag / Facet Support
- Configuration
  - COV or On demand
  - Timestamp format, Numeric precision

NF 23

CONNECTING THE WORLD

Niagara Cloud Suite

Kevin Mamajek
Tridium

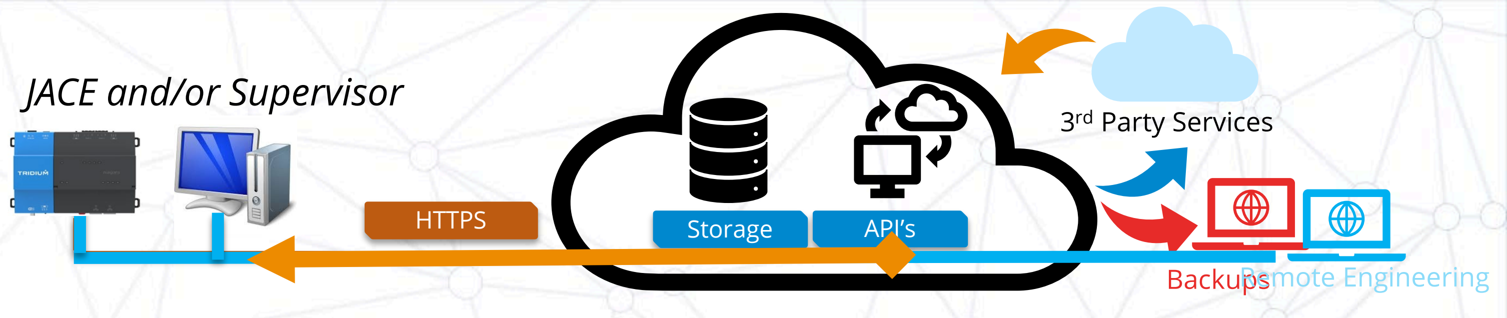
TRIDIUM

Niagara Cloud Suite

Overview

hello

Niagara Cloud Suite



Cloud Connection Services

Cloud Authentication

Encryption

Device Certs

IoT Cloud Connections

Secure Device Certificates, easy Onboarding, encryption

Niagara Data Services

Modeler

Histories

Schedule Histories

Alarm Histories

Secure Access to **Histories/Model** via APIs

Simple Historical **Graphs**

Microsoft Graph

Niagara Recover

Backup

Cloud Initiated Restore

Cloud Initiated Niagara Updates

Browser Access to **Backup files (5)**

Restore / Niagara **Updates Portal**

Niagara Remote

Wiresheet Access

Dashboard Access

Remote browser **Wiresheet Engineering**

Remote browser Niagara **Dashboards** (Px)

Remote browser **Manual Overrides**

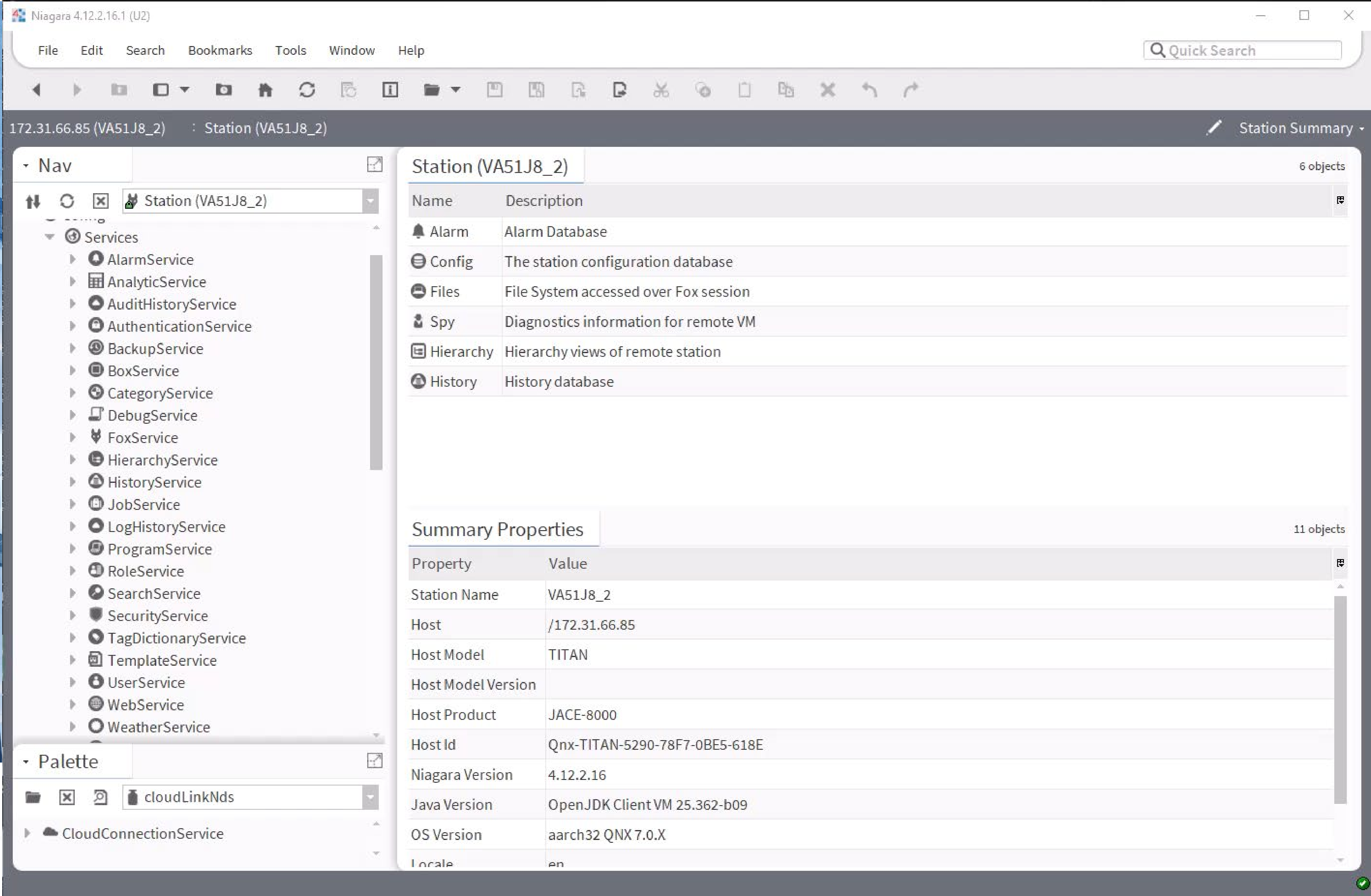
Niagara Control API's

Commands Back to Niagara

Secure **Commands** via APIs for cloud overrides and recommendations

Demonstration of

1. Registration
2. Model
3. History Selection
4. Activate History Channel
5. Confirm Backup Schedule



Services
- AlarmService
- AnalyticService
- AuditHistoryService
- AuthenticationService
- BackupService
- BoxService
- CategoryService
- DebugService
- FoxService
- HierarchyService
- HistoryService
- JobService
- LogHistoryService
- ProgramService
- RoleService
- SearchService
- SecurityService
- TagDictionaryService
- TemplateService
- UserService
- WebService
- WeatherService

Station (VA51J8_2) 6 objects

Name	Description
Alarm	Alarm Database
Config	The station configuration database
Files	File System accessed over Fox session
Spy	Diagnostics information for remote VM
Hierarchy	Hierarchy views of remote station
History	History database

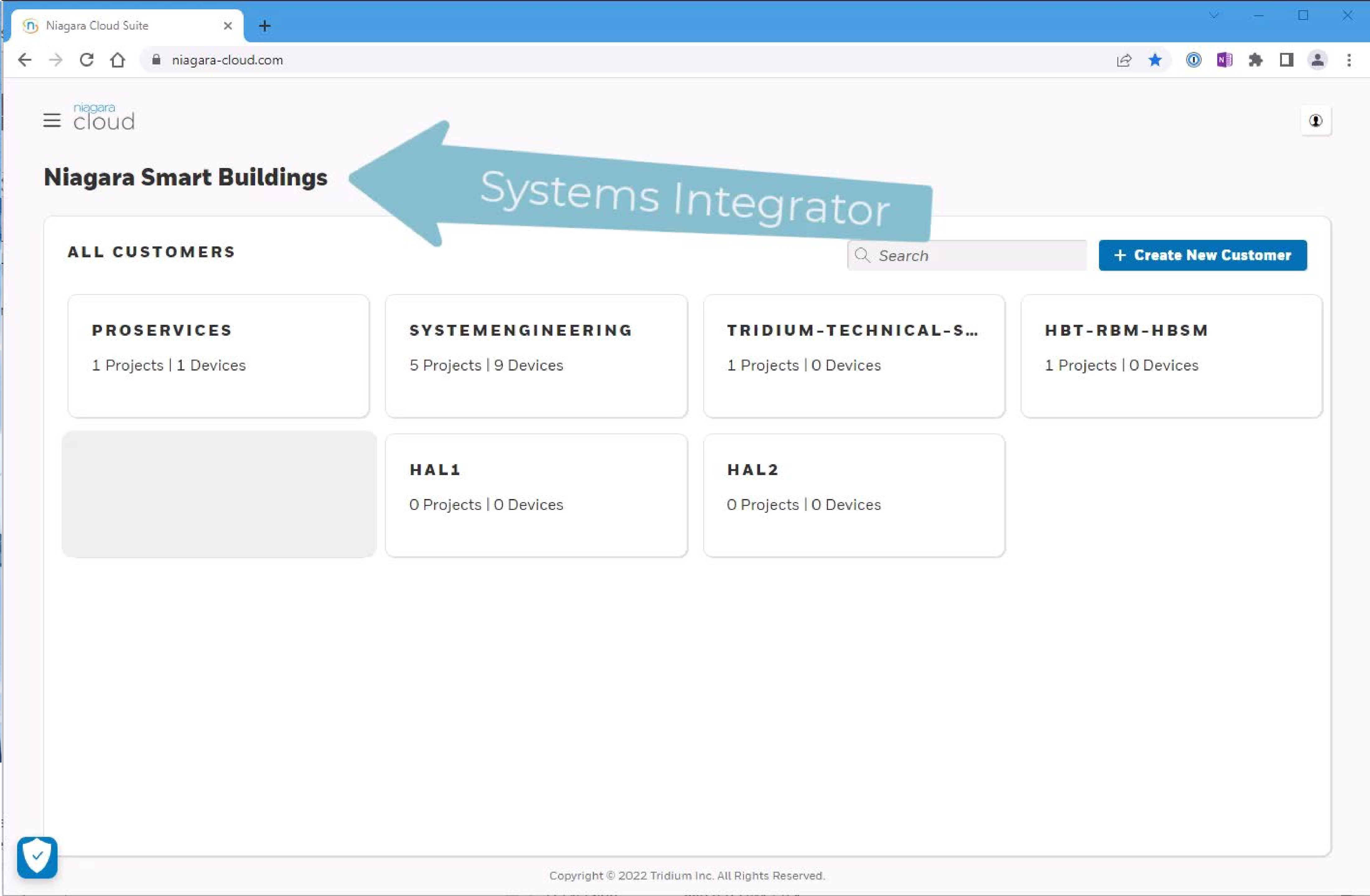
Summary Properties 11 objects

Property	Value
Station Name	VA51J8_2
Host	/172.31.66.85
Host Model	TITAN
Host Model Version	
Host Product	JACE-8000
Host Id	Qnx-TITAN-5290-78F7-0BE5-618E
Niagara Version	4.12.2.16
Java Version	OpenJDK Client VM 25.362-b09
OS Version	aarch32 QNX 7.0.X
Locale	en

NF 23

Demonstration of

1. Viewing History Data via Chart
2. Marking a backup Preferred
3. Downloading a backup for restore



Systems Integrator

Niagara Smart Buildings

ALL CUSTOMERS

PROSERVICES 1 Projects | 1 Devices	SYSTEMENGINEERING 5 Projects | 9 Devices	TRIDIUM-TECHNICAL-S… 1 Projects | 0 Devices	HBT-RBM-HBSM 1 Projects | 0 Devices
	HAL1 0 Projects | 0 Devices	HAL2 0 Projects | 0 Devices	

NF 23

CONNECTING
THE WORLD

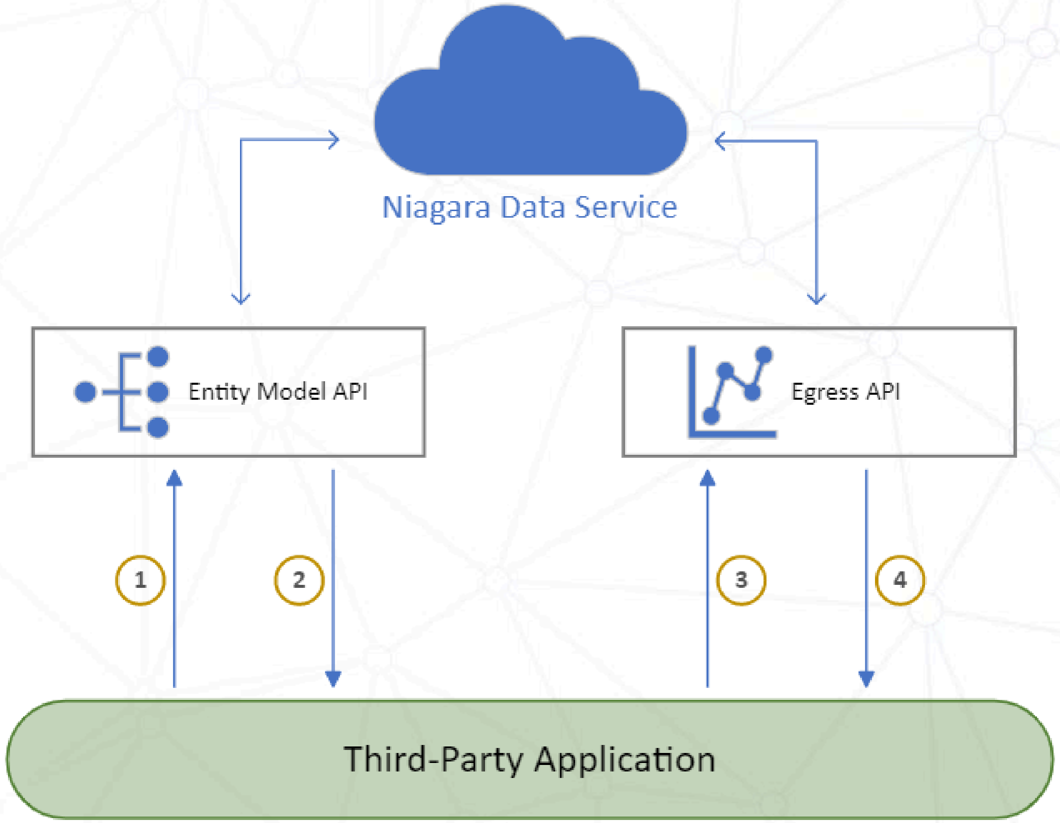
Getting Started with the Niagara Data Service API

Sarah Johnson

Tridium

TRIDIUM

NDS API Endpoints



1. Query the Entity Model API for data points based on point characteristics
2. EM API responds with **cloudIds** of points with those characteristics
3. Query Egress API with **cloudIds**
4. Receive historical data

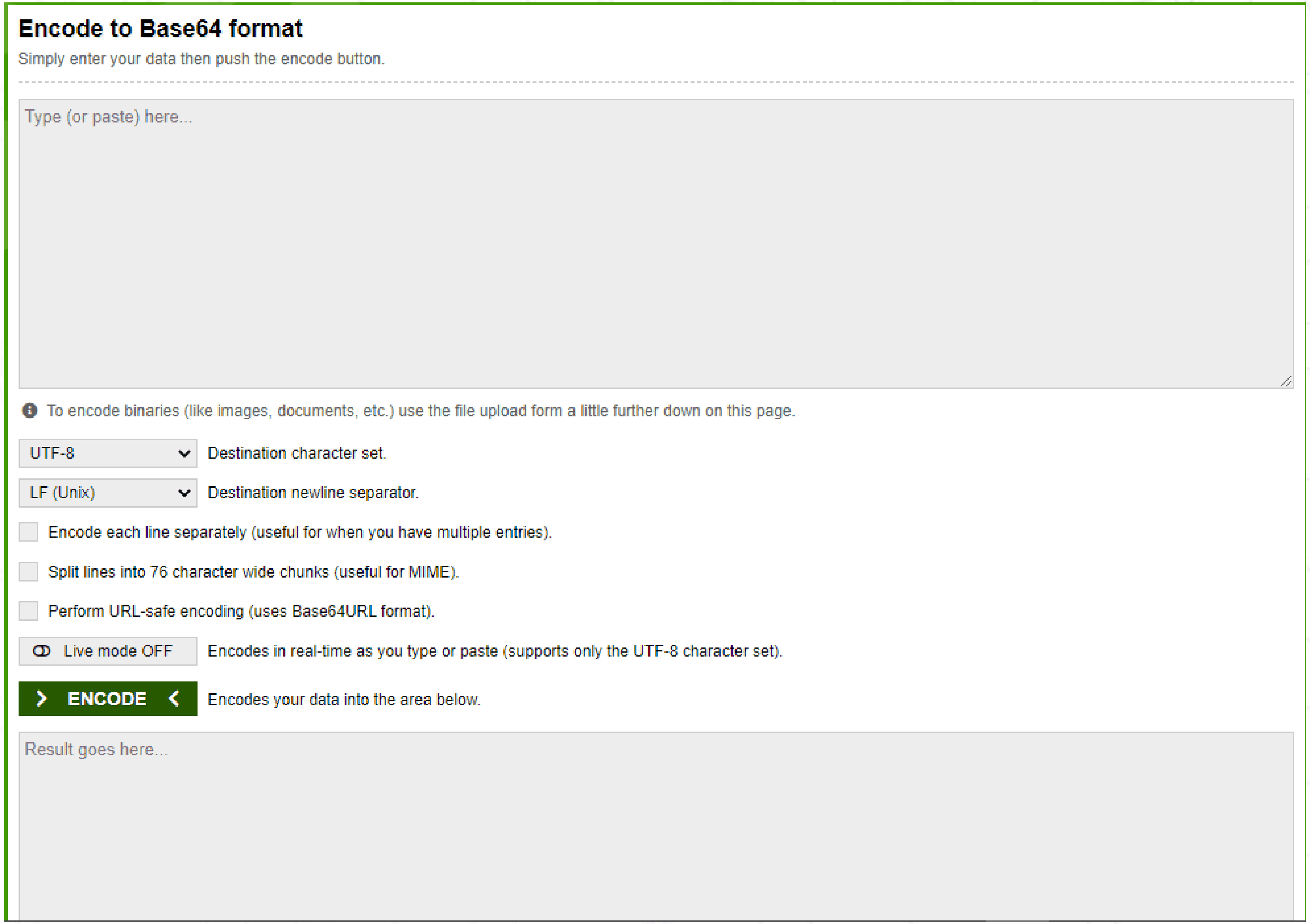
Getting Started

- Get required information from Tridium

Customer Id	Number
Client Id	String
Client Secret	*****************************
System GUID	String

- Get access token (**Postman**)
- Send first requests to API endpoints (**Swagger UI**)
- Integrate with application (**Grafana**)

Bearer Access Token



Encode to Base64 format

Simply enter your data then push the encode button.

Type (or paste) here...

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Destination character set.

LF (Unix) Destination newline separator.

□ Encode each line separately (useful for when you have multiple entries).

□ Split lines into 76 character wide chunks (useful for MIME).

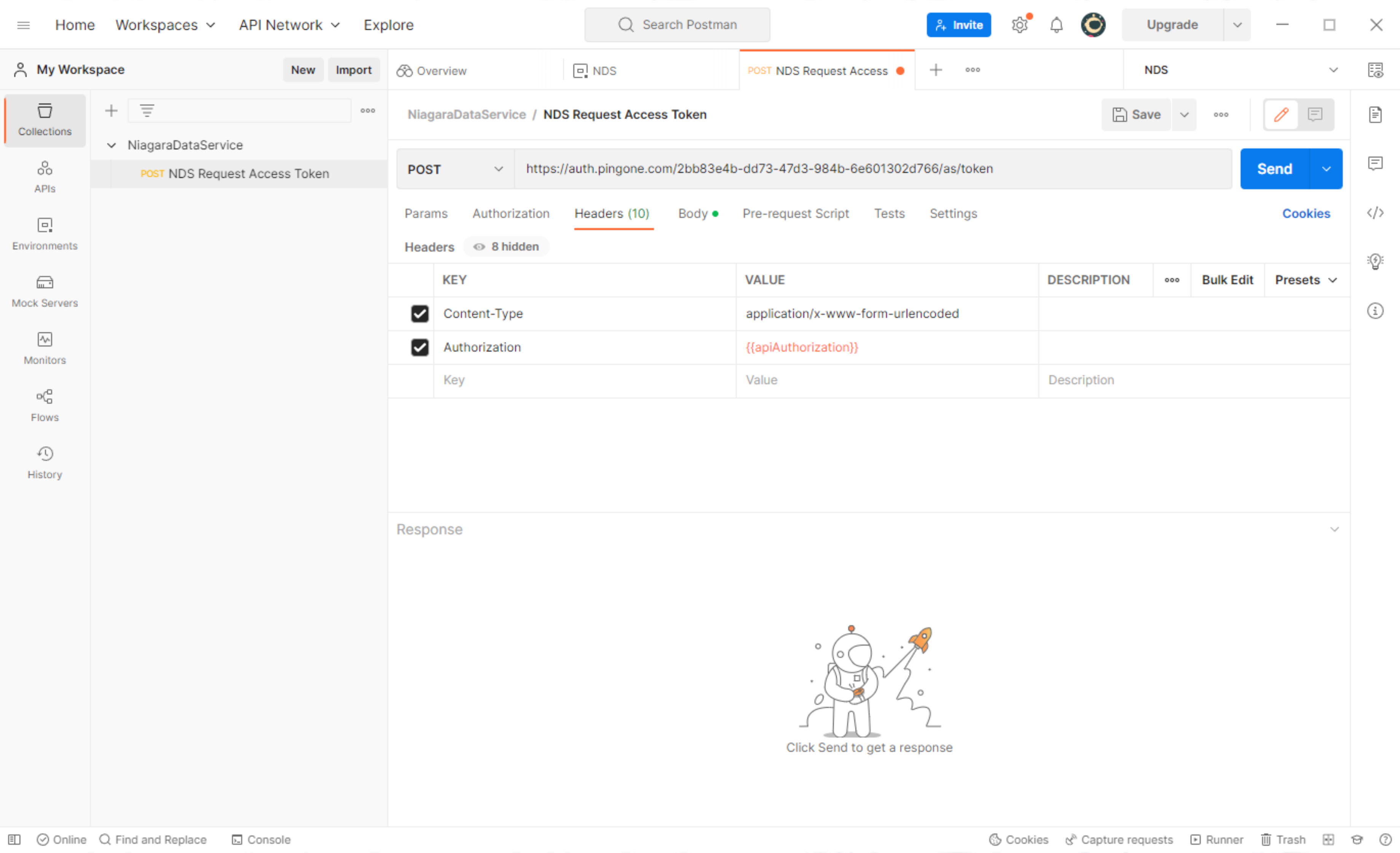
□ Perform URL-safe encoding (uses Base64URL format).

Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

ENCODE Encodes your data into the area below.

Result goes here...

Bearer Access Token



NiagaraDataService / NDS Request Access Token

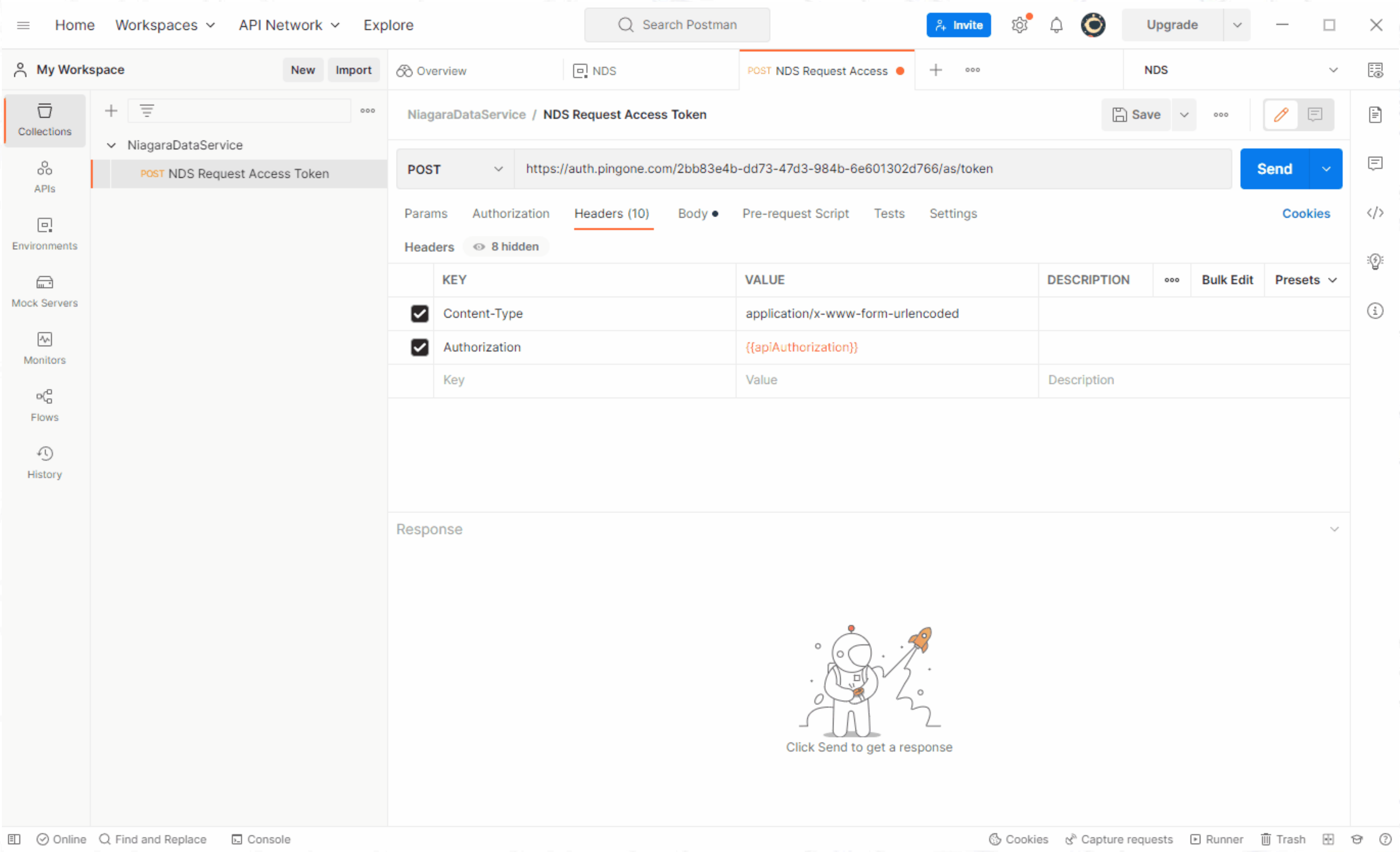
POST https://auth.pingone.com/2bb83e4b-dd73-47d3-984b-6e601302d766/as/token

	KEY	VALUE	DESCRIPTION
☑	Content-Type	application/x-www-form-urlencoded	
☑	Authorization	{{apiAuthorization}}	
	Key	Value	Description

NF 23

TRIDIUM

Bearer Access Token



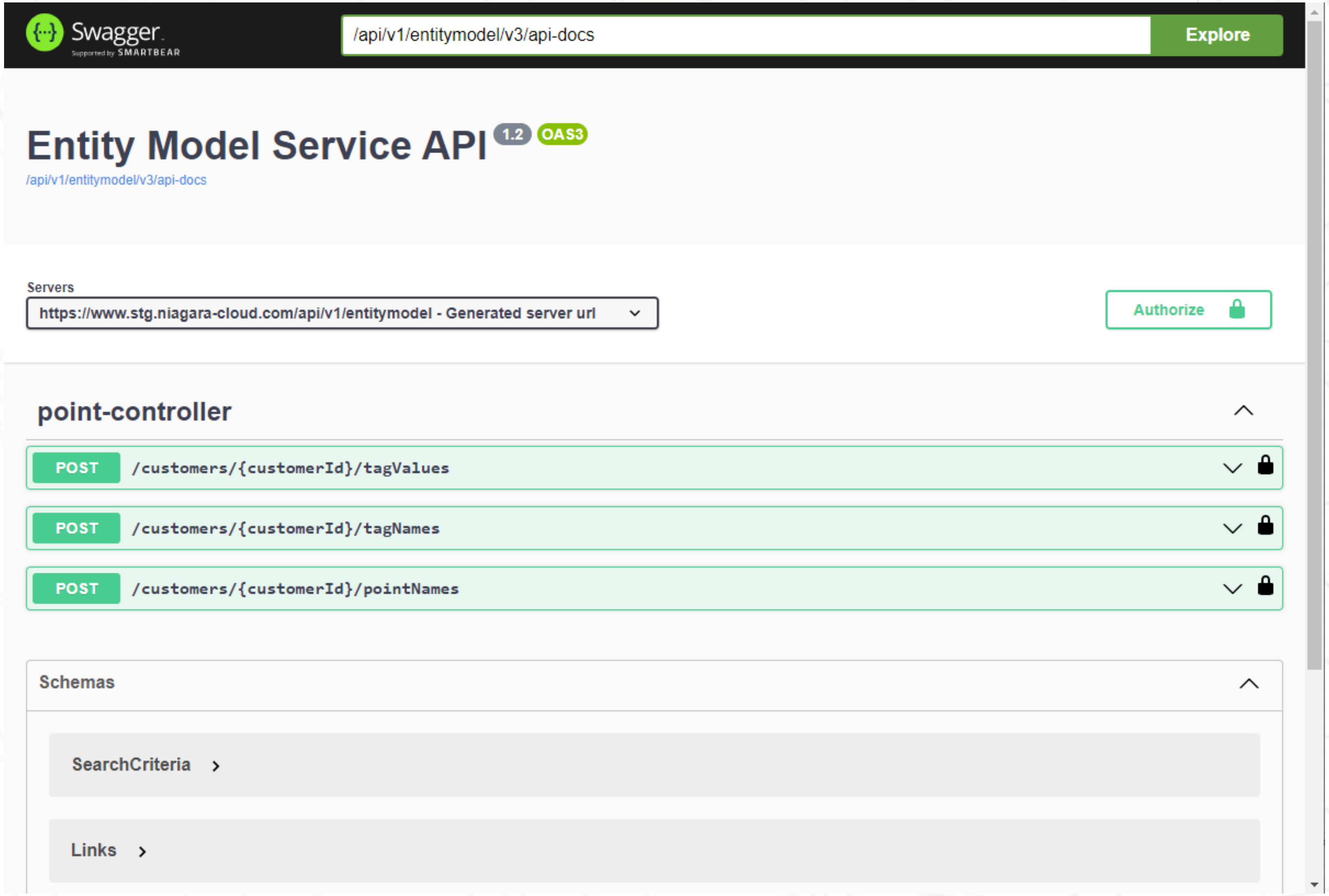
NiagaraDataService / NDS Request Access Token

POST https://auth.pingone.com/2bb83e4b-dd73-47d3-984b-6e601302d766/as/token

Headers 8 hidden

	KEY	VALUE	DESCRIPTION
☑	Content-Type	application/x-www-form-urlencoded	
☑	Authorization	{{apiAuthorization}}	
	Key	Value	Description

Sending First Requests



Swagger Supported by SMARTBEAR

/api/v1/entitymodel/v3/api-docs

Explore

Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers

https://www.stg.niagara-cloud.com/api/v1/entitymodel - Generated server url

Authorize

point-controller

POST `/customers/{customerId}/tagValues`

POST `/customers/{customerId}/tagNames`

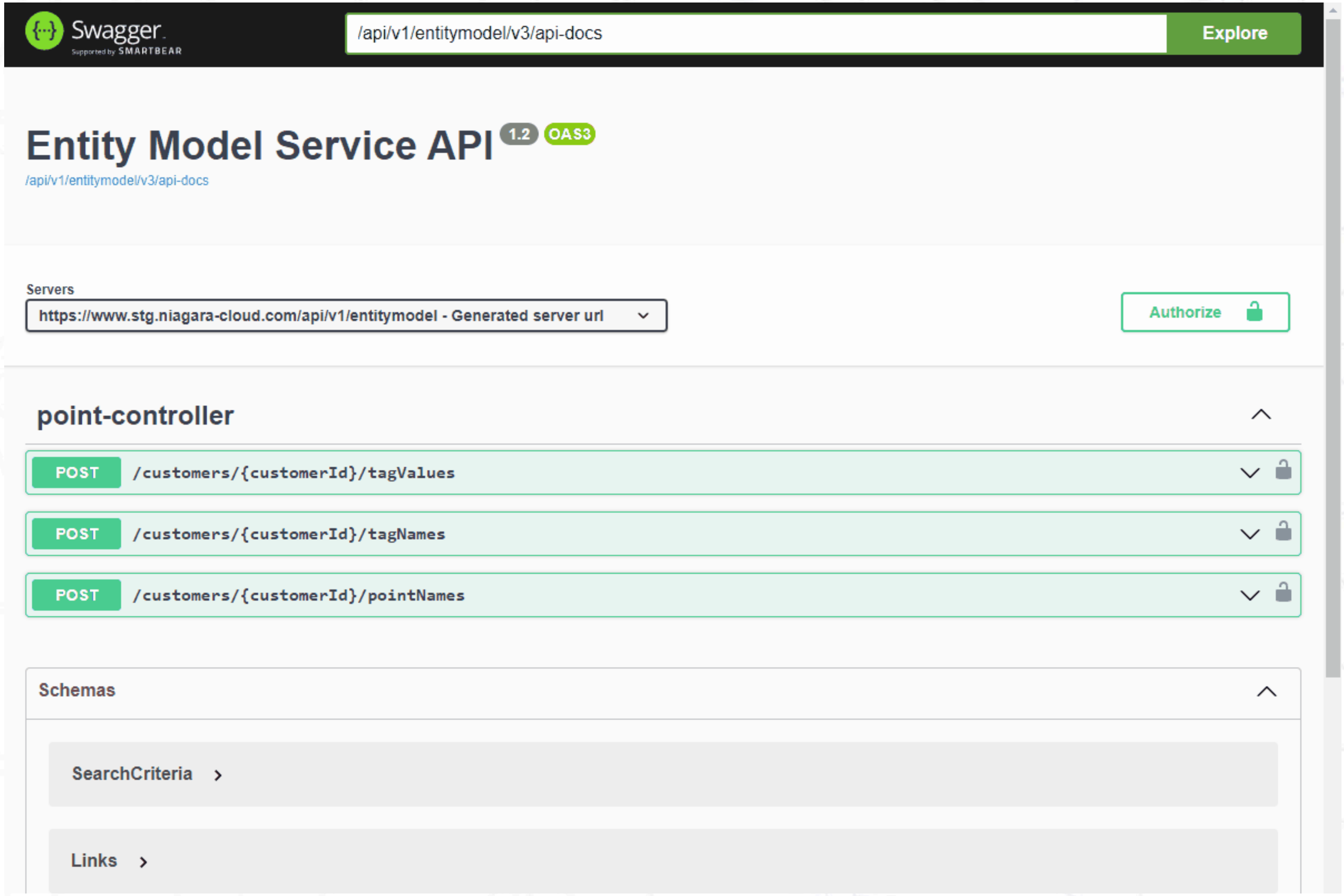
POST `/customers/{customerId}/pointNames`

Schemas

SearchCriteria >

Links >

Sending First Requests



/api/v1/entitymodel/v3/api-docs

Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers

https://www.stg.niagara-cloud.com/api/v1/entitymodel - Generated server url

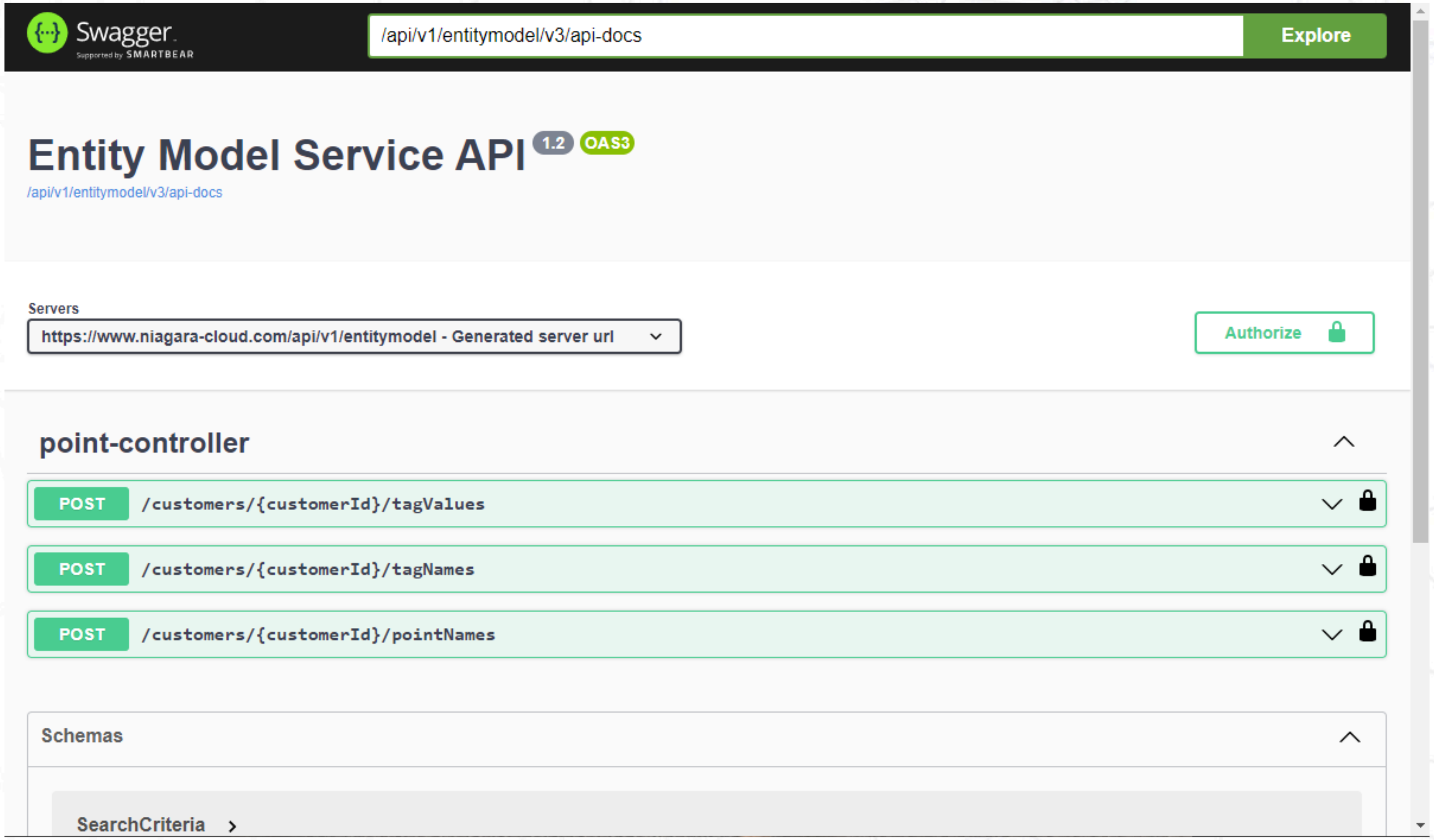
point-controller

- POST /customers/{customerId}/tagValues
- POST /customers/{customerId}/tagNames
- POST /customers/{customerId}/pointNames

Schemas

- SearchCriteria
- Links

Sending First Requests - pointNames



/api/v1/entitymodel/v3/api-docs

Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers: https://www.niagara-cloud.com/api/v1/entitymodel - Generated server url

point-controller

POST /customers/{customerId}/tagValues

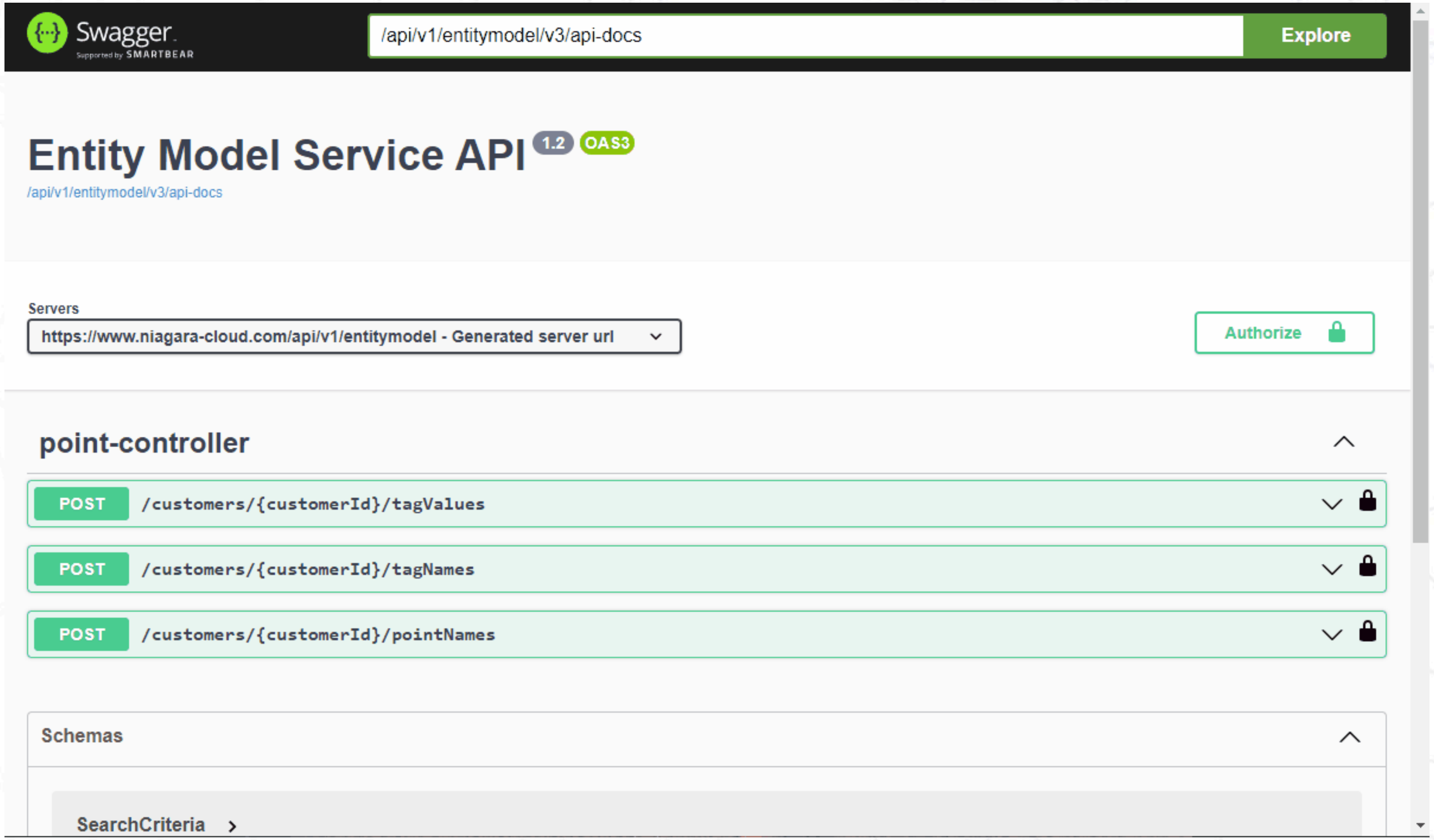
POST /customers/{customerId}/tagNames

POST /customers/{customerId}/pointNames

Schemas

SearchCriteria >

Sending First Requests - pointNames



Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers

https://www.niagara-cloud.com/api/v1/entitymodel - Generated server url

point-controller

POST /customers/{customerId}/tagValues

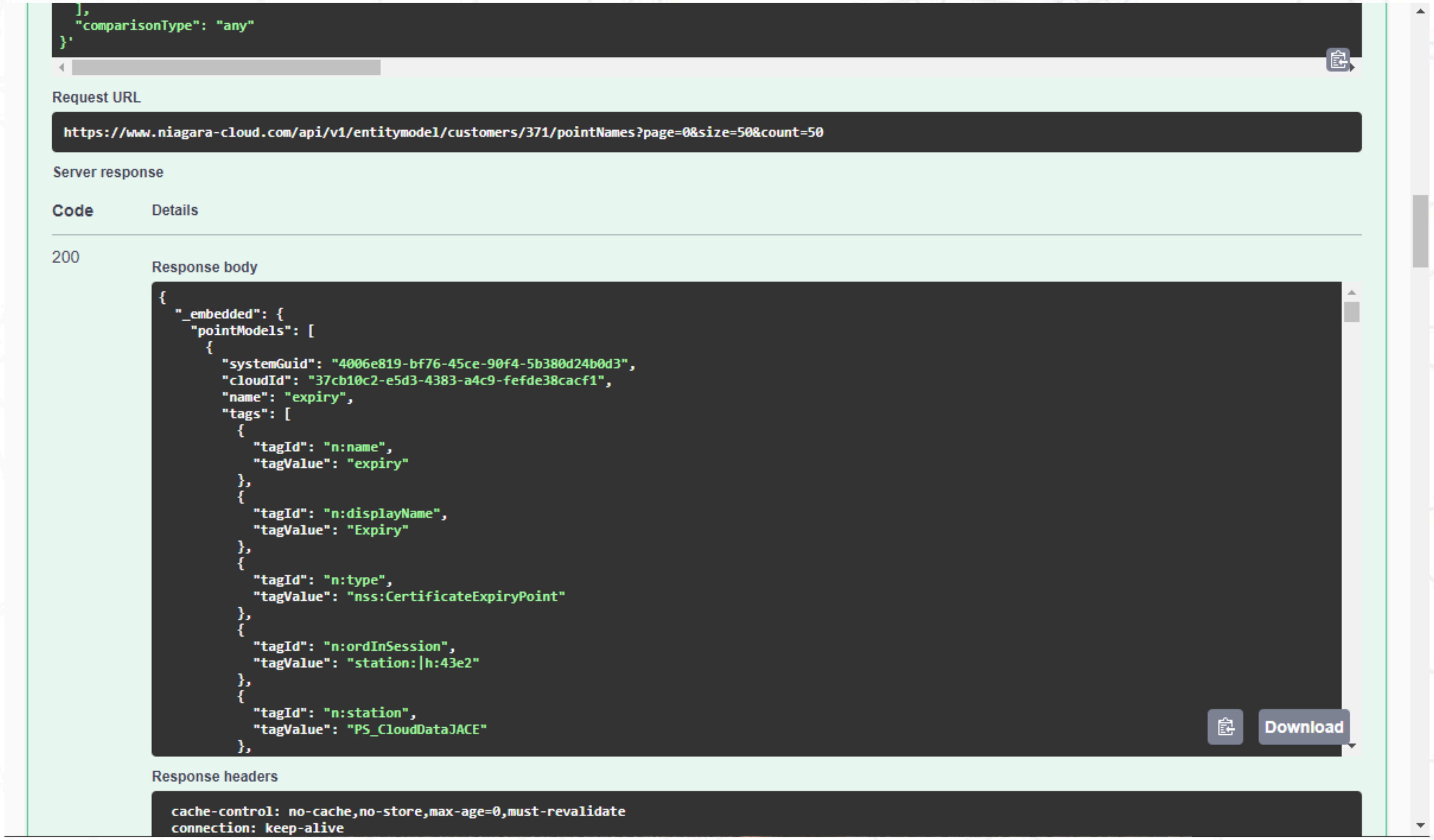
POST /customers/{customerId}/tagNames

POST /customers/{customerId}/pointNames

Schemas

SearchCriteria

Sending First Requests - pointNames



```
  ],
  "comparisonType": "any"
}'
```

Request URL

```
https://www.niagara-cloud.com/api/v1/entitymodel/customers/371/pointNames?page=0&size=50&count=50
```

Server response

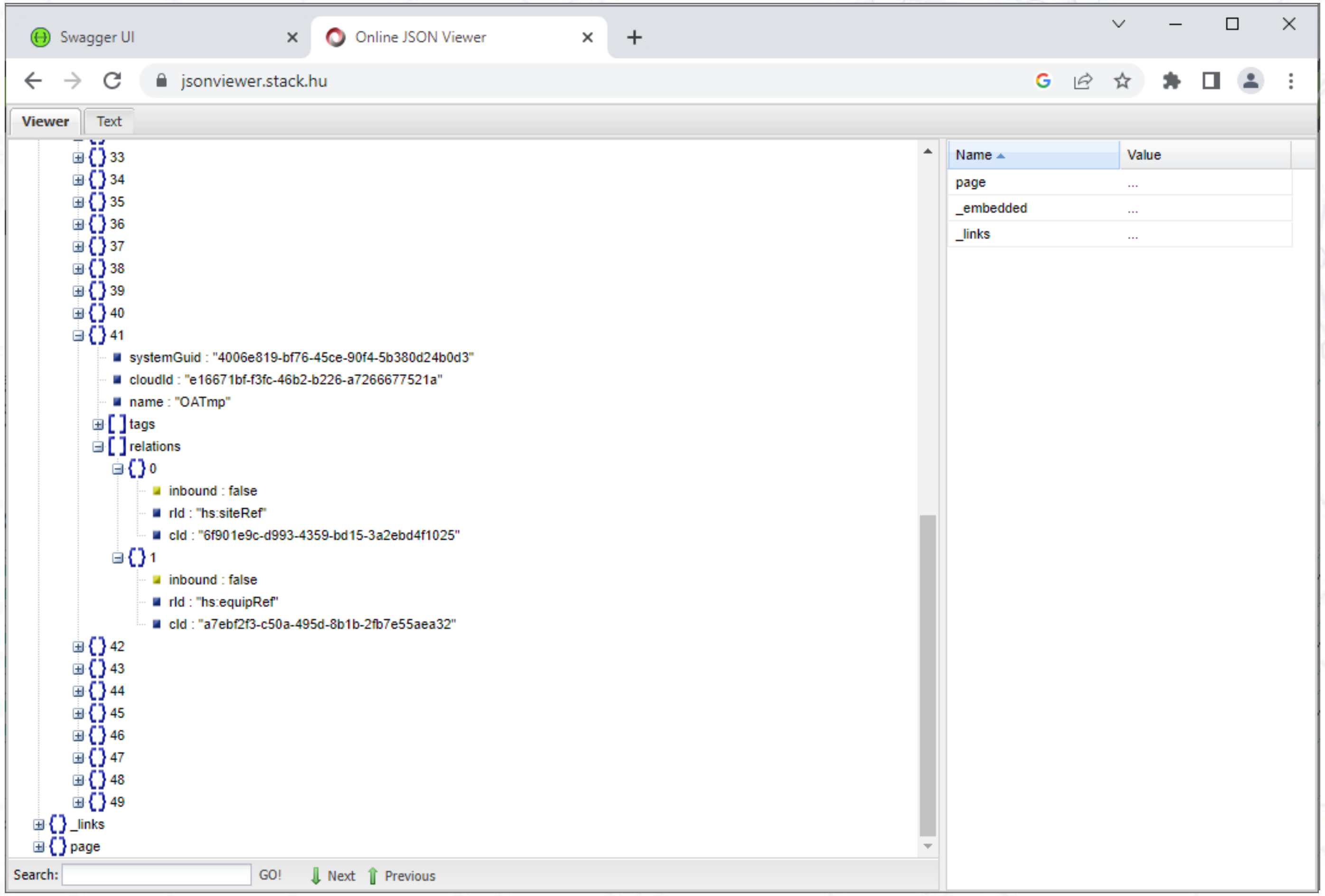
Code	Details
200	Response body

```
{
  "_embedded": {
    "pointModels": [
      {
        "systemGuid": "4006e819-bf76-45ce-90f4-5b380d24b0d3",
        "cloudId": "37cb10c2-e5d3-4383-a4c9-fefde38cacf1",
        "name": "expiry",
        "tags": [
          {
            "tagId": "n:name",
            "tagValue": "expiry"
          },
          {
            "tagId": "n:displayName",
            "tagValue": "Expiry"
          },
          {
            "tagId": "n:type",
            "tagValue": "nss:CertificateExpiryPoint"
          },
          {
            "tagId": "n:ordInSession",
            "tagValue": "station:|h:43e2"
          },
          {
            "tagId": "n:station",
            "tagValue": "PS_CloudDataJACE"
          },
```

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
```

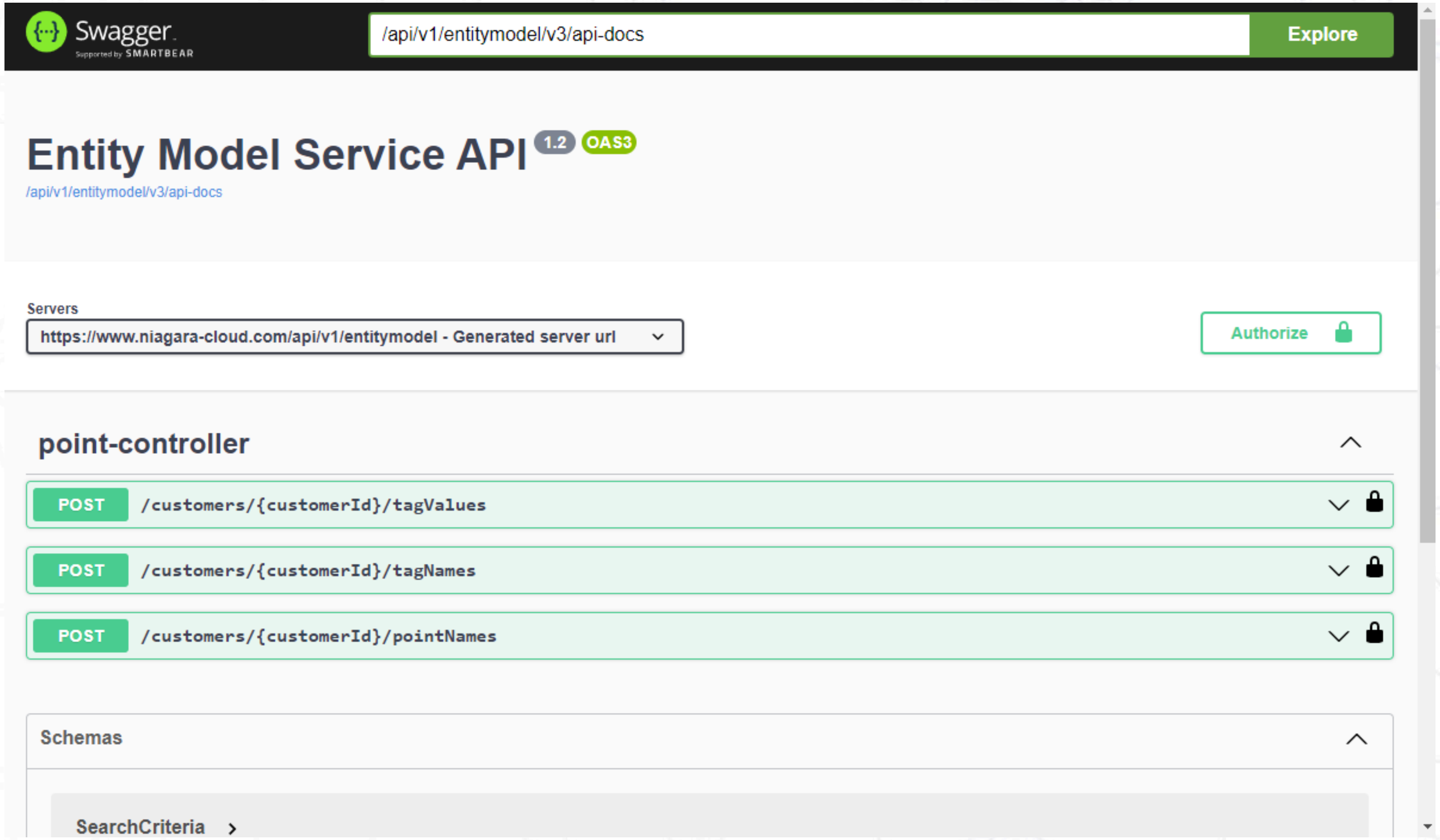
Sending First Requests - pointNames



```
    {} 33
    {} 34
    {} 35
    {} 36
    {} 37
    {} 38
    {} 39
    {} 40
    {} 41
        systemGuid : "4006e819-bf76-45ce-90f4-5b380d24b0d3"
        cloudId : "e16671bf-f3fc-46b2-b226-a7266677521a"
        name : "OATmp"
      [] tags
      [] relations
        {} 0
            inbound : false
            rId : "hs:siteRef"
            cId : "6f901e9c-d993-4359-bd15-3a2ebd4f1025"
        {} 1
            inbound : false
            rId : "hs:equipRef"
            cId : "a7ebf2f3-c50a-495d-8b1b-2fb7e55aea32"
    {} 42
    {} 43
    {} 44
    {} 45
    {} 46
    {} 47
    {} 48
    {} 49
{} _links
{} page
```

Name ▲	Value
page	...
_embedded	...
_links	...

Sending First Requests - RmTmpSp



/api/v1/entitymodel/v3/api-docs

Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers

https://www.niagara-cloud.com/api/v1/entitymodel - Generated server url

point-controller

POST /customers/{customerId}/tagValues

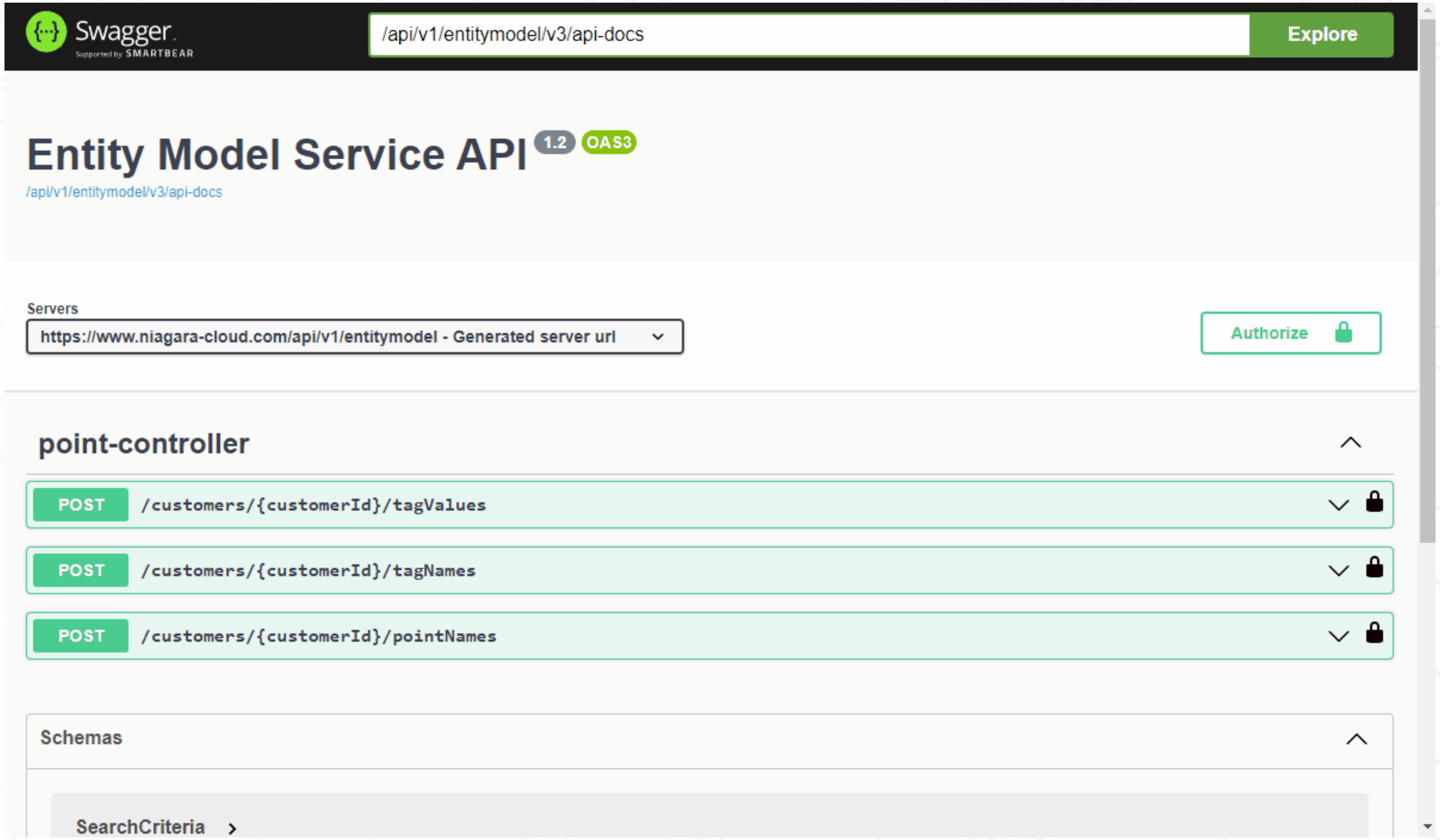
POST /customers/{customerId}/tagNames

POST /customers/{customerId}/pointNames

Schemas

SearchCriteria

Sending First Requests - RmTmpSp



/api/v1/entitymodel/v3/api-docs

Entity Model Service API

/api/v1/entitymodel/v3/api-docs

Servers

https://www.niagara-cloud.com/api/v1/entitymodel - Generated server url

point-controller

POST /customers/{customerId}/tagValues

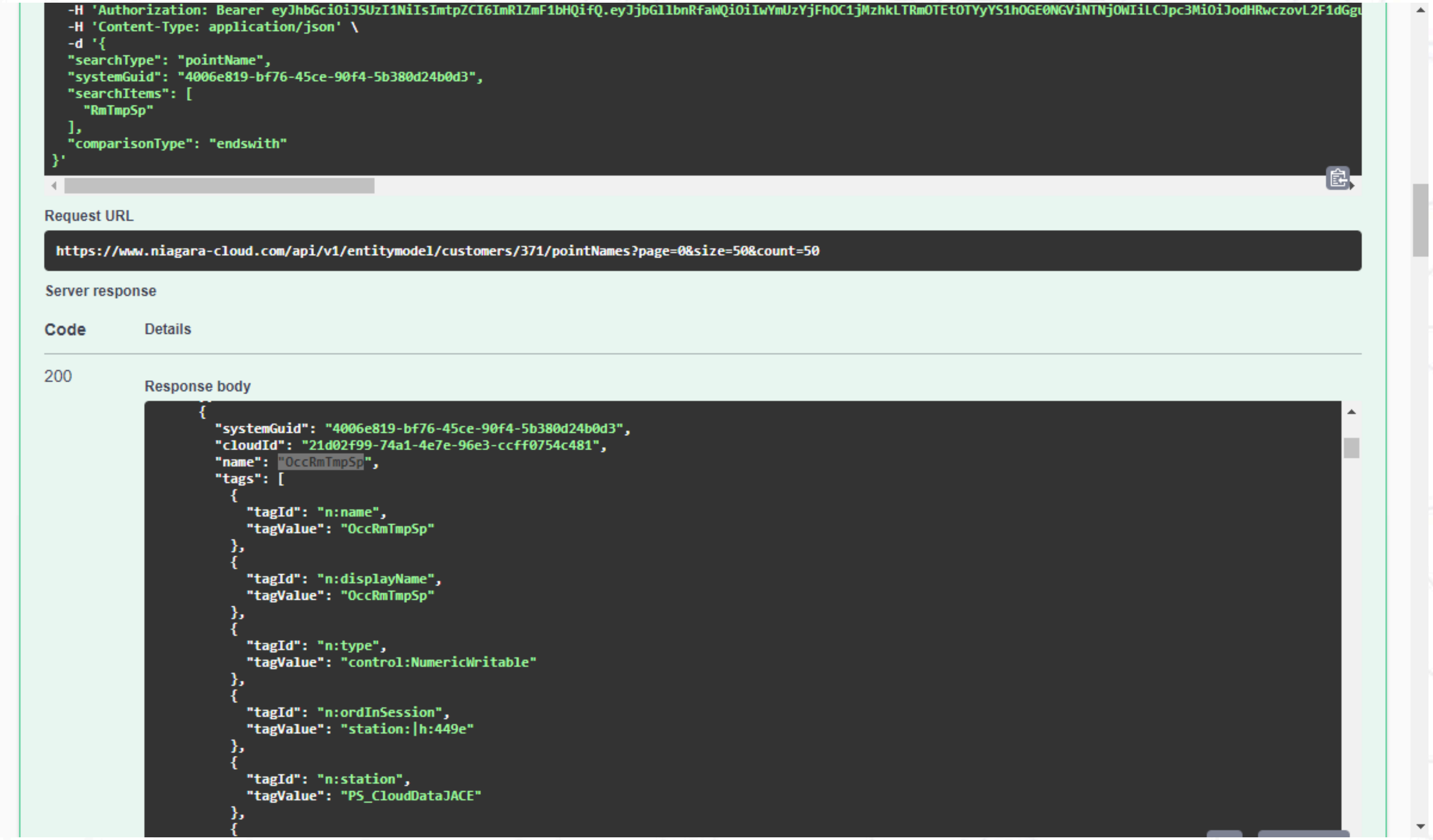
POST /customers/{customerId}/tagNames

POST /customers/{customerId}/pointNames

Schemas

SearchCriteria

Sending First Requests - RmTmpSp



```
  -H 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6ImRlZmF1bHQifQ.eyJjbGllbnRfaWQiOiIwYmUzYjFhOC1jMzhkLTRmOTEtOTYyYS1hOGE0NGViNTNjOWIiLCJpc3MiOiJodHRwczovL2F1dGgu
  -H 'Content-Type: application/json' \
  -d '{
  "searchType": "pointName",
  "systemGuid": "4006e819-bf76-45ce-90f4-5b380d24b0d3",
  "searchItems": [
    "RmTmpSp"
  ],
  "comparisonType": "endswith"
}'
```

Request URL

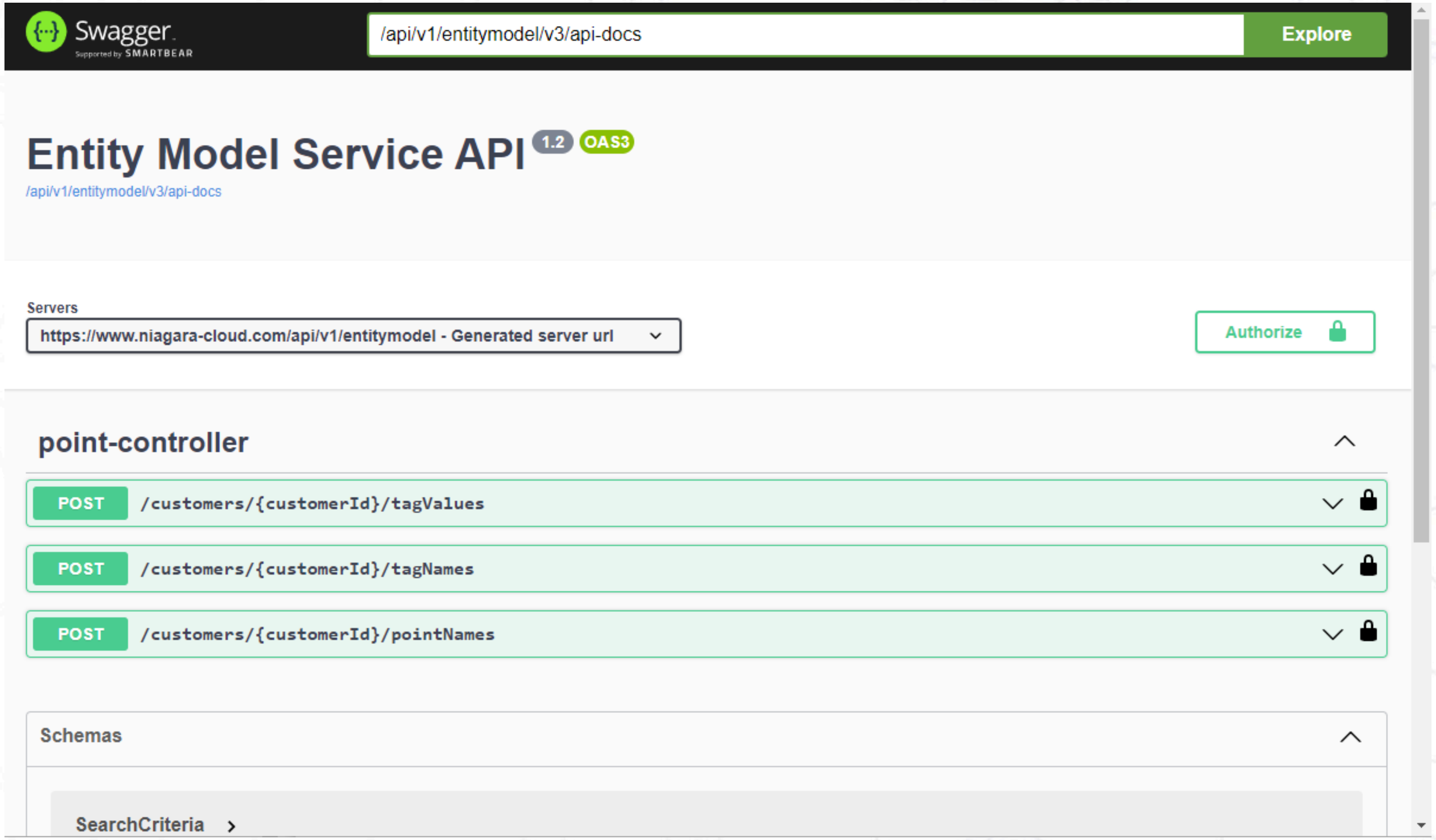
```
https://www.niagara-cloud.com/api/v1/entitymodel/customers/371/pointNames?page=0&size=50&count=50
```

Server response

Code	Details
200	Response body

```
      {
        "systemGuid": "4006e819-bf76-45ce-90f4-5b380d24b0d3",
        "cloudId": "21d02f99-74a1-4e7e-96e3-ccff0754c481",
        "name": "OccRmTmpSp",
        "tags": [
          {
            "tagId": "n:name",
            "tagValue": "OccRmTmpSp"
          },
          {
            "tagId": "n:displayName",
            "tagValue": "OccRmTmpSp"
          },
          {
            "tagId": "n:type",
            "tagValue": "control:NumericWritable"
          },
          {
            "tagId": "n:ordInSession",
            "tagValue": "station:|h:449e"
          },
          {
            "tagId": "n:station",
            "tagValue": "PS_CloudDataJACE"
          },
          {
```

Sending First Requests - tagNames



Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers

https://www.niagara-cloud.com/api/v1/entitymodel - Generated server url

point-controller

POST `/customers/{customerId}/tagValues`

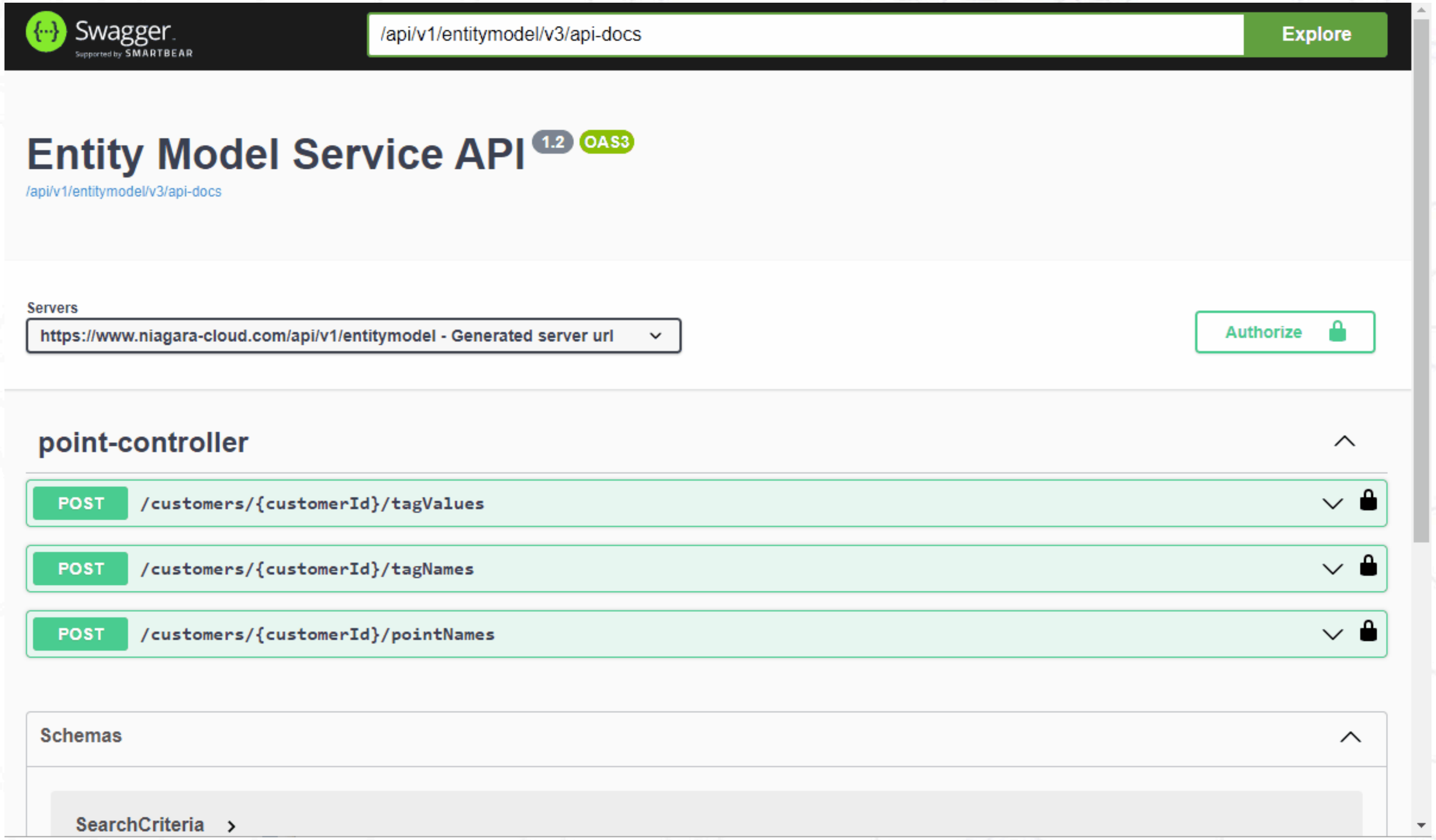
POST `/customers/{customerId}/tagNames`

POST `/customers/{customerId}/pointNames`

Schemas

SearchCriteria >

Sending First Requests - tagNames



/api/v1/entitymodel/v3/api-docs

Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers

https://www.niagara-cloud.com/api/v1/entitymodel - Generated server url

point-controller

POST /customers/{customerId}/tagValues

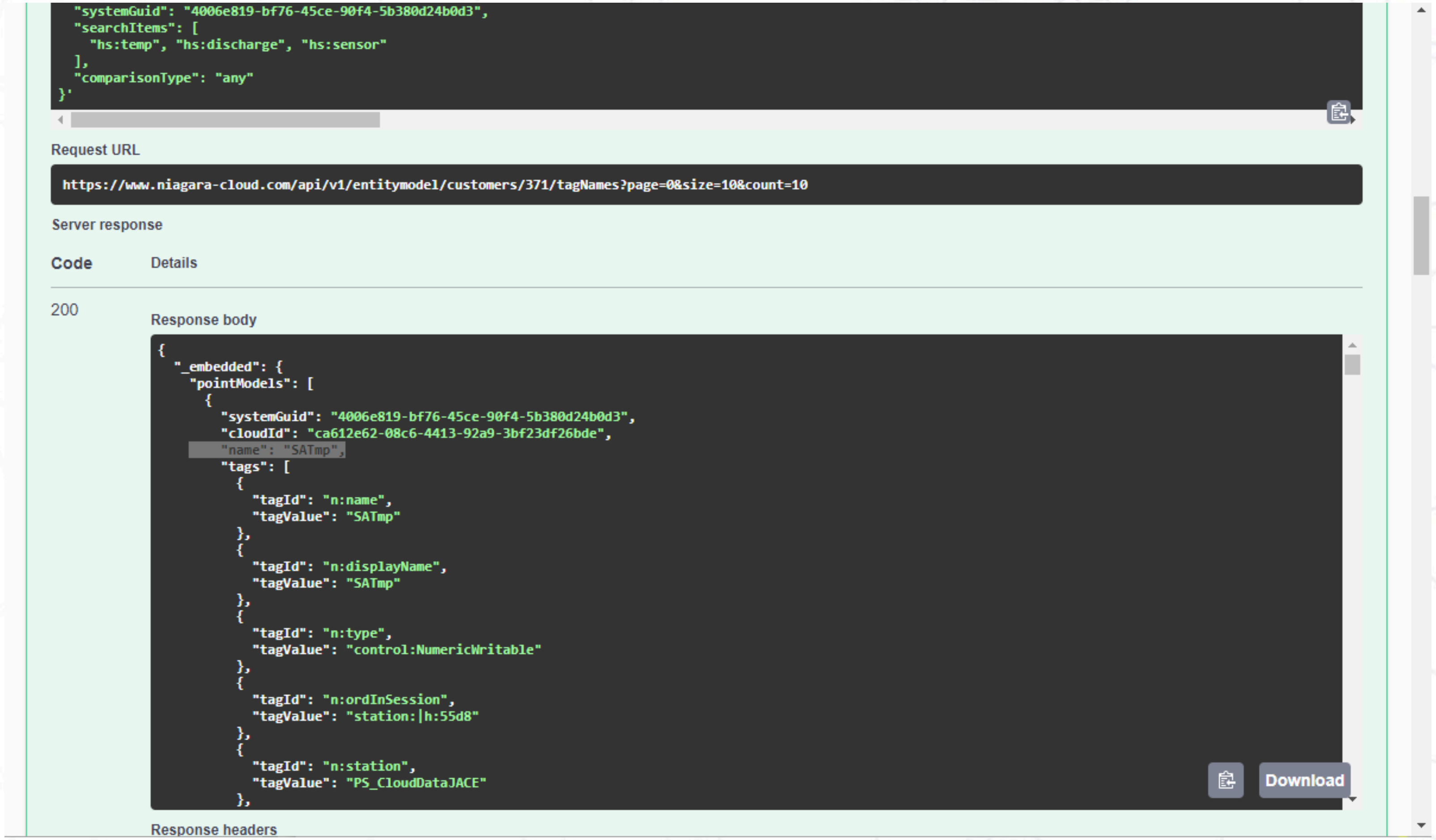
POST /customers/{customerId}/tagNames

POST /customers/{customerId}/pointNames

Schemas

SearchCriteria >

Sending First Requests - tagNames



```
  "systemGuid": "4006e819-bf76-45ce-90f4-5b380d24b0d3",
  "searchItems": [
    "hs:temp", "hs:discharge", "hs:sensor"
  ],
  "comparisonType": "any"
}'
```

Request URL

```
https://www.niagara-cloud.com/api/v1/entitymodel/customers/371/tagNames?page=0&size=10&count=10
```

Server response

Code Details

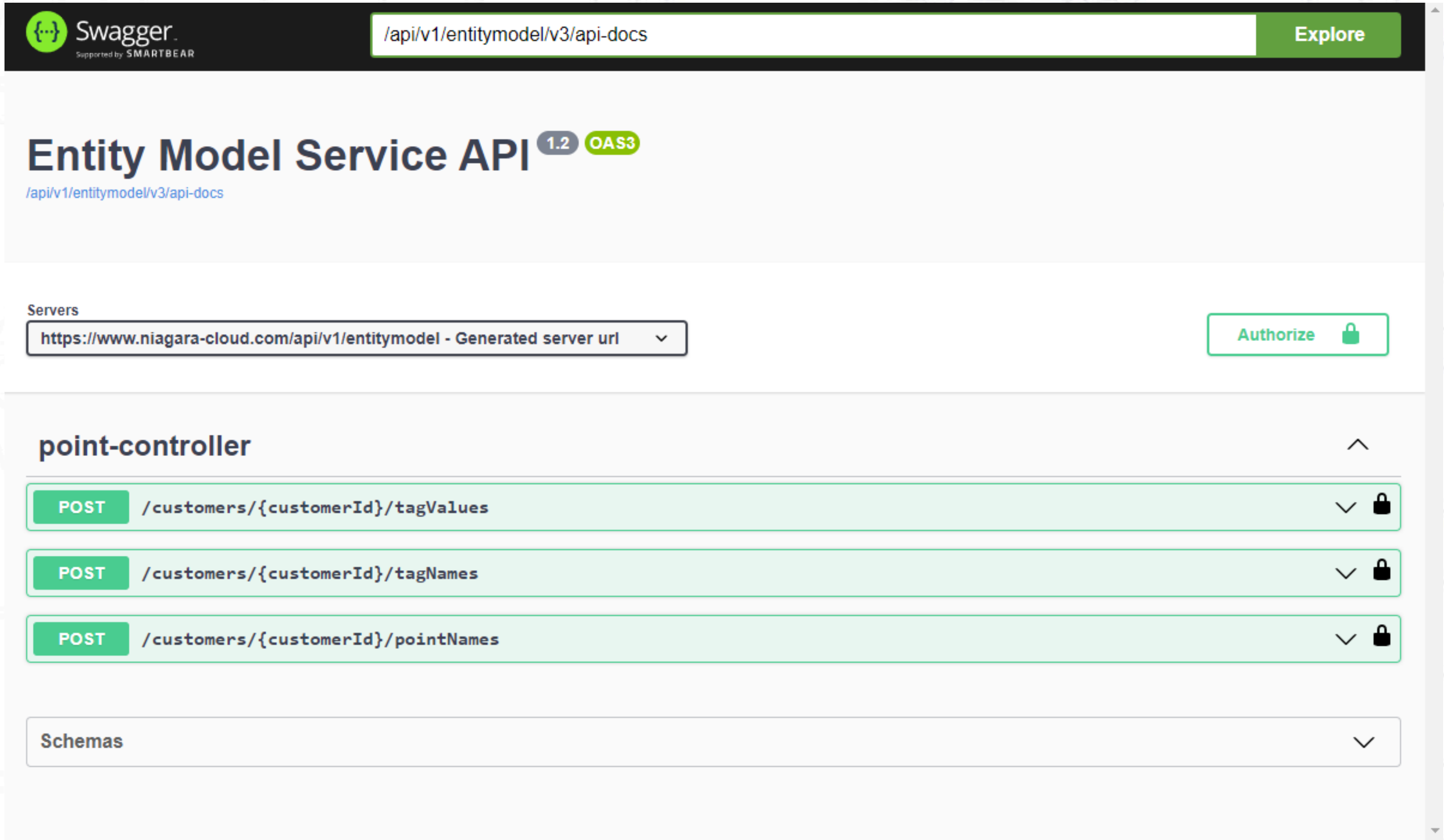
200

Response body

```
{
  "_embedded": {
    "pointModels": [
      {
        "systemGuid": "4006e819-bf76-45ce-90f4-5b380d24b0d3",
        "cloudId": "ca612e62-08c6-4413-92a9-3bf23df26bde",
        "name": "SATmp",
        "tags": [
          {
            "tagId": "n:name",
            "tagValue": "SATmp"
          },
          {
            "tagId": "n:displayName",
            "tagValue": "SATmp"
          },
          {
            "tagId": "n:type",
            "tagValue": "control:NumericWritable"
          },
          {
            "tagId": "n:ordInSession",
            "tagValue": "station:|h:55d8"
          },
          {
            "tagId": "n:station",
            "tagValue": "PS_CloudDataJACE"
          },
```

Response headers

Sending First Requests: tagValues



/api/v1/entitymodel/v3/api-docs

Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers

https://www.niagara-cloud.com/api/v1/entitymodel - Generated server url

point-controller

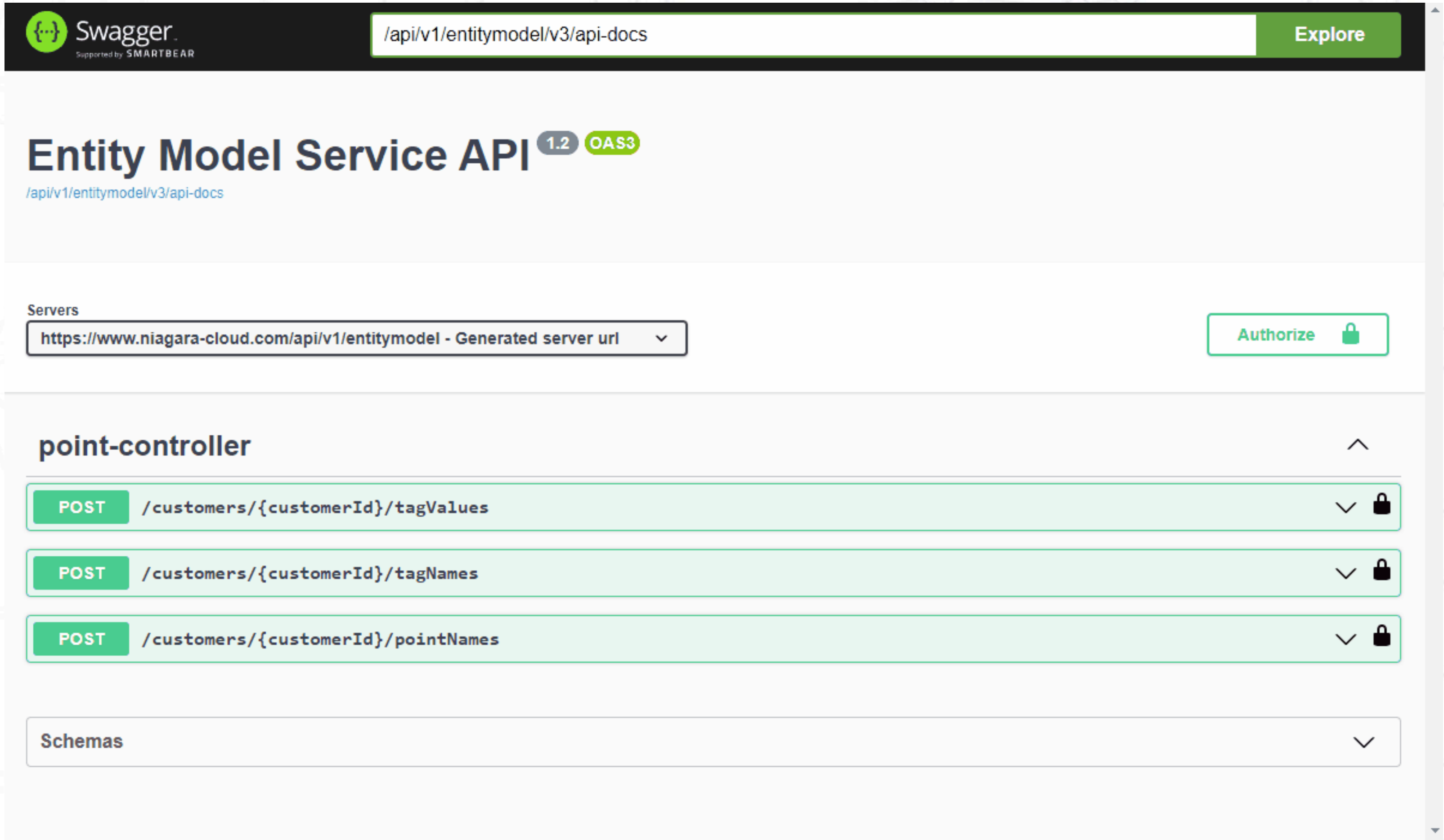
POST `/customers/{customerId}/tagValues`

POST `/customers/{customerId}/tagNames`

POST `/customers/{customerId}/pointNames`

Schemas

Sending First Requests: tagValues



/api/v1/entitymodel/v3/api-docs

Entity Model Service API 1.2 OAS3

/api/v1/entitymodel/v3/api-docs

Servers

https://www.niagara-cloud.com/api/v1/entitymodel - Generated server url

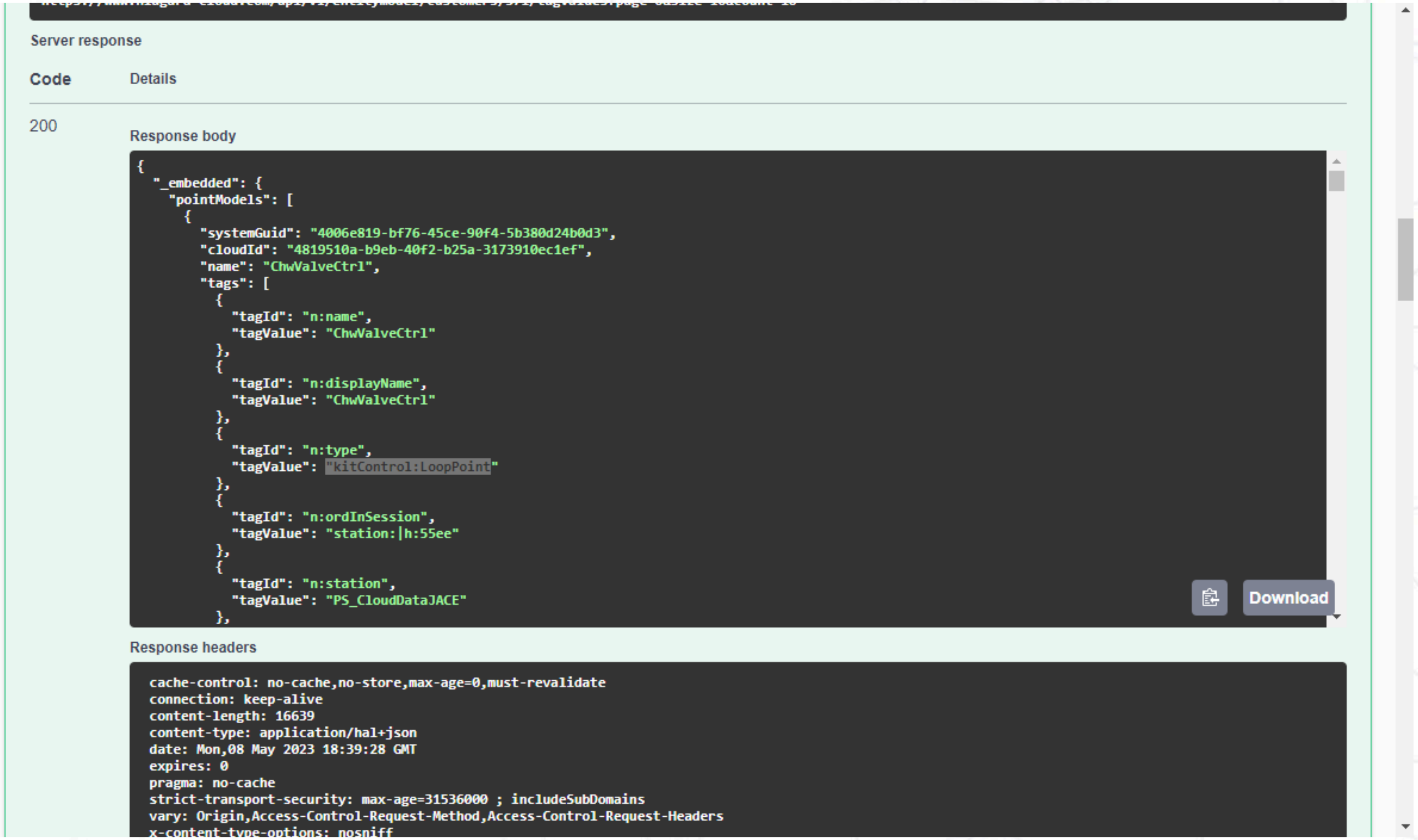
point-controller

POST `/customers/{customerId}/tagValues`

POST `/customers/{customerId}/tagNames`

POST `/customers/{customerId}/pointNames`

Sending First Requests: tagValues



Server response

Code Details

200

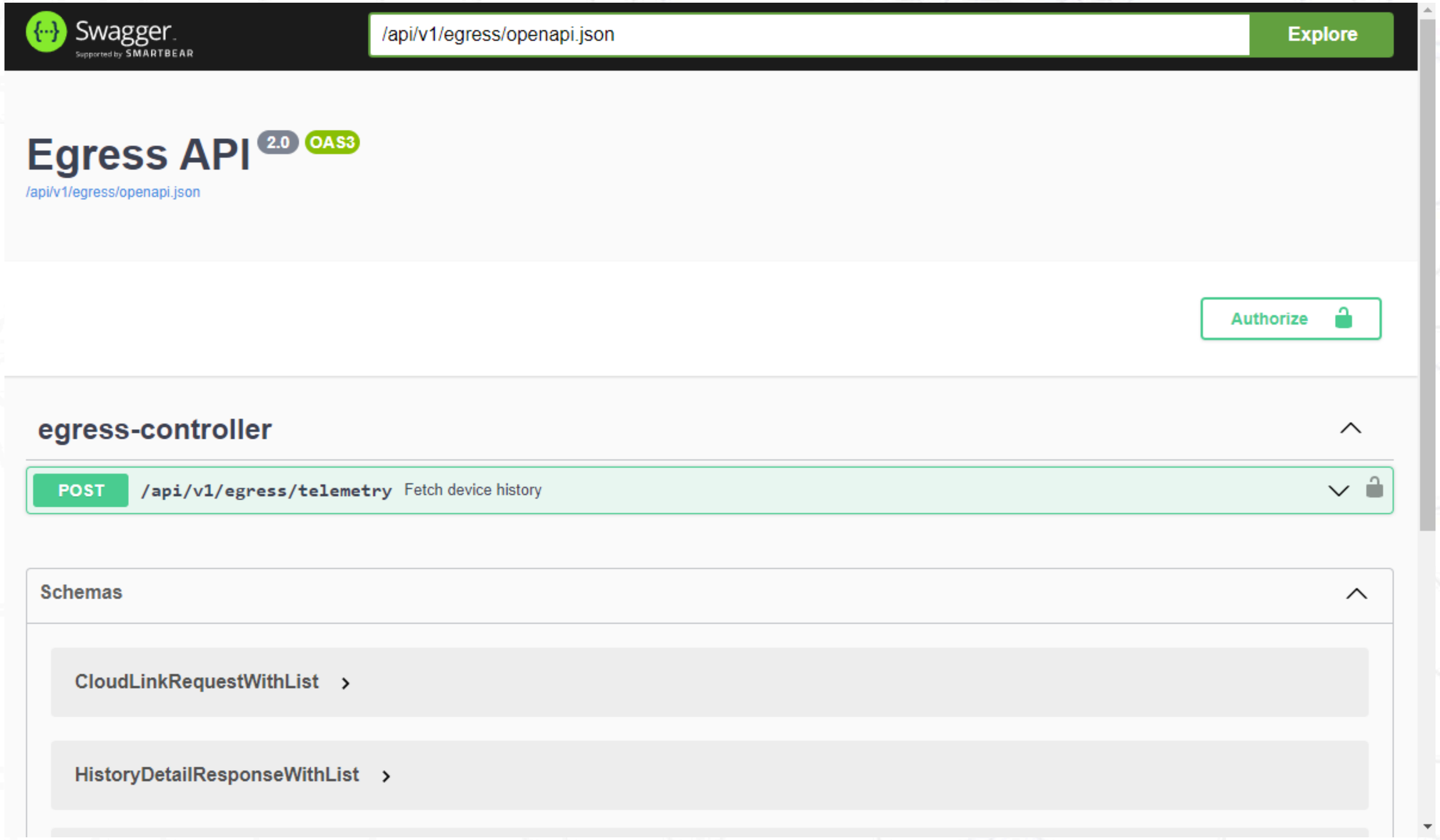
Response body

```
{
  "_embedded": {
    "pointModels": [
      {
        "systemGuid": "4006e819-bf76-45ce-90f4-5b380d24b0d3",
        "cloudId": "4819510a-b9eb-40f2-b25a-3173910ec1ef",
        "name": "ChwValveCtrl",
        "tags": [
          {
            "tagId": "n:name",
            "tagValue": "ChwValveCtrl"
          },
          {
            "tagId": "n:displayName",
            "tagValue": "ChwValveCtrl"
          },
          {
            "tagId": "n:type",
            "tagValue": "kitControl:LoopPoint"
          },
          {
            "tagId": "n:ordInSession",
            "tagValue": "station:|h:55ee"
          },
          {
            "tagId": "n:station",
            "tagValue": "PS_CloudDataJACE"
          },
```

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-length: 16639
content-type: application/hal+json
date: Mon,08 May 2023 18:39:28 GMT
expires: 0
pragma: no-cache
strict-transport-security: max-age=31536000 ; includeSubDomains
vary: Origin,Access-Control-Request-Method,Access-Control-Request-Headers
x-content-type-options: nosniff
```

Sending First Requests - Telemetry



/api/v1/egress/openapi.json

Egress API 2.0 OAS3

/api/v1/egress/openapi.json

egress-controller

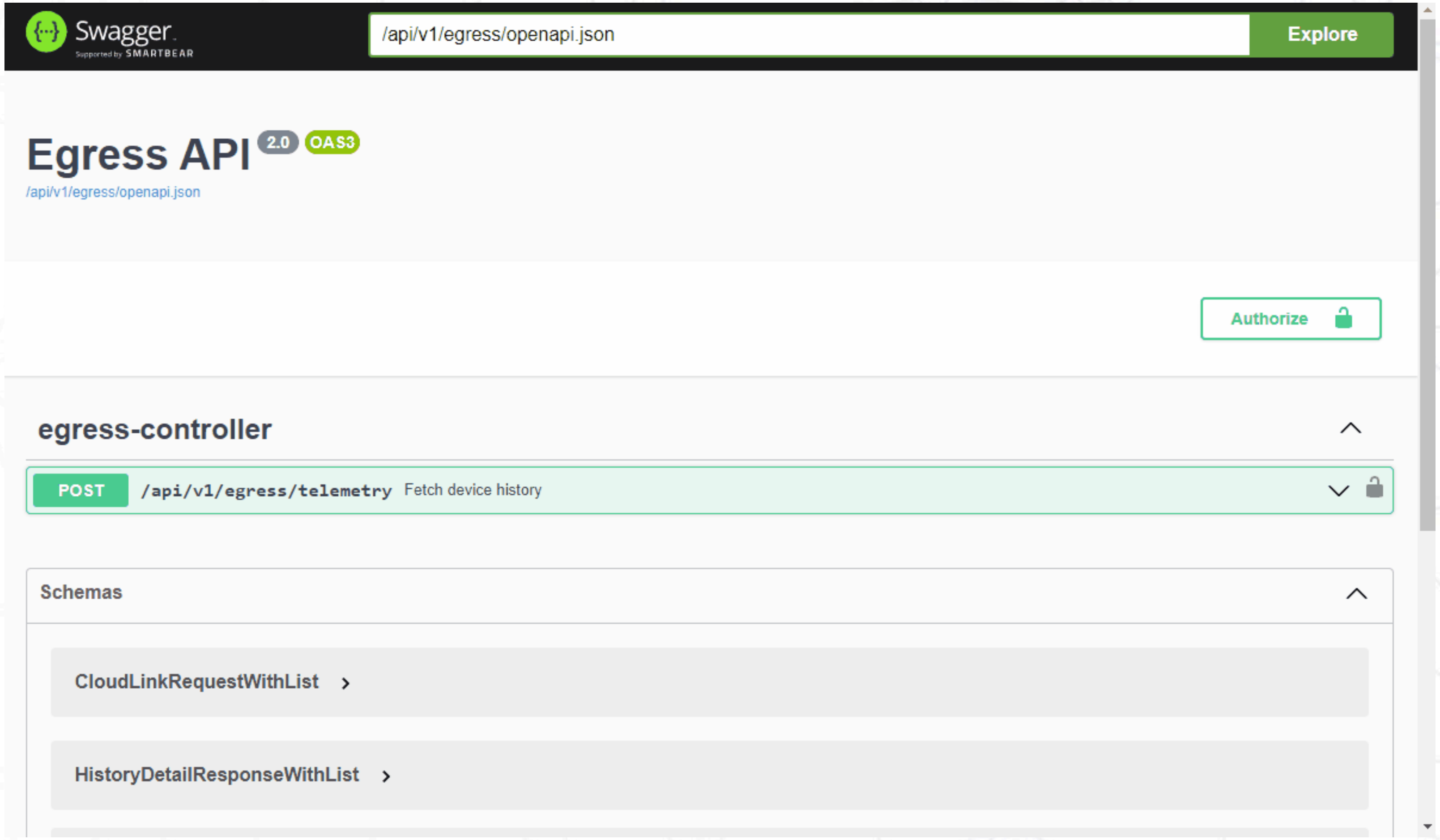
POST /api/v1/egress/telemetry Fetch device history

Schemas

CloudLinkRequestWithList

HistoryDetailResponseWithList

Sending First Requests - Telemetry



/api/v1/egress/openapi.json

Egress API 2.0 OAS3

/api/v1/egress/openapi.json

egress-controller

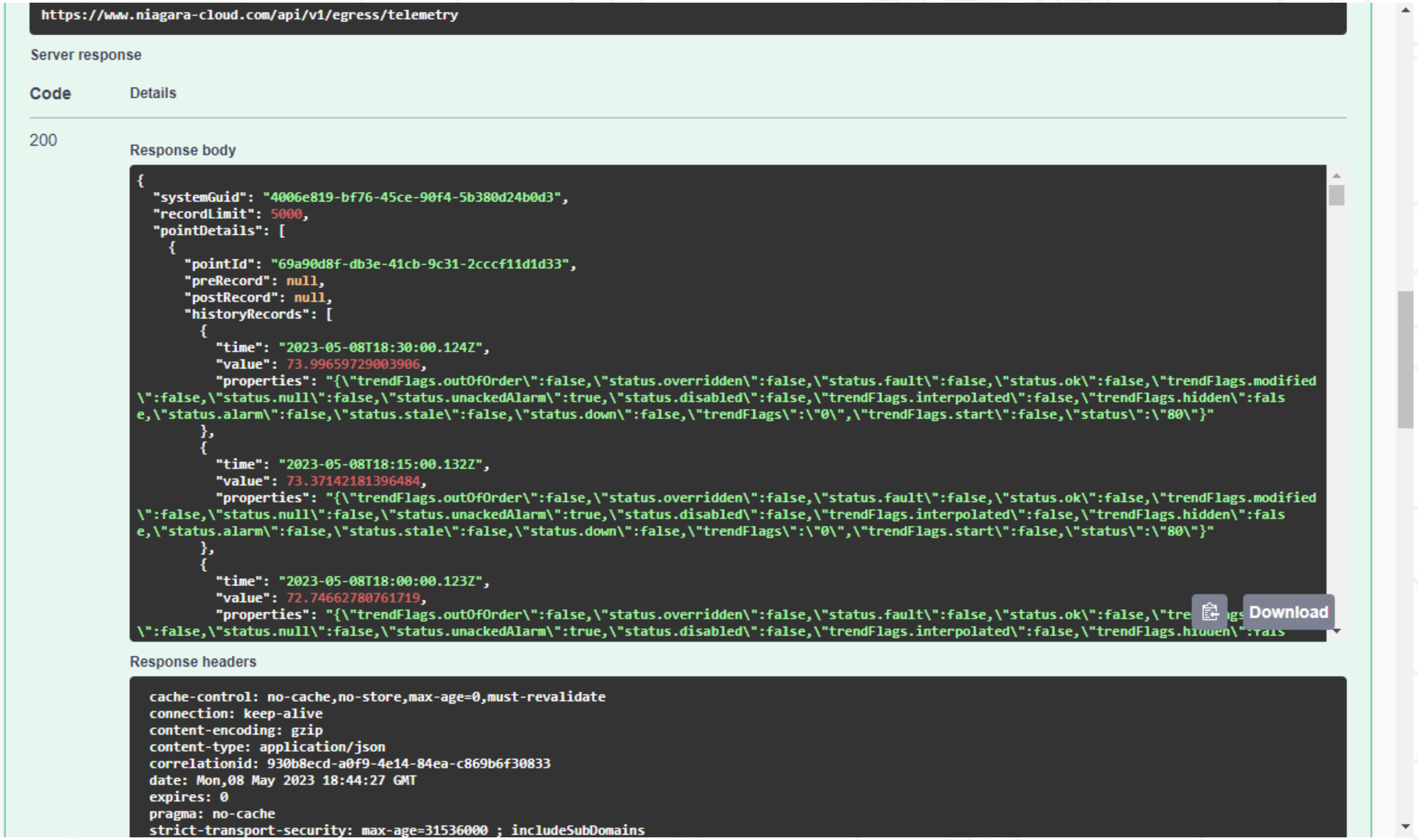
POST `/api/v1/egress/telemetry` Fetch device history

Schemas

CloudLinkRequestWithList

HistoryDetailResponseWithList

Sending First Requests - Telemetry



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

Server response

Code Details

200

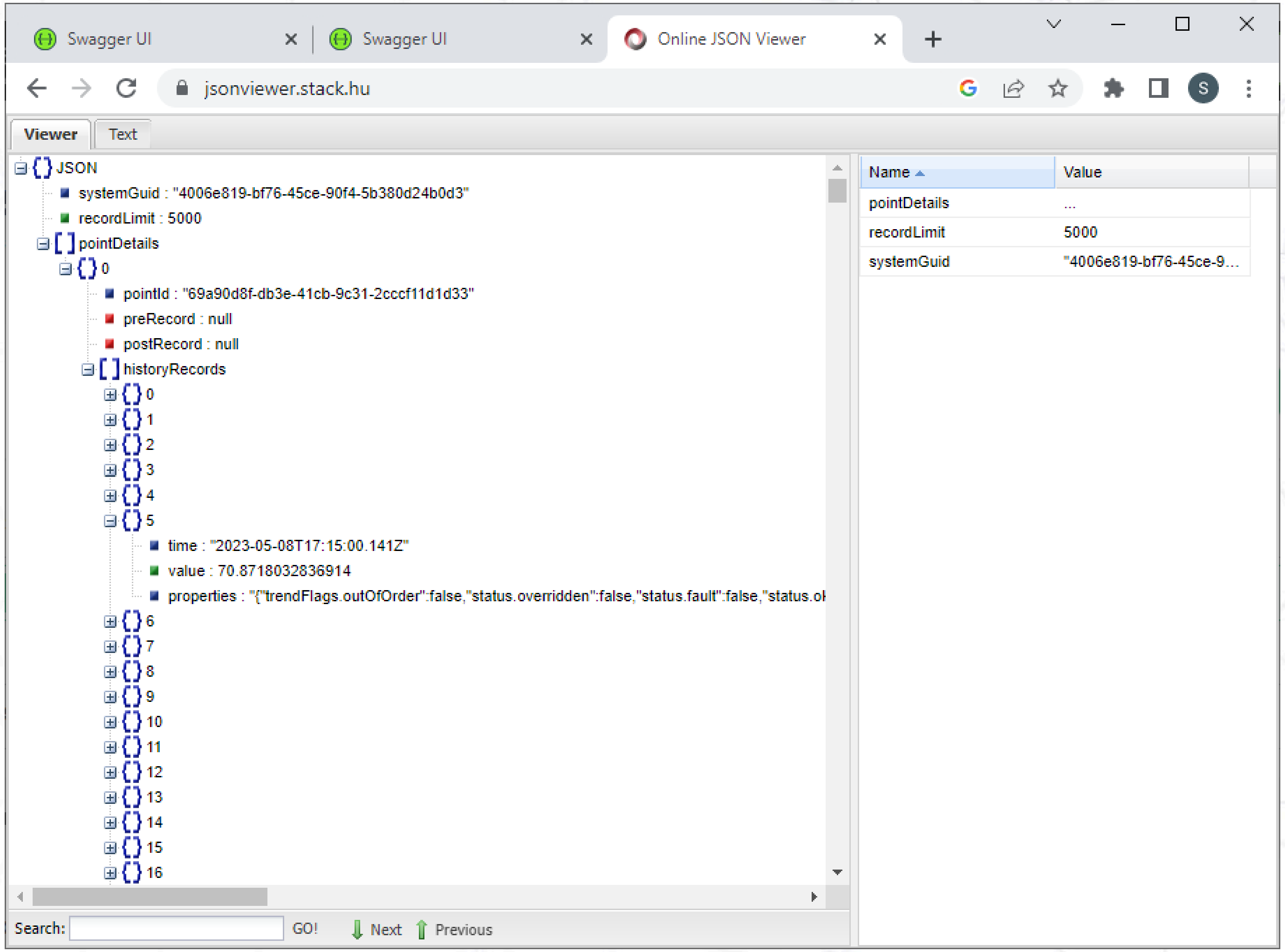
Response body

```
{
  "systemGuid": "4006e819-bf76-45ce-90f4-5b380d24b0d3",
  "recordLimit": 5000,
  "pointDetails": [
    {
      "pointId": "69a90d8f-db3e-41cb-9c31-2cccf11d1d33",
      "preRecord": null,
      "postRecord": null,
      "historyRecords": [
        {
          "time": "2023-05-08T18:30:00.124Z",
          "value": 73.99659729003906,
          "properties": "{\"trendFlags.outOfOrder\":false,\"status.overridden\":false,\"status.fault\":false,\"status.ok\":false,\"trendFlags.modified
\":false,\"status.null\":false,\"status.unackedAlarm\":true,\"status.disabled\":false,\"trendFlags.interpolated\":false,\"trendFlags.hidden\":fals
e,\"status.alarm\":false,\"status.stale\":false,\"status.down\":false,\"trendFlags\":\"0\",\"trendFlags.start\":false,\"status\":\"80\"}"
        },
        {
          "time": "2023-05-08T18:15:00.132Z",
          "value": 73.37142181396484,
          "properties": "{\"trendFlags.outOfOrder\":false,\"status.overridden\":false,\"status.fault\":false,\"status.ok\":false,\"trendFlags.modified
\":false,\"status.null\":false,\"status.unackedAlarm\":true,\"status.disabled\":false,\"trendFlags.interpolated\":false,\"trendFlags.hidden\":fals
e,\"status.alarm\":false,\"status.stale\":false,\"status.down\":false,\"trendFlags\":\"0\",\"trendFlags.start\":false,\"status\":\"80\"}"
        },
        {
          "time": "2023-05-08T18:00:00.123Z",
          "value": 72.74662780761719,
          "properties": "{\"trendFlags.outOfOrder\":false,\"status.overridden\":false,\"status.fault\":false,\"status.ok\":false,\"trendFlags.modified
\":false,\"status.null\":false,\"status.unackedAlarm\":true,\"status.disabled\":false,\"trendFlags.interpolated\":false,\"trendFlags.hidden\":fals
```

Response headers

```
cache-control: no-cache,no-store,max-age=0,must-revalidate
connection: keep-alive
content-encoding: gzip
content-type: application/json
correlationid: 930b8ecd-a0f9-4e14-84ea-c869b6f30833
date: Mon,08 May 2023 18:44:27 GMT
expires: 0
pragma: no-cache
strict-transport-security: max-age=31536000 ; includeSubDomains
```

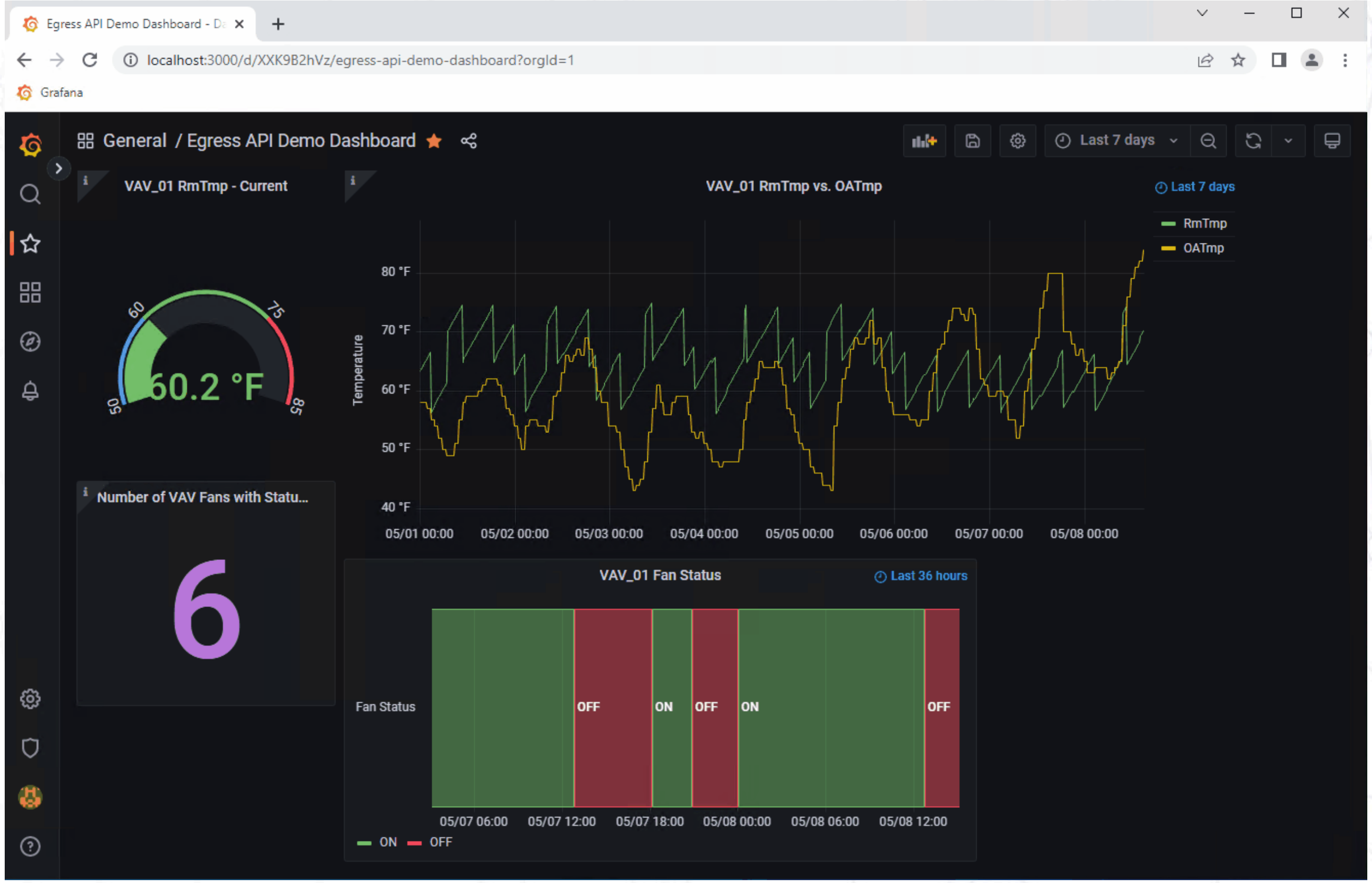
Sending First Requests - Telemetry



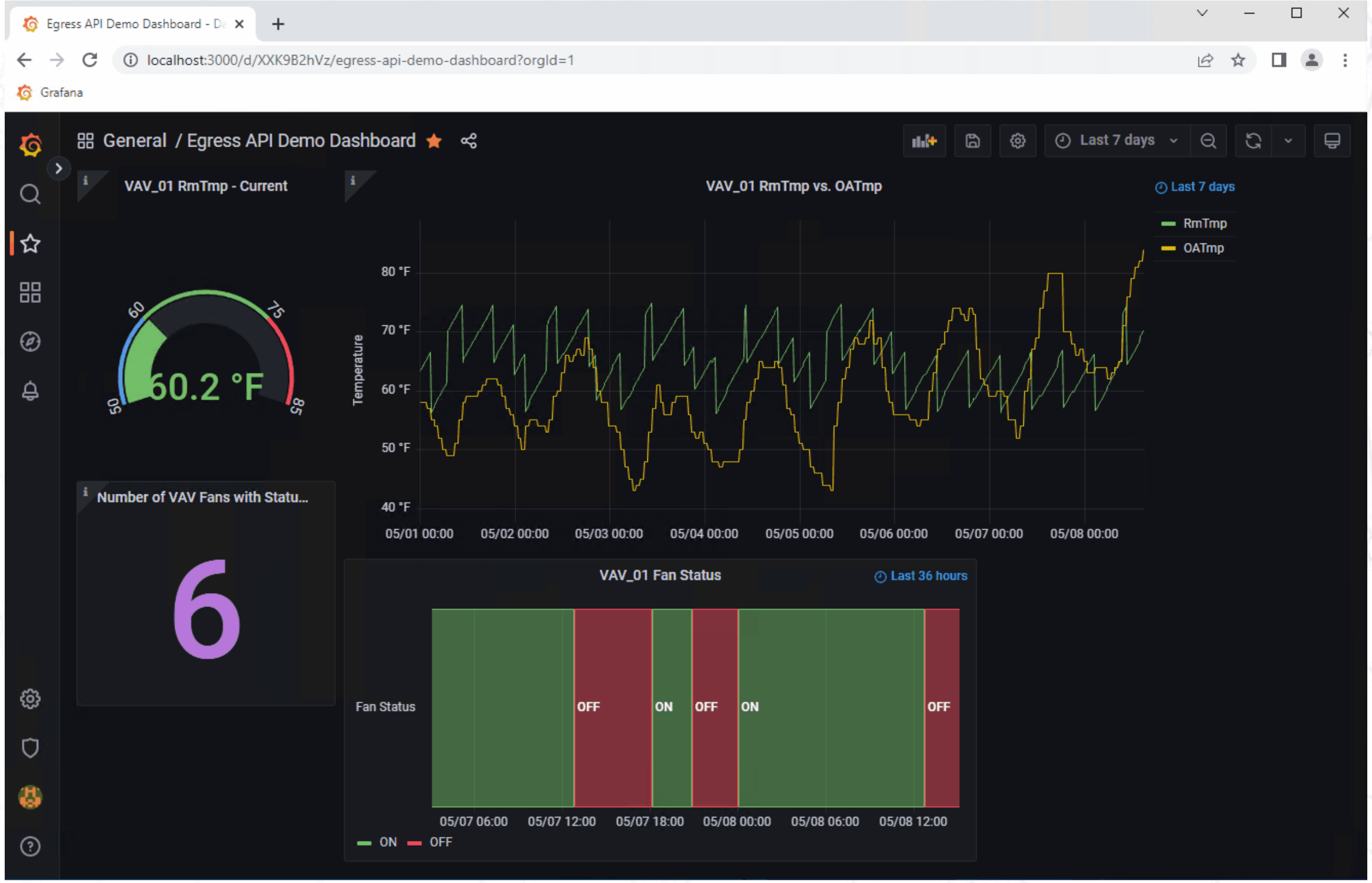
```
JSON
    systemGuid : "4006e819-bf76-45ce-90f4-5b380d24b0d3"
    recordLimit : 5000
    pointDetails
        0
            pointId : "69a90d8f-db3e-41cb-9c31-2cccf11d1d33"
            preRecord : null
            postRecord : null
            historyRecords
                0
                1
                2
                3
                4
                5
                    time : "2023-05-08T17:15:00.141Z"
                    value : 70.8718032836914
                    properties : "{"trendFlags.outOfOrder":false,"status.overridden":false,"status.fault":false,"status.ok
                6
                7
                8
                9
                10
                11
                12
                13
                14
                15
                16
```

Name ▲	Value
pointDetails	...
recordLimit	5000
systemGuid	"4006e819-bf76-45ce-9…

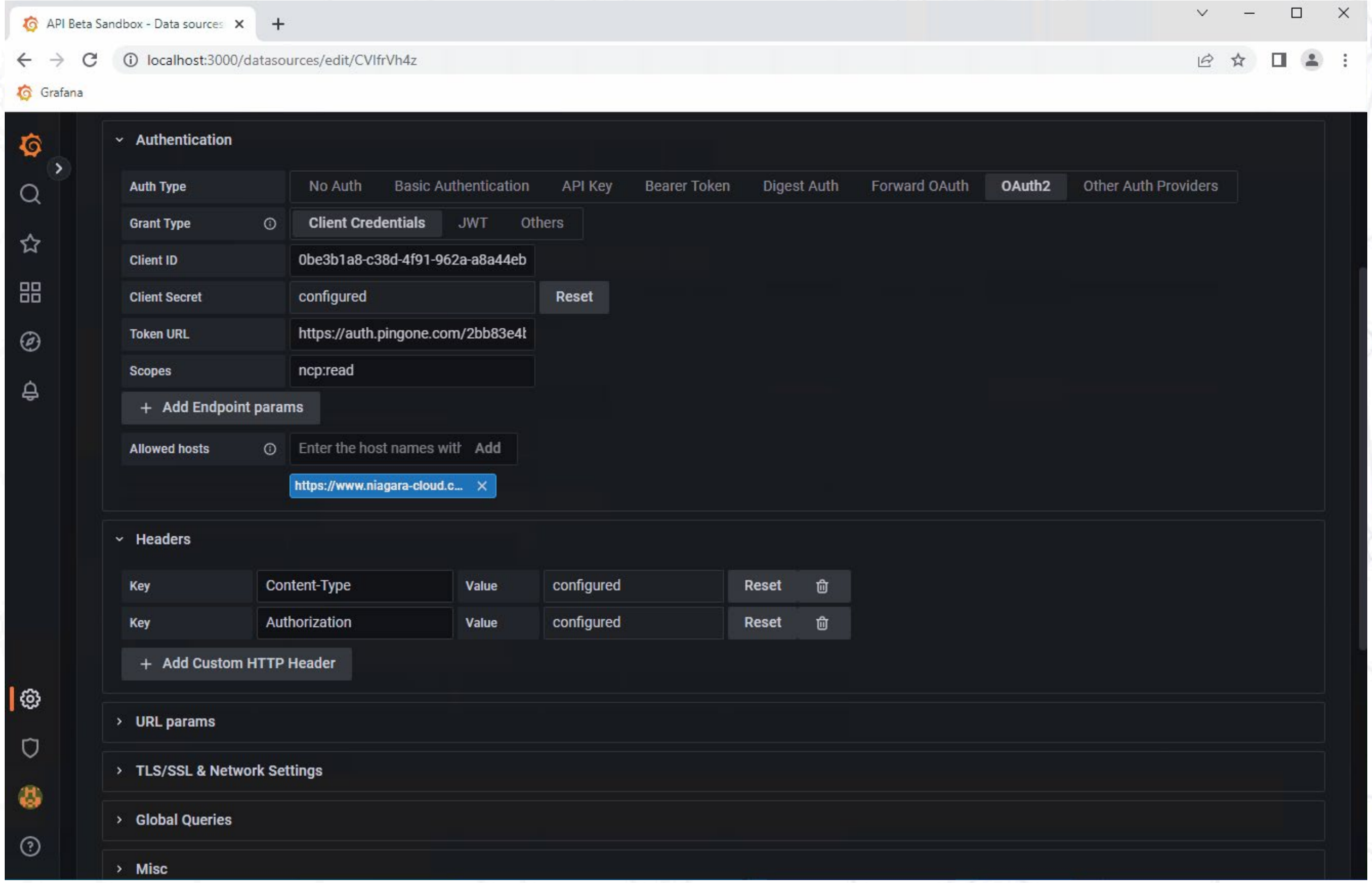
Third-Party Integration



Third-Party Integration



Third-Party Integration



API Beta Sandbox - Data sources

localhost:3000/datasources/edit/CVIfrVh4z

Grafana

Authentication

Auth Type	No Auth · Basic Authentication · API Key · Bearer Token · Digest Auth · Forward OAuth · **OAuth2** · Other Auth Providers
Grant Type	**Client Credentials** · JWT · Others
Client ID	0be3b1a8-c38d-4f91-962a-a8a44eb
Client Secret	configured · Reset
Token URL	https://auth.pingone.com/2bb83e4t
Scopes	ncp:read

+ Add Endpoint params

Allowed hosts	Enter the host names witt · Add
	https://www.niagara-cloud.c...

Headers

Key	Content-Type	Value	configured	Reset
Key	Authorization	Value	configured	Reset

+ Add Custom HTTP Header

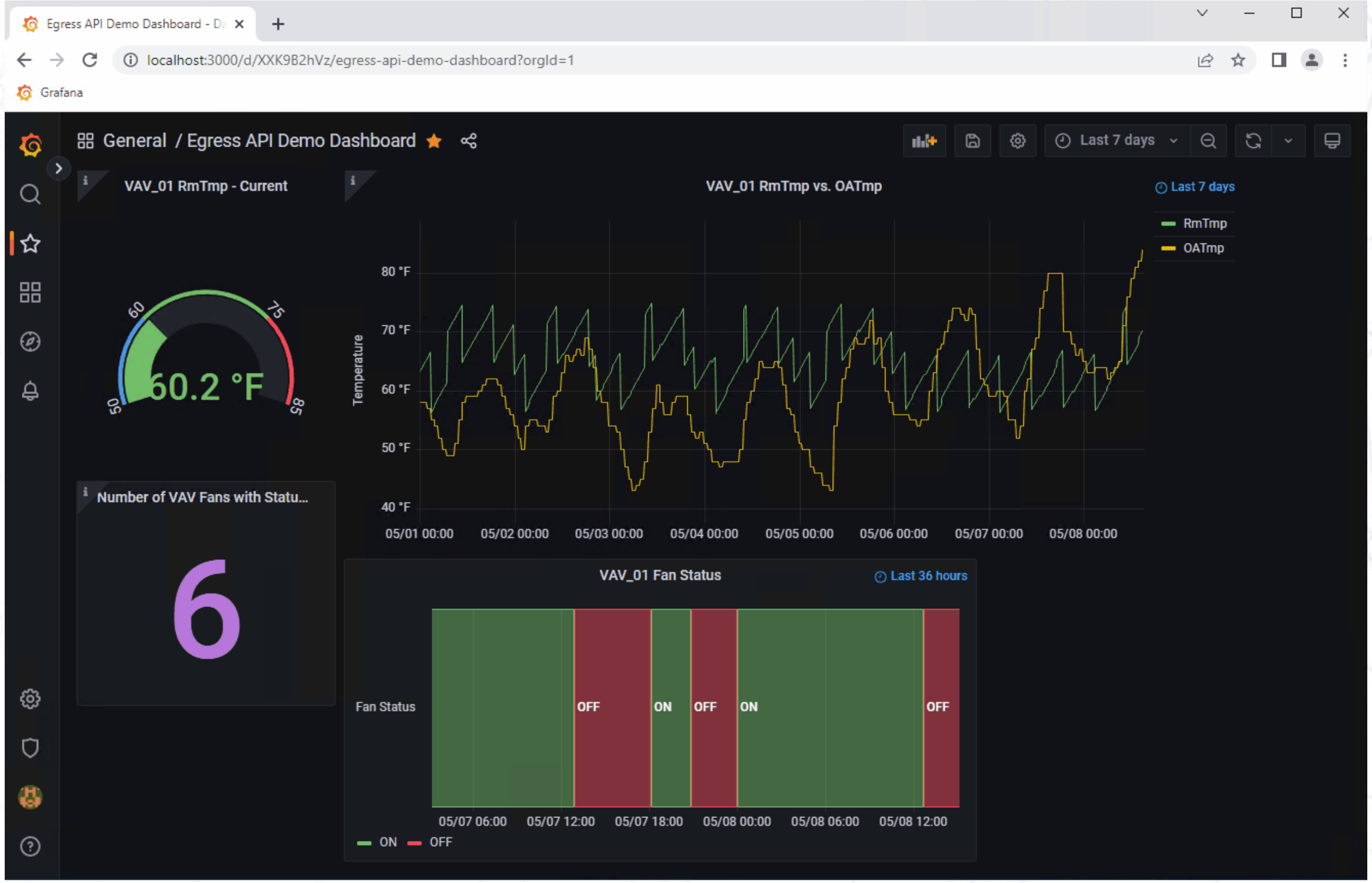
URL params

TLS/SSL & Network Settings

Global Queries

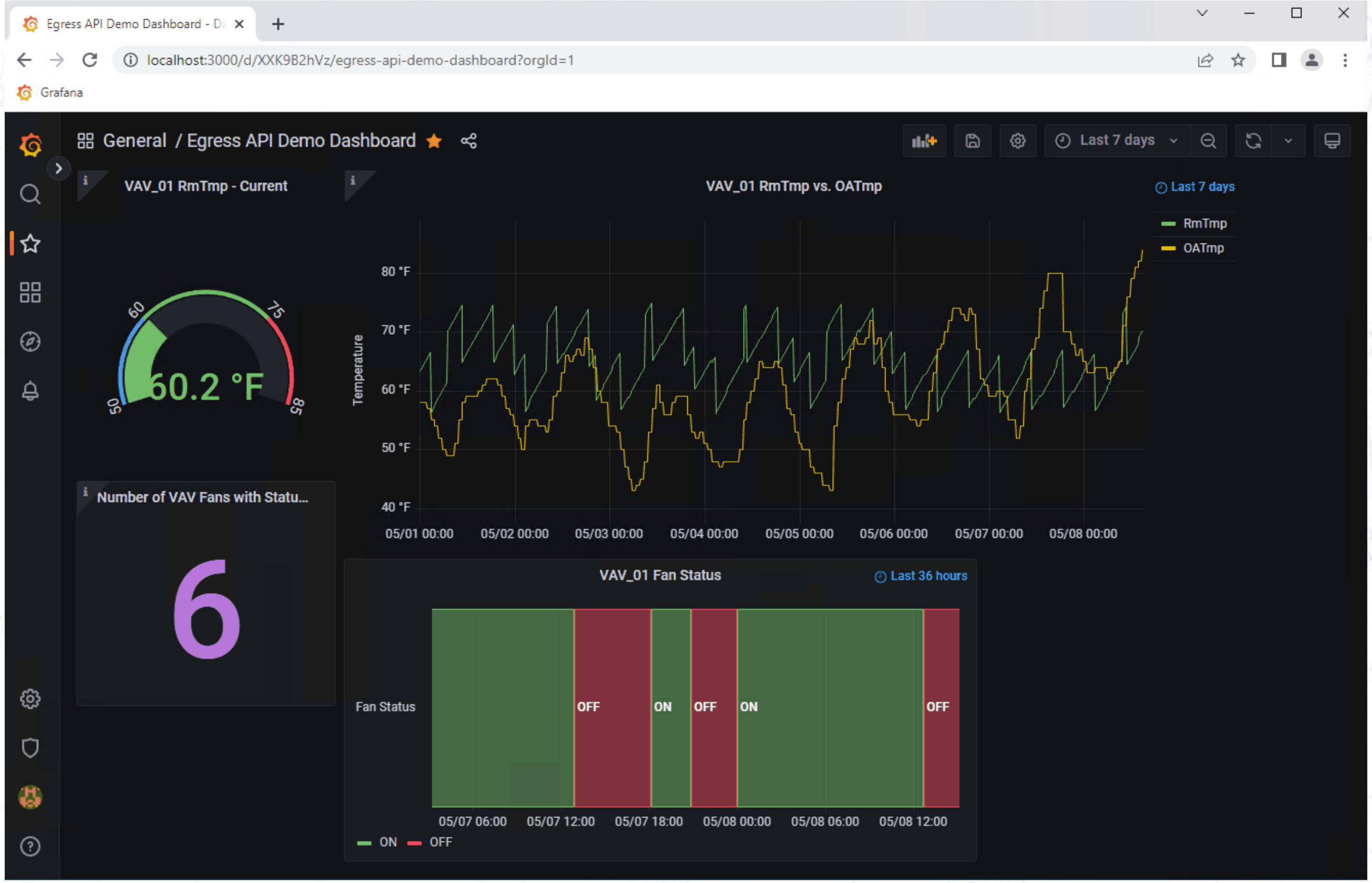
Misc

Third-Party Integration



TRIDIUM

Third-Party Integration



Egress API Demo Dashboard - D

localhost:3000/d/XXK9B2hVz/egress-api-demo-dashboard?orgId=1

Grafana

General / Egress API Demo Dashboard

Last 7 days

VAV_01 RmTmp - Current

60.2 °F

Number of VAV Fans with Statu...

6

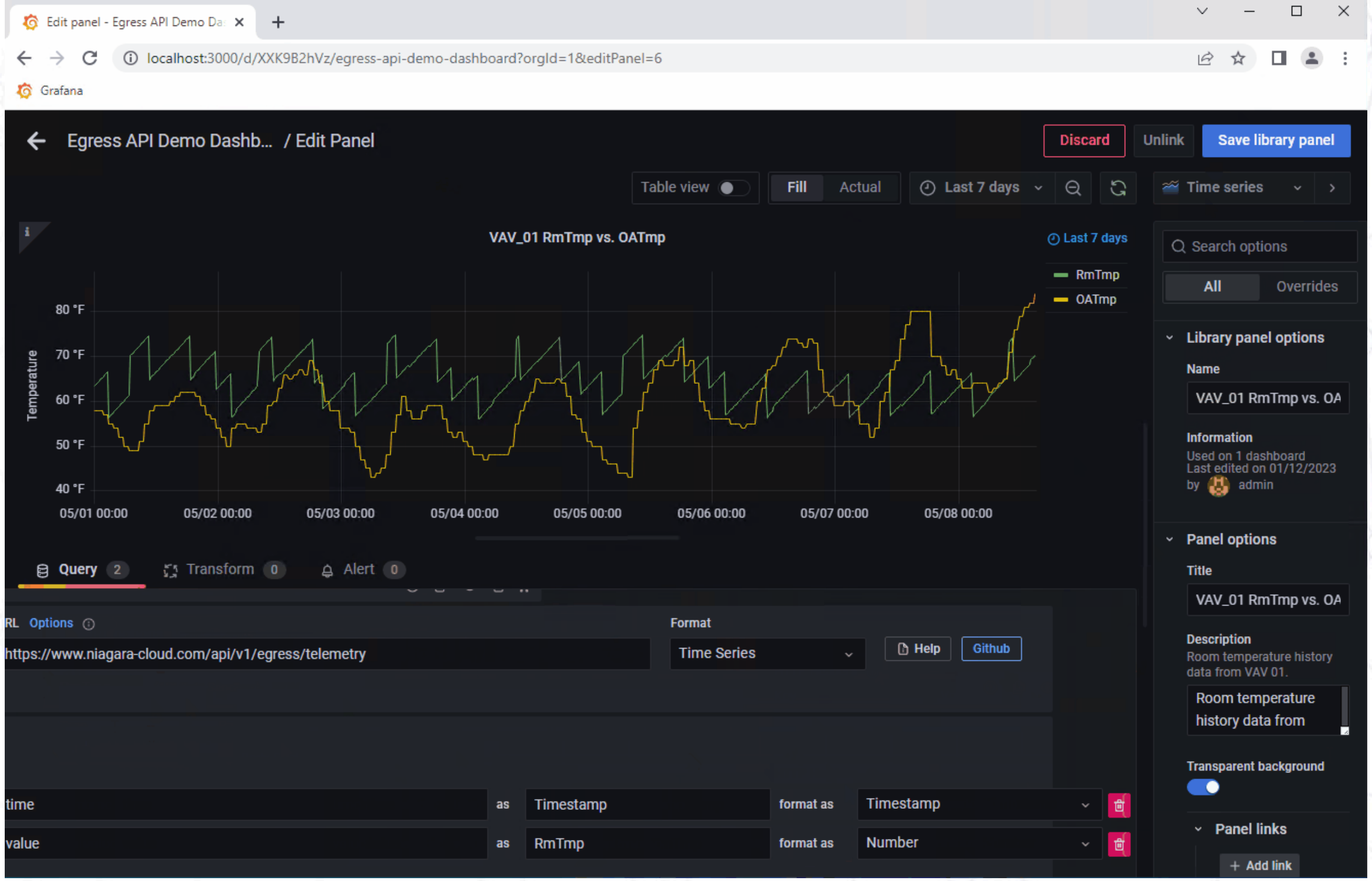
VAV_01 RmTmp vs. OATmp

Last 7 days

VAV_01 Fan Status

Last 36 hours

Third-Party Integration



localhost:3000/d/XXK9B2hVz/egress-api-demo-dashboard?orgId=1&editPanel=6

RL Options

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

Format: Time Series

time as Timestamp format as Timestamp

value as RmTmp format as Number

Library panel options

Name: VAV_01 RmTmp vs. OA

Information: Used on 1 dashboard
Last edited on 01/12/2023
by admin

Panel options

Title: VAV_01 RmTmp vs. OA

Description: Room temperature history data from VAV 01.

Room temperature history data from

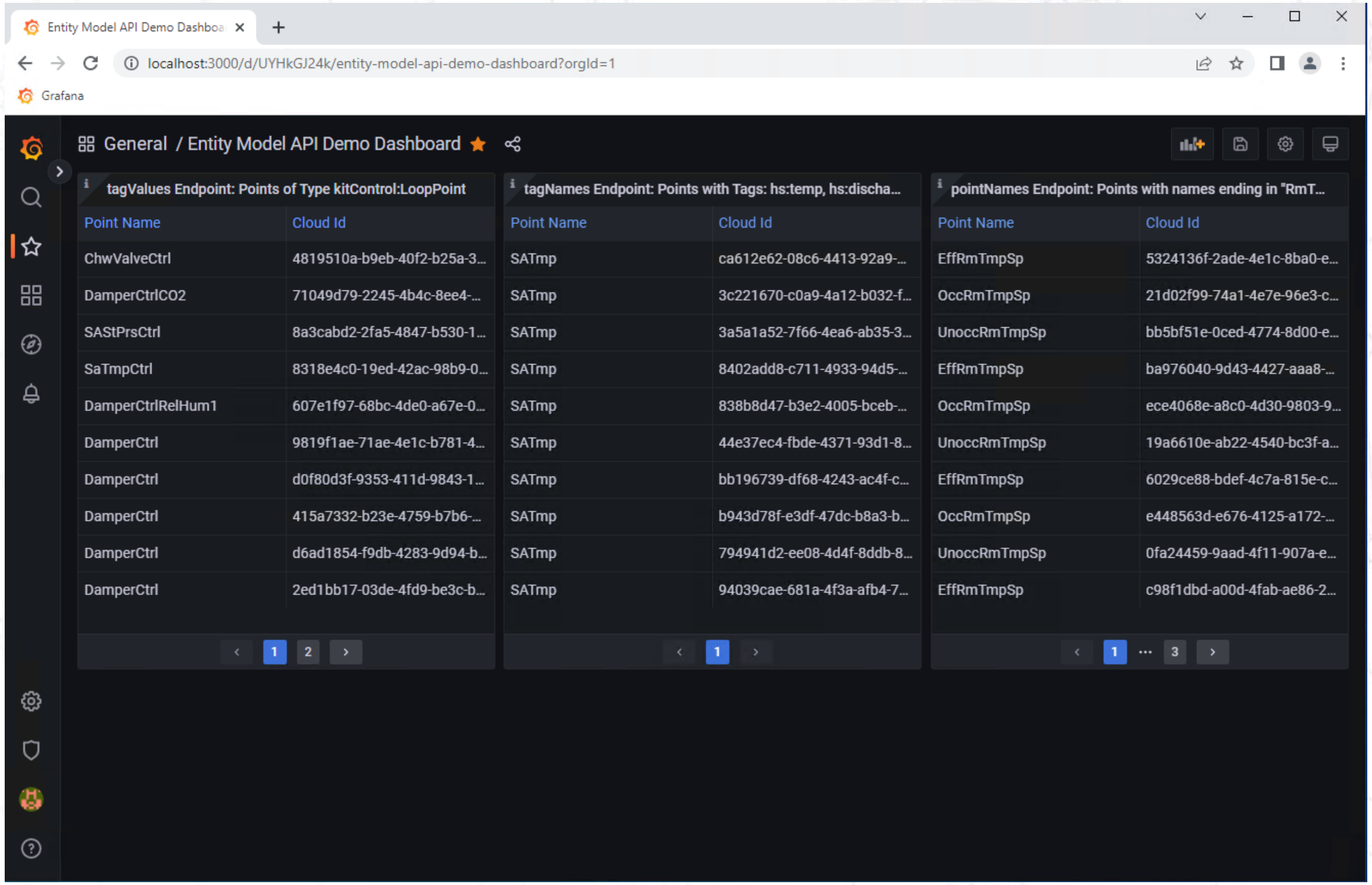
Transparent background ●

Panel links

NF 23

TRIDIUM

Third-Party Integration



Entity Model API Demo Dashboa
localhost:3000/d/UYHkGJ24k/entity-model-api-demo-dashboard?orgId=1
Grafana

General / Entity Model API Demo Dashboard

tagValues Endpoint: Points of Type kitControl:LoopPoint

Point Name	Cloud Id
ChwValveCtrl	4819510a-b9eb-40f2-b25a-3...
DamperCtrlCO2	71049d79-2245-4b4c-8ee4-...
SAStPrsCtrl	8a3cabd2-2fa5-4847-b530-1...
SaTmpCtrl	8318e4c0-19ed-42ac-98b9-0...
DamperCtrlRelHum1	607e1f97-68bc-4de0-a67e-0...
DamperCtrl	9819f1ae-71ae-4e1c-b781-4...
DamperCtrl	d0f80d3f-9353-411d-9843-1...
DamperCtrl	415a7332-b23e-4759-b7b6-...
DamperCtrl	d6ad1854-f9db-4283-9d94-b...
DamperCtrl	2ed1bb17-03de-4fd9-be3c-b...

‹ 1 2 ›

tagNames Endpoint: Points with Tags: hs:temp, hs:discha...

Point Name	Cloud Id
SATmp	ca612e62-08c6-4413-92a9-...
SATmp	3c221670-c0a9-4a12-b032-f...
SATmp	3a5a1a52-7f66-4ea6-ab35-3...
SATmp	8402add8-c711-4933-94d5-...
SATmp	838b8d47-b3e2-4005-bceb-...
SATmp	44e37ec4-fbde-4371-93d1-8...
SATmp	bb196739-df68-4243-ac4f-c...
SATmp	b943d78f-e3df-47dc-b8a3-b...
SATmp	794941d2-ee08-4d4f-8ddb-8...
SATmp	94039cae-681a-4f3a-afb4-7...

‹ 1 ›

pointNames Endpoint: Points with names ending in "RmT...

Point Name	Cloud Id
EffRmTmpSp	5324136f-2ade-4e1c-8ba0-e...
OccRmTmpSp	21d02f99-74a1-4e7e-96e3-c...
UnoccRmTmpSp	bb5bf51e-0ced-4774-8d00-e...
EffRmTmpSp	ba976040-9d43-4427-aaa8-...
OccRmTmpSp	ece4068e-a8c0-4d30-9803-9...
UnoccRmTmpSp	19a6610e-ab22-4540-bc3f-a...
EffRmTmpSp	6029ce88-bdef-4c7a-815e-c...
OccRmTmpSp	e448563d-e676-4125-a172-...
UnoccRmTmpSp	0fa24459-9aad-4f11-907a-e...
EffRmTmpSp	c98f1dbd-a00d-4fab-ae86-2...

‹ 1 ··· 3 ›

Recap

- Communication Paradigm
  1. Get CloudIDs from EntityModel
  2. Get Historical Records from Egress
- Getting Started
  - Get Bearer Access Token via Postman
  - Send First Requests via Swagger
- Integrate with Third-Party Apps

NF 23

CONNECTING THE WORLD

Q&A (if we have time)

NF 23

TRIDIUM

NF 23

CONNECTING THE WORLD