



CONNECTING
THE WORLD



NIAGARA FORUM APAC | 9-10 SEPTEMBER 2025



Y. H. Au

Sr. Solution Engineer

Tridium

NF
25

Agenda

Niagara Cloud Suite (NCS)
API Overview & Access

hello

What are the different types of data in our device?

1. Semantic Model Data
2. Telemetry Records
3. Alarm Records
4. Point Out Value

hello

Property Sheet

N

vav0_temp (Numeric Writable)

Facets

min=40.00 °C,max=130.00 °C,units=°C

Proxy Ext

null

Out

77.61 °C {unackedAlarm} @ 16

In1

- {null}

In2

- {null}

In3

- {null}

In4

- {null}

In5

- {null}

In6

- {null}

In7

- {null}

In8

- {null}

In9

- {null}

In10

- {null}

In11

- {null}

In12

- {null}

In13

- {null}

In14

- {null}

In15

- {null}

In16

77.61 °C {ok}

Fallback

100.00 °C {ok}

Override Expiration

null

interval

Numeric Interval History Ext

outOfRange

Alarm Source Ext

nc:cloudId

bc9ae561-b7d2-4426-bd3a-4able039265d

Refresh

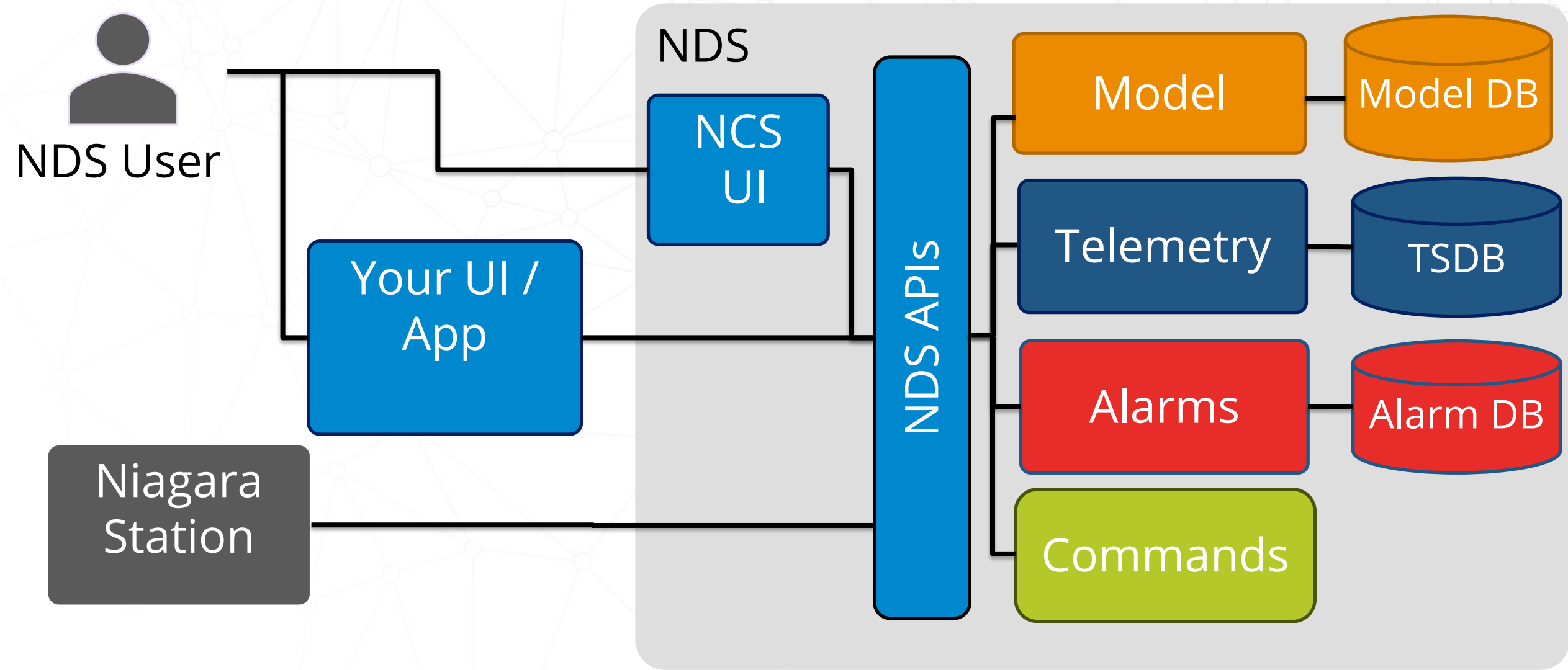
Save



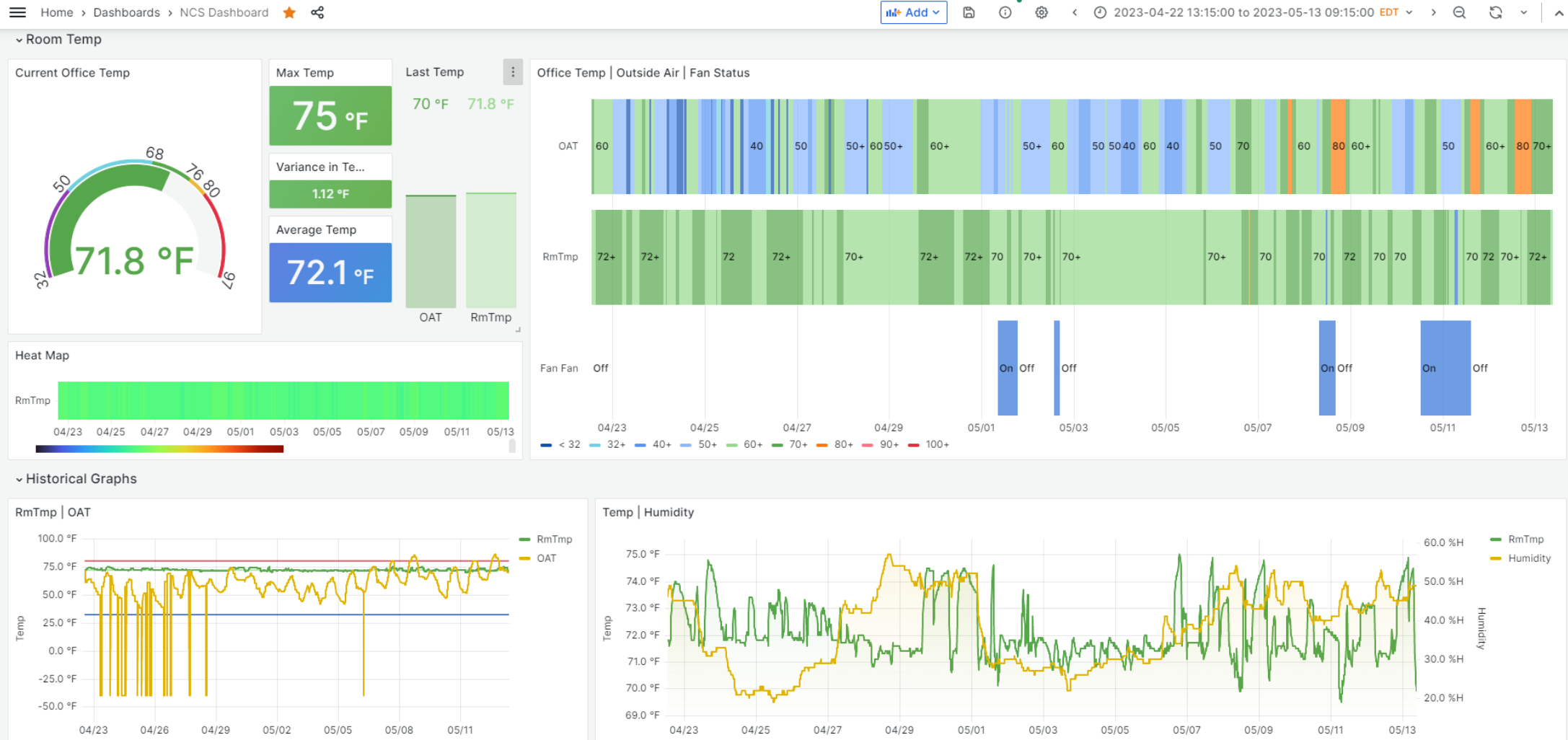
niagara cloud

SUITE™

Niagara Data Service



NDS Demo Dashboard



Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Model API
4. Query Telemetry API
5. Search Alarms
6. Read/Write points

Using NDS APIs

1. Create a Service Account

Create a Service Account

NCS Engineering

ALL CUSTOMERS

Search

LOS POLLOS HERMANOS

3 Projects | 48 Devices

GREENTECH SOLUTIONS

7 Projects | 46 Devices

ECOBUILDING DYNAMICS

0 Projects | 0 Devices

LEGHORN & SONS

0 Projects | 0 Devices

NEXTGEN AUTOMATION

0 Projects | 0 Devices

ENERGYFLOW NETWORKS

0 Projects | 0 Devices

BLUECLOUD ANALYTICS

58 Projects | 1315 Devices

PRECISION FACILITY SYSTEMS

1 Projects | 3 Devices

PROCONTROL AUTOMATION

1 Projects | 2 Devices

POWERSYNC DYNAMICS

1 Projects | 1 Devices

OMNISYS NETWORKS

0 Projects | 0 Devices

SMARTSENSE TECHNOLOGIES

1 Projects | 2 Devices



Create a Service Account



[Home](#) / [Precision Facility Systems](#)

Precision Facility Systems

Service Accounts

PROJECTS (1)

 *Search Project*

Zone 1

3

DEVICES

Search

Device	Host ID	Project	Location	Services
Zone 1 Floor 1	Win-5400-304A-D675-...	Precision Facility Systems	Bangalore	    
Zone 1 Floor 2	Win-1D90-F1E6-B253-...	Precision Facility Systems	Bengaluru	    
Zone 1 Floor 3	Win-A0C9-026F-D8FA-...	Precision Facility Systems	Bangalore	    

+ Create New Project

NF
25

Copyright © 2025 Tridium Inc. All Rights Reserved.



Create a Service Account

SERVICE ACCOUNTS

MY SERVICE ACCOUNTS

 Search

Create New Account

Service Account Name

Expiration Date

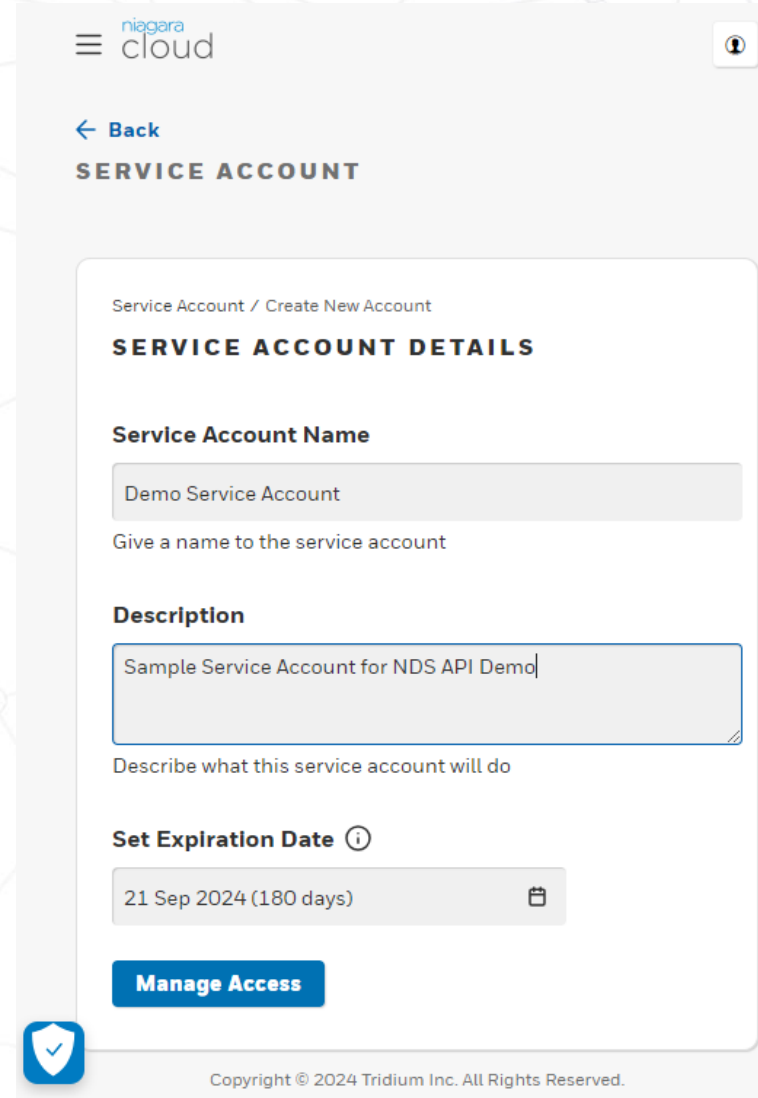
Action

No records found



Create a Service Account

1. Service Account Name
2. Description
3. Expiration Date



The screenshot shows the 'Create New Account' page for a Service Account in the Niagara Cloud interface. The page is titled 'SERVICE ACCOUNT' and includes a 'Back' link. The main section is 'SERVICE ACCOUNT DETAILS' and contains three input fields: 'Service Account Name' (with the value 'Demo Service Account'), 'Description' (with the value 'Sample Service Account for NDS API Demo'), and 'Set Expiration Date' (with the value '21 Sep 2024 (180 days)'). A 'Manage Access' button is located at the bottom of the form. The page also features a 'niagara cloud' logo in the top left and a copyright notice 'Copyright © 2024 Tridium Inc. All Rights Reserved.' in the bottom right.

niagara cloud

← Back

SERVICE ACCOUNT

Service Account / Create New Account

SERVICE ACCOUNT DETAILS

Service Account Name

Demo Service Account

Give a name to the service account

Description

Sample Service Account for NDS API Demo

Describe what this service account will do

Set Expiration Date ⓘ

21 Sep 2024 (180 days)

Manage Access

Copyright © 2024 Tridium Inc. All Rights Reserved.

Create a Service Account

Home / Precision Facility System / Service Accounts / Create Service Account

MANAGE ACCOUNT

PROJECTS AND DEVICES ACCESS ⓘ

Create Account

Role

NDS Operator

Delete Role

☐ All Projects

Zone 1 (1/3 Devices)

Search

☒ Zone 1 Floor 1

☐ Zone 1 Floor 2

☐ Zone 1 Floor 3

Role

User

Delete Role

Copyright © 2025 Tridium Inc. All Rights Reserved.



Service Account Created

SERVICE ACCOUNT CREATED

The Client ID can always be accessed from the service account page.

Your Client ID ⓘ

2210ea92-7efb-4557-8501-60a18ffe5f8e



Your Client Secret ⓘ

Nly8jkoQHo~fDjp2vtXlz7z_Bz9y07l-nEPxcxC89esP8~thJ62KgyQp...



The Client Secret is not being saved; please copy and paste it somewhere safe for future reference.

Close

Client ID

Like a username

Client Secret

Like a password

Store this SECURELY

Using NDS APIs

1. Create a Service Account
2. Obtain an access token

Getting an Access Token

OAuth request data

- grant_type: client_credentials
- scope: ncp:read

```
{  
  "grant_type": "client_credentials",  
  "scope": "ncp:read"  
}
```

Getting an Access Token

```
// Build the HTTP Request
HttpRequest tokenRequest = HttpRequest.newBuilder()

// Use NCS authentication endpoint from configuration file
.uri(URI.create(authConfig.getTokenUri()))

// Serialize request data into JSON string
.POST(toFormData(requestData))

// Include content type for the form data
.header("Content-Type", "application/x-www-form-urlencoded")

// Include the authorization header
.header("Authorization", getAuthHeader(authConfig.getClientId(),
authConfig.getClientSecret()))
.build();
```

Getting an Access Token

```
// Send HTTP request and handle response  
HttpResponse<String> response = httpClient.send(tokenRequest,  
    HttpResponse.BodyHandlers.ofString());  
NdsApiTokenHolder tokenHolder = mapper.readValue(response.body(),  
    NdsApiTokenHolder.class);  
return tokenHolder;
```

Getting an Access Token

```
{
  "access_token" :
  "eyJhbGciOiJSUzI1NiIsImtpZCI6ImRlZmF1bHQifQ.eyJjbGllbnRfaWQiOiIxZjgzN2I5My1mNmYwLTRhZDEtYTcyMy1lOTdlOWQxNzAwNTkiLCJpc3MiOiJodHRwczovL2F1dGgucGluZ29uZS5jb20vZWQ2M2M2NDUtOGNjYS00YjE3LWJmZmUtMTg1ZWQ5NDA1MjAwL2FzIiwianRpIjoiaMmI0YTlIyMWEtNzJjMS00NzEwLTk3ZTUtMjY2E1OTg4ODM2IiwiaWF0IjoxNzQxOTI1MjY4LCJleHAiOiJlZ3NDE",
  "token_type" : "Bearer",
  "expires_in" : 3600,
  "scope" : "ncp:read"
}
```

Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Model API

Model API

Entity Model Service API ^{v0} OAS 3.0

/api/v1/entitymodel/openapi.json

Authorize



point-controller



POST

/api/v1/entitymodel/customers/{customerId}/tagValues API to search points based on the given tag values



POST

/api/v1/entitymodel/customers/{customerId}/tagNames API to search points based on the given tag names



POST

/api/v1/entitymodel/customers/{customerId}/pointNames API to search points based on the given point names



Model API

Search by Point Name-

- Comparison type supported –
 - any
 - contains
 - exact
 - startswith
 - endswith

```
{  
  "systemGuid": "f3e6298b-dddb-4d6d-80a24add93d82ee5",  
  "searchItems": [  
    "AHU"  
  ],  
  "searchType": "pointName",  
  "comparisonType": "contains"  
}
```

Model API

Search by Tag Name-

- Multiple ontologies are supported

```
{
  "systemGuid": "f3e6298b-dddb-4d6d-80a2-4add93d82ee5",
  "searchItems": [
    "n:point",
    "nc:cloudId",
    "hs:air"
  ],
  "searchType": "tagName"
}
```

Model API

Search by Tag Value-

- Format
"tagId=tagValue"

```
{  
  "systemGuid": "e0123971-dcd4-4f63-8afc-12af1f2c7cf3",  
  "searchItems": [  
    "n:displayName=vav0_temp",  
    "n:type=control:NumericWritable"  
  ],  
  "searchType": "tagValue"  
}
```

Query Model API

- URI uses Customer ID
- Include access token in Authorization header

// Build EMS query request. The search criteria are expressed as a JSON object in the POST body.

```
String route = String.format(emsRoute, queryConfig.getCustomerId());
HttpRequest modelQuery = HttpRequest.newBuilder()
    .uri(URI.create(ndsApiHost + route))
    .POST(HttpRequest.BodyPublishers.ofString(queryBody))
    .header("Content-Type", "application/json")
    .header("Authorization", "Bearer " + accessToken)
    .build();
```

Query Model API

1. Send request
2. Deserialize return value
3. Return list of matching points

```
// Send HTTP request to EMS and handle response  
HttpResponse<String> response = httpClient.send(modelQuery,  
    HttpResponse.BodyHandlers.ofString());  
EmsQueryResponse body = mapper.readValue(response.body(), EmsQueryResponse.class);  
return Optional.ofNullable(body.getContent()).orElse(new  
    EmsQueryResults()).getPoints();
```

Model API Response

```
{
  "_embedded": {
    "pointModels": [
      {
        "systemGuid": "f3e6298b-dddb-4d6d-80a2-4add93d82ee5",
        "cloudId": "b5970101-bcb1-460a-ac6c-6f14f32cfc66",
        "name": "NumericWritable1",
        "tags": [
          { "tagId": "n:alarmablePoint", "tagValue": "Marker" },
          .....
        ],
        "relations": [
          { "inbound": true, "rId": "n:parent", "cId": "68377b69-7f06-4125-bbde-9b4daa0f651e" },
          { "inbound": false, "rId": "n:child", "cId": "68377b69-7f06-4125-bbde-9b4daa0f651e" }
        ],
        "facets": [
          { "key": "units", "type": "Unit", "value": "celsius(°C)" }
        ],
        "links": []
      }
    ]
  },
  "_links": {...},
  "page": {...}
}
```

Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Model API
4. Query Telemetry API

Telemetry API

Egress API ^{2.0} OAS 3.0

/api/v1/egress/openapi.json

Authorize 

egress-controller

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

Query Telemetry API

Query Parameters

- System GUID
- Telemetry ID (List)
- Start Time
- End Time
- Record Limit

```
{  
  "systemGuid": "a1b45678-0235-184b-9b45-1e671a7dfay",  
  "cloudId": [  
    "ab1c1234-6e89-44do-7ccf-68dbe693a9hew",  
    "e5babdey32-4232-4867-9a94-770cd8a1234"  
  ],  
  "startTime": "2022-01-03T06:28:14.353Z",  
  "endTime": "2022-03-23T08:59:49.427Z",  
  "recordLimit": "500",  
  "includePreRecord": true,  
  "includePostRecord": true,  
  "sortAscending": false  
}
```

Query Telemetry API

- URI is not customer specific
- Include access token in Authorization header

```
// Build request for Egress Service  
HttpRequest telemetryRequest = HttpRequest.newBuilder()  
    .uri(URI.create(ndsApiHost + dataEgressRoute))  
    .POST(HttpRequest.BodyPublishers.ofString(queryBody))  
    .header("Content-Type", "application/json")  
    .header("Authorization", "Bearer " + accessToken)  
    .build();
```

Query Telemetry API

1. Send request
2. Deserialize return value
3. Return list of history records

```
// Send HTTP request to Egress Service and handle response  
HttpResponse<String> response = httpClient.send(telemetryRequest,  
HttpResponse.BodyHandlers.ofString());  
return mapper.readValue(response.body(), TsdbQueryResponse.class);
```

Query Telemetry API

```
{
  "systemGuid": "ced128ce-a344-439d-8007-7bfc0ff08c63",
  "recordLimit": 5,
  "pointDetails": [
    {
      "pointId": "b6acb891-0d97-4ac5-a7b2-47e97f3fecf2",
      "preRecord": null,
      "postRecord": null,
      "historyRecords": [
        {
          "time": "2025-03-12T15:50:53.021Z",
          "value": 77.01333618164062,
          "properties": "{trendFlags.outOfOrder=false, status.overridden=false, status.fault=false, status.ok=false}"
        }...
      ]
    }
  ]
}
```

Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Model API
4. Query Telemetry API
5. Search Alarms

Alarm API

Alarms Service API 1.0 OAS 3.0

</api/v1/alarms/v3/api-docs>

Servers

<https://www.qa.niagara-cloud.com/api/v1/alarms> - Generated server url

Authorize



alarm-history-controller



POST </customers/{customerId}/alarms:search> Query alarms by customerId.



alarm-state-controller



GET </customers/{customerId}/devices/{deviceUuid}/alarms/state> Query alarm states by customerId, deviceUuid and, optionally, sourceId



GET </customers/{customerId}/devices/{deviceUuid}/alarms/count> Query alarm counts by customerId, deviceUuid and, optionally, sourceId



Alarm API

Alarm Search Request-

- Logical – "or", "and", "not"
- Relational - "eq", "neq", "lt", "gt", "le", "ge"

```
{
  "query": {
    "op": "and",
    "expr": [
      {
        "field": "deviceUuid",
        "op": "eq",
        "value": "f36f1268-0d4d-43d2-97fa-6c82b292ee72"
      },
      {
        "field": "generatedTime",
        "op": "gt",
        "value": "2024-07-24T00:00:00Z"
      },
      {
        "field": "alarmData.toState",
        "op": "in",
        "values": [
          "Offnormal", "Fault"
        ]
      }
    ]
  },
  "page": {...},
  "sort": {...}
}
```

Alarm API

- URI is customer specific
- Include access token in Authorization header

// Build request for Alarm Service

```
String route = String.format(alarmRoute, customerId);
HttpRequest commandRequest = HttpRequest.newBuilder()
    .uri(URI.create(ndsApiHost + route + "alarms:search"))
    .POST(HttpRequest.BodyPublishers.ofString(createAlarmRequest(deviceUuid)))
    .header("Content-Type", "application/json")
    .header("Authorization", "Bearer " + accessToken)
    .build();
```

// Send HTTP request to Alarm Service and handle response

```
HttpResponse<String> response = httpClient.send(commandRequest,
    HttpResponse.BodyHandlers.ofString());
return mapper.readValue(response.body(), AlarmQueryResponse.class);
```

Alarm API

Alarm Search Response

```
{
  "alarms": [
    {
      "deviceUuid": "2faa3a72-bbf2-4505-b837-934cb40bc491",
      "alarmUuid": "758c1eb2-b5e1-44c3-a2de-3632cf1870ce",
      "iotMessageId": "fceb756d-a41a-449d-9df1-e6caaf58bd7f",
      "cloudId": "bd44b77c-1dfd-497f-83f8-01c4aab0aa92",
      "generatedTime": "2025-03-14T19:23:34.72Z",
      "sourceState": "Normal",
      "ackState": "Unacked",
      "ackRequired": true,
      "source": "vav0_temp",
      "alarmData": {
        "AlarmSourceId": "68414eb5-f691-4328-9839-ee3f144bf16c",
        "fromState": "lowLimit",
        "lowLimit": "60.00 °F",
        "SystemGuid": "ced128ce-a344-439d-8007-7bfc0ff08c63",
        "ordList": "local:|station:|slot:/Points/building4/floor0/zone0/vav0_temp/0",
        "status": "{ok} @ 9"
      }
    }
  ]
}
```

Alarm API

Alarm Count Response

```
[
  {
    "deviceUuid": "2faa3a72-bbf2-4505-b837-934cb40bc491",
    "cloudId": "2faa3a72-bbf2-4505-b837-934cb40bc491",
    "iotMessageId": "33e970b0-35c2-440e-8a75-a6e4cf4fdc53",
    "acked": 0,
    "unacked": 499,
    "ackPending": 0,
    "total": 499,
    "lastUpdateTime": "2025-03-14T20:33:05.859Z"
  },
  {
    "deviceUuid": "2faa3a72-bbf2-4505-b837-934cb40bc491",
    "cloudId": "bd44b77c-1dfd-497f-83f8-01c4aab0aa92",
    "iotMessageId": "33e970b0-35c2-440e-8a75-a6e4cf4fdc53",
    "acked": 0,
    "unacked": 499,
    "ackPending": 0,
    "total": 499,
    "lastUpdateTime": "2025-03-14T20:33:05.859Z"
  }
]
```

Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Find points
4. Query Telemetry API
5. Search Alarms
6. Read/Write points

Command API

NDS Control Api 1.0 OAS 3.0

/api/v1/control/openapi.json

Servers

/api/v1/control

Authorize



command-controller



POST

/devices/{systemGuid}/commands/read API to read current value of one or more points



POST

/devices/{systemGuid}/commands/write API to write current value of one or more points



POST

/devices/{systemGuid}/commands/clear API to clear current value of one or more points



POST

/devices/{systemGuid}/commands/readAll API to read all current input values for one or more points



POST

/devices/{systemGuid}/commands/acknowledgeAlarm API to acknowledge one or more alarms



POST

/devices/{systemGuid}/commands/editAlarmData API to add alarm notes



POST

/devices/commands/acknowledgeAlarmBatch API to acknowledge one or more alarms for multiple devices



Executing Commands

- Read request

```
{  
  "cloudIds": [  
    "b5970101-bcb1-460a-ac6c-6f14f32cfc66"  
  ],  
  "requestProcessingPriority": 255  
}
```

- Write request

```
{  
  "points": [  
    {  
      "cloudId": "b5970101-bcb1-460a-ac6c-6f14f32cfc66",  
      "value": 12.54,  
      "duration": 50,  
      "inputPriority": 10  
    }  
  ],  
  "requestProcessingPriority": 255  
}
```

Executing Commands

- Acknowledge Alarm

```
{  "alarmUuids": [    "8b838d8f-c371-432d-96df-76c507bcfce2"  ],  "requestProcessingPriority": 255}
```

- Edit Alarm Data

```
{  "alarms": [    {      "alarmUuid": "8b838d8f-c371-432d-96df-76c507bcfce2",      "notes": "test alarm data"    }  ],  "requestProcessingPriority": 255}
```

Executing Commands

- URI contains System GUID
- Include access token in Authorization header

```
// Build request for Command Service  
String route = String.format(commandRoute, systemGuid);  
HttpRequest commandRequest = HttpRequest.newBuilder()  
    .uri(URI.create(ndsApiHost + route + "command_type"))  
  
    .POST(HttpRequest.BodyPublishers.ofString(requestBody))  
    .header("Content-Type", "application/json")  
    .header("Authorization", "Bearer " + accessToken)  
    .build();
```

Command Types:

- read
- write
- clear
- readAll
- acknowledgeAlarm

Executing Commands

1. Send command
2. Deserialize return value
3. Return command result

```
// Send HTTP request to Command Service and handle response  
HttpResponse<String> response = httpClient.send(commandRequest,  
HttpResponse.BodyHandlers.ofString());  
ReadCommandResponse commandResponse = mapper.readValue(response.body(),  
ReadCommandResponse.class);  
return commandResponse;
```

Executing Commands

Read command results

```
{
  "pointReadDetails": [
    {
      "cloudId": "36af9b6f-9c7a-4cf0-9299-d94c43f3280e",
      "value": "79.18",
      "status": "{ok} @ 16"
    }
  ]
}
```

Property Sheet	
vav0_temp (Numeric Writable)	
Facets	min=60.00 °F,max=130.00 °F,units=°F
Proxy Ext	null
Out	79.18 °F {ok} @ 16
In1	- {null}
In2	- {null}
In3	- {null}
In4	- {null}
In5	- {null}
In6	- {null}
In7	- {null}
In8	- {null}
In9	- {null}
In10	- {null}
In11	- {null}
In12	- {null}
In13	- {null}
In14	- {null}
In15	- {null}
In16	79.18 °F {ok}
Fallback	100.00 °F {ok}
Override Expiration	null
interval	Numeric Interval History Ext
nc:cloudId	36af9b6f-9c7a-4cf0-9299-d94c43f3280e
nc:telemetryId	c67b47c0-aa80-49ef-b080-5e41f24f977a

Executing Commands

Write command results

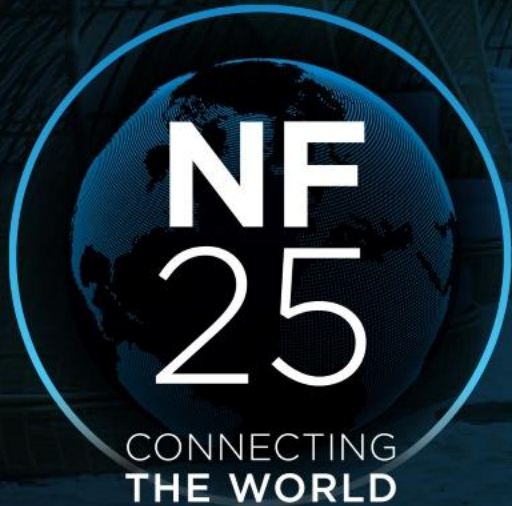
```
{
  "pointWriteDetails": [
    {
      "cloudId": "bd44b77c-1dfd-497f-83f8-01c4aab0aa92",
      "status": "success",
      "writeTime": "2025-03-15T00:19:40.159+05:30"
    }
  ]
}
```

Property Sheet		
Out	70.00 °F {unackedAlarm} @ 9	
In1	- {null}	
In2	- {null}	▼
In3	- {null}	▼
In4	- {null}	▼
In5	- {null}	▼
In6	- {null}	▼
In7	- {null}	▼
In8	- {null}	
In9	70.00 °F {ok}	▼
In10	73.99 °F {ok}	▼
In11	- {null}	▼
In12	- {null}	▼
In13	- {null}	▼
In14	- {null}	▼
In15	- {null}	▼
In16	79.01 °F {ok}	▼
Fallback	100.00 °F {ok}	▼
Override Expiration	null	
interval	Numeric Interval History Ext	
nc:cloudId	bd44b77c-1dfd-497f-83f8-01c4aab0aa92	
OutOfRangeAlarmExt	Alarm Source Ext	
nc:telemetryId	b6acb891-0d97-4ac5-a7b2-47e97f3fecf2	
cloudWriteIn9	Cloud Write Info	
Source	NDS_CONTROL_APPLICATION	
Expiration	15-Mar-2025 12:17 AM IST	
Value	70.00	

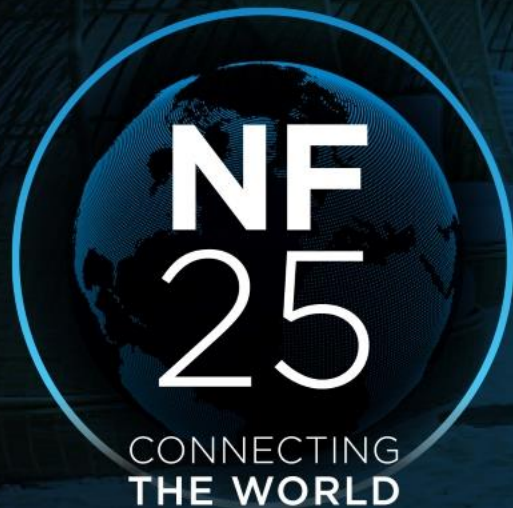
Documentation and Code

- NCS Documentation Home
 - <https://docs.niagara-community.com/category/NCS>
- NDS API Guide
 - <https://docs.niagara-community.com/bundle/NdsApi/page/index.html>
- NCS Partner Guide
 - <https://docs.niagara-community.com/bundle/NcsPartner/page/index.html>
- Example Reference code will be posted to Tridium GitHub Repository at <https://github.com/tridium>

Questions



THANK YOU





NF
25

CONNECTING
THE WORLD