

NS2024

POWER OF PARTNERSHIP

Using Niagara Cloud Suite APIs

*Norm Beekwilder &
Vidya Shivamurthy
Tridium*

NS2024
POWER OF PARTNERSHIP

Today's Topics

- Niagara Cloud Suite (NCS)
- Niagara Data Service (NDS)
- API Usage
- Demo

Niagara Cloud Suite

NDS
Model

NDS
History

NDS
Command

NDS
Alarm

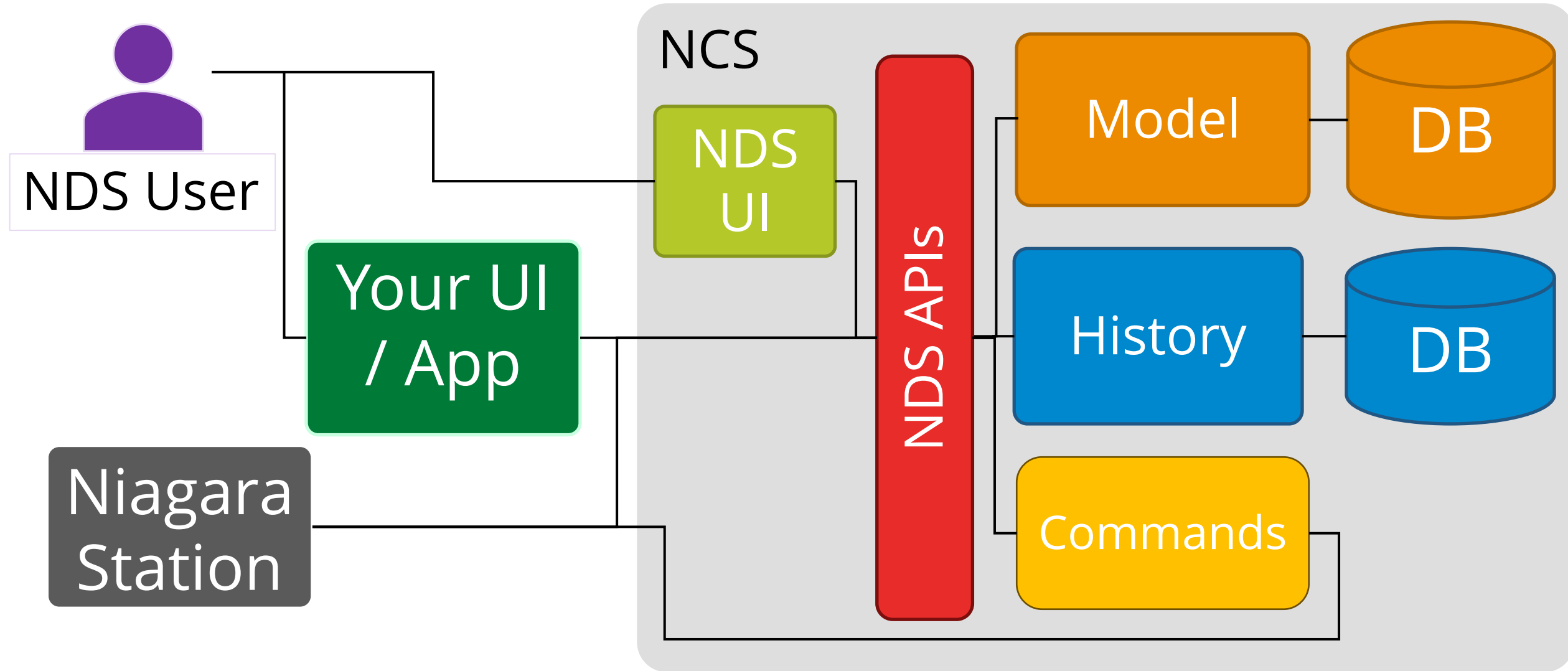
Niagara
Recover

Niagara
Remote

Niagara Data Service

NCS Management Portal

Niagara Data Service



Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Use Model API to find semantic metadata
4. Use History API to get history data
5. Use Command API to read/write values to station

Using NDS APIs

Spring Boot App
Uses HttpClient
Demonstrates

- Get access token
- Query Model API
- Query History API
- Read and Write Command APIs

```
@SpringBootApplication
public class NdsApiDemo {
    no usages
    @Bean
    public CommandLineRunner run(/unused/ RestTemplate restTemplate) throws Exception {
        return args -> {
            // Get an access token
            NdsApiTokenHolder holder = getAccessToken();
            // Query EMS for points
            List<Point> points = modelQuery(holder.getAccessToken());
            for (Point point : points) {
                // Query Data Egress service for telemetry data
                Optional<String> telemetryId = point.getTags().stream().filter(
                    t -> "nc:telemetryId".equals(t.getTagId())).map(t -> t.getTagValue()).findFirst();
                if (telemetryId.isPresent()) {
                    TsdbQueryResponse telemetryData = readTelemetryData(
                        holder.getAccessToken(), queryConfig.getSystemGuid(), telemetryId.get());
                }
                // Execute commands
                ReadCommandResponse readResponse = readPointCommand(
                    holder.getAccessToken(), queryConfig.getSystemGuid(), point.getCloudId());
                double value = Double.parseDouble(readResponse.getPointReadDetails().get(0).getValue()) + 5;
                WriteCommandResponse writeResponse = writePointCommand(holder.getAccessToken(),
                    queryConfig.getSystemGuid(), point.getCloudId(), value, duration, priority);
            }
        };
    }
}
```

Using NDS APIs

1. Create a Service Account

Create a Service Account

niagara
cloud



Niagara Smart Buildings

ALL CUSTOMERS

NSB PROFESSIONAL SERVICES
1 Projects | 1 Devices

NSB SYSTEMS ENGINEERING
10 Projects | 9 Devices

TRIDIUM EUROPE LTD
0 Projects | 0 Devices



Copyright © 2023 Tridium Inc. All Rights Reserved.

Create a Service Account

niagara
cloud

← Back

CUSTOMER

NSB Systems Engineering

Service Accounts

PROJECTS (10)

Search Project

All Projects

Navarre, FL

London, UK

Mechanicsville, VA

Richmond, VA

ACME Tomato, NC

Tofte, MN

Dubai, AE

Midlothian, VA

+ Create New Project

RICHMOND, VA

Search

Sort by

2 Results

DEVICE/HOST ID	PROJECT/LOCATION	SUBSCRIBED SERVICES	
J8blue Qnx-TITAN-6A40-B8CC-8BDF-EFE5	Richmond, VA Richmond Office	  	⋮
VA51J8_2 Qnx-TITAN-5290-78F7-0BE5-618E	Richmond, VA VA51 Office	  	⋮

NS2024

TRIDIUM

Create a Service Account



[← Back](#)

SERVICE ACCOUNTS

MY SERVICE ACCOUNTS

[Create New Account](#)

1-10 of 12

SERVICE ACCOUNT NAME	EXPIRATION DATE	ACTION
AcmeTomatoeService Client Id: bc5b3c95-6098-42aa-8a66-bff52334a7aa	17 Apr, 2024 Expires in 7 days	Details
Mamajek Client Id: 3ac26c55-87fd-4291-ad5f-a387d3693604	20 Apr, 2024 Expires in 10 days	Details
Tandem Client Id: 3f314d14-c51b-4c4b-acf6-62c4effa5b99	5 May, 2024	Details
CMcK Grafana Client Id: 375c067h-2f39-46df-ac26-4e22109215fd	15 May, 2024	Details

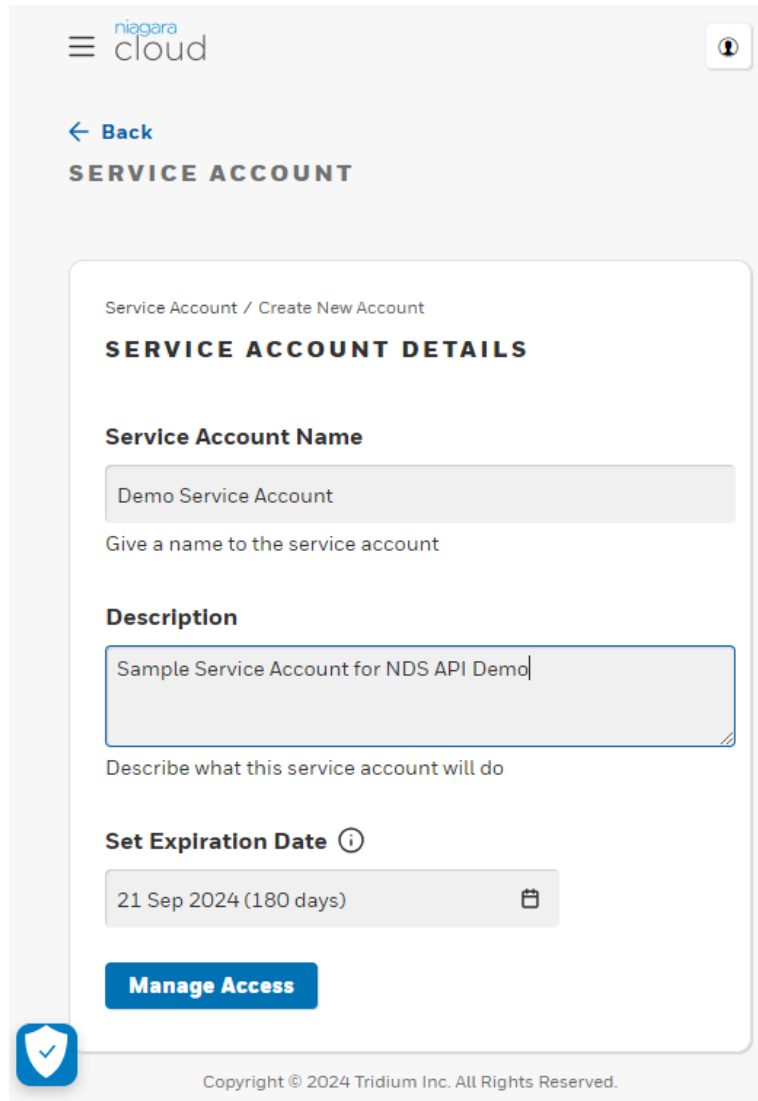
< 1 2 >

Go to Page



Copyright © 2023 Tridium Inc. All Rights Reserved.

Create a Service Account



The screenshot shows the 'Create New Account' form in the Niagara Cloud interface. The form is titled 'SERVICE ACCOUNT DETAILS' and includes the following fields:

- Service Account Name:** A text input field containing 'Demo Service Account'.
- Description:** A text input field containing 'Sample Service Account for NDS API Demo'.
- Set Expiration Date:** A date picker showing '21 Sep 2024 (180 days)'.

At the bottom of the form is a blue button labeled 'Manage Access'. The footer of the page includes a copyright notice: 'Copyright © 2024 Tridium Inc. All Rights Reserved.'

Service Account Name

Description

Expiration Date

Create a Service Account

PROJECTS AND DEVICES ACCESS ⓘ

Create Account

Role

User

Delete Role

☐ All Projects

☒ Richmond, VA (1/2 Devices)

Search

☐ alphaJ9 ☒ VAH2J80...

Role

User

User

NDS Operator

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

Using NDS APIs

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

Get an Access Token

OAuth request data

- grant_type: client_credentials
- scope: ncp:read

```
} private NdsApiTokenHolder getAccessToken() throws Exception {  
    // Use form data for authentication type and scope  
    Map<Object, Object> requestData = new HashMap<>();  
    requestData.put("grant_type", "client_credentials");  
    requestData.put("scope", "ncp:read");  
}
```

Get an Access Token

Build HTTP Request

- Token endpoint URI is documented
- client_id: from Service Account
- client_secret: from Service Account

```
// Build the HTTP Request
HttpRequest tokenRequest = HttpRequest.newBuilder()
    // Use NCS authentication endpoint from configuration file
    .uri(URI.create("https://auth.pingone.com/2bb83e4b-dd73-47d3-984b-6e601302d766/as/token"))

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

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

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

Get an Access Token

Send request

Deserialize return value

Store 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;  
... }
```

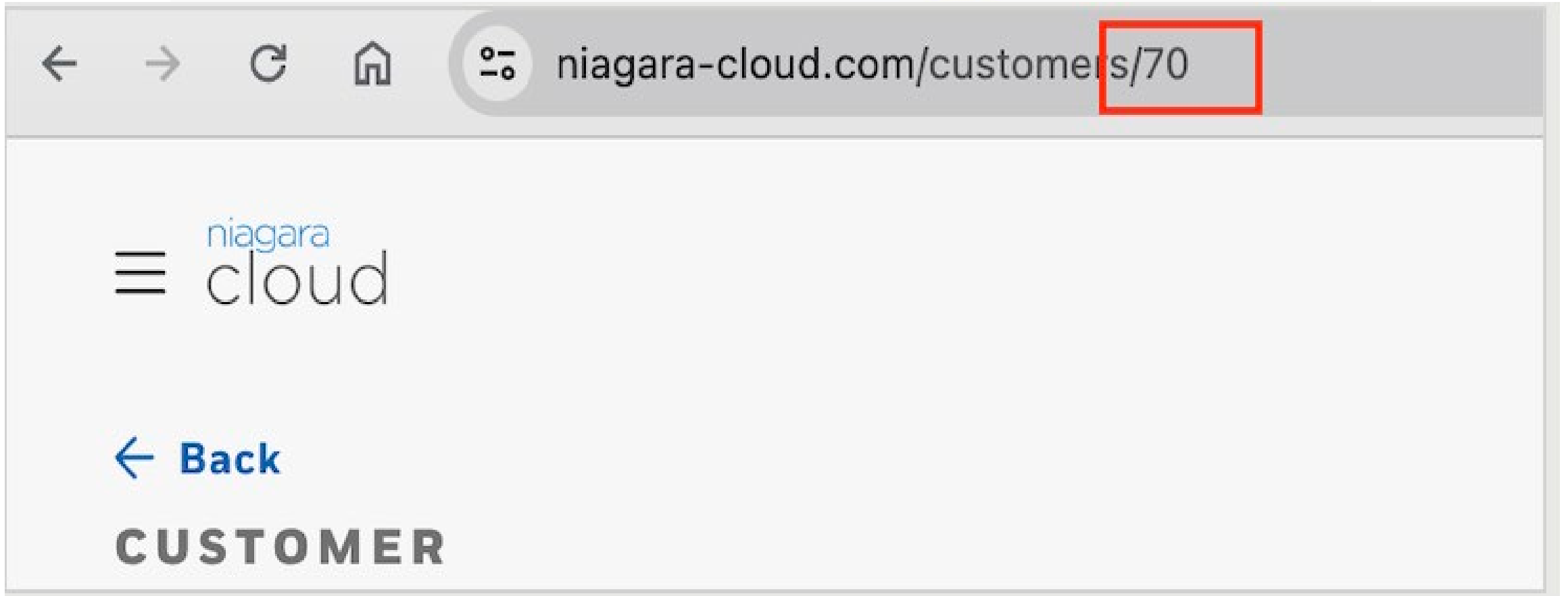
Get an Access Token

```
{  
  "access_token" : "eyJhbGciOiJSUzI1NiIsImtpZCI6ImRlZmF1bHQifQ.eyJjbGllbnRfaWQiOiJ1b3RlciIsImV4cCI6MTYxMjM0MDAwfQ",  
  "token_type" : "Bearer",  
  "expires_in" : 3600,  
  "scope" : "ncp:read"  
}
```

Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Use Model API to find semantic metadata

Model API



Model API

Search by Tag
Value

RPK Authenticator Device Registration

Device Registered	
Authenticator Enabled	
Device Connected	
System Type	
System GUID	68ce1fb6-f5b5-4ee2-aa5c-b94e9e79563e

Register Device

Force re-authentication

Model API

Search by Tag
Name

```
{  
  "searchType": "tagName",  
  "systemGuid": "11223344-5566-7788-99aa-bbccddeeff00",  
  "searchItems": [  
    "n:point,hs:air"  
  ]  
}
```

Model API

Search by Point
Name

- Comparison
Type - any,
exact, contains

```
{  
  "searchType": "pointName",  
  "systemGuid": "11223344-5566-7788-99aa-bbccddeeff00",  
  "searchItems": [  
    "SpaceTemp"  
  ],  
  "comparisonType": "exact"  
}
```

Query Model API

```
private List<Point> modelQuery(String accessToken) throws Exception {  
    // Build EMS search query  
    SearchCriteria searchQuery = new SearchCriteria(  
        queryConfig.getSearchType(),  
        queryConfig.getSystemGuid(),  
        queryConfig.getSearchItems(),  
        queryConfig.getComparisonType());  
    String queryBody = mapper.writeValueAsString(  
        searchQuery  
    );  
}
```

Query Model API

```
// Build EMS query request  
String route = String.format("/api/v1/customers/%s/pointNames", queryConfig.getCustomerId());  
HttpRequest modelQuery = HttpRequest.newBuilder()  
    .uri(URI.create("https://www.niagara-cloud.com" + route))  
    .POST(HttpRequest.BodyPublishers.ofString(queryBody))  
    .header(s: "Content-Type", s1: "application/json")  
    .header(s: "Authorization", s1: "Bearer " + accessToken)  
    .build();
```

Query Model API

Send request

Deserialize return value

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);  
List<Point> points = Optional.ofNullable(body.getContent()).orElse(new EmsQueryResults()).getPoints();  
return points;
```

Model API Response

```
{
  "_embedded" : {
    "pointModels" : [ {
      "systemGuid" : "a3151207-a891-4caa-8a53-5bd9d28a8d2c",
      "cloudId" : "762262b9-6179-4387-bbb3-01b560de405a",
      "name" : "CCVlvPos",
      "tags" : [ {
        "tagId" : "n:name",
        "tagValue" : "CCVlvPos"
      }, {
        "tagId" : "n:type",
        "tagValue" : "control:NumericWritable"
      }, {
        "tagId" : "nc:cloudId",
        "tagValue" : "762262b9-6179-4387-bbb3-01b560de405a"
      }, {
        "tagId" : "nc:telemetryId",
        "tagValue" : "d30ac887-2c91-4ce2-b9d0-50a1e893a221"
      } ],
      "relations" : [ {
        "inbound" : true,
        "rId" : "n:parent",
        "cId" : "ad0646c3-b45b-4c92-91f5-87339d0daaa7"
      }, {
        "inbound" : false,
        "rId" : "n:child",
        "cId" : "ad0646c3-b45b-4c92-91f5-87339d0daaa7"
      } ]
    } ],
    "_links" : {
      "self" : [
        {
          "href" : "https://www.niagara-cloud.com/api/v1/entitymodel/customers/5000274/pointNames?page=0&size=20&sort=id,asc"
        }
      ]
    },
    "page" : {
      "size" : 20,
      "totalElements" : 3,
      "totalPages" : 1,
      "number" : 0
    }
  }
}
```

Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Use Model API to find semantic metadata
4. Use History API to get history data

History Data API

Query Parameters

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

egress-controller

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

Parameters

No parameters

Request body required application/json

Example Value | Schema

```
{
  "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 TS DB via History API

```
private TsdbQueryResponse readTelemetryData(String accessToken, String systemGuid, String telemetryId)
    throws Exception {

    // Build telemetry query
    TsdbQuery query = TsdbQuery.builder()
        .systemGuid(systemGuid)
        .startTimeStr(queryConfig.getStartTimeStr())
        .endTimeStr(queryConfig.getEndTimeStr())
        .recordLimitStr("5")
        .cloudId(List.of(telemetryId))
        .build();
    String queryBody = mapper.writeValueAsString(query);
```

Query TS DB via History API

```
// Build request for Egress Service  
HttpRequest telemetryRequest = HttpRequest.newBuilder()  
    .uri(URI.create("https://www.niagara-cloud.com/api/v1/egress/telemetry"))  
    .POST(HttpRequest.BodyPublishers.ofString(queryBody))  
    .header(s: "Content-Type", s1: "application/json")  
    .header(s: "Authorization", s1: "Bearer " + accessToken)  
    .build();
```

Query TS DB via Egress API

Send request

Deserialize return value

Return list of history records

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

Query TS DB via Egress API

```
{
  "systemGuid": "a3151207-a891-4caa-8a53-5bd9d28a8d2c",
  "recordLimit": 1,
  "pointDetails": [{
    "pointId": "d30ac887-2c91-4ce2-b9d0-50a1e893a221",
    "preRecord": null,
    "postRecord": null,
    "historyRecords": [{
      "time": "2024-04-12T15:45:00.03Z",
      "value": 10.0,
      "properties": "{trendFlags.outOfOrder=false, status.overridden=false, st
    }]
  }]
}
```

Using NDS APIs

1. Create a Service Account
2. Obtain an access token
3. Use Model API to find semantic metadata
4. Use History API to get history data
5. Use Command API to read/write values to station

Command Service API

Execute read, write, and read inputs commands

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/readAll API to read all current input values for one or more points



Executing a Read and ReadAll Command

Commands target points by cloudId

```
private ReadCommandResponse readPointCommand(String accessToken, String systemGuid, String cloudId)
    throws Exception
{
    ReadCommandRequest request = ReadCommandRequest.builder()
        .cloudIds(List.of(cloudId))
        .build();
    String requestBody = mapper.writeValueAsString(request);
```

Executing a Write Command

Write command also takes Value, Duration, and Priority level

```
private WriteCommandResponse writePointCommand(String accessToken, String systemGuid, String cloudId,
Object value, int duration, int priority)
throws Exception
{
    WriteCommandRequest request = WriteCommandRequest.builder()
        .points(List.of(WriteInfo.builder()
            .cloudId(cloudId)
            .value(value)
            .duration(duration)
            .inputPriority(priority)
            .build()))
        .build();
    String requestBody = mapper.writeValueAsString(request);
```

Executing a Command

```
// Build request for Command Service
String route = String.format("/devices/%s/commands/", systemGuid);
HttpRequest commandRequest = HttpRequest.newBuilder()
    .uri(URI.create("https://www.niagara-cloud.com" + route + "read"))
    .POST(HttpRequest.BodyPublishers.ofString(requestBody))
    .header(s: "Content-Type", s1: "application/json")
    .header(s: "Authorization", s1: "Bearer " + accessToken)
    .build();
```

Executing a Command

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

Executing a Command

Read command results

```
{  
  "pointReadDetails": [{  
    "cloudId": "c68b9793-19e1-4d9a-9cae-899ae2b32735",  
    "value": "63.00",  
    "status": "{ok} @ 16"  
  }]  
}
```

Executing a Command

Write command results

```
{  
  "pointWriteDetails": [{  
    "cloudId": "c68b9793-19e1-4d9a-9cae-899ae2b32735",  
    "status": "success",  
    "writeTime": "2024-04-12T12:11:10.625-04:00"  
  }]  
}
```

Executing a Command

ReadAll command results

```
{
  "pointReadAllDetails": [{
    "cloudId": "c68b9793-19e1-4d9a-9cae-899ae2b32735",
    "inputValues": [{
      "value": "0.00",
      "inputPriority": "1",
      "status": "{null}",
      "sourceId": ""
    }, {
      "value": "63.00",
      "inputPriority": "16",
      "status": "{ok}",
      "sourceId": "472cba62-424f-417c-824a-379431179434"
    }, {
      "value": "55.00",
      "inputPriority": "def",
      "status": "{ok}",
      "sourceId": ""
    }
  ]
}]
}
```

Let's see this in action...

NDS API Usage Demo

```
java -jar ^
-Dspring.profiles.active=local ^
target\nds-api-demo-0.0.1-SNAPSHOT.jar
```

[illegible]

Want More?

Come see us at the Tridium Booth

- More demos
- Talk to experts

Talks

- Leveraging Niagara Cloud Suite
 - Tomorrow at 2pm, Pacific Ballroom C-D

GIVE YOUR FEEDBACK NIAGARA CLOUD SUITE

- Come to the Tradeshow Hall
- Experience a live demo of Niagara Cloud Suite
- Provide your feedback & help shape the future of Niagara Cloud Suite
- (& win one-of-many prizes)



niagara
cloud SUITE™

DEVELOPER

Niagara Cloud Suite™ is a scalable cloud-based solution that provides secure, remote building management services.

This offering provides new integration, connection, and deployment capabilities, building on Tridium's current model of open and extensible integration with devices, services and applications.

TRIDIUM

Complete this short survey to provide your feedback on Niagara Cloud Suite!

WHAT'S IN IT FOR YOU?

- ✓ Win one-of-many prizes that will be awarded at the end of Niagara Summit
- ✓ Knowledge that your feedback will help us continue to enhance Niagara Cloud Suite

TRIDIUM



niagara
cloud SUITE™
DEVELOPER

NAME:

EMAIL:

PHONE:

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 is available on the Tridium GitHub Repository at
 - <https://github.com/tridium/nds-api-demo/releases/tag/summit24>

Thank You!

NS2024

POWER OF PARTNERSHIP