

Device OnBoarding And Connection to the Dosatsu Platform

- [Introduction](#)
- [Connection Methods](#)
 - [MQTTS](#)
 - [Step#1 Equipment Registration](#)
 - [Step#2 Download device certificates](#)
 - [Step#3 Install certificate](#)
 - [Step#4 Equipment Connection](#)
 - [HTTPS](#)
 - [Step#1 IoConnection Registration](#)
 - [Step#2 Download IoConnection certificates](#)
 - [Step#3 Use IoConnection certificates](#)
 - [Step#4 Equipment Connection](#)
 - [Equipment Blocking](#)
 - [Trouble shooting](#)
 - [HTTP Status 401](#)
 - [HTTP Status 400](#)
 - [HTTP Status 200](#)

Introduction

In the document of Platform API, we have describe the APIs how to interact with the services in the Dosatsu platform. All the data managed by the Dosatsu are posted by the APIs, such as equipment and relevant data. To ensure that the APIs are used legitimately and its data transmission is secure, we need to onboard the device and secure its connection to the Dosatsu platform. The Dosatsu platform allows two connection methods to onboard device and secure connection:

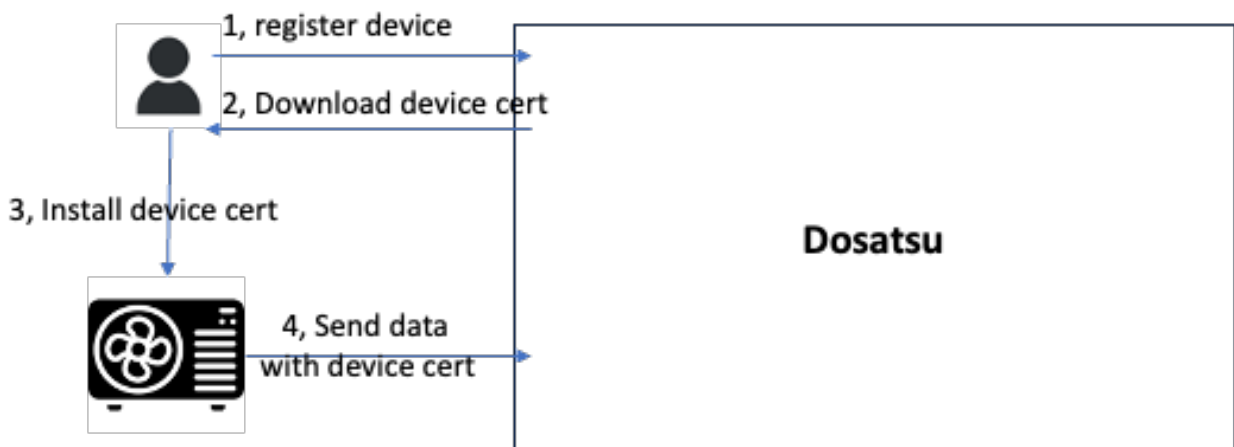
- MQTTS: For an individual device, register the device to get certificates and key for secure connection, install certificates and key on device and service, use certificate to secure a MQTTS connection with the Dosatsu platform and call the API to send data to the Dosatsu platform.
- HTTPS: For multiple equipments which share the same secure connection mechanism, typically multiple equipments managed by a same edge, register the IOConnection to get certificates and key for secure connection, install certificates and key on device and service, use certificate to secure a HTTPS connection with the Dosatsu platform to get access token for the device, and use access token to secure a HTTPS connection with the Dosatsu platform and call the API to send data to the Dosatsu platform.

In the later sections, we will explain the details of the above two connections so that the users can onboard device and secure connection to send data to the Dosatsu platform. As for sending different types of data, please refer the document of Platform API.

In this document, we use cp-stage.daikinlab.com as the base URL of the Dosatsu Platform.

Connection Methods

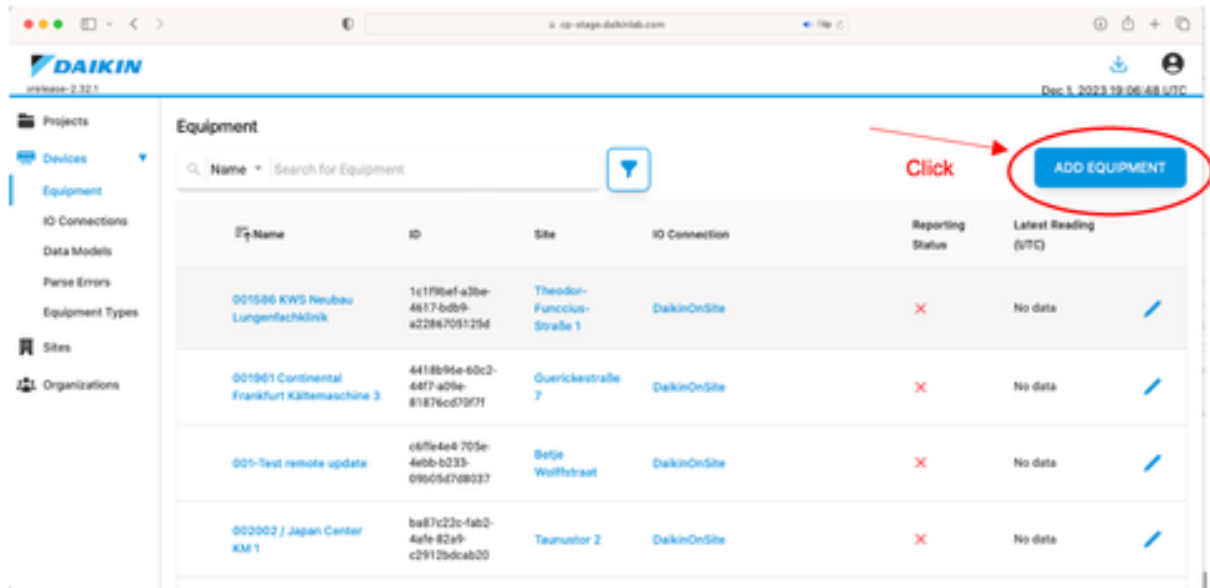
MQTT



The steps in the above picture are detailed below.

Step#1 Equipment Registration

An equipment/device that would connect to the common platform, needs to be first registered at <http://cp-stage.daikinlab.com/app/equipment>

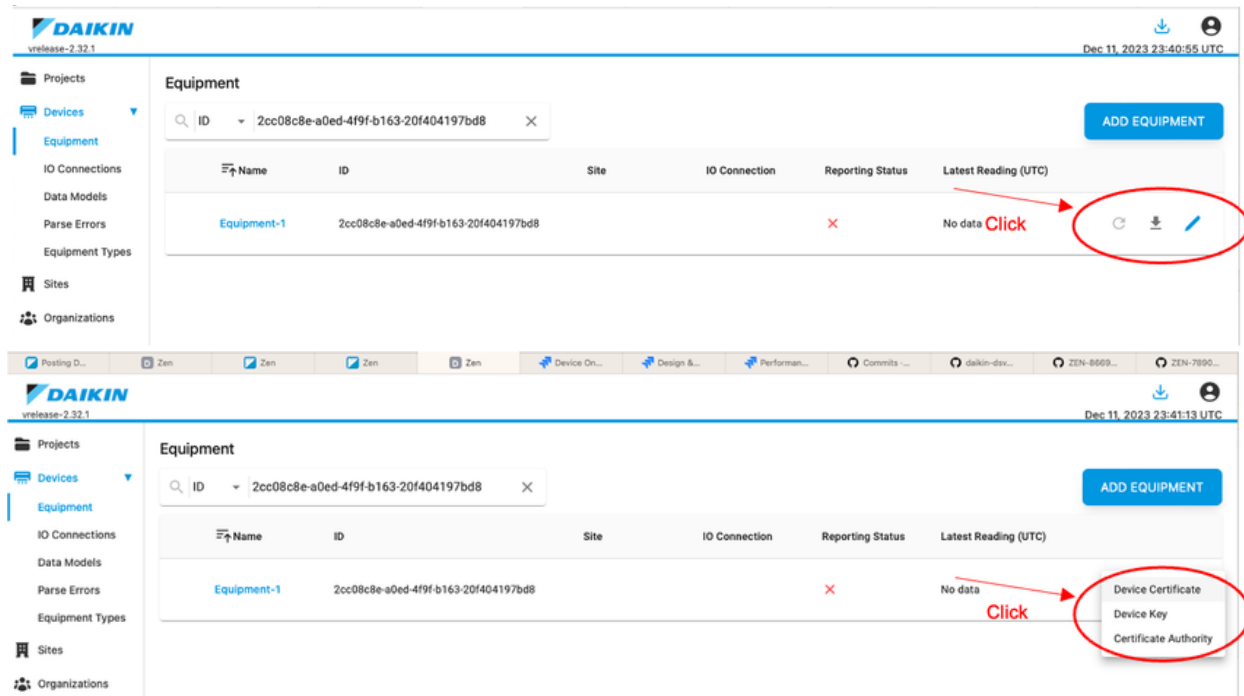


Create an equipment clicking on the blue button “ADD EQUIPMENT”.

The screenshot shows the 'Create an Equipment' form. The form includes fields for Name, Equipment Type, Reporting Frequency (seconds), Site, and Owner. A blue button with a plus sign is at the bottom.

Step#2 Download device certificates

Download the device certificate(s) by clicking the download icon. It will prompt you to save the certificates (device cert, device cert key, root ca) to your disk.



The download file names would by default be:

- Device certificate – <deviceid>.cert
- Device certificate key – <deviceid>-key.cert
- Root CA (Certificate Authority) – cp-stage-daikinlab-com.cert

Step#3 Install certificate

On an edge device, the certificate should be stored securely. Preferably on a hardware security module (HSM). For this test, download to a safe place on your computer.

Step#4 Equipment Connection

Collect Equipment ID and certificates from the previous steps before proceeding to the next step for device connections.

Use them to connect the Dosatsu platform over MQTTS. For example if you are using MQTT.fx to test connection or simulate a device, follow these instructions:

Configure MQTT.fx tool broker settings (address: <http://cp-stage.daikinlab.com> , Port: 8883) and enter Equipment ID from the Dosatsu platform into “Client ID” text field

MQTT Broker Profile Settings

Profile Name:

Profile Type:

Broker Address:

Broker Port:

Client ID:

General | User Credentials | SSL/TLS | Proxy | LWT

Connection Timeout:

Keep Alive Interval:

Clean Session: ☒

Auto Reconnect: ☐

Max Inflight:

MQTT Version: ☒ Use Default

General setting tab enter the following numbers for longer connection window

The image shows a configuration window for an MQTT client with the following settings:

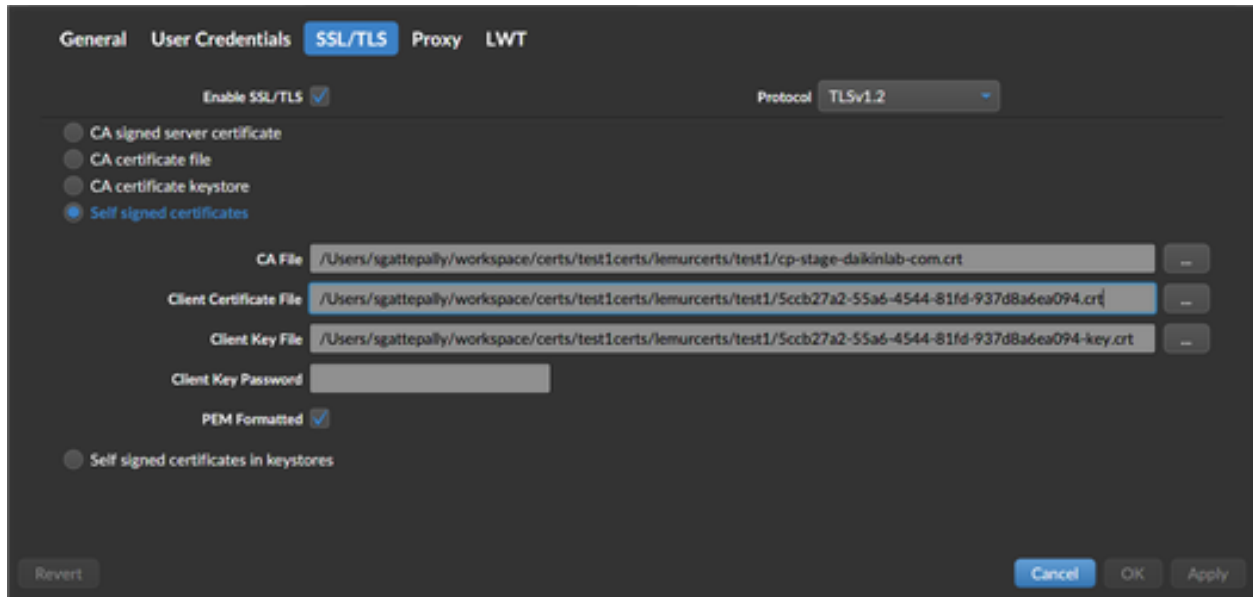
Setting	Value
Connection Timeout	60
Keep Alive Interval	30
Clean Session	<input checked="" type="checkbox"/>
Auto Reconnect	<input checked="" type="checkbox"/>
Max Inflight	10
MQTT Version	<input checked="" type="checkbox"/> Use Default 3.1.1
<input type="button" value="Clear Publish History"/>	
<input type="button" value="Clear Subscription History"/>	

Buttons at the bottom: Revert, Cancel, OK, Apply.

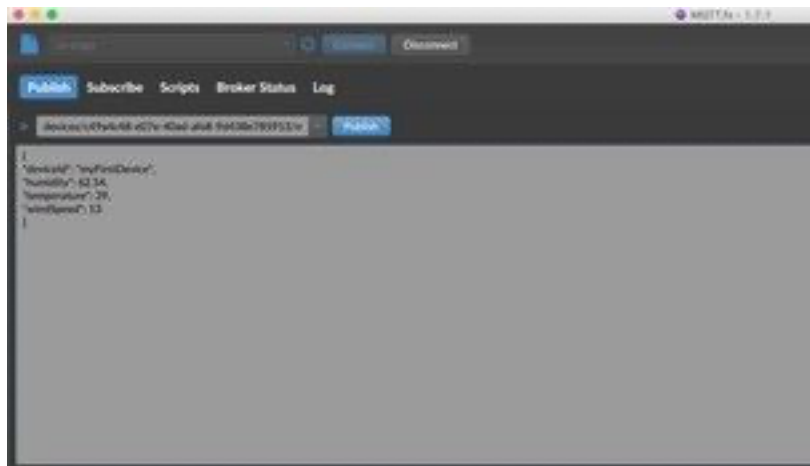
Enter dsvRbMq/<password> in the “User Credentials” tab. Get password from Support

The screenshot shows the 'Edit Connection Profiles' window. At the top, the 'Profile Name' is 'cp-stage' and the 'Profile Type' is 'MQTT Broker'. Below this, the 'MQTT Broker Profile Settings' section includes fields for 'Broker Address' (cp-stage.daikinlab.com), 'Broker Port' (8883), and 'Client ID' (c48a4c48-e07e-40ad-af8b-9d438c785913) with a 'Generate' button. A tab bar at the bottom of the settings section includes 'General', 'User Credentials' (which is selected and highlighted in blue), 'SSL/TLS', 'Proxy', and 'LWT'. Under the 'User Credentials' tab, the 'User Name' field contains 'c48a4c48' and the 'Password' field contains a series of dots. These two fields are circled in red. At the bottom of the window are buttons for 'Reset', 'Cancel', 'OK', and 'Apply'.

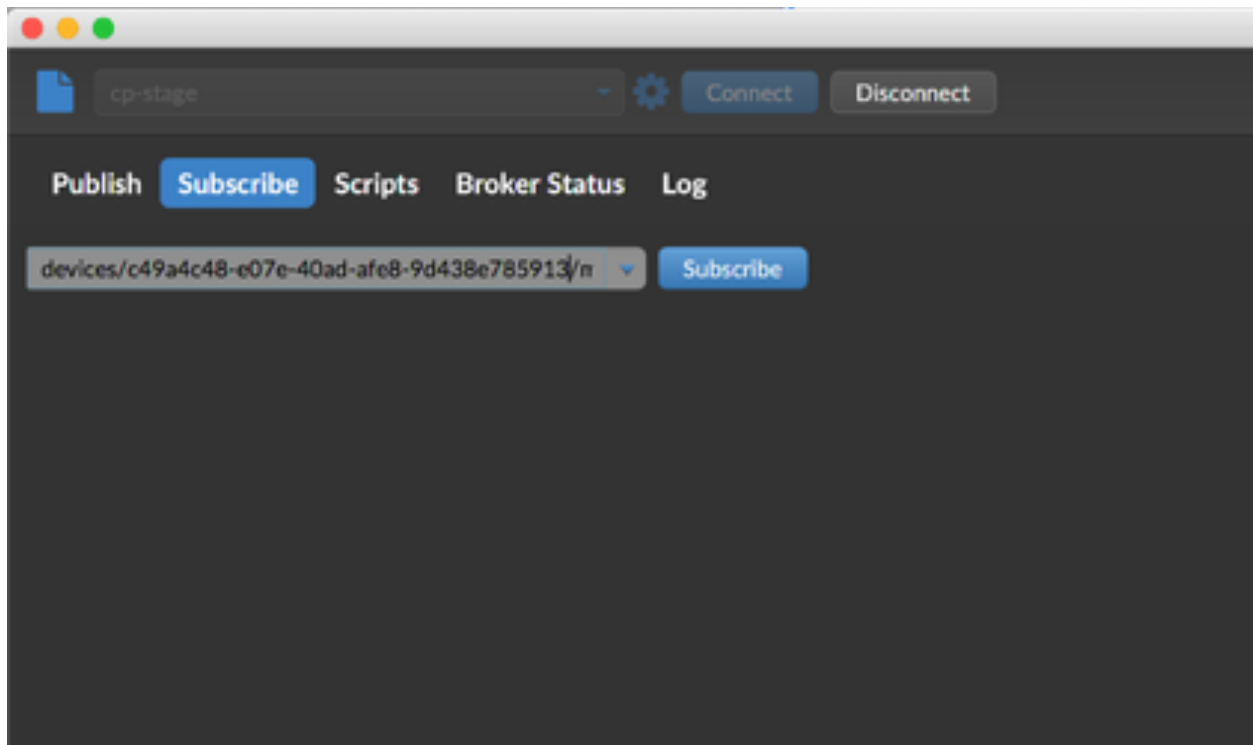
1. SSL/TLS: Select the cert files for SSL/TLS settings as shown in the picture. Earlier on this page in “Step#2 Download device certificate”, certificates were downloaded. We need to use them here.
 1. For CA File choose “cp-stage-daikinlab-com.crt” file.
 2. For Client Certificate File choose the Device certificate – <deviceid>.crt
 3. For Client Key File choose Device certificate key – <deviceid>-key.crt
 4. For Client Key Password: Keep it empty.



1. Apply and click connect on the main window.
2. To post readings to the Dosatsu platform, enter “devices/<deviceid>/messages/events” in the publish topic name and enter payload and “publish”.



To receive settings from the Dosatsu platform, enter “devices/<deviceid>/messages/devicebound” in the subscribe window and press “subscribe”.

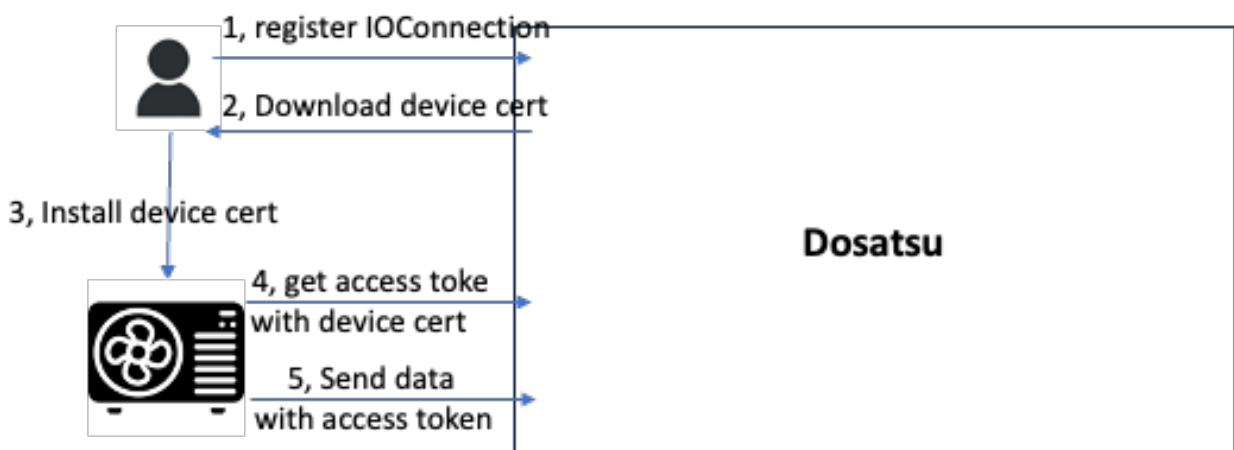


Equipment Blocking:

An administrator can block connections of a device by performing these steps:

1. Revoking certificate.
2. Change device status to 'disabled'.

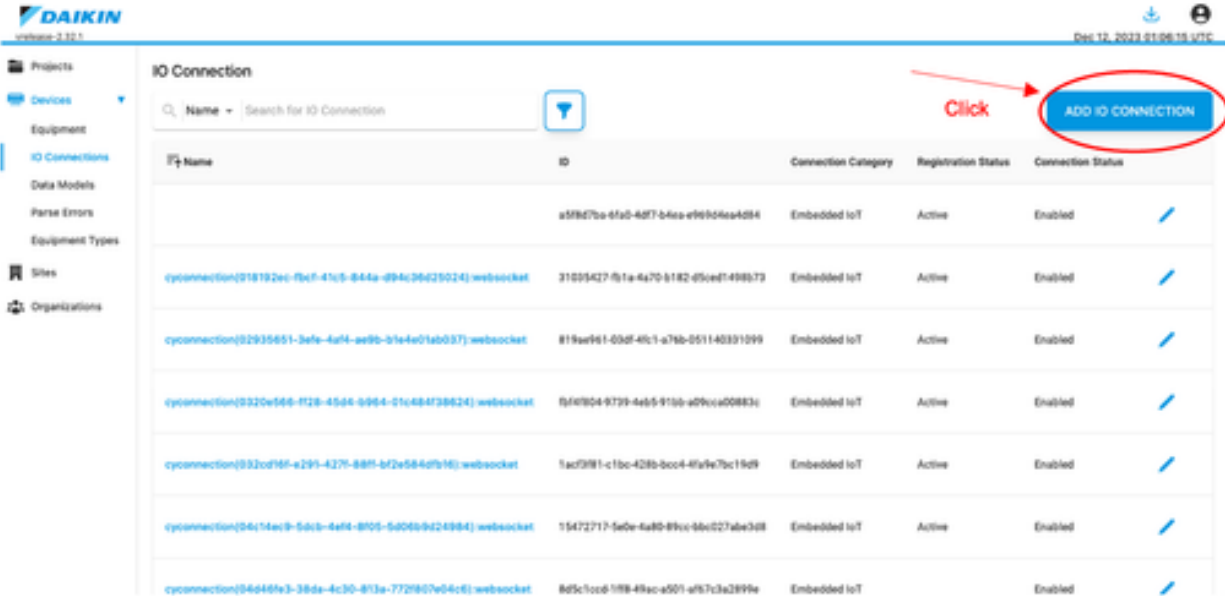
HTTPS



The steps in the above picture are detailed below.

Step#1 IoConnection Registration

For an ioconnection to connect to the common platform, it needs to be first registered at <http://cp-stage.daikinlab.com/app/ioconnection>



The screenshot displays the DAIKIN IoConnection management interface. On the left is a sidebar with navigation options: Projects, Devices, Equipment, IO Connections (selected), Data Models, Parse Errors, Equipment Types, Sites, and Organizations. The main area is titled 'IO Connection' and features a search bar with a filter icon. Below the search bar is a table listing existing connections. A red arrow points to a blue button labeled 'ADD IO CONNECTION' in the top right corner, with the word 'Click' written next to it.

Name	ID	Connection Category	Registration Status	Connection Status
	a5f8d7ba-6fa0-4d07-b4ee-e96b04ea4084	Embedded IoT	Active	Enabled
cyconnection(018192ec-fbc0-41c5-844a-d94c36d25024) websocket	310514327f51a-4a70-b182-d5ced1498b73	Embedded IoT	Active	Enabled
cyconnection(02930651-3efe-4af4-ae9b-b1e4e01ab037) websocket	819ae961-03d0-40c1-a76b-051140331099	Embedded IoT	Active	Enabled
cyconnection(0320e566-0128-45d4-9064-01c484738624) websocket	05f0f04-9739-4ab5-910b-a09uca00883c	Embedded IoT	Active	Enabled
cyconnection(032cd165-e295-4376-8805-bf2e584d0b16) websocket	1ac03081-c1bc-428b-bcc4-4fa9e7bc19d9	Embedded IoT	Active	Enabled
cyconnection(04c14ac9-5dcb-4ef4-8005-5d04b9d24384) websocket	115472717-5a0e-4a80-89cc-06c027abe3d8	Embedded IoT	Active	Enabled
cyconnection(04d446fe-338da-4c30-8f3a-7720907e04c6) websocket	8d5c1cc0-1088-49ac-a501-a057c3a3899e	Embedded IoT		Enabled

Create an ioconnection clicking on the blue button “ADD IO CONNECTION”.

Create an IO Connection

Name *

Connection Status * **Enabled**

Registration Status **Active**

Description

Contact Email * **sen.cao@dsv.daikin.com**

Connection Category * **Embedded IoT**

☐ Request Certificates

Configure Parsers

Channel Data Parser **DefaultParser** **Configure**

Secrets

Custom Attributes

CANCEL **SAVE**

To generate certificates for a new IoConnection, it is required to mark the checkbox “Request Certificates”.

Step#2 Download IoConnection Certificates

Download the device certificate(s) by clicking the download icon. It will prompt you to save the certificates (device cert, device cert key, root ca) to your disk.

DAIKIN release-2.32.1 Dec 12, 2023 01:15:05 UTC

IO Connection

ID

ADD IO CONNECTION

Name	ID	Connection Category	Registration Status	Connection Status
IoConnection-1	051dcfb5-377e-48bf-bcde-88a230af82f2	Embedded IoT	Active	Enabled

Click

Step#3 Use IoConnection Certificates

In any edge cloud which aggregates equipment data in a region and use an IOConnection to send the equipment data to our cloud, the certificate should be stored securely in the edge cloud. The certificates are used as below:

- Device certificate – <deviceid>.crt: The certificate for the IOConnection.
- Device certificate key – <deviceid>-key.crt: The certificate key for the IOConnection.
- Root CA (Certificate Authority) – cp-stage-daikinlab-com.crt: The certificate to establish HTTPS connection to the HTTP service in our cloud.

Equipment Connection

The connection requires two steps:

1. Get access token

1. POST `https://sso.daikinlab.com/auth/realms/daikin/protocol/openid-connect/token`
2. Request Headers:
3. Content-Type:application/x-www-form-urlencoded
4. Request Body:
5. `grant_type:client_credentials`
6. `client_id:cpiot`
7. `client_secret:<CLIENT_SECRET>`
8. Response:
9. {
10. `"access_token": "<ACCESS_TOKEN>",`
11. `"expires_in": 300,`
12. `"token_type": "bearer",`
13. `"not-before-policy": 1582439696,`
14. `"session_state": "5873b3b2-868c-4516-9b99-ec03cc472d11",`
15. `"scope": "profile email"`
16. }

2. Post Data: Ex:

1. POST `https://cp-stage.daikinlab.com/data`
2. Headers:
3. `ioconnectionref:<IOCONNECTION_ID>`
4. Authorization:Bearer <ACCESS_TOKEN>
5. Body:
6. {
7. `"category_name": "Allergen Count",`
8. `"category_color": "blue",`
9. `"parameter_name": "Alternaria Mold",`
10. `"log_value": 24,`
11. `"particle_category": "1",`
12. `"env_index": "4",`
13. `"log_unit": "particles",`
14. `"log_intensity": "Moderate",`
15. `"t": "2019-06-11T00:00",`
16. `"networkRef": "111-000-111"`
17. }

Equipment Blocking

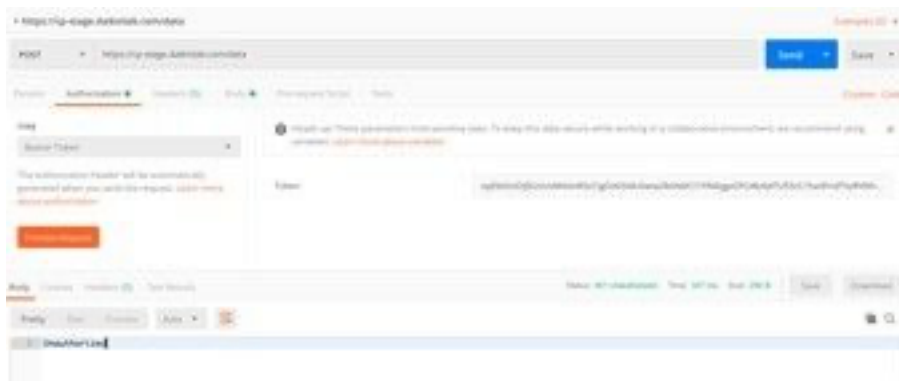
An administrator can block connections of a device by performing these steps:

1. Change device status to 'disabled'.

Trouble shooting

HTTP Status 401

HTTP Status 401 and HTTP Response “Unauthorized” indicates an expired or invalid JWT token.



If the access token expires, get a fresh access token:

POST <https://sso.daikinlab.com/auth/realms/daikin/protocol/openid-connect/token>

Request Headers:

Content-Type: application/x-www-form-urlencoded

Request Body:

grant_type: client_credentials

client_id: cpiot

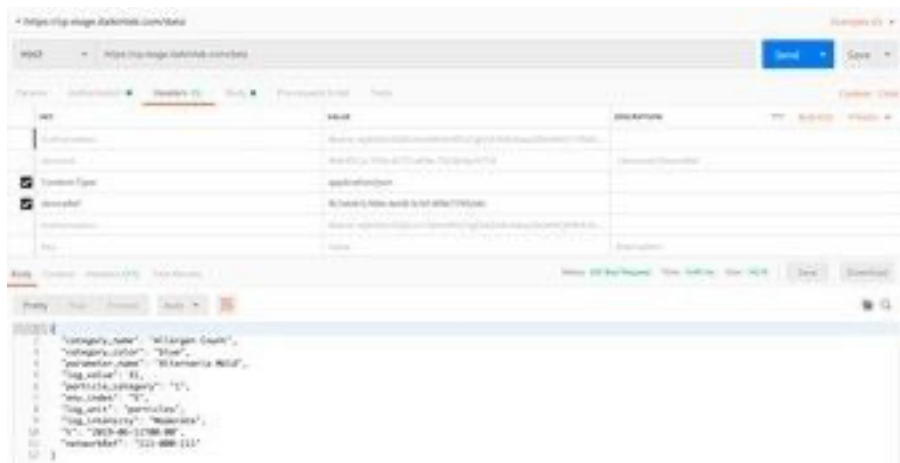
client_secret: <CLIENT_SECRET>

Response:

```
{
  "access_token": "<ACCESS_TOKEN>",
  "expires_in": 300,
  "token_type": "bearer",
  "not-before-policy": 1582439696,
  "session_state": "5873b3b2-868c-4516-9b99-ec03cc472d11",
  "scope": "profile email"
}
```

HTTP Status 400

HTTP Status 400 indicates an invalid/disabled equipment ID or invalid json body.



HTTP Status 200

HTTP Status 200 indicates a successful connection & Response

