TECHNICAL BRIEF

Accessing IQ Gateway local APIs and local UI using token

Applicable regions: North America

Contents

1. Overview

At Enphase, we create high-quality solutions that meet the highest security standards. Many installers and homeowners use local APIs or local UI on the IQ Gateway to access data. These interfaces were, in the past, protected by conventional password-based authentication. With IQ Gateway software version 7.0.x or higher, local UI and APIs need cryptographic token-based authentication to improve security. This technical brief explains:

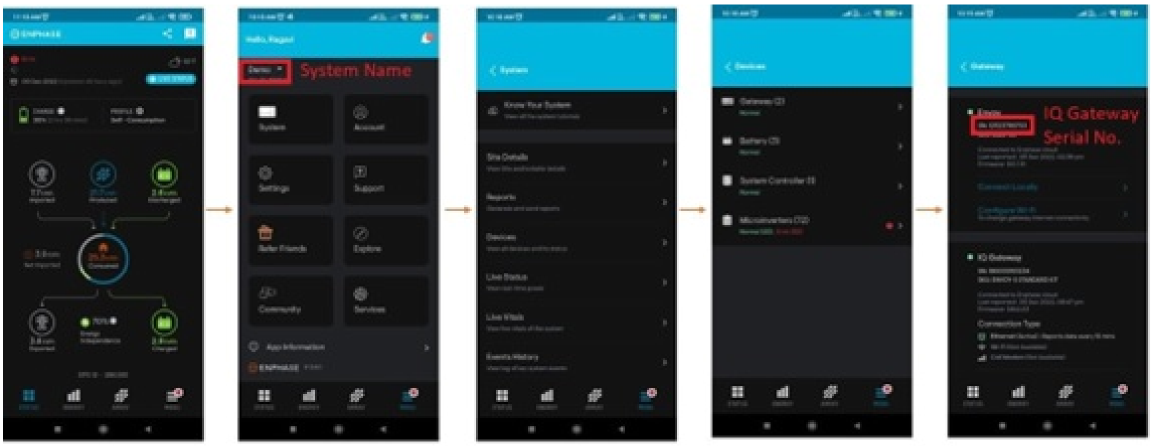
1. How to get the system name and IQ Gateway serial number?
2. How to get a token for your IQ Gateway?
3. How to access IQ Gateway local UI and APIs using the token?

The applicable SKUs are listed below:

Product	SKU
IQ Gateway	ENV-IQ-AM1-240
IQ Gateway	ENV2-IQ-AM1-240
IQ Gateway Commercial	ENV-IQ-AM3-3P
IQ Gateway Commercial 2	ENV2-IQC2-AM3-3P
IQ Gateway Metered	ENV-S-WM-230
IQ Gateway Metered	ENV-S-EM-230
IQ Gateway Standard	ENV-S-WB-230
IQ Gateway M	ENV-S-AM1-230-60
Envoy S Standard	ENV-S-AB-120-A
Envoy S Metered NA	ENV-S-AM1-120

2. Get the system name and IQ Gateway serial number

To get the system name and IQ Gateway serial number from the Enphase App, follow the steps described below:



1. Open the Enphase App.
2. Tap the **Menu** icon on the bottom right corner. The system name is displayed under the **Demo** dropdown menu.
3. Copy the system name and save it for later use.
4. In the menu items, select **System** > **Devices** > **Gateway**. The IQ Gateway serial number is specified with the label **SN:** under the corresponding IQ Gateway.
5. Copy the serial number and save it for later use.

3. Generate a token

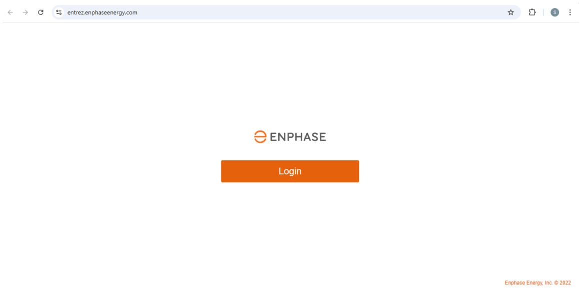
To get the system name and IQ Gateway serial number from the Enphase App, follow the steps described below:

1. Get a token through web UI.
2. Get a token programmatically through GET on a URL.
3. Retrieve the token through a Shell script.
4. Retrieve the token through the Python script.

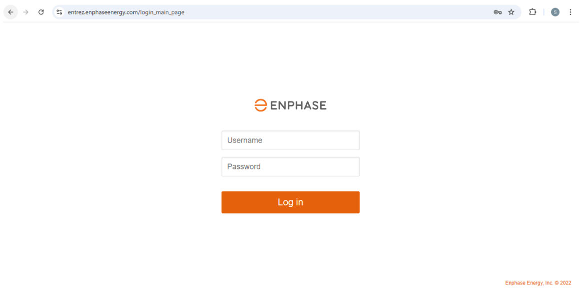
3.1 Get a token through web UI

Follow this method when you require a one-time use token or want to avoid programmatically generating a token.

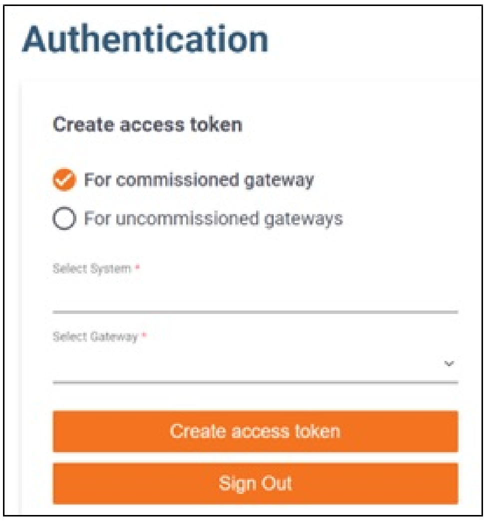
1. Open a browser and go to https://entrez.enphaseenergy.com.
2. Click the **Login** button.



3. Enter your Enphase Account credentials.

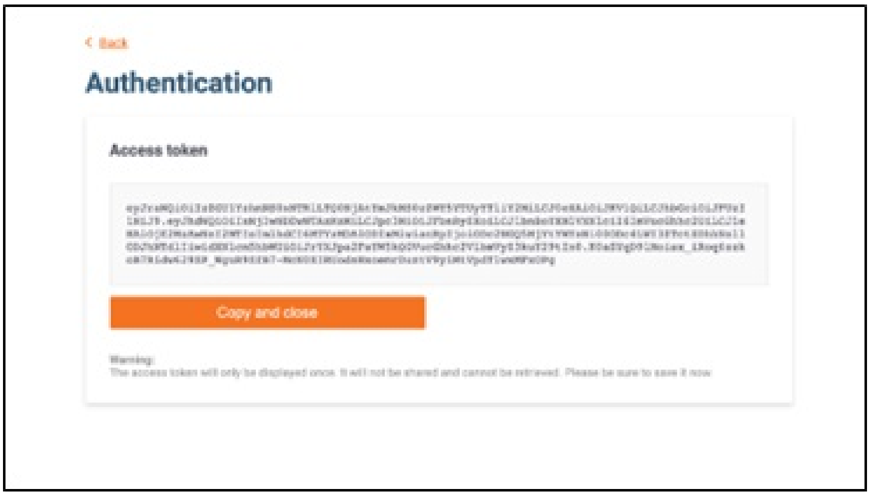


4. Select the system name and IQ Gateway serial number (use the system name and IQ Gateway serial number you got from the section Get the system name and IQ Gateway serial number) from the dropdown menu.



5. Click **Create access token** to generate a token.

6. In the **Authentication** window, click the **Copy and close** button to copy the token.



7. Once copied, paste the token,
    a. In the home automation setup where access to the IQ Gateway local APIs is required, or
    b. Into the browser, where access to the IQ Gateway local UI is required.
8. Save the token securely for future use.

For details on accessing the IQ Gateway local UI or local APIs using a token, refer to the section Access the IQ Gateway using a token.

NOTE: Tokens are valid for a finite time.

- The token is valid for one year if the user is a system owner.
- The token is valid for 12 hours if the user is an installer.

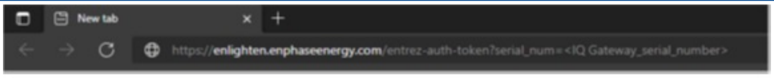
If the installer and system owner credentials are used as a self-installer, then the web UI-based token retrieval outlined in this section will result in a token valid for 12 hours. The owner can contact Enphase Support to change the credentials from the installer credentials to the system owner credentials. Alternatively, the owner can use the programmatical route to retrieve tokens outlined in the following section.

3.2 Get a token programmatically through GET on a URL

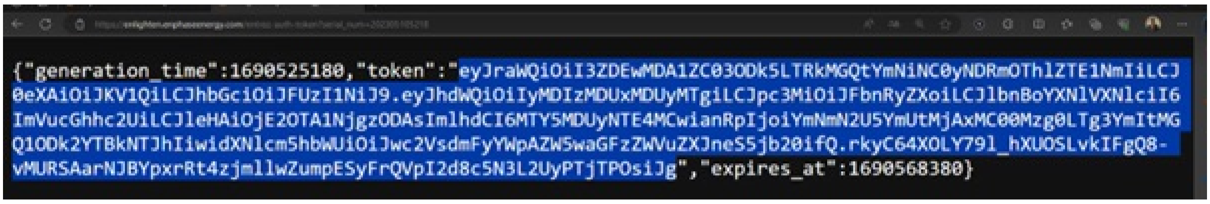
Follow the procedure to retrieve a token programmatically. Enphase provides a URL to retrieve a token and mentions the duration for which the token is valid.

To construct the correct URL for the IQ Gateway:

1. Paste the token retrieval URL into the web browser's address bar: https://enlighten.enphaseenergy.com/entrez-auth-token?serial_num=<IQ Gateway_serial_number>



2. Replace **<IQ Gateway_serial_number>** in the above URL with the serial number of the specific IQ Gateway that you obtained. Refer to the section Get the system name and IQ Gateway serial number.
3. The token is displayed as shown in the following image. Save the token securely for future use.



NOTE: When the token retrieval URL is accessed through the web browser, the browser does an HTTPS GET on the URL. The content of the response can be seen in the browser. This content contains the token, expected expiry date, and time in UNIX epoch time stamp format. The duration for which the token is valid can be verified using this information.

3.3 Shell script-based token retrieval

The following examples show how to programmatically retrieve the token through the URL using a Shell script:

1. Paste the token retrieval URL into the web browser's address bar: https://enlighten.enphaseenergy.com/entrez-auth-token?serial_num=<IQ Gateway_serial_number>
2. Replace **<IQ Gateway_serial_number>** in the above URL with the serial number of the specific IQ Gateway that you obtained. Refer to the section Get the system name and IQ Gateway serial number. .
3. The token is displayed as shown in the following image. Save the token securely for future use.

```
user='<UserName>'

password='<Password>'

envoy_serial='<Envoy_Serial_No>'

session_id=$(curl -X POST http://enlighten.enphaseenergy.com/login/login.json? -F "user[email]=$user" -F
"user[password]=$password" | jq -r ".session_id")

web_token=$(curl -X POST http://entrez.enphaseenergy.com/tokens -H "Content-Type: application/json" -d
"{\"session_id\": \"$session_id\", \"serial_num\": \"$envoy_serial\", \"username\": \"$user\"}")
```

NOTE: Replace the following items in the Shell script mentioned above:

- <Envoy_Serial_No> with the serial number of the specific IQ Gateway.
- <UserName> and <Password> with the system owner's credentials.

The variable web_token obtained in the last step of the script contains the access token.

3.4 Python script-based token retrieval

The following examples show how to programmatically retrieve the token through the URL using a Python script:

1. Paste the token retrieval URL into the web browser's address bar: https://enlighten.enphaseenergy.com/entrez-auth-token?serial_num=<IQ Gateway_serial_number>
2. Replace **<IQ Gateway_serial_number>** in the above URL with the serial number of the specific IQ Gateway that you obtained. Refer to the section Get the system name and IQ Gateway serial number. .
3. The token is displayed as shown in the following image. Save the token securely for future use.

```
import json

import requests

user='<UserName>'

password='<Password>'

envoy_serial='< Envoy_Serial_No>'

data = {'user[email]': user, 'user[password]': password}

response = requests.post('http://enlighten.enphaseenergy.com/login/login.json?', data=data) response_data = json.loads(response.text)

data = {'session_id': response_data['session_id'], 'serial_num': envoy_serial, 'username': user}

response = requests.post('http://entrez.enphaseenergy.com/tokens', json=data)

token_raw = response.text
```

NOTE: Replace the following items in the Python script mentioned above:

- <Envoy_Serial_No> with the serial number of the specific IQ Gateway.
- <UserName> and <Password> with the system owner's credentials.

The variable token_raw obtained in the last step of the script contains the access token.

4. Access the IQ Gateway using a token

Once you get the token, you can easily access the IQ Gateway local UI or the local APIs using this token.

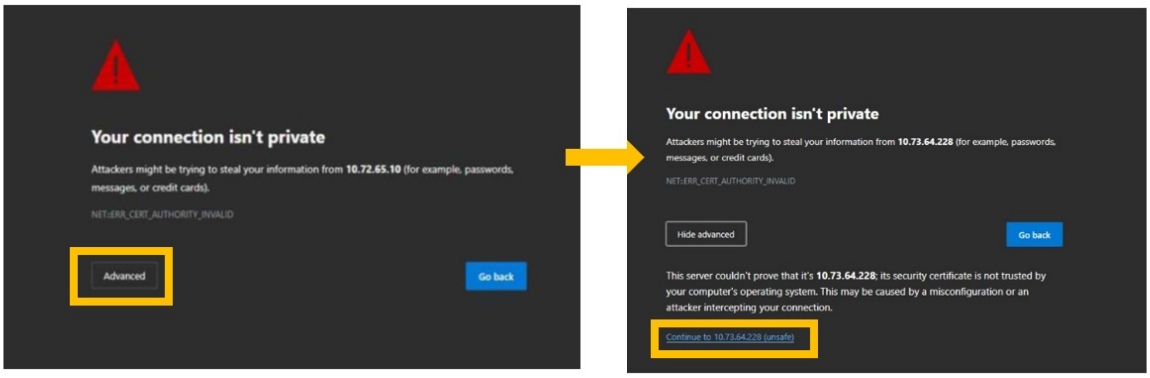
4.1 Access the IQ Gateway local UI

Follow the steps to access the IQ Gatewayvlocal UI:

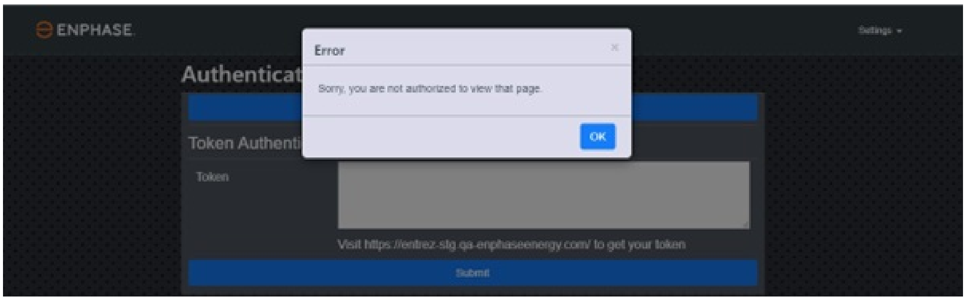
1. Open an internet browser (Internet Explorer, Mozilla Firefox, Google Chrome, or Safari) on a computer or mobile device connected to the same Local Area Network (LAN) as the IQ Gateway.
2. If the gateway
   - Has an LCD, then stop here. This is a legacy gateway and does not require token-based authentication to access local UI or local APIs.
   - Does not have an LCD. Enter https://envoy.local/ into the browser.

   If there are additional gateway units on the network, you can access them by entering the following strings in the browser window:

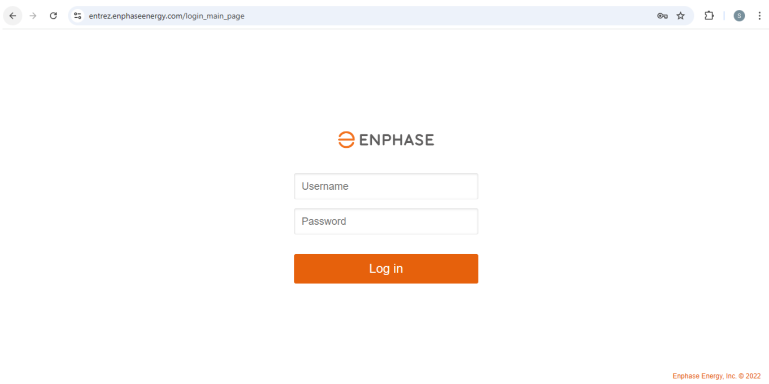
   - http://envoy-2.local
   - http://envoy-3.local, and so on.
3. IQ Gateway uses a self-signed certificate. Click **Advanced** > **Continue to** (IP address of IQ Gateway)**(unsafe)**. The browser is redirected to the IQ Gateway authentication page.



4. If there is any error, an error message pops up. Click **OK** to proceed.

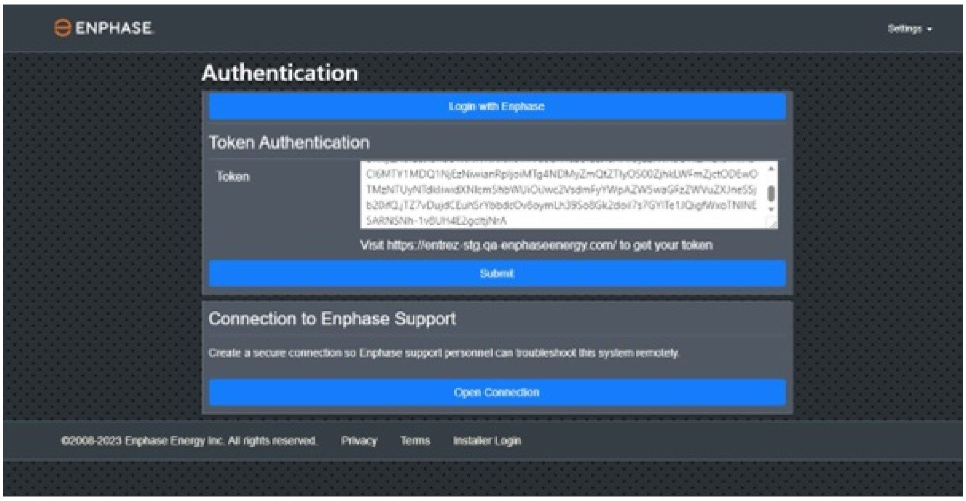


5. If you are online, click the **Login with Enphase** in the authentication screen, and enter the system owner's Enphase Cloud credentials.
   Authentication with Enphase Cloud happens automatically, and the browser is redirected to the IQ Gateway page. The token is not required in this case.



If the computer used to access the IQ Gateway is offline, proceed with step 6.

6. Paste the generated token in the **Token** field and click the **Submit** button. The browser is redirected to the IQ Gateway. The browser window displays the home screen after successfully connecting to the IQ Gateway.



4.2 Access the IQ Gateway APIs

The IQ Gateway APIs can be accessed through curl commands with the authorization bearer option and HTTPS commands. Follow the steps to access IQ GatewayAPIs:

1. Connect the computer trying to access the IQ Gateway's local APIs to the same LAN as the IQ Gateway.
2. Open the Command Prompt on the computer. Check that the router or network host is connected to the IQ Gateway by executing the ping command to the IQ Gateway. If the ping is successful, proceed with step 3. If not, check the network connectivity.
3. Enter the following curl command in the Command Prompt to access the IQ Gateway's available APIs. The format of the curl command is as follows:

   curl -f -k -H 'Accept: application/json' -H 'Authorization: Bearer <token code>' -X <API command>

   Where:

   - k: allow self-signed certificate

   - H: add http header with the token

   - X: used to pass method type for API command

   - f: fail on error (gives better output in case of unauthenticated redirection)

   - L: follow redirects (can use http and allow https redirection)

   Example: curl -f -k -H 'Accept: application/json' -H 'Authorization: Bearer eyJraWQiOi...' -X GET http://<IQ Gateway-ip>/api/v1/production/inverters

The token is valid only for one year to ensure safety, and you must generate a new token upon expiry. The following table lists a few local REST APIs of IQ Gateway.

API Name	Command	Description
System information	GET https://{IQ Gateway_ip}/info	Returns serial number, model id, and firmware

API Name	Command	Description
		version. This API is accessible without a token.
Provisioned device details	GET https://{IQ Gateway_ip}/ivp/ensemble/device_list	Returns the commissioning status of provisioned devices.
Meter details	GET https://{IQ Gateway_ip}/ivp/meters	Returns meter status, type of meter, and number of phase measurements.
Meter readings	GET https://{IQ Gateway_ip}/ivp/meters/readings	Returns measurements from Production CT, storage CT, and Consumption CT, and all are subjected to the availability of CTs.
Production meter data	GET https://{IQ Gateway_ip}/api/v1/production	Returns production energy and active power values for today, the last seven days, and lifetime in watt-hours. The API works even when the production meter is not installed and enabled at the site.
Energy data	GET https://{IQ Gateway_ip} /ivp/pdm/energy	Returns energy and active power values for microinverters, revenue grade meters, production and consumption meter for today, last seven days, and lifetime in watt-hours. The API works even when the production meter is not installed and enabled at the site.
Inverter production data	GET https://{IQ Gateway_ip}/api/v1/production/inverters	Returns maximum and last reported active power production information of the available microinverters.
Meter's live data	GET https://{IQ Gateway_ip}/ivp/livedata/status	Returns meter's live data with tasks and counters.
Power consumption data	GET https://{IQ Gateway_ip}/ivp/meters/reports/consumption	Returns power consumption information of the loads.

API Name	Command	Description
Grid readings	GET https://{IQ Gateway_ip}/ivp/meters/gridReading	Returns the voltage, current, frequency, active & reactive power at the point of grid connection.

Generate a new token using the methods outlined in the section Generate a token upon expiry. The Appendix section provides more description about these APIs and a sample JSON response obtained from the IQ Gateway when an HTTPS GET is done on any of these APIs. The commented lines (bold) are the most relevant parts including a brief explanation of the dedicated endpoint parameter.

Appendix A

Get system information

GET https://{IQ Gateway_ip}/info

Description: Returns serial number, model id, and firmware version.

Sample response:


```
<envoy_info>
<time>1658403712</time>
<device>
<sn>122125067699</sn> // Gateway serial number
<pn>800-00654-r06</pn> // Gateway part number
<software>D7.4.22</software> // Gateway eSW version number
<euaid>4c8675</euaid>
<seqnum>0</seqnum>
<apiver>1</apiver>
<imeter>true</imeter>
</device>
<web-tokens>true</web-tokens>
<package name="rootfs">
<pn>500-00001-r01</pn>
<version>02.00.00</version>
<build>1210</build>
</package>
<package name="kernel">
<pn>500-00011-r02</pn>
<version>04.04.225</version> // Kernel version number
<build>3eb4d3</build>
</package>
<package name="boot">
<pn>590-00019-r01</pn>
<version>02.00.01</version> // Bootcode version number
<build>1f421b</build>
</package>
<package name="app">
<pn>500-00002-r01</pn>
<version>07.04.22</version> // Gateway application version number
<build>d153a8</build>
```

```
</package>
<package name="devimg">
<pn>500-00005-r01</pn>
<version>01.02.398</version> // Gateway device image version number
<build>0bb4d1</build>
</package>
<package name="geo">
<pn>500-00008-r01</pn>
<version>02.01.24</version> // Gateway geo package version number
<build>a74d96</build>
</package>
<package name="backbone">
<pn>500-00010-r01</pn>
<version>07.00.20</version> // Gateway backbone package version
<build>176d57</build>
</package>
<package name="meter">
<pn>500-00013-r01</pn>
<version>03.02.08</version> // Gateway meter version
<build>1dd67a</build>
</package>
<package name="agf">
<pn>500-00012-r01</pn>
<version>02.02.00</version> // Gateway AGF version
<build>67d1d2</build>
</package>
<package name="security">
<pn>500-00016-r01</pn>
<version>02.00.00</version> // Gateway security version
<build>54a6dc</build>
</package>
<package name="essimg">
<pn>500-00020-r01</pn>
<version>22.04.52</version> // Gateway Ensemble software version
<build>996cf8</build>
</package>
<package name="full">
<pn>500-00001-r01</pn>
<version>02.00.00</version>
```

<build>1210</build>

</package>

<build_info>

<build_id>

ec2-user-envoy_uber-pkg_master:pkg-Jul-18-22-08:54:12 **// Software build date and time**

</build_id>

<build_time_gmt>1658134579</build_time_gmt>

<release_ver>02.00.2923</release_ver>

<release_stage>100-DEV</release_stage>

</build_info>

</envoy_info>

Appendix B

Get provisioned device details

GET https://{ IQ_Gateway_ip}/ivp/ensemble/device_list

Description: Returns the commissioning status of provisioned devices.

Sample response:

```
{
"usb": {
"ck2_bridge": "connected", "auto_scan": "false"
},
"devices": [
{
"serial_number": "492203008650", //Serial number of the device
"device_type": 13, // Device type 13 denotes battery,
"com_interface": 2,
"com_interface_str": "CAN", // Communication interface "status": "Connected", // Specified device type is connected
"dev_info": {
"capacity": 4960, //Capacity of the device (Units: Watt-hours)
"DER_Index": 1 // Denotes connected phase (1: Phase1,2: Phase 2, 3: Phase
3)
}
},
{
"serial_number": "122132000709",
"device_type": 0,
"com_interface": 2, "status": "Unknown"
}
]
}
```

NOTE: The enumerations associated with the device type are as follows: Unknown device = 0, Storage device = 13, and Microinverters = 14.

Appendix C

Get meter details

GET https://{ IQ_Gateway_ip}/ivp/meters

Description: Returns meter status, type of meter, and number of phase measurements.

Sample response:

{

"eid": 704643328, **// Production CT ID**

"state": "enabled", **// Enabled/Disabled production CT**

"measurementType": "production", **// Measuring CT type**

"phaseMode": "split, **// Number of phases the meter hardware can monitor** "phaseCount": 2, **// Number of phases meter is monitoring** "meteringStatus": "normal", **// Production metering status** "statusFlags": [ ]

},

{

"eid": 704643584, **// Consumption CT ID**

"state": "enabled", **// Enabled/Disabled Consumption CT**

"measurementType": "net-consumption", **// Measuring CT type**

"phaseMode": "split", **// Number of phases the meter hardware can monitor** "phaseCount": 2, **// Number of phases meter is monitoring** "meteringStatus": "normal", **// Consumption metering status** "statusFlags": [ ]

}

Appendix D

Get meter readings

GET https://{IQ_Gateway_ip}/ivp/meters/readings

Description: Returns measurements from Production CT, storage CT, and Consumption CT, and all are subject to the availability of CTs. This data will get updated once every five minutes.

Sample response:

[{eid: 704643328, **// Production CT ID**

timestamp: 1654218661, **// Epoch date and time of measured value**

actEnergyDlvd: 1608426.912, **// Overall active energy delivered**

actEnergyRcvd: 4.923, **// Overall active energy received**

apparentEnergy: 1648123.109, **// Overall apparent energy**

reactEnergyLagg: 52600.292, **// Overall lagging reactive energy**

reactEnergyLead: 19013.342, **// Overall leading reactive energy**

instantaneousDemand: 132.118, **// Instantaneous active power demand (Unit: Watts)**

activePower: 132.118, **// Instantaneous active power (Unit: Watts)**

apparentPower: 5328.778, **// Instantaneous apparent power (Unit: VA)** reactivePower: -5328.778, **// Instantaneous reactive power (Unit: VAr)**

pwrFactor: 0.025, **// Measured power factor at production CT location**

voltage: 246.377, **// Measured voltage at production CT location (Unit: Volts)**

current: 43.257, **// Measured current at production CT location (Unit: Ampere)**

freq: 59.188, **// Measured frequency at production CT location (Unit: Hertz)**

channels: [{

"eid": 1778385169, **// Production CT channel 1 (Phase L1) ID**

"timestamp": 1654218661, **// Epoch date and time of measured value in this**

channel

"actEnergyDlvd": 803639.138, **// Active energy delivered in this channel** "actEnergyRcvd": 2.650, **// Active energy received in this channel**

"apparentEnergy": 823442.481, **// Apparent energy in this channel** "reactEnergyLagg": 26264.291, **// Lagging reactive energy in this channel** "reactEnergyLead": 9545.452, **// Leading reactive energy in this channel** "instantaneousDemand": 66.037, **// Instantaneous active power demand in this channel (Unit: Watts)**

"activePower": 66.037, **// Instantaneous active power in this channel (Unit: Watts)**

"apparentPower": 2663.476, **// Instantaneous apparent power in this channel (Unit: VA)**

"reactivePower": -2663.476, **// Instantaneous reactive power in this channel (Unit: VAr)**

"pwrFactor": 0.025, **// Measured power factor at production CT location in this channel**

"voltage": 123.184, **// Measured voltage at production CT location in this channel (Unit: Volts)**

"current": 21.622, **// Measured current at production CT location in this channel (Unit: Ampere)**

"freq": 59.188 **// Measured frequency at production CT location in this channel (Unit: Hertz)**

},

{

eid: 1778385170, // Production CT channel 2 (Phase L2) ID

timestamp: 1654218661, // Epoch date and time of measured value in this channel

actEnergyDlvd: 804787.774, // Active energy delivered in this channel actEnergyRcvd: 2.273, // Active energy received in this channel

apparentEnergy: 824680.628, // Apparent energy in this channel

reactEnergyLagg: 26336.001, // Lagging reactive energy in this channel reactEnergyLead: 9467.890, // Leading reactive energy in this channel instantaneousDemand: 66.082, // Instantaneous active power demand in this channel (Unit: Watts)

activePower: 66.082, // Instantaneous active power in this channel (Unit: Watts)

apparentPower: 2665.302, // Instantaneous apparent power in this channel (Unit: VA)

reactivePower: -2665.302, // Instantaneous reactive power in this channel (Unit: VAr)

pwrFactor: 0.025, // Measured power factor at production CT location in this channel

voltage: 123.193, // Measured voltage at production CT location in this channel (Unit: Volts)

current: 21.635, // Measured current at production CT location in this channel (Unit: Ampere)

freq: 59.188 // Measured frequency at production CT location in this channel (Unit: Hertz)

},

{

eid": 1778385171, // Production CT channel 3 (Phase L3) ID

timestamp: 1654218661, // Epoch date and time of measured value in this channel

actEnergyDlvd: 0.000, // Active energy delivered in this channel actEnergyRcvd: 0.000, // Active energy received in this channel apparentEnergy: 0.000, // Apparent energy in this channel reactEnergyLagg: 0.000, // Lagging reactive energy in this channel reactEnergyLead: 0.000, // Leading reactive energy in this channel instantaneousDemand: 0.000, // Instantaneous active power demand in this channel (Unit: Watts)

activePower: 0.000, // Instantaneous active power in this channel (Unit: Watts)

apparentPower: 0.000, // Instantaneous apparent power in this channel (Unit: VA)

reactivePower: 0.000, // Instantaneous reactive power in this channel (Unit: VAr)

pwrFactor: 0.000, // Measured power factor at production CT location in this channel

voltage: 0.000, // Measured voltage at production CT location in this channel (Unit: Volts)

current: 0.000, // Measured current at production CT location in this channel (Unit: Ampere)

freq: 59.188 // Measured frequency at production CT location in this channel (Unit: Hertz)

}

]

},

{

eid: 704643584, // Storage CT ID

timestamp: 1654218661, // Epoch date and time of measured value

actEnergyDlvd: 48540.732, // Overall active energy delivered

actEnergyRcvd: 1244797.861, // Overall active energy received

apparentEnergy: 1332629.594, // Overall apparent energy

reactEnergyLagg: 13955.857, // Overall lagging reactive energy

reactEnergyLead: 30823.381, **// Overall leading reactive energy**

instantaneousDemand: -0.000, **// Instantaneous active power demand (Unit: Watts)**

activePower: -0.000, **// Instantaneous active power (Unit: Watts)**

apparentPower: 34.831, **// Instantaneous apparent power (Unit: VA)**

reactivePower: -0.000, **// Instantaneous reactive power (Unit: VAr)**

pwrFactor: 0.000, **// Measured power factor at production CT location**

voltage: 246.338, **// Measured voltage at production CT location (Unit: Volts)**

current: 0.283, **// Measured current at production CT location (Unit: Ampere)**

freq: 59.188, **// Measured frequency at production CT location (Unit: Hertz)**

channels: [{

eid: 1778385425, **// Storage CT channel 1 (Phase L1) ID**

timestamp: 1654218661, **// Epoch date and time of measured value in this channel**

actEnergyDlvd: 24176.961, **// Active energy delivered in this channel**

actEnergyRcvd: 600344.235, **// Active energy received in this channel** apparentEnergy: 644044.993, **// Apparent energy in this channel**

reactEnergyLagg: 5391.081, **// Lagging reactive energy in this channel** reactEnergyLead: 15459.001, **// Leading reactive energy in this channel** instantaneousDemand: -0.000, **// Instantaneous active power demand in this channel (Unit: Watts)**

activePower: -0.000, **// Instantaneous active power in this channel (Unit: Watts)**

apparentPower: 16.858, **// Instantaneous apparent power in this channel (Unit: VA)**

reactivePower: -0.000, **// Instantaneous reactive power in this channel (Unit: VAr)**

pwrFactor: 0.000, **// Measured power factor at production CT location in this channel**

voltage: 123.152, **// Measured voltage at production CT location in this channel (Unit: Volts)**

current: 0.137, **// Measured current at production CT location in this channel (Unit: Ampere)**

freq: 59.188 **// Measured frequency at production CT location in this channel (Unit: Hertz)**

},

{

eid: 1778385426, **// Storage CT channel 2 (Phase L2) ID**

timestamp: 1654218661, **// Epoch date and time of measured value in this channel**

actEnergyDlvd: 24363.771, **// Active energy delivered in this channel** actEnergyRcvd: 644453.626, **// Active energy received in this channel** apparentEnergy: 688584.601, **// Apparent energy in this channel**

reactEnergyLagg: 8564.776, **// Lagging reactive energy in this channel** reactEnergyLead: 15364.380, **// Leading reactive energy in this channel** instantaneousDemand: -0.000, **// Instantaneous active power demand in this channel (Unit: Watts)**

activePower: -0.000, **// Instantaneous active power in this channel (Unit: Watts)**

apparentPower: 17.973, **// Instantaneous apparent power in this channel (Unit: VA)**

reactivePower: -0.000, **// Instantaneous reactive power in this channel (Unit: VAr)**

pwrFactor: 0.000, **// Measured power factor at production CT location in this channel**

voltage: 123.186, **// Measured voltage at production CT location in this channel (Unit: Volts)**

current: 0.146, **// Measured current at production CT location in this channel (Unit: Ampere)**

freq: 59.188 **// Measured frequency at production CT location in this channel (Unit: Hertz)**

},

{

eid: 1778385427, // Storage CT channel 3 (Phase L3) ID

timestamp: 1654218661, // Epoch date and time of measured value in this channel

actEnergyDlvd: 129399.711, // Active energy delivered in this channel actEnergyRcvd: 93791.210, // Active energy received in this channel apparentEnergy: 242548.385, // Apparent energy in this channel

reactEnergyLagg: 15196.459, // Lagging reactive energy in this channel reactEnergyLead: 10272.271, // Leading reactive energy in this channel instantaneousDemand: 0.000, // Instantaneous active power demand in this channel (Unit: Watts)

activePower: 0.000, // Instantaneous active power in this channel (Unit: Watts)

apparentPower: 2697.761, // Instantaneous apparent power in this channel (Unit: VA)

reactivePower: 2697.761, // Instantaneous reactive power in this channel (Unit: VAr)

pwrFactor: 0.000, // Measured power factor at production CT location in this channel

voltage: 123.175, // Measured voltage at production CT location in this channel (Unit: Volts)

current: 21.902, // Measured current at production CT location in this channel (Unit: Ampere)

freq: 59.188 // Measured frequency at production CT location in this channel (Unit: Hertz)

}

]

},

{

eid: 704643840, // Consumption CT ID

timestamp: 1654218661, // Epoch date and time of measured value

actEnergyDlvd: 258799.422, // Overall active energy delivered

actEnergyRcvd: 187582.421, // Overall active energy received

apparentEnergy: 485096.770, // Overall apparent energy

reactEnergyLagg: 30392.918, // Overall lagging reactive energy

reactEnergyLead: 20544.543, // Overall leading reactive energy

instantaneousDemand: 0.000, // Instantaneous active power demand (Unit: Watts)

activePower: 0.000, // Instantaneous active power (Unit: Watts)

apparentPower: 5395.521, // Instantaneous apparent power (Unit: VA) reactivePower: 5395.521, // Instantaneous reactive power (Unit: VAr)

pwrFactor: 0.000, // Measured power factor at production CT location

voltage: 246.351, // Measured voltage at production CT location (Unit: Volts)

current: 43.804, // Measured current at production CT location (Unit: Ampere)

freq: 59.188, // Measured frequency at production CT location (Unit: Hertz)

channels: [{

eid: 1778385681, // Consumption CT channel 1 (Phase L1) ID

timestamp: 1654218661, // Epoch date and time of measured value in this channel

actEnergyDlvd: 129399.711, // Active energy delivered in this channel actEnergyRcvd: 93791.210, // Active energy received in this channel apparentEnergy: 242548.385, // Apparent energy in this channel

reactEnergyLagg: 15196.459, // Lagging reactive energy in this channel reactEnergyLead: 10272.271, // Leading reactive energy in this channel instantaneousDemand: 0.000, // Instantaneous active power demand in this channel (Unit: Watts)

activePower: 0.000, // Instantaneous active power in this channel (Unit: Watts)

apparentPower: 2697.761, // Instantaneous apparent power in this channel (Unit: VA) reactivePower: 2697.761, // Instantaneous reactive power in this channel (Unit: VAr)

pwrFactor: 0.000, // Measured power factor at production CT location in this channel

voltage: 123.175, // Measured voltage at production CT location in this channel (Unit: Volts)

current: 21.902, // Measured current at production CT location in this channel (Unit: Ampere)

freq: 59.188 // Measured frequency at production CT location in this channel (Unit: Hertz)

},

{

eid: 1778385682, // Consumption CT channel 2 (Phase L2) ID

timestamp: 1654218661, // Epoch date and time of measured value in this channel

actEnergyDlvd: 129399.711, // Active energy delivered in this channel actEnergyRcvd: 93791.210, // Active energy received in this channel apparentEnergy: 242548.385, // Apparent energy in this channel

reactEnergyLagg: 15196.459, // Lagging reactive energy in this channel reactEnergyLead: 10272.271, // Leading reactive energy in this channel instantaneousDemand: 0.000, // Instantaneous active power demand in this channel (Unit: Watts)

activePower: 0.000, // Instantaneous active power in this channel (Unit: Watts)

apparentPower: 2697.761, // Instantaneous apparent power in this channel (Unit: VA)

reactivePower: 2697.761, // Instantaneous reactive power in this channel (Unit: VAr)

pwrFactor: 0.000, // Measured power factor at production CT location in this channel

voltage: 123.175, // Measured voltage at production CT location in this channel (Unit: Volts)

current: 21.902, // Measured current at production CT location in this channel (Unit: Ampere)

freq: 59.188 // Measured frequency at production CT location in this channel (Unit: Hertz)

},

{

eid: 1778385683, // Consumption CT channel 3 (Phase L3) ID

timestamp: 1654218661, // Epoch date and time of measured value in this channel

actEnergyDlvd: 0.000, // Active energy delivered in this channel

actEnergyRcvd: 0.000, // Active energy received in this channel

apparentEnergy: 0.000, // Apparent energy in this channel

reactEnergyLagg: 0.000, // Lagging reactive energy in this channel

reactEnergyLead: 0.000, // Leading reactive energy in this channel instantaneousDemand: 0.000, // Instantaneous active power demand in this channel (Unit: Watts)

activePower: 0.000, // Instantaneous active power in this channel (Unit: Watts)

apparentPower: 0.000, // Instantaneous apparent power in this channel (Unit: VA)

reactivePower: 0.000, // Instantaneous reactive power in this channel (Unit: VAr)

pwrFactor: 0.000, // Measured power factor at production CT location in this channel

voltage: 0.000, // Measured voltage at production CT location in this channel (Unit: Volts)

current: 0.000, // Measured current at production CT location in this channel (Unit: Ampere)

freq: 59.188 // Measured frequency at production CT location in this channel (Unit: Hertz)

}

]

}

]

Appendix E

Get production meter data

GET https://{IQ Gateway_ip}/api/v1/production

Description: Returns watt-hour energy values in three formats, today, seven days, and lifetime. Also includes present active power. This API will work only when the production meter is installed and enabled at the site.

Sample response:

```
{
"wattHoursToday": 21674, // Total energy seen today (Units: Watt-hours) "wattHoursSevenDays": 719543, // Total energy seen in the last 7 days (Units: Watt-hours)
"wattHoursLifetime": 1608587, // Total energy seen in the lifetime (Units: Watt-hours)
"wattsNow": 227 // Instantaneous active power (Unit: Watts)
}
```

Appendix F

Get energy data

GET https://{IQ Gateway_ip}/ivp/pdm/energy

Description: Returns watt-hour energy values for microinverters, revenue grade meters, production, and consumption meter in 3 formats, today, seven days, and lifetime. Also includes present active power. This API will work even when the production meter is not installed and enabled at the site.

Sample response:

{

"production": {

"pcu": { // Measurements from microinverters

"wattHoursToday": 13251, // Total energy seen today (Units: Watt-hours) "wattHoursSevenDays": 91306, // Total energy seen in the last 7 days (Units: Watt-hours)

"wattHoursLifetime": 8250671, // Total energy seen in the lifetime (Units: Watt-hours)

"wattsNow": 596 // Instantaneous active power (Unit: Watts)

},

"rgm": { // Not applicable

"wattHoursToday": 0, // Not applicable

"wattHoursSevenDays": 0, // Not applicable

"wattHoursLifetime": 0, // Not applicable

"wattsNow": 0 // Not applicable

},

"eim": { // Measurements from Envoy Internal production meter (eim)

"wattHoursToday": 0, // Total energy seen today (Units: Watt-hours)

"wattHoursSevenDays": 0, // Total energy seen in the last 7 days (Units: Watt-hours)

"wattHoursLifetime": 0, // Total energy seen in the lifetime (Units: Watt- hours)

"wattsNow": 0 // Instantaneous active power (Unit: Watts)

}

},

"consumption": {

"eim": { // Measurements from Envoy Internal consumption meter (eim) "wattHoursToday": 0, // Total energy seen today (Units: Watt-hours) "wattHoursSevenDays": 0, // Total energy seen in the last 7 days (Units: Watt-hours)

"wattHoursLifetime": 0, // Total energy seen in the lifetime (Units: Watt- hours)

"wattsNow": 0 // Instantaneous active power (Unit: Watts)

}

}

Appendix G

Get inverter production data

GET https://{IQ_Gateway_ip}/api/v1/production/inverters

Description: Returns the available microinverters' maximum and last reported active power production information. This data will get updated once every five minutes.

Sample response:

```
[
{
"serialNumber": "121935144671", // Serial number of this microinverter
"lastReportDate": 1654171836, // Last reported time in epoch date and time format of this microinverter
"devType": 1, // Device type
"lastReportWatts": 15, // Last reported active power (Unit: Watts)
"maxReportWatts": 38 // Highest ever reported power (Unit: Watts)
}
]
```

Appendix H

Get meter’s live data

GET https://{IQ_Gateway_ip}/ivp/livedata/status

Description: Returns meter’s live data with tasks and counters.

Sample response:


```
{
"connection": {
"mqtt_state": "connected", // MQTT broker status
"prov_state": "configured", // Provisioning status
"auth_state": "ok", // Authentication status
"sc_stream": "enabled", // Status of live data being written to MQTT stream
"sc_debug": "enabled”, // Status of data being written to debug stream
},
"meters": {
"last_update": 1654221647, // Epoch timestamp of when data was last collected from the meters
"soc": 100, // State of Charge of the storage device
"main_relay_state": 0, // State of the main relay
"gen_relay_state": 5, // State of the generator relay
"backup_bat_mode": 1, // Mode of back up battery
"backup_soc": 10, // State of the backup battery
"is_split_phase": 1, // Conformance of split-phase grid type
"phase_count": 0, // Number of connected phases
"enc_agg_soc": 100, // Aggregate SoC of the Encharge
"enc_agg_energy": 24800, , // Aggregate Energy of the Encharge
"acb_agg_soc": 0, // Aggregate SoC of the AC battery
"acb_agg_energy": 0, , // Aggregate SoC of the AC battery
"pv": { // Photovoltaic meter result
"agg_p_mw": 329549, // Aggregate active power (Unit: milliwatts)
"agg_s_mva": 329549, // Aggregate apparent power (Unit: milli volt- amperes)
"agg_p_ph_a_mw": 329549, // Aggregate active power in phase A (Unit: milliwatts)
"agg_p_ph_b_mw": 0, // Aggregate active power in phase B (Unit: milliwatts)
"agg_p_ph_c_mw": 0, // Aggregate active power in phase C (Unit: milliwatts)
"agg_s_ph_a_mva": 329549, // Aggregate apparent power in phase A (Unit: milli volt-amperes)
"agg_s_ph_b_mva": 0, // Aggregate apparent power in phase B (Unit: milli volt-amperes)
"agg_s_ph_c_mva": 0 // Aggregate apparent power in phase C (Unit: milli volt-amperes)
},
```

"storage": { // Battery storge meter result

"agg_p_mw": -220800, // Aggregate active power (Unit: milliwatts)

"agg_s_mva": -559446, // Aggregate apparent power (Unit: milli volt- amperes)

"agg_p_ph_a_mw": -220800, // Aggregate active power in phase A (Unit: milliwatts)

"agg_p_ph_b_mw": 0, // Aggregate active power in phase B (Unit: milliwatts)

"agg_p_ph_c_mw": 0, // Aggregate active power in phase C (Unit: milliwatts)

"agg_s_ph_a_mva": -559446, // Aggregate apparent power in phase A (Unit: milli volt-amperes)

"agg_s_ph_b_mva": 0, // Aggregate apparent power in phase B (Unit: milli volt-amperes)

"agg_s_ph_c_mva": 0 // Aggregate apparent power in phase C (Unit: milli volt-amperes)

},

"grid": { // Grid meter result

"agg_p_mw": 0, // Aggregate active power (Unit: milliwatts)

"agg_s_mva": 0, // Aggregate apparent power (Unit: milli volt-amperes) "agg_p_ph_a_mw": 0, // Aggregate active power in phase A (Unit: milliwatts)

"agg_p_ph_b_mw": 0, // Aggregate active power in phase B (Unit: milliwatts)

"agg_p_ph_c_mw": 0, // Aggregate active power in phase C (Unit: milliwatts)

"agg_s_ph_a_mva": 0, // Aggregate apparent power in phase A (Unit: milli volt-amperes)

"agg_s_ph_b_mva": 0, // Aggregate apparent power in phase B (Unit: milli volt-amperes)

"agg_s_ph_c_mva": 0 // Aggregate apparent power in phase C (Unit: milli volt-amperes)

},

"load": { // Load meter result

"agg_p_mw": 108749, // Aggregate active power (Unit: milliwatts)

"agg_s_mva": -229897, // Aggregate apparent power (Unit: milli volt- amperes)

"agg_p_ph_a_mw": 108749, // Aggregate active power in phase A (Unit: milliwatts)

"agg_p_ph_b_mw": 0, // Aggregate active power in phase B (Unit: milliwatts)

"agg_p_ph_c_mw": 0, // Aggregate active power in phase C (Unit: milliwatts)

"agg_s_ph_a_mva": -229897, // Aggregate apparent power in phase A (Unit: milli volt-amperes)

"agg_s_ph_b_mva": 0, // Aggregate apparent power in phase B (Unit: milli volt-amperes)

"agg_s_ph_c_mva": 0 // Aggregate apparent power in phase C (Unit: milli volt-amperes)

},

"generator": { // Generator meter result

"agg_p_mw": 0, // Aggregate active power (Unit: milliwatts)

"agg_s_mva": 0, // Aggregate apparent power (Unit: milli volt-amperes) "agg_p_ph_a_mw": 0, // Aggregate active power in phase A (Unit:

milliwatts)

"agg_p_ph_b_mw": 0, // Aggregate active power in phase B (Unit: milliwatts)

"agg_p_ph_c_mw": 0, // Aggregate active power in phase C (Unit: milliwatts)

"agg_s_ph_a_mva": 0, // Aggregate apparent power in phase A (Unit: milli volt-amperes)

"agg_s_ph_b_mva": 0, // Aggregate apparent power in phase B (Unit: milli volt-amperes)

"agg_s_ph_c_mva": 0 // Aggregate apparent power in phase C (Unit: milli volt-amperes)

}
},
"tasks": {
"task_id": 27672012, // ID number of the most recent task processed
"timestamp": 1654219883 // Unix epoch timestamp of the last task processed
},
"counters": {
"main_CfgLoad": 1, // Number of times the configuration has been loaded
"main_CfgChanged": 1, // Number of times the configuration has been changed
"main_taskUpdate": 62, // Number of times a task has updated
"MqttClient_publish": 10260, // Number of times the MQTT client has published a message
"MqttClient_live_debug": 190, // Number of times the MQTT client has published a debug message
"MqttClient_respond": 260, // Number of times the MQTT client has responded to a request
"MqttClient_msgarrvd": 130, // Number of messages that have been delivered to the MQTT client
"MqttClient_create": 13, // Number of times an instance of the MQTT client has been created
"MqttClient_setCallbacks": 13, // number of times the callback has been set on the MQTT to process messages
"MqttClient_connect": 13, // Number of times a connection has been made with the MQTT client to the broker
"MqttClient_connect_err": 5, // Number of times the MQTT client has failed to connect
"MqttClient_connect_Err": 5, // Duplicate of MqttClient_connect_err
"MqttClient_subscribe": 8, // Number of times the MQTT client has subscribed to a topic
"SSL_Keys_Create": 13, // Number of Secure Socket Layer (SSL) or Transport Layer Security (TLS) keys that have been created
"sc_hdlDataPub": 9440, // Number of times the hardware description language (HDL) has been published
"sc_SendStreamCtrl": 72, // Number of times the send stream functionality has been controlled
"sc_SendDemandRspCtrl": 65517, // Number of times the send demand response stream functionality has been controlled
"rest_Meters": 7,
"rest_Status": 579 // Number of API REST calls the Gateway has handled
}
}

Appendix I

Get power consumption data

GET https://{IQ_Gateway_ip}/ivp/meters/reports/consumption

Description: Returns power consumption information of the loads. This data will get updated once every five minutes.

Sample response:

{

"createdAt": 1654625079, // Epoch date and time of measurement

"reportType": "net-consumption", // Type of reported measurement *

"cumulative": { // Cumulative results of all phases

"currW": 119.423, // Instantaneous active power (Unit: Watts)

"actPower": 119.423, // Active power (Unit: Watts)

"apprntPwr": 105.678, // Apparent power (Unit: VA)

"reactPwr": -261.046, // Reactive power (Unit: VAr)

"whDlvdCum": 43110.122, // Cumulative active energy delivered (Unit: Wh) "whRcvdCum": 0.000, // Cumulative active energy received (Unit: Wh) "varhLagCum": -25071.856, // Cumulative lagging reactive energy (Unit:

VArh)

"varhLeadCum": 35895.778, // Cumulative leading reactive energy (Unit: VArh)

"vahCum": 192725.807, // Cumulative apparent energy (Unit: VAh)

"rmsVoltage": 241.427, // Instantaneous RMS voltage (Unit: Volt)

"rmsCurrent": 0.875, // Instantaneous RMS Current (Unit: Ampere)

"pwrFactor": 1.00, // Measured power factor

"freqHz": 60.00 // Measured frequency (Unit: Hz)

},

"lines": [// Phase L1

{

"currW": 56.672, // Instantaneous active power (Unit: Watts)

"actPower": 56.672, // Active power (Unit: Watts)

"apprntPwr": 49.248, // Apparent power (Unit: VA)

"reactPwr": -136.579, , // Reactive power (Unit: VAr)

"whDlvdCum": 21051.342, // Cumulative active energy delivered (Unit: Wh) "whRcvdCum": 0.000, // Cumulative active energy received (Unit: Wh) "varhLagCum": -12541.347, // Cumulative lagging reactive energy (Unit:

VArh)

"varhLeadCum": 18473.849, // Cumulative leading reactive energy (Unit: VArh)

"vahCum": 96511.746, // Cumulative apparent energy (Unit: VAh)

"rmsVoltage": 120.673, // Instantaneous RMS voltage (Unit: Volt)

"rmsCurrent": 0.408, // Instantaneous RMS Current (Unit: Ampere) "pwrFactor": 1.00, // Measured power factor

"freqHz": 60.00 // Measured frequency (Unit: Hz)

},

{ // Phase L2

"currW": 62.751, // Instantaneous active power (Unit: Watts)

"actPower": 62.751, // Active power (Unit: Watts)

"apprntPwr": 56.430, // Apparent power (Unit: VA)

"reactPwr": -124.467, // Reactive power (Unit: VAr)

"whDlvdCum": 22058.779, // Cumulative active energy delivered (Unit: Wh) "whRcvdCum": 0.000, // Cumulative active energy received (Unit: Wh) "varhLagCum": -12530.509, // Cumulative lagging reactive energy (Unit:

VArh)

"varhLeadCum": 17421.929, // Cumulative leading reactive energy (Unit: VArh)

"vahCum": 96214.061, // Cumulative apparent energy (Unit: VAh)

"rmsVoltage": 120.753, // Instantaneous RMS voltage (Unit: Volt)

"rmsCurrent": 0.467, // Instantaneous RMS Current (Unit: Ampere)

pwrFactor": 1.00, // Measured power factor

"freqHz": 60.00 // Measured frequency (Unit: Hz)

}

]

},

}

NOTE: Net consumption denotes reporting of the load along with solar production. Total consumption denotes reporting of the load only, excluding solar production.

Appendix J

Get grid readings

GET https://{IQ_Gateway_ip}/ivp/meters/gridReading

Description: Returns the voltage, current, frequency, active & reactive power at the point of grid connection.

Sample response:


```
[
{
"channels": [
{
"phase": "L1", // Phase L1 readings
"activePower": -658.588, // Instantaneous active power in phase L1 channel (Unit: Watts)
"reactivePower": -60.154, // Instantaneous reactive power in phase L1 channel (Unit: VAr)
"voltage": 229.632, // Measured voltage in phase L1 channel (Unit: Volt)
"current": 2.885, // Measured current in phase L1 channel (Unit: Ampere)
"freq": 50.000 // Measured frequency in phase L1 channel (Unit: Hertz)
},
{
"phase": "L2", // Phase L2 readings
"activePower": -676.405, // Instantaneous active power in phase L2 channel (Unit: Watts)
"reactivePower": -67.361, // Instantaneous reactive power in phase L2 channel (Unit: VAr)
"voltage": 230.234, // Measured voltage in phase L2 channel (Unit: Volt)
"current": 2.939, // Measured current in phase L2 channel (Unit: Ampere)
"freq": 50.000 // Measured frequency in phase L2 channel (Unit: Hertz)
},
{
"phase": "L3", // Phase L3 readings
"activePower": -687.344, // Instantaneous active power in phase L3 channel (Unit: Watts)
"reactivePower": -62.442, // Instantaneous reactive power in phase L3 channel (Unit: VAr)
"voltage": 230.654, // Measured voltage in phase L3 channel (Unit: Volt)
"current": 2.996, // Measured current in phase L3 channel (Unit: Ampere)
"freq": 50.000 // Measured frequency in phase L3 channel (Unit: Hertz)
}
]
}
]
```

NOTE: Negative values indicate that active power is being exported (feed-in). Positive refers to import (grid-supply).

15. Revision history

Revision	Date	Description
TEB-00060-2.0	June 2025	• Added relevant comments for the API endpoint parameters sections. • Added API response for system information, provisioned devices, and grid reading.
TEB-00060-1.0	August 2023	Added production meter data and production data sections.

Previous releases