

---

## **Accessing IQ Gateway local APIs and local UI with token-based authentication**

### Contents

<b>Overview .....</b>	<b>2</b>
<b>Get the system name and IQ Gateway serial number .....</b>	<b>2</b>
<b>Generate a token .....</b>	<b>2</b>
<b>Get a token through web UI.....</b>	<b>3</b>
<b>Get a token programmatically through GET on a URL .....</b>	<b>5</b>
<b>Shell script-based token retrieval.....</b>	<b>5</b>
<b>Python script-based token retrieval .....</b>	<b>6</b>
<b>Access the IQ Gateway using a token .....</b>	<b>7</b>
<b>Access the IQ Gateway local UI .....</b>	<b>7</b>
<b>Access the IQ Gateway APIs .....</b>	<b>9</b>
<b>Appendix .....</b>	<b>12</b>
<b>Get meter details.....</b>	<b>12</b>
<b>Get meter readings .....</b>	<b>12</b>
<b>Get production meter data.....</b>	<b>18</b>
<b>Get production data .....</b>	<b>18</b>
<b>Get inverter production data.....</b>	<b>19</b>
<b>Get meter's live data.....</b>	<b>19</b>
<b>Get power consumption data .....</b>	<b>22</b>
<b>Revision history .....</b>	<b>26</b>

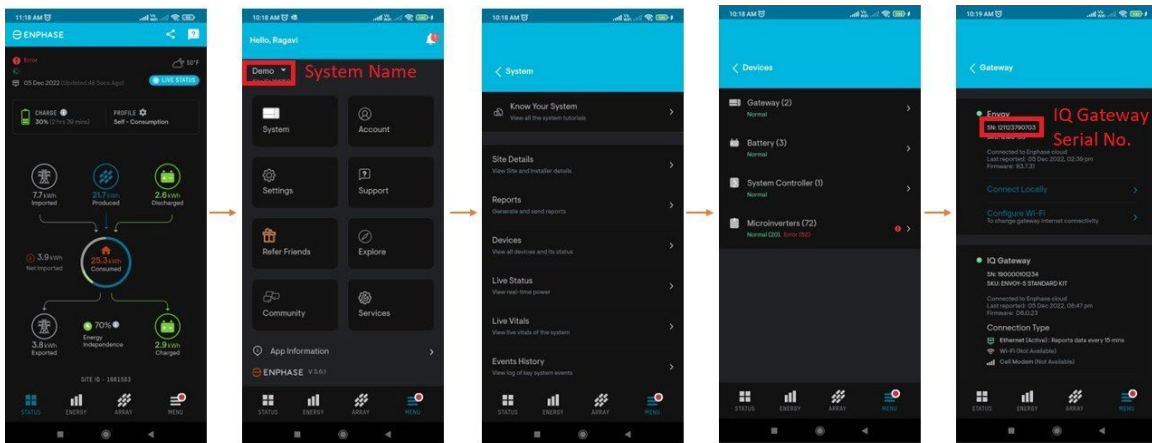
## Overview

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:

- How to get the system name and IQ Gateway serial number
- How to get a token for your IQ Gateway
- How to access IQ Gateway local UI and APIs using the token

## 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.

## Generate a token

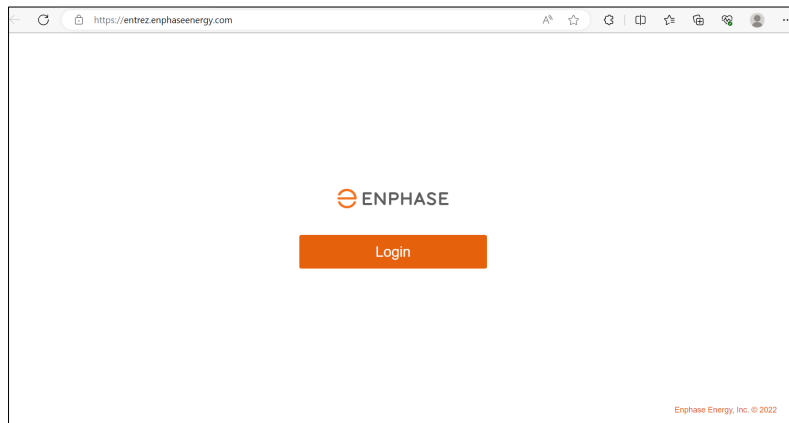
This section describes different methods to generate a unique token to access IQ Gateway local interfaces.

- Get a token through web UI
- Get a token programmatically through GET on a URL
- Retrieve the token through a Shell script
- Retrieve the token through the Python script

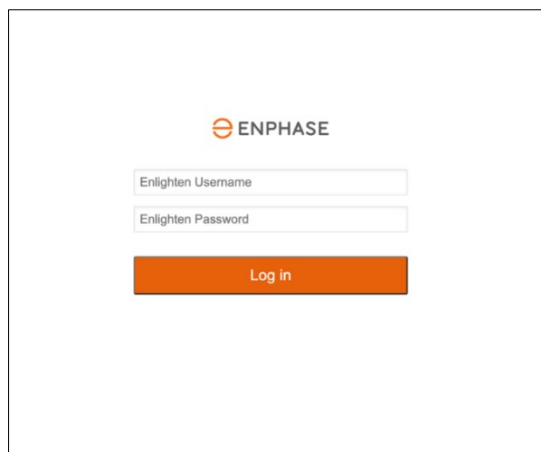
## Get a token through web UI

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 App 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.

**Authentication**

Create access token

For commissioned gateway  
 For uncommissioned gateways

Select System \*

Select Gateway \*

Create access token

Sign Out



**NOTE:** If your Enphase Cloud login is associated with multiple systems or if the system has multiple IQ Gateways, then select the system and the serial number of the IQ Gateway, which requires access through local API in the **Select System** and **Select Gateway** dropdowns, respectively.

- 5. Click the **Create access token** to generate a token.
- 6. In the **Authentication** window, click the **Copy and close** button to copy the token.

< Back

**Authentication**

Access token

```
eyJrZWeW10IiZSU011ZmhNS0xRTRlLTQ0R0JAcWYkN00zZWV5YTYyYTY1Y2N1L1JkVjE1L01L1JW10L01L01JF0zI  
1SLj9_e9zD8M1011Zm5jW8W0N0P7a80M1L1C1p1M6101JF0zY5X01L1C11m0M0K01VW01ci1iK1mV0c0hc01L1C11e  
HAL0J2MzA0NzI2NTIeIm1hdC16MTYzNDAlODI01xM1v1anRp1jo10Dc2M005M3YtYWYzN1000Dc4LW13Z7ctZDhhNz11  
0DJhNTd111wi4XN1cm5hbMU101JrYkZpaZFaYWSK0GvUcGhcZV1bmVY23kuY29tIn0_K0aSYq91M0iax_1Roq6zrk  
oB7R1d6G29E9_NguR9E2B7-McW0X1RUodsHx0emcDustv9y1MCVpdT1w0MFX0F9
```

Copy and close

Warning:  
The access token will only be displayed once. It will not be shared and cannot be retrieved. Please be sure to save it now.

- 7. Once copied, paste the token,
  - In the home automation setup where access to the IQ Gateway local APIs is required, or
  - Into the browser, 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.

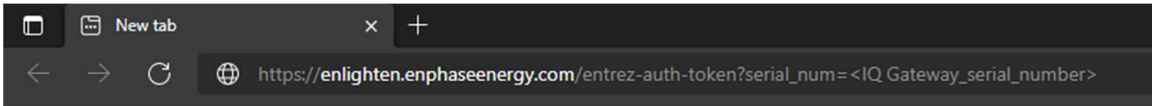
If the installer and system owner credentials are used as a self-installer, then the web UI-based token retrieval outlined in this section will result in a token valid for 12 hours. The owner can contact Enphase Support to change the credentials from the installer credentials to the system owner credentials.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

## Get a token programmatically through GET on a URL

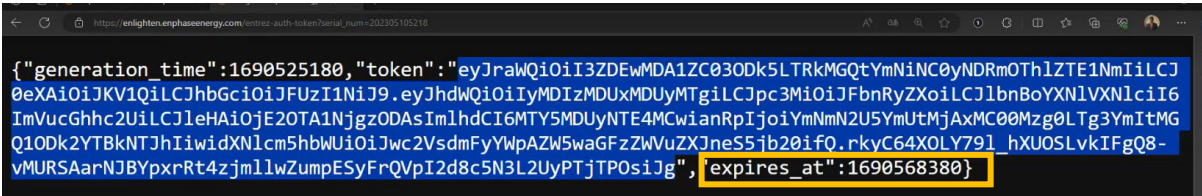
Follow the procedure to retrieve a token programmatically. Enphase provides a URL to retrieve a token and mentions the duration for which the token is valid.

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>](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 HTTP 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 timestamp format. The duration for which the token is valid can be verified using this information.

## Shell script-based token retrieval

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

```
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.

## Python script-based token retrieval

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

```
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.

## 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.

### Access the IQ Gateway local UI

Follow the steps to access the IQ Gateway local UI:

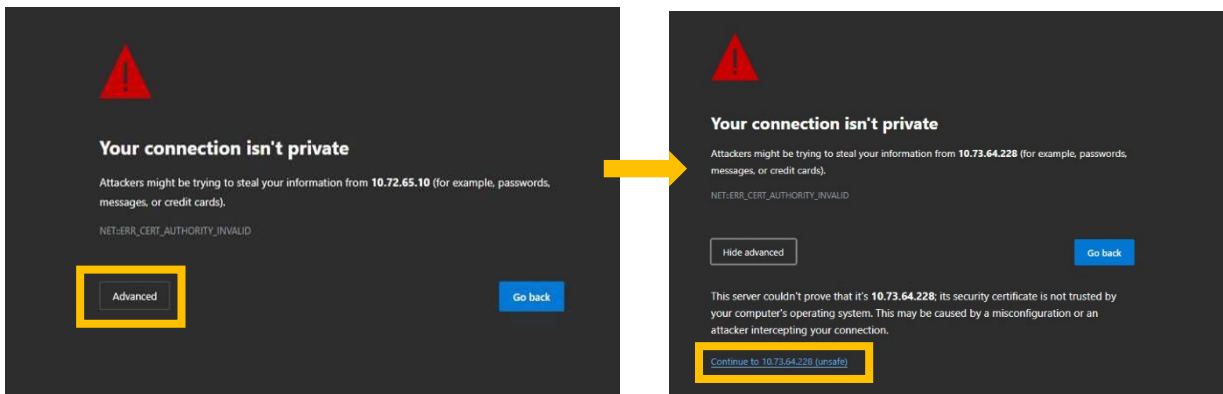
1. Open an internet browser (Internet Explorer, Mozilla Firefox, Google Chrome, or Safari) on a computer or mobile device connected to the same Local Area Network (LAN) as the IQ Gateway.
2. If the gateway
  - Has an LCD, then stop here. This is a legacy gateway and does not require token-based authentication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

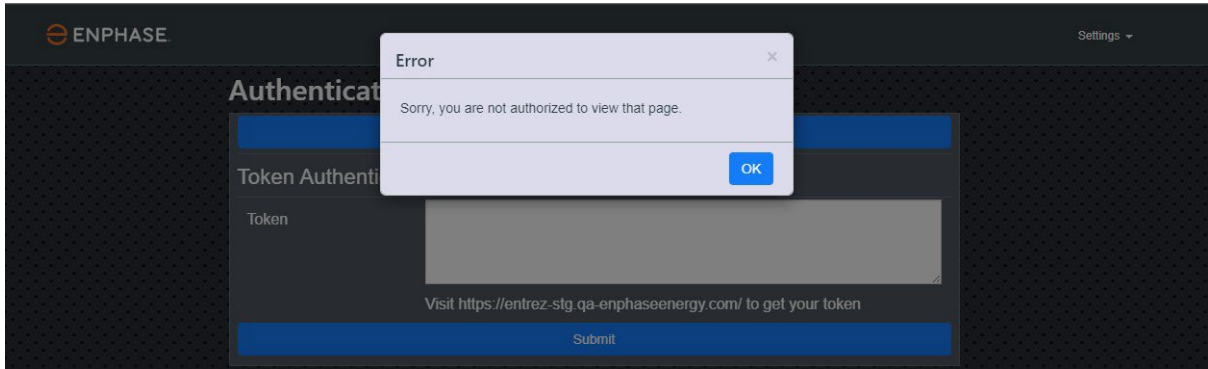
- `http://envoy-2.local`
- `http://envoy-3.local`, and so on.

3. IQ Gateway uses a self-signed certificate. Click **Advanced** > **Continue to** (IP address of IQ Gateway)(**unsafe**).

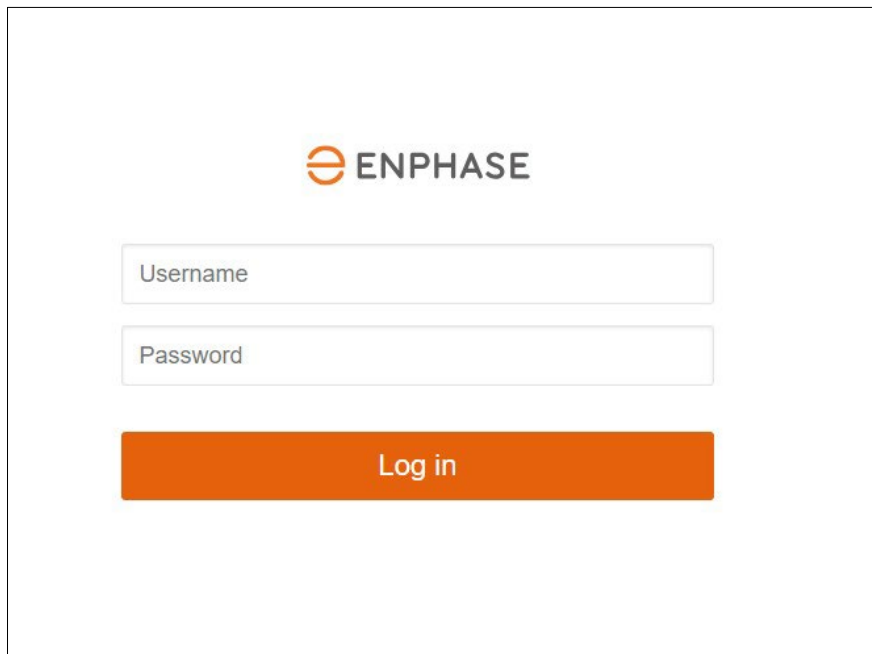
The browser is redirected to the IQ Gateway authentication page.



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



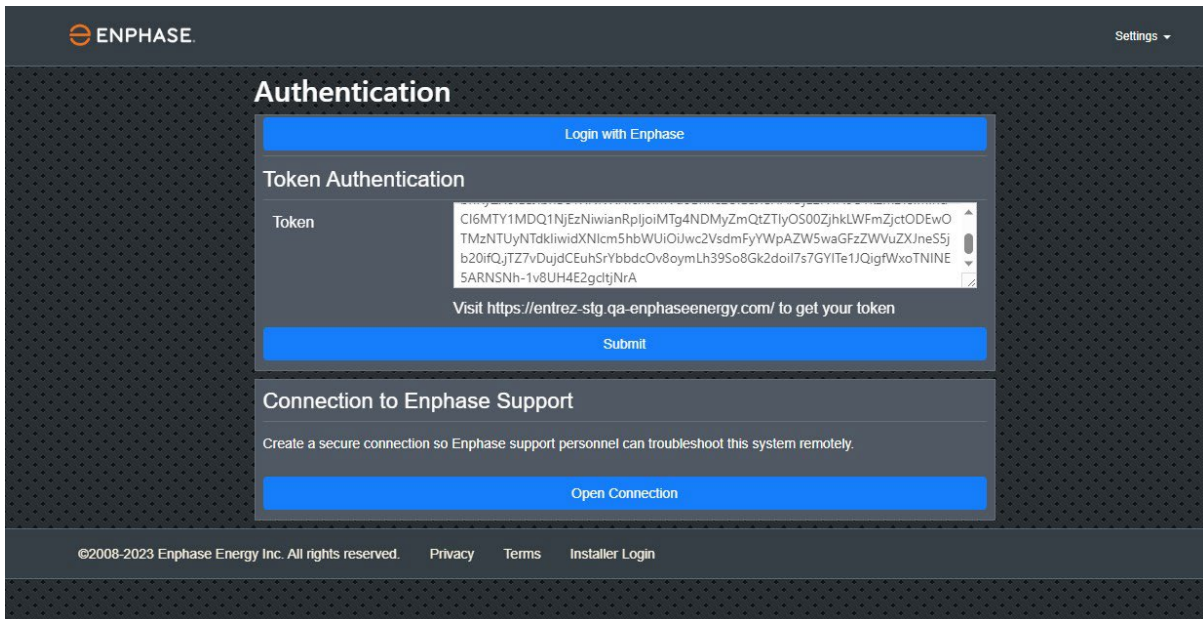
5. If you are online, click the **Login with Enphase** in the authentication screen, and enter the system owner’s Enphase Cloud credentials. Authentication with Enphase Cloud happens automatically, and the browser is redirected to the IQ Gateway page. The token is not required in this case.



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.





## 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 Gateway APIs:

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 https://<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
<a href="#">Meter details</a>	GET http://{IQ Gateway_ip}/ivp/meters	Returns meter status, type of meter, and number of phase measurements.
<a href="#">Meter readings</a>	GET http://{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.
<a href="#">Production meter data</a>	GET http://{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.
<a href="#">Production data</a>	GET http://{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.
<a href="#">Inverter production data</a>	GET http://{IQ Gateway_ip}/api/v1/production/inverters	Returns maximum and last reported active power production information of the available microinverters.

<a href="#">Meter's live data</a>	GET http://{IQ Gateway_ip}/ivp/livedata/status	Returns meter's live data with tasks and counters.
<a href="#">Power consumption data</a>	GET http://{IQ Gateway_ip}/ivp/meters/reports/consumption	Returns power consumption information of the loads.

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 HTTP GET is done on any of these APIs.

## Appendix

### Get meter details

GET http://{IQ\_Gateway\_ip}/ivp/meters

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

Sample response:

```
[
  {
    "eid": 704643328,
    "state": "enabled",
    "measurementType": "production", "phaseMode": "split",
    "phaseCount": 2,
    "meteringStatus": "normal",
    "statusFlags": [ ]
  },
  {
    "eid": 704643584,
    "state": "enabled",
    "measurementType": "net-consumption", "phaseMode": "split",
    "phaseCount": 2,
    "meteringStatus": "normal",
    "statusFlags": [ ]
  }
]
```

### Get meter readings

GET http://{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,
  "timestamp": 1654218661,
  "actEnergyDivd": 1608426.912, "actEnergyRcvd": 4.923,
  "apparentEnergy": 1648123.109, "reactEnergyLagg": 52600.292, "reactEnergyLead":
  19013.342, "instantaneousDemand": 132.118, "activePower": 132.118,
  "apparentPower": 5328.778, "reactivePower": -5328.778,
  "pwrFactor": 0.025,
  "voltage": 246.377,
  "current": 43.257,
  "freq": 59.188,
  "channels": [{
    "eid": 1778385169,
    "timestamp": 1654218661,
    "actEnergyDivd": 803639.138,
    "actEnergyRcvd": 2.650,
    "apparentEnergy": 823442.481,
    "reactEnergyLagg": 26264.291,
    "reactEnergyLead": 9545.452,
    "instantaneousDemand": 66.037,
    "activePower": 66.037,
    "apparentPower": 2663.476,
    "reactivePower": -2663.476,
    "pwrFactor": 0.025,
    "voltage": 123.184,
    "current": 21.622,
    "freq": 59.188
  },
  {
    "eid": 1778385170,
    "timestamp": 1654218661,
    "actEnergyDivd": 804787.774,
    "actEnergyRcvd": 2.273,
    "apparentEnergy": 824680.628,
    "reactEnergyLagg": 26336.001,
```

```
"reactEnergyLead": 9467.890,  
"instantaneousDemand": 66.082,  
"activePower": 66.082,  
"apparentPower": 2665.302,  
"reactivePower": -2665.302,  
"pwrFactor": 0.025,  
"voltage": 123.193,  
"current": 21.635,  
"freq": 59.188  
},  
{  
"eid": 1778385171,  
"timestamp": 1654218661,  
"actEnergyDivd": 0.000,  
"actEnergyRcvd": 0.000,  
"apparentEnergy": 0.000,  
"reactEnergyLagg": 0.000,  
"reactEnergyLead": 0.000,  
"instantaneousDemand": 0.000,  
"activePower": 0.000,  
"apparentPower": 0.000,  
"reactivePower": 0.000,  
"pwrFactor": 0.000,  
"voltage": 0.000,  
"current": 0.000,  
"freq": 59.188  
}  
]  
},  
{  
"eid": 704643584,  
"timestamp": 1654218661,  
"actEnergyDivd": 48540.732,  
"actEnergyRcvd": 1244797.861,  
"apparentEnergy": 1332629.594,  
"reactEnergyLagg": 13955.857,  
"reactEnergyLead": 30823.381,  
"instantaneousDemand": -0.000,  
"activePower": -0.000,  
"apparentPower": 34.831,
```

```

"reactivePower": -0.000,
"pwrFactor": 0.000,
"voltage": 246.338,
"current": 0.283,
"freq": 59.188,
"channels": [{
"eid": 1778385425,
"timestamp": 1654218661,
"actEnergyDivd": 24176.961,
"actEnergyRcvd": 600344.235,
"apparentEnergy": 644044.993,
"reactEnergyLagg": 5391.081,
"reactEnergyLead": 15459.001,
"instantaneousDemand": -0.000,
"activePower": -0.000,
"apparentPower": 16.858,
"reactivePower": -0.000,
"pwrFactor": 0.000,
"voltage": 123.152,
"current": 0.137,
"freq": 59.188
}],
{
"eid": 1778385426,
"timestamp": 1654218661,
"actEnergyDivd": 24363.771,
"actEnergyRcvd": 644453.626,
"apparentEnergy": 688584.601,
"reactEnergyLagg": 8564.776,
"reactEnergyLead": 15364.380,
"instantaneousDemand": -0.000,
"activePower": -0.000,
"apparentPower": 17.973,
"reactivePower": -0.000,
"pwrFactor": 0.000,
"voltage": 123.186,
"current": 0.146,
"freq": 59.188
}
    
```

```
"eid": 1778385427,  
"timestamp": 1654218661,  
"actEnergyDivd": 129399.711,  
"actEnergyRcvd": 93791.210,  
"apparentEnergy": 242548.385,  
"reactEnergyLagg": 15196.459,  
"reactEnergyLead": 10272.271,  
"instantaneousDemand": 0.000,  
"activePower": 0.000,  
"apparentPower": 2697.761,  
"reactivePower": 2697.761,  
"pwrFactor": 0.000,  
"voltage": 123.175,  
"current": 21.902,  
"freq": 59.188  
}  
]  
},  
{  
"eid": 704643840,  
"timestamp": 1654218661,  
"actEnergyDivd": 258799.422,  
"actEnergyRcvd": 187582.421,  
"apparentEnergy": 485096.770,  
"reactEnergyLagg": 30392.918,  
"reactEnergyLead": 20544.543,  
"instantaneousDemand": 0.000,  
"activePower": 0.000,  
"apparentPower": 5395.521,  
"reactivePower": 5395.521,  
"pwrFactor": 0.000,  
"voltage": 246.351,  
"current": 43.804,  
"freq": 59.188,  
"channels": [{  
"eid": 1778385681,  
"timestamp": 1654218661,  
"actEnergyDivd": 129399.711,  
"actEnergyRcvd": 93791.210,  
"apparentEnergy": 242548.385,
```



```
"reactEnergyLagg": 15196.459,  
"reactEnergyLead": 10272.271,  
"instantaneousDemand": 0.000,  
"activePower": 0.000,  
"apparentPower": 2697.761,  
"reactivePower": 2697.761,  
"pwrFactor": 0.000,  
"voltage": 123.175,  
"current": 21.902,  
"freq": 59.188  
},  
{  
"eid": 1778385682,  
"timestamp": 1654218661,  
"actEnergyDlvd": 129399.711,  
"actEnergyRcvd": 93791.210,  
"apparentEnergy": 242548.385,  
"reactEnergyLagg": 15196.459,  
"reactEnergyLead": 10272.271,  
"instantaneousDemand": 0.000,  
"activePower": 0.000,  
"apparentPower": 2697.761,  
"reactivePower": 2697.761,  
"pwrFactor": 0.000,  
"voltage": 123.175,  
"current": 21.902,  
"freq": 59.188  
},  
{  
"eid": 1778385683,  
"timestamp": 1654218661,  
"actEnergyDlvd": 0.000,  
"actEnergyRcvd": 0.000,  
"apparentEnergy": 0.000,  
"reactEnergyLagg": 0.000,  
"reactEnergyLead": 0.000,  
"instantaneousDemand": 0.000,  
"activePower": 0.000,  
"apparentPower": 0.000,  
"reactivePower": 0.000,
```

```
"pwrFactor": 0.000,
"voltage": 0.000,
"current": 0.000,
"freq": 59.188
}
]
}
]
```

## Get production meter data

**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,
"wattHoursSevenDays": 719543,
"wattHoursLifetime": 1608587,
"wattsNow": 227
}
```

## Get production data

GET [http://{IQ Gateway\\_ip}/ivp/pdm/energy](http://{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": {
      "wattHoursToday": 13251,
      "wattHoursSevenDays": 91306,
      "wattHoursLifetime": 8250671,
      "wattsNow": 596
    },
    "rgm": {
      "wattHoursToday": 0,
      "wattHoursSevenDays": 0,
      "wattHoursLifetime": 0,
      "wattsNow": 0
    },
    "eim": {
      "wattHoursToday": 0,
```

```

"wattHoursSevenDays": 0,
"wattHoursLifetime": 0,
"wattsNow": 0
},
"consumption": {
  "eim": {
    "wattHoursToday": 0,
    "wattHoursSevenDays": 0,
    "wattHoursLifetime": 0,
    "wattsNow": 0
  }
}
    
```

## Get inverter production data

GET `http://{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",
    "lastReportDate": 1654171836,
    "devType": 1,
    "lastReportWatts": 15,
    "maxReportWatts": 38
  },
  {
    "serialNumber": "121935144623",
    "lastReportDate": 1654171766,
    "devType": 1,
    "lastReportWatts": 5,
    "maxReportWatts": 5
  }
]
    
```

## Get meter's live data

GET `http://{IQ_Gateway_ip}/ivp/livedata/status`

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

Sample response:

```
{
  "connection": {
    "mqtt_state": "connected",
    "prov_state": "configured",
    "auth_state": "ok",
    "sc_stream": "enabled",
    "sc_debug": "enabled"
  },
  "meters": {
    "last_update": 1654221647,
    "soc": 100,
    "main_relay_state": 0,
    "gen_relay_state": 5,
    "backup_bat_mode": 1,
    "backup_soc": 10,
    "is_split_phase": 1,
    "phase_count": 0,
    "enc_agg_soc": 100,
    "enc_agg_energy": 24800,
    "acb_agg_soc": 0,
    "acb_agg_energy": 0,
    "pv": {
      "agg_p_mw": 329549,
      "agg_s_mva": 329549,
      "agg_p_ph_a_mw": 329549,
      "agg_p_ph_b_mw": 0,
      "agg_p_ph_c_mw": 0,
      "agg_s_ph_a_mva": 329549,
      "agg_s_ph_b_mva": 0,
      "agg_s_ph_c_mva": 0
    },
    "storage": {
      "agg_p_mw": -220800,
      "agg_s_mva": -559446,
      "agg_p_ph_a_mw": -220800,
      "agg_p_ph_b_mw": 0,
      "agg_p_ph_c_mw": 0,
      "agg_s_ph_a_mva": -559446,

```

```
"agg_s_ph_b_mva": 0,
"agg_s_ph_c_mva": 0
},
"grid": {
"agg_p_mw": 0,
"agg_s_mva": 0,
"agg_p_ph_a_mw": 0,
"agg_p_ph_b_mw": 0,
"agg_p_ph_c_mw": 0,
"agg_s_ph_a_mva": 0,
"agg_s_ph_b_mva": 0,
"agg_s_ph_c_mva": 0
},
"load": {
"agg_p_mw": 108749,
"agg_s_mva": -229897,
"agg_p_ph_a_mw": 108749,
"agg_p_ph_b_mw": 0,
"agg_p_ph_c_mw": 0,
"agg_s_ph_a_mva": -229897,
"agg_s_ph_b_mva": 0,
"agg_s_ph_c_mva": 0
},
"generator": {
"agg_p_mw": 0,
"agg_s_mva": 0,
"agg_p_ph_a_mw": 0,
"agg_p_ph_b_mw": 0,
"agg_p_ph_c_mw": 0,
"agg_s_ph_a_mva": 0,
"agg_s_ph_b_mva": 0,
"agg_s_ph_c_mva": 0
}
},
"tasks": {
"task_id": 27672012,
"timestamp": 1654219883
},
"counters": {
"main_CfgLoad": 1,
```

```

"main_CfgChanged": 1,
"main_taskUpdate": 62,
"MqttClient_publish": 10260,
"MqttClient_live_debug": 190,
"MqttClient_respond": 260,
"MqttClient_msgarrvd": 130,
"MqttClient_create": 13,
"MqttClient_setCallbacks": 13,
"MqttClient_connect": 13,
"MqttClient_connect_err": 5,
"MqttClient_connect_Err": 5,
"MqttClient_subscribe": 8,
"SSL_Keys_Create": 13,
"sc_hdlIDataPub": 9440,
"sc_SendStreamCtrl": 72,
"sc_SendDemandRspCtrl": 65517,
"rest_Meters": 7,
"rest_Status": 579
}
}
    
```

## Get power consumption data

GET [http://{IQ\\_Gateway\\_ip}/ivp/meters/reports/consumption](http://{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,
"reportType": "net-consumption",
"cumulative": {
"currW": 119.423,
"actPower": 119.423,
"apprntPwr": 105.678,
"reactPwr": -261.046,
"whDlvdCum": 43110.122,
"whRcvdCum": 0.000,
"varhLagCum": -25071.856,
    
```

```
"varhLeadCum": 35895.778,  
"vahCum": 192725.807,  
"rmsVoltage": 241.427,  
"rmsCurrent": 0.875,  
"pwrFactor": 1.00,  
"freqHz": 60.00  
},  
"lines": [  
{  
"currW": 56.672,  
"actPower": 56.672,  
"appntPwr": 49.248,  
"reactPwr": -136.579,  
"whDlvdCum": 21051.342,  
"whRcvdCum": 0.000,  
"varhLagCum": -12541.347,  
"varhLeadCum": 18473.849,  
"vahCum": 96511.746,  
"rmsVoltage": 120.673,  
"rmsCurrent": 0.408,  
"pwrFactor": 1.00,  
"freqHz": 60.00  
},  
{  
"currW": 62.751,  
"actPower": 62.751,  
"appntPwr": 56.430,  
"reactPwr": -124.467,  
"whDlvdCum": 22058.779,  
"whRcvdCum": 0.000,  
"varhLagCum": -12530.509,  
"varhLeadCum": 17421.929,  
"vahCum": 96214.061,  
"rmsVoltage": 120.753,  
"rmsCurrent": 0.467,  
"pwrFactor": 1.00,  
"freqHz": 60.00  
}  
]  
},
```

```
{
  "createdAt": 1654625079,
  "reportType": "net-consumption",
  "cumulative": {
    "currW": -1905.274,
    "actPower": -1905.274,
    "apprntPwr": -1920.786,
    "reactPwr": -260.398,
    "whDlvdCum": -152327.377,
    "whRcvdCum": 0.000,
    "varhLagCum": 32.752,
    "varhLeadCum": 35951.521,
    "vahCum": 192725.807,
    "rmsVoltage": 241.427,
    "rmsCurrent": -15.912,
    "pwrFactor": -1.00,
    "freqHz": 60.00
  },
  "lines": [
    {
      "currW": -954.876,
      "actPower": -954.876,
      "apprntPwr": -963.431,
      "reactPwr": -136.579,
      "whDlvdCum": -76608.517,
      "whRcvdCum": 0.000,
      "varhLagCum": 16.155,
      "varhLeadCum": 18488.097,
      "vahCum": 96511.746,
      "rmsVoltage": 120.673,
      "rmsCurrent": -7.984,
      "pwrFactor": -1.00,
      "freqHz": 60.00
    },
    {
      "currW": -950.398,
      "actPower": -950.398,
      "apprntPwr": -957.355,
      "reactPwr": -123.819,
      "whDlvdCum": -75718.860,
```



```
"whRcvdCum": 0.000,  
"varhLagCum": 16.598,  
"varhLeadCum": 17463.423,  
"vahCum": 96214.061,  
"rmsVoltage": 120.753,  
"rmsCurrent": -7.928,  
"pwrFactor": -1.00,  
"freqHz": 60.00  
}  
]  
}
```

## Revision history

Revision	Date	Description
TEB-00060-1.0	August 2023	Added production meter data and production data sections.
Previous release		

© 2023 Enphase Energy. All rights reserved. Enphase, the e and CC logos, IQ, and certain other marks listed at <https://enphase.com/trademark-usage-guidelines> are trademarks of Enphase Energy, Inc. in the US and other countries. Data subject to change.