SmartPVMS
24.5.0

NBI Reference

Issue	01
Date	2024-07-01

Huawei Digital Power Technologies Co., Ltd.

Contents

1 Change History

24.5.0 Northbound API Changes

03

Optimized **3.1.1 Creating an OAuth 2.0 Client for a Third-Party App**.

02

Added **7.5 Scenario-based Practices of Battery Scheduling**.

Optimized **4.2 Flow Control Using the API Account** and **5.2.1 API for Delivering Battery Charge and Discharge Tasks**.

01

API Name	Method and Path	Description	Remarks
API for Delivering a Task for Setting the Battery Working Mode	https://*Domain name of the management system*/rest/openapi/pvms/nbi/v1/control/battery/mode/async-task	This API is used to deliver the task of setting the battery working mode based on the plant DN. For the same PV plant, do not call this API repeatedly before a task is complete. The battery working mode can be set to **Maximum Self-Consumption** or **TOU**.	New API

API Name	Method and Path	Description	Remarks
API for Querying a Task for Setting the Battery Working Mode	https://*Domain name of the management system*/rest/openapi/pvms/nbi/v1/control/battery/mode/task-info	This API is used to query the execution status of the task for setting the battery working mode. Do not call this API repeatedly before a task is complete.	New API
API for Delivering a Task for Setting Battery Parameters	https://*Domain name of the management system*/rest/openapi/pvms/nbi/v1/control/battery/configuration/async-task	This API is used to deliver the task of setting the battery parameters based on the plant DN. For the same PV plant, do not call this API repeatedly before a task is complete. Users can set plant-level battery parameters (end-of-charge SOC, end-of-discharge SOC, maximum charge power, and maximum discharge power).	New API
API for Querying a Task for Setting Battery Parameters	https://*Domain name of the management system*/rest/openapi/pvms/nbi/v1/control/battery/configuration/task-info	This API is used to query the execution status of the task for setting the battery parameters. Do not call this API repeatedly before a task is complete.	New API
API for Delivering an Inverter Active Power Setting Task	https://*Domain name of the management system*/rest/openapi/pvms/nbi/v2/control/active-power-control/async-task	This API is used to deliver an inverter active power setting task based on the plant DN. For the same plant, do not call this API repeatedly before a task is complete. The active power can be controlled in two modes: unlimited and limited feed-in (kW).	New API

API Name	Method and Path	Description	Remarks
API for Querying Inverter Active Power Setting Tasks	https://management system domain name/rest/openapi/pvms/nbi/v2/control/active-power-control/task-info	Query the execution status of the inverter active power setting task. Do not call this API repeatedly before a task is complete.	New API
Hourly Plant Data API	https://Domain name of the management system/thirdData/getKpiStationHour	The **PVYield** and **inverterYield** fields are added to the returned packet.	Modified API
Daily Plant Data API	https://{*Domain name of the management system}*/thirdData/getKpiStationDay	The **PVYield** and **inverterYield** fields are added to the returned packet.	Modified API
Monthly Plant Data API	https://{*Domain name of the management system}*/thirdData/getKpiStationMonth	The **PVYield** and **inverterYield** fields are added to the returned packet.	Modified API
Yearly Plant Data API	https://Domain name of the management system/thirdData/getKpiStationYear	The **PVYield** and **inverterYield** fields are added to the returned packet.	Modified API
API for Delivering an Inverter Active Power Setting Task	https://Management system domain name/rest/openapi/pvms/nbi/v1/control/active-power-control/async-task	You are not advised to use this API. **5.2.7 API for Delivering an Inverter Active Power Setting Task** is recommended.	Modified API
API for Querying Inverter Active Power Setting Tasks	https://Management system domain name/rest/openapi/pvms/nbi/v1/control/active-power-control/task-info	You are not advised to use this API. **5.2.8 API for Querying Inverter Active Power Setting Tasks** is recommended.	Modified API

24.4.0 Northbound API Changes

01

None

24.2.0 Northbound API Changes

03

Optimized **3.1.2.2 Initiating Authorization to a Third-party App by an Owner**, **6.4 Exception Code List**, **8.1.1 Handling the Exception Returned by Calling the Authorization Request API**, **8.2.1 Why Do I Fail to Create an API Account?**, and **8.3 Why Is No Data or Only Part of Data Found When I Call a Northbound API for Data Query?**

02

Optimized **3.1 OAuth Connect** and **4.1 Flow Control Policy in OAuth Connect Mode**.

Added **8.1.6 How Do I Obtain O&M Support When Open APIs Are Accessed in OAuth Connect Mode?**

01

> ⚠️ **CAUTION**
>
> During the evolution of open APIs, the original V6 APIs are incorporated into the basic APIs. For details, see **5.1 Basic APIs**. This change does not affect the normal use of the original APIs.
>
> Some APIs are iterated, as listed in the following table. You are advised to use new high-performance APIs instead.

Added **3.1 OAuth Connect**.

Old API	New API	Description
/thirdData/getStationList	/thirdData/stations	The new API supports pagination and data query by grid connection time. For details, see **5.1.1.1 Plant List API**.
/thirdData/getDevFiveMinutes	/thirdData/getDevHistoryKpi	The new API supports a longer time range and better query performance. For details, see **5.1.2.3 Historical Device Data API**.

Old API	New API	Description
/rest/openapi/ pvms/v1/vpp/ chargeAndDischarge	/rest/openapi/ pvms/nbi/v2/control/ charge-and-discharge/ async-task	The new API can deliver different parameters to each plant. For details, see **5.2.1 API for Delivering Battery Charge and Discharge Tasks**.

2 Overview

NOTICE

Please use the latest northbound API reference document by choosing **Company Management** > **Northbound Management** > **Developer Guide**.
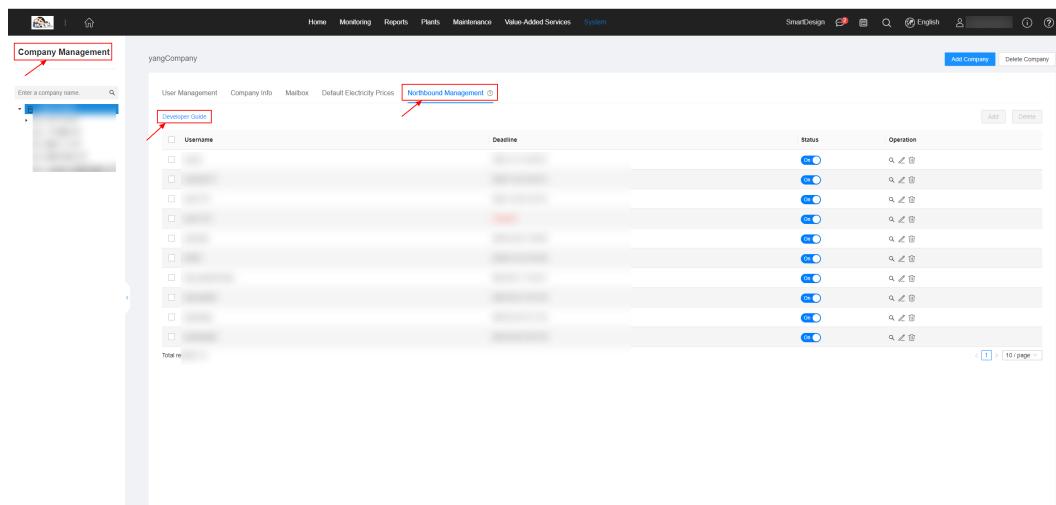
2.1 Introduction to Northbound Open APIs

Northbound Open APIs

Northbound open APIs are open interfaces based on the Representational State Transfer (REST) standard and can facilitate integration of third-party systems.

Third-party systems can use RESTful NBIs to access the resources authorized by the SmartPVMS, such as access authentication, configuration, alarm, and performance data.

APIs can be used only after authorization and the access is secure. Only HTTPS access is supported.

The JavaScript Object Notation (JSON) data format is used for data interaction. The data format is simple, easy to read and write, and occupies less network traffic than XML.

What Is REST?

REST, short for Representational State Transfer, is a design and development mode for network applications. It simplifies development and improves system scalability.

REST uses resources as its core, and resources are uniquely identified by a uniform resource identifier (URI), for example, /rest/openapi/pvms/v1/plants.

REST uses four types of standard operations to access resources: POST, GET, PUT, and DELETE.

- POST: creates resources.
- GET: queries resources.
- PUT: updates resources.
- DELETE: deletes resources.

The SmartPVMS provides external services using URIs. Users obtain SmartPVMS resources through URIs and obtain services.

HTTP Status Codes

The first line of all HTTP responses is the status line, which contains the current HTTP version number, the status code consisting of three digits, and the phrase that describes the status, which are separated by spaces.

The first digit of the status code indicates the type of the current response.

- 1xx message: The request has been received by the server and continues to be processed.
- 2xx success: The request has been received, understood, and accepted by the server.
- 3xx redirection: This request can be completed only after subsequent operations are performed.
- 4xx request error: The request contains a syntax error or cannot be executed.
- 5xx server error: An error occurs when the server processes a correct request.

2.2 API Architecture

The SmartPVMS provides a set of WebService APIs for third-party systems and third-party developers, who can construct HTTPS requests to call APIs and obtain SmartPVMS resources and data.

Figure 2-1 WebService NBI architecture
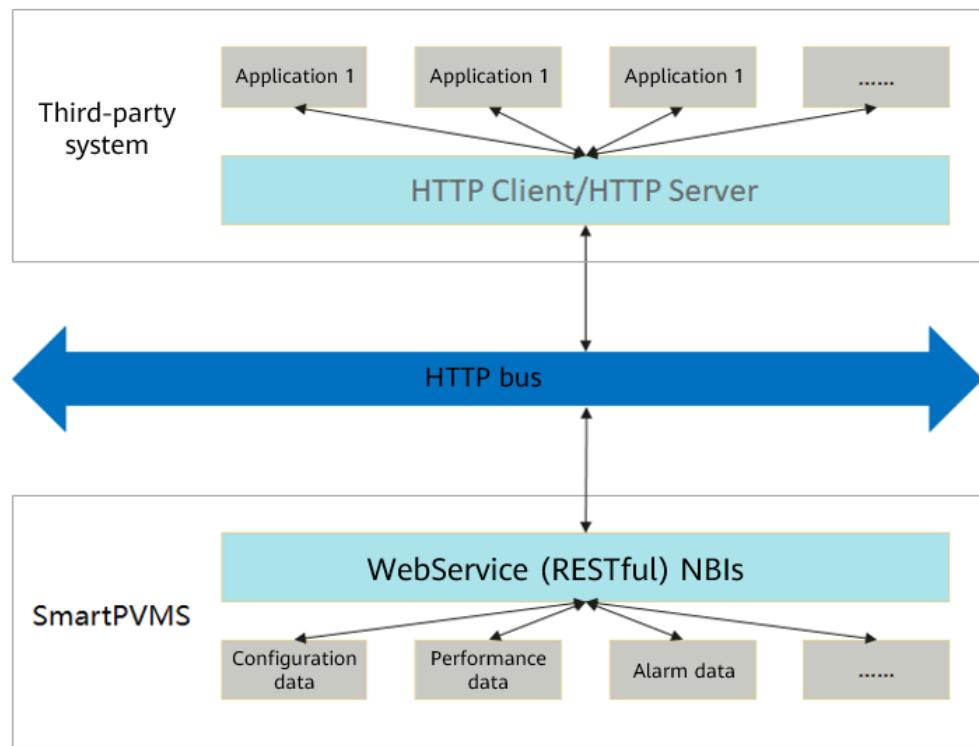
2.3 API Overview

API Category	API Label	API Name	Description
Basic APIs	Basic data	Plant List	Returns the static information about the plants on which the current account has permission by page. Some information, such as the plant ID, is the prerequisite for querying other API data. The static information changes only when the plants change. The third-party system must save the information.
		Device List	Returns the device list of a given plant by page. Some information, such as the plant ID, is the prerequisite for querying other API data. The third-party system must save the information.
	Monitoring data	Real-time plant data	Obtains real-time data such as the energy yield, revenue, and health status of the plant on the current day. The data is updated every 5 minutes. The calling party must save the data.

API Category	API Label	API Name	Description
		Real-time device data	Obtains real-time device data. The real-time data varies depending on the device type. The data is updated every 5 minutes. The calling party must save the data.
		Historical device data	Obtains the 5-minute data of a device in a given period. The data items vary depending on the device type. The calling party must save the data.
	Alarm data	Querying active alarms	Queries device active alarms of by plant or query alarms of a specific device.
	Report data	Hourly plant data	Queries the hourly data of a plant in a day. The data items include inverter energy yield and feed-in energy. The data update is delayed. Generally, only the current-day data is updated. The calling party must save the data.
		Daily plant data	Queries the daily data of a plant in a month. The data items include inverter energy yield, feed-in energy, and energy consumption. The data update is delayed. Generally, only the current-day data is updated. The calling party must save the data.
		Monthly plant data	Queries the monthly data of a plant in a year. The data items include inverter energy yield, feed-in energy, and energy consumption. The data update is delayed. Generally, only the current-month data is updated. The calling party must save the data.
		Yearly plant data	Queries the yearly data of a plant in its life cycle. The data items include inverter energy yield, feed-in energy, and energy consumption. The data update is delayed. Generally, only the current-year data is updated. The calling party must save the data.
		Daily device data	Queries the daily data of a device. The data items vary depending on the device. The data update is delayed. Generally, only the current-day data is updated. The calling party must save the data.
		Monthly device data	Queries the monthly data of a device. The data items vary depending on the device. The data update is delayed. Generally, only the current-month data is updated. The calling party must save the data.

API Category	API Label	API Name	Description
		Yearly device data	Queries the yearly data of a device. The data items vary depending on the device. The data update is delayed. Generally, only the current-year data is updated. The calling party must save the data.
Control APIs	ESS charge/ discharge	Delivering battery charge/ discharge tasks	Delivers battery charge/discharge tasks based on plant codes. For the same plant, do not call this API repeatedly before a task is complete. If there are multiple ESSs in the power plant, the task is executed on every ESS.
		Querying battery charge/ discharge tasks	Queries the delivery result of an asynchronous battery charge/discharge task. Do not call this API repeatedly before a task is complete.
	Inverter active power setting	API for Delivering an Inverter Active Power Setting Task	This API is used to deliver an inverter active power setting task based on the plant DN. For the same plant, do not call this API repeatedly before a task is complete. The active power can be controlled in two modes: unlimited and limited feed-in (kW).
		API for Querying Inverter Active Power Setting Tasks	Query the execution status of the inverter active power setting task. Do not call this API repeatedly before the task is complete.
	Setting the battery working mode	API for Delivering a Task for Setting the Battery Working Mode	This API is used to deliver the task of setting the battery working mode based on the plant DN. For the same PV plant, do not call this API repeatedly before a task is complete. The battery working mode can be set to **Maximum Self-Consumption** or **TOU**.
		API for Querying a Task for Setting the Battery Working Mode	This API is used to query the execution status of the task for setting the battery working mode. Do not call this API repeatedly before a task is complete.

API Category	API Label	API Name	Description
	Battery parameter setting	API for Delivering a Task for Setting Battery Parameters	This API is used to deliver the task of setting the battery parameters based on the plant DN. For the same PV plant, do not call this API repeatedly before a task is complete. Users can set plant-level battery parameters (end-of-charge SOC, end-of-discharge SOC, maximum charge power, and maximum discharge power).
		API for Querying a Task for Setting Battery Parameters	This API is used to query the execution status of the task for setting the battery parameters. Do not call this API repeatedly before a task is complete.

3 API Access

The SmartPVMS northbound open APIs provide two access modes for third-party systems and third-party developers.

In the scenario where a third-party enterprise platform is interconnected, OAuth Connect is used to access northbound open APIs. Basic APIs and control APIs can be used in the OAuth Connect scenario.

The platforms developed by installers can access the open APIs using API accounts. Basic APIs can be used in the API account access scenario.

NOTICE

The access to open APIs in OAuth Connect mode is for trial use only in Europe. The policy may change in the future.

3.1 OAuth Connect

In OAuth Connect mode, third-party apps need to adapt to the OAuth 2.0 protocol to access northbound open APIs.

OAuth 2.0 is an open authorization protocol that allows third-party apps to access resources authorized by owners without obtaining their usernames and passwords.

A third-party app can apply for creating an OAuth 2.0 client on the FusionSolar. After the FusionSolar owner authorizes the application, the FusionSolar owner account is bound to the client. The third-party app then can access the northbound open APIs to access the resources authorized by the owner.

The following sections describe the process of accessing northbound open APIs in OAuth Connect mode.

3.1.1 Creating an OAuth 2.0 Client for a Third-Party App

To enable a third-party app to access northbound open APIs in OAuth Connect mode, you need to apply for and create an OAuth 2.0 client on the FusionSolar. Use the following template to initiate an application and send it to

huaweipartners@huawei.com (available only in Europe). FusionSolar will help complete the application. The procedure is as follows.

Step 1 A third-party app initiates an application based on the following template.

Application Information	Description	Filled by Third-Party Platform
Agreement on Use of Huawei APIs	By initiating an application, you acknowledge and agree to the ***Agreement on Use of Huawei APIs**.	Agree
Company Name	Company name	
Registered Place	Registered place of the company	
Company Address	Company address	
Legal Entity	Legal entity	
Employer Identification Number (EIN)	Employer identification number	
Contact Number	Contact number	
Email	Email address for registering a third-party app. After the registration is successful, related information will be sent to this email address. Example: Huawei_Tool@huawei.com	
App Name	Name of a third-party app. The value can contain letters, digits, hyphens (-), and underscores (_). This field displays information about apps authorized by owners. Example: Huawei_Tool	
Description	Third-party app description. Example: a tool developed by Huawei Digital Power. It is mainly used in residential PV scenarios to provide customers with a convenient home energy management service platform.	

Application Information	Description	Filled by Third-Party Platform
Redirect URL	Callback URL in the OAuth 2.0 authentication process. The redirect URI cannot contain **..**, **redirect**, or **#**. After the owner approves the authorization, an authorization code is generated. The third-party app can obtain the authorization code through **RedirectUrl**. Example: https://www.huaweitool.com/xxx?somearg=huaweitool.mainpage	

If a third-party app needs to modify **RedirectUrl**, **RedirectUrl** used when the owner initiates authorization and the third-party app obtains the token also needs to be updated.

Step 2 After completing the application, the FusionSolar returns the following OAuth 2.0 client information to the third-party app.

Parameter	Data Type	Description
client_id	String	Client ID. It is the ID allocated by FusionSolar to a third-party app.
client_secret	String	Client secret. It is the client secret allocated by FusionSolar to a third-party app.
scope	String	Permission set. It is the permission set opened by FusionSolar to third-party apps, for example, basic APIs (pvms.openapi.basic) and control APIs (pvms.openapi.control).

After receiving **client_id** and **client_secret**, the third-party app must keep them secure.

----End

3.1.2 Integrating OAuth 2.0 to Third-Party Apps

Third-party apps can integrate the authorization request link of the FusionSolar to manage the authorization request function. When an owner clicks the corresponding function, the authorization request process starts.

3.1.2.1 Adaptation Process Overview

Third-party app clients can access northbound open APIs in OAuth Connect mode. The adaptation process is as follows: An owner accesses the authorization request function integrated by a third-party app. The FusionSolar displays the login page and authorization page for the owner to log in and authorize the OAuth 2.0 client of the third-party app, and returns the authorization code generated after the owner's authorization to the third-party app through the redirect URL. After obtaining the authorization code, the third-party app uses it to access the FusionSolar service and obtain the access token and refresh token. The third-party app then uses the access token to access open APIs. The following figure shows the process.
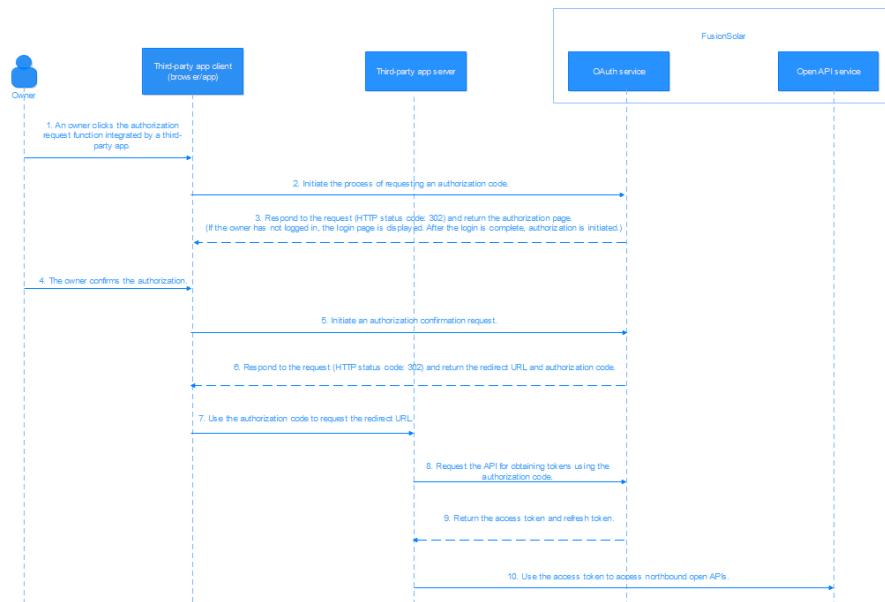
1-2: An owner accesses the FusionSolar through the authorization request link integrated to a third-party app client and **initiates the authorization code request process**.

3: The OAuth service responds to the request (HTTP status code: 302) and returns the authorization page to the third-party app client. If the owner has not logged in to FusionSolar, the OAuth service returns to the FusionSolar login page. After the owner logs in, the authorization page is displayed.

4-5: After the owner confirms the authorization, the third-party app client accesses the OAuth service to initiate an authorization request to the third-party app client (OAuth 2.0 client).

6: The OAuth service responds to the request (HTTP status code: 302) and returns the **redirect URL** and authorization code to the third-party app client.

7: The third-party app client sends a request carrying the authorization code to request the **redirect URL**.

8-9: After obtaining the authorization code, the third-party app server uses the authorization code to **send a request to the OAuth API for obtaining tokens** to obtain the access token and refresh token.

10: The third-party app server uses the access token to access the northbound open APIs for subsequent service processing.

NOTICE

An authorization code is used to generate an access token of a northbound open API. A refresh token is used for refreshing the access token.

After the owner accesses the OAuth service and initiates authorization to a third-party app, an authorization code is generated. The third-party app can use the authorization code to access the OAuth service again to obtain the access token and refresh token.

An authorization code is valid for 5 minutes. After obtaining the authorization code, the third-party app needs to obtain the access token and refresh token as soon as possible. After the authorization code expires, the owner needs to access the OAuth service again to initiate authorization to the third-party app.

When a third-party app accesses the northbound open APIs in OAuth Connect mode, the access token is used for authentication. The refresh token is used to exchange for a new access token. The access token is valid for one hour. After it expires, a third-party app cannot access the open APIs using the expired access token and needs to obtain a new access token using the refresh token. If a refresh token is used to obtain a new access token even when the old access token is still in its validity period, the old access token will become invalid, and the third-party app cannot access the open API through the old access token.

The access token and refresh token are sensitive information. Keep them secure.

3.1.2.2 Initiating Authorization to a Third-party App by an Owner

After a third-party app applies for an OAuth 2.0 client on the FusionSolar, an owner initiates authorization to the client, accesses the OAuth service, calls the authorization request API provided by the OAuth service, logs in to the FusionSolar, and selects a permission set. After the authorization is complete and successful, an authorization code is generated. The third-party app obtains the authorization code through **RedirectUrl**.

Description of the Authorization Request API

The authorization request API is provided by the OAuth service for owners to authorize OAuth 2.0 clients of third-party apps.

URL of the Authorization Request API

https://oauth2.fusionsolar.huawei.com/rest/dp/uidm/oauth2/v1/authorize

HTTP Method of the Authorization Request API

HTTP method: GET

Parameters of the Authorization Request API

Parameter	Mandatory (Yes/No)	Data Type	Description
response_type	Yes	String	Response type. The value is fixed at **code** for the OAuth 2.0 authentication mode.
client_id	Yes	String	Nine-digit **client-id** returned by the FusionSolar to a third-party app when the third-party app **applies for an OAuth 2.0 client**.
redirect_uri	Yes	String	**RedirectUrl** provided by the third-party app for the FusionSolar when the app **applies for an OAuth 2.0 client**.
scope	No	String	Permission set that a third-party app needs to apply for from the owner. If multiple permissions are required, use spaces to separate them. The space is encoded as %20 in the URL. For details about all supported permission sets, see the **scope description** returned by the FusionSolar to the third-party app when the third-party app **applies for an OAuth 2.0 client**. If the **scope** parameter is not specified, all permissions authorized by the owner are used by default.
locale	No	String	Language used by the FusionSolar to display the login page and authorization page to the owners. You are advised to leave this parameter blank. If this parameter is left blank, the OAuth service determines the language based on the language supported by the browser used by the owners.
			The following languages are supported: de-de (German/Germany), en-us (English/United States), es-es (Spanish/Spain), fr-fr (French/France), hu-hu (Hungarian/Hungary), it-it (Italian/Italy), ja-jp (Japanese/Japanese), ko-kr (Korean/Korea), nl-nl (Dutch/Netherlands), pl-pl (Polish/Poland), pt-br (Portuguese/Brazil), tr-tr (Turkish/Türkiye), uk-ua (Ukrainian/Ukraine), vi-vn (Vietnamese/Vietnam), zh-cn (Simplified Chinese/China), and zh-tw (Traditional Chinese/Taiwan).

Response Packet of the Authorization Request API

Parameter	Data Type	Description
code	String	Authorization code generated after the owner authorizes the third-party app. The authorization code is returned to the third-party app through **RedirectUrl** registered by the third-party app. The authorization code can be used to generate an access token of the northbound open API.

Link Example of the Authorization Request API

Assume that a third-party app creates an OAuth 2.0 client on the FusionSolar. Assume that the provided **RedirectUrl** is https://www.example.com/cn/ and the allocated **client_id** is **123456789**. The link of the authorization request API combined by the third-party app is as follows:

https://oauth2.fusionsolar.huawei.com/rest/dp/uidm/oauth2/v1/authorize?
response_type=code&client_id=**123456789**&redirect_uri=**https://www.example.com/cn/**

Note that if **RedirectUrl** contains characters such as **&** and is directly used to combine the URL of the authorization request API, a parameter exception page may be returned. As a result, **RedirectUrl** is unavailable. Therefore, URL encoding is required for **RedirectUrl**. For example, when **RedirectUrl** of the third-party app is **https://www.example1.com/cn/?example2=example3&example4=example5** and the URL code is **https%3A%2F%2Fwww.example1.com%2Fcn%2F %3Fexample2%3Dexample3%26example4%3Dexample5**, the URL of the authorization request API combined by the third-party app is as follows:

https://oauth2.fusionsolar.huawei.com/rest/dp/uidm/oauth2/v1/authorize?
response_type=code&client_id=**123456789**&redirect_uri=**https%3A%2F%2Fwww.example1.com%2Fcn%2F %3Fexample2%3Dexample3%26example4%3Dexample5**

Usage of the Authorization Request API

A third-party app needs to combine the access link (as shown in **Link Example of the Authorization Request API**) of the authorization request API and integrate the link into the third-party app so that the owner can use the link to access the FusionSolar login page.
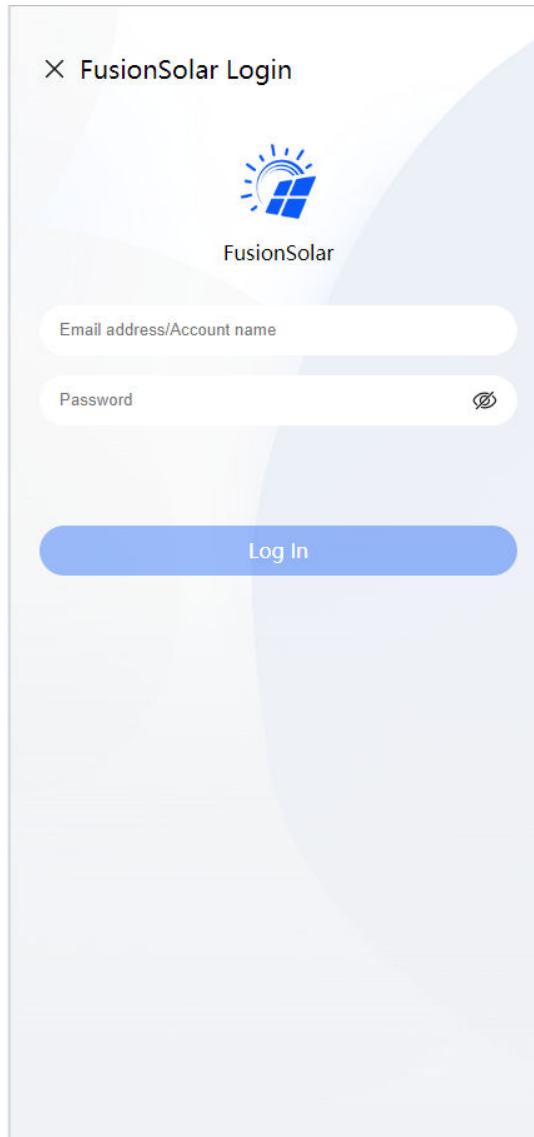
Assume that the **AppName** of a third-party app is **Third-party application**. The scope of the permission set that needs to be applied for from the owner includes basic APIs (**pvms.openapi.basic**) and control APIs (**pvms.openapi.control**). After the owner enters the FusionSolar account and password to log in to the FusionSolar, the following authorization page is displayed.
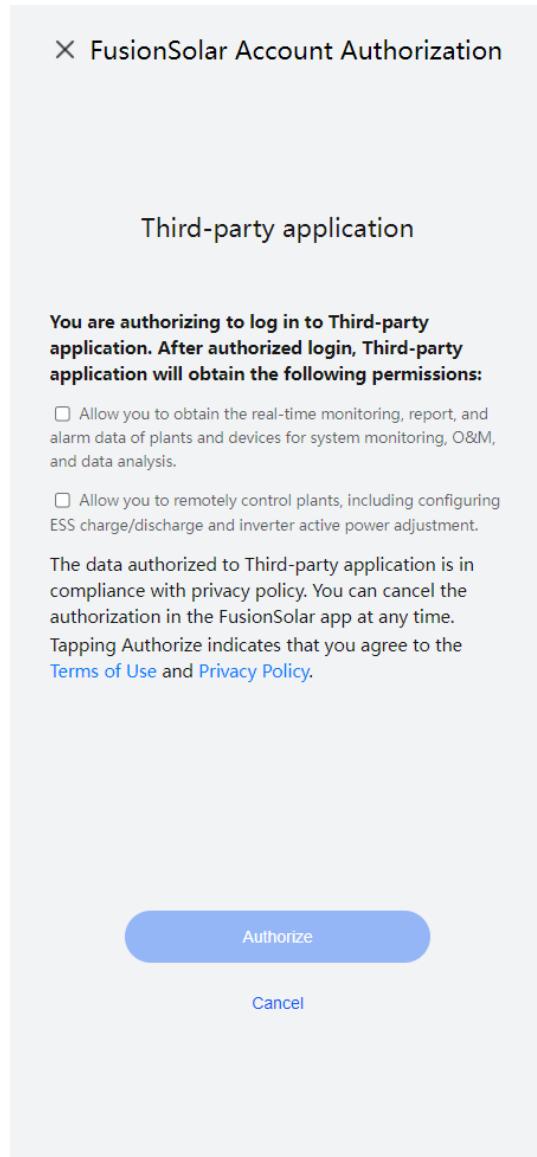
On this page, the owner can select a permission set to authorize the OAuth 2.0 client of the third-party app. After the owner completes the authorization, an authorization code is generated. The third-party app can obtain the authorization code based on the **RedirectUrl** provided for the FusionSolar when **applying for an OAuth 2.0 client**. For example, if the authorization code is B******B, the response example is as follows:

```
https://www.example.com/cn/?code=B******B
```

If the owner rejects the authorization, the third-party app can obtain the following information through **RedirectUrl**:

```
https://www.example.com/cn/?error=10002&error_description=access_is_denied
```

Exception Example of the Authorization Request API

If the **scope** parameter transferred in the combined request link is incorrect and the **RedirectUrl** provided by a third-party app for the FusionSolar when **applying for an OAuth 2.0 client** is **https://www.example.com/cn/**, the third-party app can obtain the following information based on the **RedirectUrl**:

For more exception examples and troubleshooting methods, see **8.1.1 Handling the Exception Returned by Calling the Authorization Request API**.

3.1.2.3 Obtaining the Access Token of the Open API by a Third-Party App

After obtaining the authorization code from the **RedirectUrl**, the third-party app needs to access the OAuth service again, request the token obtaining API provided by the OAuth service to obtain the access token and refresh token based on the authorization code, and select the permission set to be authorized by the owner to the third-party app.

Description of the API for Obtaining Tokens

The API for obtaining tokens is provided by the OAuth service for third-party apps to obtain the access token and refresh token. This API provides two usage methods, which are distinguished by the **grant_type** parameter transferred in the request API. When **grant_type** is set to **authorization_code**, the API for obtaining tokens can obtain the access token and refresh token based on the authorization code and the permission set selected by the owner to authorize the third-party app. When **grant_type** is set to **refresh_token**, the API for obtaining tokens can use the refresh token to obtain the access token and the permission set selected by the owner to authorize the third-party app.

Request URL of the API for Obtaining Tokens

https://oauth2.fusionsolar.huawei.com/rest/dp/uidm/oauth2/v1/token

Request Method of the API for Obtaining Tokens

HTTP method: POST

Request Headers of the API for Obtaining Tokens

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

Request Parameters of the API for Obtaining Tokens

The following table lists the request parameters for obtaining the access token and refresh token using the authorization code.

Parameter	Data Type	Description
client_id	String	Nine-digit **client-id** returned by the FusionSolar to a third-party app when the third-party app **applies for an OAuth 2.0 client**.

Parameter	Data Type	Description
code	String	**Authorization Code** generated in **3.1.2.2 Initiating Authorization to a Third-party App by an Owner**. The authorization code is called back to the third-party app through the **RedirectUrl** provided for the FusionSolar when the third-party app **applies for an OAuth 2.0 client**.
redirect_uri	String	**RedirectUrl** provided by the third-party app when it **applies for an OAuth 2.0 client**.
client_secret	String	**client-secret** returned by the FusionSolar to a third-party app when the third-party app **applies for an OAuth 2.0 client**.
grant_type	String	The value is fixed at **authorization_code**. When **grant_type** is set to **authorization_code**, the current API can obtain the open API access token and refresh token based on the authorization code.

The following table describes the request parameters for obtaining the access token using the refresh token.

Parameter	Data Type	Description
client_id	String	Nine-digit **client-id** returned by the FusionSolar to a third-party app when the third-party app **applies for an OAuth 2.0 client**.
refresh_token	String	Refresh token obtained by a third-party app using an authorization code.
client_secret	String	**client-secret** returned by the FusionSolar to a third-party app when the third-party app **applies for an OAuth 2.0 client**.
grant_type	String	The value is fixed at **refresh_token**, which indicates that the current API can use the refresh token to obtain the access token.

Response Packet of the API for Obtaining Tokens

The following table lists the parameters in the response packet for obtaining the access token and refresh token using the authorization code.

Parameter	Data Type	Description
access_token	String	Access token obtained after the third-party app obtains the owner's authorization. It can be used as the access token for accessing northbound open APIs.
refresh_token	String	Refresh token obtained after the third-party app obtains the owner's authorization. If the access token expires, you can obtain a new one using the refresh token.
expires_in	Integer	Remaining validity period of the access token, in seconds.
scope	String	Permission group of the token. If there are multiple permission groups, separate them with spaces ('').
token_type	String	Type of the returned access token. The value is fixed at **Bearer**.

The following table describes the parameters in the response packet for obtaining the access token using the refresh token.

Parameter	Data Type	Description
access_token	String	New access token obtained by the third-party app using the refresh token. It can be used as the access token for accessing northbound open APIs.
refresh_token	String	Refresh token obtained after the third-party app obtains the owner's authorization. If the access token expires, you can obtain a new one using the refresh token.
expires_in	Integer	Remaining validity period of the access token, in seconds.
scope	String	Permission group of the token. If there are multiple permission groups, separate them with spaces ('').
token_type	String	Type of the returned access token. The value is fixed at **Bearer**.

Example Request and Response for Obtaining the Access Token and Refresh Token Using the Authorization Code

Example Request

Assume that the parameters of the OAuth 2.0 client created by a third-party app on the FusionSolar are as follows: **RedirectUrl** = **https://www.example.com/cn/**,

client_id = **123456789**, **client_secret** = **A******A**, **scope** = **pvms.openapi.basic** and **pvms.openapi.control**. The request example for obtaining the access token and refresh token using the authorization codes is as follows:

```
POST /rest/dp/uidm/oauth2/v1/token HTTP/1.1
Host: oauth2.fusionsolar.huawei.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=B******B&
client_id=123456789&
client_secret=A******A&
redirect_uri=https://www.example.com/cn/
```

Response Example

After the owner authorizes a third-party app, assume that the parameters obtained by the third-party app using the authorization code are as follows: **AT** = **a******t**, **RT** = **r******t**, **scope** = **pvms.openapi.basic** and **pvms.openapi.control**. The response example for obtaining the access token and refresh token using the authorization code is as follows:

```
HTTP/1.1 200 OK

{
  "access_token": "a******t",
  "refresh_token":"r******t",
  "expires_in": 3600,
  "scope": "pvms.openapi.basic pvms.openapi.control",
  "token_type": "Bearer"
}
```

Example Request and Response for Using a Refresh Token to Obtain an Access Token

Example Request

Assume that the refresh token obtained by the third-party app using the authorization code in the **preceding procedure** is **r******t**. An example request for obtaining the access token using the refresh token is as follows:

```
POST /rest/dp/uidm/oauth2/v1/token HTTP/1.1
Host: oauth2.fusionsolar.huawei.com
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&
refresh_token=r******t&
client_id=123456789&
client_secret=A******A
```

Response Example

If the third-party app uses the refresh token **r******t** to obtain a new access token and the obtained new access token is **b******b**, an example response for obtaining the access token using the refresh token is as follows:

```
HTTP/1.1 200 OK

{
  "access_token": "b******b",
  "refresh_token":"r******t",
  "expires_in": 3600,
  "scope": "pvms.openapi.basic pvms.openapi.control",
  "token_type": "Bearer"
}
```

Handling Request Exceptions of the API for Obtaining Tokens

The following is an example of an exception when the **grant_type** parameter that is not supported is used to access the API for obtaining tokens:

```
HTTP/1.1 400
{
  "error_description": "unsupported_grant_type",
  "error": "1111"
}
```

For more exception examples and troubleshooting methods, see **8.1.2 Handling the Exception Returned by Calling the Token Obtaining API**.

⚠️ **CAUTION**

After the owner initiates authorization on the third-party app client, the third-party app obtains the authorization tokens (access token and refresh token) through the OAuth 2.0 authorization process. When an owner initiates incremental authorization through a third-party app client, the old authorization tokens automatically become invalid after the third-party app successfully obtains new authorization tokens. It is recommended that **an owner performs incremental authorization on third-party apps through the FusionSolar app**.

3.1.3 Calling Open APIs by Third-Party Apps

API Access

https://Domain name of the management system/specific API name+access request parameters

Access Process

After adapting to the OAuth Connect mode, the third-party app obtains the owner's authorization and the access token. The access token must be carried when the app accesses specific data. The third-party app calls APIs to access the specific data as required in the following process.
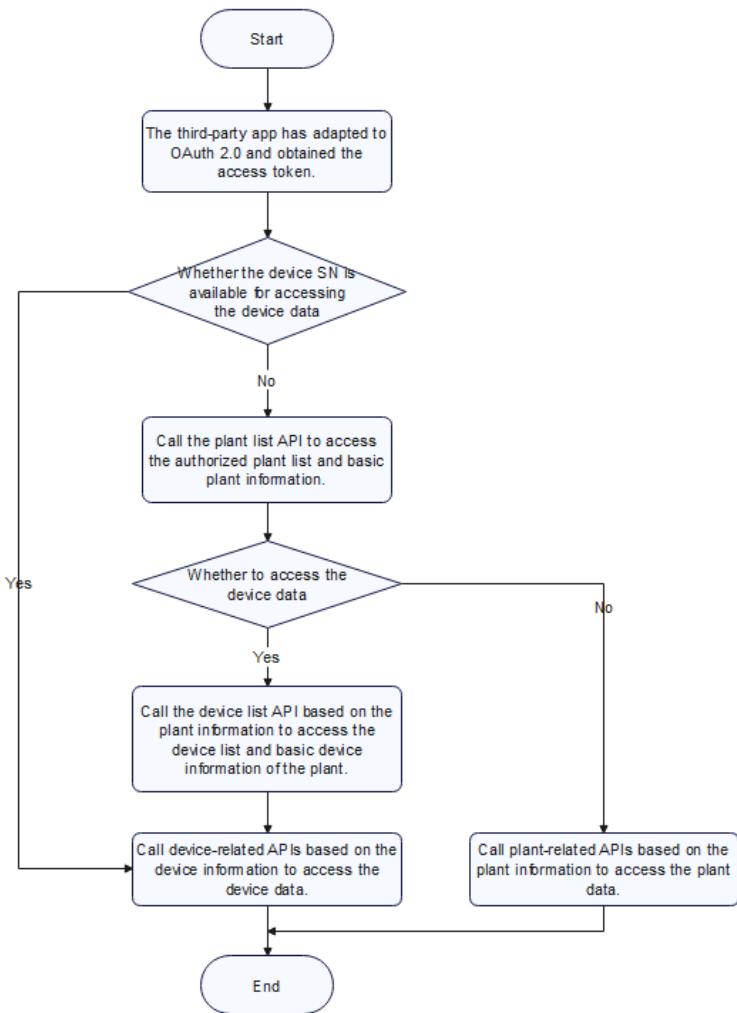
Access Example

The following uses the **Plant List API** as an example to describe how to access FusionSolar open APIs in OAuth Connect mode. After obtaining the access token, the third-party app carries it to access the plant list API.

Request example:

POST https://*Domain name of the management system*/thirdData/stations

Authorization: access token obtained by the bearer

Content-Type: application/json

Body:

```
{
  "pageNo": 1,
  "gridConnectedStartTime":1664718569000,
  "gridConnectedEndTime":1667396969000
}
```

Response examples:

Example 1: An error code is returned.

```
{
 "success": false,
 "data": null,
 "failCode": 20605,
 "message": "The time cannot be a negative number."
}
```

Example 2: The plant list data is returned.

```
{
 "success": true,
 "data": {
  "list": [
    {
      "plantCode": "NE=12345678",
      "plantName": "NMplant1",
      "plantAddress": null,
      "longitude": null,
      "latitude": null,
      "capacity": 146.5,
      "contactPerson": "",
      "contactMethod": "",
      "gridConnectionDate": "2022-11-21T16:23:00+08:00"
    },
    {
      "plantCode": "NE=23456789",
      "plantName": "plant2",
      "plantAddress": null,
      "longitude": null,
      "latitude": null,
      "capacity": 123.3,
      "contactPerson": "",
      "contactMethod": "",
      "gridConnectionDate": "2022-11-21T16:30:28-12:00"
    }
  ],
  "pageCount": 1,
  "pageNo": 1,
  "pageSize": 100,
  "total": 2
 },
 "failCode": 0,
 "message": "get plant list success"
}
```

Example 3: The access token is invalid and cannot be used to parse the authorization information. The following error information is displayed.

```
{
 "success": false,
 "failCode": 305,
 "message": "INVALID_CREDENTIAL"
}
```

Example 4: If the access token permission is insufficient, the following error information is displayed.

```
{
 "data": null,
 "failCode": 401,
 "message": "Invalid access to current interface!",
 "params": {
 },
 "success": false
}
```

Examples 1 and 2 show access to the **Plant List API** in OAuth Connect mode. The returned information varies according to the API internal services. Examples 3 and 4 indicate that the access token is invalid and the access token permission is

insufficient, respectively. For details about how to handle these exceptions, see **8.1.3 Why Does the Northbound API Return Error Code 305?** and **8.1.4 Why Does the Northbound API Return Error Code 401?**

NOTICE

The following figure shows an example of the access token carried in the request header of a northbound open API. (The value of the **Authorization** field consists of **Bearer** and **Access Token**. The **Plant List API** is used as an example.)

In addition to the exception information in **Access Example**, for details about other response information, see the response packet and response example of each open API.

3.1.4 Incremental Authorization or Authorization Revocation

Log in to the FusionSolar app as an owner, choose **Me** > **Settings** > **Account security** > **Authorization Management**, and select the third-party app that requires incremental authorization or authorization revocation. Then tap the button on the right of **Basic APIs** or **Control APIs** in the **Authorization Content** area for incremental authorization or tap **Revoke Authorization** for authorization revocation, as shown in the following figure.
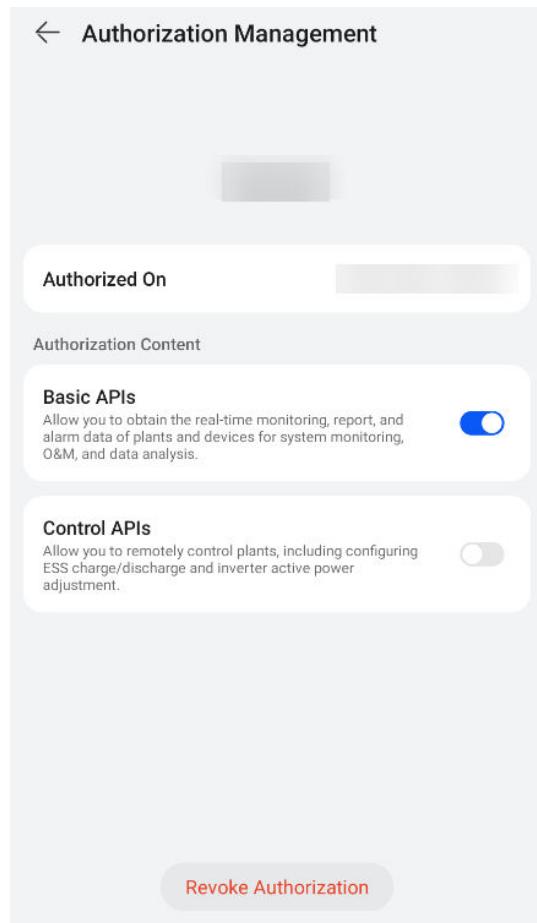
3.2 API Account Access

To use an API account to access northbound open APIs, the company administrator needs to create an API account, obtain the owner's authorization, bind the API account to the plants of the company, and use the API account to call APIs described in **3.2.2.1 Login API** to obtain a token for identity authentication of accessing other APIs.

Communication Between a Third-party System and the SmartPVMS

Figure 3-1 Communication between a third-party system and the SmartPVMS
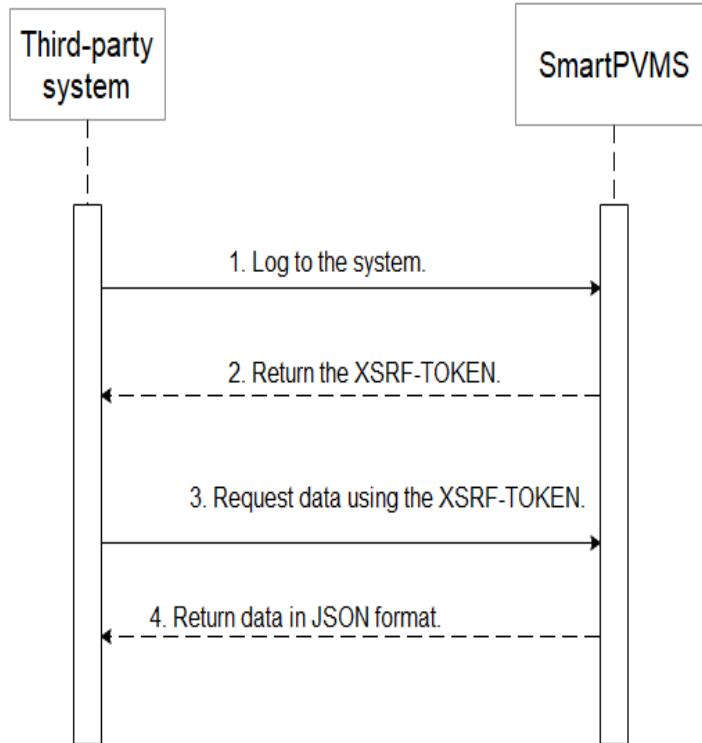
> **NOTE**
>
> 1. After the third-party system information is configured on the management system, use the API account name and password to log in to the management system from the third-party system.
>
> 2. After successful login, send requests to obtain data.
>
> 3. XSRF-TOKEN is a cross-site request token. After a user logs in to the system using the API account name and password, the system returns this token to the user. If the user adds the token to a subsequent request, the request is initiated by a logged-in API account.

3.2.1 Obtaining an Account

The permission to access northbound APIs is created by the company administrator. The procedure is as follows:

Step 1 Choose **System** > **Company Management** > **Northbound Management** from the main menu.
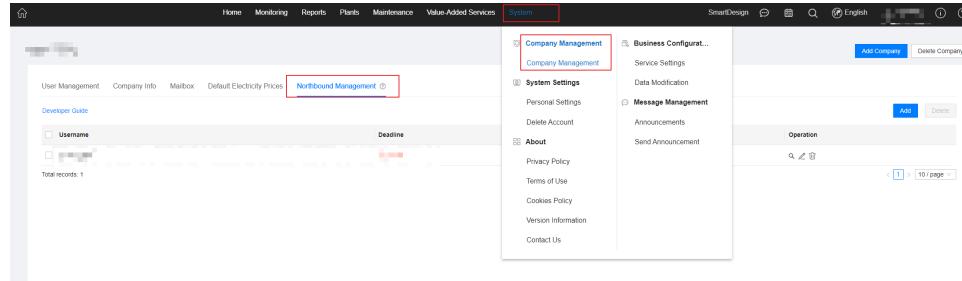
Step 2 Click **Add**. In the dialog box that is displayed, set basic information, such as
Associated account, **Username**, and **Password**.
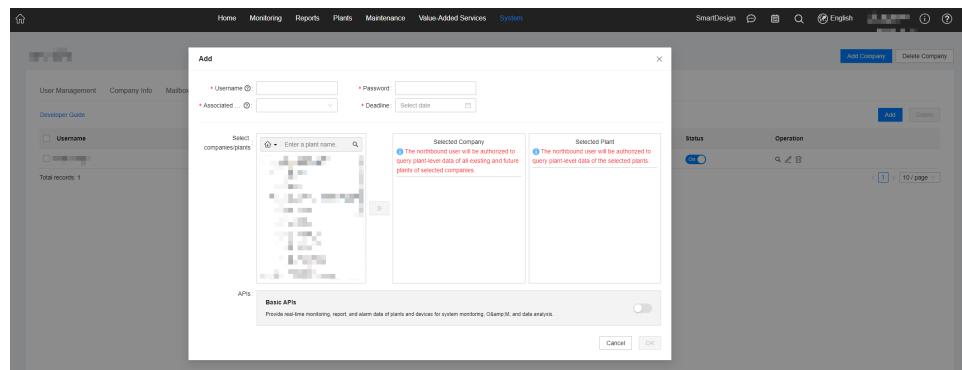
Step 3 If you enable **Basic APIs** in the **APIs** area, the API account has the permission to
access basic API data of northbound open APIs.

📖 **NOTE**

To select all plants of a company, select the company.

Step 4 **Optional:** Select the company or plant for which you want to grant permission.
After the binding, the API account has permissions of the selected company or
plant.

Step 5 Click **OK** to save the settings.

----End

3.2.2 Authentication API

An account is used to call the login API to obtain a token for identity
authentication when accessing other data query APIs and control APIs.

3.2.2.1 Login API

API Description

- Before obtaining data, the login API must be called to obtain the XSRF-
  TOKEN. The validity period of XSRF-TOKEN is 30 minutes.

- If the XSRF-TOKEN does not expire, it can be reused. If the XSRF-TOKEN has
  expired, the login API needs to be called again to obtain a new XSRF-TOKEN.

- After this API is called to log in to the system, XSRF-TOKEN is returned in the
  response header.

- A new token is generated each time this API is called, and the previously
obtained token becomes invalid.

Request URL

https://Domain name of the management system/thirdData/login

Request Mode

HTTP method: POST

Access Restrictions

If an API account enters incorrect passwords for five consecutive times within 10
minutes, the API account will be locked out for 30 minutes.

Maximum number of API calls for each API account: five times every 10 minutes.

If the access frequency exceeds the limit, the API returns error code 407.

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
userName	API account name	String	Mandatory
systemCode	Password	String	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	currentTime	Current system time, in milliseconds	Long	-

Parameter	Description	Data Type	Remarks
message	Optional message	String	If the login is successful, the document link of the current version is returned.
data	Returned data	Object	-

Example

Request example:

```
{
    "userName":"*******",
    "systemCode":"*******"
}
```

Response example:

Example 1: successful login

```
{
    "success":true,
    "data":null,
    "failCode":0,
    "params":null,
    "message":null
}
```

Example 2: failed login

```
{
    "data":null,
    "failCode":20001,
    "message":"",
    "params":{
        "currentTime":1593777870514
    },
    "success":false
}
```

NOTICE

The header of the login success response contains the XSRF-TOKEN that must be retained. In subsequent data API requests, this parameter and its value must be added to the request header and sent to the management system.

Login example:

The following are examples of the XSRF-TOKEN returned after a successful login. You can obtain the XSRF-TOKEN using either of the following methods. The first one is recommended.

The following is an example compatible with earlier versions.

The following figure shows an example of the XSRF-TOKEN carried in the request header of the data API.

3.2.2.2 Logout API

API Description

If you want the XSRF-TOKEN to expire immediately, you can call this API.

Request URL

https://Domain name of the management system/thirdData/logout

Request Mode

HTTP method: POST

Access Restrictions

Maximum number of API calls for each API account: five times every 10 minutes.

You are advised to call this API only when necessary.

If the access frequency exceeds the limit, the API returns error code 407.

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
xsrfToken	XSRF-TOKEN returned in the response header after a successful login through the login API.	String	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data		Returned data	Object	-

Example

Request example:

```
{
    "xsrfToken":"x-
apepjy1fpd2ptete1f7zuqimep7wuqen9hkb3xaourelbyrx9jio7s09hgk6ca2mdlksjdglasdhjaklsdfhhdsahwedyuio
qwehjkd"
}
```

Response example:

Example 1: successful logout

```
{
    "success":true,
    "data":null,
    "failCode":0,
    "params":{
```

```
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: failed logout

```
{
    "data":null,
    "success":false,
    "failCode":20004,
    "params":{
        "currentTime":1503046597854
    },
    "message":null
}
```

📖 NOTE

Logout example:

3.2.3 Calling an API

API Access

https://Domain name of the management system/specific API name+access request parameters

Access Process

Log in to the API on the third-party system and obtain the xsrf-token used for identity authentication. The xsrf-token must be carried when specific data is accessed. The third-party app calls APIs to access the specific data as required in the following process.
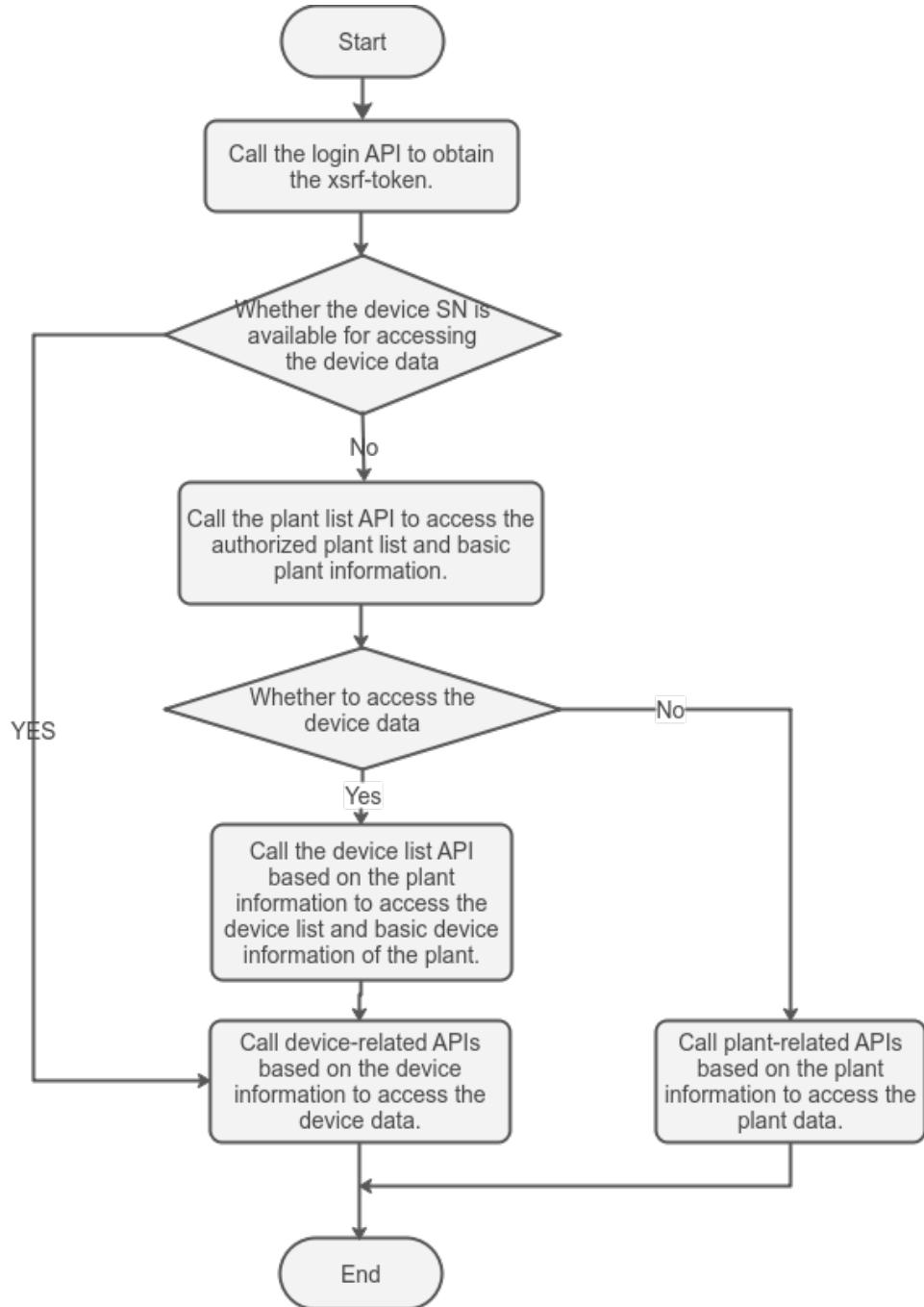
Access Example

Step 1 Create an API account on the management system and bind the account to a PV plant. For details, see **3.2.1 Obtaining an Account**.

Step 2 Obtain the login credential xsrf-token by referring to **3.2.2.1 Login API**. The xsrf-token can be reused if it is valid. For details about the validity period of the xsrf-token, see **API Description**.

Step 3 Content-Type: application/json

Enter the login credential xsrf-token in the request header to access the corresponding API to obtain data.

----End

4 Flow Control Description

The system provides the API flow control mechanism to prevent system performance deterioration caused by improper API invoking. If the maximum flow volume is exceeded, this API cannot be called and the error code 407 is returned.

NOTICE

The flow control mechanism may be modified with the system evolution in the future without notice. Obtain the latest API document to view the API flow control mechanism.

When the API flow is limited, the system displays the reason for the flow control through the error code (failCode). Wait until the API returns a normal response and call the API again.

- Error code 407: The number of API calls of a single user exceeds the upper limit. For details about the limits on API calls, see "Access Restrictions" in the corresponding API description. When this error code is returned by an API, lower your frequency of calls to this API until the frequency drops to the allowed range.

- Error code 403/429: When a large number of users initiate calls to the same API, the total number of calls to the API exceeds the upper limit at the system level. If this error message is received, wait for 1 minute and try again. If the error persists, wait for a longer time and try again. You can also minimize the number of calls.

Northbound open APIs support the following access modes. For details about flow control in OAuth Connect mode, see **4.1 Flow Control Policy in OAuth Connect Mode**. For details about flow control using an API account, see **4.2 Flow Control Using the API Account**.

4.1 Flow Control Policy in OAuth Connect Mode

The API flow control is accumulated based on the API category. Call APIs properly based on the flow control description.

API Category	Flow Control Description
Basic APIs	1000 times/day/owner. The number of flow control times is counted independently for each owner. Example: ● If an owner authorizes a third-party app, the third-party app can access the owner's resources through the API for up to 1000 times a day. ● If 100 owners authorize a third-party app, the third-party app can access the resources of each owner through the API for up to 1000 times a day.
Control APIs	100 times/day/owner. The number of flow control times is counted independently for each owner. Example: ● If an owner authorizes a third-party app, the third-party app can access the owner's resources through the API for up to 100 times a day. ● If 100 owners authorize a third-party app, the third-party app can access the resources of each owner through the API for up to 100 times a day.

4.2 Flow Control Using the API Account

The following table describes the flow control of unrestricted northbound APIs. Call APIs properly based on the flow control description.

API Category	API Label	API Name	Flow Control Description
Basic APIs	Basic data	Plant list	Maximum number of API calls for each API account per day = Roundup (Number of plants/100) x 10 + 24 Example: ● If an API account manages 20 plants, the maximum number of API calls per day = Roundup (20/100) x 10 + 24 = 1 x 10 + 24 = 34. ● If an API account manages 120 plants, the maximum number of API calls per day = Roundup (120/100) x 10 + 24 = 2 x 10 + 24 = 44.

API Category	API Label	API Name	Flow Control Description
		Device list	Flow control is performed based on the number of plants managed by API accounts. Maximum number of API calls for each API account per day = Roundup (Number of plants/100) + 24 Example: ● If an API account manages 20 plants, the maximum number of API calls per day = Roundup (20/100) + 24 = 1 + 24 = 25. ● If an API account manages 120 plants, the maximum number of API calls per day = Roundup (120/100) + 24 = 2 + 24 = 26.
	Monitoring data	Real-time plant data	Flow control is performed based on the number of plants managed by API accounts. Maximum number of API calls for each API account every 5 minutes = Roundup (Number of plants/100) Example: ● If an API account manages 20 plants: Maximum number of API calls every 5 minutes = Roundup (20/100) = 1 ● If an API account manages 120 plants: Maximum number of API calls for each API account every five minutes = Roundup (120/100) = 2

API Category	API Label	API Name	Flow Control Description
		Real-time device data	Flow control is performed based on the number of devices of each type managed by an API account. Maximum number of API calls for each API account every 5 minutes = ∑ Roundup (Number of devices of each type/100). Example: ● If an API account manages 20 inverters and 20 meters, the maximum number of API calls every 5 minutes is calculated as follows: Inverters: Roundup (20/100) = 1 Meters: Roundup (20/100) = 1 Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 1 + 1 = 2 ● If an API account manages 120 inverters and 120 meters, the maximum number of API calls every 5 minutes is calculated as follows: Inverters: Roundup (120/100) = 2 Meters: Roundup (120/100) = 2 Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 2 + 2 = 4

API Category	API Label	API Name	Flow Control Description
		Historical device data	Flow control is performed based on the number of devices of each type managed by an API account. Maximum number of API calls for each API account per day = ∑ Roundup (Number of devices of each type/10) + 24.
			Example:
			● If an API account manages 20 inverters and 20 meters, the maximum number of API calls per day is calculated as follows: Inverters: Roundup (20/10) = 2
			Meters: Roundup (20/10) = 2
			Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 2 + 2 + 24 = 28
			● If an API account manages 120 inverters and 120 meters, the maximum number of API calls per day is calculated as follows: Inverters: Roundup (120/10) = 12
			Meters: Roundup (120/10) = 12
			Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 12 + 12 + 24 = 48

API Category	API Label	API Name	Flow Control Description
	Alarm data	Querying active alarms	Flow control is performed based on the number of plants managed by API accounts and the number of devices of each type.
			Maximum number of API calls for each API account every 30 minutes = MAX (Roundup (Number of plants/100), ∑ Roundup (Number of devices of each type/100))
			Example:
			● If an API account manages 20 plants, 20 inverters, and 20 meters, the maximum number of API calls every 30 minutes is calculated as follows:
			Roundup (Number of plants/100) = Roundup (20/100) = 1
			∑ Roundup (Number of devices of each type/100) = Number of inverters + Number of meters = Roundup (20/100) + Roundup (20/100) = 1 + 1 = 2
			Total: MAX (1,2) = 2
			● If an API account manages 120 plants, 120 inverters, and 120 meters, the maximum number of API calls every 30 minutes is calculated as follows:
			Roundup (Number of plants/100) = Roundup (120/100) = 2
			∑Roundup (Number of devices of each type/100) = Number of inverters + Number of meters = Roundup (120/100) + Roundup (120/100) = 2 + 2 = 4
			Total: MAX (2,4) = 4
	Report data	Hourly plant data	Flow control is performed based on the number of plants managed by API accounts. Maximum number of API calls for each API account per day = Roundup (Number of plants/100) + 24
			Example:
			● If an API account manages 20 plants, the maximum number of API calls per day = Roundup (20/100) + 24 = 1 + 24 = 25.
			● If an API account manages 120 plants, the maximum number of API calls per day = Roundup (120/100) + 24 = 2 + 24 = 26.

API Category	API Label	API Name	Flow Control Description
		Daily plant data	Flow control is performed based on the number of plants managed by API accounts. Maximum number of API calls for each API account per day = Roundup (Number of plants/100) + 24 Example: ● If an API account manages 20 plants, the maximum number of API calls per day = Roundup (20/100) + 24 = 1 + 24 = 25. ● If an API account manages 120 plants, the maximum number of API calls per day = Roundup (120/100) + 24 = 2 + 24 = 26.
		Monthly plant data	Flow control is performed based on the number of plants managed by API accounts. Maximum number of API calls for each API account per day = Roundup (Number of plants/100) + 24 Example: ● If an API account manages 20 plants, the maximum number of API calls per day = Roundup (20/100) + 24 = 1 + 24 = 25. ● If an API account manages 120 plants, the maximum number of API calls per day = Roundup (120/100) + 24 = 2 + 24 = 26.
		Yearly plant data	Flow control is performed based on the number of plants managed by API accounts. Maximum number of API calls for each API account per day = Roundup (Number of plants/100) + 24 Example: ● If an API account manages 20 plants, the maximum number of API calls per day = Roundup (20/100) + 24 = 1 + 24 = 25. ● If an API account manages 120 plants, the maximum number of API calls per day = Roundup (120/100) + 24 = 2 + 24 = 26.

API Cat ego ry	API Label	API Name	Flow Control Description
		Daily device data	Flow control is performed based on the number of devices of each type managed by an API account. Maximum number of API calls for each API account per day = ∑ Roundup (Number of devices of each type/100) + 24. Example: ● If an API account manages 20 inverters and 20 meters, the maximum number of API calls per day is calculated as follows: Inverters: Roundup (20/100) = 1 Meters: Roundup (20/100) = 1 Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 1 + 1 + 24 = 26 ● If an API account manages 120 inverters and 120 meters, the maximum number of API calls per day is calculated as follows: Inverters: Roundup (120/100) = 2 Meters: Roundup (120/100) = 2 Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 2 + 2 + 24 = 28

API Cat ego ry	API Label	API Name	Flow Control Description
		Monthly device data	Flow control is performed based on the number of devices of each type managed by an API account. Maximum number of API calls for each API account per day = ∑ Roundup (Number of devices of each type/100) + 24. Example: • If an API account manages 20 inverters and 20 meters, the maximum number of API calls per day is calculated as follows: Inverters: Roundup (20/100) = 1 Meters: Roundup (20/100) = 1 Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 1 + 1 + 24 = 26 • If an API account manages 120 inverters and 120 meters, the maximum number of API calls per day is calculated as follows: Inverters: Roundup (120/100) = 2 Meters: Roundup (120/100) = 2 Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 2 + 2 + 24 = 28

API Category	API Label	API Name	Flow Control Description
		Yearly device data	Flow control is performed based on the number of devices of each type managed by an API account. Maximum number of API calls for each API account per day = ∑ Roundup (Number of devices of each type/100) + 24. Example: • If an API account manages 20 inverters and 20 meters, the maximum number of API calls per day is calculated as follows: Inverters: Roundup (20/100) = 1 Meters: Roundup (20/100) = 1 Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 1 + 1 + 24 = 26 • If an API account manages 120 inverters and 120 meters, the maximum number of API calls per day is calculated as follows: Inverters: Roundup (120/100) = 2 Meters: Roundup (120/100) = 2 Total: Maximum number of API calls for inverters + Maximum number of API calls for meters = 2 + 2 + 24 = 28
Control APIs	Battery configuration	Delivering battery charge/discharge tasks	Call the API only when necessary to reduce the access frequency. For the same plant, do not call this API repeatedly before a task is complete. Maximum number of API calls for each API account: once per minute.
		Querying battery charge/discharge tasks	Call the API only when necessary to reduce the access frequency. For the same plant, do not call this API repeatedly before a task is complete. Maximum number of API calls for each API account: once per minute.
	Inverter active power control	API for delivering an inverter active power setting task	Call the API only when necessary to reduce the access frequency. For the same plant, do not call this API repeatedly before a task is complete. Maximum number of API calls for each API account: once per minute.

API Category	API Label	API Name	Flow Control Description
		API for querying an inverter active power setting task	Call the API only when necessary to reduce the access frequency. For the same plant, do not call this API repeatedly before a task is complete. Maximum number of API calls for each API account: once per minute.
	Setting the battery working mode	API for Delivering a Task for Setting the Battery Working Mode	Call the API only when necessary to reduce the access frequency. For the same plant, do not call this API repeatedly before a task is complete. Maximum number of API calls for each API account: once per minute.
		API for Querying a Task for Setting the Battery Working Mode	Call the API only when necessary to reduce the access frequency. For the same plant, do not call this API repeatedly before a task is complete. Maximum number of API calls for each API account: once per minute.
	Battery parameter setting	API for Delivering a Task for Setting Battery Parameters	Call the API only when necessary to reduce the access frequency. For the same plant, do not call this API repeatedly before a task is complete. Maximum number of API calls for each API account: once per minute.
		API for Querying a Task for Setting Battery Parameters	Call the API only when necessary to reduce the access frequency. For the same plant, do not call this API repeatedly before a task is complete. Maximum number of API calls for each API account: once per minute.

5 API Reference

5.1 Basic APIs

Allow you to obtain the real-time monitoring, report, and alarm data of plants and devices for system monitoring, O&M, and data analysis.

5.1.1 Basic

5.1.1.1 Plant List API

API Description

This API is used to query the plant list. When the pagination parameters (a maximum of 100 records can be displayed on each page) and grid connection time are transferred (if only the grid connection start time is transferred, the grid connection end time is the current time by default; if only the grid connection end time is transferred, the default grid connection start time is 1970-01-01 08:00:00), the plant list is queried in pages based on the grid connection time. When only the pagination parameter is transferred, the plant list is queried in pages.

> ⚠ **CAUTION**

If no data is queried or only some data is returned when the plant list API is used, rectify the fault by referring to **8.3 Why Is No Data or Only Part of Data Found When I Call a Northbound API for Data Query?**

If the returned plant ID is inconsistent with that displayed on the SmartPVMS, rectify the fault by referring to **8.8 Why Is the Plant ID Returned by the Plant List Inconsistent with That Displayed on the SmartPVMS?**

If the plant IDs returned vary depending on open API users, rectify the fault by referring to **8.9 Why Do the Plant IDs Returned for the Same Plant Vary Depending on Open API Users?**

Request URL

https://*Domain name of the management system*/thirdData/stations

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
pageNo	Page No. of the results	Integer	Mandatory
gridConnectedStartTime	Grid connection start time (ms)	Long	Optional
gridConnectedEndTime	Grid connection end time (ms)	Long	Optional

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode	Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
message	Optional response message	-	-
data	Returned data, which contains the following information:	Map	-
> total	Total number of results	Long	-
> pageCount	Total number of pages	Long	-
> pageNo	Page No. of the results	Integer	-

Parameter	Description	Data Type	Remarks
> pageSize	Number of query results displayed on each page	Integer	-
> list	Plant information list. The plant information is as follows:	List	Plant information
>> plantCode	Plant ID, which uniquely identifies a plant.	String	-
>> plantName	Plant name	String	-
>> plantAddress	Detailed address of the plant	String	-
>> longitude	Plant longitude	Double	-
>> latitude	Plant latitude	Double	-
>> capacity	Total string capacity	Double	kWp
>> contactPerson	Plant contact	String	-
>> contactMethod	Contact information of the plant contact, such as the mobile phone number or email address	String	-
>> gridConnectionDate	Grid connection time of the plant, including the time zone	String	2020-02-06T00:00:00+08:00

Examples

Request example:

```
{
  "pageNo": 1,
  "gridConnectedStartTime":1664718569000,
  "gridConnectedEndTime":1667396969000
}
```

Response examples:

Example 1: An error code is returned.

```
{
  "success": false,
  "data": null,
  "failCode": 20605,
  "message": "The time cannot be a negative number."
}
```

Example 2: The plant list data is returned.

```
{
  "success": true,
```

```
  "data": {
    "list": [
      {
        "plantCode": "NE=12345678",
        "plantName": "NMplant1",
        "plantAddress": null,
        "longitude": null,
        "latitude": null,
        "capacity": 146.5,
        "contactPerson": "",
        "contactMethod": "",
        "gridConnectionDate": "2022-11-21T16:23:00+08:00"
      },
      {
        "plantCode": "NE=23456789",
        "plantName": "plant2",
        "plantAddress": null,
        "longitude": null,
        "latitude": null,
        "capacity": 123.3,
        "contactPerson": "",
        "contactMethod": "",
        "gridConnectionDate": "2022-11-21T16:30:28-12:00"
      }
    ],
    "pageCount": 1,
    "pageNo": 1,
    "pageSize": 100,
    "total": 2
  },
  "failCode": 0,
  "message": "get plant list success"
}
```

5.1.1.2 Device List API

API Description

This API is used to obtain basic device information. Before invoking other APIs to obtain device data, you need to call this API to obtain the device ID.

When you query devices by plant ID set, devices of a maximum of 100 plants can be queried at a time.

⚠️ CAUTION

If no data is queried or only some data is returned when the device list API is used, rectify the fault by referring to **8.3 Why Is No Data or Only Part of Data Found When I Call a Northbound API for Data Query?**

Request URL

https://*Domain name of the management system*/thirdData/getDevList

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
stationCodes	Plant ID list. Multiple plant IDs are separated by commas (,). The plant IDs are obtained from **plantCode** in **5.1.1.1 Plant List API**.	String	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag **true**: The request succeeded. **false**: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For definitions of other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	stationCodes	Plant ID list in the request parameter	String	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data	Parameters	Returned data. The data contains the object parameter list of each device.	List	-
	id	Device ID	Long	-
	devDn	Unique device ID in the system	String	
	devName	Device name	String	-
	stationCode	Plant ID	String	-

Parameter		Description	Data Type	Remarks
	esnCode	Device SN	String	-
	devTypeId	Device type ID The following device types are supported: 1: string inverter 2: SmartLogger 8: STS 10: EMI 13: protocol converter 16: general device 17: grid meter 22: PID 37: Pinnet data logger 38: residential inverter 39: battery 40: backup box 41: ESS 45: PLC 46: optimizer 47: power sensor 62: Dongle 63: distributed SmartLogger 70: safety box 60001: mains 60003: genset 60043: SSU group 60044: SSU 60092: power converter 60014: lithium battery rack 60010: AC output power distribution 23070: EMMA	Integer	-
	softwareVersion	Software version	String	-
	optimizerNumber	Quantity of optimizers	Integer	
	invType	Inverter model (only applicable to inverters)	String	-
	longitude	Longitude	Double	-

Parameter		Description	Data Type	Remarks
	latitude	Latitude	Double	-

Examples

Request example:

```
{
    "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5"
}
```

Response examples:

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20009,
    "params":{
        "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: The device list is returned.

```
{
    "success":true,
    "data":[
        {
            "id":-214543629611879,
            "devDn":"NE=45112560",
            "devName":"5fbfk4",
            "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
            "esnCode":"5fbfk4",
            "devTypeId":1,
            "softwareVersion":"V100R001PC666",
            "invType":"SUN2000-17KTL",
            "longitude":null,
            "latitude":null
        },
        {
            "id":-214091680973855,
            "devDn":"NE=4511256",
            "devName":"6fbfk11",
            "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
            "esnCode":"6fbfk11",
            "devTypeId":1,
            "softwareVersion":"V100R001PC666",
            "invType":"SUN2000-17KTL",
            "longitude":null,
            "latitude":null
        }
    ],
    "failCode":0,
    "params":{
        "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
        "currentTime":1503046597854
    },
    "message":null
}
```

📖 **NOTE**

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.2 Monitoring

5.1.2.1 Real-Time Plant Data API

API Description

This API is used to obtain real-time plant data by plant ID set. Data of a maximum of 100 plants can be queried at a time.

For details about the data list that can be queried through this API, see the real-time plant data list below.

Request URL

https://Domain name of the management system/thirdData/getStationRealKpi

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
stationCodes	Plant ID list. Multiple plant IDs are separated by commas (,). The plant IDs are obtained from **plantCode** in **5.1.1.1 Plant List API**.	String	Mandatory

Response Packet

Parameter		Description	Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	stationCodes	Plant ID list in the request parameter	String	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data	Parameters	Returned data. The data contains the real-time data object list of each plant.	List	-
	stationCode	Plant ID	String	-
	dataItemMap	Content of each data item, which is returned in the key-value format. For details about the data item list, see the real-time plant data list below.	Map	-

Real-Time Plant Data List

Key	Name	Unit	Return Value Type
day_power	Yield today	kWh	Double
month_power	Yield this month	kWh	Double
total_power	Total yield	kWh	Double
day_income	Revenue today	The currency specified in the management system	Double
total_income	Total revenue	The currency specified in the management system	Double
real_health_state	Plant health status The following plant health states are supported: 1: disconnected 2: faulty 3: healthy	None	Integer

Example

Request example:

```
{
    "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5"
}
```

Response example:

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20009,
    "params":{
        "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: Real-time plant data is returned.

```
{
    "success":true,
    "data":[
        {
            "dataItemMap":{
```

```
            "real_health_state":"3",
            "day_power":"10000",
            "total_power":"900.000",
            "day_income":"0.000",
            "month_power":"900.000",
            "total_income":"2088.000"
        },
        "stationCode":"BA4372D08E014822AB065017416F254C"
    },
    {
        "dataItemMap":{
            "real_health_state":"1",
            "day_power":"16770.000",
            "total_power":"35100.000",
            "day_income":"26832.000",
            "month_power":"35100.000",
            "total_income":"61152.000"
        },
        "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5"
    }
],
"failCode":0,
"params":{
    "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
    "currentTime":1503046597854
},
"message":null
}
```

📖 NOTE

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.2.2 Real-Time Device Data API

API Description

This API is used to obtain real-time device data by device type and device ID set. The data varies depending on device types. Data of a maximum of 100 devices of the same type can be queried at a time.

For details about the data list that can be queried through this API, see the real-time device data list below.

Request URL

https://*Domain name of the management system*/thirdData/getDevRealKpi

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
devIds	List of device IDs. The device IDs are obtained from **id** in **5.1.1.2 Device List API**. Use commas (,) to separate multiple device IDs. Either **sns** or **devIds** must be set.	String	Optional
sns	Device SN list. Multiple device SNs are separated by commas (,). Either **sns** or **devIds** must be set.	String	Optional

Parameter	Description	Data Type	Mandatory/ Optional
devTypeId	Device type ID. The values of **devTypeId** obtained in **5.1.1.2 Device List API** are used. The following device types are supported: 1: string inverter 10: EMI 17: grid meter 38: residential inverter 39: battery 41: ESS 47: power sensor 60001: mains 60003: genset 60043: SSU group 60044: SSU 60092: power converter 60014: lithium battery rack 60010: AC output power distribution	Integer	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	devIds	Device ID list in the request parameter	String	-
	sns	Device SN list in the request parameter	String	-

Parameter		Description	Data Type	Remarks
	devTypeId	Device type ID in the request parameter	Integer	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data	Parameters	Returned data. The data contains the real-time data object list of each device.	List	-
	devId	Device ID	Long	-
	sn	Device SN	String	
	dataItemMap	Content of data items, which are returned in the key-value format. Content of each data item, which is returned in the key-value format. The data item content varies depending on the device type. For details about the data item list, see the real-time device data list below.	Map	Real-time device data

Real-Time Device Data List

Device Type	Key	Name	Unit	Return Value Type	Remarks
ID: 1 String inverter	inverter_state	Inverter state. For details, see **Table 5-1**.	None	Double	
	ab_u	A-B line voltage of grid	V	Double	
	bc_u	B-C line voltage of grid	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	ca_u	C-A line voltage of grid	V	Double	
	a_u	Phase A voltage	V	Double	
	b_u	Phase B voltage	V	Double	
	c_u	Phase C voltage	V	Double	
	a_i	Phase A current of grid	A	Double	
	b_i	Phase B current of grid	A	Double	
	c_i	Phase C current of grid	A	Double	
	efficiency	Inverter conversion efficiency (manufacturer)	%	Double	
	temperature	Internal temperature	°C	Double	
	power_factor	Power factor	None	Double	
	elec_freq	Grid frequency	Hz	Double	
	active_power	Active power	kW	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	reactive_power	Output reactive power	kVar	Double	
	day_cap	Yield today	kWh	Double	
	mppt_power	MPPT total input power	kW	Double	
	pv1_u	PV1 input voltage	V	Double	
	pv2_u	PV2 input voltage	V	Double	
	pv3_u	PV3 input voltage	V	Double	
	pv4_u	PV4 input voltage	V	Double	
	pv5_u	PV5 input voltage	V	Double	
	pv6_u	PV6 input voltage	V	Double	
	pv7_u	PV7 input voltage	V	Double	
	pv8_u	PV8 input voltage	V	Double	
	pv9_u	PV9 input voltage	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv10_u	PV10 input voltage	V	Double	
	pv11_u	PV11 input voltage	V	Double	
	pv12_u	PV12 input voltage	V	Double	
	pv13_u	PV13 input voltage	V	Double	
	pv14_u	PV14 input voltage	V	Double	
	pv15_u	PV15 input voltage	V	Double	
	pv16_u	PV16 input voltage	V	Double	
	pv17_u	PV17 input voltage	V	Double	
	pv18_u	PV18 input voltage	V	Double	
	pv19_u	PV19 input voltage	V	Double	
	pv20_u	PV20 input voltage	V	Double	
	pv21_u	PV21 input voltage	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv22_u	PV22 input voltage	V	Double	
	pv23_u	PV23 input voltage	V	Double	
	pv24_u	PV24 input voltage	V	Double	
	pv25_u	PV25 input voltage	V	Double	
	pv26_u	PV26 input voltage	V	Double	
	pv27_u	PV27 input voltage	V	Double	
	pv28_u	PV28 input voltage	V	Double	
	pv1_i	PV1 input current	A	Double	
	pv2_i	PV2 input current	A	Double	
	pv3_i	PV3 input current	A	Double	
	pv4_i	PV4 input current	A	Double	
	pv5_i	PV5 input current	A	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv6_i	PV6 input current	A	Double	
	pv7_i	PV7 input current	A	Double	
	pv8_i	PV8 input current	A	Double	
	pv9_i	PV9 input current	A	Double	
	pv10_i	PV10 input current	A	Double	
	pv11_i	PV11 input current	A	Double	
	pv12_i	PV12 input current	A	Double	
	pv13_i	PV13 input current	A	Double	
	pv14_i	PV14 input current	A	Double	
	pv15_i	PV15 input current	A	Double	
	pv16_i	PV16 input current	A	Double	
	pv17_i	PV17 input current	A	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv18_i	PV18 input current	A	Double	
	pv19_i	PV19 input current	A	Double	
	pv20_i	PV20 input current	A	Double	
	pv21_i	PV21 input current	A	Double	
	pv22_i	PV22 input current	A	Double	
	pv23_i	PV23 input current	A	Double	
	pv24_i	PV24 input current	A	Double	
	pv25_i	PV25 input current	A	Double	
	pv26_i	PV26 input current	A	Double	
	pv27_i	PV27 input current	A	Double	
	pv28_i	PV28 input current	A	Double	
	total_cap	Total yield	kWh	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	open_time	Inverter startup time	ms	Double	
	close_time	Inverter shutdown time	ms	Double	
	mppt_total_cap	Total DC input energy	kWh	Double	
	mppt_1_cap	MPPT 1 DC total yield	kWh	Double	
	mppt_2_cap	MPPT 2 DC total yield	kWh	Double	
	mppt_3_cap	MPPT 3 DC total yield	kWh	Double	
	mppt_4_cap	MPPT 4 DC total yield	kWh	Double	
	mppt_5_cap	MPPT 5 DC total yield	kWh	Double	
	mppt_6_cap	MPPT 6 DC total yield	kWh	Double	
	mppt_7_cap	MPPT 7 DC total yield	kWh	Double	
	mppt_8_cap	MPPT 8 DC total yield	kWh	Double	
	mppt_9_cap	MPPT 9 DC total yield	kWh	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	mppt_10_cap	MPPT 10 DC total yield	kWh	Double	
	run_state	State (0: Disconnected 1: Connected)	None	Long	
ID: 38 Residential inverter	inverter_state	Inverter state. For details, see **Table 5-1**.	None	Double	
	ab_u	A-B line voltage of grid	V	Double	
	bc_u	B-C line voltage of grid	V	Double	
	ca_u	C-A line voltage of grid	V	Double	
	a_u	Phase A voltage	V	Double	
	b_u	Phase B voltage	V	Double	
	c_u	Phase C voltage	V	Double	
	a_i	Phase A current of grid	A	Double	
	b_i	Phase B current of grid	A	Double	
	c_i	Phase C current of grid	A	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	efficiency	Inverter conversion efficiency (manufacturer)	%	Double	
	temperature	Internal temperature	°C	Double	
	power_factor	Power factor	None	Double	
	elec_freq	Grid frequency	Hz	Double	
	active_power	Active power	kW	Double	
	reactive_power	Output reactive power	kVar	Double	
	day_cap	Yield today	kWh	Double	
	mppt_power	MPPT total input power	kW	Double	
	pv1_u	PV1 input voltage	V	Double	
	pv2_u	PV2 input voltage	V	Double	
	pv3_u	PV3 input voltage	V	Double	
	pv4_u	PV4 input voltage	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv5_u	PV5 input voltage	V	Double	
	pv6_u	PV6 input voltage	V	Double	
	pv7_u	PV7 input voltage	V	Double	
	pv8_u	PV8 input voltage	V	Double	
	pv1_i	PV1 input current	A	Double	
	pv2_i	PV2 input current	A	Double	
	pv3_i	PV3 input current	A	Double	
	pv4_i	PV4 input current	A	Double	
	pv5_i	PV5 input current	A	Double	
	pv6_i	PV6 input current	A	Double	
	pv7_i	PV7 input current	A	Double	
	pv8_i	PV8 input current	A	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	total_cap	Total yield	kWh	Double	
	open_time	Inverter startup time	ms	Double	
	close_time	Inverter shutdown time	ms	Double	
	mppt_1_cap	MPPT 1 DC total yield	kWh	Double	
	mppt_2_cap	MPPT 2 DC total yield	kWh	Double	
	mppt_3_cap	MPPT 3 DC total yield	kWh	Double	
	mppt_4_cap	MPPT 4 DC total yield	kWh	Double	
	run_state	State (0: Disconnected 1: Connected)	None	Long	
ID: 10 Environmental monitoring instrument (EMI)	temperature	Temperature	°C	Double	
	pv_temperature	PV temperature	°C	Double	
	wind_speed	Wind speed	m/s	Double	
	wind_direction	Wind direction	Degree	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	radiant_total	Daily irradiation	MJ/m^2	Double	
	radiant_line	Irradiance	W/m^2	Double	
	horiz_radiant_line	Horizontal irradiance	W/m^2	Double	Invalid field. The default value is **null**.
	horiz_radiant_total	Horizontal irradiation	MJ/m^2	Double	Invalid field. The default value is **null**.
	run_state	State (0: Disconnected 1: Connected)	None	Long	
ID: 17 Grid meter	ab_u	A-B line voltage of grid	V	Double	
	bc_u	B-C line voltage of grid	V	Double	
	ca_u	C-A line voltage of grid	V	Double	
	a_u	Phase A voltage (AC output)	V	Double	
	b_u	Phase B voltage (AC output)	V	Double	
	c_u	Phase C voltage (AC output)	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	a_i	Phase A current of grid (IA)	A	Double	
	b_i	Phase B current of grid (IB)	A	Double	
	c_i	Phase C current of grid (IC)	A	Double	
	active_power	Active power	kW	Double	
	power_factor	Power factor	None	Double	
	active_cap	Active energy (positive active energy)	kWh	Double	
	reactive_power	Reactive power	kVar	Double	
	reverse_active_cap	Negative active energy	kWh	Double	
	forward_reactive_cap	Positive reactive energy	kWh	Double	
	reverse_reactive_cap	Negative reactive energy	kWh	Double	
	active_power_a	Active power PA	kW	Double	
	active_power_b	Active power PB	kW	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	active_power_c	Active power PC	kW	Double	
	reactive_power_a	Reactive power QA	kVar	Double	Invalid field. The default value is **null**.
	reactive_power_b	Reactive power QB	kVar	Double	Invalid field. The default value is **null**.
	reactive_power_c	Reactive power QC	kVar	Double	Invalid field. The default value is **null**.
	total_apparent_power	Total apparent power	kVA	Double	
	grid_frequency	Grid frequency	Hz	Double	Invalid field. The default value is **null**.
	reverse_active_peak	Negative active energy (peak)	kWh	Double	Invalid field. The default value is **null**.
	reverse_active_power	Negative active energy (shoulder)	kWh	Double	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Ret urn Val ue Typ e	Remarks
	reverse_active_v alley	Negative active energy (off-peak)	kWh	Do ubl e	Invalid field. The default value is **null**.
	reverse_active_t op	Negative active energy (sharp)	kWh	Do ubl e	Invalid field. The default value is **null**.
	positive_active_p eak	Positive active energy (peak)	kWh	Do ubl e	Invalid field. The default value is **null**.
	positive_active_p ower	Positive active energy (shoulder)	kWh	Do ubl e	Invalid field. The default value is **null**.
	positive_active_v alley	Positive active energy (off-peak)	kWh	Do ubl e	Invalid field. The default value is **null**.
	positive_active_t op	Positive active energy (sharp)	kWh	Do ubl e	Invalid field. The default value is **null**.
	reverse_reactive_ peak	Negative reactive energy (peak)	kVar	Do ubl e	Invalid field. The default value is **null**.
	reverse_reactive_ power	Negative reactive energy (shoulder)	kVar	Do ubl e	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Return Value Type	Remarks
	reverse_reactive_valley	Negative reactive energy (off-peak)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_top	Negative reactive energy (sharp)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_peak	Positive reactive energy (peak)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_power	Positive reactive energy (shoulder)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_valley	Positive reactive energy (off-peak)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_top	Positive reactive energy (sharp)	kVar	Double	Invalid field. The default value is **null**.
ID: 47 Power sensor	meter_status	Meter state (0: offline; 1: normal)	None	Double	
	meter_u	Phase A voltage (AC output)	V	Double	
	meter_i	Phase A current of grid (IA)	A	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	active_power	Active power	W	Double	
	reactive_power	Reactive power	Var	Double	
	power_factor	Power factor	None	Double	
	grid_frequency	Grid frequency	Hz	Double	
	active_cap	Active energy (positive active energy)	kWh	Double	
	reverse_active_cap	Negative active energy	kWh	Double	
	run_state	State (0: Disconnected 1: Connected)	None	Long	
	ab_u	A-B line voltage of grid	V	Double	
	bc_u	B-C line voltage of grid	V	Double	
	ca_u	C-A line voltage of grid	V	Double	
	b_u	Phase B voltage (AC output)	V	Double	
	c_u	Phase C voltage (AC output)	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	b_i	Phase B current of grid (IB)	A	Double	
	c_i	Phase C current of grid (IC)	A	Double	
	forward_reactive_cap	Positive reactive energy	kWh	Double	Invalid field. The default value is **null**.
	reverse_reactive_cap	Negative reactive energy	kWh	Double	Invalid field. The default value is **null**.
	active_power_a	Active power PA	kW	Double	
	active_power_b	Active power PB	kW	Double	
	active_power_c	Active power PC	kW	Double	
	reactive_power_a	Reactive power QA	kVar	Double	Invalid field. The default value is **null**.
	reactive_power_b	Reactive power QB	kVar	Double	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Return Value Type	Remarks
	reactive_power_c	Reactive power QC	kVar	Double	Invalid field. The default value is **null**.
	total_apparent_power	Total apparent power	kVA	Double	Invalid field. The default value is **null**.
	reverse_active_peak	Negative active energy (peak)	kWh	Double	Invalid field. The default value is **null**.
	reverse_active_power	Negative active energy (shoulder)	kWh	Double	Invalid field. The default value is **null**.
	reverse_active_valley	Negative active energy (off-peak)	kWh	Double	Invalid field. The default value is **null**.
	reverse_active_top	Negative active energy (sharp)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_peak	Positive active energy (peak)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_power	Positive active energy (shoulder)	kWh	Double	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Return Value Type	Remarks
	positive_active_valley	Positive active energy (off-peak)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_top	Positive active energy (sharp)	kWh	Double	Invalid field. The default value is **null**.
	reverse_reactive_peak	Negative reactive energy (peak)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_power	Negative reactive energy (shoulder)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_valley	Negative reactive energy (off-peak)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_top	Negative reactive energy (sharp)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_peak	Positive reactive energy (peak)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_power	Positive reactive energy (shoulder)	kVar	Double	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Return Value Type	Remarks
	positive_reactive_valley	Positive reactive energy (off-peak)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_top	Positive reactive energy (sharp)	kVar	Double	Invalid field. The default value is **null**.
ID: 39 Residential battery	battery_status	Battery running state (0: offline; 1: standby; 2: running; 3: faulty; 4: hibernating)	None	Double	
	max_charge_power	Maximum charge power	W	Double	
	max_discharge_power	Maximum discharge power	W	Double	
	ch_discharge_power	Charge/ Discharge power	W	Double	
	busbar_u	Battery voltage	V	Double	
	battery_soc	Battery SOC	%	Double	
	battery_soh	Battery SOH (supported only by LG batteries)	None	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	ch_discharge_model	Charge/Discharge mode (0: none; 1: forced charge/discharge; 2: time-of-use price; 3: fixed charge/discharge; 4: automatic charge/discharge; 5: fully fed to grid; 6: TOU; 7: remote scheduling–max. self-consumption; 8: remote scheduling–fully fed to grid; 9: remote scheduling–TOU; 10: EMMA)	None	Double	
	charge_cap	Charged energy	kWh	Double	
	discharge_cap	Discharged energy	kWh	Double	
	run_state	State (0: Disconnected 1: Connected)	None	Long	
ID: 41 C&I and utility ESS	ch_discharge_power	Charge/Discharge power	W	Double	
	battery_soc	Battery SOC	%	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	battery_soh	Battery SOH	None	Double	
	charge_cap	Charged energy	kWh	Double	
	discharge_cap	Discharged energy	kWh	Double	
	run_state	State (0: Disconnected 1: Connected)	None	Long	
ID: 60001 Mains (supported only in the Power-M scenario)	mains_state	Mains status (0: mains unavailable; 1: mains available)	None	Double	
	ac_voltage	AC voltage	V	Double	
	ac_current	AC current	A	Double	
	active_power	Active power	kW	Double	
	ac_frequency	AC frequency	Hz	Double	
	grid_quality_grade	Power grid quality level (0: Unknown; 1: Class 1; 2: Class 2; 3: Class 3; 4: Class 4)	None	Double	
	total_energy_consumption	Total energy consumption	kWh	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	supply_duration_per_total	Total power supply duration	h	Double	
ID: 60003 Genset (supported only in the Power-M scenario)	running_state	Running status (0: unknown; 1: stopped; 2: running)	None	Double	
	output_power	Output power	kW	Double	
	load_rate	Load rate	%	Double	
ID: 60043 SSU group (supported only in the Power-M scenario)	total_output_current	Total output current	A	Double	
	total_output_power	Total output power	kW	Double	
D: 60044 SSU (supported only in the Power-M scenario)	input_voltage	Input voltage	V	Double	
	output_voltage	Output voltage	V	Double	
	output_current	Output current	A	Double	
	on_off_state	Power-on/off status (0: on; 1: off)	None	Double	
ID: 60092 Power converter (supported only in the Power-M scenario)	total_runtime	Total runtime	h	Double	
	pv_input_voltage	PV input voltage	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv_input_current	PV input current	A	Double	
	pv_input_power	PV input power	kW	Double	
	inverter_voltage	Inverter voltage	V	Double	
	inverter_frequency	Inverter frequency	Hz	Double	
	ac_output_voltage	AC output voltage	V	Double	
	ac_output_current	AC output current	A	Double	
	ac_output_frequency	AC output frequency	kW	Double	
	ac_output_apparent_power	AC output apparent power	kVA	Double	
ID: 60014 Lithium battery rack (supported only in the Power-M scenario)	battery_state	Battery status (0: initial power-on; 1: power-off; 2: float charging; 3: boost charging; 4: discharging; 5: charging; 6: testing; 7: hibernation; 8: standby)	None	Double	
	soc	SOC	%	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	charge_discharge_power	Charge/Discharge power	kW	Double	
	total_discharge	Total energy discharged	kWh	Double	
	voltage	Voltage	V	Double	
	current	Current	A	Double	
	remaining_backup_time	Remaining power reserve duration	h	Double	
	total_discharge_times	Total discharge times	times	Double	
	total_capacity	Total capacity	kWh	Double	
ID: 60010 AC output power distribution (supported only in the Power-M scenario)	ac_voltage	AC voltage	V	Double	
	ac_current	AC current	A	Double	
	ac_frequency	AC frequency	Hz	Double	
	active_power	Active power	kW	Double	

Table 5-1 Inverter state (inverter_state) description

State Value	Description
0	Standby: initializing
1	Standby: insulation resistance detecting
2	Standby: irradiation detecting
3	Standby: grid detecting
256	Start
512	Grid-connected
513	Grid-connected: power limited
514	Grid-connected: self-derating
768	Shutdown: on fault
769	Shutdown: on command
770	Shutdown: OVGR
771	Shutdown: communication interrupted
772	Shutdown: power limited
773	Shutdown: manual startup required
774	Shutdown: DC switch disconnected
1025	Grid scheduling: cosψ-P curve
1026	Grid scheduling: Q-U curve
1280	Ready for terminal test
1281	Terminal testing...
1536	Inspection in progress
1792	AFCI self-check
2048	I-V scanning
2304	DC input detection
40960	Standby: no irradiation
45056	Communication interrupted (written by SmartLogger)
49152	Loading... (written by SmartLogger)

Examples

Request example:

```
{
    "devIds":"214060404588862,213472461631079",
    "devTypeId":"1"
}
```

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20006,
    "params":{
        "devIds":"214233501711677,214060404588862",
        "devTypeId":"1",
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: Real-time data of devices is returned.

```
{
    "success":true,
    "data":[
        {
            "dataItemMap":{
                "pv7_u":0,
                "pv1_u":0,
                "b_u":0,
                "c_u":0,
                "pv6_u":0,
                "temperature":0,
                "open_time":0,
                "b_i":0,
                "bc_u":0,
                "pv9_u":0,
                "pv8_u":0,
                "c_i":0,
                "mppt_total_cap":0,
                "pv9_i":0,
                "mppt_3_cap":0,
                "run_state":0,
                "mppt_2_cap":0,
                "inverter_state":0,
                "pv8_i":0,
                "mppt_1_cap":0,
                "pv6_i":0,
                "mppt_power":0,
                "pv1_i":0,
                "total_cap":0,
                "ab_u":0,
                "pv7_i":0,
                "pv13_u":0,
                "reactive_power":0,
                "pv10_u":0,
                "pv12_i":0,
                "pv11_i":0,
                "pv3_i":0,
                "pv11_u":0,
                "pv2_i":0,
                "pv13_i":0,
                "power_factor":0,
                "pv12_u":0,
                "pv5_i":0,
                "active_power":0,
                "elec_freq":0,
                "pv10_i":0,
                "pv4_i":0,
                "mppt_4_cap":0,
                "mppt_5_cap":0,
                "mppt_6_cap":0,
```

```
                    "mppt_7_cap":0,
                    "mppt_8_cap":0,
                    "mppt_9_cap":0,
                    "mppt_10_cap":0,
                    "pv4_u":0,
                    "close_time":0,
                    "day_cap":0,
                    "ca_u":0,
                    "a_i":0,
                    "pv5_u":0,
                    "a_u":0,
                    "pv3_u":0,
                    "pv14_u":0,
                    "pv14_i":0,
                    "pv15_u":0,
                    "pv15_i":0,
                    "pv16_u":0,
                    "pv16_i":0,
                    "pv17_u":0,
                    "pv17_i":0,
                    "pv18_u":0,
                    "pv18_i":0,
                    "pv19_u":0,
                    "pv19_i":0,
                    "pv20_u":0,
                    "pv20_i":0,
                    "pv21_u":0,
                    "pv21_i":0,
                    "pv22_u":0,
                    "pv22_i":0,
                    "pv23_u":0,
                    "pv23_i":0,
                    "pv24_u":0,
                    "pv24_i":0,
                    "pv25_u":0,
                    "pv25_i":0,
                    "pv26_u":0,
                    "pv26_i":0,
                    "pv27_u":0,
                    "pv27_i":0,
                    "pv28_u":0,
                    "pv28_i":0,
                    "efficiency":0,
                    "pv2_u":0
                },
                "devId":213472461631079
            },
            {
                "dataItemMap":{
                    "pv7_u":0,
                    "pv1_u":0,
                    "b_u":0,
                    "c_u":0,
                    "pv6_u":0,
                    "temperature":0,
                    "open_time":0,
                    "b_i":0,
                    "bc_u":0,
                    "pv9_u":0,
                    "pv8_u":0,
                    "c_i":0,
                    "mppt_total_cap":0,
                    "pv9_i":0,
                    "mppt_3_cap":0,
                    "run_state":0,
                    "mppt_2_cap":0,
                    "inverter_state":0,
                    "pv8_i":0,
                    "mppt_1_cap":0,
```

"pv6_i":0,
"mppt_power":0,
"pv1_i":0,
"total_cap":0,
"ab_u":0,
"pv7_i":0,
"pv13_u":0,
"reactive_power":0,
"pv10_u":0,
"pv12_i":0,
"pv11_i":0,
"pv3_i":0,
"pv11_u":0,
"pv2_i":0,
"pv13_i":0,
"power_factor":0,
"pv12_u":0,
"pv5_i":0,
"active_power":0,
"elec_freq":0,
"pv10_i":0,
"pv4_i":0,
"mppt_4_cap":0,
"mppt_5_cap":0,
"mppt_6_cap":0,
"mppt_7_cap":0,
"mppt_8_cap":0,
"mppt_9_cap":0,
"mppt_10_cap":0,
"pv4_u":0,
"close_time":0,
"day_cap":0,
"ca_u":0,
"a_i":0,
"pv5_u":0,
"a_u":0,
"pv3_u":0,
"pv14_u":0,
"pv14_i":0,
"pv15_u":0,
"pv15_i":0,
"pv16_u":0,
"pv16_i":0,
"pv17_u":0,
"pv17_i":0,
"pv18_u":0,
"pv18_i":0,
"pv19_u":0,
"pv19_i":0,
"pv20_u":0,
"pv20_i":0,
"pv21_u":0,
"pv21_i":0,
"pv22_u":0,
"pv22_i":0,
"pv23_u":0,
"pv23_i":0,
"pv24_u":0,
"pv24_i":0,
"pv25_u":0,
"pv25_i":0,
"pv26_u":0,
"pv26_i":0,
"pv27_u":0,
"pv27_i":0,
"pv28_u":0,
"pv28_i":0,
"efficiency":0,
"pv2_u":0

```
        },
        "devId":214060404588862
      }
    ],
    "failCode":0,
    "params":{
        "devIds":"214060404588862,213472461631079",
        "devTypeId":"1",
        "currentTime":1503046597854
    },
    "message":null
}
```

📖 **NOTE**

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.2.3 Historical Device Data API

API Description

This API is used to obtain 5-minute device data in a specified time period. The 5-minute data of a maximum of 10 devices of the same type in three days can be queried at a time.

For details about the data list that can be queried through this API, see the historical device data list below.

Request URL

https://*Domain name of the management system*/thirdData/getDevHistoryKpi

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
devIds	List of device IDs. The device IDs are obtained from **id** in **5.1.1.2 Device List API**. Use commas (,) to separate multiple device IDs. Either **sns** or **devIds** must be set.	String	Optional
sns	Device SN list. Multiple device SNs are separated by commas (,). Either **sns** or **devIds** must be set.	String	Optional
devTypeId	Device type ID. Use the device type ID obtained in **5.1.1.2 Device List API**. The following device types are supported: 1: string inverter 10: EMI 17: grid meter 38: residential inverter 39: battery 41: ESS 47: power sensor 60001: mains 60003: genset 60043: SSU group 60014: lithium battery rack 60010: AC output power distribution	Integer	Mandatory
startTime	Start time, in milliseconds	Long	Mandatory
endTime	End time, in milliseconds	Long	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	devIds	Device ID list in the request parameter	String	-
	sns	Device SN list in the request parameter	String	-
	devTypeId	Device type ID in the request parameter	Integer	-
	startTime	Start time, in milliseconds	Long	-
	endTime	End time, in milliseconds	Long	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data	Parameters	Returned data. The data contains the 5-minute data object list of each device.	List	5-minute data of a device in a day
	devId	Device ID	Long	-
	sn	Device SN	String	
	collectTime	Collection time in milliseconds in the request parameter	Long	-

Parameter		Description	Data Type	Remarks
	dataItemMap	Content of each data item, which is returned in the key-value format. The data item content varies depending on the device type. For details about the data item list, see the historical device data list below.	Map	5-minute device data

Historical Device Data List

Device Type	Key	Name	Unit	Return Value Type	Remarks
ID: 1 String inverter	inverter_state	Inverter state. For details, see the following: **Table 5-2**	None	Double	
	ab_u	A-B line voltage of grid	V	Double	
	bc_u	B-C line voltage of grid	V	Double	
	ca_u	C-A line voltage of grid	V	Double	
	a_u	Phase A voltage	V	Double	
	b_u	Phase B voltage	V	Double	
	c_u	Phase C voltage	V	Double	

Device Type	Key	Name	Unit	Ret urn Val ue Typ e	Remarks
	a_i	Phase A current of grid	A	Do ubl e	
	b_i	Phase B current of grid	A	Do ubl e	
	c_i	Phase C current of grid	A	Do ubl e	
	efficiency	Inverter conversion efficiency (manufacturer)	%	Do ubl e	
	temperature	Internal temperature	℃	Do ubl e	
	power_factor	Power factor	None	Do ubl e	
	elec_freq	Grid frequency	Hz	Do ubl e	
	active_power	Active power	kW	Do ubl e	
	reactive_power	Output reactive power	kVar	Do ubl e	
	day_cap	Yield today	kWh	Do ubl e	
	mppt_power	MPPT total input power	kW	Do ubl e	
	pv1_u	PV1 input voltage	V	Do ubl e	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv2_u	PV2 input voltage	V	Double	
	pv3_u	PV3 input voltage	V	Double	
	pv4_u	PV4 input voltage	V	Double	
	pv5_u	PV5 input voltage	V	Double	
	pv6_u	PV6 input voltage	V	Double	
	pv7_u	PV7 input voltage	V	Double	
	pv8_u	PV8 input voltage	V	Double	
	pv9_u	PV9 input voltage	V	Double	
	pv10_u	PV10 input voltage	V	Double	
	pv11_u	PV11 input voltage	V	Double	
	pv12_u	PV12 input voltage	V	Double	
	pv13_u	PV13 input voltage	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv14_u	PV14 input voltage	V	Double	
	pv15_u	PV15 input voltage	V	Double	
	pv16_u	PV16 input voltage	V	Double	
	pv17_u	PV17 input voltage	V	Double	
	pv18_u	PV18 input voltage	V	Double	
	pv19_u	PV19 input voltage	V	Double	
	pv20_u	PV20 input voltage	V	Double	
	pv21_u	PV21 input voltage	V	Double	
	pv22_u	PV22 input voltage	V	Double	
	pv23_u	PV23 input voltage	V	Double	
	pv24_u	PV24 input voltage	V	Double	
	pv25_u	PV25 input voltage	V	Double	

Device Type	Key	Name	Unit	Ret urn Val ue Typ e	Remarks
	pv26_u	PV26 input voltage	V	Do ubl e	
	pv27_u	PV27 input voltage	V	Do ubl e	
	pv28_u	PV28 input voltage	V	Do ubl e	
	pv1_i	PV1 input current	A	Do ubl e	
	pv2_i	PV2 input current	A	Do ubl e	
	pv3_i	PV3 input current	A	Do ubl e	
	pv4_i	PV4 input current	A	Do ubl e	
	pv5_i	PV5 input current	A	Do ubl e	
	pv6_i	PV6 input current	A	Do ubl e	
	pv7_i	PV7 input current	A	Do ubl e	
	pv8_i	PV8 input current	A	Do ubl e	
	pv9_i	PV9 input current	A	Do ubl e	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv10_i	PV10 input current	A	Double	
	pv11_i	PV11 input current	A	Double	
	pv12_i	PV12 input current	A	Double	
	pv13_i	PV13 input current	A	Double	
	pv14_i	PV14 input current	A	Double	
	pv15_i	PV15 input current	A	Double	
	pv16_i	PV16 input current	A	Double	
	pv17_i	PV17 input current	A	Double	
	pv18_i	PV18 input current	A	Double	
	pv19_i	PV19 input current	A	Double	
	pv20_i	PV20 input current	A	Double	
	pv21_i	PV21 input current	A	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv22_i	PV22 input current	A	Double	
	pv23_i	PV23 input current	A	Double	
	pv24_i	PV24 input current	A	Double	
	pv25_i	PV25 input current	A	Double	
	pv26_i	PV26 input current	A	Double	
	pv27_i	PV27 input current	A	Double	
	pv28_i	PV28 input current	A	Double	
	total_cap	Total yield	kWh	Double	
	open_time	Inverter startup time	ms	Double	
	close_time	Inverter shutdown time	ms	Double	
	mppt_total_cap	Total DC input energy	kWh	Double	
	mppt_1_cap	MPPT 1 DC total yield	kWh	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	mppt_2_cap	MPPT 2 DC total yield	kWh	Double	
	mppt_3_cap	MPPT 3 DC total yield	kWh	Double	
	mppt_4_cap	MPPT 4 DC total yield	kWh	Double	
	mppt_5_cap	MPPT 5 DC total yield	kWh	Double	
	mppt_6_cap	MPPT 6 DC total yield	kWh	Double	
	mppt_7_cap	MPPT 7 DC total yield	kWh	Double	
	mppt_8_cap	MPPT 8 DC total yield	kWh	Double	
	mppt_9_cap	MPPT 9 DC total yield	kWh	Double	
	mppt_10_cap	MPPT 10 DC total yield	kWh	Double	
ID: 38 Residential inverter	inverter_state	Inverter state. For details, see **Table 5-2**.	None	Double	
	ab_u	A-B line voltage of grid	V	Double	
	bc_u	B-C line voltage of grid	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	ca_u	C-A line voltage of grid	V	Double	
	a_u	Phase A voltage	V	Double	
	b_u	Phase B voltage	V	Double	
	c_u	Phase C voltage	V	Double	
	a_i	Phase A current of grid	A	Double	
	b_i	Phase B current of grid	A	Double	
	c_i	Phase C current of grid	A	Double	
	efficiency	Inverter conversion efficiency (manufacturer)	%	Double	
	temperature	Internal temperature	°C	Double	
	power_factor	Power factor	None	Double	
	elec_freq	Grid frequency	Hz	Double	
	active_power	Active power	kW	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	reactive_power	Output reactive power	kVar	Double	
	day_cap	Yield today	kWh	Double	
	mppt_power	MPPT total input power	kW	Double	
	pv1_u	PV1 input voltage	V	Double	
	pv2_u	PV2 input voltage	V	Double	
	pv3_u	PV3 input voltage	V	Double	
	pv4_u	PV4 input voltage	V	Double	
	pv5_u	PV5 input voltage	V	Double	
	pv6_u	PV6 input voltage	V	Double	
	pv7_u	PV7 input voltage	V	Double	
	pv8_u	PV8 input voltage	V	Double	
	pv1_i	PV1 input current	A	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	pv2_i	PV2 input current	A	Double	
	pv3_i	PV3 input current	A	Double	
	pv4_i	PV4 input current	A	Double	
	pv5_i	PV5 input current	A	Double	
	pv6_i	PV6 input current	A	Double	
	pv7_i	PV7 input current	A	Double	
	pv8_i	PV8 input current	A	Double	
	total_cap	Total yield	kWh	Double	
	open_time	Inverter startup time	ms	Double	
	close_time	Inverter shutdown time	ms	Double	
	mppt_1_cap	MPPT 1 DC total yield	kWh	Double	
	mppt_2_cap	MPPT 2 DC total yield	kWh	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	mppt_3_cap	MPPT 3 DC total yield	kWh	Double	
	mppt_4_cap	MPPT 4 DC total yield	kWh	Double	
ID: 10 EMI	temperature	Temperature	°C	Double	
	pv_temperature	PV temperature	°C	Double	
	wind_speed	Wind speed	m/s	Double	
	wind_direction	Wind direction	Degree	Double	
	radiant_total	Daily irradiation	MJ/m^2	Double	
	radiant_line	Irradiance	W/m^2	Double	
	horiz_radiant_line	Horizontal irradiance	W/m^2	Double	Invalid field. The default value is **null**.
	horiz_radiant_total	Horizontal irradiation	MJ/m^2	Double	Invalid field. The default value is **null**.
ID: 17 Grid meter	ab_u	A-B line voltage of grid	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	bc_u	B-C line voltage of grid	V	Double	
	ca_u	C-A line voltage of grid	V	Double	
	a_u	Phase A voltage (AC output)	V	Double	
	b_u	Phase B voltage (AC output)	V	Double	
	c_u	Phase C voltage (AC output)	V	Double	
	a_i	Phase A current of grid (IA)	A	Double	
	b_i	Phase B current of grid (IB)	A	Double	
	c_i	Phase C current of grid (IC)	A	Double	
	active_power	Active power	kW	Double	
	power_factor	Power factor	None	Double	
	active_cap	Active energy (positive active energy)	kWh	Double	
	reactive_power	Reactive power	kVar	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	reverse_active_cap	Negative active energy	kWh	Double	
	forward_reactive_cap	Positive reactive energy	kWh	Double	
	reverse_reactive_cap	Negative reactive energy	kWh	Double	
	active_power_a	Active power PA	kW	Double	
	active_power_b	Active power PB	kW	Double	
	active_power_c	Active power PC	kW	Double	
	reactive_power_a	Reactive power QA	kVar	Double	Invalid field. The default value is **null**.
	reactive_power_b	Reactive power QB	kVar	Double	Invalid field. The default value is **null**.
	reactive_power_c	Reactive power QC	kVar	Double	Invalid field. The default value is **null**.
	total_apparent_power	Total apparent power	kVA	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	grid_frequency	Grid frequency	Hz	Double	Invalid field. The default value is **null**.
	reverse_active_peak	Negative active energy (peak)	kWh	Double	Invalid field. The default value is **null**.
	reverse_active_power	Negative active energy (shoulder)	kWh	Double	Invalid field. The default value is **null**.
	reverse_active_valley	Negative active energy (off-peak)	kWh	Double	Invalid field. The default value is **null**.
	reverse_active_top	Negative active energy (sharp)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_peak	Positive active energy (peak)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_power	Positive active energy (shoulder)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_valley	Positive active energy (off-peak)	kWh	Double	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Return Value Type	Remarks
	positive_active_top	Positive active energy (sharp)	kWh	Double	Invalid field. The default value is **null**.
	reverse_reactive_peak	Negative reactive energy (peak)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_power	Negative reactive energy (shoulder)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_valley	Negative reactive energy (off-peak)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_top	Negative reactive energy (sharp)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_peak	Positive reactive energy (peak)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_power	Positive reactive energy (shoulder)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_valley	Positive reactive energy (off-peak)	kVar	Double	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Return Value Type	Remarks
	positive_reactive_top	Positive reactive energy (sharp)	kVar	Double	Invalid field. The default value is **null**.
ID: 47 Power sensor	meter_status	Meter state (0: offline; 1: normal)	None	Double	
	meter_u	Grid voltage	V	Double	
	meter_i	Grid current	A	Double	
	active_power	Active power	W	Double	
	reactive_power	Reactive power	Var	Double	
	power_factor	Power factor	None	Double	
	grid_frequency	Grid frequency	Hz	Double	
	active_cap	Active energy (positive active energy)	kWh	Double	
	reverse_active_cap	Negative active energy	kWh	Double	
	ab_u	A-B line voltage of grid	V	Double	
	bc_u	B-C line voltage of grid	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	ca_u	C-A line voltage of grid	V	Double	
	b_u	Phase B voltage (AC output)	V	Double	
	c_u	Phase C voltage (AC output)	V	Double	
	b_i	Phase B current of grid (IB)	A	Double	
	c_i	Phase C current of grid (IC)	A	Double	
	forward_reactive_cap	Positive reactive energy	kWh	Double	Invalid field. The default value is **null**.
	reverse_reactive_cap	Negative reactive energy	kWh	Double	Invalid field. The default value is **null**.
	active_power_a	Active power PA	kW	Double	
	active_power_b	Active power PB	kW	Double	
	active_power_c	Active power PC	kW	Double	

Device Type	Key	Name	Unit	Ret urn Val ue Typ e	Remarks
	reactive_power_a	Reactive power QA	kVar	Do ubl e	Invalid field. The default value is **null**.
	reactive_power_b	Reactive power QB	kVar	Do ubl e	Invalid field. The default value is **null**.
	reactive_power_c	Reactive power QC	kVar	Do ubl e	Invalid field. The default value is **null**.
	total_apparent_power	Total apparent power	kVA	Do ubl e	Invalid field. The default value is **null**.
	reverse_active_peak	Negative active energy (peak)	kWh	Do ubl e	Invalid field. The default value is **null**.
	reverse_active_power	Negative active energy (shoulder)	kWh	Do ubl e	Invalid field. The default value is **null**.
	reverse_active_valley	Negative active energy (off-peak)	kWh	Do ubl e	Invalid field. The default value is **null**.
	reverse_active_top	Negative active energy (sharp)	kWh	Do ubl e	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Return Value Type	Remarks
	positive_active_peak	Positive active energy (peak)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_power	Positive active energy (shoulder)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_valley	Positive active energy (off-peak)	kWh	Double	Invalid field. The default value is **null**.
	positive_active_top	Positive active energy (sharp)	kWh	Double	Invalid field. The default value is **null**.
	reverse_reactive_peak	Negative reactive energy (peak)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_power	Negative reactive energy (shoulder)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_valley	Negative reactive energy (off-peak)	kVar	Double	Invalid field. The default value is **null**.
	reverse_reactive_top	Negative reactive energy (sharp)	kVar	Double	Invalid field. The default value is **null**.

Device Type	Key	Name	Unit	Return Value Type	Remarks
	positive_reactive_peak	Positive reactive energy (peak)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_power	Positive reactive energy (shoulder)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_valley	Positive reactive energy (off-peak)	kVar	Double	Invalid field. The default value is **null**.
	positive_reactive_top	Positive reactive energy (sharp)	kVar	Double	Invalid field. The default value is **null**.
ID: 39 Residential battery	battery_status	Battery running state (0: offline; 1: standby; 2: running; 3: faulty; 4: hibernating)	None	Double	
	max_charge_power	Maximum charge power	W	Double	
	max_discharge_power	Maximum discharge power	W	Double	
	ch_discharge_power	Charge/Discharge power	W	Double	
	busbar_u	Battery voltage	V	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
	battery_soc	Battery SOC	%	Double	
	battery_soh	Battery SOH	None	Double	
	ch_discharge_model	Charge/Discharge mode (0: none; 1: forced charge/discharge; 2: time-of-use price; 3: fixed charge/discharge; 4: automatic charge/discharge)	None	Double	
	charge_cap	Charged energy	kWh	Double	
	discharge_cap	Discharged energy	kWh	Double	
ID: 41 C&I and utility ESS	ch_discharge_power	Charge/Discharge power	W	Double	
	battery_soc	Battery SOC	%	Double	
	charge_cap	Charged energy	kWh	Double	
	discharge_cap	Discharged energy	kWh	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
ID: 60001 Mains (supported only in the Power-M scenario)	mains_state	Mains status (0: mains unavailable; 1: mains available)	None	Double	
	ac_voltage	AC voltage	V	Double	
	ac_current	AC current	A	Double	
	active_power	Active power	kW	Double	
	ac_frequency	AC frequency	Hz	Double	
ID: 60003 Genset (supported only in the Power-M scenario)	load_rate	Load rate	%	Double	
	running_state	Running status (0: unknown; 1: stopped; 2: running)	None	Double	
ID: 60043 SSU group (supported only in the Power-M scenario)	total_output_current	Total output current	A	Double	
	total_output_energy	Total output energy	kWh	Double	
ID: 60014 Lithium battery rack (supported only in the Power-M scenario)	soc	SOC	%	Double	
	voltage	Voltage	V	Double	
	average_temperature	Average temperature	°C	Double	

Device Type	Key	Name	Unit	Return Value Type	Remarks
ID: 60010 AC output power distribution (supported only in the Power-M scenario)	ac_frequency	AC frequency	Hz	Double	
	total_load_active_power	Total load active power	kW	Double	

Table 5-2 Inverter state (inverter_state) description

State Value	Description
0	Standby: initializing
1	Standby: insulation resistance detecting
2	Standby: irradiation detecting
3	Standby: grid detecting
256	Start
512	Grid-connected
513	Grid-connected: power limited
514	Grid-connected: self-derating
768	Shutdown: on fault
769	Shutdown: on command
770	Shutdown: OVGR
771	Shutdown: communication interrupted
772	Shutdown: power limited
773	Shutdown: manual startup required
774	Shutdown: DC switch disconnected
1025	Grid scheduling: cosφ-P curve
1026	Grid scheduling: Q-U curve
1280	Ready for terminal test

State Value	Description
1281	Terminal testing...
1536	Inspection in progress
1792	AFCI self-check
2048	I-V scanning
2304	DC input detection
40960	Standby: no irradiation
45056	Communication interrupted (written by SmartLogger)
49152	Loading... (written by SmartLogger)

Examples

Request example:

```
{
    "devIds":"214060404588862,213472461631079",
    "devTypeId":1,
    "startTime":1501862400000,
    "endTime":1501872400000
}
```

Response examples:

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20009,
    "params":{
        "devIds":"214060404588862,213472461631079",
        "devTypeId":1,
        "startTime":1501862400000,
        "endTime":1501872400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: 5-minute device data is returned.

```
{
    "success":true,
    "data":[
        {
            "dataItemMap":{
                "pv7_u":null,
                "pv1_u":575.3,
                "b_u":286.1,
                "c_u":286.9,
                "pv6_u":576.1,
                "temperature":44.6,
                "open_time":null,
                "b_i":24.9,
                "bc_u":495.6,
                "pv9_u":null,
```

"pv8_u":null,
"c_i":25,
"mppt_total_cap":null,
"pv9_i":null,
"mppt_3_cap":null,
"mppt_2_cap":null,
"inverter_state":512,
"pv8_i":null,
"mppt_1_cap":null,
"pv6_i":7.1,
"mppt_power":21.962,
"pv1_i":7.1,
"total_cap":655.37,
"ab_u":495.4,
"pv7_i":null,
"pv13_u":null,
"reactive_power":20.95,
"pv10_u":null,
"pv12_i":null,
"pv11_i":null,
"pv3_i":7.1,
"pv11_u":null,
"pv2_i":7.1,
"pv13_i":null,
"power_factor":0,
"pv12_u":null,
"pv5_i":7.2,
"active_power":21.05,
"elec_freq":50.05,
"pv10_i":null,
"pv4_i":7,
"mppt_4_cap":null,
"mppt_5_cap":0,
"mppt_6_cap":0,
"mppt_7_cap":0,
"mppt_8_cap":0,
"mppt_9_cap":0,
"mppt_10_cap":0,
"pv4_u":577.8,
"close_time":null,
"day_cap":159.26,
"ca_u":496.9,
"a_i":24.9,
"pv5_u":576.1,
"a_u":286,
"pv3_u":577.8,
"pv14_u":null,
"pv14_i":null,
"pv15_u":0,
"pv15_i":0,
"pv16_u":0,
"pv16_i":0,
"pv17_u":0,
"pv17_i":0,
"pv18_u":0,
"pv18_i":0,
"pv19_u":0,
"pv19_i":0,
"pv20_u":0,
"pv20_i":0,
"pv21_u":0,
"pv21_i":0,
"pv22_u":0,
"pv22_i":0,
"pv23_u":0,
"pv23_i":0,
"pv24_u":0,
"pv24_i":0,
"pv25_u":0,

```
              "pv25_i":0,
              "pv26_u":0,
              "pv26_i":0,
              "pv27_u":0,
              "pv27_i":0,
              "pv28_u":0,
              "pv28_i":0,
              "efficiency":null,
              "pv2_u":575.3
          },
          "devId":213472461631079,
          "collectTime":1501862400000
      },
      {
          "dataItemMap":{
              "pv7_u":null,
              "pv1_u":575.3,
              "b_u":286.1,
              "c_u":286.9,
              "pv6_u":576.1,
              "temperature":44.6,
              "open_time":null,
              "b_i":24.9,
              "bc_u":495.6,
              "pv9_u":null,
              "pv8_u":null,
              "c_i":25,
              "mppt_total_cap":null,
              "pv9_i":null,
              "mppt_3_cap":null,
              "mppt_2_cap":null,
              "inverter_state":512,
              "pv8_i":null,
              "mppt_1_cap":null,
              "pv6_i":7.1,
              "mppt_power":21.962,
              "pv1_i":7.1,
              "total_cap":655.37,
              "ab_u":495.4,
              "pv7_i":null,
              "pv13_u":null,
              "reactive_power":20.95,
              "pv10_u":null,
              "pv12_i":null,
              "pv11_i":null,
              "pv3_i":7.1,
              "pv11_u":null,
              "pv2_i":7.1,
              "pv13_i":null,
              "power_factor":0,
              "pv12_u":null,
              "pv5_i":7.2,
              "active_power":21.05,
              "elec_freq":50.05,
              "pv10_i":null,
              "pv4_i":7,
              "mppt_4_cap":null,
              "mppt_5_cap":0,
              "mppt_6_cap":0,
              "mppt_7_cap":0,
              "mppt_8_cap":0,
              "mppt_9_cap":0,
              "mppt_10_cap":0,
              "pv4_u":577.8,
              "close_time":null,
              "day_cap":159.26,
              "ca_u":496.9,
              "a_i":24.9,
              "pv5_u":576.1,
```

```
                                "a_u":286,
                                "pv3_u":577.8,
                                "pv14_u":null,
                                "pv14_i":null,
                                "pv15_u":0,
                                "pv15_i":0,
                                "pv16_u":0,
                                "pv16_i":0,
                                "pv17_u":0,
                                "pv17_i":0,
                                "pv18_u":0,
                                "pv18_i":0,
                                "pv19_u":0,
                                "pv19_i":0,
                                "pv20_u":0,
                                "pv20_i":0,
                                "pv21_u":0,
                                "pv21_i":0,
                                "pv22_u":0,
                                "pv22_i":0,
                                "pv23_u":0,
                                "pv23_i":0,
                                "pv24_u":0,
                                "pv24_i":0,
                                "pv25_u":0,
                                "pv25_i":0,
                                "pv26_u":0,
                                "pv26_i":0,
                                "pv27_u":0,
                                "pv27_i":0,
                                "pv28_u":0,
                                "pv28_i":0,
                                "efficiency":null,
                                "pv2_u":575.3
                        },
                        "devId":213472461631079,
                        "collectTime":1501862700000
                }
        ],
        "failCode":0,
        "params":{
                "devIds":"214060404588862,213472461631079",
                "devTypeId":1,
                "startTime":1501862400000,
                "endTime":1501872400000,
                "currentTime":1503046597854
        },
        "message":null
}
```

◯ **NOTE**

Prerequisites for obtaining data: The API account has the permission to access this API.

5.1.3 Alarm

5.1.3.1 API for Querying Active Alarms

API Description

This API is used to query the current (active) alarm information of a device. If the query is based on plants, a maximum of 100 plants can be queried at a time. If the query is based on device SNs, a maximum of 100 devices can be queried at a time. If the transferred plant ID list is not empty, device alarm information is

queried based on the plant ID list. If the plant ID list is empty and the device SN list is not empty, device alarm information is queried based on the device SN list.

Request URL

https://*Domain name of the management system*/thirdData/getAlarmList

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
stationCodes	Plant ID list. Multiple plant IDs are separated by commas (,). The plant IDs are obtained from **plantCode** in **5.1.1.1 Plant List API**. At least one of the **stationCodes** and **sns** parameters must be set. If both parameters are set, alarms are queried by **stationCodes**.	String	Optional
sns	Device SN list. Multiple device SNs are separated by commas (,). At least one of the **stationCodes** and **sns** parameters must be set. If both parameters are set, alarms are queried by **stationCodes**.	String	Optional
beginTime	Query start time, in milliseconds	Long	Mandatory
endTime	Query end time, in milliseconds	Long	Mandatory

Parameter	Description	Data Type	Mandatory/ Optional
language	Language. The value must be **zh_CN**, **en_US**, **ja_JP**, **it_IT**, **nl_NL**, **pt_BR**, **de_DE**, **fr_FR**, **es_ES**, or **pl_PL**. zh_CN: Chinese en_US: English ja_JP: Japanese it_IT: Italian nl_NL: Dutch pt_BR: Portuguese de_DE: German fr_FR: French es_ES: Spanish pl_PL: Polish	String	Mandatory
levels	Alarm severity. Multiple alarm severities are separated by commas (,), for example, **1,2**. If this parameter is not transferred or is left empty, alarms of all severities are queried by default. The following alarm severities are supported: 1: critical 2: major 3: minor 4: warning	String	Optional

Parameter	Description	Data Type	Mandatory/Optional
devTypes	Device type. Multiple device types are separated by commas (,), for example, **1,38**. If this parameter is not transferred or is left empty, alarms of all device types are queried by default. The following device types are supported: 1: string inverter 2: SmartLogger 8: STS 10: EMI 13: protocol converter 16: general device 17: grid meter 22: PID 37: Pinnet data logger 38: residential inverter 39: battery 40: backup box 45: PLC 46: optimizer 47: power sensor 62: Dongle 63: distributed SmartLogger 70: safety box 60001: mains 60003: genset 60043: SSU group 60044: SSU 60092: power converter 60014: lithium battery rack 60010: AC output power distribution	String	Optional

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag. Value: **true**: The request succeeded. **false**: The request failed.	Boolean	Request success or failure flag. Value:
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	stationCodes	Plant ID list in the request parameter.	String	-
	sns	Device SN list in the request parameter.	String	-
	beginTime	Start time in the request parameter, in milliseconds.	Long	-
	endTime	End time in the request parameter, in milliseconds.	Long	-
	language	Language in the request parameter.	String	-
	levels	Alarm severity in the request parameter.	String	-
	devTypes	Device type in the request parameter.	String	-
	currentTime	Current system time, in milliseconds.	Long	-
message		Optional message.	String	-
data	Parameters	Returned data. The data contains the alarm information list.	List	-
	stationCode	Plant ID, which uniquely identifies a plant.	String	-
	alarmName	Alarm name.	String	-
	devName	Device name.	String	-

Parameter		Description	Data Type	Remarks
	repairSuggestion	Suggestions.	String	-
	esnCode	Device SN.	String	-
	devTypeId	Device type ID. The following device types are supported: 1: string inverter 2: SmartLogger 8: STS 10: EMI 13: protocol converter 16: general device 17: grid meter 22: PID 37: Pinnet data logger 38: residential inverter 39: battery 40: backup box 45: PLC 46: optimizer 47: power sensor 62: Dongle 63: distributed SmartLogger 70: safety box 60001: mains 60003: genset 60043: SSU group 60044: SSU 60092: power converter 60014: lithium battery rack 60010: AC output power distribution	Integer	-
	causeId	Cause ID.	Integer	-
	alarmCause	Alarm cause.	String	-

Parameter		Description	Data Type	Remarks
	alarmType	Alarm type. The following alarm types are supported: 0: other alarms 1: transposition signal 2: exception alarm 3: protection event 4: notification status 5: alarm information	Integer	-
	raiseTime	Alarm generation time in milliseconds	Long	-
	alarmId	Alarm ID	Integer	-
	stationName	Plant name	String	-
	lev	Alarm severity The following alarm severities are supported: 1: critical 2: major 3: minor 4: warning	Integer	-
	status	Alarm status. The following alarm states are supported: 1: not processed (active)	Integer	-

Example

Request example:

```
{
    "stationCodes":"NE=33554785,NE=33554792",
    "sns":"Inverter01",
    "beginTime":"1664553600000",
    "endTime":"1667231999000",
    "language":"zh_CN",
    "levels":"1,2,3,4",
    "devTypes":"1,2,38,46,62"
}
```

Response examples:

Example 1: An error code is returned.

```
{
    "data": null,
```

```
      "failCode": 20055,
      "message": null,
      "params": {
         "currentTime": 1667399781133,
         "sns": "",
         "language": "zh_CN",
         "beginTime": 1664553600000,
         "devTypes": "1,2,38,46,62",
         "endTime": 1667231999000,
         "levels": "1,2,3,4",
         "stationCodes": ""
      },
      "success": false
}
```

Example 2: Alarm data of the device is returned.

```
{
   "data": [
      {
         "alarmCause": "An unrecoverable fault has occurred in the internal circuit of the device.",
         "alarmId": 2064,
"alarmName": "The device is abnormal.",
         "alarmType": 2,
         "causeId": 5,
         "devName": "Inverter-1",
         "devTypeId": 38,
         "esnCode": "Inverter05",
         "lev": 2,
         "raiseTime": 1667179861000,
         "repairSuggestion": "Turn off the AC and DC switches, wait for 5 minutes, and then turn on the AC
and DC switches. If the fault persists, contact your dealer or technical support.",
         "stationCode": "NE=33554792",
         "stationName": "hzhStation02",
         "status": 1
      },
      {
         "alarmCause": "1. The voltage of a battery expansion module is low.",
         "alarmId": 3011,
         "alarmName": "Battery expansion module undervoltage",
         "alarmType": 2,
         "causeId": 2,
         "devName": "Inverter-2",
         "devTypeId": 38,
         "esnCode": "Inverter01",
         "lev": 4,
         "raiseTime": 1665264943000,
         "repairSuggestion": "1. If the sunlight is sufficient or AC reverse charging is allowed, the Battery
[CNo] battery expansion module [SNo] (in the fault location information) can be charged when the
inverter is running.",
         "stationCode": "NE=33554785",
         "stationName": "hzhStation01",
         "status": 1
      }
   ],
   "failCode": 0,
   "message": null,
   "params": {
      "currentTime": 1667399432812,
      "sns": "Inverter01",
      "language": "zh_CN",
      "beginTime": 1664553600000,
      "devTypes": "1,2,38,46,62",
      "endTime": 1667231999000,
      "levels": "1,2,3,4",
      "stationCodes": "NE=33554785,NE=33554792"
   },
   "success": true
}
```

📖 NOTE

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.4 Report

5.1.4.1 Hourly Plant Data API

API Description

This API is used to obtain hourly plant data. Data of a maximum of 100 plants can be queried at a time.

The backend calculates the date of the collection time based on the request parameter **collectTime** (collection time in milliseconds) and the time zone where

the plant is located. Then, you can query the hourly data of the plant by plant ID in the current day. If data is generated for n ($0 \leq n \leq 24$) hours of the day, n ($0 \leq n \leq 24$) results are returned.

For details about the data list that can be queried through this API, see the hourly plant data list below.

Request URL

https://Domain name of the management system/thirdData/getKpiStationHour

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
stationCodes	Plant ID list. Multiple plant IDs are separated by commas (,). The plant IDs are obtained from **plantCode** in **5.1.1.1 Plant List API**.	String	Mandatory
collectTime	Collection time, in milliseconds	Long	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	stationCodes	Plant ID list in the request parameter	String	-

Parameter		Description	Data Type	Remarks
	collectTime	Collection time in milliseconds in the request parameter	Long	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data	Parameters	Returned data. The data contains the hourly data object list of each plant.	List	List of hourly plant data in a day
	stationCode	Plant ID	String	-
	collectTime	Collection time, in milliseconds	Long	-
	dataItemMap	Content of each data item, which is returned in key-value format. For details about the data item list, see the hourly plant data list below.	Map	-

Hourly Plant Data List

Key	Name	Unit	Return Value Type	Remarks
radiation_intensity	Global irradiation	kWh/m²	Double	
theory_power	Theoretical yield	kWh	Double	
inverter_power	Inverter yield	kWh	Double	The calculation of this indicator is inaccurate. The **inverterYield** is preferred.
ongrid_power	Feed-in energy	kWh	Double	

Key	Name	Unit	Return Value Type	Remarks
power_profit	Revenue	The currency specified in the management system	Double	
chargeCap	Charged energy	kWh	Double	
dischargeCap	Discharged energy	kWh	Double	
selfProvide	Energy consumed from PV	kWh	Double	
PVYield	PV yield	kWh	Double	
inverterYield	Inverter yield	kWh	Double	

Example

Request example:

```
{
    "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
    "collectTime":1501862400000
}
```

Response example:

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20009,
    "params":{
        "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
        "collectTime":1501862400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: Hourly plant data is returned.

```
{
    "success":true,
    "data":[
        {
            "dataItemMap":{
                "radiation_intensity":null,
                "theory_power":null,
                "inverter_power":0,
```

```
            "ongrid_power":null,
            "power_profit":0,
            "chargeCap":null,
            "dischargeCap":null,
            "selfProvide":null,

         },
         "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
         "collectTime":1501862400000
      },
      {
         "dataItemMap":{
            "radiation_intensity":null,
            "theory_power":null,
            "inverter_power":0,
            "ongrid_power":null,
            "power_profit":0,
            "chargeCap":null,
            "dischargeCap":null,
            "selfProvide":null,
         },
         "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
         "collectTime":1501866000000
      },
      {
         "dataItemMap":{
            "radiation_intensity":null,
            "theory_power":null,
            "inverter_power":0,
            "ongrid_power":null,
            "power_profit":0,
            "chargeCap":null,
            "dischargeCap":null,
            "selfProvide":null,
         },
         "stationCode":"BA4372D08E014822AB065017416F254C",
         "collectTime":1501873200000
      },
      {
         "dataItemMap":{
            "radiation_intensity":null,
            "theory_power":null,
            "inverter_power":0,
            "ongrid_power":null,
            "power_profit":0,
            "chargeCap":null,
            "dischargeCap":null,
            "selfProvide":null,
         },
         "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
         "collectTime":1501876800000
      },
      {
         "dataItemMap":{
            "radiation_intensity":null,
            "theory_power":null,
            "inverter_power":0,
            "ongrid_power":null,
            "power_profit":0,
            "chargeCap":null,
            "dischargeCap":null,
            "selfProvide":null,
         },
         "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
         "collectTime":1501880400000
      },
      {
         "dataItemMap":{
            "radiation_intensity":null,
```

```
                    "theory_power":null,
                    "inverter_power":0,
                    "ongrid_power":null,
                    "power_profit":0,
                    "chargeCap":null,
                    "dischargeCap":null,
                    "selfProvide":null,
                },
                "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
                "collectTime":1501884000000
            },
            {
                "dataItemMap":{
                    "radiation_intensity":null,
                    "theory_power":null,
                    "inverter_power":0,
                    "ongrid_power":null,
                    "power_profit":0,
                    "chargeCap":null,
                    "dischargeCap":null,
                    "selfProvide":null,
                },
                "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
                "collectTime":1501887600000
            },
            {
                "dataItemMap":{
                    "radiation_intensity":null,
                    "theory_power":null,
                    "inverter_power":0,
                    "ongrid_power":null,
                    "power_profit":0,
                    "chargeCap":null,
                    "dischargeCap":null,
                    "selfProvide":null,
                },
                "stationCode":"BA4372D08E014822AB065017416F254C",
                "collectTime":1501887600000
            }
        ],
        "failCode":0,
        "params":{
            "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
            "collectTime":1501862400000,
            "currentTime":1503046597854
        },
        "message":null
}
```

📖 **NOTE**

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.4.2 Daily Plant Data API

API Description

This API is used to obtain daily plant data. Data of a maximum of 100 plants can be queried at a time.

The backend calculates the month of the collection time based on the request parameter **collectTime** (collection time in milliseconds) and the time zone where the plant is located. Then, you can query the daily data of the plant by plant ID in the current month. If data is generated for n ($0 \le n \le 31$) days of the month, n ($0 \le n \le 31$) results are returned.

For details about the data list that can be queried through this API, see the daily plant data list below.

Request URL

https://*{Domain name of the management system}*/thirdData/getKpiStationDay

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
stationCodes	Plant ID list. Multiple plant IDs are separated by commas (,). The plant IDs are obtained from **plantCode** in **5.1.1.1 Plant List API**.	String	Mandatory
collectTime	Collection time, in milliseconds	Long	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag. **true**: The request succeeded. **false**: The request failed.	boolean	Request success or failure flag
failCode		Error code. Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	stationCodes	Plant ID list in the request parameter.	String	-
	collectTime	Collection time in milliseconds in the request parameter.	Long	-
	currentTime	Current system time, in milliseconds.	Long	-
message		Optional message.	String	-
data	Parameters	Returned data. The data contains the daily data object list of each plant.	List	Daily data list of a plant in a month
	stationCode	Plant ID.	String	-

Parameter		Description	Data Type	Remarks
	collectTime	Collection time, in milliseconds.	Long	-
	dataItemMap	Content of each data item, which is returned in key-value format. For details about the data item list, see the daily plant data list below.	Map	-

Daily Plant Data List

Key	Name	Unit	Return Value Type	Remarks
installed_capacity	Installed capacity	kW	Double	
radiation_intensity	Global irradiation	kWh/m²	Double	
theory_power	Theoretical yield	kWh	Double	
performance_ratio	Performance ratio	%		
inverter_power	Inverter yield	kWh	Double	The calculation of this indicator is inaccurate. The **inverterYield** is preferred.
ongrid_power	Feed-in energy	kWh	Double	
use_power	Consumption	kWh	Double	
power_profit	Revenue	The currency specified in the management system	Double	
perpower_ratio	Specific energy (kWh/kWp)	h	Double	

Key	Name	Unit	Return Value Type	Remarks
reduction_total_co2	CO_2 emission reduction	Ton	Double	
reduction_total_coal	Standard coal saved	Ton	Double	
reduction_total_tree	Equivalent trees planted	N/A	Double	The indicator is reserved and the value cannot be obtained.
buyPower	Energy from grid	kWh	Double	
chargeCap	Charged energy	kWh	Double	
dischargeCap	Discharged energy	kWh	Double	
selfUsePower	Consumed PV energy	kWh	Double	
selfProvide	Energy consumed from PV	kWh	Double	
PVYield	PV yield	kWh	Double	
inverterYield	Inverter yield	kWh	Double	

Example

Request example:

```
{
    "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
    "collectTime":1501862400000
}
```

Response examples:

Example 1: An error code is returned.

```
{
    "data": null,
    "failCode": 20012,
    "message": null,
    "params": {
        "currentTime": 1687788735543,
        "collectTime": -1677051223000,
        "stationCodes": "BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5"
    },
```

```
    "success": false
}
```

Example 2: Daily plant data is returned.

```
{
    "success":true,
    "data":[
        {
            "dataItemMap":{
                "use_power":288760,
                "radiation_intensity":0.6968,
                "reduction_total_co2":18.275,
                "reduction_total_coal":7.332,
                "theory_power":17559.36,
                "ongrid_power":18330,
                "power_profit":34320,
                "installed_capacity":25200,
                "perpower_ratio":0.727,
                "inverter_power":18330,
                "reduction_total_tree":999,
                "performance_ratio":89
            },
            "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
            "collectTime":1501776000000
        },
        {
            "dataItemMap":{
                "use_power":null,
                "radiation_intensity":1.4123,
                "reduction_total_co2":0.897,
                "reduction_total_coal":0.36,
                "theory_power":659.6,
                "ongrid_power":null,
                "power_profit":2088,
                "installed_capacity":467.04,
                "perpower_ratio":1.927,
                "inverter_power":18330,
                "reduction_total_tree":49,
                "performance_ratio":89
            },
            "stationCode":"BA4372D08E014822AB065017416F254C",
            "collectTime":1501776000000
        }
    ],
    "failCode":0,
    "params":{
        "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
        "collectTime":1501862400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

📖 **NOTE**

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.4.3 Monthly Plant Data API

API Description

This API is used to obtain monthly plant data. Data of a maximum of 100 plants can be queried at a time.

The backend calculates the year of the collection time based on the request parameter **collectTime** (collection time in milliseconds) and the time zone where the plant is located. Then, you can query the monthly data of the plant by plant ID in the current year. If data is generated for n ($0 \leq n \leq 12$) months of the year, n ($0 \leq n \leq 12$) results are returned.

For details about the data list that can be queried through this API, see the monthly plant data list below.

Request URL

https://*{Domain name of the management system}*/thirdData/getKpiStationMonth

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
stationCodes	Plant ID list. Multiple plant IDs are separated by commas (,). The plant IDs are obtained from **plantCode** in **5.1.1.1 Plant List API**.	String	Mandatory
collectTime	Collection time, in milliseconds	Long	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag. **true**: The request succeeded. **false**: The request failed.	boolean	Request success or failure flag
failCode		Error code. Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	stationCodes	Plant ID list in the request parameter.	String	-
	collectTime	Collection time in milliseconds in the request parameter.	Long	-
	currentTime	Current system time, in milliseconds.	Long	-
message		Optional message.	String	-

Parameter		Description	Data Type	Remarks
data	Parameters	Returned data. The data contains the monthly data object list of each plant.	List	Monthly data list of a plant in a year
	stationCode	Plant ID.	String	-
	collectTime	Collection time, in milliseconds.	Long	-
	dataItemMap	Content of each data item, which is returned in key-value format. For details about the data item list, see the monthly plant data list below.	Map	-

Monthly Plant Data List

Key	Name	Unit	Return Value Type	Remarks
installed_capacity	Installed capacity	kW	Double	
radiation_intensity	Global irradiation	kWh/m²	Double	
theory_power	Theoretical yield	kWh	Double	
performance_ratio	Performance ratio	%		
inverter_power	Inverter yield	kWh	Double	The calculation of this indicator is inaccurate. The **inverterYield** is preferred.
ongrid_power	Feed-in energy	kWh	Double	
use_power	Consumption	kWh	Double	

Key	Name	Unit	Return Value Type	Remarks
power_profit	Revenue	The currency specified in the management system	Double	
perpower_ratio	Specific energy (kWh/kWp)	h	Double	
reduction_total_co2	CO_2 emission reduction	Ton	Double	
reduction_total_coal	Standard coal saved	Ton	Double	
reduction_total_tree	Equivalent trees planted	N/A	Double	The indicator is reserved and the value cannot be obtained.
buyPower	Energy from grid	kWh	Double	
chargeCap	Charged energy	kWh	Double	
dischargeCap	Discharged energy	kWh	Double	
selfUsePower	Consumed PV energy	kWh	Double	
selfProvide	Energy consumed from PV	kWh	Double	
PVYield	PV yield	kWh	Double	
inverterYield	Inverter yield	kWh	Double	

Example

Request example:

```
{
    "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
    "collectTime":1501862400000
}
```

Response examples:

Example 1: An error code is returned.

```
{
    "data": null,
```

```
    "failCode": 20012,
    "message": null,
    "params": {
        "currentTime": 1687788735543,
        "collectTime": -1677051223000,
        "stationCodes": "BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5"
    },
    "success": false
}
```

Example 2: Monthly plant data is returned.

```
{
    "success":true,
    "data":[
        {
            "dataItemMap":{
                "use_power":288760,
                "radiation_intensity":0.6968,
                "reduction_total_co2":18.275,
                "reduction_total_coal":7.332,
                "inverter_power":null,
                "theory_power":17559.36,
                "ongrid_power":18330,
                "power_profit":34320,
                "installed_capacity":25200,
                "perpower_ratio":0.727,
                "reduction_total_tree":999,
                "performance_ratio":89
            },
            "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
            "collectTime":1501516800000
        },
        {
            "dataItemMap":{
                "use_power":null,
                "radiation_intensity":1.4123,
                "reduction_total_co2":0.897,
                "reduction_total_coal":0.36,
                "inverter_power":null,
                "theory_power":659.6,
                "ongrid_power":null,
                "power_profit":2088,
                "installed_capacity":467.04,
                "perpower_ratio":1.927,
                "reduction_total_tree":49,
                "performance_ratio":89
            },
            "stationCode":"BA4372D08E014822AB065017416F254C",
            "collectTime":1501516800000
        }
    ],
    "failCode":0,
    "params":{
        "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
        "collectTime":1501862400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

NOTE

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.4.4 Yearly Plant Data API

API Description

This API is used to obtain yearly plant data. Data of a maximum of 100 plants can be queried at a time.

Based on the plant ID, the backend queries the data of each year since the plant was constructed (including the current year).

For details about the data list that can be queried through this API, see the yearly plant data list below.

Request URL

https://Domain name of the management system/thirdData/getKpiStationYear

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
stationCodes	Plant ID list. Multiple plant IDs are separated by commas (,). The plant IDs are obtained from **plantCode** in **5.1.1.1 Plant List API**.	String	Mandatory
collectTime	Collection time, in milliseconds	Long	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	stationCodes	Plant ID list in the request parameter	String	-
	collectTime	Collection time in milliseconds in the request parameter	Long	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data	Parameters	Returned data. The data contains the yearly data object list of each plant.	List	Yearly data list of the plant since its construction

Parameter		Description	Data Type	Remarks
	stationCode	Plant ID	String	-
	collectTime	Collection time, in milliseconds	Long	-
	dataItemMap	Content of each data item, which is returned in key-value format. For details about the data item list, see the yearly plant data list below.	Map	-

Yearly Plant Data List

Key	Name	Unit	Return Value Type	Remarks
installed_capacity	Installed capacity	kW	Double	
radiation_intensity	Global irradiation	kWh/m²	Double	
theory_power	Theoretical yield	kWh	Double	
performance_ratio	Performance ratio	%		
inverter_power	Inverter yield	kWh	Double	The calculation of this indicator is inaccurate. The **inverterYield** is preferred.
ongrid_power	Feed-in energy	kWh	Double	
use_power	Consumption	kWh	Double	
power_profit	Revenue	The currency specified in the management system	Double	

Key	Name	Unit	Return Value Type	Remarks
perpower_ratio	Specific energy (kWh/kWp)	h	Double	
reduction_total_co2	CO_2 emission reduction	Ton	Double	
reduction_total_coal	Standard coal saved	Ton	Double	
reduction_total_tree	Equivalent trees planted	N/A	Double	
buyPower	Energy from grid	kWh	Double	
chargeCap	Charged energy	kWh	Double	
dischargeCap	Discharged energy	kWh	Double	
selfUsePower	Consumed PV energy	kWh	Double	
selfProvide	Energy consumed from PV	kWh	Double	
PVYield	PV yield	kWh	Double	
inverterYield	Inverter yield	kWh	Double	

Example

Request example:

```
{
    "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
    "collectTime":1501862400000
}
```

Response examples:

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20009,
    "params":{
        "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
        "collectTime":1501862400000,
```

```
      "currentTime":1503046597854
   },
   "message":null
}
```

Example 2: Yearly plant data is returned.

```
{
   "success":true,
   "data":[
      {
         "dataItemMap":{
            "use_power":288760,
            "radiation_intensity":0.6968,
            "reduction_total_co2":18.275,
            "reduction_total_coal":7.332,
            "inverter_power":null,
            "theory_power":17559.36,
            "ongrid_power":18330,
            "power_profit":34320,
            "installed_capacity":25200,
            "perpower_ratio":0.727,
            "reduction_total_tree":999,
            "performance_ratio":89
         },
         "stationCode":"5D02E8B40AD342159AC8D8A2BCD4FAB5",
         "collectTime":1483200000000
      },
      {
         "dataItemMap":{
            "use_power":null,
            "radiation_intensity":1.4123,
            "reduction_total_co2":0.897,
            "reduction_total_coal":0.36,
            "inverter_power":null,
            "theory_power":659.6,
            "ongrid_power":null,
            "power_profit":2088,
            "installed_capacity":467.04,
            "perpower_ratio":1.927,
            "reduction_total_tree":49,
            "performance_ratio":89
         },
         "stationCode":"BA4372D08E014822AB065017416F254C",
         "collectTime":1483200000000
      }
   ],
   "failCode":0,
   "params":{
      "stationCodes":"BA4372D08E014822AB065017416F254C,5D02E8B40AD342159AC8D8A2BCD4FAB5",
      "collectTime":1501862400000,
      "currentTime":1503046597854
   },
   "message":null
}
```

📖 NOTE

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.4.5 Daily Device Data API

API Description

This API is used to obtain daily device data. The daily data of a maximum of 100 devices of the same type can be queried at a time.

The backend calculates the month of the collection time based on the request parameter **collectTime** (collection time in milliseconds) and the time zone where the device is located. Then, you can query the daily data of the device by device ID in the current month. If data is generated for n ($0 \le n \le 31$) days of the month, n ($0 \le n \le 31$) results are returned.

For details about the data list that can be queried through this API, see the daily device data list below.

Request URL

https://*Domain name of the management system*/thirdData/getDevKpiDay

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
devIds	List of device IDs. The device IDs are obtained from **id** in **5.1.1.2 Device List API**. Use commas (,) to separate multiple device IDs. Either **sns** or **devIds** must be set.	String	Optional
sns	Device SN list. Multiple device SNs are separated by commas (,). Either **sns** or **devIds** must be set.	String	Optional
devTypeId	Device type ID. The values of **devTypeId** obtained in **5.1.1.2 Device List API** are used. The following device types are supported: 1: string inverter 38: residential inverter 39: battery 41: ESS	Integer	Mandatory
collectTime	Collection time, in milliseconds	Long	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-

Parameter		Description	Data Type	Remarks
	devIds	Device ID list in the request parameter	String	-
	sns	Device SN list in the request parameter	String	-
	devTypeId	Device type ID in the request parameter	Integer	-
	collectTime	Collection time in milliseconds in the request parameter	Long	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data	Parameters	Returned data. The data contains the daily data object list of each device.	List	List of daily device data in a month
	devId	Device ID	Long	-
	sn	Device SN	String	
	collectTime	Collection time, in milliseconds	Long	-
	dataItemMap	Content of each data item, which is returned in the key-value format. The data item content varies depending on the device type. For details about the data item list, see the daily device data list below.	Map	Data of a device in a day

Daily Device Data List

Device Type	Key	Name	Unit	Return Value Type
ID: 39 Residential battery	charge_cap	Charged energy	kWh	Double
	discharge_cap	Discharged energy	kWh	Double
	charge_time	Charging duration	h	Double

Device Type	Key	Name	Unit	Return Value Type
	discharge_time	Discharging duration	h	Double
ID: 1 String inverter	installed_capacity	Installed capacity	kW	Double
	product_power	Yield	kWh	Double
	perpower_ratio	Specific energy (kWh/kWp)	h	Double
ID: 38 Residential inverter	installed_capacity	Installed capacity	kW	Double
	product_power	Yield	kWh	Double
	perpower_ratio	Specific energy (kWh/kWp)	h	Double
ID: 41 C&I and utility ESS	charge_cap	Charged energy	kWh	Double
	discharge_cap	Discharged energy	kWh	Double

Examples

Request example:

```
{
    "devIds":"214060404588862,213472461631079",
    "devTypeId":1,
    "collectTime":1501862400000
}
```

Response example:

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20009,
    "params":{
        "devIds":"214060404588862,213472461631079",
        "devTypeId":1,
        "collectTime":1501862400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: Daily device data is returned.

```
{
    "success":true,
    "data":[
        {
            "dataItemMap":{
                "installed_capacity":30.24,
                "perpower_ratio":9.921,
```

```
            "product_power":300
        },
        "devId":213472461631079,
        "collectTime":1501776000000
    },
    {
        "dataItemMap":{
            "installed_capacity":30.24,
            "perpower_ratio":0.543,
            "product_power":16.43
        },
        "devId":214060404588862,
        "collectTime":1501776000000
    }
],
"failCode":0,
"params":{
    "devIds":"214060404588862,213472461631079",
    "devTypeId":1,
    "collectTime":1501862400000,
    "currentTime":1503046597854
},
"message":null
}
```

📖 **NOTE**

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.4.6 Monthly Device Data API

API Description

This API is used to obtain monthly device data. The monthly data of a maximum of 100 devices of the same type can be queried at a time.

The backend calculates the year of the collection time based on the request parameter **collectTime** (collection time in milliseconds) and the time zone where the device is located. Then, you can query the monthly data of the device by device ID in the current year. If data is generated for n ($0 \le n \le 12$) months of the year, n ($0 \le n \le 12$) results are returned.

For details about the data list that can be queried through this API, see the monthly device data list below.

Request URL

https://*Domain name of the management system*/thirdData/getDevKpiMonth

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
devIds	List of device IDs. The device IDs are obtained from **id** in **5.1.1.2 Device List API**. Use commas (,) to separate multiple device IDs. Either **sns** or **devIds** must be set.	String	Optional
sns	Device SN list. Multiple device SNs are separated by commas (,). Either **sns** or **devIds** must be set.	String	Optional
devTypeId	Device type ID. The values of **devTypeId** obtained in **5.1.1.2 Device List API** are used. The following device types are supported: 1: string inverter 38: residential inverter 39: battery 41: ESS	Integer	Mandatory
collectTime	Collection time, in milliseconds	Long	Mandatory

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-
	devIds	Device ID list in the request parameter	String	-
	sns	Device SN list in the request parameter	String	-
	devTypeId	Device type ID in the request parameter	Integer	-
	collectTime	Collection time in milliseconds in the request parameter	Long	-
	currentTime	Current system time, in milliseconds	Long	-
message		Optional message	String	-
data	Parameters	Returned data. The data contains the monthly data object list of each device.	List	List of monthly device data in a year
	devId	Device ID	Long	-
	sn	Device SN	String	
	collectTime	Collection time, in milliseconds	Long	-

Parameter		Description	Data Type	Remarks
	dataItemMap	Content of each data item, which is returned in the key-value format. The data item content varies depending on the device type. For details about the data item list, see the monthly device data list below.	Map	Data of a device in a month

Monthly Device Data List

Device Type	Key	Name	Unit	Return Value Type
ID: 39 Residential battery	charge_cap	Charged energy	kWh	Double
	discharge_cap	Discharged energy	kWh	Double
	charge_time	Charging duration	h	Double
	discharge_time	Discharging duration	h	Double
ID: 1 String inverter	installed_capacity	Installed capacity	kW	Double
	product_power	Yield	kWh	Double
	perpower_ratio	Specific energy (kWh/kWp)	h	Double
ID: 38 Residential inverter	installed_capacity	Installed capacity	kW	Double
	product_power	Yield	kWh	Double
	perpower_ratio	Specific energy (kWh/kWp)	h	Double
ID: 41 C&I and utility ESS	charge_cap	Charged energy	kWh	Double
	discharge_cap	Discharged energy	kWh	Double

Examples

Request example:

```
{
    "devIds":"214060404588862,213472461631079",
    "devTypeId":1,
    "collectTime":1501862400000
}
```

Response example:

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20009,
    "params":{
        "devIds":"214060404588862,213472461631079",
        "devTypeId":1,
        "collectTime":1501862400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: Monthly device data is returned.

```
{
    "success":true,
    "data":[
        {
            "dataItemMap":{
                "installed_capacity":30.24,
                "perpower_ratio":null,
                "product_power":300
            },
            "devId":213472461631079,
            "collectTime":1501516800000
        },
        {
            "dataItemMap":{
                "installed_capacity":30.24,
                "perpower_ratio":null,
                "product_power":16.43
            },
            "devId":214060404588862,
            "collectTime":1501516800000
        }
    ],
    "failCode":0,
    "params":{
        "devIds":"214060404588862,213472461631079",
        "devTypeId":1,
        "collectTime":1501862400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.1.4.7 Yearly Device Data API

API Description

This API is used to obtain yearly device data. The yearly data of a maximum of 100 devices of the same type can be queried at a time.

The backend queries the data of each year since the device was connected based on the device ID.

For details about the data list that can be queried using this API, see **#EN-US_TOPIC_0000001701648221/p2057810145254**.

Request URL

https://*Domain name of the management system*/thirdData/getDevKpiYear

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
devIds	List of device IDs. The device IDs are obtained from **id** in **5.1.1.2 Device List API**. Use commas (,) to separate multiple device IDs. Either **sns** or **devIds** must be set.	String	Optional
sns	Device SN list. Multiple device SNs are separated by commas (,). Either **sns** or **devIds** must be set.	String	Optional
devTypeId	Device type ID The following device types are supported: 1: string inverter 38: residential inverter 39: battery 41: ESS	Integer	Mandatory
collectTime	Collection time, in milliseconds	Long	Mandatory

☐ NOTE

Before obtaining data, you must configure related counters.

Response Packet

Parameter		Description	Data Type	Remarks
success		Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode		Error code Value **0** indicates that the status is normal. For definitions of other error codes, see **6.1 Error Code List**.	Integer	-
params	Parameters	-	-	-

Parameter		Description	Data Type	Remarks
	devIds	Device ID list in the request parameter	String	-
	sns	Device SN list in the request parameter	String	-
	devTypeId	Device type ID in the request parameter	Integer	-
	collectTime	Collection time in milliseconds in the request parameter	Long	-
	currentTime	Current system time, in milliseconds	Long	-
message	-	Optional message	String	-
data	Parameters	Returned data. The data contains the yearly data object list of each device.	List	Data list of each year since the device is connected
	devId	Device ID	Long	-
	sn	Device SN	String	
	collectTime	Collection time, in milliseconds	Long	-
	dataItemMap	Content of data items, which are returned in the key-value format. The content of data items varies according to device types. For details about the data item list, see **#EN-US_TOPIC_0000001701648221/p2057810145254**.	Map	Data of a device in a year

Yearly Device Data List

Device Type	Key	Name	Unit	Return Value Type
ID: 39 Residential battery	charge_cap	Charged energy	kWh	Double
	discharge_cap	Discharged energy	kWh	Double
	charge_time	Charging duration	h	Double
	discharge_time	Discharging duration	h	Double
ID: 1 String inverter	installed_capacity	Installed capacity	kW	Double
	product_power	Energy yield	kWh	Double
	perpower_ratio	Specific energy (kWh/kWp)	h	Double
ID: 38 Residential inverter	installed_capacity	Installed capacity	kW	Double
	product_power	Energy yield	kWh	Double
	perpower_ratio	Specific energy (kWh/kWp)	h	Double
ID: 41 C&I and utility ESS	charge_cap	Charged energy	kWh	Double
	discharge_cap	Discharged energy	kWh	Double

Examples

Request example:

```
{
    "devIds":"214060404588862,213472461631079",
    "devTypeId":1,
    "collectTime":1501862400000
}
```

Response example:

Example 1: An error code is returned.

```
{
    "success":false,
    "data":null,
    "failCode":20009,
    "params":{
        "devIds":"214060404588862,213472461631079",
        "devTypeId":1,
        "collectTime":1501862400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

Example 2: Yearly device data is returned.

```
{
    "success":true,
```

```
    "data":[
        {
            "dataItemMap":{
                "installed_capacity":30.24,
                "perpower_ratio":null,
                "product_power":300
            },
            "devId":213472461631079,
            "collectTime":1501516800000
        }
    ],
    "failCode":0,
    "params":{
        "devIds":"214060404588862,213472461631079",
        "devTypeId":1,
        "collectTime":1501862400000,
        "currentTime":1503046597854
    },
    "message":null
}
```

📖 **NOTE**

Prerequisites for obtaining data: The API account has the permission to access this API.

Request example:

5.2 Control APIs

Plants and devices can be remotely controlled through open APIs. This function is available only in OAuth Connect mode.

5.2.1 API for Delivering Battery Charge and Discharge Tasks

API Description

This API is used to deliver battery charge and discharge tasks based on plant codes. A task can be delivered to a maximum of 100 plants at a time. If there are multiple ESSs in the power plant, the task is executed on every ESS.

The battery charge and discharge task does not change the original working mode of the battery. After the task is complete, the battery continues to work in the original working mode.

Call the API only when necessary to reduce the access frequency. For the same PV plant, do not call this API repeatedly before a task is complete.

For details about the scenario-based practices of this API, see **7.5 Scenario-based Practices of Battery Scheduling**.

Request URL

https://*Domain name of the management system*/rest/openapi/pvms/nbi/v2/control/charge-and-discharge/async-task

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
tasks	List of battery charge and discharge tasks. A maximum of 100 plants can be delivered at a time.	List	M
> plantCode	Plant code, which is obtained from **plantCode** in **5.1.1.1 Plant List API**.	String	M

Parameter	Description	Data Type	Mandatory/ Optional
> dispatchSwitch	Charge/Discharge switch. 0: stop forced charge and discharge 1: forced charge 2: forced discharge	Integer	M
> controlType	1: SOC control. The target SOC is set in the forced charge/discharge command. Legacy versions may need an update to support SOC control. 2: duration control. The duration is set in the forced charge/ discharge command.	Integer	Optional for stopping forced charge/discharge.
> targetSOC	Target SOC for charge/discharge, in percentage.	Double	Optional. This parameter is mandatory for SOC control.
> dispatchTime	Charge/Discharge duration in minutes. Value range: [0,1440]	Integer	Optional. This parameter is mandatory for time control.
> powerDispatch	Power of forced charge and discharge in watt. If the value exceeds the range, the maximum value is used. The value should be greater than 0 during forced charge and smaller than 0 during forced discharge.	Integer	Optional. If this parameter is left blank, the default power is used for charge and discharge. Range of **powerDispatch**: [– maximum discharge power, maximum charge power]

> [!NOTE]
> **NOTICE**
> - This API will change the device running parameters. Exercise caution when invoking this API.
> - The LG battery does not support SOC control.
> - When a task is not complete (that is, the status bit is 1), the same signals cannot be sent to the device corresponding to the task.
> - This API can only be used to control the residential ESS.
> - When the battery is disconnected, charge and discharge tasks cannot be delivered to the battery.

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag **true**: The request succeeded. **false**: The request failed.	Boolean	Request success or failure flag
failCode	Error code **0**: successful; **1**: partially successful; **2**: failed	Integer	-
message	Description of the API returned content	String	-
data	Returned data for each request.	-	-
> requestID	Unique ID of a requested task, which can be used to query the task delivery result.	Long	-
> result	Task result	List	-
>> plantCode	Plant ID	String	-
>> sn	Inverter SN	String	-
>> dispatchResult	Charge/Discharge task delivery result	Integer	**0**: The task is received successfully. **1**: The task fails to be received.
>> subTaskId	Unique subtask ID	String	-

Parameter	Description	Data Type	Remarks
>> description	Delivery result description	String	-

API Error Code List

No.	Error Code	Description
1	305	You are not online and need to log in again.
2	401	You do not have the related data API permission.
3	407	The API access frequency is too high.
4	1	The task is partially successful.
5	2	The task fails.

Examples

Request example (the first task is time control and the second task is SOC control.)

```
{
 "tasks": [
  {
    "plantCode": "NE=12345678",
    "dispatchSwitch": 1,
    "controlType": 2,
    "dispatchTime": 600,
    "powerDispatch": 5000
  },
  {
    "plantCode": "NE=23456789",
    "dispatchSwitch": 1,
    "controlType": 1,
    "targetSOC": 100,
    "powerDispatch": 5000
  }
 ]
}
```

Response examples:

Example 1: The task fails.

```
{
   "data": null,
   "failCode": 2,
   "message": "Battery charge and discharge task list is empty.",
   "success": false
}
```

Example 2: The task fails.

```
{
   "data": {
      "requestID": null,
```

```
        "result": [
           {
              "plantCode": "NE=12345678",
              "sn": null,
              "dispatchResult": 1,
              "subTaskId": null,
              "description": "PowerDispatch is illegal. The value should be greater than 0 during forced charge
(dispatchSwitch = 1) and smaller than 0 during forced discharge (dispatchSwitch = 2)."
           },
           {
              "plantCode": "NE=23456789",
              "sn": null,
              "dispatchResult": 1,
              "subTaskId": null,
              "description": "DispatchSwitch is illegal, valid value: 0 or 1 or 2"
           }
        ]
     },
     "failCode": 2,
     "message": "Failed to deliver the battery charge/discharge task.",
     "success": false
}
```

Example 3: The task is partially successful.

```
{
   "data": {
      "requestID": 1107541629693466,
      "result": [
         {
            "plantCode": "NE=12345678",
            "sn": null,
            "dispatchResult": 1,
            "subTaskId": null,
            "description": "PowerDispatch is illegal. The value should be greater than 0 during forced charge
(dispatchSwitch = 1) and smaller than 0 during forced discharge (dispatchSwitch = 2)."
         },
         {
            "plantCode": "NE=23456789",
            "sn": "6fbfk11",
            "dispatchResult": 0,
            "subTaskId": "2004882110006589",
            "description": null
         }
      ]
   },
   "failCode": 1,
   "message": "The battery charge/discharge task is partially delivered successfully.",
   "success": true
}
```

Example 4: The task is successful.

```
{
   "data": {
      "requestID": 9694030627470695,
      "result": [
         {
            "plantCode": "NE=12345678",
            "sn": "5fbfk4",
            "dispatchResult": 0,
            "subTaskId": "4203991658344402",
            "description": null
         },
         {
            "plantCode": "NE=23456789",
            "sn": "6fbfk11",
            "dispatchResult": 0,
            "subTaskId": "1828180425476906",
            "description": null
```

```
        }
      ]
    },
    "failCode": 0,
    "message": "The battery charge/discharge task is delivered successfully.",
    "success": true
}
```

5.2.2 API for Querying Battery Charge and Discharge Tasks

API Description

This API is used to query the execution status of battery charge and discharge tasks based on requestID. One task can be queried at a time.

Call the API only when necessary to reduce the access frequency. For the same PV plant, do not call this API repeatedly before a task is complete.

Request URL

https://Domain name of the management system/rest/openapi/pvms/v1/vpp/chargeAndDischargeStatus

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
requestID	Unique ID of the requested task. The value can be obtained from *requestID* in **5.2.1 API for Delivering Battery Charge and Discharge Tasks**.	Long	M

> **NOTICE**
>
> ● If the charging and discharging task information of the battery in the plant changes, the task is processed as an exception (**status** is **0**). The exception information is described in **message**.

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag true: The request succeeded. false: The request failed.	Boolean	Request success or failure flag
failCode	Error code Value **0** indicates that the status is normal. For other error codes, see **API Error Code List**.	Integer	-
message	Optional message	String	-
data	Returned data for each request.	Map	-
> plantCode	Plant ID	String	-
> sn	Inverter SN	String	-
> remoteID	Unique subtask ID	String	-
> status	Event status. The execution status is updated every 3 minutes. If the task is not completed within 24 hours, the task times out.	Integer	**0**: complete **1**: in progress **2**: timeout or failure
> chargedCapacity	Amount of power that has been forcibly charged into batteries. If **dispatchSwitch** is not 1, null is returned.	Double	kWh
> dischargedCapacity	Amount of power that has been forcibly discharged from batteries. If **dispatchSwitch** is not 2, null is returned.	Double	kWh
> execStartTime	Time when a task is received, including the time zone information	String	2020-02-06T00:00:00+08:00

Parameter	Description	Data Type	Remarks
> execEndTime	Time when a task is completed, including the time zone information. If a task is not completed, null is returned.	String	2020-02-06T00:00:00+08:00
> message	Task result description	String	-

API Error Code List

No.	Error Code	Description
1	305	You are not online and need to log in again.
2	401	You do not have the related data API permission.
3	407	The API access frequency is too high.
4	20044	The unique ID of a charge/discharge task cannot be empty.
5	20050	The charge/discharge task query parameter does not exist.

Examples

Request example:

```
{
  "requestID": 432523532523
}
```

Response examples:

Example 1: An error code is returned.

```
{
  "success": false,
  "data": null,
  "failCode": 20008,
  "message": null
}
```

Example 2: The task status data is returned.

```
{
  "success":true,
  "failCode":0,
  "message":null,
  "data":[
    {
      "plantCode":"NE=12345678",
      "sn":"5fbfk4",
      "remoteID":"12345678",
      "status":0,
      "chargedCapacity":1000,
```

```
            "message":null,
            "execStartTime":"2020-02-06T00:00:10+08:00",
            "execEndTime":"2020-02-06T00:01:10+08:00"
        },
        {
            "plantCode":"NE=23456789",
            "sn":"6fbfk11",
            "remoteID":"23456789",
            "status":0,
            "chargedCapacity":2000,
            "message":null,
            "startTime":"2020-02-06T00:00:00+08:00",
            "endTime":"2020-02-06T00:01:00+08:00"
        }
    ]
}
```

5.2.3 API for Delivering a Task for Setting the Battery Working Mode

API Description

This API is used to set the working mode of plant-level batteries based on the plant DN. A task supports a maximum of 10 plants. Different parameters can be delivered to each plant.

The networking of a single controller (Dongle, EMMA, distributed SmartLogger, and direction connection of inverters) in a plant is supported. The battery working mode can be set to **maximumSelfConsumption** or **TOU**.

Call the API only when necessary to reduce the access frequency. For the same PV plant, do not call this API repeatedly before a task is complete.

Request URL

https://*Domain name of the management system*/rest/openapi/pvms/nbi/v1/
control/battery/mode/async-task

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
tasks	List of the tasks for setting the battery working mode. A task can be delivered to a maximum of 10 plants at a time.	List	M
> plantCode	Plant DN.	String	M

Parameter	Description	Data Type	Mandatory/ Optional
> operationMode	Working mode. TOU: time of use maximumSelfConsumption: maximum self-consumption If this parameter is set to **TOU**, you can manually set the charge and discharge time segments. This mode applies to the PV+ESS system and ESS-only system where electricity prices at peak and off-peak hours are different and power meters are available. If this parameter is set to **maximumSelfConsumption**, PV power is preferentially supplied to loads and the surplus power is used to charge batteries. If the ESS is fully charged or is being charged at full power, the surplus PV energy is fed to the power grid. When PV power is insufficient or no PV power can be generated at night, the ESS discharges power to loads. This improves the self-consumption rate and energy self-sufficiency rate, and reduces electricity costs. The grid cannot charge the ESS but can supply power to loads. This mode applies to areas where the electricity price is high, or areas	String	M

Parameter	Description	Data Type	Mandatory/ Optional
	where the FIT subsidy is low or unavailable. The PV+ESS system generates sufficient PV power for loads and uses the surplus PV power to charge the ESS (if the PV power is insufficient for loads, the TOU mode is recommended).		

Parameter	Description	Data Type	Mandatory/ Optional
> redundantPVEner-gyPriority	Preferred use of surplus PV power. fedToGridPreference: The power is fed to the grid preferentially. chargePreference: Charging is preferred. **fedToGridPreference**: When the PV power is greater than the load power, the surplus PV energy is preferentially fed to the grid. When the maximum output power of the device is reached, the surplus energy is used to charge the batteries. This setting is applicable to the scenario where the FIT is higher than the electricity price and the grid cannot charge the ESS. **chargePreference**: When the PV power is greater than the load power, the surplus PV energy is used to charge the batteries. After the maximum charge power is reached or the batteries are fully charged, the surplus PV energy is fed to the grid.	String	Optional (This parameter is available when **operationMode** is set to **TOU**.)

Parameter	Description	Data Type	Mandatory/ Optional
> allowedAcCharge-Power	Maximum power for charging batteries from grid (kW). The value range of this parameter is related to the controller in the plant. The value ranges are as follows: Dongle: [0.000, 30.000] EMMA: [0.000, 50.000] SmartLogger: [0.000, 50000.000] Inverter: [0.000, upper limit for the maximum power for charging batteries from grid] This parameter is used to set the maximum power for charging batteries from the power grid.	Double	Optional (This parameter is available when **operationMode** is set to **TOU**.)
> chargingAndDischargingTimeWindow	Charge/Discharge time window. Set the start time and end time of charge and discharge. A maximum of 14 time segments can be set. For the same plant, multiple overlapped time segments cannot be set on the same day.	List	Optional (This parameter must be set when **operationMode** is set to **TOU**.)
>> startTime	Start time, in HH:MM format. Value range: 00:00–23:59	String	M
>> endTime	End time, in HH:MM format. Value range: 00:00–23:59	String	M

Parameter	Description	Data Type	Mandatory/ Optional
>> chargeOrDischarge	Charge/Discharge Charge Discharge	String	M
>> repeat	Repeat date. A maximum of seven dates can be set for each plant. 1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Sunday	List	M

NOTICE

- This API will change the device running parameters. Exercise caution when calling this API.
- Do not send a new task to the same plant when the task is not complete (that is, the status is **RUNNING**).
- For the same plant, multiple overlapped time segments cannot be set on the same day.
- The battery working mode cannot be set for LG batteries.
- For a PV plant, if **operationMode** is set to **TOU**, the **chargingAndDischargingTimeWindow** parameter must be set.
- When the controller in the plant is disconnected, the task of setting the battery working mode cannot be delivered to the plant.

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag. **true**: The request succeeded. **false**: The request failed.	boolean	Request success or failure flag

Parameter	Description	Data Type	Remarks
failCode	Error code. 0: successful; 1: partially successful; 2: failed	Integer	-
message	Description of the API returned content.	String	-
data	Returned data for each request, including the following information:	-	-
> taskId	Unique ID of a requested task, which can be used to query the task delivery result.	String	-
> result	Task delivery result.	List	-
>> plantCode	Plant DN.	String	-
>> status	Status of the task delivered to the plant.	String	**RUNNING**: The task delivered to the plant is running. **FAIL**: The task fails to deliver to the plant.
>> message	Delivery result description.	String	-

Example

Request example (The first task is set to **maximumSelfConsumption** and the second task is set to **TOU**.):

```
{
   "tasks":[
      {
         "plantCode":"NE=123456789",
         "operationMode":"maximumSelfConsumption"
      },
      {
         "plantCode":"NE=234567891",
         "operationMode":"TOU",
         "redundantPVEnergyPriority":"fedToGridPreference",
         "allowedAcChargePower":10,
         "chargingAndDischargingTimeWindow":[
            {
               "startTime":"00:00",
               "endTime":"02:00",
               "chargeOrDischarge":"charge",
               "repeat":[
                  2,
                  3
               ]
            },
```

```
            {
               "startTime":"00:00",
               "endTime":"05:00",
               "chargeOrDischarge":"charge",
               "repeat":[
                  5,
                  6
               ]
            }
         ]
      }
   ]
}
```

Response example:

Example 1: The task fails.

```
{
   "failCode": 2,
   "success": false,
   "message": "Exist no permission resources. plantCodes: [NE=123456789, NE=234567891]",
   "data": null
}
```

Example 2: The task fails.

```
{
   "failCode": 2,
   "success": false,
   "message": "Failed to deliver the battery mode control task.",
   "data": {
      "taskId": null,
      "result": [
         {
            "plantCode": "NE=123456789",
            "status": "FAIL",
            "message": "The input parameter is out of the valid range, operationMode in:
[maximumSelfConsumption, TOU]"
         },
         {
            "plantCode": "NE=234567891",
            "status": "FAIL",
            "message": "The startTime and endTime of multiple days overlap."
         }
      ]
   }
}
```

Example 3: The task fails.

```
{
   "exceptionId":"framwork.remote.Paramerror",
   "exceptionType":"ROA_EXFRAME_EXCEPTION",
   "descArgs":null,
   "reasonArgs":[
      "tasks"
   ],
   "detailArgs":[
      "tasks size must be between 1 and 10"
   ],
   "adviceArgs":null
}
```

Example 4: The task is partially successful.

```
{
   "failCode": 1,
   "success": true,
```

```
   "message": "The battery mode task is partially delivered successfully.",
   "data": {
      "taskId": "5882491459219785",
      "result": [
         {
            "plantCode": "NE=123456789",
            "status": "FAIL",
            "message": "The configured value is the same as the current value."
         },
         {
            "plantCode": "NE=234567891",
            "status": "RUNNING",
            "message": null
         }
      ]
   }
}
```

Example 5: The task is successful.

```
{
   "failCode": 0,
   "success": true,
   "message": "The battery mode task is delivered successfully.",
   "data": {
      "taskId": "2839011389499814",
      "result": [
         {
            "plantCode": "NE=123456789",
            "status": "RUNNING",
            "message": null
         },
         {
            "plantCode": "NE=234567891",
            "status": "RUNNING",
            "message": null
         }
      ]
   }
}
```

5.2.4 API for Querying a Task for Setting the Battery Working Mode

API Description

This API is used to query the execution status of the task for setting the battery working mode based on the task ID. Only one task can be queried at a time.

Call the API only when necessary to reduce the access frequency. For the same PV plant, do not call this API repeatedly before a task is complete.

Request URL

https://*Domain name of the management system*/rest/openapi/pvms/nbi/v1/control/battery/mode/task-info

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
taskId	Unique ID of the requested task, which can be obtained from **taskId** in **5.2.3 API for Delivering a Task for Setting the Battery Working Mode**. The value contains 16 characters.	String	M

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag. The value **true** indicates that the request is successful, and the value **false** indicates that the request fails.	boolean	Request success or failure flag
failCode	Error code. Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
message	Request description. When the value of **failCode** is **0**, the request failure description is returned. In other cases, null is returned.	String	-
data	Returned data for each request, including the following information:	Object	-
> dispatchResult	dispatchResult contains the list of returned information about request execution, including the following information:	List	-
>> plantCode	Plant DN.	String	-

Parameter	Description	Data Type	Remarks
>> status	Status of the task delivered to the plant.	String	**RUNNING**: The task delivered to the plant is running. **SUCCESS**: The task is successfully delivered to the plant. **FAIL**: The task fails to deliver to the plant.
>> message	Request description. When the value of **status** is **FAIL**, the description of the plant delivery failure is returned. In other cases, no description is returned.	String	**FAILURE**: The delivery fails. **TIMEOUT**: A timeout occurs. **BUSY**: The device is busy. **INVALID**: The device is invalid. **EXCEPTION**: An exception occurs.
>> operationMode	Working mode.	String	TOU: time of use maximumSelfConsumption: maximum self-consumption
>> redundantPVEnergyPriority	Preferred use of surplus PV power.	String	fedToGridPreference: The power is fed to the grid preferentially. chargePreference: Charging is preferred.
>> allowedAcChargePower	Maximum power for charging batteries from grid (kW).	Double	-
>> chargingAndDischargingTimeWindow	Charge/Discharge time window.	List	-
>>> startTime	Start time.	String	Format: HH:MM
>>> endTime	End time.	String	Format: HH:MM

Parameter	Description	Data Type	Remarks
>>> chargeOrDischarge	Charge/Discharge	String	Charge Discharge
>>> repeat	Repeated date.	List	1: Monday 2: Tuesday 3: Wednesday 4: Thursday 5: Friday 6: Saturday 7: Sunday
> startTime	Time when a task is received, including the time zone information.	String	2020-02-06T00:00:00+08:00
> endTime	Time when a task is completed, including the time zone information. If a task is not completed, null is returned.	String	2020-02-06T00:00:00+08:00

Example

Request example:

```
{
  "taskId": "3051190140256431"
}
```

Response examples:

Example 1: An error code is returned.

```
{
    "failCode": 20620,
    "success": false,
    "message": "The taskId is not exist.",
    "data": null
}
```

Example 2: An exception is returned.

```
{
    "exceptionId": "framwork.remote.Paramerror",
    "exceptionType": "ROA_EXFRAME_EXCEPTION",
    "descArgs": null,
    "reasonArgs": [
        "taskId"
    ],
    "detailArgs": [
        "taskId may not be null"
    ],
    "adviceArgs": null
}
```

Example 3: The returned task status is **RUNNING**.

```
{
    "failCode": 0,
    "success": true,
    "message": "The battery mode task status queried successfully.",
    "data": {
        "dispatchResult": [
            {
                "plantCode": "NE=234567891",
                "status": "RUNNING",
                "message": null,
                "operationMode": "TOU",
                "redundantPVEnergyPriority": "fedToGridPreference",
                "allowedAcChargePower": 10,
                "chargingAndDischargingTimeWindow": [
                    {
                        "chargeOrDischarge": "charge",
                        "repeat": [
                            2,
                            3
                        ],
                        "startTime": "00:00",
                        "endTime": "02:00"
                    },
                    {
                        "chargeOrDischarge": "charge",
                        "repeat": [
                            5,
                            6
                        ],
                        "startTime": "00:00",
                        "endTime": "05:00"
                    }
                ]
            }
        ],
        "startTime": "2024-02-29T23:02:49+08:00",
        "endTime": null
    }
}
```

Example 4: The returned task status is **FAIL**.

```
{
    "failCode": 0,
    "success": true,
    "message": "The battery mode task status queried successfully.",
    "data": {
        "dispatchResult": [
            {
                "plantCode": "NE=234567891",
                "status": "FAIL",
                "message": "TIMEOUT",
                "operationMode": "TOU",
                "redundantPVEnergyPriority": "fedToGridPreference",
                "allowedAcChargePower": 10,
                "chargingAndDischargingTimeWindow": [
                    {
                        "chargeOrDischarge": "charge",
                        "repeat": [
                            2,
                            3
                        ],
                        "startTime": "00:00",
                        "endTime": "02:00"
                    },
                    {
                        "chargeOrDischarge": "charge",
                        "repeat": [
```

```
                    5,
                    6
                ],
                "startTime": "00:00",
                "endTime": "05:00"
              }
            ]
          }
        ]
      },
      "startTime": "2024-02-29T23:02:49+08:00",
      "endTime": "2024-02-29T23:02:55+08:00"
    }
}
```

Example 5: The returned task status is **SUCCESS**.

```
{
    "failCode": 0,
    "success": true,
    "message": "The battery mode task status queried successfully.",
    "data": {
      "dispatchResult": [
        {
          "plantCode": "NE=234567891",
          "status": "SUCCESS",
          "message": null,
          "operationMode": "TOU",
          "redundantPVEnergyPriority": "fedToGridPreference",
          "allowedAcChargePower": 10,
          "chargingAndDischargingTimeWindow": [
            {
              "chargeOrDischarge": "charge",
              "repeat": [
                2,
                3
              ],
              "startTime": "00:00",
              "endTime": "02:00"
            },
            {
              "chargeOrDischarge": "charge",
              "repeat": [
                5,
                6
              ],
              "startTime": "00:00",
              "endTime": "05:00"
            }
          ]
        }
      ],
      "startTime": "2024-02-29T23:02:49+08:00",
      "endTime": "2024-02-29T23:02:55+08:00"
    }
}
```

5.2.5 API for Delivering a Task for Setting Battery Parameters

API Description

This API is used to deliver the task for setting plant-level energy battery parameters (end-of-charge SOC, end-of-discharge SOC, maximum charge power, and maximum discharge power) based on plant DNs. A task supports a maximum of 10 plants. Different parameters can be delivered to each plant.

Call the API only when necessary to reduce the access frequency. For the same PV plant, do not call this API repeatedly before a task is complete.

Request URL

https://*Domain name of the management system*/rest/openapi/pvms/nbi/v1/
control/battery/configuration/async-task

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/ Optional
tasks	List of battery parameter setting tasks. A task can be delivered to a maximum of 10 plants at a time.	List	M
> plantCode	Plant DN.	String	M
> batteryConfigur-ationInfo	Valid parameters for setting battery parameters.	Object	M
>> endOfChargeSoc	End-of-charge SOC (unit: %). Value range: [90.0, 100.0]	Double	Optional (At least one of the four parameters must be set.)
>> endOfDischargeSoc	End-of-discharge SOC (unit: %). Value range: [0.0, 20.0]	Double	
>> maximumChargePower	Maximum charge power (unit: W). Value range: [0, upper limit of maximum charge power] If the delivered maximum charge power is greater than the upper limit of the maximum charge power, the upper limit is used by default.	Integer	

Parameter	Description	Data Type	Mandatory/Optional
>> maximumDischargePower	Maximum discharge power (unit: W). Value range: [0, upper limit of maximum discharge power] If the delivered maximum discharge power is greater than the upper limit of the maximum discharge power, the upper limit is used by default.	Integer	

NOTICE

- This API will change the device running parameters. Exercise caution when calling this API.
- Do not send a new task to the same plant when the task is not complete (that is, the status is **RUNNING**).
- Battery parameters cannot be set for LG batteries.
- When the battery is disconnected, the parameter setting task cannot be delivered to the battery.

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag. **true**: The request succeeded. **false**: The request failed.	boolean	Request success or failure flag
failCode	Error code. 0: successful; 1: partially successful; 2: failed	Integer	-
message	Description of the API returned content.	String	-
data	Returned data for each request, including the following information:	-	-

Parameter	Description	Data Type	Remarks
> taskId	Unique ID of a requested task, which can be used to query the task delivery result.	String	-
> result	Task delivery result.	List	-
>> plantCode	Plant DN.	String	-
>> status	Status of the task delivered to the plant.	String	**RUNNING**: The task delivered to the plant is running. **FAIL**: The task fails to deliver to the plant.
>> message	Delivery result description.	String	-

Example

Request example:

```
{
    "tasks": [
        {
          "plantCode": "NE=12345678",
          "batteryConfigurationInfo": {
            "endOfDischargeSoc" : 18,
            "endOfChargeSoc": 98,
            "maximumDischargePower": 48,
            "maximumChargePower": 22
          }
        },
        {
          "plantCode": "NE=23456789",
          "batteryConfigurationInfo": {
            "endOfDischargeSoc" : 18
          }
        },
        {
          "plantCode": "NE=34567890",
          "batteryConfigurationInfo": {
            "endOfChargeSoc": 98
          }
        },
        {
          "plantCode": "NE=45678901",
          "batteryConfigurationInfo": {
            "maximumDischargePower": 48,
            "maximumChargePower": 22
          }
        }
    ]
}
```

Response examples:

Example 1: The task fails.

```
{
    "failCode": 2,
    "success": false,
    "message": "Exist no permission plants, plantCodes: [NE=45111870, NE=56433603]"
}
```

Example 2: The task fails.

```
{
    "failCode": 2,
    "success": false,
    "message": "Failed to deliver the battery configuration task.",
    "data": {
        "result": [
            {
                "plantCode": "NE=12345678",
                "status": "FAIL",
                "message": "Signal dispensing happened error."
            }
        ]
    }
}
```

Example 3: The task fails.

```
{
    "exceptionId": "framwork.remote.Paramerror",
    "exceptionType": "ROA_EXFRAME_EXCEPTION",
    "descArgs": null,
    "reasonArgs": [
        "tasks"
    ],
    "detailArgs": [
        "tasks size must be between 1 and 10"
    ],
    "adviceArgs": null
}
```

Example 4: The task is partially successful.

```
{
    "failCode": 1,
    "success": true,
    "message": "The battery configuration task is partially delivered successfully.",
    "data": {
        "taskId": "1276820029354826",
        "result": [
            {
                "plantCode": "NE=12345678",
                "status": "FAIL",
                "message": "No resource permission."
            },
            {
                "plantCode": "NE=23456789",
                "status": "RUNNING"
            }
        ]
    }
}
```

Example 5: The task is successful.

```
{
    "failCode": 0,
    "success": true,
    "message": "The battery configuration task is delivered successfully.",
    "data": {
        "taskId": "6033150147370043",
        "result": [
            {
```

```
                "plantCode": "NE=12345678",
                "status": "RUNNING"
            }
        ]
    }
}
```

5.2.6 API for Querying a Task for Setting Battery Parameters

API Description

This API is used to query the execution status of the task for setting the battery parameters based on the task ID. Only one task can be queried at a time.

Call the API only when necessary to reduce the access frequency. For the same PV plant, do not call this API repeatedly before a task is complete.

Request URL

https://*Domain name of the management system*/rest/openapi/pvms/nbi/v1/control/battery/configuration/task-info

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
taskId	Unique ID of the requested task, which can be obtained from **taskId** in **5.2.5 API for Delivering a Task for Setting Battery Parameters**. The value contains 16 characters.	String	M

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag. The value **true** indicates that the request is successful, and the value **false** indicates that the request fails.	boolean	Request success or failure flag

Parameter	Description	Data Type	Remarks
failCode	Error code. Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-
message	Request description. When the value of **failCode** is **0**, the request failure description is returned. In other cases, null is returned.	String	-
data	Returned data for each request, including the following information:	Object	-
> dispatchResult	dispatchResult contains the list of returned information about request execution, including the following information:	List	-
>> plantCode	Plant DN.	String	-
>> status	Status of the task delivered to the plant.	String	**RUNNING**: The task delivered to the plant is running. **SUCCESS**: The task is successfully delivered to the plant. **FAIL**: The task fails to deliver to the plant.
>> message	Request description. When the value of **status** is **FAIL**, the description of the plant delivery failure is returned. In other cases, no description is returned.	String	**FAILURE**: The delivery fails. **TIMEOUT**: A timeout occurs. **BUSY**: The device is busy. **INVALID**: The device is invalid. **EXCEPTION**: An exception occurs.

Parameter	Description	Data Type	Remarks
>> batteryConfiguration-tionInfo	Valid parameters for setting battery parameters.	Object	-
>>> endOfChargeSoc	End-of-charge SOC.	Double	-
>>> endOfDischargeSoc	End-of-discharge SOC.	Double	-
>>> maximumChargePower	Maximum charge power.	Integer	-
>>> maximumDischargePower	Maximum discharge power.	Integer	-
> startTime	Time when a task is received, including the time zone information.	String	2020-02-06T00:00:00+08:00
> endTime	Time when a task is completed, including the time zone information. If a task is not completed, null is returned.	String	2020-02-06T00:00:00+08:00

Example

Request example:

```
{
  "taskId": "3051190140256431"
}
```

Response examples:

Example 1: An error code is returned.

```
{
    "failCode": 20620,
    "success": false,
    "message": "The taskId is not exist.",
    "data": null
}
```

Example 2: An exception is returned.

```
{
    "exceptionId": "framwork.remote.Paramerror",
    "exceptionType": "ROA_EXFRAME_EXCEPTION",
    "descArgs": null,
    "reasonArgs": [
        "taskId"
```

```
    ],
    "detailArgs": [
        "taskId may not be null"
    ],
    "adviceArgs": null
}
```

Example 3: The returned task status is **RUNNING**.

```
{
    "failCode": 0,
    "success": true,
    "message": "The battery configuration task status queried successfully.",
    "data": {
        "dispatchResult": [
            {
                "plantCode": "NE=12345678",
                "status": "RUNNING",
                "message": null,
                "batteryConfigurationInfo": {
                    "endOfChargeSoc": 91.2,
                    "endOfDischargeSoc": 11.0,
                    "maximumChargePower": 59,
                    "maximumDischargePower": 48
                }
            }
        ],
        "startTime": "2024-02-29T18:49:42+08:00",
        "endTime": null
    }
}
```

Example 4: The returned task status is **FAIL**.

```
{
    "failCode": 0,
    "success": true,
    "message": "The battery configuration task status queried successfully.",
    "data": {
        "dispatchResult": [
            {
                "plantCode": "NE=12345678",
                "status": "FAIL",
                "message": "TIMEOUT",
                "batteryConfigurationInfo": {
                    "endOfChargeSoc": 91.0,
                    "endOfDischargeSoc": 11.0,
                    "maximumChargePower": 32,
                    "maximumDischargePower": 68
                }
            }
        ],
        "startTime": "2024-02-29T15:49:45+08:00",
        "endTime": "2024-02-29T17:17:18+08:00"
    }
}
```

Example 5: The returned task status is **SUCCESS**.
```
{
    "failCode": 0,
    "success": true,
    "message": "The battery configuration task status queried successfully.",
    "data": {
        "dispatchResult": [
            {
                "plantCode": "NE=12345678",
                "status": "SUCCESS",
                "message": null,
                "batteryConfigurationInfo": {
                    "endOfChargeSoc": 90.0,
```

```
                    "endOfDischargeSoc": 13.0,
                    "maximumChargePower": 22,
                    "maximumDischargePower": 0
                }
            }
        ],
        "startTime": "2024-02-29T18:01:06+08:00",
        "endTime": "2024-02-29T18:06:18+08:00"
    }
}
```

5.2.7 API for Delivering an Inverter Active Power Setting Task

API Description

This API is used to deliver the inverter active power setting tasks at the plant level. A task supports a maximum of 10 plants. Different parameters can be delivered to each plant.

The networking of a single controller (Dongle, EMMA, distributed SmartLogger, and direction connection of inverters) in a plant is supported. The active power can be controlled in two modes: unlimited and limited feed-in (kW).

Call the API only when necessary to reduce the access frequency. For the same PV plant, do not call this API repeatedly before a task is complete.

Request URL

https://*Domain name of the management system*/rest/openapi/pvms/nbi/v2/control/active-power-control/async-task

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
tasks	List of the tasks for setting the battery working mode. A task can be delivered to a maximum of 10 plants at a time.	List	M
> plantCode	Plant DN.	String	M

Parameter	Description	Data Type	Mandatory/ Optional
> controlMode	Active power control mode. 0: unlimited 6: limited feed-in (kW) If this parameter is set to **Unlimited**, the output power of the inverter is not limited and the inverter can connect to the grid at the rated power.	String	M
> controlInfo	Parameter for setting the active power.	Object	O If **controlMode** is set to **6**, you can set this parameter to set the active power. If **controlMode** is set to **0**, ignore this parameter.

Parameter	Description	Data Type	Mandatory/ Optional
>> maxGridFeedInPower	Maximum grid feed-in power (kW). The value range of this parameter is related to the controller in the plant. The value ranges are as follows: Dongle: [-1000.000, 5000.000] EMMA: [-1000.000, inverter rated power] SmartLogger: [-1000.000, 5000.000] Distributed SmartLogger: [-1000.000, 50000.000] Inverter: [-1000.000, 5000.000] It specifies the maximum active power transmitted from the grid-connection point to the power grid.	Double	O
>> limitationMode	Limitation mode. 0: total power 1: single-phase power **Total power**: controls the total power at the grid-connection point to limit the power fed to the power grid. **Single-phase power**: controls the power of each phase at the grid-connection point to limit the power fed to the power grid.	String	O

> **NOTICE**

- This API will change the device running parameters. Exercise caution when calling this API.

- Do not send a new task to the same plant when the task is not complete (that is, the status is **RUNNING**).

- For a PV plant, if **controlMode** is set to **0**, **controlInfo** will be ignored during task delivery.

- When the SmartLogger is used, SmartLogger (**devTypeId** = 2) and the distributed SmartLogger (**devTypeId** = 63) are distinguished. The **devTypeId** can be queried in **5.1.1.2 Device List API**. The maximum grid feed-in power of the SmartLogger falls in the range of [-1000.000, 5000.000], and that of the distributed SmartLogger falls in the range of [-1000.000, 50000.000].

- For a plant where SmartLogger is used, if **controlMode** is set to **6**, **Start control** is set to **Yes** by default after a task is delivered.

- When the controller in the plant is disconnected, the active power setting task cannot be delivered to the plant.

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag. **true**: The request succeeded. **false**: The request failed.	boolean	Request success or failure flag
failCode	Error code. 0: successful; 1: partially successful; 2: failed	Integer	-
message	Description of the API returned content.	String	-
data	Returned data for each request, including the following information:	-	-
> taskId	Unique ID of a requested task, which can be used to query the task delivery result.	String	-
> result	Task delivery result.	List	-
>> plantCode	Plant DN.	String	-

Parameter	Description	Data Type	Remarks
>> status	Status of the task delivered to the plant.	String	**RUNNING**: The task delivered to the plant is running. **FAIL**: The task fails to deliver to the plant.
>> message	Delivery result description.	String	-

Example

Example request (The active power control mode of the first task is **Unlimited**, and that of the second task is **Limited feed-in (kW).**):

```
{
    "tasks":[
        {
            "plantCode":"NE=123456789",
            "controlMode":"0"
        },
        {
            "plantCode":"NE=234567891",
            "controlMode":"6",
            "controlInfo":{
                "maxGridFeedInPower":53,
                "limitationMode":"1"
            }
        }
    ]
}
```

Response examples:

Example 1: The task fails.

```
{
    "failCode": 2,
    "success": false,
    "message": "Exist no permission resources. plantCodes: [NE=123456789, NE=234567891]",
    "data": null
}
```

Example 2: The task fails.

```
{
    "failCode": 2,
    "success": false,
    "message": "Failed to deliver the active power control task.",
    "data": {
        "taskId": null,
        "result": [
            {
                "plantCode": "NE=123456789",
                "status": "FAIL",
                "message": "The input parameter is out of the valid range, limitationMode in: [0, 1]"
            },
            {
                "plantCode": "NE=234567891",
                "status": "FAIL",
```

```
            "message": "The configured value is the same as the current value."
        }
    ]
   }
}
```

Example 3: The task fails.

```
{
   "exceptionId": "framwork.remote.Paramerror",
   "exceptionType": "ROA_EXFRAME_EXCEPTION",
   "descArgs": null,
   "reasonArgs": [
       "tasks"
   ],
   "detailArgs": [
       "tasks size must be between 1 and 10"
   ],
   "adviceArgs": null
}
```

Example 4: The task is partially successful.

```
{
   "failCode": 1,
   "success": true,
   "message": "The active power control task is partially delivered successfully.",
   "data": {
       "taskId": "1556481834749924",
       "result": [
          {
              "plantCode": "NE=123456789",
              "status": "FAIL",
              "message": "The configured value is the same as the current value."
          },
          {
              "plantCode": "NE=234567891",
              "status": "RUNNING",
              "message": null
          }
       ]
   }
}
```

Example 5: The task is successful.

```
{
   "failCode": 0,
   "success": true,
   "message": "The active power control task is delivered successfully.",
   "data": {
       "taskId": "8193701385705709",
       "result": [
          {
              "plantCode": "NE=123456789",
              "status": "RUNNING",
              "message": null
          },
          {
              "plantCode": "NE=234567891",
              "status": "RUNNING",
              "message": null
          }
       ]
   }
}
```

5.2.8 API for Querying Inverter Active Power Setting Tasks

API Description

This API is used to query the execution status of the inverter active power setting task based on the task ID. Only one task can be queried at a time.

Call the API only when necessary to reduce the access frequency. For the same PV plant, do not call this API repeatedly before a task is complete.

Request URL

https://management system domain name/rest/openapi/pvms/nbi/v2/control/active-power-control/task-info

Request Mode

HTTP method: POST

Request Parameters

Parameter	Description	Data Type	Mandatory/Optional
taskId	Unique ID of the requested task, which can be obtained from **taskId** in **5.2.7 API for Delivering an Inverter Active Power Setting Task**. The value contains 16 characters.	String	M

Response Packet

Parameter	Description	Data Type	Remarks
success	Request success or failure flag. The value **true** indicates that the request is successful, and the value **false** indicates that the request fails.	boolean	Request success or failure flag
failCode	Error code. Value **0** indicates that the status is normal. For other error codes, see **6.1 Error Code List**.	Integer	-

Parameter	Description	Data Type	Remarks
message	Request description. When the value of **failCode** is **0**, the request failure description is returned. In other cases, null is returned.	String	-
data	Returned data for each request, including the following information:	Object	-
> dispatchResult	dispatchResult contains the list of returned information about request execution, including the following information:	List	-
>> plantCode	Plant DN.	String	-
>> status	Status of the task delivered to the plant.	String	**RUNNING**: The task delivered to the plant is running. **SUCCESS**: The task is successfully delivered to the plant. **FAIL**: The task fails to deliver to the plant.
>> message	Request description. When the value of **status** is **FAIL**, the description of the plant delivery failure is returned. In other cases, no description is returned.	String	**FAILURE**: The delivery fails. **TIMEOUT**: A timeout occurs. **BUSY**: The device is busy. **INVALID**: The device is invalid. **EXCEPTION**: An exception occurs.
>> controlMode	Active power control mode.	String	0: unlimited 6: limited feed-in (kW)

Parameter	Description	Data Type	Remarks
>> controlInfo	Parameter for setting the active power.	Object	This parameter is returned when **controlMode** is set to **6**.
>>> maxGridFeedInPower	Maximum grid feed-in power (kW). It specifies the maximum active power transmitted from the grid-connection point to the power grid.	Double	The value range of this parameter is related to the controller in the plant. The value ranges are as follows: Dongle: [-1000.000, 5000.000] EMMA: [-1000.000, inverter rated power] SmartLogger: [-1000.000, 5000.000] Distributed SmartLogger: [-1000.000, 50000.000] Inverter: [-1000.000, 5000.000]
>>> limitationMode	Limitation mode. **Total power**: controls the total power at the grid-connection point to limit the power fed to the power grid. **Single-phase power**: controls the power of each phase at the grid-connection point to limit the power fed to the power grid.	String	0: total power 1: single-phase power
> startTime	Time when a task is received, including the time zone information.	String	2020-02-06T00:00:00+08:00

Parameter	Description	Data Type	Remarks
> endTime	Time when a task is completed, including the time zone information. If a task is not completed, null is returned.	String	2020-02-06T00:00 :00+08:00

Example

Request example:

```
{
  "taskId": "3051190140256431"
}
```

Response examples:

Example 1: An error code is returned.

```
{
  "failCode": 20620,
  "success": false,
  "message": "The taskId is not exist.",
  "data": null
}
```

Example 2: An exception is returned.

```
{
  "exceptionId": "framwork.remote.Paramerror",
  "exceptionType": "ROA_EXFRAME_EXCEPTION",
  "descArgs": null,
  "reasonArgs": [
    "taskId"
  ],
  "detailArgs": [
    "taskId may not be null"
  ],
  "adviceArgs": null
}
```

Example 3: The returned task status is **RUNNING**.

```
{
  "failCode": 0,
  "success": true,
  "message": "The active power control task status queried successfully.",
  "data": {
    "dispatchResult": [
      {
        "plantCode": "NE=234567891",
        "controlMode": "6",
        "status": "RUNNING",
        "message": null,
        "controlInfo": {
          "maxGridFeedInPower": 53,
          "limitationMode": "1"
        }
      }
    ],
    "startTime": "2024-03-07T09:36:13+08:00",
    "endTime": null
```

```
   }
}
```

Example 4: The returned task status is **FAIL**.

```
{
   "failCode": 0,
   "success": true,
   "message": "The active power control task status queried successfully.",
   "data": {
      "dispatchResult": [
         {
            "plantCode": "NE=234567891",
            "controlMode": "6",
            "status": "FAIL",
            "message": "FAILURE",
            "controlInfo": {
               "maxGridFeedInPower": 53,
               "limitationMode": "1"
            }
         }
      ],
      "startTime": "2024-03-07T09:36:13+08:00",
      "endTime": "2024-03-07T09:36:23+08:00"
   }
}
```

Example 5: The returned task status is **SUCCESS**.

```
{
   "failCode": 0,
   "success": true,
   "message": "The active power control task status queried successfully.",
   "data": {
      "dispatchResult": [
         {
            "plantCode": "NE=234567891",
            "controlMode": "6",
            "status": "SUCCESS",
            "message": null,
            "controlInfo": {
               "maxGridFeedInPower": 53,
               "limitationMode": "1"
            }
         }
      ],
      "startTime": "2024-03-07T09:38:58+08:00",
      "endTime": "2024-03-07T09:38:58+08:00"
   }
}
```

6 References

6.1 Error Code List

No.	Error Code	Description
1	20001	The third-party system ID does not exist.
2	20002	The third-party system is forbidden.
3	20003	The third-party system has expired.
4	20004	The server is abnormal.
5	20005	The device ID cannot be empty.
6	20006	Some devices do not match the device type.
7	20007	The system does not have the desired power plant resources.
8	20008	The system does not have the desired device resources.
9	20009	Queried KPIs are not configured in the system.
10	20010	The plant list cannot be empty.
11	20011	The device list cannot be empty.
12	20012	The query time cannot be empty.
13	20013	The device type is incorrect. The API does not support operations on some devices.
14	20014	A maximum of 100 plants can be queried at a time.
15	20015	A maximum of 100 plants can be queried at a time.
16	20016	A maximum of 100 devices can be queried at a time.
17	20017	A maximum of 100 devices can be queried at a time.

No.	Error Code	Description
18	20018	A maximum of 10 devices can be operated at a time.
19	20019	The switch type is incorrect. 1 and 2 indicate switch-on and switch-off respectively.
20	20020	The upgrade package corresponding to the device version cannot be found.
21	20021	The upgrade file does not exist.
22	20022	The upgrade records of the devices in the system are not found.
23	305	You are not online and need to log in again.
24	401	You do not have the related data API permission. ● For details about how to locate the fault in the OAuth Connect scenario, see **8.1.4 Why Does the Northbound API Return Error Code 401?** ● For details about how to check the API account access scenario, see **8.2.5 Why Does the Northbound API Return Error Code 401?**
25	407	The API access frequency is too high.
26	20023	The query start time cannot be later than the query end time.
27	20024	The language cannot be empty.
28	20025	The language parameter value is incorrect.
29	20026	Only data of the latest 365 days can be queried.
30	20027	The query time period cannot span more than 31 days.
31	20028	The system does not have related user information.
32	20029	Invalid user information.
33	20030	Failed to create the I-V curve diagnosis task.
34	20034	The task does not exist.
35	20035	MPPT devices do not support backfeed current.
36	20036	The backfeed current duration of the MPPT device exceeds the maximum limit.
37	20037	The backfeed current of the MPPT device is out of range. The allowed value is (0, 15].
38	20038	In the input parameters, the authorization code list is empty (null), or the number of authorization codes is out of range. The allowed range is [0, 1000].

No.	Error Code	Description
39	20039	In the input parameters, the DOD value is out of range. The allowed range is [0, 100].
40	20040	The charge/discharge switch parameter value is invalid.
41	20041	The control type cannot be empty for forced charge and discharge.
42	20042	The target SOC for charge/discharge is empty or invalid.
43	20043	The charge/discharge duration is empty or invalid.
44	20044	The unique ID of a charge/discharge task cannot be empty.
45	20045	Unauthorized PV plants exist in the input parameters.
46	20046	Unauthorized PV plants exist in the input parameters.
47	20047	The forced charge/discharge power in the input parameters is invalid.
48	20048	Duplicate charge and discharge task ID.
49	20049	Failed to deliver the charging and discharging task.
50	20050	The charging and discharging task query parameter does not exist.
51	20051	Failed to set the battery DOD.
52	20055	The plant list and device list parameters cannot be empty at the same time.
53	20056	Resources that are not authorized by the owner exist.
54	20116	The inverter control parameter is incorrect. (The number of PV plants is greater than 100 or equal to 0, or the total number of inverters in the PV plants is greater than 200 or equal to 0.)
55	20200	The system is busy. Try again later.
56	20400	- The API account name or password is incorrect. - The API account is locked. - The password has expired. - The number of online sessions reaches the upper limit. - The third-party system ID does not exist.
57	20403	The API account login is restricted.
58	20604	The time parameter is incorrect. The start time cannot be later than or equal to the end time.

No.	Error Code	Description
59	20605	The time parameter is incorrect. The time parameter contains a negative value.
60	20606	If only the start time is entered and the end time is empty, the start time cannot be later than or equal to the current time.
61	20607	The task list is empty.
62	20608	Duplicate plant IDs exist in the input parameters.
63	20609	The plant networking is abnormal.
64	20610	The plant does not support default configuration.
65	20611	The values of input parameters exceed the valid range.
66	20612	The value of an input parameter is empty.
67	20613	The default setting task fails to be sent for all plants.
68	20614	The network communication is abnormal.
69	20615	The same task is being executed in the current plant.
70	20616	The task ID is empty.
71	20617	The query result of the default plant setting task is empty.
72	20618	The number of API calls has reached the maximum number per API account per day.
73	20619	The number of task IDs exceeds 100.
74	20620	The task ID does not exist.
75	20621	The resource is not authorized by the owner.
76	20622	The plant DN is invalid or the account does not have permission on the plant.
77	20623	The plant DN does not belong to the company or subsidiary of the owner.
78	20624	The owner has been bound to the plant.
79	20625	It is not allowed to bind a plant whose sharing request has been accepted by the owner.
80	20626	Failed to bind the plant due to an internal error.
81	21000	The basic information for the plant creation is empty.
82	21001	The plant name is empty or in an incorrect format.
83	21002	The plant type is empty or incorrect.

No.	Error Code	Description
84	21003	The grid connection time must be a positive number.
85	21004	The format of the contact name is incorrect.
86	21005	The format of the contact information is incorrect.
87	21006	The C&I plant and utility plant cannot be EV-charger-only plants.
88	21007	The grid connection time cannot be set for an EV-charger-only plant.
89	21008	The string capacity cannot be set for an EV-charger-only plant.
90	21009	The electricity price cannot be set for an EV-charger-only plant.
91	21010	**>pureChange** can only be set to **0** or **1**.
92	21011	The plant name already exists.
93	22000	No related data of connected devices was found.
94	22001	The device registration code is empty.
95	22002	The device is an unauthorized device.
96	22003	The device registration code is incorrect.
97	22004	You will be locked out for 5 minutes due to five consecutive incorrect registration codes.
98	22005	Devices other than chargers are connected to an EV-charger-only plant.
99	22006	The device SN is empty.
100	22007	The device has been bound to another plant.
101	22008	The parameters of connected devices are empty.
102	22009	iSitePower-M cannot be connected.
103	22010	The plant with the iSitePower-M cannot connect to other devices.
104	23000	The plant-level string capacity and PV-level string capacity cannot be empty at the same time.
105	23001	The format of the plant string capacity is incorrect.
106	23002	The inverter SN does not exist.
107	23003	The number of inverter SNs is incorrect.
108	23005	The inverter PV string capacity is incorrectly set.

No.	Error Code	Description
109	23006	The inverter SN or PV string capacity is empty.
110	23007	The quantity in the inverter PV string capacity setting is incorrect.
111	24000	The electricity price must be a positive number.
112	24001	The electricity price date settings must cover a complete year and without overlapping.
113	24002	The electricity price time segments must cover 24 hours of the day without overlapping.
114	24003	The date range is invalid.
115	24004	The time range is invalid.
116	24005	The company electricity price is empty.
117	24006	**>useCompanyPrice** can only be set to **0** or **1**.
118	25000	Additional information configuration is empty.
119	25001	The area code is empty or in an incorrect format.
120	25002	The plant address is empty.
121	25003	The longitude and latitude are empty or in incorrect format.
122	25004	The safe running time of the plant must be a positive number.
123	25005	The time zone is empty or the format is incorrect.
124	25006	**loadStatus** can only be set to **0** or **1**.
125	25007	The plant introduction can contain a maximum of 128 characters.
126	26000	Failed to create plants.
127	26001	A maximum of 1000 plants can be created in a day.
128	26002	The company authorized by the API account does not exist.
129	26003	Failed to bind the API account to the device.
130	26004	Failed to unbind the API account from the device.
131	30001	The device ESN list cannot be empty.
132	30002	The ESNs queried at a time cannot exceed 50.
133	30003	The account cannot be empty in the input parameter.
134	30004	The value of **pageNo** cannot be empty.

No.	Error Code	Description
135	30005	The value of **pageSize** cannot be empty.
136	30006	The value of **pageSize** is out of range. The allowed range is {10, 20, 30, 50, 100}.
137	30007	The values of **startTime** and **endTime** must be both provided or empty.
138	30008	Failed to call the internal API.
139	30009	The value of **taskName** is empty.
140	30010	The value of **dns** is empty.
141	30011	The value of **cleanStatus** is empty or invalid.
142	30012	The value of **environmentalParameters** is empty or invalid.
143	30013	The value of **modulePlaneIrradiance** or **moduleBackSurfaceTemperature** is empty when **environmentalParameters** is set to 1.
144	30014	The value of **scanPointNum** must be set to 128.
145	30015	The value of **taskId** is empty.
146	30016	The value of **dn** is empty.
147	30017	The value of **dns** is invalid. The number of devices exceeds 100 or devices on which the API account does not have permission exist.
148	30018	The value of **taskName** is invalid (for example, null field).
149	30019	The value of **moduleBackSurfaceTemperature** is out of range. The allowed range is [0.0, 100.0].
150	30020	The value of **modulePlaneIrradiance** is out of range. The allowed range is [600.0, 1500.0].
151	30021	The value of **pageNo** is smaller than 0.
152	30022	The value of **timestamp** is empty.
153	30023	The command type is invalid (for example, null).
154	30024	The power supply duration is invalid.
155	30025	The MPPT list is empty.
156	30026	The value of **mppts** is empty.
157	30027	The number of MPPTs connected to a single inverter exceeds the maximum limit (3), or the total number of MPPTs in a single task exceeds the maximum limit (32).

No.	Error Code	Description
158	30028	The backfeed current input value is invalid.
159	30029	Authentication failed. The entered plant or device parameter is invalid.
160	30030	The input parameter is incorrect.
161	30031	A maximum of 10 devices can be queried at a time.
162	30032	The time parameter is invalid. The query time segment cannot be longer than three days.
163	30033	The task is in progress.
164	30034	The returned list is empty.
165	30035	The task is to be executed.
166	30036	The task has been canceled.
167	30037	The number of reservation tasks exceeds the maximum (10).
168	30040	The PV string is not configured.
169	1	The task is partially successful.
170	2	The task fails.

6.2 Old Policy for API Flow Control

The following table describes the flow control of unrestricted northbound APIs. Call APIs properly based on the flow control description.

API Label	API Name	Flow Control Description
Basic data	Plant list	Maximum number of API calls for each API account: 10/minute
	Device list	Maximum number of API calls for each API account: 10/minute
Monitoring data	Real-time plant data	Maximum number of API calls for each API account: 30/minute
	Real-time device data	Maximum number of API calls for each API account: 10/minute
	Historical device data	Maximum number of API calls for each API account: 1/minute

API Label	API Name	Flow Control Description
Alarm data	Querying active alarms	Maximum number of API calls for each API account: 10/minute
Report data	Hourly plant data	Maximum number of API calls for each API account: 10/minute
	Daily plant data	Maximum number of API calls for each API account: 10/minute
	Monthly plant data	Maximum number of API calls for each API account: 10/minute
	Yearly plant data	Maximum number of API calls for each API account: 10/minute
	Daily device data	Maximum number of API calls for each API account: 10/minute
	Monthly device data	Maximum number of API calls for each API account: 10/minute
	Yearly device data	Maximum number of API calls for each API account: 10/minute
Control API	Delivering battery charge/discharge tasks	Maximum number of API calls for each API account: 10/minute
	Querying battery charge/discharge tasks	Maximum number of API calls for each API account: 10/minute

6.3 Time Zone Code List

Time Zone	Details	Time Zone	Details	Time Zone	Details	Time Zone	Details
54	(UTC-12:00) International Date Line West	102	(UTC-03:00) Araguaína	13	(UTC+03:00) Minsk	36	(UTC+08:00) Singapore

Time Zone	Details	Time Zone	Details	Time Zone	Details	Time Zone	Details
85	(UTC-11:00) Coordinated Universal Time-11	103	(UTC-03:00) Greenland	77	(UTC+03:00) Ankara	37	(UTC+08:00) Perth
55	(UTC-10:00) Hawaii	104	(UTC-03:00) Punta Arenas	17	(UTC+03:00) Baghdad	38	(UTC+08:00) Taipei
86	(UTC-10:00) Aleutian Islands	105	(UTC-03:00) Salvador	19	(UTC+03:00) Kuwait, Riyadh	39	(UTC+08:00) Irkutsk
87	(UTC-09:30) Marquesas Islands	106	(UTC-03:00) Saint Pierre and Miquelon	20	(UTC+03:00) Moscow, Saint Petersburg	78	(UTC+08:00) Kuala Lumpur
56	(UTC-09:00) Alaska	107	(UTC-02:00) Coordinated Universal Time-2	21	(UTC+03:00) Nairobi	134	(UTC+08:00) Ulaanbaatar
88	(UTC-09:00) Coordinated Universal Time-9	84	(UTC-01:00) Azores	79	(UTC+03:00) Riyadh	135	(UTC+08:45) Eucla
57	(UTC-08:00) Tijuana	108	(UTC-01:00) Cape Verde	119	(UTC+03:00) Istanbul	40	(UTC+09:00) Osaka, Sapporo, Tokyo

Time Zone	Details	Time Zone	Details	Time Zone	Details	Time Zone	Details
58	(UTC-08:00) Pacific Time (United States and Canada)	4	(UTC) Coordinated Universal Time	22	(UTC+03:30) Tehran	41	(UTC+09:00) Seoul
89	(UTC-08:00) Baja California	1	(UTC+00:00) Dublin, Edinburgh, Lisbon	18	(UTC+04:00) Tbilisi	42	(UTC+09:00) Yakutsk
90	(UTC-08:00) Coordinated Universal Time-8	3	(UTC+00:00) Monrovia, Reykjavik	23	(UTC+04:00) Muscat	136	(UTC+09:00) Chita
59	(UTC-07:00) Chihuahua, La Paz, Mazatlan	82	(UTC+00:00) London	24	(UTC+04:00) Yerevan	137	(UTC+09:00) Pyongyang
60	(UTC-07:00) Mountain Time (United States and Canada)	109	(UTC+00:00) São Tomé	25	(UTC+04:00) Baku	43	(UTC+09:30) Adelaide
61	(UTC-07:00) Arizona	2	(UTC+01:00) Casablanca	83	(UTC+04:00) Abu Dhabi	44	(UTC+09:30) Darwin

Time Zone	Details	Time Zone	Details	Time Zone	Details	Time Zone	Details
62	(UTC-06:00) Central Standard Time (United States and Canada)	5	(UTC+01:00) Berlin	120	(UTC+04:00) Astrakhan, Ulyanovsk	45	(UTC+10:00) Brisbane
63	(UTC-06:00) Central America	6	(UTC+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague	121	(UTC+04:00) Volgograd	46	(UTC+10:00) Guam, Port Moresby
91	(UTC-06:00) Easter Island	7	(UTC+01:00) Brussels, Copenhagen, Madrid	122	(UTC+04:00) Port Louis	47	(UTC+10:00) Hobart
92	(UTC-06:00) Guadalajara, Mexico City, Monterrey	8	(UTC+01:00) Sarajevo, Skopje, Warsaw, Zagreb	123	(UTC+04:00) Saratov	48	(UTC+10:00) Canberra, Melbourne, Sydney
93	(UTC-06:00) Saskatchewan	74	(UTC+01:00) Paris	124	(UTC+04:00) Izhevsk, Samara	138	(UTC+10:00) Vladivostok

Time Zone	Details	Time Zone	Details	Time Zone	Details	Time Zone	Details
64	(UTC-05:00) Bogotá, Lima, Quito, Rio Branco	80	(UTC+01:00) Amsterdam, Bern, Stockholm, Vienna	26	(UTC+04:30) Kabul	139	(UTC+10:30) Lord Howe Island
65	(UTC-05:00) Eastern Time (United States and Canada)	81	(UTC+01:00) Rome	27	(UTC+05:00) Ashgabat, Tashkent	49	(UTC+11:00) Magadan
94	(UTC-05:00) Havana	110	(UTC+01:00) West Africa Time	28	(UTC+05:00) Islamabad, Karachi	140	(UTC+11:00) Bougainville Island
95	(UTC-05:00) Haiti	9	(UTC+02:00) Amman	125	(UTC+05:00) Kyzylorda	141	(UTC+11:00) Norfolk Island
96	(UTC-05:00) Chetumal	10	(UTC+02:00) Harare	126	(UTC+05:00) Yekaterinburg	142	(UTC+11:00) Chokurdakh
97	(UTC-05:00) Turks and Caicos Islands	11	(UTC+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius	29	(UTC+05:30) Chennai, Kolkata, Mumbai, New Delhi	143	(UTC+11:00) Sakhalin

Time Zone	Details	Time Zone	Details	Time Zone	Details	Time Zone	Details
98	(UTC-05:00) Indiana (Eastern Time)	12	(UTC+02:00) Cairo	127	(UTC+05:30) Sri Jayawardenepura Kotte	144	(UTC+11:00) Solomon Islands, New Caledonia
66	(UTC-04:00) Caracas	14	(UTC+02:00) Windhoek	30	(UTC+05:45) Kathmandu	50	(UTC+12:00) Auckland, Wellington
67	(UTC-04:00) Atlantic Time (Canada)	15	(UTC+02:00) Athens, Bucharest	32	(UTC+06:00) Dhaka	51	(UTC+12:00) Anadyr, Petropavlovsk-Kamchatsky
68	(UTC-04:00) Guyana	16	(UTC+02:00) Jerusalem	128	(UTC+06:00) Astana	52	(UTC+12:00) Fiji
69	(UTC-04:00) Santiago	75	(UTC+02:00) Cape Town	129	(UTC+06:00) Omsk	145	(UTC+12:00) Coordinated Universal Time +12
99	(UTC-04:00) Cuiabá	111	(UTC+02:00) Beirut	33	(UTC+06:30) Yangon	146	(UTC+12:45) Chatham Islands
100	(UTC-04:00) Georgetown, La Paz, Manaus, San Juan	112	(UTC+02:00) Damascus	31	(UTC+07:00) Novosibirsk	53	(UTC+13:00) Tongatapu

Time Zone	Details	Time Zone	Details	Time Zone	Details	Time Zone	Details
101	(UTC-04:00) Asunción	113	(UTC+02:00) Tripoli	34	(UTC+07:00) Bangkok, Hanoi, Jakarta	147	(UTC+13:00) Nukualofa
70	(UTC-03:30) Newfoundland	114	(UTC+02:00) Harare, Pretoria	130	(UTC+07:00) Barnaul, Gorno-Altaysk	148	(UTC+13:00) Samoa
71	(UTC-03:00) Brasilia	115	(UTC+02:00) Chisinau	131	(UTC+07:00) Khovd	149	(UTC+13:00) Coordinated Universal Time +13
72	(UTC-03:00) Cayenne, Fortaleza	116	(UTC+02:00) Kaliningrad	132	(UTC+07:00) Krasnoyarsk	150	(UTC+14:00) Kiritimati Island
73	(UTC-03:00) Montevideo	117	(UTC+02:00) Gasha, Hebron	133	(UTC+07:00) Tomsk		
76	(UTC-03:00) Buenos Aires	118	(UTC+02:00) Khartoum	35	(UTC+08:00) Beijing		

6.4 Exception Code List

No.	Status Code	Description	Example
1	400	Parameter verification failed.	
2	500	System error.	

7 Best Practices

7.1 Obtaining a Token

A token is required for the API account created by the administrator to access northbound data. This section describes how to obtain a valid token.

The method and precautions for calling a northbound login API in HTTPS mode are as follows:

1. Method: POST

2. URL: **https://xxx.fusionsolar.huawei.com/thirdData/login** (Enter the actual URL of the NMS.)

3. Request body: API account name and password (Use **systemCode** instead of **password** as the password parameter.)

4. HTTP status code: The response status code 200 indicates that the API is successfully called.

5. Response body: If **failCode** is 0, the API login is successful.

6. Header: After successful login, find the **xsrf-token** field in the response header and its value is the credential information required for accessing northbound APIs.

> **NOTICE**
>
> 1. The validity period of a token is 30 minutes. If you continuously call a token within 30 minutes, the validity period will be automatically extended. Therefore, you must keep the token secure carefully to prevent disclosure.
>
> 2. An account can have only one online session. Repeated login will invalidate the generated token. If a token is invalid, check to eliminate the repeated login issue.
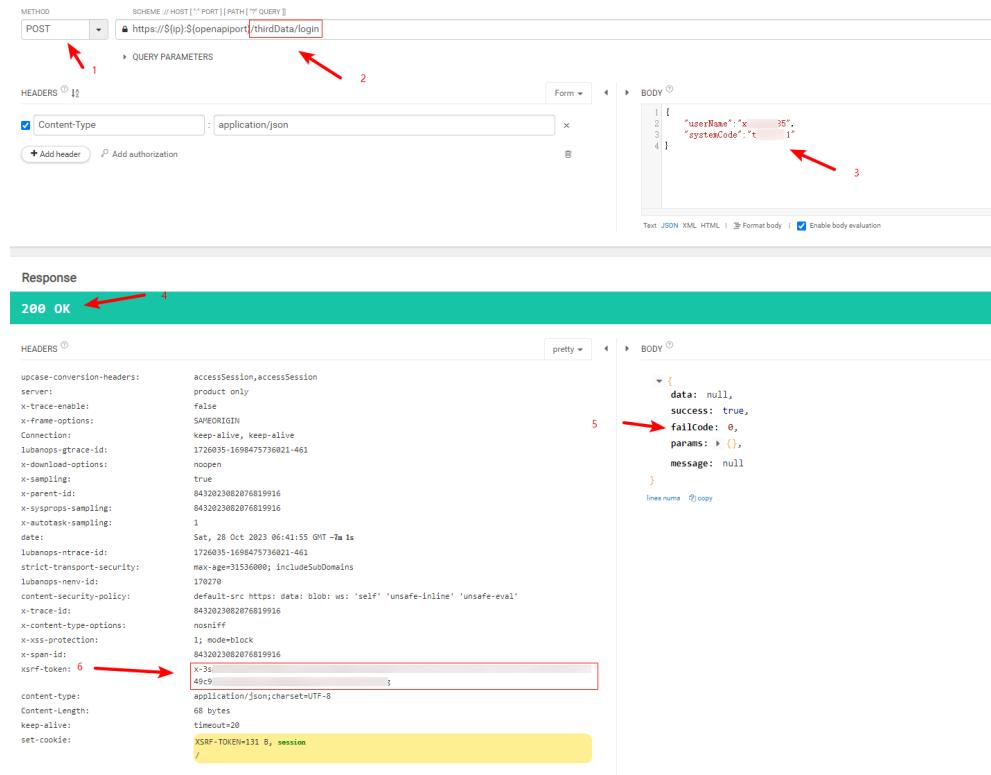
7.2 Querying the Plant List

Before querying the plant information, you need to query the plants on which the current API account has the permission as follows:

1. Method: POST

2. URL: **https://xxx.fusionsolar.huawei.com/thirdData/stations** (Enter the actual URL of the NMS.)

3. Request body: Enter the page No. of the plant list API. The following example queries page No. 1. By default, a maximum of 100 records can be returned for each page.

4. XSRF-TOKEN: Enter the token information obtained through the **/thirdData/login** API.

5. HTTP status code: The response status code 200 indicates that the API is successfully called.

6. Response body: If **failCode** is 0, the query is successful.

7. Response body: The **data** field indicates the information about the plants that you have permission to query.

📖 NOTE

Expand the details in the **data** field to obtain the basic information about a plant, including **plantCode**, the unique identifier for data query. You can save the information for future use.

7.3 Querying the Real-Time Plant Data

You can query the real-time and historical data and reports of plants. The method to obtain real-time plant data is as follows:

1. Method: POST

2. URL: **https://xxx.fusionsolar.huawei.com/thirdData/getStationRealKpi** (Enter the actual URL of the NMS.)

3. Request body: Enter the value in **plantCode** obtained from the plant list API. If multiple values are used, separate them using commas (,). A maximum of 100 values are supported.

4. XSRF-TOKEN: Enter the token information obtained through the **/thirdData/login** API.

5. HTTP status code: The response status code 200 indicates that the API is successfully called.

6. Response body: If **failCode** is 0, the query is successful.

7. Response body: The queried real-time plant data is shown in the **data** field.
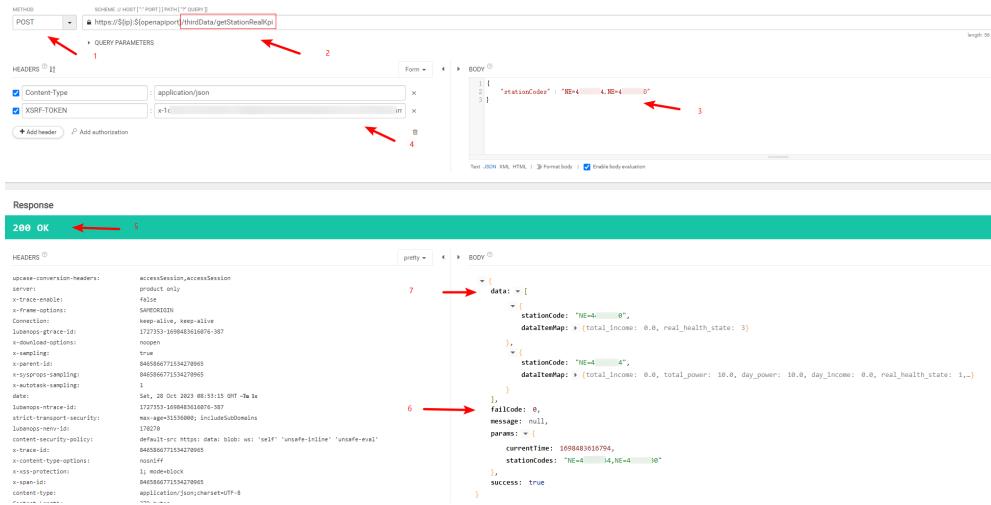
7.4 Using the Access Token to Query the Real-Time Plant Data

The method for querying real-time plant data in OAuth Connect mode is the same as that using an API account. The OAuth Connect mode is used as an example.

1. Method: POST

2. URL: **https://xxx.fusionsolar.huawei.com/thirdData/getStationRealKpi** (Enter the actual URL of the NMS.)

3. Request body: Enter the value in **plantCode** obtained from the plant list API. If multiple values are used, separate them using commas (,). A maximum of 100 values are supported.

4. Authorization: The value of **Authorization** consists of **Bearer** and an access token.

5. HTTP status code: The response status code 200 indicates that the API is successfully called.

6. Response body: If **failCode** is 0, the query is successful.

7. Response body: The queried real-time plant data is shown in the **data** field.
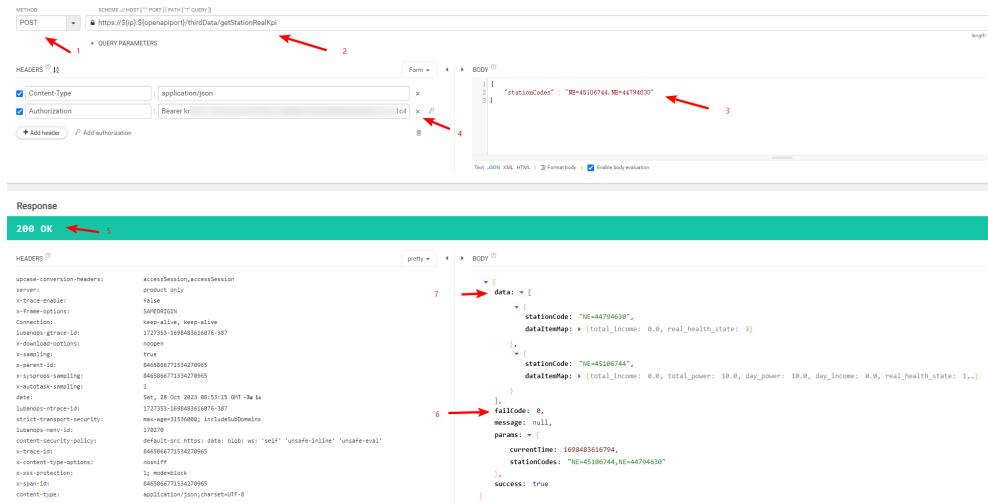
7.5 Scenario-based Practices of Battery Scheduling

The following APIs are provided for setting battery charge and discharge:

- API 1: **5.2.1 API for Delivering Battery Charge and Discharge Tasks**, which is used to set forcible battery charge and discharge, including the charge and discharge power, duration, and target SOC.

- API 2:. **5.2.3 API for Delivering a Task for Setting the Battery Working Mode**, which can be used to set the battery working mode to **TOU** or **maximumSelfConsumption**.

Through these APIs, the following practices can be completed.

Practice 1: Scenarios Where Temporary Charge and Discharge Are Required

API 1 is used to deliver a single forcible charge and discharge task. This method is easy to use.

Step 1 Use API 1 for forcible battery charge and discharge. The charge and discharge power, duration, and target SOC can be set.

----End

1. The API allows users to set the target SOC and charge/discharge duration for residential batteries.

2. API 1 does not change the original working mode of the battery. After the task is complete, the battery continues to work in the original working mode.

Practice 2: Scenarios Where Battery Charge and Discharge Need to Be Controlled Independently Without the Impact of Working Modes

APIs 1 and 2 are used to deliver forcible charge and discharge tasks. API 2 needs to be called only once for setting, and only API 1 needs to be used later.

This mode is applicable to the scenario where the charge and discharge of the battery need to be controlled independently and the working mode does not affect charge and discharge.

Step 1 Use API 2 to set **operationMode** to **TOU**, **redundantPVEnergyPriority** to **fedToGridPreference** (to avoid charging batteries using surplus PV energy), and **allowedAcChargePower** to any valid value. Set the charge and discharge time range to 1 minute. For example, set only one time window, set **startTime** to **00:05** and **endTime** to **00:06**, set **chargeOrDischarge** to **charge**, and set the repeat date to Monday. In this way, the working mode does not affect charge and discharge.

For example, for **NE=123456789**, you can deliver the setting according to the following parameters:

```
{
    "tasks":[
        {
            "plantCode":"NE=123456789",
            "operationMode":"TOU",
            "redundantPVEnergyPriority":"fedToGridPreference",
            "allowedAcChargePower":10,
            "chargingAndDischargingTimeWindow":[
                {
                    "startTime":"00:05",
                    "endTime":"00:06",
                    "chargeOrDischarge":"charge",
                    "repeat":[
                        1
                    ]
                }
            ]
        }
    ]
}
```

Step 2 Use API 1 for forcible battery charge and discharge. The charge and discharge power, duration, and target SOC can be set.

----End

8 FAQs

8.1 FAQs About the OAuth Connect Mode

8.1.1 Handling the Exception Returned by Calling the Authorization Request API

- If the following page is returned after an owner calls the authorization request API described in **Link Example of the Authorization Request API**, check whether the input **client_id**, **redirect_uri**, and **response_type** are correct. For details about how to set the request parameters, see their descriptions in **Parameters of the Authorization Request API** of **3.1.2.2 Initiating Authorization to a Third-party App by an Owner**. If **redirect_uri** contains characters such as **&** and **redirect_uri** is not URL-encoded when the URL of the authorization request API is combined, the following page may be returned when the authorization request API is called. In this case, perform URL-encoding on **redirect_uri**. For details, see the description of **RedirectUrl** in **Link Example of the Authorization Request API** of **3.1.2.2 Initiating Authorization to a Third-party App by an Owner**.

Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Mon Dec 25 10:00:18 GMT+08:00 2023
There was an unexpected error (type=Bad Request, status=400).

- If the exception "?error=invalid_scope&error_description=OAuth%202.0%20Parameter:%20scope" is returned in the callback URL when an owner calls the authorization request API (the scope parameter is required), check whether the input scope parameter is correct and whether scopes are separated by spaces.

8.1.2 Handling the Exception Returned by Calling the Token Obtaining API

- If the following error information is returned when you call the API for obtaining a token, check whether the value of **grant_type** is correct. For details about how to configure the request parameter, see the **grant_type** description in **Request Parameters of the API for Obtaining Tokens** of **3.1.2.3 Obtaining the Access Token of the Open API by a Third-Party App**.

  ```
  HTTP/1.1 400
  {
    "error_description": "unsupported_grant_type",
    "error": "1111"
  }
  ```

- If the authentication fails and the following error information is returned when you call the API for obtaining tokens, check whether **client_id** and **client_secret** are correct. For details about how to configure the request parameters, see the description of **client_id** and **client_secret** in **Request Parameters of the API for Obtaining Tokens** of **3.1.2.3 Obtaining the Access Token of the Open API by a Third-Party App**.

  ```
  HTTP/1.1 401
  {
    "error": "invalid_client"
  }
  ```

- If the following error information is returned when you call the API for obtaining a token, the input parameter verification fails.

  If the value of **grant_type** is **authorization_code**, check whether the execution interval of the operations in **3.1.2.3 Obtaining the Access Token of the Open API by a Third-Party App** exceeds 5 minutes after the authorization code is obtained in **3.1.2.2 Initiating Authorization to a Third-party App by an Owner**. If the interval exceeds 5 minutes, the authorization code has expired. In this case, the owner needs to send an authorization request again to obtain a new authorization code. Operations in **3.1.2.3 Obtaining the Access Token of the Open API by a Third-Party App** must be performed to obtain the access token and refresh token using the new authentication code within 5 minutes. If the interval does not exceed 5 minutes, check whether the values of **code** and **redirect_uri** are correct. For details about how to set the request parameters, see the description of **code** and **redirect_uri** in the access token and refresh token **request parameters** obtained using an authorization code in **3.1.2.3 Obtaining the Access Token of the Open API by a Third-Party App**.

  If **grant_type** is **refresh_token**, check whether **refresh_token** is correct. For details about how to configure request parameters, see **refresh_token** in the **request parameters** for using the refresh token to obtain the access token in **3.1.2.3 Obtaining the Access Token of the Open API by a Third-Party App**.

  ```
  HTTP/1.1 400
  {
    "error_description": "invalid_grant",
    "error": "1110"
  }
  ```

8.1.3 Why Does the Northbound API Return Error Code 305?

If error code 305 is returned when you call a northbound API, as shown in the following figure, the access token is invalid. That is, the owner's authorization information cannot be parsed through the access token. The possible cause is that

the access token has expired (validity period: one hour). In this case, you need to use the refresh token to obtain the access token again. For details, see **3.1.2.3 Obtaining the Access Token of the Open API by a Third-Party App**.
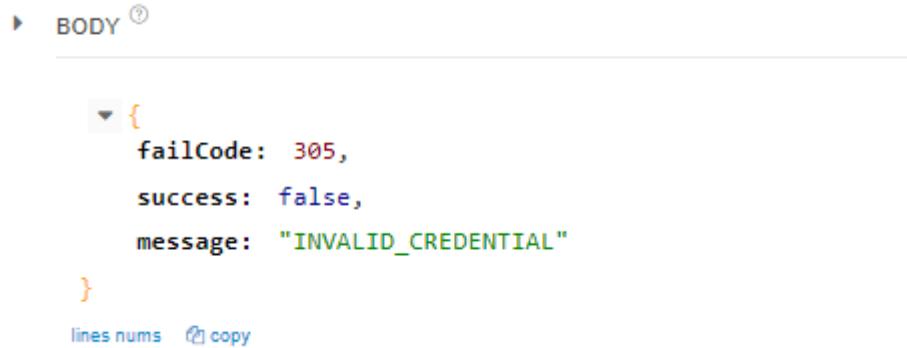
```
▶  BODY ⑦

▼ {
    failCode:  305,
    success:  false,
    message:  "INVALID_CREDENTIAL"
  }
  lines nums    copy
```

8.1.4 Why Does the Northbound API Return Error Code 401?

If error code 401 is returned when you call a northbound API, as shown in the following figure, you do not have sufficient access token permissions. That is, the permission set authorized by the owner to the OAuth 2.0 client does not contain the requested open API. In this case, the owner needs to authorize the permission group to which the open API belongs. For details, see **3.1.2.2 Initiating Authorization to a Third-party App by an Owner**.
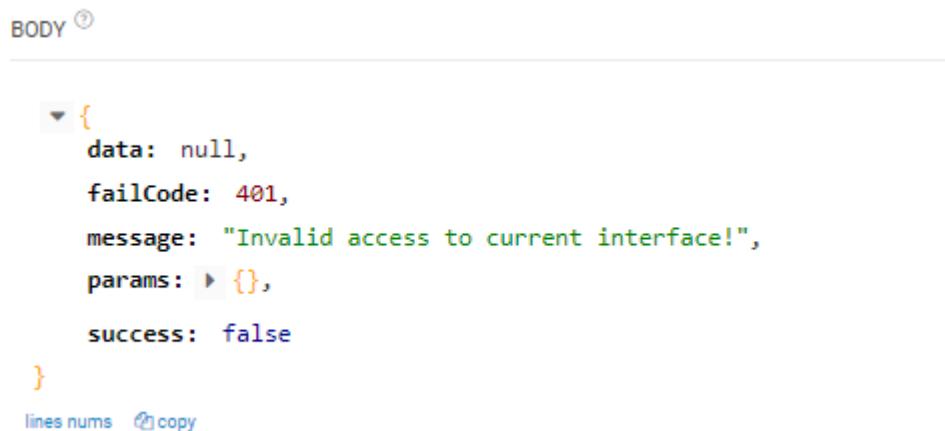
```
BODY ⑦

▼ {
    data:  null,
    failCode:  401,
    message:  "Invalid access to current interface!",
    params:  ▶ {},
    success:  false
  }
  lines nums    copy
```

8.1.5 Why Does the Northbound API Return Error Code 407 or 429?

If error code 407 or 429 is returned when you call a northbound API, as shown in the following figure, the API is called too frequently in a period of time, triggering API flow control. See **4.1 Flow Control Policy in OAuth Connect Mode** for the flow control policies in OAuth Connect mode.
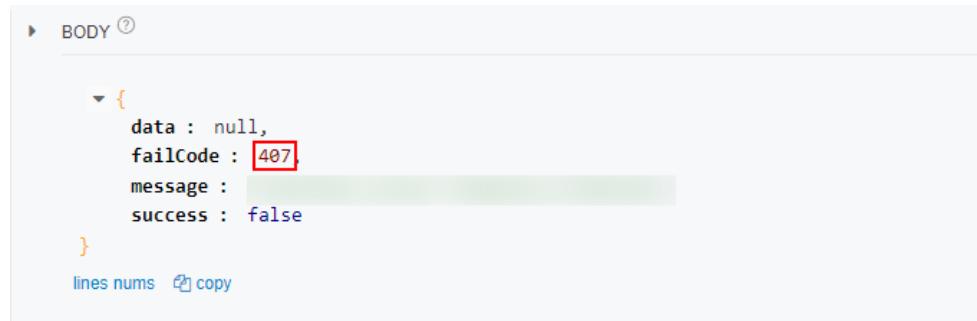
```
▸  BODY ⑦

    ▾  {
          data : null,
          failCode : 407,
          message :
          success : false
    }
    lines nums  ⊕ copy
```
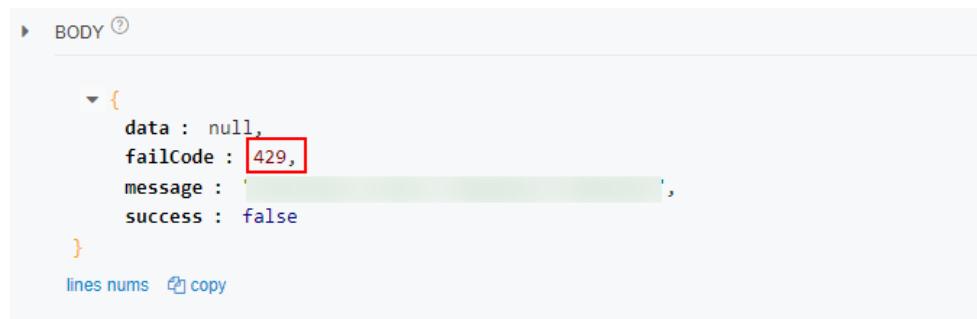
```
▸  BODY ⑦

    ▾  {
          data : null,
          failCode : 429,
          message : '                    ',
          success : false
    }
    lines nums  ⊕ copy
```

8.1.6 How Do I Obtain O&M Support When Open APIs Are Accessed in OAuth Connect Mode?

NOTICE

Connecting to open APIs in OAuth Connect mode is for trial use only in Europe. The following O&M support modes are available only to European customers.

Based on the interconnection status between the third-party system and FuisonSolar SmartPVMS, two phases are involved: commissioning phase and operation phase. The O&M support policies vary depending on the phases.

Interconnection Status	Phase Definition	O&M Support Policy	Description
Commissioning	Commission the interconnection between a third-party system and FusionSolar SmartPVMS.	Dedicated support	Customers can send service requests to the Huawei service contact person or dedicated email address. Huawei rapidly responds to and handles problems in special channels. 1. Service contact person: regional manager/FR 2. Dedicated email address: EU_DP_ServiceOperation<huaweipartners@huawei.com>

Op era tio n	The third-party system has been interconnected with FusionSolar SmartPVMS and is running properly and stably.	Remo te techn ical supp ort	Customers submit service requests through the hotline and obtain solutions from Huawei. In addition, customers can perform self-service O&M through online technical support provided by Huawei.
			1. Hotline service: Customers contact the service contact person (frontline service manager or FR), customer service hotline (0080033666666), or customer service email address (eu_inverter_support@huawei.com) and Huawei will respond to the customer service request.
			2. Remote troubleshooting: includes technical consulting services and problem handling services. Huawei provides a solution to the customer based on the service request submitted by the customer.
			3. Online technical support: Customers can access the Digital Power Smart Chatbot in either of the following methods to obtain product documents and technical support:
			(1) Log in to https://solar.huawei.com, click ≡ in the upper right corner, and choose **Support** > **Online Support**.
			(2) Log in to the FusionSolar app and choose **Services** > **Customer Service Chabot**.

8.2 FAQs About the API Account Mode

8.2.1 Why Do I Fail to Create an API Account?

An administrator can create a certain number of API accounts to access open APIs. The app permissions belong to the administrator's company. An app can be granted multiple plant resource permissions.

If you fail to create an API account, check whether the number of created API accounts exceeds the upper limit (five by default).

8.2.2 What Is the New or Old Flow Control Policy for Northbound APIs?

As the number of API accounts increases, a single API account manages an increasing number of plants and devices. To better meet user requirements, the system updates the old flow control policy for northbound APIs and launched a new policy. By default, new API accounts adopt the new policy. The differences between the new and old flow control policies are as follows:

Old policy: The static flow control policy is adopted, which means that each API account is allowed to send the same number of calling requests for one API. For details, see **6.2 Old Policy for API Flow Control**.

New policy: The dynamic flow control policy is adopted, which means that the new policy is based on the number of resources owned by API accounts. An API account with more plants and devices is allowed to send more API calling requests in a period of time. For details, see **4.2 Flow Control Using the API Account**.

8.2.3 Why Does the Login Fail After I Enter the Correct Username and Password?

If the login still fails after you enter the correct username and password, try the following steps:

Step 1 Check whether parameter names are correct.

Step 2 Check whether the API account has expired.

Step 3 Check whether the API account is disabled.

Step 4 Check whether the API account is locked. If you enter incorrect login passwords for five consecutive times, the account will be locked.

----End

8.2.4 Why Do I Need to Log In Again When I Use the Token to Call an API?

If you encounter this issue, try the following steps:

1. Check whether your token is not used for more than 30 minutes because the validity period of a token is 30 minutes. If you continuously call a token within 30 minutes, the validity period will be automatically extended.

2. An account can have only one online session. Repeated login will invalidate the generated token. If a token is invalid, check to eliminate the repeated login issue.

8.2.5 Why Does the Northbound API Return Error Code 401?

This error code is returned when you call the northbound API, as shown in the following figure.
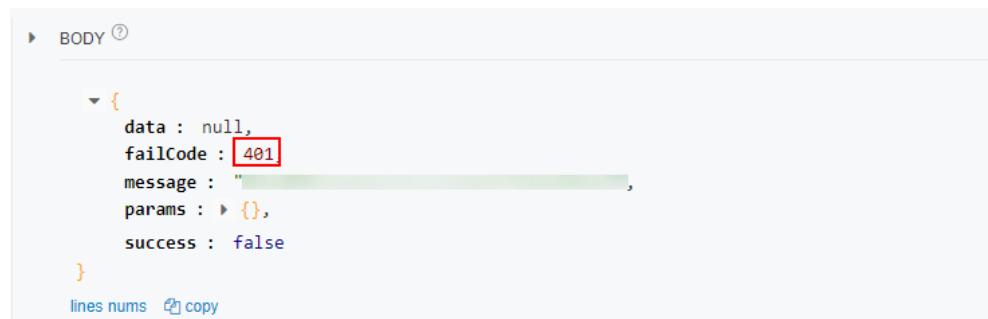
In this case, you need to check whether the API account has the permission to access the API. You can try the following steps to check the permissions for basic and control APIs:

Step 1 Log in to the NMS as an administrator and choose **System** > **Company Management** > **Northbound Management**.

Step 2 Locate the API account and click **Modify** in the **Operation** column. On the displayed page, check whether the corresponding APIs option is enabled.

----End

If the API you access is within the basic and control API ranges, you have not granted the permission on the restricted APIs.

8.2.6 Why Does the Northbound API Return Error Code 407 or 429?

If error code 407 or 429 is returned when you call a northbound API, as shown in the following figure, the API is called too frequently in a period of time, triggering API flow control.

There are new and old policies for API accounts. For details about the new policy, see **4.2 Flow Control Using the API Account**. For details about the old policy, see **6.2 Old Policy for API Flow Control**.

8.3 Why Is No Data or Only Part of Data Found When I Call a Northbound API for Data Query?

If no data or part of data about the plant or device is found when you call the API for accessing the northbound plant list to query data, try the following steps:

Step 1 Check whether the target plant or device exists and whether the plant or device is bound to the corresponding OAuth 2.0 client of the API account or third-party app by comparing the parameters returned in **5.1.1.1 Plant List API** or **5.1.1.2 Device List API**.

Step 2 Find the corresponding northbound API in **5 API Reference**, check whether the API supports the target device type, and check whether the device type is correct.

Step 3 Check whether the target device is disconnected.

Step 4 If **Configure Owner Authorization** is enabled for the company to which the plant or device belongs and the company has been bound to the OAuth 2.0 client of the API account or third-party app, as shown in the following figure, choose **Plants** > **Operation**. In the **Set Permissions** dialogue box, check whether **API Access** is enabled.
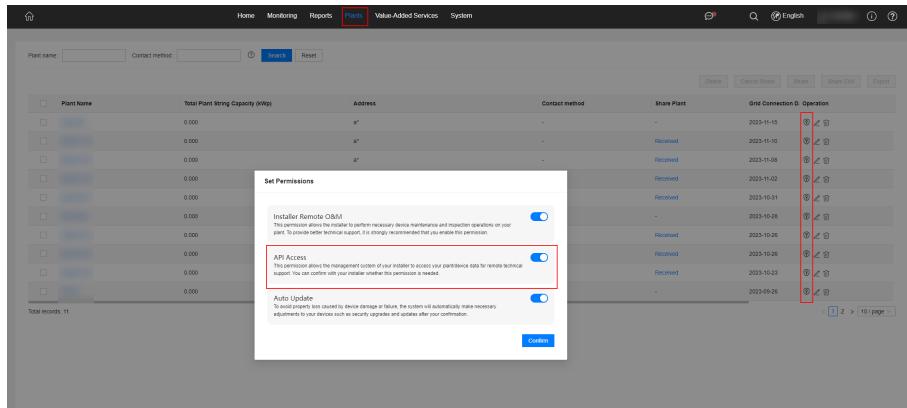
Owners can log in to the FusionSolar app, choose **Me** > **Plant management** on the toolbar at the bottom, select a plant, choose **Set Permissions**, and check whether to enable **API Access**.

Figure 8-1
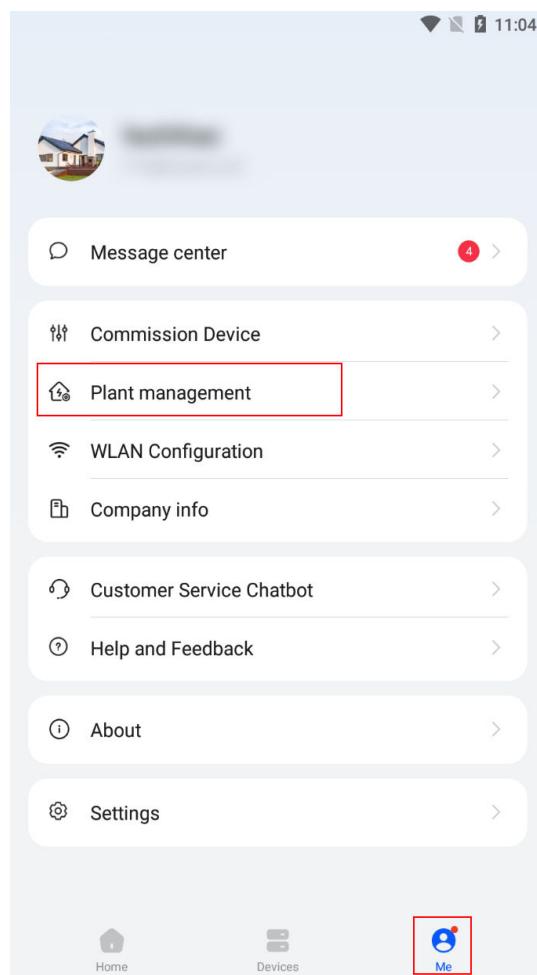
Figure 8-2
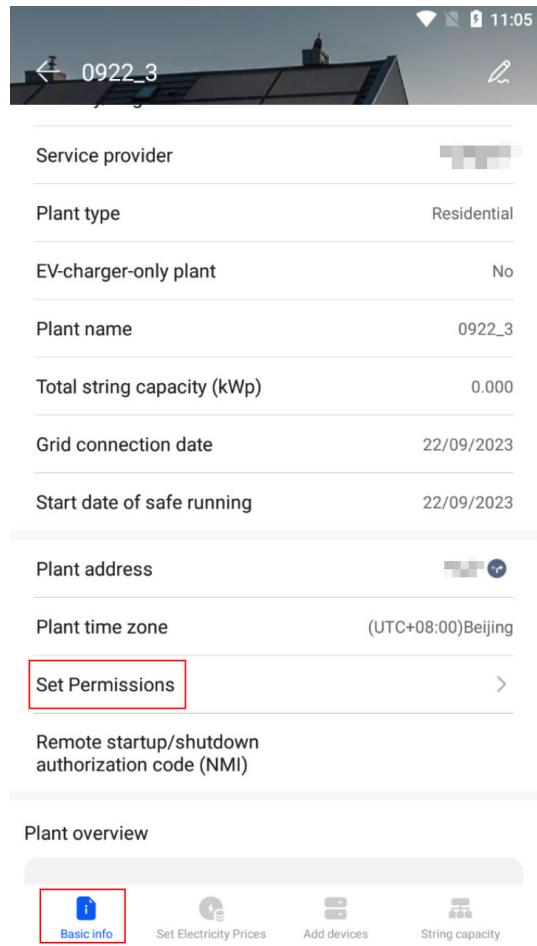
Figure 8-3
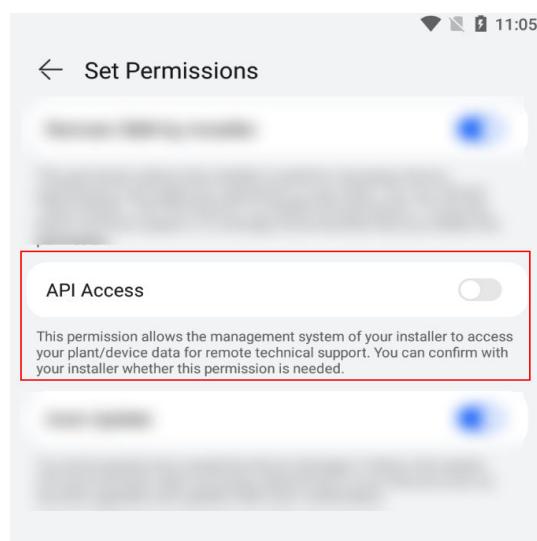
----End

8.4 How Long Is the Data Collection Time of the API for Real-Time Plant Data?

The time for the API to return collected data is usually 5 minutes.

8.5 Why Do the Hourly/Daily/Monthly/Yearly Data APIs of the Plant or the Daily/Monthly/Yearly Data APIs of the Device Sometimes Return Small Data Value?

When you call the hourly/daily/monthly/yearly data APIs of the plant or the daily/monthly/yearly data APIs of the device to query data, the data value returned by the APIs is small.

The issue occurs because data queries through these APIs are based on reports, but there is a delay in summarizing data from reports.

8.6 Why Do I Fail to Query Historical Alarms by Calling the Alarm Query API?

Currently, this API can be used to query only the current (active) alarm information of the device.

8.7 Why Does the /thirdData/getStationList API Return Error Code 401 or 402?

The reason is that APIs in Plant List API have been pre-offline. We provide the new APIs for replacement (see **5.1.1.1 Plant List API**). You are advised to replace APIs in Plant List API with APIs in **5.1.1.1 Plant List API**.

8.8 Why Is the Plant ID Returned by the Plant List Inconsistent with That Displayed on the SmartPVMS?

During the evolution of the management system, the plant IDs change. To ensure user experience of existing open API users, the management system is compatible with the plant IDs of these open API users.

8.9 Why Do the Plant IDs Returned for the Same Plant Vary Depending on Open API Users?

During the evolution of the management system, the plant IDs change. To ensure user experience of existing open API users, the management system is compatible with the plant IDs of these open API users.

If you have multiple open API accounts applied for in different periods, their plant IDs may be different.

8.10 How Do I Compare and Obtain Plants That Are Not Returned by the Plant List API?

If you find that the number of returned plants decreases when you call the API for accessing the northbound plant list, perform the following steps to obtain the plant information that is not returned:

Step 1 Log in to the management system as a company administrator and choose **System** > **Company Management** > **Northbound Management** to view the company bound to the API account.

Figure 8-4 If the page shown in the following figure is displayed, view the company bound to the API account in the plant list API.
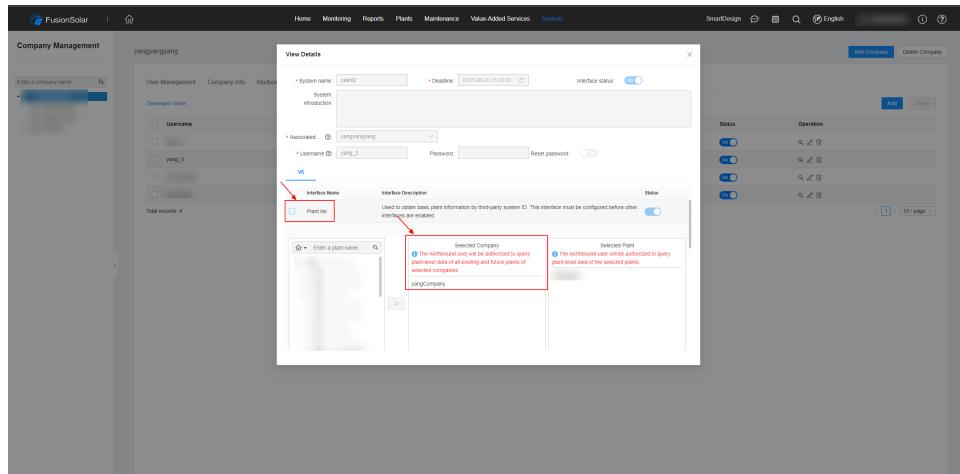
Figure 8-5 If the page shown in the following figure is displayed, view the company bound to the API account in the selected company area.
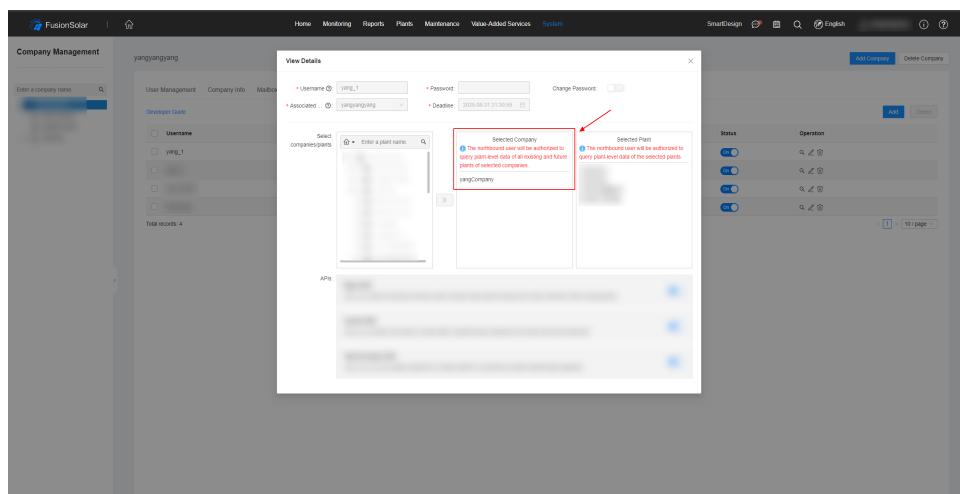
Step 2 Choose **Plants** > **Plant Management**, select the company bound to the API account, and click **Export** to obtain all the plants of the company. If there are a

large number of plants, you are advised to adjust the records displayed on each page.

Figure 8-6

Step 3 Compare the plants exported from the page with those returned by the plant list API to obtain the plants that are not returned.

----End