SUNGROW
iSolarCloud Open API

Contents

目录

1 Usage Specifications of API 7
1.1 Access Steps 7
1.2 Error Code Definition 7
1.3 Constraint Description 7
2 API Definition 10
2.1 User login 10
2.1.1 Service Description 10
2.1.2 Service Address 10
2.1.3 Input Parameter Description 10
2.1.4 Output Parameter Description 10
2.1.5 Sample 11
2.2 Service Access Status 12
2.2.1 Service Description 12
2.2.2 Service Address 12
2.2.3 Input Parameter Description 13
2.2.4 Output Parameter Description 13
2.2.5 Sample 13
2.3 Query Plant List 19
2.3.1 Service Description 19

非禁泄制

2.3.2 Service Address 19
2.3.3 Input Parameter Description 19
2.3.4 Output Parameter Description 20
2.3.5 Sample 23
2.4 Query Plant Detail 26
2.4.1 Service Description 26
2.4.2 Service Address 26
2.4.3 Input Parameter Description 26
2.4.4 Output Parameter Description 27
2.4.5 Sample 29
2.5 Query Device List under Plant 31
2.5.1 Service Description 31
2.5.2 Service Address 31
2.5.3 Input Parameter Description 31
2.5.4 Output Parameter Description 32
2.5.5 Sample 33
2.6 Query PV Inverter Real Time Data 35
2.6.1 Service Description 35
2.6.2 Service Address 35
2.6.3 Input Parameter Description 35
2.6.4 Output Parameter Description 36

2.6.5 Sample 38
2.7 Query Device Point Data in Minutes 41
2.7.1 Service Description 41
2.7.2 Service 41
2.7.3 Input Parameter Description 41
2.7.4 Output Parameter Description 42
2.7.5 Sample 42
2.8 Query Device Point Data in Days, Months and Years 44
2.8.1 Service Description 44
2.8.2 Service Address 44
2.8.3 Input Parameter Description 45
2.8.4 Output Parameter Description 46
2.8.5 Sample 47
2.9 Query Fault Alarm Information 49
2.9.1 Service Description 49
2.9.2 Service Address 49
2.9.3 Input Parameter Description 49
2.9.4 Output Parameter Description 51
2.9.5 Sample 52
2.10 Query Open Point Information 54
2.10.1 Service Description 54

2.10.2 Service Address 54
2.10.3 Input Parameter Description 55
2.10.4 Output Parameter Description 55
2.10.5 Sample 56
2.11 Query Device Property Point Value 57
2.11.1 Service Description 57
2.11.2 Service Address 57
2.11.3 Input Parameter Description 58
2.11.4 Output Parameter Description 58
2.11.5 Sample 59
2.12 Query Communication Device Info by Device SN 61
2.12.1 Service Description 61
2.12.2 Service Address 61
2.12.3 Input Parameter Description 61
2.12.4 Output Parameter Description 61
2.12.5 Sample 62
2.13 Query Device List under The User 63
2.13.1 Service Description 63
2.13.2 Service Address 63
2.13.3 Input Parameter Description 63
2.13.4 Output Parameter Description 64

严禁复制

2.13.5 Sample 66
2.14 Query Device Real Time Data 67
2.14.1 Service Description 67
2.14.2 Service Address 67
2.14.3 Input Parameter Description 68
2.14.4 Output Parameter Description 68
2.14.5 Sample 69
Appendix 1: Device Type Dictionary Definition 70
Appendix 2: API Error Code Definition 71
Appendix 3: Unencrypted API call Sample Code 73
Appendix 4: RSA Encryption Sample Code 75
Appendix 5: AES Encryption Sample Code 77
Appendix 6: AES Decryption Sample Code 78
Appendix 7: API Encrypted Call Sample Code 79
Appendix 8: API Call Help 81

严禁复制

1 Usage Specifications of API

APIs contained in this document support unencrypted call and encrypted call; encrypted call methods are presented in apendix 7. Based on the consideration of security, we recomend encrypted call.

Sample code of Encryption and decryption methods is presented in Appendix 7： API Encrypted Call Sample Code

1.1 Access Steps

Users can acquire powerstation information and device information with iSolarCloud Open API. In the first step, users call /openapi/login API to identify authentication with user account and password which are provided by iSolarCloud and acquire authorized token. Then, users can call other APIs witin token's term of validity (24 hours).

Note: Within the token's term of validity, its term of validity will be reset to 24 hours after every call. Therefore, there is no need for the users to acquire new token every time.

1.2 Error Code Definition

More details in Appendix 2： API Error Code Definition

1.3 Constraint Description

（1）User accounts, appkey and related information used in this document call APIs under the rules of 《SUNGROW iSolarCloud Open API-Authorization Instructions》and get corresponding response;

（2）Http requests are sent in POST mode

（3）Appkey with authorization is a must when users call platform service, parameter in request body is named appkey;

(4) Every API call need a token to verify identity, corresponding request body parameter is token, [/openapi/login API is an exception];

(5) parameters in request header:

Name	Type	Length	Description	Required?
Content-Type	String	32	Input: application/json;charset=UTF-8	Yes
sys_code	String	11	System code: third party call use 901	Yes
lang	String	6	Language (default as chinese) : Simplified Chinese: _zh_CN English: _en_US Japanese: _ja_JP Spanish: _es_ES German: _de_DE Brazilian Portuguese: _pt_BR Portuguese: _pt_BR French: _fr_FR Italian: _it_IT Korean: _ko_KR Dutch: _nl_NL Polish: _pl_PL Vietnamese: _vi_VN Traditional Chinese: _zh_TW	No
x-access-key	String	32	Access_key assigned by iSolarCloud	Yes
x-random-secret-key	String	16	It's secret key of this request and its plaintext length is 16 bit. It should be transmitted after RSA encryption. Output	No

			parameters use this plaintext key to process AES decryption. **This parameter is needed when API call is encrypted call.** How to acquire x-random-secret-key	

(6) Parameters in this table are needed in request body of all APIs. These parameters will not be listed in each API's definition.

Name	Type	Length	Description	Required?
appkey	String	32	Authorization code，required (APi assign appkey to client system)。	Yes
token	String	40	Token (returned by API after success login)	Yes
api_key_param	**Map**		**Other public parameter**	**No**
timestamp	String		Greenwich UNIX timestamp (millisecond) How to acquire timestamp	No
nonce	String	32	32 bit random string of numbers and letters	No

(7) Parameters in this table are contained in response body of all APIs. These parameters will not be listed in each API's definition.

Name	Type	Length	Description
req_serial_num	String	32	Request serial number
result_code	String	11	Error code
result_msg	String	100	Message Note：in API 2.7 and API 2.8, “illegal_device_list" is a list of illegal ps_key
result_data	Object		Result data：Supported data structure：String，Map，List，etc ，specific data format is decided by each

			API. Each API's output parameters are contained in result_data.

2 API Definition

2.1 User login

2.1.1 Service Description

Users use user account and password to authenticate and acquire token.

2.1.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/login

2.1.3 Input Parameter Description

Name	Type	Length	Description	Required ?
user_account	String	32	User account	Yes
user_password	String	32	User password	Yes

2.1.4 Output Parameter Description

Name	Type	Length	Description
login_state	String	2	Login status: -1: account doesn't exist 0: wrong password 1: success login 2 : account is locked due to wrong password 5: this account is locked by administrator
token	String	40	**Token returned after success authentication**
user_id	String	11	User id
user_name	String	64	User name

language	String	32	User language
user_account	String	32	User account
mobile_tel	String	20	User
email	String	64	User email address
err_times	String	2	Password error times
remain_times	String	2	Remain attempts
disable_time	String	32	Account disable date
user_master_org_id	String	11	User organization id
user_master_org_name	String	128	User organization name

2.1.5 Sample

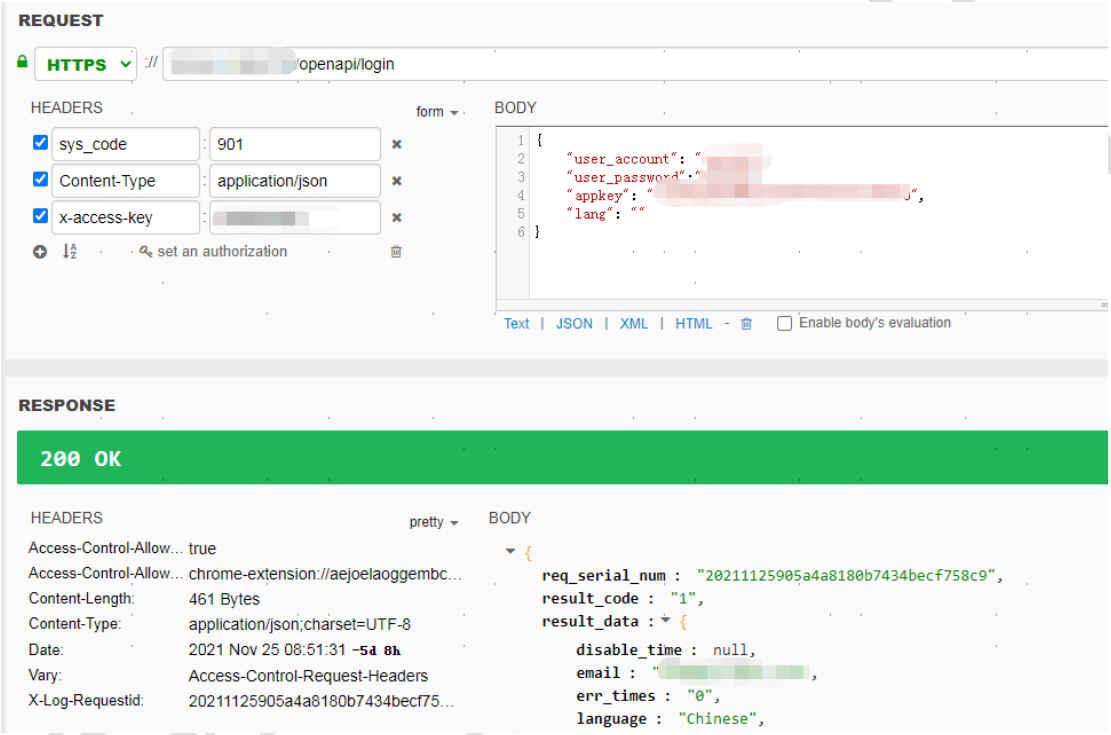
Input：


```
{
    "appkey":"********************************",
    "token":"********",
    "lang":"",
    "user_account":"********",
    "user_password":"********"
}
```


Output：


```
{
    "req_serial_num": "20211122306b4be899a371b8bca02c1a",
    "result_code": "1",
    "result_data": {
        "disable_time": null,
        "email": "***********",
        "err_times": "0",
        "language": "Chinese",
        "login_state": "1",
        "mobile_tel": "***********",
        "token": "***********",
        "user_account": "***********",
```

```
        "user_id": "**********",
        "user_master_org_id": "**********",
        "user_master_org_name": "**********",
        "user_name": "**********"
    },
    "result_msg": "success"
}
```

2.2 Service Access Status

2.2.1 Service Description

With user's appkey, get its service access status including total number of API calls of the day, maximum access times in a day, total number of API calls of the hour, maximum access times in a hour, etc.

2.2.2 Service Address

https://API domain name which is provided by iSolarCloud/openapi/getOpenApiCallInfo

2.2.3 Input Parameter Description

Name	Type	Length	Description	Required?
appkey	String	32	Authorized appkey	Yes

2.2.4 Output Parameter Description

Output Parameter	Type	Length	Description
per_hour_access_times_config	List<Map>		Maximum access times in a hour
hour	String	2	Hour Ex: 1
max_call_times	String	32	Maximum access times
per_day_access_times_config	String	32	Maximum access times in a hour
per_hour_residue_times	List<Map>		Remaining access times of each hour in a day
hour	String	2	Hour Ex: 1
times	String	32	Remaining access times in a hour
today_accessed_times	String	32	Accessed times in a day
curr_hour_accessed_times	String	32	Accessed times in a hour

2.2.5 Sample

Input:


```
{
    "appkey":"********************************",
    "token":"********",
    "lang":""
}
```


Output:


```
{
    "req_serial_num": "20211122306b4be899a371b8bca02c1a",
    "result_code": "1",
```

"result_msg": "success",
"result_data": {
"per_day_access_times_config": "******",
"curr_hour_accessed_times": "*****",
"today_accessed_times": "18",
"per_hour_access_times_config": [{
"hour": "0",
"max_call_times": *****
}, {
"hour": "1",
"max_call_times": *****
}, {
"hour": "2",
"max_call_times": *****
}, {
"hour": "3",
"max_call_times": *****
}, {
"hour": "4",
"max_call_times": *****
}, {
"hour": "5",
"max_call_times": *****
}, {
"hour": "6",
"max_call_times": *****
}, {
"hour": "7",
"max_call_times": *****
}, {
"hour": "8",
"max_call_times": *****
}, {

严禁复制

"hour": "9",
"max_call_times": *****
}, {
"hour": "10",
"max_call_times": *****
}, {
"hour": "11",
"max_call_times": *****
}, {
"hour": "12",
"max_call_times": *****
}, {
"hour": "13",
"max_call_times": *****
}, {
"hour": "14",
"max_call_times": *****
}, {
"hour": "15",
"max_call_times": *****
}, {
"hour": "16",
"max_call_times": *****
}, {
"hour": "17",
"max_call_times": *****
}, {
"hour": "18",
"max_call_times": *****
}, {
"hour": "19",
"max_call_times": *****
}, {

业务受限

"hour": "20",
"max_call_times": *****
}, {
"hour": "21",
"max_call_times": *****
}, {
"hour": "22",
"max_call_times": *****
}, {
"hour": "23",
"max_call_times": *****
}
],
"per_hour_residue_times": [{
"times": "*****",
"hour": "00"
}, {
"times": "*****",
"hour": "01"
}, {
"times": "*****",
"hour": "02"
}, {
"times": "*****",
"hour": "03"
}, {
"times": "*****",
"hour": "04"
}, {
"times": "*****",
"hour": "05"
}, {
"times": "*****",

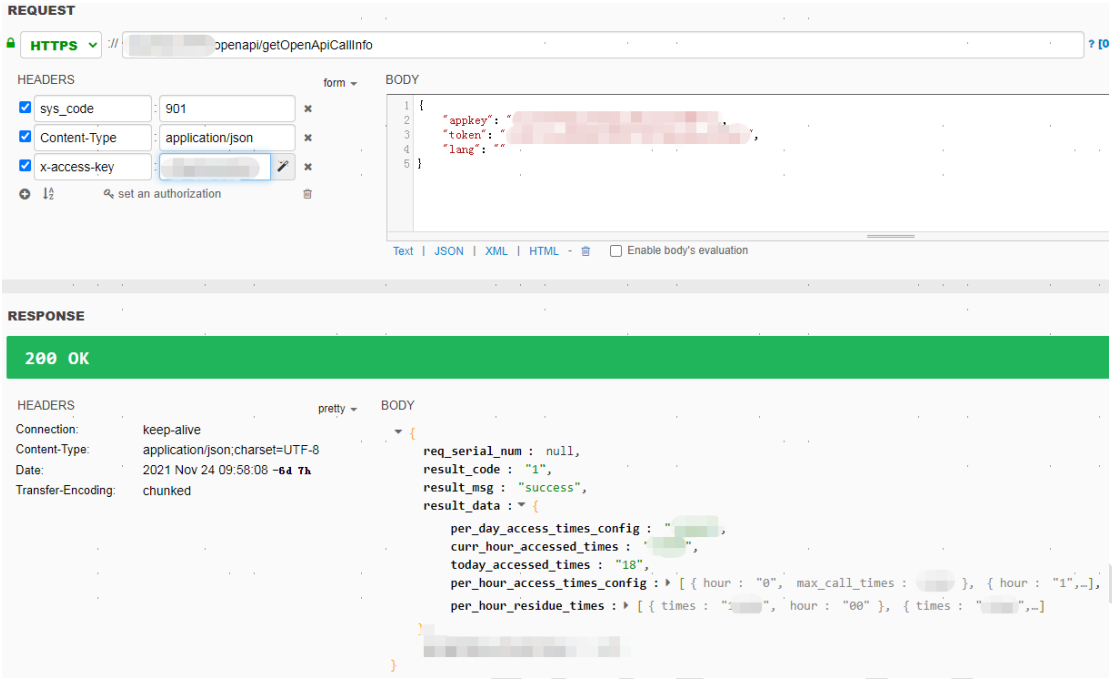
严禁复制

```
        "hour": "06"
    }, {
        "times": "*****",
        "hour": "07"
    }, {
        "times": "*****",
        "hour": "08"
    }, {
        "times": "*****",
        "hour": "09"
    }, {
        "times": "*****",
        "hour": "10"
    }, {
        "times": "*****",
        "hour": "11"
    }, {
        "times": "*****",
        "hour": "12"
    }, {
        "times": "*****",
        "hour": "13"
    }, {
        "times": "*****",
        "hour": "14"
    }, {
        "times": "*****",
        "hour": "15"
    }, {
        "times": "*****",
        "hour": "16"
    }, {
        "times": "*****",
```


业禁复制

```
            "hour": "17"
        }, {
            "times": "*****",
            "hour": "18"
        }, {
            "times": "*****",
            "hour": "19"
        }, {
            "times": "*****",
            "hour": "20"
        }, {
            "times": "*****",
            "hour": "21"
        }, {
            "times": "*****",
            "hour": "22"
        }, {
            "times": "*****",
            "hour": "23"
        }
    ]
  }
}
```


严禁复制



2.3 Query Plant List

2.3.1 Service Description

Query plant list of users or query plant list by user organization ID.

2.3.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getPowerStationList

2.3.3 Input Parameter Description

Name	Type	Length	Description	Required?
ps_name	String	32	Plant name (fuzzy query)	No
ps_type	String	16	Plant type: (use English commas to seperate multiple input types, query all types by default) 1: Utility PV 3: Commercial PV 4: Residential PV	No

			5：Residential Storage 6：Village-level Plant 7：Distributed Storage 8：Poverty Reduction Plant 9：Wind Plant	
valid_flag	String	8	Plant status types： （1：Normal，2：Disable, 3：Commissioning Unfinished）If this parameter is missing, then query normal plant. Note: If multiple plant status types need to be queried, use English commas to seperate.	No
org_id	String	11	If query plant list by organization , pass parameter user_master_org_id which is returned by login API.	No
curPage	String	11	Page number	Yes
size	String	6	Size of each page	Yes
share_type	String	6	Plant share type： 1：shared (share browse permission) 2：shared（share administrative authority） 0：Not shared;（owner's plant）	No

2.3.4 Output Parameter Description

Name	Type	Length	Description
rowCount	Integer	11	Record number
pageList	**List<Map>**		**PageList**
ps_id	Integer	11	Plant ID
ps_name	String	64	Plant name

share_type	String	1	Plant share type: 1: shared (share browse permission) 2 : shared (share administrative authority) 0: Not shared; (owner's plant)
ps_type	Integer	2	Plant type: 1: Utility PV 3: Commercial PV 4: Residential PV 5: Residential Storage 6; Village-level Plant 7: Distributed Storage 8: Poverty Reduction Plant 9: Wind Plant
valid_flag	Integer	2	Plant status 1: Normal, 2: Disable, 3: Commissioning Unfinished
install_date	String	32	Plant install date: format: yyyy-MM-dd HH:mm:ss
ps_current_time_zone	String	10	Plant current time zone
ps_location	String	1000	Plant location
description	String	1000	Plant introduction
longitude	Double	32	Longitude
latitude	Double	32	Latitude
fault_count	Integer	11	Number of fault
alarm_count	Integer	11	Number of alarm
ps_status	Integer	2	Plant status 1: online, 0: offline
ps_fault_status	Integer	2	Plant fault status 1: Fault, 2: Alarm, 4 : Normal
build_status	Integer	2	Plant Construction Status: 0: Not Built, 1: Under Construction, 2:

			Grid-connected , 3 : Proposed Construction, 4: unconnected
connect_type	Integer	11	Grid-connection Type: 1:100% Feed-in 2:Self-Consumption with Surplus Feed-in 3:Self-Consumption , Zero Export 4:Off-grid
total_energy_update_time	String	32	Last update time of plant total yield format: yyyy-MM-dd'T'HH:mm:ssXXX
today_energy_update_time	String	32	Last update time of plant yield today format: yyyy-MM-dd'T'HH:mm:ssXXX
cur_power_update_time	String	32	Last update time of real-time power format: yyyy-MM-dd'T'HH:mm:ssXXX
today_income_update_time	String	32	Last update time of today revenue format: yyyy-MM-dd'T'HH:mm:ssXXX
total_income_update_time	String	32	Last update time of total revenue format: yyyy-MM-dd'T'HH:mm:ssXXX
co2_reduce_update_time	String	32	Last update time of today CO2 reduction format: yyyy-MM-dd'T'HH:mm:ssXXX
co2_reduce_total_update_time	String	32	Last update time of total CO2 reduction format: yyyy-MM-dd'T'HH:mm:ssXXX
total_capcity_update_time	String	32	Last update time of total installed capacity format: yyyy-MM-dd'T'HH:mm:ssXXX
equivalent_hour_update_time	String	32	Last update time of today equivalent hours format: yyyy-MM-dd'T'HH:mm:ssXXX
total_energy	**Map**		**Plant Total Yield<value,unit>**
unit	String	11	Unit
value	String	11	Value
today_energy	**Map**		**Plant Yield today<value,unit>**

unit	String	11	Unit
value	String	11	Value
curr_power	**Map**		**Real-time Power<value,unit>**
unit	String	11	Unit
value	String	11	Value
today_income	**Map**		**Today Revenue<value,unit>**
unit	String	11	Unit
value	String	11	Value
total_income	**Map**		**Total Revenue<value,unit>**
unit	String	11	Unit
value	String	11	Value
co2_reduce	**Map**		**Today CO2 Reduction<value,unit>**
unit	String	11	Unit
value	String	11	Value
co2_reduce_total	**Map**		**Total CO2 Reduction<value,unit>**
unit	String	11	Unit
value	String	11	Value
total_capcity	**Map**		**Total installed Capacity<value,unit>**
unit	String	11	Unit
value	String	11	Value
equivalent_hour	**Map**		**Today Equivalent Hours<value,unit>**
unit	String	11	Unit
value	String	11	Value

2.3.5 Sample

Input：

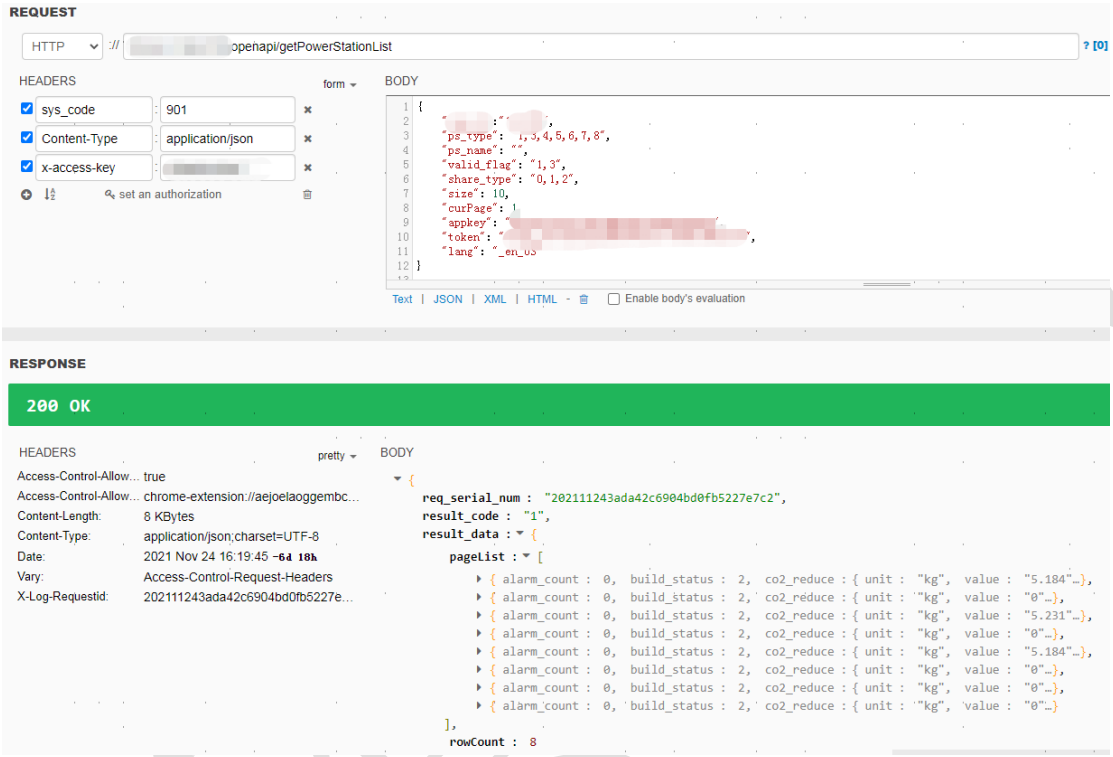
```
{
    "org_id": "*****",
    "ps_type": "1,3,4,5,6,7,8",
    "ps_name": "",
    "valid_flag": "1,3",
    "share_type": "0,1,2",
```

```
    "size": 10,
    "curPage": 1,
    "appkey": "**********",
    "token": "**********",
    "lang": ""
}
Output：
{
    "req_serial_num":"20211124c47443d9af79711b7abd552b",
    "result_code":"1",
    "result_data":{
        "pageList":[
            {
                "alarm_count":0,
                "build_status":2,
                "co2_reduce":{
                    "unit":"kilogram",
                    "value":"4.885"
                },
                "co2_reduce_total":{
                    "unit":"kilogram",
                    "value":"12.811"
                },
                "connect_type":2,
                "curr_power":{
                    "unit":"kW",
                    "value":"2.868"
                },
                "description":null,
                "equivalent_hour":{
                    "unit":"hour",
                    "value":"0.48"
                },
                "fault_count":0,
```

```
        "install_date":"2021-11-20 09:35:13",
        "latitude":**********,
        "longitude":**********,
        "ps_current_time_zone":"GMT+1",
        "ps_fault_status":3,
        "ps_id":******,
        "ps_location":"**************************",
        "ps_name":"***************",
        "ps_status":1,
        "ps_type":4,
        "share_type":"0",
        "today_energy":{
            "unit":"kWh",
            "value":"4.85"
        },
        "today_income":{
            "unit":"yuan",
            "value":"4.9"
        },
        "total_capcity":{
            "unit":"kWp",
            "value":"10"
        },
        "total_energy":{
            "unit":"kWh",
            "value":"66.69"
        },
        "total_income":{
            "unit":"yuan",
            "value":"12.85"
        },
        "valid_flag":1
    }
```


非经授权

```
        ],
        "rowCount":1
    },
    "result_msg":"success"
}
```

2.4 Query Plant Detail

2.4.1 Service Description

Get plant detail by plant ID.

2.4.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getPowerStationDetail

2.4.3 Input Parameter Description

Name	Type	Length	Description	Required?
ps_id	String	11	Plant id	Yes
sn	String	32	If plant id is missing, SN of communication	No

			device under certain plant could be used to query plant information.	

2.4.4 Output Parameter Description

Name	Type	Length	Description
ps_name	String	64	Plant name
ps_id	Integer	11	Plant ps_id
ps_type	Integer	2	Plant type: 1: Utility PV 3: Commercial PV 4: Residential PV 5: Residential Storage 6; Village-level Plant 7: Distributed storage 8: Poverty Reduction Plant 9: Wind Plant
ps_type_name	String	32	Plant type name
ps_key	String	32	Plant ps_key
ps_current_time_zone	String	10	Plant current time zone
ps_location	String	32	Plant location
longitude	Double	32	Longitude
latitude	Double	32	Latitude
design_capacity	Double	11	Installed Power ,unit: Wp
build_status	Integer	2	Plant Construction Status: 0 : Not Built , 1 : Under Construction, 2: Grid-connected, 3: Proposed Construction, 4: unconnected
connect_type	Integer	11	Grid-connection Type: 1:100% Feed-in 2:Self-Consumption with Surplus

			Feed-in 3:Self-Consumption ， Zero Export 4:Off-grid
install_date	String	32	Plant install date: Format：yyyy-MM-dd HH:mm:ss
user_moble_tel	String	32	User mobile phone number
email	String	64	User email
description	String	256	Plant introduction
ps_price	String	11	Current plant tariff per Wh
ps_price_kwh	String	11	Current plant tariff per kWh
param_income_unit_name	String	11	Tariff unit
share_type	String	11	Plant share type: 1：shared（share browse permission） 2：shared（share administrative authority） 0：Not shared;（owner's plant）
share_user_type	Integer	11	Plant share user type 1：Owner share 2：Installer share Other：Not shared plant
fault_count	Integer	11	Number of fault
alarm_count	Integer	11	Number of alarm
ps_status	Integer	2	Plant status 1：online，0：offline
ps_fault_status	Integer	2	Plant Fault Type: 1：Fault，2：Alarm，4:Normal
communication_dev_detail_list	**List**		**SN information of plant's communication device**
sn	String	32	SN
is_enable	Integer	2	SN Status:

			0:Disable,1:Normal

2.4.5 Sample

Input：

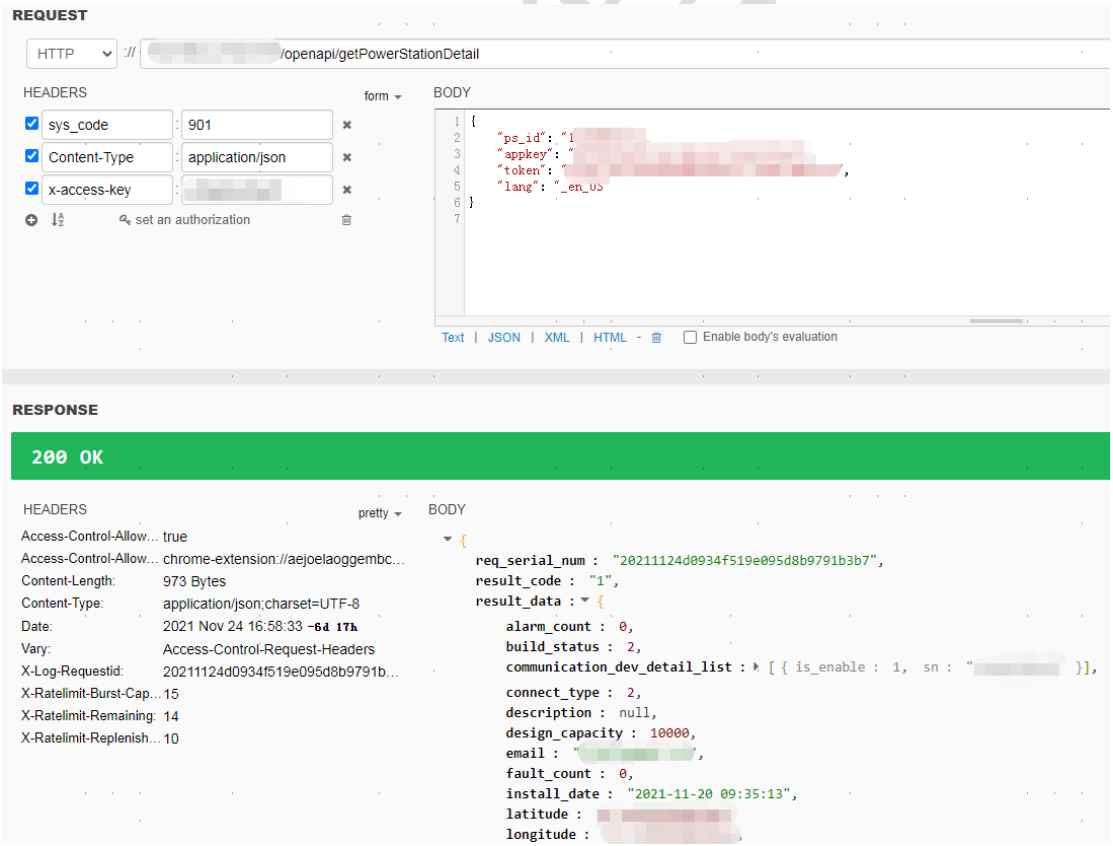

```
{
    "appkey":"********************************",
    "ps_id":"********",
    "token": "******************"
}
```


Output：


```
{
    "req_serial_num": "2021112471ab41a2b57d0090fcc9666b",
    "result_code": "1",
    "result_data": {
        "alarm_count": 0,
        "build_status": 2,
        "communication_dev_detail_list": [{
            "is_enable": 1,
            "sn": "*********"
        }
        ],
        "connect_type": 2,
        "description": null,
        "design_capacity": 10000.0,
        "email": "**********",
        "fault_count": 0,
        "install_date": "2021-11-20 09:35:13",
        "latitude": ***********,
        "longitude": ************,
        "param_income_unit_name": "CNY",
        "ps_current_time_zone": "GMT+1",
        "ps_fault_status": 3,
        "ps_id": *********,
```


严禁复制

```
            "ps_key": "********_11_0_0",
            "ps_location": "**********",
            "ps_name": "**********",
            "ps_price": null,
            "ps_price_kwh": null,
            "ps_status": 1,
            "ps_type": 4,
            "ps_type_name": "Residential",
            "share_type": "0",
            "share_user_type": null,
            "user_moble_tel": null
        },
        "result_msg": "success"
}
```

2.5 Query Device List under Plant

2.5.1 Service Description

Query device list under plant by plant ID.

2.5.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getDeviceList

2.5.3 Input Parameter Description

Name	Type	length	Description	Required?
ps_id	String	11	Plant ID	Yes
is_virtual_unit	String	1	Query virtual device? 1：query virtual device 0：query physical device this parameter is set to 0 by default Note: plant、Grid-connection Point and Unit belong to virtual device	No
device_type_list	String[]	6	Device type list. If no conditions exist, query all types of device. Ex: 11：plant 1：Iverter 3：Grid-connection Point 17：Unit More details inAppendix 1 ：Device Type Dictionary Definition	No

rel_state	String	1	Device claim status: 0: Not Claimed, 1: Claimed	No
is_get_firmware_version	String	1	Is it necessary to acquire device firmware version info: 1: Need 0: Don't need Default as 0	No
curPage	String	32	Page number	Yes
size	String	32	Page size	Yes

2.5.4 Output Parameter Description

Name	Type	Length	Description
rowCount	Integer	11	Record sum
pageList	**List<Map>**		**Result List**
ps_key	String	32	Device ps_key, it is used to query device data
uuid	Integer	11	Device uuid
type_name	String	32	Device type name
factory_name	String	64	Factory name
device_model_code	String	32	Device model name
device_model_id	Integer	11	Device model ID
device_name	String	64	Device name
device_type	Integer	11	Device type code
device_code	Integer	11	Device address code
chnnl_id	Integer	11	Device channel ID
rel_state	Integer	1	Device claim status: 0: Not Claimed 1: Claimed

device_sn	String	32	Device SN
communication_dev_sn	String	32	SN of communication device corresponding to device
dev_status	String	1	Device online status: 0: offline, 1: online
dev_fault_status	String	1	Current device fault status: 1: Fault, 2: Alarm, 4:Normal
firmware_version_info	**Map**		Device firmware version information
lcd_version	String	100	Inverter ARM software version No.
mdsp_version	String	100	Inverter MDSP software version No.
sdsp_version	String	100	Inverter SDSP software version No.
pvd_version	String	100	String detection plate software version No.
cpld_version	String	100	CPLD firmware version No.
temp_version	String	100	Temperature plate software version No.
battery_version	String	100	Battery management system single board version No.
m_version	String	100	Communication device software version No.
system_version	String	100	Communication device hardware platform version No.

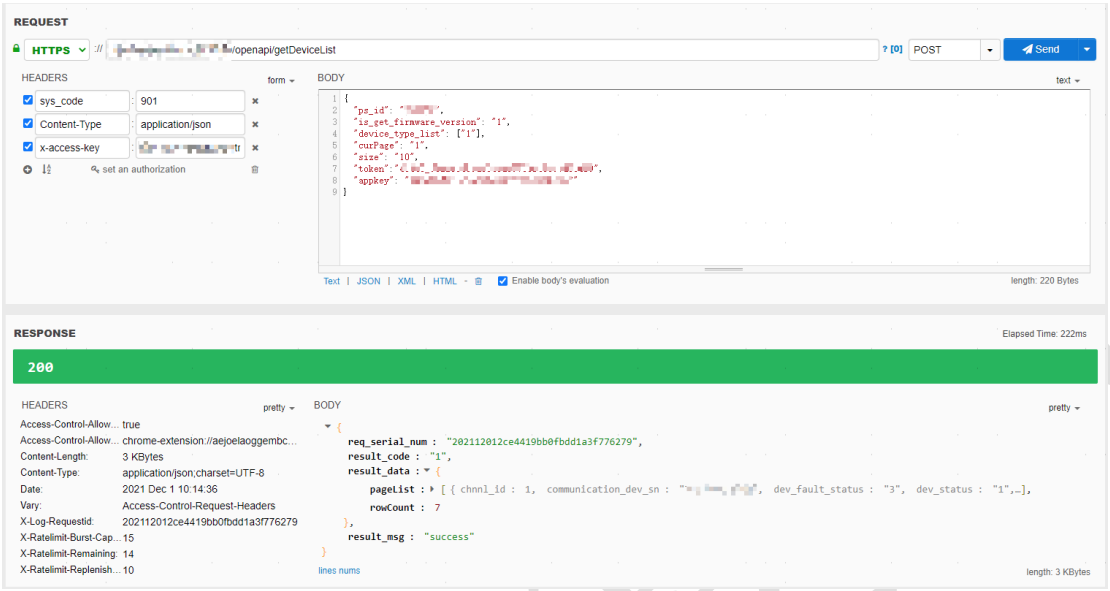
2.5.5 Sample

Input:

```
{
   "ps_id": "******",
   "is_get_firmware_version": "1",
   "device_type_list": ["1"],
   "curPage": "1",
   "size": "10",
```

"token":"**********",
"appkey": "**********"
}
Output：
{
"req_serial_num":"202112012ce4419bb0fbdd1a3f776279",
"result_code":"1",
"result_data":{
"pageList":[
{
"chnnl_id":1,
"communication_dev_sn":"**********",
"dev_fault_status":"3",
"dev_status":"1",
"device_code":20,
"device_model_code":"SG80KTL-M",
"device_model_id":155,
"device_name":"SG80KTL-M(COM4-007)_001_020",
"device_sn":"**********",
"device_type":1,
"factory_name":"********company",
"firmware_version_info":{
"cpld_version":"CPLD_SG80KTL-M_V11_A",
"lcd_version":"LCD_SG80KTL-M_V11_V01_P_M",
"mdsp_version":"MDSP_SG80KTL-M_V11_V1_B",
"pvd_version":"PVD_SG80KTL-M_V11_V1_A"
},
"ps_key":"******_1_20_1",
"rel_state":1,
"type_name":"inverter",
"uuid":1200001
}
],

```
        "rowCount":1
    },
    "result_msg":"success"
}
```

2.6 Query PV Inverter Real Time Data

2.6.1 Service Description

Query PV inverter real time point data by ps_key or SN(data unit is the minimum basic unit)

Note: Unit of return values is always basic unit. For example, the unit of yield value is Wh, the unit of power value is W, the unit of current value is A, the unit of voltage value is V. This API only applies to PV Inverter, you can use getDevicePointMinuteDataList API with corresping time interval to get data of other device.

2.6.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getPVInverterRealTimeData

2.6.3 Input Parameter Description

Name	Type	Length	Description	Required?
ps_key_list	List<String>		List of device ps_key	No
sn_list	List<String>		List of PV inverter SN Note: If ps_key_list is missing, query by	No

			sn_list	

Note：ps_key_list and sn_list can not be empty at the same time；If both of them are not empty, use ps_key_list as query condition.

2.6.4 Output Parameter Description

Name	Type	Length	Description
fail_sn_list	List<String>		List of SN with no eligible ps_key found. Note：this parameter only exists when query by sn_list
device_point_list	**List<Map>**		**List of each device's point value, key of Map is device_point**
device_point	**Map**		**Point list**
device_name	**String**		**Device name**
device_time	String	32	Data update time
ps_id	String	32	Device ps_id
ps_key	String	32	Device ps_key
uuid	String	11	Device uuid
dev_status	Integer	11	Device status: 1：online 0：offline
dev_fault_status	Integer	11	Device fault alarm status: 1：Fault 2：Alarm 4：Normal
device_sn	String	32	Device SN
communication_dev_sn	String	32	SN of communication device corresponding to device
p14	String	32	Inverter Pdc
p24	String	32	Inverter AC_power(Active Power)
p1	String	32	Inverter Daily Yield
p2	String	32	Inverter Total Tield
p15	String	32	AB Online voltage

p16	String	32	BC Online voltage
p17	String	32	CA Online voltage
p18	String	32	Phase A Voltage
p21	String	32	Phase A Current
p19	String	32	Phase B Voltage
p22	String	32	Phase B Current
p20	String	32	Phase C Voltage
p23	String	32	Phase C Current
p27	String	32	Grid Frequency
p4	String	32	Internal air temperature of machine
p5	String	32	Udc-1
p7	String	32	Udc-2
p9	String	32	Udc-3
p45	String	32	Udc-4
p47	String	32	Udc-5
p49	String	32	Udc-6
p51	String	32	Udc-7
p53	String	32	Udc-8
p55	String	32	Udc-9
p57	String	32	Udc-10
p6	String	32	Idc-1
p8	String	32	Idc-2
p10	String	32	Idc-3
p46	String	32	Idc-4
p48	String	32	Idc-5
p50	String	32	Idc-6
p52	String	32	Idc-7

严禁复制

p54	String	32	Idc-8
p56	String	32	Idc-9
p58	String	32	Idc-10

2.6.5 Sample

Input：


```
{
    "ps_key_list": ["******_1_15_1", "******_1_17_1"],
    "token":"**************************",
    "appkey": "**********************************"
}
```


Output：

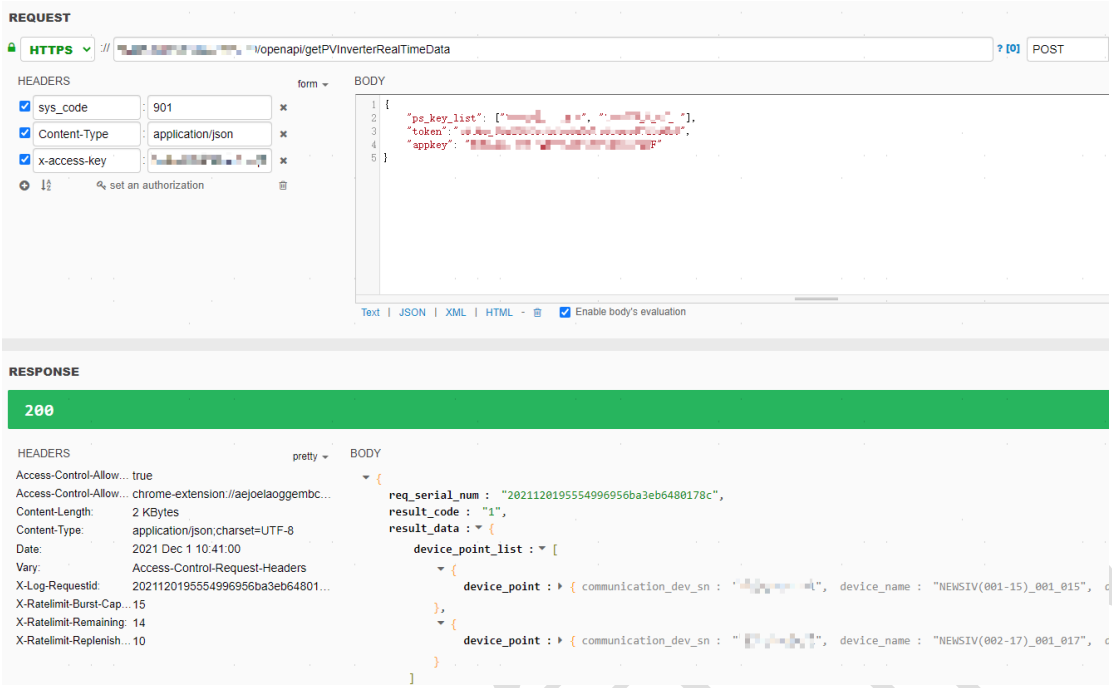

```
{
    "req_serial_num": "202112019555499695 6ba3eb6480178c",
    "result_code": "1",
    "result_data": {
        "device_point_list": [
            {
                "device_point": {
                    "communication_dev_sn": "**********",
                    "device_name": "NEWSIV(001-15)_001_015",
                    "device_sn": null,
                    "device_time": "20210831113500",
                    "p1": "190970.0",
                    "p10": null,
                    "p100": null,
                    "p101": null,
                    "p102": null,
                    "p103": null,
                    "p104": null,
                    "p105": null,
```

"p14": null,
"p15": "493.6",
"p16": "493.9",
"p17": "492.8",
"p18": null,
"p19": null,
"p2": "2096349950.0",
"p20": null,
"p21": "8.4",
"p22": "8.5",
"p23": "8.5",
"p24": null,
"p25": "20100.0",
"p26": "0.999",
"p27": "499.9",
"p304": null,
"p4": "51.1",
"p43": null,
"p45": null,
"p46": null,
"p47": null,
"p48": null,
"p49": null,
"p5": null,
"p50": null,
"p51": null,
"p52": null,
"p53": null,
"p54": null,
"p55": null,
"p56": null,
"p57": null,
"p58": null,

业务复制

```
                "p6": null,
                "p7": null,
                "p70": null,
                "p71": null,
                "p72": null,
                "p73": null,
                "p74": null,
                "p75": null,
                "p76": null,
                "p77": null,
                "p78": null,
                "p79": null,
                "p8": null,
                "p87": "1054400.0",
                "p88": null,
                "p9": null,
                "p96": null,
                "p97": null,
                "p98": null,
                "p99": null,
                "ps_id": ******,
                "ps_key": "**********",
                "uuid": *********
            }
        }
    ]
},
"result_msg": "success"
}
```


业务复制



2.7 Query Device Point Data in Minutes

2.7.1 Service Description

Query device point data in minutes based on multiple device(same type) ps_key, point_id, time interval. The maximum time interval is 3 hours and the maximum number of device is 50. If one of the points of a device doesn't have data, then this point will not be concluded in return result. The number of points in single request should be less than 50.

2.7.2 Service

https:// API domain name which is provided by iSolarCloud/openapi/getDevicePointMinuteDataList

2.7.3 Input Parameter Description

Name	Type	Length	Description	Required?
points	String	32	Point ID，such as p1 and p2. Multiple points are seperated by English commas. Specific open API definition could be acquired by getOpenPointInfo API.	Yes

ps_key_list	List<String>	32	Device ps_key list (same type of device)	Yes
start_time_stamp	String	14	Start time Format：20171018112500	Yes
end_time_stamp	String	14	End time Format：20171018112500	Yes
minute_interval	String	32	Time Interval in minutes Take 15 as an example. it represents point data every 15 minutes. If this parameter is missing or empty, query point data every 5 minutes.	No

2.7.4 Output Parameter Description

Output parameter	Type	Length	Description
result_msg	String		The value returns "success" or an illegal device collection
	Map<String,List<Map>>		Key of map is device ps_key and value of map is device's point data list, data structure is shown below
	List<Map>		Content of Map is shown below
p+point id	String	10	point（Note：Ex：p1,p83022）
time_stamp	String	14	Time

2.7.5 Sample

Input:

```
{
    "points": "p1,p2",
    "appkey": "****************",
    "ps_key_list": ['******',
    '******'],
```

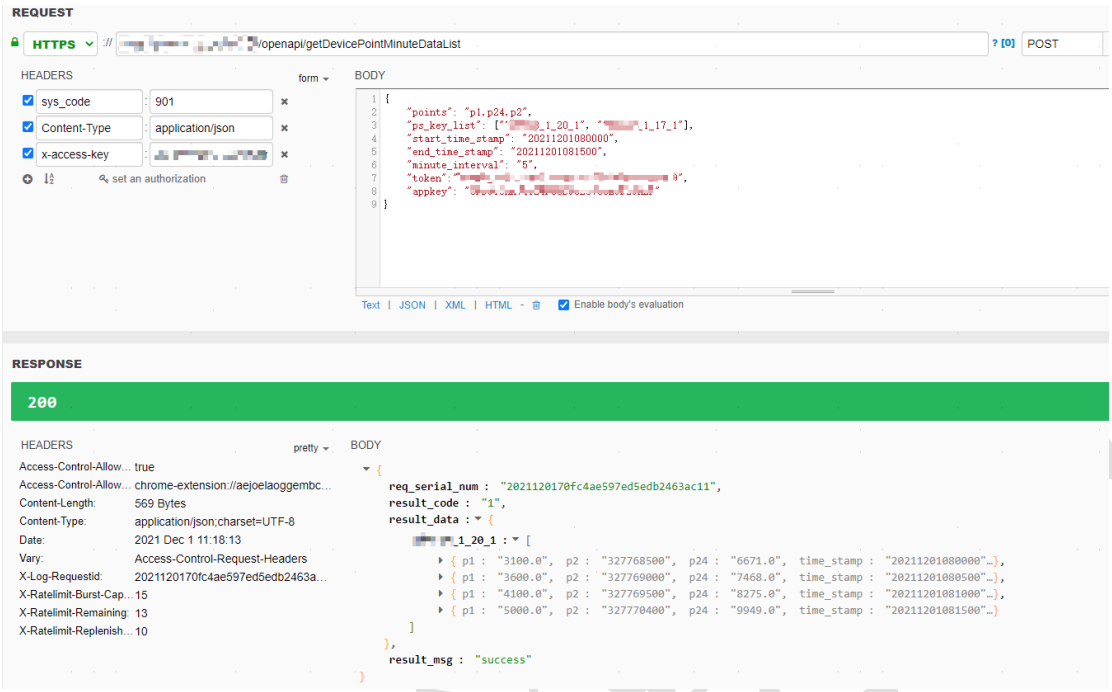
"start_time_stamp": "20200812103000",
"end_time_stamp": "20200812110000",
"minute_interval": "15",
"token": ""**********","
}

Output:

{
"req_serial_num": "1dfb85e714094df18f9a076a6ef42c88",
"result_code": "1",
"result_data": {
"****_1_3_1": [
{
"p1": "0.0",
"p2": "2262800.0",
"time_stamp": "20200812103000"
},
{
"p1": "0.0",
"p2": "2262800.0",
"time_stamp": "20200812104500"
},
{
"p1": "0.0",
"p2": "2293600.0",
"time_stamp": "20200812110000"
}
],
"****_1_5_1": [
{
"p2": "2262800.0",
"time_stamp": "20200812103000"
},
{
"p2": "2262800.0",
"time_stamp": "20200812104500"
},
{
"p2": "2262800.0",
"time_stamp": "20200812110000"
}
]

严禁复制

},
"result_msg": "illegal_device_list:xxxxxx_1_2_1,xxxxx_1_2_1,xxxx_1_2_1"
}



2.8 Query Device Point Data in Days, Months and Years

2.8.1 Service Description

Query device point data in days, months and years based on start time, end time, multiple device(same type) ps_key and data type. API supports multi points query. The maximum number of device is 50.

Note: Unit of return value is always minimum unit. For example, yield's unit is minimum unit wh and power's unit is also minimum unit w. When query device point data in days, API supports 100 days time span . When query device point data in months, API supports 24 months time span. When query device point data in years, API supports 5 years time span.

2.8.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getDevicePointsDayMonthYearDataList

2.8.3 Input Parameter Description

Name	Type	Length	Description	Required?
ps_key_list	String	32	Device ps_key List（Device of same type） Note：If query plant data，ps_key is plant ps_key. The structure of plant ps_key is plant ps_id+ "_11_0_0"，for example xxxxxx_11_0_0	Yes
data_point	String	32	Use English commas to seperate multiple points,. If device type is inverter，then p1 represents inverter yield（return value unit is wh） p2 represents inverter total yield（return value unit is wh） p24 represents inverter total active power（return value unit is w） Specific open API information could be acquired by getOpenPointInfo API.	Yes
start_time	String	32	Start time Description： when query_type is 1, date format is yyyyMMdd, when query_type is 2，date format is yyyyMM, when query_type is 3，date format is yyyy	Yes
end_time	String	32	End time Description： When query_type is 1, date format is yyyyMMdd, When query_type is 2，date format is yyyyMM, When query_type is 3，date format is	Yes

			yyyy	
data_type	String	1	1：Average，2：Peak Value，3：valley 4：Total（Day data doesn't have total value) Use English commas to seperate multiple types. 如果是可以进行合计的数据，则： When query data in days, data_type is 2. Whern query data in months, data_type is 4. When query data in years, data_type is 4.	Yes
order	String	1	Sort order： 1: In the reverse order 0: In order（In chronological order）	No
query_type	String	1	Query type： query day data：1 query month data：2 query year data：3	Yes

2.8.4 Output Parameter Description

Name	Type	Length	Description
result_msg	String		The value returns "success" or an illegal device collection
	Map<String, Map<String,List<Map>>>		Key of Map is device ps_key Value of Map is device's data. Data structure is shown below:
	Map<String,List<Map>>		Device point data: key of Map is point name and value of Map is point data.
Key of Map：points	String	32	Ex：p1

Value of Map: Data list of points	List<Map>		Data list of points
Data of different data_type（Ex：1），meaning of data_type can be found in input parameter description.	String	32	Ex: { "3": "1866392000.0", "2": "1923208000.0", "1": "1896063848.2382", "time_stamp": "201703" }, （Note：Unit of return value is always minimum unit. For example, yield's unit is minimum unit wh and power's unit is also minimum unit w.）
time_stamp	String	32	Corresponding date

2.8.5 Sample

Input:


```
{
    "ps_key_list": ['********',
    '********'],
    "data_point": "p1,p2",
    "start_time": "20200811",
    "end_time": "20200813",
    "data_type": "2",
    "order": "0",
    "query_type": "1",
    "token": "****************",
    "appkey": "************************"
}
```

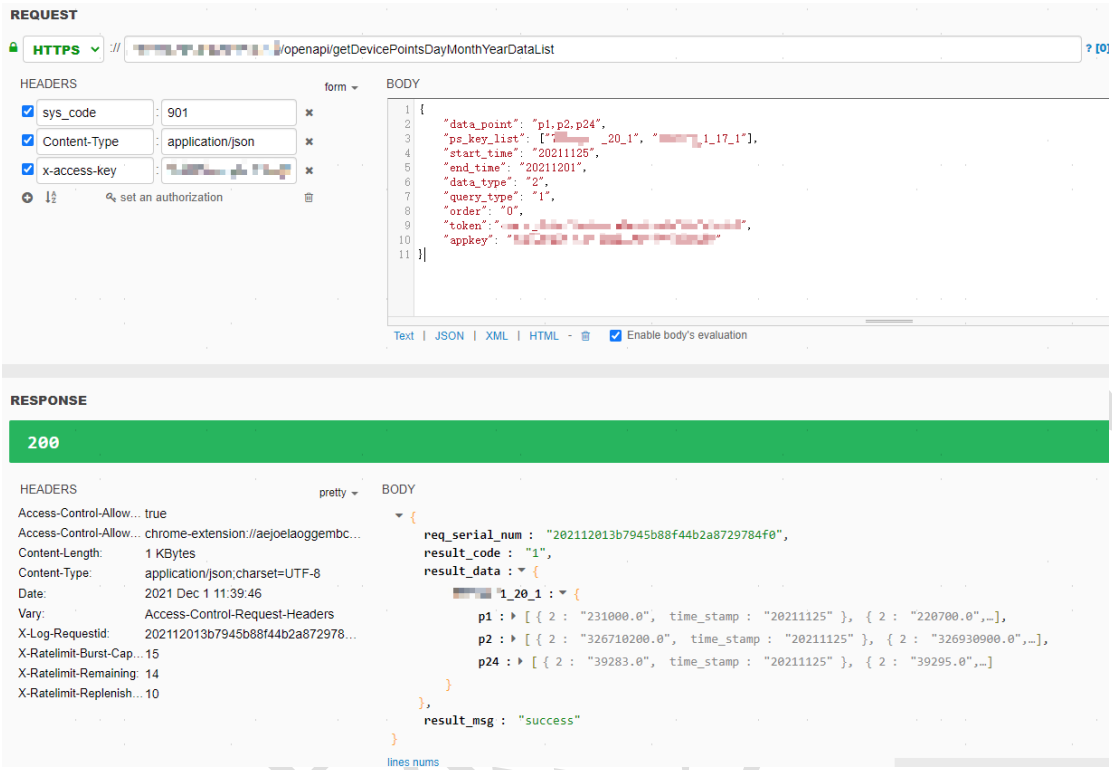

Output:


```
{
    "req_serial_num": "4402b13953344ee5a3dc9c36b6957dcc",
    "result_code": "1",
    "result_data": {
```

```
"****_1_2_1": {
    "p2": [
        {
            "2": "2262800.0",
            "time_stamp": "20200811"
        },
        {
            "2": "2262800.0",
            "time_stamp": "20200812"
        },
        {
            "2": "2262800.0",
            "time_stamp": "20200813"
        }
    ]
},
"****_1_3_1": {
    "p1": [
        {
            "2": "2820200.0",
            "time_stamp": "20200812"
        },
        {
            "2": "0.0",
            "time_stamp": "20200813"
        }
    ],
    "p2": [
        {
            "2": "2262800.0",
            "time_stamp": "20200811"
        },
        {
            "2": "5113800.0",
            "time_stamp": "20200812"
        },
        {
            "2": "5113800.0",
            "time_stamp": "20200813"
        }
    ]
```


业务复制

```
        }
    },
    "result_msg": "success"
}
```



2.9 Query Fault Alarm Information

2.9.1 Service Description

Query device fault alarm information.

2.9.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getFaultAlarmInfo

2.9.3 Input Parameter Description

Name	Type	Length	Description	Required?
fault_code	String	32	Fault unique identifier，if this paramter is passed, query fault alarm information.	No
fault_name	String	32	Fault name（fuzzy query when	No

			this parameter is not empty)	
startTime	String	12	Start time (format: yyyyMMddHHmm) This parameter can be empty Note: When query closed fault list, multi month query is not allowed. If time related parameters (start time and end time) are missing, query current month data.	No
endTime	String	12	End time (format : yyyyMMddHHmm) This parameter can be empty. Note: When query closed fault list, multi month query is not allowed. If time related parameters (start time and end time) are missing, query current month data.	No
process_status	String	11	Fault status: 8: active 9: closed If this parameter is empty, query unclosed fault.	No
fault_type	String	32	Fault type: 1: fault 2: alarm 3: prompt 4: advice (use English commas to seperate when there are multiple fault type input)	No
fault_level	String	11	Fault level: 1: important	No

严禁复制

			2：secondary 3：general 4：slight （use English commas to seperate when there are multiple fault level input）	
ps_id	String	11	Plant id	No
ps_key	String	32	Device ps_key	No
share_type	String	6	Plant share type： 1：shared（share browse permission） 2：shared（share administrative authority） 0：Not shared;（owner's plant） （use English commas to seperate when there are multiple share type inputl）	No
curPage	int	11	Page number	Yes
size	int	11	Page size	Yes

2.9.4 Output Parameter Description

Name	Type	Length	Description
rowCount	Integer	11	Sum
pageList	**List<Map>**		**Fault list**
ps_id	Integer	11	Plant id
ps_key	String	32	Device ps_key
fault_code	String	32	Dault unique id
fault_type_code	Integer	11	Fault type code
fault_type	Integer	11	Fault type：

			1：fault 2：alarm 3：prompt 4：advice
fault_level	Integer	11	Fault level： 1：important 2：secondary 3：general 4：slight
process_status	Integer	11	Fault process status: 1：unconfirmed 2：pending 3：processing 4：resolved 9：closed
fault_name	String	64	Fault name
fault_desc	String	32	Fault description
type_name	String	64	Device type name
ps_name	String	64	Plant name
device_name	String	64	Device name
device_model_code	String	32	Device model
uuid	Integer	11	Device uuid
create_time	String	19	Fault occurrence time Format：yyyy-MM-dd HH:mm:ss
over_time	String	19	Fault restore time Format：yyyy-MM-dd HH:mm:ss

2.9.5 Sample

Input：

{

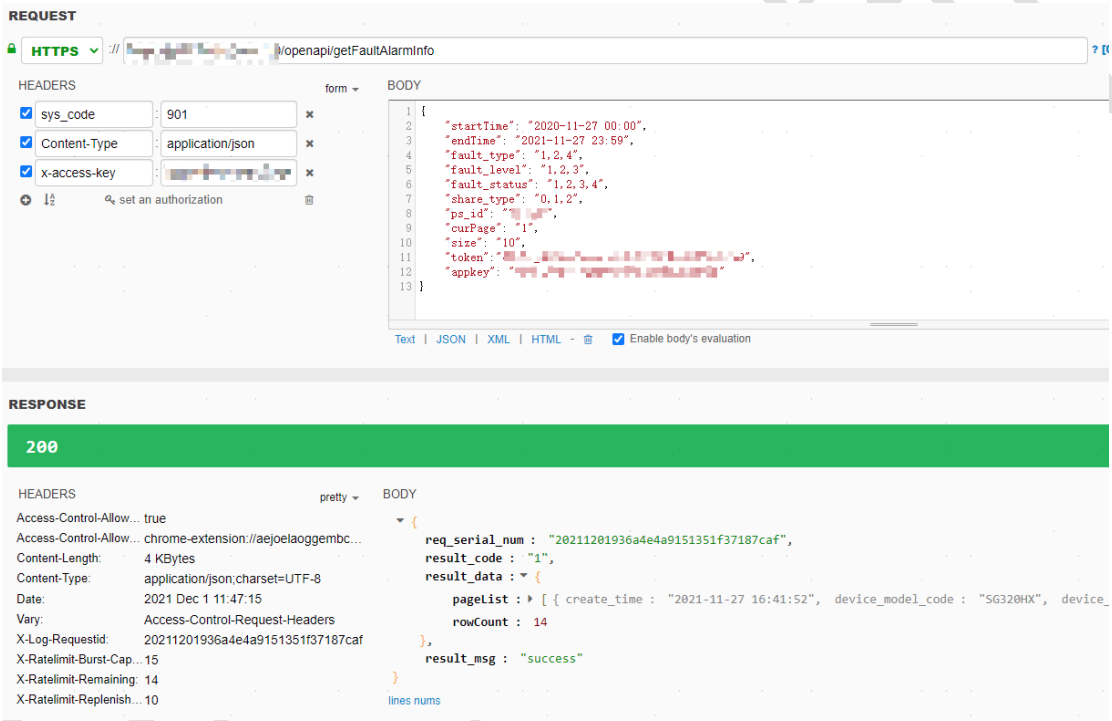
"startTime": "2020-11-27 00:00",
"endTime": "2021-11-27 23:59",
"fault_type": "1,2,4",
"fault_level": "1,2,3",
"share_type": "0,1,2",
"ps_id": "*****",
"curPage": "1",
"size": "10",
"token":"**********",
"appkey": "**********"
}

Output:

{
"req_serial_num": "20211201936a4e4a9151351f37187caf",
"result_code": "1",
"result_data": {
"pageList": [
{
"create_time": "2021-11-27 16:41:52",
"device_model_code": "SG320HX",
"device_name": "SG110CX(COM2-001)_001_0013",
"fault_code": "600**********0",
"fault_desc": "",
"fault_level": 1,
"fault_name": "grid power outage",
"fault_type": 1,
"fault_type_code": 60005,
"over_time": "",
"process_status": 1,
"ps_id": *******,
"ps_key": "*********",

业禁复制

```
                "ps_name": "************",
                "type_name": "inverter",
                "uuid": ********
            }
        ],
        "rowCount": 1
    },
    "result_msg": "success"
}
```

2.10 Query Open Point Information

2.10.1 Service Description

Query open remote signaling point, telemetry point and property point of a type of device based on device type,.

Query open telemetry point of a type of device based on device type and device model ID.

2.10.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getOpenPointInfo

2.10.3 Input Parameter Description

Name	Type	Length	Description	Required?
device_type	String	2	device type	Yes
type	String	1	point type: 1: Remote signaling point 2: Telemetry point 5: Property point	Yes
size	String	11	page size	Yes
curPage	String	11	page number	Yes
device_model_id	String	11	Device model ID : When query telemetry point data, if device_model_id is passed, then use device_model_id to query telemetry point info which is supported by this type of device. If device_model_id is missing, then query all telemetry point info which is supported by this type of device.	No

2.10.4 Output Parameter Description

Name	Type	Length	Description
rowCount	Integer	11	record sum
pageList	List<Map>		point information
point_id	Integer	11	point ID
point_name	String	64	point name
device_type	Integer	11	device type
open_point_remark	String	128	point description information
show_unit	String	11	point display unit (this parameter

			exists when queried point type is telemetry point or property point)
storage_unit	String	11	point storage unit (this parameter exists when queried point type is telemetry point or property point)

2.10.5 Sample

Input:

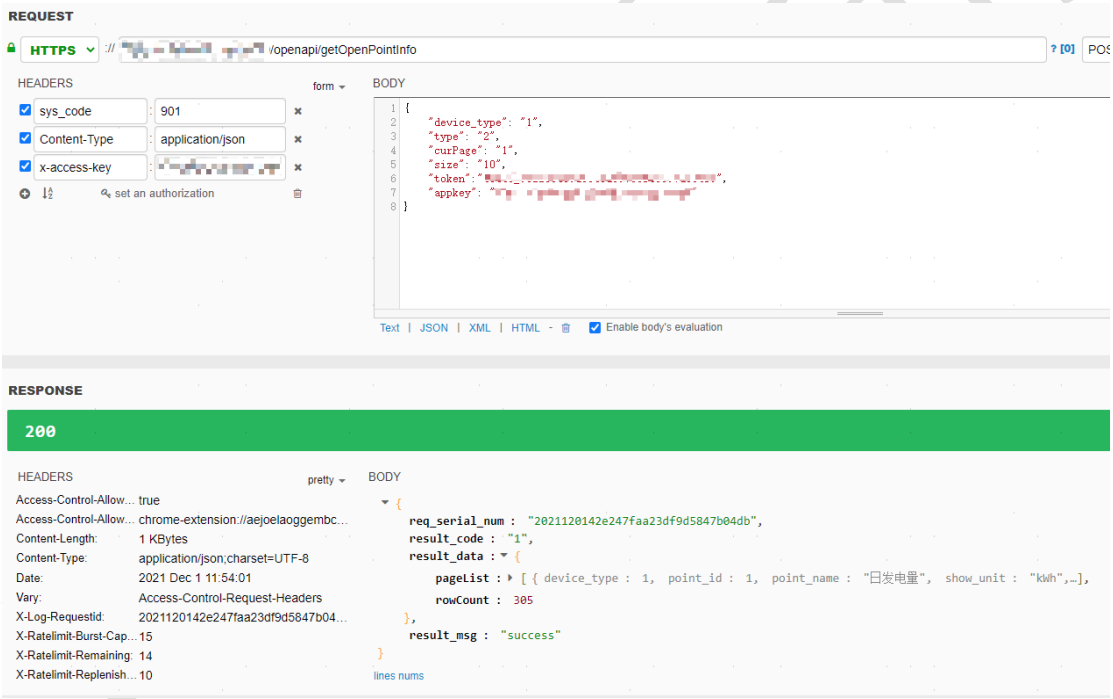

```
{
    "device_type": "1",
     "curPage": "1",
     "size": "10",
    "device_model_id": "**",
    "type": "2",
     "appkey": "********************",
     "token": "********************"
}
```


Output:


```
{
    "req_serial_num": "a42dd3ad84b94208a9925d57b5bbee12",
    "result_code": "1",
    "result_data": {
        "pageList": [
            {
                "device_type": 1,
                "open_point_remark": null,
                "point_id": 1,
                "point_name": "daily yield",
                "show_unit": "kWh",
                "storage_unit": "Wh"
            },
            {
                "device_type": 1,
                "open_point_remark": null,
```


严禁复制

```
                "point_id": 2,
                "point_name": "daily yield",
                "show_unit": "kWh",
                "storage_unit": "Wh"
            },
            ......
        ],
        "rowCount": 46
    },
    "result_msg": "success"
}
```

2.11 Query Device Property Point Value

2.11.1 Service Description

Query designated device property point value of multiple device(same type) under same plant.

2.11.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getDevPropertyPointValue

2.11.3 Input Parameter Description

Name	Type	Length	Description	Required?
ps_id	String	11	Plant id	Yes
device_type	String	2	Device type	Yes
ps_key_list	List<String>		Device ps_key list	Yes
point_id_list	List<String>		Property point id list Take properties related to Energy Storage System's battery capacity as example Point: Battery Capacity(Lithium Battery): 13258 Battery Capacity(Lead-Acid Battery): 13261 Specific open point definition could be acquired by getOpenPointInfo API.	Yes

2.11.4 Output Parameter Description

Name	Type	Length	Description
code	String	2	Result Code: 1: Success 2: Input is empty 3: No plant belongs to user 4: No data
property_point_value_list	List<Map>		Device property point data
uuid	int	11	Device uuid
device_type	int	11	Device type
ps_key	String	32	Device ps_key

property_code	String	32	Property point id
property_value	String	32	Property point value
unit	String	32	Unit corresponding to property value

2.11.5 Sample

Input:


```
{
    "appkey": "**************************",
    "ps_key_list": ['******_14_1_1'],
    "point_id_list": ['13251',
    '13257',
    '13258',
    '13261'],
    "device_type": "14",
    "ps_id": "******",
    "token": "**************************"
}
```

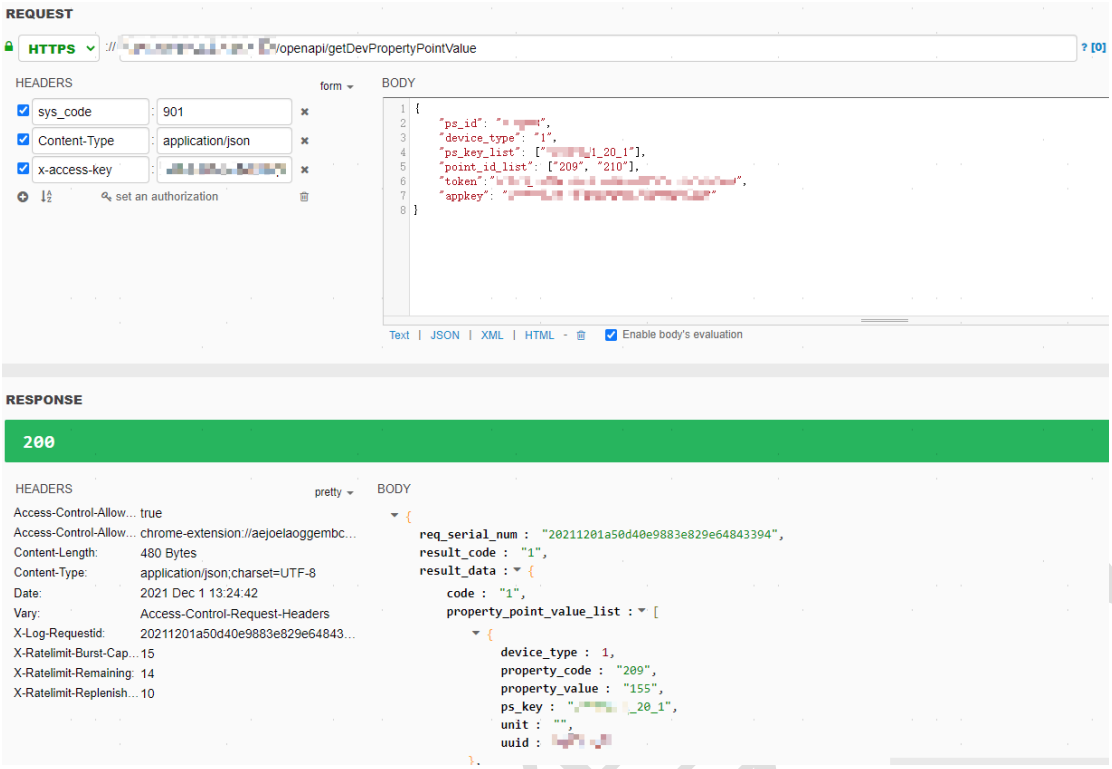

Output:


```
{
    "req_serial_num": "12acba4608464f2a80ae797da9e9c88e",
    "result_code": "1",
    "result_data": {
        "code": "1",
        "property_point_value_list": [
            {
                "device_type": 14,
                "property_code": "13251",
                "property_value": "CB0",
                "ps_key": "******_14_1_1",
                "unit": "",
                "uuid": ******
            },
            {
                "device_type": 14,
                "property_code": "13257",
```


非禁复制

```
            "property_value": "5000.0",
            "ps_key": "******_14_1_1",
            "unit": "Wp",
            "uuid": ******
        },
        {
            "device_type": 14,
            "property_code": "13258",
            "property_value": "0",
            "ps_key": "******_14_1_1",
            "unit": "Wh",
            "uuid": ******
        },
        {
            "device_type": 14,
            "property_code": "13261",
            "property_value": "200",
            "ps_key": "******_14_1_1",
            "unit": "Ah",
            "uuid": ******
        }
    ]
  },
  "result_msg": "success"
}
```


亚太区受限



2.12 Query Communication Device Info by Device SN

2.12.1 Service Description

Use device SN to query information of corresponding communication device.

2.12.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getCommunicationDevInfoByDevSn

2.12.3 Input Parameter Description

Name	Type	Length	Description	Required?
dev_sn	String	32	Device SN	Yes

2.12.4 Output Parameter Description

Name	Type	Length	Description
ps_id	int	11	Plant id

ps_name	String	64	Plant name
ps_key	String	32	Ps_key of communication device corresponding to device
uuid	int	11	Uuid of communication device corresponding to device
sn	String	32	SN of communication device corresponding to device
device_model_code	String	32	Model of communication device corresponding to device

2.12.5 Sample

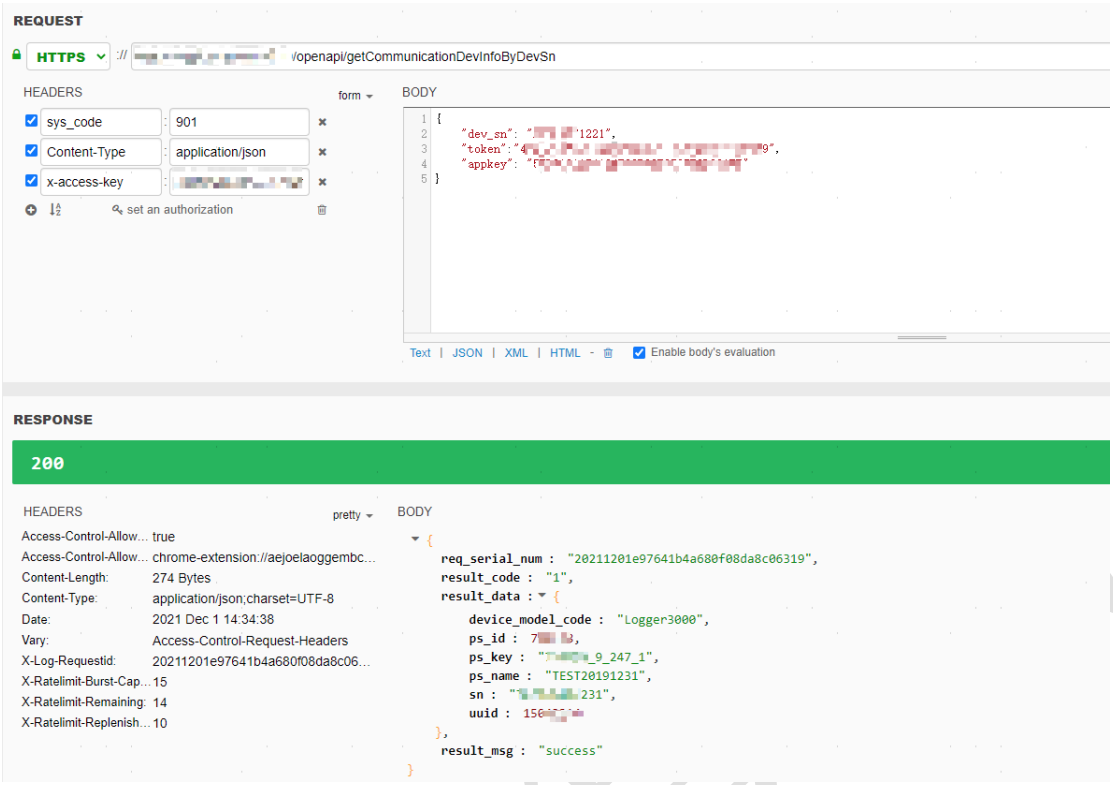
Input:


```
{
    "appkey":"******************",
    "dev_sn":"A555555",
    "token":"******************"
}
```


Output:


```
{
    "req_serial_num": "357d7ca95fe94fc8a8509c09af26d368",
    "result_code": "1",
    "result_data": {
        "device_model_code": "WiFi-V31",
        "ps_id": *****,
        "ps_key": "*****",
        "ps_name": "******",
        "sn": "******",
        "uuid": *****
    },
    "result_msg": "success"
}
```


严禁复制



2.13 Query Device List under The User

2.13.1 Service Description

Query the device list under the user

2.13.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getDeviceListByUser

2.13.3 Input Parameter Description

Name	Type	length	Description	Required?
ps_id	String	11	Plant ID	No
is_virtual_unit	String	1	Query virtual device? 1：query virtual device 0：query physical device this parameter is set to 0 by	No

			default Note: plant、Grid-connection Point and Unit belong to virtual device	
device_type_list	String[]	6	Device type list. If no conditions exist, query all types of device. Ex: 11：plant 1：Iverter 3：Grid-connection Point 17：Unit More details inAppendix 1 ：Device Type Dictionary Definition	No
rel_state	String	1	Device claim status：0：Not Claimed，1：Claimed	No
is_get_firmware_version	String	1	Is it necessary to acquire device firmware version info: 1：Need 0：Don't need Default as 0	No
curPage	String	32	Page number	Yes
size	String	32	Page size	Yes

2.13.4 Output Parameter Description

Name	Type	Length	Description
rowCount	Integer	11	Record sum
pageList	**List<Map>**		**Result List**

ps_id	String	11	Plant ID
ps_key	String	32	Device ps_key, it is used to query device data
uuid	Integer	11	Device uuid
type_name	String	32	Device type name
factory_name	String	64	Factory name
device_model_code	String	32	Device model name
device_model_id	Integer	11	Device model ID
device_name	String	64	Device name
device_type	Integer	11	Device type code
device_code	Integer	11	Device address code
chnnl_id	Integer	11	Device channel ID
rel_state	Integer	1	Device claim status: 0: Not Claimed 1: Claimed
device_sn	String	32	Device SN
communication_dev_sn	String	32	SN of communication device corresponding to device
dev_status	String	1	Device online status: 0：offline，1：online
dev_fault_status	String	1	Current device fault status: 1：Fault，2：Alarm，4:Normal
firmware_version_info	**Map**		Device firmware version information
lcd_version	String	100	Inverter ARM software version No.
mdsp_version	String	100	Inverter MDSP software version No.
sdsp_version	String	100	Inverter SDSP software version No.
pvd_version	String	100	String detection plate software version No.
cpld_version	String	100	CPLD firmware version No.
temp_version	String	100	Temperature plate software version

			No.
battery_version	String	100	Battery management system single board version No.
m_version	String	100	Communication device software version No.
system_version	String	100	Communication device hardware platform version No.

2.13.5 Sample

Input：


```
{
   "ps_id": "******",
   "is_get_firmware_version": "1",
   "device_type_list": ["1"],
   "curPage": "1",
   "size": "10",
   "token":"*****************************",
   "appkey": "**************************"
}
```


Output：


```
{
    "req_serial_num":"202112012ce4419bb0fbdd1a3f776279",
    "result_code":"1",
    "result_data":{
        "pageList":[
            {
                "ps_id":******,
                "chnnl_id":1,
                "communication_dev_sn":"***********",
                "dev_fault_status":"3",
                "dev_status":"1",
                "device_code":20,
```

```
                "device_model_code":"SG80KTL-M",
                "device_model_id":155,
                "device_name":"SG80KTL-M(COM4-007)_001_020",
                "device_sn":"**********",
                "device_type":1,
                "factory_name":"********company",
                "firmware_version_info":{
                    "cpld_version":"CPLD_SG80KTL-M_V11_A",
                    "lcd_version":"LCD_SG80KTL-M_V11_V01_P_M",
                    "mdsp_version":"MDSP_SG80KTL-M_V11_V1_B",
                    "pvd_version":"PVD_SG80KTL-M_V11_V1_A"
                },
                "ps_key":"******_1_20_1",
                "rel_state":1,
                "type_name":"inverter",
                "uuid":1200001
            }
        ],
        "rowCount":1
    },
    "result_msg":"success"
}
```


2.14 Query Device Real Time Data

2.14.1 Service Description

Query device real time point data by ps_key or SN(data unit is the minimum basic unit)

Note: Unit of return values is always basic unit. For example, the unit of yield value is Wh, the unit of power value is W, the unit of current value is A, the unit of voltage value is V.

2.14.2 Service Address

https:// API domain name which is provided by iSolarCloud/openapi/getDeviceRealTimeData

2.14.3 Input Parameter Description

Name	Type	Length	Description	Required?
ps_key_list	List<String>		List of device ps_key (same device type)	N
sn_list	List<String>		List of device SN (same device type)	N
point_id_list	List<String>		List of point id	Y
device_type	String		Device type code	Y

Note：ps_key_list and sn_list can not be empty at the same time；If both of them are not empty, use ps_key_list as query condition.

2.14.4 Output Parameter Description

Name	Type	Length	Description
fail_sn_list	List<String>		List of SN with no eligible ps_key found. Note：this parameter only exists when query by sn_list
device_point_list	**List<Map>**		**List of each device's point value, key of Map is device_point**
device_point	**Map**		**Point list**
device_name	**String**		**Device name**
device_time	String	32	Data update time
ps_id	String	32	Device ps_id
ps_key	String	32	Device ps_key
uuid	String	11	Device uuid
dev_status	Integer	11	Device status: 1：online 0：offline
dev_fault_status	Integer	11	Device fault alarm status: 1：Fault 2：Alarm 4：Normal
device_sn	String	32	Device SN

communication_dev_sn	String	32	SN of communication device corresponding to device
p+point_id	Object		Value of point,EX: p1 represents inverter yield

2.14.5 Sample

Input：


```
{
    "ps_key_list": ["******_1_15_1", "******_1_17_1"],
    "token":"**************************",
    "appkey": "************************************",
    "point_id_list":["1","2"]
}
```


Output：


```
{
    "req_serial_num": "202112019555499695 6ba3eb6480178c",
    "result_code": "1",
    "result_data": {
        "device_point_list": [
            {
                "device_point": {
                    "communication_dev_sn": "**********",
                    "device_name": "NEWSIV(001-15)_001_015",
                    "device_sn": null,
                    "device_time": "20210831113500",
                    "p1": "1970.0",
                    "p2":"1190970.0"
                    "ps_id": ******,
                    "ps_key": "**********",
                    "uuid": *********
                }
            }
```


业务受限

```
            ]
        },
        "result_msg": "success"
}
```


严禁复制

Appendix 1： Device Type Dictionary Definition

Device Type	Device Type Name	Device Type English Name
1	逆变器	Inverter
2	集装箱	Container
3	并网点	Grid-connection Point
4	汇流箱	Combiner Box
5	环境监测仪	Meteo Station
6	变压器	Transformer
7	电表	Meter
8	UPS	UPS

9	数据采集器	Data Logger
10	组串	String
11	电站	Plant
12	线路保护	Circuit Protection
13	解列装置	Splitting Device
14	储能逆变器	Energy Storage System
15	采集设备	Sampling Device
16	EMU	EMU
17	单元	Unit
18	温湿度传感器	Temperature and Humidity Sensor
19	智能配电柜	Intelligent Power Distribution Cabinet
20	显示设备	Display Device
21	交流配电柜	AC Power Distributed Cabinet
22	通信模块	Communication Module
23	系统 BMS	System-BMS
24	阵列 BMS	Array-BMS
25	直流-直流	DC-DC
26	能量管理系统	Energy Management System
27	跟踪系统	Tracking System
28	风能变流器	Wind Energy Converter
29	SVG	SVG
30	PT 柜	PT Cabinet
31	母线保护	Bus Protection
32	清扫机器人	Cleaning Device
33	直流屏	Direct Current Cabinet
34	公用测控	Public Measurement and Control

Appendix 2：API Error Code Definition

Error Code Name	Type	Error Code
success	success	1

error	Internal exception of service	-1
er_unknown_exception	unknown exception	000
er_missing_parameter:appkey	appkey should be empty	001
er_missing_parameter:token	token should be empty	002
er_missing_parameter:sys_code	sys_code should be empty	003
er_invalid_appkey	appkey is invalid	E00000
er_api_service_has_expired	API service has expired	E00001
er_parameter_decrypt_error	parameter decryption exception	E00002
er_token_login_invalid	token is invalid	E00003
er_hour_call_api_times_upper_limit	Number of API calls reach limit	E999
er_missing_parameter	required parameter is missing	009
er_parameter_value_invalid	parameter value is invalid	010
er_sql_exception	SQL Exception	011
Unauthorized access	unauthorized access	E900
Call too frequently	call too frequently	E901
Abnormal network environment	ip address switching is too frequent	E903
Request is not encrypted	Request is not encrypted	E902
Missing parameter in request header: x-random-secret-key	Missing parameter in request header：x-random-secret-key	E904
AES decryption exception	AES decryption exception	E905
RSA decryption exception	RSA decryption exception	E906
AES random secret key length must be 16	AES random secret key length must be 16	E907
Missing key parameter: api_key_param	Missing key parameter：api_key_param	E908
Invalid parameter format: nonce [32-bit string of numbers and letters]	Invalid parameter format：nonce [32 bit string of letters and numbers]	E909
Repeated request	repeated request，nonce in the	E910

	request need to regenerate	
Missing parameter in request header: x-access-key	missing parameter in request header：x-access-key	E911
Illegal x-access-key	illegal x-access-key	E912
Expired request	expired request，Differnece between timestamp（0 time zone UNIX timestamp）in request and servertime exceeds reasonable scale.	E913
Mismatched appkey and x-access-key	appkey and access-key is not match	E914
Login too frequently	login too frequentlu	E916

Appendix 3：Unencrypted API call Sample Code

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.HashMap;
import org.apache.http.HttpEntity;
import org.apache.http.client.methods.CloseableHttpResponse;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.StringEntity;
import org.apache.http.impl.client.CloseableHttpClient;
import org.apache.http.impl.client.HttpClients;
public class UserTest{
    Public static void main(String[] args)
    {
        CloseableHttpClienthttpclient = HttpClients.createDefault();
        String url = "https://API domain name which is provided by iSolarCloud/xxx/xxx";
        HttpPosthttppost = new HttpPost(url);
        try
        {
            // Header
```

```
                httppost.addHeader("sys_code", "901");
                HashMap<String, Object>req = new HashMap<String, Object>();
                // Public parameter
                req.put("appkey", "authorized appkey");
req.put("token", "token returned by login API");
                //   Service
                req.put("service", "service name");
                req.put("parameter 1", "parameter value");
                String jsonStr = com.alibaba.fastjson.JSON.toJSONString(req);
                System.out.println("send json-<" + jsonStr.toString());
                StringEntitystrEntity = new StringEntity(jsonStr);
                strEntity.setContentType("application/json");
                httppost.setEntity(strEntity);
                CloseableHttpResponse response = httpclient.execute(httppost);
                try
                {
                        HttpEntity entity = response.getEntity();
                        InputStreaminputStream = entity.getContent();
                        InputStreamReaderinputStreamReader = new InputStreamReader(
                                        inputStream, "UTF-8");
                        BufferedReader reader = new BufferedReader(inputStreamReader);
                        StringBuilder result = new StringBuilder();
                        String s;
                        while (((s = reader.readLine()) != null))
                        {
                                result.append(s);
                        }
                        reader.close();
                        System.out.println("receive json-<" + result.toString());
                }
                finally
                {
                        response.close();
                }
        }
        catch (Exception e)
        {
                e.printStackTrace();
        }
        finally
        {
```

```
                try
                {
                        httpclient.close();
                }
                catch (IOException e)
                {
                        e.printStackTrace();
                }
        }
    }
}
```

Appendix 4：RSA Encryption Sample Code

```
import org.apache.commons.codec.binary.Base64;
import org.apache.commons.io.IOUtils;
/**
* RSA Encryption rule:
* Secret key format: PKCS#8
* Output format: Base64
* Character set: utf8 encode;
**/
public String publicEncrypt(String data, String publicKey) {
    try {
            KeyFactory keyFactory = KeyFactory.getInstance("RSA");
            X509EncodedKeySpec x509KeySpec = new X509EncodedKeySpec(Base64.decodeBase64(publicKey));
            RSAPublicKey key = (RSAPublicKey)keyFactory.generatePublic(x509KeySpec);
            Cipher cipher = Cipher.getInstance("RSA");
        cipher.init(Cipher.ENCRYPT_MODE, key);
        return Base64.encodeBase64URLSafeString(rsaSplitCodec(cipher, Cipher.ENCRYPT_MODE,
            data.getBytes("UTF-8"), key.getModulus().bitLength()));
    } catch (Exception var3) {
        //Deal Exception
```

```
    }
}

private byte[] rsaSplitCodec(Cipher cipher, int opmode, byte[] datas, int keySize){
    int maxBlock = 0;
    if(opmode == Cipher.DECRYPT_MODE){
        maxBlock = keySize / 8;
    }else{
        maxBlock = keySize / 8 - 11;
    }
    ByteArrayOutputStream out = new ByteArrayOutputStream();
    int offSet = 0;
    byte[] buff;
    int i = 0;
    try{
        while(datas.length > offSet){
            if(datas.length-offSet > maxBlock){
                buff = cipher.doFinal(datas, offSet, maxBlock);
            }else{
                buff = cipher.doFinal(datas, offSet, datas.length-offSet);
            }
            out.write(buff, 0, buff.length);
            i++;
            offSet = i * maxBlock;
        }
    }catch(Exception e){
        // Deal Exception
    }
    byte[] resultDatas = out.toByteArray();
    IOUtils.closeQuietly(out);
    return resultDatas;
}
```

严禁复制

Appendix 5: AES Encryption Sample Code

```
/**
* AES encryption rule:
* Encryption mode: ECB
* Padding method: pkcs5padding
* data block: 128 bit
* Offset: no offset
* Output: hex
* Character set: utf8 encoding
**/
public String encrypt(String content, String password)   throws Exception {
    try {
        byte[] result = null;
        byte[] passwordBytes = getSecretKey(password) ;
        SecretKeySpec skeySpec = new SecretKeySpec(passwordBytes, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
        result = cipher.doFinal(content.getBytes("UTF-8"));
        return parseByte2HexStr(result) ;
    } catch (Exception e) {
        //Deal Exception
    }
}

public byte[] getSecretKey(String key) throws Exception{
    final byte paddingChar = '0';
    byte[] realKey = new byte[16];
    byte[] byteKey = key.getBytes("UTF-8");
    for (int i =0;i<realKey.length;i++){
        if (i<byteKey.length){
            realKey[i] = byteKey[i];
        }else{
```

```
                realKey[i] = paddingChar;
            }
        }
        return realKey;
    }

    public String parseByte2HexStr(byte buf[]) {
        StringBuffer sb = new StringBuffer();
        for (int i = 0; i < buf.length; i++) {
            String hex = Integer.toHexString(buf[i] & 0xFF);
            if (hex.length() == 1) {
                hex = '0' + hex;
            }
            sb.append(hex.toUpperCase());
        }
        return sb.toString();
    }
```

Appendix 6：AES Decryption Sample Code

```
/**
* Decryption mode：ECB
* Padding method：pkcs5padding
* Data block：128 bit
* Offset：no offset
* Output：hex
* Character set：utf8 encoding;
**/
public   String decrypt(String content, String password) throws Exception {
```

```
    try {
        byte[] original = null;
        byte[] decryptFrom = parseHexStr2Byte(content);
        byte[] passwordBytes = getSecretKey(password) ;
        SecretKeySpec skeySpec = new SecretKeySpec(passwordBytes, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        cipher.init(Cipher.DECRYPT_MODE, skeySpec);
        original = cipher.doFinal(decryptFrom);
        return new String(original);
    } catch (Exception e) {
        //Deal Exception
    }
}

public byte[] parseHexStr2Byte(String hexStr) {
    if (hexStr.length() < 1) {
        return null;
    }
    byte[] result = new byte[hexStr.length() / 2];
    for (int i = 0; i < hexStr.length() / 2; i++) {
        int high = Integer.parseInt(hexStr.substring(i * 2, i * 2 + 1), 16);
        int low = Integer.parseInt(hexStr.substring(i * 2 + 1, i * 2 + 2),
                16);
        result[i] = (byte) (high * 16 + low);
    }
    return result;
}
```

Appendix 7：API Encrypted Call Sample Code

1. Header encapsulation

```
String publicKey = "xxxxx" ;// publicKey assigned by iSolarCloud

// publicEncrypt methods in Appendix 3

String x-random-secret-key = publicEncrypt("A123456zA123456z", publicKey) ;//x-random-secret-key could be different in different

requests

String x-access-key = "i71w7tskmns5********i3b8zqncvay3" ;//accessKey assigned by iSolarCloud
```

2. Body encapsulation

```
// Use login API as an example，Besides request body parameter  in 2.1, api_key_param is also in need：

//nonce is a 32 bit random string of letters and numbers, it should be different in every call;

//timestamp is UNIX time stamp in milliseconds，if error code E913 is returned,

// Use GET method to call  https://API domain name which is provided by iSolarCloud/timestamp

String requestBody =

{

    "api_key_param": {

        "nonce": "cb360459bd624c6ab15308c4b6847856",

        "timestamp": "1616725497384"

    },

    "appkey": "***********",

    "login_type": "1",

    "user_account": "******",

    "user_password": "******"

};
```

3. Request call

```
CloseableHttpClient httpclient = HttpClients.createDefault();

String url = "https://API domain name which is provided by iSolarCloud/v1/userService/login";

HttpPost httppost = new HttpPost(url);

// Request header

httppost.addHeader("x-random-secret-key", x-random-secret-key);

httppost.addHeader("x-access-key", x-access-key);

..... //Set other request header

// Encryption methods in Appendix 4

String encryptedRequestBody = encrypt(requestBody, "A123456zA123456z") ;
```

```
StringEntity strEntity = new StringEntity(encryptedRequestBody);

strEntity.setContentType("application/json");

httppost.setEntity(strEntity);

CloseableHttpResponse response = httpclient.execute(httppost);

HttpEntity entity = response.getEntity();

InputStream inputStream = entity.getContent();

InputStreamReader inputStreamReader = new InputStreamReader(inputStream, "UTF-8");

BufferedReader reader = new BufferedReader(inputStreamReader);

StringBuilder responseBody = new StringBuilder();

String s = null;

while (((s = reader.readLine()) != null))

{

    responseBody.append(s);

}

reader.close();

// Decryption methods in Appendix 5

String decryptedResponseBody = decrypt(responseBody, "A123456zA123456z") ;
```

Appendix 8：API Call Help

API call address in screenshots should not be used. Use API call address in API call authorization documentation.

Pre-request of API call： Acquired user account、user password、appkey、access_key、API call address、API documentation .etc in API call authorization documentation which is provided by iSolarCloud already.

API call address:

Replace https://*API domain name which is provided by iSolarCloud* in service address in API definition with API domain name of corresponding server which is provided by iSolarCloud :

Chinese Server：https://gateway.isolarcloud.com/

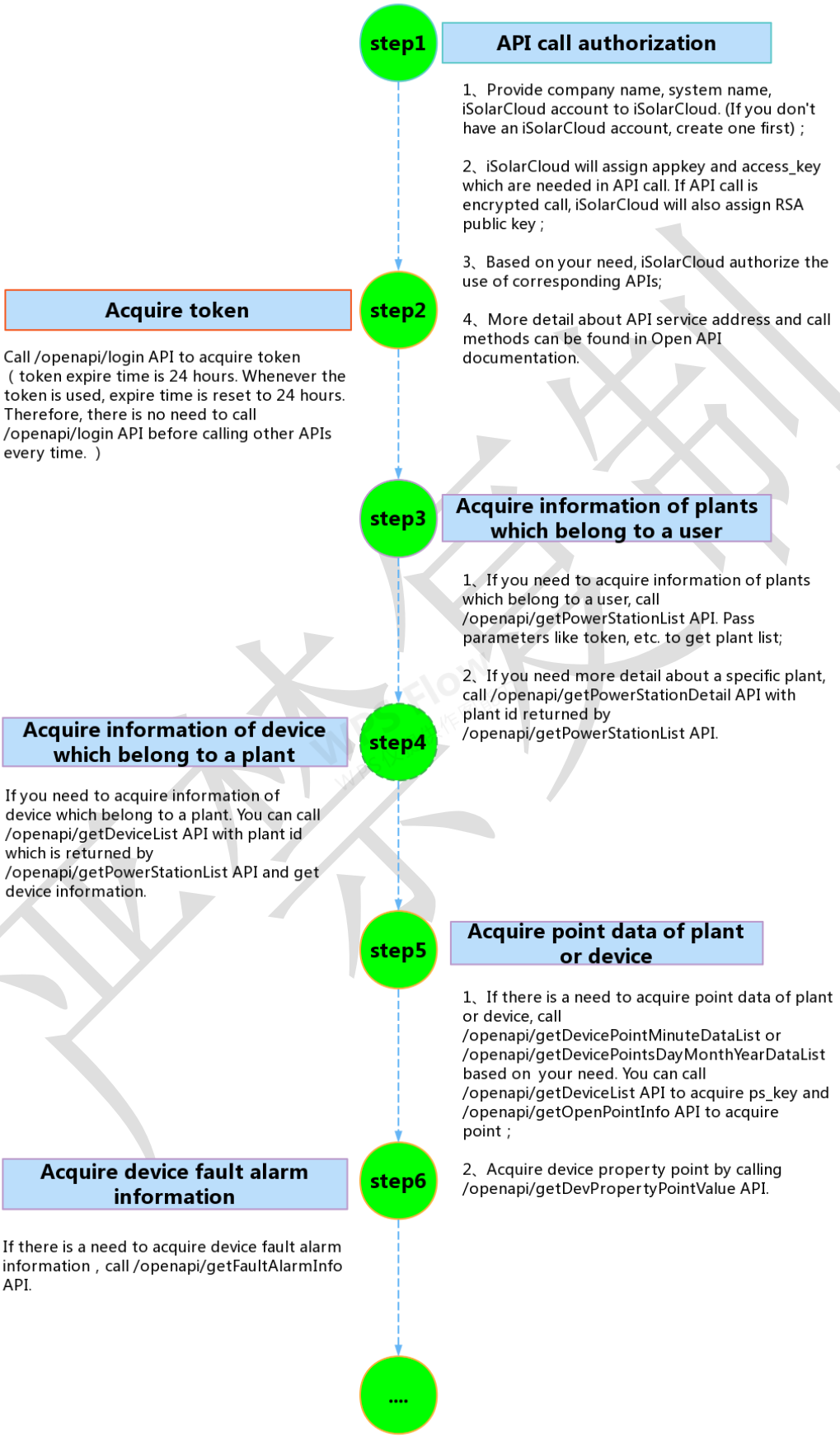
International Server：https://gateway.isolarcloud.com.hk/

European Server：https://gateway.isolarcloud.eu/

Australian Server: https://augateway.isolarcloud.com/

API Call Steps:

严禁复制


step1
API call authorization
1、Provide company name, system name, iSolarCloud account to iSolarCloud. (If you don't have an iSolarCloud account, create one first) ;
2、iSolarCloud will assign appkey and access_key which are needed in API call. If API call is encrypted call, iSolarCloud will also assign RSA public key ;
3、Based on your need, iSolarCloud authorize the use of corresponding APIs;
4、More detail about API service address and call methods can be found in Open API documentation.
step2
Acquire token
Call /openapi/login API to acquire token （token expire time is 24 hours. Whenever the token is used, expire time is reset to 24 hours. Therefore, there is no need to call /openapi/login API before calling other APIs every time. ）
step3
Acquire information of plants which belong to a user
1、If you need to acquire information of plants which belong to a user, call /openapi/getPowerStationList API. Pass parameters like token, etc. to get plant list;
2、If you need more detail about a specific plant, call /openapi/getPowerStationDetail API with plant id returned by /openapi/getPowerStationList API.
step4
Acquire information of device which belong to a plant
If you need to acquire information of device which belong to a plant. You can call /openapi/getDeviceList API with plant id which is returned by /openapi/getPowerStationList API and get device information.
step5
Acquire point data of plant or device
1、If there is a need to acquire point data of plant or device, call /openapi/getDevicePointMinuteDataList or /openapi/getDevicePointsDayMonthYearDataList based on your need. You can call /openapi/getDeviceList API to acquire ps_key and /openapi/getOpenPointInfo API to acquire point ;
2、Acquire device property point by calling /openapi/getDevPropertyPointValue API.
step6
Acquire device fault alarm information
If there is a need to acquire device fault alarm information , call /openapi/getFaultAlarmInfo API.
....