

SUNGROW
IsolarCloud Open API
Guia Prático

Sumário

1.	Especificações de uso da API	4
1.1.	Etapas de Acesso.....	4
1.2.	Definição de Erros	4
1.3.	Descrição de conteúdo	4
2.	Login.....	6
2.1.	Descrição.....	6
2.2.	Endereço da requisição	6
2.3.	Parâmetros de entrada.....	6
2.4.	Parâmetros de saída.....	6
2.5.	Exemplo.....	7
2.5.1.	Requisição	7
2.5.2.	Resposta.....	8
3.	Informações básicas da usina	8
3.1.	Descrição.....	8
3.2.	Endereço da requisição	8
3.3.	Parâmetros de entrada.....	9
3.4.	Parâmetros de saída.....	9
3.5.	Exemplo.....	11
3.5.1.	Requisição	11
3.5.2.	Resposta.....	11
4.	Lista de dispositivos da usina	12
4.1.	Descrição.....	12
4.2.	Endereço da requisição	12
4.3.	Parâmetros de entrada.....	13
4.4.	Parâmetros de saída.....	13
4.5.	Exemplo.....	14
4.5.1.	Requisição	14
4.5.2.	Resposta.....	15
5.	Dados em tempo real.....	17
5.1.	Descrição.....	17

5.2.	Endereço da requisição.....	18
5.3.	Parâmetros de entrada.....	18
5.4.	Parâmetros de saída.....	18
5.5.	Exemplo.....	19
5.5.1.	Requisição.....	19
5.5.2.	Resposta.....	20
6.	Dados históricos (Diário, mensal ou anual).....	21
6.1.	Descrição.....	21
6.2.	Endereço da requisição.....	21
6.3.	Parâmetros de entrada.....	21
6.4.	Parâmetros de saída.....	22
6.5.	Exemplo.....	23
6.5.1.	Requisição.....	23
6.5.2.	Resposta.....	24
7.	Endereço dos dispositivos.....	26
8.	Pontos de medição.....	26
8.1.1.	Usina.....	26
8.1.2.	Inversor.....	27
8.1.3.	Microinversor.....	28
9.	Definição de erros.....	28
10.	Exemplo de Aplicação.....	29
10.1.	Requisição.....	29

1. Especificações de uso da API

A API do iSolarCloud permite a integração de sistemas externos com nossos serviços de forma simples, segura e padronizada. Por meio dela, é possível consultar informações em tempo real, garantindo maior flexibilidade para automações, relatórios e integrações personalizadas.

Esse documento apresenta uma visão detalhada de como utilizar a API, incluindo:

- Método de autenticação e segurança;
- Estrutura de requisições e respostas;
- Endpoints disponíveis e suas finalidades;
- Exemplos práticos de utilização.

1.1. Etapas de Acesso

Os usuários podem obter informações sobre a usina e os dispositivos com a API do iSolarCloud. Na primeira etapa, os usuários chamam a API `/openapi/login` para identificar a autenticação com a conta de usuário e senha fornecidas pelo iSolarCloud e obter o token de autorização. Em seguida, os usuários podem chamar outras requisições na API dentro do prazo de validade do token (24 horas).

Nota: Após o período de validade do token será necessário efetuar o login novamente para gerar um novo token, desse modo, indicamos integrar a requisição de login junto ao looping de requisição, assim o token sempre será atualizado automaticamente.

1.2. Definição de Erros

Mais detalhes podem ser encontrados em **9. Definição de Erros**.

1.3. Descrição de conteúdo

(1) Credenciais de acesso (e-mail e senha) são os mesmos utilizados para acessar o iSolarCloud diretamente.

(2) Chaves de acesso (appkey/x-access-key) podem ser geradas diretamente em [Sungrow Developer Portal](#) seguindo o tutorial [Como criar as chaves de acesso API](#).

(3) Requisições via HTTP são sempre enviadas em **POST**.

(4) Domínio para todas as requisições:
<https://gateway.isolarcloud.com.hk/openapi/>

(5) Appkey é enviada em todas as requisições no **body** como **appkey**.

(6) Todas as requisições necessitam do token para verificação das credenciais. [openapi/login é uma exceção].

(7) Parâmetros necessários no **headers** de todas as requisições:

Nome	Tipo	Comprimento	Descrição	Obrigatório?
Content-Type	String	32	Valor: application/json	Sim
sys_code	String	11	Valor: 901	Sim
lang	String	6	Valor: _en_US	Sim
x-access-key	String	32	Corresponde à secret key gerada no Developer Portal	Sim

(8) Parâmetros necessários no **body** de todas as requisições:

Nome	Tipo	Comprimento	Descrição	Obrigatório?
appkey	String	32	Corresponde à AppKey gerada no Developer Portal	Sim
token	String	40	Token de acesso gerado após o login na API	Sim

2. Login

2.1. Descrição

Todos os usuários devem efetuar o login para obter o token de acesso.

2.2. Endereço da requisição

<https://gateway.isolarcloud.com.hk/openapi/login>

2.3. Parâmetros de entrada

Nome	Tipo	Comprimento	Descrição	Obrigatório?
user_account	String	32	E-mail de acesso do iSolarCloud	Sim
user_password	String	32	Senha de acesso do iSolarCloud	Sim

2.4. Parâmetros de saída

Nome	Tipo	Comprimento	Descrição
login_state	String	32	Login Status: -1: Conta não existe 0: Senha incorreta 1: Login efetuado 2: Conta bloqueada por múltiplas tentativas com a senha errada 5: Conta bloqueada pelo administrador
token	String	32	Token de acesso, retornado após login efetuado
user_id	String	11	ID do usuário

user_name	String	64	Nome do usuário
language	String	32	Idioma
user_account	String	32	Conta do usuário
mobile_tel	String	20	Telefone do usuário
email	String	64	Email do usuário
err_times	String	2	Tentativas de acesso
remain_times	String	2	Tentativas restantes
country_name	String	32	País do usuário
country_id	String	2	ID do país
user_master_org_id	String	11	ID da organização do usuário
user_master_org_name	String	128	Nome da organização do usuário

2.5. Exemplo

2.5.1. Requisição

```
{
url = "https://gateway.isolarcloud.com.hk/openapi/login"
payload = json.dumps({
  "appkey": "*****",
  "user_account": "*****",
  "user_password": "*****"
})
headers = {
  'Content-Type': 'application/json',
  'sys_code': '901',
  'lang': '_en_US',
  'x-access-key': '*****'
}
response = requests.request("POST", url, headers=headers,
data=payload)
print(response.text)
```

2.5.2. Resposta

```
{
  "req_serial_num": "*****",
  "result_code": "1",
  "result_msg": "success",
  "result_data": {
    "user_master_org_id": "*****",
    "mobile_tel": null,
    "user_name": "*****",
    "language": "*****",
    "token": "*****",
    "err_times": "0",
    "user_id": "*****",
    "login_state": "1",
    "disable_time": null,
    "country_name": "*****",
    "user_account": "*****",
    "user_master_org_name": "*****",
    "email": "*****",
    "country_id": "**"
  }
}
```

3. Informações básicas da usina

3.1. Descrição

Essa requisição apresenta informações básicas da usina como: localização, dispositivos, tipo de dispositivo, nome, data de instalação e status.

Também é possível encontrar o ID e a chave da usina, esses dados podem ser necessários para outras requisições.

3.2. Endereço da requisição

<https://gateway.isolarcloud.com.hk/openapi/getPowerStationDetail>

3.3. Parâmetros de entrada

Nome	Tipo	Exemplo	Descrição	Obrigatório?
sn	String	B2262720904	SN do dispositivo*	Sim

3.4. Parâmetros de saída

Nome	Tipo	Exemplo	Descrição
req_serial_num	String	202212236d134e3386b3c49b66c966c5	Número de identificação da requisição
result_code	String	1	Código de erro
result_data	Map	-	Conteúdo da resposta
result_data.alarm_count	Int	0	Alarmes
result_data.build_status	Int	2	Status de criação da planta: 0: Não iniciado 1: Em criação 2: Online 4: Offline
result_data.communication_dev_detail_list	Lista	-	Lista de dispositivos de comunicação da usina
result_data.communication_dev_detail_list.is_enable	Int	1	Status do dispositivo de comunicação: 1: Normal 0: Desabilitado
result_data.communication_dev_detail_list.sn	String	A2462**0013	SN dos dispositivos de comunicação
result_data.connect_type	Int	2	Tipo de conexão à rede: 1: 100% exportação 2: Autoconsumo 3: Zero Export 4: Off Grid
result_data.design_capacity	Int	10000	Potência instalada (Wp)
result_data.fault_count	Int	0	Número de falhas
result_data.email	String	*****@gmail.com	E-mail do proprietário da usina

result_data.install_date	String	2022-05-27 11:33:06	Data de conexão da usina
result_data.latitude	Num	41.6939512369 7499	Latitude
result_data.longitude	Num	123.1821215140 4492	Longitude
result_data.param_income_unit_name	String	BRL	Moeda local
result_data.ps_current_time_zone	String	GMT-3	Time Zone
result_data.ps_fault_status	Int	3	Status de operação usina: 1: Falha 2: Alarme 3: Normal
result_data.ps_id	Int	974621	ID da usina
result_data.ps_key	String	974621_11_0_0	Chave da usina
result_data.ps_location	String	Av. **** - São Paulo - SP, ***** , Brazil	Endereço da usina
result_data.ps_name	String	PlantA	Nome da usina
result_data.ps_price	String	0.001	Preço da energia em Wh
result_data.ps_price_kwh	String	1	Preço da energia em kWh
result_data.ps_status	Int	1	Status da usina: 1: Online 0: Offline
result_data.ps_type	Int	3	Tipo da usina: 1: Utility; 3: Distributed PV; 4: Residential PV; 5: Residential ESS; 6: Village; 7: Distributed ESS; 12: C&I ESS
result_data.ps_type_name	String	Commercial ESS	Nome do tipo da usina

3.5. Exemplo

3.5.1. Requisição

```
{
url =
"https://gateway.isolarcloud.com.hk/openapi/getPowerStationDetail"
payload = json.dumps({
  "appkey": "*****",
  "token": "*****",
  "sn": "*****"
})
headers = {
  'Content-Type': 'application/json',
  'sys_code': '901',
  'lang': '_en_US',
  'x-access-key': '*****'
}
response = requests.request("POST", url, headers=headers,
data=payload)
print(response.text)
```

3.5.2. Resposta

```
{
  "req_serial_num": "*****",
  "result_code": "1",
  "result_msg": "success",
  "result_data": {
    "design_capacity": null,
    "design_capacity_battery": 225000.0,
    "alarm_count": 0,
    "ps_key": "*****",
    "latitude": ***** ,
    "description": null,
    "ps_price_kwh": "1",
    "ps_fault_status": 3,
    "ps_type_name": "C&I ESS",
    "design_capacity_pcs": 110000.0,
    "build_status": 2,
    "install_date": "*****",
    "ps_type": 12,
    "email": "*****",
    "longitude": ***** ,
  }
}
```

```
"param_income_unit_name": "BRL",
"ps_price": "0.001",
"ps_name": "*****",
"share_user_type": null,

"ps_location": "*****",
"share_type": "0",
"ps_current_time_zone": "GMT-3",
"user_moble_tel": null,
"ps_id": *****,
"communication_dev_detail_list": [
  {
    "is_enable": 1,
    "sn": "*****"
  }
],
"connect_type": 2,
"ps_status": 1,
"fault_count": 0
}
```

4. Lista de dispositivos da usina

4.1. Descrição

Lista todos os dispositivos presentes na usina e informações importantes de cada dispositivo como SN, ID e tipo de dispositivo, informações que são necessárias em outras requisições como dados em tempo real.

4.2. Endereço da requisição

<https://gateway.isolarcloud.com.hk/openapi/getDeviceList>

4.3. Parâmetros de entrada

Nome	Tipo	Exemplo	Descrição	Obrigatório?
ps_id	String	476178	ID da usina	Sim
curPage	Int	1	Página	Sim
size	Int	10	Tamanho da página	Sim
is_virtualUnit	String	0	1: Dispositivo virtual (Usina, Ponto de conexão etc.) 0: Dispositivos físicos Considerado 0 por padrão	Não
device_type_list	Lista	1	Tipo de dispositivo (11: Usina, 1: Inversor). Verifique o Endereço dos dispositivos para mais detalhes. Se vazio, considera todos os dispositivos por padrão	Não
is_get_firmware_version	String	0	Deseja a versão do firmware? 0: Não; 1: Sim, padrão como "Não" se não informado	Não

4.4. Parâmetros de saída

Nome	Tipo	Exemplo	Descrição
req_serial_num	String	202212236d134e3386b3c49b66c966c5	Número de identificação da requisição
result_code	String	1	Código de erro
result_data	Map	-	Conteúdo da resposta
result_data.pageList	Lista	-	Lista de dispositivos
result_data.pageList.chnnl_id	Int	4	Channel ID do dispositivo

result_data.pageList.communication_dev_sn	String	A2462**0013	SN do dispositivo de comunicação
result_data.pageList.dev_fault_status	Int	4	Status do dispositivo: 1: Falha 2: Alarime 4: Normal
result_data.pageList.device_code	Int	247	Endereço do dispositivo
result_data.pageList.device_model_code	String	PCS	Modelo do dispositivo
result_data.pageList.device_model_id	Int	367701	ID do modelo do dispositivo
result_data.pageList.device_name	String	Commercial & Industrial ESS1	Nome do dispositivo
result_data.pageList.device_sn	String	A2462**0013	SN do dispositivo
result_data.pageList.device_type	Int	37	Código do tipo de dispositivo
result_data.pageList.ps_id	Int	473465	ID da usina
result_data.pageList.ps_key	String	7006*6130_64_247_1	PS_key do dispositivo (Utilizado para requisitar dados específicos do dispositivo)
result_data.pageList.uuid	Int	2024**5175	UUID do dispositivo

4.5. Exemplo

4.5.1. Requisição

```
url = "https://gateway.isolarcloud.com.hk/openapi/getDeviceList"

payload = json.dumps({
    "token": "*****",
    "appkey": "*****",
    "ps_id": *****,
    "device_type_list": [],
    "curPage": 1,
    "size": 50
})

headers = {
    'Content-Type': 'application/json',
    'sys_code': '901',
```

```
'lang': '_en_US',
'x-access-key': '*****'
}

response = requests.request("POST", url, headers=headers,
data=payload)

print(response.text)
```

4.5.2. Resposta

```
{
  "req_serial_num": "*****",
  "result_code": "1",
  "result_msg": "success",
  "result_data": {
    "pageList": [
      {
        "chnnl_id": 1,
        "type_name": "PCS",
        "ps_key": "16*****_37_4_1",
        "rel_time": "*****",
        "device_sn": "A24B*****2",
        "dev_status": "1",
        "device_type": 37,
        "factory_name": "SUNGROW",
        "uuid": "*****",
        "grid_connection_date": "*****",
        "device_name": "Energy Storage System1",
        "dev_fault_status": 4,
        "rel_state": 1,
        "device_code": 4,
        "ps_id": "*****",
        "device_model_id": "*****",
        "communication_dev_sn": "S2*****3",
        "device_model_code": "SC110CX"
      },
      {
        "chnnl_id": 1,
        "type_name": "Battery cluster management unit",
        "ps_key": "16****_44_3_1",
        "rel_time": "*****",
        "device_sn": null,
        "dev_status": "1",
        "device_type": 44,
        "factory_name": "SUNGROW",
```

```

"uuid": *****,
"grid_connection_date": "*****",
"device_name": "Battery Cluster Management
Unit1",
"dev_fault_status": 4,
"rel_state": 1,
"device_code": 3,
"ps_id": *****,
"device_model_id":*****,
"communication_dev_sn": "S2*****3",
"device_model_code": "CMU200CP"
},
{
"chnnl_id": 1,
"type_name": "Energy storage system",
"ps_key": "16****_78_2_1",
"rel_time": "*****",
"device_sn": null,
"dev_status": "1",
"device_type": 78,
"factory_name": null,
"uuid": *****,
"grid_connection_date": "*****",
"device_name": "Commercial & Industrial ESS1",
"dev_fault_status": 4,
"rel_state": 1,
"device_code": 2,
"ps_id": *****,
"device_model_id": null,
"communication_dev_sn": "S2*****3",
"device_model_code": null
},
{
"chnnl_id": 1,
"type_name": "Local controller",
"ps_key": "16****_45_2_1",
"rel_time": "*****",
"device_sn": "S2*****1",
"dev_status": "1",
"device_type": 45,
"factory_name": "SUNGROW",
"uuid": *****,
"grid_connection_date": "*****",
"device_name": "Local Controller1",
"dev_fault_status": 4,
"rel_state": 1,
"device_code": 2,
"ps_id": *****,
"device_model_id": *****,

```

```
    "communication_dev_sn": "S2*****3",
    "device_model_code": "LC-ST200CS"
  },
  {
    "chnnl_id": 1,
    "type_name": "Energy management system",
    "ps_key": "16****_26_247_1",
    "rel_time": "*****",
    "device_sn": "S2*****3",
    "dev_status": "1",
    "device_type": 26,
    "factory_name": "SUNGROW",
    "uuid": "*****",
    "grid_connection_date": "*****",
    "device_name": "Energy management system1",
    "dev_fault_status": 4,
    "rel_state": 1,
    "device_code": 247,
    "ps_id": "*****",
    "device_model_id": "*****",
    "communication_dev_sn": "S2*****3",
    "device_model_code": "EMS300CP"
  }
],
"rowCount": 5
}
```

5. Dados em tempo real

5.1. Descrição

Requisição de dados em tempo real.

Essa requisição sempre retorna a menor unidade de cada valor, ou seja, geração em Wh, potência em W, corrente em A e tensão em V.

A atualização dos dados ocorre a cada 5 minutos, recomenda-se que a frequência de chamados não seja inferior a esse tempo.

5.2. Endereço da requisição

<https://gateway.isolarcloud.com.hk/openapi/getDeviceRealTimeData>

5.3. Parâmetros de entrada

Nome	Tipo	Exemplo	Descrição	Obrigatório?
point_id_list	List	["83022".83033]	Endereço do ponto de medição. Verificar os Pontos de medição disponíveis para cada dispositivo	Sim
ps_key_list	List	["972018_11_0_0", "971822_11_0_0"]	Ps_key dos respectivos dispositivos. * Aceita apenas dispositivos do mesmo tipo	Não
device_type	Int	11	Código do tipo de dispositivo **	Sim
sn_list	List	['A21B1801347','A29B1902347']	SN dos respectivos dispositivos. ***	Não

* Ps_key pode ser encontrada na lista de dispositivos da usina /getDeviceList.

** O tipo de dispositivo deve ser correspondente aos dispositivos listados no ps_key_list ou sn_list.

*** Deve ser fornecido a lista de ps_key ou sn_list, ambos não podem ficar vazios.

5.4. Parâmetros de saída

Nome	Tipo	Exemplo	Descrição
req_serial_num	String	202212236d134e3386b3c49b66c966c5	Número de identificação da requisição
result_code	String	1	Código de erro
result_data	Map	-	Conteúdo da resposta
result_data.device_point_list	Lista	-	Lista de retorno dos dispositivos
result_data.device_point_list.device_point	Map	-	Lista de dados de cada dispositivo
result_data.device_point_list.device_point.communicat	String	A2462**0013	SN do dispositivo de comunicação

ion_dev_sn			
result_data.device_point_list.device_point.dev_fault_status	Int	4	Status do dispositivo: 1: Falha 2: Alarme 4: Normal
result_data.device_point_list.device_point.device_name	String	Hybrid Inverter2	Nome do dispositivo
result_data.pageList.device_model_code	String	PCS	Modelo do dispositivo
result_data.device_point_list.device_point.device_sn	String	A2462**0013	SN do dispositivo
result_data.pageList.device_name	String	Commercial & Industrial ESS1	Nome do dispositivo
result_data.pageList.device_sn	String	A2462**0013	SN do dispositivo
result_data.device_point_list.device_point.device_time	String	20241127151500	Horário da última atualização de dados
result_data.device_point_list.device_point.p+pointid	Map	-	Dados dos pontos de medição solicitados
result_data.device_point_list.device_point.ps_id	Int	970645	ID da usina
result_data.device_point_list.device_point.ps_key	String	970645_11_0_0	Ps_key do dispositivo
result_data.fail_ps_key_list	Lista	["700926130_22_24_7_2","700926130_22_24_7_3"]	Lista de ps_key que falharam na requisição

5.5. Exemplo

5.5.1. Requisição

```
url =
"https://gateway.isolarcloud.com.hk/openapi/getDeviceRealTimeData"

payload = json.dumps({
  "token": "*****",
```

```

"appkey": "*****",
"device_type": 44,
"point_id_list": [
  "59008",#Tensão máxima da célula
  "59010",#Tensão mínima da célula
  "59012",#Máx. temperatura de célula
  "59014",#Min. temperatura de célula
  "59004" #SOC
],
"ps_key_list": [
  "1*****_44_3_1"
]
})
headers = {
  'Content-Type': 'application/json',
  'sys_code': '901',
  'lang': '_en_US',
  'x-access-key': '*****'
}

response = requests.request("POST", url, headers=headers,
data=payload)

print(response.text)

```

5.5.2. Resposta

```

{
  "req_serial_num": "*****",
  "result_code": "1",
  "result_msg": "success",
  "result_data": {
    "fail_ps_key_list": [],
    "device_point_list": [
      {
        "device_point": {
          "p59008": "3300.3",
          "ps_key": "1*****_44_3_1",
          "device_sn": null,
          "dev_status": 1,
          "p59004": "37.4",
          "uuid": "*****",
          "device_name": "Battery Cluster Management
Unit1",
          "dev_fault_status": 4,

```

```

    "ps_id": "*****",
    "p59012": "27.3",
    "communication_dev_sn": "*****",
    "p59014": "25.7",
    "p59010": "3298.8",
    "device_time": "20251003102500"
  }
}
]
}
}

```

6. Dados históricos (Diário, mensal ou anual)

6.1. Descrição

Nessa requisição é possível extrair dados históricos sobre os mesmos pontos de medição utilizados nos dados em tempo real.

6.2. Endereço da requisição

<https://gateway.isolarcloud.com.hk/openapi/getDevicePointsDayMonthYearDataList>

6.3. Parâmetros de entrada

Nome	Tipo	Exemplo	Descrição	Obrigatório?
data_point	String	p13134, p13112	Pontos de medição desejados: Formato p+ ponto de medição, para múltiplos pontos, separar usando vírgula.	Sim
data_type	String	2	1: Média; 2: Pico; 3: Vale; 4: Total (apenas dados mensais e anuais têm um valor "total"). Separe vários valores com vírgulas. Se os dados	Sim

			puderem ser agregados, então: data_type é 2 ao consultar dados diários e 4 ao consultar dados mensais ou anuais.	
start_time	String	20250930	Quando query_type é "dia", o formato da data é aaaaMMdd. Quando query_type é "mês", o formato da data é aaaaMM. Quando query_type é "ano", o formato da data é aaaa.	Sim
end_time	String	20251030	Quando query_type é "dia", o formato da data é aaaaMMdd. Quando query_type é "mês", o formato da data é aaaaMM. Quando query_type é "ano", o formato da data é aaaa.	Sim
order	Int	0	Ordem dos dados: 0: Ordem cronológica 1: Invertido	Sim
query_type	String	1	1: Diário; 2: Mensal; 3: Anual	Sim
ps_key_list	List	["972018_11_0_0", "971822_11_0_0"]	Ps_key dos respectivos dispositivos. * Aceita apenas dispositivos do mesmo tipo	Sim

* Ps_key pode ser encontrada na lista de dispositivos da usina /getDeviceList.

6.4. Parâmetros de saída

Nome	Tipo	Exemplo	Descrição
req_serial_num	String	202212236d134e3 386b3c49b66c9 66c5	Número de identificação da requisição
result_code	String	1	Código de erro

result_data	Map	-	Conteúdo da resposta
result_data.{ps_key}	Map	-	Ps_key do dispositivo
result_data.{ps_key}.{p+point_id}	List	-	Ponto de medição solicitado
result_data.{ps_key}.{p+point_id}.{data_type}	String	371600	Valor
result_data.{ps_key}.{p+point_id}.time_stamp	String	20240820	Data do valor

6.5. Exemplo

6.5.1. Requisição

```
url="https://gateway.isolarcloud.com.hk/openapi/getDevicePointsDayMonthYearDataList"
```

```
payload = json.dumps({
    "token": "*****",
    "appkey": "*****",
    "data_type": 2,
    "start_time": 20250926,
    "end_time": 20251003,
    "order": 0,
    "query_type": 1,
    "ps_key_list": ["1*****_45_2_1"],
    "data_point": "p59546,p59547" #Carga e Descarga diária
})
headers = {
    'Content-Type': 'application/json',
    'sys_code': '901',
    'lang': '_en_US',
    'x-access-key': '*****'
}

response = requests.request("POST", url, headers=headers,
data=payload)

print(response.text)
```

6.5.2. Resposta

```
{
  "req_serial_num": "*****",
  "result_code": "1",
  "result_msg": "success",
  "result_data": {
    "1*****_45_2_1": {
      "559547": [
        {
          "2": "0.0000",
          "time_stamp": "20250926"
        },
        {
          "2": "0.0000",
          "time_stamp": "20250927"
        },
        {
          "2": "0.0000",
          "time_stamp": "20250928"
        },
        {
          "2": "0.0",
          "time_stamp": "20250929"
        },
        {
          "2": "0.0000",
          "time_stamp": "20250930"
        },
        {
          "2": "0.0000",
          "time_stamp": "20251001"
        },
        {
          "2": "0.0000",
          "time_stamp": "20251002"
        },
        {
          "2": "0.0000",
          "time_stamp": "20251003"
        }
      ],
      "559546": [
        {
          "2": "0.0000",
          "time_stamp": "20250926"
        }
      ]
    }
  }
}
```

```
    },  
    {  
      "2": "0.0000",  
      "time_stamp": "20250927"  
    },  
    {  
      "2": "0.0000",  
      "time_stamp": "20250928"  
    },  
    {  
      "2": "0.0",  
      "time_stamp": "20250929"  
    },  
    {  
      "2": "0.0000",  
      "time_stamp": "20250930"  
    },  
    {  
      "2": "0.0000",  
      "time_stamp": "20251001"  
    },  
    {  
      "2": "0.0000",  
      "time_stamp": "20251002"  
    },  
    {  
      "2": "0.0000",  
      "time_stamp": "20251003"  
    }  
  ]  
}  
}
```

7. Endereço dos dispositivos

Device_type	Nome
11	Usina
1	Inversor
55	Microinversor
7	Medidor
9	Data Logger

Para mais dispositivos, consultar em: [Sungrow Developer Portal](#)

8. Pontos de medição

8.1.1. Usina

Ponto	Nome	Unidade
83022	Rendimento diário	Wh
83024	Rendimento total	Wh
83033	Potência ativa	W
83002	Potência ativa (Inversor)	W
83004	Rendimento total (Inversor)	Wh
83009	Rendimento diário (Inversor)	Wh
83046	Potência ativa (PCS)	W
83052	Potência da carga	W
83118	Consumo diário da carga	Wh
83124	Consumo total da carga	Wh

83072	Energia exportada diária	Wh
83075	Energia exportada total	Wh
83129	SOC da bateria	%
83549	Potência ativa (Rede)	W

Para mais pontos, consultar em: [Sungrow Developer Portal](#)

8.1.2. Inversor

Ponto	Nome	Unidade
1	Rendimento diário	Wh
2	Rendimento total	Wh
24	Potência ativa	W
14	Potência CC	W
18	Tensão na fase A	V
19	Tensão na fase B	V
20	Tensão na fase C	V
21	Corrente na fase A	A
22	Corrente na fase B	A
23	Corrente na fase C	A
29	Status de operação	-

8.1.3. Microinversor

Ponto	Nome	Unidade
51301	Status de Operação	-
51302	Rendimento total	Wh
51303	Potência ativa	W
51305	Potência CC	W
51309	Tensão CA	V
51312	Corrente CA	A

Para pontos de medição de outros dispositivos, verifique em [Sungrow Developer Portal](#).

9. Definição de erros

Código	Nome	Descrição
1	success	Requisição bem-sucedida
-1	error	Problemas no serviço interno
001	er_missing_parameter:appkey	Necessário incluir o Appkey no body da requisição
002	er_missing_parameter:token	Necessário incluir o Token no body da requisição
003	er_missing_parameter:sys_code	Necessário incluir sys_code no Headers da requisição
E00000	er_invalid_appkey	Appkey inválida, verifique se o valor é o mesmo apresentado na aplicação dentro do Developer Portal, ou até mesmo se não tem nenhum espaço em branco no início ou fim da chave pois pode ser identificado como um caractere
E00002	er_parameter_decrypt_error	Erro no parâmetro requisitado, verifique se o valor está correto e é compatível com o tipo de dispositivo
E00003	er_token_login_invalid	Token inválido ou expirado, faça login novamente para gerar um

		novo token.
E998	er_month_call_api_times_upper_limit	Limite de requisições por mês atingido
E999	er_hour_call_api_times_upper_limit	Limite de requisições por hora atingido
009	er_missing_parameter	Necessário adicionar os pontos de medição
010	er_parameter_value_invalid	Ponto de medição inválido, verifique se ele está correto e se é compatível com o dispositivo
E900	Unauthorized access	Acesso não autorizado, verifique se todas as credenciais estão preenchidas corretamente
E901	Call too frequently	Frequência de requisições muito alta. Sugestão: adicione um intervalo de tempo as requisições para diminuir a frequência e evitar esse problema
E911	Missing parameter in request header: x-access-key	Necessário incluir o x-access-key no Headers da requisição
E912	Illegal x-access-key	X-access-key inválida, verifique se o valor é o mesmo apresentado na aplicação dentro do Developer Portal, ou até mesmo se não tem nenhum espaço em branco no início ou fim da chave pois pode ser identificado como um caractere
E914	Mismatched appkey and x-access-key	Chaves de acesso incompatíveis, verifique na sua aplicação do Developer Portal se as chaves preenchidas estão corretas
E916	Login too frequently	Muitas tentativas de login não sucedidas

10. Exemplo de Aplicação

10.1. Requisição

Exemplo feito em linguagem Python, efetua o login e solicita pontos de medição que, posteriormente podem ser salvos em json, excel ou até mesmo utilizados em um dashboard em tempo real.

```
import requests
import json
import time
import os
from datetime import datetime
from tqdm import tqdm
import pandas as pd

timestamp_str = datetime.now().strftime("%d-%m-%Y_%H-%M")
output_file = f"real_time_data{timestamp_str}.json"

# CHECKPOINT
checkpoint_file = "batch_checkpoint.txt"

# INÍCIO DO TIMER
start_time = time.time()

# FUNÇÕES DE CHECKPOINT
def ler_checkpoint():
    if os.path.exists(checkpoint_file):
        with open(checkpoint_file, "r") as f:
            return int(f.read().strip())
    return 0

def salvar_checkpoint(batch_index):
    with open(checkpoint_file, "w") as f:
        f.write(str(batch_index))

# LOGIN PARA OBTER TOKEN
print("Efetuando login para obter token...")

accesskey = *****
appkey = *****
user = *****
password = *****

login_url = "https://gateway.isolarcloud.com.hk/openapi/login"
login_payload = json.dumps({
    "appkey": appkey,
    "user_account": user,
    "user_password": password
})
headers = {
    'Content-Type': 'application/json',
    'sys_code': '901',
```

```
'lang': '_en_US',
  'x-access-key': accesskey
}

token = None
tentativas = 0
max_tentativas = 3

while not token and tentativas < max_tentativas:
    try:
        response = requests.post(login_url, headers=headers,
data=login_payload)
        response.raise_for_status()
        login_result = response.json()
        token = login_result.get("result_data", {}).get("token")

        if token:
            print("Login bem-sucedido")
        else:
            raise Exception("Token não retornado. Verifique
credenciais.")

    except Exception as e:
        tentativas += 1
        print("Erro ao obter token (tentativa {tentativas}):
{e}")
        time.sleep(60)

if not token:
    print ("Falha ao obter o token após múltiplas tentativas.
Encerrando...")
    exit(1)

# CONSULTAR DADOS EM TEMPO REAL
print("Consultando dados em tempo real...")

url_power =
"https://gateway.isolarcloud.com.hk/openapi/getDeviceRealTimeDat
a"
headers = {
    'Content-Type': 'application/json',
    'sys_code': '901',
    'lang': '_en_US',
    'x-access-key': accesskey
}

# Defina aqui todos os tipos de dispositivos que você quer
```



```
{
    "name": "Meter",
    "point_id_list": {
        "8000": "Tensão Fase A",
        "8001": "Tensão Fase B",
        "8002": "Tensão Fase C",
        "8018": "Potência Ativa Rede",
        "8022": "Potência Reativa Rede"
    },
    "ps_key": ["*****"],
    "device_type": 7
}
]

# Faz a requisição para cada tipo de dispositivo
records = []

for config in tqdm(device_configs, desc="Consultando dispositivos", unit="tipo"):

    point_ids = [str(pid) for pid in config["point_id_list"].keys()]

    payload = json.dumps({
        "token": token,
        "appkey": appkey,
        "point_id_list": point_ids,
        "ps_key_list": config["ps_key"],
        "device_type": config["device_type"]
    })

    try:
        response = requests.post(url_power, headers=headers, data=payload)
        response.raise_for_status()
        result = response.json()
        device_points = result.get("result_data", {}).get("device_point_list", [])

        for item in device_points:
            device = item.get("device_point", {})

            for point_id, descricao in config["point_id_list"].items():
                key_name = f"p{point_id}"
                value = device.get(key_name)

                records.append({
                    "Dispositivo": config["name"],
```

```
        "Descrição": descricao,
        "Valor": value,
        "Atualizado em":
datetime.now().strftime("%Y-%m-%d %H:%M:%S")

    })

    except Exception as e:
        print(f"✘ Erro ao consultar {config['name']}: {e}")

# CRIA O DATAFRAME FINAL
df = pd.DataFrame(records)

# MOSTRA O RESULTADO
print("Dados coletados:")
print(df)

# SALVA EM JSON (opcional)
df.to_json("dados_realtime.json", orient="records", indent=4,
force_ascii=False)

print("Consulta finalizada e dados salvos!")
```