APsystems OpenAPI User Manual

Version List：

Version	Author	Date	Description
V1.0	Yamin.Luo	2022/09/16	First Document
V1.1	Yamin.Luo	2023/03/24	Edit the token URL Change the expiration time of refresh token
V1.2	Yusheng.huang	2023/10/07	Change the JWT Token authentication method to signature authentication method
V1.3	Yusheng.huang	2023/10/31	Add meter interface
V1.4	Yamin.Luo	2023/11/17	
V1.5	Yamin.Luo	2023/11/17	Add inverter-level data API
V1.6	Yamin.Luo	2024/02/07	Optimize the interface for end user

Chapter

1. Overview

Welcome to APsystems' OpenAPI for developer portal. Anyone can register an API account on the platform after the application to access the system details and system data.

The OpenAPI is a REST API and delivers data in JSON format via HTTPS. It has five categories:

- **System Details API**
- **System-level Data API**
- **ECU-level Data API**
- **Meter-level Data API**
- **Inverter-level Data API**

2. Authenticate and Authorize

2.1 Register an OpenAPI Account

Send an application email to APsystems' support email address with the information below:

- Who you are?
- Why do you want to register an OpenAPI account?
- What to do with the data?

When your application was approved, you will get an email with your App Id and App Secret.

Parameter	Type	Description
App Id	string	This is a unique identify id for each OpenAPI account. It is a **32-bit** string with numbers and letters. It can not be changed when it was signed.
App Secret	string	This is the password to verify the valid OpenAPI account to generate the access token and refresh token. It is a **32-bit** string with numbers and letters. It will have a default value the first time. If you want to change it, email APsystems.

Note: Keep the access information safe and secret, do not disclose it to others.

2.2 Authentication

If you have got your App Id and App Secret, you can access the OpenAPI. And you need to calculate a signature for each request.

2.2.1 Headers: Fixed request header information

You need to put the common parameters into the headers of each API request.

Parameter	Required	Type	Description
X-CA-AppId	Y	string	The identity id of the OpenAPI account (App Id).
X-CA-Timestamp	Y	string	The timestamp you request the API.
X-CA-Nonce	Y	string	Uuid, a **32-bit** string, such as: "***5e36eab8295911ee90751eff13c2920b***"
X-CA-Signature-Method	Y	string	Algorithm "***HmacSHA256***" or "***HmacSHA1***"
X-CA-Signature	Y	string	The signature to verify your request.

2.2.2 Calculate the signature

- **Step 1: Get the parameters from the API request**

  Get the parameters below:

  HTTPMethod (GET, POST, DELETE)
  Headers (X-CA-AppId, X-CA-Timestamp, X-CA-Nonce, X-CA-Signature-Method)
  RequestPath (The last name of the path)

- **Step 2: Combine the parameters into one String**

  Combine the parameters with the orders below:

  stringToSign *= X-CA-Timestamp + "/" + X-CA-Nonce + "/" + X-CA-AppId + "/" + RequestPath + "/" + HTTPMethod + "/" + X-CA-Signature-Method*

- **Step 3: Calculate the signature with the algorithm**

Currently, we have support two algorithms to sign the parameters. You can choose one of them to calculate the signature.

- HmacSHA256

- *HmacSHA1*

Sign the ***stringToSign*** with the ***APP Secret***:

HmacSHA256

```
Mac hmacSha256 = Mac.getInstance("HmacSHA256");
byte[] appSecretBytes = appSecret.getBytes(Charset.forName("UTF-8"));
hmacSha256.init(new SecretKeySpec(appSecretBytes, 0, appSecretBytes.length, "HmacSHA256"));
byte[] md5Result = hmacSha256.doFinal(stringToSign.getBytes(Charset.forName("UTF-8")));
String signature = Base64.getEncoder().encodeToString(md5Result);
```

HmacSHA1

```
Mac hmacSha1 = Mac.getInstance("HmacSHA1");
byte[] appSecretBytes = appSecret.getBytes(Charset.forName("UTF-8"));
hmacSha1.init(new SecretKeySpec(appSecretBytes, 0, appSecretBytes.length, "HmacSHA1"));
byte[] md5Result = hmacSha1.doFinal(stringToSign.getBytes(Charset.forName("UTF-8")));
String signature = Base64.getEncoder().encodeToString(md5Result);
```

2.3 Authorization

You can access the default data as long as you get your AppId and App Secret. In addition, you can choose the category corresponding to your business.

Note: According to the access count and data range, it will cause different payments.

2.4 Base Url

The base url is “https://api.apsystemsema.com:9282”.

3. API

3.1 System Details API

3.1.1 Get Details for a Particular System

- **URL**

/user/api/v2/systems/details/{sid}

- **Method**

GET

- **Description**

This request will return the details of the system which you searched for.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.

- **Response**

Code	Example	Model Description

0	{
	"data": {
	"sid": "AZ12649A3DFF",
	"create_date": "2022-09-01",
	"capacity": "1.28",
	"type": 1,
	"timezone": "Asia/Shanghai",
	"ecu":["203000001234"]
	},
	"code": 0
	}
		- **data**(*object*)
		* **sid**(*string*)
		Unique identity id of the system.
		* **create_date**(*string,yyyy-MM-dd*)
		Register date of the system in EMA.
		* **capacity**(*string*)
		System size. Default unit is kW.
		* **type**(*int*)
		System type. Default=1.
		1, PV system.
		2, Storage system.
		3, PV & storage system.
		* **timezone**(*string*)
		The timezone to the ECU belongs
		* **ecu**(*list*)
		ECU ids are registered in this system.
		- **code**(*int*)
		Response code. Refer to 4.1 Annex 1.

		Response Code Definition.

3.1.2 Get Inverters for a Particular System

- **URL**

/user/api/v2/systems/inverters/{sid}

- **Method**

GET

- **Description**

This request will return all the devices of a system you searched for.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.

- **Response**

Code	Example	Model Description

0	{
	"data": [{
	"eid": "203000001234",
	"type": 0,
	"timezone": "Asia/Shanghai",
	"inverter": [{
	"uid": "902000001234",
	"type": "QT2D"
	},{
	"uid": "902000001235",
	"type": "QT2D"
	}]
	}],
	"code": 0
	}
		- **data**(*list*)
		List of the devices sorted by ECU
		* **eid**(*string*)
		Unique identity id of the ECU.
		* **type**(*int*)
		Type of the ECU. Default=0
		0, ECU
		1, ECU with meter activated
		2, ECU with storage activated
		* **timezone**(*string*)
		The timezone to the ECU belongs
		* **inverter**(*list*)
		List of the inverters connected to the ECU.
		✧ **uid**(*string*)
		Unique identity id of the inverter.
		✧ **type**(*string*)
		Type of the inverter.

		* **model**(*string*) Model of the storage-activated ECU. * **capacity**(*string*) The capacity of the storage-activated ECU. Default unit is kWh. - **code**(*int*) Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.1.3 Get Meters for a Particular System

- **URL**

/user/api/v2/systems/meters/{sid}

- **Method**

GET

- **Description**

This request will return all the meters of a system you searched for.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.

- **Response**

Code	Example	Model Description

0	{
	"data": ["203000001234"],
	"code": 0
	}
		- **data**(*list*)
		List of the meters
		- **code**(*int*)
		Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.2 System-level Data API

3.2.1 Get Summary Energy for a Particular System

- **URL**

/user/api/v2/systems/summary/{sid}

- **Method**

GET

- **Description**

This request will return the accumulative energy reported by inverters of a particular system that you searched for.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.

- **Response**

Code	Example	Model Description

0	{
	"data": {
	"today": "12.28",
	"month": "12.28",
	"year": "12.28",
	"lifetime": "12.28"
	},
	"code": 0
	}
		- **data**(*object*)
		* **today**(*string*)
		Accumulative energy was reported by inverters on today. Unit is kWh.
		* **month**(*string*)
		Accumulative energy was reported by inverters this month. Unit is kWh.
		* **year**(*string*)
		Accumulative energy was reported by inverters this year. Unit is kWh.
		* **lifetime**(*string*)
		Accumulative energy reported by inverters in the lifetime of the system. Unit is kWh.
		- **code**(*int*)
		Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.2.2 Get Energy in Period for a Particular System

- **URL**

/user/api/v2/systems/energy/{sid}

- **Method**

GET

- **Description**

This request will return four levels of accumulative energy reported by inverters for a particular system according to the parameters.

- **Hourly Energy:** Return hourly energy in a day. The length is 24 by default, which shows the energy calculated per hour during 0 – 23.
- **Daily Energy:** Return daily energy in a natural month. The length is equal to the number of days per month.
- **Monthly Energy:** Return monthly energy in a natural year. The length is 12 by default.
- **Yearly Energy:** Return yearly energy in a lifetime. The length is equal to the years since the installation.

Set the “energy_level” to “hourly”, “daily”, “monthly” or “yearly” to get the corresponding accumulative energy. The format of the “date_range” will change according to the “energy_level”, it is a field to limit the date to calculate the accumulative energy.

Please set the parameters available to make sure the request is responded to as expected. If the “data_rage” is later than the current time, it will be rejected.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.
energy_level	Y	string	The energy level to calculate the accumulative energy. Available values are “hourly”, “daily”, “monthly”, and “yearly”.
date_range	N	string	The data range to calculate the accumulative energy. The format needs to change according to the value of “energy_level”. - When “hourly”, the format is “yyyy-MM-dd”. - When “daily”, the format is “yyyy-MM”. - When “monthly”, the format is “yyyy”. - When “yearly”, this field is not required.

- **Response**

Code	Example	Model Description

0	{
	"data": ["567.23", "550.32", "320.12"],
	"code": 0
	}
		- **data(*list*)**
		Energy list. The length is stable in each mode. Unit is kWh.
		* 24, when querying “hourly”.
		* The number of days per month when querying “daily”.
		* 12, when querying “monthly”.
		* The number of years since installation when querying “yearly”.
		- **code(*int*)**
		Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.3 ECU-level Data API

3.3.1 Get Summary Energy for a Particular ECU

- **URL**

/user/api/v2/systems/{sid}/devices/ecu/summary/{eid}

- **Method**

GET

- **Description**

This request will return the accumulative energy reported by inverters below an ECU that you searched for.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The identity id of the system.
eid	Y	string	The identity id of ECU.

- **Response**

Code	Example	Model Description

0	{
	"data": {
	"today": "12.28",
	"month": "12.28",
	"year": "12.28",
	"lifetime": "12.28"
	},
	"code": 0
	}
		- **data**(*object*)
		* **today**(*string*)
		Accumulative energy was reported by inverters on today. Unit is kWh.
		* **month**(*string*)
		Accumulative energy was reported by inverters this month. Unit is kWh.
		* **year**(*string*)
		Accumulative energy was reported by inverters this year. Unit is kWh.
		* **lifetime**(*string*)
		Accumulative energy reported by inverters in the lifetime of the system. Unit is kWh.
		- **code**(*int*)
		Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.3.2 Get Energy in Period for a Particular ECU

- **URL**

/user/api/v2/systems/{sid}/devices/ecu/energy/{eid}

- **Method**

GET

- **Description**

This request will return four levels of accumulative energy reported by inverters below a particular ECU according to the parameters.

- **Power Telemetry:** Return the power telemetry in a day.
- **Hourly Energy:** Return hourly energy in a day. The length is 24 by default, which shows the energy calculated per hour during 0 – 23.
- **Daily Energy:** Return daily energy in a natural month. The length is equal to the number of days per month.
- **Monthly Energy:** Return monthly energy in a natural year. The length is 12 by default.
- **Yearly Energy:** Return yearly energy in a lifetime. The length is equal to the years since the installation.

Set the “energy_level” to “hourly”, “daily”, “monthly” or “yearly” to get the corresponding accumulative energy. The format of the “date_range” will change according to the “energy_level”, it is a field to limit the date to calculate the accumulative energy.

Please set the parameters available to make sure the request is responded to as expected. If the “date_rage” is later than the current time, it will be rejected.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.
eid	Y	string	The identity id of ECU.
energy_level	Y	string	The energy level to calculate the accumulative energy. Available values are “minutely”, “hourly”, “daily”, “monthly”, and “yearly”.
date_range	N	string	The data range to calculate the accumulative energy. The format needs to change according to the value of “energy_level”. - When “minutely”, the format is “yyyy-MM-dd”. - When “hourly”, the format is “yyyy-MM-dd”.

			- When “daily”, the format is “yyyy-MM”. - When “monthly”, the format is “yyyy”. - When “yearly”, this field is not required.

- **Response**

Code	Example	Model Description

0	{
	"data": ["567.23", "550.32", "320.12"],
	"code": 0
	}
		- **data(*list*)**
		When choosing “hourly”, “daily”, “monthly”, and “yearly”. Return energy list. The length is stable in each mode. Unit is kWh.
		* 24, when querying “hourly”.
		* The number of days per month when querying “daily”.
		* 12, when querying “monthly”.
		* The number of years since installation when the query “yearly”.
		- **data(*object*)**
		When choosing “minutely”. Return energy object. The length is not stable.
		* **time(*list*)**
		Time list, each point is in a string format HH:mm
		* **energy(*list*)**
		Energy list, each point is in a string format, corresponding to the time. Unit is kWh
		* **power(*list*)**
		Power list, each point is in a string format, corresponding to the time. Unit is W
		* **today(*string*)**
		Accumulative energy produced on the day. Unit is kWh
		- **code(*int*)**
		Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.4 Meter-level Data API

3.4.1 Get Summary Energy for a Particular Meter

- **URL**

/user/api/v2/systems/{sid}/devices/meter/summary/{eid}

- **Method**

GET

- **Description**

This request will return the accumulative energy reported by an Meter ECU that you searched for.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The identity id of the system.
eid	Y	string	The identity id of Meter ECU.

- **Response**

Code	Example	Model Description

0	{
	"code": 0,
	"data":
	{
	"today": {
	"consumed": "394.408090",
	"exported": "0.000000",
	"imported": "560.523540",
	"produced": "833.884550"
	},
	"month": {
	"consumed": "394.408090",
	"exported": "0.000000",
	"imported": "560.523540",
	"produced": "833.884550"
	},
	"year": {
	"consumed": "6394.408090",
	"exported": "0.000000",
	"imported": "4560.523540",
	"produced": "1833.884550"
		- **data**(*object*)
		Meter Summary energy Info.
		* **today**(*map*)
		Today's energy.
		* **month**(*map*)
		Energy of the month.
		* **year**(*map*)
		Energy of the year.
		* **lifetime**(*map*)
		Lifetime energy.
		- **code**(*int*)
		Response code. Refer to 4.1 Annex 1. Response Code Definition.

```
            },
            "lifetime": {
              "consumed": "6394.458090",
              "exported": "0.000000",
              "imported": "4561.643540",
              "produced": "1833.894550"
            }
          }
}
```


3.4.2 Get Energy in Period for a Particular Meter

- **URL**

/user/api/v2/systems/{sid}/devices/meter/period/{eid}

- **Method**

GET

- **Description**

This request will return four levels of accumulative energy reported by inverters below a particular Meter ECU according to the parameters.

- **Power Telemetry:** Return the power telemetry in a day.
- **Hourly Energy:** Return hourly energy in a day. The length is 24 by default, which shows the energy calculated per hour during 0 – 23.
- **Daily Energy:** Return daily energy in a natural month. The length is equal to the number of days per month.
- **Monthly Energy:** Return monthly energy in a natural year. The length is 12 by default.
- **Yearly Energy:** Return yearly energy in a lifetime. The length is equal to the years since the installation.

Set the “energy_level” to “hourly”, “daily”, “monthly” or “yearly” to get the corresponding accumulative energy. The format of the “date_range” will change according to the “energy_level”, it is a field to limit the date to calculate the accumulative energy.

Please set the parameters available to make sure the request is responded to as expected. If the “date_rage” is later than the current time, it will be rejected.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.

eid	Y	string	The identity id of Meter ECU.
energy_level	Y	string	The energy level to calculate the accumulative energy. Available values are “minutely”, “hourly”, “daily”, “monthly”, and “yearly”.
date_range	N	string	The data range to calculate the accumulative energy. The format needs to change according to the value of “level”. - When “minutely”, the format is “yyyy-MM-dd”. - When “hourly”, the format is “yyyy-MM-dd”. - When “daily”, the format is “yyyy-MM”. - When “monthly”, the format is “yyyy”. - When “yearly”, this field is not required.

- **Response**

Code	Example	Model Description

0	{
	"code": 0,
	"data": {
	"time": ["01", "02",...],
	"produced": ["40.300","50.016",...],
	"consumed": ["40.300","50.016",...],
	"imported": ["40.300","50.016",...],
	"exported": ["40.300","50.016",...],
	}
	
	"code": 0,
	"data": {
	"today": {
	"consumed": "5.996600",
	"exported": "0.071860",
	"imported": "3.712280",
	"produced": "2.356180"
	},
	"time": ["23:57"],
	"power": {
	"consumed": ["167.96"],
	"imported_exported": ["167.96"],
		- **data(*list*)**
		When choosing “hourly”, “daily”, “monthly”, and “yearly”. Return energy list. The length is stable in each mode. Unit is kWh.
		* 24, when querying “hourly”.
		* The number of days per month when querying “daily”.
		* 12, when querying “monthly”.
		* The number of years since installation when the query “yearly”.
		- **data(*object*)**
		When choosing “minutely”. Return energy object. The length is not stable.
		* **time(*list*)**
		Time list, each point is in a string format HH:mm
		* **energy(*map*)**
		Energy list, each point is in a string format, corresponding to the time. Unit is kWh

	<pre>"produced": ["0.00"]
	},
	"energy": {
	"consumed": ["0.015620"],
	"exported": ["0"],
	"imported": ["0.01562"],
	"produced": ["0.00000"]
	}
	}
	}</pre>
		* **power(*map*)**
		Power list, each point is in a string format, corresponding to the time. Unit is W
		* **today(*map*)**
		Accumulative energy produced on the day. Unit is kWh
		- **code(*int*)**
		Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.5 Inverter-level Data API

3.5.1 Get Summary Energy for a Particular Inverter

- **URL**

/user/api/v2/systems/{sid}/devices/inverter/summary/{uid}

- **Method**

GET

- **Description**

This request will return the energy of an inverter which you searched for.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.
uid	Y	string	The identity id of inverter.

- **Response**

Code	Example	Model Description

0	<pre>{
	"data": {
	"d1": "12.28",
	"m1": "12.28",
	"y1": "12.28",</pre>
		- **data(*list*)**
		Energy list per channel.
		✧ **d1(*string*)**
		Accumulative energy reported by channel 1 of the inverter today.

```
            "t1": "12.28",
            "d2": "12.28",
            "m2": "12.28",
            "y2": "12.28",
            "t2": "12.28",
            "d3": "12.28",
            "m3": "12.28",
            "y3": "12.28",
            "t3": "12.28",
            "d4": "12.28",
            "m4": "12.28",
            "y4": "12.28",
            "t4": "12.28"
    },
    "code": 0
}
```


- **m1**(*string*)
  Accumulative energy reported by channel 1 of the inverter in this month.
- **y1**(*string*)
  Accumulative energy reported by channel 2 of the inverter in this month.
- **t1**(*string*)
  Accumulative energy reported by channel 3 of the inverter in this month.
- **d2**(*string*)
  Accumulative energy reported by channel 2 of the inverter today.
- **m2**(*string*)
  Accumulative energy reported by channel 1 of the inverter in this month.
- **y2**(*string*)
  Accumulative energy reported by channel 2 of the inverter in this month.
- **t2**(*string*)
  Accumulative energy reported by channel 3 of the inverter in this month.
- **D3**(*string*)
  Accumulative energy reported by channel 3 of the inverter today.
- **m3**(*string*)
  Accumulative energy reported by channel 1 of the inverter in this month.
- **y3**(*string*)
  Accumulative energy reported by channel 2 of the inverter in this month.
- **t3**(*string*)
  Accumulative energy reported by channel 3 of the inverter in this month.
- **d4**(*string*)
  Accumulative energy reported by channel 4 of the inverter today.
- **m4**(*string*)
  Accumulative energy reported by channel 1 of the inverter in this month.
- **y4**(*string*)

		Accumulative energy reported by channel 2 of the inverter in this month. ✧ **t4(*string*)** Accumulative energy reported by channel 3 of the inverter in this month. - **code(*int*)** Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.5.2 Get Energy in Period for a Particular Inverter

- **URL**

/user/api/v2/systems/{sid}/devices/inverter/energy/{uid}

- **Method**

GET

- **Description**

This request will return five levels of accumulative energy below a particular inverter according to the parameters.

- **Power Telemetry:** Return the power telemetry in a day.
- **Hourly Energy:** Return hourly energy in a day. The length is 24 by default, which shows the energy calculated per hour during 0 – 23.
- **Daily Energy:** Return daily energy in a natural month. The length is equal to the number of days per month.
- **Monthly Energy:** Return monthly energy in a natural year. The length is 12 by default.
- **Yearly Energy:** Return yearly energy in a lifetime. The length is equal to the years since the installation.

Set the “energy_level” to “hourly”, “daily”, “monthly” or “yearly” to get the corresponding accumulative energy. The format of the “date_range” will change according to the “energy_level”, it is a field to limit the date to calculate the accumulative energy.

Please set the parameters available to make sure the request is responded to as expected. If the “date_range” is later than the current time, it will be rejected.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.

uid	Y	string	The identity id of inverter.
energy_level	Y	string	The energy level to calculate the accumulative energy. Available values are “minutely”, “hourly”, “daily”, “monthly”, and “yearly”.
date_range	N	string	The data range to calculate the accumulative energy. The format needs to change according to the value of “energy_level”. - When “minutely”, the format is “yyyy-MM-dd”. - When “hourly”, the format is “yyyy-MM-dd”. - When “daily”, the format is “yyyy-MM”. - When “monthly”, the format is “yyyy”. - When “yearly”, this field is not required.

- **Response**

Code	Example	Model Description

0	{
	"data": {
	"e1": ["567.23","550.32","320.12"],
	"e2": ["567.23","550.32","320.12"],
	},
	"code": 0
	}
		- **data(*object*)**
		When choosing “hourly”, “daily”, “monthly”, and “yearly”. Return energy list per channel. The length is stable in each mode. Unit is kWh.
		* **e1(object)**
		Energy list reported by channel 1.
		* **e2(object)**
		Energy list reported by channel 2.
		* **e3(object)**
		Energy list reported by channel 3.
		* **e4(object)**
		Energy list reported by channel 4.
		
		When choosing “minutely”, Return energy list per channel. The length is stable in each mode. Unit is kWh.
		* **t(*list*)**
		Time list, each point is in a string format HH:mm
		* **dc_p1(*list*)**
		DC Power on channel 1.
		* **dc_p2(*list*)**

		DC Power on channel 2. * **dc_p3(*list*)** DC Power on channel 3. * **dc_p4(*list*)** DC Power on channel 4. * **dc_i1(*list*)** DC current on channel 1. * **dc_i2(*list*)** DC current on channel 2. * **dc_i3(*list*)** DC current on channel 3. * **dc_i4(*list*)** DC current on channel 4. * **dc_v1(*list*)** DC voltage on channel 1. * **dc_v2(*list*)** DC voltage on channel 2. * **dc_v3(*list*)** DC voltage on channel 3. * **dc_v4(*list*)** DC voltage on channel 4. * **dc_e1(*list*)** DC energy on channel 1. * **dc_e2(*list*)** DC energy on channel 2. * **dc_e3(*list*)** DC energy on channel 3. * **dc_e4(*list*)** DC energy on channel 4. * **ac_v1 (*list*)** AC voltage on channel 1. * **ac_v2(*list*)** AC voltage on channel 2. * **ac_v3(*list*)** AC voltage on channel 3. * **ac_t(*list*)** AC temperature. * **ac_p(*list*)** AC power. * **ac_f(*list*)**

		AC frequency. - **code(*int*)** Response code. Refer to 4.1 Annex 1. Response Code Definition.

3.5.3 Get Energy in a Day for all inverters below a Particular ECU

- **URL**

/user/api/v2/systems/{sid}/devices/inverter/batch/energy/{eid}

- **Method**

GET

- **Description**

This request will return five levels of accumulative energy below a particular inverter according to the parameters.

- **Power Telemetry:** Return the power telemetry in a day.
- **Day Energy:** Return the total energy in a day.

Set the “energy_level” to “power” or “energy” to get the corresponding energy. The format of the “date_range” will change according to the “energy_level”, it is a field to limit the date to calculate the accumulative energy.

Please set the parameters available to make sure the request is responded to as expected. If the “date_range” is later than the current time, it will be rejected.

- **Parameters**

Parameter	Required	Type	Description
sid	Y	string	The unique identity id of the system.
eid	Y	string	The identity id of ECU.
energy_level	Y	string	The energy level to calculate the accumulative energy. Available values are “power”, “energy”.
date_range	Y	string	- The data to query. The format is “yyyy-MM-dd”.

- **Response**

Code	Example	Model Description
0	{	- **data(*object*)**

	<pre>"data": ,
	"code": 0
	}</pre>
		When choosing “energy”. Return energy list per channel. The length is stable in each mode. Unit is kWh.
		* **energy(*list*)**
		Energy list. The string format in the list is “uid-channel-energy”.
		For example, 701000001234-1-1.24
		
		When choosing “power”, Return power list per channel. Unit is W.
		* **time(*list*)**
		Time list, each point is in a string format HH:mm
		AC voltage on channel 1.
		* **power(*map*)**
		Power list, each point is in a string format, corresponding to the time.
		For example, {701000001234-1: [45,56,78,98]}
		The power telemetry length is the same as the length of time.
		
		- **code(*int*)**
		Response code. Refer to 4.1 Annex 1. Response Code Definition.

4. Annex

4.1 Annex 1. Response Code Definition

Code	Description
0	Succeed to request.
1000	Data exception.
1001	No data.
2000	Application account exception.
2001	Invalid application account.
2002	The application account is not authorized.

2003	Application account authorization expires.
2004	The application account has no permission.
2005	The access limit of the application account was exceeded.
3000	Access token exception.
3001	Missing Access token.
3002	Unable to verify Access token.
3003	Access token timeout.
3004	Refresh token timeout.
4000	Request parameter exception.
4001	Invalid request parameter.
5000	Internal server exception.
6000	Communication exception.
7000	Server access restriction exception.
7001	Server access limit exceeded.
7002	Too many requests, please request later.
7003	The system is busy, please request later.