HUB public API manual

v1.4.0

embion

Table of contents

1 Introduction

This document discribes the public customer API available for the HUB portal. This API can be used to read plant, inverter or meter data and control the plant. Using the API allows users of the Hub portal to read and process the device data in another platform or device.

1.1 Use of the API

To use the API a token needs to be generated per device. This token determines the permissions and the maximum number of requests per day. By default, a device can replay to 250 API calls per day. If more than 250 API calls are done, within one day, an error message will be returned. Please contact Embion if more than 250 calls per day per device are required.

2 About this manual

2.1 Call outs

Note

Used for notes in this documentation

Warning

Used for warnings in this documentation

Important

Used for important notes in this documentation

Tip

Used for tips in this documentation

Caution

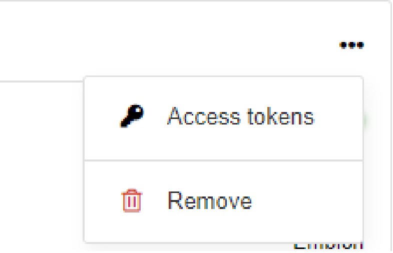
Used for caution notes in this documentation

3 Enable the API acces tokens

For each individual device which needs to be read or controlled via the public API, a token needs to be generated. The token is generated and managed in the HUB portal (`https://api.hub.embion.nl`).

3.1 Generate token in the HUB.

To generate or manage a token, open the specific device menu and open "access tokens".



The menu shows all the currently availible access tokens, and allows the user to create a new one. Per token the expiration date can be selected, and the uid's (meters and inverter) to which te token has read access can be selected.

i Tokens are generated per device

Please note, the tokens are generated per device. When multiple devices in one namespace need to be accesed by the API, a token per device needs to be generated.

! Limited number of API calls per day

The maximum number of API calls per day is limited per device (default to 250), independently of the number of tokens. The user is able to limit the number of calls per day for each token. If the total number of calls per day exceeds the 250 calls, the API will not return data for the current request.

4 API layout

Multiple API requests can be done to retreive plant data, meter data, inverter data or device status data. Below the layout and an explanation per call. Filling in your own id, API-KEY and start_date should return valid data.

! API-KEY

All API calls should include the API-KEY generated in the HUB portal in the header of the API call. For example the following information is added to the header:
API-KEY: wzae211vh4ddXlbwt4wdyX1eSjkcgt7dmpqwd5Xnk8amm

ℹ Date and time format

For all date and time values the ISO 8601 standard is used. For normal use only the date and optionally the time zone needs to be send in the requests.

Some valid date-time examples using the ISO 8601 standard:

ISO 8601 notation	Start time / date	Time zone
2022-12-14T08:00Z	14-12-2022 8:00:00	UTC
2022-12-14T08:00	14-12-2022 8:00:00	UTC
2022-12-14T08Z	14-12-2022 8:00:00	UTC
2022-12-14T08:00:00.000+0100	14-12-2022 8:00:00	GMT+1
2022-12-14	14-12-2022 00:00:00	UTC
2022-12-14GMT+0100	14-12-2022 00:00:00	GMT+1
2022-12	01-12-2022 00:00:00	UTC
2022	01-01-2022 00:00:00	UTC

4.1 Status request

This request returns status information for your device. The api is requested by sending a get command to `https://api.hub.embion.nl/v1/status` containing the following url data:

var	description	mandatory
id	unique id for the device to read (generated in the hub portal)	Yes

The following data is returned in json layout:

var	description	format
status	actual status of the plant	string
online	true if plant is online, false if offline	bool
last_contact	last contact in ISO 8601 layout	string

var	description	format
serial	serial number of device	string
version	actual software version of device	string
pn	product number of device	string
name	reference name of device	string
namespace	namespace location of device	string
status_message	returns the actual status message of the device	string
support_status	returns the actual support status, disabled or support ID when enabled	string
safe_state	true if safe_state is enabled on the device	bool
plant_control	idle, pending, sent, accepted, failed	string

> **ℹ Status API call example**
>
> Request `https://api.hub.embion.nl/v1/status?id=[id]`

Result:

```
{
    "status": "ok",
    "online": true,
    "last_contact": "2022-12-14T12:48:13.000Z",
    "serial": "0100211001090B",
    "version": "1.3.1",
    "pn": "GSE-A010-POE",
    "name": "main-solar",
    "namespace": "Embion",
    "status_message": "reducing inverters",
    "support_status": "A291D88",
    "safe_state": false,
    "plant_control": "idle"
}
```

4.2 Plant data request

Using this request, plant data can be returned. The api is requested by sending a get command to `https://api.hub.embion.nl/v1/plant` containing the following url data:

var	description	mandatory	format
id	unique id for the device to read (generated in the Hub portal)	Yes	string
period	select data return period (q: 15 minute (default), h: hourly, w: weekly, d: daily, m: monthly, y: yearly, l: last sample)	No	string
range	time range to show, d = day (default), w = week, m = month, y = year	No	string
type	type of combination of multiple datapoints (min: min value in timerange, max: max value in timerange (default), avg: average value in timerange)	No	string
start_date	date of the first sample in ISO 8601 layout, if not set current day is used	No	string

ℹ Add time to start_date

Users can optionally add a time to start_date and thereby shifting the day interval. The total number of returned entries will stay identical. When no time is defined, a day is defined between 00:00:00 and 23:59:59 in the selected timezone.

ℹ Default values

If as well start_date as range and period is not defined in the call, only the last stored sample is returned.

💡 Data time range

The user can define the start date, from which the first data point will be returned using ‘start_date’. By defining period, the date return interval is selected. By defining range, the end-date/time relative to ‘start_date’ is selected, and so the number of entries returned.

ℹ Result time limit

Please note for period q and h the maximum range is d (one day). For period d max range is w (one week). For period w max range is m (one month). For period m the max range is y (one year.)

The following data is returned in json layout:

var	description	units	format
timestamp	Timestamp of the measurement	ISO 8601	string
psol	Actual solar power	1 W	Integer
kdy	Cumulative daily yield	1 Wh	Integer
run	# inverters in RUN state	-	Integer
warn	# inverters in WARN state	-	Integer
err	# inverters in ERR state	-	Integer
red	Actual reduction value (10000 == 100% => no reduction) Represents power limit	%	Integer
var1	Free to use variable	-	Integer
var2	Free to use variable	-	Integer
var3	Free to use variable	-	Integer
var4	Free to use variable	-	Integer
in1	State of digital input 1	-	Integer 0 or 1 (bool)
in2	State of digital input 2	-	Integer 0 or 1 (bool)
out1	State of digital output 1	-	Integer 0 or 1 (bool)
out2	State of digital output 2	-	Integer 0 or 1 (bool)
con	# of inverters connected to the gateway	-	Integer
pgrid	gridpower	1 W	Integer
egi	Grid import energy	1 Wh	Integer
ege	Grid export energy	1 Wh	Integer
gil1	Grid phase 1 current	0.1 A	Integer
gil2	Grid phase 2 current	0.1 A	Integer
gil3	Grid phase 3 current	0.1 A	Integer
gul1	Grid phase 1 voltage	0.1 V	Integer
gul2	Grid phase 2 voltage	0.1 V	Integer
gul3	Grid phase 3 voltage	0.1 V	Integer

i Plant API call example

Request `https://api.hub.embion.nl/v1/plant?id=[id]&period=q&range=d&type=max&start_date=2022-12-14`

Result:

```
    {
        "timestamp": "2022-12-14T10:00:00.000Z",
        "con": 3,
        "ege": 3500,
        "egi": 2000,
        "err": 0,
        "in1": 1,
        "in2": 0,
        "out1": 0,
        "out2": 0,
        "gil1": 5,
        "gil2": 6,
        "gil3": 7,
        "gul1": 220,
        "gul2": 230,
        "gul3": 240,
        "kdy": 1010,
        "pgrid": 1000,
        "psol": 1750,
        "red": 10000,
        "run": 2,
        "var1": 1,
        "var2": 2,
        "var3": 3,
        "var4": 4,
        "warn": 1
    },
    {
        "timestamp": "2022-12-14T10:15:00.000Z",
        "con": 3,
        "ege": 4100,
        "egi": 2000,
        "err": 0,
        "in1": 1,
        "in2": 0,
        "out1": 0,
        "out2": 0,
        "gil1": 56,
        "gil2": 63,
        "gil3": 78,
        "gul1": 2218,
        "gul2": 2301,
        "gul3": 2368,
        "kdy": 12010,
        "pgrid": -11600,
        "psol": 2000,
        "red": 10000,
        "run": 3,
        "var1": 1,
        "var2": 2,
        "var3": 3,
        "var4": 4,
        "warn": 0
    }
```

4.3 Meter data request

With this API request, individual meter data can be returned. The api is requested by sending a get command to `https://api.hub.embion.nl/v1/meter` containing the following url data:

var	description	mandatory	format
id	unique id for the device to read (generated in the Hub portal)	Yes	string
uid	the uid of the meter to read, only one uid can be entered	Yes	string
period	select data return period (q: 15 minute (default), h: hourly, w: weekly, d: daily, m: monthly, y: yearly, l: last sample)	No	string
range	time range to show, d = day (default), w = week, m = month, y = year	No	string
type	type of combination of multiple datapoints (min: min value in timerange, max: max value in timerange (default), avg: average value in timerange)	No	string
start_date	date of the first sample in ISO 8601 layout, if not set current day is used	No	string

The following data is returned in json layout. Only data present is returned. Data which is not used for the specific meter is left out of the reply.

var	description	units	format
timestamp	Timestamp of the measurement	ISO 8601	string
actpow	Total active power	1 W	Integer
apparpow	Total apparent power	1 VA	Integer
reactpow	Total reactive power	1 VAR	Integer
pf	Total powerfactor	0.01 $cos(\varphi)$	Integer
pfl1	Phase 1 powerfactor	0.01 $cos(\varphi)$	Integer
pfl2	Phase 2 powerfactor	0.01 $cos(\varphi)$	Integer
pfl3	Phase 3 powerfactor	0.01 $cos(\varphi)$	Integer
actpowl1	Phase 1 active power	1 W	Integer
actpowl2	Phase 2 active power	1 W	Integer
actpowl3	Phase 3 active power	1 W	Integer
il1	Phase 1 current	0.1 A	Integer
il2	Phase 2 current	0.1 A	Integer
il3	Phase 3 current	0.1 A	Integer
vll12	Phase1-2 line-line voltage	0.1 V	Integer
vll13	Phase1-3 line-line voltage	0.1 V	Integer
vll23	Phase2-3 line-line voltage	0.1 V	Integer
vl1	Phase1 to neutral voltage	0.1 V	Integer
vl2	Phase2 to neutral voltage	0.1 V	Integer
vl3	Phase3 to neutral voltage	0.1 V	Integer
eimp	imported energy counter	1 Wh	Integer
eexp	exported energy counter	1 Wh	Integer
esolar	used solar energy counter	1 Wh	Integer
egrid	used grid energy counter	1 Wh	Integer
fgrid	Measured grid frequency	0.01 Hz	Integer
thdul1	Phase 1 voltage THD	0.01 %	Integer
thdul2	Phase 2 voltage THD	0.01 %	Integer
thdul3	Phase 3 voltage THD	0.01 %	Integer
thdil1	Phase 1 current THD	0.01 %	Integer
thdil2	Phase 2 current THD	0.01 %	Integer
thdil3	Phase 3 current THD	0.01 %	Integer
gas	Used gas counter	0.01 m3	Integer
water	Used water counter	0.01 m3	Integer
heat	Used heat counter	100 J	Integer
radi	Measured radiation	0.1 W/m2	Integer
temp	Measured temperature	0.1 C	Integer
humi	Measured humidity	0.01 %	Integer
pres	Measured pressure	1000 Pa	Integer
flow	Measured flow	0.01 liter/min	Integer
weight	Measured weigth	1 gram	Integer

Meter API call example

Request `https://api.hub.embion.nl/v1/meter?id=[id]&uid=mtr1:1&period=q&range=d&type=max&start_date=2022-12-14`

Result:

```
{
        "timestamp": "2022-12-14T08:00:00.000Z",
        "actpow": 1000,
        "actpowl1": 100,
        "actpowl2": 1200,
        "actpowl3": -300,
        "apparpow": 1005,
        "eexp": 0,
        "egrid": 13541,
        "eimp": 36578912,
        "esolar": 31575661,
        "fgrid": 5011,
        "gas": 12300,
        "il1": 1000,
        "il2": 2000,
        "il3": 500,
        "pf": 30,
        "pfl1": 50,
        "pfl2": -50,
        "pfl3": 100,
        "reactpow": 100,
        "thdil1": 100,
        "thdil2": 200,
        "thdil3": 140,
        "thdul1": 111,
        "thdul2": 15,
        "thdul3": 109,
        "ul1": 23011,
        "ul2": 24011,
        "ul3": 23544,
        "ull12": 39821,
        "ull13": 40201,
        "ull23": 39098
    },
    {
        "timestamp": "2022-12-14T08:15:00.000Z",
        "actpow": 1000,
        "actpowl1": 100,
        "actpowl2": 1200,
        "actpowl3": -300,
        "apparpow": 1005,
        "eexp": 0,
        "egrid": 13541,
        "eimp": 36578912,
        "esolar": 31575661,
        "fgrid": 5011,
        "gas": 15300,
        "il1": 1000,
        "il2": 2000,
        "il3": 500,
        "pf": 30,
        "pfl1": 50,
        "pfl2": -50,
        "pfl3": 100,
        "reactpow": 100,
        "thdil1": 100,
        "thdil2": 200,
        "thdil3": 140,
        "thdul1": 111,
        "thdul2": 15,
        "thdul3": 109,
        "ul1": 23011,
        "ul2": 24011,
        "ul3": 23544,
        "ull12": 39821,
        "ull13": 40201,
        "ull23": 39098
    }
```

4.4 Inverter data request

This API call returnes individual inverter data. The api is requested by sending a get command to `https://api.hub.embion.nl/v1/inverter` containing the following url data:

var	description	mandatory	format
id	unique id for the device to read (generated in the Hub portal)	Yes	string
uid	the uid of the inverter to read, only one uid can be entered	Yes	string
period	select data return period (q: 15 minute (default), h: hourly, w: weekly, d: daily, m: monthly, y: yearly, l: last sample)	No	string
range	time range to show, d = day (default), w = week, m = month, y = year	No	string
type	type of combination of multiple datapoints (min: min value in timerange, max: max value in timerange (default), avg: average value in timerange)	No	string
start_date	date of the first sample in ISO 8601 layout, if not set current day is used	No	string

The following data is returned in json layout. Dependend on the number of strings stringdata will contain an amount of entries.

var	description	units	format
timestamp	Timestamp of the measurement	ISO 8601	string
stat	Inverter status		Integer
kdy	Inverter daily yield	1 Wh	Integer
pac	Inverter AC power	1 W	Integer
ul1	Inverter phase 1 voltage	0.1 V	Integer
ul2	Inverter phase 2 voltage	0.1 V	Integer
ul3	Inverter phase 3 voltage	0.1 V	Integer
il1	Inverter phase 1 current	0.1 A	Integer
il2	Inverter phase 2 current	0.1 A	Integer
il3	Inverter phase 3 current	0.1 A	Integer
tmp1	Inverter internal temperature 1	0.1 C	Integer
tmp2	Inverter internal temperature 2	0.1 C	Integer
ilk	Inverter leakge current or isolation resistance	0.0001 A	Integer
arc	Inverter arc detection status		Integer
batpow	Battery power (+charge, -discharge)	1 W	Integer
batcap	Remaining battery capacity	1 Wh	Integer
batsoc	Battery State Of Charge	0.1 %	Integer
batsoh	Battery State Of Health	0.1 %	Integer
battemp	Battery temperature	0.1 C	Integer

var	description	units	format
string_data	Individual string data	**stringdata**	

Defenition of the **stringdata**:

var	description	units	format
sid	string number of inverter uid		string
idc	String current	0.1 A	Integer
udc	String voltage	0.1 V	Integer
pdc	String power	1 W	Integer
ydc	String daily yield	1 Wh	Integer
sarc	String arc detection status		Integer

ℹ Inverter API call example

Request `https://api.hub.embion.nl/v1/inverter?id=[id]&uid=inv1:1 &period=h&range=d&type=max&start_date=2022-12-14GMT+0100`

Result:

```
{
    "timestamp": "2022-12-13T23:00:00.000Z",
    "arc": 0,
    "batcap": 0,
    "batpower": 0,
    "batsoc": 0,
    "batsoh": 0,
    "battemp": 0,
    "il1": 56,
    "il2": 63,
    "il3": 77,
    "ilk": 3,
    "kdy": 1100000,
    "pac": 10000,
    "string_data": [
        {
            "sid": "1",
            "idc": 50,
            "udc": 5000,
            "pdc": 2500,
            "sarc": 0
        },
        {
            "sid": "2",
            "idc": 60,
            "udc": 6000,
            "pdc": 3600,
            "sarc": 0
        },
    ],
    "stat": 1,
    "tmp1": 531,
    "tmp2": 366,
    "ul1": 2301,
    "ul2": 2405,
    "ul3": 2508
},
{
    "timestamp": "2022-12-14T00:00:00.000Z",
    "arc": 0,
    "batcap": 0,
    "batpower": 0,
    "batsoc": 0,
    "batsoh": 0,
    "battemp": 0,
    "il1": 120,
    "il2": 130,
    "il3": 120,
    "ilk": 3,
    "kdy": 1200000,
    "pac": 14000,
    "string_data": [
        {
            "sid": "1",
            "idc": 50,
            "udc": 5000,
            "pdc": 2500,
            "sarc": 0
        },
        {
            "sid": "2",
            "idc": 60,
            "udc": 6000,
            "pdc": 3600,
            "sarc": 0
        },
    ],
    "stat": 1,
    "tmp1": 551,
    "tmp2": 346,
    "ul1": 2301,
    "ul2": 2405,
    "ul3": 2508
},
```

4.5 Plant control example

This API call allows external control of the plant. The GSE will limit any given values to the plant maximum or minimum allowed values.

Giving values exceeding the plant capabilities is allowed, but GSE will adjust to allowed values.

At least one of the control values should be given, when a certain control value is not given or the valid_time is exceeded, the control value is not activly controlled by the GSE.

The api command is given by sending a get command to `https://api.hub.embion.nl/v1/plantcontrol` containing the following POST data

var	description	mandatory	format	unit
id	unique id for the device to read (generated in the Hub portal)	Yes	string	
p_import_limit	Maximum grid import power	No	integer	W
p_export_limit	Maximum grid export power	No	integer	W
control_generation	min: minimise generation, max: maximize generation, nom: nominal generation	No	string	
control_consumption	min: minimise consumption, max: maximise consumption, nom: nominal consumption	No	string	
valid_time	Time which given command stays active on the GSE. Default and '0' is infinit	No	uinteger	s

When a plant control action was still active when sending a new command, the old command is overwritten and the return message is “overwritten”.

> ⚠ Warning
>
> Keep in mind that previously set limits are not remembered when a new command is send and overwrites the previous command.

4.5.1 Further explanation for control generation and consumption

The “control_generation” and “control_consumption” items can be used to control plant generation and consumption independently of the plant configuration.

Setting “control_generation” to “min” reduces the power generation to the minimum, resulting is solar power converters to shutdown and wind turbines to stop.

Setting “control_generation” to “nom” allows generation of solar and wind to operate normally.

Setting “control_generation” to “max” allows also the start of any extra generators (if available at plant).

Settings “control_consumption” to “min” reduces the controlable loads like heatpumps and EV-chargers to minimum consumption.

Settings “control_consumption” to “nom” enables normal controlable loads to operate within the plant limits.

Settings “control_consumption” to “max” increases the power for controlable loads to maximum. EV-chargers will increase charging power to maximum (within plant limits) and heatpumps will increase or decrease setpoint to increase power consumption.