

**HUB public API manual**



***ambion***

# Table of contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>About this manual</b>	<b>4</b>
2.1	Callouts . . . . .	4
<b>3</b>	<b>Getting started</b>	<b>5</b>
3.1	Access tokens . . . . .	5
3.1.1	Token creation . . . . .	5
3.1.2	Request limits . . . . .	5
3.1.3	Use of access token in the API . . . . .	6
3.1.3.1	Header information . . . . .	6
3.2	HTTP status codes . . . . .	6
3.3	ISO 8601 . . . . .	7
3.4	Gzip . . . . .	7
<b>4</b>	<b>API endpoints</b>	<b>9</b>
4.1	Get status endpoint . . . . .	9
4.1.1	Request headers . . . . .	9
4.1.2	Request query parameters . . . . .	9
4.1.3	Response body . . . . .	10
4.1.4	Request example . . . . .	11
4.2	Get UIDs endpoint . . . . .	12
4.2.1	Request headers . . . . .	12
4.2.2	Request query parameters . . . . .	12
4.2.3	Response body . . . . .	12
4.2.4	Request example . . . . .	13
4.3	Get plant data endpoint . . . . .	14
4.3.1	Request headers . . . . .	14
4.3.2	Request query parameters . . . . .	14
4.3.3	Response body . . . . .	15
4.3.4	Request example . . . . .	17
4.4	Get meter data endpoint . . . . .	19
4.4.1	Request headers . . . . .	19
4.4.2	Request query parameters . . . . .	19
4.4.3	Response body . . . . .	20
4.5	Request example . . . . .	22
4.6	Get inverter data endpoint . . . . .	24
4.6.1	Request headers . . . . .	24
4.6.2	Request query parameters . . . . .	24
4.6.3	Response body . . . . .	25

---


4.6.4	Request example . . . . .	27
4.7	Plant control endpoint . . . . .	29
4.7.1	Request headers . . . . .	29
4.7.2	Request body . . . . .	30
4.7.2.1	Control explanation . . . . .	31
4.7.2.2	Example request body . . . . .	31
4.7.3	Response body . . . . .	32
4.7.3.1	Example response body . . . . .	32

# 1 Introduction


This document describes the public customer REST API available for the HUB portal. The API can be used to read plant, inverter and/or meter data and to control the plant, allowing users of the HUB portal to use these components in other platforms.

## 2 About this manual


### 2.1 Callouts

 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 Getting started

### 3.1 Access tokens

All endpoints in this API require authentication using access tokens. These tokens are unique to each device and determine the permissions and the maximum number of requests that can be made per day.

#### 3.1.1 Token creation

A token can be generated for each individual device that needs to be read or controlled via the public API. The token is generated and managed through the [HUB portal](#). When multiple devices in one namespace need to be accessed by the API, a token per device needs to be generated.

##### Tip

Documentation for creating access tokens can be found [here](#)

#### 3.1.2 Request limits

There is a default limit of 3,000 requests per device per day, with each request counting towards this limit.

Although the maximum requests per day per token can be adjusted, all requests still contribute to the device's 3,000 request limit.

##### Note

If more than 3,000 calls per device per day are required, please contact [Embion](#) for further assistance and to discuss specific needs.

### 3.1.3 Use of access token in the API

All endpoints described in this document require the access token value to be present as the API-KEY in the HTTP header of the request.

#### Note

Any references to an ID in this document correspond to the token's ID.

#### Hint

The access token is structured as follows: the first 13 characters represent the token ID, followed by an "X," and then a string that completes the actual access token.

- Access token: wzae211vh4ddX1bwt4wdyX1eSjre435dmpqwd5Xnk8amm
- Token ID: wzae211vh4dd

#### 3.1.3.1 Header information

Example header information:

```
API-KEY: wzae211vh4ddX1bwt4wdyX1eSjre435dmpqwd5Xnk8amm
```

If the token is not provided or does not match the ID, a response with HTTP status code 401 will be returned, including the following body:

```
[{ "error": "Invalid ID or token used" }]
```

## 3.2 HTTP status codes

The following HTTP status codes can be present in the responses:

Code	Status	Description
200	OK	The request was successfully processed
401	Unauthorized	The access token wasn't given, is invalid, and/or doesn't match with the given id
403	Forbidden	The access token is valid, but doesn't have the correct permissions for the endpoint
429	To many requests	The maximum daily request limit of the token and/or the device has been reached

### 3.3 ISO 8601

The ISO 8601 standard is used for all date and time values. In most cases, only the date and optionally the time zone needs to be sent in the requests.

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

Notation	Date	Timezone
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

### 3.4 Gzip

Gzip compression is a data optimization technique that reduces the size of the data transferred over the internet.

When a request is made to the public-API, it can compress the response data using Gzip. This means that the data sent from the server to the customer application is smaller in size, leading to faster responses and lower bandwidth usage.

To enable Gzip compression, add the following header to the request:

Var	Description	Mandatory	Format
Accept-Encoding: gzip	Enables Gzip compression for all reply messages	No	String



## 4 API endpoints

To fetch data or control devices, HTTP requests can be sent to the API endpoints described below. Replacing placeholders with the appropriate ID, API-KEY, and start\_date should return valid data.

### 4.1 Get status endpoint

**Method:** GET | **URL:** `https://api.hub.embion.nl/v1/status`

This endpoint returns status information about the given device. The request does not have a body, but it requires a query parameter in order to work properly.

#### 4.1.1 Request headers

This endpoint requires authentication using the API-KEY header (see [Access tokens](#) above).

Var	Description	Mandatory	Format
API-KEY	Actual access token generated in the HUB	Yes	String

#### 4.1.2 Request query parameters

Query parameters must be appended to the URL, starting with a question mark (?). For example: `https://api.hub.embion.nl/v1/status?id=[id]`.

Var	Description	Mandatory
ID	ID of the token (generated in the HUB)	Yes

**4.1.3 Response body**

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
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
epex_configured	True if the device has energy price control rules (defined in the energy pricing app), false if not	bool
token	token data (see table below)	<b>token</b>

**Definition of token:**

var	description	format
control_allowed	true if control access is enabled for the given token	bool
expire_date	expiration date of the given token if set	string
request_limit	maximum daily requests for the token when configured, otherwise the configured maximum device requests (default 3000).	Integer
requests_today_token	number of requests done for the token today	Integer

#### 4.1.4 Request example

##### 💡 Example URL

```
https://api.hub.embion.nl/v1/status?id=[id]
```

```
{
  "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",
  "epex_configured": false,
  "token": {
    "control_allowed": true,
    "expire_date": "2024-09-30T00:00:00.000Z",
    "request_limit": 3000,
    "requests_today_token": 40
  }
}
```

## 4.2 Get UIDs endpoint

**Method:** GET | **URL:** `https://api.hub.embion.nl/v1/uids`

This endpoint returns all UIDs available for the given token. The result can include either all UIDs or a subset of UIDs, depending on the token's configuration.

### 4.2.1 Request headers

This endpoint requires authentication using the API-KEY header (see [Access tokens](#) above).

Var	Description	Mandatory	Format
API-KEY	Actual access token generated in the HUB	Yes	String

### 4.2.2 Request query parameters

Query parameters must be appended to the URL, starting with a question mark (?). For example: `https://api.hub.embion.nl/v1/uids?id=[id]`.

Var	Description	Mandatory
ID	ID of the token (generated in the HUB)	Yes

### 4.2.3 Response body

var	description	format
allowall	true if use of all UIDs is configured	string
uids	array of all UIDs or a subset of UIDs	array

#### 4.2.4 Request example

💡 Example URL

[https://api.hub.embion.nl/v1/uids?id=\[id\]](https://api.hub.embion.nl/v1/uids?id=[id])

```
{
  "allowall": true,
  "uids": [
    "inverter1_Growatt:1",
    "inverter1_Inv:1",
    "inverter1_Inv:2",
    "inverter1_Inv:3",
    "meter1_Hoofd:1",
    "meter1_P1:1"
  ]
}
```

## 4.3 Get plant data endpoint

**Method:** GET | **URL:** <https://api.hub.embion.nl/v1/plant>

This API endpoint returns data from the given plant. It does not have a body, but it does require a few query parameters in order to work properly.

### 4.3.1 Request headers

This endpoint requires authentication using the API-KEY header (see [Access tokens](#) above).

Var	Description	Mandatory	Format
API-KEY	Actual access token generated in the HUB	Yes	String

### 4.3.2 Request query parameters

Query parameters must be appended to the URL, starting with a question mark (?) and separated by ampersands (&). For example: [https://api.hub.embion.nl/v1/plant?id=\[id\]&period=q&start\\_date=2022-12-14](https://api.hub.embion.nl/v1/plant?id=[id]&period=q&start_date=2022-12-14).

Var	Description	Mandatory	Format
ID	ID of the token (generated in the HUB)	Yes	String
period	Select data return period: 1 last sample (default), q 15 minute, h hourly, w weekly, d daily, m monthly, y yearly	No	String
range	Time range to show: d day (default), w week, m month, y year	No	String
type	Combination type of multiple datapoints: min minimum value in timerange, max maximum 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** Last value

Note that the last data sample is only available within the last two days. If both the start date and period are not defined in the call, only the last data sample is returned if data is available; otherwise, no data will be returned.

**💡** Data time range

- Users can optionally include a time with the `start_date`, which shifts the day interval. However, the total number of returned entries remains unchanged. If no time is specified, a day is considered from 00:00:00 to 23:59:59 in the selected timezone.
- The `start_date` determines the beginning of the data range. The `period` defines the interval between returned dates, while the `range` specifies the end date/time relative to `start_date`, thereby determining the number of entries returned.

**i** Period limitations

Please note there are some limitations when combining period and range

- For period `q` (quarter-hourly) and `h` (hourly), the maximum range is `d` (one day).
- For period `d` (daily), the maximum range is `w` (one week).
- For period `w` (weekly), the maximum range is `m` (one month).
- For period `m` (monthly), the maximum range is `y` (one year.)

### 4.3.3 Response body

The plant data is included in the JSON body of the response. The actual lay-out of the body varies depending on the query parameters given in the request.

The following parameters can be present, if the data is available:

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
ky	Cumulative total yield	1 Wh	Integer
soc	Average state of charge	%	Integer
evku	Cumulative EV charger consumption	1 Wh	Integer
pev	Total EV charger power	1 W	Integer
pbat	Total battery power	1 W	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



### 4.3.4 Request example

#### 💡 Example URL

[https://api.hub.embion.nl/v1/plant?id=\[id\]&period=q&range=d&type=max&start\\_date=2022-12-14](https://api.hub.embion.nl/v1/plant?id=[id]&period=q&range=d&type=max&start_date=2022-12-14)

```
[
  {
    "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,
    "kty": 7200,
    "pgrid": 1000,
    "psol": 1750,
    "soc": 550,
    "soc": 1550,
    "pev": 750,
    "evku": 2450,
    "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,  
"kty": 7200,  
"pgrid": -11600,  
"psol": 2000,  
"soc": 550,  
"soc": 1550,  
"pev": 750,  
"evku": 2450,  
"red": 10000,  
"run": 3,  
"var1": 1,  
"var2": 2,  
"var3": 3,  
"var4": 4,  
"warn": 0  
}  
]
```

## 4.4 Get meter data endpoint

**Method:** GET | **URL:** <https://api.hub.embion.nl/v1/meter>

This API endpoint returns individual meter data. It does not have a body, but it does require a few query parameters in order to work properly.

### 4.4.1 Request headers

This endpoint requires authentication using the API-KEY header (see [Access tokens](#) above).

Var	Description	Mandatory	Format
API-KEY	Actual access token generated in the HUB	Yes	String

### 4.4.2 Request query parameters

Query parameters must be appended to the URL, starting with a question mark (?) and separated by ampersands (&). For example: [https://api.hub.embion.nl/v1/meter?id=\[id\]&uid=\[uid\]&period=q&range=d&type=max&start\\_date=2022-12-14](https://api.hub.embion.nl/v1/meter?id=[id]&uid=[uid]&period=q&range=d&type=max&start_date=2022-12-14)

Var	Description	Mandatory	Format
ID	ID of the token (generated in the HUB)	Yes	String
uid	the uid of the meter to read, only one uid can be entered	Yes	String
period	Select data return period: 1 last sample (default), q 15 minute, h hourly, w weekly, d daily, m monthly, y yearly	No	String
range	Time range to show: d day (default), w week, m month, y year	No	String
type	Combination type of multiple datapoints: min minimum value in timerange, max maximum 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** Last value

Note that the last data sample is only available within the last two days. If both the start date and period are not defined in the call, only the last data sample is returned if data is available; otherwise, no data will be returned.

**💡** Data time range

- Users can optionally include a time with the `start_date`, which shifts the day interval. However, the total number of returned entries remains unchanged. If no time is specified, a day is considered from 00:00:00 to 23:59:59 in the selected timezone.
- The `start_date` determines the beginning of the data range. The `period` defines the interval between returned dates, while the `range` specifies the end date/time relative to `start_date`, thereby determining the number of entries returned.

**i** Period limitations

Please note there are some limitations when combining period and range

- For period `q` (quarter-hourly) and `h` (hourly), the maximum range is `d` (one day).
- For period `d` (daily), the maximum range is `w` (one week).
- For period `w` (weekly), the maximum range is `m` (one month).
- For period `m` (monthly), the maximum range is `y` (one year.)

### 4.4.3 Response body

The meter data is included in the JSON body of the response. The actual lay-out of the body varies depending on the query parameters given in the request. Data that is not used by the given meter is left out from the response body.

The following parameters can be present, if the data is available:

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

## 4.5 Request example

### Example URL

[https://api.hub.embion.nl/v1/meter?id=\[id\]&uid=\[uid\]&period=q&range=d&type=max&start\\_date=2022-12-14](https://api.hub.embion.nl/v1/meter?id=[id]&uid=[uid]&period=q&range=d&type=max&start_date=2022-12-14)

```
[
  {
    "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,  
  "pf11": 50,  
  "pf12": -50,  
  "pf13": 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  
}  
]
```

## 4.6 Get inverter data endpoint

**Method:** GET | **URL:** <https://api.hub.embion.nl/v1/inverter>

This API endpoint returns individual inverter data. It does not have a body, but it does require a few query parameters in order to work properly.

### 4.6.1 Request headers

This endpoint requires authentication using the API-KEY header (see [Access tokens](#) above).

Var	Description	Mandatory	Format
API-KEY	Actual access token generated in the HUB	Yes	String

### 4.6.2 Request query parameters

Query parameters must be appended to the URL, starting with a question mark (?) and separated by ampersands (&). For example: [https://api.hub.embion.nl/v1/inverter?id=\[id\]&uid=\[uid\]&period=h&range=d&type=max&start\\_date=2022-12-14GMT+0100](https://api.hub.embion.nl/v1/inverter?id=[id]&uid=[uid]&period=h&range=d&type=max&start_date=2022-12-14GMT+0100).

Var	Description	Mandatory	Format
ID	ID of the token (generated in the HUB)	Yes	String
uid	the uid of the inverter to read, only one uid can be entered	Yes	String
period	Select data return period: 1 last sample (default), q 15 minute, h hourly, w weekly, d daily, m monthly, y yearly	No	String
range	Time range to show: d day (default), w week, m month, y year	No	String
type	Combination type of multiple datapoints: min minimum value in timerange, max maximum 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** Last value

Note that the last data sample is only available within the last two days. If both the start date and period are not defined in the call, only the last data sample is returned if data is available; otherwise, no data will be returned.

**💡** Data time range

- Users can optionally include a time with the `start_date`, which shifts the day interval. However, the total number of returned entries remains unchanged. If no time is specified, a day is considered from 00:00:00 to 23:59:59 in the selected timezone.
- The `start_date` determines the beginning of the data range. The `period` defines the interval between returned dates, while the `range` specifies the end date/time relative to `start_date`, thereby determining the number of entries returned.

**i** Period limitations

Please note there are some limitations when combining period and range

- For period `q` (quarter-hourly) and `h` (hourly), the maximum range is `d` (one day).
- For period `d` (daily), the maximum range is `w` (one week).
- For period `w` (weekly), the maximum range is `m` (one month).
- For period `m` (monthly), the maximum range is `y` (one year.)

### 4.6.3 Response body

The inverter data is included in the JSON body of the response. The actual lay-out of the body varies depending on the query parameters given in the request.

The body can contain the following parameters, if the data is available:

var	description	units	format
timestamp	Timestamp of the measurement	ISO 8601	string
stat	Inverter status		Integer
kdy	Inverter daily yield	1 Wh	Integer
ky	Inverter total yield	1 Wh	Integer
pac	Inverter AC power	1 W	Integer
psol	Inverter solar 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 leakage 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
string_data	Individual string data (see table below)	<b>stringdata</b>	

**Definition of 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

#### 4.6.4 Request example

##### Example URL

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

```
{
  "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,
  "kty": 6100000,
  "pac": 10000,
  "psol": 20000,
  "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,
```

```
    "u11": 2301,  
    "u12": 2405,  
    "u13": 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,  
    "u11": 2301,  
    "u12": 2405,  
    "u13": 2508  
  }  
}
```

## 4.7 Plant control endpoint

**Method:** POST | **URL:** <https://api.hub.embion.nl/v1/plantcontrol>

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

It is possible to send values that would exceed the capabilities of the plant, but the 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 actively controlled by the GSE.

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

The endpoint can be triggered by sending a POST request to <https://api.hub.embion.nl/v1/plantcontrol>, with a JSON body described below.

### 4.7.1 Request headers

#### ! Permission required

The Control access permission must be enabled for the token, this can be done within the token configuration form. An error response with a 403 HTTP status code will be returned otherwise.

This endpoint requires authentication using the API-KEY header (see [Access tokens](#) above).

Var	Description	Mandatory	Format
API-KEY	Actual access token generated in the HUB	Yes	String

**4.7.2 Request body****⚠ Warning**

Keep in mind that previously set limits are not remembered when a new command is sent, and that new commands overwrite the previous commands.

Var	Description	Mandat	Format	Unit
id	id of the token (generated in the HUB)	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 maximize consumption nom nominal consumption	No	String	
valid_time	Time in seconds that the given command stays active on the GSE (must be equal to or greater than 90). Will be infinite if the value is 0 or the variable wasn't given.	No	Integer	sec.

#### 4.7.2.1 Control explanation

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 in 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 heat-pumps will increase or decrease setpoint to increase power consumption.s

#### 4.7.2.2 Example request body

```
{
  "id": "119mt001pj51d",
  "p_export_limit": 20000,
  "p_import_limit": 50000,
  "control_generation": "max",
  "control_consumption": "nom",
  "valid_time": 200
}
```

### 4.7.3 Response body

The response body contains info about whether the command was successfully sent. The body will be in the JSON format and contains the following parameters:

var	description	format	optional
success	Whether the command was sent (true = sent)	boolean	No
value	Optional description message	string	Yes

The `value` field shows up if the command couldn't be sent or when an existing command was overwritten. The field can have any of the following values:

Var	Description
unsupported	The plant control feature is not supported on the device
disabled	The plant control feature is actively disabled by the device
valid_time_too_short	The <code>valid_time</code> field must be equal to or greater than 90, if it isn't this error is shown
offline	The device is offline
overwritten	The previous command will be overwritten

#### 4.7.3.1 Example response body

Plant control command successfully sent

```
{
  "success": true
}
```

Plant control command couldn't be sent (plant is offline)

```
{
  "success": false,
  "value": "offline"
}
```





All products described in this document are owned by **Embion B.V.**

**Address**

Embion B.V.  
Biestraat 1B  
5126 NH, Gilze

**Contact**

[www.embion.eu](http://www.embion.eu)  
[info@embion.nl](mailto:info@embion.nl)

Copyright 2024 - Embion B.V.