HUB public API manual

v1.19.3

embion

Table of contents

1 Introduction **3**

2 About this manual **3**
2.1 Callouts . . . 3

3 Getting started **4**
3.1 HTTP status codes . . . 4
3.2 Access tokens . . . 4
3.2.1 Generating tokens in the HUB. . . . 4
3.2.2 Using tokens in the API . . . 5

4 API endpoints **6**
4.1 Gzip compression . . . 6
4.2 Get status endpoint . . . 7
4.2.1 Request headers . . . 7
4.2.2 Request query parameters . . . 7
4.2.3 Response body . . . 8
4.3 Get plant data endpoint . . . 10
4.3.1 Request headers . . . 10
4.3.2 Request query parameters . . . 10
4.3.3 Response body . . . 12
4.4 Get meter data endpoint . . . 15
4.4.1 Request headers . . . 15
4.4.2 Request query parameters . . . 15
4.4.3 Response body . . . 16
4.5 Get inverter data endpoint . . . 20
4.5.1 Request headers . . . 20
4.5.2 Request query parameters . . . 20
4.5.3 Response body . . . 21
4.6 Plant control endpoint . . . 25
4.6.1 Request headers . . . 25
4.6.2 Request body . . . 26
4.6.3 Response body . . . 27
4.6.4 Further explanation for control generation and consumption . . . 28

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 HTTP status codes

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

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 - Too Many Requests	The maximum daily request limit of the token and/or the device has been reached

3.2 Access tokens

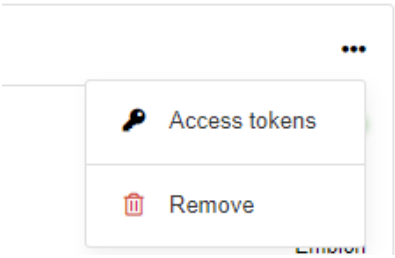
All endpoints in this API require authentication in the form of access tokens. These tokens are unique for each device and determine the permissions and the maximum requests that can be sent per day.

There is a default limit of 250 requests per device per day, each request counts towards this limit. It is possible to change the maximum requests per day per token, but do note that all requests count towards the 250 request limit of the device. Please contact Embion if more than 250 calls per device per day are required.

3.2.1 Generating tokens in the HUB.

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`).

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.

> **What if the `access tokens` option isn't present?**
>
> Users in the HUB portal can have various roles, which in turn have their own permissions. It is possible that you don't have the necessary permission to view and modify access tokens. Please make sure that the `Modifiy device tokens` permission is enabled, or ask another user to generate one.

> **Tokens are generated per device**
>
> Please note, the tokens are generated per device. When multiple devices in one namespace need to be accessed 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.

3.2.2 Using tokens in the API

All endpoints require the access token value to be present in the `API-KEY` HTTP header of the request. Please note that any id references in the endpoints correspond to the id of the token.

Example header information:

```
API-KEY: wzae211vh4ddXlbwt4wdyX1eSjkcgt7dmpqwd5Xnk8amm
```

If the token isn't given, a response with HTTP status code `401` will be returned, including the following body:

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

4 API endpoints

In order to fetch data or control devices, you can send HTTP requests to the API endpoints described below. Filling in your own id, API-KEY and start_date where needed should return valid data.

> **Date and time format**
>
> 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:

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 Gzip compression

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

var	description	mandatory	format
Accept-Encoding: gzip	Enables Gzip compression for all reply messages	No	string

4.2 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.

4.2.1 Request headers

This endpoint requires you to be authenticated using the `API-KEY` header (see Access tokens above).

var	description	mandatory	format
API-KEY	contains the actual token value (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 (`?`) and separated by ampersands (`&`). For example: `https://example.com?var1=qwerty&var2=asdf`.

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

4.2.3 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

Example response body

> **i** Status API call example
>
> Example URL:
> `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",
    "epex_configured": false
}
```

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 you to be authenticated using the `API-KEY` header (see Access tokens above).

var	description	mandatory	format
API-KEY	contains the actual token value (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://example.com?var1=qwerty&var2=asdf`

var	description	mandatory	format
id	id of the token (generated in the HUB)	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

> **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.

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.

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.)

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:

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

Example response body

> **Plant API call example**
>
> The following request URL was used:
> `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
    }
```

4.4 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 you to be authenticated using the `API-KEY` header (see Access tokens above).

var	description	mandatory	format
API-KEY	contains the actual token value (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://example.com?var1=qwerty&var2=asdf`

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 (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

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:

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

var	description	units	format
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

Example response body

> **Meter API call example**
>
> The following request URL was used:
> `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
    }
```

4.5 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.

4.5.1 Request headers

This endpoint requires you to be authenticated using the `API-KEY` header (see Access tokens above).

var	description	mandatory	format
API-KEY	contains the actual token value (generated in the HUB)	Yes	string

4.5.2 Request query parameters

Query parameters must be appended to the URL, starting with a question mark (`?`) and separated by ampersands (`&`). For example: `https://example.com?var1=qwerty&var2=asdf`

var	description	mandatory	format
id	id of the token (generated in the HUB)	Yes	string
uid	the uid (unique identifier) 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

4.5.3 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:

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
string_data	Individual string data (see table below)	**stringdata**	

Definition 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

Example response body

> **Inverter API call example**
>
> The following request URL was used:
> `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
```

```
    },
```

4.6 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 activly 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.

4.6.1 Request headers

> **!** Permission required
>
> The `Control access` permission must be enabled for the token, an error response with a `403` HTTP status code will be returned otherwise.

This endpoint requires you to be authenticated using the `API-KEY` header (see Access tokens above).

var	description	mandatory	format
API-KEY	contains the actual token value (generated in the HUB)	Yes	string

4.6.2 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	mandatory	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: maximise 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	uinteger	s

Example request body

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

4.6.3 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	descripion	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 `valid_time` 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

Example response body

Plant control command successfully sent

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

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

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

4.6.4 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.

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
info@embion.nl