# Protocol GbbConnect to GbbVictronWeb

Contents

[Protocol GbbConnect to GbbVictronWeb 1](#_Toc146295752)

[Mqtt 1](#_Toc146295753)

[GbbConnect 1](#_Toc146295754)

[Response with error 1](#_Toc146295755)

[Operation GetSOC 2](#_Toc146295756)

[Operation GetStatistic 2](#_Toc146295757)

[Operation SetSchedulers 3](#_Toc146295758)

# Mqtt

GbbVictronWeb uses Mqtt protocol to send orders to GbbConnect.

Order are send on <PlantId>/datarequest. In Payload there is Json with data.

Response is return on <PlantId>/dataresponse. In Payload there is Json with data.

# 

# GbbConnect

GbbConnect should:

* Connect to Mqtt with
  + Address: gbbconnect-mqtt.gbbsoft.pl
  + Port: 8883
  + User: Plant Id
  + Password: Plant Token
  + UseTTL: true
  + IgnoreCertificateChainErrors: true,
* Listen on <PlantId>/datarequest
* Send responses to <PlantId>/dataresponse
* Send empty payload to <PlantId>/keepalive every 1 minute

# Response with error

For every request there can be response with error:

{ Operation: “xxx”, Status: “ERROR”, ErrDesc: “<any description of error>”}

|  |  |  |
| --- | --- | --- |
| Key | Type | Value |
| Operation | String | operation from request |
| Status | String | “ERROR” |
| ErrDesc | String | Any description of error for user. |

# Operation GetSOC

Request from GbbVictronWeb:

{ Operation: “GetSOC”}

|  |  |  |
| --- | --- | --- |
| Key | Type | Value |
| Operation | String | “GetSOC” |

Response from GbbConect

{ Operation: “GetSOC”, Status: “OK”, SOC: 33.34}

|  |  |  |
| --- | --- | --- |
| Key | Type | Value |
| Operation | String | “GetSOC” |
| Status | String | “OK” |
| SOC | Decimal, 0-100 | Value of SOC |

# Operation GetStatistics

Request from GbbVictronWeb

{ Operation: “GetStatistics”, FromDate: 2023-09-01, ToDate: 2023-09-02}

|  |  |  |
| --- | --- | --- |
| Key | Type | Value |
| Operation | String | “GetStatistics” |
| FromDate | Date | Start date of period |
| ToDate | Date | End date of period |

Remarks:

* GbbConnect returns any data from this period. If there is no data then return empty table.
* (Day, Hour) must be unique in response
* If there is no data for given day and hour then this row does not appear in table

Response from GbbConnect

{ Operation: “GetStatistic”, Status: “OK”, FromDate: 2023-09-01, ToDate: 2023-09-02, Statistics: [{Day: 2023-09-01, Hour: 0, SOC: 45, MaxSOC: 1, MinSOC: 100, AvrSOC: 55, FromGridkWh: 1.23, ToGridkWh: 1.23, LoadskWh: 1.23, PVProdkWh: 1.23, SOCMin: 1.23}, …]}

|  |  |  |  |
| --- | --- | --- | --- |
| Key |  | Type | Value |
| Operation |  | String | “GetStatistic” |
| FromDate |  | Date | Date copied from request |
| ToDate |  | Date | Date copied from request |
| Statistics |  |  | List of values per day and per hour. GbbVictronWeb don’t assume any order of data |
|  | Day | Date | Date of data |
|  | Hour | Int, 0-23 | Hour of day |
|  | SOC | Decimal, 0-100 | SOC on end of hour |
|  | MinSOC | Decimal, 0-100 | Minimal SOC in this hour |
|  | MaxSOC | Decimal, 0-100 | Maximal SOC in this hour |
|  | AvrSOC | Decimal, 0-100 | Average SOC in this hour (can be (SOCMin+SOCMax)/2) |
|  | PVProdkWh | Decimal | kWh transferred from PV |
|  | FromGridkWh | Decimal, >=0 | kWh transferred from grid |
|  | ToGridkWh | Decimal, >=0 | kWh transferred to grid |
|  | LoadskWh | Decimal | kWh transferred to house |

# Operation SetSchedulers

Request from GbbVictronWeb:

{ Operation: “SetSchedulers”, Schedulers: [

{Hour: 11, FromMinute: 30, ToMinute: 59, ChargeLimitW: 3000, GridSetpointW: 150, PriceLessZero:1, Operation: "Charge", SOC: 90},

{Hour: 12, FromMinute: 0, ToMinute: 59, InputLimitW: 4000, GridSetpointW: -4000, PriceLessZero:0, Operation: "Discharge", SOC: 35},

{Hour: 12, FromMinute: 0, ToMinute: 59, GridSetpointW: -4000, PriceLessZero:0, Operation: "DisableDischarge"},

...]}

|  |  |  |  |
| --- | --- | --- | --- |
| Key |  | Type | Value |
| Operation |  | String | “SetSchedulers” |
| Schedulers |  |  | List of 24 Schedulers for next 24 hours. From 0 to 23 hour. |
|  | Hour | Int, 0-23 | Hour |
|  | FromMinute | Int, 0-59 | Start minute of hour (can be ignored) |
|  | ToMinute | Int, 0-59 | End minute of hour (can be ignored) |
|  | ChargeLimitW | Long, >=0, optional | *Only one can be present: ChargeLimitW or InputLimitW*  Set battery charge limit (W).  Missing of this key means: return to default value or switch off limit. |
|  | InputLimitW | Long, >=0, optional | Set from-grid limit (W).  Missing of this key means: return to default value or switch off limit. |
|  | GridSetpointW | Decimal | Normally <0, target W to feed-in to grid (if >0 then W to get from grid) |
|  | PriceLessZero | Int | 1 – price is less then zero, 0 - else |
| * For Operation=”Charge” | | | |
|  | Operation | String | “Charge” |
|  | SOC | Decimal, 0-100 | Force charge up to this SOC (if CurrSOC>SOC don’t discharge, work as normal) |
| * For Operation=”Discharge” | | | |
|  | Operation | String | “Discharge” |
|  | SOC | Decimal, 0-100 | Force discharge up to this SOC (if CurrSOC<SOC don’t charge, work as normal) |
| * For Operation=”DisableDischarge” | | | |
|  | Operation | String | Block Discharge |
| * For Operation=”Normal” | | | |
|  | Operation | String | Work as normal |

Remarks:

* Schedulers should be remember in GbbConnect (overriding previous data) and automatically apply to Inverter every request from GbbVinctronWeb and every hour (in case connection with GbbVictronWeb failure).

Response from GbbConnect

{ Operation=”SetSchedulers”, Status: “OK” }

|  |  |  |
| --- | --- | --- |
| Key | Type | Value |
| Operation | String | “SetSchedulers” |
| Status | String | “OK” |