

API Manual

Hycube Technologies GmbH

12.04.2019

Version 1.0

Contents

1	Introduction	3
2	URL Requests	3
2.1	Authentication	3
2.2	Informations	3
2.3	Values	4
2.3.1	realtime values	4
2.3.2	values from a given interval of time	5
3	Javascript examples	6
3.1	Javascript	6
3.2	Javascript example for ioBroker	8
4	Legend	10

1 Introduction

The CubeConnect local API is a means to get data from different Hycube devices.

The data exchange is done by making a HTTP request. The URLs for the requests are listed and explained in the following.

Please note that for the use of this API a HyWeb version 1.018 or newer is required.

2 URL Requests

2.1 Authentication

Authenticates a user and returns a token by using the URL:

<http://localIP/auth/>. (user="hycube", password="hycube")

- Format: "user:password"
- Possible outputs:
 - ok
 - unauthorized

2.2 Informations

General system information can be queried by using the following URL:

<http://localIP/info/>.

The response of the request will be a json object containing the informations.

Header values:

- Authorization: The token for the authentication should be entered here.

The response json object for this request looks like the following:

```
1 {
2   "HYCUBE CONTROLLER" : "XX-XX-XXX-XXXX" ,
3   "HYCUBE MACHINE" : "XXX" ,
4   "HYCUBE SERIAL" : "XXXXXXXXXX" ,
5   "HyWeb Version" : 1.014
6 }
```

Listing 1: response json object for the info request

2.3 Values

2.3.1 realtime values

The URL: http://localIP/get_values/ returns the realtime values from the shared memory in json format.

The contained values are the basic needed values (power, voltage, current) about the various devices like inverter, grid, battery and solar.

Header values:

- Authorization: The token for the authentication should be entered here.

The following listing shows a response json object of an eActive System for the request.

```
1 {
2   "Battery_C": 55,
3   "Battery_I": -3.03,
4   "Battery_P": -149,
5   "Battery_V": 48.93,
6   "Grid_f": 49.99,
7   "Grid_P": 64,
8   "Grid_V": 228.4,
9   "Home_P": 144,
10  "Inv1_I": 0.8,
11  "Inv1_P": 80,
12  "Inv1_V": 228.4,
13  "Solar1_P": 0,
14  "Solar1_I": 0,
15  "Solar1_V": 0,
16  "solar2_P": 0,
17  "solar2_I": 0,
18  "solar2_V": 0,
19  "solar_total_P": 0
20 }
```

Listing 2: response json object for the get_values request

2.3.2 values from a given interval of time

To get the values from a defined time interval in the current day the URL: http://localIP/day_val/?from=xxx&to=xxx has to be requested.

Header values:

- Authorization: The token for the authentication should be entered here.

Parameter:

- from: The start value in the format hh:mm:ss.
- to: The end value in the format hh:mm:ss.

```
1 {
2   "10:04:00": {
3     "Chg1_Bat_I": "1.35",
4     "Chg1_Bat_P": "71.0",
5     "Chg1_Bat_V": "52.76",
6     "Inv1_Grd_I": "4.3",
7     "Inv1_Grd_P": "967.0",
8     "Inv1_Grd_V": "229.2",
9     "Inv1_Grd_f": "49.99",
10    "Inv1_PvMpp1_I": "2.3",
11    "Inv1_PvMpp1_P": "905.0",
12    "Inv1_PvMpp1_V": "393.7",
13    "Inv1_PvMpp2_I": "0.6",
14    "Inv1_PvMpp2_P": "214.0",
15    "Inv1_PvMpp2_V": "357.7",
16    "Inv2_Grd_I": "0.0",
17    "Inv2_Grd_P": "0.0",
18    "Inv2_Grd_V": "0.0",
19    "Inv2_PvMpp1_P": "0.0",
20    "Inv2_PvMpp2_P": "0.0",
21    "Mtr1_P": "-846.5",
22    "Mtr2_P": "0.0"
23  },
24  "10:05:00": {
25    "Chg1_Bat_I": "1.33",
26    "Chg1_Bat_P": "70.0",
27    "Chg1_Bat_V": "52.9",
```

```

28     "Inv1_Grd_I": "4.3",
29     "Inv1_Grd_P": "971.0",
30     "Inv1_Grd_V": "229.1",
31     "Inv1_Grd_f": "50.02",
32     "Inv1_PvMpp1_I": "2.3",
33     "Inv1_PvMpp1_P": "905.0",
34     "Inv1_PvMpp1_V": "393.7",
35     "Inv1_PvMpp2_I": "0.6",
36     "Inv1_PvMpp2_P": "215.0",
37     "Inv1_PvMpp2_V": "359.7",
38     "Inv2_Grd_I": "0.0",
39     "Inv2_Grd_P": "0.0",
40     "Inv2_Grd_V": "0.0",
41     "Inv2_PvMpp1_P": "0.0",
42     "Inv2_PvMpp2_P": "0.0",
43     "Mtr1_P": "-850.5",
44     "Mtr2_P": "0.0"
45 }
46 }

```

Listing 3: response json object for the day_val request from 10:04:00 to 10:05:00

3 Javascript examples

In this section an example, how to write and authenticate with Javascript to use the API data from a json object is given.

3.1 Javascript

```

1
2 <!DOCTYPE html>
3 <html lang="de">
4 <head>
5 <meta charset="utf-8"/>
6 </head>
7 <body>
8 <script src="jquery-3.3.1.min.js"></script>
9 <script type="text/javascript">
10
11
12     var auth_token;

```

```

13  var user = "hycube";
14  var password = "hycube";
15  var req_url = "http://local_IP/auth/"
16
17  var ret = $.ajax({
18    url: req_url,
19    headers: { "Authorization": btoa("Basic " + user + ":" +
20      password) },
21
22    ret.done(
23      function (data, status) {
24        auth_token= data;
25
26        var req_url = "http://local_IP/get_values/";
27        var ret = $.ajax({
28          url: req_url,
29          headers: { "Authorization": auth_token },
30
31          });
32
33          ret.done(
34            function (data, status){
35              console.log(data);
36
37            });
38
39          ret.fail(
40            function (jqXHR, textStatus, errorThrown) {
41              console.log(errorThrown);
42
43            });
44
45
46          });
47
48          ret.fail(
49            function (jqXHR, textStatus, errorThrown) {
50              console.log(errorThrown);
51            });
52          });
53
54
55 </script>
56 </body>
57 </html>

```

Listing 4: Javascript example

3.2 Javascript example for ioBroker

First the Javascript adapter must be installed and started if necessary. Then the tab **scripts** has to be activated. As engine Type we choose Javascript. After writing and saving the Javascript programm, start it by pressing the play symbol in the left window to the right of the name of the program. In the following a simple Javascript example is given.

```
1 var TimeInterval= setInterval(MyJavascript,5000);
2
3 function MyJavascript(){
4   var request = require('request');
5   var user= "hycube";
6   var password= "hycube";
7   var options = {
8     url: 'http://10.1.1.51/auth/',
9     headers: {
10      "Authorization": Buffer.from("Basic " + user + ":" +
11        password).toString('base64')
12    },
13    body: '<?xml version="1.0" encoding="UTF-8"?> [...]'
14  };
15  request.post(options, function(error, response, body) {
16    if (error) {
17      console.log("Ein Fehler ist aufgetreten ")
18    } else {
19      var auth_token= body;
20
21
22      var url = require('request');
23      var options = {
24        url: 'http://10.1.1.51/get_values/',
25        headers: {
26          "Authorization": auth_token
27        },
28        body: '<?xml version="1.0" encoding="UTF-8"?> [...]'
29      };
30
31      url.post(options, function(error, response, body) {
32        if (error) {
33          console.log("Ein Fehler ist aufgetreten ")
34        } else {
35          var result = JSON.parse(body);
36
37
38
39          var GridPower= (result["Grid_P"]);
```



```

40     var BatteryPower=result["Battery_P"];
41     var HausPower= result["Home_P"];
42     var solar= result["solar1_P"];
43
44     setState('HyWeb.Batterie', BatteryPower);
45     setState('HyWeb.Grid', GridPower);
46     setState('HyWeb.Hausverbrauch', HausPower);
47     setState('HyWeb.Solar', solar);
48
49 }
50 });
51
52
53
54
55 }
56 });
57
58
59 }

```

Listing 5: Javascript example for ioBroker

4 Legend

Variable	definition	unit of measurement
Battery	Battery	_C _P _U _I
Grid	Grid (Meter1)	_P _U _f
Inv1	Inverter1	_P _U _I
Inv2	Inverter2	_P _U _I
Solar1	Solar1	_P _U _I
Solar2	Solar2	_P _U _I
Solar3	Solar3	_P _U _I
Solar4	Solar4	_P _U _I
Solar_total	the total of solar power	_P
Meter2	Meter2 for external production	_P
Meter3	Meter3 for Triactive users	_P
Home	calculated Home Power	_P

Table 1: Definition of key system for the response json object for the get_values request

Variable	definition	unit of measurement
Chg_1_Bat	Battery	_C _P _U _I
Inv_1_Grd	Inverter1 output	_P _U _I
Inv_2_Grd	Inverter2 output	_P _U _I
Inv_1_PvMpp1	Solar1	_P _U _I
Inv_1_PvMpp2	Solar2	_P _U _I
Inv_2_PvMpp1	Solar3	_P _U _I
Inv_2_PvMpp2	Solar4	_P _U _I
Mtr1	Meter1	_P _U _I
Mtr2	Meter2	_P _U _I
Mtr3	Meter3	_P _U _I

Table 2: Definition of key system for the response json object for the day_val request

Unit of measurement	Name	Unit
C	Capacity	percent
P	Power	watt (W)
U	Voltage	volt (V)
I	Current	Ampere (A)
f	frequenz	Hertz (Hz)

Table 3: Units of measurement and units

Listings

1	response json object for the info request	3
2	response json object for the get_values request	4
3	response json object for the day_val request from 10:04:00 to 10:05:00	5
4	Javascript example	6
5	Javascript example for ioBroker	8

List of Tables

1	Definition of key system for the response json object for the get_values request	10
2	Definition of key system for the response json object for the day_val request	10
3	Units of measurement and units	11