

Energy monitor: API

url, methods, parameters

Version 1.3 / 31.05.2020

API url

<https://ip-address/json/API.php/{Method}?{Parameter=}&..>

test function:

https://ip-address/json/Test_json.php

Methods

getMeters

Returns the list of meters.
- no parameters

Returns (Example):

```
{
  "status": "o.k.",
  "result": [
    {
      "meterId": "76",
      "type": "power",
      "uuid": "b6ae8c20-683c-01e5-0000-0000fe8a7823",
      "cpu_id": "00000000176b51cc",
      "parameter": [
        {
          "type": "5",
          "name": "type",
          "value": "power",
          "val_standard": "",
          "val_max": "",
          "val_min": "",
          "check": true,
          "check_message": ""
        },
        {
          "type": "0",
          "name": "title",
          "value": "Wasser",
          "val_standard": "title",
          "val_max": "",
          "val_min": "",
          "check": true,
          "check_message": ""
        },
        {
          "type": "1",
          "name": "description",
          "value": "Kanal 01",
          "val_standard": "",
          "val_max": "",
          "val_min": "",
          "check": true,
          "check_message": ""
        },
        {
          "type": "1",
          "name": "location",
          "value": "Keller",
          "val_standard": "",
          "val_max": "",
          "val_min": "",
          "check": true,
          "check_message": ""
        },
        {
          "type": "3",
          "name": "intervall",
          "value": "60",
          "val_standard": 300,
          "val_max": 86000,
          "val_min": 30,
          "check": true,
          "check_message": ""
        },
        {
          "type": "2",
          "name": "cost",
          "value": "0.00027",
          "val_standard": 0.0027,
          "val_max": 1,
          "val_min": 1.0e-5,
          "check": true,
          "check_message": ""
        },
        {
          "type": "1",
          "name": "uuidserver",
          "value": "",
          "val_standard": "",
          "val_max": "",
          "val_min": "",
          "check": true,
          "check_message": ""
        },
        {
          "type": "1",
          "name": "urlserver",
          "value": "",
          "val_standard": "",
          "val_max": "",
          "val_min": "",
          "check": true,
          "check_message": ""
        },
        {
          "type": "5",
          "name": "smartme",
          "value": "ON",
          "val_standard": "",
          "val_max": "ON",
          "val_min": "",
          "check": true,
          "check_message": ""
        }
      ]
    }
  ]
}
```

getDay

Returns the power values for a particular day.

Parameter	Description
meterId	Id of the meter as provided by getMeters
day	The day in the month from 1 to 28...31
month	The month from 1 to 12
year	The year (e.g. 2012)

Returns (Example):

```
{ "status": "o.k.", "result": [ { "power": 50.022956848145, "timeEnd": "1493736810", "unit": "Hz" },
{ "power": 50.024360656738, "timeEnd": "1493737090", "unit": "Hz" }, ... ] }
```

getLive

Returns the last measurements captured by the meter

Parameter	Description
meterId	Id of the meter as provided by getMeters

Returns (Example):

```
{ "status": "o.k.", "result": [ { "counter": 0.007, "ID": 76, "title": "Bezug", "type": "power", "unit": "W", "timeEnd": "1542984480000",
"power": 210 } ] }
```

getReading

Return the total consumption or production for a meter.

Parameter	Description
metderId	Id of the meter as provided by getMeters

Returns (Example):

```
{ "status": "o.k.", "result": [ { "title": "Bezug", "counter": 3476.572, "timeEnd": "1542984600000", "type": "power", "unit": "kWh" } ] }
```