Gate HTTP - getting information about air quality from Sensor Community

Note! The following instruction is dedicated for the second generation of HTTP Gate module (FW: 1.1.11-2218B)!

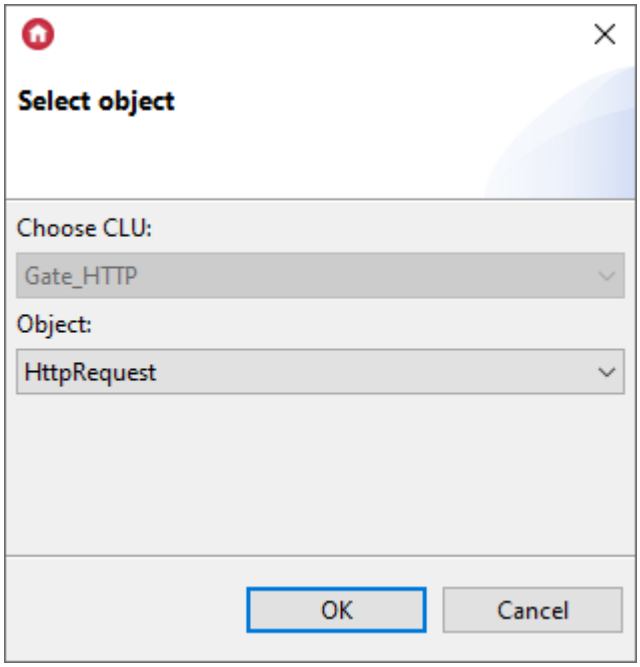
If we want to use information about air quality in the system, we may use an external service for this purpose e.g.: https://sensor.community

According to the example on the website: https://github.com/opendata-stuttgart/meta/wiki/EN-APIs, the API request might look like this: https:// data.sensor.community /airrohr/v1/sensor/**apiID**/

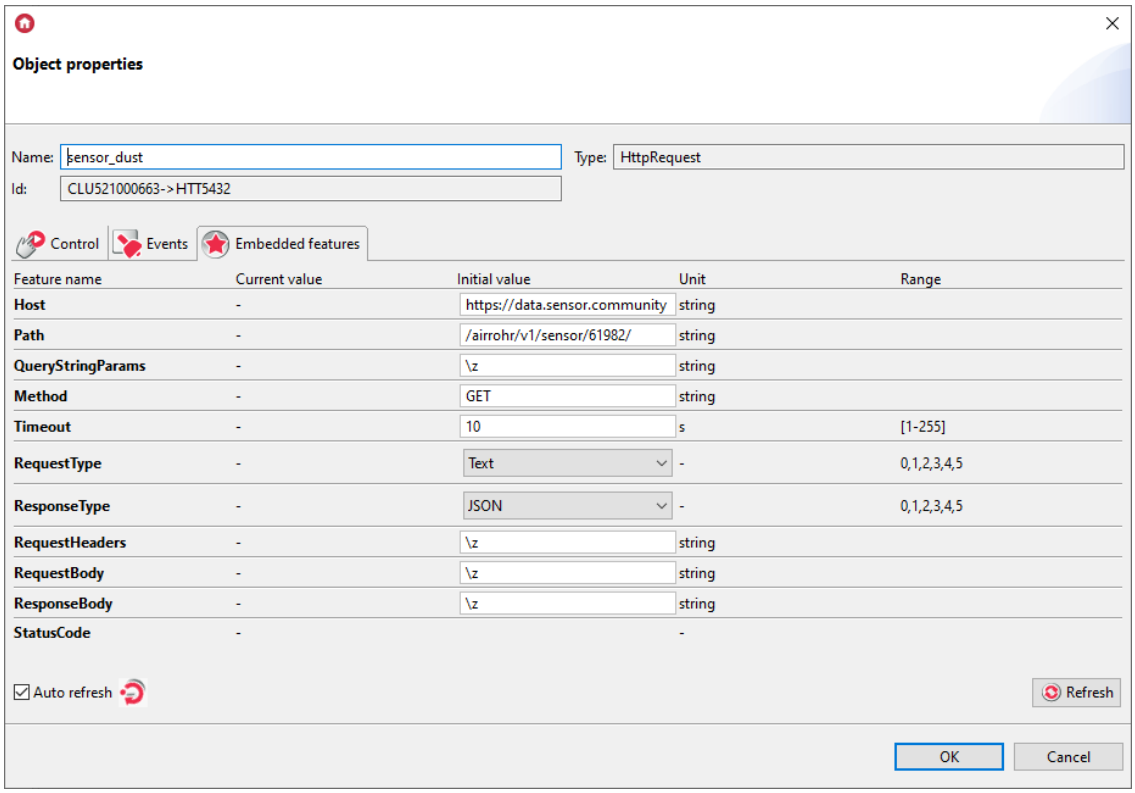
where: **apiID** it is a sensor marking visible after selecting it on the map.

1. Reading values from sensors

- Create the `HttpRequest` virtual object:



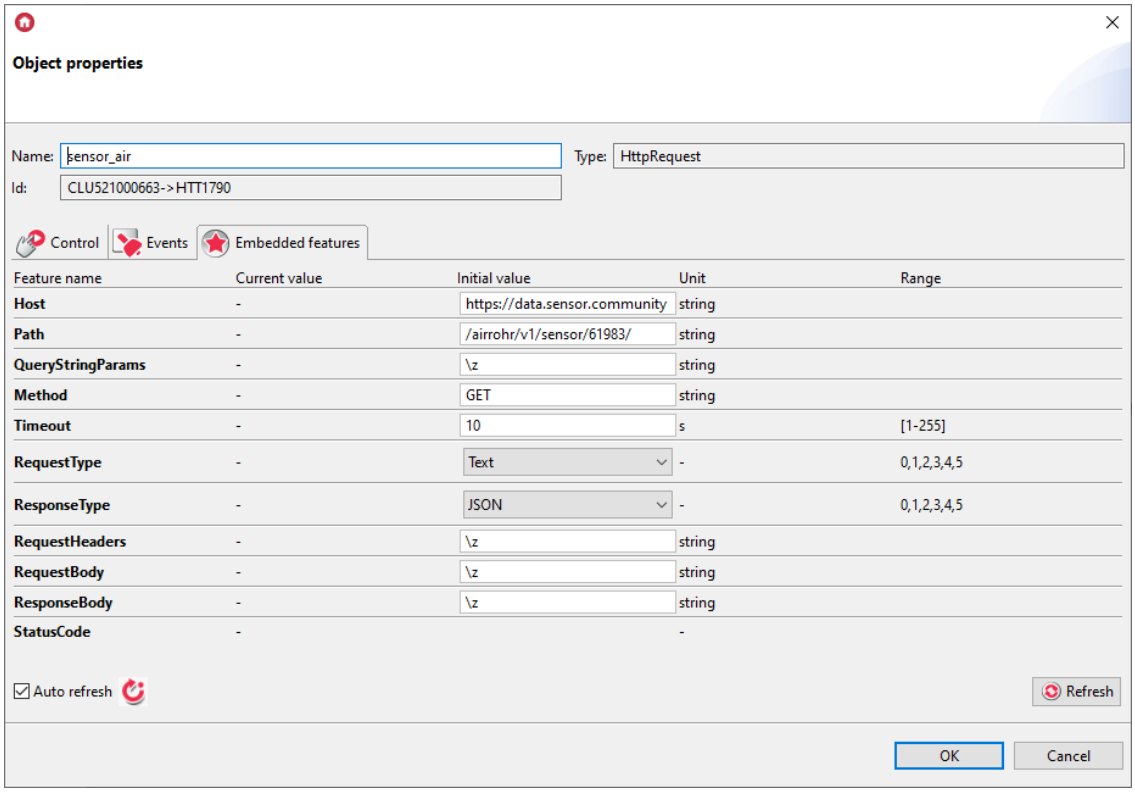
- Set the following parameters in the `HttpRequest` virtual object:



where:

```
Host: https://data.sensor.community
Path: /airrohr/v1/sensor/61982/
```

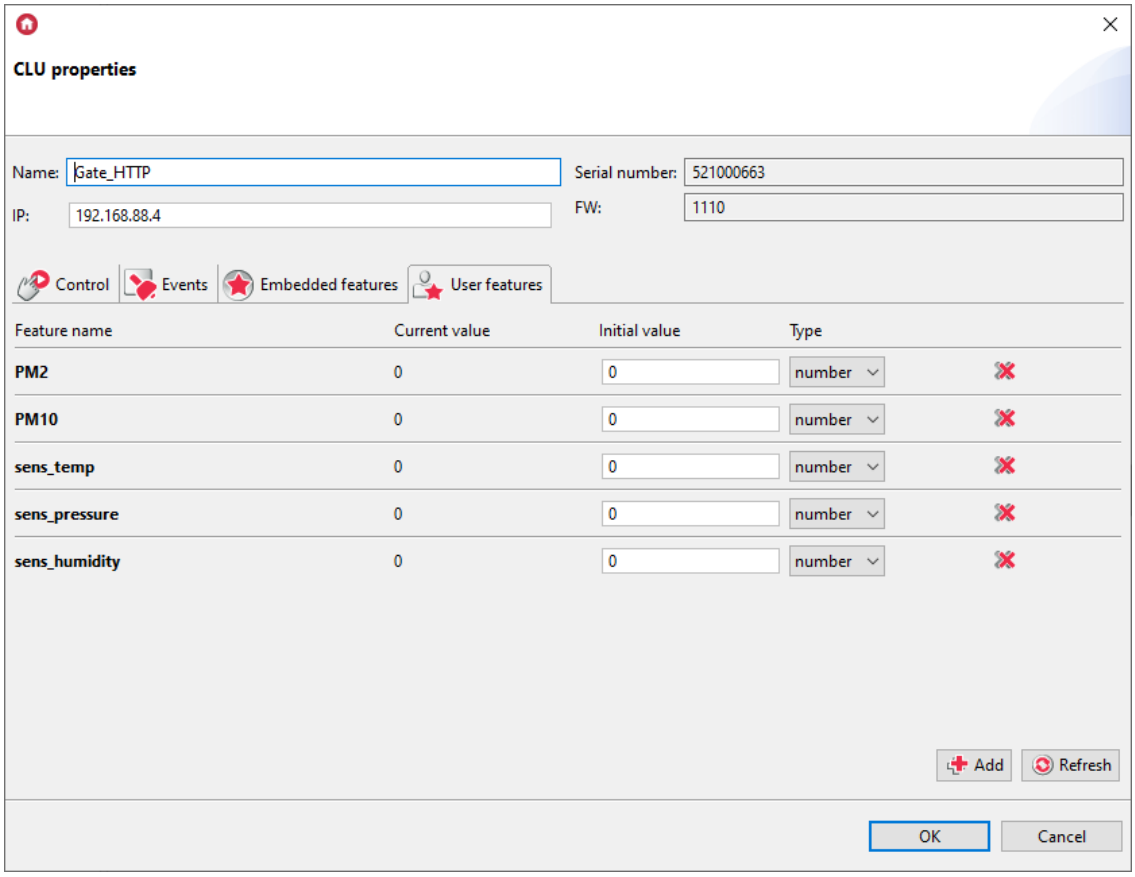
- Create another `HttpRequest` virtual object and set its parameters as follows:



where:

```
Host: https://data.sensor.community
Path: /airrohr/v1/sensor/61983/
```

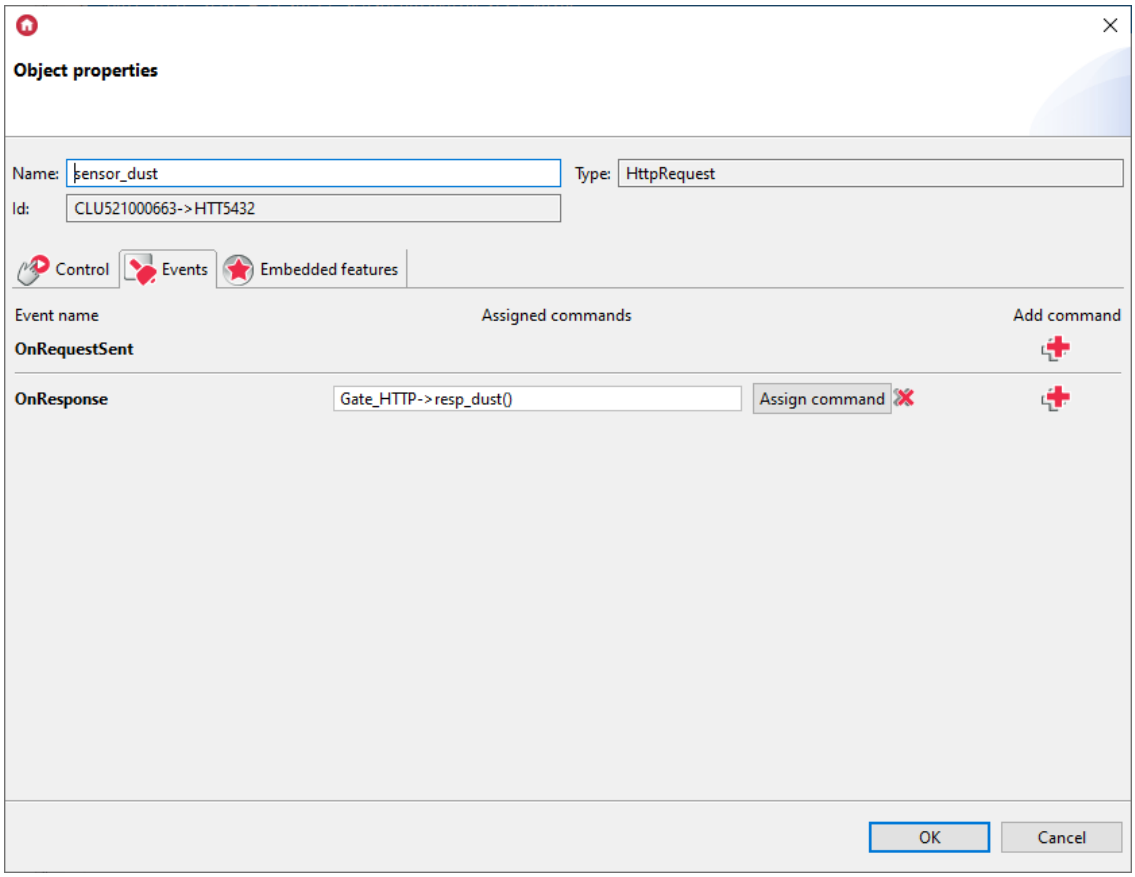
- In the next step, create user features of the *number* type:



- Then prepare a script called *resp_dust*:

```
if(Gate_HTTP->sensor_dust->StatusCode==200) then

  local resp = Gate_HTTP->sensor_dust->ResponseBody
  Gate_HTTP->PM10 = resp[1].sensordatavalues[1].value
  Gate_HTTP->PM2 = resp[1].sensordatavalues[2].value

end
```

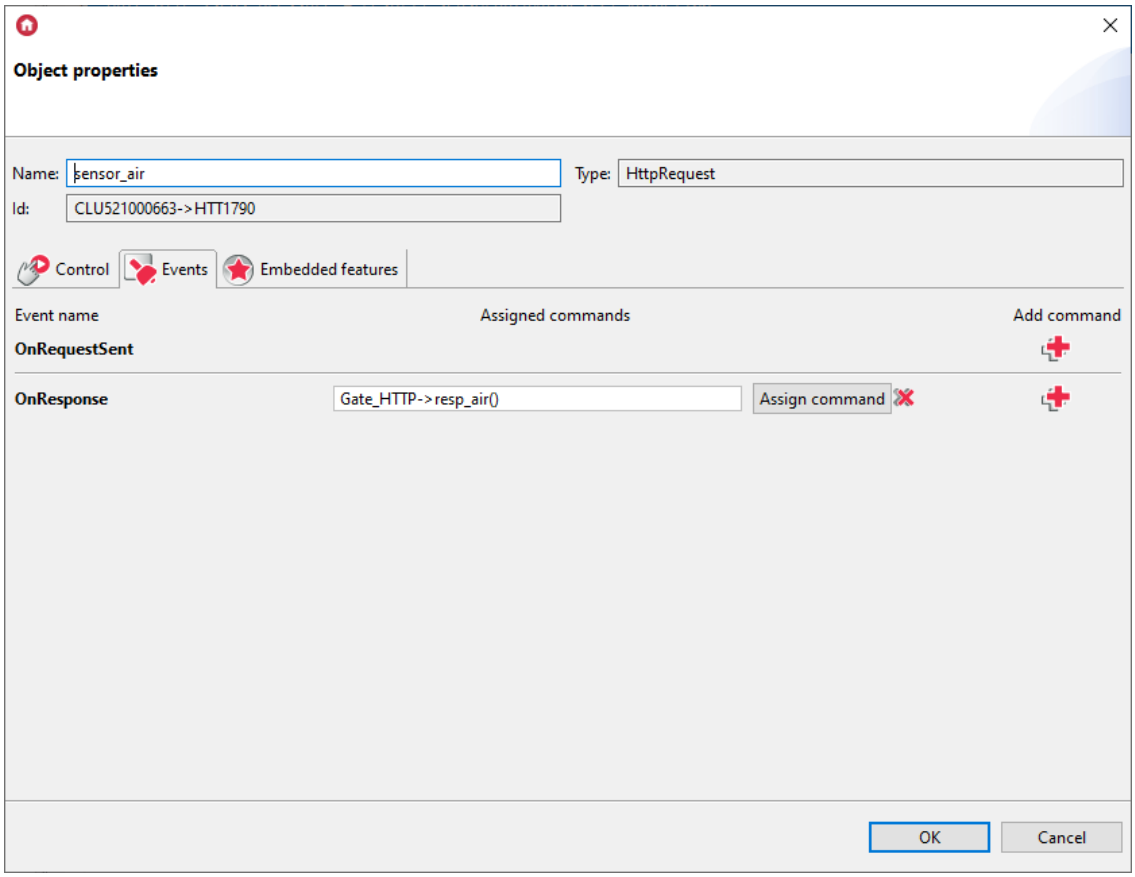
- Assign the script to the **OnResponse** event in the `HttpRequest` virtual object named *sensor_dust*:



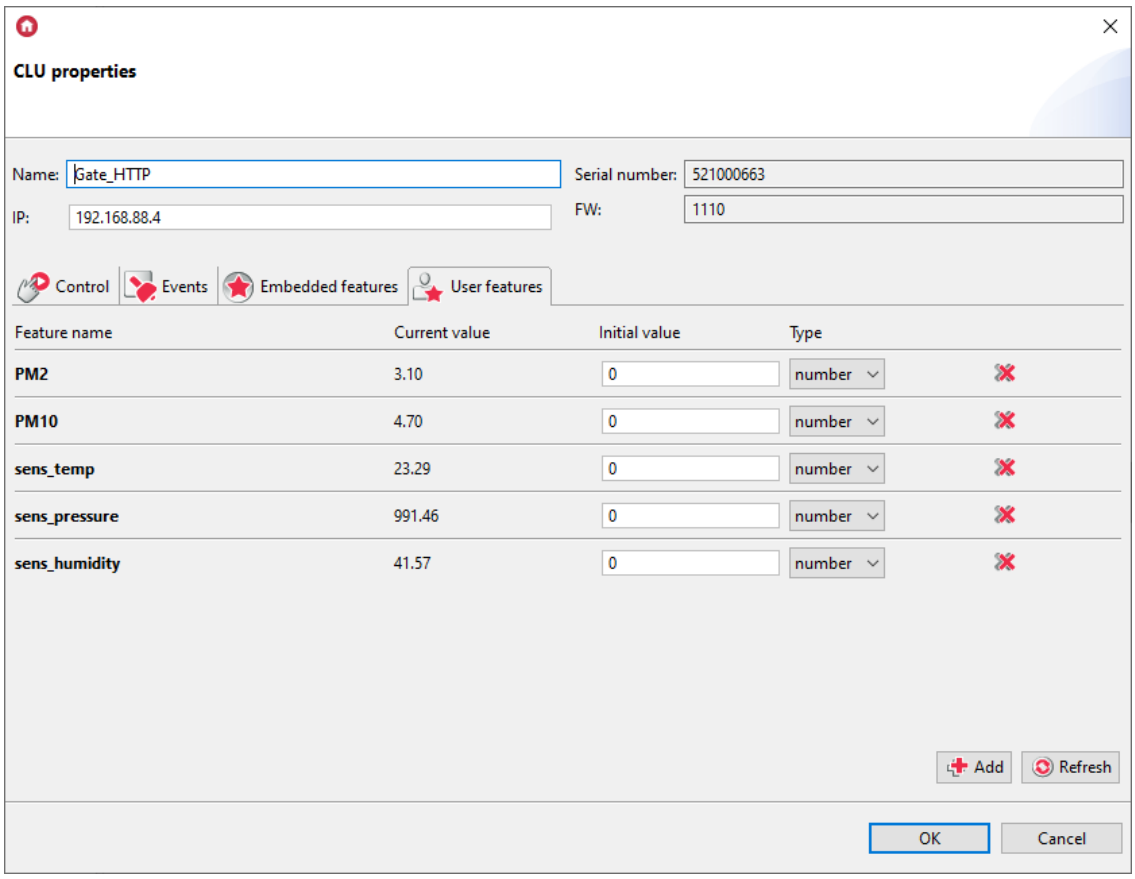
- Then prepare a script called *resp_air*:

```
if(Gate_HTTP->sensor_air->StatusCode==200) then

  local resp = Gate_HTTP->sensor_air->ResponseBody
  Gate_HTTP->sens_temp = resp[1].sensordatavalues[1].value
  Gate_HTTP->sens_pressure = resp[1].sensordatavalues[2].value/100
  Gate_HTTP->sens_humidity = resp[1].sensordatavalues[3].value

end
```

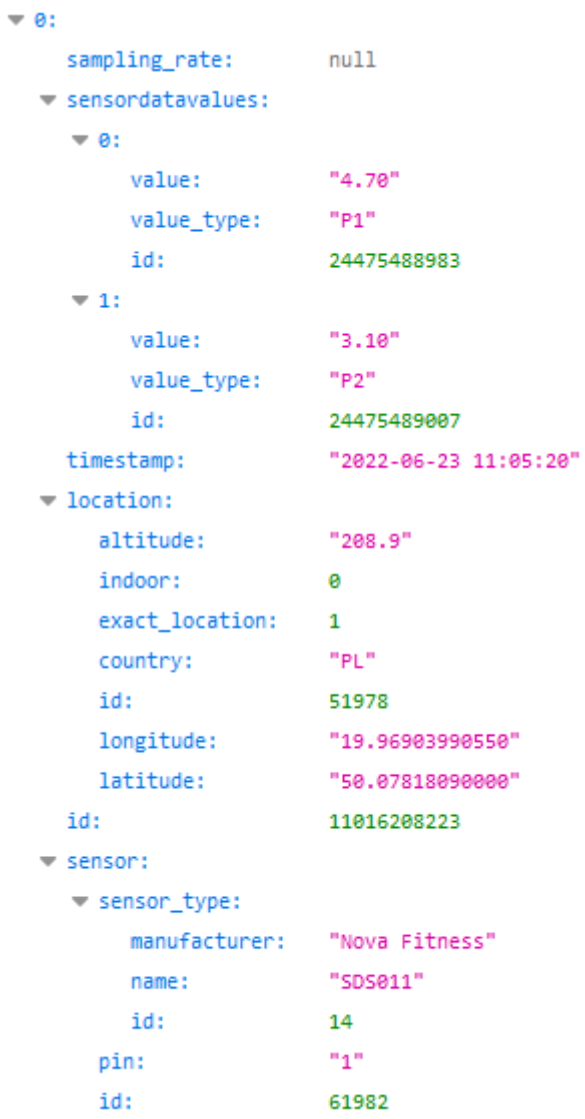
- Assign the script to the **OnResponse** event in the `HttpRequest` virtual object named *sensor_air*:



- Send the configuration to the CLU.
- After sending the configuration, in both objects call the **SendRequest** method.
- After calling the script, the **StatusCode** feature in both objects should get the value of `200`.
- User feature values should get the appropriate values:



- For comparison - responses to requests sent via the browser:



```
▼ 0:
    sampling_rate:        null
  ▼ sensordatavalues:
    ▼ 0:
          value:          "23.29"
          value_type:     "temperature"
          id:             24475490299
    ▼ 1:
          value:          "99145.56"
          value_type:     "pressure"
          id:             24475490368
    ▼ 2:
          value:          "41.57"
          value_type:     "humidity"
          id:             24475490397
    ▼ 3:
          value:          101555.43
          value_type:     "pressure_at_sealevel"
    timestamp:            "2022-06-23 11:05:27"
  ▼ location:
      altitude:           "208.9"
      indoor:             0
      exact_location:     1
      country:            "PL"
      id:                 51978
      longitude:          "19.96903990550"
      latitude:           "50.07818090000"
    id:                   11016208774
  ▼ sensor:
    ▼ sensor_type:
          manufacturer:   "Bosch"
          name:           "BME280"
          id:             17
      pin:                "11"
      id:                 61983
```


- Uzyskane dane można wyświetlić w aplikacji mobilnej, na Smart Panelu lub wykorzystać do tworzenia logiki w systemie.