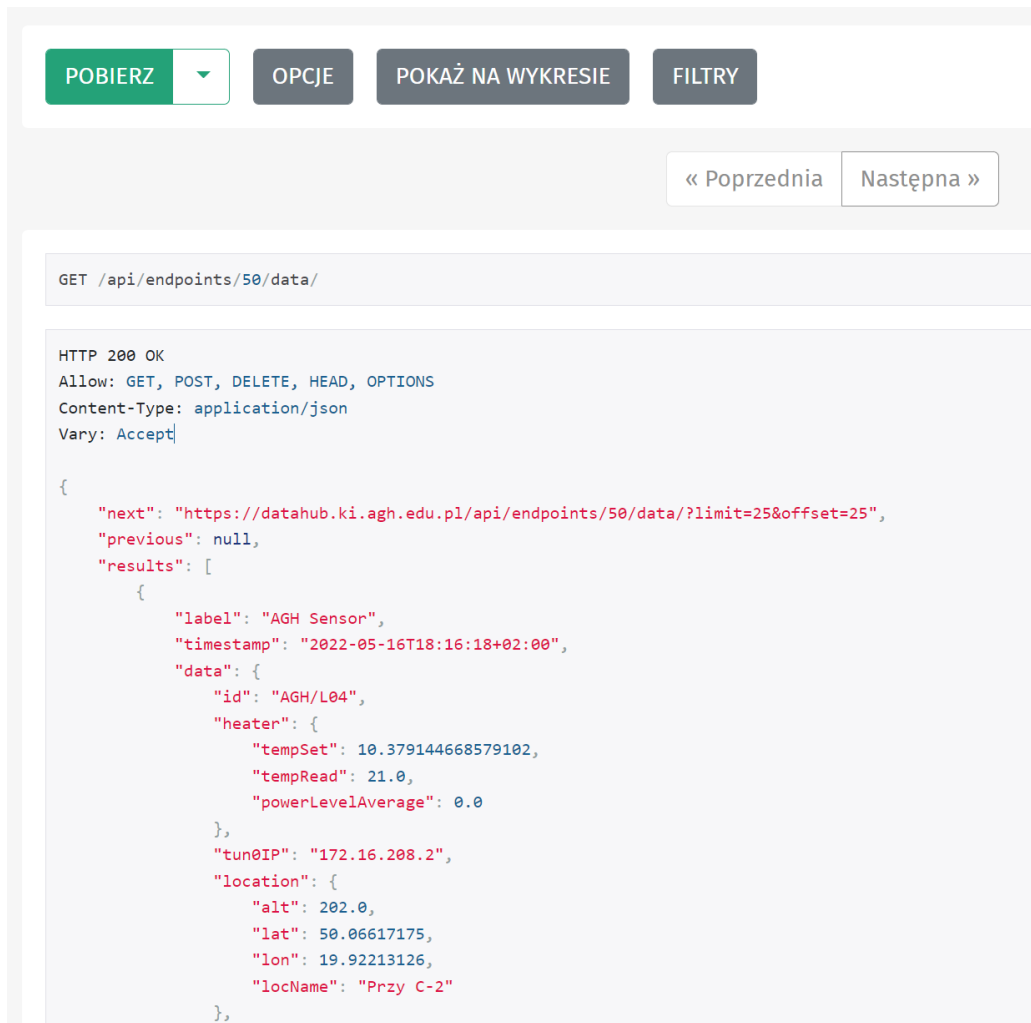


## How to create configuration file?

1. First find Api-s which you want to use on datahub.ki.agh.edu.pl. For example:  
<https://datahub.ki.agh.edu.pl/api/endpoints/50/data/>

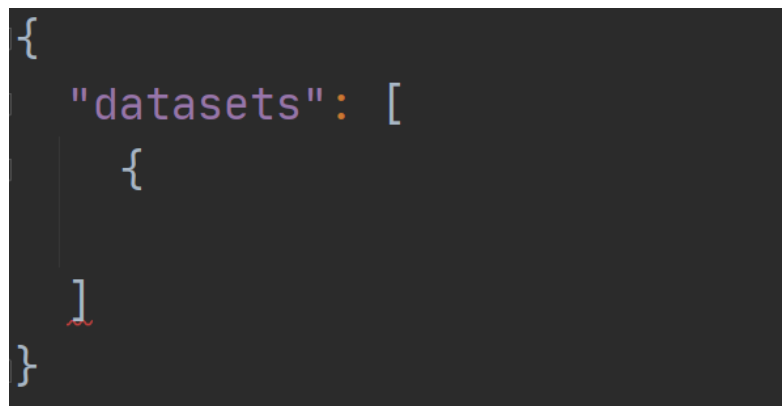


The screenshot shows the DataHub API interface. At the top, there are buttons: POBIERZ (with a dropdown arrow), OPCJE, POKAŻ NA WYKRESIE, and FILTRY. Below these are navigation buttons: « Poprzednia and Następną ». The main content area displays the HTTP response for the GET /api/endpoints/50/data/ endpoint. The response is a JSON object with the following structure:

```
HTTP 200 OK
Allow: GET, POST, DELETE, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "next": "https://datahub.ki.agh.edu.pl/api/endpoints/50/data/?limit=25&offset=25",
  "previous": null,
  "results": [
    {
      "label": "AGH Sensor",
      "timestamp": "2022-05-16T18:16:18+02:00",
      "data": {
        "id": "AGH/L04",
        "heater": {
          "tempSet": 10.379144668579102,
          "tempRead": 21.0,
          "powerLevelAverage": 0.0
        },
        "tun0IP": "172.16.208.2",
        "location": {
          "alt": 202.0,
          "lat": 50.06617175,
          "lon": 19.92213126,
          "locName": "Przy C-2"
        }
      }
    }
  ]
}
```

2. Then create JSON file. It will be our configuration file. In file make list named „datasets”:



```
{
  "datasets": [
    {
      ]
}
```

3. In the list add new record with filed „name”, „type” and „url”:

```
"name": "one day from date",  
"type": "CHART",  
"url": "https://datahub.ki.agh.edu.pl/api/endpoints/50/data/",
```

You can choose name of dataset, its type as CHART or DATA and pass url.

4. Next create list named „static\_data”, which will contain static data from all records , it will make your program faster.

```
"static_data": [  
  {  
    "name": "Location",  
    "source": "location.locName",  
    "unit": null  
  },  
  {  
    "name": "IP",  
    "source": "tun0IP",  
    "unit": null  
  }  
],  
"sensors_data": [  
  {  
    "label": "AGH Sensor",  
    "timestamp": "2022-05-06T14:40:37+02:00",  
    "data": {  
      "id": "OT/GRONIE47A",  
      "heater": {  
        "tempSet": 19.80447006225586,  
        "tempRead": 19.6875,  
        "powerLevelAverage": 0.0  
      },  
      "tun0IP": "172.16.208.5",  
      "location": {  
        "alt": 566.0,  
        "lat": 49.518305,  
        "lon": 20.271117,  
        "locName": "Posesja Gronie 47a"  
      }  
    }  
  },  
  {  
    "label": "AGH Sensor",  
    "timestamp": "2022-05-06T14:40:37+02:00",  
    "data": {  
      "id": "OT/GRONIE47A",  
      "heater": {  
        "tempSet": 19.80447006225586,  
        "tempRead": 19.6875,  
        "powerLevelAverage": 0.0  
      },  
      "tun0IP": "172.16.208.5",  
      "location": {  
        "alt": 566.0,  
        "lat": 49.518305,  
        "lon": 20.271117,  
        "locName": "Posesja Gronie 47a"  
      }  
    }  
  }  
]
```

5. After this, pick the sensor data you want to see as a result:

```
"sensors_data": [  
  {  
    "name": "Temperature",  
    "source": "heater.tempSet",  
    "unit": "celsius"  
  },  
  {  
    "name": "Pressure",  
    "source": "envSensor.pressure",  
    "unit": "Pa"  
  },  
  {  
    "name": "Humidity",  
    "source": "envSensor.relativeHumidity",  
    "unit": "%"  
  },  
  {  
    "name": "PM_1",  
    "source": "particleConcentrationSensor.concentration.pm1",  
    "unit": "µg/m3"  
  },  
  {  
    "name": "PM_2,5",  
    "source": "particleConcentrationSensor.concentration.pm2_5",  
    "unit": "µg/m3"  
  },  
  {  
    "name": "PM_10",  
    "source": "particleConcentrationSensor.concentration.pm10",  
    "unit": "µg/m3"  
  }  
],  
"envSensor": {  
  "dewPoint": 5.37914514541626,  
  "pressure": 994051.3125,  
  "temperature": 21.79364776611328,  
  "relativeHumidity": 34.705787658691406  
},  
"networkIf": "ppp0",  
"verbosity": 1,  
"networkAddr": "10.203.215.195",  
"supplyVoltage": 12.284416198730469,  
"particleConcentrationSensor": {  
  "concentration": {  
    "pm1": 2.879310369491577,  
    "pm10": 5.551723957061768,  
    "pm2_5": 3.7068965435028076  
  }  
}
```

6. Next add timestamp to to set the period from which the data will be displayed

Parametr „from” could be:

- always -> all data
- hour -> data from one hour
- day -> data from one hour
- week -> data from one hour
- month -> data from one hour
- or just date

Parametr „to” could be:

- now -> data to this moment
- or just date

```
"timestamp": {  
  "from": "day",  
  "to": "now"|  
},
```

7. Then add filters: (more in future)

8. In the end add update time:

```
"updates": {  
  "update": true,  
  "update_interval_sec": 600  
}
```

9. If u want to, you can add another dataset to list with different url, name etc.