



Ροή απόκτησης  
δεδομένων  
Node-RED & Minio  
*Case report*

**ΕΦΠ06:**

Υπολογιστικές και  
Δικτυακές Υποδομές II

George Krasakis

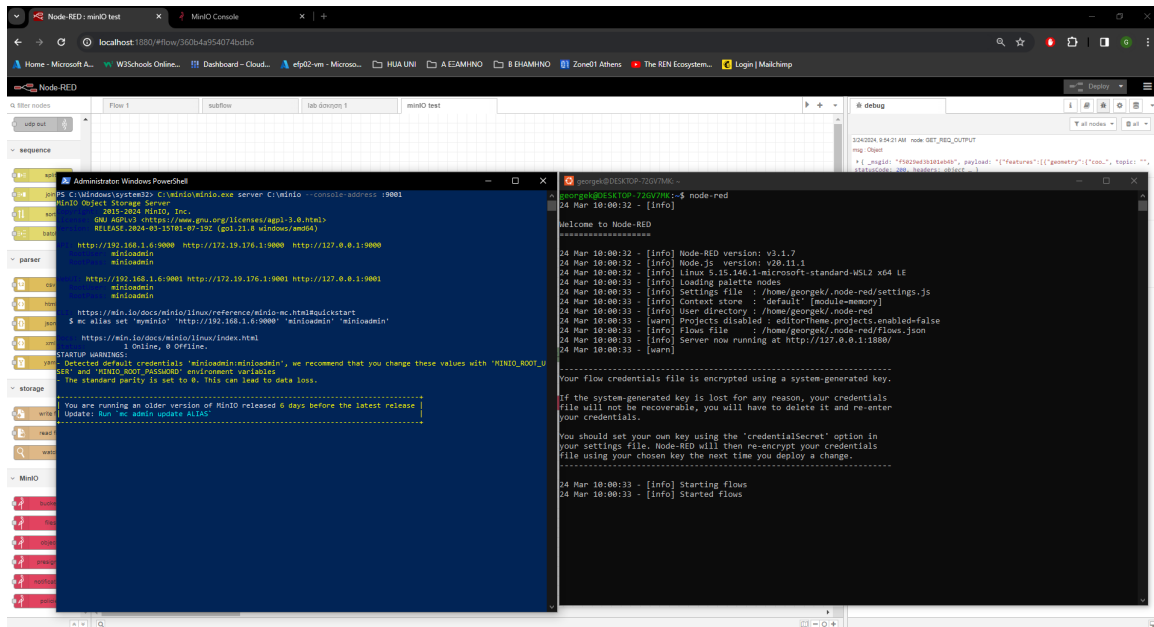
ΑΜ: αρ23012

Διδάσκων: Γ. Κουσιούρης

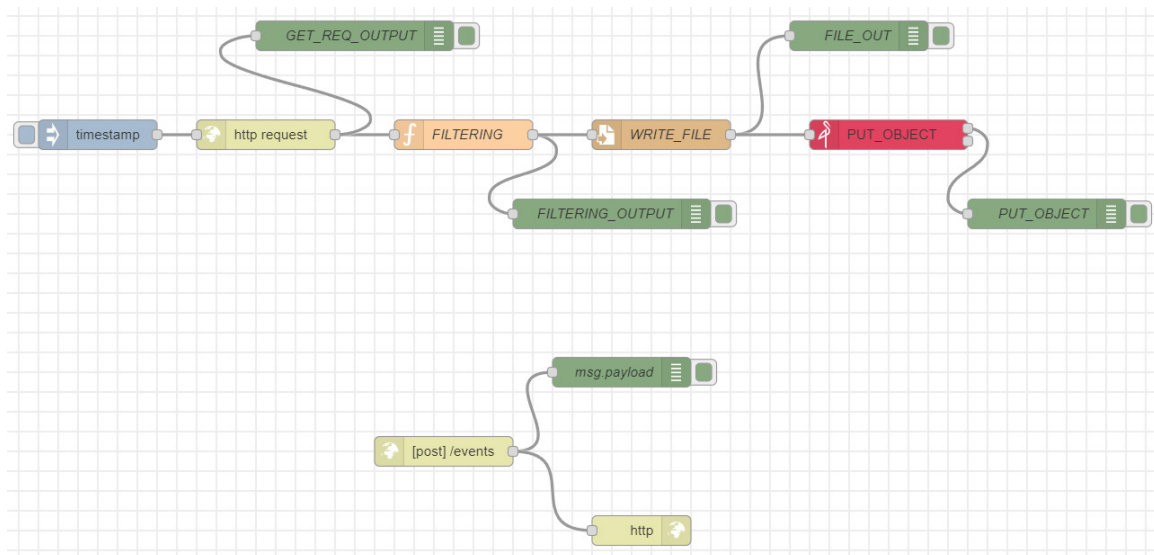
Ημ/νια: 26/03/24

# 1. Node-RED & Minio

Το Minio εκτελείται μέσω PowerShell και σε εγκατεστημένο ubuntu το Node-RED τοπικά στον ίδιο υπολογιστή.



## 2. Flow chart Node-RED



Από το golemio επέλεξα το  [Air Quality \(v2\)](#), /airqualitystations

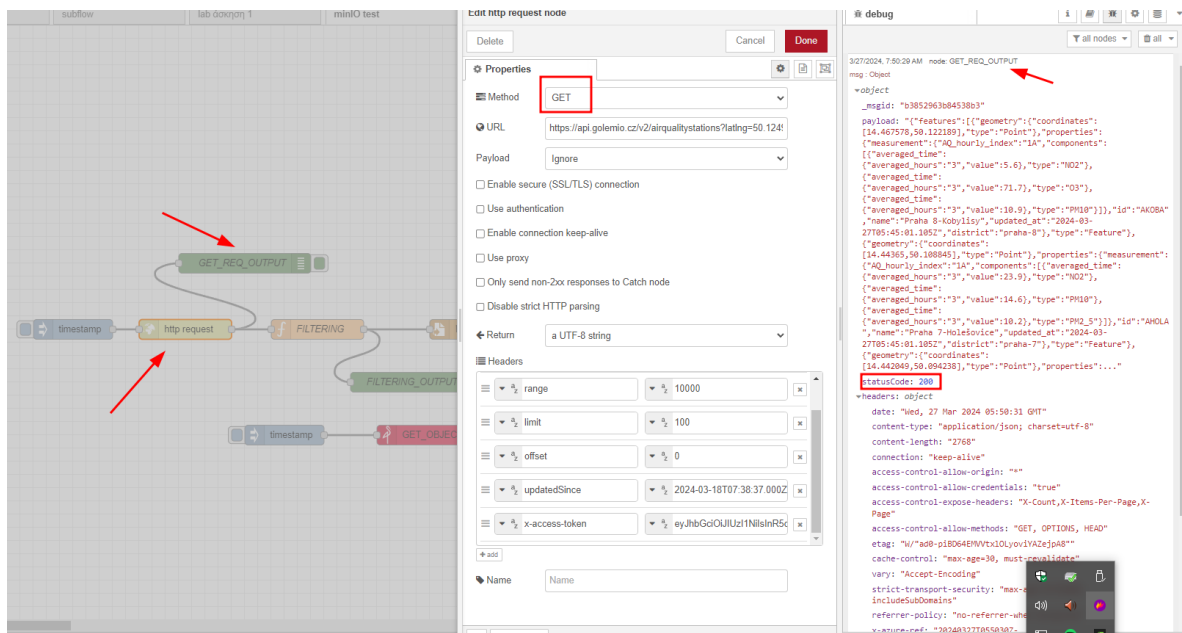
[HTTP GET] --> [JSON Parsing] --> [Function (Data Filtering)] --> [MinIO File Upload]

## 2.1 Επεξήγηση κόμβων που χρησιμοποιήθηκαν & μηνύματα

Χρησιμοποιήθηκαν οι κόμβοι:

**Timestamp:** αποτελεί έναν κόμβο που χρησιμοποιείται για τη δημιουργία μιας χρονοσήμανσης, η οποία αντιπροσωπεύει την τρέχουσα ημερομηνία και ώρα. Αυτός ο κόμβος χρησιμοποιείται συχνά σε ροές για να παρέχει περιεχόμενο σχετικά με το πότε συνέβη ένα συγκεκριμένο γεγονός ή μήνυμα.

**HTTP Request:** Ο κόμβος αυτός χρησιμοποιήθηκε για ένα GET request στο open data API golemio για να αντληθούν τα δεδομένα από το διακομιστή. Στο κόμβο αυτό χρειάστηκε να ρυθμιστούν οι απαραίτητες παράμετροι για να μπορεί να υποβληθεί το αίτημα GET. Αφού διαμορφωθεί, ο κόμβος http request αποστέλλει μια αίτηση στην καθορισμένη url και αναμένει μια απάντηση από τον διακομιστή. Όταν λαμβάνει την απάντηση από τον διακομιστή, ο κόμβος αυτός μεταβιβάζει τα δεδομένα στους επόμενους κόμβους ροής. Πιο συγκεκριμένα:



Παράμετροι που χρησιμοποιήθηκαν,

- Latlng: Ταξινόμηση με βάση την τοποθεσία (γεωγραφικό πλάτος και γεωγραφικό μήκος χωρισμένα με κόμμα, πρώτα το γεωγραφικό πλάτος).
- Range: Φιλτράρισμα με βάση την απόσταση από το γεωγραφικό πλάτος σε μέτρα (ερώτημα εύρους).
- Limit: Περιορίζει τον αριθμό των ανακτηθέντων στοιχείων σε 100.
- Offset: Αριθμός των πρώτων στοιχείων που παραλείπονται.
- UpdatedSince: Φιλτράρει όλα τα αποτελέσματα με παλαιότερη ημερομηνία από αυτήν.
- x-access-token: Το API key που χρειάζεται για να πραγματοποιηθεί το request.

**Function(Filtering):** όταν ένα μήνυμα φτάσει σε έναν κόμβο Function, εκτελείται ένας κώδικας JavaScript. Αυτός ο κώδικας μπορεί να πραγματοποιήσει διάφορες εργασίες, όπως μετασχηματισμό δεδομένων, φιλτράρισμα, επικύρωση, υπολογισμό κ.α. Μετά την εκτέλεση του κώδικα JavaScript, ο κόμβος Function μεταβιβάζει το τροποποιημένο μήνυμα στους επόμενους κόμβους της ροής. Πιο συγκεκριμένα:

```
1 // Λειτουργία για την επεξεργασία των δεδομένων
2 function processResponseData(data) {
3   // Έλεγχος αν τα δεδομένα είναι κενά ή null
4   if (!data) {
5     return [];
6   }
7   try {
8     // Ανάλυση των δεδομένων ως JSON
9     const jsonData = JSON.parse(data);
10
11     // Έλεγχος εάν τα δεδομένα απόκρισης περιέχουν κάποια χαρακτηριστικά
12     if (jsonData && jsonData.features && Array.isArray(jsonData.features)) {
13       // Εξαγωγή συναφών πληροφοριών από κάθε χαρακτηριστικό
14       const processedData = jsonData.features.map(
15         feature => {
16           return {
17             id: feature.properties.id,
18             district: feature.properties.district,
19             name: feature.properties.name,
20             AQ_hourly_index: feature.properties.measurement ? feature.properties.measurement.AQ_hourly_index : null,
21             NO2_value: feature.properties.measurement && feature.properties.measurement.components[0].averaged_time.value : null,
22             updated_at: feature.properties.updated_at
23           };
24         }
25       );
26       return processedData;
27     }
28     else {
29       return [];
30     }
31   } catch (error) {
32     console.error("Error parsing response data:", error);
33     return [];
34   }
35 }
36 // Επεξεργασία των δεδομένων απόκρισης
37 const processedData = processResponseData(msg.payload);
38
39 // Αρχειοποίηση Min και Max
40 let minNO2 = Infinity;
41 let maxNO2 = -Infinity;
42
43 // Εύρεση Μέγιστων και ελάχιστων τιμών στον πίνακα
44 processedData.forEach(item => {
45   if (item.NO2_value < minNO2) minNO2 = item.NO2_value;
46   if (item.NO2_value > maxNO2) maxNO2 = item.NO2_value;
47 });
48
49 // Προσθήκη Metadata
50 msg.metadata = {
51   'NO2_min': minNO2,
52   'NO2_max': maxNO2
53 };
54 // Επιστροφή του τροποποιημένου μηνύματος
55 msg.payload = processedData;
56 return msg;
```

```

35 }
36 // Επεξεργασία των δεδομένων απόκρισης
37 const processedData = processResponseData(msg.payload);
38
39 // Αρχικοποίηση Min και Max
40 let minNO2 = Infinity;
41 let maxNO2 = -Infinity;
42
43 // Εύρεση Μέγιστων και ελάχιστων τιμών στον πίνακα
44 processedData.forEach(item => {
45     if (item.NO2_value < minNO2) minNO2 = item.NO2_value;
46     if (item.NO2_value > maxNO2) maxNO2 = item.NO2_value;
47 });
48
49 // Προσθήκη Metadata
50 msg.metadata = {
51     'NO2_min': minNO2,
52     'NO2_max': maxNO2
53 };
54 // Επιστροφή του τροποποιημένου μηνύματος
55 msg.payload = processedData;
56 return msg;

```

Σε αυτό το τμήμα του κώδικα, τα δεδομένα απόκρισης επεξεργάζονται πρώτα με τη χρήση της συνάρτησης `processResponseData`, με αποτέλεσμα να δημιουργείται ένας πίνακας αντικειμένων που περιέχει σχετικές πληροφορίες. Δύο μεταβλητές, `minNO2` και `maxNO2`, αρχικοποιούνται με `Infinity` και `-Infinity` αντίστοιχα. Στη συνέχεια, μέσω επανάληψης στον πίνακα επεξεργασμένων δεδομένων, προσδιορίζονται οι ελάχιστες και μέγιστες τιμές `NO2` συγκρίνοντας την ιδιότητα `NO2_value` κάθε αντικειμένου με τις τρέχουσες ελάχιστες και μέγιστες τιμές, ενημερώνοντάς τες αναλόγως. Στη συνέχεια, δημιουργείται ένα αντικείμενο μεταδεδομένων, το οποίο ενσωματώνει τις ελάχιστες και μέγιστες τιμές `NO2` και το οποίο ανατίθεται στην εντολή `msg.metadata`. Τέλος, τα επεξεργασμένα δεδομένα εκχωρούνται εκ νέου στο `msg.payload`, διασφαλίζοντας ότι το τροποποιημένο μήνυμα περιλαμβάνει τόσο τα επεξεργασμένα δεδομένα όσο και τα σχετικά μεταδεδομένα πριν επιστραφεί.

The screenshot shows a Node-RED workflow. A 'FILTERING' node is connected to a 'WRITE' node. The 'FILTERING' node's 'On Message' tab is active, displaying a JavaScript function that processes incoming data. The output of the 'FILTERING' node is shown in the 'FILTERING\_OUTPUT' node's message box. The message is an object with a 'msg' property containing an array of objects. Each object represents a data point with properties like 'id', 'district', 'name', 'AQ\_hourly\_index', 'NO2\_value', and 'updated\_at'. The 'statusCode' is 200, and the 'headers' object contains a 'responseUrl' and 'redirectList'. The 'metadata' object contains 'NO2\_min' and 'NO2\_max' values.

➤ Προστέθηκε η παλέτα @reggae\_ulli/node-red-contrib-minio-all-fix

**WriteFile:** Ο κόμβος "Write File" του Node-RED χρησιμοποιείται για την εγγραφή δεδομένων σε ένα αρχείο στο σύστημα αρχείων της συσκευής που εκτελεί το Node-RED.

The screenshot shows a Node-RED workflow. A 'WRITE\_FILE' node is connected to a 'PUT' node. The 'WRITE\_FILE' node's configuration is shown, including the 'Encoding' set to 'utf8' and the 'Name' set to 'WRITE\_FILE'. The output of the 'WRITE\_FILE' node is shown in the 'FILE\_OUT' node's message box. The message is an object with a 'msg' property containing an array of objects. Each object represents a data point with properties like 'id', 'district', 'name', 'AQ\_hourly\_index', 'NO2\_value', and 'updated\_at'. The 'statusCode' is 200, and the 'headers' object contains a 'responseUrl' and 'redirectList'. The 'metadata' object contains 'NO2\_min' and 'NO2\_max' values. The 'filename' is 'FILTERING\_WRITTEN\_FILE'.

**EditFiles:** Ο κόμβος "Edit Files Minio PutObject" στο Node-RED επιτρέπει να γίνει upload ή αντικατάσταση αρχείων σε έναν κάδο MinIO. Ο κόμβος λαμβάνει ένα μήνυμα εισόδου που περιέχει τα δεδομένα και τα μεταδεδομένα του αρχείου, επεξεργάζεται το μήνυμα εισόδου και προετοιμάζει το αρχείο για μεταφόρτωση στον κάδο MinIO. ανάλογα με τη διαμόρφωση και το μήνυμα εισόδου, ο κόμβος είτε μεταφορτώνει ένα νέο αρχείο είτε αντικαθιστά ένα υπάρχον αρχείο στον καθορισμένο κάδο MinIO.

**Properties**

- Name: PUT\_OBJECT
- MiniIO Host: MiniIO Instance
- Operation: fPutObject
- Bucket: airquality
- File: Air\_quality\_data
- Path: FILTERING\_WRITEN FILE
- Meta Data: Meta Data (optional)

**Log**

```

5/19/2024, 8:18:43 PM node: FILTERING_OUTPUT
msg: Object
{
  _msgid: "49287a42baba8289", payload: [{"features":[{"geometry":{"coo...", stat
headers: object, responseUrl: "https://api.golemio.cz/v2/airq..." }
}
5/19/2024, 8:18:43 PM node: FILE_OUT
msg: Object
{
  _msgid: "49287a42baba8289", payload: array[6], statusCode: 200, headers: obj
responseUrl: "https://api.golemio.cz/v2/airq..." }
5/19/2024, 8:18:43 PM node: msg.payload
msg.payload: Object
{
  EventName: "s3:ObjectCreated:Put", Key: "airquality/Air_quality_data", Recor
5/19/2024, 8:18:43 PM node: PUT_OBJECT
msg: Object
{
  object
  _msgid: "49287a42baba8289"
  payload: object
  fPutObject: true
  etag: object
  statusCode: 200
  headers: object
  responseUrl: "https://api.golemio.cz/v2/airqualitystations?
latlng=50.124935%2C14.457284&range=5000&limit=10&offset=0&updatedSince=2024-03
18T07%3A38%3A37.000Z"
  redirectList: array[0]
  retry: 0
  metadata: object
  NO2_min: 4.4
  NO2_max: 14.2
  filename: "FILTERING_WRITEN FILE"
}

```

**Properties**

- Name: MiniIO Instance
- MiniIO Host: 192.168.1.6
- Port: 9000
- Access Key: minioadmin
- Secret Key: \*\*\*\*\*
- Use SSL: ☐

Επιπρόσθετα, υπάρχει και ένας ακόμη κόμβος post/events ο οποίος είναι ένα HTTP service για τις αλλαγές που γίνονται στον bucket που υπάρχει στο minio.

**Workflow**

```

graph LR
    GET_REQ_OUTPUT[GET_REQ_OUTPUT] --> http_request[http request]
    http_request --> FILTERING[FILTERING]
    FILTERING --> WRITE_FILE[WRITE_FILE]
    WRITE_FILE --> PUT_OBJECT[PUT_OBJECT]
    PUT_OBJECT --> FILTERING_OUTPUT[FILTERING_OUTPUT]
    FILTERING_OUTPUT --> msg_payload[msg.payload]
    msg_payload --> post_events[post events]
    post_events --> http[http]

```

**Log**

```

10/14/2024, 10:14:11 AM node: GET_REQ_OUTPUT
msg: Object
{
  _msgid: "ba1869463933447", payload: [{"features":[{"geometry":{"coo...", statusCode: 200, headers: object, responseUrl:
"https://api.golemio.cz/v2/airq..." }
}
10/14/2024, 10:14:11 AM node: FILTERING_OUTPUT
msg: Object
{
  _msgid: "ba1869463933447", payload: array[6], statusCode: 200, headers: object, responseUrl:
"https://api.golemio.cz/v2/airq..." }
10/14/2024, 10:14:11 AM node: FILE_OUT
msg: Object
{
  _msgid: "ba1869463933447", payload: array[6], statusCode: 200, headers: object, responseUrl:
"https://api.golemio.cz/v2/airq..." }
10/14/2024, 10:14:11 AM node: PUT_OBJECT
msg: Object
{
  _msgid: "ba1869463933447", payload: object, statusCode: 200, headers: object, responseUrl:
"https://api.golemio.cz/v2/airq..." }
10/14/2024, 10:14:11 AM node: msg.payload
msg.payload: Object
{
  EventName: "s3:ObjectCreated:Put",
  Key: "airquality/Air_quality_data"
}
10/14/2024, 10:14:11 AM node: post.events
post.events: Object
{
  eventVersion: "3.0",
  eventSource: "minio:s3",
  awsRegion: "",
  eventName: "s3:ObjectCreated:Put",
  eventTime: "2024-03-27T09:14:11.433Z",
  eventSource: "s3:ObjectCreated:Put",
  userIdentity: object
  requestParameters: object
  responseElements: object
  s3: object
  source: object
}

```

Εάν για παράδειγμα γίνει διαγραφή του αρχείου τότε:

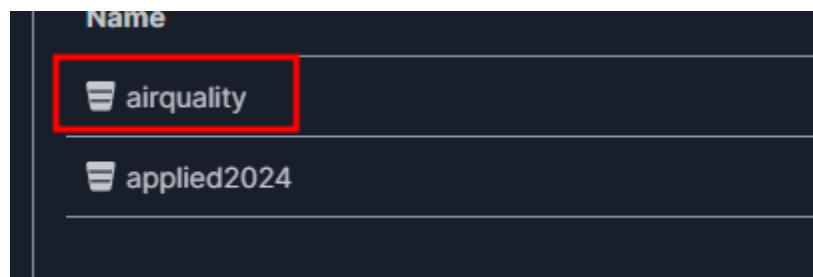
```

3/27/2024, 10:15:58 AM node: msg.payload
msg.payload : Object
  ▼ object
    EventName: "s3:ObjectAccessed:Head"
    Key: "airquality/Air_quality_data"
    ▼ Records: array[1]
      ▼ 0: object
        eventVersion: "2.0"
        eventSource: "minio:s3"
        awsRegion: ""
        eventTime: "2024-03-27T08:15:55.896Z"
        eventName: "s3:ObjectAccessed:Head"
        ▶ userIdentity: object
        ▶ requestParameters: object
        ▶ responseElements: object
        ▶ s3: object
        ▶ source: object
3/27/2024, 10:15:59 AM node: msg.payload
msg.payload : Object
  ▼ object
    EventName: "s3:ObjectRemoved:Delete"
    Key: "airquality/Air_quality_data"
    ▶ Records: array[1]

```

Αρχικά το επέλεξα, 1<sup>ο</sup> μήνυμα, και μετά διαγραφή.

### 3. Minio setup





TOTAL\_POLICY

IAM Policy

Summary

Users

Groups

Raw Policy

Policy Summary

Statements

Effect: Allow

Actions: s3:ListAllMyBuckets

Resources: arn:aws:s3:::\*

Effect: Allow

Actions: s3:GetObject  
s3:PutObject

Resources: arn:aws:s3:::your-bucket/\*

Effect: Allow

Actions: s3:GetBucketLocation  
s3:ListBucket

Resources: arn:aws:s3:::your-bucket

airquality

Created on: Tue, Mar 26 2024 12:27:41 (GMT+2) Access: PRIVATE 862.0 B - 1 Object

Rewind Refresh Upload

< airquality

Create new path

Name	Last Modified	Size
Air_quality_data	Today, 10:20	862.0 B

Status

Service

online

webhook:test

Εάν κατεβάσω το αρχείο και το ανοίξω με vscode το αποτέλεσμα θα είναι:

```
C:\> Users > George > Downloads > Air_quality_data
1 [{"id":"AKOBA","district":"praha-8","name":"Praha 8-Kobylisy","AQ_hourly_index":"1A","NO2_value":13.6,"updated_at":"2024-03-26T11:15:01.204Z"},
2 {"id":"AHOLA","district":"praha-7","name":"Praha 7-Holešovice","AQ_hourly_index":"1B","NO2_value":31.8,"updated_at":"2024-03-26T11:15:01.204Z"},
3 {"id":"AKALA","district":"praha-8","name":"Praha 8-Karlín","AQ_hourly_index":"1B","NO2_value":26.7,"updated_at":"2024-03-26T11:15:01.204Z"},
4 {"id":"AVYNA","district":"praha-9","name":"Praha 9-Vyšehrad","AQ_hourly_index":"1B","NO2_value":27.5,"updated_at":"2024-03-26T11:15:01.205Z"},
5 {"id":"AREPA","district":"praha-1","name":"Praha 1-n. Republiky","AQ_hourly_index":"1B","NO2_value":26.3,"updated_at":"2024-03-26T11:15:01.205Z"},
6 {"id":"ARIEA","district":"praha-2","name":"Praha 2-Riegrovy sady","AQ_hourly_index":"1B","NO2_value":16.7,"updated_at":"2024-03-26T11:15:01.205Z"}]
7
```

Τα δεδομένα τα οποία προβάλλονται:

- **Id:** Ο μοναδικός αναγνωριστικός κωδικός του σταθμού ποιότητας αέρα.
- **district:** Ο ωριαίος δείκτης ποιότητας του αέρα στο σταθμό, που υποδεικνύει το συνολικό επίπεδο ποιότητας του αέρα. Οι τιμές **1A** και **1B** αντιπροσωπεύουν τα διαφορετικά επίπεδα ποιότητας του αέρα, με την τιμή **1A** να υποδηλώνει καλύτερη ποιότητα αέρα από την **1B**.
- **NO2\_value:** Η τιμή του ρύπου διοξειδίου του αζώτου (NO<sub>2</sub>) που μετρήθηκε στο σταθμό. Το NO<sub>2</sub> είναι ένας κοινός ατμοσφαιρικός ρύπος που εκπέμπεται από τις εκπομπές οχημάτων και τις βιομηχανικές δραστηριότητες.
- **updated\_at:** Η χρονική σήμανση που υποδεικνύει πότε ενημερώθηκαν ή συλλέχθηκαν για τελευταία φορά τα δεδομένα.