



Habib University
shaping futures

Analog & Digital Communication - FINAL PROJECT REPORT (SPRING 2022)

Real Time Data Visualization & Alert Dashboard using Node Red and UBI Dots

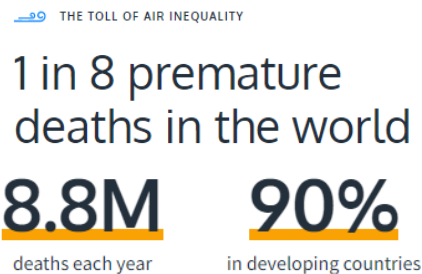
Hamza Irfan
Ayesha Zafar
Zubia Shamim

Introduction:

The curriculum of the Analog and Digital Communication course introduced a flow-based development tool for visual programming known as Node-RED which was later chosen as the basis of our project. In this project, we are expected to identify a problem in our surroundings, use data regarding that problem and employ Node-Red to read that data from Cloud and display it along with some add-ons such as actuators and sensors, which will actualize it as an IoT-based system. Hence, we decided to construct a Live Air Quality Monitoring System that employs an open data platform, OpenAq, that provides live air quality data to users around the globe.

Problem Statement:

One of the most serious public health concerns of our time is poor air quality. The fourth most significant hazard to human health is air pollution. According to the World Health Organization, more than 90% of the world breathes polluted air, with the poorest countries bearing the brunt of the burden.



Motivation:

We live in a world plagued with severe environmental issues and one of the greatest environmental risks to human health is Air Pollution. The World Health Organization (WHO) defines Air pollution as '*Contamination of the indoor or outdoor environment by any chemical, physical or biological agent that modifies the natural characteristics of the atmosphere.*' Household combustion devices, motor vehicles, industrial facilities and forest fires are just some of the common originators of air pollution. To gain an in-depth comprehension of the said issue, being aware of the associated causal factors and their dire consequences is imperative. According to WHO and OpenAq (Open Air Quality data platform), major pollutants that cause air pollution are particulate matter, Carbon monoxide, Sulfur dioxide, Ozone, Nitrogen dioxide, and Black Carbon. It is discernible that the presence of these pollutants in the atmosphere can have appalling consequences such as cardiovascular and respiratory diseases, lung cancer, stroke etc.

As afore-mentioned, poor air quality leads to a large magnitude of premature deaths; however, 91% of those premature deaths occur in low and middle-income countries and the greatest number was found to be in South-Asian and Western Pacific regions. This fact brings the said issue closer to home. The residents of Karachi, the largest cosmopolitan city in Pakistan, are just as affected by this predicament, if not more, as the rest of the world. It is essential to the nature of our project to point this out here that awareness is the key to prevention and correction. This forms the foundation of our project which is to come up with a system that creates awareness around the issue and represents it in an informed and well-displayed manner.

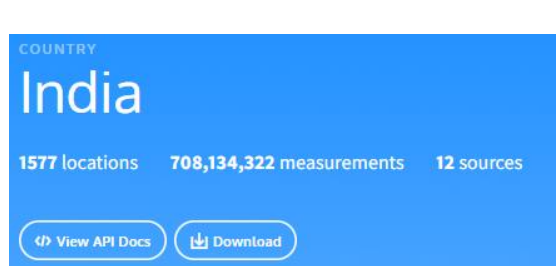
Air Pollution is one of the biggest environmental concerns of our time and air quality is witnessed to be at its worst at the moment. Adhering fully to this theme, we have decided to use an open data platform, OpenAq, which provides live air quality data from around the globe and Node-Red to build a Live Air Quality Monitoring System.

A remarkable feature of this system is that it would not just be reading data from the Cloud as per the project requirement but the data would be real-time and live, making the system much more interesting and fit to use.

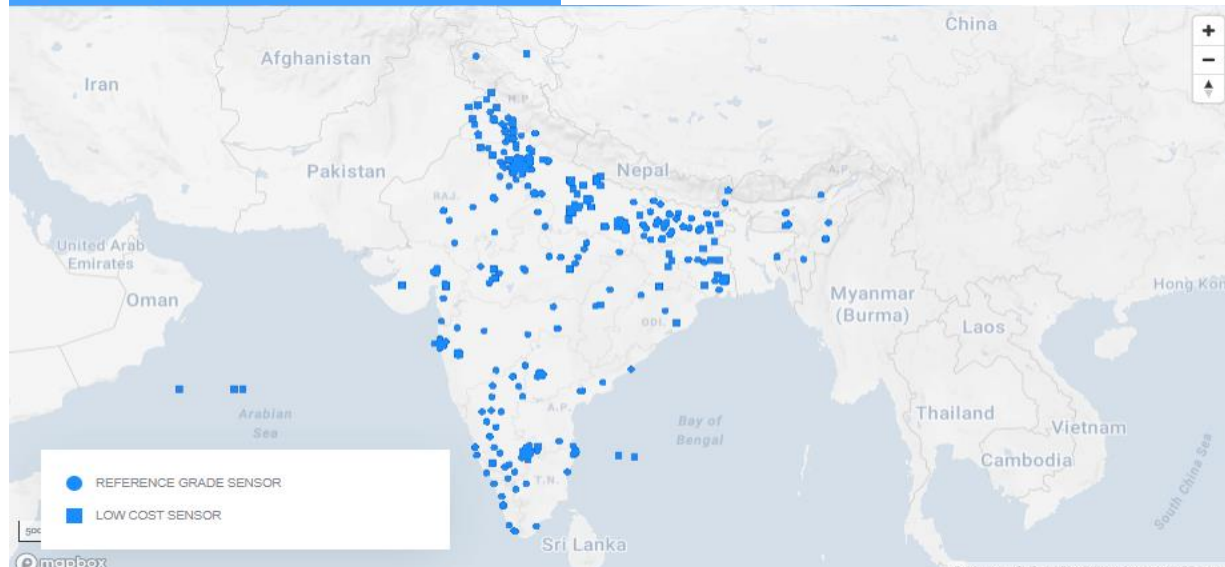
OpenAq - Our Database

Many regions only provide air quality data in human-readable or temporary formats. Even in places where data is freely available, it is exchanged in a variety of formats and industries are unable to properly utilize this data due to a lack of unified data access. OpenAq combines and harmonizes different air quality data into a common format, allowing more individuals to use it for more effective reasons.

This open source collects data from 151 different countries and primarily aggregates PM2.5, PM10, ozone (O3), sulfur dioxide (SO2), nitrogen dioxide (NO2), carbon monoxide (CO), and black carbon (BC) measurements.



For instance, if we choose India as the country, we can see that we have 1577 locations from where we can get the data for the air quality and the dataset of this country consists of 708,134,322 measurements using 12 sources.



Agra

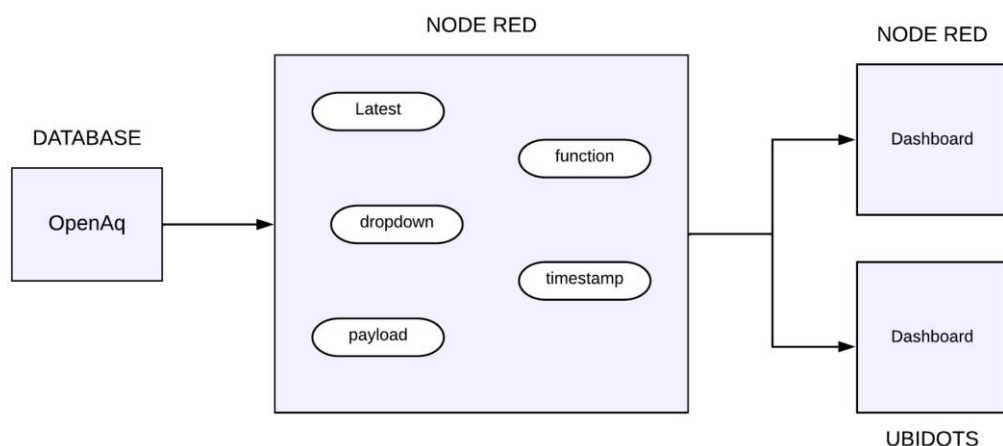
10 locations

[Download](#)

<p>UPDATED AN HOUR AGO</p> <p>Manoharpur, Agra - UPPCB</p> <p>Location ID 227535 in Agra, IN</p> <p>Reference grade Government Stationary</p> <p>COLLECTION DATES 2021/06/04 - 2022/05/09</p> <p>MEASUREMENTS 81,900</p> <p>PARAMETERS O₃ mass, PM2.5, NO₂ mass, SO₂ mass, PM10, CO mass</p> <p>SOURCE Central Pollution Control Board</p> <p>Download View More →</p>	<p>UPDATED A MONTH AGO</p> <p>Manoharpur, Agra - UPPCB</p> <p>Location ID 301748 in Agra, IN</p> <p>Reference grade Government Stationary</p> <p>COLLECTION DATES 2022/03/24 - 2022/03/24</p> <p>MEASUREMENTS 18</p> <p>PARAMETERS PM2.5, CO mass, O₃ mass, PM10, NO₂ mass, SO₂ mass</p> <p>SOURCES data.gov.in, Central Pollution Control Board</p> <p>Download View More →</p>	<p>UPDATED AN HOUR AGO</p> <p>Rohta, Agra - UPPCB</p> <p>Location ID 278159 in Agra, IN</p> <p>Reference grade Government Stationary</p> <p>COLLECTION DATES 2022/01/26 - 2022/05/09</p> <p>MEASUREMENTS 13,194</p> <p>PARAMETERS NO₂ mass, SO₂ mass, CO mass, PM2.5, O₃ mass, PM10</p> <p>SOURCE Central Pollution Control Board</p> <p>Download View More →</p>
---	---	--

Simulation Framework (NodeRed and Ubidots):

The overall flow shows that the database for our project is taken from OpenAq, where there are different locations available for different air contaminant levels. This database is fed into Node Red through nodes for openaq.org which use the OpenAq API to access the data and display on its dashboard.



Different nodes used for OpenAq:

Cities: Provides a simple listing of cities within the platform.

Config: Optionally, this node may be used to define the Open AQ API URL as well as console and file debug logging.

Countries: Provides a quick list of the locations available on the platform.

Latest: For each place in the system, it returns the most recent value for each accessible parameter.

Locations: A list of measurement locations and associated metadata is provided.

Measurements: Data on individual measures is provided.

Parameters: Provides a basic list of the platform's parameters.

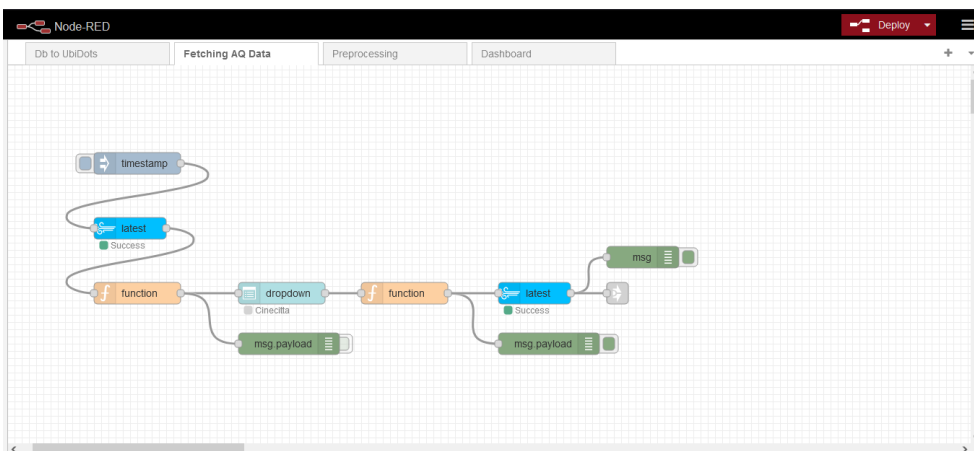
Sources: Provides a list of data sources.

How does data visualization process work?

In order to create a modular Node Red flow, we have divided our Node Red flows into different sections, each section has a specific purpose, so that if we want to change a certain functionality, we won't have to disturb the entire flow.

Fetching AQ Data Flow:

This is the first flow of our network where we are extracting data from the “latest” OpenAq node, that takes in the location and city parameter to provide all the latest air quality data. Then we have a function node to extract the data in a specific format as follows:



Latest Node Parameters:

Delete

Cancel

Done

⚙️ Properties

⚙️ 📄 🖨️

API

Add new openaq-config...

📍 Location:

Location

Location

City

Roma

Country

IT

📍 Coordinates:

Latitude

Latitude

 Longitude

Longitude

Radius (in meters)

2500

▼ Filter:

Parameter

None

Has geo

☐

⬆️ Order by:

☐ Enabled

Function Node:

Delete

Cancel

Done

⚙️ Properties

⚙️ 📄 🖨️

📁 Name

Name

⚙️ Setup

On Start

On Message

On Stop

1

2

3

4

5

6

7

let locations = [];

let results = msg.payload.results

results.forEach(obj=>{

locations.push(obj.location)

})

return {options: locations};

Output:

```
10/05/2022, 00:10:14 node: bccb5da31125297a
msg.payload: Object
▼ object
  ▶ meta: object
  ▼ results: array[1]
    ▼ 0: object
      location: "US Diplomatic Post: Karachi"
      city: "Karachi"
      country: "PK"
      ▼ coordinates: object
        latitude: 24.8415
        longitude: 67.0091
      ▼ measurements: array[1]
        ▼ 0: object
          parameter: "pm25"
          value: -999
          lastUpdated: "2022-05-09T17:00:00+00:00"
          unit: "µg/m³"
```

Once the locations are extracted from the node, it is sent to the dropdown node which will display the list of locations in country on our Node Red dashboard. Then the next function and latest node help us select the location from drop down and extract its specific information, now we have obtained the information we need, it is then sent to another flow using link out for data processing.

Function Node:

Edit function node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🔗

📌 Name

Name

📄 ▼

⚙ Setup

On Start

On Message

On Stop

1

let location = msg.payload;

2

return{location:location};

Link Out Node:

Edit link out node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🔗

📌 Name

Data

Mode

Send to all connected link nodes

▼

🔍

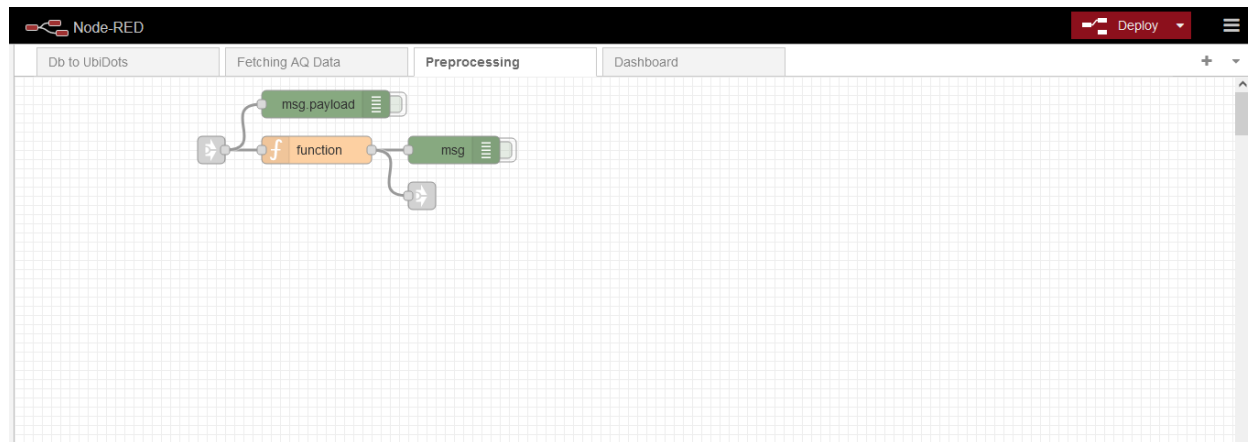
▼ Preprocessing

☒ d721a0401884c870

▼ Dashboard

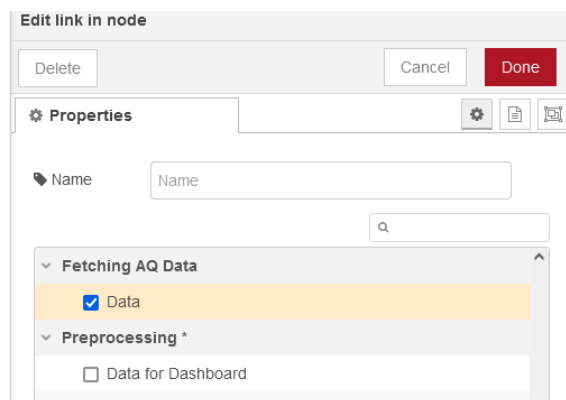
☐ Data

Preprocessing Flow:

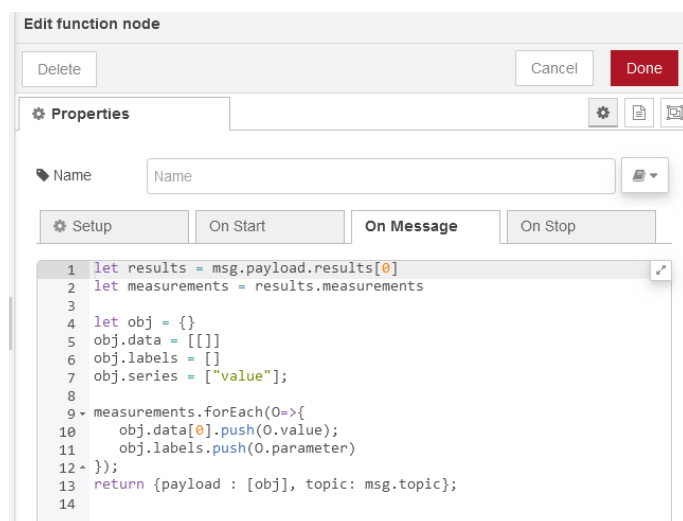


In this flow we are receiving data from the previous fetching AQ data flow through Link in node and then preparing it to be displayed on the dashboard using a function node, then sending it to Dashboard flow using another link out node.

Link In Node:



Function Node:



Link Out Node:

Edit link out node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🖨️

🔖 Name

Data for Dashboard

Mode

Send to all connected link nodes

▼

🔍

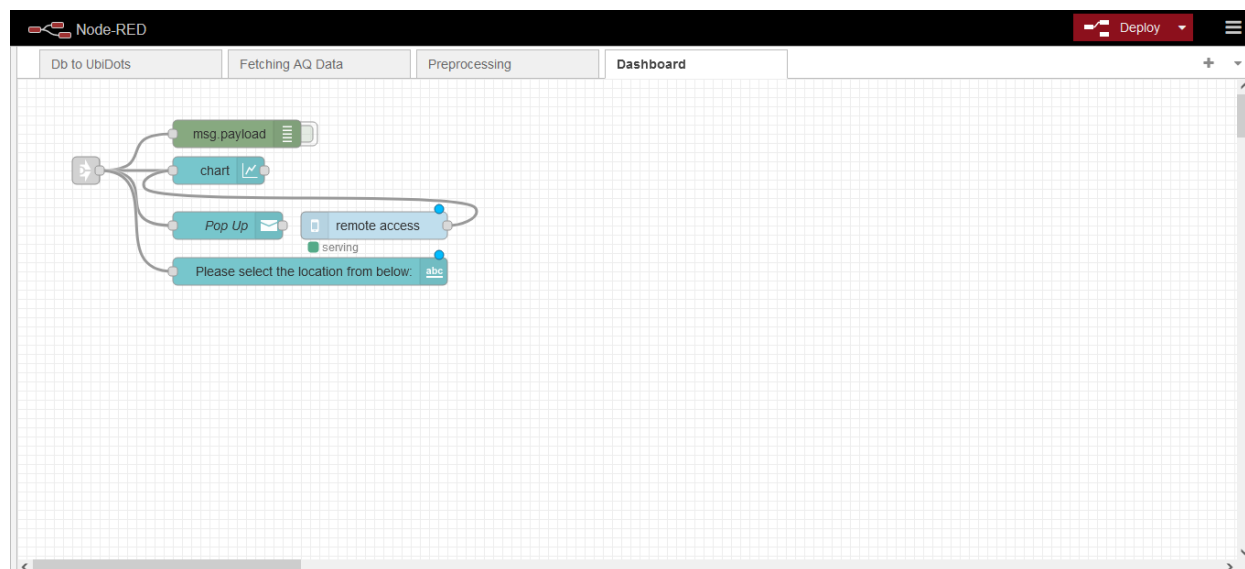
▼ Preprocessing *

☐ d721a0401884c870

▼ Dashboard

☒ Data

Dashboard Flow:



In this flow we are displaying the received data from preprocessing flow using link in and then sending it to different nodes for display on dashboard board such as chart node and mobile app for display.

Link In Node:

Edit link in node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🔗

📁 Name

Data

🔍

▼ Fetching AQ Data

☐ Data

▼ Preprocessing

☒ Data for Dashboard

Chart Node:

Edit chart node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

📁 Group

[Real Time Air Quality Monitoring] Chz ▾

✎

📏 Size

6 x 6

🏷 Label

chart

📊 Type

Bar chart

▾

Y-axis

min 0

max 200

Legend

Show ▾

☐ Use first colour for all bars

Series Colours

Remote Access Node:

Edit remote-access node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

🔧 Config

Node-RED UI ▾

✎

🏷 Name

Name

🔧 SSH log

☐ Activate verbose logging

Help

The access node allows you to access a website locally or remotely from the app. Usually this is your dashboard or a page of the UI Builder.

Create a node of type "remote access" for each page you want to access. Besides the name, this node contains only one config node. The configuration is outsourced to the config node, as these settings are also used for other functions like push notifications.

Pop up Node:

Edit notification node

Delete

Cancel

Done

Properties

Layout

OK / Cancel Dialog with Input

Send to all browser sessions.

☒

Default action label

OK

Secondary action label

Cancel

Accept raw HTML/JavaScript input in msg.payload to format popup.

☒

</> Class

Check Graphs

Topic

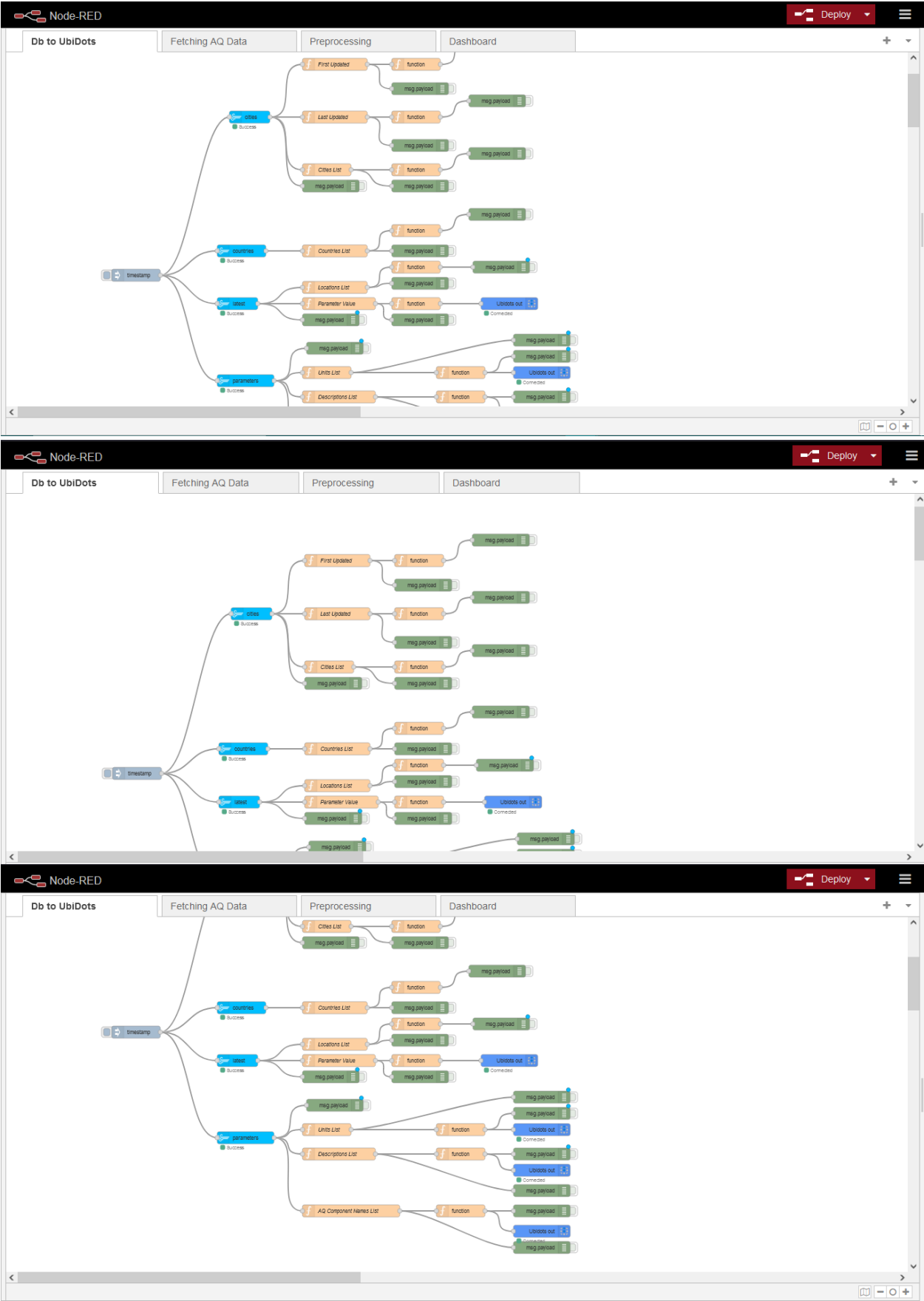
Data Levels Updated

Name

Pop Up

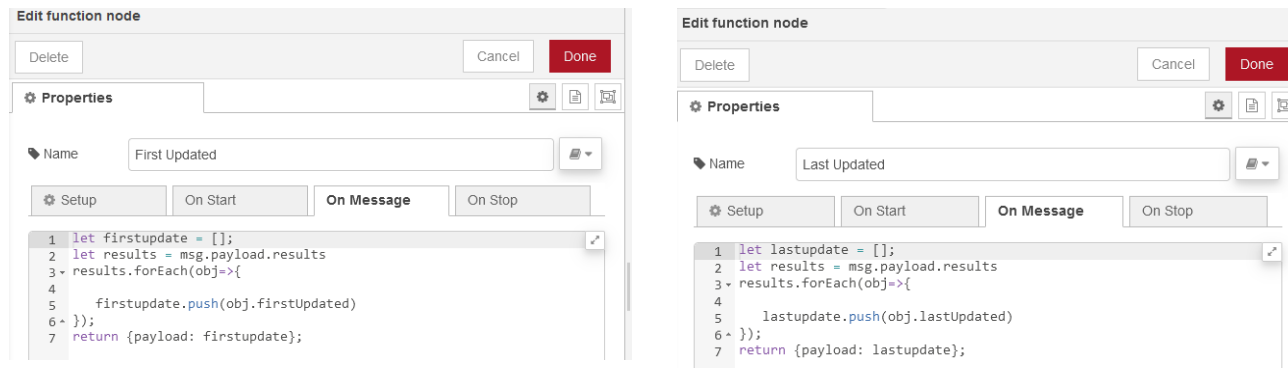
Note: checking Accept raw HTML/JavaScript can allow injection of

Database to Ubi Dots:



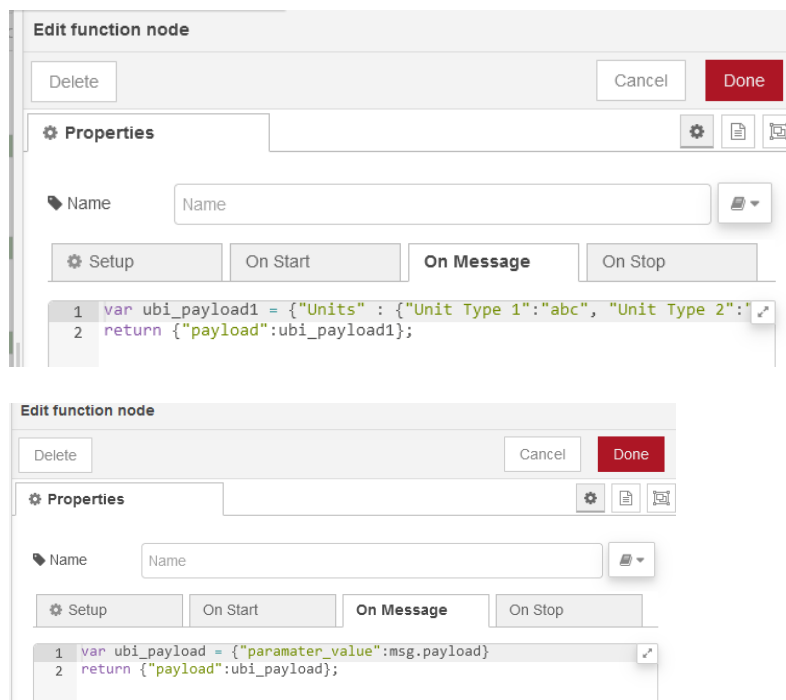
In this flow we are using all the OpenAq nodes present in Node Red, the first layer of functions is extracting data from the nodes, then the second layer of functions are preparing the data to be sent to Ubi Dots.

Function Nodes:



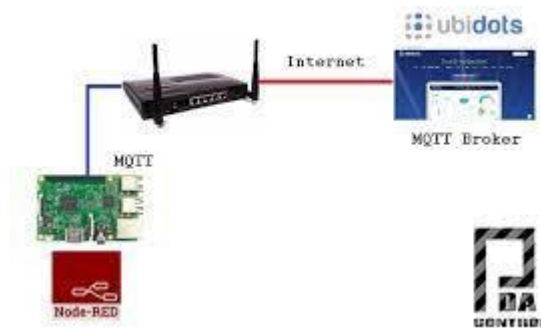
The codes used in function nodes are the same in every function node, only the output variable is different and a different value is extracted from the data nodes.

Second Function Nodes:

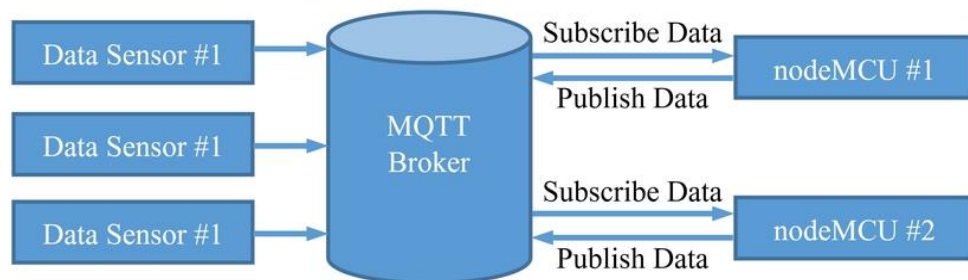


The above function nodes are using the output from previous function nodes as input to send data to ubi dots.

Protocol & Algorithm:

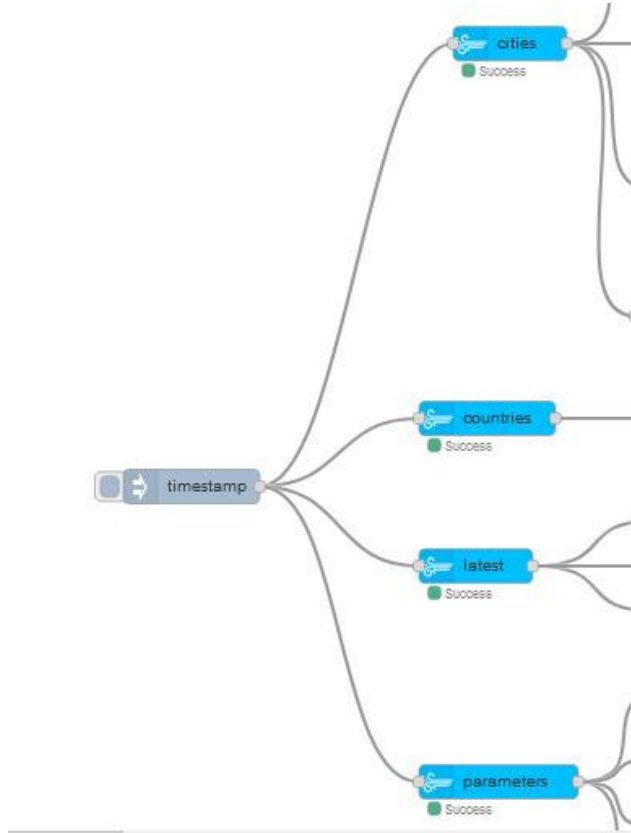


MQTT (Message Queue Telemetry Transport) is built as a super-lightweight publish/subscribe message transport that's perfect for linking faraway devices with minimal coding and network bandwidth. The MQTT Protocol is used to enhance data throughput between sensor nodes and brokers over a wireless local area network (WLAN). Data on air quality measurements is published in real time via an internet-accessible web server. The sensor acts as a publisher and the subscribers are the applications which are interested in the data being published. The MQTT broker acts as a bridge/centerpoint between the published data and the subscriber or client. Ubidots and Node Red are our MQTT clients which are connected through an MQTT broker. The four methods which the MQTT works on are connect, disconnect, publish and subscribe.

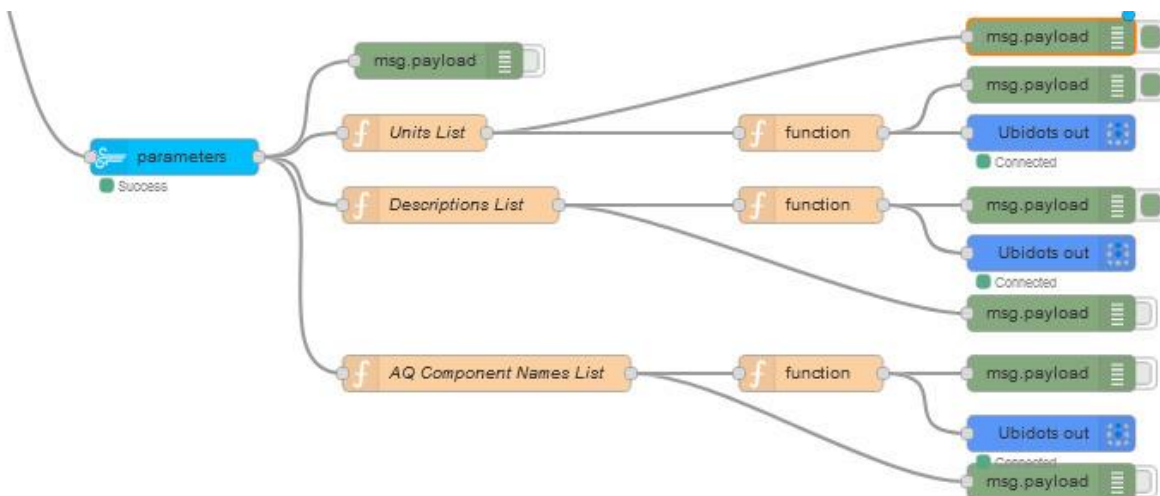


Working of an MQTT Protocol

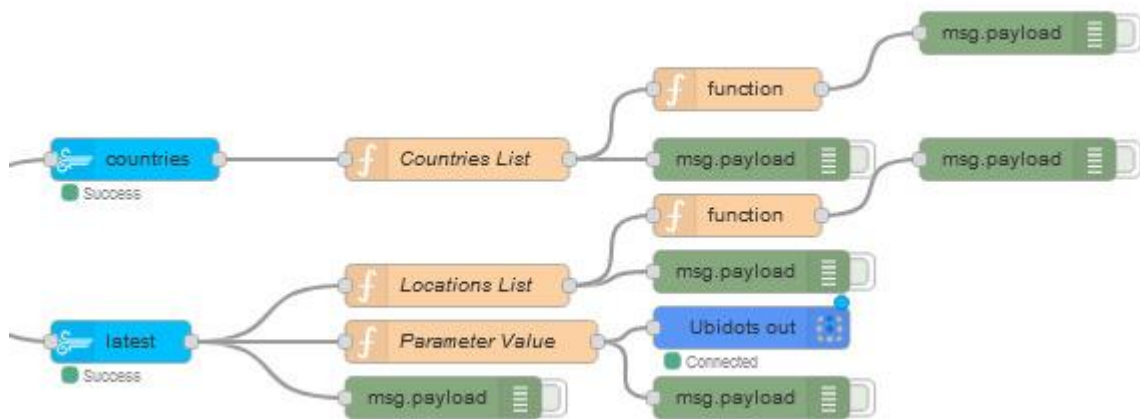
Database to Ubidots



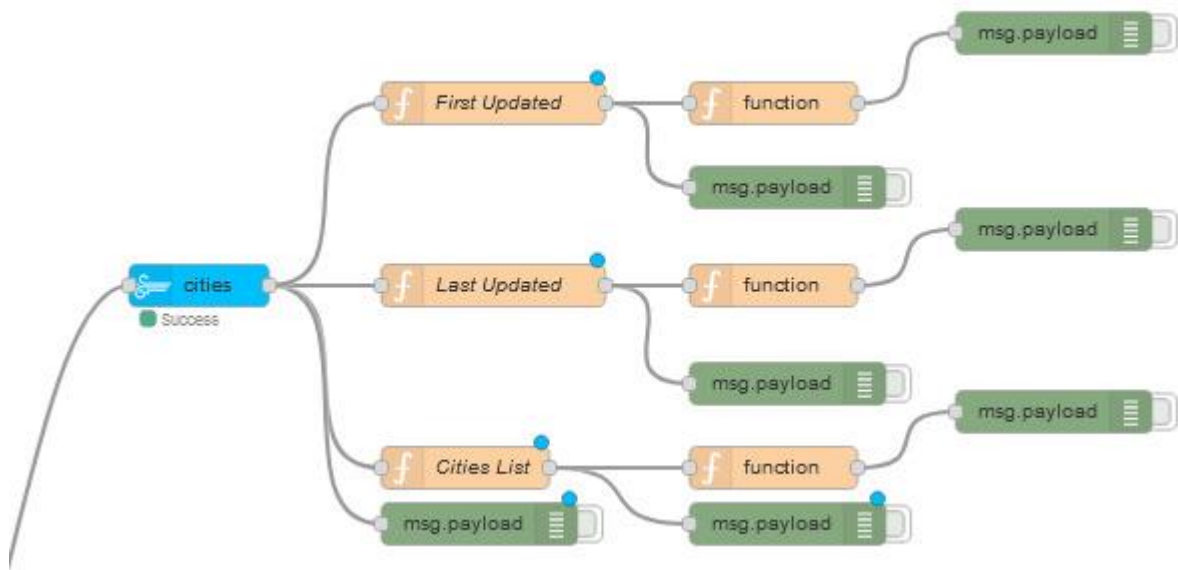
Firstly, we use the timestamp node and connect it to the nodes from the openaq API.



The parameter node is used to perform three functions i.e. to show the list of units, the description of the result and the names of the AQ components. The second function node is used to fetch the published value to ubidots.

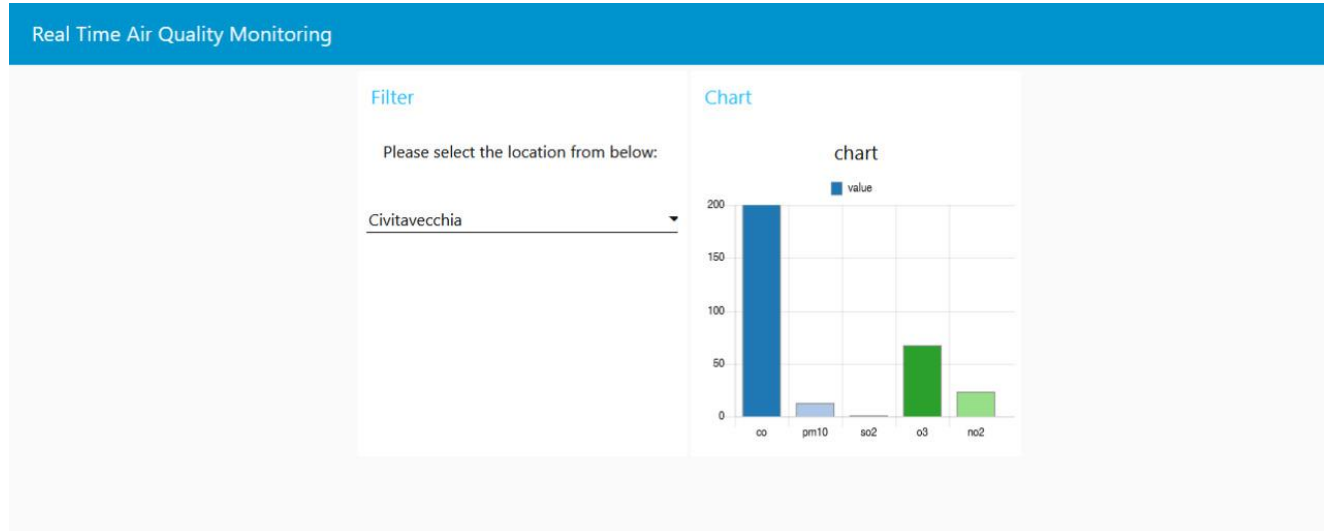


The countries node passes a list of countries to the Countries List function whereas the latest node gives information about the locations and parameter values in the API.

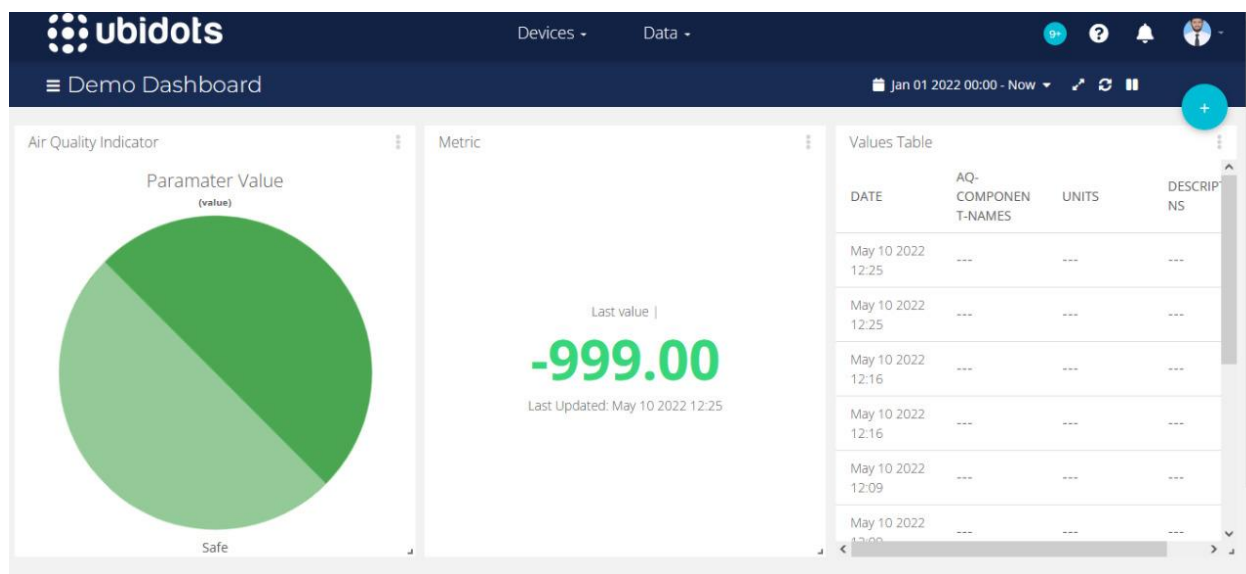


The cities node gives the information on when the value was first updated, when it was last updated and the list of the cities present.

Dashboard & Results:



Node Red Dashboard



Ubidots Dashboard

Why have we created two different dashboards?

The reason for creating two different dashboards is that Node Red provides a lot of customization and options for its dashboard also the same dashboard can be remotely connected and accessed through the mobile app, this is missing in UBI dots but UBI dots has its own advantages that it provides different actuators for data visualization which is missing in Node Red.

References:

1. <https://ieeexplore.ieee.org/document/8523891>
2. <https://openaq.org/#/>
3. <https://www.coursera.org/learn/realtime-data-visualization-dashboard-using-node-red/home/welcome>
4. <https://nodered.org/>
5. <https://randomnerdtutorials.com/what-is-mqtt-and-how-it-works/>
6. <https://flows.nodered.org/node/node-red-contrib-openaq>