

Documentation: Telegram AQI + Weather Bot (n8n Workflow)

Developed By: Khalid Imtiaz

1. Overview

This workflow enables a Telegram bot to provide real-time **Air Quality Index (AQI)** and **Weather** information for any city typed by the user.

Input: City name (sent via Telegram message)

Output: A formatted message containing:

- AQI value + category + health advice
- Weather information (temperature, humidity, condition)

Platforms & APIs Used:

- Telegram Bot
- n8n
- WAQI API (Air Quality data)
- OpenWeatherMap API (Weather data)

2. Workflow Steps

Step 1: Telegram Trigger

Purpose: Listens for messages received in Telegram.

Configuration:

- Trigger Type: **Message**
 - Extracted Data:
 - `chatId` (used later to reply)
 - city text (user input)
-

Step 2: Function Node — Extract City & chatId

Purpose: Clean city name text and pass chatId forward.

Code:

```
// Get city from Telegram message text
let city = $json.message.text || "";

// Clean formatting
city = city.trim().toLowerCase();
city = city.charAt(0).toUpperCase() + city.slice(1);

return [
  {
    json: {
      city
    }
  }
];
```

Step 3: HTTP Request Node — AQI Data

Purpose: Retrieve AQI data for the city.

Request URL:

`https://api.waqi.info/feed/{{$json.city}}/?token=YOUR_AQI_TOKEN`

Method: GET

Output: AQI value, city name, status

Step 4: HTTP Request Node — Weather Data

Purpose: Retrieve weather data for the same city.

Request URL:

```
https://api.openweathermap.org/data/2.5/weather?q={{$json.city}}&appid=YOUR_WEATHER_KEY&units=metric
```

Method: GET

Output: Temperature, humidity, weather description

Step 5: Merge Node

Purpose: Combine AQI and Weather data.

Mode: *Wait for Both Inputs*

- Ensure that **chatId** continues through the workflow.
-

Step 6: Function Node — Format Final Message

Purpose: Format AQI + Weather into a clean Telegram message.

Code:

```
const aqiData = $items("AQI Request")[0].json;           // Name must
match your node
const weatherData = $items("Weather Request")[0].json;

// Extract AQI
const aqi = aqiData?.data?.aqi || "N/A";

// AQI Category & Advice
function getAQICategory(aqi) {
```

```

    if (aqi <= 50) return {cat: "Good 😊", adv: "The air quality is good. Outdoor activities are safe."};
    if (aqi <= 100) return {cat: "Moderate 😐", adv: "Air quality is acceptable. Sensitive individuals should be cautious."};
    if (aqi <= 150) return {cat: "Unhealthy for Sensitive Groups 😔", adv: "Sensitive people should wear masks outdoors."};
    if (aqi <= 200) return {cat: "Unhealthy 😵", adv: "Limit outdoor exposure and consider wearing a mask."};
    if (aqi <= 300) return {cat: "Very Unhealthy 😱", adv: "Avoid going outside and keep windows closed."};
    return {cat: "Hazardous 🚫", adv: "Do not go outside. Take immediate health safety precautions."};
}

const {cat, adv} = getAQICategory(aqi);

// Weather Data
const temp = weatherData.main?.temp || "N/A";
const humidity = weatherData.main?.humidity || "N/A";
const desc = weatherData.weather?[0]?.description || "N/A";

// Determine City Name
const cityName = aqiData?.data?.city?.name || weatherData?.name || "Unknown City";

const message = `
🌐 *City:* ${cityName}

weathermap *Air Quality Index (AQI):* ${aqi}
• *Status:* ${cat}
• *Advice:* ${adv}

weathermap *Weather Status:*
• *Temperature:* ${temp}°C
• *Humidity:* ${humidity}%
• *Condition:* ${desc}

Stay safe and take care of your health!

```

```
`;

return [
{
  json: {
    chatId: $json.chatId,
    message
  }
}
];

```

Output Example:

```
{
  "chatId": 123456789,
  "message": "🌐 City: Lahore\nAQI: 34 (Good 😊)\nTemp: 15.9°C | Humidity: 51% | Condition: haze\nStay safe!"
}
```

Step 7: Telegram Send Message

Purpose: Send the formatted message back to the user.

Field	Value
Chat ID	<code>{{\$json.chatId}}</code>
Text	<code>{{\$json.message}}</code>
Parse Mode	Markdown

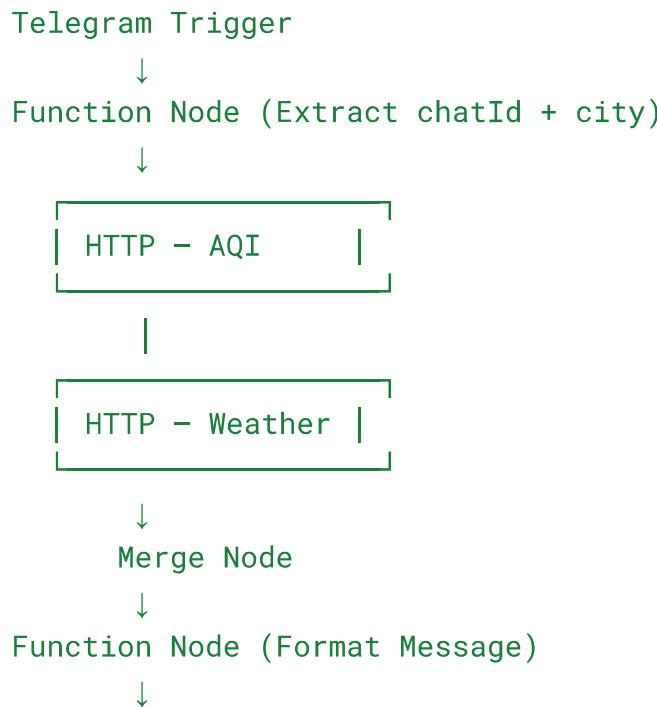
3. Important Notes

Note	Explanation
Chat ID must be preserved	Merge node can remove it — ensure it stays in the flow.
API Keys required	Replace <code>YOUR_AQI_TOKEN</code> and <code>YOUR_WEATHER_KEY</code> .
Error Handling	If data missing → return fallback health message.

Example fallback:

Air quality data for this city is currently unavailable. Please take general precautions.

4. Workflow Diagram (Simplified)



5. Features

- Any city supported
 - AQI + health advice
 - Weather conditions displayed clearly
 - Professional formatting with emojis
 - Robust against missing API data
-

6. Optional Enhancements

Enhancement	Description
City Quick Buttons	Show city list buttons to speed up input
/aqi Lahore Command	Add slash command support
Daily Notifications	Automatically send updates at specific time
Google Sheets Logging	Store all requests & analytics