# MQTT OPEN WEATHER API DOCS

BASICS FOR USING THE MQTT OPEN WEATHER API

## What is this?

This is a small project that can be used to interact with the OPEN WEATHER API through a MQTT implementation. MQTT messages are sent to the broker, which get interpreted by a simple python script that turns them into an HTTP request to the OPEN WEATHER servers. Right after that, the request response is sent back.

Shortened URL for the docs: shorturl.at/akyES

## Resources needed

- MQTT implementation to send/receive messages.
- Internet connection
- Dictionary object. Must be convertible to Python-like dict JSON.

## Basic usage

- Connect to the MQTT Broker:
  - Port 1883 and host “test.mosquitto.org”.
  - Port 1883 and host “ec2-100-27-23-78.compute-1.amazonaws.com”. Cloud-base deployment.
- Send API requests to the “reqs/XXXX” topic. XXXX Can be substituted to any valid string to be used as a topic name. For info on the request format, see the REQUEST section.
- Subscribe to the “reqs/XXXX/resp” topic. Responses will be sent to this topic.
- Read the “res” attribute from the JSON response in the “reqs/XXXX/resp”.

## (slightly more) Detailed usage

- Connect to the MQTT Broker:
  - For example test.mosquitto.org:1883. A custom implementation can be made by downloading (and lightly modifying) the python script and setting up a MQTT server. An API KEY for the Open weather API is needed. More info at: https://openweathermap.org/appid.
- Build a request.
  - A python-like dictionary JSON object is needed for the script to parse the parameters for the API call. The easiest way is to use Python, but it will work as long as the jsonification of the object has the same result. **ALL VALUES**

**MUST BE STRINGS. OTHERWISE THE SCRIPT MIGHT NOT PARSE THE VALUES PROPERLY.**

- Send the request as a stringified JSON object (the python dictionary, for example) to whichever subtopic of the “reqs” topic you’d like. Take into account that the request response will be returned in the “reqs/YOUR_TOPIC_NAME/resp/” topic.
- Wait for the response (maybe blink a couple of times, or take a breath 🤷)
    - You must be previously subscribed to the response topic.
    - In cases where the MQTT API is being used a lot (AKA reaching the limit of the free OPEN WEATHER subscription plan), the requests might **not** be sent to OPEN WEATHER API. If so, a message warning about a stressed MQTT API will be sent instead of the request response. Try again later!
- Parse the request response.
    - For ease of implementation, a stringified python dictionary object with a “res” attribute contains the response from the OPEN WEATHER API.
    - **NOT ALL THE RESPONSES MIGHT BE VALID.** Although a best effort is done to accomplish a implementation with no faults, bugs and small mistakes do sometime happen. Please, check the vailidty of the response and do not assume it always works. Some example responses can be found on the corresponding OPEN WEATHER MAP API DOCUMENTATION call (https://openweathermap.org/current).
- Done! You now have the same data you’d get from making a manual call to the OPEN WEATHER API.

# Request parameters.

In this section a list of all available parameters for each of the possible API calls is provided. Adding more parameters or minor changes might make it for the API not to work properly. As a reminder. **ALL. PARAMETERS. MUST. BE. STRINGS.** Other types might make the API not behave itself.

## API Call identification

In order for the script to know to which API to send the request to, a “CallType” parameter MUST BE DEFINED in the request. It must have one of the following values: “CurrentWeather”, ”OneTime”, ”AirPollution”, ”Geocoding”. According to this value, the API endpoint is defined. For more info about the API calls, look at https://openweathermap.org/price.

## Current weather API calls

This section portrails all the possible parameters for each of the four types of Current Weather API calls. MAX of 23 requests per minute, between ALL USERS (AKA for one Open Weather API key) and shared with the Air Pollution API calls and the Geocoding API calls.

## Common parameters

Parameters that can be defined in all the API calls for the Current Weather API.

- type: Specifies which type of the Current Weather API call to use. Must be one of the following: “city”, ”id”, ”geo”, ”zip”. Further parameters need to be defined according to the selected type. Further details explained in other subsections of the “Current Weather API Calls” section. **REQUIRED.**
- mode: Specifies the format in which to receive the data. Must be one of the following: “JSON”, “xml”, “html”. **OPTIONAL**. Default: JSON.
- units: Specifies the units in which to receive the data. Must be one of the following: “standard”, “metric”, “imperial”. **OPTIONAL**. Default: standard.
- lang: Specifies the language in which to receive the data. Must be an international language code. For more information see https://openweathermap.org/current#multi. **OPTIONAL**. Default: en.

## City Name Calls

In this section all the non-common parameters are discussed for the City Name Current Weather API calls (mode == city).

- city_name: Name of the city to search for, there are approx 200000 predefined locations. Can be more granular using both state_code and country_code parameters. **REQUIRED**.
- state_code: ISO 3166 code of the state or province in which to search for the city. For more info about ISO 3166 codes look at this online resource. https://www.iso.org/obp/ui/. **OPTIONAL**. REQUIRES city_name.
- country_code: ISO 3166 code of the country in which the state/province is located. For more info about ISO 3166 codes look at this online resource. https://www.iso.org/obp/ui/. **OPTIONAL**. REQUIRES state_code AND city_name.

## City ID Calls

In this section all the non-common parameters are discussed for the City ID Current Weather API calls (mode == id).

- id. City ID, must be one of OPEN WEATHER’s list of city ID 'city.list.json.gz'. **REQUIRED**.

## Geographic Coordinates Calls

In this section all the non-common parameters are discussed for the Geographic Coordinates Current Weather API calls (mode == geo).

- lat: Latitud of the point to get data for. Must be in decimal notation (point to separate the decimal value). **REQUIRED**.
- lon: Longitude of the point to get data for. Must be in decimal notation (point to separate the decimal value). **REQUIRED**.

### ZIP Code Calls

In this section all the non-common parameters are discussed for the City Name Current Weather API calls (mode == zip).

- zip: Zip code to search for. **REQUIRED**.
- country_code: ISO 3166 code of the country in which the stat. For more info about ISO 3166 codes look at this online resource. https://www.iso.org/obp/ui/. **REQUIRED**.

## One Time API calls

This section portrails all the possible parameters for the One Time API call. At max, ONE request every 90 seconds can be done.

### Common parameters

Parameters that can be defined in all the API calls for the One Time API.

- type: Specifies which type of the One Time API call to use. Must be one of the following: “non_historical”, ”historical”. Further parameters need to be defined according to the selected type. Further details explained in other subsections of the “One Time API Calls” section. **REQUIRED.**
- lat: Latitud of the point to get data for. Must be in decimal notation (point to separate the decimal value). **REQUIRED**.
- lon: Longitude of the point to get data for. Must be in decimal notation (point to separate the decimal value). **REQUIRED**.
- units: Specifies the units in which to receive the data. Must be one of the following: “standard”, “metric”, “imperial”. **OPTIONAL**. Default: standard.
- lang: Specifies the language in which to receive the data. Must be an international language code. For more information see https://openweathermap.org/current#multi. **OPTIONAL**. Default: en.

### Historical Calls

In this section all the non-common parameters are discussed for the Historical One Time API calls (mode == historical).

- dt: Date from the previous five days (UNIX time, UTC time zone), for which to get the data. **REQUIRED**.

### Non-Historical Calls

In this section all the non-common parameters are discussed for the Non-Historical One Time API calls (mode == non_historical).

- exclude: List of comma separated values that marks which date you DO NOT WANT to receive from the One Time API call. Possible values to be contained in the list are: “minutely”,”hourly”,”daily”,”current”,”alerts”. **OPTIONAL**. Default: EMPTY LIST.

## Air Pollution API calls

This section portrails all the possible parameters for each of the three types of Air Pollution API calls. MAX of 23 requests per minute, between ALL USERS (AKA for one Open Weather API key) and shared with the Current Weather API calls and the Geocoding API calls.

### Common parameters

Parameters that can be defined in all the API calls for the Air Pollution API.

- type: specifies which type of the Air Pollution API call to use. Must be one of the following: “current”, “forecast”, “history”. Further parameters need to be defined according to the selected type. Further details explained in other subsections of the “Air Pollution API Calls” section. **REQUIRED.**
- lat: Latitud of the point to get data for. Must be in decimal notation (point to separate the decimal value). **REQUIRED**.
- lon: Longitude of the point to get data for. Must be in decimal notation (point to separate the decimal value). **REQUIRED**.

### Current Calls

In this section all the non-common parameters are discussed for the Current Air Pollution API calls (mode == current).
Purposely left empty.

### Forecast Calls

In this section all the non-common parameters are discussed for the Forecast Air Pollution API calls (mode == forecast).
Purposely left empty.

### Historical Calls

In this section all the non-common parameters are discussed for the Historical Air Pollution API calls (mode == history).

- start: Start date (UNIX time, UTC time zone). **REQUIRED**.
- end: End date (UNIX time, UTC time zone). **REQUIRED**.

## Geocoding API calls

This section portrails all the possible parameters for each of the three types of Geocoding API calls. MAX of 23 requests per minute, between ALL USERS (AKA for one Open Weather API key) and shared with the Air Pollution API calls and the Current Weather API calls.

### Common parameters

Parameters that can be defined in all the API calls for the Geocoding API.

- type: specifies which type of the Geocoding API call to use. Must be one of the following: “reverse”, “direct_name”, “direct_zip”. Further parameters need to be defined according to the selected type. Further details explained in other subsections of the “Geocoding API Calls” section. **REQUIRED.**

- limit: specifies the maximum amount of options a Geocoding request can get. **OPTIONAL.** Default: 1.

### Reverse Calls

In this section all the non-common parameters are discussed for the Reverse Geocoding API calls (mode == reverse).

- lat: Latitud of the point to get data for. Must be in decimal notation (point to separate the decimal value). **REQUIRED**.
- lon: Longitude of the point to get data for. Must be in decimal notation (point to separate the decimal value). **REQUIRED**.

### Direct Name Calls

In this section all the non-common parameters are discussed for the Direct Name Geocoding API calls (mode == direct_name).

- city_name: Name of the city to search for, there are approx 200000 predefined locations. Can be more granular using both state_code and country_code parameters. **REQUIRED**.
- state_code: ISO 3166 code of the state or province in which to search for the city. For more info about ISO 3166 codes look at this online resource. https://www.iso.org/obp/ui/. **OPTIONAL**. REQUIRES city_name.
- country_code: ISO 3166 code of the country in which the state/province is located. For more info about ISO 3166 codes look at this online resource. https://www.iso.org/obp/ui/. **OPTIONAL**. REQUIRES state_code AND city_name.

### Direct ZIP Calls

In this section all the non-common parameters are discussed for the Direct ZIP Geocoding API calls (mode == direct_zip).

- zip: Zip code to search for the coordinates. **REQUIRED**.
- country_code: ISO 3166 code of the country in which the state/province is located. For more info about ISO 3166 codes look at this online resource. https://www.iso.org/obp/ui/. **REQUIRED**.