Open Weather One Call
Copyright 2020 J. Hershey
v1.3.1

Thank you for your interest in the **OpenWeatherOneCall** library for ESP32. This library streamlines the ability to gather information from the OpenWeatherMap website. There are two items you will need to get started:

OPENWEATHERMAP API KEY
(optional for geolocation via WiFi)
GOOGLE DEVELOPER API KEY

You can get those at **www.openweathermap.org** and **Google's developer website**. Once you have those using the library is as simple as providing the calling function with the following items:

OpenWeather Key, Google Key, Latitude, Longitude, Unit type, City ID, Excludes

Calling the weather forecast you only need call:

parseWeather(OpenWeather Key, Google Key, Latitude, longitude, Unit type, CIty ID,Excludes)

OpenWeather Key is self-explanatory
Google Key is self-explanatory

Latitude and Longitude (if known, otherwise NULL) If you are using a GPS simply use those coordinates for Latitude and Longitude.

Unit type is a boolean (true/false/blank) (METRIC, IMPERIAL, KELVIN)

CITY ID is now active and can be used if known, otherwise NULL.

If an invalid coordinate is sent (out of range of worldwide measurements) geolocation will take over and gather your current location based on WiFi triangulation. Forcing an out of bounds Latitude or Longitude ensures geolocation will be used.

Sending an invalid Google Key (NULL) will ensure coordinates are used.

If none of those are valid a return error code of -1 will take place.

Exclude options are CURRENT, DAILY, HOURLY, MINUTELY, ALERTS
And they are entered in the API CALL as **EXCL_C+EXCL_D+EXCL_H+_EXCL_M+EXCL_A**
You only need use the items you wish to exclude, this will save memory used in the parsing program by allocating only what is required for the incoming information.

As an example of what type of call to make:

parseWeather(OpenWeatherKey,NULL,Latitude,Longitude,true,NULL,EXCL_A+EXCL_M)

The above call will use YOUR gps or other manually controlled coordinates to gather weather for THAT location and returns info in METRIC units, the data for Minutely and Alerts is also excluded from the generated JSON file at OpenWeatherMap.

So if you are in St. Louis, Missouri and you know the coordinates for Los Angeles, California, use those coordinates to get the weather in LA.

parseWeather(OpenWeatherKey,NULL,NULL,NULL,true,city_id,NULL)

The above call will use a CITY ID to gather weather for THAT location, or within 20 miles and returns info in METRIC units, no excludes will be sent so ALL information available will be returned.

So if you are in St. Louis, Missouri and you know the City ID for Los Angeles, California (5368361), use that to get the weather in L.A.

parseWeather(OpenWeatherKey,GoogleKey,NULL,NULL,false,NULL,EXCL_C)

The above call will use your current location based on WiFi triangulation and returns information in IMPERIAL units and exclude CURRENT weather information (but will send back hourly, minutely, daily, and alerts).

Including a Google Key in **ANY** call will return your current location weather and will **override** latitude and longitude of another location, so be sure you are only using one of the location choices (**Google Key, City ID,** or **Latitude and Longitude**) based on your needs. ALL three do NOT need to be used.

PLEASE SEE SEPARATE DOCUMENT FOR VARIABLES AND CALLING THEM

As you can see, it couldn’t be easier to get weather information for any day, any location up to 7 days into the future.

Please **examine the examples** available to see just how to send the required pieces of information to the library. As always, if you have any questions, the creator is available on GitHub. Thanks for your interest in the library!