

SPRINT - 01

Date :	30 October 2022
Team ID :	PNT2022TMID47905
Project Name	SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

SPRINT GOALS:

1. Create and initialize accounts in various public APIs like Open Weather API.
2. Write a Python program that outputs results given the inputs like weather and location.

The screenshot shows the OpenWeather API website. The browser tabs include GitHub, IBM-EPBL/IBM-Project-31637-10, Current weather and forecast - C, and Weather API - OpenWeatherMap. The website header has the OpenWeather logo and a search bar. The main content area is titled 'Weather API' and includes a navigation bar with links like Guide, API, Dashboard, Marketplace, Pricing, Maps, Our Initiatives, Partners, Blog, For Business, jerome, and Support. The 'One Call API 3.0' section is highlighted, featuring a 'Subscribe' button and a list of features: Minute forecast for 1 hour, Hourly forecast for 48 hours, Daily forecast for 8 days, Historical data for 40+ years back by timestamp, and National weather alerts. The pricing is listed as '1,000 API calls per day for free' and '0.0012 GBP per API call over the daily limit'. A 'Subscribe to One Call by Call' button is also present. Below this, the 'Professional collections' section is visible, mentioning 'Current & Forecasts collection', 'Historical weather data collection', and 'Weather Maps collection'. The bottom of the screenshot shows a Windows taskbar with various application icons and system tray information.

GitHubIBM-EPBL/IBM-Project-31637-10Current weather and forecast - CWeather API - OpenWeatherMapOne Call API 3.0 - OpenWeather+

openweathermap.org/api/one-call-3

AppsJeromeSoftwareRechargeTicketsShopSocialIBMYouTubeMy Drive - Google...Google MeetGoogle DocsLightning Adapter...

OpenWeatherWeather in your cityGuideAPIDashboardMarketplacePricingMapsOur InitiativesPartnersBlogFor BusinessjeromeSupport

Current and forecast weather data

To get access to current weather, minute forecast for 1 hour, hourly forecast for 48 hours, daily forecast for 8 days and government weather alerts, please use this section of the documentation.

If you are interested in **historical weather data**, please read the "Historical weather data" section.

How to make an API call

API call

https://api.openweathermap.org/data/3.0/onecall?lat={lat}&lon={lon}&exclude={part}&appid={API key}

Parameters

lat, lon	required	Geographical coordinates (latitude, longitude). If you need the geocoder to automatic convert city names and zip-codes to geo coordinates and the other way around, please use our Geocoding API .
appid	required	Your unique API key (you can always find it on your account page under the "API key" tab)
exclude	optional	By using this parameter you can exclude some parts of the weather data from the API response. It should be a comma-delimited list (without spaces). Available values: <ul style="list-style-type: none">currentminutelyhourlydaily

27°C
Haze

Search

23 KB17 KBENG21/11/2021

- Current and forecasts weather data
 - How to make an API call
 - Example of API response
 - Fields in API response
- Historical weather data
 - How to make an API call
 - Example of API response
 - Fields in API response
 - List of weather condition codes
- Other features
 - Units of measurement
 - Multilingual support
 - List of national weather alerts sources
 - Call back function for JavaScript code

27°C
Haze

Search

23 KB17 KBENG21/11/2021

GitHubIBM-EPBL/IBM-Project-31637-10Current weather and forecast - CWeather API - OpenWeatherMapOne Call API 3.0 - OpenWeatherhttps://api.openweathermap.org/

api.openweathermap.org/data/2.5/weather?lat=9.9333&lon=78.1167&appid=5b470269921e324575f30d836ad5aef

AppsJeromeSoftwareRechargeTicketsShopSocialIBMYouTubeMy Drive - Google...Google MeetGoogle DocsLightning Adapter...

```
{ "coord": { "lon": 78.1167, "lat": 9.9333 }, "weather": [ { "id": 721, "main": "Haze", "description": "haze", "icon": "50n" } ], "base": { "stations": { "main": { "temp": 300.16, "feels_like": 301.94, "temp_min": 300.16, "temp_max": 300.16, "pressure": 1008, "humidity": 69, "visibility": 3000, "wind": { "speed": 1.03, "deg": 0 }, "clouds": { "all": 140 }, "dt": 1669036491, "sys": { "type": 1, "id": 9216, "country": "IN", "sunrise": 1668991463, "sunset": 1669033354, "timezone": 19000, "id": 1264521, "name": "Madurai", "cod": 200 } } } }
```

MADURAI- WEATHER API

```
{"coord":{"lon":78.1167,"lat":9.9333},"weather":[{"id":721,"main":"Haze","description":"haze","icon":"50n"}],"base":"stations","main":{"temp":300.16,"feels_like":301.94,"temp_min":300.16,"temp_max":300.16,"pressure":1008,"humidity":69},"visibility":3000,"wind":{"speed":1.03,"deg":0},"clouds":{"all":40},"dt":1669036491,"sys":{"type":1,"id":9216,"country":"IN","sunrise":1668991463,"sunset":1669033354},"timezone":19800,"id":1264521,"name":"Madurai","cod":200}
```

SPRINT 01

<https://api.openweathermap.org/data/2.5/weather?lat=9.9333&lon=78.1167&appid=5b470269921e324575ff30d836ad5aef>

PYTHON CODE

```
import requests
api_data = "
https://api.openweathermap.org/data/2.5/weather?lat=9.9333&lon=78.1167&appid=5b470269921e324575ff30d836ad5aef"
rec=requests.get(url=api_data)
data= rec.json()
print(data)
temp = data['main']['temp']
print("\nTemperature is : ", temp)
humidity = data['main']['humidity']
print("Humidity is : ", humidity)
```

// This code execution will be done in sprint 02