

Sign in

Get started

Follow

600K Followers

· Editors' Picks

Features

Deep Dives

Grow

Contribut

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

Weather Data and Forecasts from Open Weather API



Joseph Magiya · Apr 10, 2020 · 7 min read ★

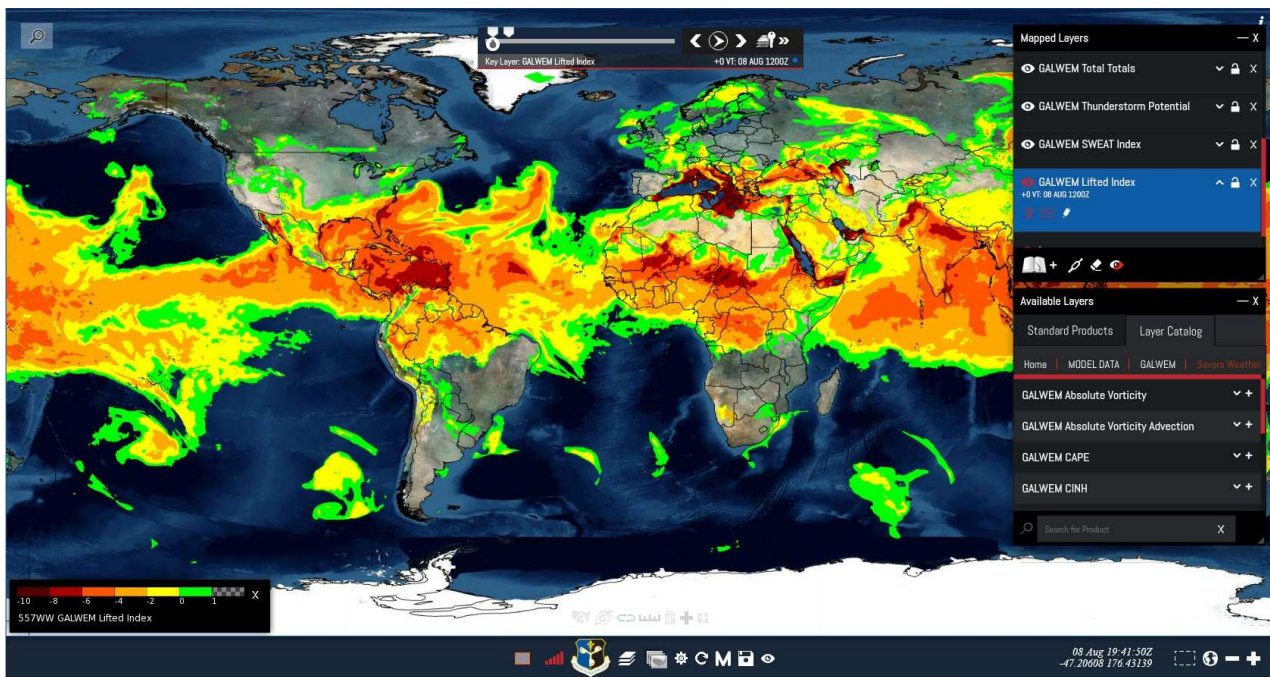


Photo By: Benjamin Newell on [afcent](#)

Adding weather to your analysis might bring something to light, or it might not. Either way, it is easier to explore and add this to your data now than ever before.

It is simpler than you think — let me show you how.

Introduction

We'll get our data from OpenWeather and use their API to get that data.

For this you'll need to sign up and confirm your account. Once you do all that, you'll receive an API Key which you'll use to fetch the data through API calls.

Current Weather Using City Name

This is the format of the API call using a city name —

*api.openweathermap.org/data/2.5/weather?q={city name},{state},
{country code}&appid={your api key}*

I've used — “b6907d289e10d714a6e88b30761fae22” as an example of an api key — but you need to change it to the one you receive from OpenWeather, this will not work.

```
City_API_endpoint = "http://api.openweathermap.org/data/2.5  
/weather?q="
City = "Nairobi"
Country = ",KE,"
join_key = "&appid=" + "b6907d289e10d714a6e88b30761fae22"
units = "&units=metric"
```

This is all we need to form our request URL, now we need to combine them to actually form a URL.

```
current_city_weather = City_API_endpoint + City + Country + join_key  
+ units  
print(current_city_weather)
```

Result

<http://api.openweathermap.org/data/2.5/weather?q=Nairobi,KE,&appid=b6907d289e10d714a6e88b30761fae22&units=metric>

If you go to that link, you'll see something like this, ofcourse if you're reading this in the future or even in the past — you'll get a different result.

```
{
  "coord": {
    "lon": 36.82,
    "lat": -1.28
  },
  "weather": [
    {
      "id": 801,
      "main": "Clouds",
      "description": "few clouds",
      "icon": "02n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 19.51,
    "feels_like": 17.72,
    "temp_min": 16.67,
    "temp_max": 21,
    "pressure": 1021,
    "humidity": 68,
    "visibility": 10000,
    "wind": {
      "speed": 4.1,
      "deg": 90
    },
    "clouds": {
      "all": 20
    },
    "dt": 1586456143,
    "sys": {
      "type": 1,
      "id": 2558,
      "country": "KE",
      "sunrise": 1586403081,
      "sunset": 1586446600,
      "timezone": 10800,
      "id": 184745,
      "name": "Nairobi",
      "cod": 200
    }
  }
}
```

Current Weather by City Name JSON Result

Before we go into how to extract the data and make it useful, let's look at the second option.

Current Weather Using Coordinates (Latitude and Longitudes)

This is my favorite one especially since I get to interact with coordinates a lot.

This is the format of the API call for a specific location using latitude and longitude — *api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={your api key}*

```
API_endpoint = "http://api.openweathermap.org"
join_1 = "&appid="
API_key = "b6907d289e10d714a6e88b30761fae22"

current_weather_lat_lon = "/data/2.5/weather?" + "lat=" +
str(round(lat,2)) + "&lon=" + str(round(long,2))
```

Let's set the lat and long in variables, you can definitely pass multiple values e.g. a dataframe you're analysing (I'll cover this later...maybe in a separate

blog).

```
lat = -1.2999792
long = 36.7728897
lat_long = "lat=" + str(lat)+ "&lon=" + str(long)
```

Let's add the coordinates and other parts to the URL and create the full URL. You can obviously do this in one line, but I prefer my code being flexible and easy to understand.

```
coord_API_endpoint = "http://api.openweathermap.org/data/2.5/
/weather?"
lat_long = "lat=" + str(lat)+ "&lon=" + str(long)
join_key = "&appid=" + "b6907d289e10d714a6e88b30761fae22"
units = "&units=metric"
```

Full URL

```
current_coord_weather_url= coord_API_endpoint + lat_long + join_key +
units
print(current_coord_weather_url)
```

Result

<http://api.openweathermap.org/data/2.5/weather?lat=-1.2999792&lon=36.7728897&appid=b6907d289e10d714a6e88b30761fae22&units=metric>

```
{
  "coord": {
    "lon": 36.77,
    "lat": -1.3
  },
  "weather": [
    {
      "id": 801,
      "main": "Clouds",
      "description": "few clouds",
      "icon": "02n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 19.45,
    "feels_like": 17.64,
    "temp_min": 16.67,
    "temp_max": 21,
    "pressure": 1021,
    "humidity": 68,
    "visibility": 10000,
    "wind": {
      "speed": 4.1,
      "deg": 90
    },
    "clouds": {
      "all": 20
    },
    "dt": 1586456886,
    "sys": {
      "type": 1,
      "id": 2558,
      "country": "KE",
      "sunrise": 1586409094,
      "sunset": 1586446611,
      "timezone": 10800,
      "id": 184745,
      "name": "Nairobi",
      "cod": 200
    }
  }
}
```

Current Weather by Coordinates JSON Result

Requesting for the Current Weather Data

For this we'll need to import a python package — requests.

```
import requests
```

Here's how we get the result of the response into a variable, which we'll later break down so it's more readable.

```
json_data = requests.get(current_coord_weather_url).json()  
print(json_data)
```

The result of this will have the same structure as the link you went to,

```
In [13]: print(json_data)  
{  
  'coord': {'lon': 36.82, 'lat': -1.28},  
  'weather': [{'id': 801, 'main': 'Clouds', 'description': 'few clouds', 'icon': '02n'}],  
  'base': 'stations',  
  'main': {'temp': 19.49, 'feels like': 17.69, 'temp_min': 16.67, 'temp_max': 21, 'pressure': 1021,  
    'humidity': 68},  
  'visibility': 10000,  
  'wind': {'speed': 4.1, 'deg': 90},  
  'clouds': {'all': 20},  
  'dt': 1586457185, 'sys': {'type': 1, 'id': 2558, 'country': 'KE', 'sunrise': 1586403081, 'sunset': 1586446600, 'timezone': 10800, 'id': 184745, 'name': 'Nairobi', 'cod': 200}
```

Putting The Current Weather Data into a Pandas Data Frame

We'll obviously need to import pandas

```
import pandas as pd
```

Let's create an empty data frame first, we'll later add the values here.

```
df_all_current_weather = pd.DataFrame()
```

Next we create empty lists for the keys in the dataframe.

```
# Create empty lists to store the JSON Data
current_weather_id = []
current_time = []
own_city_id = []
city = []
latitude = []
longitude = []
country = []
timezone = []
sunrise = []
sunset = []
temperature = []
temperature_feel = []
temperature_min = []
temperature_max = []
pressure = []
humidity = []
main = []
main_description = []
clouds = []
wind_speed = []
wind_degree = []
visibility = []
```

Then we add the data from the JSON into the list:

```
# Add JSON Data to the lists
prediction_num +=1
current_weather_id.append(prediction_num + 1)
current_time.append(pd.Timestamp.now())
own_city_id.append(json_data['id'])
city.append(json_data['name'])
latitude.append(json_data['coord']['lat'])
longitude.append(json_data['coord']['lon'])
country.append(json_data['sys']['country'])
if json_data['timezone'] >0 :
    timezone.append(("+" + str((json_data['timezone']/3600)))
else:
    timezone.append(((json_data['timezone']/3600))
sunrise.append(json_data['sys']['sunrise'])
sunset.append(json_data['sys']['sunset'])
temperature.append(json_data['main']['temp'])
temperature_feel.append(json_data['main']['feels_like'])
temperature_min.append(json_data['main']['temp_min'])
temperature_max.append(json_data['main']['temp_max'])
pressure.append(json_data['main']['pressure'])
```

```

humidity.append(json_data['main']['humidity'])
main.append(json_data['weather'][0]['main'])
main_description.append(json_data['weather'][0]['description'])
clouds.append(json_data['clouds']['all'])
wind_speed.append(json_data['wind']['speed'])
wind_degree.append(json_data['wind']['deg'])
visibility.append(json_data['visibility'])

```

Finally we write the list into the dataframe that we created

```

# Write Lists to DataFrame
df_all_current_weather['current_weather_id'] = current_weather_id
df_all_current_weather['current_time'] = current_time
df_all_current_weather['own_city_id'] = own_city_id
df_all_current_weather['city'] = city
df_all_current_weather['latitude'] = latitude
df_all_current_weather['longitude'] = longitude
df_all_current_weather['country'] = country
df_all_current_weather['timezone'] = timezone
df_all_current_weather['sunrise'] = sunrise
df_all_current_weather['sunset'] = sunset
df_all_current_weather['temperature'] = temperature
df_all_current_weather['temperature_feel'] = temperature_feel
df_all_current_weather['temperature_min'] = temperature_min
df_all_current_weather['temperature_max'] = temperature_max
df_all_current_weather['pressure'] = pressure
df_all_current_weather['humidity'] = humidity
df_all_current_weather['main'] = main
df_all_current_weather['main_description'] = main_description
df_all_current_weather['clouds'] = clouds
df_all_current_weather['wind_speed'] = wind_speed
df_all_current_weather['wind_degree'] = wind_degree
df_all_current_weather['visibility'] = visibility

```

Previewing the data frame will give you the result of the result of the API call but in a way that is more useful to analyse and join with your data.

In [21]: `df_all_current_weather.head(5)`

Out[21]:

	current_weather_id	current_time	own_city_id	city	latitude	longitude	country	timezone	sunrise	sunset	temperature	temperature_feel
0	2	2020-04-09 21:42:06.662081	184745	Nairobi	-1.28	36.82	KE	+3.0	1586403081	1586446600	19.49	17.69

Result

Forecasts

This is for those who want to know what tomorrow is going to look like.

We'll still use a few variables from the first section, and since we already know the basics, I'll shorten this section.

One thing to note is that the result is going to be a longer JSON, because it has the forecast in 3 hour intervals, 5 times. This is very beneficial for analysis and predictions.

Think about how this can affect some of your forecasting projects e.g. the amount of traffic in a city.

Forecasts by City Name

Forecasts API Call using City Name Format —

`api.openweathermap.org/data/2.5/forecast?q={city name},{state},{country code}&appid={your api key}`

```
City_API_endpoint = "http://api.openweathermap.org/data/2.5/forecast?q="
City = "Nairobi"
Country = ",KE,"
join_key = "&appid=" + "b6907d289e10d714a6e88b30761fae22"
units = "&units=metric"
```

```
city_forecast = City_API_endpoint + City + Country + join_key + units
```

```
print(city_forecast)
```

Result

`http://api.openweathermap.org/data/2.5/forecast?q=Nairobi,KE,&appid=b6907d289e10d714a6e88b30761fae22&units=metric`

Following the link — you'll find this, and this is just a part of it.

[illegible]

Forecast Result — By City Name

Forecasts by Coordinates

Forecasts API Call using Coordinates Format —

```
api.openweathermap.org/data/2.5/forecast?lat={lat}&lon={lon}&
appid={your api key}
```

```
lat = -1.2999792
long = 36.7728897
lat_long = "lat=" + str(lat)+ "&lon=" + str(long)
```

```
coord_API_endpoint = "http://api.openweathermap.org/data/2.5/forecast?"
lat_long = "lat=" + str(lat) + "&lon=" + str(long)
join_key = "&appid=" + "b6907d289e10d714a6e88b30761fae22"
units = "&units=metric"
```

```
forecast_coord_weather_url = coord_API_endpoint + lat_long + join_key
+ units
```

```
print (forecast coord weather url)
```

Result

<http://api.openweathermap.org/data/2.5/forecast?lat=-1.2999792&lon=36.7728897&appid=b6907d289e10d714a6e88b30761fae22&units=metric>

Following the link — you'll find this, very tiring to look at.

```
{
  "cod": "200",
  "message": 0,
  "cnt": 40,
  "list": [
    {
      "dt": "1586466000",
      "main": {
        "temp": 16.88,
        "feels_like": 15.5,
        "temp_min": 15.25,
        "temp_max": 16.88,
        "pressure": 1016,
        "sea_level": 1016,
        "grnd_level": 827,
        "humidity": 57,
        "temp_kf": 1.63,
        "weather": [
          {
            "id": 500,
            "main": "Rain",
            "description": "light rain",
            "icon": "10n"
          }
        ],
        "clouds": {
          "all": 65
        },
        "wind": {
          "speed": 1.41,
          "deg": 66
        },
        "rain": {
          "3h": 0.12
        },
        "sys": {
          "pod": "n"
        },
        "dt_txt": "2020-04-09 21:00:00"
      },
      "dt": "1586476800",
      "main": {
        "temp": 15.61,
        "feels_like": 14.17,
        "temp_min": 14.39,
        "temp_max": 15.61,
        "pressure": 1015,
        "sea_level": 1015,
        "grnd_level": 825,
        "humidity": 58,
        "temp_kf": 1.22,
        "weather": [
          {
            "id": 803,
            "main": "Clouds",
            "description": "broken clouds",
            "icon": "04n"
          }
        ],
        "clouds": {
          "all": 77
        },
        "wind": {
          "speed": 1.19,
          "deg": 45
        },
        "sys": {
          "pod": "n"
        },
        "dt_txt": "2020-04-10 00:00:00"
      },
      "dt": "1586487600",
      "main": {
        "temp": 14.64,
        "feels_like": 13.05,
        "temp_min": 13.82,
        "temp_max": 14.64,
        "pressure": 1017,
        "sea_level": 1017,
        "grnd_level": 827,
        "humidity": 58,
        "temp_kf": 0.82,
        "weather": [
          {
            "id": 803,
            "main": "Clouds",
            "description": "broken clouds",
            "icon": "04n"
          }
        ],
        "clouds": {
          "all": 81
        },
        "wind": {
          "speed": 1.1,
          "deg": 41
        },
        "sys": {
          "pod": "n"
        },
        "dt_txt": "2020-04-10 03:00:00"
      },
      "dt": "1586498400",
      "main": {
        "temp": 13.76,
        "feels_like": 12.08,
        "temp_min": 12.35,
        "temp_max": 13.76,
        "pressure": 1017,
        "sea_level": 1017,
        "grnd_level": 830,
        "humidity": 46,
        "temp_kf": 0.41,
        "weather": [
          {
            "id": 803,
            "main": "Clouds",
            "description": "broken clouds",
            "icon": "04n"
          }
        ],
        "clouds": {
          "all": 72
        },
        "wind": {
          "speed": 1.37,
          "deg": 93
        },
        "sys": {
          "pod": "d"
        },
        "dt_txt": "2020-04-10 06:00:00"
      },
      "dt": "1586509200",
      "main": {
        "temp": 23.86,
        "feels_like": 20.23,
        "temp_min": 23.86,
        "temp_max": 23.86,
        "pressure": 1015,
        "sea_level": 1015,
        "grnd_level": 831,
        "humidity": 30,
        "temp_kf": 0,
        "weather": [
          {
            "id": 802,
            "main": "Clouds",
            "description": "scattered clouds",
            "icon": "03d"
          }
        ],
        "clouds": {
          "all": 44
        },
        "wind": {
          "speed": 3.64,
          "deg": 118
        },
        "sys": {
          "pod": "d"
        },
        "dt_txt": "2020-04-10 09:00:00"
      },
      "dt": "1586520000",
      "main": {
        "temp": 24.91,
        "feels_like": 20.64,
        "temp_min": 24.91,
        "temp_max": 24.91,
        "pressure": 1011,
        "sea_level": 1011,
        "grnd_level": 826,
        "humidity": 29,
        "temp_kf": 0,
        "weather": [
          {
            "id": 802,
            "main": "Clouds",
            "description": "scattered clouds",
            "icon": "03d"
          }
        ],
        "clouds": {
          "all": 37
        },
        "wind": {
          "speed": 4.53,
          "deg": 117
        },
        "sys": {
          "pod": "d"
        },
        "dt_txt": "2020-04-10 12:00:00"
      },
      "dt": "1586530800",
      "main": {
        "temp": 21.61,
        "feels_like": 17.09,
        "temp_min": 21.61,
        "temp_max": 21.61,
        "pressure": 1012,
        "sea_level": 1012,
        "grnd_level": 828,
        "humidity": 39,
        "temp_kf": 0,
        "weather": [
          {
            "id": 801,
            "main": "Clouds",
            "description": "few clouds",
            "icon": "02d"
          }
        ],
        "clouds": {
          "all": 20
        },
        "wind": {
          "speed": 4.25,
          "deg": 103
        },
        "sys": {
          "pod": "d"
        },
        "dt_txt": "2020-04-10 15:00:00"
      },
      "dt": "1586541600",
      "main": {
        "temp": 16.52,
        "feels_like": 14.09,
        "temp_min": 16.52,
        "temp_max": 16.52,
        "pressure": 1016,
        "sea_level": 1016,
        "grnd_level": 827,
        "humidity": 54,
        "temp_kf": 0,
        "weather": [
          {
            "id": 802,
            "main": "Clouds",
            "description": "scattered clouds",
            "icon": "03n"
          }
        ],
        "clouds": {
          "all": 37
        },
        "wind": {
          "speed": 2.61,
          "deg": 84
        },
        "sys": {
          "pod": "n"
        },
        "dt_txt": "2020-04-10 18:00:00"
      },
      "dt": "1586552400",
      "main": {
        "temp": 15.27,
        "feels_like": 13.86,
        "temp_min": 15.27,
        "temp_max": 15.27,
        "pressure": 1016,
        "sea_level": 1016,
        "grnd_level": 827,
        "humidity": 58,
        "temp_kf": 0,
        "weather": [
          {
            "id": 803,
            "main": "Clouds",
            "description": "broken clouds",
            "icon": "04n"
          }
        ],
        "clouds": {
          "all": 55
        },
        "wind": {
          "speed": 1.04,
          "deg": 91
        },
        "sys": {
          "pod": "n"
        },
        "dt_txt": "2020-04-10 21:00:00"
      },
      "dt": "1586563200",
      "main": {
        "temp": 14.73,
        "feels_like": 13.27,
        "temp_min": 14.73,
        "temp_max": 14.73,
        "pressure": 1015,
        "sea_level": 1015,
        "grnd_level": 826,
        "humidity": 58,
        "temp_kf": 0,
        "weather": [
          {
            "id": 803,
            "main": "Clouds",
            "description": "broken clouds",
            "icon": "04n"
          }
        ],
        "clouds": {
          "all": 76
        },
        "wind": {
          "speed": 0.94,
          "deg": 51
        },
        "sys": {
          "pod": "n"
        },
        "dt_txt": "2020-04-11 00:00:00"
      },
      "dt": "1586574000",
      "main": {
        "temp": 13.67,
        "feels_like": 12.26,
        "temp_min": 13.67,
        "temp_max": 13.67,
        "pressure": 1015,
        "sea_level": 1015,
        "grnd_level": 826,
        "humidity": 58,
        "temp_kf": 0,
        "weather": [
          {
            "id": 803,
            "main": "Clouds",
            "description": "broken clouds",
            "icon": "04n"
          }
        ],
        "clouds": {
          "all": 77
        },
        "wind": {
          "speed": 1.15,
          "deg": 38
        },
        "sys": {
          "pod": "n"
        },
        "dt_txt": "2020-04-11 03:00:00"
      },
      "dt": "1586584800",
      "main": {
        "temp": 19.27,
        "feels_like": 17.87,
        "temp_min": 19.27,
        "temp_max": 19.27,
        "pressure": 1017,
        "sea_level": 1017,
        "grnd_level": 840,
        "humidity": 47,
        "temp_kf": 0.32,
        "weather": [
          {
            "id": 803,
            "main": "Clouds",
            "description": "broken clouds",
            "icon": "04d"
          }
        ],
        "clouds": {
          "all": 77
        },
        "wind": {
          "speed": 1.23,
          "deg": 86
        },
        "sys": {
          "pod": "d"
        },
        "dt_txt": "2020-04-11 06:00:00"
      },
      "dt": "1586595600",
      "main": {
        "temp": 24.45,
        "feels_like": 21.03,
        "temp_min": 24.45,
        "temp_max": 24.45,
        "pressure": 1015,
        "sea_level": 1015,
        "grnd_level": 841,
        "humidity": 30,
        "temp_kf": 0,
        "weather": [
          {
            "id": 802,
            "main": "Clouds",
            "description": "scattered clouds",
            "icon": "03d"
          }
        ],
        "clouds": {
          "all": 49
        },
        "wind": {
          "speed": 3.5,
          "deg": 118
        },
        "sys": {
          "pod": "d"
        },
        "dt_txt": "2020-04-11 09:00:00"
      },
      "dt": "1586606400",
      "main": {
        "temp": 25.4,
        "feels_like": 21.42,
        "temp_min": 25.4,
        "temp_max": 25.4,
        "pressure": 1011,
        "sea_level": 1011,
        "grnd_level": 839,
        "humidity": 29,
        "temp_kf": 0,
        "weather": [
          {
            "id": 802,
            "main": "Clouds",
            "description": "scattered clouds",
            "icon": "03d"
          }
        ],
        "clouds": {
          "all": 40
        },
        "wind": {
          "speed": 4.4,
          "deg": 116
        },
        "sys": {
          "pod": "d"
        },
        "dt_txt": "2020-04-11 12:00:00"
      }
    ]
  }
}
```

Forecast Result — By City Name

Requesting for the Forecast Weather Data

For this we'll need to import a python package — requests.

```
import requests
```

Here's how we get the result of the response into a variable, which we'll later break down so it's more readable.

```
forecast_json_data = requests.get(city_forecast).json()
```

The result of this will have the same structure as the link you went to,

```
In [25]: print(forecast_json_data)
```

```
{
  'cod': '200',
  'message': 0,
  'cnt': 40,
  'list': [
    {
      'dt': '1586466000',
      'main': {
        'temp': 17.07,
        'feels_like': 15.75,
        'temp_min': 15.81,
        'temp_max': 17.07,
        'pressure': 1016,
        'sea_level': 1016,
        'grnd_level': 838,
        'humidity': 57,
        'temp_kf': 1.26,
        'weather': [
          {
            'id': 500,
            'main': 'Rain',
            'description': 'light rain',
            'icon': '10n'
          }
        ],
        'clouds': {
          'all': 65
        },
        'wind': {
          'speed': 1.39,
          'deg': 65
        },
        'rain': {
          '3h': 0.16
        },
        'sys': {
          'pod': 'n'
        },
        'dt_txt': '2020-04-09 21:00:00'
      },
      'dt': '1586476800',
      'main': {
        'temp': 15.96,
        'feels_like': 14.62,
        'temp_min': 15.01,
        'temp_max': 15.96,
        'pressure': 1015,
        'sea_level': 1015,
        'grnd_level': 836,
        'humidity': 58,
        'temp_kf': 0.95,
        'weather': [
          {
            'id': 803,
            'main': 'Clouds',
            'description': 'broken clouds',
            'icon': '04n'
          }
        ],
        'clouds': {
          'all': 77
        },
        'wind': {
          'speed': 1.15,
          'deg': 38
        },
        'sys': {
          'pod': 'n'
        },
        'dt_txt': '2020-04-10 00:00:00'
      },
      'dt': '1586487600',
      'main': {
        'temp': 15.08,
        'feels_like': 13.71,
        'temp_min': 14.45,
        'temp_max': 15.08,
        'pressure': 1017,
        'sea_level': 1017,
        'grnd_level': 838,
        'humidity': 59,
        'temp_kf': 0.63,
        'weather': [
          {
            'id': 804,
            'main': 'Clouds',
            'description': 'overcast clouds',
            'icon': '04n'
          }
        ],
        'clouds': {
          'all': 85
        },
        'wind': {
          'speed': 1,
          'deg': 31
        },
        'sys': {
          'pod': 'n'
        },
        'dt_txt': '2020-04-10 03:00:00'
      },
      'dt': '1586498400',
      'main': {
        'temp': 19.27,
        'feels_like': 17.87,
        'temp_min': 18.95,
        'temp_max': 19.27,
        'pressure': 1017,
        'sea_level': 1017,
        'grnd_level': 840,
        'humidity': 47,
        'temp_kf': 0.32,
        'weather': [
          {
            'id': 803,
            'main': 'Clouds',
            'description': 'broken clouds',
            'icon': '04d'
          }
        ],
        'clouds': {
          'all': 77
        },
        'wind': {
          'speed': 1.23,
          'deg': 86,
          'sys': {
            'pod': 'd'
          },
          'dt_txt': '2020-04-10 06:00:00'
        },
        'dt': '1586509200',
        'main': {
          'temp': 24.45,
          'feels_like': 21.03,
          'temp_min': 24.45,
          'temp_max': 24.45,
          'pressure': 1015,
          'sea_level': 1015,
          'grnd_level': 841,
          'humidity': 30,
          'temp_kf': 0,
          'weather': [
            {
              'id': 802,
              'main': 'Clouds',
              'description': 'scattered clouds',
              'icon': '03d'
            }
          ],
          'clouds': {
            'all': 49
          },
          'wind': {
            'speed': 3.5,
            'deg': 118
          },
          'sys': {
            'pod': 'd'
          },
          'dt_txt': '2020-04-10 09:00:00'
        },
        'dt': '1586520000',
        'main': {
          'temp': 25.4,
          'feels_like': 21.42,
          'temp_min': 25.4,
          'temp_max': 25.4,
          'pressure': 1011,
          'sea_level': 1011,
          'grnd_level': 839,
          'humidity': 29,
          'temp_kf': 0,
          'weather': [
            {
              'id': 802,
              'main': 'Clouds',
              'description': 'scattered clouds',
              'icon': '03d'
            }
          ],
          'clouds': {
            'all': 40
          },
          'wind': {
            'speed': 4.4,
            'deg': 116,
            'sys': {
              'pod': 'd'
            },
            'dt_txt': '2020-04-11 12:00:00'
          }
        }
      ]
    }
  ]
}
```

Putting The Forecast Weather Data into a Pandas Data Frame

Create an empty dataframe

```
df_predictions = pd.DataFrame()
```

We first create empty lists that will later hold the json data.

```
# Creating empty lists
prediction_num = 0
list_prediction_num = []
date_time_prediction = []
owm_city_id = []
city_name = []
latitude = []
longitude = []
country_name = []
population = []
timezone = [] # Shift in seconds from UTC
sunrise = []
sunset = []
# Main
temp_prediction = []
temp_feels_like_prediction = []
temp_min_prediction = []
temp_max_prediction = []
pressure_prediction = []
sea_level_prediction = []
grnd_level_prediction = []
humidity_prediction = []
temp_kf_prediction = []
# Weather
main_weather_prediction = []
main_weather_description_prediction = []
# Clouds
clouds_prediction = []
# Wind
wind_speed_prediction = []
wind_degree_prediction = []
```

Loop Through the JSON and add (append) the data to the lists we have

created.

```
# Loop Through the JSON
for num_forecasts in forecast_json_data['list']:
    df_predictions['prediction_num'] = prediction_num
    list_prediction_num.append(prediction_num)
    date_time_prediction.append(forecast_json_data['list']
[prediction_num]['dt_txt'])

    owm_city_id.append(forecast_json_data['city']['id'])
    city_name.append(forecast_json_data['city']['name'])
    latitude.append(forecast_json_data['city']['coord']['lat'])
    longitude.append(forecast_json_data['city']['coord']['lon'])
    country_name.append(forecast_json_data['city']['country'])
    population.append(forecast_json_data['city']['population'])

    if forecast_json_data['city']['timezone'] >0 :
        timezone.append("+" + str((forecast_json_data['city']
['timezone'])/3600))
    else:
        timezone.append((forecast_json_data['city']
['timezone'])/3600)

    sunrise.append(forecast_json_data['city']['sunrise'])
    sunset.append(forecast_json_data['city']['sunset'])

# Main
temp_prediction.append(forecast_json_data['list'][prediction_num]
['main']['temp'])
temp_feels_like_prediction.append(forecast_json_data['list']
[prediction_num]['main']['feels_like'])
temp_min_prediction.append(forecast_json_data['list']
[prediction_num]['main']['temp_min'])
temp_max_prediction.append(forecast_json_data['list']
[prediction_num]['main']['temp_max'])
pressure_prediction.append(forecast_json_data['list']
[prediction_num]['main']['pressure'])
sea_level_prediction.append(forecast_json_data['list']
[prediction_num]['main']['sea_level'])
grnd_level_prediction.append(forecast_json_data['list']
[prediction_num]['main']['grnd_level'])
humidity_prediction.append(forecast_json_data['list']
[prediction_num]['main']['humidity'])
temp_kf_prediction.append(forecast_json_data['list']
[prediction_num]['main']['temp_kf'])
# Weather
main_weather_prediction.append(forecast_json_data['list']
[prediction_num]['weather'][0]['main'])

main_weather_description_prediction.append(forecast_json_data['list']
[prediction_num]['weather'][0]['description'])
# Clouds
clouds_prediction.append(forecast_json_data['list']
[prediction_num]['clouds']['all'])
# Wind
```

```

        wind_speed_prediction.append(forecast_json_data['list']
[prediction_num]['wind']['speed'])
        wind_degree_prediction.append(forecast_json_data['list']
[prediction_num]['wind']['deg'])

    prediction_num += 1

```

Put the data into a dataframe

```

# Put data into a dataframe
df_predictions['prediction_num'] = list_prediction_num
df_predictions['date_time_prediction'] = date_time_prediction
df_predictions['owm_city_id'] = owm_city_id
df_predictions['city_name'] = city_name
df_predictions['latitude'] = latitude
df_predictions['longitude'] = longitude
df_predictions['country_name'] = country_name
df_predictions['population'] = population
df_predictions['timezone'] = timezone
df_predictions['sunrise'] = sunrise
df_predictions['sunset'] = sunset

# Main
df_predictions['temp_prediction'] = temp_prediction
df_predictions['temp_feels_like_prediction'] =
temp_feels_like_prediction
df_predictions['temp_min_prediction'] = temp_min_prediction
df_predictions['temp_max_prediction'] = temp_max_prediction
df_predictions['pressure_prediction'] = pressure_prediction
df_predictions['sea_level_prediction'] = sea_level_prediction
df_predictions['grnd_level_prediction'] = grnd_level_prediction
df_predictions['humidity_prediction'] = humidity_prediction
df_predictions['temp_kf_prediction'] = temp_kf_prediction
    # Weather
df_predictions['main_weather_prediction'] = main_weather_prediction
df_predictions['main_weather_description_prediction'] =
main_weather_description_prediction
    # Clouds
df_predictions['clouds_prediction'] = clouds_prediction
    # Wind
df_predictions['wind_speed_prediction'] = wind_speed_prediction
df_predictions['wind_degree_prediction'] = wind_degree_prediction

```

Let's look at the result

Every Thursday, the Variable delivers the very best of Towards Data Science: from

In [34]:

df_predictions.head()

Out[34]:

	prediction_num	date_time_prediction	owm_city_id	city_name	latitude	longitude	country_name	population	timezone	sunrise	s
0	0	2020-04-09 21:00:00	184745	Nairobi	-1.2833	36.8167	KE	2750547	+3.0	1586403082	15864
1	1	2020-04-10 00:00:00	184745	Nairobi	-1.2833	36.8167	KE	2750547	+3.0	1586403082	15864
2	2	2020-04-10 03:00:00	184745	Nairobi	-1.2833	36.8167	KE	2750547	+3.0	1586403082	15864
3	3	2020-04-10 06:00:00	184745	Nairobi	-1.2833	36.8167	KE	2750547	+3.0	1586403082	15864
4	4	2020-04-10 09:00:00	184745	Nairobi	-1.2833	36.8167	KE	2750547	+3.0	1586403082	15864

Voila!

You can find the [Github repo](#) here.