

DE Take-home exercise

The goal is to retrieve data from an api, and format it as per the spec.

Detailed steps:

-Your code should be written in Python. If you would like to work in Go that is fine as well, but using pandas will make a lot of the work simpler.

- Fetch the weather data from this API endpoint. You will need to sign up on <https://openweathermap.org/api> to get an API key. The endpoint below uses this API: [One Call API: 5 Day / 3 Hour Forecast](#)

```
https://api.openweathermap.org/data/2.5/forecast?id={city ID}
&units=metric&appid={API key}
```

Example data returned from openWeather API

```
{
  "cod": "200",
  "message": 0,
  "cnt": 40,
  "list": [
    {
      "dt": 1596564000,
      "main": {
        "temp": 5.3,
        "feels_like": 5.3,
        "temp_min": 5.3,
        "temp_max": 5.3,
        "pressure": 1013,
        "sea_level": 1013,
        "grnd_level": 976,
        "humidity": 84,
        "temp_kf": -0.5
      },
      "weather": [
        {
          "id": 500,
          "main": "Rain",
          "description": "light rain",
          "icon": "10d"
        }
      ],
      "clouds": {
        "all": 38
      },
      "wind": {
        "speed": 4.35,
        "deg": 309,
        "gust": 7.87
      },
      "visibility": 10000,
```

```

        "pop": 0.49,
        "rain": {
            "3h": 0.53
        },
        "sys": {
            "pod": "d"
        },
        "dt_txt": "2020-08-04 18:00:00"
    },
    ...

    "city": {
        "id": 2643743,
        "name": "London",
        "coord": {
            "lat": 51.5073,
            "lon": -0.1277
        },
        "country": "GB",
        "timezone": 0,
        "sunrise": 1578384285,
        "sunset": 1578413272
    }
}

```

1. After retrieving the data, write the raw data returned from the api into a json file that can be referenced later.
2. Format the retrieved data into a pandas dataframe that contains the dt, temp, temp_min, temp_max, and wind speed columns per returned dt.
3. Identify if there is any missing data that you would have expected to be present and print a warning. There may not be anything missing which is fine, but we would like to see how you would design this validation.
4. Format the dt column such that it takes the format "YYYY-MM-DD HH:MM:SS". Assume UTC timezone.
5. Add a new column to the dataframe. The value of this column needs to consider the current date in each row, and subtract 1 hour from the time. Use the same format as in the previous step
EX: 2020-03-31 05:00:00 will be used to create a new column with value 2020-03-31 04:00:00
6. The temp column is in Celsius (assuming you use the `units=metric` in the api call). Without just changing the api call, convert these values from Celsius to Fahrenheit. No need to do it for the other temperature columns.
7. The program should save the resulting dataset in a csv.
8. Write at least 1 unit test for testing a piece of your code.
9. Your app should be dockerized so it can run off any machine. In your readme for the code, specify the commands we need to run to build the docker container. The container needs to be long lived so that once execution of the code finishes, we can get into the container to view the output files.
10. Specify any other instructions we need to run the code.