

Openweather et API

Robert CLERC

Lien vers le github :

<https://github.com/robertclerc/openweather>

Description des fichiers essentiels :

Dockerfile : Fichier permettant de créer l'image docker

City.list.json : Villes de France

requirements.txt : Librairies python installées au moment de la création de l'image Docker.

weatherOpenMap 2021 12 17 21 47 14.csv : Températures des villes de France

weatherOpenMap mean city 2021 12 17 21 47 14.csv : Températures moyennes des villes de France

weather_api_formation.py : Script Python permettant de faire les requêtes API

test.R : Script R pour afficher la carte des résultats

Résumé :

Les commandes pour faire le git et le docker (image, container et volume) étaient simples à réaliser. Pour contourner la fréquence du nombre de requêtes API limité à 60 par minutes, j'ai ajouté la commande `time.sleep(3)` dans le script python pour pouvoir faire les requêtes pour toutes les villes.

Je suis allé plus loin dans le projet en réalisant une carte statique des températures. A noté qu'il est envisageable de faire une carte dynamique de la france avec les températures par ville et/ou départements : <https://thinkr.fr/cartographie-interactive-avec-r-la-suite/>

Liste des modifications apportées au script python

API :

1 - Modification de la fonction data_organizer pour ne garder que les températures sans les autres paramètres (pression, humidité, etc).

```
def data_organizer(raw_api_dict):
    data = dict(
        city=raw_api_dict.get('name'),
        #country=raw_api_dict.get('sys').get('country'),
        temp=raw_api_dict.get('main').get('temp'),
        temp_max=raw_api_dict.get('main').get('temp_max'),
        temp_min=raw_api_dict.get('main').get('temp_min')#,
        # humidity=raw_api_dict.get('main').get('humidity'),
        # pressure=raw_api_dict.get('main').get('pressure'),
        # sky=raw_api_dict['weather'][0]['main'],
        # sunrise=time_converter(raw_api_dict.get('sys').get('sunrise')),
        # sunset=time_converter(raw_api_dict.get('sys').get('sunset')),
        # wind=raw_api_dict.get('wind').get('speed'),
        # wind_deg=raw_api_dict.get('deg'),
        # dt=time_converter(raw_api_dict.get('dt'))#,
        # cloudiness=raw_api_dict.get('clouds').get('all')
    )
    # print (data)
    return data
```

2 - Modification du nom du fichier pour ajouter la date et l'heure de l'exécution du script dans le nom du fichier.

```
date_for_path = "_" + str(datetime.datetime.now()).replace("-", "_").replace(":", "_").replace(" ", "_").replace(".", "_")[:-7]

path_1 = "weatherOpenMap" + date_for_path + ".csv"
path_2 = "weatherOpenMap_mean_city" + date_for_path + ".csv"
```

3 - Modification de la fonction WriteCsv pour mettre un mode append si le fichier existe déjà. Cela permet de voir le fichier se remplir en temps réel et éviter d'attendre la fin de l'exécution pour se rendre compte que l'exécution à échouée.

```
def WriteCSV(data,path_file):  
    file_exists = os.path.isfile(path_file)  
  
    if not file_exists:  
  
        with open(path_file, 'w') as f: # Just use 'w' mode in 3.x  
            w = csv.DictWriter(f, data.keys())  
  
            w.writeheader()  
            w.writerow(data)  
  
    else :  
  
        with open(path_file, 'a') as f: # Just use 'w' mode in 3.x  
            w = csv.DictWriter(f, data.keys())  
  
            w.writerow(data)
```

4 - Selection des villes de France dans le dictionnaire de toutes les villes

```
villes_france = villes[villes['country'] == "FR"]  
  
liste_ville = sorted(list(set(villes_france['name'])))
```

5 - Modification du main principal pour boucler sur les villes et sur les identifiants des villes (donc double boucle for) pour récupérer les températures.

```
for ville_to_get in liste_ville :  
    city_name = ""  
  
    liste_id_city = list(villes_france.id[villes_france["name"]==ville_to_get])  
    dic_ville_global = {}  
    compteur_ville = 0  
  
    for id_to_get in liste_id_city :  
  
        compteur_ville +=1  
  
        city_id = id_to_get  
  
        url=url_builder(city_id,city_name,country)
```

Ajout :

Après avoir obtenu le fichier csv avec toutes les températures de France, j'ai utilisé le package ggmap (pour le langage R) pour afficher une carte de france avec toutes les températures obtenues. Chaque point représente une ville affichée selon sa longitude et sa latitude et la couleur représente la température relevé sur openweather. Le code est disponible sur le script test.R

