

Projet Week-end

Xavier André & Romain Tavenard

1 Préambule

Pour ce projet, vous devrez travailler par groupes de 2 ou 3 et votre rendu se fera sous la forme d'un fichier Python. Ce fichier devra être nommé `P02.py` et contenir, sous la forme de commentaires dans l'en-tête du fichier les noms et numéros étudiant de tous les membres du groupe comme dans l'exemple suivant :

```
# 22000002 Paul Machin  
# 22000227 Yolène Truc  
  
# Section 1 : les imports  
# [...]
```

La date limite de rendu est indiquée sur CURSUS dans l'espace de dépôt.

2 Le problème

Des amis souhaitent se retrouver pour passer un week-end ensemble. Ils comptent se retrouver dans la ville de l'un d'entre eux. Chacun disposant de logement trop petit, les autres amis devront louer des chambres d'hôtel dans cette ville. Vous devez écrire un programme Python qui leur permette de lister les hôtels existants, en utilisant les ressources suivantes :

- les adresses des amis peuvent être obtenues via une API dont l'URL est la suivante : https://my-json-server.typicode.com/rtavenar/fake_api/adresses_amis;
- les hôtels peuvent être obtenus via l'API `data.gouv.fr` via le lien <https://tabular-api.data.gouv.fr/api/resources/3ce290bf-07ec-4d63-b12b-d0496193a535/data/>. Le programme devra afficher la liste des hôtels disponibles dans la ville de chacun des amis, en indiquant pour chaque hôtel son nom, son adresse et son nombre d'étoiles.

3 Accès aux données

Pour accéder aux données de data.gouv.fr, vous devrez les télécharger “page par page”, car l’API ne permet pas de récupérer plus de 200 entrées par requête, alors que la taille de la collection de données qui nous intéresse dépasse ce nombre.

Nous vous proposons ci-dessous un squelette de code qui permette d’effectuer cette pagination, vous êtes libre de l’utiliser ou non dans votre solution.

```
def fetch_all_data(base_url):  
    all_rows = []  
    page = 1  
  
    while True:  
        url = f"{base_url}?page_size=200&page={page}"  
  
        response = requests.get(url)  
        response.raise_for_status()  
        data = response.json()  
  
        rows = data.get("data", [])  
        if not rows:  
            break  
  
        all_rows.extend(rows)  
  
        # Si moins de 200 lignes : dernière page  
        if len(rows) < 200:  
            break  
  
        page += 1  
  
    return all_rows
```