

## TD9 : API Twitter

Dans ce TD, vous utiliserez le module ‘tweepy’ pour manipuler des données issues de l’API Twitter.

Il est grandement recommandé de consulter l’aide mémoire disponible à cette adresse : [https://raw.githubusercontent.com/alemaitr/python\\_opendata\\_l2/master/DocTweepy/doc\\_tweepy.pdf](https://raw.githubusercontent.com/alemaitr/python_opendata_l2/master/DocTweepy/doc_tweepy.pdf)

### Travail à préparer chez vous avant la séance

Suivez les indications fournies sur le document [https://raw.githubusercontent.com/alemaitr/python\\_opendata\\_l2/master/DocTweepy/compteDeveloppeur.pdf](https://raw.githubusercontent.com/alemaitr/python_opendata_l2/master/DocTweepy/compteDeveloppeur.pdf), pour créer votre compte Twitter et obtenir les clé API.

### Préambule

- Créer sur votre disque un dossier TD9 dans PythonOpenData.
- Copier le fichier `credentials.json` du TD1 dans TD9.
- Lancer l’éditeur Visual Studio Code.
- Dans Visual Studio Code, ouvrir le dossier PythonOpenData/TD9.

### Installation du module tweepy

Sur les ordinateurs de l’Université de Rennes 2, le module ‘tweepy’ n’est pas installé par défaut, vous devez donc l’installer. Pour cela, vous ouvrir le terminal dans Visual Studio Code, puis entrer la ligne :

```
pip install --user tweepy.
```

### Exercice 1 : Identification

Vous allez devoir vous authentifier sur l’API Twitter. Vous ne devrez jamais laisser apparaître vos identifiants dans votre code Python, mais les stocker dans un fichier externe.

Q.1.1. Compléter le fichier `credentials.json` avec vos identifiants de twitter, pour qu’il ait le format suivant :

```
{
  "twitter": {
    "CONSUMER_KEY": "...",
    "CONSUMER_SECRET": "...",
    "ACCESS_TOKEN": "...",
    "ACCESS_TOKEN_SECRET": "..."
  },
  "GraphHopper": {}
}
```

où les “...” seront remplacés par vos identifiants fournis par l’interface Twitter.

Q.1.2. Écrire une fonction `client_twitter` qui prend en entrée le nom d’un fichier json contenant les clés, qui lit les identifiants dans ce fichier et qui retourne un client d’accès à l’API Twitter pour ces identifiants.

## Exercice 2 : Écriture de tweets

- Q.2.1. Écrire une fonction `tweeter` qui prend en entrée un client Twitter, un message, et qui poste un tweet avec ce message. Cette fonction doit renvoyer l'identifiant du tweet ainsi créé.
- Q.2.2. Tester cette fonction pour poster un message de votre choix (ex : "Je suis en TP Python").
- Q.2.3. Écrire une fonction `efface_tweet` qui prend en entrée un client Twitter et un identifiant de tweet, et supprime le tweet correspondant.
- Q.2.4. Tester cette fonction pour supprimer un des tweets précédemment posté.

## Exercice 3 : Identifiants utilisateurs

- Q.3.1. Créer une fonction `details_utilisateur` qui prend en entrée un client et un nom d'utilisateur Twitter, et qui renvoie un dictionnaire contenant le détail des informations de cet utilisateur.  
Par exemple, pour `UnivRennes_2`, on renverra le dictionnaire suivant :

```
{ 'Utilisateur': 'UnivRennes_2',  
  'User_id': 1354601634,  
  'Nom': 'Université Rennes 2',  
  'Lieu': 'Rennes, France',  
  'Description': "Toute l'actualité des #Arts, #Lettres, #Langues,  
                  #SHS et #STAPS à l'Université #Rennes2",  
  'Date de création': '04/2013' }
```

Notons que la date de création est convertie sous forme d'une chaîne de caractère ne contenant que le mois et l'année.

- Q.3.2. Utiliser cette fonction pour obtenir votre identifiant utilisateur.

## Exercice 4 : Consultation de tweets

- Q.4.1. Écrire une fonction `derniers_tweets` qui prend en entrée un client, un identifiant d'utilisateur, un entier `n` et qui renvoie les `n` derniers tweets de l'utilisateur. Les tweets seront renvoyés sous la forme d'une liste de dictionnaires, chaque dictionnaire sera de la forme suivante :

```
{ 'date': datetime.datetime(2022, 11, 24, 11, 8, 24, tzinfo=datetime.timezone.utc),  
  'texte': 'La journée pédagogique NCU Coursus IDE@L sur le thème de l'hybridation est lancée !',  
  'tweet_id': 1595736085518245888 }
```

- Q.4.2. Utiliser cette fonction pour afficher vos 5 derniers tweets.
- Q.4.3. Compléter cette fonction pour enrichir les dictionnaires retournés avec la géo-localisation du tweet : lorsque les informations de coordonnées GPS sont fournies, ajouter deux clés "lat" et "long" contenant la latitude et la longitude du tweet.
- Q.4.4. Tester en affichant les derniers tweets de l'utilisateur `chris_suspecte`  
Voici un exemple de dictionnaire fourni :

```
{ 'date': datetime.datetime(2022, 11, 24, 6, 45, 15, tzinfo=datetime.timezone.utc),  
  'lat': 37.787994,  
  'long': -122.407437,  
  'texte': 'Je suis à San Francisco [Ceci est un faux tweet avec localisation]',  
  'tweet_id': 1595669860960419846 },
```

## Exercice 5 : Questions non guidées

Les questions ci-dessous sont plus ouvertes. Écrire le code nécessaire pour répondre à chacune des questions, en utilisant des fonctions lorsque c'est pertinent.

### Question 5.1 : Tweet favori

Problème : Parmi les 50 derniers tweet de l'Université Rennes 2 (*UnivRennes\_2*), lequel a récolté le plus de like (points favoris) ?

Exemple de résultat :

Le tweet de Université Rennes 2 ayant récolté le plus de like date du 08/11/22 à 10:04 :

```
-----
Ma thèse en 180 secondes : relevez le défi de présenter votre #thèse en 3 minutes !
Les inscriptions à la formation #MT180 sont ouvertes jusqu'au 16 décembre.
Au programme, 2 jours pour se former à la médiation #scientifique !
Dates et programmes : https://t.co/G00GvEmJNT https://t.co/Wah5HiZxlo
-----
```

### Question 5.2 : Régularité

Problème : Sur chacun des jours de septembre 2022, combien l'Université Rennes 2 (*UnivRennes\_2*) a-t-elle posté de tweets ?

Exemple de résultat :

Voici la répartition des tweets de Rennes 2 sur le mois de septembre 2022

```
{1: 8, 2: 2, 3: 0, 4: 0, 5: 2, 6: 5, 7: 2, 8: 2, 9: 2, 10: 0, 11: 0, 12: 4,
13: 6, 14: 4, 15: 3, 16: 5, 17: 0, 18: 0, 19: 4, 20: 3, 21: 2, 22: 3, 23: 6,
24: 0, 25: 0, 26: 2, 27: 1, 28: 8, 29: 4, 30: 1}
```

### Question 5.3 : Followers

Problème : Citer des utilisateurs qui suivent à la fois le compte de l'Université Rennes 2 (*UnivRennes\_2*) et celui de la ville de Rennes (*metropolerennes*) ?

Exemple de résultat :

Voici le nom de 46 followers communs à Université Rennes 2 et Rennes métropole  
Il s'agit de :

```
['Eli Ackerman', 'lumiere espoir', 'Sidahmed Ali', 'Bienfait Hugo',
 'Association Météo Bretagne', 'Léa', 'Educ_info', 'Gaïd Le Maner-Idrissi',
 'david delapousse', 'ED PROMOTION', 'Avenarius', ...]
```

Note : Il est difficile d'extraire ici tous les followers communs, car la récupération simple des followers est limitée à 1000.