

API web d'accès aux données

Aurélie Lemaitre, Romain Tavenard

1 Travail à préparer chez vous avant la séance

1. Préparez un compte Twitter à utiliser pendant la séance. Pour cela, vous pourrez soit utiliser un compte Twitter existant, soit en créer un pour l'occasion. Ensuite, vous devrez suivre les indications fournies [ici](#) pour créer une “Application”, c'est-à-dire un cadre dans lequel vous aurez le droit de faire des requêtes à l'API Twitter depuis votre code Python.

2 Cas des ordinateurs de l'Université de Rennes 2

Sur les ordinateurs de l'Université de Rennes 2, le module `tweepy` n'est pas installé par défaut. Pour pouvoir l'utiliser, il va donc falloir commencer par l'installer. Pour cela, vous devrez :

- a. Ouvrir le logiciel **Anaconda Prompt** (bienvenue dans le monde merveilleux des lignes de commande :)
- b. Entrer la ligne suivante :

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

Une fois cela fait, vous pouvez fermer la fenêtre **Anaconda Prompt** et vous devriez pouvoir utiliser ces modules sans soucis (au moins pour la durée de votre session, potentiellement un peu plus que cela si vous restez sur la même machine) dans PyCharm.

3 Énoncé

Dans ce TD, vous utiliserez le module `tweepy` pour manipuler des données issues de l'API Twitter. Il est fortement conseillé de vous aider des deux documents suivants pour ce TD :

- [l'aide en ligne de tweepy](#)
- [le document d'explication fourni pour ce cours](#)

3.1 Mise en pratique

2. Pour commencer, vous allez devoir vous authentifier sur l'API Twitter. Comme indiqué en cours, vous ne devrez jamais laisser apparaître vos identifiants dans votre code Python. Créez donc, dans le répertoire `"data"`, un fichier `credentials.json` qui ait le format suivant :

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

où les `"..."` seront remplacés par vos identifiants fournis par l'interface Twitter.

3. Écrivez une fonction qui prend en entrée le nom du fichier à lire et retourne une variable d'accès à l'API Twitter correspondant aux identifiants lus dans le fichier en question.
4. Écrivez une fonction qui prend en entrée la variable d'accès à l'API et retourne la liste des 2 derniers tweets de l'utilisateur identifié.
5. Écrivez une fonction qui prend un tweet en entrée (de type `Status`) et retourne le texte du tweet en question.
6. En utilisant la fonction de la question précédente, écrivez une fonction qui prend une liste de tweets en entrée et retourne la liste des textes des tweets en question.
7. Écrivez une fonction qui prend en entrée un tweet (de type `Status`) et retourne l'identifiant de son auteur.
8. Le fichier `tweets.csv` disponible sur CURSUS contient une liste de Tweets que vous allez maintenant poster. Pour chaque tweet, on fournit son texte ainsi que sa position GPS. Copiez ce fichier dans votre répertoire `"data"` et écrivez une fonction qui prend en entrée la variable d'accès à l'API et le chemin vers le fichier contenant les tweets et poste l'ensemble des Tweets contenus dans ce fichier. Au texte contenu dans le fichier, vous ajouterez la mention `"Ceci est un faux tweet posté depuis mon TD de Python"`. Vérifier que les Tweets apparaissent bien sur votre compte Twitter, avec la localisation voulue.
9. Améliorez la fonction de la question précédente pour qu'elle retourne la liste des identifiants des tweets créés.

10. Écrivez une fonction qui prend en entrée la variable d'accès à l'API et une liste d'identifiants de tweets et supprime tous les tweets correspondants.