

1. Objectif

Vous familiariser avec l'utilisation de bibliothèques de tests et la préparation d'une stratégie de tests automatisés. La bibliothèque *unittest* incluse avec la distribution standard de Python sera utilisée.

2. Installation de l'environnement de travail

Ce laboratoire utilise le langage Python. Vous devrez également installer la librairie *requests*, disponible sur PIP.

Ce laboratoire nécessite l'utilisation d'une clé d'API pour Twitter. Il est nécessaire de posséder un compte Twitter pour cela.

1. Rendez-vous sur <https://apps.twitter.com/>
2. Demandez de créer un compte développeur
3. Sélectionnez que vous souhaitez appliquer en tant qu'étudiant

Which best describes you?

This is how you intend to use the Twitter developer platform

Professional

Hobbyist

Academic

Academic researcher

Teacher

Student

Something else (But related to academics)

Standard application

Submit a standard application for access to the Twitter developer platform.

Get started

4. Entrez les informations nécessaires. Pour ce qui est de la description d'Intended Use, je vous propose une formule suivante : I will use the Twitter API to learn about how to query an API in Python using an authentication system. In the assignment, we will only be using the standard search API call. We will not save the data received. Choisissez un compte individuel. Vous pouvez également sélectionner que vous n'utiliserez pas les autres fonctionnalités de l'API

5. Vous devriez recevoir une confirmation relativement rapidement de Twitter. Vous pouvez ensuite créer une application pour recevoir la clé d'authentification.
6. Choisissez de créer un nouveau projet et nommez-le

Name Your Project

1 Project name

2 Use case

3 Project description

4 App set up

Your Project helps you organize your work and monitor your usage with the Twitter API.

Laboratoire4

20

7. Entrez encore une fois les informations nécessaires. Vous pouvez utiliser une formule semblable pour le *use case*.
8. Vous arriverez enfin à 3 clés; celle qui nous intéresse est le *Bearer Token*. Sauvegardez-le dans le fichier *Server.py* à la ligne 3.
9. Vous devriez être maintenant capable d'utiliser les services de Twitter par l'API.

3. Réalisation

En premier lieu, familiarisez-vous avec le code. Il s'agit d'un microserveur Python. Celui-ci est disponible à l'adresse <http://localhost:8080>. Il s'agit simplement d'un champ de recherche qui envoie à l'API de Twitter ce que vous entrez et vous retourne les 10 premiers résultats des derniers jours. On « enregistre » ensuite ces données et on les affiche à l'écran.

Il y a plusieurs composantes à l'œuvre ici. Le but est de partir de ces bouts de code fonctionnels et d'écrire une suite *LA PLUS COMPLÈTE POSSIBLE* de tests unitaires. Vous devez procéder avec la méthode TDD, tel que vu en classe, c'est-à-dire en écrivant un test qui « fail » pour ensuite renforcer la « spec » (le code déjà fourni). Par exemple, que se passe-t-il si le programme ne peut pas faire afficher les Tweets en « base de données », car le texte est *null* (*None* en Python)?

Vous pourriez dans ce cas ne rien afficher, lever une exception, etc. C'est à vous de sélectionner le comportement voulu, mais sachez que vous devez documenter ce choix. Pour les besoins du laboratoire, cela sera fait de deux manières :

1. Avec les tests unitaires
2. Avec un rapport (format Word, PDF, Markdown, Latex, etc.) qui indique les exigences que remplit maintenant le programme. Pour cela, vous pouvez les écrire comme indiqué dans le document de Spécification des exigences de la norme ISO 29110. (nul besoin du rapport complet, simplement la section nécessaire)

4. Rapport

Vous serez évalué sur la « robustesse » de votre suite de tests, sur votre rapport ainsi que le respect du TDD (un *commit* pour l'écriture du test, un *commit* pour renforcer la « spec » du programme au besoin). Il peut y avoir facilement une vingtaine de tests possible. N'hésitez pas à échanger avec moi ou Jean-Michel si vous sentez que vous n'avez pas d'idée.

Placer le code SOURCE dans un répertoire Git sur Github et m'envoyer par courriel le lien vers votre répertoire. N'oubliez pas de mettre votre répertoire en mode public. Vous pouvez également mettre directement dans votre répertoire votre rapport.

Le projet doit fonctionner sans faute simplement en compilant, autrement le projet n'est pas corrigé. LA suite de test doit pouvoir également être exécuté sans faute. La qualité du code et de la documentation, ainsi que le respect des normes seront ensuite évaluées.

Bon travail !!!