

Cloud computing

Projet

Faire place nette

Utiliser le dépôt de TD. (vous pouvez le renommer si besoin)

Déplacez votre travail de TD (fichiers, dossiers et README.md) dans un dossier du même nom à la racine du dépôt.

Créez un nouveau README.md à la racine du projet qui servira de rapport pour le projet avec pour l'instant **membres du groupe**.

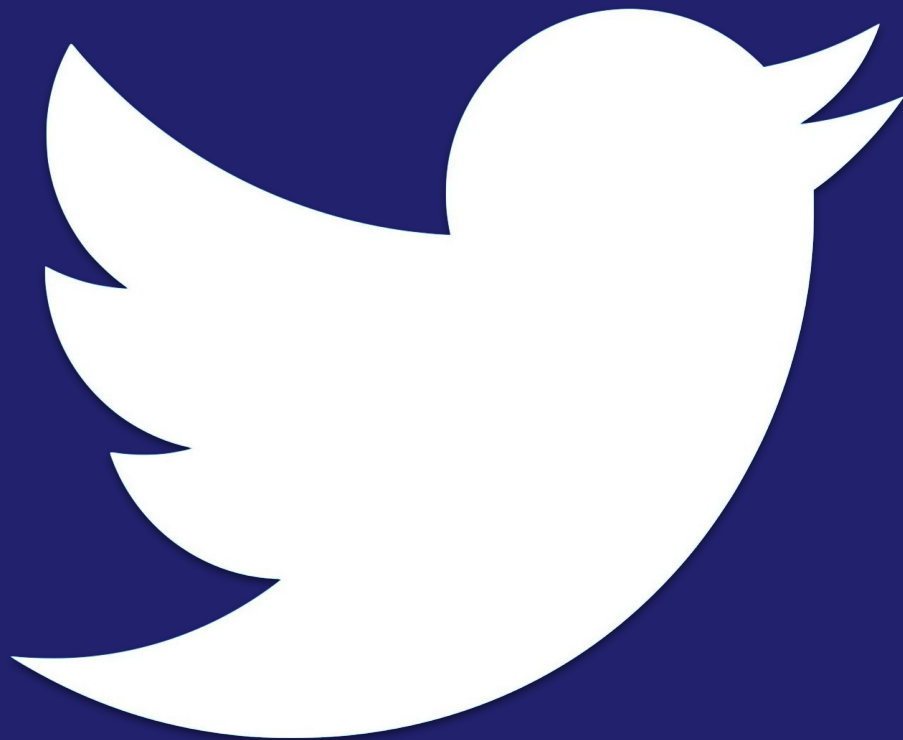
Vous viendrez l'agrémenter avec le **déroulé du projet**, les **technologies utilisées**, des **badges**, des **badges de résultats de l'exécution des CIs** et la **procédure** pour exécuter les composantes du projet.

On va refaire twitter, en mieux

21 mars 2006 (bientôt 17 ans)

5,08 milliards de dollars en 2021

Rachat par Elon Musk



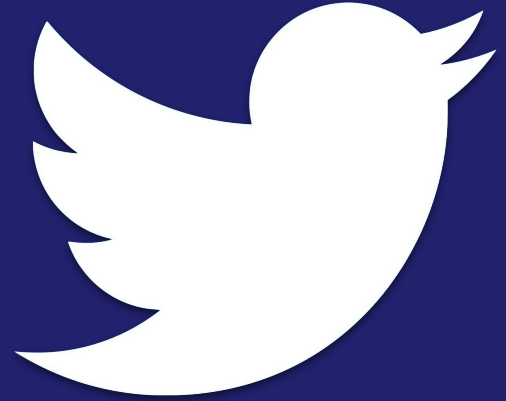
Classement des dix comptes les plus suivis
(au 5 novembre 2022) :

Rang	Comptes	Abonnés
1	@BarackObama	133 244 623
2	@elonmusk	119 897 462
3	@justinbieber	113 666 216

On va refaire twitter, en mieux

Des utilisateurs au système, twitter c'est

- Un Frontend
- Une API pour gérer les tweets
- Une DB pour stocker les tweets



Fonctionnalité de base : envoyer un tweet, afficher les tweets, afficher les tweets d'un sujet ou d'une personne en particulier, retweeter.

Portabilité

Vous organiserez votre dépôt GitHub avec deux dossiers à la racine du projet à côté du README principal: un pour le **frontend**, un pour le **backend**.

Chacun d'eux contenant un **README** spécifique et un **Dockerfile** permettant de lancer les programmes dans des conteneurs.

Encore du Flask ?



Concevoir une API simple (Python/Flask, Node ou Rust).

- Afficher tous les tweets.
- Enregistrer un tweet dans Redis.
- Attribuer un tweet à une personne.
- Retweeter.
- Afficher les sujets.
- Afficher les tweets liés à un sujet.

Tips : Avant de mettre en place les bases Redis, vous pouvez utiliser des dictionnaires pour tester vos routes et vos fonctionnalités.

Modèle de donnée

Stocker les tweets dans deux bases Redis.

- ❑ Une base contenant les tweets.

ex: key=timestamp, value='{“author”: “username”, “tweet”: “message”}'

- ❑ Une base contenant les utilisateurs dans laquelle la clé sera le nom d'utilisateur et en valeur la liste des clés de ses tweets.

ex: key=username, value=[timestamp_1, timestamp_2, timestamp_3]

Vous êtes libre de choisir une autre architecture, mais ça doit fonctionner 😊

Stocker les tweets via Redis

Utiliser [Redis](#) comme serveur de données, c'est un système de stockage clé/valeur. Il utilise le port **6379** pour communiquer.

Rappel : utilisez [redis-cli](#) pour accéder à redis et testez les commandes set, get, ...

```
docker run --name myredis -p 6379:6379 redis
```

(utilisez un autre terminal)

Remplacer le stockage précédemment
en variable par un stockage via Redis

```
import redis
r = redis.Redis(host='localhost', port=6379, db=0)

# Ajout de variable
r.set('foo', 'bar') # Retourne True si réussite

# Lecture de
r.get('foo')        # Retourne la value de foo
```


Comment gérer les sujets ?

Tout simplement en créant une clé dédiée au sujet dans la table des utilisateurs.

⇒ Gérez les sujets comme des utilisateurs !

Vous pouvez utiliser des un préfix pour éviter les conflits de clé. Par exemple, les utilisateurs auront une clé au format **u-username** et les sujets **h-hashtag**.

 *vous pouvez utiliser un autre format, précisez votre choix dans le readme*

Tips : Pour trier les liens entre sujets, utilisateurs et tweets tout se fera au traitement de la requête.

Tout est dans le détail

Il ne manque plus que l'interface utilisateur.

Avec la technologie de votre choix (HTML/CSS/JS, VueJS, React, ...) réalisé un frontend pour communiquer avec votre API. Via boutons et formulaires, il permettra d'appeler les différents endpoints de votre API et de mettre en forme les retours de l'API.

Laissez libre cours à vos envies et votre imagination pour designer votre Twitter, la forme importe peu mais elle devrait couvrir toutes fonctionnalités de l'API décrite dans la section "Encore du Flask ?" 🚀

i N'oubliez pas le Dockerfile pour permettre le lancement du frontend dans un conteneur.

i Vous décrivez les fonctionnalités du front dans votre README.md

Bonne chance 🚀

Ce projet à rendre au plus tard le 7 avril 2023 à 23h59

[sujet en version GitHub](#)