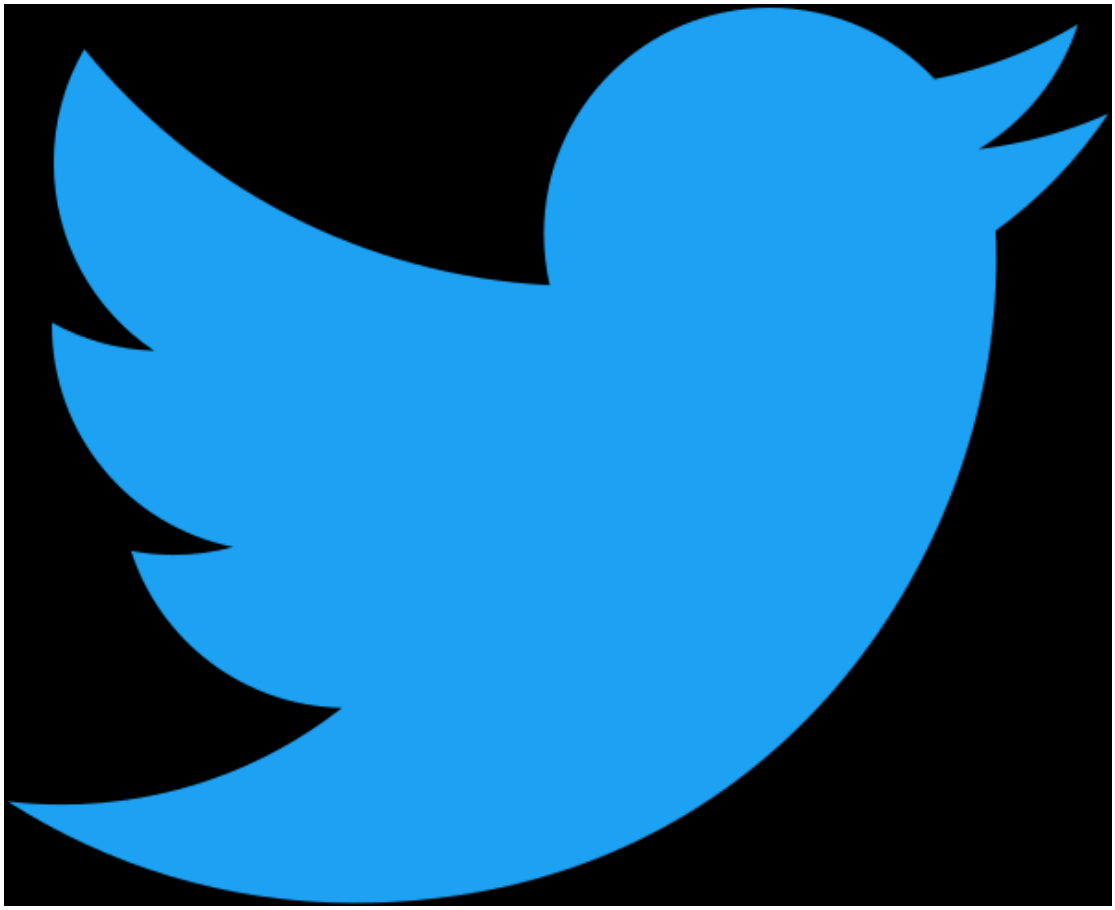


TWITTER SCRAPER

DOCUMENTAZIONE

Francesco Martinelli mat.744041
Umberto Nocerino mat.747751
Magistrale di Sicurezza Informatica

Sommario

INTRODUZIONE

WEB SCRAPING

Prima di soffermarci nelle specifiche di progetto è doveroso spiegare innanzitutto l'ambito in cui applica lo scraping e capire lo scopo di questo strumento di analisi.

I social media crescono rapidamente e sapervisi muovere tramite corretto utilizzo diviene sempre più parte cruciale della vita quotidiana, sia essa sociale o professionale. L'estrazione dei dati da piattaforme come Facebook, Twitter e Instagram utilizzando tecniche di scraping, ha reso il mining dei dati semplice ed efficace per il business. Il web scraping è una tecnica per automatizzare il processo di estrazione dei dati. Questi strumenti rendono questi dati disponibili per il marketing digitale. Quindi ci troviamo nel campo della Social Media Analytics che è un insieme di tecniche con le quali si raccolgono dati relativi alle interazioni nei Social: questi dati sono successivamente elaborati grazie ad avanzati algoritmi di Sentiment Analysis, e poi studiati, ecc., per completare il processo di analisi.

TWITTER

Twitter è un social network molto popolare che permette di comunicare attraverso messaggi brevi, foto e video da pubblicare da computer, smartphone e tablet. È gratuito e abbastanza rispettoso della privacy degli utenti. Per capire bene come funziona Twitter bisogna partire proprio dal presupposto che si parla di un mondo completamente diverso – sia nel funzionamento che nei contenuti – rispetto a quello di Facebook ad esempio. Non ci sono amicizie da accettare o contraccambiare, tutti i post sono liberamente leggibili da chiunque (eccetto i messaggi diretti) e non ci sono eventi o giochi a cui partecipare.

PYTHON WEB SCRAPING

Per quanto riguarda la fase di progettazione abbiamo innanzitutto fatto richiesta a Twitter delle API (Application Programming Interface) che come sappiamo sono un insieme di procedure atte all'espletamento di un compito specifico: nel nostro caso era quello di poter effettuare l'autenticazione e accedere ai dati dell'utente specifico. Per la richiesta abbiamo inviato una e-mail a Twitter specificando che l'utilizzo era a scopo accademico presso l'Università di Bari.

Dopodiché abbiamo analizzato i vari tool presenti in Phyton (abbiamo ritenuto che fosse il linguaggio più idoneo per portare a termine il progetto) e ci siamo soffermati andando a studiare Tweepy: che è una libreria utile per consentire una comunicazione rapida ed efficace con questo social.

Successivamente abbiamo raccolto informazioni e definito i requisiti di progetto, abbiamo progettato degli algoritmi (ad esempio quello per le interazioni) e successivamente siamo passati alla realizzazione vera e propria.

TWITTER SCRAPER

Il nostro lavoro ha previsto la creazione di un software che contiene al suo interno delle funzioni utili al raggiungimento degli obiettivi del progetto, comprese funzionalità aggiuntive, ovvero:

- Creazione file csv con interazioni dell'account (tweet, retweet, commenti, etc.)
- Creazione file csv con le sole interazioni di uno o più contatti
- Creazione file csv con i messaggi privati dell'account
- Creazione file csv con i messaggi privati con uno o più contatti
- Autenticazione account temporaneo
- Creazione file log
- File hashing con algoritmo MD5 e SHA1.
- Creazione file csv con relationship con uno o più account.

La funzione "relationship" individua la relazione che possiede l'account con un determinato utente (se si seguono a vicenda, se è bloccato, se è mutato, etc.)

Il programma contiene una interfaccia utente, o nello specifico una Web App creata tramite il framework Flask, il quale permette di eseguire le funzionalità descritte sopra tramite gli appositi bottoni. La procedura dei singoli bottoni è descritta nel manuale di Twitter Scraper.

ARCHITETTURA

- PREREQUISITI:
  Registrazione dell'account su Twitter Developer

L'architettura è composta da 3 componenti:

- Il backend gestito da codice Python.
- Web App realizzata con Flask
- Frontend realizzato in html, css e javascript.

BACKEND SCRIPT

L'interazione tra il codice e l'interfaccia avviene in maniera diretta. Dopo aver eseguito le funzioni, il risultato viene adeguato ad una struttura dati adatta al frontend.

La soluzione scelta è stata Tweepy, una libreria per Python che ci fornisce un Wrapper per l'API di Twitter che rende il processo di comunicazione rapido.

La Web App è stata hostata tramite un provider denominato *"pythonanywhere"* che consente di eseguire il codice sui loro server.

WEB APP REALIZZATA CON FLASK

L'interfaccia è gestita con Flask, un micro-framework web scritto in Python, basato sullo strumento Werkzeug WSGI e con il motore di template Jinja2. Ha licenza BSD.

FRONTEND

L'interfaccia possiede una doppia autenticazione: una iniziale, dovuta ai permessi di Twitter, e la successiva dovuta all'accesso sulla piattaforma. Dopo aver effettuato l'accesso, l'interfaccia possiede dei pulsanti che permettono l'esecuzione delle funzionalità specifiche. I pulsanti sono collegati a delle route che eseguono delle richieste http GET e POST.

INSTALLAZIONE UTENTE

Il software non necessita di nessuna installazione da parte dell'utente poiché la piattaforma è stata hostata su un server e quindi non sono richiesti dei pre-requisiti per avviare l'applicazione, se non aprire il browser e recarsi sul sito web dedicato.

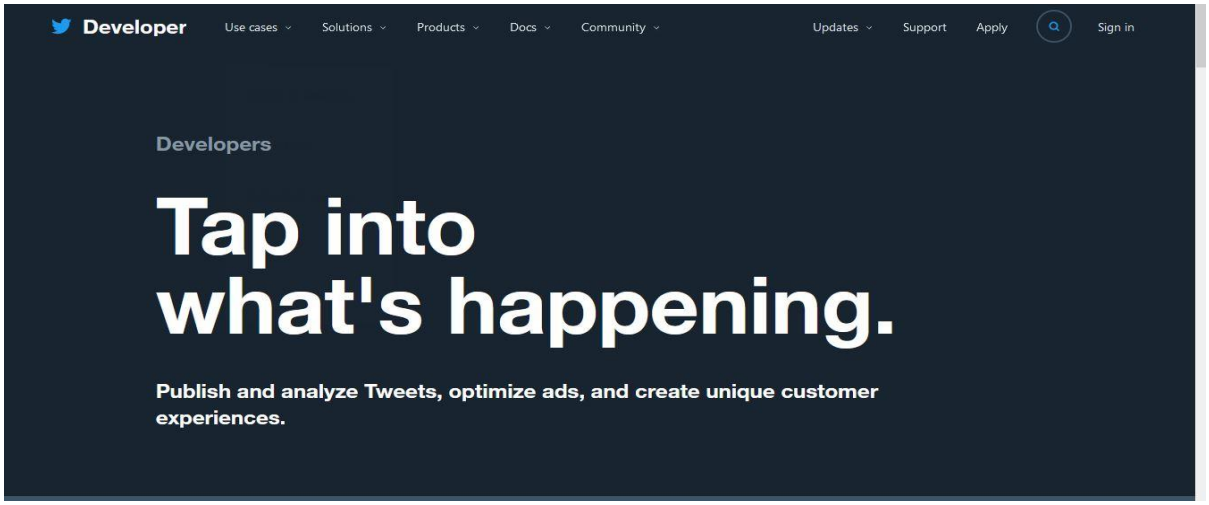
INSTALLAZIONE HOST

Per quello che concerne l'installazione della piattaforma sul provider bisogna eseguire i seguenti passi:

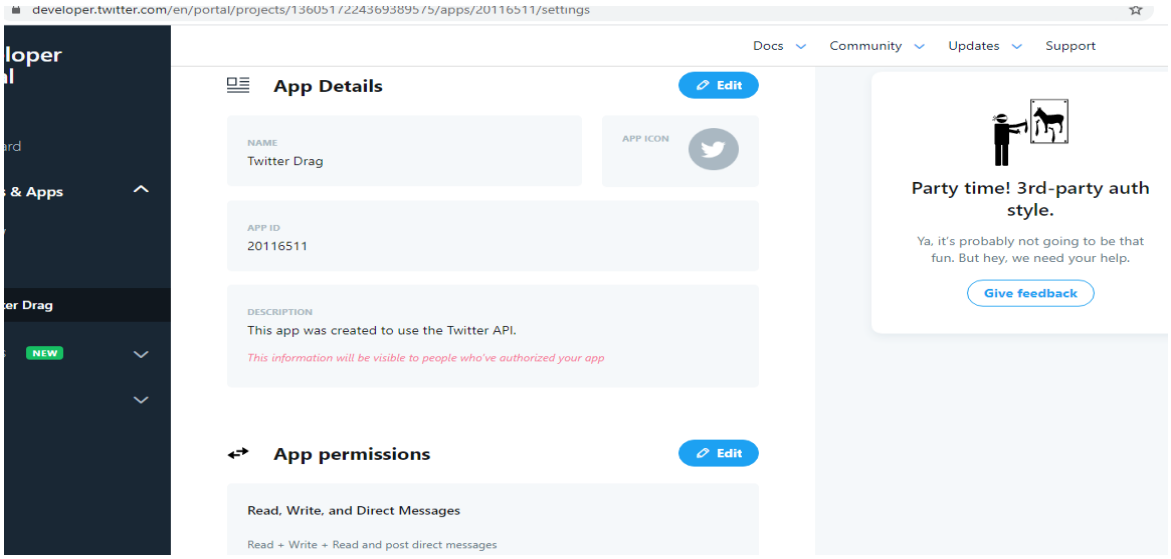
1. Iscrizione sul server: https://www.pythonanywhere.com
2. Creazione di una directory
3. Upload file “main.py” e “templates/index.html” presenti su Github.
4. Setting python: 3.8
5. Installazione librerie Phyton necessarie tramite la console bash della directory della Web App; il comando sarà: “*pip3.8 install –user [nomeLibreria]”.*
   Le librerie necessarie sono: Tweepy, Flask, Pandas, Haslib.

REGISTRAZIONE TWITTER DEVELOPER

Per effettuare l'accesso al software con il proprio account, è necessario iscriversi al seguente link: https://developer.twitter.com/en per poter avere accesso ai token al momento dell'esecuzione.



La schermata apparirà cosi dopo aver effettuato l'accesso:



I token sono presenti nella voce "App permission" e bisogna abilitare il flag per l'accesso.

AVVIO

Eseguire i comandi:

- aprire il browser

- accedere al sito: "framartin11.pythonanywhere.com"