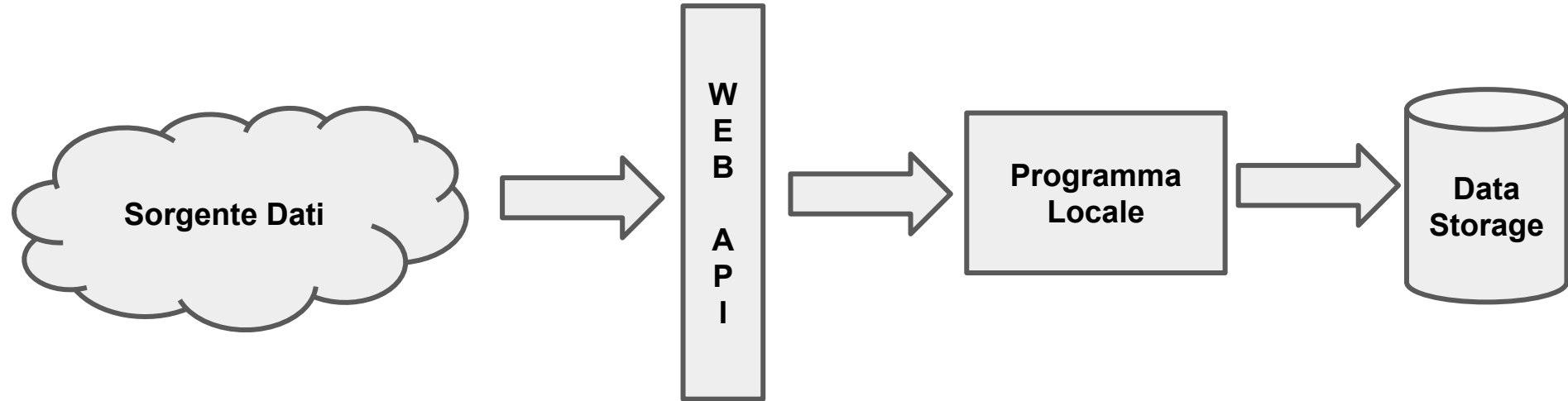


# Web API: estrazione dati da Twitter

Angelica Lo Duca  
angelica.loduca@iit.cnr.it

# Web API



Twitter  
Facebook  
YouTube  
...

Specifiche per ogni sorgente

Python, ...

Database  
CSV  
...

# API di Twitter

# Autenticazione

Twitter permette l'uso delle API solo agli sviluppatori registrati

Registrarsi sul sito:

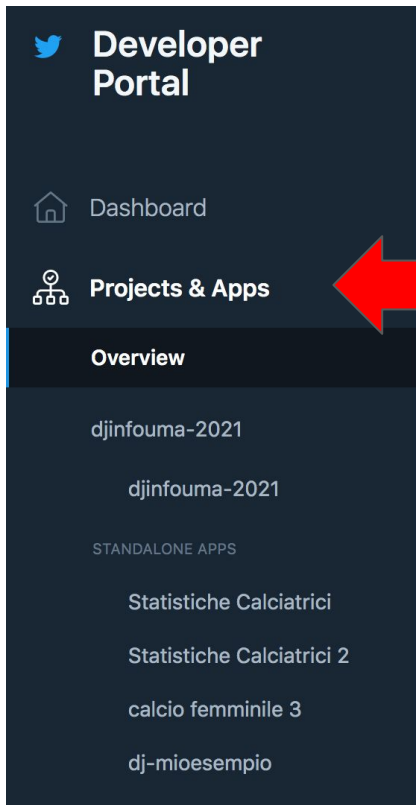
<https://developer.twitter.com/>

Registrarsi come sviluppatori - Cliccare su Apply e rispondere alle domande. Si può usare il seguente testo:

*I will use Twitter APIs for a master course in Data Journalism, hold at the University of Pisa, Italy. The use of the Twitter API is for learning purposes. Simple apps will be built to test the Twitter APIs.*

Accedere alla dashboard: <https://developer.twitter.com/en/portal/dashboard>

# Creazione di una nuova App



Selezionare Projects & Apps

Overview

djinfouma-2021

djinfouma-2021

STANDALONE APPS

Statistiche Calciatrici

Statistiche Calciatrici 2

calcio femminile 3

dj-mioesempio

NEW

Twitter API

Labs

About

Docs

Premium

Dev Environments

Subscriptions

Standard

djinfouma-2021

V1.1 ACCESS

V2 ACCESS

PROJECT APP

djinfouma-2021

Standalone Apps

V1.1 ACCESS

Standalone Apps live outside of Projects. This means that they can't use the the most current v2 Twitter API endpoints.

STANDALONE APPS

Statistiche Calciatrici

Statistiche Calciatrici 2

calcio femminile 3

dj-mioesempio

QUOTA: 5 OF 10 APPS

+ Create App



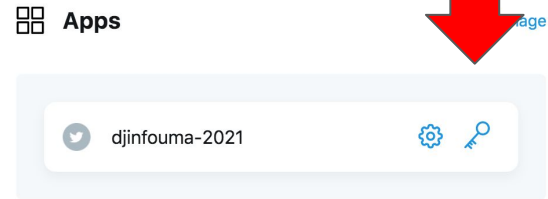
Selezionare  
Create App e  
seguire la  
procedura  
guidata

Alla fine della  
procedura  
salvare i token  
generati



1. Alla fine della procedura guidata la nuova app appare a sinistra sotto la voce Projects & Apps
2. Cliccare sulla nuova App e poi sul simbolo della chiave

3. Generare Access Token e Secret



Tweepy



# Tweepy

Libreria di Python per l'estrazione di dati da Twitter

Installare tweepy

- `pip install tweepy`
- `pip3 install tweepy`

Documentazione

<https://docs.tweepy.org/en/latest/>

# Tweepy Configurations

- **API**

- retrieve general data, including past data, user profiles and so on.

- **Stream**

- retrieve real time tweets

# Tweepy Configurations

- **API**

- retrieve general data, including past data, user profiles and so on.

- **Stream**

- retrieve real time tweets

# Creazione Oggetto API in Tweepy

```
import tweepy # importo la libreria

# definisco le credenziali
consumer_key = 'il_tuo_consumer_key'
consumer_secret = 'il_tuo_consumer_secret'
access_token = 'il_tuo_access_token'
access_token_secret = 'il_tuo_access_token_secret'

# crea un'istanza dell'oggetto Tweepy
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
```

# Ricerca Tweets

```
tweets = api.search_tweets(  
    q="#ChatGPT", # query da cercare  
    tweet_mode='extended', # recupero tutte le informazioni  
    lang='it' # specifico la lingua  
)
```

# Ricerca Tweets in un range temporale

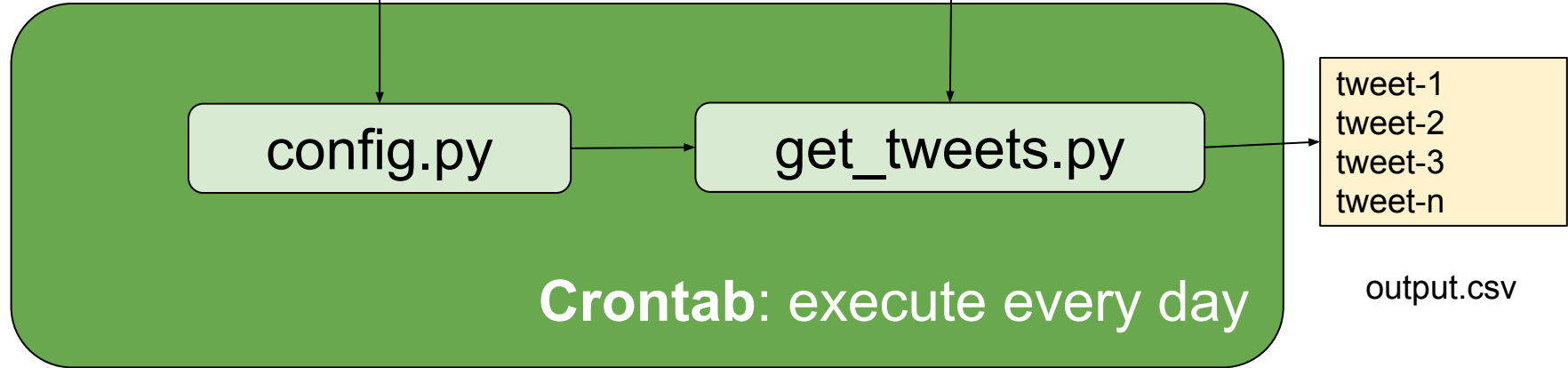
```
tweets = api.search_tweets(  
    q=f"#ChatGPT since: {yesterday} until: {today}",  
    tweet_mode='extended',  
    lang='it'  
)
```

# *Esempio*

Estrazione dei tweets del giorno precedente con hashtag #ChatGPT e automatizzazione per l'estrazione giornaliera



TWITTER\_CONSUMER\_KEY,  
TWITTER\_CONSUMER\_SECRET,  
TWITTER\_ACCESS\_TOKEN,  
TWITTER\_TOKEN\_SECRET





# config.py

```
TWITTER_CONSUMER_KEY = 'PUT HERE YOUR API KEY'
```

```
TWITTER_CONSUMER_SECRET = 'PUT HERE YOUR API SECRET'
```

```
TWITTER_ACCESS_TOKEN = 'PUT HERE YOUR ACCESS TOKEN'
```

```
TWITTER_ACCESS_TOKEN_SECRET = 'PUT HERE YOUR SECRET TOKEN'
```

# get\_tweets.py

Implementare il codice in un notebook jupyter e poi scaricarlo come file .py

The screenshot shows the Jupyter Notebook interface. The top bar displays the Jupyter logo, the filename 'get\_tweets', and the status 'Last Checkpoint: un'ora fa (autosaver)'. The 'File' menu is open, showing options like 'New Notebook', 'Open...', 'Make a Copy...', 'Save as...', 'Rename...', 'Save and Checkpoint', 'Revert to Checkpoint', 'Print Preview', 'Download as', 'Trusted Notebook', and 'Close and Halt'. The 'Download as' sub-menu is open, listing various file formats: AsciiDoc (.asciidoc), HTML (.html), LaTeX (.tex), Markdown (.md), Notebook (.ipynb), PDF via LaTeX (.pdf), REST (.rst), Python (.py), and Reveal.js slides (.slides.html). The 'Python (.py)' option is highlighted with a red box. The background shows code in a notebook cell, including imports for 'config', 'tweepy', and 'datetime', and a function definition for 'legge le credenziali salvate nel'.

# Cron Jobs

Il cron job è un sistema per programmare alcune azioni ripetitive.

Nel nostro esempio serve per scaricare ogni giorno i tweets del giorno precedente.

Richiede che la macchina su cui si esegue il cron job sia accesa nel momento in cui il cron job deve essere eseguito.

# Configurare il cronjob - Linux/ Mac OS

- aprire un terminale e scrivere `crontab -e`
  - questo comando apre un editor di testo per per inserire, modificare o cancellare un task
  - ogni task deve contenere l'orario e la data di esecuzione. L'ordine è minuti, ora, giorno del mese, mese, giorno della settimana. Si può usare l'\* per indicare tutti i valori
- inserire una nuova riga con
  - `00 01 * * * python /<path_to_file>/get_tweets.py`
  - questo esegue lo script una volta al giorno all'una di notte.

# Configurare il cronjob - Windows 10

- Seguire [questa procedura](#)