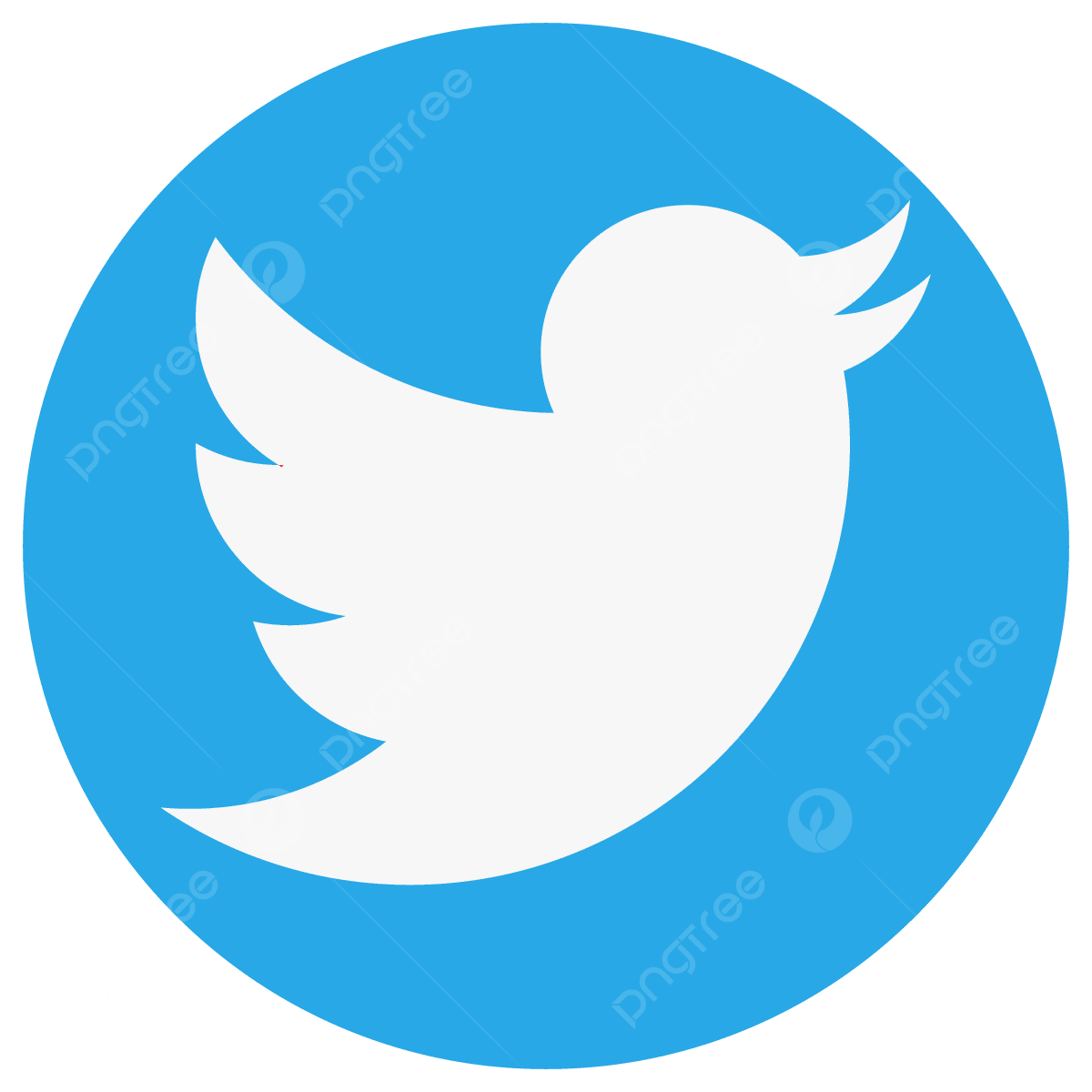
**How to scrape n number of tweets using snscrape**

## A comprehensive guide to scraping tweets coherent with Twitter



Today, data is scattered everywhere in the world. Especially in social media, there may be a big quantity of data on Facebook, Instagram, Youtube, Twitter, etc. This consists of pictures and films on Youtube and Instagram as compared to Facebook and Twitter. To get the real facts on Twitter, we need to scrape the data from Twitter and the data like (date, id, url, tweet content, user,reply count, retweet count,language, source, like count etc)from twitter.

**Introduction to snscrape**

Released on July 8, 2020, snscrape is a scraping tool for social networking services (SNS). It scrapes things like users, user profiles, hashtags, searches, threads, list posts and returns the discovered items without using Twitter’s API.

Interestingly, snscrape is not just for scraping tweets but also across various other social networking sites like Facebook, Instagram, Reddit, VKontakte, and Weibo (Sina Weibo).

**But first, is it even legal to scrape data from Twitter?**

Yes, it is very much legal to scrape data available on Twitter to the time it is used by the books. Say, for data analysis or understanding the market trend, perform sentiment analysis on a topic.

Twitter “ tolerates” polite crawlers. However, if the scraped data is publicly posted unconventionally, Twitter can shut down any API access you might have, and potentially take action against your account.

As our favourite phrase to answer any unknown space is, I’d say, ‘it depends on what are you scraping, how, and when.

Why snscrape?

Why are we even talking just about snscrape? There should be more options to do the job. Of course, there are. (or rather there were). And below are the problems with other tools or packages which leads us to snscrape

**Getting Started with snscrape**

**Requirements**

Python 3.8 or higher

libxml2 and libxslt

pandas

**Installing snscrape**

## pip3 install snscrape

However, this will not be the developer version. Instead, I recommend using the below command to download the snscrape dev version:

pip3 install git+<https://github.com/JustAnotherArchivist/snscrape.git>

**Note**: To run git cli commands, you have to have git installed before running your pip command. If you do not perform the said step, the above command will clone the repository from GitHub but will not execute if git is not installed.

Perfect, now that we’ve set up snscrape and peeripheral requirements, let’s jump right into using snscrape!

**Using snscrape**

Now, there are two ways of using snscrape

1. Using the command prompt, terminal (Converting JSON files for Python)
2. Using Python Wrapper

I prefer the Python Wrapper method because I believe it's easy to interact with data scraping, rather than engaging in a two-step process with the CLI. However, if you’re interested in knowing the process with CLI, you can refer from [here](https://github.com/MartinBeckUT/TwitterScraper/tree/master/snscrape/cli-with-python).

To explain better, **wrappers**aroundfunctions in Python allows modifying behavior of function or class. Basically, the wrapper wraps a second function to extend the behavior of the wrapped function, without permanently altering it.

# Introduction to Python

# Pandas in [Python](https://www.geeksforgeeks.org/python-programming-language/learn-python-tutorial/) is a package that is written for data analysis and

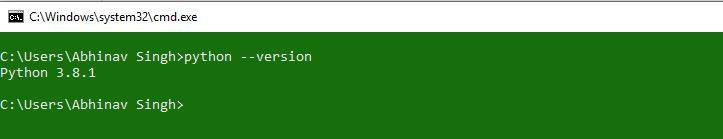
# manipulation. [Pandas](https://www.geeksforgeeks.org/introduction-to-pandas-in-python/) offer various operations and data structures

# to perform numerical data manipulations and time series. Pandas

# is an open-source library that is built over [Numpy](https://www.geeksforgeeks.org/python-numpy/) libraries. Pandas library is known for its high productivity and high performance. Pandas are popular because it makes importing and analyzing data much easier. Pandas programs can be written on any plain text editor like ****notepad, notepad++,****or anything of that sort and saved with a ****.py**** extension. To begin with, writing Pandas Codes and performing various intriguing and useful operations, one must have Python installed on their System.

# Python Installation

If Python is already installed, it will generate a message with the Python version available else install Python, please visit: **How to Install Python on**[**Windows**](https://www.geeksforgeeks.org/how-to-install-python-on-windows/)**or**[**Linux**](https://www.geeksforgeeks.org/how-to-install-python-on-linux/)**and**[**PIP**](https://www.geeksforgeeks.org/how-to-install-pip-on-windows/)**.**



## How to Download and Install Python Pandas

# Click below link:

# <https://www.geeksforgeeks.org/how-to-install-python-pandas-on-windows-and-linux/>

# Introduction to streamlit

# Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers.

## How to Download and Install Streamlit

# Click below link:

<https://docs.streamlit.io/library/get-started/installation>

**Introduction to PyMongo**

If you’d like to connect to MongoDB for Python, you’ll probably want to use Pymongo– the Python wrapper for MongoDB. When you use MongoDB and Python together, you can create, update and query collections with just a few simple lines of code. In this article, we’ll explain how to install the Python driver for MongoDB and connect to MongoDB using Python.

## How to Download and Install PyMongo

# Click below link:

<https://kb.objectrocket.com/mongo-db/how-to-install-pymongo-and-connect-to-mongodb-in-python-363#introduction>

**Attributes available in snscrape**

This image is your guide when deciding which attributes you need. I used only a few of them in my project but would highly recommend getting most of the attributes. The execution time remains the same, regardless of the number of attributes you declare



## ****Scraping tweets from a text search query****

Using the code below, we are scraping n number of tweets with the time range you want to scrape the data with the keywords . I then pulled attributes Date-time, Tweet id, url , Tweet content, User Name, Reply Count, Re-Tweet Count, Like Count, Language, Source from the tweet object.

Here is the code to scrape n number of tweets with the time range and keyword:

# importing all the required libraries

import streamlit as st

import snscrape.modules.twitter as sntwitter

import pandas as pd

import pymongo

import json

# creating title using streamlit

st.title("TWITTER SCRAPING")

client = pymongo.MongoClient("mongodb://localhost:27017")

database = client.Twitter\_scraping

collection = database.Scraped\_data

keyword = st.text\_input("ENTER THE KEYWORD OR HASHTAG")

max\_count = st.number\_input("MAXIMUM TWEET COUNT")

from\_date = st.date\_input("FROM DATE")

to\_date = st.date\_input("TO DATE")

if st.checkbox("DISPLAY SCRAPED DATA"):

    tweets\_list = []

    for i, tweet in enumerate(sntwitter.TwitterSearchScraper(f"{keyword} since:{from\_date} until:{to\_date}").get\_items()):

        if i > max\_count:

            break

        tweets\_list.append([tweet.date, tweet.id, tweet.url, tweet.content, tweet.user.username, tweet.replyCount,

                            tweet.retweetCount, tweet.lang, tweet.sourceLabel, tweet.likeCount])

    tweets\_df = pd.DataFrame(tweets\_list, columns=["DateTime", "TweetID", "URL", "Tweet Content", "UserName", "ReplyCount",

                                                   "ReTweet Count", "Language", "Source", "Like Count"])

    st.dataframe(tweets\_df)

    x = tweets\_df.to\_dict('records')

    def convert\_df(tweets\_df):

        return tweets\_df.to\_csv()

    csv = 'convert\_df(tweets\_df)'

    if st.download\_button(label="DOWNLOAD CSV", data=csv, file\_name="scraped\_data.csv", mime="text/csv"):

         st.write("DOWNLOADED CSV FILE SUCCESSFULLY")

    def convert\_df(tweets\_df):

        return tweets\_df.to\_json()

    json = 'convert\_df(tweets\_df)'

    if st.download\_button(label="DOWNLOAD JSON", data=json, file\_name="scraped\_data.json", mime="text/json"):

        st.write("DOWNLOADED JSON FILE SUCCESSFULLY")

    if st.button("UPLOAD DATA INTO MONGODB"):

        collection.insert\_many(x)

        for each in collection.find():

            print(each)

        st.write("UPLOADED DATA SUCCESSFULLY")

We have to build a solution that should be able to scrape the twitter data and store that in the database and allow the user to download the data with multiple data formats like csv and json and upload the data into mongodb.

A general execution time for the entire code could be anywhere between 1 minute — 40 minutes or even more, depending on the number of tweets fetched by your username or keyword query. If it takes longer than an hour, you might want to check your lines of code.

That’s it from my end for this blog. Thank you for reading!