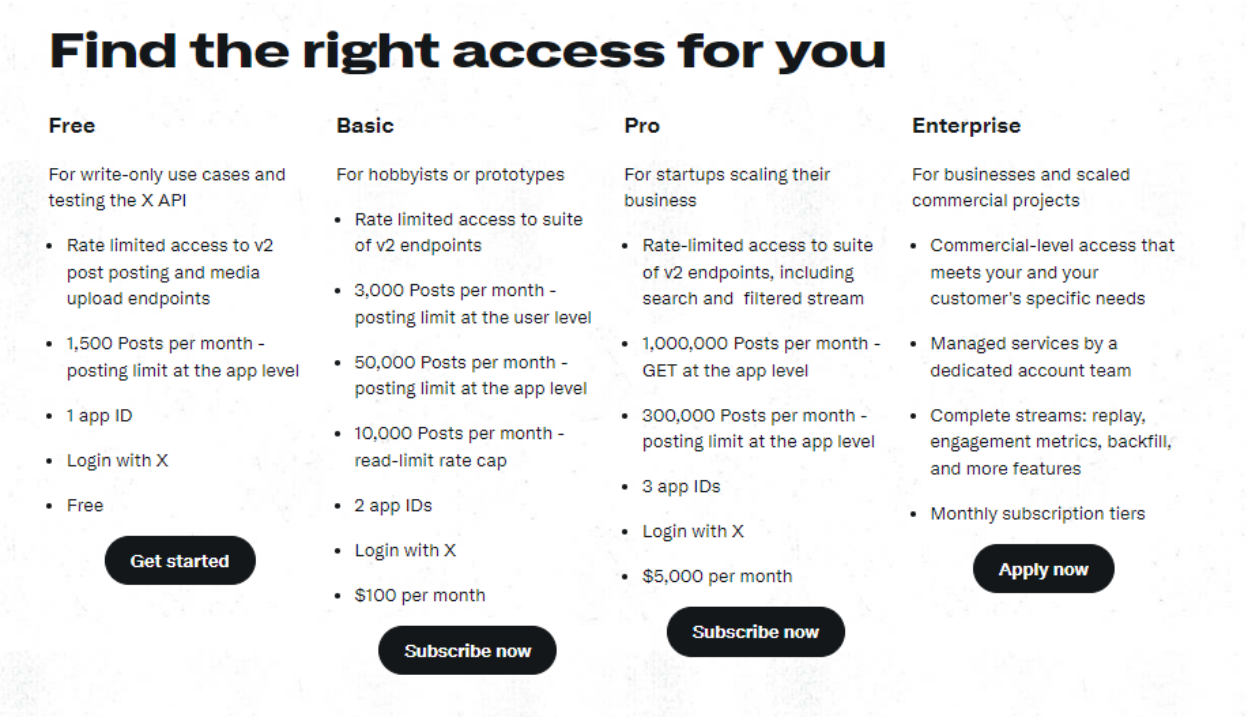
X Lessons Learned

The July 2024 Report

The EP election brief on X, part of the access://democracy project, underwent significant changes between its initial conceptual methodology and the final report. The project originally aimed to locate bot networks in the comments and mentions of popular Euro-skeptical parties on the platform. However, this scope had to be redefined to look at purely the speech of these parties and their accounts, rather than the accounts of possible bots. This was mainly due to the lack of clear resources and documentation about what is and isn't feasible with access to the X API, and the significant restrictions that have been placed on the API since 2023. In addition, X is not a platform often used for social media monitoring by the Digital Democracy team and was unfamiliar to navigate. The purpose of this document is to provide as much clarification as possible about how to work with the X API both methodologically and technically to better inform future research proposals.

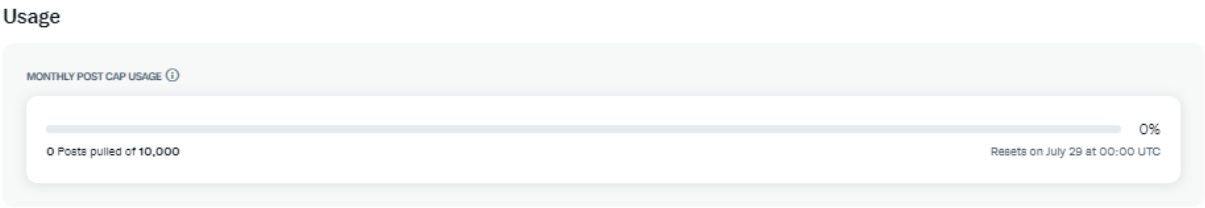
Setting Up Access

Since Elon Musk's purchase of the platform, the previously free API has been monetized to an excruciating degree, with three tiers of access available for purchase:



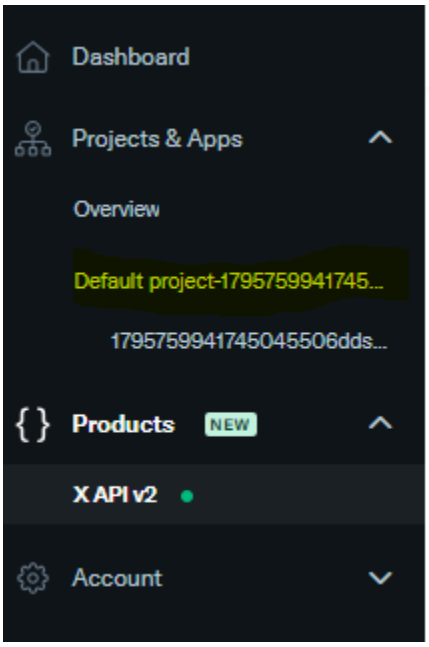
The free version is severely limited in access and unable to retrieve some of the most basic endpoints such as a user's unique ID (essential for retrieving information on an account's profile). This means for any serious analysis, the organisation must acquire at least one Basic

Developer account, which can retrieve most endpoints and up to 10,000 posts. It is possible to purchase multiple basic accounts to increase the level of access (the EP brief required 3) but this of course is dependent on funding. Because of this limitation, carefully consider the scope of your methodology in advance: 10,000 may seem like a lot but it will quickly deplete the larger the time frame and number of observed accounts. For reference, the earlier X report required 30,000 posts to retrieve 2 months of posts from 53 political accounts. There was some wiggle room with the data, but not much.

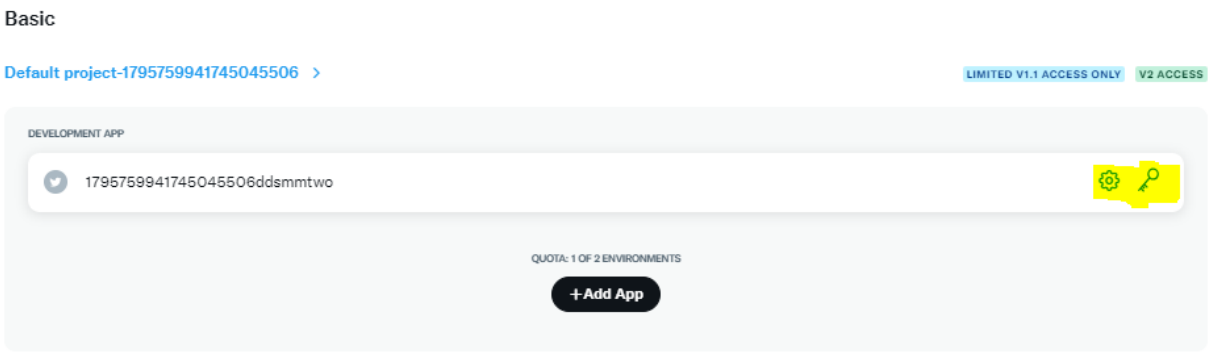


It should also be noted that the process of requesting funding for these accounts, their purchase, and set up, took quite some time (about 2 working weeks). Plan accordingly and request accounts far in advance.

When on the developer portal and signed in, you can create an App:



Then you can find your bearer token and access keys here:



You can see how these keys are used to access the API in the script below. This script was used for the previous analysis, and includes functions for retrieving the posts of a user within a certain time frame and how to retrieve the most recent mentions of that user.

Documentation and Working with the API

It can be difficult to find relevant documentation for accessing the X API using Python, as many tutorials and guides are from before the monetization. I've assembled the up to date links here, plus external tutorials that may be of use.

Retrieving Posts

Official Documentation: GET /2/tweets | Docs | Twitter Developer Platform (x.com)

The API access limit should be kept in mind for methodologies that require retrieving all posts from lots of accounts at once. Some can have only a dozen tweets in a certain time frame, others hundreds (this is especially the case for bot accounts which may post thousands of times a week) Without manually checking each account, this risks quickly depleting the post limit for a developer account. In addition, even if the access limit has not been reached, the rate limit of the API may timeout before a request is finished, as you are limited to 15 GET requests per 15 minutes for a basic account.

To address these issues, I recommend:

- Setting an artificial cap for the number of tweets retrieved from any given user. This way retrieving the tweets of one account won't risk accidentally using up your access token.
- Adding sleeper functions to retrieval requests so as not to overload the rate limit.

The data returned from the X API from a single tweet is quite significant, and contains information not just about the tweet content and engagement statistics, but also the statistics about the author (number of followers, username, location, number of tweets, and number of accounts followed). I recommend taking full advantage of this and making sure GET requests to the API retrieve all relevant information. Below is an example script:

X Analysis Documentation

The script above shows how to retrieve a user's posts within a specific time frame, all their relevant information, place it in a data frame, and then export that data. From here, usual analysis methods can be applied such as topic and sentiment analysis on the content of the tweets.

Tweepy