**Tweepy**

**An easy-to-use Python library for accessing the Twitter API**

## **Prerequisite**

In order to work with the Twitter API, you need to have a developer account and your API keys and tokens to connect to the API.

In order to work with Tweepy, make sure you have Python installed on your machine.Then, in the terminal run:

pip3 install tweepy

## **Twitter API**

The Twitter API gives developers access to most of Twitter’s functionality. You can use the API to read and write information related to Twitter entities such as tweets, users, and trends.

Technically, the API exposes dozens of HTTP endpoints related to:

* Tweets
* Retweets
* Likes
* Direct messages
* Favorites
* Trends
* Media

The Twitter API uses [OAuth](https://oauth.net/), a widely used open authorization protocol, to authenticate all the requests. Before making any call to the Twitter API, you need to create and configure your authentication credentials.

You can leverage the Twitter API to build different kinds of automations, such as bots, analytics, and other tools. Keep in mind that Twitter imposes certain restrictions and policies about what you can and cannot build using its API. This is done to guarantee users a good experience. The development of tools to spam, mislead users, and so on is forbidden.

The Twitter API also imposes rate limits about how frequently you’re allowed to invoke API methods. If you exceed these limits, you’ll have to wait between 5 and 15 minutes to be able to use the API again. You must consider this while designing and implementing bots to avoid unnecessary waits.

## **Creating Twitter API Authentication Credentials**

As we have previously seen, the Twitter API requires that all requests use OAuth to authenticate. So you need to create the required authentication credentials to be able to use the API. These credentials are four text strings:

1. Consumer key
2. Consumer secret
3. Access token
4. Access secret

If you already have a Twitter user account, then follow these steps to create the key, token, and secrets. Otherwise, you have to sign up as a Twitter user before proceeding.

Step 1: Apply for a Twitter Developer Account

Go to the [Twitter developer site](https://developer.twitter.com/) to apply for a developer account. Here, you have to select the Twitter user responsible for this account. It should probably be you or your organization.

Step 2: Create an Application

Twitter grants authentication credentials to apps, not accounts. An app can be any tool or bot that uses the Twitter API. So you need to register your app to be able to make API calls.

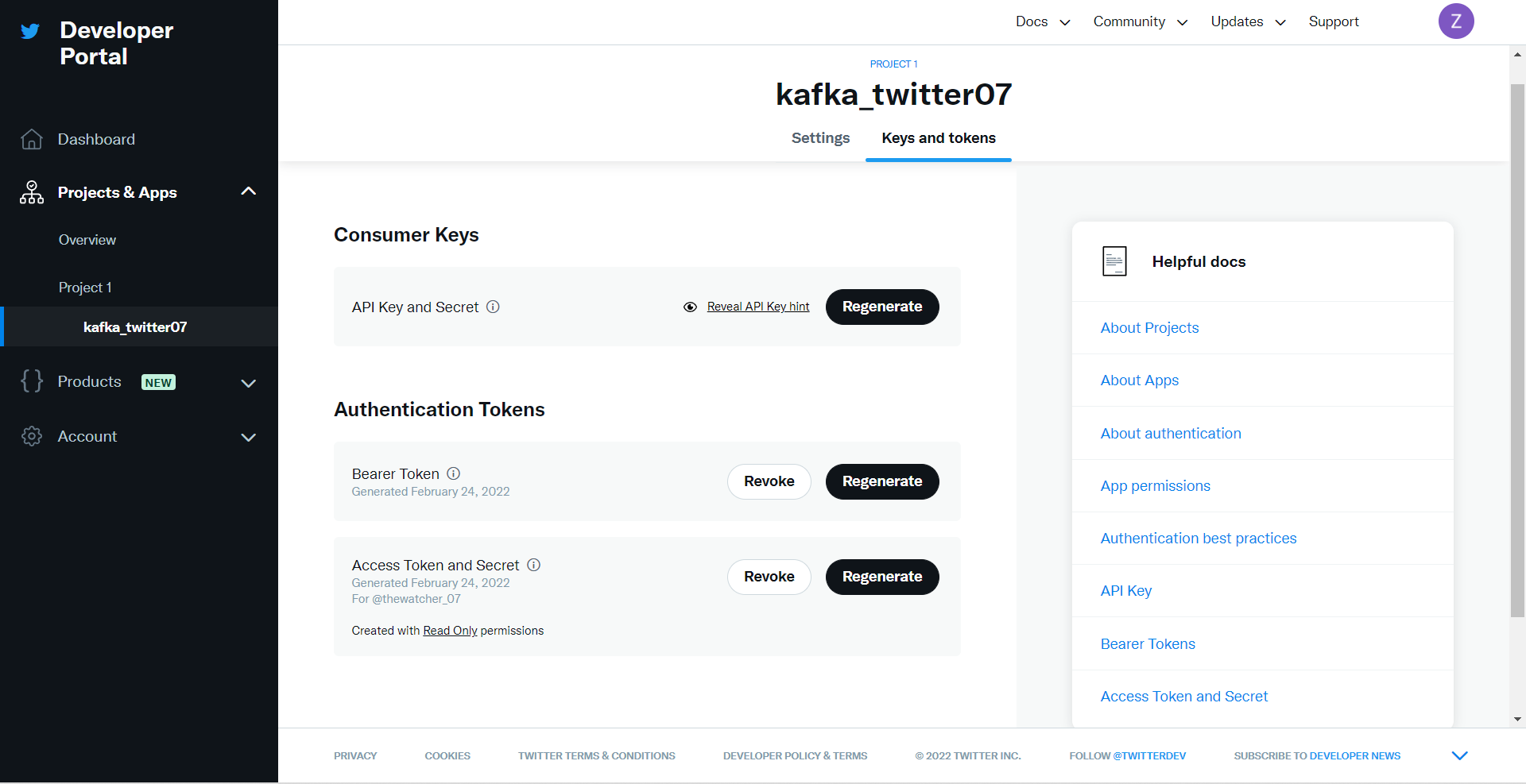
To register your app, go to [your Twitter apps page](https://developer.twitter.com/en/apps) and select the *Create an app* option.

You need to provide the following information about your app and its purpose:

* App name: a name to identify your application (such as *examplebot*)
* Application description: the purpose of your application (such as *An example bot for a Real Python article*)
* Your or your application’s website URL: required, but can be your personal site’s URL since bots don’t need a URL to work
* Use of the app: how users will use your app (such as *This app is a bot that will automatically respond to users*)

Step 3: Create the Authentication Credentials

To create the authentication credentials, go to [your Twitter apps page](https://developer.twitter.com/en/apps).



After generating the credentials, save them to later use them in your code.

## **What Is Tweepy?**

[Tweepy](https://github.com/tweepy/tweepy) is an open source Python package that gives you a very convenient way to access the Twitter API with Python. Tweepy includes a set of classes and methods that represent Twitter’s models and API endpoints, and it transparently handles various implementation details, such as:

* Data encoding and decoding
* HTTP requests
* Results pagination
* OAuth authentication
* Rate limits
* Streams

If you weren’t using Tweepy, then you would have to deal with low-level details having to do with HTTP requests, data serialization, authentication, and rate limits. This could be time consuming and prone to error. Instead, thanks to Tweepy, you can focus on the functionality you want to build.

Almost all the functionality provided by Twitter API can be used through Tweepy.