## Introduction

This guide explains how to post tweets programmatically using Python and Tweepy on Ubuntu. Additionally, the script includes features for preventing duplicate tweets, verifying posted tweets, and handling Twitter API rate limits.

## Prerequisites

Before proceeding, ensure you have:

✅ A Twitter Developer Account with API credentials  
✅ Python 3 installed on your Ubuntu system  
✅ Internet connection to install dependencies

## Step 1: Install Python on Ubuntu

Check if Python is installed by running:

python3 --version

If Python is not installed, install it using:

sudo apt update  
sudo apt install python3

Install pip for managing Python packages:

sudo apt install python3-pip

## Step 2: Install Tweepy

Install Tweepy using pip:

pip3 install tweepy

## Step 3: Set Up Twitter Developer Account

1. Visit the Twitter Developer Portal and log in.  
2. Create a new project and app.  
3. Generate and save API credentials: API Key, API Secret, Access Token, and Access Token Secret.

## Step 4: Write the Python Script to Post and Verify a Tweet

Create a Python script file:

mkdir twitter\_bot  
cd twitter\_bot  
nano twitter\_bot.py

Paste the following updated script into twitter\_bot.py:

import tweepy  
import time  
  
# Replace these with your API credentials  
API\_KEY = "your\_api\_key"  
API\_SECRET = "your\_api\_secret"  
ACCESS\_TOKEN = "your\_access\_token"  
ACCESS\_TOKEN\_SECRET = "your\_access\_token\_secret"  
BEARER\_TOKEN = "your\_bearer\_token"  
  
# Authenticate with Twitter API  
client = tweepy.Client(  
 bearer\_token=BEARER\_TOKEN,  
 consumer\_key=API\_KEY,  
 consumer\_secret=API\_SECRET,  
 access\_token=ACCESS\_TOKEN,  
 access\_token\_secret=ACCESS\_TOKEN\_SECRET  
)  
  
def post\_tweet\_and\_verify(tweet\_text):  
 """Posts a tweet, ensures uniqueness, and verifies it by checking the user's timeline."""  
 try:  
 # Prevent duplicate tweets by adding a timestamp  
 tweet\_text = tweet\_text + " " + str(time.time())  
  
 # Step 1: Post the tweet  
 response = client.create\_tweet(text=tweet\_text)  
 tweet\_id = response.data['id']  
 print(f"✅ Tweet posted successfully! Tweet ID: {tweet\_id}")  
  
 # Step 2: Get user ID  
 user\_id = client.get\_me().data['id']  
  
 # Step 3: Fetch recent tweets from user's timeline  
 tweets = client.get\_users\_tweets(user\_id, max\_results=10)  
  
 # Step 4: Verify if the tweet appears in the timeline  
 tweet\_found = False  
 for tweet in tweets.data:  
 if tweet.id == tweet\_id:  
 tweet\_found = True  
 print(f"✅ The tweet was successfully posted: {tweet.text}")  
 break  
  
 if not tweet\_found:  
 print("❌ The tweet was not found in the user's timeline.")  
  
 except tweepy.TooManyRequests:  
 print("❌ ERROR: Too many requests. Retrying in 15 minutes...")  
 time.sleep(15 \* 60) # Wait for 15 minutes before retrying  
 post\_tweet\_and\_verify(tweet\_text)  
  
 except tweepy.TweepyException as e:  
 print(f"❌ ERROR: {e}")  
  
# Take user input and post a tweet  
tweet\_content = input("Enter your tweet: ")  
post\_tweet\_and\_verify(tweet\_content)

## Step 5: Run the Python Script

Run the script using:

python3 twitter\_bot.py

## Step 6: Error Handling

The script handles several errors, including:

- 401 Unauthorized: Incorrect API credentials, check API keys.  
- 403 Forbidden: Duplicate tweet, modify content.  
- Rate Limit Exceeded: Wait before retrying, script auto-retries in 15 minutes.

## Conclusion

By following this guide, you have:  
✅ Set up a Twitter Developer account  
✅ Installed Tweepy on Ubuntu  
✅ Written a Python script to post and verify tweets  
✅ Implemented error handling and automated tweeting  
  
You are now equipped to programmatically post and manage tweets on Twitter using Python and Tweepy!