

Twitter Tracery Bot

(CheapBotsDoneQuick Alternative)

[Source Code - GitHub](#)

Author - [@loonathedorm](#)



This is a quick guide on how to set-up Twitter-Quotes-Bot either using a free platform called [‘Replit’](#) or it can also be hosted on a server if you have access to one.

INDEX

Set-Up Using Replit ---- [Page 2](#)

Set-Up For Linux ---- [Page 15](#)

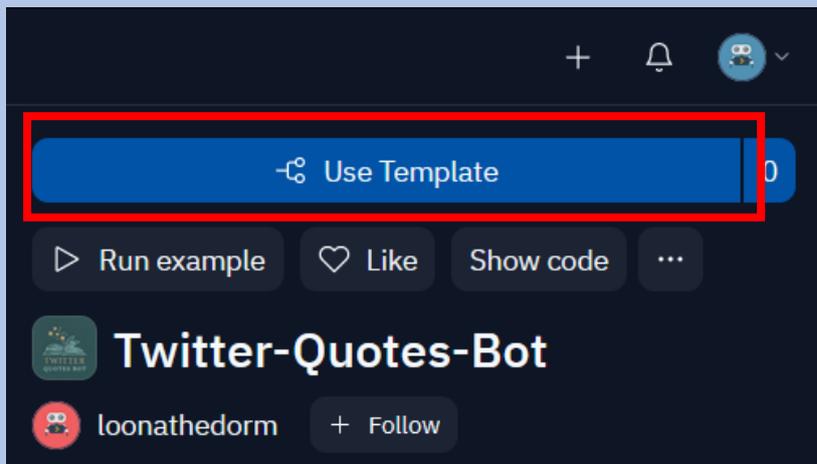
Basic Troubleshooting ---- [Page 17](#)

Updating the Bot ---- [Page 18](#)

Set-Up Using Replit :



1. Sign up on replit.com
2. Close/Skip/Dismiss any welcome screens that might pop up
3. Go to <https://replit.com/@loonathedorm/Twitter-Tracery-Bot>
4. Click on 'Use Template'



5. In the new window that shows up, change the name to anything you like. Then, Click 'Use Template' again.

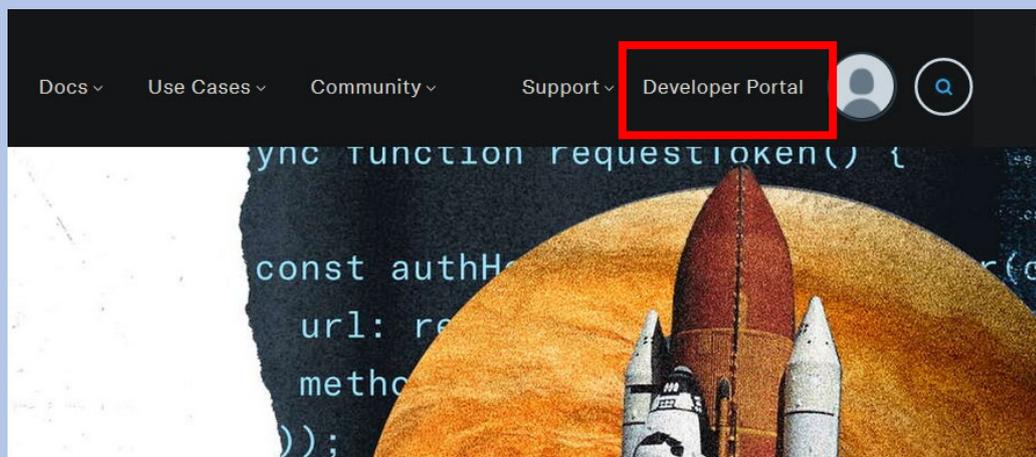


6. Dismiss any welcome screen or any screen asking for missing secrets/details.

- Now we're moving towards setting up our Twitter Developer Account.
- We need to get twitter developer account credentials.

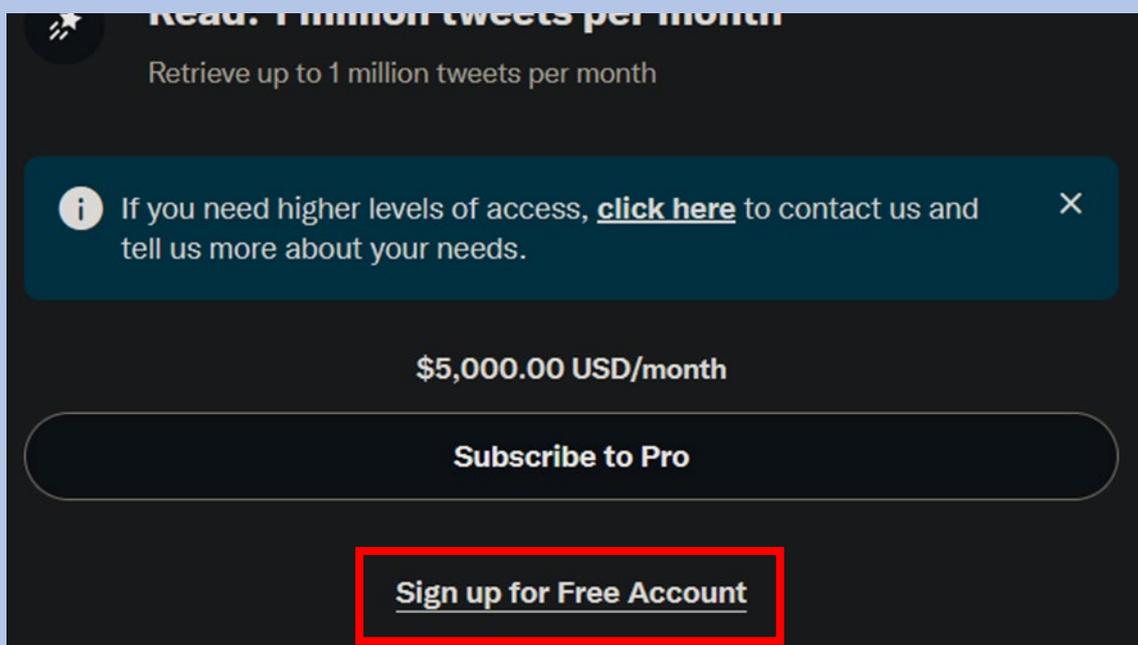
7. Make sure you're logged in to your Twitter bot account and navigate to developer.twitter.com

8. Click on 'Developer Portal' on the top



9. Scroll down and click 'Sign up for Free Account'

We do not need the expensive af paid plans.



- Now you need to describe your bot's functionality in 250+ words.

10. Tick all the checkboxes and click submit.

Developer agreement & policy

Describe all of your use cases of Twitter's data and API:
We need this information for data protection. [Learn more](#)

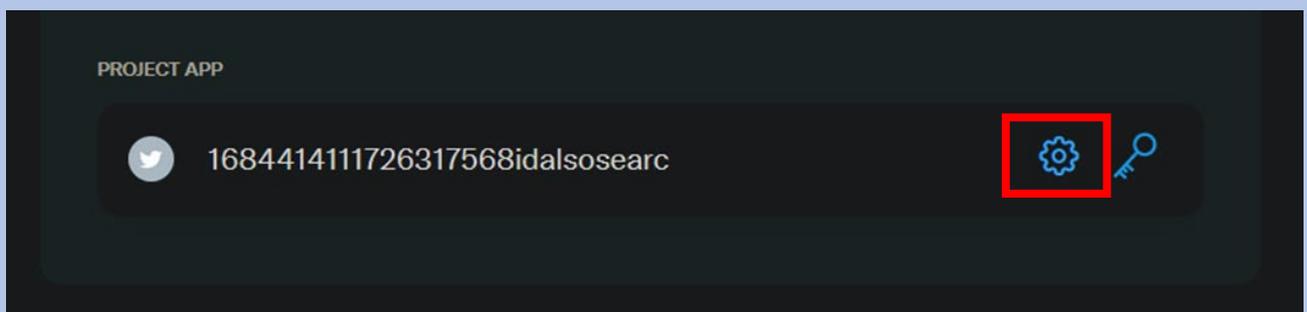
This is a twitter account that was made to post text based lyrics and/or quotes on twitter at set time intervals throughout the day. This account does not use the API to spam any messeges, nor does it abuse the api in any other way. The account also does not send any unsolicited DMs or mentions to other users.

- You understand that you may not resell anything you receive via the Twitter APIs
- You understand your Developer account may be terminated if you violate the Developer Agreement or any of the Incorporated Developer Terms
- You accept the Terms & Conditions

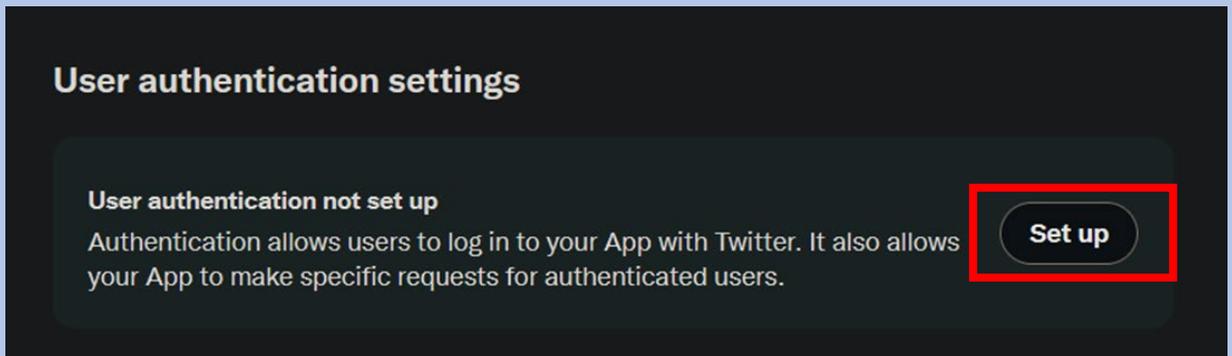
By clicking on the box, and by otherwise accessing or using any Licensed Material, you indicate that you have read and agree to this [Developer Agreement](#) and the [Twitter Developer Policy](#)

[Back](#) [Submit](#)

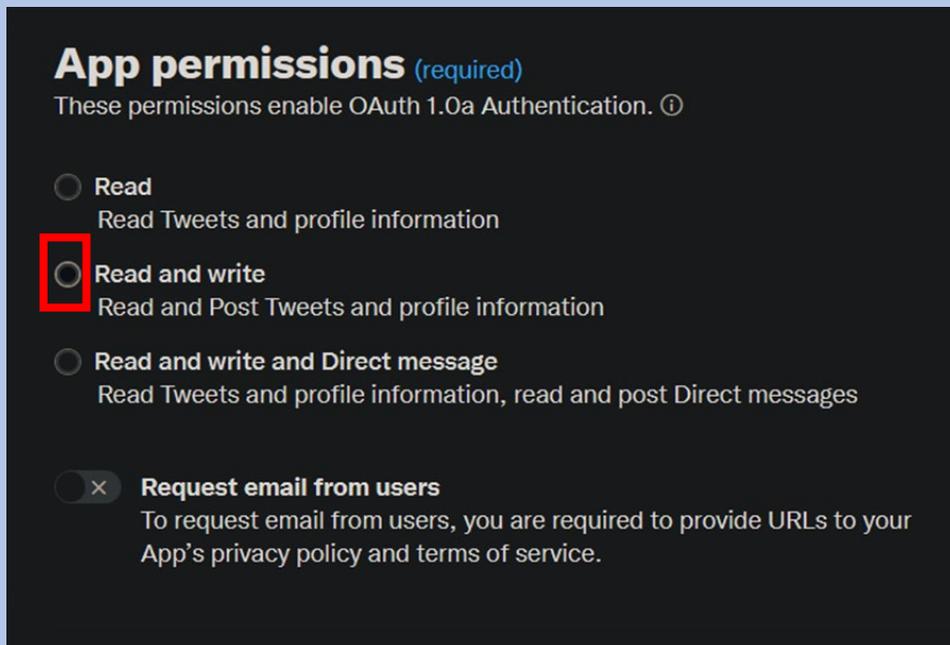
11. Now on the projects window, click the settings gear icon besides the default project app.



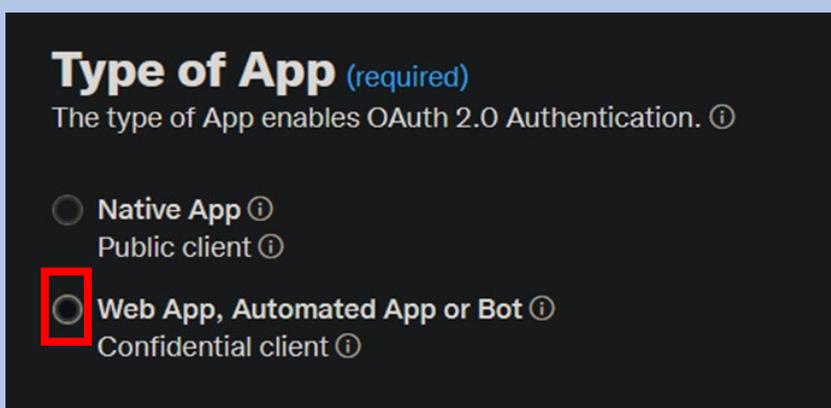
12. Scroll down to 'User Authentication Settings' and click 'Set Up'



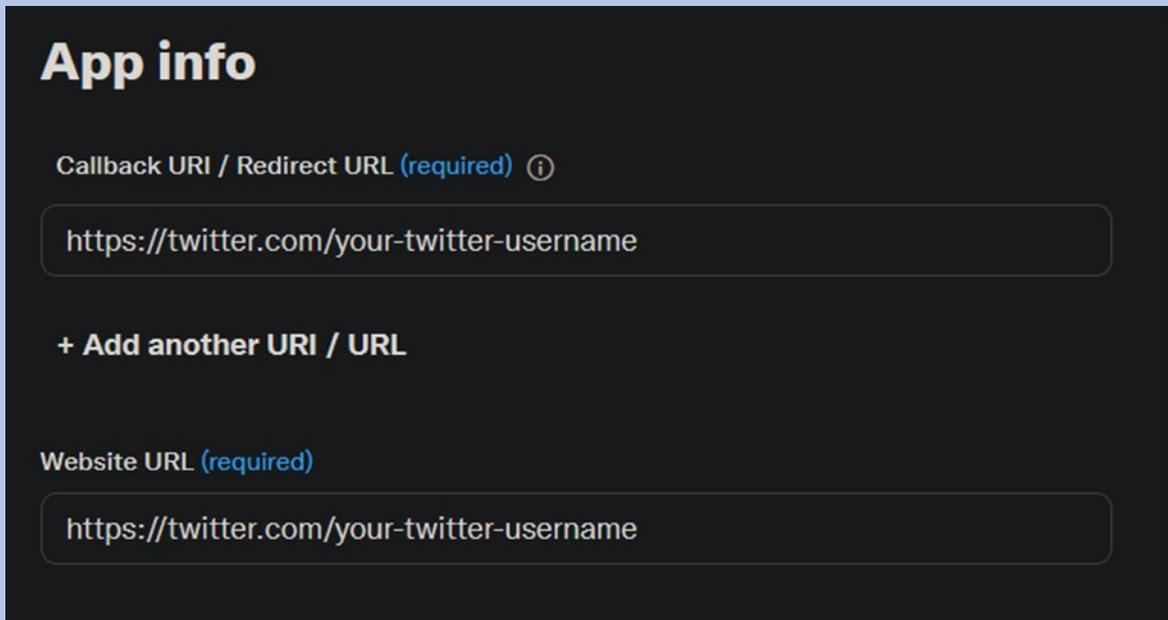
13. Tick 'Read and Write'



14. Tick 'Web App, Automated App or Bot'



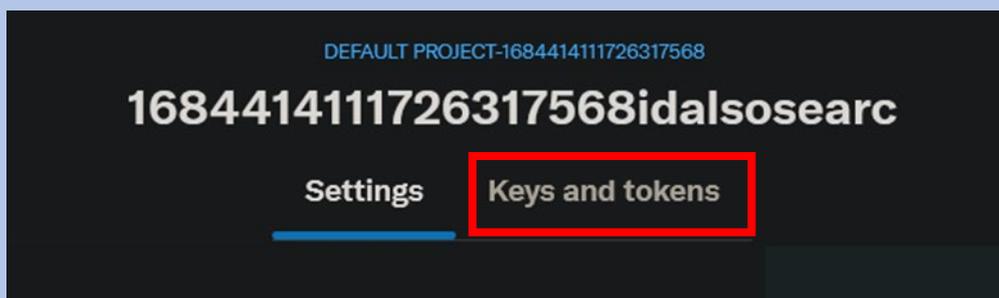
15. Under App Info, Enter your Twitter profile URL for the Callback URL and Website URL. All other fields are optional so we skip those and click 'Save'



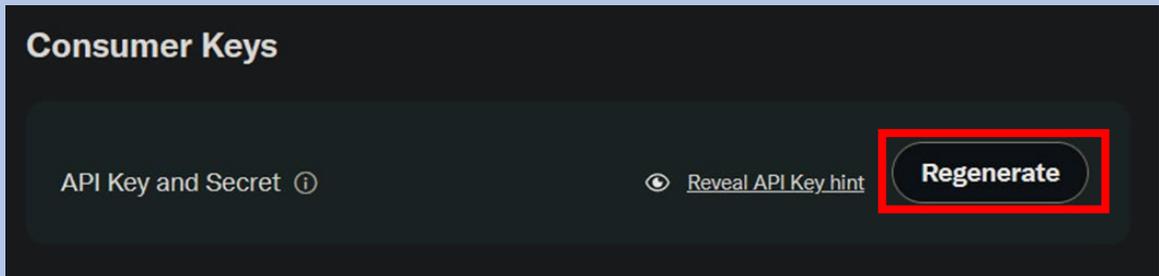
- A window shows up with OAuth Client ID and secret. You can save these if you want but we do not need them for our bot and you can always re-generate these if you need them in the future.

16. Click Done

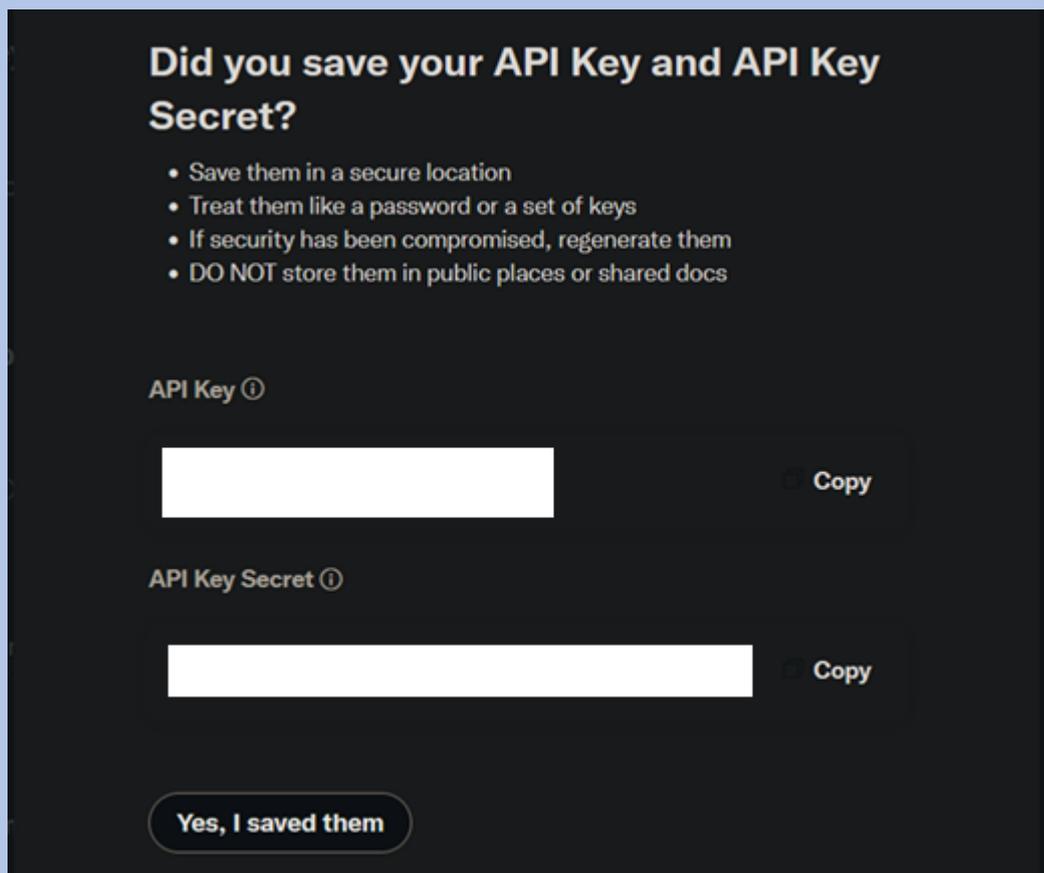
17. Now under your project app, click 'Keys and Tokens'



18. Under 'Consumer Keys' click 'Regenerate' for API Key and Secret.

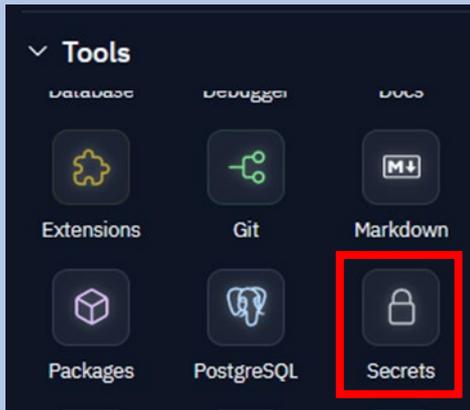


19. Save both the keys that pop-up in a notepad, we need these in the next steps. These keys are not to be shared with anyone as they allow direct access to your Twitter account.

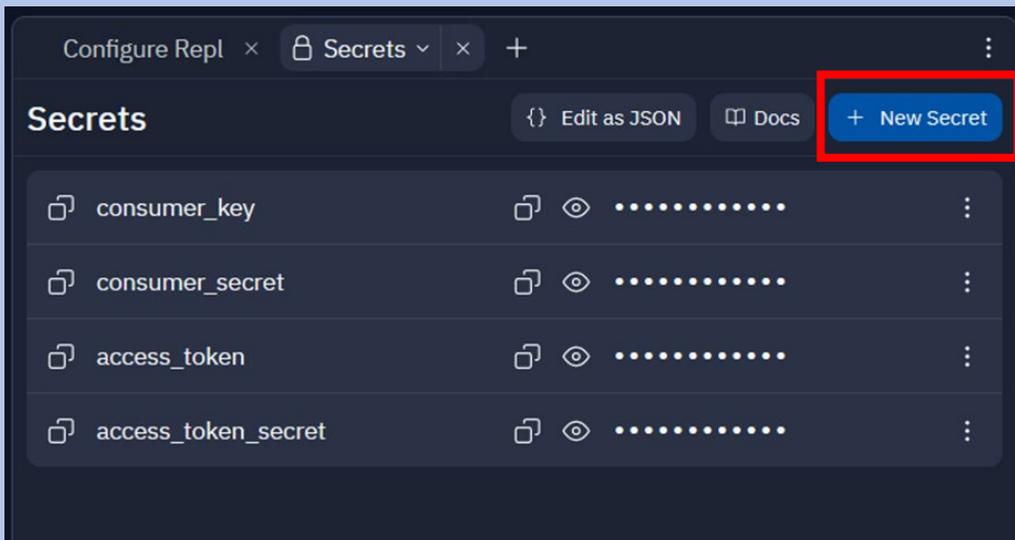


20. Do the same for Access Token and Secret under Authentication Tokens and save these as well.

21. Now come back to replit, Under Tools, click on Secrets.



22. In the secrets window on the right side, we need to add 4 secrets as shown in the picture. To add a secret, click 'New Secret' (If the secrets names already exist, just fill in the appropriate values.)



- The names and values are as follows:

consumer_key = API KEY

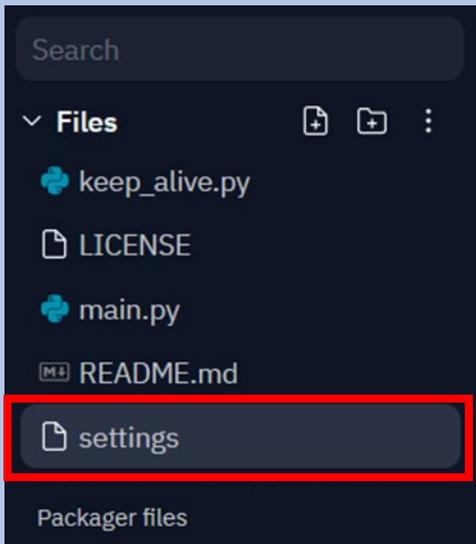
consumer_secret = API KEY SECRET

access_token = ACCESS TOKEN

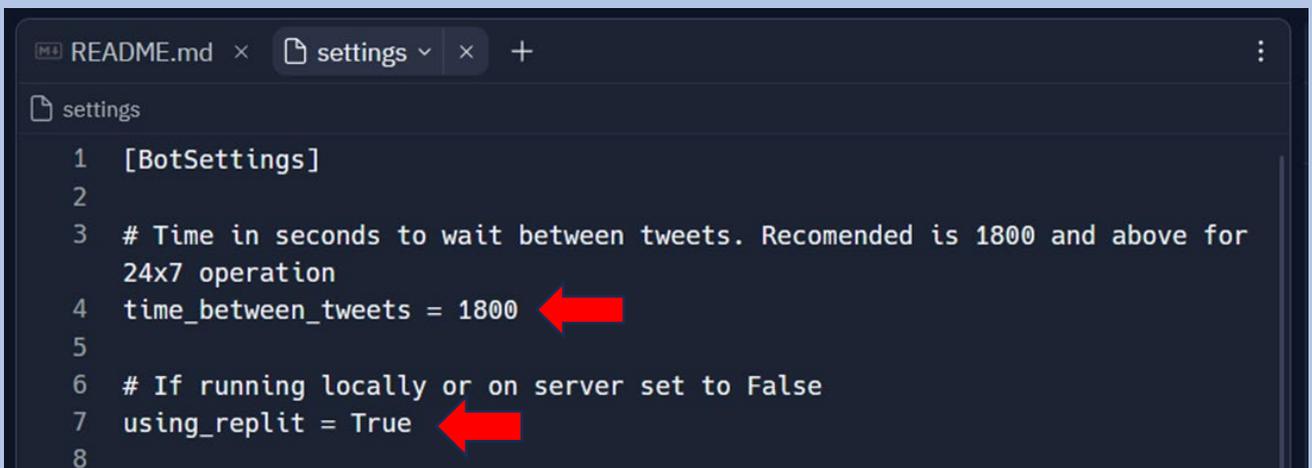
access_token_secret = ACCESS TOKEN SECRET

23. Enter the values that we saved from previous steps.

24. Now come to the Files tab on the left side. Click on the settings file.



25. Change the 'time_between_tweets' value on line 4 to whatever interval you want. This value is in seconds. Recommended to set something above 1800.

A screenshot of a code editor window. The window title bar shows 'README.md' and 'settings'. The editor displays the contents of the 'settings' file with line numbers 1 through 8. The code is as follows:

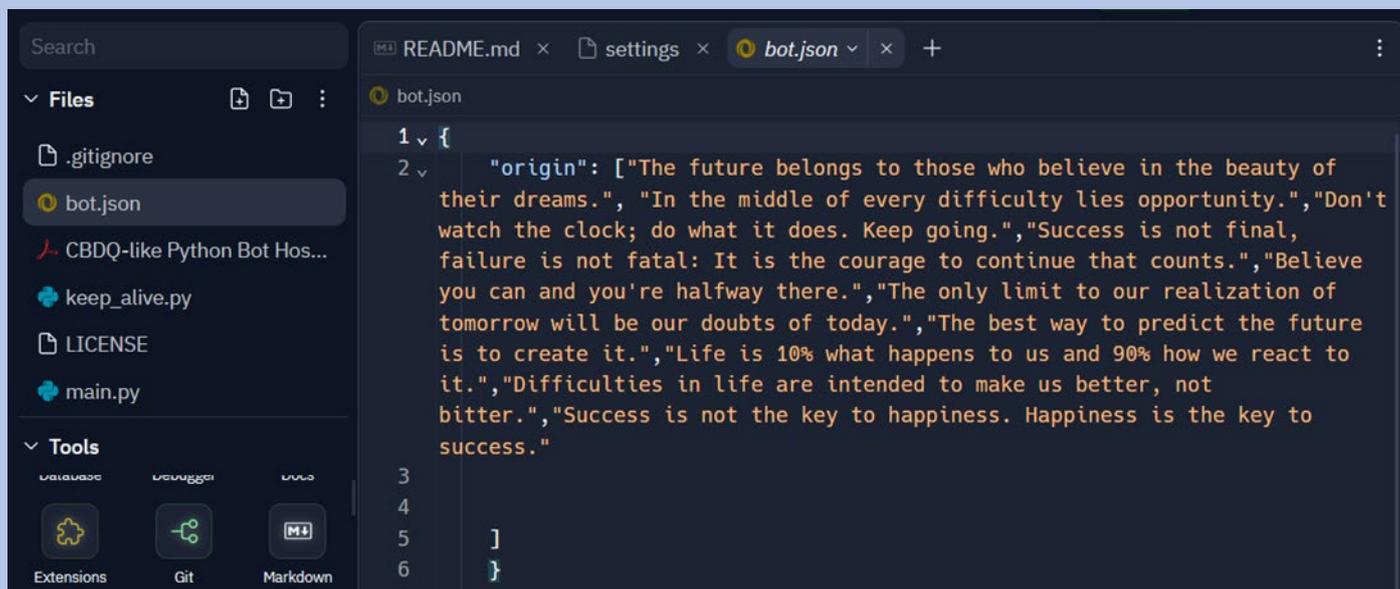
```
1 [BotSettings]
2
3 # Time in seconds to wait between tweets. Recommended is 1800 and above for
4 # 24x7 operation
5 time_between_tweets = 1800
6
7 # If running locally or on server set to False
8 using_replit = True
```

Two red arrows point to the value '1800' on line 4 and the value 'True' on line 7.

26. Make sure line 7 'using_replit' says True.

- You can change options in **[OTHER SETTINGS]** as per your preference
- For replit you do not need to make any further changes to the settings file.

- 27. Now from files, click on the bot.json file. It already contains some default text.**

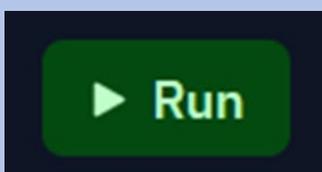


The screenshot shows a code editor interface with a dark theme. The left sidebar displays a file explorer with a 'Files' section containing .gitignore, bot.json (selected), CBDQ-like Python Bot Hos..., keep_alive.py, LICENSE, and main.py. Below it is a 'Tools' section with icons for Extensions, Git, and Markdown. The main editor area shows the bot.json file with the following JSON content:

```
1 {  
2   "origin": ["The future belongs to those who believe in the beauty of  
   their dreams.", "In the middle of every difficulty lies opportunity.", "Don't  
   watch the clock; do what it does. Keep going.", "Success is not final,  
   failure is not fatal: It is the courage to continue that counts.", "Believe  
   you can and you're halfway there.", "The only limit to our realization of  
   tomorrow will be our doubts of today.", "The best way to predict the future  
   is to create it.", "Life is 10% what happens to us and 90% how we react to  
   it.", "Difficulties in life are intended to make us better, not  
   bitter.", "Success is not the key to happiness. Happiness is the key to  
   success."  
3  
4  
5 }  
6  
7
```

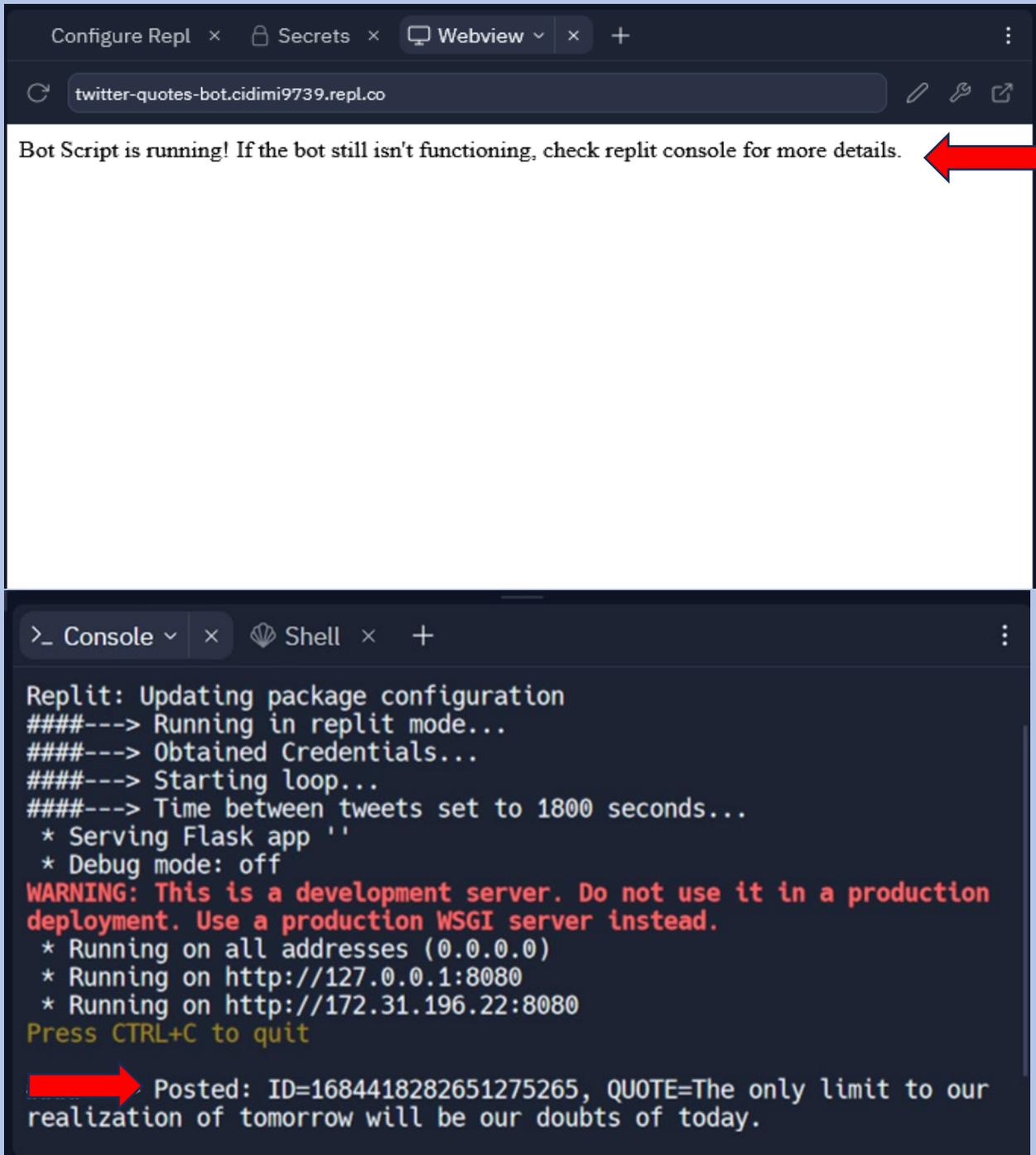
- 28. Replace the contents of this file with the contents of your json file that was being used with CheapBotsDoneQuick. (Tracery Grammar format)**

- 29. Now we are all set up. Time to run the bot. Click on the green 'Run' button on the top.**



- Keep an eye on the console window at the bottom right corner. It will show you the updates and info. Give it a few minutes to set everything up.

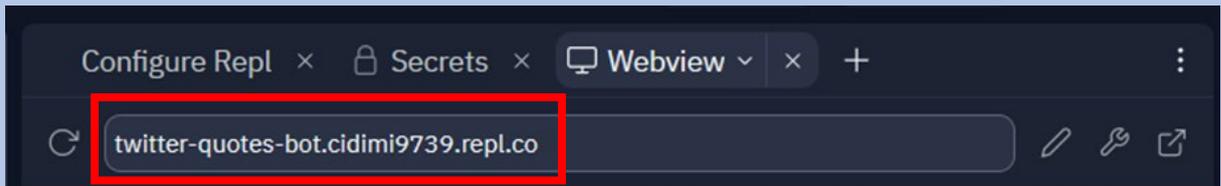
- When it's successful you should see something similar to the picture. You should see below shown message on the top window and the info for the first tweet that your bot has posted in the bottom console window.



- You should also see a new tweet on your Twitter profile.

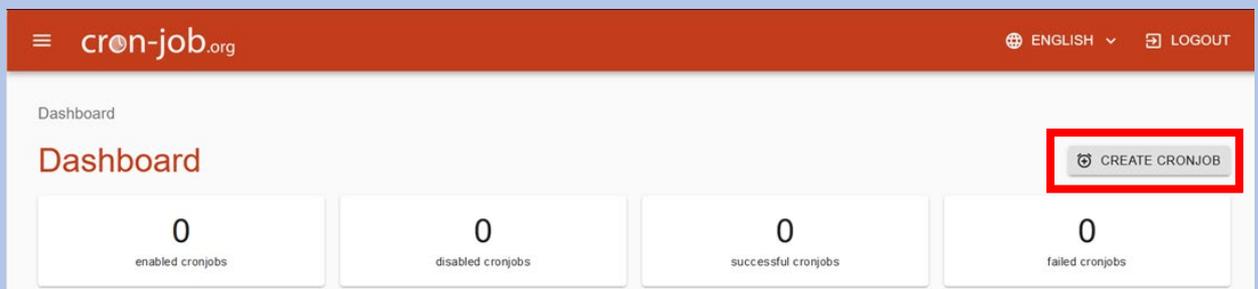
- Now we need to set-up another service that will keep the bot awake.

30. From the WebView window on the top, we need to copy the URL shown. (I'd recommend not sharing this url anywhere else publicly)



31. Go to cron-job.org and sign up. (You can just use random details and a secondary email for this)

32. On the dashboard, click 'CREATE CRONJOB' on the right side.



33. Enter details as shown below.

- **Title:** 'Anything you want'
- **URL:** Paste the URL we copied from replit earlier, include 'https://' at the beginning of the URL to avoid any errors.

34. Make sure 'Enable Job' is turned on. (Refer next picture)

35. Set 'Execution Schedule' to 1 minute.

Create cronjob

COMMON ADVANCED

Title
Twitter Quotes Bot

URL *
https://twitter-quotes-bot.cidimi9739.repl.co/

Folder
-

Enable job Save responses in job history

Execution schedule

Every 1 minute(s)
 Every day at 0 : 00

36. Next, in the notify section you can enable the first option for email notifications if/when the bot goes down.
37. Make sure the third option is disabled.

Schedule expires

Notify me when...

execution of the cronjob fails

execution of the cronjob succeeds after it failed before

the cronjob will be disabled because of too many failures

TEST RUN CREATE

38. Click Create.

- You should now see your monitor and its uptime on your dashboard and in the Cronjobs section. This should keep the Twitter bot running constantly.

Cronjobs

Cronjobs

MANAGE FOLDERS CREATE CRONJOB

Title, URL	Last execution	Next execution	Actions
<input type="checkbox"/>  Twitter Quotes Bot https://twitter-quotes-bot.cidimi9739.repl.co/	-	Today at 10:03:00 AM	<input type="button" value="HISTORY"/> <input type="button" value="EDIT"/>

- Note : Since replit is a free service, I've noticed that regardless of this monitor, the bot might go down sometimes but should also come back online automatically.
- For full 24x7 uptime you will need to host the bot on a Linux cloud server or something similar but that will most likely be a paid service.

----- End of Replit Guide -----

Set-Up for Linux :

1. Clone the GitHub repo and move into its folder:

```
git clone https://github.com/loonathedorm/Twitter-Tracery-Bot.git && cd Twitter-Tracery-Bot
```

2. Run the following command to install the required packages (Assuming you already have python installed):

```
pip3 install tweepy flask tracery3 colorama
```

3. Follow steps 12 – 25 from Replit Section to obtain Twitter Developer Account credentials and save them in a notepad.

4. Open settings file mentioned in step 29 of the replit section:

- a. Add Twitter Developer credentials into the settings file under the appropriate options :
 - i. consumer_key = API KEY
 - ii. consumer_secret = API KEY SECRET
 - iii. access_token = ACCESS TOKEN
 - iv. access_token_secret = ACCESS TOKEN SECRET
- b. Edit time interval value in line 4 as per your choice. Recommended above 1800. The value is in seconds.
- c. Set value of using_replit on line 7 to `False`
- d. You can change options in [**OTHER SETTINGS**] as per your preference

5. Open the bot.json file.

- a. Replace the sample quotes/lyrics already present in the file with your own content.
- b. Each individual quote/lyric should be enclosed in double quotes "Like This" and separated by a comma as done in the sample content.
- c. The bot.json file follows the format used in CheapBotsDoneQuick. So if you have an existing file from there, it can be used here directly.

6. Time to run the bot. We will be using the screen command to run the bot in background on our Linux server :

- a. Make sure you have screen installed:

```
sudo apt install screen
```

- b. Start new session:

```
screen -S TwitterBot
```

- c. Run the bot:

```
python3 main.py
```

7. Hit Ctrl+A+D to exit the screen session, the bot should continue to run in the background.

- To get back to the bot you can use the command `screen -r`

----- End of Linux Guide -----

Basic Troubleshooting :

- If your bot isn't working as expected, the easiest way to identify any issues is to scroll through the output in the console window on replit. You will probably see what the exact issue is.
- I have also added basic tests to the bot to narrow down possible point of failure :

a. Go to the shell window in replit and type :

python3 main.py --quote

```
~/Twitter-Quotes-Bot$ python3 main.py --quote
####--> Current Version = v3
Often, foley artists will use well controlled forest fires to simulate the sound
of a plant growing, but speeded up.
```

If you receive an error here, this means something is wrong with the bot.json file parsing.

b. Go to the shell window in replit and type :

python3 main.py --tweet

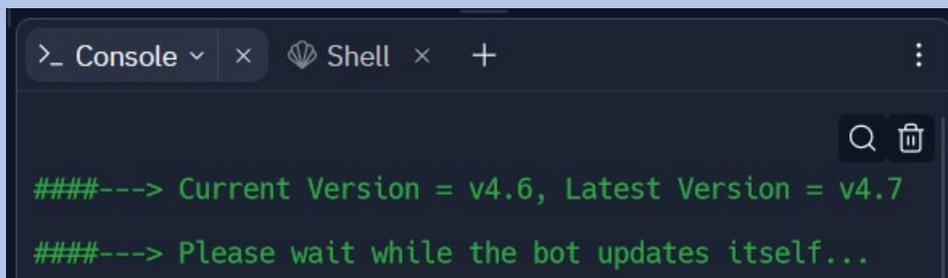
```
~/Twitter-Quotes-Bot$ python3 main.py --tweet
####--> Current Version = v3
####--> Obtained Credentials...

####--> Posted: ID=1684804480494149632, QUOTE=Often, foley artists will use a bag of
ball bearings to recreate the sound of zombies chewing on vital organs.
```

If you receive an error here, this means something is wrong with Twitter API. It could be an error on your side while entering the credentials or it could also be an error on Twitter's side like their servers being down.

Updating the Bot :

- There is a simple update checker built into the bot.
- Whenever the bot is run for the first time or restarted after going to sleep, it will automatically check for updates and download/apply them if possible

A screenshot of a terminal window with a dark background. The window title bar shows 'Console' and 'Shell'. The terminal output is in green text: '###--> Current Version = v4.6, Latest Version = v4.7' followed by '###--> Please wait while the bot updates itself...'. There are search and trash icons in the top right corner of the terminal area.

```
>_ Console x Shell x +  
###--> Current Version = v4.6, Latest Version = v4.7  
###--> Please wait while the bot updates itself...
```

A screenshot of a terminal window showing the completion of an update. The text is in green: '###--> UPDATE COMPLETE! Please Re-Run the bot to continue...'. Below that, in red text, it says 'repl process died unexpectedly:'. A small yellow cursor is visible at the end of the line.

```
###--> UPDATE COMPLETE! Please Re-Run the bot to continue...  
repl process died unexpectedly:  
^
```

- The above message means the update process is complete.
- Now you can just hit 'RUN' to restart the bot.

For any further questions/queries/ideas/suggestions, you can contact me directly on Twitter [@loonathedorm](https://twitter.com/loonathedorm)