Report

1. **Import Statements:**

   - `import json`: Imports the JSON module for working with JSON data.

   - `import time`: Imports the Time module for time-related functions.

   - `import requests`: Imports the Requests module for making HTTP requests.

   - `from ntscraper import Nitter`: Imports the Nitter class from the `ntscraper` module. This class is used for scraping Twitter data.

2. **Function Definition:**

   - `search_twitter(accounts, ticker, interval_minutes)`: Defines a function named `search_twitter` that takes three parameters:

      - `accounts`: A list of Twitter usernames to search for.

      - `ticker`: The keyword to search for in the tweets.

      - `interval_minutes`: The time interval between each scraping session in minutes.

3. **Main Loop:**

   - `while True`: Initiates an infinite loop that will continuously scrape Twitter data.

4. **Variables Initialization:**

   - `total_mentions = 0`: Initializes a variable to count the total number of mentions of the specified keyword.

   - `current_time = time.strftime("%H:%M:%S", time.localtime())`: Gets the current time in the format "hour:minute:second".

   - `all_tweets = []`: Initializes an empty list to store all scraped tweets.

5. **Scraping Loop:**

- `for account in accounts:`: Iterates over each Twitter account in the `accounts` list.

- `attempts = 3`: Initializes a variable to track the number of retry attempts for fetching tweets.

- `while attempts > 0:`: Initiates a loop to retry fetching tweets in case of errors.

- Inside the `while` loop:

  - Uses the `Nitter().get_tweets()` method to fetch tweets for the current account.

  - Extracts the tweets from the response and iterates over them.

  - Checks if each tweet contains the specified keyword (`ticker`) and updates the `total_mentions` count accordingly.

  - Appends each tweet to the `all_tweets` list.

6. **Output and Saving to JSON:**

- After scraping all accounts, it prints the total mentions of the keyword and the current time.

- Constructs the output file name based on the ticker symbol and the current date and time.

- Saves all scraped tweets to a JSON file with the constructed file name.

- If there is an error during saving, it prints an error message.

7. **Sleeping Between Scraping Sessions:**

- `time.sleep(interval_minutes * 60)`: Delays the execution of the script for the specified interval (in minutes) before starting the next scraping session.

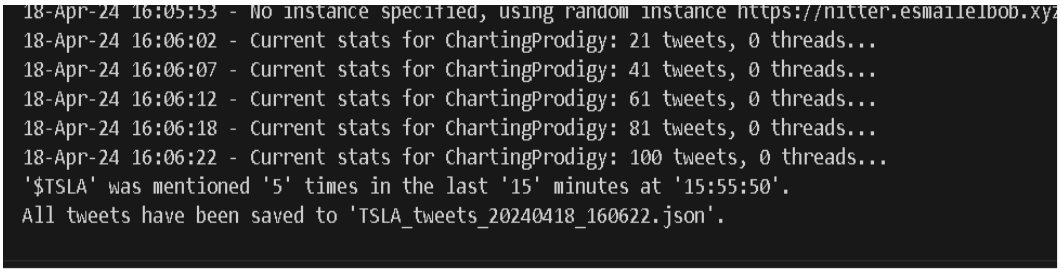
8. **Example Usage:**

- Provides an example usage of the `search_twitter` function with predefined inputs for `accounts`, `ticker`, and `interval_minutes`.

- Invokes the `search_twitter` function with the provided inputs.

This code essentially continuously scrapes Twitter data for the specified accounts, checks for mentions of a specific keyword, and saves the results to JSON files at regular intervals.

Output Example provided in code with interval time 15 min :



Another output I got with interval min 60 :

```
18-Apr-24 17:31:14 - Empty page on https://nitter.privacydev.net
'$TSLA' was mentioned '7' times in the last '60' minutes at '16:56:22'.
All tweets have been saved to 'TSLA_tweets_20240418_173116.json'.
```