

Guide to Keyword-based Weibo Post Extraction

AN Ji (ji.an@cnrs.fr)

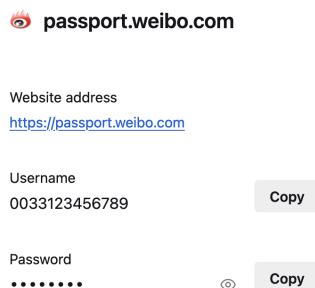
6 juillet 2024

Introduction

This is a guide to help you use my Python script to collect Weibo posts by keyword search. If you are interested in some research-related issues that need Weibo data, or if you'd like to get some ideas by glancing at how people react to some specific events or topics on Weibo, then this guide might help you save some manual data collection headaches. For advanced users, feel free to use my script to explore more features. Be there any question, comment or advice, don't hesitate to contact me via email. 🐱

1. What do we need?

- (1) Firefox browser
- (2) A valid Weibo account that has been logged in on Firefox. You can sign up with an email address or a phone number of your current location (In France, usernames will be in the format of 0033123456789). Save the login information to Firefox:



passport.weibo.com

Website address
<https://passport.weibo.com>

Username
0033123456789 Copy

Password
..... Copy

- (3) Path to your Firefox **Profile** folder
 - Firefox has a folder called **Profiles** for storing user informations
 - In this folder, there may be a few profile folders. What we need is to find the profile folder **that contains these two files: `key4.db` and `logins.json`**
 - For example, the path of my Firefox profile folder (on MacOS) is

```
/Users/ko/Library/Application Support/Firefox/Profiles/pejmtqsl.default-release-1714475519242
```

- For Windows users, it will be something like:

```
C:\Users\<your Windows login username>\AppData\Roaming\Mozilla\Firefox\Profiles\<your profile folder name>
```

- See more about how to find your Firefox profile here: [Profiles - Where Firefox stores your bookmarks, passwords and other user data](#) (you can modify the operation system at **Customize this article** on top-right)
- (4) Python environment, script and query: see below

2. Installation, query and path settings

- **Python:** go to Python's official website to [Download Python](#) on your PC. Follow the default steps to finish the installation.
- **Script:** go to my GitHub repository (<https://github.com/an-kei/SinaWeiboScraper>), click on the **Deep Green Button** `< > Code` on the top-right, click Download ZIP and store the .zip file anywhere on your PC. Decompress the zip.

File No. 1 - **WeiboScraper.py**

- This is the Python script to run later.

File No. 2 - **query.csv**.

- A CSV file is a text file. You can open it with any text editor (e.g. **TextEdit** or **BEdit** for MacOS, **Notepad** for Windows), or using a spreadsheet editor such as Microsoft Excel, Google Sheet, LibreOffice Calc, etc.

Folder "result": you can delete the whole folder because the script will create one if successfully run.

- **Query:** Modify the keywords and time range as you need. It's OK to put multiple keywords separated by a space in a single query. Make sure you have only one query per line. Then save the file. Remember to **always keep query.csv and WeiboScraper.py in the same folder**.

Remember to maintain the original format:

```
Keyword,YYYY-MM-DD,YYYY-MM-DD
Keyword1 Keyword2,YYYY-MM-DD,YYYY-MM-DD
Keyword1 Keyword2 Keyword3,YYYY-MM-DD,YYYY-MM-DD
```

Example:

```
光农互补,2024-06-01,2024-07-03
环境污染,2024-06-01,2024-07-03
征地 纠纷,2020-01-01,2020-02-01
非升即走 高校 青椒,2024-06-01,2024-07-03
```

- **Path:** Open the Python script with a text editor, go to **line 20**, replace the path with your own profile path found at (3) above. Always **keep your path in quotes** like I did below. Save the script.

```
19 query = "query.csv"
20 profile_path = "/Users/ko/Library/Application Support/Firefox/Profiles/pejmtqsl.default-release"
21 domain = "https://s.weibo.com"
22 output_dir = "./result"
```

3. Run the script in Terminal

- Once you finish setting your queries and updating your profile path in the script, you can open Terminal. See more at [Vue d'ensemble du Terminal Windows | Microsoft Learn](#) and [Guide d'utilisation de Terminal pour Mac - Assistance Apple \(FR\)](#).



- Go to the decompressed **SinaWeiboScraper** folder. For MacBook users, press **command + option + P** to show the folder path at the bottom. On the path, double-click at the folder name, choose "Open in Terminal".
- In the Terminal, simply type the following command and press Enter:

```
python3 WeiboScraper.py
```

- Now Patience! Wait until the program stops. If everything goes well, it will print some reminders in the Terminal, telling you the script is accessing the main search result page, how many pages it finds for your keyword(s), or it is extracting the posts.
- If the program didn't generate any csv file in the "result" folder (or didn't generate the desired result folder, if you deleted it previously), it means the program didn't work. In this case you might notice it prints some stuff saying:

```
Traceback (most recent call last):
File blablabla
File blablabla
... ..
OSError: [Errno55] No buffer space available
```

like this one:

```
~/Desktop/SinaWeiboScraper master > /Users/ko/Desktop/Jupyter/explore-weibo-
o-scraping
~/Desktop/Jupyter/explore-weibo-scraping > python3 WeiboScraper.py
Traceback (most recent call last):
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/
site-packages/urllib3/connectionpool.py", line 789, in urlopen
    response = self._make_request(
                ^^^^^^^^^^^^^^^^^^^^^
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/
site-packages/urllib3/connectionpool.py", line 495, in _make_request
    conn.request(
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/
site-packages/urllib3/connection.py", line 412, in request
    self.send(chunk)
  File "/Library/Frameworks/Python.framework/Versions/3.12/lib/python3.12/
http/client.py", line 1055, in send
    self.sock.sendall(data)
OSError: [Errno 55] No buffer space available

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/Users/ko/Desktop/Jupyter/explore-weibo-scraping/WeiboScraper.py",
line 247, in <module>
    WeiboKeywordSearch(query, profile_path)
```

- If you see the above feedback, press **Up**, then **Enter**, to rerun the script using the same command. Normally it will work this time, with some output printed on the screen as below:

```
~/Desktop/Jupyter/explore-weibo-scraping > python3 WeiboScraper.py
Accessing main result page: https://s.weibo.com/weibo?q=%E5%85%B9%E5%B6%9C%E4%B9%A2%E8%A1%A5&scope=ori&suball=1&timescope=custom%3A2024-05-20%3A2024-06-01 ...
Found 3 search result pages.
Processing page 1: https://s.weibo.com/weibo?q=%E5%85%B9%E5%B6%9C%E4%B9%A2%E8%A1%A5&scope=ori&suball=1&timescope=custom:2024-05-20:2024-06-01&page=1
...
Extracted 10 post(s) for search result page 1.
Processing page 2: https://s.weibo.com/weibo?q=%E5%85%B9%E5%B6%9C%E4%B9%A2%E8%A1%A5&scope=ori&suball=1&timescope=custom:2024-05-20:2024-06-01&page=2
...
Extracted 10 post(s) for search result page 2.
Processing page 3: https://s.weibo.com/weibo?q=%E5%85%B9%E5%B6%9C%E4%B9%A2%E8%A1%A5&scope=ori&suball=1&timescope=custom:2024-05-20:2024-06-01&page=3
...
Extracted 1 post(s) for search result page 3.
Processing finished. Successfully extracted 21 posts in total for keyword
光农互补 between 2024-05-20 and 2024-06-01
```

- Yay! Congratulations! You can now find the extracted posts in the “result” folder. Enjoy your discovery! 🐱

4. Tips about how to set a time range

- For each search, regardless of the time range you set, Weibo will return at most 50 result pages and each contains up to 10 posts (i.e. up to a maximum of 500 posts for a single search).
- So, in case you want to get as many posts as possible for a specific topic, you can set multiple queries using the same keyword(s) with a more narrow time range for each query.

- The optimal time range depends on how “popular” your topic is, so I would suggest that before running the script, you should firstly have a look at what your keyword(s) return as a result on the Weibo search page, especially how many pages it gives back for a test time range of, say, 10 days.
- **Example**: you notice that a 5-day time range already returns 50 result pages. Suppose you want to get as much data as possible for 1 month, you can then reduce the range to 3 days to see if it still gets 50 pages. If the total pages go down to, say, 30+ or 40+ pages, you can then select 3 days as your base time range for each search of the same keyword(s), and if necessary, enlarge or narrow again the time range accordingly as the “popularity” of your topic changes with events or time.