

[Open in app](#)[Sign up](#)[Sign In](#)[Search](#)

How to use Facebook Graph API and extract data using Python!



Ravi Ranjan · Follow

Published in [Towards Data Science](#)

4 min read · Oct 4, 2016

[Listen](#)[Share](#)

Hi all,

This is my second story on Medium.com. I have moved on from struggling to code to being a bit comfortable since the [first story](#). I wrote a Python code to extract publicly available data on Facebook. Let's dive into it.

Getting the Access Token:

To be able to extract data from Facebook using a python code you need to register as a developer on Facebook and then have an access token. Here are the steps for it.

1. Go to link developers.facebook.com, create an account there.
2. Go to link developers.facebook.com/tools/explorer.
3. Go to “My apps” drop down in the top right corner and select “add a new app”. Choose a display name and a category and then “Create App ID”.
4. Again get back to the same link developers.facebook.com/tools/explorer. You will see “Graph API Explorer” below “My Apps” in the top right corner. From “Graph API Explorer” drop down, select your app.
5. Then, select “Get Token”. From this drop down, select “Get User Access Token”. Select permissions from the menu that appears and then select “Get Access Token.”

6. Go to link <https://developers.facebook.com/tools/accesstoken>. Select “Debug” corresponding to “User Token”. Go to “Extend Token Access”. This will ensure that your token does not expire every two hours.

Python Code to Access Facebook Public Data:

Go to link <https://developers.facebook.com/docs/graph-api> if want to collect data on anything that is available publicly. See <https://developers.facebook.com/docs/graph-api/reference/v2.7/>. From this documentation, choose any field you want from which you want to extract data such as “groups” or “pages” etc. Go to examples of codes after having selected these and then select “facebook graph api” and you will get hints on how to extract information. This blog is primarily on getting events data.

First of all, import ‘urllib3’, ‘facebook’, ‘requests’ if they are already available. If not, download these libraries. Define a variable token and set its value to what you got above as “User Access Token”.

```
token= 'aiufniqaefncqiuhfencioaeusKJBNfljabicnlkjshniuwnscslkjnjndfi'
```

Getting list of Events:

Now to find information on events for any search term say “Poetry” and limiting those events’ number to 10000:

```
graph = facebook.GraphAPI(access_token=token, version = 2.7)
events = graph.request('/search?q=Poetry&type=event&limit=10000')
```

This will give a dictionary of all the events that have been created on Facebook and has string “Poetry” in its name. To get the list of events, do:

```
eventList = events['data']
```

Extracting all information for a event from the list of events extracted above:

Get the EventID of the first event in the list by

```
eventid = eventList[1]['id']
```

For this EventID, get all information and set few variables which will be used later by:

```
event1 = graph.get_object(id=eventid,
    fields='attending_count,can_guests_invite,category,cover,declined_count,description,end_time,guest_list_enabled,interested_count,is_canceled,is_page_owned,is_viewer_admin,maybe_count,noreply_count,owner,parent_group,place,ticket_uri,timezone,type,updated_time')
attenderscount = event1['attending_count']
declinerscount = event1['declined_count']
interestedcount = event1['interested_count']
maybecount = event1['maybe_count']
noreplycount = event1['noreply_count']
```

Getting the list of all those who are attending an event and converting the response into json format:

```
attenders =
requests.get("https://graph.facebook.com/v2.7/"+eventid+"/attending?
access_token="+token+"&limit="+str(attenderscount))

attenders_json = attenders.json()
```

Getting the admins of the event:

```
admins =
requests.get("https://graph.facebook.com/v2.7/"+eventid+"/admins?
access_token="+token)

admins_json = admins.json()
```

And similarly you can extract other information such as photos/videos/feed of that event if you want.

Go to <https://developers.facebook.com/docs/graph-api/reference/event/> and see “Edges” part in the documentation. See image

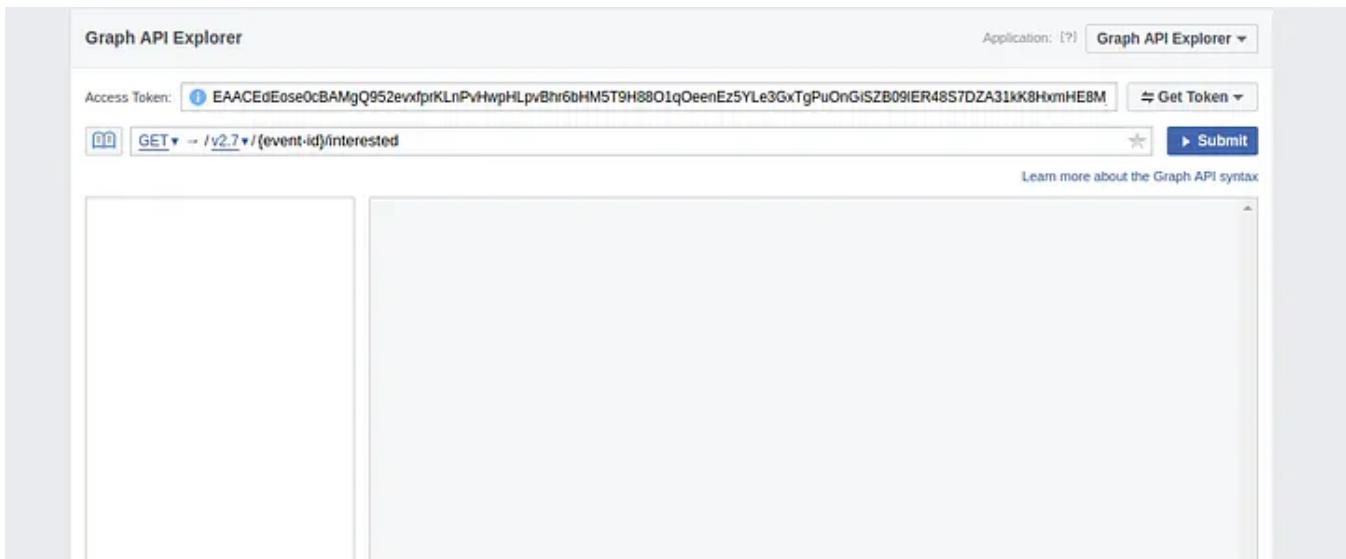
The screenshot shows the Facebook Graph API Reference for the 'Event' object. On the left, there's a sidebar with links like 'Analytics App Events Export', 'App Link Host', etc. The main content area has a table for the 'edges' field. The first row shows 'type' as an enum with values 'private', 'public', 'group', and 'community'. The second row shows 'updated_time' as a datetime field. Below this, there's a section titled 'Edges' with another table. It lists five edges: 'admins', 'attending', 'comments', 'declined', and 'interested'. Each edge is described as follows:

Edge	Description
<code>admins</code>	Users that are admins of the event
<code>attending</code>	Users attending the event
<code>comments</code>	Comments on the event
<code>declined</code>	Users who declined the event invitation
<code>interested</code>	Users interested in the event

Now, let's say, you want to have a list of all those who are interested in the event, click on ‘interested’ green word here. This will open up a new page:

The screenshot shows the details for the 'interested' edge. The sidebar on the left includes 'Event' under 'interested'. The main content area has a section titled 'Reading' which says 'Users interested in the event'. Below it is a code block showing the API endpoint: `GET /v2.7/{event-id}/interested HTTP/1.1` and 'Host: graph.facebook.com'. There are also sections for 'Permissions' (which states 'Permissions are not usually requested for this endpoint.') and 'Parameters'.

Select ‘Graph API Explorer’ here. This will open a new page:



Here, in place of {event-id}, put the id of the event, like this:



Hit submit. Also, on the same page, you will find below 'get code' option



Select it to see the code. Select 'curl' in the pop up that appears and then get the same output in the python code, write it with requests.get as it has been shown in above examples.

Hope this helps those who are beginning to work with facebook graph API. I will be happy to hear your suggestions/questions/feedback.

If you found this blog of any value to you and if you are into cryptocurrencies and if you are generous, consider sending some ripples to the following address:

Destination Tag: 5973413

Wallet Address: rLdinLq5CJood9wdjY9ZCdgyck8KGevkUj

There are too many ifs in the above statement. Send only if all the three are true!

Cryptocurrencies FTW!

Facebook

Python

Web Scraping

Data

Programming



Follow

Written by Ravi Ranjan

498 Followers · Writer for Towards Data Science

Read, Implement, Write, Repeat!!!

More from Ravi Ranjan and Towards Data Science



Ravi Ranjan

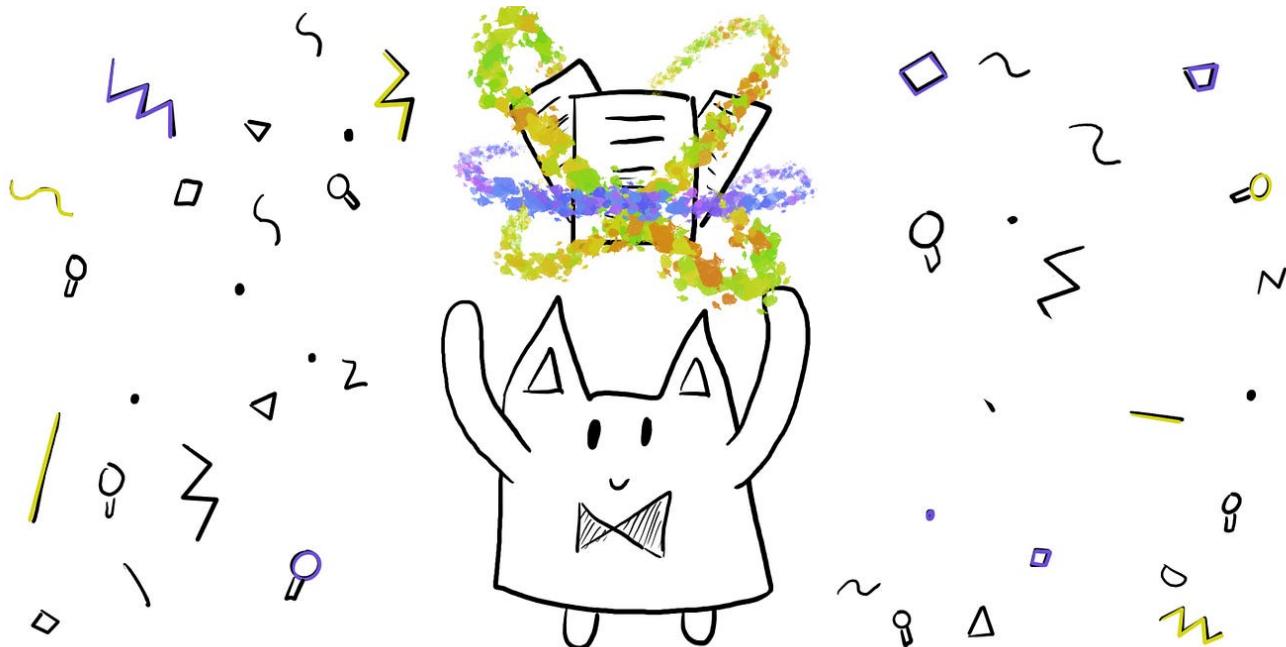
Marketing 101 in 5 minutes!

4P's, 5C's, STP! 😞Let's take a step back and begin!

6 min read · Sep 25, 2018

👏 228

💬 1



 Adrian H. Raudaschl in Towards Data Science

Forget RAG, the Future is RAG-Fusion

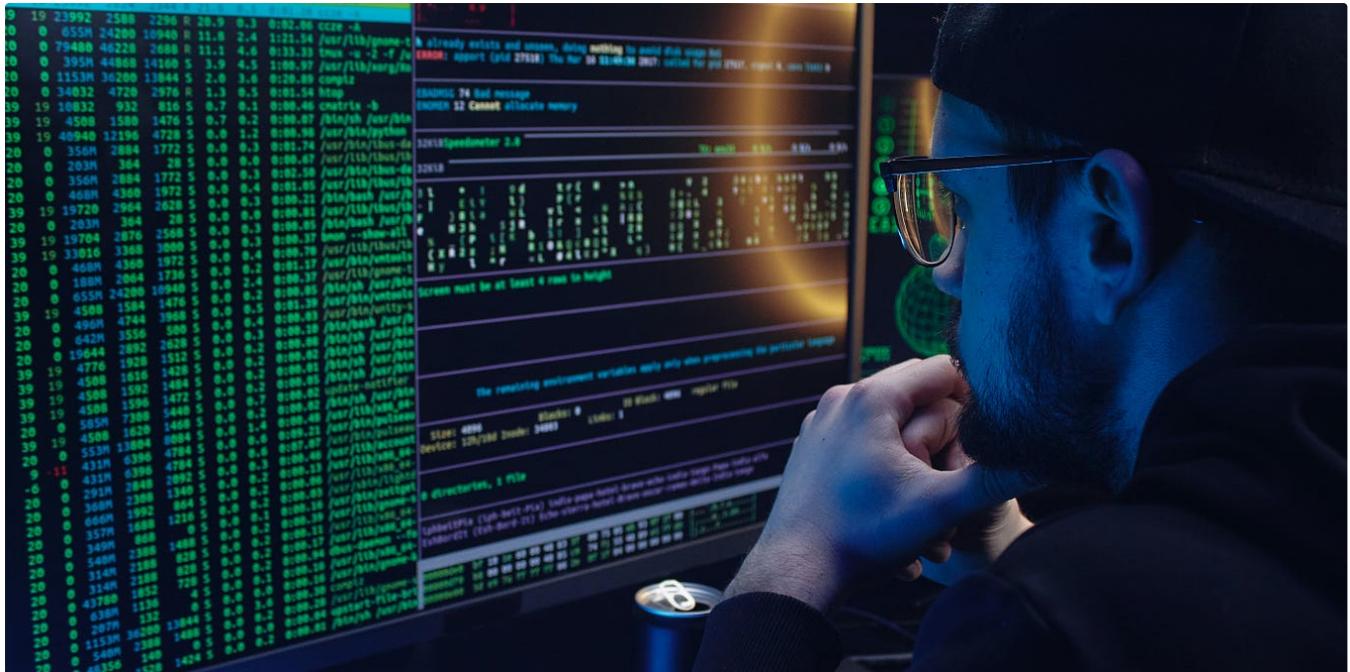
The Next Frontier of Search: Retrieval Augmented Generation meets Reciprocal Rank Fusion and Generated Queries

⭐ · 10 min read · Oct 6

👏 2.1K

💬 24





Natassha Selvaraj in Towards Data Science

Coding was Hard Until I Learned These 2 Things!

Here's what helped me go from "aspiring programmer" to actually landing a job in the field.

◆ · 7 min read · Oct 3

2.3K 29

+



Ravi Ranjan

Running a scala code as a spark-submit job using sbt

I had a very frustrating few weeks getting few stuff done. One of them was running a scala code through spark-submit.I almost hit my laptop...

2 min read · Nov 24, 2016

100

1



See all from Ravi Ranjan

See all from Towards Data Science

Recommended from Medium

```
> twscrape tweet_details 1674894268912087040 | jq
{
  "id": 1674894268912087000,
  "id_str": "1674894268912087040",
  "url": "https://twitter.com/elonmusk/status/1674894268912087040",
  "date": "2023-06-30 21:34:46+00:00",
  "user": {
    "id": 44196397,
    "id_str": "44196397",
    "url": "https://twitter.com/elonmusk",
    "username": "elonmusk",
    "displayname": "Elon Musk",
    "created": "2009-06-02 20:12:29+00:00",
    "followersCount": 145969587,
    // ...
    "_type": "sns scrape.modules.twitter.User"
  },
  // ...
}
```

 vladkens

How to still scrape millions of tweets in 2023 using twscrape

Twitter is a great place to gather data and assess various trends. Many analytics teams have used this source for their models.

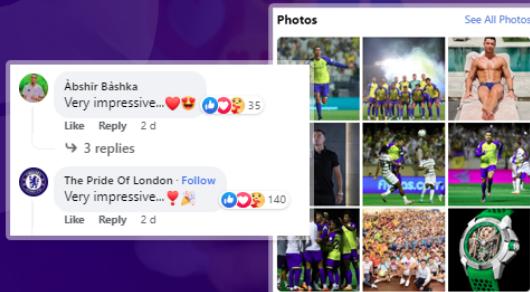
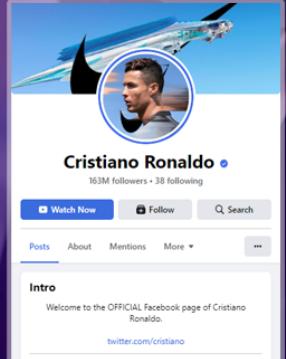
4 min read · Jul 2

21 4





2023 Guide: How to Scrape Social Media Data Using Python?

www.iwebdatascraping.com [facebook.com/iwebdatascraping](https://www.facebook.com/iwebdatascraping) info@iwebdatascraping.com

 iwebdatascraping

2023 Guide: How To Scrape Social Media Data Using Python?

Social Media Data scraping services, also popularly considered data scraping, collects data from social media platforms. It automatically...

5 min read · May 29

9



Lists



Coding & Development

11 stories · 248 saves



It's never too late or early to start something

15 stories · 189 saves



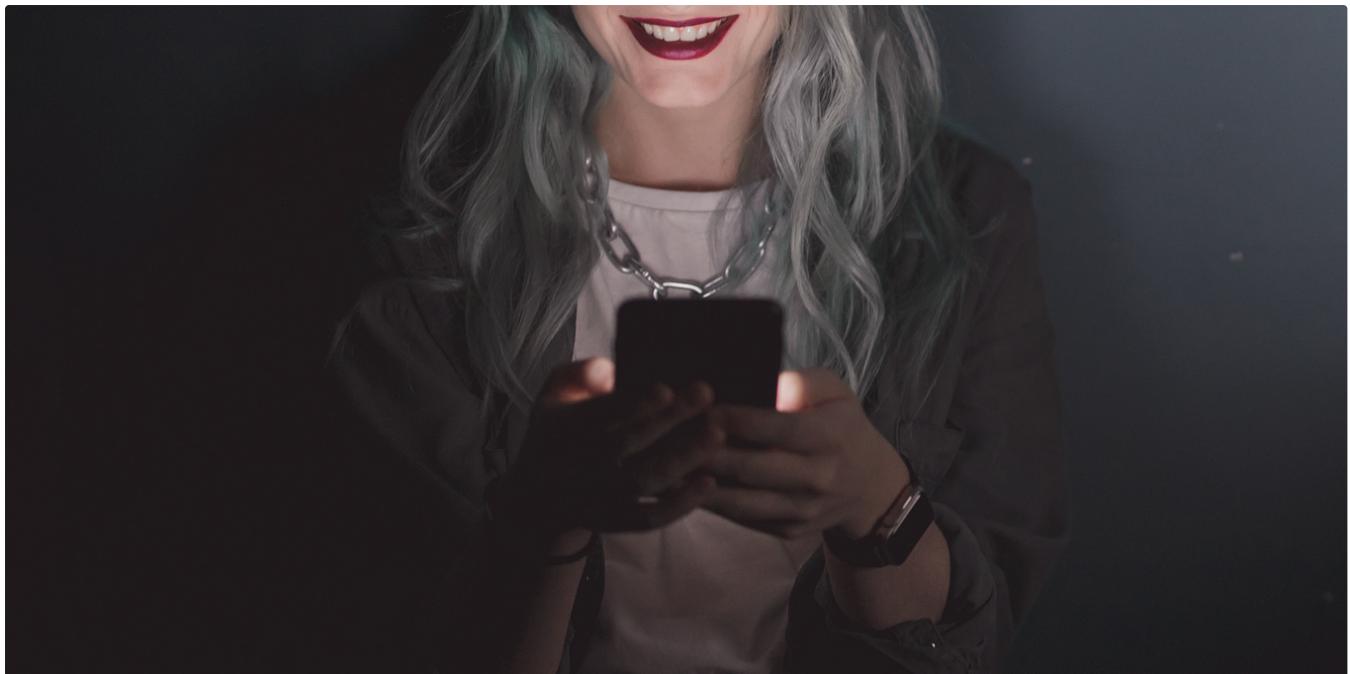
General Coding Knowledge

20 stories · 518 saves



New_Reading_List

174 stories · 171 saves



 Erland Sada

Cracking the Conversation

A Python Data Analysis Project: Extracting Valuable Insights from a WhatsApp Group Chat through EDA, Topic Modeling, and Sentiment Analysis

13 min read · May 4

 3  1



 Martin Beck in Better Programming

How to Scrape Tweets From Twitter

<https://towardsdatascience.com/how-to-use-facebook-graph-api-and-extract-data-using-python-1839e19d6999>

11/13

An up-to-date guide on scraping tweets from Twitter using Twitter's API

◆ · 7 min read · Aug 12

👏 64

🗨 1

↗+

```
get_page(self, url):
    options = Options()
    options.add_argument("--headless")
    service = Service(self.chrome_driver_path)
    driver = webdriver.Chrome(service=service, options=options)
    driver.get(url)
    page_source = driver.page_source
    driver.quit()
    return page_source
```



Evan Roberts

Scraping Dynamically Loaded Content with Selenium and BeautifulSoup

Table of Contents · Introduction · Setting up ChromeDriver and Chrome for Selenium ·
Windows Setup · Mac Setup · Linux / WSL Setup ...

17 min read · Jun 23

👏 30

🗨 1

↗+



Ebo Jackson in Level Up Coding

Building a Secure Dashboard Application with Flask

I recently built a secure web app in flask and python to serve as a corporate dashboard and I am happy to share

4 min read · Jun 29



See more recommendations