



How to scrape TikTok with Python

Published Date Read

April 22, 2025 13min



Table of Contents

Scraping **TikTok** (<https://www.tiktok.com/?lang=en>) data can unlock powerful insights, from tracking viral trends to analyzing influencer reach and engagement. Whether you're a marketer trying to identify the next big influencer, a brand looking to track trends, or a researcher analyzing user engagement, extracting data from TikTok can provide powerful insights.

In this blog, I'll show you how to scrape real-time insights from TikTok using Python.

What will you learn?

In this guide, we will use various tools to extract data from TikTok. But before we begin, let me just give you an overview of what you will learn in this article.

- Basic Python scraper setup with proper library installation.
- Scraping TikTok with a **Web Scraping API** (<https://www.scrapingdog.com/>).
- We'll be scraping key profile attributes such as follower count, following count, profile picture URL, and bio information.
- Store data in a CSV file.

If you're new to web scraping, I highly recommend reading **Web Scraping with Python** (<https://www.scrapingdog.com/blog/web-scraping-with-python/>) to gain valuable insights. This guide will help you build a solid foundation, making it easy to scrape data from virtually any website.

Requirements

I hope you have already installed Python on your computer; if not, then you can install it from [here](https://www.python.org/) (<https://www.python.org/>). Now, create a folder in which we will keep the project files.

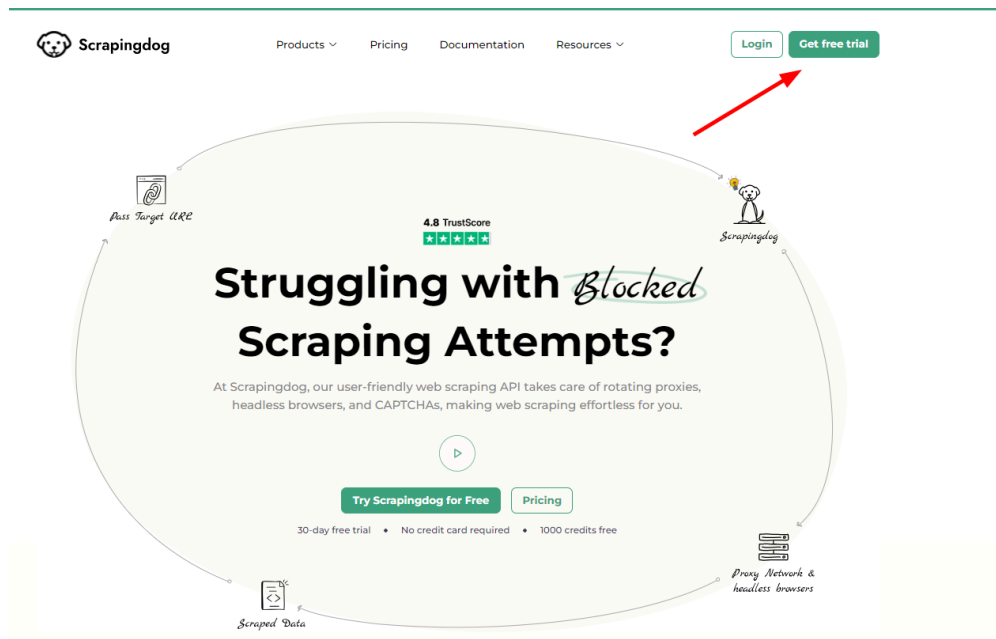
```
1 | mkdir tiktok
```

Install three libraries inside this folder.

- **requests** (<https://pypi.org/project/requests/>) for making an HTTP connection with the target website.
- **BeautifulSoup** (<https://pypi.org/project/beautifulsoup4/>) for parsing the raw data.
- **Pandas** (<https://pypi.org/project/pandas/>) for storing data in a CSV file.

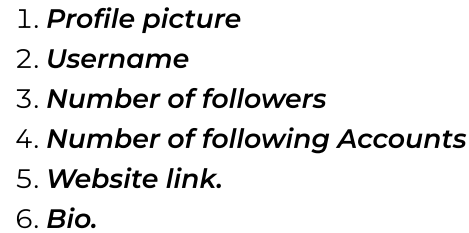
```
1 | pip install requests
2 | pip install beautifulsoup4
3 | pip install pandas
```

Now, **sign up** (<https://api.scrapingdog.com/register>) for the free trial pack. The trial pack includes **1000 credits**, which are enough to build a small scraper.



Finally, create a Python file in which we will code our TikTok scraper. I am naming this file as `social.py`.

Let's first decide what exactly we want to scrape from TikTok

[illegible]

The screenshot displays the Instagram profile of Kim Kardashian. The profile name is 'kimkardashian' with a verified badge. Below the name are buttons for 'Follow' and 'Message'. The profile statistics show '15 Following', '9.9M Followers', and '84.6M Likes'. The bio states 'No bio yet.' To the right, the browser's developer tools show the source code of the profile page. A red box highlights the user ID 'e1457k4r15' in the 'user-subtitle' attribute of a user profile element.

Follow Message

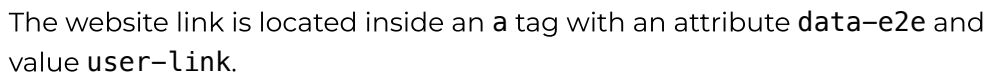
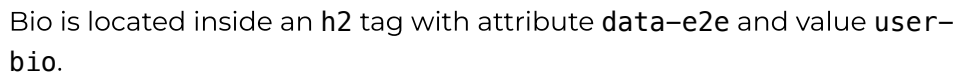
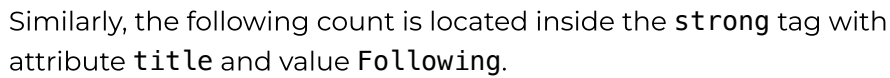
15 Following 9.9M Followers 84.6M Likes

No bio yet.

skims.social/kim_sale_tiktok

... `<div class="css-1voqj7-H3ContentInfo 14574r4" >...</div><div class="css-mgkx6-DivNumber 14574r1" >...</div><div class="css-mgkx6-DivNumber 14574r1" >...</div><strong title="Followers" data-c2e="followers-count"9.9M = 9.9M</div>Followers</div><div class="css-lou6alC-DivNumber 14574r1" >...</div></div>`

<https://www.scrapingdog.com/blog/scrape-tiktok-using-python/>



We have the location of each element. Now, we can code and extract this data.

Before we start coding, I recommend reading the [documentation \(https://docs.scrapingdog.com/web-scraping-api\)](https://docs.scrapingdog.com/web-scraping-api) of Scrapingdog.

```
1 import requests
2 from bs4 import BeautifulSoup
3
4
5 params={
6     'api_key': 'your-api-key',
7     'url': 'https://www.tiktok.com/@kimkardashian?lang=en',
8     'dynamic': 'true',
9     'wait': '3000',
10 }
11
12 response = requests.get("https://api.scrapingdog.com/scrape",
13 params=params)
14
15
16 print("status code is ",response.status_code)
17 print(response.text)
```

The code is straightforward, but let me explain you step by step.

- First, we have imported the libraries that we installed earlier.
- Then, we created a `params` object containing the necessary parameters required to make a request to the Scrapingdog Web Scrapping API.
- Using `requests`, we made a GET request to the API.
- Finally, we are printing the status of the request and the downloaded data.

Once we run this code, you will get this on your console.

```
status code is 200
<!DOCTYPE html><html lang="en" dir="ltr" data-tux-color-scheme="dark"><head><meta charset="UTF-8"><meta name="viewport"
content="width=device-width,initial-scale=1"><title>TikTok - Make Your Day</title><link rel="preload" as="script" href="
https://lf16-tiktok-web.tiktokcdn-us.com/obj/tiktok-web-tx/webmssdk/2.0.0.412/webmssdk.js"><script src="https://lf16-tik
tok-web.tiktokcdn-us.com/obj/tiktok-web-tx/tiktok_privacy_protection_framework/loader/2.0.0.243/index.js" crossorigin="a
nonymous" async="" data-business="serverless.tiktok.desktop" data-sw="/sw.js" data-env="production" data-region="us-tp"
data-sw_version="1.0.0.89,2.0.0.4"></script><script type="application/json" id="script-manager">{"imports":{"acrawler":
"https://lf16-tiktok-web.tiktokcdn-us.com/obj/tiktok-web-tx/webmssdk/2.0.0.412/webmssdk.js","secsdk_sg":"https://sf16-we
bsite-login.neutral.ttwstatic.com/obj/tiktok_web_login_static/secdsk/secdsk-lastest.umd.js","secsdk_va":"https://sf16-we
bsite-login.neutral.ttwstatic.com/obj/tiktok_web_login_static/secdsk/secdsk-lastest.umd.js","secsdk_ttp":"https://lf16-c
dn-tos.tiktokcdn-us.com/obj/static-tx/secdsk/secdsk-lastest.umd.js"}}</script><meta name="pumbaa-ctx" content="login=0,f
tc=0"><link rel="preload" href="https://lf16-tiktok-web.tiktokcdn-us.com/obj/tiktok-web-tx/tiktok/webapp/main/react-v17/
```

As you can see, we got a **200** status code. Now, we can proceed ahead and parse the data using BS4.

Parsing data from TikTok using BeautifulSoup

```
1 soup = BeautifulSoup(response.text, 'html.parser')
2
3 try:
4     obj["username"]=soup.find("h1").text
5 except:
6     obj["username"]=None
7
8 try:
9     obj["profile"]=soup.find("img",{"class":"css-1zpj2q-
10 ImgAvatar"}).get('src')
11 except:
12     obj["profile"]=None
13
14
15 try:
16     obj["following"]=soup.find("strong",{"title":"'Following'"}).text
17 except:
18     obj["following"]=None
19
20 try:
21     obj["followers"]=soup.find("strong",{"title":"'Followers'"}).text
22 except:
23     obj["followers"]=None
24
25 try:
26     obj["Bio"]=soup.find("h2",{"data-e2e":"'user-bio'"}).text
27 except:
28     obj["Bio"]=None
29
30 try:
31     obj["website"]=soup.find("a",{"data-e2e":"'user-
32 link'"}).get('href')
33 except:
34     obj["website"]=None
35
36 l.append(obj)
```

Here using the `find()` function of **BeautifulSoup** I have extracted the text value of each element. Once you run the code you should see this data on your console.

```
1 | [{ 'username': 'kimkardashian',
2 |   'profile': 'https://p19-pu-sign-useast8.tiktokcdn-us.com/tos-useast5-
3 |   avt-0068-tx/7310049872432857130~tplv-tiktokx-
4 |   cropcenter:1080:1080.jpeg?dr=9640&refresh_token=bad80b71&x-
5 |   expires=1745485200&x-
6 |   signature=C2YQZ3vduWrLxGgZ%2BaG%2FkwvZ304%3D&t=4d5b0474&ps=13740610&shp=
7 |   'following': '15',
   |   'followers': '9.9M',
   |   'Bio': 'No bio yet.',
   |   'website': 'https://www.tiktok.com/link/v2?
   |   aid=1988&lang=en&scene=bio_url&target=skims.social%2Fkim_sale_tiktok'
   | }]
```

As you can see, we are successfully able to scrape & parse all the data from TikTok with the help of Scrapingdog and Python.

Saving data to a CSV file

For this part, we will need the help of the **pandas** library.

```
1 | df = pd.DataFrame(l)
2 | df.to_csv('tiktok.csv', index=False, encoding='utf-8')
```

After running the code, you will get a **tiktok.csv** file inside your folder.

	A	B	C	D	E	F
1	username	profile	following	followers	Bio	website
2	kimkardashian	https://p19-pu-sign-useast8.tiktokcdn-us.com/tos-useast5-avt-0068-tx/7310049872432857130~tplv-tiktokx-cropcenter:1080:1080.jpeg?dr=9640&refresh_token=bad80b71&x-expires=1745485200&x-signature=C2YQZ3vduWrLxGgZ%2BaG%2FkwvZ304%3D&t=4d5b0474&ps=13740610&shp=	15	9.9M	No bio yet.	https://www.tiktok.com/link/v2?aid=1988&lang=en&scene=bio_url&target=skims.social%2Fkim_sale_tiktok
3						
4						
5						

Complete Code

We can scrape video stats too from TikTok for any particular video, I will leave this exercise to you. But for the current scenario, the code will look like this.

```

1  import requests
2  from bs4 import BeautifulSoup
3  import pandas as pd
4
5  l=[]
6  obj={}
7
8  params={
9      'api_key': 'your-api-key',
10     'url': 'https://www.tiktok.com/@kimkardashian?lang=en',
11     'dynamic': 'true',
12     'wait': '10000',
13     }
14
15     response = requests.get("https://api.scrapingdog.com/scrape",
16                             params=params)
17
18
19     print("status code is ",response.status_code)
20
21     soup = BeautifulSoup(response.text, 'html.parser')
22
23     try:

```

Recent Blogs

Scrapingbee vs ScraperAPI vs Scrapingdog: Which One Should You Choose



```

39
40     try:
41         obj["followers"]=soup.find("strong",{"title":'Followers'}).text
42     except:
43         obj["followers"]=None
44
45     try:
46         obj["Bio"]=soup.find("h2",{"data-e2e":'user-bio'}).text
47     except:
48         obj["Bio"]=None
49
50     try:
51         obj["website"]=soup.find("a",{"data-e2e":'user-
52         link'}).get('href')
53     except:
54         obj["website"]=None
55
56     l.append(obj)
57     print(l)
58     df = pd.DataFrame(l)
59     df.to_csv('tiktok.csv', index=False, encoding='utf-8')

```

Scrapingbee vs ScraperAPI vs Scrapingdog: Which One Should You Choose

In this comparison article, we will compare the top three web scraping platforms and help you choose the best one for your needs.

2025-06-09

(<https://www.scrapingdog.com/blog/scrape-tiktok-using-python/>)

Building AI Agent

In this article, we will explore the possibilities of building an AI agent using various tools and frameworks.

Building a Simple n8n AI Agent using Scrapingdog Google Search API

2025-05-26

[\(https://www.scrapingdog.com/blog/scrape-tiktok-using-python/\)](https://www.scrapingdog.com/blog/scrape-tiktok-using-python/)

Scraping API to extract TikTok data efficiently without hitting constant roadblocks. Whether you're tracking influencers, analyzing engagement, or pulling video metadata, this approach ensures reliability and scalability.

Additional Resources

- [Scrape YouTube Search Results using Python](https://www.scrapingdog.com/blog/scrape-youtube-search/)
(<https://www.scrapingdog.com/blog/scrape-youtube-search/>)
- [Scrape X \(Twitter\) Data using Python](https://www.scrapingdog.com/blog/scrape-twitter/)
(<https://www.scrapingdog.com/blog/scrape-twitter/>)
- [Scrape Instagram Using Python](https://www.scrapingdog.com/blog/scrape-instagram/)
(<https://www.scrapingdog.com/blog/scrape-instagram/>)
- [Scrape LinkedIn Profiles using Python](https://www.scrapingdog.com/blog/scrape-linkedin-profiles-using-python/)
(<https://www.scrapingdog.com/blog/scrape-linkedin-profiles-using-python/>)



Manthan Koolwal

My name is Manthan Koolwal and I am the founder of scrapingdog.com. I love creating scraper and seamless data pipelines.



Try Scrapingdog for Free!

Get 1000 free credits to spin the API. No credit card required!

Start your Free Trial
(<https://api.scrapingdog.com/register>)



Company

About

Documentation

Blog

Contact

Affiliate Program

Terms of Service

Privacy Policy

Data Processing Agreement

GDPR Compliance

SLA

● Status



Product



Scrapingdog vs Competitors



Learn Web Scraping

(h
tt
p
s:
//
w
w
w.
li
nk
e
di
n.
c
o
m
/s
c
r
a
p
i
n
g
d
o
g)
(h
tt
p
s:
//
w
m
e
di
u
m
.c
o
m
/@
d
a
r
s
h
a
n
k
h
a
n
d
e
l
w
a
l
1
2)
(h
tt
p
s:
//
w
o
u
t
u
b
e
c
o
m
/@
s
c
r
a
p
i
n
g
d
o
g
9
9
5
5)

© 2020-2025 Scrapingdog. All rights reserved.