Scrapingdog (https://www.scrapingdog.com)

How to scrape TikTok with Python

Published Date	Read
April 22, 2025	13min

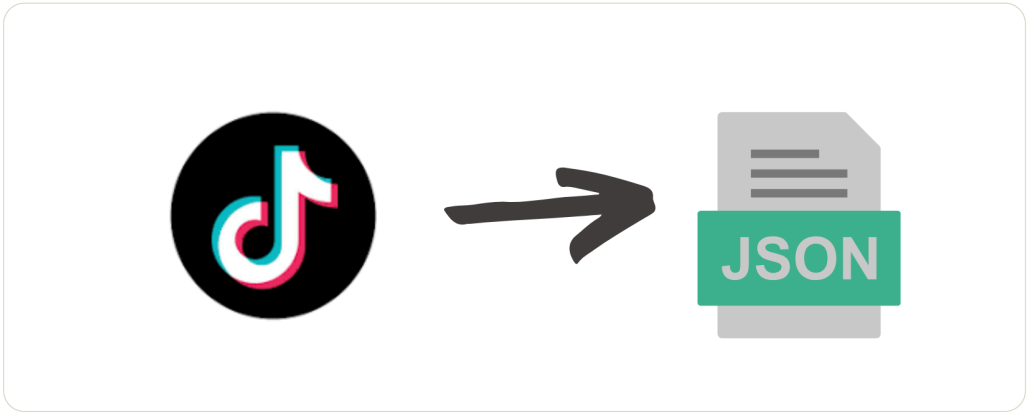


Table of Contents

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/)**. Now, create a folder in which we will keep the project files.

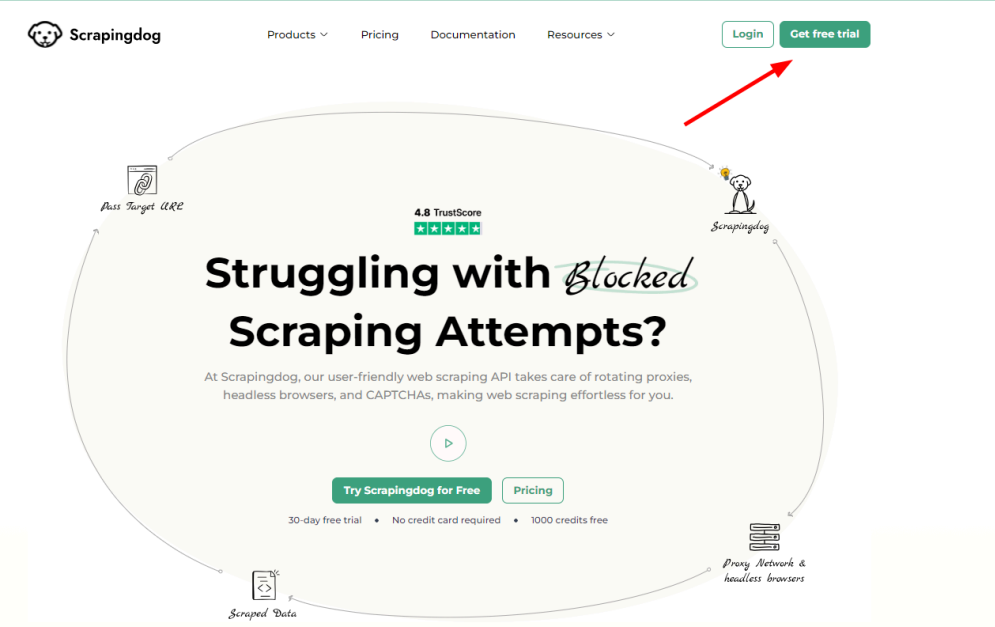
```
mkdir tiktok
```

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.

```
pip install requests
pip install beautifulsoup4
pip install pandas
```

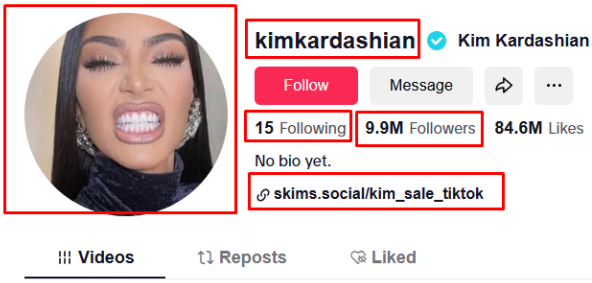
Now, **sign up (https://api.scrapingdog.com/register)** for the free trial pack. The trial pack includes **1000 credits**, which are enough to build a small scraper.



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

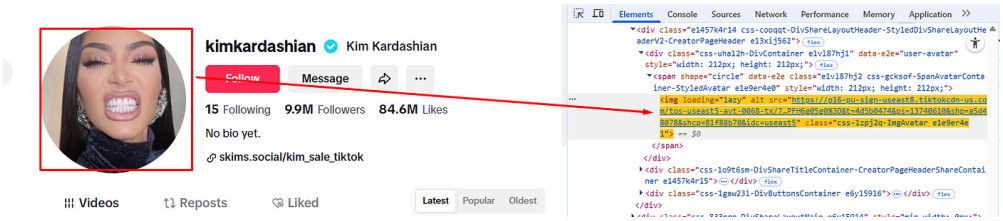
Scraping TikTok with Python

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

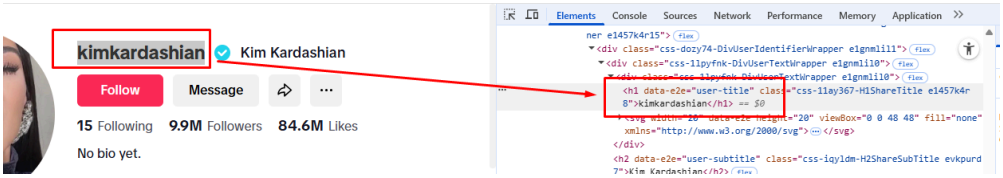


1. ***Profile picture***
2. ***Username***
3. ***Number of followers***
4. ***Number of following Accounts***
5. ***Website link.***
6. ***Bio.***

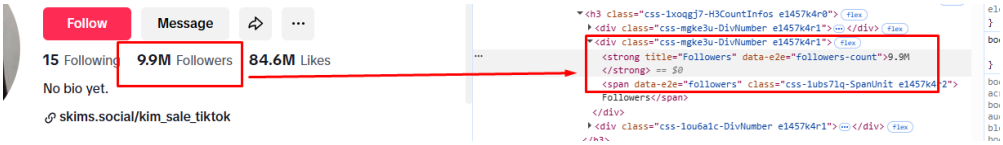
Analyzing the HTML structure of the page



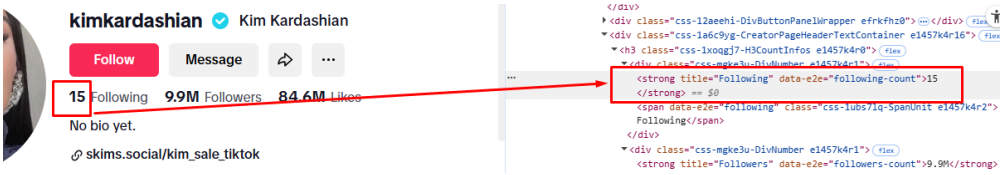
Profile picture is located inside the `img` tag with `class` *css-1zpj2q-ImgAvatar*.



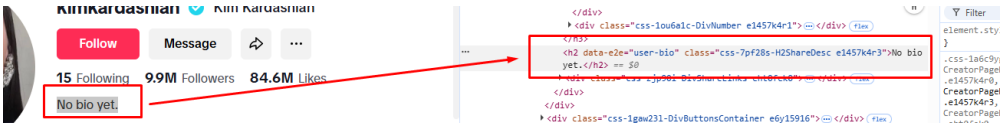
The username is located inside the `h1` tag.



The number of followers is located inside the `strong` tag with the attribute `title` and value `Followers`.



Similarly, the following count is located inside the `strong` tag with attribute `title` and value `Following`.



Bio is located inside an `h2` tag with attribute `data-e2e` and value `user-bio`.

The website link is located inside an `a` tag with an attribute `data-e2e` and value `user-link`.

Profile	img tag with class css-1zpj2q-ImgAvatar.
Username	h1 tag
Followers	strong tag with the attribute title and value Followers
Following	strong tag with attribute title and value Following
Bio	h2 tag with attribute data-e2e and value user-bio
Website	a tag with an attribute data-e2e and value user-link

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

Download data from TikTok

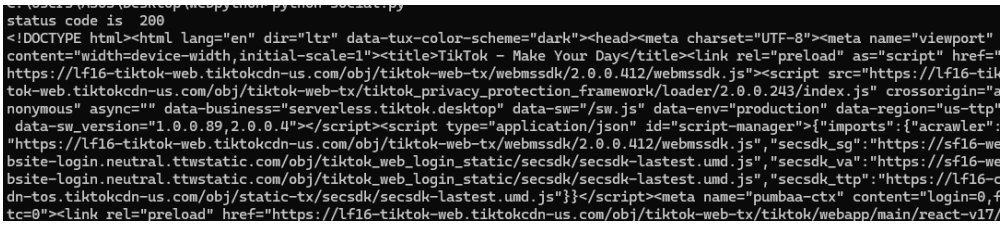
Before we start coding, I recommend reading the **documentation (https://docs.scrapingdog.com/web-scraping-api)** of Scrapingdog.

```
import requests
from bs4 import BeautifulSoup


params={
  'api_key': 'your-api-key',
  'url': 'https://www.tiktok.com/@kimkardashian?lang=en',
  'dynamic': 'true',
  'wait': '3000',
  }

response = requests.get("https://api.scrapingdog.com/scrape",
params=params)


print("status code is ",response.status_code)
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 Scraping 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.



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

```
soup = BeautifulSoup(response.text, 'html.parser')

try:
    obj["username"]=soup.find("h1").text
except:
    obj["username"]=None

try:
    obj["profile"]=soup.find("img",{"class":"css-1zpj2q-
ImgAvatar"}).get('src')
except:
    obj["profile"]=None


try:
    obj["following"]=soup.find("strong",{"title":'Following'}).text
except:
    obj["following"]=None

try:
    obj["followers"]=soup.find("strong",{"title":'Followers'}).text
except:
    obj["followers"]=None

try:
    obj["Bio"]=soup.find("h2",{"data-e2e":'user-bio'}).text
except:
    obj["Bio"]=None

try:
    obj["website"]=soup.find("a",{"data-e2e":"user-
link"}).get('href')
except:
    obj["website"]=None

l.append(obj)
```

```
print(l)
```

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.

```
[{'username': 'kimkardashian',
'profile': '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%2FkwvZ3O4%3D&t=4d5b0474&ps=13740610&shp=
'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'
}]
```

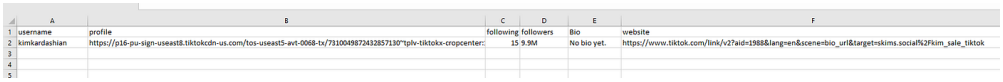
As you can see, we are successfully able to scrape & parse all the data from TikTok with the help of Scrapingdog and Python.

Saving data to a CSV file

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

```
df = pd.DataFrame(l)
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://p16-pu-sign-useast8.tiktokcdn-us.com/tos-useast5-avt-0068-tx/7310049872432857130~tplv-tiktokx-cropcenter:	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.

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

l=[]
obj={}

params={
  'api_key': 'your-api-key',
  'url': 'https://www.tiktok.com/@kimkardashian?lang=en',
  'dynamic': 'true',
  'wait': '10000',
  }

response = requests.get("https://api.scrapingdog.com/scrape",
params=params)


print("status code is ",response.status_code)

soup = BeautifulSoup(response.text, 'html.parser')

try:
```

Recent Blogs


```

try:
    obj["followers"]=soup.find("strong",{"title":'Followers'}).text
except:
    obj["followers"]=None

try:
    obj["Bio"]=soup.find("h2",{"data-e2e":'user-bio'}).text
except:
    obj["Bio"]=None

try:
    obj["website"]=soup.find("a",{"data-e2e":"user-
link"}).get('href')
except:
    obj["website"]=None

l.append(obj)
print(l)
df = pd.DataFrame(l)
df.to_csv('tiktok.csv', index=False, encoding='utf-8')
```

Scrapingbee vs Sc
Needs

In this comparison a
them on top platfor

2025-06-09

(https://www.scra

Building AI Agent

In this article, we wi

Building a Simple n8n AI Agent using Scrapingdog Google Search API

2025-05-26

(https://www.s

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/)
- Scrape X (Twitter) Data using Python (https://www.scrapingdog.com/blog/scrape-twitter/)
- Scrape Instagram Using Python (https://www.scrapingdog.com/blog/scrape-instagram/)
- 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.