Data Collection

CSC 380 - Principles of Data Science

Lecture 3.3

In the last lecture,

- Research Design for Statistical Analysis
  - Causation versus Correlation
  - Sampling
- Revisit the Data Science Process
- Data Collection
  - Till SQL
- Data Processing

In the this lecture,

- Research Design for Statistical Analysis
  - Causation versus Correlation
  - Sampling
- Revisit the Data Science Process
- Data Collection
  - Till SQL
  - API
  - Scraping
- Data Processing

How to collect data?

- Direct download and load from local storage
- Generate locally via downloaded code (e.g., simulation)
- Query data from a database.
- Query an API from the intra/internet
- Scrape data from a webpage

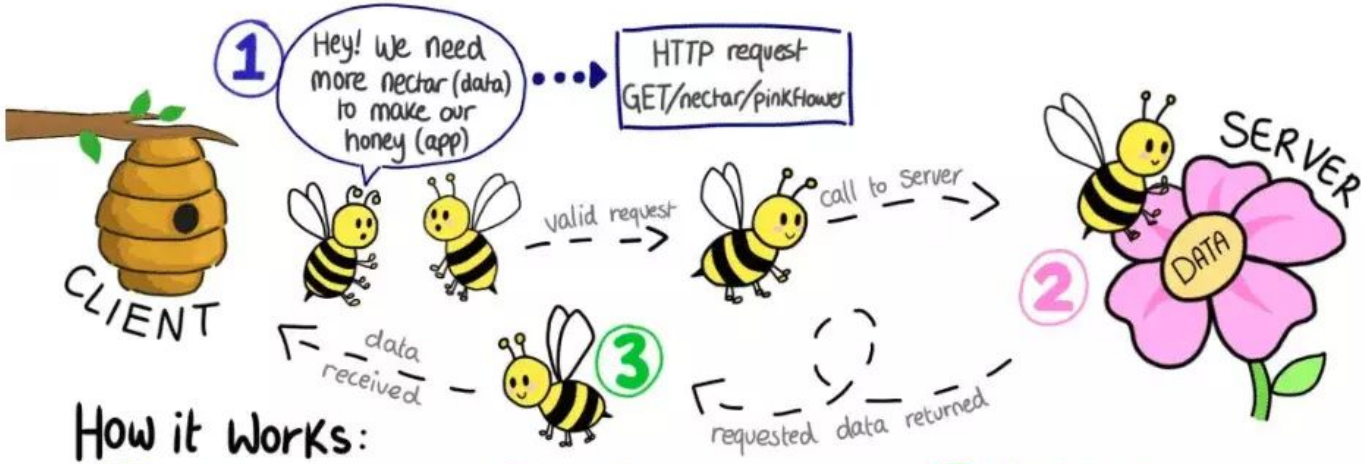
API - Application Programming Interface

Mechanisms that enable two software components to communicate with each other using a set of definitions and protocols.

Credit: https://aws.amazon.com/what-is/api/

What is an API?

An application programming interface allows two programs to communicate. On the web, APIs sit between an application and a web server, and facilitate the transfer of data.



How it Works:

1 Request
API call is initiated by by the client application via a HTTP request

2 Receive
Our worker bee acts as an API, going to a flower (server) to collect nectar (data)

3 Response
The API transfers the requested data back to the requesting application, usually in JSON format

Credit: https://rapidapi.com/blog/api-glossary/api/

tiktok api documentation

Videos | News | Images | Books | Maps | Shopping | Flights | Finance

About 7,420,000 results (0.33 seconds)

tiktok.com
https://developers.tiktok.com › doc › overview

Overview to TikTok for Developer Documentation

Here we'll show you how to set up your **TikTok Developer** account and start integrating your app with our development kits and server APIs. Onboard as a developer.

Display API · Register your app · Share Video API · Scopes

tiktok.com
https://developers.tiktok.com

TikTok for Developers | TikTok

Our **API** empowers your creators to seamlessly post their content to **TikTok**. Learn more. Embed videos. Make it easy to share your favorite **TikTok** videos ...

Products · Get Started · Research API · Login Kit for Web

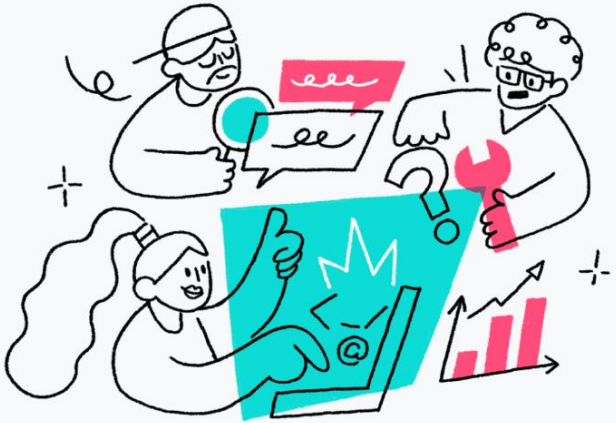
TikTok for developers

Products Docs Support Log in

Build tools for creators, researchers, and communities

TikTok for developers

Get Started



https://developers.tiktok.com/

Overview to TikTok for Developer Documentation

Welcome to the TikTok for Developers documentation. Here we'll show you how to set up your TikTok Developer account and start integrating your app with our development kits and server APIs.

Onboard as a developer

To start integrating with us, create your developer account and register your apps. Learn more.

Tip: Subscribe to our email notifications to get our latest product updates.

Browse our development kits

Source: https://developers.

- Overview
- App Management
- Our Guidelines
- Integration Essentials
- Login Kit
- Share Kit
- Content Posting API
- Display API
  - Overview
  - Get Started
  - API Reference

Source:
https://developers.tiktok.com/

Overview

The Display API contains a set of HTTP-based APIs that your product can use to display a TikTok creator's videos and their profile information.

Your platform's influencers can display their TikTok profile identities and videos to enrich content, attract more audiences, and enable their followers to view their TikTok videos without leaving your platform.

Source:
https://developers.tiktok.com/

Components

Display API has three major APIs: `/v2/user/info/`, `/v2/video/list/`, and `/v2/video/query/`.

/v2/user/info/: Get a TikTok user's basic profile information. This includes the user's open_id, avatar_url, display_name, profile_deep_link, and bio_description.

/v2/video/list/: Get the metadata of a TikTok user's recently uploaded videos.

/v2/video/query/: Get the metadata of TikTok a user's videos filtered by video Id.

Source:
https://developers.tiktok.com/

Overview

The `/v2/user/info/` endpoint returns some basic information for a given TikTok user.

HTTP URL	https://open.tiktokapis.com/v2/user/info/
HTTP Method	GET
Scope	Needs relevant scopes to be authorized by the TikTok user through the authorization flow.

Source:
https://developers.tiktok.com/

Request

Source:
https://developers.tiktok.com/

Headers

Key	Type	Description	Example	Required
Authorization	string	The token that bears the authorization of the TikTok user, which is obtained through /oauth/access_token/.	Bearer act.1d1021d2aee3d41fee2d2add43456badMFZnrhFhfWotu3Ecuiuka27L56lr!2323	true

Query Parameters

Key	Type	Description	Example	Required
fields	string	The set of user fields to request for	open_id,union_id,avatar_url	true

Example

```
curl -L -X GET 'https://open.tiktokapis.com/v2/user/info/?fields=open_id,union_id,avatar_url' \
-H 'Authorization: Bearer act.1d1021d2aee3d41fee2d2add43456badMFZnrhFhfWotu3Ecuiuka27L56lr!2323'
```

Response

Key	Type
data	map<string, User Object>
error	Error Object

Source:
https://developers.tiktok.com/

User Object

Source: https://developers.tiktok.com/

Field	Type	Description	Authorized Scope
open_id	string	The unique identification of the user in the current application.Open id for the client	user.info.basic
union_id	string	The unique identification of the user across different apps for the same developer. For example, if a partner has X number of clients, it will get X number of open_id for the same TikTok user, but one persistent union_id for the particular user	user.info.basic
avatar_url	string	User's profile image	user.info.basic
avatar_url_100	string	User`s profile image in 100x100 size	user.info.basic
avatar_large_url	string	User's profile image with higher resolution	user.info.basic
display_name	string	User's profile name	user.info.basic

Source: https://developers.tiktok.com/

bio_description	string	User's bio description if there is a valid one	user.info.profile
profile_deep_link	string	The link to user's TikTok profile page	user.info.profile
is_verified	boolean	Whether TikTok has provided a verified badge to the account after confirming that it belongs to the user it represents	user.info.profile
follower_count	int64	User's followers count	user.info.stats
following_count	int64	The number of accounts that the user is following	user.info.stats
likes_count	int64	The total number of likes received by the user across all of their videos	user.info.stats
video_count	int64	The total number of publicly posted videos by the user	user.info.stats

Example


```
{
   "data":{
      "user":{
         "avatar_url":"https://p19-sign.tiktokcdn-us.com/tos-avt-0068-
tx/b17f0e4b3a4f4a50993cf72cda8b88b8~c5_168x168.jpeg",
         "open_id":"723f24d7-e717-40f8-a2b6-cb8464cd23b4",
         "union_id":"c9c60f44-a68e-4f5d-84dd-ce22faeb0ba1"
      }
   },
   "error":{
      "code":"ok",
      "message":"",
      "log_id":"20220829194722CBE87ED59D524E727021"
   }
}
```


Source:
https://developers.tiktok.com/

```
>>> import requests
>>> r = requests.get('https://httpbin.org/basic-auth/user/pass', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"authenticated": true, ...'
>>> r.json()
{'authenticated': True, ...}
```

Source: https://pypi.org/project/requests/

How to collect data?

- Direct download and load from local storage
- Generate locally via downloaded code (e.g., simulation)
- Query data from a database.
- Query an API from the intra/internet
- Scrape data from a webpage

How to collect data?

- Direct download and load from local storage
- Generate locally via downloaded code (e.g., simulation)
- Query data from a database.
- Query an API from the intra/internet
- Scrape data from a webpage

Robots.txt

- Filename used for implementing the Robots Exclusion Protocol,
- Indicate to visiting web crawlers and other web robots which portions of the website they are allowed to visit.
- Voluntary compliance.

2.1 File Types

General structured data:

- Comma-Separated Value (CSV) files & strings
- Javascript Object Notation (JSON) files & strings
- HTML, XHTML, XML files & strings

Domain-specific structured data:

- Shapefiles: geospatial vector data (OpenStreetMap)
- RVT files: architectural planning (Autodesk Revit)
- You can make up your own! Always document it.

2.1 File Types

General structured data:

- Comma-Separated Value (CSV) files & strings
- Javascript Object Notation (JSON) files & strings
- HTML, XHTML, XML files & strings

Domain-specific structured data:

- Shapefiles: geospatial vector data (OpenStreetMap)
- RVT files: architectural planning (Autodesk Revit)
- You can make up your own! Always document it.

2.1.1 CSV - Comma Separated Value

Plain text file that stores data by delimiting data entries with commas.

CSV - Comma Separated Value

Plain text file that stores data by delimiting data entries with commas.

Week No, Lecture No, Topic, Objectives, Duration

3, 3, Data Collection and Data Processing, Learn about abc, 90mins

CSV - Comma Separated Value

Plain text file that stores data by delimiting data entries with commas.

Week No, Lecture No, Topic, Objectives, Duration(in minutes)

3, 3, Data Collection and Data Processing, Learn about abc,

Week No	Lecture No	Topic	Objectives	Duration (in minutes)
3	3	Data Collection and Data Processing	Learn about abc	90

2.1.2 JSON - Javascript Object Notation (JSON)

Syntax :

- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

JSON objects are written inside curly braces.

2.1.2 JSON - Javascript Object Notation (JSON)

Week No, Lecture No, Topic, Objectives, Duration(in minutes)

3, 3, Data Collection and Data Processing, Learn about abc,


```
{
   "Week No": 3,
   "Lecture No": 3,
   "Topic": " Data Collection and Data Processing",
   "Objectives": " Learn about abc",
   "Duration(in minutes)": 90
 }
```

2.1.3 HTML

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

2.1.4 XML (Extensible Markup Language)

- Provides rules to define any data.
- Very self descriptive

2.1.4 XML (Extensible Markup Language)

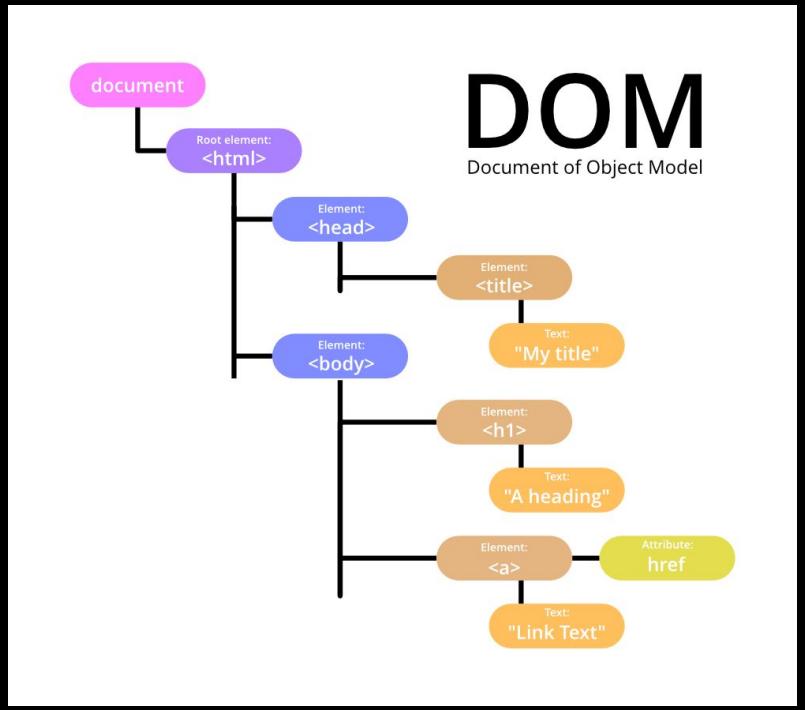
```
<announcement>
  <from>Instructor</from>
  <heading>Lecture 3.3</heading>
  <body>Lecture 3.3 on Data Collection and Data Processing is released.</body>
</announcement>
```

DOM (Document Object Model)

Defines the logical structure of documents and the way a document is accessed and manipulated

Definition from : https://www.w3.org/TR/WD-DOM/introduction.html

DOM (Document Object Model)



Definition from : https://www.geeksforgeeks.org/what-is-document-object-in-java-dom/

2.1.5 XHTML (EXtensible HyperText Markup Language)

A stricter, more XML-based version of HTML.


gifs.com

2. Data Processing

Lecture content/topics in this section is derived from Prof.John P Dickerson’s lectures in CMSC641

2.1 Libraries

Many Libraries to work with data

- Used for numerical and scientific computing

- Examples :
  - NumPy/SciPy – numerical and scientific function libraries.
  - numba – Python compiler that support JIT compilation.
  - ALGLIB – numerical analysis library.
  - pandas – high-performance data structures and data analysis
  - tools.
  - pyGSL – Python interface for GNU Scientific Library.
  - ScientificPython – collection of scientific computing modules

Credit : Prof.John P Dickerson’s lectures in CMSC641

Commonly used packages

- NumPy
- SciPy
- Matplotlib & Seaborn – plotting libraries
- iPython via Jupyter – interactive computing
- Pandas – data analysis library
- SymPy – symbolic computation library

Lecture content/topics in this section is derived from Prof.John P Dickerson’s lectures in CMSC641

Switch to Jupyter Notebook

In the this lecture,

- Research Design for Statistical Analysis
  - Causation versus Correlation
  - Sampling
- Revisit the Data Science Process
- Data Collection
  - Till SQL
  - API
  - Scraping
- Data File Types
- Data Processing:
  - Libraries
  - Numpy
  - Scipy
  - Restart pandas .. till datetime