YouTube Views Tracker

User Manual

Author: Coşkun Deniz <codenineeight@gmail.com>

Created at: March 2022

Table of Contents

Introduction

This tool monitors view counts for YouTube videos and reports the results on Excel or Google Sheets documents. The user can use either video URLs or channel links to track video counts.

Supported Features & Package Levels

There are 3 types of packages: basic(BP), standard(SP), and premium(PP).

Basic Package (BP)

- Read video URLs (up to 1000) from txt, csv, or xlsx files.
- Output results to Excel.
- Chart for the most watched 10 videos on Excel output.

Standard Package (SP)

- Features in BP
- Read video URLs from Google Sheets document.
- Read video URLs from channel links (up to 3).
- Output results to Google Sheets.
- Installation of the script on the user side.
- Test the script on the user side.

Premium Package (PP)

- Features in SP
- Read video URLs from channel links (up to 5).
- Schedule for repeated runs.
- Run as Docker container.

All packages contain the source code and user manual document.

How to Setup

Installing Dependencies

Run the following commands to install required packages in the project directory.

```
python -m venv env
source env/bin/activate
python -m pip install -r requirements.txt
```

Creating Docker Image Manually (PP)

Docker image will be provided with the delivered package, but if you want to build it yourself, run the following command in the project directory.

```
docker build -t yt_views_tracker .
```

Note that there is a dot at the end.

Install docker from here for your system if you don’t already have it.

If you want to create a tar file from Docker image, run the following command after build.

```
docker save yt_views_tracker:latest | gzip > yt_views_tracker.tar.gz
```

You can load the image on another system with the below command.

```
docker load -i yt_views_tracker.tar.gz
```

You will see an output like below.

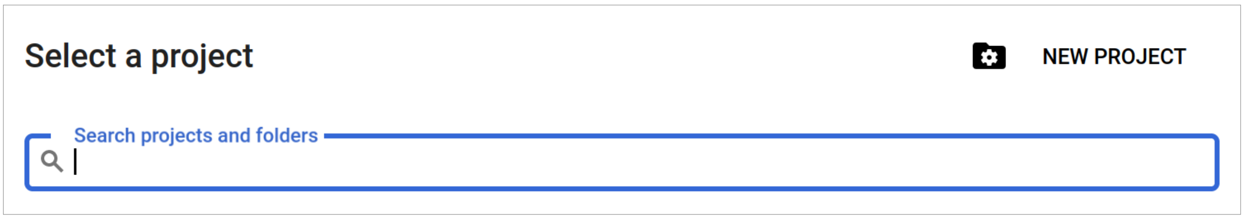
```
161bb48217a5: Loading layer
[==================================================>]  2.048kB/2.048kB
3ab11ff81fc5: Loading layer
[==================================================>]  108.5kB/108.5kB
293a28bb0e11: Loading layer
[==================================================>]   7.35MB/7.35MB
ae8c8923029b: Loading layer
[==================================================>]  9.649MB/9.649MB
Loaded image: yt_views_tracker:latest
```

If you type *docker images*, you can see the extracted image.

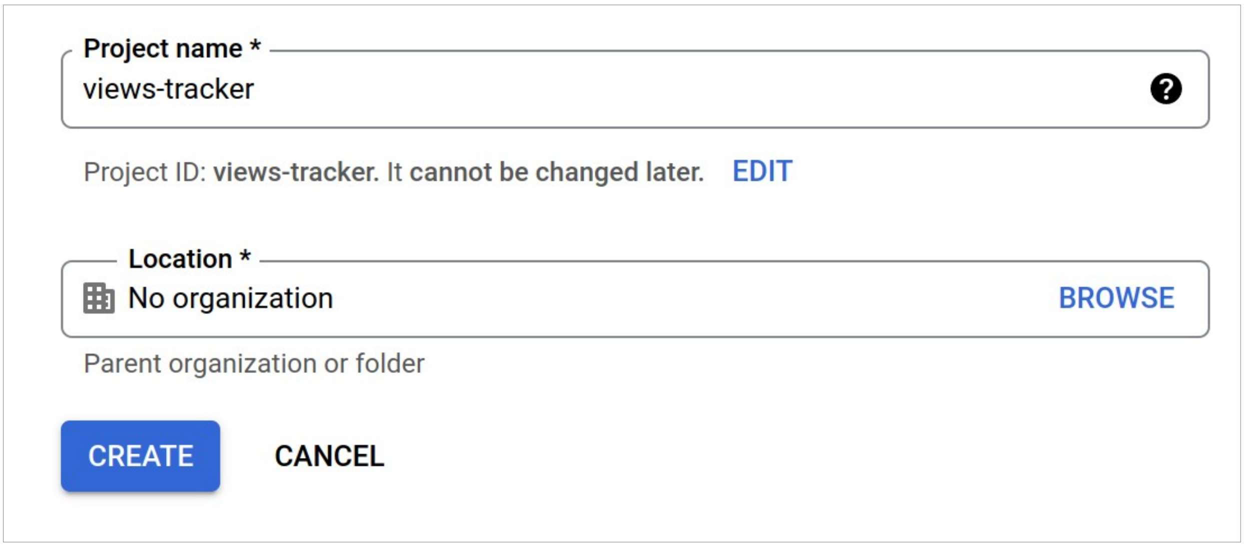
```
REPOSITORY                 TAG          IMAGE ID       CREATED        SIZE
yt_views_tracker           latest       907250b0ddd8   8 days ago     62.9MB
```

Creating a Project on Google Cloud Platform for Google Sheets Usage (SP, PP)

1. Go to Google Developers Console and create a new project (or select the one you already have).



Write a name for your project and click the *CREATE* button.



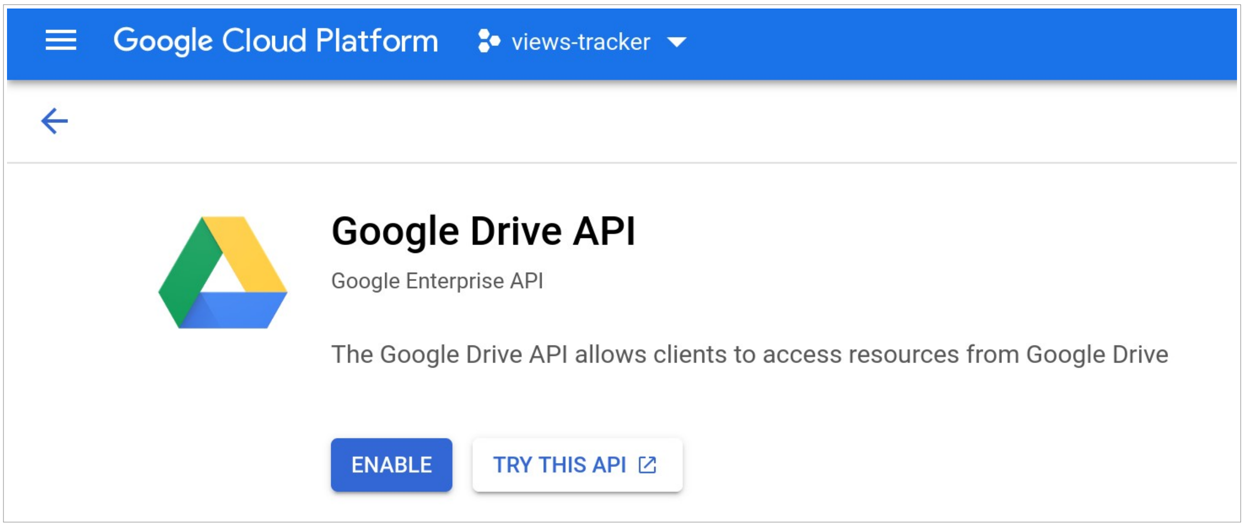
Project name * views-tracker

Project ID: views-tracker. It cannot be changed later.

Location * No organization

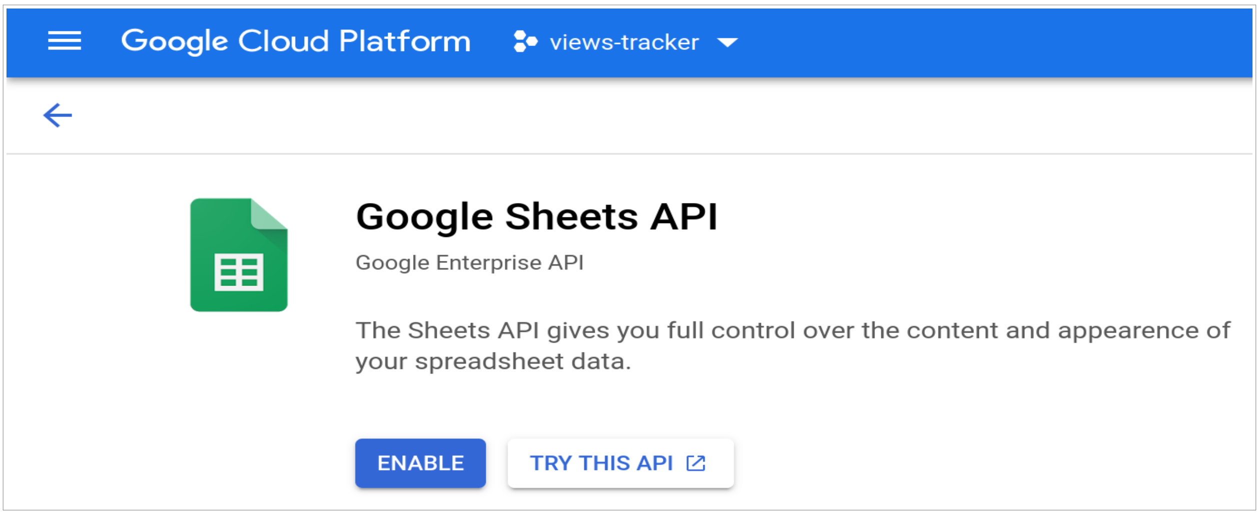
Parent organization or folder

2. Select APIs & Services → Library from the side menu. First search for “Google Drive API” and enable it.



The Google Drive API allows clients to access resources from Google Drive

Then search for “Google Sheets API” and enable it.



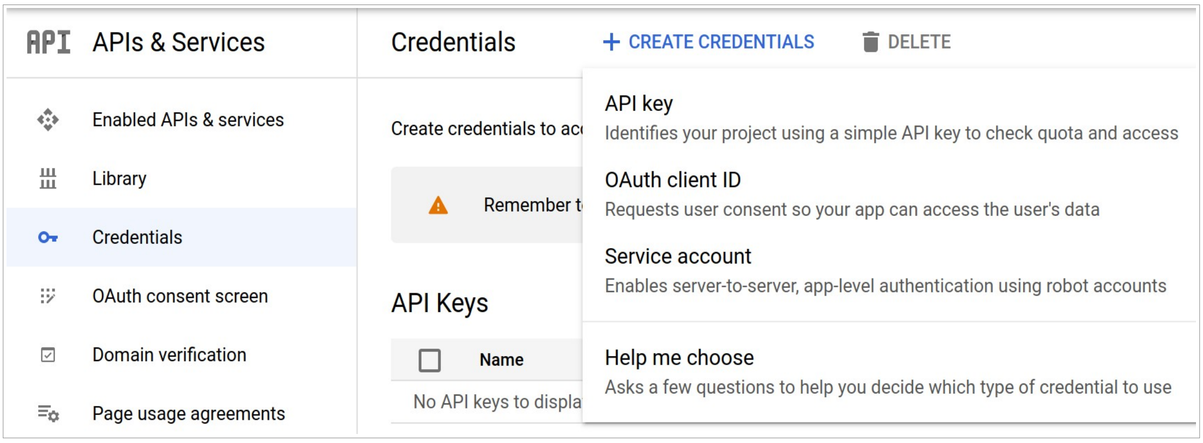
The Sheets API gives you full control over the content and appearence of your spreadsheet data.

3. Create credentials for service account.

Since it’s a separate account, by default it does not have access to any spreadsheet until you share it with this account. Just like any other Google account.

1. Go to APIs & Services → Credentials from the side menu.

2. Choose “CREATE CREDENTIALS > Service account”.



API key
Identifies your project using a simple API key to check quota and access

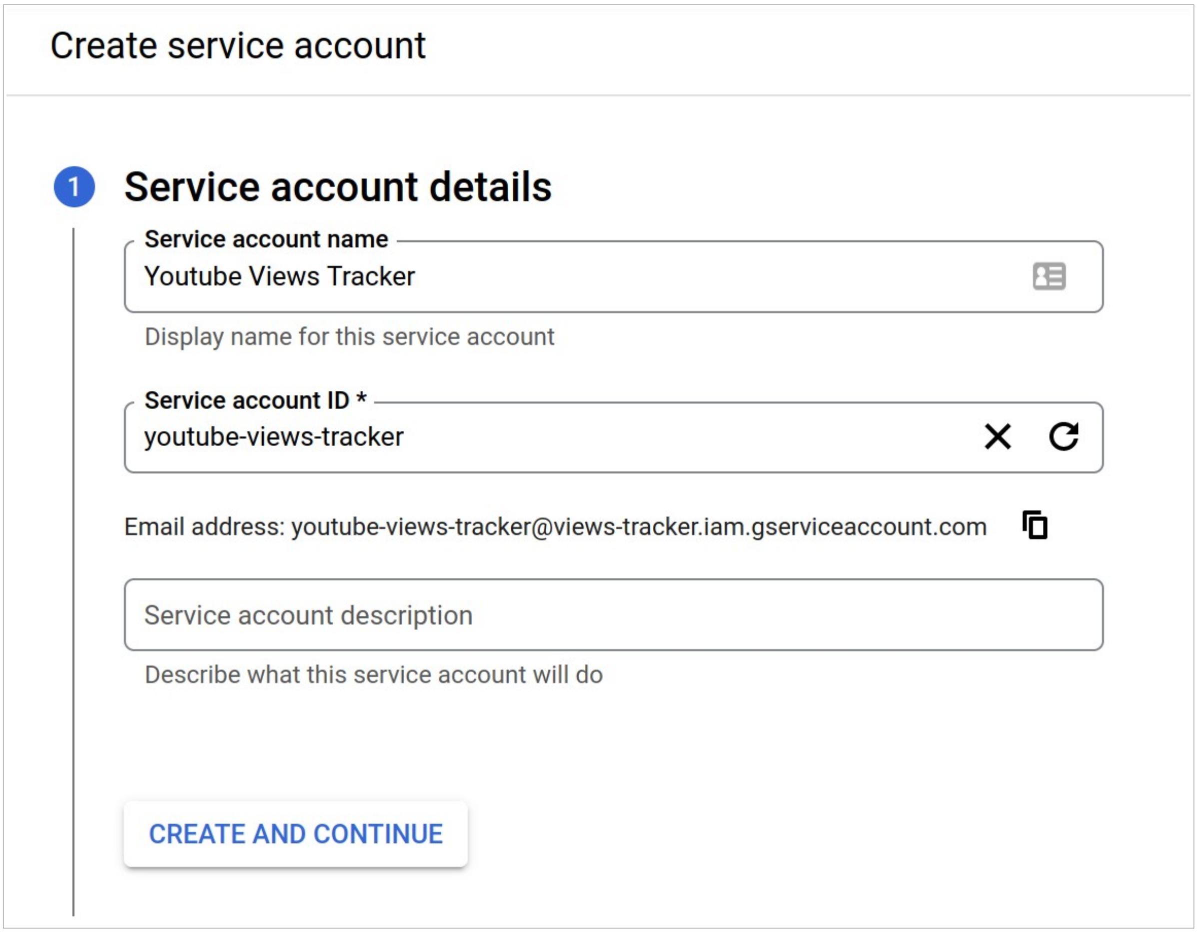
OAuth client ID
Requests user consent so your app can access the user's data

Service account
Enables server-to-server, app-level authentication using robot accounts

Help me choose
Asks a few questions to help you decide which type of credential to use

3. Fill out the form.

4. Click “CREATE AND CONTINUE” and “DONE”.



Create service account

1 Service account details

Service account name: Youtube Views Tracker

Display name for this service account

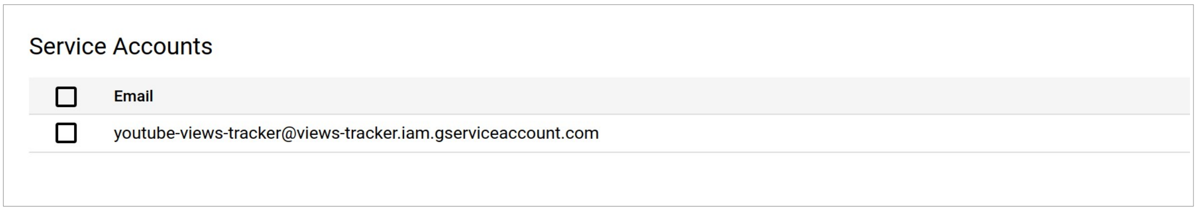
Service account ID *: youtube-views-tracker

Email address: youtube-views-tracker@views-tracker.iam.gserviceaccount.com

Service account description

Describe what this service account will do

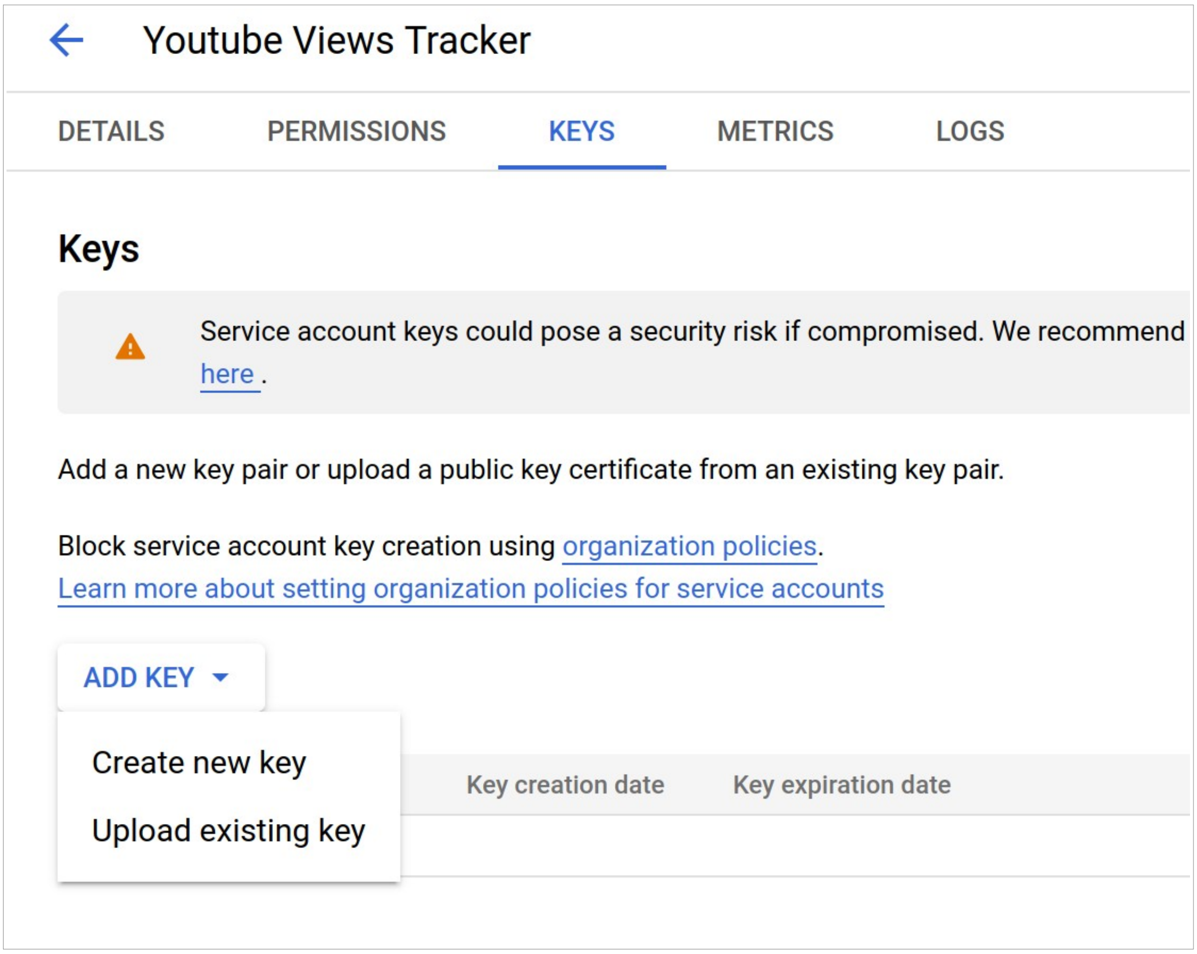
5. Click on the email link under Service Accounts section.



Service Accounts

□	Email
□	youtube-views-tracker@views-tracker.iam.gserviceaccount.com

6. Go to KEYS tab and select “Create new key” from “ADD KEY” drop down menu.

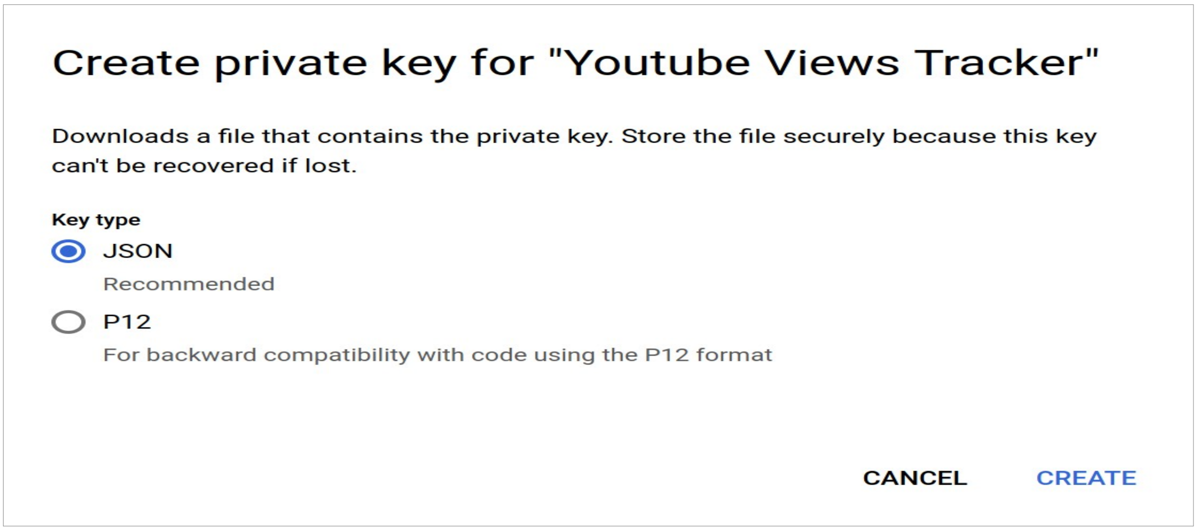


Service account keys could pose a security risk if compromised. We recommend
here .

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using organization policies.
Learn more about setting organization policies for service accounts

7. Select JSON from the popup menu and click CREATE.



Create private key for "Youtube Views Tracker"

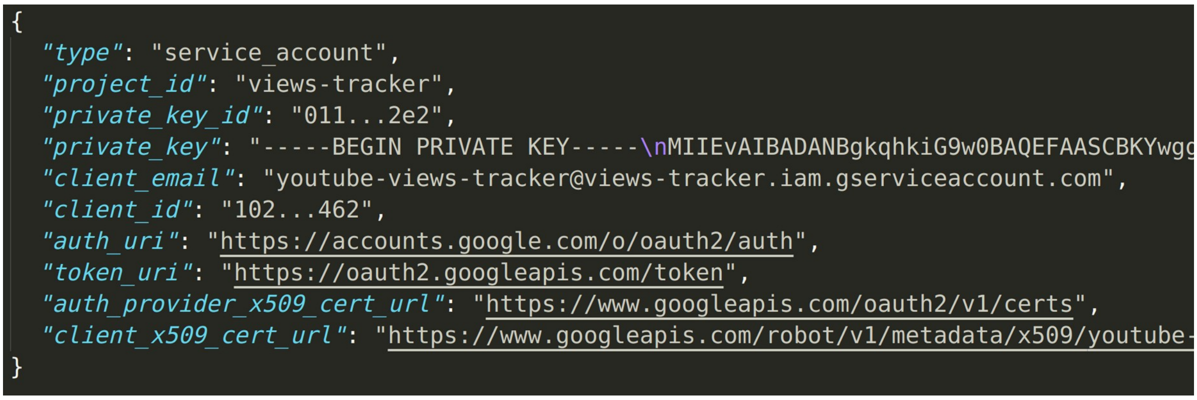
Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

Key type

● JSON
Recommended

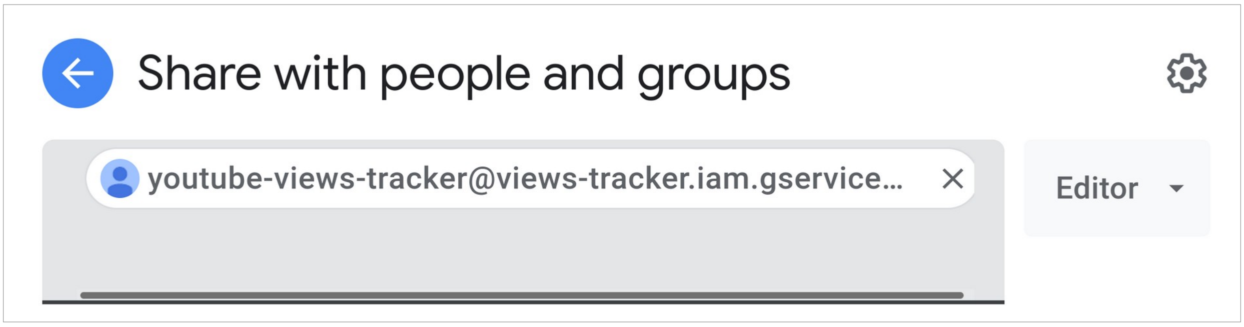
○ P12
For backward compatibility with code using the P12 format

8. JSON file with credentials will be downloaded automatically. It may look like this



```
{
  "type": "service_account",
  "project_id": "views-tracker",
  "private_key_id": "011...2e2",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwgg
  "client_email": "youtube-views-tracker@views-tracker.iam.gserviceaccount.com",
  "client_id": "102...462",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/youtube-
}
```

4. Rename the downloaded file as “credentials.json” and copy it to the project directory.
5. Share your document with the value of “client_email” field if you will read urls from a Google Sheets document.



Share with people and groups

youtube-views-tracker@views-tracker.iam.gservice...

Editor

How to Use

Basic Usage

For the most basic usage, the script requires at least one of “-c”, “-f”, or “-ch” options for config file, urls file and channel links respectively.

```
python yt_views_tracker.py -c
python yt_views_tracker.py -f ~/video_urls.txt
python yt_views_tracker.py -ch “https://www.youtube.com/c/ArjanCodes/”
```

Results are written to results.xlsx file by default in views, title, url order.

Logs can be seen in the yt_views.log file in the project directory.

Supported Arguments

Short Argument	Long Argument	Description	Default Value
-h	--help	Show help message and exit.	
-c	--useconfig	Read configuration from json file.	False
-cf	--configfile	Read configuration from given json file.	config.json
-f	--urlsfile	Read video urls from given file.	
-ch	--channels	Channel urls separated by comma.	
-ot	--output_type	Output file type (excel or gsheets).	excel
-of	--output_file	Output file name.	results.xlsx
-uc	--url_column	Url column index for csv, xlsx, or Google Sheets input.	0
-sm	--share_mail	Mail address to share Google Sheets document.	
-s	--schedule	Interval to run as scheduled task.	NONE

The above list contains all options that are supported in the Premium Package. If you have other packages, one or more options may not be available for your package. Please see the Supported Features & Package Levels section.

Example Commands

- Read video URLs from video_urls.csv and output results to video_view_counts.xlsx file.

  ```
  python yt_views_tracker.py -f ~/video_urls.csv -of video_view_counts.xlsx
  ```

- Use options from custom_config.json file.

  ```
  python yt_views_tracker.py -c -cf custom_config.json
  ```

- Read video URLs from the third column of video_urls.csv file.

  ```
  python yt_views_tracker.py -f ~/video_urls.csv -uc 2
  ```

- If you will use Google Sheets for getting video URL inputs, add “gsheets-” prefix to the urlsfile(f) parameter.

  ```
  python yt_views_tracker.py -f gsheets-video_urls
  ```

  This command will read URLs from Google Sheets document named “video_urls”.

- Read video URLs from Google Sheets document using file link.

  ```
  python yt_views_tracker.py -f 
  gsheets-https://docs.google.com/spreadsheets/d/1dtFZbg4Gm8mCopO9fpZ-
  DwJ37uLMopgLePgFdigutuI
  ```

- Read video URLs from txt file and output results to Google Sheets document named “view_results”.

  ```
  python yt_views_tracker.py -f ~/video_urls.txt -ot gsheets -of 
  view_results -sm codenineeight@gmail.com
  ```

  Share mail parameter is needed, so that you can see the document on your account.

- If you will use config file and read URLs from channel links, you should keep the *urlsfile* field empty in the json file.

  ```
  {
        "urlsfile": "",
        "channels": [
            "https://www.youtube.com/c/ArjanCodes/",
            "https://www.youtube.com/c/Coreyms"
        ],
        "output_type": "excel",
        "output_file": "results.xlsx",
        "url_column": 0,
        "share_mail": "codenineeight@gmail.com",
        "schedule": "NONE"
  }
  ```

  ```
  python yt_views_tracker.py -c
  ```

Scheduling (PP)

You can run the script in scheduled intervals. Scheduling is not active by default so the script will run only once and exit after completion. You have 9 scheduling options as shown below.

1. 10 minutes (MIN10)
2. 15 minutes (MIN15)
3. 30 minutes (MIN30)
4. 1 hour (HOUR1)
5. 3 hours (HOUR3)
6. 6 hours (HOUR6)
7. 8 hours (HOUR8)
8. 12 hours (HOUR12)
9. Once a day at "HH:MM"

- Run every 30 minutes

  ```
  python yt_views_tracker.py -f video_urls.csv -s MIN30
  ```

- Run once a day at 15:30

  ```
  python yt_views_tracker.py -f video_urls.csv -s 15:30
  ```

Running Docker Image (PP)

- Run the following command first for once.

  ```
  docker run --rm -it -v view_results_volume:/src yt_views_tracker
  ```

  view_results_volume is the storage that will be shared with docker container.
  yt_views_tracker is the name of the image.

- If you will use with Google Sheets, copy your credentials file to the shared directory.

  ```
  sudo cp /path/to/credentials.json /var/lib/docker/volumes/view_results_volume/_data
  ```

- If you will use the urls file, copy urls file to the shared directory.

  ```
  sudo cp /path/to/video_urls.txt /var/lib/docker/volumes/view_results_volume/_data
  ```

- Run the container by passing arguments as usual.

  ```
  docker run --rm -it -v view_results_volume:/src yt_views_tracker -f video_urls.csv
  ```

  ```
  docker run --rm -it -v view_results_volume:/src yt_views_tracker -ch
  "https://www.youtube.com/c/ArjanCodes/"
  ```

  ```
  docker run --rm -it -v view_results_volume:/src yt_views_tracker -f video_urls.txt -ot
  gsheets -of view_results -sm codenineeight@gmail.com
  ```

- Copy the results to a different directory on the host device.

  ```
  sudo cp /var/lib/docker/volumes/view_results_volume/_data/results.xlsx ~/views_output
  ```

- If you will use config file, copy your config file to the shared directory, and run the container with "-c" and "-cf" options.

  ```
  sudo cp /path/to/config_for_docker.json /var/lib/docker/volumes/view_results_volume/_data
  ```

  ```
  docker run --rm -it -v view_results_volume:/src yt_views_tracker -c -cf 
  config_for_docker.json
  ```

See here for sharing volumes on Windows.

Running Tests

Install the *pytest* package.

```
python -m pip install pytest
```

Run the following command in the project directory.

```
pytest -vs
```

If you want to check coverage report, install the coverage package first.

```
python -m pip install coverage
```

Run the tests with the coverage command.

```
coverage run --omit *dist-packages* -m pytest -vs
```

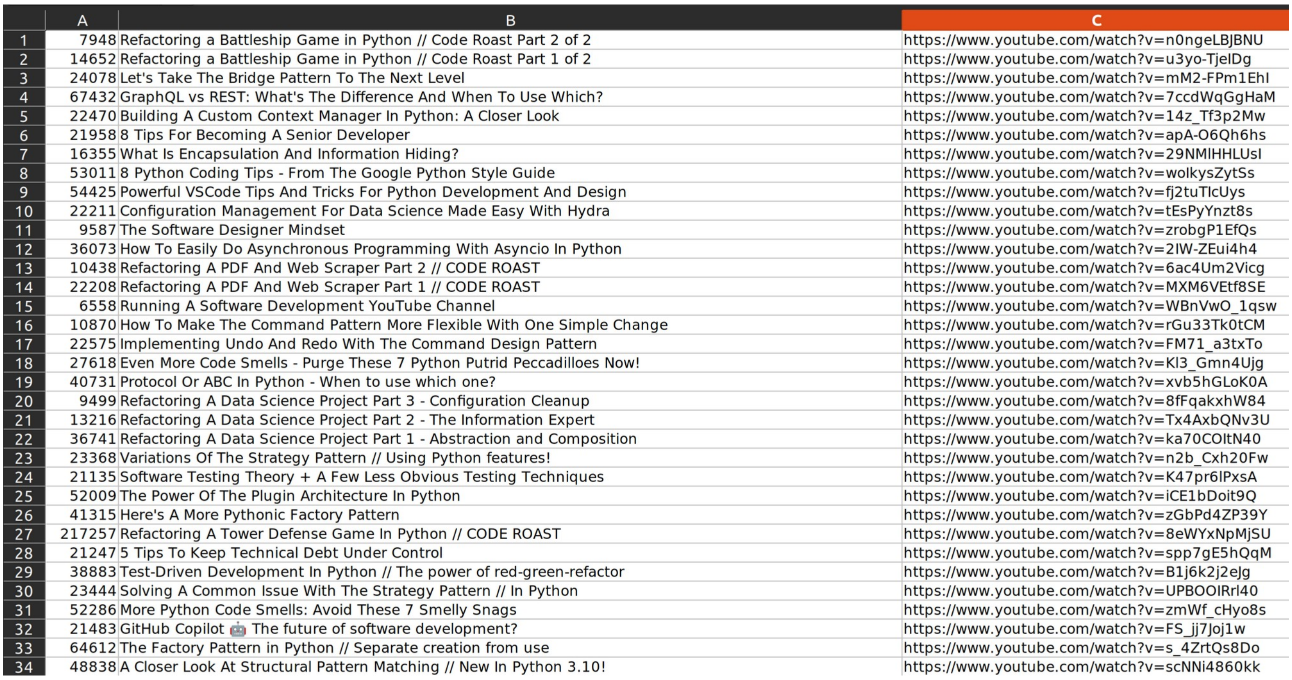
Run the following command to see the html report.

```
coverage html --omit *dist-packages,yt_views_tracker.py*
```

You can see the report by opening index.html file located under htmlcov folder in the project directory.

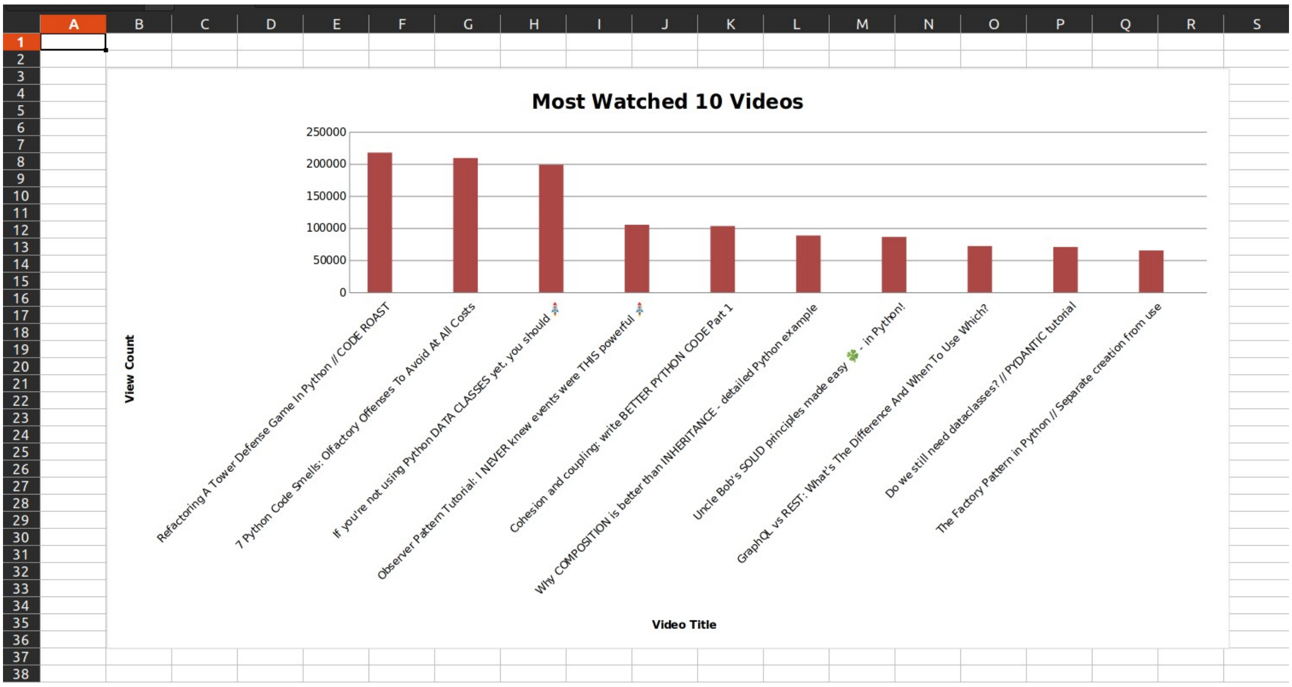
Screenshots

Excel Result

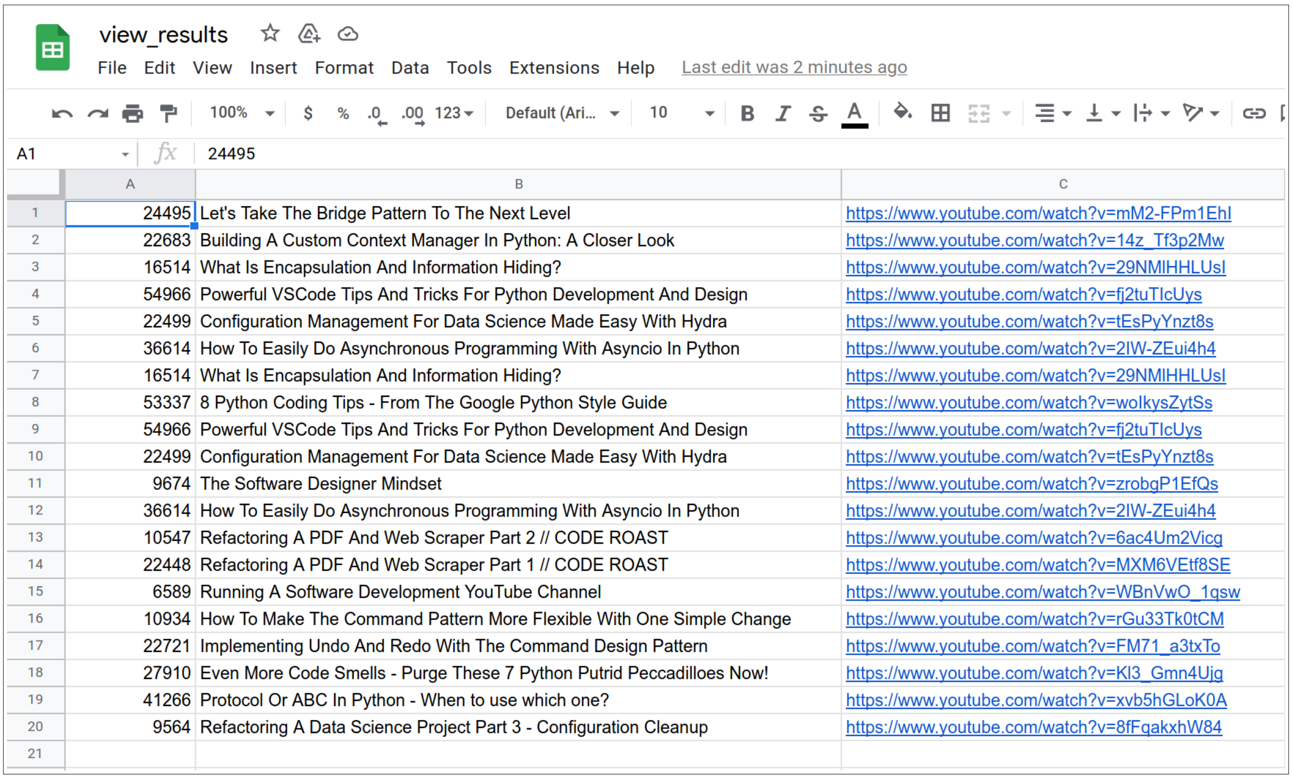


	A	B	C
1	7948	Refactoring a Battleship Game in Python // Code Roast Part 2 of 2	https://www.youtube.com/watch?v=n0ngeLBJBNU
2	14652	Refactoring a Battleship Game in Python // Code Roast Part 1 of 2	https://www.youtube.com/watch?v=u3yo-TjelDg
3	24078	Let's Take The Bridge Pattern To The Next Level	https://www.youtube.com/watch?v=mM2-FPm1EhI
4	67432	GraphQL vs REST: What's The Difference And When To Use Which?	https://www.youtube.com/watch?v=7ccdWqGgHaM
5	22470	Building A Custom Context Manager In Python: A Closer Look	https://www.youtube.com/watch?v=14z_Tf3p2Mw
6	21958	8 Tips For Becoming A Senior Developer	https://www.youtube.com/watch?v=apA-O6Qh6hs
7	16355	What Is Encapsulation And Information Hiding?	https://www.youtube.com/watch?v=29NMlHHLUsI
8	53011	8 Python Coding Tips - From The Google Python Style Guide	https://www.youtube.com/watch?v=wolkysZytSs
9	54425	Powerful VSCode Tips And Tricks For Python Development And Design	https://www.youtube.com/watch?v=fj2tuTIcUys
10	22211	Configuration Management For Data Science Made Easy With Hydra	https://www.youtube.com/watch?v=tEsPyYnzt8s
11	9587	The Software Designer Mindset	https://www.youtube.com/watch?v=zrobgP1EfQs
12	36073	How To Easily Do Asynchronous Programming With Asyncio In Python	https://www.youtube.com/watch?v=2IW-ZEui4h4
13	10438	Refactoring A PDF And Web Scraper Part 2 // CODE ROAST	https://www.youtube.com/watch?v=6ac4Um2Vicg
14	22208	Refactoring A PDF And Web Scraper Part 1 // CODE ROAST	https://www.youtube.com/watch?v=MXM6VEtf8SE
15	6558	Running A Software Development YouTube Channel	https://www.youtube.com/watch?v=WBnVwO_1qsw
16	10870	How To Make The Command Pattern More Flexible With One Simple Change	https://www.youtube.com/watch?v=rGu33Tk0tCM
17	22575	Implementing Undo And Redo With The Command Design Pattern	https://www.youtube.com/watch?v=FM71_a3txTo
18	27618	Even More Code Smells - Purge These 7 Python Putrid Peccadilloes Now!	https://www.youtube.com/watch?v=Kl3_Gmn4Ujg
19	40731	Protocol Or ABC In Python - When to use which one?	https://www.youtube.com/watch?v=xvb5hGLoK0A
20	9499	Refactoring A Data Science Project Part 3 - Configuration Cleanup	https://www.youtube.com/watch?v=8fFqakxhW84
21	13216	Refactoring A Data Science Project Part 2 - The Information Expert	https://www.youtube.com/watch?v=Tx4AxbQNv3U
22	36741	Refactoring A Data Science Project Part 1 - Abstraction and Composition	https://www.youtube.com/watch?v=ka70COltN40
23	23368	Variations Of The Strategy Pattern // Using Python features!	https://www.youtube.com/watch?v=n2b_Cxh20Fw
24	21135	Software Testing Theory + A Few Less Obvious Testing Techniques	https://www.youtube.com/watch?v=K47pr6lPxsA
25	52009	The Power Of The Plugin Architecture In Python	https://www.youtube.com/watch?v=iCE1bDoit9Q
26	41315	Here's A More Pythonic Factory Pattern	https://www.youtube.com/watch?v=zGbPd4ZP39Y
27	217257	Refactoring A Tower Defense Game In Python // CODE ROAST	https://www.youtube.com/watch?v=8eWYxNpMjSU
28	21247	5 Tips To Keep Technical Debt Under Control	https://www.youtube.com/watch?v=spp7gE5hQqM
29	38883	Test-Driven Development In Python // The power of red-green-refactor	https://www.youtube.com/watch?v=B1j6k2j2eJg
30	23444	Solving A Common Issue With The Strategy Pattern // In Python	https://www.youtube.com/watch?v=UPBOOIRrl40
31	52286	More Python Code Smells: Avoid These 7 Smelly Snags	https://www.youtube.com/watch?v=zmWf_cHyo8s
32	21483	GitHub Copilot 🤖 The future of software development?	https://www.youtube.com/watch?v=FS_jj7Joj1w
33	64612	The Factory Pattern in Python // Separate creation from use	https://www.youtube.com/watch?v=s_4ZrtQs8Do
34	48838	A Closer Look At Structural Pattern Matching // New In Python 3.10!	https://www.youtube.com/watch?v=scNNi4860kk

Excel Chart



Google Sheets Result


view_results
File Edit View Insert Format Data Tools Extensions Help
Last edit was 2 minutes ago
100%
$ % .0 .00 123
Default (Ari...
10
B I S A
A1
fx

A1: 24495

	A	B	C
1	24495	Let's Take The Bridge Pattern To The Next Level	https://www.youtube.com/watch?v=mM2-FPm1EhI
2	22683	Building A Custom Context Manager In Python: A Closer Look	https://www.youtube.com/watch?v=14z_Tf3p2Mw
3	16514	What Is Encapsulation And Information Hiding?	https://www.youtube.com/watch?v=29NMIHHLUsI
4	54966	Powerful VSCode Tips And Tricks For Python Development And Design	https://www.youtube.com/watch?v=fj2tuTIcUys
5	22499	Configuration Management For Data Science Made Easy With Hydra	https://www.youtube.com/watch?v=tEsPyYnzt8s
6	36614	How To Easily Do Asynchronous Programming With Asyncio In Python	https://www.youtube.com/watch?v=2IW-ZEui4h4
7	16514	What Is Encapsulation And Information Hiding?	https://www.youtube.com/watch?v=29NMIHHLUsI
8	53337	8 Python Coding Tips - From The Google Python Style Guide	https://www.youtube.com/watch?v=woIkysZytSs
9	54966	Powerful VSCode Tips And Tricks For Python Development And Design	https://www.youtube.com/watch?v=fj2tuTIcUys
10	22499	Configuration Management For Data Science Made Easy With Hydra	https://www.youtube.com/watch?v=tEsPyYnzt8s
11	9674	The Software Designer Mindset	https://www.youtube.com/watch?v=zrobgP1EfQs
12	36614	How To Easily Do Asynchronous Programming With Asyncio In Python	https://www.youtube.com/watch?v=2IW-ZEui4h4
13	10547	Refactoring A PDF And Web Scraper Part 2 // CODE ROAST	https://www.youtube.com/watch?v=6ac4Um2Vicg
14	22448	Refactoring A PDF And Web Scraper Part 1 // CODE ROAST	https://www.youtube.com/watch?v=MXM6VEtf8SE
15	6589	Running A Software Development YouTube Channel	https://www.youtube.com/watch?v=WBnVwO_1qsw
16	10934	How To Make The Command Pattern More Flexible With One Simple Change	https://www.youtube.com/watch?v=rGu33Tk0tCM
17	22721	Implementing Undo And Redo With The Command Design Pattern	https://www.youtube.com/watch?v=FM71_a3txTo
18	27910	Even More Code Smells - Purge These 7 Python Putrid Peccadilloes Now!	https://www.youtube.com/watch?v=Kl3_Gmn4Ujg
19	41266	Protocol Or ABC In Python - When to use which one?	https://www.youtube.com/watch?v=xvb5hGLoK0A
20	9564	Refactoring A Data Science Project Part 3 - Configuration Cleanup	https://www.youtube.com/watch?v=8fFqakxhW84
21			

Support

If you need support, you can contact me by emailing to codenineeight@gmail.com with the “yt_views_tracker” prefix in the subject. You can also see my Upwork profile here.