

YouTube Comments Retrieval Script Documentation

Overview

This script utilizes the YouTube Data API to retrieve comments from videos in a specified channel. It follows a structured process, including obtaining the channel ID, playlist ID, and then retrieving video details and comments.

Prerequisites

1. **API Key:** You need a valid YouTube Data API key to authenticate your requests.
2. **Python Libraries:** Ensure you have the necessary Python libraries installed. You can install them using: “pip install google-api-python-client”

Code:

```
# Import required modules

api_key = 'YOUR_API_KEY' #can be acquired from google developers youtube v3 section

from googleapiclient.discovery import build

from googleapiclient.errors import HttpError

# Build the YouTube API service

youtube = build('youtube', 'v3', developerKey=api_key)

# Specify the channel username or ID

channel_username = 'sample_username'

# Get the channel ID from the provided username

channels_response = youtube.channels().list(part='id',
forUsername=channel_username).execute()

channel_id = channels_response['items'][0]['id']

# Get the videos from the channel

playlist_response = youtube.channels().list(part='contentDetails',
id=channel_id).execute()

playlist_id = playlist_response['items'][0]['contentDetails']['relatedPlaylists']['uploads']

# Get the video details from the playlist
```

```

videos_response = youtube.playlistItems().list(part='snippet',
playlistId=playlist_id).execute()

# Retrieve comments from each video

for video in videos_response['items']:

    video_id = video['snippet']['resourceId']['videoId']

    try:

        comments_response = youtube.commentThreads().list(part='snippet',
videoId=video_id).execute()

        # Print comments

        for comment in comments_response['items']:

            text_display = comment['snippet']['topLevelComment']['snippet']['textDisplay']

            print(text_display)

    except HttpError as e:

        # Check if comments are disabled

        error_details = e.error_details[0]

        if 'commentsDisabled' in error_details['reason']:

            print(f"Comments are disabled for the video with ID {video_id}")

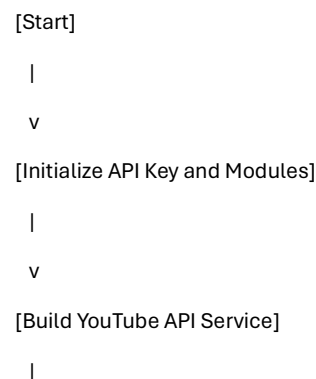
        else:

            # Handle other errors

            print(f"Error retrieving comments for video with ID {video_id}: {e}")

```

Flow Chart



```
v
[Retrieve Channel ID]
|
v
[Retrieve Playlist ID]
|
v
[Retrieve Video Details from Playlist]
|
v
[Iterate Over Videos]
|
v
[Retrieve Comments for Each Video]
|
v
[Print Comments]
|
v
[Handle Exceptions]
|
v
[End]
```

Output

The script outputs the comments for each video in the specified channel. Comments are printed to the console. If comments are disabled for a video, it indicates that in the output.

Example:

```

api_key = 'AIzaSyDFvWQqeA62QPwrSNVLlY6w985W_3h6-qM'

from googleapiclient.discovery import build
from googleapiclient.errors import HttpError

# Build the YouTube API service
youtube = build('youtube', 'v3', developerKey=api_key)

# Specify the channel username or ID
channel_username = 'pikachu'

# Get the channel ID from the provided username
channels_response = youtube.channels().list(part='id', forUsername=channel_username).execute()
channel_id = channels_response['items'][0]['id']

# Get the videos from the channel
playlist_response = youtube.channels().list(part='contentDetails', id=channel_id).execute()
playlist_id = playlist_response['items'][0]['contentDetails']['relatedPlaylists']['uploads']

# Get the video details from the playlist
videos_response = youtube.playlistItems().list(part='snippet', playlistId=playlist_id).execute()

# Retrieve comments from each video
for video in videos_response['items']:
    video_id = video['snippet']['resourceId']['videoId']

    try:
        comments_response = youtube.commentThreads().list(part='snippet', videoId=video_id).execute()

        # Print comments
        for comment in comments_response['items']:
            text_display = comment['snippet']['topLevelComment']['snippet']['textDisplay']
            print(text_display)

    except HttpError as e:
        # Check if comments are disabled
        error_details = e.error_details[0]

```

```

except HttpError as e:
    # Check if comments are disabled
    error_details = e.error_details[0]
    if 'commentsDisabled' in error_details['reason']:
        print(f"Comments are disabled for the video with ID {video_id}")
    else:
        # Handle other errors
        print(f"Error retrieving comments for video with ID {video_id}: {e}")

```

Output:

```
PS C:\Users\DND\python> python -u "c:\Users\DND\python\twitter_code\yt_stealer.py"
Holy hell that's the real pikachu
I dont care how far it is, i m running like that to pokemon center as soon as the fight ends♥
Pikachu where are you going
The cameraman: <br><br>Run more faster!
<b>Pikachu using quick attack</b>
3E♥
Hola 🙌
Hola 🙌
Hola 🙌
HOHOHOH Wow 😮
RUN
😬
danggg, actual pikachu channel with the username, awesome!
Ohh look, another unoriginal YTer jumping on the band wagon.
Bro is it me or is it viral because every time I see someone run and I kept hearing the song
Antidote from route 1 🤔
So true
Just give up
Yea
```