Report on YouTube Comments Extraction using Python into CSV Export Script

Open Source Programming club

Chandra Srinivas P 21BCE1593

Anand Nimmalapudi 21BCE5945

1. Introduction:

The provided Python script leverages the YouTube Data API to extract comments from a specified video and saves them in CSV format. This report provides an overview of the script, its functionality, and steps to execute it

2. Script Overview:

2.1. Dependencies:

The script utilizes the following Python libraries:

- **google-api-python-client**: Provides an interface to the YouTube Data API.
- **csv**: Enables reading and writing CSV files.

2.2. Authentication:

The script uses an API key for authentication, and the provided key "AIzaSyDtz1Jxn2KwMRBA282sLPingrY2uOXqnC0" is incorporated into the script.

2.3. YouTube API Client Setup:

The script sets up a YouTube API client using the **googleapiclient.discovery.build** method, specifying the API key, service name ('youtube'), and API version ('v3').

CODE:(Python)

```
# Importing necessary libraries
import csv  # For handling CSV files
import google_auth_oauthlib.flow
import googleapiclient.discovery
import googleapiclient.errors
```

```
from google.oauth2.credentials import Credentials


# Setting up API credentials
api_key = 'YOUR_API_KEY'  # Your YouTube Data API key
api_service_name = 'youtube'
api_version = 'v3'


# Creating a YouTube API client
youtube = googleapiclient.discovery.build(api_service_name, api_version,
developerKey=api_key)


# Specifying the video ID for the comments you want to extract
video_id = 'YOUR_VIDEO_ID'  # The ID of the video from which you want to extract
comments


# Requesting comments for the specified video
request = youtube.commentThreads().list(
    part='snippet',
    videoId=video_id,
    textFormat='plainText'
)
response = request.execute()


# Extracting comments
comments_data = []


for comment in response['items']:
    # Extracting author and comment text
    snippet = comment['snippet']['topLevelComment']['snippet']
    author = snippet['authorDisplayName']
```

```
    text = snippet['textDisplay']
    comments_data.append([author, text])

# Saving comments to CSV file
csv_file_path = 'youtube_comments.csv'  # Path to save the CSV file

with open(csv_file_path, 'w', newline='', encoding='utf-8') as csv_file:
    csv_writer = csv.writer(csv_file)
    csv_writer.writerow(['Author', 'Comment'])  # Writing header row

    # Writing each comment to the CSV file
    for comment_data in comments_data:
        csv_writer.writerow(comment_data)

print(f'Comments have been saved to {csv_file_path}')  # Printing a message indicating successful saving
```

2.4. Video and Comment Retrieval:

The user-provided video ID "UP1tWImU_b8" is used to send a request to the YouTube API to retrieve comments associated with the specified video.

2.5. CSV Data Preparation:

The script extracts relevant information (author and text) from the API response and organizes it into a list (**comments_data**) for further processing.

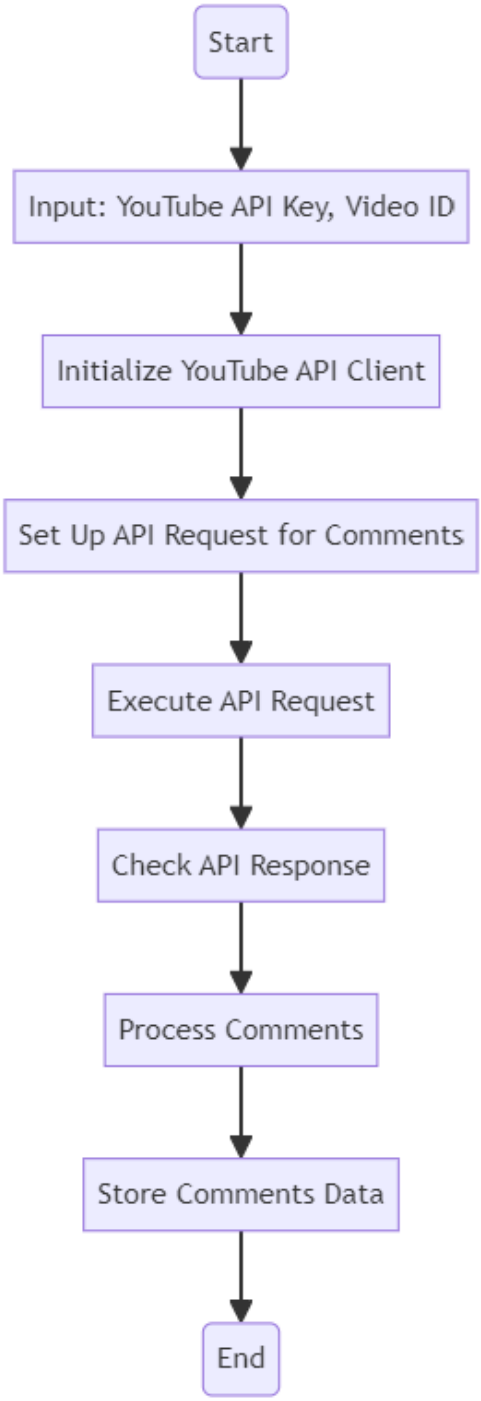
2.6. CSV File Creation:

The script creates a CSV file named 'youtube_comments.csv' in the same directory as the script. The extracted comments, along with author names, are written to this CSV file using the **csv.writer** module.

2.7. Execution and Output:

When executed, the script prints the comments to the console and saves them in a CSV file. The user is informed of the file's creation and location.

Flowchart:


Start
Input: YouTube API Key, Video ID
Initialize YouTube API Client
Set Up API Request for Comments
Execute API Request
Check API Response
Process Comments
Store Comments Data
End

3. Usage:

3.1. Prerequisites:

- Python installed on the system.
- Required libraries installed using **pip install google-api-python-client google-auth google-auth-oauthlib google-auth-httplib2**.

3.2. API Key:

- The provided API key "AIzaSyDtz1Jxn2KwMRBA282sLPingrY2uOXqnC0" is used in the script.

3.3. Video ID:

- The video ID "UP1tWImU_b8" is used for comment extraction. Users can replace it with the desired video ID.

3.4. Execution:

- Run the script using **python script_name.py** in the command prompt.

4. Conclusion:

This script simplifies the extraction of comments from the specified YouTube video using an API key. Its functionality serves as a foundation for advanced YouTube API interactions.

This guide aims to assist users in effectively utilizing the script with the provided API key and video ID while being mindful of YouTube's limitations and costs. Setting up a virtual environment and understanding the quota system are essential for efficient and cost-effective data gathering.

YouTube data offers valuable insights for sentiment analysis, trend research, or general curiosity. This guide equips users with the necessary knowledge and tools to initiate their data extraction journey confidently. By comprehending YouTube's intricacies and maximizing the script's capabilities, users can effectively leverage the platform's data for diverse purposes.

GIT HUB repository link:-

https://github.com/AnandNimmalapudi/OPSC_Mavericks.git

Extracted csv file:-

https://drive.google.com/file/d/1RHD2WQdAd91de95QyYUs3ns7Ov3RdzqP/view?usp=drivesdk