YouTube Data API

is it possible to crawl and download the following video data from Youtube?
Engagement metrics such as:
Likes
Shares
Views
Comments
View Through Rate
Video Title
Video Description
etc.

Yes, it is possible to retrieve various engagement metrics and metadata from YouTube videos using official APIs and tools. Here's how you can access the data you're interested in:

✅ Engagement Metrics and Video Metadata

The **YouTube Data API v3** allows you to fetch the following public data for any video: Medium +1

- **Likes** (if publicly visible) Google for Developers
- **Views**
- **Comments count**
- **Video title**
- **Video description**
- **Thumbnails**
- **Published date**

To retrieve this information, use the `Videos.list` endpoint with the `part` parameters set to `snippet` and `statistics`. You can request data for up to 50 video IDs per call. Stack Overflow

For example, in Python:

```python
```

```
from googleapiclient.discovery import build

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

video_ids = ['VIDEO_ID1', 'VIDEO_ID2']  # Replace with actual video IDs
response = youtube.videos().list(
    part='snippet,statistics',
    id=','.join(video_ids)
).execute()

for item in response['items']:
    title = item['snippet']['title']
    description = item['snippet']['description']
    views = item['statistics'].get('viewCount')
    likes = item['statistics'].get('likeCount')
    comments = item['statistics'].get('commentCount')
    print(f"Title: {title}, Views: {views}, Likes: {likes}, Comments: {comments}")
```

💬 Retrieving Comments

To fetch comments from a video, use the `commentThreads().list` method: Medium

python

```
comments = []
next_page_token = None

while True:
    response = youtube.commentThreads().list(
        part='snippet',
        videoId='VIDEO_ID',  # Replace with actual video ID
        textFormat='plainText',
        maxResults=100,
        pageToken=next_page_token
    ).execute()

    for item in response['items']:
        comment = item['snippet']['topLevelComment']['snippet']['textDisplay']
        comments.append(comment)

    next_page_token = response.get('nextPageToken')
    if not next_page_token:
        break
```

This script retrieves up to 100 comments per page and continues fetching until all comments are retrieved.

📊 Additional Metrics via YouTube Analytics API

For channel owners, the **YouTube Analytics API** provides deeper insights, including:

Google for Developers +1

- **Shares**
- **Average view duration**
- **View-through rate** (as `averageViewPercentage`)
- **Subscribers gained/lost**
- **Estimated minutes watched**

To access these metrics, you must authenticate as the channel owner and use the `reports.query` endpoint with appropriate `metrics` and `dimensions` parameters. For example, to get metrics per video:

```http
GET https://youtubeanalytics.googleapis.com/v2/reports
?ids=channel==MINE
&startDate=2025-01-01
&endDate=2025-04-10
&metrics=views,likes,shares,averageViewDuration
&dimensions=video
&access_token=YOUR_ACCESS_TOKEN
```

This request retrieves views, likes, shares, and average view duration for each video on your channel within the specified date range.

🔍 Third-Party Tools

If you prefer not to implement API calls yourself, third-party tools like [LenosTube's Engagement Calculator] can provide quick insights into a video's engagement metrics. These tools typically require only the video URL and can display metrics such as likes, comments, and estimated engagement rates. Lenos

---

⚠️ Important Considerations

- **API Quotas**: The YouTube Data API has usage quotas. Ensure you monitor your quota usage to avoid exceeding limits.
- **Private Data**: Some metrics, like shares and detailed audience retention, may not be publicly available and require channel owner authentication. Google for Developers
- **Rate Limits**: Be mindful of rate limits when making API requests, especially when fetching comments or data for multiple videos. Medium

If you need assistance with setting up API access, writing scripts, or understanding specific metrics, feel free to ask!

Sources

what about the actual comment text, is it possible to crawl that as well?

Yes, you can retrieve the actual comment text from YouTube videos. The recommended approach is to use YouTube’s official Data API rather than directly scraping the website. Here’s how this works and what you need to consider:

Using the YouTube Data API for Comments

Comment Threads Endpoint:

The YouTube Data API v3 provides a `commentThreads` endpoint that allows you to retrieve the public comments on a video. This endpoint returns detailed information including:

- **Comment text:** The full text of each comment.
- **Author information:** Data like the commenter's display name, profile image URL, and channel ID.
- **Engagement details:** Such as the number of likes a comment has received.
- **Replies:** For comments with replies, you can retrieve nested responses.

Basic Code Example in Python:

Below is an illustrative example demonstrating how you might retrieve comments for a specific video:

```python
from googleapiclient.discovery import build

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

def get_video_comments(video_id, max_results=100):
    comments = []
    next_page_token = None

    while True:
        response = youtube.commentThreads().list(
            part='snippet',
            videoId=video_id,
            textFormat='plainText',
            maxResults=max_results,
            pageToken=next_page_token
        ).execute()

        for item in response.get('items', []):
            comment = item['snippet']['topLevelComment']['snippet']['textDisplay']
            comments.append(comment)

        next_page_token = response.get('nextPageToken')
        if not next_page_token:
            break

    return comments
```

```python

# Example usage:
video_id = 'VIDEO_ID_HERE'  # Replace with the target video's ID
comments = get_video_comments(video_id)
for index, comment in enumerate(comments, start=1):
    print(f"{index}: {comment}")
```

This script uses the `commentThreads` API to cycle through all available pages of comments. Note that you can adjust the `maxResults` parameter (up to a limit of 100) based on your needs.

Important Considerations

- **API Quota:** Each API call uses a portion of your quota. Retrieving a large number of comments, especially for very popular videos, may quickly consume your quota. It's important to handle quota limits and possibly cache the results if you plan to make frequent requests.
- **Rate Limiting:** The API enforces limits on the number of requests per second. Plan your retrieval process accordingly to avoid hitting those limits.
- **Public Data:** The API only returns public comments. If any comments are marked as private or moderated, they will not be accessible through the API.
- **Compliance with Terms:** Ensure that your usage of the API and any collected data complies with YouTube's Terms of Service and privacy policies. Scraping techniques not using the API are discouraged and may violate those terms.
- **Authentication:** While most comment data is public, accessing detailed analytics or comments on private videos may require OAuth 2.0 authentication with appropriate permissions if you’re accessing content on your own channel.

Conclusion

Retrieving actual comment text from YouTube videos is not only possible but also straightforward with the YouTube Data API. This approach allows you to systematically access the data you need while complying with YouTube's policies. If you have further questions about implementing this in your project or need details about handling specific data points, feel free to ask!