

Ultimate X/Tweeter Scraping Machine

Get Tweets

Method: GET

URL: https://api.twitterapi.io/twitter/tweet/advanced_search

Authentication: Generic Credential Type

Send Query Parameters: Enabled

Parameter 1:

Name: query

Value: openai

Parameter 2:

Name: queryType

Value: Top

Options - Pagination

Pagination Mode: Response Contains Next URL

Next URL: [https://api.twitterapi.io/twitter/tweet/advanced_search?cursor={{ \\$response.body.next_cursor }}](https://api.twitterapi.io/twitter/tweet/advanced_search?cursor={{ $response.body.next_cursor }})

Pagination Complete When: Other

Complete Expression:

```
{{ $response.body.has_next_page == false }}
```

Limit Pages Fetched: Enabled

Extract Tweet Content (Code)

```
// Initialize an empty array to collect all tweets
let allTweets = [];

// Loop through each input item
for (let item of $input.all()) {
  if (item.json.tweets && Array.isArray(item.json.tweets)) {
    allTweets = allTweets.concat(item.json.tweets);
  }
}

// Function to format the date in a more human-readable way
function formatDate(dateString) {
  if (!dateString) return '';

  try {
    const date = new Date(dateString);
    return date.toLocaleString('en-US', {
      year: 'numeric',
      month: 'long',
      day: 'numeric',
      hour: '2-digit',
    });
  } catch (error) {
    return '';
  }
}
```

```

        minute: '2-digit'
    });
} catch (error) {
    console.log("Error formatting date:", error);
    return dateString;
}
}

// Convert and structure the tweets properly
const formattedTweets = allTweets.map(tweet => ({
    json: {
        tweetId: tweet.id || "",
        url: tweet.url || "",
        content: tweet.text || "",
        likeCount: tweet.likeCount || 0,
        retweetCount: tweet.retweetCount || 0,
        replyCount: tweet.replyCount || 0,
        quoteCount: tweet.quoteCount || 0,
        viewCount: tweet.viewCount || 0,
        createdAt: formatDate(tweet.createdAt)
    }
})));

// Output all tweets as separate items
return formattedTweets;

```

GOOGLE SHEETS ← COPY HERE

Format for AI (Code)

```

const allItems = $input.all();

const combined = allItems.map(item => item.json.Content).join('\n\n---\n\n');

return [{ json: { combinedContent: combined } }];

```

AI Summary Generator

Prompt:

You are a skilled social media analyst specializing in summarizing high-impact Twitter/X content.

You will receive a string that includes several high-performing tweets combined into one text block. Each tweet is separated by a line of "---".

Tweets input is:

```
{{ $json.combinedContent }}
```

Your tasks:

1. Write a concise one-sentence overview summarizing today's main topic or pattern.
2. List the Top 3 Most Impactful Tweets. For each tweet, include:

- A brief summary
- Why it stands out
- A short snippet of the tweet (up to 280 characters)

3. Highlight one tweet that is surprising, controversial, or unique, if any, and explain why.
4. Suggest two intelligent next actions based on today's tweet content.

Email Formatting Instructions:

- Format the final output as a clean **HTML email report**.
- Use the following styling:
 - `

` for the main title
 - `

- ` - ` - Use ` - Use `

Email Sending Instructions:

Use the Gmail tool to send the final HTML report.

- **To**: [Your email address here or leave blank if dynamically defined]
- **Subject**: 🚀 Daily Twitter/X Highlights – {{ \$now.plus(8,hours).format('yyyy-MM-dd HH:mm') }}
- **Body**: the HTML report you generated

Make sure the email is **HTML formatted**, easy to scan, and includes only the structured summary — do not include the original tweet input or any internal explanations. Just output the final result and send it via Gmail.