CRAWLING TWITTER DATA

Konstantinos Semertzidis
WHAT TYPES OF INFORMATION CAN WE EXTRACT?

• Information about a user
• Friends & Followers of a user
• Tweets published by a user
• Search results on Twitter
• Places & Geo
TWITTER API

REST APIs

• provides Twitter functionality
• read / write / read DM (Tweet, Follow, DM, etc)
• To collect information a user must explicitly request it

Streaming APIs

• Once a request for information is made, the Streaming APIs provide a continuous stream of updates with no further input from the user. (Tweets in real-time)
TWITTER DEVELOPERS

Website: https://dev.twitter.com
TWITTER DEVELOPERS

- API resource documentation -- https://dev.twitter.com/docs
- Twitter libraries -- https://dev.twitter.com/docs/twitter-libraries
- Source examples -- https://dev.twitter.com/docs/open-source-examples
APIs EXAMPLES

- GET followers/ids
  https://api.twitter.com/1.1/followers/ids.json?cursor=-1&screen_name=sitestreams&count=5000

- GET friends/ids
  https://api.twitter.com/1.1/followers/ids.json?cursor=-1&screen_name=sitestreams&count=5000

- GET users/show
  https://api.twitter.com/1.1/users/show.json?screen_name=rsarver
GET followers/ids

• **screen_name / user_id**
The screen_name / user_id of the user for whom to return results for.

• **cursor**
Causes the list of connections to be broken into pages of no more than 5000 IDs at a time. If no cursor is provided, a value of -1 will be assumed, which is the first "page."

• **stringify_ids**
Many programming environments will not consume our Tweet ids due to their size. Provide this option to have ids returned as strings instead

• **count**
Specifies the number of IDs attempt retrieval of, up to a maximum of 5,000 per distinct request
GET followers/ids (Returned Result)

```
{
  "previous_cursor": 0,
  "ids": [143206502,
           143201767,
           777925],
  "previous_cursor_str": "0",
  "next_cursor": 0,
  "next_cursor_str": "0"
}
```
APIs LIMITS

In API version 1.1:

- Window: 15 minutes
- GET requests: 15 calls/15 minutes or 180 calls/15 minutes
- User limit: maximum requests per user
- App limit: maximum requests per application (including all users)
- Authentication is required
CREATE AN APPLICATION

Create an application

Application Details

Name:

Your application name. Must be unique to the cloud of a user and/or user-facing authentication context, e.g., "twitter".

Description:

Your application description, which will be shown in user-facing authentication contexts. Between 0 and 200 characters.

Website:

Your applications public-facing home page, where users can go to download media such as icons to find out more information about your application. This fully qualified URL is used in the context of user-facing authentication. More about using your application.

Callback URL:

Optional callback URL for your application. This URL, if used, is the location that the user is redirected to after permission is granted.

Developer Rules Of The Road

Rules of the Road

Twitter provides an open platform that supports the millions of people around the world who are sharing and discovering what's happening now. The users are the creators and consumption patterns that evolve and shape Twitter, and the effectiveness of using Twitter. At the same time, we aim to strike a balance between encouraging interesting development and protecting both "Twitter's" and "users" rights.

So, we've come up with a set of Developer Rules of the Road (Rules) that describe the policies and philosophy around what type of innovation is permitted with the Twitter API, and the rules of the road we've set for the developer community. To be clear, the rules apply to all applications, and are not limited to using the Twitter API, so please check back periodically for the most current version. Don't do anything prohibited by the Rules, but talk to us if you think we should update our Rules to allow development that further pushes the envelope!

If you will eventually need more than 5 million user tokens for your projects, you will need to talk to us directly about access to the Twitter API.

Yes, I agree

By clicking the "I agree" button, you acknowledge that you have read and understood this agreement and agree to be bound by its terms and conditions.

CAPTCHA

Please type the 4 word tokens.

(Checkbox)

Create your Twitter application!
APPLICATION DETAILS

GET followings

Organization
Information about the organization or company associated with your application. This information is optional.

Organization
None

Organization website
None

OAuth settings
Your application’s OAuth settings. Keep the "Consumer secret" a secret. This key should never be human-readable in your application.

Access level
Read-only

Consumer key
https://twitter.com/your_consumer_key

Consumer secret
akbYc7t[l:h;MvAdHDJt1cET5lsc1MKyXpF

Request token URL
https://api.twitter.com/oauth/request_token

Authorize URL
https://api.twitter.com/oauth/authorize

Access token URL
https://api.twitter.com/oauth/access_token

Callback URL
None

Sign in with Twitter
No
TWITTER LIBRARIES

- ActionScript/Flash
- C++
- Clojure
- Erlang
- Java
- JavaScript
- .NET
- Objective-C / Cocoa
- Perl
- PHP
- Python
- Ruby
- Scala
TWITTER 4J

• Is an unofficial Java library for the Twitter API
• Easy integration between a Java App and the Twitter service.
• 100% Pure Java – works on Java Platform version 5 or later
• Built-in Oauth support
• Compatible with Twitter API 1.1

http://www.twitter4j.org
HOW TO USE TWITTER 4J

• Download the latest stable version -- http://twitter4j.org/en/index.html#download

• Add twitter4j-core-version.jar to your application classpath

• JavaDoc -- http://twitter4j.org/javadoc/index.html

• Twitter4j.Twitter interface -- http://twitter4j.org/javadoc/twitter4j/Twitter.html
public class TwitterAccessToken {

    private static final String CONSUMER_KEY = "406Agt1HXwedQn";
    private static final String CONSUMER_SECRET = "RAshVRAs1R9pK5qBoCenVx2uSUXYXUj1j3t";

    public static void main(String[] args) throws Exception {
        Twitter twitter = new TwitterFactory().getInstance();
        twitter.setOAuthConsumer(CONSUMER_KEY, CONSUMER_SECRET);
        RequestToken requestToken = twitter.getOAuthRequestToken();
        AccessToken accessToken = null;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        while (null == accessToken) {
            System.out.println("Open the following URL and grant access to your account:");
            System.out.println(requestToken.getAuthorizationURL());
            System.out.print("Enter the PIN (if available) or just hit enter. [PIN]: ");
            String pin = br.readLine();
            try {
                if (pin.length() > 0) {
                    accessToken = twitter.getOAuthAccessToken(requestToken, pin);
                } else {
                    accessToken = twitter.getOAuthAccessToken();
                }
            } catch (TwitterException e) {
                if (401 == e.getStatusCode()) {
                    System.err.println("Unable to get the access token.");
                } else {
                    e.printStackTrace();
                }
            }
        }
        System.out.println("Access Token: " + accessToken.getToken());
        System.out.println("Access Token Secret: " + accessToken.getTokenSecret());
    }
}
public class TwitterAccessToken
{
    private static final String CONSUMER_KEY = "4v6qMjXmdqGn";
    private static final String CONSUMER_SECRET = "RaSkVnA=1NpR5q8cDeMvZuSbXyX5vJ1Jj72F";

    public static void main(String[] args) throws Exception {
        Twitter twitter = new TwitterFactory().getInstance();
        twitter.setOAuthConsumer(CONSUMER_KEY, CONSUMER_SECRET);
        RequestToken requestToken = twitter.getOAuthRequestToken();
        AccessToken accessToken = null;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        while (null == accessToken)
        {
            System.out.println("Open the following URL and grant access to your account:");
            System.out.println(requestToken.getAuthorizationURL());
            System.out.println("Enter the PIN (if available) or just hit enter.[PIN]:");
            String pin = br.readLine();
            try {
                if (pin.length() > 0)
                    accessToken = twitter.getOAuthAccessToken(requestToken, pin);
                else
                    accessToken = twitter.getOAuthAccessToken();
            } catch (TwitterException e) {
                if (401 == e.getStatusCode())
                    System.err.println("Unable to get the access token.");
                else
                    e.printStackTrace();
            }
        }
        System.out.println("Access Token: " + accessToken.getToken());
        System.out.println("Access Token Secret: " + accessToken.getTokenSecret());
    }
}
An application would like to connect to your account

The application Twitter-Connect Test by Twitter-Connect would like the ability to access and update your data on Twitter. Sign out if you want to connect to an account other than twitter-user10.

Allow Twitter-Connect access?

By clicking "Allow" you continue to operate under Twitter’s Terms of Service. In particular, some usage information will be shared back with Twitter. For more, see our Privacy Policy.

Deny  Allow
OAUTH PIN

You've successfully granted access to Coding the Tweet!

Simply return to Coding the Tweet and enter the following PIN to complete the process.

5191192
public class TwitterAccessToken
{
    private static final String CONSUMER_KEY = "4v6AqtlBxWdqu";
    private static final String CONSUMER_SECRET = "RAsvVRAwInP9o98cDmVx2uSUsXYXugJ18F";

    public static void main(String[] args) throws Exception {
        Twitter twitter = new TwitterFactory().getInstance();
        twitter.getOAuthConsumer(CONSUMER_KEY, CONSUMER_SECRET);
        RequestToken requestToken = twitter.getOAuthRequestToken();
        AccessToken accessToken = null;
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        while (null == accessToken)
        {
            System.out.println("Open the following URL and grant access to your account:");
            System.out.println(requestToken.getAuthorizationURL());
            System.out.println("Enter the PIN (if available) or just hit enter.[PIN]:");
            String pin = br.readLine();
            try {
                if (pin.length() > 0)
                    accessToken = twitter.getOAuthAccessToken(requestToken, pin);
                else
                    accessToken = twitter.getOAuthAccessToken();
            } catch (TwitterException e) {
                if (401 == e.getStatusCode())
                    System.err.println("Unable to get the access token.");
                else
                    e.printStackTrace();
            }
        }
        System.out.println("Access Token: " + accessToken.getToken());
        System.out.println("Access Token Secret: " + accessToken.getTokenSecret());
    }
}
public class GET_followersIDs {
    private final static String CONSUMER_KEY = "T9KqEzPGuPzd01Cp";
    private final static String CONSUMER_KEY_SECRET = "80BY1JDsR7RpJQhvV2WsvWbr1gIO4Qf0Gm";

    public void start() throws TwitterException {
        Twitter twitter = new TwitterFactory().getInstance();
        twitter.setOAuthConsumer(CONSUMER_KEY, CONSUMER_KEY_SECRET);

        String accessToken = "1275957220-QayyT44zIFX0DmqW73Ny2aM4gTTPmNeUSe6hx";
        String accessTokenSecret = "raqZLgSlhUK138ck7lmG3AFAEYy5RsttVXeWE";
        AccessToken oauthAccessToken = new AccessToken(accessToken, accessTokenSecret);
        twitter.setOAuthAccessToken(oauthAccessToken);

        try {
            long userId = 12563526, cursor = -1;
            Ids ids = twitter.getFollowersIDs(userId, cursor);

            for (long id : ids.getIds())
                System.out.println(id);
        }
        catch (TwitterException te) {
            te.printStackTrace();
            System.out.println("Failed to get followers' ids: " + te.getMessage());
        }
    }

    public static void main(String[] args) throws Exception {
        new GET_followersIDs().start();
    }
}
public class GET_followersIDs {
    private final static String CONSUMER_KEY = "T9XqSzPGu5w5d01Cp";
    private final static String CONSUMER_KEY_SECRET = "80ByIJDxK7RpJqhvZVsw6br1qIO4qhGGM";

    public void start() throws TwitterException {
        Twitter twitter = new TwitterFactory().getInstance();
        twitter.setOAuthConsumer(CONSUMER_KEY, CONSUMER_SECRET);

        String accessToken = "1275995220-QsyyT44MzIpOfnqM6ZNzaRkX4gTf9MeUGe6kh";
        String accessTokenSecret = "rzoQZLgSbHEEK150k7Lm6SSIAF9EFy5RzAVxXeWE";
        AccessToken oauthAccessToken = new AccessToken(accessToken, accessTokenSecret);
        twitter.setOAuthAccessToken(oauthAccessToken);

        try {
            long userId = 12563526, cursor = -1;
            IDs ids = twitter.getFollowersIDs(userId, cursor);
            for (long id : ids.getIDs())
                System.out.println(id);
        }
        catch (TwitterException te) {
            te.printStackTrace();
            System.out.println("Failed to get followers' ids: " + te.getMessage());
        }
    }

    public static void main(String[] args) throws Exception {
        new GET_followersIDs().start();
    }
}
THANK YOU!

QUESTIONS ?