

Crawling Twitter Data

Konstantinos Semertzidis

ksemer@cs.uoi.gr



What types of information can we extract?

- Information about a user
- User's Followers or Friends
- Tweets published by a user
- Search results on Twitter
- Places & Geo

What types of information can we extract?

- Information about a user
- User's Followers or Friends
- Tweets published by a user
- Search results on Twitter
- Places & Geo



HOW?

Twitter API

REST APIs

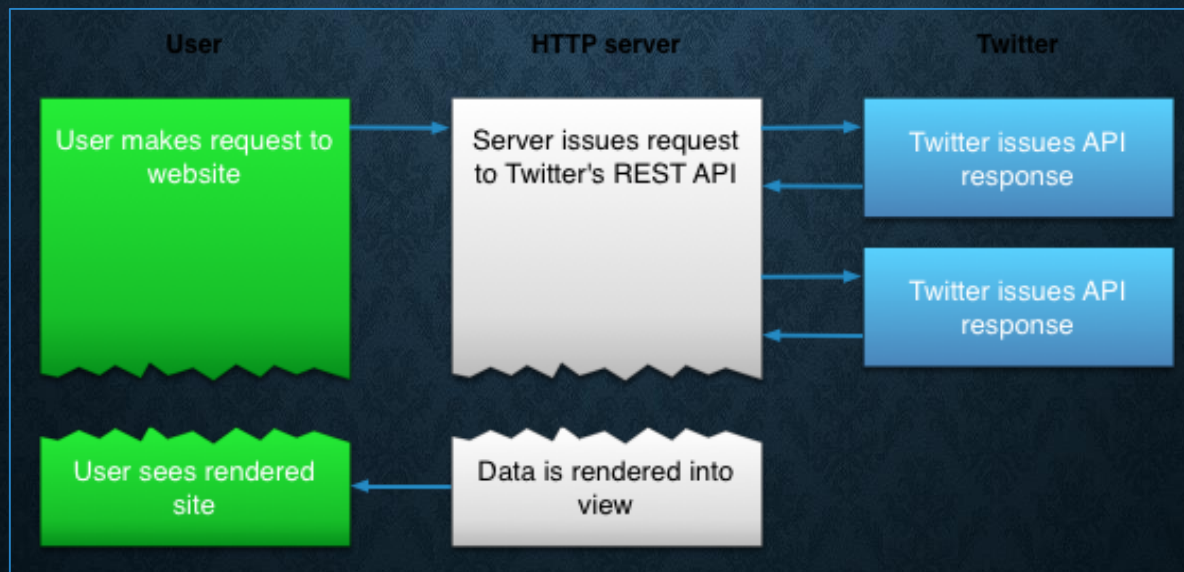
- The REST APIs provides programmatic access to read and write Twitter data

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)

Search API

- The Twitter Search API searches against a sampling of recent Tweets published in **the past 7 days**.



Twitter developers



Website: <https://dev.twitter.com/>

API resource documentation: <https://dev.twitter.com/docs>

Twitter libraries: <https://dev.twitter.com/docs/twitter-libraries>

Rest API Methods (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 friends/ids (Example Result)

```
1.{
2."previous_cursor": 0,
3."ids": [
4.143206502,
5.143201767,
6.777925
7.],
8."previous_cursor_str": "0",
9."next_cursor": 0,
10."next_cursor_str": "0"
11.}
```

REST & SEARCH APIs Limits

Resource Information	
Rate Limited?	Yes
Requests per rate limit window	15/user 15/app
Authentication	Required
Response Formats	json
HTTP Methods	GET
Resource family	followers
Response Object	Users
API Version	v1.1

GET followers API Limits:

- Window: 15 minutes
- Requests per rate limit window:
 - 15 calls/user and 15 calls/app
- Authentication is required

Check: <https://dev.twitter.com/rest/public/rate-limiting>

STREAMING APIs Limits

- No rate limit
- Streaming API allows to be streamed up to 1% tweets of the total volume

<https://dev.twitter.com/streaming/overview>

Libraries To Integrate AN Application With The Twitter Service

Available libraries:

- ActionScript/Flash, C++, Clojure, Erlang, Java, Javascript, .NET,
- Objective-C / Cocoa, Perl, PHP, Python, Ruby, Scala

<https://dev.twitter.com/docs/twitter-libraries>

Twitter4j

- 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
- Website: <http://www.twitter4j.org>

How To Use Twitter4j

- 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/en/javadoc.html>

Create An Application

twitter developers Search API Health Blog Discussions Documentation semerkon

Create an application

Application Details

Name: *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description: *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

WebSite: *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL:

Where should we return after successfully authenticating? For [@Anywhere](#) applications, only the domain specified in the callback will be used. [OAuth 1.0a](#) applications should explicitly specify their `oauth_callback` URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Developer Rules Of The Road

Last Update - 1st of June 2011

Rules of the Road

Twitter maintains an open platform that supports the millions of people around the world who are sharing and discovering what's happening now. We want to empower our ecosystem partners to build valuable businesses around the information flowing through 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 content and information shared on Twitter.

The Rules will evolve along with our ecosystem as developers continue to innovate and find new, creative ways to use the Twitter API, so please check back periodically to see the most current version. Don't do anything prohibited by the Rules, but talk to us if you think we should make a change or give you an exception.

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 understand this agreement and agree to be bound by its terms and conditions.

CAPTCHA

Please type the two words below.

CAPTCHA™

stop spam, need tweets.

Create your Twitter application

<https://apps.twitter.com/>


Application Details

Developers API Health Blog Discussions Documentation Search

Home → My applications

GET followings

Details Settings OAuth tool @Anywhere domains Reset keys Delete

 Get followings of a user
[@ksemer](http://cs.uoi.gr/~ksemer)

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 About the application permission model
Consumer key	hTk84TLwVImTbey
Consumer secret	akkyL85jLBFAVxBaSTL6oTD1ctLAVXxgdF
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

GET followers/ids Code Sample

```
public class GET_followersIDs
{
    private final static String CONSUMER_KEY = "T9XqEzPGuPxd0lCp";
    private final static String CONSUMER_KEY_SECRET = "80BYIJDsR7RpJQhvzVswWbr1gIO4QhfGGm";

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

        String accessToken = "1275997220-QsyYT44MzIPXODnqWQ7NZaHX4gTTPNeUGe6hk";
        String accessTokenSecret = "rmeQZLgSbhEK158ok7LmSS3AFA9KYy5RztaVXeWE";
        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(); }
}
```

Streaming Code Example (1)

```
import twitter4j.FilterQuery;
import twitter4j.Status;
import twitter4j.StatusDeletionNotice;
import twitter4j.StatusListener;
import twitter4j.TwitterStream;
import twitter4j.TwitterStreamFactory;
import twitter4j.User;
import twitter4j.conf.ConfigurationBuilder;

public class SimpleStream {
    public static void main(String[] args) {
        ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true);
        cb.setOAuthConsumerKey("*****");
        cb.setOAuthConsumerSecret("*****");
        cb.setOAuthAccessToken("*****");
        cb.setOAuthAccessTokenSecret("*****");

        TwitterStream twitterStream = new
TwitterStreamFactory(cb.build()).getInstance();

        StatusListener listener = new StatusListener() {
```


Streaming Code Example (2)

```
        @Override
        public void onStatus(Status status) {
            User user = status.getUser();

            // gets Username
            String username =
status.getUser().getScreenName();
            System.out.println(username);
            String profileLocation = user.getLocation();
            System.out.println(profileLocation);
            long tweetId = status.getId();
            System.out.println(tweetId);
            String content = status.getText();
            System.out.println(content + "\n");

        }

        @Override
        public void onTrackLimitationNotice(int arg0) {
            // TODO Auto-generated method stub

        }

    };
    FilterQuery fq = new FilterQuery();

    String keywords[] = {"ireland"};

    fq.track(keywords);

    twitterStream.addListener(listener);
    twitterStream.filter(fq);

}
}
```

OAUTH Code Example

```
public class TwitterAccessToken
{
    private static final String CONSUMER_KEY = "4U6AQtlHXwdqN";
    private static final String CONSUMER_SECRET = "RashVRAsINpR5qBcDenVx2uSUSXYXUjJiBF";

    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());
    }
}
```


Authorization URL



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?

Deny

Allow

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](#).

Twitter takes your privacy very seriously.

Only click "Allow" for applications you trust. Allowing this application to connect to your account may give Twitter-Connect access to your Direct Messages (DMs), or the ability to Tweet on your behalf.

You may revoke access to this application at any time by visiting your Settings page.

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

Thank You!