

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)

HOW TO USE THE TWITTER API

TWITTER DEVELOPERS

The screenshot shows the Twitter Developers website. At the top, there is a navigation bar with the Twitter logo, 'Developers', and links for 'API Health', 'Blog', 'Discussions', and 'Documentation'. A search bar is also present. The main content area features a large banner titled 'More downloads for your app with Twitter Cards'. The banner includes a diagram showing a cycle of three mobile devices (a smartphone, a tablet, and a laptop) with 'INSTALL' buttons, connected by blue arrows. Below the banner, there are five icons representing different features: Twitter Cards, Embedded Timelines, Embedded Tweets, Tweet Button, and Follow Button. At the bottom, there are sections for 'Recent posts from Twitter Developer Blog' and 'Create applications that integrate Twitter'.

Developers API Health Blog Discussions Documentation Search

More downloads for your app with Twitter Cards

Twitter Cards offer a fast and easy way to grow your user base for mobile apps. Simply add some new markup to your pages: when users tweet links to your domain, Cards will let other users viewing those Tweets to download and launch your app across a number of mobile platforms.

[Learn More](#)

Twitter Cards Embedded Timelines Embedded Tweets Tweet Button Follow Button

Recent posts from Twitter Developer Blog

- Nov 12 [Introducing custom timelines: create timelines of Tweets for everyone](#)
- Sep [Streaming SSL certificate updates](#)

Create applications that integrate Twitter

[Get started with the platform](#)
Explore the documentation and manage your apps

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)

```
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. }
```

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

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

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.



 © 2009 Google. All Rights Reserved.

Create your Twitter application


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
<http://cs.uoi.gr/~ksemere/>

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	akkyL85jLBFAVxBaSTL6cTD1ct1AVXxgdF
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>

OAUTH CODE SAMPLE

```
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());
    }
}
```

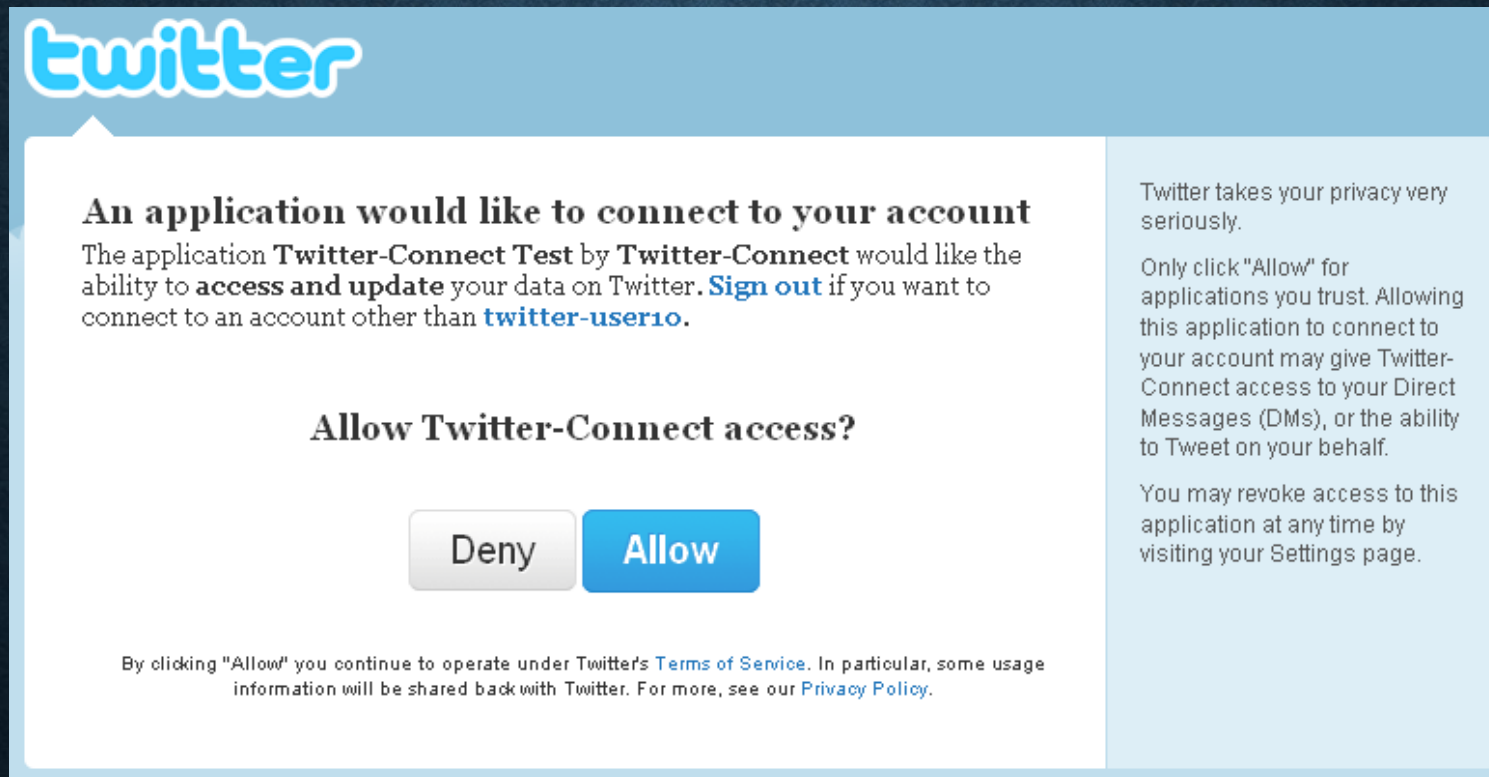

OAUTH CODE SAMPLE

```
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



The image shows a screenshot of a Twitter authorization dialog box. At the top left is the Twitter logo. The main content area has a white background with a light blue border. It contains a heading, a paragraph of text, a question, two buttons, and a disclaimer. On the right side, there is a light blue sidebar with three paragraphs of text.

twitter

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](#).

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



OAUTH CODE SAMPLE

```
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());
    }
}
```

GET followers/ids CODE SAMPLE

```
public class GET_followersIDs
{
    private final static String CONSUMER_KEY = "T9XqEzPGuPxd0lCp";
    private final static String CONSUMER_KEY_SECRET = "80BYIJDsR7RpJQhvZVswWbrlgIO4QhfGGm";

    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 oathAccessToken = new AccessToken(accessToken, accessTokenSecret);
        twitter.setOAuthAccessToken(oathAccessToken);

        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(); }
}
```

GET followers/ids CODE SAMPLE

```
public class GET_followersIDs
{
    private final static String CONSUMER_KEY = "T9XqEzPGuPxd0lCp";
    private final static String CONSUMER_KEY_SECRET = "80BYIJDsR7RpJQhvZVswWbrlgIO4QhfGGm";

    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 oathAccessToken = new AccessToken(accessToken, accessTokenSecret);
        twitter.setOAuthAccessToken(oathAccessToken);

        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 ?

