



# Social Media Apps Programming

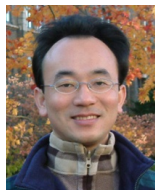
## Twitter API

1051SMAP13

TLMXM1A (8648) (M2143) (Fall 2016)

(MIS MBA) (2 Credits, Elective) [Full English Course]

Wed 8,9 (15:10-17:00) B310



Min-Yuh Day, Ph.D.

Assistant Professor

Department of Information Management

Tamkang University

<http://mail.tku.edu.tw/myday>

2016-12-28



# Course Schedule (1/3)

Week	Date	Subject/Topics
1	2016/09/14	Course Orientation and Introduction to Social Media and Mobile Apps Programming
2	2016/09/21	Introduction to Android / iOS Apps Programming
3	2016/09/28	Developing Android Native Apps with Java (Eclipse) (MIT App Inventor)
4	2016/10/05	Developing iPhone / iPad Native Apps with Swift (XCode)
5	2016/10/12	Mobile Apps using HTML5/CSS3/JavaScript
6	2016/10/19	jQuery Mobile

# Course Schedule (2/3)

Week	Date	Subject/Topics
7	2016/10/26	Create Hybrid Apps with PhoneGap
8	2016/11/02	jQuery Mobile/PhoneGap
9	2016/11/09	jQuery Mobile/PhoneGap
10	2016/11/16	Midterm Exam Week (Midterm Project Report)
11	2016/11/23	Case Study on Social Media Apps Programming and Marketing in Google Play and App Store
12	2016/11/30	Invited Talk [B302b]: Challenges in Natural Language Processing: Question Answering and Dialog System Invited Speaker: Prof. Yoshinobu Kano, Associate Professor, Faculty of Informatics, Shizuoka University, Japan

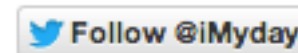
# Course Schedule (3/3)

Week	Date	Subject/Topics
13	2016/12/07	Google Cloud Platform
14	2016/12/14	Google App Engine and Google Map API
15	2016/12/21	Facebook API (Facebook JavaScript SDK) (Integrate Facebook with iOS/Android Apps)
16	2016/12/28	Twitter API
17	2017/01/04	Final Project Presentation
18	2017/01/11	Final Exam Week (Final Project Presentation)



# Outline

- **Twitter Developers**
  - Twitter Platform Objects
- **Twitter for Websites**
- **Twitter Search API**
- **Twitter REST API**
- **Twitter Streaming API**



# Twitter



## Starbucks

TOP LATEST NEWS PHOTOS VIDEOS MORE

Related searches

- starbucks **singapore**
- starbucks **christmas**

Who to follow · Refresh · View all

- Vine Help** @vinehelp
- EUMLab** @EUMLab
- HarvardFamilyResProj** @HF...

Find friends

Trends · Change

- #Taipei**
- Christmas**  
3.21M Tweets
- #China**  
@TW\_nextmedia is Tweeting about this
- #TuesdayMotivation**

Top news

[View all](#)



Starbucks is welcoming 2017 with three elaborate new drinks

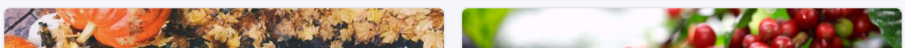
Look good, feel good.

**Mashable**

Dec 27, 2016

People

[View all](#)



# What can your business do...in just 140 characters?





What can your business do...in just 140 characters?



Twitter · 76 videos

 **Subscribe** 46,631

82,560

 695  30

<http://www.youtube.com/watch?v=BGirUZq1WtQ>



TWEETS **85.1K** FOLLOWING **99.3K** FOLLOWERS **11.8M** LIKES **10.2K** LISTS **1** MOMENTS **4**



Following

## Starbucks Coffee

@Starbucks

Inspiring and nurturing the human spirit— one person, one cup, one neighborhood at a time.

Seattle, WA

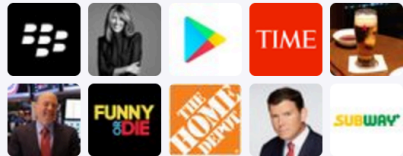
[starbucks.com](http://starbucks.com)

Joined November 2006

Tweet to

Message

36 Followers you know



1,442 Photos and videos

Featured Tweets Tweets & replies Media

### Latest Tweets



Starbucks Coffee @Starbucks · 21h

An espresso favorite—all dressed up just for you. The new [#TuxedoMocha](#). Here 'till the new year. 🎉



Who to follow · Refresh · View all



Vine Help @vinehelp

Follow



EUMLab @EUMLab

Follow



HarvardFamilyResProj @H...

Follow

Find friends

Trends · Change

[#Taipei](#)

[Christmas](#)  
3.36M Tweets

[#China](#)  
@TW\_nextmedia is Tweeting about this

[#TuesdayMotivation](#)



# Facebook

← → ↻ <https://www.facebook.com/Starbucks>

f Starbucks



Min-Yuh

Home



Starbucks ✓

@Starbucks

Home

Posts

Videos

Photos

Locations

About

Likes

Pinterest

Open Jobs

International

Create a Page



👍 Liked ▾

📡 Following ▾

➦ Share

⋮ More ▾

Send Message

## Featured For You

Get in touch with Starbucks



Save Starbucks

Food & Beverage Company

🔍 Search for posts on this Page



Invite friends to like this Page

Inspiring and nurturing the human spirit -- one person, one cup, and one neighborhood at a time <http://www.youtube.com/starbucks>



36,541,238 Likes

I-hsien Ting and 103 other friends like this



<https://www.facebook.com/Starbucks>

# Twitter vs. Facebook



Twitter, Inc. [US] <https://twitter.com/Starbucks/tweets>

Home Notifications Messages Search

**Starbucks Coffee** @Starbucks

85.1K TWEETS 99.3K FOLLOWING 11.8M FOLLOWERS 10.2K LIKES 1 LISTS 4 MOMENTS

Featured Tweets Tweets & replies Media

**Starbucks Coffee** @Starbucks · 21h  
An espresso favorite—all dressed up just for you. The new #TuxedoMocha. Here 'till the new year. 🎉

Seattle, WA  
[starbucks.com](http://starbucks.com)

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



Starbucks

Liked Following Share More

People

227,083 People Talking About This

32,015,776 People Checked In Here

36,541,240 Total Page Likes  
▲0.1% from last week

30,157 New Page Likes  
▲5.2%

— This week  
— Last week

12/21 12/22 12/23 12/24 12/25 12/26 12/27

34,966,924 Total Page Follows  
0% from last week

Home Posts Videos Photos Locations About Likes Pinterest Open Jobs International

Create a Page

[https://www.facebook.com/pg/Starbucks/likes/?ref=page\\_internal](https://www.facebook.com/pg/Starbucks/likes/?ref=page_internal)

# Twitter Developers

← → ↻ <https://dev.twitter.com> ☆

 Developers

Products

Documentation

Community

Build

My apps

Join

## Welcome to the Twitter Platform

Dive into documentation

Learn about our products

Welcome to the Twitter Platform

Twitter Developer Communities

#HelloWorld 2016

Mobile app playbook

Customer service

Learn more about Fabric

## Explore our products



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

# Twitter Developers Documentation

Documentation | Twitter D x

← → ↻ <https://dev.twitter.com/docs> ☆ ☰


Developers API Health Blog Discussions **Documentation** Search Sign in


Home


## Documentation


**Getting Started** with the Twitter Platform is easy. Jump right into the [API resource documentation](#), explore the [developer console](#), or [manage your apps](#). Have a question? Read the [FAQ](#). Don't know what to build? Check out our [Case Studies](#).

**A field guide to Twitter Platform objects**

 **Tweets**


 **Users**


 **Entities**

 **Places**

Explore the Twitter platform's variety of flora and fauna with this field guide to the most frequently observed wild objects.

### Twitter cards

 **Richard Henry** @richardhenry 4m  
That's a whole lot of people... [nyti.ms/yzg6Wq](https://nyti.ms/yzg6Wq)  
[Hide summary](#) [Reply](#) [Retweet](#) [Favorite](#)

**Parade of Fans for Houston's Funeral**  
By Sarah Maslin Nir @SarahMaslinNir   
NEWARK — The guest list and the parade of limousines

### Embedded Timelines

[Tweets from @twitter/more-twitter-accounts](#)

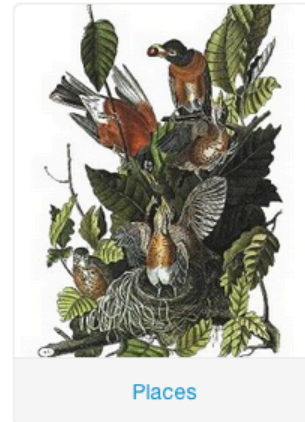
[Embeddable timelines](#) make it possible to display public timelines on your website with a single line of code.

### Embedded Tweets



# A field guide to Twitter Platform objects

The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/platform-objects>. The page header includes the Twitter logo, the word "Developers", and navigation links for "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible. The main content area features the title "A field guide to Twitter Platform objects" and an introductory paragraph: "Like any ecosystem, the Twitter platform has a variety of flora and fauna. Use this field guide to better understand the most frequently observed wild objects." Below this text are four illustrated cards, each with a bird illustration and a label: "Tweets" (two blue birds), "Users" (a brown bird on a branch), "Entities" (a yellow bird on a yellow flower), and "Places" (a bird on a nest with other birds). At the bottom of the page, there is a "Follow @twitterapi" button and a footer with links for "API Terms", "API Status", "Blog", "Discussions", "Documentation", and a note that the site is supported by Acquia.



# Tweets

Tweets are the basic atomic building block of all things Twitter.



# Users

Users can be anyone or anything.



# Entities

Entities provide metadata and additional contextual information about content posted on Twitter.



# Places

Places are specific, named locations with corresponding geo coordinates.



# Tweets

Tweets | Twitter Developer x

← → ↻ <https://dev.twitter.com/docs/platform-objects/tweets> ☆ ☰

Developers API Health Blog Discussions Documentation Search Sign in


Home → Documentation → Platform Objects Tweet

## Tweets

[View](#) [What links here](#)

Updated on Tue, 2013-08-13 16:29 API version 1 API version 1.1 **Natural habitat**

Tweets are the basic atomic building block of all things Twitter. [Users tweet](#) Tweets, also known more generically as "status updates." Tweets can be [embedded](#), [replied to](#), [favorited](#), [unfavorited](#) and [deleted](#).



**Brian Sutorius**  
@bsuto [Follow](#)

The "http:/" at the beginning of URLs is a command to the browser. It stands for "head to this place:" followed by two laser-gun noises.

5:29 AM - 22 Feb 2012

4,300 RETWEETS 1,836 FAVORITES

← ↻ ⭐


**Field Guide**

Consumers of Tweets should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing. Please note that Tweets found in Search results vary somewhat in structure from this document.

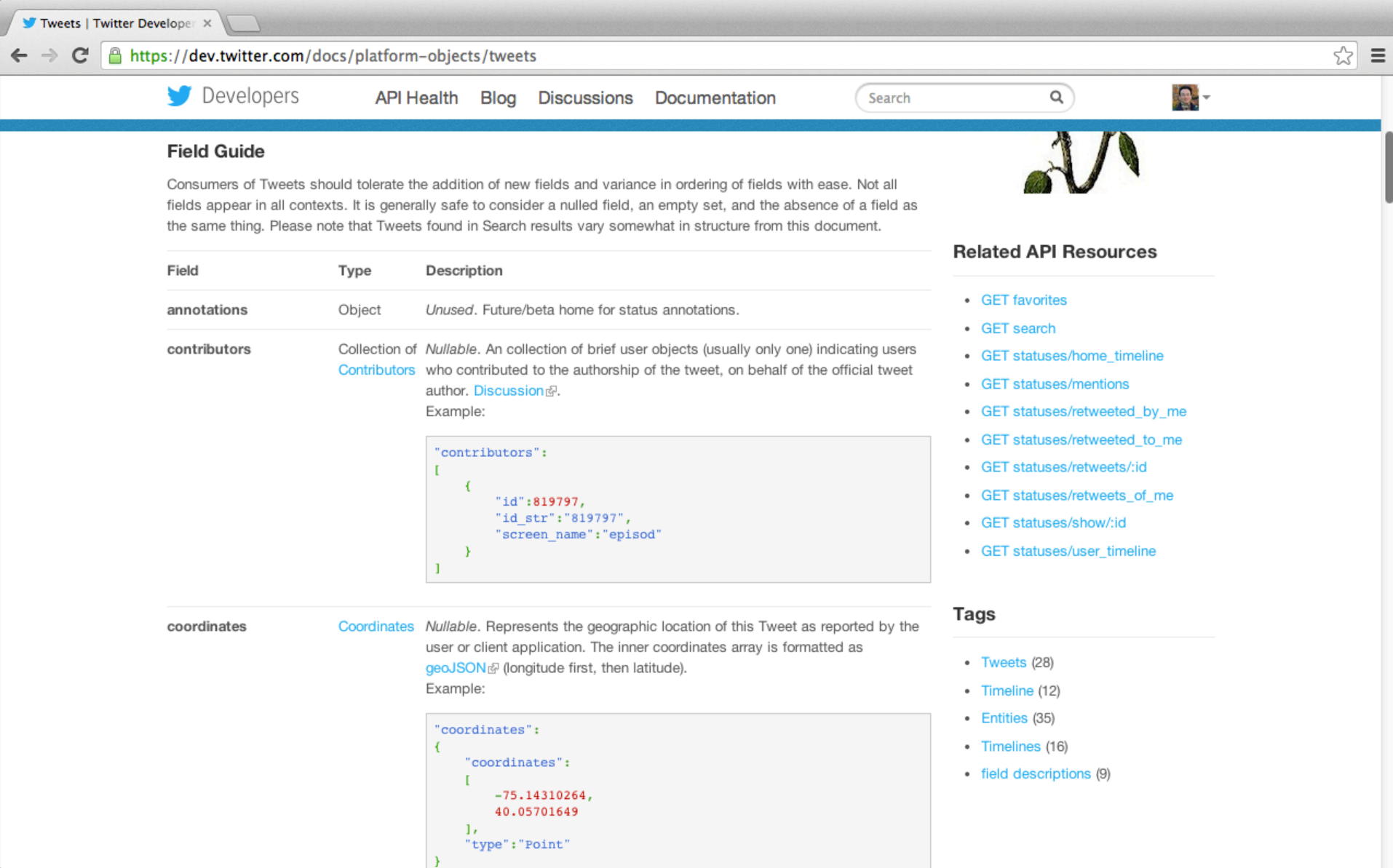
Field	Type	Description
annotations	Object	Unused. Future/beta home for status annotations.

**Related API Resources**

- [GET favorites](#)
- [GET search](#)



# Tweets



The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/platform-objects/tweets>. The page title is "Tweets | Twitter Developer". The navigation bar includes "Developers", "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible.

## Field Guide

Consumers of Tweets should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing. Please note that Tweets found in Search results vary somewhat in structure from this document.

Field	Type	Description
<b>annotations</b>	Object	<i>Unused.</i> Future/beta home for status annotations.
<b>contributors</b>	Collection of <a href="#">Contributors</a>	<i>Nullable.</i> An collection of brief user objects (usually only one) indicating users who contributed to the authorship of the tweet, on behalf of the official tweet author. <a href="#">Discussion</a> . Example: <pre>"contributors": [   {     "id":819797,     "id_str":"819797",     "screen_name":"episod"   } ]</pre>
<b>coordinates</b>	<a href="#">Coordinates</a>	<i>Nullable.</i> Represents the geographic location of this Tweet as reported by the user or client application. The inner coordinates array is formatted as <a href="#">geoJSON</a> (longitude first, then latitude). Example: <pre>"coordinates": {   "coordinates":   [     -75.14310264,     40.05701649   ],   "type":"Point" }</pre>

## Related API Resources

- [GET favorites](#)
- [GET search](#)
- [GET statuses/home\\_timeline](#)
- [GET statuses/mentions](#)
- [GET statuses/retweeted\\_by\\_me](#)
- [GET statuses/retweeted\\_to\\_me](#)
- [GET statuses/retweets/:id](#)
- [GET statuses/retweets\\_of\\_me](#)
- [GET statuses/show/:id](#)
- [GET statuses/user\\_timeline](#)

## Tags

- [Tweets](#) (28)
- [Timeline](#) (12)
- [Entities](#) (35)
- [Timelines](#) (16)
- [field descriptions](#) (9)



# Tweets

Tweets | Twitter Developer x

https://dev.twitter.com/docs/platform-objects/tweets

Developers API Health Blog Discussions Documentation Search

**user** [Users](#) The user who posted this Tweet. Perspectival attributes embedded within this object are unreliable. See [Why are embedded objects stale or inaccurate?](#). Example:

```
"user":{"statuses_count":3080, "favourites_count":22, "protected":false, "profile_text_color":"437792", "profile_image_url": "...", "name":"Twitter API", "profile_sidebar_fill_color":"a9d9f1", "listed_count":9252, "following":true, "profile_background_tile":false, "utc_offset":-28800, "description":"The Real Twitter API. I tweet about API changes, service issues and happily answer questions about Twitter and our API. Don't get an answer? It's on my website.", "location":"San Francisco, CA", "contributors_enabled":true, "verified":true, "profile_link_color":"0094C2", "followers_count":665829, "url":"http://dev.twitter.com", "default_profile":false, "profile_sidebar_border_color":"0094C2", "screen_name":"twitterapi", "default_profile_image":false, "notifications":false, "display_url":null, "show_all_inline_media":false, "geo_enabled":true, "profile_use_background_image":true, "friends_count":32, "id_str":"6253282", "entities":{"hashtags":[], "urls":[], "user_mentions":[]}, "expanded_url":null, "is_translator":false, "lang":"en", "time_zone":"Pacific Time (US & Canada)", "created_at":"Wed May 23 06:01:13 +0000 2007", "profile_background_color":"e8f2f7", "id":"6253282", "follow_request_sent":false, "profile_background_image_url_https": "...", "profile_background_image_url": "...", "profile_image_url_https": "..."}}
```

**withheld\_copyright** Boolean When present and set to "true", it indicates that this piece of content has been withheld due to a [DMCA complaint](#). Example:

```
"withheld_copyright": true
```

**withheld\_in\_countries** Array of Strings When present, indicates a list of uppercase [two-letter country codes](#) this content is withheld from. See [New Withheld Content Fields in API Responses](#)



# Users

Users | Twitter Developers x

https://dev.twitter.com/docs/platform-objects/users

Developers API Health Blog Discussions Documentation Search

Home → Documentation → Platform Objects Tweet

## Users


View What links here

Updated on Mon, 2013-05-20 07:28 API version 1 API version 1.1

### Natural habitat

Users can be anyone or anything. They [tweet](#), [follow](#), [create lists](#), have a [home\\_timeline](#), can be [mentioned](#), and can be [looked up](#) in bulk.

Users can be found [tweeting](#), [following](#), and [favoriting](#) on Twitter.



**twitterapi** Twitter API ✓  
*The Real Twitter API. I tweet about API changes, service issues and happily answer questions about Twitter and our API. Don't get an answer? It's on my website.*  
<http://dev.twitter.com>

Followers **988,982** Following **33**


GET /jobs

(JSON)

### Field Guide

Consumers of Users should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing.

Field	Type	Description
<code>contributors_enabled</code>	Boolean	Indicates that the user has an account with "contributor mode" enabled, allowing for Tweets issued by the user to be co-authored by another account. Early users



### Related API Resources

- [GET account/verify\\_credentials](#)
- [GET blocks/blocking](#)

# Users

Users | Twitter Developers x

https://dev.twitter.com/docs/platform-objects/users

Developers API Health Blog Discussions Documentation Search

status

Tweets

*Nullable.* If possible, the user's most recent tweet or retweet. In some circumstances, this data cannot be provided and this field will be omitted, null, or empty. Perspectival attributes within tweets embedded within users cannot always be relied upon. See [Why are embedded objects stale or inaccurate?](#)

Example:

```
"status": {
  "coordinates": null,
  "favorited": false,
  "truncated": false,
  "created_at": "Tue Apr 17 16:38:18 +0000 2012",
  "id_str": "192290904646754304",
  "entities": {
    "urls": [

  ],
    "hashtags": [

  ],
    "user_mentions": [
      {
        "name": "Micah McVicker",
        "id_str": "166661446",
        "id": 166661446,
        "indices": [
          0,
          14
        ],
        "screen_name": "MicahMcVicker"
      }
    ]
  },
  "in_reply_to_user_id_str": "166661446",
  "contributors": null,
  "text": "@MicahMcVicker make sure you're using
include_rts=true and no other filters, then walking your
timeline by since_id and max_id",
  "retweet_count": 0,
  "in_reply_to_status_id_str": "192290470427246594",
  "id": 192290904646754304,
  "geo": null
```

# Entities

Twitter Developers | Twitter Developer x

https://dev.twitter.com/docs/platform-objects/entities

Developers API Health Blog Discussions Documentation Search

Home → Documentation → Platform Objects Tweet

## Entities

View What links here

Updated on Mon, 2013-12-16 15:46 API version 1 API version 1.1

Entities provide metadata and additional contextual information about content posted on Twitter. Entities are never divorced from the content they describe. In API v1.1, entities will be returned wherever Tweets are found in the API. Entities are instrumental in resolving URLs.

Read [Entities in Twitter Objects](#) for a more comprehensive guide to how entities are used throughout Twitter objects.


### Field Guide

Consumers of Entities should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing.

Field	Type	Description
hashtags	Array of Object	Represents hashtags which have been parsed out of the Tweet text. Example: <pre>"hashtags": [{"indices": [32, 36], "text": "lol"}]</pre>
media	Array of Object	Represents media elements uploaded with the Tweet. Example: <pre>"media": [{"type": "photo", "sizes": {"thumb": {"h": 150, "resize": "crop", "w": 150}, "large": {"h": 238, "resize": "fit", "w": 226}, "medium": {"h": 238, "resize": "fit", "w": 226}, "small": {"h": 238, "resize": "fit", "w": 226}}</pre>

### Natural habitat

Entities are most often found within Tweets.



### Tags

- Tweets (28)
- Entities (35)

# Entities

Entities | Twitter Developer x


https://dev.twitter.com/docs/platform-objects/entities

Developers API Health Blog Discussions Documentation Search

## Field Guide

Consumers of Entities should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing.

Field	Type	Description
<b>hashtags</b>	Array of <a href="#">Object</a>	Represents hashtags which have been parsed out of the Tweet text. Example: <pre>"hashtags": [{"indices": [32,36], "text": "lol"}]</pre>
<b>media</b>	Array of <a href="#">Object</a>	Represents media elements uploaded with the Tweet. Example: <pre>"media": [{"type": "photo", "sizes": {"thumb": {"h": 150, "resize": "crop", "w": 150}, "large": {"h": 238, "resize": "fit", "w": 226}, "medium": {"h": 238, "resize": "fit", "w": 226}, "small": {"h": 238, "resize": "fit", "w": 226}}, "indices": [15,35], "url": "http://t.co/rJC5Pxsu", "media_url": "http://p.twimg.com/AZVLmp-CIAAbkyy.jpg", "display_url": "pic.twitter.com/rJC5Pxsu", "id": "114080493040967680", "id_str": "114080493040967680", "expanded_url": "http://twitter.com/yunorno/status/114080493036773378/photo/1", "media_url_https": "https://p.twimg.com/AZVLmp-CIAAbkyy.jpg"}]</pre>
<b>urls</b>	Array of <a href="#">Object</a>	Represents URLs included in the text of a Tweet or within textual fields of a <a href="#">user object</a> . Tweet Example: <pre>"urls": [{"indices": [32,52], "url": "http://t.co/IowBrTZR", "display_url": "youtube.com/watch?v=oHg5SJ\au2026", "expanded_url": "http://www.youtube.com/watch?v=oHg5SJYRHA0"}]</pre> User Example:




## Tags

- [Tweets](#) (28)
- [Entities](#) (35)
- [Mentions](#) (20)
- [t.co](#) (43)
- [field descriptions](#) (9)
- [hashtags](#) (4)
- [finding media](#) (4)
- [finding links](#) (2)

# Places

Places | Twitter Developers x

← → ↻ <https://dev.twitter.com/docs/platform-objects/places> ☆ ☰

Developers API Health Blog Discussions Documentation Search 

Home → Documentation → Platform Objects 🐦 Tweet


## Places

[View](#) [What links here](#)

Updated on Mon, 2013-10-28 04:57 API version 1 API version 1.1

### Natural habitat

Places can be found throughout the natural universe, but typically only appear attached to [Tweets](#) on Twitter.



### Field Guide

Consumers of Places should tolerate the addition of new fields and variance in ordering of fields with ease. Not all fields appear in all contexts. It is generally safe to consider a nulled field, an empty set, and the absence of a field as the same thing.

Field	Type	Description
<b>attributes</b>	Object	Contains a hash of variant information about the place. See <a href="#">About Geo Place Attributes</a> . Example: <pre>"attributes": {   "street_address": "795 Folsom St",   "623:id": "210176",   "twitter": "twitter" },</pre>

**bounding\_box** [Object](#) A bounding box of coordinates which encloses this place.

### Related API Resources

- [GET geo/id/:place\\_id](#)

# Places

Places | Twitter Developers x

https://dev.twitter.com/docs/platform-objects/places

Developers API Health Blog Discussions Documentation Search

<b>country_code</b>	String	Shortened country code representing the country containing this place. Example: <pre>"country_code": "FR"</pre>
<b>full_name</b>	String	Full human-readable representation of the place's name. Example: <pre>"full_name": "Paris, Paris"</pre>
<b>id</b>	String	ID representing this place. Note that this is represented as a string, not an integer. Example: <pre>"id": "7238f93a3e899af6"</pre>
<b>name</b>	String	Short human-readable representation of the place's name. Example: <pre>"name": "Paris"</pre>
<b>place_type</b>	String	The type of location represented by this place. Example: <pre>"place_type": "city"</pre>
<b>url</b>	String	URL representing the location of additional place metadata for this place. Example: <pre>"url": "http://api.twitter.com/1/geo/id/7238f93a3e899af6.json"</pre>

### Tags

- Places & Geo (28)
- Geo (4)

# Twitter for Websites

The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/twitter-for-websites>. The page title is "Twitter for Websites" and it features a navigation bar with links for "Developers", "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible in the header.

The main content area is titled "Twitter for Websites" and is divided into two columns:

- Twitter Buttons:** This section explains how to select the best button for a website, whether for distributing content, communicating with followers, or discussing a topic using a hashtag. It displays four button examples: "Tweet" (18.6K), "Follow @twitter" (10M+ followers), "Tweet #LetsFly", and "Tweet to @twitterapi". Below the buttons are links to "Create a button for your website", "Read the documentation for the Tweet Button", and "Read the documentation for the Follow Button".
- Embedded Tweets:** This section explains that tweets are dynamic and interactive media. It features an embedded tweet from Twitter (@twitter) with the text "Let's Fly! [blog.twitter.com/2011/12/lets-f...](http://blog.twitter.com/2011/12/lets-f...)" and a timestamp of "1:43 AM - 9 Dec 2011". The tweet includes a "Follow" button and a preview of a blog post titled "Let's Fly | Twitter Blogs" with a blue Twitter bird icon. Below the tweet are statistics for "3,228 RETWEETS" and "1,488 FAVORITES", and a link to "Learn how to embed a Tweet".

# Follow Button

The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/follow-button>. The page title is "Follow Button" and it is part of the "Documentation" section. The page includes a navigation bar with links for "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible. The main content area features a "View" button and a "What links here" button. The text indicates that the Follow Button is a small widget that allows users to easily follow a Twitter account from any webpage. It uses the same implementation model as the Tweet Button and its integration is just as simple. A note states that by using the Follow Button, users agree to the Developer Rules of the Road. The page also includes an "Overview" section and a "Related Case Studies" section with a link to "The Tweet and Follow Buttons deliver big value for Etsy sellers". A "Tags" section lists "Twitter for Websites (32)", "Javascript (46)", and "widgets (33)". An inset image shows a Twitter profile for "AstroIronMike" (Col. Mike Fincke) with a "Follow" button overlaid on the profile picture.

Follow Button | Twitter Dev x

<https://dev.twitter.com/docs/follow-button>

Developers API Health Blog Discussions Documentation Search

Home → Documentation Tweet

## Follow Button

View What links here

Updated on Tue, 2013-12-17 13:40 API version 1 API version 1.1 **Related Case Studies**

The Follow Button is a small widget which allows users to easily follow a Twitter account from any webpage. The Follow Button uses the same implementation model as the [Tweet Button](#), and its integration is just as simple.

By using the Follow Button, you agree to the [Developer Rules of the Road](#).

### Overview

The Follow Button is a small widget which allows users to easily follow a Twitter account from any webpage. The Follow Button uses the same implementation model as the [Tweet Button](#), and its integration is just as simple.

- [The Tweet and Follow Buttons deliver big value for Etsy sellers](#)

### Tags

- [Twitter for Websites](#) (32)
- [Javascript](#) (46)
- [widgets](#) (33)

Follow @astroironmike 18.4K followers

Col. Mike Fincke (@AstroIronMike) on Twitter

**AstroIronMike** Col. Mike Fincke  
Colonel, USAF and NASA Astronaut  
<http://www.nasa.gov/astronauts>

Followers 18,410 Following 10

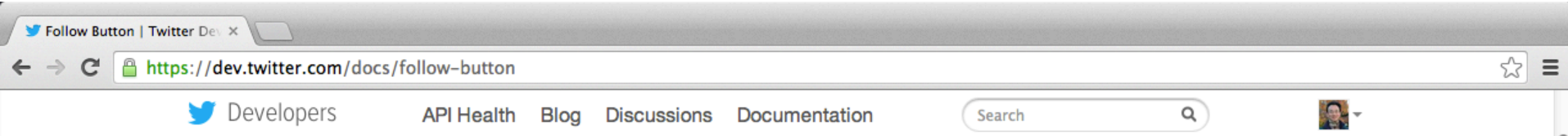
Tweets

**AstroIronMike** Col. Mike Fincke  
Just landed in Amsterdam on the way to Cologne to celebrate German Space Day. There will be a tweetup...

<https://dev.twitter.com/docs/follow-button>



# Follow Button



## Overview

The Follow Button is a small widget which allows users to easily follow a Twitter account from any webpage. The Follow Button uses the same implementation model as the [Tweet Button](#), and its integration is just as simple.



## Tags

- [Twitter for Websites](#) (32)
- [Javascript](#) (46)
- [widgets](#) (33)

This documentation page is for developers and users who wish to build their own Follow Buttons. If you are looking for a quick way to build a Follow Button for your website, you can visit our [Follow Button Configuration Page](#).

## User Interaction

User interaction flow

# Twitter buttons

Follow Button | Twitter Dev | Twitter buttons | About

https://about.twitter.com/resources/buttons#follow

About

Products Company Press Investor relations Blogs

## Twitter buttons

Add buttons to your website to help your visitors share content and connect on Twitter.

Choose a button

Share a link  Follow  Hashtag  Mention

[Tweet](#) 93 [Follow @twitter](#) [Tweet #TwitterStories](#) [Tweet to @support](#)

---

### Button options

User

Show username

Large button

Opt-out of tailoring Twitter [?]

Language

### Preview and code

Try out your button, then copy and paste the code below into the HTML for your site.

[Follow @iMyday](#)

```
<a href="https://twitter.com/iMyday" class=  
<script>function(d,s,id){var js,fjs=d.getElem
```

<https://about.twitter.com/resources/buttons#follow>

# Follow @iMyday



```
<a href="https://twitter.com/iMyday" class="twitter-follow-button" data-show-count="false">Follow @iMyday</a>  
<script>!function(d,s,id){var  
js,fjs=d.getElementsByTagName(s)[0],p=/^http:/.test(d.location)  
?'http':'https';if(!d.getElementById(id)){js=d.createElement(s);js.i  
d=id;js.src=p+'://platform.twitter.com/widgets.js';fjs.parentNode  
.insertBefore(js,fjs);}}(document, 'script', 'twitter-wjs');</script>
```

# Test Twitter Button on jsbin.com

The screenshot shows a web browser window with three tabs: "Follow Button | Twitter Dev", "Twitter buttons | About", and "JS Bin - Collaborative JavaScript". The address bar shows the URL "jsbin.com/oXOZuFl/1/edit". The browser interface includes a "Bins" menu, "Add library", "Share", and tabs for "HTML", "CSS", "JavaScript", "Console", and "Output".

The "HTML" tab is active, displaying the following code:



```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>JS Bin</title>
</head>
<body>
  <a href="https://twitter.com/iMyday" class="twitter-follow-button" data-show-count="false">Follow @iMyday</a>
  <script>!function(d,s,id){var js,fjs=d.getElementsByTagName(s)[0],p=/^http:/.test(d.location)?'http':'https';if(!d.getElementById(id)){js=d.createElement(s);js.id=id;js.src=p+'//platform.twitter.com/widgets.js';fjs.parentNode.insertBefore(js,fjs);}(document, 'script', 'twitter-wjs');
</script>
</body>
</html>
```


The "Output" tab is active, showing the rendered result: a Twitter follow button labeled "Follow @iMyday".

# Follow @iMyday



Follow Button | Twitter Dev x | Twitter buttons | About x | JS Bin - Collaborative JavaS x | Min-Yuh Day (@iMyday) on x

Twitter, Inc. [US] [https://twitter.com/intent/user?original\\_referer=http%3A%2F%2Fplatform.twitter.com%2Fwidgets%2Ffollow\\_button.1387492107.html&region...](https://twitter.com/intent/user?original_referer=http%3A%2F%2Fplatform.twitter.com%2Fwidgets%2Ffollow_button.1387492107.html&region...)

  iMyday ▾


 **Min-Yuh Day** @iMyday [Edit your profile →](#)  
*Assistant Professor, Department of Information Management, Tamkang University*  
<http://mail.tku.edu.tw/myday>

Followers **14**      Following **408**


      

**Tweets**

---

 **Min-Yuh Day** @iMyday  
Good morning 2014 [twitter.com/iMyday/status/...](https://twitter.com/iMyday/status/...)  
about 1 hour ago    Reply    Favorite

---

 **Min-Yuh Day** @iMyday  
Morning, 2014.01.01 [twitter.com/iMyday/status/...](https://twitter.com/iMyday/status/...)  
about 9 hours ago

[View Min-Yuh Day's full profile →](#)

# Twitter Search API

The image is a screenshot of a web browser displaying the Twitter Search API documentation. The browser's address bar shows the URL <https://dev.twitter.com/docs/using-search>. The page header includes the Twitter logo, the word "Developers", and navigation links for "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible in the header. The main content area features the title "Using the Twitter Search API" and two buttons: "View" and "What links here". A note indicates the page was updated on Wednesday, 2013-10-02 at 06:34, and a badge for "API version 1.1" is present. The text explains that the Twitter Search API is part of Twitter's v1.1 REST API and allows queries against recent or popular tweets. It notes that the API is focused on relevance and not completeness, and suggests using the Streaming API for completeness. A detailed reference to the API endpoint is provided as [GET search/tweets](#). A section titled "How to build a query" provides a step-by-step guide: 1. Search for tweets referencing @twitterapi on [twitter.com/search](#). 2. Copy the URL, which in this case is <https://twitter.com/search?q=%40twitterapi>. 3. Replace "https://twitter.com/search" with "https://api.twitter.com/1.1/search/tweets.json" to get <https://api.twitter.com/1.1/search/tweets.json?q=%40twitterapi>. 4. Execute this URL to do the search in the API. A final note states that API v1.1 requires authentication, and refers to [Authentication & Authorization](#) documentation for more details. On the right side of the page, there are sections for "Related open issues" (with a link to "Search API: '-filter:links' not affecting results"), "Related Questions" (with a link to "Why are the Tweets I'm looking for not in Twitter Search, the Search API, or Search widgets?"), "Related Case Studies" (with a link to "Esri enriches maps with Tweets and the Streaming API"), and "Tags" (with links to "Search (34)" and "finding tweets (49)").

Using the Twitter Search API

Updated on Wed, 2013-10-02 06:34

API version 1.1

The Twitter Search API is part of Twitter's v1.1 REST API. It allows queries against the indices of recent or popular Tweets and behaves similarly to, but not exactly like the Search feature available in Twitter mobile or web clients, such as [Twitter.com search](#).

Before getting involved, it's important to know that the Search API is focused on relevance and not completeness. This means that some Tweets and users may be missing from search results. If you want to match for completeness you should consider using a [Streaming API](#) instead.

A detailed reference on this API endpoint can be found at [GET search/tweets](#).

### How to build a query

The best way to build a query and test if it's valid and will return matched Tweets is to first try it at [twitter.com/search](#). As you get a satisfactory result set, the URL loaded in the browser will contain the proper query syntax that can be reused in the API endpoint. Here's an example:

1. We want to search for tweets referencing @twitterapi account. First, we run the search on [twitter.com/search](#)
2. Check and copy the URL loaded. In this case, we got: <https://twitter.com/search?q=%40twitterapi>
3. Replace "https://twitter.com/search" with "https://api.twitter.com/1.1/search/tweets.json" and you will get:  
**<https://api.twitter.com/1.1/search/tweets.json?q=%40twitterapi>**
4. [Execute this URL](#) to do the search in the API

Please note that now API v1.1 requires that the request must be authenticated, check [Authentication & Authorization](#) documentation for more details on how to do it. Also note that the search results at twitter.com may return historical results while the Search API usually only serves tweets from the past week.

### Related open issues

- [Search API: "-filter:links" not affecting results](#)

### Related Questions

- [Why are the Tweets I'm looking for not in Twitter Search, the Search API, or Search widgets?](#)

### Related Case Studies

- [Esri enriches maps with Tweets and the Streaming API](#)

### Tags

- [Search \(34\)](#)
- [finding tweets \(49\)](#)

# Twitter REST API

The screenshot shows the Twitter REST API documentation page. The browser address bar displays <https://dev.twitter.com/docs/api>. The page header includes navigation links for Developers, API Health, Blog, Discussions, and Documentation, along with a search bar and a user profile picture. The main content area features the title "The Twitter REST API" and a "Jump to" dropdown menu. Two main sections are visible: "REST API version 1.1" and "REST API version 1".

Home

## The Twitter REST API

[Jump to](#)

### REST API version 1.1

The most recent version of the Twitter REST API.

- [API v1.1 Resources »](#)
- [Rate Limiting in API v1.1 »](#)
- [Authenticating »](#)
- [Announcement »](#)

### REST API version 1

Version 1 of the REST API is now deprecated and will cease functioning in the coming months. Migrate to version 1.1 today.

[Review the deprecated version 1 API »](#)

[Follow @twitterapi](#)

[API Terms](#) [API Status](#) [Blog](#) [Discussions](#) [Documentation](#) A Drupal community site supported by Acquia

# Streaming API

The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/streaming-apis>. The page title is "The Streaming APIs". It features a navigation bar with "Developers", "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible. The main content area includes a breadcrumb "Home → Documentation", a "Tweet" button, and a dropdown menu for "The Streaming APIs". Below this, there are buttons for "View" and "What links here". The page is updated on "Mon, 2012-09-24 13:47" and has two "API version 1.1" tags. The "Overview" section explains that the streaming APIs provide low latency access to Twitter's global stream of Tweet data. It lists three types of streams: "Public streams" (public data), "User streams" (single-user), and "Site streams" (multi-user). The "Differences between Streaming and REST" section notes that streaming requires a persistent HTTP connection. On the right, there are sections for "Related open issues" (missing favorite\_count field), "Related Questions" (rate limits, difference between User Streams and Site Streams), and "Related Case Studies" (Zappos, Esri).

The Streaming APIs

View What links here

Updated on Mon, 2012-09-24 13:47

API version 1 API version 1.1

## Overview

The set of streaming APIs offered by Twitter give developers low latency access to Twitter's global stream of Tweet data. A proper implementation of a streaming client will be pushed messages indicating Tweets and other events have occurred, without any of the overhead associated with polling a REST endpoint.

Twitter offers several streaming endpoints, each customized to certain use cases.

<b>Public streams</b>	Streams of the public data flowing through Twitter. Suitable for following specific users or topics, and data mining.
<b>User streams</b>	Single-user streams, containing roughly all of the data corresponding with a single user's view of Twitter.
<b>Site streams</b>	The multi-user version of user streams. Site streams are intended for servers which must connect to Twitter on behalf of many users.

## Differences between Streaming and REST

Connecting to the streaming API requires keeping a persistent HTTP connection open. In many cases this involves thinking about your application differently than if you were interacting with the REST API. For an example, consider a web application which accepts user requests, makes one or more requests to Twitter's API, then formats and prints the result to the user, as a response to the user's initial request:

## Related open issues

- Tweets streamed in the Streaming API are missing the favorite\_count field

## Related Questions

- How are rate limits determined on the Streaming API?
- What's the difference between User Streams and Site Streams?

## Related Case Studies

- Zappos uses Twitter to generate over 1,200 conversations per month with its customers
- Esri enriches maps with Tweets and the Streaming API



# Twitter REST API

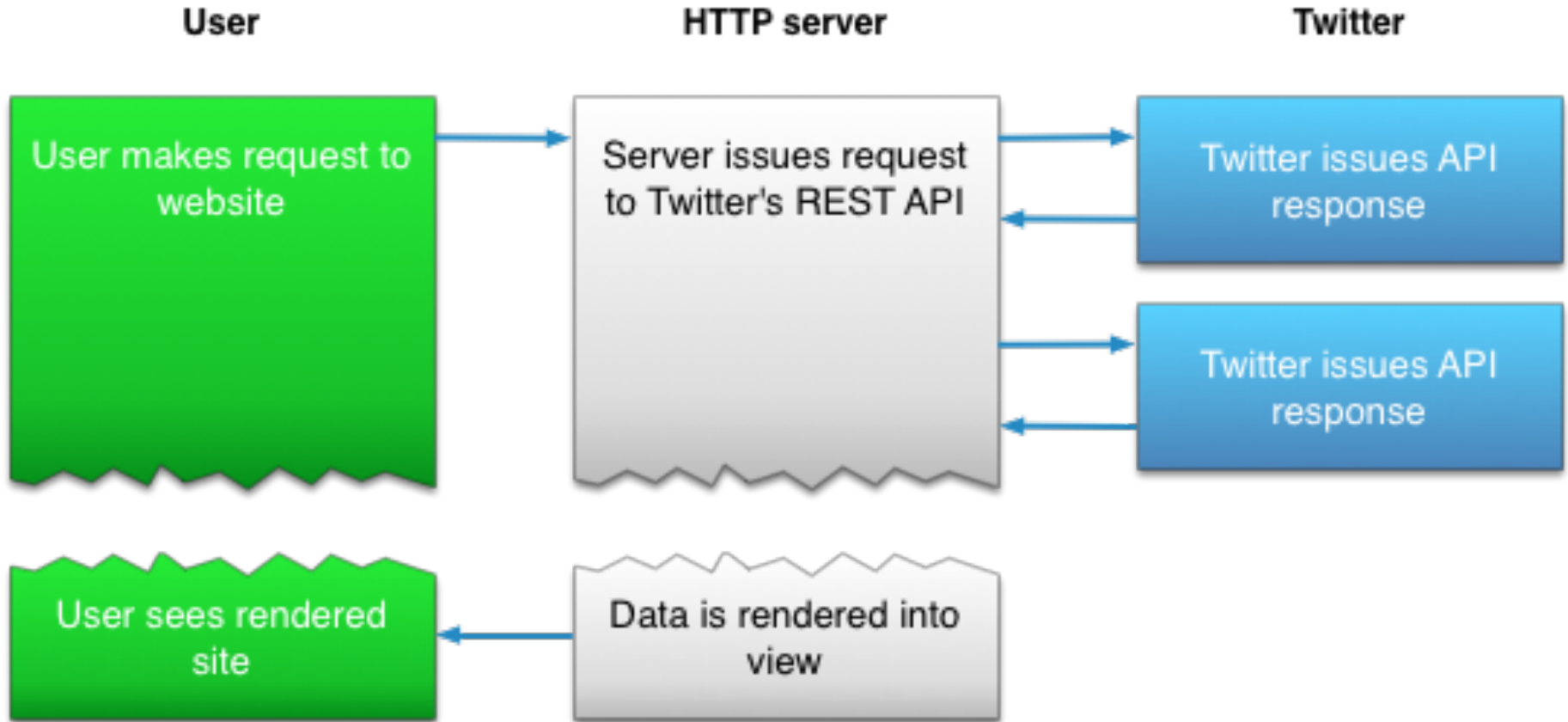
The REST API provides simple interfaces for most Twitter functionality.

# Twitter Streaming API

The Streaming API is a family of powerful real-time APIs for Tweets and other social events.

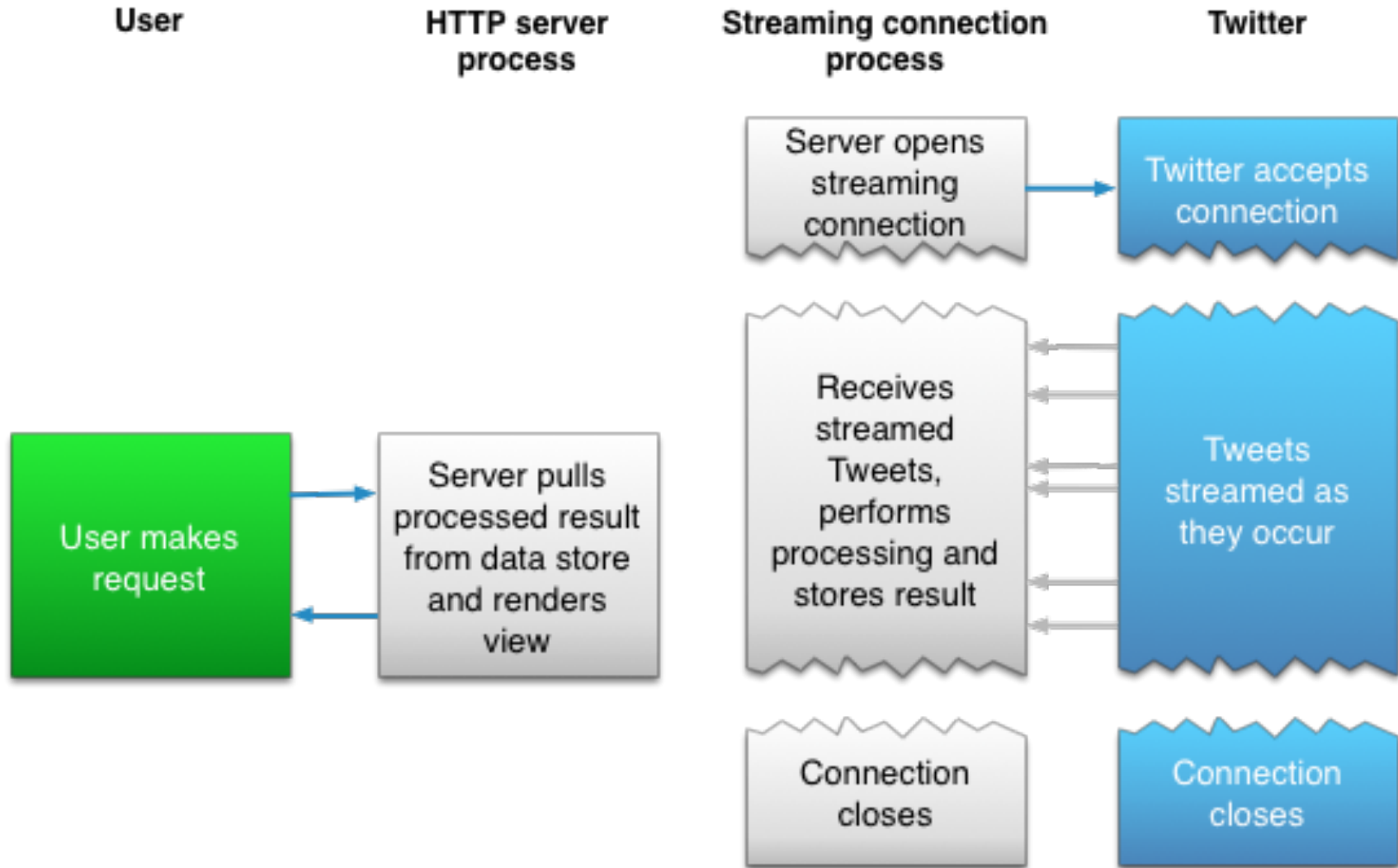
# Differences between Streaming and REST API

## Twitter REST API



# Differences between Streaming and REST API

## Twitter Streaming API



# Exploring the Twitter API

The screenshot shows a web browser window with the URL <https://dev.twitter.com/console>. The page title is "Exploring the Twitter API". The navigation bar includes "Developers", "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile icon are also present. The main content area features a breadcrumb trail: "Home → Documentation → Getting Started". A "Tweet" button is visible in the top right. The main heading is "Exploring the Twitter API", with a "Getting Started" dropdown menu to its right. Below the heading are two buttons: "View" and "What links here". A timestamp indicates the page was updated on Thursday, 2013-01-17 at 07:58. The text explains that users can explore the API on their own machine using "Twitter for Mac" or the "Twurl" tool, and also provides a link to a console provided by @apigee. The console interface is shown below, featuring a "Service" dropdown, an "Authentication" dropdown set to "No Auth", and a "Working..." status indicator. The "Request URL" section includes a dropdown for the HTTP method (currently "GET") and a text input field with a placeholder "← Select an API method". A "Send" button is located to the right of the input field. Below the "Request URL" section are tabs for "Query", "Template", "Headers", and "Body". The bottom of the console is divided into two columns: "Request" and "Response". The "Request" column contains the text "Select an API method". The "Response" column is currently empty and includes a "Snapshot" button.

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



# Create a new app

My applications | Twitter

https://dev.twitter.com/apps

Developers API Health Blog Discussions Documentation Search

Home

## My applications

Looks like you haven't created any applications yet!

Create a new application

**Apps**

Follow @twitterapi

API Terms API Status Blog Discussions Documentation A Drupal community site supported by Acquia

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



# Create a new app

Create an application | Twi x

https://dev.twitter.com/apps/new

Developers API Health Blog Discussions Documentation Search

Home → My applications

## 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.



# Create a new app

Create an application | Twi x

← → ↻ <https://dev.twitter.com/apps/new> ☆ ☰

Developers API Health Blog Discussions Documentation Search 

Last update: July 2, 2016.

## 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 describes 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 current version. Don't do anything prohibited by the Rules and talk to us if you think we should make a change or give you an exception.



If your application will eventually need more than 1 million user tokens, or you expect your [embedded Tweets](#) and [embedded timelines](#) to exceed 10 million daily impressions, you will need to talk to us directly about your access to the Twitter API as you may be subject to additional terms. Furthermore, applications that attempt to replicate Twitter's core user experience (as described in Section I.5 below) will need our permission to

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

This question is for testing whether you are a human visitor and to prevent automated spam submissions.



[Privacy & Terms](#)

Create your Twitter application

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





# Create a new app

iMydayMobileApp | Twitter x

← → ↻ <https://dev.twitter.com/apps/5602991/show> ☆ ☰

Developers API Health Blog Discussions Documentation Search

[Home](#) → [My applications](#)

## iMydayMobileApp

[Details](#) [Settings](#) [OAuth tool](#) [@Anywhere domains](#) [Reset keys](#) [Delete](#)

iMyday Mobile App  
<http://mail.tku.edu.tw/myday/app>

### 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 <a href="#">About the application permission model</a>
Consumer key	1KHzfFM3ejM6OQ
Consumer secret	TMsvzTNuTNDrUnY7hb



# Create a new app

iMydayMobileApp | Twitter x

https://dev.twitter.com/apps/5602991/show

Developers API Health Blog Discussions Documentation Search

## 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 <a href="#">About the application permission model</a>
Consumer key	1KHzfFM3ejM6OQ
Consumer secret	TMsvzTNuTNDrUnY7hb5ZI
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

## Your access token

Use the access token string as your "oauth\_token" and the access token secret as your "oauth\_token\_secret" to sign requests with your own Twitter account. Do not share your oauth\_token\_secret with anyone.

Access token	227161-Ky65DQp715thnDTC
Access token secret	gTTTz6CDg8iApgAP81q7lc
Access level	Read-only

[Recreate my access token](#)



# Create a new app

**Consumer key** IKHzfFM3ejM6O\*\*\*\*\*

**Consumer secret**

TMsvzTNUtNDrUnY7hb5ZlarXqZDnsKW\*\*\*\*\*

Request token URL

[https://api.twitter.com/oauth/request\\_token](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](https://api.twitter.com/oauth/access_token)

# Update Twitter's App Settings

The screenshot shows a web browser window with the URL `https://dev.twitter.com/apps/5602991/settings`. The page title is "iMydayMobileApp | Twitter". The navigation bar includes links for "Developers", "API Health", "Blog", "Discussions", and "Documentation", along with a search bar and a user profile picture. The breadcrumb trail is "Home → My applications".

## iMydayMobileApp

Details Settings OAuth tool @Anywhere domains Reset keys Delete


### 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.)

### Application Icon


 Change icon:

# Update Twitter's App Settings

The screenshot shows the Twitter Developer Portal interface. The browser address bar displays the URL `https://dev.twitter.com/apps/5602991/settings`. The page title is "iMydayMobileApp | Twitter". The navigation bar includes links for "Developers", "API Health", "Blog", "Discussions", and "Documentation", along with a search bar and a user profile picture.

## Application Icon

Change icon:

  No file chosen

Maximum size of 700k. JPG, GIF, PNG.

## Application Type

Access:

- Read only
- Read and Write
- Read, Write and Access direct messages

What type of access does your application need? Note: @Anywhere applications require read & write access. Find out more about our [Application Permission Model](#).

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.

Allow this application to be used to [Sign in with Twitter](#)

When enabled your application can be used to "Sign in with Twitter". When disabled your application will not be able to use `/oauth/authenticate` and any request to it will instead redirect the user to `/oauth/authorize`

## Organization

Organization name:

# Update Twitter's App Settings

The screenshot shows a web browser window with the address bar displaying `https://dev.twitter.com/apps/5602991/settings`. The page header includes the Twitter logo, navigation links for 'Developers', 'API Health', 'Blog', 'Discussions', and 'Documentation', a search bar, and a user profile picture. The main content area is divided into sections for 'Access', 'Callback URL', and 'Organization'. The 'Access' section has three radio button options: 'Read only', 'Read and Write' (which is selected), and 'Read, Write and Access direct messages'. Below this is a note about access types and a link to the 'Application Permission Model'. The 'Callback URL' section has an empty text input field and a note explaining its purpose. The 'Organization' section has two text input fields for 'Organization name' and 'Organization website', each with a note below it. At the bottom, a blue button labeled 'Update this Twitter application's settings' is highlighted with a red rectangular border.

iMydayMobileApp | Twitter x

← → ↻ <https://dev.twitter.com/apps/5602991/settings> ☆ ☰

Twitter Developers API Health Blog Discussions Documentation Search [Profile Picture]

**Access:**

Read only

Read and Write

Read, Write and Access direct messages

What type of access does your application need? Note: @Anywhere applications require read & write access. Find out more about our [Application Permission Model](#).

---

**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.

---

Allow this application to be used to [Sign in with Twitter](#)

When enabled your application can be used to "Sign in with Twitter". When disabled your application will not be able to use `/oauth/authenticate` and any request to it will instead redirect the user to `/oauth/authorize`

---

**Organization**

**Organization name:**

The organization or company behind this application, if any.

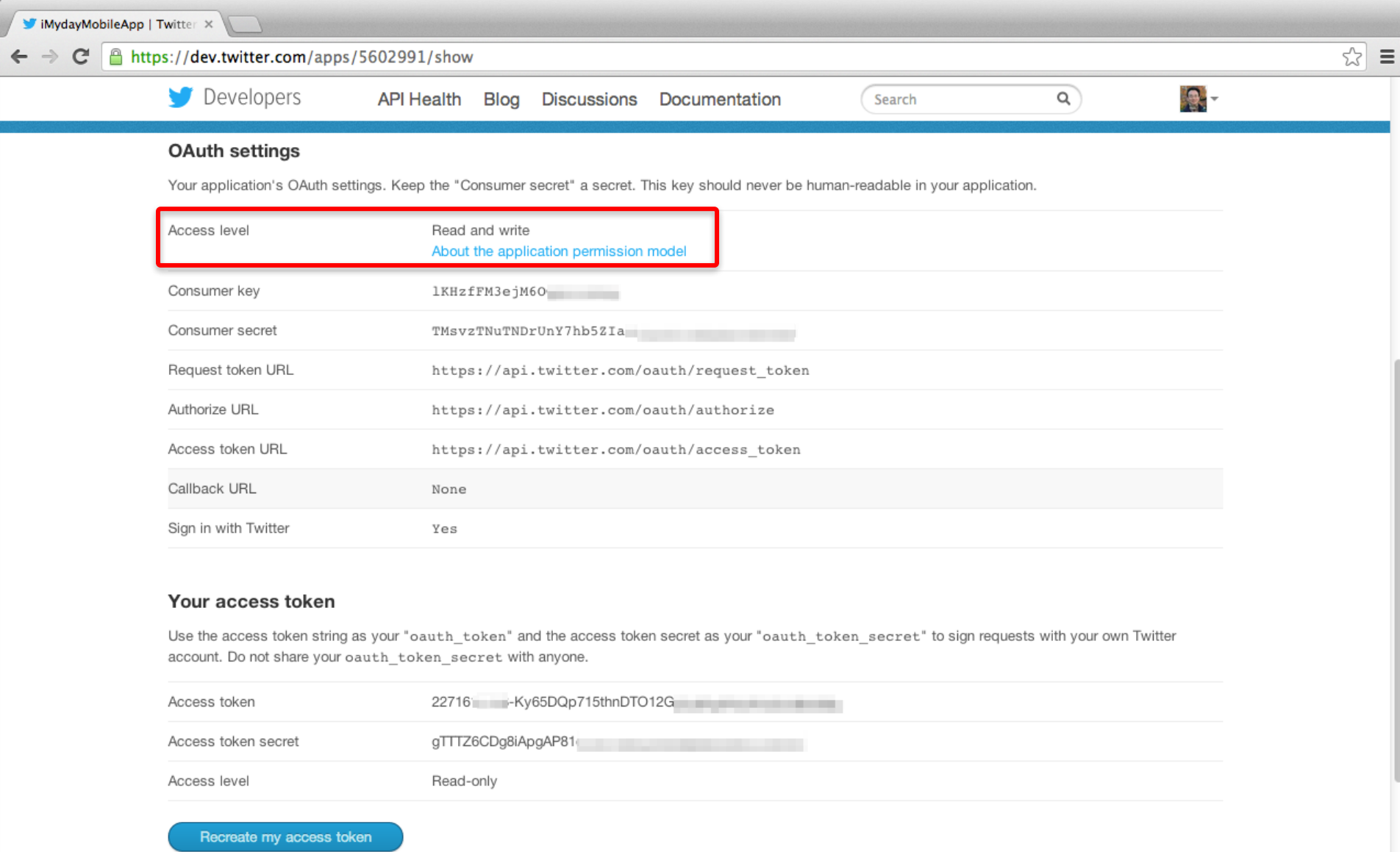
---

**Organization website:**

The organization or company behind this application's web page, if any.


[Update this Twitter application's settings](#)

# OAuth Settings Updated



iMydayMobileApp | Twitter x

← → ↻ <https://dev.twitter.com/apps/5602991/show> ☆ ☰

Developers API Health Blog Discussions Documentation Search 

## 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 and write <a href="#">About the application permission model</a>
Consumer key	1KHzfFM3ejM6O
Consumer secret	TMsvzTNuTNDrUnY7hb52Ia
Request token URL	<a href="https://api.twitter.com/oauth/request_token">https://api.twitter.com/oauth/request_token</a>
Authorize URL	<a href="https://api.twitter.com/oauth/authorize">https://api.twitter.com/oauth/authorize</a>
Access token URL	<a href="https://api.twitter.com/oauth/access_token">https://api.twitter.com/oauth/access_token</a>
Callback URL	None
Sign in with Twitter	Yes

## Your access token

Use the access token string as your "oauth\_token" and the access token secret as your "oauth\_token\_secret" to sign requests with your own Twitter account. Do not share your oauth\_token\_secret with anyone.

Access token	22716-Ky65DQp715thnDTO12G
Access token secret	gTTTZ6CDg8iAppAP81
Access level	Read-only

[Recreate my access token](#)

# Twitter REST API v1.1 Resources

REST API v1.1 Resources | x

← → ↻ <https://dev.twitter.com/docs/api/1.1> ☆ ☰

Developers API Health Blog Discussions Documentation Search Sign in

Home

## REST API v1.1 Resources

Jump to

### Timelines

Timelines are collections of Tweets, ordered with the most recent first.

Resource	Description
<a href="#">GET statuses/mentions_timeline</a>	Returns the 20 most recent mentions (tweets containing a users's @screen_name) for the authenticating user. The timeline returned is the equivalent of the one seen when you view your mentions on twitter.com. This method can only return up to 800 tweets. See Working with Timelines for...
<a href="#">GET statuses/user_timeline</a>	Returns a collection of the most recent Tweets posted by the user indicated by the screen_name or user_id parameters. User timelines belonging to protected users may only be requested when the authenticated user either "owns" the timeline or is an approved follower of the owner. The timeline...
<a href="#">GET statuses/home_timeline</a>	Returns a collection of the most recent Tweets and retweets posted by the authenticating user and the users they follow. The home timeline is central to how most users interact with the Twitter service. Up to 800 Tweets are obtainable on the home timeline. It is more volatile for users that follow...
<a href="#">GET statuses/retweets_of_me</a>	Returns the most recent tweets authored by the authenticating user that have been retweeted by others. This timeline is a subset of the user's GET statuses/user_timeline. See Working with Timelines for instructions on traversing timelines.

### Tweets

Tweets are the atomic building blocks of Twitter, 140-character status updates with additional associated metadata. People tweet for a variety of reasons about a multitude of topics.

Resource	Description
----------	-------------



# GET search/tweets

The screenshot shows the Twitter API documentation page for the GET search/tweets endpoint. The page is titled "GET search/tweets" and is part of the "REST API" documentation. It includes a navigation bar with "Developers", "API Health", "Blog", "Discussions", and "Documentation" links. A search bar and a "Sign in" button are also present. The main content area features a "View" button and a "What links here" button. The text describes the endpoint's purpose, notes that it is not an exhaustive source of tweets, and mentions improvements in API v1.1. A "Resource URL" section provides the endpoint path: https://api.twitter.com/1.1/search/tweets.json. A "Parameters" section lists the 'q' parameter (required) and the 'geocode' parameter (optional). A "Related open issues" section highlights a problem with the 'count' parameter. A "Resource Information" table lists details such as rate limits, authentication requirements, and response formats.

GET search/tweets | Twitter x

https://dev.twitter.com/docs/api/1.1/get/search/tweets

Developers API Health Blog Discussions Documentation Search Sign in

Home → Documentation → REST API Tweet

## GET search/tweets

View What links here

Updated on Thu, 2013-03-07 09:35 API version 1.1

Returns a collection of relevant [Tweets](#) matching a specified query.

Please note that Twitter's search service and, by extension, the Search API is not meant to be an exhaustive source of Tweets. Not all Tweets will be indexed or made available via the search interface.

In API v1.1, the response format of the Search API has been improved to return [Tweet objects](#) more similar to the objects you'll find across the REST API and platform. You may need to tolerate some inconsistencies and variance in perspectival values (fields that pertain to the perspective of the authenticating user) and embedded user objects.

To learn how to use [Twitter Search](#) effectively, consult our guide to [Using the Twitter Search API](#). See [Working with Timelines](#) to learn best practices for navigating results by `since_id` and `max_id`.

### Resource URL

https://api.twitter.com/1.1/search/tweets.json

### Parameters

<b>q</b> required	A UTF-8, URL-encoded search query of 1,000 characters maximum, including operators. Queries may additionally be limited by complexity.  <b>Example Values:</b> @noradio
<b>geocode</b> optional	Returns tweets by users located within a given radius of the given latitude/longitude. The location is preferentially taking from the Geotagging API, but will fall back to their Twitter profile. The parameter value is specified by

### Related open issues

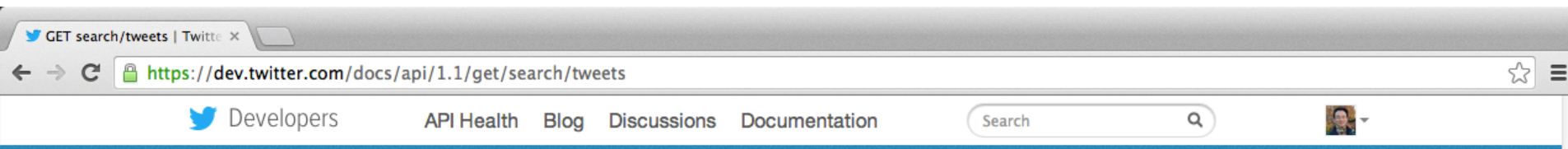
[Search API: Count param does not work when result type is set to mixed](#)

### Resource Information

Rate Limited?	Yes
Requests per rate limit window	180/user 450/app
Authentication	Required
Response Formats	json
HTTP Methods	GET
Resource family	search
Response Object	Tweets
API Version	v1.1

<https://dev.twitter.com/docs/api/1.1/get/search/tweets>

# GET search/tweets



## Example Request

Use the [OAuth tool](#) in this page sidebar to generate the OAuth signature for this request.

**GET** `https://api.twitter.com/1.1/search/tweets.json?q=%23freebandnames&since_id=24012619984051000&max_id=250126199840518145&result_type=mixed&count=4`

```
1. {
2.   "statuses": [
3.     {
4.       "coordinates": null,
5.       "favorited": false,
6.       "truncated": false,
7.       "created_at": "Mon Sep 24 03:35:21 +0000 2012",
8.       "id_str": "250075927172759552",
9.       "entities": {
10.        "urls": [
11.
12.        ],
13.        "hashtags": [
14.          {
15.            "text": "freebandnames",
16.            "indices": [
17.              20,
18.              34
19.            ]
20.          }
21.        ],
22.        "user_mentions": [
23.
24.        ]
25.      },
26.      "in_reply_to_user_id_str": null,
27.      "contributors": null,
28.      "text": "Aggressive Ponytail #freebandnames",
29.      "metadata": {
30.        "iso_language_code": "en",
31.        "result_type": "recent"
32.      },
33.      "retweet_count": 0
```

<https://dev.twitter.com/docs/api/1.1/get/search/tweets>

# GET search/tweets

```
https://api.twitter.com/1.1/search/tweets.json?q=%23freebandnames&since_id=24012619984051000&max_id=250126199840518145&result_type=mixed&count=4
```



# OAuth Tool

via the API to geocode arbitrary locations; however you can use this `geocode` parameter to search near geocodes directly. A maximum of 1,000 distinct "sub-regions" will be considered when using the radius modifier.

**Example Values:** `37.781157,-122.398720,1mi`

<b>lang</b> optional	Restricts tweets to the given language, given by an <a href="#">ISO 639-1</a> code. Language detection is best-effort.  <b>Example Values:</b> <code>eu</code>
<b>locale</b> optional	Specify the language of the query you are sending (only <code>ja</code> is currently effective). This is intended for language-specific consumers and the default should work in the majority of cases.  <b>Example Values:</b> <code>ja</code>
<b>result_type</b> optional	Optional. Specifies what type of search results you would prefer to receive. The current default is "mixed." Valid values include: <ul style="list-style-type: none"><li>* <code>mixed</code>: Include both popular and real time results in the response.</li><li>* <code>recent</code>: return only the most recent results in the response</li><li>* <code>popular</code>: return only the most popular results in the response.</li></ul> <b>Example Values:</b> <code>mixed, recent, popular</code>
<b>count</b> optional	The number of tweets to return per page, up to a maximum of 100. Defaults to 15. This was formerly the "rpp" parameter in the old Search API.  <b>Example Values:</b> <code>100</code>
<b>until</b> optional	Returns tweets generated before the given date. Date should be formatted as YYYY-MM-DD. Keep in mind that the search index may not go back as far as the date you specify here.  <b>Example Values:</b> <code>2012-02-01</code>

### OAuth tool

This tool will generate the OAuth signature for the [example request](#)

Select one of your Apps ▾

[Generate OAuth signature](#)

### Related Documentation

- [GET users/search](#)
- [Using the Twitter Search API](#)
- [GET geo/search](#)
- [Finding Tweets about Places](#)

### Related Questions

- [Why are the Tweets I'm looking for not in Twitter Search, the Search API, or Search widgets?](#)

### Tags

- [Search](#) (34)
- [finding tweets](#) (49)

# OAuth Tool

## OAuth tool

---

This tool will generate the OAuth signature for the [example request](#)


**Generate OAuth signature**

# OAuth Tool

The screenshot shows a web browser window with the URL `https://dev.twitter.com/apps/5602991/oauth?nid=10287`. The page title is "iMydayMobileApp | Twitter". The navigation bar includes "Developers", "API Health", "Blog", "Discussions", "Documentation", a search box, and a user profile picture. The breadcrumb trail is "Home → My applications". The main heading is "iMydayMobileApp". Below it are tabs for "Details", "Settings", "OAuth tool" (which is active), "@Anywhere domains", "Reset keys", and "Delete". The "OAuth Settings" section contains four input fields: "Consumer key" (value: IKHzfFM3ej...), "Consumer secret" (value: TMsVzTNuTNDr...), "Access token" (value: 22716...-Ky...), and "Access token secret" (value: gTTTZ6CDg8iApg...). Each secret field has a note: "Remember this should not be shared." The "Request Settings" section is partially visible at the bottom.

iMydayMobileApp | Twitter

[←](#) [→](#) [↻](#) <https://dev.twitter.com/apps/5602991/oauth?nid=10287> ☆ ☰

Developers API Health Blog Discussions Documentation Search [Profile]

Home → My applications

## iMydayMobileApp

Details Settings **OAuth tool** @Anywhere domains Reset keys Delete

### OAuth Settings

**Consumer key:** \*

**Consumer secret:** \*

  
Remember this should not be shared.

**Access token:** \*

**Access token secret:** \*

  
Remember this should not be shared.

### Request Settings

# OAuth Tool

The screenshot shows the Twitter OAuth Tool interface. At the top, the browser address bar displays the URL `https://dev.twitter.com/apps/5602991/oauth?nid=10287`. The page header includes navigation links for "Developers", "API Health", "Blog", "Discussions", and "Documentation", along with a search bar and a user profile picture. The main content area is titled "Request Settings" and contains the following fields:

- Request type:** A dropdown menu with radio buttons for GET (selected), POST, DELETE, PUT, and HEAD.
- Request URI:** A text input field containing `https://api.twitter.com/1.1/search/tweets.json`. Below the field is a note: "The full URI, without parameters. For example: `https://api.twitter.com/1/statuses/home_timeline.json`".
- Request query:** A text input field containing `q=%23freebandnames&since_id=24012619984051000&max_id=250126199840518145&result_type=mixed&count=4`. Below the field is a note: "The parameters for your request. For example: `include_entities=true&page=2`. Note these parameters will be sent on the querystring for GET requests, and in the request body for POST requests."

At the bottom of the settings area, there is a blue button with the text "See OAuth signature for this request", which is highlighted with a red rectangular border.

**See OAuth signature for this request**

# OAuth Signing Results

The screenshot shows the Twitter Developer console for the application 'iMydayMobileApp'. The browser address bar shows the URL `https://dev.twitter.com/apps/5602991/oauth?nid=10287`. The page title is 'iMydayMobileApp' and the navigation bar includes 'Developers', 'API Health', 'Blog', 'Discussions', and 'Documentation'. The main content area has tabs for 'Details', 'Settings', 'OAuth tool', '@Anywhere domains', 'Reset keys', and 'Delete'. The 'OAuth tool' tab is active, displaying 'OAuth Signing Results'. An important note states: 'Important: This will only be valid for a few minutes. Also remember the cURL command will actually execute the request.' Below this, three sections are shown: 'Signature base string', 'Authorization header', and 'cURL command', each with its corresponding value.

Home → My applications

## iMydayMobileApp

Details Settings **OAuth tool** @Anywhere domains Reset keys Delete

### OAuth Signing Results

**Important:** This will only be valid for a few minutes. Also remember the cURL command will actually execute the request.

Signature base string `GET&https%3A%2F%2Fapi.twitter.com%2F1.1%2Fsearch%2Ftweets.json&count%3D4%26max_id%3D2501261998&oauth_consumer_key%3DlKHzfFM3jM6[REDACTED]&oauth_nonce%3D121adf0a26f314c57ce4[REDACTED]&oauth_signature_method%3DHMAC-SHA1&oauth_timestamp%3D138869[REDACTED]&oauth_token%3D227161[REDACTED]-Ky65DQp715thnDT012GZUIPej[REDACTED]&oauth_version%3D1.0%26q%3D%2523freebandnames%26result_type%3Dmixed%26since_id%3D24012619984051000`

Authorization header `Authorization: OAuth oauth_consumer_key="lKHzfFM3jM6[REDACTED]", oauth_nonce="121adf0a26f314c57ce4[REDACTED]", oauth_signature="Zq5vnDcf1276prp2[REDACTED]", oauth_signature_method="HMAC-SHA1", oauth_timestamp="138869[REDACTED]", oauth_token="227161[REDACTED]-Ky65DQp715thnDT012GZUIPej[REDACTED]", oauth_version="1.0"`

cURL command `curl --get 'https://api.twitter.com/1.1/search/tweets.json' --data 'count=4&max_id=2501261998&q=%23freebandnames&result_type=mixed&since_id=24012619984051000' --header 'Authorization: OAuth oauth_consumer_key="lKHzfFM3jM6[REDACTED]", oauth_nonce="121adf0a26f314c57ce4[REDACTED]", oauth_signature="Zq5vnDcf1276prp2[REDACTED]", oauth_signature_method="HMAC-SHA1", oauth_timestamp="138869[REDACTED]", oauth_token="227161[REDACTED]-Ky65DQp715thnDT012GZUIPej[REDACTED]", oauth_version="1.0" --verbose`



# Web Intents

Web Intents | Twitter Developer x

← → ↻ <https://dev.twitter.com/docs/intents> ☆ ☰

Developers API Health Blog Discussions Documentation Search [Profile]

Home → Documentation [Tweet](#)

## Web Intents

[View](#) [What links here](#)

Updated on Thu, 2013-04-18 12:34 [API version 1](#) [API version 1.1](#) [Related Case Studies](#)

By using Web Intents, you agree to the [Developer Rules of the Road](#).

**Make it easy to bring interactivity to Tweets that you display on the web.**

Web Intents provide popup-optimized flows for working with Tweets & Twitter Users: Tweet, Reply, Retweet, Favorite, and Follow. They make it possible for users to interact with Twitter content in the context of your site, without leaving the page or having to authorize a new app just for the interaction. Web intents are mobile friendly, and super easy to implement.

[Web Intents](#)

### Related Case Studies

- [ESPN uses Web Intents to send thousands of Tweets every day](#)

### Tags

- [Web Intents](#) (24)
- [Twitter for Websites](#) (32)
- [Javascript](#) (46)

Post a Tweet on Twitter

Twitter, Inc. [US] [https://twitter.com/intent/tweet?in\\_reply\\_to=4705005051...](https://twitter.com/intent/tweet?in_reply_to=4705005051...)

Reply to a Tweet

**jack** Jack Dorsey  
5 years ago today we started programming Twitter ("twtr" for short).  
8 days later the first tweet was sent: [#twtr](http://t.co/Vi5iI5A)  
2:43 PM Mar 13th

@jack

Post a Tweet on Twitter

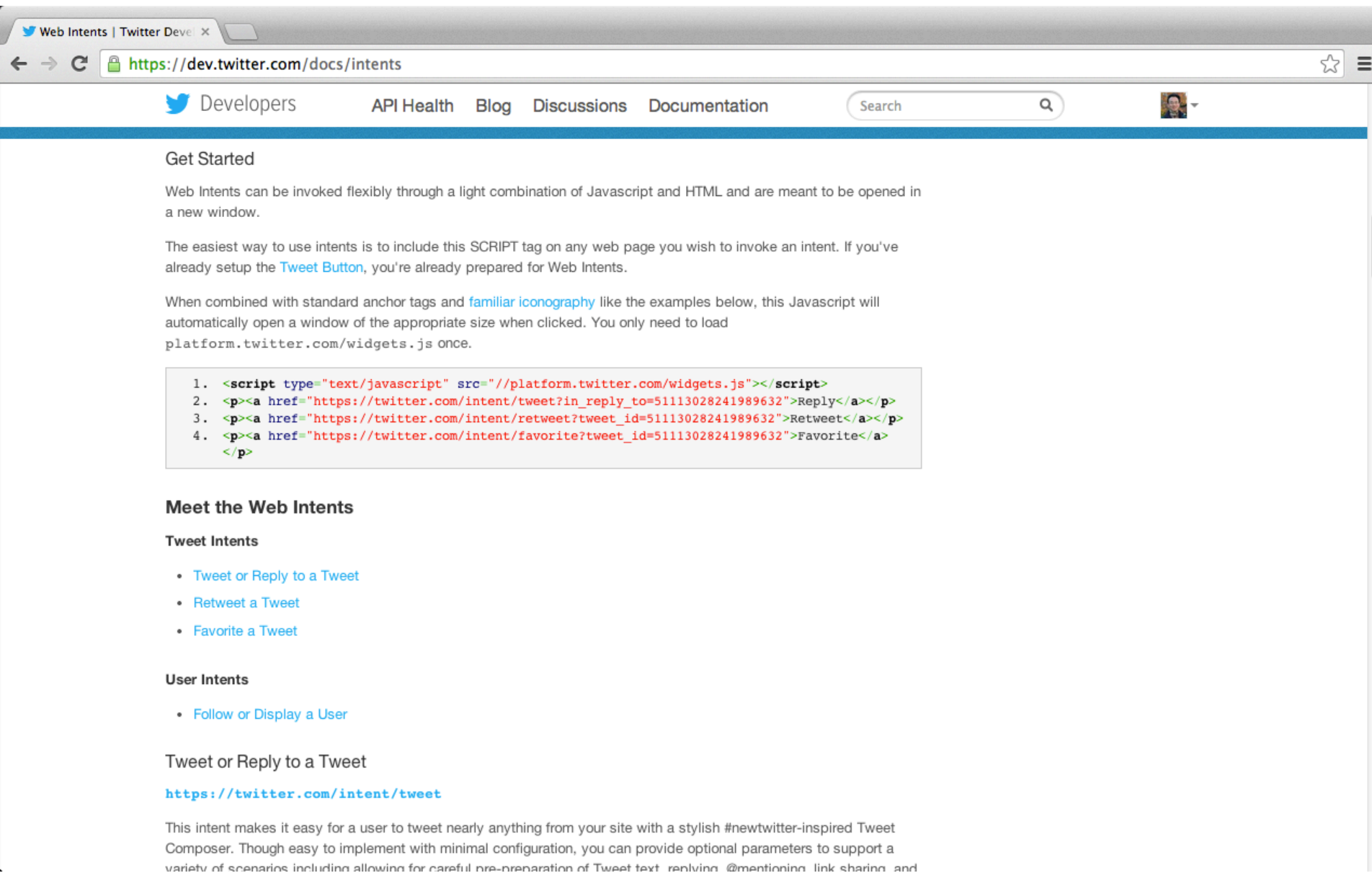
Twitter, Inc. [US] [https://twitter.com/intent/retweet?tweet\\_id=4705005051...](https://twitter.com/intent/retweet?tweet_id=4705005051...)

Retweet to your followers?

**Jack** Jack Dorsey  
5 years ago today we started programming Twitter ("twtr" for short).  
8 days later the first tweet was sent: [#twtr](http://t.co/Vi5iI5A)

<https://dev.twitter.com/docs/intents>

# Web Intents



The image shows a browser window with the URL <https://dev.twitter.com/docs/intents>. The page title is "Web Intents | Twitter Developer". The navigation bar includes "Developers", "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible. The main content area is titled "Get Started" and explains that Web Intents can be invoked flexibly through a combination of Javascript and HTML. It provides the easiest way to use intents by including a SCRIPT tag on any web page. When combined with standard anchor tags and familiar iconography, this Javascript will automatically open a window of the appropriate size when clicked. Below this text is a code block containing four examples of HTML code for different intents: Reply, Retweet, and Favorite. The "Meet the Web Intents" section follows, with sub-sections for "Tweet Intents" (Reply to a Tweet, Retweet a Tweet, Favorite a Tweet) and "User Intents" (Follow or Display a User). The "Tweet or Reply to a Tweet" section provides the URL <https://twitter.com/intent/tweet> and explains that this intent makes it easy for a user to tweet nearly anything from your site with a stylish #newtwitter-inspired Tweet Composer.

Web Intents can be invoked flexibly through a light combination of Javascript and HTML and are meant to be opened in a new window.

The easiest way to use intents is to include this SCRIPT tag on any web page you wish to invoke an intent. If you've already setup the [Tweet Button](#), you're already prepared for Web Intents.

When combined with standard anchor tags and [familiar iconography](#) like the examples below, this Javascript will automatically open a window of the appropriate size when clicked. You only need to load `platform.twitter.com/widgets.js` once.

```
1. <script type="text/javascript" src="//platform.twitter.com/widgets.js"></script>
2. <p><a href="https://twitter.com/intent/tweet?in_reply_to=51113028241989632">Reply</a></p>
3. <p><a href="https://twitter.com/intent/retweet?tweet_id=51113028241989632">Retweet</a></p>
4. <p><a href="https://twitter.com/intent/favorite?tweet_id=51113028241989632">Favorite</a>
   </p>
```

## Meet the Web Intents

### Tweet Intents

- [Tweet or Reply to a Tweet](#)
- [Retweet a Tweet](#)
- [Favorite a Tweet](#)

### User Intents

- [Follow or Display a User](#)

### Tweet or Reply to a Tweet

<https://twitter.com/intent/tweet>

This intent makes it easy for a user to tweet nearly anything from your site with a stylish #newtwitter-inspired Tweet Composer. Though easy to implement with minimal configuration, you can provide optional parameters to support a variety of scenarios including allowing for careful pre-preparation of Tweet text, replying @mentioning link sharing, and

# JavaScript Interfaces for Twitter for Websites

The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/intents/events>. The page title is "JavaScript Interfaces for Twitter for Websites". The navigation bar includes "Developers", "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible. The main content area features a breadcrumb trail: "Home → Documentation → Web Intents". Below the title, there are buttons for "View" and "What links here". A timestamp indicates the page was updated on Saturday, 2013-07-13 at 15:30. The text explains that by using Twitter JavaScript, users agree to the Developer Rules of the Road. It describes how to integrate Twitter products like Tweet buttons and Follow buttons using the `widgets.js` file. A code block shows the initialization of the `window.twttr` object. The page also includes a "Related open issues" section with a link to a GitHub issue and a "Tags" section listing "Web Intents (24)", "Twitter for Websites (32)", "Javascript (46)", and "analytics (4)".

JavaScript Interfaces for Twitter for Websites

Home → Documentation → Web Intents

View What links here

Updated on Sat, 2013-07-13 15:30

By using Twitter Javascript, you agree to the [Developer Rules of the Road](#).

If you're integrating your site with Twitter using [Twitter for Websites](#) and [Web Intents](#), you can enhance your application using JavaScript functions and events.

Twitter for Websites products—Tweet buttons, Follow buttons, embedded Tweets and timelines—are all loaded using a JavaScript utility named `widgets.js`. When adding a Twitter widget to your page, this JavaScript file is included in the HTML embed code, or you can directly include `http://platform.twitter.com/widgets.js` in your page, using the following code:

```
1. window.twttr = (function (d,s,id) {
2.   var t, js, fjs = d.getElementsByTagName(s)[0];
3.   if (d.getElementById(id)) return; js=d.createElement(s); js.id=id;
4.   js.src="https://platform.twitter.com/widgets.js"; fjs.parentNode.insertBefore(js, fjs);
5.   return window.twttr || (t = { _e: [], ready: function(f){ t._e.push(f) } });
6. })(document, "script", "twitter-wjs");
```

By default, `widgets.js` will find mark-up in a page and convert basic, functional mark-up into rich interactive widgets. In addition, there are a number of functions of `widgets.js` that allow developers to work with Twitter content dynamically, after the page has loaded:

- [Late initialization of widgets](#)
- [Factory methods for dynamic widgets](#)

... ..

Web Intents

Related open issues

- [data.target not set on the 'tweet' web intent event if the tweet button was created by createShareButton](#)

Tags

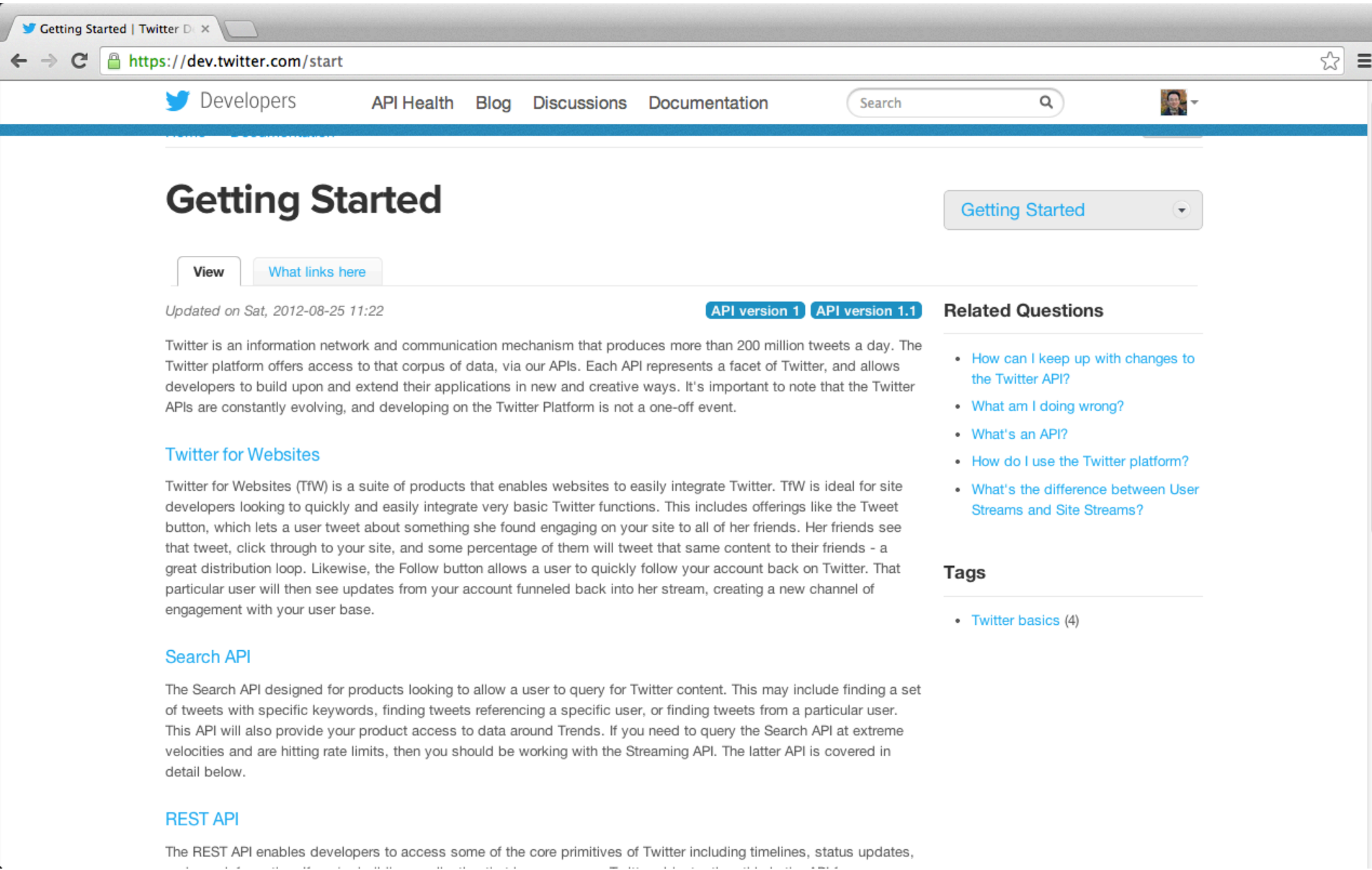
- [Web Intents](#) (24)
- [Twitter for Websites](#) (32)
- [Javascript](#) (46)
- [analytics](#) (4)

<https://dev.twitter.com/docs/intents/events>

# JavaScript Interfaces for Twitter for Websites

```
1. window.twttr = (function (d,s,id) {
2.   var t, js, fjs = d.getElementsByTagName(s)[0];
3.   if (d.getElementById(id)) return; js=d.createElement(s); js.id=id;
4.   js.src="https://platform.twitter.com/widgets.js"; fjs.parentNode.insertBefore(js, fjs);
5.   return window.twttr || (t = { _e: [], ready: function(f){ t._e.push(f) } });
6. })(document, "script", "twitter-wjs");
```

# Twitter API Getting Started



The screenshot shows a web browser window with the URL <https://dev.twitter.com/start>. The page title is "Getting Started | Twitter D...". The navigation bar includes "Developers", "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible. The main content area features a large "Getting Started" heading, a "View" button, and a "What links here" button. The text is updated on Saturday, 2012-08-25 at 11:22. There are two buttons for "API version 1" and "API version 1.1". The main text describes Twitter as an information network and communication mechanism that produces more than 200 million tweets a day. It mentions the Twitter platform offers access to that corpus of data, via our APIs. Each API represents a facet of Twitter, and allows developers to build upon and extend their applications in new and creative ways. It's important to note that the Twitter APIs are constantly evolving, and developing on the Twitter Platform is not a one-off event.

## Twitter for Websites

Twitter for Websites (TfW) is a suite of products that enables websites to easily integrate Twitter. TfW is ideal for site developers looking to quickly and easily integrate very basic Twitter functions. This includes offerings like the Tweet button, which lets a user tweet about something she found engaging on your site to all of her friends. Her friends see that tweet, click through to your site, and some percentage of them will tweet that same content to their friends - a great distribution loop. Likewise, the Follow button allows a user to quickly follow your account back on Twitter. That particular user will then see updates from your account funneled back into her stream, creating a new channel of engagement with your user base.

## Search API

The Search API designed for products looking to allow a user to query for Twitter content. This may include finding a set of tweets with specific keywords, finding tweets referencing a specific user, or finding tweets from a particular user. This API will also provide your product access to data around Trends. If you need to query the Search API at extreme velocities and are hitting rate limits, then you should be working with the Streaming API. The latter API is covered in detail below.

## REST API

The REST API enables developers to access some of the core primitives of Twitter including timelines, status updates,


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

# Twitter REST API certificate updates

The screenshot shows a web browser window with the address bar containing the URL <https://dev.twitter.com/blog/rest-api-ssl-certificate-updates>. The browser's navigation bar includes back, forward, and refresh buttons. The Twitter website header features the Twitter logo, navigation links for Developers, API Health, Blog, Discussions, and Documentation, a search bar, and a user profile picture. Below the header, a breadcrumb trail reads "Home → Blog → Announcements" and a "Tweet" button is visible. The main content area displays the title "REST API SSL certificate updates" in a large, bold font. Below the title are two tabs: "View" (selected) and "What links here". The post is attributed to "@lfcipriani lfcipriani" and dated "Posted on Tue, 2013-11-19 12:40". The post text explains that as of the end of 2013, browsers and certificate authorities will no longer support 1024-bit RSA certificates. It states that the current SSL certificate on api.twitter.com is signed with an older Verisign G2 root CA certificate, and a new Verisign Class 3 Secure Server CA - G3 certificate will be deployed on December 10th, 2013. The new certificate has a 2048-bit key length. The text advises that all HTTP clients must trust the new root certificate to connect to the API. A list of Verisign root certificates is provided, with a link to <https://www.symantec.com/page.jsp?id=roots>. A reference is made to a "Guide to Connecting with SSL" and a "discussion topic" for further information. The post concludes with an "Update [Dec 10th, 2013]: the new certificates were deployed." At the bottom of the post, there are navigation links: "← Previous blog post", "Announcements", and "Next blog post →".

REST API SSL certificate updates

View What links here

 @lfcipriani lfcipriani  
Posted on Tue, 2013-11-19 12:40

At the end of 2013, all Browsers and Certificate Authorities will no longer support 1024 bits RSA certificates to be compliant to National Institute of Standards and Technology (NIST) [guidelines](#).

The SSL certificate currently used on api.twitter.com is signed with the older Verisign G2 root CA certificate.

Due to NIST guidelines, api.twitter.com will change to a new certificate on Dec 10th, 2013. The new certificate will be signed with VeriSign Class 3 Secure Server CA - G3, which has the 2048 bits key length needed to meet recommended security levels.

This means that all HTTP clients used by your application must trust the new root certificate, otherwise you won't be able to connect in the API. To ensure proper SSL certificate verification across all of Twitter's services, your software should include all Verisign Root Certificates in its CAFile or other respective keystore. The root certificates are available at the following link:

- Verisign (<https://www.symantec.com/page.jsp?id=roots>)

For more guidelines on using SSL with the Twitter API, see our [Guide to Connecting with SSL](#). If you're continuing to have issues with the transition, you can join in on this [discussion topic](#).

**Update [Dec 10th, 2013]:** the new certificates were deployed.

← Previous blog post      Announcements      Next blog post →

<https://dev.twitter.com/blog/rest-api-ssl-certificate-updates>

# Connecting to Twitter API using SSL

The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/security/using-ssl>. The page title is "Connecting to Twitter API using SSL". The navigation bar includes "Developers", "API Health", "Blog", "Discussions", and "Documentation". A search bar and a user profile picture are also visible. The breadcrumb trail is "Home → Documentation → Getting Started → Security Best Practices". There is a "Tweet" button and a "Getting Started" dropdown menu. The page content includes a "View" button, a "What links here" button, and a timestamp "Updated on Fri, 2013-10-11 04:08". The main text describes the preferred method of connecting to the Twitter API over HTTPS and discusses OAuth, cipher selection, and verification. A "Tags" section lists "Security (8)", "ssl (16)", and "best practices (9)".

Connecting to Twitter API

← → ↻ <https://dev.twitter.com/docs/security/using-ssl> ☆ ☰

Developers API Health Blog Discussions Documentation Search [Profile Picture]

Home → Documentation → Getting Started → Security Best Practices [Tweet]

## Connecting to Twitter API using SSL

[View] [What links here] [Getting Started]

Updated on Fri, 2013-10-11 04:08 [API version 1] [API version 1.1] Tags

- Security (8)
- ssl (16)
- best practices (9)

The preferred method of connecting to the Twitter API is over HTTPS. While both HTTPS and Non-HTTPS connections to the Twitter API are supported, communicating over SSL preserves user privacy by protecting information between the user and the Twitter API as it travels across the public Internet.

### Using OAuth isn't enough

While OAuth is mandated and protects the user from having their password captured in transit by substituting an OAuth token for the user's credentials, it's not enough to ensure complete privacy.

### Cipher Selection

Twitter's servers and your client will negotiate a cipher spec upon connection. When possible, it's best to use the Twitter supplied cipher default (currently RC4) for session encryption. While other ciphers may offer better performance or security (and may be supported by both your client and Twitter's servers,) the preferred cipher as negotiated by our servers is typically the best available for communication. We do not recommend overriding the negotiated selection in your code.

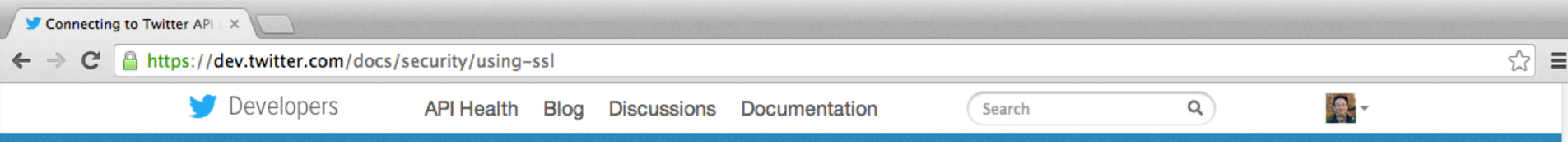
### Verification

#### Validating and/or "pinning" the Twitter SSL Certificate in your code

Twitter's SSL Certificates for `api.twitter.com` are signed by Verisign. For Assets, such as those on `si0.twimg.com` (through `si4.twimg.com`), those certificates are signed by Verisign and/or Contendo depending on the geographically closest CDN server.

Your application should ensure that the certificate chain returned for the all Twitter servers is signed by one of our approved vendors (Verisign EV for `twitter.com`, Versign for `api.twitter.com`, and DigiCert for others) and not other CA roots.

# Connecting to Twitter API using SSL



## Verification

Validating and/or "pinning" the Twitter SSL Certificate in your code

Twitter's SSL Certificates for api.twitter.com are signed by Verisign. For Assets, such as those on si0.twimg.com (through si4.twimg.com), those certificates are signed by Verisign and/or Contendo depending on the geographically closest CDN server.

Your application should ensure that the certificate chain returned for the all Twitter servers is signed by one of our approved vendors (Verisign EV for twitter.com, Verisign for api.twitter.com, and Digicert for others) and not other CA roots.

As of this writing, api.twitter.com's certificate is:

```
1. Certificate:
2.   Data:
3.     Version: 3 (0x2)
4.     Serial Number:
5.       72:bf:38:3a:9e:11:3c:1b:13:90:8e:1a:9f:60:2c:ae
6.     Signature Algorithm: sha1WithRSAEncryption
7.     Issuer: C=US, O=VeriSign, Inc., OU=VeriSign Trust Network, OU=Terms of use at
      https://www.verisign.com/rpa (c)09, CN=VeriSign Class 3 Secure Server CA - G2
8.     Validity
9.       Not Before: May  2 00:00:00 2012 GMT
10.      Not After : May  3 23:59:59 2013 GMT
11.     Subject: C=US, ST=California, L=San Francisco, O=Twitter, Inc., OU=Twitter
      Security, CN=api.twitter.com
```

As of this writing, api.twitter.com is signed by the *VeriSign Class 3 Secure Server CA - G2* root certificate with keyid:A5:EF:0B:11:CE:C0:41:03:A3:4A:65:90:48:B2:1C:E0:57:2D:7D:47

Your code should trust both the Verisign "G2" and "G3" root certificates. After May 3rd, 2013, Twitter's certificates will be signed by the G3 root as Verisign is deprecating the G2 root. Prepare for this by trusting the correct root certificates.

Validate against the minimum number of root certificates

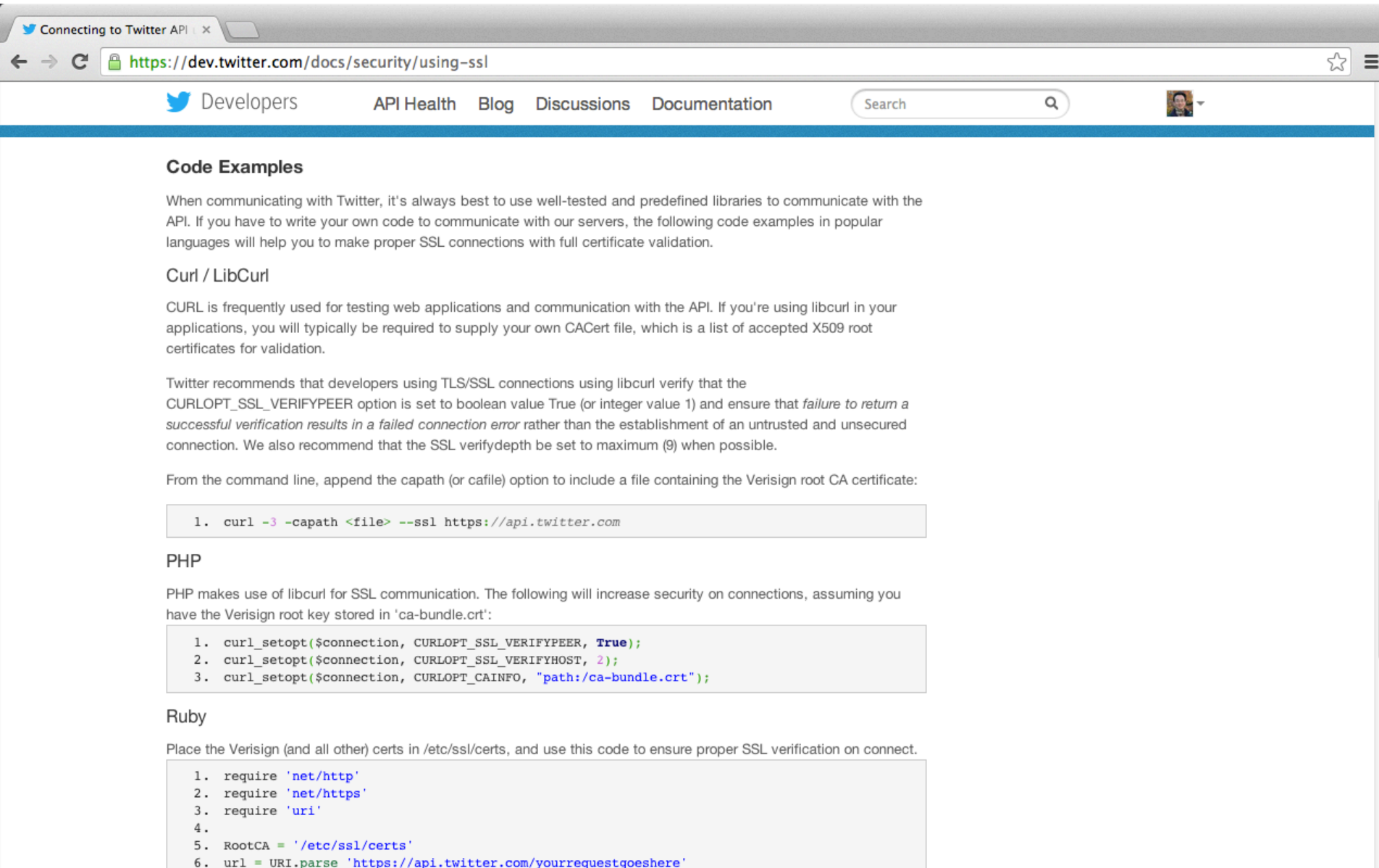
Don't rely on the local operating system to validate the certificate if possible. This can be tampered with by malware, local IT staff, or other bad actors. Validate against the known vendors for api.twitter.com as listed above. Don't include more certificates in your application's trusted CA Root store from vendors that Twitter hasn't listed.



# Connecting to Twitter API using SSL

```
1. Certificate:
2.   Data:
3.     Version: 3 (0x2)
4.     Serial Number:
5.       72:bf:38:3a:9e:11:3c:1b:13:90:8e:1a:9f:60:2c:ae
6.     Signature Algorithm: sha1WithRSAEncryption
7.     Issuer: C=US, O=VeriSign, Inc., OU=VeriSign Trust Network, OU=Terms of use at
https://www.verisign.com/rpa (c)09, CN=VeriSign Class 3 Secure Server CA - G2
8.     Validity
9.       Not Before: May  2 00:00:00 2012 GMT
10.      Not After  : May  3 23:59:59 2013 GMT
11.     Subject: C=US, ST=California, L=San Francisco, O=Twitter, Inc., OU=Twitter
Security, CN=api.twitter.com
```

# Connecting to Twitter API using SSL



Connecting to Twitter API

← → ↻ <https://dev.twitter.com/docs/security/using-ssl> ☆ ☰

Developers API Health Blog Discussions Documentation Search

## Code Examples

When communicating with Twitter, it's always best to use well-tested and predefined libraries to communicate with the API. If you have to write your own code to communicate with our servers, the following code examples in popular languages will help you to make proper SSL connections with full certificate validation.

### Curl / LibCurl

CURL is frequently used for testing web applications and communication with the API. If you're using libcurl in your applications, you will typically be required to supply your own CACert file, which is a list of accepted X509 root certificates for validation.

Twitter recommends that developers using TLS/SSL connections using libcurl verify that the `CURLOPT_SSL_VERIFYPEER` option is set to boolean value `True` (or integer value `1`) and ensure that *failure to return a successful verification results in a failed connection error* rather than the establishment of an untrusted and unsecured connection. We also recommend that the `SSL verifydepth` be set to maximum (`9`) when possible.

From the command line, append the `capath` (or `cafile`) option to include a file containing the Verisign root CA certificate:

```
1. curl -3 -capath <file> --ssl https://api.twitter.com
```

### PHP

PHP makes use of libcurl for SSL communication. The following will increase security on connections, assuming you have the Verisign root key stored in 'ca-bundle.crt':

```
1. curl_setopt($connection, CURLOPT_SSL_VERIFYPEER, True);
2. curl_setopt($connection, CURLOPT_SSL_VERIFYHOST, 2);
3. curl_setopt($connection, CURLOPT_CAINFO, "path:/ca-bundle.crt");
```

### Ruby

Place the Verisign (and all other) certs in `/etc/ssl/certs`, and use this code to ensure proper SSL verification on connect.

```
1. require 'net/http'
2. require 'net/https'
3. require 'uri'
4.
5. RootCA = '/etc/ssl/certs'
6. url = URI.parse 'https://api.twitter.com/yourrequestgoeshere'
```

# Connecting to Twitter API using SSL

From the command line, append the `capath` (or `cafile`) option to include a file containing the Verisign root CA certificate:

```
1. curl -3 -capath <file> --ssl https://api.twitter.com
```

## PHP

PHP makes use of libcurl for SSL communication. The following will increase security on connections, assuming you have the Verisign root key stored in 'ca-bundle.crt':

```
1. curl_setopt($connection, CURLOPT_SSL_VERIFYPEER, True);
2. curl_setopt($connection, CURLOPT_SSL_VERIFYHOST, 2);
3. curl_setopt($connection, CURLOPT_CAINFO, "path:/ca-bundle.crt");
```

## Ruby

Place the Verisign (and all other) certs in `/etc/ssl/certs`, and use this code to ensure proper SSL verification on connect.

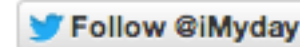
```
1. require 'net/http'
2. require 'net/https'
3. require 'uri'
4.
5. RootCA = '/etc/ssl/certs'
6. url = URI.parse 'https://api.twitter.com/yourrequestgoeshere'
7. http = Net::HTTP.new(url.host, url.port)
8. http.ca_path = RootCA
9. http.verify_mode = OpenSSL::SSL::VERIFY_PEER
10. http.verify_depth = 9
11. request = Net::HTTP::Get.new(url.path)
12. # handle oauth here, or whatever you need to do...
13. response = http.request(request)
14.
15. # ... process response ...
```

## Python

See this [Stack Overflow post regarding SSL verification](#) under Python. Depending on the Python version and SSL Library you are using, verification may be supported natively in the library, or you may have to extract the certificate from the connection and perform additional verification steps.

# Summary

- **Twitter Developers**
  - Twitter Platform Objects
- **Twitter for Websites**
- **Twitter Search API**
- **Twitter REST API**
- **Twitter Streaming API**



# References

- Twitter Developers, <https://dev.twitter.com/>