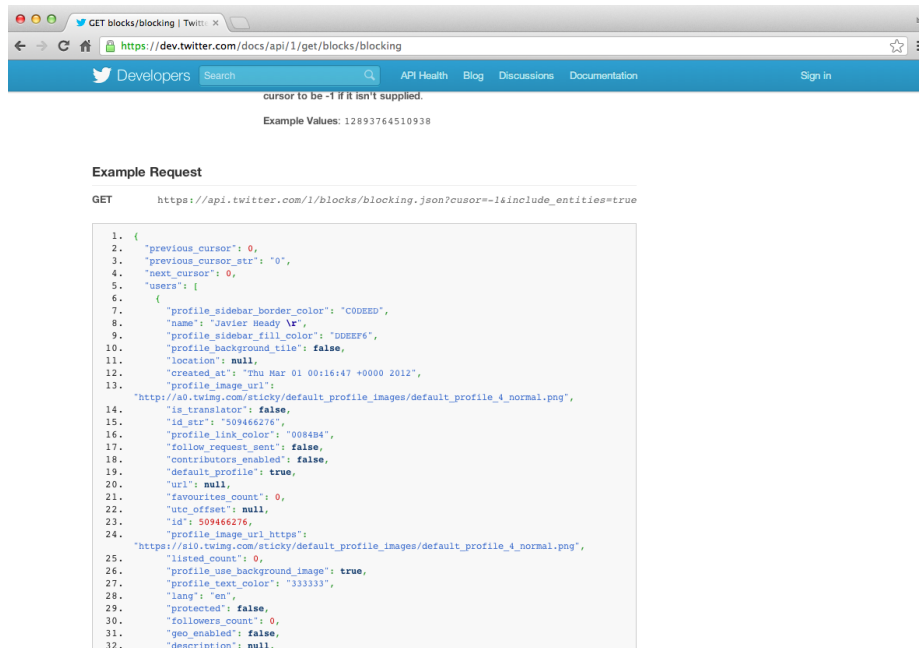




Reading data from APIs

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Application programming interfaces

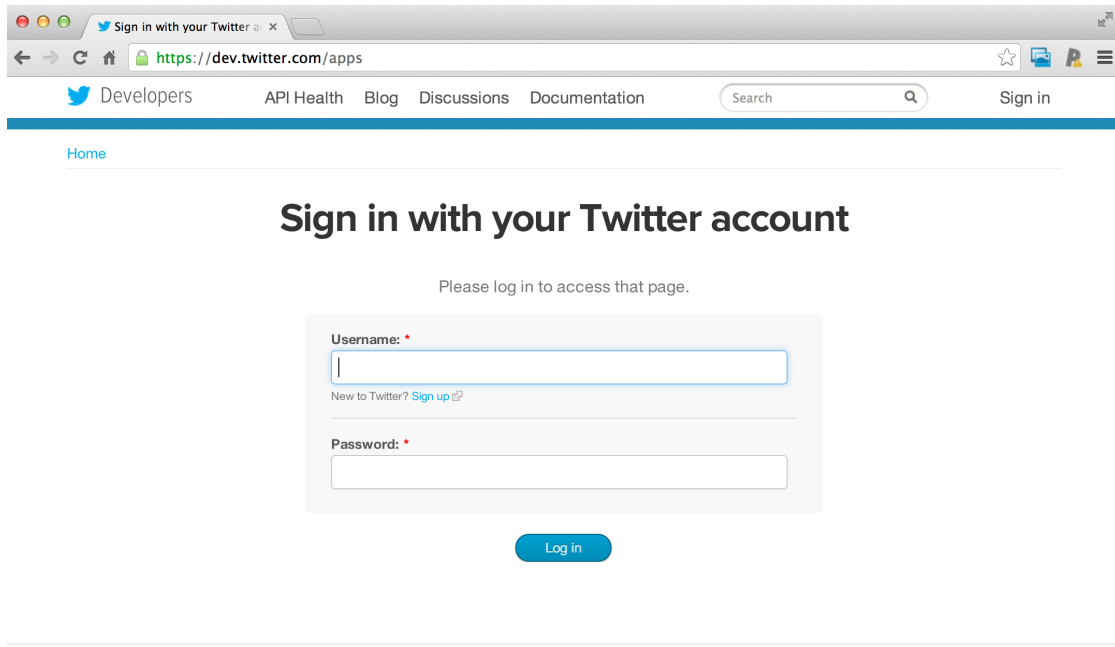


The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/api/1/get/blocks/blocking>. The page is titled "GET blocks/blocking | Twitter" and includes a search bar and navigation links like "API Health", "Blog", "Discussions", "Documentation", and "Sign in". Below the header, there is a note: "cursor to be -1 if it isn't supplied." and "Example Values: 12893764510938". The main content area is titled "Example Request" and shows a GET request to the endpoint: `https://api.twitter.com/1/blocks/blocking.json?cursor=-1&include_entities=true`. Below the request, there is a JSON response snippet with line numbers 1 through 32. The JSON object contains fields for "previous_cursor", "next_cursor", "users", and "profile_image_url". The "users" array contains one user object with various profile details.

```
1. {
2.   "previous_cursor": 0,
3.   "previous_cursor_str": "0",
4.   "next_cursor": 0,
5.   "users": [
6.     {
7.       "profile_sidebar_border_color": "COBEEED",
8.       "name": "Javier Heady \r",
9.       "profile_sidebar_fill_color": "DDEEFF",
10.      "profile_background_tile": false,
11.      "location": null,
12.      "created_at": "Thu Mar 01 00:16:47 +0000 2012",
13.      "profile_image_url":
14.        "http://a0.twimg.com/sticky/default_profile_images/default_profile_4_normal.png",
15.      "is_translator": false,
16.      "id_str": "509466276",
17.      "profile_link_color": "0084B4",
18.      "follow_request_sent": false,
19.      "contributors_enabled": false,
20.      "default_profile": true,
21.      "url": null,
22.      "favourites_count": 0,
23.      "utc_offset": null,
24.      "id": 509466276,
25.      "profile_image_url_https":
26.        "https://s10.twimg.com/sticky/default_profile_images/default_profile_4_normal.png",
27.      "listed_count": 0,
28.      "profile_use_background_image": true,
29.      "profile_text_color": "333333",
30.      "lang": "en",
31.      "protected": false,
32.      "followers_count": 0,
33.      "geo_enabled": false,
34.      "description": null,
```

<https://dev.twitter.com/docs/api/1/get/blocks/blocking>

Creating an application



The screenshot shows a web browser window with the address bar displaying `https://dev.twitter.com/apps`. The page title is "Sign in with your Twitter" and the URL is `https://dev.twitter.com/apps`. The navigation bar includes links for "Developers", "API Health", "Blog", "Discussions", and "Documentation", along with a search bar and a "Sign in" button. The main content area features a "Home" link and a large heading "Sign in with your Twitter account". Below this, a message states "Please log in to access that page." and a login form is displayed. The form has two input fields: "Username: *" and "Password: *". The "Username" field is currently empty. Below the "Username" field, there is a link "New to Twitter? Sign up" with an external link icon. The "Password" field is also empty. A blue "Log in" button is located below the password field.

Sign in with your Twitter

[https://dev.twitter.com/apps](#)

Developers API Health Blog Discussions Documentation Search Sign in

[Home](#)

Sign in with your Twitter account

Please log in to access that page.

Username: *

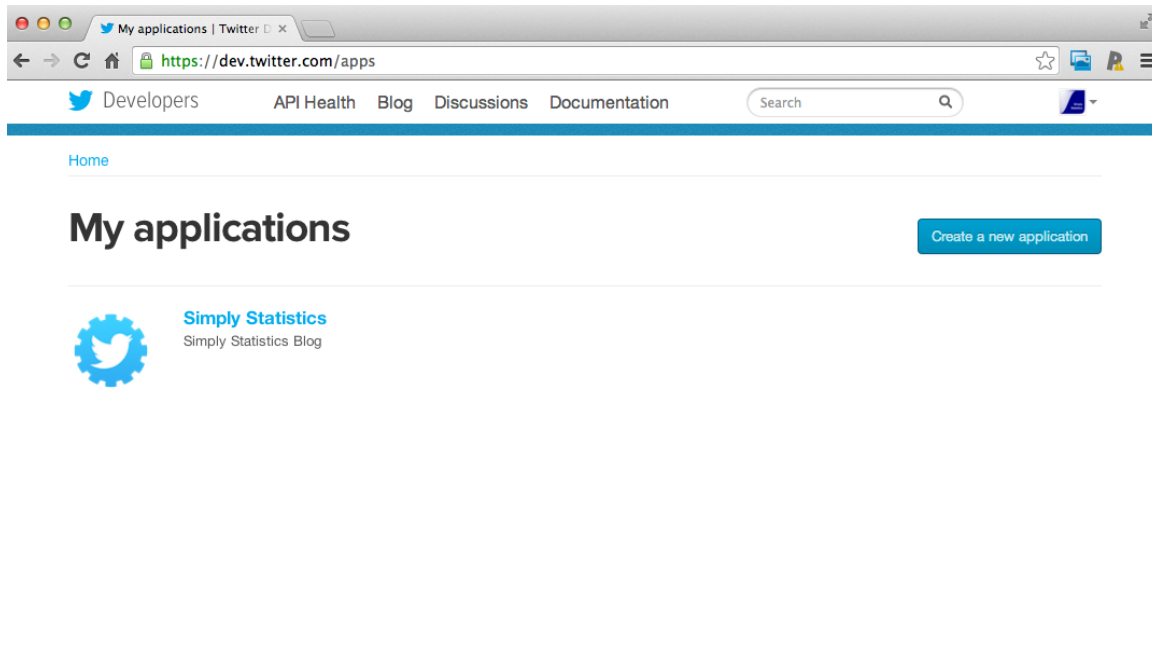
New to Twitter? [Sign up](#)

Password: *

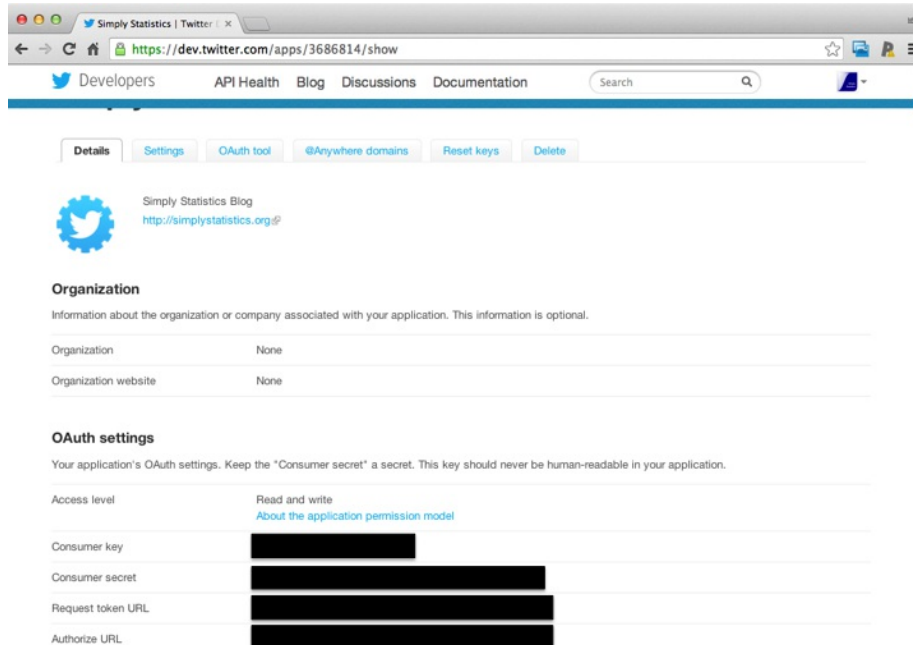
Log in

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


Creating an application



Creating an application



The screenshot shows a web browser window with the URL `https://dev.twitter.com/apps/3686814/show`. The page title is 'Simply Statistics | Twitter'. The navigation bar includes links for 'Developers', 'API Health', 'Blog', 'Discussions', and 'Documentation', along with a search bar. The main content area has tabs for 'Details', 'Settings', 'OAuth tool', '@Anywhere domains', 'Reset keys', and 'Delete'. The 'Details' tab is active, showing the application's profile picture (a blue gear with a Twitter bird), name 'Simply Statistics Blog', and website 'http://simplystatistics.org/'. Below this is the 'Organization' section, which is currently empty. The 'OAuth settings' section is also visible, showing the access level as 'Read and write' and a link to 'About the application permission model'. The OAuth settings table lists the consumer key, consumer secret, request token URL, and authorize URL, all of which are redacted with black bars.

Details	
	
Simply Statistics Blog http://simplystatistics.org/	
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 and write About the application permission model
Consumer key	[Redacted]
Consumer secret	[Redacted]
Request token URL	[Redacted]
Authorize URL	[Redacted]

Accessing Twitter from R

```
myapp = oauth_app("twitter",  
                  key="yourConsumerKeyHere", secret="yourConsumerSecretHere")  
sig = sign_oauth1.0(myapp,  
                    token = "yourTokenHere",  
                    token_secret = "yourTokenSecretHere")  
homeTL = GET("https://api.twitter.com/1.1/statuses/home_timeline.json", sig)
```

Converting the json object

```
json1 = content(homeTL)
json2 = jsonlite::fromJSON(toJSON(json1))
json2[1,1:4]
```

	created_at	id	id_str
1	Mon Jan 13 05:18:04 +0000 2014	4.225984e+17	422598398940684288

1 Now that P. Norvig's regex golf IPython notebook hit Slashdot, let's see if our traffic spike tops th

How did I know what url to use?

The screenshot shows a web browser window with the URL `https://dev.twitter.com/docs/api/1.1/get/statuses/home_timeline`. The page is titled "GET statuses/home_timeline" and is part of the Twitter API documentation. It includes a "View" button and a "What links here" button. The page content describes the endpoint, its purpose, and provides example values for parameters like `count` and `since_id`. A sidebar on the right contains "Resource Information" and "OAuth tool" sections.

Developers API Health Blog Discussions Documentation Search

Home → Documentation → REST API Tweet

GET statuses/home_timeline

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

Updated on Wed, 2012-09-05 10:06 API version 1.1

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 many users or follow users who tweet frequently.

See [Working with Timelines](#) for instructions on traversing timelines efficiently.

Resource URL

`https://api.twitter.com/1.1/statuses/home_timeline.json`

Parameters

count optional	Specifies the number of records to retrieve. Must be less than or equal to 200. Defaults to 20. Example Values: 5
since_id optional	Returns results with an ID greater than (that is, more recent than) the specified ID. There are limits to the number of Tweets which can be accessed through the API. If the limit of Tweets has occurred since the <code>since_id</code> , the <code>since_id</code> will be forced to the oldest ID available. Example Values: 1332

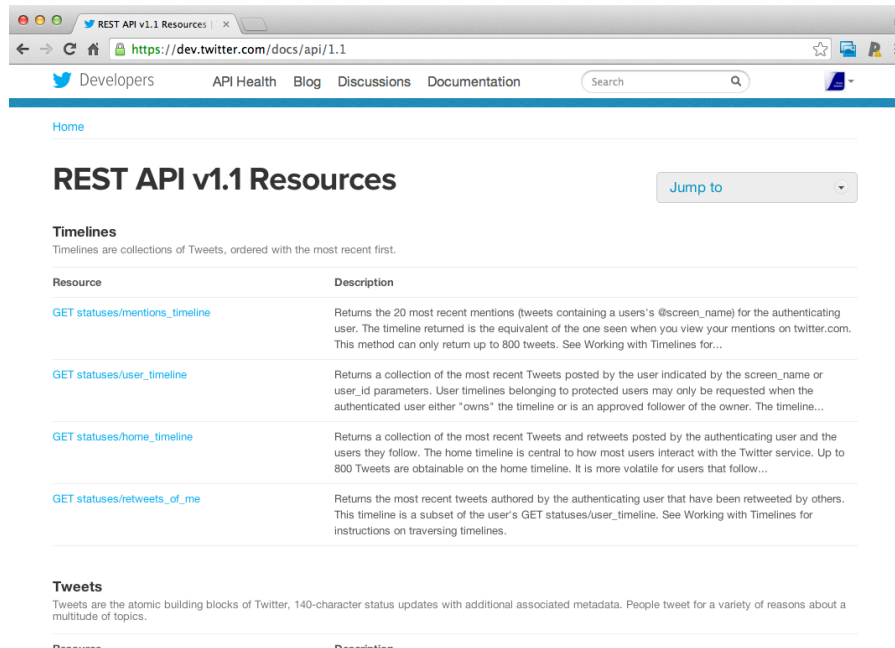
Resource Information

Rate Limited?	Yes
Requests per rate limit window	15/user
Authentication	Requires user context
Response Formats	json
HTTP Methods	GET
Resource family	statuses
Response Object	Tweets
API Version	v1.1

OAuth tool

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

In general look at the documentation



The screenshot shows a web browser window with the URL <https://dev.twitter.com/docs/api/1.1>. The page title is "REST API v1.1 Resources". Below the title, there is a "Jump to" dropdown menu. The main content is divided into two sections: "Timelines" and "Tweets". The "Timelines" section includes a table with two columns: "Resource" and "Description". The table lists four resources: "GET statuses/mentions_timeline", "GET statuses/user_timeline", "GET statuses/home_timeline", and "GET statuses/retweets_of_me". Each resource has a corresponding description. The "Tweets" section is partially visible at the bottom of the screenshot.

REST API v1.1 Resources

Jump to

Timelines

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

Resource	Description
GET statuses/mentions_timeline	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...
GET statuses/user_timeline	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...
GET statuses/home_timeline	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...
GET statuses/retweets_of_me	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
----------	-------------

<https://dev.twitter.com/docs/api/1.1/overview>

In general look at the documentation

- htrr allows GET, POST, PUT, DELETE requests if you are authorized
- You can authenticate with a user name or a password
- Most modern APIs use something like oauth
- htrr works well with Facebook, Google, Twitter, Github, etc.