Data Exports: Twitter

Page
Page-level metrics will give you data aggregated at a daily level, so you can look back and see what trends are occurring on your Page over time.

Tweet
Tweet-level metrics show the scope and strength of influence of a tweet, which gives you the ability to analyse the content you're delivering and how well it is resonating with not only your community, but also the potential audience accessible through your community.

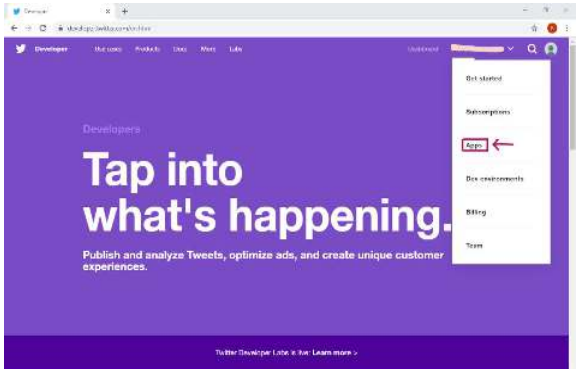
Hashtag
Hashtag-level metrics indicate the amount of attention and engagement a certain topic/event is able to evoke.

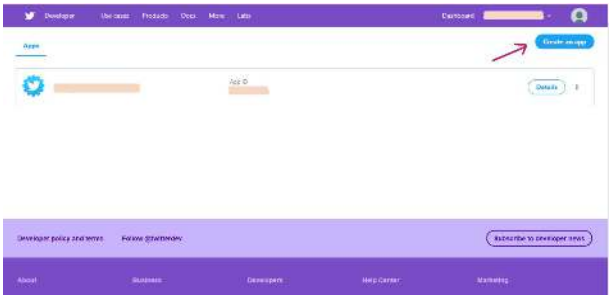
Media
Video-level metrics helps you understand how media contents contribute to your overall reach and engagement.

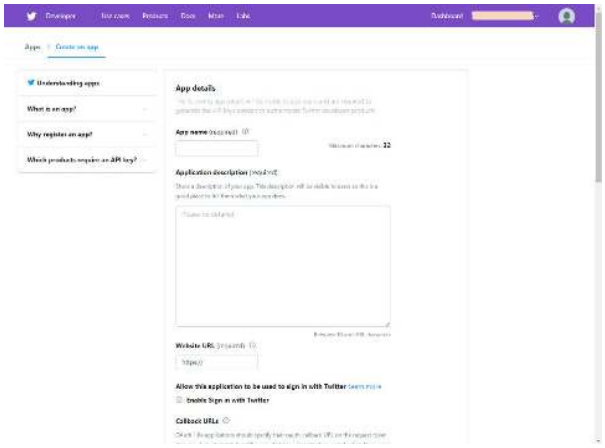
How do I extract tweets and comments?

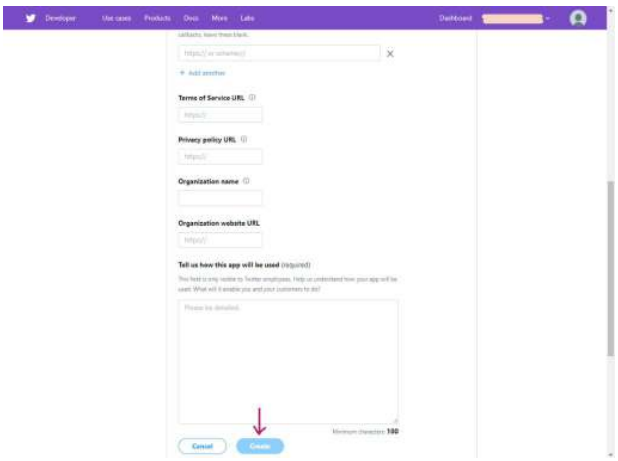
Twitter comments, as well as other Twitter data, can be extracted by making use of a Twitter developer's account and R.

First, apply for a Twitter developer account at *https://developer.twitter.com* and complete the required steps in the application. Once approved, visit your developers page and create an app.



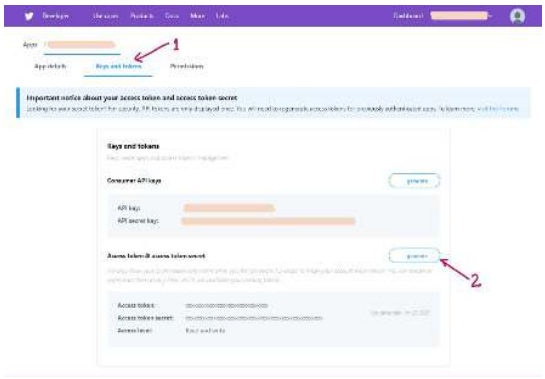






Now you have a new app! You can always find it on the apps page (developer.twitter.com/en/apps) and click on the “Details” icon to view.

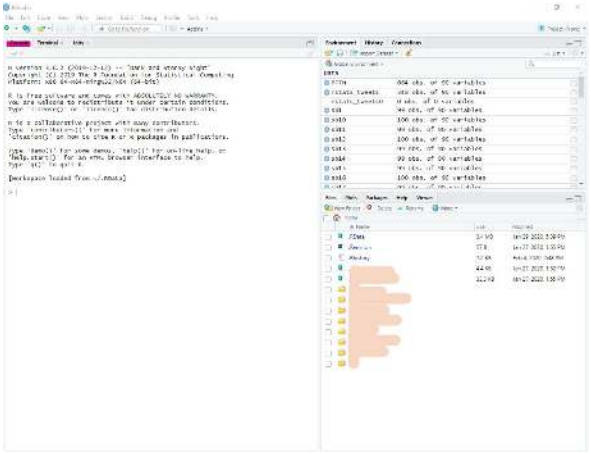
Finally, collect the Application Programming Interface (API) keys under Keys and tokens. Generate your access token and access token secret. All the four keys on this page will be used, so keep them somewhere.



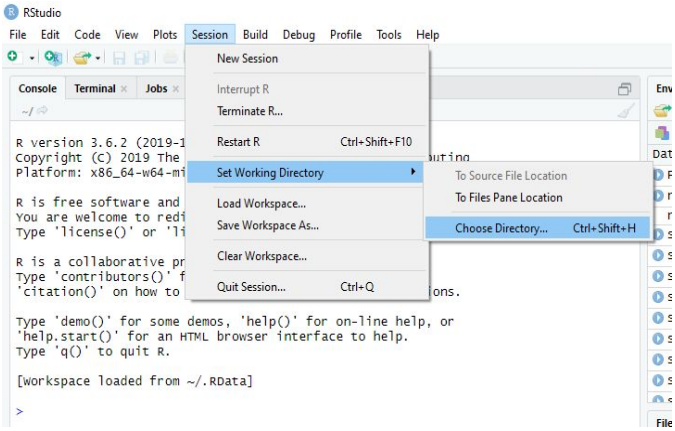
Build a connection between R and Twitter

Install R and R studio first on your device. Both are free!

Open R studio. Now, you want to make use of your Twitter developer’s status through R commands, so you give R access to your developer’s account.



*This is the R studio interface. The “Console" tab is where you type your commands and get R’s feedback. But first, to make your future work easier, set up a work directory where you intend to store all related documents.



Now the basic set-ups are done and you can start dealing with Twitter. Click in the console and start typing your commands. To finish and run a command, hit the Enter key.

Install package “rtweet” and load it

install.packages("rtweet")

library(rtweet)

Store all the API keys

api_key <- "..."

api_secret_key <- "..."

access_token <- "..."

access_token_secret <- "..."

#replace the ... with corresponding keys and tokens.

Authenticate the connection

token <- create_token (

app = "NAMEOFYOURAPP", #use the name you entered for your app

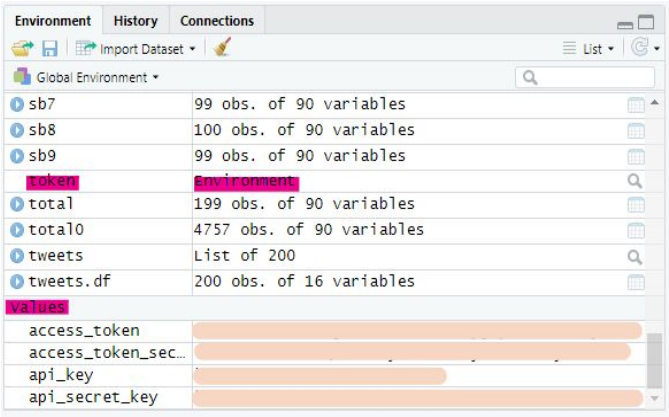
consumer_key = api_key,

consumer_secret = api_secret_key,

access_token = access_token,

access_secret = access_token_secret)

This will create an environment in the workspace called “token”, and you should see the values of your API keys in the “Environment” tab in the upper-right part of R studio. That means a connection between R and Twitter has been built successfully!



Extract relevant tweets (regular approaches)

To search for tweets that are relevant to a topic (e.g. according to hashtags)

DF <-search_tweets (q="YOURKEYWORD", n=NUMBER OF TWEETS)

You’ll have a data frame that contains the requested data. But only recent tweets (6-9 days) can be extracted with the standard API product! To extract all it is NECESSARY to update to a more advanced API package. See *advanced approaches*. You can click on the data frame in the “Environment tab” to view the dataset.

Alternatively, to extract tweets and replies from a certain account

DF<-get_timeline(q="@SCREENNAME", n=NUMBER OF TWEETS)

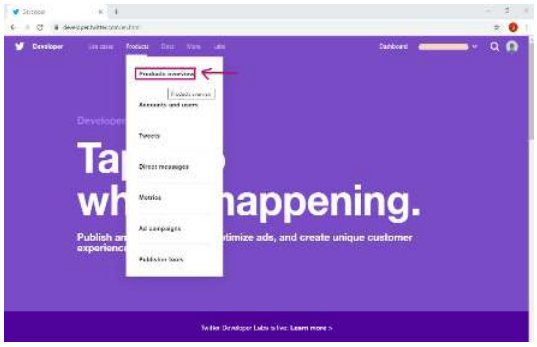
Note that you need to search by the screen name (the one starting with @), NOT the page name.

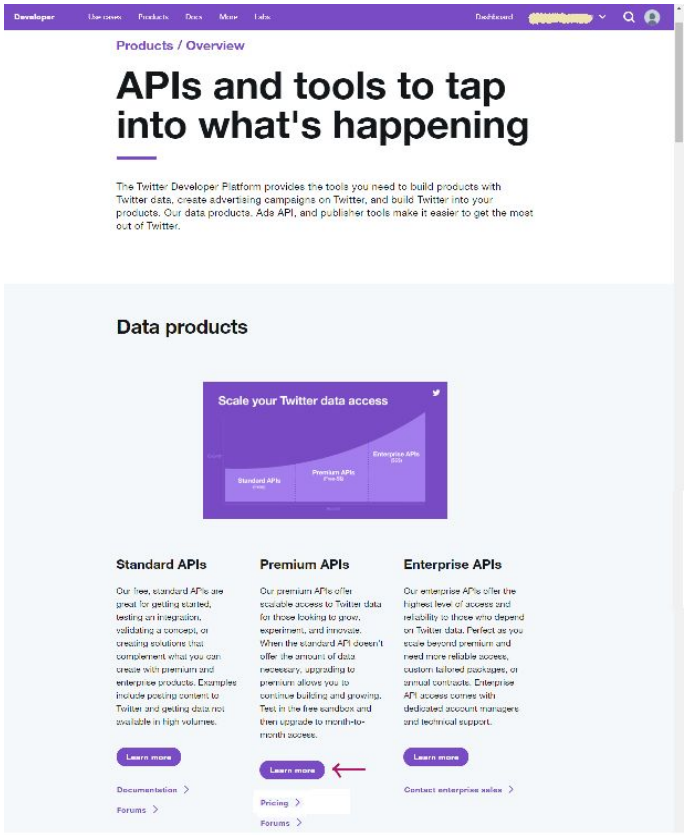
You can set the n to a larger number to make it convenient. However, there's an upper limit of 3,200 tweets.

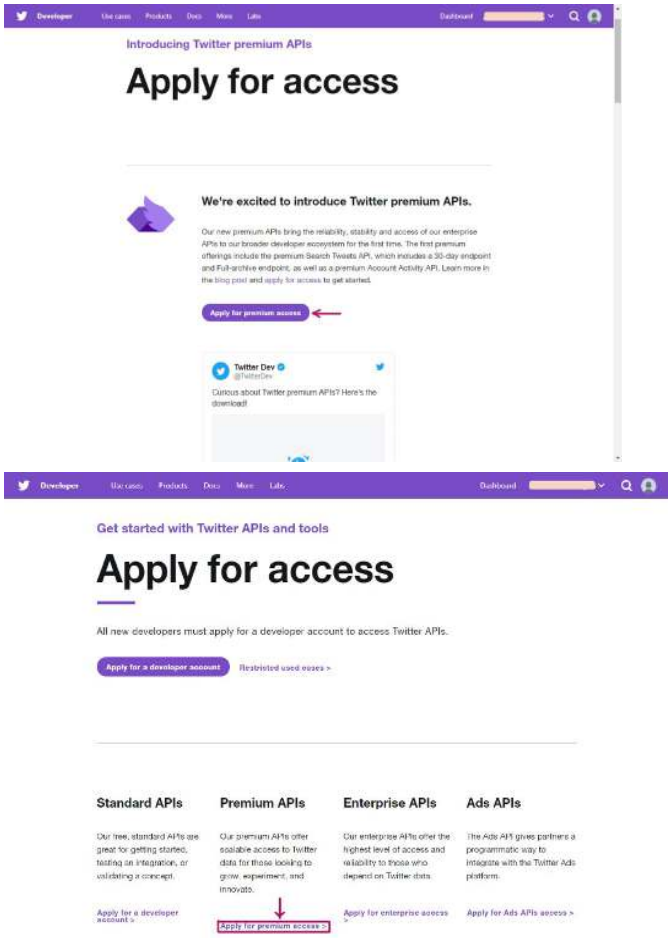
If you want to extract historical comments rather than tweets, there's no direct way to do that with regular approaches. But you can turn to *advanced approaches* in the next section.

Extract comments (advanced approaches)

The regular approaches are handy, but there are obvious limitations when you want to extract comments. The first approach does give you a sheet containing comments, but it has a strict time restriction; the second approach does not allow you to see comments from other users besides the account of interest. Therefore, you may want to upgrade to an advanced developer toolkit. **API Sandbox** can be a good choice because it's FREE (bad news is that it requires more manual input if you expect to have large data). You do this by going to your developer's webpage and click on the *Products* tab. The Sandbox has two versions: 30-Days and full archive. Choose one (or both) according to your needs and give the new dev environment(s) a label (random name).

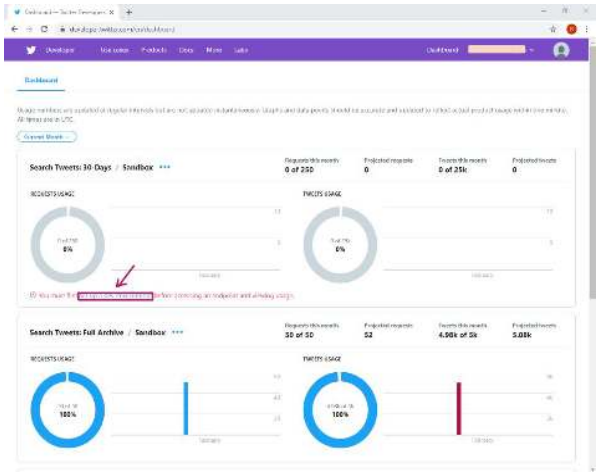

Developer
Dashboard
Products overview
Accounts and users
Tweets
Direct messages
Metrics
Ad campaigns
Publisher tools
Twitter Developer Labs is live. Learn more >

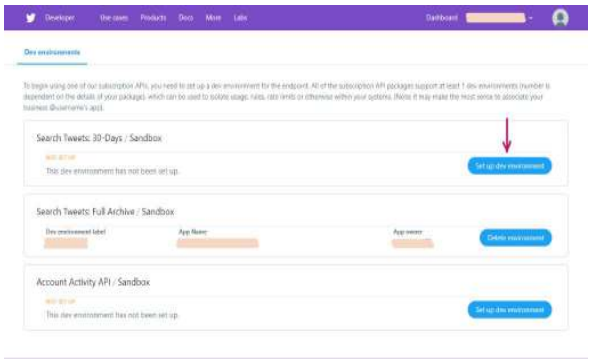

Developer
Use cases
Products
Docs
More
Labs
Dashboard
Products / Overview
APIs and tools to tap into what's happening
The Twitter Developer Platform provides the tools you need to build products with Twitter data, create advertising campaigns on Twitter, and build Twitter into your products. Our data products, Ads API, and publisher tools make it easier to get the most out of Twitter.
Data products
Scale your Twitter data access
Standard APIs
Premium APIs
Enterprise APIs
Standard APIs
Our free, standard APIs are great for getting started, testing an integration, validating a concept, or creating solutions that complement what you can create with premium and enterprise products. Examples include posting content to Twitter and getting data not available in high volumes.
Learn more
Documentation >
Forums >
Premium APIs
Our premium APIs offer scalable access to Twitter data for those looking to grow, experiment, and innovate. When the standard API doesn't offer the amount of data necessary, upgrading to premium allows you to continue building and growing. Test in the free sandbox and then upgrade to month-to-month access.
Learn more
Pricing >
Forums >
Enterprise APIs
Our enterprise APIs offer the highest level of access and reliability to those who depend on Twitter data. Perfect as you scale beyond premium and need more reliable access, custom tailored packages, or annual contracts. Enterprise API access comes with dedicated account managers and technical support.
Learn more
Contact enterprise sales >

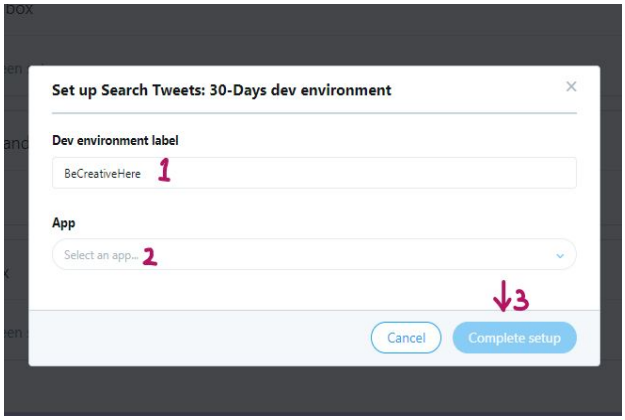


Now that you have chosen your product(s), you can now see what they will enable you to do on your developer's dashboard (*https://developer.twitter.com/en/dashboard*). This is also where you'll see the usage of your products.

To activate a product, set up a dev environment and label it.







The above example is based on Sandbox 30-Days, but you can choose according to your needs and budget. Sandbox is a premium offer, albeit free.

After completing the setup, you will be redirected to the Dashboard and you will notice that the line in red “you must first set up a dev environment..." under your selected product is gone. This means that you have successfully activated the product and are ready to proceed to advanced extraction!

Search through the entire history (full archive)

DF<- search_fullarchive (q="@SCREENNAME", n=100, fromDate = YYYYMMDDHHMM, toDate=YYYYMMDDHHMM, env_name = "YOURLABEL")

 # you can request data 50 times per month, 100 tweets per request, meaning that n has to be no larger than 100.

 # you don't always need to havethe fromDate term in your command, especially when you're going backwards into history from a later date for 100 tweets.

Search through the past month (30-Days)

DF<- search_30day (q="@SCREENNAME", n=100, fromDate = YYYYMMDDHHMM, toDate=YYYYMMDDHHMM, env_name = "YOURLABEL")

You can request data 250 times per month, 100 tweets per request.

Again, n<=100 and fromDate isn't always necessary.

This only gives you data from the past month, and the full archive option can do the same for you. But it's useful when you need past month's data urgently and you know there were more than 5000 tweets where the page was tagged or replied to.

Note that the data you get will contain 1) comments under the page's tweets, 2) retweets of the page's tweets and 3) tweets created by other users where the page is tagged. However, it's not hard to tell which ones are comments and filter out the rest (see *Filtering* section for details).

With the upper limits for the number of tweets you may get per request though, you might have to extract multiple times when there are too many tweets. In that case, you can merge all the datasets into a big, neat one. Here is one of the simple ways to do that.

FINALDATA<-rbind(DF1, DF2,..., DFn)

Transfer data to a csv file

install package "data.table" and load it

Install.packages("data.table")

library(data.table)

use the "fwrite" function

fwrite(FINALDATA, file = "CSVNAME.csv")

This is one way to do things. Regular way of write.csv does not work because the data frame contains list variables.

Alternative way (less efficient):

sapply(DF,class) #see which variables are list-type

unlist(DF$COLUMNNAME) #unlist those variables **OR**

toString(DF$COLUMNNAME) #transfer the variables to string-type

write.csv(DF,"CSVNAME.csv")

The csv file will be automatically saved to your work directory. Now you have your data!