How to Query Twitter Data

In this project, you'll be using **Tweepy** to query Twitter's API for additional data beyond the data included in the WeRateDogs Twitter archive. This additional data will include retweet count and favorite count.

Some APIs are completely open, like MediaWiki (accessed via the **wptools** library) in Lesson 2. Others require authentication. The Twitter API is one that requires users to be authorized to use it. This means that before you can run your API querying code, you need to set up your own Twitter application. And before that, you must sign up for a Twitter account. **This guide** describes the setup process well.* Once you have these set up, the following code, which is provided in the **Getting started** portion of the tweepy documentation, will create an API object that you can use to gather Twitter data.

```python
import tweepy

consumer_key = 'YOUR CONSUMER KEY'
consumer_secret = 'YOUR CONSUMER SECRET'
access_token = 'YOUR ACCESS TOKEN'
access_secret = 'YOUR ACCESS SECRET'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth)
```

Note: If you can't set up an application because of mobile verification issues, please email Rick Gaston (rick@udacity.com).

Tweet data is stored in JSON format by Twitter. Getting tweet JSON data via tweet ID using Tweepy is described well in this **StackOverflow answer**. Note that setting the `tweet_mode` parameter to `'extended'` in the `get_status` call, i.e., `api.get_status(tweet_id, tweet_mode='extended')`, can be useful.

Also, note that the tweets corresponding to a few tweet IDs in the archive may have been deleted. **Try-except blocks** may come in handy here.

Do Not Include Your API Keys, Secrets, and Tokens in Your Submission

Do **not** include your API keys, secrets, and tokens in your project submission. This is standard practice for APIs and public code.

Twitter's Rate Limit

Twitter's API has a rate limit. Rate limiting is used to control the rate of traffic sent or received by a server. As per **Twitter's rate limiting info page**:

> Rate limits are divided into 15 minute intervals

To query all of the tweet IDs in the WeRateDogs Twitter archive, 20-30 minutes of running time can be expected. Printing out each tweet ID after it was queried and **using a code timer** were both helpful for sanity reasons. Setting the `wait_on_rate_limit` and `wait_on_rate_limit_notify` parameters to `True` in the `tweepy.api` class is useful as well.

Writing and Reading Twitter JSON

you will soon assess and clean. This **Reading and Writing JSON to a File in Python** article from Stack Abuse, will be useful.

Twitter logo

NEXT