day_two

February 24, 2016

1 Day 2

A vast amount of data exists on the web and is now publicly available. In this section, we give an overview of popular ways to retrieve data from the web, and walk through some important concerns and considerations.

1.1 Background

** 1) How does the web work? **
** - a) Examining a http request through your browser (Chrome/Firefox) **
** - b) Examining a http request through your console **
** 2) Web terminology: some important distinctions **
** - a) Web scraping vs APIs - what's the difference? **
** - b) Web scrapers vs crawlers & spiders - what's the difference? **
** 3) Building friendly bots: robots.txt and legality **

1.2 Tutorial

** 1) Creating a friendly bot on Wikipedia **
** 2) Twitter API **

2 Background:

2.1 1) How does the web work?

An extremely simplified model of the web is as follows. The World Wide Web is said to follow a client-server architecture, where clients (etc. the web browser on your computer) send requests to servers, and servers respond with resources. When you enter a URL (or Uniform Resource Locator) into your browser, your browser sends a http request with information about the resource you are looking for to a remote server, which the server returns, if available.

A server can be understood as a computer that has various files (resources) stored in its system, and that returns those files if it receives requests in a format it understands.

2.2 1a). Examining a request through your browser (Chrome/Firefox)

You can view the request sent by your browser by:

1) Opening a new tab in your browser

2) Enabling developer tools (**View -> Developer -> Developer Tools in Chrome** and **Tools -> Web Developer -> Toggle Tools in Firefox**)

3) Loading or reloading a web page (etc. www.google.com)

4) Navigating to the Network tab in the panel that appears at the bottom of the page.

2.2.1 Chrome Examine Request Example

2.2.2 Firefox Examine Request Example

These requests you send follow the HTTP protocol (Hypertext Transfer Protocol), part of which defines the information (along with the format) the server needs to receive to return the right resources. Your HTTP request contains **headers**, which contains information that the server needs to know in order to return the right information to you.

2.3 1b). Examining a http request through the console

Let's now try accessing the same server by using requests. Now, instead of sending the server a request through your browser, you are sending the server a request programmatically, through your console. The server returns some output to you, which the requests module parses as a python object.

```
In [1]: import requests

        r = requests.get("http://www.google.com")
```

This response object contains various information about the request you sent to the server, the resources returned, and information about the response the server returned to you, among other information. These are accessible through the **request** attribute, the **content** attribute and the **headers** attribute respectively, which we'll each examine below.

```
In [2]: type(r.request), type(r.content), type(r.headers)
```

```
Out[2]: (requests.models.PreparedRequest,
         bytes,
         requests.structures.CaseInsensitiveDict)
```

Here, we can see that **request** is an object with a custom type, **content** is a str value and **headers** is an object with "dict" in its name, suggesting we can interact with it like we would with a dictionary.

If we recall our simple model of the web, we sent a http request through our console to a remote server, which returned a response. Both the request and response contains information that first allows the server to determine the right resource to return, and then typically, our browser to interpret the returned object.

The content is the actual resource returned to us - let's take a look at the content first before examining the request and response objects more carefully. (We select the first 1000 characters b/c of the display limits of Jupyter/python notebook.)

```
In [3]: from pprint import pprint
        pprint(r.content[0:1000])
```

```
(b'<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang='
 b'"en"><head><meta content="Search the world\'s information, including webp'
 b'ages, images, videos and more. Google has many special features to help you '
 b'find exactly what you\'re looking for." name="description"><meta content='
 b'"noodp" name="robots"><meta content="text/html; charset=UTF-8" http-equiv="C'
 b'ontent-Type"><meta content="/images/branding/googleg/1x/googleg_standard_col'
 b'or_128dp.png" itemprop="image"><title>Google</title><script>(function(){wind'
 b"ow.google={kEI:'RB_OVoeNBom2jwOuqr3wBg',kEXPI:'18168,3700315,3700388,4028790"
 b',4029815,4031109,4032678,4033307,4036509,4036527,4038012,4039268,4042491,404'
 b'2784,4042792,4043492,4045841,4046187,4048909,4049501,4049549,4049557,4050908'
 b',4051241,4051558,4051596,4052304,4054284,4054385,4054551,4055202,4055744,405'
 b'6038,4057169,4057586,4057836,4058117,4058125,4058128,4058316,4058328,4058337'
 b',4058384,4058420,4058624,4058938,4059021,4059043,4059318,4059438,4059635,405'
 b'9767,4059860,4060683')
```

2.3.1 HTML: language for computers

The content returned is written in **HTML (HyperText Markup Language)**, which is the default format in which web pages are returned. The content looks like gibberish at first, with little to no spacing. The reason for this is that this output is not designed for us to read, but for the browser to parse and present in a visual interface.

The HTML raw document contains both the text in the web page, such as "Google Research" or "I'm Feeling Lucky", as well as tags and information about how the text is to be formatted and presented, including positioning, font size and the layout of the site. When we begin writing our web scraper for Wikipedia, we'll go into more detail how to navigate and parse the HTML structure to locate and extract the data you need.

If you save a web page as a ".html" file, and open the file in a text editor like Notepad++ or Sublime Text, this is the same format you'll see. Opening the file in a browser (i.e. by double-clicking it) gives you the Google home page you are familiar with.

Next, let's take a look at the request attribute. Notice that the request attribute is attached to our response object returned from requests.get, i.e. the http request has already been sent and the request attribute is provided for convenience to see what request headers you sent, after-the-fact.

Let's print out the headers associated with our request. The **url** and **method** attribute contains other key information associated with the request. We can see the **headers**, **url** and **method** attributes in the dir, you can also use the **getattr** function or just check to see if a word is in the headers list (if the headers list is too long).

```
In [4]: r.request.headers
```

```
Out[4]: {'Accept': '*/*', 'User-Agent': 'python-requests/2.9.1', 'Accept-Encoding': 'gzip, deflate', 'C
```

2.3.2 Printing information associated with request

```
In [5]: pprint("url: " + r.request.url)
        pprint("method: " + r.request.method)
```

```
'url: http://www.google.com/'
'method: GET'
```

```
In [6]: pprint(r.request.headers.items())
```

```
ItemsView({'Accept': '*/*', 'User-Agent': 'python-requests/2.9.1', 'Accept-Encoding': 'gzip, deflate',
```

The method associated with the request (GET here) is part of a number of other methods defined in the HTTP Protocol, including GET, POST, PUT, DELETE, etc.

Of these, the most common are GET and POST, with the GET method typically used for data retrieval and the POST method used to make changes in the server's database. We shall return to GET again in our Wikipedia web scraping tutorial, which is usually the only method used for web scraping.

We won't go too much into what some of these other header fields mean, which you should be able to find references for easily online (etc: https://en.wikipedia.org/wiki/List_of_HTTP_header_fields).

Nonetheless, when troubleshooting your code for extracting data from the web, you'll often find yourself examining the header fields for both the request and response messages.

To round out this section, let's briefly examine the headers associated with the response (rather than the request) with the techniques we've learned, which are directly available in the main response object we have been working with.

```
In [7]: pprint(response.headers.items())
```

```
---------------------------------------------------------------------------

NameError                                 Traceback (most recent call last)
```

```
    <ipython-input-7-1002f5c92f98> in <module>()
----> 1 pprint(response.headers.items())


    NameError: name 'response' is not defined
```

2.3.3 End Note: Browser vs. Console

From the server's perspective, the request it receives from your browser is not so different from the request received from your console (though some servers use a range of methods to determine if the request comes from a "valid" person using a browser, versus an automated program.)

The server relies on the header request fields to determine what to return, and includes a number of header fields in its response, in addition to its content.

The main difference is that in the browser, you interact with the server via a graphical user interface (GUI), so that much of the header specification, both in the request and response, remain invisible to you. In your console, you often have to specify or parse this content manually - while this involves more work, it also allows you a great deal more flexibility, and the ability to automate certain tasks.

2.4 2) Web terminology: some important distinctions

2.5 2b) Menagerie of tools: crawlers, spiders, scrapers - what's the difference?

Web crawlers or spiders are used by search engines to index the web. The metaphor is that of an automated bot with long, spindly legs, traversing from hyperlink to hyperlink. Search engines use these crawlers to continually traverse the web and index new or changed content, so that our search queries reflect the most recent and up-to-date content.

Web scraping is a little different. While many of the tools used may be identical or similar, web scraping "focuses more on the transformation of unstructured data on the web, typically in HTML format, into structured data that can be stored and analyzed in a central local database or spreadsheet." (https://en.wikipedia.org/wiki/Web_scraping) In other words, web scraping focuses on translating data into a form ready for storage and analysis (versus just indexing).

In many cases, to the server, these processes look somewhat identical. Resources are sent in response to requests. Rather, it is what is done to those resources after they are sent, and the overall goal, that differentiates web crawling and scraping.

Most websites want crawlers to find them so their pages appear on popular search engines, but see no clear-cut benefit when their content is parsed and converted into usable data. Beyond research, many companies also use web scraping (in a legal grey area or illegally) to repurpose content, etc, a real estate website scraping data from Craigslist to re-post as listings on their website.

2.6 4) Considerate robots and legality

Typically, in starting a new web scraping project, you'll want to follow these steps:
1) Find the websites' robots.txt and do not access those pages through your bot
2) Make sure your bot does not make too many requests in a specific period (etc. by using Python's sleep.wait function)
3) Look up the website's term of use or terms of service.

We'll discuss each of these briefly.

2.6.1 What data owners care about

Data owners are concerned with:
1) Keeping their website up
2) Protecting the commercial value of their data

Their policies and responses differ with respect to these two areas. You'll need to do some research to determine what is appropriate with regards to your research.

1) Keeping their website up Most commercial websites have strategies to throttle or block IPs that make too many requests within a fixed amount of time. Because a bot can make a large number of requests in a small amount of time (etc. entering 100 different terms into Google in one second), servers are able to determine if traffic is coming from a bot or a person (among many other methods). For companies that rely on advertising, like Google or Twitter, these requests do not represent "human eyeballs" and need to be filtered out from their bill to advertisers.

In order to keep their site up and running, companies may block your IP temporarily or permanently if they detect too many requests coming from your IP, or other signs that requests are being made by a bot instead of a person. If you systematically down a site (such as sending millions of requests to an official government site), there is the small chance your actions may be interpreted maliciously (and regarded as hacking), with risk of prosecution.

2) Protecting the commercial value of their data Companies are also typically very protective of their data, especially data that ties directly into how they make money. A listings site (like Craigslist), for instance, would lose traffic if listings on its site were poached and transfered to a competitor, or if a rival company used scraping tools to derive lists of users to contact. For this reason, companies' term of use agreements are typically very restrictive of what you can do with their data.

Different companies may have a range of responses to your scraping, depending on what you do with the data. Typically, repurposing the data for a rival application or business will trigger a strong response from the company (i.e. legal attention). Publishing any analysis or results, either in a formal academic journal or on a blog or webpage, may be of less concern, though legal attention is still possible.

2.6.2 robots.txt: internet convention

The robots.txt file is typically located in the root folder of the site, with instructions to various services (User-agents) on what they are not allowed to scrape.

Typically, the robots.txt file is more geared towards search engines (and their crawlers) more than anything else.

However, companies and agencies typically will not want you to scrape any pages that they disallow search engines from accessing. Scraping these pages makes it more likely for your IP to be detected and blocked (along with other possible actions.)

Below is an example of reddit's robots.txt file: https://www.reddit.com/robots.txt 80legs User-agent: 008 Disallow: /

User-Agent: bender Disallow: /my$_s hiny_m etal_a ss$

User-Agent: Gort Disallow: /earth

User-Agent: * Disallow: /*.json Disallow: /*.json-compact Disallow: /*.json-html Disallow: /*.xml Disallow: /*.rss Disallow: /*.i Disallow: /*.embed Disallow: /*/comments/*?*sort= Disallow: /r/*/comments/*/*/c* Disallow: /comments/*/*/c* Disallow: /r/*/submit Disallow: /message/compose* Disallow: /api Disallow: /post Disallow: /submit Disallow: /goto Disallow: /*after= Disallow: /*before= Disallow: /domain/*t= Disallow: /login Disallow: /reddits/search Disallow: /search Disallow: /r/*/search Allow: / User blahblahblah provides a concise description of how to read the robots.txt file: https://www.reddit.com/r/learnprogramming/comments/3l1lcq/how_do_you_find_out_if_a_website_is_scrapable/ - The bot that calls itself 008 (apparently from 80legs) isn't allowed to access anything - bender is not allowed to visit my$_s hiny_m etal_a ss(it's a Futurama joke, the page doesn't actually exist)$ – $Gort isn't allowed to visit Earth(another joke, from The Day the Earth Stood Still)$ – $Other scrapers should avoid checking the API methods or "compose message" or 'search" or the "over 18?" page (because those aren't$ $wildcard category of what the site generally do not want bots to access. You should make sure your scraper does not access any of those$

3 Let's get started!

Now that we've gone through major concepts and tried out a few code snippets, let's hone our Python skills and build two basic bots, one on Wikipedia, and one using Twitter's API.

3.1 6) Tutorial 1: Creating a friendly bot on Wikipedia

Our first use case involves scraping some basic information about technology companies from Wikipedia. Say you are the chief innovation officer of a small city in the San Francisco Bay Area. A number of large-scale local investments in office space have taken place, with space opening up over the next few years. You wish to be part of the trend of technology companies moving out of San Francisco and Silicon Valley. You have been networking and talking to companies at events and conferences, but would like a more systematic way of identifying companies to focus on.

You notice a list of 179 technology companies based in the San Francisco area on Wikipedia: https://en.wikipedia.org/wiki/Category:Technology_companies_based_in_the_San_Francisco_Bay_Area

Your goal is to scrape basic useful information about each company in a list, into which you can do some summary statistics to identify companies or even industries you are interested in focusing on.

** In particular, you want to know: **
1) what industry they are in
2) where the company is currently headquartered
3) the number of employees
4) website address of the company

This will allow you to know the current and budding tech hubs in the Bay area, get a better sense of your competition, and the number of jobs you can attract to your city. For convenience, you also collate the website addresses of the companies to pull into your list.

3.2 Examining the webpage structure

The first step is to figure out whether and how easily the data you want can be extracted, first by examining the webpage structure in your browser, then on your console.

You can inspect any element in your browser by right-clicking it and selecting inspect, which will bring up the Developer Tools pane.

Typically, you'll first want to identify the element that you want to pull data from. Next, you'll need to figure out a strategy to locate and "crawl" through relevant pages. In a forum, for instance, a bot may be set up to click the "Next page" button once all posts on a single page have been visited and saved. More advanced strategies would include visiting all hyperlinks on every page visited, so that the bot continually updates the list of links to crawl through.

Inspecting the Index page First, you will want to inspect the element associated with each link we want to visit on the index page.

Next, you will want to inspect the element with the data we would like to extract, corresponding to each link on the index page.

Inspecting each individual page In our case, it looks like the format of the data in both the index and individual pages are regular enough for us to be able to parse them programatically. We next confirm this by interacting with both the index and individual pages in our console.

3.3 Interacting with the webpage through the console

After examining the webpage structure through your browser, now it's time to interact with the underlying html code (what you see in the inspect element page) directly in your console. Both processes are useful to coming up with a strategy of how (and whether) data from the website can be scraped.

First, import requests and BeautifulSoup. Downloading a html copy of the site is as simple as:

```
In [8]: from bs4 import BeautifulSoup

        r = requests.get('http://en.wikipedia.org/wiki/Category:Technology_companies_based_in_the_San_F
        print(r.content[0:300])
```

```
b'<!DOCTYPE html>\n<html lang="en" dir="ltr" class="client-nojs">\n<head>\n<meta charset="UTF-8" />\n<ti
```

Once you've downloaded the html file, you'll now want to pass it into BeautifulSoup. BeautifulSoup converts the html file in an easily searchable and navigable structure, which you'll see in our examples below.

```
In [9]: soup = BeautifulSoup(r.content)
        type(soup)
```

```
/Users/dillon/anaconda/lib/python3.5/site-packages/bs4/__init__.py:166: UserWarning: No parser was explic

To get rid of this warning, change this:

 BeautifulSoup([your markup])

to this:

 BeautifulSoup([your markup], "lxml")

  markup_type=markup_type))
```

```
Out[9]: bs4.BeautifulSoup
```

You should have the browser page open side by step, identifying the elements you want to extract using the Inspect element tool, and then using BeautifulSoup's functions to see if you can retrieve them in the console. You can also scroll over each element in the Elements tab to see what they correspond to on the web page.

If you scroll over the div with id "mw-pages" on the Elements tab, you'll see that it corresponds to the entire "Technology companies based in the San Francisco Bay Area" pane.

Let's first try to select this, and confirm we've selected the right element by printing out the result. In the code, we are telling soup to find any elements with the "div" element tag, with id "mw-pages" that we saw in the inspect element pane.

```
In [10]: company_section = soup.findAll("div", {"id": "mw-pages"})
         print(type(company_section))
```

```
<class 'bs4.element.ResultSet'>
```

As we navigate the result returned, we see that it is a "ResultSet", which suggests that it can be retrieved by index. You can also just try it out.

```
In [11]: print(company_section[0])
```

```
<div id="mw-pages">
<h2><span id="Pages_in_category"></span>Pages in category "Technology companies based in the San Francis
<p>The following 179 pages are in this category, out of 179 total. This list may not reflect recent cha
</p><div class="mw-content-ltr" dir="ltr" lang="en"><div class="mw-category"><div class="mw-category-gr
<ul><li><a href="/wiki/HP_3PAR" title="HP 3PAR">HP 3PAR</a></li></ul></div><div class="mw-category-group
<ul><li><a href="/wiki/Achronix" title="Achronix">Achronix</a></li>
<li><a href="/wiki/Advanced_Micro_Devices" title="Advanced Micro Devices">Advanced Micro Devices</a></li
<li><a href="/wiki/Aerohive_Networks" title="Aerohive Networks">Aerohive Networks</a></li>
```

Affymetrix
Agami Systems
Agilent Technologies
AirTouch
AKM Semiconductor, Inc.<
All Power Labs
Alphabet Energy
AlphaSense
Altamira Software
AltaVista
Altos Computer Systems</li
Alza
<a href="/wiki/AMAX_Information_Technologies" title="AMAX Information Technologies">AMAX Information
American Logic Machines<
Amiga Corporation
Antec
Anthera Pharmaceuticals<
Apple Inc.
ASSIA (company)
Audience (company)
AuraOne Systems
Aureal Semiconductor
<a href="/wiki/BioMarin_Pharmaceutical" title="BioMarin Pharmaceutical">BioMarin Pharmaceutical<
BioPharm (US company)
Biosearch Technologies</li
BrightSource Energy
Brderbund
<a class="mw-redirect" href="/wiki/Byte_Shop" title="Byte Shop">B
Cask (company)
Cerego
Cetus Corporation
Cisco Systems
Clean Edge
Complete Genomics
Corsair Components
Covad
<a href="/wiki/Crossbar_(computer_hardware_manufacturer)" title="Crossbar (computer hardware manufac
Cutter Laboratories
Cypress Semiconductor
</
Deeplearning4j
DiscoveRx
DNA2.0
DOER Marine
Dolby Laboratories
Double Robotics
Dust Networks <div class="mw-
<a href="/wiki/Electronics_for_Imaging" title="Electronics for Imaging">Electronics for Imaging<
Energy Recovery Inc.
Enphase Energy
Envivio
Etec Systems, Inc.
Everex
Exidy <h3>F</
<a href="/wiki/Fairchild_Semiconductor" title="Fairchild Semiconductor">Fairchild Semiconductor<

Fortify Software
Four-Phase Systems
Foveon
<a href="/wiki/Franklin_Ophthalmic_Instruments" title="Franklin Ophthalmic Instruments">Franklin Oph
Genentech
General Magic
Genesys (company)
GlassPoint Solar
GlobalFoundries
Green Charge Networks
Gusto (software) <div c
Handspring (company)
<a href="/wiki/Hercules_Computer_Technology" title="Hercules Computer Technology">Hercules Computer
<a href="/wiki/Hewlett_Packard_Enterprise" title="Hewlett Packard Enterprise">Hewlett Packard Enterp
Hewlett-Packard
Hoopla Software
HP Inc.
<a href="/wiki/Human_Engineered_Software" title="Human Engineered Software">Human Engineered Softwar
Hurricane Electric
Ikanos Communications</l
Impax Laboratories
Intel
InvenSense
InVision Technologies
</
Jennerex
Juniper Networks <div c
Kiva Software
KleenSpeed Technologies<
Kosan Biosciences <di
Lam Research
LeapFrog Enterprises
Lexar
<a href="/wiki/Lightwave_Electronics_Corporation" title="Lightwave Electronics Corporation">Lightwav
Linear Technology <di
Made In Space, Inc.
Makani Power
Maxim Integrated
Intel Security
<a href="/wiki/McCune_Audio/Video/Lighting" title="McCune Audio/Video/Lighting">McCune Audio/Video/L
Media Vision
Medivation
Meka Robotics
<a href="/wiki/Mendel_Biotechnology,_Inc." title="Mendel Biotechnology, Inc.">Mendel Biotechnology,
Meru Networks
MeWe
<a href="/wiki/Meyer_Sound_Laboratories" title="Meyer Sound Laboratories">Meyer Sound Laboratories</
MongoLab
Mozilla Corporation
</d
Nanosolar
National Semiconductor</li
Navigenics
Netscape
NeXT
Next Thing Co.

NovaBay Pharmaceuticals<
Numenta <h
Oberheim Electronics
OLogic
OpenAI
Oracle Corporation
<a href="/wiki/Osborne_Computer_Corporation" title="Osborne Computer Corporation">Osborne Computer C
OSIsoft <h
P.A. Semi
<a class="mw-redirect" href="/wiki/PA_Semiconductor" title="PA Se
Palm, Inc.
Palo Alto Networks
<a href="/wiki/Peninsula_Engineering_Group,_Inc." title="Peninsula Engineering Group, Inc.">Peninsul
Primus Power
Processor Technology
Prosetta
Pyramid Technology
Qualcomm Atheros
Quantum Effect Devices</li
Quark Pharmaceuticals
Qume <h3>R</h3>
Rambus
Recommind
Redwood Robotics <div c
Sangfor Technologies
Seagate Technology
Sensory, Inc.
Shugart Associates
Sidecar (company)
Silego Technology Inc.</li
Silicon Graphics
Siluria Technologies
Skymind
Sling Media
SmugMug
SolarCity
<a href="/wiki/Solido_Design_Automation" title="Solido Design Automation">Solido Design Automation</
SoloPower
Solyndra
Stem Cell Theranostics</li
SunPower
Sunrun
Supertek Computers
Sybase
Synaptics <div class="mw-category-gr
Tabula (company)
Talari Networks
Tandem Computers
Tengen (company)
Theranos
Thoratec
Tout (company)
Treasure Data <div class="mw-
Ubiquitous Energy

```
<li><a href="/wiki/Umtech" title="Umtech">Umtech</a></li>
<li><a href="/wiki/UTStarcom" title="UTStarcom">UTStarcom</a></li></ul></div><div class="mw-category-gr
<ul><li><a href="/wiki/Vivante_Corporation" title="Vivante Corporation">Vivante Corporation</a></li>
<li><a href="/wiki/Volterra_Semiconductor" title="Volterra Semiconductor">Volterra Semiconductor</a></li
<li><a href="/wiki/VPL_Research" title="VPL Research">VPL Research</a></li>
<li><a href="/wiki/VSee" title="VSee">VSee</a></li>
<li><a href="/wiki/VW_Electronics_Research_Laboratory" title="VW Electronics Research Laboratory">VW Ele
<ul><li><a href="/wiki/@WalmartLabs" title="@WalmartLabs">@WalmartLabs</a></li>
<li><a href="/wiki/Willow_Garage" title="Willow Garage">Willow Garage</a></li>
<li><a href="/wiki/Wind_River_Systems" title="Wind River Systems">Wind River Systems</a></li></ul></div>
<ul><li><a href="/wiki/Xilinx" title="Xilinx">Xilinx</a></li></ul></div><div class="mw-category-group">
<ul><li><a href="/wiki/Zenefits" title="Zenefits">Zenefits</a></li>
<li><a href="/wiki/Zscaler" title="Zscaler">Zscaler</a></li></ul></div></div></div>
</div>
```

You can see at the start of the element retrieved that it is indeed a division with id "mw-pages" - we can confirm by browsing the text that we've selected the correct element. Next, let's retrieve each section (corresponding to each alphabet), now searching the company section with class type "mw-category-group".

```
In [12]: each_alphabet = company_section[0].find_all("div", {"class":"mw-category-group"})
         print(len(each_alphabet))
         print(each_alphabet[0])
```

```
26
<div class="mw-category-group"><h3>3</h3>
<ul><li><a href="/wiki/HP_3PAR" title="HP 3PAR">HP 3PAR</a></li></ul></div>
```

Finally, within each section, we want to pull out the individual hyperlinks corresponding to each company. Let's use the second element in the index (the letter "A" instead of the category group for "3") as it has more than one company.

```
In [13]: alphabet_a = each_alphabet[1]
         print(alphabet_a)
```

```
<div class="mw-category-group"><h3>A</h3>
<ul><li><a href="/wiki/Achronix" title="Achronix">Achronix</a></li>
<li><a href="/wiki/Advanced_Micro_Devices" title="Advanced Micro Devices">Advanced Micro Devices</a></li
<li><a href="/wiki/Aerohive_Networks" title="Aerohive Networks">Aerohive Networks</a></li>
<li><a href="/wiki/Affymetrix" title="Affymetrix">Affymetrix</a></li>
<li><a href="/wiki/Agami_Systems" title="Agami Systems">Agami Systems</a></li>
<li><a href="/wiki/Agilent_Technologies" title="Agilent Technologies">Agilent Technologies</a></li>
<li><a href="/wiki/AirTouch" title="AirTouch">AirTouch</a></li>
<li><a href="/wiki/AKM_Semiconductor,_Inc." title="AKM Semiconductor, Inc.">AKM Semiconductor, Inc.</a><
<li><a href="/wiki/All_Power_Labs" title="All Power Labs">All Power Labs</a></li>
<li><a href="/wiki/Alphabet_Energy" title="Alphabet Energy">Alphabet Energy</a></li>
<li><a href="/wiki/AlphaSense" title="AlphaSense">AlphaSense</a></li>
<li><a href="/wiki/Altamira_Software" title="Altamira Software">Altamira Software</a></li>
<li><a href="/wiki/AltaVista" title="AltaVista">AltaVista</a></li>
<li><a href="/wiki/Altos_Computer_Systems" title="Altos Computer Systems">Altos Computer Systems</a></li
<li><a href="/wiki/Alza" title="Alza">Alza</a></li>
<li><a href="/wiki/AMAX_Information_Technologies" title="AMAX Information Technologies">AMAX Information
<li><a href="/wiki/American_Logic_Machines" title="American Logic Machines">American Logic Machines</a><
<li><a href="/wiki/Amiga_Corporation" title="Amiga Corporation">Amiga Corporation</a></li>
<li><a href="/wiki/Antec" title="Antec">Antec</a></li>
<li><a href="/wiki/Anthera_Pharmaceuticals" title="Anthera Pharmaceuticals">Anthera Pharmaceuticals</a><
```

```
<li><a href="/wiki/Apple_Inc." title="Apple Inc.">Apple Inc.</a></li>
<li><a href="/wiki/ASSIA_(company)" title="ASSIA (company)">ASSIA (company)</a></li>
<li><a href="/wiki/Audience_(company)" title="Audience (company)">Audience (company)</a></li>
<li><a href="/wiki/AuraOne_Systems" title="AuraOne Systems">AuraOne Systems</a></li>
<li><a href="/wiki/Aureal_Semiconductor" title="Aureal Semiconductor">Aureal Semiconductor</a></li></ul>
```

We next want to select all elements with the "li" tag, and print them out to make sure they correspond to what we expect to see on the page.

```
In [14]: company_list = alphabet_a.find_all("li")
         for i in company_list:
             print("")
             print(i)
```

```
<li><a href="/wiki/Achronix" title="Achronix">Achronix</a></li>

<li><a href="/wiki/Advanced_Micro_Devices" title="Advanced Micro Devices">Advanced Micro Devices</a></li

<li><a href="/wiki/Aerohive_Networks" title="Aerohive Networks">Aerohive Networks</a></li>

<li><a href="/wiki/Affymetrix" title="Affymetrix">Affymetrix</a></li>

<li><a href="/wiki/Agami_Systems" title="Agami Systems">Agami Systems</a></li>

<li><a href="/wiki/Agilent_Technologies" title="Agilent Technologies">Agilent Technologies</a></li>

<li><a href="/wiki/AirTouch" title="AirTouch">AirTouch</a></li>

<li><a href="/wiki/AKM_Semiconductor,_Inc." title="AKM Semiconductor, Inc.">AKM Semiconductor, Inc.</a><

<li><a href="/wiki/All_Power_Labs" title="All Power Labs">All Power Labs</a></li>

<li><a href="/wiki/Alphabet_Energy" title="Alphabet Energy">Alphabet Energy</a></li>

<li><a href="/wiki/AlphaSense" title="AlphaSense">AlphaSense</a></li>

<li><a href="/wiki/Altamira_Software" title="Altamira Software">Altamira Software</a></li>

<li><a href="/wiki/AltaVista" title="AltaVista">AltaVista</a></li>

<li><a href="/wiki/Altos_Computer_Systems" title="Altos Computer Systems">Altos Computer Systems</a></li

<li><a href="/wiki/Alza" title="Alza">Alza</a></li>

<li><a href="/wiki/AMAX_Information_Technologies" title="AMAX Information Technologies">AMAX Information

<li><a href="/wiki/American_Logic_Machines" title="American Logic Machines">American Logic Machines</a><

<li><a href="/wiki/Amiga_Corporation" title="Amiga Corporation">Amiga Corporation</a></li>

<li><a href="/wiki/Antec" title="Antec">Antec</a></li>

<li><a href="/wiki/Anthera_Pharmaceuticals" title="Anthera Pharmaceuticals">Anthera Pharmaceuticals</a><

<li><a href="/wiki/Apple_Inc." title="Apple Inc.">Apple Inc.</a></li>
```

```
<li><a href="/wiki/ASSIA_(company)" title="ASSIA (company)">ASSIA (company)</a></li>

<li><a href="/wiki/Audience_(company)" title="Audience (company)">Audience (company)</a></li>

<li><a href="/wiki/AuraOne_Systems" title="AuraOne Systems">AuraOne Systems</a></li>

<li><a href="/wiki/Aureal_Semiconductor" title="Aureal Semiconductor">Aureal Semiconductor</a></li>
```

If we select one company and print it out, we can see we're pretty close.

```
In [15]: one_company = company_list[0]
         print(one_company)

<li><a href="/wiki/Achronix" title="Achronix">Achronix</a></li>
```

We can also select the next child element by doing the following:

```
In [16]: one_company.a

Out[16]: <a href="/wiki/Achronix" title="Achronix">Achronix</a>
```

And finally, get the attributes associated with the "a" hyperlink tag, which returns a Python dictionary.

```
In [17]: one_company.a.attrs

Out[17]: {'href': '/wiki/Achronix', 'title': 'Achronix'}
```

Now that we've received the element containing the element we want, we can also print out its parents to view the position within the html "tree."

```
In [18]: print(type(one_company.a.parents))
         for i in one_company.a.parents:
             print(i.name)

<class 'generator'>
li
ul
div
div
div
div
div
div
div
div
body
html
[document]
```

This lets you see the different nested elements you'll need to traverse to get to the element you need. In many cases, you'll use explicit selection of the element together with the find_all command to isolate the element you need.

Finally, let's write a loop to store all of our desired hyperlink dictionaries in a single Python list.

```
In [19]: link_list = []
         for each_section in company_section:
             company_list = each_section.find_all("li")
             for each_company in company_list:
                 new_dict = each_company.a
                 link_list.append(new_dict)
         print(len(link_list))
```

```
179
```

We now have a list of 175 hyperlinks to loop through for our next section.
Now using the list, let's load the first page and locate the text elements we want

```
In [20]: example_site = link_list[0]
         print(example_site)

         company_page = requests.get("http://wikipedia.org" + example_site['href'])
```

```
<a href="/wiki/HP_3PAR" title="HP 3PAR">HP 3PAR</a>
```

```
In [21]: print(company_page.content[0:200])
```

```
b'<!DOCTYPE html>\n<html lang="en" dir="ltr" class="client-nojs">\n<head>\n<meta charset="UTF-8" />\n<ti
```

In your browser, you should be using inspect element to confirm the position of the desired element in the html tree. We can see the element is a table with class name "infobox vcard". Let's try to select this next. First, we need the html document into soup as we did before. (We convert to string just to allow us to print the first 500 charactes of the text here.)

```
In [22]: soup = BeautifulSoup(company_page.content)
         info_box = soup.find("table", {"class": "infobox vcard"})
         print(str(info_box)[0:500])
```

```
<table class="infobox vcard" style="width:22em">
<caption class="fn org">HP 3PAR</caption>
<tr>
<th scope="row" style="padding-right:0.5em;">
<div style="padding:0.1em 0;line-height:1.2em;"><a href="/wiki/Types_of_business_entity" title="Types of
</th>
<td class="category" style="line-height:1.35em;">Subsidiary</td>
</tr>
<tr>
<th scope="row" style="padding-right:0.5em;">Industry</th>
<td class="category" style="line-height:1.35em;"><a href="/wiki/Data_storage_dev

/Users/dillon/anaconda/lib/python3.5/site-packages/bs4/__init__.py:166: UserWarning: No parser was explic

To get rid of this warning, change this:

 BeautifulSoup([your markup])

to this:

 BeautifulSoup([your markup], "lxml")

  markup_type=markup_type))
```

Now, using the various tools we've had before, we can drill down to the specific element containing the data we need. As before, we select and print a single row to help guide the process.

```
In [23]: table_elements = info_box.find_all("tr")
         one_row = table_elements[0]
         print(one_row)
```

```
<tr>
<th scope="row" style="padding-right:0.5em;">
<div style="padding:0.1em 0;line-height:1.2em;"><a href="/wiki/Types_of_business_entity" title="Types of
</th>
<td class="category" style="line-height:1.35em;">Subsidiary</td>
</tr>
```

First, let's try to select the element containing the variable name "Type".

```
In [24]: print(one_row.th)
         print("")
         print(one_row.th.div)
         print("")
         print(one_row.th.div.text)
```

```
<th scope="row" style="padding-right:0.5em;">
<div style="padding:0.1em 0;line-height:1.2em;"><a href="/wiki/Types_of_business_entity" title="Types of
</th>

<div style="padding:0.1em 0;line-height:1.2em;"><a href="/wiki/Types_of_business_entity" title="Types of

Type
```

Next, let's select the element containing the variable value, in this case "Subsidiary".

```
In [25]: print(one_row.td)
         print("")
         print(one_row.td.text)
```

```
<td class="category" style="line-height:1.35em;">Subsidiary</td>

Subsidiary
```

Now, let's loop through all rows to get all data that's available on the company. Depending on how well-structured the data is, this can be something of a trial and error process.

```
In [26]: for one_row in table_elements:
             print(one_row.th.div.text + ": " + one_row.td.text)
```

```
Type: Subsidiary


        ---------------------------------------------------------------------------

        AttributeError                            Traceback (most recent call last)

        <ipython-input-26-5678f42391a1> in <module>()
          1 for one_row in table_elements:
    ----> 2     print(one_row.th.div.text + ": " + one_row.td.text)


        AttributeError: 'NoneType' object has no attribute 'text'
```

We get an AttributeError for the "NoneType" object due to some of the "th" elements being empty. If we do some simple Exception capturing, we can get the loop to run through.

```
In [27]: for one_row in table_elements:
             try:
                 print(one_row.th.div.text + ": " + one_row.td.text)
             except Exception:
                 continue
```

```
Type: Subsidiary
Area served: Worldwide
Key people: David C. Scott
(President), (CEO) & (Director)
Operating income: US$ -3.33 million (FY10)
Net income: US$ -3.18 million (FY10)
Number of employees: 657 (FY10)
```

Now that we have the data we need, let's store it in a Python dictionary.

```
In [28]: new_dict = {}
         for one_row in table_elements:
             try:
                 print(one_row.th.div.text + ": " + one_row.td.text)
                 new_dict[one_row.th.div.text] = one_row.td.text
             except Exception:
                 continue
```

```
Type: Subsidiary
Area served: Worldwide
Key people: David C. Scott
(President), (CEO) & (Director)
Operating income: US$ -3.33 million (FY10)
Net income: US$ -3.18 million (FY10)
Number of employees: 657 (FY10)
```

We can browse the dictionary to make sure it is capturing the data correctly.

```
In [29]: print(new_dict.keys())
         print(new_dict)
```

```
dict_keys(['Type', 'Key people', 'Net income', 'Area served', 'Number of employees', 'Operating income']
{'Type': 'Subsidiary', 'Key people': 'David C. Scott\n(President), (CEO) & (Director)', 'Net income': 'U
```

Let's quickly rehash what we did. We first extracted a list of hyperlinks from our index page, storing in our link_list variable. Next, we visited one of the pages, pulled its html into soup, and extracted the data from the element with class name "infobox vcard" into our new_dict variable.

The next step is to write an overall loop so that we can collect the "infobox vcard" data for all elements in our list. info_box = soup.find("table", {"class": "infobox vcard"})

```
In [30]: list_of_dicts = []

         for each_link in link_list[0:3]:
             print("")
             print(each_link)
             print("")
             company_page = requests.get("http://wikipedia.org" + each_link['href'])
```

```
            soup = BeautifulSoup(company_page.content)
            info_box = soup.find("table", {"class": "infobox vcard"})
            table_elements = info_box.find_all("tr")
            new_dict = {}
            new_dict['Company_name'] = each_link['title']
            for one_row in table_elements:
                try:
                    print(one_row.th.div.text + ": " + one_row.td.text)
                    # we convert to string as a precaution to make sure more complex elements are stor
                    new_dict[one_row.th.div.text] = str(one_row.td.text)
                except Exception:
                    continue
            # add the dictionary after we've added all variable names and values to each dictionary
            list_of_dicts.append(new_dict)

        print("")
        print(len(list_of_dicts))
```

```
<a href="/wiki/HP_3PAR" title="HP 3PAR">HP 3PAR</a>

Type: Subsidiary
Area served: Worldwide
Key people: David C. Scott
(President), (CEO) & (Director)
Operating income: US$ -3.33 million (FY10)
Net income: US$ -3.18 million (FY10)
Number of employees: 657 (FY10)

<a href="/wiki/Achronix" title="Achronix">Achronix</a>

Type: Private
Key people: Robert Blake (CEO), John Lofton Holt (Chairman), Rahul Nimaiyyar (VP HW Engineering), Kamal
Number of employees: <200

<a href="/wiki/Advanced_Micro_Devices" title="Advanced Micro Devices">Advanced Micro Devices</a>

Type: Public
Area served: Worldwide
Key people: Lisa Su (CEO)[2]
Bruce Claflin (Executive Chairman)
Operating income:  -$155million (2014)[5]
Net income:  -$403million (2014)[5]
Number of employees: 9,687 (2014)[5]

3

/Users/dillon/anaconda/lib/python3.5/site-packages/bs4/__init__.py:166: UserWarning: No parser was explic

To get rid of this warning, change this:

 BeautifulSoup([your markup])

to this:

 BeautifulSoup([your markup], "lxml")
```

```
  markup_type=markup_type))
```

Let's browse our list_of_dicts object to make sure it contains the data we need.

```
In [31]: print(list_of_dicts)
```

```
[{'Type': 'Subsidiary', 'Key people': 'David C. Scott\n(President), (CEO) & (Director)', 'Net income':
```

Typically, a "friendly" bot would try to space out the number of requests (in addition to not scraping the pages robots.txt disallows and obeying the general terms of service) so that the server can manage its traffic. One simple way to do this is to add a time.sleep command into your loop.

As the entire class will be sharing the same IP, it's recommended that you add a longer wait time and limit the number of companies from link_list you scrape while in class.

```
In [32]: import time

         list_of_dicts = []

         for each_link in link_list[0:3]:
             print("")
             wait_time = 1
             print('waiting ' + str(wait_time) + "s...")
             time.sleep(1)
             print("")
             print(each_link)
             print("")
             company_page = requests.get("http://wikipedia.org" + each_link['href'])
             soup = BeautifulSoup(company_page.content)
             info_box = soup.find("table", {"class": "infobox vcard"})
             table_elements = info_box.find_all("tr")
             new_dict = {}
             new_dict['Company_name'] = each_link['title']
             for one_row in table_elements:
                 try:
                     print(one_row.th.div.text + ": " + one_row.td.text)
                     # we convert to string as a precaution to make sure more complex elements are stor
                     new_dict[one_row.th.div.text] = str(one_row.td.text)
                 except Exception:
                     continue
             # add the dictionary after we've added all variable names and values to each dictionary
             list_of_dicts.append(new_dict)
```

```
waiting 1s...

<a href="/wiki/HP_3PAR" title="HP 3PAR">HP 3PAR</a>

Type: Subsidiary
Area served: Worldwide
Key people: David C. Scott
(President), (CEO) & (Director)
Operating income: US$ -3.33 million (FY10)
Net income: US$ -3.18 million (FY10)
Number of employees: 657 (FY10)
```

```
waiting 1s...

<a href="/wiki/Achronix" title="Achronix">Achronix</a>

Type: Private
Key people: Robert Blake (CEO), John Lofton Holt (Chairman), Rahul Nimaiyyar (VP HW Engineering), Kamal
Number of employees: <200

waiting 1s...

<a href="/wiki/Advanced_Micro_Devices" title="Advanced Micro Devices">Advanced Micro Devices</a>

Type: Public
Area served: Worldwide
Key people: Lisa Su (CEO)[2]
Bruce Claflin (Executive Chairman)
Operating income:  -$155million (2014)[5]
Net income:  -$403million (2014)[5]
Number of employees: 9,687 (2014)[5]

/Users/dillon/anaconda/lib/python3.5/site-packages/bs4/__init__.py:166: UserWarning: No parser was explic

To get rid of this warning, change this:

 BeautifulSoup([your markup])

to this:

 BeautifulSoup([your markup], "lxml")

  markup_type=markup_type))
```

Now we have our list of dictionaries containing the data we want for each company. If you browse a few of the company pages, you'll notice that the number of variables each "infobox vcard" has is not always the same. Some dictionaries will have fields that others don't.

To store this data flexibly, we can iterate through all our dictionaries and collect all keys from them.

```
In [33]: key_list = []
         for each_dict in list_of_dicts:
             for key in each_dict.keys():
                 key_list.append(key)
         print(key_list)
```

```
['Type', 'Key people', 'Net income', 'Area served', 'Company_name', 'Number of employees', 'Operating in
```

Next, we convert the list to a set to remove all repeat keys. This then contains all unique keys across our dictionaries.

```
In [34]: key_set = set(key_list)
         print(key_set)
```

```
{'Type', 'Company_name', 'Operating income', 'Key people', 'Net income', 'Area served', 'Number of emplo
```

We convert the set back to a list (and sort it) as csv.DictWriter takes a list for its fieldnames parameter.

```
In [35]: final_key_list = sorted(list(key_set))
         print(final_key_list)
```

```
['Area served', 'Company_name', 'Key people', 'Net income', 'Number of employees', 'Operating income',
```

With our complete key list, we can now write our dictionary into a csv file.

```
In [36]: import csv

         outpath= "../data/02_companies.csv"
         outfile = open(outpath, 'w')

         writer = csv.DictWriter(outfile, fieldnames=final_key_list, dialect='excel')

         writer.writeheader()
         for row in list_of_dicts:
             writer.writerow(row)
             print(row)
         outfile.close()
         print("done")
```

```
{'Type': 'Subsidiary', 'Key people': 'David C. Scott\n(President), (CEO) & (Director)', 'Net income': 'U
{'Company_name': 'Achronix', 'Number of employees': '<200', 'Type': 'Private', 'Key people': 'Robert Bla
{'Type': 'Public', 'Key people': 'Lisa Su (CEO)[2]\nBruce Claflin (Executive Chairman)', 'Net income':
done
```

You can open the file in Excel to view the data. Congrats, you just scraped valuable data for your project off the web!

4 Creating data with web APIs

Most people who think they want to do web scraping actually want to pull data down from site-supplied APIs. Using an API is better in almost every way, and really the only reason to scrape data is if:

1. The website was constructed in the 90s and does not have an API; or,
2. You are doing something illegal

If LiveJournal has an API, the website you are interested in probably does too.

4.1 What is an API?

API is shorthand for **A**pplication **P**rogramming **I**nterface, which is in turn computer-ese for a middleman.

Think about it this way. You have a bunch of things on your computer that you want other people to be able to look at. Some of them are static documents, some of them call programs in real time, and some of them are programs themselves.

Solution 1 You publish login credentials on the internet, and let anyone log into your computer
Problems:

1. People will need to know how each document and program works to be able to access their data
2. You don't want the world looking at your browser history

Solution 2 You paste everything into HTML and publish it on the internet
Problems:

1. This can be information overload
2. Making things dynamic can be tricky

Solution 3 You create a set of methods to act as an intermediary between the people you want to help and the things you want them to have access to.

Why this is the best solution:

1. People only access what you want them to have, in the way that you want them to have it
2. People use one language to get the things they want

Why this is still not Panglossian:

1. You will have to explain to people how to use your middleman

4.2 Twitter's API

Twitter has an API - mostly written for third-party apps - that is comparatively straightforward and gives you access to *nearly* all of the information that Twitter has about its users, including:

1. User histories
2. User (and tweet) location
3. User language
4. Tweet popularity
5. Tweet spread
6. Conversation chains

Also, Twitter returns data to you in json, or **J**ava **S**cript **O**bject **N**otation. This is a very common format for passing data around http connections for browsers and servers, so many APIs return it as a datatype as well (instead of using something like xml or plain text).

Luckily, json converts into native Python data structures. Specifically, every json object you get from Twitter will be a combination of nested `dicts` and `lists`, which you learned about yesterday. This makes Twitter a lot easier to manipulate in Python than html objects, for example.

Here's what a tweet looks like:

```
In [37]: import json

         with open('../data/02_tweet.json','r') as f:
             a_tweet = json.loads(f.read())
```

We can take a quick look at the structure by pretty printing it:

```
In [38]: from pprint import pprint

         pprint(a_tweet)
```

```
{'contributors': None,
 'coordinates': None,
 'created_at': 'Thu Apr 02 06:09:39 +0000 2015',
 'entities': {'hashtags': [], 'symbols': [], 'urls': [], 'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 583511591334719488,
 'id_str': '583511591334719488',
 'in_reply_to_screen_name': None,
```

```
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'lang': 'ht',
'place': None,
'retweet_count': 0,
'retweeted': False,
'source': '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>',
'text': '.IPA rettiwT eht tuoba nraeL',
'truncated': False,
'user': {'contributors_enabled': False,
         'created_at': 'Thu Apr 02 05:54:25 +0000 2015',
         'default_profile': True,
         'default_profile_image': False,
         'description': '',
         'entities': {'description': {'urls': []}},
         'favourites_count': 0,
         'follow_request_sent': False,
         'followers_count': 0,
         'following': False,
         'friends_count': 0,
         'geo_enabled': False,
         'id': 3129088320,
         'id_str': '3129088320',
         'is_translation_enabled': False,
         'is_translator': False,
         'lang': 'en',
         'listed_count': 0,
         'location': '',
         'name': 'Yelekreb Bald',
         'notifications': False,
         'profile_background_color': 'C0DEED',
         'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
         'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
         'profile_background_tile': False,
         'profile_image_url': 'http://pbs.twimg.com/profile_images/583509317476712449/mkd8KGeu_normal.jp
         'profile_image_url_https': 'https://pbs.twimg.com/profile_images/583509317476712449/mkd8KGeu_no
         'profile_link_color': '0084B4',
         'profile_location': None,
         'profile_sidebar_border_color': 'C0DEED',
         'profile_sidebar_fill_color': 'DDEEF6',
         'profile_text_color': '333333',
         'profile_use_background_image': True,
         'protected': False,
         'screen_name': 'tob_pohskrow',
         'statuses_count': 1,
         'time_zone': None,
         'url': None,
         'utc_offset': None,
         'verified': False}}
```


Time for a challenge! Let's see how much you remember about lists and dicts from yesterday. Go into the challenges directory and try your hand at `02_scraping/C_json.py`.

4.3 Authentication

Twitter controls access to their servers via a process of authentication and authorization. Authentication is how you let Twitter know who you are, in a way that is very hard to fake. Authorization is how the account owner (which will usually be yourself unless you are writing a Twitter app) controls what you are allowed to do in Twitter using their account. In Twitter, different levels of authorization require different levels of authentication.

Because we want to be able to interact with everything, we'll need the highest level of authorization and the strictest level of authentication. In Twitter, this means that we need two sets of ID's (called keys or tokens) and passwords (called secrets):

- consumer_key
- consumer_secret
- access_token_key
- access_token_secret

We'll provide some for you to use, but if you want to get your own you need to create an account on Twitter with a verified phone number. Then, while signed in to your Twitter account, go to: https://apps.twitter.com/. Follow the prompts to generate your keys and access tokens. Note that getting the second ID/password pair requires that you manually set the authorization level of your app.

We've stored our credentials in a separate file, which is smart. However, we have uploaded it to Github so that you have them too, which is not smart.

You should NEVER NEVER NEVER do this in real life.

We've stored it in YAML format, because it is more human-readible than JSON is. However, once it's inside Python, these data structures behave the same way.

```
In [39]: import yaml

         with open('../etc/creds.yml', 'r') as f:
             creds = yaml.load(f)
```

We're going to load these credentials into a requests module specifically designed for handling the flavor of authentication management that Twitter uses.

```
In [40]: from requests_oauthlib import OAuth1Session

         twitter = OAuth1Session(**creds)
```

That ** syntax we just used is called a "double splat" and is a python convenience function for converting the key-value pairs of a dictionary into keyword-argument pairs to pass to a function.

4.4 Accessing the API

Access to Twitter's API is organized through URLs called "endpoints". An endpoint is the location at which you can submit a request for Twitter to do something for you.

For example, the "endpoint" to search for specific kinds of tweets is at:

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

whereas posting new tweets is at:

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

For more information on the REST APIs, end points, and terms, check out: https://dev.twitter.com/rest/public. For the Streaming APIs: https://dev.twitter.com/streaming/overview.

All APIs on Twitter are "rate-limited" - this means that you are only allowed to ask a set number of questions per unit time (to keep their servers from being overloaded). This rate varies by endpoint and authorization, so be sure to check their developer site for the action you are trying to take.

For example, at the lowest level of authorization (Twitter calls this `application only`), you are allowed to make 450 search requests per 15 minute window, or about one every two seconds. At the highest level of authorization (Twitter calls this `user`) you can submit 180 requests every 15 minutes, or only about once every five seconds.

> side note - Google search is the worst rate-limiting I've ever seen, with an allowance of one hundred requests per day, or about once every *nine hundred seconds*

Let's try a couple of simple API queries. We're going to specify query parameters with `param`.

```
In [41]: search = "https://api.twitter.com/1.1/search/tweets.json"

         r = twitter.get(search, params={'q' : 'technology'})
```

This has returned an http response object, which contains data like whether or not the request succeeded:

```
In [42]: r.ok

Out[42]: True
```

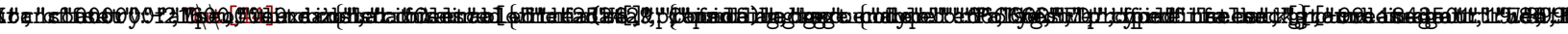
You can also get the http response code, and the reason why Twitter sent you that code (these are all super important for controlling the flow of your program).

```
In [43]: r.status_code, r.reason

Out[43]: (200, 'OK')
```

The data that we asked Twitter to send us in r.content

```
In [44]: r.content
```



But that's not helpful. We can extract it in python's representation of json with the `json` method:

```
In [45]: r.json()

Out[45]: {'search_metadata': {'completed_in': 0.072,
            'count': 15,
            'max_id': 702604626965696513,
            'max_id_str': '702604626965696513',
            'next_results': '?max_id=702604579188416511&q=technology&include_entities=1',
            'query': 'technology',
            'refresh_url': '?since_id=702604626965696513&q=technology&include_entities=1',
            'since_id': 0,
            'since_id_str': '0'},
           'statuses': [{'contributors': None,
             'coordinates': None,
             'created_at': 'Wed Feb 24 21:22:51 +0000 2016',
             'entities': {'hashtags': [],
              'symbols': [],
              'urls': [{'display_url': 'nyti.ms/1WJf4NV',
                'expanded_url': 'http://nyti.ms/1WJf4NV',
                'indices': [139, 140],
                'url': 'https://t.co/kFDYwJoiV5'}],
              'user_mentions': [{'id': 1754641,
                'id_str': '1754641',
```

```
   'indices': [3, 19],
   'name': 'NYT Business',
   'screen_name': 'nytimesbusiness'}]},
'favorite_count': 0,
'favorited': False,
'geo': None,
'id': 702604626965696513,
'id_str': '702604626965696513',
'in_reply_to_screen_name': None,
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': None,
'possibly_sensitive': False,
'retweet_count': 7,
'retweeted': False,
'retweeted_status': {'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:17:01 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'nyti.ms/1WJf4NV',
    'expanded_url': 'http://nyti.ms/1WJf4NV',
    'indices': [116, 139],
    'url': 'https://t.co/kFDYwJoiV5'}],
  'user_mentions': []},
 'favorite_count': 6,
 'favorited': False,
 'geo': None,
 'id': 702603158221144064,
 'id_str': '702603158221144064',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 7,
 'retweeted': False,
 'source': '<a href="http://www.socialflow.com" rel="nofollow">SocialFlow</a>',
 'text': 'Why the Apple case matters: With the Internet of Things, every home appliance coul
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Wed Mar 21 14:49:39 +0000 2007',
  'default_profile': False,
  'default_profile_image': False,
```

```
  'description': 'Business news from The New York Times.',
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'nytimes.com/business',
      'expanded_url': 'http://nytimes.com/business',
      'indices': [0, 22],
      'url': 'http://t.co/OKrfdBy4ch'}]}},
  'favourites_count': 789,
  'follow_request_sent': False,
  'followers_count': 728646,
  'following': False,
  'friends_count': 617,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 1754641,
  'id_str': '1754641',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 13338,
  'location': 'New York, NY',
  'name': 'NYT Business',
  'notifications': False,
  'profile_background_color': 'FFFFFF',
  'profile_background_image_url': 'http://pbs.twimg.com/profile_background_images/4433659/twi
  'profile_background_image_url_https': 'https://pbs.twimg.com/profile_background_images/4433
  'profile_background_tile': True,
  'profile_image_url': 'http://pbs.twimg.com/profile_images/2037622389/NYT_Twitter_Business_n
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/2037622389/NYT_Twitter_Bus
  'profile_link_color': '004276',
  'profile_sidebar_border_color': '323232',
  'profile_sidebar_fill_color': 'E7EFF8',
  'profile_text_color': '000000',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'nytimesbusiness',
  'statuses_count': 123516,
  'time_zone': 'Eastern Time (US & Canada)',
  'url': 'http://t.co/OKrfdBy4ch',
  'utc_offset': -18000,
  'verified': False}},
'source': '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android
'text': 'RT @nytimesbusiness: Why the Apple case matters: With the Internet of Things, every
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Wed Sep 21 19:57:48 +0000 2011',
 'default_profile': False,
 'default_profile_image': False,
 'description': 'Solitary and likes reading. Oh cocktails too. People not so much.',
 'entities': {'description': {'urls': []}},
 'favourites_count': 457,
 'follow_request_sent': False,
 'followers_count': 1453,
 'following': False,
 'friends_count': 2075,
```

```
 'geo_enabled': True,
 'has_extended_profile': True,
 'id': 377576061,
 'id_str': '377576061',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 32,
 'location': 'Lagos, Nigeria',
 'name': 'Oluwaseun Esq.',
 'notifications': False,
 'profile_background_color': '9AE4E8',
 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme16/bg.gif',
 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme16/bg.gif',
 'profile_background_tile': False,
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/377576061/1400343430',
 'profile_image_url': 'http://pbs.twimg.com/profile_images/669235191160918016/EQ09b_UK_normal
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/669235191160918016/EQ09b_UK
 'profile_link_color': '0084B4',
 'profile_sidebar_border_color': 'BDDCAD',
 'profile_sidebar_fill_color': 'DDFFCC',
 'profile_text_color': '333333',
 'profile_use_background_image': True,
 'protected': False,
 'screen_name': 'OluwaseunEsq',
 'statuses_count': 45657,
 'time_zone': 'Africa/Lagos',
 'url': None,
 'utc_offset': 3600,
 'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:51 +0000 2016',
 'entities': {'hashtags': [{'indices': [87, 95], 'text': 'PGCEmix'}],
  'symbols': [],
  'urls': [],
  'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 702604625426444289,
 'id_str': '702604625426444289',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'retweet_count': 0,
 'retweeted': False,
 'source': '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android
```

```
 'text': 'The blog struggle is real. I thought I was op-it, but technology is showing me fla
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Sat Sep 18 14:46:42 +0000 2010',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Xhosa Maties Music God Love Laughter Loyalty Music FashionHealthy body ima
  'entities': {'description': {'urls': []}},
  'favourites_count': 14,
  'follow_request_sent': False,
  'followers_count': 123,
  'following': False,
  'friends_count': 294,
  'geo_enabled': True,
  'has_extended_profile': True,
  'id': 192223545,
  'id_str': '192223545',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 0,
  'location': 'South Africa, Cape town ',
  'name': 'Thando PK Sayiti',
  'notifications': False,
  'profile_background_color': '709397',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme6/bg.gif',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme6/bg.gif',
  'profile_background_tile': False,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/192223545/1447002380',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/663402195824599040/vydMZL6C_norma
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/663402195824599040/vydMZL6
  'profile_link_color': 'FF3300',
  'profile_sidebar_border_color': '829D5E',
  'profile_sidebar_fill_color': '99CC33',
  'profile_text_color': '3E4415',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'PkThando',
  'statuses_count': 382,
  'time_zone': None,
  'url': None,
  'utc_offset': None,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:50 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'abcn.ws/1T7Few1',
    'expanded_url': 'http://abcn.ws/1T7Few1',
    'indices': [116, 139],
    'url': 'https://t.co/Td5vEB6PrA'}],
  'user_mentions': []},
 'favorite_count': 0,
```

```
'favorited': False,
'geo': None,
'id': 702604624637911040,
'id_str': '702604624637911040',
'in_reply_to_screen_name': None,
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': None,
'possibly_sensitive': False,
'retweet_count': 0,
'retweeted': False,
'source': '<a href="http://ifttt.com" rel="nofollow">IFTTT</a>',
'text': "RT breakingbytes: Apple CEO Cook: Aiding FBI in accessing gunman's phone would set
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Wed Sep 24 15:43:12 +0000 2014',
 'default_profile': False,
 'default_profile_image': False,
 'description': 'Em a LoOzeR!!! Not a Looser.',
 'entities': {'description': {'urls': []}},
 'favourites_count': 766,
 'follow_request_sent': False,
 'followers_count': 1537,
 'following': False,
 'friends_count': 17,
 'geo_enabled': False,
 'has_extended_profile': False,
 'id': 2830050282,
 'id_str': '2830050282',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 35,
 'location': 'Land of Loozers',
 'name': 'LoOzeR Xam',
 'notifications': False,
 'profile_background_color': '000000',
 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_tile': False,
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/2830050282/1426246044',
 'profile_image_url': 'http://pbs.twimg.com/profile_images/573377533367877633/Q_z3mgkb_normal
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/573377533367877633/Q_z3mgkb
 'profile_link_color': '89C9FA',
 'profile_sidebar_border_color': '000000',
 'profile_sidebar_fill_color': '000000',
 'profile_text_color': '000000',
 'profile_use_background_image': False,
 'protected': False,
```

```
  'screen_name': 'LoozerX',
  'statuses_count': 22835,
  'time_zone': None,
  'url': None,
  'utc_offset': None,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:50 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'bit.ly/21h6Tzd',
    'expanded_url': 'http://bit.ly/21h6Tzd',
    'indices': [114, 137],
    'url': 'https://t.co/NuC9B3l7pw'}],
  'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 702604622460944384,
 'id_str': '702604622460944384',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 0,
 'retweeted': False,
 'source': '<a href="http://twitterfeed.com" rel="nofollow">twitterfeed</a>',
 'text': 'TSA: Were committed to indigenous technology for National transformation  SystemSp
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Mon Nov 26 00:39:35 +0000 2012',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'BBM PIN 748BA2BC | BBM Channel C00206169 | We deliver breaking news from a
  'entities': {'description': {'urls': [{'display_url': 'facebook.com/AfricaNewsPress',
      'expanded_url': 'http://facebook.com/AfricaNewsPress',
      'indices': [89, 112],
      'url': 'https://t.co/w8ccJGi98t'}]},
   'url': {'urls': [{'display_url': 'AfricaNewsPress.com',
      'expanded_url': 'http://www.AfricaNewsPress.com',
      'indices': [0, 23],
      'url': 'https://t.co/d7VxwZGxaw'}]}},
  'favourites_count': 9,
  'follow_request_sent': False,
  'followers_count': 4808,
  'following': False,
  'friends_count': 0,
```

```
 'geo_enabled': True,
 'has_extended_profile': False,
 'id': 971051394,
 'id_str': '971051394',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 93,
 'location': 'Nigeria',
 'name': 'Africa Nigeria Press',
 'notifications': False,
 'profile_background_color': '352726',
 'profile_background_image_url': 'http://pbs.twimg.com/profile_background_images/766851362/73
 'profile_background_image_url_https': 'https://pbs.twimg.com/profile_background_images/76685
 'profile_background_tile': True,
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/971051394/1358510375',
 'profile_image_url': 'http://pbs.twimg.com/profile_images/3166965886/28155752aea8d1400bee54
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/3166965886/28155752aea8d140
 'profile_link_color': 'D02B55',
 'profile_sidebar_border_color': '000000',
 'profile_sidebar_fill_color': '252429',
 'profile_text_color': '666666',
 'profile_use_background_image': True,
 'protected': False,
 'screen_name': 'AfricaNewsPress',
 'statuses_count': 331123,
 'time_zone': 'West Central Africa',
 'url': 'https://t.co/d7VxwZGxaw',
 'utc_offset': 3600,
 'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:49 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'bbc.co.uk/news/technolog',
    'expanded_url': 'http://www.bbc.co.uk/news/technology-35648921',
    'indices': [25, 48],
    'url': 'https://t.co/VIgcZ3kIjX'}],
  'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 702604618115715072,
 'id_str': '702604618115715072',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
```

```
 'possibly_sensitive': False,
 'retweet_count': 0,
 'retweeted': False,
 'source': '<a href="http://twitter.com/#!/download/ipad" rel="nofollow">Twitter for iPad</a
 'text': 'Amazing. Check this out! https://t.co/VIgcZ3kIjX',
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Sat Sep 05 11:04:17 +0000 2009',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Spurs fan. Brought up on Steve Perryman and Glenn Hoddle. Love golf, motorb
  'entities': {'description': {'urls': []}},
  'favourites_count': 1477,
  'follow_request_sent': False,
  'followers_count': 162,
  'following': False,
  'friends_count': 646,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 71778418,
  'id_str': '71778418',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 5,
  'location': '',
  'name': 'eddie tribe',
  'notifications': False,
  'profile_background_color': '642D8B',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme10/bg.gif',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme10/bg.gif',
  'profile_background_tile': True,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/71778418/1413491663',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/522846924786126848/2wKrb98W_normal
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/522846924786126848/2wKrb98W
  'profile_link_color': 'FF0000',
  'profile_sidebar_border_color': '65B0DA',
  'profile_sidebar_fill_color': '7AC3EE',
  'profile_text_color': '3D1957',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'eddietribe',
  'statuses_count': 2104,
  'time_zone': None,
  'url': None,
  'utc_offset': None,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:47 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'bit.ly/1nr748y',
    'expanded_url': 'http://bit.ly/1nr748y',
```

```
   'indices': [79, 102],
   'url': 'https://t.co/QaNm9E2Lpt'}],
 'user_mentions': [{'id': 388358316,
   'id_str': '388358316',
   'indices': [3, 18],
   'name': 'Craig Wigginton',
   'screen_name': 'CraigWigginton'}]},
'favorite_count': 0,
'favorited': False,
'geo': None,
'id': 702604610712838144,
'id_str': '702604610712838144',
'in_reply_to_screen_name': None,
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': None,
'possibly_sensitive': False,
'retweet_count': 1,
'retweeted': False,
'retweeted_status': {'contributors': None,
 'coordinates': None,
 'created_at': 'Thu Feb 04 18:48:41 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'bit.ly/1nr748y',
    'expanded_url': 'http://bit.ly/1nr748y',
    'indices': [59, 82],
    'url': 'https://t.co/QaNm9E2Lpt'}],
  'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 695318072677179392,
 'id_str': '695318072677179392',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 1,
 'retweeted': False,
 'source': '<a href="http://www.linkedin.com/" rel="nofollow">LinkedIn</a>',
 'text': "2016 Telecom Industry Outlook for the US.  Here's my take. https://t.co/QaNm9E2Lp
 'truncated': False,
```

```
 'user': {'contributors_enabled': False,
  'created_at': 'Mon Oct 10 16:26:11 +0000 2011',
  'default_profile': True,
  'default_profile_image': False,
  'description': 'Deloitte Telecom US National Sector Leader. Focus on helping the Telecom c
  'entities': {'description': {'urls': []}},
  'favourites_count': 51,
  'follow_request_sent': False,
  'followers_count': 575,
  'following': False,
  'friends_count': 79,
  'geo_enabled': True,
  'has_extended_profile': False,
  'id': 388358316,
  'id_str': '388358316',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 62,
  'location': '',
  'name': 'Craig Wigginton',
  'notifications': False,
  'profile_background_color': 'C0DEED',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_tile': False,
  'profile_image_url': 'http://pbs.twimg.com/profile_images/2165138568/Wigginton_-_business_c
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/2165138568/Wigginton_-_bus
  'profile_link_color': '0084B4',
  'profile_sidebar_border_color': 'C0DEED',
  'profile_sidebar_fill_color': 'DDEEF6',
  'profile_text_color': '333333',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'CraigWigginton',
  'statuses_count': 617,
  'time_zone': 'Eastern Time (US & Canada)',
  'url': None,
  'utc_offset': -18000,
  'verified': False}},
'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
'text': "RT @CraigWigginton: 2016 Telecom Industry Outlook for the US.  Here's my take. http
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Mon Apr 13 15:37:51 +0000 2009',
 'default_profile': False,
 'default_profile_image': False,
 'description': 'Focusing on new and counter-intuitive thinking and  always enjoy the challe
 'entities': {'description': {'urls': []}},
 'favourites_count': 50,
 'follow_request_sent': False,
 'followers_count': 308,
 'following': False,
 'friends_count': 395,
```

```
  'geo_enabled': True,
  'has_extended_profile': True,
  'id': 30887106,
  'id_str': '30887106',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 23,
  'location': 'Johannesburg',
  'name': 'Mark Casey',
  'notifications': False,
  'profile_background_color': 'EDECE9',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme3/bg.gif',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme3/bg.gif',
  'profile_background_tile': False,
  'profile_image_url': 'http://pbs.twimg.com/profile_images/182573502/Mark_Casey_0075_normal.j
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/182573502/Mark_Casey_0075_n
  'profile_link_color': '088253',
  'profile_sidebar_border_color': 'D3D2CF',
  'profile_sidebar_fill_color': 'E3E2DE',
  'profile_text_color': '634047',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'Cathsaigh',
  'statuses_count': 422,
  'time_zone': 'Pretoria',
  'url': None,
  'utc_offset': 7200,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:45 +0000 2016',
 'entities': {'hashtags': [{'indices': [0, 11], 'text': 'technology'}],
  'symbols': [],
  'urls': [{'display_url': 'bit.ly/1SV6ei7',
    'expanded_url': 'http://bit.ly/1SV6ei7',
    'indices': [114, 137],
    'url': 'https://t.co/Z8ZBBvDs75'}],
  'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 702604603850756096,
 'id_str': '702604603850756096',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
```

```
 'retweet_count': 0,
 'retweeted': False,
 'source': '<a href="http://twitterfeed.com" rel="nofollow">twitterfeed</a>',
 'text': '#technology Google Fiber coming to San Francisco, using someone elses fiber:   sub
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Fri Jan 06 22:37:06 +0000 2012',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Working hard for my clients,I am a Realtor in the Houston, TX area. Perfor
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'texasrealestateunlimited.com',
      'expanded_url': 'http://texasrealestateunlimited.com',
      'indices': [0, 23],
      'url': 'https://t.co/6gmAl8kHda'}]}},
  'favourites_count': 27,
  'follow_request_sent': False,
  'followers_count': 18259,
  'following': False,
  'friends_count': 120,
  'geo_enabled': False,
  'has_extended_profile': True,
  'id': 457003916,
  'id_str': '457003916',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 70,
  'location': 'Houston, TX',
  'name': 'CiCi Rodriguez',
  'notifications': False,
  'profile_background_color': '022330',
  'profile_background_image_url': 'http://pbs.twimg.com/profile_background_images/409889034/ho
  'profile_background_image_url_https': 'https://pbs.twimg.com/profile_background_images/40988
  'profile_background_tile': True,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/457003916/1443594905',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/1898180594/C_Coleman-Headshot_norm
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1898180594/C_Coleman-Headsh
  'profile_link_color': '0084B4',
  'profile_sidebar_border_color': 'A8C7F7',
  'profile_sidebar_fill_color': 'C0DFEC',
  'profile_text_color': '333333',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'txrelocator',
  'statuses_count': 6903,
  'time_zone': 'Central Time (US & Canada)',
  'url': 'https://t.co/6gmAl8kHda',
  'utc_offset': -21600,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:44 +0000 2016',
 'entities': {'hashtags': [],
```

```
 'symbols': [],
 'urls': [{'display_url': 'nyti.ms/1oHK4n5',
   'expanded_url': 'http://nyti.ms/1oHK4n5',
   'indices': [139, 140],
   'url': 'https://t.co/W4hkw2OkYl'}],
 'user_mentions': [{'id': 14434070,
   'id_str': '14434070',
   'indices': [3, 15],
   'name': 'NYTimes Bits',
   'screen_name': 'nytimesbits'}]},
'favorite_count': 0,
'favorited': False,
'geo': None,
'id': 702604596032753664,
'id_str': '702604596032753664',
'in_reply_to_screen_name': None,
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': None,
'possibly_sensitive': False,
'retweet_count': 8,
'retweeted': False,
'retweeted_status': {'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:17:01 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'nyti.ms/1oHK4n5',
    'expanded_url': 'http://nyti.ms/1oHK4n5',
    'indices': [116, 139],
    'url': 'https://t.co/W4hkw2OkYl'}],
  'user_mentions': []},
 'favorite_count': 7,
 'favorited': False,
 'geo': None,
 'id': 702603157281611776,
 'id_str': '702603157281611776',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 8,
 'retweeted': False,
```

```
 'source': '<a href="http://www.socialflow.com" rel="nofollow">SocialFlow</a>',
 'text': 'Why the Apple case matters: With the Internet of Things, every home appliance coul
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Fri Apr 18 15:19:26 +0000 2008',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Tech news and analysis, plus interesting links and retweets from Times te
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'nytimes.com/technology',
      'expanded_url': 'http://nytimes.com/technology',
      'indices': [0, 22],
      'url': 'http://t.co/XaZCeioBAC'}]}},
  'favourites_count': 293,
  'follow_request_sent': False,
  'followers_count': 275129,
  'following': False,
  'friends_count': 155,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 14434070,
  'id_str': '14434070',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 10413,
  'location': '',
  'name': 'NYTimes Bits',
  'notifications': False,
  'profile_background_color': '9AE4E8',
  'profile_background_image_url': 'http://pbs.twimg.com/profile_background_images/4780380/twi
  'profile_background_image_url_https': 'https://pbs.twimg.com/profile_background_images/4780
  'profile_background_tile': True,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/14434070/1360101471',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/2078138859/NYT_Twitter_bits_norma
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/2078138859/NYT_Twitter_bit
  'profile_link_color': '0000FF',
  'profile_sidebar_border_color': '87BC44',
  'profile_sidebar_fill_color': 'E0FF92',
  'profile_text_color': '000000',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'nytimesbits',
  'statuses_count': 22950,
  'time_zone': 'Eastern Time (US & Canada)',
  'url': 'http://t.co/XaZCeioBAC',
  'utc_offset': -18000,
  'verified': True}},
'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a
'text': 'RT @nytimesbits: Why the Apple case matters: With the Internet of Things, every hom
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Tue Nov 04 14:51:05 +0000 2014',
 'default_profile': True,
```

```
  'default_profile_image': False,
  'description': 'DBT: Eine Community von APA & sd one fr die digitale Kommunikations-, IT- u
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'dbt.at',
      'expanded_url': 'http://www.dbt.at',
      'indices': [0, 22],
      'url': 'http://t.co/NmuPuhpVQz'}]}},
  'favourites_count': 705,
  'follow_request_sent': False,
  'followers_count': 364,
  'following': False,
  'friends_count': 211,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 2860806191,
  'id_str': '2860806191',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'de',
  'listed_count': 41,
  'location': 'Vienna, Austria',
  'name': 'dbt_at',
  'notifications': False,
  'profile_background_color': 'C0DEED',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_tile': False,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/2860806191/1415698936',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/529970865656438784/ppe1tIVi_normal
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/529970865656438784/ppe1tIVi
  'profile_link_color': '0084B4',
  'profile_sidebar_border_color': 'C0DEED',
  'profile_sidebar_fill_color': 'DDEEF6',
  'profile_text_color': '333333',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'dbt_at',
  'statuses_count': 1787,
  'time_zone': 'Ljubljana',
  'url': 'http://t.co/NmuPuhpVQz',
  'utc_offset': 3600,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:43 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [],
  'user_mentions': [{'id': 311399175,
    'id_str': '311399175',
    'indices': [0, 10],
    'name': 'Dame Sir Ron',
    'screen_name': 'paddo_ron'}]},
 'favorite_count': 0,
```

```
'favorited': False,
'geo': None,
'id': 702604593197391877,
'id_str': '702604593197391877',
'in_reply_to_screen_name': 'paddo_ron',
'in_reply_to_status_id': 702603642990252033,
'in_reply_to_status_id_str': '702603642990252033',
'in_reply_to_user_id': 311399175,
'in_reply_to_user_id_str': '311399175',
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': None,
'retweet_count': 0,
'retweeted': False,
'source': '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>',
'text': '@paddo_ron The TECHNOLOGY, the INTERNET is giving these cultures a sudden glimse at
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Sat May 11 18:59:45 +0000 2013',
 'default_profile': False,
 'default_profile_image': False,
 'description': 'Whoever made this chicken should have his hands cut off then have his feet
 'entities': {'description': {'urls': []},
  'url': {'urls': [{'display_url': 'youtube.com/watch?v=_6_fGd',
     'expanded_url': 'https://www.youtube.com/watch?v=_6_fGdVpJ1U',
     'indices': [0, 23],
     'url': 'https://t.co/HZmFsq16L1'}]}},
 'favourites_count': 574,
 'follow_request_sent': False,
 'followers_count': 151,
 'following': False,
 'friends_count': 615,
 'geo_enabled': True,
 'has_extended_profile': False,
 'id': 1421340492,
 'id_str': '1421340492',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 8,
 'location': 'Isla Island',
 'name': 'El Generalissimo',
 'notifications': False,
 'profile_background_color': '000000',
 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_tile': False,
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/1421340492/1456341516',
 'profile_image_url': 'http://pbs.twimg.com/profile_images/695918409867927553/rnjvSIk5_normal
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/695918409867927553/rnjvSIk5
 'profile_link_color': '334422',
 'profile_sidebar_border_color': '000000',
 'profile_sidebar_fill_color': '000000',
```

```
  'profile_text_color': '000000',
  'profile_use_background_image': False,
  'protected': False,
  'screen_name': 'GeneralissimoEl',
  'statuses_count': 391,
  'time_zone': 'Atlantic Time (Canada)',
  'url': 'https://t.co/HZmFsq16L1',
  'utc_offset': -14400,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:41 +0000 2016',
 'entities': {'hashtags': [{'indices': [32, 43], 'text': 'Healthcare'},
   {'indices': [44, 55], 'text': 'Technology'}],
  'symbols': [],
  'urls': [{'display_url': 'superbcrew.com/startel-integr',
    'expanded_url': 'http://www.superbcrew.com/startel-integrates-with-leading-healthcare-tec
    'indices': [66, 89],
    'url': 'https://t.co/a6UY8HGwIT'}],
  'user_mentions': []},
 'favorite_count': 1,
 'favorited': False,
 'geo': None,
 'id': 702604586251583488,
 'id_str': '702604586251583488',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 0,
 'retweeted': False,
 'source': '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>',
 'text': 'Startel Integrates With Leading #Healthcare #Technology Providers https://t.co/a6U
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Mon Dec 16 12:12:11 +0000 2013',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Tech news and stories. Website covering technology and internet companies.
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'SuperbCrew.com',
      'expanded_url': 'http://SuperbCrew.com',
      'indices': [0, 22],
      'url': 'http://t.co/jlPC6fu7Ax'}]}},
  'favourites_count': 16611,
  'follow_request_sent': False,
  'followers_count': 39567,
  'following': False,
```

```
 'friends_count': 28915,
 'geo_enabled': False,
 'has_extended_profile': False,
 'id': 2248660566,
 'id_str': '2248660566',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 1233,
 'location': 'Email: Hello@SuperbCrew.com',
 'name': 'SuperbCrew.com',
 'notifications': False,
 'profile_background_color': 'C0DEED',
 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_tile': False,
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/2248660566/1399290222',
 'profile_image_url': 'http://pbs.twimg.com/profile_images/484367594028158976/MB2-pKsg_normal
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/484367594028158976/MB2-pKsg
 'profile_link_color': '0084B4',
 'profile_sidebar_border_color': '000000',
 'profile_sidebar_fill_color': 'DDEEF6',
 'profile_text_color': '333333',
 'profile_use_background_image': True,
 'protected': False,
 'screen_name': 'SuperbCrew_com',
 'statuses_count': 4690,
 'time_zone': 'Eastern Time (US & Canada)',
 'url': 'http://t.co/jlPC6fu7Ax',
 'utc_offset': -18000,
 'verified': False}},
{'contributors': None,
 'coordinates': {'coordinates': [-99.133208, 19.4326077], 'type': 'Point'},
 'created_at': 'Wed Feb 24 21:22:41 +0000 2016',
 'entities': {'hashtags': [{'indices': [15, 24], 'text': 'MexicoDF'},
   {'indices': [25, 29], 'text': 'job'},
   {'indices': [111, 117], 'text': 'Sales'},
   {'indices': [118, 125], 'text': 'Hiring'},
   {'indices': [126, 136], 'text': 'CareerArc'}],
  'symbols': [],
  'urls': [{'display_url': 'bit.ly/1R3Xxji',
    'expanded_url': 'http://bit.ly/1R3Xxji',
    'indices': [87, 110],
    'url': 'https://t.co/xqinhBwg4a'}],
  'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'geo': {'coordinates': [19.4326077, -99.133208], 'type': 'Point'},
 'id': 702604585907531776,
 'id_str': '702604585907531776',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
```

```
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': {'attributes': {},
 'bounding_box': {'coordinates': [[[-99.1843501, 19.3998346],
    [-99.122382, 19.3998346],
    [-99.122382, 19.4658366],
    [-99.1843501, 19.4658366]]],
  'type': 'Polygon'},
 'contained_within': [],
 'country': 'Mxico',
 'country_code': 'MX',
 'full_name': 'Cuauhtmoc, Distrito Federal',
 'id': 'bfc35dcc7e63252a',
 'name': 'Cuauhtmoc',
 'place_type': 'city',
 'url': 'https://api.twitter.com/1.1/geo/id/bfc35dcc7e63252a.json'},
'possibly_sensitive': False,
'retweet_count': 0,
'retweeted': False,
'source': '<a href="http://www.tweetmyjobs.com" rel="nofollow">TweetMyJOBS</a>',
'text': 'See our latest #MexicoDF #job and click to apply: Technology Sales Representative I
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Thu Jul 12 17:13:18 +0000 2012',
 'default_profile': False,
 'default_profile_image': False,
 'description': 'Follow this account for geo-targeted Sales job tweets in Mexico from TweetM
 'entities': {'description': {'urls': []},
  'url': {'urls': [{'display_url': 'tweetmyjobs.com',
     'expanded_url': 'http://tweetmyjobs.com',
     'indices': [0, 22],
     'url': 'http://t.co/I9eYZ8Ji6L'}]}},
 'favourites_count': 0,
 'follow_request_sent': False,
 'followers_count': 349,
 'following': False,
 'friends_count': 280,
 'geo_enabled': True,
 'has_extended_profile': False,
 'id': 633894355,
 'id_str': '633894355',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 48,
 'location': 'Mexico',
 'name': 'Mex Sales',
 'notifications': False,
 'profile_background_color': '253956',
 'profile_background_image_url': 'http://pbs.twimg.com/profile_background_images/613240362/Tw
 'profile_background_image_url_https': 'https://pbs.twimg.com/profile_background_images/61324
 'profile_background_tile': False,
```

```
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/633894355/1351015907',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/2429189271/Logo_tmj_new2b_normal.p
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/2429189271/Logo_tmj_new2b_n
  'profile_link_color': '0084B4',
  'profile_sidebar_border_color': 'C0DEED',
  'profile_sidebar_fill_color': 'DDEEF6',
  'profile_text_color': '333333',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'tmjmexsales',
  'statuses_count': 75,
  'time_zone': 'Arizona',
  'url': 'http://t.co/I9eYZ8Ji6L',
  'utc_offset': -25200,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:41 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'sumo.ly/fUyI',
    'expanded_url': 'http://sumo.ly/fUyI',
    'indices': [117, 140],
    'url': 'https://t.co/53Ym5Q90AY'}],
  'user_mentions': []},
 'favorite_count': 1,
 'favorited': False,
 'geo': None,
 'id': 702604583076429824,
 'id_str': '702604583076429824',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 0,
 'retweeted': False,
 'source': '<a href="http://www.linkedin.com/" rel="nofollow">LinkedIn</a>',
 'text': 'The environmental tech summit is going to tackle two main questions. WHY technolog
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Thu Aug 13 02:55:53 +0000 2009',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'I am an #Environmental Scientist and a Thinking #Environmentalist. Follow m
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'about.me/nicholasmoran',
      'expanded_url': 'http://about.me/nicholasmoran',
      'indices': [0, 23],
```

```
      'url': 'https://t.co/y1wbVIm5NN'}]}},
  'favourites_count': 52,
  'follow_request_sent': False,
  'followers_count': 161,
  'following': False,
  'friends_count': 309,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 65252834,
  'id_str': '65252834',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 10,
  'location': 'Miami, FL',
  'name': 'Nicholas Moran',
  'notifications': False,
  'profile_background_color': '000000',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_tile': False,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/65252834/1452659917',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/687130390973280256/0gae8yD7_normal
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/687130390973280256/0gae8yD7
  'profile_link_color': '4A913C',
  'profile_sidebar_border_color': '000000',
  'profile_sidebar_fill_color': '000000',
  'profile_text_color': '000000',
  'profile_use_background_image': False,
  'protected': False,
  'screen_name': 'namoran',
  'statuses_count': 894,
  'time_zone': 'Eastern Time (US & Canada)',
  'url': 'https://t.co/y1wbVIm5NN',
  'utc_offset': -18000,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:40 +0000 2016',
 'entities': {'hashtags': [{'indices': [139, 140], 'text': 'tech'},
   {'indices': [139, 140], 'text': 'technology'}],
  'symbols': [],
  'urls': [{'display_url': 'sociably.me/Kbt2DK',
    'expanded_url': 'http://sociably.me/Kbt2DK',
    'indices': [115, 138],
    'url': 'https://t.co/3RhoRmEi1M'}],
  'user_mentions': [{'id': 23954895,
    'id_str': '23954895',
    'indices': [3, 16],
    'name': 'Jason Hartsoe',
    'screen_name': 'jasonhartsoe'}]},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
```

```
'id': 702604581147172864,
'id_str': '702604581147172864',
'in_reply_to_screen_name': None,
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': None,
'possibly_sensitive': False,
'retweet_count': 1,
'retweeted': False,
'retweeted_status': {'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:18:56 +0000 2016',
 'entities': {'hashtags': [{'indices': [122, 127], 'text': 'tech'},
   {'indices': [128, 139], 'text': 'technology'}],
  'symbols': [],
  'urls': [{'display_url': 'sociably.me/Kbt2DK',
    'expanded_url': 'http://sociably.me/Kbt2DK',
    'indices': [97, 120],
    'url': 'https://t.co/3RhoRmEi1M'}],
  'user_mentions': []},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 702603640167501824,
 'id_str': '702603640167501824',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 1,
 'retweeted': False,
 'source': '<a href="http://dlvr.it" rel="nofollow">dlvr.it</a>',
 'text': 'Computer Science Is Now A High School Graduation Requirement In Chicagos Public S
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Thu Mar 12 14:19:10 +0000 2009',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Founder and CEO of @QRlitx. Web and Mobile Application Developer. QR Code
  'entities': {'description': {'urls': []}},
  'favourites_count': 0,
  'follow_request_sent': False,
  'followers_count': 1588,
```

```
  'following': False,
  'friends_count': 73,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 23954895,
  'id_str': '23954895',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 815,
  'location': 'Hickory, NC USA',
  'name': 'Jason Hartsoe',
  'notifications': False,
  'profile_background_color': 'CFCFCF',
  'profile_background_image_url': 'http://pbs.twimg.com/profile_background_images/348435937/c
  'profile_background_image_url_https': 'https://pbs.twimg.com/profile_background_images/3484
  'profile_background_tile': True,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/23954895/1399628122',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/464699985032773632/Ijsr_hOo_norma
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/464699985032773632/Ijsr_hO
  'profile_link_color': '292848',
  'profile_sidebar_border_color': '292848',
  'profile_sidebar_fill_color': 'F5F5F5',
  'profile_text_color': '5C5B75',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'jasonhartsoe',
  'statuses_count': 362112,
  'time_zone': 'Eastern Time (US & Canada)',
  'url': None,
  'utc_offset': -18000,
  'verified': False}},
'source': '<a href="https://roundteam.co" rel="nofollow">RoundTeam</a>',
'text': 'RT @jasonhartsoe: Computer Science Is Now A High School Graduation Requirement In C
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Thu Jul 30 05:17:41 +0000 2015',
 'default_profile': True,
 'default_profile_image': False,
 'description': 'Leader, Mentor, Online Coach, Momentum Creator, 10X Your Online Business',
 'entities': {'description': {'urls': []},
  'url': {'urls': [{'display_url': 'myroadto6figures.com',
     'expanded_url': 'https://www.myroadto6figures.com',
     'indices': [0, 23],
     'url': 'https://t.co/nYf7IXzJLJ'}]}},
 'favourites_count': 137,
 'follow_request_sent': False,
 'followers_count': 2353,
 'following': False,
 'friends_count': 4311,
 'geo_enabled': False,
 'has_extended_profile': False,
 'id': 3395092114,
 'id_str': '3395092114',
```

```
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 598,
  'location': 'New York, NY',
  'name': 'Teddy Awa',
  'notifications': False,
  'profile_background_color': 'C0DEED',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_tile': False,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/3395092114/1455752677',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/700103469521313792/g7mQ70_g_normal
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/700103469521313792/g7mQ70_g
  'profile_link_color': '0084B4',
  'profile_sidebar_border_color': 'C0DEED',
  'profile_sidebar_fill_color': 'DDEEF6',
  'profile_text_color': '333333',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'Teddyborntowin',
  'statuses_count': 17248,
  'time_zone': None,
  'url': 'https://t.co/nYf7IXzJLJ',
  'utc_offset': None,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:40 +0000 2016',
 'entities': {'hashtags': [{'indices': [118, 133],
    'text': 'socialmedialaw'}],
  'symbols': [],
  'urls': [{'display_url': 'defenseone.com/technology/201',
    'expanded_url': 'http://www.defenseone.com/technology/2016/02/military-funded-study-predi
    'indices': [80, 103],
    'url': 'https://t.co/1UOYOI0mWK'}],
  'user_mentions': [{'id': 17276849,
    'id_str': '17276849',
    'indices': [3, 16],
    'name': 'Kent Ninomiya',
    'screen_name': 'kentninomiya'},
   {'id': 17276849,
    'id_str': '17276849',
    'indices': [104, 117],
    'name': 'Kent Ninomiya',
    'screen_name': 'kentninomiya'}]},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 702604580216033280,
 'id_str': '702604580216033280',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
```

```
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': None,
'possibly_sensitive': False,
'retweet_count': 1,
'retweeted': False,
'retweeted_status': {'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:19:15 +0000 2016',
 'entities': {'hashtags': [{'indices': [100, 115],
    'text': 'socialmedialaw'}],
  'symbols': [],
  'urls': [{'display_url': 'defenseone.com/technology/201',
    'expanded_url': 'http://www.defenseone.com/technology/2016/02/military-funded-study-pred
    'indices': [62, 85],
    'url': 'https://t.co/1UOYOIOmWK'}],
  'user_mentions': [{'id': 17276849,
    'id_str': '17276849',
    'indices': [86, 99],
    'name': 'Kent Ninomiya',
    'screen_name': 'kentninomiya'}]},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 702603722862366722,
 'id_str': '702603722862366722',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 1,
 'retweeted': False,
 'source': '<a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>',
 'text': 'Military-Funded Study Predicts When Youll Protest on Twitter https://t.co/1UOYOIOm
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Mon Nov 10 00:42:34 +0000 2008',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Social Media Lawyer and Journalist focused on social media/mass media con
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'ninomiyalaw.com',
      'expanded_url': 'http://ninomiyalaw.com',
      'indices': [0, 23],
      'url': 'https://t.co/qAn8Woh28M'}]}},
```

```
  'favourites_count': 38,
  'follow_request_sent': False,
  'followers_count': 1053,
  'following': False,
  'friends_count': 4928,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 17276849,
  'id_str': '17276849',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 9,
  'location': 'Texas, USA',
  'name': 'Kent Ninomiya',
  'notifications': False,
  'profile_background_color': '000000',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_tile': False,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/17276849/1451499418',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/698266197972791296/8gMuwbLm_norma
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/698266197972791296/8gMuwbl
  'profile_link_color': '3B5998',
  'profile_sidebar_border_color': '000000',
  'profile_sidebar_fill_color': '000000',
  'profile_text_color': '000000',
  'profile_use_background_image': False,
  'protected': False,
  'screen_name': 'kentninomiya',
  'statuses_count': 584,
  'time_zone': 'Central Time (US & Canada)',
  'url': 'https://t.co/qAn8Woh28M',
  'utc_offset': -21600,
  'verified': False}},
'source': '<a href="http://www.netvibes.com/" rel="nofollow">Netvibes Widget</a>',
'text': 'RT @kentninomiya: Military-Funded Study Predicts When Youll Protest on Twitter htt
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Mon Feb 08 06:44:05 +0000 2016',
 'default_profile': False,
 'default_profile_image': False,
 'description': 'pudding tastes better with a plastic spoon',
 'entities': {'description': {'urls': []}},
 'favourites_count': 0,
 'follow_request_sent': False,
 'followers_count': 41,
 'following': False,
 'friends_count': 127,
 'geo_enabled': False,
 'has_extended_profile': False,
 'id': 4887156766,
 'id_str': '4887156766',
 'is_translation_enabled': False,
```

```
  'is_translator': False,
  'lang': 'en',
  'listed_count': 47,
  'location': '',
  'name': 'Angie',
  'notifications': False,
  'profile_background_color': '000000',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_tile': False,
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/4887156766/1455996890',
  'profile_image_url': 'http://pbs.twimg.com/profile_images/701127947630080001/x-4v0apn_normal
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/701127947630080001/x-4v0apn
  'profile_link_color': 'E81C4F',
  'profile_sidebar_border_color': '000000',
  'profile_sidebar_fill_color': '000000',
  'profile_text_color': '000000',
  'profile_use_background_image': False,
  'protected': False,
  'screen_name': 'angelicabroni',
  'statuses_count': 251,
  'time_zone': 'Pacific Time (US & Canada)',
  'url': None,
  'utc_offset': -28800,
  'verified': False}},
{'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:22:40 +0000 2016',
 'entities': {'hashtags': [{'indices': [139, 140], 'text': 'fb'}],
  'symbols': [],
  'urls': [{'display_url': 'deepmind.com/health',
    'expanded_url': 'http://deepmind.com/health',
    'indices': [89, 112],
    'url': 'https://t.co/pfJnxjCkLB'}],
  'user_mentions': [{'id': 5715682,
    'id_str': '5715682',
    'indices': [3, 17],
    'name': 'Julian Huppert',
    'screen_name': 'julianhuppert'}]},
 'favorite_count': 0,
 'favorited': False,
 'geo': None,
 'id': 702604579188416512,
 'id_str': '702604579188416512',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
```

```
'retweet_count': 1,
'retweeted': False,
'retweeted_status': {'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 19:45:15 +0000 2016',
 'entities': {'hashtags': [{'indices': [133, 136], 'text': 'fb'}],
  'symbols': [],
  'urls': [{'display_url': 'deepmind.com/health',
    'expanded_url': 'http://deepmind.com/health',
    'indices': [70, 93],
    'url': 'https://t.co/pfJnxjCkLB'}],
  'user_mentions': []},
 'favorite_count': 2,
 'favorited': False,
 'geo': None,
 'id': 702580066325291012,
 'id_str': '702580066325291012',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 1,
 'retweeted': False,
 'source': '<a href="http://www.echofon.com/" rel="nofollow">Echofon</a>',
 'text': 'What are your reactions to the announcement today of DeepMind Health? https://t.c
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Wed May 02 15:13:31 +0000 2007',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Former MP for Cambridge \nLiberal Democrat',
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'julianhuppert.org.uk',
      'expanded_url': 'http://www.julianhuppert.org.uk',
      'indices': [0, 22],
      'url': 'http://t.co/rJhUUDlXK2'}]}},
  'favourites_count': 66,
  'follow_request_sent': False,
  'followers_count': 19598,
  'following': False,
  'friends_count': 190,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 5715682,
  'id_str': '5715682',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
```

```
  'listed_count': 809,
  'location': 'Cambridge, UK',
  'name': 'Julian Huppert',
  'notifications': False,
  'profile_background_color': '9AE4E8',
  'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
  'profile_background_tile': False,
  'profile_image_url': 'http://pbs.twimg.com/profile_images/1365532172/Huppert_image_normal.j
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1365532172/Huppert_image_n
  'profile_link_color': 'FDBC30',
  'profile_sidebar_border_color': '87BC44',
  'profile_sidebar_fill_color': 'E0FF92',
  'profile_text_color': '000000',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'julianhuppert',
  'statuses_count': 67403,
  'time_zone': 'London',
  'url': 'http://t.co/rJhUUDlXK2',
  'utc_offset': 0,
  'verified': True}},
'source': '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android
'text': 'RT @julianhuppert: What are your reactions to the announcement today of DeepMind H
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Tue Dec 22 23:21:09 +0000 2009',
 'default_profile': True,
 'default_profile_image': False,
 'description': 'Software engineer @qinec; Lib Dem; cyclist; parent.',
 'entities': {'description': {'urls': []}},
 'favourites_count': 6757,
 'follow_request_sent': False,
 'followers_count': 257,
 'following': False,
 'friends_count': 644,
 'geo_enabled': True,
 'has_extended_profile': False,
 'id': 98743094,
 'id_str': '98743094',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 25,
 'location': 'Loughton, England',
 'name': 'George Lund',
 'notifications': False,
 'profile_background_color': 'C0DEED',
 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
 'profile_background_tile': False,
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/98743094/1355527961',
 'profile_image_url': 'http://pbs.twimg.com/profile_images/700307960732393472/jw4-R6VG_normal
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/700307960732393472/jw4-R6V
```

```
    'profile_link_color': '0084B4',
    'profile_sidebar_border_color': 'C0DEED',
    'profile_sidebar_fill_color': 'DDEEF6',
    'profile_text_color': '333333',
    'profile_use_background_image': True,
    'protected': False,
    'screen_name': 'georgelunduk',
    'statuses_count': 4709,
    'time_zone': 'London',
    'url': None,
    'utc_offset': 0,
    'verified': False}}]}
```

This has some helpful metadata about our request, like a url where we can get the next batch of results from Twitter for the same query:

```
In [46]: data = r.json()
         data['search_metadata']

Out[46]: {'completed_in': 0.072,
          'count': 15,
          'max_id': 702604626965696513,
          'max_id_str': '702604626965696513',
          'next_results': '?max_id=702604579188416511&q=technology&include_entities=1',
          'query': 'technology',
          'refresh_url': '?since_id=702604626965696513&q=technology&include_entities=1',
          'since_id': 0,
          'since_id_str': '0'}
```

The tweets that we want are under the key "statuses"

```
In [47]: statuses = data['statuses']
         statuses[0]

Out[47]: {'contributors': None,
          'coordinates': None,
          'created_at': 'Wed Feb 24 21:22:51 +0000 2016',
          'entities': {'hashtags': [],
           'symbols': [],
           'urls': [{'display_url': 'nyti.ms/1WJf4NV',
             'expanded_url': 'http://nyti.ms/1WJf4NV',
             'indices': [139, 140],
             'url': 'https://t.co/kFDYwJoiV5'}],
           'user_mentions': [{'id': 1754641,
             'id_str': '1754641',
             'indices': [3, 19],
             'name': 'NYT Business',
             'screen_name': 'nytimesbusiness'}]},
          'favorite_count': 0,
          'favorited': False,
          'geo': None,
          'id': 702604626965696513,
          'id_str': '702604626965696513',
          'in_reply_to_screen_name': None,
          'in_reply_to_status_id': None,
```

```
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': None,
'possibly_sensitive': False,
'retweet_count': 7,
'retweeted': False,
'retweeted_status': {'contributors': None,
 'coordinates': None,
 'created_at': 'Wed Feb 24 21:17:01 +0000 2016',
 'entities': {'hashtags': [],
  'symbols': [],
  'urls': [{'display_url': 'nyti.ms/1WJf4NV',
    'expanded_url': 'http://nyti.ms/1WJf4NV',
    'indices': [116, 139],
    'url': 'https://t.co/kFDYwJoiV5'}],
  'user_mentions': []},
 'favorite_count': 6,
 'favorited': False,
 'geo': None,
 'id': 702603158221144064,
 'id_str': '702603158221144064',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'en',
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'place': None,
 'possibly_sensitive': False,
 'retweet_count': 7,
 'retweeted': False,
 'source': '<a href="http://www.socialflow.com" rel="nofollow">SocialFlow</a>',
 'text': 'Why the Apple case matters: With the Internet of Things, every home appliance could
 'truncated': False,
 'user': {'contributors_enabled': False,
  'created_at': 'Wed Mar 21 14:49:39 +0000 2007',
  'default_profile': False,
  'default_profile_image': False,
  'description': 'Business news from The New York Times.',
  'entities': {'description': {'urls': []},
   'url': {'urls': [{'display_url': 'nytimes.com/business',
      'expanded_url': 'http://nytimes.com/business',
      'indices': [0, 22],
      'url': 'http://t.co/OKrfdBy4ch'}]}},
  'favourites_count': 789,
  'follow_request_sent': False,
  'followers_count': 728646,
  'following': False,
```

```
  'friends_count': 617,
  'geo_enabled': False,
  'has_extended_profile': False,
  'id': 1754641,
  'id_str': '1754641',
  'is_translation_enabled': False,
  'is_translator': False,
  'lang': 'en',
  'listed_count': 13338,
  'location': 'New York, NY',
  'name': 'NYT Business',
  'notifications': False,
  'profile_background_color': 'FFFFFF',
  'profile_background_image_url': 'http://pbs.twimg.com/profile_background_images/4433659/twitt
  'profile_background_image_url_https': 'https://pbs.twimg.com/profile_background_images/443365
  'profile_background_tile': True,
  'profile_image_url': 'http://pbs.twimg.com/profile_images/2037622389/NYT_Twitter_Business_nor
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/2037622389/NYT_Twitter_Busin
  'profile_link_color': '004276',
  'profile_sidebar_border_color': '323232',
  'profile_sidebar_fill_color': 'E7EFF8',
  'profile_text_color': '000000',
  'profile_use_background_image': True,
  'protected': False,
  'screen_name': 'nytimesbusiness',
  'statuses_count': 123516,
  'time_zone': 'Eastern Time (US & Canada)',
  'url': 'http://t.co/OKrfdBy4ch',
  'utc_offset': -18000,
  'verified': False}},
'source': '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a
'text': 'RT @nytimesbusiness: Why the Apple case matters: With the Internet of Things, every I
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Wed Sep 21 19:57:48 +0000 2011',
 'default_profile': False,
 'default_profile_image': False,
 'description': 'Solitary and likes reading. Oh cocktails too. People not so much.',
 'entities': {'description': {'urls': []}},
 'favourites_count': 457,
 'follow_request_sent': False,
 'followers_count': 1453,
 'following': False,
 'friends_count': 2075,
 'geo_enabled': True,
 'has_extended_profile': True,
 'id': 377576061,
 'id_str': '377576061',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 32,
 'location': 'Lagos, Nigeria',
 'name': 'Oluwaseun Esq.',
```

```
 'notifications': False,
 'profile_background_color': '9AE4E8',
 'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme16/bg.gif',
 'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme16/bg.gif',
 'profile_background_tile': False,
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/377576061/1400343430',
 'profile_image_url': 'http://pbs.twimg.com/profile_images/669235191160918016/EQ09b_UK_normal.j
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/669235191160918016/EQ09b_UK_n
 'profile_link_color': '0084B4',
 'profile_sidebar_border_color': 'BDDCAD',
 'profile_sidebar_fill_color': 'DDFFCC',
 'profile_text_color': '333333',
 'profile_use_background_image': True,
 'protected': False,
 'screen_name': 'OluwaseunEsq',
 'statuses_count': 45657,
 'time_zone': 'Africa/Lagos',
 'url': None,
 'utc_offset': 3600,
 'verified': False}}
```

This is one tweet.

> Depending on which tweet this is, you may or may not see that Twitter automatically pulls out links and mentions and gives you their index location in the raw tweet string

Twitter gives you a whole lot of information about their users, including geographical coordinates, the device they are tweeting from, and links to their photographs.

Twitter supports what it calls query operators, which modify the search behavior. For example, if you want to search for tweets where a particular user is mentioned, include the at-sign, `@`, followed by the username. To search for tweets sent to a particular user, use `to:username`. For tweets from a particular user, `from:username`. For hashtags, use `#hashtag`.

For a complete set of options: https://dev.twitter.com/rest/public/search.

Let's try a more complicated search:

```
In [48]: r = twitter.get(search, params={
                 'q' : 'happy',
                 'geocode' : '37.8734855,-122.2597169,10mi'
             })
         r.ok

Out[48]: True

In [49]: statuses = r.json()['statuses']
         statuses[0]

Out[49]: {'contributors': None,
          'coordinates': {'coordinates': [-122.41910988, 37.78288968], 'type': 'Point'},
          'created_at': 'Wed Feb 24 21:13:38 +0000 2016',
          'entities': {'hashtags': [{'indices': [63, 69], 'text': 'vacay'},
            {'indices': [70, 86], 'text': 'romanticgetaway'}],
          'symbols': [],
          'urls': [{'display_url': 'instagram.com/p/BCLwZp2P-2P/',
            'expanded_url': 'https://www.instagram.com/p/BCLwZp2P-2P/',
            'indices': [88, 111],
            'url': 'https://t.co/Jx7GVQ3CKa'}],
```

```
 'user_mentions': []},
'favorite_count': 0,
'favorited': False,
'geo': {'coordinates': [37.78288968, -122.41910988], 'type': 'Point'},
'id': 702602307062648833,
'id_str': '702602307062648833',
'in_reply_to_screen_name': None,
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'is_quote_status': False,
'lang': 'en',
'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
'place': {'attributes': {},
 'bounding_box': {'coordinates': [[[-122.514926, 37.708075],
    [-122.357031, 37.708075],
    [-122.357031, 37.833238],
    [-122.514926, 37.833238]]],
  'type': 'Polygon'},
 'contained_within': [],
 'country': 'United States',
 'country_code': 'US',
 'full_name': 'San Francisco, CA',
 'id': '5a110d312052166f',
 'name': 'San Francisco',
 'place_type': 'city',
 'url': 'https://api.twitter.com/1.1/geo/id/5a110d312052166f.json'},
'possibly_sensitive': False,
'retweet_count': 0,
'retweeted': False,
'source': '<a href="http://instagram.com" rel="nofollow">Instagram</a>',
'text': 'Happy birthday to the most wonderful man ever! I love you Ty!! #vacay #romanticgetawa
'truncated': False,
'user': {'contributors_enabled': False,
 'created_at': 'Thu Apr 10 19:53:05 +0000 2014',
 'default_profile': True,
 'default_profile_image': False,
 'description': 'Live the dream. I do every day.',
 'entities': {'description': {'urls': []}},
 'favourites_count': 35,
 'follow_request_sent': False,
 'followers_count': 168,
 'following': False,
 'friends_count': 115,
 'geo_enabled': True,
 'has_extended_profile': False,
 'id': 2437450614,
 'id_str': '2437450614',
 'is_translation_enabled': False,
 'is_translator': False,
 'lang': 'en',
 'listed_count': 7,
 'location': 'Los Angeles, CA',
```

```
'name': 'Allison Hawkstone',
'notifications': False,
'profile_background_color': 'C0DEED',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_tile': False,
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/2437450614/1450893408',
'profile_image_url': 'http://pbs.twimg.com/profile_images/679722341132320774/2uu0KXpm_normal.
'profile_image_url_https': 'https://pbs.twimg.com/profile_images/679722341132320774/2uu0KXpm_n
'profile_link_color': '0084B4',
'profile_sidebar_border_color': 'C0DEED',
'profile_sidebar_fill_color': 'DDEEF6',
'profile_text_color': '333333',
'profile_use_background_image': True,
'protected': False,
'screen_name': 'ActHawkstone',
'statuses_count': 130,
'time_zone': None,
'url': None,
'utc_offset': None,
'verified': False}}
```

If we want to store this data somewhere, we can output it as json using the json library from above. However, if you're doing a lot of these, you'll probaby want to use a database to handle everything.

```
In [50]: with open('my_tweets.json', 'w') as f:
             json.dump(statuses, f)
```

To post tweets, we need to use a different endpoint:

```
In [51]: post = "https://api.twitter.com/1.1/statuses/update.json"
```

And now we can pass a new tweet (remember, Twitter calls these 'statuses') as a parameter to our post request.

```
In [52]: r = twitter.post(post, params={
                 'status' : "I stole Juan's Twitter credentials"
             })
         r.ok

Out[52]: False
```

Other (optional) parameters include things like location, and replies.

4.5 Scheduling

The real beauty of bots is that they are designed to work without interaction or oversight. Imagine a situation where you want to automatically retweet everything coming out of the D-Lab's twitter account, "@DLabAtBerkeley". You could:

1. spend the rest of your life glued to D-Lab's twitter page and hitting refresh; or,
2. write a function

We're going to import a module called `time` that will pause our code, so that we don't hit Twitter's rate limit

```
In [53]: import time

         def retweet():
             r = twitter.get(search, {'q':'DLabAtBerkeley'})
             if r.ok:
                 statuses = r.json()['statuses']
                 for update in statuses:
                     username = item['user']['screen_name']
                     parameters = {'status':'HOORAY! @' + username}
                     r = twitter.post(post, parameters)
                     print(r.status_code, r.reason)
                     time.sleep(5)
```

But you are a human that needs to eat, sleep, and be social with other humans. Luckily, Linux systems have a time-based daemon called `cron` that will run scripts like this *for you*.

People on windows and macs will not be able to run this. That's okay.

The way that `cron` works is it reads in files where each line has a time followed by a job (these are called cronjobs). You can edit your crontab by typing `crontab -e` into a terminal.

They looks like this:

```
In [54]: with open('../etc/crontab_example', 'r') as f:
             print(f.read())
```

```
# In a user's crontab, jobs run under that user
# Time is specified as <min> <hour> <day> <month> <wday>
# To specify any time, use `*`
# For unknown reasons, cronjobs fail unless the tab ends with a newline

00 08 * * 1 echo "It is 8am on Monday" >> /var/dumblog
```

This is telling `cron` to print that statement to a file called "dumblog" at 8am every Monday.

It's generally frowned upon to enter jobs through crontabs because they are hard to modify without breaking them. The better solution is to put your timed command into a file and copy the file into `/etc/cron.d/`. These files look like this:

```
In [55]: with open('../etc/crond_example', 'r') as f:
             print(f.read())
```

```
#!/bin/bash
# First, make sure you specify all of the paths that you might need to run
# your task. If you aren't sure, copy the entire $PATH variable

PATH=/home/dillon/.conda/envs/py27/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

# Then, specify when you want the task to occur; the user account to run it;
# and the job

@hourly dillon cd ~/scripts; python simple.py
```

At this point, you might be a little upset that you can't do this on your laptop, but the truth is you don't really want to run daemons and cronjobs on your laptop, which goes to sleep and runs out of batteries. This is what servers are for (like AWS).

4.6 Now it is time for you to make your own twitter bot!

To get you started, we've put a template in the `scripts` folder. Try it out, but be generous with your `time.sleep()` calls as the whole class is sharing this account.

If you have tried to run this, or some of the earlier code in this notebook, you have probably encountered some of Twitter's error codes. Here are the most common, and why you are triggering them.

1. `400 = bad request` - This means the API (middleman) doesn't like how you formatted your request. Check the API documentation to make sure you are doing things correctly.

2. `401 = unauthorized` - This either means you entered your auth codes incorrectly, or those auth codes don't have permission to do what you're trying to do. It takes Twitter a while to assign posting rights to your auth tokens after you've given them your phone number. If you have just done this, wait five minutes, then try again.

3. `403 = forbidden` - Twitter won't let you post what you are trying to post, most likely because you are trying to post the same tweet twice in a row within a few minutes of each other. Try changing your status update. If that doesn't fix it, then you are either:

   A. Hitting Twitter's daily posting limit. They don't say what this is.

   B. Trying to follow too many people, rapidly following and unfollowing the same person, or are otherwise making Twitter think you are a spambot

4. `429 = too many requests` - This means that you have exceeded Twitter's rate limit for whatever it is you are trying to do. Increase your `time.sleep()` value.