

Python + Twitter: REST API for Place Based Location

Tonight's Learnings

- You guys just used Twitter platform as a GPS...for free (sort of).
- Learn how to install a Python module from source
- Learn how to do more arcpy geoprocessing
- Learn how to use embedded links in ArcMap & ArcGIS Online

Python to Consume Twitter “Search” REST API

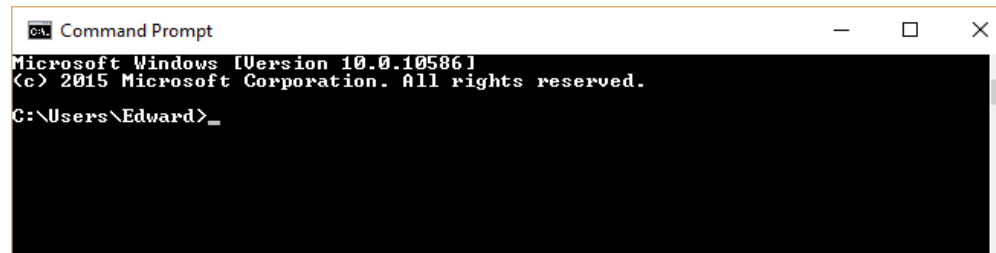
- Install the “tweepy” Python module
- Use tweepy to authenticate to Twitter and search using the API
- Process the results to extract geographic information
- Create a feature class to hold the results
- Use the feature class to map the results using ArcMap or AGO

Install “tweepy”

- Copy **O:\Courses\AppGIS\TwitterMapping** to your **H:\Documents** folder
- The tweepy 3.5.0 library was downloaded as a zip file and extracted from <https://github.com/tweepy/tweepy>
- Normally, we would use the “pip” library to install tweepy, but we can’t do that on the lab computers due to their configuration.
- Instead, we will “install from source” code.

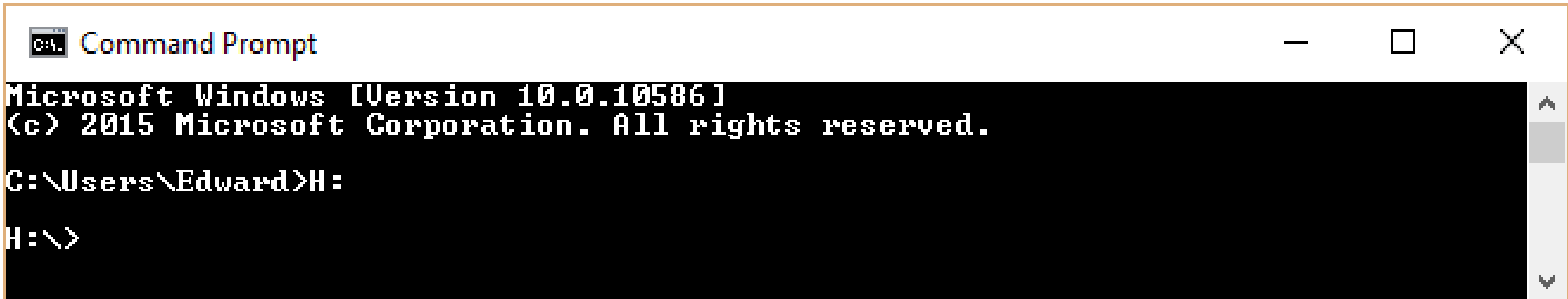
Install “tweepy”

- Open up a command shell. Go to start > search and type “cmd” and hit enter. The Windows command prompt should appear.



```
Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\Edward>
```

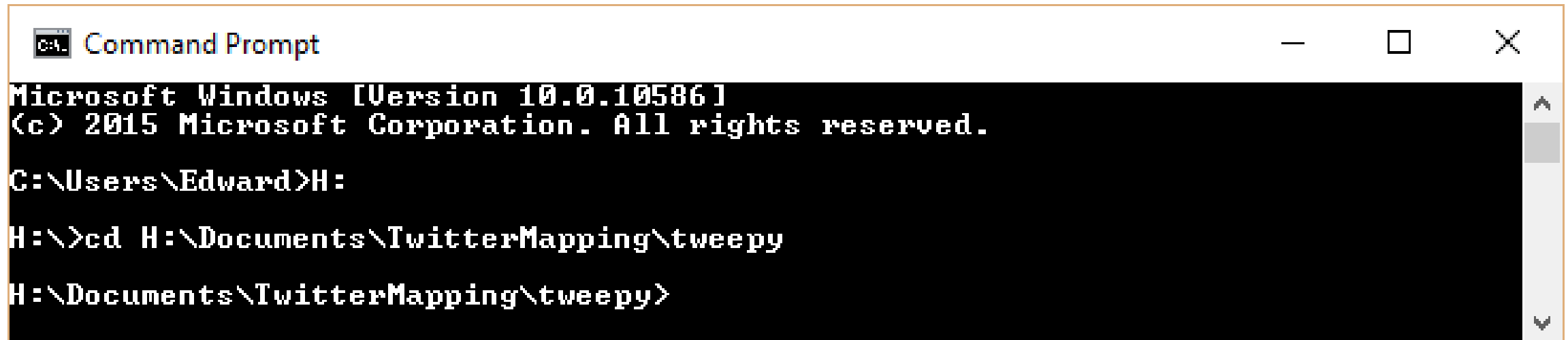
- Change your working directory using the “cd” command to the tweepy directory. But first just type “H:” and hit enter, in order to change drives.



```
Command Prompt
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Users\Edward>H:
H:\>
```

Install “tweepy”

- Now change your directory to **H:\Documents\TwitterMapping\tweepy**


A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt". The window content shows the following text: "Microsoft Windows [Version 10.0.10586] (c) 2015 Microsoft Corporation. All rights reserved. C:\Users\Edward>H: H:\>cd H:\Documents\TwitterMapping\tweepy H:\Documents\TwitterMapping\tweepy>". The text is white on a black background. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Edward>H:
H:\>cd H:\Documents\TwitterMapping\tweepy
H:\Documents\TwitterMapping\tweepy>
```

Install “tweepy”

- Open up Windows Explorer and navigate to the **H:\Documents\TwitterMapping\tweepy** folder. There is a Python script called “setup.py”. We will run this file using Python. It will install the tweepy library so that it can be imported into Python.
- In your command prompt, type and hit enter:
`c:\python27\arcgis10.3\python.exe setup.py install`



A screenshot of a Windows Command Prompt window. The title bar reads "C:\ Command Prompt". The command prompt shows the current directory as "H:\Documents\TwitterMapping\tweepy" and the command being executed is "c:\python27\arcgis10.3\python.exe setup.py install". The window has standard Windows window controls (minimize, maximize, close) in the top right corner and a scrollbar on the right side.

```
C:\ Command Prompt
H:\Documents\TwitterMapping\tweepy>c:\python27\arcgis10.3\python.exe setup.py install
```

Install “tweepy”

- This will install tweepy.
- You only need to do this once. To test it go over to PyScripter and type “import tweepy” into the Python command prompt.
- If you don’t get an error, it worked!

Python Interpreter

```
*** Python 2.7.8 (default, Jun 30 2014, 16:03:49) [MSC v.1500 32 bit (Intel)] on win32. ***  
>>> import tweepy  
>>> |
```


Now, Let's Develop Python Code

- We are going to use 3 files for this project:
 - TwitterMapping\Scripts\twitter_response.py
 - Inspect the Twitter response
 - TwitterMapping\Scripts\make_fc.py
 - Make geodatabase & feature class
 - TwitterMapping\Scripts\twitter_processor.py
 - Put it all together to generate a search, process the response, and create/update a feature class in a geodatabase.

Let's search Twitter for #appgis

- Open up **H:\Documents\TwitterMapping\Scripts\twitter_response.py**
- In this scripts we are going to:
 - Authenticate to Twitter using Oauth
 - Search Twitter for #appgis tweets (otherwise known as status)
 - Inspect the results for relevant geographic information

```
import json
import tweepy
from tweepy import OAuthHandler
```

```
# These are the credentials needed to make requests using the Twitter API. These
# are considered your API keys
# To obtain your own credentials, you need to register an app with Twitter at apps.twitter.com
```

```
consumer_key = 'Z1fB6taapkl02oxrldaGwvuyX'
consumer_secret = 'HBjIckV80PMPr1kDyQWzV2r9lZG8K7o1V3d4i041v7ixMCl3oE'
access_token = '3912791837-Ml0oWsfNOEplAbrv3S0kTmiRxsxTzXOpapGuPDG'
access_secret = 'wCLsJegdtznzt4DPO1yeNEV48h4pTS57bx3qgUDY86v1K'
```

```
# Use OAuth to set your access token.
```

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

```
# Create the tweepy api object. Use the access token as input.
```

```
api = tweepy.API(auth)
```

Create a search term to query the Twitter Search API

query = "%23appgis" # the %23 stands for #.

Ask twitter to give us the first tweet that uses the #appgis hashtag

tweets = api.search(q=query, count=1)

how many records where returned? Should be 1

print len(tweets)

inspect the tweets object to see what it is.

print type(tweets)

<class 'tweepy.models.SearchResults'> This is a special tweepy object called SearchResults. You can iterate through this object.

Using the json module will make the response more readable. Print the first item in the tweets SearchResult object.

print(json.dumps(tweets[0]._json, indent=2, sort_keys=True))

Look at the Response...It's a dictionary

- Open up the response file at
H:\Documents\TwitterMapping\Docs\twitter_response.txt
- There is all sorts of information
- We are interested in the geographic information and the tweet text
- The response is broken into key:value pairs. The keys we are interested in are:
 - **geo** – contains precise coordinates (i.e. Lat/Long)
 - **place** – contains a bounding box for a “place”.
 - **text** – the text of the tweet
 - **user** – user information

Make a feature class to hold the data

- Open up **H:\Documents\TwitterMapping\Scripts\make_fc.py**
- In this script we are going to:
 - Create a file geodatabase. If one exists already, delete it
 - Create a spatial reference
 - Create a feature class
 - Add fields to the feature class

Put It All Together

- Open up **H:\Documents\TwitterMapping\Scripts\twitter_processor.py**
- Now you can run multiple queries

Challenge #1: Make Better Code

- Instead of deleting and recreating the file geodatabase/feature class every time, use the Truncate Table tool to drop all records in the table
- Change the below lines of code in twitter_processor.py to use the Truncate Table tool. Here is a [link](#) to the tool reference.

```
if arcpy.Exists(fgdb):
```

```
    arcpy.Delete_management(fgdb)
```

```
arcpy.CreateFileGDB_management(arcpy.env.workspace, fgdb_name)
```

```
arcpy.CreateFeatureclass_management(fgdb, fc_name, geometry_type, "",  
"DISABLED", "DISABLED", spatial_reference)
```


```
for field in fields:
```

```
    arcpy.AddField_management(fc, field[0], field[1])
```


Create your own searches

- Copy the code in **twitter_processor.py** into a new file and alter the code to make new searches. Make sure to limit your **max_tweets** variable so you don't crash your computer with millions of tweets!

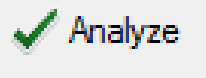
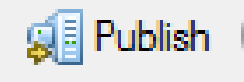
Inspect Your Data

- First bring your features into ArcMap.
- Notice that you can enable pop-ups with the  button. Click the button, then click the points in the map to view the pop-up. Now, you can click the link in the pop-up to view the tweet! You can also do this with the identify tool.

Challenge #2: Share as a Map Package

- After bringing your features into ArcMap, style your points and add any reference data you like.
- Now, click File > Share As > Map Package... This will bundle your map up along with your data to enable you to share your map & data.
- You can either upload to your AGO Org, or save locally. Sign into your AGO Org.
- Follow the [reference page](#) for how to upload your map package. Make sure to bring your source data along!
- Go to your AGO Org to verify success.

Challenge #3: Share as a Service

- In ArcMap, make sure the only layer in your map is “tweets”.
- Now, click File > Share As > Service... You must be logged into AGO to do this.
- This will enable you to share your “tweets” as a feature service in AGO
- Choose “Publish a Service” and click Next.
- Choose “My Hosted Services” and give your service a name. Click Next.
- In Capabilities, turn off “Tiled Mapping” and turn on “Feature Access”.
- In Description, enter the required fields.
- Take the default values for everything else.
- Click the  button. This will report if you have done anything incorrectly. If there are no errors, click the  button.
- Go to your AGO Org verify success.
- View the full documentation [here](#).

Challenge #3: Share as a Service

- In AGO, you can now create a new map. You could also use WebAppbuilder to create a customized web application without writing any code. <https://doc.arcgis.com/en/web-appbuilder/create-apps/make-first-app.htm>
- Here's mine. I made a map based on the hashtag #ThingsBernieHates. Apparently, this is sarcastic hashtag about Bernie Sanders (I think).
- <http://efarrell.maps.arcgis.com/apps/SimpleViewer/index.html?appid=100cee15ab124007b13e8d23a1ac54d2>
- Share your maps on the Blackboard Wiki Page