Webscraping Reddit—Python Reddit API Wrapper (PRAW) Tutorial for Windows

Eric Plog Follow
Jun 11 · 4 min read

In this brief tutorial, I will provide a high level overview of PRAW and provide examples utilized in my Data Science Immersive boot camp project. I learned to use PRAW from the following website:

http://www.storybench.org/how-to-scrape-reddit-with-python/

Requirements

- Python 3.x
- Interactive Development Environment (IDE)—i.e, Jupyter Notebooks OR a Text Editor—i.e., Atom
- PRAW
- Pandas
- A Reddit account

Installing PRAW

```
pip install praw
```

Updating Praw

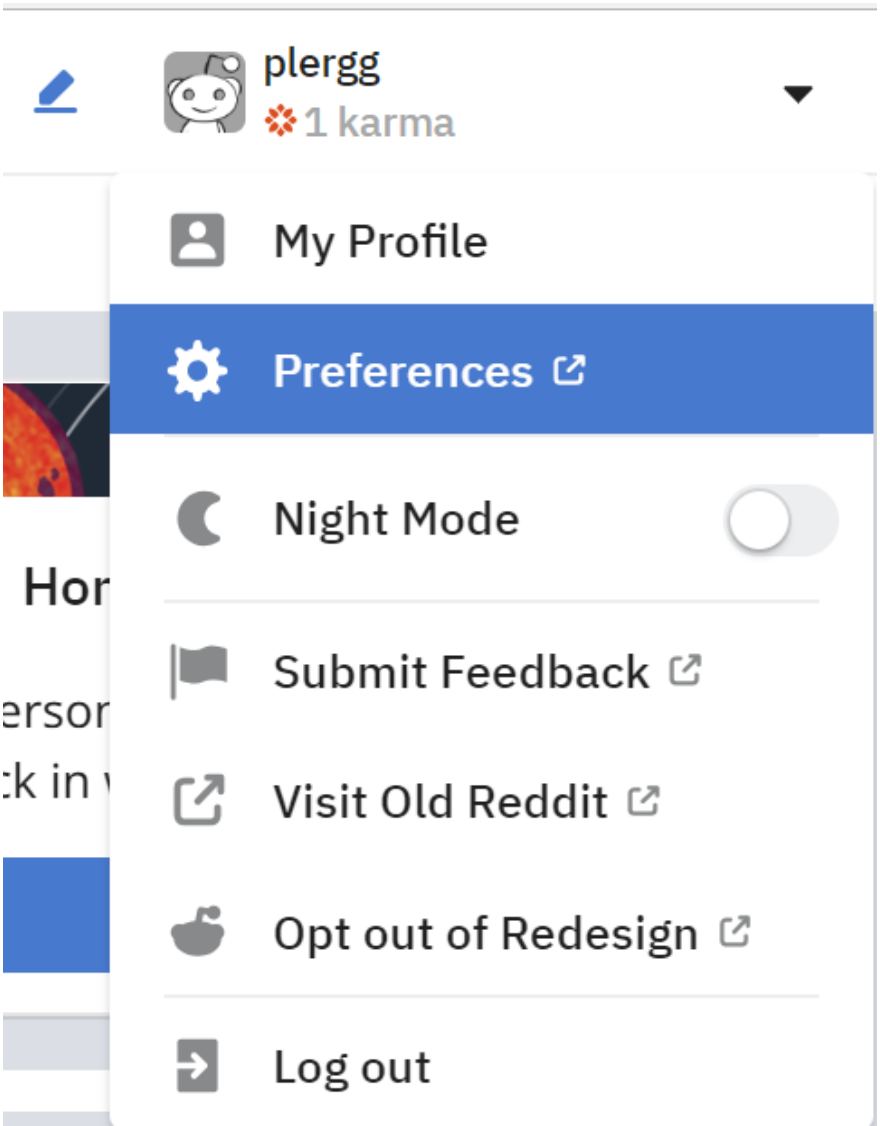
```
pip install --upgrade praw
```

Installing Pandas

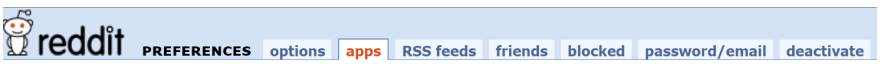
```
conda install pandas
```

Register your "application" on Reddit

1. Log into your Reddit account.
2. Click the down arrow to the right of your name, then click the *Preferences* button.

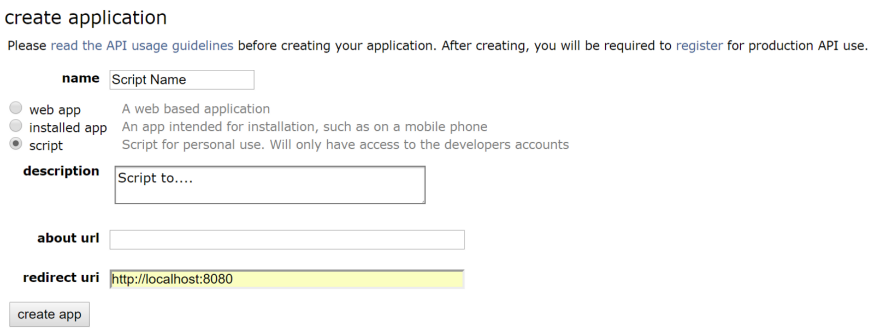


3. Click the *app* tab.

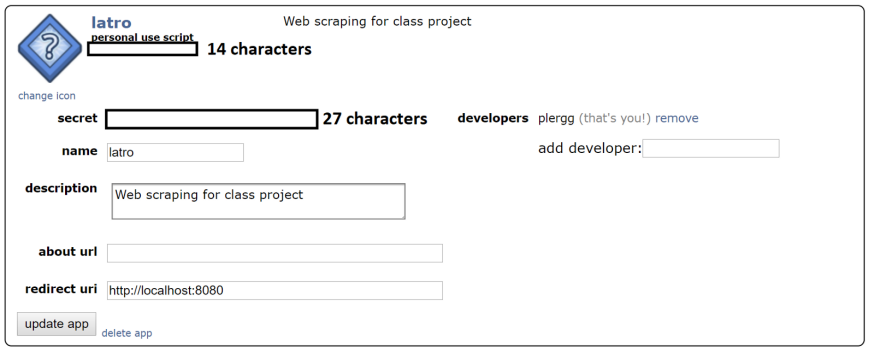


4. Click the *create another app* button at the bottom left of your screen.

5. Populate your script with the required fields. Refer to the screen shot below:



6. Hit the *create app* button once you have populated all fields. You should now have a script which resembles the following:



Utilizing PRAW within an Interactive Development Environment

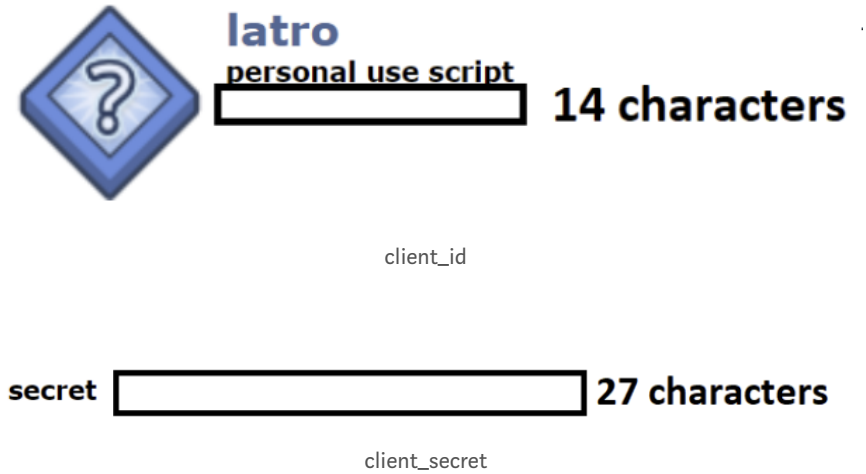
1. Import required packages

```
import praw
import pandas as pd
import datetime as dt #only if you want to analyze the
date created feature
```

2. Call the praw.Reddit function and assign it to a variable

```
# I assigned reddit as the variable name, you can call
it whatever you want to.

reddit = praw.Reddit(client_id='14_CHARS_IDENTIFIER',

client_secret='27_CHARS_SECRET_ID',
                    user_agent='SCRIPT_NAME',
                    username='REDDIT_USER_NAME',
                    password='REDDIT_LOGIN_PASSWORD')
```

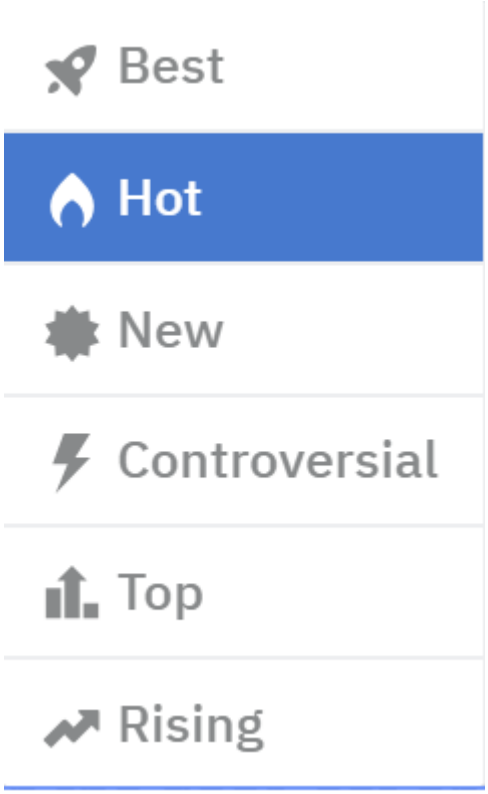
```
# IF YOU HAVE ANY SPACES BETWEEN THE CHARACTERS AND
THE QUOTES, YOU WILL RECEIVE AN ERROR.

# GOOD: '14_CHARS_IDENTIFIER'
# BAD: ' 14_CHARS_IDENTIFIER '
```



3. You can sort Reddit threads by the following:

- Best
- Hot
- New
- Controversial
- Top
- Rising

For the sake of my project, I utilized hot and top threads, but you can experiment with any of the aforementioned sorting methods.



4a. Pulling 1,000 (maximum pull limit) hot threads with no sub-reddit filtering:

```
no_subreddit = reddit.subreddit('all')
hot = no_subreddit.hot(limit=1000)
```

4b. Pulling 1,000 hot /r/python and /r/Rlanguage hot threads:

```
subreddit1 = reddit.subreddit('python')
python_subreddit = subreddit1.hot(limit=1000)
subreddit2 = reddit.subreddit('$language')
r_subreddit = subreddit2.hot(limit=1000)
```

5. Create an empty dictionary and iterate through chosen features to append to the dictionary:

```
dict =        { "title":[],
                "subreddit":[],
                "score":[],
                "id":[],
                "url":[],
                "comms_num": [],
                "created": [],
                "body":[]}
```

```
dict = { "title":[],
                "subreddit":[],
                "score":[],
                "id":[],
                "url":[],
                "comms_num": [],
                "created": [],
                "body":[]}
```

```
for submission in python_subreddit:
    dict["title"].append(submission.title)
    dict['subreddit'].append(submission.subreddit)
    dict["score"].append(submission.score)
    dict["id"].append(submission.id)
    dict["url"].append(submission.url)
    dict["comms_num"].append(submission.num_comments)
    dict["created"].append(submission.created)
    dict["body"].append(submission.selftext)
```

```
for submission in r_subreddit:
    dict["title"].append(submission.title)
    dict['subreddit'].append(submission.subreddit)
    dict["score"].append(submission.score)
    dict["id"].append(submission.id)
    dict["url"].append(submission.url)
    dict["comms_num"].append(submission.num_comments)
    dict["created"].append(submission.created)
    dict["body"].append(submission.selftext)
```

If you hit tab after *submission.* you can see which features you can pull from Reddit.

6. Convert the dictionary to a Pandas data frame, and export to csv

```
df = pd.DataFrame(dict)
df = pd.read_csv('./project 3 praw pythonR v1.csv')
```

If you want to pull more data, change the file name of the .csv, and combine all csv's into one master csv.

7. (Optional) If you opted to use the *created* feature, you must convert the date into a readable format

```
def get_date(created):
    return dt.datetime.fromtimestamp(created)
```

```
df["created"] = df['created'].apply(get_date)
```

8. Drop duplicate data based on the *id* unique identifier

```
df.drop_duplicates(subset=['id'], inplace=True)
```

9. One model I created was to categorize if a thread was posted in /r/Python or /r/Rlanguage. In order to dummy the column, I had to apply a lambda function to change the features' data type from Subreddit to an array:

```
In [8]: df['subreddit'].unique()
Out[8]: array([Subreddit(display_name='Python'),
               Subreddit(display_name='Rlanguage')], dtype=object)

In [9]: df['subreddit']= df['subreddit'].apply(lambda x: x.display_name)

In [10]: df['subreddit'].unique()
Out[10]: array(['Python', 'Rlanguage'], dtype=object)
```

10. Now we can use Pandas to dummy the column into a single column of 1's and 0's

```
df = pd.get_dummies(df, columns=['subreddit'],
drop_first=True)
```

11. Select your x and y variables

```
X = df['body'] # Try adding additional features!
y = df['subbreddit_space']
```

12. Train, test, split your x and y variables

```
X_train, X_test, y_train, y_test = train_test_split(X,
y, random_state=42)
```

13. Vectorize text data for analysis

```
tvec = TfidfVectorizer(stop_words='english',
max_features=1000, ngram_range=[1,2])
X_train_tvec = tvec.fit_transform(X_train)
X_test_tvec = tvec.transform(X_test)
```

14. Check for top words utilized

```
indices = np.argsort(tvec.idf_)[::]
features = tvec.get_feature_names()
top_n = 15
top_features = [features[i] for i in indices[:top_n]]
print(top_features)
```

15. Train your model

```
rf = RandomForestClassifier()
rf.fit(X_train_tvec,y_train)
```

16. Score your model

```
rf.score(X_test_tvec, y_test)
```