```python

# coding: utf-8

# ## Non-subtweets Downloader Jupyter Notebook-in-Progress

# In[ ]:


import tweepy
import json


# #### Set up access to the API

# In[ ]:


consumer_key, consumer_secret, access_token, access_token_secret = open("../../credentials_alternate.txt").read().split("\n")


# In[ ]:


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


# #### Specifically take advantage of built-in methods to handle Twitter API rate limits

# In[ ]:


api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True, compression=True)


# #### Find tweets with replies that do not claim it is a subtweet

# In[ ]:

def first_tweet(tweet_status_object):
    try:
        return first_tweet(api.get_status(tweet_status_object.in_reply_to_status_id_str, tweet_mode="extended"))
    except tweepy.TweepError:
        return tweet_status_object

def get_subtweets(max_tweets=10000000, query=("subtweet AND @ since:2018-03-01 exclude:retweets filter:replies")):
    subtweets_ids_list = []
    subtweets_list = []
    i = 0
    for potential_subtweet_reply in tweepy.Cursor(api.search, lang="en", tweet_mode="extended", q=query).items(max_tweets):
        i += 1
        # print("\tTweet #{0} may be a reply to a subtweet: {1}".format(i, potential_subtweet_reply.full_text.replace("\n", " ")))
        potential_subtweet_original = first_tweet(potential_subtweet_reply)
        if not potential_subtweet_original.in_reply_to_status_id_str and potential_subtweet_original.user.lang == "en":
            if potential_subtweet_original.id_str in subtweets_ids_list or "subtweet" in potential_subtweet_original.full_text or "Subtweet" in potential_subtweet_original.full_text or "SUBTWEET" in potential_subtweet_original.full_text:
                continue
            else:
                subtweets_ids_list.append(potential_subtweet_original.id_str)
                subtweets_list.append({"tweet_data": potential_subtweet_original._json, "reply": potential_subtweet_reply._json})
                with open("../data/other_data/subtweets.json", "w") as outfile:
                    json.dump(subtweets_list, outfile, indent=4)
                print("Tweet #{0} was a reply to a subtweet: {1}\n".format(i, potential_subtweet_original.full_text.replace("\n", " ")))
    return subtweets_list


# In[ ]:


subtweets_list = get_subtweets()
print("Total: {}".format(len(subtweets_list)))

```