

SocketNotebook

November 15, 2018

1 Sending tweets notebook

1.0.1 Soufiane MOUTEI - Ahmed BEN SAAD

```
In [ ]: from tweepy.streaming import StreamListener
        from tweepy import OAuthHandler, Stream
        import socket, json, time
```

Let's start by some initializations:

```
In [ ]: # Go to http://apps.twitter.com and create an app.
        # The consumer key and secret will be generated for you after
        consumer_key="xKdRp33xaDiojU4wMF26iZGtu"
        consumer_secret="9ezEX71a2mXsftBA6A8GVfuvoRhXdsy2DBYqwnVBKA1L0tEkm8"

        # After the step above, you will be redirected to your app's page.
        # Create an access token under the "Your access token" section
        access_token="493430618-Q4R134ZfZvL6ZsulWXfcdJAnV6Yzk5PWPQSwYpz8"
        access_token_secret="qYgOMoOB1Z0gJjIsCfuTEGI5SWLXjv21w4KIFdON0e3tS"

        # Subject of the tags
        subject = "france"
```

Now we'll create a class that serves as tweets listener, it will get them and send them via the socket

```
In [ ]: class TweetsListener(StreamListener):
        def __init__(self, csocket, time_limit=180):
            self.start_time = time.time()
            self.limit = time_limit
            self.client_socket = csocket

        def on_data(self, data):
            if (time.time() - self.start_time) >= self.limit:
                return False

            try:
                print(json.loads(data)["created_at"], ":", json.loads(data)["text"])
```

```

        self.client_socket.send(data.encode())
    except BaseException as e:
        print("Error on_data: %s" % str(e))
    return True

    def on_error(self, status):
        print(status)
        return True

```

We'll create a socket object and start then to send tweets to port 7000.

```

In [ ]: s = socket.socket()           # Create a socket object
        host = "localhost"           # Get local machine name
        port = 7000                   # Reserve a port for your service.
        s.bind((host, port))         # Bind to the port

        print("Listening on port: %s" % str(port))

        s.listen(5)                   # Now wait for client connection.
        c_socket, addr = s.accept()   # Establish connection with client.

        print("Received request from:", addr)

        auth = OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_token, access_token_secret) # Set access token

        twitter_stream = Stream(auth, TweetsListener(c_socket)) # Start streaming
        twitter_stream.filter(languages=["en", "fr"], track=[subject]) # Filter by subject and l

```

Finally, we close the socket once we finish.

```

In [ ]: s.close()

```