

04_streaming

September 19, 2016

1 Streaming APIs

```
In [ ]: import tweepy
import json

consumer_key = ""
consumer_secret = ""
access_token = ""
access_token_secret = ""

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

API = tweepy.API(auth)

In [ ]:
```

1.1 Utilizar la Streaming API de Twitter tiene 3 etapas:

- 1) Crear una clase hija de la clase `tweepy.StreamListener`.
- 2) Utilizar esa clase para crear un objeto de tipo `Stream`.
- 3) Conectar a la API de Twitter utilizando el `Stream`. Step 1: Creating a `StreamListener`

Devuelve una colección de tweets relevantes que coincidan con una consulta especificada.
`API.search(q[, lang][, locale][, rpp][, page][, since_id][, geocode][, show_user])`

1.1.1 1) Crear una clase hija de la clase `tweepy.StreamListener`.

```
In [ ]: class MiStreamListener(tweepy.StreamListener):

    def on_status(self, status):
        print (status.text)
        # substituir por lo que se debe hacer al recibir un tweet

    def on_error(self, status_code):
        if status_code == 420: # hemos llegado al limite
            print ('desconecto')
            #devolver False en on_data desconecta el stream
            return False
```

1.1.2 2) Utilizar esa clase para crear un objeto de tipo Stream.

```
In [ ]: miListener = MiStreamListener()
        miStream = tweepy.Stream(auth = API.auth, listener=miListener)
```

1.1.3 3) Conectar a la API de Twitter utilizando el Stream.

```
In [ ]: query = ["big data", "felicidad"]
        miStream.filter(track=query)
        #miStream.filter(track=query, async=True) # Tweepy offers a convenient asy

In [ ]: # desconectamos el listener
        miStream.disconnect()
```

1.2 Ejercicio:

Obtener los trending topics a nivel de Valencia y montar un Stream para recuperar los tweets con alguno de los 3 primeros trending topics.

2 Minando Twitter

2.0.1 Lluís F. Hurtado (lhurtado at dsic.upv.es)

```
In [ ]:
```