

Autenticando o app

Durante os nossos exemplos vamos utilizar a biblioteca Tweepy, com vários recursos prontos para conectarmos o Python a um app do Twitter em poucos passos.

<https://www.tweepy.org/>(<https://www.tweepy.org/>)

In [1]:

```
import tweepy as tw
```

Agora preencha as informações abaixo de acordo com as chaves de acesso do seu app.

In [4]:

```
consumer_key= 'key'  
consumer_secret= 'secret'  
access_token= 'token'  
access_token_secret= 'token_secret'
```

Para autenticar, vamos utilizar o método OAuth de autenticação passando os parâmetros que definimos acima.

In [4]:

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

Agora podemos instanciar uma conexão com a API, usando a autenticação criada acima.

In [5]:

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

Postando um tweet

In [6]:

```
status = api.update_status("API test! #twitterapi")
```

O tweet postado gera um objeto com várias informações sobre a postagem

In [10]:

```
status._json
```

Out[10]:

```
{'created_at': 'Sun May 05 14:14:11 +0000 2019',
 'id': 1125041002849406977,
 'id_str': '1125041002849406977',
 'text': 'API test! #twitterapi',
 'truncated': False,
 'entities': {'hashtags': [{'text': 'twitterapi', 'indices': [10, 21]}],
 'symbols': [],
 'user_mentions': [],
 'urls': []},
 'source': '<a href="https://github.com/jcalvesoliveira/" rel="nofollow">a
ula-rest-ppt</a>',
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'in_reply_to_screen_name': None,
 'user': {'id': 917548352446791685,
 'id_str': '917548352446791685',
 'name': 'Julio C. Oliveira',
 'screen_name': 'jcalvesol',
 'location': '',
 'description': '',
 'url': 'https://t.co/8FDlrf16JV',
 'entities': {'url': {'urls': [{'url': 'https://t.co/8FDlrf16JV',
 'expanded_url': 'https://github.com/jcalvesoliveira/',
 'display_url': 'github.com/jcalvesoliveir...',
 'indices': [0, 23]}]}},
 'description': {'urls': []}},
 'protected': False,
 'followers_count': 10,
 'friends_count': 151,
 'listed_count': 0,
 'created_at': 'Tue Oct 10 00:32:29 +0000 2017',
 'favourites_count': 581,
 'utc_offset': None,
 'time_zone': None,
 'geo_enabled': True,
 'verified': False,
 'statuses_count': 7,
 'lang': 'en',
 'contributors_enabled': False,
 'is_translator': False,
 'is_translation_enabled': False,
 'profile_background_color': 'F5F8FA',
 'profile_background_image_url': None,
 'profile_background_image_url_https': None,
 'profile_background_tile': False,
 'profile_image_url': 'http://pbs.twimg.com/profile_images/95399363753638
7072/yXsa1yCb_normal.jpg',
 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/9539936
37536387072/yXsa1yCb_normal.jpg',
 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/91754835244
6791685/1516284817',
 'profile_link_color': '1DA1F2',
 'profile_sidebar_border_color': 'C0DEED',
 'profile_sidebar_fill_color': 'DDEEF6',
 'profile_text_color': '333333',
 'profile_use_background_image': True,
 'has_extended_profile': True,
```

```
'default_profile': True,
'default_profile_image': False,
'following': False,
'follow_request_sent': False,
'notifications': False,
'translator_type': 'none'},
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'is_quote_status': False,
'retweet_count': 0,
'favorite_count': 0,
'favorited': False,
'retweeted': False,
'lang': 'in'}
```

In [19]:

```
status.text
```

Out[19]:

```
'API test! #twitterapi'
```

Procurando por tweets

Podemos utilizar a API do Twitter também para procurar tweets por palavras-chave

In [28]:

```
search_words = "#python"
date_since = "2019-05-02"
```

In [29]:

```
tweets = tw.Cursor(api.search,
                    q=search_words,
                    lang="en",
                    since=date_since).items(10)
```

In []:

```
for tweet in tweets:
    print(tweet.text)
```

Podemos ver várias informações sobre as pessoas que postaram esse twitter, como nome e localização.

In []:

```
tweets = tw.Cursor(api.search,
                    q=search_words,
                    lang="en",
                    since=date_since).items(20)

users_locs = [[tweet.user.screen_name, tweet.user.location] for tweet in tweets]
```

In [66]:

```
users_locs
```

Out[66]:

```
[]
```

NLP - Natural Language Processing

Agora vamos utilizar a biblioteca TextBlob para gerar algumas informações sobre os tweets

In [23]:

```
from textblob import TextBlob
```

In [24]:

```
frase = TextBlob("Coding in Python is great for learning Machine Learning skills")
```

Primeiro podemos detectar o idioma de um texto.

In [25]:

```
frase.detect_language()
```

Out[25]:

```
'en'
```

Uma vez que a biblioteca consegue identificar o idioma, também é possível fazer a tradução do texto.

In [27]:

```
frase.translate(to='pt')
```

Out[27]:

```
TextBlob("Codificar em Python é ótimo para aprender habilidades de Aprendi  
zado de Máquina")
```

Agora pode utilizar um modelo de machine learning pré-treinado para identificar o sentimento e subjetividade da frase.

In [26]:

```
frase.sentiment
```

Out[26]:

```
Sentiment(polarity=0.8, subjectivity=0.75)
```

Streaming API

Além da API de pesquisa, o Twitter oferece uma API de streaming, que possibilita capturar todos os tweets sobre um determinado assunto, a partir do momento que iniciamos a streaming.

Primeiramente, vamos definir uma classe para inserirmos uma lógica toda vez que um tweet for identificado.

In [42]:

```
class MyStreamListener(tw.StreamListener):

    def on_status(self, status):
        print(status.text)
        print('-----')
        print('Idioma:')
        try:
            print(TextBlob(status.text).detect_language())
        except:
            print('Não foi possível detectar o idioma!')
        print('-----')
        print('Tradução:')
        try:
            print(TextBlob(status.text).translate(to='pt'))
        except:
            print('Não foi possível traduzir o tweet!')
        print('-----')
        print(TextBlob(status.text).sentiment)
        print('\n')
```

Vamos autorizar nossa stream, usando a mesma autorização que criamos anteriormente.

In [43]:

```
myStream = tw.Stream(auth = auth, listener=MyStreamListener())
```

Agora podemos iniciar o streaming.

In []:

```
myStream.filter(track=['python'])
```