

SMA_Lab-4_Mahalakshmi18

September 1, 2021

0.1 205229188

0.2 MahalakshmiS

0.3 Lab 4. Retriving the page's feed and measuring engagement using Facebook Graph API

0.3.1 Log in to your Facebook account and go to <https://developers.facebook.com/tools/explorer/> to obtain the ACCESS_TOKEN and access the graph API

```
[11]: import facebook
import requests
ACCESS_TOKEN="EAADOETpiG6sBA0pv60ZBWVZCJZAdOA3YxzlwP6JP7hw1bQgn2LqjWrugRoA7ZAftCuDKjzSrvmqNqga"
graph = facebook.GraphAPI(access_token= ACCESS_TOKEN, version="2.12")
```

0.3.2 Create a facebook page and find the fans associated with it

```
[12]: page_id= "110014798077031"
graph.get_object(id=page_id, fields=['fan_count'])['fan_count']
```

[12]: 0

0.3.3 Retriving the page's feed

```
[14]: def retrieve_page_feed(page_id, n_posts):
    """Retrieve the first n_posts from a page's feed in reverse
    chronological order."""
    feed = graph.get_connections(page_id, 'posts')
    posts = []
    posts.extend(feed['data'])
    while len(posts) < n_posts:
        try:
            feed = requests.get(feed['paging']['next']).json()
            posts.extend(feed['data'])
        except KeyError:
            # When there are no more posts in the feed, break
            print('Reached end of feed.')
            break
```

```

    if len(posts) > n_posts:
        posts = posts[:n_posts]

    print('{} items retrieved from feed'.format(len(posts)))
    return posts

def get_post_message(post):
    try:
        message = post['story']
    except KeyError:
        # Post may have 'message' instead of 'story'
        pass
    try:
        message = post['message']
    except KeyError:
        # Post has neither
        message = ''
    return message.replace('\n', ' ')

id="110014798077031"
for artist in [id]:
    print()
    feed = retrieve_page_feed(artist, 5)
    for i, post in enumerate(feed):
        message = get_post_message(post)[:50]
        print('{0} - {1}...'.format(i+1, message))

```

Reached end of feed.
0 items retrieved from feed

0.3.4 Measure the response to a post in terms of likes, shares, and comments

```

[18]: def measure_response(post_id):
    """Returns the number of likes, shares, and comments on a
    given post as a measure of user engagement."""
    likes = graph.get_object(id=post_id,
                             fields=['likes.limit(0).summary(true)'])\
                ['likes']['summary']['total_count']

    return likes
post_id="me"
likes = measure_response(post_id)
print(likes)

```

137