Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 10, October 2015, pg.331 – 333

RESEARCH ARTICLE

Use of Social Media API and Data Extraction for Financial Service on Twitter

Mr. Pankaj Uchil
Department of Computer Engineering, Rajiv Gandhi Institute of Technology, India
pankaj.uchil@live.in

Abstract— This document aims to explain a way to make use of the publicly available social media API to provide a means to build an automated service on Twitter that can enhance audience interaction with a particular brand on Social Media. For the purpose of building this service, we will make use of Python in order to scrape the relevant data from a website and then supply it using Twitter libraries in Python in the form of a Direct Message to the user.

Keywords— Social Media, API, Python, Data retrieval, Interactivity, Automation

I. Introduction

The prevalence of Social Media all around us has resulted in people being involved for longer periods in the use of Social Media platforms be it Facebook, Twitter et cetera. In this age of Digital Media, we have come across an opportunity to be innovative and create new mediums for users to interact with a particular brand and thus increase traction for brands in the Digital Media space. This service provides a way to perfectly align technology with interactivity that provides the users Net Asset Values in their Direct Message to a user who has tweeted a particular pre-specified shortcode. This will generate interest in the followers of the account because of the sheer novelty of this service. The building of this service involves the use of Python programming language for Data retrieval though the use of a web scraper using the BeautifulSoup library of Python and then extract the necessary values from the JSON values which are returned after we make use of urllib2 library and the linking of the Twitter API with the Tweepy library in Python to do carry out extract meta-data from the given tweet, check if the user is following the account and check which particular fund the user is interested in and then serves the relevant values in the form of a Direct Message to the user to maintain privacy of this data. In the process of building such an automated service, we have to take care not to exceed the rate limits which are laid out under the Twitter API guidelines. The cron job running on the server should be configured in such a manner that it is running at time intervals that adhere to time based rate limitations of this API. If we ignore this aspect of this service, we run the risk of getting errors in our code and in the worst case scenario, a complete suspension of the Twitter account. Hence, the coding has to be done in such a way that only the optimal number of calls are made to the Twitter API after each time interval.

II. WORKFLOW OF THE SERVICE

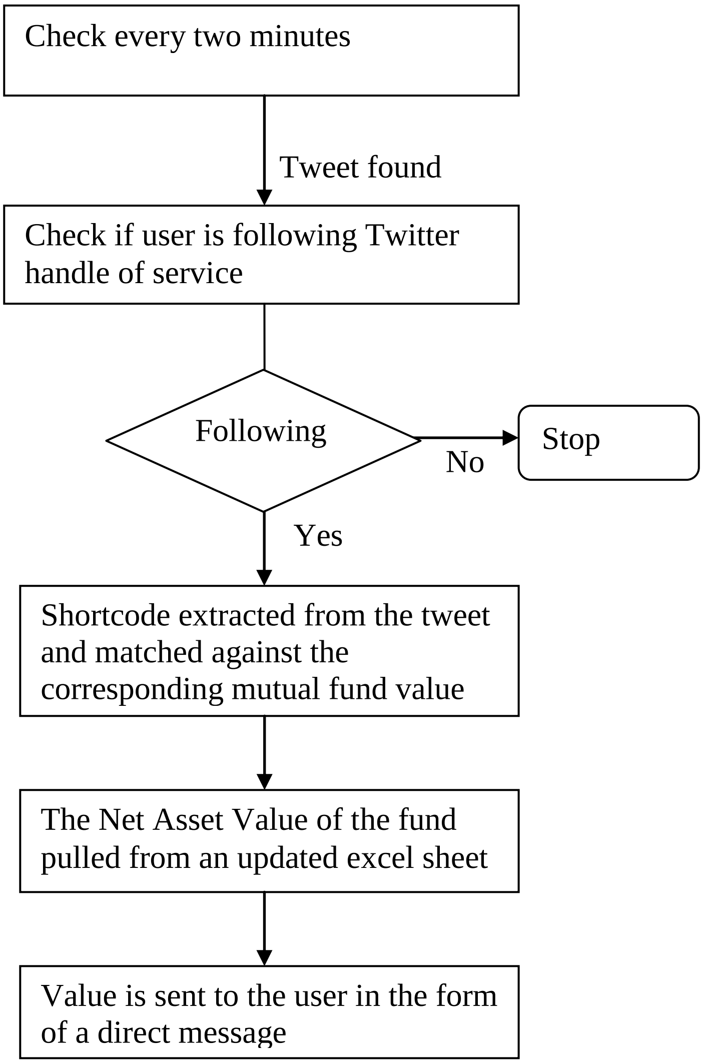
The way the service on Twitter will work is listed from the perspective of both the client-side process and the server-side process.

A. *Client-Side*

- The user will follow the specified Twitter handle
- On following, he will get information about the various shortcodes with the help of a creative in the form of a pinned Tweet.
- The user will have to mention the handle and tweet the shortcode corresponding to the mutual fund value he or she wants.

B. *Server-Side*

- The server will check every two minutes for a new tweet mention.
- In the event that a tweet is retrieved, the Twitter handle is filtered to check if the user is following the handle on which the service is running.
- If the user is following, the shortcode in the tweet is extracted and matched with the values pulled from a regularly updated excel sheet on the server.
- The Net Asset Value is then sent across to the user in the form of a Direct Message alongwith a timestamp to make each Direct Message unique.


Check every two minutes
Tweet found
Check if user is following Twitter handle of service
Following
No
Stop
Yes
Shortcode extracted from the tweet and matched against the corresponding mutual fund value
The Net Asset Value of the fund pulled from an updated excel sheet
Value is sent to the user in the form of a direct message

FLOW CHART OF SERVER-SIDE PROCESS

III. TECHNIQUES USED AND TECHNICAL SPECIFICATIONS

The technology used to develop this service and the technical specifications of the Twitter API are as follows:

1. Python Libraries used

1.1. Tweepy

Tweepy is an open-source library that is built with the purpose of using the Twitter API with Python to help authentication of a Twitter handle using OAuth and gathering the information relating to tweets of the tweets using the native API.

1.2. BeautifulSoup

Beautiful Soup provides a few simple methods and Pythonic idioms for navigating, searching, and modifying a parse tree: a toolkit for dissecting a document and extracting what you need. Beautiful Soup sits on top of popular Python parsers like lxml and html5lib, allowing you to try out different parsing strategies or trade speed for flexibility. This library is used to write a web scraper to pull all relevant data from a particular website.

1.3. Xlswriter

Xlsxwriter is a Python module that can be used to write text, numbers, formulas and hyperlinks to multiple worksheets in an Excel 2007+ XLSX file. It supports features such as formatting, merged cells, integration with Pandas et cetera. This library is used to populate an excel sheet with the values that have been pulled by the web scraper.

2. Twitter API specifications

2.1. Time-based limitations

Rate limits in version 1.1 of the API are divided into 15 minute intervals, which is a change from the 60 minute blocks in version 1.0. The GET requests limitations for the API is set at 15 requests per 15 minute window.

2.2. Volume-based limitations

The current limitation of the amount of Direct Messages that can be sent from a particular Twitter account, according to the official Twitter documentation is set at 1000 messages per day.

IV. FUTURE SCOPE

This particular service is built as a facilitator to provide information from the financial domain. In the same way, this service can be used in a variety of domains such as healthcare, entertainment et cetera by just altering the way the particular data is accessed after the extraction process.

V. CONCLUSION

The aim of this paper is to show a method through which we can build an automated service which sends the value of a mutual fund in the form of a Direct Message to the user, using the Twitter API and the Python programming language in response to a mention on the configured Twitter account.

REFERENCES

[1] Matthew A. Russell, “Mining the Social Web, 2nd Edition”.

[2] Ryan Mitchell, “Web Scraping with Python: Collecting Data from the Modern Web”.

[3] Official Twitter documentation: https://dev.twitter.com/overview/documentation.

[4] Skog, D. (2005). Social interaction in virtual communities: The significance of technology. International Journal of Web Based Communities, 1 (4), 464-474.

[5] Antonio Tapiador, Víctor Sánchez, Joaquín Salvachúa,” An Analysis of Social Network Connect Services”

[6] Benslimane, D., Dustdar, S., and Sheth, A. (2008). Services mashups: The new generation of web applications. Internet Computing, IEEE, 12(5):13 –15.

[7] Euro Python:https://ep2013.europython.eu/media/conference/slides/social-network-analysis-in-python.pdf.