Project 1: Facebook Graph API for Java

Ronak Shah
La Martiniere for Boys, Kolkata

October 10, 2012

1 **Facebook API for Java**

When one thinks of a social networking website, one tends to think of the iconic blue and white logo, representing an interlinked global community of over 1 billion strong, facebook.com. With its humble beginnings as an algorithm on a foggy glass plate, facebook.com is now a billion dollar enterprise that is growing exponentially with every passing minute. With greater number of users comes greater number of experiments and innovations. Facebook is now not only a social networking website providing users with top class messaging and communication techniques, but it is also a host of numerous applications with a variety of functions and purposes. The rapid growth of these apps can be attributed to the facebook Graph API, open to all developers interested in developing and publicizing apps on facebook for leisure, business, surveys, etc. The facebook developers' page, developers.facebook.com, gives users access to the Graph API explorer, a tool useful in the viewing and development of the various apps that we see online. Using this Graph API for Java, I designed a code that displays the inner workings of the apps that we see on facebook. The piece of code that I have written executes various GET requests to facebook.com, stores and operates upon the data received, and, displays the output as per the pre-decided function of the app.

2 Implementation: restfb.com

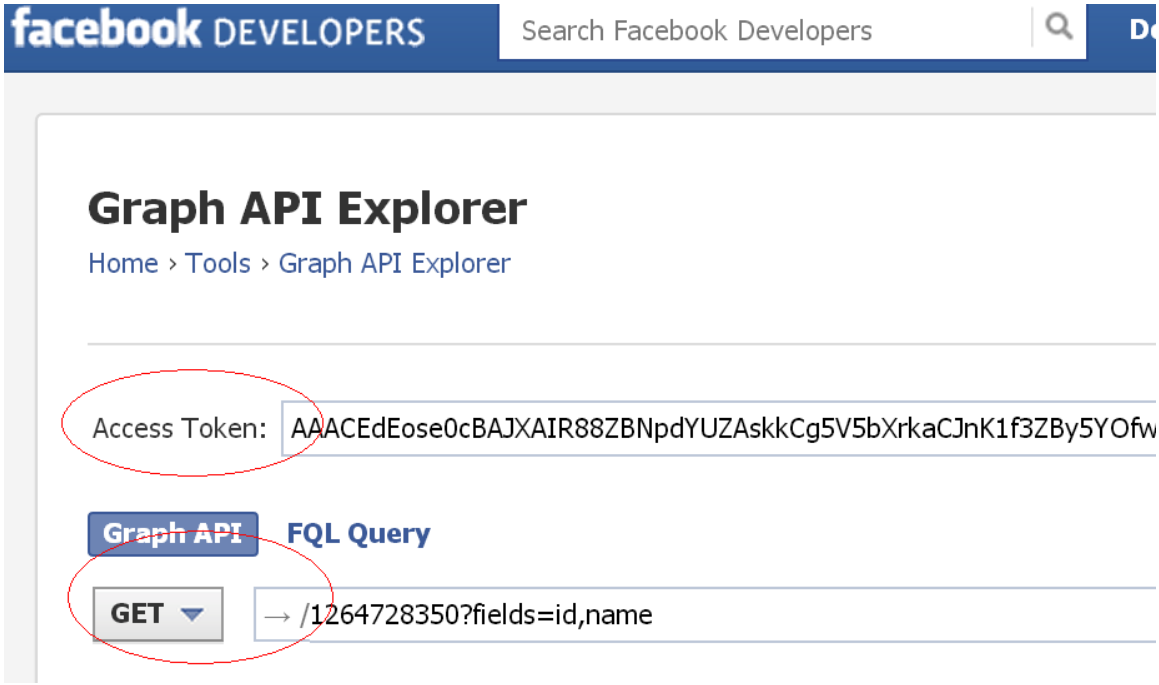
Implementing the code requires the prior installation of the facebook library on the standard Java IDE that is to be used. For this, a zip document is available on restfb.com as a restfb zip file. Restfb.com is a website that provides various sample codes and an API of its own that allows users to access the Graph API explorer on facebook.com. Restfb provides a separate Java API of its own mentioned along side the various other introductory notes on its home page. The restfb API is extremely flexible and can be loaded on to not only Java IDEs, but also, python, GitBash, etc. I chose Java due to its popularity among amateur coders like myself. After downloading and installing the restfb library on to a Java IDE, the next step would be to import all the requisite packages and start coding!

3 Project Aim: Popularity App

The code that we will design will have to have a purpose, it cannot be an application that simply obtains a user's data and does nothing with it. So, we devise a purpose of the app. Since we can make use of the GET requests and functions to obtain data about a user's status update and its related information, we shall make use of these requests to obtain a user's status updates, the lists of people who have "liked" each of these statuses, and the updated time of each of these updates. We shall then process this information and try to obtain the lists of people who follow the user the most and the most frequently used words in the status updates. For this, first, we need to download all the status updates on to our computer systems. We need to tackle numerous hurdles before we can actually operate upon the data that we download from facebook.com; these will be discusses in the forthcoming sections.

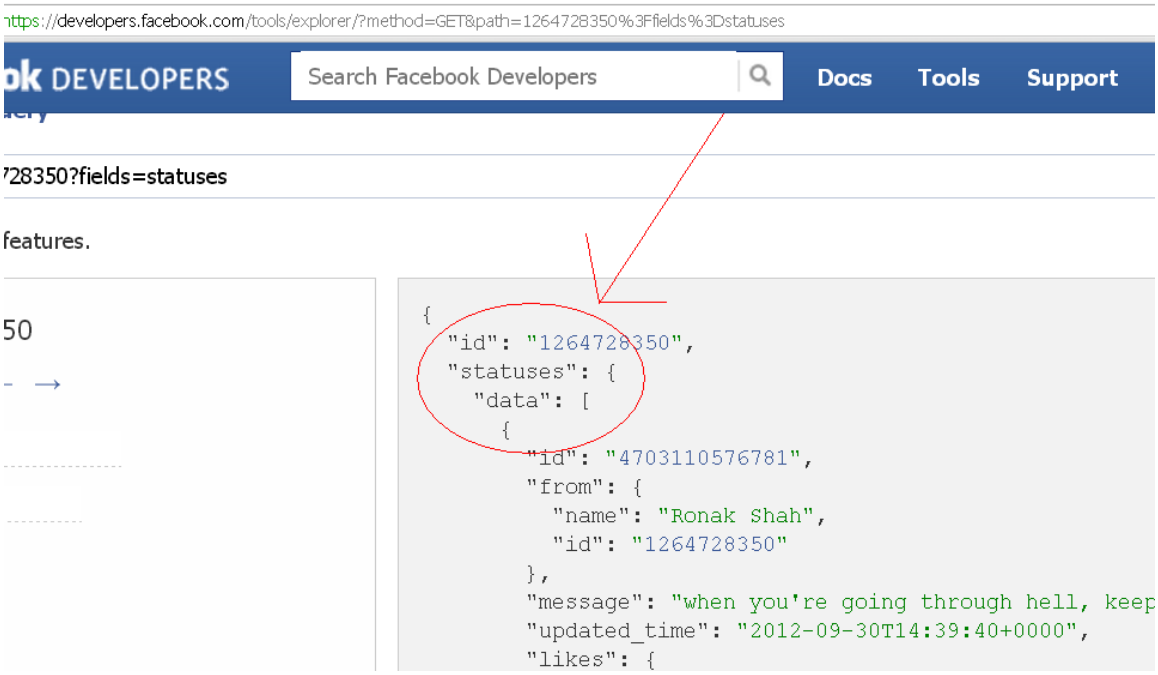
4 Hurdle One: Access Token

When one visits the developers.facebook page, one can use the Graph API explorer provided by facebook itself. To allow the explorer to obtain access to the user's data and information, the user needs to allow the explorer access to the details he wishes to see on the explorer screen. This permission(s) comes in the form of an access token. Access Tokens are permissions that each app needs to have before it can access the user's data and information. Access token are of two types: one type for applications and one type for users. For our code, we will make use of user access tokens because we are accessing the information and data on only a given user's profile. This access token can be obtained from the Graph API explorer itself. Access tokens are provided for various types of requests as well, for instance, for accessing status messages, accessing user birthday, relationship status, friend list, posting status messages, deleting messages, etc. Broadly, they types can be classified under: GET, POST, and DELETE. We will concern ourselves with only GET requests as they pertain to our purpose of the code. Access tokens thus play a crucial role in the development of our code, for it gives us an insight of the level of security that facebook provides its user with, ensuring that the user is responsible for all the damage caused to him/her by codes of malicious intent and likewise.



5 Hurdle two: Fetching objects and Parsing

After we obtain the access token, we need to create an instance of the facebookClient class from the restfb library so as to obtain access to a specified user's account. We utilize the access token to notify facebook.com that the given user has granted our code access to obtain the required information, in the form of GET requests, from his/her profile. This notification is crucial, because without this notification, the GET requests will be declined and the code will report a run time error. After this is done, we need to fetch objects from the user's profile. These objects contain the information that our code needs: the user's status updates and its related data and information. We shall download this data in the form of Json Objects, because not only does it reduce the amount of code to be written thereby reducing the run time in terms of constant time, but also, it makes the parsing of the data obtained to String form easier. Using the required functions provided by the API on restfb.com, we can fetch the required objects and can easily parse them to obtain the data that we need. Again, from the Graph API explorer, we can see the format in which the data has been stored on facebook.com, and we need to download the Json Objects in the similar manner. The image shows that the status updates are stored in a Json Object named "status" so we need to write some code that obtains the Json Array "data" within the object "status".



This is achieved by the following code:

FacebookClient facebookClient = new DefaultFacebookClient(Access_token);

JsonObject status = facebookClient.fetchObject("user/statuses", JsonObject.class);

So, we now have a Json Object comprising all the status updates that the user has made, and we also have related information about each of the statuses.

6 Hurdle three: Obtaining likes

If we take a look at the Graph API explorer, we can see that there is a Json Object called "likes" under each status update. This comprises a Json Array called "data" that contains the names and facebook IDs of the people (user's friends) who have "liked" the user's status update. We need this piece of information as well, so we need to find a way to download this piece of information too.

To implement this, we use the following code:

```
for(i=0;i<status.getJsonArray("data").length();i++,arraycounter++)
    {
      JsonObject datadownload=status.getJsonArray("data").getJsonObject(i);
//Loop body
}
```

Using this code, we can download the object at any index along the Json Array, and each of these objects will contain the name and facebook ID of a user's friend who has liked the user's status.
To store these likes, we make use of a jagged array of java.lang.String type. Using jagged arrays becomes convenient because each row will have a different length due to the different number of friends who like a particular status update. The number of rows is determined by the number of status updates that we download. Each status update has its unique set of users who like the status, so each row has its unique length and unique list of names to store. Therefore, each row corresponds to a unique status update. Once we store it in the jagged array, we proceed to the operations we need to perform on the data that we now have in hand.

7 Problem one: Finding the follower

Once we have with us on our system the names of the users who liked our user's statuses, we need to find the friend who follows our user the most. For this, we perform a simple frequency search on the jagged array of names that we have, i.e., we find the String that occurs the in the most number of rows. This logic follows from the idea that no friend can

like a status twice. Since each row corresponds to a unique status update, the friend whose name occurs in the most rows will be the religious follower we are looking for. A simple string frequency search will do the trick in our case and give us the friend who follows our user the most.

8 Problem two: Finding the word

To solve this problem, we need to download each status message and form a jagged array containing the words of these messages. From the Graph API explorer, we can see that each status update is stored under the Json Array "data" under the Json Object "statuses". To obtain these messages, we create an Array List of java.lang.String format. We call this array list "names" and store each status message in it. To obtain the status message, we simply reference the datadownload object that we have created earlier and add the status message to the array list.

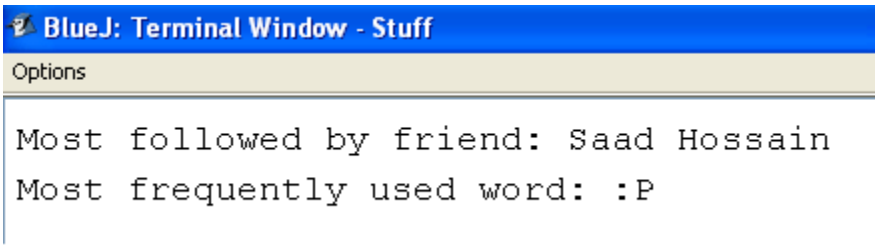
To perform this action, we use the following piece of code:

JsonObject datadownload=status.getJsonArray("data").getJsonObject(index);
message.add(datadownload.getString("message"));

So, now that we have an array list of all the status messages, we put each word of the status update into a jagged array. Each row of the jagged array corresponds to a particular status update, and, again, utilizing the String frequency search algorithm, we can find the word that is used most frequently in each status update.

9 Results

On completing the code, writing out the numerous functions and algorithms, we obtain the following results about user: ronak.shah.775



According to the code, Ronak Shah's friend Saad Hossain follows him the most because he has liked the most number of statuses that Ronak has put up. It also states that the most frequently used word by user Ronak is ":P". This was to account the fact that even emoticons are treated as words due to their frequent use and the immense importance they hold in communications over facebook today.

10 Sources

1. http://www.restfb.com/ : Source for the restfb zip file
2. http://www.restfb.com/javadoc/index.html : Java API
3. https://developers.facebook.com/tools/explorer : Graph API explorer, facebook.com