

Facebook Integration with RESTFB API

¹ JASMEET KAUR, ² NEHA SINGH

¹ Student M.Tech Computer Science, ITM University, Gurgaon, Haryana, India
² Asst. Professor, Dept. of CSE and IT, ITM University, Gurgaon, Haryana, India.

ABSTRACT

In almost all the websites present over www, business people has made a significant attempt to integrate its capacity with Facebook to have a connection of millions of users worldwide. Who might have thought of connecting with social media 10 years ago? Here is the answer: Social media integration was not available and most importantly industry overlooked its importance but then. In the early months of 2010 the things were going to get seriously change by the game changers of this e business world by connecting more and more users to have incredible customer reach to sell out their n number of products over all 'range. After the evaluation of Web Services integration Facebook has launched its new connector knows as REST FB API. REST FB is an easy way to connect with your website with Facebook. Facebook is leading social media and if you want to create a social presence of your product of any kind, Facebook integration is a must in these days.

I. INTRODUCTION

RestFB is a simple and flexible Facebook Graph API and Old REST API client written in Java. . It is open source software to integrate Facebook with other web channels. Its features include minimal public API, maximal extensibility, robustness in the face of frequent Facebook API changes, simple metadata-driven configuration and having zero dependencies. Another important aspect is that you can fetch basic user data from Facebook using RestFb. RestFB is a simple and flexible Facebook REST API client written in Java. You can extract friend list of Facebook in your java application, also can post on your walls using RESTFB.

Now a day's majority of web sites are integrated with social networking sites like Facebook and Twitter. The social network site is not complete unless it exposes it's API and allows other web sites to talk to it .The web site cannot feel the same unless it allows its users to have interaction with the social networking site. Facebook

now days come with a convenient Graph API, they have deprecated the old REST APIs. More details on Graph API can be found here <http://developers.facebook.com/docs/api/>. This research paper will cover how a Facebook status message can be shared from a JSP page. For a web site or a web page to talk to Facebook. You need to allow the web site to talk to Facebook .i.e. Facebook needs to authorize the Web site on behalf of the user. Facebook OAuth2.0 protocol is used to authorize the user.

Additionally clouds are heterogeneous ecosystems that connect many different services from many different companies written in many different technology stacks. The complexities of the underlying stacks and protocols should be abstracted away from the business logic so this ecosystem of services can easily connect and work in harmony. A great example of this concept in practice is how simply we can "plugin" social media functionality from Facebook, Twitter, etc. Technically behind those widely used APIs are some very diverse and complex systems. All of that complexity is hidden from us as end users and frankly speaking in just a few minutes we can connect our applications and leverage all of that wonderful social functionality that has become vital for sociality these days. That is a perfect statement to think upon. Facebook comes with set of permissions, in general when a web site is authorized to use Facebook it means only the general public information of the user can be used. Facebook has concept of extended permissions which can be used for extended permission levels like 'publish status messages', sending emails', publish your hangouts etc. What these two APIs can do is:

Get profile picture and name of the user. Get profile picture and name of the user's friends who are also using my app. Also it can post into the user's stream. Allow users to invite their friends to use the application etc.

II. GRAPH API

RestFB is a light weight java implementation of the graph api. But the good thing is it also supports the old rest api. Old rest api is sometimes needed when graph api is not enough. This API is the

primary way for apps to read and write to the Facebook social graph. The Graph API has multiple versions available. It's a low-level

HTTP-based API that you can use to query data; post new stories, upload photos and a variety of other tasks that an app might need to do as per the requirements. The Graph API is named after the idea of a 'social connectivity graph' - a representation of the information on Facebook composed of: nodes (basically "entities" such as a User, a Profile Pic, a Page, a Comment etc) and edges (i.e. the connections between those " entities ", such as a Page's albums, or a Photo's Comments etc) and thirdly fields (info about those " entities ", such as the birthday of a User, the favorite holiday destination of the user or the name of a Page etc).

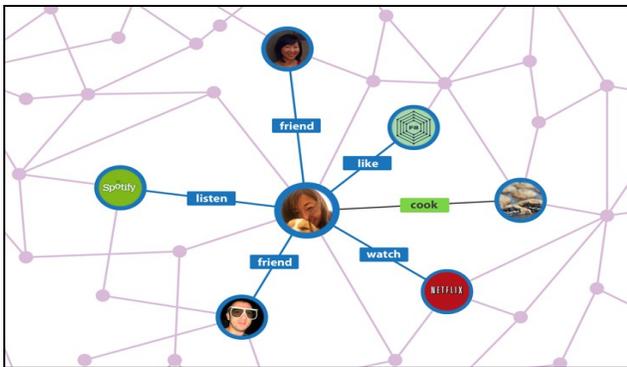


Figure 1-Connecting People

The Graph API is HTTP based so works with any language that has an HTTP library, such as curls, urllib etc. A developer can also use the Graph API directly in your browser, for example [this Graph API request](#) is equivalent to.

```
GET graph.facebook.com
/facebook/picture?
redirect=false
```

Most Graph API requests will require the use of access tokens which your app can generate by implementing Facebook Login.

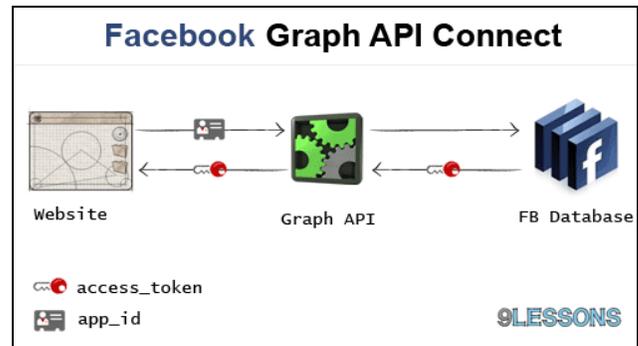


Figure 3- Graph API Connector

The Graph API was launched in March 2010 with the intention of replacing the older REST API. However, there is still functionality in the REST API that does not have any similar method in the new Graph API. Facebook has declared its intention to replace the REST API with the Graph API so it's probably a good idea to use the Graph API whenever possible to help the code.

Essentially there are three methods of using the Graph API: requesting data, posting data and deleting data. Some data can be requested without authentication while most data is dependent on authentication, which user is authenticated, the permissions that user has given, the role that user can play, and the permissions given by the object being queried etc. Although posting the data have similar restrictions. Deleting data is very limited and usually can only be done to by originally posted by the current application and user.

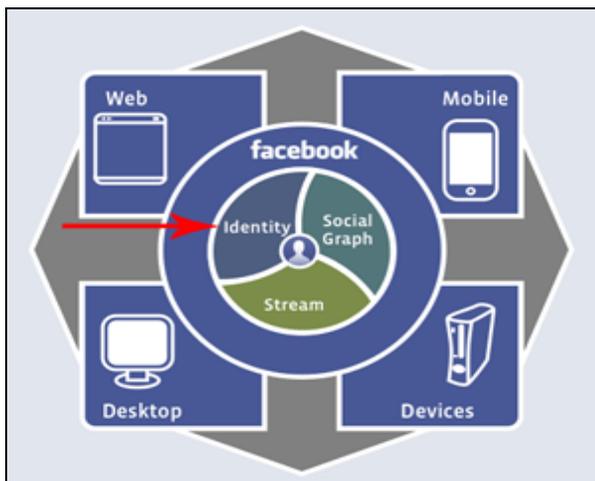


Figure 2 – Connecting Multi Channels

III. OLD REST API

REST stands for Representational State Transfer. It is an architectural pattern for developing web services as opposed to a specification. REST web services communicate over the HTTP specification, using HTTP vocabulary: It contains methods (GET, POST, etc.), HTTP URI syntax (paths, parameters, etc.), media types (xml, json, html, plain text, etc) and HTTP Response codes etc. Its characteristics include uniform interface, decoupled client-server interaction, it is stateless, cacheable, layered etc. Also it is extensible through code on demand. A REST service framework provides a controller for routing HTTP requests to a request handler according to the HTTP method used (e.g. GET, POST), supplied path information (e.g. /service/listItems), query, form, and path parameters and also to headers, cookies, etc.

Now this API has been deprecated by Facebook. The Facebook's legacy REST API does not work for new Facebook apps registered after Apr/2013. However, REST API is still supported for existing Facebook app created before Apr/2013. Now graph API has taken its place.

REST API is not a bad implementation, but the problem with this api is that it uses Facebook old rest api for everything. . Facebook has officially deprecated REST API and says the future is Graph api. The reason is Graph API is much faster and error prone compared to old api. Also lots of timeouts from the old rest api, but the graph api proves to be really fast.

The REST API is the original API for developing on the Facebook platform. Around one year back Facebook decided to go in a new era and launched their new Graph API. It's best to avoid using these methods if there's an equivalent method in the Graph API as Facebook has clearly indicated that they're moving away from these methods.

IV. REST FB BASIC IMPLEMENTATION

RestFB is a simple and flexible Facebook Graph API and Old REST API client written in Java. Its latest version can be downloaded from github/restfb website. Firstly download the restfb jar from the link <http://restfb.com/>.

For maven users following **dependency** needs to be incorporated in the application:

```
<dependency>
<groupId>com.restfb</groupId>
<artifactId>restfb</artifactId>
<version>1.6.14</version>
</dependency>
```

Basically its main purpose of using is to minimize public API, to provide maximum extensibility and robustness in the face of frequent Facebook API changes. It is simple metadata-driven configuration with zero dependencies

Creating **DefaultFacebookClient** is the startup way of initializing the framework:

```
FacebookClient facebookClient = new
DefaultFacebookClient(MY_ACCESS_TOKEN);
```

Furthermore for security purposes **OAuth** access can be taken by navigating to Graph API Explorer app.

Now below code should be written to **fetch single usable objects**:

```
User user = facebookClient.fetchObject("prod", User.class);
Page page = facebookClient.fetchObject("activity", Page.class);

out.println("User name: " + user.getName());
out.println("Page likes: " + page.getLikes());
```

Using Graph API a developer can also **fetch multiple objects** as shown in below example:

```
FetchObjectsResults fetchObjectsResults =
facebookClient.fetchObjects(Arrays.asList("prod", "activity"),
FetchObjectsResults.class);

out.println("User name: " + fetchObjectsResults.prod.getName());
out.println("Page likes: " + fetchObjectsResults.activity.getLikes());
```

A standalone class to hold results:

```
public class FetchMyObjectsResults {
    @Facebook
    User prod;

    //Facebook field name in the annotation.

    @Facebook("activity")
    Page page;
}
```

For **logging** purpose RestFB uses java.util.logging to log the data that's sent over the wire to and from Facebook. Anytime a developer can look at the log output to make sure that RestFB is sending the data as what is expected and that Facebook is returning the correct JSON. A developer can also use Log4j for logging.

Now it's the time to **fetch Connections**:

```
Connection<User> myFriends =
facebookClient.fetchConnection("prod/friends", User.class);
Connection<Post> myFeed =
facebookClient.fetchConnection("prod/feed", Post.class);
```

```
out.println("Total friends: " + myFriends.getData().size());
out.println("First item in the feed: " + myFeed.getData().get(0));
```

```
// Connections support paging and are iterable
for (List<Post> myFeedConnectionPage : myFeed)
    for (Post post : myFeedConnectionPage)
        out.println("Post: " + post);
```

Searching with the help of Connections can be made in similar below patterns:

```
// It is a special case of fetching Connections -
// just pass few more parameters
Connection<Post> publicSearch =
facebookClient.fetchConnection("search", Post.class,
Parameter.with("q", "watermelon"), Parameter.with("type", "post"));
```

```
Connection<User> targetedSearch =
facebookClient.fetchConnection("prod/home", User.class,
Parameter.with("k", "John"), Parameter.with("type", "user"));
```

```
out.println("Public search is here: " +
publicSearch.getData().get(0).getMessage());
```

```
out.println("Posts on my wall by friends named John: " +  
targetedSearch.getData().size());
```

Publishing a status message

```
FacebookType publishMessageResponse =  
facebookClient.publish(prod/feed", FacebookType.class,  
Parameter.with("message", "Hello Facebook!"));
```

V. CONCLUSION

As this era progresses, integration of social media like Facebook would be vital for every business on this earth. The way Facebook make it happen to connect with users is not been achieved by anyone else so far. Here from scalability of business point of view integrating Facebook apps with REST FB would be must to have for all jargon dream sellers. Though few of the complexities and challenges evolve during integration which needs to cater gracefully to satisfy secured business goals.

VI. REFERENCES

- [1] Facebook® Application Development For Dummies®, **By: Jesse Stay**
- [2] <http://cagrikilit.com/2011/01/06/restfb-an-easy-way-to-integrate-with-facebook/>
- [3] <http://restfb.com/javadoc/>
- [4] <http://restfb.com/legacy-rest-api.html>
- [5] <http://rest.elkstein.org/>
- [6] <http://java4eo.blogspot.in/2012/10/java-restfb-and-facebook-authentication.html>
- [7] <https://developers.facebook.com/docs/apps/changelog>



Jasmeet Kaur is an active participant in various international journals and conferences for her research paper work in web services. She is a Masters in Computer Science and currently pursuing her MTech in CSE from ITM University Gurgaon. Her interest areas include Web services, Operating Systems, and Algorithms etc.



Neha Singh She is an MTech from Computer Science and currently pursuing Ph.D. She is working as an Assistant Professor in ITM University Gurgaon. She has published various research papers in International journals and conferences. Her interest areas include web services, java architecture etc.