***Federated Login***

Author: Aravinth

Date: 29.10.2013

*1. Facebook*

In order to use Facebook Login API, an App of Yetu must be registered. In return Facebook provides an App ID which will be used as a credential while initiating Facebook Login [1]. Yetu-App must specify an URL and only the calls made from that End-Point will be authenticated.

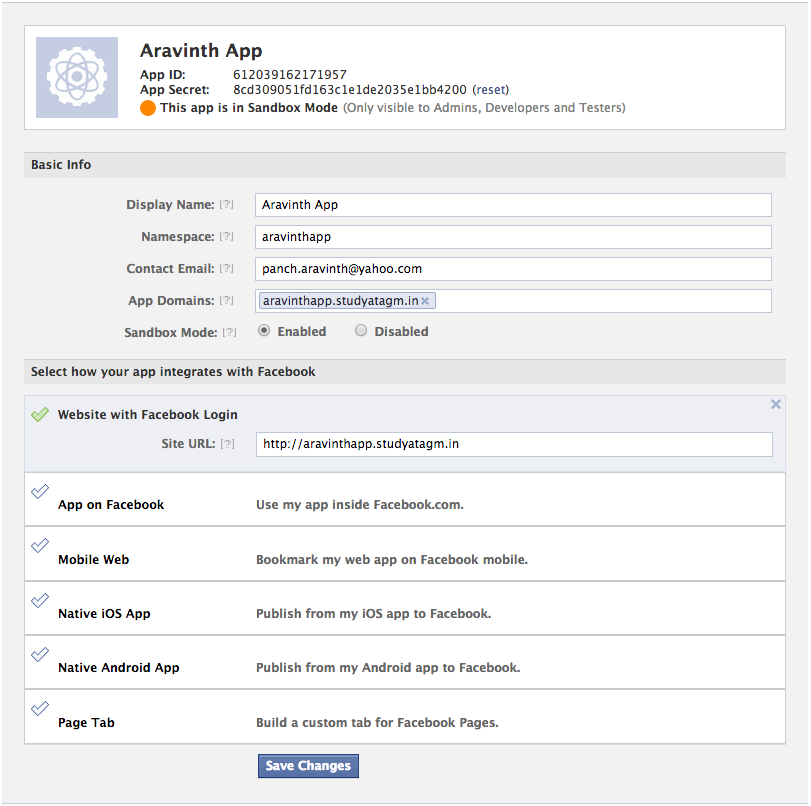


Fig.1 - Facebook App Registration

Fig.1 shows an app registration. Above registered sample app can be accessed at the specified URL for learning purposes. During the development cycle, if Yetu-App is deployed on a local server, then Site URL must be specified as http://localhost:8080/apphome or another Yetu two Facebook App IDs, one for Development and one for Release must be created.

**1.1 Login Flow**

Facebook apps can use one of several login flows, depending on the target device and the project. The login flow for web apps are

* Checking login status
* Logging people in
* Confirming identity
* Storing access tokens and login status
* Logging people out

**1.2 SDKs**

Facebook provides developers with various SDKs to fulfill the state of Cross-Platform [1]. The available SDKs are

* Web-Login with Javascript (Client Side AJAX Authentication)
* Web-Login without Javascript (Server Side / URL based oAuth authentication)
* Native iOS
* Native Android

**1.2.1 Javascript SDK**

Browser based application can use Facebook Javascript SDK where AJAX requests will be sent to Facebook’s Login API and will be authenticated. Response to the authentication requests will be sent as JSON with Access Token and some Session related information.

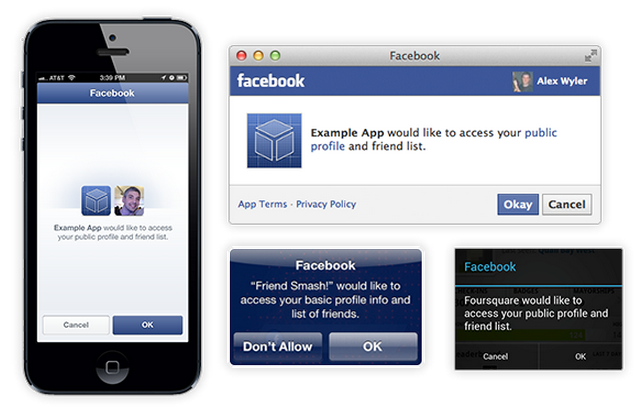


Fig.2 - Login PopUps Prompted by Facebook

Facebook’s Javascript SDK must be loaded from its servers (JS shouldn’t to be downloaded as advised) which inturn creates a Global Object named “FB”. Thereby every methods such as login, logout, getLoginStatus, etc of Facebook API will be directly available under this method. Facebook’s Javascript SDK can also loaded from its servers via JQuery and Require.JS.

Facebook API will allow the user to login only using the Login Dialog Box of Facebook and it will be prompted by Facebook automatically as shown in Fig.2. It also provides plugins to built in Facebook components such Login Button, Graphs etc. Users’ public data can also be retrieved by accessing FB.api.

Session State can be maintained across the pages by storing the access tokens. However, It is done by the Javascript SDK itself and no need to implement that feature. FB.getLoginStatus method will also enable us to verify the login status anytime.

**1.2.2 Non-Javascript SDk**

Facebook uses oAuth [5] authentication procedures for Server Side Authentication. It can also be used in Webview (WebApp inside Desktop APP). The same Login flow as described above will be used to achieve this.



Fig.3 - Facebook oAuth End-Point

Yetu-App sends a HTTP request to a oAuth End-Point (Fig.3) of Facebook API with various parameters such as Yetu-App ID and redirect URI. Facebook API will authenticate the user and send session details as Query Strings or Fragmented Parameters.

Access Tokens must be handled and stored as session variables. Also It must be identified whether Login Access Token is stored for the same person who accepted the Yetu-App’s Facebook login. User can be verified by an another parameter called as code which is sent with authenticated response. Apart from that this whole process must be handled in the backend. This is where the difference between Javascript and Non-Javascript SDK lies. In Javascript SDK storing and inspecting Access Tokens and Identifying the Users are done automatically.

*2. Google*

Google uses federated Login [2], based on the OpenID standard. OpenID [4] provides a framework in which users can establish an account with an OpenID provider such as Google, Yahoo, Facebook, etc.

Google supports the OpenID 2.0 protocol, providing authentication support as an OpenID provider. On request from a third-party site, Google authenticates users who are signing in with an existing Google account, and returns to the third-party site an identifier that the site can use to recognize the user. This identifier is consistent, enabling the third-party site to recognize the user across multiple sessions.

**2.1 Login Flow**

OpenID login authentication for web applications involves a sequence of interactions between your web application, Google's login authentication service (Fig.4), and the end user. The diagram and sequence below describe the process as recommended by Google.

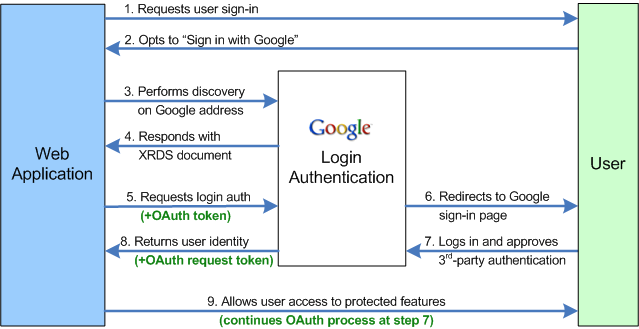


Fig. 4 - Google - OpenID Authentication Flow

**2.2 Extensions**

Google also supports the following extensions

* OpenID Attribute Exchange 1.0 (to get User Data)
* OpenID User Interface 1.0 (to login using Google’s Dialog Boxes)
* OpenID+OAuth Hybrid protocol (to get oAuth Access Sequence with OpenID responses)
* PAPE specification (re-prompting authentication)

**2.3 SDKs**

Google Login using OpenID must be implemented at the server side using the OpenID libraries or sending HTML requests directly to the End-Point of Google OpenID Login. Google uses Java Library [3] which is provided by OpenID. There are also varieties of native language libraries are available.

* C#
* C++
* Java
* Javascript (for Node.JS)
* Python
* PHP
* Ruby on Rails



Fig. 5 - Sample HTTP Request to Google’s OpenID EndPoint

with OpenID Specific Parameters

Fig.5 shows a simple HTTP request sent to Google’s EndPoint with OpenID parameters. There is a large collection of optional parameters provided by OpenID to suit the customer’s need. OpenID User Interface 1.0 allows us to customize the prompt window.

*Summary*

There are various Single-SignOn methods are available and OpenID is the most used open standard federated login.

The Authentication Workflow works in a way that Yetu-App sends a handshake request to the OpenID/Facebook Login API and receives a response with USER INTERFACE URL which is nothing but a Pop-Up Window of Google/Facebook allowing the user to enter their credentials and allows Yetu-App to access user specific informations.

Once it is been authenticated Access Token and USER IDENTITY CODE will be sent back from the servers. Access Tokens are session specific and expires and USER IDENTITY CODE are consistent across the session for a respective Domain. That means USER IDENTITY CODE used for logging in userXYZ is different for http://yetu.com and http://yetu.de. Therefore these USER IDENTITY CODES or simply Gmail or Facebook ids can be used to map an user inside YETU database.

Finally, Google Login can be easily implemented using OpenID and even though Facebook is an active member of OpenID, Facebook advises developers to use Facebook Login API as described in the section 1.

*References*

[1] [Facebook Login API / Facebook Connect](https://developers.facebook.com/docs/facebook-login)

[2] [Google Login using OpenID](https://developers.google.com/accounts/docs/OpenID)

[3] [OpenID Java Library](https://code.google.com/p/openid4java/?hl=de)

[4] [OpenID](http://openid.net)

[5] [oAuth 2.0](http://oauth.net/2)