**Sprint 1 (Members area)**

**Created : 7-20-2013 ola lawal**

Important notes:

User profile must be first authenticated , and the profile id must be re-trived to use the services internal to Anewluv.

\*\* Coming soon : API key will be required on an app per app basis to access the member services. So as you code note you might have to add the follow header to your JSON calls \*\*

The Homepage

To access these API’s user must be validated via either OPEN ID ( face book, Yahoo,Google) or via the api with their username and password

**AuthenticationService.Docx** ( containes the documentation for required authentication calls)

List of API service calls within the authenciation service used to validate the user profile :

1. UserName and Password -

see AuthenticaionService APi document use the call : **validateuserbyusernamepassword**

1. OpenID/Email (i.e facebook , google , yahoo)

See AuthenticationServiceApi document use the call : **validateuserbyopenid**

**Main Service calls needed broken down by section**.

Steps -

Step 1)

Once user is authenticated access to the home page and the member services are allowed.

After the user is logged in either via registration APIs or the Login api the user’s profile ID should be cached or locally stored to be used with the API. To re-retrive the profile id see documentation in

MemberService.docx(containes documentation for the member service calls)

There are three ways to get the profile id similar to the authentication.

1. If the user name logged in with username/password use the following service call

See MemberService.docx use the call : **getprofileidbyusername**

1. If the user name logged in with Face book or open Id use the following service call

See MemberService.docx use the call : **getprofileidbyopenid**

1. If the user is already logged on or you do not have acces to either of those items above use

See MemberService.docx use the call : **getprofileidbyscreenname**

Step 2)

**Profile Activity Counters**

See the service call document for **MemberActionsService.docx**

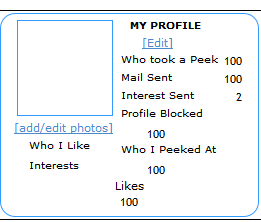
**\*\* only documentation for likes and blocks is listed right now but the service calls for peeks, interests and are the same , just different urls see the service help page :**

[**http://173.160.122.195/MemberActionsService/MemberActionsService.svc/Rest/help**](http://localhost/Shell.MVC2.Web.MemberActionsService/MemberActionsService.svc/Rest/help)

You are now ready to make the service calls to member actions to populate the counters on the page

See the section below.

WEB



MOBILE

Who took a peek count – See **MemberActionsService.docx** use the call getwhopeekedatmecount

Insterest Sent - **MemberActionsService.docx** use the call getwhoiaminterestedincount

Mail Sent : not implemented yet

Profile Blocked - See **MemberActionsService.docx** use the call getwhoiblockedcount

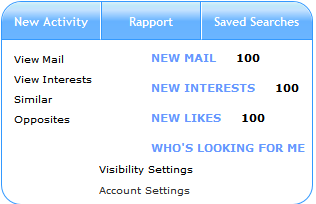
Who I peeked at - See **MemberActionsService.docx** use the call getwhoipeekedatcount

Likes - See **MemberActionsService.docx** use the call getwholikesmecount

Who I like ( text goes below) - See **MemberActionsService.docx** use the call getwhoilikecount

Interests – See **MemberActionsService.docx** use the call getwhoisinterestedinmecount

**New Activity panel (WEB )**



New Mail – Non implemented yet :

New Interestets – See **MemberActionsService.docx** use the call getwhoisinterestedinmenewcount

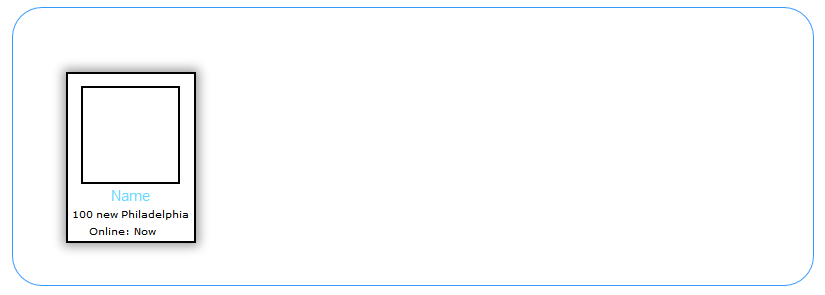
New Likes - See **MemberActionsService.docx** use the call getwhoislikesmenewcount

Who’s looking for me : Not implemented yet

**New Activity panel (MOBILE )**

**My Matches Section ( on mobile this is at the top)**

See the service call document for **MembersMapperService**.docx for WEB/Mobile Matches



The data for this panel is a list of up to 50 profiles pulled from the service call :

1. **MembersMapperService**.docx use the call Getquickmatches

Detaililed instructions

Sample Displaying Macthes via MVC.

The matches should be displayed in either an MVC template or partial page as to allow for proper asynch display of the data.

I.e say your home page is called **Home.chtml** you should have a div or area where you can embed a partial page called something like **\_Matches.chtml**

Once you call the the service that passes the data :

– See **MembersMapperService.docx** use the call getquickmatches

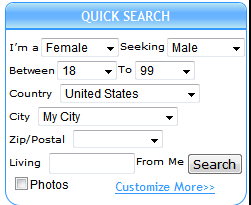
you can then create a loop in your html page to show the matches using page real estate as allowed in your web page or native application. For details on what fields are what see the service doc.

For paging use either a scrollable bar or something similar to this:

<http://getbootstrap.com/2.3.2/components.html#pagination>

pagination must be on bottom. For web.

**Quick Search (on mobile this is a separate search page)**



The quick search defaults to the user’s country and city as well as zip/postal code

The default value in living for me is 500

That information is pulled from the user’s search settings details coming soon.