communitybrands

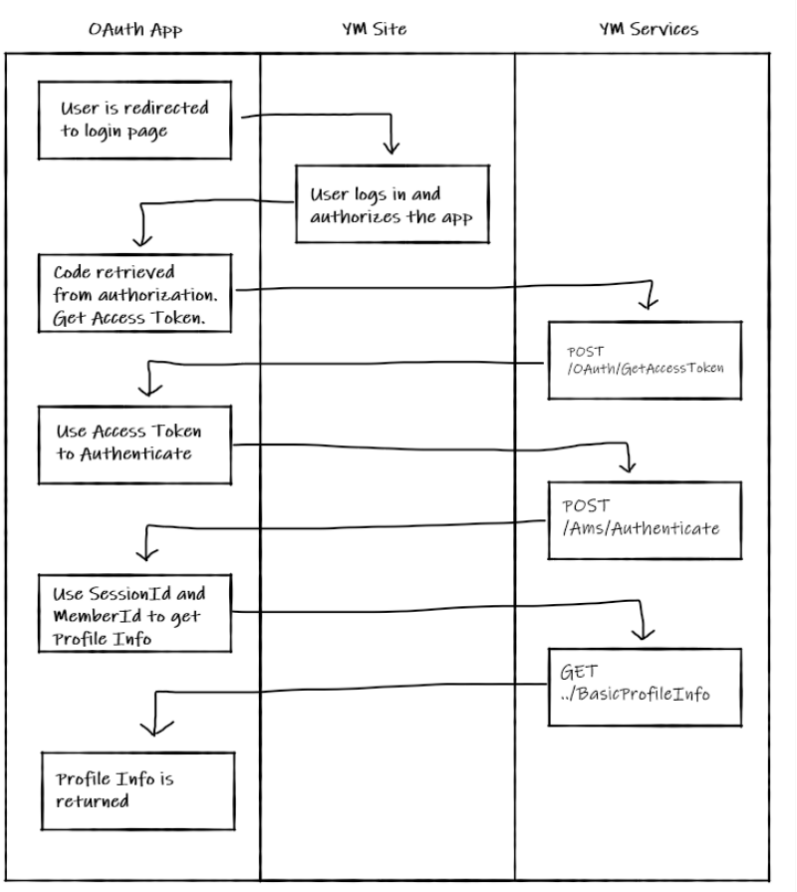
yourmembership®

Getting Started with OAuth

Overview

The general OAuth flow consists of the following steps:

1. The user signs in on your YM site and authorizes your app.
2. Get the code from the authorization and get the access token.
3. Use the access token to authenticate the user.
4. Use the session from the authentication to make calls to the API on behalf of the user (e.g., getting prole



Getting the user's authorization

In order to get the user's authorization, you need to have them log in from a specific path on your site (lock.aspx) along with some parameters. Let's take the following table and add example to them:

Parameter	Description	Example
Base URL	Primary URL for your YM site	https://www.professional.com
app_id	The App ID of your OAuth app as generated in YM	AbCdEfG12345
Redirect_uri	The redirect URL, as entered in your OAuth app	https://members.pro.app/callback
Scope	The scope(s) established in your OAuth app: basic_profile, full_profile, or both (basic_profile, full_profile)	basic_profile

Taking these values, we would build the URL as follows:

https://www.professional.com/lock.aspx?app_id=AbCdEfG12345&redirect_uri=https://members.pro.app/callback&scope=basic_profile

The parameters can be in any order, but this is where you would direct your members to sign in and authorize the application. On successful authorization, the browser will redirect to your redirect URL

(e.g., https://members.pro.app/callback) with a query string parameter "code" that is used to get the access token (e.g., https://members.pro.app/callback?code=code101010).

Getting the access token

After authorization and getting redirected to the redirect URL, this route should be handling the code that is passed as a query string. The parameter is labeled as "code" and is used with the GetAccessToken service in the REST API. Continuing on our example, here are the parameters needed:

Parameter	Description	Example
GetAccessToken Route	The route to get the access token	/OAuth/GetAccessToken
AppID	The App ID of the OAuth app as generated in YM	AbCdEfG12345
AppSecert	The App Secret of your OAuth app as generated in YM	SECRETAbCdEfG12345
GrantType	The grant type to get the Access Token. Possible values: Code, RefreshToken	Code
**Code*	*The code from the authorization step*	*Code101010*
**RefreshToken*	*The refresh token of the record in question*	*Refresh101010*

In this case, we would be using the Code parameter and not the RefreshToken parameter. The refresh token is used when you already have the refresh token and need to get a new access token. Once you have all your parameters, you would make the following call:

Endpoint	Type
https://ws.yourmembership.com/OAuth/GetAccessToken	POST

Body
 	
{
AppId: "AbCdEfG12345",
AppSecert: "SECRETAbCdEfG12345",
GrantType: "Code",
Code: "code101010"
}	

A successful response from this call will return a series of datapoints including the AccessToken and its expiration. This token will be used to authenticate to the REST Services. For this example, let's say the access token returned was a1b2c3d4e5.

Authenticating the user to services

Now that you have the access token, you can now pass that to the Auth service to get the necessary session created. Continuing on our example, here are the parameters needed:

Parameter	Description	Example
Auth Route	The route to authenticate to services	/Ams/Authenticate
ConsumerKey	The App ID of your OAuth app as generated in YM	AbCdEfG12345
ConsumerSecret	The App Secret of your OAuth app as generated in YM	SECRETAbCdEfG12345
AccessToken	The access token returned from the GetAccessTokenService	A1b2c3d4e5
ClientID	The ID of your YM site	12345
UserType	The type of user authenticating to the service. Possible values: Admin, Member	Member

Once you have all of your parameters situated, you would make the following call:

Endpoint	Type
https://ws.yourmembership.com/Ams/Authenticate	POST

Body

```
{
   ConsumerKey: "AbCdEfG12345",
   ConsumerSecret: "SECRETAbCdEfG12345",
   AccessToken: "a1b2c3d4e5",
   ClientID: 12345,
   UserType: "Member",
}
```

A successful authentication will return another series of values including two very important values:

- SessionId: The value to be passed into the “X-SS-ID” header for subsequent requests.
- MemberId: The ID to be passed in any path variables for the member.

These two, in conjunction with the ClientID, will be sued to make calls to other services as needed.

Getting the authenticated user's profile information

Since we authenticated the user with the Auth service, we can now use the Session ID, the Member ID, and the Client ID to get the member's prole information using the BasicMemberProfile service. We are using this service versus the MemberProfile service as our application is only using the basic_profile scope. Continuing our example, here are the parameters needed:

Parameter	Description	Example
BasicMemberProfile Route	The route to get the user's information	/Ams/:ClientID/Member/:MemberID/BasicMemberProfile
X-SS-ID	The SessionId returned from the auth service	AUTH123
ClientID	The ID of your YM site	12345
MemberID	The ID of your member record as returned from the auth service	987654321

Once you have all of your parameters situated, you would make the following call:

Endpoint	Type
/Ams/12345/Member/987654321/BasicMemberProfile	GET

Headers
 	
{
…
X-SS-ID: ”AUTH123”,
…
}	