Using OAuth Authorization Grant type retrieve user information from the Facebook.

What is OAuth (Open Authorization), and how it works?

OAuth allows notifying a resource provider (e.g. Facebook) that the resource owner (e.g. you) grants permission to a third-party (e.g. a Facebook Application) access to their information (e.g. the list of your friends). You can use OAuth to get a token via a callback url and then use that token to make calls to the Facebook API to get their use data until the token expires. Websites rely on it because it allows programmers to access their data without the user having to directly disclose their information and spread their credentials around online but still provide a level of protection to the data.

For an example consider this scenario,

Say you have an existing Gmail account. You decide to join LinkedIn. Adding all of your many, many friends manually is tiresome and error-prone. You might get fed up half-way or insert typos in their e-mail address for invitation. So you might be tempted not to create an account after all.

Facing this situation, LinkedIn has the Good Idea(TM) to write a program that adds your list of friends automatically because computers are far more efficient and effective at tiresome and error prone tasks. Without an API for exchanging this list of contacts, you would have to give LinkedIn the username and password to your Gmail account, thereby giving them too much power.

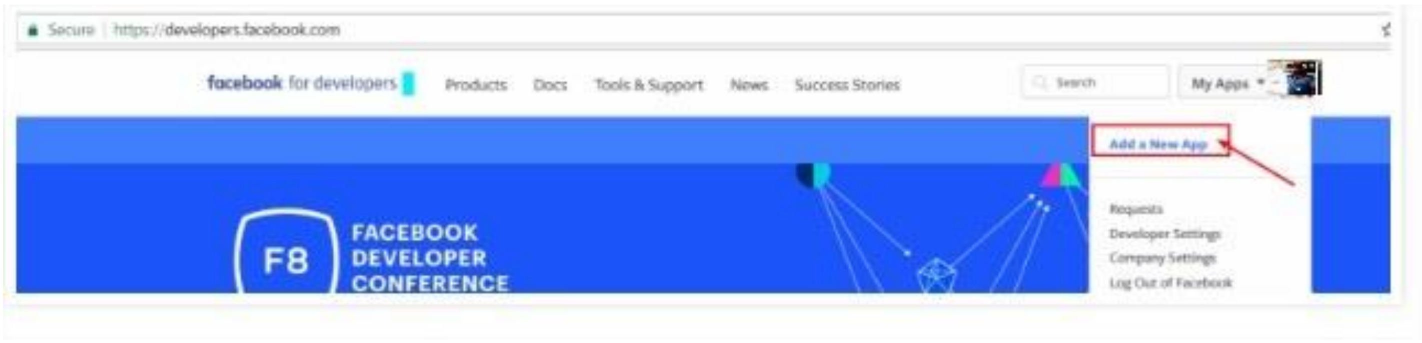
This is where OAuth comes in. If your Gmail supports the OAuth protocol, then LinkedIn can ask you to authorize them to access your Gmail list of contacts.

OAuth allows for:

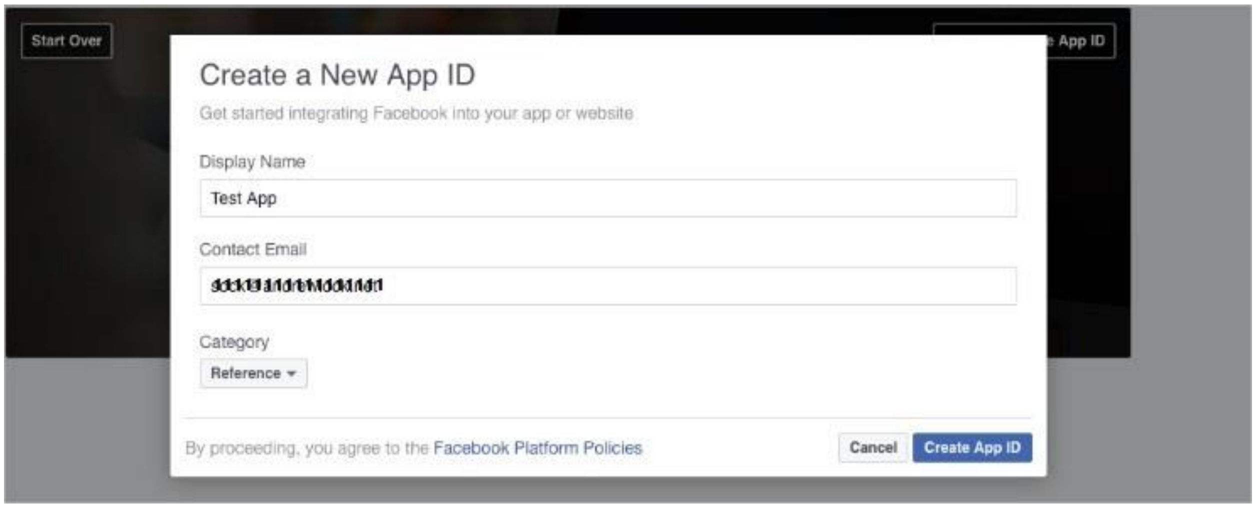
- Different access levels: read-only VS read-write. This allows you to grant access to your user list or a bi-directional access to automatically synchronize your new LinkedIn friends to your Gmail contacts.
- Access granularity: you can decide to grant access to only your contact information (username, e-mail, date of birth, etc.) or to your entire list of friends, calendar and what not.
- It allows you to manage access from the resource provider's application. If the third-party application does not provide mechanism for cancelling access, you would be stuck with them having access to your information. With OAuth, there is provision for revoking access at any time. For more about the OAuth [1.5].

Configure the Facebook Application.

First step is to create an application in the developer account on Facebook and add a new application [3].



Provide a display name for your application and your contact email and create the application.



Create a New App ID

Get started integrating Facebook into your app or website

Display Name

Test App

Contact Email

[redacted]

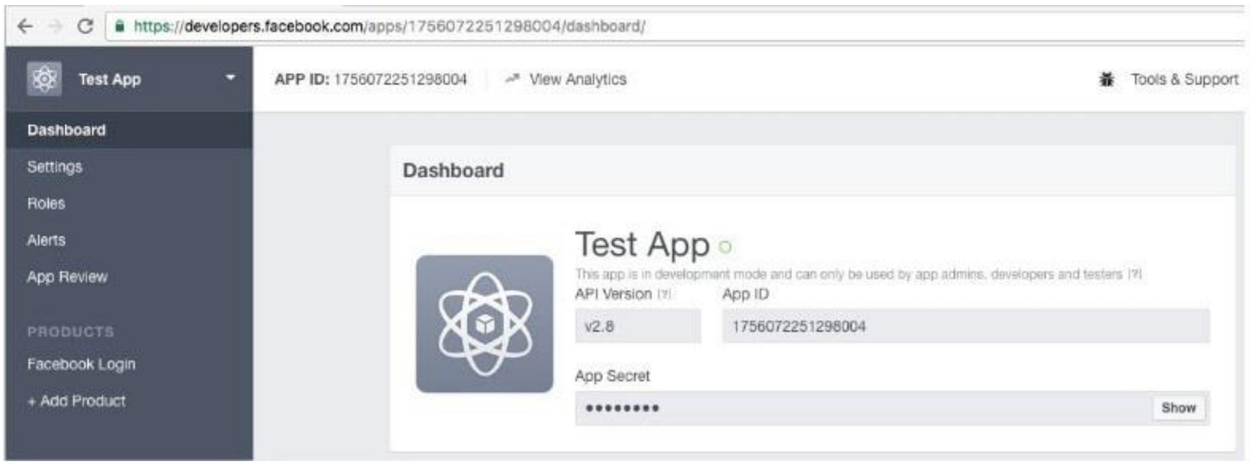
Category

Reference

By proceeding, you agree to the Facebook Platform Policies

Once created, navigate to https://developers.facebook.com/apps and click on your application's icon. You will be taken to your app's basic details. Here you can obtain the App Id and App Secret you will need in your application. Make a note of them (store them using your secrets manager).

In the Dashboard, you can see the App ID and the App Secret for your app. In OAuth terminology, we call the same as Client ID and Client Secret, or Consumer Key and Consumer Secret.



APP ID: 1756072251298004

Dashboard

Test App

This app is in development mode and can only be used by app admins, developers and testers [?]

API Version [?]: v2.8

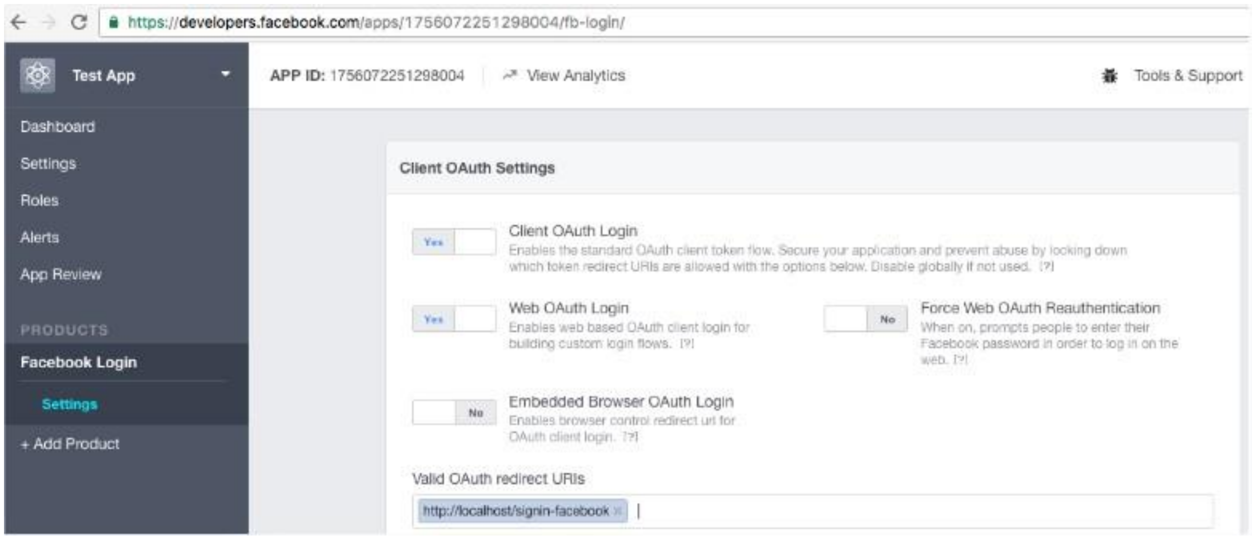
App ID: 1756072251298004

App Secret: •••••••••

The last step is to configure the redirect URI for your application. Click on '+ Add Product' at the bottom of the menu and choose Facebook Login. This will enable OAuth for your application, and allow you to set the REDIRECT_URL for your application.

This URL should be within your client web application and Facebook will send all responses to this URL. However, for trying out this flow, you don't need to have a working URL available. You can simply provide a dummy URL here for the moment. The same URL you add here should be sent along with requests in next steps.

The redirect path for the Facebook middleware is /signin-facebook. In my case, here it runs the app locally.



APP ID: 1756072251298004

Client OAuth Settings

Client OAuth Login: Yes
Enables the standard OAuth client token flow. Secure your application and prevent abuse by locking down which token redirect URIs are allowed with the options below. Disable globally if not used. [?]

Web OAuth Login: Yes
Enables web based OAuth client login for building custom login flows. [?]

Force Web OAuth Reauthentication: No
When on, prompts people to enter their Facebook password in order to log in on the web. [?]

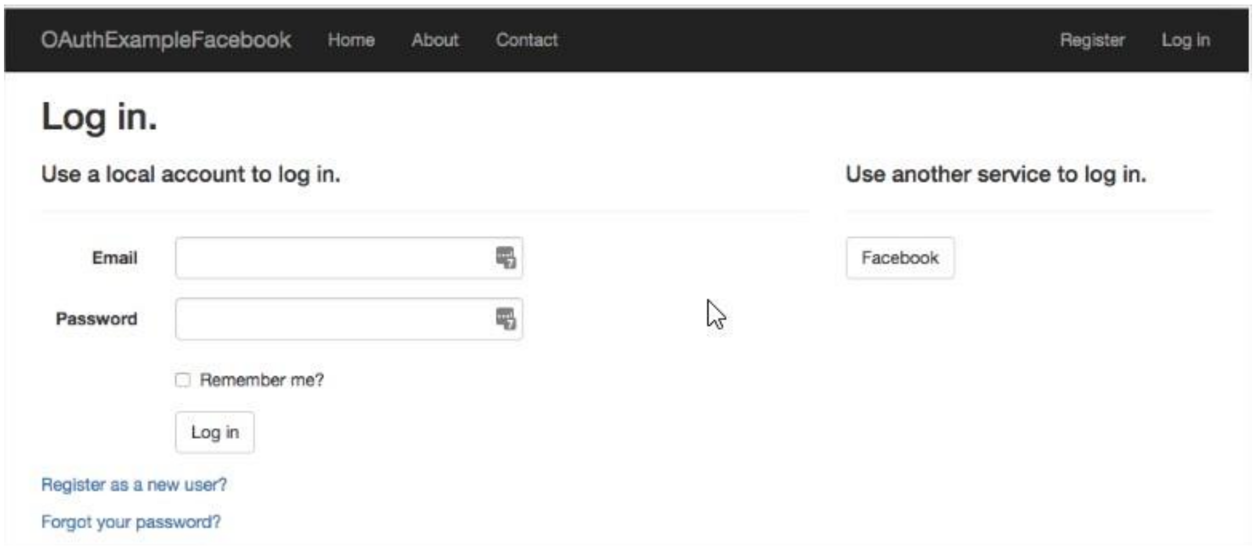
Embedded Browser OAuth Login: No
Enables browser control redirect url for OAuth client login. [?]

Valid OAuth redirect URIs: http://localhost/signin-facebook

Now we have successfully registered our app in Facebook and configured it. You need to take down the App ID and App Secret which is generated for your app and also the Redirection Endpoint URL which you defined where we will use these three values in next steps when making requests to Facebook for retrieving user resources.

Authorizing to obtain the authorization code.

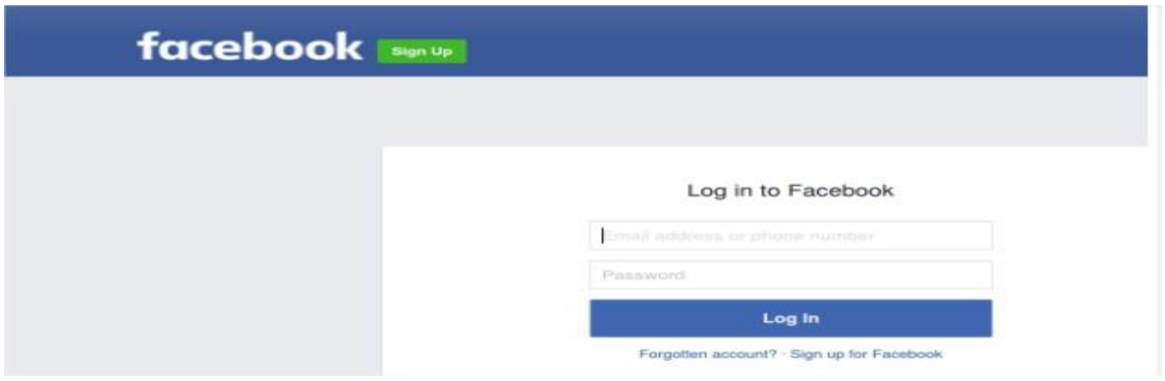
When the user requests a page on your app that requires authorizations, they will be redirected to the login page. Here they can either login using a username and password to create an account directly with the site, or they can choose to login with an external provider - in this case just Facebook. In order to obtain the authorization code from facebook, it needs to send a HTTP GET request to the Authorize Endpoint of Facebook, which is https://www.facebook.com/dialog/oauth Along with the request, it is needed to send several parameters.



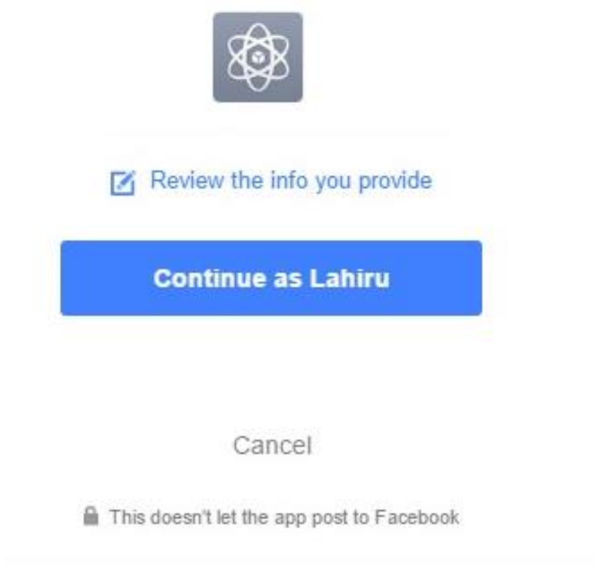
Sample Request:

```
https://www.facebook.com/v2.8/dialog/oauth?client_id=CLIENT_ID&scope=public_p
rofile,email&response_type=code&redirect_uri=REDIRECT_URI&state=STATE_TOKEN
```

If you are not logged into facebook in the browser, first it will ask you to login.


facebook
Sign Up
Log in to Facebook
Email address or phone number
Password
Log In
Forgotten account? · Sign up for Facebook

Once you login, it will show the following popup. We call this as the "User Consent Page" in OAuth terminology. In there, it will show what are the resources from the user account that this external app would be able to access on behalf of you.


Review the info you provide
Continue as Lahiru
Cancel
This doesn't let the app post to Facebook

If the user clicks OK, then Facebook sends another 302 response to the browser, with a URL similar to the following (used port as 5000)

```
http://localhost:5000/signin-facebook?code=AUTH_CODE&state=STATE_TOKEN
```

Facebook has provided an AUTH_CODE, along with the STATE_TOKEN, supplied with the initial redirect. The state can be verified to ensure that requests are not being forged by comparing it to the version stored in our session. The AUTH_CODE however is only temporary, and cannot be directly used to access the user details we need. Instead, it is needed to exchange it for an access token with the Facebook Authorization server.

Obtaining the Access Token.

The client web application has to send a HTTP POST request to the Token Endpoint of facebook sending the authorization code received in previous step. The Token Endpoint of facebook is https://graph.facebook.com/oauth/access_token .

We need to send the following parameters in the body of the HTTP POST request.

```
POST /v2.8/oauth/access_token HTTP/1.1
Host: graph.facebook.com
Content-Type: application/x-www-form-urlencoded

grant_type=authorization_code&
code=AUTH_CODE&
redirect_uri=REDIRECT_URI&
client_id=CLIENT_ID&
client_secret=CLIENT_SECRET
```

- **grant_type=authorization_code** - The grant type for this flow is authorization code
- **code=AUTH_CODE_HERE** - This is the code you received in the query string
- **redirect_uri=REDIRECT_URI** - Must be identical to the redirect URI provided in the original link
- **client_id=CLIENT_ID** - The client ID you received when you first created the application
- **client_secret=CLIENT_SECRET** - Since this request is made from server-side code, the secret is included [2].

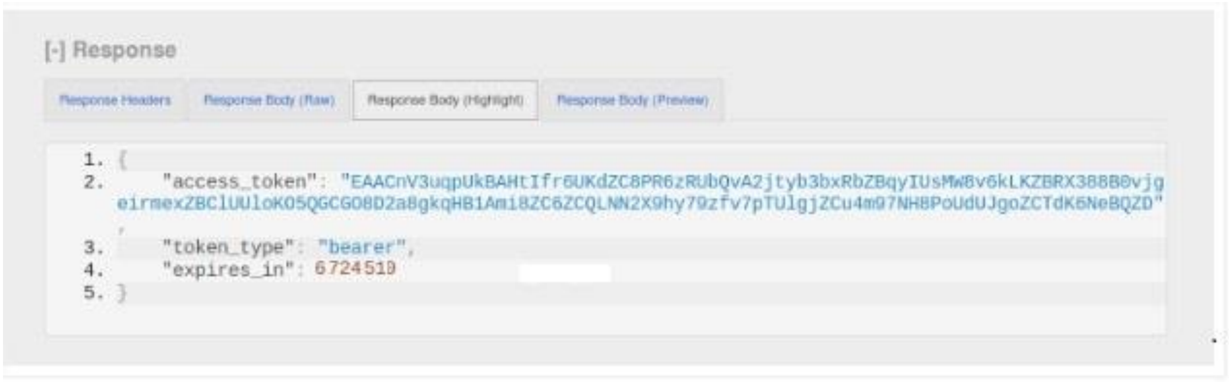
If the token is accepted by Facebook's Authorization server, then it will respond with (among other things) an ACCESS_TOKEN. This access token allows our application to access the resources (scopes) we requested at the beginning of the flow, but we don't actually have the details we need in order to create the Claims for our user yet.

After receiving and storing the access token, our app can now contact Facebook's Resource server. We are still completely server-side at this point, communicating directly with Facebook's user information endpoint.

In addition to that, it needs to send credentials of the facebook application (App ID and App Secret) in the HTTP Header. Here, it wants to combine the App ID and Secret separating them in via Colon (:) and the value should be encoded in Base64.

```
GET
/v2.8/me?access_token=ACCESS_TOKEN&fields=name%2Cemail%2Cfirst_name%2Clast_na
me
Host: graph.facebook.com
```

In the Response, we receive the Access Token.



```
{
    "access_token": "EAACnV3uqpUkBAHtIfr6UKdZC8PR6zRUbQVA2jtyb3bxRbZBqyIUsMW8v6kLKZBRX388B0vjgeirmexZBClUUloKO5QGCGO8D2a8gkqHB1Ami8ZC6ZCQLNN2X9hy79zfv7pTUlgjZCu4m97NH8PoUdUJgoZCTdK6NeBQZD",
    "token_type": "bearer",
    "expires_in": 6724510
}
```

Providing the Access Token retrieving the resources from Facebook.

Now that we have received the OAuth access token from facebook [3], in all the requests we make to the facebook API, we need to include it as a HTTP header.

Send a HTTP GET request to https://graph.facebook.com/v2.8/me/feed?limit=25 and in response, you will get the user's timeline posts. You will get a JSON response. You can limit the number of results using the limit query parameter.

Get User's Facebook ID

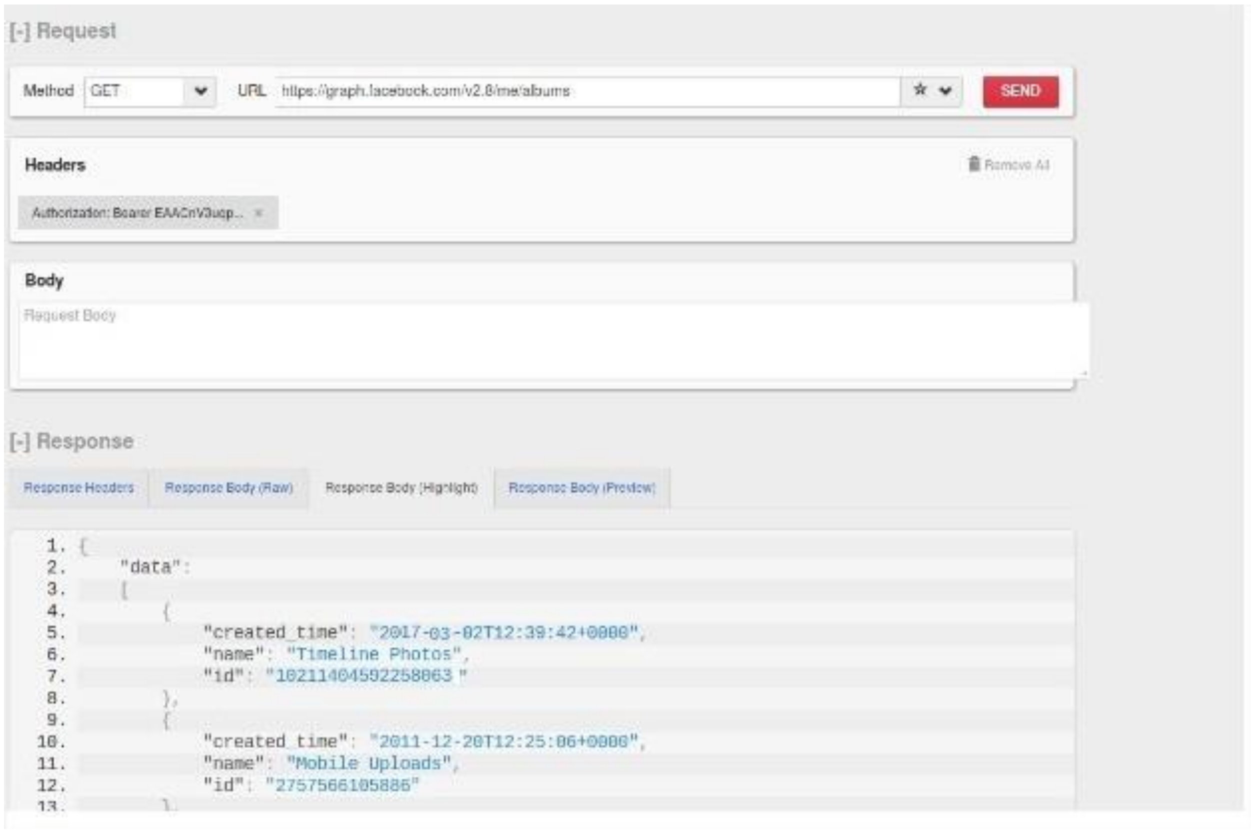
For invoking many operations like retrieving user's photos, albums etc. We need to know the user's facebook ID. For that, we can send a HTTP GET request to the URL https://graph.facebook.com/v2.8/me?fields=id which would return the ID in a JSON response.

```
{
  "id": "3147895201XXXXXXX"
}
```

For more information, you can refer the documentation [4].

Retrieving User’s Photo Album Details

For this, it needs to send a HTTP GET request to https://graph.facebook.com/v2.8/me/albums which would return a JSON response with the photo album details.



[-] Request

Method GET URL https://graph.facebook.com/v2.8/me/albums SEND

Headers

Authorization: Bearer EAACnV3uqp...

Body

Request Body

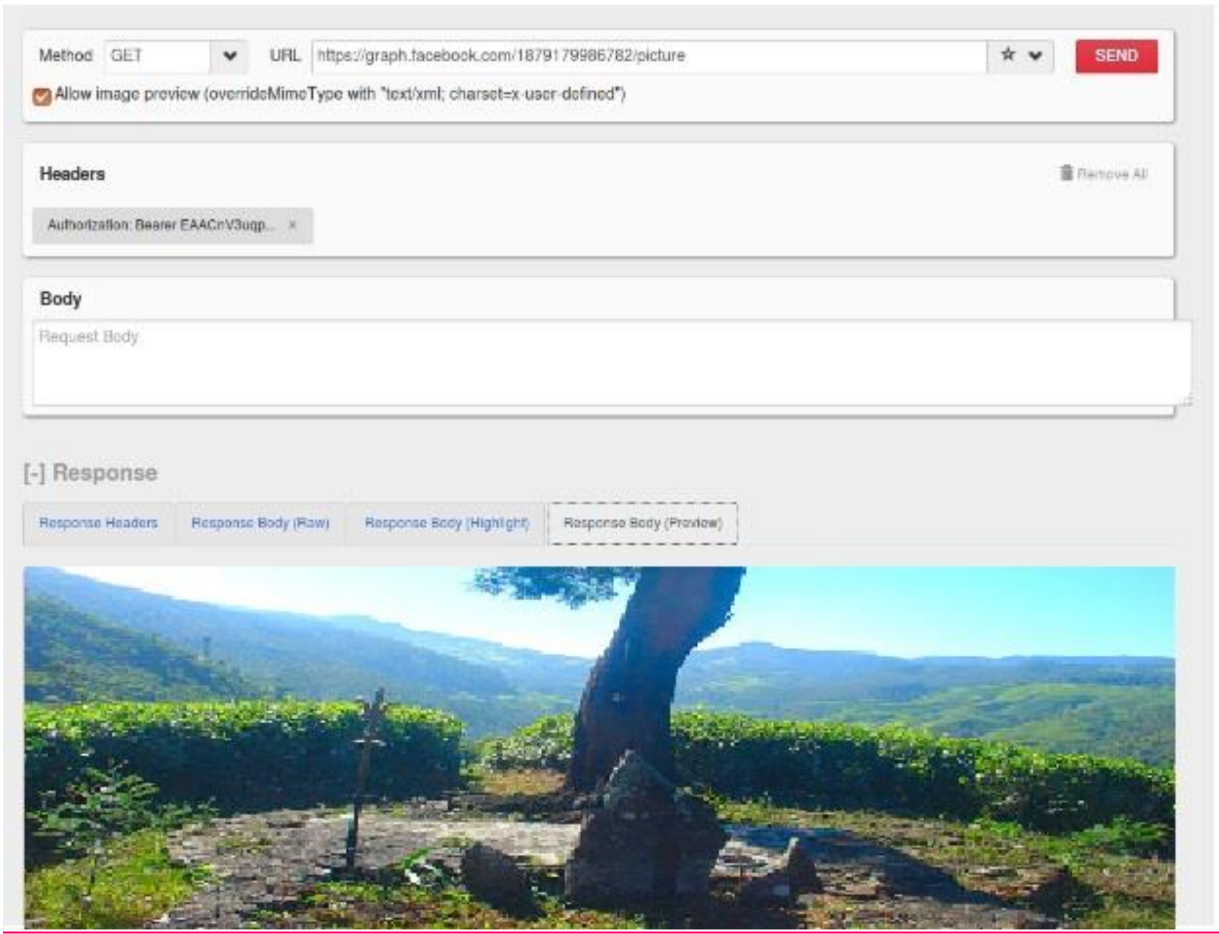
[-] Response

Response Headers | Response Body (Raw) | Response Body (Highlight) | Response Body (Preview)

```
{
    "data":
    [
        {
            "created_time": "2017-03-02T12:39:42+0000",
            "name": "Timeline Photos",
            "id": "10211404592258063 "
        },
        {
            "created_time": "2011-12-20T12:25:06+0000",
            "name": "Mobile Uploads",
            "id": "2757566105886"
        }
```

Retrieving a Photo by ID

It is known the ID of a photo from previous step, then it can get the details of it by sending a HTTP GET request. https://graph.facebook.com/<photo ID>/picture.



Method: GET

URL: https://graph.facebook.com/1879179986782/picture

Allow image preview (overrideMimeType with "text/xml; charset=x-user-defined")

Headers

Authorization: Bearer EAACnV3uqp...

Body

Request Body

[-] Response

Reference

[1] https://en.wikipedia.org/wiki/OAuth
[2] https://github.com/thariyarox/facebook-oauth-samples/tree/master/facebookapp/src/main/java/com/tharindu/oauth/facebookapp/util
[3] https://developers.facebook.com
[4] https://developers.facebook.com/docs/graph-api/reference/
[5] https://tools.ietf.org/html/rfc6749

By S.H.M Lahiru Prabath Balasuriya.