

Assignment 2

Using OAuth 2.0 Authorization framework

Viraj Dissanayake

MS22904928

Using OAuth 2.0 Authorization framework

Author – Viraj Dissanayake

In this article you will be able to see how third party applications obtains restricted user information from a resource owner.

The process of accessing the data by the client depends on the access token

This access token provides the capability to the client to grant the access to the user data, which has been allowed by the resource owner and the user

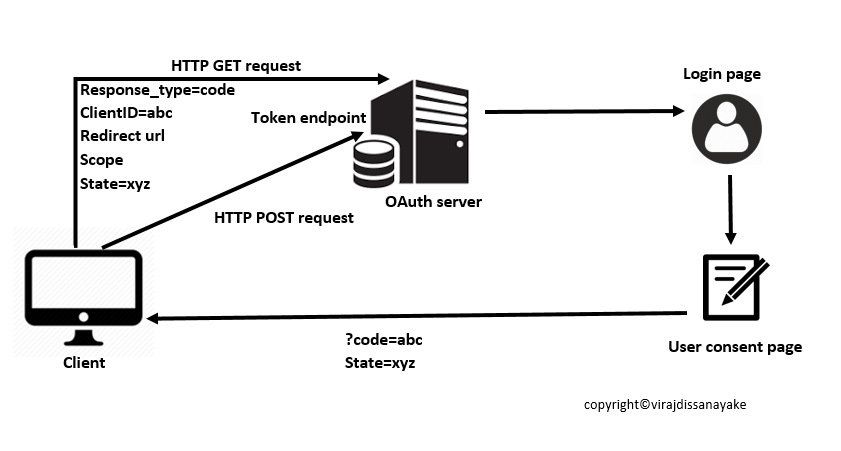
The way that the client get access token is known as Grant type

There are four main grant types, which will be discussing on this article

1. Authorization code grant type
2. Implicit grant type
3. Client credentials grant type
4. Resource owner password credentials grant type

This link will provide the access to the OAuth 2.0 Authorization framework speciation document - <https://tools.ietf.org/html/rfc6749>

1. Authorization code grant type



The above flow can be stated as follows

* Direct the user to the OAuth server
* User approves the app request
* Redirect the user to the application (authorization code is send as a query parameter)
* Authorization code get exchanged to an access token

The client identifier and redirection URI is sent to authorization end point of the authorization server, as the initial step

When sending the request there are some parameters you need to set

response type and client\_id are defined as required field and redirect\_uri, scope, state can be defined as optional parameters

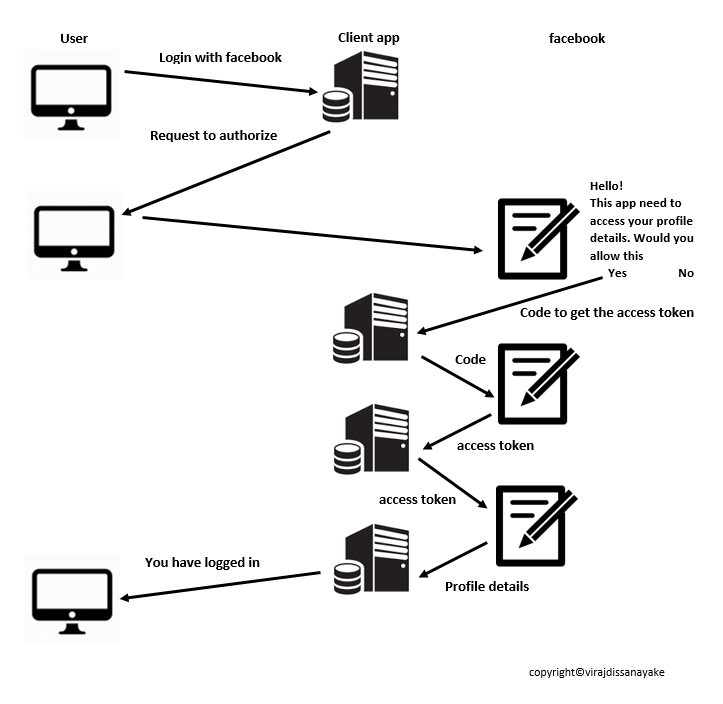
After resource owner is get authenticated by authorization server sends back an authorization code in the redirect URI

The client sends a POST request to the Token endpoint of the authorization server by including the authorization code

After authenticating client and validating the authorization code, the authorization server sends the access token to the client as the response

This grant type is usually used by web apps and native apps which the client has to be confidential. And it is use to refer the access token up on user authorization to the app. This can be also used to get the refresh token. Refresh token is also credentials, which used to obtain access token. When the current access token gets expired or become invalid, the authorization server provides a refresh token to the client.

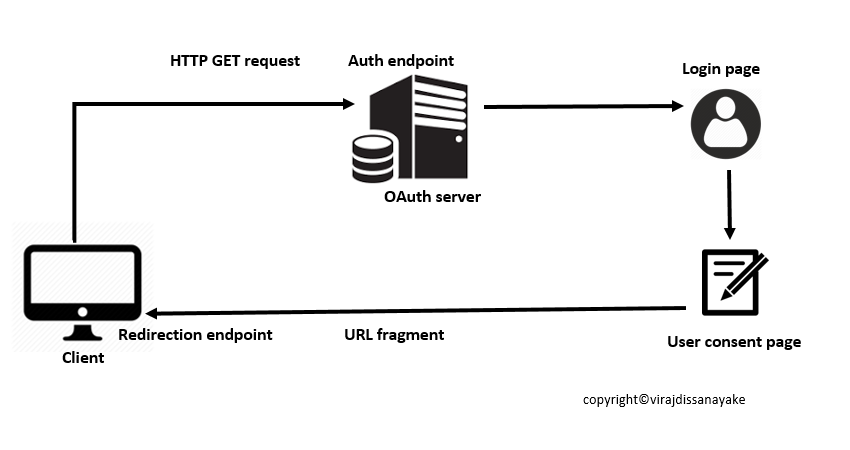
Since we use the Authorization grant type in the app we are going to develop, let’s look at the high level communication between the app, user and the resource server.



When user select the Login with Facebook option, the client app requests the user to allow the app to access the profile details which required by the user to log in the user.

If the user allows that, the resource owner, in this case the facebook send a random code to get the access token. Using this access token, the client app grant access to the required details which has been allowed to use.

1. Implicit grant type



Typically implemented for client side scripting, which the clients are public. Use to get access token, not refresh tokens.

Upon authorization request, the client receives the access token.

Implicit grant type does not authenticate the client

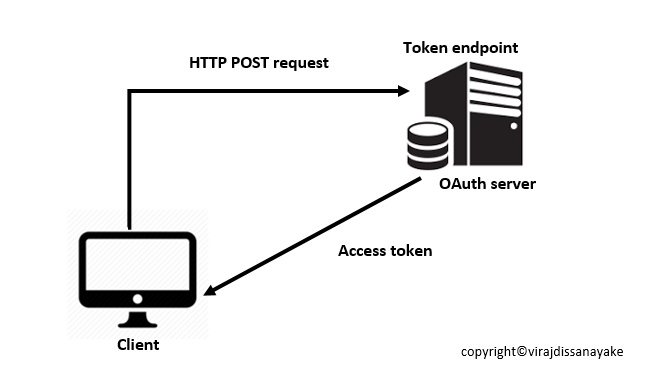
The client sends the client identifier, redirection URI, scope and state to the Authorization endpoint of the authorization server

When sending the request there are some parameters you need to set

response type and client\_id are defined as required field and redirect\_uri, scope, state can be defined as optional parameters

Then the resource owner gets authorization server and after successful authentication and grant access.

The authorization server includes the access token in the URI fragment and send to the client and client extract the access token from the URI fragment

1. Client credential grant type

This grant type is not ideal to protect user data

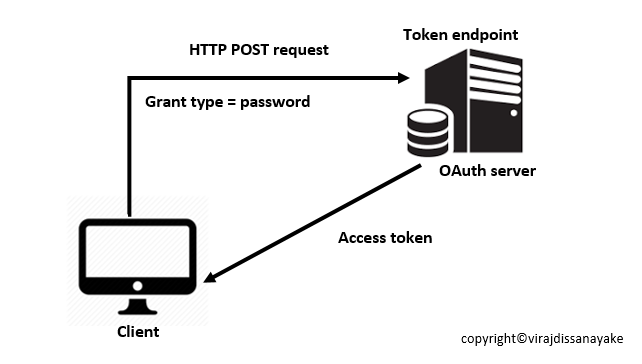
Typically use in server to server communication

When the protected resources are under the client, it can use only the client credentials and gain the access token

In the initial step, the client authenticates the authorization server and request for the access token by sending a request to the token endpoint of the authorization server

grant type is defined as required field and scope can be defined as optional parameters

Then the authorization server sends the access token to the client after validating the client

1. Resource owner password credential grant type

The password credential grant type is use in ‘Trusted apps’ like facebook app

When the resource owner has to have the trust relationship, this grant type is used.

The client should be capable to obtain resource owner credentials.

As the initial step, the resource owner has to provide credentials (username and password) to the client. When sending the request there are some parameters you need to set

grant type, username and password are defined as required field and redirect\_uri, scope, state can be defined as optional parameters

Then the client make request for the access token from the authorization server token endpoint

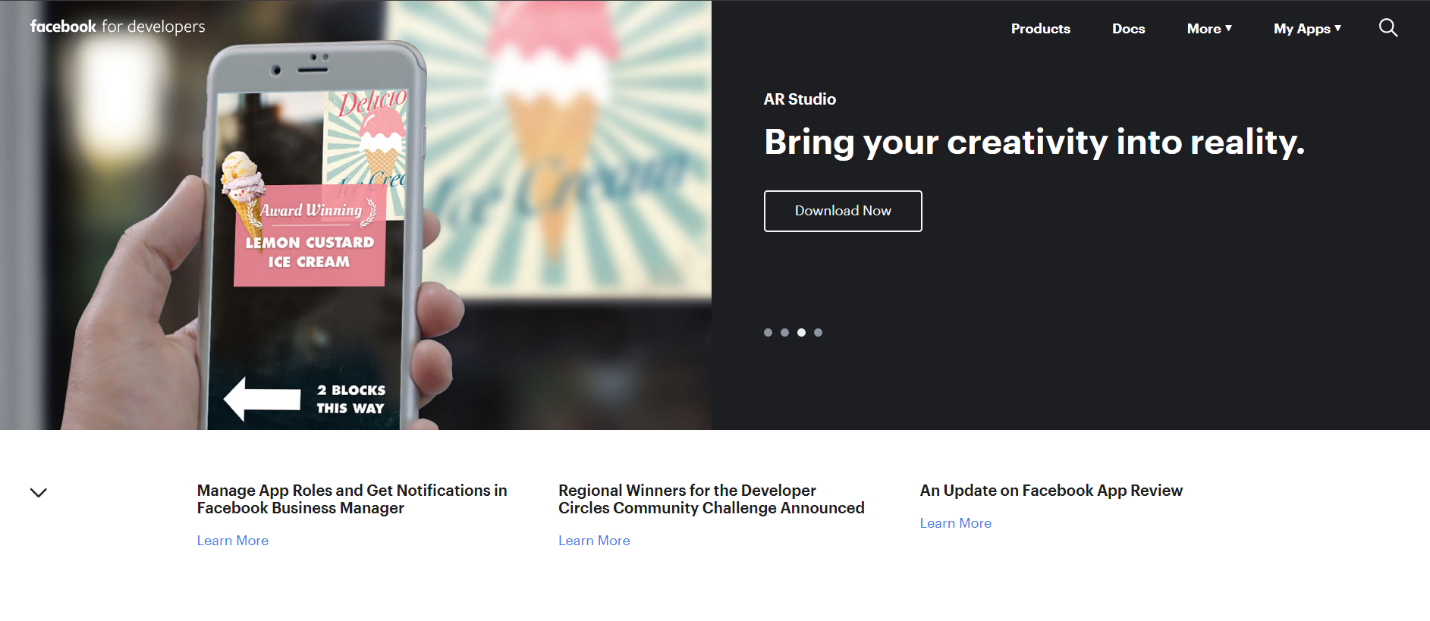
Upon successful client authentication and resource owner credential validation, the authorization server sends the access token to the client

A Client app to retrieve protected user data

I will use a Facebook app as the third party application and Facebook Endpoint as the resource owner to demonstrate what happens underneath the hood.

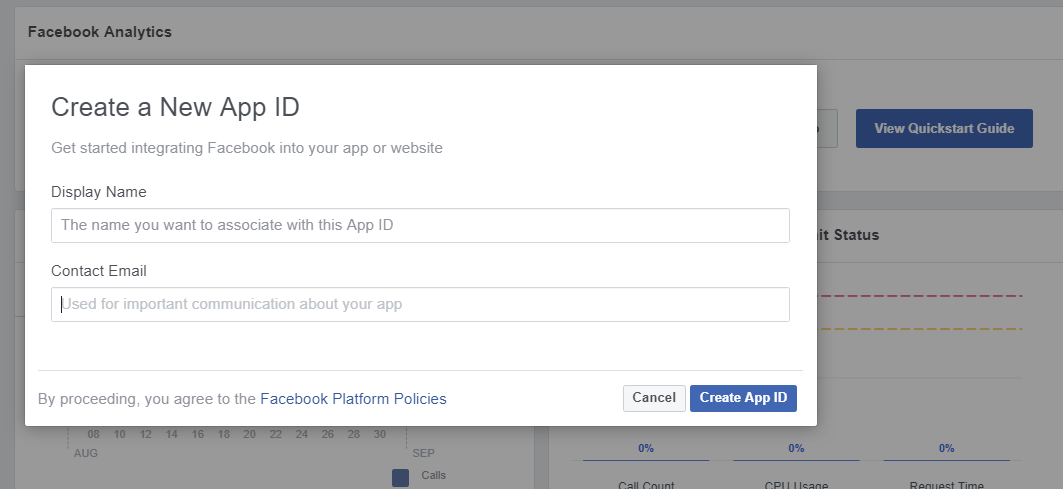
Building project environment

First, you will need to create Facebook Developer account using this link <https://developers.facebook.com/>

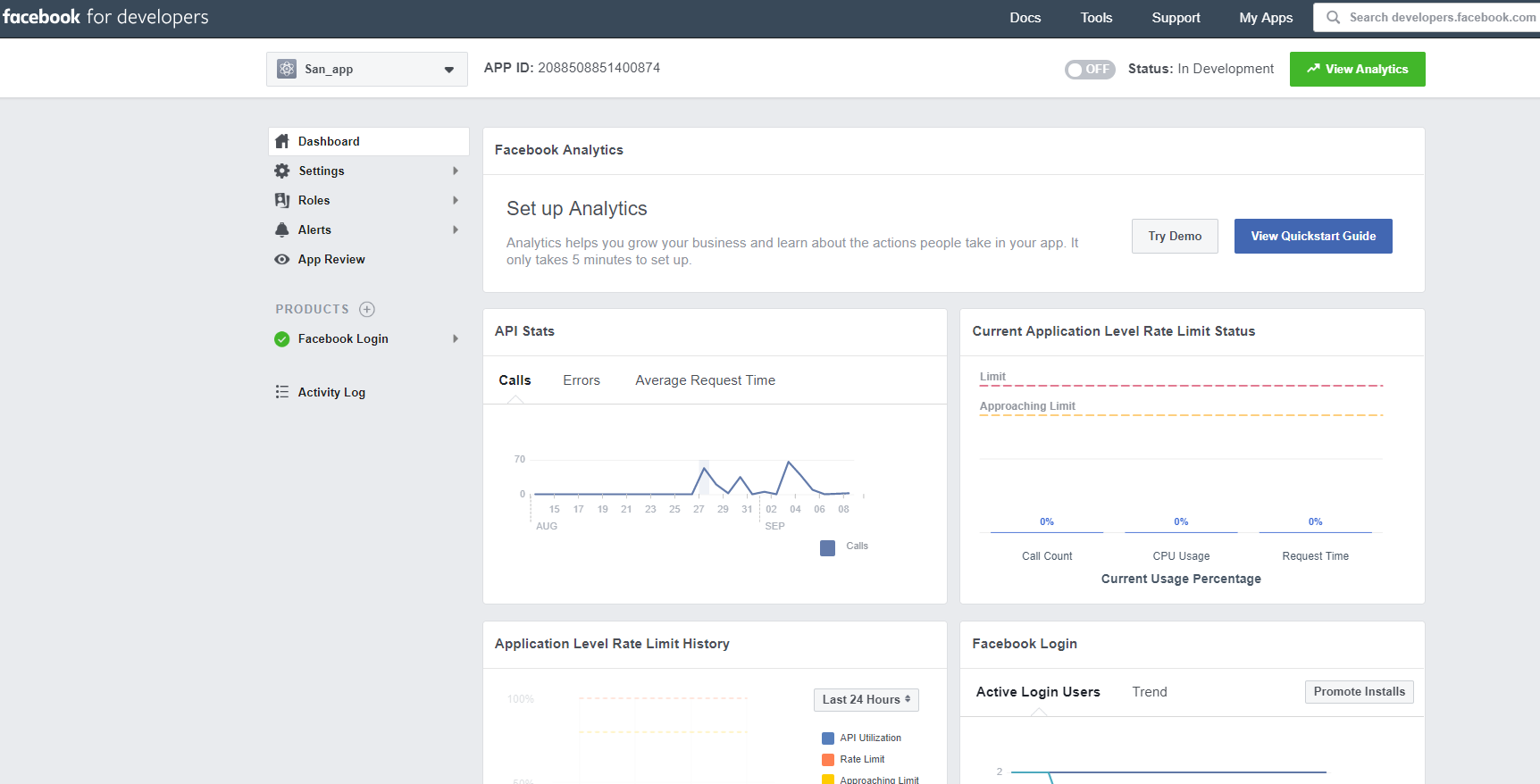


After creating an account, you need to create your first Facebook app.

Click on Add new app. It will prompt you to enter a name for the app and your email address.

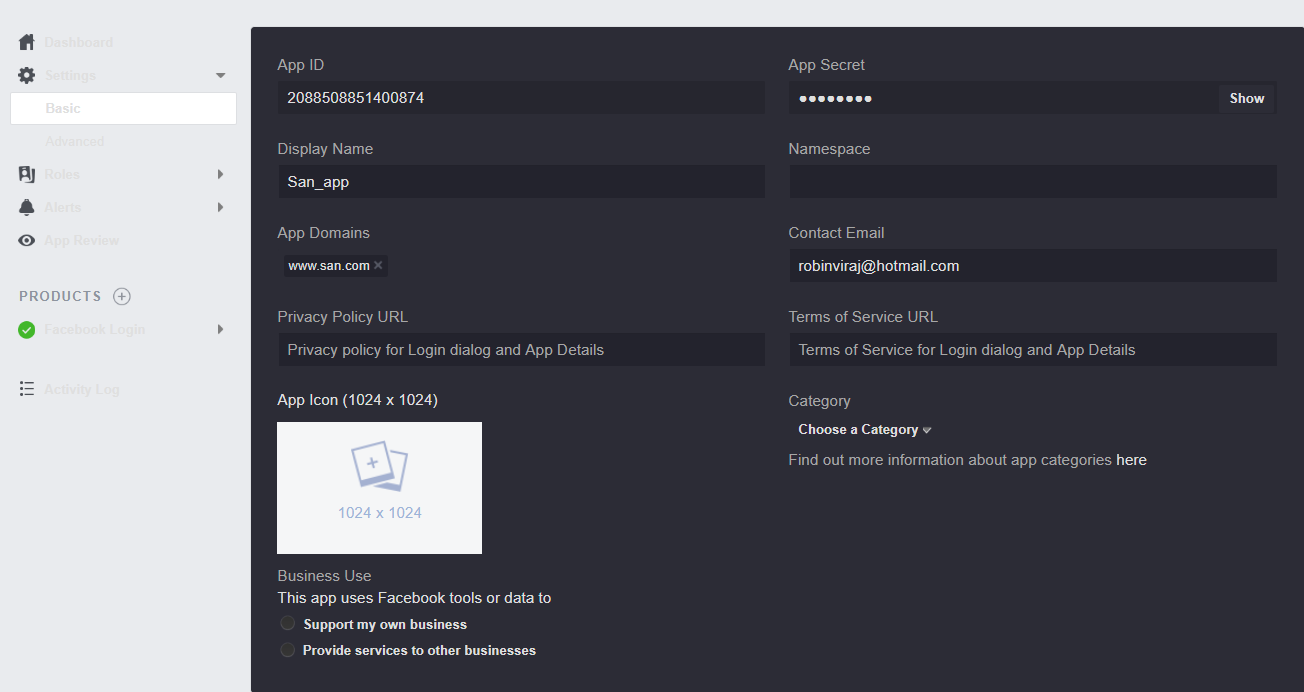


After creating the app, you will be redirected to the Dashboard of your app



There are few attributes of this app that you need to know

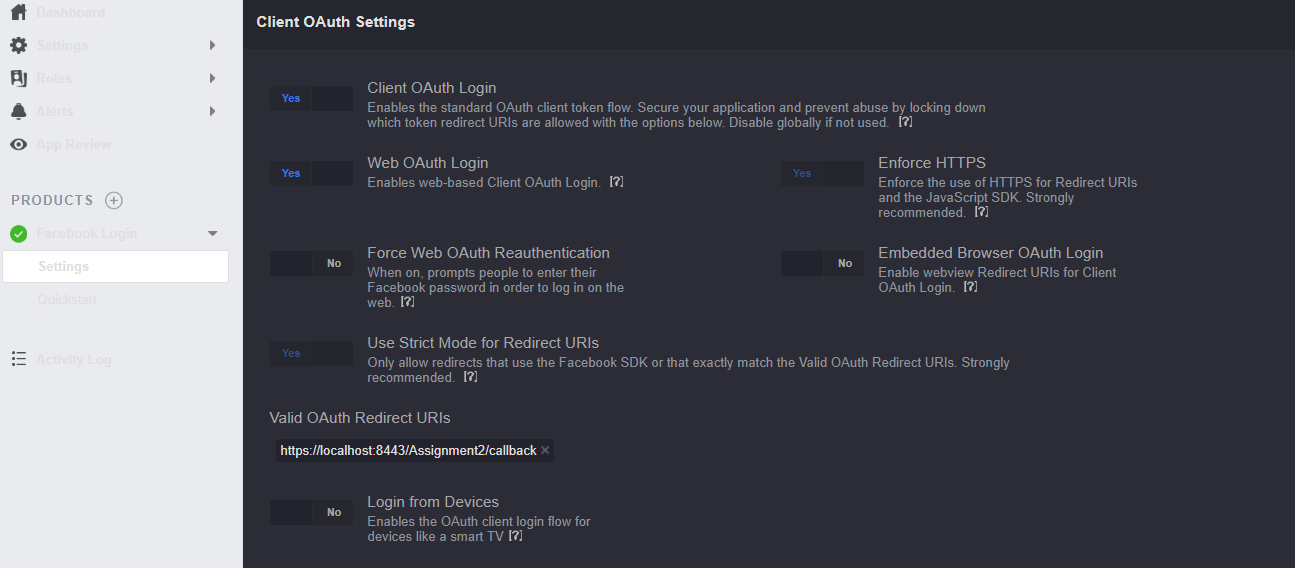
Go to settings tab on the left side panel and click on Basic



In here, you can view your APP ID also known as the Client ID, App secret which is also known as client secret and the Display name of your app

You can customize your app icon by clicking on App Icon and selecting the required image as the icon

Now go to the Facebook login tab, which is under the PRODUCTS category.

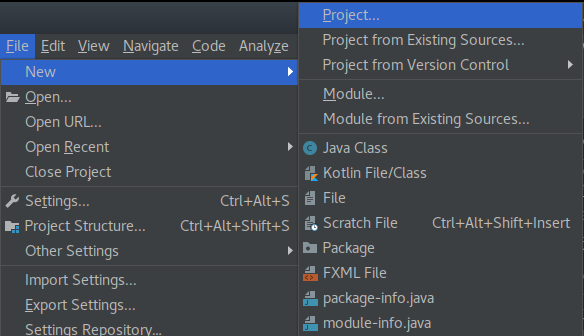


Turn on Client OAuth Login, Web OAuth login.

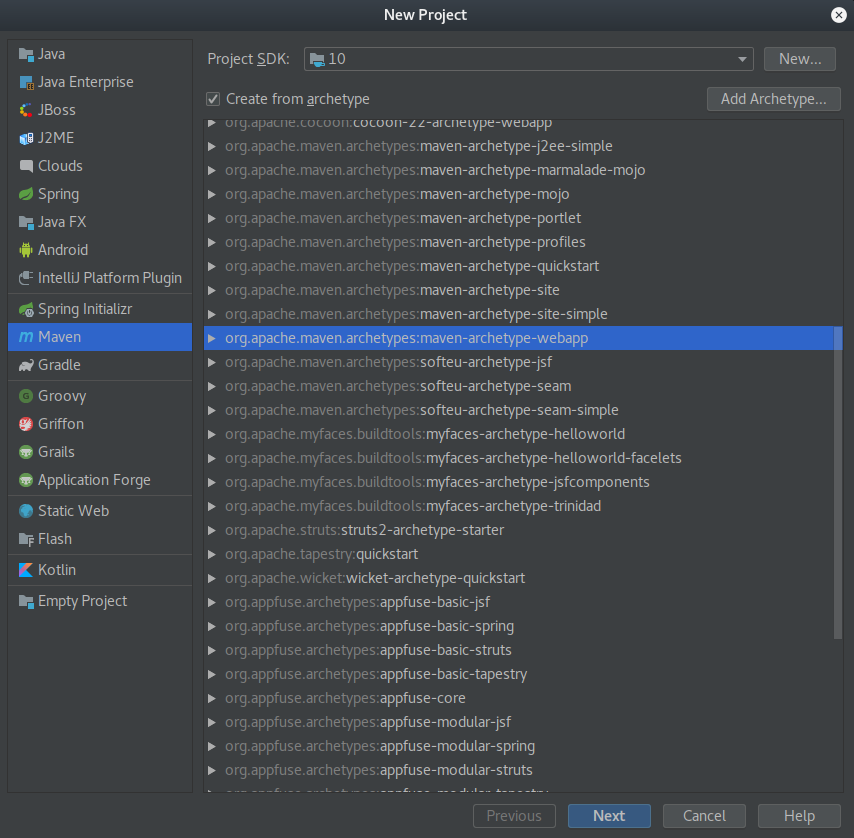
In here, you can see the field called OAuth Redirect URI. You need to provide the url of your web application in this field.

Then open IntelliJ IDEA Ultimate and create a Maven project by following these steps

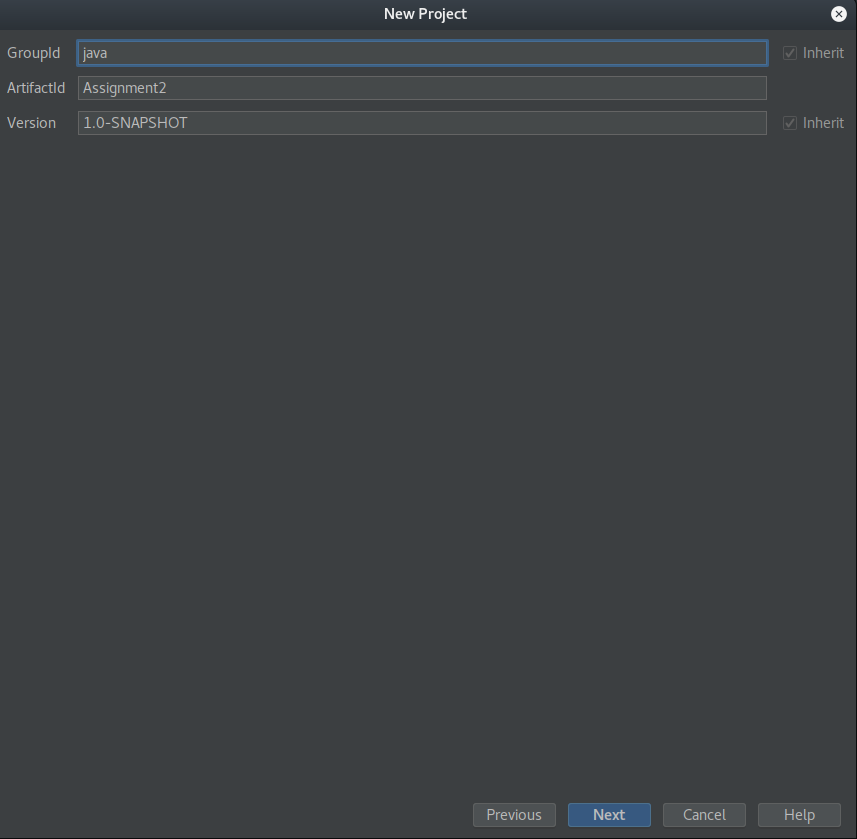
Go to Files 🡪 New 🡪 Project



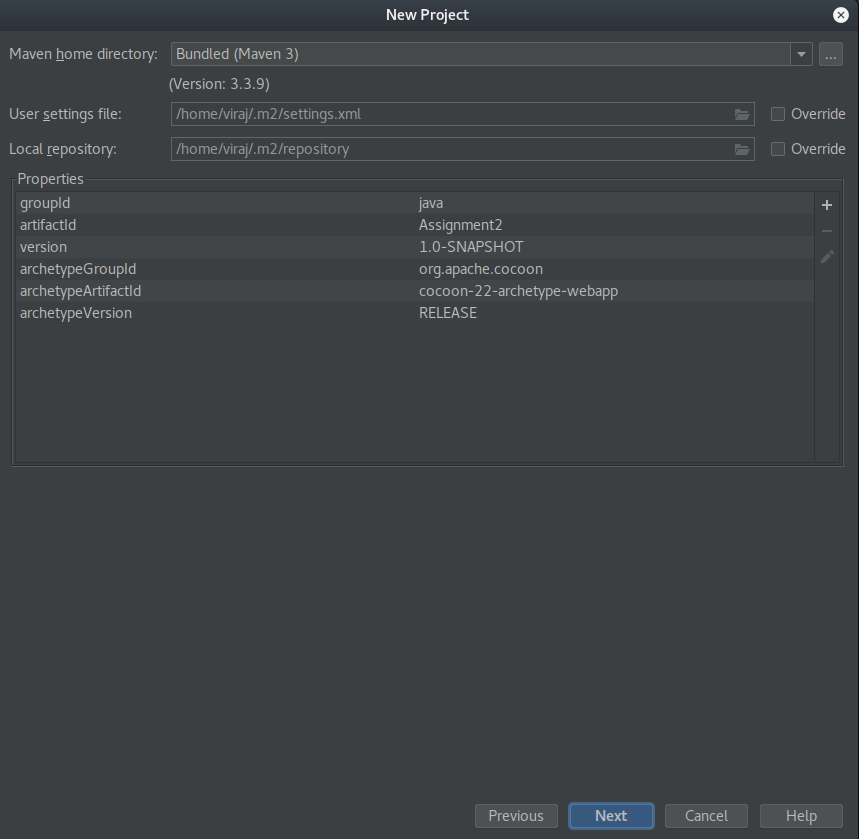
Select Maven from the left hand side panel and then select circled option (archetype) and click Next



Then provide a Group ID and Artifact ID for the project and click Next

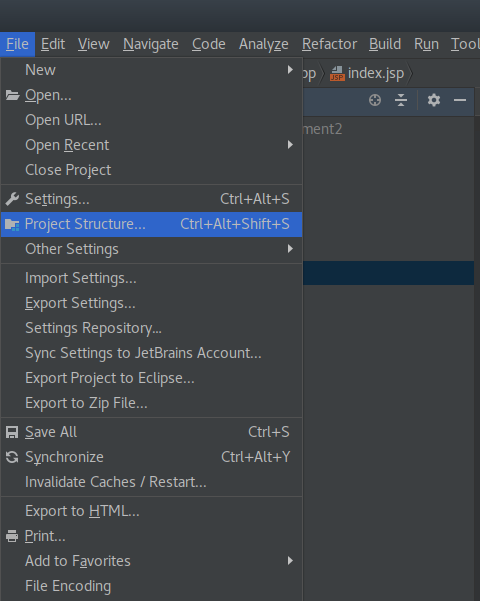


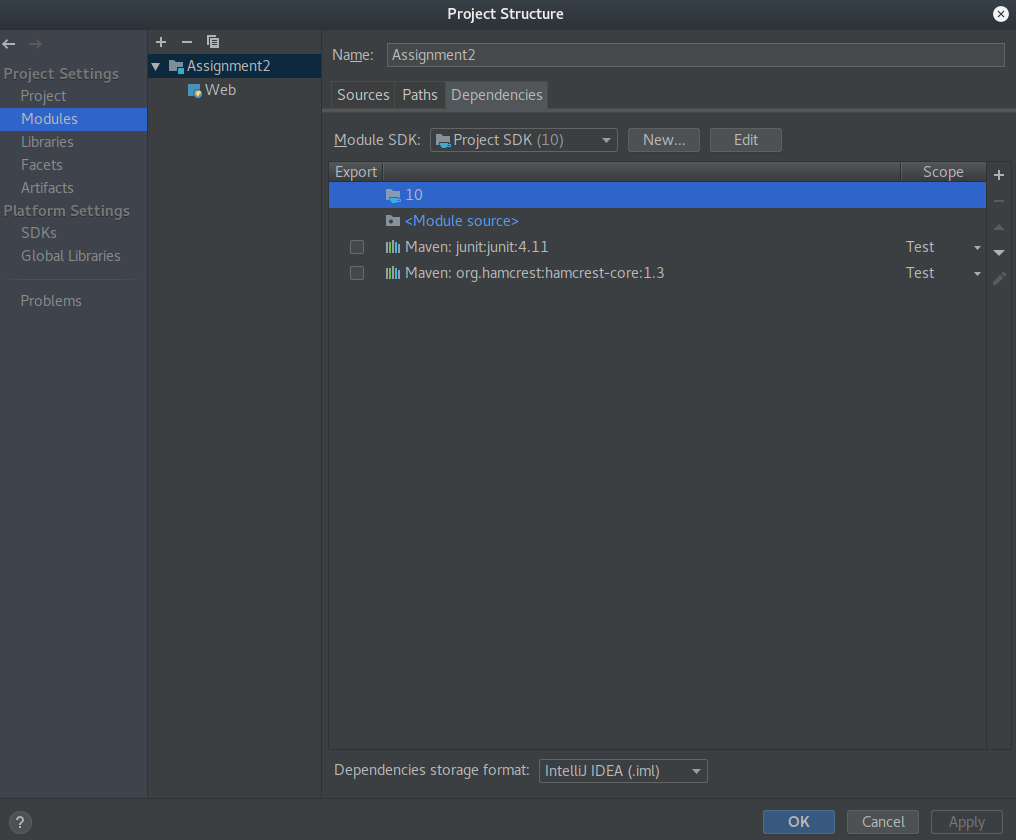
Click Next

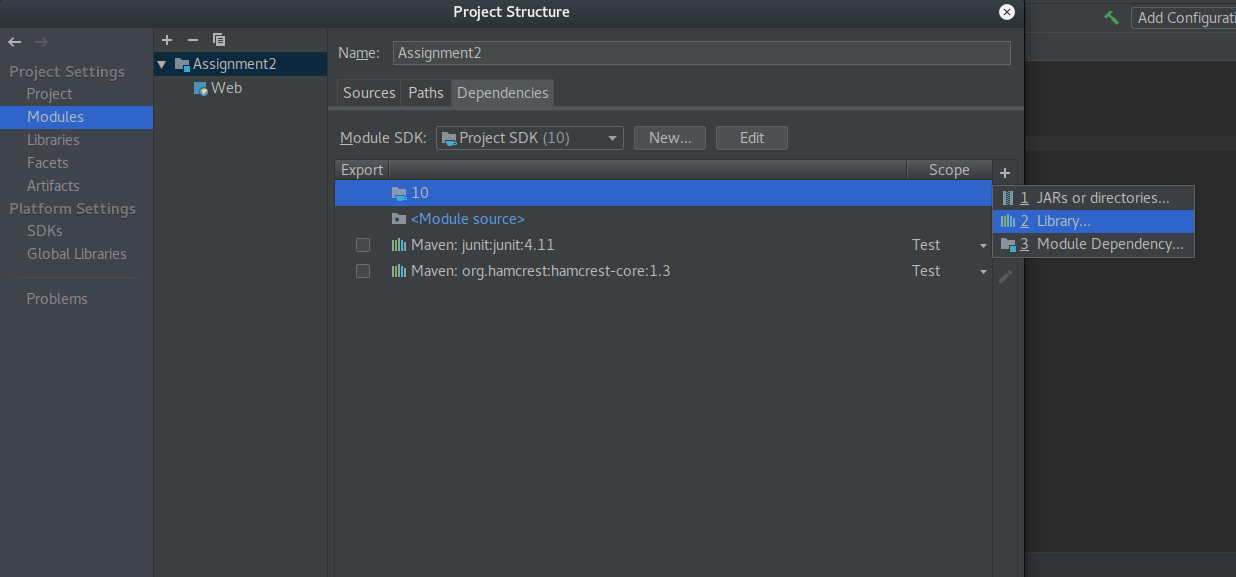


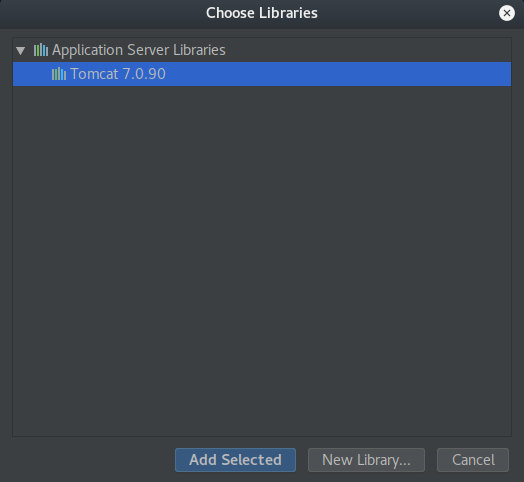
Then wait until the dependencies get download. This could take few minutes.

Then go to Files and select Project Structure from the drop down list



Then select the Modules tab from the left hand side panel

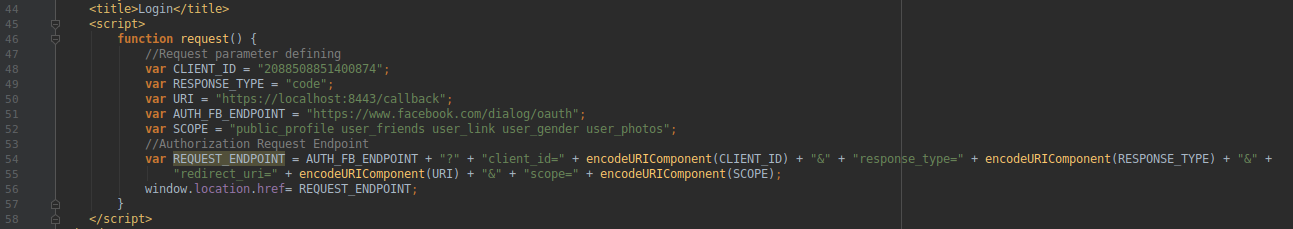
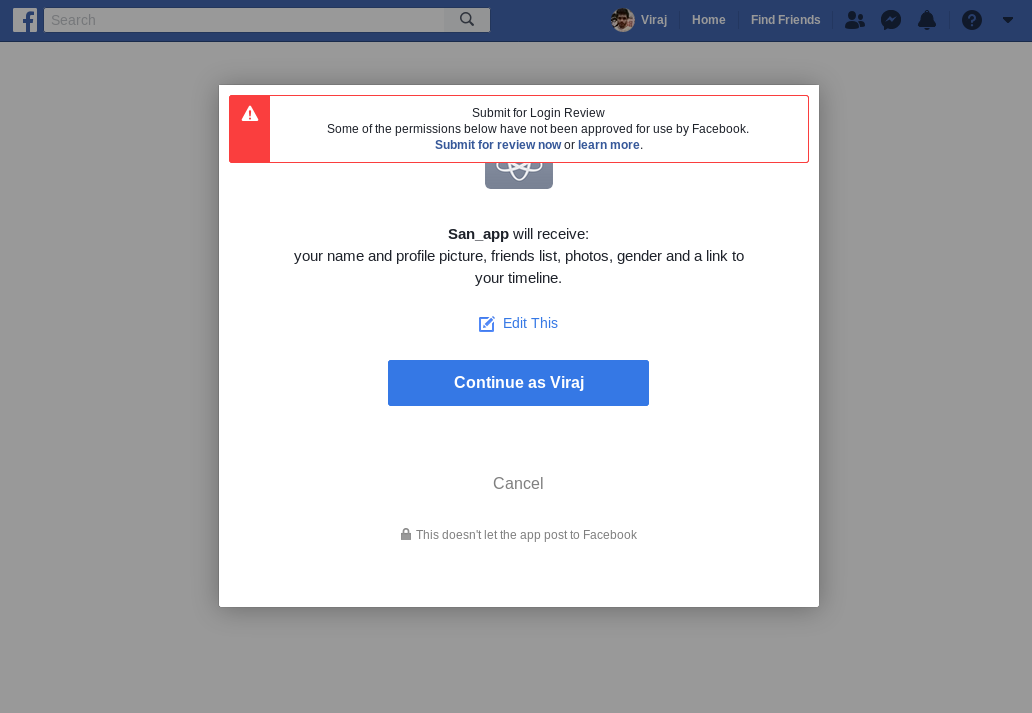
Click on the + mark on the left hand side corner

Click on Library. Then select the Tomcat application server library

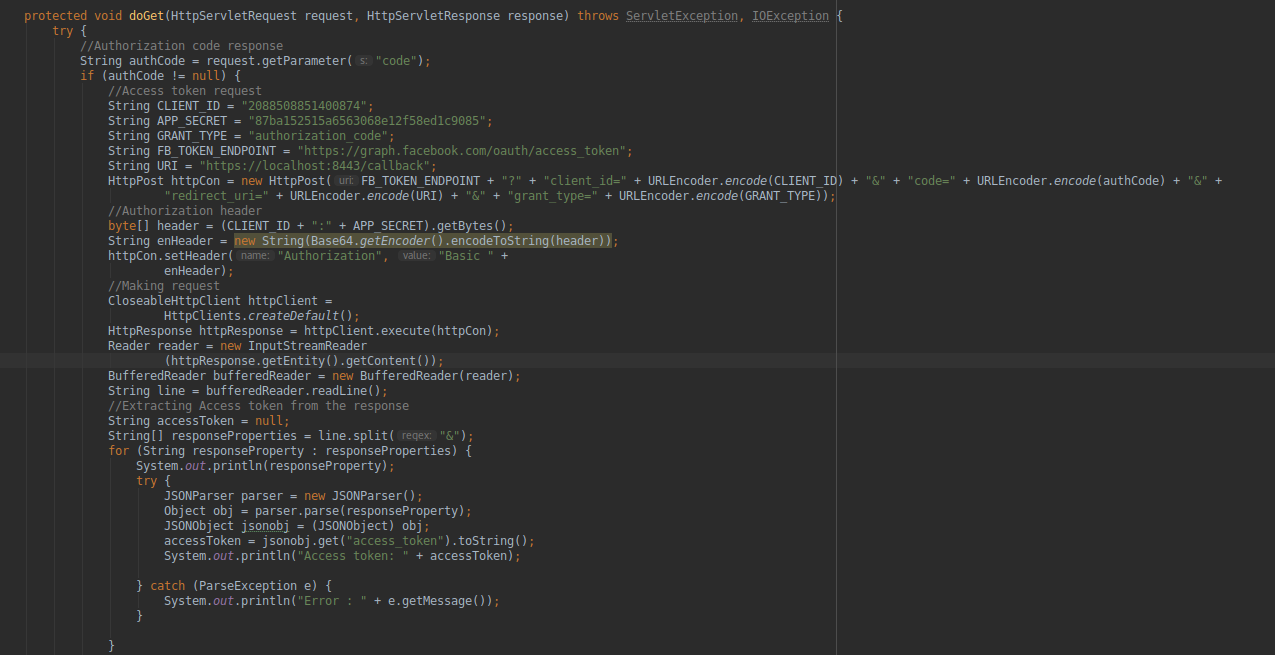
Now you can start the coding part of the project



First create a login jsp page

When user click on login with facebook button, we need to redirect the user to the Request endpoint.

When user Approve it, we need get the code from the query parameter and use it to get the access token. Then we can use that access token to gain access to the user data.

For that create a Java file

