OAuth 2.0: Authorization Code Flow

INTRODUCTION

As general web applications being server-side applications where the source code is not public mostly using the authorization code flow (OAuth 2.0 RFC 6749, section 4.1), which exchanges an authorization code for a token. As mentioned above the application must be server-side since during the exchange the application’s App Secret will transfer to the user account or the requested party where it will store on that end for further use.

HOW IT WORKS

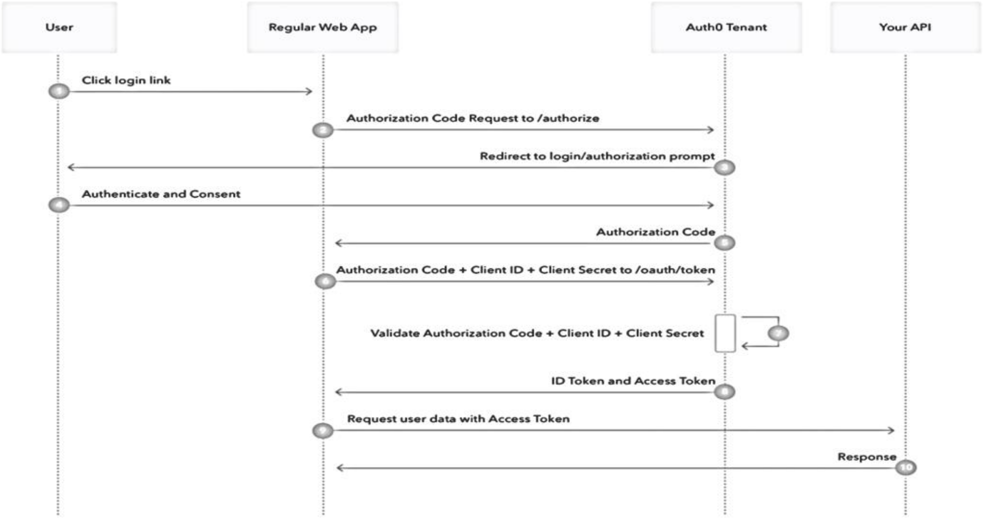

User
Regular Web App
Auth0 Tenant
Your API
Click login link
Authorization Code Request to /authorize
Redirect to login/authorization prompt
Authenticate and Consent
Authorization Code
Authorization Code + Client ID + Client Secret to /oauth/token
Validate Authorization Code + Client ID + Client Secret
ID Token and Access Token
Request user data with Access Token
Response
10

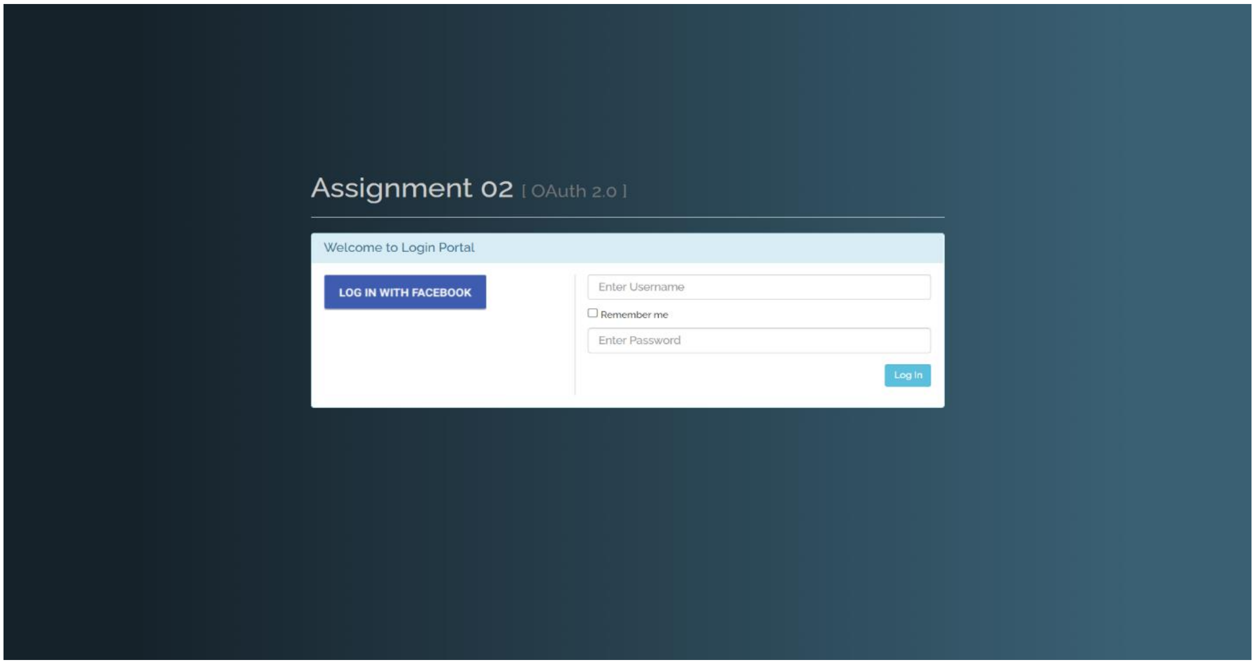
Fig 1. Authorization Code Flow [2]

1. User logs in to the application as usual [Fig. 2].

2. User is redirected to the authorization server (/authorize endpoint) by OAuth's software development kit.

3. User needs to log in resulting in an authorization prompt where it is redirected by the authorization server.

4. User authentication is done by a configured login option and prompts a consent with permissions requested where the web application can use it future when needed.



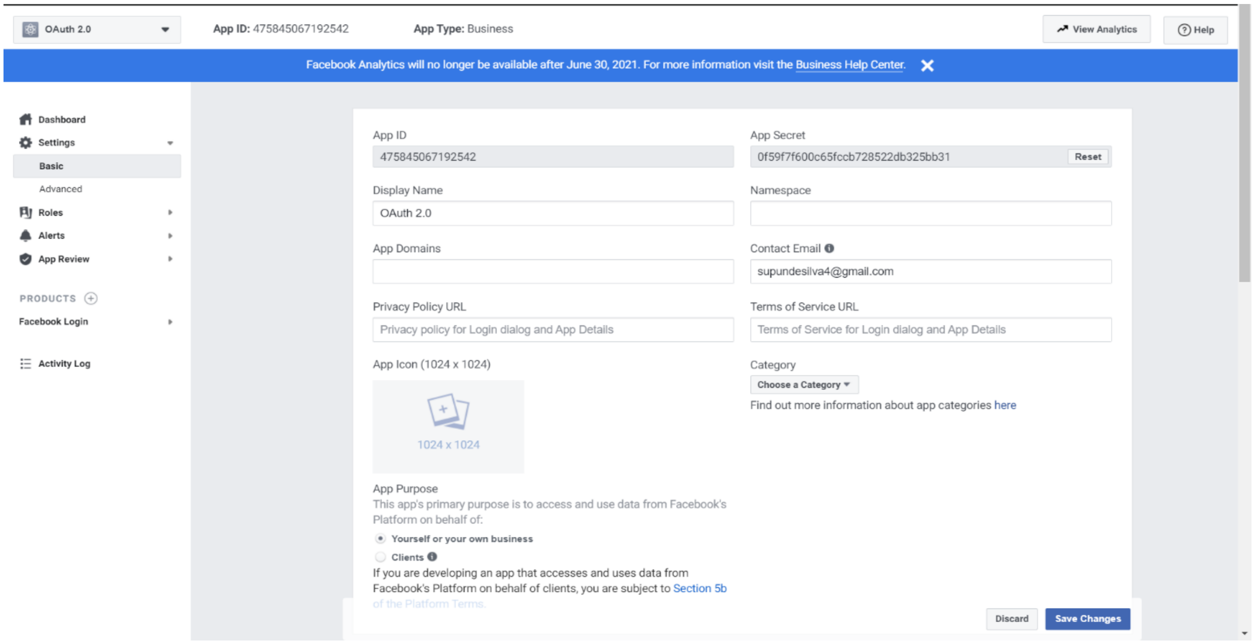
Assignment 02 [OAuth 2.0]

Welcome to Login Portal

Fig 2. User Log in Page

5. Users are redirected back to the web application with an authorization code by the authorization server.

6. Authorization server (/oauth/token endpoint) catches this code when it is sent by the OAuth's software development along with the application's App ID and App Secret [Fig. 3].



App ID: 475845067192542 **App Type:** Business

Facebook Analytics will no longer be available after June 30, 2021. For more information visit the Business Help Center.

App ID: 475845067192542

App Secret: 0f59f7f600c65fccb728522db325bb31

Display Name: OAuth 2.0

Namespace:

App Domains:

Contact Email: supundesilva4@gmail.com

Privacy Policy URL: Privacy policy for Login dialog and App Details

Terms of Service URL: Terms of Service for Login dialog and App Details

App Icon (1024 x 1024)

Category

Find out more information about app categories here

App Purpose

This app's primary purpose is to access and use data from Facebook's Platform on behalf of:

● Yourself or your own business

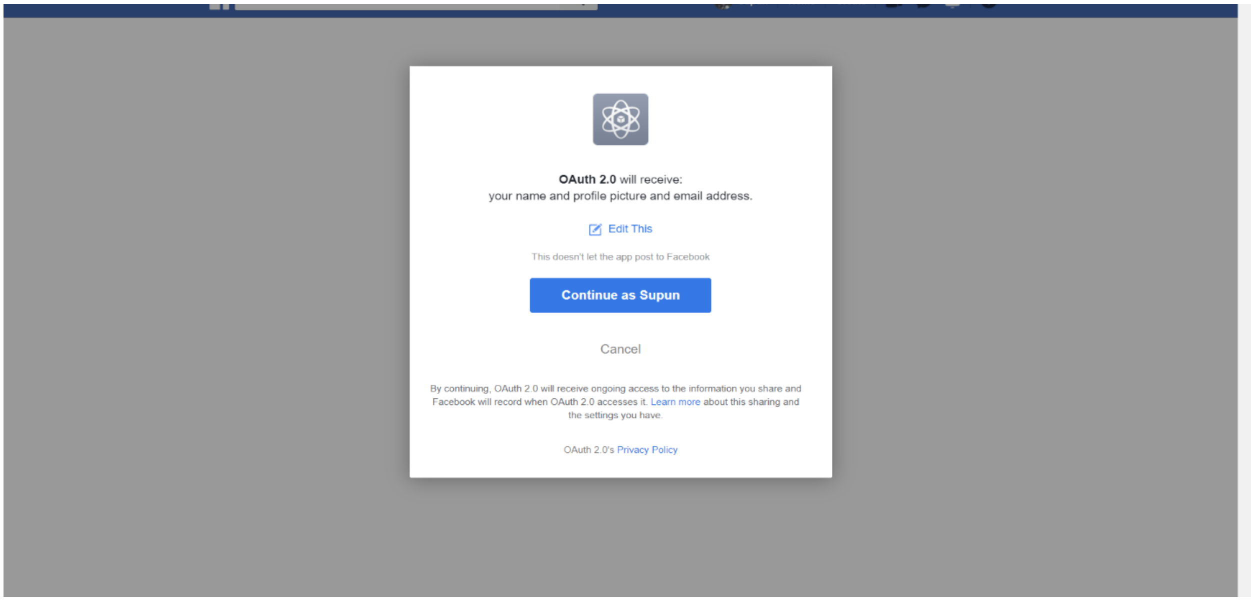
○ Clients

If you are developing an app that accesses and uses data from Facebook's Platform on behalf of clients, you are subject to Section 5b of the Platform Terms.

Fig 3. Application Properties

7. The code, App ID, and App Secret are verified by the authorization server.

8. A response with an ID and access token will be sent by the authorization server.



OAuth 2.0 will receive:
your name and profile picture and email address.

This doesn't let the app post to Facebook

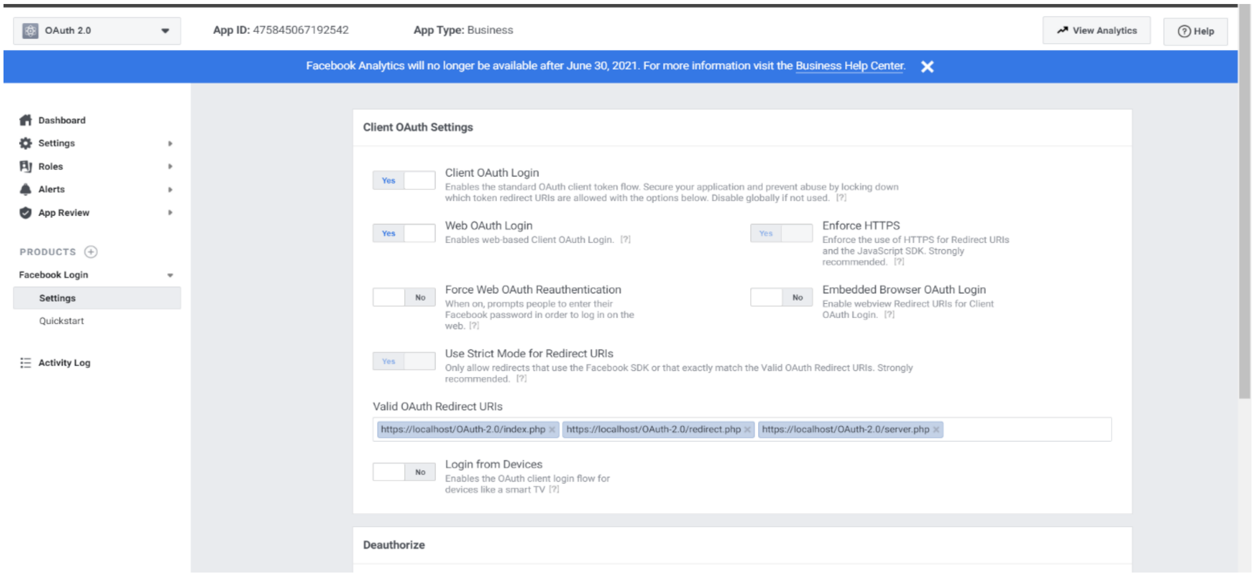
By continuing, OAuth 2.0 will receive ongoing access to the information you share and Facebook will record when OAuth 2.0 accesses it. Learn more about this sharing and the settings you have.

OAuth 2.0's Privacy Policy

Fig 4. User Consent Prompt

9. Your application can use the Access Token to call an API to access information about the user.

10. A response from the application programming interface with requested data.



OAuth 2.0

App ID: 475845067192542

App Type: Business

Facebook Analytics will no longer be available after June 30, 2021. For more information visit the Business Help Center.

Client OAuth Settings

Yes — Client OAuth Login
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. [?]

Yes — Web OAuth Login
Enables web-based Client OAuth Login. [?]

Yes — Enforce HTTPS
Enforce the use of HTTPS for Redirect URIs and the JavaScript SDK. Strongly recommended. [?]

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

No — Embedded Browser OAuth Login
Enable webview Redirect URIs for Client OAuth Login. [?]

Yes — Use Strict Mode for Redirect URIs
Only allow redirects that use the Facebook SDK or that exactly match the Valid OAuth Redirect URIs. Strongly recommended. [?]

Valid OAuth Redirect URIs

https://localhost/OAuth-2.0/index.php https://localhost/OAuth-2.0/redirect.php https://localhost/OAuth-2.0/server.php

No — Login from Devices
Enables the OAuth client login flow for devices like a smart TV [?]

Deauthorize

Fig 5. Application URI Properties

REFERENCES

[1] The OAuth 2.0 Authorization Framework. [Online]. Available: https://tools.ietf.org/html/rfc6749.

[2] Authorization Code Flow. [Online]. Available: https://auth0.com/docs/flows/authorization-code-flow

[3] Permissions Reference. [Online]. Available: https://developers.facebook.com/docs/permissions/reference/

[4] Permissions with Facebook Login. [Online]. Available: https://developers.facebook.com/docs/facebook-login/permissions/overview/

[5] Retrieving User Resources from Facebook over the OAuth 2.0 Authorization Code Grant Type. [Online]. Available: http://www.securityinternal.com/2017/04/retrieving-user-resources-from-facebook.html.