Prafful Agarwal • 2nd
Software Engineer at Google
11h •

Follow

Give me 2 minutes. I'll give you back the 90 minutes of notes I made on how OAuth 2.0 works.

♦ What is OAuth 2.0?

OAuth 2.0 is an open standard for authorization used to provide secure, delegated access to resources without sharing login credentials.

It allows third-party applications to request limited access to user accounts without exposing the user's password.

OAuth 2.0 is widely adopted by platforms like Google, Facebook, GitHub, and many others to grant third-party apps controlled access to their users' data.

♦ Where is OAuth 2.0 Used?

OAuth 2.0 is commonly used in situations like:
- Allowing third-party apps to access social media profiles without needing to log in directly.
- Enabling integrations between software services, such as allowing an app to post content on your behalf on a social platform.
- Granting access to APIs, such as those used by cloud services or financial platforms.

♦ How Does OAuth 2.0 Work?

OAuth 2.0 involves interactions between four primary components: the Client, Authorization Server, Resource Server, and the Resource Owner (the user).

Here's a simplified step-by-step of how it works:

1. Authorization Grant Request

- Client (the app) requests permission from the Resource Owner (user) to access protected resources. This can happen when a user tries to log in or link an account.

- The Authorization Server provides an Authorization Grant to the Client (Step A in the diagram).

2. Access Token & Refresh Token

- The Authorization Grant is validated, and the Client receives an Access Token and, sometimes, a Refresh Token (Step B).

- The Access Token is used to access protected resources on behalf of the user, while the Refresh Token can be used to request a new access token when the old one expires.

3. Accessing Resources

- The Client sends the Access Token to the Resource Server to request access to specific resources (Step C).

- The Resource Server verifies the token and, if valid, allows access (Step D).

4. Token Validation

- If the token is invalid or expired, the Resource Server responds with an Invalid Token Error (Step F).

- The client must then either obtain a new token or refresh the existing one.

5. Token Refresh

- If the Access Token expires, the Client can use the Refresh Token to request a new Access Token from the Authorization Server (Step G).

- This process does not require the user to log in again and extends the session.

6. Renewal and Optional Refresh

- After the refresh, a new Access Token (and sometimes a Refresh Token) is provided for further use (Step H).

TL;DR:

1. Authorization Grant: Client requests permission from the user.

2. Access Token Issued: Authorization server issues access and refresh tokens.

3. Protected Resources Accessed: Client uses the token to access resources.

4. Token Refresh: Tokens are refreshed automatically when expired.

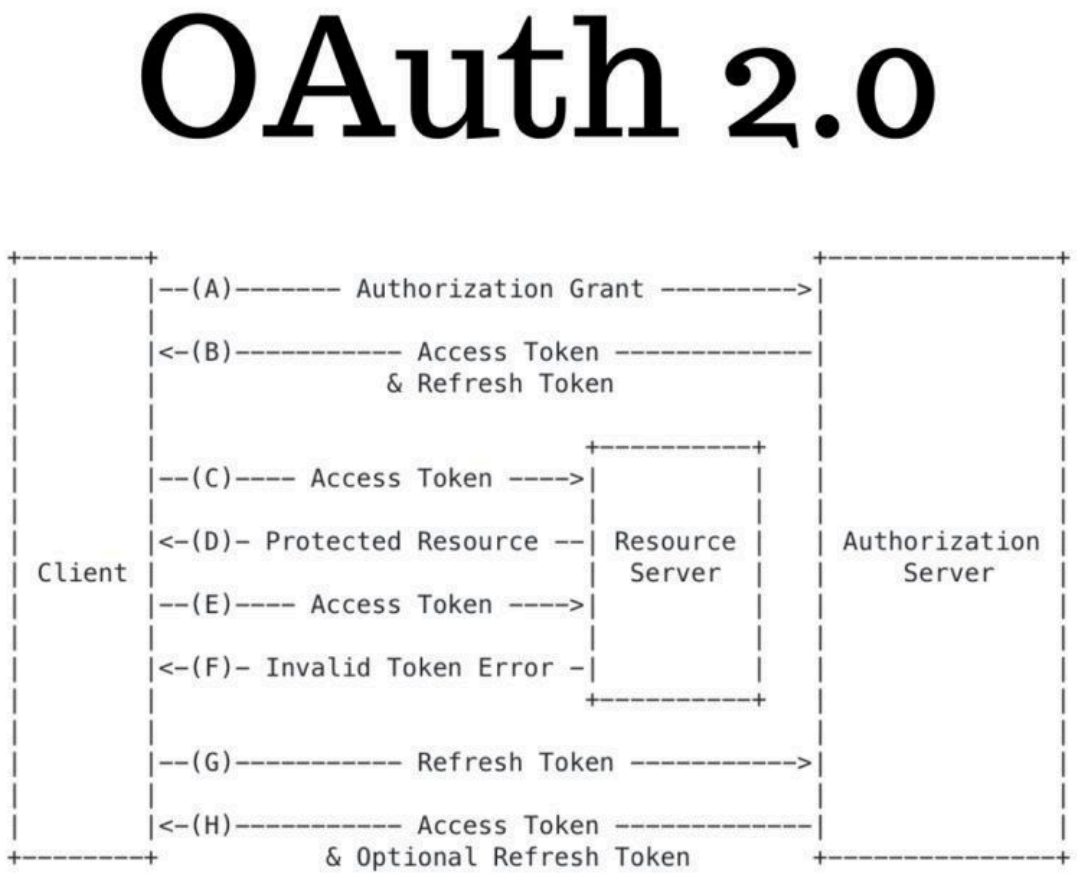

Figure 2: Refreshing an Expired Access Token