What is OAuth 2.0, and how is it different from OpenID Connect?

20 November 2026 · 5 min read · blogs.iictp.org/what-is-oauth-and-openid-connect

OAuth 2.0 is a standard for authorisation: it lets an application get limited access to a resource on a user's behalf, using an access token, without ever seeing the user's password. OpenID Connect is a layer on top of OAuth 2.0 for authentication: it adds an ID token that tells the application who the user is.

TL;DR The short version

- OAuth 2.0 is for authorisation: it gives an application limited access on a user's behalf through an access token, without seeing the password.
- OpenID Connect adds authentication on top, with an ID token that says who the user is.
- New applications should use the authorisation code flow with PKCE, validate every token, and keep access tokens short-lived.

Read online, listen to it, or change the text size: https://blogs.iictp.org/what-is-oauth-and-openid-connect

Who are the parties in OAuth?

The resource owner is usually the user. The client is the application asking for access. The authorisation server signs the user in, asks for consent and issues tokens. The resource server is the API that accepts those tokens. Keeping these roles straight makes most OAuth discussions easier.

Which flow should a new application use?

For almost every application, including browser and mobile apps, use the authorisation code flow with PKCE, defined in RFC 7636. The OAuth 2.0 security best current practice, RFC 9700, says not to use the older implicit flow or the password grant. For one service calling another with no user involved, use the client credentials flow.

What is the difference between an access token and an ID token?

An access token is for the API: the client sends it with requests and should not need to read it. An ID token, from OpenID Connect, is for the client: it states who signed in, for which application and when. Do not send ID tokens to APIs as if they were access tokens.

What are scopes?

Scopes name the access the client is asking for, such as reading a calendar but not changing it. Ask for the narrowest scopes that do the job, which is least privilege applied to tokens.

What mistakes are most common?

Storing tokens where page scripts can read them, accepting tokens without checking the signature, issuer, audience and expiry, using long-lived access tokens instead of short ones with refresh tokens, and allowing loosely matched redirect URIs. Using a well-maintained library and an established identity provider avoids most of these.

Related credential: Certified Technical Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. IETF. RFC 6749, The OAuth 2.0 Authorization Framework. 2012.
2. IETF. RFC 7636, Proof Key for Code Exchange (PKCE). 2015.
3. IETF. RFC 9700, Best Current Practice for OAuth 2.0 Security. 2025.
4. OpenID Foundation. OpenID Connect Core 1.0. openid.net.

Published by IICTP, an independent technical credential body, on 20 November 2026. The current version of this post is at https://blogs.iictp.org/what-is-oauth-and-openid-connect. More guides: blogs.iictp.org.