

INTRODUCTION TO IDENTITY

What is OpenID Connect (OIDC)?

OpenID Connect or OIDC is an identity protocol that utilizes the authorization and authentication mechanisms of OAuth 2.0. The OIDC final specification was published on February 26, 2014, and is now widely adopted by many identity providers on the Internet.

OIDC was developed by the [OpenID Foundation](#), which includes companies like Google and Microsoft. While [OAuth 2.0 is an authorization protocol](#), OIDC is an identity authentication protocol and may be used to verify the identity of a user to a client service, also called **Relying Party**. In addition, users' claims like, for example, name, email address, etc., may also be shared on request.

A wide variety of clients may use [OpenID Connect](#) (OIDC) to identify users, from single-page applications (SPA) to native and mobile apps. It may also be used for Single Sign-On (SSO) across applications. OIDC uses **JSON**



Web Tokens (JWT), HTTP flows and avoids sharing user credentials with services.

OpenID Connect has consent built-in. This is important as OIDC is often used in consumer-facing services (e.g., a Relying Party), where the sharing of personal data requires the user's explicit consent.

These features, along with the simplicity of implementation, make OpenID Connect a useful protocol when a user's identity is required and a powerful alternative to the more complex SAML 2.0. It is also especially suitable for mobile apps.

How Does OpenID Connect Fit with OAuth2?

OIDC utilizes OAuth 2.0 as an underlying protocol. The principal extensions are a special scope value ("openid"), the use of an extra token (the ID Token, which encapsulates the identity claims in JSON format), and the emphasis on authentication rather than authorization. Also, in OIDC, the term "flow" is used in place of OAuth2 "grant"

Principles and Definitions in OpenID Connect

The **OIDC provider** (generally called the OpenID Provider or Identity Provider or **IdP**) performs user authentication, user consent, and token issuance. The client or service requesting a user's identity is normally called the **Relying Party (RP)**. It can be, for example, a web application, but also a JavaScript application or a mobile app.

Being built on top of OAuth 2.0, OpenID Connect uses tokens to provide a simple identity layer integrated with




ID Token: Specific to OIDC, the primary use of this token in JWT format is to provide information about the authentication operation's outcome. Upon request, it may provide the identity data describing a user profile. The data about the authentication result and the user profile information are called **claims**. The user profile claims may be any data that is pertinent to the Relying Party for identification purposes, such as a persistent ID, email address, name, etc.

Access Token: Defined in OAuth2, this (optional) short lifetime token provides access to specific user resources as defined in the scope values in the request to the authorization server.

Refresh Token: Coming from OAuth2 specs, this token is usually long-lived and may be used to obtain new access tokens.

ID Tokens should be digitally signed to prevent tampering. They may also be encrypted to provide additional privacy, although, in many cases, transport layer security (HTTPS) is sufficient. For SPAs and mobile apps, ID Token encryption is not useful, as the decryption key can be discovered easily.

OIDC Flows

The choice of OpenID Connect flow depends on the type of application and its security requirements.  are three common flows:

- **Implicit Flow:** In this flow, commonly used by SPAs, tokens are returned directly to the RP in a redirect URI.
- **Authorization Code Flow:** This flow is more secure than Implicit, as tokens are not returned directly. For native/mobile apps and SPA, security may be enhanced by using **Proof Key for Code Exchange**.
- **Hybrid Flow:** Combining Implicit and Authorization Code flows, here, the ID Token is returned directly to the RP, but the access token is not. Instead, an authorization code is returned that is exchanged for an access token.

What Can an Identity Provider Use to Authenticate Users Using OIDC?

The OpenID Provider determines the authentication methods available to authenticate users when they sign in to their IdP account and possibly consent to release their identity data to the RP. OIDC specs say nothing about the mechanics of user authentication itself. The IdP can offer single or multiple factors e.g.

- Username/password
- Single-use code delivered out of band, e.g., by SMS or email
- Code generated by App (OATH, TOTP or HOTP)
- Biometric via App
- Federated (e.g., Facebook, GoogleID)



The RP can also have a say on IdP authentication. For example, [multifactor authentication](#) may be requested. However, the way an IdP authenticates users is out of the scope of OIDC.

Want to learn more?

Keep reading at our [Intro to IAM](#) page to explore more topics around Identity and Access Management.

QUICK ASSESSMENT

How are OAuth 2 and OpenID Connect related?

- ☐ They both have consent built-in
- ☐ They can both be used for Single Sign-On (SSO)
- ☐ They both use JWT
- ☐ OpenID Connect is based on OAuth2

SUBMIT



QUICK ASSESSMENT

What is the best OIDC flow to use with a mobile app?

- ☐ Implicit
- ☐ Authorization Code with PKCE
- ☐ Authorization Code
- ☐ Hybrid

Enter your email to take the full assessment

Work email



We'd also like to reach out when we have something worth sharing. You can always change your mind and [opt out at any time.](#)

☐ Email me product updates, resources, surveys, offers, and event info.

☐ No, thank you. I'll just sign up.

START ASSESSMENT

**Secure access for
everyone. But not just
anyone.**

TRY AUTH0 FOR FREE

TALK TO SALES



Features



Pricing

Solution Types



Support



Developers



Company



Offices

We're remote friendly, with office locations around the world: [Seattle](#), [London](#), [Buenos Aires](#), [Sydney](#), [Singapore](#) and [Tokyo](#).

Contact Us:

10800 NE 8th St, Suite 700

Bellevue, WA 98004

+1 (888) 235-2699





English



[Status](#)

[Legal](#)

[Privacy](#)

[Terms](#)



© 2013 - 2021 Auth0® Inc. All Rights Reserved.

