

Connecting an Application Using OIDC

Overview

This guide explains how to connect an OpenID Connect-compatible application to ID Everywhere for centralized user authentication.

Before beginning, confirm that your application supports **OpenID Connect (OIDC)** and allows configuration of a custom identity provider.

Prerequisites

You will need:

- Administrative access to your ID Everywhere tenant.
- Permission to configure authentication in the application you're connecting.
- The application's authorized redirect or callback URL.
- A user account authorized to sign in through ID Everywhere.

Step 1 — Open OIDC Applications

Sign in to your ID Everywhere administration portal:

<https://app.ideverywhere.com>

Navigate to **OIDC Applications** in your tenant administration interface.

Choose the option to add a new application.

Step 2 — Enter the application details

Provide a recognizable application name.

For example:

- Customer Portal
- Employee Dashboard
- Internal CRM
- Support Application

Choose a name that helps administrators identify the integration later.

Step 3 — Select the application type

Choose the client type based on the application's architecture.

Public Application (PKCE)

Use for browser-only, desktop, and mobile applications that cannot securely store a client secret.

Web / Server Application (Confidential)

Use for applications with a trusted backend that securely stores a client secret and performs the token exchange.

See *Choosing the Right OIDC Application Type* for additional guidance.

Step 4 — Configure authorized redirect URLs

The redirect URL, sometimes called a callback URL, tells ID Everywhere where to send the browser after authentication.

Obtain this URL from the application's OIDC configuration instructions.

For example:

```
https://portal.example.com/auth/callback
```

Register the exact URL in ID Everywhere.

The redirect URL used during sign-in must match an authorized redirect URL registered for the client.

Pay attention to:

- `https://` versus `http://`
- Domain and subdomain
- URL path
- Trailing slash
- Port number, when applicable
- Query parameters, when applicable

Do not guess the callback URL. Use the value provided by the application.

For production web applications, use HTTPS.

Step 5 — Configure scopes

Scopes tell ID Everywhere what categories of information the application requests.

ID Everywhere advertises these scopes:

Scope	Purpose
<code>openid</code>	Required for OIDC authentication
<code>profile</code>	Requests standard profile information
<code>email</code>	Requests email-related identity information
<code>groups</code>	Requests group-related information where available

A common starting configuration is:

```
openid profile email
```

Always include `openid` when performing OIDC authentication.

Request only the scopes the application needs.

Step 6 — Save the application

After creating the application, ID Everywhere provides its connection information.

Record the **Client ID**.

For a confidential application, securely save the **Client Secret** when it is provided.

Do not place client secrets in public documentation, support tickets, or client-side code.

Step 7 — Configure the third-party application

Open the third-party application's authentication settings.

Where supported, use ID Everywhere's OIDC discovery URL:

```
https://app.ideverywhere.com/.well-known/openid-configuration
```

This discovery document publishes the provider's endpoints and supported OIDC capabilities.

If the application requires individual endpoint URLs, use:

Setting	URL
Issuer	<code>https://app.ideverywhere.com</code>
Authorization Endpoint	<code>https://app.ideverywhere.com/oauth2/authorize</code>
Token Endpoint	<code>https://app.ideverywhere.com/oauth2/token</code>
UserInfo Endpoint	<code>https://app.ideverywhere.com/oauth2/userinfo</code>
JWKS Endpoint	<code>https://app.ideverywhere.com/.well-known/jwks.json</code>
Discovery Document	<code>https://app.ideverywhere.com/.well-known/openid-configuration</code>

Enter the Client ID and, for confidential clients, the Client Secret.

Select **Authorization Code** as the grant type.

For public clients, enable **PKCE** with **S256**.

For confidential clients, use the client authentication method supported by the application and ID Everywhere, such as **Client Secret Basic**.

Step 8 — Test the connection

Start a new sign-in attempt from the connected application.

A successful flow generally looks like this:

1. The application redirects your browser to ID Everywhere.
2. You authenticate using your ID Everywhere credentials.
3. ID Everywhere returns your browser to the registered callback URL.
4. The application completes the token exchange.
5. The application validates the identity response.
6. You are signed in to the connected application.

The application must validate the ID token rather than simply trusting unverified token contents.

Step 9 — Verify user information

Depending on the requested scopes and the application's configuration, ID Everywhere may provide information such as:

- Unique user identifier (`sub`)
- Name
- Email address
- Tenant identifier
- Application role

Not every claim is guaranteed in every response. Applications should request appropriate scopes and handle optional claims correctly.

Security best practices

- Use a separate OIDC registration for each distinct application.
- Restrict redirect URLs to known, trusted destinations.
- Protect confidential client secrets.
- Use HTTPS.
- Request only necessary scopes.

- Validate token signatures, issuer, audience, and expiration.
- Review account access when users are disabled or removed.
- Rotate credentials if they may have been compromised.

Frequently asked questions

Can I use the same Client ID for several applications?

Separate registrations are recommended because each application may have different redirect URLs, credentials, and security requirements.

Can I change the redirect URL later?

Update the authorized redirect URLs in ID Everywhere when the connected application's callback URL changes.

What if the application asks for a SAML metadata URL?

That application is requesting SAML configuration, not OIDC. The current ID Everywhere OIDC integration cannot be substituted for SAML metadata.

What if the application asks for an OAuth 2.0 Client Credentials grant?

The current documented IDE OIDC integration uses Authorization Code. Client Credentials is a different grant and is not currently supported by this integration.

Related articles

- [Choosing the Right OIDC Application Type](#)
- [Understanding OIDC Settings and Security Terms](#)

- Troubleshooting OIDC Connections
-

Revision #1

Created 2026-10-10 22:11:01 UTC by Administrator

Updated 2026-10-10 22:11:20 UTC by Administrator