Recast

OAuth 2.0

Last Modified on 04.22.26

OAuth 2.0 / OpenID Connect are industry standards for exchanging identification and authorization data between trusted parties. Application Workspace implements an OAuth authorization server that also supports the OpenID Connect standards.

OIDC limitations

- Application Workspace supports OpenID Connect but does not implement the OIDC discovery endpoint.
- Administrators must configure relying-party components with the JWKS endpoint provided by Zone.
- Token validation assumptions based on discovery metadata do not apply, because there is no discovery endpoint.
- The typ numeric claim on the Token is Application Workspace's own discriminator, not the standard OIDC typ claim.

Settings screen

Allow requesting metadata – If enabled, external parties can receive the metadata of this identity provider. In most cases, it's recommended to allow this so that federations can be created quick and easily. This can always be disabled after setting up a federation.
Metadata URL – The URL where the metadata is published. Note: even if metadata queries are not allowed, an administrator can always download the metadata by using the download button. (.well-known/openid-configuration is also supported)

Clients screen

Clients must be defined before they can make use of this authorization server.

Create client dialog box

Name – The name of the client, visible for users when they log out.
Client ID – The OAuth 2.0 Client ID used to identify the Client with the Authorization Server.
Support OpenID Connect – If enabled, the *OpenID Connect* scope is supported by default for this client. You do not have to add it in the **Supported scopes** tab of the **Edit client dialog box**.

Edit client dialog box

Secrets tab

Client can use a secret for trusted requests to the authorization server. Keep this value safe as only the Client and Authorization Server knows it.
If the **Require secret for all token endpoint requests** checkbox is selected,
then every grant type (AuthorizationCode, RefreshToken and Password) on the token endpoint should provide a client secret.

Redirect urls tab

After a client requested an authorization code, the Application Workspace Server needs to send back the user. The redirect URLs option ensures that authorization codes are not shared with unwanted parties.

Example: `https://oauth-client.liquit.com/callback`

Supported scopes tab

A list of scopes that the client has access to.

Logout tab

Most OAuth 2.0 implementation do not support logout. You can still logout of a client by specifying a normal logout web page.

Logout URL – Insert the client logout URL. If the logout page supports redirection after logging out, the client logout URL should incorporate the variable "${slo.return.url". This variable will generate a URL that directs back to the Application Workspace, enabling the logout status to be reported to the logout page.
Logout reporting – If enabled, the logout page will wait for the status report. Otherwise the logout page assumes a successful logout after a few seconds.
Embedded logout supported – Enable this option if the client supports being logged out within an HTML iframe. If there are problems with logging out, we recommend you disable this option.

Access conditions tab

With access control you can limit which users can make use of this client. When the access is denied for a user, the user will be redirected to the requesting web application where an access denied error message will be displayed.

The filters available are the same as those for contexts.

Advanced tab

Access token section
Access token lifetime – The length of time an access token is valid. The default value is 1 hour.
Allow implicit Access Token – If enabled, the client can request an access token from the authorization endpoint and skip the authorization code part of the flow.

Refresh token section
Refresh token lifetime – The length of time an access token is valid. The default value is 90 days.

ID token section
ID token lifetime – The length of time an ID token is valid. The default value is 10 hours.
Allow implicit ID Token – If enabled, the client can request an ID token from the authorization endpoint and skip the authorization code part of the flow.

Code section
Code lifetime – The length of time an authorization code is valid. The default value is 1 minute.
PKCE field – PKCE (RFC 7636) is an extension to the Authorization Code flow to prevent certain attacks and to be able to securely perform the OAuth exchange from public clients. The following options are available:

- *Never require PKCE* – A client never has to provide a code challenge.
- *Always require PKCE, allow plain challenge* – A client always has to provide a code challenge which doesn't need to be hashed.
- *Always require PKCE, require hashed challenge* – A client always has to provide a code challenge which needs to be hashed.
- *Only require PKCE when no secret is provided, allow plain challenge* – A client must provide a code challenge when not providing a secret. The code challenge doesn't need to be hashed.
- *Only require PKCE when no secret is provided, require hashed challenge* – A client must provide a code challenge when not providing a secret. The code challenge needs to be hashed.

Recast