

Advanced Topics



Kevin Dockx

ARCHITECT

@KevinDockx <https://www.kevindockx.com>



Coming Up



Token Lifetimes and Expiration

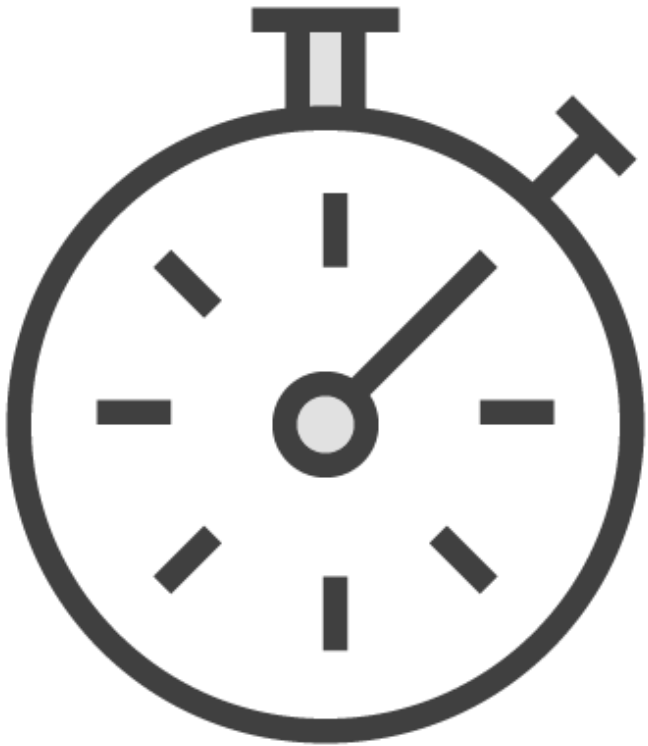
Gaining Long-Lived Access with Refresh Tokens

Reference Tokens and Revocation

Validation Procedures



Token Lifetimes and Expiration



Tokens have a limited lifetime

If a token has expired, validation will fail

Token Lifetimes and Expiration

Identity token

Very short lifetime (default: 5 minutes)

Used right after delivery

Applications often implement their own expiration policies

Access token

Longer lifetime (default: 1 hour)

Must be renewed to regain access to resources

The IDP controls the expiration policy



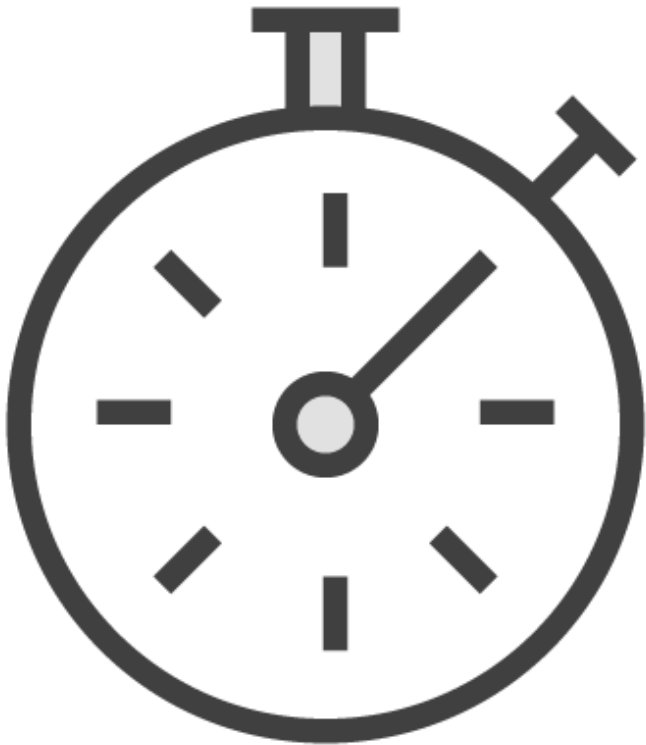
Demo



Token Lifetimes and Expiration



Gaining Long-lived Access with Refresh Tokens

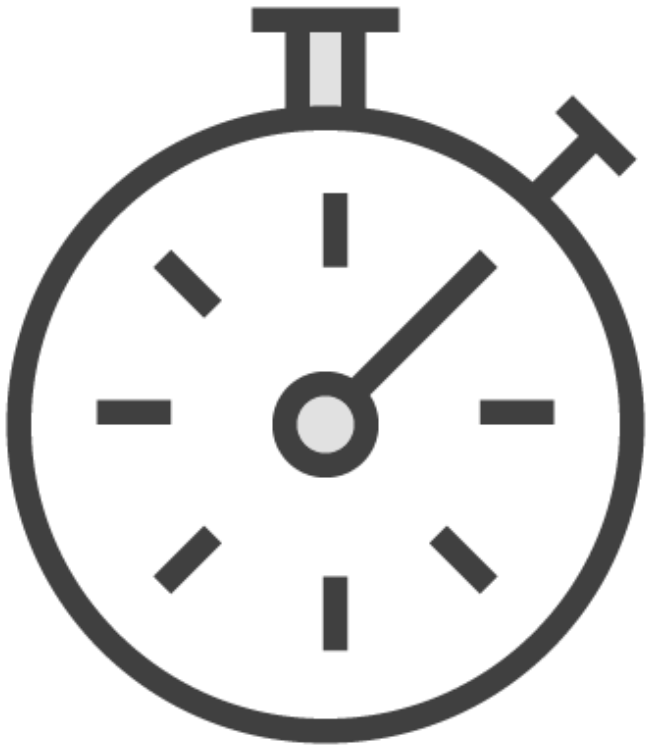


When a token expires, the flow can be triggered again to get a new one

Confidential clients can use refresh tokens to get new tokens via the back channel

A refresh token is a credential to get new tokens

Gaining Long-lived Access with Refresh Tokens



Tokens are refreshed via the token endpoint

A client must authenticate itself when refreshing tokens

Scope: “offline_access”

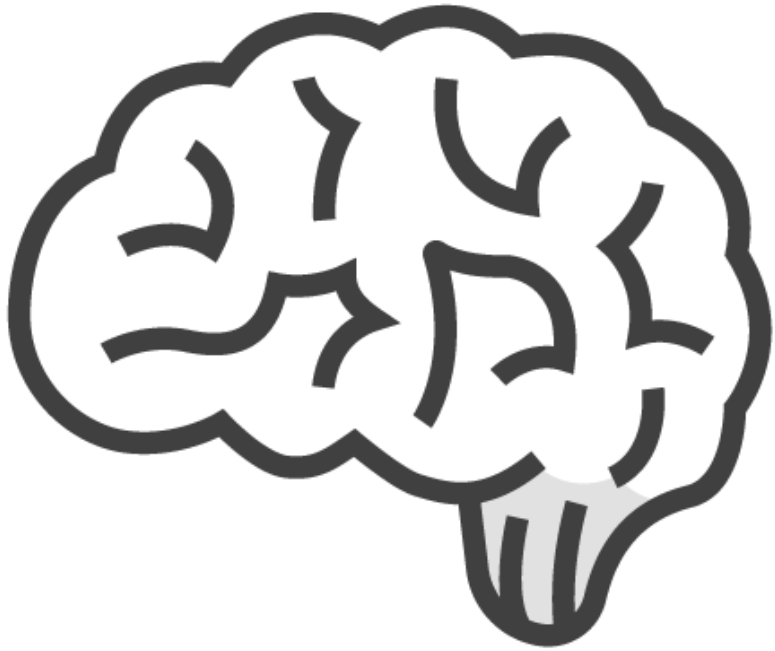
Demo



Gaining Long-Lived Access With Refresh Tokens



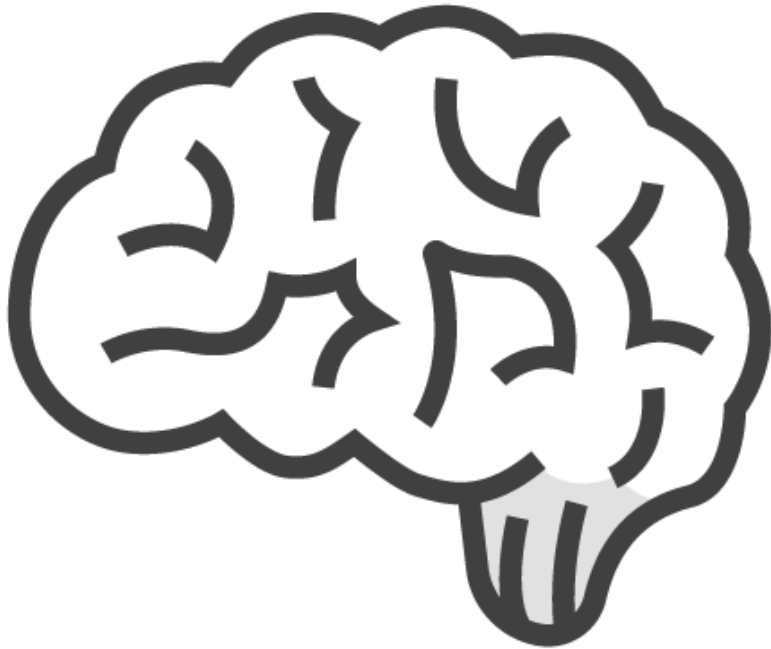
Working with Reference Tokens



Self-contained tokens (like JWT) can be validated without communicating with the IDP on each call

... but they don't offer direct lifetime control

Working with Reference Tokens



A reference token is an identifier, linked to a token stored at level of the IDP

Token introspection endpoint

More direct lifetime control, but also more communication with the IDP

Demo



Working With Reference Tokens



Token Revocation



Tokens can be revoked through an administration tool

Clients can programmatically revoke tokens via the token revocation endpoint

Demo



Revoking Tokens



Token Validation



Middleware takes care of validation

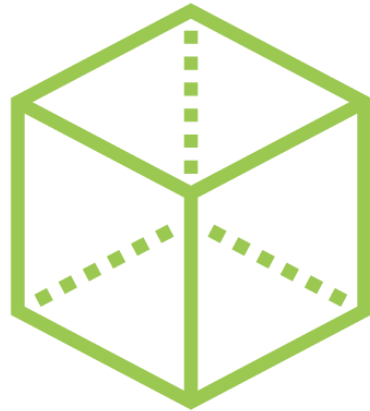
Validation procedures can differ between flows

Not every client or IDP uses the same validation procedures

Validation Procedures



Identity token
(client level)



Authorization
code
(client level)



Identity token
(client level,
from token
endpoint)



Access token
(client level)



Access token
(API level)

Identity Token Validation



Signature

Nonce

Issuer

Audience

Expiration

Authorization Code Validation



Code hash is calculated from the authorization code

Must match `c_hash` in identity token: this links the authorization code to the identity token

Identity Token Validation (Token Endpoint)



Signature / Nonce / Issuer / Audience / Expiration

Subject and Issuer claims must match those from the identity token returned from the authorization endpoint

Access Token Validation (Client)



Access token hash is calculated from the access token

Must match at_hash in identity token: this links the access token to the identity token

“The methods used by the resource server to validate the access token are beyond the scope of this specification but generally involve an interaction or coordination between the resource server and the authorization server.”

OAuth2 specification



Access Token Validation (API)



Signature

Issuer

Expiration

Audience

Access Token Validation (API)



Audience value gives access to a set of resources

Scopes define which specific (sub)set of resources the token allows access to

Summary



Tokens have a limited lifetime

Refresh tokens can be used to gain long-lived access for confidential clients



Summary



Reference tokens are identifiers linked to a token at level of the IDP

- Better control over lifetime
- More communication with the IDP

Tokens can be revoked by calling the token revocation endpoint