Article

Creating a client secret

Generate a signed token to identify your client application.

Overview

JSON Web Token (JWT) is an open-standard (RFC 7519) that defines a way to transmit information securely. Account and Organizational Data Sharing requires JWTs to authorize each validation request. Create the token, then sign it with the private key you downloaded from your developer account.

To create a signed JWT:

1. Create the JWT header.
2. Create the JWT payload.
3. Sign the JWT.

To create a JWT, use the following fields and values in the JWT header:

`alg`
: The algorithm used to sign the token. For Account and Organizational Data Sharing, use `ES256`.

`kid`
: A 10-character key identifier generated for the Account and Organizational Data Sharing private key associated with your developer account.

The JWT payload contains information specific to the Account and Organizational Data Sharing REST API and the client app, such as the issuer, subject, and expiration time. Use the following claims in the payload:

`iss`
: The *issuer* registered claim identifies the principal that issued the client secret. Because the client secret belongs to your developer team, use the 10-character Team ID associated with

your developer account.

`iat`

The *issued at* registered claim indicates the time at which you generated the client secret, in terms of the number of seconds following the epoch (January 1 1970 00:00:00), in UTC.

`exp`

The *expiration time* registered claim identifies the time on or after which the client secret expires. It's an error to request an expiration time more than `15777000` seconds (six months) in the future, as measured by the clock on Apple's servers.

`aud`

The *audience* registered claim identifies the intended recipient of the client secret. Because the client secret is sent to the validation server, use `https://appleid.apple.com` as the audience.

`sub`

The *subject* registered claim identifies the principal that is the subject of the client secret. Because this client secret is meant for your app, use the same App ID or Services ID that you use as the `client_id` to generate and refresh tokens. The value is case-sensitive.

After creating the JWT, sign it using the Elliptic Curve Digital Signature Algorithm (ECDSA) with the P-256 curve and the SHA-256 hash algorithm. A decoded `client_secret` JWT token has the following format:

```
{
    "alg": "ES256",
    "kid": "ABC123DEFG"
}
{
    "iss": "DEF123GHIJ",
    "iat": 1437179036,
    "exp": 1493298100,
    "aud": "https://appleid.apple.com",
    "sub": "com.mytest.app"
}
```

Regardless of the programming language you're using with the Account and Organizational Data Sharing REST API, there are a variety of open source libraries available online for creating and signing JWT tokens. For more information, see JWT.io.

See Also

Generating tokens

`Fetch Apple's public key for verifying token signature`

Retrieve the public key associated with the cryptographic identity Apple uses to sign the token.

`Generate and validate tokens`

Validate an authorization grant code delivered to your app to obtain tokens, or validate an existing refresh token.