Article

Generating reader tokens for the Verifier API

Configure your server to generate reader tokens to prepare a device for mobile document reading.

Overview

The Verifier API supports displaying your brand's name and logo in the reader UI during a document request. To show your brand's name and logo, your app needs to pass a reader token as an argument to the `prepare(using:)` method.

Reader tokens are JSON Web Tokens (JWT) generated by your server containing your brand ID, key ID, and a reader instance identifier. You can obtain a brand ID and key ID by enrolling with Apple Business Register.

The person who enrolls your organization with Apple Business Register will need to enter your company's business name, and upload your brand's logo and a public signing key.

Generate a reader token on your server

You need to link a reader token to the device that you'll use to read documents, as well as your brand information from Apple Business Register. To generate a reader token:

1. Fetch the device's `readerInstanceIdentifier`.
2. Send the reader instance identifier to your server.
3. On your server, create a JWT containing the device's reader instance identifier, along with your brand and key IDs from Apple Business Register.
4. Sign the JWT with your server's private signing key and return the token to your app.

After creating the reader token, confirm the validity of the token authorization by calling `prepare(using:)`. If the method returns successfully, the framework returns a `MobileDocumentReaderSession` and the device is ready to read mobile documents. After using a reader token to prepare a reader device, your app can cache and reuse it for up to 48 hours.

A JWT has two sections, a header and a payload. The header describes the token and the cryptographic operations applied to the payload. The payload contains a set of cryptographically signed claims.

Construct a token with these fields in the header:

`alg`
: The algorithm you use to sign the token. Use the ES-256 algorithm to sign your token.

`kid`
: A key identifier that represents your server's signing key. You can obtain this value from Apple Business Register.

`typ`
: A type parameter that you set to `JWT`.

In the payload section of the token, include the following:

`aud`
: The audience of the token. Set this value to `apple-identityservices-v1`.

`iss`
: The issuer of the token. This is the Brand ID that you obtain from Apple Business Register.

`sub`
: The subject of the token. This is the reader instance identifier obtained from your app.

`iat`
: The Issued At registered claim key. The value of this claim indicates the token creation time, in terms of the number of seconds since UNIX Epoch, in UTC.

`exp`
: The Expiration Time registered claim key. The value of this claim indicates when the token expires, in terms of the number of seconds since UNIX Epoch, in UTC. This must be set to a value after and no later than 5 minutes after the value of `iat`.

When decoded, a token for use with the Verifier API has the following format:

```
{
    "alg": "ES256",
    "kid": "ABCDEFGHIJ",
    "typ": "JWT"
}
```

```
{
    "aud": "apple-identityservices-v1",
    "iss": "KLMNOPQRST",
    "sub": "UVWXYZ1234",
    "iat": 1685980800,
    "exp": 1685981100
}
```

To learn more about JWT, see the JSON Web Token (JWT) specification. You can find a collection of libraries for generating signed tokens at JWT.io.

See Also

Mobile document reader

Adopting the Verifier API in your iPhone app

Configure and test ID Verifier support in your app for reading mobile documents.

{} Checking IDs with the Verifier API

Read and verify mobile driver’s license information without any additional hardware.

`class MobileDocumentReader`

An object for configuring mobile document reading on the current device.

`class MobileDocumentReaderSession`

The object you use to start reading a mobile document.