Article

Generating JSON Web Tokens for API requests

Create JSON Web Tokens signed with your private key to authorize requests for App Store Server API and External Purchase Server API.

Overview

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a way to securely transmit information. The App Store Server API and External Purchase Server API require a JWT to authorize each request you make to the API. You create the token, signing it with the private key you downloaded from App Store Connect. For more information about creating keys, see Creating API keys to authorize API requests.

To generate a signed JWT:

1. Create the JWT header.

2. Create the JWT payload.

3. Sign the JWT.

Include the signed JWT in the authorization header of each API request. Generate a new signed JWT for each new request.

> **Tip**
>
> The App Store Server Library provides an API client and creates JWTs for use with the App Store Server API. For more information, see Simplifying your implementation by using the App Store Server Library.

Create the JWT header

To create a JWT to communicate with the App Store Server API or External Purchase Server API, use the following fields and values in the header:

Header Field	Value
`alg` - Encryption Algorithm	`ES256` All JWTs must be signed with ES256 encryption
`kid` - Key ID	Your private key ID from App Store Connect (Ex: `2X9R4HXF34`)
`typ` - Token Type	`JWT`

To get your key ID, copy it from App Store Connect by logging in to App Store Connect, then:

1. Select Users and Access, then select the Keys tab.
2. The key IDs appear in a column under the Active heading. Hover the cursor next to a key ID to display the Copy Key ID link.
3. Click Copy Key ID.

If you have more than one API key, copy the key ID of the private key that you use to sign the JWT.

Here's an example of a JWT header:

```
{
  "alg": "ES256",
  "kid": "2X9R4HXF34",
  "typ": "JWT"
}
```

Create the JWT payload

The JWT payload contains information specific to the App Store Server API and External Purchase Server API, such as issuer ID and expiration time. Use the following fields — also known as JWT claims — to include these values in the JWT payload:

Payload Field	Value
`iss` - Issuer	Your issuer ID from the Keys page in App Store Connect (Ex: “`57246542–96fe–1a63–e053–0824d011072a"`)
`iat` - Issued At	The time at which you issue the token, in UNIX time, in seconds (Ex: `1623085200`)
`exp` - Expiration Time	The token’s expiration time, in UNIX time, in seconds. Tokens that expire more than 60 minutes after the time in `iat` are not valid (Ex: `1623086400`)
`aud` - Audience	`appstoreconnect-v1`
`bid` - Bundle ID	Your app’s bundle ID (Ex: “`com.example.testbundleid`”)

To get your issuer ID, log in to App Store Connect, then:

1. Select Users and Access, then select the Keys tab.

2. The issuer ID appears near the top of the page. To copy the issuer ID, click Copy next to the ID.

Here’s an example of a JWT payload:

```
{
  "iss": "57246542-96fe-1a63e053-0824d011072a",
  "iat": 1623085200,
  "exp": 1623086400,
  "aud": "appstoreconnect-v1",
  "bid": "com.example.testbundleid"
}
```

Note that the JWT is valid for up to one hour after the time you indicate in the `iat` field, or it expires sooner if you set the `exp` field for an earlier time.

Sign the JWT

Use the private key associated with the key ID you specified in the header to sign the token using ES256 encryption.

There are a variety of open source libraries available online for creating and signing JWT tokens. See JWT.io for more information. For calls to the App Store Server API, consider using the App Store Server Library to create the JWTs instead. For more information, see Simplifying your implementation by using the App Store Server Library.

Include the JWT in the authorization header of the request

After you create and sign the JWT, provide it in the request's authorization header as a bearer token.

The following example for the App Store Server API shows a `curl` command using a bearer token. Replace the text `[signed token]` with the value of the signed JWT itself. Replace `{transactionId}` with a transaction identifier of your customer.

```
curl -v -H 'Authorization: Bearer [signed token]'
"https://api.storekit.itunes.apple.com/inApps/v1/subscriptions/{transactionId}"
```

See Also

Essentials

- Simplifying your implementation by using the App Store Server Library

  Use Apple's open source library to create JSON Web Tokens (JWT) to authorize your calls, verify transactions, extract transaction identifiers from receipts, and more.

- Creating API keys to authorize API requests

  Create API keys you use to sign JSON Web Tokens and authorize API requests.

- Identifying rate limits

  Recognize the rate limits that apply to App Store Server API endpoints and handle them in your code.

- App Store Server API changelog

  Learn about new features and updates in the App Store Server API.