

[App Store Server API](#) / Simplifying your implementation by using the App Store Server Library

## Article

# 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.

## Overview

The App Store Server Library is an open source library from Apple, available in four languages. It makes adopting the [App Store Server API](#) and working with JSON Web Signature (JWS) transactions easier. Find the App Store Server Library for each language in the following GitHub repositories:

- Swift: [App Store Server Swift Library](#)
- Java: [App Store Server Java Library](#)
- Python: [App Store Server Python Library](#)
- Node: [App Store Server Node Library](#)

Choose the language that best supports your server and expertise.

The App Store Server Library offers the following capabilities:

- An API client that encodes [App Store Server API](#) requests, decodes the responses, and creates the JSON Web Token (JWT) you use to authenticate the calls. For more information on using JWTs, see [Generating JSON Web Tokens for API requests](#).
- Functions that verify JWS transactions, to verify that Apple signed the transaction data you get in API responses, from [App Store Server Notifications V2](#) and from devices using [StoreKit](#). See the functions `verifyAndDecodeTransaction`, `verifyAndDecodeApp`

Transaction, and `verifyAndDecodeRenewalInfo`, available in each language the library supports.

- A utility that extracts transaction identifiers from receipts. The App Store Server API endpoints take a transaction identifier in the path parameter. Use this utility as you migrate from using `verifyReceipt` with [App Store Receipts](#) to using the App Store Server API for transaction information.
- A function that generates JWS signatures, which you use in your app for promotional offer signatures, Advanced Commerce API in-app requests, and introductory offer eligibility. For more information, see [Generating JWS to sign App Store requests](#).

For more information, see the WWDC23 session [Meet the App Store Server Library](#).

---

## See Also

### Essentials

#### Creating API keys to authorize API requests

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

#### 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.

#### 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.