Documentation Language: Swift

StoreKit / … / Original API for In-App Purchase / Choosing a receipt validation technique

Article

Choosing a receipt validation technique

Select the type of receipt validation, on the device or on your server, that works for your app.

Overview

> **Note**
>
> The receipt isn't necessary if you use `AppTransaction` to validate the app download, or `Transaction` to validate in-app purchases. Only use the receipt if your app uses the Original API for In-App Purchase, or needs the receipt to validate the app download because it can't use `AppTransaction`.

An App Store receipt provides a record of the sale of an app and any purchases the person makes within the app. You can authenticate purchased content by adding receipt validation code to your app or server. Receipt validation requires an understanding of secure coding techniques to employ a solution that's secure and unique to your app.

Choose a validation technique

There are two ways to verify a receipt's authenticity:

- Locally, on the device. Validating receipts locally requires code that reads and validates a binary file that Apple encodes and signs as a PKCS #7 container. For more information, see Validating receipts on the device.
- On your server with the App Store. Validating receipts with the App Store requires secure connections between your app and your server, and between your server and the App Store. For more information, see Validating receipts with the App Store.

Compare the approaches and determine the method that best fits your app and your infrastructure. You can also choose to implement both approaches. For managing auto-renewable subscriptions, see the following table for the key advantages that server-side receipt validation provides over on-device receipt validation:

Capability	**On-device validation**	**Server-side validation**
Validates authenticity of receipt	Yes	Yes
Includes subscription renewal transactions	Yes	Yes
Includes additional subscription information	No	Yes
Resistant to device clock change	No	Yes

Receipts contain non-consumable in-app purchases, auto-renewable subscriptions, and non-renewing subscriptions indefinitely. Consumable in-app purchases remain in the receipt until you call `finishTransaction(_:)`. You may choose to maintain and manage records of consumable in-app purchases on your server.

Get the latest receipt

The App Store updates receipts immediately after completed purchases. When you call verifyReceipt from your server, the App Store returns the latest transaction information, regardless of the contents of the receipt you send in the request.

On the device, the system updates the receipt immediately when it has an internet connection, and any of the following occur:

- The customer completes an in-app purchase.
- The app launches its transaction observer (`SKPaymentTransactionObserver`) and has unfinished transactions or subscription renewals.
- The app calls `restoreCompletedTransactions()` or `restoreCompletedTransactions(withApplicationUsername:)` to restore transactions.
- The app sends a request to `SKReceiptRefreshRequest` to get a receipt if the receipt is invalid or missing.

Transactions can also occur at times when the app isn’t running. When necessary, call `restoreCompletedTransactions()` to ensure the receipt you’re working with is up-to-date. For example, if a customer purchases an auto-renewable subscription on another device, call this method to get those transactions and update the receipt.

To ensure that your app receives all transactions, add the transaction observer, `add(_:)`, at app launch time. For more information, see Setting up the transaction observer for the payment queue.

Related Sessions from WWDC 2018

Session 705: Engineering Subscriptions

See Also

Purchase validation

Validating receipts with the App Store

Verify transactions with the App Store on a secure server.

~~`var appStoreReceiptURL: URL? { get }`~~

The file URL for the bundle’s App Store receipt.

Deprecated

~~`class SKReceiptRefreshRequest`~~

A request to the App Store to get the app receipt, which represents the customer’s transactions with your app.

Deprecated