

[StoreKit](#) / [In-App Purchase](#) / Supporting subscription offer codes in your app

Article

Supporting subscription offer codes in your app

Provide subscription service for customers who redeem offer codes through the App Store or within your app.

Overview

To help you acquire, retain, and win back subscribers, you can use offer codes. Offer codes are alphanumeric codes that provide subscriptions at a discount or for free for a specific duration. Configure the offers and create offer codes in App Store Connect, and distribute them to your customers. Customers can redeem offer codes in the App Store, using offer code redemption URLs, or in your app if it implements one of the following APIs:

- [`offerCodeRedemption\(isPresented:onCompletion:\)`](#) on iOS, iPadOS, macOS, and visionOS
- [`presentOfferCodeRedeemSheet\(in:\)`](#) on iOS, iPadOS, and visionOS
- [`presentOfferCodeRedeemSheet\(from:\)`](#) on macOS

When customers redeem a valid offer code, your app receives a successful transaction. If customers redeem offer codes in the App Store and don't have your app installed, they're prompted to download it as part of the redemption flow. Successfully redeeming an offer code entitles the customer to the auto-renewable subscription, the same as a purchase does. Your app needs to provide service for the product.

Set up offer codes in App Store Connect

Configure offers and manage your offer codes in App Store Connect. You can have up to 10 active offers per subscription, and create codes for a maximum of 1,000,000 redemptions per app, per

quarter. There are two types of offer codes: one-time use codes, and custom codes. The offer code redemption APIs support both.

To distribute offer codes to your customers, download them from App Store Connect. For more information on creating and distributing offer codes, and to learn which type of offer code may work for your campaign, see [Set up offer codes](#).

Redeem offer codes in your app

To display the system sheet for customers to redeem offer codes within your app, call one of the redemption APIs, depending on your app's UI implementation:

- Call `offerCodeRedemption(isPresented:onCompletion:)` if your app uses SwiftUI.
- Call `presentOfferCodeRedeemSheet(in:)` if your app uses UIKit.
- Call `presentOfferCodeRedeemSheet(from:)` if your app uses AppKit for macOS.

The redemption sheet takes care of the redemption flow, including alerting users about invalid entries, as appropriate. Invalid entries may include, for example, expired offer codes, invalid codes, or codes that would result in a subscription downgrade.

When customers redeem an offer code, StoreKit emits the resulting transaction in [updates](#). Set up a transaction listener as soon as your app launches to receive new transactions while the app is running.

Including the redemption sheet in your app is recommended, but optional. For more guidance on supporting offer code redemption within your app, see Human Interface Guidelines > [In-app purchase](#).

Support offer codes redeemed outside of your app

Customers may redeem offer codes outside your app, by entering the offer code in the App Store, or by using a redemption URL. To handle offer codes — and other transactions that can occur outside of your app — your app needs to use [updates](#) on [Transaction](#) to receive new transactions while the app is running. Create a [Task](#) to iterate through the transactions from the listener as soon as your app launches. For more information and sample code, see [updates](#).

When the app launches, it needs to check [all](#) or [currentEntitlements](#) on [Transaction](#) to get any transactions that may have occurred while the app wasn't running. Process the transactions to ensure your app provides service for all products it's entitled to.

Identify subscriptions purchased with offer codes

When customers successfully redeem subscription offer codes, the transaction or subscription renewal information contain fields that identify the offer and its offer type. Find the offer code details in the transaction information, in your app and on your server, as follows.

In your app, use the following StoreKit APIs to locate the offer code information:

- See the [Transaction](#) properties [offerID](#) and [offerType](#). An offer type value of [code](#) indicates the customer redeemed an offer code.
- Some offer code redemptions may apply to an auto-renewable subscription's next renewal period, for example, if the customer is already subscribed. In this case, see the [Product.SubscriptionInfo.RenewalInfo](#) properties [offerID](#) and [offerType](#). An offer type value of [code](#) indicates the customer redeemed an offer code.

On your server, use the following server-side APIs to locate offer code information:

- In the [App Store Server API](#), when you call endpoints such as [Get Transaction History](#), [Get All Subscription Statuses](#), and others, the response contains the signed transaction, [JWSTransaction](#). In its decoded payload, [JWSTransactionDecodedPayload](#), look for the fields [offerIdentifier](#) and [offerType](#). An [offerType](#) value of 3 indicates the customer redeemed an offer code.
- The [App Store Server Notifications V2](#) sends a notification with an [OFFER_REDEEMED notificationType](#) when someone redeems an offer code for an active subscription. It sends a [SUBSCRIBED](#) notification type if someone redeems the offer code as an initial purchase or to resubscribe. The decoded payloads [JWSTransactionDecodedPayload](#) and [JWSRenewalInfoDecodedPayload](#) contain the fields [offerIdentifier](#) and [offerType](#). An [offerType](#) value of 3 indicates the customer redeemed an offer code.

Provide subscription service to new and existing customers

When you acquire new customers with an offer code, they already have an auto-renewable subscription when they open your app for the first time. In addition to providing subscription service, you may need to update your backend system's records. Your app follows these steps:

1. When the app launches, check [all](#) or [currentEntitlements](#) on [Transaction](#) to get all transactions or current entitlements, respectively. StoreKit automatically validates the transactions, and returns verified results in [VerificationResult.verified\(_\)](#). To perform your own validation, use the [jwsRepresentation](#) property.
2. To determine if a subscription results from an offer code redemption, check the [offerID](#) and [offerType](#) properties on the subscription's [Transaction](#).
3. Provide the subscription service based on the offer and call [finish\(\)](#) on [Transaction](#).

4. Guide new customers through your new-user experience, as needed. Update your backend system's records.



When an existing customer redeems an offer code within your app, the transaction comes in through the [updates](#) sequence on [Transaction](#). Process the transaction as usual, providing service based on the offer, and call [finish\(\)](#).

Supporting systems earlier than iOS 16 and iPadOS 16

If your app runs on iOS 16 or earlier, and iPadOS 16 or earlier, use [presentCodeRedemptionSheet\(\)](#) to display the offer code redemption UI. Otherwise, use the APIs referred to in this article. For more information about supporting earlier systems, see [Implementing offer codes in your app](#).

See Also

Subscription offers and offer codes

-  Supporting win-back offers in your app
Re-engage previous subscribers with a free or discounted offer for an auto-renewable subscription, for a specific duration.
-  Merchandising win-back offers in your app
Present win-back offers to eligible customers in your app with the win-back offer sheet or by implementing custom merchandising.

`struct SubscriptionOffer`
Information about a subscription offer that you configure in App Store Connect.

`struct OfferType`
The types of offers for auto-renewable subscriptions.