Sample Code

Understanding StoreKit workflows

Implement an in-app store with several product types, using StoreKit views.

Download

iOS 26.0+ | iPadOS 26.0+ | macOS 26.0+ | Xcode 26.0+

Overview

StoreKit offers an expansive API for creating an almost limitless variety of in-app stores. This sample app shows you precisely what you need to implement in order to get up and running with a store in your app that can implement several In-App Purchase (IAP) types, including subscriptions.

Configure and run the project

The design of this sample doesn’t require any special setup, or even the creation of products in App Store Connect; it uses StoreKit Testing in Xcode and the Transaction Manager window in Xcode to test In-App Purchase.

This project comes with the following products set up locally:

- A consumable item
- A non-consumable item
- A collection of consumable items
- A monthly subscription
- A yearly (annual) subscription
- A premium yearly (annual) subscription, with a “1 month free” introductory offer.

For more information on the product types that App Store Connect supports, see In-app purchase types

> **Note**
>
> The transactions this app demonstrates don't get information from App Store Connect and don't incur charges. Xcode simulates successful transactions without processing actual payments.

To see the pre-configured local StoreKit products, select the `LocalConfiguration` file in the project's file navigator. This file lists all of the types, descriptions, and properties of the demo products the app supports.

For more information on using the `LocalConfiguration` file, including creating products and inspecting or modifying transactions, see Testing in-app purchases with StoreKit transaction manager in Xcode.

To run the app, press the Xcode run button, or press command-R. To purchase any demo products, click or tap the buy button on a product, and the system presents the payment sheet that you can either accept (purchase) or cancel. The app shows the status and results of purchases on the status display near the top of the app's window.

Reset the project's In-App Purchases

After you purchase products in the sample app, to reset the product to its initial unpurchased state, select Debug > StoreKit > Manage Transactions. In the sample app's Transactions panel, select each transaction you want to delete, then click the minus ("-") button at the bottom of the list to delete it. In addition, to reset the consumable item information that the app stores, run the command `defaults delete com.example.apple–samplecode.StoreKitWorkflows` in Terminal.

For more information on all the capabilities StoreKit provides for the construction of in-app stores and customized StoreKit Views, see StoreKit views.

See Also

In-App Purchase

In-App Purchase

Offer content and services in your app across Apple platforms using a Swift-based interface.

Getting started with In-App Purchase using StoreKit views

Set up an in-app store using SwiftUI and StoreKit views.