Article

Testing at all stages of development with Xcode and the sandbox

Verify your implementation of In-App Purchases by testing your code throughout its development.

Overview

Use the Apple sandbox and Xcode test environments to test your implementation of in-app purchases using the StoreKit framework. Comprehensive testing can help you:

- Ensure a seamless purchase flow to provide a positive customer experience in your app.
- Implement sound logic that covers all scenarios, such as purchases, restores, and subscription offers.
- Validate that purchases behave correctly in production after your app is available in the App Store.

The tools you need to test in-app purchases, non-renewing subscriptions, and auto-renewable subscriptions from early development through beta testing are:

StoreKit Testing in Xcode
: For early development, continuous integration, and debugging. For more information, see StoreKit Test.

Sandbox
: For testing scenarios using in-app purchase data you set up in App Store Connect. For more information, see Testing In-App Purchases with sandbox.

TestFlight
: For managing beta testing with internal and external testers. TestFlight uses a beta build of your app or App Clip that you upload to App Store Connect. For more information, see Beta Testing Made Simple with TestFlight.

Choose the tools that support the test scenarios you need. Make sure you're able to perform the setup required for the tools you choose.

During the early stages of development, you may not be ready to configure in-app purchases in App Store Connect. StoreKit Testing in Xcode lets you configure the information locally. You can test StoreKit transactions before you create Sandbox Apple Accounts, without a network connection. You can test your app in Simulator or on real devices.

After you set up in-app purchases in App Store Connect, start using the sandbox environment to test the product information your app will use in production. Testing in the sandbox lets you test transactions from end-to-end and from your app to your server. You can also test any server-to-server functionalities your app depends on, such as transaction validation and App Store Server Notifications.

TestFlight lets you get feedback from members of your team or from external testers. TestFlight uses the sandbox environment for in-app purchases. Transactions and purchases that occur in the sandbox don't incur charges. The following table compares the test environments and features:

Test environment	Requires App Store Connect setup	Provides receipts and JSON Web Signature (JWS) transactions signed by the App Store	Provides subscription renewal information signed by the App Store
StoreKit Testing in Xcode	No	No (signed by Xcode)	No (signed by Xcode)
Sandbox	Yes	Yes	Yes
TestFlight (uses the sandbox)	Yes	Yes	Yes

None of the test environments charge users when they test buying a product. The App Store doesn't send emails for purchases or refunds made in the test environments.

Control the test environment

To set up and run test scenarios, you often need to control the test environment. For example, you may want to reset a test account to rerun the same test multiple times, or mimic actions users take outside your app that affect the test conditions. The following table shows the capabilities each tool has to control the test environment:

Test scenario	Sandbox	StoreKit Testing in Xcode
Test different storefronts to affect price tiers and locale	Yes	Limited (no price tiers)
Clear the purchase history	Yes	Yes
Test subscription upgrades, downgrades, cross-grades, and auto-renew cancellations	Yes	Yes
Reset eligibility for introductory offers	Yes	Yes
Introduce forced StoreKit errors for testing	No	Yes
Speed up or slow down the rate of time for testing subscription renewals	Yes	Yes

For more information about speeding up renewal periods for testing, see Test in-app purchases.

Test common StoreKit scenarios

All apps that offer in-app purchases need to support restoring purchases, displaying in-app purchases to the customer, and handling basic transactions. The following table lists common test scenarios and whether they're testable in the sandbox or Xcode:

Test scenario	Sandbox	StoreKit Testing in Xcode
Restore purchases	Yes	Yes
Finish a transaction with `finish()` or `finishTransaction(_:)`	Yes	Yes
Buy a consumable or non-consumable in-app purchase	Yes	Yes
Repurchase a non-consumable purchase for repeated testing	Yes	Yes
Purchase an auto-renewable subscription	Yes	Yes
Purchase a non-renewing subscription	Yes	Yes

Test scenario	Sandbox	StoreKit Testing in Xcode
Refund an in-app purchase	Yes	Yes
Test an interrupted purchase, where the user must complete actions outside the app	Yes	Yes
Test a failed purchase attempt when payment authorization fails	Yes	Yes
Retrieve configured in-app purchases from App Store Connect	Yes	Yes (optionally); can also retrieve data from a StoreKit configuration file
Manage subscriptions within your app with `showManageSubscriptions(in:)` and `manageSubscriptionsSheet(isPresented:)`	Yes	Yes
Initiate a refund request. For more information, see Testing refund requests.	Yes	Yes

Test subscriptions and Ask to Buy

Depending on the in-app purchases your app offers, you may need to test scenarios that involve auto-renewing subscriptions, introductory offers, promotional offers, and Ask to Buy. The following table lists test scenarios and whether they're testable in the sandbox or Xcode:

Test scenario	Sandbox	StoreKit Testing in Xcode
Initiate an Ask to Buy transaction that results in a deferred state	Yes	Yes
Resolve an Ask to Buy transaction by approving or rejecting it	No	Yes
Redeem an introductory offer for an auto-renewable subscription	Yes	Yes
Redeem a promotional offer for an auto-renewable subscription	Yes	Yes

Test scenario	Sandbox	StoreKit Testing in Xcode
Redeem an offer code for an auto-renewable subscription	No	Yes
Process a subscription renewal	Yes	Yes
Process a revoked or refunded subscription	Yes	Yes
Respond to a customer canceling a subscription and disabling auto-renew	Yes	Yes
Respond to an expired subscription	Yes	Yes
Process a subscription upgrade or downgrade	Yes	Yes
Process a subscription cross-grade with the same or different duration	Yes	Yes
Test a price increase for an auto-renewable subscription	No	Yes
Test billing retry and billing grace period	Yes	Yes

For more information, see Approve what kids buy with Ask to Buy and Testing introductory offers.

See Also

In-App Purchase Testing

- Testing In-App Purchases with sandbox

  Test your implementation of In-App Purchases using real product information and server-to-server transactions in the sandbox environment.

- Testing refund requests

  Test your app's implementation of refund requests, and your app's and server's handling of approved and declined refunds.

- Testing win-back offers in Xcode

  Validate your app's handling of win-back offers that you configure for the testing environment.

- Testing Ask to Buy in Xcode

Validate your app's handling of Ask To Buy in the testing environment.