Article

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.

Overview

Use the Apple sandbox environment to test your implementation of In-App Purchases that use the StoreKit framework on devices with real product information from App Store Connect. Transactions from test accounts don't incur charges because the sandbox environment simulates successful transactions using the App Store's infrastructure without processing actual payments.

Apps in the development and beta-testing stages of your product cycle use the sandbox environment for In-App Purchases. These apps include:

- Development-signed apps you build and run from Xcode
- Apps you download from TestFlight

When you sign in to a Sandbox Apple Account on your device, you gain access to sandbox settings that control the test environment. Use these settings to change the subscription renewal rate, clear purchase history, simulate win-back offers, and more.

Prepare for sandbox testing

Before you start testing In-App Purchases in the sandbox environment, make sure that:

1. Your Apple Developer Program account is active. For more information, see Become a member.
2. Your membership Account Holder has signed the Paid Applications Agreement, as described in Sign and update agreements.
3. You set up the product information in App Store Connect for the app you're testing. At minimum, set up a product reference name, product ID, a localized name, and a price. For more

information, see Overview for configuring in-app purchases.

4. You create Sandbox Apple Accounts in App Store Connect. For more information, see Create a Sandbox Apple Account.
5. You grant permission for locally installed apps to run on the device if you test a development-signed build on iOS, watchOS, or visionOS devices. For more information, see Enabling Developer Mode on a device.

For more information on the workflow from signing agreements to testing, see App Store Connect workflow.

Next, prepare your app to test:

- During developement, build and run your app from Xcode. At this stage, you don't need to upload your app binary to App Store Connect to test it in the sandbox environment.
- During beta testing, download your app from TestFlight.

Finally, to gain access to the sandbox settings, follow the instructions below to sign in to your Sandbox Apple Account. The instructions differ for a development-signed app and for an app you download from TestFlight.

Sign in to your Sandbox Apple Account for a development-signed app

In iOS and iPadOS, the sandbox account appears in Settings > Developer after the first time you use the device to attempt a purchase in a development-signed app. There's no need to sign out of the non-Sandbox Apple Account. Sign in using a Sandbox Apple Account.

In macOS, the sandbox account appears in App Store Settings after the first time you use the device to make an In-App Purchase. Open App Store > Settings and click the Sign In button to sign in using a Sandbox Apple Account. When you're done testing, sign out of the Sandbox Apple Account from the App Store Settings.

If your test device has macOS 11.1 or earlier installed, sign out of the Mac App Store, then build your app in Xcode and launch it from Finder.

The first time you make a purchase in a development-signed app, sign in to the App Store using your Sandbox Apple Account to begin testing. When you sign in, the text `[Environment: Sandbox]` appears, indicating that you've connected to the test environment. If `[Environment: Sandbox]` doesn't appear, you're using the production environment.

Sign in to your Sandbox Apple Account for a TestFlight app

Apps that you download from TestFlight always run in the sandbox environment. However, to access the sandbox controls for testing, you need to sign out of your production Apple Account in Media & Purchases, and sign in with a Sandbox Apple Account in Developer settings.

Follow these steps to access sandbox controls for TestFlight in iOS:

1. Download your beta app from TestFlight. You need to be signed in with a production Apple Account to download a beta app. For more information, see TestFlight Overview.
2. Open Settings and select your Apple Account.
3. Select Media & Purchases.
4. On the pop-up menu, select Sign Out.
5. Next, open Settings and select Developer. (This option is available when you enable Developer Mode).
6. Scroll to the bottom of the screen, and select Sign In under Sandbox Apple Account. Enter your Sandbox Apple Account credentials.

Your device now has access to the sandbox controls for testing your beta app, and In-App Purchases use your Sandbox Apple Account.

Important

Signing out of Media & Purchases can remove your access to purchased content in production apps on your device. Consider using a dedicated testing device when testing a TestFlight app with sandbox controls.

If you sign back in to your Apple Account in Media & Purchases, apps you install through TestFlight revert to attributing In-App Purchases to your Apple Account and not to your Sandbox Apple Account.

Manage sandbox settings from App Store Connect or iOS Account Settings

Sandbox settings enable you to control the testing environment to set up your tests. You can manage the settings in App Store Connect or in iOS Account Settings, whichever you prefer. The settings affect the sandbox the same way, regardless of which method you choose. However, some settings are available only in App Store Connect. The following table compares the settings on both platforms:

Sandbox setting or feature	Available in App Store Connect	Available in iOS
Create or delete a Sandbox Apple Account	Yes	No
Create or delete a Sandbox Test Family	Yes	No
Remove a member from a Sandbox Test Family	Yes	No
Change the storefront country or region	Yes	No
Change the subscription renewal rate	Yes	Yes
Enable interrupted purchases	Yes	Yes
Clear purchase history for a Sandbox Apple Account	Yes	Yes
Change sharing status for a Sandbox Test Family member	Yes	Yes
Select whether to display win-back offer sheets in the app	No	Yes

To access the Account Settings in iOS, follow these steps:

1. Open Settings and select Developer.
2. Select the Sandbox Apple Account.
3. On the pop-up sheet, select Manage. The Account Settings page appears.

> **Note**
>
> Changes you make to product metadata in App Store Connect can take up to one hour to appear in the sandbox environment.

For information on all aspects of managing sandbox settings in App Store Connect, see Overview of testing in sandbox.

Get server notifications for the sandbox environment

Along with testing In-App Purchase in your app, you can also test related server operations in the sandbox environment. To enable App Store Server Notifications for the sandbox environment, see Enabling App Store Server Notifications.

As you test your app, your server receives notifications for In-App Purchase transactions. The notification payload identifies the sandbox environment in the `environment` property of the `data` object. For more information, Testing App Store server notifications.

The App Store Server API is also available in the sandbox environment. Use the endpoints' sandbox URLs to get information about transactions that occur during testing.

Test In-App Purchases for all regions

In-app products you create through App Store Connect are available for sale in every App Store region. You can also maintain a list of product identifiers that you make available in specific regions. To test In-App Purchases in multiple regions using the same Sandbox Apple Account:

1. Open App Store Connect.
2. Click the Sandbox Apple Account.
3. Change the App Store Country or Region setting.

For more information, see Manage Sandbox Apple Account settings.

To activate a storefront after you change the region in App Store Connect, sign out of the Sandbox Apple Account account on the device and sign back in. When testing this activation, you see the In-App Purchases appropriate to the region assigned to your Sandbox Apple Account. Changing the App Store Country or Region setting affects the storefront value in your app. For more information, see `Storefront`.

Clear the purchase history for a Sandbox Apple Account

To manage repeated testing using the same Sandbox Apple Account, clear its purchase history in App Store Connect or from Account Settings in iOS. On your iOS device, follow these steps:

1. Open Settings and select Developer.
2. Select the Sandbox Apple Account.
3. On the pop-up sheet, select Manage.
4. On the Account Settings page, select Clear Purchase History.
5. Sign out of the Sandbox Apple Account to clear the purchase history cache on the device. Sign back in to continue testing.

For information about clearing the purchase history from App Store Connect, see Manage Sandbox Apple Account settings.

Clearing the purchase history for Sandbox Apple Accounts with a high number of purchases can take longer than several minutes. When the process completes, the transaction history is empty and the Sandbox Apple Account is eligible for introductory offers. If you're signed in with the Sandbox Apple Account, you can purchase subscriptions previously subscribed to without causing duplicate purchases.

> **Note**
>
> Clearing the purchase history doesn't affect In-App Purchases that customers make on the App Store.

Topics

Product identifiers and requests

- Testing fetching product identifiers

  Verify that your app receives the correct product identifiers by inspecting or replicating your app's process for retrieving the identifiers.

- Testing invalid product identifier handling

  Verify that your app correctly handles invalid product identifiers.

- Testing a product request

  Verify that requests for products function properly in the sandbox environment by inspecting the App Store response.

Payment transactions

- Testing purchases made outside your app

  Verify that your app receives and handles transactions that occur outside your app, such as subscription purchases, renewals, and offer and promo code redemptions.

- Testing win-back offers in the sandbox environment

  Verify that your app receives and handles win-back offer transactions, including those made outside your app.

- Testing an interrupted purchase

Verify that your app handles an interrupted purchase by inspecting and invoking payment transactions.

Testing failing subscription renewals and In-App Purchases

Verify that your app handles failed subscription renewals that are in the billing retry or billing grace period states, as well as failed In-App Purchases.

Testing a payment request

Verify that requests for payment function properly in the sandbox environment by inspecting the calls to the payment transaction observer.

Subscriptions

Testing an auto-renewable subscription

Verify that your app handles a subscription lapse properly using the accelerated time rates within the sandbox environment.

Testing resubscribing from the subscriptions page

Verify that your app can reactivate an expired subscription by receiving a transaction callback or inspecting an updated receipt.

Testing disabling auto-renew

Verify that your app receives subscription updates when a user cancels a subscription by verifying the receipt or receiving a notification.

Family Sharing

Testing Family Sharing

Verify that your app handles auto-renewable subscriptions and non-consumable in-app purchases that family members share with Family Sharing.

Age Assurance

Testing Age Assurance in Sandbox

Check that your app responds correctly to Age Assurance scenarios and consent revocation using the Sandbox environment.

Refunds

Testing refund requests

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

Server notifications

- Testing App Store server notifications

  Confirm that App Store Server Notifications service responds properly in the sandbox environment.

Transaction observer

- Testing transaction observer code

  Verify that your app activates its payment transaction observer by using breakpoints.

- Testing a successful transaction

  Confirm that your app can make a successful transaction in the sandbox environment by inspecting the transaction.

- Testing complete transactions

  Verify that your app completes transactions properly by confirming that any downloadable purchases are present on your test device.

See Also

In-App Purchase Testing

- 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.

- 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.