

[StoreKit](#) / [Product](#) / `purchase(options:)`

## Instance Method

# purchase(options:)

Initiates a purchase for the product with the App Store and displays the confirmation sheet.

iOS 15.0+ | iPadOS 15.0+ | macOS 12.0+ | tvOS 15.0+ | watchOS 8.0+

```
@MainActor
func purchase(options: Set<Product.Purchase
Option> = []) async throws -> Product.PurchaseResult
```

## Parameters





### options

A set of options you can associate with the purchase.

## Return Value

Returns a [Product.PurchaseResult](#).

## Mentioned in

-  Supporting promoted In-App Purchases in your app
-  Merchandising win-back offers in your app
-  Sending Advanced Commerce API requests from your app
-  Supporting win-back offers in your app

## Discussion

StoreKit provides several APIs you can use to enable customers to initiate a purchase. Before using `purchase(options:)` consider the following APIs and choose the one that best suits your app's implementation:

- Use `PurchaseAction` for apps that use `SwiftUI` on any platform, including multi-scene apps for visionOS.
- Use `purchase(confirmIn:options:)` for apps that use `UIKit`.
- Use `purchase(confirmIn:options:)` for apps that run on macOS and use `AppKit`.
- Use `purchase(options:)` for apps that runs on watchOS.

### Important

If you use StoreKit views such as `ProductView`, `StoreView`, or `SubscriptionStoreView` you don't need to call any other API to initiate a purchase. StoreKit manages the purchase action automatically, including presenting the purchase confirmation UI. For more information, see [StoreKit views](#).

## Use the purchase API

Call the `purchase(options:)` method when a customer initiates a purchase, either within your app or after selecting a promoted in-app purchase on the App Store. This method brings up the system-confirmation sheet. The user can confirm to complete the transaction or cancel it.

Include the purchase options to provide additional information about the purchase, such as:

- `appAccountToken(_ :)` to associate the purchase with the resulting transaction
- `promotionalOffer(offerID:keyID:nonce:signature:timestamp:)`, if the customer is redeeming a promotional offer for an auto-renewable subscription
- `quantity(_ :)`, if the customer is purchasing more than one of the product

The following example illustrates calling `purchase(options:)` using the `options` parameter to provide an app account token:

```
let appAccountToken = <# Generate an app account token. #>
let purchaseResult = try await product.purchase(options: [
    .appAccountToken(appAccountToken)
])
```

If you're testing your app in the sandbox environment, test an Ask to Buy scenario by setting the `simulatesAskToBuyInSandbox(_ :)` purchase option to true. For more information about

Ask to Buy, see [Approve what kids buy with Ask to Buy](#).

This method may throw a [Product.PurchaseError](#) or [StoreKitError](#).

For more information about purchases that users initiate on the App Store, see [Promoting In-App Purchases](#).

---

## See Also

### Purchase requests and results

`struct PurchaseAction`

An action that starts an In-App Purchase.

`enum PurchaseResult`

The result of a purchase.