



Getting Started with Apple Pay

Version 1.0

September 2014

Overview

Apple Pay provides an easy and secure way for users to buy physical goods and services in your app. Using Touch ID, users can authorize payments using credit and debit card payment credentials that are stored on iPhone 6 and iPhone 6 Plus. These models contain a Secure Element, isolating card payment credentials from the main processor where your app runs.

Apple Pay or In-App Purchase

It is important to understand the difference between Apple Pay and In-App Purchase. Use Apple Pay to sell physical goods such as groceries, clothing, and appliances. Also use Apple Pay for services such as club memberships, hotel reservations, and tickets for events. On the other hand, use In-App Purchase to sell virtual goods such as premium content for your app, and subscriptions for digital content.

The [PassKit](#) framework provides APIs for Apple Pay. The [StoreKit](#) framework provides APIs for In-App Purchase.

Prerequisites

In addition to implementing Apple Pay with the PassKit framework, you must:

- Set up an account with a payment processor or gateway, if you don't already have one
- Register a Merchant Identifier via [Certificates, Identifiers & Profiles](#)
- Submit a [Certificate Signing Request](#) to obtain Public and Private keys that will be used to encrypt and decrypt Payment Tokens
- Include an Apple Pay entitlement in your app.

App Review Guidelines

Your app must comply with the requirements specified in Section 29 of the [App Review Guidelines](#).

Payment Providers

You can find a list of payment providers who support Apple Pay with their SDKs on developer.apple.com/apple-pay/. Using one of these SDKs is highly recommended. Contact your payment provider for more information.

The alternative is to provide your own server-side solution to receive payments from your app, decrypt payment tokens and interface with the payment provider. Handling credit and debit card payments can be complicated and unless you already have the expertise and systems in place, an SDK from a payment provider is the quickest and most reliable way to support Apple Pay in your app.

Presenting the Apple Pay button

PassKit provides the APIs that your app will use to determine if it is running on a device with a Secure Element and if the device has been provisioned with payment cards that you support.

Placing the Apple Pay button in your app must be done in accordance with the [Apple Pay Identity Guidelines](#)

Presenting the Payment Sheet

When your user selects goods or services to buy, and selects Apple Pay as the payment method, you create a payment request and ask PassKit to present the payment sheet to the user. See Figure 1.

Your app specifies the contents of the payment sheet but it does not control the user's interaction with the sheet. You must decide if it makes sense to present shipping and billing information, shipping method, and other line items to the user.

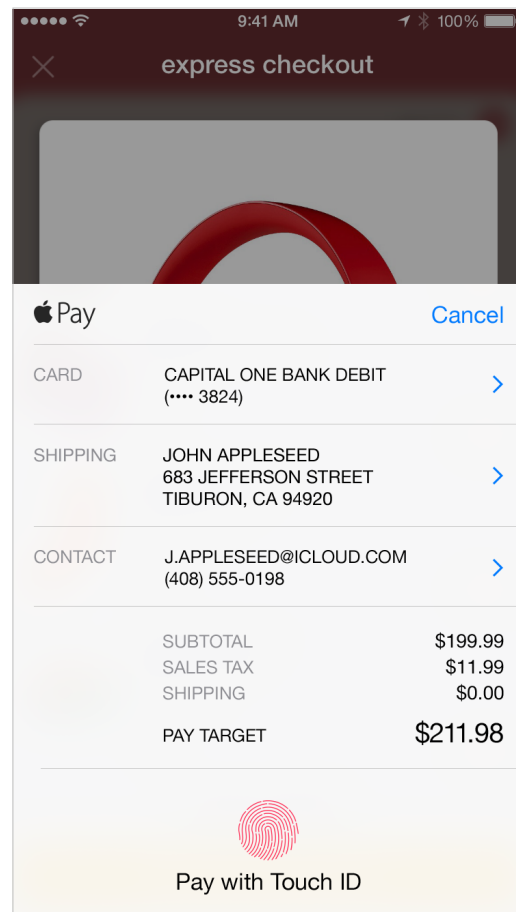


Figure 1: Payment Sheet

The Payment Token

Once authorized by the user with Touch ID, your app receives a payment token from PassKit. The payment token encapsulates the information needed to complete a payment transaction. It includes a cryptogram, unique to the specific purchase, that can be decrypted with your private key or when the payment information is transmitted to a payment processor's server that has your private key.

Figure 2 illustrates a typical payment flow. First the app checks that it can offer Apple Pay as a payment method. In this example, the app needs the postal code from the selected shipping address to calculate shipping cost and update the total amount due. When the user authorizes payment, your app receives a payment token from the Secure Element, via PassKit.

Finally the app calls appropriate APIs in the payment processor SDK to pass the payment information to the payment processor, they process the transaction.

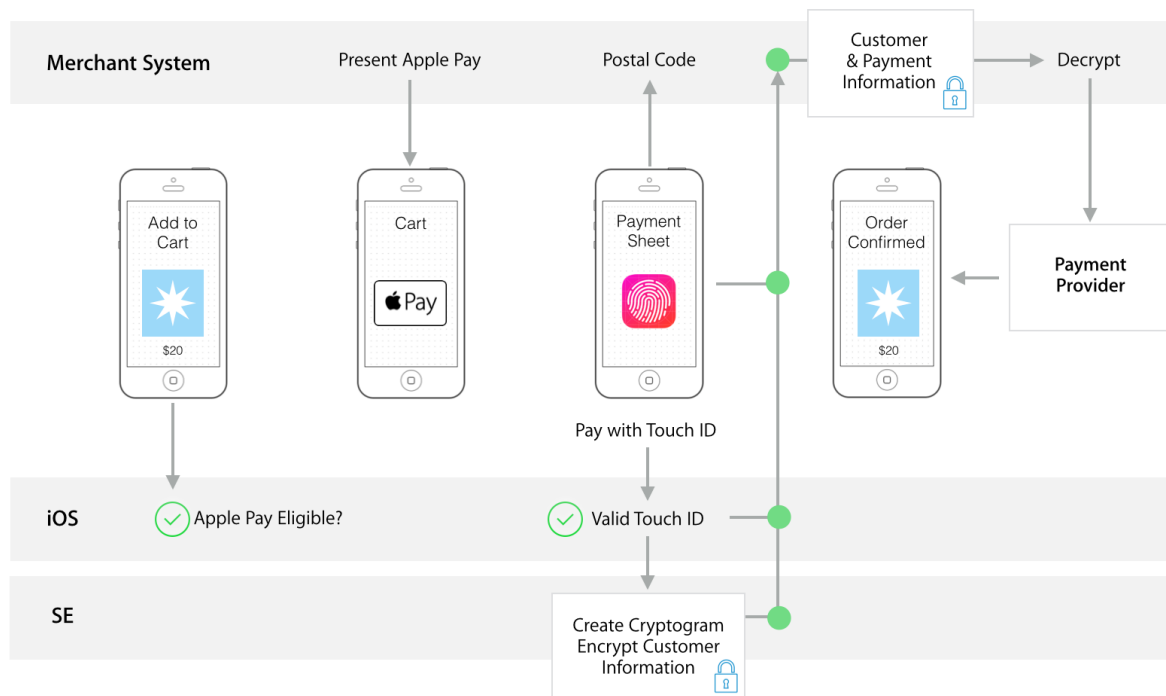


Figure 2: Payment Flow

Supported Transaction Types

Payment Processor SDKs that support Apple Pay are required to handle the following kinds of eCommerce transactions.

Authorizations	To reserve funds on a customer's account
Capture	To transfer money to your bank account once an order is successfully completed
Partial shipment	To divide a purchase into multiple payments for goods that are not shipped together
Recurring	To handle repeating payments for services like a monthly gym membership
Refunds	To return money to a customer's account
Chargeback	To handle fraudulent or disputed transactions

Common Questions and Answers

Which Payment Providers support this service?

For a list of payment provider please visit <https://developer.apple.com/apple-pay/>

Which payment networks are supported?

Visa, MasterCard, and American Express

Which card types are supported?

Both credit and debit cards from the major issuing banks are supported.

How much does it cost to accept Apple Pay?

Apple does not charge users, merchants or developers to use Apple Pay for payments. Your credit and debit transactions will continue to be handled by the payment networks.