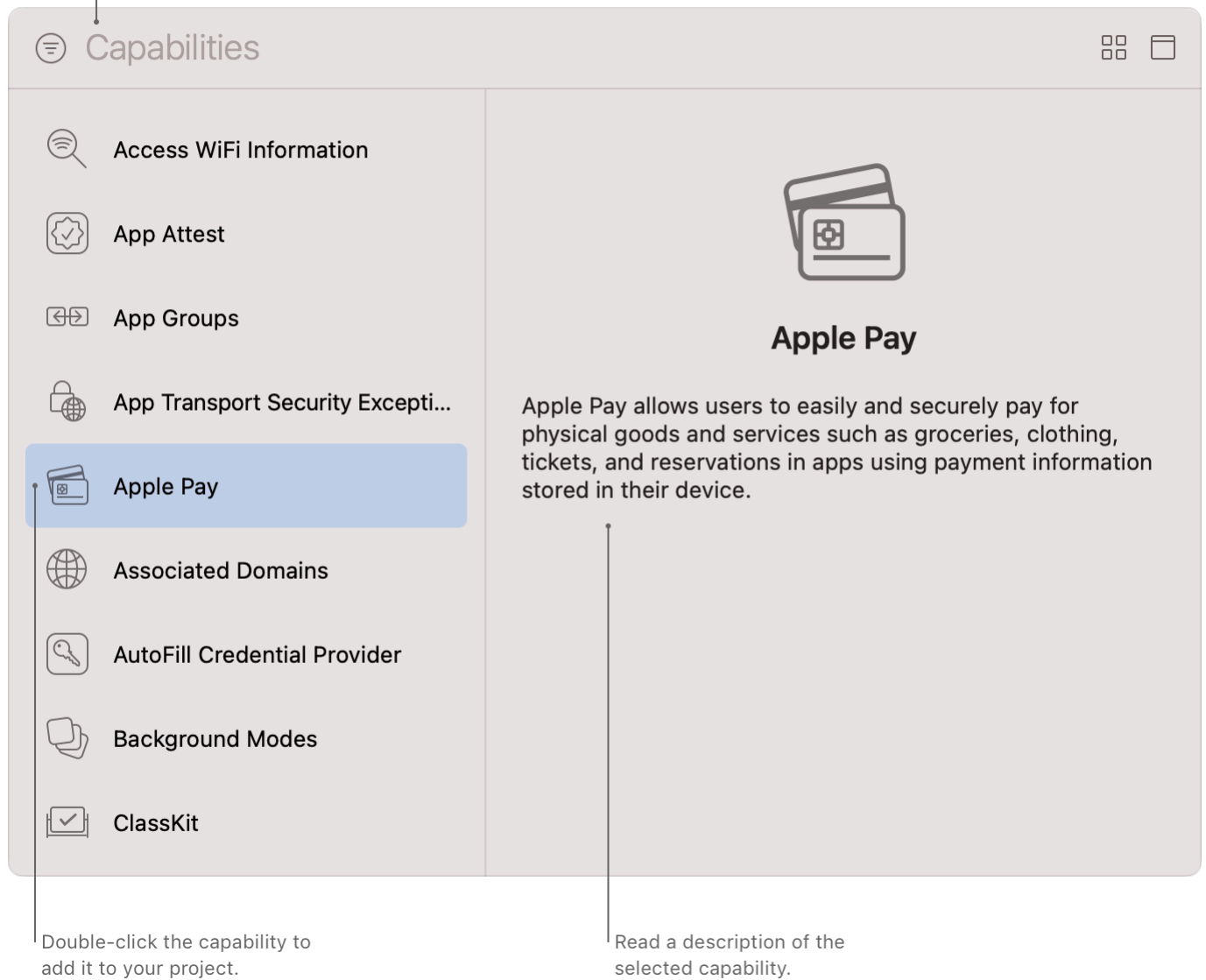# Capabilities

Enable services that Apple provides, such as In-App Purchase, Push Notifications, Apple Pay, iCloud, and many others.

Overview

Capabilities simplify the configuration process for many of Apple's services, some of which require you to configure specific `Entitlements` or change your app's provisioning profile. When you add a capability to an app or other target in your project, Xcode automatically configures that target to use the corresponding service. For example, Xcode might add a required entitlement to a new entitlements file and configure the project to use those entitlements. When Xcode needs you to provide additional information, it presents a simplified UI for you to specify that information.

⊜ Capabilities   ⊞ ☐

🔍 Access WiFi Information

☑ App Attest

⟷ App Groups

🔒 App Transport Security Excepti...

💳 Apple Pay

🌐 Associated Domains

🔑 AutoFill Credential Provider

⬚ Background Modes

☑ ClassKit

Apple Pay

Apple Pay allows users to easily and securely pay for physical goods and services such as groceries, clothing, tickets, and reservations in apps using payment information stored in their device.

Note

You add most capabilities directly from Xcode, but some app services — such as Game Center and In-App Purchase — require additional setup in App Store Connect and your developer account. For more information, see the documentation below for the specific capability.

Xcode automatically manages your target's entitlements file based on the capabilities you add to that target. If you need to edit the file manually, see Editing property list files.

Topics

Essentials

📄 Adding capabilities to your app

Configure your target to include and customize capabilities that provide access to Apple's app services.

App execution

Configuring background execution modes

Indicate the background services your app requires to continue executing in the background in iOS, iPadOS, tvOS, visionOS, and watchOS.

Configuring custom fonts

Register your app as a provider or consumer of systemwide custom fonts.

Configuring game controllers

Enhance gameplay input by enabling the discovery, configuration, and use of physical game controllers.

Configuring Maps support

Register your iOS routing app to provide point-to-point directions to Maps and other apps.

Configuring Siri support

Enable your app and its Intents extension to resolve, confirm, and handle user-driven Siri requests for your app's services.

Commerce

Configuring Apple Pay support

Process payments in your app using the payment information the user stores on their device.

Configuring Sign in with Apple support

Allow users to create an account and sign in to your app with their Apple Account.

Configuring Wallet support

Access the user's Wallet to add, update, and display your app's passes.

Data management

Configuring an associated domain

Create a two-way association between your app and your website to enable universal links, Handoff, App Clips, and shared web credentials.

Configuring app groups

Enable communication and data sharing between multiple installed apps created by the same developer.

📄 Configuring iCloud services

Share user or app data among multiple instances of your app running on different devices.

Network

📄 Configuring network extensions

Customize the various capabilities of your app's networking stack, such as proxying DNS queries or creating packet tunnels.

📄 Registering your app with APNs

Communicate with Apple Push Notification service (APNs) and receive a unique device token that identifies your app.

📄 Configuring Group Activities

Leverage FaceTime infrastructure to create coordinated experiences users can share.

📄 Configuring media device discovery

Add a third-party media device or protocol as a streaming option in the same system menu as AirPlay.

Security

📄 Configuring the hardened runtime

Protect the runtime integrity of your macOS app by restricting access to sensitive resources and preventing common exploits.

📄 Configuring the macOS App Sandbox

Protect system resources and user data from compromised apps by restricting access to the file system, network connections, and more.

📄 Configuring keychain sharing

Share keychain items between multiple apps belonging to the same developer.

📄 Protecting local app data using containers on macOS

Secure your app's local storage data from unauthorized access and modification.

📄 Accessing app group containers in your existing macOS app

Ensure your app has app group container entitlements and macOS can authorize them.

User data

📄 **Configuring HealthKit access**

Read and write health and activity data in the Health app.

📄 **Configuring HomeKit access**

Discover compatible accessories and communicate with configured accessories and services to perform actions.

See Also

Xcode IDE

☰ **Projects and workspaces**

Manage the code and resources you use to build apps, libraries, and other software for Apple platforms.

☰ **Source control management**

Back up your files, collaborate with others, and tag your releases with Git source control support in Xcode.

☰ **Build system**

Compile your code into a binary format, and customize your project settings to build your code.

☰ **Command-line tools**

Develop and customize your projects in Terminal.