Framework

FinanceKit

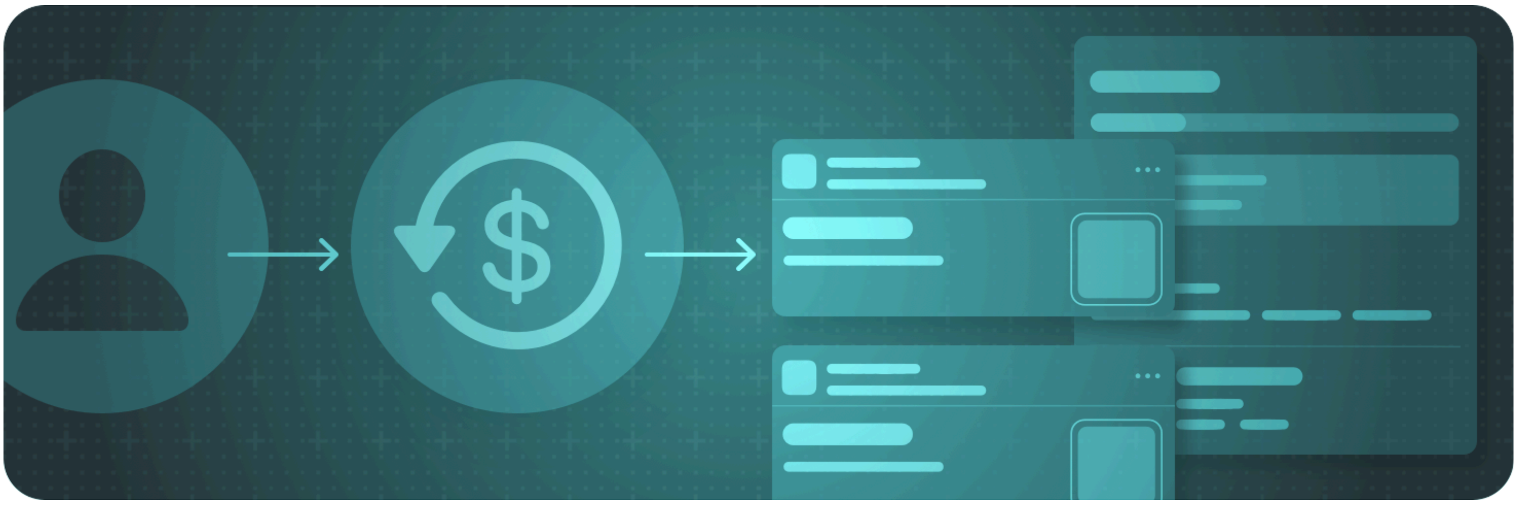
Access financial data and interact with Apple Card, Apple Cash, and orders in Wallet.

iOS 17.0+ | iPadOS 17.0+ | Mac Catalyst 17.0+

Overview

Use FinanceKit to access on-device financial data, Apple Cash, and interact with orders in Apple Wallet.

You interact with FinanceKit through `FinanceStore` and FinanceKitUI, which presents a standardized UI that people access.



Using the FinanceKit framework, you can:

- Access on-device financial data.
- Add, update, store, and interact with orders in Wallet.
- Interact with Apple Card and Apple Cash.

Important

To access someone's financial data, you must meet the criteria outlined in Get started with FinanceKit, request the FinanceKit managed entitlement, hold an organization-level Apple Developer account, be logged in as Account Holder, and include the `NSFinancialDataUsageDescription` string in your `Info.plist`. Apple reviews each application using defined criteria. If your request meets the criteria, Apple adds the entitlement to your developer account by using managed capabilities. To request access, see the FinanceKit managed entitlement request form. For more information about managed entitlements, see Provisioning with capabilities.

Topics

Essentials

{} Implementing a background delivery extension

Receive up-to-date financial data in your app and its extensions by adding a background delivery extension.

FinanceKit updates

Learn more about changes to FinanceKit.

Data storage

`class FinanceStore`

Secure storage for Apple Wallet orders.

Authorization

`func authorizationStatus() async throws -> AuthorizationStatus`

Checks the authorization status for the calling application.

`func requestAuthorization() async throws -> AuthorizationStatus`

Prompts a person to give FinanceKit authorization to access financial data.

`enum AuthorizationStatus`

Accounts

`func accounts(query: AccountQuery) async throws -> [Account]`

Returns a list of accounts a person added to their Wallet that meet the criteria in the provided account query.

`func accountHistory(since: FinanceStore.HistoryToken?, isMonitoring: Bool) -> FinanceStore.History<Account>`

Returns a list of accounts a person added since a time specified by the provided financial history token.

`struct AssetAccount`

A structure that describes the characteristics of an asset account.

`struct LiabilityAccount`

A structure that describes the characteristics of a liability account.

`enum Account`

A structure that describes a financial account.

Balances

`func accountBalances(query: AccountBalanceQuery) async throws -> [AccountBalance]`

Returns a list of balances that meet the criteria in the provided account query.

`func accountBalanceHistory(forAccountID: UUID, since: FinanceStore.HistoryToken?, isMonitoring: Bool) -> FinanceStore.History<AccountBalance>`

Returns the account balance history since a time specified by the provided financial history token.

`struct AccountBalance`

A structure that describes the financial balance of an account at a specific point in time. The financial balance of an account at a specific point in time.

`struct AccountBalanceQuery`

A structure that defines an account balance query.

`struct Balance`

A structure that describes an account balance.

`enum CreditDebitIndicator`

Values that the framework uses to describe transactions as credits or debits.

`enum CurrentBalance`

Values that describe the state of an account's credit balance.

Orders

`struct FullyQualifiedOrderIdentifier`

A structure that specifies the characteristics of an order.

`func saveOrder(signedArchive: Data) async throws -> FinanceStore.SaveOrderResult`

Adds an order to the store or updates an existing order.

Transactions

`func transactionHistory(forAccountID: UUID, since: FinanceStore.HistoryToken?, isMonitoring: Bool) -> FinanceStore.History<Transaction>`

Returns the transactions for the specified account ID, optional starting time, and monitoring indicator for long running transaction queries.

`func transactions(query: TransactionQuery) async throws -> [Transaction]`

Returns transactions that match the provided transaction query.

`struct AccountQuery`

A structure that defines an account query.

`struct AccountCreditInformation`

A structure that describes the credit information associated with an account.

`struct CurrencyAmount`

A structure that describes a monetary amount and its currency.

`struct Transaction`

A structure that represents a transaction relating to a specific financial account.

`struct TransactionQuery`

A structure that describes the parameters to use for a transaction query.

`enum TransactionType`

Values that describe kinds of transactions.

`enum TransactionStatus`

Values that describe the status of a transaction.

Queries

`struct HistoryToken`

A structure that describes the starting point to use for financial data queries.

Merchant categories

`struct MerchantCategoryCode`

Errors

`enum FinanceError`

Values that describe errors that may occur when accessing financial data.

Protocols

`protocol BackgroundDeliveryExtension`

An extension used to receive updates about changes to data within the finance store.

`protocol BackgroundDeliveryExtensionProviding`