

[FinanceKit](#) / [FinanceStore](#)

Class

FinanceStore

Secure storage for Apple Wallet orders.

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

```
class FinanceStore
```

Topics

Retrieving the shared instance

```
static let shared: FinanceStore
```

The shared instance of the store that you make calls through.

Determining data availability

```
static func isDataAvailable(FinanceStore.DataType) -> Bool
```

Returns a Boolean value that indicates if data that represents the provided type is available in the finance store.

Checking authorization status and requesting 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.

Finding accounts

```
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.

```
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.

Getting 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.

Searching for a specific order

```
func containsOrder(matching: FullyQualifiedOrderIdentifier, updatedDate: Date?) async throws -> FinanceStore.ContainsOrderResult
```

Checks whether the finance store contains an order.

Saving or updating orders

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

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

Monitoring 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.

Enumerations

```
enum ContainsOrderResult
```

Result type for queries against the finance store's orders.

```
enum DataType
```

Values that describe the kinds of data in the finance store.

```
enum SaveOrderResult
```

Result type for the finance store's save order method.

```
enum BackgroundDataType
```

Types of data in the finance store supported by background delivery.

```
enum UpdateFrequency
```

Frequencies of background delivery updates.

Structures

```
struct Changes
```

A structure that records changes to the finance store.

```
struct History
```

A structure the framework uses to collect and iterate over finance store model objects.

```
struct HistoryToken
```

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

Instance Methods

```
func disableAllBackgroundDelivery()
```

Disables background delivery for all data types.

```
func disableBackgroundDelivery(for: [FinanceStore.BackgroundDataType])
```

Disables background delivery for the specified types.

```
func enableBackgroundDelivery(for: [FinanceStore.BackgroundDataType],  
frequency: FinanceStore.UpdateFrequency)
```

Enables background delivery for the specified types and frequency.

Relationships

Conforms To

Sendable, SendableMetatype