Structure

HistoryDescriptor

A type that describes the criteria, and, optionally, sort order, to use when fetching history data

iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 18.0+ | macOS 15.0+ | tvOS 18.0+ | visionOS 2.0+ | watchOS 11.0+ | Swift 5.9+

```
struct HistoryDescriptor<TransactionType> where TransactionType : History
Transaction
```

Topics

Creating a descriptor

`init(predicate: Predicate<TransactionType>?)`

Initializes a new history descriptor with the provided predicate.

`init(predicate: Predicate<TransactionType>?, sortBy: [SortDescriptor<TransactionType>])`

Initializes a new history descriptor with the provided predicate and sort descriptor.

Getting the descriptor configuration

`var fetchLimit: UInt64`

The maximum number of transactions to retrieve from the model store's history.

`var predicate: Predicate<TransactionType>?`

The predicate used to initialize the history descriptor.

`var sortBy: [SortDescriptor<TransactionType>]`

The sort descriptor to use to sort the returned history data.

See Also

Model life cycle

`class ModelContainer`

An object that manages an app's schema and model storage configuration.

`class ModelContext`

An object that enables you to fetch, insert, and delete models, and save any changes to disk.

Fetching and filtering time-based model changes

Track all inserts, updates, and deletes that occur in a data store and process them as a series of chronological transactions.

Deleting persistent data from your app

Explore different ways to use SwiftData to delete persistent data.

Reverting data changes using the undo manager

Automatically record data change operations that people perform in your SwiftUI app, and let them undo and redo those changes.

Syncing model data across a person's devices

Add the required capabilities and define a compatible schema to enable SwiftData to automatically sync your app's model data using iCloud.

Concurrency support

Types you use to access model attributes and perform storage-related tasks in a safe and isolated way.