

[SwiftData](#) / DataStore

Protocol

# DataStore

An interface that enables SwiftData to read and write model data without knowledge of the underlying storage mechanism.

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+

```
protocol DataStore : AnyObject
```

## Topics

### Creating a data store

```
init(Self.Configuration, migrationPlan: (any SchemaMigrationPlan.Type)?) throws
```

Required

### Accessing store information

```
var configuration: Self.Configuration
```

Required

```
associatedtype Configuration : DataStoreConfiguration
```

Required

```
protocol DataStoreConfiguration
```

```
var identifier: String
```

**Required**

```
var schema: Schema
```

**Required**

## Processing fetch requests

```
func fetch<T>(DataStoreFetchRequest<T>) throws -> DataStoreFetchResult<T, Self.Snapshot>
```

**Required**

```
struct DataStoreFetchRequest
```

```
struct DataStoreFetchResult
```

```
associatedtype Snapshot : DataStoreSnapshot
```

**Required**

```
protocol DataStoreSnapshot
```

```
typealias DataStoreSnapshotValue
```

```
func fetchCount<T>(DataStoreFetchRequest<T>) throws -> Int
```

**Required** Default implementation provided.

```
func fetchIdentifiers<T>(DataStoreFetchRequest<T>) throws -> [PersistentIdentifier]
```

**Required** Default implementation provided.

## Persisting model data

```
func save(DataStoreSaveChangesRequest<Self.Snapshot>) throws -> DataStoreSaveChangesResult<Self.Snapshot>
```

**Required**

```
struct DataStoreSaveChangesRequest
```

```
class DataStoreSaveChangesResult
```

## Removing all persisted model data

```
func erase() throws
```

**Required** Default implementation provided.

## Sharing cached data between model contexts

```
func initializeState(for: EditingState)
```

**Required** Default implementation provided.

```
struct EditingState
```

```
func cachedSnapshots(for: [PersistentIdentifier], editingState: EditingState) throws -> [PersistentIdentifier : Self.Snapshot]
```

**Required** Default implementation provided.

```
func invalidateState(for: EditingState)
```

**Required** Default implementation provided.

---

## Relationships

### Inherited By

DataStoreBatching

### Conforming Types

DefaultStore

---

## See Also

### Model storage

{} Maintaining a local copy of server data

Create and update a persistent store to cache read-only network data.

```
class DefaultStore
```

A data store that uses Core Data as its underlying storage mechanism.

```
protocol DataStoreBatching
```

An interface that enables a custom data store to support batch requests.

```
protocol HistoryProviding
```

An interface that enables a custom data store to provide the history of changes for its persisted models.

{} Building a document-based app using SwiftData

Code along with the WWDC presenter to transform an app with SwiftData.

```
struct ModelDocument
```

A document type that uses SwiftData to manage its storage.