

[SwiftData](#) / Concurrency support

API Collection

# Concurrency support

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

## Topics

### Model actors

`macro ModelActor()`

Converts a Swift actor into a model actor by generating boilerplate code that fulfills the requirements of the associated protocol.

`protocol ModelActor`

An interface for providing mutually-exclusive access to the attributes of a conforming model.

### Model executors

`class DefaultSerialModelExecutor`

An object that safely performs storage-related tasks using an isolated model context.

`protocol SerialModelExecutor`

An interface for performing serial storage-related tasks using an isolated model context.

`protocol ModelExecutor`

An interface for performing storage-related tasks using an isolated model context.

# 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.

`struct HistoryDescriptor`

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

`{}` [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.