Core Data / Core Data stack

API Collection

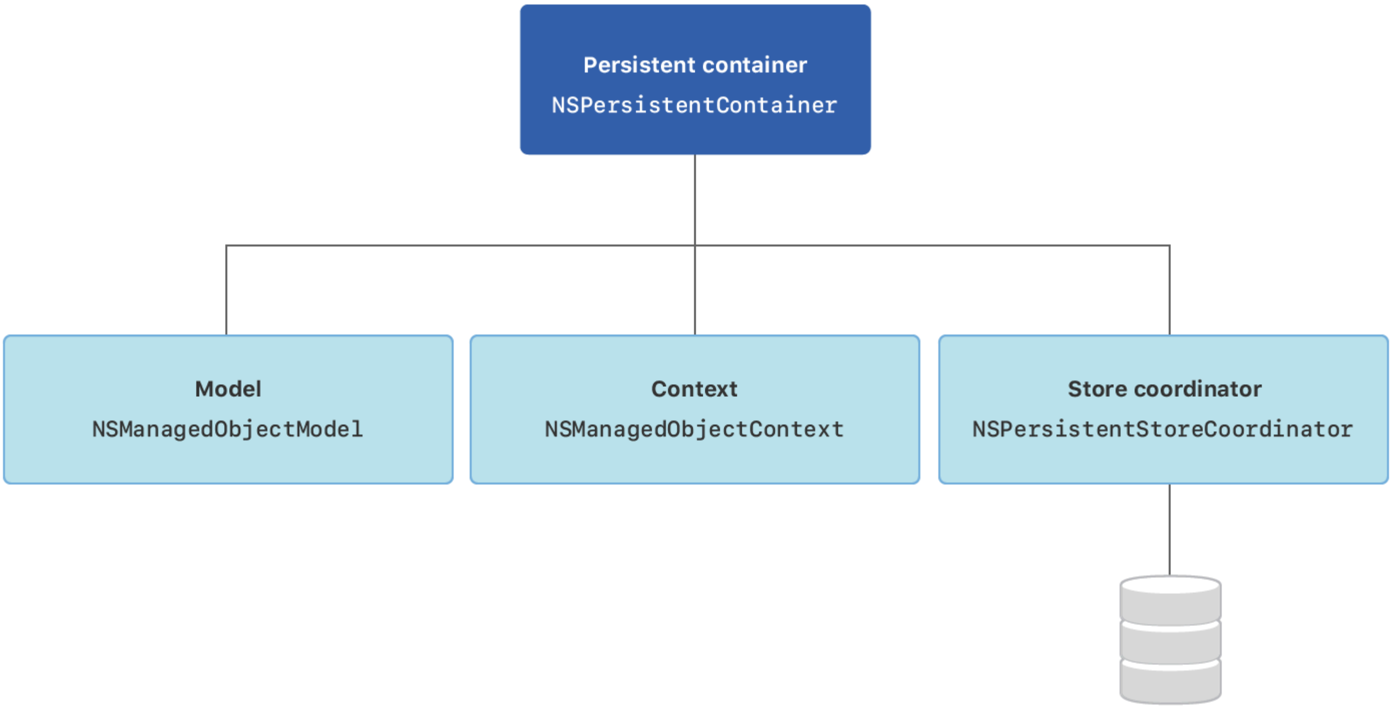
Core Data stack

Manage and persist your app’s model layer.

Overview

Core Data provides a set of classes that collaboratively support your app’s model layer:

- An instance of `NSManagedObjectModel` describes your app’s types, including their properties and relationships.
- An instance of `NSManagedObjectContext` tracks changes to instances of your app’s types.
- An instance of `NSPersistentStoreCoordinator` saves and fetches instances of your app’s types from stores.



You use an `NSPersistentContainer` instance to set up the model, context, and store coordinator simultaneously.

Topics

Stack Setup

`class NSPersistentContainer`

A container that encapsulates the Core Data stack in your app.

Object Modeling

`class NSManagedObjectModel`

A programmatic representation of the `.xcdatamodeld` file describing your objects.

`class NSEntityDescription`

A description of a Core Data entity.

`class NSPropertyDescription`

A description of a single property belonging to an entity.

`class NSAttributeDescription`

A description of a single attribute belonging to an entity.

`class NSDerivedAttributeDescription`

A description of an attribute that derives its value by performing a calculation on a related attribute.

`class NSRelationshipDescription`

A description of a relationship between two entities.

Object Management

`class NSManagedObjectContext`

An object space to manipulate and track changes to managed objects.

`class NSManagedObject`

The base class that all Core Data model objects inherit from.

`class NSManagedObjectID`

A compact, universal identifier for a managed object.

Store Coordination

`class NSPersistentStoreCoordinator`

An object that enables an app’s contexts and the underlying persistent stores to work together.

`class NSPersistentStore`

The abstract base class for all Core Data persistent stores.

`class NSPersistentStoreDescription`

A description object used to create and load a persistent store.

`class NSPersistentStoreRequest`

Criteria used to retrieve data from or save data to a persistent store.

`class NSPersistentStoreResult`

The abstract base class for results returned from a persistent store coordinator.

`class NSPersistentStoreAsynchronousResult`

A concrete class used to represent the results of an asynchronous request.

`class NSSaveChangesRequest`

An encapsulation of a collection of changes to be made by an object store in response to a save operation on a managed object context.

`class NSAtomicStore`

An abstract superclass that you subclass to create a Core Data atomic store.

`class NSAtomicStoreCacheNode`

A concrete class that you use to represent basic nodes in a Core Data atomic store.

`class NSIncrementalStore`

An abstract superclass defining the API through which Core Data communicates with a store.

`class NSIncrementalStoreNode`

A concrete class used to represent basic nodes in a Core Data incremental store.

See Also

Essentials

Creating a Core Data model

Define your app’s object structure with a data model file.

Setting up a Core Data stack

Set up the classes that manage and persist your app’s objects.

Handling Different Data Types in Core Data

Create, store, and present records for a variety of data types.

Linking Data Between Two Core Data Stores

Organize data in two different stores and implement a link between them.