

Core Data

Core Data 是一种在 Swift 和 iOS 应用程序之间广泛使用的持久化框架。它允许开发者将应用程序的数据模型（对象）保存到本地持久存储中，如 SQLite 数据库，然后在需要时从本地加载该数据。

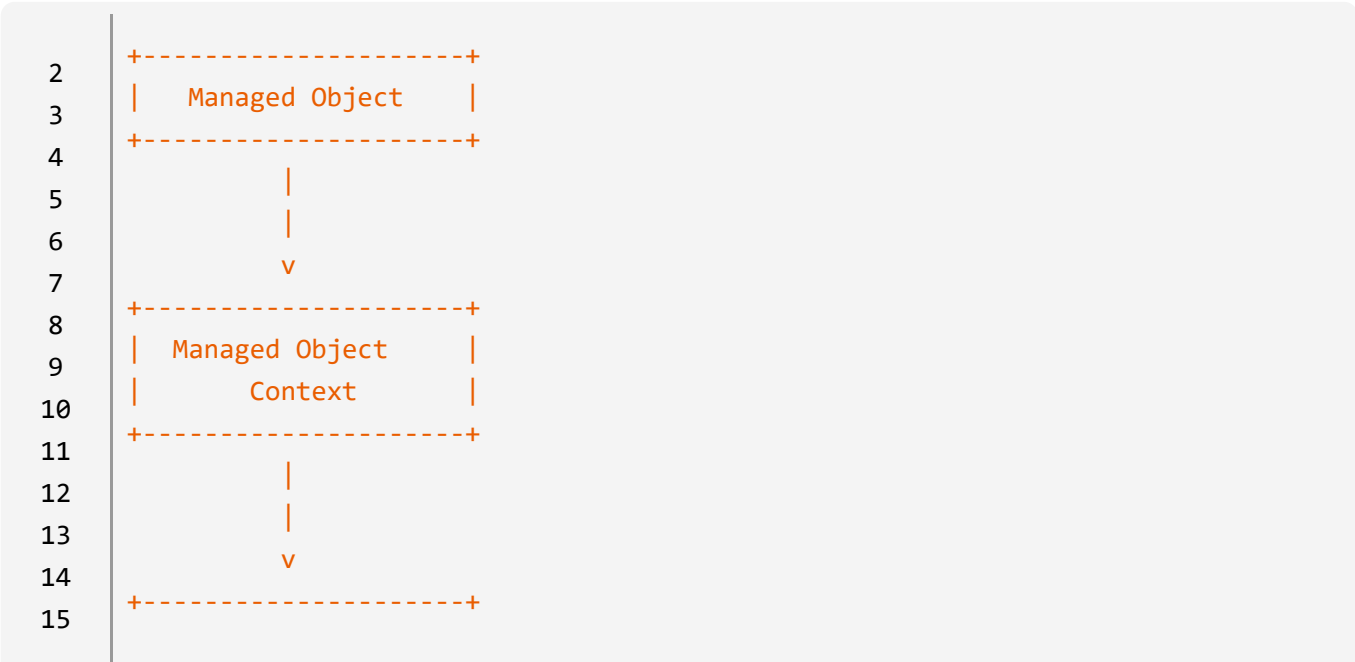
Core Data 可以处理许多应用程序所需的数据管理任务，包括创建、读取、更新和删除数据记录。

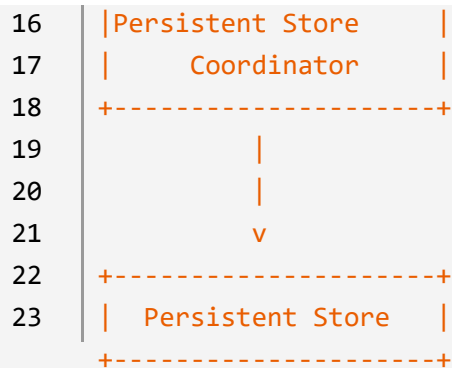
在 Swift 编程中，通过操作托管对象上下文和用预先定义的数据模型创建托管对象实例，Core Data 能帮助你轻松管理应用程序的数据，并确保数据与本地存储保持同步。这意味着当用户退出应用程序时，他们之前的数据和配置将得到保留，下次使用应用程序时会加载这些数据。

关联数据库的概念来看，Core Data 的 Entity 與 Attribute 大約可以比對到 Table 和 Field

Core Data 是一个完整的持久化框架，由多个组件组成，这些组件一起协作来管理应用程序的数据模型。以下是 Core Data 的主要组件：

- 1. Managed Object Model（托管对象模型）：定义了数据模型中的实体、属性和关系，它是Core Data的核心部分。
- 2. Managed Object（托管对象）：是实体在应用程序中的表示，开发人员可以通过Managed Object对实体的属性和关系进行操作。
- 3. Managed Object Context（托管对象上下文）：管理托管对象的生命周期、状态变化以及持久化操作。它是应用程序与数据存储器之间的桥梁。
- 4. Persistent Store Coordinator（持久化存储协调器）：管理应用程序的持久化存储，负责将托管对象的数据存储在持久化存储器（通常是SQLite数据库）中。
- 5. Persistent Store（持久化存储器）：实际存储应用程序的数据的地方，它是一个底层的数据存储器，通常是一个SQLite数据库。
- 6. Fetch Request（获取请求）：用于检索托管对象的数据的查询对象，可以使用它来执行复杂的数据查询操作。
- 7. Fetched Results Controller（获取结果控制器）：将Fetch Request的结果转换为可供用户界面显示的数据。





在这个示意图中，箭头表示组件之间的关系：

1. Managed Object 是托管对象模型中定义的实体在应用程序中的表示，它们由 Managed Object Context 进行管理和操作
2. Managed Object Context 是应用程序与数据存储器之间的桥梁，负责管理托管对象的生命周期、状态变化以及持久化操作
3. Persistent Store Coordinator 管理应用程序的持久化存储，并负责将托管对象的数据存储在持久化存储器中，通常是一个 SQLite 数据库
4. Persistent Store 是实际存储应用程序数据的地方，它是一个底层的数据存储器
5. FetchRequest 是用于检索托管对象的数据的查询对象，Fetched Results Controller 将 Fetch Request 的结果转换为可供用户界面显示的数据

Q: NSPersistentContainer 是什么？

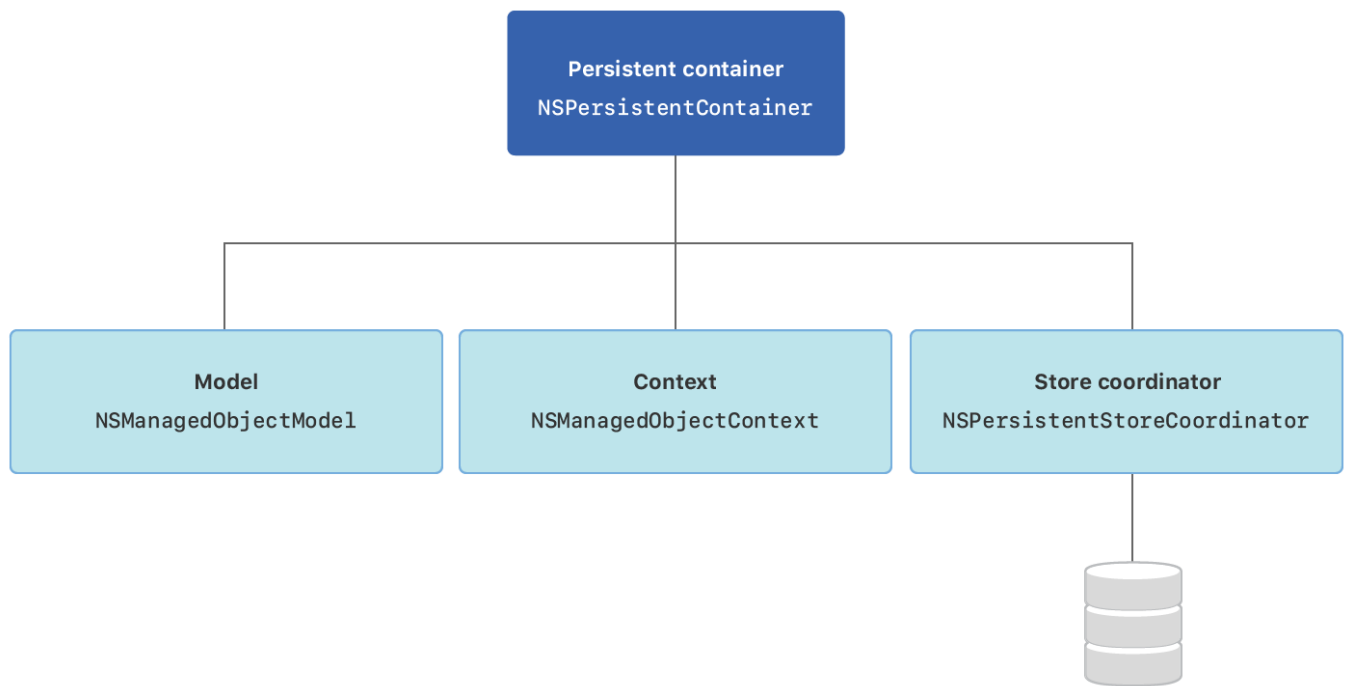
NSPersistentContainer 是 Core Data 中新增的一个类，它是一个高级别的对象，用于管理托管对象模型、托管对象上下文和持久化存储器之间的关系。

NSPersistentContainer 简化了 Core Data 的配置和使用，它将多个组件封装在一个对象中，简化了 Core Data 的初始化过程。

使用 NSPersistentContainer，开发人员无需手动配置托管对象模型、持久化存储器和持久化存储协调器等组件，它们都被自动创建和配置。NSPersistentContainer 还提供了方便的 API，使开发人员可以轻松地进行常见的数据操作，例如创建托管对象、查询数据、保存数据等。

NSPersistentContainer 还提供了多线程支持，使得多个线程可以同时访问 Core Data 的组件。它为每个线程提供了一个独立的托管对象上下文，每个上下文都使用相同的持久化存储器和持久化存储协调器。这使得开发人员可以轻松实现多线程数据操作，提高了应用程序的性能和响应速度。

总的来说，NSPersistentContainer 简化了 Core Data 的配置和使用，并提供了多线程支持，使得开发人员可以更加方便地管理和操作应用程序的数据模型。



You use an [NSPersistentContainer](#) instance to set up the model, context, and store coordinator simultaneously.
