DATAART®
Enjoy IT

Core Data: есть ли жизнь за пределами главного потока?

Что дает Core Data

- Поддержка связей между объектами (relationships);
- Отложенный (ленивый) доступ к данным для снижения потребления памяти;
- NSFetchedResultsController для интеграции с UI;
- KVC & KVO;
- Высокоуровневый язык запросов с сортировками и фильтрами;
- Слияние конфликтующих изменений;
- Отслеживание изменений; Undo/Redo;
- Миграция и версионность;
- Валидация

Альтернативы

- NSUserDefaults;
- Property lists;
- NSCoding protocol;
- SQLite + third-party libraries (например, SQLite Persistent Objects by Jeff Lamarche);

Основные классы Core Data

- NSPersistentStoreCoordinator;
- NSManagedObjectModel;
- NSManagedObjectContext;
- NSManagedObject;

Параллельное выполнение. Зачем?

Организация concurrency в Apple iOS

- POSIX threads;
- NSThread;
- Grand Central Dispatch (GCD);
- Методы NSObject;
- Operation queues

Пример работы с хранилищем Core Data

- iOS устройство (клиент) взаимодействует с сервером для получения данных;
- Данные локально кэшируются при помощи Core Data.

Пример 1: single thread

- Самое простое решение;
- Высокая скорость реализации решения;
- Появляются задержки реагирования UI;
- Во избежание неприятных задержек UI «красиво» блокируется;

Пример 2: single writer & multiple readers

- UI работает без задержек;
- Необходимо в любой момент и из любого потока работать с актуальными данными;

Решение: https://bitbucket.org/dmakarenko/cdoperation

- (id) init: (NSManagedObjectContext*)mainContext

- запомнить поток вызова init (initialThread);
- подписаться на уведомления главного контекста (mainContext);
- при получении нотификации добавить ее в список полученных;

- (void) main

- создать собственный локальный контекст;
- подписаться на нотификации локального контекста;
- при получении нотификации выполнить слияние в потоке initialThread;
- выполнить основной код операции;
- провести слияние всех изменений главного контекста в контекст операции

Пример 3: multiple writers

- Почти нет отличий от предыдущего примера;
- Также используется CDOperation, без доработок;
- В нескольких NSManagedObjectContext’ах могут конфликтовать изменения.

Merge policies

- NSErrorMergePolicy;
- NSMergeByPropertyStoreTrumpMergePolicy;
- NSMergeByPropertyObjectTrumpMergePolicy;
- NSOverwriteMergePolicy;
- NSRollbackMergePolicy;

Простейшие оптимизации

- использование нескольких NSPersistentStoreCoordinator’ов для независимых хранилищ;
- оптимизация NSFetchRequest’ов и выполнение их синхронно;
- уменьшение размеров локального хранилища;
- сокращение частоты вызовов метода **save:** класса NSManagedObjectContext;

Спасибо за внимание

Q&A