УДК 519.004.054

Прозрачная миграция бэкенда для клиентских приложений.

Р. И. Маракулин, А. А. Соболев

Московский физико-технический институт (государственный университет)

В настоящее время огромное число приложений используют различные хранилища данных, будь то удаленные сервисы, такие как Parse, iCloud или локальные SQL, NoSQL хранилища. Одни приложения хранят фотографии пользователей, сообщения с комментариями, игровые данные на удаленных серверах. Другие сохраняют списки контактов, заметки, почту локально. Большой спрос на хранилища привел к появлению огромного числа всевозможных механизмов и сервисов, предоставляющих свои услуги для чтения, записи информации, такие как Dropbox, AppleiCloud, GoogleCloudPlatform, MicrosoftOneDrive, Parse. И у каждого сервиса есть свой API для доступа к данным. Выбрав при проектировании приложения один из них, мы будем вынуждены переписывать огромные участки GUI кода, если захотим сменить его впоследствии. Также, при написании иного приложения, но с той же базой данных, другая команда разработчиков вынуждена изучать API незнакомого сервиса. В данной работе предлагается решение, основанное на API, представленномApple еще с платформой iOS 5.0. Основной целью работы является обеспечение единого интерфейса доступа к данным, вне зависимости от бэкенда с помощью инструмента Apple (NSIncrementalStore API).

В рамках данной работы реализованфрэймворк, который является прослойкой между CoreData API и любым бэкендом. CoreData является удобным фрэймворком для работы с хранимыми на устройстве данными. В данной работе вызовы CoreData используются для обращения к выбранному бэкенду. NSIncrementalStore, являясь абстрактным подклассом NSPersistentStore, позволяет создать свое хранилище, которое будет использоваться стэкомCoreData. Более того, возможно подключение нескольких различных хранилищ в одном приложении.

В качестве примера реализована миграция данных приложения с Parse на CloudKit.

Весь процесс создания приложения теперь имеет две четко разграниченные стадии:

- Создание клиентского приложения, использующего для доступа к данным CoreData API
- Обеспечение поддержки протокола для выбранного хранилища.

Данное решение имеет ряд преимуществ, таких, например, как обеспечение независимости клиенсткого приложения от бэкенда, легкая смена бэкенда, возможность

быстрого прототипирования, при котором готовое приложение создается без использования бэкенда, а подключение его происходит позднее.
Дальнейшая работа заключается в создании универсальных классов хранилищ для конкретных сервисов и возможности простого подключения дополнительных сервисов.

Литература:

1. Matt Thompson About NSIncrementalStore, http://nshipster.com/nsincrementalstore
2. Apple Inc. NSIncrementalStore Class Reference, https://developer.apple.com/library/prerelease/mac/documentation/CoreData/Reference/NSIncrementalStore_Class/index.html
3. Apple Inc. Core Data Programming Guide, https://developer.apple.com/library/watchos/documentation/Cocoa/Conceptual/CoreData/index.html
4. Apple Inc. NSIncrementalStore Programming Guide, https://developer.apple.com/library/mac/documentation/DataManagement/Conceptual/IncrementalStorePG/IncrementalStorePG.pdf