SQLITE – БАЗА ДАННЫХ В МОБИЛЬНОЙ СИСТЕМЕ ANDROID

РЭЧИЛЭ Юлиан, КОРНЕСКУ Роберт

Технический Университет Молдовы

Аннотация: *Статья посвящена анализу системы управления базами данных в мобильной системе Android - SQLite. Показано назначение СУБД, принцип работы, а так же ее использование на примере приложения LittleChef.*

Ключевые слова: *СУБД, SQLite, Android, база данных.*

1. Введение

SQLite — компактная встраиваемая реляционная СУБД (рисунок 1). Исходный код библиотеки передан в общественное достояние. В 2005 году проект получил награду Google-O'Reilly Open Source Awards.

В мобильных приложениях в ряде случаев может быть полезно использование простой внутренней базы данных, такой как SQLite. В этой статье мы рассматриваем Android SQLite и вспомогательные классы для создания и обслуживания баз данных. Мы обсудим создание и использование заранее заполненной базы данных, а также будем использовать SQLite для создания внутренней базы данных для образца ресторанного приложения.

Рис. 1. Логотип СУБД SQLite

2. Обзор SQLite

В Android существует встроенная поддержка базы данных SQLite. Поддерживаются все функции SQLite, предоставляется API оболочки с совместимым интерфейсом.

API Android SQLite является типичным; разработчику следует реализовать всю обработку базы данных, включая создание, управление версиями, обновления базы данных и прочие настройки. Если нужно использовать заранее заполненную базу данных SQLite, требуется дополнительная настройка.

Прямое использование API Android SQLite может привести к написанию большого количества шаблонного кода. Существует несколько библиотек Android, помогающих упростить этот процесс. Они обладают и рядом других возможностей в дополнение к удобному и эффективному использованию баз данных SQLite в приложениях Android. SQLiteAssetHelper - одна из таких библиотек. Эта библиотека популярна в сообществе разработчиков Android.

3. Принцип работы

Слово «встраиваемый» (embedded) означает, что SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения

транзакции записи весь файл, хранящий базу данных, блокируется; ACID-функции достигаются в том числе за счёт создания файла журнала.

Несколько процессов или потоков могут одновременно без каких-либо проблем читать данные из одной базы. Запись в базу можно осуществить только в том случае, если никаких других запросов в данный момент не обслуживается; в противном случае попытка записи оканчивается неудачей, и в программу возвращается код ошибки. Другим вариантом развития событий является автоматическое повторение попыток записи в течение заданного интервала времени.

3. Пример приложения для ресторана – LittleChef

Для демонстрации использования базы данных SQLite и библиотеки *SQLiteAssetHelper* library рассмотрен образец приложения для ресторана - Little Chef (рисунок 2).

Это приложение позволяет пользователю просматривать различные категории меню и выбирать различные элементы.

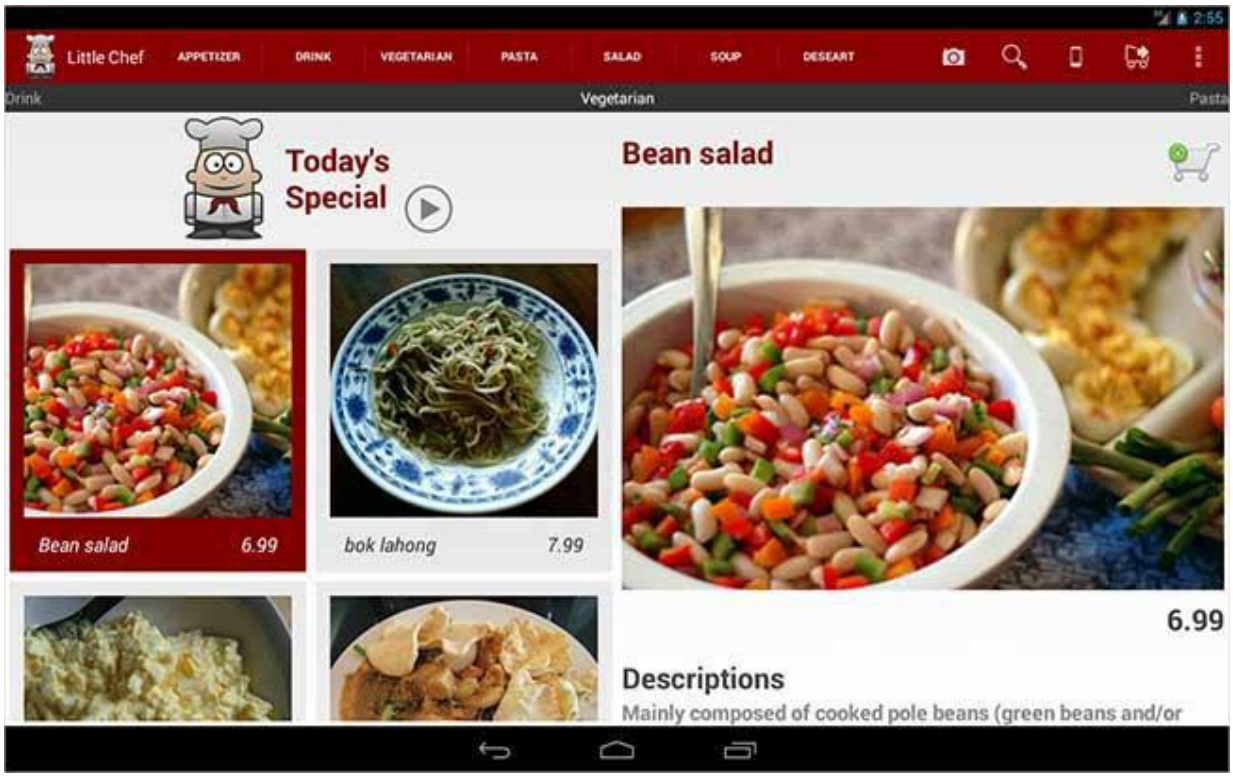

Рис. 2. Пример приложения для ресторана LittleChef

Приложение LittleChef для ресторана может быть использовано поваром или пользователем для просмотра различных категорий и элементов меню. В образце приложения используется жест прокрутки для переключения между категориями, а при выборе какого-либо элемента в меню отображаются подробные сведения.

Для сохранения всех категорий меню и подробных сведений об элементах можно использовать базу данных SQLite. В дальнейших версиях приложения можно расширить базу данных, чтобы поддерживать другие виды данных приложения: данные о продажах, о программах лояльности клиентов, индивидуальные настройки для каждого пользователя и т.д.

4. Доступ к элементам меню базы данных ресторана

При инициализации SQLiteAssetHelper автоматически скопирует заранее заполненную базу данных ресторана из папки assets по соответствующему пути к базе данных. Последующие вызовы будут снова использовать этот экземпляр базы данных до тех пор, пока не будет запрошено обновление. Не рекомендуется обрабатывать доступ к базе данных в главном потоке. Можно использовать SQLiteAssetHelper для добавления дополнительных абстракций, таких как интерфейс поставщика содержимого Android.

В зависимости от требованиях к приложению и сценариев использования можно добавить в ресторанное приложение и другие возможности: поддержку обновления базы данных, управление

версиями и даже поддержку серверов. При серверной реализации внутренней базы данных мы можем использовать локальную базу данных SQLite в качестве временного кеша и для возможности автономной работы.

Заключение

В статье мы обсудили использование СУБД SQLite в приложениях Android. Были рассмотрены API Android SQLite и вспомогательные классы. Был использован образец ресторанного приложения для демонстрации использования заранее заполненных баз данных с библиотекой SQLiteAssetHelper.

Литература

1. SQLITE. [Электронный ресурс]. - Режим доступа: https://ru.wikipedia.org/wiki/SQLite
2. Using Databases.[Электронный ресурс]. - Режим доступа: http://developer.android.com/guide/topics/data/data-storage.html#db
3. Saving Data Using the Room Persistence Library [Электронный ресурс]. - Режим доступа: http://developer.android.com/training/basics/data-storage/databases.html
4. Android-Sqlite-Asset-Helper [Электронный ресурс]. - Режим доступа: https://github.com/jgilfelt/android-sqlite-asset-helper