

ДОСЛІДЖЕННЯ ЛОКАЛЬНИХ БАЗ ДАНИХ ДЛЯ ANDROID ПРИСТРОЇВ. РОЗРОБКА МЕТОДУ ОБРОБКИ ДАНИХ

Вінницький національний технічний університет

Анотація

Для розкриття теми проекту . Був проведений аналіз сучасних фреймворкі та бібліотек для мови програмування Kotlin для вибору найкращого способу зберігання даних локально в випадку відсутності інтернету або способу ртримання даних з хмари на Android пристроях. За результатами було обрано локальну базу даних основану на Dagger2 Hilt room основанна на SQLite.

Ключові слова: Android, Java ,Kotlin, Room, SQL, Dagger2Hilt.

Abstract

To reveal the theme of the project. An analysis of modern frameworks and libraries for the Kotlin programming language was performed to select the best way to store data locally in the absence of the Internet or how to retrieve data from the cloud on Android devices. Based on the results, a local database based on Dagger2 Hilt room based on SQLite was selected.

Keywords: Android, Java ,Kotlin, Room, SQL, Dagger2Hilt.

Смартфони стали невід'ємною частиною нашого повсякденного життя. Щоденно ми використовуємо різні додатки , щоб послухати музику, подивитись фільм чи почитати книгу. Для цього додаткам потрібно обробити та передати дані. Але не завжди у нас є доступ до інтернету для отримання даних з серверів, і щоб зробити додаток який швидко обробить інформації в онлайн режимі і зможе потім використовувати цю інформацію в оффлайні.

Вибір бази даних – це найголовніше у сучасному додатку. Локальна, хмарна або комбінованна реалізація різняться в своїй ціні та часі витрачених на розробку.

Існує три способи роботи з даними в БД:

1) Створити порожню структуру бази даних. Користувач працює з додатком (створює замітки, видаляє їх) і база даних наповнюється. Прикладом може служити додаток NotePad на вашому Android-девайсі.

2) Мати готову БД, наповнену даними, яку потрібно поширювати з додатком, або парсити дані з файлу в кеш пристрою.

3) Отримувати дані з мережі, в міру необхідності.

Room - це новий спосіб зберегти дані додатків в Android-додатку, представлений в 2018 році на Google I/O. Це частина нової Android Architecture, група бібліотек від Google, які підтримують доречну архітектуру додатків. Room пропонується в якості альтернативи Realm, ORMLite, GreenDao і багатьом іншим.

Room - це високорівнева інтерфейс для низькорівневих прив'язок SQLite, вбудованих в Android. Він виконує велику частину своєї роботи під час компіляції, створюючи API-інтерфейс поверх вбудованого SQLite API, що дозволяє йому підтягувати дані з мережі за допомогою API- запроса.

На відміну від більшості Object-Relational Mapping, Room використовує обробник анотації для виконання всієї своєї манери збереження даних. Це означає, що ні класи додатків, ні класи моделей не повинні нічого розширювати в Room, на відміну від багатьох інших Object-Relational Mapping, включаючи Realm і SugarORM. Як можна побачити при помилках з анотаціями Query () вище, також отримуємо можливість перевірки коректності SQL-запитів під час компіляції, що може заощадити багато часу та ресурсів.

Також Room дозволяє спостерігати за змінами даних, інтегруючи їх як з API LiveData Архітектурних Компонентів, так і з RxJava 2. Це означає, що якщо ми створимо складну схему, де зміни в базі даних повинні з'являтися в декількох місцях вашого застосування, Room робить повідомлення про зміни. Це потужне доповнення може бути включено одним рядком. Все, що потрібно зробити, це змінити тип значень, що повертаються. Також Room має обмеженням , що він не буде обробляти відносини з іншими типами сутностей для вас автоматично, як і інші ORM.

Спираючись на стандартні анотації javax.inject (JSR 330), кожен клас легко перевірити. Не потрібна мати купу зразків, щоб просто замінити клас або сервіс

Структури введення залежностей існують протягом багатьох років із цілим рядом API для налаштування та виклику. Dagger 2 є першим, хто реалізує повний стек із згенерованим кодом. Основним принципом є створення коду, що імітує

код, який користувач може писати від руки, щоб гарантувати, що введення залежностей є настільки простим, відстежуваним і продуктивним, наскільки це може бути.

Hilt забезпечує стандартний спосіб включити введення залежності Dagger у програму Android, його цілями є:

Спрощення інфраструктури, пов'язаної з Dagger, для програм Android.

Створення стандартного набору компонентів і областей для полегшення налаштування, читабельності / розуміння та спільного використання коду між програмами.

Забезпечити простий спосіб надати різні прив'язки до різних типів збірки (наприклад, тестування, налагодження чи випуск).

Огляд дизайну рукоятки

Hilt працює, створюючи для вас код налаштування Dagger. Це забирає більшу частину шаблону використання Dagger і насправді просто залишає аспекти визначення того, як створювати об'єкти і куди їх вводити. Hilt генерує компоненти Dagger і код для автоматичного введення ваших класів Android (таких як дії та фрагменти) для вас.

Hilt генерує набір стандартних компонентів Android Dagger на основі вашого перехідного шляху до класу. Для цього потрібно позначити модулі Dagger анотаціями Hilt, щоб сказати Hilt, в який компонент вони повинні входити. Отримання об'єктів у ваших класах фреймворків Android здійснюється за допомогою іншої анотації Hilt, яка генерує код інжекції Dagger у базовий клас, який ви розширите. Для користувачів Gradle розширення цього класу здійснюється за допомогою перетворення байт-коду під капотом.

Розглянувши різні засоби для локальної обробки та збереження даних, а також для створення коректної бази даних яким могла зв'язуватись з хмарою, було обрано Room. Для вирішення поставленої задачі узгоджено та застосовано такі можливості: Dagger2, Hilt, Room. Саме це дозволить додатку бути надзвичайно швидким та надійним у використанні, а також здешевить собівартість.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Android Programming In Kotlin: Layouts. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <https://www.i-programmer.info/programming/android/13260-androidprogramming-in-kotlin-layouts.html?start=2>

2. **ORM** (Object-Relational Mapping). [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <https://ru.wikipedia.org/wiki/ORM>
3. Dagger Hilt. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <https://dagger.dev/hilt/>
4. Локальные базы данных в Android: выбор лучшего для вашего проекта. [Електронний ресурс] : [Веб-сайт] – Електронні дані. - Режим доступу: <https://proandroiddev.com/cache-data-in-android-2e4042b07df35>

Козубський Владислав Володимирович – студент групи 1КІ-20м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: kozubzkiy@gmail.com

Науковий керівник: Ткаченко Олександр Миколайович – к.т.н, доцент, відповідальний за наукову роботу на кафедрі обчислювальної техніки, Вінницький національний технічний університет, Вінниця, e-mail: alextk1960@gmail.com

Kozubskiy Vlad - student of the group 1KI-20m, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: kozubzkiy@gmail.com

Scientific adviser: Alexander Tkachenko - Ph.D., Associate Professor, responsible for scientific work at the Department of Computer Engineering, Vinnitsa National Technical University, Vinnitsa, e-mail: alextk1960@gmail.com