Разработка мобильных приложений

Работа с данными

План

1. Хранение данных
2. Взаимодействие с системой
3. Предоставление доступа к данным приложения

Хранение данных

Хранение данных

1. Ресурсы
2. Хранение настроек
3. Работа с файлами
4. Работа с базами данных

Ресурсы

https://developer.android.com/guide/topics/resources/providing-resources.html

Ресурсы

1. xml
2. raw
3. assets

Demo: RawResources

Demo: Assets

Хранение настроек (SharedPreferences)

https://developer.android.com/guide/topics/data/data-storage.html#pref

https://developer.android.com/training/data-storage/shared-preferences.html

https://developer.android.com/reference/android/content/SharedPreferences.html

SharedPreferences

- getSharedPreferences(String name)
- getPreferences() – принадлежит Activity
- PreferenceManager.getDefaultSharedPreferences() – принадлежит всему приложению

Поддерживаемые типы SharedPreferences

1. Integer
2. Long
3. Boolean
4. String
5. Float
6. Double

Основные операции SharedPreferences

1. SharedPreferences#get...(String);
2. SharedPreferences.Editor#put...(String, Object);
3. SharedPreferences.Editor#commit(); - синхронная запись на диск
4. SharedPreferences.Editor#apply(); - асинхронная запись на диск

Режимы доступа SharedPreferences

1. MODE_PRIVATE
2. ~~MODE_WORLD_READABLE~~
3. ~~MODE_WORLD_WRITEABLE~~
4. ~~MODE_MULTI_PROCESS~~

Demo: Preferences

Settings

1. Создание экранов настроек любой сложности
2. Полная настройка из xml
3. Автоматическая синхронизация с SharedPreferences

Settings

https://developer.android.com/guide/topics/ui/settings/

Demo: Settings

https://github.com/googlesamples/android-preferences

Работа с файлами

https://developer.android.com/training/basics/data-storage/files.html

https://developer.android.com/training/data-storage/app-specific

Режимы доступа к файлам

1. MODE_PRIVATE
2. MODE_APPEND
3. ~~MODE_WORLD_READABLE~~
4. ~~MODE_WORLD_WRITEABLE~~
5. ~~MODE_MULTI_PROCESS~~

Виды файловых хранилищ

1. Внутреннее
2. Внешнее

Внутреннее файловое хранилище

1. Приватный кэш
2. Приватное файловое хранилище

Demo: InternalStorage

Внешнее файловое хранилище

1. Кэш приложения
2. Файлы приложения
3. Общедоступные файлы (Android 10: только через специальное API или поставить targetSdk < 29)

Проверка доступа на чтение к внешнему хранилищу

```
public boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state) ||
        Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        return true;
    }
    return false;
}
```

Проверка доступа на запись к внешнему хранилищу

```
public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}
```

Пермишены доступа к внешнему хранилищу

- android.permission.READ_EXTERNAL_STORAGE
- android.permission.WRITE_EXTERNAL_STORAGE

Demo: ExternalStorage

Способы преобразования объектов в байтовый массив

1. Parcelable
2. Serializable

Parcelable

https://developer.android.com/guide/components/activities/parcelables-and-bundles.html

https://developer.android.com/reference/android/os/Parcelable.html

@Parcelize https://kotlinlang.org/docs/tutorials/android-plugin.html#parcelable-implementations-generator

Serializable

https://developer.android.com/reference/java/io/Serializable.html

Parcelable vs Serializable

- Parcelable – более быстрый, можно использовать только с Intent или Bundle
- Serializable – можно писать в файл

Demo: SerializableParcelable

Базы данных

1. SQLite
2. RoomDatabase

SQLite

https://developer.android.com/training/data-storage/sqlite

http://www.sqlite.org

Поддерживаемые типы данных

1. TEXT
2. INTEGER
3. NUMERIC
4. REAL
5. BLOB

Create subclass of SQLiteOpenHelper

1. SQLiteOpenHelper(Context context, String name, Cur sorFactory factory, int version);
2. onCreate(SQLiteDatabase db);
3. onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion);
4. onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion).

Использование SQLiteOpenHelper

1. SQLiteDatabase db = getReadableDatabase();
2. SQLiteDatabase db = getWritableDatabase();

SQLiteDatabase

1. insert(...);
2. update(...);
3. delete(...);
4. query(...);
5. execSQL(...);

Cursor

https://developer.android.com/reference/android/database/Cursor.html

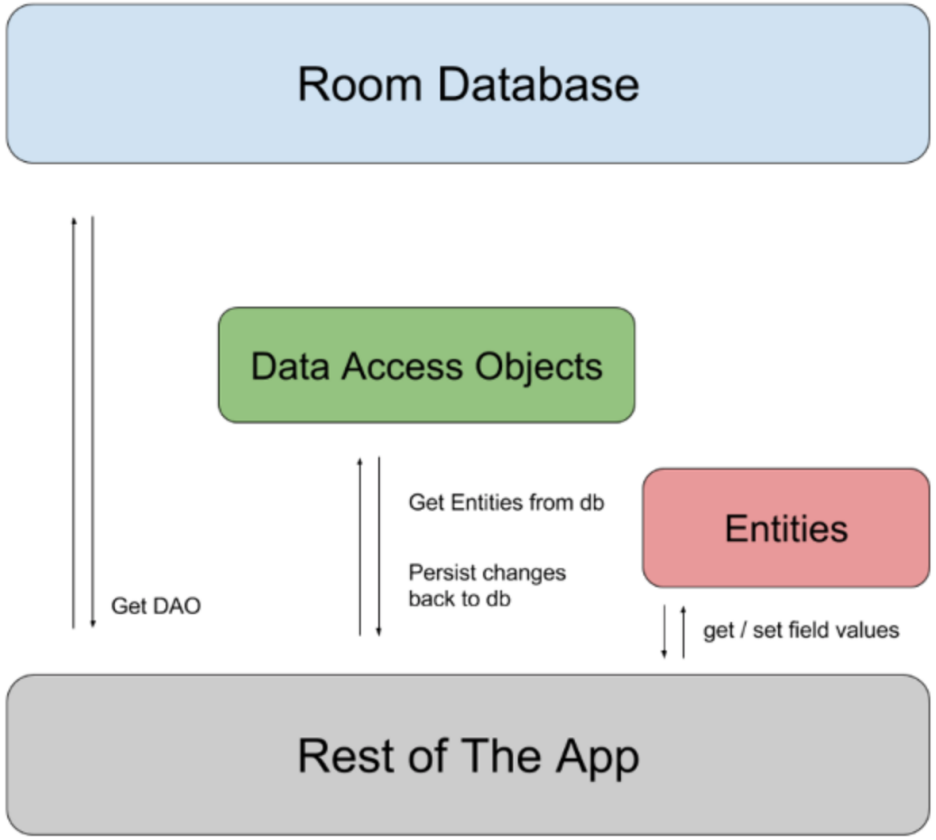
Demo: DatabaseStorage

Room Persistence Library

1) Подключить зависимость "androidx.room:room-runtime:2.2.4»
2) Database, Entity, DAO

https://developer.android.com/training/data-storage/room/index.html

Room Persistence Library



Demo: RoomDatabaseStorage

Взаимодействие с системой

Взаимодействие с системой

1. Системные сервисы
2. Системные контент провайдеры
3. Системные уведомления

Системные сервисы

- Context#getSystemService(Class<T> serviceClass):

https://developer.android.com/reference/android/content/Context.html#getSystemService(java.lang.Class<T>)

- Context#getPackageManager()

Demo: PackageManager

Полезные ссылки

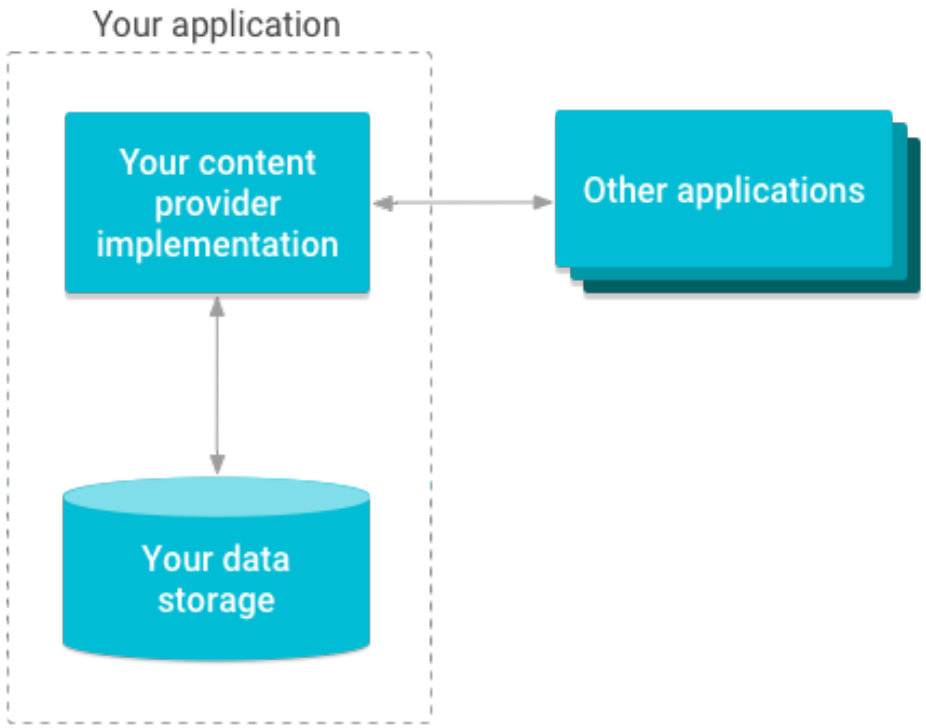
https://developer.android.com/guide/components/intents-filters.html

https://developer.android.com/training/basics/intents/filters.html

ContentProvider

- Предоставляет доступ к структурированному набору данных
- Инкапсулирует источник данных (при изменении хранилища нужно адаптировать ContentProvider, но другие приложения ничего не заметят)
- Предоставляет механизмы для обеспечения безопасности

ContentProvider



Системные Content Provider’ы

- Контакты
- Журнал вызовов
- Календарь
- Настройки

ContentProvider

https://developer.android.com/guide/topics/providers/content-providers.html

https://developer.android.com/guide/topics/providers/content-provider-basics.html

ContentProvider: что нужно

- Разрешение на чтение
- URI (content://authority/table_name)
- Имена столбцов
- Типы данных в столбцах

ContentResolver API

- query(Uri, projection, where, selectionArgs, sortOrder)
- insert(Uri, ContentValues)
- update(Uri, ContentValues, where, selectionArgs)
- Delete(Uri, where, selectionArgs)
- registerContentObserver(Uri, notifyForDescendants, ContentObserver)

ContentProvider: типы данных

- TEXT
- INTEGER
- LONG
- FLOAT
- DOUBLE
- BLOB
- MIME (text/html, собственные строки)

System broadcasts

https://developer.android.com/guide/components/broadcasts.html

https://developer.android.com/reference/android/content/Intent.html

Broadcasts

- Ресиверы получают Intent, где action описывает, что именно произошло
- Intent-filters: на какие action’ы подписывается ресивер
- onReceive на главном потоке
- Завершается onReceive – компонент может быть уничтожен
- Explicit broadcasts
- Пермишены

System broadcasts

- Broadcast-ресиверы, объявленные в манифесте: перехватят сообщение, даже если приложение не запущено, только для explicit
- Broadcast-ресиверы, зарегистрированные у контекста: работают для implicit broadcast’ов

Demo: Broadcasts

Предоставление доступа к данным приложения

MediaStore & MediaScannerConnection

https://developer.android.com/reference/android/provider/MediaStore.html

https://developer.android.com/reference/android/media/MediaScannerConnection.html

Content Provider

https://developer.android.com/guide/topics/providers/content-provider-creating.html

Demo: ContentProvider + ContentProviderUser

DocumentProvider

- Позволяет пользователю самому выбирать файлы
- Участники: клиентское приложение, DocumentProvider, Picker
- Процесс: клиентское приложение отправляется Intent (ACTION_OPEN_DOCUMENT, ACTION_CREATE_DOCUMENT), возможно с фильтрами -> система предлагает пользователю выбрать приложение-провайдер –> дальнейшее взаимодействие только с провайдером

Document Provider

https://developer.android.com/guide/topics/providers/document-provider.html

https://developer.android.com/guide/topics/providers/create-document-provider.html

File Provider

https://developer.android.com/training/secure-file-sharing/setup-sharing.html

Demo: FileProvider

Что осталось за кадром

- бэкап данных приложения:

https://developer.android.com/guide/topics/data/backup

- сохранение состояния UI:

https://developer.android.com/topic/libraries/architecture/saving-states

- сохранение игрового прогресса в облако:

https://developers.google.com/games/services/common/concepts/savedgames

Что осталось за кадром

- использование облачных БД:

https://firebase.google.com/docs/database/