

ANDROID ALKALMAZÁSFEJLESZTÉS

Adattárolás

SharedPreferences
Belső - külső tároló
PreferenceActivity
Felhő alapú tárolás
SQLite



Sicz-Mesziár János

sicz-mesziar.janos@nik.uni-obuda.hu

Mezei József

mezei.jozsef@nik.uni-obuda.hu

2018. március 17.

Shared Preference

Egy egyszerű megoldás primitív értékek perzisztens tárolására.

Az alábbiakat lehet: boolean, float, int, long, string.

Felhasználói beállításoknál is erre építenek, de arra van egységes megoldás!

Tárolás:

```
String PrefFileName = "MyPrefName";
SharedPreferences settings = getSharedPreferences(PrefFileName, 0);
SharedPreferences.Editor editor = settings.edit();
editor.putInt("test", 1027);
editor.commit();
```

Visszaállítás:

```
SharedPreferences settings = getSharedPreferences(PrefFileName, 0);
int szam = settings.getInt("test", 0);
```

Belső tároló

Fájlmentés közvetlenül a belső tárolóra.

Ha az alkalmazást törlik, akkor ezek a fájlok is törlődnek!

Alapértelmezetten más alkalmazás nem fér hozzá.

Tárolás:

```
String FILENAME = "StorageFile";
String saveThis = "Helló Storage!";
FileOutputStream fos = openFileOutput(FILENAME, MODE_PRIVATE);
fos.write(saveThis.getBytes());
fos.close();
```

```
/data/data/hu.uniobuda.nik.StorageTest/files/StorageFile
```

Betöltés:

```
FileInputStream fis = openFileInput(FILENAME);
byte[] buffer = new byte[1024]; int len;
while((len = fis.read(buffer)) > 0)
    Log.d("TAG", new String(buffer, 0, len));
fis.close();
```

Továbbá: `fileList()`, `deleteFile()`, `getDir()`, ...

Külső tároló - SD kártya

- Bármely másik alkalmazás (v. felhasználó) által olvasható és írható, osztott háttértár.
- Bármikor eltávolíthatja a felhasználó.



Már kevésbé jellemző az MTP képes készülékek miatt!

Ellenőrizni kell az elérhetőségét és állapotát:

```
String state = Environment.getExternalStorageState();
if(state.equals(Environment.MEDIA_MOUNTED)){
    // Elérhető, írható és olvasható
}else if(state.equals(Environment.MEDIA_MOUNTED_READ_ONLY)){
    // Elérhető, de csak olvasható
}else{
    // Más állapotban van... DE se nem írható, se nem olvasható
}

String sdcard =
    Environment.getExternalStorageDirectory().getAbsolutePath();
```

Jogosultság: `android.permission.WRITE_EXTERNAL_STORAGE`

Scoped Directory Access

A *READ_EXTERNAL_STORAGE* vagy a *WRITE_EXTERNAL_STORAGE* hozzáférést ad az összes publikus könyvtárhoz a külső tárolón.

A [Storage Access Framework](#) könyvtárkiválasztásra tervezték ami szükségtelen ha mindig ugyanazzal a könyvtárral dolgozol.



Fenti probléma feloldása a **Scoped Directory Access**, ahol a third-party alkalmazás csak a megadott könyvtárhoz vagy fájlhoz fog tudni hozzáférni, nem pedig a teljes külső tárolóhoz.

Android 7.0 óta

Bővebben:

<https://developer.android.com/training/articles/scoped-directory-access.html>

Preference Activity



Kimondottan felhasználói beállítások tárolása.

Néhány beépített form:

CheckBoxPreference, EditTextPreference,
ListPreference, RingtonePreference + Egyéni!

Témák szerint kategorizálhatunk.

Preference XML létrehozása:

1. File > New > Other... > Android XML File
2. Layout helyett Preference típus!
3. Add > ChechBoxPreference, ...

Java osztály:

1. extends PreferenceActivity
2. onCreate() implementálása
3. addPreferencesFromResource()

+ **AndroidManifest.xml**

Felhő alapú tárolás – Data backup

Adatmentés felhőbe v. visszaállítási pont beállításokról.

Akkor jó, ha:

- Factory reset történt, pl.: felhasználó által
- Új készüléket vett/kapott a felhasználó

**Ez nem
szinkronizáció!**

Nem garantált, hogy minden eszközön elérhető!

- Az alkalmazás működést ez nem befolyásolja, pusztán nem lesz restore() hívás

Használata

- BackupAgent
- BackupAgentHelper
- [Részletek itt](#)

Google Dashboard-ban ☺ →

<https://www.google.com/dashboard/>



Application	Backup date	Backup size
Android Wallpaper	Feb 7, 2012 11:25 PM	437.36 KB
Android Browser	Sep 20, 2011 8:14 PM	813 B
Android System Settings	Mar 2, 2012 5:21 PM	2.47 KB
Android Market	Dec 21, 2011 11:46 AM	8 B

Felhő alapú tárolás – **Storage services**

Fájlok kezelése a felhőben:



Google Drive Android API

<https://developers.google.com/drive/android/>



DropBox Core Android API

<https://www.dropbox.com/developers/core/start/android>





SQLite



SQLite-ról

- Egy önálló, szerver nélküli, konfigurálást nem igénylő, tranzakciós adatbázis motor. – www.sqlite.org
- Android alatt teljes SQLite támogatás.
- Alkalmazáson belül bárhonnán elérhető, de azon kívül nem!
- ContentProvider-ek segítségével lehetséges az alkalmazások közötti adatcsere is.
 - Pl.: Böngésző könyvjelzők, előzmények, kontaktok, ...
- Megszokott SQL kódok használata:
 - INSERT,
 - SELECT,
 - UPDATE,
 - DELETE,
 - CREATE, ...

SQLiteOpenHelper

- Egy tervezési minta, feladata az adatbázis kezelésének megkönnyítése.
- Előnye, hogy jól követhető vele az adatstruktúra változása az egyes alkalmazásverziók alatt.

```
public class DatabaseHandler extends SQLiteOpenHelper {  
  
    private static final int DATABASE_VERSION = 1;  
    private static final String DATABASE_NAME = "dbname";  
  
    public DatabaseHandler(Context context) {  
        super(context, DATABASE_NAME, null, DATABASE_VERSION);  
    }  
  
    @Override  
    public void onCreate(SQLiteDatabase db) {  
        String CREATE_CONTACTS_TABLE = "CREATE TABLE ...";  
        db.execSQL(CREATE_CONTACTS_TABLE);  
    }  
  
    @Override  
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
        db.execSQL("DROP TABLE IF EXISTS ...");  
        onCreate(db);  
    }  
}
```

SQLite műveletek

Fontosabb SQL műveletek:

- `SQLiteDatabase.create(...)`
- `SQLiteDatabase.delete(...)`
- `SQLiteDatabase.query(...)`
- `SQLiteDatabase.rawQuery(...)`
- `SQLiteDatabase.execSQL(...)`
- `SQLiteDatabase.insert(...)`
- `SQLiteDatabase.replace(...)`

Lekérdezések esetén a visszatérési érték egy `Cursor`, mellyel végig mehetünk az eredmény listán:

- `get[Int | Short | Long | Float | Double | String](...)`
- `moveTo[First | Last | Next | Position](...)`
- `getColumn[Count | Index | Name](...)`

SQLite eszközök

Néhány ajánlott SQLite adatbázist kezelő GUI alkalmazás:

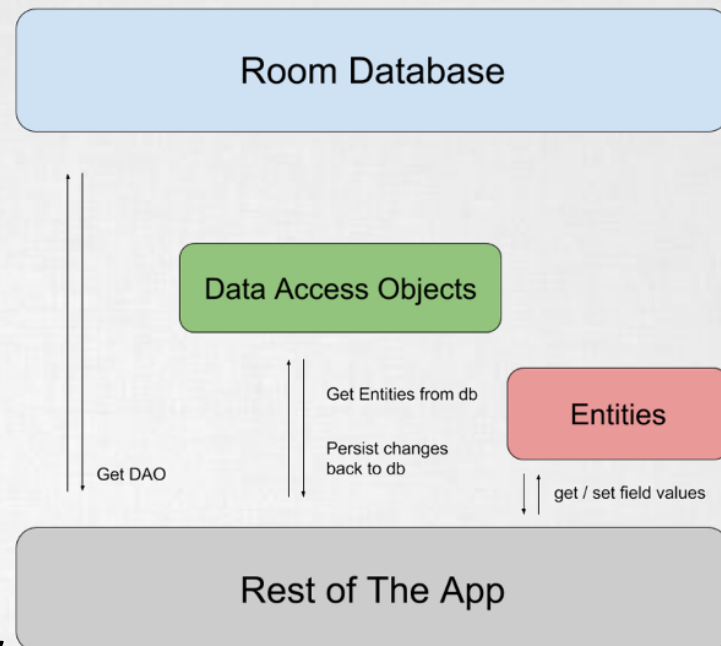
- SQLite Studio
<http://sqlitestudio.pl/>
- SQLite Database browser
<http://sqlitebrowser.sourceforge.net/>
- SQLite Administrator
<http://sqliteadmin.orbmu2k.de/>
- SQLite Manager Firefox add-on
<https://addons.mozilla.org/en-US/firefox/addon/sqlite-manager/>
- SQLite Spy
<http://www.yunqa.de/delphi/doku.php/products/sqlitespy/index>

Room Persistence Library

A Room egy absztrakciós réteg az SQLite felett. Célja, hogy az adatbázissal járó *boilerplate* kódokat elhagyjuk.

3 alap komponense van:

1. Database:
maga az adatbázist képviseli
2. Entity:
egy adatbázis sort reprezentáló osztály
3. DAO:
az adatbázis műveletek interfésze



Bővebben:

<https://developer.android.com/topic/libraries/architecture/room.html>