Formation Android

Horacio Gonzales

Spaniard lost in Brittany, Java developer, dreamer and all-around geek

- Senior Software Engineer at Cityzen Data

Cityzen Data provides a scalable, secure, ethical
and open platform for sensor data

- Leader du FinistJUG, du GDG Finistère et de la communauté BreizhBeans

http://lostinbrittany.org/
+Horacio.Gonzalez
@LostInBrittany

Stéphane Castrec

Ingénieur logiciel
- Développeur au CM Arkéa
- CTO à EQwall

@_stephane_

StephaneC

Android Storage

3 options

- Preferences
- Base de données
- Fichiers

Les préférences

Les préférences

- Clef / Valeur
  - Stockage en fichier
- Lié à un contexte
- Peut être “Private” ou “Shared”

Les préférences

Très utilisé pour:

- Settings
- Login (email)
- Données “simples”

Les préférences

Utilisation en lecture

```
public static String getValue(final Context context, String key){
    SharedPreferences sharedPref = context.getSharedPreferences(PREFS, Context.MODE_PRIVATE);
    return sharedPref.getString(key, "");
}
```

Les préférences

Utilisation en écriture

```
public static void setValue(final Context context, String key, String value){
    SharedPreferences sharedPref = context.getSharedPreferences(PREFS, Context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sharedPref.edit();
    editor.putString(key, value);
    editor.commit();
}
```

Les préférences

Utilisation en écriture

```
SharedPreferences sharedPref = getActivity().getPreferences(Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedPref.edit();
editor.putInt(getString(R.string.saved_high_score), newHighScore);
editor.commit();
```


KEEP
CALM
AND
GIVE IT
A TRY

La base de données

La base de données

Base de donnée SQLite

- Types supportés:
  - TEXT (eq String en Java)
  - INTEGER (eq long en Java)
  - REAL (eq double en Java)

Sauvegarde

- DATA/data/APP_NAME/databases/FILENAME

La base de données

Comment l'utiliser?

1. Définir le schéma de données
2. Créer la base
3. Ecrire des informations
4. Lire des informations

La base de données

1. Définir le schéma de données

```
public final class FeedReaderContract {
    // To prevent someone from accidentally instantiating the contract class,
    // give it an empty constructor.
    public FeedReaderContract() {}

    /* Inner class that defines the table contents */
    public static abstract class FeedEntry implements BaseColumns {
        public static final String TABLE_NAME = "entry";
        public static final String COLUMN_NAME_ENTRY_ID = "entryid";
        public static final String COLUMN_NAME_TITLE = "title";
        public static final String COLUMN_NAME_SUBTITLE = "subtitle";
        ...
    }
}
```

La base de données

2. Créer la base

```
private static final String TEXT_TYPE = " TEXT";
private static final String COMMA_SEP = ",";
private static final String SQL_CREATE_ENTRIES =
    "CREATE TABLE " + FeedEntry.TABLE_NAME + " (" +
    FeedEntry._ID + " INTEGER PRIMARY KEY," +
    FeedEntry.COLUMN_NAME_ENTRY_ID + TEXT_TYPE + COMMA_SEP +
    FeedEntry.COLUMN_NAME_TITLE + TEXT_TYPE + COMMA_SEP +
    ... // Any other options for the CREATE command
    " )";

private static final String SQL_DELETE_ENTRIES =
    "DROP TABLE IF EXISTS " + FeedEntry.TABLE_NAME;
```

La base de données

2. Créer la base

```
public class FeedReaderDbHelper extends SQLiteOpenHelper {
    // If you change the database schema, you must increment the database version.
    public static final int DATABASE_VERSION = 1;
    public static final String DATABASE_NAME = "FeedReader.db";

    public FeedReaderDbHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(SQL_CREATE_ENTRIES);
    }
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // This database is only a cache for online data, so its upgrade policy is
        // to simply to discard the data and start over
        db.execSQL(SQL_DELETE_ENTRIES);
        onCreate(db);
    }
    public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        onUpgrade(db, oldVersion, newVersion);
    }
}
```

La base de données

3. Ecrire les informations

```
// Gets the data repository in write mode
SQLiteDatabase db = mDbHelper.getWritableDatabase();

// Create a new map of values, where column names are the keys
ContentValues values = new ContentValues();
values.put(FeedEntry.COLUMN_NAME_ENTRY_ID, id);
values.put(FeedEntry.COLUMN_NAME_TITLE, title);
values.put(FeedEntry.COLUMN_NAME_CONTENT, content);

// Insert the new row, returning the primary key value of the new row
long newRowId;
newRowId = db.insert(
         FeedEntry.TABLE_NAME,
         FeedEntry.COLUMN_NAME_NULLABLE,
         values);
```

La base de données

4. Lire les informations

```
SQLiteDatabase db = mDbHelper.getReadableDatabase();

// Define a projection that specifies which columns from the database
// you will actually use after this query.
String[] projection = {
    FeedEntry._ID,
    FeedEntry.COLUMN_NAME_TITLE,
    FeedEntry.COLUMN_NAME_UPDATED,
    ...
    };

// How you want the results sorted in the resulting Cursor
String sortOrder =
    FeedEntry.COLUMN_NAME_UPDATED + " DESC";

Cursor c = db.query(
    FeedEntry.TABLE_NAME,  // The table to query
    projection,                               // The columns to return
    selection,                                // The columns for the WHERE clause
    selectionArgs,                            // The values for the WHERE clause
    null,                                     // don't group the rows
    null,                                     // don't filter by row groups
    sortOrder                                 // The sort order
    );
```


KEEP
CALM
AND
GIVE IT
A TRY