Développement Application Android

Base de Données

Anthony Chomienne

CPE Lyon

 2023

Développement Application Android — Plan

Développement Application Android — Plan

SQLiteOpenHelper

- Deux méthodes à surcharger
- `public void onCreate(SQLiteDatabase db);`
- `public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion);`

SQLiteDatabase: Opérations

- Quelle Méthode pour quelle opération?
- `public void execSQL(String sql);`
- `public long insert(String table, String nullColumnHack, ContentValues values);`
- `public int update(String table, ContentValues values, String whereClause, String[] whereArgs);`
- `public int delete(String table, String whereClause, String[] whereArgs);`
- `public Cursor query(String table,String[] columns, String whereClause, String[] whereArgs, String groupBy, String having, String orderBy);`

whereArgs et whereClause

- `String whereClause = "id= ? and age > ?"`
- `String[] whereArgs = {String.valueOf(id),String.valueOf(age)}`

SQLiteDatabase: Transactions

- La gestion se fera dans un try/catch
- `public void beginTransaction();`
- `public void endTransaction();`
- `public void setTransactionSuccessful();`

SQLiteDatabase: Transactions

```
db.beginTransaction();
try{
//do Some Work
   db.setTransactionSuccessful();
}
catch(){
   //do Some Treatment
}
finaly{
   db.endTransaction();
}
```

Utilisation d'un Singleton 1/3

- Permet de n'avoir qu'une instance du SQLiteOpenHelper à la fois.
- Permet de contrôler la version qu'on utilise de manière centralisée
- Un seul point d'entré vers la base de données

UTILISATION D'UN SINGLETON 2/3

```
public class Database {
    private static Database instance = null;
    private SQLOpenHelper helper;
    public static Database getInstance(Context
        context){
         if(instance == null)
              instance = new Database(context);
         return instance;
    }

    private Database(Context context){
         helper = new
             MySqlOpenHelper(context,"MyDatabase.db",
             null, 2);
    }
}
```

Utilisation d'un Singleton 3/3

```
Database.getInstance(context).myQuery();
```

Développement Application Android — Plan

GRADLE

```
implementation
    'androidx.room:room-runtime:2.5.0'
annotationProcessor
    "androidx.room:room-compiler:2.5.0"
```

Entité

```
@Entity
public class User {
    @PrimaryKey
    public int uid;

    @ColumnInfo(name = "first_name")
    public String firstName;

    @ColumnInfo(name = "last_name")
    public String lastName;
}
```

DAO

```
@Dao
public interface UserDao {
    @Query("SELECT * FROM user")
    List<User> getAll();

    @Query("SELECT * FROM user WHERE first_name
       LIKE :first AND " +
           "last_name LIKE :last LIMIT 1")
    User findByName(String first, String last);

    @Insert
    void insertAll(User... users);

    @Delete
    void delete(User user);
}
```

Database

```
@Database(entities = {User.class}, version = 1)
public abstract class AppDatabase extends
    RoomDatabase {
     public abstract UserDao userDao();
}
```

Usage

```
AppDatabase db =
    Room.databaseBuilder(getApplicationContext(),
     AppDatabase.class,
        "userDatabase.db").build();

UserDao userDao = db.userDao();
List<User> users = userDao.getAll();
```

Gradle

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

Développement Application Android — Plan

Sources et Biblio

- http://www.idc.com/prodserv/smartphone-os-market-share.jsp
- Part de marché Mobile en France Janvier 2015
- http://opensignal.com/reports/2015/08/android-fragmentation/
- http://developer.android.com/
- http://www.statista.com/
- https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad