

Hướng dẫn lập trình cơ bản với Android - Bài 6

Reflink: <http://diendan.vietandroid.com/cac-bai-hoc-co-kem-ma-nguon/991-huong-dan-lap-trinh-co-ban-voi-android-bai-6-a.html>

List tutorial

Bài 0 - Cài đặt và sử dụng Android với Eclipse

Bài 1 - Cơ bản Android

Bài 2 - Xây dựng giao diện đơn giản

Bài 3 - ViewGroup và Custom Adapter

Bài 4 - Intent và Broadcast Receiver

Bài 5 - Service

Bài 6 - SQLite

Bài 7 - Content Provider

Bài 8 : Thread & Handler (Coming soon)

Bài 6 : Android SQLite Database

Trong bài hôm nay mình sẽ giới thiệu với các bạn cơ sở dữ liệu trên Android - SQLiteDatabase.

0. Giới thiệu SQLite Database

Mỗi ứng dụng đều sử dụng dữ liệu, dữ liệu có thể đơn giản hay đôi khi là cả 1 cấu trúc. Trong Android thì hệ cơ sở dữ liệu được sử dụng là SQLite Database, đây là hệ thống mã nguồn mở được sử dụng rộng rãi trong các ứng dụng (Mozilla Firefox sử dụng SQLite để lưu trữ các dữ liệu về cấu hình, iPhone cũng sử dụng cơ sở dữ liệu là SQLite).

Trong Android , cơ sở dữ liệu mà bạn tạo cho 1 ứng dụng thì chỉ ứng dụng đó có quyền truy cập và sử dụng, các ứng dụng khác thì không. Khi đã được tạo, cơ sở dữ liệu SQLite được chứa trong thư mục `/data/data/<package_name>/databases` .

Tiếp theo mình sẽ giới thiệu với các bạn cách tạo và sử dụng cơ sở dữ liệu SQLite . Cách tốt nhất để hiểu lý thuyết là thực hành, các bạn có thể tạo ngay 1 project trong workspace đặt tên là : SQLiteDemo. Ứng dụng này đơn giản chỉ tạo 1 cơ sở dữ liệu USER bao gồm 2 trường (id , name) , ngoài ra có các thao tác thêm , đọc , sửa , xóa .

Các bước thực hiện với CSDL:

1. Tạo 1 CSDL (thông thường chỉ cần làm 1 lần)
2. Mở CSDL đó
3. Thêm giá trị vào trong table
4. Truy vấn.
5. Đóng CSDL

1. Tạo cơ sở dữ liệu.

Đầu tiên các bạn add 1 class DBAdapter để xử lý tất cả các thao tác liên quan đến CSDL.

Mã:

```
public static final String KEY_ID = "_id";
public static final String KEY_NAME = "name";

private DatabaseHelper mDbHelper;
private SQLiteDatabase mDB;
```

```

        private static final String DATABASE_CREATE = "create table
users (_id integer primary key autoincrement, "
        + "name text not null);";
        private static final String DATABASE_NAME = "Database_Demo";
        private static final String DATABASE_TABLE = "users";
        private static final int DATABASE_VERSION = 2;

        private final Context mContext;

```

Tạo 1 lớp bên trong DBAdapter được extend từ lớp SQLiteOpenHelper, override 2 phương thức onCreate() và onUpgrade() để quản lý việc tạo CSDL và version của CSDL đó.

Mã:

```

        private static class DatabaseHelper extends SQLiteOpenHelper{

            public DatabaseHelper(Context context, String name,
                CursorFactory factory, int version) {
                super(context, name, factory, version);
                // TODO Auto-generated constructor stub
            }

            @Override
            public void onCreate(SQLiteDatabase db) {
                // TODO Auto-generated method stub
                db.execSQL(DATABASE_CREATE);
            }

            @Override
            public void onUpgrade(SQLiteDatabase db, int oldVersion,
int newVersion) {
                // TODO Auto-generated method stub
                Log.i(TAG, "Upgrading DB");
                db.execSQL("DROP TABLE IF EXISTS users");
                onCreate(db);
            }
        }

```

2. Mở CSDL :

Mã:

```

        public DBAdapter open()
        {
            mDbHelper = new DatabaseHelper(mContext, DATABASE_NAME,
null, DATABASE_VERSION);
            mDB = mDbHelper.getWritableDatabase();
            return this;
        }

```

3. Thêm giá trị vào CSDL

Mã:

```

        public long createUser(String name){
            ContentValues inititalValues = new ContentValues();
            inititalValues.put(KEY_NAME, name);
            return mDB.insert(DATABASE_TABLE, null, inititalValues);
        }

```

4. Truy vấn

Bạn có thể get toàn bộ data hoặc có thể get data theo ID (tiện cho việc chỉnh sửa hay cập nhật thông tin của từng bản ghi).

Mã:

```
public Cursor getAllUsers(){
    return mDB.query(DATABASE_TABLE, new String[] {KEY_ID,
KEY_NAME}, null, null, null, null, null);
}
```

Còn rất nhiều các thao tác như sửa, xóa, update.... bản ghi, các bạn có thể tự phát triển. Tất cả các chức năng đó đều được cung cấp bởi lớp SQLiteDatabase, các bạn chỉ cần cụ thể hóa bằng các câu truy vấn là được.

5. Đóng CSDL

Mã:

```
public void close(){
    mDbHelper.close();
}
```

6. Sử dụng CSDL

Để test CSDL mà bạn vừa tạo, các bạn có thể thêm 1 vài dòng code để thêm 1 user và hiển thị CSDL lên màn hình thông qua lớp Activity ban đầu:

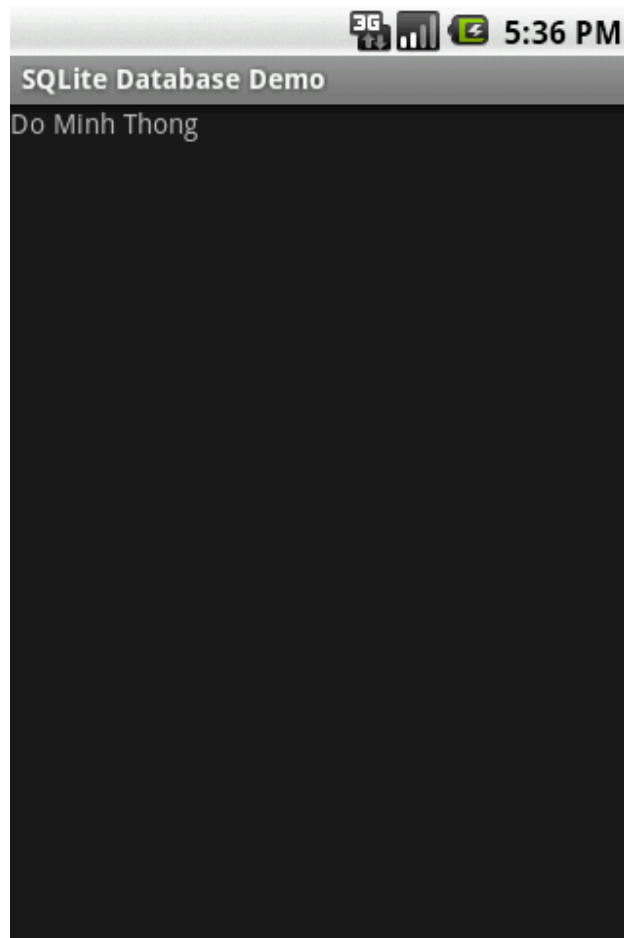
Ở đây mình create 1 user thông qua câu lệnh mDB.createUser("Username"); sau đó

Mã:

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    mDB = new DBAdapter(this);
    mDB.open();
    mDB.createUser("Do Minh Thong");
    getData();
}

private void getData(){
    mCursor = mDB.getAllUsers();
    startManagingCursor(mCursor);
    String[] from = new String[]{DBAdapter.KEY_NAME};
    int[] to = new int[] {R.id.text1};
    SimpleCursorAdapter users = new SimpleCursorAdapter(this,
R.layout.users_row, mCursor, from, to);
    setListAdapter(users);
}
```

Kết quả :



Sourcecode đầy đủ : <http://www.mediafire.com/?yzw2d1ijymo>

Bài này là bài cơ sở để mình viết tiếp Phần 7 : ContentProvider , các bạn chú ý theo dõi nhé 😊

GL&HL All.