

# Hướng dẫn lập trình cơ bản với Android

## Part 22

### 2. Tạo và sử dụng 1 Content Provider do người dùng tự định nghĩa

Để dễ hiểu hơn các bạn mở Project ContentProviderDemo trong Sourcecode đã down về. Trong Project đó mình tạo 1 Content Provider Books, mỗi bản ghi Book bao gồm 2 trường : ID và Title.

Sau đây là các bước để tạo 1 Content Provider cơ bản ( cụ thể là tạo ContentProvider Book)

#### 1. Tạo 1 class thừa kế lớp ContentProvider

Mã:

```
public class BookProvider extends ContentProvider
```

2. Định nghĩa 1 biến Uri (public static final ) được gọi CONTENT\_URI. Các xâu này luôn được bắt đầu bằng “content://” tiếp theo đó là nội dung của mà ContentProvider xử lý. Xâu này phải có đặc tính là duy nhất.

Mã:

```
public static final String PROVIDER_NAME =  
"com.vietandroid.provider.Books";  
public static final Uri CONTENT_URI =  
Uri.parse("content://" + PROVIDER_NAME + "/books");
```

3. Khai báo các xâu để định nghĩa cho từng thuộc tính tương ứng với các cột giá trị từ Cursor.

Mã:

```
public static final String _ID = "_id";  
public static final String TITLE = "title";
```

4. Chúng ta cần tạo hệ thống chứa dữ liệu cho ContentProvider, có thể chứa dưới nhiều hình thức : sử dụng XML, thông qua CSDL SQLite, hay thậm chí là Webservice. Trong Demo này chúng ta sử dụng cách phổ biến nhất đó là SQLite:

Mã:

```
private SQLiteDatabase bookDB;
private static final String DATABASE_NAME =
"Books";
private static final String DATABASE_TABLE =
"titles";
private static final int DATABASE_VERSION = 1;
```

**5. Định nghĩa tên của các cột mà chúng ta sẽ trả lại giá trị cho các clients.** Nếu chúng ta đang sử dụng Database ContentProvider hay các lớp SQLiteOpenHelper, tên các cột này chính là id của các cột trong cơ sở dữ liệu SQL. Trong trường hợp này, chúng ta phải gộp cả cột có giá trị là số nguyên được gọi “\_id” để định nghĩa id của mỗi bản ghi.

Nếu đang sử dụng cơ sở dữ liệu SQLite, nó sẽ là INTEGER PRIMARY KEY AUTOINCREMENT. Tùy chọn AUTOINCREMENT không bắt buộc, có tác dụng tự động tăng ID của mỗi bản ghi lên nếu người dùng không nhập. Android cung cấp SQLiteOpenHelper giúp tạo và quản lý các phiên bản của cơ sở dữ liệu.

Mã:

```
private static final String DATABASE_CREATE =
    "create table " + DATABASE_TABLE +
    " (_id integer primary key autoincrement, "
    + "title text not null);";

private static class DatabaseHelper extends
    SQLiteOpenHelper
    {
        public DatabaseHelper(Context context) {
            super(context, DATABASE_NAME , null,
DATABASE_VERSION);
        }

        @Override
        public void onCreate(SQLiteDatabase db) {
            db.execSQL(DATABASE_CREATE);
        }
    }
```

```

        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int
oldVersion, int newVersion) {
            db.execSQL("DROP TABLE IF EXISTS
titles");
            onCreate(db);
        }
    }
}

```

**6.** Nếu chúng ta muốn public các dữ liệu kiểu byte như bitmap thì các trường mà chứa dữ liệu này nên là một chuỗi với 1 content://URI cho file đó. Đây chính là liên kết để các ứng dụng khác có thể truy cập và sử dụng dữ liệu bitmap này.

**7.** Sử dụng Cursor để thao tác trên tập dữ liệu : query (), update(), insert(), delete()..... Có thể gọi phương thức ContentResolver.notifyChange() để biết khi nào dữ liệu được cập nhật.

Add Book

Mã:

```

        @Override
        public Uri insert(Uri uri, ContentValues values) {
            long rowID = bookDB.insert(DATABASE_TABLE,
"", values);
            if(rowID > 0)
            {
                Uri mUri =
ContentUris.withAppendedId(CONTENT_URI, rowID);

                getContext().getContentResolver().notifyChange(mUri, null);

                return mUri;
            }
            throw new SQLException("Failed to insert new
row into " + uri);
        }
    }
}

```

## Get All Books

Mã:

```
@Override
public Cursor query(Uri uri, String[] projection,
String selection,
String[] selectionArgs, String
sortOrder) {
    SQLiteQueryBuilder sqlBuilder = new
    SQLiteQueryBuilder();
    sqlBuilder.setTables(DATABASE_TABLE);
    if(uriMatcher.match(uri) == BOOK_ID)
        sqlBuilder.appendWhere(_ID + "=" +
uri.getPathSegments().get(1));
    if(sortOrder == null || sortOrder == "")
        sortOrder = TITLE;
    Cursor c = sqlBuilder.query(bookDB,
projection, selection, selectionArgs, null, null,
sortOrder);

    c.setNotificationUri(getContext().getContentResolv
er(), uri);
    return c;
}
}
```

Mình chỉ demo 2 chức năng là thêm sách và lấy toàn bộ bản ghi trong CSDL , ngoài ra các phương thức edit, sửa , update, xóa... các bạn có thể tự làm .

## 8. Khai báo Content Provider trong file AndroidManifest.xml

Mã:

```
<provider android:name = "BookProvider"
    android:authorities="com.vietandroid.provider.Books" />
```

Như vậy chúng ta đã tạo xong ContentProvider Book tự định nghĩa.