**[Content Provider tutorial in Android with Example](http://www.compiletimeerror.com/2013/12/content-provider-in-android.html)**

**I**n Android security model, one application cannot directly access (read/write) other application's data. Every application has its own id data directory and own protected memory area.  
Content provider is the best way to share data across applications. Content provider is a set of data wrapped up in a custom API to read and write. Applications/Processes have to register themselves as a provider of data. Other applications can request Android to read/write that data through a fixed API.  
Content provider API adheres to CRUD principle.  
  
  
Examples for content provider are Contacts-that exposes user information to other applications, Media store-Allows other applications to access,store media files.

Above diagram shows how content provider works. App 1 stores its data in its own database and provides a provider. App 2 communicates with the provider to access App 1's data.  
  
Content providers are simple interfaces which uses standard insert(), query(), update(), delete() methods to access application data. So it is easy to implement a content provider.  
  
A special URI starting with ***content://***will be assigned to each content providers and that will be recognized across applications.  
  
**Writing a content provider:**  
The ContentProvider class is the central component of a content provider. To create a content provider we have to  
1. Create sub class for ContentProvider.  
2. Define content URI  
3. Implement all the unimplemented methods. insert(), update(), query(), delete(), getType().  
4. Declare the content provider in AndroidManifest.xml  
  
**Defining URI:**  
Content provider URI consists of four parts.  
content://authority/path/id  
content:// All the content provider URIs should start with this value  
'authority' is Java namespace of the content provider implementation. (fully qualified Java package name)  
'path' is the virtual directory within the provider that identifies the kind of data being requested.  
'id' is optional part that specifies the primary key of a record being requested. We can omit this part to request all records.  
  
**Adding new records:**  
We need to override insert() method of the ContentProvider to insert new record to the database via content provider. The caller method will have to specify the content provider URI and the values to be inserted without the ID. Successful insert operation will return the URI value with the newly inserted ID.  
For example: If we insert a new record to the content provider content://com.example/sample  
the insert method will return content://com.example/sample/1  
  
**Updating records:**  
To update one or more records via content provider we need to specify the content provider URI. update() method of the ContentProvider is used to update records. We have to specify the ID of the record to update a single record. To update multiple records, we have to specify 'selection' parameter to indicate which rows are to be changed.  
This method will return the number of rows updated.  
  
**Deleting records:**  
Deleting one or more records is similar to update process. We need to specify either ID or 'selection' to delete records. delete() method of the ContentProvider will return the number of records deleted.  
  
**Querying the content provider:**  
To query the data via content provider we orverride query() method of ContentProvider. This method has many parameters.  We can specify list columns to put into the result cursor using 'projection' parameter. We can specify 'selection' criteria. We can specify 'sortOrder' too.  
If we do not specify projection, all the columns will be included in the result cursor. If we do not specify sortOrder the provider will choose its own sort order.  
 **getType() method:**  
This method is used handle requests for the MIME type of the data at the given URI. We use either vnd.android.cursor.item or vnd.android.cursor.dir/  
vnd.android.cursor.item is used to represent specific item. Another one is used to specify all items.  
  
**Registering the provider in AndroidManifest.xml**  
As with any other major parts of Android application, we have to register the content providers too in the AndroidManifest.xml. <provider> element is used to register the content providers. <application> is its parent element.

<provider

android:name=".MyProvider"

android:authorities="com.example.contentproviderexample.MyProvider">

</provider>

Here authorities is the URI authority to access the content provider. Typically this will be the fully qualified name of the content provider class.

**Sample application:**

In this sample, we are going to create two application. First application will share its data through content provider and the second application will access the data.

**1. Application 'MyProvider'**

This application will create a content provider and share its data. This uses SQLite database 'mydb'. It has a table called 'names'. The table names have two columns id,name.  
The URI of this content provider is content://com.example.contentproviderexample.MyProvider/cte  
We have layout to insert new records to the database table.  
  
**Layout xml file:**

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<TextView

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Name" />

<EditText

android:id="@+id/txtName"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" />

<Button

android:id="@+id/btnAdd"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:onClick="onClickAddName"

android:text="Add Name" />

</LinearLayout>

**Content provider class:**  
MyProvider.java

**package** com.example.contentproviderexample;

**import** **java.util.HashMap**;

**import** **android.content.ContentProvider**;

**import** **android.content.ContentUris**;

**import** **android.content.ContentValues**;

**import** **android.content.Context**;

**import** **android.content.UriMatcher**;

**import** **android.database.Cursor**;

**import** **android.database.SQLException**;

**import** **android.database.sqlite.SQLiteDatabase**;

**import** **android.database.sqlite.SQLiteOpenHelper**;

**import** **android.database.sqlite.SQLiteQueryBuilder**;

**import** **android.net.Uri**;

**public** **class** **MyProvider** **extends** ContentProvider {

**static** **final** String PROVIDER\_NAME = "com.example.contentproviderexample.MyProvider";

**static** **final** String URL = "content://" + PROVIDER\_NAME + "/cte";

**static** **final** Uri CONTENT\_URI = Uri.parse(URL);

**static** **final** String id = "id";

**static** **final** String name = "name";

**static** **final** **int** uriCode = **1**;

**static** **final** UriMatcher uriMatcher;

**private** **static** HashMap<String, String> values;

**static** {

uriMatcher = **new** UriMatcher(UriMatcher.NO\_MATCH);

uriMatcher.addURI(PROVIDER\_NAME, "cte", uriCode);

uriMatcher.addURI(PROVIDER\_NAME, "cte/\*", uriCode);

}

**@Override**

**public** **int** **delete**(Uri uri, String selection, String[] selectionArgs) {

**int** count = **0**;

**switch** (uriMatcher.match(uri)) {

**case** **uriCode:**

count = db.delete(TABLE\_NAME, selection, selectionArgs);

**break**;

**default**:

**throw** **new** **IllegalArgumentException**("Unknown URI " + uri);

}

getContext().getContentResolver().notifyChange(uri, **null**);

**return** count;

}

**@Override**

**public** String **getType**(Uri uri) {

**switch** (uriMatcher.match(uri)) {

**case** **uriCode:**

**return** "vnd.android.cursor.dir/cte";

**default**:

**throw** **new** **IllegalArgumentException**("Unsupported URI: " + uri);

}

}

**@Override**

**public** Uri **insert**(Uri uri, ContentValues values) {

**long** rowID = db.insert(TABLE\_NAME, "", values);

**if** (rowID > **0**) {

Uri \_uri = ContentUris.withAppendedId(CONTENT\_URI, rowID);

getContext().getContentResolver().notifyChange(\_uri, **null**);

**return** \_uri;

}

**throw** **new** **SQLException**("Failed to add a record into " + uri);

}

**@Override**

**public** **boolean** **onCreate**() {

Context context = getContext();

DatabaseHelper dbHelper = **new** DatabaseHelper(context);

db = dbHelper.getWritableDatabase();

**if** (db != **null**) {

**return** **true**;

}

**return** **false**;

}

**@Override**

**public** Cursor **query**(Uri uri, String[] projection, String selection,

String[] selectionArgs, String sortOrder) {

SQLiteQueryBuilder qb = **new** SQLiteQueryBuilder();

qb.setTables(TABLE\_NAME);

**switch** (uriMatcher.match(uri)) {

**case** **uriCode:**

qb.setProjectionMap(values);

**break**;

**default**:

**throw** **new** **IllegalArgumentException**("Unknown URI " + uri);

}

**if** (sortOrder == **null** || sortOrder == "") {

sortOrder = name;

}

Cursor c = qb.query(db, projection, selection, selectionArgs, **null**,

**null**, sortOrder);

c.setNotificationUri(getContext().getContentResolver(), uri);

**return** c;

}

**@Override**

**public** **int** **update**(Uri uri, ContentValues values, String selection,

String[] selectionArgs) {

**int** count = **0**;

**switch** (uriMatcher.match(uri)) {

**case** **uriCode:**

count = db.update(TABLE\_NAME, values, selection, selectionArgs);

**break**;

**default**:

**throw** **new** **IllegalArgumentException**("Unknown URI " + uri);

}

getContext().getContentResolver().notifyChange(uri, **null**);

**return** count;

}

**private** SQLiteDatabase db;

**static** **final** String DATABASE\_NAME = "mydb";

**static** **final** String TABLE\_NAME = "names";

**static** **final** **int** DATABASE\_VERSION = **1**;

**static** **final** String CREATE\_DB\_TABLE = " CREATE TABLE " + TABLE\_NAME

+ " (id INTEGER PRIMARY KEY AUTOINCREMENT, "

+ " name TEXT NOT NULL);";

**private** **static** **class** **DatabaseHelper** **extends** SQLiteOpenHelper {

DatabaseHelper(Context context) {

**super**(context, DATABASE\_NAME, **null**, DATABASE\_VERSION);

}

**@Override**

**public** **void** **onCreate**(SQLiteDatabase db) {

db.execSQL(CREATE\_DB\_TABLE);

}

**@Override**

**public** **void** **onUpgrade**(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {

db.execSQL("DROP TABLE IF EXISTS " + TABLE\_NAME);

onCreate(db);

}

}

}

**Activity calss:**  
MainActivity.java

**package** com.example.contentproviderexample;

**import** **android.app.Activity**;

**import** **android.content.ContentValues**;

**import** **android.net.Uri**;

**import** **android.os.Bundle**;

**import** **android.view.Menu**;

**import** **android.view.View**;

**import** **android.widget.EditText**;

**import** **android.widget.Toast**;

**public** **class** **MainActivity** **extends** Activity {

**@Override**

**protected** **void** **onCreate**(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

**@Override**

**public** **boolean** **onCreateOptionsMenu**(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

**return** **true**;

}

**public** **void** **onClickAddName**(View view) {

ContentValues values = **new** ContentValues();

values.put(MyProvider.name, ((EditText) findViewById(R.id.txtName))

.getText().toString());

Uri uri = getContentResolver().insert(MyProvider.CONTENT\_URI, values);

Toast.makeText(getBaseContext(), "New record inserted", Toast.LENGTH\_LONG)

.show();

}

}

**AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.contentproviderexample"

android:versionCode="1"

android:versionName="1.0" >

<uses-sdk

android:minSdkVersion="8"

android:targetSdkVersion="16" />

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme" >

<activity

android:name="com.example.contentproviderexample.MainActivity"

android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<provider

android:name=".MyProvider"

android:authorities="com.example.contentproviderexample.MyProvider"

android:exported="true"

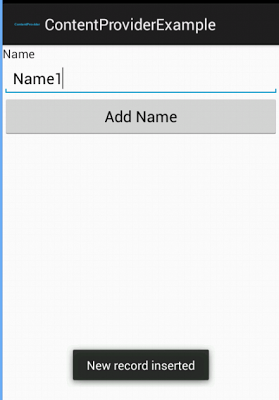
android:multiprocess="true" >

</provider>

</application>

</manifest>

**Output screenshots:**

[](http://4.bp.blogspot.com/-KxUhRli8O_4/UqYgLPj2V6I/AAAAAAAAAiM/kLW2RpxTJs0/s1600/1.png)

**2. Application 'MyUser'**

This application is very simple, it will use the query() method to access the data stored in the 'MyProvider' application. This application is having a layout to display the data.  
  
**Layout xml file:**

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<Button

android:id="@+id/btnRetrieve"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:onClick="onClickDisplayNames"

android:text="Display names" />

<TextView

android:id="@+id/res"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="20dp"

android:clickable="false"

android:ems="10" >

</TextView>

</LinearLayout>

**Activity class:**  
MainActivity.java  
We have used CusrsorLoader to load the data.

**package** com.example.contentprovideruser;

**import** **android.database.Cursor**;

**import** **android.net.Uri**;

**import** **android.os.Bundle**;

**import** **android.support.v4.app.FragmentActivity**;

**import** **android.support.v4.app.LoaderManager**;

**import** **android.support.v4.content.CursorLoader**;

**import** **android.support.v4.content.Loader**;

**import** **android.view.Menu**;

**import** **android.view.View**;

**import** **android.widget.TextView**;

**public** **class** **MainActivity** **extends** FragmentActivity **implements** LoaderManager.LoaderCallbacks<Cursor> {

TextView resultView=**null**;

CursorLoader cursorLoader;

**@Override**

**protected** **void** **onCreate**(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

resultView= (TextView) findViewById(R.id.res);

}

**@Override**

**public** **boolean** **onCreateOptionsMenu**(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

**return** **true**;

}

**public** **void** **onClickDisplayNames**(View view) {

getSupportLoaderManager().initLoader(**1**, **null**, **this**);

}

**@Override**

**public** Loader<Cursor> **onCreateLoader**(**int** arg0, Bundle arg1) {

cursorLoader= **new** CursorLoader(**this**, Uri.parse("content://com.example.contentproviderexample.MyProvider/cte"), **null**, **null**, **null**, **null**);

**return** cursorLoader;

}

**@Override**

**public** **void** **onLoadFinished**(Loader<Cursor> arg0, Cursor cursor) {

cursor.moveToFirst();

StringBuilder res=**new** StringBuilder();

**while** (!cursor.isAfterLast()) {

res.append("\n"+cursor.getString(cursor.getColumnIndex("id"))+ "-"+ cursor.getString(cursor.getColumnIndex("name")));

cursor.moveToNext();

}

resultView.setText(res);

}

**@Override**

**public** **void** **onLoaderReset**(Loader<Cursor> arg0) {

// TODO Auto-generated method stub

}

**@Override**

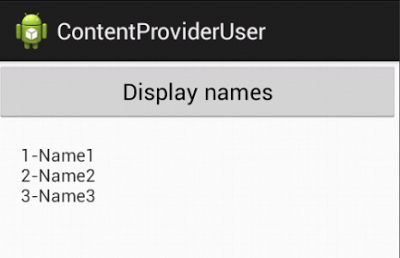
**public** **void** **onDestroy**() {

**super**.onDestroy();

}

}

**Output Screenshots:**

[](http://4.bp.blogspot.com/-hfU5QC1aOoU/UqYgA_OHzGI/AAAAAAAAAiE/5vUjv1bVq3E/s1600/2.png)

**Source code** of content provider application: 'MyProvider'  
[**MyProvider.zip**](https://sites.google.com/site/compiletimeerrorcom/android-programming/MyProvider.zip)  
  
**Source code** of the user application: 'MyUser'  
[**MyUser.zip**](https://sites.google.com/site/compiletimeerrorcom/android-programming/MyUser.zip)