系统ContentProvider

ContentProvider简介

（1）数据库在Android当中是私有的，不能将数据库设为WORLD\_READABLE，每个数据库都只能被创建它的包访问。

（2）只有创建这个数据库的应用程序才可访问它。

（3）数据库不能跨越进程和包的边界，直接访问别的应用程序的数据库

（4）如果需要在进程间传递数据，可以使用ContentProvider来实现。

ContentProvider的功能和意义

（1）为了在应用程序之间交换数据，Android提供了ContentProvider

（2）ContentProvider是不同应用程序之间进行数据交换的标准API

（3）当一个应用程序需要把自己的数据暴露给其他应用程序使用时，该应用程序可以通过提供ContentProvider来实现；

（4）其他应用程序需要使用这些数据时，可以通过ContentResolver来操作ContentProvider暴露的数据。



ContentProvider的使用

（1）统一资源标识（URI）

（2）每个ContentProvider都拥有一个公共的URI，这个URI代表其共享的数据

（3）典型的URI结构如下

**content:// com.iotek.exampleprovider/ student/12**

模式：类似协议。内容URI的模式始终是content://

命名空间：由ContentProvider规定，用来区分不同的ContentProvider

路径：用来确定数据集合，可包含多个部分，每个部分由/隔开，一般由两 部分构成student表示某个数据集合，12表示该数据集中的某条数据

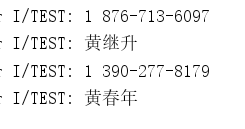
使用系统ContentProvider管理联系人

系统对联系人管理的主要Uri

ContactsContract.Contacts.CONTENT\_URI 联系人名字

ContactsContract.CommonDataKinds.Phone.CONTENT\_URI 联系人电话

2.X之后的版本：ContactsContract.DATA.CONTENT\_URI



com.android.providers.contacts

错误整理：

得到的Cursor结果集，Cursor游标是指向第一条记录之前，也就是-1的位置。

activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
  
 <**TextView  
 android:id="@+id/textView"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="手机联系人"  
 android:textColor="#F00"  
 android:textSize="30dp"**/>  
  
 <**ListView  
 android:id="@+id/listView1"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_alignParentTop="true"** />  
</**LinearLayout**>

contact\_item.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"**>  
  
  
 <**TextView  
 android:id="@+id/text1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Name"  
 android:textAppearance="?android:attr/textAppearanceLarge"**/>  
  
 <**TextView  
 android:id="@+id/text2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Phone"  
 android:textAppearance="?android:attr/textAppearanceLarge"**/>  
</**LinearLayout**>

User.java

**package** com.hjs.systemcontentproviderdemo2.contentprovider.entities;  
  
**public class** User {  
 **public** User() {  
 **super**();  
 }  
  
 **public** User(**int** \_id, String displayName, String phone) {  
 **this**.**\_id** = \_id;  
 **this**.**displayName** = displayName;  
 **this**.**phone** = phone;  
 }  
  
 **private int \_id**;  
 **private** String **displayName**;  
 **private** String **phone**;  
  
 **public int** get\_id() {  
 **return \_id**;  
 }  
  
 **public void** set\_id(**int** \_id) {  
 **this**.**\_id** = \_id;  
 }  
  
 **public** String getDisplayName() {  
 **return displayName**;  
 }  
  
 **public void** setDisplayName(String displayName) {  
 **this**.**displayName** = displayName;  
 }  
  
 **public** String getPhone() {  
 **return phone**;  
 }  
  
 **public void** setPhone(String phone) {  
 **this**.**phone** = phone;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "User{"** +  
 **"\_id="** + **\_id** +  
 **", displayName='"** + **displayName** + **'\''** +  
 **", phone='"** + **phone** + **'\''** +  
 **'}'**;  
 }  
}

MainActivity.java

**package** com.hjs.systemcontentproviderdemo2.contentprovider;  
  
**import** android.Manifest;  
**import** android.content.ContentResolver;  
**import** android.content.pm.PackageManager;  
**import** android.database.Cursor;  
**import** android.net.Uri;  
**import** android.os.Build;  
**import** android.provider.ContactsContract;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.support.v4.content.ContextCompat;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.util.Log;  
**import** android.widget.ListView;  
**import** android.widget.SimpleAdapter;  
  
**import** com.hjs.systemcontentproviderdemo2.contentprovider.entities.User;  
  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
**import** java.util.List;  
**import** java.util.Map;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** ListView **lstContacts**;  
 **private** List<User> **userList** = **new** ArrayList<User>();  
 **private** List<Map<String,String>> **data** = **new** ArrayList<Map<String,String>>(); *//这是SimpleAdapter数据参数格式所要求的List<extends Map<String,?>>* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **lstContacts**=(ListView)findViewById(R.id.***listView1***);  
 requestPower();  
 *//test();  
 //getAllContacts();* getAllContacts2();  
  
 **for** (User user:**userList**){  
 Map<String,String> item = **new** HashMap<String, String>();  
 item.put(**"name"**,user.getDisplayName());  
 item.put(**"phone"**,user.getPhone());  
 **data**.add(item);  
 }  
 SimpleAdapter adpter = **new** SimpleAdapter(**this**,  
 **data**,  
 R.layout.***contact\_item***,  
 **new** String[]{**"name"**,**"phone"**},  
 **new int**[]{R.id.***text1***,R.id.***text2***});  
  
 **lstContacts**.setAdapter(adpter);  
 }  
  
  
 *//Android官方申请动态权限写法* **public void** requestPower() {  
 *//判断是否已经赋予权限* **if** (ContextCompat.*checkSelfPermission*(**this**,  
 Manifest.permission.***READ\_CONTACTS***)  
 != PackageManager.***PERMISSION\_GRANTED***) {  
 *//如果应用之前请求过此权限但用户拒绝了请求，此方法将返回 true。即第一次拒绝，就不再请求* **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(**this**,  
 Manifest.permission.***READ\_CONTACTS***)) {*//这里可以写个对话框之类的项向用户解释为什么要申请权限，并在对话框的确认键后续再次申请权限.它在用户选择"不再询问"的情况下返回false* } **else** {  
 *//申请权限，字符串数组内是一个或多个要申请的权限，1是申请权限结果的返回参数，在onRequestPermissionsResult可以得知申请结果* ActivityCompat.*requestPermissions*(**this**,  
 **new** String[]{Manifest.permission.***READ\_CONTACTS***}, 1);  
 }  
 }  
 }  
  
  
  
 *//使用新版本完成查询* **public void** getAllContacts2(){  
 Uri uri = ContactsContract.Data.***CONTENT\_URI***;  
 *//所有的数据一般存在在data1(电话、Email、联系人...)  
 //查询data1，mimetype\_id，raw\_contact\_id* Cursor cursor = getContentResolver().query(uri,**null**,**null**,**null**,**"display\_name"**);  
 **int** index\_contact\_id=cursor.getColumnIndex(ContactsContract.Data.***CONTACT\_ID***);  
 **int** index\_data1 = cursor.getColumnIndex(ContactsContract.Data.***DATA1***);  
 **int** index\_mimetype = cursor.getColumnIndex(ContactsContract.Data.***MIMETYPE***);  
  
 cursor.moveToFirst();  
 User user = **new** User();  
 String id = cursor.getString(index\_contact\_id);  
 **int** tempId = Integer.*parseInt*(id);  
 user.set\_id(tempId);  
  
 **do** {  
 id=cursor.getString(index\_contact\_id);  
 **if** (Integer.*parseInt*(id) != tempId) {  
 **userList**.add(user);  
 user = **new** User();  
 tempId = Integer.*parseInt*(id);  
 user.set\_id(tempId);  
 }  
  
 String data1 = cursor.getString(index\_data1);  
 String mimetype = cursor.getString(index\_mimetype);  
  
 *//名称* **if** (mimetype.equals(**"vnd.android.cursor.item/name"**)) {  
 user.setDisplayName(data1);  
 }  
 *//电话* **if** (mimetype.equals(**"vnd.android.cursor.item/phone\_v2"**)) {  
 user.setPhone(data1);  
 }  
 }**while** (cursor.moveToNext());  
 **userList**.add(user);  
 }  
  
  
  
  
 *//使用老版本所公开的Uri完成查询* **public void** getAllContacts(){  
 ContentResolver cr = getContentResolver();  
 Uri uri = ContactsContract.Contacts.***CONTENT\_URI***;*//联系人名称* Uri phoneUri = ContactsContract.CommonDataKinds.Phone.***CONTENT\_URI***;*//电话号码* Cursor curName = cr.query(uri,**null**,**null**,**null**,**null**,**null**);  
  
 **while** (curName.moveToNext()){  
 User user = **new** User();  
 String contact\_id = curName.getString(curName.getColumnIndex(ContactsContract.Contacts.***\_ID***));  
 user.set\_id(Integer.*parseInt*(contact\_id));  
 user.setDisplayName(curName.getString(curName.getColumnIndex(ContactsContract.Contacts.***DISPLAY\_NAME***)));  
  
 Cursor curPhone = cr.query(phoneUri,  
 **null**,  
 ContactsContract.CommonDataKinds.Phone.***CONTACT\_ID***+**"=?"**,  
 **new** String[]{contact\_id},  
 **null**,  
 **null** );  
  
 *//注意：得到的Cursor结果集，curPhone游标是指向第一条记录之前的，即-1* **while**(curPhone.moveToNext()) {  
 user.setPhone(curPhone.getString(curPhone.getColumnIndex(ContactsContract.CommonDataKinds.Phone.***NUMBER***)));  
 }  
 **userList**.add(user);  
 }  
 }  
  
  
  
  
 *//测试* **public void** test(){  
  
*/\* //1.ContentResolver  
 ContentResolver cr = getContentResolver();  
 //ContentProvider公开接口 insert，update，delete，query  
 Uri uri = ContactsContract.Contacts.CONTENT\_URI;//联系人名称  
 Cursor cur = cr.query(uri,  
 new String[]{ContactsContract.Contacts.DISPLAY\_NAME},  
 null,  
 null,  
 null,  
 null);  
  
 while (cur.moveToNext()){  
 String name = cur.getString(0);  
 Log.i("display",name);  
 }\*/  
  
  
/\* ContentResolver cr = getContentResolver();  
 Uri uri = ContactsContract.CommonDataKinds.Phone.CONTENT\_URI;  
 Cursor cur = cr.query(uri,  
 new String[]{ContactsContract.CommonDataKinds.Phone.NUMBER},  
 null,  
 null,  
 null);  
  
 while(cur.moveToNext()){  
 String number = cur.getString(0);  
 Log.i("display",number);  
 }\*/  
  
 //2.x以上的版本可用* ContentResolver cr = getContentResolver();  
 Uri uri = ContactsContract.Data.***CONTENT\_URI***;  
 Cursor cur = cr.query(uri,  
 **new** String[]{ContactsContract.Data.***DATA1***},  
 **null**,  
 **null**,  
 **null**,  
 **null**);  
 **while** (cur.moveToNext()){  
 String data1 = cur.getString(0);  
 Log.*i*(**"TEST"**,data1);  
 }  
 }  
}

AndroidManifest.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.hjs.systemcontentproviderdemo2.contentprovider"**>  
  
 <**uses-sdk  
 android:maxSdkVersion="16"  
 android:minSdkVersion="16"**/>  
 <**uses-permission android:name="android.permission.READ\_CONTACTS"** />  
 <**uses-permission android:name="android.permission.WRITE\_CONTACTS"** />  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".MainActivity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>