

通过 ContentProvider 分享数据

1.1 知识解析

如果我们想通过 ContentProvider 给其他 app 分享数据，可以自己定义一个 ContentProvider 来实现。

在AndroidManifest.xml中声明一个provider
`<provider ...>...</provider>`

创建MyProvider类，继承自ContentProvider。
定义该provider提供的数据集的URI和字段名

在MyProvider中实现ContentProvider的6个
abstract method: query, insert, update,
delete, getType, onCreate

在其他app中使用ContentResolver通过URI访
问MyProvider提供的数据库

这里重点在于定义自己的 ContentProvider 类，需要继承 ContentProvider 并重写下面方法：

`Uri insert(Uri uri, ContentValues values)`，将新增的数据以 Uri 的方式返回。

需要使用 UriMatcher 对象去区分不同的 Uri，进行不同的数据库操作，下同

`int delete(Uri uri, String selection, String[] selectionArgs)`：将删除的数据记录数返回

`int update(Uri uri, ContentValues values, String selection, String[] selectionArgs)`：将修改的数据记录数返回

`Cursor query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)`: 将查询结果返回

定义 `ContentProvider` 类要点:

创建一个 `UriMatcher` 变量, 加入对 URI 的匹配定义:

- 对有 C 类字段的 URI 的匹配
- 对有 D 类字段的 URI 的匹配
- 其他 URI 的匹配

覆盖 `int getType(Uri uri)` 方法, 使用 `UriMatcher` 对 `uri` 进行匹配, 返回不同的匹配类型: 是对整个“数据库表”进行操作还是只针对单笔数据进行操作:

- `vnd.android.cursor.dir`——整个“表”
- `vnd.android.cursor.item`——单一数据

最后, 需要在 `AndroidManifest.xml` 中对它进行配置:

```
<provider android:authorities= "list"
    android:enabled=[ "true" | "false" ]
    android:exported=[ "true" | "false" ]
    //API17 后, 如果要暴露给其他应用, 则必须设置为 true。
    android:name= "string"
    android:readPermission= "string"
    android:writePermission= "string" >
</provider>
```

这样就可以在其他 app 中访问你的应用中的数据了。

例如:

```
ContentResolver contentResolver = this.getContentResolver();
Uri insertUri =
Uri.parse("content://com.exam.db.teacherProvider/teacher");//uri
要通过文档告诉第三方 app 开发者
ContentValues values = new ContentValues();
values.put("name", et1.getText().toString());//可以操作的
字段名字如 name/title 等, 需要通过文档告诉第三方 app 者
```

```
values.put("title", et2.getText().toString());

values.put("sex", rb1.isChecked());

values.put("date_added", "2015-08-18 12:23:32");

Uri uri = contentResolver.insert(insertUri, values);
```

1.2 功能演示



1.4 职业素质

ContentProvider：为存储和获取数据提供统一的接口。可以在不同的应用程序之间共享数据。Android 已经为常见的一些数据提供了默认的 ContentProvider