# ContentProvider总结

目录

[ContentProvider总结 1](#_Toc43582437)

[一、内容提供器 2](#_Toc43582438)

[二、访问其他程序中的数据 2](#_Toc43582439)

[三、访问示例 3](#_Toc43582440)

## 一、内容提供器

内容提供器主要用于跨程序共享数据，同时保证被访问数据的安全性。

不同于文件存储和SharedPreferences存储中的两种全局可读写操作模（

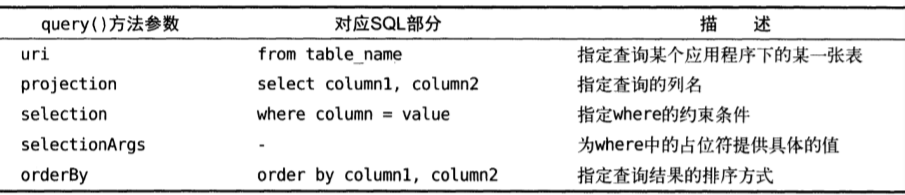
4.2开始废弃了），ContentProvider可以选择只对哪一部分的数据进行共享，从而保证程序中的隐私数据不会泄露。

## 二、访问其他程序中的数据

要访问内容提供器中共享的数据，需要使用ContentResolver类，它提供了一系列方法用于对数据增删改查。

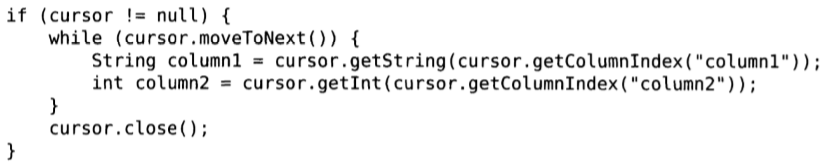
2.1 query()

参数说明：

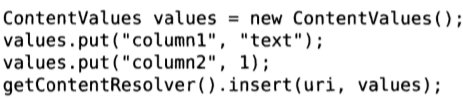




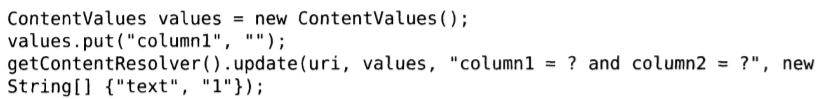
查询后返回一个Cursor对象。



2.2 insert()



2.3 update()

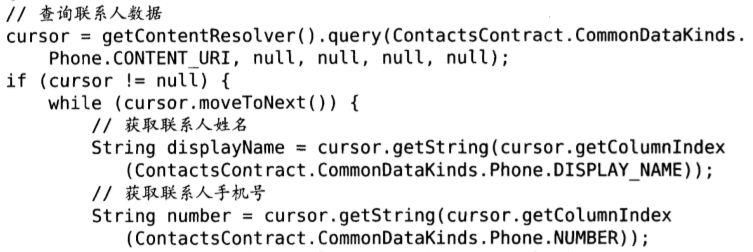


* 1. delete()



## 三、访问示例

3.1 读取联系人



3.2 读取sd卡中的视频文件



小结：使用ContentResolver查询数据时，有两个核心参数：uri，projection。

在获取系统的数据时，Uri和projection(列名)通过对应的类获取，比如联系人信息，由ContactsContract提供；视频文件信息由MediaStore.Video提供。

四、创建内容提供者

要创建内容提供者，需要继承ContentProvider，并实现其中的方法。

1. onCreate()

初始化内容提供器的时候调用。当有ContentResolver访问程序数据时，内

容提供器才会被初始化。通常在此完成数据库的创建和升级，返回值表示是否成功。

增删改查，使用的都是SQLiteDatebase提供的方法。需要注意的是，不同的uri有一点区别：

URI的两种主要写法：

调用方期望访问com.example.app中的table表。

content://com.example.app.provider/table

调用方期望访问com.example.app中的table表中id为1的数据。

content://com.example.app.provider/table/1

可以通过通配符的方式来匹配这两种格式的URI。

\*：匹配任意长度的任意字符。

#：匹配任意长度的数字。

匹配任意表的URI的格式：

content://com.example.app.provider/\*

匹配table表中的任意一行的URI格式：

Content://com.example.app.provider/table/#

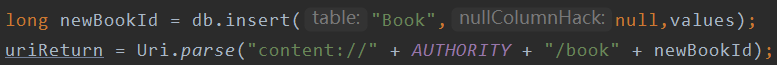
因此，在使用增删改查的四个方法时，需要先判断一下uir的类型，对于查看全表的，用标准方法即可。对于带id的，需要先获取id，在根据id去操作。

获取id：uro.getPathSegments().get(1);

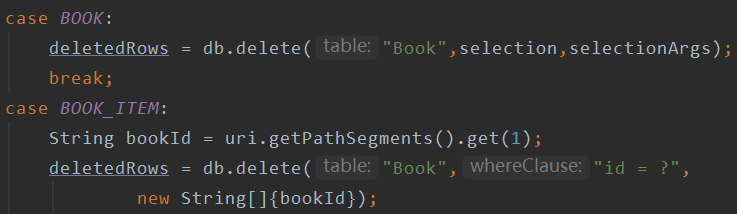
使用此方法可以获取到uri最后的id。

四种操作具体如下：（执行完相应操作后，需要给出返回值）

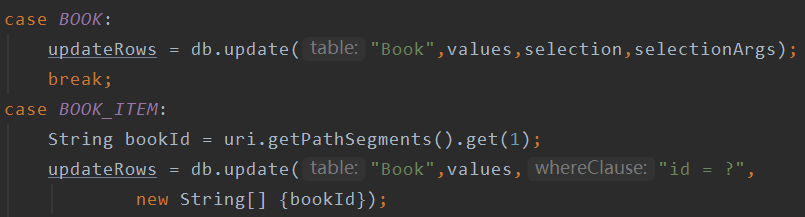
1. insert()



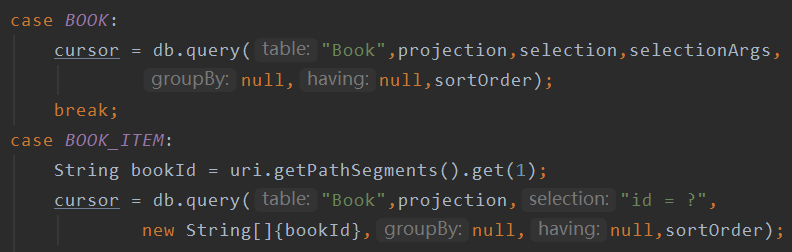
1. delete()



1. update()

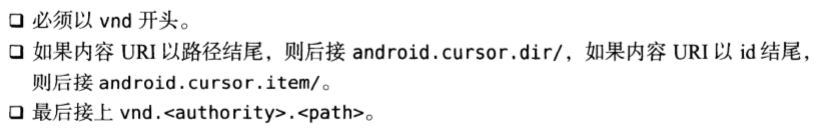


1. query()



1. getType()

这个方法要返回一个uri的MIME类型。是一个字符串，格式如下。



四、总结

ContentProvider作为跨程序共享数据的途径之一，在获取数据的程序中，核心在于ContentResolver类，使用这个类可以获取其他程序中的数据。

对于提供数据的程序，需要在ContentProvider提供的方法中重写具体的操作逻辑，这也是实现控制数据安全的关键。每个方法都有对应的返回值，这些返回值将作为结果提供给请求方。

ContentResolver、ContentProvider、数据库或者文件之间的关系：

ContentProvider算是一个守卫者，它看管着数据，外界（其他程序）想要获取数据（实施者是ContentResolver），必须经过它的同意（判断uri），并根据请求，执行相应操作，最终返回数据。这样就可以在保证数据安全的情况下，对外提供数据。