
ContentProvider 的基础知识

1. ContentProvider 简介

Content Providers 管理访问结构化的数据集。它们可以封装这些数据，并且为定义安全的数据提供机制。Content providers 是标准的接口，它能将一个线程中的数据与其他线程中的运行的代码进行连接。也就是说 Content providers 支持跨应用间的访问。

2. ContentProvider 的使用场景

当你想要在 Content provider[内容提供者] 中访问数据，你要在你的应用程序的 Context 中使用 ContentResolver[内容解析者] 对象与 provider 进行连接作为客户端[通过 Context 中 getContentResolver()方法]。这个 ContentResolver 会与 provider 对象进行连接，provider 它是一个继承 Content provider 的类的实例。provide 这个对象会接受从客户端请求的数据，执行请求动作，返回结果。

如果你不打算将自己的数据分享给其他应用程序，就不必要开发自己的 内容提供者[提供者]，然而你想要在你的应用程序中提供自定义的搜索建议你需开发内容提供者，如果你想要复制数据或者文件到其它应用程序，你就需要拥有自己的 provider。

Android 自己提供了一个 content providers 来管理数据，如果 音频，视频，或者个人信息等。你可以在 android.provider 包中查看，这些提供者可以给任何应用程序所访问问题

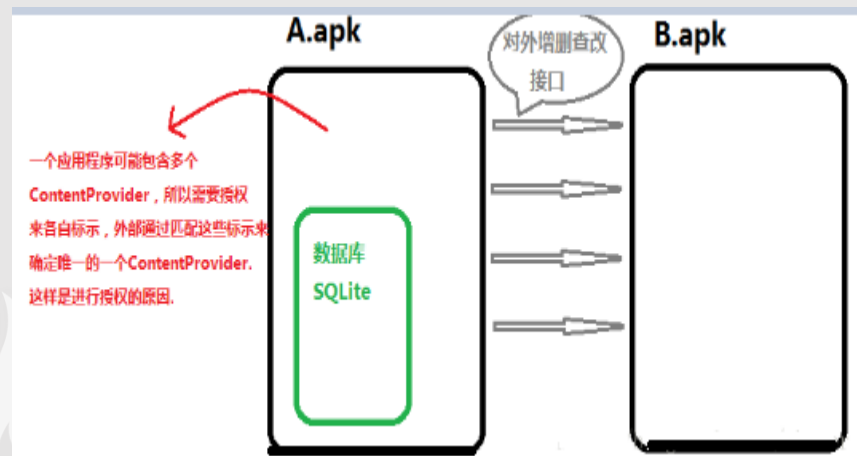
3. 如何创建 ContentProvider

A. 什么时候决定创建一个内容提供者呢？

- 1). 你需要提供完整的数据和文件给其他的应用程序.
- 2). 你想要允许用户从你的应用程序复制完整的数据到其他的应用程序

3). 你想要使用搜索框架来提供自定义的搜索建议。

B. 创建内容提供者需要掌握的知识



C. 查看 API 文档中 ContentProvider 类的说明，可以发现这是一个抽象类

Content providers 是 Android 应用程序中主要的组成部分之一，为应用程序提供对外的内容，它封装了数据然后提供给那些通过简单 ContentResolver 接口的应用程序，这些数据是可以跨应用访问的。

对与解析者来说，它会通过 ContentResolver 会发出一个请求，这个请求是否通过是通过去检查由系统给予的 URI 的授权，授权是由 content provider 注册的，在 AndroidManifest.xml 中去注册这种授权。UriMatcher 类可以帮助解析 URI。

D. 在这里为什么要注册 URI 授权呢？

原因是一个应用程序中可以有多个 content provider，但是每一个 content provider 的授权都是唯一的，所以内容解析者可以通过这种 URL 的授权来找到自己想要的 content provider。这里的 URI 的授权也可以理解为 content provider 对外的一个访问的路径。

E. 注意定义好一个内容提供者之后需要在 AndroidManifest.xml 中

注册

`android:authorities` 授权属性，表示外部应用程序访问当前内容提供者的标识符，它是自定义的，一般我们是以“包名 + 类名”的形式来定义的。有些人可以理解为 URI 路径

F. 查看 UriMatcher 类的概要描述

这是一个在 content provider 中帮助匹配 URIs 的实用类。

查看 `public void addURI (String authority, String path, int code)` 方法

这个方法是用来表示在 content provider 里面添加外部对其的匹配的规则，当 URI 被匹配的时候，就会返回 code 码，URI 节点可以精确的匹配字符串，“*”号匹配任何的字符串“#”号只能匹配数字。[比如删除一条记录中 ID 往往是数字]

G. 参数说明

authority：授权，就是 AndroidManifest.xml 中的授权

path：匹配路径(通常是一个表名)[* 可以作为匹配任意字符的通配符，# 可以作为匹配数字的通配符]。

【注意这里如果是单条记录，需要添加 /# 标示符】

H. 查看 content provider 中的 `public abstract String getType (Uri uri)` 方法

根据给定的 URI 来实现处理 MIME 类型的请求，对于单条记录返回的 MIME 类型是以 `vnd.android.cursor.item` 开始的，对于多条记录返回的 MIME 类型是以 `vnd.android.cursor.dir/` 开始的。这个方法可以在多线程环境下被调用。详细参考 Processes and Threads.

I. 查看 content provider 中 public abstract Uri insert (Uri uri, ContentValues values) 方法

实现这个方法来处理插入一个新行的请求，在插入后可以友好的调用 notifyChange() 方法。

J. 参数说明：

uri：这种格式 "content:// URI" 的插入请求[后续会认真的讲解这一部分内容]。此处不为空

values：添加到数据库的 ContentValues 类型集合。博客前面章节讲过次内容，详情可以去参考。此处不为空

返回 新插入选项的 URI，可以给其他用户去使用。

K. 删除操作

查看 content provider 的 public abstract int delete (Uri uri, String selection, String[] selectionArgs) 方法

实现这个方法主要是用来处理删除一行或者多行的请求操作,实现这个删除操作需要 有 selection 语句，你可以在删除之后调用 notifyDelete() 方法来友好的提示

实现这个方法还需要在 URI 的末尾解析出行的 ID，如果一个指定的行被删除之后，例如，客户端在创建 SQL 语句的时候会通过 content://contacts/people/22 的这个 URI，来解析出末尾的 ID 号，确定删除 ID 为 22 的这个记录。

参数说明：

URI ： 完整的 URI 路径，包括行 ID

selection ： 可选项，在删除行的时候适用

返回 ： 影响数据库的行数

