Android ContentProvider的启动过程

参考：<https://blog.csdn.net/luoshengyang/article/details/6963418>

1. 安装应用程序时，并不会把相应的Content Provider加载到内存中来，系统采取了懒加载的机制，等到第一次要使用这个Content Provider的时候，系统才会把它加载到内存中来，下次再要使用这个Content Provider的时候，就可以直接返回了；
2. 通过context.getContentResolver()得到ContentResolver，调用ContentResolver.acqireProvider()（其实现在ApplicationContentResolve. acquireProvider()中）校验uri的scheme是否是以content://开头；通过uri.getAuthority()得到authority；
3. ApplicationContentResolve. acquireProvider()进入ActivityThread. getExistingProvider()，通过authority在保存本地已经存在的ContextProvider集合mProviderMap中查找是否存在与上一步获取的authority对应的ContextProvider；存在则直接返回，否则通过ActivityManagerService.getContentProvider()获取一个ContentProviderHolder对象的holder；
4. 在ActivityManagerService.getContentProvider()中，首先判断Content Provider是否允许在客户进程中加载（由Content Provider配置的multiprocess属性决定），然后判断该ContentProvider是否正在被其它应用程序加载；通过startProcessLocked()启动一个新进程加载Content Provider；
5. 在ActivityManagerService中，有两个成员变量是用来保存系统中的Content Provider信息的，一个是以Content Provider的authoriry值为键值来保存的mProvidersByName，一个是以Content Provider的类名为键值来保存的mProvidersByClass；一个Content Provider可以有多个authority。
6. 通过ActivityManagerService.attachApplicationLocked()调用ApplicationThread.bindApplication()，将相关信息封装成AppBindData对象，以消息的形式发送到主线程，最终由ActivityThread.handleBindApplication()处理。
7. ActivityThread.handleBindApplication()中通过 ActivityThread.install- ContentProviders()将Content Proivder的信息安装在本地；并且为每个Content Provider创建一个ContentProviderHolder对象来保存相关的信息；
8. 调用ActivityManagerService.publishContentProviders()来通知ActivityManagerService服务，这个进程中所要加载的Content Provider；
9. ActivityThread.installProvider()通过ClassLoader在应用程序进程中把相应的Content Provider类加载进来
10. 调用ContentProvider.getIContentProvider()获取一个Binder对象，并保存在ActivityManagerService中，其他应用程序通过该Binder与相应的Content Provider通信；
11. 调用ContentProvider.attachInfo()调用Content Provider的onCreate()执行初始化的工作；