安卓面试突破专题课程

**Android 基础与底层机制**

1. 数据库的操作类型有哪些，如何导入外部数据库？（Ricky）

读懂题目。如果碰到问题比较模糊的时候可以适当问问面试官。

配合面试官来面试：面试是一个相互了解的过程，要充分利用面试的题目和时间把自己的能力和技术展现出来，面试官能够看到你的真实技术。

1. 使用数据库的方式有哪些？
2. openOrCreateDatabase(String path);
3. 继承SqliteOpenHelper类对数据库及其版本进行管理(onCreate,onUpgrade)

当在程序当中调用这个类的方法getWritableDatabase()或者getReadableDatabase();的时候才会打开数据库。如果当时没有数据库文件的时候，系统就会自动生成一个数据库。

1. 操作的类型：增删改查CRUD

直接操作SQL语句：SQliteDatabase.execSQL(sql);

面向对象的操作方式：SQLiteDatabase.insert(table, nullColumnHack, ContentValues);

如何导入外部数据库？

一般外部数据库文件可能放在SD卡或者res/raw或者assets目录下面。

写一个DBManager的类来管理，数据库文件搬家，先把数据库文件复制到”/data/data/包名/databases/”目录下面，然后通过db.openOrCreateDatabase(db文件),打开数据库使用。

我上一个项目就是这么做的，由于app上架之前就有一些初始数据需要内置，也会碰到数据的升级等问题，我是这么做的…… 同时我碰到最有意思的问题就是关于数据库并发操作的问题，比如：多线程操作数据库的时候，我采取的是封装使用互斥锁来解决……

1. 是否使用过本地广播，和全局广播有什么差别？（Ricky）

引入本地广播的机制是为了解决安全性的问题：

1. 正在发送的广播不会脱离应用程序，比用担心app的数据泄露；
2. 其他的程序无法发送到我的应用程序内部，不担心安全漏洞。（比如：如何做一个杀不死的服务---监听火的app 比如微信、友盟、极光的广播，来启动自己。）
3. 发送本地广播比发送全局的广播高效。（全局广播要维护的广播集合表 效率更低。全局广播，意味着可以跨进程，就需要底层的支持。）

本地广播不能用静态注册。----静态注册：可以做到程序停止后还能监听。

使用：

1. 注册

LocalBroadcastManager.getInstance(this).registerReceiver(new XXXBroadCastReceiver(), new IntentFilter(action));

1. 取消注册：

LocalBroadcastManager.getInstance(this).unregisterReceiver(receiver)

1. 是否使用过 IntentService，作用是什么， AIDL 解决了什么问题？ (小米) （Ricky）

如果有一个任务，可以分成很多个子任务，需要按照顺序来完成，如果需要放到一个服务中完成，那么使用IntentService是最好的选择。

一般我们所使用的Service是运行在主线程当中的，所以在service里面编写耗时的操作代码，则会卡主线程会ANR。为了解决这样的问题，谷歌引入了IntentService.

IntentService的优点：

1. 它创建一个独立的工作线程来处理所有一个一个intent。
2. 创建了一个工作队列，来逐个发送intent给onHandleIntent()
3. 不需要主动调用stopSelf()来结束服务，因为源码里面自己实现了自动关闭。
4. 默认实现了onBind()返回的null。
5. 默认实现的onStartCommand()的目的是将intent插入到工作队列。

总结：使用IntentService的好处有哪些。首先，省去了手动开线程的麻烦；第二，不用手动停止service；第三，由于设计了工作队列，可以启动多次---startService(),但是只有一个service实例和一个工作线程。一个一个熟悉怒执行。

**AIDL 解决了什么问题？**

AIDL的全称：Android Interface Definition Language，安卓接口定义语言。

由于Android系统中的进程之间不能共享内存，所以需要提供一些机制在不同的进程之间进行数据通信。

远程过程调用：RPC—Remote Procedure Call。 安卓就是提供了一种IDL的解决方案来公开自己的服务接口。AIDL:可以理解为双方的一个协议合同。双方都要持有这份协议---文本协议 xxx.aidl文件（安卓内部编译的时候会将aidl协议翻译生成一个xxx.java文件---代理模式：Binder驱动有关的，Linux底层通讯有关的。）

在系统源码里面有大量用到aidl，比如系统服务。

电视机顶盒系统开发。你的服务要暴露给别的开发者来使用。

讲解Binder机制。

1. Activity、 Window、 View 三者的差别， fragment 的特点？（360）（Ricky）

Activity、 Window、 View 三者如何协同显示界面的。---考点：显示的过程(view 绘制流程)源码的熟悉度。

Activity剪窗花的人（控制的）；Window窗户（承载的一个模型）；View窗花（要显示的视图View）；LayoutInflater剪刀---将布局（图纸）剪成窗花。

（Alt+方向箭头）

**fragment 的特点？（你用fragment有没有领略到一些乐趣，或者有没有踩过什么坑？）**

fragment的设计主要是把Activity界面包括其逻辑打碎成很多个独立的模块，这样便于模块的重用和更灵活地组装呈现多样的界面。

1. Fragment可以作为Activity界面的一个部分组成；
2. 可以在一个Activity里面出现多个Fragment，并且一个fragment可以在多个Activity中使用；
3. 在Activity运行中，可以动态地添加、删除、替换Fragment。
4. Fragment有自己的生命周期的，可以响应输入事件。

踩过的坑：1.重叠；2. 注解newAPI（兼容包解决）；3. Setarguement()初始化数据;4. 不能在onsave...（）方法后，commit; 5. 入栈出栈问题; --事务。像Activity跳转一样的效果，同时返回的时候还能回到之前的页面(fragment)并且状态都还在。6.replace(f1,f2)严重影响生命周期:add()+show+hide

5. 描述一次网络请求的流程（新浪）（Jason）

6. Handler、 Thread 和 HandlerThread 的差别（小米）（Jason）

7. 低版本 SDK 实现高版本 api（小米）（Ricky）

两种情况：

1. 一般很多高版本的新的API都会在兼容包里面找到替代的实现。比如fragment。

Notification，在v4兼容包里面有NotificationCompat类。5.0+出现的backgroundTint，minSdk小于5.0的话会包检测错误，v4兼容包DrawableCompat类。

1. 没有替代实现就自己手动实现。比如：控件的水波纹效果—第三方实现。

或者直接在低版本去除这个效果。

1. 补充:如果设置了minSDK但是代码里面使**用了高**版本的API，会出现检测错误。需要在代码里面使用声明编译检测策略，比如：@SuppressLint和@TargetApi注解提示编译器编译的规则。@SuppressLint是忽略检测；@TargetApi=23，会根据你函数里面使用的API，严格地匹配SDK版本，给出相应的编译错误提示。
2. 为了避免位置的错误，最好不要使用废弃api。（一般情况下不会有兼容性问题，后面可能会随时删除这个API方法；性能方面的问题。）
3. <http://chinagdg.org/2016/01/picking-your-compilesdkversion-minsdkversion-targetsdkversion/>

8. launch mode 应用场景（百度、小米、乐视）（Ricky）

栈：先进后出

标准模式

SingleTop：使用场景：浏览器的书签；通讯消息聊天界面。

SingleTask：使用场景：某个Activity当做主界面的时候。

SingleInstance：使用场景：比如浏览器BrowserActivity很耗内存，很多app都会要调用它，这样就可以把该Activity设置成单例模式。比如：闹钟闹铃。

9. touch 事件传递流程（小米）（Ricky）

10. view 绘制流程（百度）（Ricky）

Measure：测量，测量自己。如果是ViewGroup就需要测量里面的所有childview.

测量的结果怎么办？setMeasuredDimension(resolveSizeAndState(maxWidth, widthMeasureSpec, childState), heightSizeAndState);设置自己的大小。

Layout: 摆放，把自己摆放在哪个位置。如果是ViewGroup就需要发放里面的所有childview.

怎么去具体摆放呢？

Draw:绘制

/\*

\* Draw traversal performs several drawing steps which must be executed

\* in the appropriate order:

\*

\* 1. Draw the background

\* 2. If necessary, save the canvas' layers to prepare for fading

\* 3. Draw view's content

\* 4. Draw children

\* 5. If necessary, draw the fading edges and restore layers

\* 6. Draw decorations (scrollbars for instance)

\*/

11. 什么情况导致内存泄漏（美团）（Ricky）

12. ANR 定位和修正（Ricky）

13. 什么情况导致 oom（乐视、美团）（Ricky）

14. Android Service 与 Activity 之间通信的几种方式（Ricky）

15. Android 各个版本 API 的区别（Ricky）

16. 如何保证一个后台服务不被杀死,比较省电的方式是什么？（百度）（Ricky）

17. Requestlayout， onlayout， onDraw， DrawChild 区别与联系（猎豹）（Ricky）

18. invalidate()和 postInvalidate() 的区别及使用（百度）（Ricky）

19. Android 动画框架实现原理（Ricky）

20. Android 为每个应用程序分配的内存大小是多少？（美团）（Ricky）

22. LinearLayout 对比 RelativeLayout（百度）（Ricky）

23. 优化自定义 view（百度、乐视、小米）（Ricky）

24. ContentProvider（乐视）（Ricky）

25. fragment 生命周期（Ricky）

26. volley 解析（美团、乐视）（Ricky）

27. Android Glide 源码解析（Ricky）

28. Android 属性动画特性（乐视、小米）（Ricky）