**Android 使用Intent传递大量数据问题总结**

**前言**

平时我们在 Android 组件之间传递数据一般使用Intent都能解决,但是在传递的数据较大时（比如一个size>1000的列表），Intent就不能用了，如果非要用的话就会崩溃：TransactionTooLargeException。 [官方文档如是说：](https://link.juejin.cn/?target=https%3A%2F%2Fdeveloper.android.google.cn%2Fguide%2Fcomponents%2Factivities%2Fparcelables-and-bundles)

通过 intent 发送数据时，应小心地将数据大小限制为几 KB。发送过多数据会导致系统抛出 TransactionTooLargeException 异常。

Intent 无法传递大数据是因为其内部使用了 Binder 通信机制，Binder 事务缓冲区限制了传递数据的大小。Binder 事务缓冲区的大小限定在 1MB，但是这个尺寸是共享的，也就是并不是传递 1MB 以下的数据就绝对安全，要视当前的环境而定。 不要挑战 Intent 传递数据大小的极限，对于大数据，例如长字符串、Bitmap 等，不要考虑 Intent 传递数据的方案。下面介绍几种替代的解决方案。

**使用单例**

代码如下，比较简单，在此就不再多介绍了，只需要在传递时 set,在获取时 get。

public class MusicListHolder {

private ArrayList<MusicInfo> musicInfoList;

public ArrayList<MusicInfo> getMusicInfoList() {

return musicInfoList;

}

public void setMusicInfoList(ArrayList<MusicInfo> musicInfoList) {

this.musicInfoList = musicInfoList;

}

private static final MusicListHolder holder = new MusicListHolder();

public static MusicListHolder getInstance() {

return holder;

}

}

复制代码

注意事项：这种方法不可用于多进程，因为不同的进程获取到的单例并非同一个单例，也就是获取不到数据。

**使用EventBus**

EventBus 是一个 Android 端优化的 Publish/subscribe 消息总线，简化了应用程序内各个组件间、组件与后台线程间的通信。在《阿里巴巴Android开发手册》中也有推荐：“Activity 间的数据通信，对于数据量比较大的，避免使用 Intent + Parcelable 的方式，可以考虑 EventBus 等替代方案，以免造成 TransactionTooLargeException。”。具体使用可参见：[EventBus](https://link.juejin.cn/?target=https%3A%2F%2Fgithub.com%2Fgreenrobot%2FEventBus)

**使用 Application**

将数据保存在Application中，用的时候取出来。这样整个应用都能够读写这个数据。使用很方便，在此就不多讲了。但是使用时是有一些问题的要注意的。

有时候因为内存不足等原因，我们的应用会被系统强制杀死，此时再次点击进入应用时，系统会直接进入被杀死前的那个界面。但是此时Application却是新创建的，我们也就无法拿到之前存取的数据，如果不加以判断，则会导致空对象的问题。

**使用建议：**

* 使用时一定要做好非空判断
* 如果数据为空，可以考虑逻辑上让应用直接返回到最初的Activity。

**持久化数据**

将数据保存在文件里。常见的手段有sqlite、shared preference、file等。

**优点**：

* 应用中所有地方都可以访问
* 不会轻易丢失

**缺点**：

* 操作麻烦
* 效率低下
* 读取易出错