## AIDL Android Interface Define Language

主要用于进程间多任务分发通信

如果只是进程间通信，不需要多任务，可以使用Messenger替代

如果在同一个进程中，可以考虑使用Binder替代

官方介绍地址  
<https://developer.android.google.cn/guide/components/aidl>

AIDL类似于C语言中的头文件左右，定义可供C/S双方使用的序列化类，接口

oneway关键字用于确保来自远程的调用方立刻返回，不在阻塞等待执行结果，被调用方无法确保来自不同线程的调用次序

## 可支持的传输格式包括

Java基本型

String CharSequence 封装类型

List 内容为上述两种及自定义的Parcelabel类和接口，可以包含泛型String，实际类Array List

Map 内容为上述基本型及自定义的Parcelabel类和接口，不能包含泛型如Integer 实际类是HashMap

对于自定义的类型和接口，即使在同一包内，也需要import语句导入

## 定义服务接口注意事项

AIDL将对象编组成可并入进程的原语，以供其他进程组装调用

In表示客户端输入，out表示服务器输出，inout表示两者皆可修改

所有的原语对象参数方向必须为in类型

非原语对象可以自定义方向

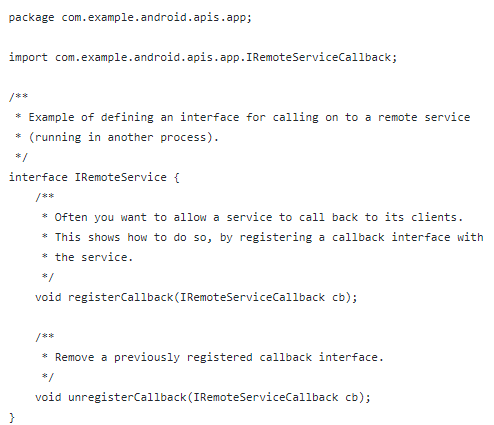
编组开销较大，确保方向正确

可以定义String或者int类型常量

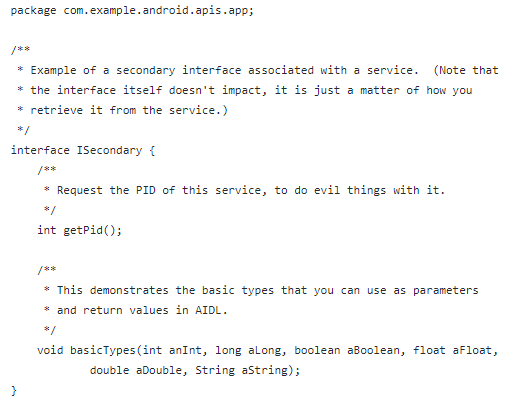
## 主要步骤，以APIDEMO中的Remote Service为例

1 定义接口

定义在.aidl文件中，该文件需要c/s双方共同放置到src/aidl目录下，编译工具会在build中间过程gen目录下产生对应的java文件，该java文件用于参与编译开发使用。以IRemoteService.aidl为例，定义了客户端可操作的方法，比如注册和注销回调函数，这个是用于观察者模式的一种结构，目的是让服务能够在特殊时刻回调客户端的方法。



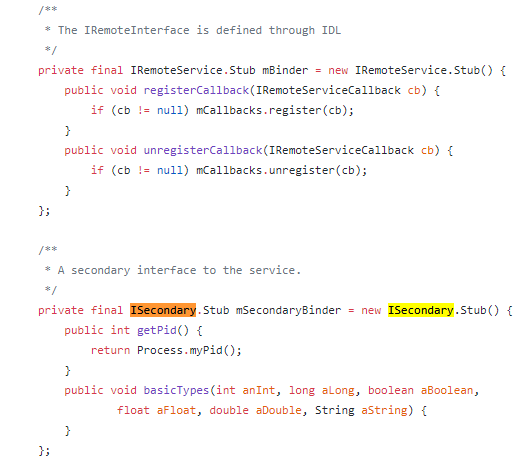
在举个例子



这个定义了方法，用于返回进程名和处理一些基本类型变量，因为没有用到自定义的接口或者类，所以没有import语句。

2 实现接口

实现该aidl接口的代码较为直接，也就是创建binder类对象，Stub是存留在服务器端的binder对象，可以定义提供的服务接口的具体实现，示例如下



Stub是AIDL接口的内部抽象实现类，用于驻留在服务端接收来自客户端代理的数据流，根据不同的数据流来实现特定的方法和处理结果。

3 暴露接口

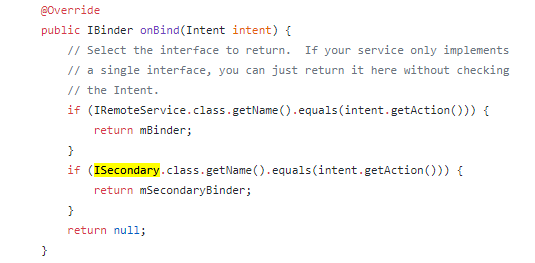
实现的接口需要暴露给客户端，通过绑定服务的方式，

当绑定时，服务返回特定类型的Stub对象，

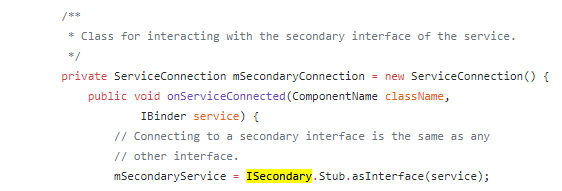
同一个服务可以返回若干个binder对象

不能确保调用方是否来自ui线程，所以需要注意防止ANR

服务方的任何异常都不会回传给调用方



返回的binder对象，调用方会得到接口类型的proxy对象，代理对象和stub对象直接存在直接沟通处理的方式，可以通过分析aidl生产的java文件得知，代理对象也是一种接口类型的对象，其组装数据并调用mRemote（也就是stub对象）的交易方法，双方沟通好形式和参数后，进行代码调用和返回



得到的接口类型对象可直接调用AIDL方法

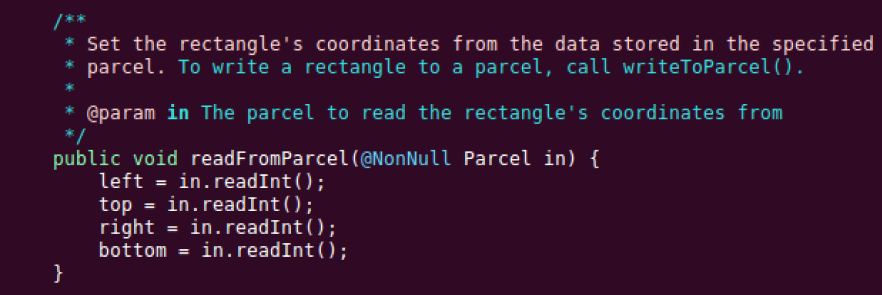
## 传递序列化对象

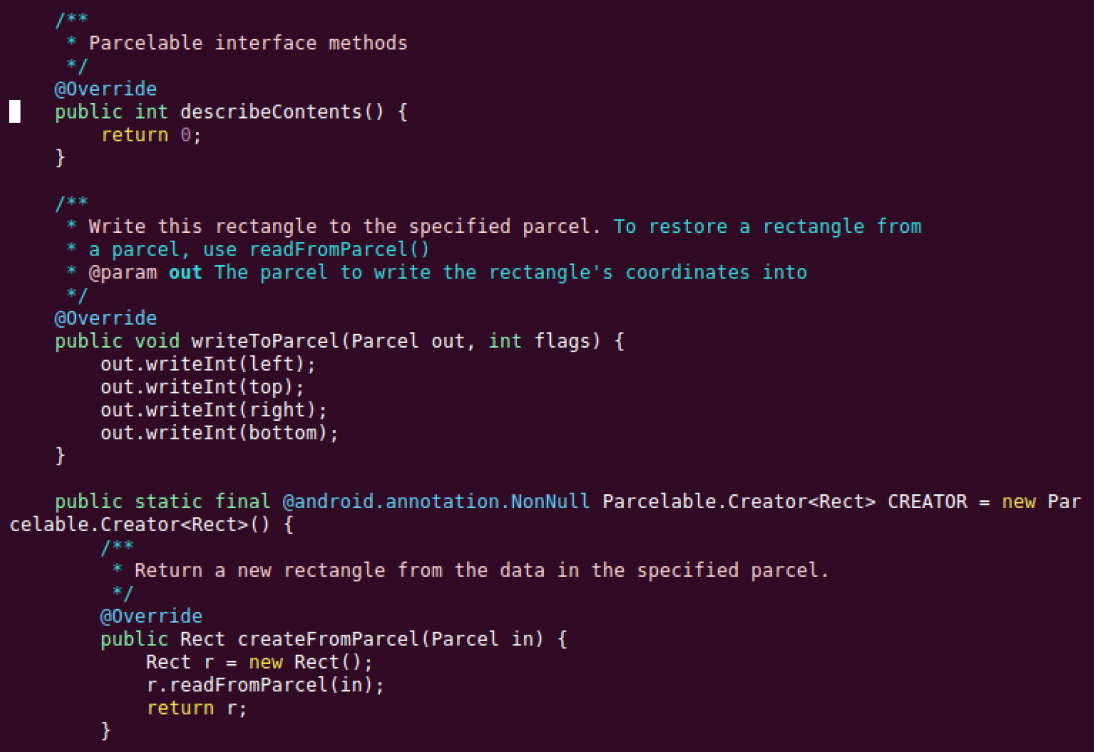
AIDL接口方式还可以使用自定义的类和接口

自定义的类必须是实现Parcelabel接口，系统通过该接口实现分解和编组

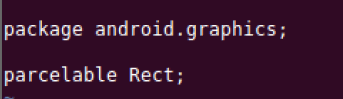
需要在单独的aidl文件中声明自定义的类

使用自定义类的aidl接口需要显示import该类





用于AIDL接口的自定义序列化的类需要声明



声明后的类可用于正常的aidl接口调用

## 使用Bundle传递自定义对象