



链滴

Android进程间通讯之messenger

作者: [asd19860](#)

原文链接: <https://ld246.com/article/1444638195677>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

平时一说进程间通讯，大家都会想到AIDL，其实messenger和AIDL作用一样，都可以进行进程间通信。它是基于消息的进程间通信，就像子线程和UI线程发送消息那样，是不是很简单，还不用去写AIDL文件，是不是有点小爽。哈哈。

此外，还支持记录客户端对象的Messenger，然后可以实现一对多的通信；甚至作为一个转接处，任两个进程都能通过服务端进行通信。

与 AIDL 比较：

当您需要执行 IPC 时，为您的接口使用 Messenger 要比使用 AIDL 实现更加简单，因为 Messenger 会将所有服务调用排入队列，而纯粹的 AIDL 接口会同时向服务发送多个请求，服务随后必须应对多线程处理。

对于大多数应用，服务不需要执行多线程处理，因此使用 Messenger 可让服务一次处理一个调用。如果您的服务必须执行多线程处理，则应使用 AIDL 来定义接口。

接下来看下怎么写：

服务端：

- 1.创建一个handler对象，并实现hanlemessage方法，用于接收来自客户端的消息，并作处理
- 2.创建一个messenger(送信人),封装handler
- 3.messenger创建一个IBinder对象，通过onBind返回给客户端

客户端：

- 1.在activity中绑定服务
- 2.创建ServiceConnection并在其中使用 IBinder 将 Messenger实例化
- 3.使用Messenger向服务端发送消息
- 4.解绑服务
- 5.服务端中在 handleMessage() 方法中接收每个 Message

这样，客户端并没有调用服务的“方法”。而客户端传递的“消息”（Message 对象）是服务在其 Handler 中接收的。

上面实现的仅仅是单向通信，即客户端给服务端发送消息，如果我需要服务端给客户端发送消息又该怎么做呢？

其实，这也是很容易实现的，下面就让我们接着上面的步骤来实现双向通信吧

- 1.在客户端中创建一个Handler对象，用于处理服务端发过来的消息
- 2.创建一个客户端自己的messenger对象，并封装handler。
- 3.将客户端的Messenger对象赋给待发送的Message对象的replyTo字段
- 4.在服务端的Handler处理Message时将客户端的Messenger解析出来，并使用客户端的Messenger对象给客户端发送消息

这样就实现了客户端和服务端的双向通信了。

注意：注：Service在声明时必须对外开放，即`android:exported="true"`；另外，我在开发完APP都用一些APP在线自动化测试工具进行测试：www.ineice.com