

# 前言

---

为使应用程序之间能够彼此通信，Android提供了IPC（Inter Process Communication，进程间通信）的一种独特实现：AIDL（Android Interface Definition Language，Android接口定义语言）。

网上看了几篇关于AIDL的文章，写得都很不错，不过例子构造大多略微复杂：建立两个Android项目，一个是client，一个是server（提供service）。

本文将通过一个项目来介绍AIDL用法，包含了service和client。可能简单了些，不过轻省许多。

## 一、AIDL介绍

---

在Android中，每个应用（Application）执行在它自己的进程中，无法直接调用到其他应用的资源，这也符合“沙箱”的理念。所谓沙箱原理，一般来说用在移动电话业务中，简单地说旨在部分地或全部地隔离应用程序。

因此，在Android中，当一个应用被执行时，一些操作是被限制的，比如访问内存，访问传感器，等等。这样做可以最大化地保护系统，免得应用程序“为所欲为”。

那我们有时需要在应用间交互，怎么办呢？于是，Android需要实现IPC协议。然而，这个协议还是有点复杂，主要因为需要实现数据管理系统（在进程或线程间传递数据）。为了暂时减缓这个“会呼吸的痛”，Android为我们实现了自己的IPC，也就是梁静茹，oh，sorry，是AIDL：]

## 二、定义

---

AIDL是IPC的一个轻量级实现，用了对于Java开发者来说很熟悉的语法。Android也提供了一个工具，可以自动创建Stub（类构架，类骨架）。当我们需要在应用间通信时，我们需要按以下几步走：

1. 定义一个AIDL接口
2. 为远程服务（Service）实现对应Stub
3. 将服务“暴露”给客户程序使用

## 三、用例：HelloSumAIDL

---

AIDL的语法很类似Java的接口（Interface），只需要定义方法的签名。

AIDL支持的数据类型与Java接口支持的数据类型有些不同

1. 所有基础类型（int, char, 等）
2. String, List, Map, CharSequence等类
3. 其他AIDL接口类型
4. 所有Parcelable的类

为了更好地展示AIDL的用法，我们来看一个很简单的例子：两数相加。

### 3.1 创建工程

---

事不宜迟，我们就创建一个Android项目。以下是项目的基本信息（不一定要一样）：

- 项目名称：HelloSumAIDL
- 目标平台：4.3
- 包名：com.android.hellosumaidl
- Activity名称：HelloSumAidlActivity

## 3.2 创建工程

在com.android.hellosumaidl这个包中，新建一个普通文件（New->File），取名为IAdditionService.aidl。在这个文件中输入以下代码：

```
package com.android.hellosumaidl;
// Interface declaration
interface IAdditionService {
    // You can pass the value of in, out or inout
    // The primitive types (int, boolean, etc) are only passed by in
    int add(in int value1, in int value2);
}
```

一旦文件被保存，Android的AIDL工具会在gen/com/android/hellosumaidl这个文件夹里自动生成对应的IAdditionService.java这个文件。因为是自动生成的，所以无需改动。这个文件里就包含了Stub，我们接下来要为我们的远程服务实现这个Stub。

## 3.3 实现远程服务

首先我们在com.android.hellosumaidl这个包中新建一个类，取名叫AdditionService.java。为了实现我们的服务，我们需要让这个类中的onBind方法返回一个IBinder类的对象。这个IBinder类的对象就代表了远程服务的实现。为了实现这个服务，我们要用到自动生成的子类IAdditionService.Stub。在其中，我们也必须实现我们之前在AIDL文件中定义的add()函数。下面是我们远程服务的代码：

```
package com.android.hellosumaidl;
import android.app.Service;
import android.content.Intent;
import android.os.IBinder;
import android.os.RemoteException;
/*
 * This class exposes the service to client
 */
public class AdditionService extends Service {
    @Override
    public void onCreate() {
        super.onCreate();
    }

    @Override
    public IBinder onBind(Intent intent) {
        return new IAdditionService.Stub() {
            /*
             * Implement com.android.hellosumaidl.IAdditionService.add(int, int)
             */
            @Override
            public int add(int value1, int value2) throws
RemoteException {
                return value1 + value2;
            }
        };
    }

    @Override
    public void onDestroy() {
```

```
        super.onDestroy();
    }
}
```

## 3.4 “暴露”服务

一旦实现了服务中的onBind方法，我们就可以把客户程序（在这里是HelloSumAidlActivity.java）与服务连接起来了。为了建立这样的一个链接，我们需要实现ServiceConnection类。我们在HelloSumAidlActivity.java创建一个内部类AdditionServiceConnection，这个类继承ServiceConnection类，并且重写了它的两个方法：onServiceConnected和onServiceDisconnected。下面给出内部类的代码：

```
/*
 * This inner class is used to connect to the service
 */
class AdditionServiceConnection implements ServiceConnection {
    public void onServiceConnected(ComponentName name, IBinder boundService) {
        service = IAdditionService.Stub.asInterface((IBinder)boundService);
        Toast.makeText(HelloSumAidlActivity.this, "Service connected",
            Toast.LENGTH_LONG).show();
    }

    public void onServiceDisconnected(ComponentName name) {
        service = null;
        Toast.makeText(HelloSumAidlActivity.this, "Service disconnected",
            Toast.LENGTH_LONG).show();
    }
}
```

这个方法接收一个远程服务的实现作为参数。这个实现随后被转换（cast）为我们自己的AIDL的实现。我们使用 IAdditionService.Stub.asInterface((IBinder)boundService)。

## 3.5 相关代码

为了完成我们的测试项目，我们需要首先改写main.xml（主界面的格局文件）和string.xml（字符串定义文件）：

**main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:textSize="22sp" />

    <EditText
        android:id="@+id/value1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="@string/hint1" >
    </EditText>
```

```

<TextView
    android:id="@+id/TextView01"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/plus"
    android:textSize="36sp" />

<EditText
    android:id="@+id/value2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="@string/hint2" >
</EditText>

<Button
    android:id="@+id/buttonCalc"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="@string/equal" >
</Button>

<TextView
    android:id="@+id/result"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/result"
    android:textSize="36sp" />

</LinearLayout>

```

## string.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">HelloSumAIDL</string>
    <string name="hello">Hello Sum AIDL</string>
    <string name="result">Result</string>

    <string name="plus">+</string>
    <string name="equal">=</string>

    <string name="hint1">value 1</string>
    <string name="hint2">value 2</string>

</resources>

```

最后，我们的HelloSumAidlActivity.java如下：

```

package com.android.hellosumaidl;
import android.os.Bundle;
import android.os.IBinder;
import android.os.RemoteException;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

```

```

import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.content.ServiceConnection;
public class HelloSumAidlActivity extends Activity {
    IAdditionService service;
    AdditionServiceConnection connection;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        initService();

        Button buttonCalc = (Button)findViewById(R.id.buttonCalc);

        buttonCalc.setOnClickListener(new OnClickListener() {
            TextView result = (TextView)findViewById(R.id.result);
            EditText value1 = (EditText)findViewById(R.id.value1);
            EditText value2 = (EditText)findViewById(R.id.value2);

            @Override
            public void onClick(View v) {
                int v1, v2, res = -1;
                v1 = Integer.parseInt(value1.getText().toString());
                v2 = Integer.parseInt(value2.getText().toString());

                try {
                    res = service.add(v1, v2);
                } catch (RemoteException e) {
                    e.printStackTrace();
                }

                result.setText(Integer.valueOf(res).toString());
            }
        });
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        releaseService();
    }
}

```

/\*

```

    * This inner class is used to connect to the service
    */
    class AdditionServiceConnection implements ServiceConnection {
        public void onServiceConnected(ComponentName name, IBinder boundService)
    {
        service = IAdditionService.Stub.asInterface((IBinder)boundService);
        Toast.makeText(HelloSumAidlActivity.this, "Service connected",
Toast.LENGTH_LONG).show();
        }

        public void onServiceDisconnected(ComponentName name) {
            service = null;
            Toast.makeText(HelloSumAidlActivity.this, "Service disconnected",
Toast.LENGTH_LONG).show();
        }
    }

    /*
    * This function connects the Activity to the service
    */
    private void initService() {
        connection = new AdditionServiceConnection();
        Intent i = new Intent();
        i.setClassName("com.android.hellosumaidl",
com.android.hellosumaidl.AdditionService.class.getName());
        boolean ret = bindService(i, connection, Context.BIND_AUTO_CREATE);
    }

    /*
    * This function disconnects the Activity from the service
    */
    private void releaseService() {
        unbindService(connection);
        connection = null;
    }
}

```