

Android Studio中创建AIDL Service

标签：Android Service AIDL

2015-01-15 12:49 1538人阅读 评论(0) 收藏 举报

分类：

Android (7)

博主允许不得转载。

- 1. 一建立AIDL文件
- 2. 二编写Service子类
- 3. 三客户端调用

版权声明：
本文为博主
原创文章，

用startService，bindService创建的服务只能在本应用程序内访问，如果要使得本程序的服务能够被其他应用程序访问，这时候就要使用远程过程调用(Remote Procedure Call，RPC)方式来实现，安卓定义了一种接口定义语言Android Interface Definition Language，简称AIDL。今天记录一下如何在Android Studio下创建AIDL Service。

建立ADIL Service的步骤比建立普通Service要多一些，主要有：

- 1、创建AIDL文件，在这里面定义远程接口。
- 2、生成Java接口文件。
- 3、建立一个Service的子类，并且记得在AndroidManifest.xml文件中配置。

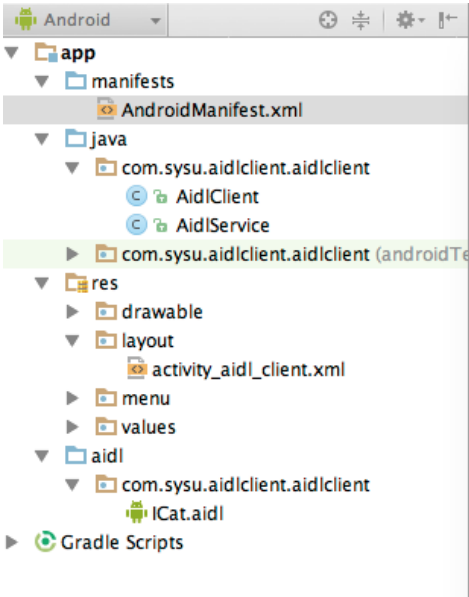
在客户端调用ADIL Servie：

- 1、拷贝服务器端的AIDL文件，并生成Java接口文件。
- 2、用BindService来调用Service,与调用普通Service相类似，只是获取IBinder的方式有点不一样。

下面结合例子来看看，编程环境是Android Studio v1.0.1，本例子基于《疯狂Android讲义(第二版)》10.2节的例子。本AIDL Service只有两个方法，返回一个字符串和double型数据。

一、建立AIDL文件

在项目名称上右键>NEW>AIDL>AIDL File，这样就创建了一个ADIL文件，命名为ICat



文件内容为：

```
[java]
01. // ICat.aidl
02. package com.sysu.aidlclient.aidlclient;
03.
04. interface ICat {
05.     String getColor();
06.     double getWeight();
```

```
07. | }
```

然后菜单中选择Build>Rebulid Project，这样就生成了java接口文件，地址在项目文件夹/app/build/generated/aidl里面。

二、编写Service子类：

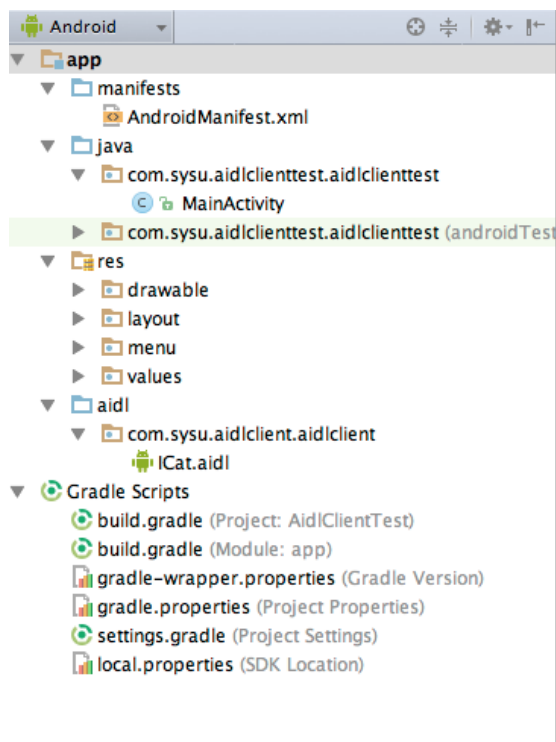
这一步跟建立普通的Service步骤一样，同样右键>New>Service，这样会创建一个Service的子类以及在AndroidManiFest中添加Service的内容，主要要添加action标签并且定义android:name。它的onBind方法返回的IBinder对象要是ICat.Stub的子类的实例，ICat.Stub这个类基本不用去管它，是自动生成的。

```
[java]
01. public class AidlService extends Service {
02.
03.     private CatBinder catBinder;
04.     //此处要继承Stub,实现ICat和IBinder接口
05.     <span style="color:#ff0000;">public class CatBinder extends ICat.Stub
06.     {
07.         @Override
08.         public String getColor() throws RemoteException {
09.             return "get from remote service";
10.         }
11.
12.         @Override
13.         public double getWeight() throws RemoteException {
14.             return 999.9;
15.         }
16.     }</span>
17.
18.     @Override
19.     public void onCreate() {
20.         super.onCreate();
21.         <span style="color:#ff0000;">catBinder = new CatBinder();</span>
22.
23.         @Override
24.         public IBinder onBind(Intent intent) {
25.             <span style="color:#ff0000;">return catBinder;</span>
26.         }
27.
28.         @Override
29.         public void onDestroy() {
30.             System.out.println("remote service destroy");
31.         }
32.     }
```

然后将应用部署到手机上。

三、客户端调用

首先要拷贝AIDL文件，这里要保证文件的内容一模一样，包括包的名称，比如本例子中服务器端AIDL文件所在包的名称是com.sysu.aidlclient.aidlcilent，如何做到这一点，先新建一个项目，然后在：项目文件夹/app/src/main目录下建立一个aidl文件夹，与java文件夹同级，在Android Studio中就可以看到这个目录，在这个目录上右键New>Package，建立一个com.sysu.aidlclient.aidlclient的包，再将aidl文件拷进去。这样才能保证生成的java接口文件完全一样，否则会提示找不到接口。



在MainActivity中调用，用bindService方法，记得要定义一个显式的intent，如红色代码获取Service的IBinder的代理

```
[java]
01. private ServiceConnection conn = new ServiceConnection() {
02.     @Override
03.     public void onServiceConnected(ComponentName name, IBinder service) {
04.         <span style="color:#ff0000;">catService = ICat.Stub.asInterface(service);</span>
05.     }
06.
07.     @Override
08.     public void onServiceDisconnected(ComponentName name) {
09.         catService = null;
10.     }
11. }
12. };
```

```
[java]
01. @Override
02. protected void onCreate(Bundle savedInstanceState) {
03.     super.onCreate(savedInstanceState);
04.     setContentView(R.layout.activity_main);
05.
06.     Intent intent = new Intent();
07.     intent.setAction("com.sysu.aidlclient.action.AIDL_SERVICE");
08.     //this is important
09.     intent.setPackage("com.sysu.aidlclient.aidlclient");
10.     bindService(intent, conn, Service.BIND_AUTO_CREATE);
11. }
```

之后就可以用catService来调用远程service中的方法了

```
[javascript]
01. public void getRemoteService(View view)
02. {
03.     try
04.     {
05.         Toast.makeText(this, "the color and weight is: " + catService.getColor() + ":",
06.             Toast.LENGTH_LONG).show();
07.         System.out.println(catService.getColor() + "----" + catService.getWeight());
08.     } catch (RemoteException e)
```

```
09.         {  
10.             e.printStackTrace();  
11.         }  
12.         System.out.println("button click");  
13.     }
```

这里用到的AIDL只涉及到一些基本的数据类型，还可以用到一些自定义数据类型，这样步骤会稍微多一些，另开一篇文章记录。