启动server有两种方式

1）**StartService()启动Service**  
2）**BindService()启动Service**

1. startServer启动

//TestService1.java

public class TestService1 extends Service {

//必须要实现的方法

@Override

public IBinder onBind(Intent intent) {

return null;

}

//Service被创建时调用

@Override

public void onCreate() {

super.onCreate();

}

//Service被启动时调用

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

return super.onStartCommand(intent, flags, startId);

}

//Service被关闭之前回调

@Override

public void onDestroy() {

super.onDestroy();

}

}

//------------------------------

//----------------------------------

//AndroidManifest.xml完成Service注册 部分

<!-- 配置Service组件,同时配置一个action -->

<service android:name=".TestService1">

<intent-filter>

<action android:name="com.jay.example.service.TEST\_SERVICE1"/>

</intent-filter>

</service>

//-----------------------------------

//-------------------------

//MainActivity

public class MainActivity extends Activity {

private Button start;

private Button stop;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

start = (Button) findViewById(R.id.btnstart);

stop = (Button) findViewById(R.id.btnstop);

//创建启动Service的Intent,以及Intent属性

final Intent intent = new Intent();

intent.setAction("com.jay.example.service.TEST\_SERVICE1");

//为两个按钮设置点击事件,分别是启动与停止service

start.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

startService(intent);

}

});

stop.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

stopService(intent);

}

});

}

}

//-----------------------------

2 BindServer启动

ServiceConnection对象:监听访问者与Service间的连接情况,如果成功连接,回调 onServiceConnected(),

如果异常终止或者其他原因终止导致Service与访问者断开 连接则回调onServiceDisconnected方法,调用unBindService()不会调用该方法!

onServiceConnected方法中有一个IBinder对象,该对象即可实现与被绑定Service 之间的通信!我们再开发Service类时,默认需要实现IBinder onBind()方法,该方法返回的 IBinder对象会传到ServiceConnection对象中的onServiceConnected的参数,我们就可以 在这里通过这个IBinder与Service进行通信!

//-------------------------------------

//TestService2.java:

public class TestService2 extends Service {

private final String TAG = "TestService2";

private int count;

private boolean quit;

//定义onBinder方法所返回的对象

private MyBinder binder = new MyBinder();

public class MyBinder extends Binder

{

public int getCount()

{

return count;

}

}

//必须实现的方法,绑定改Service时回调该方法

@Override

public IBinder onBind(Intent intent) {

Log.i(TAG, "onBind方法被调用!");

return binder;

}

//Service被创建时回调

@Override

public void onCreate() {

super.onCreate();

Log.i(TAG, "onCreate方法被调用!");

//创建一个线程动态地修改count的值

new Thread()

{

public void run()

{

while(!quit)

{

try

{

Thread.sleep(1000);

}catch(InterruptedException e){e.printStackTrace();}

count++;

}

};

}.start();

}

//Service断开连接时回调

@Override

public boolean onUnbind(Intent intent) {

Log.i(TAG, "onUnbind方法被调用!");

return true;

}

//Service被关闭前回调

@Override

public void onDestroy() {

super.onDestroy();

this.quit = true;

Log.i(TAG, "onDestroyed方法被调用!");

}

@Override

public void onRebind(Intent intent) {

Log.i(TAG, "onRebind方法被调用!");

super.onRebind(intent);

}

}

//----------------------------------------

//----------------------------------------

//AndroidManifest.xml

<service android:name=".TestService2" android:exported="false">

<intent-filter>

<action android:name="com.jay.example.service.TEST\_SERVICE2"/>

</intent-filter>

</service>

//-----------------------------------------------------

//----------------------------

//MainActivity.java:

public class MainActivity extends Activity {

private Button btnbind;

private Button btncancel;

private Button btnstatus;

//保持所启动的Service的IBinder对象,同时定义一个ServiceConnection对象

TestService2.MyBinder binder;

private ServiceConnection conn = new ServiceConnection() {

//Activity与Service断开连接时回调该方法

@Override

public void onServiceDisconnected(ComponentName name) {

System.out.println("------Service DisConnected-------");

}

//Activity与Service连接成功时回调该方法

@Override

public void onServiceConnected(ComponentName name, IBinder service) {

System.out.println("------Service Connected-------");

binder = (TestService2.MyBinder) service;

}

};

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

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

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

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

final Intent intent = new Intent();

intent.setAction("com.jay.example.service.TEST\_SERVICE2");

btnbind.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

//绑定service

bindService(intent, conn, Service.BIND\_AUTO\_CREATE);

}

});

btncancel.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

//解除service绑定

unbindService(conn);

}

});

btnstatus.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

Toast.makeText(getApplicationContext(), "Service的count的值为:"

+ binder.getCount(), Toast.LENGTH\_SHORT).show();

}

});

}

}

//-------------------------------------