



常州大学  
CHANGZHOU UNIVERSITY

# 实 验 报 告

## 第三次实验

课程名称： 移动终端软件开发

实验名称： Service 的使用

学 院： 信息数理学院 班 级： 计算机 162

学 号： 16430121 姓 名： 林锦雄

完成日期： 2019 年 5 月 8 日

指导教师： 高晋树 成 绩：



## 一、实验目的

1. 了解 Service 的原理和用途;
2. 掌握服务的使用方法。

## 二、实验过程

### 1、利用服务实现计算 BMI 指数

首先在自己的 java 文件夹里新建一个 service，在里面写上 onBind、onUnbind、onCreate、onStartCommand、onDestory（已经有的不用）等方法。

在界面上添加一个 Button 控件，命名为“计算”，效果如图 1 所示。



图 1 “计算”控件

为“计算”控件添加点击事件，当界面上的 editText 姓名、身高和体重输入之后，点击计算，则会调用服务来计算，并用 Toast 给出提示信息。首先定义一个 intent 与 service 绑定，用来传值。然后为“计算”按钮设置 setOnClickListener 点击事件，首先用 intent.putExtra 方法将三个值传到服务，服务用 getStringExtra 接收，然后根据传过来的值写出具体的判断 bmi 指数的语句，并用 Toast 显示，之后用 bindService 启动服务。

因为 bindService 里面需要一个 connection 参数，所以需要定义一个 ServiceConnection。在这里可以写一个延时方法，里面写上 unbindService 用来解绑服务。因为用 bind 启动服务的话必须得解绑才可以再次绑定，这样可以实现点击多次计算得到正确的返回值。

```
1. 延时代码：
2. TimerTask task = new TimerTask() {
3.     @Override
4.     public void run() {
5.         unbindService(connection); //解除绑定
6.     }
7. };
8. Timer timer = new Timer();
9. timer.schedule(task, 2000); //2 秒后执行 TimerTask 的 run 方法
```

如果输入正确，则显示计算结果，效果如图 2 所示。



图 2 显示提示信息

1. 关键代码:

```
2. public IBinder onBind(Intent intent) {
3.
4.     String name = intent.getStringExtra("name");
5.     String height = intent.getStringExtra("height");
6.     String weight = intent.getStringExtra("weight");
7.     if(height.isEmpty()||weight.isEmpty()){
8.         Toast.makeText(this, "身高或体重为空!!!", Toast.LENGTH_LONG).show();
9.     }
10.    if(!height.isEmpty()&&!weight.isEmpty()) {
11.        height1 = Double.parseDouble(height);
12.        weight1 = Double.parseDouble(weight);
13.        bmi = weight1/((height1/100)*(height1/100));
14.        if(bmi<18.5) {
15.            Toast.makeText(this, name + "体型偏瘦"+"\n"+"bmi 为:
"+bmi, Toast.LENGTH_LONG).show();
16.            bmistring = "体型偏瘦";
17.        }
18.        else if(bmi>=18.5&&bmi<24) {
19.            Toast.makeText(this, name + "体重正常"+"\n"+"bmi 为:
"+bmi, Toast.LENGTH_LONG).show();
20.            bmistring = "体重正常";
21.        }
22.        else if(bmi>=24&&bmi<28) {
```

```
23.         Toast.makeText(this, name + "体型偏胖"+"\n"+"bmi 为:
"+bmi, Toast.LENGTH_LONG).show();
24.         bmistring = "体型偏胖";
25.     }
26.     else if(bmi>=28) {
27.         Toast.makeText(this, name + "体型肥胖"+"\n"+"bmi 为:
"+bmi, Toast.LENGTH_LONG).show();
28.         bmistring = "体型肥胖";
29.     }
30. }
31. return binder;
32. }
```

在 secondActivity 的信息显示界面需要把体质信息显示上去。这就需把 service 计算的值传到主 Activity 中，再从主 Activity 传到 secondActivity。通过 Binder 传值，在 service 写一个类用来返回 bmi，然后在 onBind 中返回那个 binder，之后再在 connection 中的 onServiceConnected 方法中调用 service 中的类，从而得到 bmistring，然后再通过 putString 和 getStringExtra 传到信息显示界面即可。

```
1. 控件关键代码:
2. final Intent intent3 = new Intent("shiyang3.service"); //服务
3. intent3.setPackage("com.example");
4. button3.setOnClickListener(new View.OnClickListener() {
5.     @Override
6.     public void onClick(View v) {
7.         intent3.putExtra("name",editText.getText().toString());
8.         intent3.putExtra("height",editText2.getText().toString());
9.         intent3.putExtra("weight",editText3.getText().toString());
10.        bindService(intent3,connection,BIND_AUTO_CREATE);
11.    }
12. });
```

点击查看，效果如图 3 所示。



图 3 信息显示

1. 关键代码:

```
2. private MyService.BmiBinder binder;
3. private ServiceConnection connection = new ServiceConnection() {
4.     @Override
5.     public void onServiceConnected(ComponentName name, IBinder service) {
6.         //得到服务的返回 binder
7.         binder = (MyService.BmiBinder)service;
8.         bmi = binder.getbmi();
9.         //bmilist.add(bmi);
10.    }
11.    @Override
12.    public void onServiceDisconnected(ComponentName name) {
13.    }
14. };
15.
16. BmiBinder binder = new BmiBinder();
17. class BmiBinder extends Binder { //服务中定义类
18.     String getbmi(){
19.         return bmistring;
20.     }
21. }
```

### 三、实验总结

通过本次实验，首先我对 Service 服务有了更好地运用，利用 onBind 可以实现 Activity 和服务之间的通信，并且绑定之后需要解绑才可以再次绑定，而 onStart 比较方便，但不利于传值处理。并且 Activity 和服务之间的通信用 intent 也是十分方便的。

本次实验，我反思了一下，总结了一些不足之处，比如传值处理还可以用列表，存的更好，传值也还可以用其他的方法。

### 四、思考与练习

如何在程序关闭的时候同时停止服务并不会自动重启？

可以用 HandlerThread，将它的 Looper 给 handler 就可以，具体用法如下：

```
1. HandlerThread handlerThread=new HandlerThread("threadTag");  
2. handlerThread.start();  
3. Handler handler=new Handler(handlerThread.getLooper());  
4. handler.post(test);
```

这样 test 不会阻塞 UI 线程，在 activity 的 onDestroy() 方法中 handler.removeCallbacks(test) 。