

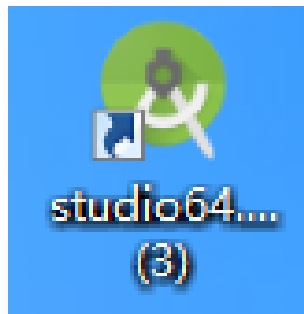


电子信息工程学院

移动终端开发技术

# 服务服务的基本用法

讲 师：陈媛媛



1

多线程编程

2

服务的基本用法

 返回目录

## 7.2 服务的基本用法

7.2.1

● 定义一个服务

7.2.2

● 启动和停止服务

7.2.3

● 活动和服务进行通信

# 定义一个服务

在一个项目中新增一个名为 `MyService` 的类，并让它继承自 `Service`。

```
public class MyService extends Service {  
    @Override  
    public IBinder onBind(Intent intent) {  
        return null;  
    }  
}
```

应该在服务中去处理一些事情

# 定义一个服务

// onCreate()方法会在服务创建的时候调用。

```
public void onCreate() {  
    super.onCreate();  
}
```

//onStartCommand()方法会在每次服务启动的时候调用

```
public int onStartCommand(Intent intent, int flags, int startId) {  
    return super.onStartCommand(intent, flags, startId);  
}
```

//onDestroy()方法会在服务销毁的时候调用，去回收那些不再使用的资源。

```
public void onDestroy() {  
    super.onDestroy();  
}
```

重写 Service 中方法

# 启动和停止服务

主要是借助 Intent 来实现的，调用 `startService()` 方法来启动服务，调用 `stopService()` 方法来停止服务。



Tag	Text
MyService	onCreate executed
MyService	onStartCommand executed

Tag	Text
MyService	onDestroy executed

# 活动和服务进行通信

希望在 MyService 里提供一个下载功能，然后在活动中可以决定何时开始下载，以及随时查看下载进度。实现这个功能的思路是创建一个专门的 Binder 对象来对下载功能进行管理。重写 onBind() 方法。

```
Intent bindIntent = new Intent(this, MyService.class);  
bindService(bindIntent, connection, BIND_AUTO_CREATE); // 绑定服务  
unbindService(connection); // 解绑服务
```





1

多线程编程

2

服务的基本用法

3

服务的生命周期



# 服务的生命周期

调用Context的startService()方法，相应的服务就会启动起来，并回调 onStartCommand()方法。如果这个服务之前还没有创建过， onCreate()方法会先于 onStartCommand()方法执行。服务启动了之后会一直保持运行状态，直到 stopService()或 stopSelf()方法被调用。这时服务中的 onDestroy()方法就会执行，表示服务已经销毁了。

还可以调用 Context 的 bindService()来获取一个服务的持久连接，这时就会回调服务中的 onBind()方法。类似地，如果这个服务之前还没有创建过， onCreate()方法会先于 onBind()方法执行。当调用 unbindService()方法时， onDestroy()方法也会执行。

对一个服务既调用了 startService()方法，又调用了 bindService()方法的情况下要同时调用 stopService()和 unbindService()方法， onDestroy()方法才会执行。



1

定义一个服务

---

2

启动和停止服务

---

3

服务与活动相互通信

---