

绑定服务

1.1 知识解析

使用 `bindService(Intent service, ServiceConnection conn, int flags)` 来绑定一个服务。其中：

- `Intent`：要绑定的 `service` 的 `intent`
 - `ServiceConnecion`：当绑定服务的时候，需要实现这个类，并实现它的 2 个回调方法：
 - `onServiceConnected(ComponentName name, IBinder service)`：当和服务建立连接的时候会被调用，此时可以得到服务的 `onService()` 中返回的 `Binder` 对象
 - `onServiceDisconnected(ComponentName name)`：当和服务的连接终止的时候会被调用
 - `flag`：指定绑定方式的标记，在 `Context` 中定义了相关常量
- 在 `Service` 中，需要覆盖 `onBind()` 方法：

`IBinder onBind(Intent intent)`：返回一个 `IBinder` 对象，在服务被连接上的时候，这个 `IBinder` 将会传递给 `ServiceConnection` 的 `onServiceConnected()` 方法。接收一个 `intent` 对象，可以接收从绑定服务的时候发送的 `intent` 信息

它有一个子类 `Binder`，可以在 `Service` 中定义一个 `Binder` 的子类，并且在这个 `Binder` 子类中定义一个可以返回这个 `Service` 对象的方法，这样，这个 `Service` 对象就会被传递给 `client` 端的 `ServiceConnection` 对象，然后可以在 `client` 端调用 `Service` 的方法了。

`unBindService(ServiceConnection conn)`：在 `Context` 中解除绑定的时候，调用这个方法解除绑定。

`onUnbind(Intent intent)`：在 `Service` 中，回调这个方法。可以在这个方法里清理一些资源。

1.3 实战操作

```
MyBinder binder;
```

```
@Override
```

```
public IBinder onBind(Intent intent) {  
    // TODO Auto-generated method stub  
    binder = new MyBinder();  
    return binder;  
}
```

```
public class MyBinder extends Binder{  
    public MyService getService(){  
        return MyService.this;  
    }  
}
```

```
MyService.MyBinder binder;
```

```
ServiceConnection conn = new ServiceConnection(){
```

```
@Override
```

```
public void onServiceConnected(ComponentName name, IBinder service) {  
    // TODO Auto-generated method stub  
    binder = (MyService.MyBinder)service;  
    ser = binder.getService();  
}
```

```
@Override
```

```
public void onServiceDisconnected(ComponentName name) {  
    // TODO Auto-generated method stub  
    binder = null;  
    ser = null;  
}
```

```
};
```

1.4 职业素质

android中服务是运行在后台的东西，级别与activity差不多。既然说service是运行在后台的服务，那么它就是不可见的，没有界面的东西。你可以启动一个服务Service来播放音乐，或者记录你地理信息位置的改变，或者启动一个服务来运行并一直监听某种动作。

Service 和其他组件一样，都是运行在主线程中，因此不能用它来做耗时的请求或者动作。你可以在服务中开一个线程，在线程中做耗时动作。