

安卓 Lab3 实验报告

1. Service 生命周期和流程图

onCreate: 服务被创建时调用。

onBind: 绑定服务时调用，返回一个 `IBinder` 对象供客户端调用。

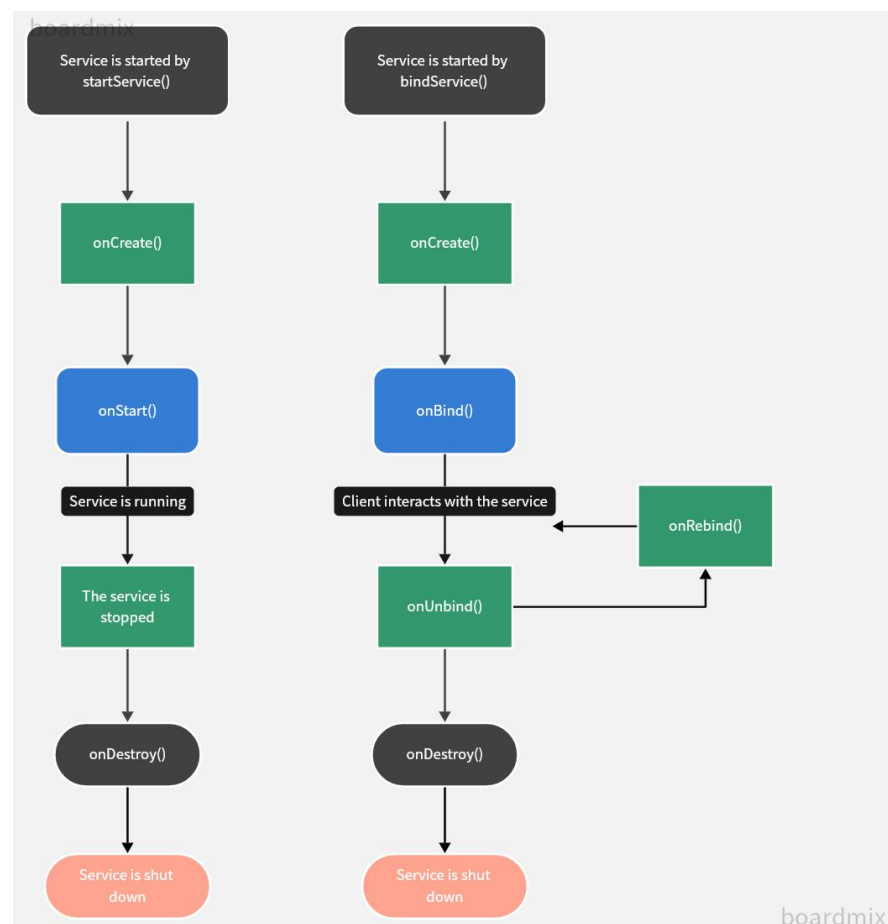
onUnbind: 当另一个组件通过调用 `unbindService()` 与服务解绑时，系统将调用此方法。

onRebind: 当旧的组件与服务解绑后，另一个新的组件与服务绑定，

`onUnbind()` 返回 `true` 时，系统将调用此方法。

onStartCommand: 启动服务时调用，用于处理启动命令。

onDestroy: 服务被销毁时调用。



2.观察子线程 ID:

```
// 获取子线程ID
long threadId = Thread.currentThread().getId();
// 在日志中输出子线程ID
Log.d( tag: "MusicPlayerServices", msg: "Thread ID: " + threadId);
```

2031-2100	MusicPlayerServices	com.example.service	D	Thread ID: 101
2031-2100	MusicPlayerServices	com.example.service	D	Thread ID: 101
2031-2100	MusicPlayerServices	com.example.service	D	Thread ID: 101
2031-2100	MusicPlayerServices	com.example.service	D	Thread ID: 101
2031-2100	MusicPlayerServices	com.example.service	D	Thread ID: 101
2031-2100	MusicPlayerServices	com.example.service	D	Thread ID: 101
2031-2100	MusicPlayerServices	com.example.service	D	Thread ID: 101
2031-2100	MusicPlayerServices	com.example.service	D	Thread ID: 101

3.启动和绑定服务

```
Intent intent = new Intent(getApplicationContext(),MusicPlayerServices.class);
bindService(intent, conn, BIND_AUTO_CREATE);
//startService(intent);
```

相同点:

- 1.二者均是由 Service 类扩展而来的, 可以通过扩展 Service 类开实现自己的服务。
- 2.二者都可以通过 Intent 传输数据

不同点:

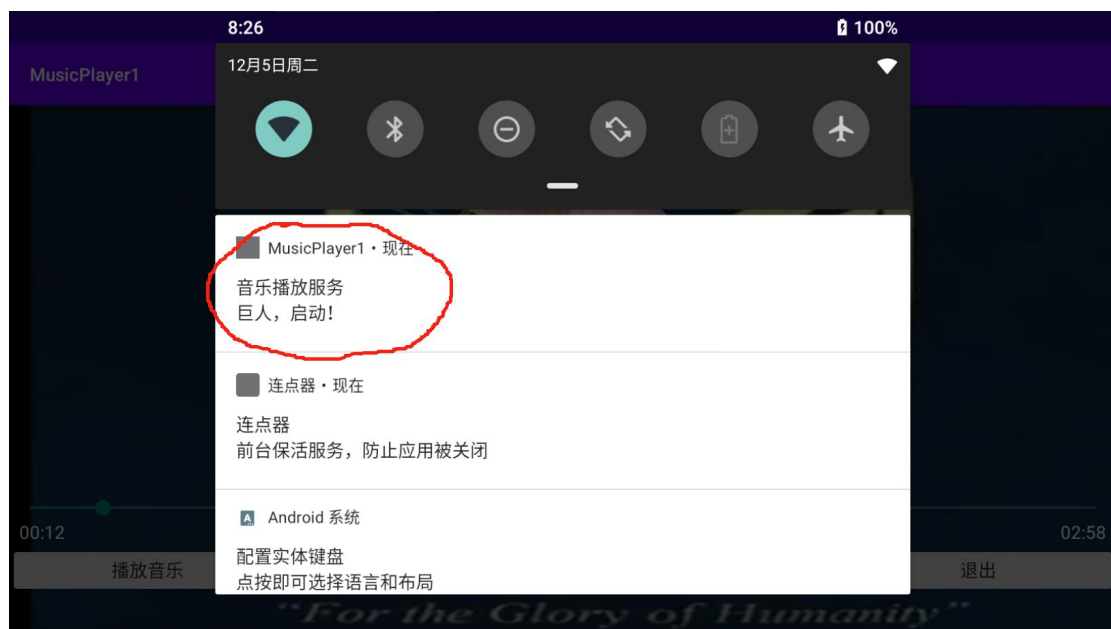
- 1.启动服务独立于调用者生命周期, 而绑定服务与调用者生命周期绑定。
- 2.方法实现不同, 启动服务: startService(Intent intent), 通过 Intent 启动服务。而绑定服务: bindService(Intent intent, ServiceConnection conn, int flags), 是通过 ServiceConnection 实现绑定。

3.启动服务通过 **Intent** 进行通信，无法直接调用服务中的方法，而绑定服务通过 **IBinder** 进行通信，可以直接调用服务中的方法。

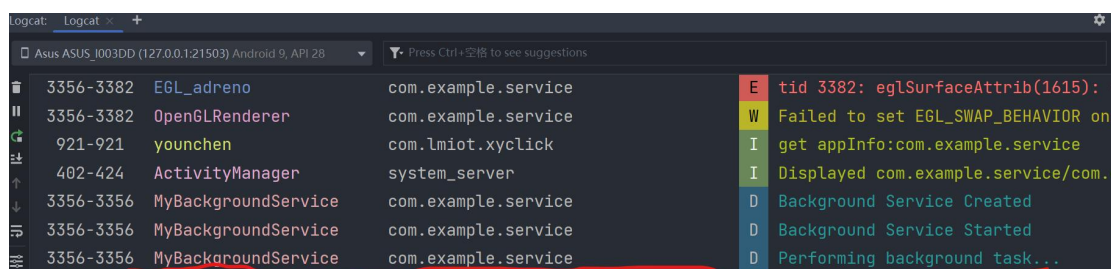
4.启动服务适用于一次性任务，而绑定服务适用于长时间的任务，需可以实现双向通信。

4.前台服务和后台服务

本 Lab 实现的前台服务：在通知栏显示通知、播放音乐。



后台服务：在 **Log** 中输出“后台服务被创建、开始以及执行”



前台服务：

- 1.通知用户可见： 前台服务通常会在任务栏展现一个通知，通知用户有一个正在运行的服务。
- 2.长时间运行： 前台服务通常用于需要长时间运行的任务，例如音乐播放、位置跟踪等。用户通过通知可以知道服务的存在。
- 3.与用户交互： 用户可以通过通知与服务进行互动，例如通过通知栏上的按钮控制音乐播放。

后台服务：

- 1.轻量级任务： 后台服务适用于执行一些轻量级、不占用过多系统资源且不需要用户直接感知的任务，例如数据同步、数据处理、将用户数据上传至云端等。
- 2.定时任务： 后台服务可以在固定的时间间隔执行任务，例如定时从网络下载数据。

5.Service 和 activity 的双向通信

在本 Lab 录制的 demo 演示视频中体现了双向通信，当用户点击对应的播放、暂停、退出等按钮时候，屏幕都会显示一条信息展现当前音乐的状态。

