将Android进行到底之服务（service）

时光剑客 LV.3
2023年01月03日 08:24 · 阅读 2127

关注

前言

我们都知道，字节最近发布了PICO 4 VR眼镜，我买了一个，体验还行。因为我也是做VR眼镜的Android应用层开发的，所以想把自己项目中遇到的一些Android技术分享给读者。近些年随着VR眼镜的兴起，Android的的服务（Service）和广播（Broadcast）以及内容提供者（Content Provider）越来越被重用，相反Activity这个曾经很吃香的组件在VR眼镜的开发过程中却用的不多。本节我会介绍Android的服务在VR眼镜中的使用，服务（Service）是Android的四大组件之一，很多Android的萌新在开发的时候肯定知道这个组件但是用得不是很多，毕竟刚入行更多的时候是写界面。很少会有机会去接触服务这种没有用户界面的组件。所以，让我们一起看看服务的使用吧。

服务（Service）是一种可以在后台长时间执行而没有用户界面的组件。需要注意的是由于它运行在UI线程，因此不能做耗时操作，否则可能会引起ANR（Application Not Response: 应用无响应）。

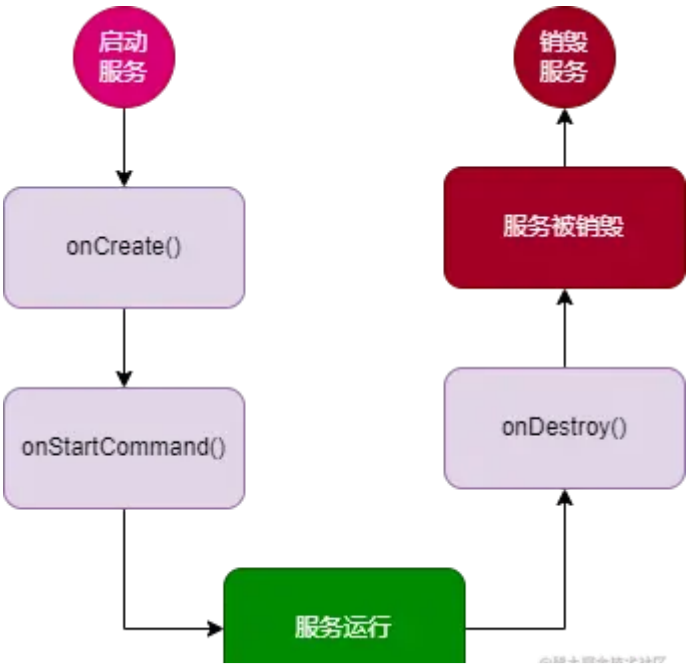
注意：服务在其托管进程的主线程中运行，它既不创建自己的线程，也不在单独的进程中运行（除非另行指定）。如果服务将执行任何 CPU 密集型工作或阻止性操作（例如 MP3 播放或联网），则应通过在服务内创建新线程来完成这项工作。通过使用单独的线程，您可以降低发生“应用无响应”(ANR) 错误的风险，而应用的主线程仍可继续专注于运行用户与 Activity 之间的交互。

> 简单说就是服务运行在主线程，不要在服务中做耗时操作，否则可能会引起ANR

二、服务（Service）的生命周期

服务的生命周期有两种形式，。因为服务可以和Activity绑定，也可以不绑定。当我们的activity需要和服务通信的时候，是需要把服务和Activity进行绑定的，因此服务的生命周期分为未绑定Activity的和绑定Activity的。我们可以用以下两张图来描述这两种绑定方式的生命周期：

（1）未绑定Activity的服务生命周期图



> onCreate()方法，它会在onStartCommand()或者的onBind()方法之前被系统调用，作用是做一些初始化相关的操作。如果服务已经在运行，则onCreate()方法不会被调用
> **(2)onStartCommand()** 当一个组件（如Activity）通过startService()请求启动服务时，系统会调用onStartCommand()方法，这个方法一旦执行，就会在后台无限期执行。如果实现了这个方法，那么就需要我们调用stopSelf()或者时stopService()来停止服务。

java 复制代码

```java
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    return START_STICKY;
}
```

> 我们可以看到这个方法有个返回值，这个返回值代表什么意思呢？我们一起来看看。这个返回值是指用于描述系统应如何在系统终止服务的情况下继续运行服务。从 onStartCommand() 返回的值必须是以下常量之一：

START_NOT_STICKY

如果系统在 onStartCommand() 返回后终止服务，则除非有待传递的挂起 Intent，否则系统不会重建服务。这是最安全的选项，可以避免在不必要时以及应用能够轻松重启所有未完成的作业时运行服务。

START_STICKY

如果系统在 onStartCommand() 返回后终止服务，则其会重建服务并调用 onStartCommand()，但不会重新传递最后一个 Intent。相反，除非有挂起 Intent 要启动服务，否则系统会调用包含空 Intent 的 onStartCommand()。在此情况下，系统会传递这些 Intent。此常量适用于不执行命令、但无限期运行并等待作业的媒体播放器（或类似服务）。

START_REDELIVER_INTENT

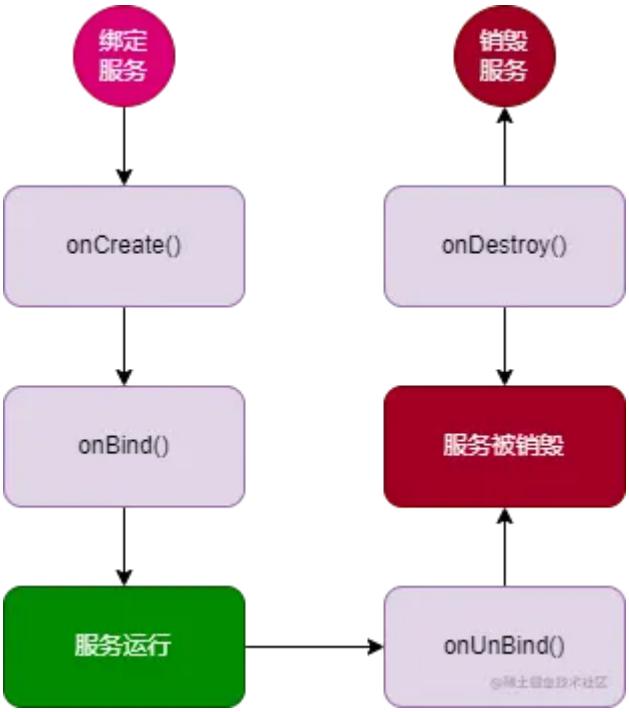
如果系统在 onStartCommand() 返回后终止服务，则其会重建服务，并通过传递给服务的最后一个 Intent 调用 onStartCommand()。所有挂起 Intent 均依次传递。此常量适用于主动执行

START_STICKY这个值，它的意思是启动了服务后，当服务被杀死的时候，系统会尽最大了的努力重启服务。其他的值读者可以去看API说明，这里不一一列举了。

（3）onDestroy() 当某个操作导致服务停止，比如执行了stopService(),那么服务接下来会执行onDestroy()方法。服务一般应在这个方法中执行资源的释放操作

注意：用户可以查看其设备上正在运行的服务。如果他们发现自己无法识别或信任的服务，则可以停止该服务。为避免用户意外停止您的服务，您需要在应用清单的 `<service>` 元素中添加 `android:description`。请在描述中用一个短句解释服务的作用及其提供的好处。

（2）绑定Activity的服务生命周期图



上图中的onCreate()和onDestroy方法都和前面讲的一样，这里我们只讲onBind()和onUnbind()方法

（1）onBind()

当一个组件想通过调用bindService()与服务绑定，比如我们使用AIDL做进程间通信时，系统会调用这个方法，在这个方法的实现中，我们必须通过返回IBinder提供一个接口，来供客户端和服务器端用来与服务进行通信。

三、服务的相关操作

（1）启动服务

我们可以通过将 Intent 传递给 startService() 或 startForegroundService()，从 Activity 或其他应用组件启动服务。Android 系统会调用服务的 onStartCommand() 方法，并向其传递 Intent，从而指定要启动的服务

注意：如果您的应用面向 API 级别 26 或更高版本，除非应用本身在前台运行，否则系统不会对使用或创建后台服务施加限制。如果应用需要创建前台服务，则其应调用 startForegroundService()。此方法会创建后台服务，但它会向系统发出信号，表明服务会将自行提升至前台。创建服务后，该服务必须在五秒内调用自己的 startForeground() 方法。

（2）停止服务

启动服务必须管理自己的生命周期。换言之，除非必须回收内存资源，否则系统不会停止或销毁服务，并且服务在 onStartCommand() 返回后仍会继续运行。服务必须通过调用 stopSelf() 自行停止运行，或由另一个组件通过调用 stopService() 来停止它。

一旦请求使用 stopSelf() 或 stopService() 来停止服务，系统便会尽快销毁服务。如果服务同时处理多个对 onStartCommand() 的请求，则不应在处理完一个启动请求之后停止服务，因为可能已收到新的启动请求（**在第一个请求结束时停止服务会终止第二个请求**）。为避免此问题，我们可以使用 stopSelf(int) 确保服务停止请求始终基于最近的启动请求。

换言之，在调用 stopSelf(int) 时，我们需传递与停止请求 ID 相对应的启动请求 ID（传递给 onStartCommand() 的 startId）。此外，如果服务在您能够调用 stopSelf(int) 之前收到新启动请求，则 ID 不匹配，服务也不会停止。

注意：为避免浪费系统资源和消耗电池电量，请确保应用在工作完成之后停止其服务。如有必要，其他组件可通过调用 stopService() 来停止服务。即使为服务启用绑定，如果服务收到对 onStartCommand() 的

> 绑定服务允许应用组件通过调用 bindService() 与其绑定，从而创建长期连接。此服务通常不允许组件通过调用 startService() 来启动它。

如需与 Activity 和其他应用组件中的服务进行交互，或需要通过进程间通信 (IPC) 向其他应用公开某些应用功能，则应创建绑定服务。

如要创建绑定服务，需通过实现 onBind() 回调方法返回 IBinder，从而定义与服务进行通信的接口。然后，其他应用组件可通过调用 bindService() 来检索该接口，并开始调用与服务相关的方法。服务只用于与其绑定的应用组件，因此若没有组件与该服务绑定，则系统会销毁该服务。我们不必像通过 onStartCommand() 启动的服务那样，以相同方式停止绑定服务。

如要创建绑定服务，我们必须定义指定客户端如何与服务进行通信的接口。服务与客户端之间的这个接口必须是 IBinder 的实现，并且服务必须从 onBind() 回调方法返回该接口。收到 IBinder 后，客户端便可开始通过该接口与服务进行交互。

多个客户端可以同时绑定到服务。完成与服务的交互后，客户端会通过调用 unbindService() 来取消绑定。如果没有绑定到服务的客户端，则系统会销毁该服务。

（4）在前台运行服务

> 前台服务是用户主动意识到的一种服务，因此在内存不足时，系统也不会考虑将其终止。前台服务必须为状态栏提供通知，将其放在运行中的标题下方。这意味着除非将服务停止或从前台移除，否则不能清除该通知。

注意：如果应用面向 Android 9（API 级别 28）或更高版本并使用前台服务，则其必须请求 FOREGROUND_SERVICE 权限。这是一种普通权限，因此，系统会自动为请求权限的应用授予此权限。如果面向 API 级别 28 或更高版本的应用试图创建前台服务但未请求 FOREGROUND_SERVICE，则系统会抛出 SecurityException。

> 如要请求让服务在前台运行，请调用 startForeground()。此方法采用两个参数：唯一标识通知的整型数和用于状态栏的 Notification。此通知必须拥有 PRIORITY_LOW 或更高的优先级。下面是官网示例：

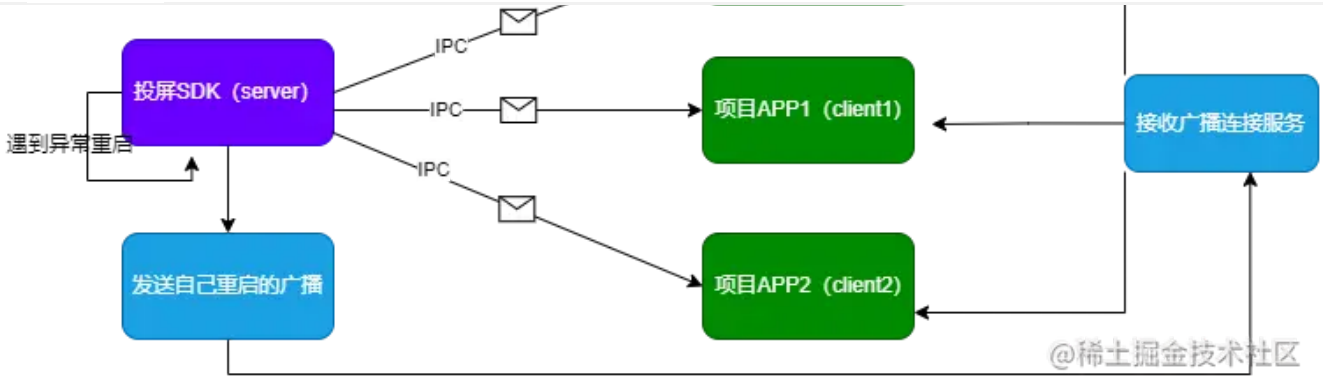
```
            PendingIntent.getActivity(this, 0, notificationIntent, 0)
        }

    val notification: Notification = Notification.Builder(this, CHANNEL_DEFAULT_IMPORTANCE)
            .setContentTitle(getText(R.string.notification_title))
            .setContentText(getText(R.string.notification_message))
            .setSmallIcon(R.drawable.icon)
            .setContentIntent(pendingIntent)
            .setTicker(getText(R.string.ticker_text))
            .build()

    startForeground(ONGOING_NOTIFICATION_ID, notification)
```

注意：提供给 startForeground() 的整型 ID 不得为 0。

如要从前台移除服务，则调用 stopForeground()。此方法采用布尔值，指示是否需同时移除状态栏通知。此方法不会停止服务。但是，如果您在服务仍运行于前台时将其停止，则通知也会随之移除。

（5）服务的容错处理

这里的容错处理主要是用在我们使用服务来做一些IPC通信的时候，比如有这样一个例子，我们需要将一个投屏的SDK接入到项目中，这时我们可以将投屏的SDK做二次封装，然后再给项目用，但是我们又怕这个投屏的SDK写的BUG影响我们的APP，所以为了防止投屏SDK崩溃而导致我们的应用崩溃，我们选择将投屏的SDK再开一个进程去实现，也就是将投屏的SDK做成一个服务端，我们的项目做客户端。客户端和服务端可以通过IPC来通信。这样投屏SDK就算崩溃也是在它自己的进程崩溃，不会影响我们的APP进程：



当我们将投屏SDK放到服务端的时候，如果服务端因为异常重启了，我们需要做一些容错处理让客户端可以不受服务端异常的影响继续使用投屏服务。一般情况下，假如我们客户端因为异常重启了，是可以继续正常使用服务端的投屏服务的，因为我们的服务端一直都活着。但是假如服务端因为异常重启了，我们的客户端是不知道的，因为C-S架构就是客户端请求，服务端响应。

> **这种情况下我们的容错方案就是：当服务端发生异常重启的时候，通过广播的方式，像所有连接它的客户端发出自己重启的广播，让所有的客户端重新连接一下它，然后就可以正常使用投屏服务了。是不是很简单，原理如上图所示。**

总结

以上就是今天要讲的内容，本文介绍了服务的生命周期，使用方法以及注意事项，并且结合实际的应用场景解析服务。包括服务的创建和停止，容错处理。由于篇幅的原因，本文的介绍无法详尽所有有关的知识，感兴趣的伙伴可以移步官网。欢迎评论区一起讨论，一起进步。

分类： Android 标签： Android Kotlin

文章被收录于专栏：

将Android进行到底

用于分享Android的基础知识，帮助自己稳固基础，帮助萌新入门...

订阅专栏

首页 ▾ 探 🔍

相关小册

VIP Flutter 完全手册

小德_Kurt LV.4 4904购买

¥14.95 ~~¥29.9~~ 首单券后价

VIP Android 进阶：基于 Kotlin 的 Android App 开发实践

Tony沈哲 LV.5 1305购买

¥9.95 ~~¥19.9~~ 首单券后价

评论

看完啦， 登录 分享一下感受吧～

相关推荐

1月前 Android Kotlin

android 微信抢红包工具 AccessibilityService

1147 40 2

2月前 Android Kotlin

Android App封装 ——架构（MVI + kotlin + Flow）

1.3w 161 37

1年前 Android Kotlin

手动埋点转无痕埋点，如何做到代码“零”入侵

1.5w 113 37

1年前 前端 算法

15 评论 收藏

5年前 Android

好用漂亮的Android 表格框架

5.6w 798 103

3年前 面试 前端

【1 月最新】前端 100 问：能搞懂 80% 的请把简历给我

62.8w 10933 376

1年前 Android

Android Service(三) 保证Service不被后台销毁

2889 11 6

8月前 Android Kotlin

黑科技！让Native Crash 与ANR无处发泄！

1.4w 167 41

4年前 React.js Android Kotlin

学 Flutter，能挽救Android 开发吗?

1.4w 33 151

1年前 Android Kotlin

Android正确的保活方案，不要掉进保活需求死循环陷进

1.7w 221 45

3月前 Android 源码 架构

【Android R】车载 Android 核心服务 - CarPropertyService

3091 17 3

2年前 Kotlin Android

Android 面试黑洞——当我按下 Home 键再切回来，会发生什么?

1.3w 189 18

1年前 Android Android Jetpack Kotlin

1年前 后端 Java

Service层的接口是不是多此一举?

1.9w 55 44

1年前 Android

Android Service(四) Service使用实例

1723 9 5

5月前 Android Kotlin

落地 Kotlin 代码规范，DeteKt 了解一下~

9191 142 16

1年前 HarmonyOS Android

鸿蒙Harmony谈了这么久，和Android到底啥区别?

3.4w 54 22

1年前 前端 面试

后端一次给你10万条数据,如何优雅展示，面试官到底考察我什么?

19.9w 552 339

1年前 前端 Vue.js 面试

后端一次给你10万条数据，如何优雅展示，到底考察我什么?

12.4w 1525 256

4月前 Android 源码 架构

【Android R】车载 Android 核心服务 - CarService 解析

4052 34 5

敖丙 1年前 后端 面试 Java

为什么我的Service无法注入进来?

3.5w 66 3

彭旭锐 11月前 Android Android Jetpack Kotlin

15 评论 收藏

程序员江同学 1年前 Kotlin Android

【带着问题学】协程到底是怎么切换线程的?

7973 79 9

AndroidLMY 1年前 Kotlin Android

Android悬浮窗看这篇就够了

2.2w 105 17

随心____ 5年前 Retrofit Kotlin RxJava

Kotlin结合Rxjava和Retrofit做到极简网络请求

8555 131 10

码个蛋 5年前 Android RxJava Kotlin

花了 4 个月整理了 50 篇 Android 干货文章

4.2w 2589 47

玉刚说 4年前 JavaScript Android 服务器

Android Webview H5 秒开方案实现

4.2w 755 33

JowayYoung 3年前 JavaScript ECMAScript 6

1.5万字概括ES6全部特性(已更新ES2020)

23.1w 5932 235

业凉凉 1年前 Android Kotlin Android Jetpack

学不动也要学，Jetpack Compose 写一个 IM APP（一）

1.3w 152 47

KunMinX 6月前 Android Android Jetpack Kotlin

Android：解决 MVI 架构实战痛点

9966 63 28

富途web开发团队 3年前 前端

扔物线 8月前 Android Kotlin Android Jetpack

【面试黑洞】Android 的键值对存储有没有最优解?

6905 138 17

程序员江同学 1年前 Kotlin Android

【带着问题学】协程到底是什么?

1.2w 178 37

技术小黑屋 5年前 Android Kotlin

为什么我要改用 Kotlin

2.5w 246 17

大胃粥 2年前 Android

Android O对后台Service限制

4605 17 评论

fundroid 9月前 Android Android Jetpack Kotlin

Compose 动画边学边做 - 夏日彩虹

9194 70 12

唐子玄 2年前 Kotlin Android

换一个思路，超简单的RecyclerView预加载

1.3w 93 17

张晋涛 1年前 Service Mesh Kubernetes 云原生

倍受关注的 Cilium Service Mesh 到底怎么玩？ - 上手实践

1.6w 13 1

菜卷子 1年前 Kotlin Android

写给Android开发者的协程基本原理

8143 96 13

Season3266 1年前 Android Kotlin

15 评论 收藏

前端论道 3年前 前端 PWA

Service Worker 从入门到出门

1.5w 379 25

wow_PingCode 4年前 PWA

Service Worker

3118 39 3

寻找海蓝96 3年前 面试 前端

面试官到底想看什么样的简历?

7.4w 1703 123

tangxd3 1年前 前端 Node.js

Egg 服务 Service

701 8 7

Neal_yang 3年前 面试 前端

一个合格(优秀)的前端都应该阅读这些文章

37.0w 9091 398

拉丁吴 4年前 Google Android Java

Android 目前最稳定和高效的UI适配方案

4.9w 1074 161

wanbo 3年前 Android Kotlin

Android 中不应该使用 Enum 吗?

1.1w 91 27

Ruheng 5年前 Android Kotlin

使用 Circular Reveal 动画让页面跳转更炫酷

8508 281 11

唐子玄 2年前 性能优化 Android Kotlin

小码哥哥 3年前 Android

Android组件化神器 —— ServicePool

5466 33 36

亦枫 5年前 Android Kotlin

当 Kotlin 遇上 Android KTX，岂止丝滑?

8861 79 1

geniusmart 5年前 Android Kotlin

Kotlin 之美—DSL篇

1.2w 80 5

已禁用 5年前 Kotlin React.js iOS

重磅消息！Kotlin要支持iOS开发和Web开发了！

1.3w 240 27

黄燕斌 5年前 安卓

好用漂亮的Android 表格框架2

5.2w 168 52

安迪詹妮弗 3年前 科特林 安卓

Androidx 下 Fragment 懒加载的新实现

2.0w 247 37

郭东东 4年前 面试

中高级前端大厂面试秘籍，为你保驾护航金三银四，直通大厂(上)

64.7w 7048 495

JD-CP 4年前 安卓 科特林 最有价值球员

Kotlin + MVP + Flutter，让你可以在自己的项目中集成 Flutter 并使用

8265 190 16