目录

[一、开发环境说明 2](#_Toc795600045)

[二、API说明 2](#_Toc788388990)

[1、摄像头服务开启接口 2](#_Toc622628364)

[2、摄像头服务关闭接口 3](#_Toc1419405652)

[3、人脸处理监听器设置接口 3](#_Toc2064548266)

[4、人脸注册任务开启接口 3](#_Toc1538032415)

[5、人脸注册回调接口 3](#_Toc937353297)

[6、人脸识别任务开启接口 4](#_Toc813532282)

[7、人脸识别回调接口 4](#_Toc940584433)

[8、人脸处理任务关闭接口 4](#_Toc1509829781)

[9、人脸删除接口 4](#_Toc1763760809)

[10、手势处理监听器设置接口 4](#_Toc909227382)

[11、手势识别任务开启接口 5](#_Toc60860388)

[12、手势识别回调接口 5](#_Toc862692137)

[13、手势识别关闭接口 5](#_Toc1687110483)

[三、Use Case处理流程 6](#_Toc1977739070)

[四、导入AAR包 7](#_Toc429404553)

[五、创建人脸手势处理服务 7](#_Toc1428463400)

[六、检测硬件使用权限及请求相应权限 10](#_Toc85421372)

[七、启动服务 11](#_Toc1796856255)

[八、接口说明 14](#_Toc2053009147)

# 一、开发环境说明

开发环境：Android Studio

Android SDK版本：

minSdkVersion 21

compileSdkVersion 23

targetSdkVersion 23

# 二、API说明

## 1、摄像头服务开启接口

功能：开启摄像头功能

/\*

\*@ context: 描述程序的上下文

\*/

static boolean startCameraService(Context context);

## 2、摄像头服务关闭接口

功能：关闭摄像头功能

/\*

\*@ context: 描述程序的上下文

\*/

static boolean stopCameraService(Context context);

## 3、人脸处理监听器设置接口

功能：设置人脸处理回调接口的对象

/\*

\*@ listener: 设置接收回调接口数据的对象

\*/

setOnFaceProcessAvailableListener(OnFaceProcessAvailableListener listener);

## 4、人脸注册任务开启接口

功能：启动人脸注册任务

boolean mm\_start\_face\_register();

## 5、人脸注册回调接口

功能：接收人脸注册结果

Status取值定义:

interface Face\_Status {

int FACE\_NORMAL = 0; /\*表示注册成功\*/

int FACE\_NO\_FACE = 1; /\*表示没检测到人脸\*/

}

/\*

\*@ status: 描述注册信息，比如是否注册成功，或者是否检测到人脸,用来提示用户使人脸在UI圆圈中合适的位置;

\*@ img: 需要显示在注册的圆圈中的图片

\*@ age: 检测出来人的年龄

\*@ gender: 检测出来人的性别

\*@ id: 成功注册后获得的唯一id

\*/

void Face\_Register\_CB\_Func(int status, Bitmap img, int age, int gender, int id);

## 6、人脸识别任务开启接口

功能：启动人脸识别任务

boolean mm\_start\_face\_recognition();

## 7、人脸识别回调接口

功能：接收人脸识别结果

/\*

\*@ img: 识别出来时人的图片

\*@ age: 识别出来人的年龄

\*@ gender: 识别出来人的性别

\*@ id: 识别出注册时的唯一id,如果为0xffffffff表示没有注册

\*/

void Face\_Recognition\_CB\_Func(Bitmap img, int age, int gender, int id);

## 8、人脸处理任务关闭接口

功能：关闭人脸处理任务,清空监听器

boolean mm\_stop\_face\_process();

## 9、人脸删除接口

功能：用来删除某个注册的人脸

/\*

\*@ id: 要删除的人脸id

\*/

boolean mm\_faceid\_delete(int id);

## 10、手势处理监听器设置接口

功能：设置手势处理回调接口的对象

/\*

\*@ listener: 设置接收回调接口数据的对象

\*/

setOnGestureProcessAvailableListener(OnGestureProcessAvailableListener listener);

## 11、手势识别任务开启接口

功能：启动手势识别任务

boolean mm\_start\_gesture\_recognition();

## 12、手势识别回调接口

功能：接收手势识别结果

手势id取值定义:

interface Gesture\_Pose{

GESTURE\_POSE\_0=0, /\*表示手势0\*/

GESTURE\_POSE\_1=1, /\*表示手势1\*/

GESTURE\_POSE\_2=2, /\*表示手势2\*/

GESTURE\_POSE\_3=3, /\*表示手势3\*/

GESTURE\_POSE\_4=4, /\*表示手势4\*/

GESTURE\_POSE\_MAC=0xff, /\*表示没识别\*/

}

/\*

\*@ id: 识别出来手的pose

\*/

void Gesture\_Recognition\_CB\_Func(int id);

## 13、手势识别关闭接口

功能：关闭手势识别

boolean mm\_stop\_gesture\_recognition();

# 三、Use Case处理流程

User

SDK

开启摄像头

开启摄像头服务

startCameraService()

创建服务

创建人脸处理对象

设置监听器

开启人脸注册

开启人脸注册任务

mm\_start\_face\_register()

处理人脸注册结果

返回人脸注册结果

Face\_Register\_CB\_Func()

返回状态

返回注册结果

关闭人脸注册

关闭人脸注册任务

mm\_stop\_face\_process()

返回状态

设置监听器

开启人脸识别

开启人脸识别任务

mm\_start\_face\_recognition()

返回人脸识别结果

Face\_Recognition\_CB\_Func()

返回识别结果

处理人脸识别结果

关闭人脸识别

关闭人脸识别任务

mm\_stop\_face\_process()

返回状态

创建手势识别对象

设置监听器

开启手势识别

开启手势识别任务

mm\_start\_gesture\_recognition()

返回手势识别结果

Gesture\_Recognition\_CB\_Func()

返回识别结果

处理手势识别结果

删除人脸

删除人脸

mm\_faceid\_delete()

返回状态

关闭手势识别

关闭手势识别任务

mm\_stop\_gesture\_recognition()

返回状态

关闭摄像头

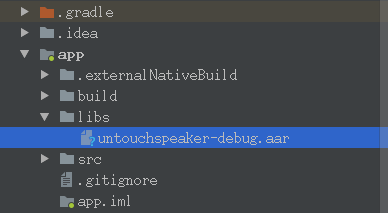
关闭摄像头服务

stopCameraService()

返回状态

# 四、导入AAR包

1. 将AAR包复制到app的libs目录下，如下图：



1. 配置build.gradle文件：

在app下的build.gradle中加入

repositories{

flatDir{

dirs ‘libs’

}

}

在dependencies中加入

compile(name:’untouchspeaker-debug’,ext:’aar’)

Name指定的是要导入的aar包名称；ext指定的是要导入的包后缀

1. 同步gradle即可。
2. 如果调试需要更换AAR包，需要先把libs目录下的AAR包删除，然后屏蔽

repositories{

flatDir{

dirs ‘libs’

}

}

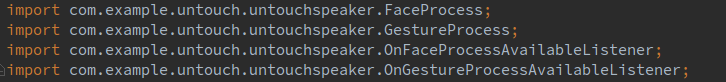
和compile(name:’untouchspeaker-debug’,ext:’aar’)，点击同步gradle，然后再从步骤1开始导入新的AAR包即可。

# 创建人脸手势处理服务

1. 创建untouchService类，继承自父类Service，重载onCreate，onStartCommand，onDestroy和onBind方法；
2. 在Android Manifest.xml文件中的<application>标签中声明服务：



1. 导入人脸处理包和手势处理包：



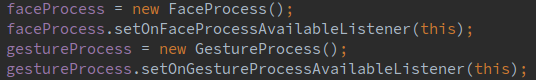
1. 继承回调接口的监听器接口：



1. 定义人脸处理对象和手势处理对象：



1. 在onCreate方法中设置监听器：



1. 重载回调接口：
2. 人脸注册回调接口：



1. 人脸识别回调接口：



1. 手势识别回调接口：



1. 调用相应的方法：
2. 在onStartCommand方法中调用人脸注册启动方法：



1. 在人脸注册回调接口中判断注册成功，调用人脸处理关闭方法关闭人脸注册任务：

aar44

设置人脸处理监听器：

aar45

调用人脸识别启动方法：



1. 在人脸识别函数中判断id号为5时，调用人脸处理关闭方法：



调用人脸删除方法：



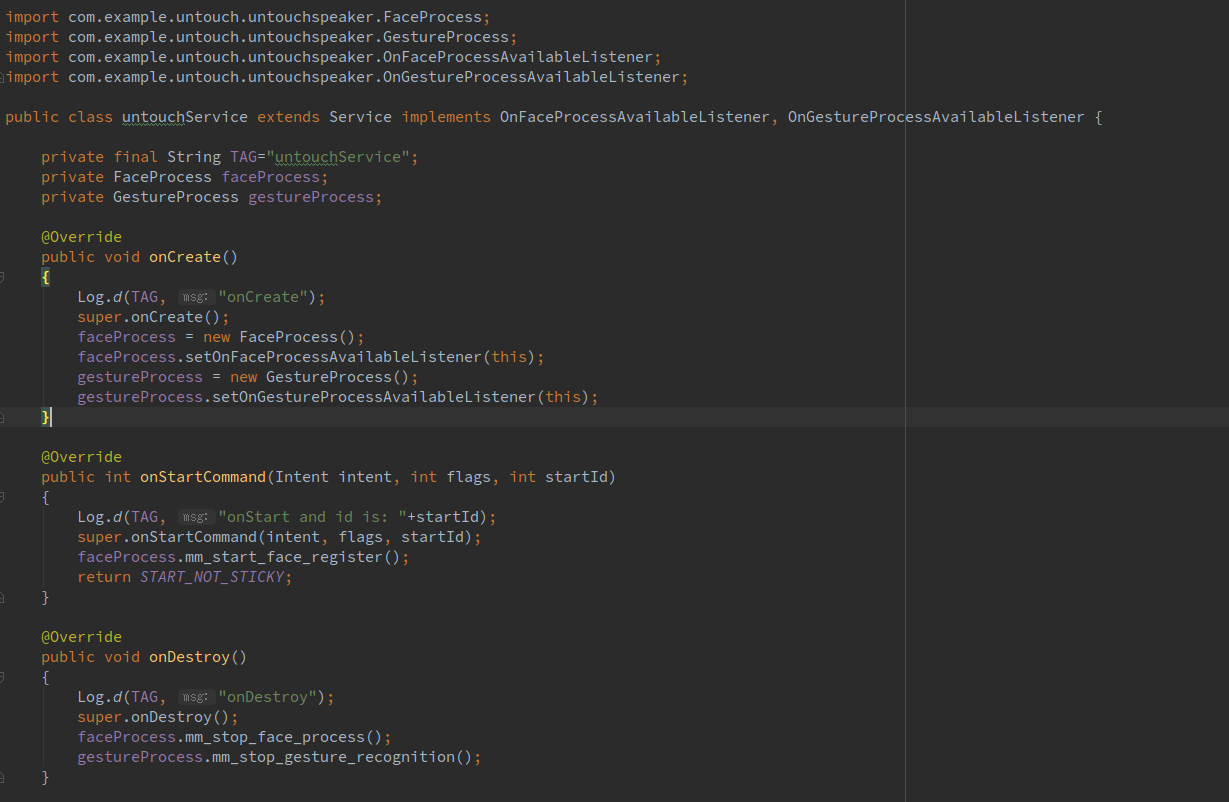
调用手势识别启动方法：

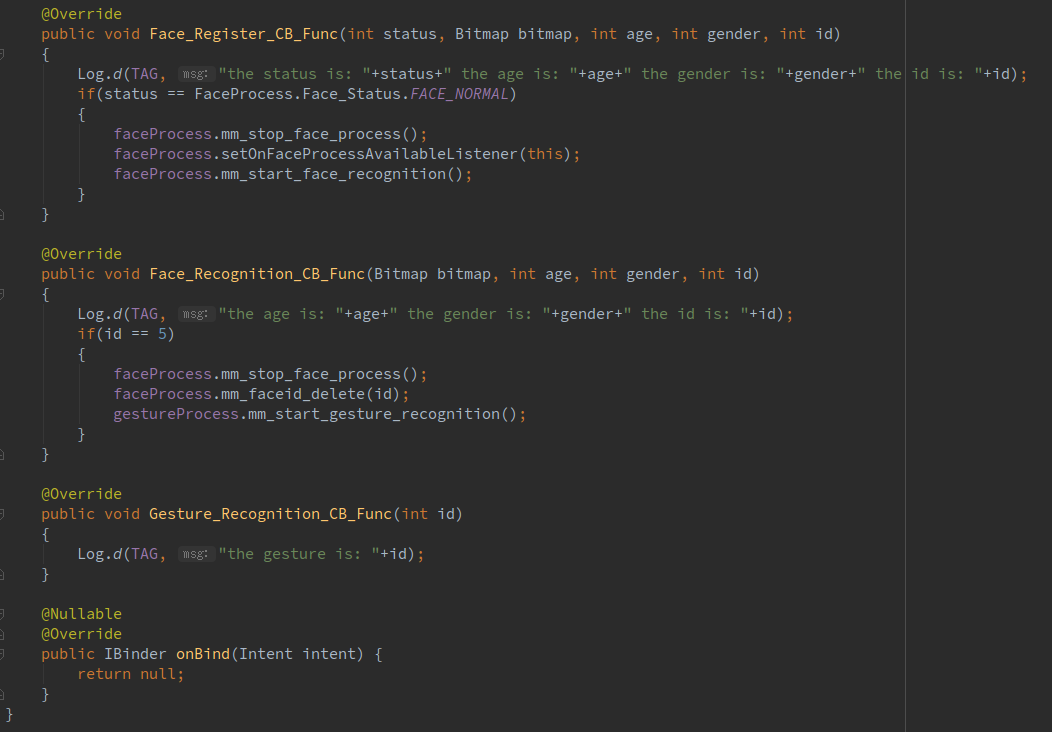


1. 在服务停止方法onDestroy中调用人脸处理和手势识别关闭方法：



1. 人脸手势处理服务完成方法说明：



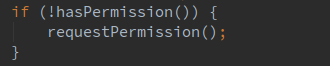


# 六、检测硬件使用权限及请求相应权限

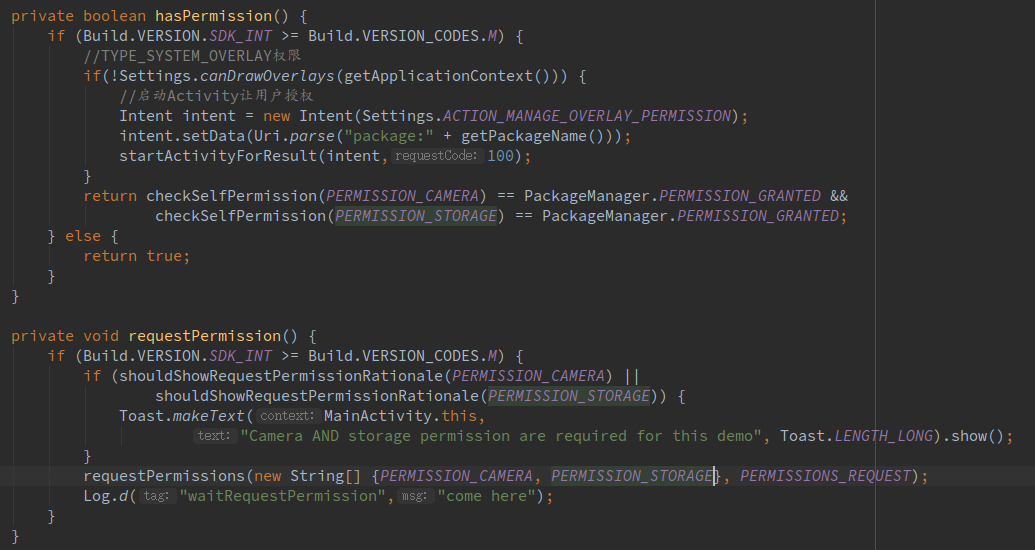
1. 定义变量：



1. 在MainActivity的onCreate方法中检测相应权限，如果没有则进行请求：



1. 检测权限和请求权限方法：

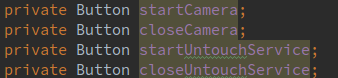


# 七、启动服务

1. 导入摄像头服务包：



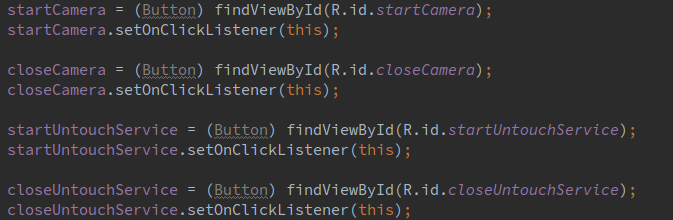
1. 定义四个按钮，分别对应启动摄像头服务，关闭摄像头服务，启动人脸手势处理服务，关闭人脸手势处理服务：



1. 继承按钮监听器接口：



1. 定义的四个按钮对象和界面上定义的四个按钮关联并设置监听器：



1. 在按钮回调接口onClick中实现启动和关闭相应服务：
2. 摄像头启动服务接口：



1. 摄像头关闭服务接口：



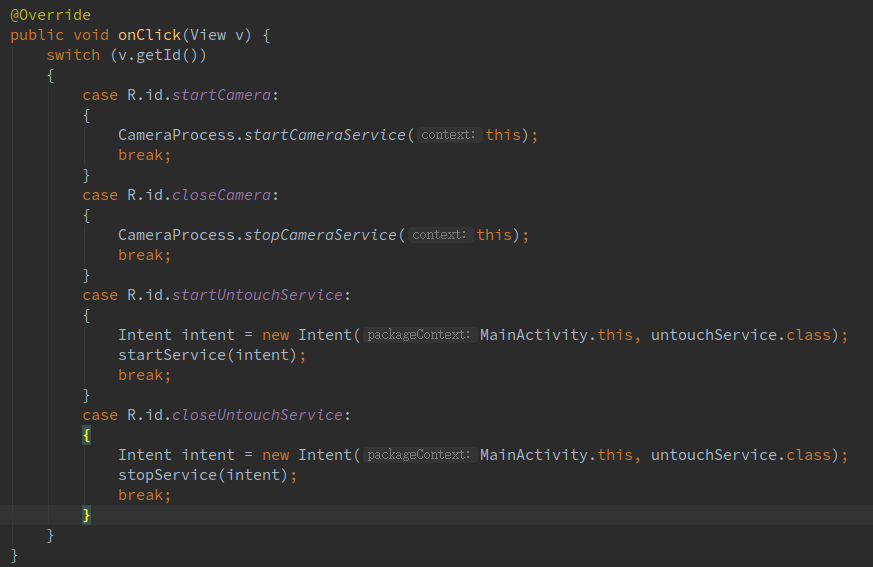
1. 人脸手势处理服务启动接口：

aar46

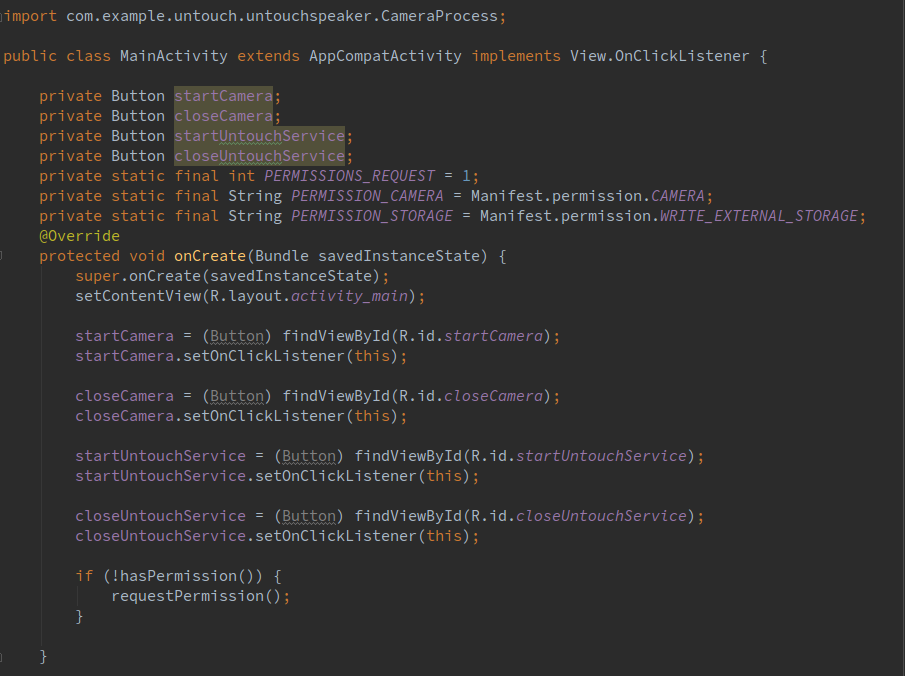
1. 人脸手势处理服务关闭接口：

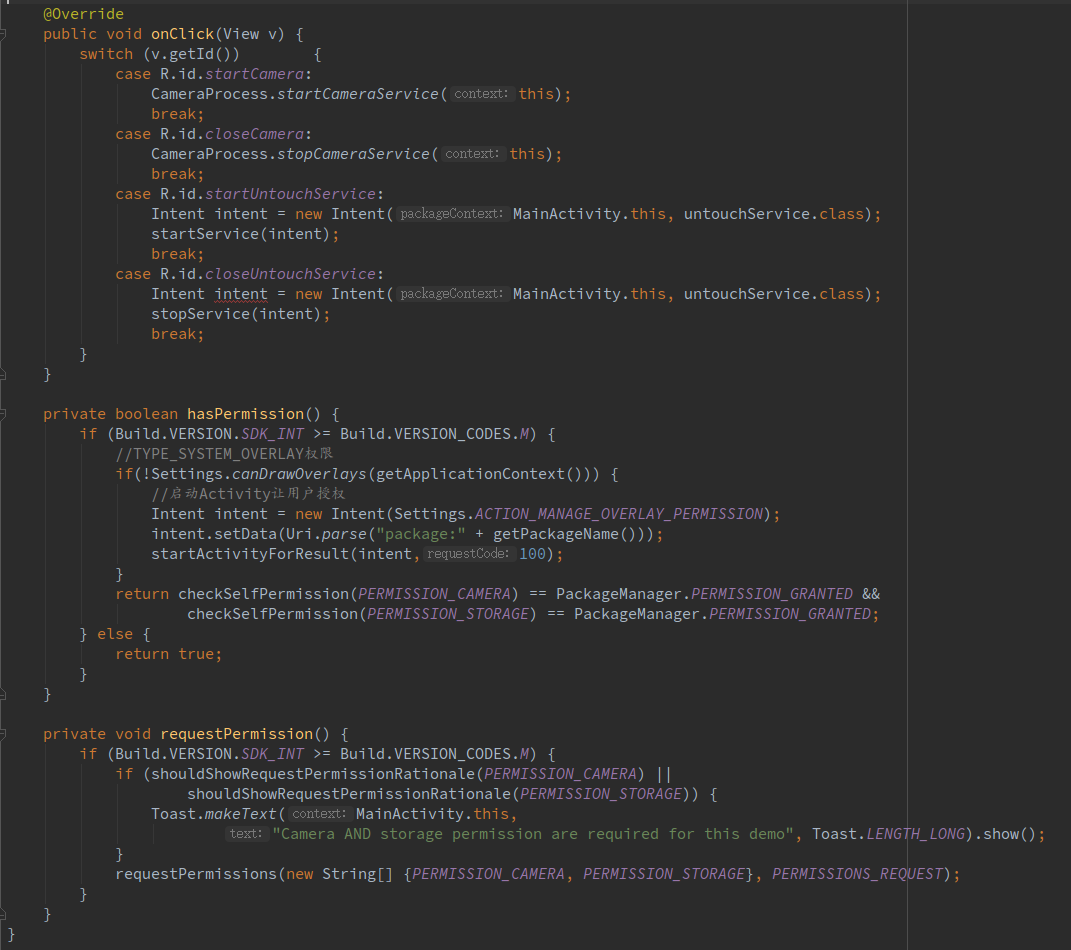
aar47

1. 按钮相应回调接口处理：



1. 完整方法说明：





# 八、接口说明

1. 先调用摄像头启动服务，在调用人脸处理和手势识别服务；
2. 先停止人脸处理和手势识别服务，再调用摄像头停止服务；
3. 调用人脸注册方法后，会发送5次人脸注册失败，然后发送人脸注册成功，图片bitmap一直为空，age、gender、id为从0开始累加的数；
4. 调用人脸识别方法后，发送人脸识别结果，图片bitmap为空，age、gender、id为从0开始累加的数；
5. 调用手势识别方法后，会连续发送手势识别结果，id为从0开始累加的数。
6. 调用人脸处理关闭方法后，会同时关闭人脸注册和人脸识别功能，并清空监听器，所以如果只是关闭一个功能，再开启下一个功能之前需要重新注册监听器。