

Android通知栏前台服务

一、前台服务的简单介绍

前台服务是那些被认为用户知道且在系统内存不足的时候不允许系统杀死的服务。前台服务必须给状态栏提供一个通知，它被放到正在运行(Ongoing)标题之下——这意味着通知只有在这个服务被终止或从前台主动移除通知后才能被解除。

最常见的表现形式就是**音乐播放服务**，应用程序后台运行时，用户可以通过通知栏，知道当前播放内容，并进行暂停、继续、切歌等相关操作。

二、为什么使用前台服务

后台运行的Service系统优先级相对较低，当系统内存不足时，在后台运行的Service就有可能被回收，为了保持后台服务的正常运行及相关操作，可以选择将需要保持运行的Service设置为前台服务，从而使APP长时间处于后台或者关闭（进程未被清理）时，服务能够保持工作。

三、前台服务的详细使用

1. 创建服务内容，如下（四大组件不要忘记清单文件进行注册，否则启动会找不到服务）；

```
public class ForegroundService extends Service {

    private static final String TAG = ForegroundService.class.getSimpleName();

    @Override
    public void onCreate() {
        super.onCreate();
        Log.e(TAG, "onCreate");
    }

    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        Log.e(TAG, "onBind");
        return null;
    }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        Log.e(TAG, "onStartCommand");
        return super.onStartCommand(intent, flags, startId);
    }

    @Override
    public void onDestroy() {
        Log.e(TAG, "onDestroy");
        super.onDestroy();
    }

}
```

2. 创建服务通知内容，例如音乐播放，蓝牙设备正在连接等：

```
/**
 * 创建服务通知
 */
private Notification createForegroundNotification() {
    NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);

    // 唯一的通知通道的id.
    String notificationChannelId = "notification_channel_id_01";

    // Android8.0以上的系统，新建消息通道
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        //用户可见的通道名称
        String channelName = "Foreground Service Notification";
        //通道的重要程度
        int importance = NotificationManager.IMPORTANCE_HIGH;
        NotificationChannel notificationChannel = new NotificationChannel(notificationChannelId,
channelName, importance);
        notificationChannel.setDescription("Channel description");
        //LED灯
        notificationChannel.enableLights(true);
        notificationChannel.setLightColor(Color.RED);
        //震动
        notificationChannel.setVibrationPattern(new long[]{0, 1000, 500, 1000});
        notificationChannel.enableVibration(true);
        if (notificationManager != null) {
            notificationManager.createNotificationChannel(notificationChannel);
        }
    }

    NotificationCompat.Builder builder = new NotificationCompat.Builder(this,
notificationChannelId);
    //通知小图标
    builder.setSmallIcon(R.drawable.ic_launcher);
    //通知标题
    builder.setContentTitle("ContentTitle");
    //通知内容
    builder.setContentText("ContentText");
    //设定通知显示的时间
    builder.setWhen(System.currentTimeMillis());
    //设定启动的内容
    Intent activityIntent = new Intent(this, NotificationActivity.class);
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 1, activityIntent,
PendingIntent.FLAG_UPDATE_CURRENT);
    builder.setContentIntent(pendingIntent);

    //创建通知并返回
    return builder.build();
}
```

3. 启动服务时，创建通知：

```
@Override
public void onCreate() {
    super.onCreate();
    Log.e(TAG, "onCreate");
    // 获取服务通知
    Notification notification = createForegroundNotification();
    //将服务置于启动状态 ,NOTIFICATION_ID指的是创建的通知的ID
    startForeground(NOTIFICATION_ID, notification);
}
```

4. 停止服务时，移除通知：

```
@Override
public void onDestroy() {
    Log.e(TAG, "onDestroy");
    // 标记服务关闭
    ForegroundService.serviceIsLive = false;
    // 移除通知
    stopForeground(true);
    super.onDestroy();
}
```

5. 判断服务是否启动及获取传递信息：

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    Log.e(TAG, "onStartCommand");
    // 标记服务启动
    ForegroundService.serviceIsLive = true;
    // 数据获取
    String data = intent.getStringExtra("Foreground");
    Toast.makeText(this, data, Toast.LENGTH_SHORT).show();
    return super.onStartCommand(intent, flags, startId);
}
```

以上就是前台服务的创建过程，相关注释已经很明白了。

注：一般前台服务会在状态栏显示一个通知，最典型的应用就是音乐播放器，只要在播放状态下，就算休眠也不会被杀，如果不想显示通知，只要把参数里的int设为0即可。

服务创建完毕，接下来就可以进行服务的启动了，启动前不要忘记在清单文件中进行前台服务权限的添加：

```
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
```

6. 服务的启动和停止

```
//启动服务
if (!ForegroundService.serviceIsLive) {
    // Android 8.0使用startForegroundService在前台启动新服务
    mForegroundService = new Intent(this, ForegroundService.class);
    mForegroundService.putExtra("Foreground", "This is a foreground service.");
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        startForegroundService(mForegroundService);
    } else {
        startService(mForegroundService);
    }
}
```

```
    }  
} else {  
    Toast.makeText(this, "前台服务正在运行中...", Toast.LENGTH_SHORT).show();  
}
```

```
//停止服务  
mForegroundService = new Intent(this, ForegroundService.class);  
stopService(mForegroundService);
```

posted @ 2020-03-30 17:31 几圈年轮 阅读(5578) 评论(0) 编辑 收藏 举报