

快来了解下 Android P 兼容与适配

黄俊彬

作者：黄俊彬

链接：

<https://www.jianshu.com/p/9e9e902ea039>

本文由作者授权发布。

P 正式发布啦，适配的工作应该可以了解一波了。

1

P 新特性

Android P 在Google I/O2018正式发版，全新的手势操作选项。底部虚拟键将由小白点和一颗返回键取代。通过轻触回到主页、长按呼出语音助手。

新的特性主要有：

- 利用 Wi-Fi RTT 进行室内定位
- 刘海屏API支持
- 通知栏功能增强
- 多摄像头支持和摄像头更新
- HDR VP9 视频、HEIF 图像压缩和 Media API

详细可参考Google官方文档介绍。

<https://developer.android.google.cn/preview/features>

2

P 兼容优化

根据官方的API迁移指南，对应用比较影响的有如下几点：

non-SDK接口的使用

一般来说，SDK 接口是指在 Android 框架[软件包索引](#)中记录的接口。 对非 SDK 接口的处理是 API 抽象化的实现细节；其会随时更改，恕不另行通。

<https://developer.android.google.cn/reference/packages>

Android P 引入了针对非 SDK 接口的新使用限制，无论是直接使用还是通过反射或 JNI 间接使用。 无论应用是引用非 SDK 接口还是尝试使用反射或 JNI 获取其句柄，均适用这些限制。

名单分类

- Light grey list: targetSDK \geq P时，警告；
- Dark grey list: targetSDK $<$ P时，警告； \geq p时，不允许调用；
- Black list:三方应用不允许调用；

名单查看：

<https://android.googlesource.com/platform/frameworks/base/+master/config/>

具体影响

访问方式	结果
Dalvik 指令引用字段	引发 NoSuchFieldError
Dalvik 指令引用函数	引发 NoSuchMethodError
通过 Class.getDeclaredField() 或 Class.getField() 反射	引发 NoSuchFieldException
通过 Class.getDeclaredMethod() 或 Class.getMethod() 反射	引发 NoSuchMethodException
通过 Class.getDeclaredFields() 或 Class.getFields() 反射	结果中未出现非 SDK 成员
通过 Class.getDeclaredMethods() 或 Class.getMethods() 反射	结果中未出现非 SDK 成员
通过 env->GetFieldID() 调用 JNI	返回 NULL , 引发 NoSuchFieldError
通过 env->GetMethodID() 调用 JNI	返回 NULL , 引发 NoSuchMethodError

挖孔屏适配

谷歌P版本提供了统一的挖孔屏方案和三方适配挖孔屏方案：

- 对于有状态栏的页面，不会受到挖孔屏特性的影响；
- 全屏显示的页面，系统挖孔屏方案会对应用界面做下移避开挖孔区显示；
- 已经适配的P的应用的全屏页面可以通过谷歌提供的适配方案使用挖孔区，真正做到全屏显示

总的来说，就是P版本已经坐了兼容，全屏显示和状态栏显示，都会避开挖空区域显示。

但注意对于沉浸式的显示要注意，避免挖空挡住UI布局，需要做好适配。





示例一





示例二

P版本三方适配挖孔屏方案

1、 新增挖孔屏挖孔尺寸和位置接口

```
class WindowInsets {  
  
    DisplayCutout getDisplayCutout();  
  
}
```

```

class DisplayCutout {

    int getSafeInsetLeft();

    int getSafeInsetTop();

    int getSafeInsetRight();

    int getSafeInsetBottom();

    Region getBounds();

}

```

2、新窗口布局模式，允许应用程序请求是否在挖孔区域布局：

```

class WindowManager.LayoutParams {

    int layoutInDisplayCutoutMode;

    final int LAYOUT_IN_DISPLAY_CUTOUT_MODE_DEFAULT;

    final int LAYOUT_IN_DISPLAY_CUTOUT_MODE_ALWAYS;

    final int LAYOUT_IN_DISPLAY_CUTOUT_MODE_NEVER;

}

```

layoutInDisplayCutoutMode值说明：

a) LAYOUT_IN_DISPLAY_CUTOUT_MODE_DEFAULT：默认情况下，全屏窗口不会使用到挖孔区域，非全屏窗口可正常使用挖孔区域。

b) LAYOUT_IN_DISPLAY_CUTOUT_MODE_ALWAYS：窗口声明使用挖孔区域

c) LAYOUT_IN_DISPLAY_CUTOUT_MODE_NEVER: 窗口声明不使用挖孔区域

Battery Improvements

谷歌在P版本之前没有一个完整的功耗解决方案，OEM厂商分别开发各自的功耗方案，管控手段都包括了清理应用，功耗得到优化，但是同时也影响了三方应用的一些功能正常使用，谷歌为了解决这个问题在P版本提出了自己的功耗解决方案。

主要方案：

AAB (Auto Awesome Battery)：

- 1、通过ML算法将应用进行分类，不同类型的应用功耗管控策略不一样
- 2、 Firebase Cloud Messaging (FCM):管控三方消息接收的频率
- 3、谷歌提供了统一的应用的管控方法：Forced App Standby (FAS)，谷歌不会通过清理应用来优化功耗

Extreme Battery Saver (EBS) 谷歌超级省电模式；

Smart screen brightness: 屏幕亮度调节优化算法。

影响

谷歌功耗方案对三方应用各种管控，存在导致应用后台功能无法正常使用的可能，特别是：IM、邮箱、闹钟、音乐（直播）、地图导航、运动健康、下载、日历等应用影响比较大。目前通过谷歌提供的调试命令验证：所有的应用都有可能被分到管控的类型，对三方的后台功能是有影响的。

不允许共享WebView数据目录

应用程序不能再跨进程共享单个WebView数据目录。如果您的应用有多个使用WebView，CookieManager或android.webkit包中的其他API的进程，则当第二个进程调用WebView方法时，您的应用将崩溃。

该特性只影响已经适配P的应用，也就是targetSDK Version>=P。

移除对 Build.serial 的直接访问

现在，需要 Build.serial 标识符的应用必须请求 READ_PHONE_STATE 权限，然后使用 Android P 中新增的新 Build.getSerial() 函数。

SELinux 禁止访问应用的数据目录

系统强制每个应用的 SELinux 沙盒对每个应用的私有数据目录强制执行逐个应用的 SELinux 限制。现在，不允许直接通过路径访问其他应用的数据目录。应用可以继续使用进程间通信（IPC）机制（包括通过传递 FD）共享数据。

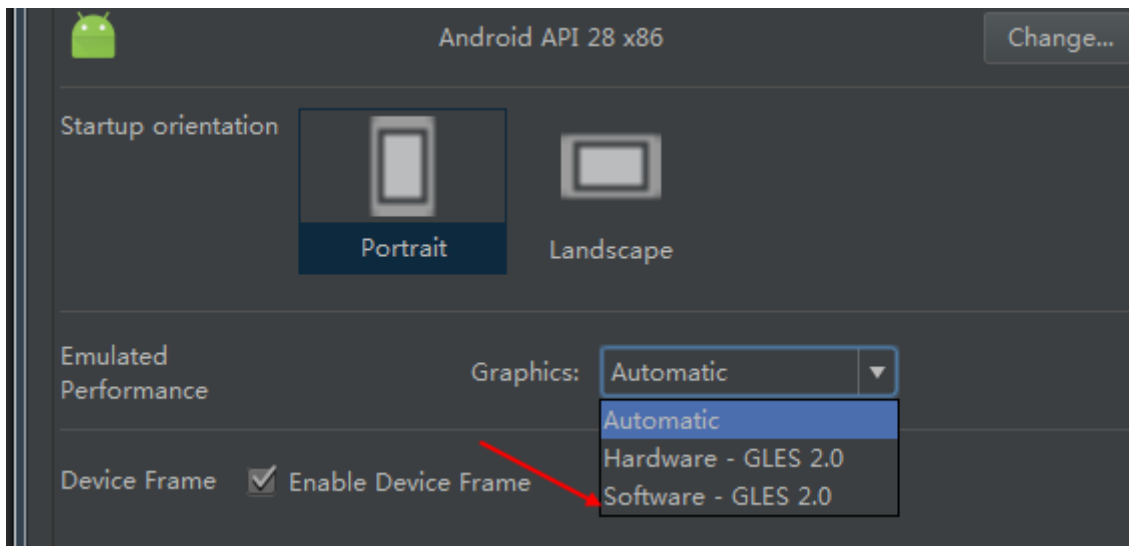
3

项目升级P实操

使用模拟器

- 1、使用Android studio 3.1或者更高的版本
- 2、升级Android API 28版本，SDK Tools升级至Android SDK Build-Tools 28
- 3、下载API 28的虚拟机镜像
- 4、设置启动虚拟机

这里遇到一个坑，虚拟机一直起不来。注意Graphics设置为Software-GLES 2.0



更新项目Gradle配置

```
androidTargetSdkVersion = 28
```

```
androidCompileSdkVersion = 28
```

模拟机开发者选项设置

开发者选项支持设置模拟刘海屏缺口，项目可通过设置，进行应用的测试。





项目升级遇到问题

这里注意我们应用是直接由24升级到28.

```
1、 java.lang.SecurityException: Failed to find provider null for user 0;
expected to find a valid ContentProvider for this authority
```

问题原因：项目使用了ActiveAndroid，在 8.0 或 8.1 系统上使用 26 或以上的版本的 SDK 时，调用 ContentResolver 的 notifyChange 方法通知数据更新，或者调用 ContentResolver 的 registerContentObserver 方法监听数据变化时，会出现上述异常。

解决方案：

在清单文件配置

```
<provider
    android:name="com.activeandroid.content.ContentProvider"
    android:authorities="com.ylmf.androidclient"
    android:enabled="true"
    android:exported="false">

</provider>
```

2、CLEARTEXT communication to life.115.com not permitted by network security policy

问题原因： Android P 限制了明文流量的网络请求，非加密的流量请求都会被系统禁止掉

解决方案：

在资源文件新建xml目录，新建文件

```
<?xml version="1.0" encoding="utf-8"?>

<network-security-config>

    <base-config cleartextTrafficPermitted="true" />
```

</network-security-config>

清单文件配置: android:networkSecurityConfig="@xml/network_security_config"

但还是建议都使用https进行传输。

3、8.0，静态广播无法正常接收

问题原因: Android 8.0 引入了新的广播接收器限制，因此您应该移除所有为隐式广播 Intent 注册的广播接收器。

解决方案:

使用动态广播代替静态广播

4、Caused by: java.lang.IllegalStateException: Only fullscreen opaque activities can request orientation

问题原因: Android 8.0 非全屏透明页面不允许设置方向

解决方案: android:windowIsTranslucent设置为false

总结

注意事项

1. 随着Google新的版本发布，项目应尽快跟进兼容。否则后面有多个版本的迭代，遗留的坑会比较多
2. 项目可在独立分支升级兼容，然后进行业务功能测试，发现问题进行针对性处理。可先解决一些闪退，功能不正常的问题。
3. 多参考Google官方的版本适配介绍，有详细的技术指导

参考文献

- Google官方文档

<https://developer.android.google.cn/preview/>

- **Android P版本应用兼容性适配技术指导**

<https://devcenter.huawei.com/consumer/cn/devservice/doc/50115>

文中部分问题没有指出如何适配，可以参考上面的华为的技术文章。

此外给大家贴一个 Google 的应用兼容性最佳实践：

来源：Google Developers

包括 Android O、Android P（预览版）的新特性、行为变更、应用开发中常见的兼容性最佳实践，以及测试、兼容系统时需要注意的一些事项。

推荐阅读：

[玩 Android 又添加了一个很赞的功能](#)

[推荐几个优质Flutter 开源项目](#)

[视频切换无缝衔接 打造丝滑体验](#)



公众号

扫一扫 关注我的公众号

如果你想要跟大家分享你的文章，欢迎投稿~

└(^0^)- 明天见！

[阅读原文](#)