

Android 0 适配详细指南

ChayWong

作者: ChayWong

链接:

<https://www.jianshu.com/p/567348203016>

本文由作者授权发布。

I

前言

最近 Google 对 Google Play 上架的应用提出了目标 API 等级要求。

从 2018 年 8 月 1 日起, 所有向 Google Play 首次提交的新应用都必须针对 Android 8.0 (API 等级 26) 开发; 2018 年 11 月 1 日起, 所有 Google Play 的现有应用更新同样必须针对 Android 8.0。

- **Google Play 目标 API 等级 (targetSdkVersion) 重要变更要求**

<https://developer.android.com/distribute/best-practices/develop/target-sdk?hl=zh-cn>

同时, 国内的华为、360、应用宝也要求开发者适配 Android P, 否则应用将被不推荐、隐藏甚至下架(华为), 可以看出国内应用市场对于推动应用适配新 API 的决心, 虽然没有强制要求适配, 但也算国内应用市场的一大进步, 相信很快就会有其他应用市场跟进。

为保障华为用户的使用体验, 华为应用市场已在7月份启动Android P版本应用适配检测工作, 针对未做适配的应用开发者陆续进行邮件通知。

请您对应用适配这一环节加以重视, 并于2018年10月底前完成Android P版本适配工作并自检通过。针对未适配或在Android P版本体验欠佳的应用, 华为应用市场将在Android P版本机型上采取下架、不推荐更新或屏蔽策略, 可能会对您的推广、用户口碑及品牌产生影响。

- **华为公告**

<https://developer.huawei.com/consumer/cn/notice/20180915>

- **360公告**

<http://bbs.360.cn/thread-15374485-1-1.html>

- **应用宝公告**

2

意义

我们都知道每次 Android 版本的更新都会新增一大波优化功能，比如 Android M 上引入了运行时权限，Android N 上带来了 Doze 模式，Android O 上的后台执行限制

总的来说，新版本会让我们的 Android 设备更流畅，更省电，隐私性更好。

应用也可以利用新版本的特性以提升用户体验，如 Android M 引入可指定通知栏颜色，以实现完全沉浸式体验。

所以当然要适配新版本啦

新版本新增了很多优化，但同时 APP 的限制也更多，比如在新版本上应用保活越来越困难，应用无法在后台做一些偷偷摸摸的事情，有些应用提供商肯定不乐意啦，因此国内很多应用对 Android API 适配的积极性不高。

现在各大应用市场联合起来催促开发者适配 Android API，这种情况很定会慢慢有所改观，国内的 Android 应用生态也会越来越好。

3

正文

扯了这么多，说到底，还是想说为了国内的 Android 应用生态，大家赶紧适配起来新版本吧。

俗话说，不能一口吃个胖子，因此我们先从 Android O 适配器来，也能达到 Google 的要求。

下面是考拉 APP 适配 Android O 的记录

1. 通知栏

Android 8.0 引入了通知渠道，其允许您为要显示的每种通知类型创建用户可自定义的渠道。用户界面将通知渠道称之为通知类别。

针对 8.0 的应用，创建通知前需要创建渠道，创建通知时需要传入 channelId，否则通知将不会显示。示例代码如下：

```
// 创建通知渠道
```

```
private void initNotificationChannel() {
```

```

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {

            CharSequence name = mContext.getString(R.string.app_name);

            NotificationChannel channel = new NotificationChannel(mChannelId, name, NotificationManager.IMPORTANCE_DEFAULT);

            mNotificationManager.createNotificationChannel(channel);

        }

    }

    // 创建通知传入channelId

    NotificationCompat.Builder builder = new NotificationCompat.Builder(context, NotificationBarManager.getInstance().getChannelId());

```

<https://developer.android.com/about/versions/oreo/android-8.0?hl=zh-cn#notifications>

2. 后台执行限制

如果针对 Android 8.0 的应用尝试在不允许其创建后台服务的情况下使用 `startService()` 函数，则该函数将引发一个 `IllegalStateException`。

我们无法得知系统如何判断是否允许应用创建后台服务，所以我们目前只能简单 `try-catch startService()`，保证应用不会 crash，示例代码：

```

Intent intent = new Intent(getApplicationContext(), InitializeService.class);

intent.setAction(InitializeService.INITIALIZE_ACTION);

intent.putExtra(InitializeService.EXTRA_APP_INITIALIZE, appInitialize);

ServiceUtils.safeStartService(mApplication, intent);

public static void safeStartService(Context context, Intent intent) {

    try {

        context.startService(intent);

    } catch (Throwable th) {

        DebugLog.i("service", "start service: " + intent.getComponent() + "error: " + th);

        ExceptionUtils.printStackTrace(th);

    }

}

```

<https://developer.android.com/about/versions/oreo/android-8.0-changes?hl=zh-cn#back-all>

3. 允许安装未知来源应用

针对 8.0 的应用需要在 AndroidManifest.xml 中声明 REQUEST_INSTALL_PACKAGES 权限，否则将无法进行应用内升级。

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

4. 主题的 Activity 设置屏幕方向

针对 8.0 的应用，设置了透明主题的Activity，再设置屏幕方向，代码如下：

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

    <item name="android:windowIsTranslucent">true</item>

</style>

<activity

    android:name=".MainActivity"

    android:screenOrientation="portrait"

    android:theme="@style/AppTheme">

</activity>
```

将会抛出以下异常：

```
java.lang.IllegalStateException: Only fullscreen opaque activities can request orientation
```

大概意思是：只有不透明的全屏Activity可以自主设置界面方向。

即使满足上述条件，该异常也并非一定会出现，为什么这么说，看下面两种表现：

- targetSdk=26，满足上述条件，API 26 手机没问题，API 27 手机没问题
- targetSdk=27，满足上述条件，API 26 手机Crash，API 27 手机没问题

有点摸不清 Google 的套路了……

可知，targetSdk=26 时，API 26 和 27 都没有问题，所以这个坑暂时放在适配 API 27 时再填吧。

5. 桌面图标适配

针对 8.0 的应用如果不适配桌面图标，则应用图标在 Launcher 中将会被添加白色背景：



适配方法：

[一起来学习Android 8.0系统的应用图标适配吧](#)

适配后的效果：



6. 隐式广播

由于 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。将它们留在原位并不会在构建时或运行时令应用失效，但当应用运行在 Android 8.0 上时它们不起任何作用。

显式广播 Intent（只有您的应用可以响应的 Intent）在 Android 8.0 上仍以相同方式工作。

这个新增限制有一些例外情况。如需查看在以 Android 8.0 为目标平台的应用中仍然有效的隐式广播的列表，请参阅隐式广播例外。

<https://developer.android.com/about/versions/oreo/android-8.0-migration?hl=zh-cn#rbr>

我对隐式广播的理解：

未指定广播接收器类名，通过 Action 发送。如有不妥，还请指教。

需要检查应用静态注册的隐式广播，需要改为动态注册。

7. 网络连接和 HTTP(S) 连接

Android 8.0 对网络连接和 HTTP(S) 连接行为做出了以下变更：

- 无正文的 OPTIONS 请求具有 Content-Length: 0 标头。之前，这些请求没有 Content-Length 标头。
- HttpURLConnection 在包含斜线的主机或颁发机构名称后面附加一条斜线，使包含空路径的网址规范化。例如，它将 `http://example.com` 转化为 `http://example.com/`。

- 通过 `ProxySelector.setDefault()` 设置的自定义代理选择器仅针对所请求的网址（架构、主机和端口）。因此，仅可根据这些值选择代理。传递至自定义代理选择器的网址不包含所请求的网址的路径、查询参数或片段。
- URI 不能包含空白标签。

之前，平台支持一种权宜方法，即允许主机名称中包含空白标签，但这是对 URI 的非法使用。此权宜方法只是为了确保与旧版 libcore 兼容。开发者如果对 API 使用不当，将会看到一条 ADB 消息：“URI `example..com` 的主机名包含空白标签。此格式不正确，将不被未来的 Android 版本所接受。”Android 8.0 废除了此权宜方法；系统对格式错误的 URI 会返回 `null`。

- Android 8.0 在实现 `HttpsURLConnection` 时不会执行不安全的 TLS/SSL 协议版本回退。
- 对隧道 HTTP(S) 连接处理进行了如下变更：

在通过连接建立隧道 HTTP(S) 连接时，系统会在 Host 行中正确放置端口号（:443）并将此信息发送至中间服务器。之前，端口号仅出现在 CONNECT 行中。

系统不再将隧道连接请求中的 `user-agent` 和 `proxy-authorization` 标头发送至代理服务器。

在建立隧道时，系统不再将隧道 `Http(s)URLConnection` 中的 `proxy-authorization` 标头发送至代理。相反，由系统生成 `proxy-authorization` 标头，在代理响应初始请求发送 HTTP 407 后将其发送至此代理。

同样地，系统不再将 `user-agent` 标头由隧道连接请求复制到建立隧道的代理请求。相反，库为此请求生成 `user-agent` 标头。

- 如果之前执行的 `connect()` 函数失败，`send(java.net.DatagramPacket)` 函数将会引发 `SocketException`。如果存在内部错误，`DatagramSocket.connect()` 会引发 `pendingSocketException`。对于 Android 8.0 之前的版本，即使 `send()` 调用成功，后续的 `recv()` 调用也会引发 `SocketException`。为确保一致性，现在这两个调用均会引发 `SocketException`。
- 在回退到 TCP Echo 协议之前，`InetAddress.isReachable()` 会尝试执行 ICMP。

对于某些屏蔽端口 7（TCP Echo）的主机（例如 `google.com`），如果它们接受 ICMP Echo 协议，现在也许能够访问它们。

对于确实无法访问的主机，此项变更意味着调用需要两倍的时间才能返回结果。

<https://developer.android.com/about/versions/oreo/android-8.0-changes?hl=zh-cn#networking-all>

这点应用一般无需适配。

8. 视图焦点

可点击的 View 对象现在默认也可以成为焦点。如果您希望 View 对象可点击但不可成为焦点，请在包含 View 的布局 XML 文件中将 `android:focusable` 属性设置为 `false`，或者将 `false` 传递至应用界面逻辑中的 `setFocusable()`。

<https://developer.android.com/about/versions/oreo/android-8.0-changes?hl=zh-cn#o-vf>

这点基本无需适配。

9. 权限

在 Android 8.0 之前，如果应用在运行时请求权限并且被授予该权限，系统会错误地将属于同一权限组并且在清单中注册的其他权限也一起授予应用。

对于针对 Android 8.0 的应用，此行为已被纠正。系统只会授予应用明确请求的权限。然而，一旦用户为应用授予某个权限，则所有后续对该权限组中权限的请求都将被自动批准。

例如，假设某个应用在其清单中列出 `READ_EXTERNAL_STORAGE` 和 `WRITE_EXTERNAL_STORAGE`。应用请求 `READ_EXTERNAL_STORAGE`，并且用户授予了该权限。如果该应用针对的是 API 级别 24 或更低级别，系统还会同时授予 `WRITE_EXTERNAL_STORAGE`，因为该权限也属于同一 `STORAGE` 权限组并且也在清单中注册过。如果该应用针对的是 Android 8.0，则系统此时仅会授予 `READ_EXTERNAL_STORAGE`；不过，如果该应用后来又请求 `WRITE_EXTERNAL_STORAGE`，则系统会立即授予该权限，而不会提示用户。

<https://developer.android.com/about/versions/oreo/android-8.0-changes?hl=zh-cn#rmp>

考拉中的权限都是按需申请的，不需要修改。

10. Tinker

特别是在Android N之后，由于混合编译的inline策略修改，对于市面上的各种方案都不太容易解决。而Tinker热补丁方案不仅支持类、So以及资源的替换，它还是2.X-8.X(1.9.0以上支持8.X)的全平台支持。

<https://github.com/Tencent/tinker/wiki>

经测试，Tinker在8.0上功能正常。

总结

随着 Android 版本的不断迭代，Android 系统的体验已经越来越好了。

Android 系统的绚烂多彩离不开广大的 Android 开发者，我们作为开发者，我们应该尽快适配 Android 新版本，让我们的应用拥有最好的体验。

本篇介绍了 Android O 的适配要点，如有不足，还请指教。

下一篇将会介绍 Android P 的适配，敬请期待。

参考：

- Android 绿色应用公约

<https://green-android.org/>

- 统一推送联盟

<http://chinaupa.com/>

最后推荐一下我做的网站，玩Android: wanandroid.com，包含详尽的知识体系、好用的工具，还有本公众号文章合集，欢迎体验和收藏！

推荐阅读:

[Android 技能图谱学习路线](#)

[屯书的时候到了~](#)



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](#)