重要变更 | Android 11 中的软件包可见性

谷歌开发者
已认证的官方帐号

已关注

11 人赞同了该文章

在 Android 10 及之前的版本中，应用可以通过 queryIntentActivities() 这样的方法获取到设备中所有已安装的应用列表。在大多数情况下，这种访问权限远超出了应用实际所需要的权限范围。随着我们不断加强对隐私保护的关注，我们将在 Android 11 上引入一些新的变化，从而改变应用查询用户已安装应用并与之交互的方式。为了达到这一目的，我们为特定设备上所安装的应用列表带来了更好的访问控制。

为了更好地 "问责" 访问已安装应用的行为，默认情况下，以 Android 11 为目标平台 (目标 API level 为 30) 的应用默认将只能检测到部分过滤后的已安装应用。如果想获取更多别的已安装应用列表信息，则需要在应用内的 Android manifest 中添加 元素，从而拓宽访问范围。

在 **大部分常见场景** 下，包括任何以 startActivity() 启动的 intents，您不需要做任何改动。而 **其他场景**，比如从您应用的界面中直接打开某个特定的第三方应用，则需要开发者们显式地声明应用的包名或者 **intent filter 签名**，如下所示:

```
<manifest package="com.example.game">
  <queries>
   <!-- 声明所要进行交互的应用名 -->
    <package android:name="com.example.store" />
    <package android:name="com.example.service" />

    <!--
        声明要查询的 intents

        例如下列标签声明了一个自定义的分享视图的 intent
    -->

    <intent>
      <action android:name="android.intent.action.SEND" />
```

```
        <data android:mimeType="image/jpeg" />
      </intent>

    </queries>

    ...

</manifest>
```

如果您使用 **Custom Tab** 来打开 URL 链接，您也许会调用 resolveActivity() 和 queryIntentActivities() 来启动一个非浏览器应用 (前提是您安装了处理该 URL 的应用)。在 Android 11 中，则有 **更好的办法** 来对此进行处理: 使用 intent 的 **FLAG_ACTIVITY_REQUIRE_NON_BROWSER** 标记，而不是去查询其他的应用。如果在您使用此标记调用 startActivity() 时启动了浏览器，则会抛出一个 ActivityNotFoundException 异常，此时您的应用可以对此异常进行处理，转而使用 Custom Tab 来打开 URL 链接。

```
try {
    val intent = Intent(ACTION_VIEW, Uri.parse(url)).apply {

        // 非浏览器应用会直接处理该 URL（默认情况下）
        // 用户也可以在消除歧义对话框中选择非浏览器应用

        addCategory(CATEGORY_BROWSABLE)
        flags = FLAG_ACTIVITY_NEW_TASK or FLAG_ACTIVITY_REQUIRE_NON_BROWSER
    }
    startActivity(intent)
} catch (e: ActivityNotFoundException) {
    // 只能使用浏览器应用，或者默认使用浏览器处理该 intent。
}
```

在极少数情况下，您的应用可能需要查询设备上所有已安装的应用或与之进行交互，不管这些应用包含哪些组件。为了允许您的应用看到其他所有已安装应用，Android 11 引入了 **QUERY_ALL_PACKAGES** 权限。在即将发布的政策更新中，Google Play 会为需要 QUERY_ALL_PACKAGES 权限的应用提供相关指南。

您可以将 API Level 设为 30，并使用 Android Studio 3.2 以上和最新发布的相应 Android Gradle 插件，即可在应用中添加 元素。您可以在 **开发者文档 — Android 11 中的软件包可见性** 中找到更多有关软件包可用性的使用信息和用例。

Android Studio 和 Gradle 对该功能的支持

如果您使用的 Android Gradle 插件版本是 4.1 和以上版本的话，就可以正常使用新的 元素，因为旧版本的 Gradle 插件并不兼容此元素。如果您使用了 ，或者是依赖了支持 Android 11 的库或 SDK，则可能会引起 manifest 冲突从而出现合并 manifest 的错误。例如，在构建应用时，在 Build Output Window 中可能会看到以下错误:

```
Android resource linking failed

/Users/sample/AndroidStudioProjects/MyApp/app/build/intermediates/merged_manife
```

在 Build Output Window 中可能还会出现这样一条错误信息，引导您去查看 Manifest 合并日志 (Manifest merger logs):

```
Manifest merger failed with multiple errors, see logs
```

展开 **Merged Manifest 视图**后，会出现一条附加的报错信息:

```
Error: Missing 'package' key attribute on element package
```

修复 Android Gradle 插件的问题

解决以上错误的最好办法就是将 **Android Gradle 插件升级到 4.1 Beta 版本**。

但是，并不是所有开发者都能够使用最新的版本，一些项目中可能会依赖老版本的 Gradle 或者代码库，而它们与 4.1 版本的 Android Gradle 插件有兼容性问题。

因此，近期我们为 Android Gradle 插件发布了一个 **小版本 (dot releases) 的升级**，以便兼容 元素:

当前您使用的 Android Gradle 插件版本	建议您升级至版本
4.1.*	N/A（不需升级）
4.0.*	4.0.1
3.6.*	3.6.4
3.5.*	3.5.4
3.4.*	3.4.3
3.3.*	3.3.3

举个例子，如果您正在使用 4.0.0 版本的 Android Gradle 插件，就可以在项目级别的 build.gradle 文件中将相关依赖升级到上图中对应的版本。

```
buildscript {
    repositories {
        google()
        jcenter()
    }

    dependencies {
        // classpath 'com.android.tools.build:gradle:4.0.0'
        classpath 'com.android.tools.build:gradle:4.0.1'
    }
}
```

了解更多 Android 11 相关信息，请查阅以下资源:

- **Android 11 中的软件包可见性文档**
- **Android Gradle 插件版本说明**

发布于 10-14

Android　Android 开发