



首页 > 程序开发 > 移动开发 > Android > 正文

Android5.0 , 6.0 , 7.0新特性整理

2017-01-20 10:01:27

0个评论

来源：站在风口.....

收藏

我要投稿



Unable to Connect to Site.

The Proxy was unable to connect to the remote site. This typically occurs if the site is down or not responding to requests. If you feel you have reached this page in error, or believe the site to be up -

背景

Android5.0 (Android Lollipop) 是谷歌公司2014年10月发布的全新安卓系统，至今已经两年多。然而由于国产手机对安卓ROM的深度定制或修改，以及手机厂商、芯片制造商、运营商之间错综复杂的关系，我们更新最新安卓版本往往需要1年左右时间的等待。2015年5月发布了Android6.0 (Android Marshmallow)，2016年8月发布了Android7.0(Android Nougat)。目前来说，用上6.0系统的手机都不太多，Android4.4和Android5.1版本占比重较高。但是，随着时间推移，手机陆续都会升级到更高的系统版本，所以，提前学习到各版本的新特性，把一下好的效果添加到代码中，并尽量做到向下兼容，应用才能拥有更好的体验，面对升级过后的系统才能从容不迫。

概述内容

从Android 5.0 (Android Lollipop) 开始，android迎来了扁平化时代，使用一种新的Material Design 设计风格，设计了全新的通知中心，开始支持多种设备。在性能上，放弃了之前一直使用的Dalvik虚拟机，改用ART模式，程序加载时间大幅提升。增加了Battery Saver模式来进行省电处理，以及全新的“最近应用程序”。

Android 6.0 (Android Marshmallow) 最大的亮点是为用户提供了两套相互独立的解决方案，简单的说，就是为每位用户的每一个应用提供了两套数据存储方案。一套存储工作资料，另一套存储个人信息。另外Android M系统层面加入指纹识别，还加入了运行时权限，加入了App Standby (应用待机)、Doze (瞌睡) Exemptions (豁免) 等模式来加强电源管理。

Android 7.0 (Android Nougat)，在性能处理有了巨大的提升，同时对文件数据加密，更加安全。 添加了分屏多任务，重新设计了通知，改进Doze休眠机制等。总而言之，Android N将更快，更高，更强。

Android 5.0新特性

Material Design

提起Android5.0，就不得不说Material Design，扁平化的设计理念。新的视觉语言，在基本元素的处理上，借鉴了传统的印刷设计，字体版式、网格系统、空间、比例、配色、图像使用等这些基础的平面设计规范。

之前的一些需要自定义的效果，现在都提供了系统级的支持，用起来更加方便了，而且android提供的效果看起来更加流畅。这里篇幅较多，本着不重复造轮子的思想，笔者找了一片写的很全面的Material Design的网站，图文并茂，值得学习。



Unable to Conn

The Proxy was unable to connect to the remote site. responding to requests. If you feel you have reached please submit a ticket via the link provided below.

URL: http://pos.baidu.com/s?hei=250&wid=300&di=u%2Fwww.2cto.com%2Fkf%2F201701%2F591290.htr exps=111000&cce=true&cdo=-1&drs=1&tlm=150348!

文章	推荐
· Android二维码扫描	
· Android热补丁动态修复技术（四）：自	
· Android中基于RxJava的响应式编程	
· android 中vector的用法的一些看法	
· Android 6.0 IMS流程(一)——IMS开机	
· HTTPS理论基础及其在Android中的最佳实	
· 安卓Zygote详解	
· 关于Fragment的方方面面	



Unable to Conn

The Proxy was unable to connect to the remote site. responding to requests. If you feel you have reached please submit a ticket via the link provided below.

URL: http://pos.baidu.com/s?hei=250&wid=300&di=u%2Fwww.2cto.com%2Fkf%2F201701%2F591290.htr oss=1578x17946&ant=0<r=https%3A%2F%2Fwww.

点击排行
· [Android Studio 权威教程] 断点调
· RecyclerView完全解析,让你从此爱上它
· Android M新控件之AppBarLayout , Nav
· Android Studio中如何引用图片资源
· Android之开发常用颜色
· Android Studio安装配置详细步骤（图
· 框架模式 MVC 在Android中的使用
· Android的AlertDialog详解



Unable to Conn

The Proxy was unable to connect to the remote site. responding to requests. If you feel you have reached please submit a ticket via the link provided below.

URL: http://pos.baidu.com/s?hei=250&wid=300&di=u%2Fwww.2cto.com%2Fkf%2F201701%2F591290.htr ari=2&par=1591x1017&dtm=HTML POST&oss=1578

本站名称: 红黑联盟


谷歌的Android系统已经可以在所有设备的基础上运行。不过，每个设备都有它自己的独特之处。

**证书 + 能力**
安全工程师 软件工程师 网站工程师 网络工程师 电脑工程师
为新手量身定做的课程，让菜鸟快速变身高手 正规公司助您腾飞

不断增加新科目
立即加入

谷歌在Android Lollipop中加入了全新风格的**通知系统**。改进后的通知系统会优先显示对用户来说比较重要的信息，而将不太紧急的内容隐藏起来。用户只需要向上滑动就可做查看全部的通知内容。

新的通知系统另外一个很酷的新功能是在锁屏界面也可以直接查看通知消息了。不仅如此，用户还可以直接在锁屏的情况下就行回复或进入应用。另外，如果在操作手机的过程中有电话进入，也不会进行全画面切换，而是同样以弹出通知的方式告知用户。

支持64位ART虚拟机

新系统不仅在视觉效果上带来了巨大的变化，Android Lollipop还在内部的性能上进行了飞跃。首先，新系统放弃了之前一直使用的Dalvik虚拟机，改用了ART模式，实现了真正的跨平台编译，在ARM、X86、MIPS等，无处不在。

ART 虚拟机编译器在内存占用及应用程序加载时间上进行了大幅提升，谷歌承诺所有性能都会比原来提升一倍。另外，对64位的支持也让ART虚拟机如鱼得水，开发者可以针对像ARM Cortex-A57这样的64位架构核心开发应用程序。Android Lollipop支持更大的寄存器，支持新的指令集，提升了内存寻址空间，未来Android智能手机将支持4GB以上的内存。

Project Volta电池续航改进计划

Project Volta计划增加了新工具可以让开发者能够更容易的找出为何自己的应用程序会对电量产生比较大的影响，同时确保在执行某型任务时将手机电量的影响降至最低。首先，Battery Historian可以列出手机电量消耗的详细情况，帮助开发者识别电量消耗的原因或者是哪个硬件或任务对电池寿命的影响比较大；而Job Scheduler API则可以让开发者更容易的选择合适的时机触发电量消耗比较高的任务，避免在低电量或未完成充电时更新应用程序。

上面是针对开发者的改进，而在用户层面上，Android Lollipop增加了Battery Saver模式，这与三星和HTC上的超级省电模式有些类似。在低电量的时候系统会自动降低屏幕亮度、限制自动更换背景等功能。

全新的“最近应用程序”

除了界面风格设计的改变之外，新的最近应用界面还借鉴了Chrome浏览器的理念，采用单独的标签展示方式。更重要的是，谷歌已经向开发者开放了API，所以第三方开发人员可以利用这个改进为特定的应用增加全新的功能。

改进安全性

现在个人识别解锁还是一个比较新鲜的概念，当用户的蓝牙耳机连接到手机或平板电脑时，设备可以基于当前的位置或用户的声音自动解锁。比如当特定的智能手表出现在Android设备的附近，那么就会直接绕过锁屏界面进行操作。而Android Lollipop也增加了这种针对特定特任识别解锁的模式。换句话说，当设备没有检测到附近有可用的信任设备时，就会启动安全模式防止未授权访问。

另外，Android Lollipop还默认开启了系统数据加密功能，并且通过SELinux执行应用程序，这就意味着对于恶意软件来说，新系统变得更加安全。

不同数据独立保存

谷歌表示Android Lollipop将拥有一个全新的特性，让用户通过一台设备就可以搞定所有的工作和休闲娱乐活动。该特性首先将各种数据独立保存，并且让所有新数据的生成都有依据。

我们已经看到谷歌已经与三星加强了合作，包括三星的Knox安全系统同样可以像Android

Lollipop一样将重要数据和其它数据分开保存。另外谷歌还允许向三星以外的Android设备提供支持。

改进搜索

谷歌将新系统的搜索功能重点放在了“重新发现”上，因此这意味着Google Search将会更好的意识到用户正在做什么。比如系统会根据用户当前的位置自动过滤无关的搜索结果。

另外，当用户在进行应用搜索时，可以直接展示相似或部分提示，并且进入特定的应用程序而无需将内容全部输入。

支持蓝牙4.1、USB Audio、多人分享等其它特性

Android Lollipop还增加了多个新的API支持、蓝牙4.1、USB Audio外接音响及多人分享等功能。其中多人分享功能可以在用户手机丢失的情况下，使用其它Lollipop设备登录账户，从云端下载联系人、日历等资料，并且不影响其它设备的内容。

Android 6.0 新特性

动态权限管理

Android 6.0引入了动态权限管理，将标记为危险的权限从安装时权限(Install Time Permission)模型移动到运行时权限模型（Runtime Permissions）：

安装时权限模型(Android5.1以及更早)：用户在应用安装和更新时，对危险权限授权。但是OEM和运行商预装的应用将自动预授权。

运行时权限(Android6.0及以后)：用户在应用运行时，对应用授予危险权限。由应用决定何时去申请权限（例如，在应用启动时或者用户访问某个特性时），但必须容许用户来授予或者拒绝应用对特定权限组的访问。OEM和运营商可以预装应用，但是不能对权限进行预授权。

运行时权限提供给用户关于应用所需权限更多的相关上下文和可视性，这也让开发者帮助用户更好的理解：为什么应用需要所请求的权限，授权将有什么样的好处，拒绝将有何种不便。用户可以通过设置中的菜单来撤销应用的权限。

受影响的权限

系统要求标记为危险（dangerous）的权限使用运行时权限模型。查看危险权限列表可以使用命令：adb shell pm list permissions -g -d。

Android6.0并不改变正常权限的行为。正常权限指的是所有非危险（non-dangerous）权限，包括normal，system和signature 权限。正常权限风险较低，用于容许应用以最小风险来访问隔离的应用级别的特性。在Android5.1和早期版本中，系统在安装时自动将正常权限授予请求的应用，并不提示用户

请求权限

运行时权限模型适用于所有应用，包括预装应用。应用软件的要求包括：

运行时权限模型必须在所有运行Android6.0的设备上是一致的。这通过CTS来实施。

应用必须在运行时提示用户进行授权。

带有危险权限的预装应用，必须符合API level 23，必须维护Android6.0的AOSP权限模型（例如，应用安装的UI流程不应该脱离AOSP的packageInstaller的实现；用户可以撤销预装应用的危险权限；等等）。

无界面的应用必须使用Activity来申请权限，或者与其他有相应权限的应用共享UID。

权限迁移

在设备从Android5.x升级到Android6.0之后，授予应用的权限仍然有效。但是用户可以在任何时候撤销这些权限。

指纹识别（Fingerprint Support）

说到指纹识别，很多用户都会觉得现在的中高端安卓手机都支持，但事实上这些安卓手机的指纹识别都是各个厂商自行开发的并没有系统底层的支持。Android 6.0则在系统层面加入指纹识别，能提供原生指纹识别API，这不但降低了厂商开发指纹识别模块的成本，最重要的是原生指纹识别将会大大提升安卓手机的指纹识别支付安全性。

APP关联（App Links）

Android M加强了软件间的关联，谷歌在现场展示了一个例子，比如你的手机邮箱里收到一封邮件，内文里有以个Twitter链接，用户点击该链接可以直接跳转到Twitter应用，而不再是网页。

Chrome App Links这个就相当于在Web中使用Scheme来进行App的跳转功能一样，只不过这里Chrome默认就支持了，用户只需要搜索相关的关键词，就可以直接显示App，然后通过App Link直接启动App。这个东西对用户来说应该是一件非常好、方便的事情，但由于国内这些浏览器的问题，不知道能否会兼容这个功能。

Android pay

Android支付统一标准。新的M系统中集成了Android Pay。其特性在于简洁、安全、可选性。Android Pay是一个开放性平台，用户可以选择谷歌的服务或者使用银行的APP来使用它，Android Pay支持4.4以后系统设备，谷歌宣布Android Pay已经与美国三大运营商700多家商店达成合作，支付功能可以使用指纹来进行支付。

电源管理

Android6.0引入了下列特性来延长电池使用时间：

App Standby（应用待机）：系统可以将未使用的应用置为AppStandby模式，临时限制这些应用的网络访问，延迟同步和后台job

Doze（瞌睡）：如果用户在某个时间期限内未主动使用设备（屏幕关闭、设备静止），平台将进入深度睡眠状态。因为该特性要求平台检测静止状态，因此只有那些在Sensor HAL中实现了显式移动检测API的设备中才有效。

Exemptions（豁免）：预装在设备上的系统应用和云消息服务，默认通常被豁免。应用开发者可以要求其应用使用这种设定。用户也可以在设置中来豁免某些应用。

App Standby

对于不常用的应用的，AppStandby通过延迟后台网络活动和任务来延长电池寿命。
生命周期

检测：当设备不在充电时，且在某个特定时钟期限内和亮屏时长中，用户没有直接或者间接启动该应用。间接启动指的是：前台应用访问另外某个应用的service。**App Standy**模式中：平台限制应用一天内最多访问一次网络，延迟其同步和其他后台任务。退出：当应用被激活，或者设备充电时，平台将该应用移出App Standby状态。

活动的应用不受AppStandby的影响。活动的含义是：

进程处于前台（Activity 或者前台服务，或者由另外一个Activity 或前台服务所使用），例如 notification listener，辅助服务，live 墙纸，等等。

供用户查看的通知，可以是锁屏通知或者通知图标。

用户显式启动的应用。

Doze

支持Doze功能的要求：

设备在Sensor HAL实现了significant motion detector (SMD) APIs 。

设备有某个云消息服务，如 Google Cloud Messaging (GCM)。这使得设备可以知道何时从Doze中唤醒。

生命周期

检测：当未在充电时，且当设备静止且灭屏了一段时间。 **Doze**期间：平台尝试让系统处于休眠状态，周期性地进入在一个维持窗口恢复正常操作，然后进入更长的休眠状态。在休眠状态：

禁止网络访问

忽略 wakelock

延迟闹钟，除了闹铃 和使用setAndAllowWhileIdle()设置的闹钟，以荣喜日历等应用显示事件提醒

停止WIFI扫描

同步和后台任务被推迟到下一个维护窗口。

接收短信和MMS消息的应用位于一个临时白名单，这样他们可以完成其工作。 退出：

用户交互

设备移动

亮屏

闹铃

集成Doze

确认设备支持 SENSOR_TYPE_SIGNIFICANT_MOTION . 确认设备安装了一种云消息服务。 在配置文件overlay/frameworks/base/core/res/res/values/config.xml中，修改：bool

name=”config_enableAutoPowerModes”>true 检查装应用和服务：

Use the new power-saving optimization guidelines. For details, see Testing and optimizing applications.

OR , Are exempted from Doze and App Standby. For details, see Exempting applications. 确认必要的服务从Doze中豁免。

Exemptions

你可以在设置中，手动将某些应用从Doze和AppStandby中排除。

存储

在之前，安卓系统虽然可以支持TF卡扩展，但TF卡扩展的容量只能当成外置存储。虽然用户可以把照片、音乐等内容放进去，但App不能安装到其中。在安卓6.0中，当首次插入TF卡的时候，系统会询问会将TF卡作为外置还是内置储存。如果选择作为内置存储，系统则会对TF卡格式化并加密——注意，这会丢失数据！然后，TF卡的容量就和设备的内置存储融合了。

一般来说，设备的内置存储速度会比TF卡快上不少，稳定性也更高，因此如果不是容量特别紧张，并不建议把TF当成内置储存植入。同时，市面上支持TF卡扩展的安卓设备也越来越少，安卓6.0的这个新特性也许会遭到冷遇。

Android 7.0 新特性

Android7.0提供新功能以提升性能、生产效率和安全性。

关于Android N的性能改进，Android N建立了先进的图形处理Vulkan系统，能少的减少对CPU的占用。与此同时，Android N加入了JIT编译器，安装程序快了75%，所占空间减少了50%。

在安全性上，Android N加入了全新安全性能，其中包括基于文件的数据加密。谷歌移动版Chrome能识别恶意网站。

Android N可以进行无缝更新，与Chromebook一样，用户将不再需要下载安装，也不再需要进行重启。

在效率提升上，Android N可以自动关闭用户较长时间未使用的应用程序。在通知上新增了直接回复功能，并支持一键全部清除功能

分屏多任务

进入后台多任务管理页面，然后按住其中一个卡片，然后向上拖动至顶部即可开启分屏多任务，支持上下分栏和左右分栏，允许拖动中间的分割线调整两个APP所占的比例。

全新下拉快捷开关页

在Android 7.0中，下拉打开通知栏顶部即可显示5个用户常用的快捷开关，支持单击开关以及长按进入对应设置。如果继续下拉通知栏即可显示全部快捷开关，此外在快捷开关页右下角也会显示一个“编辑”按钮，点击之后即可自定义添加/删除快捷开关，或拖动进行排序。

新通知消息

通知消息快捷回复

加入了全新的API，支持第三方应用通知的快捷操作和回复，例如来电会以横幅方式在屏幕顶部出现，提供接听/挂断两个按钮；信息/社交类应用通知，还可以直接打开键盘，在输入栏里进行快捷回复。

通知消息归拢

Android7.0会将同一应用的多条通知提示消息归拢为一项，点击该项即可展开此前的全部通知，允许用户对每个通知执行单独操作。

夜间模式

Android7.0中重新加入了夜间深色主题模式，该功能依然需要在系统调谐器中开启，从顶部下划打开快捷设置页，然后长按其中的设置图标，齿轮旋转10秒钟左右即可提示已开启系统调谐器，之后用户在设置中即可找到“系统调谐器”设置项。点开其中的“色彩和外观”，即可找到夜间模式，开启后即可使用全局的深色主题模式，同时亮度和色彩也会进行一定的调整，该功能可以基于时间或地理位置自动开启。另外，系统调谐器中也提供了RGB红绿蓝三色调节滑动条，允许用户手动精细调节，例如减少蓝色或增加红色以提供类似护眼模式的效果。

流量保护模式

新增的流量保护模式不仅可以禁止应用在后台使用流量，还会进一步减少该应用在前台时的流量使用。推测其有可能使用了类似Chrome浏览器的数据压缩技术。此外，谷歌还扩展了ConnectivityManager API的能力，使得应用可以检测系统是否开启了流量保护模式，或者检测自己是否在白名单中。Android7.0允许用户单独针对每个应用，选择是否开启数据保护模式。

全新设置样式

Android7.0启用了全新的设置样式，首先每个分类下各个子项之间的分割线消失了，只保留分类之间的分割线。全新的设置菜单还提供了一个绿色的顶栏，允许用户通过后方的下拉箭头，快速设定勿扰模式等。除了勿扰模式外，顶栏菜单还可以显示诸多其他的设置状态，例如数据流量的使用情况，自动亮度是否开启等。谷歌也在安卓7.0的设置中加入了汉堡菜单，在二级设置界面中的左上角，你就会看到这个汉堡菜单，点击后即可看到所有设置项，方便用户快速跳转。

改进的Doze休眠机制

谷歌在Android7.0中对Doze休眠机制做了进一步的优化，在此前的Android6.0中，Doze深度休眠机制对于改善安卓的续航提供了巨大的作用。而在Android67.0中，谷歌对Doze进行了更多的优

化，休眠机制的使用规则和场景有所扩展，例如只要手动在后台删掉应用卡片，关屏后该应用就会被很快深度休眠。

系统级电话黑名单功能

Android7.0将电话拦截功能变成了一个系统级功能。其它应用可以调用这个拦截名单，但只有个别应用可以写入，包括拨号应用、默认的短信应用等。被拦截号码将不会出现在来电记录中，也不会出现通知。另外用户也可以通过账户体系备份和恢复这个拦截名单，以便快速导入其它设备或账号。

菜单键快速应用切换

双击菜单键，就能自动切换到上一个应用。此外，如果你不停地点击菜单键的话，就会在所有应用中不间断地轮换，应用窗口会自动放大，顶部还会出现倒计时条，停止点击且倒计时结束后，当前应用会自动放大并返回到前台。

对于开发的影响

各个新版本的Android系统除了提供了诸多新特性和功能外，还对系统和API行为做出了各种变更。以下重点介绍您应该了解并在开发应用时加以考虑的一些主要变更。（其中某些变更与上面提到的新特性有重复的地方，姑且听之）

Android 5.0行为变更

API级别：21

1. Android Runtime（ART）

大多数 Android 应用无需任何更改就可以在 ART 下工作。不过，部分适合 Dalvik 的技术并不适用于 ART。如需了解有关最重要问题的信息，请参阅在 Android Runtime (ART) 上验证应用行为。如存在以下情况，应特别注意：

- 1

2

3
- 您的应用使用 Java 原生接口（JNI）运行 C/C++ 代码。

您使用生成非标准代码的开发工具（例如，一些代码混淆工具）。

您使用与压缩垃圾回收不兼容的技术
- ?

2. 声音和振动

如果您当前使用 Ringtone、MediaPlayer 或 Vibrator 类向通知中添加声音和振动，则移除此代码，以便系统可以在“优先”模式中正确显示通知。取而代之的是，使用 Notification.Builder 方法添加声音和振动。

将设备设为 RINGER_MODE_SILENT 可使设备进入新的优先模式。如果您将设备设为 RINGER_MODE_NORMAL 或 RINGER_MODE_VIBRATE，则设备将退出优先模式。

以前，Android 使用 STREAM_MUSIC 作为主流式传输来控制平板电脑设备上的音量。在 Android 5.0 中，手机和平板电脑设备的主音量流式传输现已合并，由 STREAM_RING 或 STREAM_NOTIFICATION 进行控制。

3. 锁定屏幕可见性

默认情况下，在 Android 5.0 中，通知现在显示在用户的锁定屏幕上。用户可以选择保护敏感信息不被公开，在此情况下，系统会自动删减通知显示的文本。要自定义此删减的通知，请使用 setPublicVersion()。

如果通知不包含个人信息，或者您想允许媒体播放控件显示在通知上，则调用 setVisibility() 方法并将通知的可见性级别设为 VISIBILITY_PUBLIC。

4. 浮动通知

现在，当设备处于活动状态时（即，设备未锁定且其屏幕已打开），通知可以显示在小型浮动窗口中（也称为“浮动通知”）。这些通知看上去类似于精简版的通知，只是浮动通知还显示操作按钮。用

户可以在不离开当前应用的情况下处理或清除浮动通知。

可能触发浮动通知的条件示例包括：

用户的 Activity 处于全屏模式中（应用使用 `fullScreenIntent`）

通知具有较高的优先级并使用铃声或振动

如果您的应用在以上任何情形下实现通知，请确保系统正确显示浮动通知。

5. `getRecentTasks()`

为提升用户隐私的安全性，现已弃用 `ActivityManager.getRecentTasks()` 方法。对于向后兼容性，此方法仍会返回它的一小部分数据，包括调用应用自己的任务和可能的一些其他非敏感任务（如首页）。如果您的应用使用此方法检索它自己的任务，则改用 `getAppTasks()` 检索该信息。

6. 绑定到服务

`Context.bindService()` 方法现在需要显式 Intent，如果提供隐式 intent，将引发异常。为确保应用的安全性，请使用显式 intent 启动或绑定 Service，且不要为服务声明 intent 过滤器。

7. `webview`

Android 5.0 更改了应用的默认行为。

如果您的应用是面向 API 级别 21 或更高级别：

默认情况下，系统会阻止混合内容和第三方 Cookie。要允许混合内容和第三方 Cookie，请分别使用 `setMixedContentMode()` 和 `setAcceptThirdPartyCookies()` 方法。

系统现在可以智能地选择要绘制的 HTML 文档部分。这个新的默认行为有助于减少内存占用和提升性能。如果您要一次渲染整个文档，可通过调用 `enableSlowWholeDocumentDraw()` 停用此优化。

如果您的应用是面向低于 21 的 API 级别：系统允许混合内容和第三方 Cookie，并始终一次渲染整个文档。

Android 6.0行为变更

API级别：23

1. 运行时权限

对于以 Android 6.0（API 级别 23）或更高版本为目标平台的应用，请务必在运行时检查和请求权限。要确定您的应用是否已被授予权限，请调用新增的 `checkSelfPermission()` 方法。要请求权限，请调用新增的 `requestPermissions()` 方法。即使您的应用并不以 Android 6.0（API 级别 23）为目标平台，您也应该在新权限模式下测试您的应用。

2. 取消支持Apache HTTP客户端

Android 6.0 版移除了对 Apache HTTP 客户端的支持。如果您的应用使用该客户端，并以 Android 2.3（API 级别 9）或更高版本为目标平台，请改用 `HttpURLConnection` 类。此 API 效率更高，因为它可以通过透明压缩和响应缓存减少网络使用，并可最大限度降低耗电量。要继续使用 Apache HTTP API，您必须先 `build.gradle` 文件中声明以下编译时依赖项：

```
1 | android {
2 |     useLibrary 'org.apache.http.legacy'
3 | }
```

3. BoringSSL

Android 正在从使用 OpenSSL 库转向使用 BoringSSL 库。如果您要在应用中使用 Android

NDK，请勿链接到并非 NDK API 组成部分的加密库，如 libcrypto.so 和 libssl.so。这些库并非公共 API，可能会在不同版本和设备上毫无征兆地发生变化或出现故障。此外，您可能让自己暴露在安全漏洞的风险之下。请改为修改原生代码，以通过 JNI 调用 Java 加密 API，或静态链接到您选择的加密库。

4. 通知

此版本移除了 Notification.setLatestEventInfo() 方法。请改用 Notification.Builder 类来构建通知。要重复更新通知，请重复使用 Notification.Builder 实例。调用 build() 方法可获取更新后的 Notification 实例。

adb shell dumpsys notification 命令不再打印输出您的通知文本。请改用 adb shell dumpsys notification --noredact 命令打印输出 notification 对象中的文本。

5. 音频管理器变更

不再支持通过 AudioManager 类直接设置音量或将特定音频流静音。setStreamSolo() 方法已弃用，您应该改为调用 requestAudioFocus() 方法。类似地，setStreamMute() 方法也已弃用，请改为调用 adjustStreamVolume() 方法并传入方向值 ADJUST_MUTE 或 ADJUST_UNMUTE。

6. 相机服务变更

在此版本中，相机服务中共享资源的访问模式已从之前的“先到先得”访问模式更改为高优先级进程优先的访问模式。对服务行为的变更包括：

根据客户端应用进程的“优先级”授予对相机子系统资源的访问权，包括打开和配置相机设备。带有对用户可见 Activity 或前台Activity 的应用进程一般会被授予较高的优先级，从而使相机资源的获取和使用更加可靠；

当高优先级的应用尝试使用相机时，系统可能会“驱逐”正在使用相机客户端的低优先级应用。在已弃用的 Camera API 中，这会导致系统为被驱逐的客户端调用 onError()。在 Camera2 API 中，这会导致系统为被驱逐的客户端调用 onDisconnected()；

在配备相应相机硬件的设备上，不同的应用进程可同时独立打开和使用不同的相机设备。但现在，如果在多进程用例中同时访问相机会造成任何打开的相机设备的性能或能力严重下降，相机服务会检测到这种情况并禁止同时访问。即使并没有其他应用直接尝试访问同一相机设备，此变更也可能导致低优先级客户端被“驱逐”。

更改当前用户会导致之前用户帐户拥有的应用内活动相机客户端被驱逐。对相机的访问仅限于访问当前设备用户拥有的用户个人资料。举例来说，这意味着，当用户切换到其他帐户后，“来宾”帐户实际上无法让使用相机子系统的进程保持运行状态

Android 7.0行为变更

API级别：24

1. 电池和内存

Android 7.0 包括旨在延长设备电池寿命和减少 RAM 使用的系统行为变更。这些变更可能会影响您的应用访问系统资源，以及您的应用通过特定隐式 intent 与其他应用交互的方式。

2. Project Svelte：后台优化

Android 7.0 移除了三项隐式广播，以帮助优化内存使用和电量消耗。此项变更很有必要，因为隐式广播会在后台频繁启动已注册侦听这些广播的应用。删除这些广播可以显著提升设备性能和用户体验。

移动设备会经历频繁的连接变更，例如在 WLAN 和移动数据之间切换时。目前，可以通过在应用

清单中注册一个接收器来侦听隐式 `CONNECTIVITY_ACTION` 广播，让应用能够监控这些变更。由于很多应用会注册接收此广播，因此单次网络切换即会导致所有应用被唤醒并同时处理此广播。

同理，在之前版本的 Android 中，应用可以注册接收来自其他应用（例如相机）的隐式 `ACTION_NEW_PICTURE` 和 `ACTION_NEW_VIDEO` 广播。当用户使用相机应用拍摄照片时，这些应用即会被唤醒以处理广播。

为缓解这些问题，Android 7.0 应用了以下优化措施：

面向 Android 7.0 开发的应用不会收到 `CONNECTIVITY_ACTION` 广播，即使它们已有清单条目来请求接受这些事件的通知。在前台运行的应用如果使用 `BroadcastReceiver` 请求接收通知，则仍可以在主线程中侦听 `CONNECTIVITY_CHANGE`。

应用无法发送或接收 `ACTION_NEW_PICTURE` 或 `ACTION_NEW_VIDEO` 广播。此项优化会影响所有应用，而不仅仅是面向 Android 7.0 的应用。

如果您的应用使用任何 `intent`，您仍需要尽快移除它们的依赖关系，以正确适配 Android 7.0 设备。Android 框架提供多个解决方案来缓解对这些隐式广播的需求。例如，`JobScheduler` API 提供了一个稳健可靠的机制来安排满足指定条件（例如连入无限流量网络）时所执行的网络操作。您甚至可以使用 `JobScheduler` 来适应内容提供程序变化。

3. 系统权限更改

为了提高私有文件的安全性，面向 Android 7.0 或更高版本的应用私有目录被限制访问 (0700)。此设置可防止私有文件的元数据泄漏，如它们的大小或存在性。此权限更改有多重副作用：

- 1.私有文件的文件权限不应再由所有者放宽，为使用 `MODE_WORLD_READABLE` 和/或 `MODE_WORLD_WRITEABLE` 而进行的此类尝试将触发 `SecurityException`。
注：迄今为止，这种限制尚不能完全执行。应用仍可能使用原生 API 或 `File` API 来修改它们的私有目录权限。但是，我们强烈反对放宽私有目录的权限。
- 2.传递软件包网域外的 `file://` URI 可能给接收器留下无法访问的路径。因此，尝试传递 `file://` URI 会触发 `FileUriExposedException`。分享私有文件内容的推荐方法是使用 `FileProvider`。
- 3.`DownloadManager` 不再按文件名分享私人存储的文件。旧版应用在访问 `COLUMN_LOCAL_FILENAME` 时可能出现无法访问的路径。面向 Android 7.0 或更高版本的应用在尝试访问 `COLUMN_LOCAL_FILENAME` 时会触发 `SecurityException`。通过使用 `DownloadManager.Request.setDestinationInExternalFilesDir()` 或 `DownloadManager.Request.setDestinationInExternalPublicDir()` 将下载位置设置为公共位置的旧版应用仍可以访问 `COLUMN_LOCAL_FILENAME` 中的路径，但是我们强烈反对使用这种方法。对于由 `DownloadManager` 公开的文件，首选的访问方式是使用 `ContentResolver.openFileDescriptor()`。

4. 在应用件共享文件

对于面向 Android 7.0 的应用，Android 框架执行的 `StrictMode` API 政策禁止在您的应用外部公开 `file://` URI。如果一项包含文件 URI 的 `intent` 离开您的应用，则应用出现故障，并出现 `FileUriExposedException` 异常。

要在应用间共享文件，您应发送一项 `content://` URI，并授予 URI 临时访问权限。进行此授权的最简单方式是使用 `FileProvider` 类。

5. 屏幕缩放

Android 7.0 支持用户设置显示尺寸，以放大或缩小屏幕上的所有元素，从而提升设备对视力不

佳用户的可访问性。用户无法将屏幕缩放至低于最小屏幕宽度 sw320dp，该宽度是 Nexus 4 的宽度，也是常规中等大小手机的宽度。

当设备密度发生更改时，系统会以如下方式通知正在运行的应用：

如果是面向 API 级别 23 或更低版本系统的应用，系统会自动终止其所有后台进程。这意味着如果用户切换离开此类应用，转而打开 Settings 屏幕并更改 Display size 设置，则系统会像处理内存不足的情况一样终止该应用。如果应用具有任何前台进程，则系统会如处理运行时更改中所述将配置变更通知给这些进程，就像对待设备屏幕方向变更一样。

如果是面向 Android 7.0 的应用，则其所有进程（前台和后台）都会收到有关配置变更的通知，如处理运行时更改中所述。

大多数应用并不需要进行任何更改即可支持此功能，不过前提是这些应用遵循 Android 最佳做法。具体要检查的事项：

- 1.在屏幕宽度为 sw320dp 的设备上测试您的应用，并确保其充分运行。
- 2.当设备配置发生变更时，更新任何与密度相关的缓存信息，例如缓存位图或从网络加载的资源。当应用从暂停状态恢复运行时，检查配置变更。
- 注：如果您要缓存与配置相关的数据，则最好也包括相关元数据，例如该数据对应的屏幕尺寸或像素密度。保存这些元数据便于您在配置变更后决定是否需要刷新缓存数据。
- 3.避免用像素单位指定尺寸，因为像素不会随屏幕密度缩放。应改为使用与密度无关像素 (dp) 单位指定尺寸。

6. 检查你的应用是否使用私有库

为帮助您识别加载私有库的问题，logcat 可能会生成一个警告或运行时错误。例如，如果您的应用面向 API 级别 23 或更低级别，并在运行 Android 7.0 的设备上尝试访问私有库，您可能会看到一个类似于下面所示的警告：

```
1 | 03-21 17:07:51.502 31234 31234 W linker : ?
2 | library "libandroid_runtime.so"("/system/lib/libandroid_runtime.so") |
```

这些 logcat 警告通知您哪个库正在尝试访问私有平台 API，但不会导致您的应用崩溃。但是，如果应用面向 API 级别 24 或更高级别，logcat 会生成以下运行时错误，您的应用可能会崩溃：

```
1 | java.lang.UnsatisfiedLinkError: dlopen failed: ?
2 | library "libcutils.so"("/system/lib/libcutils.so") needed or dlopen
3 | at java.lang.Runtime.loadLibrary0(Runtime.java:977)
4 | at java.lang.System.loadLibrary(System.java:1602)
```

如果您的应用使用动态链接到私有平台 API 的第三方库，您可能也会看到上述 logcat 输出。利用 Android 7.0DK 中的 readelf 工具，您可以通过运行以下命令生成给定 .so 文件的所有动态链接的共享库列表：

```
aarch64-linux-android-readelf -dW libMyLibrary.so
```

7. 其他重要说明

(1)如果一个应用在 Android 7.0 上运行，但却是针对更低 API 级别开发的，那么在用户更改显示尺寸时，系统将终止此应用进程。应用必须能够妥善处理此情景。否则，当用户从最近使用记录中恢复运行应用时，应用将会出现崩溃现象。

您应测试应用以确保不会发生此行为。要进行此测试，您可以通过 DDMS 手动终止应用，以造成相同的崩溃现象。

在密度发生更改时，系统不会自动终止面向 N 及更高版本的应用；不过，这些应用仍可能对配置变更做出不良响应。

(2)Android 7.0 上的应用应能够妥善处理配置变更，并且在后续启动时不会出现崩溃现象。您可以通过更改字体大小 (Setting >Display > Font size) 并随后从最近使用记录中恢复运行应用，来验证应用行为。

(3)由于之前的 Android 版本中的一项错误，系统未能将对主线程上的一个 TCP 套接字的写入操作举报为违反严格模式。Android 7.0 修复了此错误。呈现出这种行为的应用现在会引发 android.os.NetworkOnMainThreadException 。一般情况下，我们不建议在主线程上执行网络操作，因为这些操作通常会出现可能导致 ANR 和卡顿的高尾延迟。

(4)Debug.startMethodTracing() 方法系列现在默认在您的共享存储空间上的软件包特定目录中存储输出，而非 SD 卡根目录。这意味着应用不再需要请求 WRITE_EXTERNAL_STORAGE 权限来使用这些 API 。

(5)许多平台 API 现在开始检查在 Binder 事务间发送的大负载，系统现在会将 TransactionTooLargeExceptions 作为 RuntimeExceptions 再次引发，而不再只是默默记录或抑制它们。一个常见例子是在 Activity.onSaveInstanceState() 上存储过多数据，导致 ActivityThread.StopInfo 在您的应用面向 Android 7.0 时引发 RuntimeException。

(6)如果应用向 View 发布 Runnable 任务，并且 View 未附加到窗口，系统会用 View 为 Runnable 任务排队；在 View 附加到窗口之前，不会执行 Runnable 任务。此行为会修复以下错误：

如果一项应用是从并非预期窗口 UI 线程的其他线程发布到 View，则 Runnable 可能会因此运行错误的线程。

如果 Runnable 任务是从并非环路线程的其他线程发布，则应用可能会曝光 Runnable 任务。

(7)如果 Android 7.0 上一项有 DELETE_PACKAGES 权限的应用尝试删除一个软件包，但另一项应用已经安装了这个软件包，则系统需要用户进行确认。在这种情况下，应用在调用 PackageInstaller.uninstall() 时预计的返回状态应为 STATUS_PENDING_USER_ACTION 。

(8)名为 Crypto 的 JCA 提供程序已弃用，因为它仅有的 SHA1PRNG 算法为弱加密。应用无法再使用 SHA1PRNG（不安全地）派生密钥，因为不再提供此提供程序。

对于目前应用的更改建议

对于Android新特性的使用，不可能一蹴而就，但是手机在不断升级，Andriod6.0，7.0也会成为安卓阵营中的主力，Android应用也需要陆续的把这些新特性或者新API里的方法用上，做到向下兼容，紧跟时代潮流。

在实际过程中需要各方面协调。拿Android 5.0上引入的Material Design来说，从设计上讲，设计人员、美工和开发人员都需要了解一下Material Design设计语言，相互协调，统一风格；从开发上讲，需要根据新特性和api来优化或者适当重构代码；另外有必要重新编码基类和工具类，比如 notifications，toolbars，toast，cards，dialogs等来适应新的安卓风格。



Unable to Connect to Site.

点击复制链接 与好友分享!

[回本站首页](#)

相关TAG标签 [Android](#)

上一篇：[Android GridViewd 简单代码实现](#)

下一篇：[android自学之第一个HelloWorld程序](#)

相关文章

- | | |
|---|--|
| android libaray的创建和使用 | android中调用webservice |
| android中json转换成List | Android ListView Adapter |
| android AudioTrack不能播放awr | Android异步操作AsyncTask |
| android-仿QQtab | android database sqlite的使用 |

Android之ImageSwitcher,Gallery用法

Android面试系列之一

热门专题推荐

python

div+css

css教程html5html教程jquery

Android SDK

phpmysqloracle



图文推荐



Unable to Connect to Site.

Android零基础入门第

一个Activity中多个F

系统或JDK为32位时安

安卓人脸检测之FaceD

登录

来说两句吧...

畅言一下

还没有评论，快来抢沙发吧！

红黑联盟正在使用畅言

13 of 13

2017年08月23日 19:20