# Android-transulcent-status-bar

 发表于 2016-03-20 |  分类于 [Android适配经验](http://niorgai.github.io/categories/Android%E9%80%82%E9%85%8D%E7%BB%8F%E9%AA%8C/)

透明状态栏适配.主要适用场景为一个Activity多个tab在不同状态栏模式下切换.

Github Demo 链接: [StatusBarCompat](https://github.com/niorgai/StatusBarCompat" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank)

## **参考文章:**

1. [由沉浸式状态栏引发的血案](http://www.jianshu.com/p/140be70b84cd?utm_source=tuicool&utm_medium=referral" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank)
2. [Translucent System Bar 的最佳实践](http://www.jianshu.com/p/0acc12c29c1b" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank)
3. [该使用 fitsSystemWindows 了！](http://blog.chengyunfeng.com/?p=905" \l "ixzz43Roc61no" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank)
4. [Android 透明状态栏实现方案](http://gold.xitu.io/entry/570736071ea493005dc640ac" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank)
5. [更简单更全的material design状态栏](http://blog.csdn.net/u013260551/article/details/51150336" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank)

首先强调,对于状态栏的处理有两种不同的方式, 这里从[Translucent System Bar 的最佳实践](http://www.jianshu.com/p/0acc12c29c1b" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank)直接盗了两张图做对比~.

| **全屏( ContentView 可以进入状态栏)** | **非全屏 ( ContentView 与状态栏分离, 状态栏直接着色)** |
| --- | --- |
| [IMG_256](http://7sbqys.com1.z0.glb.clouddn.com/fullScreen.png) | [IMG_257](http://7sbqys.com1.z0.glb.clouddn.com/Toolbar.png) |

先定义几个名词:

1. ****全屏模式****: 左边图所示.
2. ****着色模式****: 右边图所示.
3. ****ContentView****: activity.findViewById(Window.ID\_ANDROID\_CONTENT) 获取的 View , 即 setContentView 方法所设置的 View, 实质为 FrameLayout.
4. ****ContentParent****: ContentView 的 parent , 实质为 LinearLayout.
5. ****ChildView****: ContentView 的第一个子 View ,即布局文件中的 root layout .

再介绍一下相关的函数:

1. fitsSystemWindows, 该属性可以设置是否为系统 View 预留出空间, 当设置为 true 时,会预留出状态栏的空间.
2. ContentView, 实质为 ContentFrameLayout, 但是重写了 dispatchFitSystemWindows 方法, 所以对其设置 fitsSystemWindows 无效.
3. ContentParent, 实质为 FitWindowsLinearLayout, 里面第一个 View 是 ViewStubCompat, 如果主题没有设置 title ,它就不会 inflate .第二个 View 就是 ContentView.
4. requestApplyInsets(), 当窗口(Window)大小改变了,通知 View 去消费窗口的改变.
5. FLAG\_TRANSLUCENT\_STATUS, 设置全屏的标志位, 此时界面可以延伸到状态栏.

## **5.0以上的处理:**

自5.0引入 Material Design ,状态栏对开发者更加直接,可以直接调用 setStatusBarColor 来设置状态栏的颜色.

****着色模式:****

通过查看 setStatusBarColor() 方法的文档,发现在调用该方法时需要设置以下属性:

1. 添加 FLAG\_DRAWS\_SYSTEM\_BAR\_BACKGROUNDS Flag(绘制系统栏).
2. 清除 FLAG\_TRANSLUCENT\_STATUS Flag(透明状态栏).
3. 调用 setStatusBarColor() 设置状态栏颜色.

****全屏模式:****

由于 5.0 以上为状态栏添加了一个阴影, 所以为全屏模式添加了是否隐藏状态栏阴影的方法.

隐藏阴影

* 1. 像着色模式一样添加 flag ,然后通过 setStatusBarColor() 设置颜色为透明.
  2. 通过 setSystemUiVisibility(View.SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE | View.SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN) 隐藏状态栏颜色.

显示阴影

* 1. 设置 FLAG\_TRANSLUCENT\_STATUS 来隐藏状态栏.
  2. 通过 setSystemUiVisibility(View.SYSTEM\_UI\_FLAG\_VISIBLE) 来恢复默认状态栏样式.

## **4.4-5.0的处理:**

4.4-5.0因为没有直接的 API 可以调用,需要自己兼容处理.参考了网上的解决方法及结合我自己遇到的坑,最后想出的解决办法如下:

****着色模式****

1. 向 DecorView 中添加一个 View, 高度为状态栏的高度(反射获取).
2. 将 ChildView 的 marginTop 加上状态栏的高度,以此来模拟 fitsSystemWindows.
3. 设置 ChildView 的 fitsSystemWindow 为 false, 不预留系统栏位置.
4. 为 DecorView 设置一个 tag, 防止重复添加 View.

这里与其他地方不同的是:

1. 向 ContentView 添加 View 在部分机型(华为)上没有效果.
2. 向 ContentParent 上添加 View 会有一条黑线.
3. 使用 marginTop 而不是 fitsSystemWindows 是因为无法在****不重启 Activity****的情况下切换 root layout 的fitsSystemWindows属性, 即直接设置不会生效, 所以用 marginTop 来模拟.

****全屏模式****

1. 设置 ChildView 的 fitsSystemWindow 为 false, 不预留系统栏位置.
2. 如果在 ChildView 的 marginTop 中添加了状态栏的高度, 则移除.
3. 设置 tag, 防止重复移除.

****CollaspingToolbarLayout****  
这个 support 包中的控件, 由于重写了 onApplySystemInsets() 方法, 按照我所理解的状态栏模式, 它在滑动时在两种模式中切换, 对此我的兼容方法就是让其处于 ****着色模式**** 下,在滑动时保持状态栏颜色不变. 当然有更好的解决办法, 但我这里为了方便调用(只需要传递 Activity 对象), 就用了比较简单的处理方法.

## **CollapsingToolbarLayout**

support 包中提供的 CollapsingToolbarLayout, 在使用的时候大概的布局是这样的(参考 [CheeseSquare](https://github.com/chrisbanes/cheesesquare" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank) ):

<CoordinatorLayout

android:fitsSystemWindows="true">

<AppBarLayout

android:fitsSystemWindows="true">

<CollapsingToolbarLayout

android:fitsSystemWindows="true">

<View

android:fitsSystemWindows="true"

app:layout\_collapseMode="parallax"/>

<ToolBar

app:layout\_collapseMode="pin"/>

</CollapsingToolbarLayout>

<AppBarLayout/>

<View

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"/>

</CoordinatorLayout>

在适配的时候主要遇到以下几个问题:

1. 如果按照我上面介绍的方式来说, 它在显示图片的时候是 全屏模式 , 在显示 Toolbar 的时候是 着色模式.
2. CollapsingToolbarLayout 的内部在 5.0 及以上的版本中通过 OnApplyWindowInsetsListener() 中获取的 Insets 对象, 在 Layout 的过程中将 View 向下偏移了, 所以它在 5.0 及以上系统中可以占据状态栏, 在 4.4 系统上则不能.
3. 仿照 [CheeseSquare](https://github.com/chrisbanes/cheesesquare" \t "http://niorgai.github.io/2016/03/20/Android-transulcent-status-bar/_blank) 这个库中的写法, CollapsingToolbarLayout 需要设置 FitsSystemWindow 为 true, 而我在前面兼容 全屏模式和 着色模式 的时候, 都是设置 FitsSystemWindow 为 false. 同时目前我没找到方法在不重启 Activity 的情况下切换FitsSystemWindow.

在参考了网上的文章后,最后决定通过以下方法来处理.

4.4版本:

* 1. 变为全屏模式,设置 View 的 fitsSystemWindow 为 false, 这一步可以使 CollapsingToolbarLayout 占据状态栏, Toolbar 也是.
  2. 改变 Toolbar 的高度, 加上状态栏的高度, 让 Toolbar 挡住状态栏位置, 同时为 Toolbar 添加 paddingTop , 这样就可以让 title 正常显示.
  3. 添加假的 StatusView 模拟状态栏颜色, 通过 AppBarLayout 的 OnOffsetChangedListener() 监听 AppBarLayout 的滑动, 使StatusView 跟随 CollapsingToolbarLayout 的显示隐藏.

5.0及以上版本:

* 1. 变为全屏模式,设置 View 的 fitsSystemWindow 为 false, 这一步可以使 CollapsingToolbarLayout 占据状态栏, Toolbar 也是.
  2. 改变 Toolbar 的高度, 加上状态栏的高度, 让 Toolbar 挡住状态栏位置, 同时为 Toolbar 添加 paddingTop , 这样就可以让 title 正常显示. 这都和 4.4 一样.
  3. 为 CollapsingToolbarLayout 设置 OnApplyWindowInsetsListener() 使其正常 Layout ,因为此时 CollapsingToolbarLayout 已经显示到状态栏了, 不需要在 Layout 过程中向下偏移, 但是此时collapsingToolbarLayout.setStatusBarScrimColor() 也就****无效****了.
  4. 通过 AppBarLayout 的 OnOffsetChangedListener() 监听 AppBarLayout 的滑动, 利用 setStatusBarColor() 设置状态栏颜色即可.

## **关于博客和库**

博客主要提供思路解析,因为博客的通知不及时,大家有问题有想法想交流时还请在 Github issues 页联系.