

所谓“沉浸式”状态栏

这里所说的沉浸式状态栏，就是指上面的效果，状态栏和 Toolbar 的颜色保持一致，融为一体的效果。

版本差异及解决方法

本文所用的示例使用的 style 风格是 NoActionBar 的，标题栏使用的是 Toolbar 控件，请知悉。

Android4.4

Android4.4 以前的版本，状态栏的颜色都是黑色的，而且无法修改；但一般 APP 的 Toolbar 都不会设置为黑色，于是，两者

有十分明显的颜色区分，各自占有不同的区域，填充不同的颜色。简单的说，Android4.4 以前的版本是无法做到

沉浸式的效果的（做系统开发的除外），所以如果想要统一风格的话，可以设置 APP 最小支持的版本为 4.4，如果

不行，就没办法了，只能 4.4 以前一个样式，4.4 及以后一个样式。

Android4.4 开始，新增了设置状态栏背景色透明的属性，新增的属性是这两个：

```
WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS
```

```
WindowManager.LayoutParams.FLAG_TRANSLUCENT_NAVIGATION
```

这两个属性分别设置状态栏和导航栏背景色为全透明。（这里只讨论状态栏，只设置上面一个属性就行，导航栏的类似）

实现方法很简单，在 Activity 初始化时，调用以下代码：

```
// Translucent status bar if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {  
  
    Window window = getWindow();  
  
    window.addFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);  
  
}
```

由于是代码设置，所以无法通过 xml 预览到想要的结果，一定要在 4.4 真机上运行才能看到效果。

到这里可能存在两个问题：

1. Toolbar 把状态栏的空间占用了，挤到一起。
2. 状态栏和 Toolbar 并没有完全融为一体，而是从上到下，由黑色渐变到 Toolbar 的颜色。

第一个现象是必现的，解决方法是在 Toolbar 的布局文件里加上一个属性：

```
android:fitsSystemWindows="true"
```

这个属性必须加在 Toolbar 或者 Toolbar 的父控件上，也就是，如果 Toolbar 直接写在 Activity 的布局文件里，则在 Toolbar 上

加这个属性，如果 Toolbar 是 include 到 Activity 的布局文件里，则可以加到 Toolbar 的父控件里；

```
<?xml version="1.0" encoding="utf-8"?><LinearLayout  
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical"
```

```
    tools:context="com.chengsy.immersive.MainActivity">
```

```
    <android.support.v7.widget.Toolbar
```

```
        android:id="@+id/toolbar"
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:background="@color/colorPrimary"
```

```
        android:fitsSystemWindows="true"
```

```
        app:title="@string/app_name"
```

```
        app:titleTextColor="#FFFFFF">
```

```
    </android.support.v7.widget.Toolbar>
```

```
    <TextView
```

```
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
```

```
        android:layout_centerInParent="true"
```

```
        android:text="Hello World!" />
```

```
</LinearLayout>
```

上面这种情况，如果写到 `LinearLayout` 中，就会出现这个问题，`StatusBar` 的背景色跟整个窗口的背景色一样（或者跟桌面颜色一样），而不是和 `Toolbar` 的颜色一样。效果如下：

第二个现象根据不同的系统厂商，效果不一样，也就是不同的手机厂商做的不一样，没办法，但是不是太影响沉浸式的效果。

效果如下：

总结一下，4.4 版本，需要设置两个地方来实现沉浸式效果：

-

设置状态栏透明

-
-

设置 `Toolbar` 的 `fitsSystemWindows` 属性为 `true`

-

Android5.0

先看看通过上面的设置，程序运行在 `Android5.0` 的设备上是什么效果：

可能的效果也有两个，一个是我们想要的效果，沉浸式，不再配图了，另一个是这样的：

这种效果是状态栏的颜色上面覆盖了一层半透明的颜色，不是全透明。当然，原因你懂得，不同厂商系统设计师的 `idea` 是不一样的，

但是这里说一下，谷歌官方的 `Mertial Design` 的设计规范是如上图所示，并不是沉浸式的效果，但国内的 APP 现在普遍比较喜欢沉浸式的效果。

那么如果要统一 5.0 的效果跟 4.4 的效果保持一致，全部都是沉浸式该怎么办？？？

这么办---

首先，`Android5.0` 开始，系统又新增了设置状态栏颜色值的属性和接口：

```
WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS
```

通过这个属性来设置设置状态栏的背景色，来实现两者融为一体的效果，代码实现如下：

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {

getWindow().addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);

    getWindow().clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);

    getWindow().setStatusBarColor(Color.TRANSPARENT); // SDK21
```

```
}
```

原理很简单，首先要清空之前设置的 `FLAG_TRANSLUCENT_STATUS` 属性，然后添加修改状态栏背景色的属性 `FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS`，最后给状态栏设置为全透明。

这里有坑，有的 5.x 的设备，如果调用这句代码

```
getWindow().clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
```

状态栏会变黑色，这时候需要去掉这句。但是去掉这句代码，在 6.0+ 系统上半透明的状态栏又出现了。所以处理方法是判断如果是某种特殊的系统，不调用上述代码。

Android6.0

对于 Android6.0 及以后的系统，对这方面的支持就已经很完善和统一了，通过 4.4 设置的 `FLAG_TRANSLUCENT_STATUS` 属性和

`fitsSystemWindows` 属性就可以达到想要的效果了，不需要做特殊处理，版本内暂时不存在机型之间的差异。

但是 6.0 也不是完全没新的东西（指的是状态栏这一块内容），6.0 新增了设置状态栏里内容色调的属性和接口，通过设置如下属性，可以把状态栏的文字色调由亮色改为暗色，亮色是白色，暗色是黑色：

```
getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);
```

看名字感觉跟我的描述不一样啊，其实，这个属性是把状态栏标记为亮色调，从而，系统会自动把状态栏的文字内容变为

暗色调，如果你的 APP 的 Toolbar 颜色是亮色的，再配上亮色的内容就会不明显，上面的代码可以解决这个问题，自动将状态栏的内容改为暗色调。

其他问题

-

输入框的兼容问题

-

如果想要页面底部的输入框可以被键盘顶起来，并且不影响页面的沉浸式效果，需要做两点：

- AndroidManifest.xml 文件配置键盘属性：`android:windowSoftInputMode="adjustResize"`

- Activity 页面根布局设置 `android:fitsSystemWindows="true"` 属性

布局代码如下：

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

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:fitsSystemWindows="true"

    android:orientation="vertical"

    tools:context="com.chengsy.immersive.MainActivity">

    <android.support.v7.widget.Toolbar

        android:id="@+id/toolbar"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:background="@color/colorPrimary"

        android:fitsSystemWindows="true"

        app:title="@string/app_name"

        app:titleTextColor="#FFFFFF">

    </android.support.v7.widget.Toolbar>

    <LinearLayout

        android:layout_width="match_parent"

        android:layout_height="match_parent"

        android:layout_below="@+id/toolbar"

        android:gravity="bottom">

        <EditText

            android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"

        android:text="Hello World!" />

</LinearLayout>

</RelativeLayout>
```

通过上面的处理，状态栏的颜色不对了，变成了白色或者桌面的背景色，问题又来了。解决方法的原理就是给状态栏填充上颜色，

但是，给状态栏修改颜色的属性和接口在 5.0 才出现，[4.4--5.0]版本的怎么办？用 `SystemBarTintManager`，Github 上的开源库，解决眼前的问题。使用很简单，引用这个库：

```
compile 'com.readystatesoftware.systembartint:systembartint:1.0.3'
```

代码中设置：

```
private SystemBarTintManager tintManager;

tintManager = new SystemBarTintManager(this);

tintManager.setStatusBarTintColor(getResources().getColor(R.color.colorPrimary));

tintManager.setStatusBarTintEnabled(true);
```

-

其他第三方控件的兼容，如 `ActionMode` 等

-

有用到这个的自己去查、去尝试吧，这个控件好久不用了，也不知道什么效果，可以参考这篇文章里的解决方法：

[沉浸式状态栏实现及遇到的坑](#)

-

系统厂商的兼容，如 MIUI 系统等

-

这里也介绍了 MIUI 系统的适配问题，供参考：

[沉浸式状态栏实现及遇到的坑](#)

总结

由于 Android 系统的开放性，以及系统厂商的不统一，导致 Android 系统的适配成为了另程序员头疼的一大问题，沉浸式的效果同样

不好做到完全统一样式，只能尽力而为之。通过上面的操作，基本可以保证大多数机型和系统的沉浸式效果，但仍然有个别无法适配的系统或机型，这里也无法一一列举所有的情形，需要程序员们有针对性的设计代码，解决问题。

这里给出自己总结的一个方法，可以在 BaseActivity 中调用：

首先，Toolbar 要设置 fitsSystemWindows 属性，如果页面包含 EditText，需同时在页面根布局添加 fitsSystemWindows 属性；

其次，引用 SystemBarTintManager 库提供支持；

最后，调用如下代码：

```
private SystemBarTintManager tintManager;protected void initWindow() {  
    // 4.4 及以上版本设置状态栏透明  
    Window window = getWindow();  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {  
        // Translucent status bar  
        window.addFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);  
    }  
  
    // 解决4.4-5.0 版本之间，页面包含EditText 无法适配的问题  
    {  
        // create our manager instance after the content view is set  
        mTintManager = new SystemBarTintManager(this);  
        // enable status bar tint  
        mTintManager.setStatusBarTintEnabled(true);  
        // enable navigation bar tint  
        mTintManager.setNavigationBarTintEnabled(true);  
  
        // 自定义状态栏的颜色  
        mTintManager.setStatusBarTintColor(getResources().getColor(R.color.colorPrimary));  
    }  
  
    // 解决[5.0-5.1.1]版本状态栏没有全透明的系统适配问题  
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {  
        // 解决部分 5.x 系统使用状态栏透明属性后状态栏变黑色，不使用这句代码，在 6.0 设备上又出现半透明状态栏  
        // 需要特殊处理  
        window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);  
  
        window.getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN  
            | View.SYSTEM_UI_FLAG_LAYOUT_STABLE);  
        window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);  
        window.setStatusBarColor(Color.TRANSPARENT);  
    }  
  
    // 把状态栏标记为浅色，然后状态栏的字体颜色自动转换为深色。
```

```
// if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {  
//  
getWindow().getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LIGHT_STATUS_BAR);  
// }
```