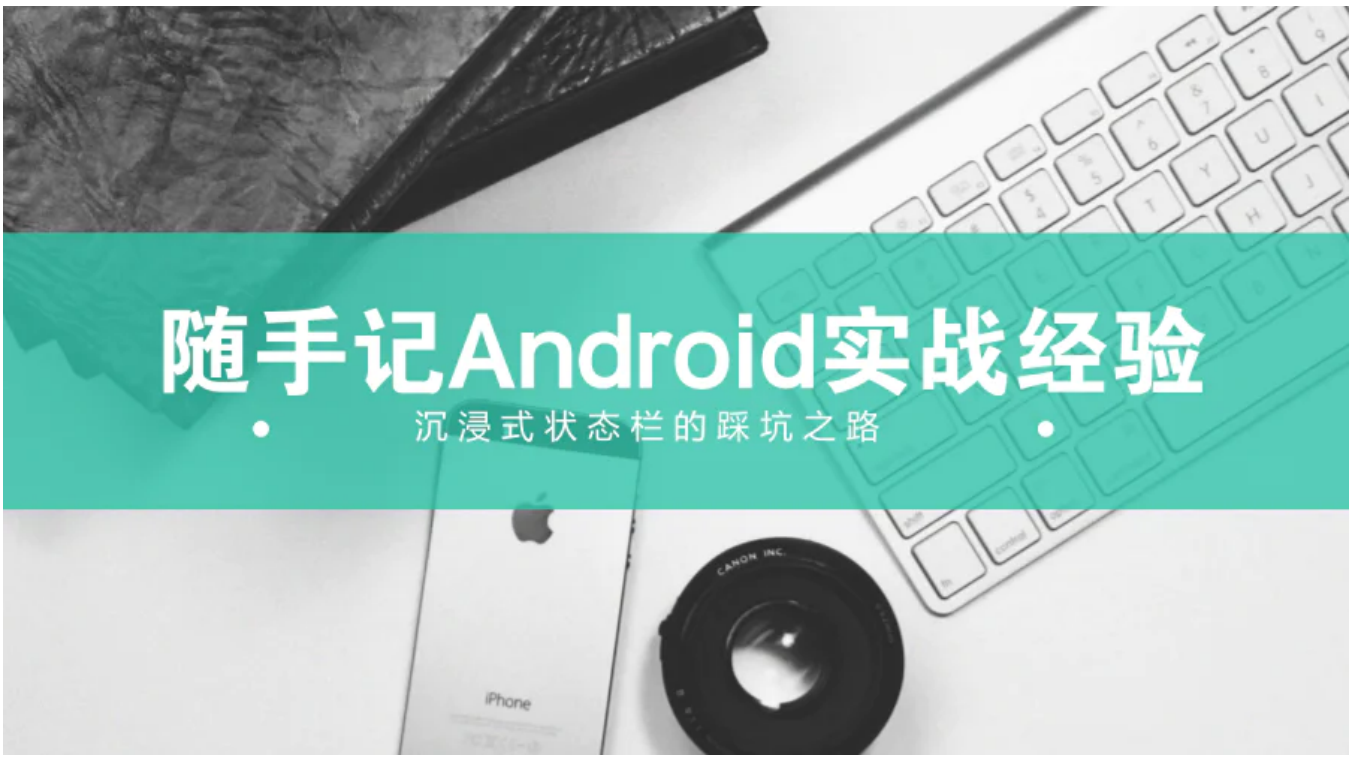
随手记技术团队 Lv3
2017年12月05日 阅读 15433

关注



随手记Android沉浸式状态栏的踩坑之路

> 欢迎关注微信公众号「随手记技术团队」，查看更多随手记团队的技术文章。
> 本文作者：刘玲
> 原文链接：mp.weixin.qq.com/s/d6D_rYmzl...

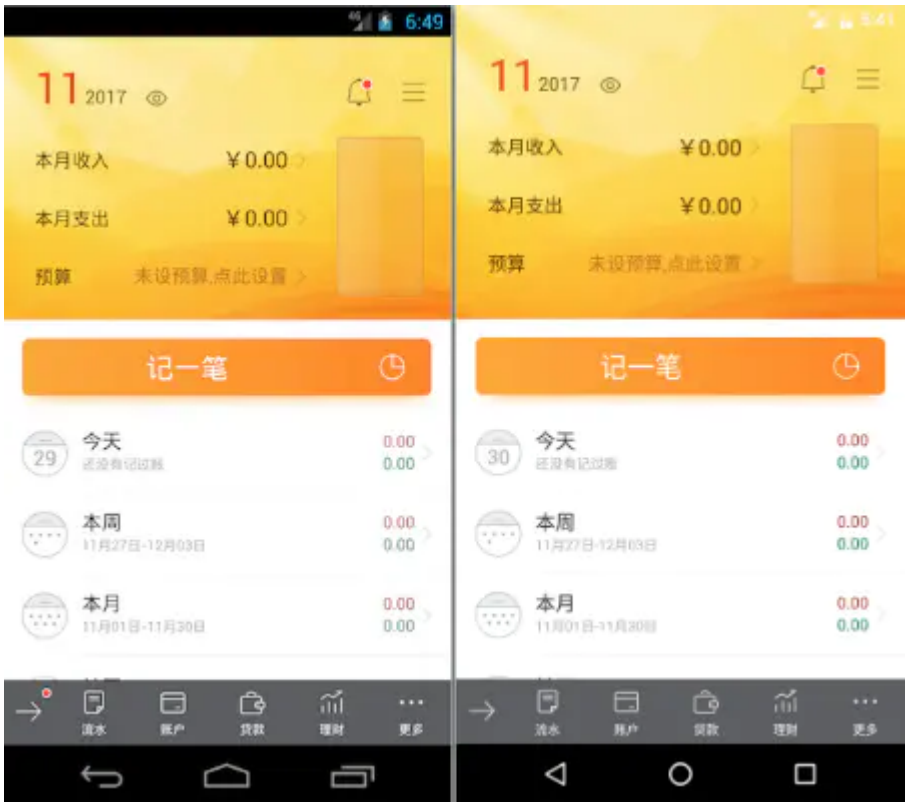
1-前言

关于“沉浸式状态栏”这种叫法，有的朋友可能会觉得不妥。但是目前网上大部分讲到“沉浸式状态栏”基本都是指“透明状态栏”，所以这里就不讨论其对错了（其实有时候错的多了，也就成了对的了），大家知道是说的“透明状态栏”就行了，下文都是称这种效果为“沉浸式状态栏”。

在Android 4.4之前，所有应用都是无法设置状态栏的背景颜色的，都是跟着系统来的（黑色背景状态

首页 探索掘金

Android客户端也在年初的时候也支持了沉浸式状态栏。在实现沉浸式状态栏效果的过程中踩了不少的坑，特此记录下来。 下图为随手记Android客户端设置沉浸式状态栏前后的效果对比图：



对比两种效果，很明显下面设置了沉浸式状态栏的看上去更协调、更美观一点。

2-如何实现沉浸式状态栏

2.1-Android 4.4以上实现方式

由于沉浸式状态栏设置是在Android 4.4之后才提供的，所以我们需要对Android 4.4以上的系统做适配。Android 4.4有两种方式可以实现沉浸式状态栏，一种是在资源文件中设置，一种是在代码中设置。

2.1.1-资源文件中设置沉浸式状态栏

首先，我们要修改values/styles.xml，在里面添加一个空的style，继承自BaseTheme。

复制代码

```
<resources>
    <!-- Base application theme. -->
    <style name="BaseTheme" parent="Theme.AppCompat.Light.NoActionBar">
```

首页 探索掘金

```
        <item name="colorAccent">@color/colorAccent</item>
    </style>

    <style name="AppTheme" parent="BaseTheme" />
</resources>
```

然后在values-v19目录下的styles.xml文件（如果项目中没有就新建一个，在4.4以上的系统就会读取该目录下的资源文件）添加如下代码：

复制代码

```
<resources>
    <style name="AppTheme" parent="BaseTheme">
        <item name="android:windowTranslucentStatus">true</item>
    </style>
</resources>
```

然后将App的主题设置为AppTheme即可。 注：android:windowTranslucentStatus这个属性是v19开始引入的。

2.1.2-在代码中设置

在代码中实现更为方便一点，我们只需要在BaseActivity中添加一个FLAG_TRANSLUCENT_STATUS的flag即可。

复制代码

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
    getWindow().addFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
}
```

通过上述两种方法设置之后，效果图如下：

我们会发现，仅仅通过上述设置Toolbar会顶到状态栏里面去。通常大家会想到使用fitsSystemWindows属性来解决此问题。

> fitSystemWindows官方描述：Boolean internal attribute to adjust view layout based on system windows such as the status bar. If true, adjusts the padding of this view to leave space for the system windows. Will only take effect if this view is in a non-embedded activity. 简单描述：这个属性的作用是让view可以根据系统窗口(如status bar)来调整自己的布

首页 探索掘金

我们试着给Toolbar设置一下fitsSystemWindows属性为true。布局代码如下：

复制代码

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <android.support.v7.widget.Toolbar
        android:id="@+id/my_toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"
        android:minHeight="?attr/actionBarSize"
        android:fitsSystemWindows="true"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar" />

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:padding="16dp">

        <RelativeLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="#e0e0e0"
            android:layout_gravity="bottom">
            <EditText
                android:layout_width="match_parent"
                android:layout_height="40dp"
                android:fitsSystemWindows="true"
                android:background="@drawable/edit_text_rect_bg" />
        </RelativeLayout>
    </FrameLayout>
</LinearLayout>
```

上面代码在Android 4.4和Android 5.0+上面对比效果图如下：

由上面对比图我们可以看出来，在Android 4.4上面状态栏是全透明的，而在Android 5.0+上面状

首页 探索掘金

注：有些4.4的系统上面状态栏并不是全透明的，而是渐变的。

2.2-Android 5.0以上实现方式

上面已经实现了沉浸式状态栏的效果了，但是如果运行在Android 5.0以上的机器上面，会发现大部分手机会出现状态栏是半透明的。

也有些App在Android 5.0以上就是这种状态栏半透明的效果，比如QQ。但是有些产品和设计就是想统一风格，全部都实现全透明的状态栏。那怎么办呢？Android自5.0起，又为我们提供了设置状态栏颜色的API，我们可以自己设置状态栏的颜色。

复制代码

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

}
```

添加上述代码后再在Android 5.0+上运行看效果，状态栏已经变成全透明了，和上图Android 4.4效果一样的，这里就不再附图了。

2.3-Android 6.0以上设置状态栏字体颜色

大部分手机默认状态栏字体颜色是白色的，如果Toolbar或者界面头部的颜色较浅，那么状态栏上白色的字看不怎么清楚。 Android 6.0以后，我们可以使用代码将状态栏字体的颜色设置为黑色了，代码如下：

复制代码

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

3-踩过的坑

本以为上面基本已经完美实现了沉浸式状态栏了，没想到测试的时候还是发现了一系列的坑。

3.1-软键盘弹出时Toolbar被顶上去了

首页 探索掘金 登

如果在界面中有EditText或者其他输入框的话，会发现当软件盘弹出的时候Toolbar里面的内容都被顶上去了，如下图所示：

这是为什么呢？经研究发现原来是fitsSystemWindows属性搞的鬼。哪个View设置了fitsSystemWindows=true，这个View就会被软件盘顶上去。所以说，fitsSystemWindows不能乱用，会有意想不到的坑。 那能不能不用fitsSystemWindows呢？当然可以。前面也说了，fitsSystemWindows=true的作用是给View增加值为状态栏高度的padding，那我们何不自己手动给Toolbar添加padding呢？ 我们去掉Toolbar上的fitsSystemWindows属性，并设置一下Toolbar的padding，代码如下：

复制代码

```
protected void setStatusBarPaddingAndHeight(View toolBar) {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
        if (toolBar != null) {
            int statusBarHeight = getSystemBarHeight(this);
            toolBar.setPadding(toolBar.getPaddingLeft(), statusBarHeight, toolBar.getPaddin
                    toolBar.getPaddingBottom());
            toolBar.getLayoutParams().height = statusBarHeight +
                    (int)TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DIP, 45, getReso
        }
    }
}
```

去掉Toolbar的fitsSystemWindows属性，并加上上面的代码，软键盘弹出时Toolbar正常了。 目前随手记Android项目中就是使用代码添加padding的方式替代fitsSystemWindows属性的。

3.2-软键盘弹出时EditText等输入框会被软件盘覆盖掉

上面软件盘将Toolbar顶上去的示例图中，我们还会发现一个问题，就是软键盘弹出时EditText并没有跟着弹出来而是被软键盘覆盖掉了。

上面说Toolbar加了fitsSystemWindows属性之后会被软键盘顶上去，那么我们给输入框加一个fitsSystemWindows属性是否刚好就能解决输入框被覆盖的问题呢？果断试一下！

试了之后发现，果然可以，但是输入框的高度变了，其实是输入框的padding增加了状态栏的高度。很显然，这并不是一个很好的解决方式。 后来在stackoverflow上找到了一个解决方法：解决FLAG_TRANSLUCENT_STATUS导致输入框被软键盘覆盖的解决方案

首页 探索掘金

复制代码

```
public class AndroidBug5497Workaround {

    public static void assistActivity(View content) {
        new AndroidBug5497Workaround(content);
    }

    private View mChildOfContent;
    private int usableHeightPrevious;
    private ViewGroup.LayoutParams frameLayoutParams;

    private AndroidBug5497Workaround(View content) {
        if (content != null) {
            mChildOfContent = content;
            mChildOfContent.getViewTreeObserver().addOnGlobalLayoutListener(new ViewTreeObse
                public void onGlobalLayout() {
                    possiblyResizeChildOfContent();
                }
            });
            frameLayoutParams = mChildOfContent.getLayoutParams();
        }
    }

    private void possiblyResizeChildOfContent() {
        int usableHeightNow = computeUsableHeight();
        if (usableHeightNow != usableHeightPrevious) {
            //如果两次高度不一致
            //将计算的可视高度设置成视图的高度
            frameLayoutParams.height = usableHeightNow;
            mChildOfContent.requestLayout();//请求重新布局
            usableHeightPrevious = usableHeightNow;
        }
    }

    private int computeUsableHeight() {
        //计算视图可视高度
        Rect r = new Rect();
        mChildOfContent.getWindowVisibleDisplayFrame(r);
        return r.bottom;
    }

}
```

添加上面的类，然后在Activity的onCreate方法中的setContentView后面加上如下代码：

复制代码

```
AndroidBug5497Workaround.assistActivity(findViewById(android.R.id.content));
```

首页 探索掘金 登

然后运行，输入框能够正常被顶上去，而且输入框的布局又没有受到影响。

该方案的原理是，给界面的根布局设置一个监听器，当界面大小有变化的时候，如键盘弹出的时候，重新设置一下根布局的高度，再调用requestLayout对界面进行重绘。

目前随手记Android就是使用这个方案，截止到目前也没有发现这种方案会带来其他什么问题。

3.3-华为EMUI3.1上的坑

将上面的沉浸式代码放在EMUI3.1系统的手机（如华为荣耀7）上面跑，会发现根本没有沉浸式效果，状态栏是透明的，显示的是桌面上的颜色，如下图：

经验证，原来是EMUI3.1系统的原因，很多App也是在EMUI3.0上有沉浸式的效果，到了EMUI3.1却没有效果了。在EMUI3.1没有沉浸式效果如果和4.4以前一样是黑的也就算了，这样透明的显示桌面颜色实在难看。 后来发现去掉下面这句代码，可以让其有沉浸式的效果。

复制代码

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

效果如下：

不过它的状态栏不是全透明的，而是像某些4.4的系统一样是渐变的，不过总比显示桌面颜色的效果好。 这里我们加一个判断，判断如果不是EMUI3.1的系统，才调用clearFlags清除掉FLAG_TRANSLUCENT_STATUS。 具体代码如下：

复制代码

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
    Window window = getWindow();
    window.addFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
        // 因为EMUI3.1系统与这种沉浸式方案API有点冲突，会没有沉浸式效果。
        // 所以这里加了判断，EMUI3.1系统不清除FLAG_TRANSLUCENT_STATUS
        if (!isEMUI3_1()) {
            window.clearFlags(WindowManager.LayoutParams.FLAG_TRANSLUCENT_STATUS);
        }
        window.getDecorView().setSystemUiVisibility(View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN
```

首页 探索掘金 登...

```
            window.setStatusBarColor(Color.TRANSPARENT);
        }
    }

    public static boolean isEMUI3_1() {
        if ("EmotionUI_3.1".equals(getEmuiVersion())) {
            return true;
        }
        return false;
    }

    private static String getEmuiVersion(){
        Class<?> classType = null;
        try {
            classType = Class.forName("android.os.SystemProperties");
            Method getMethod = classType.getDeclaredMethod("get", String.class);
            return (String)getMethod.invoke(classType, "ro.build.version.emui");
        } catch (Exception e){
        }
        return "";
    }
```

3.4-CoordinatorLayout+AppBarLayout滚动隐藏导航栏遇到沉浸式状态栏的坑

这个坑主要是在做理财头条需求的时候碰到的。

> 需求背景：头条功能需要实现二级TabLayout导航，第一级是Toolbar（头条、产品和发现），第二级是是头条里面各个栏目切换的TabLayout。需要实现的效果是，在头条Fragment中，滑动帖子列表可以隐藏和显示一级导航Toolbar。一级导航Toolbar显示的时候，左右滑动是切换一级导航的Tab（即头条、发现和产品）。当在头条Fragment中上滑滚动帖子列表隐藏一级导航Toolbar后，左右滑动是切换二级导航的tab（即头条各个栏目）。效果见下图。

滚动列表隐藏和显示Toolbar，首先肯定是想到CoordinatorLayout+AppBarLayout。基于项目中已实现的沉浸式效果，添加修改Activity中的布局：

复制代码

```
<android.support.design.widget.CoordinatorLayout
    android:id="@+id/coordinator_layout"
    android:layout_height="match_parent"
    android:layout_width="match_parent">
```

首页 探索掘金 登...

```
        android:layout_height="wrap_content"
        app:elevation="0dp">
        <android.support.v7.widget.Toolbar
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            app:contentInsetStart="0dip"
            app:contentInsetLeft="0dip"
            app:contentInsetEnd="0dip"
            app:layout_scrollFlags="scroll|enterAlways">
            ...部分代码省略...
           <android.support.design.widget.TabLayout
               android:id="@+id/tab_layout"
               android:layout_width="wrap_content"
               android:layout_height="match_parent"
               android:layout_centerHorizontal="true"
               app:tabBackground="@null"
               app:tabIndicatorColor="@color/tab_text_selected_color"
               app:tabIndicatorHeight="2dip"
               app:tabMode="fixed"
               app:tabGravity="fill"
               app:tabPaddingStart="14dp"
               app:tabPaddingEnd="14dp"
               app:tabTextAppearance="@style/FinanceTabTextAppearance"
               app:tabSelectedTextColor="@color/tab_text_selected_color"
               app:tabTextColor="@color/tab_text_unselected_color" />
        </android.support.v7.widget.Toolbar>
    </android.support.design.widget.AppBarLayout>
    <android.support.v4.view.ViewPager
        android:id="@+id/pager"
        android:layout_width="match_parent"
        android:paddingBottom="50dp"
        android:layout_height="match_parent"
        android:overScrollMode="never"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"/>
</android.support.design.widget.CoordinatorLayout>
```

布局是在Toolbar中添加一个TabLayout作为一级导航的tab。然后使用一个ViewPager，给该ViewPager添加了三个Fragment，分别是头条、产品和发现的Fragment。其中，头条Fragment中又嵌套了TabLayout和ViewPager。 基于沉浸式的实现方案，在代码中给AppBarLayout添加一个状态栏高度的padding。本以为可以大功告成了，结果发现运行之后，在上滑隐藏AppBarLayout之后再下拉，会超出下拉范围，也就是下拉的时候会多出一条状态栏高度的空白，效果如下图顶部：

经过不断尝试和探索，发现给Activity添加如下flag即可：

首页 探索掘金 登...

复制代码

```
this.getWindow().addFlags(WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS);
```

嗯，不错，滑动问题解决了。但心里总是不安，总感觉有坑。后面发现确实有坑，添加了这个flag后，部分带虚拟按键的华为手机出现虚拟按键挡住底部布局的问题，经验证只有EMUI3.1才有这个问题（又是EMUI3.1，已无力吐槽）。 最后百般周折，终于找到有效解决CoordinatorLayout+AppBarLayout并给AppBarLayout设置paddingtop之后的滑动问题的方法了。

复制代码

```
this.getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);
this.getWindow().addFlags(WindowManager.LayoutParams.FLAG_FORCE_NOT_FULLSCREEN);
```

本以为上面解决方案已经完美没有任何问题了，没想到还是有坑。不久后测试发现一个现网问题：当WebView中的输入框获取焦点软键盘弹出后，退出界面时底部布局出现软键盘大小的黑块。如下图所示:

经排查，此问题就是由于上面那段代码引起的。 没办法，只能去掉上面那段代码，寻找另外的解决方案来处理CoordinatorLayout+AppBarLayout并给AppBarLayout设置paddingtop的滑动问题了。 后来在发现在Activity的onCreate方法中加上下面一段代码就可以完美解决这个问题。

复制代码

```
if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP) {
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.root_container_layout), new
        @Override
       public WindowInsetsCompat onApplyWindowInsets(View v, WindowInsetsCompat insets) {
            return insets.consumeSystemWindowInsets();
       }
    });
}
```

4-总结

上面就是随手记Android项目中沉浸式状态栏实现过程中遇到的坑以及解决方案。最终随手记Android实现状态栏效果后在不同机型上面效果图如下:

经过沉浸式状态栏的开发，发现几个容易踩的坑需要注意： 1.fitsSystemWindows=true要慎用，很多

首页 探索掘金 登

2.WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS不要用，会导致EMUI3.1的系统下面虚拟按键挡住布局；

5-参考文档

stackoverflow.com/questions/7...

文章分类 Android 文章标签 Java

随手记技术团队 Lv3
技术研发管理 @ 随手记
获得点赞 1,055 · 获得阅读 45,354

关注

安装掘金浏览器插件
打开新标签页发现好内容，掘金、GitHub、Dribbble、ProductHunt 等站点内容轻松获取。快来安装掘金浏览器插件获取高质量内容吧！

输入评论...

todoit Lv3 技术总监 @ 创业公司
如果可以给个 demo 该多好
3年前 1 回复

安琪 Android开发 @ 大连华...
能判断不同手机的型号吗?
3年前 回复

875430315
AndroidBug5497Workaround 这个把键盘顶上去 会看到上个界面的视图,
3年前 回复

某裸睡的鱼 程序员
感谢

首页 探索掘金

getglory

状态栏弄得快吐了 特别是弄视频直播横竖屏的时候

3年前 回复

Skylake01 Android兼职前端...

同样，有段时间我也弄这个玩意，状态栏其实还好，虚拟导航栏更麻烦

3年前

darksider Lv1 Android

很详细，很清晰

3年前 回复

相关推荐

小傅哥 2小时前 后端 Java GitHub

调研字节码插桩技术，用于系统监控设计和实现

一套线上系统是否稳定运行，取决于它的运行健康度，而这包括；调用量、可用率、影响时...

551 14 2

敖丙 3天前 Java 面试

B站五面面经（附过程、答案）

上周午休我刷手机的时候看到26群在那疯狂刷恭喜，我以为发生了什么，原来是晨曦进了B...

9830 166 39

秦二爷 2天前 后端 Java

起底JVM内存管理及性能调优【80+页Keynote私享】

图片较多，可能会引起你的不适，文末有PDF下载方式。 本文出自二爷箱底下的一份陈年文...

3274 85 30

lcomplete 21小时前 后端 Java

[Java 开发实战]💻高级工程师的自我修养之单元测试（一）：DAO 层测试

通过本篇文章的学习，你将掌握 JUnit5 + Testcontainers + Liquibase + h2 进行 DAO层单元测试的方法。

971 8 4

老马啸西风 11小时前 后端 Java Spring Boot

springboot 实现拦截器的 3 种方式介绍及异步执行的思考

程序猿DD 23小时前 Java

Spring官方发布新成员：Spring GraphQL

近日，在GraphQL Java诞生6周年的时候，Spring社区通过博客宣布正式创建全新项目：Spring GraphQL，同时还...

1149 5 2

一起随缘 1天前 Java 后端

手动实现第三方jar包修改并重新打包

前言 开发过程中，项目中总会引入一些第三方依赖包，以便通过直接调用jar包中的方法来...

690 14 4

青Cheng序员石头 1天前 面试 Java

面时莫慌 | 你好，谈谈对Synchronized的理解？（五）

上一小节讲到了两个线程竞争锁资源，未获取到锁资源的线程在自旋策略范围内未获取到锁...

443 15 5

秦二爷 1天前 后端 Java

并发王者课-铂金9：互通有无-Exchanger如何完成线程间的数据交换

欢迎来到《并发王者课》，本文是该系列文章中的第22篇，铂金中的第9篇。在本文中，将...

974 7 评论

MacroZheng 3天前 Java 后端 GitHub

全新一代API网关，带可视化管理，文档贼友好！

推荐一款功能强大的API网关`apisix`，自带可视化管理功能，多达三十种插件支持，希望对...

3619 33 4