# Android

## Android Layout字母自动大写，avd上也是：

android:textAllCaps="false"

## Android 让app状态栏透明

代码中：

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*);  
}

Style中：

要根据api不同进行区别配置

<**item** name="android:windowTranslucentStatus">true</**item**>  
<**item** name="android:windowTranslucentNavigation">true</**item**>  
<**item** name="android:statusBarColor">#00000000</**item**>

Xml中

android:fitsSystemWindows="true"

## android Transaction（事务）

事务（Transaction）是并发控制的单位，是用户定义的一个操作序列。这些操作要么都做，要么都不做，是一个不可分割的工作单位。通过事务，SQL Server能将逻辑相关的一组操作绑定在一起，以便服务器保持数据的完整性

当对多个表进行更新的时候，某条执行失败。为了保持数据的完整性，需要使用事务回滚。

Android –AlertDialog复用自定义添加的view内控件无法获取

((EditText)(dialog.getWindow().findViewById(R.id.*dialpg\_editText*)))

## Android 给虚拟Android设备上传文件

D:\Genymotion\Genymotion\tools>adb push C:\Users\ai\Desktop\WeX5One.apk /sdcard/WeX5One.apk

## Android ARM转X86

An error occured while deploying the file.   
This probably means that the app contains ARM native code and your Genymotion device cannot run ARM instructions. You should either build your native code to x86 or install an ARM translation tool in your device.

部署文件时出错。   
这可能意味着应用程序包含本地ARM代码和你的genymotion设备无法运行ARM指令。你可以建立你的原生代码的x86或在您的设备上安装一个臂的翻译工具。

解决方法

1.用Android Studio 创建一个ARM的虚拟机；（当然这个不是你想要的）

2.下载Genymotion-ARM-Translation-Librarities工具转换包；下载路径：https://pan.baidu.com/s/1kUAftyR

将下载号的工具包直接拖拽到Genymotion中，然后提示重启模拟器

已下载于：D:\point\_to\_desktop\android\other

## Android 波纹效果（Ripple）

在Android L5.0中加入了触摸反馈动画。

其中最明显，最具代表性的就是波纹动画，比如当点击按钮时会从点击的位置产生类似于波纹的扩散效果。

**波纹效果（Ripple）：**

当你使用了Material主题后，波纹动画会自动应用在所有的控件上，我们当然可以来设置其属性来调整到我们需要的效果。

可以通过如下代码设置波纹的背景：

**android:background="?android:attr/selectableItemBackground"**波纹有边界

**android:background="?android:attr/selectableItemBackgroundBorderless"**波纹超出边界

**设置颜色**

我们也可以通过设置xml属性来调节动画颜色，从而可以适应不同的主题：

**android:colorControlHighlight**：设置波纹颜色

**android:colorAccent**：设置checkbox等控件的选中颜色

为view添加波纹效果：

   在布局管理器的属性中添加：

android:clickable="true"

android:background="?android:attr/selectableItemBackground"

为Textview添加波纹效果：

在属性中添加：

android:clickable="true"

android:background="?android:attr/selectableItemBackgroundBorderless"

## Android 在子线程中为什么不能用Toast；

因为Toast在创建的时候会依赖于一个Handler，并且一个Handler是需要有一个Looper才能够创建，而普通的线程是不会自动去创建一个Looper对象，比如说在某个Activity中能new一个Handler是因为Android系统在启动一个Activity的时候会默认的创建一个Looper对象，因此不能够在子线程中显示Toast，你可以在开启的子线程中执行Looper.prepare()来构建一个Looper，然后在显示Toast，但是不要忘记执行Looper.loop()来加载这个Looper，当然，也可以使用主线程的Looper，获取主线程的Looper的方法是Looper.getMainLooper()；同时需要注意的是，同样的不能在子线程中去更新UI界面，因为Toast是相对独立于UI界面的，就好比应用虽然crash掉了，并且已经返回到home界面，但是Toast依然会在hone界面显示出来。

注脚1可能没说清楚，昨天写的时候搞清了现象但没找出原因，刚刚找到了原因：——局部内部类的对象可以访问同一个方法中被定义为final的局部变量。定义为final后，编译程序的实现方法：将所有的局部内部类对象要访问的final型局部变量，都拷贝成为该内部类对象中的一个数据成员。这样，即使栈中局部变量（含final）已死亡，但由于拷贝到内部类中的数据成员任是final,其值永不变，因而局部内部类对象在方法内局部变量死亡后，照样可以访问final型局部变量。就是代码里的下面这个位置：

MainActivity.this.runOnUiThread(new Runnable(){.............}) //201行开始

new Runnable(){.............}是一个匿名内部类（可以理解为Runnable的子类），这里作为runOnUiThread方法的实参（局部内部类的对象）。

参考的原文在这里：<http://blog.csdn.net/cpp597455873/article/details/46287853>

## Android 真机或虚拟机上安装失败 Build -ReBuilde

Installation failed with message Failed to finalize session : INSTALL\_FAILED\_INVALID\_APK: /data/app/vmdl1841863905.tmp/11\_app-debug signatures are inconsistent. It is possible that this issue is resolved by uninstalling an existing version of the apk if it is present, and then re-installing.

WARNING: Uninstalling will remove the application data!

Do you want to uninstall the existing application?

<http://stackoverflow.com/questions/42668595/installation-failed-to-finalize-session-signatures-are-inconsistent-android>

## Android 修改状态栏字体颜色

<http://blog.csdn.net/java04/article/details/51542583>

Android 6.0开始，谷歌官方提供了支持，在style属性中配置android:windowLightStatusBar   
即可， 设置为true时，当statusbar的背景颜色为淡色时，statusbar的文字颜色会变成灰色，为false时同理。

<style name="statusBarStyle" parent="@android:style/Theme.DeviceDefault.Light">

<item name="android:statusBarColor">@color/status\_bar\_color</item>

<item name="android:windowLightStatusBar">false</item>

</style>

## Android snackbar 无动画

手机 – 设置 – 辅助功能 – 服务 – fooView（关闭）

## Android Studio 导入jar包后编译成功但代码提示无法使用

导入后放到 src/main/libs