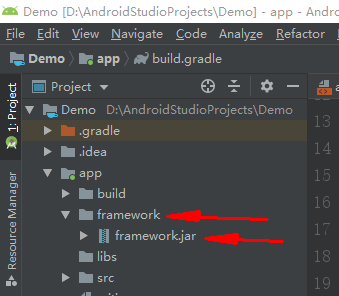
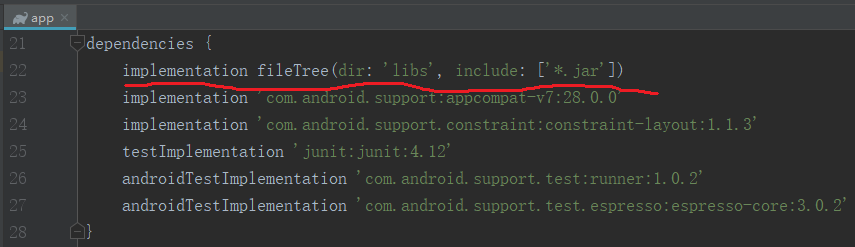


我们在开发的时候会调用到一些系统的类，这些类的class保存在android.jar中，而Google公司在给我们编译android.jar的时候并不会把带有@hide注解的类/方法/属性编译到android.jar中，所以我们访问不到这些标有@hide注解的类/方法/属性，但是这些类/方法/属性是真实存在于Android手机系统中，所以解决方案就是获取到系统中的这个完整的jar放在我们的项目中，让编译器先从这个jar中查找类即可。获取完整jar在编译android系统的时候就可以获取到这个jar了。把这个jar重命名为framework.jar，保存到项目中，比如保存在项目中的framework目录下（这个目录名字可以随便起的），如下：

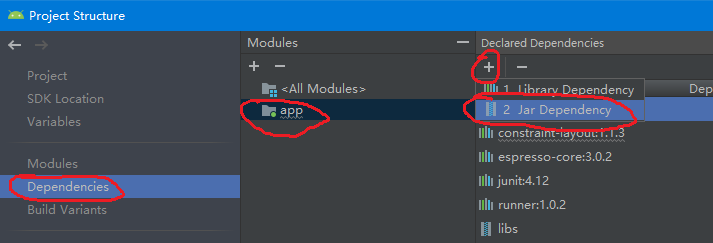


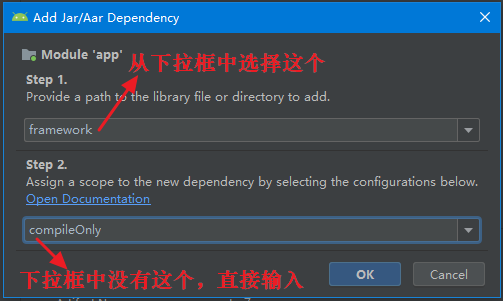
这个jar我们只需要在写代码的时候和编译代码的时候用到，打包apk时并不需要打包到apk中，因为这个jar的所有类在手机的Android系统中已经存在了，所以不能把这个jar放到libs目录中，根据gradle中的配置，放到libs目录中的jar是会被打包到apk的，如下：



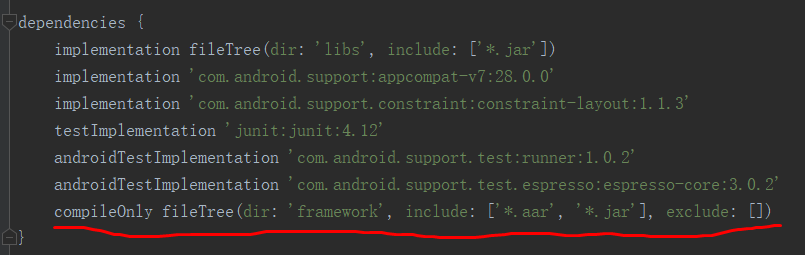
那如何设置为只编译，不打包呢？使用compileOnly，方法如下：

打开项目设置，快捷键：Ctrl + Shift + Alt + S



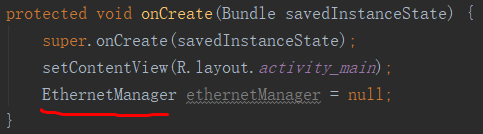


点击“OK”按钮即可，此时可查看模块的build.gradle文件中多了一行代码，如下：

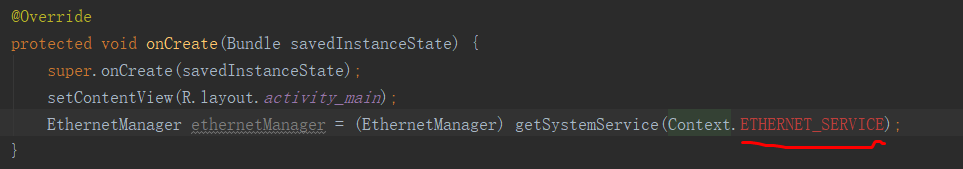


comileOnly的功能就是让其指定的jar只在编译时使用，打包时不会被打进apk中。

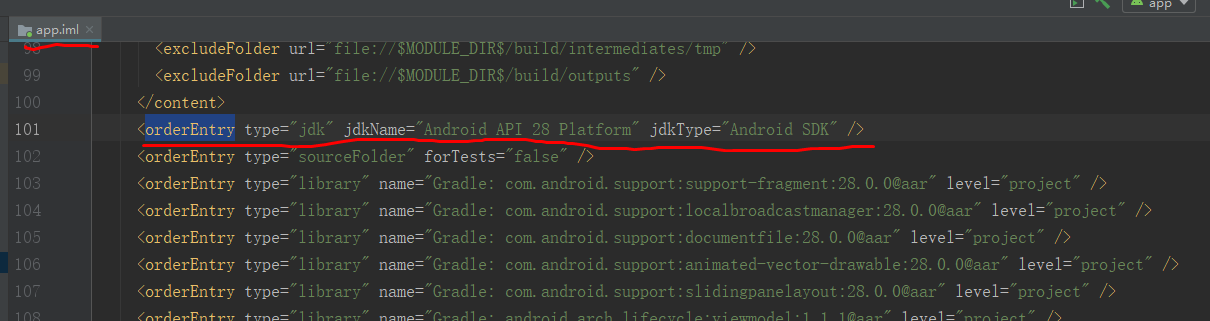
以后我们如果还有其他的jar或者aar也有相同的需求（只编译不打包），则把它放入到framework文件夹下即可。此时我们可以试试访问一个被隐藏的类EthernetManager，EthernetManager用于控制Android系统插入网线时使用动态还是静态的方式分配ip（什么？手机怎么能插入网线？可以的，市面上有Type-C转USB网卡，还有机顶盒一般都有网线接口，机顶盒也是Android系统的），访问一下这个类，如下：



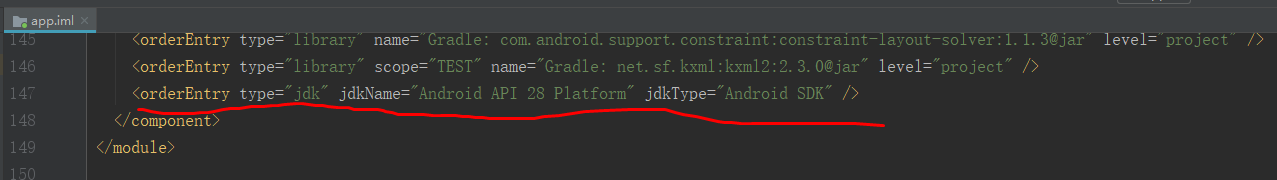
好像已经大功告成了，别急，还有问题呢。如果我们访问的只是隐藏的方法或属性，而类并不是隐藏的，这时就会有问题，如下：



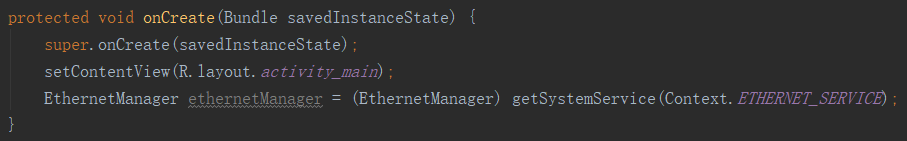
Context中的ETHERNET\_SERVICE属性是隐藏的，AndroidStudio在找一个类的时候先从哪里找呢？我们的应用可能依赖了很多库或jar，先从哪个开始找呢？这取决于模块的.iml文件中的orderEntry标签顺序，如下：



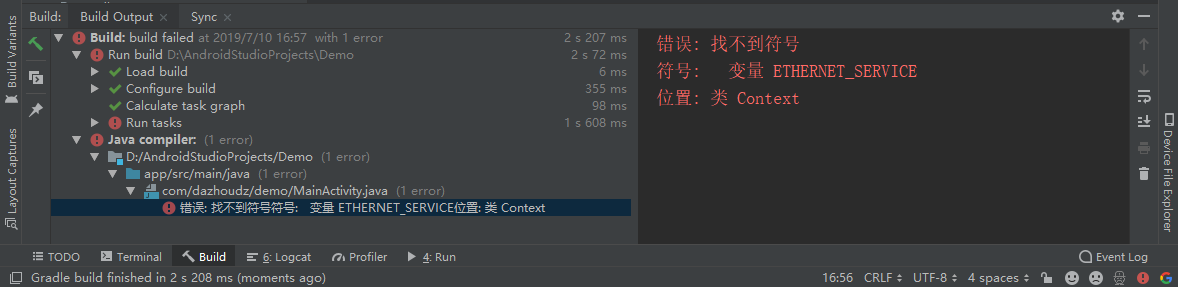
我们可以看到“Android SDK”是排在最前面的，所以先从Android SDK中查找Context，在这个Context中ETHERNET\_SERVICE这个隐藏属性肯定是找不到的，我们把Android SDK移到最后，如下：



此时再查看我们的代码，已经可以访问隐藏属性了，如下：



好像OK了，但是，此时运行项目后你就会看到有异常：



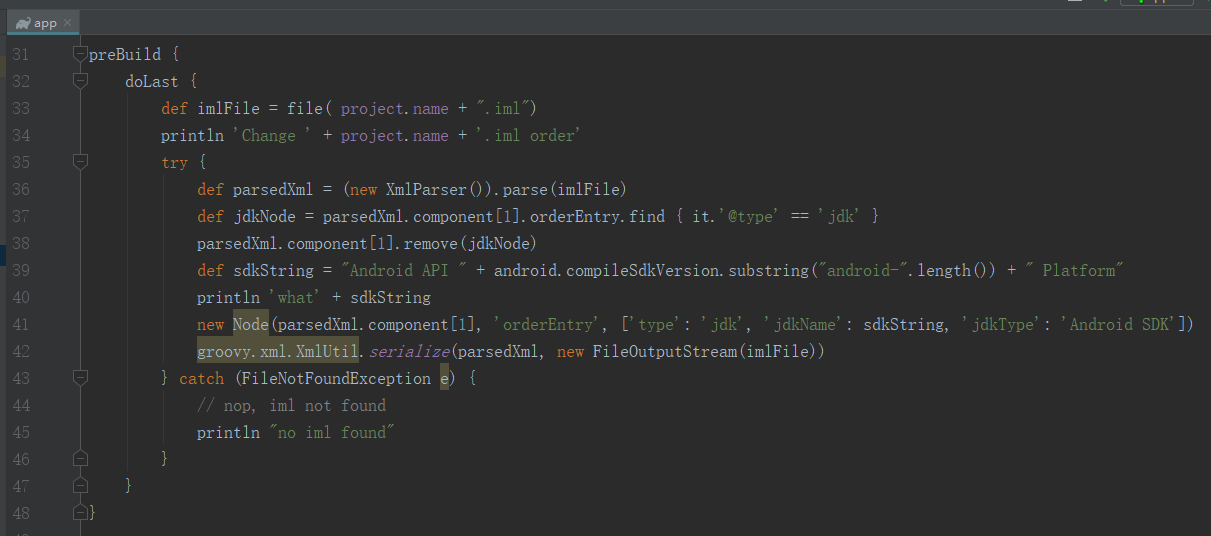
问题是说找不到Context中有ETHERNET\_SERVICE这个变量，见了鬼了！此时在项目的build.gradle中加入如下配置：



Gradle修改后会要求你同步一下，同步一下之后，你会发现app.iml中Android SDK的顺序又回到了最前面，我们把它再移到最后，然后运行项目，OK，通过了！

为什么要设置两个地方啊？好麻烦，我也懒得去搜索寻找为什么，反正问题已经解决了，我自己的猜测：调整app.iml中Android SDK的位置，这是给Android Studio这个IDE看的，因为IDE能发现我们写的代码中哪里有错，然后会提示出来，比如一个类有没有这个属性，IDE就根据我们在app.iml中声明的顺序去找，找不到就在Java源代码直接显示出错误。而在我们打包apk的时候需要把我们的.java的源代码编译为.class文件，Java编译器在编译.java文件时也需要知道我们引用的Context. ETHERNET\_SERVICE类正确与否，如果不正确是不会编译通过的，那Java编译器去哪找这个Context. ETHERNET\_SERVICE呢？答：还是去了Android SDK中查找的，这说明app.iml中Android SDK的顺序并不能决定编译时编译器查找类时的顺序，所以我们通过在项目的build.gradle的配置就是为了告诉Java编译器先去framework.jar中查找。 总结就是app.iml中的设置是给IDE用的，项目build.gradle中的设置是给Java编译器用的，也不知道理解的对不对，我自己瞎猜的。

最后一个问题，就是每次修改Gradle之后需要同步，一同步就会发现app.iml中Android SDK的顺序又回到了最前面，老是要手动去调很是烦人的，解决办法如下，在模块的build.gradle中加入如下任务：



每次同步Gradle时这个任务就会自动执行，它的功能就是把Android SDK的顺序移到最后。

总结：

需要3个设置，在项目的build.gradle中有1个设置，在模块的build.gradle中有两个设置，如下：

项目build.gradle



模块build.gradle

