



آموزش برنامه نویسی اندروید در محیط اندروید استودیو

اجرا و خطایابی پروژه اندروید بر روی دیوایس واقعی

مدرس : سید مهدی مطهری

www.android-studio.ir

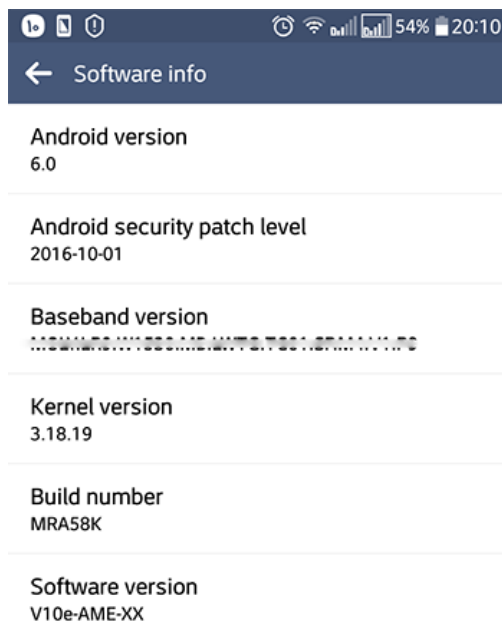


به نام خدا

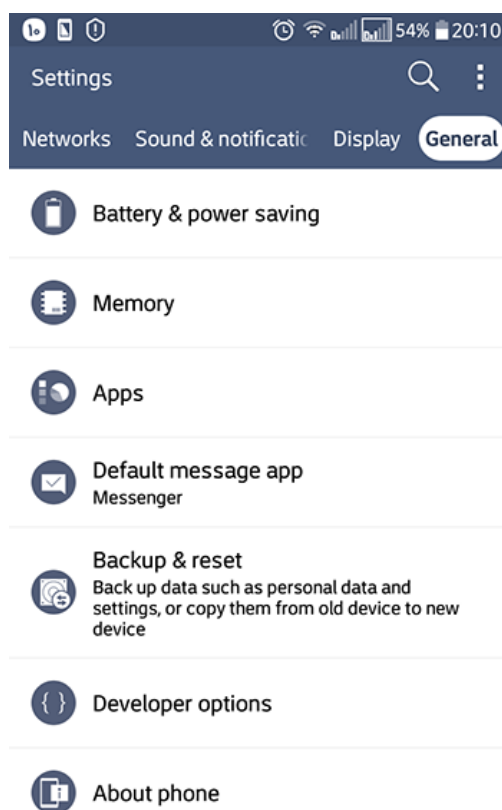
در مباحث گذشته با مفهوم شبیه ساز آشنا شدیم. از شبیه ساز اندروید استودیو (AVD) و شبیه سازهای ثانویه مانند Genymotion می توان جهت اجرا (Run) و همچنین خطایابی یا دیباگ (Debug) پروژه های اندرویدی استفاده کرد که قبلا با شبیه ساز جنی موشن به طور کامل آشنا شدیم. مزیت شبیه سازها این بود که نیاز به تهیه سخت افزار و دیوایس جداگانه را برای توسعه دهنده مرتفع ساخته و در یک محیط شبیه سازی شده اندروید این امکان را مهیا می کرد تا پروژه خود را روی تمامی نسخه های اندروید تست و دیباگ کنیم. با این حال توسعه دهنده ممکن است بنا به دلایلی در برخی مواقع مایل به اجرا و دیباگ اپلیکیشن خود بر روی دیوایس های واقعی گوشی و یا تبلت باشد.

احتمالا اولین راهی که به ذهن اکثر ما می رسد، ساخت پکیج نصبی apk، انتقال فایل به دیوایس و سپس نصب آن می باشد. مسلما این فرایند زمان زیادی را از ما می گیرد. یعنی اگر لازم باشد ۱۰ بار تغییراتی را در پروژه ایجاد و سپس مجدد روی دیوایس واقعی اجرا یا دیباگ کنیم، این پروسه باید ۱۰ بار تکرار شود.

گوگل راه ساده تری را پیش روی توسعه دهندگان اندروید قرار داده. به اینصورت که با اتصال دیوایس از طریق کابل USB به کامپیوتر به طور مستقیم در محیط اندروید استودیو امکان Run و Debug پروژه وجود دارد که این قابلیت USB Debugging نام دارد. برای استفاده از این قابلیت ابتدا باید آن را در قسمت تنظیمات دیوایس اندرویدی خود فعال کنیم. در قسمت Settings یا تنظیمات اندروید گزینه ای با عنوان Developer Options وجود دارد که مربوط به تنظیمات USB Debugging می شود. توجه داشته باشید اگر سیستم عامل دیوایس اندروید ۴.۲ و به بالا می باشد، گزینه Developer Options به صورت پیش فرض مخفی است که برای فعال سازی و نمایش آن لازم است به قسمت About phone رفته و ۷ بار روی قسمت Build number ضربه بزنیم (Build number معمولا در قسمت Software info قرار دارد). حالا Developer Options فعال شده و با برگشت مجدد به Settings این گزینه مشاهده می شود:



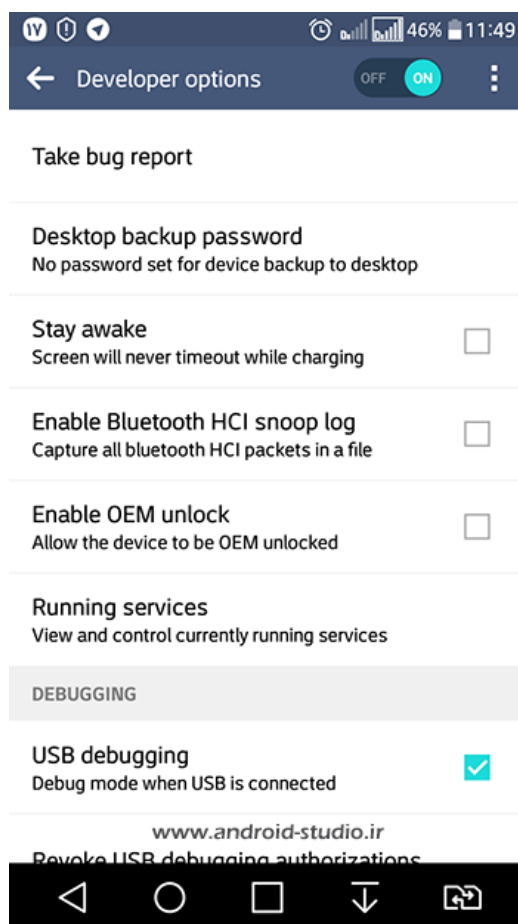
◀ ○ ◻ ▾ ↺



◀ ○ ◻ ⌵ ↺



وارد Developer Options شده و USB Debugging را فعال می کنیم:



توجه داشته باشید ممکن است در دیوایس های مختلف، مسیر دستیابی به این گزینه اندکی متفاوت باشد.

در مرحله بعد باید درایور دیوایس را روی کامپیوتر نصب کنیم. البته نصب درایور فقط در سیستم عامل ویندوز انجام می شود و در سیستم عامل های مک و لینوکس نیازی به نصب درایور نیست (لینک توضیحات مربوط به سیستم عامل های مک و لینوکس در انتهای مبحث قرار گرفته).

درایور دیوایس های سری Nexus (بجز Galaxy Nexus) که توسط گوگل تهیه و منتشر شده، مشترک هستند و Google USB Driver نام دارد. این درایور در SDK Manager در قسمت SDK Tools لیست شده که می بایست دانلود و نصب شود (و یا برای دریافت فایل نصب آفلاین به [فصل دوم](#) مراجعه کنید).

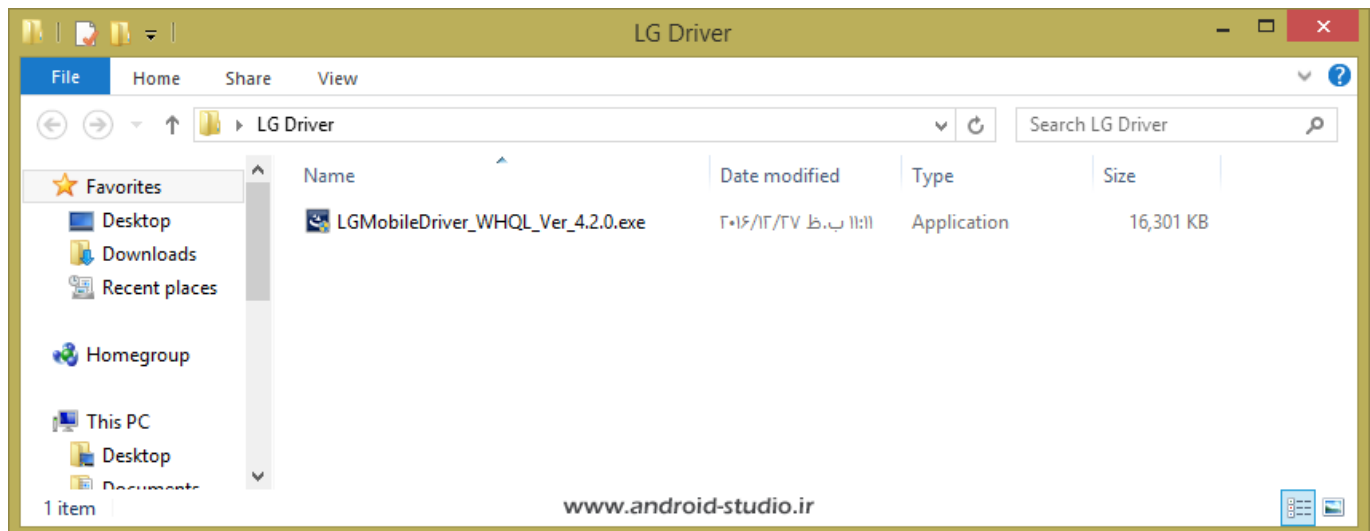
نکته: مسیر قرارگیری Google USB Driver (ابتدا فولدر usb_driver از حالت فشرده خارج و سپس به مسیر زیر منتقل گردد):

<sdk location>\extras\google\

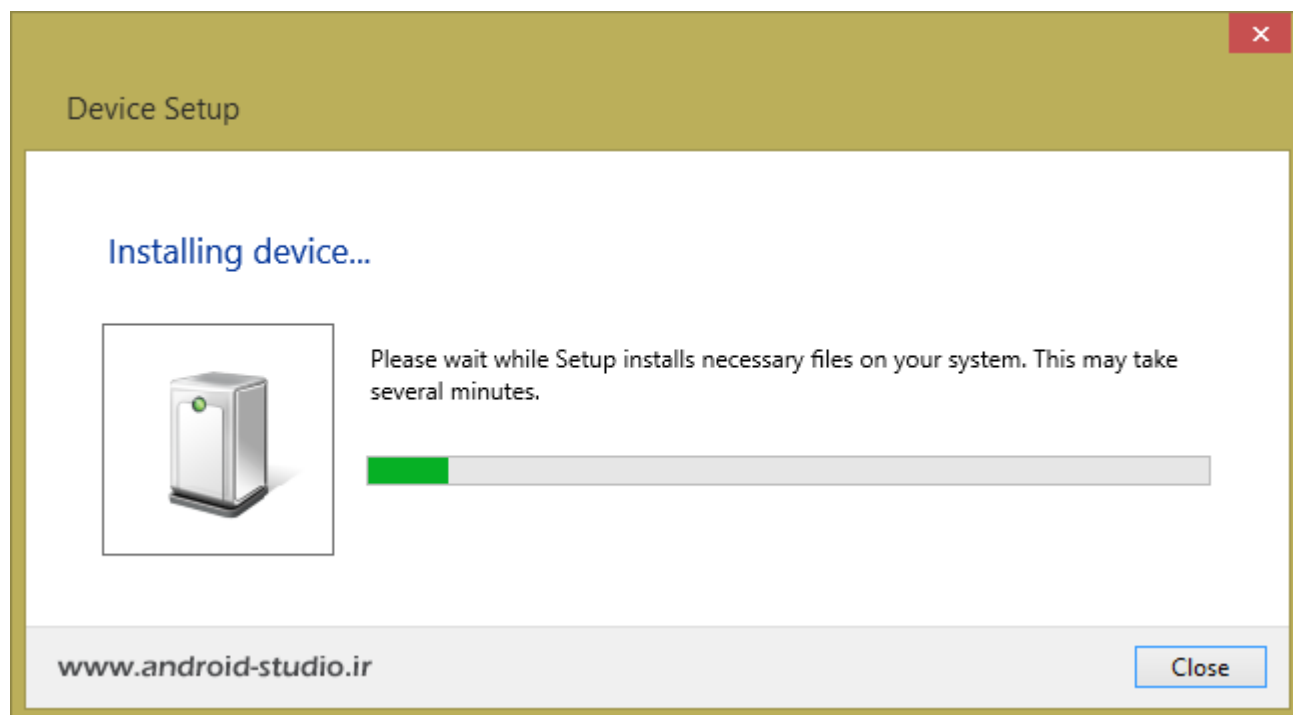


نکته: در اکثر موارد دیوایس های غیر نکسوس نیز با Google USB Driver سازگار بوده و نیازی به دانلود و نصب درایور اختصاصی شرکت مربوطه نیست و برای راحتی کار می توانید ابتدا با USB Driver تست کنید. با این حال اگر جواب نگرفتید پاراگراف بعد را مطالعه کنید.

اما سایر دیوایس ها مانند سامسونگ، ال جی، سونی و... درایور مجزا دارند که هر کمپانی برای دیوایس های خود یک درایور بر روی وب سایت خود منتشر می کند (عموما هر کمپانی برای تمامی دیوایس های خود یک درایور مشترک ارائه می دهد). دیوایسی که در این آموزش استفاده شده مربوط به کمپانی LG می باشد که با جستجوی عبارت LG Mobile Driver در گوگل، اولین نتیجه مربوط به وب سایت ال جی و دانلود درایور بود. درایور را دانلود می کنم. حجم درایور زیاد نیست (درایوری که من دانلود کردم ۱۶ مگابایت است).

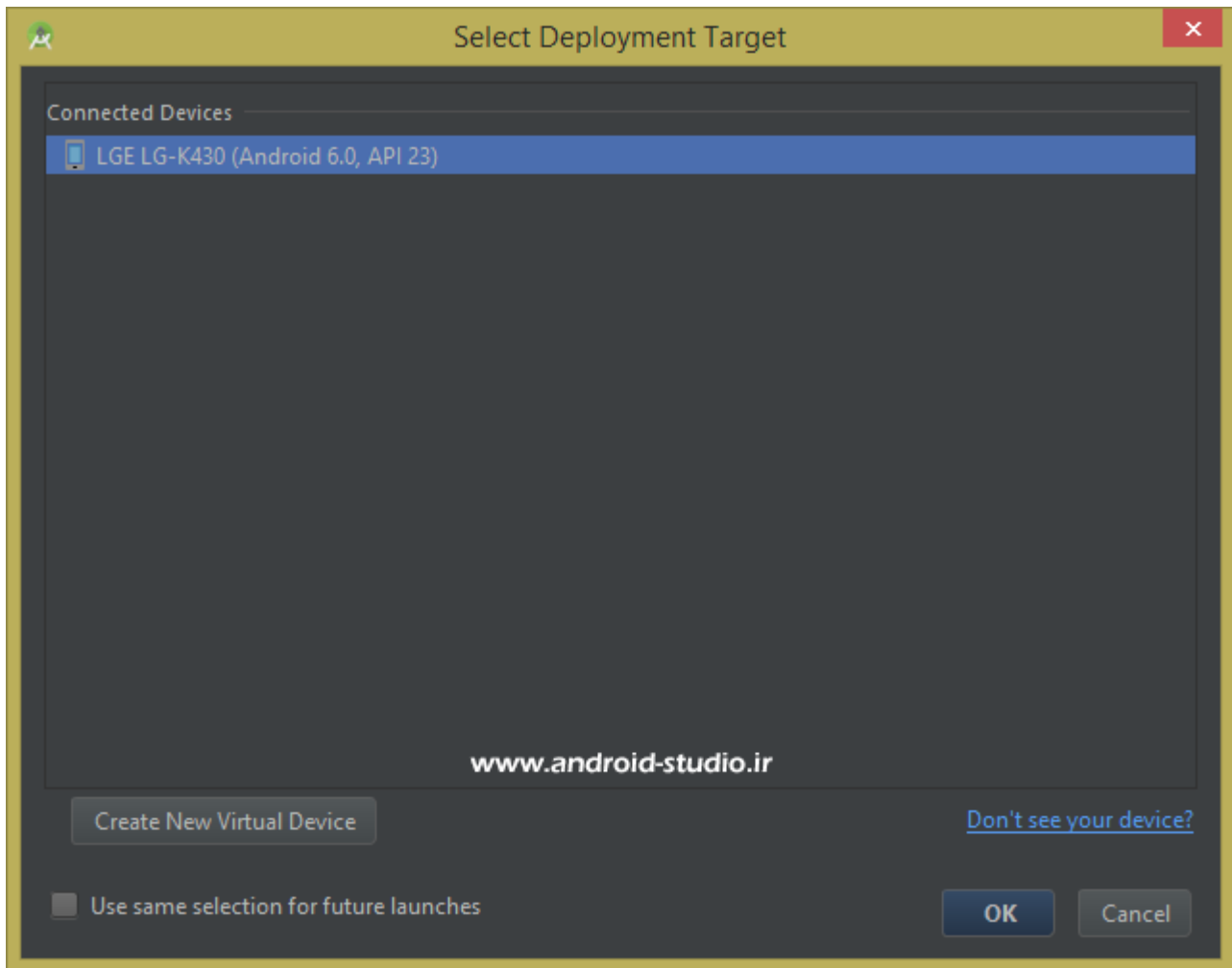


قبل از نصب درایور لازم است دیوایس توسط کابل USB به کامپیوتر متصل شود. پس از شناسایی دیوایس توسط ویندوز، فایل درایور را اجرا و فرایند نصب را شروع می کنم. پس از اتمام عملیات نصب، احتمالاً پنجره ای ظاهر می شود که نشان دهنده نصب درایور دیوایس روی سیستم عامل است (مشابه آنچه هنگام اولین اتصال یک USB Flash به کامپیوتر اتفاق می افتد). منتظر می مانم تا به اتمام برسد (ممکن است چند دقیقه زمان ببرد).



اگر اتصال USB به درستی برقرار بوده و مشکل دیگری روند نصب درایور را مختل نکرده باشد، عملیات نصب با موفقیت به پایان برسد.

به سراغ اندروید استودیو می روم. پروژه ی Intent که مربوط به آموزش قبل بود را باز می کنم. می خواهم پروژه ما بدون نیاز به ساخت پکیج apk و انتقال دستی به دیوایس، اجرا شود. پروژه را Run می کنم. اگر مراحل را به درستی انجام داده باشم باید دیوایسی که توسط کابل USB به کامپیوتر متصل شده در لیست Connected Devices نمایش داده شود:



نکته : در سیستم عامل ویندوز اگر در پنجره بالا دیوایس نمایش داده نمی شود به احتمال زیاد در نصب درایور مشکلی بوجود آمده است. در حینی که دیوایس به کامپیوتر متصل است به Device Manager ویندوز رفته و نام دیوایس خود را در زیرشاخه Other Devices بیابید. سپس روی آن راست کلیک کرده و گزینه Update Driver Software را انتخاب کنید. در مرحله بعد گزینه Browse my computer for driver software را انتخاب کرده و در مرحله نهایی مسیری که درایور را نصب کرده اید انتخاب کنید.

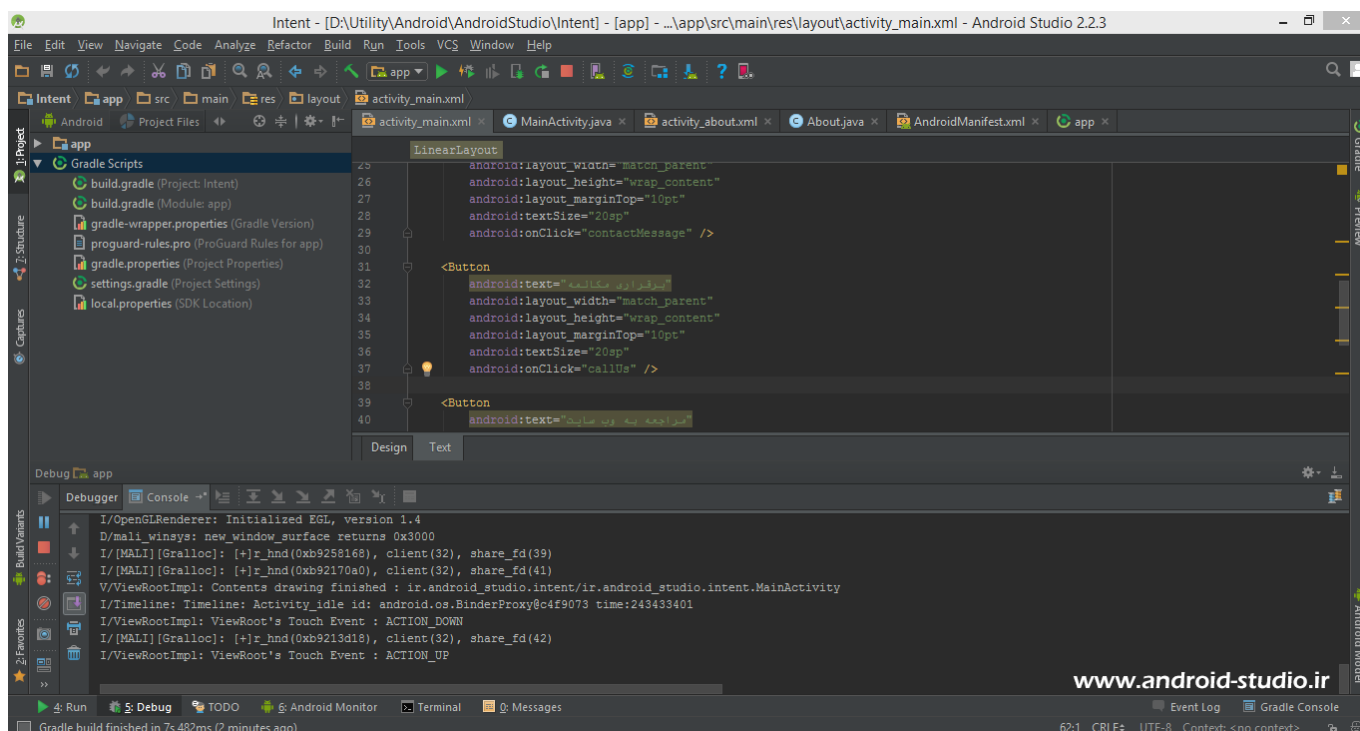
دیوایس را انتخاب می کنم. در این لحظه اگر گوشی یا تبلت در حالت قفل قرار دارد لازم است آن را از حالت قفل خارج کنم. ممکن است قبل از اجرای پروژه روی دیوایس پیغامی با این مضمون که "آیا اجازه استفاده از دیوایس توسط این کامپیوتر به عنوان دیباگر می دهید؟" نمایش داده شود که آنرا تایید می کنیم. پس از تایید این پیغام پروژه اجرا می شود:



پروژه اندروید من روی گوشی به درستی اجرا شد و قادر به تست قسمت های مختلف اپلیکیشن هستم.

یادآوری: اگر minSdkVersion پروژه از نسخه API دیوایس بالاتر باشد امکان اجرا و یا دیباگ میسر نخواهد بود و لازم است موقتاً آن را تغییر دهیم.

علاوه بر Run کردن پروژه، امکان دیباگ نیز وجود دارد. تفاوت Run و Debug در این است که در دیباگ می توان اشکالات برنامه را مشاهده و بررسی کرد.



گزینه Debug در سمت راست گزینه Run و یا از طریق منوی Run



برای دیباگ کردن پروژه لازم است این قابلیت در پروژه فعال گردد که به **دوروش** امکان پذیر است:

۱- اضافه شدن ویژگی debuggable با مقدار true به فایل build.gradle درون buildTypes :

```
android {
    buildTypes {
        debug {
            debuggable true
        }
    }
}
```



بلاک android پروژه ما پس از اضافه شدن دستور دیباگ:

```
android {
    compileSdkVersion 24
    buildToolsVersion "24.0.2"
    defaultConfig {
        applicationId "ir.android_studio.intent"
        minSdkVersion 10
        targetSdkVersion 24
        versionCode 6
        versionName "1.6"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
        debug {
            debuggable true
        }
    }
}
```

حالا توسط گزینه Sync Now پروژه را سینک می کنیم.

۲- اضافه کردن debuggable به تگ باز <application> درون مانیفست :

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme"
    android:debuggable="true">
```

و در نهایت دیباگ پروژه با انتخاب گزینه Debug.

تذکر: قبل از تهیه نسخه نهایی اپلیکیشن جهت انتشار عمومی برای کاربران، حتما قابلیت debug را از پروژه حذف کنید. فعال بودن دیباگ در برنامه ها، کدهایی به برنامه اضافه می کند که موجب افزایش حجم نهایی اپلیکیشن می گردد.

توضیحات مربوط به سیستم عامل های لینوکس و مک :

<https://developer.android.com/studio/run/device.html>



با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش های بهتر یاری فرمائید.

www.android-studio.ir