

به نام خدا

آموزش سیزدهم

اهداف آموزشی این قسمت عبارتند از:

۱. معرفی فایل `AndroidManifest.xml`

۲. آشنایی با `Version Code`

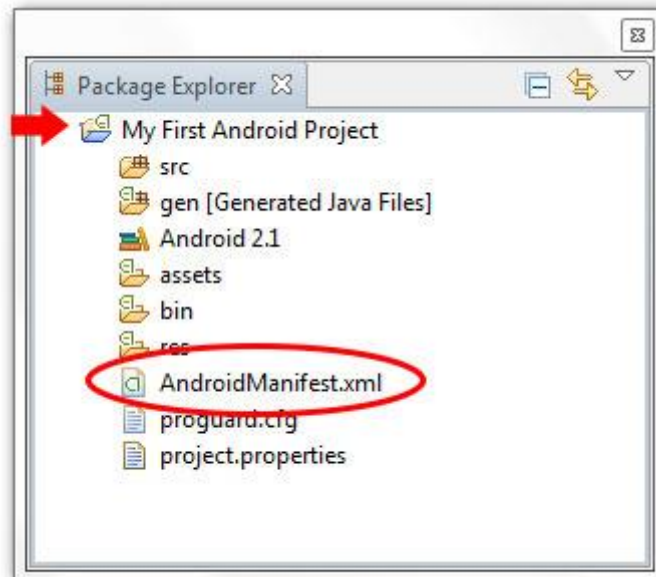
۳. آشنایی با `Version Name`

۴. معرفی فایل `project.properties`

پس از آشنایی با `Android Building Blocks` یا همان "آجرهای تشکیل دهنده یک اپلیکیشن اندرویدی" طی قسمت های نهم و دهم و یازدهم، در این قسمت از آموزش قصد داریم تا دو فایل بسیار مهم در توسعه اندروید را معرفی کنیم.

معرفی فایل `AndroidManifest.xml`

هر اپلیکیشن می بایست دارای فایلی تحت عنوان `AndroidManifest.xml` در دایرکتوری اصلی خود که همان `Root Directory` است باشد (به حروف کوچک و بزرگ در نام این فایل توجه کنید). برای درک بهتر `Root Directory` به تصویر زیر توجه کنید:



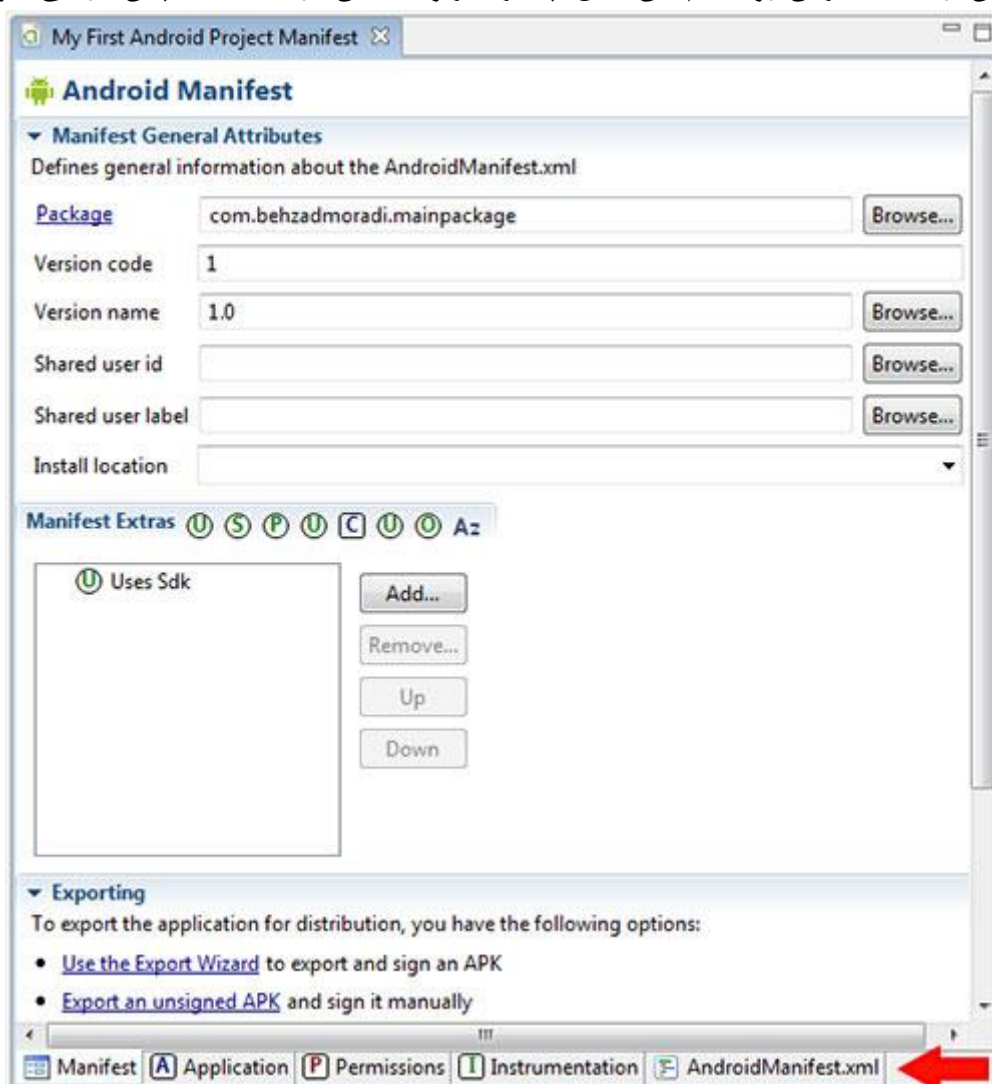
در تصویر فوق این فایل با یک بیضی قرمز رنگ نشان داده شده است. جایگاه فلش قرمز رنگ قرار گرفته است نشانگر `Root Directory` یا همان "دایرکتوری اصلی" اپلیکیشن است. به عبارت

دوره آموزش اندروید

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

دیگر تنها جاییکه این فایل می تواند قرار گیرد زیرشاخه همین دایرکتوری است که اگر جای این فایل تغییر یابند اپلیکیشن به هیچ وجه کار نخواهد کرد.
در واقع پس از کلیک کردن روی نام این فایل، پنجره مربوط به آن در محیط اکتیو است می شود:



همانطور که در تصویر فوق نشان داده شده است پس از باز کردن فایل Manifest در قسمت پایین پنجره مربوط به این فایل پنج Tab مجزا از یکدیگر دیده می شود. کلید این Tab ها به جز Tab یی تحت عنوان AndroidManifest.xml که در تصویر فوق با یک فلش قرمز رنگ نشان داده شده است دارای یک محیط گرافیکی هستند و این در حالی است که این محیط گرافیکی تا حدودی اعمال تغییرات در فایل Manifest را برای توسعه دهندگان مبتدی آسان ساخته است. به طور مثال در تصویر فوق Tab مربوط به Manifest باز است و در قسمت فوقانی این Tab اطلاعات مربوط به پیکج و Version Code و غیره دیده می شود. به طول مثال با تغییر Version Code از عدد 1 به عدد

2 بدون نیاز به کد نویسی می توان Version Code را آپدیت کرد(در ادامه همین آموزش با Version Code بیشتر آشنا خواهیم شد).

اکثر برنامه نویسان حرفه ای با حالت نمایش XML یی این فایل کار می کنند. در واقع با کلیک کردن روی Tab مرتبط با AndroidManifest.xml محتویات این فایل برای پروژه ای که در آموزش هشتم تحت عنوان My First Android Project ساختیم به شکل زیر نمایش داده خواهد شد:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.behzadmoradi.mainpackage"
    android:versionCode="1"
    android:versionName="1.0" >
    <uses-sdk android:minSdkVersion="7" />
    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

کاری که این فایل انجام می دهد این است که اطلاعاتی کلی پیرامون اپلیکیشن طراحی شده در سیستم اندروید می دهد. به عبارت دیگر، این فایل Component هایی را که ما در برنامه خود مورد استفاده قرار داده ایم را به سیستم عامل اندروید معرفی می کند. از سوی دیگر این فایل Permission ها یا "مجوزهایی" که کاربر برای نصب برنامه می بایست صادر کند را نیز در بر می گیرد. فرض کنیم که برنامه ای طراحی کرده ایم که نیاز به دسترسی به اینترنت دارد. حال اگر کسی بخواهد این برنامه طراحی شده توسط ما را مورد استفاده قرار دهد، مجبور است در حین نصب اجازه دسترسی به اینترنت را به این اپلیکیشن صادر کند.

همانطور که در آموزش اول توضیح داده شد، سیستم عامل اندروید دارای نسخه های مختلفی است. چنانچه ما برنامه ای طراحی کنیم که برای مثلا نسخه Gingerbread یا نسخه 3.0 طراحی شده باشد، این حداقل نسخه مورد نیاز برای اجرای این اپلیکیشن یا همان Minimum SDK در فایل Manifest ذکر خواهد شد و چنانچه کاربری که سیستم عامل اندروید وی پایین تر از نسخه 3.0 باشد بخواهد برنامه طراحی شده ما را استفاده کند در حین نصب فایل Manifest این مسئله را خواهد فهمید

دوره آموزش اندروید

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

و از نصب برنامه جلوگیری به عمل خواهد آورد. در واقع در کد فوق دستور `android:minSdkVersion="7"` حداقل SDK مورد نیاز اپلیکیشن را مشخص کرده است.

مواردی دیگری علاوه بر موارد فوق نیز در این فایل مشاهده می شود که در ذیل به آنها اشاره می کنیم: نامی که برای پکیج اپلیکیشن مورد استفاده قرار دادیم در این فایل قرار گرفته است. علت قرار گرفته نام پکیج در فایل Manifest این است نام این پکیج به منزله یک معرف منحصر به فرد از اپلیکیشن برای سیستم عامل اندروید می باشد. همانطور که در کد فوق مشاهده می شود `package="com.behzadmoradi.mainpackage"` نشانگر نام پکیجی است که کلاس های جاوای اپلیکیشن در آن قرار گرفته اند.

کلیه `Activity` ها و `Service` های به کار گرفته شده در اپلیکیشن که تحت عنوان `Component` های اپلیکیشن شناخته می شوند نیز در این فایل معرفی می شوند. در کد فوق خطی که حاوی `android:name=".MainActivity"` می باشد تنها `Activity` یی که در حین ساخت پروژه تعریف کردیم را نشان می دهد.

علاوه بر موارد فوق، فایل `Manifest` اطلاعاتی پیرامون نسخه اپلیکیشن ما به کاربران نیز ارائه می دهد. اگر در حین دانلود یا نصب اپلیکیشن های اندروید توجه کرده باشید، دیده اید که `Version` یا نسخه اپلیکیشن مد نظر به کاربران اطلاع داده می شود (مثلاً نسخه 4 اپلیکیشن). به طور کلی می توان گفت که هر اپلیکیشن اندروید دارای یک `Version Code` و یک `Version Name` است.

آشنایی با `Version Code`

منظور از `Version Code` یک عدد است و کاربرد این عدد برای کاربران اپلیکیشن نیست بلکه برای `Google Play` است تا از طریق این عدد از یک سو خود اپلیکیشن را شناسایی کند و از سوی دیگر آپدیت های اپلیکیشن را پیگیری کند. به عبارت دیگر کاربران به هیچ وجه این کد را مشاهده نخواهند کرد. نکته ای که برای تعیین شماره `Version Code` همواره می بایست مد نظر قرار داد این است که این شماره از هر عددی می تواند شروع شود اما همواره پس از آپدیت کردن اپلیکیشن این عدد می بایست نسبت به نسخه قبلی عدد بزرگ تری باشد. به طور مثال اگر نسخه اول اپلیکیشن دارای `Version Code` یی معادل با 1 بود نسخه بعدی پس از آپدیت توسط توسعه دهنده اپلیکیشن می بایست مثلاً معادل با 2 باشد.

به منظور درک بهتر این مسئله سناریوی زیر را در نظر می گیریم:

فرض کنیم که ما یک اپلیکیشن طراحی کرده و `Version Code` آن را معادل با 1 قرار داده ایم و آن را از طریق `Google Play` در دنیا منتشر می کنیم. پس از مدتی مشخص می شود که

دوره آموزش اندروید

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

اپلیکیشن ما پس از ۲۰ دقیقه کار کردن Crash می کند. بنابراین سعی می کنیم نواقص اپلیکیشن را رفع کرده، مجدد آن را Compile می کنیم و آن را در Google Play مجدد منتشر می کنیم. نکته ای که فراموش کرده ایم پس از رفع نواقص اپلیکیشن آپدیت کنیم Version Code اپلیکیشن مان است. در واقع پس از آنکه نسخه اصلاح شده را در Google Play منتشر کردیم Google Play متوجه نخواهد شد که ویرایش جدید اپلیکیشن روی سرور اش قرار گرفته از اینرو به کاربرانش که این اپلیکیشن را دانلود کرده اند نمی تواند اطلاع رسانی کند که نسخه جدید و اصطلاح شده اپلیکیشن را مجدد دانلود کنند. اما اگر Version Code را به عدد 2 تغییر دهیم این مسئله به اطلاع Google Play خواهد رسید و بالتبع کاربرانی که از این اپلیکیشن استفاده می کنند نیز از آپدیت اپلیکیشن مطلع خواهند شد.

همانطور که در کد فوق ملاحظه می شود دستور `android:versionCode="1"` نشانگر نسخه یک اپلیکیشن ما است که در اختیار Google Play قرار می گیرد.

آشنایی با Version Name

به طور کلی Version Name نامی است برای رویت کابرنی است که اپلیکیشن را دانلود می کنند. نحوه نامگذاری Version Name می تواند بر اساس استراتژی باشد که توسعه دهنده اپلیکیشن برای اپلیکیشن هایش در نظر می گیرد. به طور مثال می توان Version Name را بر اساس تاریخ انتشار اپلیکیشن در نظر گرفت.

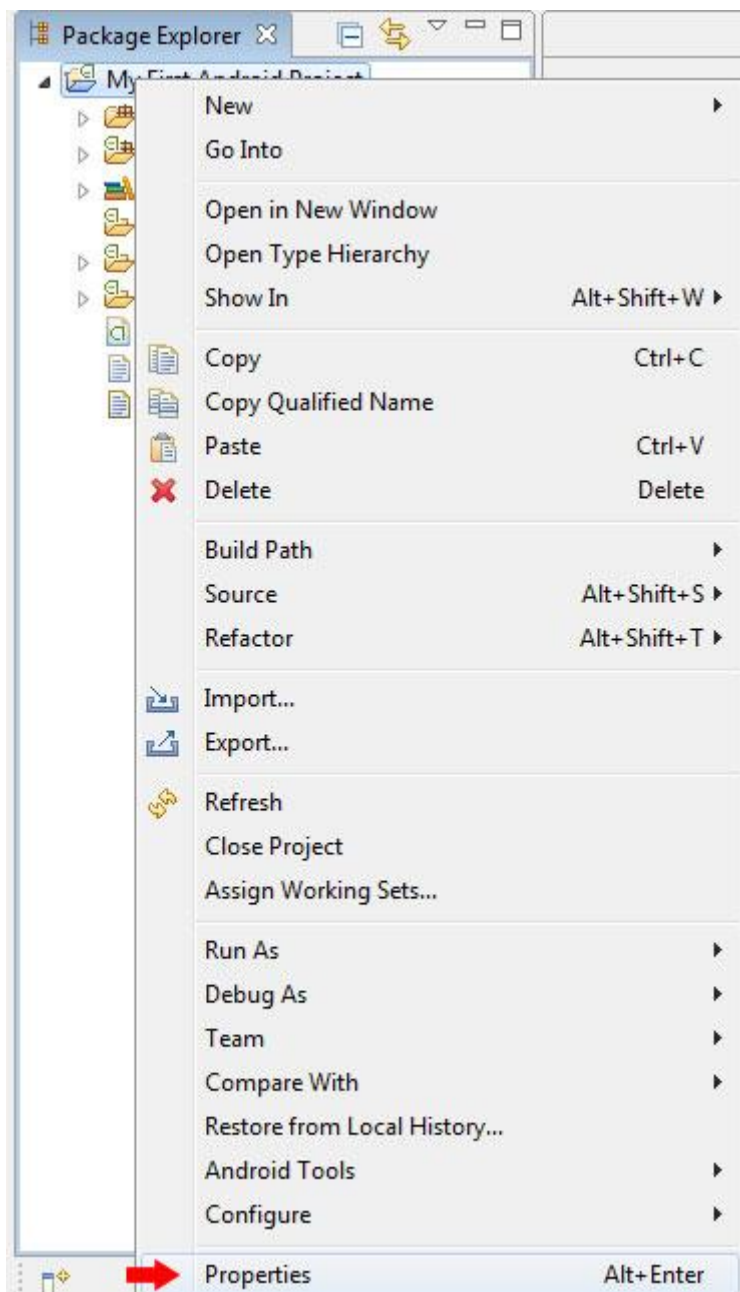
در کد فوق دستور `android:versionName="1.0"` نشانگر کد 1.0 است که به کاربران نشان داده می شود.

معرفی فایل project.properties

این فایل نیز همانند فایل Manifest از اهمیت زیادی در اجرای اپلیکیشن برخوردار است. یکی از دلایل اصلی وجود این فایل در یک اپلیکیشن ایجاد ارتباط مابین اپلیکیشن، اکلیپس و ADT می باشد(برای آشنایی بیشتر با ADT به آموزش هفتم راجعه نمایید). از جمله اطلاعات مهمی که در این فایل قرار می گیرد Target SDK است(نام دیگر Target SDK عبارت Build SDK است). همانطور که در آموزش های پیشین توضیح داده شد Target SDK مشخص می کند که ما اپلیکیشن خود را برای کدام نسخه از سیستم عامل اندروید می سازیم. در واقع زمانیکه در آموزش هشتم ما اولین پروژه خود را در اکلیپس تحت عنوان My First Android Project ساختیم، نسخه سیستم عامل مد نظر یا

همان Target SDK یا نسخه 2.1 انتخاب کردیم. حال اطلاعات مربوط به Target SDK در این فایل ذخیره شده اند.

نکته ای که در ارتباط با این فایل بسیار مهم است این است که این فایل به هیچ وجه نمی بایست بر خلاف فایل Manifest به صورت دستی ویرایش شود. برای ویرایش این فایل می بایست مطابق تصویر زیر عمل کرد:

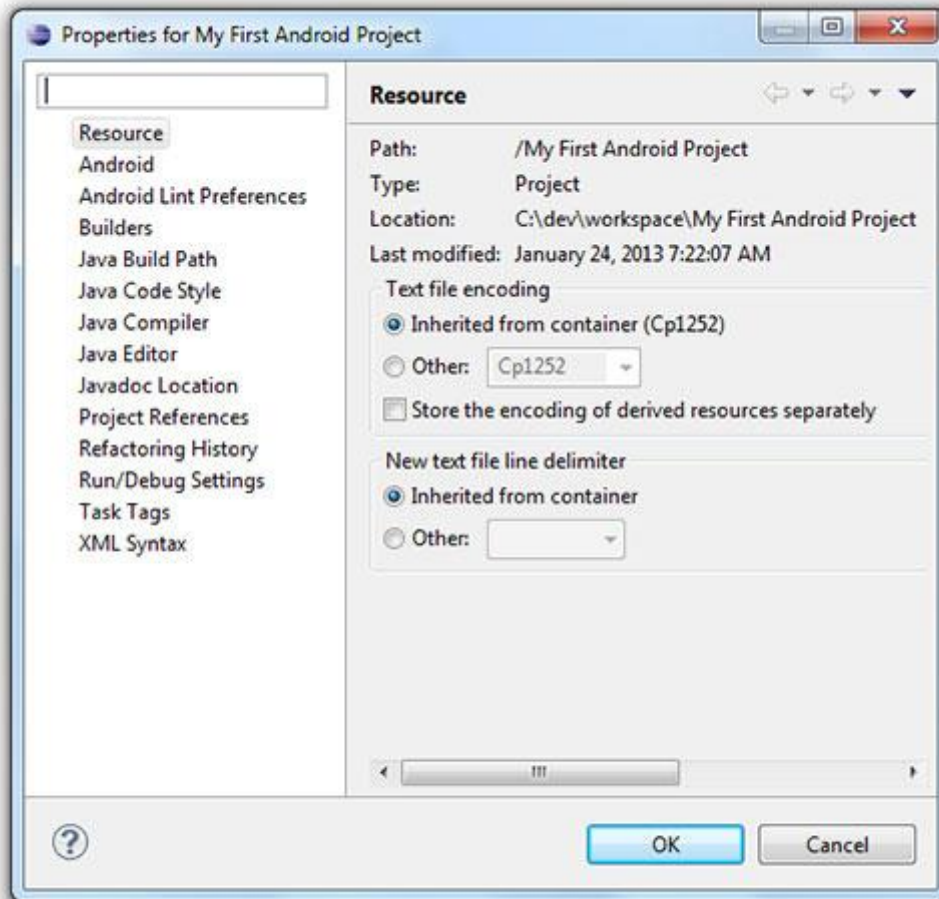


همانطور که در تصویر فوق نشان داده شده است، روی نام پروژه ای که ساخته ایم کلیک راست نموده سپس همانطور که با فلش قرمز رنگ نشان داده شده است روی آخرین گزینه از پنجره ای باز می شود یا همان گزینه Properties کلیک می کنیم و پنجره ای به صورت زیر باز خواهد شد:

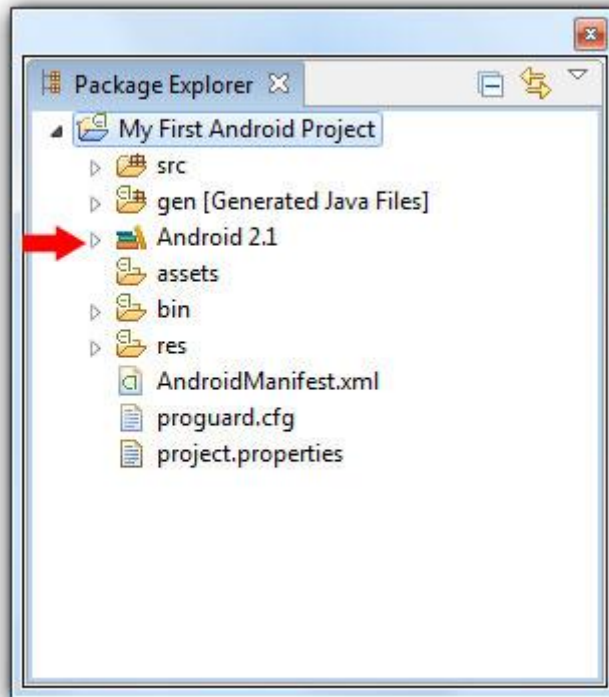
دوره آموزش اندروید

کلیه حقوق متعلق به وب سایت نردبان است.

مدرس: بهزاد مرادی

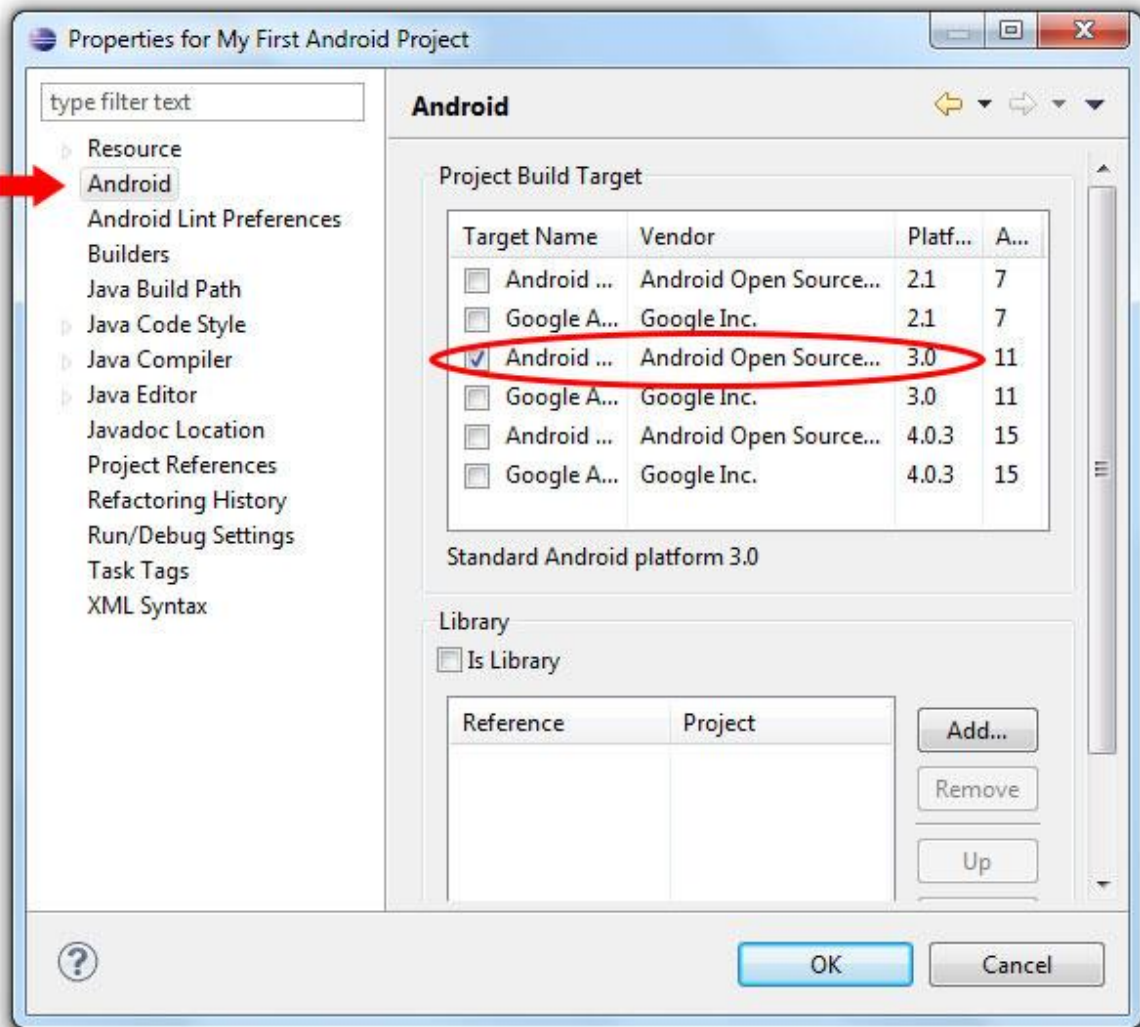


پروژه ای که در قسمت هشتم ساختیم و در آن نسخه 2.1 را به عنوان Target SDK مبنا قرار دادیم را مجدد مد نظر قرار می دهیم:

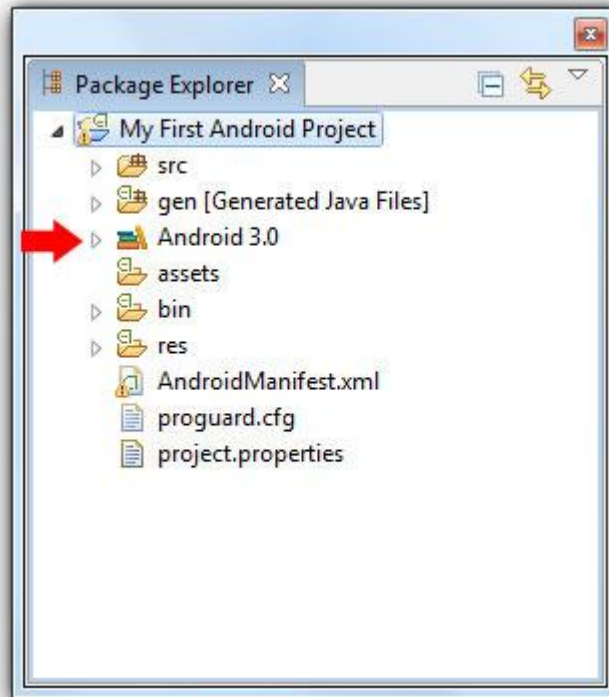


همانطور که در تصویر فوق مشخص شده است، فولدر مربوط به Library های همان نسخه ای از Target SDK که در حین ساخت پروژه انتخاب کردیم وارد اپلیکیشن شده است. حال فرض کنیم که نسخه اندروید 2.1 پاسخگوی نیازهای ما نیست و قابلیت هایی را که مد نظر ما است تا در اپلیکیشن استفاده کنیم را ساپورت نمی کند و می بایست از نسخه های جدیدتر سیستم عامل اندروید مثل نسخه 3 تحت عنوان Honeycomb به معنی "شانه عسل" استفاده کنیم.

از آنجا که می دانیم اطلاعات مربوط به Target SDK در فایل project.properties ذخیره می شود، همانطور که در قسمت بالا توضیح داده شد این فایل را باز کرده و مطابق تصویر زیر نسخه Target SDK را ارتقاء می دهیم:



پس از کلیک روی گزینه Properties همانطور که در تصویر فوق با فلش قرمز رنگ نشان داده شده است روی گزینه Android کلیک کرده سپس از ستون سمت راست گزینه مرتبط با اندروید نسخه 3.0 را انتخاب نموده و دکمه OK را می‌زنیم (لازم به ذکر است نسخه‌هایی از SDK اندروید در این لیست نمایش داده می‌شوند که در حین آماده‌سازی محیط برنامه نویسی اکلیپس دانلود کرده و به اکلیپس معرفی کرده ایم).



اکنون همانطور که در تصویر فوق مشاهده می شود Library مرتبط با نسخه سیستم عامل اندروید 3.0 وارد پروژه شده و از این پس خواهیم توانست از قابلیت های معرفی شده در نسخه 3.0 در طراحی اپلیکیشن خود استفاده کنیم.

پس از مطالعه این آموزش انتظار می رود بتوانیم به سوالات زیر پاسخ بدهیم:

۱. مکان قرارگیری فایل `AndroidManifest.xml` کجا است؟
 ۲. چه اطلاعاتی در فایل `Manifest` قرار می گیرند؟
 ۳. نحوه نامگذاری `Version Code` به چه شکل است؟
 ۴. کاربرد `Version Code` چیست؟
 ۵. کاربرد `Version Name` چیست؟
 ۶. نحوه تغییر نسخه اندروید یک اپلیکیشن به چه شکل است؟
 ۷. آیا می توان فایل `project.properties` را به صورت دستی ویرایش کرد یا خیر؟
- در آموزش آتی با نحوه بهینه سازی اپلیکیشن برای ساپورت دستگاه هایی با اندازه صفحه نمایش متفاوت آشنا خواهیم شد.