UNIVERSITY OF ALBERTA
FACULTY OF SCIENCE
Department of Computing Science

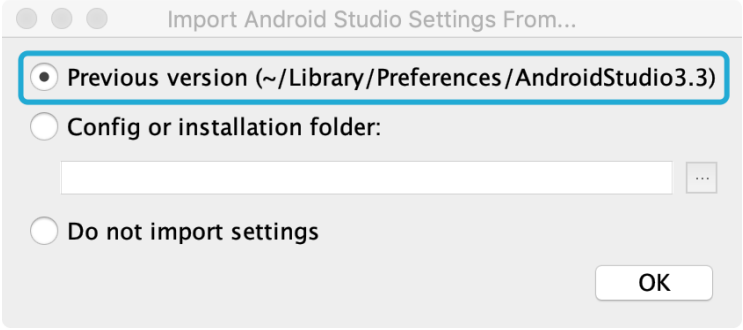
SOFTWARE DESIGN AND ARCHITECTURE

Android Studio 3 to 4 && Android 10 (API 29) Upgrade Tutorial

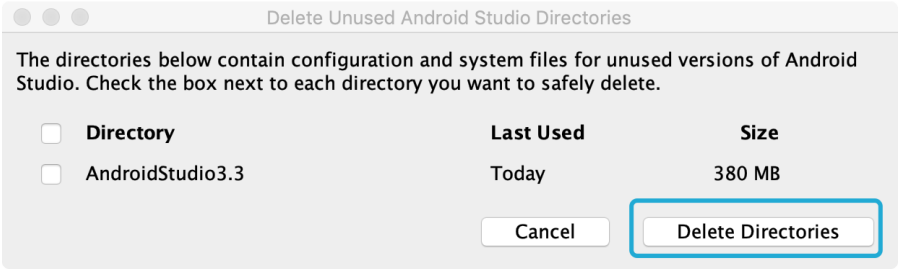
If you are using Android Studio 3.4 with API 26 (Android 8) from our previous courses/sessions, **we recommend** you to start from scratch: install the newest Android Studio and download the newest codebase.
However, if you would like to manually upgrade the code yourself, please do the following steps to make the old code work in Android Studio 4.1 with API 29 (Android 10).

1. **Download** Android Studio (version 4.1.0 or newer). Replace the old Android Studio on your machine with the newly downloaded software. https://developer.android.com/studio

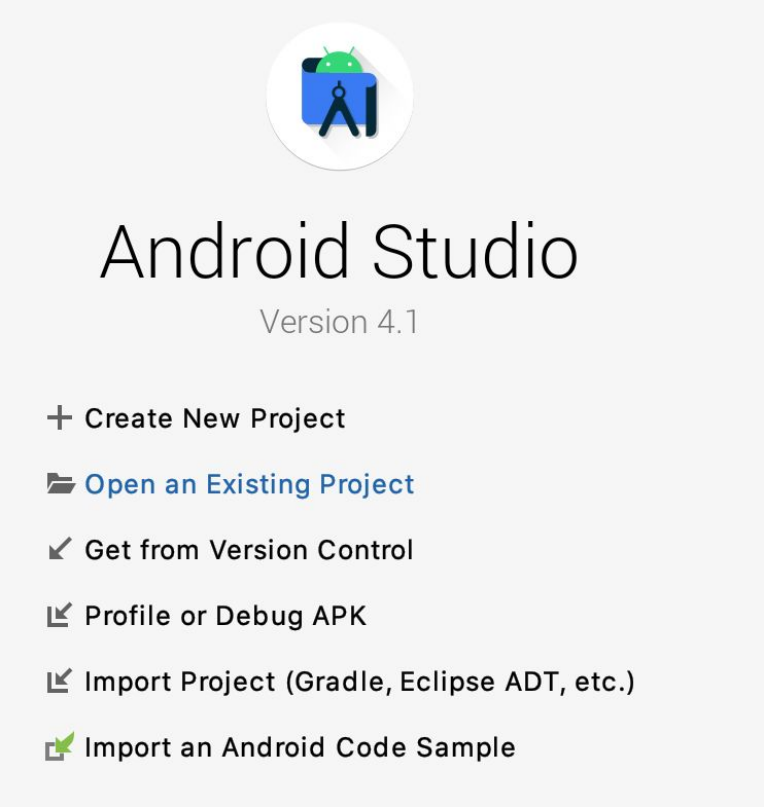
2. Install it. (It might ask where you want to import your settings from. Keep it as from the "Previous version")



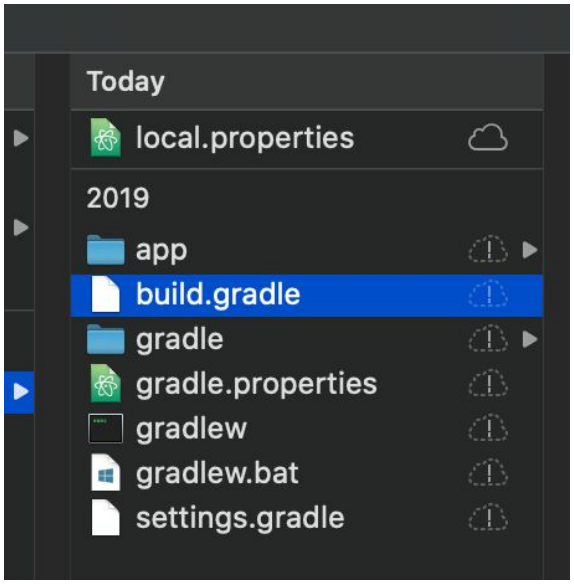
You can also delete **Unused Android Studio Directories** from old versions if you are not using them anymore.



3. Open pre-existing codebase:
   Open Android Studio and click **"Open an existing Android Studio project"**

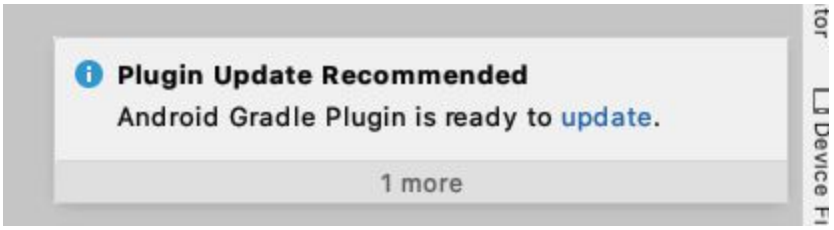


- Navigate and select **build.gradle**

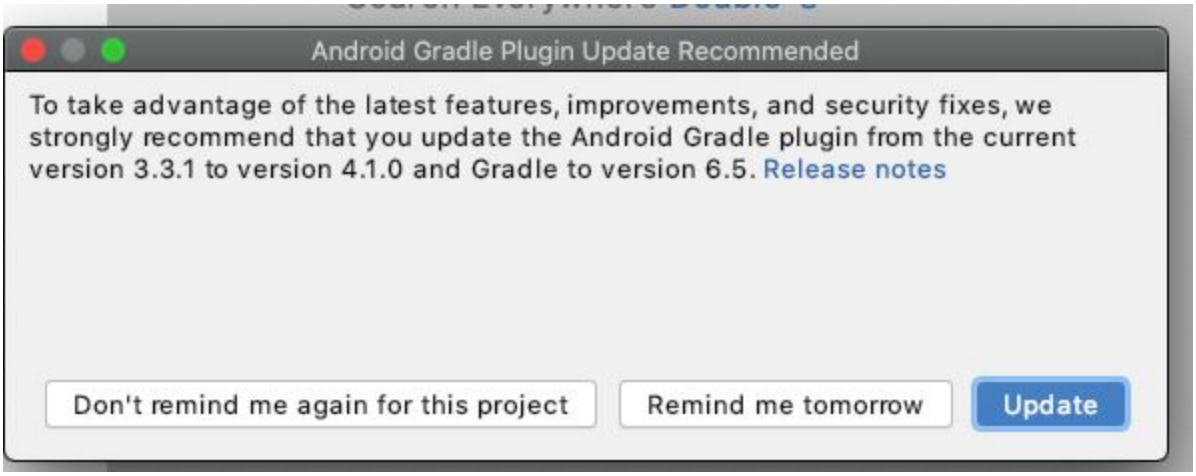


- Once you click **OK** it will start building the project, which may take a few minutes to complete.
- You may need to download or update Java -- Android Studio will let you know when you try to build the project.
- Update everything it prompts you to update.

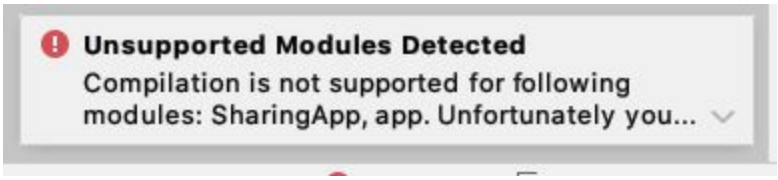
4. (Optional) It might prompt you to update Android Gradle Plugin; if so, select **update**.



As it suggested, the new gradle version will have the newest features, improvements, and bug fixes, so let's go ahead and select **Update.**



5. (Optional) if you happen to get this error:

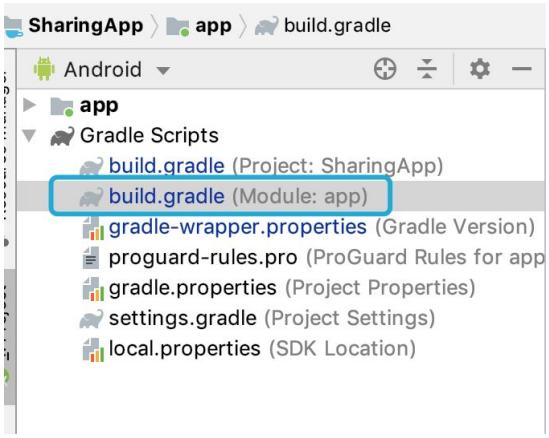


Please do the following:

1- close the project
2- close `Android Studio` IDE
3- delete the `.idea` directory
4- delete all `.iml` files
5- open `Android Studio` IDE and import the project

（Souce: Stackoverflow)

6. Open your module-level `build.gradle` file



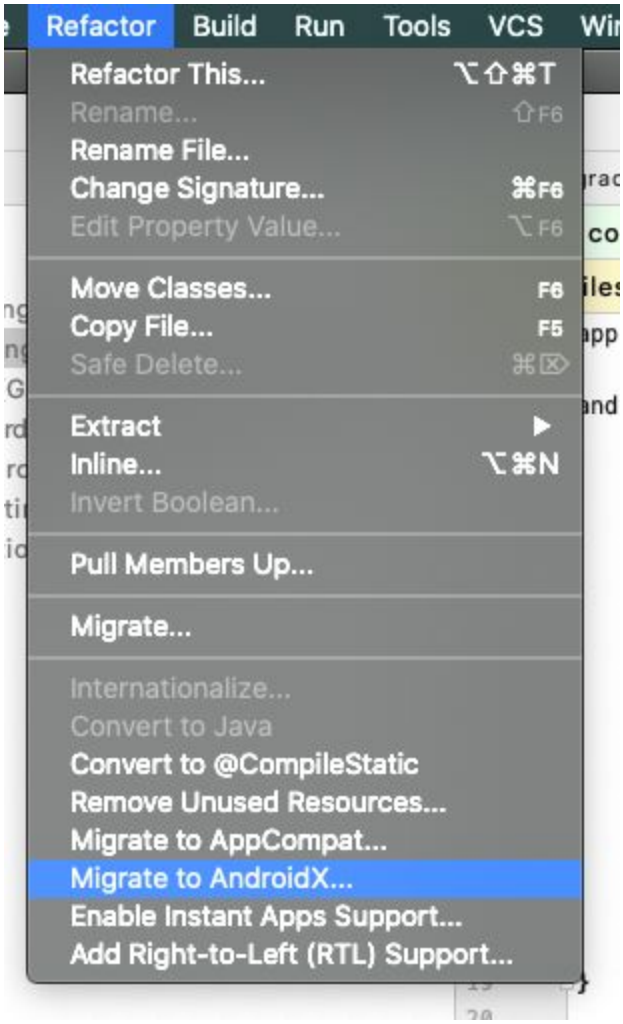
and update the `compileSdkVersion` and `targetSdkVersion` to 29:

```
android {
    compileSdkVersion 29
    defaultConfig {
        applicationId "com.example.sharingapp"
        minSdkVersion 19
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('
        }
    }
}
```

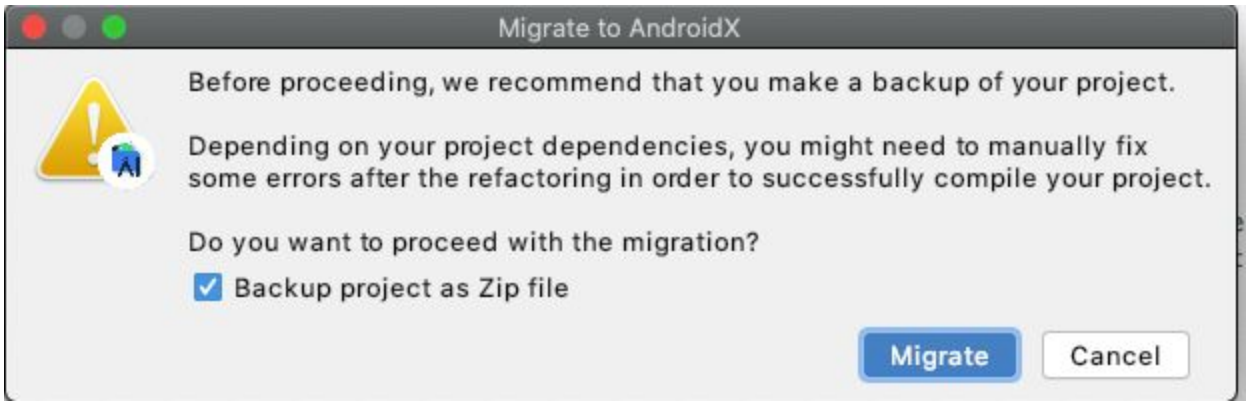
You will notice that some dependencies will turn red immediately.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    androidTestImplementation('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    implementation files('src/include/gson-2.8.2-SNAPSHOT.jar')
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support:support-v4:26.1.0'
    implementation 'com.android.support:design:26.1.0'
    testImplementation 'junit:junit:4.12'
}
```

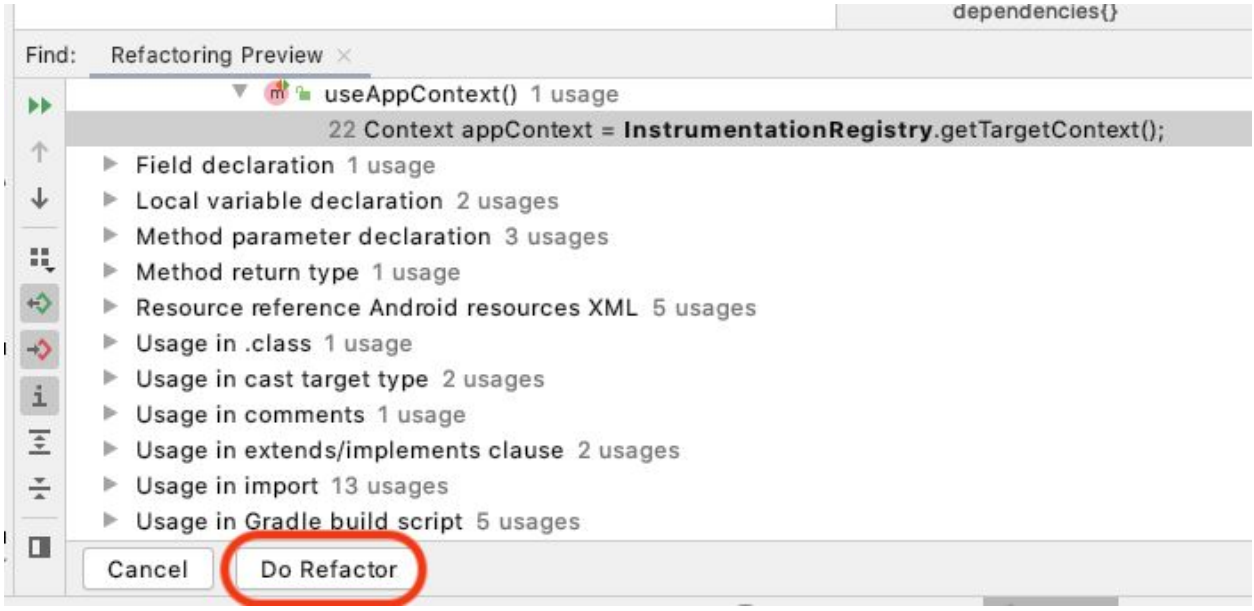
To solve this, go to **Refactor** and select **Migrate to Android X:**



Select **Migrate.**



After backing up your project, it will start looking for Usages in your codebase:

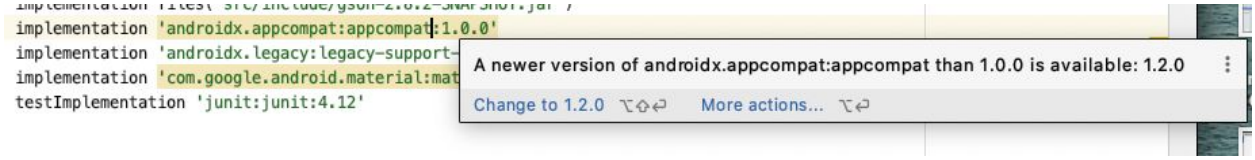


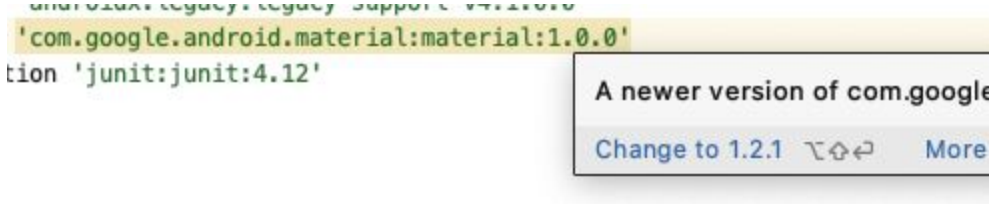
Select **"Do Refactor".** After it's done, you can see that dependencies are updated:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    androidTestImplementation('androidx.test.espresso:espresso-core:3.1.0', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    implementation files('src/include/gson-2.8.2-SNAPSHOT.jar')
    implementation 'androidx.appcompat:appcompat:1.0.0'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'com.google.android.material:material:1.0.0'
    testImplementation 'junit:junit:4.12'
}
```

You will notice that two of the dependencies have a yellow background. That's because Android Studio is trying to tell you that these dependencies have a newer version that's available.

Hover over to each dependency that has a yellow background, and as suggested, select "**Change to 1.2.0**" and "**Change to 1.2.1**" accordingly.





Once this step is done, your grade file will look like this:

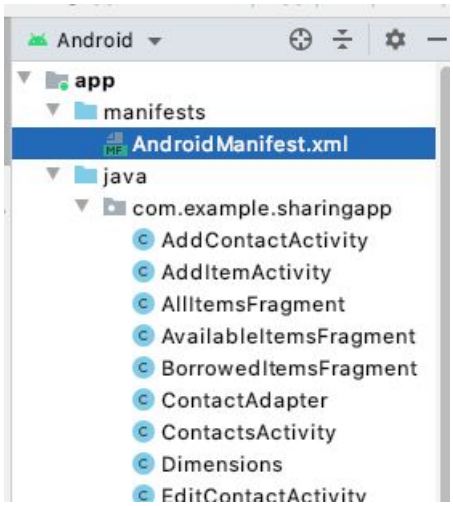
```
apply plugin: 'com.android.application'

android {
   compileSdkVersion 29
   defaultConfig {
       applicationId "com.example.sharingapp"
       minSdkVersion 19
       targetSdkVersion 29
       versionCode 1
       versionName "1.0"
       testInstrumentationRunner
'androidx.test.runner.AndroidJUnitRunner'
   }
   buildTypes {
       release {
           minifyEnabled false
           proguardFiles
getDefaultProguardFile('proguard-android.txt'),
'proguard-rules.pro'
       }
   }
}

dependencies {
   implementation fileTree(dir: 'libs', include: ['*.jar'])

androidTestImplementation('androidx.test.espresso:espresso-core
:3.1.0', {
       exclude group: 'com.android.support', module:
'support-annotations'
   })
   implementation files('src/include/gson-2.8.2-SNAPSHOT.jar')
   implementation 'androidx.appcompat:appcompat:1.2.0'
   implementation 'androidx.legacy:legacy-support-v4:1.0.0'
   implementation 'com.google.android.material:material:1.2.1'
```

```
    testImplementation 'junit:junit:4.12'
}
```

You may also notice there's a tip that appears on the top too. Click **"Ok, apply suggestions!"**

Once done, click **Sync Now:**

7. Since our targetSdkVersion is set to be 29 (Android 10), we will experience tiger restrictions around Scoped storage APIs. As a result of that, we will need to add one more permission in the manifest file.
   Open the manifest file:



   Add this line:

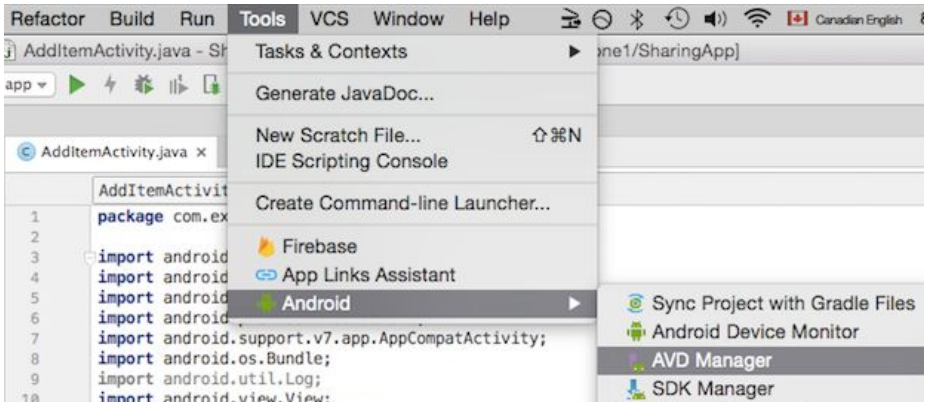
```
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

   Before <application> tag:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sharingapp">

    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Sharing App"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
```

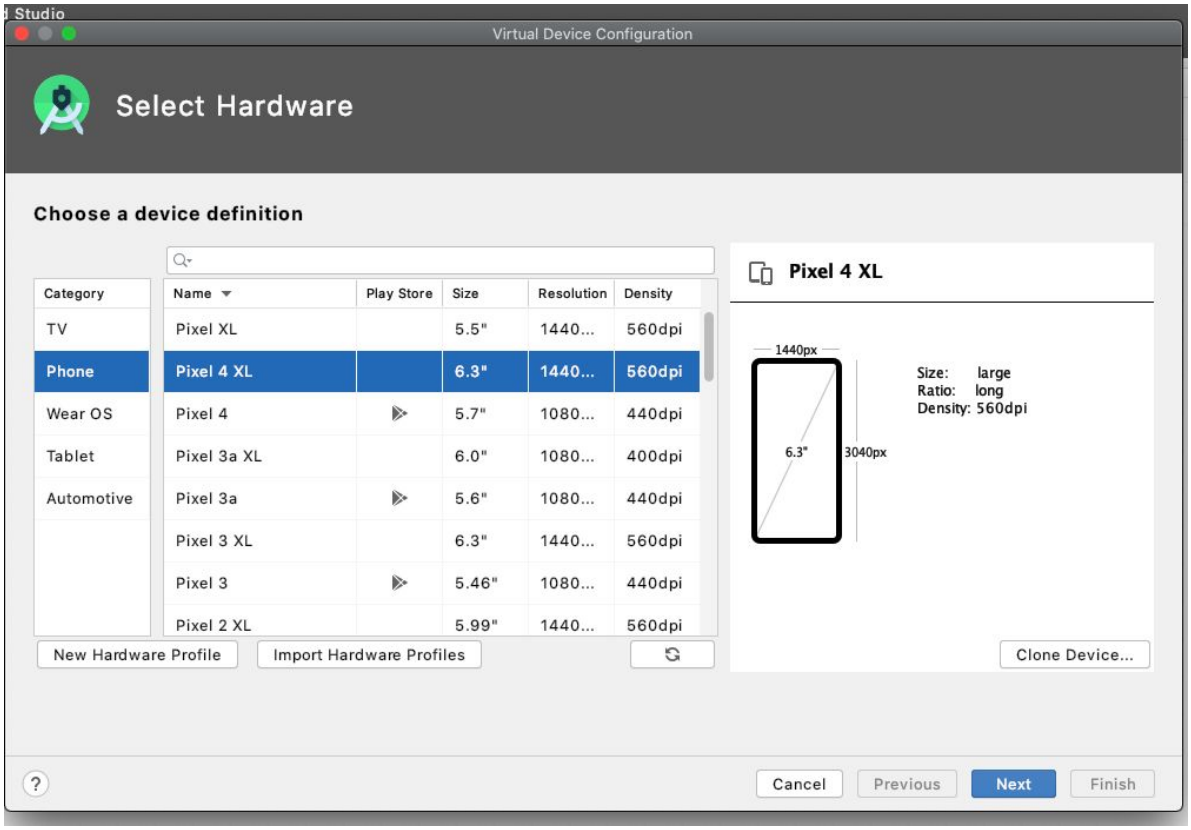
8. Next you will need to **create an Android Virtual Device (AVD)** i.e., an android emulator.

- Open the AVD Manager: **Tools → Android → AVD Manager**



- Click **Create Virtual Device**

- Select the any device of your choice - here we use **Pixel 4 XL**; click **Next**



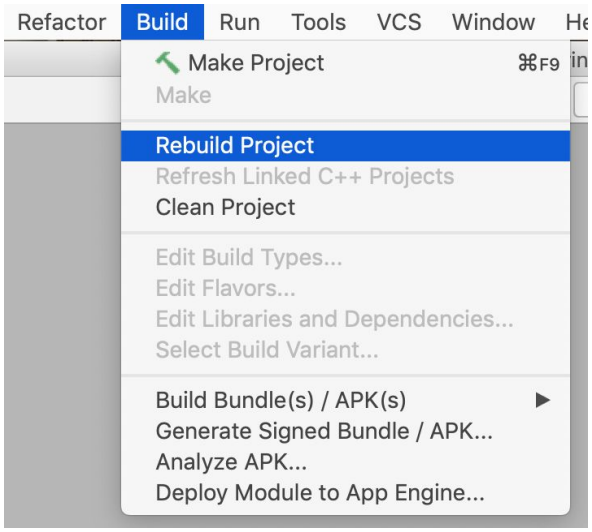
- Select ***Q*** **-** API level 29 (Aka Android 10). You may have to download this first and this may take a while.

Release Name	API Level	ABI	Target
R	R	x86	Android 11.0 (Google APIs)
R	30	x86	Android 11.0 (Google APIs)
Q	29	x86	Android 10.0 (Google APIs)
Oreo Download	*27*	*x86*	*Android 8.1 (Google APIs)*
Oreo	26	x86	Android 8.0 (Google APIs)
Nougat Download	*25*	*x86*	*Android 7.1.1 (Google APIs)*
Nougat Download	*24*	*x86*	*Android 7.0 (Google APIs)*
Marshmallow Download	*23*	*x86*	*Android 6.0 (Google APIs)*
Lollipop Download	*22*	*x86*	*Android 5.1 (Google APIs)*

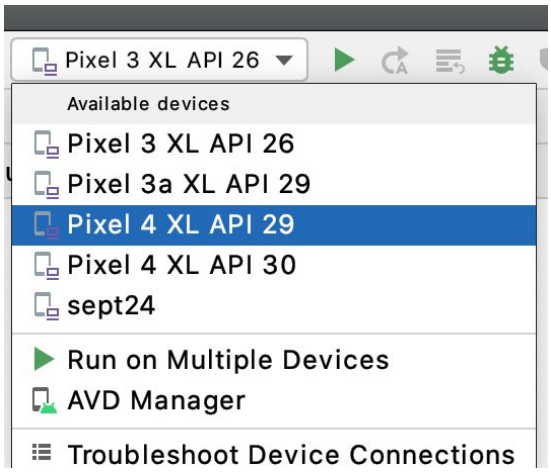
- It will prompt you to give it a name to identify it. The name will have no effect on how the program runs, as long as you know which one you need to run the app.

- Click **Finish** when this is complete.

9. Rebuild the app by clicking: **Build** -> **Rebuild Project**

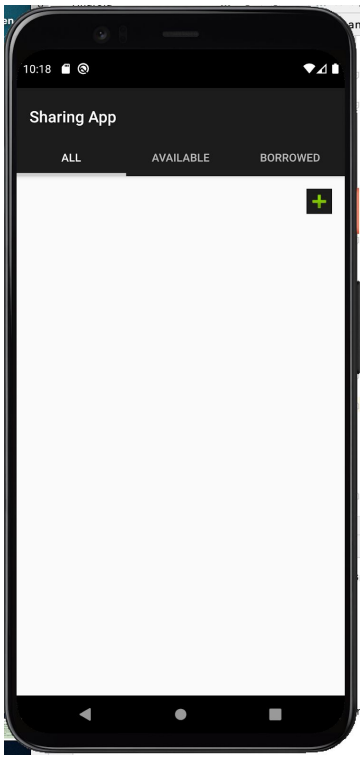


10. Select the virtual device you just added from the drop down menu:



11. Click the **play button** to run the app.

- The emulator takes a while to load, install and run your app. Be patient!
- Once your emulator finish loading and the app will start:



Congrats, you can now start programming in Android Studio! Play around with the app to see how it works!