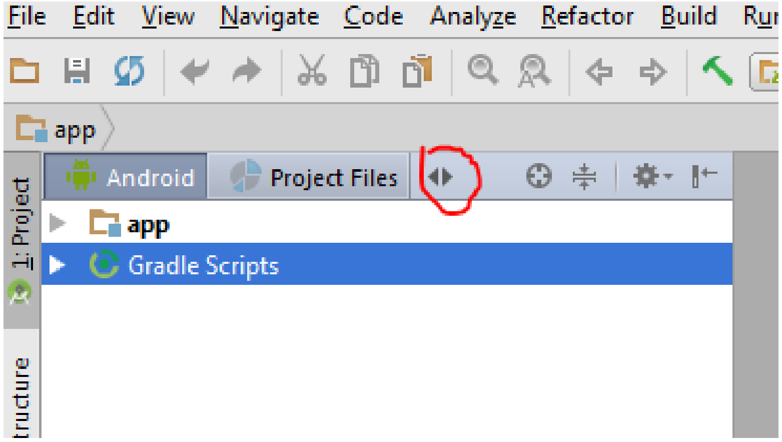
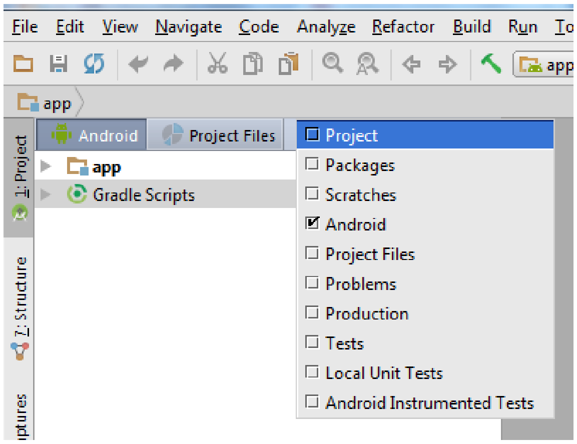
Introduction to Project Structure

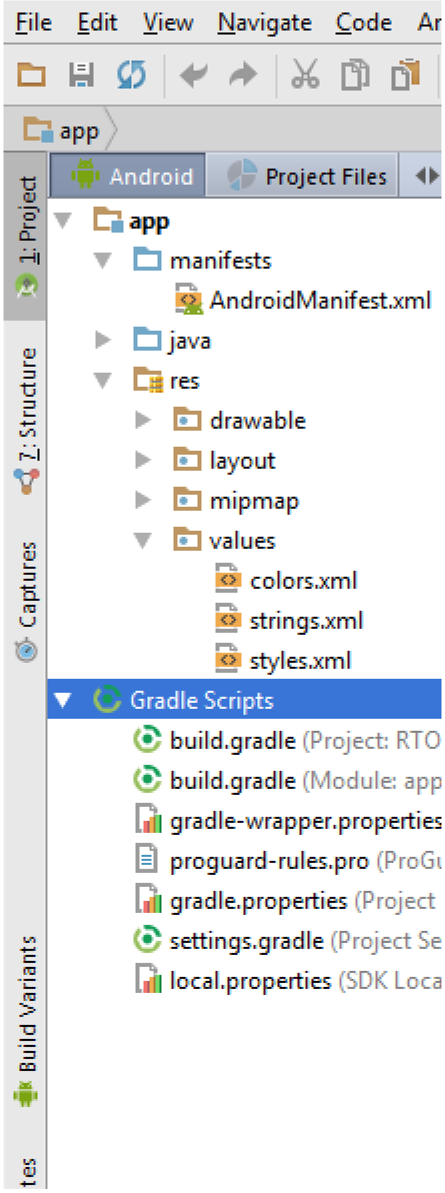
1. Project View
   Always make sure you select Android as it is very easy to navigate between different components of the project, while u can also go for Project view only when u want to see outputs/libs or other folders.





Note: The below Project Structure is respective to Android View.

2. Project Structure



```
app
  manifests
    AndroidManifest.xml
  java
  res
    drawable
    layout
    mipmap
    values
      colors.xml
      strings.xml
      styles.xml
Gradle Scripts
  build.gradle (Project: RTO
  build.gradle (Module: app
  gradle-wrapper.properties
  proguard-rules.pro (ProGu
  gradle.properties (Project
  settings.gradle (Project Se
  local.properties (SDK Loca
```

Two main Directories

A . app

B. Gradle Scripts.

App

Android Manifest File

AndroidManifest.xml is one of the most important file in the Android project structure.It contains information of the package, including components of the application such as activities, services, broadcast receivers, content providers etc.
It performs the following tasks:

- It is responsible to protect the application to access any protected parts by providing the permissions
- It also declares the android api that the application is going to use
- It lists the instrumentation classes. The instrumentation classes provides profiling and other informations. These information are removed just before the application is published etc.

java

The java folder contains the Java source code files of the application organized into packages. We can have more than one package in the Android application. Its always a good practice to break the source code of the application into different packages based on its core functionality. All the source files of the Activities, Services etc. go into this folder.

res

Res folder is where all the external resources for the application such as images, layout XML files, strings, animations, audio files etc. are stored.

- **Drawable**: This folder contains the bitmap file to be used in the program. There are different folders to store drawables. They are drawable-ldpi, drawable-mdpi, drawable-hdpi,drawable-xdpi etc. The folders are to provide alternative image resources to specific screen configurations. Ldpi, mdpi & hdpi stands for low density, medium density & high density screens respectively. The resources for each screen resolutions are stored in respective folders and the android system will choose it according to the pixel density of the device
- **Layout**: It contains XML files that define the User Interface of the application
- **Menu**: XML files that define menus for the application goes into this folder
- **Mipmap**: The mipmap folders is used for placing the app icons only. Any other drawable assets should be placed in the relevant drawable folders as before.
- **Values** : XML files that define simple values such as strings, arrays, integers, dimensions, colors, styles etc. are placed in this folder

Gradle Scripts

Gradle scripts are used to automate tasks.For the most part, Android Studio performs application builds in the background without any intervention from the developer. This build process is handled using the Gradle system, an automated build toolkit designed to allow the ways in which projects are built to be configured and managed through a set of build configuration files.It uses a language called groovy.

Buid.gradle(Project):Top-level build file where you can add configuration options common to all sub-projects/modules.

Build.gradle(Module):

- As is evident from the file content, the build file begins by declaring the use of the Gradle Android plug-in.
- The android section of the file then states the version of both the SDK and the Android Build Tools that are to be used when building.
- The items declared in the defaultConfig section define elements that are to be generated into the module's AndroidManfest.xml file during the build. These settings, which may be modified in the build file, are taken from the settings entered within Android Studio when the module was first created.
- The items declared in the defaultConfig section define elements that are to be generated into the module's AndroidManfest.xml file during the build. These settings, which may be modified in the build file, are taken from the settings entered within Android Studio when the module was first created.
- Finally, the dependencies section lists any local and remote dependencies on which the module is dependent.