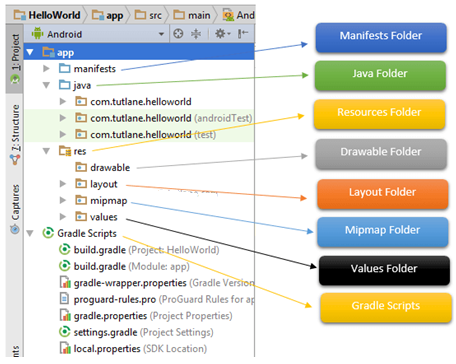
**Android App / Project Folder Structure**

To implement android apps, **Android Studio** is the official IDE (Integrated Development Environment) which is freely provided by Google for android app development.

Once we [setup android development environment](https://www.tutlane.com/tutorial/android/android-development-environment-setup) using android studio and if we create a sample application using android studio, our project folder structure will be like as shown below.

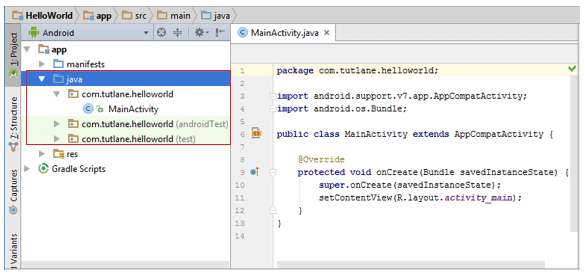


The Android project structure on the disk might be differs from the above representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown instead of **Android**.

The android app project will contain a different type of app modules, source code files and resource files. We will explore all the folders and files in android app.

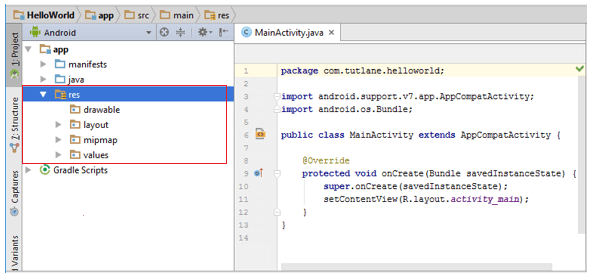
**Java Folder**

This folder will contain all the java source code (**.java**) files which we’ll create during the application development, including JUnit test code. Whenever we create any new project / application, by default the class file **MainActivity.java** will create automatically under the package name “**com.mms.helloworld**” like as shown below.



**res (Resources) Folder**

It’s an important folder which will contain all non-code resources, such as bitmap images, UI strings, XML layouts like as show below.



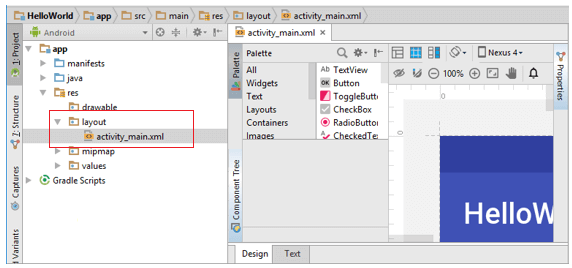
The res (**Resources**) will contain a different type of folders those are

Drawable Folder (res/drawable)

It will contains the different type of images as per the requirement of application. It’s a best practice to add all the images in **drawable** folder other than app / launcher icons for the application development.

**Layout Folder (res/layout)**

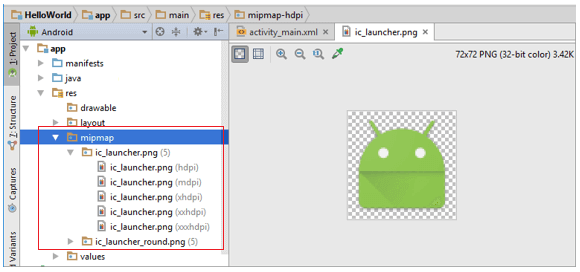
This folder will contain all XML layout files which we used to define the user Interface of our application. Following is the structure of **layout** folder in android application.



**Mipmap Folder (res/mipmap)**

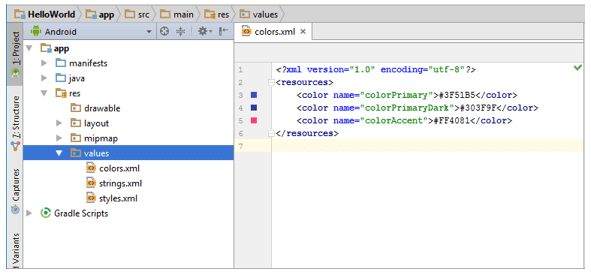
This folder will contain app / launcher icons which are used to show on the home screen. It will contain different density type of icons such as hdpi, mdpi, xhdpi, xxhdpi, xxxhdpi, to use different icons based on the size of device.

Following is the structure of **mipmap** folder in android application.



**Values Folder (res/values)**

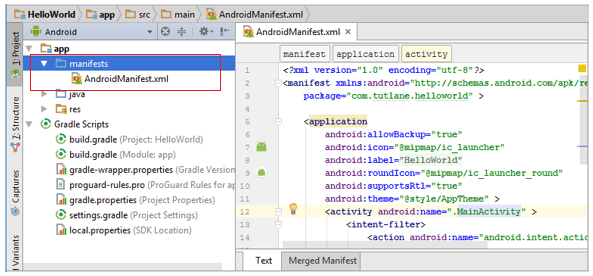
This folder will contain a various XML files, such as strings, colors, styles definitions and static array of strings or integers. Following is the structure of **values** folder in android application.



**Manifests Folder**

This folder will contain a manifest file (**AndroidManifest.xml**) for our android application. This manifest file will contain information about our application such as android version, access permissions, metadata, etc. of our application and its components. The manifest file will act as an intermediate between android OS and our application.

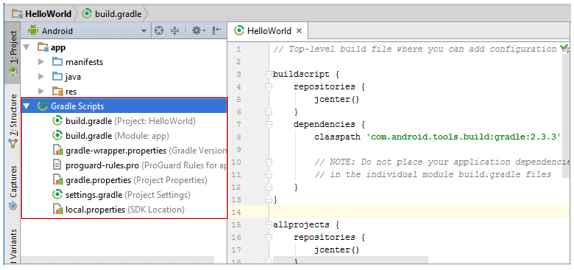
Following is the structure of **mainfests** folder in android application.



Gradle Scripts

In android, Gradle means automated build system and by using this we can define a build configuration that apply to all modules in our application. In gradle **build.gradle (Project)**, **build.gradle (Module)** are used to build configurations that apply to all our app modules or specific to one app module.

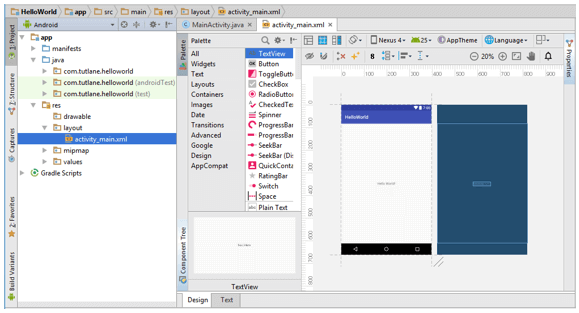
Following is the structure of **Gradle Scripts** in android application.



Following are the important files which we need to implement an app in android studio.

**Android Layout File (activity\_main.xml)**

The UI of our application will be designed in this file and it will contain **Design** and **Text** modes. It will exists in **layouts**folder and the structure of **activity\_main.xml** file in **Design** mode like as shown below.



We can make a required design modifications in **activity\_main.xml** file either using **Design** or **Text** modes. If we switch to **Text** mode **activity\_main.xml** file will contain a code like as shown below.

<?xml version="1.0" encoding="utf-8"?>  
<android.support.constraint.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools=<http://schemas.android.com/tools>

    android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent"  
    tools:context="com.tutlane.helloworld.MainActivity">

    <TextView  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="Hello World!"  
        app:layout\_constraintBottom\_toBottomOf="parent"  
        app:layout\_constraintLeft\_toLeftOf="parent"  
        app:layout\_constraintRight\_toRightOf="parent"  
        app:layout\_constraintTop\_toTopOf="parent" />

</android.support.constraint.ConstraintLayout>

**Android Main Activity File (MainActivity.java)**

The main activity file in android application is **MainActivity.java** and it will exists in **java** folder. The **MainActivity.java** file will contain the java code to handle all the activities related to our app.

Following is the default code of **MainActivity.java** file which is generated by our [HelloWorld application](https://www.tutlane.com/tutorial/android/android-hello-world-app-example).

package com.sarker.helloworld;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
    }  
}

**Android Manifest File (AndroidManifest.xml)**

Generally our application will contain multiple [activities](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) and we need define all those [activities](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) in **AndroidManifest.xml**file. In our manifest file we need to mention the main activity for our app using **MAIN** action and **LAUNCHER** category attributes in **intent filters** (<intent-filter>). In case if we didn’t mention MAIN action or LAUNCHER category for main activity, our app icon will not appear in home screen’s list of apps.

Following is the default code of **AndroidManifest.xml** file which is generated by our **HelloWorld** application.

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.sarker.helloworld" >  
  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic\_launcher"  
        android:label="@string/app\_name"  
        android:roundIcon="@mipmap/ic\_launcher\_round"  
        android:supportsRtl="true"  
        android:theme="@style/AppTheme" >  
        <activity android:name=".MainActivity" >  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>

These are the main folders and files required to implement an application in android studio. If you want to see the actual file structure of the project, select **Project** from the **Project** dropdown instead of **Android**.