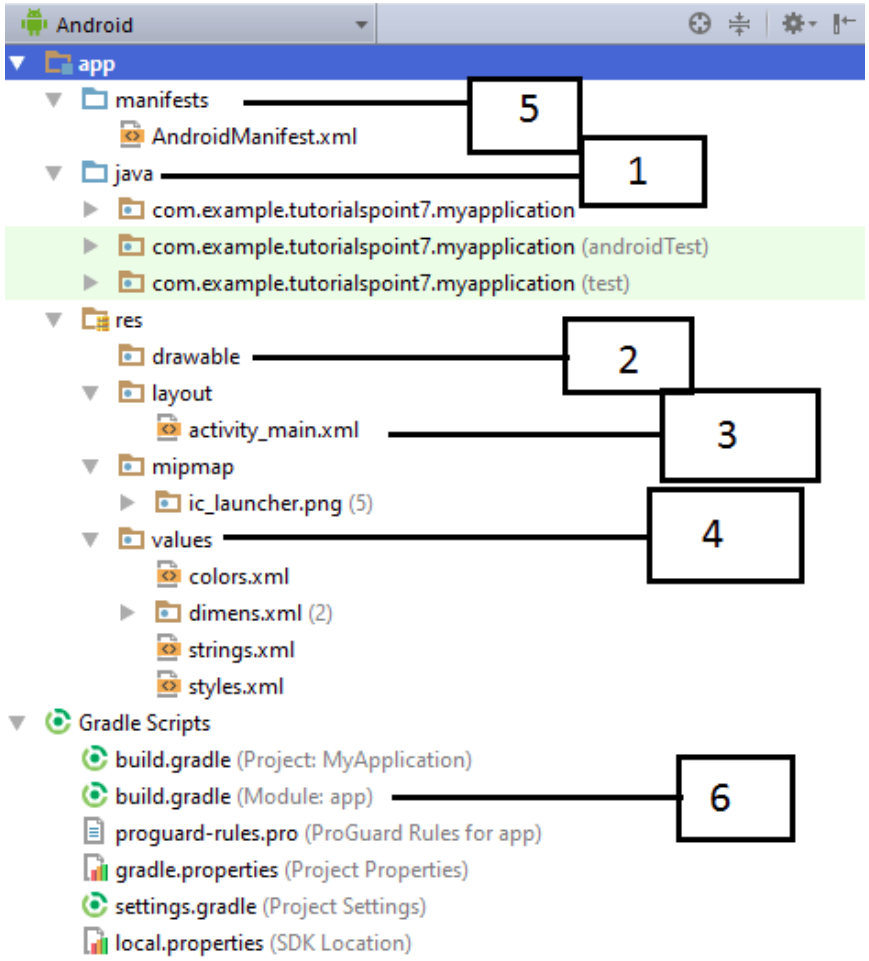
Android HelloWorld - Example

Tushar B. Kute,
http://tusharkute.com

Anatomy of Android Application



Anatomy of Android Application

- Java
  - This contains the .java source files for your project. By default, it includes an MainActivity.java source file having an activity class that runs when your app is launched using the app icon.
- res/drawable-hdpi
  - This is a directory for drawable objects that are designed for high-density screens.
- res/layout
  - This is a directory for files that define your app's user interface.

Anatomy of Android Application

- res/values
  - This is a directory for other various XML files that contain a collection of resources, such as strings and colours definitions.
- AndroidManifest.xml
  - This is the manifest file which describes the fundamental characteristics of the app and defines each of its components.
- Build.gradle
  - This is an auto generated file which contains compileSdkVersion, buildToolsVersion, applicationId, minSdkVersion, targetSdkVersion, versionCode and versionName

The Main Activity file

- The main activity code is a Java file MainActivity.java.
- This is the actual application file which ultimately gets converted to a Dalvik executable and runs your application.
- Here, R.layout.activity_main refers to the activity_main.xml file located in the res/layout folder.
- The onCreate() method is one of many methods that are figured when an activity is loaded.

The Manifest file

- Whatever component you develop as a part of your application, you must declare all its components in a manifest.xml which resides at the root of the application project directory.
- This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS.
- Following is the list of tags which you will use in your manifest file to specify different Android application components –
  - <activity>elements for activities
  - <service> elements for services
  - <receiver> elements for broadcast receivers
  - <provider> elements for content providers

The Strings file

- The strings.xml file is located in the res/values folder and it contains all the text that your application uses.
- For example, the names of buttons, labels, default text, and similar types of strings go into this file. This file is responsible for their textual content.

The Layout file

- The activity_main.xml is a layout file available in res/layout directory, that is referenced by your application when building its interface.
- You will modify this file very frequently to change the layout of your application.

The Android resources

- There are many more items which you use to build a good Android application.
- Apart from coding for the application, you take care of various other resources like static content that your code uses, such as bitmaps, colors, layout definitions, user interface strings, animation instructions, and more.
- These resources are always maintained separately in various sub-directories under res/ directory of the project.

The Android resources

- color/
  - XML files that define a state list of colors. They are saved in res/color/ and accessed from the R.color class.
- drawable/
  - Image files like .png, .jpg, .gif or XML files that are compiled into bitmaps, state lists, shapes, animation drawable. They are saved in res/drawable/ and accessed from the R.drawable class.
- layout/
  - XML files that define a user interface layout. They are saved in res/layout/ and accessed from the R.layout class.

Accessing resources in Code

- When your Android application is compiled, a R class gets generated, which contains resource IDs for all the resources available in your res/ directory.
- You can use R class to access that resource using sub-directory and resource name or directly resource ID.
- Example:

  To access res/drawable/myimage.png and set an ImageView you will use following code –

- ```
  ImageView imageView = (ImageView) findViewById(R.id.myimageview);
  imageView.setImageResource(R.drawable.myimage);
  ```
  ```

Accessing resources in Code

- Consider next example where res/values/strings.xml has following definition –

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string  name="hello">Hello, World!</string>
</resources>
```

- Now you can set the text on a TextView object with ID msg using a resource ID as follows –

```
TextView msgTextView = (TextView)
findViewById(R.id.msg);
msgTextView.setText(R.string.hello);
```

Accessing resources in Code

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="vertical" >
   <TextView android:id="@+id/text"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello, I am a TextView" />
   <Button android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Hello, I am a Button" />
</LinearLayout>
```

- This application code will load this layout for an Activity, in the onCreate() method as follows:

```
public void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.main_activity);
}
```

Accessing resources in Code

- Consider the following resource XML res/values/strings.xml file that includes a color resource and a string resource –

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="opaque_red">#f00</color>
    <string name="hello">Hello!</string>
</resources>
```

- Now you can use these resources in the following layout file to set the text color and text string as follows –

```
<?xml version="1.0" encoding="utf-8"?>
<EditText
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:textColor="@color/opaque_red"
    android:text="@string/hello" />
```

Thank you

This presentation is created using LibreOffice Impress 4.2.8.2, can be used freely as per GNU General Public License

Web Resources
http://mitu.co.in
http://tusharkute.com

Blogs
http://digitallocha.blogspot.in
http://kyamputar.blogspot.in

tushar@tusharkute.com