

Android Programming with Passion...

Chapter I : Introduction to Android

Karim DJAAFAR
JEE Architect Evangelist

Last Version date :

February 2019

Current version of Android Studio 3.3.1



AGENDA

- Android Overview
- Android Architecture
- Installation and Configuration
- Building Blocks of Android application



OBJECTIVES OF THIS COURSE

- Understand Android Architecture and main components



Android Overview



What is Android ?

- Android is Google's open source and free Java-based platform for mobile development
- It enables developers to build real-world mobile applications using the Android software development kit (SDK) and publish them to the Android market
- Android comes with several application programming interfaces (APIs) that make the task of developing full-featured applications easy (You can even use a camera, accelerometer, or GPS in an Android application)



What is Android ?

- Using Android, you can develop applications for a wide variety of devices, including phones, e-book readers, netbooks, and GPS units
- Android was initially developed by Android, Inc., a small Palo Alto, California, company
- Google bought this company in July 2005 and released the Android SDK in November 2007
- Periodically, Google releases Android SDK updates, the current release is **Android 9 (API Level 28)**



Android 9 Features

- Android 9 (API level 28) introduces great new features and capabilities for users and developers
 - Indoor positioning with Wi-Fi RTT
 - Display cutout support for cameras and speakers
 - Android 9 introduces several enhancements to notifications, all of which are available to developers targeting API level 28 and above
 - Enhanced messaging experience
 - Channel settings, broadcasts, and Do Not Disturb
 - Multi-camera support and camera updates
 - *ImageDecoder* for drawables and bitmaps
 - Animation
 - HDR VP9 Video, HEIF image compression, and Media APIs
 - ...



Android Studio

- In order to write an Android application, we are going to need a development environment
- Google has made a very useful tool for all Android Developers, the Android Studio :
 - Android Studio is the official IDE for Android development, and with a single download includes everything you need to begin developing Android apps
 - Included in the download kit, are the **Software Development Kit (SDK)**, with all the Android libraries we may need, and the infrastructure to download the many Android emulator instances, so that we can initially run our application, without needing a real device

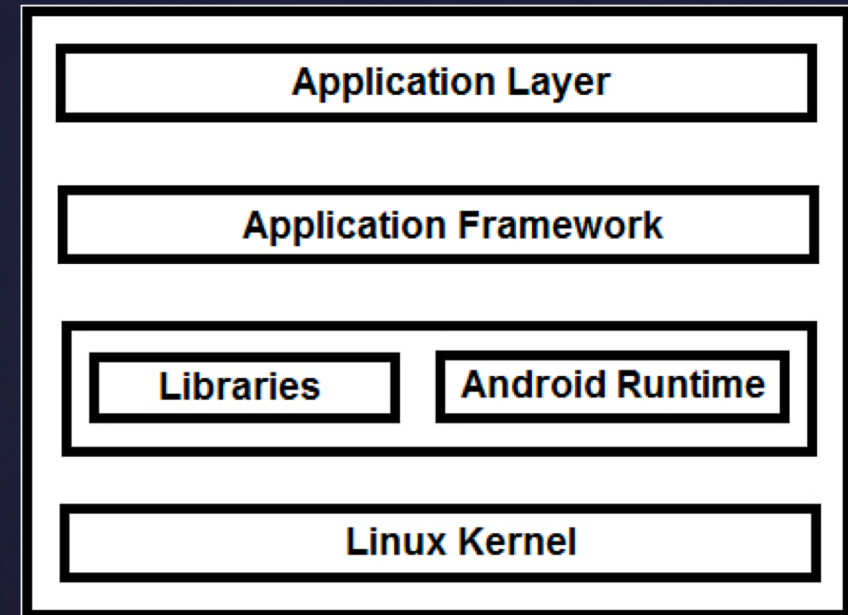


Android Architecture



Android software stack

- The Android software stack consists of a Linux kernel and a collection of C/C++ libraries that are exposed through an application framework for application development
- The Android software stack consists of four main layers, as shown in figure below



Linux Kernel

- **Linux kernel**—The kernel on which Android is based contains device drivers for various hardware components of an Android device, including Display, Camera, Keypad, Wi-Fi, Flash Memory, and Audio



Libraries

- The next layer on top of the Linux kernel is the libraries that implement different Android features. A few of these libraries are listed here:
 - **WebKit library**—Responsible for browser support.
 - **FreeType library**—Responsible for font support.
 - **SQLite library**—Provides database support.
 - **Media libraries**—Responsible for recording and playback of audio and video formats.
 - **Surface Manager library**—Provides graphics libraries that include SGL and OpenGL for 2D and 3D graphics support.



Android runtime

- The engine at the same layer as the libraries
- It provides a set of core Android libraries and a ***Dalvik*** virtual machine that enable developers to write Android applications using Java
- The core Android libraries provide most of the functionality available in the core Java libraries, as well as the Android-specific libraries



Application Framework

- Provides the classes that enable application developers to develop Android applications
- It manages the user interface, application resources, and abstraction for hardware access

Application Layer

- Displays the application developed and downloaded by users, along with the built-in applications provided with the Android device itself

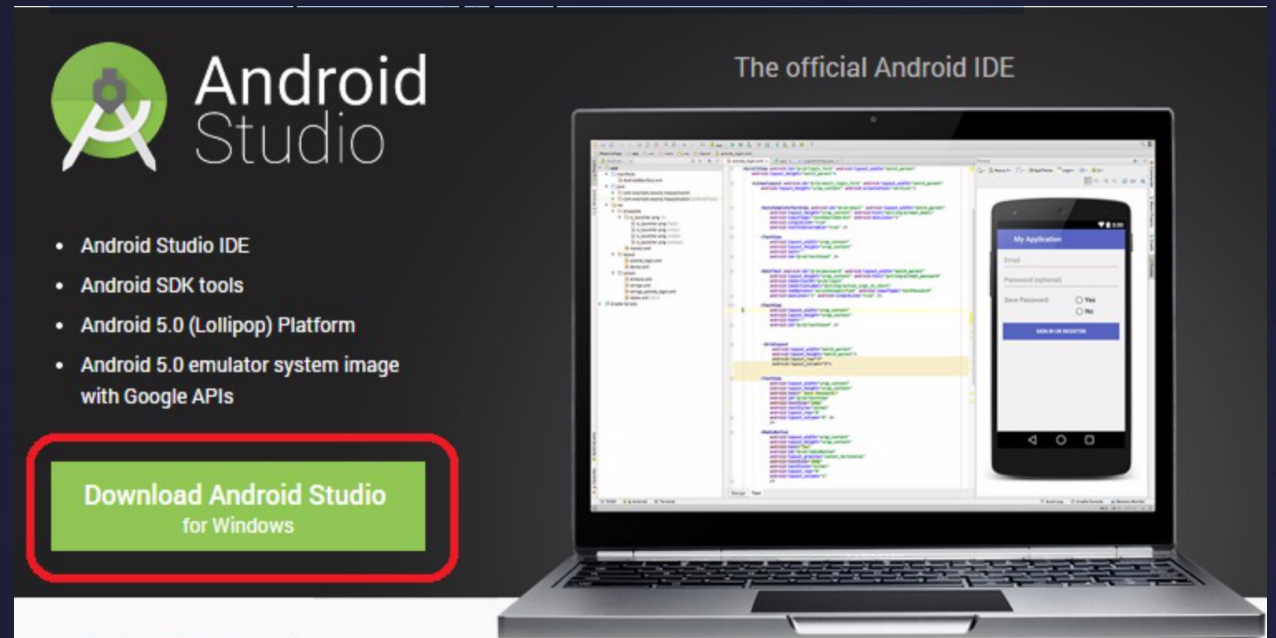


Android Installation and Configuration



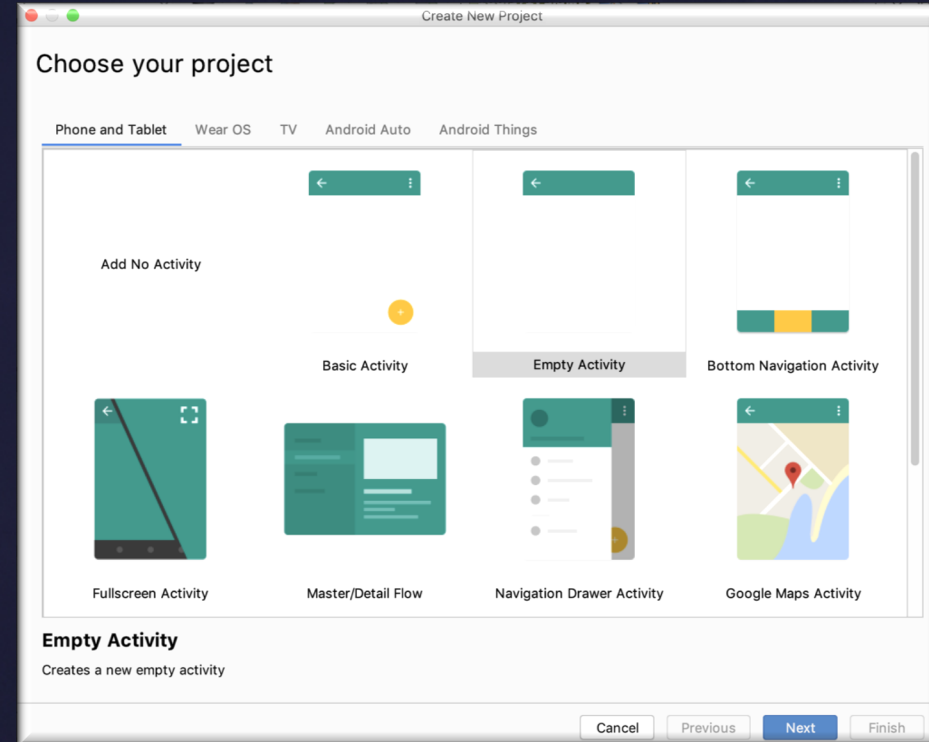
Software needed

- JDK (Java Development Kit)
 - Java language to build Android application
- Android Studio
 - IDE, SDK, Libraries, samples, documentation, emulators, debugger, command line tool



Creating Project

- Automatic creation of building blocks of an Android application
 - Starting Activity class
 - Layout resource file
 - AndroidManifest.xml
 - strings.xml
 - Android library



Lab 1:

Installing and Configuring the Android



Getting Started Android: Building Blocks of Android application



Android Project Structure

- Before we try to make our first Android application, we should first see the basic parts of an Android application project, in order to recognize them and be able to understand them better :
 - **Activities** The **Activities** are the main Java classes, that contain the Android code with which we are going to develop, what do we want the application to do
 - **Layouts** The Layouts are the main xml files, that contain the Android xml code with which we are going to develop, how will our application views look like



Android Project Structure

- **Values**

- Animation Resources
- Color State List Resource
- Drawable Resources
- Layout Resource
- Menu Resource
- String Resources
- Style Resource



Android Project Structure

- **Drawables** A drawable resource is a general concept for a graphic that can be drawn to the screen
- There are several different types of ***drawables*** , we will see them in some labs
 - Bitmap File
 - Nine-Patch File
 - Layer List
 - ...



Android Studio App Components

- AndroidManifest.xml
- Activity class
- Resource files
 - Layout
 - strings.xml, etc..
- Android library (automatically configured)

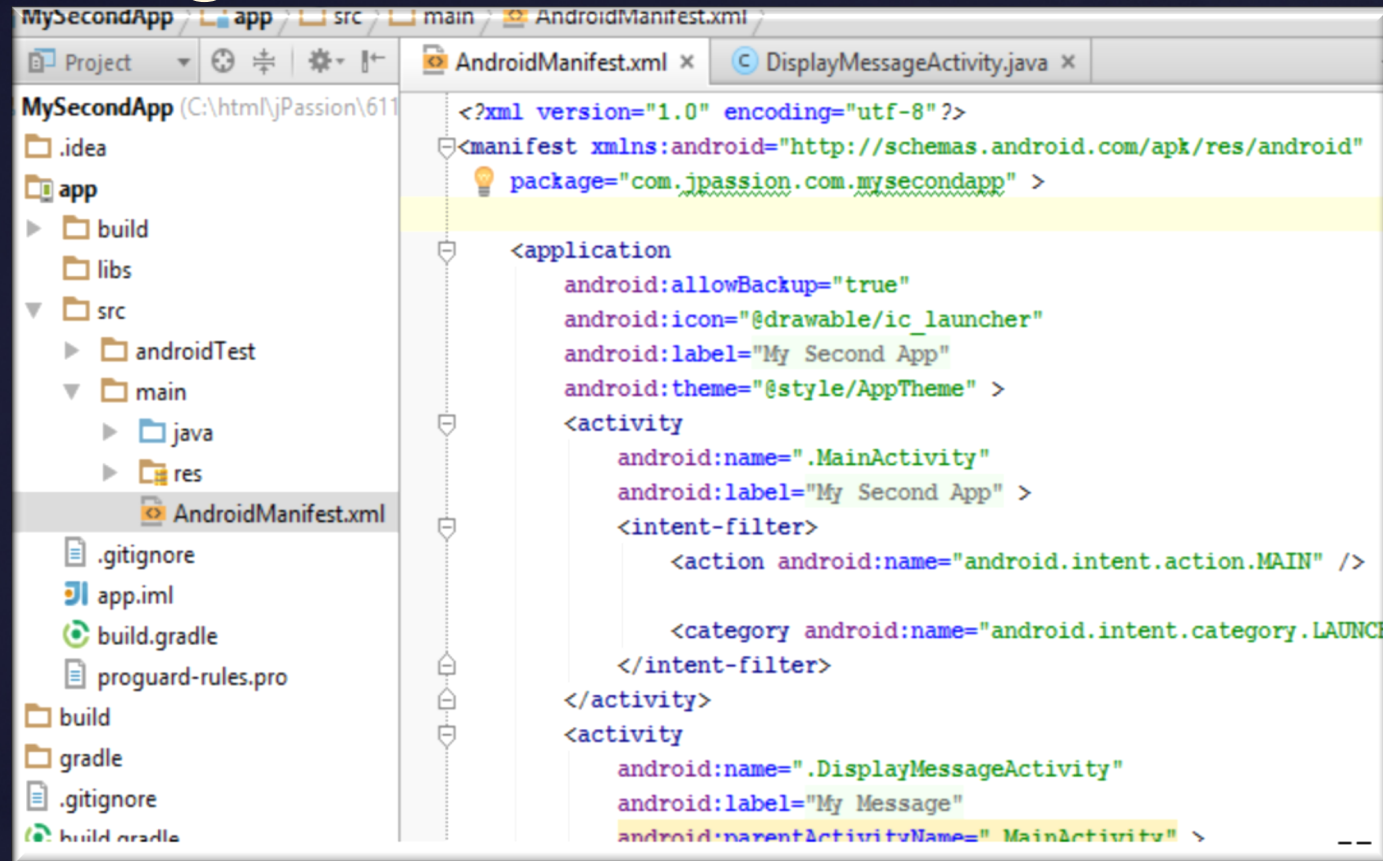


AndroidManifest.xml file

- Java package for the application
 - The package name serves as a unique identifier for the application
- Application name and icon
- Application version information
- Activities
 - One activity is designated as a starting Activity
- Every application must have a manifest file called *AndroidManifest.xml* file (with precisely that name) in its root directory
- The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code



Building Block AndroidManifest.xml



Activity class : Quick Overview

- Each **Activity** class typically represents a screen
 - Like a JSP page in a Web application
- The *onCreate()* method of Activity class gets called by the Android system when your Activity starts
 - You create your screen UI inside *onCreate()* method
- Every Activity has to be described in the *AndroidManifest.xml* file
- An Activity is typically chosen as a starting one of your application - like a class that has a *main()* method in Java
 - Through special configuration in *AndroidManifest.xml*

•



Main Activity Class

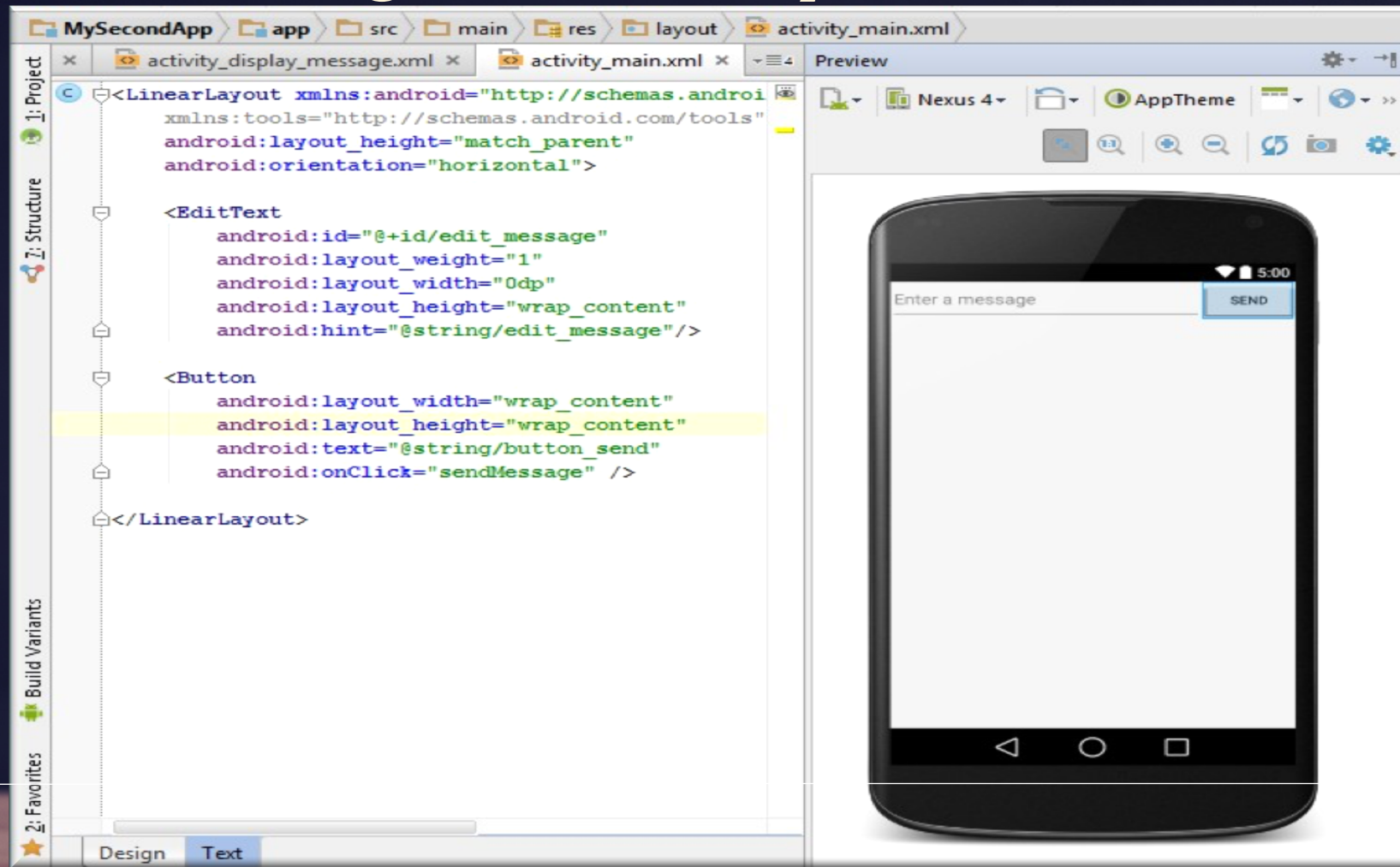
```
package com.jp passion.com.myfirstapp;  
Import ...  
public class MainActivity extends ActionBarActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) (...)  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) (...)  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item) (...)
```


Layout resource files

- Every screen has a corresponding layout resource file
 - Unless you create screen UI programmatically
- Each activity class specifies which layout resource file to use for each screen it represents
 - **setContentView(R.layout.activity_main);**
- Located under */res/layout* directory
 - */res/layout/main.xml* layout resource file is referred to as **R.layout.activity_main**
- **activity_main** is defined at create wizards and user can set name



Building block layout resource

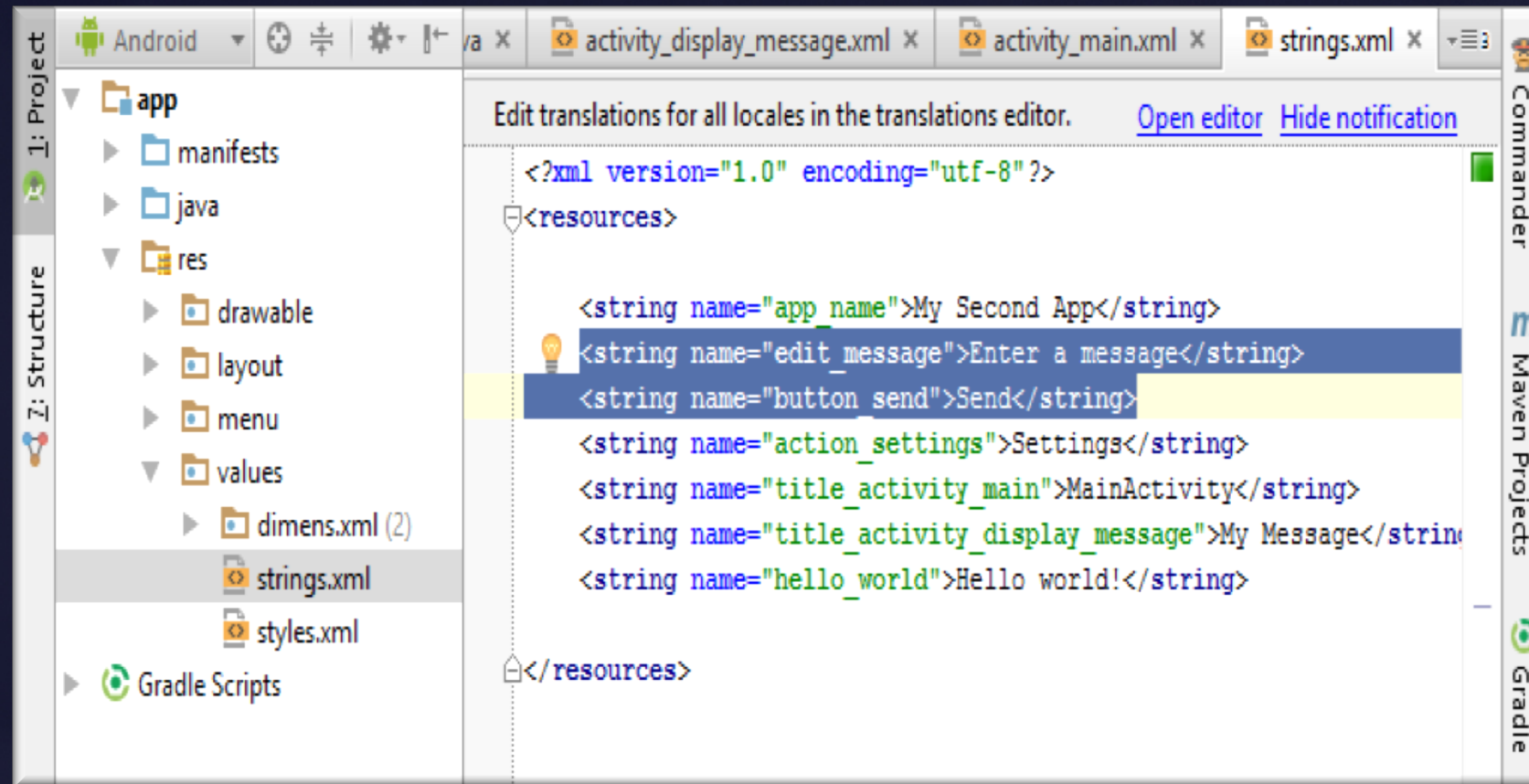


Resource files : strings.xml

- Let you define the text strings of your applications in a well-known file
 - Rather than in Java code
 - Rather than in Layout resource files
- The strings are then referred to through the names assigned to them
 - `@string/edit_message` (in the layout resource file)
 - `R.string.edit_message` (in the Java code)
- Located under `/res/values` directory



Building block values resource



Lab 2:

Android steps by steps



In Summary



In this module we saw all the tools and frameworks to set up our Android environment



Your friend to teach you
JPassion.com
With Passion !