Getting Started (Initial Setup)

1. Android Studio Installation - Windows

a. You can find the installer at

http://developer.android.com/sdk/installing/index.html?pkg=studio

b. If you downloaded an .exe file (recommended), double-click to launch it.

If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio > bin folder and launch studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines).

c. Follow the setup wizard in Android Studio and install any SDK packages that it recommends.

Not mandatory (Only for systems with the error)

d. On some Windows systems, the launcher script does not find where Java is installed. If you encounter this problem, you need to set an environment variable indicating the correct location.

Select Start menu > Computer System Properties >;

Advanced System Properties. Then open Advanced tab > Environment Variables and add a new system

variable JAVA_HOME that points to your JDK folder, for example: C:\Program Files\Java\jdk1.7.0_21

2. Android Studio Installation - Mac

a. You can find the installer at

https://developer.android.com/studio

b. Launch the Android Studio DMG file.

c. Drag and drop Android Studio into the Applications folder, then launch Android Studio.

d. Select whether you want to import previous Android Studio settings, then click OK.

e. The Android Studio Setup Wizard guides you through the rest of the setup, which includes downloading Android SDK components that are required for development.

3. Android Studio Installation - Linux

a. You can find the installer at

https://developer.android.com/studio

b. Unpack the .zip file you downloaded to an appropriate location for your applications, such as within /usr/local/ for your user profile, or /opt/ for shared users.

c. If you're using a 64-bit version of Linux, make sure you first install the required libraries for 64-bit machines.

d. To launch Android Studio, open a terminal, navigate to the android-studio/bin/ directory, and execute studio.sh.

e. Select whether you want to import previous Android Studio settings or not, then click OK.

f. The Android Studio Setup Wizard guides you through the rest of the setup, which includes downloading Android SDK components that are required for development.

Required libraries for 64-bit machines

g. If you are running a 64-bit version of Ubuntu, you need to install some 32-bit libraries with the following command:

```
sudo apt-get install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1
libbz2-1.0:i386
```

h. If you are running 64-bit Fedora, the command is:

```
sudo yum install zlib.i686 ncurses-libs.i686 bzip2-libs.i686
```

4. **SDK Manager (Only for custom installation)**

a. On the top panel you will see two android specific options. Hover over them and click on the SDK manager. Here are different components that you should install from the SDK manager.

b. SDK Tools: These are some tools required for doing Android development. This includes tools like SDK manager and Virtual device manager.

c. SDK Platform Tools: Important tools like adb that are required for debugging a lot of issues related to android devices and software.

d. SDK Build Tools: These tools are required to build, run and test android applications.

5. ***In the SDK manager you will see different installable for different Android API levels. Here are the different packages that you find inside each API Level.***

a. SDK Platform: This is the main package that you need to test your app against a particular api level.

b. System Images: You will find multiple system images for each api. These images are used to emulate a specific API level on the emulator machine. You would need to install this along with the platform if you want to test your code using an emulator for a particular android api level.

c. Sources for SDK: This has the source code for the particular SDK.

d. Google API's: You will have to install these in case you want your emulator to have google apps like Google maps etc.

e. Samples: Includes sample code files showing example use of different api's from that API level.

6. ***Extras***

a. Android Support Library: This includes some additional API's and it also provides backward compatibility by providing support for newer api's in older Android releases.

b. Google Play Services: Allows your app to access Google maps API, Google Wallet etc.

c. HAXM (Hardware Accelerated Execution Manager): is an engine created by Intel to accelerate the Android emulation. This tool leads to a considerable improvement in the speed of the emulator and is the prime factor why we should prefer intel based system images.

d. Android API Specific Packages: In the SDK manager you will see different installable for different Android API levels. Here are the different packages that you find inside each API Level.