

Android Module

Android is an operating system based on Linux with a Java programming interface for mobile devices such as Smartphone (Touch Screen Devices who supports Android OS) as well for Tablets too.

Android was developed by the **Open Handset Alliance** (OHA), which is led by Google, and other companies like Samsung, Sony, Infinix, Huawei, Google, Techno and many more.

In **Android Module we will** cover complete topics from basic to advanced level those are

- 1.[Android Architecture](#)
- 2.[Android Environment Setup](#)
- 3.[Android Services](#)
- 4.[Android Fragments](#)
- 5.[Android Intents and Filters](#)
- 6.[Android UI Controls](#)
- 7.[Android UI Themes](#)
- 8.[Android Notifications](#)
- 9.[Google Maps](#)
- 10.[Accessing APIs](#)
- 11.[Android Database Connection](#)
- 12.[Accessing Files](#)
- 13.[Android Video/Audio](#)
- 14.[Android GPS](#)
- 15.[Android Animations](#) ... and many more topics in Android

Audience

The android tutorial is targeted for beginners as well as experienced people who wish to advance their skill in Mobile App development. Knowledge in Kotlin Programming language is required and any other server side language I.e python, php etc.

Android Features

Android is a powerful open-source operating system which provides a lot of great features, those are

- It's open-source and we can customize the OS based on our requirements.
- It supports connectivity for GSM, CDMA, WIFI, NFC, Bluetooth, etc. for telephony or data transfer. It will allow us to make or receive a calls / SMS messages and we can send or retrieve data across mobile networks
- By using WIFI technology we can pair with other devices using apps
- Android has multiple APIs to support location-based services such as GPS
- We can perform all data storage related activities by using lightweight database [SQLite](#), Room etc
- It has a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video and having a different image format like JPEG, PNG, GIF, BMP, MP3, etc.
- It has extensive support for multimedia hardware control to perform playback or recording using camera and microphone
- It has an integrated open-source WebKit layout based web browser to support HTML5, CSS3
- It supports a multi-tasking, we can move from one task window to another and multiple applications can run simultaneously
- It will give a chance to reuse the application components and the replacement of native applications.
- We can access the hardware components like Camera, GPS, and Accelerometer
- It has support for 2D/3D Graphics
- Include gestures
- Finger and Eye detection
- and many more ..

SDK Levels

<https://source.android.com/setup/start/build-numbers>

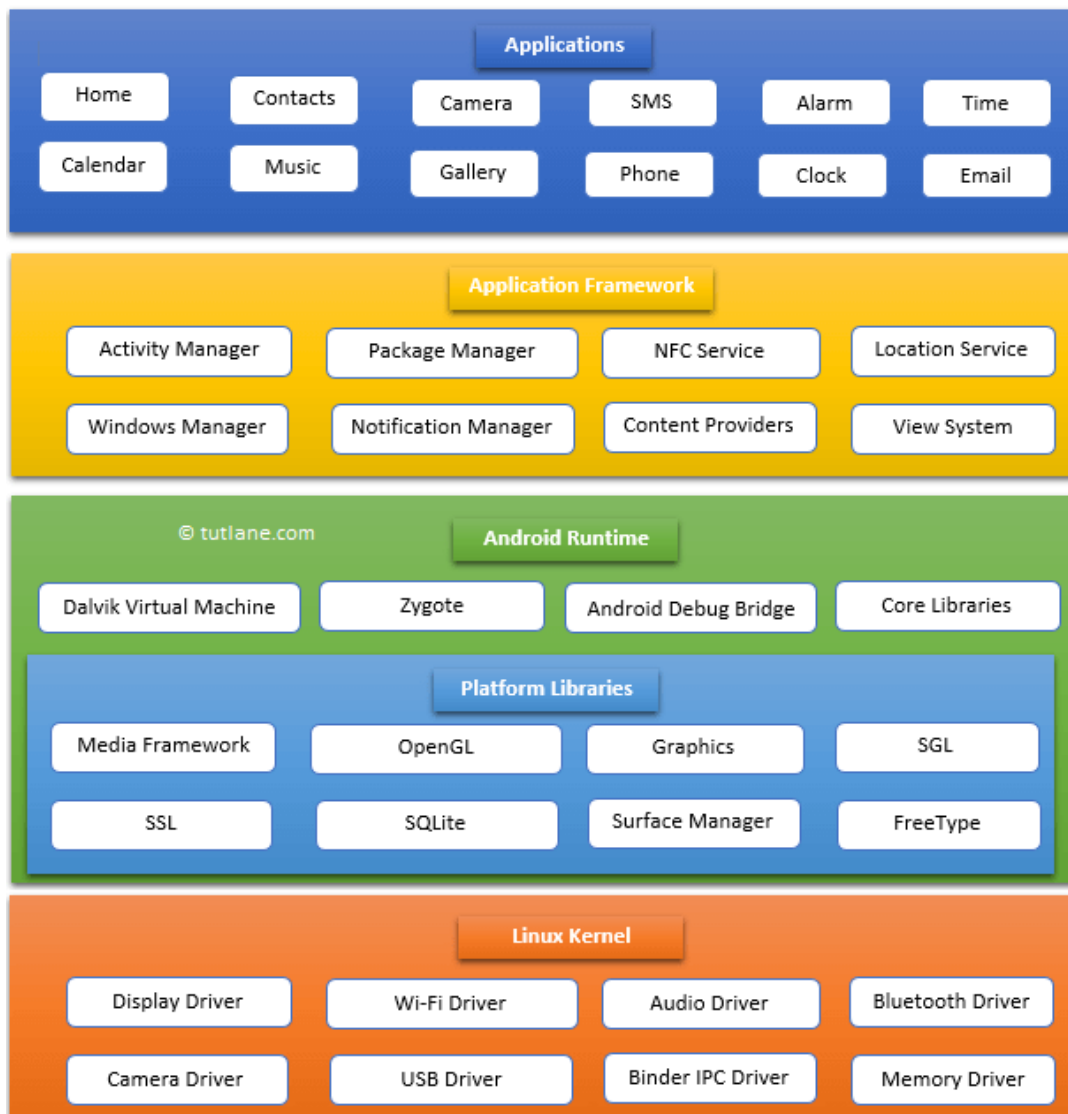
Android architecture is a software stack of components to support mobile device needs. Android software stack contains a Linux Kernel, collection of c/c++ libraries which are exposed through an application framework services, runtime, and application.

Following are main components of android architecture those are

- 1.Applications
- 2.Android Framework
- 3.Android Runtime
- 4.Platform Libraries
- 5.Linux Kernel

In these components, the **Linux Kernel** is the main component in android to provide its operating system functions to mobile and **Dalvik Virtual Machine (DVM)** which is responsible for running a mobile application.

Following is the pictorial representation of android architecture with different components.



Applications

The top layer of the android architecture is **Applications**. The native and third-party applications like contacts, email, music, gallery, clock, games, etc. whatever we will build those will be installed on this layer only.

The application layer runs within the Android run time using the classes and services made available from the application framework.

Application Framework

The **Application Framework** provides the classes used to create Android applications. It also provides a generic abstraction for hardware access and manages the user interface and application resources. It basically provides the services through which we can create a particular class and make that class helpful for the Application creation.

The application framework includes services like telephony service, location services, notification manager, NFC service, view system, etc. which we can use for application development as per our requirements.

Android Runtime

Android Runtime environment is an important part of Android rather than an internal part and it contains components like **core libraries** and the **Dalvik virtual machine**. The Android run time is the engine that powers our applications along with the libraries and it forms the basis for the application framework.

Dalvik Virtual Machine (DVM) is a register-based virtual machine like Java Virtual Machine (JVM). It is specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It relies on the Linux kernel for threading and low-level memory management.

The **core libraries** in android runtime will enable us to implement android applications using standard JAVA programming language.

Platform Libraries

The **Platform Libraries** includes various C/C++ core libraries and Java-based libraries such as SSL, libc, Graphics, SQLite, Webkit, Media, Surface Manger, OpenGL, etc. to provide support for Android development.

The following are the summary details of some core android libraries available for android development.

- Media library for playing and recording audio and video formats
- The Surface manager library to provide a display management
- SGL and OpenGL Graphics libraries for 2D and 3D graphics
- SQLite is for database support and FreeType for font support
- Web-Kit for web browser support and SSL for Internet security.

Linux Kernel

Linux Kernel is a bottom layer and heart of the android architecture. It manages all the drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are mainly required for the android device during the runtime.

The Linux Kernel will provide an abstraction layer between the device hardware and the remainder of the stack. It is responsible for memory management, power management, device management, resource access, etc.