**Chapter3: DESCRIPTION OF TOOL**

# 3.1 Android Studio

Android Studio is the official [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Google](https://en.wikipedia.org/wiki/Google)'s [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [operating system](https://en.wikipedia.org/wiki/Operating_system), built on [JetBrains](https://en.wikipedia.org/wiki/JetBrains)' [IntelliJ IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA) software and designed specifically for [Android development](https://en.wikipedia.org/wiki/Android_software_development). It is available for download on [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) based operating systems. It is a replacement for the [Eclipse Android Development Tools](https://en.wikipedia.org/wiki/Eclipse_(software)#Android_Development_Tools) (ADT) as primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the [Google I/O](https://en.wikipedia.org/wiki/Google_I/O) conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.1 released in March 2018.[1]

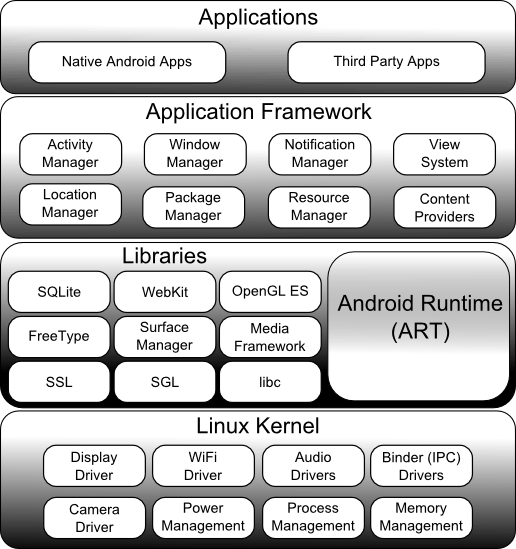
The following features are provided in the current stable version:

* [Gradle](https://en.wikipedia.org/wiki/Gradle)-based build support
* Android-specific [refactoring](https://en.wikipedia.org/wiki/Code_refactoring) and quick fixes
* [Lint](https://en.wikipedia.org/wiki/Lint_(software)) tools to catch performance, usability, version compatibility and other problems
* [ProGuard](https://en.wikipedia.org/wiki/ProGuard_(software)) integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allows users to drag-and-drop UI components, option to [preview layouts](https://en.wikipedia.org/wiki/WYSIWYG) on multiple screen configurations
* Support for building [Android Wear](https://en.wikipedia.org/wiki/Android_Wear) apps
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
* Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Android Studio supports all the same programming languages of [IntelliJ](https://en.wikipedia.org/wiki/IntelliJ), and [PyCharm](https://en.wikipedia.org/wiki/PyCharm) e.g. [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), and [Kotlin](https://en.wikipedia.org/wiki/Kotlin_(programming_language)); and Android Studio 3.0 supports "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects [backport](https://en.wikipedia.org/wiki/Backporting) some Java 9 features.

**Android Architecture:**

Android is structured in the form of a software stack comprising applications, an operating system, run-time environment, middleware, services and libraries. This architecture can, perhaps, best be represented visually as outlined in Figure 8-1. Each layer of the stack, and the corresponding elements within each layer, are tightly integrated and carefully tuned to provide the optimal application development and execution environment for mobile devices.



3.1.1 Android Architecture

## Android Runtime - ART

When an Android app is built within Android Studio it is compiled into an intermediate bytecode format (referred to as DEX format). When the application is subsequently loaded onto the device, the Android Runtime (ART) uses a process referred to as Ahead-of-Time (AOT) compilation to translate the bytecode down to the native instructions required by the device processor. This format is known as Executable and Linkable Format (ELF).

Each time the application is subsequently launched, the ELF executable version is run, resulting in faster application performance and improved battery life.

This contrasts with the Just-in-Time (JIT) compilation approach used in older Android implementations whereby the bytecode was translated within a virtual machine (VM) each time the application was launched.

## Android Libraries

In addition to a set of standard Java development libraries (providing support for such general purpose tasks as string handling, networking and file manipulation), the Android development environment also includes the Android Libraries. These are a set of Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access.

A summary of some key core Android libraries available to the Android developer is as follows:

* android.app – Provides access to the application model and is the cornerstone of all Android applications.
* android.content – Facilitates content access, publishing and messaging between applications and application components.
* android.database – Used to access data published by content providers and includes SQLite database management classes.

* android.graphics – A low-level 2D graphics drawing API including colors, points, filters, rectangles and canvases.
* android.hardware – Presents an API providing access to hardware such as the accelerometer and light sensor.
* android.opengl – A Java interface to the OpenGL ES 3D graphics rendering API.
* android.os – Provides applications with access to standard operating system services including messages, system services and inter-process communication.
* android.media – Provides classes to enable playback of audio and video.
* android.net – A set of APIs providing access to the network stack. Includes android.net.wifi, which provides access to the device’s wireless stack.
* android.print – Includes a set of classes that enables content to be sent to configured printers from within Android applications.
* android.provider – A set of convenience classes that provide access to standard Android content provider databases such as those maintained by the calendar and contact applications.
* android.text – Used to render and manipulate text on a device display.
* android.util – A set of utility classes for performing tasks such as string and number conversion, XML handling and date and time manipulation.
* android.view – The fundamental building blocks of application user interfaces.
* android.widget - A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
* android.webkit – A set of classes intended to allow web-browsing capabilities to be built into applications.

## Application Framework

The Application Framework is a set of services that collectively form the environment in which Android applications run and are managed. This framework implements the concept that Android applications are constructed from reusable, interchangeable and replaceable components. This concept is taken a step further in that an application is also able to publish its capabilities along with any corresponding data so that they can be found and reused by other applications.

The Android framework includes the following key services:

* Activity Manager – Controls all aspects of the application lifecycle and activity stack.
* Content Providers – Allows applications to publish and share data with other applications.
* Resource Manager – Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
* Notifications Manager – Allows applications to display alerts and notifications to the user.
* View System – An extensible set of views used to create application user interfaces.
* Package Manager – The system by which applications are able to find out information about other applications currently installed on the device.
* Telephony Manager – Provides information to the application about the telelphony services available on the device such as status and subscriber information.
* Location Manager – Provides access to the location services allowing an application to receive updates about location changes.[2]

**3.2 JDK(Java Development kit)**

The Java Development Kit (JDK) is an implementation of either one of the [Java Platform, Standard Edition](https://en.wikipedia.org/wiki/Java_Platform,_Standard_Edition), [Java Platform, Enterprise Edition](https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition), or [Java Platform, Micro Edition](https://en.wikipedia.org/wiki/Java_Platform,_Micro_Edition) platformsreleased by [Oracle Corporation](https://en.wikipedia.org/wiki/Oracle_Corporation) in the form of a binary product aimed at [Java](https://en.wikipedia.org/wiki/Java_(programming_language))developers on [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)), [Linux](https://en.wikipedia.org/wiki/Linux), [macOS](https://en.wikipedia.org/wiki/MacOS) or [Windows](https://en.wikipedia.org/wiki/Windows). The JDK includes a private JVM and a few other resources to finish the development of a Java Application. Since the introduction of the [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) platform, it has been by far the most widely used Software Development Kit ([SDK](https://en.wikipedia.org/wiki/Software_development_kit)). On 17 November 2006, Sun announced that they would release it under the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) (GPL), thus making it [free software](https://en.wikipedia.org/wiki/Free_software). This happened in large part on 8 May 2007, when Sun contributed the source code to the [OpenJDK](https://en.wikipedia.org/wiki/OpenJDK).[5]

The JDK has as its primary components a collection of programming tools, including:

* [appletviewer](https://en.wikipedia.org/wiki/AppletViewer) – this tool can be used to run and debug Java applets without a web browser
* apt – the [annotation-processing tool](https://en.wikipedia.org/wiki/Metadata_facility_for_Java)[[4]](https://en.wikipedia.org/wiki/Java_Development_Kit#cite_note-apt-4)
* extcheck – a utility that detects JAR file conflicts
* idlj – the IDL-to-Java compiler. This utility generates Java [bindings](https://en.wikipedia.org/wiki/Language_binding) from a given [Java IDL](https://en.wikipedia.org/wiki/Java_Interface_Definition_Language) file.
* jabswitch – the [Java Access Bridge](https://en.wikipedia.org/wiki/Java_Access_Bridge). Exposes assistive technologies on Microsoft Windows systems.
* java – the [loader](https://en.wikipedia.org/wiki/Loader_(computing)) for Java applications. This tool is an interpreter and can interpret the class files generated by the [javac](https://en.wikipedia.org/wiki/Javac) compiler. Now a single launcher is used for both development and deployment. The old deployment launcher, jre, no longer comes with Sun JDK, and instead it has been replaced by this new java loader.
* [javac](https://en.wikipedia.org/wiki/Javac) – the [Java compiler](https://en.wikipedia.org/wiki/Java_compiler), which converts source code into [Java bytecode](https://en.wikipedia.org/wiki/Java_bytecode)
* [javadoc](https://en.wikipedia.org/wiki/Javadoc) – the documentation generator, which automatically generates documentation from [source code](https://en.wikipedia.org/wiki/Source_code) comments
* [jar](https://en.wikipedia.org/wiki/JAR_(file_format)#Extraction) – the archiver, which packages related class [libraries](https://en.wikipedia.org/wiki/Library_(computer_science)) into a single [JAR file](https://en.wikipedia.org/wiki/Jar_(file_format)). This tool also helps manage JAR files.
* javafxpackager – tool to package and sign JavaFX applications
* jarsigner – the jar signing and verification tool
* javah – the C header and stub generator, used to write native methods
* javap – the class file [disassembler](https://en.wikipedia.org/wiki/Disassembler)
* javaws – the [Java Web Start](https://en.wikipedia.org/wiki/Java_Web_Start) launcher for JNLP applications
* [JConsole](https://en.wikipedia.org/wiki/JConsole) – Java Monitoring and Management Console
* jdb – the [debugger](https://en.wikipedia.org/wiki/Debugger)
* jhat – Java Heap Analysis Tool (experimental)
* jinfo – This utility gets configuration information from a running Java process or crash dump. (experimental)
* jmap [Oracle jmap - Memory Map](http://docs.oracle.com/javase/7/docs/technotes/tools/share/jmap.html)– This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump. (experimental)
* jmc – Java Mission Control
* jps – Java Virtual Machine Process Status Tool lists the instrumented HotSpot Java Virtual Machines (JVMs) on the target system. (experimental)
* jrunscript – Java command-line [script](https://en.wikipedia.org/wiki/Shell_script) [shell](https://en.wikipedia.org/wiki/Shell_(computing)).

**3.3 SDK(Software Development Kit)**

A software development kit (SDK or devkit) is typically a set of [software development](https://en.wikipedia.org/wiki/Software_development) tools that allows the creation of [applications](https://en.wikipedia.org/wiki/Application_software) for a certain [software](https://en.wikipedia.org/wiki/Software) package, [software framework](https://en.wikipedia.org/wiki/Software_framework), hardware platform, [computer system](https://en.wikipedia.org/wiki/Computer_system), [video game console](https://en.wikipedia.org/wiki/Video_game_console), [operating system](https://en.wikipedia.org/wiki/Operating_system), or similar development platform. To enrich applications with advanced functionalities, advertisements, push notifications and more, most app developers implement specific software development kits. Some SDKs are critical for developing a platform-specific app. For example, the development of an [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) app on [Java](https://en.wikipedia.org/wiki/Java_(programming_language)) platform requires a [Java Development Kit](https://en.wikipedia.org/wiki/Java_Development_Kit), for [iOS](https://en.wikipedia.org/wiki/IOS) apps the [iOS SDK](https://en.wikipedia.org/wiki/IOS_SDK), and for [Universal Windows Platform](https://en.wikipedia.org/wiki/Universal_Windows_Platform) the [.NET Framework SDK](https://en.wikipedia.org/wiki/.NET_Framework_SDK). There are also SDKs that are installed in apps to provide analytics and data about activity. Prominent examples include [Google](https://en.wikipedia.org/wiki/Google), [InMobi](https://en.wikipedia.org/wiki/InMobi) and [Facebook](https://en.wikipedia.org/wiki/Facebook).

It could be something as simple as the implementation of one or more [application programming interfaces](https://en.wikipedia.org/wiki/Application_programming_interface) (APIs) in the form of on-device [libraries](https://en.wikipedia.org/wiki/Library_(computing)) to interface to a particular [programming language](https://en.wikipedia.org/wiki/Programming_language), or to include sophisticated hardware that can communicate with a particular [embedded system](https://en.wikipedia.org/wiki/Embedded_system). Common [tools](https://en.wikipedia.org/wiki/Software_development_tool) include debugging facilities and other [utilities](https://en.wikipedia.org/wiki/Utility_program), often presented in an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE). SDKs also frequently include sample code and supporting technical notes or other supporting documentation to help clarify points made by the primary reference material.

## Details

SDKs often include [licenses](https://en.wikipedia.org/wiki/License_compatibility) that make them unsuitable for building software intended to be developed under an incompatible license. For example, a proprietary SDK is generally incompatible with [free software](https://en.wikipedia.org/wiki/Free_software) development, while a [GPL](https://en.wikipedia.org/wiki/GNU_General_Public_License)-licensed SDK could be incompatible with proprietary software development. [LGPL](https://en.wikipedia.org/wiki/GNU_Lesser_General_Public_License) SDKs are typically safe for proprietary development.

The average [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) [mobile app](https://en.wikipedia.org/wiki/Mobile_app) implements 15.6 separate SDKs, with gaming apps implementing on average 17.5 different SDKs. The most popular SDK categories for Android mobile apps are analytics and advertising.

SDKs can be unsafe (because they are implemented within apps, but yet run separate code). Malicious SDKs (with honest intentions or not) can violate users' [data privacy](https://en.wikipedia.org/wiki/Data_privacy), damage app performance, or even cause apps to be banned from [Google Play](https://en.wikipedia.org/wiki/Google_Play) or iTunes.[[2]](https://en.wikipedia.org/wiki/Software_development_kit#cite_note-TechCrunch-2) New technologies allow app developers to control and monitor client SDKs in real time.

Developers typically get the SDK from the target system developer. Often the SDK can be downloaded directly via the Internet or via SDK marketplaces. Many SDKs are provided for free to encourage developers to use the system or language. Sometimes this is used as a marketing tool. Freely offered SDKs are often able to monetize, based on user data taken from the apps, which serves the interests of larger players in the ecosystem.

An SDK for an operating system add-on (for instance, [QuickTime](https://en.wikipedia.org/wiki/QuickTime) for [classic Mac OS](https://en.wikipedia.org/wiki/Classic_Mac_OS)) might include the add-on software itself to be used for development purposes, although not necessarily for redistribution together with the developed product. Between platforms where it is possible to develop applications that can at least start up on a system configuration without the add-on installed, and that use a [Gestalt](https://en.wikipedia.org/wiki/Gestalt_(Mac_OS))-style run-time environment query to determine whether the add-on is present, and ones where the application will simply fail to start, it is possible to build a single binary that will run on configurations with and without the add-on present, although operating with reduced functionality in the latter situation.

Providers of SDKs for specific systems or [subsystems](https://en.wikipedia.org/wiki/Subsystems) sometimes substitute a more specific term instead of *software*. For instance, both [Microsoft](https://en.wikipedia.org/wiki/Microsoft) and [Apple](https://en.wikipedia.org/wiki/Apple_Inc.) provide driver development kits (DDK) for developing [device drivers](https://en.wikipedia.org/wiki/Device_driver).

**3.4 Database**

### **3.4.1 SQLite**

SQLite is a relational database management system (RDBMS). If most RDBMSs such as MySQL, Oracle, etc. are standalone server processes, then SQLite is embedded because it is provided in the form of a library that is linked in applications.[3]

Like other RDBMSs, data is accessed in a SQLite database by using Structured Query Language (SQL).

### **Android SQLite Java Classes**

**Cursor**: a class provides access to the results of a database query. Its methods include:

* close(): release all resources used by cursor and close it.
* getCount(): returns the number of rows contained within the result set.
* moveToFirst(): moves to the first row in the result set.
* moveToLast(): moves to the last row in the result set.
* moveToNext(): moves to the next row in the result set.
* move(): moves by a specified offset from the current position in the result set.
* get<type>() (such as getInt(), getDouble(), so on): returns the value of the specified <type> contained at the specified column index of the row at the current cursor position.

**SQLiteDatabase**provides the primary interface between the application code and underlying SQLite database. Its methods include:

* insert(): inserts a new row into a database table.
* delete(): deletes rows from a database table
* query(): performs a specified database query and returns matching results via a Cursor object.
* execSQL(): executes a single SQL Statement that does not return result data.
* rawQuery(): executes an SQL query statement and returns matching results in the form of a Cursor object.

**SQLiteOpenHelper** is designed to make it easier to create and update databases. Its methods include:

* onCreate(): called when the database is created for the first time.
* onUpgrade(): called in the event that the application code contains a more recent database version number reference.
* onOpen(): called when the database is opened.
* getWritableDatabase(): opens or creates a database for reading and writing.
* getReadableDatabase(): creates or opens a database for reading only.
* close(): closes the database.

**ContentValues** allows key/value pairs to be declared consisting of table column identifiers and the values to be stored in each column. Its methods include:

* put(): adds a value to the set.

**3.4.2 Firebase:**

**Firebase** is a [mobile](https://en.wikipedia.org/wiki/Mobile_application) and [web application](https://en.wikipedia.org/wiki/Web_application) development platform developed by Firebase, Inc. in 2011, then acquired by [Google](https://en.wikipedia.org/wiki/Google) in 2014. Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that weren't chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in April 2012.

Firebase Inc. raised [seed funding](https://en.wikipedia.org/wiki/Seed_funding) in May 2012. The company further raised [Series A funding](https://en.wikipedia.org/wiki/Series_A_funding) in June 2013. In October 2014, Firebase was acquired by Google. In October 2015, Google acquired Divshot to merge it with the Firebase team. Since the acquisition, Firebase has grown inside Google and expanded their services to become a unified platform for mobile developers. Firebase now integrates with various other Google services to offer broader products and scale for developers. In January 2017, Google acquired Fabric and [Crashlytics](https://en.wikipedia.org/wiki/Crashlytics) from [Twitter](https://en.wikipedia.org/wiki/Twitter) to join those services to the Firebase team. According to the report, Firebase would be launching Cloud Firestore, a Document Database, in October 2017. [4]

Firebase evolved from Envolve, a prior startup founded by James Tamplin and Andrew Lee in 2011. Envolve provided developers an API that enables the integration of online chat functionality into their websites. After releasing the chat service, Tamplin and Lee found that it was being used to pass application data that weren't chat messages. Developers were using Envolve to sync application data such as game state in real time across their users. Tamplin and Lee decided to separate the chat system and the real-time architecture that powered it. They founded Firebase as a separate company in April 2012.

Firebase Inc. raised [seed funding](https://en.wikipedia.org/wiki/Seed_funding) in May 2012. The company further raised [Series A funding](https://en.wikipedia.org/wiki/Series_A_funding) in June 2013. In October 2014, Firebase was acquired by Google. In October 2015, Google acquired Divshot to merge it with the Firebase team. Since the acquisition, Firebase has grown inside Google and expanded their services to become a unified platform for mobile developers. Firebase now integrates with various other Google services to offer broader products and scale for developers. In January 2017, Google acquired Fabric and [Crashlytics](https://en.wikipedia.org/wiki/Crashlytics) from [Twitter](https://en.wikipedia.org/wiki/Twitter) to join those services to the Firebase team. According to the report, Firebase would be launching Cloud Firestore, a Document Database, in October 2017.

### **Develop**

#### **Firebase Cloud Messaging**

Formerly known as [Google Cloud Messaging](https://en.wikipedia.org/wiki/Google_Cloud_Messaging) (GCM), [Firebase Cloud Messaging](https://en.wikipedia.org/wiki/Firebase_Cloud_Messaging) (FCM) is a cross-platform solution for messages and notifications for [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [iOS](https://en.wikipedia.org/wiki/IOS), and [web applications](https://en.wikipedia.org/wiki/Web_application), which currently can be used at no cost.

#### **Firebase Auth**

Firebase Auth is a service that can authenticate users using only client-side code. It supports [social login providers](https://en.wikipedia.org/wiki/Social_login) Facebook, GitHub, Twitter and Google (and [Google Play Games](https://en.wikipedia.org/wiki/Google_Play_Games)). Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.

#### **Realtime Database**

Firebase provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. The company provides client libraries that enable integration with [Android](https://en.wikipedia.org/wiki/Android_(operating_system)), [iOS](https://en.wikipedia.org/wiki/IOS), [JavaScript](https://en.wikipedia.org/wiki/JavaScript), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [Objective-C](https://en.wikipedia.org/wiki/Objective-C), [swift](https://en.wikipedia.org/wiki/Swift_(programming_language)) and [Node.js](https://en.wikipedia.org/wiki/Node.js) applications.The database is also accessible through a REST API and bindings for several [JavaScript frameworks](https://en.wikipedia.org/wiki/JavaScript_frameworks) such as [AngularJS](https://en.wikipedia.org/wiki/AngularJS), [React](https://en.wikipedia.org/wiki/React_(JavaScript_library)), [Ember.js](https://en.wikipedia.org/wiki/Ember.js) and [Backbone.js](https://en.wikipedia.org/wiki/Backbone.js). The REST API uses the [Server-Sent Events](https://en.wikipedia.org/wiki/Server-sent_events) protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the realtime database can secure their data by using the company's server-side-enforced security rules. Cloud Firestore which is Firebase's next generation of the Realtime Database was released for beta use.

#### **Firebase Storage**

Firebase Storage provides secure file uploads and downloads for Firebase apps, regardless of network quality. The developer can use it to store images, audio, video, or other user-generated content. Firebase Storage is backed by Google Cloud Storage.

#### **Firebase Hosting**

Firebase Hosting is a static and dynamic [web hosting service](https://en.wikipedia.org/wiki/Web_hosting_service) that launched on May 13, 2014. It supports hosting static files such as [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets), [HTML](https://en.wikipedia.org/wiki/HTML), [JavaScript](https://en.wikipedia.org/wiki/JavaScript) and other files, as well as [dynamic Node.js support through Cloud Functions](https://firebase.google.com/docs/hosting/functions). The service delivers files over a [content delivery network](https://en.wikipedia.org/wiki/Content_delivery_network) (CDN) through [HTTP Secure](https://en.wikipedia.org/wiki/HTTP_Secure) (HTTPS) and [Secure Sockets Layer](https://en.wikipedia.org/wiki/Secure_Sockets_Layer)encryption (SSL). Firebase partners with Fastly, a CDN, to provide the CDN backing Firebase Hosting. The company states that Firebase Hosting grew out of customer requests; developers were using Firebase for its real-time database but needed a place to host their content.

#### **Crashlytics**

Crash Reporting creates detailed reports of the errors in the app. Errors are grouped into clusters of similar stack traces and triaged by the severity of impact on app users. In addition to automatic reports, developer can log custom events to help capture the steps leading up to a crash. Before acquiring Crashlytics, Firebase was using its own Firebase Crash Reporting.

#### **Performance**

Firebase Performance provides insights into an app's performance and the latencies the app's users experience.

#### **Firebase Test Lab for Android**

Firebase Test Lab for Android provides cloud-based infrastructure for testing Android apps. With one operation, developers can initiate testing of their apps across a wide variety of devices and device configurations. Test results—including logs, videos, and screenshots—are made available in the project in the Firebase console. Even if a developer hasn't written any test code for their app, Test Lab can exercise the app automatically, looking for crashes.

### **Grow**

#### **Firebase Notifications**

Firebase Notifications is a service that enables targeted user notifications for mobile app developers at no cost.

#### **Firebase App Indexing**

Firebase App Indexing, formerly Google App Indexing, gets an app into Google Search. Adding App Indexing promotes both types of app results within Google Search and also provides query autocompletions.

#### **Firebase Dynamic Links**

Firebase Dynamic Links are smart URLs that dynamically change behavior to provide the best experience across different platforms.

#### **Firebase Invites**

Firebase Invites is a cross-platform solution for sending personalized email and SMS invitations, on-boarding users, and measuring the impact of invitations.

#### **Firebase Remote Config**

Firebase Remote Config is a cloud service that lets developers change the behavior and appearance of their apps without requiring users to download an app update.