



# A simplified guide to android app development

FATIUL HUQ SUJOY

What you're about to read is a generalised account of the different aspects and to-dos of developing an Android app from scratch. I do not go into the nitty-gritty details with an aim to reach out to those of you who have a very minimal knowledge about the world of coding but have a latent aspiration to make your own Android app. So let's begin.

## Languages to Learn

The most used and recommended language is Java. Java is an object oriented language the syntax of which is similar to the C family. So if you already know C, C++, C# or languages like that then learning Java won't be that hard.

Java covers most of the code in an Android project. But to make it compatible with the Android operating system, a significant amount of unique libraries are needed which you won't find in a Java project. Although Java stands as a basic language for Android, you need to learn the "Android language," its libraries and APIs.

For the front-end a.k.a. the user interface, Android uses XML. But usually the IDE (Integrated Development Environment) takes care of most of the work and all you need is a basic understanding of the syntax. If you've learned HTML or CSS then you'd find XML pretty familiar.

If you're using a database for back-end codes you can use PHP and sorts. For databases there is MySQL while Android itself provides SQLite; more on that later.

## Other Languages

Although Java is the principle language for Android application development, there

are few other options out there as well.

If you're familiar with C# then you can use Microsoft Visual Studio to make apps with C#.

A popular replacement for Java and XML is the Corona SDK which uses the LUA programming language. The language is more concise than that of Java and is used mostly for game development.

Speaking of which, game development is a pain in Java, and so there are many other platforms for that, such as GameSalad or engines like Unreal or Unity.

## The IDE

We will focus on the Java side of things. The dedicated IDE, or simply put, the software to code in, is Android Studio. Do not be fooled by the innocence of its name, you will never forgive it for the pains it will provide.

Its notoriety comes from its massive consumption of RAM, its freakishly slow pace and its complete hostility to beginners. But it still is the only solid IDE we have for now, so better learn to love it.

For installation I highly recommend checking out YouTube tutorials since there are some plugins and updates that are required, and a guide will save you future troubles.

Fair warning, the IDE needs a minimum of 4 GB RAM to function. But even with that you'd have to wait decades

before the loading screen finally passes, only to start building the Gradle for another decade or two. 8 GB or more is perfect for a smooth development experience.

## Helping Hands

You might think Android application development is rather a new or niche practice to have a sizeable online community support. You'd be wrong.

The internet is filled with blogs, videos and forums where you can find handy tutorials to start you off and solutions to your most specific and trivial problems. For tutorials you can check out YouTube playlists from channels like 'thenewboston' or videos from MOOCs like Udacity. Tutorials Point and Android's Developer website cover all the necessary lessons and documentations. For specific queries there's nothing better than Stack Overflow.

## Database Management

If your program is the slightest bit advanced then you will definitely need a place to store data. For local storage Android provides three principal methods: Shared Preferences, File Storage and an SQLite database. Among these only the SQLite database is a bit difficult to com-

prehend in the beginning but it is the only one with most applicable for large amount of data with the features of a database.

For remote storages, the most popular option is using MySQL. As a beginner you can use software like XAMPP with which you can set up a local Apache server on your PC and use their provided database. You can access the server through your local network i.e. your phone can access your laptop's server if they're using internet from the same router. XAMPP also provides an admin panel where you can view and configure the data in your database.

There are online services that cater public domains for your Database if you want devices outside your local network to access the data. 000webhost is a site where you can get websites and Databases free with limited features. If you can afford it, you can always opt for paid services.

## Testing out your Program

Android Studio has an emulator, a phone simulator, to test your program on. But that is usually slow. The better option is to connect your phone to your PC with a USB cable and hit 'run'. The apk will be installed in your phone so you can use the app even after disconnecting. You can also build a standard apk to distribute among your peers for their collective whoa.

## Publishing your 'Codechild'

To see your app in the Google Playstore, you first need to open a developer account in their site. You need to pay a one-time fee of \$25. Finish with email verification and upload your app. Yeah, that's all it takes.

