My First Babble App Tutorial

This article is a walkthrough of the process of building your first Android App using the babble-android library.

Babble

Babble, mobile ad hoc block chain, yadda, yadda, yadda. **//TODO**

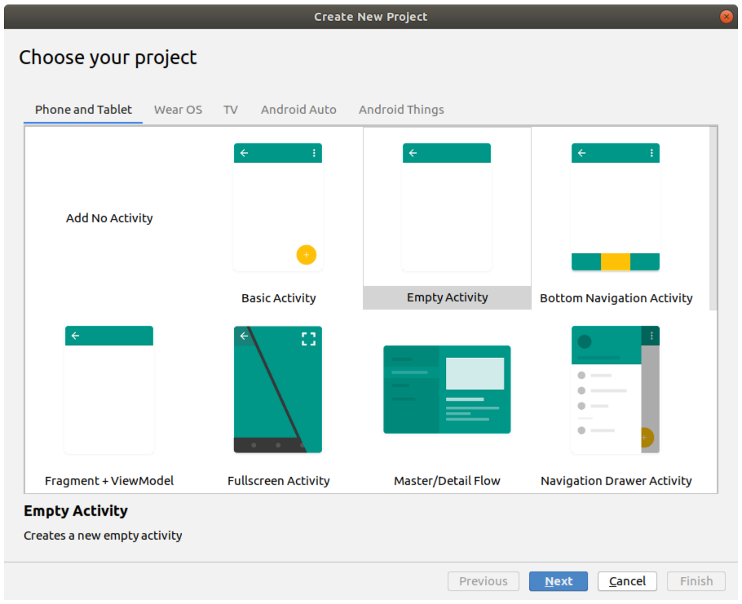
Prerequisites

We will assume that you have installed Android Studio, an appropriate Android SDK (I am using version 29) and Android NDK.

This tutorial is going to assume deployment to a physical Android device. Thus you will need an Android device (Android device requirements **//TODO**) with the developer options turned on, debugging enabled, and a suitable USB cable. You could use the android emulator, but that is beyond the scope of this article.

Our First Minimal App

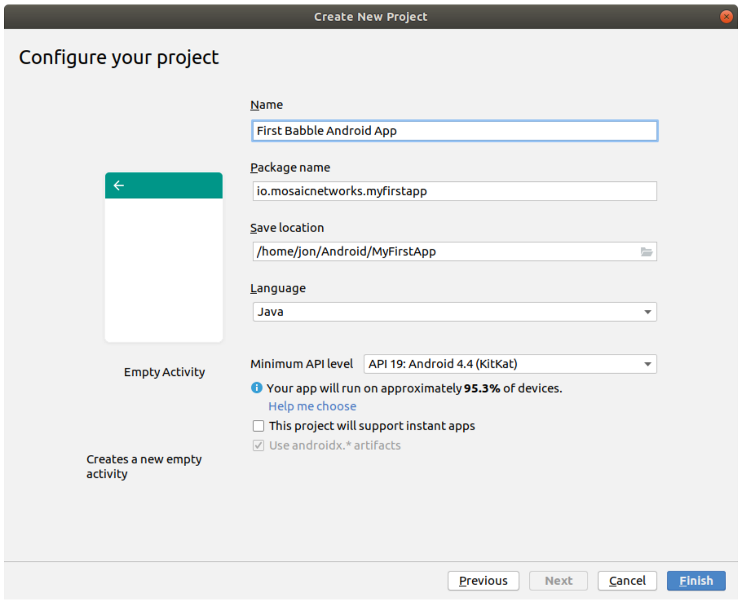
First up we will create a minimal app to test our environment and prove that we have loaded the `babble-android` library correctly. Fire up Android Studio and select `File/New Project`. You are asked to choose a project template.



Empty Activity

Creates a new empty activity

Choose *Empty Activity* from the *Phone and Tablet* tab, and click Next.



Create New Project

Configure your project

Name: First Babble Android App

Package name: io.mosaicnetworks.myfirstapp

Save location: /home/jon/Android/MyFirstApp

Language: Java

Minimum API level: API 19: Android 4.4 (KitKat)

Your app will run on approximately **95.3%** of devices.

Help me choose

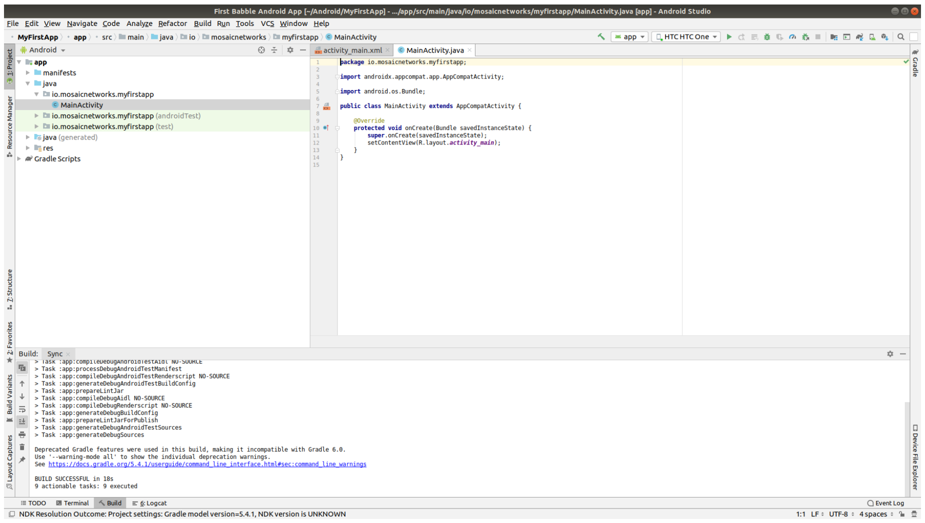
☐ This project will support instant apps

☑ Use androidx.* artifacts

Empty Activity

Creates a new empty activity

The options here should be self-explanatory. We would recommend not using spaces in the Package Name or the Save Location. Do not set the Minimum API Level below 19. Click Finish.



First Babble Android App [~/Android/MyFirstApp] - .../app/src/main/java/io/mosaicnetworks/myfirstapp/MainActivity.java [app] - Android Studio

```java
package io.mosaicnetworks.myfirstapp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```
> Task :app:compileDebugAndroidTestAidl NO-SOURCE
> Task :app:processDebugAndroidTestManifest
> Task :app:compileDebugAndroidTestRenderscript NO-SOURCE
> Task :app:generateDebugAndroidTestBuildConfig
> Task :app:prepareLintJar
> Task :app:compileDebugAidl NO-SOURCE
> Task :app:compileDebugRenderscript NO-SOURCE
> Task :app:generateDebugBuildConfig
> Task :app:prepareLintJarForPublish
> Task :app:generateDebugAndroidTestSources
> Task :app:generateDebugSources

Deprecated Gradle features were used in this build, making it incompatible with Gradle 6.0.
Use '--warning-mode all' to show the individual deprecation warnings.
See https://docs.gradle.org/5.4.1/userguide/command_line_interface.html#sec:command_line_warnings

BUILD SUCCESSFUL in 18s
9 actionable tasks: 9 executed
```

NDK Resolution Outcome: Project settings: Gradle model version=5.4.1, NDK version is UNKNOWN

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
MyFirstApp > app > src > main > java > io > mosaicnetworks > myfirstapp > MainActivity
Android
app
manifests
java
io.mosaicnetworks.myfirstapp
MainActivity
io.mosaicnetworks.myfirstapp (androidTest)
io.mosaicnetworks.myfirstapp (test)
java (generated)
res
Gradle Scripts
activity_main.xml
MainActivity.java
app
HTC HTC One
Build: Sync
TODO
Terminal
Build
6: Logcat
Event Log
1:1 LF UTF-8 4 spaces

Android Studio will open, and after expanding some of the menus it will look something like above.

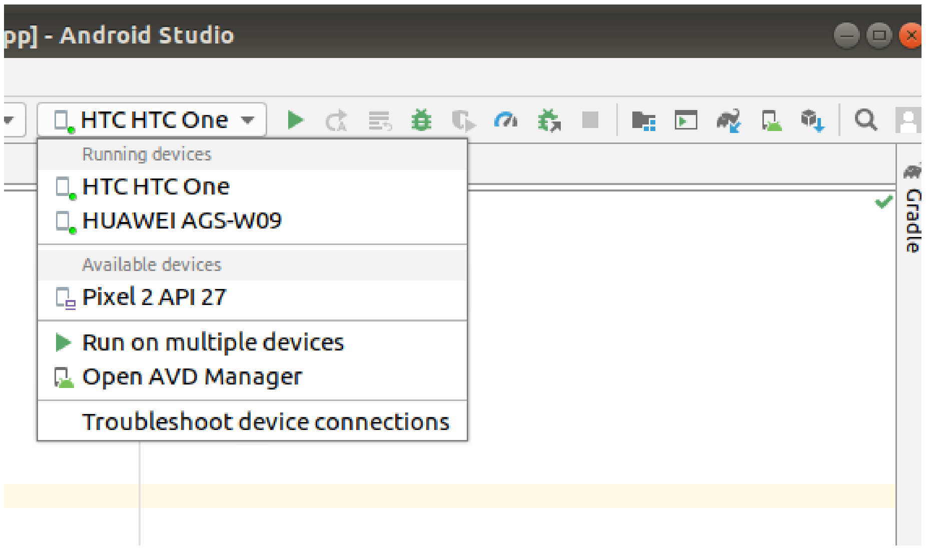
N.B. the screen will change a few seconds after opening when the initial gradle scripts complete. The status bar at the bottom of the window should tell you this *is happening.*

Connect your android device to your computer via a USB cable. First we will test that the Android Debug Bridge (**adb**) can see the device.

```
jon@hpjon:~/Android/MyFirstApp$ adb devices
List of devices attached
4JPNU18709118621    device
jon@hpjon:~/Android/MyFirstApp$
```

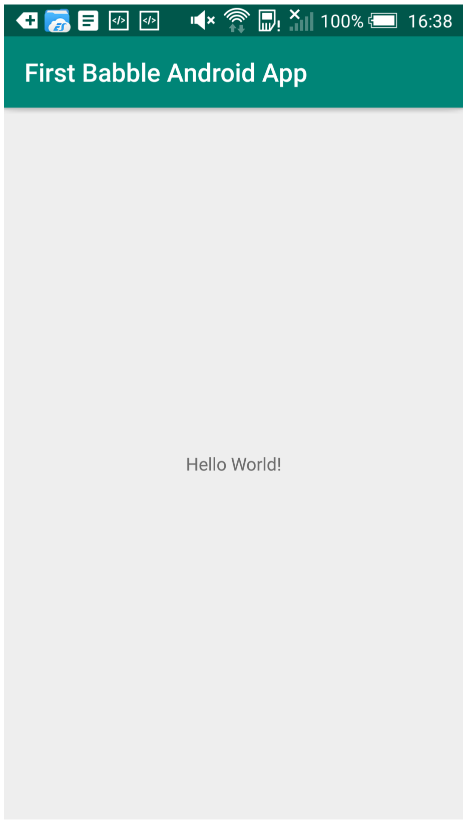
The command `adb kill-server` will reset this connection.

If you can see a device listed, go back to Android Studio. and in the top right is a target device dropdown. Select your device from the list.

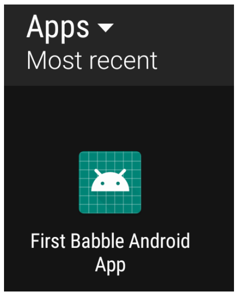


If you cannot find your device in the list, the `Troubleshoot device connections` option on that menu should help.

Then press that little green triangle to the right of the dropdown device menu. Gradle then builds the app, which is then installed onto the physical android device that you selected. The whole process tool about 20 seconds on my laptop (feel free to buy me a quicker one).



If you look on the device, you should find the app installed, as below:

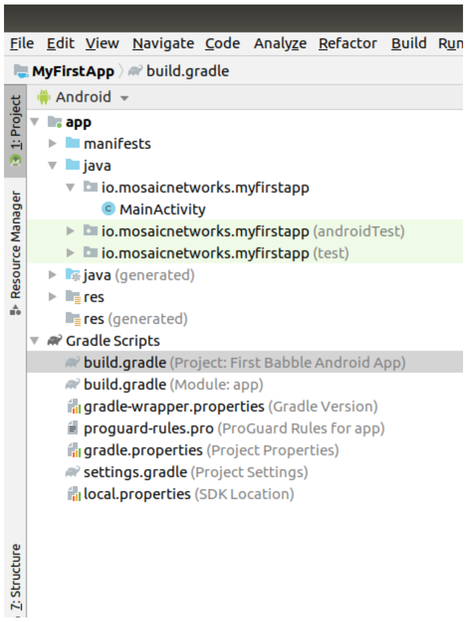


Congratulations, you have an app. Now onto Babble. . .

Our First Babble App

We will now integrate the `babble-android` library into our skeleton app. We will use it to generate a key pair – just to prove that we have a working library instance.

The library is hosted **jcenter**. To make it available, we need to amend some gradle scripts.


File Edit View Navigate Code Analyze Refactor Build Run
MyFirstApp
build.gradle
Android
1: Project
Resource Manager
Z: Structure
app
manifests
java
io.mosaicnetworks.myfirstapp
MainActivity
io.mosaicnetworks.myfirstapp (androidTest)
io.mosaicnetworks.myfirstapp (test)
java (generated)
res
res (generated)
Gradle Scripts
build.gradle (Project: First Babble Android App)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)

In Android Studio, double click on the Project `build.gradle` as highlighted in the screenshot above.

We then add the 3 line `maven` instruction as below:

```
allprojects {
    repositories {
        google()
        jcenter()
        maven {
            url 'https://dl.bintray.com/mosaicnetworks/maven'
        }
    }
}
```

Which leaves the entire file looking like this:

```
// Top-level build file where you can add configuration options common to all sub-projects/

buildscript {
    repositories {
        google()
        jcenter()

    }
    dependencies {
        classpath 'com.android.tools.build:gradle:3.5.2'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        google()
        jcenter()
        maven {
            url 'https://dl.bintray.com/mosaicnetworks/maven'
        }
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

Next we need to amend the app `build.gradle` (it is below the Project `build.gradle` in the screenshot above. We add an implementation line to the bottom dependencies section.

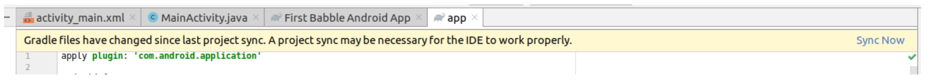
```
implementation 'io.mosaicnetworks:babble:0.1.0'
```

This leaves us with this full file:

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 29
    buildToolsVersion "29.0.2"
    defaultConfig {
        applicationId "io.mosaicnetworks.myfirstapp"
        minSdkVersion 19
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguar
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'io.mosaicnetworks:babble:0.1.0'
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
}
```

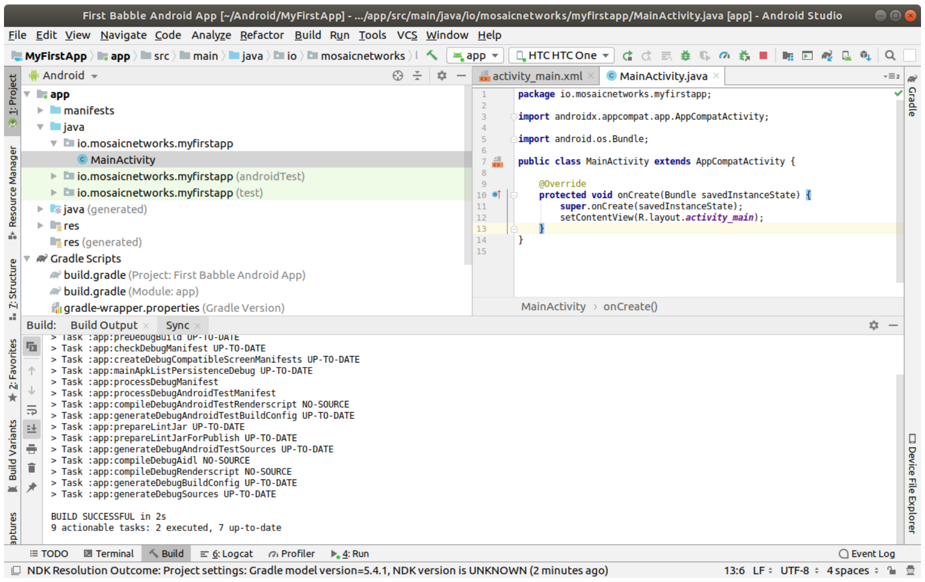


Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly. Sync Now

```
apply plugin: 'com.android.application'
```

When you get a message like the above, click the `Sync Now` link on the right of the message.

The library should now be included in the project. So lets use it! Open `MainActivity.java` as below:



First Babble Android App [~/Android/MyFirstApp] - .../app/src/main/java/io/mosaicnetworks/myfirstapp/MainActivity.java [app] - Android Studio

```java
package io.mosaicnetworks.myfirstapp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

```
> Task :app:preDebugBuild UP-TO-DATE
> Task :app:checkDebugManifest UP-TO-DATE
> Task :app:createDebugCompatibleScreenManifests UP-TO-DATE
> Task :app:mainApkListPersistenceDebug UP-TO-DATE
> Task :app:processDebugManifest
> Task :app:processDebugAndroidTestManifest
> Task :app:compileDebugAndroidTestRenderscript NO-SOURCE
> Task :app:generateDebugAndroidTestBuildConfig UP-TO-DATE
> Task :app:prepareLintJar UP-TO-DATE
> Task :app:prepareLintJarForPublish UP-TO-DATE
> Task :app:generateDebugAndroidTestSources UP-TO-DATE
> Task :app:compileDebugAidl NO-SOURCE
> Task :app:compileDebugRenderscript NO-SOURCE
> Task :app:generateDebugBuildConfig UP-TO-DATE
> Task :app:generateDebugSources UP-TO-DATE

BUILD SUCCESSFUL in 2s
9 actionable tasks: 2 executed, 7 up-to-date
```

NDK Resolution Outcome: Project settings: Gradle model version=5.4.1, NDK version is UNKNOWN (2 minutes ago)

Add the lines below underneath the last import statement. The lines will appear greyed out, as the import is not yet used. As well as babble we are importing the `Log` package to write to the Android logs.

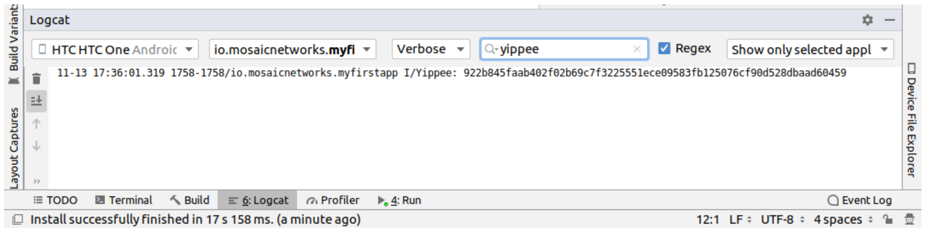
```java
import io.mosaicnetworks.babble.node.KeyPair;
import android.util.Log;
```

Add the following lines as the last line of the `onCreate` function

```
KeyPair kp = new KeyPair();
Log.i("Yippee",kp.privateKey);
```

This code generates a key pair and writes the private code to the logs.

Save all the files and run your app.



```
11-13 17:36:01.319 1758-1758/io.mosaicnetworks.myfirstapp I/Yippee: 922b845faab402f02b69c7f3225551ece09583fb125076cf90d528dbaad60459
```

Install successfully finished in 17 s 158 ms. (a minute ago)

The app looks exactly as per the previous iteration, so lets take a look under the hood. Press logcat, as highlighted in gray in the screenshot above. Then type `yippee` in the search box at the top of that window to filter the logs. You should have a freshly generated private key in there.