

**Note:** Assuming you have JDK 1.8 already set up in your system. Otherwise, download and install Java using this link -

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

## How to setup Environment (Android Studio, Flutter, Dart) and Open Existing Project from File System:

- Download Android Studio from the official website and install.
- Setup Android Studio
  - Download Flutter and Dart:
    - Go to File -> Setting -> Plugins
    - Install Flutter (it should ask for installing Dart also)
  - Setup SDK:
    - Find the Icon (top right) that says SDK Manager and click it.
    - Click Android SDK
    - Select Android 10.0, Android 9.0 and download the SDK
    - Go to File -> Project Structure -> SDKs
    - Select **Android API 28** under Build target.
    - Select JDK 1.8 (it should be already set)
    - Select OK and exit.
- Open project:
  - Download project from github from master branch
  - In android studio, go to File -> open, file chooser will open.
  - Browse to the project directory and choose the folder "**WooLaLa2020**" and open it in android studio.
- After opening project:
  - Open a terminal and cd to **WooLaLa2020/woolala\_app/** directory.
  - Then execute **flutter clean** and **flutter pub get**
  - Then go to Build -> Flutter -> Build APK
  - it should start building apk and most likely will show a bunch of errors relating to- **Keystore, flutter path** etc. Regardless, please do all steps below.

## Running on Android Studio:

After setting up the steps above and fixing the errors below, you can run the project within the **woolala\_app** directory. There are some steps that may need to be done for proper execution.

- Select correct Virtual Device:
  - Click the phone option with the Android logo/head located in the top right corner or navigate to AVD manager from searching within "Help".
  - Create a new Virtual Device, pick an Android phone of choice, and navigate to the next page.
  - Navigate to the "**x86 images**" tab. Make sure to choose an image here with ABI -> **x86\_64 compatibility**; safest choice is Pie -> 28 -> x86\_64.

- Download image if not already downloaded. Default settings after this are fine.
- Run application:
  - Select the leftmost field to the play/run button at the top of the screen.
  - Wait for loading and then open the Virtual Device you created.
  - Once this loads and turns on, run the app using **flutter run --release**.
  - NOTE: the fixes below for login will only run while under release mode, flutter run --debug or flutter run will pull up the app, but will not apply login fixes.

### **Error: “Flutter/dart sdk not set in local.properties”**

You need to set the flutter sdk path in local.properties file.

- Go to **WooLaLa2020/woolala\_app/android/** and open **local.properties** file.
- If you had downloaded flutter and dart before using previous step, open a terminal and do the following commands to know the sdk path:
  - which flutter (linux/mac)
  - where flutter (windows)
  - which dart (linux/mac)
  - where dart (windows)
- Paste the path under **flutter.sdk** or **dart.sdk**

### **Error: “Keystore file not set for signing config file”**

If you get an error “**keystore file not set.....**”, chances are you need to create your own keystore file (key.jks) using a tool named **keytool**. Follow below steps -

1. Use this command to generate key.jks file:
 

```
keytool -genkey -v -keystore ~/key.jks -keyalg RSA -keysize 2048 -validity 10000 -alias key
```

It will ask for a bunch of information, so go ahead and provide the information in the terminal.
2. After successful key generation, there will be a **key.jks** file created at **/home/<username>/** directory. Cut the file from this directory and put inside the **WooLaLa2020/woolala\_app/android/app/** directory.
3. Create a **key.properties** file in the **WooLaLa2020/woolala\_app/android/** directory, if it does not exist.
4. Put the following information in the key.properties file:
 

```
storePassword=<password you choose in step1>
keyPassword=<password you choose in step1>
keyAlias=key
```

storeFile=key.jks

5. Now the keystore issue should be gone.

### **Error: Login Issue for Google Sign In through local firebase:**

This app creates a local firebase database and stores credentials for local demo. To run the app with the local firebase database, you will have to modify the app in the pubspec.yaml file to add some more dependencies for **firebase**.

### **Error: Login Issue for Google Sign In using firebase:**

1. Follow the last instructions to generate a **key.jks** file.
2. List release key from the Keystore using *keytool* with following command:  
Keytool -list -v -keystore <key.jks file location>

This command will generate a bunch of important encryption keys. Copy them and save them in a txt file.

3. Login in **firebase.google.com** using  
Username: [fashionxtllc@gmail.com](mailto:fashionxtllc@gmail.com)  
Password: Fash10nxt\*7
4. Add your SHA1/SHA256 keys from step2 into google firebase application configuration in the "Add Fingerprint" section.
5. Now you should be able to login for the app using google sign-in.

### **Adding Key hash for Facebook login:**

<https://stackoverflow.com/questions/7506392/how-to-create-android-facebook-key-hash>

### **Insecure http Requests:**

If you are attempting to run a local node.js server on your computer and you are running into an error that is either not allowing for insecure http or spitting out an "Unrecognized character at 0 <Doctype HTML>," do the following.

- For Android, see this link:  
<https://github.com/flutter/flutter/issues/66275#issuecomment-710161329>.

- Add the code- `android:usesCleartextTraffic="true"`- to the path: under line 14 of `android/app/src/main/AndroidManifest.xml`
- For iOS, you must set some variables in your `info.plist`
  - <https://stackoverflow.com/questions/64197752/bad-state-insecure-http-is-not-allowed-by-platform>