Jasonelle Android Setup Tools

These are a bunch of scripts and docker files that will help you developing apps with Jasonette in Android.

Table of Contents

Directory Structure 1
Installation 2
  Linux/Mac 2
  Windows 2
Android 2
  Jasonette 2
  Keystore Creation 3
    Example build.gradle 3
  Edit your files 4
  Edit your settings 4
Commands 4
  Installation Command 4
  Download and unzip the android directory 5
  Create Android Keystore 5
  Build Android APK in Debug Mode 5
  Build Android APK in Release Mode 5
License 5
Credits 6

Directory Structure

When unzipping you can see the following directory structure:

```
. (root)
├──── Dockerfile
├──── Makefile
├──── app
│     ├──── build
│     ├──── file
│     ├──── keys
│     └──── settings
├──── docs
├──── jasonette
└──── tools
    └──── windows
```

Installation

The only requirement is *Docker Desktop* at least *v20.10.0*. You can read the installation docs at https://docs.docker.com/get-docker/. Be sure that is up and running before installing the *Jasonelle Android Setup Tools*.

Linux/Mac

Linux and *Mac* use the Make command. It is a fairly standard tool.

- Install Make in MacOS
- Install Make in Ubuntu Linux

Open the **terminal** application, go to the *root* directory and run:

```
$ make install
```

Windows

Open a new **cmd** application inside the *root* directory and execute:

```
.\tools\windows\install.cmd
```

Android

Jasonette

The main directory is *jasonette/android*. Here you can put your own jasonette project.

Keystore Creation

Before sending your app to the *PlayStore* you need to create a new *Keystore* that will sign the app.

Follow the instructions in the Android Studio Help Center (https://developer.android.com/studio/publish/app-signing.html#generate-key) to generate a new key. It must be different from any previous keys. Alternatively, you can use the following command line to generate a new key:

```
$ keytool -genkeypair -alias upload -keyalg RSA -keysize 2048 -validity 9125 -keystore
keystore.jks
```

This key must be a 2048 bit RSA key and have 25-year validity. Export the certificate for that key to PEM format:

```
$ keytool -export -rfc -alias upload -file upload_certificate.pem -keystore
keystore.jks
```

- **Important**: Keep in mind that the generated **jks** or **keystore** file will be used for signing all future updates of your app. Please make sure you store it in a safe place. If you lost it, you will need to contact *Google Support* to upload a new **keystore** file. The **keystore** file contains both *private* and *public* keys. **pem** and **der** files are only public keys.
- **Extra Important**: When creating the **keystore** the command will ask to input a password. Make sure you use a secure password and remember it. Otherwise your **keystore** will not be possible to sign the *APK* and you would need to recreate a new **keystore**.

You can use the **Keystore Tool** to ease the keystore creation.

More information on creating and using a **keystore** can be found at https://positive-stud.medium.com/step-by-step-guide-to-generate-a-signed-apk-using-android-studio-1e22ab7b3e86 and https://developer.android.com/training/articles/keystore

Example build.gradle

This is a sample *build.gradle* file which was configured for using the **jasonette.keystore** file.

```
    // ...
    signingConfigs {
        release {
            storeFile file('jasonette.keystore')
            storePassword 'your_key_store_password'
            keyAlias 'your_key_alias'
            keyPassword 'your_key_file_alias_password'
        }
    }
    buildTypes {
        release {
            debuggable false
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
rules.pro'

            // set the signing config for the release apk
            signingConfig signingConfigs.release
        }
    }
    // ...
```

Edit your files

The main directory for your files is `app/file`. Where every file here will be copied to `jasonette/android`. By default it was included three example apps. Select your initial app in `app/settings/android/strings.xml`.

Edit your settings

The following files:

- AndroidManifest.xml
- build.gradle
- settings.xml

will be stored inside `app/settings/android`. They will overwrite the files inside `jasonette/android`.

Commands

Installation Command

Use this command to install all the dependencies of Docker and Jasonelle. This is the first command you must execute before everything else. Be sure that the *Docker Service* is running and all the directories have **read/write** (*chmod 755*) permissions.

- Linux/Mac: `$ make install`
- Windows: `.\tools\windows\install.cmd`

Download and unzip the android directory

Danger: It will delete the jasonette/android directory if exists.

- Linux/Mac: `$ make da`
- Windows: `.\tools\windows\android-download-latest.cmd`

Create Android Keystore

Use this command to create a new file **.keystore* that will be stored inside *app/keys*. This file is needed to build a release APK and sign in.

If you already have a **.keystore* file then this step is optional. Be sure to name the file **jasonette.keystore**.

- Linux/Mac: `$ make ki`
- Windows: `.\tools\windows\android-key-tool.cmd`

Build Android APK in Debug Mode

Use this command to create an APK in debug mode. It will copy files inside *app/file* and *app/settings/android*. The APK will be copied to *app/build* directory.

- Linux/Mac: `$ make bad`
- Windows: `.\tools\windows\android-build-debug.cmd`

Build Android APK in Release Mode

Use this command to create an APK in release mode. It will copy files inside app/file app/settings/android. Also it will copy the keystore file. Make sure your build.gradle file is configured with the keystore file. The APK will be copied to build/ directory.

- Linux/Mac: `$ make bar`
- Windows: `.\tools\windows\android-build-release.cmd`

License

The following files are under the GNU AFFERO GENERAL PUBLIC LICENSE (Version 3).

- *Dockerfile*
- *Makefile*

- All files inside *tools* directory

Other files are under their respective licenses.

Credits

```
Made with <i class="fa fa-heart">&#9829;</i> by <a href="https://ninjas.cl"
target="_blank">Ninjas.cl</a>.
```