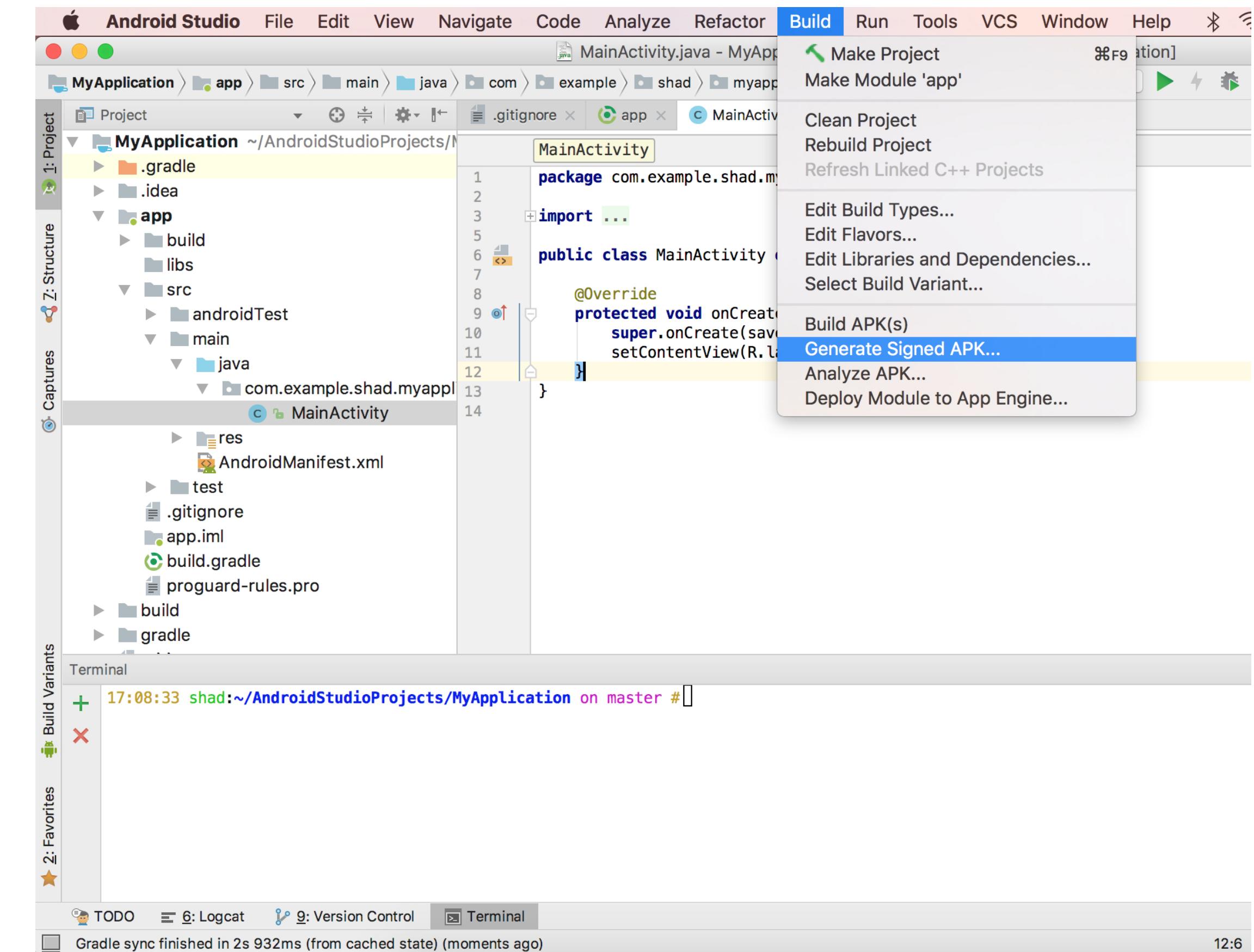


Certificates and keystores

When you sign an APK, the signing tool attaches the public-key certificate to the APK. The public-key certificate serves as a "fingerprint" that uniquely associates the APK to you and your corresponding private key.

This helps Android ensure that any future updates to your APK are authentic and come from the original author. The key used to create this certificate is called the app signing key.



Generate a key and keystore

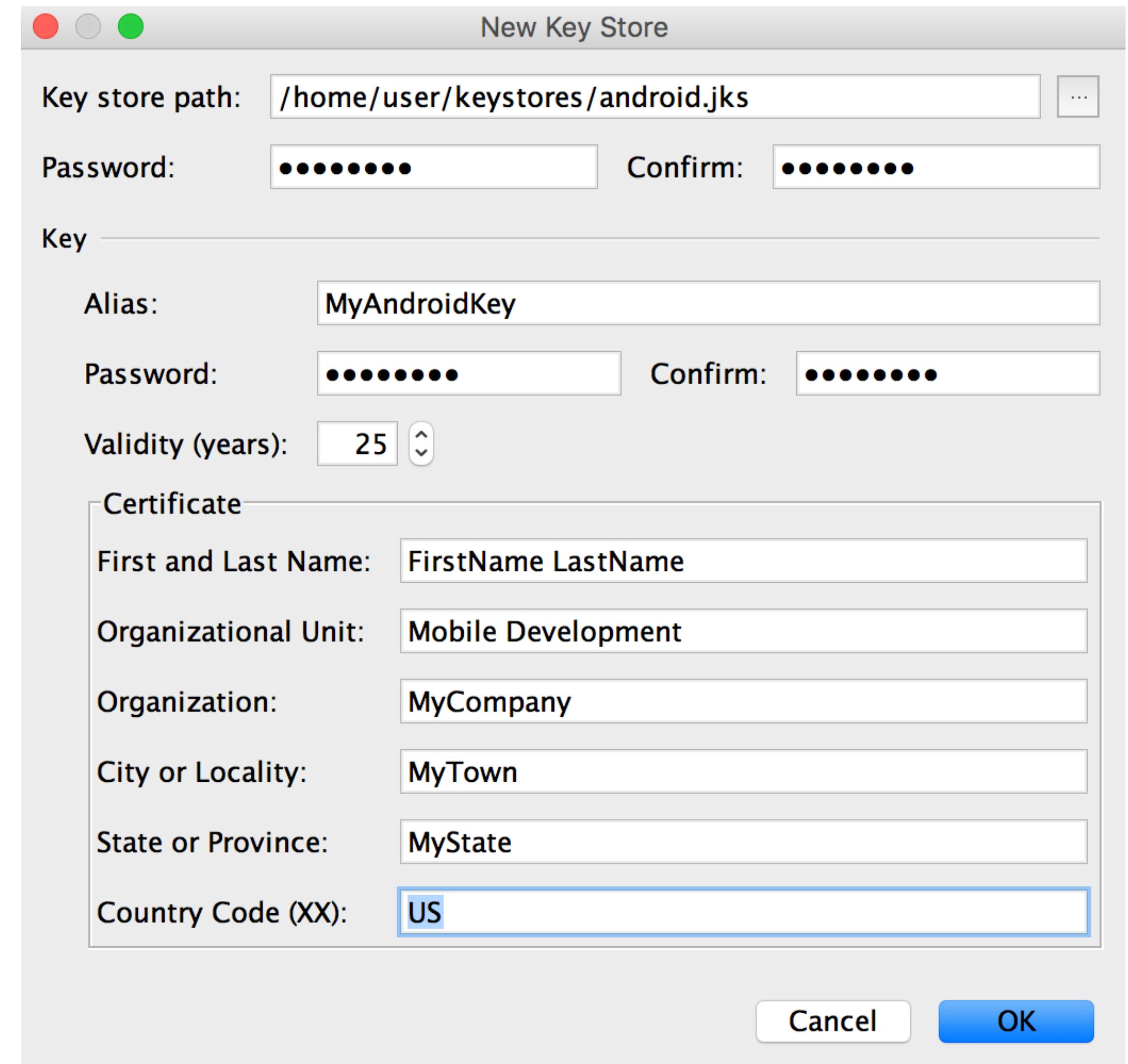
1. In the menu bar, click **Build > Generate Signed APK > Next > Create new.**

2. Keystore

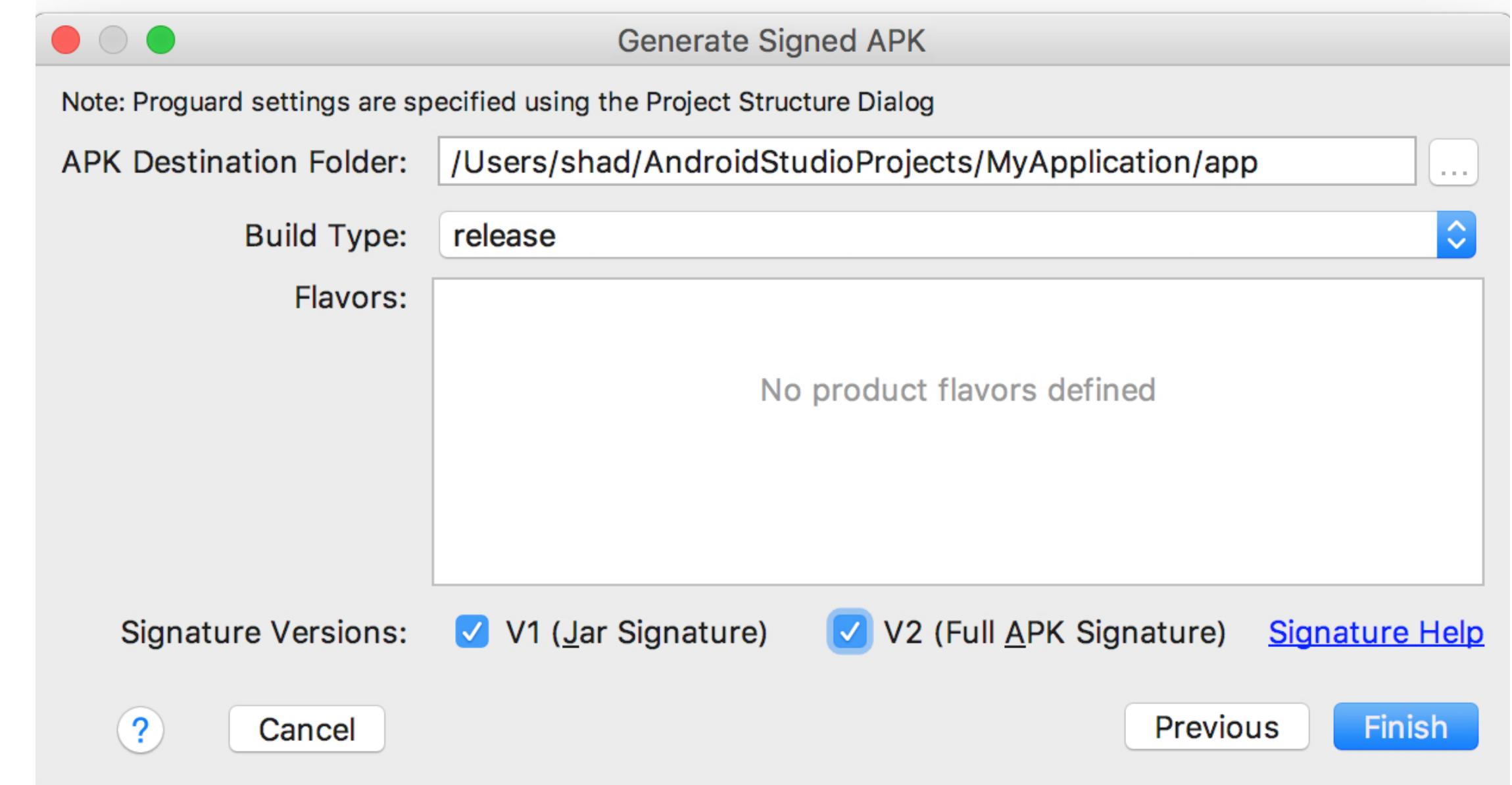
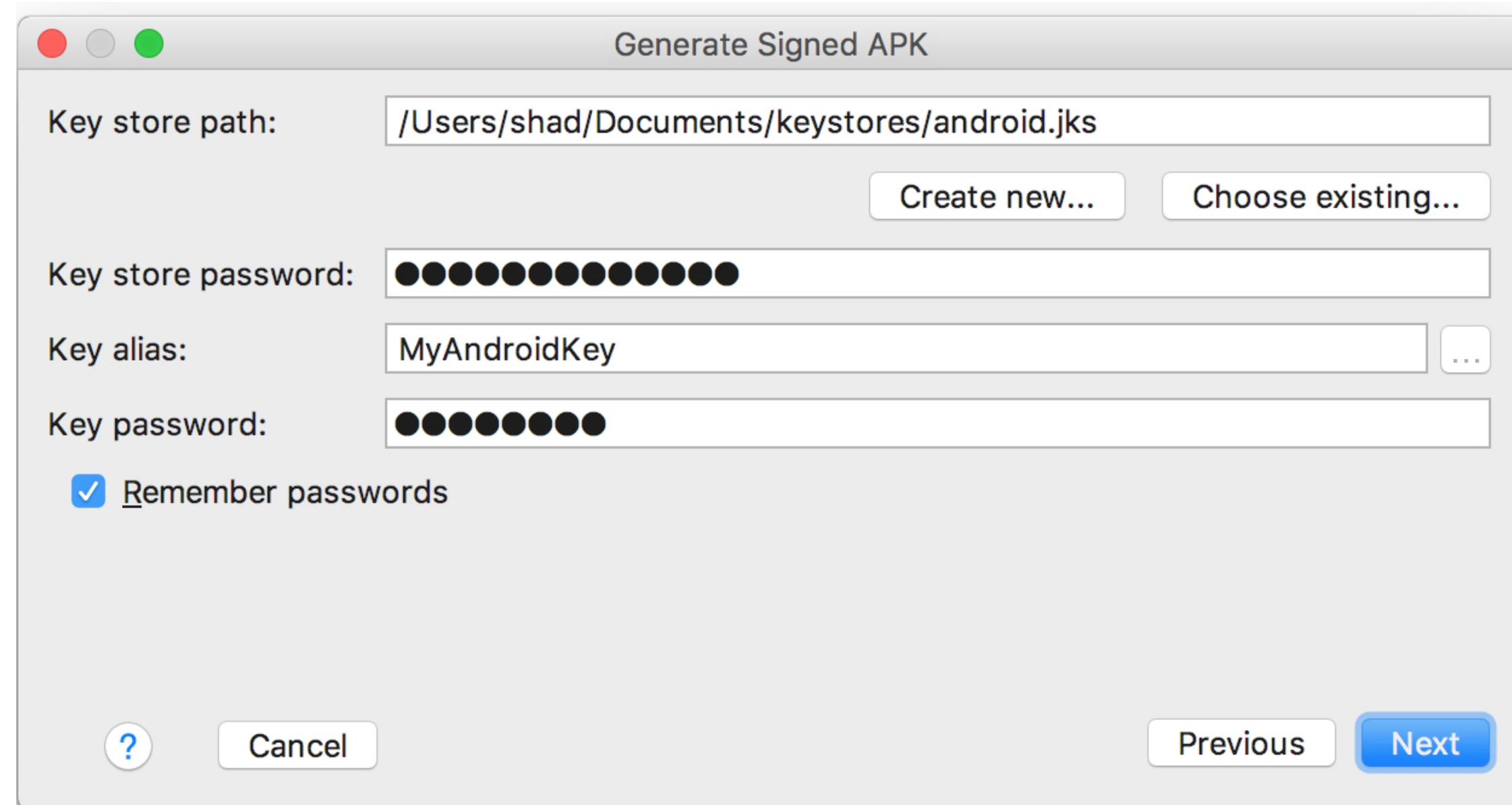
- **Key store path:** Select the location where your keystore should be created.
- **Password:** Create and confirm a secure password for your keystore.

3. Key

- **Alias:** Enter an identifying name for your key.
- **Password:** Create and confirm a secure password for your key. This should be different from the password you chose for your keystore
- **Validity (years):** Set the length of time in years that your key will be valid. Your key should be valid for at least 25 years, so you can sign app updates with the same key through the lifespan of your app.
- **Certificate:** Enter some information about yourself for your certificate. This information is not displayed in your app, but is included in your certificate as part of the APK.

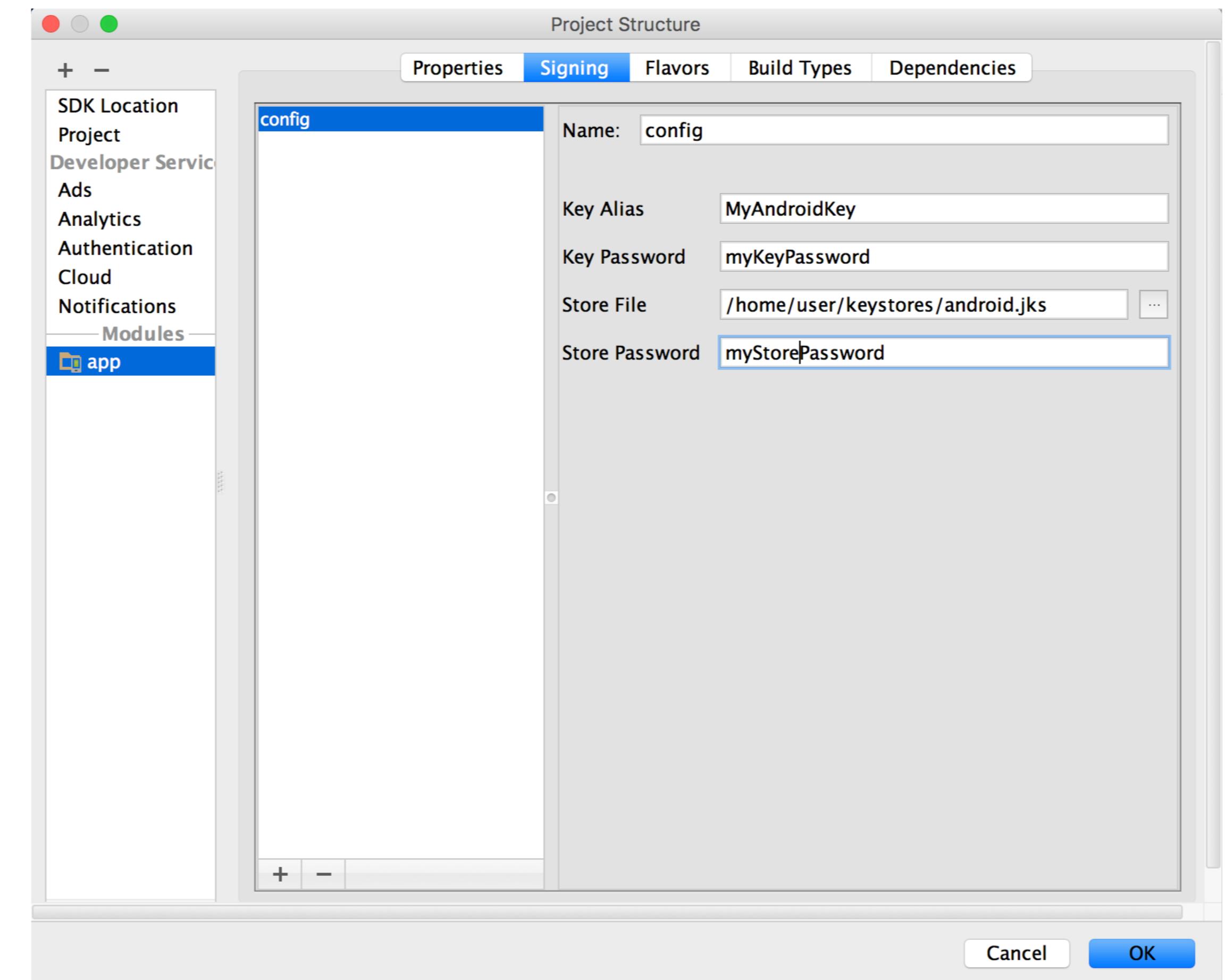


Sign an APK



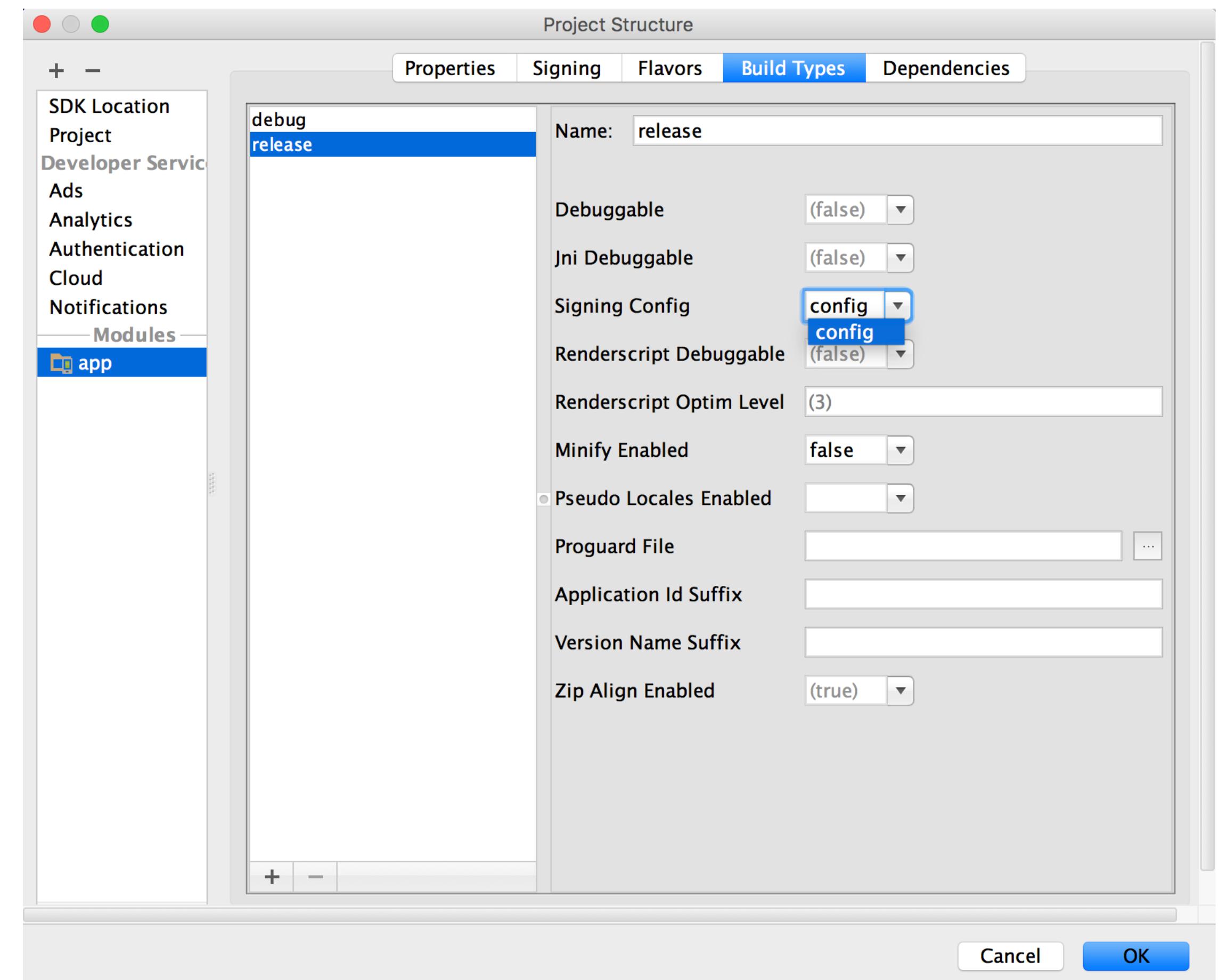
Automatically sign an APK

1. In the **Project** window, right click on your app and click **Open Module Settings**.
2. On the **Project Structure** window, under **Modules** in the left panel, click the module you would like to sign.
3. Click the **Signing** tab, then click **Add +**.
4. Select your **keystore** file, enter a name for this signing configuration (as you may create more than one), and enter the required information.

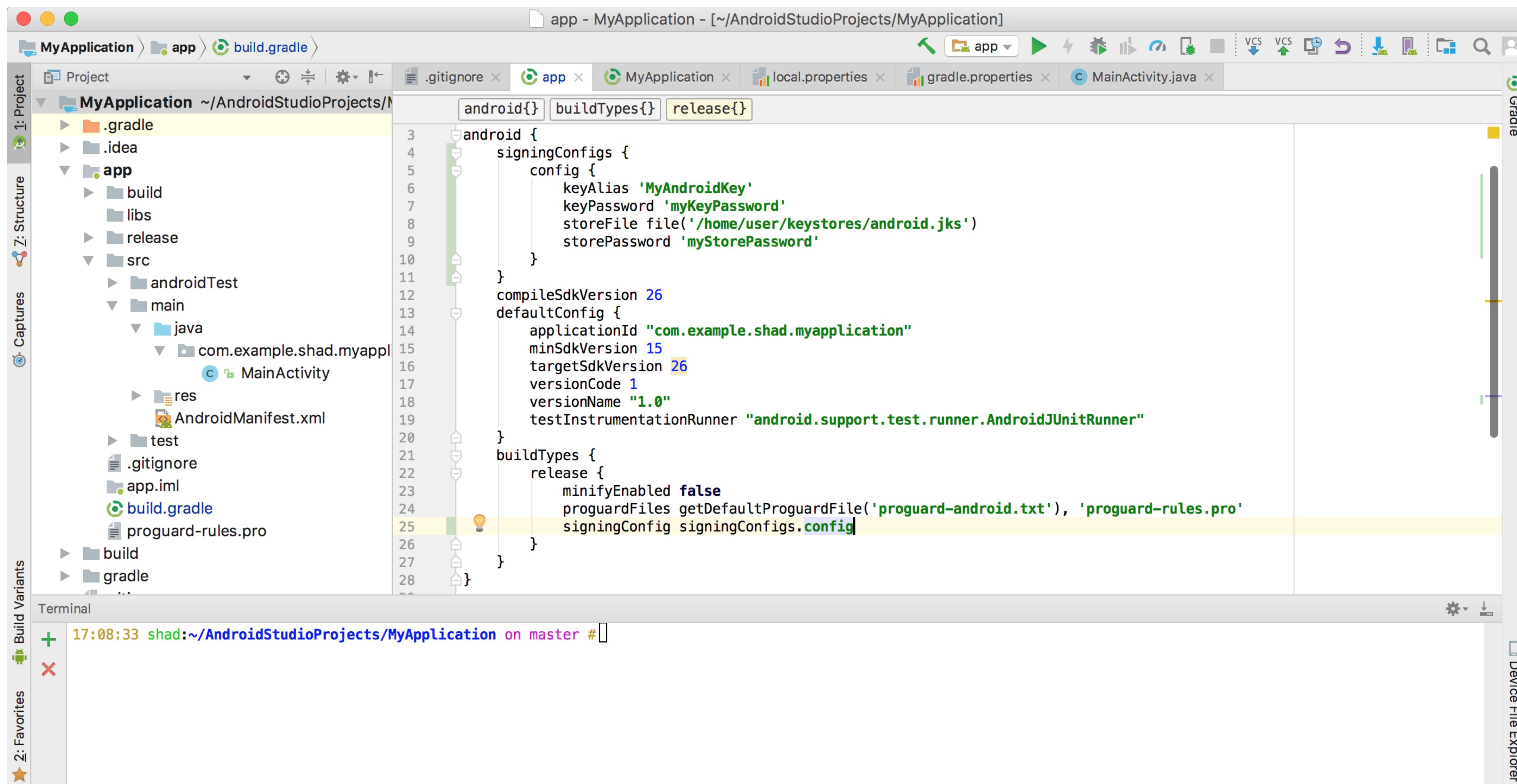


Automatically sign an APK

5. Click the **Build Types** tab.
6. Click the **release** build.
7. Under **Signing Config**, select the signing configuration you just created.
8. Click **OK**.



Automatically sign an APK



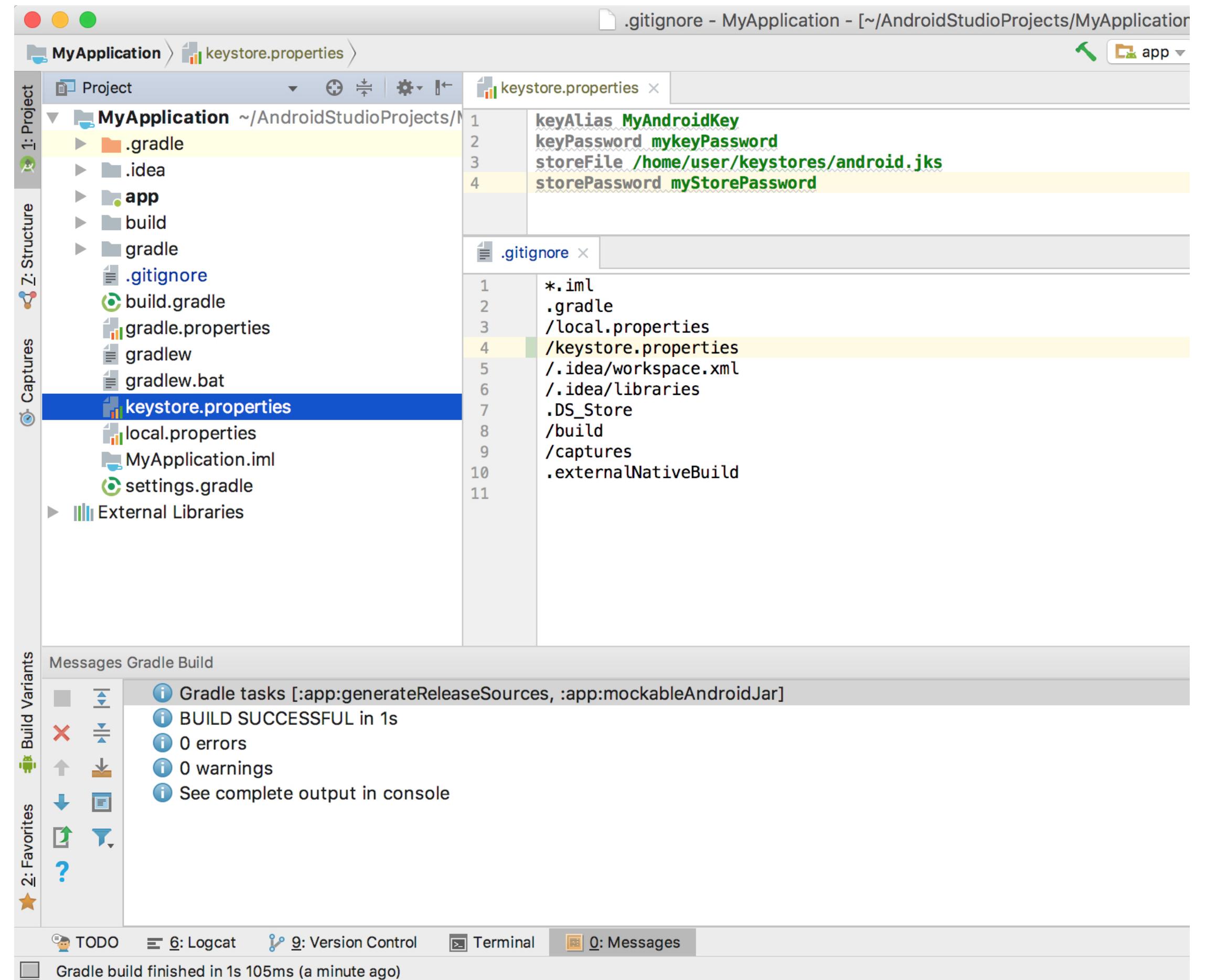
The screenshot shows the Android Studio interface with the project 'MyApplication' open. The left sidebar displays the project structure, including the app module which contains build, libs, release, src (with androidTest and main), res, test, .gitignore, app.iml, build.gradle, and proguard-rules.pro. The main editor window shows the build.gradle file with the following content:

```
3 android {
4     signingConfigs {
5         config {
6             keyAlias 'MyAndroidKey'
7             keyPassword 'myKeyPassword'
8             storeFile file('/home/user/keystores/android.jks')
9             storePassword 'myStorePassword'
10        }
11    }
12    compileSdkVersion 26
13    defaultConfig {
14        applicationId "com.example.shad.myapplication"
15        minSdkVersion 15
16        targetSdkVersion 26
17        versionCode 1
18        versionName "1.0"
19        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
20    }
21    buildTypes {
22        release {
23            minifyEnabled false
24            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
25            signingConfig signingConfigs.config
26        }
27    }
28}
```

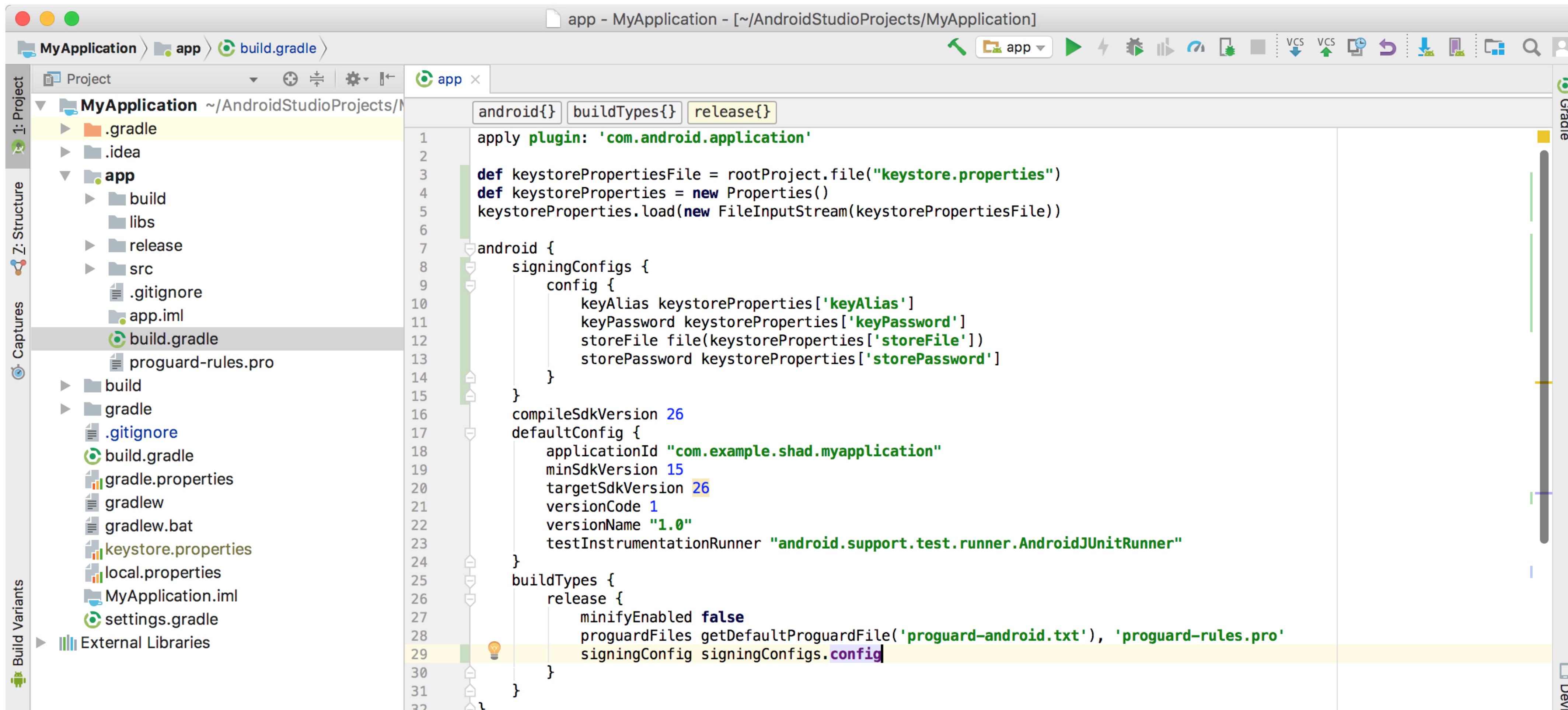
The code editor has syntax highlighting and a code completion tooltip visible at line 25. The bottom terminal window shows the command 'shad:~/AndroidStudioProjects/MyApplication on master #'. The right sidebar includes tabs for Device File Explorer and Build Variants.

Hide Signing Information

1. Create a file named **keystore.properties** in the root directory of your project. This file should contain your signing information.
2. Be sure to keep the `keystore.properties` file secure. Add this file to **.gitignore**.



Hide Signing Information



```
1 apply plugin: 'com.android.application'
2
3 def keystorePropertiesFile = rootProject.file("keystore.properties")
4 def keystoreProperties = new Properties()
5 keystoreProperties.load(new FileInputStream(keystorePropertiesFile))
6
7 android {
8     signingConfigs {
9         config {
10            keyAlias keystoreProperties['keyAlias']
11            keyPassword keystoreProperties['keyPassword']
12            storeFile file(keystoreProperties['storeFile'])
13            storePassword keystoreProperties['storePassword']
14        }
15    }
16    compileSdkVersion 26
17    defaultConfig {
18        applicationId "com.example.shad.myapplication"
19        minSdkVersion 15
20        targetSdkVersion 26
21        versionCode 1
22        versionName "1.0"
23        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
24    }
25    buildTypes {
26        release {
27            minifyEnabled false
28            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
29            signingConfig signingConfigs.config
30        }
31    }
32}
```