

Singing your android application

<http://developer.android.com/tools/publishing/app-signing.html>

Adding Google Maps to the android application

1. To use google maps, install google play services. In Eclipse :

Window -> Android SDK Manager -> Extras -> Google Play services

2. Import Google Play services library. In Eclipse :

File -> Import -> Android -> Existing Android Code into Workspace

Then reach the path where your android map sdk is installed and traverse to the location :
extras/google/google_play_services/libproject/google-play-services_lib.

NOTE : *It the case of CAVU2, we should use android-maps-extension instead of the original maps.*

You can download the android-maps-extension project from

<https://code.google.com/p/android-maps-extensions> or

<http://androidmapsextensions.com>

The tested version, used for CAVU2 app, is android-maps-extensions-1.7.0 . You can download the android-maps-extensions-1.7.0 from

<ftp://ftp.dev.videonext.net/cavu/android/libraries>

Open the android-maps-extension as a project in a current workspace.

In the "android-maps-extension" project go to Properties -> Android and check it as a library.

In the "CAVU Android" Project go to Properties -> Android -> Library, add the android-maps-extension library from the drop down list.

Make sure in "CAVU Android" AndroidManifest.xml file the minimum sdk version is 15!

3. Get Google Map Api Key

An API key is required to work with google maps. The API key is based on the SHA1 fingerprint. SHA1 fingerprint should be obtained in a result of a special request in a command line (cmd.exe for Windows, terminal for Mac).

The command line to obtain a **debug** API key (recommended for code development):

In Linux-

```
keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey -storepass android -keypass android
```

In Windows-

```
keytool -list -v -keystore "C:\Users\your_user_name\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
```

Enter your username in place of your_user_name

The command line to obtain a **release** API key (recommended for application which should be delivered on a device NOT from usb port):

```
keytool -list -v -keystore "C:\Users\your_user_name\.android\my-release-key.keystore" -alias cavukeystore -storepass android -keypass android
```

Enter your previously created application keystore in place of my-release-key

IMPORTANT NOTE : The keys must be used correctly : Debug API Key for development/debug mode, and the Release API key for the application which is going to the production, as otherwise, the application map screen(s) is not showing maps . A user sees the white screen instead.

The result looks like the example below :

```
=====
Alias name: cavukeystore
Creation date: Jul 5, 2012
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=zlata fomenko
Issuer: CN=zlata fomenko
Serial number: 4ff51337
Valid from: Thu Jul 05 00:08:23 EDT 2012 until: Mon Jun 29 00:08:23 EDT 2037
Certificate fingerprints:
    MD5: A0:95:00:34:14:39:BB:6A:47:CC:5B:81:00:B5:77:7F
    SHA1: 7B:9A:89:58:79:82:14:80:30:2A:98:E9:5D:88:E0:D4:E2:71:CE:EE
    Signature algorithm name: SHA1withRSA
    Version: 3
=====
```

We will need a SHA1 value for the next step.

4. Register the application with the Google APIs Console

Reach the Google APIs Console from the developer's google account by url :
<https://code.google.com/apis/console/>

If you still didn't create your project on a console, do it now.

Open your project.

Select **APIs& auth** and activate the **Google Maps Android API v2**.

Select **APIs& auth** -> **Credentials** to obtain keys.

If you need a new key, create **New Key -> Android key**. Copy SHA1 certificate fingerprint (you obtained it using keytool) in the field, add the package name separated by semicolon to the end . An example :

3C:3C:33:60:1D:77:10:0D:50:FC:A2:05:32:82:D0:E7:FD:DF:12:0A;com.videonext.cavu.api371

Create a key.

In our case, both API Keys (debug and release) were already created.

The Debug API key:

AlzaSyA3qlsX3GJ9z7gnrU3VIVenO4Jo4CmYjCg

The Release API key :

AlzaSyDWo8RoX9pe8VDkObMiLC30JlrYZPPTb2o

Any of these keys may be replaced by a new, properly created key.

Add an obtained API Key in the AndroidManifest.xml.

AndroidManifest.xml should contain following, marked bold, inside an application tag :

```
<application
    android:allowBackup="true"
    android:icon="@drawable/fancy_blue_on_black"
    android:label="@string/app_name"
    android:theme="@style/CustomTheme" >
    <meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="AlzaSyDWo8RoX9pe8VDkObMiLC30JlrYZPPTb2o" />
    .....
</application>
```

Also, following permissions should be added in the manifest :

```
<permission
```

```
    android:name="com.videonext.cavu.permission.MAPS_RECEIVE"  
    android:protectionLevel="signature" >  
</permission>
```

```
<uses-permission android:name="com.videonext.cavu.permission.MAPS_RECEIVE" />  
<uses-permission  
android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
```