



Getting your Google API v2 key for BT Maps (Android)

Preface

The new version of Android is coming! And with it will come a new slew of different improvements, one which everyone seems to be waiting for; the Maps.

Once you have the newest version for Android, you'll need to obtain a Google 'API Key' to utilize maps in your application (this applies whether Android or iOS).

This tutorial shows you the basics of how to get an API key and insert it into your Android BT Project.

We assume a few of the following circumstances:

- 1) You're somewhat familiar with the Eclipse environment.
- 2) You already have a [Google developer account](#). If you don't, get one. They're free and easy to obtain.
- 3) You already installed the Eclipse Keytool. If you haven't, do so. A good introductory link for that information is [located here](#).
- 4) Read through this document, and picture yourself performing the steps. This does two things; familiarize yourself with the tasks required, and allows you to realize what resources you'll need to complete the tasks. Make a list, prepare your resources. Then perform the tasks.

I can only hope that Eclipse for the PC is very similar to Eclipse for the Macintosh. I've done this on my Macintosh. Apologies in advance if somethings don't seem right on a PC. I don't develop BT stuff on my PC's.



Buzztouch project AndroidManifest.xml

```
51 <!-- application -->
52 <application android:name="BT_appDelegate"
53             android:icon="@drawable/icon"
54             android:label="@string/app_name"
55             android:theme="@style/hostThemeNoActionBar"
56             android:debuggable="true">
57
58
59     <!-- Google Cloud Messaging Project Number
60          Replace "YOUR_GOOGLE_GCM_PROJECT_NUMBER"
61          See http://developer.android.com/google/gcm/
62          -->
63     <meta-data android:name="googleCloudMessagingProjectNumber"
64               android:value="YOUR_GOOGLE_GCM_PROJECT_NUMBER"/>
65
66     <!-- Google Maps v2 API Key
67          Replace "GOOGLE_MAPS_FOR_ANDROID_V2_API_KEY_GOES_HERE" on the next line with the Google Maps for Android API Key provided
68          See: https://developers.google.com/maps/documentation/android/start#installing_the_google_maps_android_v2_api
69          -->
70     <meta-data android:name="com.google.android.maps.v2.API_KEY" android:value="GOOGLE_MAPS_FOR_ANDROID_V2_API_KEY_GOES_HERE"/>
71
72     <!-- Google Cloud Messaging -->
73     <receiver android:name="com.buzzTouch.BT_gcmReceiver"
74             android:permission="com.google.android.c2dm.permission.SEND" >
75         <intent-filter>
76             <action android:name="com.google.android.c2dm.intent.RECEIVE" />
77             <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
78         </intent-filter>
79     </receiver>
80
81     <!-- Activity, it may or may not load a splash screen fragment (alwain) -->
82     <activity android:name="com.buzzTouch.MainActivity"
83             android:label="@string/app_name" android:configChanges="keyboardHidden|orientation"
84             android:launchMode="singleTask" android:windowSoftInputMode="adjustResize">
85         <intent-filter>
86             <action android:name="android.intent.action.MAIN" />
87             <category android:name="android.intent.category.LAUNCHER" />
88         </intent-filter>
89     </activity>
```

Like it or not, Google is your easiest source for maps. That said, you have to register as a Developer, and generate an API Key in order to use them. This will attempt to help you.

This will be your AndroidManifest.xml file, in the root of your Android project. View it as an XML File by clicking on the XML Tab

If you want to use Google Maps in your project, you'll have to get an API key. Since API v1 keys has been deprecated and are no longer issued, we must now move on to API v2. Not a problem, just a little different than the way it used to be. Stay close and pay attention, and you'll be fine.



Google Developer Website

Google Maps Android API v2 — Google Developers

https://developers.google.com/maps/documentation/android/

Google Developers

Google Maps Android API v2 X Search

smugwimp@gmail.com Sign out

Home Products Conferences Showcase Live Groups

Google Maps Android API v2 744 Feedback on this document

Developer Guide

- Introduction
- Getting Started
- Map Objects
- ▶ Drawing on the Map
- Interacting with the map
- Location Data
- Changing the View

▶ API Reference

- Blog
- Maps API Forum
- FAQ
- Releases
- Terms of Service

Google Maps Android API v1 Deprecated

Google Maps Android API v2

The new Google Maps Android API v2 allows you to offer interactive, feature-rich maps to users of your Android application. This version offers the following improvements:

- The API is now distributed as part of the [Google Play services SDK](#), which you can download with the Android SDK Manager. To learn how to install the package, see [installing the Maps API SDK](#).
- Maps are now encapsulated in the [MapFragment](#) class, an extension of Android's [Fragment](#) class. Now you can add a map as a piece of a larger [Activity](#). With a [MapFragment](#) object, you can show a map by itself on smaller screens, such as mobile phones, or as a part of a more complex UI on larger-screen devices, such as tablets.
- Because maps are encapsulated in the [MapFragment](#) class, you can implement them by extending the Android standard [Activity](#) class, rather than extending the [MapActivity](#) used in version 1.
- The Maps API now uses vector tiles. Their data representation is smaller, so maps appear in your apps faster, and use less bandwidth.
- Caching is improved, so users will typically see a map without empty areas.

[Learn more](#)

Login to your account at the Google Developers Website (developers.google.com). When you do, click on the link 'Getting Started'.



Getting Started with the Google API v2

The screenshot shows the Google Developers website for the Google Maps Android API v2. The page title is "Getting Started". The main content area contains a list of links for navigation, with "Obtaining an API Key" highlighted in yellow. A purple arrow points from a circled "1" to this link. Below the links is an "Overview" section with a list of steps for getting started.

Getting Started

Before you can begin working with the API, you will need to download the API and ensure that you have a Google Maps Android API v2 key. Both the API and the key are freely available.

- [Overview](#)
- [Getting the Google Maps Android API v2](#)
- [The Google Maps API Key](#)
- [Displaying certificate information](#)
- [Creating an API Project](#)
- [Obtaining an API Key](#)
- [Adding the API Key to your application](#)
- [Specify settings in the Application Manifest](#)
- [Specifying permissions](#)
- [Requiring OpenGL ES version 2](#)
- [Add a Map](#)

Overview

Creating a new Android application that uses the Google Maps Android API v2 requires several steps. Many of the steps outlined in this section will only have to be performed once, but some of the information will be a handy reference for future applications. The overall process of adding a map to an Android application is as follows:

1. Download and configure the [Google Play services](#) SDK. The Google Maps Android API is distributed as part of this SDK. Users of the [Google Maps Mobile SDK for Business](#) must download and configure the Google Maps Mobile SDK for Business static library.
2. [Obtain an API key](#). To do this, you will need to register a project in the Google APIs Console, and get a signing certificate for your app.
3. [Specify settings](#) in the Application Manifest.
4. Add a map to a new or existing Android project.

1) Click on 'Obtaining an API Key' (or scroll down to the subsection manually)



Obtaining an API Key Subsection

Google Maps Android API v2 — Google Developers

https://developers.google.com/maps/documentation/android/start#obtaining_an_api_key

good idea to use a new project for Google Maps Android API, so select the project name in the upper left hand corner and then click **Create**.

2. You should see a list of APIs and services in the main window. If you don't, select **Services** from the left navigation bar.
3. In the list of services displayed in the center of the page, scroll down until you see **Google Maps Android API v2**. To the right of the entry, click the switch indicator so that it is **on**.
4. This displays the [Google Maps Android API Terms of Service](#). If you agree to the terms of service, click the checkbox below the terms of service, then click **Accept**. This returns you to the list of APIs and services.

You're now ready to get a Maps API key.

Obtaining an API Key

If your application is registered with the Google Maps Android API v2 service, then you can request an API key. It's possible to register more than one key per project.

To get the key:

1. Navigate to your project in the [Google APIs Console](#).
2. In the left navigation bar, click **API Access**.
3. In the resulting page, click **Create New Android Key...**
4. In the resulting dialog, enter the SHA-1 fingerprint, then a semicolon, then your application's package name. For example:

```
BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75;com.example.android.mapexample
```
5. The Google APIs Console responds by displaying **Key for Android apps (with certificates)** followed by a forty-character API key, for example:

```
AIzaSyBdV1-cTICSwYKr295SuvNw7dbMuDt1KG0
```
6. Copy this key value. You will use it in the next step.

Adding the API Key to your application

The final step is to add the API key to your application. It goes in your application's manifest, contained in the file `AndroidManifest.xml`. From there, the Maps API reads the key value and passes it to the Google Maps server, which then confirms that you have access to Google Maps data.

To add the key to your application:

1. In `AndroidManifest.xml`, add the following element as a child of the `<application>` element, by inserting it just before the closing tag `</application>`:

```
<meta-data
  android:name="com.google.android.maps.v2.API_KEY"
```

Click on 'Navigate to your project in the Google API Console'



Google Developer Dashboard

Google APIs Console

Search Images Maps Play YouTube News Gmail Drive More

smugwimp@gmail.com | Settings | Help | Sign out

Google apis

API Project

Overview
Services
Team
API Access
Billing
Reports
Quotas

Dashboard

Project Summary

Name	API Project
Project Number	2
Project ID	mjpgsguam001
Google+ Page	Request connection
Owners	smugwimp@gmail.com - you
Current charges	Click here to administer your billing settings...

Service

Service	Status
Google Maps API v2	No known issues
Google Maps API v3	No known issues
Google Maps Coordinate API	No known issues
Google Maps SDK for iOS	No known issues

[Code Home](#) - [Privacy Policy](#)

Send Feedback

- 1) You should arrive at your API 'dashboard'. You'll see a summary of your current Google Service subscriptions, and the status of them.
- 2) Click on the link marked 'API Access'



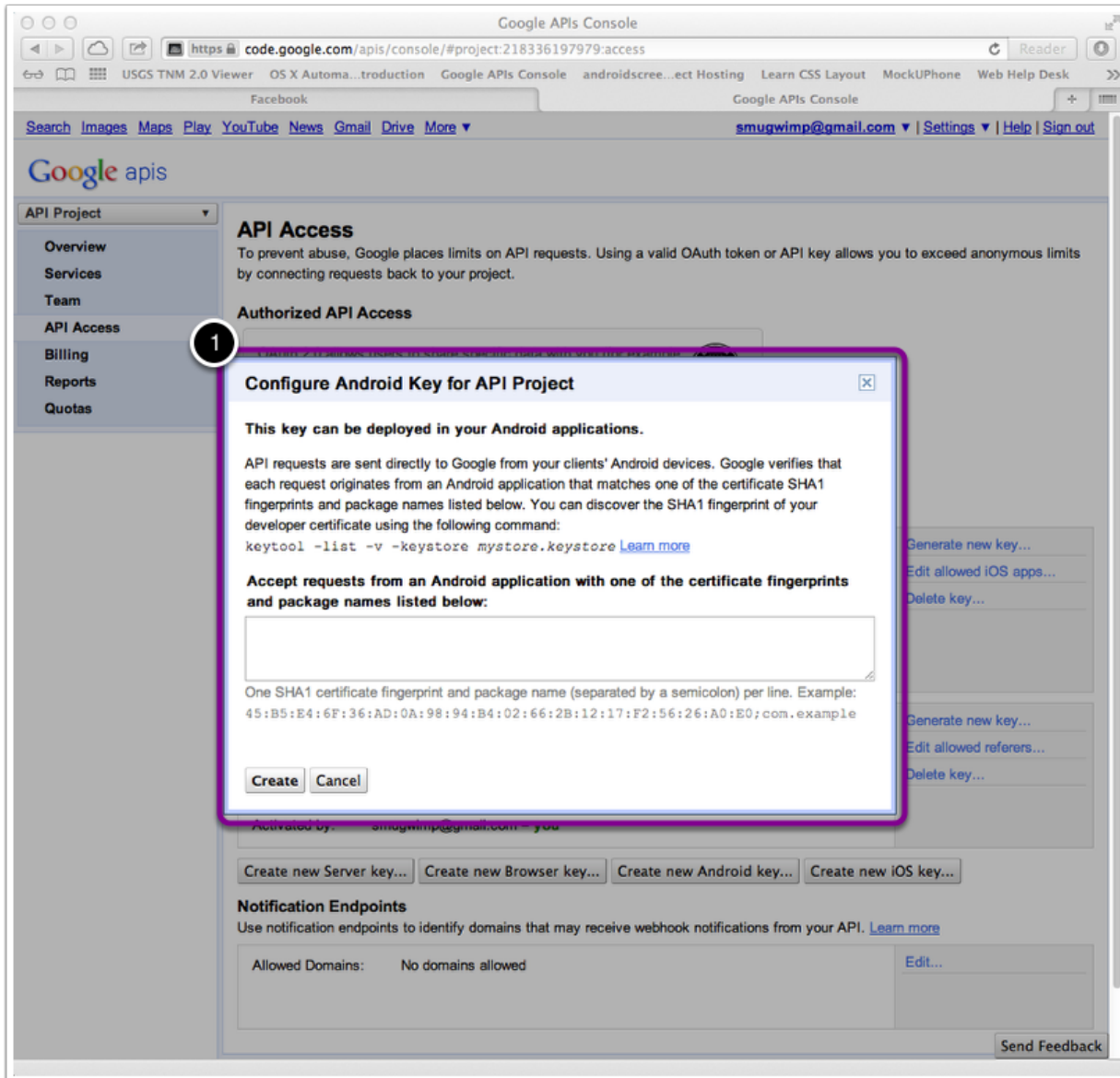
Google Developer API Access Summary

The screenshot shows the Google APIs Console interface. The left sidebar contains navigation links: Overview, Services, Team, API Access (selected), Billing, Reports, and Quotas. The main content area is titled 'API Access' and includes a 'Create an OAuth 2.0 client ID...' button. Below this, there are two sections for API keys: 'Key for iOS apps (with bundle identifiers)' and 'Key for browser apps (with referers)'. Each section has a 'Generate new key...' button. At the bottom, there are buttons for 'Create new Server key...', 'Create new Browser key...', 'Create new Android key...' (highlighted in yellow), and 'Create new iOS key...'. Three numbered callouts (1, 2, 3) are placed over the image: callout 1 points to the 'Create new Android key...' button, callout 2 points to the 'Create new iOS key...' button, and callout 3 points to the 'Create new Android key...' button.

- 1) You should arrive at your API page. It will have a listing of your current API's (if available)
- 2) Click on the button marked 'Create new Android key...'
- 3) If you need an iOS key, it's this button. But for the moment, we're doing Android.



SHA1 Key and Project Name entry

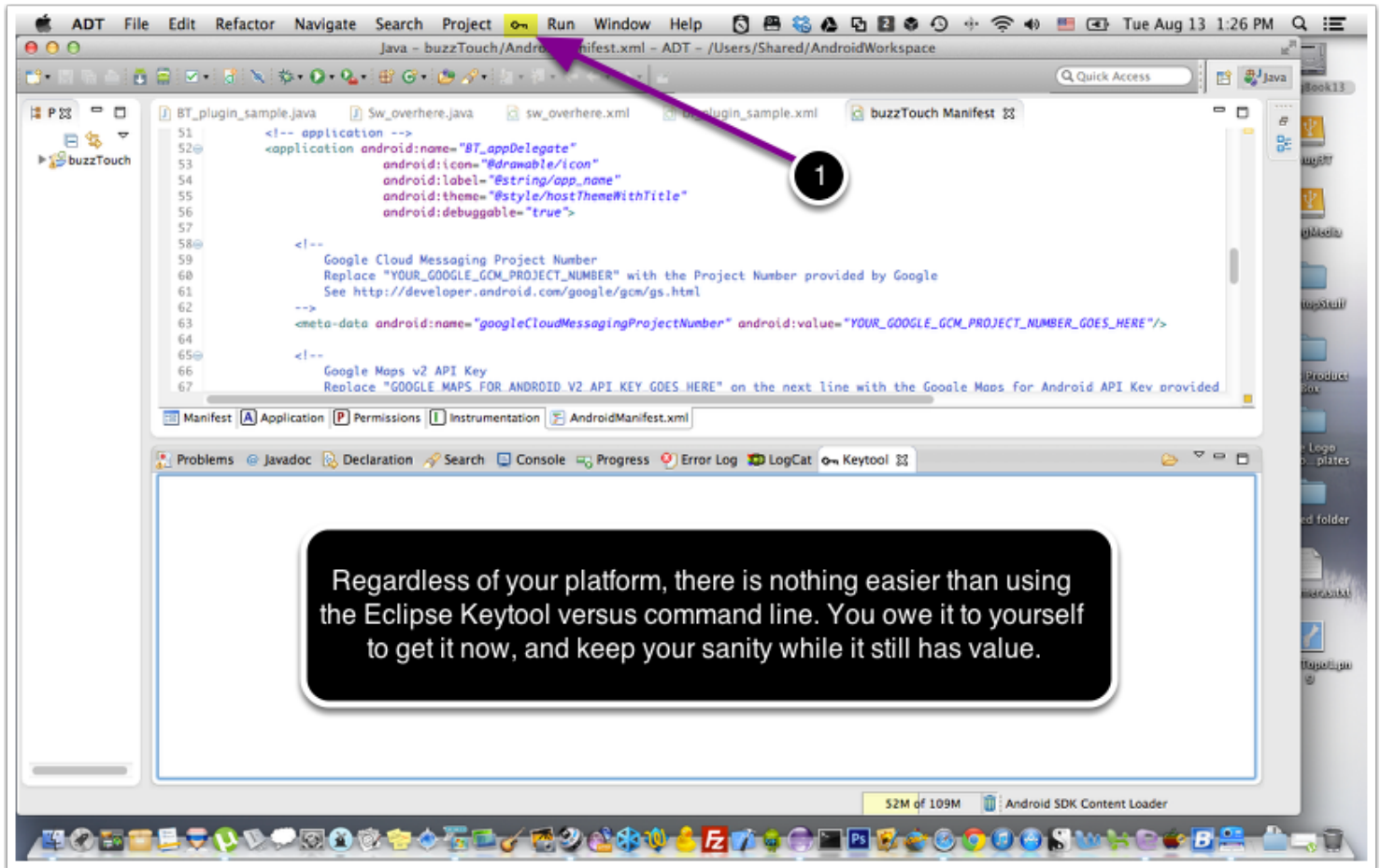


1) Ok this box is asking you for the SHA1 fingerprint for your (intended) Android application.

So for a minute, let's move over to Eclipse, where I hope you've already installed the Eclipse Keytool. If you haven't done it already, step back and complete that before proceeding. If you need instruction, Mark (GoNorthWest) did an excellent job and details can be found in this [BT Forum Post](#).



Eclipse Keytool



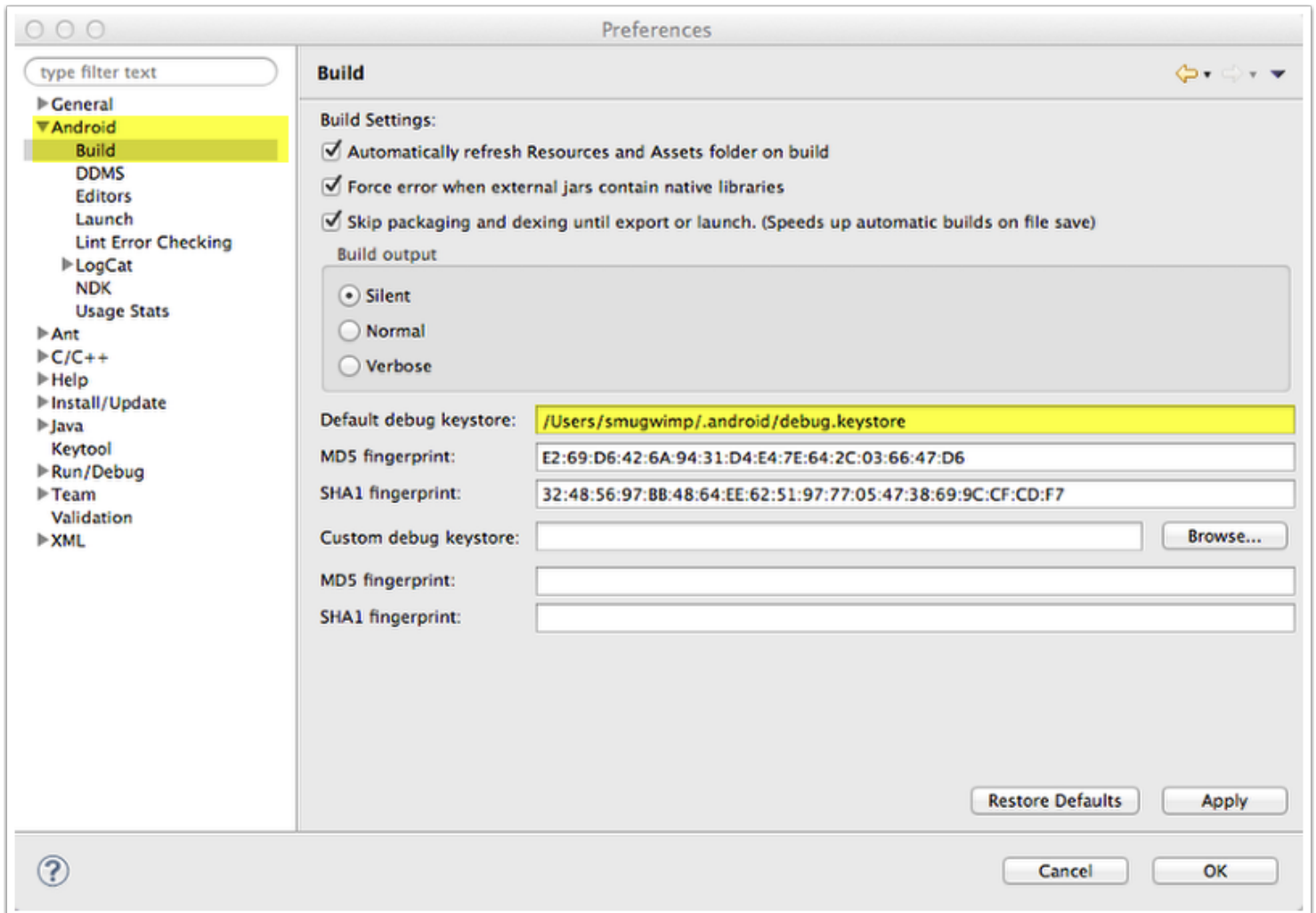
Click on the key tool menu item.

- 1) If you need to create and open a keystore, do so.
- 2) If you just need to open a keystore, do so.

Either way, you'll need to open your keystore. You'll have two; Debug and Release. It really depends on you, but I typically go ahead and install my 'Release' keys into my apps, because it's one less thing to worry about later. And sometimes dealing with so many API keys, the boundary of 'Debug' and 'Release' can get confusing. It's up to you.



Forget Waldo. Where is my debug and release keystore kept at?



On the macintosh, the android keystores are in a hidden folder named '.android' which is located within your user directory (the same spot as documents, videos, music, etcetera, but you usually cannot see it. However, you can navigate to it using the keytool - go to your user directory and look inside the '.android' directory. Your debug keystore will be named 'debug.keystore' and your release keystore will be named whatever 'you' named it, if you have created one already. Otherwise when you do create one, just call it 'release.keystore'. It makes it easier on you later when you're looking again.

On the PC it depends on your Windows Version.

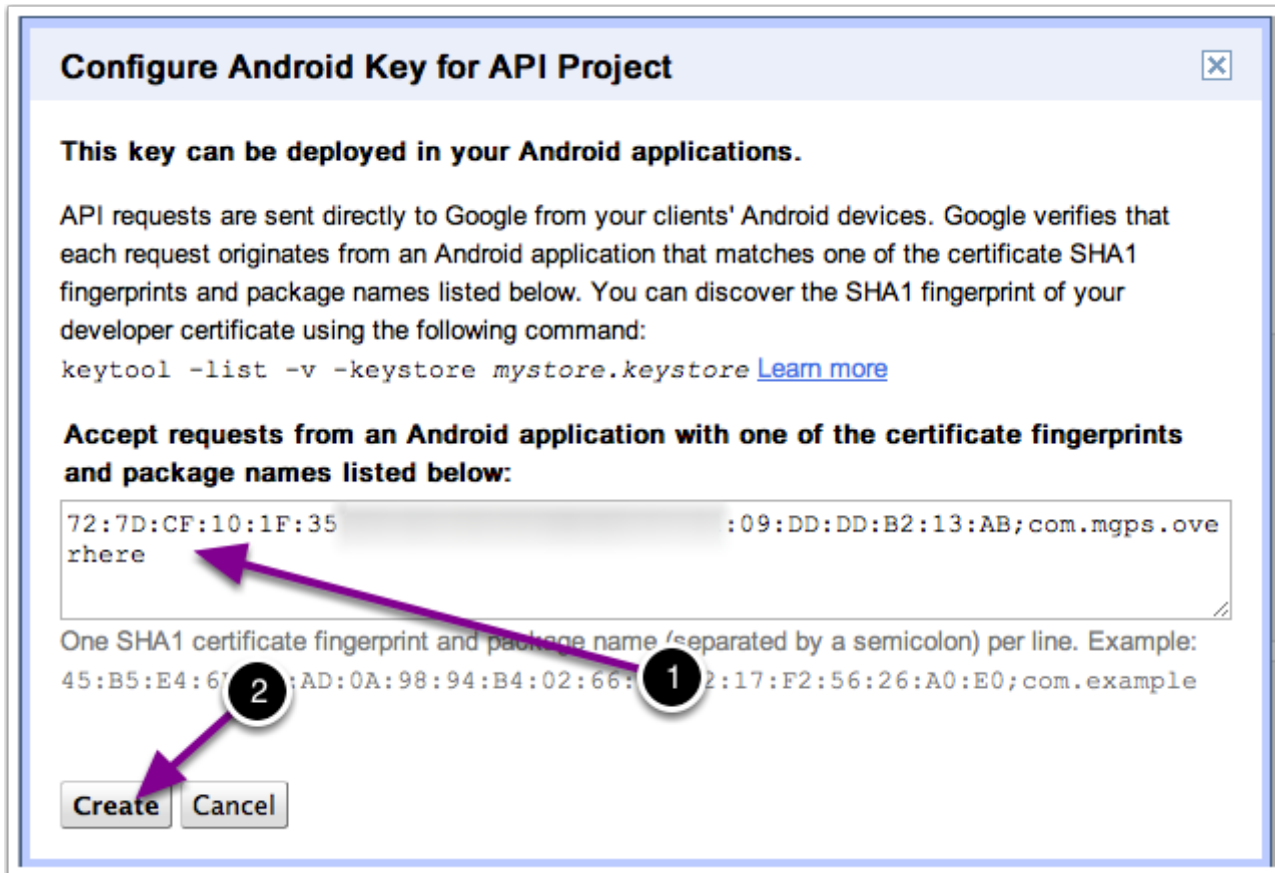
In XP it 'should' be located at C:\Documents and Settings\[User Name]\.android

In Vista/7+ it 'should' be located at C:\Users\.android

Regardless of platform, if you go to your Eclipse preferences, under the 'Android > Build' section, you should have a field with the location of your default keystore location.



Entering your SHA1 Fingerprint and App Name



- 1) Paste your SHA1 Fingerprint in the dialog box. Add a semicolon (;) and the name of your app (com.something).
- 2) When you're finished, Press the 'Create' button



Viewing your newly generated Google Map API Key for Android

The screenshot shows the Google APIs Console interface. The left sidebar contains navigation links: Overview, Services, Team, API Access (selected), Billing, Reports, and Quotas. The main content area is titled 'API Access' and includes a 'Create an OAuth 2.0 client ID...' button. Below this, there are three sections for API keys:

- Key for Android apps (with certificates)**:
 - API key: AIzaSyBx9n...Y51tm-PgIt8
 - Android apps: 72:7D:CF:10:1F:35:F5:08:31:09:DD:DD:B2:13:AB;com.mgps.overhere
 - Activated on: Aug 12, 2013 8:41 PM
 - Activated by: smugwimp@gmail.com - you
- Key for iOS apps (with bundle identifiers)**:
 - API key: AIzaSyDVY7Ed...oo2joIaHk
 - iOS apps: com.mgps.google, com.mgps.oog
 - Activated on: Mar 12, 2013 1:20 PM
 - Activated by: smugwimp@gmail.com - you
- Key for browser apps (with referers)**:
 - API key: AIzaSyBw9Ji...30wC6Idtzo
 - Referers: Any referer allowed
 - Activated on: Aug 12, 2012 10:51 PM
 - Activated by: smugwimp@gmail.com - you

At the bottom of the console, there are buttons for 'Create new Server key...', 'Create new Browser key...', 'Create new Android key...', 'Create new iOS key...', and 'Send Feedback'.

There you go! Just copy the API Key in the New record of your API Summary page...



Pasting your API key into the project AndroidManifest.xml file within Eclipse

```
51 <!-- application -->
52 <application android:name=".BT_appDelegate"
53             android:icon="@drawable/icon"
54             android:label="@string/app_name"
55             android:theme="@style/HostThemeWithTitle"
56             android:debuggable="true">
57
58
59     <!-- Google Cloud Messaging Project Number
60          Replace "YOUR_GOOGLE_GCM_PROJECT_NUMBER" with the Project Number provided by Google
61          See http://developer.android.com/google/gcm/gs.html
62     -->
63     <meta-data android:name="googleCloudMess" android:value="YOUR_GOOGLE_GCM_PROJECT_NUMBER_GOES_HERE"/>
64
65
66     <!-- Google Maps v2 API Key
67          Replace "GOOGLE_MAPS_FOR_ANDROID_V2_API_KEY_GOES_HERE" on the next line with the Google Maps for Android API Key provided
68          See: https://developers.google.com/maps/documentation/android/v2/#installing_the_google_maps_android_v2_api
69     -->
70     <meta-data android:name="con.google.android.maps.v2.API_KEY" android:value="AIzaSyBx9ms7...o0YS1tm-PgIt8"/>
71
72
73     <!-- Google Cloud Messaging -->
74     <receiver android:name="com.buzzTouch.BT_gcmReceiver"
75             android:permission="com.google.android.c2dm.permission.SEND" >
76         <intent-filter>
77             <action android:name="com.google.android.c2dm.intent.RECEIVE" />
78             <action android:name="com.google.android.c2dm.intent.REGISTRATION" />
79             <category android:name="com.buzzTouch" />
80         </intent-filter>
81     </receiver>
82     <service android:name=".BT_gcmIntentService" />
83
84     <!-- BT_activity_start is the default, beginning activity, it may or may not load a splash screen fragment (alugin) -->
85     <activity android:name=".BT_activity_start" android:label="@string/app_name" android:configChanges="keyboardHidden|orientation"
86             <intent-filter>
87                 <action android:name="android.intent.action.MAIN" />
88                 <category android:name="android.intent.category.LAUNCHER" />
89             </intent-filter>
90     </activity>
```

- 1) Paste your newly acquired API Key between the quotation marks, replacing the 'holder' text.
- 2) **Remember** if you generated a 'Release' key, then you need to generate 'Release' builds for the API to correctly operate. Same principle applies if you generated a 'Debug' build of the API key.