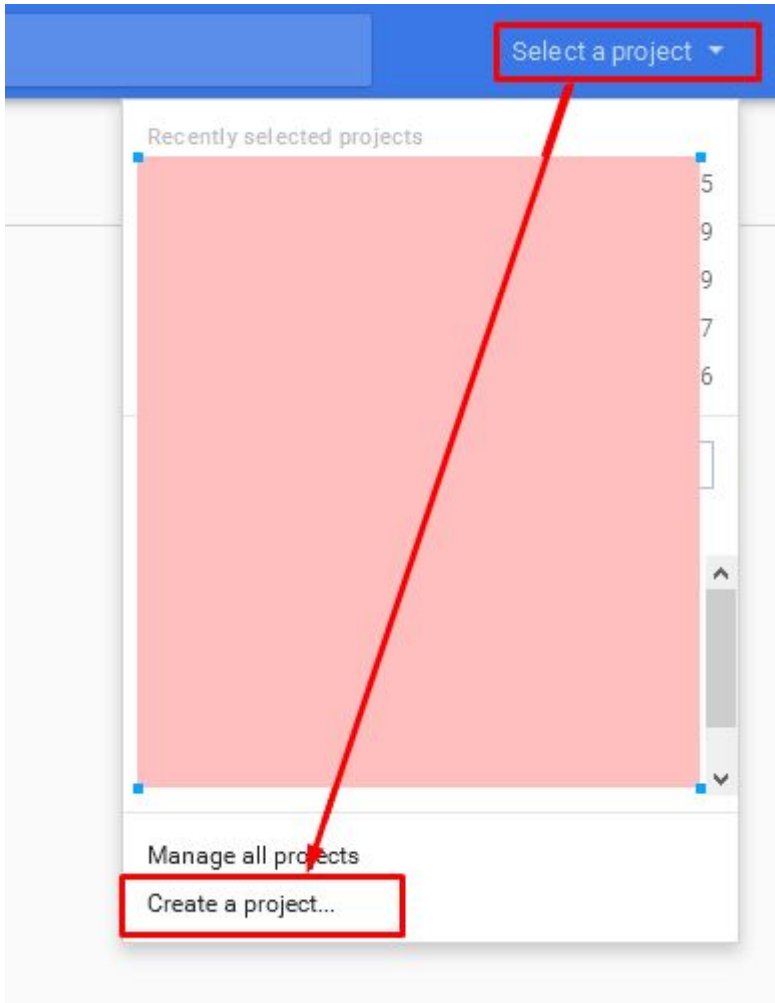


Open <https://console.developers.google.com>

Create new project



New Project

Project name

Demo project

Your project ID will be vast-maxim-121616 [Edit](#)

[Show advanced options...](#)

Create

Cancel

Manage API-s

The image shows the Google Developers Console interface. At the top, the header reads "Google Developers Console" with a search bar and a "Demo project" dropdown. The left sidebar contains "Home", "Dashboard" (highlighted with a red box), and "Activity". The main content area is titled "Dashboard" and features several blue cards: "Use Google APIs" (with a sub-card "API Enable and manage APIs" also highlighted with a red box), and "Try Compute Engine". A dropdown menu is open on the right, showing "Recently selected projects" (empty) and "All projects" with a search filter. The "All projects" list contains one entry: "Demo project" with ID "vast-maxim-121616", which is checked and highlighted with a red box. Below the dropdown are sections for "Try App Engine environment" (with links for documentation and guided walkthroughs for Go, Java, PHP, and Python) and "Managed VM" (with a "Start the Node.js" link).

Google Developers Console

API Manager

Overview

Google APIs Enabled APIs (7)

Search all 100+ APIs

Popular APIs

- Google Cloud APIs**
 - Compute Engine API
 - BigQuery API
 - Cloud Storage Service
 - Cloud Datastore API
 - Cloud Deployment Manager API
 - Cloud DNS API
 - More
- Mobile APIs**
 - Google Cloud Messaging
 - Google Play Game Services
 - Google Play Developer API
 - Google Places API for Android
- Advertising APIs**
 - AdSense Management API
 - DCM/DFA Reporting And Trafficking API
 - Ad Exchange Seller API
 - Ad Exchange Buyer API
 - DoubleClick Search API
 - DoubleClick Bid Manager API
- Google Maps APIs**
 - Google Maps Android API
 - Google Maps SDK for iOS
 - Google Maps JavaScript API
 - Google Places API for Android
 - Google Places API for iOS
 - Google Maps Roads API
 - More
- Social APIs**
 - Google+ API
 - Blogger API
 - Google+ Pages API
 - Google+ Domains API
- Other popular APIs**
 - Analytics API
 - Translate API
 - Custom Search API
 - URL Shortener API
 - PageSpeed Insights API
 - Fusion Tables API
 - Web Fonts Developer API

Enable API:

Google Developers Console

API Manager

Overview

Enable API

Google Maps Android API

Add maps based on Google Maps data to your Android application with the Google Maps Android API. The API automatically handles access to Google Maps servers, map display and response to user gestures such as clicks and drags. [Learn more](#)

Using credentials with this API

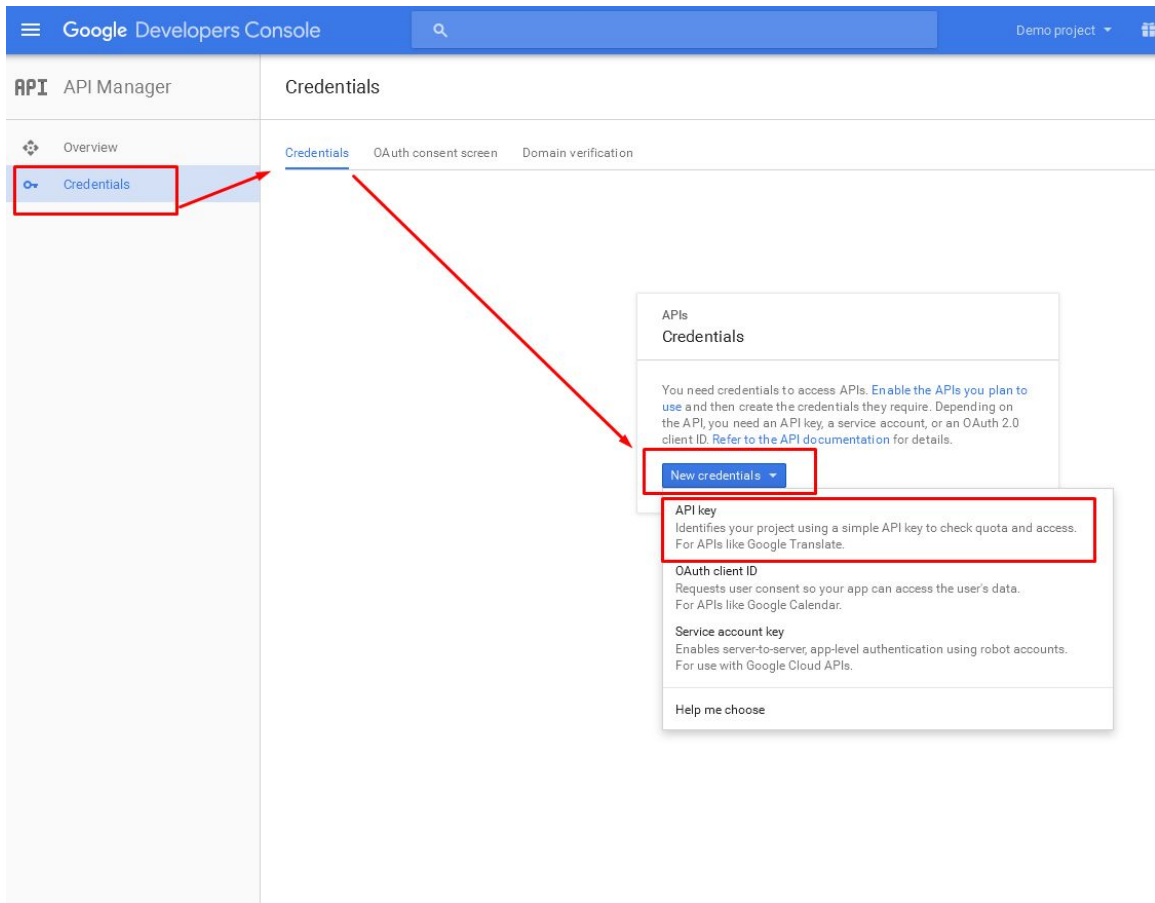
Using an API key

To use this API you need an API key. An API key identifies your project to check quotas and access. Go to the Credentials page to get an API key. You'll need a key for each platform, such as Web, Android, and iOS. [Learn more](#)

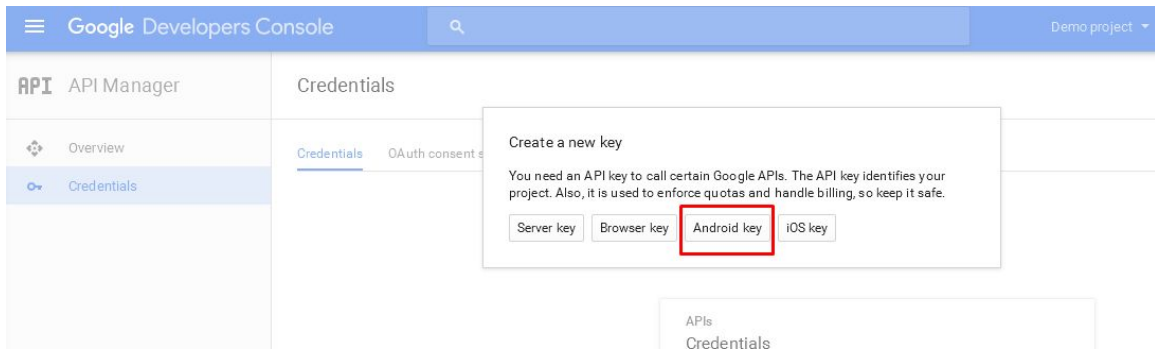
```

graph LR
    A[Your application] --> B[API key]
    B --> C[Google]
  
```

Create API key:

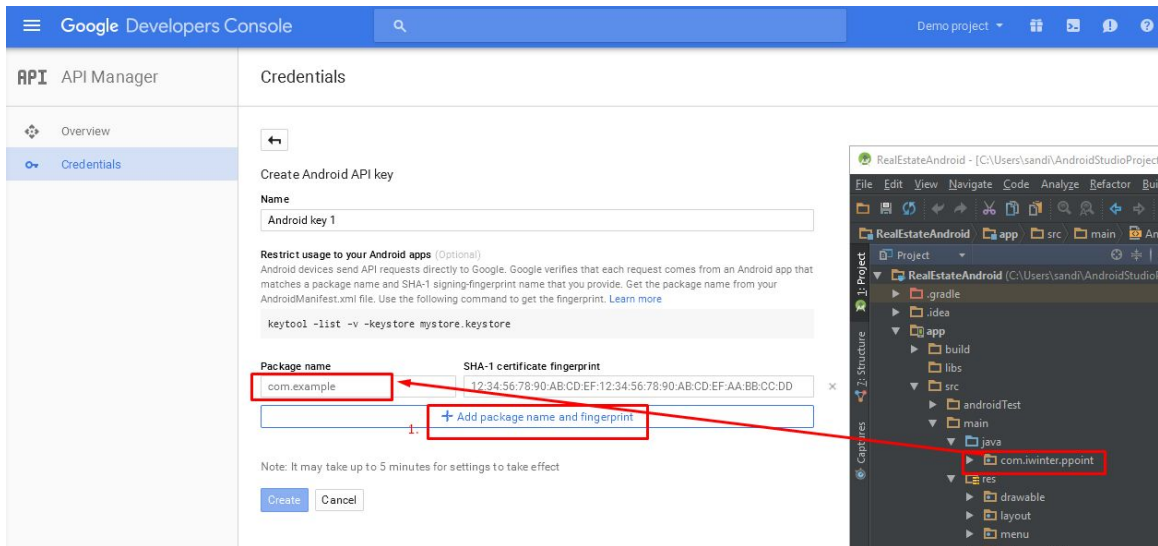


Of course Android key:



If you plan to release app on market then my suggestion is to change package name in project, guide is available in app documentation. In other case your app will become suspended from market very soon.

Enter package name:

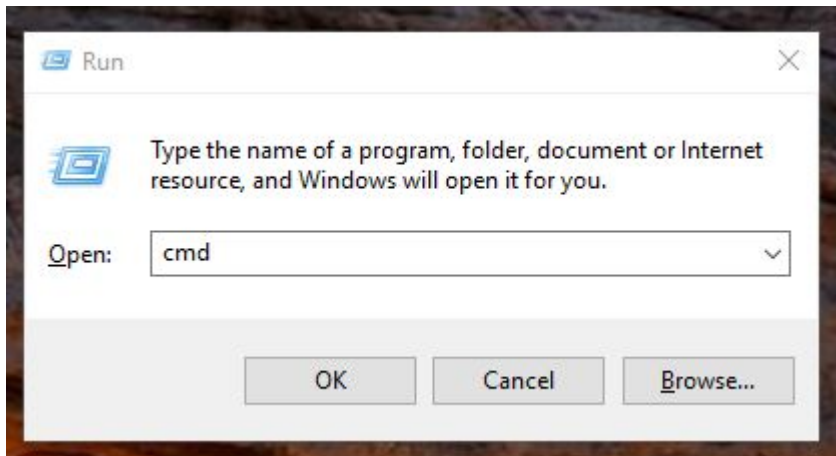


SHA1 should be generated on your PC via terminal/command prompt

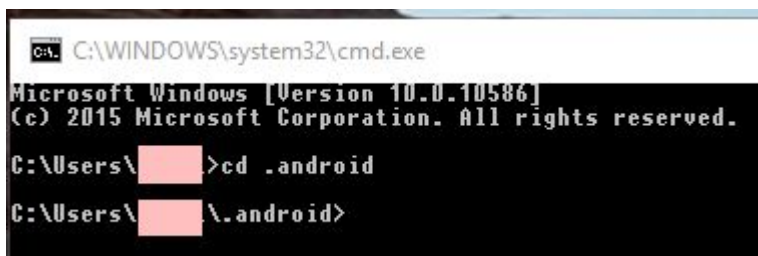
I will give example for debug version (for google market release you should follow guide by google)

You should find location of debug.keystore file in your system, on my PC it's located in C:\Users\{user}\.android, in my case {user} represents windows username

So open command prompt (Windows key + R), enter "cmd" and click ok



I'm automatically located in C:\Users\{user}, now just enter cd .android



With dir command you can check if it's located there:

```
C:\Users\ [redacted] \.android>dir
Volume in drive C has no label.
Volume Serial Number is 2E6E-6D1B

Directory of C:\Users\ [redacted] \.android

06.02.2016.  23:12    <DIR>          .
06.02.2016.  23:12    <DIR>          ..
04.02.2016.  13:10                12 .emu-update-last-check
26.10.2015.  13:10            1,708 adbkey
26.10.2015.  13:10             716 adbkey.pub
01.01.2016.  23:37             123 adb_usb.ini
09.02.2016.  13:25             138 androidwin.cfg
09.01.2016.  21:43    <DIR>          avd
01.01.2016.  22:59    <DIR>          cache
05.02.2016.  18:20             165 ddms.cfg
01.01.2016.  22:43            2,146 debug.keystore
01.01.2016.  23:02             784 default.keyset
04.02.2016.  13:10             171 modem-nv-ram-5554
08.01.2016.  14:16    <DIR>          monitor-workspace
05.02.2016.  17:56            2,233 re-android-key.jks
09.02.2016.  13:25             82 repositories.cfg
02.01.2016.  00:08            1,023 sites-settings.cfg
01.01.2016.  18:36    <DIR>          studio
                12 File(s)              9,301 bytes
                6 Dir(s)   7,534,075,904 bytes free
```

Now you can try to run command:

```
keytool -list -v -keystore debug.keystore
```

```
C:\Users\ [redacted] \.android>keytool -list -v -keystore debug.keystore
'keytool' is not recognized as an internal or external command,
operable program or batch file.
```

If you receive same error, then you should find where your keytool.exe program is located because windows doesn't know where it is...

In my system it's located in: C:\Program Files\Java\jre7\bin\keytool.exe

So I will type:

```

C:\Users\...\.android>"C:\Program Files\Java\jre7\bin\keytool.exe" -list -v -keystore debug.keystore
Enter keystore password:
***** WARNING WARNING WARNING *****
* The integrity of the information stored in your keystore *
* has NOT been verified! In order to verify its integrity, *
* you must provide your keystore password. *
***** WARNING WARNING WARNING *****

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: androiddebugkey
Creation date: 01.01.2016.
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 10cb3623
Valid from: Fri Jan 01 22:43:09 CET 2016 until: Sun Dec 24 22:43:09 CET 2045
Certificate fingerprints:
  MD5: CC:1B:61:63:6B:00:8D:76:C8:B7:CA:9E:EE:4A:77:85
  SHA1: 05:30:69:9E:7A:AF:A9:92:CC:2F:4F:3B:0C:8A:CA:A5:F8:32:0E:AF
  SHA256: F4:59:CB:72:49:E5:DD:AF:FF:8D:35:83:08:2D:09:19:FD:F6:81:DF:A1:BC:55:30:FE:18:93:13:47:CB:7D:4E
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B5 BA C8 DA 9C 8D 20 97  2D 7C 5B 67 88 C1 E9 35  ..... --[g...5
0010: 3F AF BA A1
]
]
*****
*****

```

no password, just press enter button on keyboard

You need this code

Now copy this code into developer console:

android:validateSigning (LEGACY)

Android devices send API requests directly to Google. Google verifies that each request comes from an Android app that matches a package name and SHA-1 signing fingerprint name that you provide. Get the package name from your AndroidManifest.xml file. Use the following command to get the fingerprint. Learn more

```
keytool -list -v -keystore mystore.keystore
```

Package name	SHA-1 certificate fingerprint
com.example	12:34:56:78:90:AB:CD:EF:12:34:56:78:90:AB:CD:EF:AA:BB:CC:DD

[+ Add package name and fingerprint](#)

Note: It may take up to 5 minutes for settings to take effect

```

Your keystore contains 1 entry
Alias name: androiddebugkey
Creation date: 01.01.2016.
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 10cb3623
Valid from: Fri Jan 01 22:43:09 CET 2016 until: Sun Dec 24 22:43:09 CET 2045
Certificate fingerprints:
  MD5: CC:1B:61:63:6B:00:8D:76:C8:B7:CA:9E:EE:4A:77:85
  SHA1: 05:30:69:9E:7A:AF:A9:92:CC:2F:4F:3B:0C:8A:CA:A5:F8:32:0E:AF
  SHA256: F4:59:CB:72:49:E5:DD:AF:FF:8D:35:83:08:2D:09:19:FD:F6:81:DF:A1:BC:55:30:FE:18:93:13:47:CB:7D:4E
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:
#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B5 BA C8 DA 9C 8D 20 97  2D 7C 5B 67 88 C1 E9 35  ..... --[g...5
0010: 3F AF BA A1
]
]

```

Click on create:



Create Android API key

Name

Android key 1

Restrict usage to your Android apps (Optional)

Android devices send API requests directly to Google. Google verifies that each request comes from an Android app that matches a package name and SHA-1 signing-fingerprint name that you provide. Get the package name from your AndroidManifest.xml file. Use the following command to get the fingerprint. [Learn more](#)

```
keytool -list -v -keystore mystore.keystore
```

Package name

com.iwinter.ppoint

SHA-1 certificate fingerprint

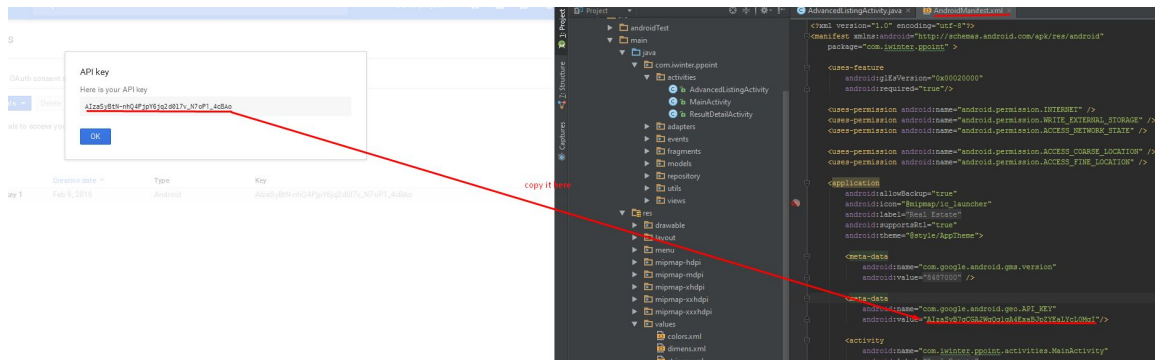
05:30:69:9F:7A:AE:A9:92:CC:2E:4E:3B:DC:BA:C8:85:F8:32:DE:AF

+ Add package name and fingerprint

Note: It may take up to 5 minutes for settings to take effect



Enter key into your AndroidManifest.xml:



Now you should wait up to few hours that key become activated before usage