# Google Map v2

# Với Eclipse

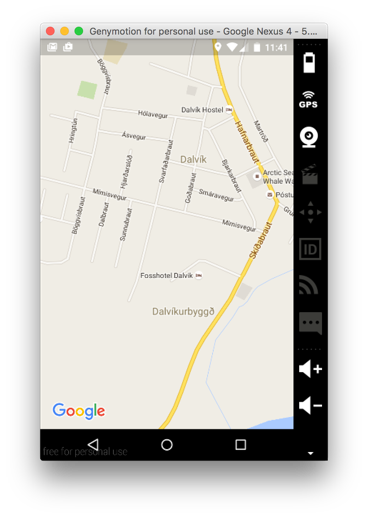
|  |
| --- |
| Cài đặt API và máy ảo và Google play Service.  1. Vào Android SDK Manager chọn một máy ảo và check vào GoogleAPI (nếu chưa check)      1. Cuộn xuống, trong phần Extras check vào Google Play services.      1. Nhấn vào nút Install , chọn accept các gói rồi install.  **Import “google-play-services\_lib”**  1. Trên Eclipse chọn File -> Import 2. Trong cửa sổ import chọn tab Android (không phải General). Chọn Existing Android Code into Workspace. Next.      1. Trong cửa sổ import Project dẫn đường dẫn đến:   android-sdk\extras\google\google\_play\_services\libproject\google-play-services\_lib. Check vào Copy projects into workspase rồi nhấn Finish.   Tạo project và nhúng thư viện  1. Tạo một project mới tên GoogleMapv2 (version từ 3.0 trở lên). 2. Click phải project vừa tạo chọn Properties. 3. Cửa sổ properties mở ra chọn mục Android.      1. Trong phần “Library” nhấn vào nút “Add”. Khi cửa sổ project Selection hiện ra chọn google-play-services\_lib rồi nhấn OK.      1. Thấy phần Reference có dấu check màu xanh. Nhấn OK.      1. Để ý thấy trong project thư mục Android Dependencies.    Lấy KeyTìm đường dẫn debug.keystore  1. Tìm đường dẫn file “debug.keystore” bằng cách chọn Window -> Preferences -> Android -> Build. Nhìn vào Default debug keystore để thấy đường dẫn đến debug.keystore.    Lấy SHA1  1. Mở command line bằng cách ngoài window chọn start -> run gõ: cmd . Enter. 2. Dùng lệnh CD để đi vào jdk thư mục như hình      1. Gõ lệnh theo cú pháp   keytool –v –list –keystore “<duongdandenfile debug.keystore>”  khi nó hỏi password thì cứ enter.    Copy lại dòng SHA1. Máy tôi là : CE:33:FE:81:C9:03:1D:C7:F3:93:2A:62:2B:67:3F:C1:9C:BC:DA:88 Lấy google api key để dùng cho chương trình.  1. Truy cập vào địa chỉ : <https://code.google.com/apis/console> 2. Dùng tài khoản Gmail để login. 3. Nhấn vào nút Create Project      1. Trong Servies cuộn xuống kiếm mục “Google Maps Android API v2 và nhấn vào OFF để chuyển sang trạng thái On.      1. Cửa sổ tiếp theo nói về các điều khoản (khỏi đọc) kéo xuống check vào “I agree to these terms” và nhấn nút Accept. 2. Nó sẽ quay về cửa sổ services và ta thấy mục đã được On lên. Mục bên trái ta chọn qua API Access.      1. Cuộn xuống và nhấn nút .      1. Cửa sổ Configure mở lên ta paste SHA1 rồi thêm ; và đến tên project vào ô nhập rồi chọn nút Create. Vd:   BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75;com.example.android.mapexample     1. Cửa sổ popup đóng lại và ta thấy đã có được API key của tôi là:   AIzaSyDmLbAdM-rgr29VWy0GD-TSddgi0vnY3rY   Cài google service packages; gms and vending  1. Tạo một máy ảo có target la googleAPIs      1. Chạy máy ảo vừa tạo. 2. Copy 2 file **com.google.android.gms-19032013.apk** và com.android.vending-19032013.apk vào thư mục “<duongdan>\android-sdk\platform-tools” (C:\Program Files (x86)\Android\android-sdk\platform-tools). 3. Start -> run -> gõ: cmd để vào command line. Dùng lệnh CD để đi đến thư mục platform-tools. 4. Gõ lệnh để cài   adb install com.google.android.gms-19032013.apk  Và  adb install com.android.vending-19032013.apk  Nếu nó báo đã tồn tại thì thôi. |

# Với Android studio & genymotion

## Cài app store và map cho genymotion

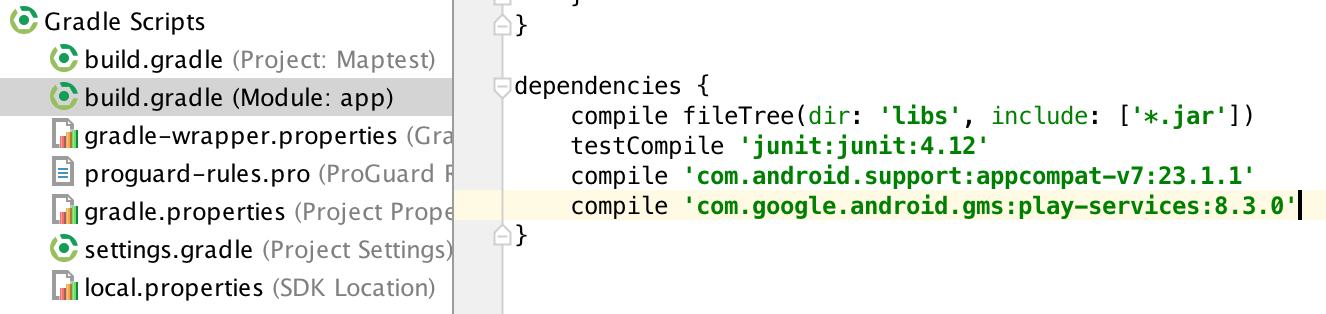
Tham khảo địa chỉ : https://github.com/codepath/android\_guides/wiki/Genymotion-2.0-Emulators-with-Google-Play-support

Cài cho được app store. Truy cập app store và download đặt google map lên genymotion, chạy map trên máy ảo genymotion ra như hình. Nếu cài không được đành chấp nhận chạy trên máy thật.



## Tạo project, cài google play service

1. Tạo một project mới.
2. Vào gradle cài google play service như hình



## Keytool và debug.keystore để lấy sha1

Trên Window keytool nằm ở thư mục java/ và debug.keystore nằm ở C:/user/<tenuser>/.android. Tham khảo cách lấy ở phần trên

Trên MacOS :

Keytool nằm ở đường dẫn /Library/Java/JavaVirtualMachines/jdk1.7.0\_45.jdk/Contents/Home/bin/keytool

Và debug.keystore nằm ở đường dẫn

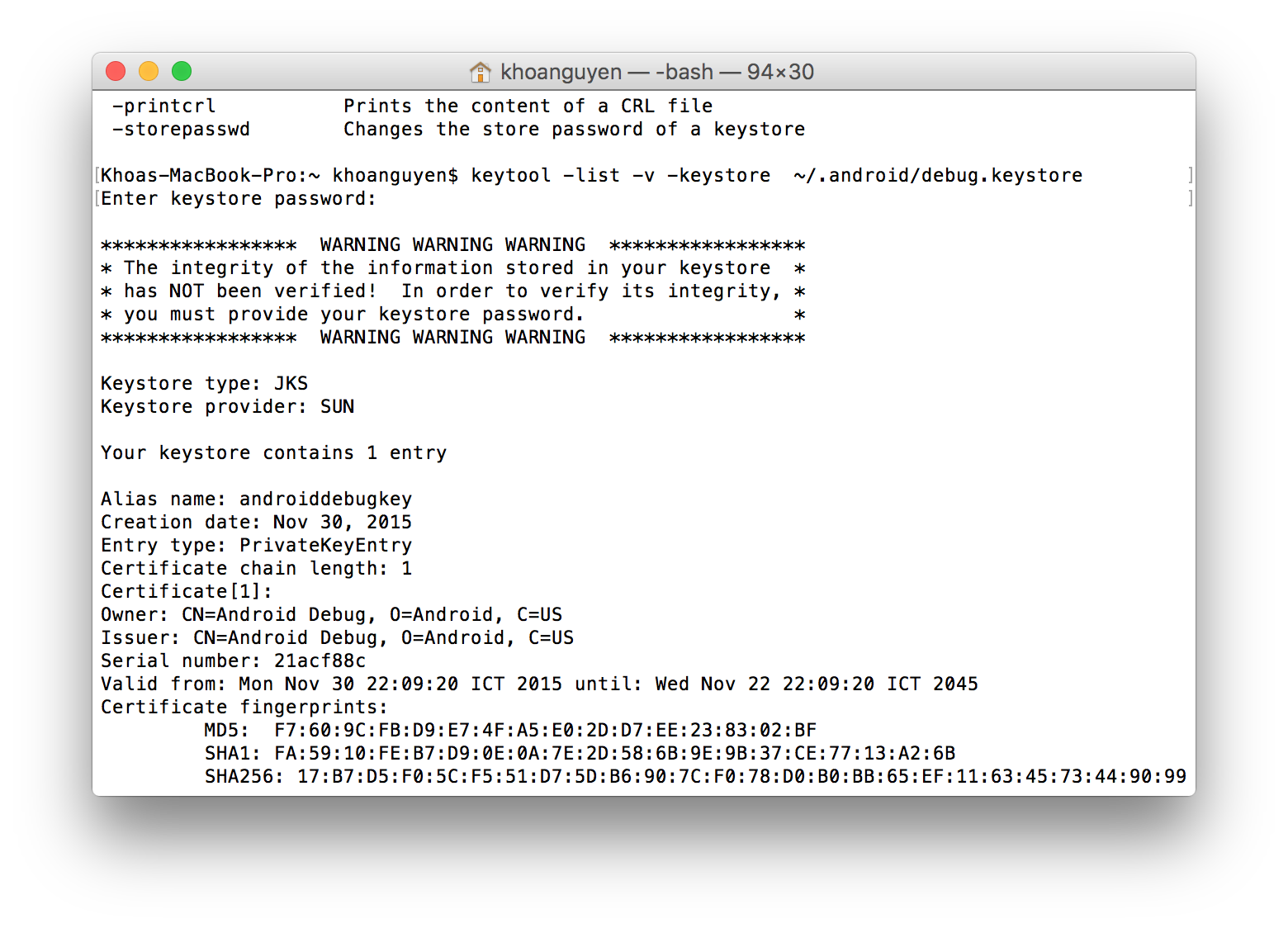
~/.android/debug.keystore

tuy nhiên ta không cần dẫn vào keytool

1. Mở terminal. Gõ vào lệnh:

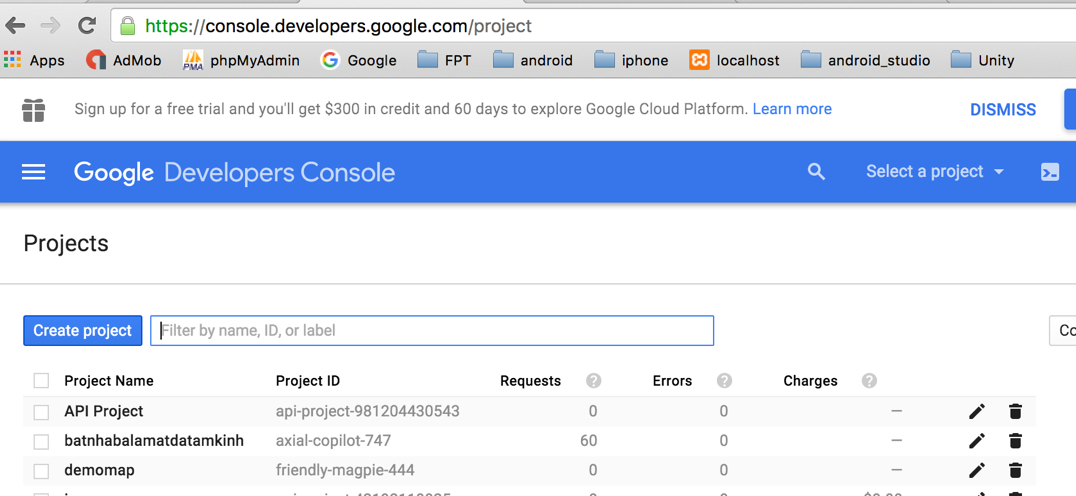
Keytool –list –v –keystore ~/.android/debug.keystore

Như hình bên dưới, nếu nó ra password thì đánh enter rồi copy dòng SHA1

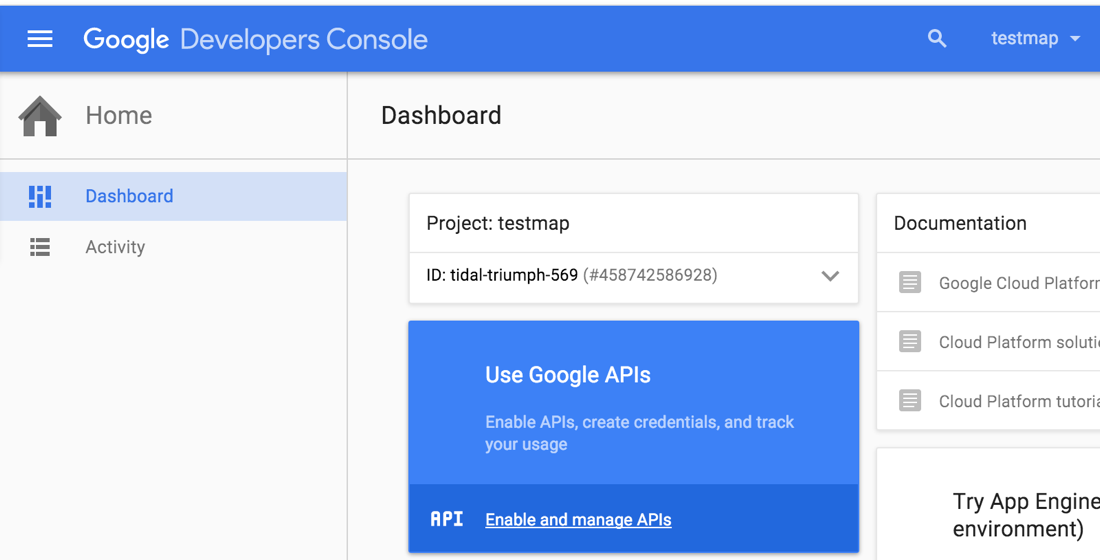


## Lấy key

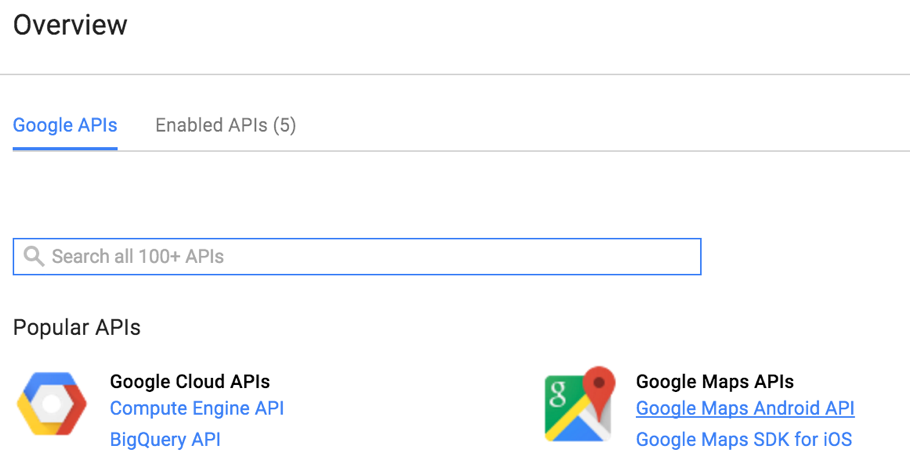
1. Truy cập vào địa chỉ : <https://code.google.com/apis/console>
2. Dùng tài khoản Gmail để login.
3. Nhấn vào nút Create Project



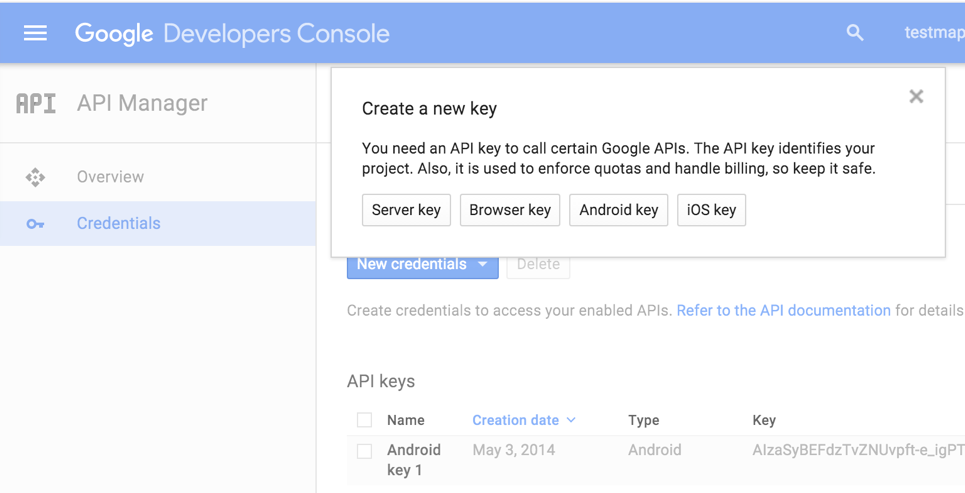
Chọn project vừa tạo, chọn Dashboard, chọn API Enable and manager APIs



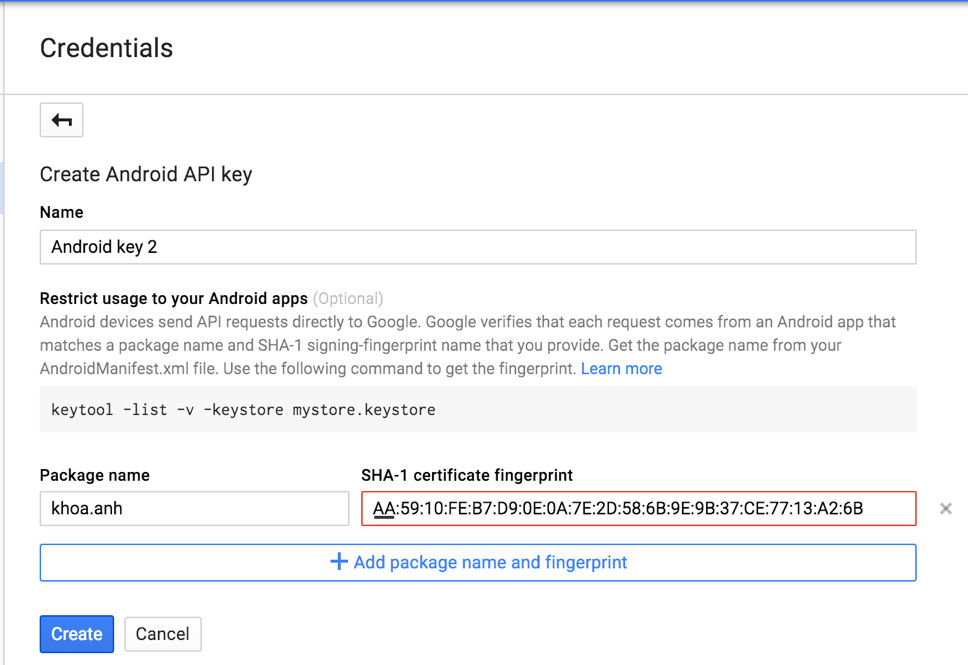
chọn đến mục Google APIs, tìm Google Maps Android API nhấn vào link rồi nhấn nút Enable



Chọn tiếp mục Credentials chọn new credentials chọn Android key



Điền sha1 và package name rồi nhấn Create



Sau khi nó tạo xong sẽ có API key copy API key để đó

# Tạo map

1. Mở tập tin AndroidManifest, dưới thẻ uses-sdk ta cấp thêm các permission như code bên dưới trong đó các quyền có ý nghĩa như sau:

INTERNET: Sử dụng bởi API để download bản đồ từ server Google Map.

ACCESS\_NETWORK\_STATE: cho phép API kiểm tra trạng thái kết nối để quyết định xem có download được dữ liệu hay không.

READ\_GSERVICES: cho phép API truy cập Google web-based services.

WRITE\_EXTERNAL\_STORAGE: Cho phép API cache dữ liệu bản đồ vào trong external storage.

ACCESS\_COARSE\_LOCATION: cho phép API dùng wifi định vị thiết bị.

ACCESS\_FINE\_LOCATION: cho phép API dùng Global Positioning System (GPS) để định vị thiết bị trong 1 phạm vi nhỏ.

<permission

android:name=*"com.example.androidmapsv2.permission.MAPS\_RECEIVE"*

android:protectionLevel=*"signature"* />

<uses-permission

android:name=*"com.example.androidmapsv2.permission.MAPS\_RECEIVE"*/>

<uses-permission

android:name=*"com.google.android.providers.gsf.permission.READ\_GSERVICES"*/>

<uses-permission

android:name=*"android.permission.INTERNET"*/>

<uses-permission

android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"*/>

<uses-permission

android:name=*"android.permission.ACCESS\_COARSE\_LOCATION"*/>

<uses-permission

android:name=*"android.permission.ACCESS\_FINE\_LOCATION"*/>

1. Dưới các thẻ permission ta viết thêm thẻ để dùng openGL như sau:

<uses-feature

android:glEsVersion=*"0x00020000"*

android:required=*"true"*/>

1. Trong thẻ Application, trên thẻ Activity ta thêm thẻ meta-data như bên dưới và thay thuộc tính android:value bằng với API key mà ta đã lấy được ở trên.

<meta-data

android:name=*"com.google.android.maps.v2.API\_KEY"*

android:value=*"AIzaSyDmLbAdM-rgr29VWy0GD-TSddgi0vnY3rY"*/>

1. Mở file layout chính thêm vào the fragment như bên dưới

<fragment

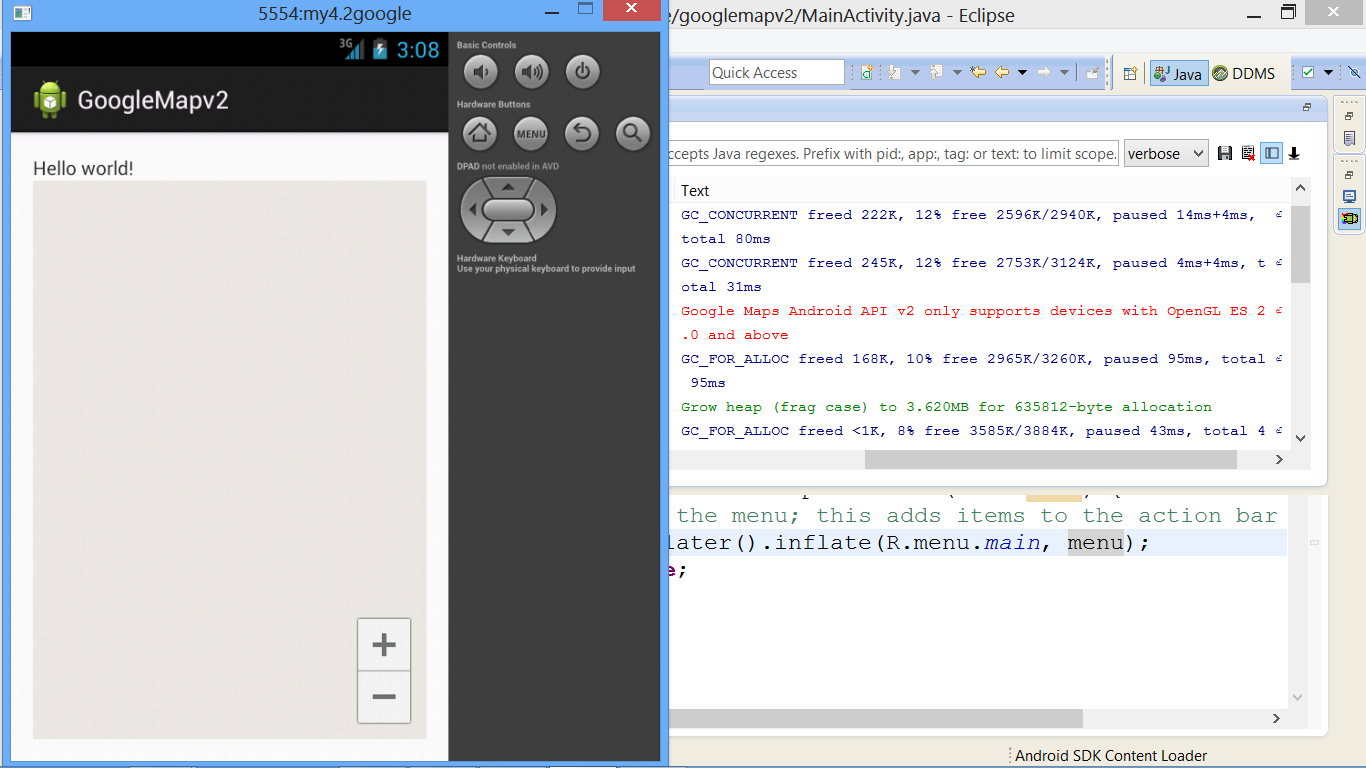
android:id=*"@+id/map"*

android:layout\_width=*"match\_parent"*

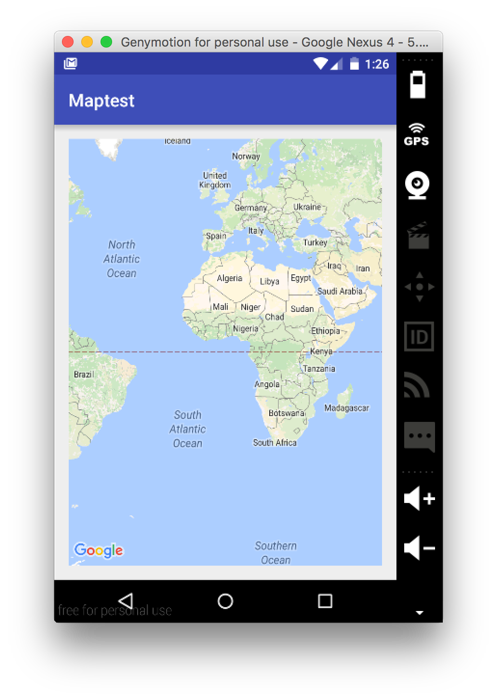
android:layout\_height=*"match\_parent"*

class=*"com.google.android.gms.maps.MapFragment"*/>

1. Chạy chương trình và xem kết quả. Chú ý nếu map không lên và xem thông báo có lỗi “Google Maps Android API v2 only supports device with OpenGL ES 2.0 and above thì yên tâm chương trình đã đúng và chạy được trên máy thật, máy ảo chạy không được là do PC không support nổi OpenGL ES2.0 thôi.



Chạy đúng sẽ lên như sau:



Chạy hoài mà không được thì tốt nhất là chạy máy thật nhé.

# Map Object

Lớp làm việc với Map có tên là GoogleMap. Bên trong UI Map được biểu diễn bởi MapFragment hoặc MapView. GoogleMap sẽ làm các thao tác tự động gồm:

* Kết nối với Google Maps service.
* Download bản đồ về.
* Hiển thị bản đồ lên.
* Hiển thị các control khác như pan và zoom.
* Pan và Zoom hỗ trợ luôn gestures.

## Tạo đối tượng map trong mã bằng cách:

1. Khai báo biến toàn cục GoogleMap tên vd: map.
2. Trong hàm onCreate ta gán cho đối tượng map bằng cách. Dùng getFragmentManager() để lấy điều khiển Fragment, gọi hàm findFragmentById để ánh xạ đến fragment chứa map. Sau đó ép toàn bộ về đối tượng MapFragment và cuối cùng gọi hàm getMap để có được đối tượng GoogleMap.

GoogleMap map;

@Override

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

map= ((MapFragment)getFragmentManager().findFragmentById(R.id.*map*)).getMap();

}

# UI Control

Map API có sẵn một số ui-control tương tự như trong chương trình google map. Ta có thể bật/tắt các control này bằng cách dùng lớp UiSettings và gọi GoogleMap.getUiSettings để lấy nó ra và tùy chỉnh. Thay đổi trên đối tượng này sẽ lập tức thực thi trên map.

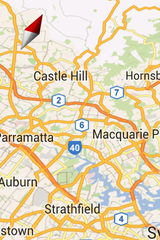
## Zoom Control

.setZoomControlsEnabled(boolean) : ẩn/hiện nút zoom-in/zoom-out



## Compass

.setCompassEnabled(boolean): ẩn/hiện la bàn trên góp trên bên trái của map. La bàn chỉ hiện khi ta dùng 2 ngón tay để quay (bearing) hoặc nghiêng (tilt) map. Khi ta click vào la bàn, thì độ quay và độ nghiêng của map sẽ về 0 (mặc định). Ta không thể cho la bàn luôn luôn hiển thị.



## Nút My Location

.setMyLocationButtonEnabled(boolean) Nút My Location xuất hiện trên góp trên bên phải của map và nó cũng chỉ hiện thị khi thuộc tính mylocation được gán là true. Khi nó được hiển thị thì khi ta click vào camera sẽ di chuyển đến vị trí hiện tại (GPS) của thiết bị đó nếu nó định vị được.



## Zoom gestures (cảm ứng cử chỉ)

Hành động zoom dựa vào những hành vi sau:

* Chạm tay 2 lần (double tap) để phóng to 1 mức (zoom-in)
* 2 ngón tay chạm 1 lần để thu nhỏ 1 mức (zoom-out)
* 2 ngón tay bóp / căng (pinch/stretch) để zoom-in/zoom-out
* 1 ngón chạm và giữ yên, 1 ngón nữa chạm và trược lên xuống để zoom-in/zoom-out

Ta có thể bật tắt các hành vi này bằng .setZoomGesturesEnabled(boolean).

## Scroll (pan) gestures

Ta có thể di chuyển camera trên map bằng cách dùng 1 ngón tay chạm vào và di chuyển. bật tắt hành vi này bằng .setScrollGesturesEnabled(boolean)

## Tilt gestures

Ta có thể chạm 2 ngón tay vào và kéo lên xuống đồng thời để nghiêng map. Bật/tắt bằng .setTiltGesturesEnabled(boolean).

## Rotate gestures

Chạm 2 ngón tay và quay để quay map. Bật/tắt bằng .setRotateGesturesEnabled(boolean).

## Demo bật/tắt các control UI

1. Tiếp theo bài trước. Trong onCreate thêm lệnh sau:

UiSettings uisetting=map.getUiSettings();

uisetting.setCompassEnabled(**true**);

uisetting.setZoomControlsEnabled(**true**);

uisetting.setScrollGesturesEnabled(**true**);

uisetting.setTiltGesturesEnabled(**true**);

uisetting.setMyLocationButtonEnabled(**true**);

map.setMyLocationEnabled(**true**);

# Map Type

Có nhiều Map type (loại bản đồ) trong Google Maps Android API. Một map type thể hiển cách hiển thị của bản đồ. VD một bản đồ thông thường hiển thị bản đồ hình vẽ với các đường biên giới giữa các quốc gia, một bản đồ dạng road map hiển thị đường đi trong 1 thành phố hay một vùng.

Thông thường ta dùng 4 loại là:

* Normal : bản đồ về đường đi. Đường xá, công trình và một số đặc điểm tự nhiên ví dụ sông ngòi v.v… được hiển thị. Tên đường cũng được hiển thị.
* Hybrid: bản đồ ảnh vệ tinh với đường xá được thêm vào, thông tin của đường xá cũng được thêm.
* Satellite: bản đồ ảnh vệ tinh không có đường xá và thông tin đường xá.
* Terrain: bản đồ địa hình. Bao gồm màu sắc thể hiện cho địa hình, một số con đường và thông tin cũng được thêm vào.
* None: nạp lên như 1 lưới rỗng không có gì cả.

Muốn thay đổi map tye ta dùng hàm setMapType() và truyền vào map type.

## Thay đổi map type

1. Mở file “menu/main.xml” và thêm các item như sau:

<menu xmlns:android=*"http://schemas.android.com/apk/res/android"* >

<item

android:id=*"@+id/action\_settings"*

android:orderInCategory=*"100"*

android:showAsAction=*"never"*

android:title=*"@string/action\_settings"*/>

<item

android:id=*"@+id/normal"*

android:title=*"normal"* />

<item

android:id=*"@+id/hybrid"*

android:title=*"hybrid"* />

<item

android:id=*"@+id/satellite"*

android:title=*"satellite"* />

<item

android:id=*"@+id/terrain"*

android:title=*"terrain"* />

</menu>

1. Mở file java chính và override lên phương thức onOptionsItemSelected và viết như sau:

@Override

**public** **boolean** onOptionsItemSelected(MenuItem item) {

// **TODO** Auto-generated method stub

**switch** (item.getItemId()) {

**case** R.id.*normal*:

map.setMapType(GoogleMap.*MAP\_TYPE\_NONE*);

**break**;

**case** R.id.*hybrid*:

map.setMapType(GoogleMap.*MAP\_TYPE\_HYBRID*);

**break**;

**case** R.id.*satellite*:

map.setMapType(GoogleMap.*MAP\_TYPE\_SATELLITE*);

**break**;

**case** R.id.*terrain*:

map.setMapType(GoogleMap.*MAP\_TYPE\_TERRAIN*);

**break**;

**default**:

**break**;

}

**return** **super**.onOptionsItemSelected(item);

}