Tutorial de uso de mapas Google para Android

Para poder realizar el presente tutorial, lo primero que requerimos hacer es obtener una clave de Google que nos dará acceso a poder usar los tutoriales Google.

1.- Obtener una clave de Google

En Windows lo primero que hay que hacer es ir al menú ejecutar y escribir *Cmd*, de este modo entraremos a la linea de comando. Dentro de la linea de comando hay que ir a la carpeta que contiene java, esta tendrá una dirección similar a esta: C:\Program Files\Java\jdk1.6.0_14\bin

En un explorador de Windows buscar el archivo debug.keystore ubicado en la carpeta Android en un path similar a este : C:\Documents and Settings\<user>\.android\debug.keystore

Ya en la carpeta /bin ejecutar el siguiente código:
keytool -list -alias androiddebugkey -keystore <path_to_debug_keystore>.keystore -storepass android -keypass android

Cambiando <path_to_debug_keystore>.keystore por la dirección de archivo debug.keystore que buscamos anteroirmente

Para Linux o MacOS: buscar el archivo debug.keystore para eso entrar a terminal y escribir:

locate debug.keystore

el resultado será algo así:
/home/soltux/.android/debug.keystore

escribir en la terminal
keytool -list -alias androiddebugkey -keystore <path_to_debug_keystore>.keystore -storepass android -keypass android

remplazando <path_to_debug_keystore>.keystore por el resultado de la búsqueda, en mi caso será algo asi:
keytool -list -alias androiddebugkey -keystore /home/soltux/.android/debug.keystore -storepass android -keypass android

Posteriormente entraremos a: https://developers.google.com/android/maps-api-signup?hl=es-ES
Donde dice *My certificate's MD5 fingerprint*: copiar el fingerprint que obtuvieron:

en su consola se debe ver algo así:
Certificate fingerprint (MD5): 94:1E:43:49:87:73:BB:E6:A6:88:D7:20:F1:8E:B5:98
Deben copiar 94:1E:43:49:87:73:BB:E6:A6:88:D7:20:F1:8E:B5:98

Acepten las condiciones de servicio y hagan click en Generate API KEY

El resultado será una clave como estas: 0F8CaQ6i_HEJ4F38zvKj8jf5am5bWtf7Bl_IYbQ
La misma que deben guardar en un lugar seguro para no tener que hacer este proceso nuevamente, cada vez que hagan un proyecto con Google maps. (nota: esta clave solo sirve para la instalación actual del androidsdk, si tienen otras instalaciones del Android SDK en otros equipos estos igual requerirán su propia clave para poder acceder al servicio).

2.- El proyecto en eclipse

Paso 1: crear un proyecto llamado ejemploMapaGoogle o algo similar.

Paso 2: seleccionar la versión de android para el proyecto, en este caso deberán usar un google api.

Paso 3: crear el proyecto

Paso 4: Entrar al Android manifest file del proyecto.

Paso 5: agregar la librería de mapas de Google

```
<uses-library android:name="com.google.android.maps" />
```

Esto nos permitirá utilizar las funciones de mapas google en nuestro proyecto... el Android Manifest File se verá similar a esto:

```
<activity android:name="RecordarPassActivity"></activity>
<uses-library android:name="com.google.android.maps" />

   </application>
```

Paso 6: agregar permisos de uso de internet a nuestra aplicación:

```
<uses-permission android:name="android.permission.INTERNET" />
```

El Android Manifest File debería quedar algo parecido a esto:

```
<uses-permission android:name="android.permission.INTERNET" />
<application
      android:icon="@drawable/ic_launcher"
      android:label="@string/app_name" android:name="MyApplication">
      <activity
…
```

Paso 7: si queremos que no aparezca el titulo de la aplicación para tener mas espacio para el mapa, podemos añadir la propiedad *notitlebar* a nuestra actividad, esto se hace dentro del Android Manifest File agregando:
android:theme="@android:style/Theme.NoTitleBar" >

Dentro de las propiedades de la actividad:

Paso 8: abrir el archivo main.xml .

Paso 9: Cambiar el contenido de main.xml por este:

```
<?xml version="1.0" encoding="utf-8"?>
<com.google.android.maps.MapView
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:id="@+id/VistaMapa"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:apiKey="0F8CaQ6i_HEIkubtfJbfIdFTgn1tc65_mnG1i0A"
/>
```

esto nos permitirá definir una vista de mapa de Google, y añadirle propiedades a nuestro mapa. Nota: no olviden cambiar el código del apiKey por el que ustedes generaron anteriormente.

Paso 10: ir a ejemploMapaGoogle.java o el archivo que creamos con nuestro proyecto.

Paso 11: cambiar el extends Activity por extends MapActivity, luego presionar si están en eclipse ctrl+shift+o eso permitirá al programa auto importar las librerías necesarias , en caso de que no usen eclipse deberán importar
import com.google.android.maps.MapActivity;

Esto nos permitirá usar algunos métodos diseñados exclusivamente para mapas.

Paso 12: como podrán apreciar Eclipse aun debe marcar un pequeño error en la linea ejemploMapaGoogle extends MapActivity.

Esto debido a que los mapas de Google requieren que obligatoriamente definamos si es que estamos usando información de rutas o no, en el caso de este ejemplo no vamos a dibujar ninguna ruta, por lo que lo definiremos como falso. Entonces debemos agregar lo siguiente:

```
@Override
        protected boolean isRouteDisplayed() {
                // TODO Auto-generated method stub
                return false;
```

```
        }
```

Paso 13: agregar el método onCreate() como de costumbre

```
public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
}
```

Paso 14: definir que la vista será main.xml

```
setContentView(R.layout.main);
```

Paso 15: añadiremos controles de zoom para que puedan interactuar con el mapa:

```
MapView mapView = (MapView) findViewById(R.id.VistaMapa);
mapView.setBuiltInZoomControls(true);
```

El método setBuiltInZoomControls permitirá que podamos utilizar los botones de zoom que aparecen en todos los mapas.

Paso 16: para que el usuario pueda interactuar con el mapa y usar los controles correctamente debemos añadirle la propiedad clickable, que permitirá que se pueda interactuar con el mapa, para eso agregaremos:

```
android:clickable="true"
```

en nuestro archivo main.xml como propiedad del mapa.

Versión 2012.05.05