

Servicios Basados en Localización (LBS)

Tema 6

Desarrollo de aplicaciones en Android

Ramón Alcarria
Miguel Ángel Manso



1

API de Localización

- Existen dos APIs de localización en Android
 - Google Location Services for Android
 - No lo vamos a ver en clase
 - android.location

Google Location Services

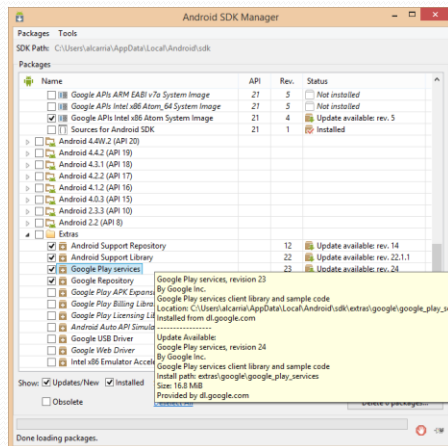
- Google Location Services for Android
 - Un servicio de localización más potente, disponible como parte de Google Play Services
 - Tenemos que incluir estos servicios en nuestra app.
- Pasos
 - Asegurarnos que nos hemos descargado el componente de Google Play Services a través del SDL Manager
 - Agregar la librería a nuestro proyecto

LBS 2014-2015

3

Google Location Services

- Descarga de Google Play Services

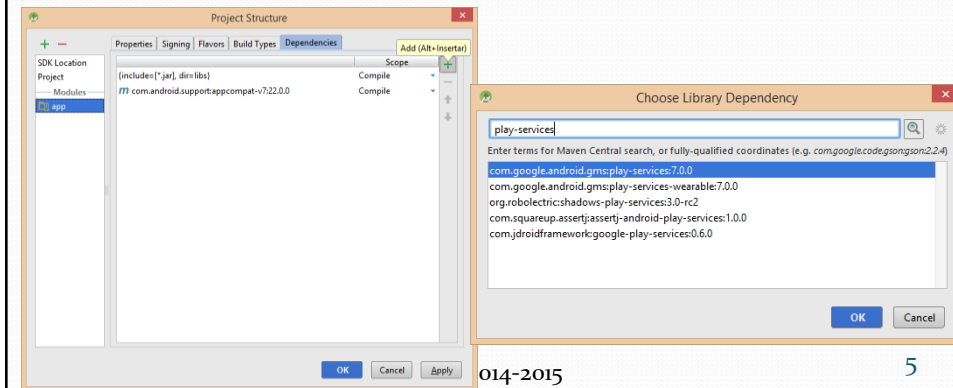


LBS 2014-2015

4

Google Location Services

- Añadir librería a nuestro proyecto
 - Click derecho sobre nuestro proyecto app: *Open Module Settings*
 - En *Dependencies* añadimos nueva *Library Dependency*



014-2015

5

Google Location Services

- Comprobaremos como se ha añadido la dependencia en *build.gradle (Module: app)*

```
dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')
    compile 'com.android.support:appcompat-v7:22.0.0'
    compile 'com.google.android.gms:play-services:7.0.0'
}
```

LBS 2014-2015

6

Google Location Services

- Utilizando los servicios de localización
 - Necesitamos especificar permisos en el manifiesto
 - ACCESS_COARSE_LOCATION
 - ACCESS_FINE_LOCATION -> Más precisa

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.google.android.gms.location.sample.basiclocationsample" >

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

Google Location Services

- Utilizando los servicios de localización
 - Necesitamos una instancia de GoogleApiClient
 - En el onCreate llamamos al método *buildGoogleApiClient()*

```
protected synchronized void buildGoogleApiClient() {
    mGoogleApiClient = new GoogleApiClient.Builder(this)
        .addConnectionCallbacks(this)
        .addOnConnectionFailedListener(this)
        .addApi(LocationServices.API)
        .build();
}
```

- Por supuesto habrá que definir el objeto *mGoogleApiClient* (lo hacemos como variable de clase)

Google Location Services

- Obtener la última localización conocida
 - En nuestra actividad implementamos las interfaces: ConnectionCallbacks, OnConnectionFailedListener
 - Añadimos los métodos que faltan

android.location

- Utilizando los servicios de localización
 - Necesitamos especificar permisos en el manifiesto
 - ACCESS_COARSE_LOCATION
 - ACCESS_FINE_LOCATION -> Más precisa

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.google.android.gms.location.sample.basiclocationsample" >

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

android.location

- LocationManager, permite:
 - Obtener la última localización conocida
 - Obtener la lista de LocationProviders
 - Registrarnos/desregistrarnos a actualización de la posición de un usuario dada por un LocationProvider
 - Disparar Intents si el dispositivo se aproxima a un punto especificado.

```
// Referencia a Location Manager
    LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);
```

android.location

- Última localización conocida

```
// Referencia a Location Manager
    LocationManager locationManager = (LocationManager)
this.getSystemService(Context.LOCATION_SERVICE);

    Location loc =
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

    String lat = String.valueOf(loc.getLatitude());
    String lon = String.valueOf(loc.getLongitude());
    Toast.makeText(this, "Latitud: "+lat+" Longitud: "+lon,
Toast.LENGTH_LONG).show();
```

android.location

- Obtener el mejor proveedor de localización

```
Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE);
```

int	ACCURACY_COARSE	A constant indicating an approximate accuracy requirement
int	ACCURACY_FINE	A constant indicating a finer location accuracy requirement
int	ACCURACY_HIGH	A constant indicating a high accuracy requirement - may be used for horizontal, altitude, speed or bearing accuracy.
int	ACCURACY_LOW	A constant indicating a low location accuracy requirement - may be used for horizontal, altitude, speed or bearing accuracy.
int	ACCURACY_MEDIUM	A constant indicating a medium accuracy requirement - currently used only for horizontal accuracy.
int	NO_REQUIREMENT	A constant indicating that the application does not choose to place requirement on a particular feature.
int	POWER_HIGH	A constant indicating a high power requirement.
int	POWER_LOW	A constant indicating a low power requirement.
int	POWER_MEDIUM	A constant indicating a medium power requirement.

```
String provider =
LocationManager.getBestProvider(criteria,
true);
```

Provider:

```
LocationManager.NETWORK_PROVIDER
LocationManager.PASSIVE_PROVIDER
LocationManager.GPS_PROVIDER
```

LBS 2014-2015

13

android.location

- Actualizaciones de localización
 - Definimos el escuchador de actualizaciones (*listener*)

```
// Define a listener that responds to location updates
LocationListener locationListener = new LocationListener() {
    public void onLocationChanged(Location location) {
        // Called when a new location is found by the network location provider.
        manejadorLocalizacion(location);
    }

    public void onStatusChanged(String provider, int status, Bundle extras) {}

    public void onProviderEnabled(String provider) {}

    public void onProviderDisabled(String provider) {}
};
```

LBS 2014-2015

14

android.location

- Actualizaciones de localización
 - Registramos el *listener*

```
locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
0, 0, locationListener);
```

- Creamos nuestro manejador de la localización

```
private manejadorLocalizacion(Location location);
```

Más información en

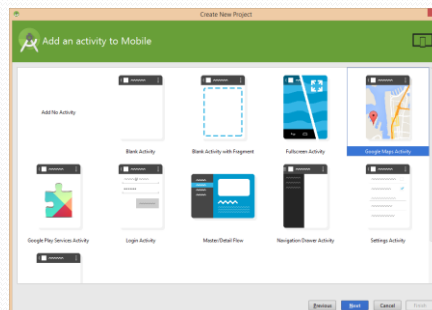
<https://developer.android.com/guide/topics/location/strategies.html>

LBS 2014-2015

15

Google Maps

- Es parte de Google Play Services y tenemos que instalar este servicio como en el caso de Google Location Services
- Creamos un proyecto de Google Maps, según aparece en el Android Studio



16

Google Maps

- Hay que obtener un API_Key

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="es.upm.gee-ny.googlemaps" >

    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="com.google.android.providers.gsf.permission.READ_GSERVICES" />
    <!--
    The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
    Google Maps Android API v2, but are recommended.
    -->
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <meta-data
            android:name="com.google.android.gms.version"
            android:value="@integer/google_play_services_version" />
        <meta-data
            android:name="com.google.android.maps.v2.API_KEY"
            android:value="@string/google_maps_key" />

        <activity
            android:name=".MapsActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

```

<resources>
    <!--
    TODO: Before you run your application, you need a Google Maps API key.
    To get one, follow this link, follow the directions and press "Create" at the end:
    https://console.developers.google.com/floves/enableapi?enableapi=maps_android_backend&keyType=CLIENT
    You can also add your credentials to an existing key, using this line:
    BF:97:07:21:77:DF:A4:9A:0E:43:7B:55:92:03:8A:C4:1B:96:E2:81:res.upm.gee-ny.googlemaps
    Once you have your key (it starts with "AIza"), replace the "google_maps_key"
    string in this file.
    -->
    <string name="google_maps_key" translatable="false" templateMergeStrategy="preserve">
        YOUR_KEY_HERE
    </string>
</resources>

```

2015

17

Google Maps

- Obtención de API_Key
 - Recuperar información de nuestro certificado de aplicación
 - Registrar un proyecto en la consola de desarrolladores de Google.
 - Añadir el API de Mapas como servicio para ese proyecto.
 - Pedir una o más claves
 - Añadir la clave a la aplicación

LBS 2014-2015

18

Google Maps: API_Key

- Certificado de aplicación
 - Vamos a C:\Users\\.android
 - Comprobamos que existe el archive debug.keystore
 - Abrimos una consola del sistema e introducimos:

```
keytool -list -v -keystore "%USERPROFILE%\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
```

- Apuntamos la secuencia SHA1

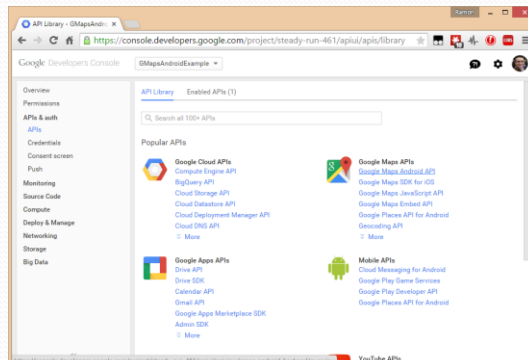
```

Tipo de entrada: PrivateKeyEntry
Longitud de la cadena de Certificado: 1
Certificado[1]:
Propietario: CN=Android Debug, O=Android, C=US
Emisor: CN=Android Debug, O=Android, C=US
N.mero de serie: 6f8c6f14
Vlido desde: Sun Mar 22 16:51:50 CET 2015 hasta: Tue Mar 14 16:51:50 CET 2045
Huellas digitales del Certificado:
MD5: 97:5E:44:02:4A:CB:23:D1:CB:A3:AB:6C:EA:60:14:6A
SHA1: BF:97:07:21:77:DF:A4:9A:0E:43:7B:55:92:03:5A:C4:1B:96:E2:E1:08:E0:D7:F8:A2:4F:8B:E4:E7:FA:D1
SHA256: 6C:03:A4:D6:12:74:6F:57:49:EC:81:EB:32:3A:2B:62:85:A4:65:73:D1:08:E0:D7:F8:A2:4F:8B:E4:E7:FA:D1
Nombre del Algoritmo de Firma: SHA256withRSA
Versi:n: 3
Extensiones:
  
```

19

Google Maps: API_Key

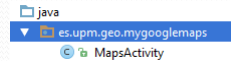
- Registro de proyecto
 - Vamos a <https://console.developers.google.com/>
 - Creamos un Nuevo proyecto
 - En APIs activamos Google Maps Android API



20

Google Maps: API_Key

- Pedir la clave
 - En Credentials pulsa “Create new Key” -> “Android Key”
 - En el cuadro de texto introduce tu clave SHA-1 seguida de “;” y de el nombre del paquete de tu aplicación



- Una vez hecho esto copiamos la clave que se ha generado en: *Key for Android applications*

Google Maps: API_Key

- Añadir la clave a la aplicación
 - En google_maps_api.xml copia la clave generada

```

MapsActivity.java x  AndroidManifest.xml x  google_maps_api.xml x
<resources>
  <!--
  TODO: Before you run your application, you need a Google Maps API key.

  To get one, follow this link, follow the directions and press "Create" at the end:
  https://console.developers.google.com/flows/enableapi?enableid=maps_android_backend&keyType=CLIENT

  You can also add your credentials to an existing key, using this line:
  BF:97:07:21:77:DF:A4:9A:0E:43:7B:55:92:03:5A:C4:1B:96:E2:E1:es.upm.geo.mygooglemaps

  Once you have your key (it starts with "AIza"), replace the "google_maps_key"
  string in this file.
  -->
  <string name="google_maps_key" translatable="false" templateMergeStrategy="preserve">
    YOUR_KEY_HERE
  </string>
</resources>
  
```

Google Maps: Desarrollo

- Necesitamos permisos en el manifiesto

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

- Los dispositivos tienen que ser compatibles con OpenGL2, así que añadimos en el manifiesto

```
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true"/>
```

LBS 2014-2015

23

Google Maps: Desarrollo

- Nuestro proyecto nos define
 - GoogleMap mMap
- Dentro del método setUpMap() podemos:
 - Cambiar el tipo de mapa

int	MAP_TYPE_HYBRID	Satellite maps with a transparent layer of major streets.
int	MAP_TYPE_NONE	No base map tiles.
int	MAP_TYPE_NORMAL	Basic maps.
int	MAP_TYPE_SATELLITE	Satellite maps with no labels.
int	MAP_TYPE_TERRAIN	Terrain maps.

- Habilitar Google Street View
 - <https://developers.google.com/maps/documentation/android/streetview>

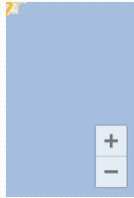
LBS 2014-2015

24

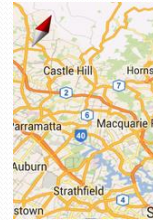
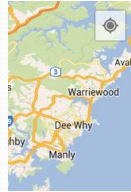
Google Maps: Desarrollo

- Añadir controles

```
UiSettings uiSettings = mMap.getUiSettings();
```



```
uiSettings.setMyLocationButtonEnabled(true);
```



```
uiSettings.setCompassEnabled(true);
```

```
uiSettings.setZoomControlsEnabled(true);
```

LBS 2014-2015

25

Google Maps: Desarrollo

- Añadir controles

```
UiSettings uiSettings = mMap.getUiSettings();
```



```
uiSettings.setMapToolbarEnabled(true)
```



```
uiSettings.setIndoorLevelPickerEnabled(true);
```

LBS 2014-2015

26

Google Maps: Desarrollo

- Manejar pulsaciones en pantalla

```
mMap.setOnMapClickListener(new GoogleMap.OnMapClickListener (){
    public void onMapClick(LatLng point){
        Toast.makeText(MapsActivity.this, "Clicked: " + point,
            Toast.LENGTH_SHORT).show();
    }
});
```

- Pulsaciones largas

```
mMap.setOnMapLongClickListener(new GoogleMap.OnMapLongClickListener (){
    public void onMapLongClick(LatLng point){
        Toast.makeText(MapsActivity.this, "Long clicked: " + point,
            Toast.LENGTH_SHORT).show();
    }
});
```

LBS 2014-2015

27

Google Maps: Desarrollo

- Cambiar el nivel de Zoom

```
map.animateCamera(CameraUpdateFactory.zoomIn());
map.animateCamera(CameraUpdateFactory.zoomOut());
map.animateCamera(CameraUpdateFactory.zoomBy(-2));
map.animateCamera(CameraUpdateFactory.zoomTo(10));
```

LBS 2014-2015

28

Google Maps: Desarrollo

- Añadir marcadores

- Normal

```
map.addMarker(new MarkerOptions()
    .position(new LatLng(10, 10))
    .title("Hello world"));
```

- Movable

```
map.addMarker(new MarkerOptions()
    .position(new LatLng(15, 15))
    .title("Marcador2")
    .draggable(true));
```

- Podemos cambiar el icono, hacerlo invisible, rotarlos, etc.:

- <https://developer.android.com/reference/com/google/android/gms/maps/model/MarkerOptions.html>

Google Maps: Desarrollo

- Añadir descripción a los marcadores

```
Marker melbourne = mMap.addMarker(new MarkerOptions()
    .position(new LatLng(-37.81319, 144.96298) )
    .title("Melbourne")
    .snippet("Population: 4,137,400"));
```

Google Maps: Desarrollo

- Polilíneas

```
//Creamos una polilínea a través de una Clase de opciones. Especificamos los puntos de la polilínea
PolylineOptions rectOptions = new PolylineOptions()
    .add(new LatLng(37.35, -122.0))
    .add(new LatLng(37.45, -122.0)) // Al norte del punto anterior
    .add(new LatLng(37.45, -122.2)) // Al oeste del punto anterior
    .add(new LatLng(37.35, -122.2)) // Al sur del punto anterior
    .add(new LatLng(37.35, -122.0)); // Cerramos la polilínea.

// Creamos la polilínea y le añadimos las opciones anteriores
Polyline polyline = myMap.addPolyline(rectOptions);
```

- Polígonos

```
// Creamos un polígono a través de una Clase de Opciones. Especificamos los puntos del polígono
PolygonOptions rectOptions = new PolygonOptions()
    .add(new LatLng(37.35, -122.0), //Mismos puntos que en el ejemplo anterior
        new LatLng(37.45, -122.0),
        new LatLng(37.45, -122.2),
        new LatLng(37.35, -122.2),
        new LatLng(37.35, -122.0));

// Creamos el polígono y le añadimos las opciones anteriores
Polygon polygon = myMap.addPolygon(rectOptions);
```