UNIVERSITÀ
DEGLI STUDI
DE L'AQUILA

disim

Mobile Programming Laboratory

ANDROID
Google Maps

Teachers

Ing. Tarquini Francesco, Ph.D

Ph.D in Computer Science Engineering

francesco.tarquini@univaq.it

Ing. D'Errico Leonardo

Ph.D Student in Computer Science Engineering

leonardo.derrico@graduate.univaq.it

Teaching Materials

Available on MOODLE platform

http://www.didattica.univaq.it

Google Drive Repository

https://drive.google.com/drive/folders/1ISqZfn0i9Ub3eWNXbvW00rd0hD9ya8OL?usp=sharing

Topics

- Google Maps
  - API Key
  - Maps
  - Markers

Google Maps

Allow your users to explore the world with rich maps provided by Google.

The Google Maps Android API allows you to include maps and customized mapping information in your app.

Since Google Maps Android API v2, you can embed maps into an activity as a fragment with a simple XML snippet.

Google Maps - API Key

To obtain SHA-1 of you app, you can invoke Java keytool command on the debug.keystore or release.keystore to publish the app.

MacOS and Linux: ~/.android/

Windows: C:\Users\<YOUR_NAME>\.android\

Open terminal and run

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

The default password is **android**

Google Maps - API Key

Google allows you to use the Google Maps by using an API Key linked to your developer account.

The first step is get an API key from Google API Console
https://console.developers.google.com/flows/enableapi?apiid=maps_android_backend&reusekey=true

Create or select a project
Click Continue to enable the Google Maps Android API
On Credentials page, get an API Key with Android restriction
In the Restriction section, select Android apps, then enter your app's SHA-1 fingerprint and package name.

SHA-1 Example: BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75
PACKAGE NAME Example: com.example.android.mapexample

Click save

The API Key is something like this

AIzaSyBdVl-cTICSwYKrZ95SuvNw7dbMuDt1KG0

Google Maps - Maps

Now you can add maps to your Activity.

Include maps dependency in build.gradle

```
implementation 'com.google.android.gms:play-services-maps:15.0.1'
```

Create the XML layout file of Activity and add the map fragment

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:id="@+id/map"
  tools:context=".MapsActivity"
  android:name="com.google.android.gms.maps.SupportMapFragment" />
```

Google Maps - Maps

Create the Activity class that implements OnMapReadyCallback interface.

Get the MapFragment from XML and init Google maps

```
SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
      .findFragmentById(R.id.map);
mapFragment.getMapAsync(this);
```

Inside onMapReady we can configure the Map

```
@Override
public void onMapReady(GoogleMap googleMap) {
   // My Configuration
}
```

Google Maps - Maps

Add API Key to manifest to allow your app to utilize the Maps.

You must include API Key as a Meta-Data as child of application element.

```
<application …>

  <meta-data
     android:name="com.google.android.maps.v2.API_KEY"
     android:value="YOUR_API_KEY"/>

  …

</application>
```

NOTE: it is a best practice use string resource by calling your API Key

Google Maps - Maps

Add a Marker in a location is very easy.

```
LatLng laquila = new LatLng(42.367422, 13.349200);

googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(laquila, 12f));

MarkerOptions markerOptions = new MarkerOptions();
markerOptions.position(laquila);
markerOptions.title("City of L'Aquila");
markerOptions.snippet("Snippet of L'Aquila");
markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BLUE));
Marker marker = googleMap.addMarker(markerOptions);
```

To remove the marker from map

```
marker.remove();
```