



This article explains how to use [Google Maps](#) data in a mobile application.

Google Maps offers [REST](#) services that allow accessing its data with simple HTTP requests, so they can be easily integrated into mobile applications.

Contents

- [1 Sign up for a Google Maps API key](#)
- [2 Static maps](#)
- [3 Geocode an address](#)
- [4 Proxy server, usage limits](#)
- [5 Sample application](#)
- [6 Related content](#)

Sign up for a Google Maps API key

NOTE: Usage of this code with the free Google Maps API Key breaks Google's [Terms and Conditions](#) (section 10.8). You should purchase an Enterprise License if you wish to use the Google Maps API as shown in this example.

First you need to sign up on this page:

<http://code.google.com/apis/maps/signup.html>

Once you have signed up, you get a key (a simple string) that you can use for all your queries to Google Maps services.

Static maps

Standard Google Maps code is suitable for Web applications. However, it includes a lot of Ajax functionalities that are not really useful if you are building a mobile application. The solution is to use the [static maps](#) service that allows retrieving single images that can easily be used in mobile applications.

The static maps service supports different image formats (png32, GIF, JPG) and customizable image size, so you can get perfect images for all purposes. For example, if you want to retrieve the location at:

- latitude: 41.867878
- longitude: 12.471516

You can simply retrieve this URL with an HTTP GET request:

`http://maps.google.com/staticmap?center=41.867878,12.471516&format=png32&zoom=8&size=240x320&key=`

This way you will get a PNG32 image with a width of 240 pixels and a height of 320 pixels, centered at point (41.867878,12.471516), and with a zoom level of 8 (the zoom range is from 0 to a maximum level of 19)



Geocode an address

From Google Maps docs:

Geocoding is the process of converting addresses (such as "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739)

The following example describes building an application that displays the address typed by the end user. First you need to geocode its address into geographic coordinates.

To do this, Google Maps offers another REST service that can easily be accessed with simple HTTP requests.

If you want to geocode this address

Leicester Square, London

Request this URL from your code

```
http://maps.google.com/maps/geo?q=Leicester%20Square,%20London&output=csv&key=<API_KEY>
```

and you will get this output:

```
200,6,51.510605,-0.130728
```

Where:

- The first number is a code, which in this case (200) means that geocoding has been successful (for a full list of status codes see: [\[1\]](#))
- The second number gives a measure of the geocoding accuracy (from 0 to 9 - maximum accuracy)
- The 3rd and 4th numbers represent latitude and longitude of the geocoded address, so these are the coordinates used to retrieve the map through the static map service.

As you can see, there is an 'output' parameter in the geocode request. This means that you can choose the output format you prefer. The supported formats are:

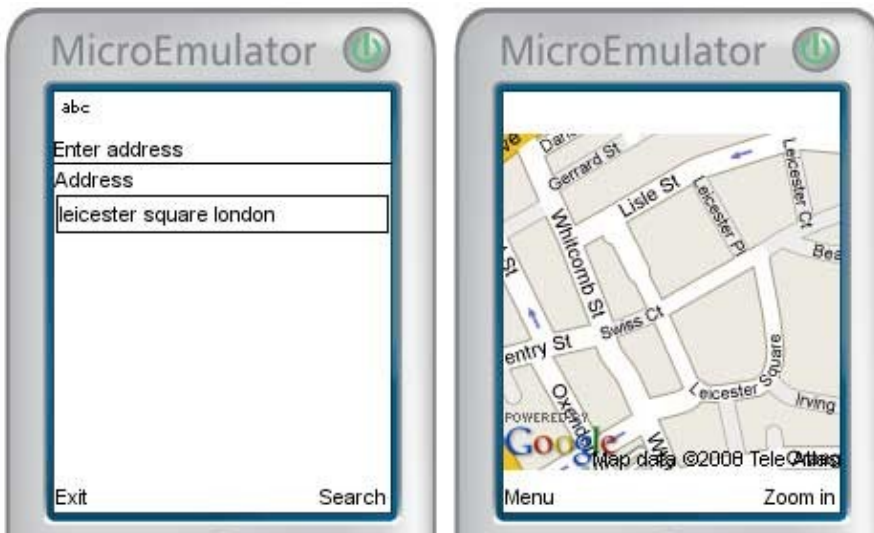
- xml
- kml (same as xml, but with different Content-Type)
- json (not really useful for mobile apps)
- csv (comma-separated values)

Proxy server, usage limits

Since your Google Maps API key is bound to a specific URL, in order to access map services **you need to setup a proxy server** that will receive HTTP requests from the mobile application and forward them to Google Maps REST URLs, returning Google responses to mobile clients. *(as pointed out in the Comment page, this is not a fully clear point yet)*

Also, be aware that there is a **limit to the number of requests**, both for static maps and geocode service, you can do each day. For personal uses they are more than enough, but you need to keep this issue in mind if you plan to develop commercial services.

Sample application



A sample J2ME application, using the approach described here, is available on this page: [Google Maps J2ME Test](#)

Google Maps J2ME API source code used in this example is also available here: [J2ME Google Maps API](#)

Related content

- [J2ME Google Maps API](#)
- [Location API](#)
- [How to get Location Using Location API JSR 179](#)
- [Java ME Location API](#)