

# GOOGLE MAPS HACKS

*Tips and Tools for Geographic  
Searching and Remixing*



O'REILLY®

*Rich Gibson & Schuyler Erle*

# Google Maps Hacks™

by Rich Gibson and Schuyler Erle

Copyright © 2006 O'Reilly Media, Inc. All rights reserved.  
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North,  
Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*safari.oreilly.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

**Editor:** Simon St. Laurent  
**Production Editor:** Jamie Peppard  
**Copyeditor:** Derek Di Matteo  
**Indexer:** Nancy Crompton

**Cover Designer:** Marcia Friedman  
**Interior Designer:** David Futato  
**Cover Illustrator:** Karen Montgomery  
**Illustrators:** Robert Romano, Jessamyn  
Read, and Lesley Borash

## Printing History:

January 2006: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. The *Hacks* series designations, *Google Maps Hacks*, the image of an antique globe, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

*Small print:* The technologies discussed in this publication, the limitations on these technologies that technology and content owners seek to impose, and the laws actually limiting the use of these technologies are constantly changing. Thus, some of the hacks described in this publication may not work, may cause unintended harm to systems on which they are used, or may not be consistent with applicable user agreements. Your use of these hacks is at your own risk, and O'Reilly Media, Inc. disclaims responsibility for any damage or expense resulting from their use. In any event, you should take care that your use of these hacks does not violate any applicable laws, including copyright laws.



This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 0-596-10161-9  
[M]

[5/06]

This excerpt is protected by copyright law. It is your responsibility to obtain permissions necessary for any proposed use of this material. Please direct your inquiries to [permissions@oreilly.com](mailto:permissions@oreilly.com).

**HACK**  
**#10****Add a Google Map to Your Web Site**

Here's how to get started using the Google Maps API.

At O'Reilly's Where 2.0 conference on June 29, 2005, Google announced an official and documented API for Google Maps. The API makes it possible for anyone to add a Google Map to a web page by cutting and pasting a few lines of JavaScript from the Google Maps Developer's site.

People reacted to the new API in one or more ways. My first act was to scratch my own itch by writing a bit of code to display my GPS waypoints on a Google Map. Fortunately, better GPX-to-Google Maps solutions have been created, one of which is documented in "View Your GPS Tracklogs in Google Maps" [Hack #37]. After scratching that itch, I looked to our Geocoder.us site. Schuyler had spent a lot of time figuring out the Census Bureau's public TIGER/Line Map Server API, and how to display the resulting map with a neat little zoomable interface. The results were slow and clunky, but they worked.

The Google Maps API gets rid of the need for that level of head scratching! The march of progress in computers (possibly in society at large) works by first figuring out ways to do new things, and then progressively making those tasks easier, and leaving the old practitioners to eat cat food and write programs for their Osborne luggable computer.

I used Google Maps to bring the geocoder.us site into the protective embrace of the Google Maps API. Geocoder.us, online at <http://geocoder.us/>, is a free U.S. address geocoder. You can go to the web site and get the latitude and longitude for a U.S. street address. You can also use a web service interface to get the latitude and longitude automatically for a group of addresses [Hack #62]. You can geocode using Google Maps by scraping their search results, but it's not a part of the official API, and doing so violates Google's terms and conditions of service. By contrast, the Geocoder.us site is based on free data without limited terms of service for non-commercial use.


Figure 2-1 shows the results of geocoding the address of O'Reilly Media's headquarters with the original TIGER/Line map, with a pushpin showing the location of the address that we just looked up. We'd like to replace this somewhat slow map generated by the Census Bureau with the much faster, more attractive, and more easily navigable maps offered by Google Maps. (The original Geocoder.us map view can be seen at [http://geocoder.us/demo\\_tiger.cgi](http://geocoder.us/demo_tiger.cgi).)

## Get a Developer Key

The first step in putting a Google Map on your page is to generate a developer's key, which is an alphanumeric string that identifies your web site to

**geocoder.us**  
find the latitude & longitude of any US address - for free

---

|                  |   |   |
|------------------|---|---|
| <b>Address</b>   | 1005 Gravenstein Hwy N<br>Sebastopol CA 95472 |  |
| <b>Latitude</b>  | 38.411908 °                                   |   |
| <b>Longitude</b> | -122.842232 °                                 |   |

**Search for another address:**

Figure 2-1. The Census Bureau map originally used by <http://geocoder.us/>

Google, and helps them track usage of Google Maps. Having to sign up for a developer's key can be something of an annoyance, but it's a small price to pay for being able to include free (as in beer) maps on your web site with such relative ease.

You'll need a distinct developer's key for each directory on your site that includes Google Maps. You don't need a key for each individual web page or script. So if you have several pages that generate calls to Google Maps from the same directory, you only need one key.

Fortunately Google has made getting developer's keys as easy as filling in a web form. The Google Maps API page is at <http://www.google.com/apis/maps/>. This includes links to documentation, examples, Terms of Use, and the page to get your key. There is a human version of the Terms of Use, then the full legalese version. Figure 2-2 shows the form with the URL we want to use for our maps. You must agree to the Terms of Service, then click Generate API Key.



In our case, we wanted to enable Google Maps for a single script on our server. If you want to enable Google Maps for a whole directory, you can leave off the script name and just specify the host name and directory portion of the URL. Unfortunately, the API key isn't good for directories inside the one you specify, just the files and scripts in that directory.

Almost instantly, a key will be generated, along with an example web page that Google refers to this as the "Hello World" of Google Maps. To put this

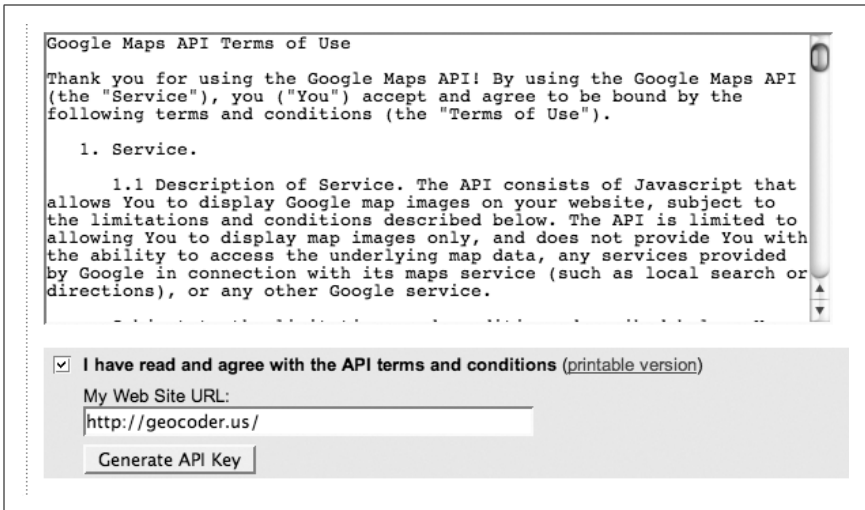


Figure 2-2. Enter a server and path to generate a developer’s key

on your web site, copy the HTML/JavaScript section in Example 2-1 and paste it into a new file on your own web site in the directory that you used when you created the developer’s key.

*Example 2-1. Google Maps “Hello World”*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <script
      src="http://maps.google.com/maps?file=api&v=1&key=[your API key]"
      type="text/javascript"></script>
  </head>
  <body>
    <div id="map"
      style="width: 500px; height: 400px; border: 1px solid #979797"></div>
    <script type="text/javascript">
      //

      var map = new GMap(document.getElementById("map"));
      map.addControl(new GSmallMapControl());
      map.centerAndZoom(new GPoint(-122.1419, 37.4419), 4);

      //]]&gt;
    &lt;/script&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre>
</div>
<div data-bbox="122 931 553 949" data-label="Page-Footer">
<p>38 | Chapter 2, Introducing the Google Maps API</p>
</div>
```



Developer keys work only when they are used on a web page that lives in the server and directory that you specified when you created the key. So you can't copy this listing and have it work until you change the developer's key to match your own. In general, most of the code examples in this book will require you to substitute your own valid developer key in order for them to work.

## Hello, World!

The “Hello World” page shown in Example 2-1 is a standard HTML page, with a bit of JavaScript. The first interesting part is the opening script element:

```
<script
src="http://maps.google.com/maps?file=api&v=1&key=[Your API Key]"
type="text/javascript"></script>
```

This imports the Google Maps JavaScript library into our page. A JavaScript-compliant browser will automatically fetch the contents of the provided URL. Google can then compare the developer's key and the server name and path that is included in the HTTP headers of your request with their records, to see if they match.



The `v=1` parameter in the above URL is important, because it specifies the Google Maps API version that your script expects. If Google ever changes 'its API in such a way that backwards compatibility is broken, the `v` parameter will allow your script to continue to function with the original API and give you some breathing room to update your code to the newer version of the API.

The next three interesting lines are:

```
var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.centerAndZoom(new GPoint(-122.1419, 37.4419), 4);
```

These lines are pretty much self explanatory (for an object-oriented JavaScript programmer). But you don't need to understand much to put powerful maps on your own pages!

By default the size of the map is determined by the size of the HTML element that contains the map. In this example, we are using the `div` element to define a division in the page, which provides an area that you can control and format independently from other parts of the page.

The first line creates a new `GMap` object and places it within the `div` named `map`. (There's nothing magic about the name of the `div` element, by the way—we could call it “Tim,” and so long as the JavaScript mentioned the

same name, it would still work.) The next line adds the small pan and zoom control to the map, and the third line centers and zooms the map to longitude -122.1419, latitude 37.4419 at zoom level 4.

In our example, the `div` element is 500×400 pixels high and has a 1-pixel-wide gray border around the edge. You can also specify the width and height in percentages, such as `style="width: 50%; height: 40%"`. The border itself is totally optional, but it does set the map off nicely from the rest of the page.

```
<div id="map"
  style="width: 500px; height: 400px; border: 1px solid #979797"></div>
```

The *demo.cgi* page at <http://geocoder.us/> was already template driven, so to add Google Maps functionality I added the `script=` line to load the Google Maps library, and then included these lines in my template:

```
<div id="map" style="width: 500px; height: 300px; border: 1px solid
#979797"></div>
<script type="text/javascript">
//

var map = new GMap(document.getElementById("map"));
map.addControl(new GSmallMapControl());
map.centerAndZoom(new GPoint([% long %], [% lat %]), 4);

var point = new GPoint([% long %],[% lat %]);
var marker = new GMarker(point);
map.addOverlay(marker);

//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="120 571 879 687" data-label="Text"><p>The map will automatically size itself to fit within the <code>&lt;div id="map"...&gt;</code> tag. In our templating system (Perl’s Template Toolkit, as it happens), <code>[% long %]</code> will be replaced with the contents of the variable <code>long</code>, or the longitude. The only differences from the sample code are that the sample constants for <code>lat</code> and <code>long</code> are replaced with variables that will be set in our program, and that a point marker is added for the location of the address the user looked up.</p></div><div data-bbox="120 708 432 729" data-label="Section-Header"><h2>Getting Outside of Your Head</h2></div><div data-bbox="120 736 880 852" data-label="Text"><p>The “Hello World” example presumes that the HTML <code>script</code> element that imports the Google Maps API library into your web page is nestled safe within the HTML document’s <code>head</code> element. Certainly, this is the right place for it to go, but web browsers are perfectly capable of handling <code>script</code> elements elsewhere in an HTML document. Furthermore, situations will occur where you might want to include the API library from elsewhere—say, for</p></div><div data-bbox="120 930 554 949" data-label="Page-Footer"><hr/>40 | Chapter 2, Introducing the Google Maps API</div>
```

example, one where you have an HTML templating system that provides a boilerplate header and footer for each page on your site. In this circumstance, you don't want the API library to be imported into every page on the site, because every page outside the directory associated with your developer's key will load up with a developer key error message.

Fortunately, you can indeed import the API library almost anywhere in your document, so long as it appears before the JavaScript code that needs to use it. The only thing you really need to know is that some browsers—Internet Explorer, in particular—will wait for a script element to execute before rendering the rest of the page, to make sure that the JavaScript itself doesn't modify the page layout. For some reason, this behavior sometimes has a bad interaction with the Google Maps API when the script element is used outside the head—a JavaScript execution error is the most common result. The workaround is to add a `defer="defer"` attribute to the script element, which will tell the browser not to worry about it and get on with rendering the page. In that case, our earlier script element example looks like this:

```
<script src="http://maps.google.com/maps?file=api&v=1&key  
=[Your API Key]"type="text/javascript" defer="defer"></script>
```

## Getting Right to the Point

Once you've got a Google Map on your page, adding points to it is easy. You'll first create a new `GPoint` object, then create a marker icon at that point, and finally add that marker to the map. We'll look more at adding points and lines to Google Maps in other hacks. For now, enjoy Figure 2-3, which shows a pretty Google Map replacing our TIGER map.

But is that (always) better? Are there reasons not to use Google Maps? Yes! Google Maps are great, and Google has a history and reputation of being the good guys, but it is a profit-making business and its goals might not be your goals. The Google Maps terms of service are extremely generous, but when you use Google Maps, you are relying on Google. There are restrictions on what you can do with Google Maps; for example, Google Maps cannot be used on a site that is inaccessible to the general public, such as a paid premium content site or a corporate intranet. There are limitations on volume, as well: if you expect more than 50,000 hits in a day, Google expects to hear from you first. You can't do certain things, such as scrape Google's images or remove its imprint from its imagery, and it has explicitly reserved the right to put ads on the maps at any time. You can read more about the fine details at <http://www.google.com/apis/maps/faq.html>, but you should also review the terms of use at <http://www.google.com/apis/maps/terms.html> to be on the safe side.

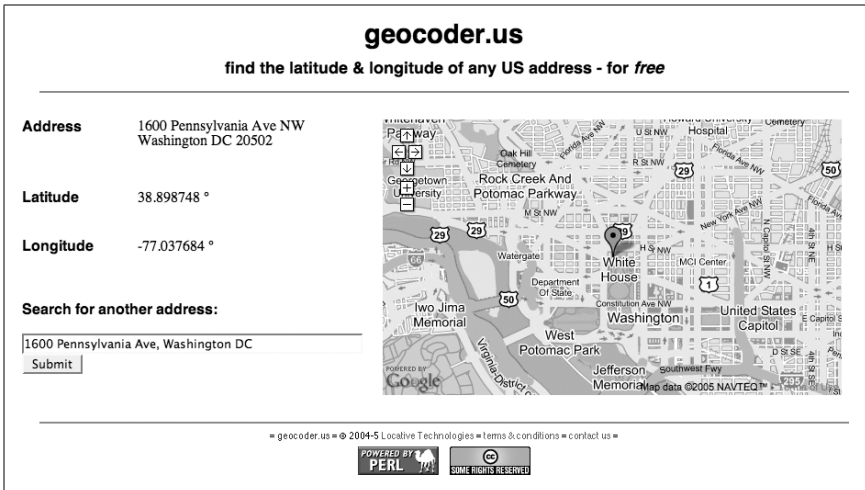


Figure 2-3. <http://geocoder.us/> with a Google Map

There are (at least currently) limits on the data available from Google. There is far more aerial and satellite data and map imagery available on the Web [Hack #12] from public Web Mapping Service (WMS) servers than is available from Google.

### See Also

- Google Maps are free-as-in-beer but not free-as-in-speech. So if the power, beauty, and ease of use of Google Maps don't meet your needs, projects such as Geoserver (<http://geoserver.sf.net/>), Mapserver (<http://mapserver.gis.umn.edu/>), and the Ka-Maps client interface to Mapserver (<http://ka-maps.sf.net/>) may fill the bill. The downside, as is often the case with open source software, is that you may have to do more of the work yourself! O'Reilly's *Mapping Hacks* and *Web Mapping Illustrated* have much more to say about free and open source mapping solutions.