Beginning Google Maps Applications with PHP and Ajax

From Novice to Professional

Michael Purvis
Jeffrey Sambells
and Cameron Turner

Apress®

Beginning Google Maps Applications with PHP and Ajax: From Novice to Professional

ISBN-13 (pbk): 978-1-59059-707-1

ISBN-10 (pbk): 1-59059-707-9

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Jason Gilmore
Technical Reviewer: Terrill Dent
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade
Project Manager: Elizabeth Seymour
Copy Edit Manager: Nicole LeClerc
Copy Editor: Marilyn Smith
Assistant Production Director: Kari Brooks-Copony
Production Editor: Katie Stence
Compositor: Kinetic Publishing Services, LLC
Proofreader: Liz Welch
Indexer: Beth Palmer
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

The source code for this book is available to readers at http://www.apress.com in the Source Code section or at the official book site, http://googlemapsbook.com.

To Anne and Jim, that with God's grace,
I might one day be so loving a parent.
—Michael Purvis

Dedicated to my loving wife, Stephanie, always by my side as my navigator in life.
May we never lose our way in this crazy world.
And also to my parents, Frank and Linda,
who taught me to always look beyond the horizon.
—Jeffrey Sambells

I dedicate this book to my amazing wife, Tanya, and our son, Owen.
Tanya is the ultimate teammate and life partner—
always willing to dive into an adventure or opportunity regardless of the size.
I'd also like to thank my parents, Barry and Lorna, for supporting me
in all my ambitions and encouraging me to take risks and pursue dreams.
Without all of you, I would never have agreed to write my first book
about a moving-target topic such as Google Maps,
on a compressed timeline, with a newborn baby!
To everyone else who helped out in the last few months, thank you.
We couldn't have completed this book without your help and patience.
—Cameron Turner

Contents at a Glance

Contents

Foreword

In the Beginning...

In the history of the Internet, 2005–2006 will be remembered as the year when online mapping finally came of age. Prior to 2005, MapQuest and other mapping services allowed you to look up directions, search for locations, and map businesses, but these searches were limited, usually to the companies the services had partnered with, so you couldn't search for any location. On February 8, 2005, Google changed all that. As it does with many of its services, Google quietly released the beta of Google Maps to its Labs incubator (`http://labs.google.com`) and let word-of-mouth marketing promote the new service.

By all accounts, Google Maps was an instant hit. It was the first free mapping service to provide satellite map views of any location on the earth, allowing anyone to look for familiar places. This started the "I can see my house from here" trend, and set the blogosphere abuzz with links to Google Maps locations around the world.

Like other mapping services, Google Maps offered directions, city and town mapping, and local business searches. However, what the Google Maps engineers buried within its code was something that quickly set it apart from the rest. Although unannounced and possibly unplanned, they provided the means to manipulate the code of Google Maps to plot your own locations. Moreover, you could combine this base mapping technology with an external data source to instantly map many location-based points of information. And all of this could be done on privately owned domains, seemingly independent of Google itself.

At first, mapping "hackers" unlocked this functionality, just as video gamers hack into games by entering simple cheat codes. They created their own mapping services using Google Maps and other sources. One of the first these was `Housingmaps.com`, which combined the `craigslist.org` housing listings with a searchable Google Maps interface. Next came Adrian Holovaty's `chicagocrime.org`, which offered a compelling way to view crime data logged by the Chicago Police Department. These home-brewed mapping applications were dubbed "hacks," since Google had not sanctioned the use of its code in external domains on the Web.

The major change came in June 2005, when Google officially introduced the Google Maps API, which is the foundation for this book. By releasing this API, Google allowed programmers the opportunity to build an endless array of applications on top of Google Maps. Hundreds of API keys were registered immediately after the announcement, and many sites integrating Google Maps appeared within days. The *map mashup* was born.

The Birth of the Google Maps Mania Blog

The Google Maps labs beta site had been public for barely a month when I tried it for the first time. I was fascinated. While combing through the blogosphere looking for more information, I started to see a trend toward Google Maps hacks, how-to sites, Firefox extensions, and websites indexing specific satellite images. I thought that others could benefit from an aggregation of all of these ideas into one themed blog. Thus, my Google Maps Mania blog was born.

Google Maps Mania is more accurately described as a "meta-site," as host Leo Laporte pointed out when I was a guest on his NPR G4techTV radio show in November 2005.

April 13, 2005, saw these as my first posts:

> **Title: Google Maps Mania**
>
> If you're like me you were absolutely floored when Google came out with the Google Maps service. Sure, it's just another mapping service. Until you realize it's full potential. The ability to toggle between regular street/road maps and a satellite view is unreal. I've started to see a lot of buzz around the blogging community about Google Maps so I've decided to help you keep up with the Google Maps related sites, blogs and tools that are cropping up. Stay tuned.
>
> **Title: Google Sightseeing**
>
> The first Google Maps related site of note is Google Sightseeing. This blog tracks interesting satellite shots as submitted by its visitors, then organizes them by interest area like buildings, natural landmarks and stadiums. It's a pretty nifty site. Google Sightseeing even posted my suggestion of Toronto's Rogers Centre (Skydome) and the CN Tower!
>
> **Title: Flickr Memory Maps**
>
> Here's a Flickr group that took off fast. Memory Maps is a Flickr group that contains maps with captions describing memories they have of those areas or specific notes about different areas. Kind of cool.
>
> **Title: Make your own multimedia Google map**
>
> Google Blogoscoped tipped me off on this link. Seems Engadget has a page which gives some pretty good directions on how to create your own annotated multimedia Google map. There is some pretty serious direction here which includes inserting pictures and movies from the annotations. I'd like to see an example of this.
>
> **Title: My GMaps**
>
> myGmaps enables you to create, save and host custom data files and display them with Google Maps. Create push-pin spots on any map of your choice. Mark your house, where an event will be held, or the route of a fun-run as a few examples. Then you can publish the map that you've created to your own website.

These postings represented an interesting cross-section of the ideas, concepts, and websites that I had come across in the two short months since Google Maps came to the Web. In the year between the start of Google Maps Mania and the release of the second-generation API (which this book is based on) in April 2006, I have made over 900 posts and attracted more than 6,000 daily readers to the blog, including the architects of the API itself. I've been Slashdotted, Dug (at Digg), and linked to from the *New York Times* site, as well as the sites of hundreds of other mainstream papers and magazines. In June 2006, Google arranged for my entire family to travel across the country so I could speak at the Google Geo Developer Day in advance of the Where 2.0 conference.

So many interesting mashups have been created using the Google Maps API that it's becoming impossible to keep up with all of them. I liken this to the early days of the Web when search directories began to manually catalog new web pages as they came online. The volume of new sites quickly became too huge to handle manually, and Google itself was born.

You can see why the Google Maps API offers the key for the next killer apps on the Web. It has been the missing link to take the Web to the next level.

This book will provide you the means to take part in this evolution of the Web. I hope to be posting about the interesting and unique map creations that you build after reading this book. Your creations will inspire others to do similar things, and together, we will continue to grow the Internet, one mapping application at a time. Let me know if you build something cool!

Mike Pegg
Google Maps Mania (`http://www.gmapsmania.com`)

About the Authors

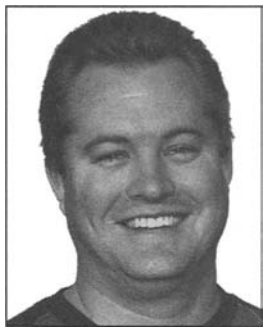
MICHAEL PURVIS is a Mechatronics Engineering student at the University of Waterloo, in Ontario. He is a mostly self-taught programmer. Prior to discovering PHP, he was busy making a LEGO® Mindstorms kit play Connect 4. Currently, he maintains an active community site for classmates, built mostly from home-brewed extensions to PunBB and MediaWiki.

He has written about CSS for *Position Is Everything*, and occasionally participates in the css-discuss mailing list. He particularly enjoys those clever layouts that mix negative margins, relative positioning, and bizarre float tricks to create fiendish, cross-browser, flexible-width concoctions. These and other nontechnical topics are discussed on his weblog at uwmike.com.

Offline, he enjoys cooking, cycling, and social dancing. He has worked with We-Create, Inc. on a number of PHP-based projects, and has a strong interest in independent web standards.

JEFFREY SAMBELLS is a graphic designer and self-taught web applications developer best known for his unique ability to merge the visual world of graphics with the mental realm of code. With a Bachelor of Technology degree in Graphic Communications Management along with a minor in Multimedia, Jeffrey was originally trained for the traditional paper-and-ink printing industry, but he soon realized the world of pixels and code was where his ideas would prosper. In late 1999, he cofounded We-Create, Inc., an Internet software company based in Waterloo, Ontario, which began many long nights of challenging and creative innovation. Currently, as Director of Research and Development for We-Create, Jeffrey is responsible for investigating new and emerging Internet technologies and integrating them using web standards-compliant methods. In late 2005, he also became a Zend Certified Engineer.

When not playing at the office, Jeffrey enjoys a variety of hobbies from photography to woodworking. When the opportunity arises, he also enjoys floating in a canoe on the lakes of Algonquin Provincial Park or going on an adventurous, map-free, drive with his wife. Jeffrey also maintains a personal website at JeffreySambells.com, where he shares thoughts, ideas, and opinions about web technologies, photography, design, and more. He lives in Ontario, Canada, eh, with his wife, Stephanie, and their little dog, Milo.



CAMERON TURNER has been programming computers since his first VIC 20 at age 7. He has been developing interactive websites since 1994. In 1999, he cofounded We-Create, Inc., which specializes in Internet software development. He is now the company's Chief Technology Officer. Cam obtained his Honors degree in Computer Science from the University of Waterloo with specialization in applied cryptography, database design, and computer security.

Cam lives in Canada's technology capital of Waterloo, Ontario, with his wife, Tanya, son Owen, and dog Katie. His hobbies include biking, hiking, water skiing, and painting. He maintains a personal blog at CamTurner.com, discussing nontechnical topics, thoughts, theories, and family life.

About the Technical Reviewer

TERRILL DENT is enrolled in Honors Mathematics at the University of Waterloo. His major interests center around Internet culture, twentieth century history, and economic theory. Terrill.ca is home to his weblog, and MapLet.ca is the front for his web application ventures, where he lets his acute attention to detail show through. Apart from work, he busies himself with fine arts, cycling, and an occasional novel.

Acknowledgments

The authors would like to thank Mike Pegg of Google Maps Mania for giving Apress our names when contacted about doing a book on Google Maps. This book would not have been possible without his encouragement, support, generosity, and friendship.

Thanks to Terrill for finding the errors of our bleary-eyed coding sessions and helping make this book what it is today.

Thanks to Jason, Elizabeth, Marilyn, Katie, Julie, and the rest of the team at Apress. We hope that working with us has been as much fun for you as working with you was for us.

PART 1

Your First Google Maps

CHAPTER 1

Introducing Google Maps

It's hard to argue that Google Maps hasn't had a fundamental effect on the mapping world. While everyone else was still doing grainy static images, Google developers quietly developed the slickest interface since Gmail. Then they took terabytes of satellite imagery and road data, and just gave it all away for free.

We're big fans of Google Maps and excited to get started here. We've learned a lot about the Google Maps API since it was launched, and even more during the time spent writing and researching for this book. Over the course of the coming chapters, you're going to move from simple tasks involving markers and geocoding to more advanced topics, such as how to acquire data, present many data points, and provide a useful and attractive user interface.

A lot of important web technologies and patterns have emerged in parallel with the Google Maps API. But whether you call it Ajax or Web 2.0 is less important than what it means: that the little guy is back.

You don't need an expensive development kit to use the Google Maps API. You don't need a computer science degree, or even a lot of experience. You just need a feel for what's important data and an idea of what you can do to present it in a visually persuasive way.

We know you're eager to get started on a map project, but before we actually bust out the JavaScript, we wanted to show you two simple ways of creating ultra-quickie maps: using KML files and through the Wayfaring map site.

Using either of these approaches severely limits your ability to create a truly interactive experience, but no other method will give you results as quickly.

KML: Your First Map

The map we're working on here is actually Google Maps itself. In June 2006, Google announced that the official maps site would support the plotting of KML files. You can now simply plug a URL into the search box, and Google Maps will show whatever locations are contained in the file specified by the URL. We aren't going to go in depth on this, but we've made a quick example to show you how powerful the KML method is, even if it is simple.

Note KML stands for Keyhole Markup Language, which is a nod to both its XML structure and Google Earth's heritage as an application called Keyhole. Keyhole was acquired by Google late in 2004.

We created a file called `toronto.kml` and placed the contents of Listing 1-1 in it. The paragraph blurbs were borrowed from Wikipedia, and the coordinates were discovered by manually finding the locations on Google Maps.

Listing 1-1. *A Sample KML File*

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.google.com/earth/kml/2">
<Document>
  <name>toronto.kml</name>
  <Placemark>
    <name>CN Tower</name>
    <description>The CN Tower (Canada's National Tower, Canadian National Tower),
    at 553.33 metres (1,815 ft., 5 inches) is the tallest freestanding structure on land.
    It is located in the city of Toronto, Ontario, Canada, and is considered the
    signature icon of the city. The CN Tower attracts close to two million visitors
    annually.

    http://en.wikipedia.org/wiki/CN_Tower</description>
    <Point>
      <coordinates>-79.386864,43.642426</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

In the actual file (located at `http://googlemapsbook.com/chapter1/kml/toronto.kml`), we included two more `Placemark` elements, which point to other well-known buildings in Toronto. To view this on Google Maps, paste that URL into the Google Maps search field. Alternatively, you can just visit this link:

```
http://maps.google.com/maps?f=q&hl=en&q=http://googlemapsbook.com/chapter1/kml/
toronto.kml
```

You can see the results of this in Figure 1-1.

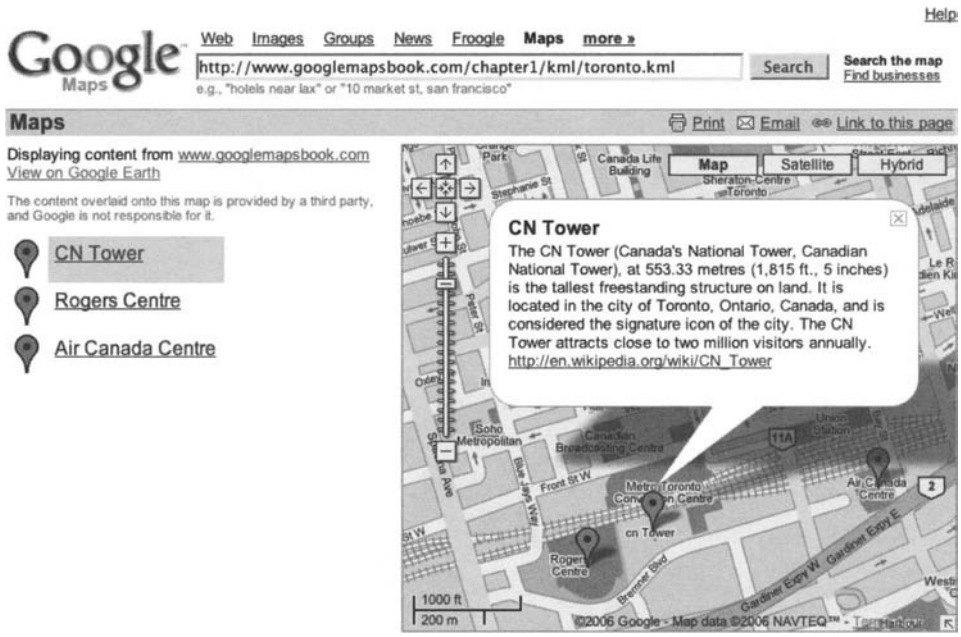


Figure 1-1. *A custom KML data file being displayed at maps.google.com*

Now, is that a quick result or what? Indeed, if all you need to do is show a bunch of locations, it's possible that a KML file will serve your purpose. If you're trying to link to your favorite fishing spots, you could make up a KML file, host it somewhere for free, and be finished.

But that wouldn't be any fun, would it? After all, as cool as the KML mapping is, it doesn't actually offer any interactivity to the user. In fact, most of the examples you'll work through in Chapter 2 are just replicating the functionality that Google provides here out of the box. But once you get to Chapter 3, you'll start to see things that you can do *only* when you harness the full power of the Google Maps API.

Before moving on, though, we'll take a look at one other way of getting a map online quickly.

Wayfaring: Your Second Map

A number of services out there let you publish free maps of quick, plotted-by-hand data. One of these, which we'll demonstrate here, is Wayfaring.com (Figure 1-2). Wayfaring has received attention and praise for its classy design, community features (such as commenting and shared locations), and the fact that it's built using the popular Ruby on Rails framework.

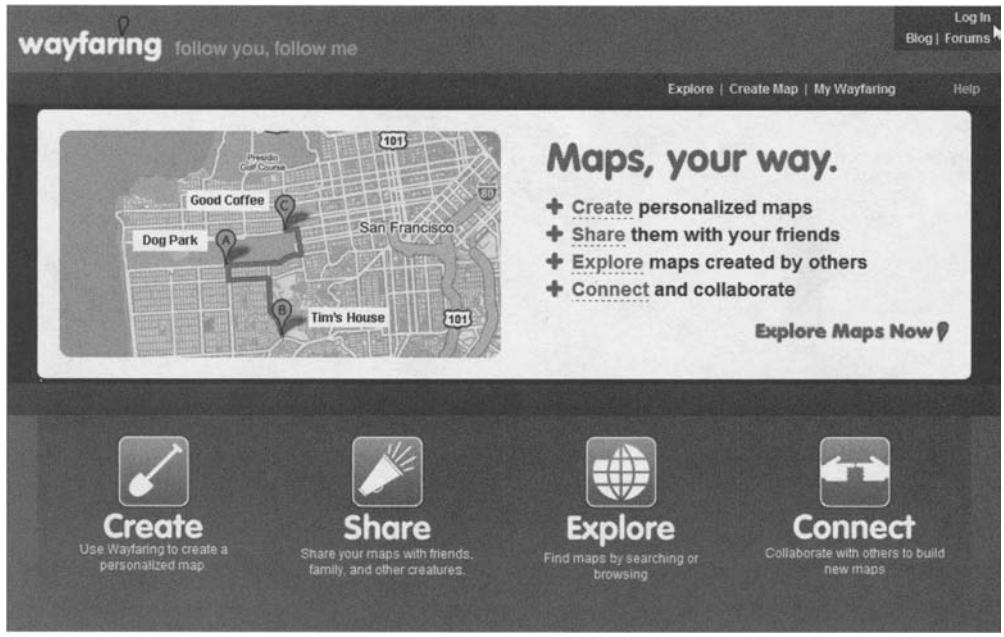


Figure 1-2. *Wayfaring.com home page*

Wayfaring is a mapping service that uses the Google Maps API and allows users to quickly create maps of anything they would like. For example, some people have made maps of their vacations; others have identified interesting aspects of their hometown or city. As an example, we'll walk you through making a quick map of an imaginary trip to the Googleplex, in Mountain View, California.

Point your browser at `http://www.wayfaring.com` and follow the links to sign up for an account. Once you've created and activated your account, you can begin building your map. Click the Create link.

Adding the First Point

We'll start by adding the home airport for our imaginary journey. In our case, that would be Pearson International Airport in Toronto, Ontario, Canada, but you could use the one closest to you. Since Pearson is an international location (outside the United States), we need to drag and zoom the map view until we find it. If you're in the United States, you could use instead the nifty Jump To feature to search by text string. Figure 1-3 shows Pearson nicely centered and zoomed.

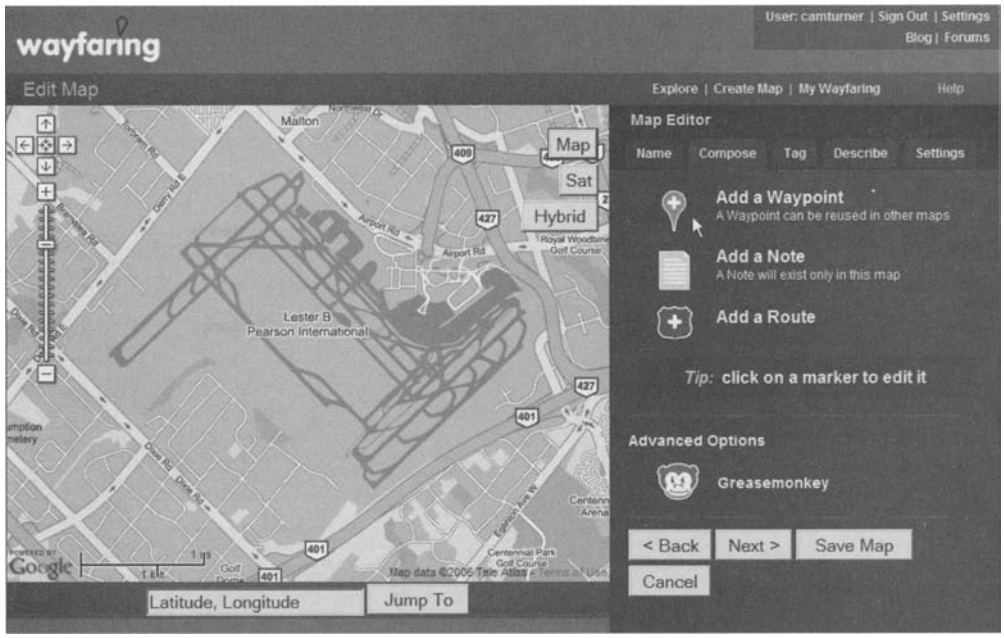


Figure 1-3. *Lester B. Pearson International Airport, Toronto, Ontario*

Once you've found your airport, you can click Next and name the map. After clicking ahead, you should be back at the main Map Editor screen.

Select Add a Waypoint from the list of options on the right. You'll be prompted to name the waypoint. We'll call ours "Lester B Pearson International Airport." However, as we type, we find that Wayfaring is suggesting this exact name. This means that someone else on some other map has already used this waypoint, and the system is giving us a choice of using their point or making one of our own. It's a safe bet that most of the airports you could fly from are already in Wayfaring, so feel free to use the suggested one if you would like. For the sake of completeness, we'll quickly make our own. Click Next to continue.

The next two screens ask you to tag and describe this point in order to make your map more searchable for other members. We'll add the tags "airport Toronto Ontario Canada" and give it a simple description. Finally, click Done to commit the point to the map, which returns you to the Map Editor screen.

Adding the Flight Route

The next element we're going to add to our map is a route. A route is a line made up of as many points as you would like. We'll use two routes in this example. The first will be a straight line between the two airports to get a rough idea of the distance the plane will have to travel to get us to Google's headquarters. The second will be used to plot the driving path we intend to take between the San Francisco airport and the Googleplex.

To begin, click Add a Route, name the route (something like "airplane trip"), and then click your airport. A small, white dot appears on the place you clicked. This is the first point on your line. Now zoom out, scroll over to California, and zoom in on San Francisco. The airport

we'll be landing at is on the west side of the bay. Click the airport here, too. As you can see in Figure 1-4, a second white dot appears on the airport and a blue line connects the two points. You can see how far your flight was on the right side of the screen, underneath the route label. Wow, our flight seems to have been over 2000 miles! If you made a mistake and accidentally clicked a few extra times in the process of getting to San Francisco, you can use the Undo Last option. Otherwise, click Save.

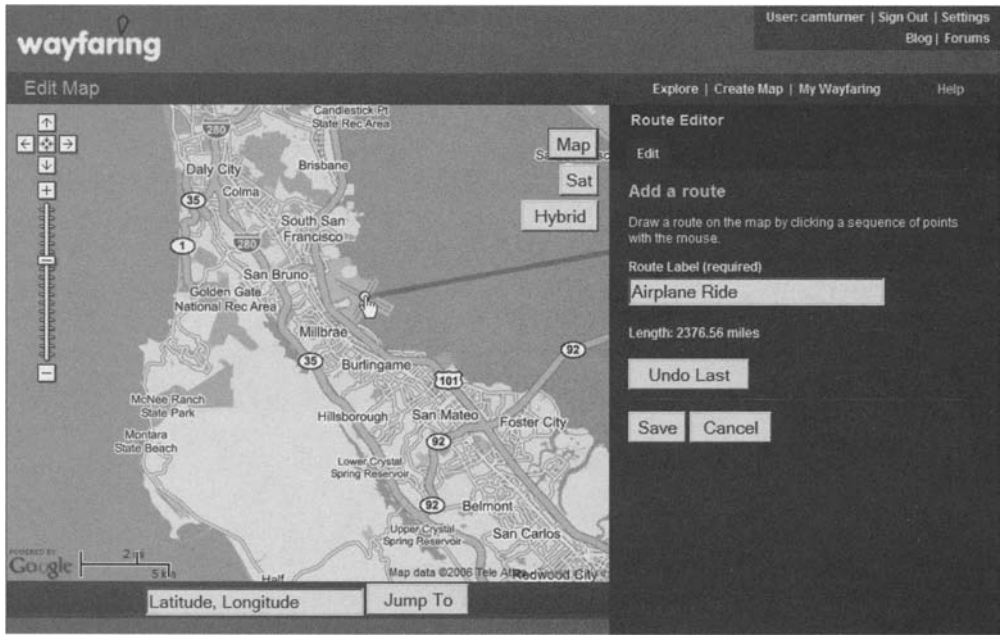


Figure 1-4. *Our flight landing at San Francisco International Airport*

Adding the Destination Point

Now that you're in San Francisco, let's figure out how to get to the Googleplex directly. Click Add a Waypoint. Our destination is Google, so we've called the new point "The Googleplex" and used the address box feature to jump directly to 1600 Amphitheatre Pky, Mountain View, CA 94043. Wayfaring is able to determine latitude and longitude from an address via a process called *geocoding*, which you'll be seeing a lot more of in Chapter 4.

To confirm you're in the right place, click the Sat button on the top-right corner of the map to switch it over to satellite mode. You should see something close to Figure 1-5.

Figure 1-5. *The Googleplex*

Excellent! Save that waypoint.

Adding a Driving Route

Next, let's figure out how far of a drive we have ahead of us. Routes don't really have a starting and ending point in Wayfaring from a visual point of view, so we can start our route from the Googleplex and work our way backwards. Switch back into map (or hybrid) mode so you can see the roads more clearly. From the Map Editor screen, select Add a Route and click the point you just added. Use 10 to 20 dots to carefully trace the trip from Mountain View back up the Bayshore Freeway (US Highway 101) to the airport. By our tracing, we end up with about 23 miles of fun driving on this California highway, as shown in Figure 1-6.

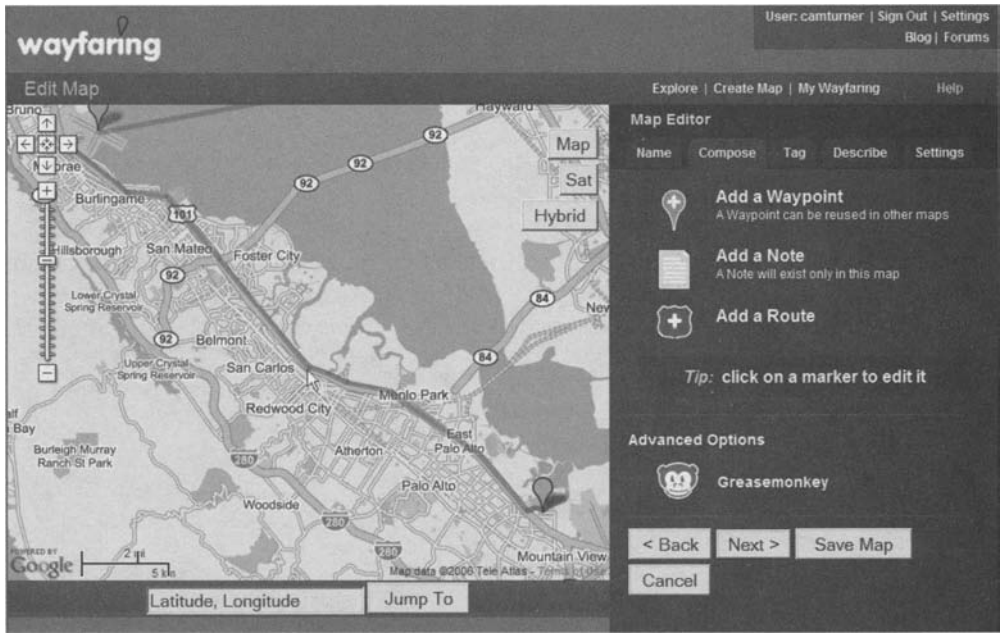


Figure 1-6. *The drive down the Bayshore Freeway to the Googleplex*

That's it. You can use the same principles to make an annotated map of your vacation or calculate how far you're going to travel, and best of all, it's a snap to share it. To see our map live, visit `http://www.wayfaring.com/maps/show/17131`.

Of course, since this *is* a programming book, you're probably eager to dig into the code and make something really unique. Wayfaring may be nice, but the whole point of a mashup is to automate the process of getting a lot of data combined together.

Tip *Mashup* is a term that originates from DJs and other musicians who create new compositions by "mashing" together samples from existing songs. A classic example of this is *The Grey Album*, which joins the a capella versions of tracks from Jay-Z's *The Black Album* with unauthorized clips from *The White Album*, by The Beatles. In the context of this book, *mashup* refers to the mashing of data from one source with maps from Google.

What's Next?

Now that these examples are out of the way, we hope you're eager to learn how to build your own mashups from the ground up. By the end of Part 1 of this book, you'll have the skills to do everything you've just done on Wayfaring (except the route lines and distances, which are covered in Chapter 10) using JavaScript and XHTML. By the book's conclusion, you'll have learned most of the concepts needed to build your own Wayfaring clone!

So what exactly is to come? We've divided the book into three parts and two appendixes. Part 1 goes through Chapter 4 and deals with the basics that a hobbyist would need to get started. You'll make a map, add some custom pins, and geocode a set of data using freely available services. Part 2 (Chapters 5 through 8) gets into more map development topics, like building a usable interface, dealing with extremely large groups of points, and finding sources of raw information you may need to make your professional map ideas a reality. Part 3 (Chapters 9 through 11) dives into advanced topics: building custom map overlays such as your own info window and tooltip, creating your own map tiles and projections, using the spherical equations necessary to calculate surface areas on the earth, and building your own geocoder from scratch. Finally, one appendix provides a reference guide to the Google Maps version 2 API, and another points to a few places where you can find neat data for extending the examples here, and to inspire your own projects.

We hope you enjoy!

CHAPTER 2

Getting Started

In this chapter, you'll learn how to create your first Google map project, plot some markers, and add a bit of interactivity. Because JavaScript plays such a central role in controlling the maps, you'll also start to pick up a few essentials about that language along the way.

In this chapter, you'll see how to do the following:

- Get off the ground with a basic map and a Google Maps API key.
- Separate the map application's JavaScript functions, data, and XHTML.
- Unload finished maps to help browsers free their memory.
- Create map markers and respond to clicks on them with an information pop-up.

The First Map

In this section, you'll obtain a Google Maps API key, and then begin experimenting with it by retrieving Google's starter map.

Keying Up

Before you start a Google Maps web application, you need sign up for a Google Maps API key. To obtain your key, you must accept the Google Maps API Terms of Use, which stipulate, among other things, that you must not steal Google's imagery, obscure the Google logo, or hold Google responsible for its software. Additionally, you're prevented from creating maps that invade privacy or facilitate illegal activities.

Google issues as many keys as you need, but separate domains *must* apply for a separate key, as each one is valid for only a specific domain and subdirectory within that domain. For your first key, you'll want to give Google the root directory of your domain or the space in which you're working. This will allow you to create your project in any subdirectory within your domain. Visit `http://www.google.com/apis/maps/signup.html` (Figure 2-1) and submit the form to get your key. Throughout this book, nearly all of the examples will require you to include this key in the JavaScript `<script>` element for the Google Maps API, as we're about to demonstrate in Listing 2-1.