developerWorks®

Mastering Ajax, Part 9: Using the Google Ajax Search API

Take advantage of public APIs in your asynchronous applications

Skill Level: Intermediate

Brett McLaughlin
Author and Editor
O'Reilly Media Inc.

23 Jan 2007

Making asynchronous requests isn't just about talking to your own server-side programs. You can also communicate with public APIs like those from Google or Amazon, and add more functionality to your Web applications than just what your own scripts and server-side programs provide. In this article, Brett McLaughlin teaches you how to make and receive requests and responses from public APIs like those supplied by Google.

So far in this series, the focus was exclusively on situations in which your client Web pages make requests to your server-side scripts and programs. This is how probably 80 to 90 percent of Ajax applications -- asynchronous Web applications that use the `XMLHttpRequest` object -- work. However, there is a fairly serious limitation in this approach: you're limited by your own ingenuity and programming skills, or at least by the ingenuity and programming skills of the programmers on your team, in your company.

Almost certainly, sometimes you find'll that you really want to do something, but you don't have the technical knowledge necessary to achieve that goal. Perhaps you don't know some piece of syntax, or how to come up with the right algorithm. In other cases, you might not have the data or resources (either people-resources or data-resources) to fill your needs. It's in these cases that you might find yourself thinking, "Gee, if I could just use that *other* person's code!" This article is meant to address that particular case.

Open source scripts and programs

Before I get into the meat of this article -- using public APIs in your Web applications -- it's worth mentioning the existence of open source scripts and programs. Without getting too deeply into detail, *open source* is a term used to describe code that can be, to some degree, freely used and reused in your own applications. Consult Resources for relevant links, but at a 10,000-foot view, you can take open source code that someone else has written, and simply drop it into your own environment, without charge or (much) restriction.

If you make use of open source code, sometimes you'll have to add some extra documentation to your applications, or perhaps contribute changes you make to the open source program or script back to the community. Whatever the particulars of the program that you use, the end result is that you can use code that you didn't have to write, and perhaps wouldn't have been able to write without a lot of help and resources that you don't have. Projects like Apache make it easy to take advantage of the work that others have done -- and don't worry; they want you to use their work!

Articles and tutorials online

It would also be pretty silly in an article published on IBM developerWorks not to mention the wealth of resources, from articles to tutorials to white papers, available on the Internet. There are literally hundreds of thousands of pieces of instruction online, and you could probably find close to a thousand articles about Ajax -- I'm already almost up to ten articles in this series alone! Most of these articles have working code, examples, downloads, and all sorts of other goodies available for your use.

If you don't have the ability to code the server-side program or script you want to use, or can't find an open source program or script to do what you need, hop over to Google and try entering in a basic description of what you're looking for. You'll often find an article, or tip, or some other snippet online that might help you do just what you need. Hop over to developerWorks and do the same; often, you'll find exactly the piece of code, or even entire script, that you were looking for, complete with helpful comments and a description of how it works.

Using public APIs

Many times, you'll face a problem that isn't just technical. You don't need help writing a particular script or piece of code; rather, you need data or resources that you simply don't have. In these cases, even if a tutorial or open source script were available to help you, you'd still need more. Consider, for example, a case in which you want to place a search engine on your Web page. That presumes that you have the data that you want to search -- but what if you want to search beyond the data your company or organization has available?

In cases where you are limited not by technical ability, but by data, a *public API* might help you solve your problem. A public API allows you to use a program hosted on someone else's servers, using someone else's data. In general, the API itself defines how you interact with the program. For instance, a public API to the Google search engine would let you make search requests, but Google's code would search Google's data, and return the results to your program. You not only get the benefit of someone else's technical skills in writing these programs, but you also get the benefit of data stores far beyond your or your company's ability to support.

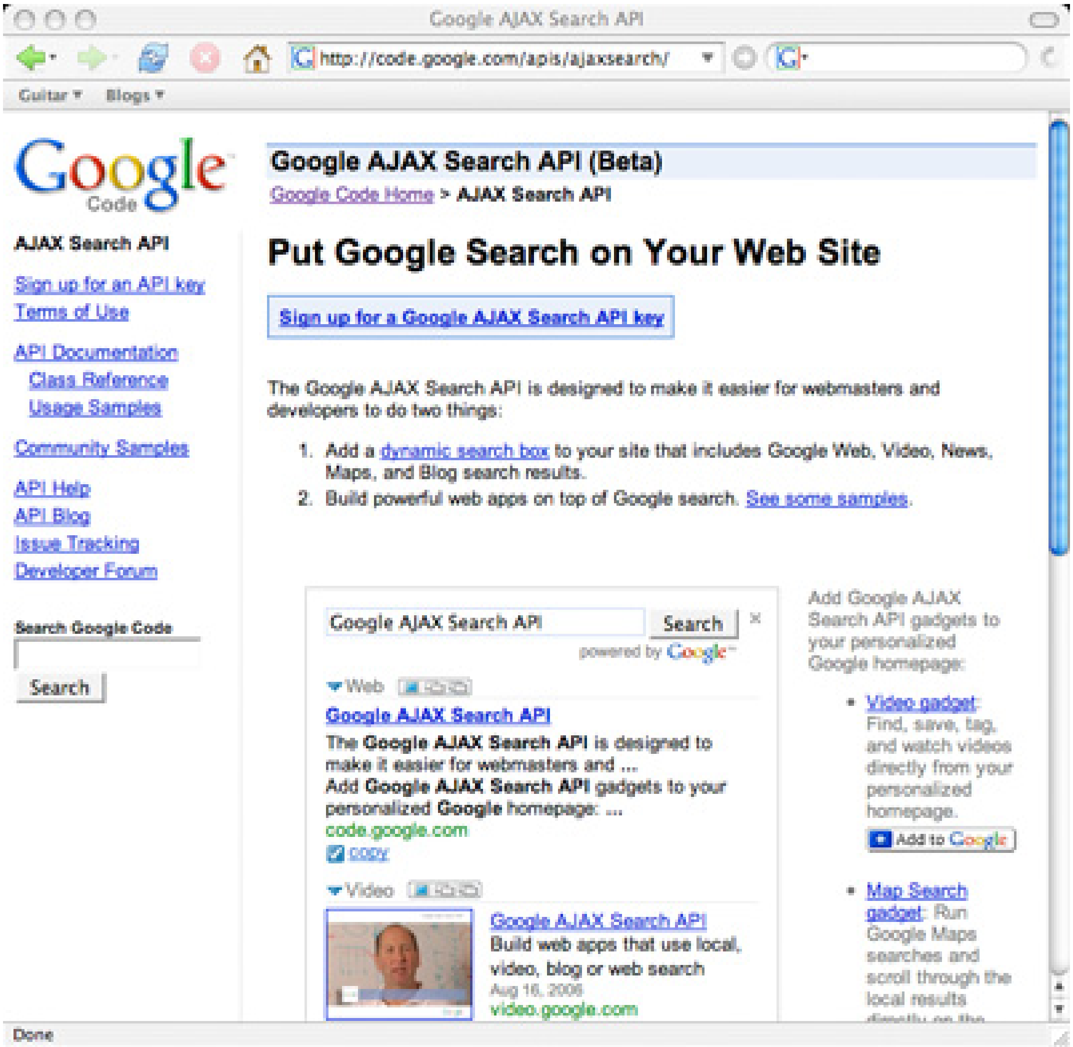
Getting set up to use the Google Ajax Search API

Google remains, arguably, the breakthrough application of the online era. Grandmothers and four-year-olds know about Google, even if they don't understand how anything else online works. Google runs such a massively popular and useful search engine, and seems so committed to providing (mostly) free services that it's no surprise that it has a public API available for you to use in your own programs. In this section, you'll get set up to use the Google API, and prepare to figure out exactly how you can make an asynchronous application talk to Google.

Getting a developer key from Google

This article will focus specifically on Google's Ajax Search API. You can find out more online about this particular API by visiting the Google Ajax Search API home page, which is shown in Figure 1. (See Resources for a link.)

Figure 1. Google's Ajax Search API page



Google AJAX Search API (Beta)

Google Code Home > AJAX Search API

AJAX Search API

Sign up for an API key
Terms of Use

API Documentation
Class Reference
Usage Samples

Community Samples

API Help
API Blog
Issue Tracking
Developer Forum

Search Google Code

Put Google Search on Your Web Site

Sign up for a Google AJAX Search API key

The Google AJAX Search API is designed to make it easier for webmasters and developers to do two things:

1. Add a dynamic search box to your site that includes Google Web, Video, News, Maps, and Blog search results.
2. Build powerful web apps on top of Google search. See some samples.

Google AJAX Search API

powered by Google

▾Web

Google AJAX Search API
The **Google AJAX Search API** is designed to make it easier for webmasters and ...
Add **Google AJAX Search API** gadgets to your personalized **Google** homepage: ...
code.google.com
copy

▾Video

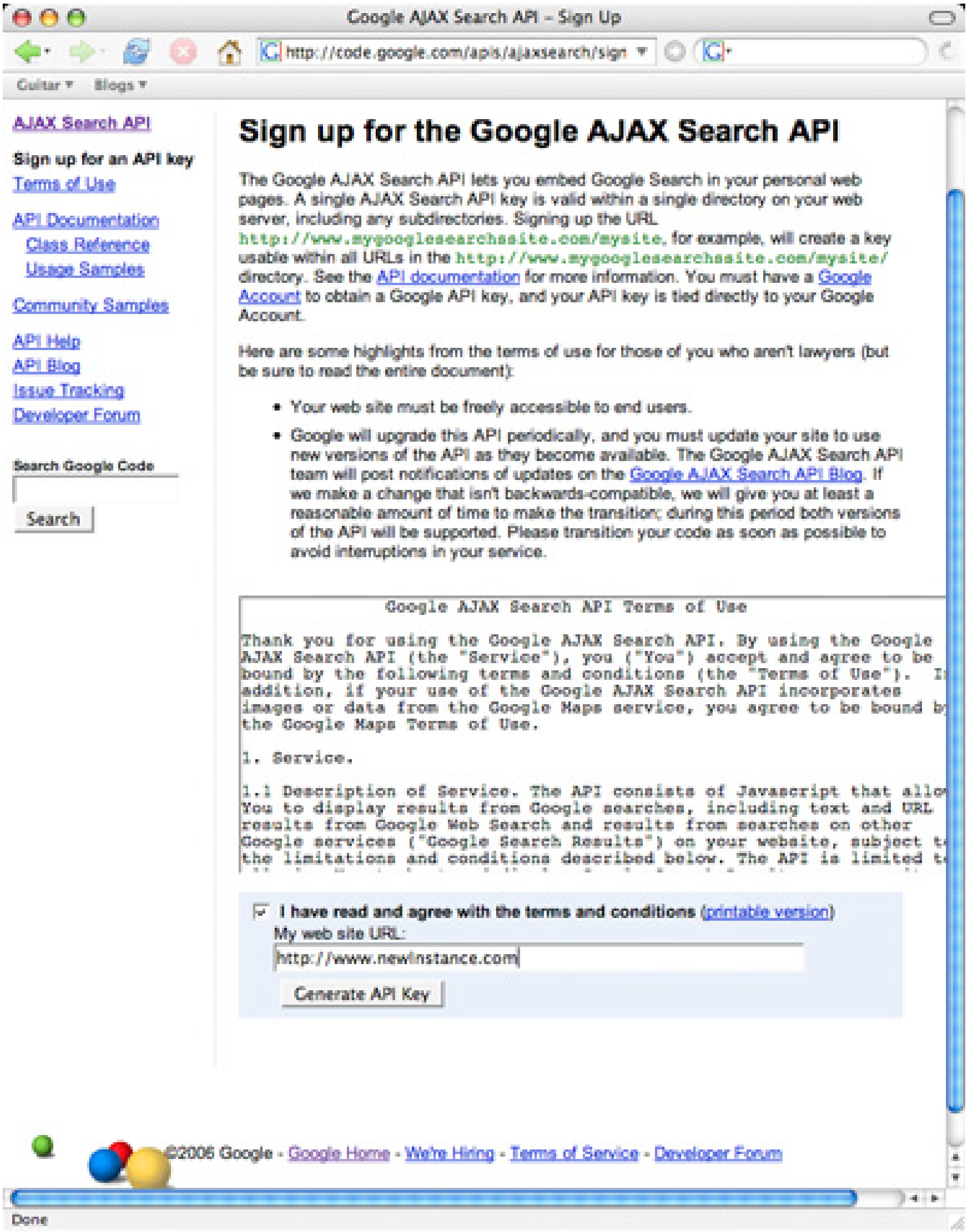
Google AJAX Search API
Build web apps that use local, video, blog or web search
Aug 16, 2006
video.google.com

Add Google AJAX Search API gadgets to your personalized Google homepage:

- Video gadget: Find, save, tag, and watch videos directly from your personalized homepage.
- Map Search gadget: Run Google Maps searches and scroll through the local results directly on the

Your first step is to click the **Sign up for a Google AJAX Search API key** link. This will take you to a page where you can sign up to use the Google API. You'll need to accept some terms of use -- all of which are pretty harmless, as far as I can tell -- and supply the URL of the Web site where your application will run (see Figure 2).

Figure 2. Signing up for Google's Ajax Search API



Sign up for the Google AJAX Search API

The Google AJAX Search API lets you embed Google Search in your personal web pages. A single AJAX Search API key is valid within a single directory on your web server, including any subdirectories. Signing up the URL http://www.mygooglesearchsite.com/mysite, for example, will create a key usable within all URLs in the http://www.mygooglesearchsite.com/mysite/ directory. See the API documentation for more information. You must have a Google Account to obtain a Google API key, and your API key is tied directly to your Google Account.

Here are some highlights from the terms of use for those of you who aren't lawyers (but be sure to read the entire document):

- Your web site must be freely accessible to end users.
- Google will upgrade this API periodically, and you must update your site to use new versions of the API as they become available. The Google AJAX Search API team will post notifications of updates on the Google AJAX Search API Blog. If we make a change that isn't backwards-compatible, we will give you at least a reasonable amount of time to make the transition; during this period both versions of the API will be supported. Please transition your code as soon as possible to avoid interruptions in your service.

```
                 Google AJAX Search API Terms of Use

Thank you for using the Google AJAX Search API. By using the Google
AJAX Search API (the "Service"), you ("You") accept and agree to be
bound by the following terms and conditions (the "Terms of Use"). I
addition, if your use of the Google AJAX Search API incorporates
images or data from the Google Maps service, you agree to be bound b
the Google Maps Terms of Use.

1. Service.

1.1 Description of Service. The API consists of Javascript that allo
You to display results from Google searches, including text and URL
results from Google Web Search and results from searches on other
Google services ("Google Search Results") on your website, subject t
the limitations and conditions described below. The API is limited t
```

☑ I have read and agree with the terms and conditions (printable version)

My web site URL: http://www.newinstance.com

Generate API Key

What URL should I use?

The URL that Google asks for is more or less actually the *domain* where you run your site. If you have your own domain, like I do, you can just use http://www.newinstance.com (replace my domain with your own, obviously). You only need to be more specific if your site uses a subdomain or a particular path within a larger domain; if that's the case, you might need to use a URL like http://www.earthlink.net/~bmclaugh or http://brett.earthlink.net. Other than those special cases, though, don't try to be overly specific with the URL you supply to Google -- just give it the root URL you use to access your entire Web site, and you can use the API anywhere within that URL.

Once you've read the agreement and checked the checkbox, enter in your URL, click **Generate API Key,** and wait a second or two. At this point, you'll have to log in to Google, or create an account. This is a pretty standard process, and you should be able to handle it on your own. Once you're finished, you'll get a nice reply screen back, which gives you a very long key, repeats back your URL, and even gives you a sample page. The key will look something like this:

```
ABQIAAAAjtuhyCXrSHOFSz7zK0f8phSA9fWLQO3TbB2M9BRePlYkXeAu8lHeUgfgRs0eIWUaXg
```

Google's API documentation

Before you start to use the key you've just been assigned, you take some time to check out Google's API documentation (there's a link at the bottom of the page that supplied your key, and that link is repeated in the Resources of this article). Even though you can get a really good start through this article, you'll find that Google's API documentation is a good read, and will probably give you some interesting ideas for how you can use Google on your own site, in your own unique applications.

The simplest Google search Web application

To get you used to seeing what's possible, let's take the sample Web page that Google provides, change it up just a bit, and see how it works.

Creating a search box

Listing 1 shows a pretty simple Web page; go ahead and type it into your favorite editor, save it, and upload it to the domain or URL that you supplied to Google in the previous section.

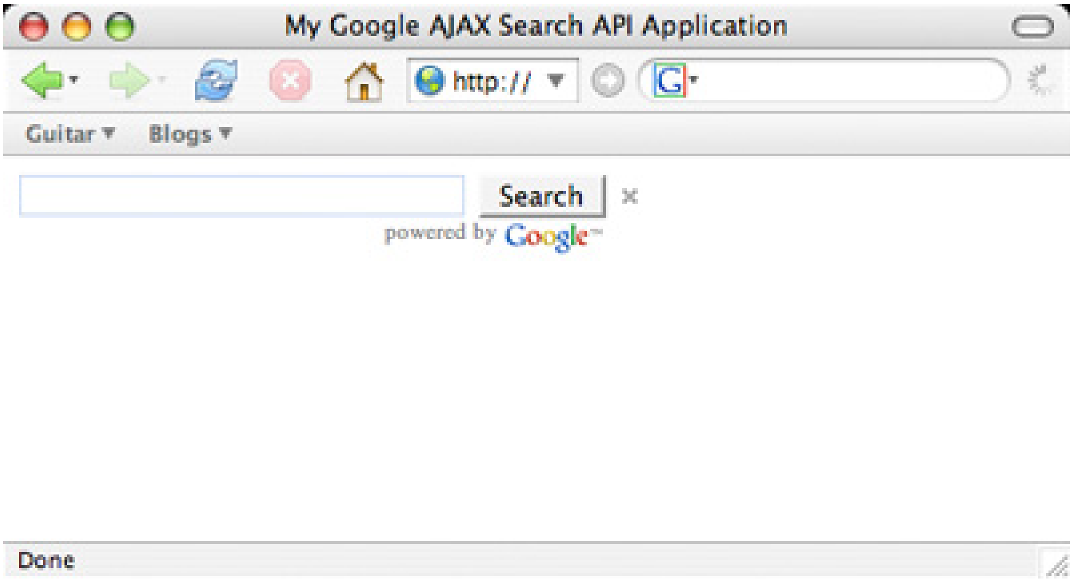
Listing 1. The HTML for a simple Google search application

```
<html>
 <head>
  <title>My Google AJAX Search API Application</title>
  <link href="http://www.google.com/uds/css/gsearch.css"
        type="text/css" rel="stylesheet" />
  <script
        src="http://www.google.com/uds/api?file=uds.js&v=1.0&key=YOUR KEY HERE"
        type="text/javascript"> </script>
  <script language="Javascript" type="text/javascript">
    function OnLoad() {
      // Create the Google search control
      var searchControl = new GSearchControl();
```

```
        // These allow you to customize what appears in the search results
        var localSearch = new GlocalSearch();
        searchControl.addSearcher(localSearch);
        searchControl.addSearcher(new GwebSearch());
        searchControl.addSearcher(new GvideoSearch());
        searchControl.addSearcher(new GblogSearch());

        // Tell Google your location to base searches around
        localSearch.setCenterPoint("Dallas, TX");

        // "Draw" the control on the HTML form
        searchControl.draw(document.getElementById("searchcontrol"));
      }
    </script>
   </head>

   <body onload="OnLoad()">
    <div id="searchcontrol" />
   </body>
 </html>
```

Be sure to replace the bolded text with your own developer key from Google. When you load the page, you'll get something that looks a lot like Figure 3.

Figure 3. The simplest search form for Google

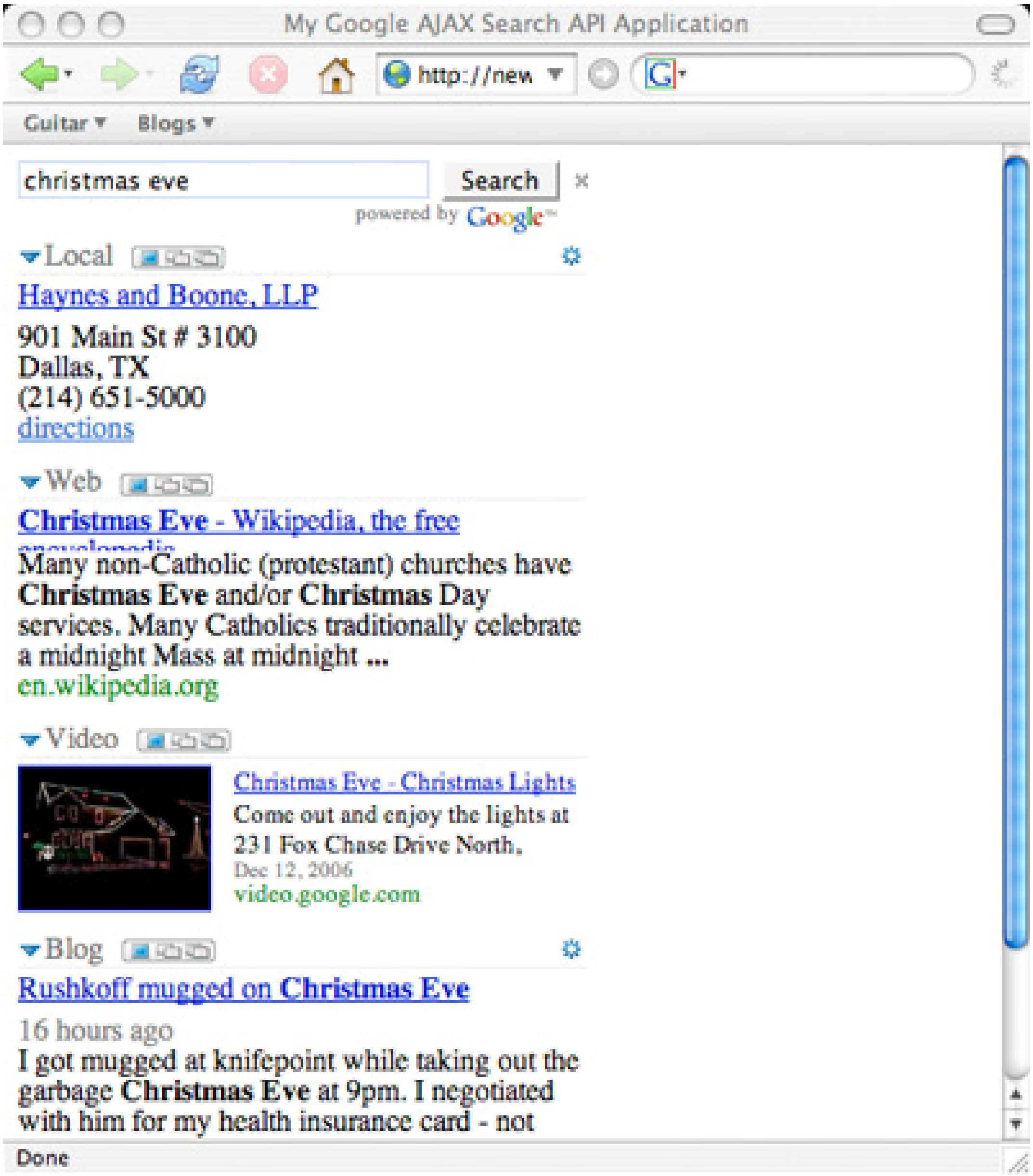


There's not much to it, but all the power of Google is sitting behind that little control.

Running a search

Type in a search term and click **Search** to send Google into action. You'll quickly see several results pop up, as shown in Figure 4.

Figure 4. Google's search results



christmas eve

Local

[Haynes and Boone, LLP]

901 Main St # 3100
Dallas, TX
(214) 651-5000
directions

Web

[Christmas Eve - Wikipedia, the free]

Many non-Catholic (protestant) churches have **Christmas Eve** and/or **Christmas** Day services. Many Catholics traditionally celebrate a midnight Mass at midnight ...
en.wikipedia.org

Video

[Christmas Eve - Christmas Lights]
Come out and enjoy the lights at 231 Fox Chase Drive North.
Dec 12, 2006
video.google.com

Blog

[Rushkoff mugged on **Christmas Eve**]

16 hours ago
I got mugged at knifepoint while taking out the garbage **Christmas Eve** at 9pm. I negotiated with him for my health insurance card - not

Adding in a pre-search

Obviously, the page looks a whole lot better once you do a search; the video, blogs, and search results give the page a nicer look. Thus, you might want to add a *pre-search* -- a search term that you define, the results of which will come up the first time a user loads your page. To do that, just add the bolded line in Listing 2 into your JavaScript.

Listing 2. Adding a pre-search term

```
    function OnLoad() {
      // Create the Google search control
      var searchControl = new GSearchControl();

      // These allow you to customize what appears in the search results
      var localSearch = new GlocalSearch();
      searchControl.addSearcher(localSearch);
      searchControl.addSearcher(new GwebSearch());
      searchControl.addSearcher(new GvideoSearch());
      searchControl.addSearcher(new GblogSearch());

      // Tell Google your location to base searches around
      localSearch.setCenterPoint("Dallas, TX");

      // "Draw" the control on the HTML form
      searchControl.draw(document.getElementById("searchcontrol"));

      searchControl.execute("Christmas Eve");
    }
```

Obviously, you can insert your own initial search term into your code to customize what comes up when the page loads.

JavaScript breakdown

Before I move on, take a quick look at what these basic commands do. First, a new `GSearchControl` is created, as illustrated in Listing 3. This is the construct that lets you do all the searching:

Listing 3. Creating a new GSearchControl

```


    function OnLoad() {
      // Create the Google search control
      var searchControl = new GSearchControl();

      ...
    }
```

Next, the code sets up a new local search, using a `GlocalSearch`; this is a special Google construct that allows you to perform a search based on a specific location. The local search is illustrated in Listing 4.

Listing 4. Setting up a new local search

```


    function OnLoad() {
      // Create the Google search control
      var searchControl = new GSearchControl();

      // These allow you to customize what appears in the search results
      var localSearch = new GlocalSearch();
      ...

      // Tell Google your location to base searches around
      localSearch.setCenterPoint("Dallas, TX");

      ...
    }
```

This is all pretty self-explanatory, once you learn the objects and methods to call. The code in Listing 4 creates a new local searcher, and then sets the center point of the search.

Next up in Listing 5 are several lines that tell the search control exactly what types of searches it should perform.

Listing 5. Search types to allow

```
function OnLoad() {
  // Create the Google search control
  var searchControl = new GSearchControl();

  // These allow you to customize what appears in the search results
  var localSearch = new GlocalSearch();
  searchControl.addSearcher(localSearch);
  searchControl.addSearcher(new GwebSearch());
  searchControl.addSearcher(new GvideoSearch());
  searchControl.addSearcher(new GblogSearch());

  // Tell Google your location to base searches around
  localSearch.setCenterPoint("Dallas, TX");

  ...
}
```

You can look up most of these search types, here's a short summary:

- `GwebSearch`: An object for searching the Web, which is what Google is best known for.
- `GvideoSearch`: This object looks for video related to your search terms.
- `GblogSearch`: This object focuses specifically on searching through blogs, which are structured and tagged a bit differently from other Web content types.

You've already seen how to pre-load a specific search. All that's left, then, is the `draw()` method call, illustrated in Listing 6. You give this call a DOM element in your HTML (for a refresher on the DOM, check out previous articles in this series in the Resources). The control will then magically appear on your form, ready for use.

Listing 6. Drawing the search control

```
function OnLoad() {
  // Create the Google search control
  var searchControl = new GSearchControl();

  // These allow you to customize what appears in the search results
  var localSearch = new GlocalSearch();
  searchControl.addSearcher(localSearch);
  searchControl.addSearcher(new GwebSearch());
  searchControl.addSearcher(new GvideoSearch());
  searchControl.addSearcher(new GblogSearch());
```

```
        // Tell Google your location to base searches around
        localSearch.setCenterPoint("Dallas, TX");

        // "Draw" the control on the HTML form
        searchControl.draw(document.getElementById("searchcontrol"));

        searchControl.execute("Christmas Eve");
      }
```

But where's the Ajax?

It's actually not readily apparent where the asynchrony happens in this simple search box. Sure, it's pretty cool that you can throw up a Google search box somewhere within your own Web application, but this is still a series on Ajax apps, not Google searches. So where's the Ajax?

Enter in a search term and click the **Search** button, and you'll notice a very Ajaxian response: the search results come up without a page reload. That's one of the hallmarks of most Ajax apps -- the visible content changing without reloading pages. Clearly, something is going on beyond the normal request/response model. But where is `XMLHttpRequest`? Where is the `request` object that you've learned about for so many articles? And other than that one `getElementById()` method, where is the DOM and page manipulation? Well, it's all wrapped up in two lines within your HTML.

Google takes care of the JavaScript

The first line of note is one I haven't talked much about yet, and it's illustrated in Listing 7.

Listing 7. The all-important JavaScript file

```
<script
    src="http://www.google.com/uds/api?file=uds.js&v=1.0&key=[YOUR GOOGLE KEY]"
    type="text/javascript"> </script>
```

The syntax here isn't particularly interesting, but the point is that Google hosts a file named uds.js, which has within it all the JavaScript needed for the search box to operate. This is using someone else's code in the truest sense: you're even letting the third party host the code your application uses. That's pretty significant, because Google handles upkeep and maintenance, and when the company upgrades the JavaScript file, you get the benefits automatically. Google won't change the API without letting you know, so your code will keep working even if the JavaScript file is changed.

The GSearchControl object

The other thing somewhat hidden from view is the code for the `GSearchControl` object, created in the `onLoad()` JavaScript function. All you have to do to create this object is invoke the code in Listing 8.

Listing 8. Creating a GSearchControl object

```
// Create the Google search control
var searchControl = new GSearchControl();
```

The necessary HTML is also pretty bland: just a `div` tag with an ID that your JavaScript can refer to, as shown in Listing 9.

Listing 9. All the HTML you'll need to create a search control

```
<div id="searchcontrol" />
```

What does Google's JavaScript look like?

Google's JavaScript isn't so easy to get at. First, the uds.js JavaScript file figures out some locality settings, handles some Google-specific tasks, verifies your Google key, and then loads two *other* scripts: http://www.google.com/uds/js/locale/en/uistrings.js and http://www.google.com/uds/js/uds_compiled.js. You can actually pull up these two files and wade through them if you're interested, but beware: it's tough going through advanced code, and the formatting is awful! For most of you, simply knowing how to use these files should be your goal, rather than worrying too much about what they do at a line-by-line level.

But, again, behind the scenes, Google's code is doing all sorts of things. It's creating a new text box, some iconic graphics, and a button that calls a JavaScript function, also in Google's JavaScript. So you get all sorts of behavior for free; while you should understand the basics of how this happens, it's convenient that you can simply use this code, and get on with the rest of your application.

Ajax is more than just the code you write

So Ajax apps aren't just about you using `XmlHttpRequest`; they're about a way to approach Web applications, and are based upon asynchrony. Even though you didn't write any Ajax-specific code, you have created an Ajax app. Thank Google: it did most of the work, and you get the benefit!

Going further with Google's Ajax Search API

At this point, it's up to you to take these steps and adapt them to your own applications. At the simplest end of the spectrum, you can drop a `div` into your own

Web pages and add the JavaScript shown in Listing 1 to your Web page; you'll have Google search all ready for use.

But the fun doesn't have to stop there. You don't need to limit yourself to this specific set of options or controls. Play around with the Web results, the blog results, and the video results, and integrate each into your Web applications where appropriate. You might want to offer multiple search controls, each of them focusing on a specific type of result. You might want to include the Google search control in a `span` element, right in the middle of the rest of your app's content, rather than off to the side in a `div`. Whatever the case, you really should ensure that Google's search is molded to fit your needs, rather than modifying your applications to suit Google.

In conclusion

It shouldn't be hard to build on what you've learned here and start to add Google search boxes and other uses of the Google API to your own Ajax apps. More importantly, though, you should now have a pretty good idea of how to use public APIs in general. For example, Amazon.com offers a public API that allows you to perform the same kind of Web searches you performed with Google on books and other merchandise from Amazon's store. You can start to hunt around for your own favorite public APIs, and move beyond just what your own coding skills allow for. In fact, you can very easily create a site that integrates with Google, Amazon.com, Flickr, and much more.

So while it's important to get a handle on how to use Google -- because of the search algorithms and enormous data stores that Google offers -- it's even *more* important to learn how to use *any* public API. You should also start thinking about your application as more than a sum of your coding skills; instead, it can simply be a gateway to various pieces of data. That data might be stored on the servers of Google, Amazon.com, del.icio.us, or anywhere else. Add to that your own business's or project's take on that data, and you've got a very powerful and robust solution that goes well beyond what you could code on your own.

So go out and make *big* applications. Use data from all sorts of places, and don't be confined by what you've coded yourself. Enjoy, and I'll come back to some more technical issues, like data formats, in the next article of this series.

Resources

Learn

- Mastering Ajax: Read the previous articles in this series.
- The open source definition: Get the scoop from the Open Source Initiative.
- Google Ajax Search API Developer Guide: With these docs, walk through every aspect of using your Google key and the Ajax search API.
- XML technical library: See the developerWorks XML Zone for a wide range of technical articles and tips, tutorials, standards, and IBM Redbooks.
- developerWorks Web development zone: Expand your site development skills with articles and tutorials that specialize in Web 2.0, Ajax, wikis, PHP, mashups, and other Web projects
- developerWorks Open source zone: Find resources for open source development and implementation.
- xml.com: One of the easiest-to-understand online resources for everything XML. Start there if you're not already an experienced XML programmer.
- Build dynamic Java applications (Philip McCarthy, developerWorks, September 2005): Take a look at Ajax from the server side, using a Java perspective.
- Java object serialization for Ajax (Philip McCarthy, developerWorks, October 2005): Examine how to send objects over the network, and interact with Ajax, from a Java perspective.
- Call SOAP Web services with Ajax (James Snell, developerWorks, October 2005): Dig into a fairly advanced article on integrating Ajax with existing SOAP-based Web services.
- The DOM Home Page at the World Wide Web Consortium: Visit the starting place for all things DOM-related.
- The DOM Level 3 Core Specification: Get the definitions for the core Document Object Model, from the available types and properties to the usage of the DOM from various languages.
- The ECMAScript language bindings for DOM: Explore this spec, of great interest to any JavaScript programmers wanting to use the DOM from their code.
- Ajax: A new approach to Web applications (Jesse James Garrett, *Adaptive Path,* February 2005): Read the article that coined the Ajax moniker and considered required reading for all Ajax developers.

Get products and technologies

- Google Ajax Search API home page: Start the process of adding Google searches to your Ajax apps here.
- *Head Rush Ajax,* Brett McLaughlin (O'Reilly Media, 2006): Take the ideas in this article and loads them into your brain, Head First style.
- *Java and XML, Second Edition,* Brett McLaughlin (O'Reilly Media, Inc., 2001): Check out the author's discussion of XHTML and XML transformations.
- *JavaScript: The Definitive Guide,* David Flanagan (O'Reilly Media, Inc., 2001): Dig into extensive instruction on working with JavaScript and dynamic Web pages. The upcoming edition adds two chapters on Ajax.
- *Head First HTML with CSS & XHTML,* Elizabeth and Eric Freeman, O'Reilly Media, Inc., 2005): Learn more about standardized HTML and XHTML, and how CSS can be applied to HTML.
- IBM trial software: Build your next development project with software available for download directly from developerWorks.

Discuss

- developerWorks blogs: Get involved in the developerWorks community.

About the author

Brett McLaughlin

Brett McLaughlin has worked in computers since the Logo days. (Remember the little triangle?) In recent years, he's become one of the most well-known authors and programmers in the Java and XML communities. He's worked for Nextel Communications, implementing complex enterprise systems; at Lutris Technologies, actually writing application servers; and most recently at O'Reilly Media, Inc., where he continues to write and edit books that matter. Brett's upcoming book, *Head Rush Ajax*, brings the award-winning and innovative Head First approach to Ajax. His last book, *Java 1.5 Tiger: A Developer's Notebook*, was the first book available on the newest version of Java technology. And his classic *Java and XML* remains one of the definitive works on using XML technologies in the Java language.