A Guide to Public Data

Data Source Handbook

O'REILLY®

Pete Warden

Data Source Handbook

Pete Warden

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Data Source Handbook
by Pete Warden

Copyright © 2011 Pete Warden. 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 (*http://my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editor: Mike Loukides
Production Editor: Teresa Elsey
Proofreader: Teresa Elsey
Cover Designer: Karen Montgomery
Interior Designer: David Futato
Illustrator: Robert Romano

Printing History:
February 2011: First Edition.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Data Source Handbook,* the image of a common kite, 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.

ISBN: 978-1-449-30314-3

[LSI]

1295970672

Table of Contents

Preface ..... **vii**

Data Source Handbook **1**
Websites 1
WHOIS 1
Blekko 2
bit.ly 3
Compete 3
Delicious 4
BackType 5
PagePeeker 5
People by Email 5
WebFinger 6
Flickr 6
Gravatar 6
Amazon 7
AIM 7
FriendFeed 8
Google Social Graph 8
MySpace 9
Github 10
Rapleaf 10
Jigsaw 11
People by Name 11
WhitePages 11
LinkedIn 11
GenderFromName 11
People by Account 12
Klout 12
Qwerly 12
Search Terms 12
BOSS 13

Download from Wow! eBook <www.wowebook.com>

Blekko 13
Bing 14
Google Custom Search 14
Wikipedia 15
Google Suggest 15
Wolfram Alpha 16
Locations 16
SimpleGeo 17
Yahoo! 18
Google Geocoding API 18
CityGrid 19
Geocoder.us 19
Geodict 19
GeoNames 20
US Census 20
Zillow Neighborhoods 21
Natural Earth 22
US National Weather Service 23
OpenStreetMap 24
MaxMind 24
Companies 24
CrunchBase 24
ZoomInfo 25
Hoover's 25
Yahoo! Finance 26
IP Addresses 26
MaxMind 26
Infochimps 27
Books, Films, Music, and Products 27
Amazon 27
Google Shopping 27
Google Book Search 28
Netflix 28
Yahoo! Music 29
Musicbrainz 29
The Movie DB 29
Freebase 30

Data Source Handbook

Websites

WHOIS

The `whois` Unix command is still a workhorse, and I've found the web service a decent alternative, too. You can get the basic registration information for any website. In recent years, some owners have chosen "private" registration, which hides their details from view, but in many cases you'll see a name, address, email, and phone number for the person who registered the site. You can also enter numerical IP addresses here and get data on the organization or individual that owns that server.

Unfortunately the terms of service of most providers forbid automated gathering and processing of this information, but you can craft links to the Domain Tools site to make it easy for your users to access the information:

```
<a href="http://whois.domaintools.com/www.google.com">Info for www.google.com</a>
```

There is a commercial API available through whoisxmlapi.com that offers a JSON interface and bulk downloads, which seems to contradict the terms mentioned in most WHOIS results. It costs $15 per thousand queries. Be careful, though; it requires you to send your password as a nonsecure URL parameter, so don't use a valuable one:

```
curl "http://www.whoisxmlapi.com/whoisserver/WhoisService?\
domainName=oreilly.com&outputFormat=json&userName=<username>&password=<password>"
{"WhoisRecord": {
  "createdDate": "26-May-97",
  "updatedDate": "26-May-10",
  "expiresDate": "25-May-11",
  "registrant": {
    "city": "Sebastopol",
    "state": "California",
    "postalCode": "95472",
    "country": "United States",
    "rawText": "O'Reilly Media, Inc.\u000a1005 Gravenstein Highway North
 \u000aSebastopol, California 95472\u000aUnited States\u000a",
```

bit.ly

The bit.ly API lets you access analytics information for a URL that's been shortened. If you're starting off with a full URL, you'll need to call the lookup function to obtain the short URL. You can sign up for API access here. This is most useful if you want to gauge the popularity of a site, either so you can sort and filter links you're displaying to a user or to feed into your own analysis algorithms:

```
curl "http://api.bit.ly/v3/clicks?login=<login>&apiKey=<key>&\
shortUrl=http://bit.ly/hnB7HI"

{"status_code": 200, "data": {
  "clicks": [{
    "short_url": "http://bit.ly/hnB7HI",
    "global_hash": "gKGd7s",
    "user_clicks": 9,
    "user_hash": "hnB7HI",
    "global_clicks": 36}]},
  "status_txt": "OK"
}
```

Compete

The Compete API gives a very limited amount of information on domains, a trust rating, a ranking for how much traffic a site receives, and any online coupons associated with the site. Unfortunately, you don't get the full traffic history information that powers the popular graphs on the web interface. The terms of service also rate-limit you to 1,000 calls a day, and you can't retain any record of the information you pull, which limits its usefulness:

```
curl "http://api.compete.com/fast-cgi/MI?d=google.com&ver=3&apikey=<key>&size=large"

<ci>
        <dmn>
                <nm>google.com</nm>
                <trust caption="Trust">
                        <val>green</val>
                        <link>http://toolbar.compete.com/trustgreen/google.com</link>
                <icon>...</icon>
                </trust>
                <rank caption="Profile">
                        <val>1</val>
                        <link>http://toolbar.compete.com/siteprofile/google.com</link>
                    <icon>...</icon>
                </rank>
...
```

BackType

BackType keeps track of the public conversations associated with a web page and offers an API to retrieve them from your own service. The service rate-limits to 1,000 calls a day, but from talking to BackType, it seems they're keen to help if you want higher usage.

The information is usually used to display related conversations in a web interface, but, with a bit of imagination, you could use it to identify users related to a particular topic or gauge the popularity of a page instead:

```
curl "http://api.backtype.com/connect.json?\
url=http://www.techcrunch.com/2009/03/30/if-bitly-is-worth-8-million-tinyurl-is\
-worth-at-least-46-million/&key=0cd9bd64b6dc4e4186b9"

{"startindex":1,"itemsperpage":25,"next_page":2,"comments":[{"comment":
{"id":"000032ca7e8b26f9d79b549cb451b518",
  "url":"http:\/\/blog.saush.com\/2009\/04\/13\/
clone-tinyurl-in-40-lines-of-ruby-code\/#comment-1476",
  "content":"...",
  "date":"2010-12-06 17:10:53"},
  "blog":{"id":13002,
    "url":"http:\/\/blog.saush.com\/","title":"saush.com"},
  "post":{"url":"http:\/\/blog.saush.com\/2009\/04\/13\/
clone-tinyurl-in-40-lines-of-ruby-code\/",
    "title":"Clone TinyURL in 40 lines of Ruby code"},
  "author":{"name":"Cpchhukout",
 "url":"http:\/\/newwave.hoha.ru\/maxim_axenov?ref=wmbasta"},
...
```

PagePeeker

If you're displaying a lot of URLs to your users, it can be handy to give them visual cues. This simple web service gives you an easy way to do that by embedding HTML images of site favicons:

```
<img src="http://pagepeeker.com/f/wikipedia.org" border="0" width="16px"
 height="16px">
```

People by Email

These services let you find information about users on their systems using an email address as a search term. Since it's common to have email addresses for your own users, it's often possible to fetch additional data on them from their other public profiles. For example, if you retrieve a location, real name, portrait, or description from an external service, you can use it to prepopulate your own "create a profile" page. You can find open source code examples demonstrating how to use most of these APIs at *http://github.com/petewarden/findbyemail*, and there's a live web demo at *http://web.mailana.com/labs/findbyemail/*.

people have signed up, so it can be a good way of providing at least default avatars for your own users. You could also save yourself some coding by directing new users to Gravatar's portrait creation interface. There's also a profile lookup API available, but I haven't had any experience with how well-populated this is:

```
md5 -s pete@mailana.com
MD5 ("pete@mailana.com") = 03e801b74b01f23957a3afdd9aaaed00

<img src="http://www.gravatar.com/avatar/03e801b74b01f23957a3afdd9aaaed00" />
```

Figure 3. Gravatar portrait image

Amazon

Like Yahoo!, Amazon doesn't expose very much information about each user when you look up an email address, but you can often get at least a location. The sheer size of Amazon's user base means that you'll find information on a large percentage of emails. There's also the chance to discover public wishlists, which could be helpful for creating default interests for your new users' profiles.

The API is REST/XML-based, but it does require a somewhat complex URL signing scheme for authentication.

AIM

You can look up an AOL Instant Messenger account from an email address, and you get a portrait image and username back. The exact information returned depends on whether the user is online, and you'll only get a default image if he or she is away. The service uses a REST/JSON API, and it requires a sign up to access:

```
curl "http://api.oscar.aol.com/presence/get?f=json&k=<key>&\
t=petewarden%40aol.com&emailLookup=1&notFound=1"

{"response":{"statusCode":200, "statusText":"Ok", "data":{"users":[{
  "emailId":"petewarden@aol.com",
  "aimId":"petewarden",
  "displayId":"petewarden",
  "state":"offline",
  "userType":"aim",
  "presenceIcon":"http://o.aolcdn.com/aim/img/offline.gif"
}]}}}
```

```
curl "http://socialgraph.apis.google.com/lookup?\
q=mailto%3asearchbrowser%40gmail.com&fme=1&edi=1&edo=1&pretty=1&sgn=1&callback="

{ "canonical_mapping": {
  "mailto:searchbrowser@gmail.com": "sgn://mailto/?pk\u003dsearchbrowser@gmail.com"
 },
 "nodes": {
  "sgn://mailto/?pk\u003dsearchbrowser@gmail.com": {
   "attributes": {
   },
   "claimed_nodes": [
   ],
   "unverified_claiming_nodes": [
    "sgn://typepad.com/?ident\u003dpetewarden"
   ],
   "nodes_referenced": {
   },
   "nodes_referenced_by": {
    "sgn://typepad.com/?ident\u003dpetewarden": {
     "types": [
      "me"
     ]
    }
   }
  }
 }
}
```

MySpace

The early social network still holds information on a lot of people, and it exposes a surprisingly large amount, including things like age and gender. This could come in handy if you need to do a demographic analysis of your user base, though with the lack of activity on the site, the information will become less useful as time goes by. You can use the API without any authentication:

```
curl "http://api.myspace.com/opensearch/people?searchBy=email&\
searchTerms=bill%40example.com"

{"startIndex":"1","itemsPerPage":"10","totalResults":"2",
 "resultCount":"2","searchId":"34848869-de3b-415a-81ab-5df0b1ed82eb","entry":[{
  "id":"myspace.com.person.3430419",
  "displayName":"bill",
  "profileUrl":"http:\/\/www.myspace.com\/3430419",
  "thumbnailUrl":"http:\/\/x.myspacecdn.com\/images\/no_pic.gif",
  "msUserType":"RegularUser",
  "gender":"Female",
  "age":"31",
  "location":"",
  "updated":"12\/12\/2010 6:49:11 PM",
  "isOfficial":"0"},{
  "id":"myspace.com.person.146209268",
  "displayName":"Andy",
  "profileUrl":"http:\/\/www.myspace.com\/146209268",
```

```
  "thumbnailUrl":"http:\/\/x.myspacecdn.com\/images\/no_pic.gif",
  "msUserType":"RegularUser",
  "gender":"Male",
  "age":"34",
  "location":"",
  "updated":"3\/26\/2010 1:14:00 PM",
  "isOfficial":"0"}]}
```


Github

If you're targeting people who are likely to be developers, there's a good chance they'll have github accounts, and if they've opted-in to being found by email address, you'll be able to pull up their public details. The API doesn't require authorization, or even registration, and it gives you information on users' companies, real names, locations, and any linked sites, like blogs:

```
curl "http://github.com/api/v2/xml/user/email/pete%40petewarden.com"

<?xml version="1.0" encoding="UTF-8"?>
<user>
  <gravatar-id>9cbf603d5f93133178367214f1e091b9</gravatar-id>
  <company>Mailana Inc</company>
  <name>Pete Warden</name>
  <created-at type="datetime">2009-12-03T08:29:50-08:00</created-at>
  <location>Boulder, CO</location>
  <public-repo-count type="integer">26</public-repo-count>
  <public-gist-count type="integer">0</public-gist-count>
  <blog>http://petewarden.typepad.com/</blog>
  <following-count type="integer">0</following-count>
  <id type="integer">161459</id>
  <type>User</type>
  <permission nil="true"></permission>
  <followers-count type="integer">58</followers-count>
  <login>petewarden</login>
  <email>pete@petewarden.com</email>
</user>
```

Rapleaf

Originally, Rapleaf's API returned information about a person's social networking accounts if you supplied an email, but it has recently switched to offering demographic data on age, gender, income, and address instead. The FindByEmail code still uses the old V2 API. Since the service gathers data without any user involvement (though it does operate an opt out system), it's been controversial.

algorithms to handle a lot of variants and nicknames. Nothing like this will be 100 percent accurate, but it's great for applications like demographic analysis where occasional errors don't matter:

```
require_once('genderfromname.php');

print gender("Jon");    // prints 'm'
```

People by Account

Klout

Klout's API will give you an influence score for a given Twitter username. You can then use this information to help prioritize Twitter accounts within your own service (for example, by highlighting links shared by people with higher reputation or spam filtering those with low scores):

```
http://api.klout.com/1/klout.xml?key=[your_api_key]&users=[usernames]
```

Qwerly

This service allows you to link Twitter usernames with accounts on other sites. Unfortunately, the data is still pretty sparse, and the Facebook account lookup doesn't return any useful information, but it's still worth a look:

```
curl "http://api.qwerly.com/v1/twitter/petewarden.json?api_key=<key>"

{ "location":"Boulder, CO",
  "name":"Pete Warden",
  "twitter_username":"petewarden",
  "qwerly_username":null,
  "services":[
    {"type":"github","url":"http://github.com/petewarden","username":"petewarden"},
    {"type":"twitter","url":"http://twitter.com/petewarden","username":"petewarden"},
    {"type":"klout","url":"http://klout.com/petewarden","username":"petewarden"}
]}
```

Search Terms

Sometimes you're trying to match a word or phrase with some web pages within your service, either for traditional user-driven search or as part of a backend analysis process. The biggest downside of most of the APIs is usually their restrictive terms of service, especially if you're doing further processing with the results instead of showing them directly to users, so make sure you read the fine print. You can find PHP example code for Bing, BOSS, and Google on my blog.

```
          "short_host" : "herbalremedies.com",
          "url" :
 "http://www.herbalremedies.com/the-bible-cure-for-headaches-don-colbert.html"
      },
...
```

To obtain an API key, email *apiauth@blekko.com*. The terms of service are somewhat restrictive, but the service is small and hungry enough to be flexible in practice (at least until it becomes large and well fed).

Bing

Microsoft offers quite a comprehensive set of search APIs for standard web results, along with images, news, and even local businesses. Though the terms of service make it clear the service is intended only for end-user-facing websites, the lack of rate limits is very welcome. You'll need to obtain an API key before you can use the API:

```
curl "http://api.bing.net/json.aspx?AppId=&Query=pete+warden&Sources=Web"

{"SearchResponse":{
  "Version":"2.2",
  "Query":{"SearchTerms":"pete warden"},
  "Web":{
    "Total":276000,"Offset":0,"Results":[
      {"Title":"Pete Warden",
      "Description":"I've had reports of problems running these with the latest
 After Effects CS3. I'm not working with AE at the moment, so I haven't been able to
 investigate and fix the problems.",
      "Url":"http:\/\/petewarden.com\/",
...
```

Google Custom Search

As the king of search, Google doesn't have much of an incentive to open up its data to external developers...and it shows. Google killed off the Ajax Search API that allowed access to the same results as the web interface and replaced it with the more restrictive Custom Search version. You'll need to sign up to get access, and you start with a default of only 100 queries per day, with any additional calls requiring approval from the company. You can also only search a specific slice of the Web, which you'll need to specify up front:

```
curl "https://www.googleapis.com/customsearch/v1?\
key=<key>&cx=017576662512468239146:omuauf_lfve&alt=json&\
q=pete%20warden&prettyprint=true"

{"kind": "customsearch#search",
 "url": {
  "type": "application/json",
...
 "items": [
  {
```

```
 <suggestion data="san francisco is the best city in the world"/>
 <num_queries int="63800000"/></CompleteSuggestion>
 <CompleteSuggestion><suggestion data="san francisco is gay"/>
 <num_queries int="24100000"/></CompleteSuggestion>
 <CompleteSuggestion><suggestion data="san francisco is burning"/>
 <num_queries int="11200000"/></CompleteSuggestion>
 <CompleteSuggestion><suggestion data="san francisco is overrated"/>
 <num_queries int="409000"/></CompleteSuggestion>
</toplevel>
```

Wolfram Alpha

The Wolfram Alpha platform pulls together a very broad range of facts and figures on everything from chemistry to finance. The REST API takes in some search terms as input, and returns an XML document containing the results. The output is a series of sections called *pods*, each containing text and images ready to display to users. Unfortunately there's no easy way to get a machine-readable version of this information, so you can't do further processing on the data within your application. It's still a rich source of supplemental data to add into your own search results, though, which is how Bing is using the service.

If you're a noncommercial user, you can make up to 2,000 queries a month for free, and you can experiment with the interactive API console if you want to explore the service. The commercial rates range between two and six cents a call, depending on the volume. The terms of use prohibit any caching of the data returned from the service and you'll need to sign up for a key to access it:

```
curl "http://api.wolframalpha.com/v2/query?\
appid=<key>input=General%20Electric&format=image,plaintext,cell,minput"

<?xml version='1.0' encoding='UTF-8'?>
<queryresult success='true'
    error='false'
...
 <pod title='Latest trade'
     scanner='FinancialData'
     id='Quote'
     position='200'
     error='false'
     numsubpods='1'>
  <subpod title=''>
   <plaintext>$19.72(GE | NYSE | Friday 1:00:18 pm PST | 27 hrs ago)</plaintext>
...
```

Locations

Geographic information is such a wide field that it probably deserves its own guide, but here I'm going to focus on the most useful and accessible data sources I've found. All of these take some kind of geographic location, either a place name, an address, or latitude/longitude coordinates, and return additional information about that area.

Yahoo!

Yahoo! has been a surprising leader in online geo APIs, with Placefinder for converting addresses or place names into coordinates, GeoPlanet for getting category and neighborhood information about places, and Placemaker for analyzing text documents and extracting words or phrases that represent locations. You'll need to sign up for an app ID, but after that it's a simple REST/JSON interface.

You can also download a complete list of the locations that Yahoo has in its database, holding their names and the WOEID identifier for each. This can be a useful resource for doing offline processing, though it is a bit hobbled by the lack of any coordinates for the locations:

```
curl "http://where.yahooapis.com/geocode?\
q=1600+Pennsylvania+Avenue,+Washington,+DC&appid=<App ID>&flags=J"

{"ResultSet":{"version":"1.0","Error":0,"ErrorMessage":"No error","Locale":"us_US",
"Quality":87,"Found":1,"Results":[{
  "quality":85,
  "latitude":"38.898717","longitude":"-77.035974",
  "offsetlat":"38.898590","offsetlon":"-77.035971",
  "radius":500,"name":"","line1":"1600 Pennsylvania Ave NW",
  "line2":"Washington, DC  20006","line3":"",
  "line4":"United States","house":"1600",
  "street":"Pennsylvania Ave NW",
...
```

Google Geocoding API

You can only use this geocoding API if you're going to display the results on a Google Map, which severely limits its usefulness. There's also a default limit of 2,500 requests per day, though commercial customers get up to 100,000. It doesn't require any key or authentication, and it also supports "reverse geocoding," where you supply a latitude and longitude and get back nearby addresses:

```
curl "http://maps.googleapis.com/maps/api/geocode/json?\
address=1600+Amphitheatre+Parkway,+Mountain+View,+CA&sensor=false"

{ "status": "OK",
  "results": [ {
    "types": [ "street_address" ],
    "formatted_address": "1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA",
    "address_components": [ {
      "long_name": "1600",
      "short_name": "1600",
      "types": [ "street_number" ]
    }, {
      "long_name": "Amphitheatre Pkwy",
      "short_name": "Amphitheatre Pkwy",
      "types": [ "route" ]
    }, {
...
```

CityGrid

With listings for eighteen million US businesses, this local search engine offers an API to find companies that are near a particular location. You can pass in a general type of business or a particular name and either latitude/longitude coordinates or a place name. The service offers a REST/JSON interface that requires a sign up, and the terms of service and usage requirements restrict the service to user-facing applications. CityGrid does offer an unusual ad-driven revenue sharing option, though, if you meet the criteria:

```
curl "http://api.citygridmedia.com/content/places/v2/search/where?\
where=94117&what=bakery&format=json&publisher=<publisher>&api_key=<key>"

{"results":{"query_id":null,
...
"locations":[{"id":904051,"featured":false,"name":"Blue Front Cafe",
  "address":{"street":"1430 Haight St",
 "city":"San Francisco","state":"CA","postal_code":"94117"},
...
```

Geocoder.us

The Geocoder.us website offers a commercial API for converting US addresses into location coordinates. It costs $50 for 20,000 lookups, but thankfully, Geocoder has also open-sourced the code as a Perl CPAN module. It's straightforward to install, but the tricky part is populating it with data, since it relies on Tiger/Line data from the US Census. You'll need to hunt around on the Census website to locate the files you need, and then they're a multigigabyte download.

Geodict

An open source library similar to Yahoo!'s Placemaker API, my project takes in a text string and extracts country, city, and state names from it, along with their coordinates. It's designed to run locally, and it only spots words that are highly likely to represent place names. For example, Yahoo! will flag the "New York" in "New York Times" as a location, whereas Geodict requires a state name to follow it or a location word like *in* or *at* to precede it:

```
./geodict.py -f json < testinput.txt

[{"found_tokens": [{
  "code": "ES", "matched_string": "Spain",
  "lon": -4.0, "end_index": 4, "lat": 40.0,
  "type": "COUNTRY", "start_index": 0}]},
...
```

Next, select the statistics that you're interested in (Figure 5).

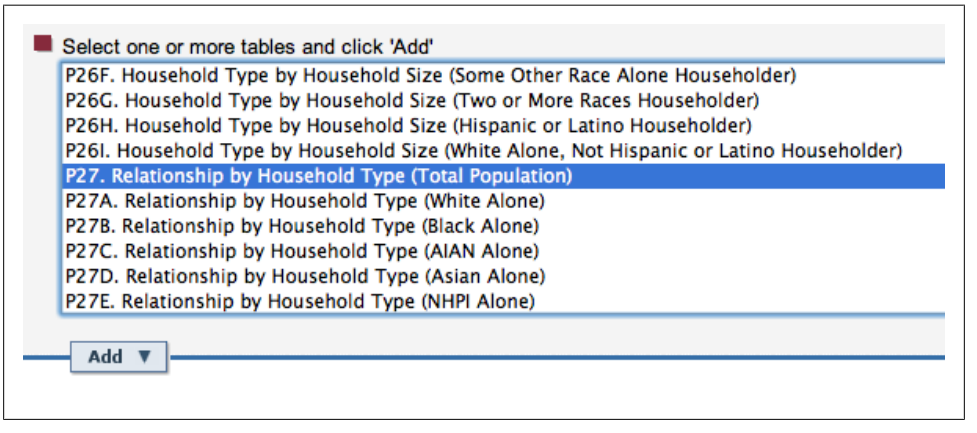

Figure 5. US Census site

Then you'll be able to download that data as a text file. The format is a bit odd, a table with columns separated by "|" (pipe) characters, but with a bit of find-and-replace magic, you can convert them into standard CSV files readable by any spreadsheet.

To ensure privacy, the Census does rearrange its very detailed data in some cases, without invalidating its statistical significance. For example, if there's only one Iranian doctor in Boulder, Colorado, that person may be included in a neighboring city's data instead and swapped with an equivalent person's information, so the overall statistics on income, etc. are unaffected.

There's also a wealth of shape data available, giving detailed coordinates for the boundaries of US counties, states, cities, and congressional districts.

Zillow Neighborhoods

The only US boundary that the Census doesn't offer is neighborhoods. Thankfully, the Zillow real estate site has made its neighborhood data available as Creative Commons–licensed downloads (Figure 6).

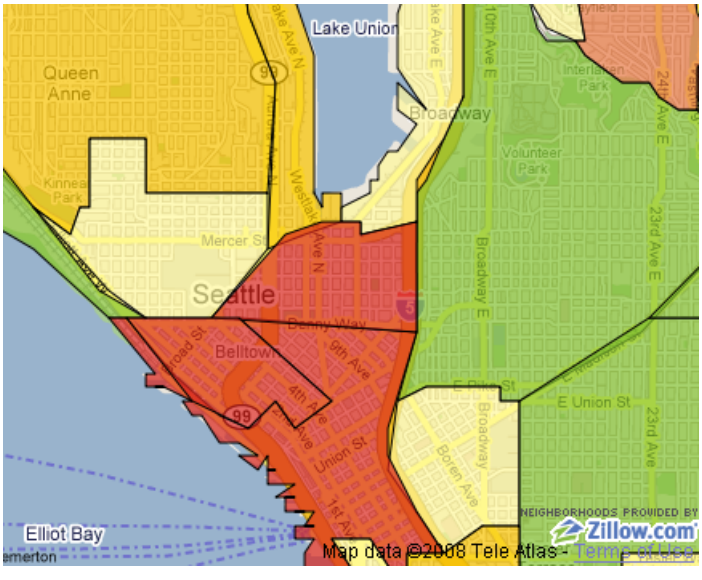

Figure 6. Zillow neighborhood data

Natural Earth

Natural Earth offers a very clean, accurate, and detailed public domain collection of country and province boundaries for the entire planet, available for download in bulk (Figure 7). You'll need some geo knowledge to convert them into a usable form, but it's not too much of a challenge. For example, you could load the shapefiles into PostGIS and then easily run reverse geo-code queries to discover which country and state a point lies within.

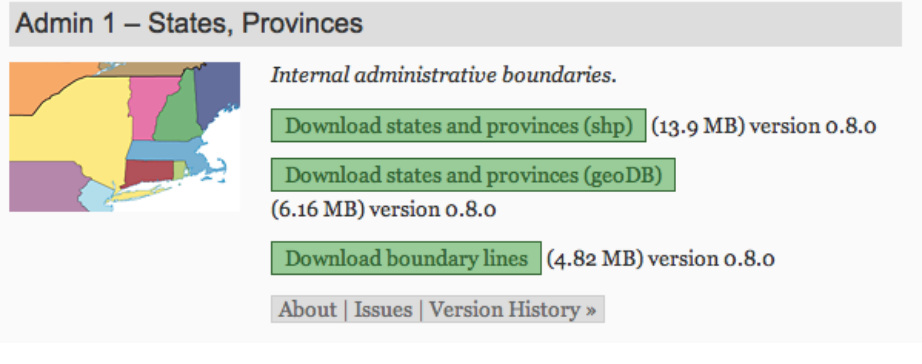

Figure 7. Natural Earth data

```
        <value>20</value>
      </temperature>
    </parameters>
  </data>
</dwml>
```

OpenStreetMap

The volunteers at OpenStreetMap have created a somewhat-chaotic but comprehensive set of geographic information, and you can download everything they've gathered as a single massive file. One unique strength is the coverage of areas in the developing world that are absent from commercial databases, and since it's so easy to change, even US locations are often more up-to-date with recent changes than more traditional maps. The downside of the system is that it's designed for navigation, not analysis, so a lot of information about administrative boundaries and other nonphysical attributes is missing. The Nominatim project attempts to organize the data into a form you can use to look up street addresses, but the lack of good coverage of things like postal codes limits its usefulness. Reconstructing some structure from the soup of roads and points is also computationally very taxing, easily taking a couple weeks of computation time on a high-end machine.

If you're only working in a more limited geographic region, you may want to look at the Cloudmade extracts, which contain subsets for different areas and attributes.

MaxMind

This is one of the simplest but most useful data sets for geographic applications. It's a CSV file containing information on 2.7 million cities and towns around the world. It has the latitude and longitude coordinates, region, country, and alternate names for all of them, and the population for many thanks to Stefan Helder's data. You can just load this file into your favorite database, index by the key you want to query on, and you've got a perfectly workable local service for working with addresses and other locations:

```
...
us,new woodstock,New Woodstock,NY,,42.8483333,-75.8547222
us,new york,New York,NY,8107916,40.7141667,-74.0063889
us,new york mills,New York Mills,NY,,43.1052778,-75.2916667
us,newark,Newark,NY,9365,43.0466667,-77.0955556
...
```

Companies

CrunchBase

TechCrunch has accumulated information on more than 50,000 companies and 70,000 people and has made it available both through a web interface and a simple REST/

JSON API. You don't need to authenticate, and it's all under a Creative Commons license. There aren't any official bulk downloads available, but I created the CrunchCrawl project to use the API to pull down information about all companies in the system.

To get information about a particular company, you'll first need to search on its name to find the CrunchBase ID; then you can access the full set of data. As you might imagine, the database has excellent coverage of technology companies but very little on other industries:

```
curl "http://api.crunchbase.com/v/1/company/facebook.js"

{"name": "Facebook",
 "permalink": "facebook",
 "crunchbase_url": "http://www.crunchbase.com/company/facebook",
 "homepage_url": "http://facebook.com",
 "blog_url": "http://blog.facebook.com",
 "blog_feed_url": "http://blog.facebook.com/atom.php",
 "twitter_username": "facebook",
...
```

ZoomInfo

Covering around four million US companies, ZoomInfo has broader coverage and is less tech-centric than CrunchBase. It offers a REST interface with XML output, with only 2,000 queries a day permitted, and terms of service that require branding on any services you create using it:

```
curl "http://api.zoominfo.com/PartnerAPI/XmlOutput.aspx?\
query_type=company_search_query&pc=<key>&companyName=Facebook"

<?xml version="1.0" encoding="utf-8"?>
...
<TotalResults>84</TotalResults><CompanySearchResults>
<CompanyRecord><CompanyID>32394811</CompanyID>
<ZoomCompanyUrl>http://www.zoominfo.com/Search/CompanyDetail.aspx?
CompanyID=32394811&amp;cs=QEB7k47oc&amp;pc=publicapi
</ZoomCompanyUrl>
<CompanyName>Facebook</CompanyName>
<CompanyTicker/>
<Website>www.facebook.com</Website>
<CompanyAddress><City>Palo Alto</City><State>CA</State><CountryCode>USA</CountryCode>
```

Hoover's

With the most comprehensive database of company information, the Hoover's API is heavily geared toward enterprise-level users. Hoover's does offer a REST/JSON interface, somewhat strangely using HTTP Basic Authentication headers for the key information. You only get three months of free access to the API (for testing purposes) before you have to talk to the sales staff about your application, but at least you can sign up

initially online with no wait. To get information about a company, do a search to retrieve the company ID within the service. Then you can pull down address, revenue, and employee count, along with lots of other statistics.

Yahoo! Finance

For many years, Yahoo! has let you look up trading information for companies based on their stock tickers. This isn't an official API, so it may be removed or altered at any point, and the terms under which you can use the information are unclear. I'm including it despite those warnings, because it does make some unique information available. Unusually, it returns its results as a CSV file, which is great if you want to run it through further spreadsheet processing, but which will require a little parsing work from a scripting language:

```
curl "http://download.finance.yahoo.com/d/quotes.csv?\
s=BBDB.TO+NT.TO+GE+MSFT&f=snl1d1t1ohgdr"

"BBD-B.TO","BOMBARDIER INC., ",5.26,"1/7/2011","4:40pm",5.40,5.41,5.24,0.10,15.91
"NT.TO","NT.TO",0.00,"N/A","11:-40am",N/A,N/A,N/A,N/A,N/A
"GE","General Electric ",18.43,"1/7/2011","4:00pm",18.58,18.66,18.20,0.46,20.26
"MSFT","Microsoft Corpora",28.60,"1/7/2011","4:00pm",28.64,28.74,28.25,0.55,12.42
```

IP Addresses

You almost always know the IP address of the machine a user is connecting to your web service from, and it's possible to guess a lot of information based on that number. In fact, it's so powerful that it's treated as personally identifiable information in the EU, so for privacy-sensitive applications I'd recommend you remove IP addresses from your logs entirely.

You do need to be careful, though, since the information derived from IP addresses is unreliable. In particular it's not a good idea to make hard-and-fast decisions based on the geographic results. It can be very useful to use that information to intelligently set defaults for languages or home locations, though, as long as the user can correct them subsequently.

MaxMind

The GeoLite City database by MaxMind is one of my all-time favorite resources. It's a comprehensive data set listing the city and country locations of a large number of IP addresses, released under a data license that does require credit but is otherwise very open. It also comes bundled with a set of APIs for a lot of different languages that make it easy to do the IP to location lookup from within your own programs. GeoLite City also offers a richer commercial data set that you can drop into a system using the free files, without requiring any code changes.

```
  {
   "kind": "shopping#product",
   "id": "tag:google.com,2010:shopping/products/10048/16324480569195213774",
   "selfLink": "https://www.googleapis.com/shopping/search/v1/
public/products/10048/gid/16324480569195213774",
   "product": {
    "googleId": "16324480569195213774",
    "author": {
     "name": "Staples",
     "accountId": "10048"
    },
    "creationTime": "2010-04-18T16:18:40.000Z",
    "modificationTime": "2011-01-07T19:15:40.000Z",
    "country": "US",
    "language": "en",
    "title": "Olympus Stylus TOUGH-3000 Digital Camera, Pink",
    "description": "The STYLUS TOUGH-3000 is loads of fun for the whole family
...
```


Google Book Search

If you're looking specifically for a book, this second API from Google will probably be more useful than the more general Shopping engine. You can do a search using all the advanced operators from the Book Search web interface, looking for titles, authors, or ISBNs. You don't have to authenticate or get an API key to use this, and the results are available as a public Atom-format RSS feed, so it's unclear what restrictions there are on its use:

```
curl "http://books.google.com/books/feeds/volumes?q=instrumentality+of+mankind"
```

Netflix

If you're dealing with movies or TV shows, this API offers a lot of information. The formal terms of service are reasonable, and the ten commandments version on the main page should be an example to API providers everywhere, making clear what Netflix's goals are without clouding them in legalese—"Delight Customers," "Handle Content with Care," and "Respect the Netflix Brand." The rate limits might be a little tight, at 5,000 per day, but that's per user if you're doing calls for logged-in Netflix accounts. Unfortunately, almost all of the methods do require three-legged OAuth authorization, apart from auto-complete. The good news is that you can download the entire 1.3GB, 130,000+ title database through this OAuth-required link: *http://api.netflix.com/catalog/titles/full?v=2.0*.

Having the bulk data available means you can integrate it very flexibly with your own backend. It still comes with the same restrictions as the API, but these seem well-crafted to cramp innovation as little as possible.

Yahoo! Music

This service lets you query both a large back catalog of music and information on the current charts using YQL. You're limited to 5,000 queries a day, and the information you get back on artists isn't very extensive, but it's simple to access, as it doesn't require any authentication:

```
curl "http://query.yahooapis.com/v1/public/yql?\
q=select%20*%20from%20music.artist.search%20where%20keyword%3D'Rihanna'&format=json"

{"query":{"count":"4","created":"2011-01-09T16:45:20Z","lang":"en-US","results":{
  "Artist":[{"catzillaID":"1927869111","flags":"124547","hotzillaID":"1809845326",
 "id":"19712698","name":"Rihanna","rating":"-1","trackCount":"453",
 "url":"http://new.music.yahoo.com/rihanna/",
 "website":"http://rihannanow.com","ItemInfo":{"Relevancy":{"index":"465"}}},
...
```

Musicbrainz

This site has assembled a large collection of information on music artists and works, and it has made the results available for download under an open license. It's reusable under a mixture of public domain and Creative Commons licensing, depending on the attributes you're looking at.

You can also access the same data through the online REST/XML API, where you can look up artists and works and get back quite a lot of information, not only about the people and albums, but also about their relationships to one another:

```
curl "http://musicbrainz.org/ws/1/artist/?type=xml&name=Tori+Amos"

<?xml version="1.0" encoding="UTF-8"?>
...
<artist type="Person" id="c0b2500e-0cef-4130-869d-732b23ed9df5" ext:score="100">
<name>Tori Amos</name><sort-name>Amos, Tori</sort-name>
<life-span begin="1963-08-22"/></artist>
...
```

The Movie DB

A rival site to the well-established IMDB, Movie DB has an API that gives you access to details on a wide range of movies. The terms of service aren't too constraining, with no requirement that your project be immediately end-user-facing, though Movie DB does discourage caching of data. I couldn't find any information on rate limits:

```
curl "http://api.themoviedb.org/2.1/Movie.search/en/xml/<key>/Transformers"

<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
  <opensearch:Query searchTerms="transformers"/>
  <opensearch:totalResults>8</opensearch:totalResults>
  <movies>
```