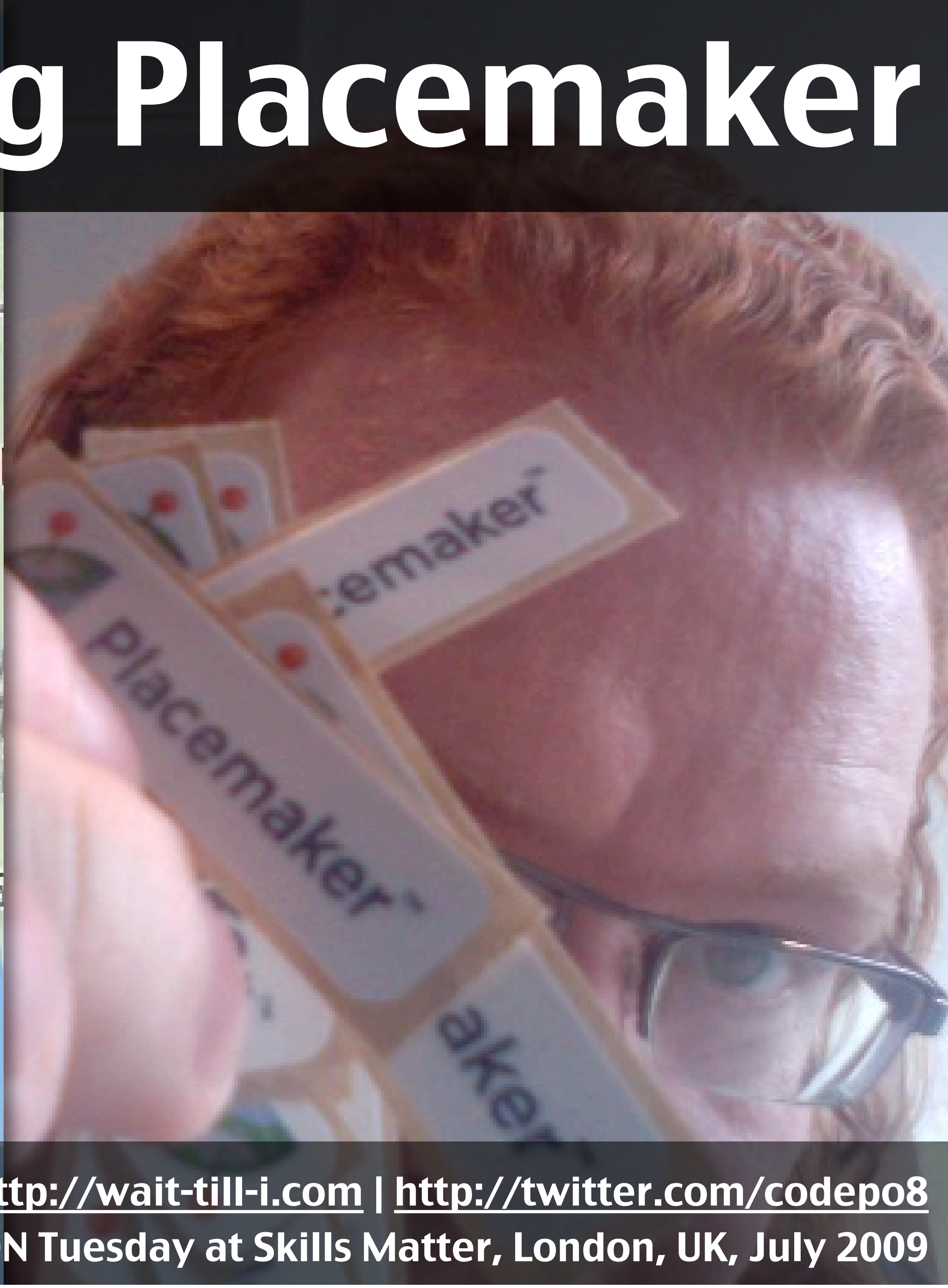
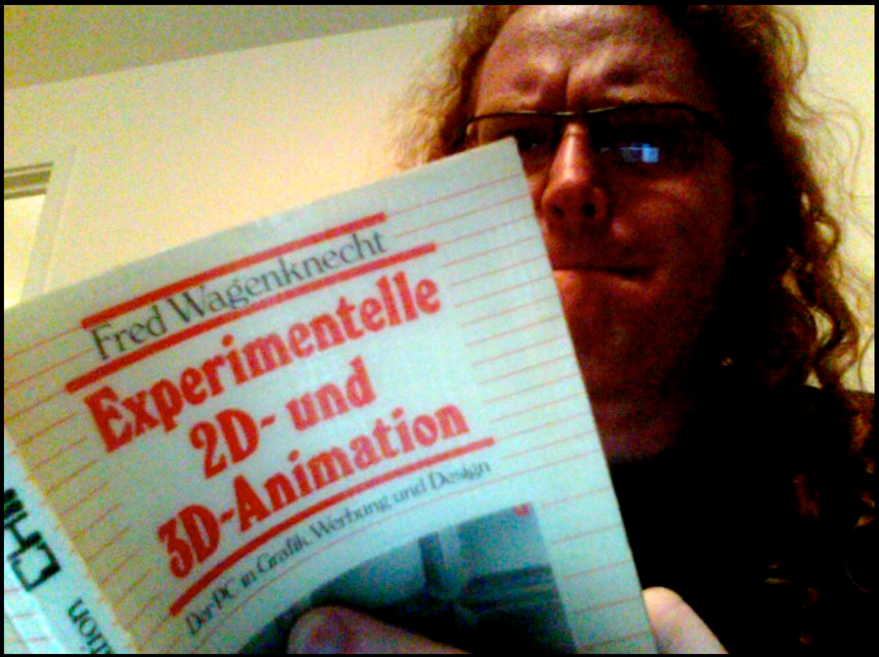
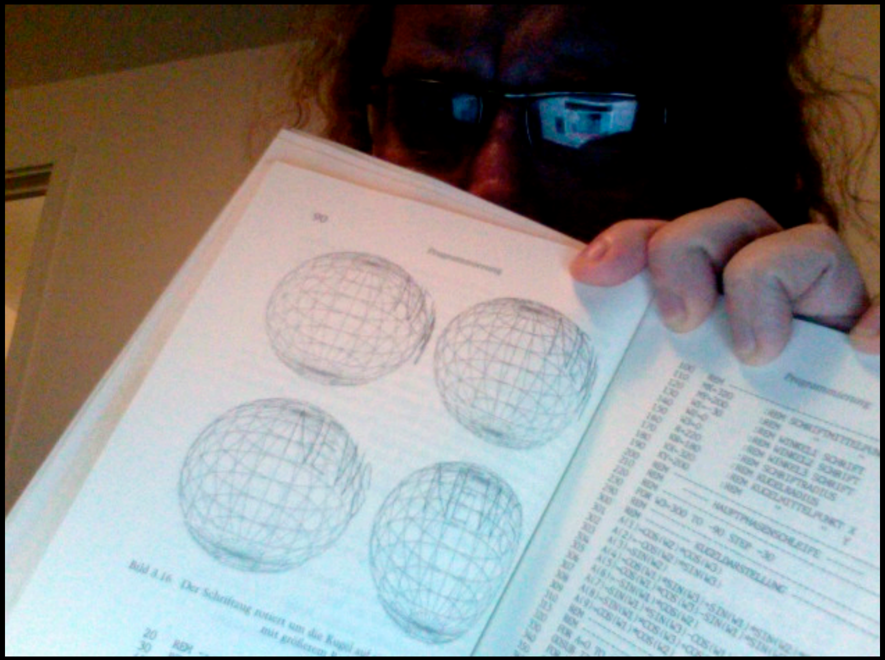
Introducing Placemaker



Christian Heilmann | http://wait-till-i.com | http://twitter.com/codepo8
YDN Tuesday at Skills Matter, London, UK, July 2009

Spatial

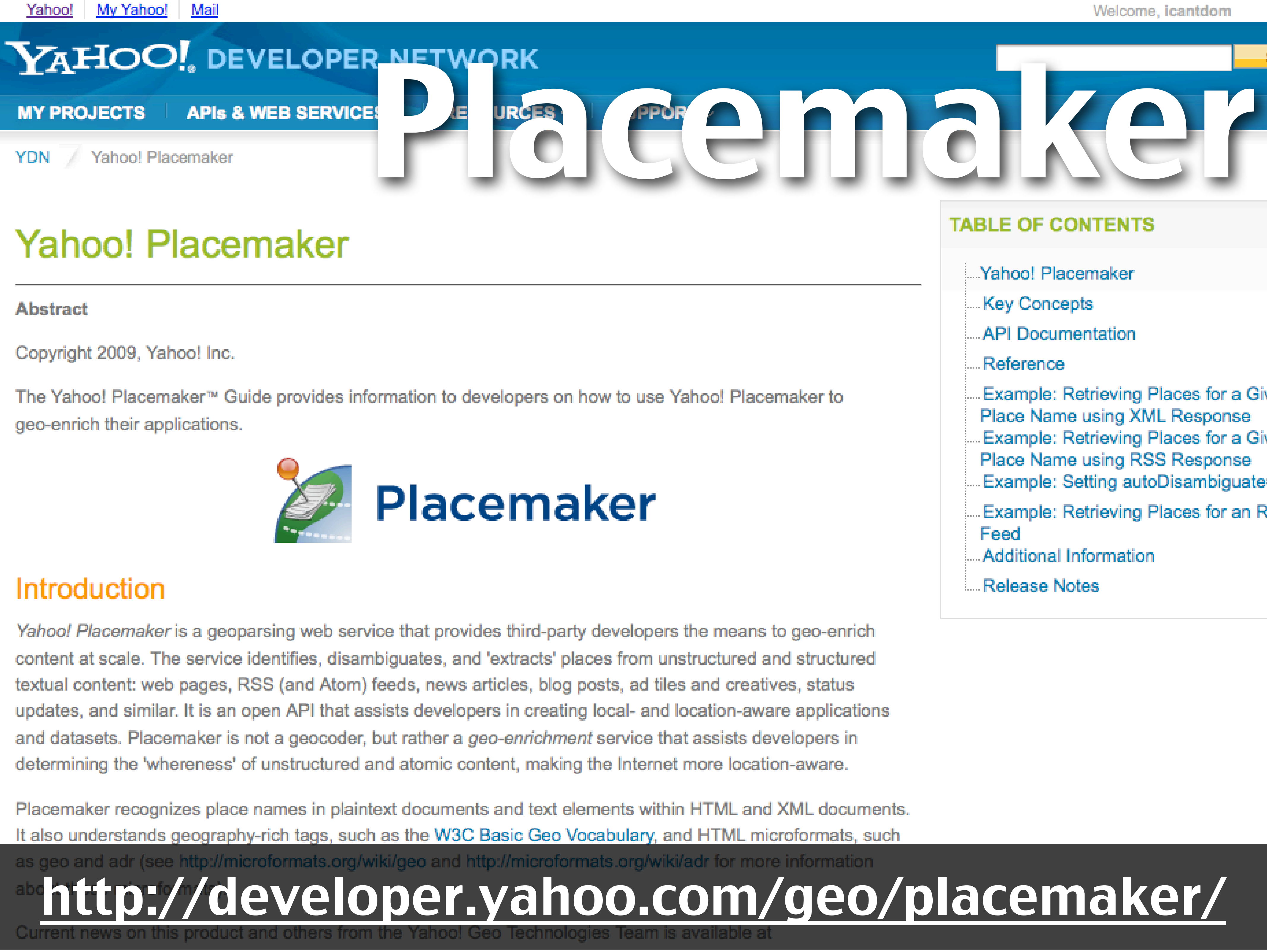

Fred Wagenknecht
Experimentelle 2D- und 3D-Animation
Der PC in Grafik, Werbung und Design



Fanboy

Hello, I am Chris. Hacker by passion. When I went to the first WhereCamp about two years ago I thought nobody can out-geek me. I was wrong. Geolocation and Geocoding is quite some hard-core branch of geekery. So let me tell you about a nice little product that makes things easy for you.

Placemaker



Yahoo! Placemaker

Abstract

Copyright 2009, Yahoo! Inc.

The Yahoo! Placemaker™ Guide provides information to developers on how to use Yahoo! Placemaker to geo-enrich their applications.

Introduction

Yahoo! Placemaker is a geoparsing web service that provides third-party developers the means to geo-enrich content at scale. The service identifies, disambiguates, and 'extracts' places from unstructured and structured textual content: web pages, RSS (and Atom) feeds, news articles, blog posts, ad tiles and creatives, status updates, and similar. It is an open API that assists developers in creating local- and location-aware applications and datasets. Placemaker is not a geocoder, but rather a *geo-enrichment* service that assists developers in determining the 'whereness' of unstructured and atomic content, making the Internet more location-aware.

Placemaker recognizes place names in plaintext documents and text elements within HTML and XML documents. It also understands geography-rich tags, such as the W3C Basic Geo Vocabulary, and HTML microformats, such as geo and adr (see http://microformats.org/wiki/geo and http://microformats.org/wiki/adr for more information abo

Current news on this product and others from the Yahoo! Geo Technologies Team is available at

TABLE OF CONTENTS

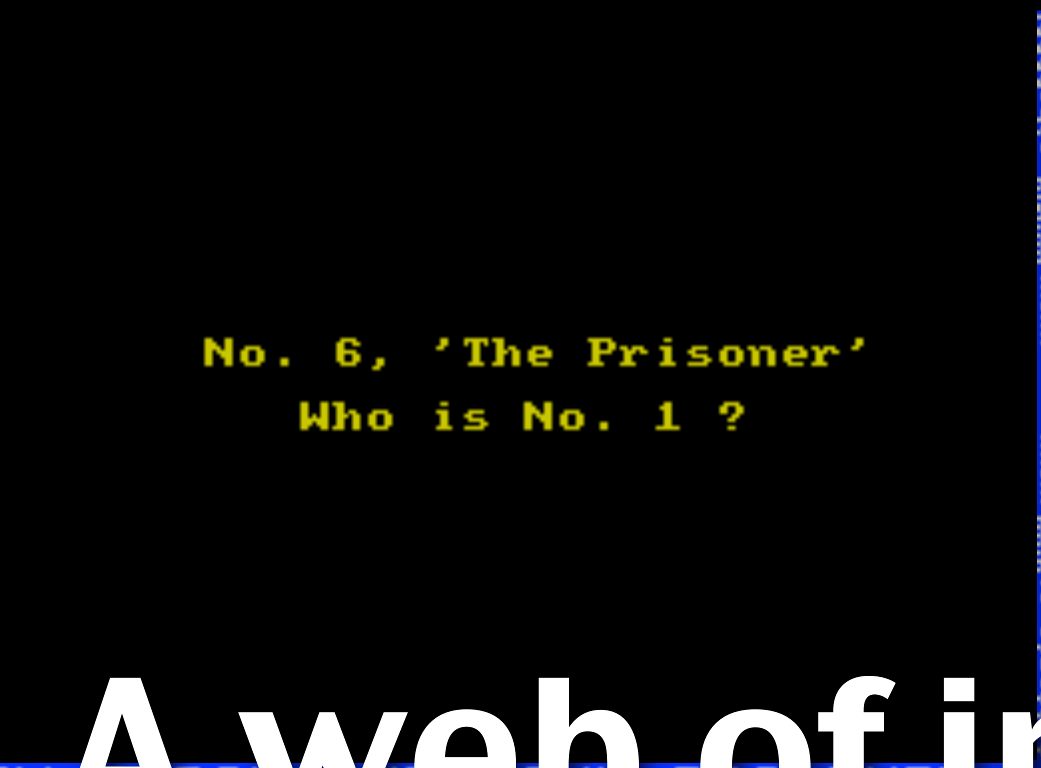
http://developer.yahoo.com/geo/placemaker/

This is Yahoo Placemaker and it is an API. You give it a URL to get data from or a text to extract geographical information from. Here are the docs. Now go forth and build cool stuff.

Thanks.

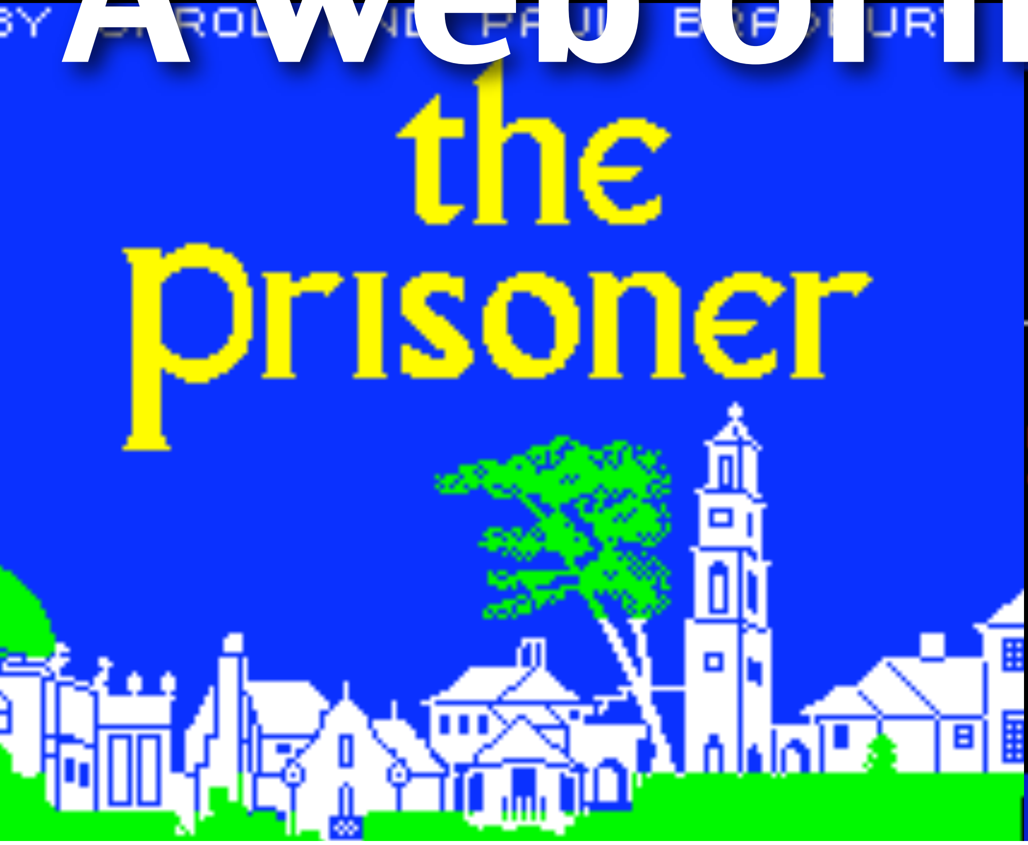
Any questions?

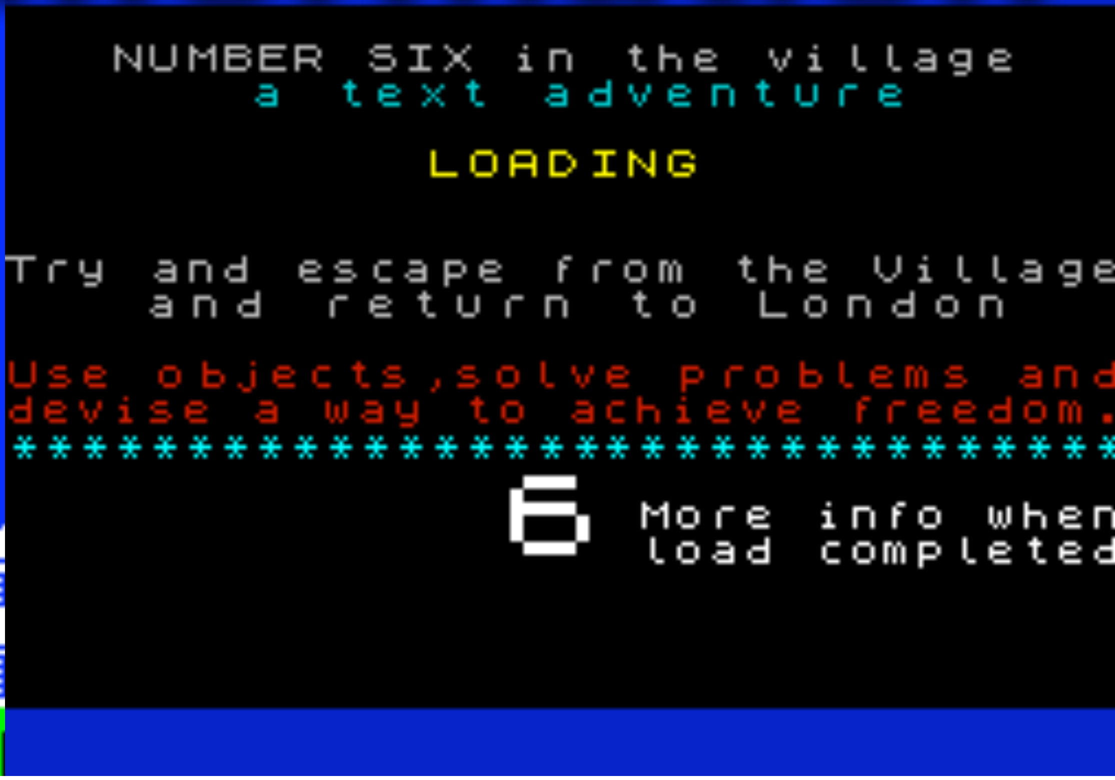
A web of information



No. 6, 'The Prisoner'

Who is No. 1 ?





NUMBER SIX in the village
a text adventure

LOADING

Try and escape from the Village
and return to London

Use objects,solve problems and
devise a way to achieve freedom.

****

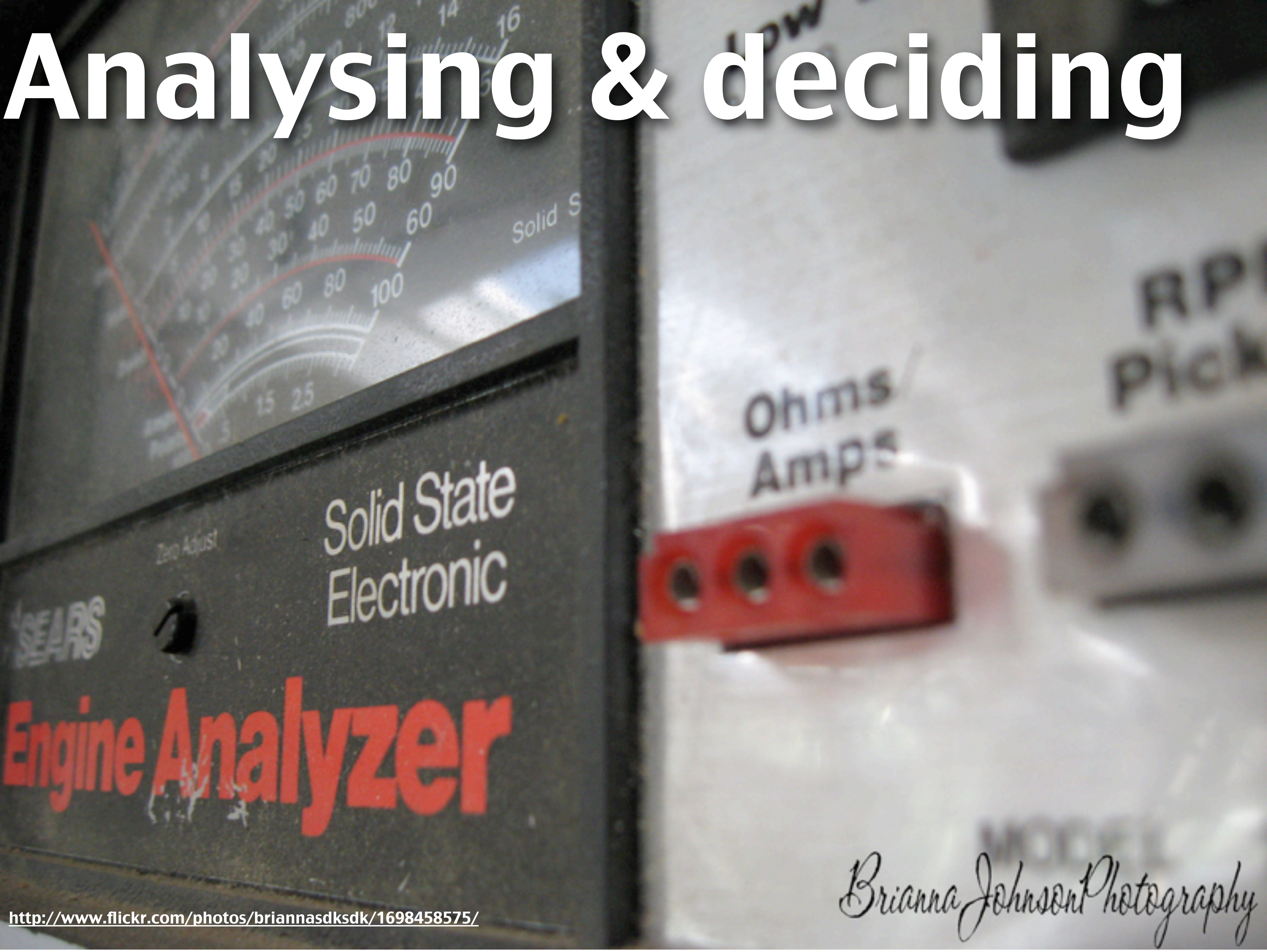
6 More info when
load completed

The web is full of information. Which is cool. The problem is that we accumulated and still accumulate more and more information without giving it proper structure.

Searching & Finding

Search engines help us find stuff. However, as being found means making money the first search results are not necessarily the best – only the ones that have been promoted the best way.

Analysing & deciding



http://www.flickr.com/photos/briannasdksdk/1698458575/

Analyzing all the data of the web is a massive job. And computers are stupid. Computers are decision engines that would be thoroughly stumped when asked "do I look fat in this dress" as they forget the underlying dangers in answering this question in one way or another.

Human additives

This is why we need humans. By enriching our content with structured, easier to parse data we make it easier for machines to harvest only the necessary parts of our documents. In the past that was keywords, now we use microformats and tagging. The latter is very useful as it can be crowdsourced. People tagging my photos on flickr or my site on del.icio.us make it easier for them to find them later on and give me an idea what keywords I hadn't thought of.

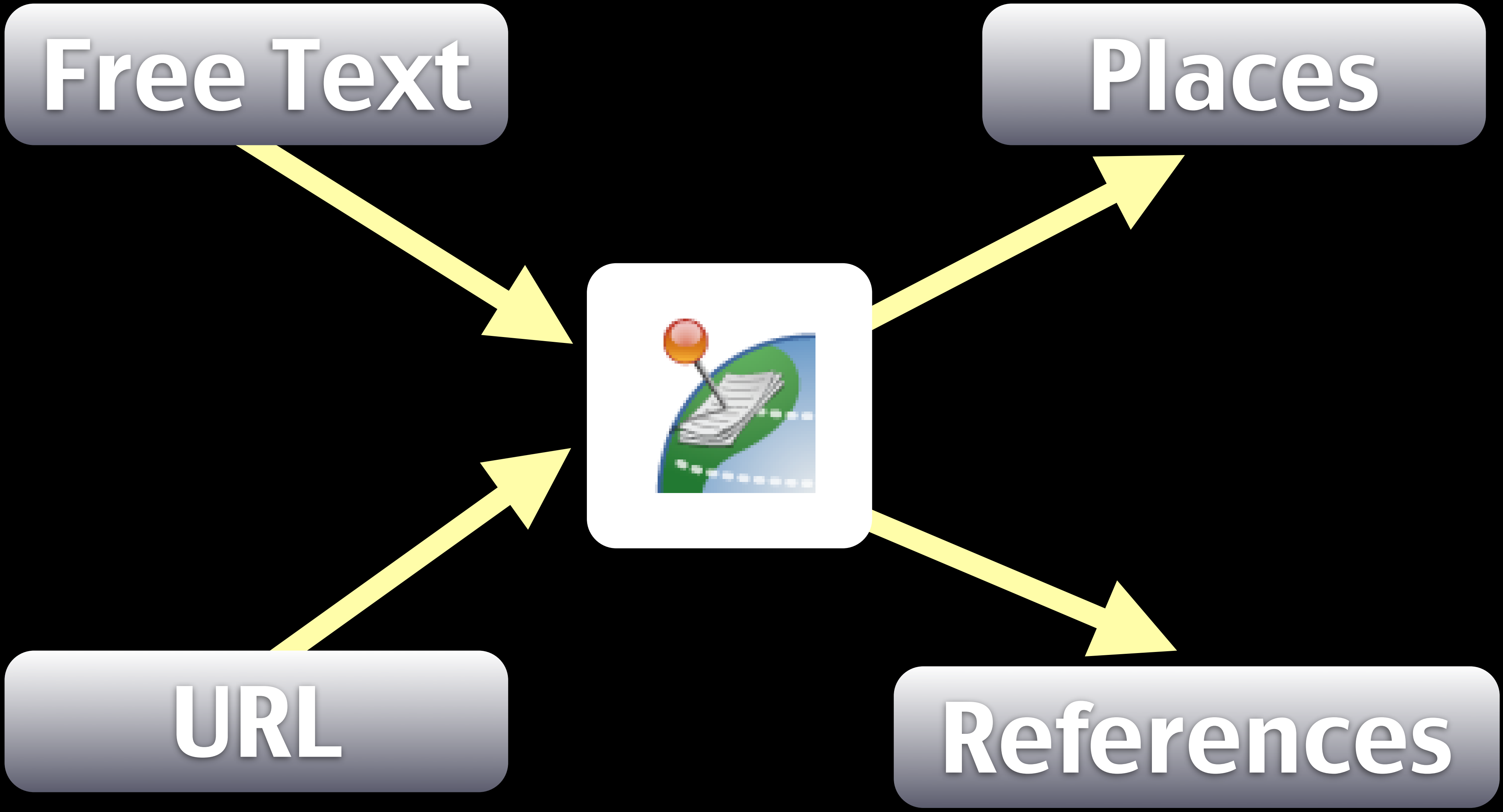
Mobility

http://vimeo.com/2638558

This is all fine and good, but the real change we see in behaviour of web users is that we become more and more mobile. This also means that people can locate themselves on the planet and expect information from their physical surroundings. In other words, for our content to be found we need to have geographical information in there that defines the locality of the text, not only what it talks about.

Hidden goodies

Texts on the web have all kind of great information in them – if we find a tool that finds and marks them for us.



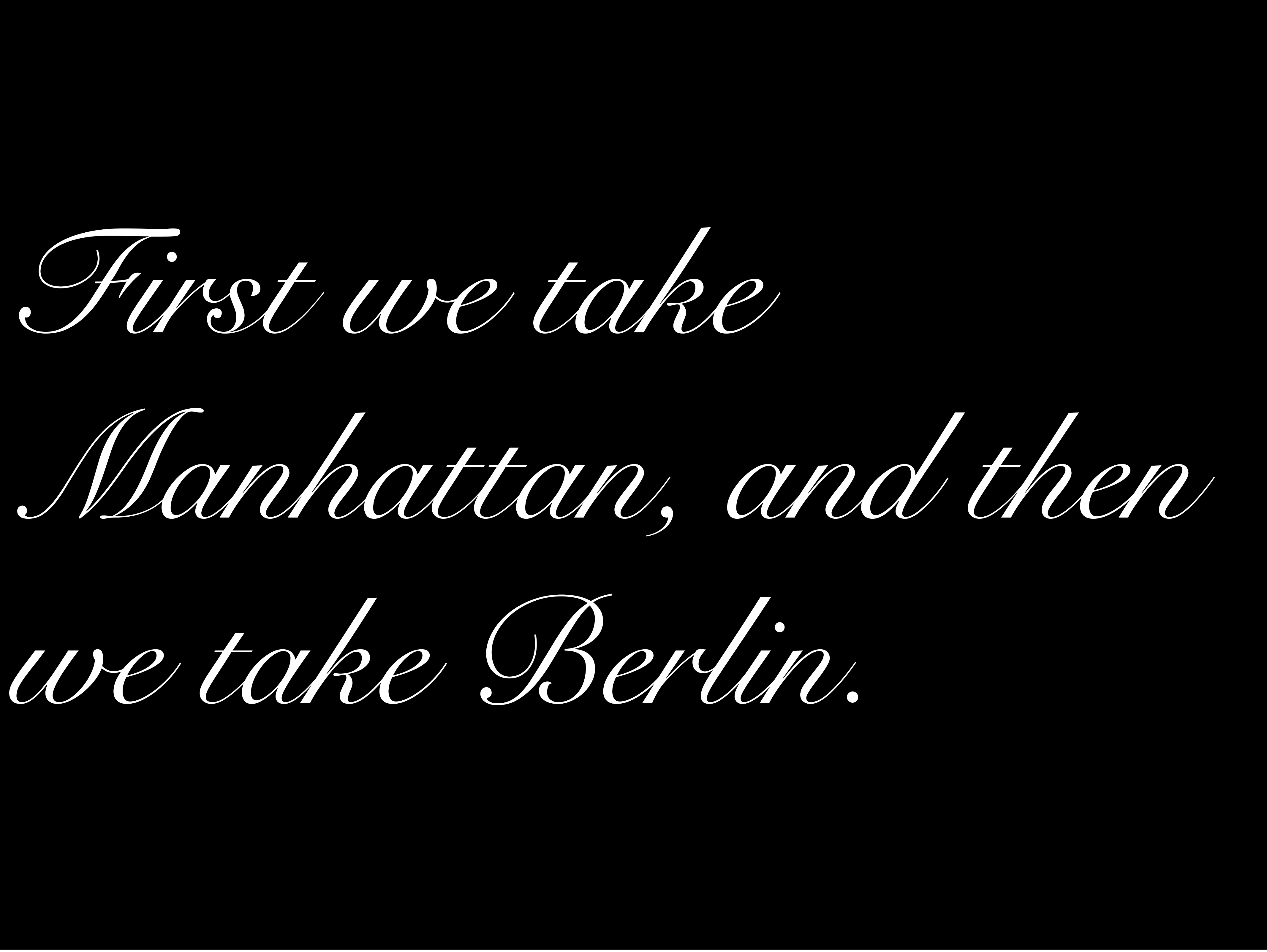
Free Text

URL

Places

References

Placemaker is a service that converts text or URLs in places and references.



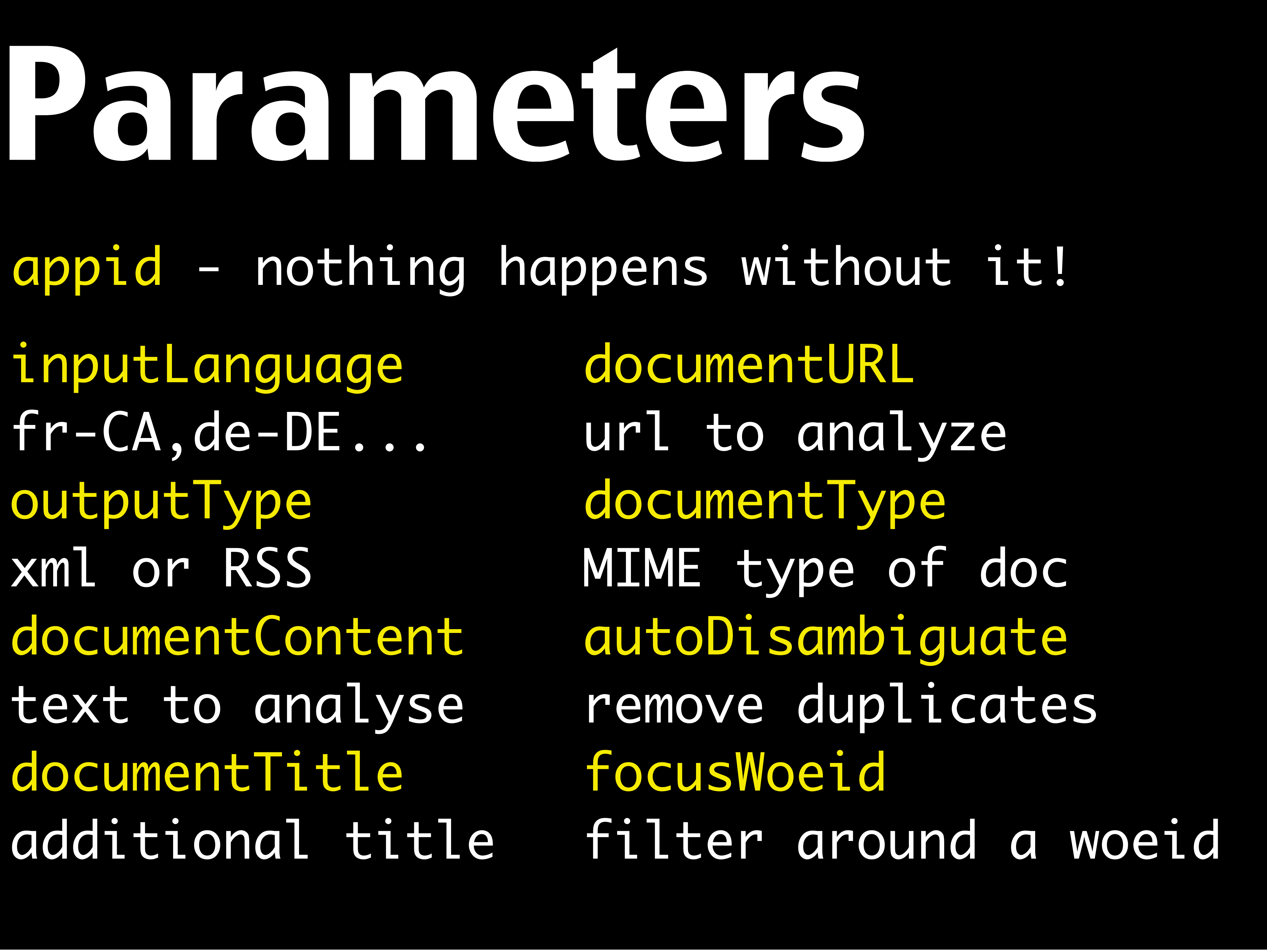
First we take Manhattan, and then we take Berlin.

Let’s take this classic sentence for example.

Calling Placemaker

```
http://wherein.yahooapis.com/v1/
document

documentContent=First+we+take
+Manhattan+and+then+we+take
+Berlin.
documentType=text/plain
appid=my_appid
```

To find the geographical information in this text simply send it to Placemaker via POST.



Parameters

appid - nothing happens without it!

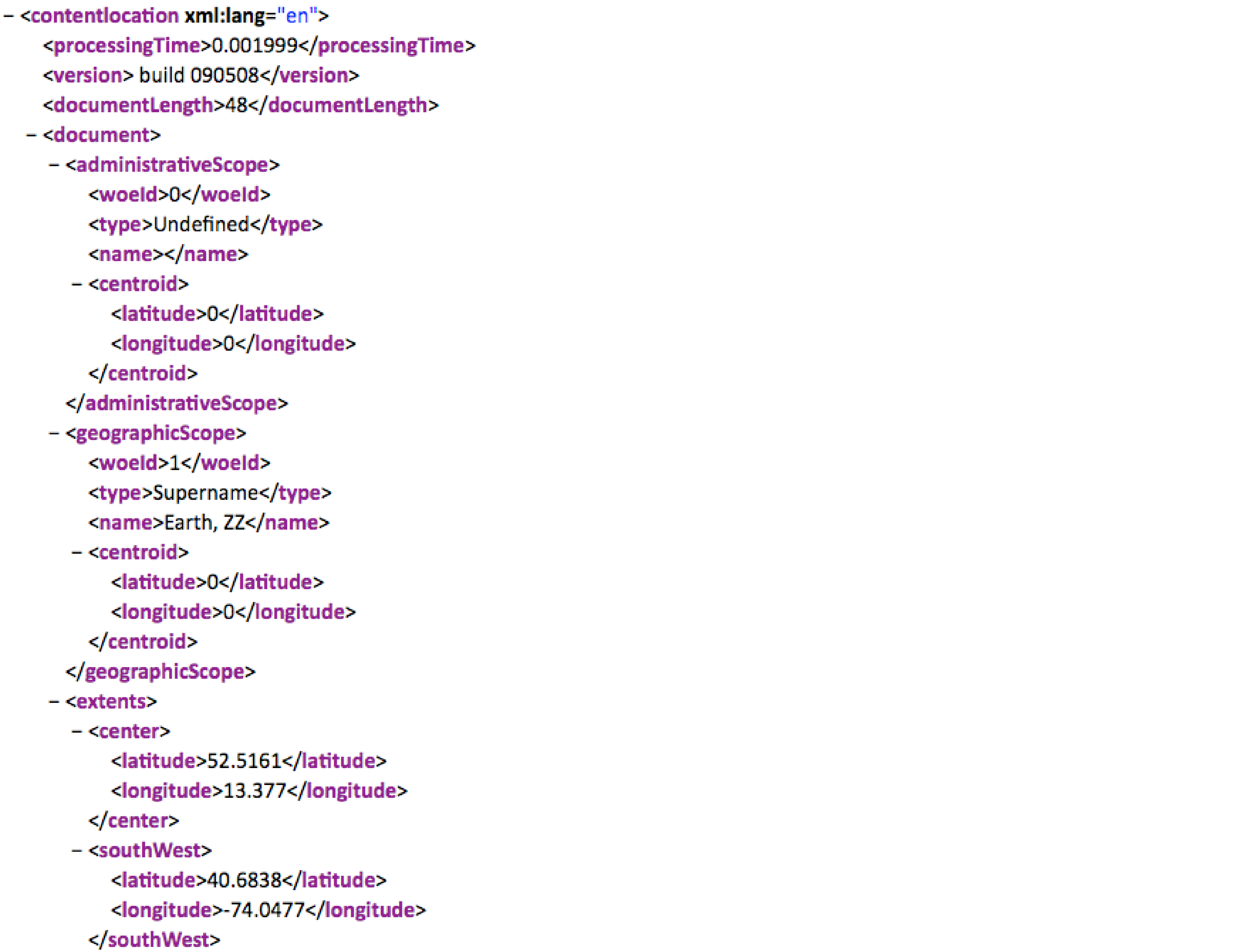
inputLanguage
fr-CA,de-DE...
outputType
xml or RSS
documentContent
text to analyse
documentTitle
additional title

documentURL
url to analyze
documentType
MIME type of doc
autoDisambiguate
remove duplicates
focusWoeid
filter around a woeid

Placemaker takes several parameters that allows you to get the data you want.

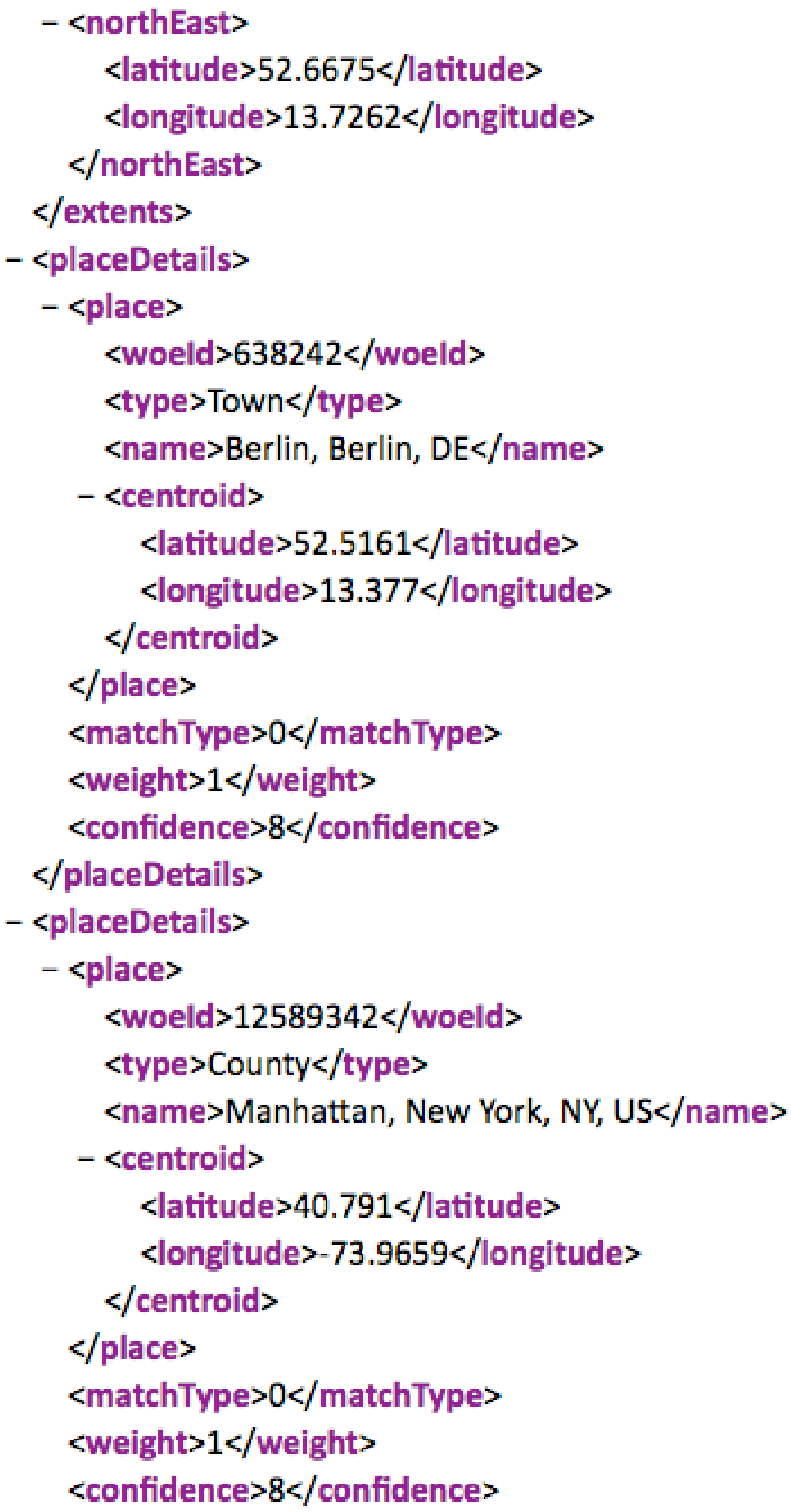
First we take Manhattan, and then we take Berlin.

Let’s take this classic sentence and run it through Placemaker.

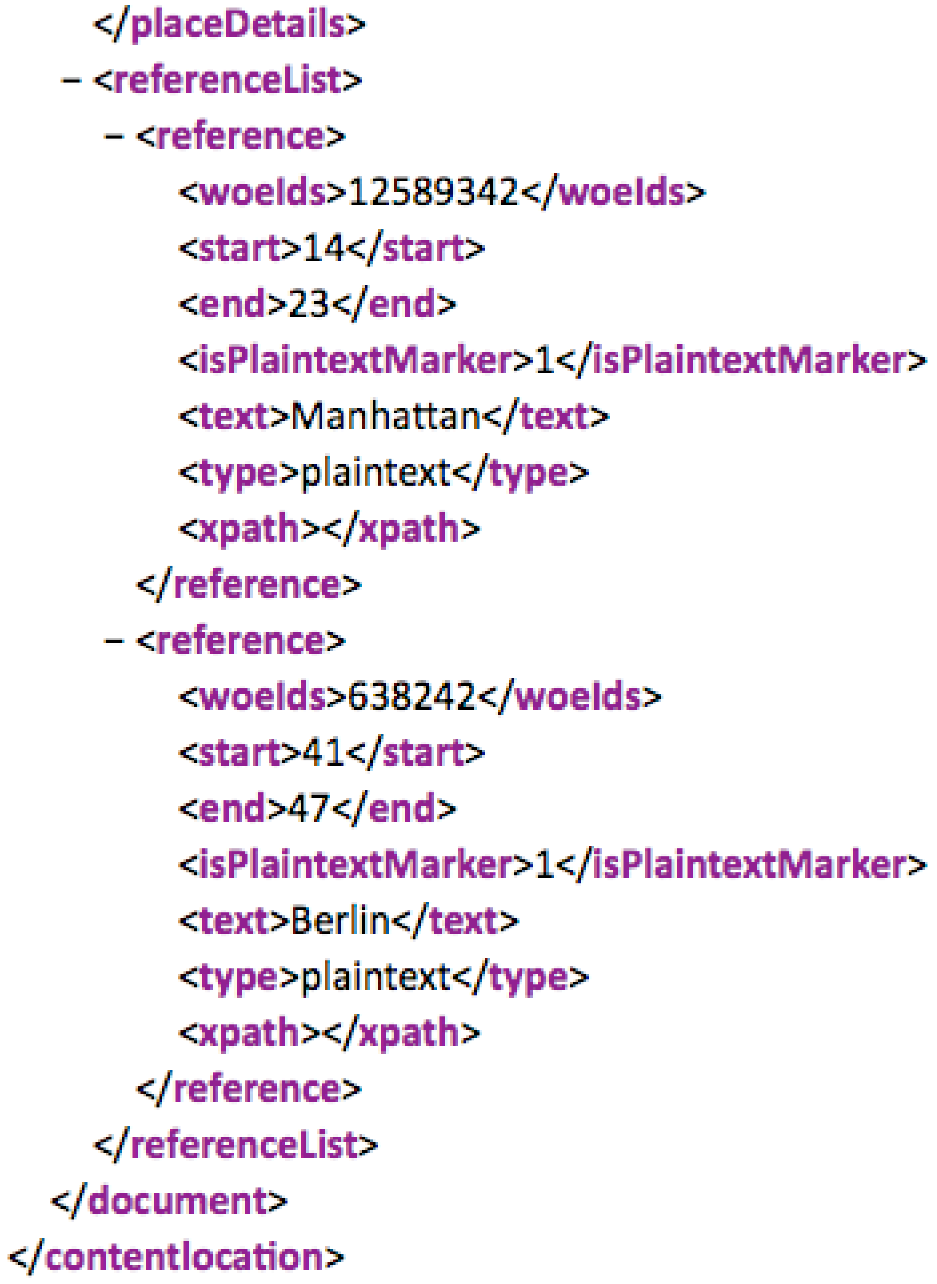


```
- <contentlocation xml:lang="en">
    <processingTime>0.001999</processingTime>
    <version> build 090508</version>
    <documentLength>48</documentLength>
  - <document>
    - <administrativeScope>
        <woeId>0</woeId>
        <type>Undefined</type>
        <name></name>
      - <centroid>
          <latitude>0</latitude>
          <longitude>0</longitude>
        </centroid>
      </administrativeScope>
    - <geographicScope>
        <woeId>1</woeId>
        <type>Supername</type>
        <name>Earth, ZZ</name>
      - <centroid>
          <latitude>0</latitude>
          <longitude>0</longitude>
        </centroid>
      </geographicScope>
    - <extents>
      - <center>
          <latitude>52.5161</latitude>
          <longitude>13.377</longitude>
        </center>
      - <southWest>
          <latitude>40.6838</latitude>
          <longitude>-74.0477</longitude>
        </southWest>
```

This is the result we get back from Placemaker when we send the text through it.

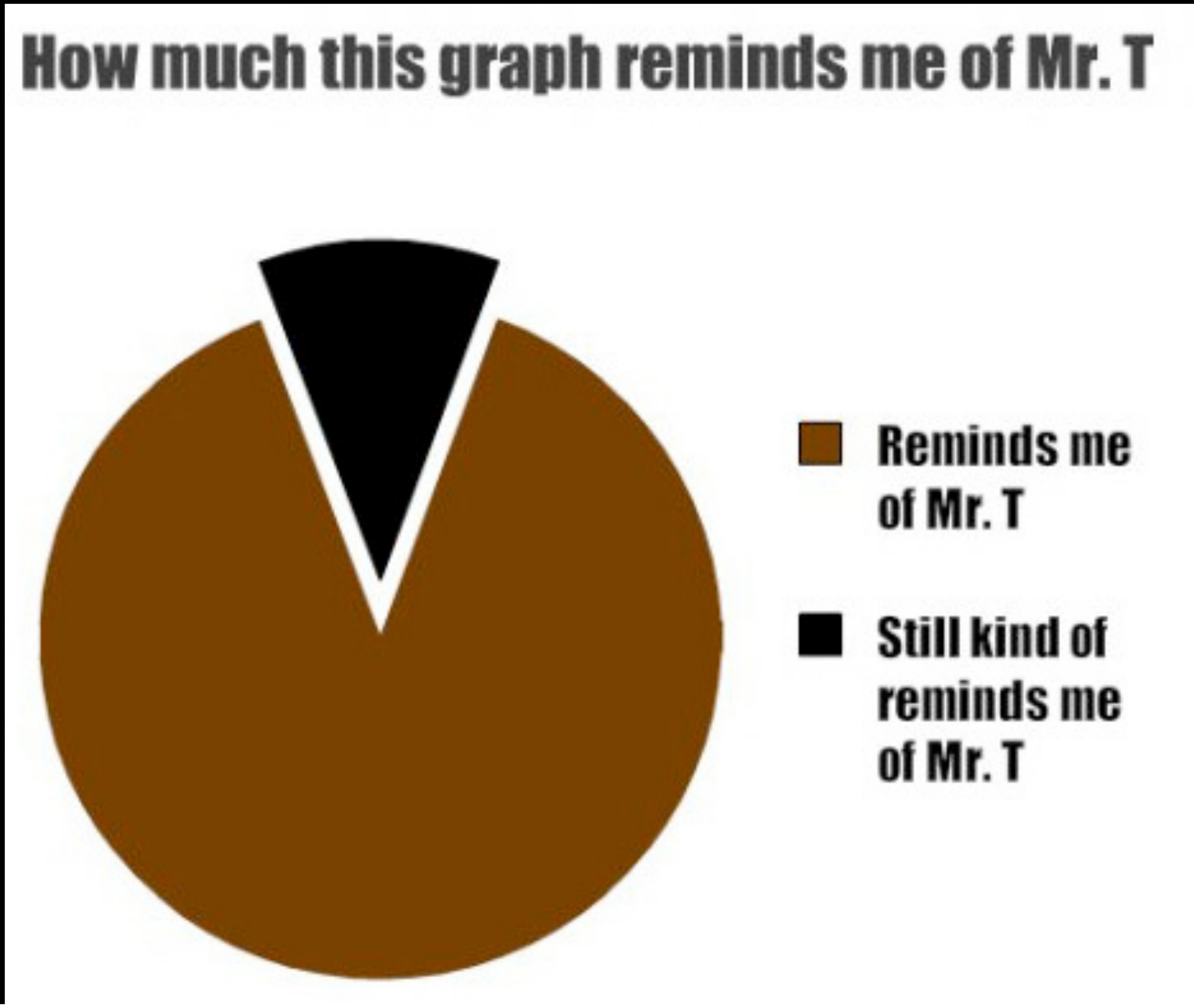


```
    − <northEast>
        <latitude>52.6675</latitude>
        <longitude>13.7262</longitude>
      </northEast>
    </extents>
  − <placeDetails>
    − <place>
        <woeId>638242</woeId>
        <type>Town</type>
        <name>Berlin, Berlin, DE</name>
      − <centroid>
          <latitude>52.5161</latitude>
          <longitude>13.377</longitude>
        </centroid>
      </place>
      <matchType>0</matchType>
      <weight>1</weight>
      <confidence>8</confidence>
    </placeDetails>
  − <placeDetails>
    − <place>
        <woeId>12589342</woeId>
        <type>County</type>
        <name>Manhattan, New York, NY, US</name>
      − <centroid>
          <latitude>40.791</latitude>
          <longitude>-73.9659</longitude>
        </centroid>
      </place>
      <matchType>0</matchType>
      <weight>1</weight>
      <confidence>8</confidence>
```



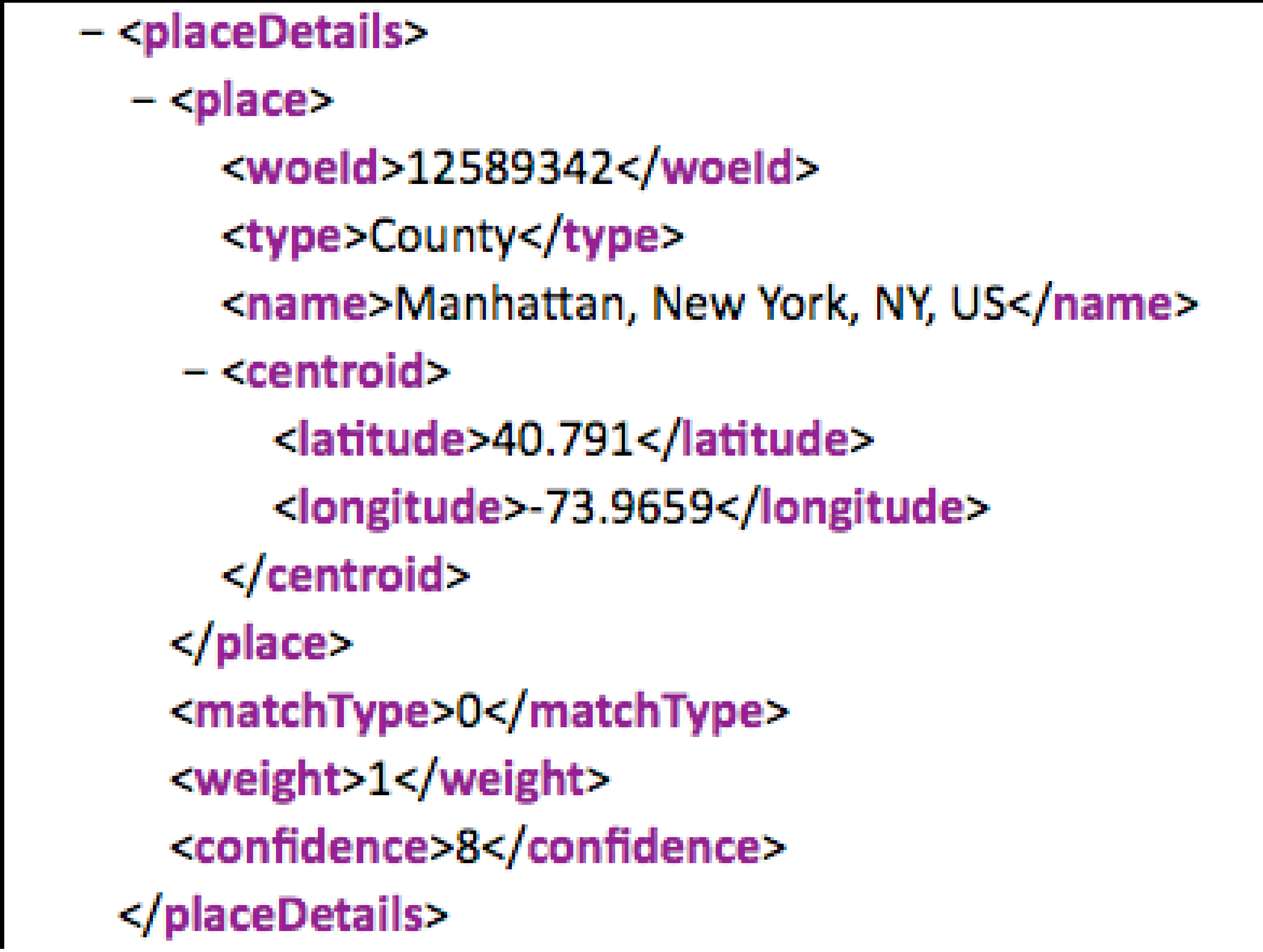
```
      </placeDetails>
    − <referenceList>
      − <reference>
          <woeIds>12589342</woeIds>
          <start>14</start>
          <end>23</end>
          <isPlaintextMarker>1</isPlaintextMarker>
          <text>Manhattan</text>
          <type>plaintext</type>
          <xpath></xpath>
        </reference>
      − <reference>
          <woeIds>638242</woeIds>
          <start>41</start>
          <end>47</end>
          <isPlaintextMarker>1</isPlaintextMarker>
          <text>Berlin</text>
          <type>plaintext</type>
          <xpath></xpath>
        </reference>
      </referenceList>
    </document>
  </contentlocation>
```

Working with Placemaker results



As with any data, it is important to understand what the interesting parts of the data are.

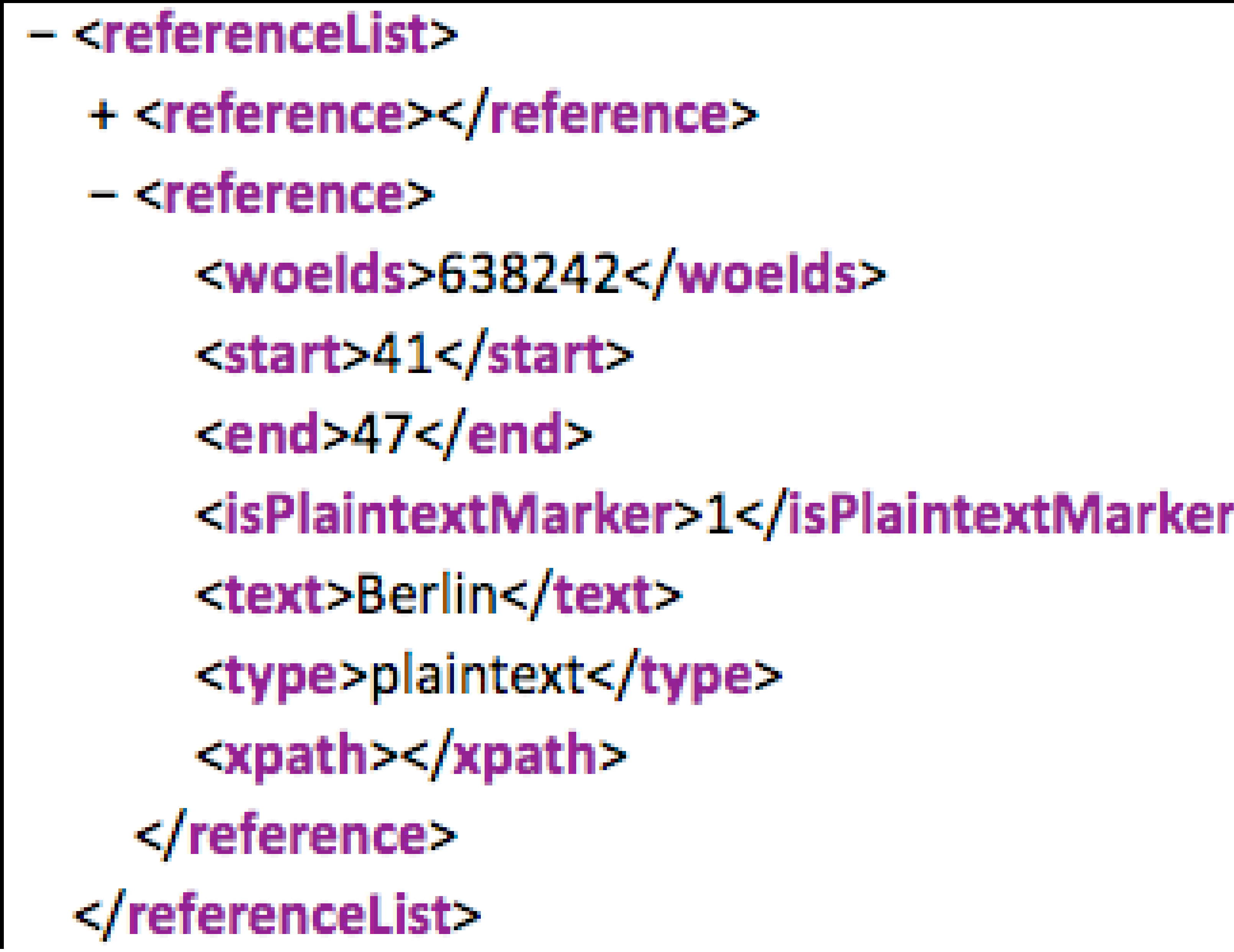
Places



```
– <placeDetails>
  – <place>
      <woeId>12589342</woeId>
      <type>County</type>
      <name>Manhattan, New York, NY, US</name>
    – <centroid>
        <latitude>40.791</latitude>
        <longitude>-73.9659</longitude>
      </centroid>
    </place>
    <matchType>0</matchType>
    <weight>1</weight>
    <confidence>8</confidence>
  </placeDetails>
```

First up is a list of places the API found in the text. These are PlaceDetails elements with a nested place element:

References



```
- <referenceList>
   + <reference></reference>
   - <reference>
        <woelds>638242</woelds>
        <start>41</start>
        <end>47</end>
        <isPlaintextMarker>1</isPlaintextMarker
        <text>Berlin</text>
        <type>plaintext</type>
        <xpath></xpath>
     </reference>
  </referenceList>
```

This is cool, but it doesn't tell us where this information came from. For this there is a referenceList element with an array of references. In order to find out where the text Placemaker found as a match is located in the document, you either get start and end for text content of the XPATH for structured content (XML/RSS).

Annoyances

There are a few annoyances when it comes to working with Placemaker.

50,000 bytes

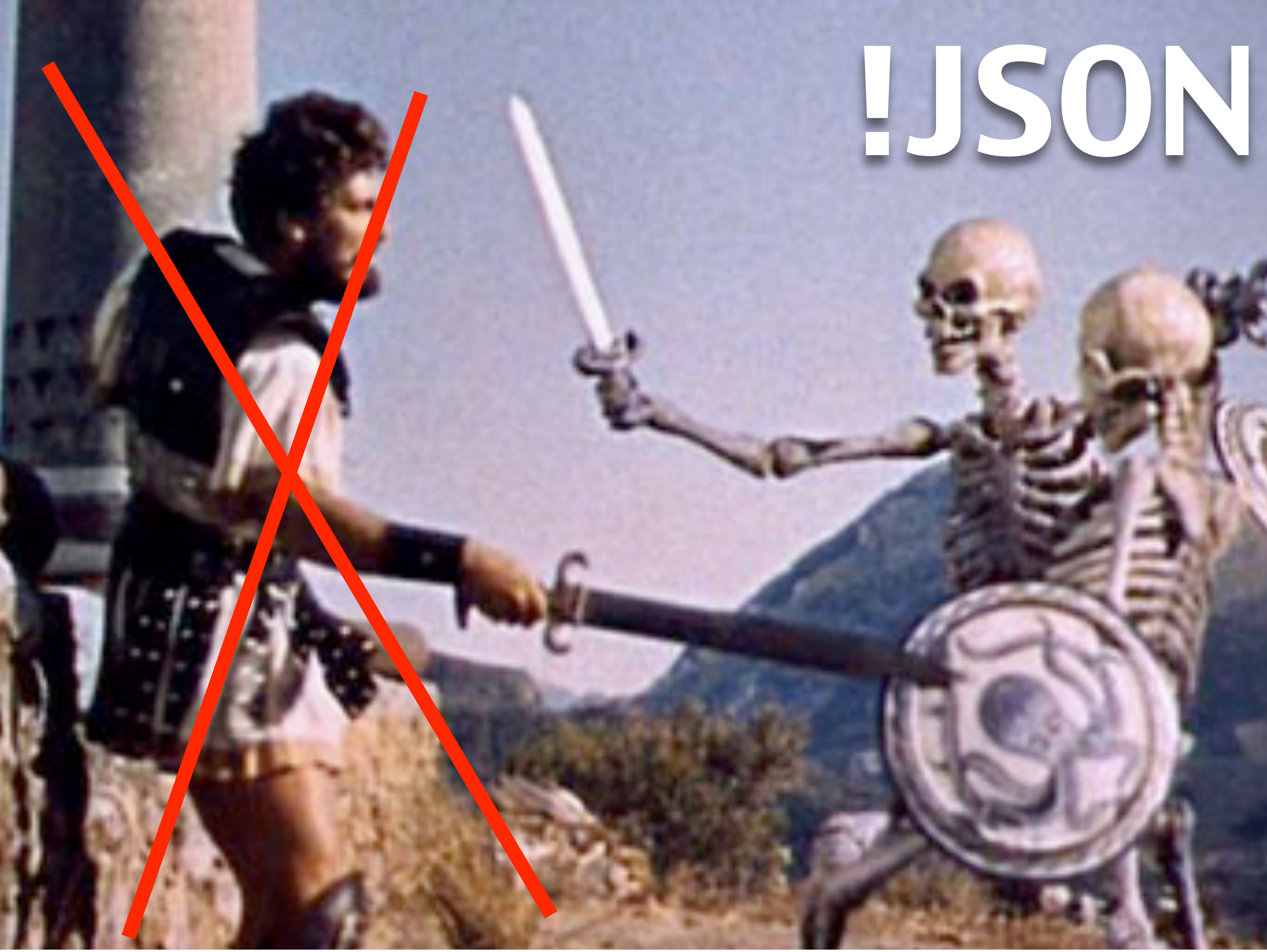
Placemaker text requests are limited to 50,000 bytes which is a lot but may not be enough, especially when the data you try to analyse is not your own.

The broken web

The web is full of terribly badly organized content. Using the documentUrl property Placemaker happily loads the information but can choke on some things content providers sadly enough do - like encoding documents in UTF-8 and then using non-UTF-8 characters.

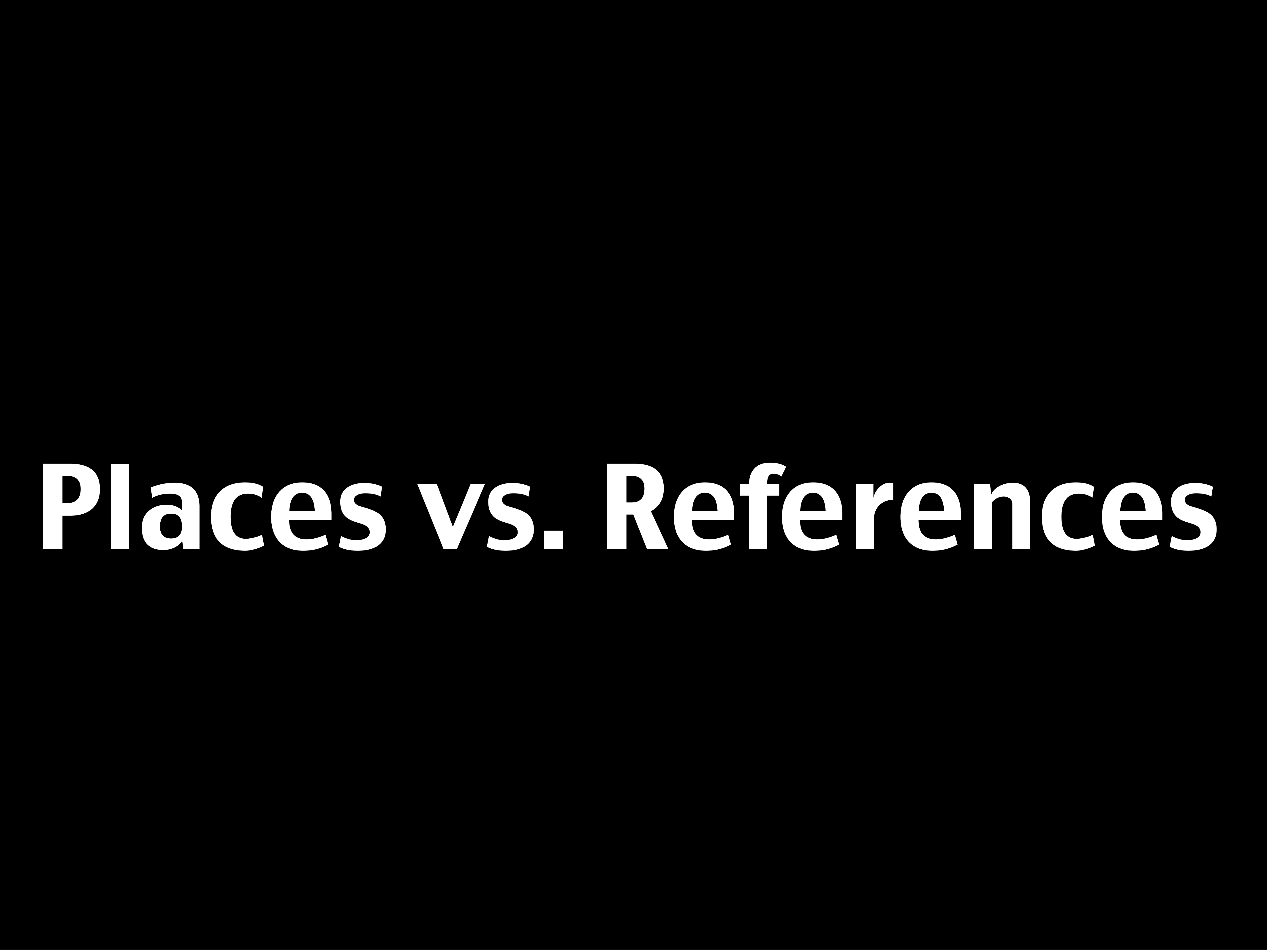
POST vs. GET

Placemaker does not allow for GET requests, even if they were short enough to go through without problems.



!JSON

Placemaker has no JSON output at the moment, which means you cannot use results in JavaScript without writing an own converter.

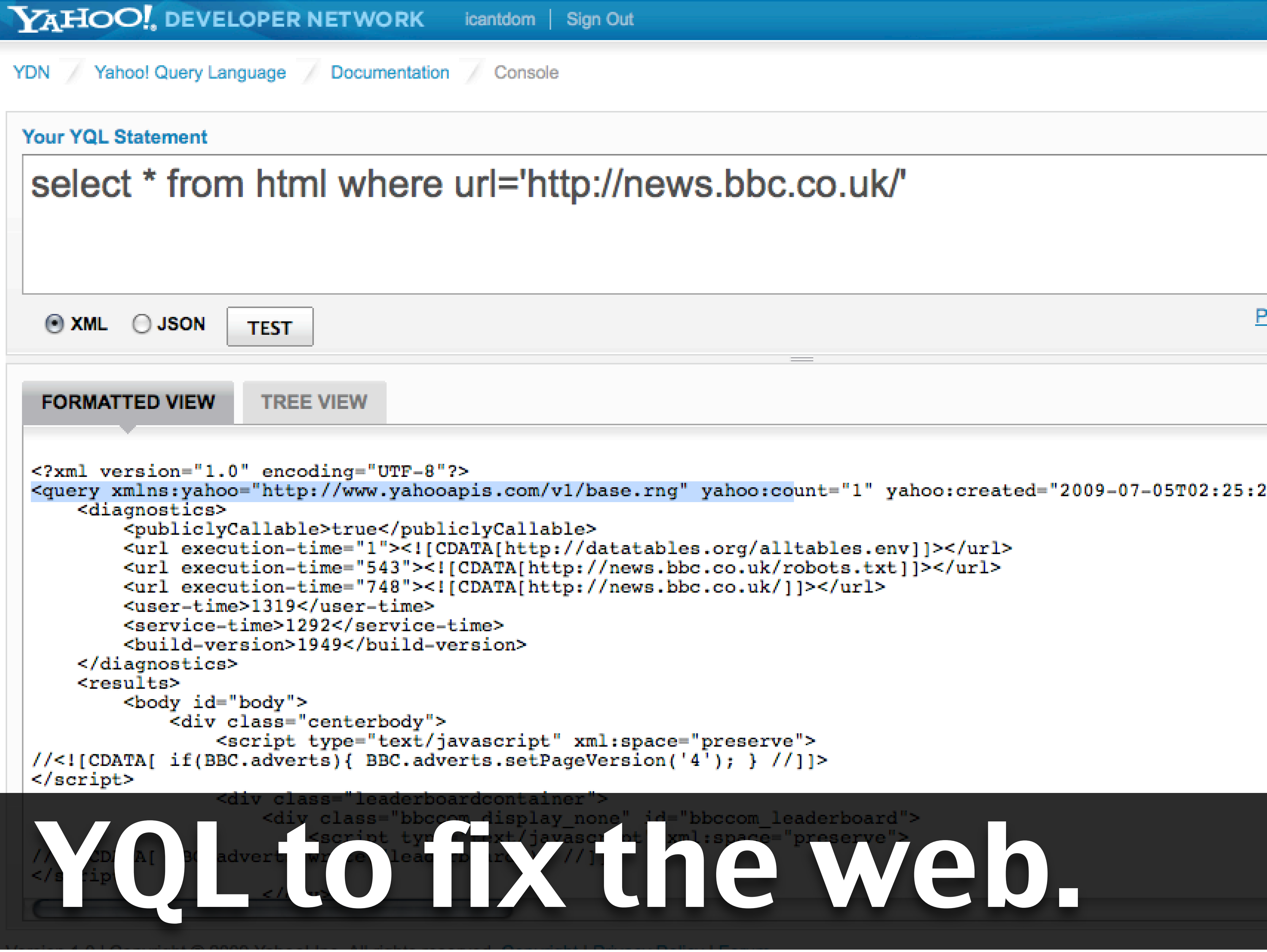


Places vs. References

There is a list of Places and a list of References, but they are not directly connected. Furthermore References have a parent element whereas places don’t.

Fixes

As with anything on the web, there are ways to work around these annoyances.

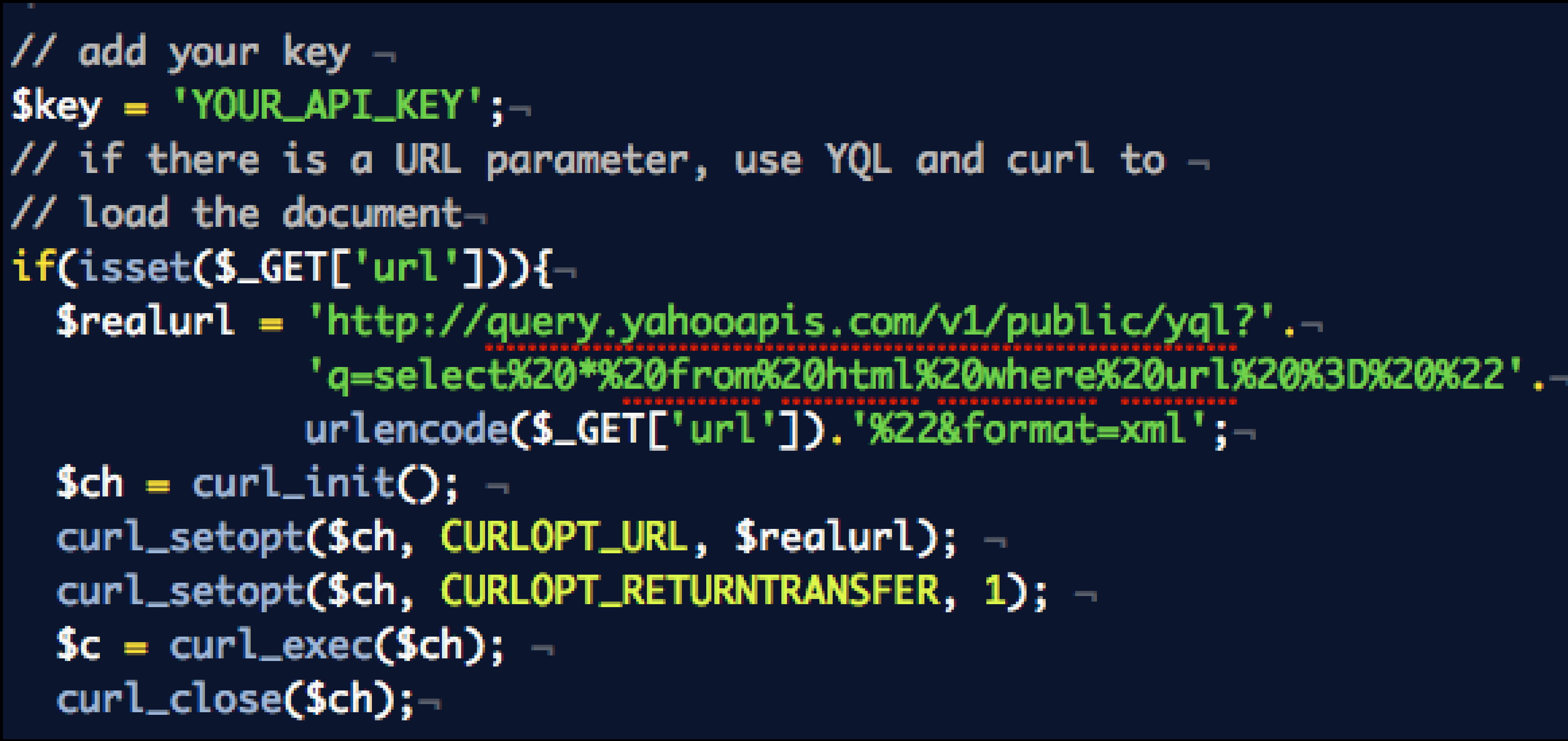


select * from html where url='http://news.bbc.co.uk/'

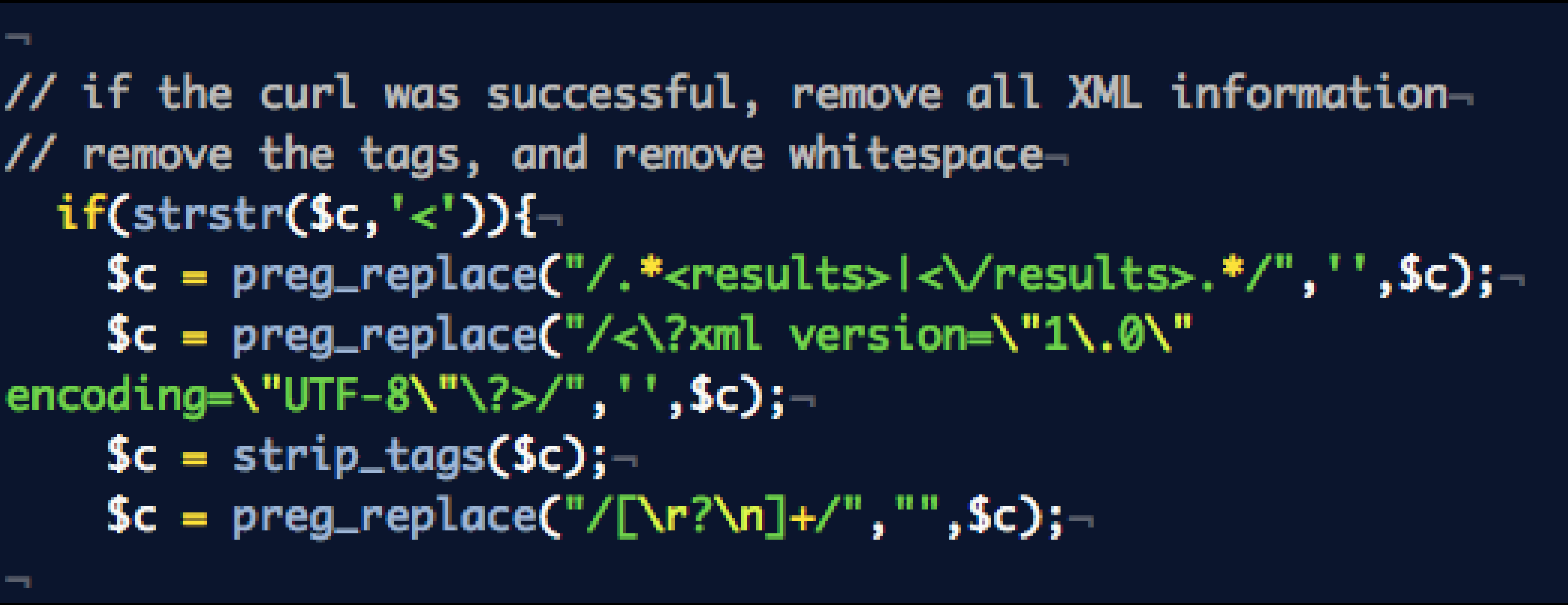
```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns:yahoo="http://www.yahooapis.com/v1/base.rng" yahoo:count="1" yahoo:created="2009-07-05T02:25:2
    <diagnostics>
        <publiclyCallable>true</publiclyCallable>
        <url execution-time="1"><![CDATA[http://datatables.org/alltables.env]]></url>
        <url execution-time="543"><![CDATA[http://news.bbc.co.uk/robots.txt]]></url>
        <url execution-time="748"><![CDATA[http://news.bbc.co.uk/]]></url>
        <user-time>1319</user-time>
        <service-time>1292</service-time>
        <build-version>1949</build-version>
    </diagnostics>
    <results>
        <body id="body">
            <div class="centerbody">
                <script type="text/javascript" xml:space="preserve">
//<![CDATA[ if(BBC.adverts){ BBC.adverts.setPageVersion('4'); } //]]>
</script>
                <div class="leaderboardcontainer">
                    <div class="bbccom_display_none" id="bbccom_leaderboard">
```

YQL to fix the web.

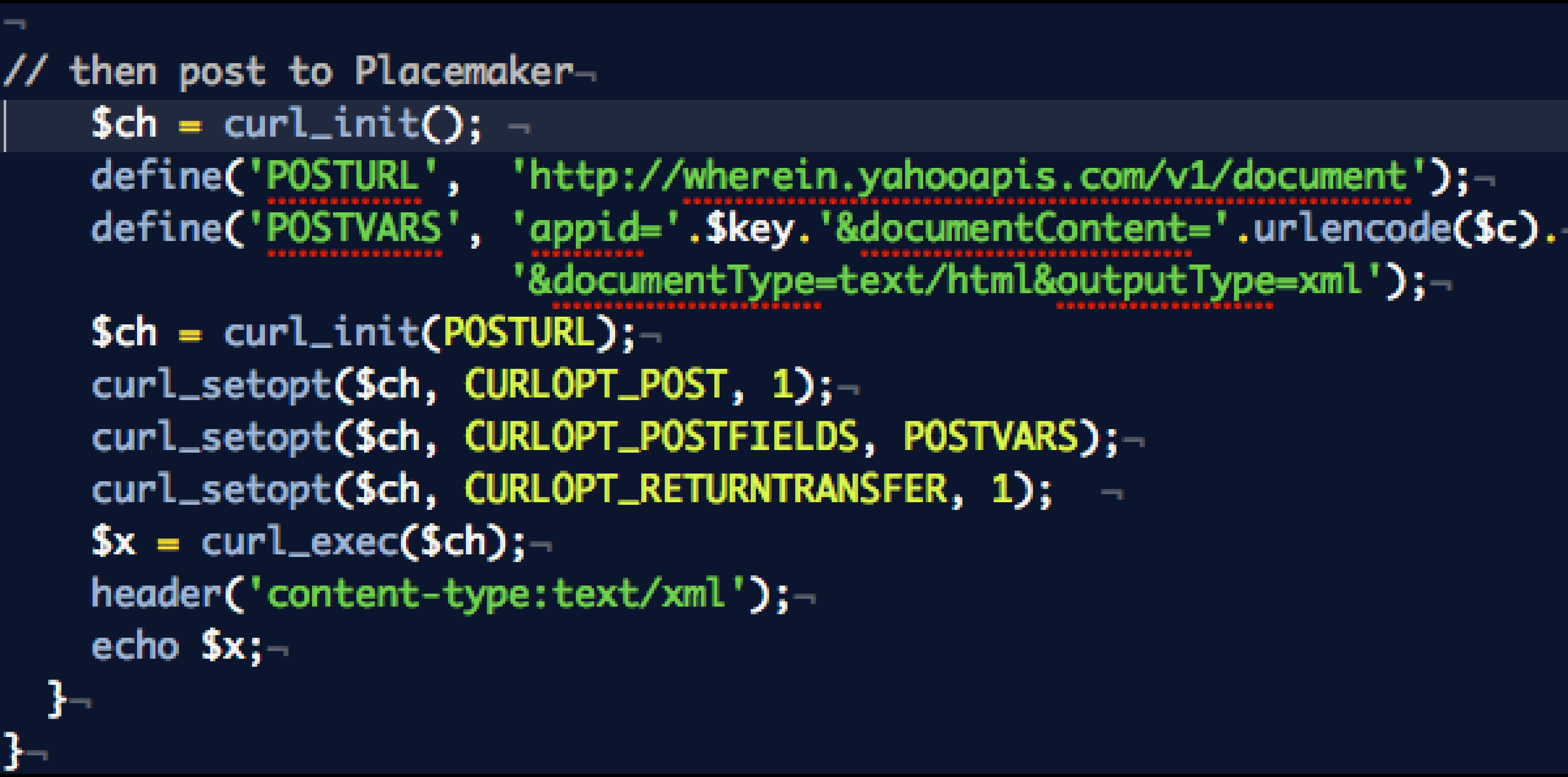
YQL has an HTML parser which doesn’t mind bad encodings and runs results through HTML tidy to clean it up.



```
// add your key
$key = 'YOUR_API_KEY';
// if there is a URL parameter, use YQL and curl to
// load the document
if(isset($_GET['url'])){
  $realurl = 'http://query.yahooapis.com/v1/public/yql?'.
             'q=select%20*%20from%20html%20where%20url%20%3D%20%22'.
             urlencode($_GET['url']).'%22&format=xml';
  $ch = curl_init();
  curl_setopt($ch, CURLOPT_URL, $realurl);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
  $c = curl_exec($ch);
  curl_close($ch);
```

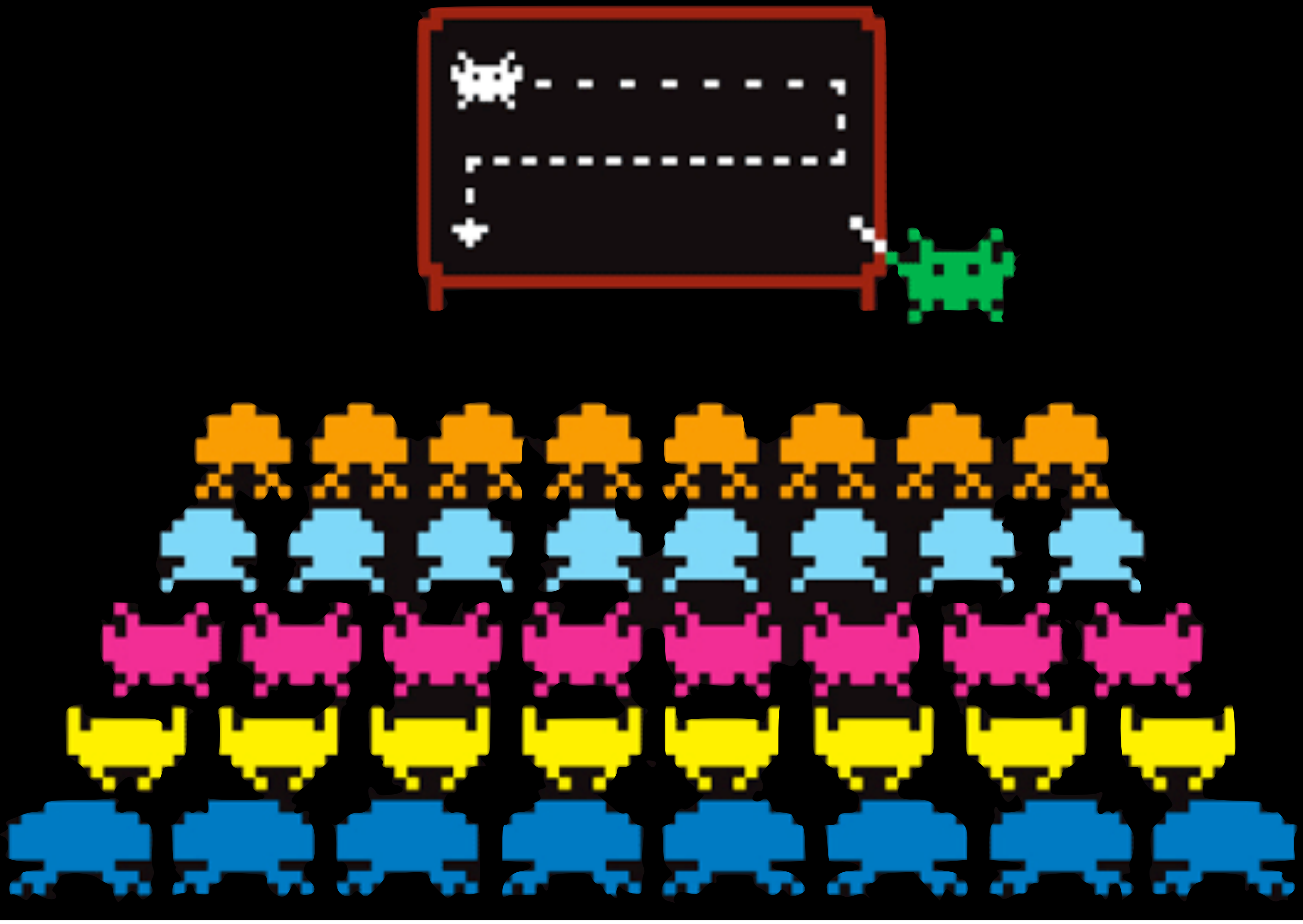


```
¬
// if the curl was successful, remove all XML information¬
// remove the tags, and remove whitespace¬
  if(strstr($c,'<')){¬
    $c = preg_replace("/.*<results>|<\/results>.*/",'',$c);¬
    $c = preg_replace("/<\?xml version=\"1\.0\"
encoding=\"UTF-8\"\?>/",'',$c);¬
    $c = strip_tags($c);¬
    $c = preg_replace("/[\r?\n]+/","",$c);¬
¬
```

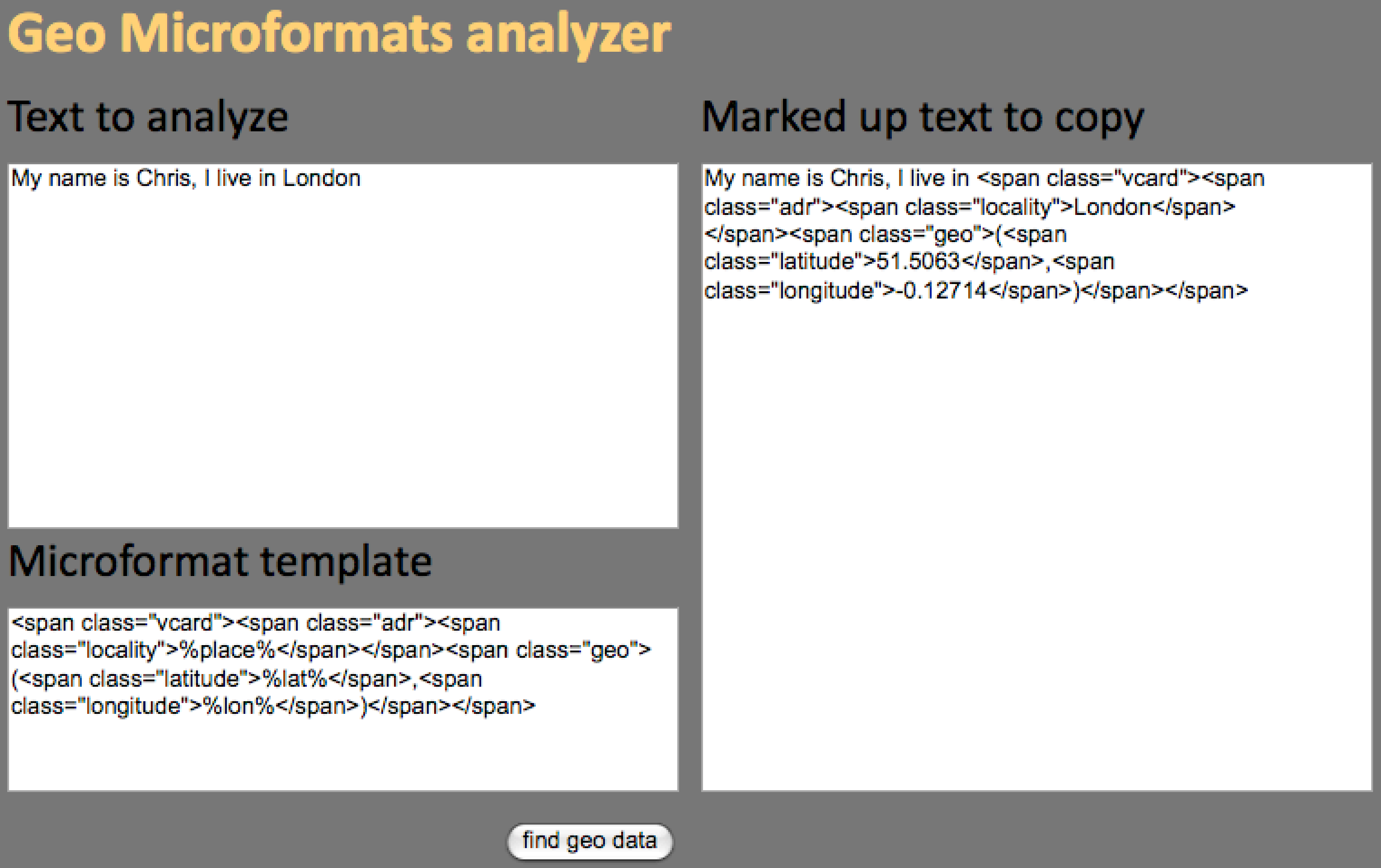


```
¬
// then post to Placemaker¬
    $ch = curl_init(); ¬
    define('POSTURL',  'http://wherein.yahooapis.com/v1/document');¬
    define('POSTVARS', 'appid='.$key.'&documentContent='.urlencode($c).
                       '&documentType=text/html&outputType=xml');¬
    $ch = curl_init(POSTURL);¬
    curl_setopt($ch, CURLOPT_POST, 1);¬
    curl_setopt($ch, CURLOPT_POSTFIELDS, POSTVARS);¬
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  ¬
    $x = curl_exec($ch);¬
    header('content-type:text/xml');¬
    echo $x;¬
  }¬
}¬
```

Connecting places and references



If we want to use places and bring them back into the original text we need to go through a small conversion process.



Geo Microformats analyzer

Text to analyze

My name is Chris, I live in London

Microformat template

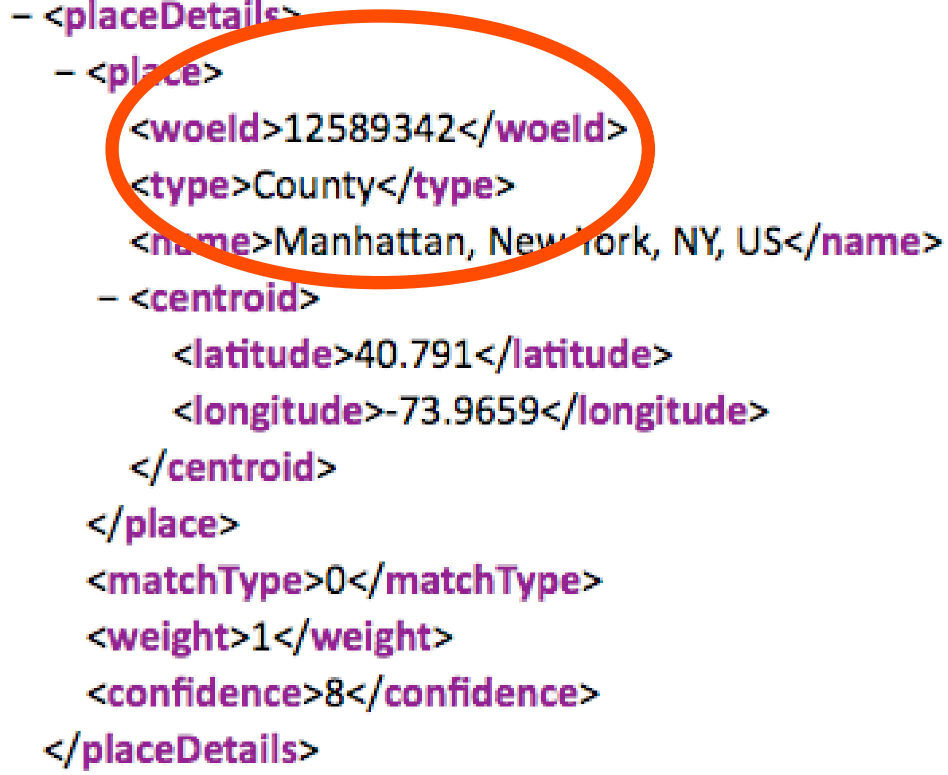
```
<span class="vcard"><span class="adr"><span
class="locality">%place%</span></span><span class="geo">
(<span class="latitude">%lat%</span>,<span
class="longitude">%lon%</span>)</span></span>
```

Marked up text to copy

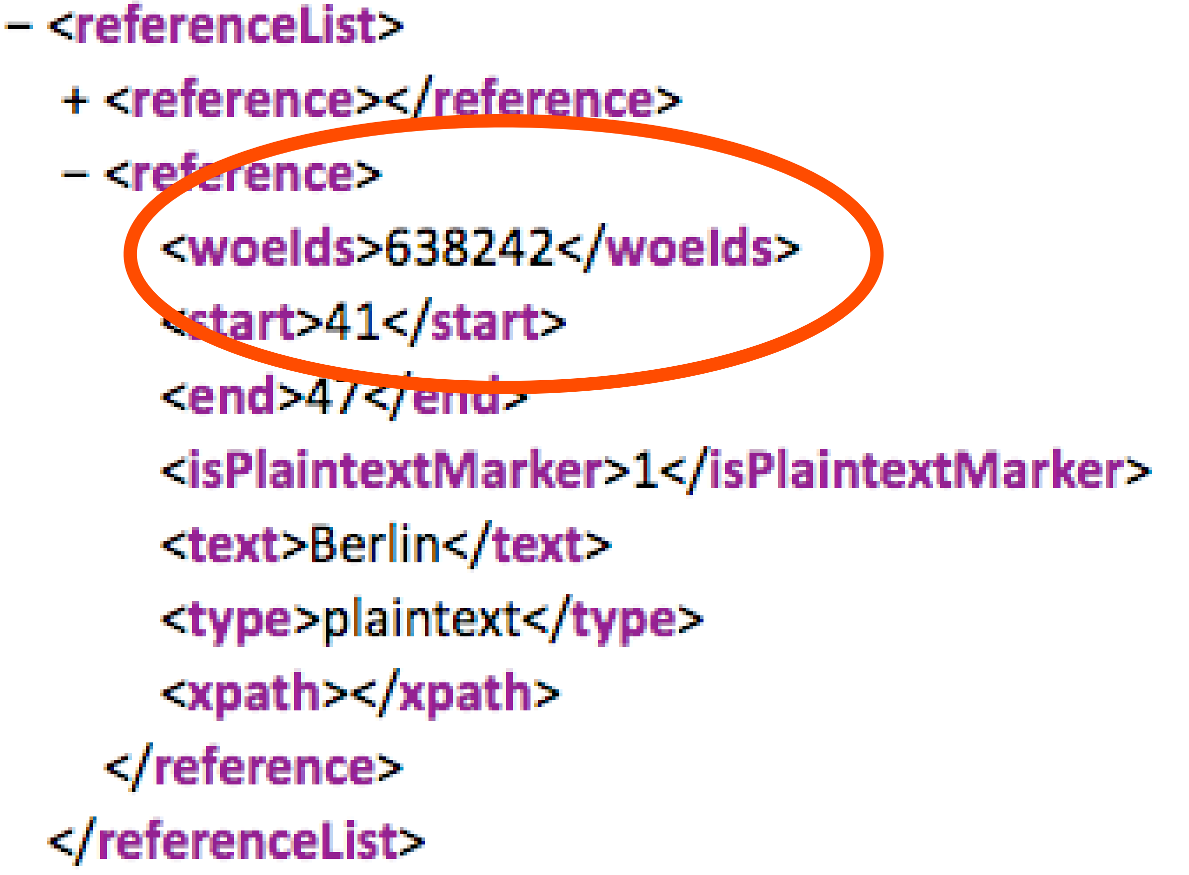
```
My name is Chris, I live in <span class="vcard"><span
class="adr"><span class="locality">London</span>
</span><span class="geo">(<span
class="latitude">51.5063</span>,<span
class="longitude">-0.12714</span>)</span></span>
```

http://isithackday.com/hacks/placemaker/simpletext.php

Let’s build a simple tool that inserts Geo microformats into a text using Placemaker.

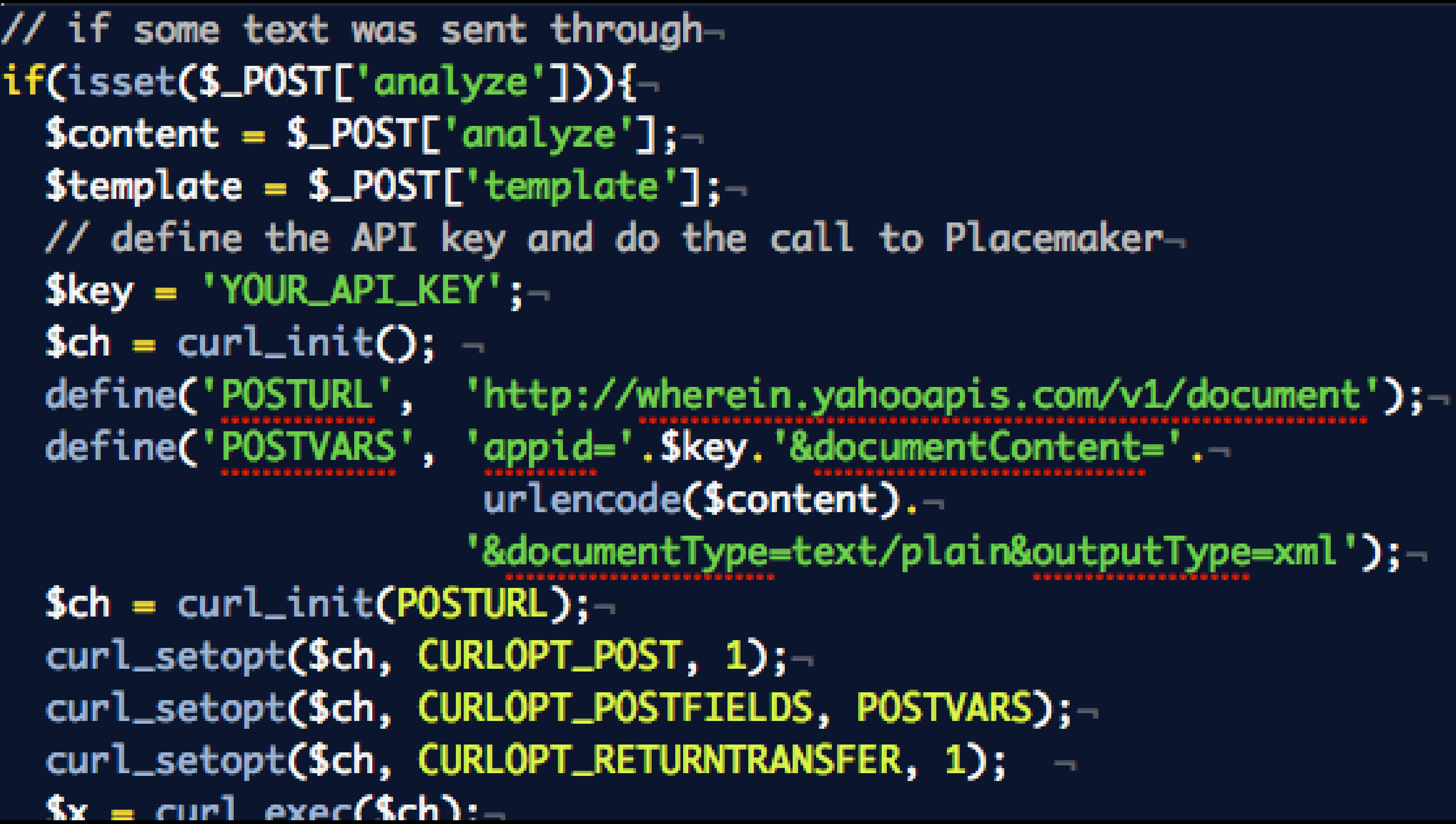


```
- <placeDetails>
  - <place>
      <woeId>12589342</woeId>
      <type>County</type>
      <name>Manhattan, New York, NY, US</name>
    - <centroid>
        <latitude>40.791</latitude>
        <longitude>-73.9659</longitude>
      </centroid>
    </place>
    <matchType>0</matchType>
    <weight>1</weight>
    <confidence>8</confidence>
  </placeDetails>
```

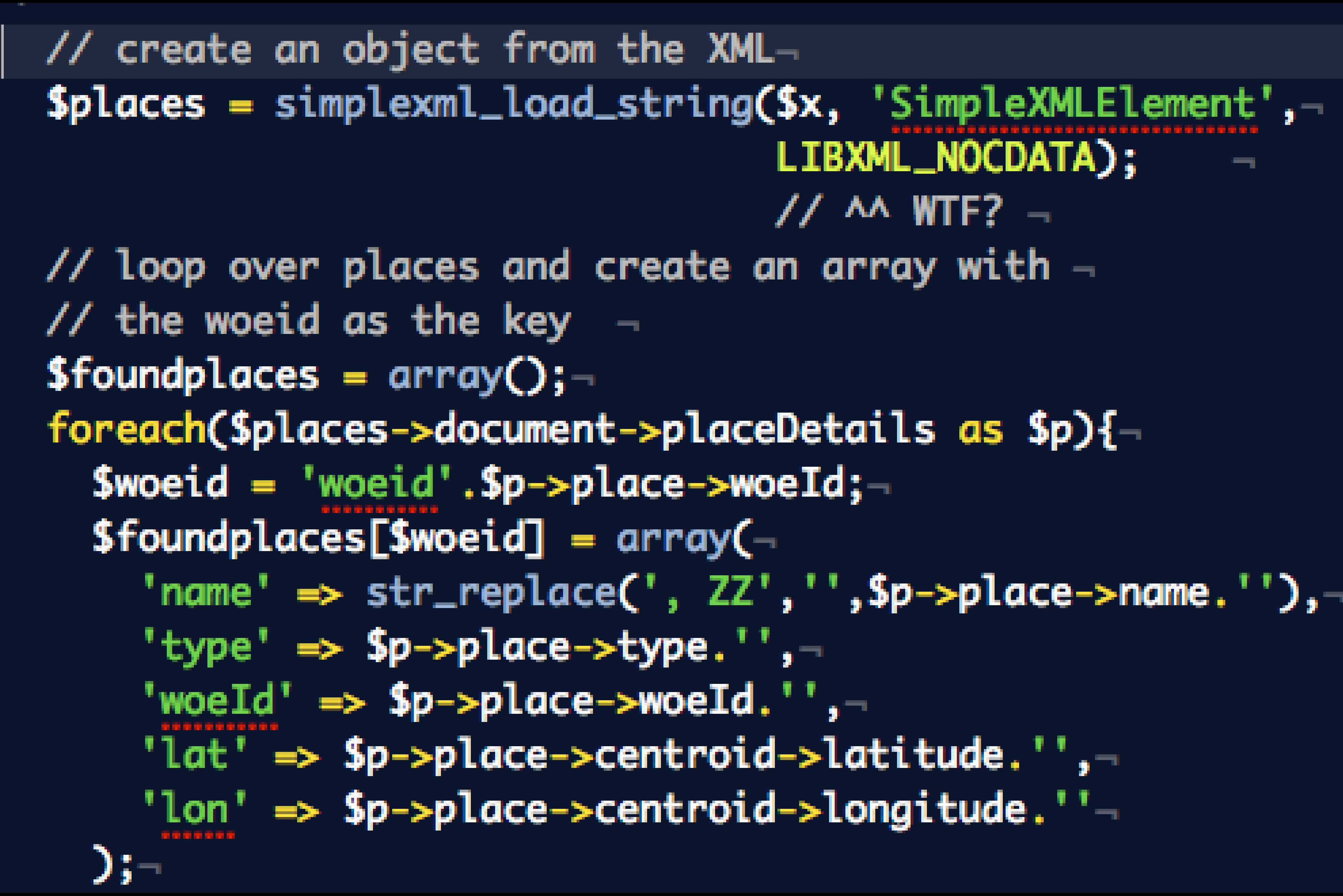


```
- <referenceList>
  + <reference></reference>
  - <reference>
      <woeIds>638242</woeIds>
      <start>41</start>
      <end>47</end>
      <isPlaintextMarker>1</isPlaintextMarker>
      <text>Berlin</text>
      <type>plaintext</type>
      <xpath></xpath>
    </reference>
  </referenceList>
```

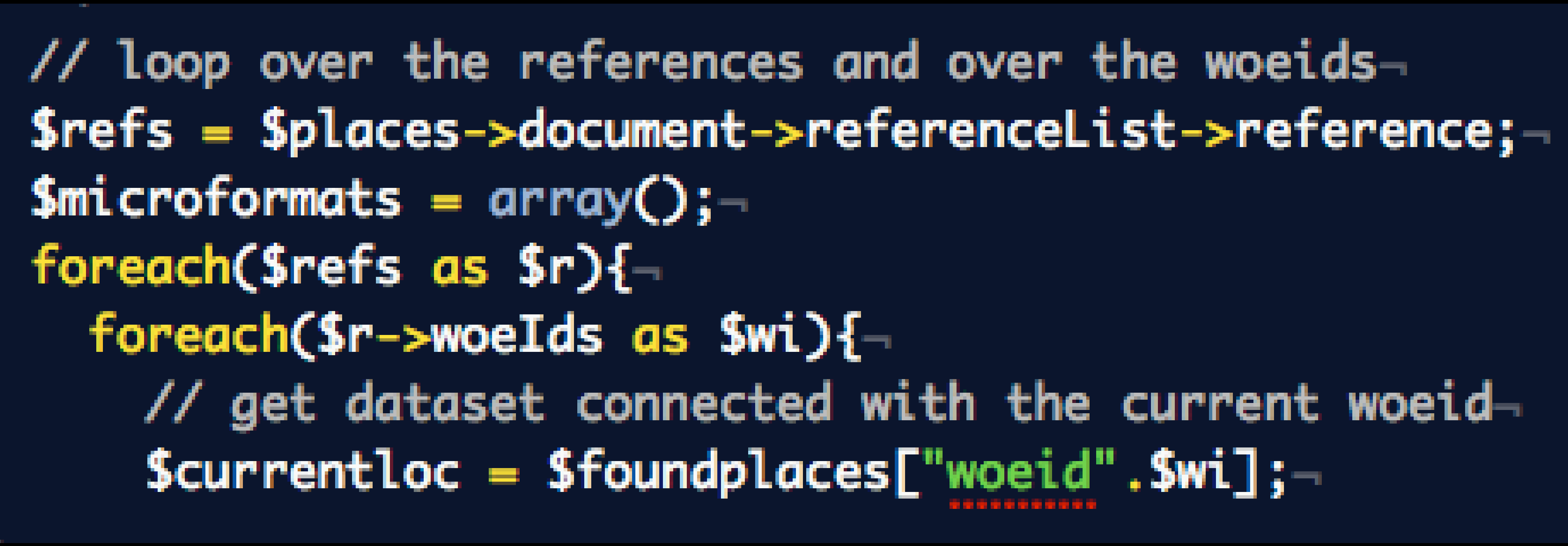
Both Places and References have woeids. The only issue is that references may have more than one :-(



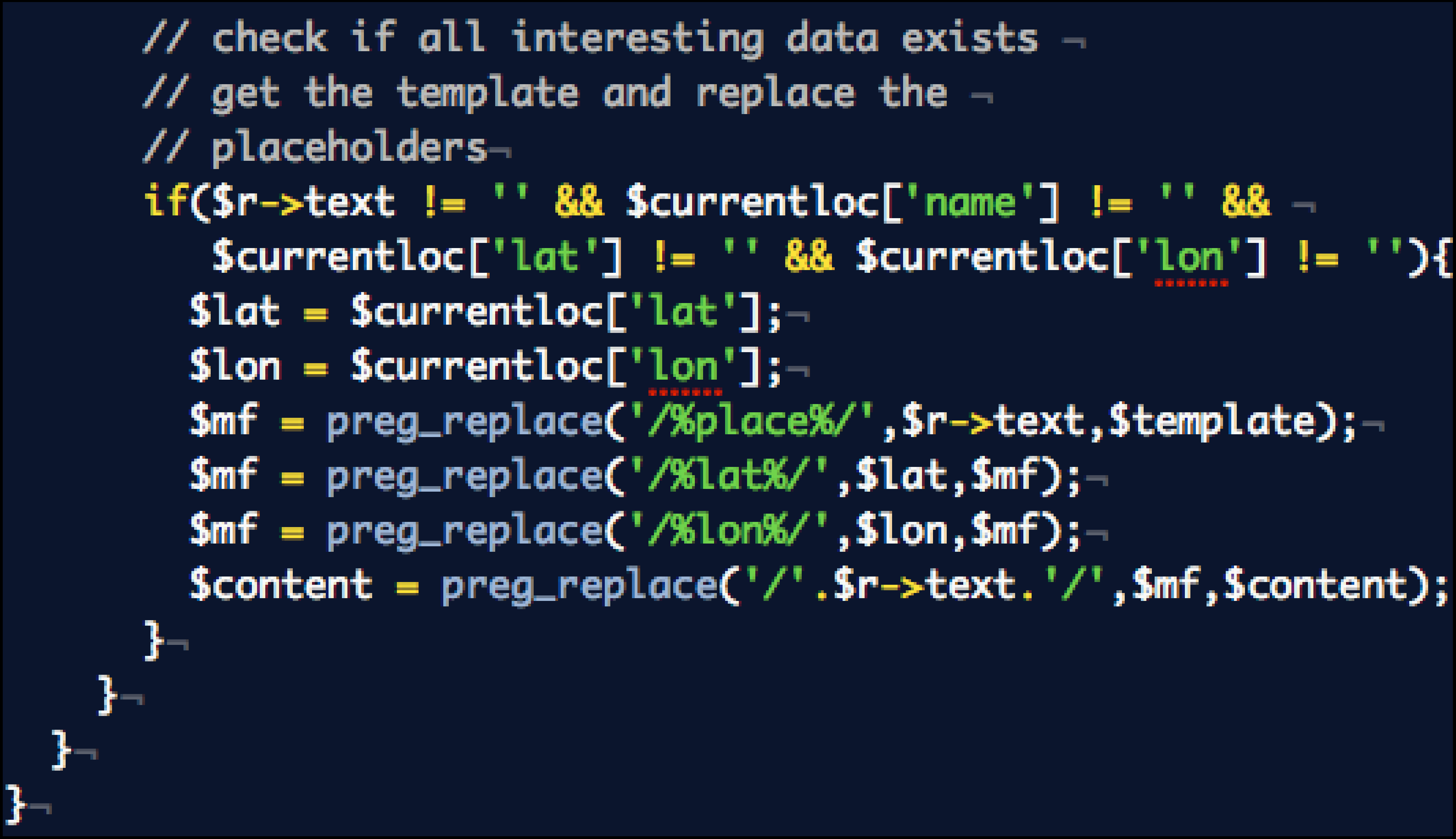
```
// if some text was sent through
if(isset($_POST['analyze'])){
  $content = $_POST['analyze'];
  $template = $_POST['template'];
  // define the API key and do the call to Placemaker
  $key = 'YOUR_API_KEY';
  $ch = curl_init(); 
  define('POSTURL',  'http://wherein.yahooapis.com/v1/document');
  define('POSTVARS', 'appid='.$key.'&documentContent='.
                      urlencode($content).
                     '&documentType=text/plain&outputType=xml');
  $ch = curl_init(POSTURL);
  curl_setopt($ch, CURLOPT_POST, 1);
  curl_setopt($ch, CURLOPT_POSTFIELDS, POSTVARS);
  curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);  
  $x = curl_exec($ch);
```



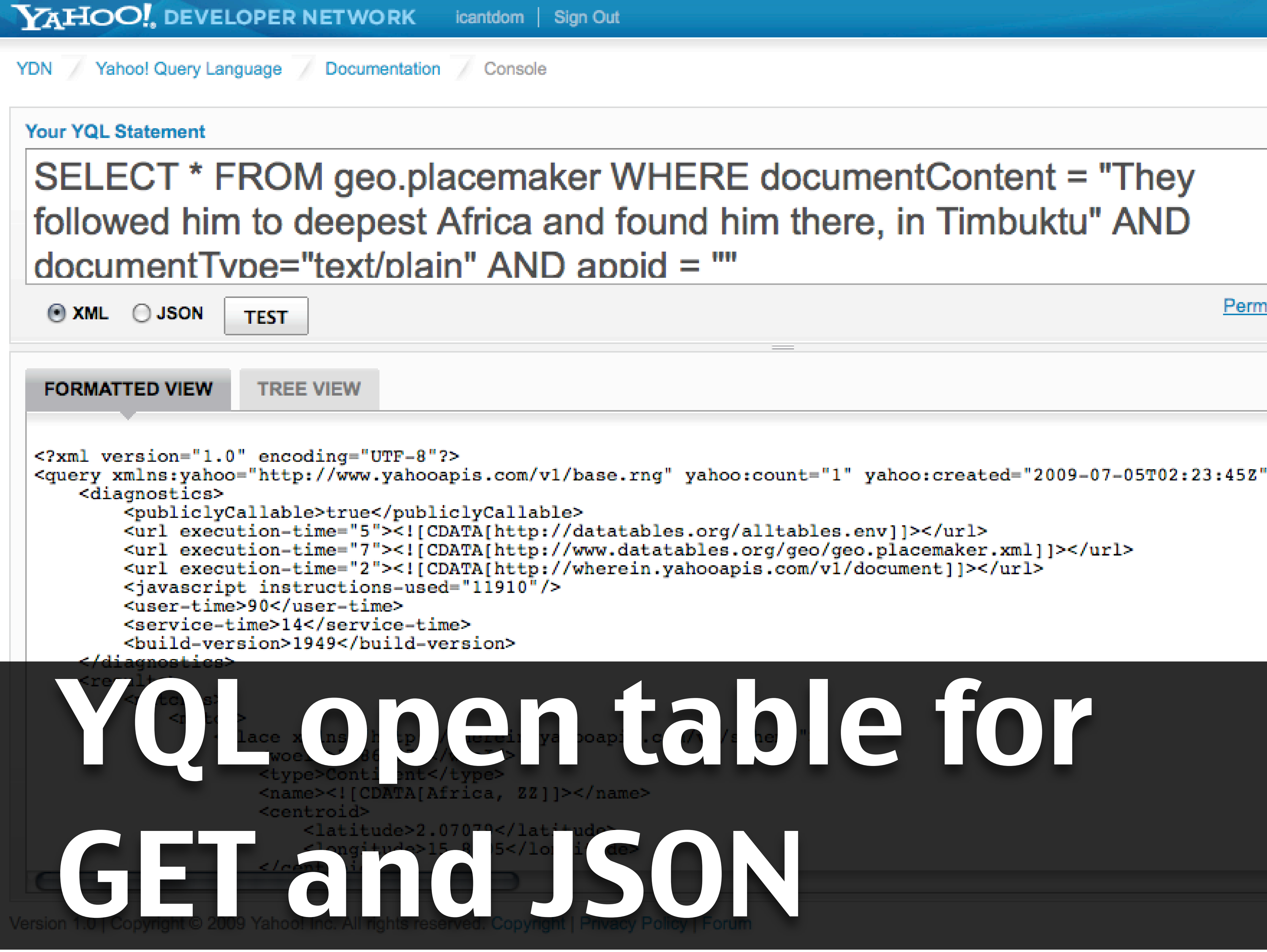
```php
// create an object from the XML
$places = simplexml_load_string($x, 'SimpleXMLElement',
                                LIBXML_NOCDATA);
                                // ^^ WTF?
// loop over places and create an array with
// the woeid as the key
$foundplaces = array();
foreach($places->document->placeDetails as $p){
  $woeid = 'woeid'.$p->place->woeId;
  $foundplaces[$woeid] = array(
    'name' => str_replace(', ZZ','',$p->place->name.''),
    'type' => $p->place->type.'',
    'woeId' => $p->place->woeId.'',
    'lat' => $p->place->centroid->latitude.'',
    'lon' => $p->place->centroid->longitude.''
  );
```



```
// loop over the references and over the woeids¬
$refs = $places->document->referenceList->reference;¬
$microformats = array();¬
foreach($refs as $r){¬
  foreach($r->woeIds as $wi){¬
    // get dataset connected with the current woeid¬
    $currentloc = $foundplaces["woeid".$wi];¬
```



```
      // check if all interesting data exists
      // get the template and replace the
      // placeholders
      if($r->text != '' && $currentloc['name'] != '' &&
         $currentloc['lat'] != '' && $currentloc['lon'] != ''){
        $lat = $currentloc['lat'];
        $lon = $currentloc['lon'];
        $mf = preg_replace('/%place%/',$r->text,$template);
        $mf = preg_replace('/%lat%/',$lat,$mf);
        $mf = preg_replace('/%lon%/',$lon,$mf);
        $content = preg_replace('/'.$r->text.'/',$mf,$content);
      }
    }
  }
}
```


YAHOO! DEVELOPER NETWORK
icantdom | Sign Out
YDN / Yahoo! Query Language / Documentation / Console
Your YQL Statement

SELECT * FROM geo.placemaker WHERE documentContent = "They followed him to deepest Africa and found him there, in Timbuktu" AND documentType="text/plain" AND appid = ""

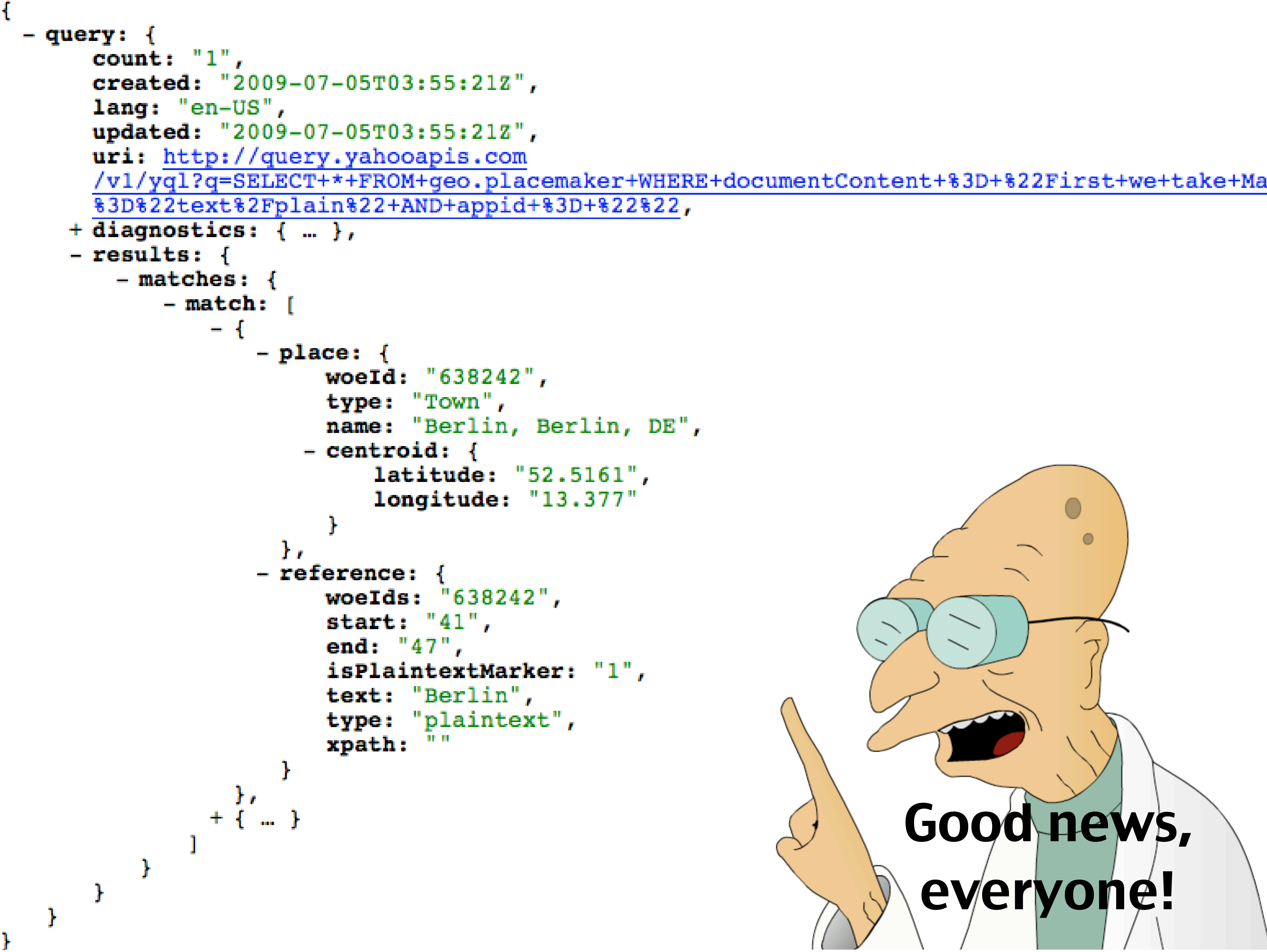
● XML ○ JSON TEST
Perm
FORMATTED VIEW
TREE VIEW

```
<?xml version="1.0" encoding="UTF-8"?>
<query xmlns:yahoo="http://www.yahooapis.com/v1/base.rng" yahoo:count="1" yahoo:created="2009-07-05T02:23:45Z"
    <diagnostics>
        <publiclyCallable>true</publiclyCallable>
        <url execution-time="5"><![CDATA[http://datatables.org/alltables.env]]></url>
        <url execution-time="7"><![CDATA[http://www.datatables.org/geo/geo.placemaker.xml]]></url>
        <url execution-time="2"><![CDATA[http://wherein.yahooapis.com/v1/document]]></url>
        <javascript instructions-used="11910"/>
        <user-time>90</user-time>
        <service-time>14</service-time>
        <build-version>1949</build-version>
    </diagnostics>
                    <type>Continent</type>
                    <name><![CDATA[Africa, ZZ]]></name>
                    <centroid>
```

Version 1.0 | Copyright © 2009 Yahoo! Inc. All rights reserved. Copyright | Privacy Policy | Forum

YQL open table for GET and JSON

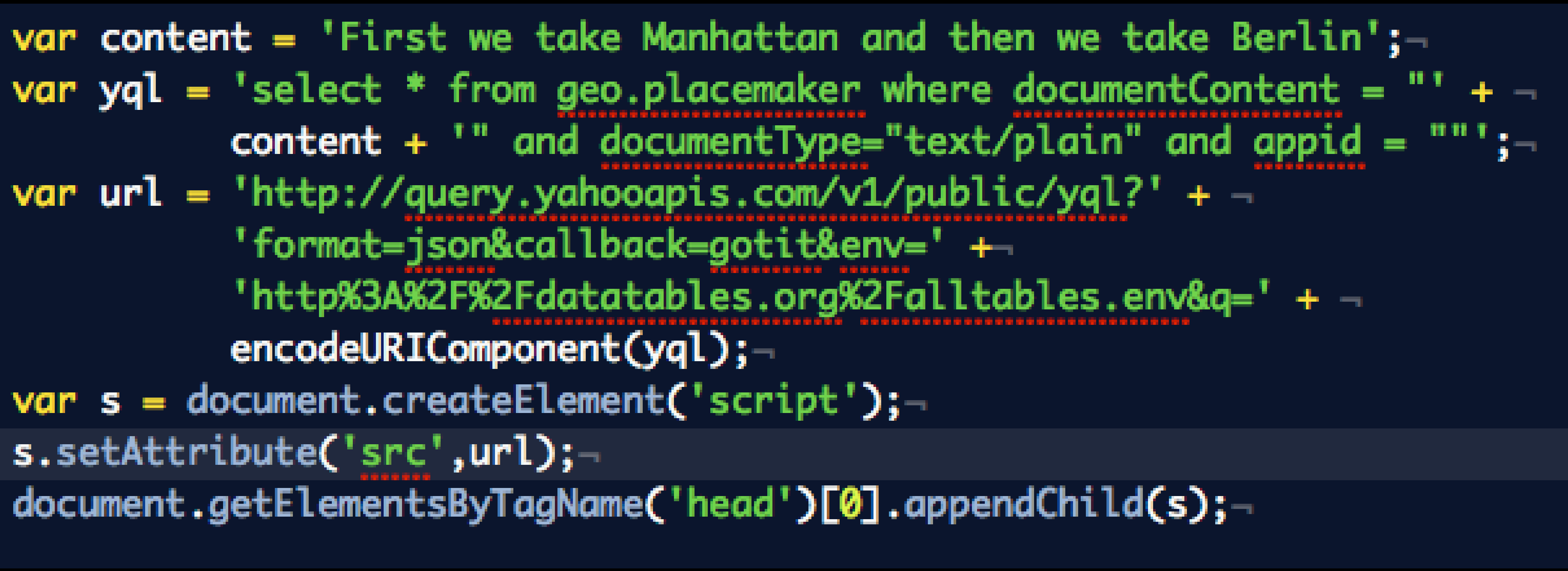
Another thing YQL allows developers to do is to extend it with own open tables that run JavaScript conversions on the server side. One of those is the YQL open table which does all the things Placemaker does on the server and offers JSON output.



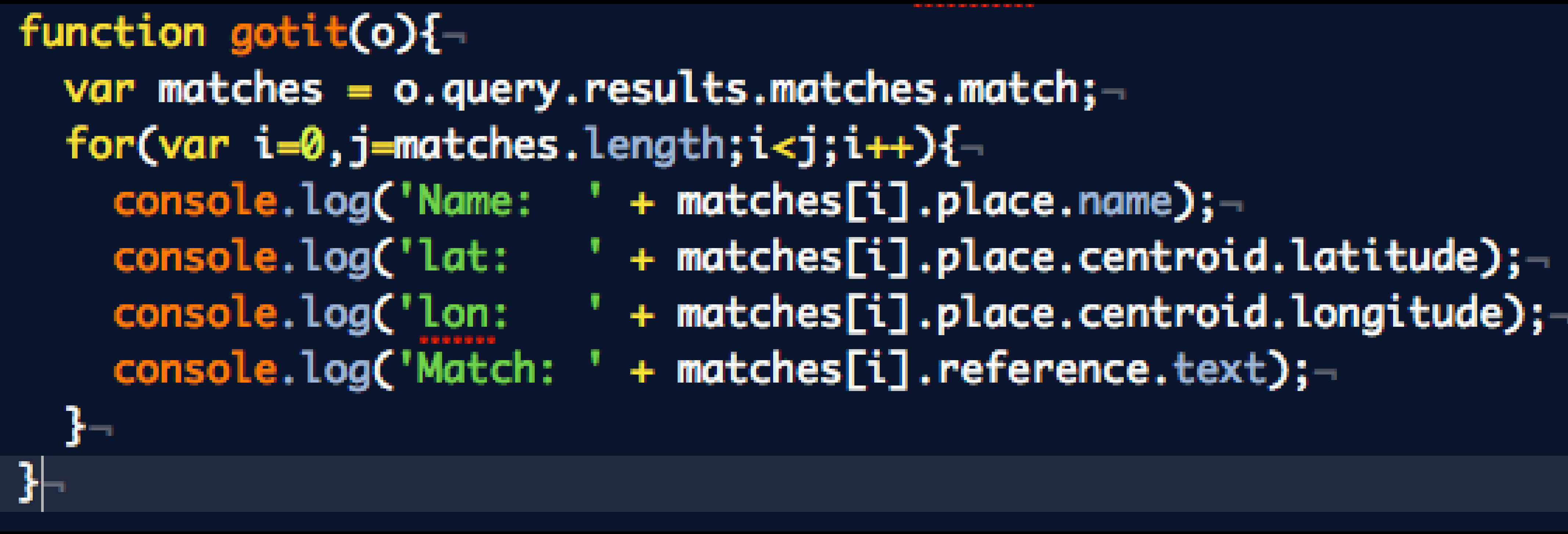
```
{
 - query: {
       count: "1",
       created: "2009-07-05T03:55:21Z",
       lang: "en-US",
       updated: "2009-07-05T03:55:21Z",
       uri: http://query.yahooapis.com
       /v1/yql?q=SELECT+*+FROM+geo.placemaker+WHERE+documentContent+%3D+%22First+we+take+Ma
       %3D%22text%2Fplain%22+AND+appid+%3D+%22%22,
     + diagnostics: { … },
     - results: {
         - matches: {
             - match: [
                 - {
                     - place: {
                           woeId: "638242",
                           type: "Town",
                           name: "Berlin, Berlin, DE",
                         - centroid: {
                               latitude: "52.5161",
                               longitude: "13.377"
                           }
                       },
                     - reference: {
                           woeIds: "638242",
                           start: "41",
                           end: "47",
                           isPlaintextMarker: "1",
                           text: "Berlin",
                           type: "plaintext",
                           xpath: ""
                       }
                   },
                 + { … }
               ]
           }
       }
   }
}
```

The really nice thing of the open YQL table is that it already puts the places and references together.

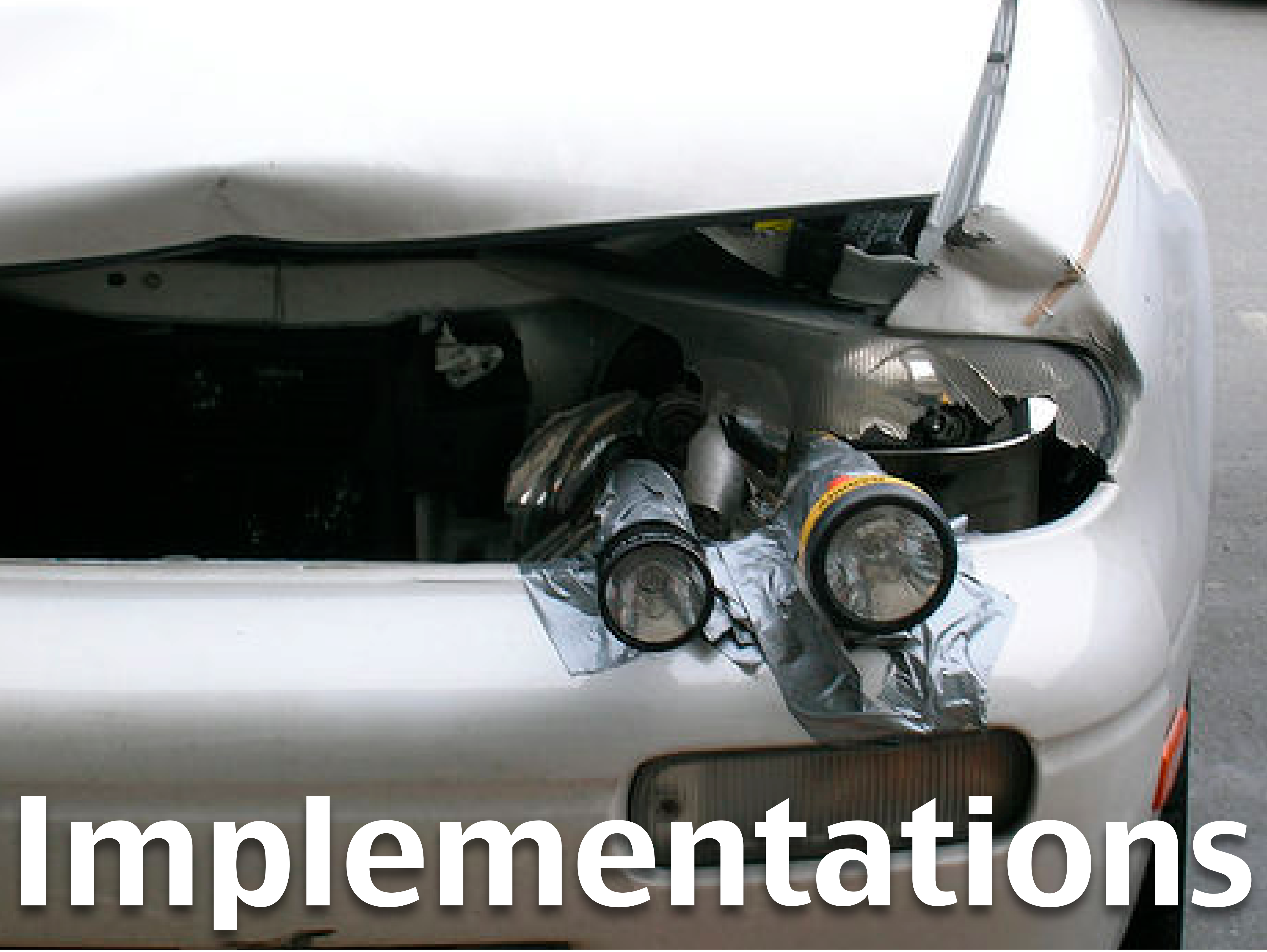
http://isithackday.com/hacks/placemaker/placemaker-javascript.html



```
var content = 'First we take Manhattan and then we take Berlin';
var yql = 'select * from geo.placemaker where documentContent = "' +
          content + '" and documentType="text/plain" and appid = ""';
var url = 'http://query.yahooapis.com/v1/public/yql?' +
          'format=json&callback=gotit&env=' +
          'http%3A%2F%2Fdatatables.org%2Falltables.env&q=' +
          encodeURIComponent(yql);
var s = document.createElement('script');
s.setAttribute('src',url);
document.getElementsByTagName('head')[0].appendChild(s);
```

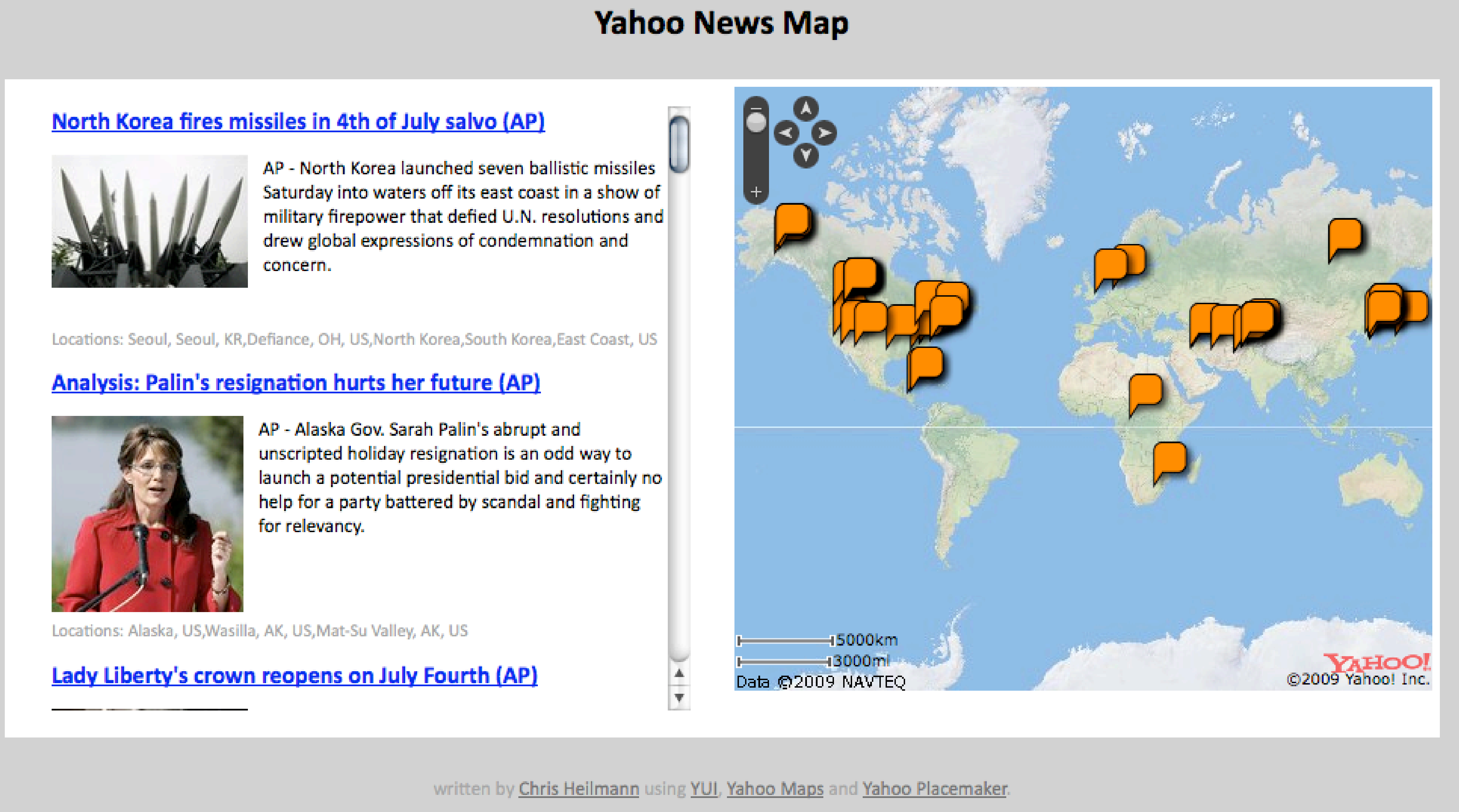


```
function gotit(o){
  var matches = o.query.results.matches.match;
  for(var i=0,j=matches.length;i<j;i++){
    console.log('Name:  ' + matches[i].place.name);
    console.log('lat:   ' + matches[i].place.centroid.latitude);
    console.log('lon:   ' + matches[i].place.centroid.longitude);
    console.log('Match: ' + matches[i].reference.text);
  }
}
```



Implementations

Now let’s look at a few things I’ve built with Placemaker.



Yahoo News Map

[North Korea fires missiles in 4th of July salvo (AP)]

AP - North Korea launched seven ballistic missiles Saturday into waters off its east coast in a show of military firepower that defied U.N. resolutions and drew global expressions of condemnation and concern.

Locations: Seoul, Seoul, KR,Defiance, OH, US,North Korea,South Korea,East Coast, US

[Analysis: Palin's resignation hurts her future (AP)]

AP - Alaska Gov. Sarah Palin's abrupt and unscripted holiday resignation is an odd way to launch a potential presidential bid and certainly no help for a party battered by scandal and fighting for relevancy.

Locations: Alaska, US,Wasilla, AK, US,Mat-Su Valley, AK, US

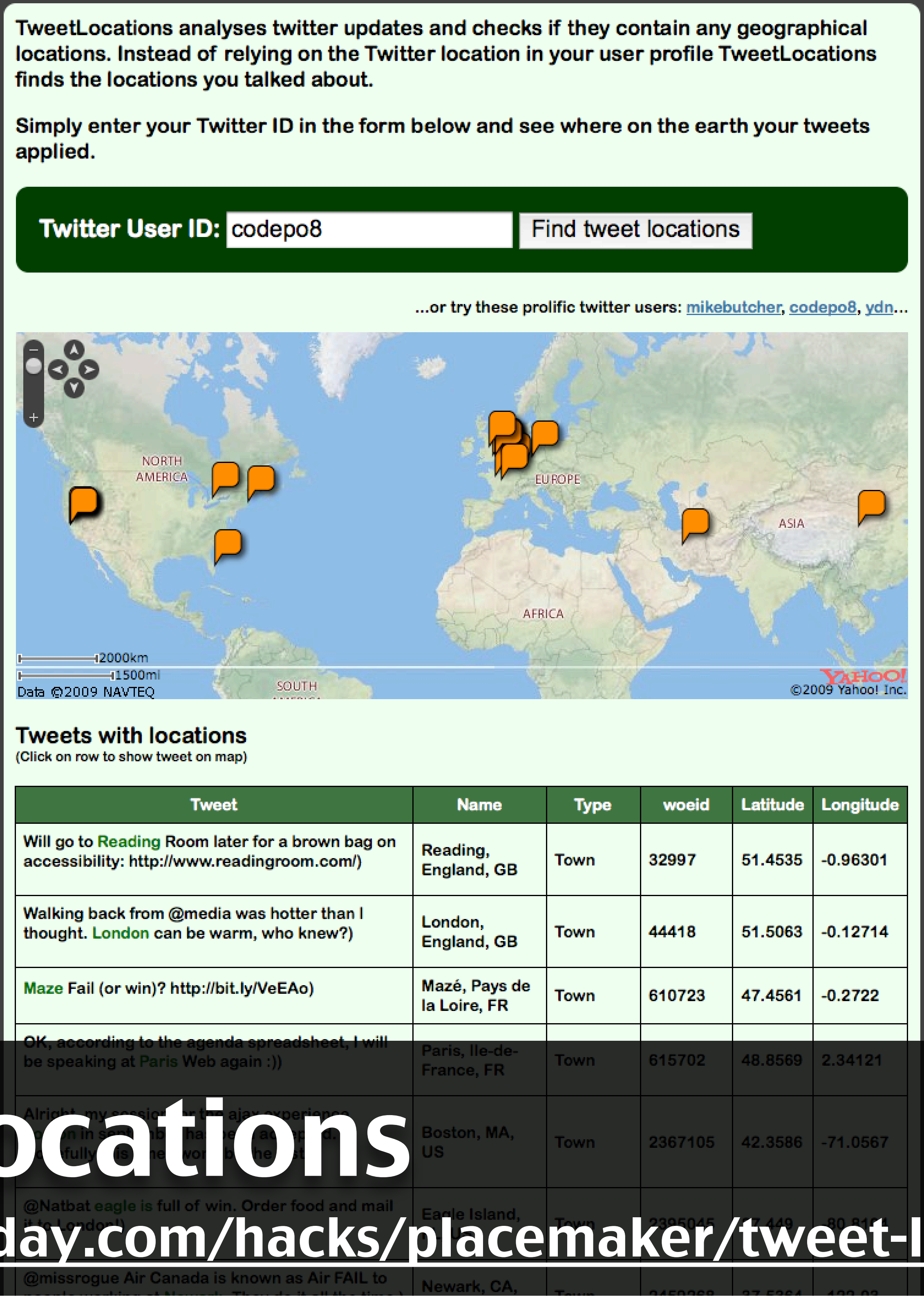
[Lady Liberty's crown reopens on July Fourth (AP)]

written by Chris Heilmann using YUI, Yahoo Maps and Yahoo Placemaker.

Yahoo News Map

http://isithackday.com/hacks/placemaker/map.php

Yahoo News Map uses the Yahoo RSS feed run through Placemaker to show news on a map and allow to navigate with the map.



TweetLocations analyses twitter updates and checks if they contain any geographical locations. Instead of relying on the Twitter location in your user profile TweetLocations finds the locations you talked about.

Simply enter your Twitter ID in the form below and see where on the earth your tweets applied.

Twitter User ID: codepo8 Find tweet locations

...or try these prolific twitter users: mikebutcher, codepo8, ydn...

Tweets with locations

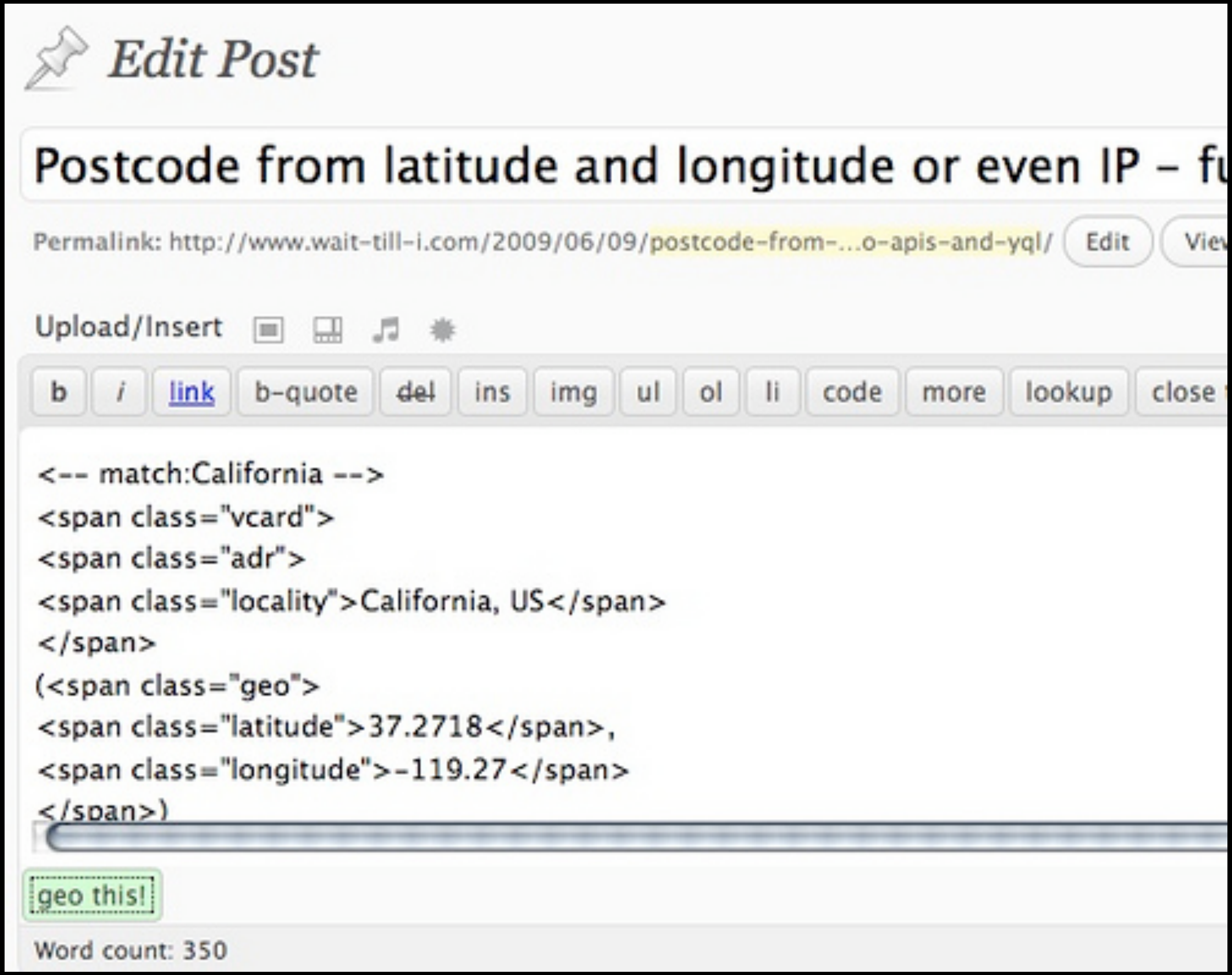
(Click on row to show tweet on map)

Tweet	Name	Type	woeid	Latitude	Longitude
Will go to Reading Room later for a brown bag on accessibility: http://www.readingroom.com/)	Reading, England, GB	Town	32997	51.4535	-0.96301
Walking back from @media was hotter than I thought. London can be warm, who knew?)	London, England, GB	Town	44418	51.5063	-0.12714
Maze Fail (or win)? http://bit.ly/VeEAo)	Mazé, Pays de la Loire, FR	Town	610723	47.4561	-0.2722
OK, according to the agenda spreadsheet, I will be speaking at Paris Web again :))	Paris, Ile-de-France, FR	Town	615702	48.8569	2.34121
Alright, my session for the ajax experience [illegible] in september has been accepted. [illegible]	Boston, MA, US	Town	2367105	42.3586	-71.0567
@Natbat eagle is full of win. Order food and mail it to London!)	Eagle Island, [illegible]	Town	2395045	47.449	-80.8101
@missrogue Air Canada is known as Air FAIL to [illegible]	Newark, CA,	Town	[illegible]	[illegible]	[illegible]

TweetLocations

http://isithackday.com/hacks/placemaker/tweet-locations.php

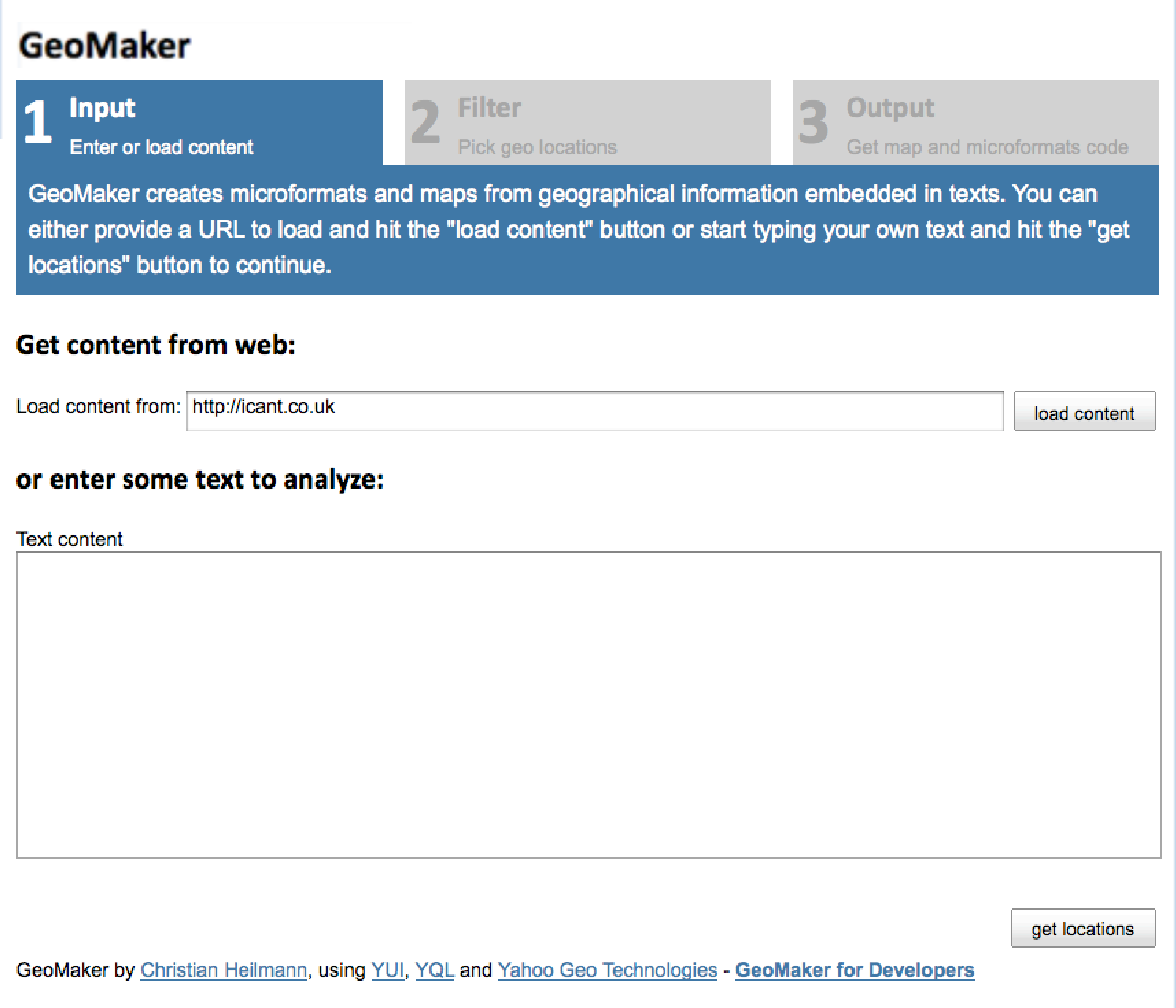
Tweetlocations shows a map of your latest tweets.


Edit Post
Postcode from latitude and longitude or even IP – fu
Permalink: http://www.wait-till-i.com/2009/06/09/postcode-from-...o-apis-and-yql/
Edit
Viev
Upload/Insert
b
i
link
b-quote
del
ins
img
ul
ol
li
code
more
lookup
close
geo this!
Word count: 350

```
<-- match:California -->
<span class="vcard">
<span class="adr">
<span class="locality">California, US</span>
</span>
(<span class="geo">
<span class="latitude">37.2718</span>,
<span class="longitude">-119.27</span>
</span>)
```

Geo this! (Greasemonkey)

http://icanhaz.com/geothis



GeoMaker creates microformats and maps from geographical information embedded in texts. You can either provide a URL to load and hit the "load content" button or start typing your own text and hit the "get locations" button to continue.

Get content from web:

Load content from: http://icant.co.uk

or enter some text to analyze:

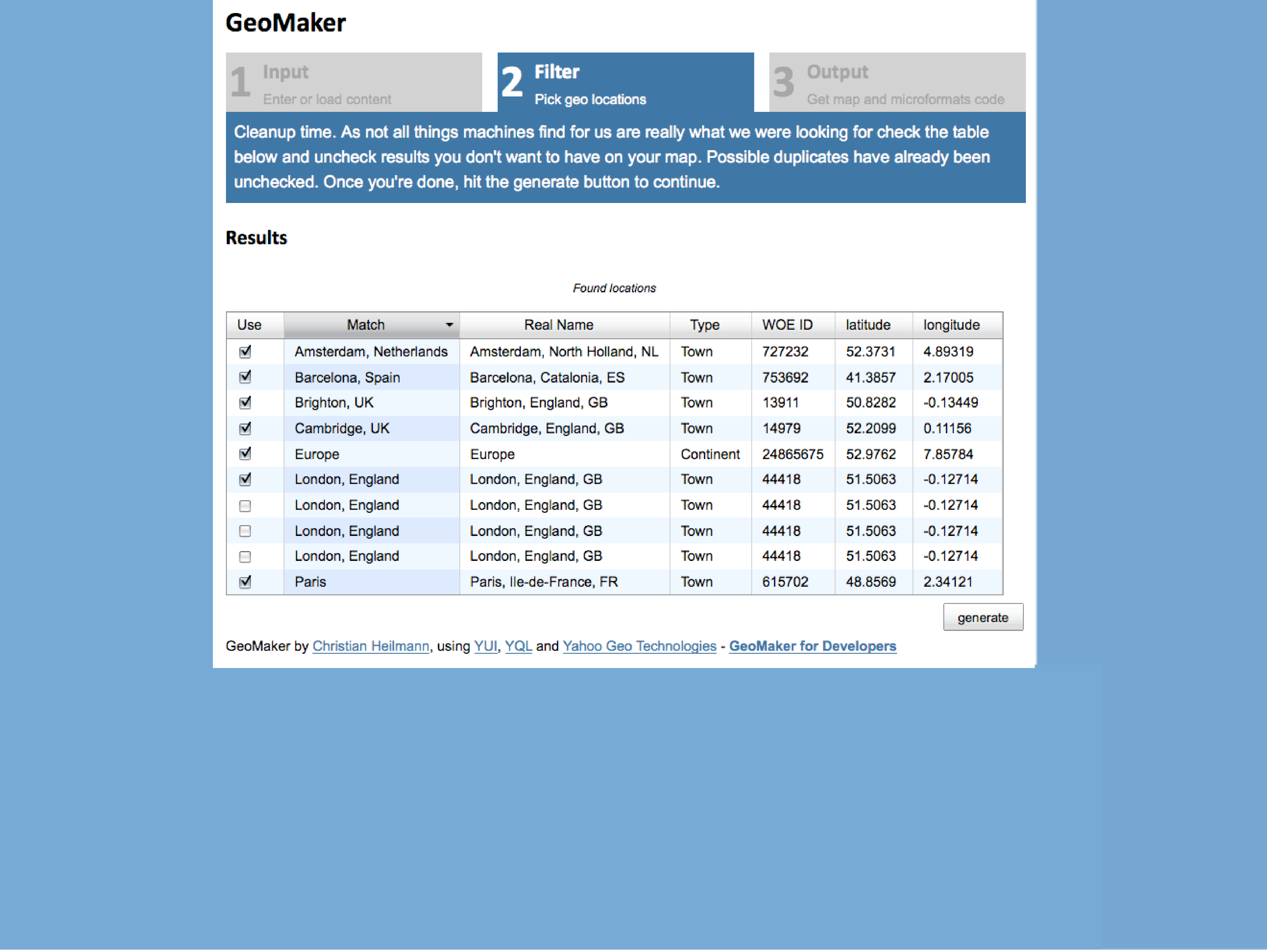
Text content

GeoMaker by Christian Heilmann, using YUI, YQL and Yahoo Geo Technologies - **GeoMaker for Developers**

GeoMaker

http://icant.co.uk/geomaker/

GeoMaker is a frontend to Placemaker that turns a URL or a text into a map.



GeoMaker

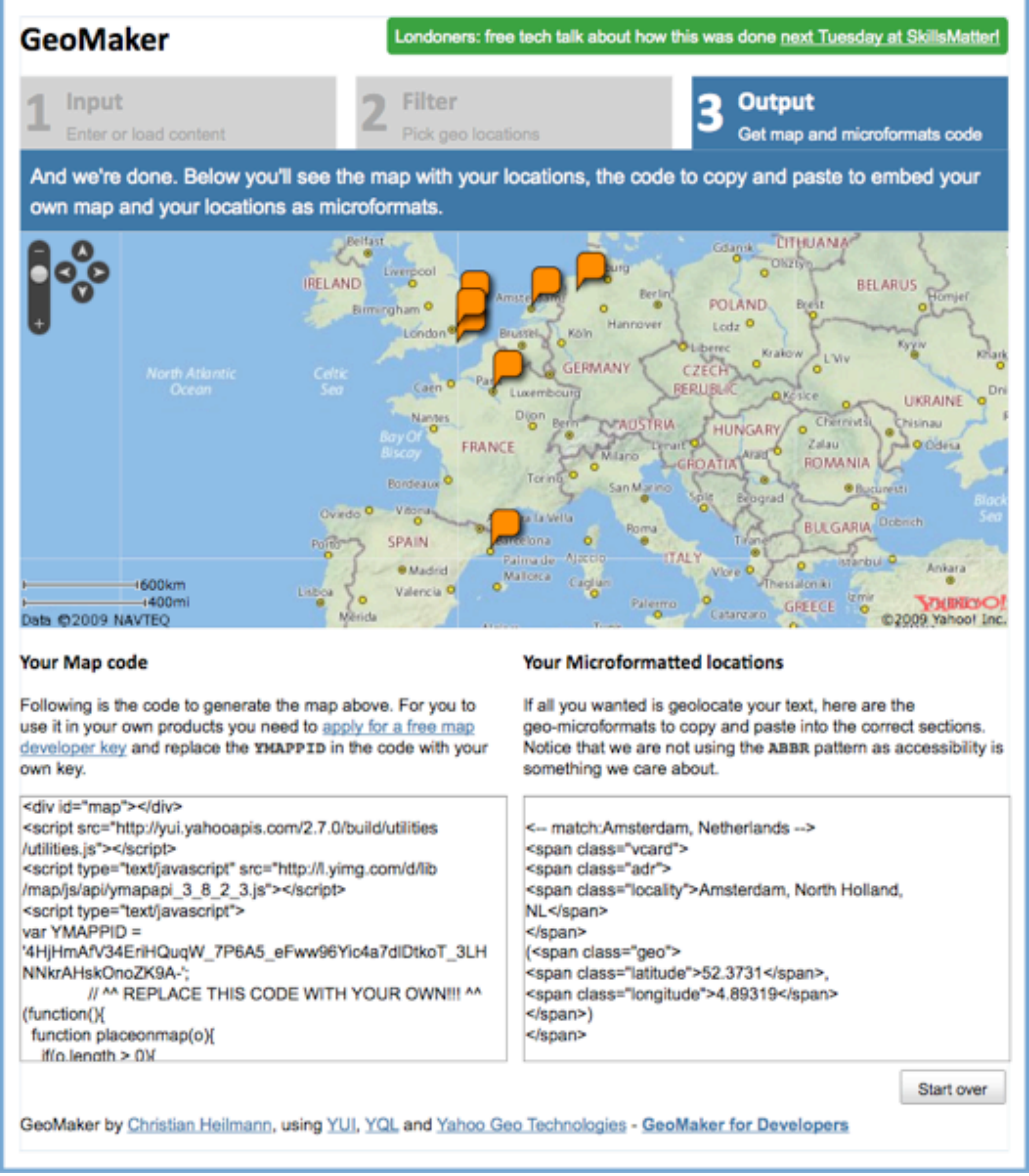
Cleanup time. As not all things machines find for us are really what we were looking for check the table below and uncheck results you don't want to have on your map. Possible duplicates have already been unchecked. Once you're done, hit the generate button to continue.

Results

Found locations

Use	Match	Real Name	Type	WOE ID	latitude	longitude
☑	Amsterdam, Netherlands	Amsterdam, North Holland, NL	Town	727232	52.3731	4.89319
☑	Barcelona, Spain	Barcelona, Catalonia, ES	Town	753692	41.3857	2.17005
☑	Brighton, UK	Brighton, England, GB	Town	13911	50.8282	-0.13449
☑	Cambridge, UK	Cambridge, England, GB	Town	14979	52.2099	0.11156
☑	Europe	Europe	Continent	24865675	52.9762	7.85784
☑	London, England	London, England, GB	Town	44418	51.5063	-0.12714
□	London, England	London, England, GB	Town	44418	51.5063	-0.12714
□	London, England	London, England, GB	Town	44418	51.5063	-0.12714
□	London, England	London, England, GB	Town	44418	51.5063	-0.12714
☑	Paris	Paris, Ile-de-France, FR	Town	615702	48.8569	2.34121

GeoMaker by Christian Heilmann, using YUI, YQL and Yahoo Geo Technologies - **GeoMaker for Developers**



GeoMaker

And we're done. Below you'll see the map with your locations, the code to copy and paste to embed your own map and your locations as microformats.

Your Map code

Following is the code to generate the map above. For you to use it in your own products you need to apply for a free map developer key and replace the **YMAPPID** in the code with your own key.

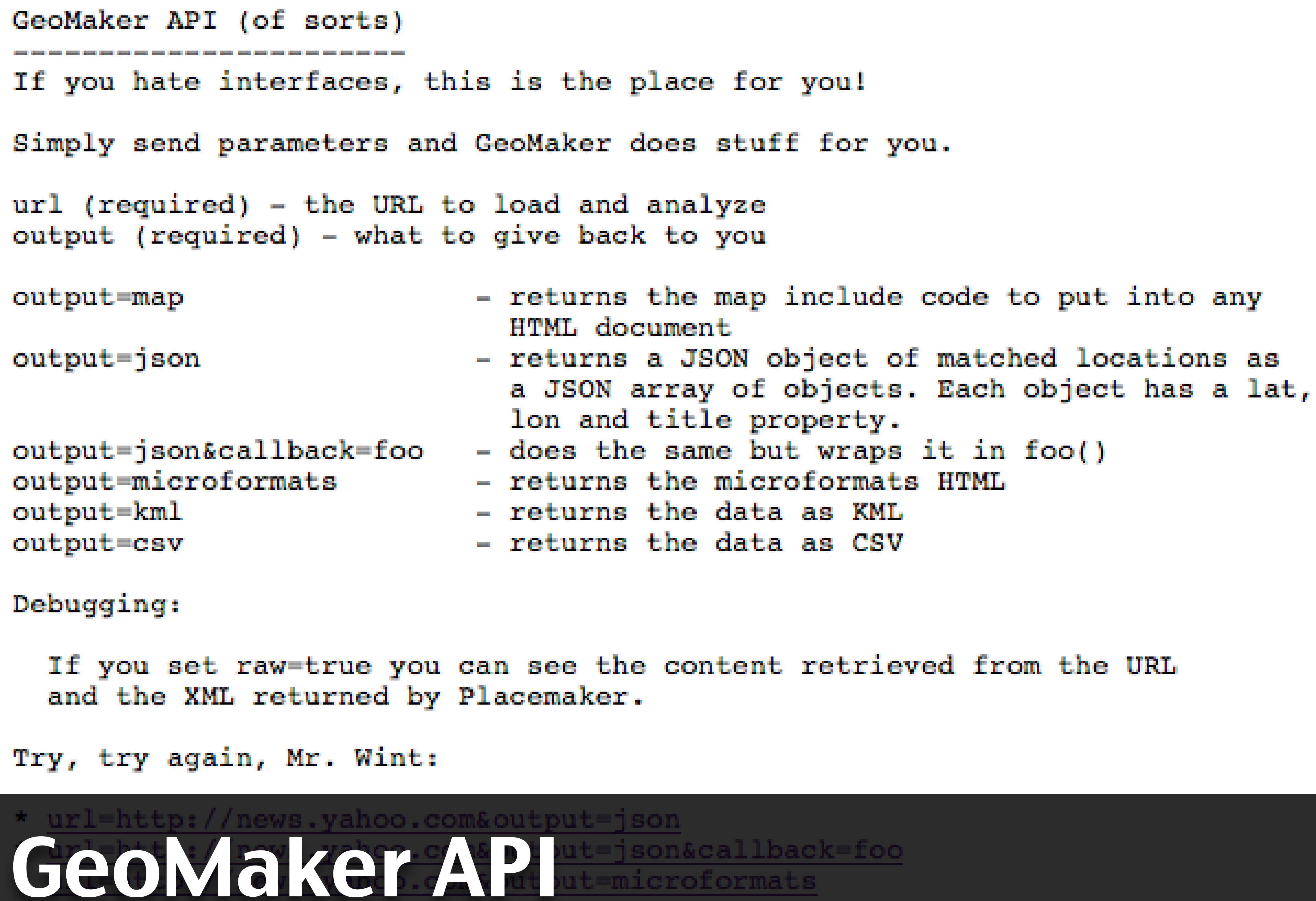
```
<div id="map"></div>
<script src="http://yui.yahooapis.com/2.7.0/build/utilities
/utilities.js"></script>
<script type="text/javascript" src="http://l.yimg.com/d/lib
/map/js/api/ymapapi_3_8_2_3.js"></script>
<script type="text/javascript">
var YMAPPID =
'4HjHmAfV34EriHQuqW_7P6A5_eFww96Yic4a7dlDtkoT_3LH
NNkrAHskOnoZK9A-';
            // ^^ REPLACE THIS CODE WITH YOUR OWN!!! ^^
(function(){
  function placeonmap(o){
    if(o.length > 0){
```

Your Microformatted locations

If all you wanted is geolocate your text, here are the geo-microformats to copy and paste into the correct sections. Notice that we are not using the **ABBR** pattern as accessibility is something we care about.

```
<-- match:Amsterdam, Netherlands -->
<span class="vcard">
<span class="adr">
<span class="locality">Amsterdam, North Holland,
NL</span>
</span>
(<span class="geo">
<span class="latitude">52.3731</span>,
<span class="longitude">4.89319</span>
</span>)
</span>
```

GeoMaker by Christian Heilmann, using YUI, YQL and Yahoo Geo Technologies - **GeoMaker for Developers**

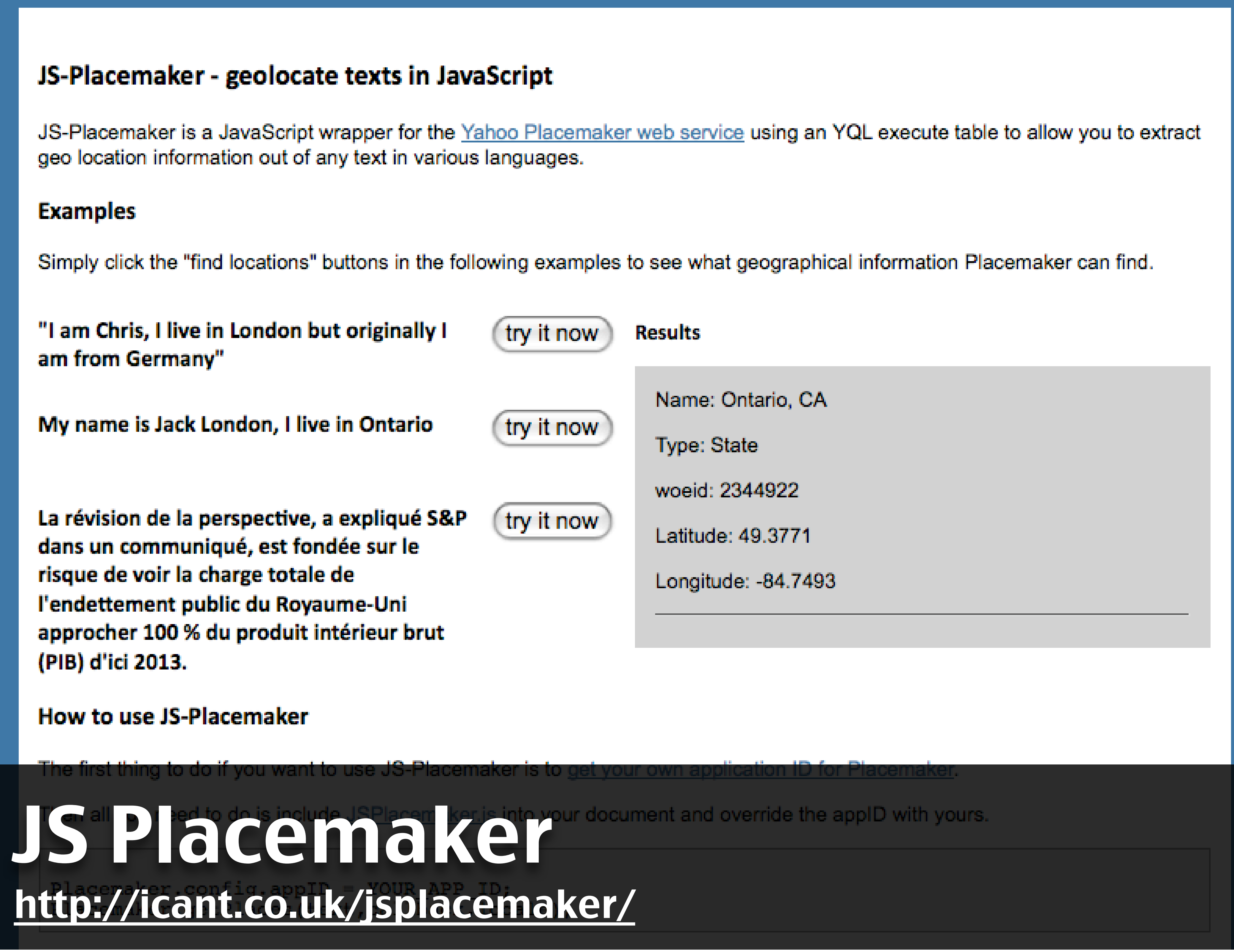


```
GeoMaker API (of sorts)
-----------------------
If you hate interfaces, this is the place for you!

Simply send parameters and GeoMaker does stuff for you.

url (required) - the URL to load and analyze
output (required) - what to give back to you

output=map                  - returns the map include code to put into any
                              HTML document
output=json                 - returns a JSON object of matched locations as
                              a JSON array of objects. Each object has a lat,
                              lon and title property.
output=json&callback=foo    - does the same but wraps it in foo()
output=microformats         - returns the microformats HTML
output=kml                  - returns the data as KML
output=csv                  - returns the data as CSV

Debugging:

  If you set raw=true you can see the content retrieved from the URL
  and the XML returned by Placemaker.

Try, try again, Mr. Wint:

* url=http://news.yahoo.com&output=json
* url=http://news.yahoo.com&output=json&callback=foo
* url=http://news.yahoo.com&output=microformats
```

GeoMaker API

http://icant.co.uk/geomaker/api.php

GeoMaker also has an own API that makes it easy to convert URLs to all kind of handy formats.



JS-Placemaker - geolocate texts in JavaScript

JS-Placemaker is a JavaScript wrapper for the Yahoo Placemaker web service using an YQL execute table to allow you to extract geo location information out of any text in various languages.

Examples

Simply click the "find locations" buttons in the following examples to see what geographical information Placemaker can find.

"I am Chris, I live in London but originally I am from Germany"

My name is Jack London, I live in Ontario

La révision de la perspective, a expliqué S&P dans un communiqué, est fondée sur le risque de voir la charge totale de l'endettement public du Royaume-Uni approcher 100 % du produit intérieur brut (PIB) d'ici 2013.

Results

Name: Ontario, CA

Type: State

woeid: 2344922

Latitude: 49.3771

Longitude: -84.7493

How to use JS-Placemaker

The first thing to do if you want to use JS-Placemaker is to get your own application ID for Placemaker.

Then all you need to do is include JSPlacemaker.js into your document and override the appID with yours.

```
Placemaker.config.appID = YOUR_APP_ID;
```

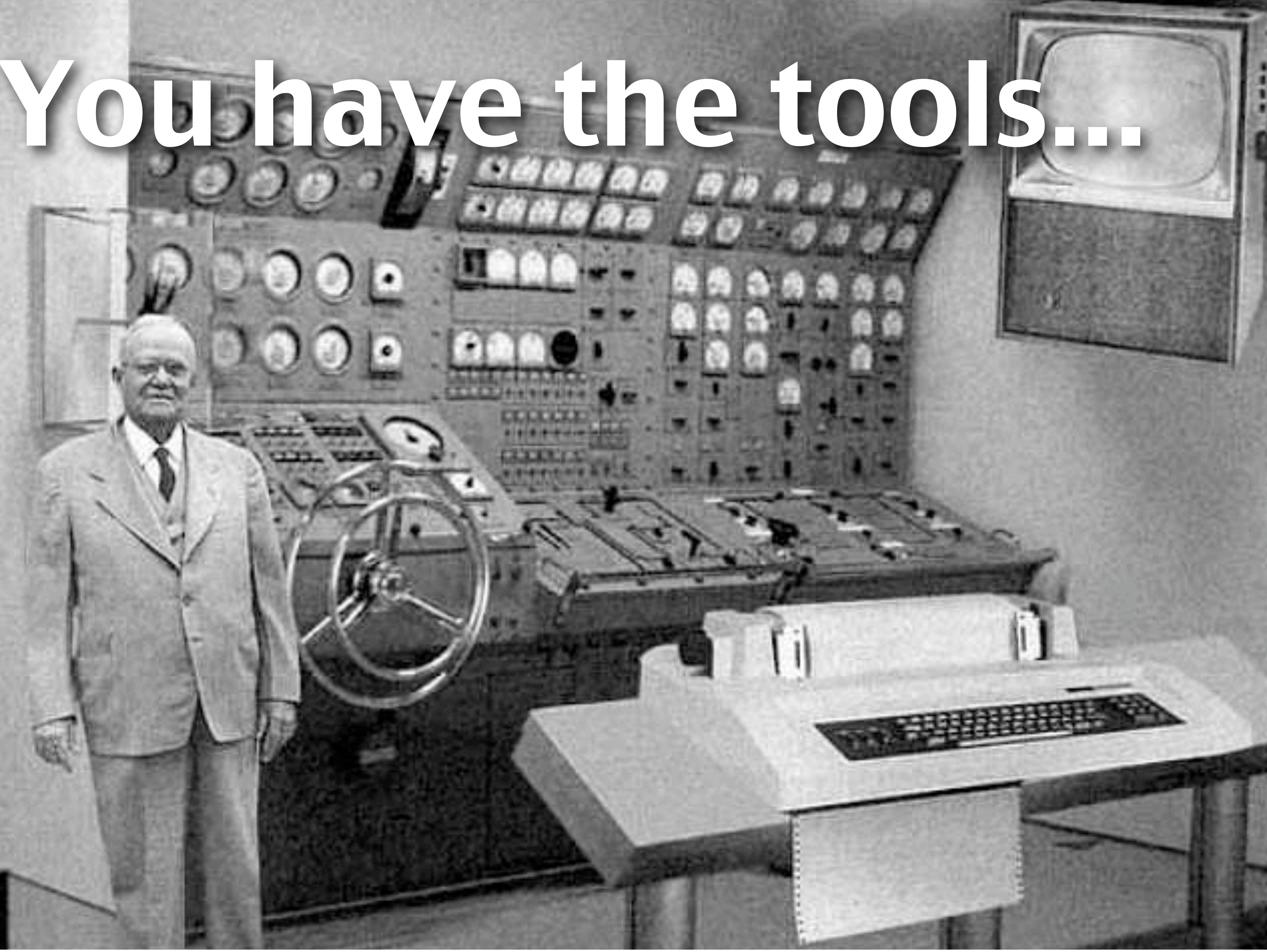
JS Placemaker

http://icant.co.uk/jsplacemaker/

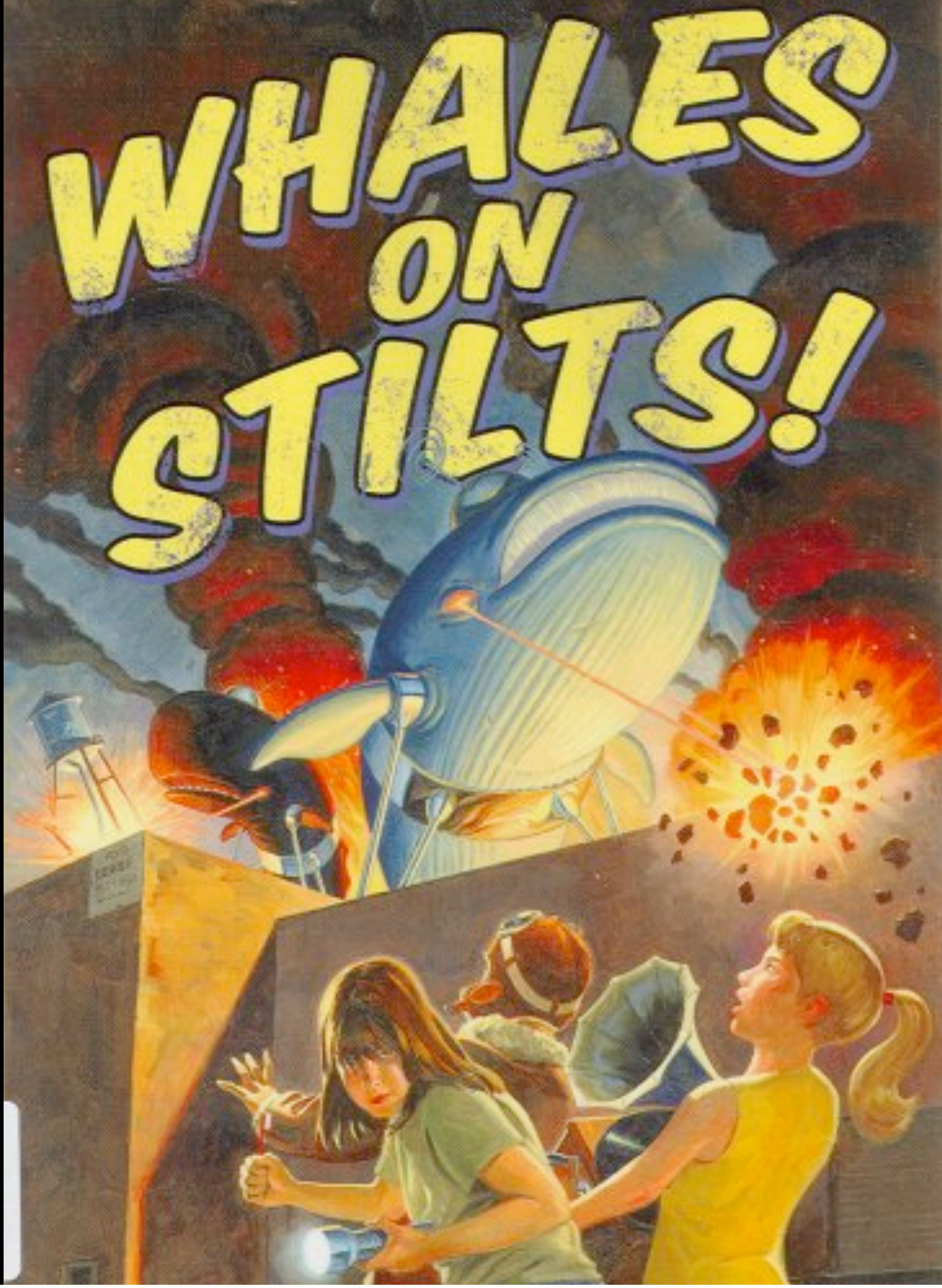
JS Placemaker is a JavaScript wrapper for Placemaker using the open YQL table.

You have the data…

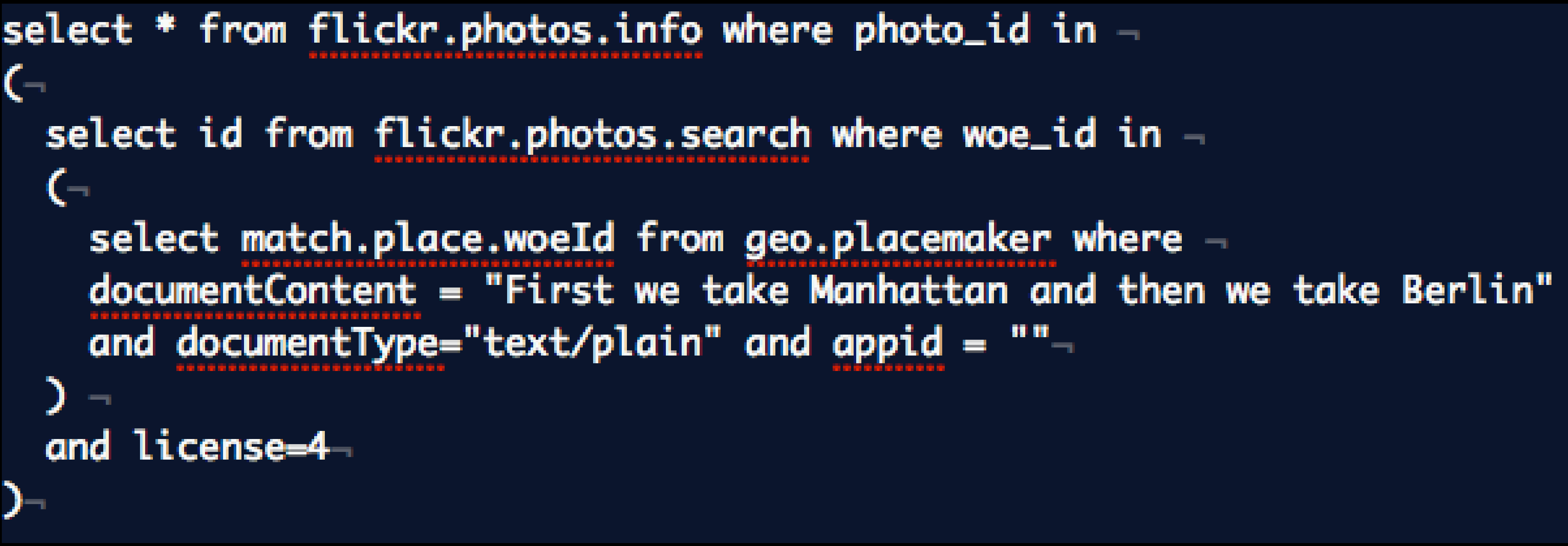
You have the tools...



All you need is a great idea.

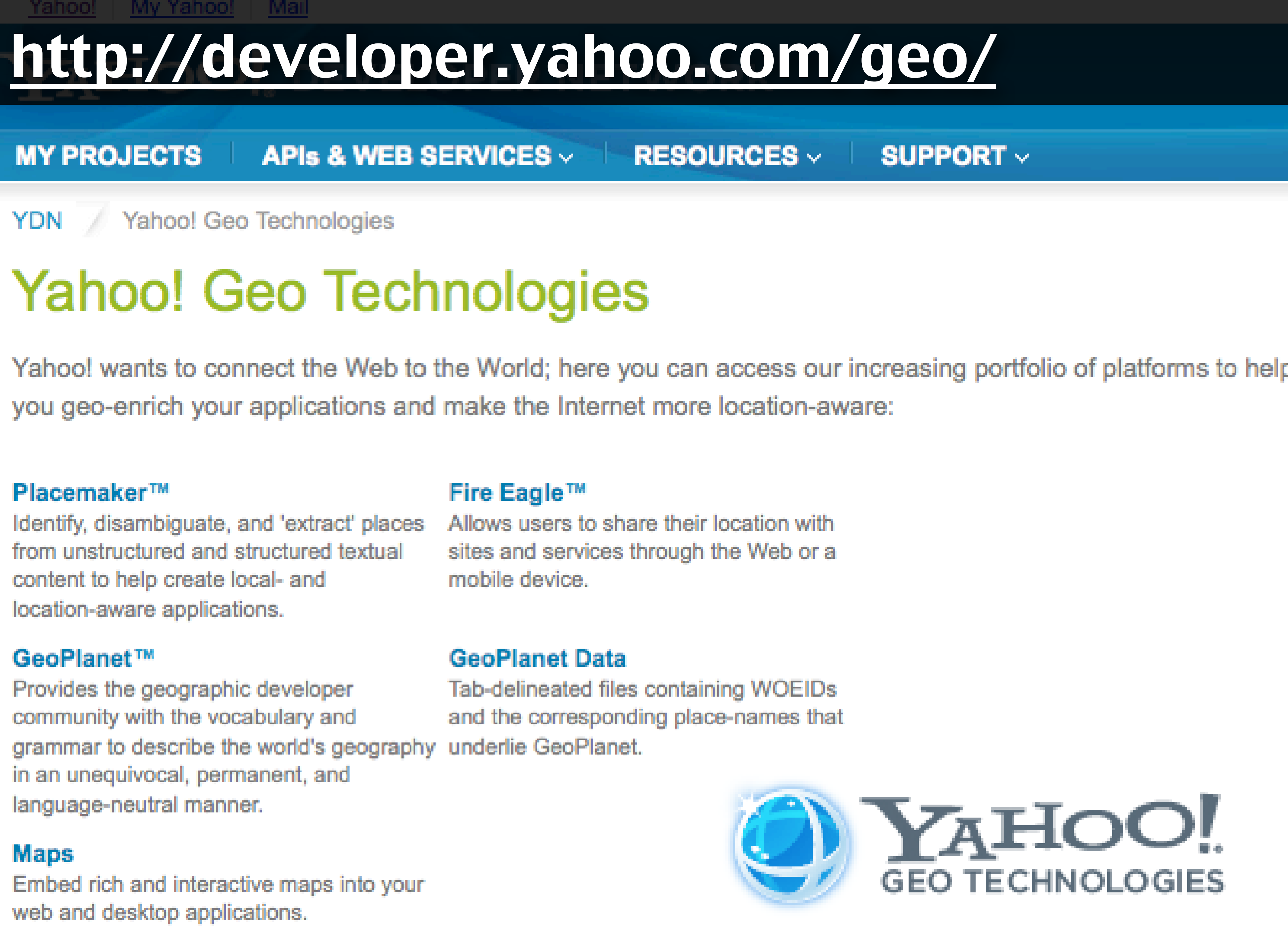


Flickr knows woeid :)



```
select * from flickr.photos.info where photo_id in
(
  select id from flickr.photos.search where woe_id in
  (
    select match.place.woeId from geo.placemaker where
    documentContent = "First we take Manhattan and then we take Berlin"
    and documentType="text/plain" and appid = ""
  )
  and license=4
)
```

http://developer.yahoo.com/geo/



YDN / Yahoo! Geo Technologies

Yahoo! Geo Technologies

Yahoo! wants to connect the Web to the World; here you can access our increasing portfolio of platforms to help you geo-enrich your applications and make the Internet more location-aware:

Placemaker™
Identify, disambiguate, and 'extract' places from unstructured and structured textual content to help create local- and location-aware applications.

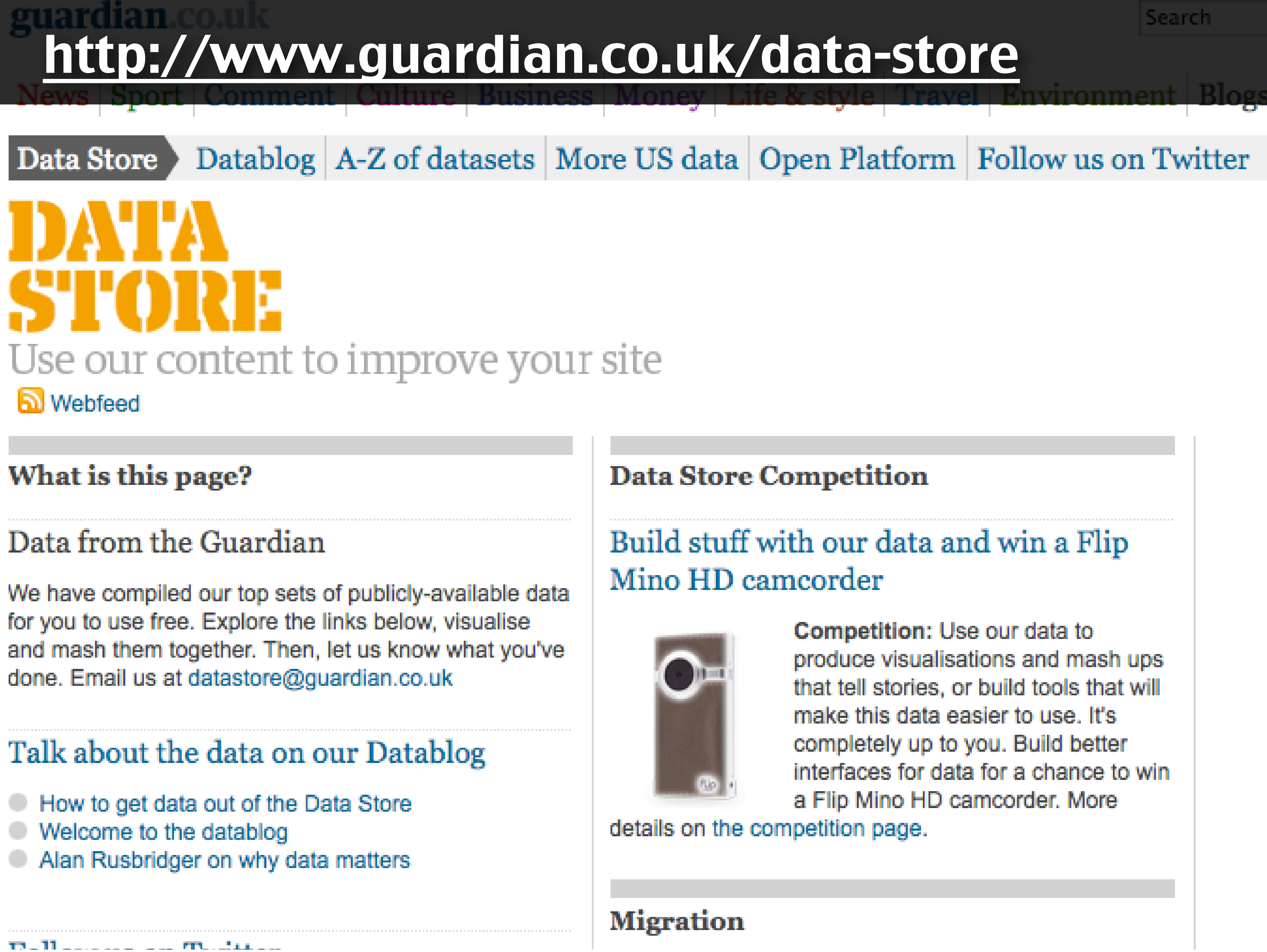
GeoPlanet™
Provides the geographic developer community with the vocabulary and grammar to describe the world's geography in an unequivocal, permanent, and language-neutral manner.

Maps
Embed rich and interactive maps into your web and desktop applications.

Fire Eagle™
Allows users to share their location with sites and services through the Web or a mobile device.

GeoPlanet Data
Tab-delineated files containing WOEIDs and the corresponding place-names that underlie GeoPlanet.

http://www.guardian.co.uk/data-store



Use our content to improve your site

What is this page?

Data from the Guardian

We have compiled our top sets of publicly-available data for you to use free. Explore the links below, visualise and mash them together. Then, let us know what you've done. Email us at datastore@guardian.co.uk

Talk about the data on our Datablog

- How to get data out of the Data Store
- Welcome to the datablog
- Alan Rusbridger on why data matters

Data Store Competition

Build stuff with our data and win a Flip Mino HD camcorder

Competition: Use our data to produce visualisations and mash ups that tell stories, or build tools that will make this data easier to use. It's completely up to you. Build better interfaces for data for a chance to win a Flip Mino HD camcorder. More details on the competition page.

Migration

THANKS!

Keep in touch:

Christian Heilmann

http://wait-till-i.com

http://scriptingenabled.org

http://twitter.com/codepo8