Geolocation and Maps with PHP and MongoDB

4developers - Poznań, Poland - April 18th, 2012

Derick Rethans - derick@derickrethans.nl - twitter: @derickr

http://joind.in/6314

About Me

Derick Rethans

- Dutchman living in London
- PHP mongoDB driver maintainer for 10gen (the company behind mongoDB)
- Author of Xdebug
- Author of the mcrypt, input_filter, dbus, translit and date/time extensions

10gen

mongoDB

The Earth is

not a sphere...

... but a bit of a pear.

The Earth's shape

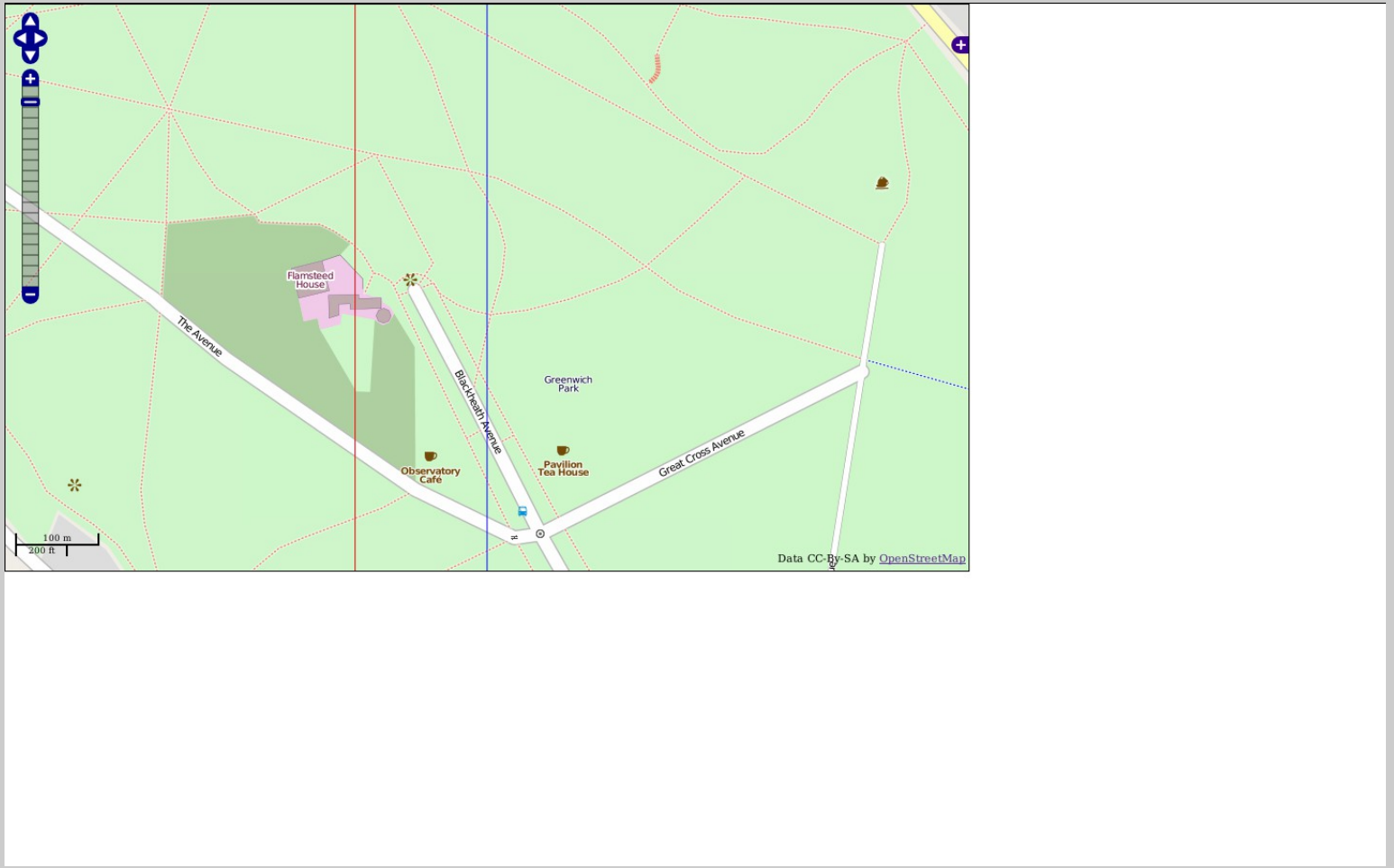
In cartography, the Earth's shape has to be approximated: a reference ellipsoid

- specify the Earth's radius and a flattening constant
- different ones are in use
- also called datum or geodetic system
- for coordinates, a meridian (0° longitude) should also be specified

Important ones are:

- WGS84: That's what GPS uses
- OSGB36: That's what Ordnance Survey uses
- ED50: That's what we use in most of Europe

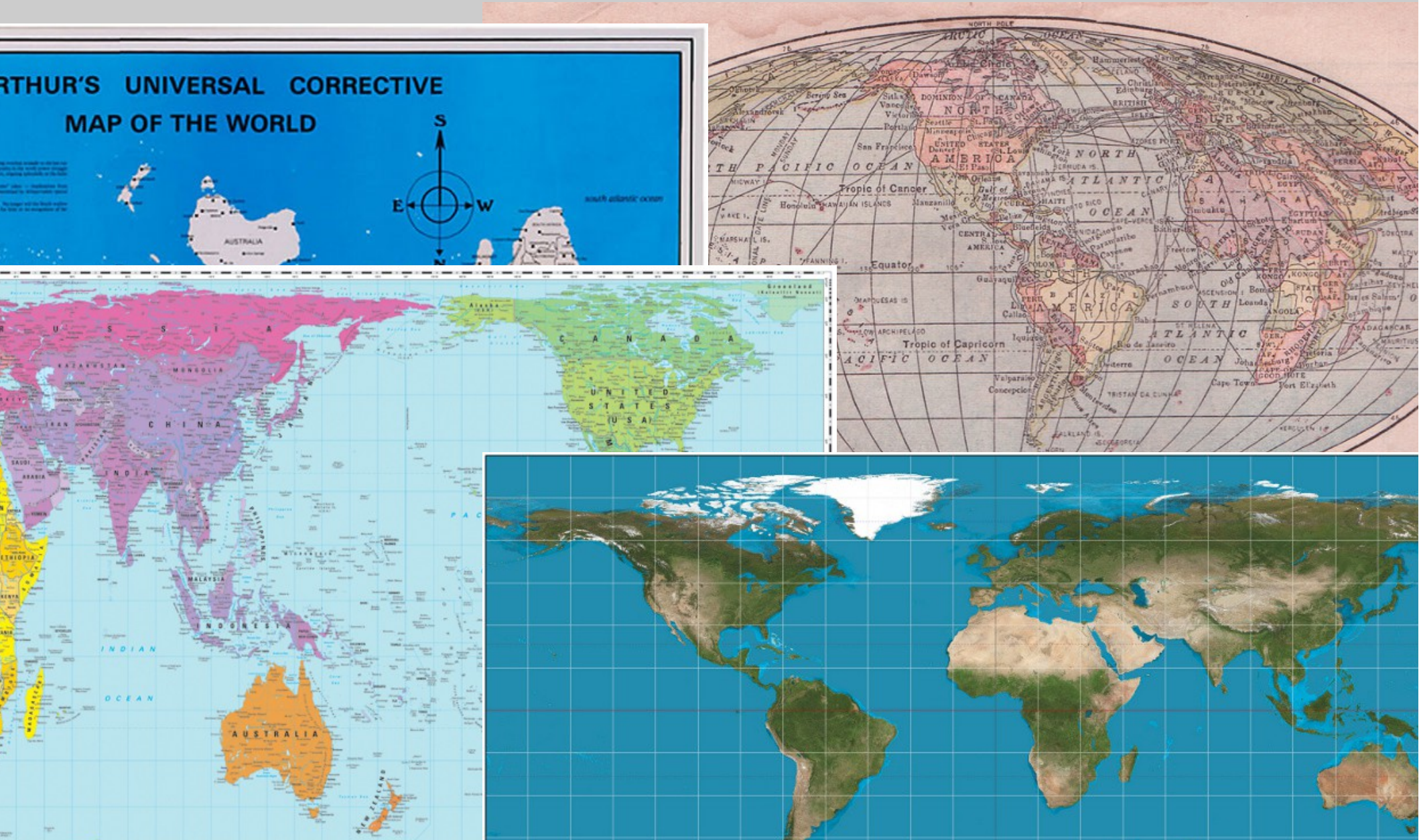
Greenwich Meridian



Greenwich Meridian
IRTS Meridian

Map Projections

Different projections have different strengths



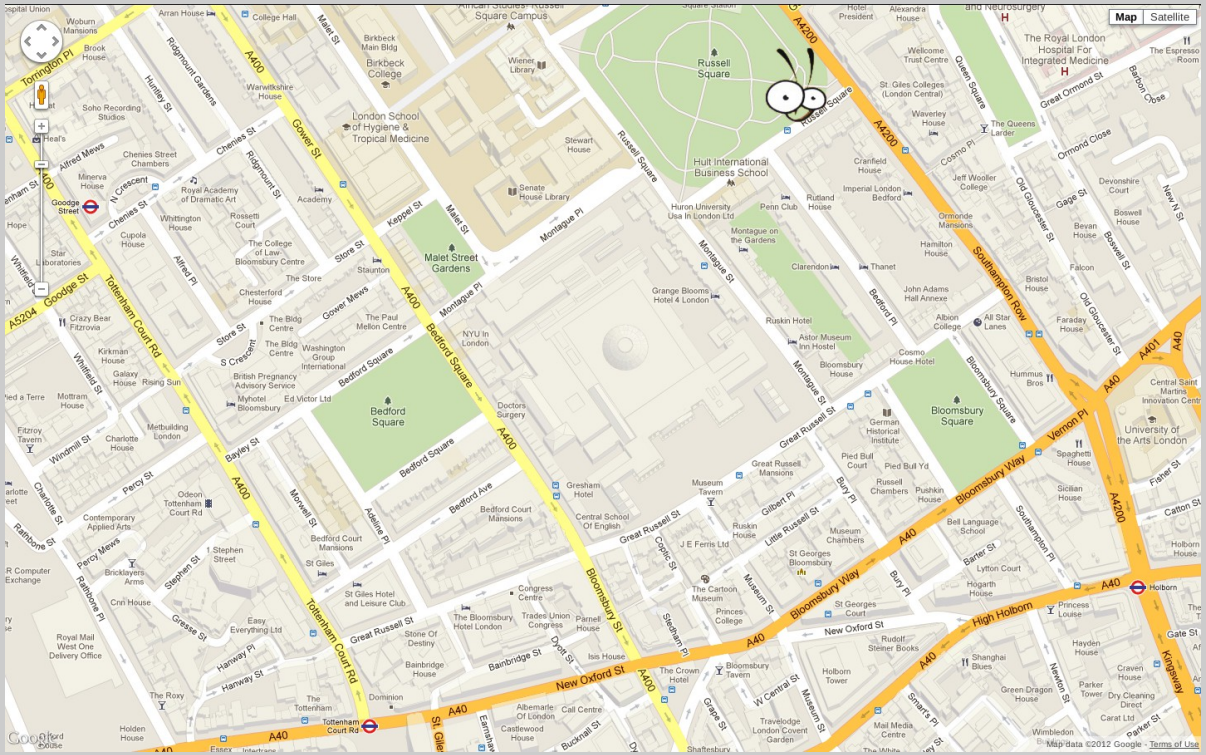
OpenStreetMap

- "Wikipedia for Map Data"
- Licensed under the Creative Commons Attribution-ShareAlike 2.0 licence (CC-BY-SA): Licensed under the Open Database License (ODbL 1.0): You are free to copy, distribute, transmit and adapt our maps and data, as long as you credit OpenStreetMap and its contributors. If you alter or build upon our maps or data, you may distribute the result only under the same licence.
- Rendered map:
- A lot of data is not rendered, but is available.

What is OpenStreetMap?

- A database with geographic data of the whole planet under a free license
- APIs to add, retrieve and query map data
- A social platform for improving the map

What is Google Maps?



In words:

- It gives you "free" map tiles
- It gives you satellite imagery
- It gives you StreetView™
- It gives you a free JavaScript library
- It gives you a free geo-coding API

Terms and Conditions

Google Maps:

- Can I generate a map image using the Google Static Maps API which I store and serve from my website? — You may not store and serve copies of images generated using the Google Static Maps API from your website.
- What transaction limits apply to the Maps API? — From October 1st 2011 commercial web sites and applications using the Maps API may at no cost generate: up to 2,500 map transactions
- Geocoding Usage Limits: Use of the Google Geocoding API is subject to a query limit of 2,500 geolocation requests per day

OSM:

- OpenStreetMap is open data, licensed under the Creative Commons Attribution-ShareAlike 2.0 licence (CC-BY-SA).

Showing a Map

OpenLayers

```
<?xml version="1.0" encoding="iso-8859-1"?>
 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
 <html xmlns="http://www.w3.org/1999/xhtml" lang="EN">
     <head>
     <style>
         html,body { margin: 0; padding: 0; width: 1004px; height: 590px; }
         #map { width: 100%; height: 100%; border: 1px solid black; float: left; z-index: -1; }
         div.olControlAttribution { bottom: 0.5em; font-size: 70%; }
     </style>
     <script src='OpenLayers.js'></script>
     <script src='osm/OpenStreetMap.js'></script>
     <script type="text/javascript">
     var map; //complex object of type OpenLayers.Map

     var lat=51.51922
     var lon=-0.12736
     var zoom=17

     function init() {
         map = new OpenLayers.Map ("map", {
             controls:[
                 new OpenLayers.Control.PanZoomBar(),
                 new OpenLayers.Control.Attribution()],
             projection: new OpenLayers.Projection("EPSG:900913"),
             displayProjection: new OpenLayers.Projection("EPSG:4326")
         } );

         layerMapnik = new OpenLayers.Layer.OSM.Mapnik("Mapnik");
         map.addLayer(layerMapnik);

         var lonLat = new OpenLayers.LonLat(lon, lat).
             transform(map.displayProjection, map.projection);
         map.setCenter(lonLat, zoom);
     }
     </script>
 </head>
 <body onload="init();">
     <div id='map'></div>
 </body>
 </html>
```

Showing a Map

Leaflet

```
<!DOCTYPE html>
  <html>
  <head>
      <title>Leaflet Quick Start Guide Example</title>
      <meta charset="utf-8" />

      <link rel="stylesheet" href="leaflet/leaflet.css" />
      <!--[if lte IE 8]><link rel="stylesheet" href="leaflet/leaflet.ie.css" /><![endif]-->
      <script src="leaflet/leaflet.js"></script>
  </head>
  <body>
      <div id="map" style="width: 1004px; height: 590px"></div>
      <script type="text/javascript">
          var map = new L.Map('map');

          var osmUrl = 'http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
              osmAttrib = 'Map data &copy; 2011 OpenStreetMap contributors',
              osm = new L.TileLayer(osmUrl, {maxZoom: 18, attribution: osmAttrib});

          map.setView(new L.LatLng(51.5179, -0.12), 13).addLayer(osm);
      </script>
  </body>
  </html>
```

Looking up latitude and longitude from a location


```
<?php
$baseUrl = 'http://nominatim.openstreetmap.org/search?format=json&limit=1';
$name = urlencode( ':-:location:-:' );
$data = file_get_contents( "{$baseUrl}&q={$name}" );
$json = json_decode( $data );

var_dump( $json[0] );
?>
```

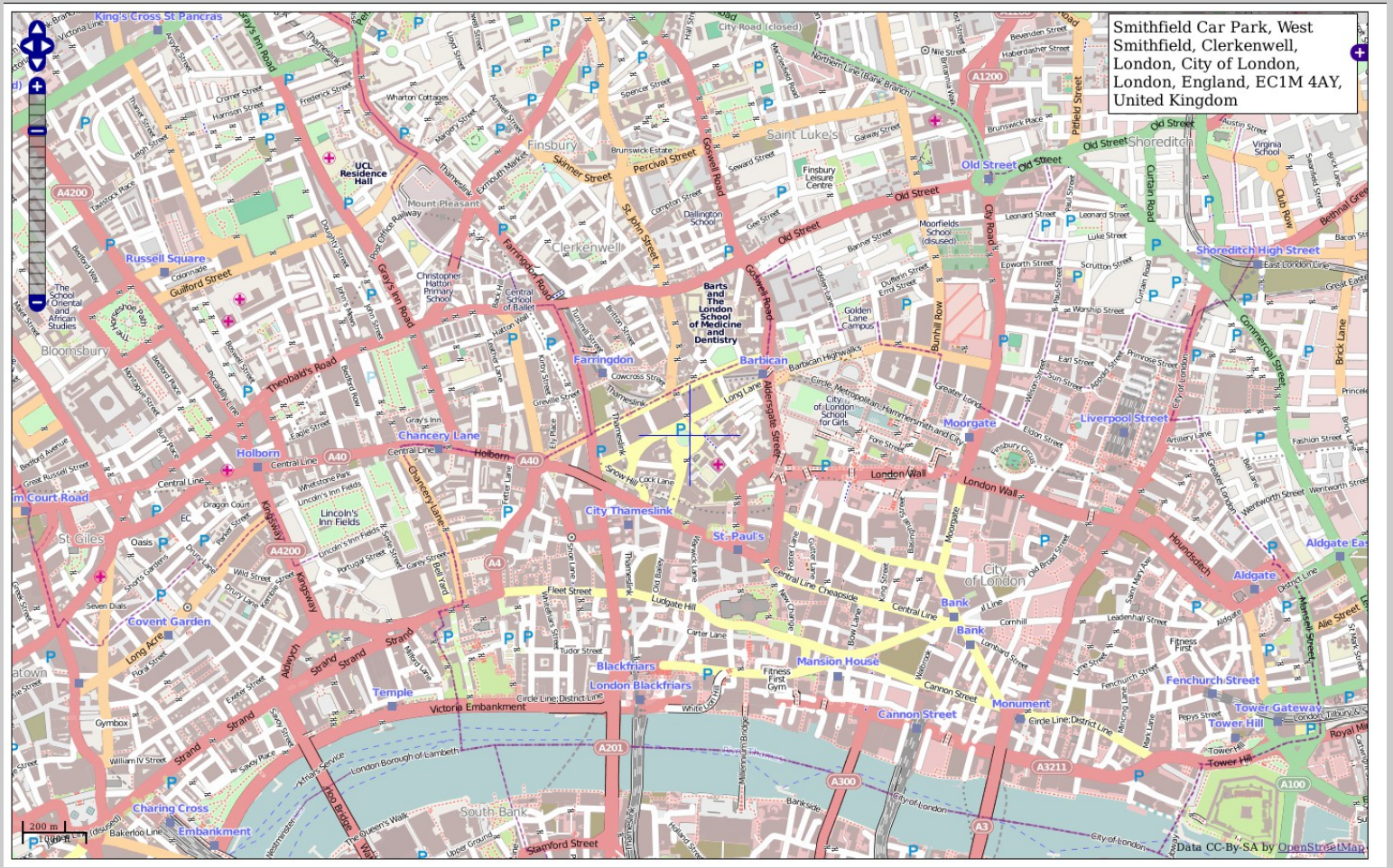
Looking up latitude/longitude

Different services

- Nominatim:http://nominatim.openstreetmap.org/search?format=json&limit=1&q=London
- ?

Reverse Geocoding

Finding a name for coordinates



Finding a name for the current location

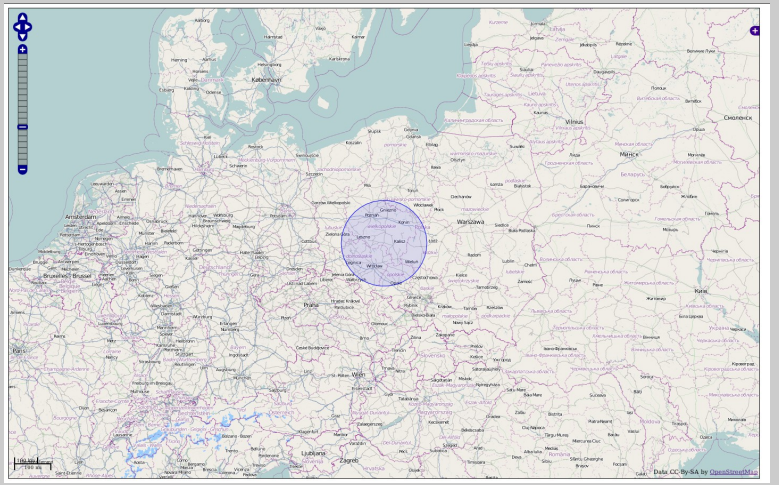
Different services

- Geonames:http://ws.geonames.org/findNearbyPlaceNameJSON?username=derick&style=full&lat={$lat}&lng={$lon}
- Nominatim:http://nominatim.openstreetmap.org/reverse?format=json&lat={$lat}&lon={$lon}&zoom={$z}

Finding the user

Using JavaScript to locate the user

```
function getPosition()
 {
   navigator.geolocation.getCurrentPosition(iKnowWhereYouAre, notTheFaintestClue,
 {timeout:30000});
 }

 function notTheFaintestClue()
 {
 }

 function iKnowWhereYouAre(position)
 {
   var lonLat = new OpenLayers.LonLat(
     position.coords.longitude, position.coords.latitude
   ).transform(map.displayProjection, map.projection);
   map.setCenter(lonLat, zoom);

   center = map.getCenter().
     transform(map.getProjectionObject(), new OpenLayers.Projection("EPSG:4326"));

   factor = Math.cos(center.lat / (180/Math.PI)), 10 + map.getZoom() * 2;

   multiFeature = new OpenLayers.Feature.Vector(
     OpenLayers.Geometry.Polygon.createRegularPolygon(
       new OpenLayers.Geometry.Point(
         center.lon, center.lat
       ).transform(new OpenLayers.Projection("EPSG:4326"), map.getProjectionObject()),
       position.coords.accuracy / factor, 10
     ),
     {
       color: 'blue',
       align: 'rt'
     }
   );

   vectorLayer.removeAllFeatures();
   vectorLayer.drawFeature(multiFeature);
   vectorLayer.addFeatures([multiFeature]);
 }
```



The Data

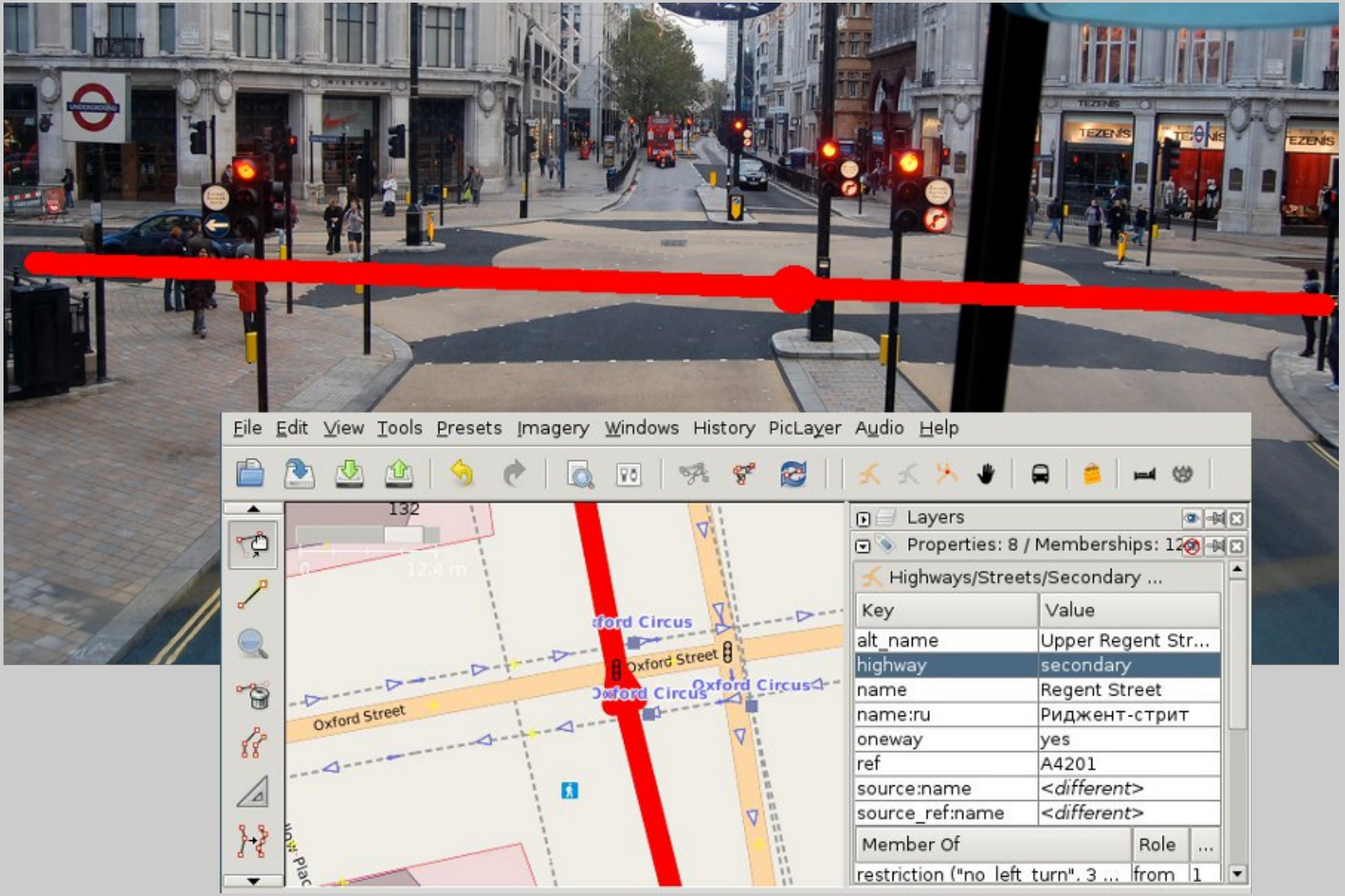
- Nodes (Lat/Lon point)

  <node id='459517295' lat='50.0100766' lon='8.3162402' user='WoGo' timestamp='2009-08-09T11:45:33Z' uid='152395' version='1' changeset='2083951'>
- Ways (Ordered interconnection of nodes)
- Areas (Closed ways)

  <way id='76174399' user='Derick Rethans' uid='37137' timestamp='2010-09-06T08:30:14Z' version='1' changeset='5695697'>
  <nd ref='898861293'/>
  <nd ref='898861305'/>
  <nd ref='898861298'/>
  <nd ref='898861315'/>
  <nd ref='898861293'/>
  ...
  </way>
- Tags (Describe an element)

  <tag k='addr:housenumber' v='375'/>
  <tag k='addr:street' v='Kilburn High Road'/>
  <tag k='amenity' v='pub'/>
  <tag k='building' v='yes'/>
  <tag k='name' v='North London Tavern'/>

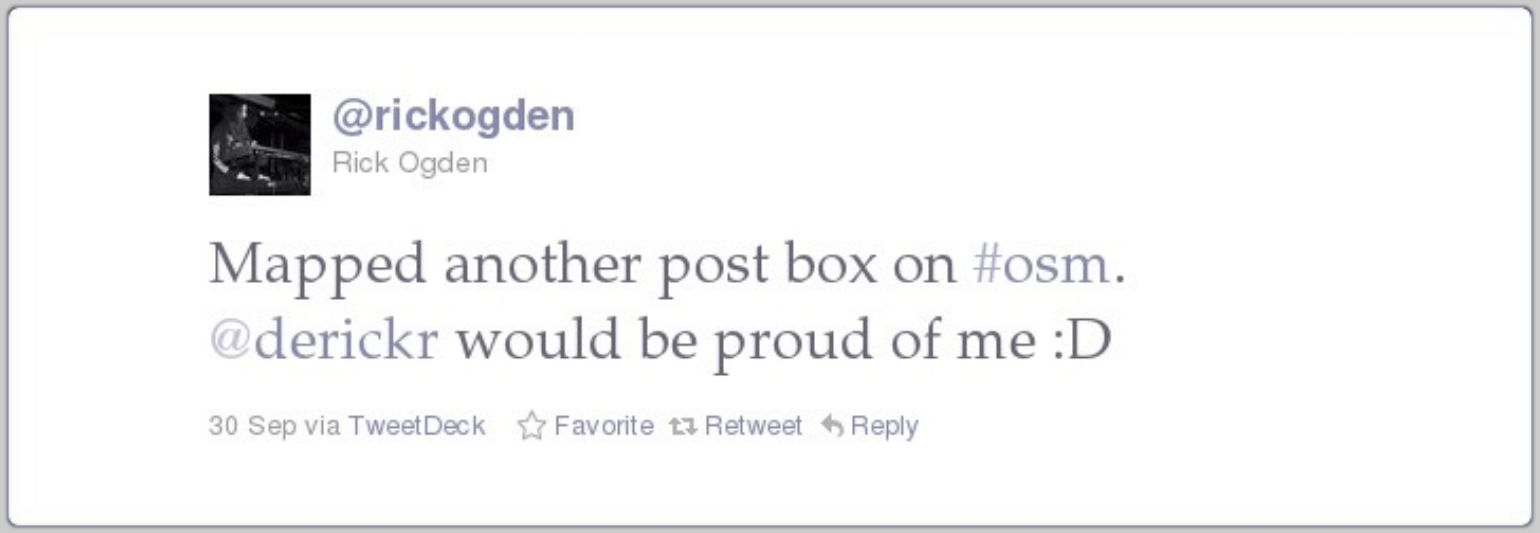
Editting tags



Mapping parties

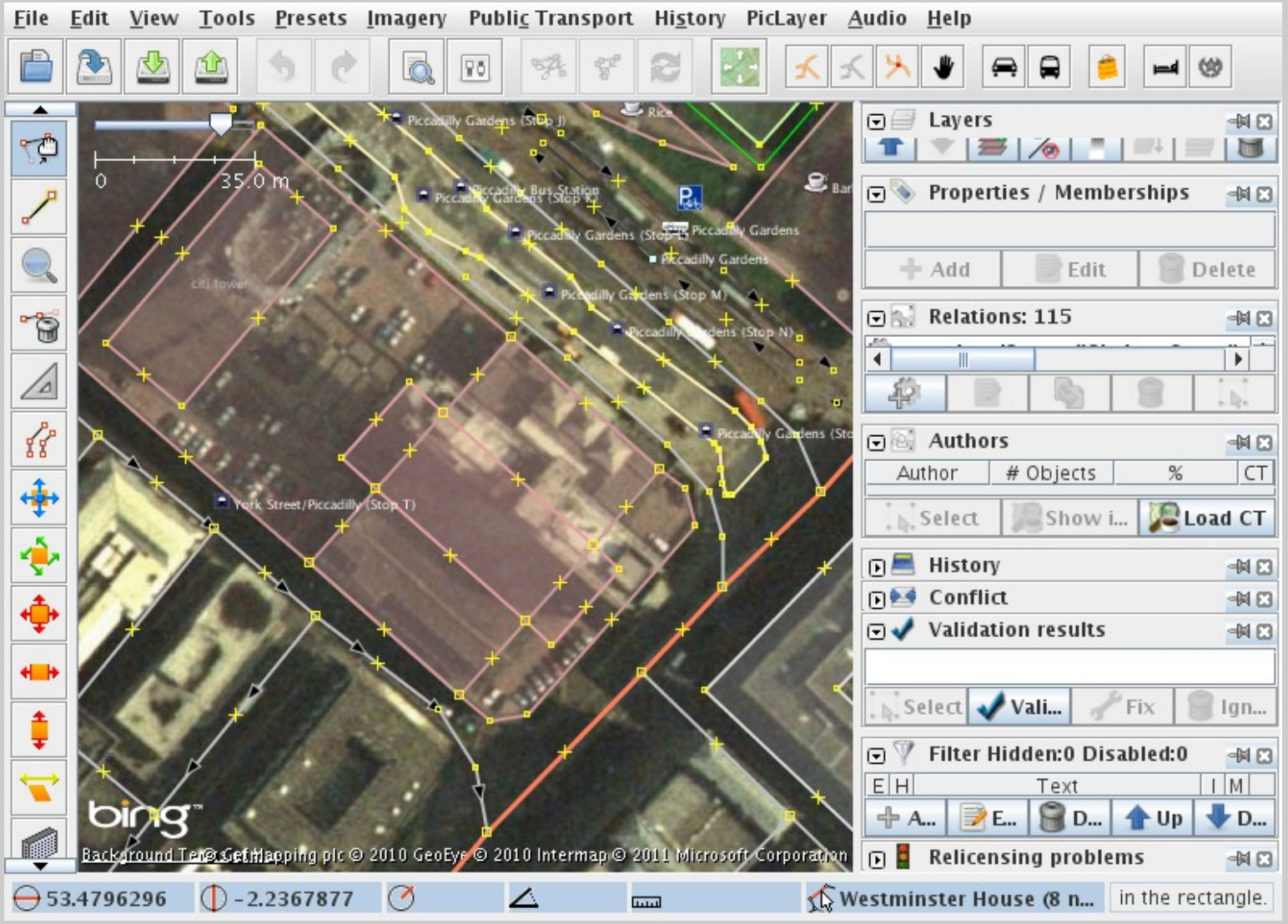
As OSM's main purpose is creating a database of free map data:

- data needs to be generated (surveyed), which is best done by hand
- data needs to be added to the database: Potlatch, JOSM
- surveying is more fun if you're not alone
- why not do it in a group?

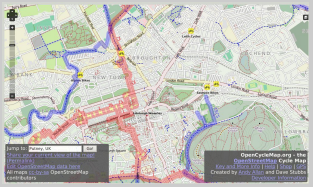


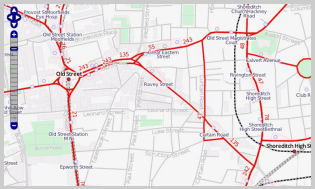
Mapping parties

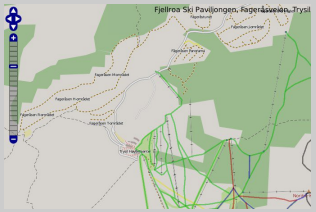
Editting the map

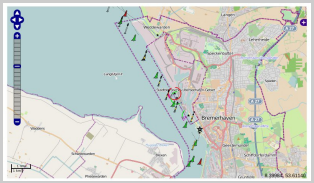


Different visualisations of the data









Using OpenStreetMap data

Process:

- Use XAPI, API or extracts to fetch data
- Parse XML file with PHP into a DB (MongoDB)
- Query database
- Show data
- Profit (or in our case, we found a restaurant)!

Fetching OSM data

```
wget http://download.geofabrik.de/osm/europe/great_britain/england.osm.bz2
<?xml version='1.0' encoding='UTF-8'?>
<osm version="0.6" generator="Osmosis 0.39">
  <bound box="51.26000,-0.56300,51.68000,0.28000" origin="http://www.openstreetmap.org/api/0.6"/>
  <node id="44" version="3" timestamp="2011-09-08T20:53:43Z" uid="15867" user="mattfromderby" changeset="9249514"
lat="51.5250157" lon="-0.1485302"/>
  <node id="47" version="2" timestamp="2011-09-08T20:53:43Z" uid="15867" user="mattfromderby" changeset="9249514"
lat="51.525182" lon="-0.1479761"/>
  <node id="52" version="3" timestamp="2011-09-08T21:13:19Z" uid="15867" user="mattfromderby" changeset="9249514"
lat="51.5289059" lon="-0.1458559"/>
...
  <node id="1571982070" version="1" timestamp="2011-12-31T20:37:18Z" uid="545369" user="bigfatfrog67" changeset="10257991"
lat="51.5087759" lon="-0.1253258">
    <tag k="name" v="Charing Cross"/>
    <tag k="railway" v="subway_entrance"/>
    <tag k="source:name" v="survey"/>
    <tag k="source:railway" v="survey"/>
  </node>
...
  <way id="74" version="5" timestamp="2010-12-12T22:43:15Z" uid="508" user="Welshie" changeset="6643030">
    <nd ref="196101"/>
    <nd ref="196100"/>
    <nd ref="196331"/>
    <tag k="abutters" v="retail"/>
    <tag k="highway" v="primary"/>
    <tag k="maxspeed" v="30 mph"/>
    <tag k="name" v="Ballards Lane"/>
    <tag k="ref" v="A598"/>
  </way>
...
  <relation id="69" version="13" timestamp="2009-02-09T14:57:24Z" uid="45027" user="PA94" changeset="307506">
    <member type="way" ref="3793834" role="outer"/>
    <member type="way" ref="26454599" role="inner"/>
    <tag k="created_by" v="xybot"/>
    <tag k="type" v="multipolygon"/>
  </relation>
...
</osm>
```

Importing data

MongoDB is schema less, so fits the OSM data model quite well, but you still need to think about how to store the data best:

```
$doc = array(
    'name' => 'A440',
    'tags = [
        'highway' => 'secondary',
        'oneway' => 'yes'
    ]
);

$db->poi->ensureIndex( [ 'name' => 1 ] );
$db->poi->ensureIndex( [ 'tags.highway' => 1 ] );
$db->poi->ensureIndex( [ 'tags.oneway' => 1 ] );
```

or:

```
$doc = array(
    'name' => 'A440',
    'tags' = [
        { 'k' => 'highway', 'v' => 'secondary' },
        { 'k' => 'oneway', 'v' => 'yes' }
    ]
);

$db->poi->ensureIndex( [ 'name' => 1 ] );
$db->poi->ensureIndex( [ 'tags.k' => 1, 'tags.v' => 1 ] );
```

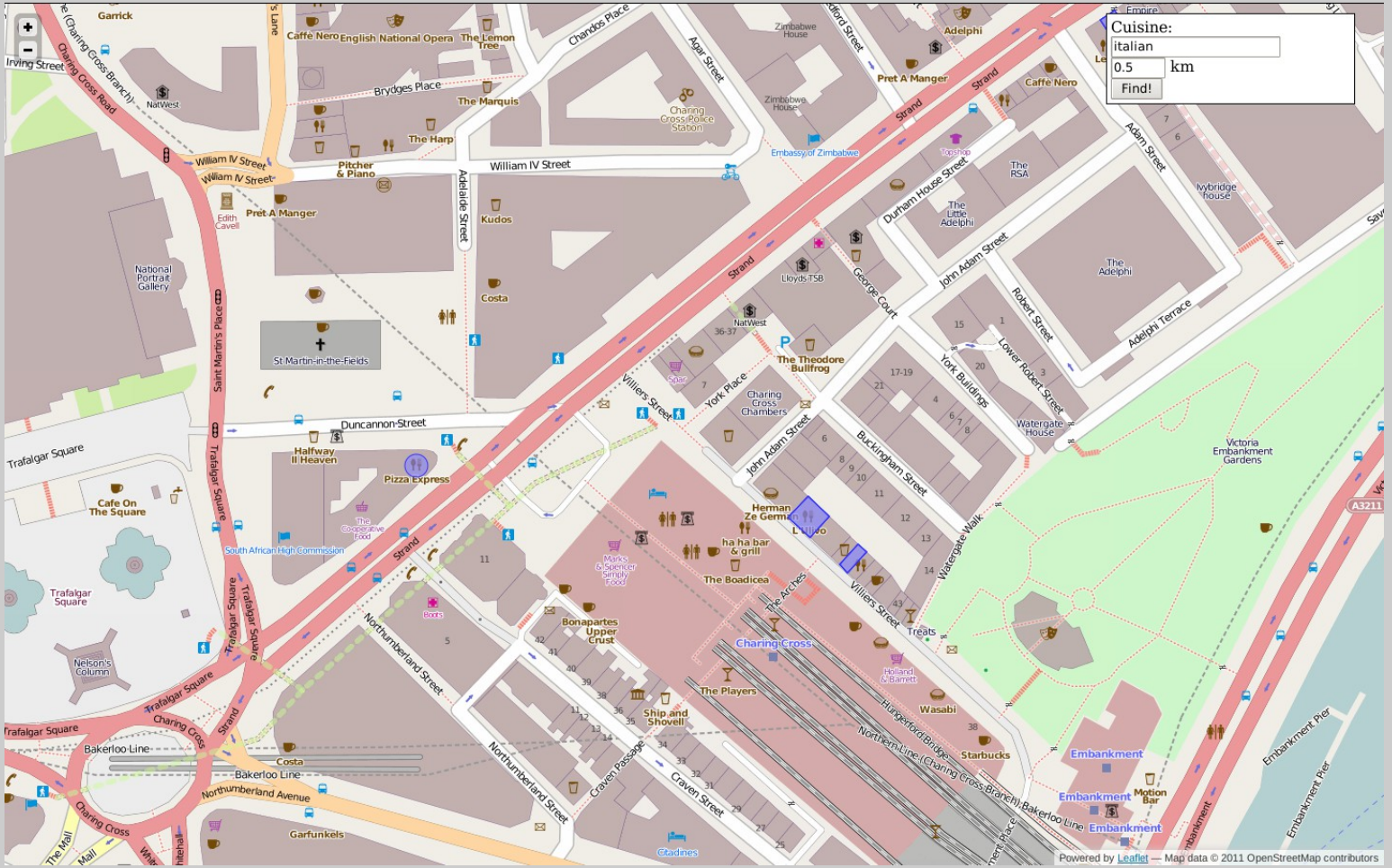
Importing data


```
{
    "_id" : ObjectId("4f8d9b6644670adc440510eb"),
    "id" : "w1572026939",
    "type" : NumberLong(2),
    "name" : "Noura",
    "address" : "Curzon Street 16 W1J 5HP",
    "tags" : {
        "addr:housenumber" : "16",
        "addr:postcode" : "W1J 5HP",
        "addr:street" : "Curzon Street",
        "amenity" : "restaurant",
        "building" : "yes",
        "contact:phone" : "+44 20 74951050",
        "cuisine" : "lebanese"
    },
    "tags_indexed" : [
        { "k" : "addr:housenumber", "v" : "16" },
        { "k" : "addr:postcode", "v" : "W1J 5HP" },
        { "k" : "addr:street", "v" : "Curzon Street" },
        { "k" : "amenity", "v" : "restaurant" },
        { "k" : "building", "v" : "yes" },
        { "k" : "contact:phone", "v" : "+44 20 74951050" },
        { "k" : "cuisine", "v" : "lebanese" }
    ],
    "loc" : [
        [ -0.1479191, 51.5067744 ],
        [ -0.1480757, 51.5067491 ],
        [ -0.1481252, 51.5068679 ],
        [ -0.1479687, 51.5068932 ],
        [ -0.1479191, 51.5067744 ]
    ]
}
```

Querying our database

```
<?php
$m = new Mongo( 'mongodb://localhost:27017' );
$c = $m->demo->poi;
$c->ensureIndex( array( 'loc' => '2d' ) );

$wantedD = isset($_GET['d']) ? $_GET['d']: 1;
$query = array(
    'tags_indexed.k' => 'cuisine',
    'tags_indexed.v' => $_GET['cuisine']
);

if ( $_GET['cuisine'] == 'all' )
{
    $query = array(
        'tags_indexed.k' => 'amenity',
        'tags_indexed.v' => 'restaurant' );
}

$s = $d->command(
    array(
        'geoNear' => 'poi',
        'spherical' => true,
        'near' => array(
            (float) $_GET['lon'],
            (float) $_GET['lat']
        ),
        'num' => 1000,
        'maxDistance' => $wantedD / 6371.01,
        'query' => $query,
    )
);
```

Finding Food



Resources

- Slides: http://derickrethans.nl/talks/:-:talk_id:-:
- Contact me: Derick Rethans: @derickr, derick@derickrethans.nl
- Feedback: http://joind.in/6314

10gen

mongoDB