Location-aware Content Tag: <location>,&location markup

Ole Aamot
ole@aamotsoftware.com
http://piperpal.com/o

<location> and &location tag for location-based markup

I am introducing the new <location> tag and &<location> syntax for document markup of location-based content in HTML by creating the Data Type Definition published on http://github.com/location

The new tag will make it easier for content providers to state that the HTML content they are publishing is geographically positioned and meant for indexing and displaying on services like Google Maps / Google Now.

The motivation behind this work is new location-based content retrieval, and would open a range of new ways to advertise geographically within a given radius and time interval.

Example: &concert

HTML Document with <location> tag with &concert example

Daryl Hall & John Oates playing a concert at the Greek Theater at UC Berkeley. The location tag is &concert.

```
<!DOCTYPE location SYSTEM
“http://raw.githubusercontent.com/location/location/master/location-1.0.dtd”>
<location name=”concert” data=”Daryl Hall & John Oates,
Greek Theater, 29.08.2015 from $69.50”
link=”http://www.hallandoates.com/tour.html?dc_id=793”
geo=”37.873596,-122.25443” radius=”150km”
notBefore=”2015-08-25T20:00”
notAfter=”2015-08-29T20:00” price=”$69.50”/>
```

JSON Data:

```
location [{
     name: “concert”,
     data: “Daryl Hall & John Oates, Greek Theater
29.08.2015; Prices from $69.50”,
     link:
“http://www.hallandoates.com/tour.html?dc_id=793”,
     geo: “37.873596,-122.25443”,
     radius: “150km”,
     notBefore: “2015-08-25T20:00”,
     notAfter: “2015-08-29T20:00”,
     price: “$69.50”,
}]
```

Piperpal: Location-aware markup

My idea is that the content on a website is marked up according to a location tag and that this tag decides which content that is meant to be indexed and eligble for placing a ad on. The content producers would tag their content with a location tag, the advertisements mark up their catalog with location-aware radius ads, and the users can embed the ads in their posts with a new location tag syntax such as &concert that could be implemented by Google services such as Gmail and Google+.

Pseudo code for logic of search for nearby matches

```
if (notBefore < NOW() < notAfter || ((&UserLoc - radius) < (geo) < (&UserLoc + radius)) → display
```

Piperpal is the first site that I am implementing for making a resource for location-based tags.

On http://piperpal.com/ I added the form that lets the users insert the Name, Location, and Service parameter and pay by credit card (via Stripe) to add the entry.

The mapping between the content produced by a content provider, the ad by the advertiser and the tagging by individual user is done on www.piperpal.com/<location> where the location tag example for <location> would be &concert and the URI would be www.piperpal.com/concert

Actual SQL Database representation

```
CREATE TABLE piperpal (
       id MEDIUMINT(8) UNSIGNED NOT NULL AUTO_INCREMENT,
       name VARCHAR(100) NOT NULL DEFAULT '',
       service VARCHAR(1024) NOT NULL,
       location VARCHAR(1024) NOT NULL,
       modified TIMESTAMP NOT NULL DEFAULT
       CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
       created TIMESTAMP NULL DEFAULT NULL,
       glat DECIMAL(10, 8) NOT NULL,
       glon DECIMAL(11, 8) NOT NULL,
       paid MEDIUMINT(16) UNSIGNED NOT NULL,
       token VARCHAR(32) NOT NULL,
       type VARCHAR(32) NOT NULL,
       email VARCHAR(256) NOT NULL,
       PRIMARY KEY (id));
);
```

The Apache API mapping of “http://piperpal.com/concert”

```
RewriteEngine On
RewriteRule ^([\sa-åA-Å0-9]+)$ /api/ [L]
```

The API will query the user’s link location and geoposition through the AJAX script and present a user with content in the nearest geographical approximity according to the Haversine formula.

```
<div id="log"></div>
<script>
   $(document).ready(function(){
        setInterval(function(){
          if (navigator.geolocation) {
              navigator.geolocation.getCurrentPosition(ajaxCall);
          } else{
              $('#log').html("GPS is not available");
          }
          function ajaxCall(position){
            var latitude = position.coords.latitude;
            var longitude = position.coords.longitude;
            var location = window.location.pathname.substr(1);
            $.ajax({
                      url: "/api/pull.php",
                      type: 'POST', //I want a type as POST
                      data: {'latitude': latitude, 'longitude' : longitude,
                            'location' : location },
                      success: function(response) {
                         $('#log').html(response);
                      }
              });
            }
          },1500);
         });
</script>
```

jQuery representation of “http://piperpal.com/concert”

Piperpal provides a jQuery function with a variable with a the list of all location entries for a given location tag nearby a geographical position for the device.
This jQuery function can then be used by third-party developers who want to show all entries for a given location-based tag entry accepted on piperpal.com.

SQL Query for computing shortest Haversine distance

The distance along the surface of the (spherical) Earth between two arbitrary points, in degrees is determined by the Spherical Cosine Law, also known as the Haversine Formula. We use this law to compute the nearest entries on piperpal.com measured from the user’s geolocation.

```
SELECT DISTINCT
id,name,service,location,modified,created,glat,glon,pa
id,token,type,email,111.045*DEGREES(ACOS(COS(RADIANS(l
atpoint))*COS(RADIANS(glat))*COS(RADIANS(longpoint)-RA
DIANS(glon))+SIN(RADIANS(latpoint))*SIN(RADIANS(glat
)))) AS distance_in_km FROM piperpal JOIN (SELECT  " .
$_POST['latitude'] . " AS latpoint, " .
$_POST['longitude'] . " AS longpoint) AS p ON 1=1
WHERE name = '" . $_POST['location'] . "' ORDER BY
distance_in_km;
```

The jQuery function on “http://piperpal.com/concert”

```
$(function(){ var locations = [ { id: '6', name:
'concert', service: 'Music', location:
'http://www.hallandoates.com/tour.html?dc_id=793',
modified: '2015-08-26 19:34:25', created: '2015-08-26
19:34:25', glat: '37.87901440, glon: '-122.26243000',
paid: '1', token: 'tok_16e7n0AZBHUS3EAZeIekv0Ii',
type: 'card', email: 'oka@oka.no' },
];
```

Future Piperpal work

Convolutional Neural Net to group the nearby points.

Plan to do lookups with a Convolution Neural Network [CNN] using a matrix with all of the entries within a radius away from the user’s location for a given time as computed by a convolution matrix for the longitude and latitude.

$$CNN(lat, lon) = \left[\int_{-90}^{90} F(lat) \left[G(v) e^{2\pi i v lat - lat'} dv \right] dlat', \int_{-180}^{180} F(lon) \left[G(v) e^{2\pi i v lon - lon'} dv \right] dlon' \right]$$

(Actual implementation may differ from the equation.)

Provide control panel that costs $25 per year/usage.

Wizard to define service or offer.

Service Category
Example: Shoes or clothing. Sports clothing/shoes.
Describe your service/offer: Shoes
Example: Welcome to Oslo’s largest flip flop offers.

Upload/Connect image, url, video, audio.

Decide amount of clicks and enter the email address for accessing statistics.

Payment: Stripe / Paypal / Google Wallet / Android Pay

Statistics

References

[CNN] Amani V. Peddada, James Hong: Geo-Location Estimation with Convolutional Neural Networks http://cs231n.stanford.edu/reports/CS231N_Final_Report_amanivp_jamesh93.pdf