

The `<location>` tag and `piperpal.com`

We propose the new HTML tag, the *location tag* `<location></location>`, structured in accordance with the W3C HTML standard for elements in a modern web browser and on an indexing client. The location tag would contain the following attributes with types defined in RELAX NG (<http://books.xmlschemata.org/relaxng/page2.html>)

name - string of characters (xsd:string)
service - string of characters (xsd:string)
duration - integer or float value (xsd:float)
latitude - float value (xsd:float)
longitude - float value (xsd:float)
altitude - float value (xsd:float)
accuracy - float value (xsd:float)
altitudeAccuracy - float value (xsd:float)
heading - string of characters (xsd:string)
description - string of characters (xsd:string)
address - string of characters (xsd:string)
correspondence - string of characters (xsd:string)
highlights - string of characters (xsd:string)
aka - string of characters (xsd:string)
txt - semicolon separated words in a string of characters (xsd:string)

As an example, we offer:

```
<location
  name="office"
  latitude="42.361860"
  longitude="-71.091840"
  altitude="6.706"
  accuracy="20.5"
  altitudeAccuracy="1.6"
  heading=""
  speed=""
  description="W3C MIT office"
  address="32 Vassar Street, Cambridge, MA 02139 USA"
  correspondence="public-html@w3.org"
  highlights="World Wide Web Museum"
  aka="W3C Headquarters"
  txt="watch:Apple;phone:Motorola">W3C MIT Office</location>
```

Piperpal will provide the testing platform for the new <location> tag element and <http://facebook.com/geolocation> will serve as the hub for discussion and feedback during the marketing phase.

Four aspects of a proposed business model revolving around the proposed standard are described in this document:

1. involving the creation of data sets (Sets) through form entries from customers who pay to interrelate digital resources (images, text, hyperlinks etc.) with a geolocation coordinate obtained automatically from the submitting party's device or manually entered/map indicated.
2. involving direct queries by users in the Piperpal Search Engine handled by the Piperpal Search Algorithm operating on Piperpal's stored Sets
3. involving calls to Piperpal's api from content vendors for an original, submitted document to be appended with valid location tags, hereunder the automated selection of Sets from which to generate said location tags and the automated creation of syntax correct location tags from the qualifying (selected) Sets
4. involving the construction of map overlays based on the sets of interrelated

Customers may set location tag attribute values through Piperpal's API (1) against payment. An online form will allow discrete attribute value settings and the input will be stored in SQL database structures as *Atoms*. *Atoms* garnered in a particular entry comprise *Sets* (of interrelated attribute values).

In the case of direct queries to its search engine (2), typed queries or submitted positions may instigate searches.

Piperpal's functionality will allow users to have their position sampled at set rate intervals, thus instantiating new searches by virtue of movement alone. Search results in all events will be ranked by distances and/or attribute weighing and the algorithm may also consider a user's path and previously stated or inferred preferences. The step concerning an evaluation of the relationship between a user's location and the coordinates specified in the *Sets* is described in (3) below.

In terms of the representation of search results, a user may receive structured information as a result of a query, or series of queries, both in the form of listings and map overlays (4).

Piperpal will also handle calls to its api from content vendors (3).

In the case of calls to api.piperpal.com from content vendors, Piperpal run an algorithm (Palgora©) for the creation of valid, syntax correct location tags (SCLT) from Sets.

Execution of the SCLT creation from the attribute atoms of Sets will be contingent on the algorithm's assessment of an implied relationship between Sets and the geographical location of the user.

More specifically, the algorithm evaluates a *relative distance* between Sets and the user to identify qualifying Sets.

The relative distance is computed from two sets of GPS coordinates:

- 1) the user's as submitted by content vendor in the call to Piperpal's API
- 2) the *latitude* and *longitude* atoms in the Sets

and held against the validity radius specified for the Set by the *radius* atom.

The attribute atoms from a Set are extracted and added to syntax in concordance with the location tag standard for the creation of an SCLT if the user's distance from the coordinates specified for the Set by its *latitude* and *longitude* atoms is smaller than the radius of propagation conferred upon the Set as indicated by the *radius* attribute atom.

The resulting SCLT is then amended according to content vendor's delineation of attributes, resulting in a *Tag to Append* (TTA).

Content vendor may read and perform operations on the data contained in the TTA. Such operations could include performing *intra mural* searches from strings contained in the TTA for the structured display of information in response to a user's original query/request for content.

```
Client -> (glat,glon)      -> Location      <- Content Server
Opera/Firefox/Android     -> Piperpal      <- Foobar Times
Query                     Response         Result
POST / HTTP/1.1          -> 200 OK       <- GET /Music HTTP/1.1
Host: piperpal.com                               Host: piperpal.com
Content-Length: 24
glat=60&glon=10&grad=10
```

Map Overlays : Vectors and raster images based on a location, ala
YayLocation <http://yay.oka.no/>

Computing Piperpal's convolution matrix for a given (glat,glon) pair,
time period and service parameters:

<http://www.piperpal.com/api/location/json.php?service=Music&glat=60&glon=10>

```
Piperpal(lat=37.87903460,lon=-122.26244120,service=Music) = [
{"id": "5", "name": "hallandoates", "service": "Music", "location":
"http://www.hallandoates.com/tour.html?dc_id=793", "modified":
"2015-08-26 18:26:26", "created": "2015-08-26 18:26:26", "glat":
"37.87903460", "glon": "-122.26244120", "paid": "1", "token":
"tok_16e6jBAZBHUS3EAZnO0OMegA", "type": "card", "distance":
"8278.911686556956", "email": "oka@oka.no"}
]
```