GÉOLOCALISATION & WORDPRESS

WILLY BAHUAUD

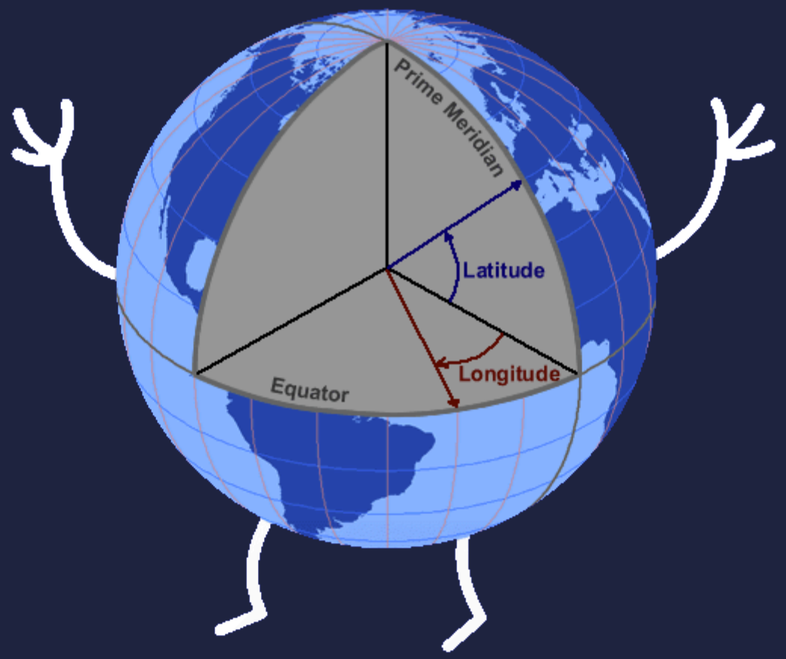
Développeur / intégrateur / designer

BLOG WABEO.FR

ASSO WP TECH

ASSO WPFR

QU'EST CE QUE LA GÉOLOCALISATION ?



PROPOSER A UN VISITEUR DES CONTENUS EN FONCTION D'UN EMPLACEMENT GÉOGRAPHIQUE QU'IL FOURNI

ANNUAIRE / CARTE DE POINTS D'INTÉRÊT À PROXIMITÉ / AGENCE IMMOBILIÈRE / STORE LOCATOR / PETITES ANNONCES / JOBBOARD / SITE DE RENCONTRE / SITE DE PRESSE / ...

DES SOLUTIONS TOUTES FAITES

Thèmes / Plugins

L'AVANTAGE DU SUR-MESURE

DEMO TIME

QUE VOUS POUVEZ TÉLÉCHARGER SUR WABEO.FR/CONF-GEOLOC/CODE.ZIP

RÉCUPÉRER LES COORDONNÉES GPS DE SES INTERNAUTES

SOUMISSION DE FORMULAIRE

HTTPS://MAPS.GOOGLEAPIS.COM/MAPS/API/GEOCODE/JSON?ADDRESS=NANTES

ANALYSE DE L'IP AVEC *Geo-IP*

3 FOURNISSEURS (PARMI D'AUTRES...) :

IP2LOCATION / DB-IP / GEOLITE

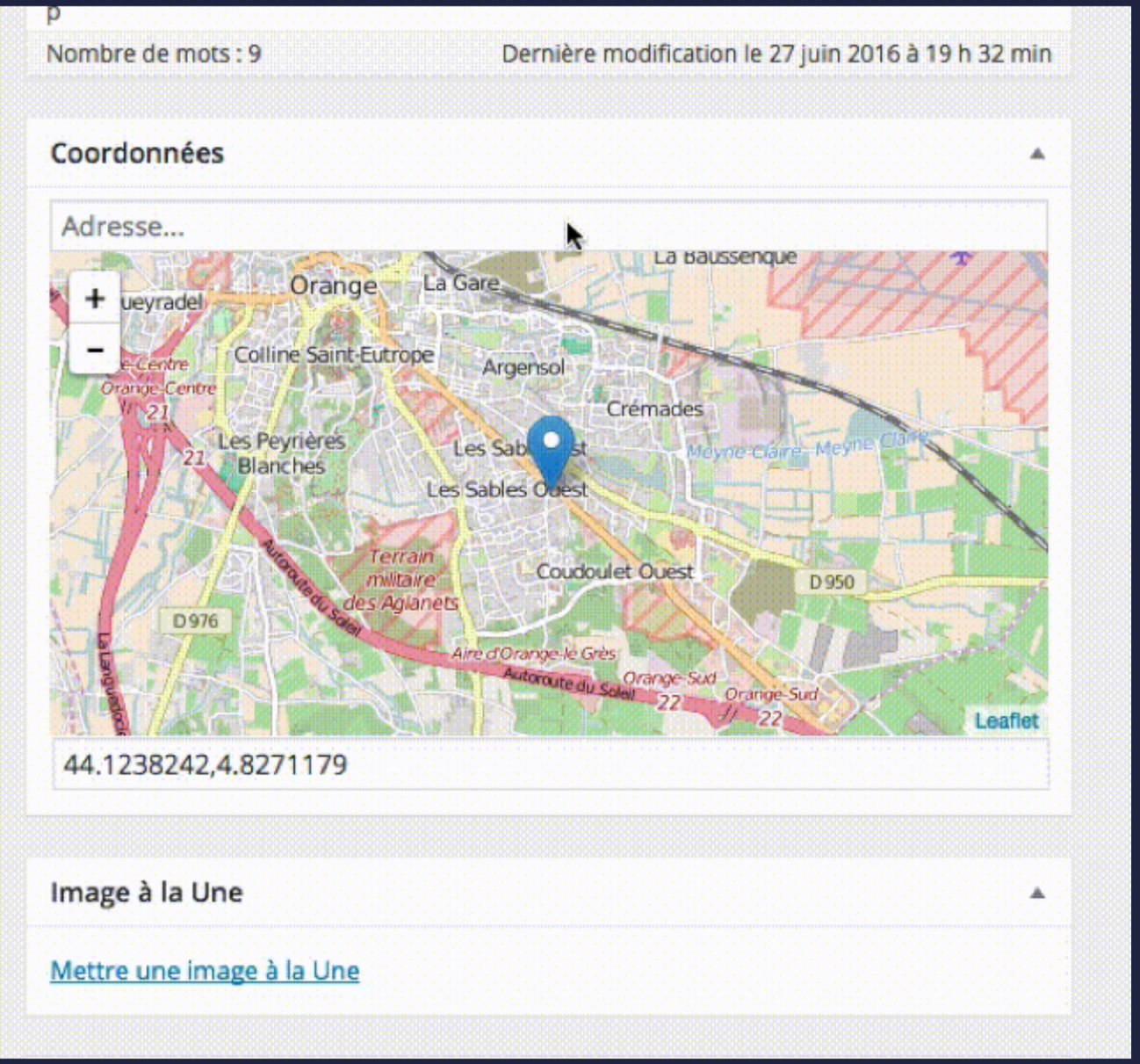
XXX.XXX.XXX.XXX → COORDONNÉES GPS DE LA VILLE

API HTML5 GEOLOCATION

```
if ( navigator.geolocation ) {
    // ok ?
    navigator.geolocation
    .getCurrentPosition( callback, erreur );
} else {
    alternative(); //autre solution
}

function callback( position ) {
    var lat = position.coords.latitude;
    var lng = position.coords.longitude;
    console.log( lat, lng );
}
```

COMMENT ASSIGNER DES COORDONNÉES GPS À SES CONTENUS ?

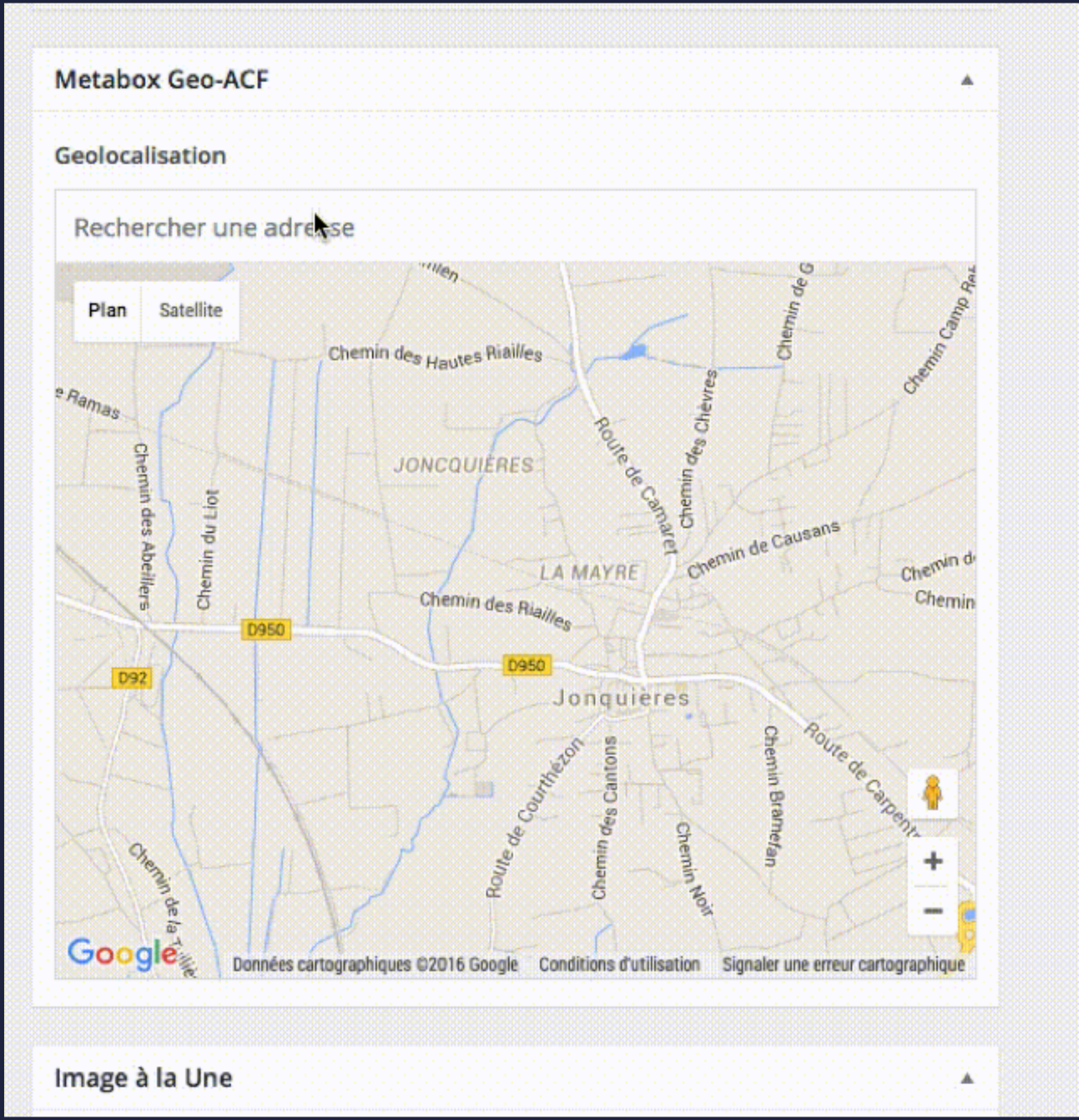


META BOX SUR-MESURE

(HTTPS://WABEO.FR/METABOX-GEOLOCALISATION-V2/)

META BOX
ACF

STOCK LES COORDONNÉES SOUS FORME DE TABLEAU SÉRIALISÉ



ISOLER LATITUDE ET LONGITUDE AVEC ACF

```
add_action( 'update_postmeta', 'tg_update_postmeta', 10, 4 );
function tg_update_postmeta( $id, $post_id, $key, $value ) {
    if ( 'acf_meta_key' == $key ) {
        $value = maybe_unserialize( $value );
        remove_action( 'update_postmeta', 'tg_update_postmeta' );

        update_post_meta( $post_id, 'lat', $value['lat'] );
        update_post_meta( $post_id, 'lng', $value['lng'] );

        add_action( 'update_postmeta', 'tg_update_postmeta', 10, 4 );
    }
}
```

GEO QUERY

LA MÉTHODE POUR SERVIR LES CONTENUS EN FONCTION DE LEURS POSITIONS GPS

$$\text{RAYON TERRE} \times \text{ACOS}(\text{COS}(lat1) \times \text{COS}(lat2) \times \text{COS}(lng2 - lng1) + \text{SIN}(lat1) \times \text{SIN}(lat2))$$

EXEMPLE DE GEO_QUERY

```
new WP_Query( array(
    'post_type' => 'post',
    'geo_query' => array(
        'lat'       => 41.7,
        'lng'       => -1.1,
        'compare'   => '<='
    ),
    'orderby'   => 'distance'
) );
```

Passer de ça...

```
SELECT wp_posts.*
FROM wp_posts
WHERE 1=1
AND wp_posts.post_type = 'post'
```

… à ça :

```sql
SET @lat_user = 48.85; -- latitude
SET @lng_user = 2.35; -- longitude
SET @rayon_terre = 6371; -- rayon terre en km
SELECT wp_posts.*, ( @rayon_terre * acos( cos( radians( @lat_user ) )
* cos( radians( metalat.meta_value ) )
* cos( radians( metalng.meta_value ) - radians( @lng_user ) )
+ sin( radians( @lat_user ) )
* sin( radians( metalat.meta_value ) ) ) ) AS distance
FROM wp_posts
INNER JOIN wp_postmeta AS metalat
ON ( wp_posts.ID = metalat.post_id )
INNER JOIN wp_postmeta AS metalng
ON ( wp_posts.ID = metalng.post_id )
WHERE 1=1
AND wp_posts.post_type = 'post' -- cpt
AND metalat.meta_key = 'lat' -- clé latitude
AND metalng.meta_key = 'lng' -- clé longitude
```

WP_SEARCH_STOPWORDS, POSTS_SEARCH_ORDERBY, SPLIT_THE_QUERY, POSTS_REQUEST_IDS, OLD_SLUG_REDIRECT_URL, POSTS_SEARCH, POSTS_WHERE, POSTS_JOIN, COMMENT_FEED_JOIN, COMMENT_FEED_WHERE, COMMENT_FEED_GROUPBY, COMMENT_FEED_ORDERBY, COMMENT_FEED_LIMITS, POSTS_WHERE_PAGED, POSTS_GROUPBY, POSTS_JOIN_PAGED, POSTS_ORDERBY, POSTS_DISTINCT, POST_LIMITS, POSTS_FIELDS, POSTS_CLAUSES, POSTS_WHERE_REQUEST, POSTS_GROUPBY_REQUEST, POSTS_JOIN_REQUEST, POSTS_ORDERBY_REQUEST, POSTS_DISTINCT_REQUEST, POSTS_FIELDS_REQUEST, POST_LIMITS_REQUEST, POSTS_CLAUSES_REQUEST, POSTS_REQUEST, POSTS_RESULTS, COMMENT_FEED_JOIN, COMMENT_FEED_WHERE, COMMENT_FEED_GROUPBY, COMMENT_FEED_ORDERBY, COMMENT_FEED_LIMITS, THE_PREVIEW, THE_POSTS, FOUND_POSTS_QUERY, FOUND_POSTS

WP_SEARCH_STOPWORDS, POSTS_SEARCH_ORDERBY, SPLIT_THE_QUERY, POSTS_REQUEST_IDS, OLD_SLUG_REDIRECT_URL, POSTS_SEARCH, POSTS_WHERE, POSTS_JOIN, COMMENT_FEED_JOIN, COMMENT_FEED_WHERE, COMMENT_FEED_GROUPBY, COMMENT_FEED_ORDERBY, COMMENT_FEED_LIMITS, POSTS_WHERE_PAGED, POSTS_GROUPBY, POSTS_JOIN_PAGED, POSTS_ORDERBY, POSTS_DISTINCT, POST_LIMITS, POSTS_FIELDS, POSTS_CLAUSES, POSTS_WHERE_REQUEST, POSTS_GROUPBY_REQUEST, POSTS_JOIN_REQUEST, POSTS_ORDERBY_REQUEST, POSTS_DISTINCT_REQUEST, POSTS_FIELDS_REQUEST, POST_LIMITS_REQUEST, POSTS_CLAUSES_REQUEST, POSTS_REQUEST, POSTS_RESULTS, COMMENT_FEED_JOIN, COMMENT_FEED_WHERE, COMMENT_FEED_GROUPBY, COMMENT_FEED_ORDERBY, COMMENT_FEED_LIMITS, THE_PREVIEW, THE_POSTS, FOUND_POSTS_QUERY, FOUND_POSTS

POSTS_FIELDS

```
add_filter( 'posts_fields', 'willy_geo_fields', 10, 2 );
function willy_geo_fields( $fields, $q ) {
    if ( isset( $q->query_vars['geo_query']['lat'],
          $q->query_vars['geo_query']['lng'],
          $q->query_vars['geo_query']['distance'] ) ) {

        global $wpdb;
        $fields .= $wpdb->prepare(", ( 6371 * acos( cos( radians( %f ) )
                        * cos( radians( metalat.meta_value ) )
                        * cos( radians( metalng.meta_value ) - radians( %f ) )
                        + sin( radians( %f ) )
                        * sin( radians( metalat.meta_value ) ) ) ) AS distance",
                      $q->query_vars['geo_query']['lat'],
                      $q->query_vars['geo_query']['lng'],
                      $q->query_vars['geo_query']['lat'] );
    }
    return $fields;
}
```

POSTS_JOIN

```
add_filter( 'posts_join', 'willy_geo_join', 10, 2 );
function willy_geo_join( $join, $q) {
    if ( isset( $q->query_vars['geo_query']['lat'],
          $q->query_vars['geo_query']['lng'],
          $q->query_vars['geo_query']['distance'] ) ) {
        global $wpdb;
        $join .= " INNER JOIN $wpdb->postmeta AS metalat
            ON ( $wpdb->posts.ID = metalat.post_id )";
        $join .= " INNER JOIN $wpdb->postmeta AS metalng
            ON ( $wpdb->posts.ID = metalng.post_id )";
    }
    return $join;
}
```

POSTS_WHERE

```
add_filter( 'posts_where', 'willy_geo_where', 10, 2 );
function willy_geo_where( $where, $q ) {
    if ( isset( $q->query_vars['geo_query']['lat'],
          $q->query_vars['geo_query']['lng'],
          $q->query_vars['geo_query']['distance'] ) ) {

        global $wpdb;
        $where .= " AND metalat.meta_key = 'lat'";
        $where .= " AND metalng.meta_key = 'lng'";
    }
    return $where;
}
```

POSTS_GROUPBY

```
add_filter( 'posts_groupby', 'willy_geo_distance', 10, 2 );
function willy_geo_distance( $groupby, $q ) {
    if ( isset( $q->query_vars['geo_query']['lat'],
          $q->query_vars['geo_query']['lng'],
          $q->query_vars['geo_query']['distance'] ) ) {

        $compare = '<=';
        if ( isset( $q->query_vars['geo_query']['compare'] )
          && in_array(
            $q->query_vars['geo_query']['compare'],
            array( '<', '<=', '>', '>=' ) ) ) {
                $compare = $q->query_vars['geo_query']['compare'];
        }
        global $wpdb;
        $groupby .= $wpdb->prepare( "$wpdb->posts.ID HAVING distance $compare %d",
                      $q->query_vars['geo_query']['distance'] );
    }
    return $groupby;
}
```

POSTS_ORDERBY

```
add_filter( 'posts_orderby', 'willy_geo_orderby_distance', 10, 2 );
function willy_geo_orderby_distance( $orderby, $q ) {
    if ( isset( $q->query_vars['orderby'],
          $q->query_vars['geo_query']['lat'],
          $q->query_vars['geo_query']['lng'],
          $q->query_vars['geo_query']['distance'] )
      && 'distance' == $q->query_vars['orderby'] ) {

        $order = in_array(
            $q->query_vars['order'],
            array( 'ASC', 'DESC' ) )
            ? $q->query_vars['order'] : 'ASC';
        $orderby = 'distance ' . $order;
    }
    return $orderby;
}
```

PERFORMANCE DES REQUÊTES GÉOLOCALISÉES

Testé sur un site à 11000 contenus :
0.002s de temps d'exécution

Pistes d'amélioration :
- MySQL Geometry (`ST_Distance_Sphere`)
- Redis Geospatial index