

[Code-Web.org](http://www.code-web.org)

Développement web en toute liberté!

- Recherche
- [Blog](#)
- [Plugins DC2](#)
- [Contact](#)
- [À propos](#)

Google Map API V3 : Geocoding : Conversion d'adresses postales en coordonnées GPS



Nous allons étudier, dans ce billet, un point important proposé par l'**API Google Map**. En effet, nous avons vu jusque là comment manipuler les cartes à l'aide de coordonnées GPS. Mais, la plupart du temps, nous souhaiterons **utiliser des adresses postales**. Nous allons donc voir comment passer de l'une à l'autre donnée : cette opération s'appelle **géocoder une adresse**.

Pour aborder ce nouveau billet, vous devrez être à l'aise avec les langages souvent présenté sur ce site, à savoir **JavaScript** et/ou **Php**. Par ailleurs, vous devrez connaître les formats **JSon** et/ou **XML**.

Géocodeur et Web Service

Pour la conversion d'adresses postales en coordonnées GPS, ou **géocodage d'adresses**, nous allons utiliser ce que l'on appelle, dans le jargon, un **géocodeur**. Il en existe différents et Google nous propose le siens au travers de son API. Ce **géocodeur** est accessible sous la forme d'un **Web Service** qui fournira ses réponses aux formats **JSon** et **XML**.

Il est, par ailleurs, possible d'interroger ce **Web Service** aussi bien côté serveur que côté client. Pour le côté client, les requêtes seront formulées à l'aide de JavaScript. Pour des interrogations côté serveur, nous verrons ici comment procéder en PHP.

Avant de démarrer, il est important de noter que **ce service limite le nombre de requêtes par tranche de 24h et par IP à 15 000**. Pour des services à fort trafic, il faudra donc préférer les requêtes exécutés côté client. Cela-dit, vous pouvez demander une augmentation des limitations liées à votre adresse IP en prenant contact avec Google. Votre demande sera étudiée.

Géocodage côté client (JavaScript)

A priori, envoyer des requêtes **Ajax** depuis votre site vers le domaine **maps.google.com** ne paraît pas évident. En effet, la **Same origin policy** renverra systématiquement une erreur étant donné que vos requêtes seront envoyées vers un nom de domaine différent du votre. Sachez cependant qu'il existe des solutions de pour réaliser des **requêtes cross-domain** en JavaScript, mais, ce n'est pas l'objet de ce billet. En l'occurrence, nous utiliserons **simplement l'objet geocoder** intégré à l'**API Google Map**.

Cet objet vous permettra de ne pas vous encombrer la gestion des requêtes au serveur et de la lecture de ses

réponses. Il s'agit donc bien d'une API permettant d'interroger le Web Service Geocoder de Google Map.

Nous commençons très simplement par instancier notre Geocoder :

```
var myGeocoder = new google.maps.Geocoder();
```

Cet objet dispose d'une méthode *geocode* qui prends deux paramètres. Le premier correspond aux informations de localisation et le second à la fonction à appeler lorsque la requête est terminée (ou fonction de *callback*).

Nous allons d'abord étudier les informations de localisation que nous pouvons fournir à cette méthode. Comme à l'accoutumée, ces différentes options sont regroupées dans un objet statique :

- **address** : de type *string*, l'adresse postale complète à rechercher,
- **latLng** : de type *LatLng*, les coordonnées GPS à rechercher,
- **bounds** : de type *LatLngBounds*, la zone dans laquelle rechercher,
- **language** : de type *string*, la langue de réponse,
- **region** : de type *string*, le code du pays dans lequel rechercher.

Vous l'aurez peut être compris, le **géocodeur Google** permet de rechercher les coordonnées **GPS** d'une adresse postale, mais aussi, inversement l'adresse postale correspondant à des coordonnées GPS.

Ainsi, on ne peut pas dire que *address* et *latLng* sont des options obligatoires, mais au moins l'une ou l'autre doivent être renseignées en fonction du type de recherche effectuée.

L'option *language* définit la langue dans laquelle sera formulée l'adresse postale retournée. Cette option n'est donc pas utile pour géocoder une adresse postale.

Enfin, les options *bounds* et *region* influencent la recherche mais sans la restreindre complètement. L'option *region* prends en paramètre un code de pays (FR pour la France par exemple). Ce code permet au géocodeur d'identifier une recherche provenant de France, favorisant ainsi les résultats de recherches situés en France en cas de doute.

Nous allons donc préparer nos options pour géocoder une adresse. Ici, je vous propose de localiser le Palais de l'Élysée où vous pourrez envoyer vos courriers à monsieur le président :

```
var GeocoderOptions = {
  'address' : 'Palais de l'Elysée 55, rue du faubourg Saint-Honoré 75008 Paris',
  'region' : 'FR'
}
```

Nous avons donc maintenant renseigné toutes les informations requises pour effectuer une recherche. Cela-dit, nous ne serons pas encore en mesure de traiter le résultat de celle-ci. Pour ce faire, nous allons employer une fonction de *callback* que nous passerons comme second paramètre à la méthode *geocode* du géocodeur de l'API Google.

Cette fonction prendra deux paramètres :

- **results** qui contiendra les résultats de la recherche,
- **status** qui nous renseigne sur l'état finale de la requête.

Nous pouvons donc écrire la fonction suivante pour traiter le résultat :

```
function GeocodingResult( results , status )
{
  // Traitement du résultat
}
```

Naturellement, cette fonction ne fait rien pour l'instant. Vous pouvez éventuellement lui ajouter un `console.log(results , status)` pour visualiser ce que vous retourne le géocodeur. Enfin, nous envoyons notre demande de géocodage de l'adresse au serveur Google :

```
myGeocoder.geocode( GeocoderOptions, GeocodingResult );
```

Étant donné que la fonction que nous avons donné à l'objet de géocodage des adresses ne fait rien, vous ne verrez rien se passer. Je vous propose donc maintenant un exemple plus complet ([voir le résultat](#)) de ce que nous pouvons faire en [créant une carte](#) puis en géocodant l'adresse du palais de l'Élysée. Nous plaçons ensuite un [marker](#) sur les coordonnées que nous retourne le serveur et nous centrons enfin la vue sur ce marker.

```
window.onload = function() {
    // Une variable pour contenir notre future marker
    var myMarker = null;

    // Des coordonnées de départ
    var myLatLng = new google.maps.LatLng(-34.397, 150.644);

    // Les options de notre carte
    var myOptions = {
        zoom: 15,
        center: myLatLng,
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

    // On crée la carte
    var myMap = new google.maps.Map(
        document.getElementById('map'),
        myOptions
    );

    // L'adresse que nous allons rechercher
    var GeocoderOptions = {
        'address' : 'Palais de l'Elysée 55, rue du faubourg Saint-Honoré 75008 Paris',
        'region' : 'FR'
    };

    // Notre fonction qui traitera le resultat
    function GeocodingResult( results , status )
    {
        // Si la recher à fonctionné
        if( status == google.maps.GeocoderStatus.OK ) {

            // S'il existait déjà un marker sur la map,
            // on l'enlève
            if(myMarker != null) {
                myMarker.setMap(null);
            }

            // On crée donc un nouveau marker sur l'adresse géocodée
            myMarker = new google.maps.Marker({
                position: results[0].geometry.location,
                map: myMap,
                title: "Palais de l'Elysée"
            });
```

```

// Et on centre la vue sur ce marker
myMap.setCenter(results[0].geometry.location);

} // Fin si status OK

} // Fin de la fonction

// Nous pouvons maintenant lancer la recherche de l'adresse
var myGeocoder = new google.maps.Geocoder();
myGeocoder.geocode( GeocoderOptions, GeocodingResult );
}

```

Je pense que cet exemple est suffisamment parlant pour se passer d'explications. Ce que vous devez savoir, c'est que le géocodeur pourra éventuellement vous retourner plusieurs résultats. C'est pourquoi *results* est un tableau et que nous traitons ici le premier résultat. La variable *status* pourra vous remonter des erreurs dont vous trouverez la liste sur [la documentation officielle de l'API](#).

Géocodage côté serveur (PHP)

Le géocoding d'adresse côté serveur n'est pas très compliqué. Il vous suffit d'envoyer une requête vers le serveur de géocodage de Google avec les bons paramètres et d'analyser la réponse qu'il vous retourne. L'adresse pour interroger le géocodeur est de la forme suivante :

```
http://maps.googleapis.com/maps/api/geocode/[FORMAT]?address=[ADRESSE]&sensor=false
```

Encore une fois, nous retrouvons la variable **sensor**. Celle-ci indique si votre visiteur est localisé à l'aide d'un GPS, vous devrez donc passer sa valeur à *true* si tel est le cas sinon laissez *false*. Vous pouvez remplacer [FORMAT] par *json* ou *xml* selon le format de données que vous préférez. Remplacez enfin [ADRESSE] par l'adresse que vous souhaitez rechercher. Pour rechercher le palais de l'Élysée dans le format *json*, nous obtenons donc l'URL suivante (cliquez pour voir le résultat) :

```
http://maps.googleapis.com/maps/api/geocode/json?address=Palais de l'Elysée 55, rue du faubourg
Saint-Honoré 75008 Paris&sensor=false
```

Avec ces informations, il devient alors très simple de retrouver et de traiter le résultat avec du code PHP. Je vous propose donc un exemple simple :

```
<?php
```

```
// On prépare l'adresse à rechercher
```

```
$address = "Palais de l'Elysée 55, rue du faubourg Saint-Honoré 75008 Paris";
```

```
// On prépare l'URL du géocodeur
```

```
$geocoder = "http://maps.googleapis.com/maps/api/geocode/json?address=%s&sensor=false";
```

```
// Pour cette exemple, je vais considérer que ma chaîne n'est pas
```

```
// en UTF8, le géocoder ne fonctionnant qu'avec du texte en UTF8
```

```
$url_address = utf8_encode($address);
```

```
// Penser à encoder votre adresse
```

```
$url_address = urlencode($url_address);
```

```
// On prépare notre requête
```

```
$query = sprintf($geocoder,$url_address);
```

```
// On interroge le serveur
$results = file_get_contents($query);

// On affiche le résultat
var_dump(json_decode($results));
```

Cet exemple suppose que vous souhaitez utiliser le format **json**, mais vous pouvez aussi modifier l'URL pour obtenir une réponse en **XML** auquel cas vous pourrez utiliser [SimpleXML](#) par exemple pour traiter la réponse.

[<< Sommaire des articles Google Map API V3](#)

Billets similaires

- [Découverte de l'API Google Map V3](#)
- [Google Map API V3 : Les différents types de cartes](#)
- [Google Map API V3 : Contenu de la carte complètement personnalisé](#)
- [Google Map API V3 : Placer un marker sur la carte](#)

Par [Thierry Geindre](#) | lundi 4 février 2013 | 15:04

Commentaires

[aucun commentaire](#)

Catégorie

› [Java Script](#)

Tags

[api](#), [google](#), [google map](#), [v3](#)

 [Fil des commentaires de ce billet](#)

 [SHARE](#)    ...

« [Tracker la recherche interne avec Google Analytics](#) [7 astuces pour diminuer le taux de rebond de votre](#) »



Ajouter un commentaire

Nom ou pseudo :

Adresse email :

Site web (facultatif) :

Commentaire :

Le code HTML est affiché comme du texte et les adresses web sont automatiquement transformées.

Antispam : 10 + 1 =

[prévisualiser](#)

Catégories

- [Html / CSS](#) (9)
- [Java Script](#) (8)
- [PHP](#) (9)
- [Dotclear](#) (18)
- [Référencement](#) (4)
- [Actualité](#) (18)
- [Web Design](#) (4)
- [Ressources](#) (7)
- [Code-Web.org](#) (9)
- [Système](#) (1)
- [Technologie](#) (2)
- [Hébergement](#) (5)

Billets similaires

- [Découverte de l'API Google Map V3](#)
- [Google Map API V3 : Les différents types de cartes](#)
- [Google Map API V3 : Contenu de la carte complètement personnalisé](#)
- [Google Map API V3 : Placer un marker sur la carte](#)

Partenaires

[Tutoriels Jailbreak iPhone](#)

[Désimlocker gratuitement votre iPhone](#)

[Forum Francophone sur le blogging](#)


Sondage

Vous venez souvent ?

- ☐ C'est la première fois.
- ☐ De temps en temps.
- ☐ Oui souvent !
- ☐ Je suis complètement accroc !

[Voter !](#)[Résultats](#)

Publicité

Choisir sa pub 

Cartes Gratuites

www.MapsGalaxy.com
Itinéraire Instantanément
Télécharger l'application!

Téléchargez Google Chrome

www.google.com/chrome
Le Navigateur Gratuit Qui
Vous Permet D'en Faire
Plus Sur Le Web !

Les Tarifs des Mutuelles

Santiane.fr/Comparateur_...
Trouvez les Meilleurs
Contrats Comparez les
Formules, Economisez!

Opération Des Yeux Laser

expert-vision-center.com/
Les Dernières Technologies
Laser Pour Corriger Votre
Vue

Chirurgie de la myopie

www.cliniquedelavision.com
Lasik, Femto, Aberromètre
Sécurité, Précision,
Expérience

Tags

- [widget](#)
- [hosting](#)
- [automatisation](#)
- [hébergement](#)
- [google](#)
- [css](#)
- [api](#)
- [google map](#)
- [v3](#)
- [apache](#)
- [actualité](#)
- [accessibilité](#)
- [référencement](#)
- [plugin](#)
- [design](#)
- [php](#)
- [code-web.org](#)
- [html](#)
- [ressources](#)
- [dotclear](#)

[Tous les tags](#)

[Annoncer sur Code-web.org](#) - Propulsé par [Dotclear](#) - Thème Freshy de [Julien de Luca](#) adapté depuis Wordpress