

Géolocalisez vos visiteurs grâce à leur IP

Par matheod



www.openclassrooms.com

*Licence Creative Commons 6 2.0
Dernière mise à jour le 17/08/2012*

Sommaire

Sommaire	2
Lire aussi	1
Géolocalisez vos visiteurs grâce à leur IP	3
Récupérer la base de données GeoLite MaxMind	3
Utiliser l'API PHP	4
Quelques précisions... ..	4
Obtenez une adresse grâce à Google Maps	5
Obtenir des informations plus poussées	6
Partager	8



Géolocalisez vos visiteurs grâce à leur IP



Par

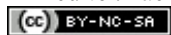
matheod

Mise à jour : 17/08/2012

Difficulté : Facile



Durée d'étude : 2 heures



N'avez-vous jamais eu envie de géolocaliser vos visiteurs ? Pour faire des statistiques, préremplir des formulaires, détecter des multicomptes, placer vos membres sur une carte, ou bien tout juste par curiosité ?

Peu importe votre motif, vous apprendrez à géolocaliser vos visiteurs très simplement, et sans dépendre directement de services externes.

Sommaire du tutoriel :



- Récupérer la base de données GeoLite MaxMind
- Utiliser l'API PHP
- Obtenez une adresse grâce à Google Maps

Récupérer la base de données GeoLite MaxMind

MaxMind est une société qui propose un service de géolocalisation. Si cette entreprise propose une version payante de ses services, elle propose également une version gratuite pour les personnes ne pouvant se permettre de se payer une licence ou souhaitant tester leur solution. Celle-ci, bien que moins précise, apporte tout de même une bonne précision au niveau de la ville. C'est donc à partir de GeoLite (le nom de la version gratuite) que se basera ce tutoriel, mais rien ne vous empêche de faire de même avec la version payante.

Tout le système repose sur une base de données, c'est ce que GeoLite fournit. Mais que contient cette base de données ? Eh bien elle relie des pages d'adresses IP à des locations.



Seules les IP fixes, c'est-à-dire qui ne changent pas, peuvent être géolocalisées. Les IP dynamiques retourneront des résultats très différents de la réalité.

Il faut donc tout d'abord télécharger cette base de données d'environ 20 Mo. Celle-ci sera utilisée à l'aide d'une API. Différentes API sont disponibles (C, PHP, Java, Perl, Python...) dont vous trouverez de l'aide, en anglais, [ici](#). Dans ce tutoriel, je vous expliquerai comment utiliser l'API PHP.

Voici maintenant, le lien que vous attendez tous... le lien de téléchargement de la base de données GeoLite. Eh bien, le [voici](#) !



Heu merci, mais j'en fais quoi de ton fichier après ?

Du calme, du calme !

On va commencer par le décompresser, ce sera déjà un bon début. 🤪

Pour ce faire, vous devez utiliser un logiciel tel que Winzip, Winrar, ou 7-Zip que vous pouvez trouver en version portable [ici](#).

Citation : Site web de GeoLite



Under the license agreement, all advertising materials and documentation mentioning features or use of this database must display the following acknowledgment: "This product includes GeoLite data created by MaxMind, available from <http://www.maxmind.com/>."

Vous devez donc préciser sur votre site que vous utilisez la base de données Geolite et faire un lien vers [maxmind](http://www.maxmind.com/).

Utiliser l'API PHP

Je vous conseille avant toute chose de créer un dossier `geoloc` où l'on y mettra tous les fichiers nécessaires à la géolocalisation, cela évitera de vous perdre ~~sur tout si vous êtes aussi bordéliques que moi.~~

Tout d'abord, *upload*ez votre fichier `GeoLiteCity.dat` dans votre dossier.



`GeoLiteCity.dat` est un fichier binaire, si l'*upload* échoue, choisissez le type de transfert « binaire » dans votre logiciel FTP. Pensez à le remettre en mode ASCII juste après.

Maintenant, téléchargez [l'API PHP](#) et décompressez là. Nous allons avoir besoins des fichiers suivant :

- `geoip.inc`
- `geoipcity.inc`
- `geoipregionvars.php`

Ces fichiers sont nécessaires pour extraire les informations de la base de données, et vous permettre de les récupérer facilement. Mettez-les dans votre dossier `geoloc`.

Créez maintenant, toujours dans le dossier `geoloc`, un fichier `exemple.php` qui contient le code suivant.

Code : PHP - exemple.php

```
<?php
include("geoipcity.inc");
include("geoipregionvars.php");

$gi = geoip_open(realpath("GeoLiteCity.dat"), GEOIP_STANDARD);

$record = geoip_record_by_addr($gi, $_SERVER['REMOTE_ADDR']);

echo $record->country_name . "\n";
echo $GEOIP_REGION_NAME[$record->country_code][$record->region] .
"\n";
echo $record->city . "\n";
echo $record->postal_code . "\n";
echo $record->latitude . "\n";
echo $record->longitude . "\n";

geoip_close($gi);

?>
```

Et accédez à ce fichier via votre navigateur, vous devriez voir votre localisation ou dans le cas d'une IP dynamique, des informations erronées.



Si votre site tourne en local, à l'aide de [Wamp](#) par exemple, vous vous retrouverez face à une page vide vu que `<?php $_SERVER['REMOTE_ADDR'] ; ?>` retournera `127.0.0.1`.

Quelques précisions...

Code : PHP - exemple.php

```
<?php
$record = geoip_record_by_addr($gi, $_SERVER['REMOTE_ADDR']);
```

Cette fonction nous permet de créer un objet contenant toutes les informations de localisation. Cette fonction prend deux paramètres :

- la variable contenant la base de données ;
- l'adresse IP à géolocaliser.



Ici, `<?php $_SERVER['REMOTE_ADDR']; ?>` est utilisée. Cette variable superglobale contient l'adresse IP de la personne qui affiche la page.

Mais nous aurions tout aussi bien pu mettre par exemple l'adresse IP 12.210.21.121 :

Code : PHP - exemple.php

```
<?php
$record = geoip_record_by_addr($gi, "12.210.21.121");
```

Les informations offertes par la géolocalisation sont les suivantes :

Code : PHP - exemple.php

```
<?php
echo $record->country_name . "\n"; // nom du pays
echo $GEOIP_REGION_NAME[$record->country_code][$record->region] .
"\n"; // nom de la region
echo $record->city . "\n"; // nom de la ville
echo $record->postal_code . "\n"; // code postal
echo $record->latitude . "\n"; // latitude
echo $record->longitude . "\n"; // longitude
```



`$record->region` ne retourne pas le nom de la région !

En effet, il faut utiliser `<?php $GEOIP_REGION_NAME[$record->country_code][$record->region]; ?>` qui se base sur `$record->country_code` ainsi que sur `$record->region` qui lui renvoie un identifiant de région.

Il existe également `$record->country_code`, `$record->country_code3`, `$record->area_code`, `$record->metro_code` et `$record->continent_code`.



Certaines informations, comme `$record->postal_code` ne fonctionnent pas dans tous les pays. En l'occurrence, si vous souhaitez récupérer le code postal d'un internaute français, c'est impossible avec cette méthode. Mais ne vous inquiétez pas, vous trouverez une solution grâce à Google dans la suite du tutoriel.

Si vous n'avez besoin que du pays de l'internaute, vous pouvez directement utiliser cette fonction :

Code : PHP

```
<?php
echo geoip_country_name_by_addr($gi, $_SERVER['REMOTE_ADDR']);
?>
```

Obtenez une adresse grâce à Google Maps

Il est possible d'obtenir une localisation plus précise grâce à Google Maps. Bien sûr, ce n'est pas précis à 100 % mais c'est toujours mieux que le nom d'une ville. Pour commencer, l'URL magique :

<http://maps.google.com/maps/geo?output=csv&q=latitude,longitude>

Par exemple, rendez-vous ici :

<http://maps.google.com/maps/geo?output=csv&q=48.8566667,2.3609871>

Que voit-on ?

Citation : Google Map

200,8,"13-17 Rue Pavée, 75004 Paris, France"

200 veut dire que tout va bien, 8 c'est la précision, et, voilà enfin ce qui nous intéresse... l'adresse !

Voici donc le script pour récupérer l'adresse :

Code : PHP

```
<?php
include("geoipcity.inc");
include("geoipregionvars.php");

$gi = geoip_open(realpath("GeoLiteCity.dat"), GEOIP_STANDARD);

$record = geoip_record_by_addr($gi, $_SERVER['REMOTE_ADDR']);

$la = $record->latitude;
$lo = $record->longitude;

$url = "http://maps.google.com/maps/geo?output=csv&q=" . $la . ", " . $lo;

if($csv = file_get_contents($url))
{
    if(substr($csv, 0, 3) != 200)
    {
        die("Erreur");
    }
    else
    {
        $adresse = substr($csv, 7, -1);
        echo $adresse;
    }
}
else
{
    echo "Erreur";
}

geoip_close($gi);

?>
```

C'est très simple, on récupère la latitude et longitude comme on a vu précédemment, puis on récupère les infos via l'url magique, et on supprime l'inutile en tronquant la chaîne avec la fonction `substr`.

Et nous avons donc affiché, par exemple : 13-17 Rue Pavée, 75004 Paris, France.

Obtenir des informations plus poussées

Il existe un moyen d'obtenir encore plus d'informations. Hélas, cela ne fonctionne que sur une version de php ultérieure à 5.2.0 à cause de la fonction `json_decode` qui n'existe que depuis cette version. Si vous avez une version antérieure à cette dernière, vous devriez trouver des bibliothèques qui permettent de simuler cette fonction.

Ce coup-ci, nous allons utiliser l'url suivante : <http://maps.google.com/maps/geo?output=json&q=latitude,longitude>

Si vous avez testé cette url (en lui fournissant une latitude et longitude), vous avez dû voir plein d'informations sous une structure particulière (c'est du JSON pour ceux que ça intéresse). Seul problème, php ne connaît pas cette structure qui devrait

vous faire penser aux [tableaux php](#).

D'ailleurs, nous allons justement transformer ce code JSON en tableau php, et si vous avez suivi le cours, vous avez dû deviner comment nous allons faire.

Comment ? Eh bien avec la fameuse fonction [json_decode](#) !

Nous allons d'ailleurs afficher ce tableau afin que vous voyez quelles informations vous sont accessibles (si vous connaissez le JSON, faites attention, le code fourni par google est très mal indenté et vous risquez de vous tromper en récupérant les valeurs).

Code : PHP

```
<?php
include("geoipcity.inc");
include("geoipregionvars.php");

$gi = geoip_open(realpath("GeoLiteCity.dat"), GEOIP_STANDARD);

$record = geoip_record_by_addr($gi, $_SERVER['REMOTE_ADDR']);

$la = $record->latitude;
$lo = $record->longitude;

$url = "http://maps.google.com/maps/geo?output=json&q=".$la.", ".$lo;
if($json = file_get_contents($url))
{
    $informations = json_decode($json, true);
    if($informations['Status']['code']!=200)
    {
        die("Erreur");
    }
    else
    {
        print_r($informations);
    }
}
else
{
    echo "Erreur";
}

geoip_close($gi);

?>
```

Rendez-vous sur la page et regardez le panel d'informations auquel vous avez accès ! Et bonne nouvelle, elles sont sous la forme d'un simple tableau php. Ainsi, si vous souhaitez récupérer le code postal il faut faire comme cela :

Code : PHP

```
<?php
include("geoipcity.inc");
include("geoipregionvars.php");

$gi = geoip_open(realpath("GeoLiteCity.dat"), GEOIP_STANDARD);

$record = geoip_record_by_addr($gi, $_SERVER['REMOTE_ADDR']);

$la = $record->latitude;
$lo = $record->longitude;

$url = "http://maps.google.com/maps/geo?output=json&q=".$la.", ".$lo;
if($json = file_get_contents($url))
{
    $informations = json_decode($json, true);
    if($informations['Status']['code']!=200)
    {
        die("Erreur");
    }
}
```

```
    }  
    else  
    {  
    echo  
$informations["Placemark"][0]["AddressDetails"]["Country"]["AdministrativeArea"]  
    }  
    }  
    else  
    {  
        echo "Erreur";  
    }  
  
    geoip_close($gi);  
  
    ?>
```



Pourquoi utiliser `$info["Placemark"][0]` et pas `$info["Placemark"][1]` par exemple ?

Très bonne question ! Vous avez dû remarquer que vous receviez plusieurs informations de géolocalisation. Eh bien c'est pour la simple et bonne raison que google détecte plusieurs lieux disponibles en fonction des coordonnées que vous lui avez passées.

On prend tout simplement la première localisation car c'est la plus précise ;-)

Le tutoriel est enfin fini, vous allez pouvoir réaliser votre rêve :

Géolocaliser la belle Cyntia69, membre de votre site depuis trois mois, et ainsi pouvoir lui déclarer votre flamme en bas de son immeuble. 😊

En tout cas, j'espère que vous avez aimé mon tutoriel.

Si vous avez des questions ou des problèmes, n'hésitez pas (mais ça ne devrait pas arriver).

Partager



Ce tutoriel a été corrigé par les [zCorrecteurs](#).