

Affichez le nombre de visiteurs connectés

Par Mathieu Nebra (Mateo21)



www.openclassrooms.com

*Licence Creative Commons 6 2.0
Dernière mise à jour le 23/02/2011*

Sommaire

Sommaire	2
Affichez le nombre de visiteurs connectés	3
Réalisation du compteur de connectés	3
Étape 1 : prérequis	3
Étape 2 : préparation du script	3
Étape 3 : à vous de jouer !	4
Étape 4 : correction	5
Étape 5 : améliorez ce script !	6
Partager	6



Affichez le nombre de visiteurs connectés

Par



Mathieu Nebra (Mateo21)

Mise à jour : 23/02/2011

Difficulté : Facile



En tant que webmaster, vous aimeriez certainement savoir combien de personnes sont connectées sur votre site. Après tout, pourquoi pas, c'est tout à fait le genre de trucs qu'on peut faire en PHP. 😊

La bonne nouvelle, c'est que c'est assez simple à faire. Mais il faut bien en comprendre le fonctionnement, donc soyez attentifs à ce que je vais vous dire, parce que ça ne marche peut-être pas comme vous le pensez. 😊

Réalisation du compteur de connectés

Étape 1 : prérequis

Les prérequis pour ce script sont :

- savoir lire et écrire dans une table MySQL ;
- savoir compter le nombre d'entrées dans une table ;
- savoir manier les timestamp ;
- savoir récupérer l'adresse IP du visiteur.

Voilà, c'est tout. 😊



Hey mais attends, on ne sait pas comment récupérer l'adresse IP du visiteur, nous ! Alors comment on va faire ?

Pas de panique, c'est très simple. L'adresse IP de la personne qui visite la page se récupère grâce à ce qu'on appelle une variable superglobale. Ainsi, la variable `$_SERVER['REMOTE_ADDR']` contiendra l'information recherchée : l'adresse IP du visiteur. Pour en savoir plus sur ces variables superglobales, je vous invite à lire [la partie qui y est consacrée](#) dans mon tutoriel sur PHP.

Étape 2 : préparation du script

C'est là qu'il faut bien m'écouter, parce que le fonctionnement de ce script est un peu particulier. On va créer une table MySQL appelée "connectes", avec seulement 2 champs :

- **ip** : dans ce champ de type "VARCHAR" et de longueur maximale de 15 caractères, nous stockerons *temporairement* les adresses IP des visiteurs ;
- **timestamp** : dans ce champ de type "INT", nous stockerons le timestamp, c'est-à-dire le moment exact auquel le visiteur a chargé une page sur votre site.

Voici donc ce que vous devriez avoir à la création de la table :

Champ	Type [Documentation]	Taille/Valeurs*
ip	VARCHAR	15
timestamp	INT	

Vous remarquerez que, exceptionnellement, il n'y a pas de champ "id". Les cas où on n'a pas besoin de champ ID sont très rares, il faut le savoir. Ici, le script ne nous impose pas d'utiliser un ID, donc on n'en crée pas.

Lorsqu'elle sera en activité, votre table ressemblera à ça :

←T→		ip	timestamp
		80.200.235.16	1094405892
		83.114.126.110	1094405899
		80.14.201.247	1094405906
		82.253.125.36	1094405906
		81.243.233.4	1094405912
		81.242.178.85	1094405919

Vous voyez en fait la liste des personnes connectées sur votre site (représentées par leur adresse IP), et le timestamp qu'il était au moment où chaque personne a chargé une page sur votre site.

Maintenant, il va falloir que vous compreniez comment le script fonctionne, c'est très important. Dans l'ordre, nous allons faire ceci :

- À chaque fois qu'un visiteur charge une page de votre site, on regarde dans la table "connectes" si son IP est déjà inscrite. 2 cas sont possibles :
 - L'IP n'est pas dans la table** : on crée une nouvelle entrée dans laquelle on met son adresse IP. On note en même temps l'heure (le timestamp) qu'il est.
 - L'IP est déjà dans la table** : on ne crée pas de nouvelle entrée, on met simplement à jour le timestamp associé à son IP, pour indiquer l'heure à laquelle il vient de charger la dernière page.
- Ensuite, on regarde les autres adresses IP déjà enregistrées dans la table. On supprime toutes les IP qui ont été inscrites depuis plus de 5 minutes.



Cela veut dire qu'on considère qu'un visiteur qui n'a pas chargé de nouvelle page sur le site depuis plus de 5 minutes est parti. Je sais que ça peut paraître bizarre, mais c'est comme ça qu'on fait car on n'a pas d'autres solutions.

- Enfin, dernière étape (la plus simple), on compte toutes les adresses IP qui restent dans la table. Ce nombre, c'est le nombre de visiteurs actuellement connectés sur votre site ! Vous n'avez plus qu'à l'afficher !

Par exemple, sur le Site du Zéro, mon compteur de connectés ressemble à ça :

Il y a actuellement 86 Zéros connectés !

Étape 3 : à vous de jouer !

Voilà, je vous ai dévoilé le fonctionnement du script. Après, si vous l'appliquez correctement, vous allez voir (ô surprise !) que ça fonctionne ! 😊

Un dernier détail tout de même : je vous conseille de taper ce script dans une page PHP appelée "connectes.php" qui ne contiendra que ça.

Ensuite, sur chaque page de votre site, vous incluez la page "connectes.php" comme ceci pour faire marcher le compteur : <?

```
php include('connectes.php'); ?>
```



N'oubliez aucune page, car si vous ne placez pas votre compteur sur une ou plusieurs pages, il n'indiquera pas le nombre correct de personnes connectées ! (Il en indiquera moins que ce qu'il y en a.)

Étape 4 : correction

Correction ?

Ce script est assez simple dans la mesure où je vous en ai détaillé le fonctionnement. Si vous appliquez ce que j'ai dit, cela ne peut que fonctionner !

Si par hasard vous avez bloqué sur un passage, ce n'est pas bien grave : le script est simple, et ce ne sont pas vos connaissances en PHP qui sont à remettre en cause. Vous avez simplement lu un peu rapidement les explications que je vous ai données plus haut. 😊

Voici donc le script du nombre de visiteurs connectés :

Code : PHP

```
<?php
// Connexion à MySQL
mysql_connect("localhost", "sdz", "mot_de_passe");
mysql_select_db("coursphp");

// -----
// ÉTAPE 1 : on vérifie si l'IP se trouve déjà dans la table.
// Pour faire ça, on n'a qu'à compter le nombre d'entrées dont le
// champ "ip" est l'adresse IP du visiteur.
$retour = mysql_query('SELECT COUNT(*) AS nbre_entrees FROM
connectes WHERE ip=\'' . $_SERVER['REMOTE_ADDR'] . '\'');
$donnees = mysql_fetch_array($retour);

if ($donnees['nbre_entrees'] == 0) // L'IP ne se trouve pas dans la
table, on va l'ajouter.
{
    mysql_query('INSERT INTO connectes VALUES(\'' .
$_SERVER['REMOTE_ADDR'] . '\', ' . time() . ')');
}
else // L'IP se trouve déjà dans la table, on met juste à jour le
timestamp.
{
    mysql_query('UPDATE connectes SET timestamp=' . time() . ' WHERE
ip=\'' . $_SERVER['REMOTE_ADDR'] . '\'');
}

// -----
// ÉTAPE 2 : on supprime toutes les entrées dont le timestamp est
// plus vieux que 5 minutes.

// On stocke dans une variable le timestamp qu'il était il y a 5
minutes :
$timestamp_5min = time() - (60 * 5); // 60 * 5 = nombre de secondes
écoulées en 5 minutes
mysql_query('DELETE FROM connectes WHERE timestamp < ' .
$timestamp_5min);

// -----
// ÉTAPE 3 : on compte le nombre d'IP stockées dans la table. C'est
// le nombre de visiteurs connectés.
$retour = mysql_query('SELECT COUNT(*) AS nbre_entrees FROM
connectes');
$donnees = mysql_fetch_array($retour);

// Ouf ! On n'a plus qu'à afficher le nombre de connectés !
echo '<p>Il y a actuellement ' . $donnees['nbre_entrees'] . '
```

```
visiteurs connectés sur mon site !</p>';  
?>
```

Comme vous pouvez le voir, le script respecte strictement les étapes que je vous ai indiquées au début de ce TP.

Il y a seulement 2 points où il fallait un peu réfléchir (rien de bien méchant 😊). Voici quelques informations pour vous aider à les comprendre :

- *Vérifier si l'IP du visiteur se trouve déjà dans la table* : on peut le faire de plusieurs manières ; celle qui me semble la plus simple et la plus logique est de compter le nombre de fois où cette IP apparaît dans la table. Deux réponses sont possibles :
 - **0 fois** : l'IP ne se trouve pas dans la table, c'est donc un nouveau visiteur qui vient d'arriver.
 - **1 fois** : l'IP se trouve déjà dans la table, c'est un visiteur qui était déjà sur le site et qui vient de charger une nouvelle page. On met juste à jour son timestamp pour se rappeler que ce visiteur a récemment chargé une page sur votre site.
- *Supprimer les timestamp vieux de plus de 5 minutes* : c'est assez simple quand on y pense, mais il fallait trouver le truc. En fait, on fait une requête SQL dans laquelle on demande de supprimer toutes les entrées dont le timestamp est inférieur au timestamp qu'il était il y a 5 minutes :
'WHERE timestamp < ' . \$timestamp_5min
Le plus "dur" est de retrouver le timestamp qu'il était il y a 5 minutes. Il suffit de soustraire $60 * 5$ (le nombre de secondes en 5 min) au timestamp actuel, et le tour est joué ! 😊

Étape 5 : améliorez ce script !

Là, je dois reconnaître que j'ai franchement pas beaucoup d'idées pour améliorer ce script. 😊

Voici ce que je peux vous proposer, mais je vous préviens que ça n'a rien de très excitant :

- Affichez "Il y a 1 visiteur connecté" au lieu de "Il y a 1 visiteurs connectés" pour respecter le singulier/pluriel (ah la bonne vieille grammaire française 😊).
- Vous pouvez aussi créer un système de "record". À chaque fois que vous avez le nombre de visiteurs connectés, vous le comparez avec le record enregistré dans un fichier "record.txt" par exemple. Vous saurez faire cela lorsque vous aurez lu le chapitre sur les fichiers. Ce "record.txt" ne contiendra qu'un nombre : c'est le record du nombre de connectés. Si vous le désirez, vous pouvez aussi enregistrer dans le fichier la date à laquelle le record s'est produit (utilisez un timestamp !).
- Modifiez le script pour qu'il marche sur plusieurs sites différents à la fois. Ensuite, créez un site où vous proposez ce service aux petits webmasters débutants pour seulement 10 euros, devenez riche, rachetez Microsoft, ruinez Bill Gates, et devenez enfin Maître du Monde. 😊

Vous voyez, comme quoi, même avec un petit script comme celui-ci, on peut faire des tas de choses étonnantes. 😊

J'espère que ce TP vous aura été utile, je pense que nombreux d'entre vous sont ceux qui souhaitent avoir un tel compteur sur leur site !

Ah, et si vous rachetez Microsoft, soyez cool : pensez à moi, je ne dirais pas non si on m'offrait ne serait-ce qu'1% de l'entreprise.



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