

Pré-requis

Afin de bien comprendre le tutorial, assurez-vous de maîtriser les points suivants :

- ◆ Utilisation des sessions
- ◆ Connexion et requête vers Mysql
- ◆ Commandes SQL INSERT et UPDATE
- ◆ La concaténation

Introduction

Nous avons vu dans le tutorial sur l'espace membres que les sessions peuvent servir à protéger une partie d'un site. Nous allons dans ce tutorial voir une autre utilisation des sessions, et ce pour faire des statistiques. Cela permettra entre autre de connaître quelle page a vu le visiteur, combien de temps il est resté sur chaque page, savoir sur quel site il a cliqué pour venir sur le vôtre, etc...

Ce tutorial se compose de deux parties:

- ◆ Récupérer les informations sur le visiteur et les sauvegarder dans une table (cette partie ci)
- ◆ Exploitation de ces données pour faire des statistiques (la seconde partie)

Création de la table

Les informations sur le visiteur vont être stockées dans une table. Celle-ci contiendra les champs suivants :

- ◆ id
- ◆ sessid: contient le n° de session unique généré par php
- ◆ referer: il s'agit de l'url du site sur lequel le visiteur a cliqué pour venir sur votre site
- ◆ parcours: contient les url des page demandées. Les url seront séparées par des point-virgules.
- ◆ time: contient les heures de chargement de chaque page (elles aussi séparées par des point-virgules)

La requête sql pour créer la table

Récupération des informations

Nous avons donc trois informations à récupérer : le referer, l'url demandée et l'heure précise à laquelle la page a été chargée.

Pour les deux premières informations, ce sont les variables d'environnement qui vont nous servir. Il s'agit d'informations envoyées par le navigateur lorsque vous vous rendez sur un site. PHP nous met entre autre à disposition : `$_SERVER['HTTP_REFERER']` et `$_SERVER['REQUEST_URI']`. Ces variables sont accessibles à n'importe quel moment dans votre script php. Vous l'avez compris, la première contient l'url du referer et la seconde, par deduction, l'url demandée.

Pour l'heure à laquelle la page a été demandée, nous allons la récupérer à l'aide de la fonction `time()`. Cette fonction renvoie le nombre de secondes écoulées depuis le 1 janvier 1970. Vous ne voyez peut être pas l'utilité d'un tel format, mais ce'est le plus simple à utiliser. Il nous permet à la fois de connaître le jour de la visite, ainsi qu'entre chaque page, en faisant une simple différence, de connaître en secondes combien de temps le visiteur est resté sur une page.

En fait, parmi les trois informations à stocker, le referer n'a besoin d'être enregistré qu'une seule fois car il ne varie pas si le visiteur change de page. Pour le parcours et le temps, à chaque changement de page il va falloir compléter ces deux variables et les réenregistrer dans la base.

Prenons par exemple un visiteur qui était sur google et qui a cherché le mot php. Dans la page des résultats, google a trouvé le site phpdebutant.org. Un lien vers phpdebutant figure donc sur cette page. Curieux, le visiteur clique sur ce lien en se disant "enfin je vais pouvoir apprendre php" :-). Il arrive donc sur phpdebutant.org.

Et c'est là que notre script entre en scène. Il va falloir récupérer les informations souhaitées. On aura donc besoin des variables `$_SERVER['HTTP_REFERER']` et `$_SERVER['REQUEST_URI']`, ainsi que de la fonction `time()`. Ensuite on stocke ces informations à l'aide d'une requête sql du type "INSERT INTO...". Il nous faut donc le code suivant quand le visiteur arrive sur le site :

Code pour l'arrivée d'un visiteur (stat.php)

```
// pensez à ouvrir une connexion mysql ici
// cf cours sur mysql à droite

$sql = "INSERT INTO statistique (referer,time,parcours) ";
$sql .= "VALUES
('".$_SERVER['HTTP_REFERER']."',".time().",'".$_SERVER['REQUEST_URI']."' );";

mysql_query($sql) or die('Erreur SQL !
'.$sql.'
'.mysql_error());
?>
```

Dans notre exemple, on aurait donc un premier enregistrement qui aurait cette allure-là :

id	sessid	referer	time	parcours
1	e9679b...	http://www.google.fr/search?q=php	1002644882	http://www.phpdebutant.org

Maintenant si le visiteur décide de commencer à lire les exercices, donc de changer de page, il va falloir transmettre de page en page la variable qui contient le temps, et la variable qui contient les pages demandées. Pour cela on va utiliser les sessions. Il va falloir modifier un peu notre code précédent afin de transmettre ces deux informations. Les plus attentifs d'entre vous se seront sans doute rendus compte que l'on a pas encore utilisé le champs sessid de la table. Il va désormais nous servir à identifier la session car il va contenir le n° de session unique généré par php. Pour connaître ce numero, il suffit d'utiliser la fonction `session_id()`. Voici donc le code quand un visiteur arrive.

Code pour l'arrivée d'un visiteur (stat.php)

```
// pensez à ouvrir une connexion mysql ici
// cf cours sur mysql à droite

// on crée $parcours et $time
$time = time();
$parcours = $_SERVER['REQUEST_URI'];

// on ajoute le champs sessid à la requête qui a pour valeur
// le n° de session généré par php et renvoyé par session_id()

$sql = "INSERT INTO statistique (referer,time,parcours,sessid) ";
```

```

$sql .= "VALUES ('".$_SERVER['HTTP_REFERER'].",
'$time','$parcours','".session_id()."')";

mysql_query($sql) or die('Erreur SQL !
'.$sql.'
'.mysql_error());

// on ouvre une session
session_start();

// on enregistre les variables time et parcours comme variable de
session
$_SESSION['time'] = "$time";
$_SESSION['parcours'] = "$parcours";

?>

```

Comme prévu le visiteur change de page et se rend sur la page de l'exo 1 de phpdebutant. Grâce aux sessions, on va pouvoir récupérer les variables \$time et \$parcours. Il va désormais falloir ajouter respectivement à ces deux variables ce que renvoie la fonction time(), et la variable \$_SERVER['REQUEST_URI']. Une fois ces deux variables modifiées, il faut les modifier l'enregistrement de la table correspondant à la session pour tenir compte de ce changement de variable. La seule chose que l'on connaisse de cet enregistrement est le champ sessid qui contient le n° de session. La requête sql aura donc cette allure-là : "UPDATE WHERE sessid="".session_id().""

Code pour un visiteur déjà présent sur le site (stat.php)

```

// pensez à ouvrir une connexion mysql ici
// cf cours sur mysql à droite

// on ouvre la session
session_start();

// on ajoute la nouvelle page vue, séparée par un point virgule, à la
variable de session
$_SESSION['parcours'] .= ';' . $_SERVER['REQUEST_URI'];

// on ajoute à la variable de session l'heure à laquelle la page a été
vue
$_SESSION['time'] .= ';' . time();

// on met la table à jour
$parcours=$_SESSION['parcours'];
$time=$_SESSION['time'];
$sql = "UPDATE statistique SET parcours='$parcours',time='$time'";
$sql .= "WHERE sessid='".session_id()."";

mysql_query($sql) or die('Erreur SQL !
'.$sql.'
'.mysql_error());

?>

```

Dans notre exemple, le premier enregistrement aura désormais cette allure-là :

id	sessid	referer	time	parcours
1	e9679b...	http://www.google.fr/search?q=php	1002644882;1002645306	http://www.phpdebutant.org; http://www.phpdebutant.org/?page=exo_001

On a donc désormais un morceau de code à exécuter si c'est la première fois que le visiteur arrive sur le site, et un autre morceau de code si le visiteur a déjà vu le site. Vous allez me demander comment savoir si le visiteur arrive pour la première fois sur le site ? Très simplement, si c'est le cas, alors le visiteur n'aura pas de session d'ouverte. Les variables time et parcours ne seront donc pas enregistrées comme variable de session. C'est donc la fonction `isset` qui renvoie `false` si la variable n'a pas été enregistrée qui va nous servir. Ce qui nous donne le script suivant pour joindre nos deux morceaux de code précédents :

Code pour l'arrivée d'un visiteur (stat.php)

```
// pensez à ouvrir une connexion mysql ici
// cf cours sur mysql à droite

// on ouvre la session
session_start();

if(!isset($_SESSION['parcours']))
{
// il s'agit d'un nouveau visiteur, on crée $_SESSION['parcours'] et
$_SESSION['time']
$_SESSION['time'] = time();
$_SESSION['parcours'] = $_SERVER['REQUEST_URI'];

$parcours=$_SESSION['parcours'];
$time=$_SESSION['time'];
// on crée un enregistrement dans la table
$sql = "INSERT INTO statistique (referer,time,parcours,ssid)";
$sql .= "VALUES
('".$_SERVER['HTTP_REFERER']."','$time','$parcours','".session_id()."')";
}
else
{
//Sinon c'est un visiteur déjà présent, on ajoute la nouvelle page
vue, séparée par un point virgule, à la variable de session
$_SESSION['parcours'] .= ';' . $_SERVER['REQUEST_URI'];

// on ajoute à la variable de session l'heure à laquelle la page a été
vue
$_SESSION['time'] .= ';' . time();

// on met la table à jour
$parcours=$_SESSION['parcours'];
$time=$_SESSION['time'];
$sql = "UPDATE statistique SET parcours='$parcours',time='$time'";
$sql .= "WHERE ssid='".session_id()."";
}

//Enfin on exécute la requête dans tous les cas
mysql_query($sql) or die('Erreur SQL !
' . $sql . '
' . mysql_error());
```

Mise en place du script

Nous voilà donc avec un script qui nous permet de sauvegarder quelques informations sur le visiteur. Pour le mettre en place sur votre site, c'est très simple. Il suffit juste de l'inclure en haut de toutes vos pages comme ceci:

```
include( 'stat.php' );
```

Ce code doit être impérativement placé tout en haut de chaque page. Il ne faut en effet aucun affichage de html avant l'appel à la fonction `session_start()` sous peine de provoquer une erreur.

Faites bien attention aussi à supprimer les `session_start()` qui auraient déjà été mis sur vos pages.

Pensez bien enfin que les sessions ont été introduites avec PHP4, donc vous devez disposer de PHP4 pour pouvoir exécuter ce script

Conclusion

On dispose désormais d'une table qui va contenir des données brutes. Il va donc falloir créer un script qui analyse ces données pour les présenter de manière compréhensible (tableau voire graphique...). C'est le but de la seconde partie (à venir) de ce tutorial. Entraînez vous car le code risque de se compliquer un peu plus :-)