

Les sessions

Les sessions permettent de stocker dans un petit fichier sur le serveur des données spécifiques à un utilisateur du site.

Par exemple :

- si l'utilisateur est connecté sur le site,
- le nom de l'utilisateur pour éviter de le rechercher dans la base,
- une préférence de l'utilisateur pour le site (affichage des couleurs par exemple)
- ou un message de retour que l'on pourra transmettre lors d'une prochaine requête (flashBag).

Pour retrouver ces données à chaque requête vers le serveur, le navigateur de l'utilisateur transfère un cookie spécifique appelé Cookie de Session. Ce cookie contient un numéro unique (id de session).

Avec cet id de session PHP peut retrouver le fichier de session de l'utilisateur sur le serveur. Si on lui en donne l'ordre PHP récupère ces données de session dans le fichier et les place dans la super globale `$_SESSION` qui devient accessible dans notre fichier. Cette super globale est un tableau qui contient des index et des valeurs associées que nous avons définies ultérieurement dans un autre fichier PHP par exemple.

Une session, comme son nom l'indique, est valable pour une session de navigation sur le site. Si l'utilisateur arrête de naviguer sur le site pendant un certain temps (30 minutes par défaut), sa session sera supprimée. Il devra alors par exemple se reconnecter !

On peut régler le temps que dure une session, ou rouvrir une session automatiquement en utilisant des techniques avec une base de données et des cookies que nous verrons plus tard.

Comment créer une variable de session :

login.php

```
<?php
```

```
    session_start(); //toujours en haut du fichier PHP car le buffer Apache doit-être vide, en
    effet cette instruction modifie les Header de la réponse HTTP en y plaçant un cookie de sessions
```

```
    //si utilisateur est présent dans la base et user et mot de passe sont identique :
```

```
    $_SESSION['connect'] = true ; //on met dans la session que l'utilisateur est bien connecté !
```

```
    $_SESSION['flashbag'] = 'Bravo connexion réussit' ;
```

```
    //redirection
```

```
    header('location:index.php') ;
```

```
?>
```

Comment lire une variable de session :

index.php

```
<?php
```

```
    session_start(); //toujours en haut du fichier PHP car le buffer Apache doit-être vide, en
    effet cette instruction modifie les Header de la réponse HTTP en y plaçant un cookie de sessions
```

```
    if($_SESSION['connect'] === true)
```

```
        echo 'vous êtes bien connecté !
```

```
    Else
```

```
        header('location:login.php') ;
```

?>

Voilà c'est tout simple. Regardez dans les cookies de votre navigateur à l'aide des outils de développement (F12). Vous y verrez un cookie de session. Si vous le supprimez votre session ne pourra plus être retrouvée sur le serveur.

Pour en savoir plus :

<http://www.pierre-giraud.com/php-mysql/cours-complet/php-session.php>

Bonne lecture !

Le Hashage de mot de passe

Pour comprendre le Hashage il faut savoir qu'une fois Hashée une chaîne de caractère ne peut plus être décodée. On stocke les mots de passe de cette façon dans la base de données pour éviter que ces données soient piratées ou simplement récupérées par un admin du site malveillant.

Alors comment on fait pour vérifier que l'utilisateur fournit le bon mot de passe qui est stocké en base si ce dernier est hashé. Et bien il suffit de hashé le mot de passe fourni et de comparer les deux chaînes, si le mot de passe est le bon les deux chaînes de caractères seront identiques.

Comment on hash un mot de passe, et bien je vous invite à lire ceci :

<http://php.net/manual/fr/faq.passwords.php>

Bonne lecture, tout y est !

L'upload de fichier

Mais comment on upload un fichier sur un site ??

Pioui ben c'est pas compliqué, il faut 2 choses en HTML déjà :

- précisé que notre formulaire sera composé de contenu texte (les champs de formulaire) et de contenu binaire (la ou les images). Pour cela la méthode doit-être POST pour envoyer un contenu volumineux, et on doit rajouter un enctype pour préciser que le formulaire sera composé de plusieurs parties :
<form action="action.php" **method="post" enctype="multipart/form-data"**>
- ajouter un champ de type file dans notre formulaire pour y accueillir le fichier :
<input type="file" name="imageentete">

En PHP, il faut plusieurs petites choses :

- savoir où récupérer le fichier. Pour les données texte du formulaire ça se passe dans la super globale \$_POST, ça on sait faire ! Pour la ou les fichiers ça se passe dans la super globale \$_FILES. Un petit var_dump de \$_FILES devrait vous aider !
- Le fichier uploadé par le serveur est placé dans un répertoire temporaire par le serveur Web. PHP doit maintenant déplacer ce fichier de son répertoire temporaire vers le répertoire définitif de votre application. Pour cela on récupère son emplacement temporaire dans le tableau \$_FILES puis on le déplace avec le bon nom dans notre répertoire grâce à la fonction move_uploaded_file().
- Enfin une dernière chose, PHP est configuré pour recevoir un maximum de 2 Mo de données sur vos machines. Ces paramètres sont normalement réglables dans le

fichier php.ini. Si vous envoyez un fichier de plus de 2Mo (même un peu moins) il ne sera pas accepté par PHP.

Pour en savoir plus :

<http://php.net/manual/fr/features.file-upload.php>

<http://php.net/manual/fr/function.move-uploaded-file.php>

Bonne lecture !