ECHANGE D'INFORMATIONS PHP $_SESSION

http://php.net/manual/fr/langref.php
http://www.w3schools.com/php/
https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql

SOMMAIRE

Edition : mai 2023

ECHANGE D'INFORMATIONS EN PHP

$_FILE

0-1. Les variables super globales en PHP

Ce sont des tableaux associatifs

http://php.net/manual/fr/language.variables.superglobals.php

- Les variables superglobales sont des tableaux associatifs.
- Elles sont accessibles à tout moment dans la page, quel que soit le contexte (particulièrement dans les fonctions).

S_GET

- $_GET est un tableau associatif contenant des couples clé-valeurs décrit dans une URL et provenant d'un <a href> ou d'un formulaire <form> ou directement de l'URL.

$_POST

- $_POST est un tableau associatif contenant des couples clé-valeurs provenant d'un formulaire <form>.

$_REQUEST

- $_REQUEST contient par défaut $_COOKIE, $_GET et $_POST.

$_SESSION

- Une « session » c'est une instance de fonctionnement d'un programme :
  - ➔ Elle a un début et une fin.
- Pour un site web, une session, c'est une utilisation du site par un client, c'est-à-dire un navigateur :
  - ➔ Elle a un début et une fin. La session s'arrête si le navigateur s'arrête ou après une trop longue période sans utilisation
- Chaque session contient des informations qui peuvent être accessibles par toutes les pages, pendant toute la durée de vie de la session :
  - ➔ Ces informations sont rangées dans le tableau associatif : $_SESSION

S_FILE

- $_FILES : est une variable utilisée pour les téléchargements de fichier via HTTP par la méthode POST. **Cf. chapitre suivant**.

$_COOKIE

- $_COOKIE est une variable utilisée pour conserver des informations sur a machine client via des cookies. **Cf. chapitre suivant**.

$_SERVER et $_ ENV

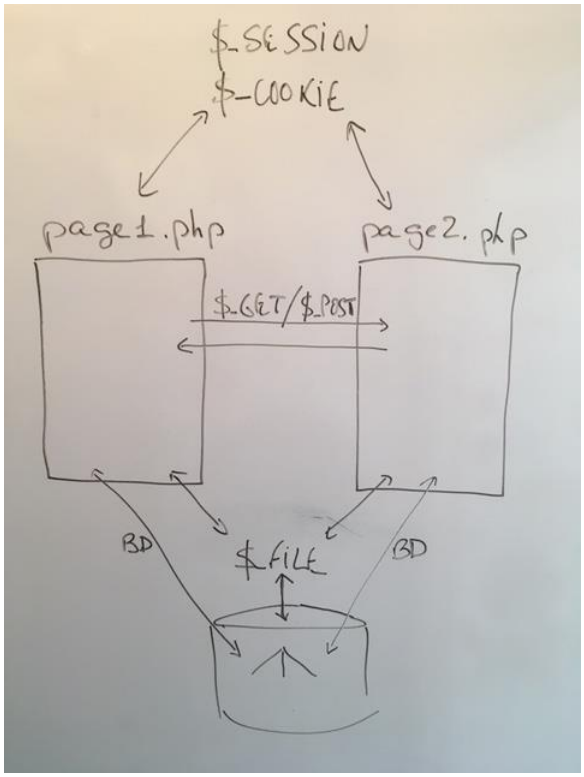
- $_SERVER : variables concernant le serveur et le programme.
  - ➔ On y trouve par exemple : $_SERVER[‘REMOTE_ADDR’] qui est l’adresse IP du client qui demande la page courante.
- $_ENV : variables concernant les variables d’environnement du serveur. Cette superglobale est rarement utilisée.

Afficher les superglobales : print_r() ou var_dump()

- Les fonctions print_r et var_dump affiche les tableaux associatifs en ligne.
- Avec les balise <pre> </pre>, on peut avoir un affichage avec une ligne par couple key-value.

0-2. Échanges entre les pages

- Quand une page appelle une autre page, elle peut fournir de l'information dans les tableaux $_GET ou $_POST.
- Les pages peuvent lire et écrire dans le tableau $_SESSION qu'elle vont donc partager.
- Les pages peuvent écrire des cookies et lire ces cookies dans le tableau $_COOKIES.
- Les pages peuvent lire et écrire dans des fichiers qu'elles peuvent partager.
- Les pages peuvent lire et écrire dans une base de données qu'elles peuvent partager.
- Les pages peuvent lire des informations concernant le serveur et le client dans les tableaux $_ENV et $_SERVER



0-3. Appel d'une page

On peut appeler une page de 3 façons :

< a href= « url » >

Dans ce cas, l'information circule forcément par un **$_GET** et directement sur l'url.
https://www.w3schools.com/tags/tag_a.asp

< form action= « url » method= « » >

Dans ce cas, l'information circule par **$_GET ou $_POST** selon la valeur de l'attribut method dans la balise form (GET ou POST).
http://www.w3schools.com/tags/att_form_action.asp

header('Location: mapage.php');

header() permet de demander une URL, un peu comme un href.
Dans ce cas l'information circule forcément par un **$_GET** et directement sur l'url.
On peut donc passer des paramètres, comme pour toute URL.
http://php.net/manual/fr/function.header.php

header() est utile en MVC : il permet d'avoir une page index à la racine qui appelle un premier contrôleur :

```
<?php
  header('Location: ./ctrl/mapage.php');
?>
```

1. $_SESSION

Présentation

- Une « session » c'est une instance de fonctionnement d'un programme :
  - ➔ Elle a un début et une fin.
- Pour un site web, une session, c'est une utilisation du site par un client, c'est-à-dire un navigateur :
  - ➔ Elle a un début et une fin. La session s'arrête si le navigateur s'arrête ou après une trop longue période sans utilisation
- Chaque session contient des informations qui peuvent être accessibles par toutes les pages, pendant toute la durée de vie de la session :
  - ➔ Ces informations sont rangées dans le tableau associatif : $_SESSION

Précision technique

- Comment fait le serveur pour savoir qu'il s'agit de tel ou tel utilisateur ?
- C'est le cookie de session qui permet de garantir cela. Les cookies sont gérés par le tableau associatif $_COOKIE. Les données sont enregistrées côté client.
- Ce cookie n'a de sens que pour le serveur web et ne donne aucune information sur l'utilisateur. On y reviendra dans le chapitre sur les cookies.

➔ https://openclassrooms.com/courses/concevez-votre-site-web-avec-php-et-mysql/session-cookies

Utilisation du tableau associatif $_SESSION

- Le démarrage d'une session doit se faire avant tout code HTML.

session_start()

https://www.php.net/manual/fr/function.session-start.php
https://www.w3schools.com/php/php_sessions.asp

- Pour pouvoir utiliser la variable $_SESSION il faut appeler la fonction session_start().
- Toutes les pages qui veulent utiliser $_SESSION doivent appeler session_start().

```
< ?php
session_start();
?>
<DOCTYPE html>
```

- L'appel de la fonction doit se faire au tout début de la page, avant le <DOCTYPE html>

$_SESSION

- $_SESSION s'utilise comme un tableau associatif.
- Pour enregistrer une donnée dans $_SESSION on écrira :

```
$_SESSION['administrateur']='Bertrand';
```

- Une fois la clé « administrateur » créée, on peut tester si elle existe :

```
if (isset($_SESSION['administrateur']){
    echo 'Bonjour'. $_SESSION['administrateur'];
    echo 'Vous êtes connecté et pouvez accéder au back-office';
}
```

Terminer une session

Plusieurs techniques peuvent terminer une session.

1 : unset($_SESSION['administrateur'])

- Un « unset » ne finit pas vraiment la session, mais supprime une clé dans $_SESSION. Un test avec « isset » donnera donc une réponse fausse.

2 : $_SESSION = array()

- Cette instruction supprime toutes les variables de session.

3 : patienter ! timeout

- On bout d'un certain temps sans activité, le « timeout », les variables de session sont supprimées.

4 : se déconnecter du serveur = fermer le client, donc le navigateur.

- Si on quitte le navigateur, les variables de session seront supprimées.
- A noter que fermer la page sans fermer le navigateur ne supprime pas les variables de session.
- A noter aussi que certains sites conservent les variables de session (la connexion typiquement) même quand on a fermé le navigateur. Dans ce cas, ils utilisent les cookies. On y reviendra.

5 : session_destroy();

http://php.net/manual/fr/function.session-destroy.php
https://www.w3schools.com/php/php_sessions.asp

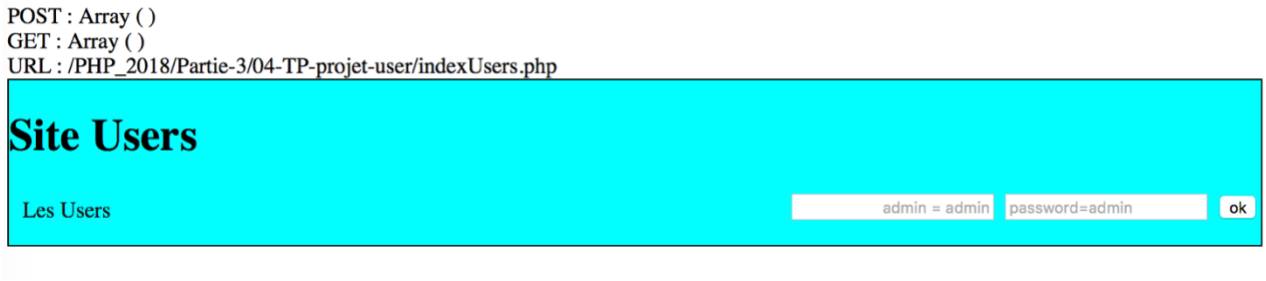
- La fonction bloque l'effet du session_start() sans supprimer les données.
- Un nouveau session_start() permet d'y réaccéder.

2. Exemples de mises en pratique

2.1 : login et mot de passe

Objectif :

On ajoute le formulaire de connexion dans le <header>, du côté du <nav>



Validation

Si le login / mot de passe fonctionne, on enregistre une variable avec le login dans $_SESSION : elle permettra d'afficher des informations spécifiques à l'utilisateur connecté.

2.2 : erreur de login

Objectif :

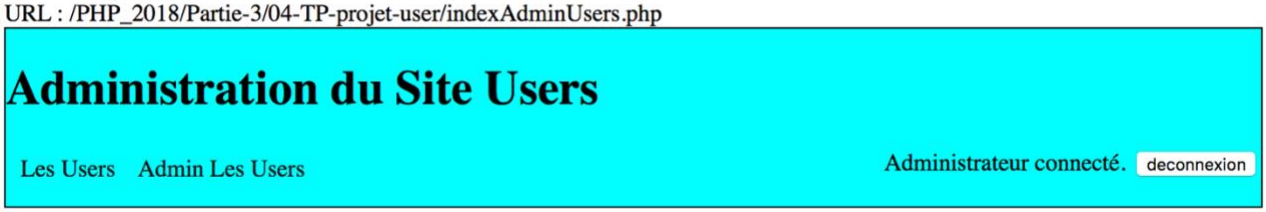
En cas d'erreur de login, on affiche un message d'erreur :

2.3 : administrateur connecté

Objectif :

Quand un administrateur est connecté, il accède à un menu spécial : <header> et <nav> spécial.
On sait que l'administrateur est connecté parce qu'on a enregistré son login dans $_SESSION.

POST : Array ([admin] => admin [password] => admin)
GET : Array ( )
URL : /PHP_2018/Partie-3/04-TP-projet-user/indexAdminUsers.php



Débug :

Pour voir l'administrateur connecté, on ajoute un

```
echo'SESSION : '; print_r($_SESSION);echo'<br/>';  pour le debug.
```

2.4 : déconnexion

Objectif :

L'objectif est de se déconnecter.

Si on a choisi la déconnexion, il faut faire un unset $_SESSION['admin'].
Quand de déconnecte, on revient sur la page d'utilisateur non connecté.