

## TP n° 6 bis

### Persistence des données chez le client

#### 1 Persistence par les Sessions

Les cookies nous ont donné une première façon de stocker les données entre deux visites d'un même utilisateur, mais ils ont de nombreux désavantages : ils ont une durée de vie limitée, on doit faire confiance au navigateur, et ils forcent à échanger une grande quantité de données à chaque requête (imaginez si vous aviez stocké une photo dans le profil de l'utilisateur !).

En plus, les cookies ne garantissent aucune sécurité : si vous voulez stocker une variable qui indique que l'utilisateur s'est loggué avec son login et mot de passe, cette variable ne peut pas être stockée par le client car il pourrait la falsifier.

PHP offre une technique avancée, appelée "Sessions PHP" qui offre la persistance des données en faisant abstraction des cookies et des techniques de pseudo-persistance.

##### Exercice 1 — *Persistence par les sessions*

1. Créez une page `countSession.php` qui compte le nombre de visites grâce aux sessions. Cette fois-ci vous pouvez tranquillement promettre un milliard d'euros...
2. Allez voir les cookies sauvegardés par votre navigateur, vous remarquerez que `pams.script` a enregistré un nouveau cookie nommé `PHPSESSID`. Effacez ce cookie et visitez à nouveau la page, que remarquez-vous ?

Ce n'est pas sorcier : les sessions PHP se servent des cookies pour permettre la persistance des données. Cependant le cookie `PHPSESSID` ne contient aucune des données que vous avez stockées dans le tableau `$_SESSION` : ce cookie ne contient rien d'autre que l'id de session. L'id de session est une chaîne de caractères aléatoire et unique créée par le serveur pour identifier le client ; lors du premier appel de `session_start()` les actions suivantes sont exécutées par PHP :

1. un nouvel id est généré,
2. un fichier temporaire correspondant à l'id est créé **sur le serveur**,
3. un cookie `PHPSESSID` contenant l'id de session est envoyé **au client**.

##### Exercice 2 — *Profil utilisateur avec les sessions*

Finissez l'Exercice 4 du TP précédent si cela n'est pas déjà fait.

Reprenez les fichiers de cet exercice et adaptez les de façon à utiliser des sessions à la place des cookies.

#### 2 Organiser le code

On va ici mieux organiser les contenus de notre site. Commencez par faire l'Exercice 5 du TP précédent si cela n'est pas déjà fait.

##### Exercice 3 — *Pseudoframes*

Si vous essayez de remplacer l'appel à `header` par un `include`, le site ne va plus marcher car vous aurez deux appels consécutifs à la fonction `session_start()` (essayez pour voir). L'architecture des pseudoframes est la solution.

1. Créez une page `accueil.php` et transférez-y le contenu de `index.php`.
2. Effacez `session_start()` de toutes les pages à l'exception de `index.php`. Cette dernière doit prendre un paramètre GET `action` qui indique quelle action entreprendre, les valeurs possibles sont :

- inscription : le formulaire d'inscription sera affiché,
  - modifier : le formulaire d'inscription pré-rempli sera affiché,
  - sauvegarder : les données sur l'utilisateur envoyées par la méthode POST seront vérifiées et sauvegardées dans le tableau `$_SESSION`,
- Si aucune action est spécifiée, la page d'accueil sera affichée. La page `index.php` s'occupera de traiter les données en entrée et de charger la bonne page via la fonction `include`.

Ceci n'est pas encore totalement satisfaisant : si un utilisateur ne passe pas par la page `index.php` et demande directement la page `sauvegarde.php`, des erreurs se produiront (essayez). La solution consiste à faire en sorte qu'aucune page, à l'exception de `index.php`, ne contienne d'instructions d'affichage.

#### Exercice 4 — Des fonctions à la place des affichages

1. Modifiez le fichier `accueil.php` pour qu'il contienne deux fonctions `accueil` et `accueil_logged`. Les deux fonctions ne font rien d'autre que retourner une chaîne de caractères : la première retourne le contenu à afficher si l'utilisateur n'a pas renseigné ses données ; la deuxième prend en argument les données de l'utilisateur et retourne le contenu à afficher. Le fichier ne doit contenir rien d'autre que les deux fonctions : pas d'affichages, pas de code HTML en dehors de `<?php ... ?>`.
2. Modifiez de façon semblable `sauvegarde.php` et `inscription.php`. Maintenant vous avez trois fichiers qui ne contiennent que des fonctions ; ce type de fichier est souvent dit une bibliothèque PHP.
3. Modifiez `index.php` pour que les trois bibliothèques soient incluses une seule fois au début de la page. À la place des `include` que vous aviez placés auparavant, vous mettrez des appels aux fonctions de vos bibliothèques.
4. Dirigez votre navigateur sur `sauvegarde.php`. Que se passe-t-il, maintenant ?

Si vous avez correctement fait l'exercice, le résultat sera beaucoup plus concis qu'avant. Félicitations ! Vous avez maintenant une architecture générale très robuste que vous pourrez utiliser partout, en particulier pour votre projet.

Attention à n'inclure qu'une seule fois chaque fichier ! Si maintenant vous incluez deux fois le fichier `accueil.php`, vous aurez une erreur car les fonctions `accueil` et `accueil_logged` seront définies deux fois. Pour aider les développeurs de sites complexes, PHP met à disposition une fonction `include_once` qui inclut un fichier seulement s'il ne l'a pas déjà été. De manière générale, `include_once` est préférable pour les bibliothèques (ce qui est ici votre cas), alors que `include` est plus indiqué pour des fichiers qui font de l'affichage.

## 3 Partie optionnelle

Nous allons ajouter du contenu à notre site en utilisant les principes couverts jusqu'ici.

**Exercice 5 — Rajouter du contenu au site** Maintenant que vous avez un moyen robuste de créer une session utilisateur, vous êtes prêts à rajouter du contenu à votre site. N'oubliez pas de continuer à utiliser l'architecture que nous venons de voir, et de toujours vous poser la question de où stocker les variables (GET, POST, COOKIE ou SESSION).

1. Créez un fichier `preferences.php` qui contiendra un formulaire permettant à un utilisateur enregistré de sélectionner les couleurs du fond et du texte qu'il souhaite utiliser. Toutes les autres pages du site seront ensuite affichées en utilisant ces paramètres. Rajouter des options à votre gré.
2. Créez un jeu "Plus ou Moins". Le serveur choisi aléatoirement un nombre entre 0 et 100 à faire deviner à l'utilisateur. À chaque étape, l'utilisateur soumet un nombre et le serveur répond si le nombre à deviner est plus grand ou plus petit. Le joueur perd s'il met plus de sept coups à trouver.
3. Rajoutez progressivement de la difficulté : à chaque victoire successive, augmentez la taille de l'intervalle parmi lequel le nombre est choisi (et adaptez le nombre de coups autorisés en conséquence).