

# PHP : Cookies et Sessions

## L3 Informatique - UE Développement Web

David Lesaint  
david.lesaint@univ-angers.fr



Janvier 2020

## Rappel

HTTP est un protocole de transmission sans état (stateless).

- Aucun lien n'est établi par définition entre deux requêtes émanant d'un même poste client : eg. navigation par hyperlien, saisie de formulaire.

## Motivations

Conserver des informations pour améliorer le service rendu par un site à un utilisateur.

- Personnaliser le contenu des pages grâce aux cookies.
- Partager des informations entre différentes pages grâce aux sessions.

## Cookies

Fichiers écrits par un script sur le poste client.

- Chaque cookie ne peut dépasser 4Ko.
- Chaque site ne peut écrire plus de 20 cookies sur le poste client.

L'un des mécanismes possibles pour la mise en place de sessions.

La création et le stockage de cookies peuvent être désactivés par l'utilisateur sur son navigateur.

# Ecriture des cookies

```
boolean setcookie(string nom [, string valeur,  
int datefin, string chemin, string domaine,  
bool securite)
```

nom	Le nom du cookie (même règles que pour les variables).
valeur	La valeur associée au cookie.
datefin	Date d'expiration sous la forme d'un timestamp Unix. Exemple : <code>time()+86400</code> pour 24h.
chemin	Chemin du dossier des scripts autorisés à accéder au cookie. Exemple : <code>/</code> pour l'intégralité des scripts du domaine.
domaine	Nom entier du domaine où l'accès au cookie est autorisé. Exemple : <code>www.acme.com</code>
securite	TRUE si transmission https requise, FALSE sinon (http par défaut).

Aucun contenu HTML (en dur, via `echo ...`) ne doit être envoyé avant appel à `setcookie` comme pour header.

# Ecriture des cookies

## cookies-creation.php

```
1 // Cookie valable uniquement pour la session
2 setcookie("cookie", "session");
3 // Cookie valable 15 secondes
4 setcookie("cookie15s", "15s", time()+15);
5 // Cookie valable 1 heure
6 setcookie("cookie1h", "1h", time()+3600);
7 // Cookie valable 1 journée
8 setcookie("cookie1j", "1j", time()+86400);
9 // Cookie "sensible"
10 setcookie("CB", "5612 1234 5678 1234", time()+86400, "/client/paiement/", "www.acme.com", TRUE);
```

## Tableau de cookies

```
setcookie("personne[prenom]", "John");
```

- Crée un tableau de cookies (sans guillemets pour les clés).

## exemple12-2.php

```
1 $tabcook = array("prenom"=>"Paul", "nom"=>"Char", "ville"=>"Marseille");
2 foreach($tabcook as $cle=>$valeur) {
3     setcookie("client2[$cle]", $valeur, time()+7200);
4 }
```

# Effacement et suppression de cookies

## Autres usages de `setcookie`

```
setcookie("nom");
```

- Efface la valeur précédente du cookie.

```
setcookie("nom", "v", time()-3600);
```

- Supprime le cookie (réutiliser la valeur courante à l'appel).

## cookies-effacement-destruction.php

```
1 setcookie("cookie15s", "30s", time()+30); //Réaffectation
2 setcookie("cookie1j"); //Effacement
3 setcookie("cookie1h", "1h", time()-1); //Destruction
4 print_r($_COOKIE);
```

# Lecture des cookies

## Avec superglobale \$\_COOKIE

En utilisant le nom du cookie comme clé.

- Toutes les pages du script ont accès immédiat aux cookies.
- La page créant le cookie n'y a accès qu'à son rechargement.

### cookies-lecture.php

```
1 $ck1 = $_COOKIE["cookie"] ?? "no-cookie";
2 $ck2 = $_COOKIE["cookie15s"] ?? "no-cookie15s";
3 $ck3 = $_COOKIE["cookie1h"] ?? "no-cookie1h";
4 $ck4 = $_COOKIE["cookie1j"] ?? "no-cookie1j";
5 echo $ck1 . "<br/>";
6 echo $ck2 . "<br/>";
7 echo $ck3 . "<br/>";
8 echo $ck4 . "<br/>";
9 print_r($_COOKIE);
```

HTTP ne permet pas de conserver des informations provenant d'une page pour les utiliser dans une autre.

## Support des sessions (PHP 4+)

- Permet de conserver et partager des informations entre pages pour un même visiteur.
- Aucun autre visiteur n'a accès à ces données.



# Mécanisme des sessions en PHP

- Création ou réouverture d'une session par appel à `session_start` en début de script.
- Chaque session est nommée et a un identifiant unique (géré automatiquement par PHP ou controllable).
- Transmission du nom et identifiant de session d'une page à l'autre via cookie (automatique) ou par ajout (manuel) aux hyperliens des pages HTML renvoyées.
- A chaque session correspond la superglobale `$_SESSION` qui permet de partager des données entre scripts appelant `session_start`. La sauvegarde s'effectue automatiquement sur un dossier du serveur.
- PHP sérialise les données de session avant arrêt du script puis les désserialise à l'appel de `session_start` sur la base du nom et de l'identifiant de session transmis.

# Sessions avec cookies

## Uniquement si

- Ecriture de cookies autorisée sur poste client (`setcookie(...)` retourne `TRUE`).
- `session.use_cookies=on` dans `php.ini`

## `session_start()`

A appeler en début de script(s) pour activer les sessions.

- PHP se charge de l'écriture et de la lecture du cookie.

## Partager une variable de session

```
1 session_start();
2 echo "Nom de session : ", session_name(), "<br/>";
3 echo "Id de session : ", session_id(), "<br/>";
4 $_SESSION["variable"]='valeur';

1 session_start();
2 echo "Nom de session : ", session_name(), "<br/>";
3 echo "Id de session : ", session_id(), "<br/>";
4 echo '$_SESSION["variable"]=', $_SESSION["variable"];
```

# Sessions avec cookies

```
unset ($SESSION["ma-var"]);
```

- Pour détruire une variable de session.

```
session_unset();
```

- Pour détruire les variables d'une session.

```
session_destroy();
```

- Pour détruire la session.

# Sessions sans cookies

Si l'écriture de cookies est impossible.

```
session_start()
```

A appeler en début de script(s) pour activer les sessions.

- Nom de session par défaut : PHPSESSID.
- Identifiant de session (26 caractères aléatoires) généré automatiquement, eg. 56epahncd9lh3mjap2eqorgf98.
- Nom et id de session stockés dans constante `SID`.

Recharge automatiquement "la" session si `SID` est ajoutée aux hyperliens.

## Usage de l'identifiant de session (session-sans-cookie.php)

```
1 session_start();
2 empty($_SESSION['count']) ? $_SESSION['count']=1 : $_SESSION['count']++;
3 <p>Vous avez vu cette page<?php echo $_SESSION['count']; ?>fois.
4 Pour continuer,
5 <?php $query_string = session_name().'='.session_id(); // or echo SID; ?>
6 <a href="session-sans-cookie.php?<?php echo htmlspecialchars($query_string);
?>">
7 cliquez ici</a>.
8 </p>
```