

Université Sidi Mohamed Ben Abdllah  
Faculté des sciences Dhar El Mahraz  
Licence professionnelle SIGL

# **Web dynamique: Cookies et sessions en PHP**

Réalisé par: Pr. Mahraz Med Adnane  
Année universitaire:2016/2017



# Les cookies

Les cookies servent à enregistrer et retenir des informations concernant le système de l'internaute. Le plus généralement, ils sont utilisés pour :

- stocker des informations saisies (ou des choix faits) par l'internaute (ex. : des préférences)
- effectuer du traitement statistique

Un cookie est donc un fichier texte placé sur le système de l'internaute, il est tout à fait possible à un internaute averti de modifier les cookies sur son système.

# Cookies en PHP

- Placer un cookie sur le client :

```
bool setcookie(    string name  
                  [, string value  
                  [, int    expires  
                  [, string path  
                  [, string domain  
                  [, bool    secure  
                  [, bool    httponly  
                  )
```

Domaine à partir duquel peuvent être envoyés des cookies

Cookie uniquement par HTTP ?

true → uniquement HTTP

false → défaut, HTTP, JavaScript, ...

# Cookies en PHP

- Pour lire (et utiliser) un cookie, on utilise la superglobale \$\_COOKIE :

```
if(!isset($_COOKIE["compteur"])){  
    setcookie("compteur","1");  
} else{  
    $i=$_COOKIE["compteur"];  
    $i++;  
    setcookie("compteur",$i);  
}
```

# Cookies en PHP

Pour supprimer un cookie, il n'existe là encore aucune fonction spécifique... Il suffit de recréer le cookie sans valeur :

```
setcookie("nom du cookie");
```

Ou bien, on peut lui attribuer une date d'expiration antérieure à la date courante.  
`setcookie("nom du cookie","",1);` // la valeur 1 correspond au 1er janvier 1970 à minuit et 1 seconde)

# Cookies en PHP

Pour stocker un tableau dans un cookie (un cookie ne peut stocker qu'une seule valeur), on utilise une fonction de linéarisation pour transformer le tableau en chaîne, et une fonction de "dé-linéarisation" pour retransformer la chaîne en tableau lors de la lecture du cookie.

- Fonction de linéarisation :
  - `$chaine = serialize ($tableau) ;`
- Fonction de dé-linéarisation :
  - `$tableau = unserialize ($chaine) ;`

# Cookies en PHP

```
if(!isset($_COOKIE["unTableau"])){  
    $tab = array("pomme","poire","banane","fraise");  
    $valeur = serialize($tab);  
    setcookie("unTableau",$valeur);  
} else{  
    print_r(unserialize($_COOKIE["unTableau"]));  
}
```

# Cookies en PHP

Une deuxième manière pour linéariser un tableau:

```
$login="scott";  
$pass="7788";  
$arr=compact("login","pass");  
$var=implode("&",$arr);  
setcookie("exemple",$var);  
if (isset($_COOKIE['exemple'])) {  
    $tab=explode("&,$_COOKIE['exemple']");  
    print_r($tab);  
}
```



# Cookies en PHP

## Exemple de choix des langues

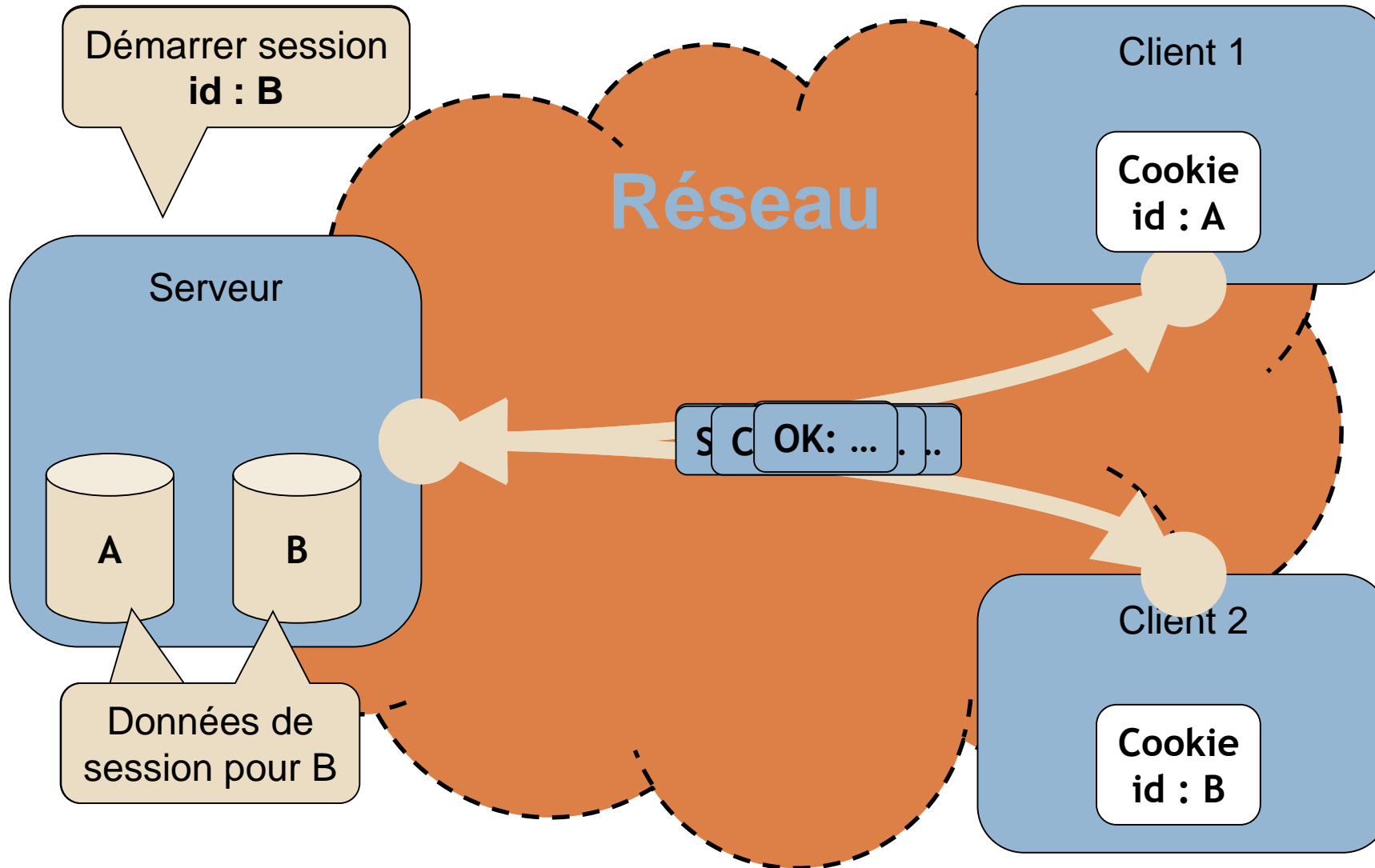
```
<?php
if(isset($_GET['pays'])){
    $pays=$_GET['pays'];
} elseif(isset($_COOKIE['pays'])) {
    $pays=$_COOKIE['pays'];
} else{
    $pays='fr';
}
```

```
$temps = 365*24*3600; // 1an
setCookie("pays",$pays,time()+$temps);
switch($pays) {
    case 'fr': print '<p>Bonjour</p>'; break;
    case 'ar': print '<p>السلام عليكم </p>';
break;
}
?>
<a href="?pays=fr"> France </a><br/>
<a href="?pays=ar"> Arabe </a><br/>
```

# Sessions

- Une session est un fichier stocké sur le serveur (dans le dossier /tmp/ de l'arborescence du serveur).
- Pour chaque internaute, la session renvoie un identifiant unique (c'est cet identifiant qui identifie également le fichier du serveur).
- Ce fichier-session peut stocker tous types de données.
- Chaque donnée est automatiquement "linéarisée" au stockage, et automatiquement "délinéarisée" à la lecture.
- Quand une session est créée, cela crée automatiquement un cookie sur le pc de l'utilisateur, nous appelons cela le cookie de session (PHPSESSID)

# Sessions



# Sessions

- Créer ou restaurer une session
  - `bool session_start ( void )`
- Manipuler les données de session
  - Tableau associatif superglobal `$_SESSION`
- Lire ou définir l'identifiant de session
  - `string session_id ( [string id] )`
- Détruire toutes les variables d'une session
  - `void session_unset ( void )`
- Détruire une session
  - `bool session_destroy ( void )`

# Sessions

Une fois la session initialisée, toutes les variables qu'elle contient sont disponibles depuis la superglobale `$_SESSION` :

```
<?php
session_start();
if(!isset($_SESSION["compteur"])){
    $_SESSION["compteur"]=1;
}else{
    $_SESSION["compteur"]++;
}
?>
```

# Sessions

- Les objets peuvent être stockés dans les données de session
  - Linéarisation de l'objet → réduit à une chaîne
  - Restauration possible (PHP >= 4)

```
session_start() ;  
$p = new Point(12, 15) ;  
$_SESSION['monpoint'] = $p ;
```

```
session_start() ;  
$p = $_SESSION['monpoint'] ;
```

# Sessions

Méthodes magiques `__sleep` et `__wakeup`

- `__sleep`

- Appelée avant la linéarisation
- Doit retourner un tableau contenant le nom des attributs à linéariser
- Choix des attributs à sauvegarder
- Termine correctement l'objet (déconnexion BD, ressource ouverte, requête en cours, ...)

- `__wakeup`

- Appelée après la délinéarisation
- Restaurer des attributs non sauvegardés (BD, ...)

# Sessions

```
class Point {  
    private $coord ;  
    private $info ;  
    public function __construct($_x,  
$_y) {  
        $this->coord['x'] = $_x ;  
        $this->coord['y'] = $_y ;  
        $this->info = 'Nouveau' ;  
    }  
    public function __sleep() {  
        return array('coord') ;  
    }  
}
```

```
public function __wakeup() {  
    $this->info = 'Restauré' ;  
}  
  
public function affiche() {  
    var_dump($this) ;  
}  
}
```



# Sessions

```
<?php
require_once('point.inc.php') ;
session_start() ;
$p = new Point(1, 12) ;
$_SESSION['point'] = $p ;
?>

<?php
require_once('point.inc.php') ;
session_start() ;
$p = $_SESSION['point'] ;
$p->affiche() ;
?>
```

Linéarisation :

object(*Point*)[ 1]  
private 'coord' => array (size=2)  
    'y' => int 1 'x' => int 12

```
public function __wakeup() {
    $this->info = 'Restauré' ;
}
```

'Restauré' (length=9)

Délinéarisation :  
appel de **\_\_wakeup()**