



PHP WEB +

Redirections

PHP 2

- Ne pas renvoyer au navigateur la page demandée.

```

graph LR
    Laptop[Laptop] --> login1[login.php]
    login1 --> accueil[accueil.php]
    accueil -- OK -->|oui| Laptop
    accueil -- OK -->|non| login2[login.php]
    login2 --> Laptop
  
```

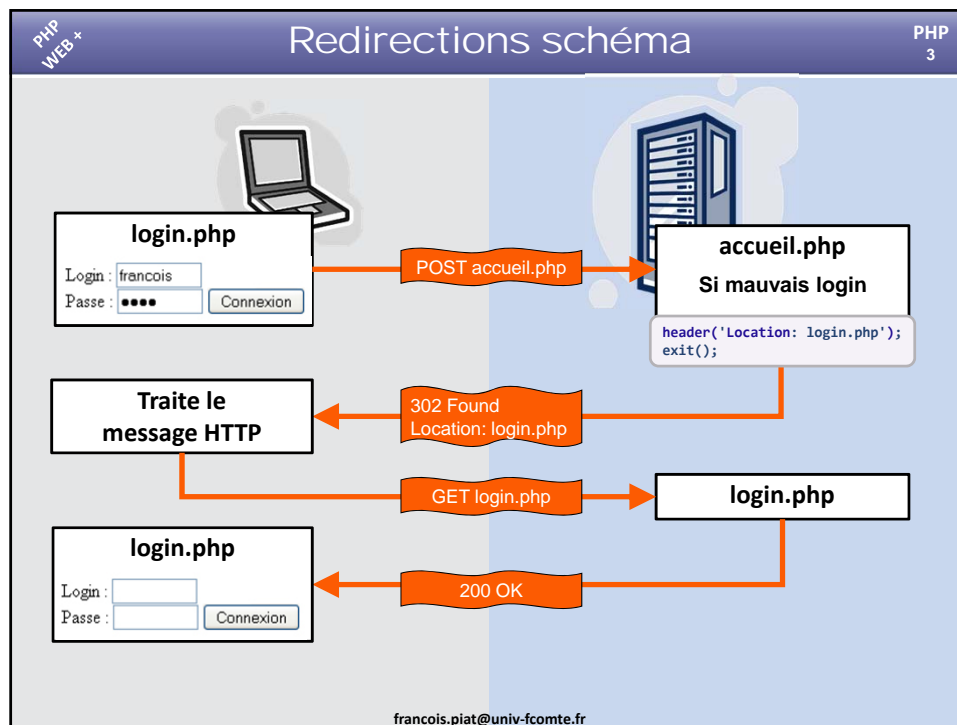
- Forcer un en-tête HTTP **Location** avec la fonction **header()**.

```

if (verifLogin() === false) {
    header('Location: login.php');
    exit();
}
  
```

Important !

francois.piat@univ-fcomte.fr



PHP WEB+ Caching PHP 4

### Gestion de la mise en cache des pages par le navigateur.

```

$CacheDate = gmdate('D, d M Y H:i:s'). ' GMT';

header("Expires: $CacheDate");
header("Last-Modified: $CacheDate");
header('Cache-Control: no-store, no-cache, must-revalidate,
pre-check=0, post-check=0, max-age=0');
header('Pragma: no-cache');
  
```

francois.piat@univ-fcomte.fr

PHP  
WEB +

Bufferisation ?

PHP  
5

- Les envois de données au navigateurs sont stockés dans un tampon (buffer).
- Le tampon est vidé :
  - en une seule fois à la fin du script ou
  - quand le développeur le veut.
- Avantages :
  - éviter les erreurs d'envoi d'en-têtes après du contenu,
  - gain de temps dans le transport par le protocole HTTP,
  - gestion des erreurs éventuels : on peut afficher une page "propre",
  - mettre en place un système de cache des pages demandées.

francois.piat@univ-fcomte.fr

PHP  
WEB +

Les fonctions de bufferisation

PHP  
6

<code>ob_start()</code>	Démarre la mise en tampon des sorties.
<code>ob_flush()</code>	Envoie le contenu du tampon au navigateur.
<code>ob_end_flush()</code>	Envoie le contenu du tampon au navigateur et arrête la mise en tampon.
<code>ob_end_clean()</code>	Supprime le contenu du tampon, sans l'envoyer au navigateur, et arrête la mise en tampon.
<code>ob_get_contents()</code>	Retourne le contenu du tampon sous la forme d'une chaîne de caractères.

francois.piat@univ-fcomte.fr

PHP  
WEB+

Exemple gestion d'erreurs

PHP  
7

```

<?php
ob_start();

set_error_handler('afficheErreurs');

. . . code . . .

function afficheErreurs($Num, $Msg, $Fichier, $Ligne) {
    ob_end_clean();
    echo '<html><head><title>Erreur</title></head>',
        '<body>',
        '<p>Erreur ', $Num,
        'ligne ', $Ligne,
        'dans le script ', $Fichier,
        '<br>Texte : ', $Msg,
        '</p></body></html>';
    exit();
}
?>

```

francois.piat@univ-fcomte.fr

PHP  
WEB+

Exemple d'envoi fractionné

PHP  
8

```

<?php
ob_start();

// Génération du début du code HTML
// Liens sur les feuilles de styles
// Liens sur les fichiers JavaScript

echo '<html><head> . . .',
    '<link href="lesStyles.css" rel="stylesheet">',
    ...
    '<script src="javascript.js">',
    ...
    '<body>';

ob_flush();

// Suite des traitements ...

```

francois.piat@univ-fcomte.fr

PHP  
WEB+

Exemple compression

PHP  
9

- Les navigateurs acceptent du contenu compressé avec l'algorithme gzip :  
Accept-Encoding: gzip, deflate
- `ob_start()` accepte en paramètre le nom d'une fonction qui sera appelée quand la mise en tampon est finie.
- `ob_gzhandler()` : compression du tampon si le client supporte gzip.

```
<?php
ob_start('ob_gzhandler');
. . .
```

francois.piat@univ-fcomte.fr

PHP  
WEB+

Exemple mise en cache

PHP  
10

```
ob_start();
$Fichier = 'la_demande.html';
//-----
if (file_exists($Fichier)) {
    $temps = time() - filemtime($Fichier);
    if ($temps <= 3600) {
        readfile($Fichier);
        exit();
    }
}

//-----
echo '<html><head><title> ...';
...
... traitement long ou qui demande beaucoup de ressources
...
echo '</body></html>';

//-----
$Cache = fopen($Fichier, 'w');
fwrite($Cache , ob_get_contents());
fclose($Cache);
```

francois.piat@univ-fcomte.fr

PHP  
WEB +

Cookies

PHP  
11

- Les cookies sont des chaînes de caractères
  - identifiées par un nom,
  - passées **automatiquement** par le serveur et par le navigateur **dans chaque séquence** (request / response) du dialogue client/serveur.
- **setcookie()** permet de créer des cookies qui seront envoyés au navigateur.
  - Attention au domaine sur lequel s'applique le cookie (voir tutoriel pour détails).
- **\$\_COOKIE** est le tableau associatif super-global qui permet de **lire** les valeurs des cookies.

francois.piat@univ-fcomte.fr



Sessions

## 2 problèmes

PHP  
13

- **HTTP est un protocole sans état.**
  - Les requêtes sont indépendantes les unes des autres.
  - Le serveur traite une demande, envoie une réponse et coupe la communication.
  - Le serveur ne sait pas si des demandes consécutives viennent du même client.
- **Les sites / applications Web ont besoin :**
  - de savoir si des demandes consécutives viennent du même client,
  - d'associer des informations à ces demandes de même origine,
  - de stocker et partager ces informations entre les différentes pages.

francois.piat@univ-fcomte.fr

Sessions

## Numéro de session

PHP  
14

- **Un numéro de session (session ID ou SID) est automatiquement généré pour identifier un utilisateur.**
- **Le SID est passé dans tous les échanges client/serveur.**
  - Dans les URLs de liens / de pages.
  - Dans un cookie.

```
GET /index.php HTTP/1.1
Host: www.lesite.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9.0.3) Gecko/2008092417 Firefox/6.0.3
Accept: text/html,application/xhtml+xml,application/xml;
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Connection: keep-alive
Cookie: PHPSESSID=541ed88dd36b0aa35d4e3ed7dc19acaa
```

francois.piat@univ-fcomte.fr

Sessions
Stockage
PHP 15

- **Un fichier est associé au SID.**
  - Stocké sur le serveur.
  - Contient les informations que le développeur veut y stocker.
- **Fichier temporaire.**
  - Sur un serveur, tous les sites partagent le même répertoire de session
  - Possibilité de vols.
- **Au démarrage d'un script :**
  - le nom du fichier est lu dans le cookie de session,
  - le fichier est ouvert et lu,
  - son contenu est désérialisé et devient le tableau `$_SESSION`.

francois.piat@univ-fcomte.fr

Sessions
`$_SESSION`
PHP 16

- **Informations stockées = variables de session.**
  - Peuvent contenir n'importe quels types de valeurs.
  - Accessibles par le tableau associatif super-global `$_SESSION`
- **`session_start()` : démarre une session**

```

graph TD
    SID{SID ?} -- oui --> Lecture[Lecture fichier]
    SID -- non --> InitSID[Init SID]
    Lecture --> Session[→ $_SESSION]
    InitSID --> InitCookie[Init cookie]
    InitCookie --> InitSession[Init $_SESSION]
    InitSession --> Session
    
```

francois.piat@univ-fcomte.fr



