



Cours PHP

Cookies, fichiers, mails et images

Par Gauthier GARNIER

Contact mail et jabber : garnier.gauthier@gmail.com



Bases PHP – Gestion des Cookies

- Les Cookies
 - Définition
 - Taille limitée
 - Stocké coté client (par le navigateur)
 - Mauvaise réputation
 - Lire/écrire/supprimer un cookie
 - PHP (à travers HTTP)
 - JS (en direct avec le navigateur)

Bases PHP – Gestion des Cookies

- Etudiez les headers HTTP
 - Outil de Webrankinfo :
<http://www.webrankinfo.com/outils/header.php>
 - Extention Firefox : liveHTTPHeaders :
<http://extensions.geckozone.org/LiveHTTPHeaders>
 - telnet :

```
> telnet www.smile.fr 80
```

```
GET /index.php HTTP/1.1  
Host: www.smile.fr
```


Bases PHP – Tableaux de cookies ?

- Cookie a valeur simple :

```
setcookie("test", "Le test a fonctionné" );
```

- Pour réaliser un tableau de cookies :

```
setcookie("test[13h40]", "Il est 13h40" );  
setcookie("test[14h12]", "Il est 14h12" );  
setcookie("test[14h42]", "Il est 14h42" );
```

Dans ce cas, les valeurs seront placées dans un tableau PHP

```
Array (  
    [test] => Array (  
        [13h40] => Il est 13h40  
        [14h12] => Il est 14h12  
        [14h42] => Il est 14h42  
    )  
)
```

Bases PHP – Manipuler les Cookies

- Récupérer le contenu d'un cookie
 - La variable : `$_COOKIE`
- Envoyer un cookie
 - **setcookie()** définit un cookie qui sera envoyé avec le reste des en-têtes. Comme pour les autres en-têtes, les cookies doivent être envoyés *avant* tout autre sortie (c'est une restriction du protocole HTTP, pas de PHP).

```
bool setcookie ( string $name  
                [, string $value  
                [, int $expire  
                [, string $path  
                [, string $domain  
                [, bool $secure  
                [, bool $httponly ]]]]] )
```

- Supprimer un cookie ?
 - `$value` → FALSE

Bases PHP – Sessions

- Définition

La session est un mécanisme qui permet à PHP de garder "en mémoire" un nombre illimité de valeurs entre plusieurs requêtes d'un même utilisateur.

➔ Stocké coté serveur

- Utilité

La session est un mécanisme qui permet à PHP de garder "en mémoire" un **nombre illimité de valeurs** entre plusieurs requêtes d'un même utilisateur.

- Confusion

L'utilisation de session est très souvent combinée à l'utilisation d'un cookie qui stockera le « PHPSESSID », les deux solutions sont complémentaires.

Bases PHP – Sessions

- `session_start()`
`session_start()` crée une session (ou restaure celle trouvée sur le serveur, via l'identifiant de session passé dans une requête GET, POST ou par un cookie).
Surveiller l'ouverture de session et l'envoi des headers HTTP
- SID
 - Constante permettant de passer facilement les variables de sessions
 - Si les cookies sont désactivés, vaut :
« PHPSESSID=0b520296150bc6e81aafcc39e65cf885 »
- Exemple : Compter le nombre de passages d'un utilisateur sur une page

```
<?php
    session_start();
    if (empty($_SESSION['count'])) {
        $_SESSION['count'] = 1;
    } else {
        $_SESSION['count']++;
    }
?>
```


Bases PHP – Mails (1/5)

- Fonction PHP :

- Mail simple

```
<?php
    mail('email_destinataire', 'sujet', 'message');
?>
```

- Mail avec header

```
<?php
    $headers = 'From: "nom"<adresse@fai.fr>'. "\n";
    $headers .= 'Reply-To: adresse_de_reponse@fai.fr'. "\n";
    $headers .= 'Content-Type: text/plain; charset="iso-8859-1"' . "\n";
    $headers .= 'Content-Transfer-Encoding: 8bit';

    if(mail('adresse_du_destinataire@fai.fr', 'Sujet', 'Message de test', $headers)) {
        echo 'Le message a bien été envoyé';
    }
    else {
        echo 'Le message n\'a pu être envoyé';
    }
?>
```


Bases PHP – Mails (2/5)

- Fonction PHP :
 - Mail HTML

```
<?php
$headers = 'From: "nom"<adresse@fai.fr>'. "\n";
$headers .= 'Reply-To: adresse_de_reponse@fai.fr'. "\n";
$headers .= 'Content-Type: text/html; charset="iso-8859-1"'. "\n";
$headers .= 'Content-Transfer-Encoding: 8bit';

$message = '<html><body>Un message de test<br><p><a href="">lien</a></p></body></html>';

if(mail('adresse_du_destinataire@fai.fr', 'Sujet', $message, $headers)) {
    echo 'Le message a été envoyé';
} else {
    echo 'Le message n\'a pu être envoyé';
}

?>
```

Bases PHP – Mails (3/5)

- Fonction PHP :
 - Mail HTML et Texte

```
<?php
// DECLARATION DES VARIABLES
$destinataire='mail_destinataire@fai.fr';
$email_expediteur='votre_mail@fai.fr';
$email_reply='email_de_reponse@fai.fr';

$message_texte='Bonjour,.'.\n\n'. 'Message au format texte';
$message_html=<html>
    <head>
        <title>Titre</title>
    </head>
    <body>Test de message</body>
</html>;

//GENERE LA FRONTIERE ENTRE TEXTE ET HTML
$frontiere = '-----=' . md5(uniqid(mt_rand()));

//HEADERS DU MAIL
$headers = 'From: "Nom" <'. $email_expediteur. '>'. "\n";
$headers .= 'Return-Path: <'. $email_reply. '>'. "\n";
$headers .= 'MIME-Version: 1.0'. "\n";
$headers .= 'Content-Type: multipart/alternative; boundary="'. $frontiere. '"';
```

```
//MESSAGE TEXTE
$message = 'This is a multi-part message in MIME format.'. "\n\n";

$message .= '--'. $frontiere. '--'. "\n";
$message .= 'Content-Type: text/plain; charset="iso-8859-1"'. "\n";
$message .= 'Content-Transfer-Encoding: 8bit'. "\n\n";
$message .= $message_texte. "\n\n";

//MESSAGE HTML
$message .= '--'. $frontiere. '--'. "\n";
$message .= 'Content-Type: text/html; charset="iso-8859-1"'. "\n";
$message .= 'Content-Transfer-Encoding: 8bit'. "\n\n";
$message .= $message_html. "\n\n";

$message .= '--'. $frontiere. '--'. "\n";

if(mail($destinataire,$sujet,$message,$headers)) {
    echo 'Le mail a été envoyé';
} else {
    echo 'Le mail n\'a pu être envoyé';
}
?>
```


Bases PHP – Mails (4/5)

- Fonction PHP :
 - Mail avec pièces jointes

```
<?php

// Reprise du code de l'exemple : Mail HTML + Texte

//PIECE JOINTE
$message .= 'Content-Type: image/jpeg; name="nom_du_fichier.jpg"'."\n";
$message .= 'Content-Transfer-Encoding: base64'."\n";
$message .= 'Content-Disposition:attachement; filename="nom_du_fichier.jpg"'."\n\n";

$message .= chunk_split(base64_encode(file_get_contents('nom_du_fichier.jpg')))."\n";

?>
```

Bases PHP – Mails (5/5)

- Fonction mail() de PHP
 - Complexité et lourdeur

- Fonction PEAR::Mail

<http://pear.php.net/manual/fr/package.mail.mail.send.php>

- Fonction Zend_Mail

<http://framework.zend.com/manual/fr/zend.mail.html>

Bases PHP – Envoi de Fichiers (1/4)

- Formulaire HTML d'envoi d'un fichier

<!-- Le type d'encodage des données, enctype, DOIT être spécifié comme ce qui suit -->

```
<form enctype="multipart/form-data" action="_URL_" method="post">
```

<!-- MAX_FILE_SIZE doit précéder le champs input de type file -->

```
<input type="hidden" name="MAX_FILE_SIZE" value="30000" />
```

<!-- Le nom de l'élément input détermine le nom dans le tableau \$_FILES -->

Envoyez ce fichier :

```
<input name="userfile" type="file" />
```

```
<input type="submit" value="Envoyer le fichier" />
```

```
</form>
```

Bases PHP – Envoi de Fichiers (2/4)

- Formulaire HTML d'envoi d'une liste de fichiers

<!-- Le type d'encodage des données, enctype, DOIT être spécifié comme ce qui suit -->

```
<form enctype="multipart/form-data" action="_URL_" method="post">
```

<!-- MAX_FILE_SIZE doit précéder le champs input de type file -->

```
<input type="hidden" name="MAX_FILE_SIZE" value="30000" />
```

<!-- Le nom de l'élément input détermine le nom dans le tableau \$_FILES -->

Envoyez ces fichiers :

```
<input name="userfile[]" type="file" />
```

```
<input name="userfile[]" type="file" />
```

```
<input name="userfile[]" type="file" />
```

```
<input name="userfile[]" type="file" />
```

```
<input name="userfile[]" type="file" />
```

```
<input type="submit" value="Envoyer les fichiers" />
```

```
</form>
```


Bases PHP – Envoi de Fichiers (3/4)

- Récupération des fichiers correctement uploadés :

```
foreach ($_FILES["userfile"]["error"] as $key => $error) {  
    if ($error == UPLOAD_ERR_OK) {  
        $tmp_name = $_FILES["userfile"]["tmp_name"][$key];  
        $name = $_FILES["userfile"]["name"][$key];  
        move_uploaded_file($tmp_name, "data/$name");  
    }  
}
```

Bases PHP – Envoi de Fichiers (4/4)

- Conseils sur la gestion des fichiers uploadés

Comme d'habitude, lorsque l'on introduit sur une machine (*ici le serveur*) des documents provenant de l'extérieur, il faut être extrêmement prudent :

- protéger les accès à l'upload de fichiers (*Authentication HTTP Basic ou Digest, verrouillage par adresse IP, protections gérées depuis PHP*),
- effectuer toutes les vérifications imaginables avant d'accepter un document (*type du fichier, taille maximum, extension...*),
- ne pas rendre directement accessibles les documents déposés,
- et surtout ne JAMAIS rendre exécutable un document provenant de l'extérieur et non vérifié...

- Configurations à surveiller

- Vérifier les directives : `file_uploads`, `upload_max_filesize`, `upload_tmp_dir`, `post_max_size` et `max_input_time` dans *php.ini*

Bases PHP – Images

- Gestion d'images via PHP :
 - GD
 - Intégré simplement à PHP
 - Performance et qualité médiocre
 - Outil externe ImageMagick
 - Outil externe
 - Bonne qualité du résultat



Bases PHP – Rappels Apache

- Apache est un serveur web
- Avec PHP, il doit être installer en mod MPM prefork (Multi-Processing Module) -> car le moteur PHP n'est pas nativement thread-safe
- Pour effectuer la configuration propre d'un site, on utilise ce que l'on appelle des VirtualHost (Hote virtuel)
- Cela permet d'avoir une configuration Apache dédié à un site (SiteName)
- Cf : <http://httpd.apache.org/docs/2.2/en/vhosts/examples.html>