

SERVEUR WEB APACHE TP-3

Documents d'erreurs, Dossiers personnels des utilisateurs.

Objectifs du TP

- 1- Personnaliser les erreurs http 404, 403 et 500
- 2- Configuration de l'accès aux répertoires personnels des utilisateurs
- 3- Utilisation du fichier « .htaccess » pour permettre aux utilisateurs de personnaliser une partie de la configuration

1- Personnalisation des erreurs.

Quand le serveur rencontre une erreur, il avertit l'utilisateur en lui envoyant une page indiquant le code http de l'erreur. Les pages configurées dans « **apache** » ne sont certainement pas adaptées à la charte graphique de notre site.

Pour personnaliser ces pages, on utilise la directive « **ErrorDocument** » dans le fichier de configuration de « **apache** ». Cette directive s'applique selon la section de configuration dans laquelle elle est définie.

Par exemple, on peut définir cette directive pour le répertoire de base dans la section de configuration globale. Puis la redéfinir pour un répertoire spécifique qui peut être réel ou virtuel.

Voici les trois syntaxes possibles :

```
ErrorDocument 500 http://monsite.com/erreurs/er500.aspx
ErrorDocument 404 /erreurs/er404.html
ErrorDocument 403 "Désolé, vous n'avez pas l'autorisation d'accès"
```

On peut aussi définir un chemin relatif pour accéder à la page d'erreur ; « **erreurs/err404.htm** »

Il faut alors que ces pages existent et que les chemins soient corrects, sinon vous allez générer une « **double erreur !** ».

2- Répertoires personnels des utilisateurs

Dans le système d'exploitation « **linux/unix** » les dossiers personnels des utilisateurs sont tous sous le dossier « **/home** ». Sous « **Windows** », ils sont dans le dossier « **Users** ».

Que ce soit sur « **Linux** » ou « **Windows** », chaque utilisateur a son propre dossier et possède tous les droits sur ce dossier. Il est donc libre, s'il le veut de partager un dossier (autrement dit un site perso) avec tout le monde.

Si « **Apache** » est configuré pour ça, et si on suppose que l'utilisateur s'appelle « **toto** ». Alors on pourra accéder au dossier partagé par cet utilisateur par l'URL : <http://nomdusite.com/~toto/>

Remarquez le caractère « **~** » (prononcez : **tild**) et le nom de l'utilisateur.

C'est très pratique, car il suffit de créer un compte pour l'utilisateur pour qu'il ait le droit de partager des données sur le web. Aucune autre configuration ne sera nécessaire.

Passons à la pratique pour mieux comprendre.

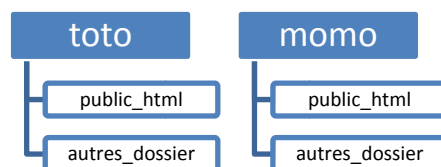
ATTENTION :

Il ne faut surtout pas toucher aux comptes systèmes des machines dans lesquelles vous allez faire ces manipulations.

Puisque nous voulons juste tester, nous allons simuler des utilisateurs en créant un dossier que nous allons appeler « **users** » sous le répertoire « **Apache** ». Nous aurons donc :

« **C:\Program Files\Apache Software Foundation\Apache2.2\users** »

Pour pouvoir tester, nous allons créer deux sous-dossiers :



Les dossiers « **public_html** » seront les dossiers à partager.

Maintenant côté « **Apache** », nous allons éditer le fichier de configuration « **httpd.conf** ».

a- Activation du module « **userdir** ».

Dans le fichier de configuration dé-commenter la ligne : (aux alentours de 126)

```
LoadModule userdir_module modules/mod_userdir.so
```

b- Dé-commenter la ligne :

```
# User home directories
Include conf/extra/httpd-userdir.conf
```

Cela va inclure le fichier « **httpd-userdir.conf** » suivant :

```

# Nom du dossier de partage
UserDir " C:/chemin_vers_dossier_users/*/public_html "
# Section de configuration relative aux dossiers "public_html"
# de tous les utilisateurs.
# Remarquez le caractère JOKER « * » qui masque tous
# les noms d'utilisateurs.
<Directory "C:/chemin_vers_dossier_users/*/public_html">
    # Cette directive concerne le fichier « .htaccess »
    # qu'on verra plus loin.
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    # Ici, on limite l'accès en terme de methodes HTTP utilisées
    <Limit GET POST OPTIONS>
        Order allow,deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS>
        Order deny,allow
        Deny from all
    </LimitExcept>
</Directory>

```

Les modifications à apporter sont marquées en rouge.

On peut alors tester après avoir redémarré « Apache ». En y accédant : <http://localhost/~momo>

3- Le fichier « .htaccess »

Le fichier « .htaccess » permet à l'utilisateur même s'il n'a pas accès au fichier de configuration de paramétrer l'accès à son propre répertoire personnel.

Dans la section précédente, quand nous avons réglé les droits d'accès aux répertoires des utilisateurs, nous avons ajouté la directive « **AllowOverride** ». Celle-ci a pour rôle de donner les autorisations nécessaire pour pouvoir utiliser « .htaccess ».

Reamrque :

*Le nom « .htaccess » n'est pas obligatoire, on pourrait utiliser un autre nom comme « .htentree » ou autre. Mais il est conseillé de garder le même nom. Ce nom est configuré par la directive « **accessFileName** » dans le fichier « **httpd-default.conf** ». Nous y reviendrons une prochaine fois.*

Pour le moment, on peut juste utiliser les des fichiers « .htaccess » dans le dossier « **public_html** » des utilisateurs pour tester.

Ajouter un fichier nommé « **.htaccess** » dans le répertoire « **public_html** ». et mettez-y les directives suivantes :

```
Options -indexes
DirectoryIndex blablabla.html
Allow from 127.0.0.1
```

Essayez d'accéder à :

http://votre_adresse/~toto

<http://localhost/~toto>

Ajouter un dossier « docs » dans le dossier « **public_html** » et mettez-y le fichier « **.htaccess** » suivant :

```
Options +indexes
Allow from all
```

Essayez d'accéder à :

http://votre_adresse/~toto/docs

<http://localhost/~toto/docs>

Que constatez-vous ?