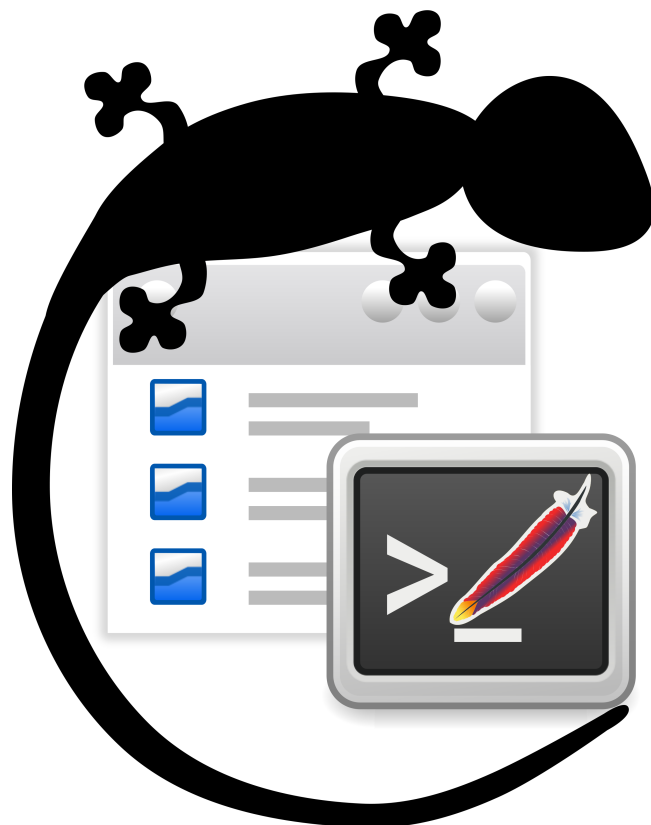




Installation sur un serveur personnel

Sébastien Raud



beCms version 1.0b

<http://www.becms.org>

Date du document : 30 avril 2013

Licence Creative Commons (CC BY - NC - SA 2.0)

Table des matières

1	Introduction	1
1.1	Contenu du document	1
1.2	beCms en bref	1
1.3	Pré-requis	2
2	Installation du serveur Apache 2 et de PHP 5	3
2.1	Introduction	3
2.2	Systèmes GNU/Linux	4
2.2.1	Debian, Ubuntu et distributions dérivées	4
2.2.1.1	Installation	4
2.2.1.2	Ajouter le mod_rewrite à Apache	4
2.2.1.3	Références	5
2.2.2	openSUSE	5
2.2.2.1	Installation via YaST	5
2.2.2.2	Installation via ZYpp	5
2.2.2.3	Ajouter le mod_rewrite à Apache	6
2.2.2.4	Références	6
2.2.3	Red Hat, CentOS	7
2.2.3.1	Installation	7
2.2.3.2	Ajouter le mod_rewrite à Apache	7
2.2.3.3	Références	7
2.2.4	Fedora	8
2.2.4.1	Installation	8
2.2.4.2	Ajouter le mod_rewrite à Apache	8
2.2.4.3	Références	8
2.3	Systèmes Windows	9
2.3.1	Téléchargement de WampServer	9
2.3.2	Installation de WampServer	9
2.3.3	Lancer WampServer et démarrer les services	9
2.3.4	Ajouter le mod_rewrite à Apache	11
3	Préparer beCms	13
3.0.5	Généralités	13
3.0.6	Système GNU/Linux : fixer les droits	14
3.0.6.1	Utilisateur et Groupe Apache	14

3.0.6.2	Ajouter votre utilisateur au groupe Apache	15
3.0.6.3	Changer les droits du répertoire de beCms	15
3.0.6.4	Ajouter les droits en écriture sur le répertoire de beCms	15
4	Configuration Apache	17
4.1	Introduction	17
4.2	Paramétrer un hôte virtuel	17
4.2.1	Déclaration de l'adresse web	17
4.2.1.1	Systèmes GNU/Linux : modification du fichier hosts	17
4.2.1.2	Systèmes Windows : modification du fichier hosts	18
4.2.1.3	Explications sur le fichier hosts	18
4.2.2	Ajout de l'hôte virtuel dans Apache	19
4.2.2.1	Systèmes GNU/Linux	19
4.2.2.2	Système Windows	19
4.2.3	Contenu du fichier mon-becms.conf	20
4.2.3.1	Notes sur l'arborescence des fichiers	21
4.2.4	Test	22
4.3	Principe des fichiers .htaccess	22
4.3.1	Introduction	22
4.4	Création du fichier .htaccess	23
4.5	Configuration de base	24
4.5.1	Bloquer le listage des répertoires	24
4.5.2	Réécriture d'URLs	24
4.5.3	Protection du répertoire /cms	25
4.6	Aller plus loin dans la configuration	25
4.6.1	Options de configuration de PHP	25
4.6.1.1	Register globals	26
4.6.1.2	Magic quotes	26
4.6.1.3	Short open tag	26
4.6.2	Améliorer la sécurité en utilisant la réécriture d'URLs	27
4.6.2.1	Autoriser l'accès aux fichiers du répertoire /web	27
4.6.2.2	Autoriser l'accès à un fichier situé à la racine du CMS	27
4.6.2.3	Améliorer les performances du cache	28
4.6.2.4	Exemple de configuration du mod_rewrite	30
4.6.3	Optimiser votre site web	31
4.6.3.1	Activer la compression	31
4.6.3.2	Mise en cache navigateur et dates d'expirations	32
4.6.4	Exemple complet de fichier .htaccess	33
5	Configurer le CMS	37

5.1	Introduction	37
5.2	Accès à l'interface de configuration	37
5.3	Présentation de l'interface de configuration	37
5.3.1	Onglet « Informations serveur »	38
5.3.2	Onglet « Gestion des utilisateurs »	38
5.3.3	Onglet « Droits répertoires et fichiers »	40
5.3.4	Fin de l'installation	40
6	Lexique	43
7	Licence et crédits	45
7.1	Licence	45
7.2	Crédits	45

Introduction

1.1 Contenu du document

Ce document présente l'installation de beCms 1.0b sur un serveur personnel, en prenant l'exemple de votre poste personnel comme serveur.

Ce document se base sur l'utilisation du serveur web Apache 2.x^{1 2}.

Ce document porte sur les systèmes d'exploitation GNU/Linux (différentes distributions majeures) et Windows.

Ce document peut également servir de base, avec de légères adaptations, pour une installation sur un serveur dédié ou sur d'autres systèmes d'exploitations.



Il existe d'autres serveurs que Apache, comme nginx^a, lighttpd^b, cherokee^c ou IIS. Mais Apache étant le serveur le plus populaire chez les hébergeurs, cette documentation ne porte que sur Apache.

- a. <http://wiki.nginx.org>
- b. <http://www.lighttpd.net/>
- c. <http://www.cherokee-project.com/>

1.2 beCms en bref

beCms est un CMS (Content Management System) basique, mais extensible, développé en PHP (PHP Hypertext Processor) 5.0.

- 1. <http://httpd.apache.org/>
- 2. http://fr.wikipedia.org/wiki/Apache_HTTP_Server

beCms permet de gérer des sites web simples. Il offre une interface de gestion de sites web pour créer ou modifier des pages et ajouter des documents ou des images. Il demande peu de ressources, n'utilise que des bibliothèques PHP standard et peut donc être utilisé sur un grand nombre de serveurs web.

1.3 Pré-requis

Avant de commencer une installation de beCms, il convient de télécharger l'archive (fichier becms-*.zip ou becms-*.tar.gz) du cms depuis le site <http://www.becms.org>.

Votre système doit posséder :

- Un serveur PHP :
 - PHP 5.x ;
 - la librairie GD est un plus pour la génération automatique d'images miniatures.
- Apache 2 :
 - le module « mod_rewrite » est un plus pour avoir des URLs élégantes.

L'installation et la configuration du serveur Apache 2 et de PHP 5 sur différents systèmes d'exploitations fait l'objet de chapîtres dédiés.

Vous devez également posséder un navigateur web, l'idéal étant d'avoir un navigateur récent.

Enfin, un logiciel de décompression est nécessaire pour utiliser l'archive (le fichier téléchargé depuis le site <http://www.becms.org>). Si vous n'en possédez pas, vous pouvez utiliser le logiciel 7-zip³, logiciel libre et multi-plateformes.

3. <http://www.7-zip.org/>

Installation du serveur Apache 2 et de PHP 5

2.1 Introduction

Cette partie présente l'installation du serveur Apache 2 et de PHP 5 sur différents systèmes d'exploitations.

Nous installerons :

- Apache 2 et le module de réécriture d'URLs `mod_rewrite`, qui permet d'obtenir des URLs élégantes ;
- PHP 5 et au moins le module GD2 qui permet de créer des images miniatures.

D'autres modes d'installation sont possibles, mais ils ne sont pas décrits dans ce document.



En cas de problème lors de cette installation, référez-vous à la documentation officielle de votre système ou de votre distribution, ou à la documentation fournie par la communauté liée à votre distribution.

2.2 Systèmes GNU/Linux

2.2.1 Debian, Ubuntu et distributions dérivées

2.2.1.1 Installation



La méthode d'installation proposée se base sur Apt.
Si vous utilisez le gestionnaire de paquets Aptitude, remplacez la commande `apt-get` par `aptitude` dans les instructions.

L'installation du serveur Apache et de PHP 5 peut se faire simplement en ligne de commande (dans un terminal) par les instructions suivantes :

```
# mise à jour de la liste des paquets
sudo apt-get update

# installation du serveur Apache 2
sudo apt-get install apache2

# installation de PHP 5 et du module PHP 5 / Apache 2
sudo apt-get install php5 libapache2-mod-php5

# installation de la bibliothèque GD de PHP 5
sudo apt-get install php5-gd

# redémarrer Apache
sudo /etc/init.d/apache2 restart
```

2.2.1.2 Ajouter le `mod_rewrite` à Apache

Pour ajouter le `mod_rewrite` à Apache, il faut exécuter la commande suivante :

```
# installation du module de réécriture d'URL d'Apache
sudo a2enmod rewrite
```

2.2.1.3 Références

- <http://wiki.debian.org/LaMp> ;
- <http://doc.ubuntu-fr.org/lamp> ;
- <http://doc.ubuntu-fr.org/apache2> ;
- <http://doc.ubuntu-fr.org/php>.

2.2.2 openSUSE

2.2.2.1 Installation via YaST

L'installation du serveur Apache et de PHP 5 peut se faire simplement en ligne de commande (dans un terminal) par les instructions suivantes :

```
# installation du serveur Apache 2
yast2 -i apache2

# configuration du serveur : lancement automatique au démarrage de la machine
chkconfig --add apache2

# démarrage du serveur
/etc/init.d/apache2 start

# installation de PHP 5 et du module PHP 5 / Apache 2
yast2 -i apache2-mod_php5

# installation de la bibliothèque GD de PHP 5
yast2 -i php5-gd

# redémarrage du serveur
/etc/init.d/apache2 start
```

2.2.2.2 Installation via ZYpp

L'installation du serveur Apache et de PHP 5 peut se faire simplement en ligne de commande (dans un terminal) par les instructions suivantes :

```
# installation du serveur Apache 2
zypper install apache2

# configuration du serveur : lancement automatique au démarrage de la machine
chkconfig --add apache2
```

```
# démarrage du serveur
/etc/init.d/apache2 start

# installation de PHP 5 et du module PHP 5 / Apache 2
zypper install apache2-mod_php5

# installation de la bibliothèque GD de PHP 5
zypper install php5-gd

# redémarrage du serveur
/etc/init.d/apache2 start
```

2.2.2.3 Ajouter le mod_rewrite à Apache

Pour ajouter le mod_rewrite à Apache, il faut :

- éditer le fichier `/etc/sysconfig/apache2` en tant qu'utilisateur `root` ;
- chercher dans le fichier la ligne qui commence par **APACHE_MODULES** ;
- ajouter **rewrite** à cette ligne, par exemple : `APACHE_MODULES="actions alias auth_basic authn_file authz_host authz_groupfile authz_default authz_user autoindex cgi dir env expires include log_config mime negotiation setenvif ssl userdir php5 rewrite"` ;
- enregistrer le fichier ;
- lancer la commande `SuSEconfig` pour mettre à jour la configuration ;
- redémarrer le serveur : `run /etc/init.d/apache2 restart`.

2.2.2.4 Références

- <http://www.howtoforge.com/installing-apache2-with-php5-and-mysql-support-on-opensuse-11.2-lamp> ;
- <http://www.unixmen.com/installing-apache2-with-php5-and-mysql-on-opensuse-121/> ;
- http://enarion.net/web/htaccess/mod_rewrite-on-suse/.

2.2.3 Red Hat, CentOS

2.2.3.1 Installation

L'installation du serveur Apache et de PHP 5 peut se faire simplement en ligne de commande (dans un terminal) par les instructions suivantes :

```
# installation du serveur Apache 2
sudo yum install httpd

# démarrage du serveur
sudo /etc/init.d/httpd start

# installation de PHP 5 et des modules utiles
sudo yum install php php-common php-gd php-mbstring php-mcrypt php-devel php-xml

# démarrage du serveur
sudo /etc/init.d/httpd start

# configuration du serveur : lancement automatique au démarrage de la machine
sudo chkconfig httpd on
```

2.2.3.2 Ajouter le mod_rewrite à Apache

Pour ajouter le mod_rewrite à Apache, il faut :

- éditer le fichier `/etc/httpd/conf/httpd.conf` en tant qu'utilisateur root ;
- chercher la ligne contenant le texte `LoadModule rewrite_module modules/mod_rewrite.so` et la décommenter (enlever le # de début de ligne) ;
- enregistrer le fichier ;
- redémarrer le serveur : `sudo /etc/init.d/httpd start`.

2.2.3.3 Références

- <http://articles.slicehost.com/2008/2/6/centos-installing-apache-and-php5>.

2.2.4 Fedora

2.2.4.1 Installation

L'installation du serveur Apache et de PHP 5 peut se faire simplement en ligne de commande (dans un terminal) par les instructions suivantes :

```
# installation du serveur Apache 2
su -c 'yum install httpd-manual'

# démarrage du serveur
su -lc 'service httpd start'

# installation de PHP 5
su -lc 'yum install php'

# installation des modules utiles de PHP 5
yum install php-gd php-mbstring

# redémarrage du serveur
su -lc 'service httpd reload'
```

2.2.4.2 Ajouter le mod_rewrite à Apache

Le mod_rewrite est installé par défaut. Mais il faut le rendre accessible par les fichiers **.htaccess**. Pour se faire, il faut :

- éditer le fichier /etc/httpd/conf/httpd.conf en tant qu'utilisateur root ;
- rechercher l'instruction AllowOverride None et la changer en AllowOverride All ;
- enregistrer le fichier ;
- redémarrer le serveur : su -lc 'service httpd reload'.

2.2.4.3 Références

- http://doc.fedora-fr.org/wiki/Installation_et_configuration_d%27Apache ;
- http://doc.fedora-fr.org/wiki/Installation_et_configuration_de_PHP.

2.3 Systèmes Windows

Une solution simple pour installer, configurer et utiliser Apache 2 et PHP 5 est d'installer le logiciel libre « WampServer ».

2.3.1 Téléchargement de WampServer

Téléchargez le fichier d'installation de WampServer depuis le site de l'application : <http://www.wampserver.com>

2.3.2 Installation de WampServer

Lorsque le fichier d'installation est téléchargé, il suffit de double-cliquer sur le fichier puis de suivre la procédure d'installation.

Si l'on se contente de cliquer sur le bouton « next », le répertoire d'installation par défaut est `c:\wamp`.

À la fin de l'installation, l'installateur propose de lancer WampServer.

2.3.3 Lancer WampServer et démarrer les services

Pour lancer WampServer, il faut aller dans le menu Démarrer, trouver l'item « WampServer » et lancer « start WampServer ».

Une icône s'ajoute près de l'heure, en bas à droite de l'écran (zone systray) :



L'icône peut prendre trois couleurs :

- rouge lorsque rien n'est activé ;
- orange lorsque les services sont démarrés, mais hors ligne ;
- vert lorsque les services sont démarrés et en ligne.

Un clic droit sur l'icône permet de configurer la langue du logiciel et de quitter le logiciel.

Le clic gauche sur l'icône permet d'accéder aux fonctionnalités du logiciel :

- **localhost** : lance le navigateur web sur la page d'accueil de votre installation web ;

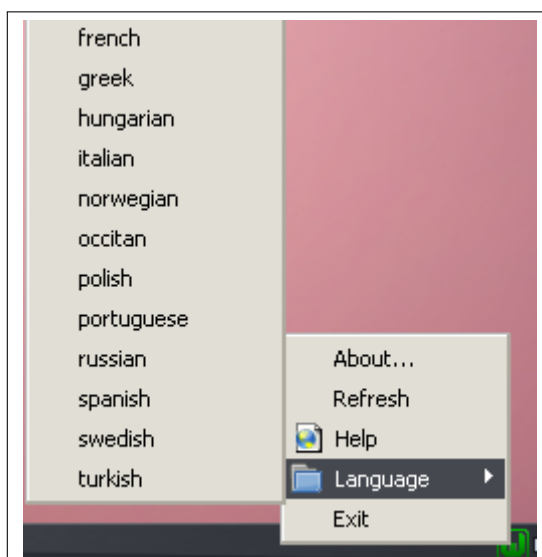


FIGURE 2.1 – WampServer – Clic droit

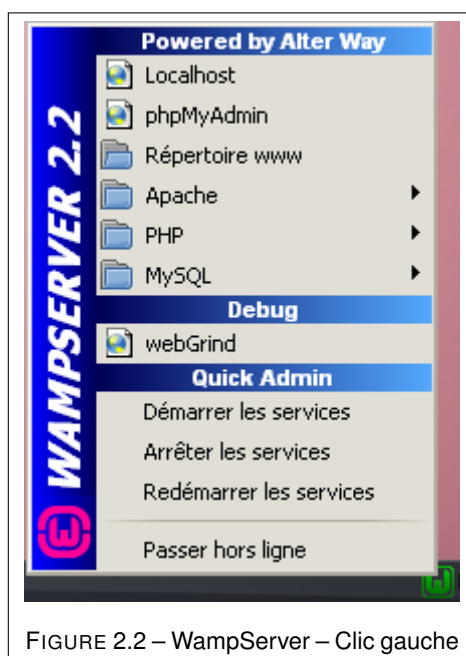


FIGURE 2.2 – WampServer – Clic gauche

- **phpMyAdmin** : lance le navigateur web sur la page de gestion de la base de données MySQL ;
- **Répertoire www** : c'est le répertoire web racine, celui qui peut contenir vos sites web ;
- **Apache, PHP, MySQL** : accès à la configuration du serveur Apache, de PHP et de MySQL ;
- **Démarrer les services, Arrêter les services, Redémarrer les services** : gestion des services ;
- **Passer en ligne, Passer hors ligne** : activer ou désactiver les services.

Pour activer Apache et PHP, il faut donc :

- Démarrer les services ;
- Passer en ligne.

2.3.4 Ajouter le `mod_rewrite` à Apache

Pour ajouter le `mod_rewrite` il faut :

- faire un clic gauche sur l'icone WampServer ;
- aller dans Apache/Modules Apache ;
- cliquer sur **rewrite_module**.

Préparer beCms

3.0.5 Généralités

Lorsque le serveur Apache et PHP 5 sont installés sur votre machine, vous pouvez déjà préparer le répertoire qui contiendra beCms.

Pour cela, téléchargez une archive de beCms sur le site <http://www.becms.org>. Vous devrez ensuite décompresser l'archive dans un répertoire qui deviendra le répertoire de votre site. Le répertoire décompressé porte un nom long, vous pouvez le renommer en `becms` par exemple. C'est ce nom qui sera utilisé dans la suite de la documentation.

Sur les systèmes GNU/Linux le répertoire `/var/www` est normalement dédié aux sites web. Vous pouvez donc décompresser l'archive dans ce répertoire.

Sur les systèmes Windows qui utilisent WampServer, le répertoire `c:\wamp\www` (pour une installation par défaut), ou le répertoire qui se trouve sous la dénomination **Répertoire www** lors du clic gauche sur l'icône WampServer, est normalement dédié aux sites web. Vous pouvez donc décompresser l'archive dans ce répertoire.

La suite de la documentation utilise le répertoire `/var/www/becms`, veillez à adapter le nom à votre configuration (par exemple `c:\wamp\www\becms...`)



Ce répertoire sera le répertoire d'installation de beCms, donc le répertoire de votre site web.

N'oubliez pas où se trouve ce répertoire, vous en aurez besoin pour la suite !

3.0.6 Système GNU/Linux : fixer les droits

Lorsque vous avez décompressé l'archive, que ce soit dans un sous-répertoire de `/var/www` ou ailleurs, un problème de droits utilisateurs risque de se poser.

En effet le serveur Apache est généralement un utilisateur particulier appartenant à un groupe utilisateur.

Ce mode de fonctionnement permet une plus grande sécurisation du système et un paramétrage fin. La contre-partie est que, par défaut, suite à la décompression de l'archive, Apache ne pourra écrire dans votre répertoire, et le CMS est alors inutilisable.

Pour résoudre ce souci, vous pouvez :

- ajouter l'utilisateur courant (celui que vous êtes) au groupe lié à Apache ;
- faire appartenir le répertoire du CMS (et ses sous-répertoires) à l'utilisateur et au groupe liés à Apache ;
- ajouter les droits d'écriture sur le répertoire du CMS (et ses sous-répertoires) à votre utilisateur.

3.0.6.1 Utilisateur et Groupe Apache

Le tableau ci-dessous liste l'utilisateur et le groupe Apache par distribution :

Distribution	Utilisateur	Groupe
Debian, Ubuntu et distributions dérivées	www-data	www-data
openSuse	wwwrun	www
Red Hat, CentOS, Fedora	apache	apache

Pour vérifier que l'utilisateur et le groupe sont bien ceux indiqués dans le tableau, vous pouvez lancer les commandes (exemple pour Debian, Ubuntu...) :

```
# l'utilisateur www-data existe-t-il?  
grep www-data /etc/passwd  
  
# le groupe www-data existe-t-il?  
grep www-data /etc/group
```

Si vous obtenez une réponse du type `www-data:x:33:33:www-data:/var/www:/bin/sh` au premier test et une réponse du type `www-data:x:33:utilisateur`, c'est que ce sont les bonnes informations.

Si ce n'est pas le cas, référez vous à la documentation de votre distribution.

3.0.6.2 Ajouter votre utilisateur au groupe Apache

La commande pour ajouter votre utilisateur au groupe Apache (exemple pour Debian, Ubuntu...) :

```
sudo useradd -G www-data $USER
```

3.0.6.3 Changer les droits du répertoire de beCms

Si le CMS est installé sous répertoire `/var/www/becms`, la commande est :

```
sudo chown -R www-data:www-data /var/www/becms
```

La syntaxe est de la forme `utilisateur:groupe`.

3.0.6.4 Ajouter les droits en écriture sur le répertoire de beCms

Si le CMS est installé sous le répertoire `/var/www/becms`, la commande est :

```
sudo chmod -R g+w /var/www/becms
```


Configuration Apache

4.1 Introduction

Après avoir installé Apache, PHP 5 et beCms sur votre poste, il faut maintenant configurer le serveur pour pouvoir faire fonctionner le site.

4.2 Paramétrer un hôte virtuel

Le serveur web Apache peut gérer plusieurs sites web grâce à la notion d'**hôtes virtuels**. Chaque site peut alors avoir une configuration spécifique.

Nous allons ici créer un **hôte virtuel** qui permettra de lier une adresse web – nous utiliserons l'URL `http://www.mon-becms.com` – à notre répertoire d'installation de beCms (cf Préparer beCms).

4.2.1 Déclaration de l'adresse web

L'adresse web `www.mon-becms.com` est une adresse fictive : elle n'existera que sur le poste serveur.

Vote système possède un fichier nommé `hosts` qui sert à déclarer ce type d'adresse web.

4.2.1.1 Systèmes GNU/Linux : modification du fichier `hosts`

Sur les systèmes GNU/Linux, c'est le fichier `/etc/hosts` qui permet la déclaration d'une adresse web locale.

Vous devez donc :

- éditer le fichier `/etc/hosts` en tant qu'utilisateur `root` ;

- ajouter dans ce fichier la ligne : `127.0.0.1 www.mon-becms.com` ;
- enregistrer le fichier.

4.2.1.2 Systèmes Windows : modification du fichier hosts

Sur les système Windows, c'est le fichier `c:\windows\system32\drivers\etc\hosts` qui permet la déclaration d'une adresse web locale.



Suivant les versions de Windows le répertoire `c:\windows` peut être remplacé par `c:\winnt`.

La documentation utilise le répertoire `c:\windows`, il vous faut donc adapter le nom du répertoire à votre système.

Vous devez donc :

- éditer le fichier `c:\windows\system32\drivers\etc\hosts` ;
- ajouter dans ce fichier la ligne : `127.0.0.1 www.mon-becms.com` ;
- enregistrer le fichier.

4.2.1.3 Explications sur le fichier hosts

Ce fichier sert à faire un lien entre une adresse web – ici `www.mon-becms.com` – et une adresse IP (Internet Protocol).

La syntaxe générale du fichier est :

```
adresse_ip adresse_web_1 adresse_web_2 ... adresse_web_n
```

La séparation entre les champs peut être un ou plusieurs espaces blancs et / ou une ou plusieurs tabulations.

Le caractère dièse (#) permet d'ajouter des commentaires. Ainsi vous pouvez par exemple écrire dans le fichier :


```
# site de test beCms
127.0.0.1      www.mon-becms.com
```

L'adresse IP 127.0.0.1 est l'adresse de votre serveur local, c'est une adresse IP standard.

Maintenant lorsque vous demandez à votre navigateur d'aller à l'adresse web `http://www.mon-becms.com` la requête du navigateur arrive à votre serveur Apache.

4.2.2 Ajout de l'hôte virtuel dans Apache

Nous allons maintenant indiquer à Apache que l'URL `http://www.mon-becms.com` correspond au répertoire d'installation du CMS.

4.2.2.1 Systèmes GNU/Linux

Le tableau ci-dessous indique le répertoire de configuration des **hôtes virtuels** d'Apache par distribution :

Distribution	Répertoire
Debian, Ubuntu et distributions dérivées	<code>/etc/apache2/sites-enabled</code>
openSuse	<code>/etc/apache2/vhosts.d/</code>
Red Hat, CentOS, Fedora	<code>/etc/httpd/conf.d/</code>

C'est donc dans ce répertoire qu'il faudra créer un fichier de configuration.

Vous pouvez déjà créer ce fichier, que nous allons nommer `mon-becms.conf`, par la commande suivante (exemple pour Debian, Ubuntu... indiquez le répertoire correspondant à votre distribution) :

```
sudo touch /etc/apache2/sites-enabled/mon-becms.conf
```

4.2.2.2 Système Windows

Avec WampServer, le répertoire de configuration des **hôtes virtuels** est le répertoire `c:\wamp\bin\apache\apache2.2.22\conf\extra`.



Le nom du répertoire peut être modifié en fonction de votre installation : la partie `apache2.2.22` étant variable en fonction du numéro de version de votre serveur Apache.

Dans ce répertoire, vous pouvez créer un fichier `mon-becms.conf`.

Ensuite il faut éditer le fichier `c:\wamp\bin\apache\apache2.2.22\conf\httpd.conf` puis chercher la ligne : `# Virtual hosts`.

Après cette ligne, vous pouvez ajouter la ligne suivante :

```
Include conf/extra/mon-becms.conf
```

qui indique à Apache d'utiliser le fichier `mon-becms.conf` comme un fichier d'hôte virtuel.

4.2.3 Contenu du fichier `mon-becms.conf`

C'est ce fichier `mon-becms.conf` qui doit indiquer à Apache le lien entre l'URL `www.mon-becms.com` et le répertoire d'installation du CMS. Son contenu doit être le suivant :

```
01 <VirtualHost *:80>
02     # activation de la réécriture d'URLs
03     RewriteEngine on
04
05     # déclaration nom de domaine
06     ServerName www.mon-becms.com
07
08     # répertoire correspondant : adaptez-le à votre répertoire
09     DocumentRoot /var/www/becms
10
11     # fichier PHP qui doit être lancé par le serveur
12     DirectoryIndex index.php
13
14     # fichier de configuration Apache utilisé dans le répertoire
15     AccessFileName .htaccess
16
17     # autorisations particulières pour le répertoire du CMS
18     # adaptez-le à votre répertoire
19     <Directory /etc/var/becms>
20         AllowOverride All
21         Allow from All
```

```
22     </Directory>
23 </VirtualHost>
```

Après avoir enregistré le fichier, il faut redémarrer le serveur Apache.

4.2.3.1 Notes sur l'arborescence des fichiers

Lorsque l'archive est décompressée, l'arborescence de fichiers est la suivante :

```
index.php
install.php
+ cms
+ web
    index.php
    install.php
```

Dans le fichier de configuration de l'hôte **vituel**, nous avons fait pointer le serveur vers le fichier `index.php` du répertoire racine de beCms.

Cependant le répertoire `/web` du CMS contient également un fichier `index.php`, qui a le même rôle que celui du répertoire racine. L'intérêt de ce second fichier est de simplifier la sécurisation du CMS.

En effet, en indiquant au serveur le répertoire racine comme point de départ du site, les fichiers du répertoire `/cms`, qui sont les fichiers du cœur du logiciel, sont accessibles aux internautes. Il faut alors ajouter un fichier de configuration spécifique dans ce répertoire pour en interdire l'accès.

En faisant pointer Apache directement sur le répertoire `/web` les fichiers du répertoire `/cms` sont automatiquement inaccessibles.

Si vous déployez beCms sur un serveur personnel (ou uniquement sur votre poste), vous avez tout intérêt à faire pointer Apache sur le répertoire `/web` :

```
01 <VirtualHost *:80>
02     # activation de la réécriture d'URLs
03     RewriteEngine on
04
05     # déclaration nom de domaine
06     ServerName www.mon-becms.com
07
08     # répertoire correspondant : adaptez-le à votre répertoire
09     DocumentRoot /var/www/becms/web
10
```

```
11  # fichier PHP qui doit être lancé par le serveur
12  DirectoryIndex index.php
13
14  # fichier de configuration Apache utilisé dans le répertoire
15  AccessFileName .htaccess
16
17  # autorisations particulières pour le répertoire du CMS
18  # adaptez-le à votre répertoire
19  <Directory /etc/var/becms/web>
20      AllowOverride All
21      Allow from All
22  </Directory>
23 </VirtualHost>
```

Par contre si vous déployez beCms sur votre poste pour avoir une installation proche de ce que vous aurez chez un hébergeur mutualisé, il vaut mieux laisser la configuration vers le répertoire racine : il est en effet peu probable que votre hébergeur permette de pointer directement sur le répertoire /web.

4.2.4 Test

À ce stade, beCms peut déjà être lancé. L'URL <http://www.mon-becms.com> doit afficher la page d'accueil par défaut de beCms.

Cependant le serveur n'est pas encore sécurisé et n'est pas non plus configuré pour l'utilisation de la réécriture d'URLs.

Ces dernières étapes de configuration vont se faire par les fichiers **.htaccess** : des fichiers de configuration d'Apache au niveau de votre site web.

4.3 Principe des fichiers .htaccess

4.3.1 Introduction

Le serveur web Apache permet l'utilisation de fichiers de configuration pour un répertoire web et ses sous-répertoires.

Ces fichiers, nommés **.htaccess**, peuvent contenir des règles de configuration. Les règles d'un fichier **.htaccess** fonctionnent pour le répertoire dans lequel est placé le fichier ainsi que pour les sous-répertoires.

Ces règles permettent par exemple d'interdire l'accès des internautes à un répertoire, de protéger l'accès par un mot de passe ou encore de définir des adresses web élégantes.

Un fichier **.htaccess** est un fichier texte : vous pouvez l'éditer avec un éditeur de texte (type Bloc-notes, SciTE ¹).

Les lignes qui commencent par le caractère dièse (#) sont des commentaires.

Vous pouvez placer d'autres fichiers **.htaccess** dans les sous-répertoires. Les règles, ou **directives**, des fichiers **.htaccess** des sous-répertoires se cumulent avec celles des répertoires parents.

Si une **directive** d'un fichier **.htaccess** d'un sous-répertoire correspond à celle d'un répertoire parent, elle annule ou complète la **directive** parente.



Si l'accès à une page de votre site donne une réponse du type « Erreur 500 », c'est que vous avez utilisé une **directive** (règle de configuration) qui n'est pas disponible.

4.4 Création du fichier **.htaccess**

Créez un fichier **.htaccess** dans le répertoire sur lequel pointe Apache (en fonction de la configuration de votre fichier `mon-becms.conf`).



Windows ne permet pas de créer directement un fichier **.htaccess**. Pour le créer, ouvrez votre éditeur (par exemple SciTE) puis passez par l'éditeur pour enregistrer le fichier sous le nom **.htaccess**.

Ouvrez ce fichier : les parties qui suivent indiquent ce qu'il convient de mettre dans ce fichier.

1. <http://www.scintilla.org/SciTE.html>, éditeur avec coloration syntaxique libre et multi-plateformes.

4.5 Configuration de base

4.5.1 Bloquer le listage des répertoires

Par défaut Apache affiche la liste des fichiers d'un répertoire appelé par un internaute.

Par exemple les fichiers javascripts se trouvent dans le répertoire `/web/js`, ce répertoire peut être appelé par l'URL `http://www.mon-becms.com/web/js`.

Pour bloquer le listage des répertoires, vous pouvez ajouter le code suivant en début de fichier **.htaccess** :

```
1 # bloquer le listage des répertoires
2 Options -Indexes
```

4.5.2 Réécriture d'URLs

La réécriture d'URLs permet d'obtenir pour votre site web des adresses web élégantes. Avec un outil comme beCms les adresses web par défaut sont de la forme : `http://www.mon-becms.com/index.php`.

La réécriture d'URLs permet de remplacer la forme d'adresse par défaut par une forme plus élégante : `http://www.mon-becms.com/ma-page`.

L'option de réécriture d'URL a été activée dans la configuration de l'hôte virtuel. Il faut maintenant indiquer à Apache, via le fichier **.htaccess**, les règles de réécriture.

```
01 # si le module de réécriture est disponible
02 <IfModule mod_rewrite.c>
03 # activer le module
04 RewriteEngine On
05
06 # si le fichier demandé existe
07 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f
08 # autoriser l'accès aux fichiers du répertoire /web,
09 # uniquement si la configuration de l'hôte virtuel pointe sur la racine!!!
10 RewriteCond $0 (.*?web/././+)$
11 # on retourne le fichier à l'internaute
12 RewriteRule ^(.*)$ $1 [QSA,L]
13
14 # si la demande porte sur le fichier "install.php", on l'exécute
15 RewriteCond %{REQUEST_URI} ~/install.php$
16 RewriteRule install.php$ install.php [QSA,L]
17
18 # pour les autres fichiers on appelle le fichier "index.php"
```

```
19 RewriteRule ^(.*)$ index.php [QSA,L]
20 </IfModule>
```

4.5.3 Protection du répertoire /cms

Comme indiqué dans les « Notes sur l'arborescence de fichiers », les fichiers du répertoire /cms doivent être accessibles à l'outil mais pas aux internautes. Pour ce faire vous pouvez créer un fichier **.htaccess** dans le répertoire /cms, avec le contenu suivant :

```
1 # aucun accès au répertoire ni aux fichiers
2 order deny, allow
3 deny from all
```



Ne pas créer ce fichier peut être une faille de sécurité pour votre site si la configuration de l'hôte virtuel pointe sur la racine.

4.6 Aller plus loin dans la configuration

Votre serveur est configuré. Il est néanmoins possible d'améliorer la configuration serveur de votre site afin de mieux le protéger ou de mieux l'optimiser. C'est ce qui est expliqué dans la suite de cette documentation.

4.6.1 Options de configuration de PHP

Il est possible de configurer un minimum PHP² via le fichier **.htaccess**. Il peut être intéressant de forcer la configuration de quelques options.

La syntaxe générale de ce type de configuration est :

2. <http://php.net/manual/fr/configuration.changes.php>

```
1 SetEnv nom_option_php valeur
```

4.6.1.1 Register globals

L'option de configuration « register_globals » de PHP est un héritage de PHP 4 devenue obsolète depuis la version 5.3.0 et supprimée depuis la version 5.4.0.

Cette option permettait de transformer automatiquement en variables le contenu des formulaires, cookies... Ce n'était pas une bonne pratique, et cela pouvait même s'avérer dangereux. Aussi il est recommandé de désactiver cette option :

```
1 SetEnv register_globals 0
```

4.6.1.2 Magic quotes

L'option « magic_quotes » permettait d'échapper automatiquement les caractères guillemets simples ('), guillemets doubles ("), antislash (\) et NUL avec un antislash.

Cette fonctionnalité s'est avérée finalement peu pratique, entre autres à cause des différences de configuration entre serveurs, et est devenue obsolète depuis la version 5.3.0 et supprimée depuis la version 5.4.0.

Il est donc recommandé de désactiver cette option :

```
1 SetEnv magic_quotes 0
```

4.6.1.3 Short open tag

L'option « short_open_tag » permet d'ouvrir une balise PHP en utilisant le code `<?>` au lieu de la version complète `<?php>`.

Cette option ne rend pas possible l'utilisation de code PHP dans un fichier XML, car le code `<?xml>` d'en-tête XML sera interprété comme une ouverture de code PHP.

Il est donc recommandé de désactiver cette option et de toujours commencer le code PHP par la version longue.

Pour désactiver cette option :

```
1 SetEnv short_open_tag 0
```

4.6.2 Améliorer la sécurité en utilisant la réécriture d'URLs

Il est possible d'améliorer la sécurité de votre installation en utilisant la réécriture d'URLs. On verra qu'il est également possible de l'utiliser pour améliorer l'accessibilité du site.

Cette partie présente donc différents points, un extrait de fichier **.htaccess** pour la configuration du `mod_rewrite` est donné en exemple en fin de partie.

La configuration donnée dans les sous-parties doit être placée dans le fichier **.htaccess** entre `<IfModule mod_rewrite.c>` et `</IfModule>`.

4.6.2.1 Autoriser l'accès aux fichiers du répertoire /web

Le répertoire `/web` du CMS contient les fichiers ressources (javascripts, css, images...) qui doivent être accessibles aux internautes. Il est possible de n'autoriser l'accès qu'à ces fichiers et de tester leur existence :

```
1 # si le fichier demandé existe
2 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f
3
4 # si le fichier est dans le répertoire /web,
5 # uniquement si la configuration de l'hôte virtuel pointe sur la racine!!!
6 RewriteCond $0 (.*(?:web/./+.)$)
7
8 # on retourne le fichier à l'internaute
9 RewriteRule ^(.*)$ $1 [QSA,L]
```

Si le fichier demandé n'existe pas, l'internaute sera redirigé vers la page « 404 » du CMS.

4.6.2.2 Autoriser l'accès à un fichier situé à la racine du CMS

Il peut être intéressant d'autoriser l'accès à un ou plusieurs fichiers situés à la racine du CMS et qui ne sont pas gérés par le CMS, comme par exemple un fichier `favicon.ico`³, qui

3. <http://fr.wikipedia.org/wiki/Favicon>

permet à Internet Explorer d'afficher l'icone d'un site, ou encore le fichier `robots.txt`⁴ qui permet de donner des instructions de parcours du site pour les moteurs de recherche.

La syntaxe générale est :

```

1  # si la demande porte sur le fichier "nom_du_fichier"
2  RewriteCond %{REQUEST_URI} ~/nom_du_fichier$
3  # si le fichier demandé existe
4  RewriteCond %{DOCUMENT_ROOT}/nom_du_fichier -f
5  # on retourne le fichier
6  RewriteRule ^(.*)$ $1 [QSA,L]
```

Exemple pour les fichiers `favicon.ico` et `robots.txt` :

```

01 # si la demande porte sur le fichier favicon.ico
02 RewriteCond %{REQUEST_URI} ~/favicon.ico$
03 # si le fichier favicon.ico existe
04 RewriteCond %{DOCUMENT_ROOT}/favico.ico -f
05 # on retourne le fichier
06 RewriteRule ^(.*)$ $1 [QSA,L]
07
08 # si la demande porte sur le fichier robots.txt
09 RewriteCond %{REQUEST_URI} ~/robots.txt$
10 # si le fichier robots.txt existe
11 RewriteCond %{DOCUMENT_ROOT}/robots.txt -f
12 # on retourne le fichier
13 RewriteRule ^(.*)$ $1 [QSA,L]
```

De la même façon il est possible d'autoriser, ou non, l'accès au fichier d'installaton du CMS `install.php` :

```

1  # si la demande porte sur le fichier install.php
2  RewriteCond %{REQUEST_URI} ~/install.php$
3  # si le fichier install.php existe
4  RewriteCond %{DOCUMENT_ROOT}/install.php -f
5  # on retourne le fichier
6  RewriteRule ^(.*)$ $1 [QSA,L]
```

4.6.2.3 Améliorer les performances du cache

beCms permet de mettre des pages en cache. Ce système permet de réduire le nombre de calculs nécessaires pour l'affichage d'une page.

Le principe général du cache est le suivant :

4. <http://fr.wikipedia.org/wiki/Robots.txt>

- l'utilisateur du CMS paramètre une durée de cache pour une page donnée depuis l'interface d'administration ;
- lorsque la page est demandée une première fois par un internaute, la page est enregistrée dans un sous-répertoire de `/web/cache`, le fichier enregistré contient le code HTML de la page ainsi qu'une instruction PHP pour gérer la date de péremption du fichier caché ;
- lorsque la page est demandée à nouveau par des internautes, le CMS regarde si le fichier cache existe. Si c'est la cas, c'est ce fichier qui retourné.

Il est possible d'améliorer les performances du cache en appelant directement le fichier cache, et donc ne plus passer par le CMS :

```
01  # si la demande de l'internaute est du type GET
02  RewriteCond %{REQUEST_METHOD} GET
03  # si la requête n'est pas du type HTTPS
04  RewriteCond %{HTTPS} off
05  # si le fichier /web/cache/http/nom_de_domaine/requete.cache.php existe
06  RewriteCond %{DOCUMENT_ROOT}/web/cache/http/%{HTTP_HOST}/%{REQUEST_URI}.cache.php
-f
07  # on exécute le fichier cache
08  RewriteRule ^(.*) web/cache/http/%{HTTP_HOST}/$1.cache.php [L]
09
10  # idem que précédent pour le cas d'une requete HTTPS
11  # vous pouvez ignorer cette configuration si vous
12  # n'utilisez jamais ce mode
13  RewriteCond %{REQUEST_METHOD} GET
14  RewriteCond %{HTTPS} on
15  RewriteCond %{DOCUMENT_ROOT}/web/cache/https/%{HTTP_HOST}/%{REQUEST_URI}.cache.php
-f
16  RewriteRule ^(.*) web/cache/https/%{HTTP_HOST}/$1.cache.php [L]
```



Si la configuration de l'hôte virtuel pointe sur le répertoire `/web`, il faut supprimer `/web` des chemins.

4.6.2.4 Exemple de configuration du mod_rewrite



Si la configuration de l'hôte virtuel pointe sur le répertoire /web, il faut supprimer /web des chemins.

Voici un exemple complet de configuration du mod_rewrite, tel que vous pourriez l'utiliser (avec vos paramètres de site) :

```

01 # si le module de réécriture est disponible
02 <IfModule mod_rewrite.c>
03 # activer le module
04 RewriteEngine On
05
06 # retourner les fichiers du répertoire /web
07 # si le fichier demandé existe
08 RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f
09 # si le fichier est dans le répertoire /web,
10 # uniquement si la configuration de l'hôte virtuel pointe sur la racine!!!
11 RewriteCond $0 (.*?web/././+)$
12 # on retourne le fichier à l'internaute
13 RewriteRule ^(.*)$ $1 [QSA,L]
14
15 # retourner le fichier /favicon.ico
16 # si la demande porte sur le fichier favicon.ico
17 RewriteCond %{REQUEST_URI} ~/favicon.ico$
18 # si le fichier favicon.ico existe
19 RewriteCond %{DOCUMENT_ROOT}/favico.ico -f
20 # on retourne le fichier
21 RewriteRule ^(.*)$ $1 [QSA,L]
22
23 # retourner le fichier /robots.txt
24 # si la demande porte sur le fichier robots.txt
25 RewriteCond %{REQUEST_URI} ~/robots.txt$
26 # si le fichier robots.txt existe
27 RewriteCond %{DOCUMENT_ROOT}/robots.txt -f
28 # on retourne le fichier
29 RewriteRule ^(.*)$ $1 [QSA,L]
30
31 # retourner le fichier /install.php
32 # si la demande porte sur le fichier install.php
33 RewriteCond %{REQUEST_URI} ~/install.php$
34 # si le fichier install.php existe
35 RewriteCond %{DOCUMENT_ROOT}/install.php -f
36 # on retourne le fichier
37 RewriteRule ^(.*)$ $1 [QSA,L]
38
39 # optimiser les fichiers mis en cache

```

```
40 # si la demande de l'internaute est du type GET
41 RewriteCond %{REQUEST_METHOD} GET
42 # si le fichier est en cache
43 RewriteCond %{DOCUMENT_ROOT}/web/cache/{HTTP_HOST}/{REQUEST_URI}.cache.php -f
44 # on exécute le fichier cache
45 RewriteRule ^(.*) web/cache/{HTTP_HOST}/$1.cache.php [L]
46
47 # si aucune des règles précédentes n'est valable pour la demande,
48 # on lance le fichier /index.php qui gère le CMS
49 RewriteRule ^(.*)$ index.php [QSA,L]
50 </IfModule>
```

4.6.3 Optimiser votre site web

D'autres modules d'Apache permettent d'optimiser votre site afin que son chargement soit plus rapide. Cette section donne le détail de quelques optimisations envisageables.



Dans l'installation d'Apache effectuée précédemment nous n'avons installé que le module `mod_rewrite` permettant la réécriture d'URLs. Reportez-vous à la partie **Ajouter le mod_rewrite à Apache** de votre système pour voir comment ajouter de nouveaux modules.

4.6.3.1 Activer la compression

Pour améliorer les performances de votre site et limiter la bande passante utilisée, il est possible de compresser des données envoyées au navigateur.

Il n'est pas nécessaire de compresser toutes les données : les fichiers images (gif, jpeg ou png) par exemple sont des formats nativement compressés.

Pour cette configuration, vous devez activer les modules :

```
- mod_deflate;
- mod_filter;
- mod_ext_filter;
- mod_include;
```

```

- mod_charset_lite;

- mod_header.

```

Ensuite vous pouvez placer la configuration suivante en début de votre fichier **.htaccess** :

```

01 # activer la compression
02 SetOutputFilter DEFLATE
03
04 # certains navigateurs ont des problèmes avec la compression, c'est pourquoi nous
faisons des exceptions
05 BrowserMatch ^Mozilla/4 gzip-only-text/html
06 BrowserMatch ^Mozilla/4\.0[678] no-gzip
07 BrowserMatch \bMSIE !no-gzip !gzip-only-text/html
08
09 # les images n'ont pas besoin d'être compressées puisqu'elles le sont déjà (jpg,
gif...)
10 SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
11 # pour les proxy
12 Header append Vary User-Agent env=!dont-vary

```

4.6.3.2 Mise en cache navigateur et dates d'expirations

Les navigateurs peuvent gérer une mise en cache interne. Cela permet de ne pas demander systématiquement une information au serveur web si celle-ci a déjà été demandée.

Pour un fonctionnement optimal du cache il faut pouvoir indiquer au navigateur :

- qu'un fichier doit être mis en cache ;
- une date d'expiration pour le cache.

Pour cette configuration, vous devez activer les modules :

```

- mod_expires;

- mod_headers.

```

Voici un exemple de configuration permettant d'indiquer des délais d'expiration et la mise en cache des fichiers :

```

01 # si le mode "expires" d'Apache est disponible
02 <IfModule mod_expires.c>
03 # activation du module
04 ExpiresActive On

```

```

05
06  # les fichiers images ont un délai d'expiration de 2592000 secondes : 1 mois
07  ExpiresByType image/jpg "access plus 2592000 seconds"
08  ExpiresByType image/jpeg "access plus 2592000 seconds"
09  ExpiresByType image/png "access plus 2592000 seconds"
10  ExpiresByType image/gif "access plus 2592000 seconds"
11  AddType image/x-icon .ico
12  ExpiresByType image/ico "access plus 2592000 seconds"
13  ExpiresByType image/icon "access plus 2592000 seconds"
14  ExpiresByType image/x-icon "access plus 2592000 seconds"
15
16  # les fichiers css et javascripts ont un délai d'expiration de 1 semaine
17  ExpiresByType text/css "access plus 1 week"
18  ExpiresByType text/javascript "access plus 1 week"
19  ExpiresByType application/javascript "access plus 1 week"
20  ExpiresByType application/x-javascript "access plus 1 week"
21 </IfModule>
22
23  # si le module "headers" d'Apache est disponible
24 <IfModule mod_headers.c>
25  # les fichiers images ont une mise en cache de 1 mois
26  <FilesMatch "\.(ico|jpe?g|png|gif|gz)$">
27    Header set Cache-Control "max-age=2592000, public, proxy-revalidate"
28  </FilesMatch>
29
30  # les fichiers css et javascripts on une mise en cache de 1 semaine
31  <FilesMatch "\.(css)$">
32    Header set Cache-Control "max-age=604800, public, proxy-revalidate"
33  </FilesMatch>
34  <FilesMatch ".*css.*\.php$">
35    Header set Cache-Control "max-age=604800, public, proxy-revalidate"
36  </FilesMatch>
37  <FilesMatch "\.(js)$">
38    Header set Cache-Control "max-age=604800, private, proxy-revalidate"
39  </FilesMatch>
40  <FilesMatch ".*js.*\.php$">
41    Header set Cache-Control "max-age=604800, private, proxy-revalidate"
42  </FilesMatch>
43 </IfModule>

```

4.6.4 Exemple complet de fichier .htaccess



Si la configuration de l'hôte virtuel pointe sur le répertoire /web, il faut supprimer /web des chemins.

L'exemple suivant montre un fichier **htaccess** complet contenant l'ensemble des instructions décrites dans le document :

```

001 # bloquer le listage des répertoires
002 Options -Indexes
003
004 # options PHP
005 SetEnv register_gobal 0
006 SetEnv magic_quotes 0
007 SetEnv short_open_tag 0
008
009 # activer la compression
010 SetOutputFilter DEFLATE
011
012 # certains navigateurs ont des problèmes avec la compression, c'est pourquoi nous
    faisons des exceptions
013 BrowserMatch ^Mozilla/4 gzip-only-text/html
014 BrowserMatch ^Mozilla/4\.0[678] no-gzip
015 BrowserMatch \bMSIE !no-gzip!gzip-only-text/html
016
017 # les images n'ont pas besoin d'être compressées puisqu'elles le sont déjà (jpg,
    gif...)
018 SetEnvIfNoCase Request_URI \.(?:gif|jpe?g|png)$ no-gzip dont-vary
019 # Pour les proxy
020 Header append Vary User-Agent env=!dont-vary
021
022 # si le mode "expires" d'Apache est disponible
023 <IfModule mod_expires.c>
024     # activation du module
025     ExpiresActive On
026
027     # les fichiers images ont un délai d'expiration de 2592000 secondes : 1 mois
028     ExpiresByType image/jpg "access plus 2592000 seconds"
029     ExpiresByType image/jpeg "access plus 2592000 seconds"
030     ExpiresByType image/png "access plus 2592000 seconds"
031     ExpiresByType image/gif "access plus 2592000 seconds"
032     AddType image/x-icon .ico
033     ExpiresByType image/ico "access plus 2592000 seconds"
034     ExpiresByType image/icon "access plus 2592000 seconds"
035     ExpiresByType image/x-icon "access plus 2592000 seconds"
036
037     # les fichiers css et javascripts ont un délai d'expiration de 1 semaine
038     ExpiresByType text/css "access plus 1 week"
039     ExpiresByType text/javascript "access plus 1 week"
040     ExpiresByType application/javascript "access plus 1 week"
041     ExpiresByType application/x-javascript "access plus 1 week"
042 </IfModule>
043
044 # si le module "headers" d'Apache est disponible
045 <IfModule mod_headers.c>
046     # les fichiers images ont une mise en cache de 1 mois
047     <FilesMatch "\.(ico|jpe?g|png|gif|gz)$">
048         Header set Cache-Control "max-age=2592000, public, proxy-revalidate"
049     </FilesMatch>
050

```



```
051  # les fichiers css et javascripts on une mise en cache de 1 semaine
052  <FilesMatch "\.(css)$">
053      Header set Cache-Control "max-age=604800, public, proxy-revalidate"
054  </FilesMatch>
055  <FilesMatch ".*css.*\.(php)$">
056      Header set Cache-Control "max-age=604800, public, proxy-revalidate"
057  </FilesMatch>
058  <FilesMatch "\.(js)$">
059      Header set Cache-Control "max-age=604800, private, proxy-revalidate"
060  </FilesMatch>
061  <FilesMatch ".*js.*\.(php)$">
062      Header set Cache-Control "max-age=604800, private, proxy-revalidate"
063  </FilesMatch>
064  </IfModule>
065
066  # si le module de réécriture est disponible
067  <IfModule mod_rewrite.c>
068      # activer le module
069      RewriteEngine On
070
071      # si le fichier demandé existe
072      RewriteCond %{DOCUMENT_ROOT}%{REQUEST_URI} -f
073      # si le fichier est dans le répertoire /web,
074      # uniquement si la configuration de l'hôte virtuel pointe sur la racine!!!
075      RewriteCond $0 (.*(?:web/./+).+)$
076      # on retourne le fichier à l'internaute
077      RewriteRule ^(.*)$ $1 [QSA,L]
078
079      # retourner le fichier /favicon.ico
080      # si la demande porte sur le fichier favicon.ico
081      RewriteCond %{REQUEST_URI} ^/favicon.ico$
082      # si le fichier favicon.ico existe
083      RewriteCond %{DOCUMENT_ROOT}/favico.ico -f
084      # on retourne le fichier
085      RewriteRule ^(.*)$ $1 [QSA,L]
086
087      # retourner le fichier /robots.txt
088      # si la demande porte sur le fichier robots.txt
089      RewriteCond %{REQUEST_URI} ^/robots.txt$
090      # si le fichier robots.txt existe
091      RewriteCond %{DOCUMENT_ROOT}/robots.txt -f
092      # on retourne le fichier
093      RewriteRule ^(.*)$ $1 [QSA,L]
094
095      # retourner le fichier /install.php
096      # si la demande porte sur le fichier install.php
097      RewriteCond %{REQUEST_URI} ^/install.php$
098      # si le fichier install.php existe
099      RewriteCond %{DOCUMENT_ROOT}/install.php -f
100      # on retourne le fichier
101      RewriteRule ^(.*)$ $1 [QSA,L]
102
103      # optimiser les fichiers mis en cache
104      # si la demande de l'internaute est du type GET
105      RewriteCond %{REQUEST_METHOD} GET
106      # si le fichier est en cache
```

```
107 RewriteCond %{DOCUMENT_ROOT}/web/cache/%{HTTP_HOST}/%{REQUEST_URI}.cache.php -f
108 # on exécute le fichier cache
109 RewriteRule ^(.*) web/cache/%{HTTP_HOST}/$1.cache.php [L]
110
111 # si aucune des règles précédentes n'est valable pour la demande,
112 # on lance le fichier /index.php qui gère le CMS
113 RewriteRule ^(.*)$ index.php [QSA,L]
114 </IfModule>
```

Configurer le CMS

5.1 Introduction

Votre serveur est maintenant configuré et beCms est en place... Il reste néanmoins à le paramétrer.

La configuration de base est simple et s'effectue par une interface dédiée.

5.2 Accès à l'interface de configuration

La page de configuration de beCms est accessible par l'URL <http://www.mon-becms.com/install.php>.

5.3 Présentation de l'interface de configuration

L'interface de configuration correspond à l'image ci-dessous :

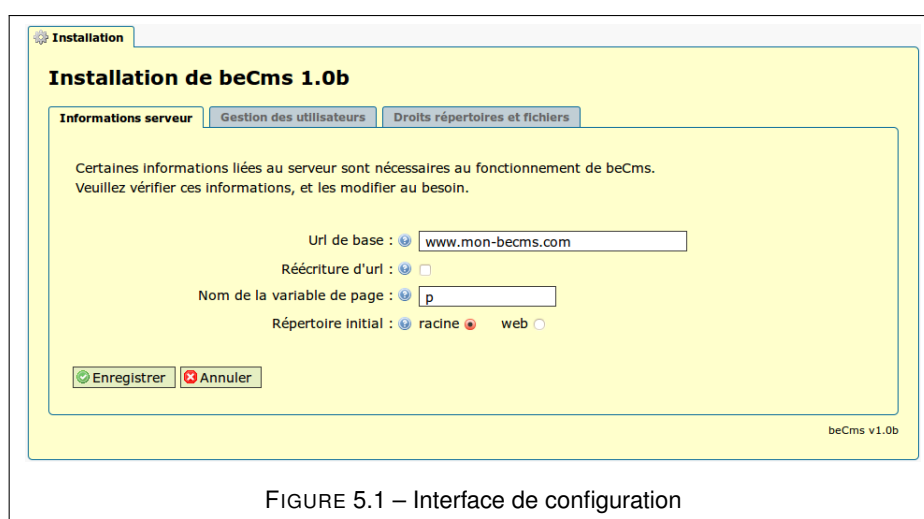


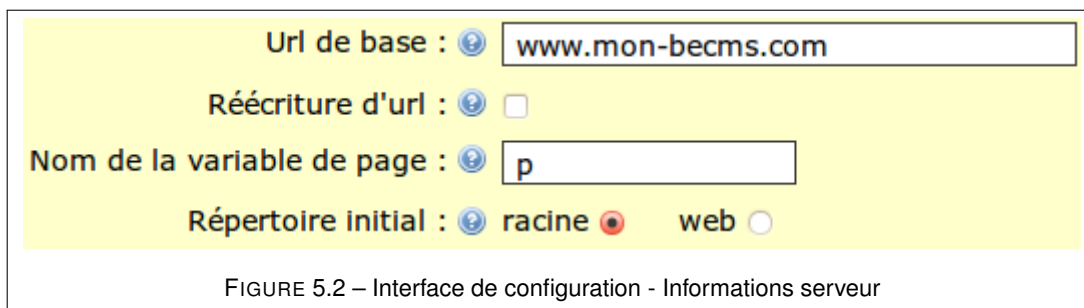
FIGURE 5.1 – Interface de configuration

Elle se décompose en trois onglets :

- **Informations serveur** : permet de paramétrer les informations nécessaires au bon fonctionnement du CMS sur le serveur ;
- **Gestion des utilisateurs** : permet de définir deux utilisateurs initiaux pour le CMS ;
- **Droits répertoires et fichiers** : permet de vérifier que les répertoires et fichiers possèdent les bons droits en lecture et écriture.

5.3.1 Onglet « Informations serveur »

Cet onglet permet de paramétrer les informations de fonctionnement de beCms sur votre serveur.



Url de base :

Réécriture d'url :

Nom de la variable de page :

Répertoire initial : racine web

FIGURE 5.2 – Interface de configuration - Informations serveur

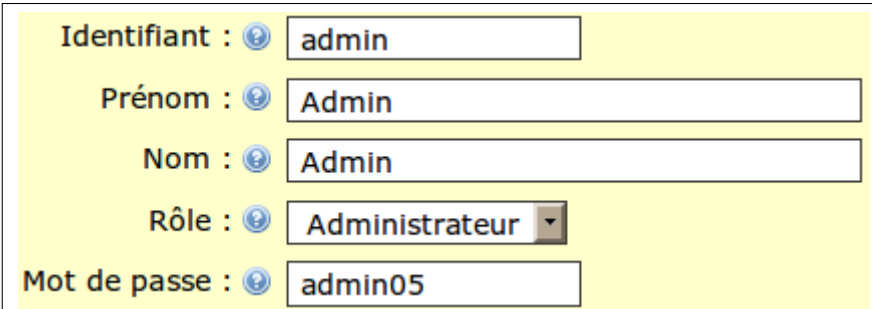
Il se compose de quatre champs pré-remplis :

- **Url de base** : c'est l'URL de votre site, c'est à dire son adresse web. Vous devez y indiquer l'adresse du site sans le protocole (http ou https) ;
- **Réécriture d'url** : indique si la réécriture d'URLs est active. La réécriture d'URLs permet d'obtenir pour votre site web des adresses web élégantes. En suivant cette documentation nous avons activé la réécriture et configuré Apache pour que le site puisse l'utiliser. Vous pouvez donc cocher cette option ;
- **Nom de la variable de page** : permet de désigner le nom de la variable de page utilisée par beCms dans le cas où la réécriture d'URL n'est pas active ;
- **Répertoire initial** : indique si le fichier `index.php` utilisé est celui du répertoire `racine` de votre site ou celui du répertoire `/web`. Reportez-vous à la section « Notes sur l'arborescence de fichiers » pour plus d'informations.

5.3.2 Onglet « Gestion des utilisateurs »

Cet onglet permet de définir deux utilisateurs initiaux pour le CMS :

- un **administrateur**, qui peut gérer le site, la configuration du CMS et les utilisateurs ;
- un **utilisateur** qui peut gérer le site.



The screenshot shows a configuration form with five fields, each with a question mark icon to its left. The fields are: 'Identifiant' with the value 'admin', 'Prénom' with 'Admin', 'Nom' with 'Admin', 'Rôle' with a dropdown menu showing 'Administrateur', and 'Mot de passe' with 'admin05'. The entire form is set against a light yellow background.

FIGURE 5.3 – Interface de configuration - Configuration d'un utilisateur

La configuration d'un utilisateur se fait via cinq champs :

- **Identifiant** : identifiant de connexion de l'utilisateur utilisé pour entrer dans l'interface de gestion de beCms. **L'identifiant doit contenir au moins 4 caractères.** ;
- **Prénom** : prénom de l'utilisateur ;
- **Nom** : nom de l'utilisateur ;
- **Rôle** : indique si l'utilisateur est du type **administrateur** (droits de gestion du site, de la configuration du CMS et des utilisateurs) ou simple **utilisateur** (droits de gestion du site) ;
- **Mot de passe** : mot de passe utilisé pour entrer dans l'interface de gestion de beCms. **Le mot de passe doit contenir au moins 6 caractères.**



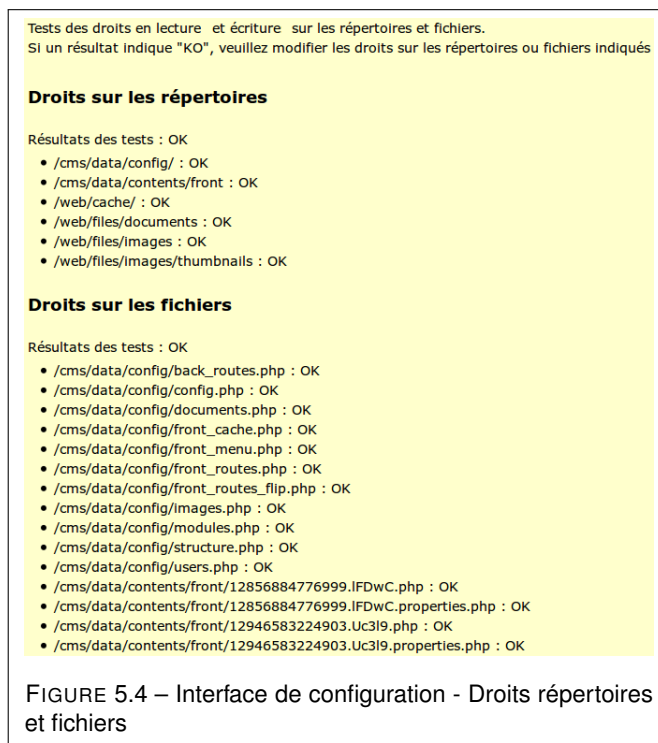
Il est obligatoire d'avoir au moins un **administrateur**.



Il est important de ne pas laisser les identifiants et mots de passe par défaut !
Les informations par défaut sont les mêmes pour toutes les installations de beCms : les laisser est donc une faille de sécurité !

5.3.3 Onglet « Droits répertoires et fichiers »

Cet onglet permet de vérifier que les répertoires et fichiers possèdent les bons droits en lecture et écriture.



Sur un serveur mutualisé l'hébergeur fixe généralement les droits lors du transfert FTP. Vous ne devriez donc pas avoir de problème et tous les résultats de tests doivent indiquer « OK ».

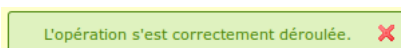
En cas de test négatif (« KO ») vous pouvez :

- soit essayer de modifier les droits via votre logiciel client FTP ;
- soit contacter votre hébergeur pour régler le problème.

5.3.4 Fin de l'installation

Lorsque la configuration est terminée, vous pouvez la valider en cliquant sur le bouton « Enregistrer » en bas de page.

Une boîte de dialogue doit indiquer que l'opération s'est correctement déroulée :



Un texte en début de page confirme l'opération et donne le lien pour accéder à l'interface de connexion de beCms.

A screenshot of a confirmation message in French, displayed in a yellow-highlighted box. The text reads: "L'enregistrement de la configuration s'est bien déroulée. Vous pouvez modifier à nouveau les options ou tester la connexion à votre interface d'administration : <http://www.mon-becms.com/admin/login>." The URL is underlined and blue, indicating it is a hyperlink.

L'enregistrement de la configuration s'est bien déroulée. Vous pouvez modifier à nouveau les options ou tester la connexion à votre interface d'administration : <http://www.mon-becms.com/admin/login>.

FIGURE 5.5 – Texte de confirmation

Vous pouvez maintenant vous connecter sur votre interface d'administration.

Bonne continuation avec beCms !

Lexique

CMS

Content Management System, ou **Système de Gestion de Contenu** en français. Logiciel permettant la création et la mise à jour d'un site web.

FTP

File Transfer Protocol, ou **Protocole de Tranfert de Fichiers** est un protocole de communication permettant de transférer des fichiers d'un ordinateur vers un autre.

PHP

PHP Hypertext Preprocessor. Langage de programmation principalement destiné au développement de sites web.

Serveur web

Le terme de seueur web représente deux choses : d'une part l'ordinateur qui est relié au réseau internet et transfère la demande de l'internaute à un logiciel, d'autre part le logiciel (serveur HTTP) qui traite effectivement la demande.

Licence et crédits

7.1 Licence

La présente documentation est placée sous licence Creative Commons (CC BY - NC - SA 2.0)¹.

beCms est placé sous licence publique générale GNU version 3 (GNU GPLv3)².

7.2 Crédits

beCms utilise :

- la bibliothèque javascript **jQuery**³ sous double licence MIT / GNU GPLv2⁴ ;
- l'éditeur javascript WYSIWYG **weEd** sous licence GNU GPLv3 ;
- l'éditeur javascript syntaxique de code source **CodeMirror**⁵ sous licence type MIT⁶ ;
- le jeu d'icônes **Silk**⁷ sous double licence Creative Commons Attribution 2.5 / Creative Commons Attribution 3.

1. <http://creativecommons.org/>

2. <http://www.gnu.org/licenses/gpl.html>

3. <http://jquery.com/>

4. <http://jquery.org/license/>

5. <http://codemirror.net/>

6. <http://codemirror.net/LICENSE>

7. <http://www.famfamfam.com/lab/icons/silk/>