

INSTALLATION ET CONFIGURATION D'UN SERVEUR WEB

(2ème édition)

Par **Maisse Sébastien**



Document en date du 05 décembre 2005

Sommaire

Concernant ce document :	3
Préambule :	4
Qu'est-ce qu'un serveur web ?.....	4
Téléchargement d'APACHE :.....	5
Installation d'APACHE :.....	5
Configuration d'APACHE :.....	14
Qu'est-ce que PHP ?.....	18
Téléchargement de PHP :.....	18
Installation de PHP :.....	19
Qu'est-ce qu'un serveur mysql ?.....	24
Téléchargement de MySQL :.....	25
Installation de MySQL :.....	26
Configuration de MySQL :.....	31
Quelques interfaces graphiques :.....	41
Ressources :.....	47
Quelques url's :.....	47
Quelques livres :.....	47
Conclusion :.....	48
Merci !.....	48

Concernant ce document :

Ce document a pour vocation de vous aider à mettre en place un serveur web sur une plateforme Microsoft Windows (version XP, 2000, 2003). Cette documentation est une mise à jour d'un précédent tutoriel que j'avais fait il y a 2 ans (en 2003 pour la première version du présent document) de cela.

Le précédent tutoriel est toujours en ligne et rencontre un vif succès l'heure d'aujourd'hui, d'où ma décision de réaliser une deuxième édition prenant en compte les dernières versions des logiciels qui vont vous permettre de rendre disponible votre futur serveur internet.

Les logiciels que nous allons mettre en application sont tous disponibles gratuitement ou en open source. Ils sont les concurrents des logiciels commerciaux qui sont vendus par de grandes et puissantes sociétés informatiques avides de monopole pour les solutions qu'ils proposent.

Vous allez être accompagnés tout au long de ce tutoriel dans l'installation mais aussi la configuration de chaque logiciel ce qui vous permettra au final d'avoir un serveur web (apache) supportant le langage PHP et pourvu d'un serveur de données (mysql).

Si lors de l'exploitation de ce document vous rencontrez des erreurs, ou des incompréhensions vous pourrez obtenir de l'aide en fin de ce document.

Je vous souhaite à présent une bonne lecture ainsi qu'une bonne installation/configuration de votre futur serveur.

Préambule :

Bienvenue dans ce document qui a pour but de vous faire découvrir l'installation du serveur web Apache avec le support PHP ainsi que du serveur de données MySQL dans un environnement Windows (XP, 2000 ou 2003).

Apache est le travail de la **Apache Software Foundation** qui réunit d'autres projets.

Mais avant tout, une petite explication (provenant du site wikipedia.org) de ce qu'est un serveur web et à quoi il est destiné.

Qu'est-ce qu'un serveur web ?

Un serveur HTTP ou démon HTTP ou HTTPd (HTTP daemon) ou (moins précisément) serveur Web, est un logiciel servant des requêtes respectant le protocole de communication client-serveur HyperText Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.

Un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur Web. Le terme « serveur Web » peut aussi désigner le serveur HTTP (le logiciel) lui-même. Les deux termes sont utilisés pour le logiciel car le protocole HTTP a été développé pour le Web et les pages Web sont en pratique toujours servies avec ce protocole. D'autres ressources du Web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servies avec d'autres protocoles.

Les serveurs HTTP les plus utilisés sont :

- Apache HTTP Server de la Apache Software Foundation, successeur du NCSA httpd
- Internet Information Services de Microsoft (IIS)
- Sun ONE de Sun Microsystems (anciennement iPlanet de Netscape Communications Corporation)
- Le serveur Web Zeus de Zeus Technology

Le plus populaire est Apache HTTP Server qui sert environ 69% des sites Web en octobre 2005 selon Netcraft.

Historiquement, d'autres serveurs HTTP importants furent CERN httpd, développé par les inventeurs du Web, abandonné le 15 juillet 1996 et NCSA HTTPd, développé au NCSA en même temps que NCSA Mosaic, abandonné mi-1994.

Il existe aussi des serveurs HTTP qui sont des serveurs d'applications capables de faire serveur HTTP, comme Roxen ou Caudium.

Téléchargement d'APACHE :

Avant l'installation, nous devons tout d'abord récupérer l'archive contenant notre futur serveur web. Pour ce faire rendez-vous sur le site apache.org. De là, il faut récupérer la version 2.0.XX de Apache pour environnement Windows.

Dans le cas présent, nous allons baser notre installation sur Apache version **2.0.55**, version qui est actuellement diffusée à l'heure où j'écris ce document.

Nom du fichier téléchargé : **apache_2.0.55-win32-x86-no_ssl.msi**

Bien entendu, il va de soi que vous pouvez baser votre installation sur une version supérieure. Vous pouvez effectuer le téléchargement de l'application à cette url :

<http://httpd.apache.org/download.cgi>

Ceci fait, nous pouvons à présent installer Apache sur notre machine.

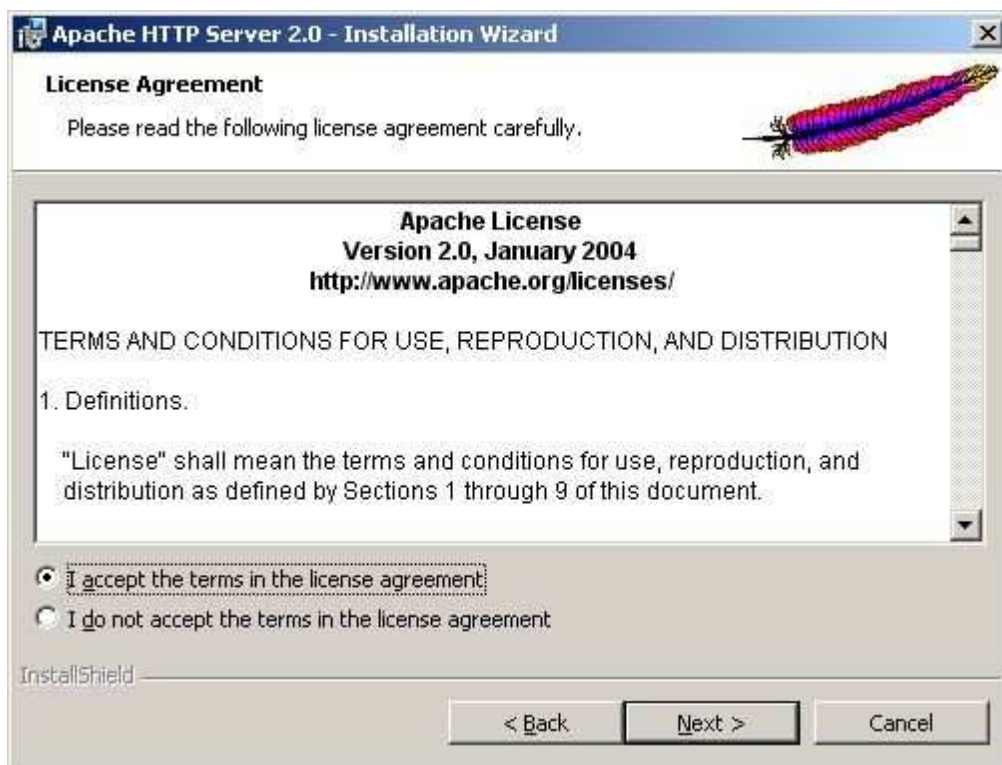
Installation d'APACHE :

Localisez l'archive que vous avez téléchargée, par défaut sur le bureau si vous avez téléchargé avec firefox sinon une petite recherche (sur votre disque dur) vous permettra de localiser le programme d'installation d'Apache.

*Un double-clic lancera le programme d'installation... On clique sur **Next**.*



Nous devons tout d'abord accepter la licence d'utilisation de Apache. La licence est similaire à une licence [GPL](#).

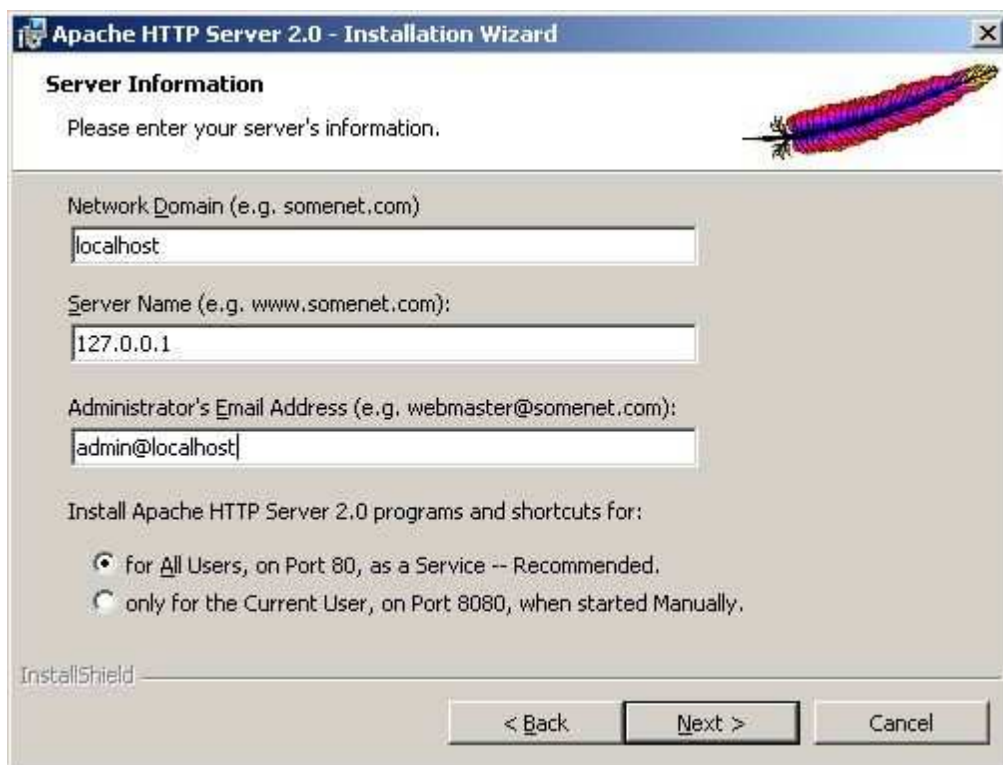


On coche **I accept the terms in the licence agreement** puis on clique sur Next. Ceci étant si vous n'acceptez pas les termes de la licence, l'installation s'achèvera là...

Une petite explication de ce qu'est Apache. On clique sur **Next**.



A cet écran, nous allons configurer le nom de domaine, le nom du serveur ainsi que l'adresse e-mail de l'administrateur du serveur.



Network Domain : Nom de domaine du serveur, Localhost correspondant à l'hôte local dans le cas présent.

Server Name : Nom du serveur, ici, nous allons mettre 127.0.0.1 qui correspond à l'ip interne de la machine.

Administrator's Email Address : l'adresse e-mail de l'administrateur en l'occurrence vous.

For All Users, on Port 80, as Service : cette option permet d'installer apache pour **tout les comptes utilisateurs** en utilisant le **port 80** (port utilisé par défaut pour un serveur web) **en tant que service** (le service est utilisé par Windows XP ou 2000).

only for the Current User, on Port 8080, when started Manually : cette option permet d'installer apache pour **l'utilisateur courant** (qui procède à l'installation) en utilisant le **port 8080** et avec un **démarrage manuel**. Cette dernière est recommandée si vous ne souhaitez pas mettre en production votre serveur web, mais que vous l'utilisez dans un objectif de développement web.

Dans le cas présent, je procède à une installation de type « tout utilisateurs, sur le port 80 en tant que service ».

On clique sur **Next** pour continuer l'installation...

On choisi le type d'installation pour notre serveur web :

- **Typical** : Installation typique (va installer les principales éléments pour l'utilisation du serveur) – recommandé si vous n'avez pas trop d'idée des composants dont vous avez besoin.
- **Custom** : Installation personnalisée (on défini les composants que l'on souhaite installer pour notre serveur)



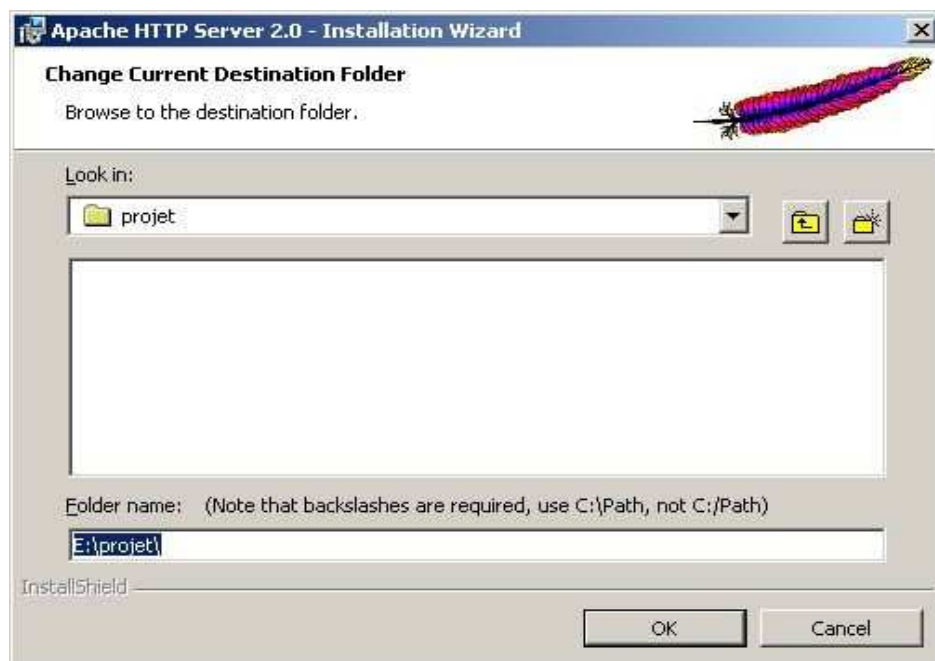
Dans, le cas présent, je choisis une installation de type **Custom**, et je clique sur **Next**.

Nous avons (ci-dessous) les principaux composants pour notre serveur web, à noter que tous ne sont pas utiles si l'on pense utiliser notre serveur de manière standard.

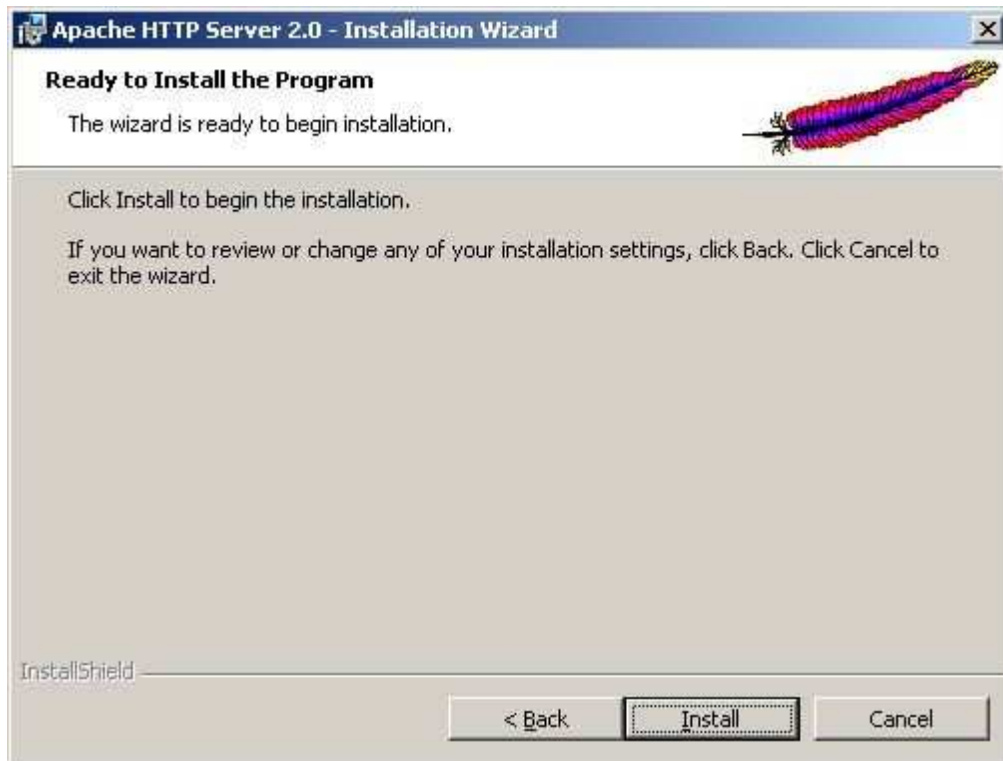


Dans le cas présent, on souhaite procéder à l'installation dans un répertoire autre que celui par défaut « **C:\Program Files\Apache Group** ».

Pour ce faire, on clique sur le bouton **Change...**

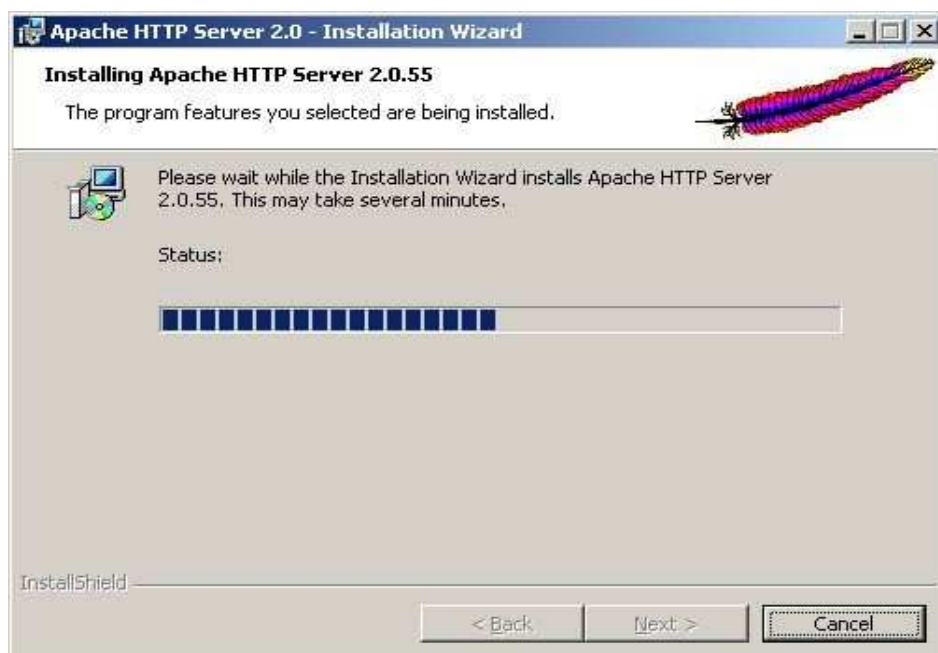


Ici, Apache sera installé dans le répertoire « **E:\projet** ». On clique sur le bouton **OK** pour valider puis sur le bouton **Next** pour continuer notre installation.



On clique sur le bouton **Install** pour procéder à l'installation.

Patience, installation en cours...



Si vous souhaitez annuler votre installation, cliquer sur le bouton **Cancel**.

Une fois l'installation terminée, nous avons l'écran ci-dessous qui s'affiche.



On clique sur le bouton **Finish**.

A présent, notre serveur est opérationnel, d'ailleurs, si l'installation était standard, les services sont déjà en action.

Par ailleurs, si vous regardez dans le menu Démarrer / Programmes, vous devriez avoir le menu suivant :



Vous aurez pu aussi remarquer la présence de **Apache Service Monitor** dans votre systray (à côté l'heure) qui est représenté par l'icône ci-dessous :



Cet utilitaire vous permet de démarrer / arrêter /redémarrer le service Apache2. Pour accéder à Apache monitor, un clique droit sur l'icône vous affichera le menu ... **Open Apache Monitor**. Ce dernier vous permettant de lancer ou non votre Apache.

Note : Si vous souhaitez lancer votre service apache sans pour cela passer par **Apache Service Monitor**. Il vous suffit de faire un **clique droit sur Poste de Travail / Gérer** ensuite, aller dans la rubrique **Services et applications / Services**. Là, on trouve le service **Apache2** sur lequel on va faire un clique droit, puis **démarrer** ou **arrêter** selon le cas.

Une astuce toute simple est d'ouvrir une invite de commande, pour cela, **Démarrer / Exécuter** puis taper **CMD** (ou cmd, ça n'a pas d'importance). La fenêtre ouverte, on tape

NET START APACHE2 (pour démarrer le service)

NET STOP APACHE2 (pour arrêter le service)

Voilà, votre service sera lancé ou arrêté selon le cas.

Ceci étant nous allons effectuer une vérification pour savoir si notre serveur est bien en cours de fonctionnement.

Ma première visite !

Pour cela, on ouvre notre navigateur favori (Firefox par exemple), puis on tape dans la barre d'adresse :

<http://localhost>

ou

<http://127.0.0.1>

Si lors de l'installation vous avez choisi l'option « **only for the Current User, on Port 8080, when started Manually** », vous devez ajouter **:8080** à la fin de l'url comme ceci :

<http://localhost:8080>

ou

<http://127.0.0.1:8080>

Ceci n'étant valide que dans le cas où vous avez entré les mêmes informations indiquées lors de l'installation.

Si vous avez rentré une ip de votre réseau local en lieu et place de **127.0.0.1**. Votre serveur est visible à l'url suivante depuis n'importe quel poste de votre réseau [LAN](#).

<http://192.168.0.1> (par exemple)

Si tout a été bien installé et fonctionne, vous devriez avoir un écran similaire à celui-ci.

Si vous lisez cette page, c'est que les propriétaires de ce domaine viennent d'installer le serveur web Apache avec succès. Ils doivent maintenant ajouter du contenu à ce répertoire et remplacer cette page, ou bien faire pointer le serveur vers l'endroit où se trouve le contenu réel du site.

Vous voyez cette page au lieu du site attendu ?

Vous voyez cette page parce que l'administrateur du site a modifié la configuration de ce serveur Web. Veuillez contacter l'**administrateur du site concerné**. La Fondation Apache (Apache Software Foundation), qui produit le logiciel Apache utilisé par ce site, n'a rien à voir avec la maintenance de ce site et ne peut intervenir sur sa configuration.

La [documentation](#) Apache est incluse dans cette distribution.

Le webmaster de ce site peut librement utiliser l'image ci-dessous sur un site web utilisant le logiciel Apache. Merci d'avoir choisi Apache !

Votre serveur web est à présent online.

En cas de problème :

Si pour une raison ou une autre, vous n'arrivez pas à accéder à la page par défaut de apache (ci-dessus), il vous faut vérifier les points suivants :

Vérifier que le service apache2 est bien démarré, pour cela utiliser le **Gestionnaire de l'ordinateur** rubrique **Services et applications / Services** dans lequel vous aurez l'état actuel du service en question.

Vérifier le fichier log de apache, ce dernier se nomme **error.log** est ce trouve dans le sous-répertoire **logs** de votre Apache. Celui-ci inscrivant les errors qui peuvent être intervenu lors du lancement du service ainsi que pendant son fonctionnement.

Configuration d'APACHE :

La configuration de Apache se fait de manière simple et ce via un seul et unique fichier de configuration. Le fichier de configuration du serveur web se nomme **httpd.conf** (un fichier texte qui sera édité avec le bloc-notes) et est situé dans le sous-répertoire **conf** d'Apache.

Ce fichier contient les principaux éléments pour faire en sorte que votre serveur web tourne sans encombre. Une modification dans ce fichier peut rendre indisponible Apache.

Voyons quelques paramètres paramétrables sans trop de difficulté.

Tout d'abord, une chose bien utile si vous ne souhaitez pas utiliser le répertoire de base de Apache pour vos documents web. Par défaut, le sous-répertoire qui contient les pages web se nomme **htdocs**, si vous souhaitez modifier cela, repérez le paramètre « **DocumentRoot** » puis modifier comme ceci :

DocumentRoot "e:/projet/www"

On aura pris soit de créer le sous-répertoire **www** dans **e:/projet** avant même d'avoir fait la modification dans le fichier de configuration, sinon cela aurait pour effet de générer une erreur lors du lancement d'Apache.

Si pour une raison ou pour un autre, on souhaite modifier l'adresse e-mail de l'administrateur du serveur, on repère le paramètre **ServerAdmin** puis on lui indique en valeur une adresse e-mail (de préférence valide).

ServerAdmin toto@nomdedomaine.com

Si l'on souhaite indiquer les fichiers qui seront traités comme des fichiers de base du serveur web, c'est-à-dire la page par défaut d'un répertoire web, nous pouvons modifier pour cela le paramètre **DirectoryIndex**.

DirectoryIndex index.htm index.html index.php index.php5

Ici, toutes les pages qui se nomment index.html, index.htm, index.php ou index.php5 seront prises en compte par le serveur web comme page par défaut d'un site web.

Pour faire en sorte que le visiteur est un minimum d'information concernant votre serveur lorsque une page d'erreur type 404 s'affiche, nous pouvons modifier la valeur du paramètre **ServerTokens**.

ServerTokens Prod

En donnant la valeur **Prod** cela permet de ne fournir que le nom du serveur, soit dans le cas présent **Apache**, il n'y aura aucune information concernant la version utilisée ni d'autres informations qui pourraient renseigner une personne mal intentionnée.

Par ailleurs, je veille à fournir une adresse e-mail qui pourrait permettre au visiteur de m'informer d'un éventuel problème sur le serveur. Pour ce faire je modifie la valeur du paramètre **Server Signature**.

Comme ceci :

ServerSignature Email

Ce qui au final lorsqu'un message d'erreur est affiché permet à tout visiteur de pouvoir prévenir l'administrateur du serveur.

Une option qui est très utile est l'utilisation du module status, un module est une fonction qui permet d'ajouter des fonctions à votre serveur web.

Le module status permet dans le cas présent d'obtenir des informations en quasi temps réel sur l'état du serveur.

Pour ce faire dans le fichier de configuration, je vais rechercher la ligne suivante :

```
# LoadModule status_module modules/mod_status.so
```

Dans le cas présent, la ligne est actuellement en commentaire puisque ayant un # en son début de ligne.

Donc, on décommente tout d'abord la ligne :

```
LoadModule status_module modules/mod_status.so
```

Puis l'on recherche les quelques lignes ci-dessous (lignes qui dans leur version d'origine sont là aussi commentés #) :

```
<Location /server-status>  
  SetHandler server-status  
  Order deny,allow  
  Deny from all  
  Allow from 127.0.0.1  
</Location>
```

Ces quelques lignes permettent de rendre ou non disponible l'état du serveur. Le **Deny from all** permet tout d'abord **interdit l'accès à tout le monde** puis avec l'option **Allow from**

