

Chapitre 2

Serveurs web

Protocole HTTP : (HyperText Transfer Protocol)

- Le protocole HTTP est le protocole de transport de données le plus utilisé sur Internet depuis 1990.
- La version 0.9 était uniquement destinée à transférer des données sur Internet (en particulier des pages Web écrites en HTML.
- La version 1.0 du protocole (la plus utilisée) permet désormais de transférer des messages avec des en-têtes décrivant le contenu du message.
- Le but du protocole HTTP est de permettre un transfert des fichiers entre un navigateur (le client) et un serveur Web.

Pourquoi utiliser HTTP ?

HTTP est devenu le protocole de communication de l'Internet. Il :

- est disponible sur toutes les plates-formes ;
- est simple. Ne requière que peu de support pour fonctionner correctement ;
- offre un niveau de sécurité simple et efficace ;
- est utilisable à travers des pare-feu.

HTTP fonctionne selon le schéma classique client/serveur :

- connexion du client vers le serveur ;
- demande d'une information via une **méthode** ;
- renvoi de l'**information** ou une **erreur** ;
- déconnexion.

Code de retour

1xx : Information

2xx : Succès (par exemple : 200 ok).

3xx : Redirection.

4xx : Erreurs (par exemple : 404 Not Found).

5xx : Erreurs venant du serveur HTTP (par exemple : 501 Not Implemented).

On verra en TP plus de codes de retours.

Les méthodes HTTP sont les suivantes :

- GET** : demande de la ressource située à l'URL spécifiée ;
- HEAD** : demande de l'en-tête de la ressource située à l'URL spécifiée ;
- POST** : envoi de données au programme situé à l'URL spécifiée ;
- PUT** : envoi de données à l'URL spécifiée ;
- DELETE** : suppression de la ressource située à l'URL spécifiée.

Requête HTTP

Une requête HTTP est un ensemble de lignes envoyé au serveur par le navigateur. Elle comprend :

- une ligne de requête précise la méthode qui doit être appliquée, et la version du protocole utilisée. La ligne comprend trois éléments séparés par un espace :
 - la méthode ;
 - l'URL ;
 - la version du protocole utilisé par le client (généralement HTTP/1.0) ;

Exemple :

GET / HTTP/1.0

- les champs d'en-tête de la requête : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la requête et/ou le client (Navigateur, système d'exploitation, ...).

Une réponse HTTP est un ensemble de lignes envoyées au navigateur par le serveur. Elle comprend :

- ❶ Une ligne de statut composée de trois éléments séparés par un espace :
 - La version du protocole utilisé
 - Le code de statut
 - La signification du code
- ❷ Les champs d'en-tête de la réponse : un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur.
- ❸ Le corps de la réponse : contient le document demandé

Exemple de requête/réponse HTTP

Requête :

GET / HTTP/1.0

Réponse :

version du protocole utilisé

Code du statut

Signification du Code

HTTP/1.1 200 OK

Date: Fri, 23 Mar 2018 12:55:58 GMT

Server: Apache/2.4.7 (Ubuntu)

Last-Modified: Thu, 22 Feb 2018 19:10:57 GMT

ETag: "2d-565d1cdbdb012"

Accept-Ranges: bytes

Content-Length: 45

Connection: close

Content-Type: text/html

informations
sur
la réponse et le
serveur

<h1>SMP</h1>

Sciences de la Matière Physique

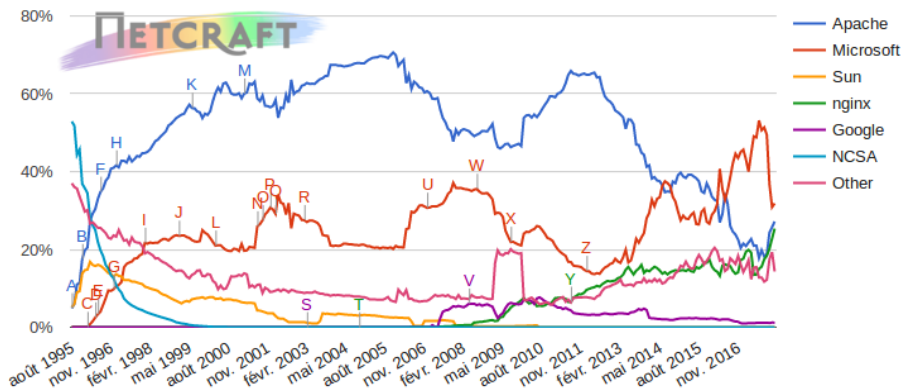
Document
demandé

Principaux serveurs :

- Apache
- Microsoft : Internet Information Server (IIS)
- nginx
- gws (Google Web Server)

Serveurs Web en ligne 2018

D'après Netcraft¹, le serveur apache est le deuxième parmi les serveurs web les plus utilisés.



1. <http://survey.netcraft.com>

Serveurs Web en ligne 2018

Le tableau suivant, montre la part du marché des principaux fournisseurs des serveurs web (selon netcraft).

Développement	12/2017	Pourcentage	01/2018	Pourcentage	Changement
Microsoft	536 million	30,89%	575 million	31,85%	0.96
Apache	446 million	25,74%	491 million	27,21%	1.47
nginx	396 million	22,83%	458 million	25,39%	2.56
Google	21 million	1,23%	22 million	1,20%	-0.03

Serveur web apache

Comme on l'a vu, apache² est parmi les serveurs web les plus populaires. Il est robuste et extensible. Il est distribué sous une licence "Open source" (Licence Apache).

Il est disponible sur plusieurs plateformes (Linux,windows, ...)

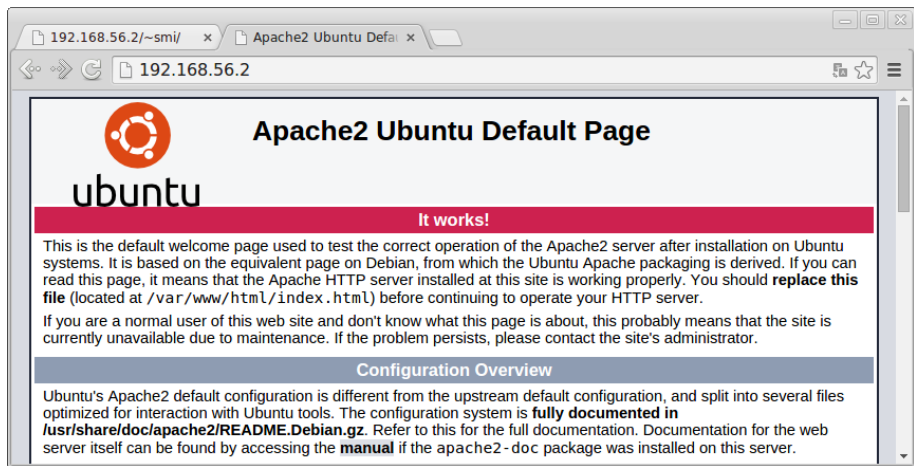
2. Site officiel : <http://httpd.apache.org/>

Une installation minimale peut être faite en ligne de commande de la façon suivante :

```
#sudo apt-get install apache2
```

Vérification de l'installation

Pour vérifier l'installation, il suffit d'utiliser un navigateur web.



Fichiers et répertoires de configuration

Les fichiers et répertoires de configuration d'apache se trouvent dans le répertoire **/etc/apache2** :

apache2.conf : fichier de configuration principale.

envvars : contient les variables d'environnement propres à apache.

conf-available/ : contient des fichiers de configuration additionnels disponibles.

conf-enabled/ : contient des fichiers de configuration activés. Ils sont utilisés dans **apache2.conf** par la ligne :

```
IncludeOptional conf-enabled/*.conf
```

ports.conf : directives de configuration pour les ports et les adresses IP d'écoutes.

Fichiers et répertoires de configuration

mods-available/ : contient une série de fichiers **.load** et **.conf**. Un fichier **.load** contient les paramètres de configuration nécessaires pour charger un module en question. Le fichier **.conf** correspondant, les paramètres de configuration nécessaires pour utiliser le module en question.

mods-enabled/ : pour utiliser un module (activer), il faut mettre un lien symbolique vers le fichier **.load** (et **.conf**, s'il existe) du module associé dans le dossier **mods-available**.

sites-available/ : même chose que **mods-available/**, mais cette fois pour les sites virtuels. Ce n'est pas obligé d'avoir le même nom pour le site et le fichier.

sites-enabled/ : même chose que **mods-enabled/**.

magic : instructions pour déterminer le type **MIME** d'un fichier (**M**ultipurpose **I**nternet **M**ail **E**xtensions - Extensions Multi-usages de la Messagerie par Internet). Par exemple text/html et image/gif.

Remarque :

Par défaut, un seul serveur est disponible (le serveur par défaut). Il est disponible dans **apache2.conf** par la ligne :

```
IncludeOptional sites-enabled/*.conf
```

Activer/désactiver un module

Les commandes **a2enmod** et **a2dismod** sont disponibles pour activer ou désactiver un module.

Exemple : pages web personnelles

Pour permettre aux utilisateurs d'avoir leurs propres pages web disponibles via un lien de type :

`http://NomSite/~utilisateur`

`http://localhost/~smi`

On tape la commande

```
#a2enmod userdir
```

Il faut ensuite redémarrer apache en tapant la commande :

```
#service apache2 restart
```

Exemple d'une page personnelle :

Dans le répertoire personnelle de l'utilisateur **smi**, il faut créer le répertoire **public_html** avec les droits **-rwxr-xr-x** et mettre dedans le fichier **index.html**, avec les droits **-rwxr--r--**.



Les commandes **a2ensite** et **a2dissite** sont disponibles pour activer ou désactiver un site. On verra leurs utilisation dans les sections suivantes.

Configuration de base

Avant de commencer la configuration, il faut faire une sauvegarde des fichiers que vous voulez modifier. Par exemple :

```
#cp apache2.conf apache2.conf.save
```

- Port à écouter (ports.conf) :

```
Listen 80
```

- Emplacement par défaut des pages html : **/var/www/html**
mettre les fichiers concernant le site web dans ce répertoire.

- Pages par défaut (mods-enabled/dir.conf) :

```
DirectoryIndex index.html index.cgi index.pl  
index.php index.xhtml index.htm
```

Sites virtuelles

Apache permet de gérer plusieurs sites web. Chaque site est appelé serveur virtuel et possède sa propre configuration.

Il y a deux types de serveurs virtuels :

- 1 Serveurs par nom ; les mêmes sites utilisent la même adresse IP.
Par exemple :

IP	Nom
192.168.56.2	smi.ump.ma
192.168.56.2	sma.ump.ma

- 2 Serveurs par adresse IP ; chaque site utilise sa propre adresse IP.
Par exemple :

IP	Nom
192.168.56.2	smi.ump.ma
192.168.56.10	sma.ump.ma

Configuration de deux sites virtuels par nom

Dans cet exemple, nous allons configurer deux sites virtuels, le premier **smi.ump.ma** et le deuxième **sma.ump.ma**. Les deux sites utilisent la même adresse IP **192.168.56.2**.

Il faut déclarer les deux noms dans le fichier **/etc/hosts** :

192.168.56.2	smi.ump.ma	www.smi.ump.ma
192.168.56.2	sma.ump.ma	www.sma.ump.ma

On pourra utiliser un serveur DNS pour déclarer les noms (voir chapitre concernant le DNS).

Configuration de deux sites virtuels par nom

Il faut créer les répertoires **smi** et **sma** associés dans **/var/www/html** :

```
# mkdir /var/www/html/smi  
# mkdir /var/www/html/sma
```

Dans **/etc/apache2/sites-available/**, il faut créer deux fichiers :
smi.conf et **sma.conf**

Contenu du fichier smi.conf

```
<VirtualHost *:80>  
    DocumentRoot /var/www/html/smi  
    ServerName smi.ump.ma  
    ServerAlias www.smi.ump.ma  
</VirtualHost>
```

Avec :

DocumentRoot : emplacement par défaut des pages html ;

ServerName : nom du serveur virtuel ;

ServerAlias : autre nom (alias) du serveur virtuel.

Contenu du fichier sma.conf

```
<VirtualHost *:80>  
    DocumentRoot /var/www/html/sma  
    ServerName sma.ump.ma  
    ServerAlias www.sma.ump.ma  
</VirtualHost>
```

Activation des deux sites

Il faut activer les deux sites en tapant les commandes :

```
# a2ensite smi  
# a2ensite sma
```

Après l'activation, il faut recharger le serveur apache en tapant la commande

```
# service apache2 reload
```

Les deux sites seront accessibles via les liens :

<http://smi.ump.ma> ou <http://www.smi.ump.ma>

<http://sma.ump.ma> ou <http://www.sma.ump.ma>

Dans cet exemple, nous allons configurer un nouveau site virtuel **smp.ump.ma**, qui utilise une adresse IP différente.

Dans cet exemple, la machine doit être munie, soit de plusieurs interfaces réseaux soit de plusieurs adresses IP associées à la même interface réseau (on parle d'IP aliasing).

IP aliasing

Pour affecter une seconde adresse IP à une interface réseau, il faut exécuter la commande :

```
# ifconfig eth0:0 192.168.56.10 up
```

Remplacez eth0 par une autre interface (par exemple eth1).

L'interface dispose, maintenant, de deux adresses distinctes :

- Adresse IP : 192.168.56.2
- Alias IP : 192.168.56.10

A vérifier avec la commande :

```
# ifconfig
```

Pour rendre la configuration permanente, il faut ajouter les lignes suivantes au fichier **/etc/network/interfaces** :

```
auto eth0:0
iface eth0:0 inet static
address 192.168.56.10
netmask 255.255.255.0
```

Remarque : on peut ajouter autant d'interfaces qu'on veut (eth0:1, eth0:2 ...).

Configuration du site virtuel

Il faut ajouter au fichier **/etc/hosts**, les lignes suivantes :

```
192.168.56.10    smp.ump.ma  
192.168.56.10    www.smp.ump.ma
```

Il faut créer le répertoire **smp** associé dans **/var/www/html** :

```
# mkdir /var/www/html/smp
```

Dans **/etc/apache2/sites-available/**, il faut créer le fichier : **smp.conf**

Contenu du fichier smp.conf

```
<VirtualHost 192.168.56.10:80>  
    DocumentRoot /var/www/html/smp  
    ServerName    smp.ump.ma  
    ServerAlias   www.smp.ump.ma  
</VirtualHost>
```

Activation du nouveau site

Il faut activer le site en tapant la commande :

```
# a2ensite smp
```

Après l'activation, il faut recharger le serveur apache en tapant la commande

```
# service apache2 reload
```

Le nouveau site sera accessible via les liens :

<http://smp.ump.ma> ou <http://www.smp.ump.ma>

Sécuriser apache

Apache est très modulaire. Dans le chapitre suivant, on verra un module concernant **php**. Dans cette section, on va utiliser un module important dans l'aspect sécurité. Le module **mod_ssl** ajoute la possibilité de crypter les communications entre le client et le serveur.

Le module **mod_ssl** se trouve dans le package **apache2-common**. Pour l'activer, il faut taper la commande :

```
sudo a2enmod ssl
```

suivie de la commande

```
sudo service apache2 restart
```

Après l'activation, il faut utiliser le préfixe **https://** devant l'adresse du serveur dans la barre du navigateur (par exemple :

<https://192.168.56.2/>).

On a vu un exemple d'utilisation avec **wireshark** (<https://www.wireshark.org>).