

Web

Le serveur HTTP Apache

IUT de Fontainebleau

8 janvier 2014

- 1 Modèle client/serveur
- 2 Le protocole HTTP
- 3 Configuration Apache

- 1 Modèle client/serveur
- 2 Le protocole HTTP
- 3 Configuration Apache



Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.
 - Interprète les commandes de formatages et de mises en forme.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.
 - Interprète les commandes de formatages et de mises en forme.
 - Va chercher et affiche des images.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.
 - Interprète les commandes de formatages et de mises en forme.
 - Va chercher et affiche des images.
 - Animation(javascript, gifs, etc....)

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.
 - Interprète les commandes de formatages et de mises en forme.
 - Va chercher et affiche des images.
 - Animation(javascript, gifs, etc....)
 - Affiche enfin la page.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.
 - Interprète les commandes de formatages et de mises en forme.
 - Va chercher et affiche des images.
 - Animation(javascript, gifs, etc....)
 - Affiche enfin la page.
- Paramétrage à plusieurs niveaux.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.
 - Interprète les commandes de formatages et de mises en forme.
 - Va chercher et affiche des images.
 - Animation(javascript, gifs, etc....)
 - Affiche enfin la page.
- Paramétrage à plusieurs niveaux.
 - Valeurs par défaut du navigateur.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.
 - Interprète les commandes de formatages et de mises en forme.
 - Va chercher et affiche des images.
 - Animation(javascript, gifs, etc....)
 - Affiche enfin la page.
- Paramétrage à plusieurs niveaux.
 - Valeurs par défaut du navigateur.
 - Valeurs fixées par le document.

Le Client

- Le Web est un ensemble de documents pouvant contenir des liens vers d'autres documents.
- Consultation via un navigateur.
 - Il récupère la page demandée.
 - Interprète les commandes de formatages et de mises en forme.
 - Va chercher et affiche des images.
 - Animation(javascript, gifs, etc....)
 - Affiche enfin la page.
- Paramétrage à plusieurs niveaux.
 - Valeurs par défaut du navigateur.
 - Valeurs fixées par le document.
 - Préférences de l'utilisateur (stockées par le navigateur)

Le Serveur

- Il écoute en boucle les requêtes formulées par les clients.

Le Serveur

- Il écoute en boucle les requêtes formulées par les clients.
- Il vérifie la validité de la requête.

Le Serveur

- Il écoute en boucle les requêtes formulées par les clients.
- Il vérifie la validité de la requête.
- Satisfait la requête : envoi du fichier, des images, du code javascript, d'un message d'erreur, demande d'authentification,

Le Serveur

- Il écoute en boucle les requêtes formulées par les clients.
- Il vérifie la validité de la requête.
- Satisfait la requête : envoie du fichier, des images, du code javascript, d'un message d'erreur, demande d'authentification,
- Il peut exécuter localement un programme qui va générer une réponse HTML (pages dynamiques).

1 Modèle client/serveur

2 Le protocole HTTP

- Requêtes
- Réponses

3 Configuration Apache

Requête HTTP

- 1 Une ligne de requête, précisant
 - La méthode
 - L'URL
 - La version du protocole utilisé par le client (généralement HTTP/1.0)
- 2 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, ...). Chacune de ces lignes est composée d'un nom de champ, suivi de deux points " : " et de la valeur de l'en-tête.
- 3 Le corps de la requête : c'est un ensemble de lignes optionnelles devant être séparées des lignes précédentes par une ligne vide et permettant par exemple un envoi de données par une commande POST lors de l'envoi de données au serveur par un formulaire.

Méthodes

GET	Requête de la ressource située à l'URL spécifiée
HEAD	Requête 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

Entêtes http

Accept	Type de contenu accepté par le browser (par exemple text/html). Voir types MIME
Accept-Charset	Jeu de caractères attendu par le browser
Accept-Encoding	Codage de données accepté par le browser
Accept-Language	Langage attendu par le browser (anglais par défaut)
Authorization	Identification du browser auprès du serveur
Content-Encoding	Type de codage du corps de la requête
Content-Language	Type de langage du corps de la requête
Content-Length	Longueur du corps de la requête
Content-Type	Type de contenu du corps de la requête (par exemple text/html). Voir types MIME
Date	Date de début de transfert des données
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur
From	Permet de spécifier l'adresse e-mail du client
If-Modified-Since	Permet de spécifier que le document doit être envoyé s'il a été modifié depuis une certaine date
Link	Relation entre deux URL
Orig-URL	URL d'origine de la requête
Referer	URL du lien à partir duquel la requête a été effectuée
User-Agent	Chaîne donnant des informations sur le client, comme le nom et la version du navigateur, du système d'exploitation

Réponse http

- 1 Une ligne de statut : c'est une ligne précisant la version du protocole utilisé et l'état du traitement de la requête à l'aide d'un code et d'un texte explicatif. La ligne comprend trois éléments devant être séparés par un espace :
 - La version du protocole utilisé
 - Le code de statut
 - La signification du code
- 2 Les champs d'en-tête de la réponse : il s'agit d'un ensemble de lignes facultatives permettant de donner des informations supplémentaires sur la réponse et/ou le serveur. Chacune de ces lignes est composée d'un nom de champ, suivi de deux points " : " et de la valeur de l'en-tête.
- 3 Le corps de la réponse : il contient le document demandé.

Entêtes réponse http

Content-Encoding	Type de codage du corps de la réponse
Content-Language	Type de langage du corps de la réponse
Content-Length	Longueur du corps de la réponse
Content-Type	Type de contenu du corps de la réponse (par exemple text/html). Voir types MIME
Date	Date de début de transfert des données
Expires	Date limite de consommation des données
Forwarded	Utilisé par les machines intermédiaires entre le browser et le serveur
Location	Redirection vers une nouvelle URL associée au document
Server	Caractéristiques du serveur ayant envoyé la réponse

Code de retour

Code	Message
10x	Information
20x	Réussite (200 → OK, etc ...)
30x	Redirection
40x	Erreur client (400 → BAD REQUEST, 404 → NOT FOUND , etc ...)
50x	Erreur serveur (500 → INTERNAL ERROR, etc ...)

Exemple

Requête

```
telnet www 80
Trying 172.16.1.22...
Connected to www.
Escape character is '^]'.
GET http://www.arda/sitebp/exemple.html HTTP/1.0
```

Réponse

```
HTTP/1.1 200 OK
Date: Tue, 06 Jul 2010 12:19:14 GMT
Server: Apache/2.2.12 (Unix) mod_ssl/2.2.12 OpenSSL/0.9.9-dev DAV/2 PHP/5.2
Last-Modified: Tue, 06 Jul 2010 12:17:09 GMT
ETag: "34bc46-39-48ab7084989fd"
Accept-Ranges: bytes
Content-Length: 57
Connection: close
Content-Type: text/html

<html>
<body>
<p>ceci est un exemple</p>
</body>
</html>
```

1 Modèle client/serveur

2 Le protocole HTTP

- Requêtes
- Réponses

3 Configuration Apache

rien à voir avec ...



Le serveur apache (version 2.2)

- La fondation apache, parmi tous ses projets, développe et maintient un serveur HTTP open-source, qui reste le plus populaire, et ce sur la plupart des OS.



- Il y a des alternatives :
 - libres : lighttpd, Nginx, etc ...
 - propriétaires : ISS, etc ...
- Par défaut, apache http écoute sur toutes les interfaces réseaux sur le port 80.

Listen 80

- La configuration du serveur est stockée dans le fichier `httpd.conf`. Voici les principaux paramètres de bases :

Identification du serveur et emplacement des fichiers

Directives

```
ServerName www.example.com:80  
ServerAdmin root@localhost
```

```
ServerRoot /etc/httpd
```

```
# DocumentRoot /var/www/html
```

Nom du serveur

S'il a un problème, le serveur écrit un message à cette adresse

Il s'agit du répertoire où le serveur trouvera son répertoire de configuration. On trouve dans `/etc/httpd`, un lien vers `/var/log/httpd/access_log`, le fichier-journal des accès aux ressources, réussis ou non (le consulter) et `error_log` celui des erreurs

fixe la racine du serveur Web, c'est-à-dire le répertoire de base où sont cherchées par défaut les pages html

Types MIME reconnus : directive `TypesConfig` dans `mime.types`

Configuration réseau

Directives

<code># Listen 80</code>	Apache écoute sur le port tcp usuel
<code>Timeout 300</code>	Paramètre important qui fixe la durée (en secondes) d'attente maximum du serveur d'une réponse à une requete envoyée à un programme extérieur (comme un gestionnaire de base de données)
<code>KeepAliveon ...</code>	Autorise les connexions persistantes d'un client, afin de lui permettre l'envoi de plusieurs requetes sans déconnexion, avec un plafond fixé pour un client, pour servir aussi d'éventuels autres clients! et un temps d'attente maxi de la requete suivante provenant du meme client.
<code>KeepAliveTimeout</code>	temps à partir duquel la connexion est coupée

Processus

Directives

<code># User http Group http</code>	Utilisateur et groupe sous lequel est exécuté le service httpd.
<code>MinSpareServers 4 ...</code>	Nombres maximum et minimum de processus serveurs devant être en permanence disponibles, en attente de nouvelles connexions clients
<code>StartServers 4</code>	Nombre de processus serveurs démarrés à l'initialisation, en plus du processus père. Ceci explique pourquoi la requête <code>ps aux grep httpd</code> renvoie 5 PID.
<code>MaxClients 150</code>	Nombre maximum de processus qu'Apache peut lancer et gérer simultanément. Ce nombre ne peut pas excéder 254
<code>MaxRequestsPerChild 500</code>	Nombre maximum de requêtes HTTP traitées par un processus enfant avant qu'il ne soit éliminé.

Le système de fichiers

Directives

DirectoryIndex index.html
...

Alias
ScriptAlias
AccessFileName .htaccess

UserDir

Il est courant d'omettre le nom du fichier de la page d'accueil d'un site ou de l'un de ses sous-répertoires. Pour ne pas retourner systématiquement une erreur 404 signalant une adresse erronée, le serveur possède une liste standard de noms de fichiers qu'il s'efforce de trouver dans le répertoire. Cette liste ordonnée est indiquée par la clause `DirectoryIndex` association d'une URL à un répertoire local idem, mais uniquement pour les CGI
Cette clause fixe le nom du fichier à trouver dans un répertoire pour donner à apache la manière de le servir
spécifie le nom du répertoire web des utilisateurs (usuellement `public_html`)

Contrôle des traitements

Directives

Redirect	redirection d'une URL sur une autre. Exemple : Redirect /google http://www.google.fr
AddHandler	association d'une extension à un gestionnaire. Exemple : AddHandler cgi-script .cgi
ErrorDocument	spécifie le document à renvoyer si erreur . Exemple : ErrorDocument 404 /doc_missing.html

Contrôles d'accès aux répertoires

Chaque répertoire auquel Apache accède peut être configuré particulièrement (ceci s'applique aussi à ses sous-répertoires). Le paramétrage de rep se précise dans un conteneur, ensemble de clauses situées entre les balises `<Directory rep>` et `</Directory>`.

Contrôles d'accès

```
# Politique par défaut : accès interdit à  
# tous à partir de la racine /  
<Directory />  
order deny, allow  
deny from all  
Options None  
AllowOverride None  
</Directory>
```

Contrôles d'accès

```
# Pour la racine du serveur WEB, il faut bien permettre l'accès
<Directory /var/www/html>
# Options possibles : "None", "All", ou plusieurs
# combinaisons de: "Indexes", "Includes",
# "FollowSymLinks", "ExecCGI", ou "MultiViews".
Options Indexes Includes FollowSymLinks
# AllowOverride = All pour donner
# la priorité aux fichiers .htaccess
AllowOverride All
order allow,deny
# allow from = all pour permettre à tout
# le monde d'accéder aux documents
allow from all
</Directory>
```

Les clauses s'appliquent aussi à tous les sous-répertoires, à moins d'une redéfinition.

- On peut préciser les hôtes dont les requêtes seront traitées, et ceux dont les requêtes seront rejetées (clause allow et deny).
- Les paramètres d'Options permettent de contrôler l'action d'Apache sur les répertoires

Options

All None	toutes aucune option(s) permise(s)
ExecCGI	exécution de scripts autorisée
FollowSymLinks	le serveur suivra les liens symboliques rencontrés dans le répertoire
Includes	permet l'utilisation de SSI (Server Side Includes)
IncludesNOEXEC	permet l'utilisation de SSI sauf les directives #exec et #include
Indexes	autorise l'affichage du contenu d'un répertoire (si un fichier par défaut n'y est pas trouvé)

- AllowOverride précise la manière avec laquelle des directives contenues dans un fichier .htaccess seront prises en compte, si ces directives supplantent ou non celles qui sont dans le présent conteneur.









Contrôles d'accès aux répertoires

```
<Limit PUT DELETE>  
Order deny,allow  
Deny from all  
</Limit>
```

<Limit> a pour but de limiter les effets des contrôles d'accès aux méthodes HTTP spécifiées. Est associée avec les directives :

- Order et Allow.
- de type require :
 - require [group|user] nom1 , . . . , nomn
 - require valid-user
- AuthName : Information transmise à l'utilisateur
- AuthType : Basic|Digest
- AuthUserFile : chemin absolu vers le .htpasswd
- AuthGroupFile : chemin absolu vers le .htgroup

order deny,allow ou order allow,deny

request match status	Allow,Deny result	Deny,Allow result
matches Allow only		
matches Deny only		
no match		
matches both Allow-Deny		

Contrôle d'accès à des fichiers, des urls

`Files` : limite la portée des directives qu'elle contient aux fichiers précisés

```
<Files "?at.*">  
# Les directives insérées ici  
# s'appliqueront aux fichiers  
# cat.html, bat.html, hat.php  
# et ainsi de suite.  
</Files>
```

`Location` : limite la portée des directives contenues aux URLs définies par l'argument URL.

```
<Location /status>  
SetHandler server-status  
Order Deny,Allow  
Deny from all  
Allow from .example.com  
</Location>
```

Alias

- Il peut être utile de remplacer un chemin de répertoires par un nom symbolique. Ces répertoires alias peuvent être paramétrés comme les autres.
- Exemple significatif : Il s'agit d'accéder par l'alias doc aux documents HTML du serveur Linux et de ses différentes applications et services installés, qui sont regroupés dans /usr/share/doc. On réserve cette consultation aux machines du réseau local.

```
# pour accéder à la doc directement avec l'url http://Serveur/doc
Alias /doc /usr/share/doc
# pour accorder les permissions d'accès limitées au site doc
<Directory /usr/share/doc>
order deny,allow
deny from all
# permission à partir de localhost
allow from localhost, 127.0.0.1
# permission à partir des stations d'un sous-domaine
# de l'établissement
allow from domaine1.arda
Options Indexes FollowSymLinks
</Directory>
```

Repertoire et authentification

AllowOverride autorise la redéfinition de directives locales à un répertoire dans un fichier .htaccess situé dans ce dernier :

- None : les fichiers .htaccess sont ignorés
- All : tout type de redéfinition est autorisé dans le .htaccess
- AuthConfig : autorise l'authentification d'utilisateurs
- FileInfo : autorise les directives liées aux types de documents
- Indexes : autorise l'indexation des répertoires
- Limit : autorise les directives de gestion d'accès
- Options : autorise les directives se rapportant aux fonctionnalités des répertoires

Répertoire et authentification

Déclaration d'authentification nécessaire

```
<Directory /chez_moi>  
AuthUserFile /chez_moi/.htpasswd  
AuthGroupFile /dev/null  
AuthName "Acces Restreint"  
AuthType Basic  
<Limit GET POST>  
require valid-user  
</Limit>  
</Directory>
```

Délégation au .htaccess

```
<Directory /chez_toi>  
AllowOverride FileInfo AuthConfig Limit  
Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec  
</Directory>
```

Le fichier .htaccess

```
AuthName TEXTE
AuthType Basic
AuthUserFile <chemin absolu>/.htpasswd
AuthGroupFile <chemin absolu >/.htgroup
<Limit GET POST>
require group groupe1 . . . groupeN
require user utilisateur1 . . . utilisateurN
</Limit>
```

- La commande htpasswd permet de générer des fichiers .htpasswd.
- L'authentification peut se faire sur ldap, etc ...

Serveurs virtuels

Plusieurs serveurs web sur une même machine Deux possibilités de serveurs virtuels

- basés sur les IP : une adresse IP pour chaque serveur virtuel
- basés sur les noms : même adresse IP, mais nom pour chaque serveur virtuel

Basés sur les ip

Directive

```
<VirtualHost nom|IP > ...</VirtualHost>
```

Exemples Configuration des serveurs par ajout de directives internes.
Chaque serveur virtuel doit connaître à la fois son IP et son nom.

Directive

```
<VirtualHost 173.194.37.104 >  
ServerAdmin canari@titi.fr  
DocumentRoot /titi/documents  
ServerName www.titi.fr  
ErrorLog /titi/logs/error_log  
TransferLog /titi/logs/access_log  
</VirtualHost>  
<VirtualHost 173.194.37.105>  
ServerAdmin matou@grosminet.fr  
DocumentRoot /grosminet /documents  
ServerName www.grosminet.fr  
ErrorLog /grosminet/logs/error_log  
TransferLog /grosminet/logs/access_log  
</VirtualHost>
```

basés sur les noms

Directives

```
NameVirtualHost IP[ :Port]
<VirtualHost nom|IP > ...</VirtualHost>
```

Exemple

```
NameVirtualHost 193.254.105.78
<VirtualHost 193.254.105.78 >
ServerAdmin canari@titi.fr
DocumentRoot /titi /documents
ServerName www.titi.fr
ErrorLog /titi/logs/error_log
TransferLog /titi/logs/access_log
</VirtualHost>
<VirtualHost 193.254.105.78 >
ServerAdmin matou@grosminet.fr
DocumentRoot /grosminet/documents
ServerName www.grosminet.fr
ErrorLog /grosminet/logs/error_log
TransferLog /grosminet/logs/access_log
</VirtualHost>
```