

LINUX SERVER ADMINISTRATION

DOCUMENTATIE CURS

DOCUMENTATIE

INTREABA PROFESORUL

CURSURILE MELE

7 Serverul Apache » 7.5 Configurare Apache » 7.5.3 Directive principale

- 1. Shell Scripts
- 2. Linux Kernel
- 3. Serverul DHCP
- 4. Serverul FTP
- 5. NFS - Network File System
- 6. Serverul DNS
- 7. Serverul Apache
 - 7.1 Protocolul HTTP
 - 7.2 Prezentare generala server
 - 7.3 Compilare si instalare
 - 7.4 Structura Apache
 - 7.5 Configurare Apache
 - 7.5.1 Sectiuni de configurare
 - 7.5.2 Configurare Minimala
 - 7.5.3 Directive principale
 - 7.5.4 Virtual Hosting
 - 7.5.5 Controlul accesului la resurse
 - 7.6 PHP
 - 7.7 Securitate Web
- 8. Serverul MySQL
- 9. NETFILTER
- 10. Sistemul de e-Mail
- 11. Serverul Postfix
- 12. Serverul POP/IMAP
- 13. Managementul Logurilor
- 14. Exemple practice (Ubuntu 14.04 LTS)
- 15. Webmin

Directive principale

1. Adresa si portul pe care va asculta serverul.

```
Listen 192.168.0.13:80
```

In cazul in care serverul are mai multe interfete si deci IP-uri si se doreste ca Apache sa asculte pe toate interfețele se folosește:

```
Listen *:80
```

2. Userul si grupul sub care ruleaza Apache

```
User wwwuser
Group wwwgroup
```

3. Numele folosit de server pentru a se identifica clientilor. Valoarea directivei ServerName este in mod normal numele serverului (**Exemplu:** www.crystallmind.ro). Se poate folosi si adresa IP.

```
ServerName 192.168.0.13:80
```

4. Directiva ServerRoot reprezinta directorul in care se gaseste Apache si toate fisierele necesare lui.

Daca pe viitor in fisierul de configurare NU se va folosi o cale absoluta si o cale relativa aceasta va fi adaugata la ServerRoot.

```
ServerRoot /opt/apache
```

Exemplu: `dir1/access` inseamna `/opt/apache/dir1/access`

5. Directorul din care Apache va servi continut clientilor. Acesta este Webspac-e-ul.

Se poate folosi directiva Alias pentru a indica alte locatii

```
DocumentRoot /opt/apache/htdocs
```

6. Modulul dir_module introduce directiva DirectoryIndex.

DirectoryIndex reprezinta fisierul servit default clientilor in momentul in care acestia nu specifica in mod express un fisier.

Exemplu: daca DirectoryIndex are valoarea `index.php`, iar un client se conecteaza la server folosind <http://www.crystallmind.ro>, serverul va furniza clientului fisierul `index.php` din DocumentRoot ca si cand acesta s-ar fi conectat la server folosind <http://www.crystallmind.ro/index.php>

```
LoadModule dir_module modules/mod_dir.so
<IfModule dir_module>
    DirectoryIndex index.html index.php home.php
</IfModule>
```

7. Directiva IndexOptions genereaza index-ul unui director automat, asemanator comenzii `ls`. Directiva este introdusa de `autoindex_module`

```
LoadModule autoindex_module modules/mod_autoindex.so
IndexOptions FancyIndexing FoldersFirst
IndexIgnore *.jpeg
```

8. Directivele Alias si ScriptAlias se folosesc pentru a mapa un URL cu un director din sistemul de fisiere care se gaseste in afara directorului definit de DocumentRoot. Browserul poate accesa astfel fisiere din afara lui DocumentRoot. Directiva ScriptAlias se foloseste daca directorul target contine doar scripturi CGI. Cele 2 directive sunt introduse de `alias_module` care trebuie incarcat.

```
LoadModule alias_module modules/mod_alias.so
Alias /icons /var/apache/images
```

Alias se foloseste pentru a mapa o resursa din webspac-e cu una locala, iar Redirect pentru a mapa o resursa cu una remote.

Directiva Redirect este introdusa de `mod_alias`

```
Redirect /help http://www.invata-online.ro
```

9. Directiva ErrorLog stabileste fisierul in care serverul va salva mesajele de eroare. Foarte util pentru troubleshooting si debugging.

```
ErrorLog logs/error_log

#LogLevel: Control the number of messages logged to the error_log.
#Possible values include: debug, info, notice, warn, error, crit, alert, emerg.

LogLevel warn
```

10. Directiva ErrorDocument specifica fisierul sau mesajul pe care serverul il returneaza in caz de eroare standard. Exista mai multe asa numite HTTP Status Codes in functie de statusul paginii cerute serverului. Multe dintre acestea sunt coduri de eroare. [Detalii.](#)

```
#Modificarea paginilor default care vor fi furnizate in caz de eroare
ErrorDocument 500 "The server made a boo boo."
ErrorDocument 404 "Lipseste!"
#ErrorDocument 404 http://www.google.com
#ErrorDocument 404 /missing.html
ErrorDocument 402 http://www.example.com/subscription_info.html
```

Nota

Apache status return codes (lista incompleta):

```
400 Bad Request - HTTP_BAD_REQUEST
401 Authorization Required - HTTP_UNAUTHORIZED
402 Payment Required - HTTP_PAYMENT_REQUIRED
403 Forbidden - HTTP_FORBIDDEN
404 Not Found - HTTP_NOT_FOUND
405 Method Not Allowed - HTTP_METHOD_NOT_ALLOWED
406 Not Acceptable - HTTP_NOT_ACCEPTABLE
407 Proxy Authentication Required - HTTP_PROXY_AUTHENTICATION_REQUIRED
408 Request Time-out - HTTP_REQUEST_TIME_OUT
409 Conflict - HTTP_CONFLICT
410 Gone - HTTP_GONE
411 Length Required - HTTP_LENGTH_REQUIRED
413 Request Entity Too Large - HTTP_REQUEST_ENTITY_TOO_LARGE
414 Request-URI Too Large - HTTP_REQUEST_URI_TOO_LARGE
500 Internal Server Error - HTTP_INTERNAL_SERVER_ERROR
501 Method Not Implemented - HTTP_NOT_IMPLEMENTED
503 Service Temporarily Unavailable - HTTP_SERVICE_UNAVAILABLE
505 HTTP Version Not Supported - HTTP_VERSION_NOT_SUPPORTED
```

11. Pentru a seta optiuni pentru un anumit director din WebSpace se folosesc directive in cadrul unui container de tipul <Directory>

```
<Directory directorul_caruia_ii_setam_optiunile>
    Optiune1
    Optiune2
</Directory>
```

Exemplu:

```
<Directory /opt/apache/htdocs/docs>
    Options Indexes FollowSymLinks
    #apartine modulului core si poate sa apara doar in <Directory> global. Specifica directivele care pot aparea in .htaccess si
    #suprascriu directivele din contextul global sau <Directory>. Daca AllowOverride este NONE, serverul nici macar nu va mai
    #incerca sa citeasca .htaccess
    AllowOverride ALL
</Directory>
```

Important

`.htaccess` este un fisier special care se foloseste pentru a seta optiuni pentru un director din WebSpace. La pornire serverul scaneaza directoarele din WebSpace pentru fisierul `.htaccess` si incarca directive folosite in acesta. Este modalitatea prin care se pot seta optiunile pentru diferite directoare fara a modifica fisierul global de configurare `httpd.conf`. Nu orice directiva poate fi folosita in `.htaccess`.

Deci pentru a seta optiuni pentru un director acestea pot sa apara in fisierul global de configurare (doar root il poate modifica) sau intr-un fisier numit `.htaccess` in directorul respectiv (orice user cu drepturi de scriere pe director poate crea si modifica `.htaccess`).

12. Directiva `FilesMatch` se foloseste de cele mai multe ori pentru restrictionarea accesului la anumite fisiere.

Exemplu: se doreste configurarea serverului astfel incat acesta sa nu transmita clientilor anumite fisiere cum ar fi imagini sau documente.

Poate fi folosita si in `.htaccess` (in contrast cu `<Directory>` si `<Location>` care nu pot fi folosite in `.htaccess`)

Permite folosirea de regular expressions.

Este echivalenta cu directiva `<Files>`, dar de preferat. Daca se foloseste `<Files>`, stringul care trebuie "prins" de regular expression trebuie precedat cu "~" (tilda).

```
<FilesMatch "^\.ht">
    Order allow,deny
    Deny from all
</FilesMatch>

<Files ~ "\.(gif|jp?g|png)$">
    Order allow,deny
    Allow from 192.168.0.2
</Files>
```

Pentru detalii despre directivele `Order`, `Allow` si `Deny` cititi capitolul "Controlul accesului la resurse" a acestui curs.

13. Modulul `mime_module` se foloseste pentru a asocia o anumita actiune a serverului in functie de extensia fisierului cerut de client. [Detalii](#).

Exemplu: clientul cere serverului un fisier `html` sau un fisier `pdf`.

```
LoadModule mime_module modules/mod_mime.so
DefaultType text/plain
TypesConfig conf/mime.types
```

De obicei exista fisierul `mime.types` care contine toate informatiile necesare despre tipurile de fisiere si care se include in `httpd.conf`

Resurse:

- [Directivele Apache 2.4](#)
- [.htaccess](#) - Documentatia oficiala