

# Apache 2 : options et directives de configuration

## Introduction

### Sections **<Directory>/<DirectoryMatch>**, **<Location>/<LocationMatch>** et **<Files>/<FilesMatch>**

Les sections **<Directory>/<DirectoryMatch>** et **<Location>/<LocationMatch>** regroupent des directives qui s'appliquent à un emplacement précisé et à ses descendants, sous-répertoires et fichiers.

**<Directory>** : l'argument est un emplacement du système de fichiers.

**<DirectoryMatch>** : comme **<Directory>** mais l'argument est une expression rationnelle.

2. **<Location>**, l'argument fourni est une URL et non un emplacement du système de fichiers. Ne pas utiliser les sections **<Location>** pour contrôler l'accès aux répertoires du système de fichiers <sup>1)</sup>

**<LocationMatch>** : comme **<Location>** mais l'argument est une expression rationnelle.

3. **<Files>**, l'argument fourni est un nom de fichier, éventuellement avec jokers. Tous les fichiers de ce nom seront concernés.

**<FilesMatch>** : comme **<Files>** mais l'argument est une expression rationnelle.

Les sections sont traitées selon l'ordre d'apparition dans le fichier de configuration.

## Syntaxe

L'argument peut contenir des caractères génériques :

?

un caractère quelconque

\*

toute chaîne de caractères



[]

intervalles de caractères

/

Les caractères génériques ne peuvent pas remplacer un / (**<Directory "/home/user/public\_html">** ne conviendra pas pour **\*/home/user/public\_html**, alors que **<Directory "/home/\*/public\_html">**



slashes (/)

conviendra.)

un ou plusieurs / sont compris tels quels :

**aaa/////bbb** dans **url** → **aaa/////bbb**

Dans la requête, les slashes multiples sont compris comme des slashes uniques. Par exemple, **<Location "/abc/def">** répondra à une requête de la forme **/abc//def**.

## Directory

```
<Directory chemin_répertoire>
...
</Directory>
```

- **chemin\_répertoire** : chemin absolu d'un répertoire dans le système de fichiers entourer de guillemets les chemins contenant des espaces.

## 2. Location :

```
<Location url>
...
</Location>
```

- **url** : chemin d'URL de la forme **/chemin**, sans protocole, nom d'hôte, port, ou autre.

## 3. Files

```
<Files filename>
...
</Files>
```

## 4. DirectoryMatch, LocationMatch ou FilesMatch :

```
<(DirectoryMatch|LocationMatch|FilesMatch) regex>
...
<(/DirectoryMatch|LocationMatch|FilesMatch)>
```

Comme les sections **Directory**, **Location** ou **FilesMatch**

Les directives d'une section **<Directory>** ne s'appliqueront pas aux fichiers du même répertoire si on y a accédé autrement, par exemple via un lien symbolique.

## Exemples

```
<Directory "/usr/local/httpd/htdocs">
Options Indexes FollowSymLinks
</Directory>
```

Pour éviter un accès à tout le système de fichiers sans restriction, ajouter un bloc du style

```
<Directory "</pre>

```

et affiner la configuration pour les répertoires à rendre accessibles.

```
<Location "/private1">
    # ...
</Location>
```

Sans slash de fin, les directives de la section s'appliquent à **/private1**, **/private1/** et **/private1/file.txt**, mais pas à **/private1other**

```
<Location "/private2/">
    # ...
</Location>
```

Avec un slash de fin, les directives de la section s'appliquent à **/private2/** et à **/private2/file.txt**, mais pas à **/private2other**

**<Location "/">** est un cas particulier qui s'applique à l'ensemble du serveur.

```
<Files "cat.html">
    # Insérez ici des directives qui s'appliquent à cat.html
</Files>
```

```
<Files "?at.*">
    # S'applique à cat.html, bat.html, hat.php etc.
</Files>
```

## Exemples avec des expressions rationnelles

```
<DirectoryMatch "^/www/(.+)?[0-9]{3}/">
    # ...
</DirectoryMatch>
```

pour les sous-répertoires de /www/ dont le nom se compose de trois chiffres.

```
<LocationMatch "/(extra|special)/data">
    # ...
</LocationMatch>
```

### **"/(extra|special)/data"**

URL contenant les sous-chaînes /extra/data ou /special/data.

### **"^/(extra|special)/data"**

Pour une URL « commençant par /extra/data ou /special/data », préfixez l'expression rationnelle avec un ^.

## Directives utilisables dans ces sections

### Options

Définit les fonctionnalités disponibles pour un répertoire particulier

#### Syntaxe

```
Options [+|-]option [[+|-]option] ...
```

Si plusieurs directives Options s'appliquent à un répertoire, c'est la plus spécifique qui est utilisée et les autres sont ignorées → les options ne sont pas fusionnées.

Toute option précédée d'un + est ajoutée à la liste des options courantes

toute option précédée d'un - est supprimée de la liste des options courantes.

On n'a pas le droit de mélanger des options avec et sans +/- ; cela ferait planter le démarrage du serveur.

#### Valeur par défaut

```
Options FollowSymLinks
```

#### option

##### Valeurs possibles

##### None

Aucune fonctionnalité spécifique n'est activée

##### All

Toutes les options sauf MultiViews

##### ExecCGI

Autorisation de l'exécution de scripts CGI à l'aide du module mod\_cgi

##### FollowSymLinks

Le serveur va suivre les liens symboliques dans le répertoire concerné <sup>2)</sup>

##### Includes

inclusions autorisées côté serveur (SSI) à l'aide du module mod\_include

##### IncludesNOEXEC

Les inclusions côté serveur (SSI) sont permises, mais **#exec cmd** et **#exec cgi** sont désactivés. L'utilisation de **#include virtual** pour les scripts CGI est cependant toujours possible depuis des répertoires définis par ScriptAlias.

##### Indexes

Si l'URL de la requête correspond au répertoire concerné, et si aucun DirectoryIndex (par exemple index.html) n'est défini pour ce répertoire, le module mod\_autoindex va renvoyer un listing formaté du répertoire.

##### MultiViews

Les vues multiples ("multiviews") à contenu négocié à l'aide du module mod\_negotiation sont autorisées.

##### SymLinksIfOwnerMatch

Le serveur ne suivra que les liens symboliques qui renvoient vers un fichier ou un répertoire dont le propriétaire est le même que celui du lien.

**Par exemple,**

sans aucun symbole + et - :

```
<Directory "/web/docs">
  Options Indexes FollowSymLinks
</Directory>

<Directory "/web/docs/spec">
  Options Includes
</Directory>
```



Ici, seule l'option Includes sera prise en compte pour le répertoire /web/docs/spec.

Par contre, si la seconde directive Options utilise les symboles + et - :

```
<Directory "/web/docs">
  Options Indexes FollowSymLinks
</Directory>

<Directory "/web/docs/spec">
  Options +Includes -Indexes
</Directory>
```

alors, les options FollowSymLinks et Includes seront prises en compte pour le répertoire /web/docs/spec.

**AllowOverride**

## Syntaxe

```
AllowOverride All|None|directive-type [directive-type] ...
```

## Valeur par défaut

```
AllowOverride None
```

Autorise ou non l'utilisation des fichiers **.htaccess**.

Il est préférable de les désactiver (par défaut)

Pour les réactiver :

```
AllowOverride All
```

## Require all



**Require all** remplace les anciennes directives ~~Order~~ ~~Allow from all~~ et ~~Deny from all~~ désormais obsolètes.

### Syntaxe

```
Require all (granted|denied)
```

granted

accès autorisé pour toutes les requêtes

denied

accès interdit pour toutes les requêtes

### Exemples :



```
Require all granted
```

autorise l'accès à ce répertoire

```
Require all denied
```

interdit l'accès à ce répertoire

## CGIMapExtension

Localise l'interpréteur des scripts CGI

### Syntaxe

```
CGIMapExtension chemin CGI .extension
```

Cette directive localise pour le serveur Apache trouve l'interpréteur servant à exécuter les scripts CGI.

Par exemple, avec la définition

```
CGIMapExtension sys:\foo.nlm .foo
```

tous les fichiers scripts CGI possédant une extension .foo seront passés à l'interpréteur foo.

# Hôtes virtuels (Directive <VirtualHost>)

## Directives utilisables dans ces sections

### ServerAdmin

Adresse e-mail incluse par le serveur dans ses messages d'erreur au client.

Syntaxe

```
ServerAdmin email-address|URL
```

Il est souvent mieux de dédier une adresse pour cela, comme :

```
ServerAdmin www-admin@foo.example.com
```

### DocumentRoot

Cette directive définit le répertoire à partir duquel httpd va servir les fichiers.

C'est la racine principale de l'arborescence des documents visible depuis Internet.

Syntaxe

```
DocumentRoot chemin_répertoire
```

chemin\_répertoire

Le répertoire défini par la directive DocumentRoot ne doit pas comporter de slash final.

Si chemin\_répertoire n'est pas un chemin absolu, il est considéré comme relatif au chemin défini par la directive ServerRoot.

S'il ne correspond pas à un Alias, le chemin de l'URL sera ajouté par le serveur à la racine des documents afin de construire le chemin du document recherché.

Valeur par défaut

```
DocumentRoot "/usr/local/apache/htdocs"
```

Exemple :



```
DocumentRoot "/usr/web"
```

un accès à <http://my.example.com/index.html> se réfère alors à /usr/web/index.html.

## ServerName

Nom d'hôte et port que le serveur utilise pour s'identifier

Syntaxe

```
ServerName [protocole://]nom-de-domaine|adresse-ip[:port]
```

**ServerName** (éventuellement conjointement avec **ServerAlias**) identifie de manière unique un hôte virtuel basé sur le nom.

Si vous utilisez des hôtes virtuels basés sur le nom, **ServerName** dans une section `<VirtualHost>` spécifie quel **Hostname** doit apparaître dans l'en-tête Host: de la requête pour correspondre à cet hôte virtuel.

## ServerAlias

La directive **ServerAlias** définit les noms alternatifs pour un hôte, à utiliser avec des hôtes virtuels. Peut inclure des caractères génériques.

Syntaxe

```
ServerAlias hostname [hostname] ...
```

```
<VirtualHost *:80>
  ServerName server.example.com
  ServerAlias server server2.example.com server2
  ServerAlias *.example.com
  UseCanonicalName Off
  # ...
</VirtualHost>
```

Les hôtes virtuels basés sur `<virtualhost>` sont traités dans l'ordre où ils apparaissent dans la configuration.

Le premier **ServerName** ou **ServerAlias** correspondant est utilisé.

## Directives

### ServerPath

Chemin-URL pour un hôte virtuel accédé par un navigateur incompatible.

Syntaxe :



## ServerPath URL-path

Contexte : virtual host

### <VirtualHost>

```
<VirtualHost>
...
</VirtualHost>
```

groupe des directives qui s'appliquent uniquement à un hôte virtuel particulier.

Lorsque le serveur reçoit une requête pour un document sur un hôte virtuel, il utilise les directives incluses dans la section <VirtualHost>

Syntaxe :

```
<VirtualHost addr[:port] [addr[:port]] ...>
...
</VirtualHost>
```

Addr

peut être l'un des suivants, éventuellement suivi d'un signe deux-points et d'un numéro de port (ou \*) :

- L'adresse IP de l'hôte virtuel
- Le caractère \*, qui agit comme un caractère générique et correspond à n'importe quelle adresse IP.
- La chaîne `_default_`, un alias pour \*

```
<VirtualHost 10.1.2.3:80>
  UseCanonicalName Off
  ServerAdmin webmaster@host.example.com
  ServerName server.example.com
  ServerAlias www.server.example.com server server2.example.com server2
  ServerAlias *.example.com
  DocumentRoot "/www/docs/host.example.com"
  ErrorLog "logs/host.example.com-error_log"
  TransferLog "logs/host.example.com-access_log"
</VirtualHost>
```

Chaque hôte virtuel doit correspondre à un nom d'hôte différent pour le serveur.

ServerAdmin

Adresse e-mail que le serveur inclut dans les messages d'erreur envoyés au client

ServerName

Syntaxe :

```
ServerName hostname
```

Un `ServerName` doit être spécifié dans chaque bloc `<VirtualHost>`. S'il est absent, le `ServerName` sera hérité de la configuration du serveur principal. Lorsqu'une requête est reçue, le serveur cherche la meilleure correspondance d'un `<VirtualHost>` basée sur IP et port

- Si aucune correspondance n'est trouvée, la configuration du serveur principal est utilisée.
- Si plusieurs hôtes virtuels ont la même adresse IP et le même port, le serveur recherche la meilleure correspondance avec le nom d'hôte demandé.
- Si aucun hôte virtuel ne correspond, c'est le premier hôte virtuel avec l'adresse IP qui sera utilisé → pour une adresse IP et une combinaison de ports donnés, le premier hôte virtuel listé est l'hôte virtuel par défaut.

## ServerAlias

Syntaxe :

```
ServerAlias hostname [hostname] ...
```

Définit les noms alternatifs pour un hôte. `ServerAlias` peut inclure des caractères génériques, le cas échéant.

Les hôtes virtuels correspondants le mieux sont traités dans l'ordre dans lequel ils apparaissent dans la configuration. Le premier `ServerName` ou `ServerAlias` correspondant est utilisé.

## DocumentRoot

Syntaxe :

```
DocumentRoot directory-path
```

Définit le répertoire à partir duquel `httpd` servira les fichiers.

Le serveur ajoute le chemin de l'URL demandée à la racine du document pour faire le chemin du document

Exemple :

```
DocumentRoot "/usr/web"
```

Un accès à <http://my.example.com/index.html> → `/usr/web/index.html`

Si le chemin du répertoire n'est pas absolu, il est supposé relatif au `ServerRoot`.

`DocumentRoot` doit être spécifié sans barre oblique finale.

Require all denied

## Voir aussi

- **(fr)** <https://httpd.apache.org/docs/2.4/fr/mod/core.html>

Basé sur « <https://httpd.apache.org/docs/2.4/fr> » (documentation officielle)

1)

Les URLs sont indépendantes du système de fichiers → des URLs différentes peuvent mener au même emplacement

2)

Bien que le serveur suive les liens symboliques, il ne modifie pas le nom de chemin concerné défini par la section <Directory>

From:

<https://www.nfrappe.fr/doc-0/> - **Documentation du Dr Nicolas Frappé**

Permanent link:

<https://www.nfrappe.fr/doc-0/doku.php?id=logiciel:internet:apache:config:start>

Last update: **2022/08/13 22:14**

