

## » Idź do

- Spis treści
- Przykładowy rozdział

## » Katalog książek

- Katalog online
- Zamów drukowany katalog

## » Twój koszyk

- Dodaj do koszyka

## » Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

## » Czytelnia

- Fragmenty książek online

## » Kontakt

Helion SA  
ul. Kościuszki 1c  
44-100 Gliwice  
tel. 032 230 98 63  
e-mail: helion@helion.pl  
© Helion 1991-2008

## Apache 2. Leksykon kieszonkowy

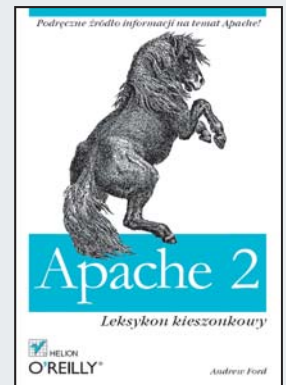
Autor: Andrew Ford

Tłumaczenie: Grzegorz Werner

ISBN: 978-83-246-2179-8

Tytuł oryginału: [Apache 2 Pocket Reference: For Apache Programmers & Administrators](#)

Format: 115×170, stron: 224



### Podręczne źródło informacji na temat Apache!

- Jak wykorzystać mechanizm mod\_rewrite?
- Jak kontrolować dostęp do zasobów?
- Jak stosować bezpieczny protokół SSL?

Apache jest jednym z najpopularniejszych serwerów HTTP. Jego zalety to niezawodność, wydajność i elastyczność. Można go uruchomić praktycznie w dowolnym systemie. Istotny jest również fakt, że Apache to rozwiązanie całkowicie bezpłatne! Pierwsza oficjalna wersja tego serwera ukazała się w 1995 roku. Oznaczono ją numerem 0.6.2. Po blisko piętnastu latach rozwiązanie to zyskało ogromną rzeszę użytkowników. Społeczność związana z tym narzędziem jest niezwykle chętna do pomocy, co niewątpliwie stanowi jeszcze jeden niezmiernie ważny atut, przemawiający na korzyść Apache.

Niniejszy leksykon jest świetną pozycją dla tych osób, które miały już styczność z serwerem Apache lub chcą szybko opanować konkretne zagadnienia związane z jego konfiguracją. Dzięki tej książce poznasz wszystkie dyrektywy, które mogą być przydatne w codziennej pracy. Dowiesz się, jak skonfigurować środowisko serwera, oraz poznasz zasady odwzorowywania adresów URL. Ponadto nauczysz się zapewniać odpowiednią kontrolę dostępu do poszczególnych zasobów i korzystać z takich mechanizmów, jak filtry, metadane lub procedury obsługi. Wiele cennych informacji, zawartych w tej poręcznej książce, sprawi, że codzienna praca z serwerem Apache będzie dla Ciebie przyjemnością!

- Ogólne zagadnienia związane z pracą i konfiguracją Apache
- Konfiguracja wirtualnych hostów (VirtualHost)
- Zarządzanie procesami
- Odwzorowanie adresów URL
- Przekształcanie adresów z wykorzystaniem mod\_rewrite
- Kontrola dostępu
- Uwierzytelnianie z użyciem mechanizmów HTTP Basic, HTTP Digest
- Zastosowanie metadanych dokumentów
- Manipulowanie nagłówkami HTTP
- Wykorzystanie skryptów CGI
- Zastosowanie WebDAV
- Używanie filtrów
- Buforowanie treści
- Konfiguracja i wykorzystanie bezpiecznego protokołu SSL
- Logowanie zdarzeń

**Skonfiguruj Apache zgodnie ze swoimi potrzebami!**

---

# Spis treści

<b>Przedmowa .....</b>	<b>7</b>
<b>1. Wprowadzenie .....</b>	<b>10</b>
Przegląd architektoniczny	10
Jak uzyskać Apache'a?	13
Uruchamianie i zatrzymywanie Apache'a	14
Pliki konfiguracyjne	17
<b>2. Środowisko serwera .....</b>	<b>28</b>
Główny serwer i hosty wirtualne	28
Konfiguracja sieci	32
Zarządzanie procesami	37
Ogólne dyrektywy konfiguracyjne	42
Szkielet DBD	50
Szkielet LDAP	53
<b>3. Odzworowywanie adresów URL .....</b>	<b>58</b>
Wstępna konfiguracja środowiska	58
Podstawowa translacja adresów URL	61
Aliasy i przekierowania	62
Żądania katalogów	65
Przekształcanie adresów URL z wykorzystaniem modułu mod_rewrite	66
Dokumenty z negocjacją treści	72

Katalogi użytkowników	73
Poprawianie błędnie wpisanych adresów URL	74
Mapy obrazkowe	75
Masowy hosting wirtualny	78
<b>4. Kontrola dostępu .....</b>	<b>80</b>
Kontrola dostępu niezwiązana z kontami użytkowników	80
Uwierzytelnianie	82
Dostawcy usług uwierzytelniania	86
Autoryzacja	91
Uwierzytelnianie i autoryzacja LDAP	95
<b>5. Metadane dokumentu .....</b>	<b>100</b>
Standardowe atrybuty dokumentu	100
Automatyczne określanie typu	105
Wygasanie dokumentów	106
Manipulowanie nagłówkami HTTP	107
Metapliki CERN	110
Śledzenie użytkowników	111
Zmienne środowiskowe	112
<b>6. Procedury obsługi .....</b>	<b>114</b>
Konfiguracja procedur obsługi	115
Pliki statyczne	116
Automatycznie generowane indeksy katalogów	116
Skrypty CGI	122
API rozszerzeń serwera internetowego	124
WebDAV	126
Strony statusu generowane przez serwer	127
Dostosowane komunikaty o błędach	128

<b>7. Filtry .....</b>	<b>130</b>
Proste konfiguracje filtrów	130
Dynamiczna konfiguracja filtrów	132
Filtry zewnętrzne	135
Dyrektywy modułu mod_deflate	137
Server-Side Includes (SSI)	139
Zastępowanie danych w treści odpowiedzi	143
<b>8. Buforowanie .....</b>	<b>144</b>
Proste buforowanie zorientowane na pliki	144
Buforowanie inteligentne	145
Buforowanie danych w pamięci	148
Buforowanie danych na dysku	150
<b>9. Pośredniczenie .....</b>	<b>152</b>
Podstawowa konfiguracja pośredniczenia	152
Łańcuchy pośredniczenia	157
Zwykły serwer pośredniczący	159
Odwrotny serwer pośredniczący	160
Równoważenie obciążenia	165
<b>10. Obsługa SSL/TLS .....</b>	<b>166</b>
Opcje zabezpieczonego serwera	166
Certyfikaty	169
Buforowanie sesji	172
Kontrola dostępu oparta na SSL	174
Dyrektywy związane z pośredniczeniem	178
<b>11. Rejestrowanie informacji .....</b>	<b>182</b>
Rejestrowanie błędów	182
Rejestrowanie żądań	184
Rejestrowanie skryptów	187
Rejestrowanie ruchu SSL	188
Rejestrowanie śledcze	189

<b>A</b>	<b>Programy i moduły .....</b>	<b>190</b>
	Narzędzia pomocnicze	190
	Moduły	191
<b>B</b>	<b>Zmienne środowiskowe CGI .....</b>	<b>195</b>
	Standardowe zmienne CGI	195
	Dodatkowe zmienne SSL	197
<b>C</b>	<b>Formaty czasu strftime() .....</b>	<b>201</b>
<b>D</b>	<b>Kody stanu HTTP .....</b>	<b>203</b>
<b>E</b>	<b>Pola nagłówka HTTP .....</b>	<b>205</b>
	Skorowidz .....	209

# Rozdział 1. Wprowadzenie

W tym rozdziale krótko opisano architekturę Apache'a, pobieranie oprogramowania, uruchamianie i zatrzymywanie serwera oraz podstawy pliku konfiguracyjnego.

## Przegląd architektoniczny

Apache zwykle działa jako usługa systemowa, tak zwany demon, z procesem lub wątkiem macierzystym, który nadzoruje procesy lub wątki potomne przetwarzające żądania klientów. Pominąwszy pewne rdzenne mechanizmy, większość funkcji jest realizowana przez moduły, które mogą być statycznie połączone z serwerem albo wczytywane dynamicznie podczas rozruchu.

Systemy operacyjne różnią się pod względem implementacji funkcji takich jak łączność sieciowa i wieloprzetwarzanie. W wersji 2.0 Apache'a wprowadzono moduły MPM (*MultiProcessing Module*), które obsługują łączność sieciową i szeregowanie w sposób dostosowany do konkretnych systemów operacyjnych i wzorców użycia. Są one wymienione w tabeli 1.1. Moduły MPM używają natywnych mechanizmów systemu operacyjnego i zapewniają szeregowanie z wykorzystaniem procesów, wątków albo ich kombinacji. Apache w danym momencie używa tylko jednego modułu MPM, który musi być statycznie skompilowany z serwerem.

Moduły MPM, inne moduły oraz rdzeń serwera WWW wykorzystują środowisko uruchomieniowe Apache Portable Runtime (APR), które zapewnia spójny, niezależny od platformy interfejs do bazowego systemu operacyjnego. APR zawiera wywołania API zapewniające dostęp do baz danych SQL i serwerów LDAP; są one używane w modułach *mod\_dbd* i *mod\_ldap*, które udostępniają standardowe mechanizmy wykorzystywane przez inne moduły.

Tabela 1.1. Moduły MPM

Moduł	Opis
<i>beos</i>	Wielowątkowy moduł MPM dla systemu operacyjnego BeOS.
<i>event</i>	Eksperymentalny wariant modułu MPM <i>worker</i> .
<i>mpm_netware</i>	Wątkowy moduł MPM dla systemu Novell Netware.
<i>mpm_winnt</i>	Dwuprocesowy, wielowątkowy moduł MPM dla systemu Windows.
<i>mpm_os2</i>	Hybrydowy wieloprocesowy, wielowątkowy moduł MPM dla systemu O/S2.
<i>prefork</i>	Tradycyjny bezwątkowy moduł MPM ze wstępnym rozwidlaniem procesu.
<i>worker</i>	Hybrydowy wieloprocesowy, wielowątkowy moduł MPM.

## Przegląd operacyjny

Podczas rozruchu Apache przechodzi fazę inicjalizacyjną, zanim wejdzie w stan operacyjny. Podczas tej fazy Apache odczytuje i weryfikuje pliki konfiguracyjne, otwiera połączenia sieciowe i pliki dziennika, pozyskuje zasoby systemowe i tworzy pulę procesów potomnych lub wątków, które będą obsługiwać żądania. Apache jest zwykle uruchamiany z przywilejami użytkownika *root*, ale zrzeka się ich przed wejściem w stan operacyjny.

Po wejściu w stan operacyjny procesy potomne lub wątki Apache'a zaczynają obsługiwać przychodzące żądania. Żądania są przetwarzane w kilku etapach, a na każdym z nich dostępne są punkty wejścia, dzięki którym moduły mogą wziąć udział w przetwarzaniu. Dla każdego punktu wejścia Apache kolejno wywołuje zarejestrowane funkcje, dopóki jedna z nich nie zasygnalizuje, że przetwarzanie dobiegło końca albo że wystąpił błąd.

Moduły rejestrują procedury obsługi dla faz, w których muszą wpłynąć na przetwarzanie żądania. Ogólnie rzecz biorąc, moduł rejestruje procedury obsługi tylko dla jednej lub dwóch faz.

Kolejność faz jest następująca:

#### *Analiza składniowa żądania*

Adres URL żądania jest odwzorowywany na przestrzeń nazw systemu plików.

#### *Kontrola bezpieczeństwa*

Kontrola dostępu, uwierzytelnianie i autoryzacja.

#### *Przygotowanie żądania*

Adres URL żądania i odwzorowana ścieżka do pliku są dopasowywane do konfiguracji, aby ustalić odpowiednią procedurę obsługi treści oraz filtry i ustawić inne metadane.

#### *Generowanie treści*

Wykonanie wybranej procedury obsługi z ewentualnymi filtrami.

#### *Rejestrowanie żądania*

Zapisywanie żądania w dzienniku.

Obraz ten komplikuje nieco to, że moduły mogą wykonywać kolejne żądania, aby zwrócić dokument inny od pierwotnie żadanego albo sprawdzić, jaka byłaby odpowiedź, gdyby zażądano innego zasobu.

## **Bieżące wersje Apache'a**

Kiedy powstawała ta książka (latem 2008 roku), w powszechnym użytku były trzy podstawowe wersje Apache'a: 1.3, 2.0 i 2.2.

Apache 1.3 został wydany w czerwcu 1998 roku i przez wiele lat był najpopularniejszym serwerem WWW. W 2000 roku rozpoczęto prace nad nową architekturą Apache'a; pierwsze wydanie produkcyjne nowej wersji, Apache 2.0, zostało udostępnione w kwietniu 2002 roku. Jednocześnie wprowadzono nowy



sposób numerowania wersji: wersje poboczne oznaczone numerami nieparzystymi, takie jak 2.1 lub 2.3, to wersje rozwojowe; wersje poboczne oznaczone numerami parzystymi, takie jak 2.0 lub 2.4, to wersje stabilne. Pierwsze wydanie wersji 2.2 pojawiło się w 2005 roku, a najnowsze udostępniono w lipcu 2008 roku.

W witrynie WWW Apache'a znajduje się dokumentacja zmian wprowadzonych w poszczególnych wersjach oraz notatki dotyczące aktualizacji.

## Jak uzyskać Apache'a?

Serwer Apache jest dostępny w wersjach dla większości współczesnych platform komputerowych — większość dystrybucji Linuksa i BSD oferuje go jako standardowy pakiet, jest on również dołączony do systemu Mac OS X. Wersję dla systemu Microsoft Windows, a także pakiety źródłowe i inne pakiety binarne, można pobrać z witryny Apache'a i serwerów lustrzanych. Warto zapoznać się ze specyfiką używanej dystrybucji, ponieważ twórcy pakietów zmieniają pewne szczegóły, aby dostosować się do konwencji obowiązujących na docelowej platformie.

Można również dość łatwo skompilować Apache'a z kodu źródłowego, co ma tę zaletę, że zapewnia pełną kontrolę nad budową serwera, statycznie dołączanymi modułami itd. Dystrybucja źródłowa zawiera szczegółowe instrukcje dotyczące procesu kompilacji.

Dystrybucja źródłowa domyślnie instaluje serwer w podkatalogach katalogu `/usr/local/apache2`, które są wymienione w tabeli 1.2. Twórcy większości innych dystrybucji zmieniają ten układ.

Dystrybucje Apache'a zawierają kilka modułów i programów narzędziowych; są one wymienione w dodatku A.

Tabela 1.2. Układ standardowych katalogów Apache'a

Katalog	Zawartość
<i>bin</i>	Pliki programów (pliki programów administracyjnych są często umieszczane w katalogu <i>sbin</i> ).
<i>build</i>	Pliki używane przez program narzędziowy <i>aspx</i> .
<i>cgi-bin</i>	Skrypty CGI.
<i>conf</i>	Pliki konfiguracyjne (często przechowywane w hierarchii katalogów <i>/etc</i> ).
<i>error</i>	Komunikaty o błędach HTTP w wielu językach.
<i>htdocs</i>	Dokumenty HTML.
<i>icons</i>	Pliki ikon.
<i>include</i>	Pliki nagłówkowe C wymagane do kompilacji zewnętrznych modułów.
<i>logs</i>	Pliki dziennika i pliki stanu, na przykład plik PID; pliki stanu są często przechowywane w katalogu <i>run</i> .
<i>man</i>	Strony podręcznika systemowego (często przechowywane w systemowych katalogach <i>man</i> ).
<i>manual</i>	Lokalna kopia podręcznika Apache'a.
<i>modules</i>	Dołączane moduły.

## Uruchamianie i zatrzymywanie Apache'a

Większość dystrybucyjnych pakietów Apache'a konfiguruje serwer tak, aby był uruchamiany podczas startu systemu i zatrzymywany podczas jego zamykania.

W systemach uniksowych Apache zwykle działa jako demon. Do Apache'a dołączony jest skrypt powłoki, *apachectl*, który automatyzuje proces uruchamiania i zatrzymywania demona. Skrypt ten jest zwykle wywoływany przez systemowy skrypt rozruchowy. Apache reaguje na następujące sygnały wysłane do procesu macierzystego (którego identyfikator jest zapisany w pliku PID):

TERM

Zatrzymuje serwer, nakazując procesowi macierystemu usunięcie wszystkich procesów potomnych i zakończenie działania.

HUP

Ponownie uruchamia serwer, nakazując procesowi macierzystemu usunięcie wszystkich procesów potomnych, ponowne odczytanie plików konfiguracyjnych i uruchomienie nowych procesów potomnych. Podczas restartu statystyki serwera są zerowane.

USR1

Inicjuje łagodny restart. Procesy potomne kończą działanie po przetworzeniu bieżącego żądania albo natychmiast, jeśli obecnie nie obsługują żadnego żądania. Proces macierzysty ponownie odczytuje pliki konfiguracyjne i zaczyna uruchamiać nowe procesy potomne, aby utrzymać odpowiednią liczbę procesów serwera. Podczas łagodnego restartu statystyki serwera nie są zerowane.

WINCH

Inicjuje łagodne zatrzymanie. Procesy potomne kończą działanie po przetworzeniu bieżącego żądania albo natychmiast, jeśli nie obsługują żadnego innego żądania. Proces macierzysty usuwa plik PID i przestaje nasłuchiwać żądań we wszystkich portach, ale kontynuuje monitorowanie, aż pozostałe procesy potomne zakończą działanie albo upłynie limit czasu.

Jeśli Apache w systemie Windows został zainstalowany jako usługa, można uruchamiać i zatrzymywać go za pomocą poleceń NET START i NET STOP albo programu Apache Service Monitor.

## Opcje wiersza polecenia

W przypadku ręcznego uruchamiania Apache'a program serwera przyjmuje następujące opcje wiersza polecenia:

-C *dyrektywa*

Przetwarza dyrektywę przed odczytaniem plików konfiguracyjnych.

- c *dyrektywa*  
Przetwarza dyrektywę po odczytaniu plików konfiguracyjnych.
- d *katalog*  
Ustawia początkową wartość dyrektywy `ServerRoot`.
- D *parametr*  
Definiuje parametr, którego można używać w sekcjach `<IfDefine>`. Niektóre opcje rozruchowe przywołuje się przez ustawianie parametrów (`DEBUG`, `FOREGROUND`, `NO_DETACH`, `ONE_PROCESS`).
- E *nazwa\_pliku*  
Określa plik dziennika błędów na czas rozruchu serwera.
- e *poziom*  
Określa poziom rejestrowania zdarzeń na czas rozruchu serwera.
- f *plik*  
Określa główny plik konfiguracyjny (domyślnie `conf/httpd.conf`).
- h  
Wyświetla krótkie podsumowanie dostępnych opcji wiersza polecenia.
- k *polecenie*  
Wykonuje jedno z następujących poleceń: `start`, `restart`, `graceful`, `stop` lub `graceful-stop`. Ponadto, tylko w systemie Windows: `install`, `uninstall`.
- L  
Wyświetla listę dostępnych dyrektyw konfiguracyjnych (udostępnianych przez wkompiłowane moduły) i kończy działanie programu.
- l  
Wyświetla listę wkompiłowanych modułów i kończy działanie programu.

- M Wyświetla listę wkompileowanych oraz współdzielonych modułów i kończy działanie programu (odpowiednik -D DUMP\_MODULES).
- n *nazwa*  
Tylko w Windows: nazwa usługi Apache'a.
- S Pokazuje ustawienia wirtualnego hosta (odpowiednik -D DUMP\_VHOSTS).
- t Testuje składnię plików konfiguracyjnych, sprawdza, czy istnieją główne katalogi dokumentów, i kończy działanie programu.
- v Wyświetla numer wersji oraz datę kompilacji i kończy działanie programu.
- V Pokazuje ustawienia kompilacji i kończy działanie programu.
- w Tylko w Windows: pozostawia otwarte okno konsoli po uruchomieniu Apache'a.
- X Uruchamia serwer w jednoprosesowym trybie diagnostycznym (odpowiednik -D DEBUG).

## Pliki konfiguracyjne

Każdy aspekt działania Apache'a jest kontrolowany przez dyrektywy zawarte w plikach konfiguracyjnych. Kiedy Apache uruchamia się, najpierw odczytuje główny plik konfiguracyjny z lokalizacji

domyślnej albo z lokalizacji określonej przez opcję `-f` wiersza polecenia. Dodatkowe pliki konfiguracyjne mogą zostać dołączone za pomocą dyrektywy `Include`.

Pliki konfiguracyjne to zwykle pliki tekstowe, które zawierają dyrektywy, puste wiersze i komentarze. Początkowe znaki odstępów w wierszu są ignorowane, podobnie jak puste wiersze. Wiersze zaczynające się od znaku hash (`#`) są uważane za komentarze.

## Format dyrektyw konfiguracyjnych

Dyrektywy konfiguracyjne Apache'a są opisywane w standardowym formacie pokazanym poniżej.

DirectorySlash	SVDH (Indexes)
<code>mod_dir (B)</code>	ON
<code>DirectorySlash { ON   OFF }</code>	
<b>Zgodność:</b> 2.0.51 i nowsze wersje	
Jeśli jest ustawiona na <code>ON</code> , żądania katalogu, które nie są zakończone znakiem ukośnika, zostaną przekierowane pod ten sam adres URL z dołączonym ukośnikiem, aby automatyczne indeksy katalogów i względne adresy URL działały poprawnie.	

Górny wiersz zawiera po lewej stronie nazwę dyrektywy, a po prawej listę kontekstów, w których można używać dyrektywy. Konteksty oznaczono skrótami opisanymi w tabeli 1.3. Jeśli dyrektywa może być używana w pliku konfiguracyjnym katalogu i jest kontrolowana przez kategorię dyrektywy `AllowOverride`, to słowo kluczowe kategorii jest wymienione w nawiasie po skrótach kontekstu.

Tabela 1.3. Skróty kontekstu

Kontekst	Opis
S	Dyrektywy można używać w kontekście globalnym, tzn. w pliku konfiguracyjnym serwera poza sekcjami wirtualnego hosta albo kontenera systemu plików.
V	Dyrektywy można używać w sekcjach wirtualnego hosta.
D	Dyrektywy można używać w sekcjach kontenera systemu plików (<Directory>, <Files>, <Location> i warianty *Match).
H	Dyrektywy można używać w plikach konfiguracyjnych dotyczących poszczególnych katalogów (domyślnie o nazwie <i>.htaccess</i> ).
*	Wskazuje, że dyrektywy można używać więcej niż raz w danym kontekście.

Drugi wiersz podaje nazwę modułu Apache’a, który implementuje dyrektywę wymienioną po lewej stronie (lista modułów dołączonych do dystrybucji Apache’a znajduje się w dodatku A). Status modułu znajduje się w nawiasie i jest oznaczony skrótami z tabeli 1.4. Moduł może być oznaczony jako „MPM” (w takim przypadku zostaną odnotowane moduły MPM implementujące dyrektywę) albo jako „rdzenny”, co oznacza, że dyrektywa jest implementowana przez rdzenny moduł Apache’a. Domyślna wartość dyrektywy jest podana po prawej stronie.

Tabela 1.4. Kody statusu modułów

Status	Opis
B	Moduł podstawowy — dołączony do dystrybucji Apache’a i domyślnie wkompiłowany.
E	Moduł rozszerzenia — dołączony do dystrybucji Apache’a, ale domyślnie niewkompiłowany.
X	Moduł eksperymentalny — dołączony do dystrybucji Apache’a, ale domyślnie niewkompiłowany.

Następny wiersz określa składnię dyrektywy, po czym następują informacje o zgodności (jeśli mają zastosowanie). Wielkość liter w dyrektywach nie jest rozróżniana, podobnie jak w większości argumentów, które nie odnoszą się do obiektów wrażliwych na wielkość liter, takich jak nazwy plików.

## Podstawowe dyrektywy w pliku konfiguracyjnym

Te dyrektywy określają położenie plików konfiguracyjnych, nazwy dodatkowych plików wczytywanych podczas rozruchu serwera, nazwy plików konfiguracyjnych, dotyczących poszczególnych katalogów, oraz dyrektywy, które można stosować w tych plikach.

Dyrektywy konfiguracyjne Apache'a są opisywane w standardowym formacie pokazanym poniżej.

### **ServerRoot**

**S**

rdzenny                      zależnie od ustawień kompilacyjnych

`ServerRoot katalog`

Główny katalog serwera. Może zostać zmieniony za pomocą opcji `-d` wiersza polecenia. Ścieżki względne w innych dyrektywach, takich jak `Include` i `LockFile`, są interpretowane jako odnoszące się do tego katalogu. Pakiety binarne często mają inne ustawienia domyślne niż w standardowym układzie katalogów Apache'a.

### **Include**

**SVD\***

rdzenny

`Include { ścieżka_do_pliku | katalog }`

**Zgodność:** dopasowywanie symboli wieloznacznych dostępne w wersji 2.0.41 i nowszych

Odczytuje i przetwarza zawartość określonego pliku konfiguracyjnego, który jest logicznie wstawiany w miejscu dyrektywy. Nazwa



pliku może zawierać symbole wieloznaczne w stylu powłoki; w takim przypadku zostaną dołączone wszystkie pasujące pliki w kolejności leksykograficznej. W razie podania katalogu zostaną dołączone wszystkie pliki znajdujące się w tym katalogu i wszystkich jego podkatalogach, co nie jest zalecane, ponieważ może doprowadzić do wczytania niepożądanych plików.

## **AccessFileName**

**SV\***

rdzenny

.htaccess

AccessFileName *nazwa\_pliku*...

Określa nazwę pliku stosowanego do konfigurowania poszczególnych katalogów. Choć nazwa dyrektywy oraz jej wartość domyślna sugerują co innego, w pliku tym można stosować nie tylko dyrektywy kontroli dostępu. Kategorie dozwolonych dyrektyw są określone przez dyrektywę AllowOverride.

## **AllowOverride**

**D**

rdzenny

All

AllowOverride *kategoria* ...

Dyrektywa AllowOverride jest dozwolona tylko w sekcjach <Directory>, które nie używają wyrażenia regularnego. Określa ona, czy z katalogów należących do danej sekcji będą odczytywane pliki konfiguracyjne, a jeśli tak, to które kategorie dyrektyw wymienione w tabeli 1.5 będą dozwolone w tych plikach. Jeśli plik konfiguracyjny katalogu będzie zawierać niedozwolone dyrektywy, zostanie wygenerowany wewnętrzny błąd serwera.

Dyrektywa AllowOverride zastępuje ustawienia zdefiniowane dla katalogów wyższego poziomu. Słowa kluczowe All i None są interpretowane tak samo jak inne, co pozwala na pewne dziwne, ale poprawne kombinacje.

Tabela 1.5. Kategorie dyrektyw w plikach konfiguracyjnych dotyczących poszczególnych katalogów

Kategoria	Opis
None	Pliki konfiguracyjne dotyczące poszczególnych katalogów w ogóle nie są odczytywane.
All	W plikach konfiguracyjnych dotyczących poszczególnych katalogów można używać wszystkich dyrektyw.
AuthConfig	Dyrektywy związane z uwierzytelnianiem i autoryzacją.
FileInfo	Dyrektywy kontrolujące atrybuty dokumentów.
Indexes	Dyrektywy związane z indeksowaniem katalogów.
Limit	Dyrektywy kontroli dostępu.
Options	Opcje katalogów.

## Sekcje warunkowe

Sekcje warunkowe zawierają zbiory dyrektyw, które Apache powinien zignorować podczas analizy składniowej pliku konfiguracyjnego, jeśli nie jest spełniony warunek określony w dyrektywie rozpoczynającej sekcję. Sekcje warunkowe można zagnieźdzać.

**<IfDefine>** **SVDH\***  
rdzenny

```
<IfDefine [!]parametr >  
...  
</IfDefine>
```

Dyrektywy zawarte w sekcji są odczytywane tylko wtedy, gdy określony parametr jest zdefiniowany za pomocą opcji `-d` wiersza polecenia — albo kiedy nie jest zdefiniowany, jeśli poprzedzono go znakiem wykrzyknika (!).

---

rdzenny

```
<IfModule [!]moduł >
...
</IfModule>
```

Dyrektywy zawarte w sekcji są odczytywane tylko wtedy, gdy określony moduł jest aktywny — albo kiedy nie jest aktywny, jeśli poprzedzono go znakiem wykrzyknika (!). Moduły można określać za pomocą identyfikatora lub nazwy, łącznie z końcową literą *.c*, która jest wyświetlana przez opcję *-1* wiersza polecenia. Możliwość użycia identyfikatorów modułów wprowadzono w wersji 2.0. Dyrektywy tej można użyć do odróżnienia wersji 1.3 od nowszych wersji, ponieważ w serwerach 2.0 moduł rdzenny nosi nazwę *core.c*, a w serwerach 1.3 nosił nazwę *http\_core.c*.

---

mod\_version (E)

```
<IfVersion [[!]operator] wersja_główna[.wersja_poboczna
↳[.poprawka]] >
...
</IfVersion>
```

**Zgodność:** 2.0.56 i nowsze wersje

Dyrektywy zawarte w sekcji są odczytywane tylko wtedy, gdy wersja Apache'a spełnia określone kryteria. Jeśli numery poprawki i wersji pobocznej zostaną pominięte, będą potraktowane tak, jakby były ustawione na zero. Operatorem porównania może być jeden z następujących symboli: *=*, *<*, *<=*, *>* lub *>=* (*==* to synonim *=*).

Obsługiwane są również wyrażenia regularne; można użyć operatora *~* i określić wersję za pomocą łańcucha albo użyć operatora *=* i określić wersję za pomocą */wyrażenia\_regularnego/*.

Wszystkie operatory mogą być poprzedzone znakiem wykrzyknika (!), który odwraca ich znaczenie.

## Sekcje kontenerowe

Sekcje kontenerowe pozwalają ograniczyć zasięg dyrektyw do katalogu, nazwy pliku, adresu URL albo metody żądania. Dyrektywy `<Directory>`, `<DirectoryMatch>`, `<Files>` oraz `<FilesMatch>` rozpoczynają kontenery systemu plików, a dyrektywy `<Location>` i `<LocationMatch>` rozpoczynają kontenery przestrzeni WWW. Dyrektywy `<Limit>` i `<LimitExcept>` rozpoczynają sekcje kontenerowe ograniczone do określonej metody żądania.

Dyrektywy kontenerowe systemu plików i przestrzeni WWW inne niż `*Match` przyjmują argument w postaci wieloznacznego wzorca w stylu powłoki. Mają one alternatywną postać, w której pierwszy argument jest tyldą (~), a drugi jest interpretowany jako wyrażenie regularne. Postać ta jest równoważna odpowiedniej dyrektywie `*Match`, która powinna być preferowana, ponieważ tyldę można łatwo przeoczyć.

Wzorce wieloznaczne w stylu powłoki mogą zawierać metaznaki i wyrażenia w nawiasach kwadratowych: `?` dopasowuje pojedynczy znak, `*` dopasowuje dowolną liczbę znaków, a `[wyrażenie]` dopasowuje dowolny ze znaków albo zakresów znaków określonych w nawiasie.

Podczas przetwarzania żądania dyrektywy zawarte w sekcjach systemu plików i przestrzeni WWW są stosowane w następującej sekwencji:

1. Sekcje `<Directory>` bez wyrażeń regularnych oraz pliki konfiguracyjne dotyczące poszczególnych katalogów, od najkrótszej do najdłuższej ścieżki do pliku. Pliki konfiguracyjne dotyczące poszczególnych katalogów mają pierwszeństwo przed sekcjami `<Directory>`.
2. Sekcje `<DirectoryMatch>`.
3. Sekcje `<Files>` i `<FilesMatch>`.
4. Sekcje `<Location>` i `<LocationMatch>`.

Dyrektywy w sekcjach `<Directory>` i `<DirectoryMatch>` oraz pliki konfiguracyjne katalogów obowiązują również w podkatalogach, chyba że zostaną później przesłonięte.

Sekcje kontenerowe dotyczące wirtualnych hostów są stosowane po sekcjach dotyczących głównego serwera.

---

## **<Directory>** **SV\***

rdzenny

```
<Directory wzorzec >
...
</Directory>
```

Kontener na dyrektywy, które dotyczą tylko katalogów pasujących do określonego wzorca (i ich podkatalogów).

---

## **<DirectoryMatch>** **SV\***

rdzenny

```
<DirectoryMatch wyrażenie_regularne >
...
</DirectoryMatch>
```

Dyrektywy zawarte w tej sekcji dotyczą tylko katalogów (i ich podkatalogów), które pasują do określonego wyrażenia regularnego.

---

## **<Files>** **SVDH\***

rdzenny

```
<Files wzorzec >
...
</Files>
```

Dyrektywy zawarte w tej sekcji dotyczą tylko plików, które pasują do określonego wzorca.

## **<FilesMatch>** **SVDH\***

---

rdzenny

```
<FilesMatch wyrażenie_regularne >  
...  
</FilesMatch>
```

Dyrektywy zawarte w tej sekcji dotyczą tylko plików, które pasują do określonego wyrażenia regularnego.

## **<Location>** **SV\***

---

rdzenny

```
<Location wzorzec >  
...  
</Location>
```

Dyrektywy zawarte w tej sekcji dotyczą tylko adresów URL, które pasują do określonego wzorca.

## **<LocationMatch>** **SV\***

---

rdzenny

```
<LocationMatch wyrażenie_regularne >  
...  
</LocationMatch>
```

Dyrektywy zawarte w tej sekcji dotyczą tylko adresów URL, które pasują do określonego wyrażenia regularnego.

## **<Limit>** **SVDH\***

---

rdzenny

```
<Limit metoda ... >  
...  
</Limit>
```

Dyrektywy zawarte w tej sekcji dotyczą tylko pasujących metod.

rdzenny

```
<LimitExcept metoda ... >
```

```
...
```

```
</LimitExcept>
```

Dyrektywy zawarte w tej sekcji dotyczą tylko niepasujących metod.