

# OpenSUSE APACHE2

## Instalacja

Aby sprawdzić czy mamy możliwość zainstalowania lub czy występuje w naszych repozytoriach serwer apache2 wydajemy polecenie:

```
sudo zypper search apache2
```

S	Name	Summary	Type
	apache2	The Apache HTTPD Server	package
	apache2-devel	The Apache HTTPD Server	package
	apache2-event	The Apache HTTPD Server	package
	apache2-icons-oxygen	Oxygen icons for Apache 2	package
	apache2-manual	The Apache HTTPD Server	package
	apache2-mod_apparmor	AppArmor module for apache2	package
	apache2-mod_auth_gssapi	GSSAPI Module for Apache	package
	apache2-mod_auth_mellon	A SAML 2.0 authentication module for the Apache2	package
	apache2-mod_auth_mellon-diagnostics	Build of mod_auth_mellon with diagnostic logging	package
	apache2-mod_auth_mellon-doc	Documentation for mod_auth_mellon module	package
	apache2-mod_auth_openidc	Apache2.x module for an OpenID Connect enabled	package
	apache2-mod_authn_otp	Apache module for one-time password authentication	package
	apache2-mod_dnssd	Apache2 module for Zeroconf support via DNS-SD	package
	apache2-mod_encoding	Non-ASCII filename interoperability module for Apache	package
	apache2-mod_evasive	Denial of Service evasion module for Apache	package
	apache2-mod_fcgid	Alternative FastCGI module for Apache2	package
	apache2-mod_jk	Connectors between Apache and Tomcat Servlet Container	package
	apache2-mod_maxminddb	MaxMind DB Apache Module	package
	apache2-mod_mono	Run ASP.NET Pages on Unix with Apache and Mono	package
	apache2-mod_nss	SSL/TLS module for the Apache HTTP server	package
	apache2-mod_perl	Embedded Perl for Apache	package
	apache2-mod_perl-devel	Embedded Perl for Apache - Development package	package
	apache2-mod_php7	PHP7 module for the Apache 2.x webserver	package
	apache2-mod_php8	PHP8 module for the Apache 2.x webserver	package
	apache2-mod_security2	Web Application Firewall for apache httpd	package
	apache2-mod_uwsgi	uWSGI Module for Apache 2.0	package
	apache2-mod_wsgi	A WSGI interface for Python3 web applications	package
	apache2-prefork	The Apache HTTPD Server	package
	apache2-utils	The Apache HTTPD Server	package
	apache2-worker	The Apache HTTPD Server	package
	dehydrated-apache2	Apache Integration for dehydrated	package
	perl-Apache2-AuthCookieDBI	An AuthCookie module backed by a DBI database	package
	rubygem-passenger-apache2	Passenger apache module	package

```
localhost:/home/woodruf # _
```

Aby zainstalować nasz serwer wydajemy polecenie

```
sudo zypper install apache2
```

lub bezpośredni z kota roota

```
zypper install apache2
```

```
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf # zypper install apache2  
Loading repository data...  
Reading installed packages...  
Resolving package dependencies...  
  
The following 2 recommended packages were automatically selected:  
  apache2-utils w3m  
  
The following 8 NEW packages are going to be installed:  
  apache2 apache2-prefork apache2-utils libapr-util1-0 libapr1-0 libgc1 system-user-wwwrun w3m  
  
8 new packages to install.  
Overall download size: 3.3 MiB. Already cached: 0 B. After the operation, additional 8.9 MiB will be  
used.  
Continue? [y/n/v/...? shows all options] (y):
```

Po udanym procesie instalacji możemy sprawdzić jakie pakiety zostały zainstalowane

```
sudo zypper search apache2
```

S	Name	Summary	Type
i+	apache2	The Apache HTTPD Server	package
	apache2-devel	The Apache HTTPD Server	package
	apache2-event	The Apache HTTPD Server	package
	apache2-icons-oxygen	Oxygen icons for Apache 2	package
	apache2-manual	The Apache HTTPD Server	package
	apache2-mod_apparmor	AppArmor module for apache2	package
	apache2-mod_auth_gssapi	GSSAPI Module for Apache	package
	apache2-mod_auth_mellon	A SAML 2.0 authentication module for the Apac->	package
	apache2-mod_auth_mellon-diagnostics	Build of mod_auth_mellon with diagnostic logg->	package
	apache2-mod_auth_mellon-doc	Documentation for mod_auth_mellon module	package
	apache2-mod_auth_openidc	Apache2.x module for an OpenID Connect enable->	package
	apache2-mod_authn_otp	Apache module for one-time password authentic->	package
	apache2-mod_dnssd	Apache2 module for Zeroconf support via DNS-SD	package
	apache2-mod_encoding	Non-ASCII filename interoperability module fo->	package
	apache2-mod_evasive	Denial of Service evasion module for Apache	package
	apache2-mod_fcgid	Alternative FastCGI module for Apache2	package
	apache2-mod_jk	Connectors between Apache and Tomcat Servlet ->	package
	apache2-mod_maxminddb	MaxMind DB Apache Module	package
	apache2-mod_mono	Run ASP.NET Pages on Unix with Apache and Mono	package
	apache2-mod_nss	SSL/TLS module for the Apache HTTP server	package
	apache2-mod_perl	Embedded Perl for Apache	package
	apache2-mod_perl-devel	Embedded Perl for Apache - Development package	package
	apache2-mod_php7	PHP7 module for the Apache 2.x webserver	package
	apache2-mod_php8	PHP8 module for the Apache 2.x webserver	package
	apache2-mod_security2	Web Application Firewall for apache httpd	package
	apache2-mod_uwsgi	uwsgi Module for Apache 2.0	package
	apache2-mod_wsgi	A WSGI interface for Python3 web applications->	package
i	apache2-prefork	The Apache HTTPD Server	package
i	apache2-utils	The Apache HTTPD Server	package
	apache2-worker	The Apache HTTPD Server	package
	dehydrated-apache2	Apache Integration for dehydrated	package
	perl-Apache2-AuthCookieDBI	An AuthCookie module backed by a DBI database	package
	rubygem-passenger-apache2	Passenger apache module	package

```
localhost:/home/woodruf # _
```

## Uruchamianie/Zatrzymanie/Autostart

W systemie OpenSUSE tak i większości linuxów do celu manipulacjami usługami możemy wykorzystać program systemctl.

Aby sprawdzić obecny stan naszego serwera wydajemy polecenie:

```
sudo systemctl status apache2
```

```
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf # systemctl status apache2  
■ apache2.service - The Apache Webserver  
  Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled; vendor preset: disabled)  
  Active: inactive (dead)  
localhost:/home/woodruf # _
```

Aby włączyć usługę do autostartu wydajemy polecenie:

```
sudo systemctl enable apache2
```

```
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf # systemctl enable apache2  
Created symlink /etc/systemd/system/httpd.service → /usr/lib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/apache.service → /usr/lib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /usr/lib/systemd/system/apache2.service.  
localhost:/home/woodruf #
```

Uruchomienie naszej usługi

```
sudo systemctl start apache2
```

```
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf # systemctl start apache2
localhost:/home/woodruf # systemctl status apache2
● apache2.service - The Apache Webserver
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2021-11-15 13:12:51 CET; 7s ago
     Main PID: 2272 (httpd-prefork)
    Status: "Processing requests..."
       Tasks: 6
          CPU: 61ms
     CGroup: /system.slice/apache2.service
            └─2272 /usr/sbin/httpd-prefork -DSYSCONFIG -C "PidFile /run/httpd.pid" -C "Include /etc>
              └─2285 /usr/sbin/httpd-prefork -DSYSCONFIG -C "PidFile /run/httpd.pid" -C "Include /etc>
                └─2286 /usr/sbin/httpd-prefork -DSYSCONFIG -C "PidFile /run/httpd.pid" -C "Include /etc>
                  └─2287 /usr/sbin/httpd-prefork -DSYSCONFIG -C "PidFile /run/httpd.pid" -C "Include /etc>
                    └─2288 /usr/sbin/httpd-prefork -DSYSCONFIG -C "PidFile /run/httpd.pid" -C "Include /etc>
                      └─2289 /usr/sbin/httpd-prefork -DSYSCONFIG -C "PidFile /run/httpd.pid" -C "Include /etc>

Nov 15 13:12:51 localhost systemd[1]: Starting The Apache Webserver...
Nov 15 13:12:51 localhost start_apache2[2272]: AH00558: httpd-prefork: Could not reliably determine
Nov 15 13:12:51 localhost systemd[1]: Started The Apache Webserver.
lines 1-18/18 (END)
```

W przypadku zatrzymania wykorzystujemy polecenie:

```
sudo systemctl stop apache2
```

## Sprawdzenie portu działania

Standardowo serwer stron WWW działa na porcie 80 (strony nieszyfrowane) i 443 (strony szyfrowane). Aby to sprawdzić możemy wykorzystać polecenie `lsof` sprawdzając co działa na porcie 80

```
sudo lsof -i:80
```

```
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf # lsof -i:80
COMMAND  PID  USER  FD  TYPE  DEVICE  SIZE/OFF  NODE  NAME
httpd-pre 2272  root   4u  IPv6  19634    0t0    TCP  *:http (LISTEN)
httpd-pre 2285  wwwrun 4u  IPv6  19634    0t0    TCP  *:http (LISTEN)
httpd-pre 2286  wwwrun 4u  IPv6  19634    0t0    TCP  *:http (LISTEN)
httpd-pre 2287  wwwrun 4u  IPv6  19634    0t0    TCP  *:http (LISTEN)
httpd-pre 2288  wwwrun 4u  IPv6  19634    0t0    TCP  *:http (LISTEN)
httpd-pre 2289  wwwrun 4u  IPv6  19634    0t0    TCP  *:http (LISTEN)
localhost:/home/woodruf # lsof -i:443
localhost:/home/woodruf # _
```

## Konfiguracja

---

Konfiguracja serwera znajduje się w katalogu /etc/apache2. Znajduje się to sporo plików oraz kilka katalogów.

```
cd /etc/apache2
ls
```

```

localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf # cd /etc/apache2/
localhost:/etc/apache2 # ls
charset.conv          magic                mod_status.conf     ssl.csr
conf.d                mime.types           mod_userdir.conf    ssl.key
default-server.conf  mod_autoindex-defaults.conf  mod_usertrack.conf  ssl.prm
errors.conf           mod_cgid-timeout.conf  protocols.conf       sysconf ig.d
global.conf           mod_info.conf         server-tuning.conf   uid.conf
httpd.conf            mod_log_config.conf  ssl-global.conf      vhosts.d
listen.conf           mod_mime-defaults.conf  ssl.crl
loadmodule.conf       mod_reqtimeout.conf   ssl.crt
localhost:/etc/apache2 #

```

Poniżej informacje przeznaczenia poszczególnych plików:

- charset.conv - Określa, które zestawy znaków mają być używane dla różnych języków.
- conf.d/\*.conf - Pliki konfiguracyjne dodane przez inne moduły. W razie potrzeby te pliki konfiguracyjne można dołączyć do konfiguracji hosta wirtualnego. Zobacz vhosts.d/vhost.templateprzykłady. W ten sposób możesz udostępnić różne zestawy modułów dla różnych hostów wirtualnych.
- default-server.conf - Globalna konfiguracja dla wszystkich wirtualnych hostów z rozsądnymi wartościami domyślnymi. Zamiast zmieniać wartości, nadpisz je konfiguracją hosta wirtualnego.
- errors.conf - Definiuje, jak Apache reaguje na błędy. Aby dostosować te komunikaty dla wszystkich wirtualnych hostów, edytuj ten plik. W przeciwnym razie nadpisz te dyrektywy w konfiguracjach hosta wirtualnego.
- httpd.conf - Główny plik konfiguracyjny serwera Apache. Unikaj zmiany tego pliku. Zawiera głównie instrukcje i ustawienia globalne. Zastąp ustawienia globalne w odpowiednich plikach konfiguracyjnych wymienionych tutaj. Zmień ustawienia specyficzne dla hosta (takie jak dokument główny) w konfiguracji hosta wirtualnego.
- listen.conf - Wiąże Apache z określonymi adresami IP i portami. Tutaj konfigurowany jest również hosting wirtualny oparty na nazwach.
- magic - Dane dla modułu mime\_magic, który pomaga Apache automatycznie określić typ MIME nieznanego pliku. Nie zmieniaj tego pliku.
- mime.types - Typy MIME znane systemowi (w rzeczywistości jest to link do /etc/mime.types). Nie edytuj tego pliku. Jeśli chcesz dodać typy MIME niewymienione tutaj, dodaj je do mod\_mime-defaults.conf.
- mod\_\*.conf - Pliki konfiguracyjne modułów instalowanych domyślnie. Zauważ, że pliki konfiguracyjne



modułów opcjonalnych znajdują się w katalogu conf.d.

- server-tuning.conf - Zawiera dyrektywy konfiguracyjne dla różnych MPM i ogólne opcje konfiguracyjne, które kontrolują wydajność Apache. Wprowadzając tutaj zmiany, przetestuj prawidłowo swój serwer sieciowy.
- ssl-global.conf oraz ssl.\* - Globalna konfiguracja SSL i dane certyfikatu SSL.
- sysconfig.d/\*.conf - Pliki konfiguracyjne generowane automatycznie z /etc/sysconfig/apache2. Nie zmieniaj żadnego z tych plików — /etc/sysconfig/apache2 zamiast tego edytuj . Nie umieszczaj innych plików konfiguracyjnych w tym katalogu.
- uid.conf - Określa, pod jakim identyfikatorem użytkownika i grupy działa Apache. Nie zmieniaj tego pliku.
- vhosts.d/\*.conf - Tutaj powinna znajdować się konfiguracja hosta wirtualnego. Katalog zawiera pliki szablonów dla hostów wirtualnych z i bez SSL. Każdy plik w tym katalogu kończący się na .conf jest automatycznie uwzględniany w konfiguracji Apache.

## FireWall

---

W systemie działa standardowo firewall. W celu sprawdzenia do jakich usług (portów) mamy dostęp oraz w jakiej strefie możemy skorzystać z polecenie firewall-cmd

```
sudo firewall-cmd --list-all
```

```
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf # firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
localhost:/home/woodruf # _
```

Jak widać nie ma na liście naszej usługi, należy dodać ją wykonując poniższe polecenie:

```
sudo firewall-cmd --add-service=apache2
```

```
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf # firewall-cmd --add-service=apache2
success
localhost:/home/woodruf # firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: apache2 dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
localhost:/home/woodruf #
```

Pojawił się nasz serwer - apache2 w sekcji services.

## Działanie

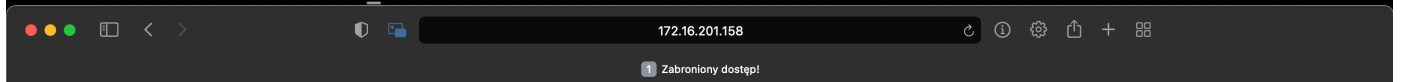
---

Najważniejszą zyczą to przetestowanie naszego serwera. Aby tego dokonać sprawdzamy jaki adres IP posiada maszyna na której jest uruchomiony serwer (na razie nie funkcjonujemy za pomocą domen) i możemy zauważyć że jest nim 172.16.201.158.

```
sudo ip a
```

Jeżeli wpisujemy go w adresie przeglądarki otrzymamy widok jak poniżej.

```
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf #
localhost:/home/woodruf # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:9d:15:6d brd ff:ff:ff:ff:ff:ff
    inet 172.16.201.158/16 brd 172.16.255.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe9d:156d/64 scope link
        valid_lft forever preferred_lft forever
localhost:/home/woodruf #
```



## Zabroniony dostęp!

Nie masz prawa dostępu do żadanego katalogu. W katalogu nie ma indeksu lub katalog jest zabezpieczony przed odczytem.

Jeśli myślisz, że jest to błąd tego serwera, skontaktuj się z [administratorem](#).

### Error 403

[172.16.201.158](#)  
Apache

Niektórzy pewnie powiedzą że serwer może nie działa lub działa nieprawidłowo jednak tak nie jest. Serwer nie posiada po prostu przy obecnej konfiguracji żadnej stworzonej strony więc wyświetla powyższy komunikat.

Możemy to naprawić tworząc prostą stronę w języku HTML. Domyślna konfiguracja otwiera strony umieszczone w katalogu `/srv/www/htdocs`

```
cd /srv/www/htdocs
```

```
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf # cd /srv/www/htdocs/  
localhost:/srv/www/htdocs # ls  
localhost:/srv/www/htdocs #
```

Tworzymy plik o nazwie index.html za pomocą edytora np. nano

```
sudo nano index.html
```

```
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf #  
localhost:/home/woodruf # cd /srv/www/htdocs/  
localhost:/srv/www/htdocs # ls  
localhost:/srv/www/htdocs # nano index.html_
```

Umieszczamy w nim przykładowy wpis:

```
<html>  
  <body>  
    <h1>Witam na serwerze Apache2</h1>  
  </h1>  
</body>  
</html>
```

I zapisujemy zmiany.

```
GNU nano 5.9                                index.html                                Modified
<html>
<body>
  <h1>Witam na serwerze Apache2</h1>
</body>
</html>
```

GNU nano 5.9 interface showing the content of index.html. The file content is: <html> <body> <h1>Witam na serwerze Apache2</h1> </body> </html>. The bottom status bar shows various nano editor shortcuts: ^G Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^\_ Replace, ^K Cut, ^U Paste, ^T Execute, ^J Justify, ^C Location, ^\_ Go To Line, ^-U Undo, ^-E Redo.

Nasz plik został utworzony i znajduje się w katalogu /srv/www/htdocs. Po odświeżeniu naszej strony (F5 lub CTRL+r) zostanie wyświetlony nasz tekst: *Witam na serwerze Apache2*.

```
localhost:/srv/www/htdocs #
localhost:/srv/www/htdocs #
localhost:/srv/www/htdocs # pwd
/srv/www/htdocs
localhost:/srv/www/htdocs # ls
index.html
localhost:/srv/www/htdocs #
```

Terminal screenshot showing the file location and listing. The terminal output is: localhost:/srv/www/htdocs #, localhost:/srv/www/htdocs #, localhost:/srv/www/htdocs # pwd, /srv/www/htdocs, localhost:/srv/www/htdocs # ls, index.html, localhost:/srv/www/htdocs #. Below the terminal is a browser window showing the URL 172.16.201.158 and the rendered page content: Witam na serwerze Apache2.

**Witam na serwerze Apache2**

# Moduły

---

Aby zwiększyć możliwości naszego serwera możemy załadować dodatkowe moduły. Część z nich jest standardowo dostarczana z serwerem, ale nie musi być aktywna.

Aby to sprawdzić wydajemy polecenie wyświetlające załadowane moduły

```
sudo a2enmod list
```

```
localhost:/etc/apache2 #  
localhost:/etc/apache2 #  
localhost:/etc/apache2 #  
localhost:/etc/apache2 # a2enmod -l  
actions alias auth_basic authn_core authn_file authz_host authz_groupfile authz_core authz_user auto  
index cgi dir env expires include log_config mime negotiation setenvif ssl socache_shmcb userdir req  
timeout  
localhost:/etc/apache2 #
```

Wszystkie moduły bazowe i rozszerzenia są szczegółowo opisane w dokumentacji Apache. Poniżej przedstawię krótki opis najważniejszych modułów. Więcej informacji można znaleźć na stronie <http://httpd.apache.org/docs/2.4/mod/>.

- `mod_actions`

Udostępnia metody do wykonania skryptu, gdy wymagany jest określony typ MIME (na przykład `application/pdf`), plik z określonym rozszerzeniem (na przykład `.rpm`) lub określona metoda żądania (na przykład `GET`). Ten moduł jest domyślnie włączony.

- `mod_alias`



Zapewnia `Alias` i `Redirect` dyrektywy, za pomocą których można zmapować adres URL do określonego katalogu ( `Alias` ) lub przekierować żądany adres URL do innej lokalizacji. Ten moduł jest domyślnie włączony.

- `mod_auth*`

Moduły uwierzytelniania udostępniają różne metody uwierzytelniania: uwierzytelnianie podstawowe za pomocą `mod_auth_basic` lub uwierzytelnianie szyfrowane za pomocą

`mod_auth_digest`. `mod_auth_basic` i `mod_auth_digest` musi być połączony z modułem dostawcy uwierzytelniania `mod_authn_*` (na przykład `mod_authn_file` do uwierzytelniania opartego na plikach tekstowych) oraz z modułem autoryzacji `mod_authz_*` (na przykład `mod_authz_user` do autoryzacji użytkownika). Więcej informacji na ten temat można znaleźć w *dokumentacji Authentication HOWTO* pod adresem <http://httpd.apache.org/docs/2.4/howto/auth.html> .

- `mod_auth_openidc`

`mod_auth_openidc` jedyny certyfikowany sposób korzystania z OpenID Connect z serwerem Apache HTTP. (Zobacz <https://openid.net/developers/certified/> .)

- `mod_autoindex`

Autoindex generuje wykazy katalogów, gdy nie ma pliku indeksu (na przykład `index.html` ). Wygląd i działanie tych indeksów można konfigurować. Ten moduł jest domyślnie włączony. Jednak listy katalogów są domyślnie wyłączone za pomocą `Options` dyrektywy — nadpisz to ustawienie w konfiguracji hosta wirtualnego. Domyślny plik konfiguracyjny tego modułu znajduje się pod adresem `/etc/apache2/mod_autoindex-defaults.conf` .

- `mod_cgi`

`mod_cgi` jest potrzebny do wykonywania skryptów CGI. Ten moduł jest domyślnie włączony.

- `mod_deflate`

Korzystając z tego modułu, Apache można skonfigurować tak, aby kompresował określone typy plików w locie przed ich dostarczeniem.

- `mod_dir`

`mod_dir` dostarcza `DirectoryIndex` dyrektywę, za pomocą której można skonfigurować, które pliki są automatycznie dostarczane, gdy żądany jest katalog ( `index.html` domyślnie). Zapewnia również automatyczne przekierowanie do prawidłowego adresu URL, gdy żądanie katalogu nie zawiera końcowego ukośnika. Ten moduł jest domyślnie włączony.

- `mod_env`

Steruje środowiskiem przekazywanym do skryptów CGI lub stron SSI. Zmienne środowiskowe można ustawiać, usuwać lub przekazywać z powłoki, która wywołała `httpd` proces. Ten moduł jest domyślnie włączony.

- `mod_expires`

Dzięki `mod_expires` , możesz kontrolować, jak często serwer proxy i pamięć podręczna przeglądarki odświeżają Twoje dokumenty, wysyłając `Expires` nagłówek. Ten moduł jest domyślnie włączony.

- `mod_http2`

Dzięki `mod_http2`, Apache zyskuje obsługę protokołu HTTP/2. Można to włączyć, określając `Protocols` `h2 http/1.1` w `VirtualHost`.

- `mod_include`

`mod_include` umożliwia korzystanie z funkcji dołączania po stronie serwera (SSI), które zapewniają podstawową funkcjonalność dynamicznego generowania stron HTML. Ten moduł jest domyślnie włączony.

- `mod_info`

Zapewnia kompleksowy przegląd konfiguracji serwera pod adresem <http://localhost/server-info/>. Ze względów bezpieczeństwa zawsze należy ograniczać dostęp do tego adresu URL. Domyślnie tylko `localhost` dostęp do tego adresu URL jest możliwy. `mod_info` jest skonfigurowany w `/etc/apache2/mod_info.conf`.

- `mod_log_config`

Za pomocą tego modułu możesz skonfigurować wygląd plików dziennika Apache. Ten moduł jest domyślnie włączony.

- `mod_mime`

Moduł mime zapewnia, że plik jest dostarczany z poprawnym nagłówkiem MIME na podstawie rozszerzenia nazwy pliku (na przykład `text/html` dla dokumentów HTML). Ten moduł jest domyślnie włączony.

- `mod_negotiation`

Niezbędne do negocjacji treści. Więcej informacji można znaleźć pod [adresem http://httpd.apache.org/docs/2.4/content-negotiation.html](http://httpd.apache.org/docs/2.4/content-negotiation.html). Ten moduł jest domyślnie włączony.

- `mod_rewrite`

Zapewnia funkcjonalność programu `mod_alias`, ale oferuje więcej funkcji i elastyczność. Dzięki `mod_rewrite` usłudze możesz przekierowywać adresy URL na podstawie wielu reguł, nagłówków żądań i nie tylko.

- `mod_setenvif`

Ustawia zmienne środowiskowe na podstawie szczegółów żądania klienta, takich jak ciąg przeglądarki wysyłany przez klienta lub adres IP klienta. Ten moduł jest domyślnie włączony.

- `mod_spelling`

`mod_spelling` próbuje automatycznie poprawiać błędy typograficzne w adresach URL, takie jak błędy wielkich liter.

- `mod_ssl`

Umożliwia szyfrowane połączenia między serwerem WWW a klientami. Zobacz [Sekcja 24.6, „Konfigurowanie bezpiecznego serwera WWW z SSL”](#), aby uzyskać szczegółowe informacje. Ten moduł jest domyślnie włączony.

- `mod_status`

Zawiera informacje o aktywności i wydajności serwera pod adresem <http://localhost/server-status/>. Ze względów bezpieczeństwa zawsze należy ograniczać dostęp do tego adresu URL. Domyślnie `localhost` dostęp do tego adresu URL jest możliwy tylko . `mod_status` jest skonfigurowany w `/etc/apache2/mod_status.conf`.

- `mod_suexec`

`mod_suexec` umożliwia uruchamianie skryptów CGI dla innego użytkownika i grupy. Ten moduł jest domyślnie włączony.

- `mod_userdir`

Włącza katalogi specyficzne dla użytkownika dostępne w . Dyrektywa musi być określony w konfiguracji. Ten moduł jest domyślnie włączony. `~*USER*/~`UserDir`

Jeżeli jest potrzeba włączenia któregoś z modułów należy skorzystać z polecenia:

```
sudo a2enmod userdir
```

```
localhost:/etc/apache2 #  
localhost:/etc/apache2 #  
localhost:/etc/apache2 #  
localhost:/etc/apache2 # a2enmod userdir  
"userdir" already present  
localhost:/etc/apache2 # _
```

## Moduły zewnętrzne

---

Tutaj znajdziesz listę wszystkich zewnętrznych modułów dostarczonych z openSUSE Leap . Znajdź dokumentację modułu w wymienionym katalogu.

- `mod_apparmor`

Dodaje obsługę Apache, aby zapewnić ograniczenie AppArmor do poszczególnych skryptów CGI obsługiwanych przez moduły, takie jak `mod_php5` .Nazwa opakowania: `apache2-mod_apparmor` Więcej informacji: *Książka „ \*Przewodnik dotyczący bezpieczeństwa i utwardzania\* ”*

- `mod_php5`

PHP to wieloplatformowy osadzony język skryptowy HTML działający po stronie serwera. Nazwa opakowania: `apache2-mod_php5` Plik konfiguracyjny: `/etc/apache2/conf.d/php5.conf` Więcej informacji: `/usr/share/doc/packages/apache2-mod_php5`

- `mod_python`

`mod_python` umożliwia osadzenie Pythona w serwerze Apache HTTP w celu znacznego zwiększenia wydajności i dodatkowej elastyczności w projektowaniu aplikacji internetowych. Nazwa opakowania: `apache2-mod_python` Więcej informacji: `/usr/share/doc/packages/apache2-mod_python`

- `mod_security`

`mod_security` udostępnia zaporę aplikacji sieci Web, która chroni aplikacje sieci Web przed szeregiem ataków. Umożliwia również monitorowanie ruchu HTTP i analizę w czasie rzeczywistym.Nazwa opakowania: `apache2-mod_security2` Plik konfiguracyjny: `/etc/apache2/conf.d/mod_security2.conf` Więcej informacji: `/usr/share/doc/packages/apache2-mod_security2` Dokumentacja: <http://modsecurity.org/documentation/>

## Programy

---

Wraz z instalacją serwera Apache2 zostają zainstalowane dodatkowe oprogramowanie. Poniżej przykład takich programów.

```
localhost:/etc/apache2/vhosts.d # a2
a2disflag a2dismod a2enflag a2enmod
localhost:/etc/apache2/vhosts.d # a2
```

```
localhost:/etc/apache2/vhosts.d # a2disflag
a2disflag: enable/disable a flag in APACHE_SERVER_FLAGS in /etc/sysconfig/apache2

usage: a2disflag [-d] flag

HTTPD_INSTANCE=<instance_name> environment variable can be used to specify
apache instance (see README-instances.txt); sysconfig file is expected
at /etc/sysconfig/apache2@<instance_name> .
localhost:/etc/apache2/vhosts.d #
```

```
localhost:/etc/apache2/vhosts.d # a2
a2disflag a2dismod a2enflag a2enmod
localhost:/etc/apache2/vhosts.d # a2dismod
a2dismod: enable/disable an apache module in APACHE_MODULES in /etc/sysconfig/apache2

usage: a2dismod [-d] module
       a2dismod -l                list modules
       a2dismod -q module         query if module is installed

HTTPD_INSTANCE=<instance_name> environment variable can be used to specify
apache instance (see README-instances.txt); sysconfig file is expected
at /etc/sysconfig/apache2@<instance_name> .
localhost:/etc/apache2/vhosts.d # _
```

## Dodatek

---

### Katalog stron błędów

Podczas działania serwera mogą wystąpić jakieś problemy wówczas przeglądarka informuje nas odpowiednim komunikatem. Informacje o danym błędzie zapisane są w poszczególnych plikach umiejscowionych w katalogu `/usr/share/apache2/error`

Istnieje możliwość własnego dostosowania informacji jaka zostanie wyświetlona podczas wystąpienia danego błędu.

```
localhost:/etc/apache2 # ls /usr/share/apache2/  
error icons lua-server-status script-helpers  
localhost:/etc/apache2 # ls /usr/share/apache2/error/  
HTTP_BAD_GATEWAY.html.var          HTTP_REQUEST_ENTITY_TOO_LARGE.html.var  
HTTP_BAD_REQUEST.html.var          HTTP_REQUEST_TIME_OUT.html.var  
HTTP_FORBIDDEN.html.var            HTTP_REQUEST_URI_TOO_LARGE.html.var  
HTTP_GONE.html.var                 HTTP_SERVICE_UNAVAILABLE.html.var  
HTTP_INTERNAL_SERVER_ERROR.html.var HTTP_UNAUTHORIZED.html.var  
HTTP_LENGTH_REQUIRED.html.var       HTTP_UNSUPPORTED_MEDIA_TYPE.html.var  
HTTP_METHOD_NOT_ALLOWED.html.var    HTTP_VARIANT_ALSO_VARIES.html.var  
HTTP_NOT_FOUND.html.var            README  
HTTP_NOT_IMPLEMENTED.html.var       contact.html.var  
HTTP_PRECONDITION_FAILED.html.var   include  
localhost:/etc/apache2 # _
```