

## Instalujemy PHP:

```
sudo apt-get install php libapache2-mod-php
```

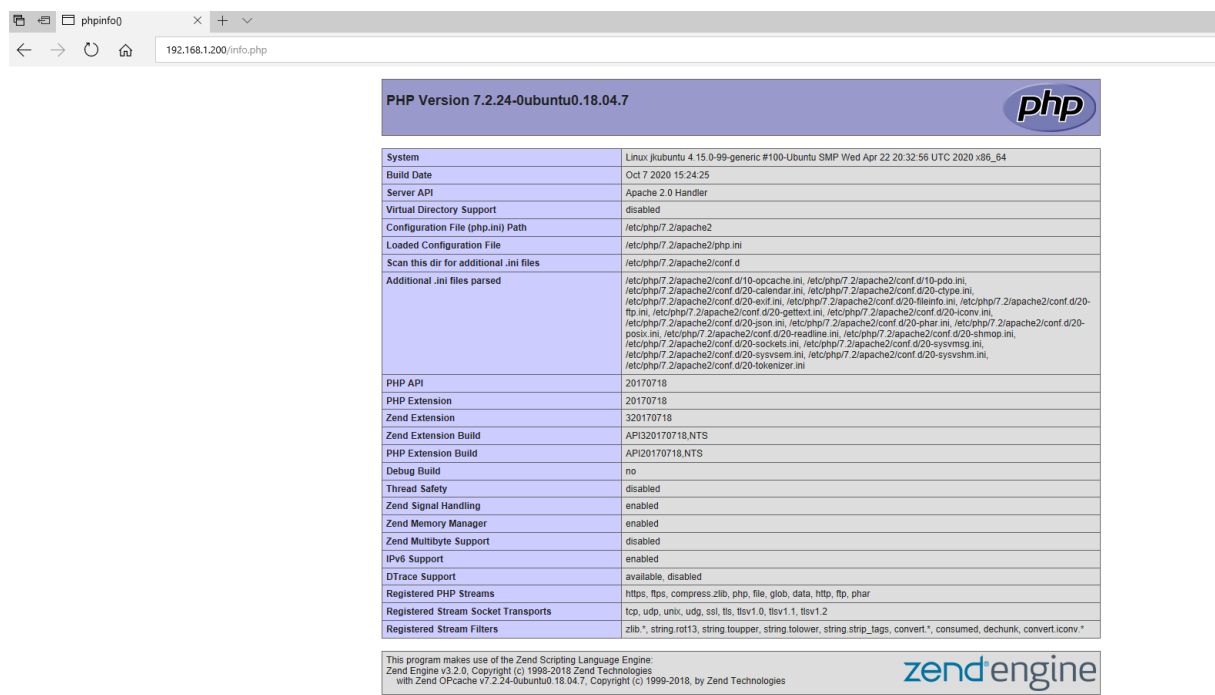
Robimy pliczek testowy PHP gdzie wywołujemy funkcje wyświetlającą informacje na temat PHP w naszym systemie:

```
sudo nano /var/www/html/info.php
```

```
<?php
    phpinfo();
?>
```

restartujemy Apache i testujemy:

```
sudo systemctl restart apache2
```



PHP Version 7.2.24-0ubuntu0.18.04.7	
System	Linux jkubuntu 4.15.0-99-generic #100-Ubuntu SMP Wed Apr 22 20:32:56 UTC 2020 x86_64
Build Date	Oct 7 2020 15:24:25
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-opcache.ini, /etc/php/7.2/apache2/conf.d/10-pdo.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-enf.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-ldap.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-syssem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v2.2.0, Copyright (c) 1998-2018 Zend Technologies  
with Zend OPcache v7.2.24-0ubuntu0.18.04.7, Copyright (c) 1999-2018, by Zend Technologies

## Ustawienie, żeby pierwszym otwieranym plikiem był index.php a nie index.html

```
sudo nano /etc/apache2/mods-enabled/dir.conf
```

przestawiamy index.php na pierwszą pozycję w tej sekcji:

```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>
```



```
Plik Maszyna Widok Wejście Urządzenia Pomoc
GNU nano 2.9.3 dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.php index.html index.cgi index.pl index.xhtml index.htm
</IfModule>
# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Tworzymy w katalogu strony domyślnej Apache pliczek index.php

```
nano /var/www/html/index.php
```

i wpisujemy taką zawartość:

```
<?php
echo "strona domyslna w PHP !!!!";
?>
```

Po zapisaniu możemy przetestować – mimo że w katalogu są teraz dwa pliki index:

```
root@jkubuntu:/var/www/html# ls -l
total 20
-rwxr-xr-x 1 root root 10904 lis 20 00:02 index.html
-rw-r--r-- 1 root root 46 lis 24 14:07 index.php
-rw-r--r-- 1 root root 44 lis 24 12:02 info.php
```

to po wpisaniu adresu serwera zamiast tej strony domyślnej Ubuntu powinien się wyświetlić nasz pliczek PHP



## Włączenie katalogów użytkowników – użytkownicy będą mogli umieszczać strony WWW w swoich katalogach

Włączamy mod do Apache2 odpowiedzialny za tą funkcję

```
sudo a2enmod userdir
```

Nie trudno się domyślić, że konfiguracja tego moda znajduje się w katalogu mods-enabled, jak zajrzemy tam, to widać, że strony użytkowników będą przechowywane w ich katalogach domowych w podkatalogu public\_html

```
nano /etc/apache2/mods-enabled/userdir.conf
```

```
GNU nano 2.9.3 userdir.conf
<IfModule mod_userdir.c>
  UserDir public_html
  UserDir disabled root

  <Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    Require method GET POST OPTIONS
  </Directory>
</IfModule>
```

Jeżeli byśmy chcieli żeby na przykład podkatalog na strony to był np.: user\_sites, to oczywiście w powyższy pliku można by to przekonfigurować – my zostawiamy tak jak jest

**Przelogujemy się** na zwykłego użytkownika, np. **zsme (stworzcie sobie usera inazwisko i robcie to z jego konta)** i stwórzmy katalog public\_html w jego katalogu domowym

```
mkdir ~/public_html
```

Zróbmy index.html i sprawdzamy czy działa (ukradnijmy z któregoś hosta wirtualnego z poprzedniego ćwiczenia ):

```
cp /var/www/zsme.local/html/index.html /home/zsme/public_html/index.html
```

Edytujemy ten pliczek do takiej postaci:

```
GNU nano 2.9.3 /home/zsme/public_html/index.html
<html>
<head></head>
<body>
<h1>Strona uzytkownika zsme</h1>
</body>
</html>
```

teraz po wpisaniu adresu naszego serwera w ten sposób

```
http://192.168.1.200/~zsme/index.html
```

powinna się zgłosić strona danego usera (w naszym przypadku to będzie user zsme)



## Strona uzytkownika zsme

### PHP z katalogów użytkowników:

Domyślnie obsługa PHP w katalogach użytkowników jest wyłączona – jeżeli użytkownik zrobi jakiś plik PHP w swoim katalogu public\_html i spróbuje go wywołać, ten po prostu nie zadziała. Żeby włączyć tą obsługę musimy wyedytować ustawienia moda php, jak wszystkie inne mody znajdują się one oczywiście w katalogu mods-enabled w ustawieniach Apache:

```
nano /etc/apache2/mods-enabled/ php7.2.conf
```

```
# Running PHP scripts in user directories is disabled by default
#
# To re-enable PHP in user directories comment the following lines
# (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
# prevents .htaccess files from disabling it.
<IfModule mod_userdir.c>
  <Directory /home/*/public_html>
    php_admin_flag engine Off
  </Directory>
</IfModule>
```

Zgodnie z informacją z komentarza komentujemy 5 linijek, zapisujemy plik i restartujemy Apache.

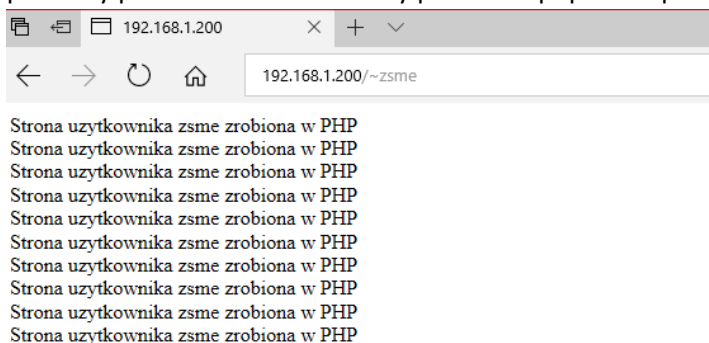
```
# Running PHP scripts in user directories is disabled by default
#
# To re-enable PHP in user directories comment the following lines
# (from <IfModule ...> to </IfModule>.) Do NOT set it to On as it
# prevents .htaccess files from disabling it.
<#IfModule mod_userdir.c>
  <#Directory /home/*/public_html>
  <#    php_admin_flag engine Off
  <#  </Directory>
<#</IfModule>
```

Teraz otwieramy sobie drugą konsolę (Ctrl+Alt+F2) i logujemy się na zsmc (wy oczywiście na swojego usera inazwisko) i robimy pliczek testowy php o nazwie index.php z pętlą:

```
nano ~/public_html/index.php
```

```
<?php
for($i=0;$i<10;$i++){
echo "Strona uzytkownika zsmc zrobiona w PHP<br>";
}
?>
```

Po wpisaniu w przeglądarkę adresu strony użytkownika zsmc zgodnie z wcześniejszą konfiguracją pierwszy powinien zostać otwarty plik index.php i interpreter PHP powinien wykonać naszą pętlę:



### Wyświetlanie błędów w plikach PHP

Domyślnie PHP ma wyłączone wyświetlanie komunikatów o błędach które zrobimy w plikach php, zamiast nich wyświetla pustą stronę - na serwerze produkcyjnym to podejście jest jak najbardziej w porządku ze względu na bezpieczeństwo naszej strony (po co ktoś ma znać nazwy naszych zmiennych itd.).

Jako użytkownik zsme skopiuję sobie nasz plik index.php i nazwę kopie blad.php

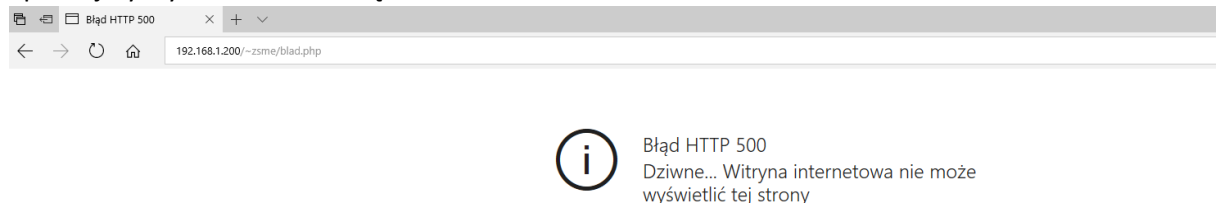
```
cp ~/public_html/index.php ~/public_html/blad.php
```

```
zsme@jkubuntu:~/public_html$ ls -l
total 12
-rw-r--r-- 1 zsme zsme 87 lis 24 16:26 blad.php
-rwxr-xr-x 1 zsme zsme 88 lis 24 15:05 index.html
-rw-r--r-- 1 zsme zsme 87 lis 24 15:17 index.php
zsme@jkubuntu:~/public_html$
```

Zróbmy teraz w naszym pliku blad.php specjalnego popularnego błędem – usuńmy średnik

```
GNU nano 2.9.3 blad.php
<?php
for($i=0;$i<10;$i++){
    echo "Strona uzytkownika zsme zrobiona w PHP</br>"
}
?>
```

Spróbujmy wyświetlić ta stronę:



Strona po prostu się nie wyświetla (w innych przeglądarkach może to być pusta strona). Natomiast dla naszej wygody, jako że to środowisko służące tylko do ćwiczeń, dobrze sobie włączyć obsługę wyświetlania błędów.

Musimy włączyć wyświetlanie komunikatów przez PHP w pliku php.ini

```
nano /etc/php/7.2/apache2/php.ini
```

Szukamy w pliku sekcji dotyczące błędów (sam plik jest mocno rozbudowany)

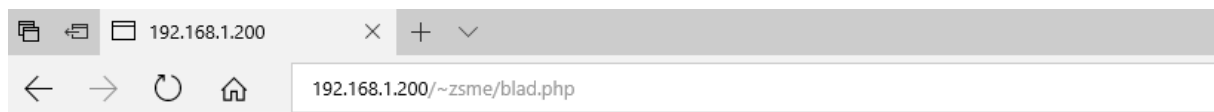
```
GNU nano 2.9.3 /etc/php/7.2/apache2/php.ini
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Error handling and logging ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

; This directive informs PHP of which errors, warnings and notices you would like
; it to take action for. The recommended way of setting values for this
; directive is through the use of the error level constants and bitwise
; operators. The error level constants are below here for convenience as well as
; some common settings and their meanings.
; By default, PHP is set to take action on all errors, notices and warnings EXCEPT
; those related to E_NOTICE and E_STRICT, which together cover best practices and
; recommended coding standards in PHP. For performance reasons, this is the
; recommend error reporting setting. Your production server shouldn't be wasting
; resources complaining about best practices and coding standards. That's what
; development servers and development settings are for.
; Note: The php.ini-development file has this setting as E_ALL. This
; means it pretty much reports everything which is exactly what you want during
; development and early testing.
;
; Error Level Constants:
; E_ALL          - All errors and warnings (includes E_STRICT as of PHP 5.4.0)
; E_ERROR        - fatal run-time errors
; E_RECOVERABLE_ERROR  - almost fatal run-time errors
; E_WARNING      - run-time warnings (non-fatal errors)
; E_PARSE        - compile-time parse errors
; E_NOTICE       - run-time notices (these are warnings which often result
;                  from a bug in your code, but it's possible that it was
;                  intentional (e.g., using an uninitialized variable and
;                  relying on the fact it is automatically initialized to an
;                  empty string)
;
```

I poniżej ustawiamy zmienną **display\_errors = On**

```
; stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
; On or stdout = Display errors to STDOUT
; Default Value: On
; Development Value: On
; Production Value: Off
; http://php.net/display-errors
display_errors = On
```

Zapisujemy, restartujemy Apache i sprawdzamy jak teraz działa



**Parse error: syntax error, unexpected '}', expecting ',' or ';' in /home/zsme/public\_html/blad.php on line 5**

Jak widać, teraz widzimy dokładną informację o błędzie – w naszym przypadku braku średnika na końcu 5 linijki.