

[Przewodnik administratora](#) > Instalacja na systemie Windows przy wykorzystaniu serwera Nginx

Menu

1. [Wstęp](#)
2. [Instalacja środowiska kontroli wersji](#)
3. [Instalacja i konfiguracja serwera Nginx](#)

Wstęp

W instalacjach, w których nie można wykorzystać serwera Apache należy użyć darmowego prostego serwera Nginx. Niniejsza instrukcja przedstawia sposób instalacji i konfiguracji serwera www opartego o takie komponenty jak:

- Serwer Nginx
- PHP
- Serwer baz danych PostgreSQL (na Windows zaleca się wersję 32bit)

W dalszym opisie zakłada się, że zainstalowano i skonfigurowano już serwer baz danych PostgreSQL. Opis instalacji serwera PostgreSQL dostępny jest na stronie: <http://support.edokumenty.eu/trac/wiki/AdminGuide/Procedures/WindowsPostgreSQL>

Instalacja środowiska kontroli wersji

Instalacja środowiska kontroli wersji została opisana już w artykule: [Instalacja na serwerze Windows \(Zend Server\)](#)

[Powrt do menu](#)

Instalacja i konfiguracja serwera Nginx

W pierwszym kroku należy pobrać następujące składniki:

- Nginx dla Windows <http://nginx.org/download/nginx-1.7.6.zip>
- PHP w wersji 5.4 NTS <http://windows.php.net/downloads/releases/php-5.4.33-nts-Win32-VC9-x86.zip>
- Zend Guard Loader <http://www.zend.com/en/products/guard/downloads#Windows>

Po pobraniu należy rozpakować archiwa. Sugerowane katalogi, to dla Nginx C:\nginx, php <Dysk instalacji>\nginx\php\. Zend Guard Loader-a można wkopiować do katalogu PHP (odniesienie do konfiguracji jest w php.ini).

Przykładowa konfiguracja Nginx-a jest zawarta w pliku <Dysk instalacji>\nginx\conf\nginx.conf. Zakłada ona, że katalogiem instalacji systemu eDokumenty jest katalog: D:\eDokumenty\public_html

```
#user nobody;
worker_processes 1;

#error_log logs/error.log;
#error_log logs/error.log notice;
#error_log logs/error.log info;

#pid logs/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include mime.types;
    default_type application/octet-stream;

    #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
```

```

#           '$status $body_bytes_sent "$http_referer" '
#           '"$http_user_agent" "$http_x_forwarded_for"';

#access_log logs/access.log main;

sendfile      on;
#tcp_nopush   on;

#keepalive_timeout 0;
keepalive_timeout 65;

#gzip on;

server {
    listen      8080;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    "D:/eDokumenty/public_html/";
        index  index.html index.htm index.php engine engine.php;
    }

    #error_page 404          /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root    html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80
    #
    #location ~ \.php$ {
    #    proxy_pass http://127.0.0.1;
    #}

    # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
    #

    location ~ \.php($|/) {

        # root            html;
        include fastcgi_params;
        fastcgi_split_path_info    ^(.+?\.php)(/.*)$;
        fastcgi_pass    127.0.0.1:9000;
        # fastcgi_index  engine.php;
        fastcgi_index  index.php;
        fastcgi_param  SCRIPT_FILENAME D:/eDokumenty/public_html/$fastcgi_script_name;
        fastcgi_param  PATH_INFO      $fastcgi_path_info;
        include        fastcgi_params;
    }

    location ~ engine {
        include fastcgi_params;

```

```

        fastcgi_split_path_info ^(.+engine)(/.*);$
        fastcgi_index engine;
        fastcgi_param SCRIPT_FILENAME D:/eDokumenty/public_html/$fastcgi_script_name;
        fastcgi_pass 127.0.0.1:9000;
    }

    # deny access to .htaccess files, if Apache's document root
    # concurs with nginx's one
    #
    #location ~ /\.ht {
    #    deny all;
    #}
}

# another virtual host using mix of IP-, name-, and port-based configuration
#
#server {
#    listen      8000;
#    listen      somename:8080;
#    server_name somename alias another.alias;

#    location / {
#        root    html;
#        index   index.html index.htm;
#    }
#}

# HTTPS server
#
#server {
#    listen      443 ssl;
#    server_name localhost;

#    ssl_certificate      cert.pem;
#    ssl_certificate_key  cert.key;

#    ssl_session_cache    shared:SSL:1m;
#    ssl_session_timeout  5m;

#    ssl_ciphers  HIGH:!aNULL:!MD5;
#    ssl_prefer_server_ciphers  on;

#    location / {
#        root    html;
#        index   index.html index.htm;
#    }
#}
}

```

Kolejnym plikiem konfiguracji jest plik fastcgi.conf w tym samym katalogu. Jego zawartość winna być następująca:

```

fastcgi_param SCRIPT_FILENAME    $document_root$fastcgi_script_name;
fastcgi_param QUERY_STRING       $query_string;
# fastcgi_param REQUEST_METHOD    $request_method;

```

```
fastcgi_param CONTENT_TYPE      $content_type;
fastcgi_param CONTENT_LENGTH    $content_length;

fastcgi_param SCRIPT_NAME       $fastcgi_script_name;
fastcgi_param REQUEST_URI       $request_uri;
fastcgi_param DOCUMENT_URI      $document_uri;
fastcgi_param DOCUMENT_ROOT     $document_root;
fastcgi_param SERVER_PROTOCOL   $server_protocol;
fastcgi_param HTTPS             $https if_not_empty;

fastcgi_param GATEWAY_INTERFACE CGI/1.1;
fastcgi_param SERVER_SOFTWARE   nginx/$nginx_version;

fastcgi_param REMOTE_ADDR       $remote_addr;
fastcgi_param REMOTE_PORT       $remote_port;
fastcgi_param SERVER_ADDR       $server_addr;
fastcgi_param SERVER_PORT       $server_port;
fastcgi_param SERVER_NAME       $server_name;

# PHP only, required if PHP was built with --enable-force-cgi-redirect
fastcgi_param REDIRECT_STATUS   200;
```

Konfiguracja PHP jest analogiczna, jak dla Zend-a, z tymże w sekcja Zend powinna wyglądać następująco:

```
[Zend]
zend_extension  = "D:\nginx\ZendLoader.dll"
zend_loader.enable = 1
zend_loader.disable_licencing = 0
```