

# warsztaty: wp-cli zaawansowane

---

- warsztaty: wp-cli zaawansowane
  - Instalacja projektu lokalnie
  - Konfiguracja SSH
  - Ustawienia Direct Admin
  - Przesłanie plików na serwer
  - Uruchomienie domen na serwerze
  - Plik konfiguracyjny wp-cli.yml
    - Polecane ustawienia w `~/.wp-cli/config`
    - Przykładowe ustawienia `wp-cli.yml`
  - Polecenie `wp package`
    - `wp-cli login`
    - `wp cli faker`
    - `wp cli rename db prefix`
    - `wp cli wp-sec`
    - `wp cli db-checkpoint`
  - Polecenie `wp scaffold`
  - Zaawansowane polecenia
  - Skrypty shell

## Instalacja projektu lokalnie

```
wp core download --locale=pl_PL --skip-content
wp config create --dbname=wpcli --dbuser=root --dbpass=root
wp db create

// localhost:8888
wp core install --prompt

wp server
```

## Konfiguracja SSH

### 1. Wygenerowanie klucza SSH

```
cd ~/.ssh

// Unix
ssh-keygen -t ecdsa -b 256 -C "wpcli"
```

### 1. Dodanie klucza publicznego w seohost.pl

```
// Unix
cat ~/.ssh/wpcli.pub | pbcopy

// Windows 10
cat ~/.ssh/id_ecdsa.pub | clip
```

<https://h25.seohost.pl:2222/>

<input type="checkbox"/>	Komentarz	Odcisk	Typ	Rozmiar	Opcje
<input type="checkbox"/>	marcin@devzilla.pl	SHA256:BdGfh1/OKOPy/aVzX5BefXVotC/wUQ/ww+4ZKJmYvxk	ssh-ed25519	256	
<input type="checkbox"/>	marcin@marcin.local	SHA256:yX5CYY24O64ITvQPE8kzgzHrZg3MhCsJASFdtNre3L/Q	ssh-rsa	2048	
<input type="checkbox"/>	Gitlab CI Example	SHA256:zoMT5jN/ObrRvpX+4Lef9LD6GGIMqs22UMNJDVITXoY	ssh-ed25519	256	

```
ssh srv30926@h25.seohost.pl -p 57185 -i ~/.ssh/seohost

ssh-add seohost

ssh srv30926@h25.seohost.pl -p 57185
```

1. Dodanie klucza do ssh-agent (opcjonalnie), ponieważ możemy to sobie obejść przez plik `.ssh/config`

```
// check if ssh agent is running
eval "$(ssh-agent -s)"

// add key to ssh agent
ssh-add ~/.ssh/nameofkey

// verify keys added to ssh-agent
ssh-add -l
```

1. Skonfigurowanie pliku `.ssh/config`

```
Host seohost
  User srv30926
  HostName h25.seohost.pl
  Port 57185
  IdentityFile ~/.ssh/seohost
  PreferredAuthentications publickey
  IdentitiesOnly yes
```

### 1. Przetestowanie czy wszystko działa poprawnie

```
ssh seohost
```

## Ustawienia Direct Admin

- ustawienie subdomeny
  - dev.srv00000.seohost.com.pl
- dodanie 2 baz danych (prod i staging)
  - zapisanie sobie danych do bazy

## Przesłanie plików na serwer

```
// utworzenie kopii bazy danych
wp search-replace lokalna-domena.test srv30926.seohost.com.pl --
export=export.sql

// Unix
zip export.zip export.sql

// windows
Tar -a -cf export.zip export.sql

// utworzenie pliku rsync-exclude.txt
wp-config.php
cgi-bin
.DS_Store
.Thumb
.git

// prod -> Unix
rsync -avzu --progress -h --delete --exclude-from="rsync-exclude.txt" .
seohost:/home/srv30926/domains/srv30926.seohost.com.pl/public_html

// stage
rsync -avzu --progress -h --delete --exclude-from="rsync-exclude.txt" .
seohost:/home/srv30926/domains/dev.srv30926.seohost.com.pl/public_html
```

## Uruchomienie domen na serwerze

- dodanie SSLa
- <https://dev.srv30926.seohost.com.pl/>

```
ssh seohost

cd domains/srv30926.seohost.com.pl/public_html

wp config create --dbname=... --dbuser=... --dbpass=...

wp db import export.sql

rm export.sql

wp user update 1 --user_pass=marcin

// to samo dla domeny dev / stage
```

## Plik konfiguracyjny wp-cli.yml

- Informacje

W przypadku wielokrotnego użycia, WP-CLI może również odczytać opcje z pliku konfiguracyjnego YAML (np. wp-cli.yml). WP-CLI automatycznie wykrywa pliki konfiguracyjne na podstawie reguł zdefiniowanych poniżej. Te pliki konfiguracyjne umożliwiają określenie wartości domyślnych zarówno dla parametrów globalnych, jak i argumentów specyficznych dla podkomendnych.

Argumenty są interpretowane zgodnie z porządkiem pierwszeństwa, od najwyższego priorytetu do najniższego:

1. Argumenty wiersza poleceń
2. `wp-cli.local.yml` - przydatne dla specyficznych ustawień dla dewelopera (dodane do .gitignore)
3. `wp-cli.yml` - ustawienia per projekt (może być trzymane w repo)
4. `~/wp-cli/config.yml` - globalny plik do którego mają dostęp wszystkie instancje / polecenia
5. wartości domyślne

## Polecane ustawienia w `~/wp-cli/config`

```
config create:
dbuser: root
dbpass: root
dbhost: localhost
extra-php: |
  define( 'WP_DEBUG', true );
  define( 'WP_DEBUG_DISPLAY', true );
```

```
define( 'WP_DEBUG_LOG', false );
define( 'DISALLOW_FILE_EDIT', true );

core download:
  locale: pl_PL
  skip-content: true

core install:
  admin_email: marcin@krzeminski.net
  admin_password: marcin
  admin_user: marcin

plugin install:
  activate: true

plugin uninstall:
  deactivate: true

search-replace:
  all-tables: true
  skip-columns: guid
  report-changed-only: true

scaffold child-theme:
  activate: true
  author: Marcin Krzemiński
  author_uri: https://krzeminski.net

search-replace:
  all-tables: true
  skip-columns: guid
  report-changed-only: true

scaffold plugin:
  plugin_author: Marcin Krzemiński
  plugin_author_uri: https://krzeminski.net
  activate: true
  skip-tests: true

scaffold underscores:
  activate: true
  sassify: true
  author: Marcin Krzemiński
  author_uri: https://krzeminski.net
```

### Przykładowe ustawienia `wp-cli.yml`

```
disabled_commands
  - db drop
  - site empty
```

- Połączenie SSH z naszymi stronami

💡 Niestety póki co nie jest możliwe trzymanie w jednym folderze wp-cli.local.yml oraz wp-cli.yml. Jeśli tak to tylko wp-cli.local.yml jest czytany

```
// wp-cli.yml
path: /
disabled_commands:
  - db drop
  - site empty
  - plugin delete
@prod:
  ssh: seohost
  path: /home/srv30926/domains/srv30926.seohost.com.pl/public_html

@stage:
  ssh: seohost
  path: /home/srv30926/domains/dev.srv30926.seohost.com.pl/public_html

@all:
  - @prod
  - @stage
```

Kilka przykładowych zadań

1. Aktualizacja wtyczek
  1. zainstalowanie wtyczki z wybraną wersją
  2. aktualizacja
2. Dodanie konta użytkownika
  1. `wp @prod user create --prompt`
3. Reset hasła
  1. `wp @prod user marcin reset-password`
4. Wyłączenie indeksacji
  1. `wp @dev option set blog_public 0`

## Polecenie `wp package`

<http://wp-cli.org/package-index/>

💡 Niestety na hostingu musimy poprosić o instalację jeśli chcemy wykorzystywać te polecenia zdalnie

wp-cli login

Logowanie z poziomu wp-cli

- <https://github.com/aaemnnosttv/wp-cli-login-command>

```
wp package install aaemnnosttv/wp-cli-login-command
```

```
wp login install --activate

// tworzy automatyczny link, który wygasa po 15 minutach lub po użyciu
załogowania
wp login create marcin
wp login as marcin

// tworzy automatyczny link i wysyła go na email uzytkownika
// localhost:8025 - mailhog
wp login email marcin

// automatycznie otwiera dashboard w domyślnej przeglądarce
// wp login as -> alias dla wp login create
wp login as marcin --launch
```

## wp cli faker

Generuje testowane dane dla WordPressa oraz WooCommerce

- <https://github.com/Yoast/wp-cli-faker>

```
wp package install yoast/wp-cli-faker

wp faker core content

wp plugin install woocommerce perfect-woocommerce-brands --activate

wp faker woocommerce products

wp site empty
```

## wp cli rename db prefix

Zmiana prefiksu tabel

- aktualizuje także plik `wp-config.php`, jeśli mamy niestandardowe ustawienia to możemy użyć flagi `--no-config-update`

```
wp package install iandunn/wp-cli-rename-db-prefix

wp config get table_prefix

wp rename-db-prefix n5a_

wp config get table_prefix
```

## wp cli wp-sec

Sprawdza naszą instalację w oparciu o bazę [wpvulndb.com] (<http://wpvulndb.com>)

token: Q7LRWyJTop8n50Ss8F0PLH1wfsh7BwnF3GkITnBLPhQ

```
wp package install https://github.com/markri/wp-sec/archive/refs/heads/master.zip

wp wp-sec check --token=Q7LRWyJTop8n50Ss8F0PLH1wfsh7BwnF3GkITnBLPhQ
```

## wp cli db-checkpoint

Twórz i przywracaj snapshoty bazy danych

- <https://github.com/binarygary/db-checkpoint>

```
wp package install binarygary/db-checkpoint

wp dbsnap
wp dbsnapback

wp dbsnap NAME
wp dbsnapback NAME
```

## Polecenie wp scaffold

```
// child theme
wp theme install kadence
wp scaffold child-theme kadence-child --parent_theme=kadence

// plugin
wp scaffold plugin gliwice --plugin_name=Gliwice

// underscores -> Node 12, 14 lub 16
wp scaffold underscores wpcli --sassify
cd wp-content/themes/wpcli
npm i
npx browser-sync start --proxy 'scaffold.test' --files '**/*'
npm run watch

// post type
wp scaffold post-type product --plugin=gliwice

// taxonomy
wp scaffold taxonomy product-category --post_types=product --plugin=gliwice

// block
```



```
wp scaffold block slider --plugin=gliwice
```

## Zaawansowane polecenia

```
// czyści bazę danych, NIE usuwa plików z uploads, opcji oraz użytkowników
wp site empty --yes

// import mediów
wp media import media/misc/*
wp media import media/images/*.jpg

// "poproś wyszukiwarki o nieindeksowanie"
wp option set blog_public 0

// usuwamy wszystkie komentarze oznaczone jako spam
wp comment delete $(wp comment list --status=spam --format=ids)

// usuwamy wszystkich uzytkownikow z wyjatkiem ID = 1
wp user delete $(wp user list --field=ID --exclude=1) --reassign=1

// instalujemy wtyczke w konkretnej wersji
wp plugin update contact-form-7 --version=5.0.5

// instalujemy motyw w konkretnej wersji
wp theme update astra --version=3.7.6

// reinstalujemy WP do konkretnej wersji
wp core download --version=6.1 --force

// duplikujemy post
wp post create --from-post=1

// instalowanie ulubionych wtyczek z pliku
wp plugin install $(cat plugins.txt)

// listowanie plikow wtyczki lub motywu
ls -al $(dirname $(wp plugin path contact-form-7))
ls -al $(dirname $(wp theme path astra))
```

### Przykładowy plik `plugins.txt`

```
contact-form-7 --version=5.7.4
duplicate-post
antispam-bee
litespeed-cache
seo-by-rank-math
insert-headers-and-footers
sierotki
```

```
simple-page-ordering  
google-site-kit  
file-changes-monitor
```

## Skrypty shell

```
sh bin/backup.sh  
sh bin/cc.sh  
  
// przy pierwszym deploy importuje takze users, zmieniamy haslo na  
serwerze i wylaczamy tabele users  
sh bin/deploy.sh  
  
sh bin/import.sh  
sh bin/info.sh  
sh bin/install.sh  
sh bin/update.sh
```