



WP-CLI

L'interfaccia a riga di comando per WordPress



Maurizio Pelizzone

Daddy

PHP Senior Developer

WordPress Solutions Architect

Speaker, trainer and consultant

WordPress proud user

maurizio@mavida.com

<http://www.mavida.com>

<http://maurizio.mavida.com>

<https://twitter.com/miziomon>

<http://www.slideshare.net/miziomon>

<http://www.linkedin.com/in/mauriziopelizzone>



alla memoria di Paolo Wolly Valenti



Avvertenza #1: Talk noioso ad alto contenuto nerd

(ci sono 50 slide...)



“WP-CLI è un insieme di strumenti da linea di comando per la gestione delle installazioni WordPress. Potete aggiornare plugin, configurare installazioni multisito e molto altro, senza utilizzare un browser web.”

<https://wp-cli.org/it/>



“L’obiettivo di WP-CLI è quello di fornire un’interfaccia da linea di comando per le azioni che normalmente si possono effettuare attraverso l’area amministrativa”



<https://wp-cli.org/it/>



“WP-CLI include anche dei comandi per molte operazioni che normalmente non sarebbe possibile svolgere nell’area amministrativa di WordPress. “



<https://wp-cli.org/it/>



The command line interface for WordPress

Italiano

Commands

Configuration

Handbook

Blog

Contributing



[WP-CLI](#) è un insieme di strumenti da linea di comando per la gestione delle installazioni [WordPress](#). Potete aggiornare plugin, configurare installazioni multisito e molto altro, senza utilizzare un browser web.

Per rimanere aggiornati, seguite [@wpcli su Twitter](#) oppure [iscrivetevi alla nostra newsletter](#).



Testing

passing

issue resolution

1 d

open issues

3%

Collegamenti rapidi: [Utilizzo](#) | [Installazione](#) | [Supporto](#) | [Estendere](#) | [Contribuire](#) | [Crediti](#)

Utilizzo

L'obiettivo di WP-CLI è quello di fornire un'interfaccia da linea di comando per le azioni che normalmente si possono effettuare attraverso l'area amministrativa. Per esempio, `wp plugin install --activate` ([doc](#)) vi permette di installare ed attivare un plugin di WordPress:





[Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

 

 [wp-cli / wp-cli](#) Public

[Watch](#) 175 [Fork](#) 948 [Starred](#) 4.6k

[Code](#) [Issues](#) 182 [Pull requests](#) 13 [Discussions](#) [Actions](#) [Wiki](#) [Security](#) 1 [Insights](#)

[master](#) 3 branches 64 tags [Go to file](#) [Add file](#) [Code](#)

 **schlessera** Bump VERSION to 2.8.0-alpha ✓ 581c954 5 days ago 🕒 9,912 commits

📁 .github	Update file(s) from wp-cli/github	6 days ago
📁 bin	Update credits	14 months ago
📁 features	Move WP_CLI_CONFIG_SPEC_FILTER_CALLBACK test to flags.feature	12 days ago
📁 php	Merge pull request #5637 from johnbillion/docker-run	12 days ago
📁 templates	Allow user to suppress global parameters info from help display	2 years ago
📁 tests	Merge branch 'master' into docker-run	4 months ago
📁 utils	Add testing GitHub Actions workflow	2 years ago
📄 .actrc	Update file(s) from wp-cli/github	2 years ago
📄 .editorconfig	Revert "Remove WP-CLI files unrelated to wp package *"	6 years ago
📄 .gitattributes	Use str_replace instead of preg_replace to normalize EOLs of doc comm...	5 years ago
📄 .gitignore	Remove Composer.lock file from repository	11 months ago
📄 .mailmap	Revert "Remove WP-CLI files unrelated to wp package *"	6 years ago
📄 CONTRIBUTING.md	Use a shortened CONTRIBUTING blurb in README, and point to handbo...	5 years ago
📄 LICENSE	Updated license date	2 years ago
📄 README.md	Update README.md file	6 days ago
📄 VERSION	Bump VERSION to 2.8.0-alpha	5 days ago
📄 behat.yml	Update Behat configuration file	2 years ago
📄 composer.json	Update branch alias to v2.8.x	12 days ago

About

⚙ WP-CLI framework

[wp-cli.org/](#)

[cli](#) [php](#) [wordpress](#) [wp-cli](#)

[framework](#) [hacktoberfest](#)

📖 Readme

📄 MIT license

★ 4.6k stars

👁 175 watching

🍴 948 forks

Releases

 49

📦 **Release 2.7.1** Latest
6 days ago

[+ 48 releases](#)

Packages

No packages published

Used by

 3.8k

         + 3,831



Possiamo fidarci?



Sì



Version 0.3

by [scribu](#), on 06 Nov 2011

Version 0.3.0 is now available.

Changelog

Commands:

- >> added `wp sql`
- >> improved `wp option`
- >> improved `wp plugin` and `wp theme`
- >> added `wp generate`
- >> added `wp core version`

Internals:

- >> added `--version` special parameter
- >> added `--blog` global parameter
- >> added bash completion script
- >> added pear installer

Contributors to this release: [andreascreten](#), [scribu](#)

27 Mar 2014

 scribu

 v0.12.0

 a9b9c2d

Compare 

Version 0.12.0

Release notes: <http://wp-cli.org/blog/version-0.12.html>

▼ Assets 3

 wp-cli-0.12.0.phar	654 KB	27 Mar 2014
 Source code (zip)		04 Oct 2013
 Source code (tar.gz)		04 Oct 2013



 9,912 commits

6 days ago

 schlessera

 v2.7.1

 1ddc754 

Compare 

Release 2.7.1 Latest

See <https://make.wordpress.org/cli/2022/10/18/wp-cli-v2-7-1-release-notes/>

▼ Assets 8

 wp-cli-2.7.1.phar	6.42 MB	6 days ago
 wp-cli-2.7.1.phar.asc	516 Bytes	6 days ago
 wp-cli-2.7.1.phar.gpg	1.51 MB	6 days ago
 wp-cli-2.7.1.phar.md5	33 Bytes	6 days ago
 wp-cli-2.7.1.phar.sha256	65 Bytes	6 days ago
 wp-cli-2.7.1.phar.sha512	129 Bytes	6 days ago
 Source code (zip)		6 days ago
 Source code (tar.gz)		6 days ago

  2 2 people reacted



**Ma alla fine, cosa posso farci
con sta roba?**



wp admin	Open /wp-admin/ in a browser.	wp option	Retrieves and sets site options, including plugin and WordPress settings.
wp cache	Adds, removes, fetches, and flushes the WP Object Cache object.	wp package	Lists, installs, and removes WP-CLI packages.
wp cap	Adds, removes, and lists capabilities of a user role.	wp plugin	Manages plugins, including installs, activations, and updates.
wp cli	Reviews current WP-CLI info, checks for updates, or views defined aliases.	wp post	Manages posts, content, and meta.
wp comment	Creates, updates, deletes, and moderates comments.	wp post-type	Retrieves details on the site's registered post types.
wp config	Generates and reads the wp-config.php file.	wp profile	
wp core	Downloads, installs, updates, and manages a WordPress installation.	wp rewrite	Lists or flushes the site's rewrite rules, updates the permalink structure.
wp cron	Tests, runs, and deletes WP-Cron events; manages WP-Cron schedules.	wp role	Manages user roles, including creating new roles and resetting to defaults.
wp db	Performs basic database operations using credentials stored in wp-config.php.	wp scaffold	Generates code for post types, taxonomies, plugins, child themes, etc.
wp dist-archive	Create a distribution archive based on a project's .distignore file.	wp search-replace	Searches/replaces strings in the database.
wp embed	Inspects oEmbed providers, clears embed cache, and more.	wp server	Launches PHP's built-in web server for a specific WordPress installation.
wp eval	Executes arbitrary PHP code.	wp shell	Opens an interactive PHP console for running and testing PHP code.
wp eval-file	Loads and executes a PHP file.	wp sidebar	Lists registered sidebars.
wp export	Exports WordPress content to a WXR file.	wp site	Creates, deletes, empties, moderates, and lists one or more sites on a multisite installation.
wp find	Find WordPress installations on the filesystem.	wp super-admin	Lists, adds, or removes super admin users on a multisite installation.
wp help	Gets help on WP-CLI, or on a specific command.	wp taxonomy	Retrieves information about registered taxonomies.
wp i18n	Provides internationalization tools for WordPress projects.	wp term	Manages taxonomy terms and term meta, with create, delete, and list commands.
wp import	Imports content from a given WXR file.	wp theme	Manages themes, including installs, activations, and updates.
wp language	Installs, activates, and manages language packs.	wp transient	Adds, gets, and deletes entries in the WordPress Transient Cache.
wp maintenance-mode	Activates, deactivates or checks the status of the maintenance mode of a site.	wp user	Manages users, along with their roles, capabilities, and meta.
wp media	Imports files as attachments, regenerates thumbnails, or lists registered image sizes.	wp widget	Manages widgets, including adding and moving them within sidebars.
wp menu	Lists, creates, assigns, and deletes the active theme's navigation menus.		
wp network	Perform network-wide operations.		



wp db <command>

Performs basic database operations using credentials stored in wp-config.php.

[View Open Issues \(1\)](#)[View Closed Issues \(97\)](#)[Create New Issue](#)

EXAMPLES

```
# Create a new database.
$ wp db create
Success: Database created.

# Drop an existing database.
$ wp db drop --yes
Success: Database dropped.

# Reset the current database.
$ wp db reset --yes
Success: Database reset.

# Execute a SQL query stored in a file.
$ wp db query < debug.sql
```

[Top ↑](#)

<https://developer.wordpress.org/cli/commands/db/>



Qualche esempio?



wp core update

Updates WordPress to a newer version.



[View Open Issues \(7\)](#)

[View Closed Issues \(29\)](#)

[Create New Issue](#)

Defaults to updating WordPress to the latest version.

If you see “Error: Another update is currently in progress.”, you may need to run `wp option delete core_updater.lock` after verifying another update isn’t actually running.

OPTIONS

[<zip>]

Path to zip file to use, instead of downloading from wordpress.org.

[--minor]

Only perform updates for minor releases (e.g. update from WP 4.3 to 4.3.3 instead of 4.4.2).

[--version=<version>]

Update to a specific version, instead of to the latest version. Alternatively accepts ‘nightly’.

[--force]

Update even when installed WP version is greater than the requested version.

[--locale=<locale>]

Select which language you want to download.

[--insecure]

Retry download without certificate validation if TLS handshake fails. Note: This makes the request vulnerable to a MITM attack.



wp plugin install

Installs one or more plugins.



[View Open Issues \(4\)](#)

[View Closed Issues \(43\)](#)

[Create New Issue](#)

OPTIONS

`<plugin|zip|url>...`

One or more plugins to install. Accepts a plugin slug, the path to a local zip file, or a URL to a remote zip file.

`[--version=<version>]`

If set, get that particular version from wordpress.org, instead of the stable version.

`[--force]`

If set, the command will overwrite any installed version of the plugin, without prompting for confirmation.

`[--activate]`

If set, the plugin will be activated immediately after install.

`[--activate-network]`

If set, the plugin will be network activated immediately after install

`[--insecure]`

Retry downloads without certificate validation if TLS handshake fails. Note: This makes the request vulnerable to a MITM attack.



wp plugin activate

Activates one or more plugins.



[View Open Issues \(1\)](#)

[View Closed Issues \(10\)](#)

[Create New Issue](#)

OPTIONS

[<plugin>...]

One or more plugins to activate.

[--all]

If set, all plugins will be activated.

[--exclude=<name>]

Comma separated list of plugin slugs to be excluded from activation.

[--network]

If set, the plugin will be activated for the entire multisite network.

[Top ↑](#)



wp plugin update

Updates one or more plugins.



[View Open Issues \(5\)](#)

[View Closed Issues \(28\)](#)

[Create New Issue](#)

OPTIONS

[<plugin>...]

One or more plugins to update.

[--all]

If set, all plugins that have updates will be updated.

[--exclude=<name>]

Comma separated list of plugin names that should be excluded from updating.

[--minor]

Only perform updates for minor releases (e.g. from 1.3 to 1.4 instead of 2.0)

[--patch]

Only perform updates for patch releases (e.g. from 1.3 to 1.3.3 instead of 1.4)

[--format=<format>]

Render output in a particular format.

default: table

options:

- table

- csv

- json

- summary



wp theme install

Installs one or more themes.



[View Open Issues \(3\)](#)

[View Closed Issues \(20\)](#)

[Create New Issue](#)

OPTIONS

`<theme|zip|url>...`

One or more themes to install. Accepts a theme slug, the path to a local zip file, or a URL to a remote zip file.

`[--version=<version>]`

If set, get that particular version from wordpress.org, instead of the stable version.

`[--force]`

If set, the command will overwrite any installed version of the theme, without prompting for confirmation.

`[--activate]`

If set, the theme will be activated immediately after install.

`[--insecure]`

Retry downloads without certificate validation if TLS handshake fails. Note: This makes the request vulnerable to a MITM attack.



wp search-replace

Searches/replaces strings in the database.



[View Open Issues \(13\)](#)

[View Closed Issues \(126\)](#)

[Create New Issue](#)

Searches through all rows in a selection of tables and replaces appearances of the first string with the second string.

By default, the command uses tables registered to the `$wpdb` object. On multisite, this will just be the tables for the current site unless `--network` is specified.

Search/replace intelligently handles PHP serialized data, and does not change primary key values.

OPTIONS

`<old>`

A string to search for within the database.

`<new>`

Replace instances of the first string with this new string.

`[<table>...]`

List of database tables to restrict the replacement to. Wildcards are supported, e.g. `'wp_*options'` or `'wp_post*'`.

`[--dry-run]`

Run the entire search/replace operation and show report, but don't save changes to the database.



wp media <command>

Imports files as attachments, regenerates thumbnails, or lists registered image sizes.

[View Open Issues \(1\)](#)[View Closed Issues \(67\)](#)[Create New Issue](#)

EXAMPLES

```
# Re-generate all thumbnails, without confirmation.
$ wp media regenerate --yes
Found 3 images to regenerate.
1/3 Regenerated thumbnails for "Sydney Harbor Bridge" (ID 760).
2/3 Regenerated thumbnails for "Boardwalk" (ID 757).
3/3 Regenerated thumbnails for "Sunburst Over River" (ID 756).
Success: Regenerated 3 of 3 images.

# Import a local image and set it to be the featured image for a post.
$ wp media import ~/Downloads/image.png --post_id=123 --title="A downloaded picture" --featured_image
Success: Imported file '/home/person/Downloads/image.png' as attachment ID 1753 and attached to post 123 as

# List all registered image sizes
$ wp media image-size
```

name	width	height	crop
full			N/A
twentyfourteen-full-width	1038	576	hard
large	1024	1024	soft
medium_large	768	0	soft
medium	300	300	soft
thumbnail	150	150	hard



wp post delete

Deletes an existing post.

[View Open Issues \(0\)](#)[View Closed Issues \(5\)](#)[Create New Issue](#)

OPTIONS

<id>...

One or more IDs of posts to delete.

[--force]

Skip the trash bin.

[--defer-term-counting]

Recalculate term count in batch, for a performance boost.

[Top ↑](#)

EXAMPLES

```
# Delete post skipping trash
$ wp post delete 123 --force
Success: Deleted post 123.

# Delete all pages
$ wp post delete $(wp post list --post_type='page' --format=ids)
Success: Trashed post 1164.
Success: Trashed post 1186.

# Delete all posts in the trash
$ wp post delete $(wp post list --post_status=trash --format=ids)
Success: Deleted post 1268.
Success: Deleted post 1294.
```



wp db reset

Removes all tables from the database.



[View Open Issues \(0\)](#)

[View Closed Issues \(5\)](#)

[Create New Issue](#)

Runs `DROP_DATABASE` and `CREATE_DATABASE` SQL statements using `DB_HOST`, `DB_NAME`, `DB_USER` and `DB_PASSWORD` database credentials specified in `wp-config.php`.

OPTIONS

`[--dbuser=<value>]`

Username to pass to mysql. Defaults to `DB_USER`.

`[--dbpass=<value>]`

Password to pass to mysql. Defaults to `DB_PASSWORD`.

`[--yes]`

Answer yes to the confirmation message.

`[--defaults]`

Loads the environment's MySQL option files. Default behavior is to skip loading them to avoid failures due to misconfiguration.



wp user create

Creates a new user.



[View Open Issues \(2\)](#)

[View Closed Issues \(21\)](#)

[Create New Issue](#)

OPTIONS

`<user-login>`

The login of the user to create.

`<user-email>`

The email address of the user to create.

`[--role=<role>]`

The role of the user to create. Default: default role. Possible values include 'administrator', 'editor', 'author', 'contributor', 'subscriber'.

`[--user_pass=<password>]`

The user password. Default: randomly generated.

`[--user_registered=<yyyy-mm-dd-hh-ii-ss>]`

The date the user registered. Default: current date.

`[--display_name=<name>]`

The display name.

`[--user_nicename=<nice_name>]`

A string that contains a URL-friendly name for the user. The default is the user's username.



Automatizzare installazione standard



```
1.  #!/bin/bash
2.
3.  wp core download
4.  wp core config --dbhost=host --dbname=prefix --dbuser=username
    --dbpass=password
5.  wp core install --url=tuosito.it --title="Il mio sito"
    --admin_name=admin_26J8 --admin_password=L716YpRCPc
    --admin_email=mail@gmail.com
6.  wp theme install blocksy --activate
7.  wp scaffold child-theme tuosito --parent_theme=blocksy
8.  wp plugin install seo-by-rank-math --activate
9.  wp plugin install jetpack --activate
10. wp user create utente_cliente mail_cliente@gmail.com --role=author
11. curl -N https://loripsum.net/api/5 | wp post generate --post_content
    --count=10
12. curl -N https://loripsum.net/api/5 | wp post generate --post_content
    --post_type=page --count=10
```



Avvertenza #2:
Siete ancora in tempo ad
uscire...



Creare comandi custom



1. `<?php`

2.

3. `if (defined("WP_CLI") && WP_CLI) {`

4. `WP_CLI::add_command( 'pippo', function() {`

5. `WP_CLI::success( 'comando ok');`

6. `});`

7. `}`

8.



1. `<?php`

2.

3. `if (defined("WP_CLI") && WP_CLI) {`

4. `WP_CLI::add_command("pippo", "ClassePippo");`

5. `}`

6.

7.

8.



```
1. <?php
```

```
2. class ClassePippo{
```

```
3.     public function __invoke($args, $params) {
```

```
4.         WP_CLI::success("comando ok");
```

```
5.         // faccio roba
```

```
6.         WP_CLI::success("fine ");
```

```
7.     }
```

```
8. }
```

```
9.
```



Come lo eseguo?



wp pippo



**Posso aggiungere
argomenti e parametri ?**



Posso accedere alle funzioni di WordPress ?



**Posso usare altri comandi
wp-cli dentro il mio codice ?**



Sì



Cosa ci posso fare?



1. importazione magazzino / listini / promozioni
2. esportazione ordini su gestionale aziendale
3. invio delle leads su CRM esterni
4. sincronizzazione documenti area riservata



Vantaggi



#automazione
#procedureperiodiche
#evitareerroreumano
#notimeout



Come lo installo se il mio servizio di hosting non lo supporta?



```
1. # curl -O https://raw.githubusercontent.com/wp-cli/builds/gh-pages/phar/wp-cli.phar
2. # php wp-cli.phar --info
3. # chmod +x wp-cli.phar
4. # sudo mv wp-cli.phar /usr/local/bin/wp
5. # wp --info
6.
```



**“Io non posso insegnare niente a
nessuno, io posso solo farli
pensare.”**



Socrate



That's all Folks!



Domande?



Grazie!

Maurizio Pelizzone

@miziomon

maurizio@mavida.com

<http://maurizio.mavida.com>



Mavida S.n.c. -

Via Vandalino, 142 - 10142 Torino TO

+39 011.322.362 info@mavida.com -

www.mavida.com

