

Composer for Frontend Developers

If you are a web designer or working on the CSS on your website, this page contains is all you need to know about Composer.

Composer is an opensource PHP packaging system to manage dependencies.

This makes it easy to adapt package installs and updates to your workflow, allowing you to test new/updated packages in a development environment, put the changes in your version control system (git, Subversion, Mercurial, etc.), pull in those changes on a staging environment and, when approved, put it in production.

Troubleshooting

You may experience some latency in dependency resolution: everything is going normally.

If you are interested by the process, do your Composer commands with the `--verbose` option activated.

Option `verbose -v`

Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug.

Usage:

```
php -d memory_limit=-1 composer.phar <command> --verbose  
(-v|vv|vvv)
```

Useful commands

install

The `install` command reads the `composer.lock` file from the current directory, processes it, and downloads and installs all the libraries and dependencies outlined in that file. If the file does not exist it will look for `composer.json` and do the same.

Usage

```
php -d memory_limit=-1 composer.phar install --dry-run  
--prefer-dist
```

In this example the `dry-run` option is mentioned to prevent you from doing anything critical. (This option outputs the operations but will not execute anything and implicitly enables the verbose mode).

Documentation with complete usage:

```
php -d memory_limit=-1 composer.phar install
[--prefer-source] [--prefer-dist] [--dry-run] [--dev]
[--no-dev] [--no-plugins] [--no-custom-installers]
[--no-scripts] [--no-progress] [-v|vv|vvv|--verbose]
[-o|--optimize-autoloader] [packages1] ... [packagesN]
```

update

The `update` command reads the `composer.json` file from the current directory, processes it, and updates, removes or installs all the dependencies.

Interesting options:

To limit the update operation to a few packages, you can list the package(s) you want to update as such:

```
php -d memory_limit=-1 composer.phar update
vendor/package1 foo/mypackage
```

You may also use an asterisk (*) pattern to limit the update operation to package(s) from a specific vendor:

```
php -d memory_limit=-1 composer.phar update
vendor/package1 foo/*
```