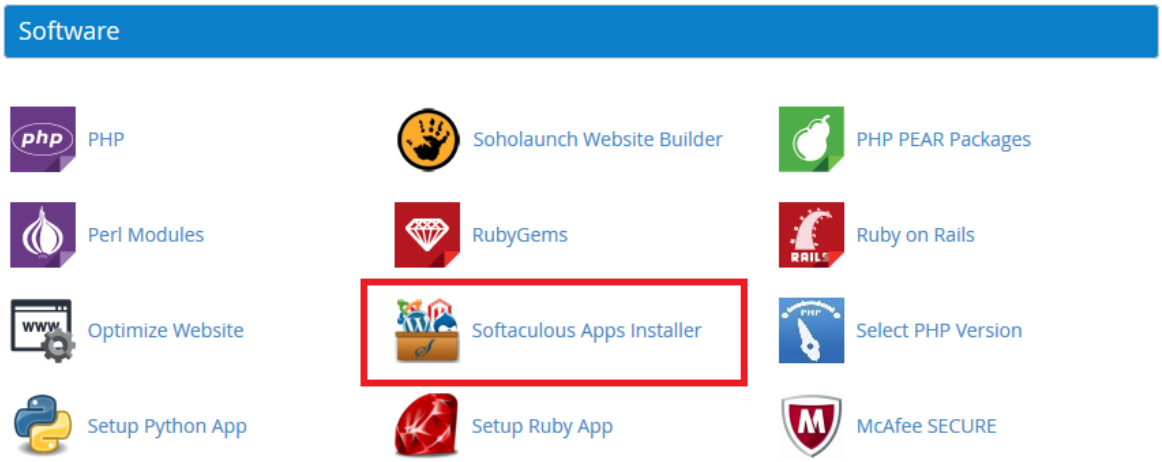
Install Laravel (and Composer) on a cPanel Server

Before we begin, I suppose your first question is "What is Laravel?" **Laravel** is a free, open source PHP web application framework, designed for the development of MVC web applications. Laravel is built on top of several Symfony components, giving your application a great foundation of well-tested and reliable code.

The Easy Way:
As a GlowHost customer, you can always install Laravel in just one click. It can be found in your GlowHost control panel when you click on the Softaculous Icon.

1.Navigate to Softaculous in cPanel:

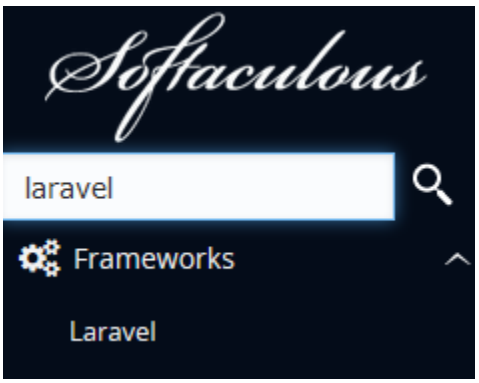
Paper lantern theme:



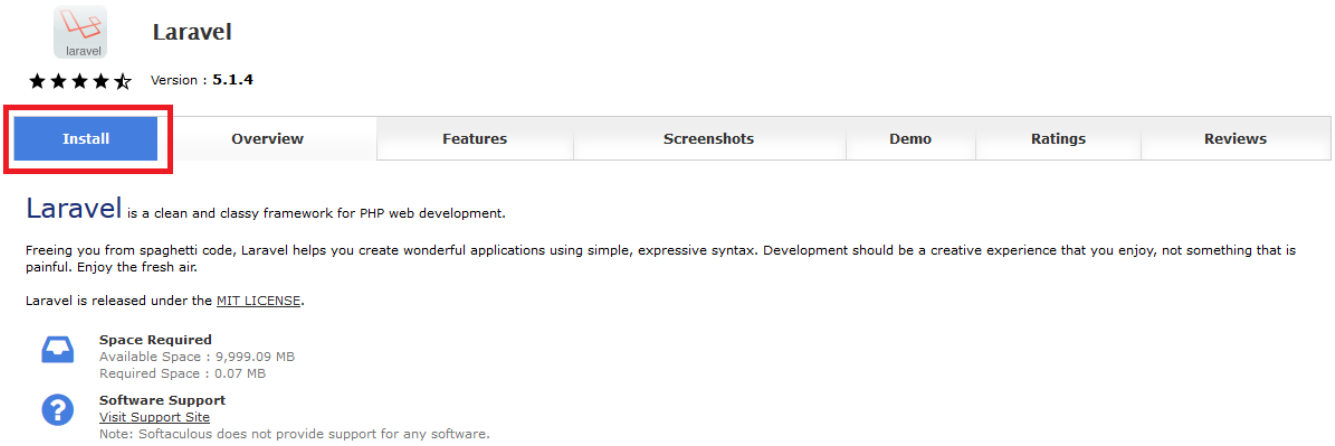
x3 Theme:



2.Search for Laravel in the search box and click on the result



3.Click install



Done!

Manual install

We'll show you how to install it manually for those of you who are interested in what's happening behind the scenes or want to learn how to install this without an auto-installer.

1. You will need SSH access. Please contact us and we will enable normal shell for your account.

2. Install Composer (it is a dependency manager for PHP, and required for Laravel). Please note, you can always ask our support to do this. If you decide to perform installation on your own, please use these commands:

a)

```
     user@server [~]# cd bin/
user@server [~/bin]# wget https://getcomposer.org/installer
user@server [~/bin]# php installer --check
All settings correct for using Composer
user@server [~/bin]# php installer
All settings correct for using Composer
Downloading...

Composer successfully installed to: /home/user/bin/composer.phar
Use it: php composer.phar
user@server [~/bin]# rm -f installer
```

Composer's site, getcomposer.org, suggests to run this command: "curl -sS https://getcomposer.org/installer | php", but no professional would execute a script without checking it first.

A Small Tip #1:

Add the following lines to ~/.bashrc to run the commands as "composer" instead of the full command of "php -d disable_functions= ~/bin /composer.phar":

```
      alias php="php -d disable_functions="
alias composer="php ~/bin/composer.phar"
```

> ⚠ "-d disable_functions=" is very important parameter here, because composer requires some disabled functions (like escapeshellarg or proc_open).

b) Install your first project:

```
      user@server [~/public_html]# mkdir project/
user@server [~/public_html]# cd project/
```

create composer.json file with the following code:

```
       {
"require": {
"monolog/monolog": "1.0.*"
}
}
```

and run:

```
      user@server [~/public_html/project]# composer install
Loading composer repositories with package information
Installing dependencies (including require-dev)
- Installing monolog/monolog (1.0.2)
Downloading: 100%

Writing lock file
Generating autoload files
```

Great, everything is working! Now you can remove your first project blocked URL

3. Install Laravel:

```
      user@server [~]# mkdir project
user@server [~]# cd project/
user@server [~/project]#


user@server [~/project]# composer create-project laravel/laravel --prefer-dist
```

Important: for some reasons laravel installation requires 390Mb of RAM (and this is not mentioned on their site), so if you are planning to install Laravel, please consider our Professional Hosting Package, since it will satisfy all the requirements.

Don't worry if you get the following error:

```
     Script php artisan optimize handling the post-install-cmd event returned with an error

[RuntimeException]
Error Output:

  [Symfony\Component\Process\Exception\RuntimeException]
  The Process class relies on proc_open, which is not available on your PHP installation.

optimize [--force] [--psr]
```

That is because the post-install-cmd scripts are running from a parent process and without our settings (see tip #1). All you need to do to fix the problem is to manually run post-install-cmd scripts (view composer.json file in laravel/ folder).

Here is an example:

```
      user@server [~/project/laravel]# php artisan clear-compiled
user@server [~/project/laravel]# php artisan optimize
Generating optimized class loader
Compiling common classes
```

Same thing when you are running the update for Laravel.

A Small Tip #2:

Hosting with Glowhost you can set custom PHP version in your cPanel ("*Select php version*" area) where you can enable "proc_open" function for your whole account. If you changed PHP version, please make sure that **Phar** and **Json** extentions are enabled.

An Important Tip #3 (special for cPanel servers):

Now we need to remove the public_html folder and make a symbolic link to public folder in our project (please be sure that the public_html folder is empty or you have a fresh backup copy of any data you may have there):

```
     user@server[~]# cd ~
user@server[~]# rm -rf public_html/
user@server[~]# ln -s project/laravel/public/ public_html
```

CHECK THE SITE:

blocked URL

Some configuration or paths may be different. We only show you an example of installation. Each case is unique.

Related articles

- Install Laravel (and Composer) on a cPanel Server
- Composer installation
- Composer usage example