



## CHAPTERS

[Installing Laravel \[6th Edition\]](#)[Building Our First Website \[6th Edition\]](#)[Building A Support Ticket System \[6th Edition\]](#)[Building A Blog Application \[6th Edition\]](#)[Deploying Our Laravel Applications \[6th Edition\]](#)

**Important:** The book now supports Laravel 5.7 and Bootstrap 4. This chapter is complete. We will be updating all chapters regularly to fix bugs and mistakes.

# Chapter 1 - Installing Laravel

There are many ways to install Laravel. We can install Laravel directly on our main machine, or we can use all-in-one server stacks such as MAMP, XAMPP, etc. We have a huge selection of ways to choose.

In this book, I will show you the most popular one: [Laravel Homestead](#).

## Introducing CLI (Command Line Interface)

If you haven't heard about CLI, Terminal or Git, this section is for you. If you know how to use the CLI already, you may skip this section.

Working with Laravel requires a lot of interactions with the CLI, thus you will need to know how to use it.

### CLI for MAC OSX

Luckily, on Mac, you can find a good CLI called **Terminal** at [/Applications/Utilities](#).

Most of what you do in the **Terminal** is enter specific text strings, then press **Return** to execute them.

Alternatively, you can use [iTerm 2](#).

### CLI for Windows

Unfortunately, the default CLI for Windows (cmd.exe) is not good, you may need another one.

The most popular one called **Git Bash**. You can download and install it here:

<http://msysgit.github.io>

Most of what you do in **Git Bash** is enter specific text strings, then press **Enter** to execute them.

Alternatively, you may use [Cygwin](#).

### CLI for Linux

On Linux, the CLI is called **Terminal** or **Konsole**. If you know how to install and use Linux, I guess you've known how to use the CLI already.

## Installing Laravel Using Homestead

### What is Homestead?

Nowadays, many developers are using a virtual machine (VM) to develop dynamic websites and applications. You can run a web server, a database server and all your scripts on that virtual machine. You can create many VM instances and work on various projects. If you don't want any VM anymore, you can safely delete it without affecting anything. You can even re-create the VM in minutes!

We call this: "Virtualization."

There are many options for virtualization, but the most popular one is VirtualBox from Oracle. VirtualBox will help us to install and run many virtual machines as we like on our Windows, Mac, Linux or Solaris operating systems. After that, we will use a tool called Vagrant to manage and configure our virtual development environments.

In 2014, Taylor Otwell - the creator of Laravel - has introduced Homestead.

Homestead is a Vagrant based Virtual Machine (VM) and it is based on Ubuntu. It includes everything we need to start developing Laravel applications. That means, when we install Homestead, we have a virtual server that has PHP, Nginx, databases and other packages. We can start creating our Laravel application right away.

Here is a list of included software:

- Ubuntu 18.04
- Git
- PHP 7.2
- PHP 7.1

- PHP 7.0
- PHP 5.6
- Nginx
- Apache (Optional)
- MySQL
- MariaDB (Optional)
- Sqlite3
- PostgreSQL
- Composer
- Node (With Yarn, Bower, Grunt, and Gulp)
- Redis
- Memcached
- Beanstalkd
- Mailhog
- Neo4j (Optional)
- MongoDB (Optional)
- Elasticsearch (Optional)
- ngrok
- wp-cli
- Zend Z-Ray
- Go
- Minio

You may check out the **Homestead Documentation** at:

<https://laravel.com/docs/master/homestead>

## How to install Homestead?

In May 2015, the Laravel official documentation has been updated. The recommended way to install Homestead is using Git.

There are three steps to install Homestead using this method.

- **Step 1:** Install VirtualBox
- **Step 2:** Install Vagrant
- **Step 3:** Install Homestead

Let's start by installing VirtualBox and Vagrant first.

### Step 1 - Installing VirtualBox

First, we need to go to:

<https://www.virtualbox.org/wiki/Downloads>

Choose a VirtualBox for your platform and install it.

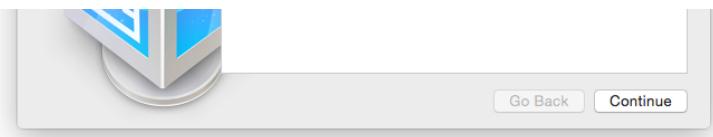
Make sure that you download the correct version for your operating system.

The **stable release is version 5.2.18**. You can use a newer version if you want, but if you have any problems, try to use this version.

If you're using Windows, double click the **.exe** setup file to install VirtualBox.

If you're using Mac, simply open the VirtualBox **.dmg** file and click on the **.pkg** file to install.





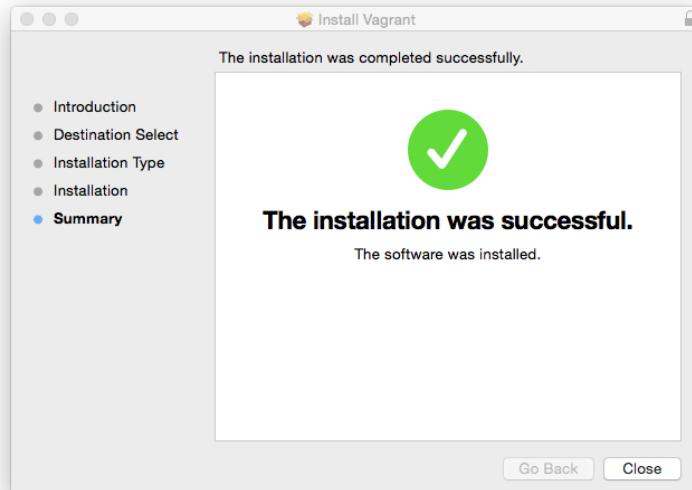
## Step 2 - Installing Vagrant

The next step is to install Vagrant. Please go to:

<http://www.vagrantup.com/downloads.html>

**Note:** The stable release is **version 2.1.4**, which can be found at <https://releases.hashicorp.com/vagrant/2.1.4/>. You may use a newer version, but if you encounter any error, try to **reinstall version 2.1.4**.

If you're using Mac, download the **.dmg** file -> Open the **downloaded file** -> Click on the **Vagrant.pkg** file to install it.



If you still don't know how to install, there is an official guide on Vagrant website:

<http://docs.vagrantup.com/v2/installation>

## Step 3 - Install Homestead (Using Git Clone)

You can install Homestead just by **cloning the Homestead Repository**.

You will need to install **Git** first if you don't have it on your system.

**Note:** if you don't know how to run a command, please read **Introducing CLI (Command Line Interface)** section.

### Install Git on Mac

The easiest way is to install the **Xcode Command Line Tools**. You can do this by simply running this command:

```
1 | xcode-select --install
```

Click **Install** to download **Command Line Tools** package.

Alternatively, you can also find the **OSX Git installer** at this website:

<http://git-scm.com/download/mac>

### Install Git on Windows

You can download **GitHub for Windows** to install Git:

<https://windows.github.com>

### Install Git on Linux/Unix

You can install **Git** by running this command:

```
1 | sudo yum install git
```

If you're on a **Debian-based** distribution, use this:

```
1 | sudo apt-get install git
```

For more information and other methods, you can see this guide:

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

When you have **Git** installed. Enter the following command to your **Terminal** (or **Git Bash**):

```
1 git clone https://github.com/laravel/homestead.git Homestead
```

```
~ git clone https://github.com/laravel/homestead.git Homestead
Cloning into 'Homestead'...
remote: Counting objects: 875, done.
remote: Total 875 (delta 0), reused 0 (delta 0), pack-reused 875
Receiving objects: 100% (875/875), 131.31 KiB | 94.00 KiB/s, done.
Resolving deltas: 100% (504/504), done.
Checking connectivity... done.
```

Once downloaded, go to the **Homestead** directory by using **cd** command:

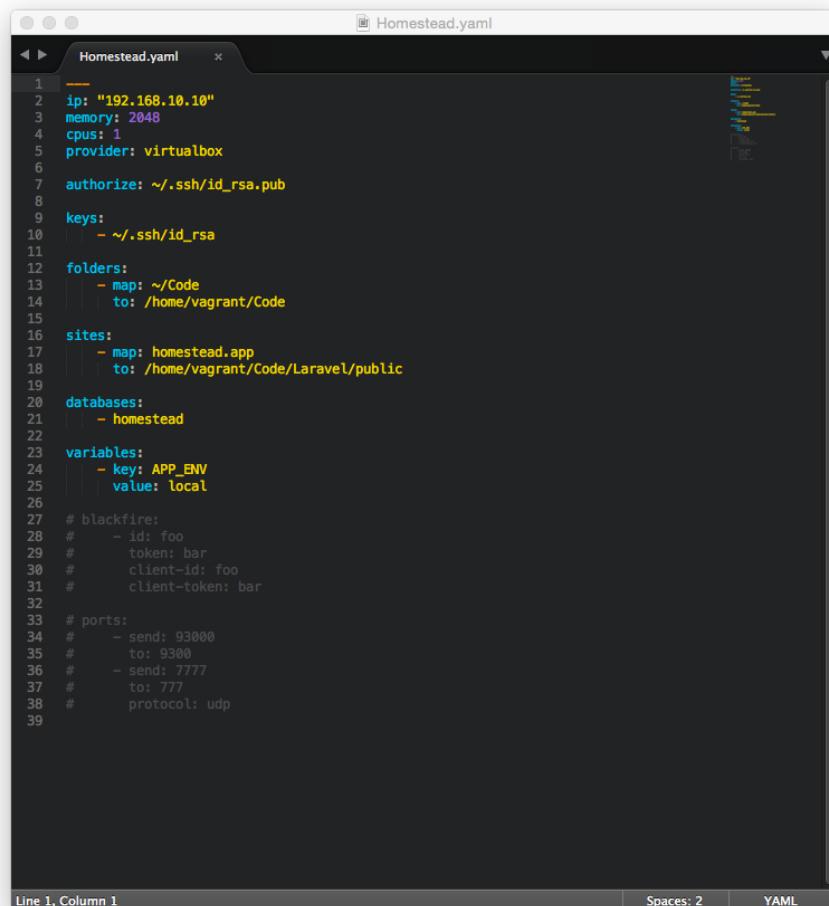
```
1 cd Homestead
```

Run this command to create **Homestead.yaml** file:

```
1 bash init.sh
```

The **Homestead.yaml** file will be placed in the **Homestead** directory. Open it with a text editor to edit it.

**Note:** In older versions of Homestead, the **Homestead.yaml** file will be placed in your **~/.homestead** directory. Please note that the **~/.homestead** directory is **hidden** by default, make sure that you can see hidden files.



If you know how to use **VI** or **VIM**, use this command to edit the file:

```
1 vi ~/Homestead/Homestead.yaml
```

Alternatively, you can use this command to open the file with your text editor:

```
1 open ~/Homestead/Homestead.yaml
```

**Note:** Your system path may be different. Try to find `Homestead.yaml`.

## Configure Homestead

The structure of the `Homestead.yaml` is simple. There are 7 sections. Let's see what they do.

### First section - Configure VM

```
1 ip: "192.168.10.10"
2 memory: 2048
3 cpus: 1
4 provider: virtualbox
```

As you can see, we can configure the IP address, memory, cpus and provider of our VM. This section is not important, so we can just leave it as it is.

### Second and third section - Configure SSH

```
1 authorize: ~/.ssh/id_rsa.pub
2
3 keys:
4   - ~/.ssh/id_rsa
```

Basically, we need to generate an **SSH key** for Homestead to authenticate the user and connect to the VM. If you're working with **Git**, you may have an **SSH key** already. If you don't have it, simply run this command to generate it:

```
1 ssh-keygen -t rsa -C "you@homestead"
```

The command will generate an **SSH key** for you and put it in the `~/.ssh` directory automatically, you don't need to do anything else.

### Fourth section - Configure shared folder

We use `folders` section to specify the directory that we want to share with our Homestead environment. If we add, edit or change any files on our local machine, the files will be updated automatically on our Homestead VM.

```
1 folders:
2   - map: ~/Code
3     to: /home/vagrant/Code
```

We can see that the `~/Code` directory has been put there by default. This is where we put all the files, scripts on our local machine. Feel free to change the link if you want to put your codes elsewhere.

**Note:** If you're using Windows, you may need to use a full path. For example: `- map: C:\Users\YourUSERNAME\Documents\Projects\Laravel\Homestead\Code`

Please note that everytime you change the path, you have to run these commands to reload and update Homestead:

```
1 vagrant halt
2 vagrant up --provision
```

The `/home/vagrant/Code` is a path to the `Code` directory on our VM. Usually, we don't need to change it.

### Fifth section - Map a domain

```
1 sites:
2   - map: homestead.test
3     to: /home/vagrant/Code/Laravel/public
```

This section allows us to map a domain to a folder on our VM. For example, we can map `homestead.test` to the `public` folder of our Laravel project, and then we can easily access our Laravel app via this address: "<http://homestead.test>".

**Note:** You can use the `.app` extension if you like (For example, <http://homestead.app>). However, if you're using the latest version of Google Chrome, you should change the `.app` extension to `.test` extension.

Remember that, when we add any domain, we must edit the `hosts` file on our local machine to redirect requests to our Homestead environment.

On Linux or Mac, you can find the `hosts` file at `/etc/hosts` or `/private/etc/hosts`. You can edit the `hosts` file using this command:

```
1 | sudo open /etc/hosts
```

If you know how to use **VI** or **VIM**, use this command to edit the file:

```
1 | sudo vim /etc/hosts
```

On Windows, you can find the `hosts` file at `C:\Windows\System32\drivers\etc\hosts`.

After opening the file, you need to add this line at the end of the file:

```
1 | 192.168.10.10 homestead.test
```

Done! When we launch Homestead, we can access the site via this address.

<http://homestead.test>

Please note that we can change the address (`homestead.test`) to whatever we like.

All sites will be accessible by HTTP via port `8000` and HTTPS via port `44300` by default (Homestead port).

## Sixth section - Configure database

```
1 | databases:
2 |   - homestead
```

This is the database name of our VM. As usual, we just leave it as it is.

## Seventh section - Add custom variables

```
1 | variables:
2 |   - key: APP_ENV
3 |     value: local
```

If we want to add some custom variables to our VM, we can add them here. It's not important, so let's move to the next fun part.

## Launching Homestead

Once we have edited `Homestead.yaml` file, `cd` to the `Homestead` directory, run this command to boot our virtual machine:

```
1 | vagrant up
```

It may take a few minutes...

If you see this error:

```
Bringing machine 'default' up with 'virtualbox' provider...
There are errors in the configuration of this machine. Please fix
the following errors and try again:
vm:
* The host path of the shared folder is missing: ~/Code
```

It means that you don't have `Code` directory on your main machine. You can create one, or change the link to any folder that you like.

Executing this command to create a new `Code` folder:

```
1 | sudo mkdir ~/Code
```

To prevent possible errors when creating Laravel, try to **set right permissions** for the `Code` folder by running:

```
1 | chmod -R 0777 ~/Code
```

If everything is going fine, we should see:

```
Bringing machine 'default' up with 'virtualbox' provider...
--> default: Importing base box 'laravel/homestead'...
--> default: Matching MAC address for NAT networking...
--> default: Checking if box 'laravel/homestead' is up to date...
--> default: Setting the name of the VM: homestead
--> default: Fixed port collision for 22 => 2222. Now on port 2200.
--> default: Clearing any previously set network interfaces...
--> default: Preparing network interfaces based on configuration...
--> default: Adapter 1: nat
```

```
default: Adapter 2: hostonly
=> default: Forwarding ports...
```

Now we can access our VM using:

```
1 vagrant ssh
```

```
Welcome to Ubuntu 14.10 (GNU/Linux 3.16.0-23-generic x86_64)

 * Documentation: https://help.ubuntu.com/
Last login: Sun May 31 13:27:51 2015 from 10.0.2.2
vagrant@homestead:~$
```

**Note:** If it asks for a password, type `vagrant`.

To make sure that everything is ok, run `ls` command:

```
Homestead [master] vagrant ssh
Welcome to Ubuntu 14.10 (GNU/Linux 3.16.0-23-generic x86_64)

 * Documentation: https://help.ubuntu.com/
Last login: Mon Oct 27 02:22:37 2014 from 10.0.2.2
vagrant@homestead:~$ ls
Code
```

If you can see the `Code` directory there, you have Homestead installed correctly!

Excellent! Let's start installing Laravel!

## Installing Laravel

When you have installed Homestead, create a new Laravel app is so easy!

As I've mentioned before, the `Code` directory is where we will put our Laravel apps. Let's go there!

```
1 cd Code
```

You should notice that the directory is empty. There are two methods to install Laravel.

### Install Laravel Via Laravel Installer

This method is recommended. It's newer and faster. You should use this method to create your Laravel application.

First, we need to use **Composer** to download the **Laravel installer**.

```
1 composer global require "laravel/installer"
```

```
vagrant@homestead:~$ ls
Code
vagrant@homestead:~$ cd Code
vagrant@homestead:~/Code$ composer global require "laravel/installer=~1.1"
Changed current directory to /home/vagrant/.composer
./composer.json has been updated
Loading composer repositories with package information
Updating dependencies (including require-dev)
- Installing guzzlehttp/streams (2.1.0)
  Downloading: 100%
- Installing guzzlehttp/guzzle (4.2.3)
  Downloading: 100%
- Installing laravel/installer (v1.2.0)
  Downloading: 100%
Writing lock file
Generating autoload files
vagrant@homestead:~/Code$
```

Once downloaded, you can create a new Laravel project by using this command:

```
1 laravel new nameOfYourSite
```

**Laravel Installer** will download the **latest Laravel version** and install it. To install a **specific Laravel version**, you may use this command instead:

```
1 laravel new nameOfYourSite --5.7
```

This command is used to download **Laravel 5.7**.

**Note:** If the latest version of Laravel is **5.7**, you can't run this command. You may install Laravel 5.6 instead by using the `--5.6` flag. It is recommended to use Laravel 5.7 to learn the basics of Laravel Framework. You can upgrade to a newer version later.

You're free to change the `nameOfYourSite` to whatever you like, but remember to edit the `sites` section of `Homestead.yaml` to match your site's name.

For instance, in `Homestead.yaml`, we specify the name of our app is `Laravel`

```
1 | sites:
2 |   - map: homestead.test
3 |     to: /home/vagrant/Code/Laravel/public
```

We will need to run this command to create a new `Laravel` site

```
1 | laravel new Laravel
```

You should see this:

```
vagrant@homestead:~/Code$ laravel new Laravel
Crafting application...
Generating optimized class loader
Compiling common classes
Application key [r0F37Lf0ER06izLB19iv0afECgfM1rM3] set successfully.
Application ready! Build something amazing.
vagrant@homestead:~/Code$
```

If you see this error when using the `laravel new` command:

```
1 | laravel: command not found
```

We have to edit the `.bashrc` file, type:

```
1 | nano ~/.bashrc
```

Add this line at end of the file:

```
1 | alias laravel='~/config/composer/vendor/bin/laravel'
```

Press `Ctrl + X`, then `Y`, then `Enter` to **exit and save** the file.

Lastly, run this command:

```
1 | source ~/.bashrc
```

Now you should be able to create a new Laravel app using:

```
1 | laravel new Laravel
```

You should see this:

```
Generating optimized class loader
You are running composer with xdebug enabled. This has a major impact on runtime performance. See https://getcomposer.org/xdebug
> php artisan key:generate
Application key [base64:okNyej8EqSVRrq+IoSwmYiCv7drYvnJaur4Y0YlP4=] set successfully.
Application ready! Build something amazing.
```

Next, open your web browser and go to <http://homestead.test>

# Laravel

[DOCUMENTATION](#) [LARACASTS](#) [NEWS](#) [FORGE](#) [GITHUB](#)

Congratulations! You've installed Laravel! It's time to create something amazing!

Install Laravel using composer create-project

If you don't like to use **Laravel Installer**, or you have any problems with it, feel free to use `composer create-project` to create a new Laravel app:

```
1 | composer create-project laravel/laravel nameOfYourSite "~5.7.7"
```

This command is used to download **Laravel 5.7.7**, which is a stable version. If you want to use the latest version, use:

```
1 | composer create-project laravel/laravel nameOfYourSite
```

**Note:** It is recommended to use Laravel 5.7 to learn the basics of Laravel Framework. You can upgrade to a newer version later.

You're free to change the **nameOfYourSite** to whatever you like, but remember to edit the **sites** section of **Homestead.yaml** to match your site's name.

For instance, in **Homestead.yaml**, we specify the name of our app is **Laravel**

```
1 | sites:
2 |   - map: homestead.test
3 |     to: /home/vagrant/Code/Laravel/public
```

We will need to run this command to create **Laravel** site

```
1 | composer create-project laravel/laravel Laravel
```

Alternatively, we can create a new **Laravel** folder, **cd** to it, and create our Laravel app there:

```
1 | mkdir Laravel
2 | cd Laravel
3 | composer create-project laravel/laravel --prefer-dist
```

You should see this:

```
Writing lock file
Generating autoload files
> Illuminate\Foundation\ComposerScripts::postUpdate
> php artisan optimize
Generating optimized class loader
> php artisan key:generate
Application key [base64:BMFMK0vcEp0homAtj+uR5xwBLyR1IZf52Zo2HNn0rhM=] set successfully.
```

Open your web browser, go to <http://homestead.test>

# Laravel

DOCUMENTATION

LARACASTS

NEWS

FORGE

GITHUB

**Note:** if you cannot access the site, try to add the port into the URL: <http://homestead.test:8000>.

Congratulations! You've installed Laravel! It's time to create something amazing!

## Checking Laravel version

We can check what version of Laravel that we've installed by simply running this command **at the root of our application**:

```
1 | php artisan --version
```

or

```
1 | php artisan -v
```

A line will be printed out:

```
1 | Laravel Framework 5.7.7
```

As you see, I'm using **Laravel 5.7.7**.

**Important note:** If you're not using Laravel 5.7, please download and install Laravel 5.7 to avoid possible errors.

## Updating Homestead box

Sometimes, when we run the `vagrant up` command, we might see this message:

```
1 | ==> default: Checking if box 'laravel/homestead' is up to date...
2 | ==> default: A newer version of the box 'laravel/homestead' is available! You currently
3 | ==> default: have version '0.4.4'. The latest is version '2.1.0'. Run
4 | ==> default: `vagrant box update` to update.
```

As you may have noticed, this means a newer version of Homestead box is available. We can update the box by running this command:

```
1 | vagrant box update
```

You'll see something like this:

```
1 | ==> default: Updating 'laravel/homestead' with provider 'vmware_desktop' from version
2 | ==> default: '0.4.4' to '2.1.0'...
3 | ==> default: Loading metadata for box 'https://atlas.hashicorp.com/laravel/homestead'
4 | ==> default: Adding box 'laravel/homestead' (v2.1.0) for provider: vmware_desktop
5 |     default: Downloading: https://atlas.hashicorp.com/laravel/boxes/homestead/versions/2.1.0/providers/vmware_desktop.box
6 |     default: Progress: 64% (Rate: 863k/s, Estimated time remaining: 0:06:47)
```

**Note:** Be sure to backup your files and databases first. It may take a long time to complete.

After that, `vagrant ssh` into your Homestead and run:

```
1 | sudo apt-get update
2 | sudo apt-get upgrade
```

If it asks anything, type `y`.

Please note that the **master branch** (the latest version) may not always be stable. If your app is running fine, you don't have to update Homestead.

## Updating Homestead using Git

If you want to upgrade your Homestead to the latest version or use another version of Homestead, you may use this method.

First, please backup the **Homestead.yaml** file and your database first to ensure that you won't lose any data.

Go to your homestead **root directory**:

```
1 cd Homestead
```

**Note:** Your path could be different.

Next, run this command to get a list of Homestead versions:

```
1 git fetch origin
```

You'll see something like this:

```
1 remote: Counting objects: 19, done.
2 remote: Compressing objects: 100% (3/3), done.
3 remote: Total 19 (delta 11), reused 12 (delta 11), pack-reused 5
4 Unpacking objects: 100% (19/19), done.
5 From https://github.com/laravel/homestead
6     393c4bd..74749a5  master      -> origin/master
7 * [new tag]      v7.2.0      -> v7.2.0
8 * [new tag]      v7.3.0      -> v7.3.0
```

After that, pick the version that you like.

```
1 git checkout v7.12.0
```

Currently, the stable version is `v7.12.0`.

Run these commands to destroy and update your vagrant:

```
1 vagrant destroy
2 rm -rf .vagrant
3 vagrant up
```

**Note:** You might need to download and install the latest version of Vagrant.

Done! You're good to go!

**Note:** In this book, we'll be using Homestead `v7.12.0` (which is a stable version), so please use the same version to avoid possible errors.

## Generate new application key

Sometimes, you might see this error when generating a new application:

```
1 "No application encryption key has been specified."
```

To fix this, you just need to update the Laravel Installer by running this command:

```
1 composer global require "laravel/installer"
```

After that, simply generate a new **application key**:

```
1 php artisan key:generate
```

This should fix the bug.

6 Comments   Learning Laravel

 Login ▾

 Recommend 5    Tweet    Share

Sort by Best ▾



Join the discussion...

LOG IN WITH



OR SIGN UP WITH DISQUS 



Name

**bernard robert** • a year ago

Can you share the installation on window machine?

1 ^ | v • Reply • Share >

**Rui M.** • 16 days ago

Hi, have problems in

Running first Laravel Mix task. Doesn't understand how to use the steps described. I'm very lost about. Have windows 10 installed. and all npm listed doesn't work with me.

any idea?

Thanks

^ | v • Reply • Share >

**NinpouSensha** • 4 months ago

Very helpfull thank you

^ | v • Reply • Share >

**Aleksandr** • 5 months ago

Hello! I'm getting Error: [InvalidArgumentException]

Could not find package laravel/laravel with version ~5.7.7.

And when I'm trying

composer create-project laravel/laravel --prefer-dist

Installing laravel/laravel (v5.4.30)

the version is not latest

^ | v • Reply • Share >

**Bruno Matthy** • 10 months ago

As usual, windows users are left out in the cold.

folders:

- map: ~/Code

to: /home/vagrant/Code

1) This will never work - you need to add the complete path to map e.g.

C:\Users\MYUSERNAME\Documents\Projects\Laravel\Homestead\Code

2) The Code folder is not auto-generated in the Vagrant box when you run it without this - it's just a calvary trip to get this working on Windows for newbies. After changing the path to the full location run this:

vagrant halt

vagrant up --provision

Many books are like this - Vagrant is left untouched and people who are stuck will have to rely on Google to hope for a solution. too bad this one isn't any different.

^ | v • Reply • Share >

**learninglaravel** Mod → Bruno Matthy • 9 months ago

Thank you for your suggestion. We will update the book soon. Have a nice day!

2 ^ | v • Reply • Share >

[Subscribe](#)

[Add Disqus to your site](#)

[Disqus' Privacy Policy](#)

**DISQUS**



Students and Teachers, save up to 60% on Adobe Creative Cloud ads via Carbon