

АйТиРевью

Блог об IT



МЕНЮ

Установка и работа с Laravel через Vagrant

Фреймворк Laravel предоставляет свою Linux-сборку под названием Homestead ([текущая документация](#) и [страница на vagrantup.com](#)) для разработчиков со всеми необходимыми компонентами и даже с небольшим запасом. Эта сборка вполне подходит и для других фреймворков.

В текущую версию Laravel Homestead входят следующие пакеты:

- Ubuntu 18.04;
- Git;
- PHP 7.3;
- PHP 7.2;
- PHP 7.1;
- PHP 7.0;
- PHP 5.6;
- Nginx;
- MySQL;
- Imm for MySQL or MariaDB database snapshots;
- Sqlite3;
- PostgreSQL;
- Composer;

- Node (With Yarn, Bower, Grunt, and Gulp);
- Redis;
- Memcached;
- Beanstalkd;
- Mailhog;
- avahi;
- ngrok;
- Xdebug;
- XHProf / Tideways / XHGui;
- wp-cli.

Для использования Homestead-дистрибутива вам нужна основа – виртуальная машина ([VirtualBox](#), [VMWare](#), [Parallels](#) и т.д.) и надстройка над ней для упрощения работы – [Vagrant](#).

В документации к [Laravel Homestead](#) есть подробная инструкция, но она может показаться сложной для новичка. В этой инструкции описаны два основных подхода для работы с проектами на Laravel: 1 – генерировать для каждого проекта отдельный Homestead и 2 – установить одну версию Homestead для всех проектов сразу.

Давайте сразу перейдем к делу. Для начала добавьте box Homestead’a в ваш Vagrant вот такой командой:

```
vagrant box add laravel/homestead
```

Вариант №1

(использование отдельного Homestead для каждого проекта)

Этот вариант часто используется для разработки сайтов на Laravel под заказ и передачи вместе с файлами конфигурации Vagrant’a, т.е. мы генерируем окружение и файлы конфигурации этого окружения передаем заказчику вместе с исходниками. Заказчик на своём компьютере может сразу же запустить вашу конфигурацию с помощью одной команды – “vagrant up”. Схема работы выглядит следующим образом: **один** Homestead обеспечивает работу **одного** сайта.

Пошаговая настройка (все команды запускаются из корневой директории проекта):

1. Установка Laravel через composer:

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

2. Установка дополнительного пакета Laravel для генерации файлов конфигурации Homestead:

```
composer require laravel/homestead --dev
```

Рекомендую использовать флаг “--dev” для того, чтобы не устанавливать этот пакет на “боевом” сервере. Да и все остальные пакеты используемые только для разработки следует помечать этим флагом.

3. Генерация конфигурационных файлов Vagrant:

Для Windows

```
vendor\bin\homestead make
```

или:

Для Linux

```
php vendor/bin/homestead make
```

4. Настройка файла конфигурации Homestead – Homestead.yaml

Вот пример одного моего сайта для Windows:

../Homestead.yaml

```
ip: 192.168.10.11
memory: 2048
cpus: 1
provider: virtualbox
authorize: c:/users/yourusername/.ssh/mykey.pub
keys:
  - c:/users/yourusername/.ssh/myprivatekey.ppk
folders:
```

```
-  
    map: 'd:\projects\site\src'  
    to: /home/vagrant/code  
sites:  
    -  
        map: site.loc  
        to: /home/vagrant/code/public  
databases:  
    - homestead  
name: homestead  
hostname: 127.0.0.1
```

Приватный и публичный ключи для Windows необходимо сгенерировать с помощью PuttyGen или аналогичной программы и поместить в корневую директорию вашей учетной записи.

5. Добавление пары “IP” и “site.domain” в hosts-файл

Откройте файл с хостами:

Для Windows

c:\windows\system32\drivers\etc\hosts

или

Для Linux

/etc/hosts

И добавьте в него адрес нового сайта:

192.168.10.11 site.loc

6. Запуск Vagrant:

vagrant up

Вариант №2

(использование Homestead для множества проектов сразу)

Данный вариант предназначен для централизации управления проектами на своём компьютере. Если вы ведете множество проектов и не хотите настраивать Homestead под каждый из них в отдельности, то этот вариант вам точно подойдет. Схема работы выглядит следующим образом: **один** Homestead обеспечивает работу **множества** сайтов.

Пошаговая настройка (все команды запускаются из корневой директории проекта):

1. Создайте директорию новую директорию для Homestead и перейдите в неё в командной строке:

```
cd ПУТЬ_ДО_ДИРЕКТОРИИ
```

2. Склонируйте официальный репозиторий Homestead с github в созданную папку:

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

3. Выберите актуальную ветку репозитория (для разных версий Laravel'а она отличаться, в текущем примере используется Laravel 6.x):

```
git checkout v10.1.0
```

4. Запустите скрипт инициализации Homestead:

Linux/Mac CMD

```
bash init.sh
```

или

Windows

```
init.bat
```

5. Настройте файл конфигурации Homestead – Homestead.yaml: укажите используемую виртуальную машину (в моем случае “provider: virtualbox”), ключи авторизации, список сайтов и другие параметры. Пример:

Файл конфигурации: Homestead.yaml

```
ip: "192.168.10.15"
memory: 2048
cpus: 1
provider: virtualbox

authorize: c:/users/username/.ssh/mykey.pub

keys:
  - c:/users/username/.ssh/myprivatekey.ppk

folders:
  - map: 'd:\projects\site1'
    to: /home/vagrant/site1

  - map: 'd:\projects\site2'
    to: /home/vagrant/site2

sites:
  - map: site1.loc
    to: /home/vagrant/site1

  - map: site2.loc
    to: /home/vagrant/site2

databases:
  - site1_db
  - site2_db
```

В этой конфигурации созданы два сайта – “http://site1.loc” и “http://site2.loc”, которые будут работать на вашей основной системе (в моём случае Windows 10). Для того, чтобы эти сайты стали доступны, вам необходимо прописать в файл hosts настройки для каждого, вот так:

Файл .hosts:

```
192.168.10.15 site1.loc
192.168.10.15 site2.loc
```

К сожалению, на русском языке нет документации для текущей версии Laravel – 6, поэтому в “полезные ссылки” помещу ссылки на старые версии. Рекомендую для разработчиков изучающих Laravel читать документацию с официального сайта на английском языке.

Полезные ссылки:

- [Официальная документация – установка Laravel 6.](#)
- [Официальная документация – Homestead.](#)
- [Документация по Homestead на русском \(версия 5.4\).](#)
- [Еще одна документация по использованию Homestead на русском \(какая версия не ясно, для 5.x\).](#)
- [Как создать ключ для авторизации по SSH и добавить его на сервер?](#)
- [Как найти и отредактировать файл Hosts.](#)
- [IDE-Helper для Laravel – github.com/barryvdh/laravel-ide-helper.](#)

Возможно, Вам Будет Интересно:

1. **Установка PHP на Windows**
2. **Laravel IDE Helper Generator**
3. **Windows WSL (Windows Subsystem for Linux)**
4. **PHP PhantomJS**

20.01.2020

Категория(-и): Laravel, PHP, Vagrant

#Homestead #hosts #Laravel #Linux #PHP #Ubuntu #Vagrant #Windows

Добавить комментарий

Для отправки комментария вам необходимо авторизоваться.

ITReviewChannel.RU © 2019-2022