# En MVC-Applikation

Jag tänker gå igenom hur vi utvecklar en liten MVC-applikation med hjälp av Composer i Homestead.

## Domämnamn

Först behöver vi uppdatera filen C:\Windows\System32\drivers\etc\hosts med vårt nya projekt, genom att starta notebook som administratör.

På Mac OS använder vi sudo och vi:

sudo vi \private\etc\hosts

eller

sudo vi etc\hosts

I Vi går vi med piltangenterna till den position där vi vill lägga till en ny rad. Då trycker vi tangenten i (insert) och skriver den nya rader. Därefter Escape och :wq (write and quit).

hosts

# Copyright (c) 1993-2009 Microsoft Corp.

#

# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.

#

# This file contains the mappings of IP addresses to host names. Each

# entry should be kept on an individual line. The IP address should

# be placed in the first column followed by the corresponding host name.

# The IP address and the host name should be separated by at least one

# space.

#

# Additionally, comments (such as these) may be inserted on individual

# lines or following the machine name denoted by a '#' symbol.

#

# For example:

#

# 102.54.94.97 rhino.acme.com # source server

# 38.25.63.10 x.acme.com # x client host

# localhost name resolution is handled within DNS itself.

# 127.0.0.1 localhost

# ::1 localhost

192.168.10.10 homestead.test

**192.168.10.10 test.mvc**

Därefter öppnar vi filen i Homestead.yaml i vår Homestead-katalog och lägger till sökvägen till vår domän.

Homestead.yaml

ip: "192.168.10.10"

memory: 2048

cpus: 2

provider: virtualbox

folders:

- map: C:\Users\Stefan Bjornander\Documents\homestead\code

to: /home/vagrant/code

sites:

- map: homestead.test

to: /home/vagrant/code/public

**- map: test.mvc**

**to: /home/vagrant/code/public/test\_mvc**

databases:

- homestead

features:

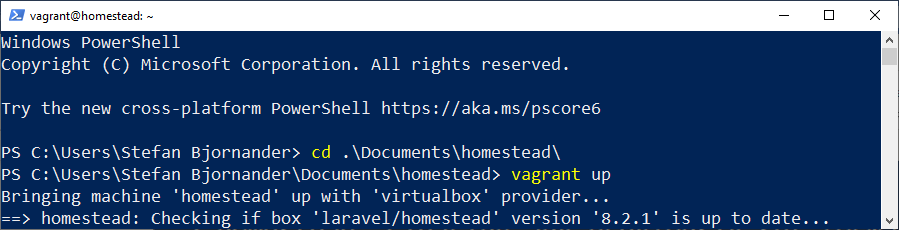
- mariadb: false

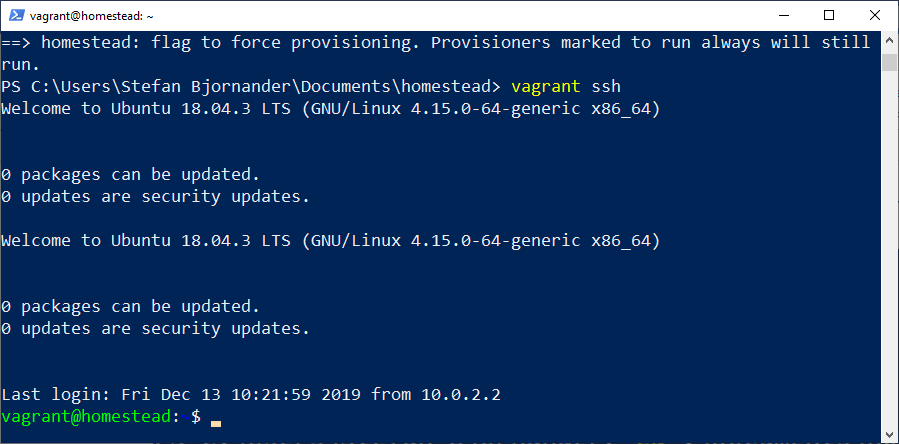
- ohmyzsh: false

- webdriver: false

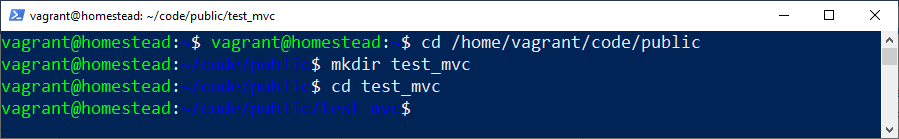
## PowerShell

Vi öppnar ett fönster i PowerShell i Windows eller bash i Mac OS.

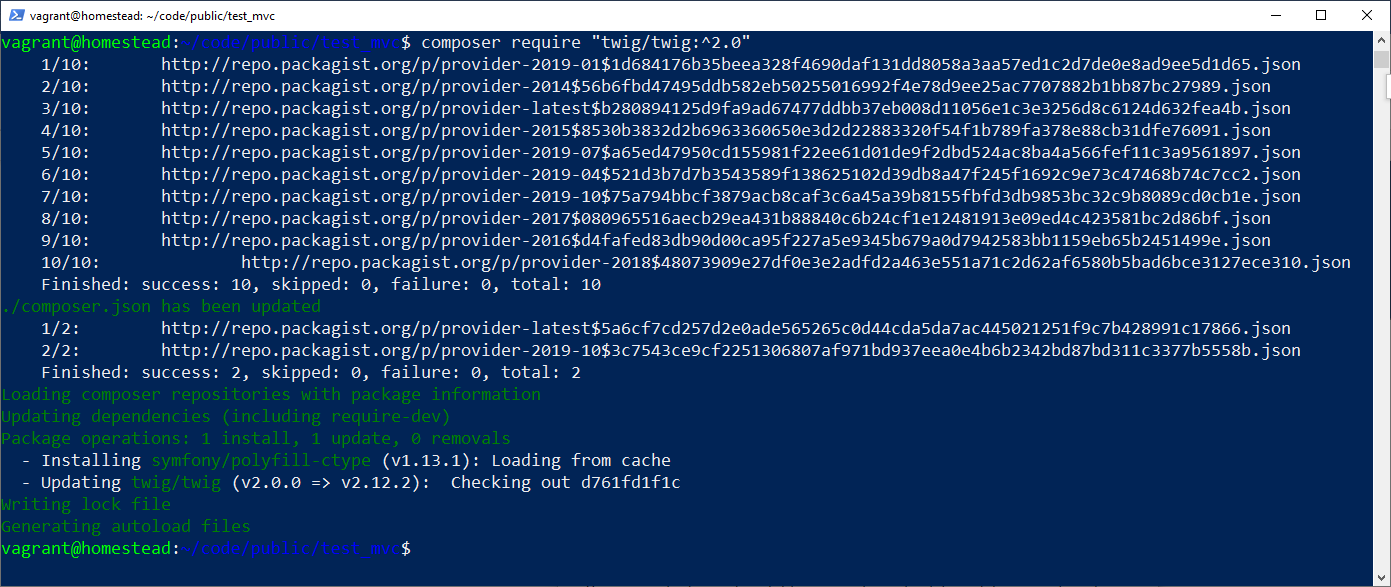




Vi går till vår public-katalog, i mitt fall /home/vagrant/code/public. Därefter skapar en ny katalog för vårt projekt, som jag kallar test\_mvc, och går ni i den katalogen.



Vi laddar in bibliotekskod för Twig med hjälp av Composer.

Detta har bland annat genererat filen composer.json.

Composer.json

{

"require": {

"twig/twig": "^2.0"

}

}

Vi lägger till några rader för att autoloading skall fungera.

{

"require": {

"twig/twig": "^2.0"

},

**"autoload": {**

**"psr-4": {**

**"Main\\": "src"**

**}**

**}**

}

Nu är det dags att skriva vår första källkodsfil i PHP: Index.php.

Index.php

<?php

use Main\Router;

use Main\Request;

require\_once \_\_DIR\_\_ . "/vendor/autoload.php";

$loader = new Twig\_Loader\_Filesystem(\_\_DIR\_\_ . "/views");

$view = new Twig\_Environment($loader);

$request = new Request();

$router = new Router();

echo $router->route(new Request());

?>

Vår request-klass är ganska enkel.

Request.php

Stefan Björnander

Chas Academy AB