

November 30, 2015 / comment 0

## Installing WordPress with Homebrew on OS X Yosemite

I'm a Mac/OS X user and I like Homebrew. So when I wanted to install WordPress locally for the first time in years, I didn't want to install some all-in-one MAMP package that might be difficult to administer and uninstall. Here's what I did to get WordPress running.

Basically, I followed this guide: <https://echo.co/blog/os-x-1010-yosemite-local-development-environment-apache-php-and-mysql-homebrew>. Here's a summary of the steps:

1. Install MySQL via Homebrew.
2. Configure MySQL, in particular my.cnf.
3. Add MySQL to brew services so that it starts on boot via launchd.
4. Remove OS X's built-in Apache from launchd's list.
5. Add the homebrew-dupes Homebrew repository to satisfy Apache's zlib dependency.
6. Install Apache via Homebrew (using the brewed OpenSSL and event MPM support options).
7. Install the mod\_fastcgi module for Apache via Homebrew (using the brewed Apache option).
8. Remove any references to mod\_fastcgi in Apache's httpd.conf.
9. Add logic to Apache's httpd.conf to forward PHP requests to PHP-FPM via mod\_fastcgi (a long command that you can copy and paste into the terminal is provided in the guide in order to do this).
10. Make a ~/Sites directory to host the WordPress installation as well as "logs" and (option-

ally) “ssl” subdirectories.

11. OPTIONAL: Populate a `~/Sites/httpd-vhosts.conf` file in order to allow us to access our WordPress installs via URL's like “projectname.dev”. (A template for the vhosts file is provided in the guide. Note, I skipped this and simply use localhost.)
12. OPTIONAL: Create an SSL certificate to allow local encrypted connections.
13. Add Apache to brew services so that it starts on boot via launchd. (NB: using the default `httpd.conf`, Apache will be listening on ports 8080 and 8443, since listening on ports 80 and 443 would require root.)
14. OPTIONAL: Add a firewall rule to forward ports 80 and 443 to ports 8080 and 8443, respectively, so that you don't have to specify the port in your web browser. (The guide has a template for this.)
15. Install PHP via Homebrew.
16. Set timezone and other settings for PHP in `PHP.ini`.
17. Fix a pear and pecl permissions problem with `chmod`.
18. OPTIONAL: Install the Opcache extension via Homebrew to speed up PHP.
19. OPTIONAL: Bump up the Opcache memory limit.
20. Add PHP to brew services so that it starts on boot via launchd.
21. OPTIONAL: Install DNSMasq via Homebrew (for accessing Apache via URL's).
22. OPTIONAL: Configure DNSMasq.
23. OPTIONAL: Add DNSMasq to brew services so that it starts on boot via launchd.
24. OPTIONAL: Configure OS X to use localhost for URL requests that end in `.dev`.

It all worked perfectly for me. I left out all the steps that I marked as “optional” except for those relating to Opcache. The other optional steps had to do with SSL and setting up Apache so that you can access your WordPress installation via a URL like “`http://project-name.dev`” instead of “`http://localhost:8080`”. I didn't think that all the extra complexity was worth it.

Note that in this install, Apache runs on port 8080, which can conflict with all manner of web server stacks. I discovered this when trying to fire up a Node server only to have it raise out an error about the port being taken (and I found out that Apache was listening on the port by running:

```
lsof -i tcp:8080
```

To stop Apache, simply run:

```
brew services start httpd22
```

---

Share this:



---

#### Related

One feature that makes WP Engine worth it

November 13, 2015

In "Other"

How do Karma, Webpack, and karma-webpack interact?

January 21, 2016

With 2 comments

So, you're a Rails developer and you want to learn

"Node"?

December 8, 2015

In "Node.js"

Tweet



Pin It

Filed under: Other

Tagged: apache, dnsmasq, homebrew, lsof, mac, mod\_fastcgi, mysql, opcache, os x, php, php-fpm, ssl, vhosts, wordpress, yosemite

---

« PREVIOUS POST

NEXT POST »

---