

Laravel M1





Last update: Jan 4, 2023

Documenting the experience of working on an m1 Apple computer as a Laravel programmer.

Links

- [ARM Shaming](#): listing all Docker repos that don't work on m1
- [Is Apple Silicon Ready?](#) Broader list of apps ready for m1
- [Source Code](#) for this site




Legend

- | | |
|---|----------------------------|
|  | Haven't tested yet |
|  | Doesn't work as of testing |
|  | Works, through Rosetta 2 |
|  | Works natively |

Laravel Command-line tools

-  Laravel Installer

Laravel internal tools

-  Artisan
-  Tinker
-  Horizon

Laravel development environments

-  Laravel Valet
-  Laravel Valet Share

? Laravel Homestead

Laravel-ancillary command-line tools

- ✓ PHP
- ✓ Vim
- ✓ Composer
- 👤 [Homebrew](#)
- ✓ npm
- ✓ Lambo
- ✓ oh-my-zsh
- ✓ Homebrew Redis
- ✓ Redis
- ✓ MySQL
- ✓ PostgreSQL
- ✓ memcached
- ✓ PECL
- ? pecl install mongodb
- ⊘ pecl install redis
- ? Vagrant

Common macOS tools used by Laravel developers

- ✓ iTerm
- ✓ Tinkerwell
- ✓ Sublime Text
- ✓ VS Code
- ✓ PHPStorm
- ✓ TablePlus
- ? DBngin
- ✓ GitHub Desktop
- ✓ Slack
- ✓ Discord
- ✓ Trello
- ✓ Rocket
- ? Parallels
- ✓ Bartender

- ✓ Breaktime
- ✓ **Rogue Amoeba tools**
- ⊘ Screenflow
- ✓ OBS
- ✓ Karabiner Elements
- ? Invoker

Docker-based tools

- ⊘ Docker
- ⊘ Vessel
- ⊘ Takeout

Testing on a brand new machine, having taken these steps:

THURSDAY
MID-
AFTERNOON

Download iTerm

Install oh-my-zsh, which prompted installing the command line developer tools

Install oh-my-zsh again

``sudo mkdir /usr/local/bin && chmod 775 /usr/local/bin`` and hope that's not the wrong thing to do

Install composer and move to `/usr/local/bin` and wonder why i have to sudo it (``sudo mv composer.phar /usr/local/bin/composer``)

Install the Laravel installer (``composer global update laravel/installer``)

Use the installer (``mkdir ~/Sites && cd ~/Sites && laravel new m1suckazzzzzz``)

Womp, no find Laravel, fix our Path. edit `~/zshrc``, and add this line up top: ``export PATH="$PATH:$HOME/composer/vendor/bin``

Use the installer (``mkdir ~/Sites && cd ~/Sites && laravel new m1suckazzzzzz``)

Read and consider this article: [Run x86 Apps including homebrew on the Terminal on Apple Silicon](#), then decide nope

Decide on this one instead: [Homebrew on Apple Silicon](#)

THURSDAY
12:18AM

In iTerm, run: ``arch -x86_64 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"``. Get error: "arch: posix_spawn: /bin/bash: Bad CPU type in executable" ... Googling, no luck. Gonna try again in the morning

Stubborn. Stayed up. Found answer here:

<https://github.com/Homebrew/brew/issues/7857#issuecomment-729726634>

``/usr/sbin/softwareupdate --install-rosetta --agree-to-license``

``arch -x86_64 /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"`` YAY!

THURSDAY
12:28AM

A few errors... ``get_owner:1: command not found: stat -f`` happened twice. Hit the "Press RETURN to continue or " text.. hit return a few times, tried other keys, nothing's working. Gonna let it just run overnight I guess. 12:28am

``composer require tightenco/lambo && cd ~/Sites && lambo mynewsite`` .. works great ✓

THURSDAY
12:36AM

Tried to run the Brew installer. Hangs on "Press RETURN" again. 🙄 Really to bed this time. 12:36am

THURSDAY
7:30PM

Woke up to Jess Archer giving my sleep-addled brain the answer on my sudo permissions issue above, which was: you didn't chown /usr/local/bin, dummy!

I think my problem with the Homebrew installer is probably because I created the /usr/local/bin directory first and it's farting out on it, so I'm going to try to change its permissions myself. Homebrew suggests I need to make it group writeable, make

the group "admin", and make me the owner. I believe that's `sudo chown -R matthewstauffer:admin /usr/local/bin`, so let's try it!.. worked

... but the Homebrew installer is still hanging. It feels like it's not actually getting my "RETURN" press? I'm going to download it, edit it, and see if that is indeed the problem. Looks like this RETURN prompt is only used once, and it makes me wonder whether it's having trouble reading my TTY input or whatever, so I'm replace the internals of the wait_for_user method with "echo 'yay'". So I created a local copy of the file, made that change, and now I'm running it myself... `arch -x86_64 /bin/bash installhomebrew.sh`

No errors on the installer!! Let's goooooo

Installed the latest macOS update which took several hours.

Tried to use `git` and got this error: `xcrun: error: invalid active developer path (/Library/Developer/CommandLineTools), missing xcrun at: /Library/Developer/CommandLineTools/usr/bin/xcrun` ... let's get to googling. Google says the goal 'ol `xcode-select --install`, which I could've swore I already did? But let's do it again.

Hooray! Git works again! Now on to Homebrew; gonna run `brew update` but I have to remember to prefix it: `arch -x86_64 brew` which is a PITA so I'm gonna build a passthrough command `arm` that shortcuts that same thing.

Gonna add this to my `.zshrc` and see how it treats me for creating basically an ARM repl...

```
arm () { while read -ra c; do arch -x86_64 "${c[@]}" done } .. womp that didn't work oh well
```

Huh. in my zshrc I see this flag commented out: `export ARCHFLAGS="--arch x86_64"` .. think i'm gonna uncomment that.

OK, let's try it the old way. `arch -x86_64 bash update`

UGHHH shoot me... back to good old `arch: posix_spawn: brew: bad CPU type in executable` . WTF?! `/usr/sbin/softwareupdate --install-rosetta --`

`agree-to-license` again I guess?

HEYYYY it's doing something...

```
arch -x86_64 brew install php
```

Warning: You are using macOS 11.0. We do not provide support for this released but not yet supported version. You will encounter build failures with some formulae. Please create pull requests instead of asking for help on Homebrew's GitHub, Twitter or any other official channels. You are responsible for resolving any issues you experience while you are running this released but not yet supported version.

PHP installed!

```
brew services start php
```

 worked!

THURSDAY
1:30PM

Meetings.. back later

THURSDAY
5:19PM

OK, let's go. I read the output of this blogging format and I hate it so I'm gonna fix it quickly before I get back to M1.

THURSDAY
5:44PM

OK, spent literally 30 minutes tweaking the blog. Let's get to Valet. `composer global require laravel/valet`

Install worked with no errors. `valet install` let's go... asks for my password, gonna CTRL-C out and see if `valet trust` works on M1/Big Sur... uhhhh.. trust is not defined? Didn't I write the freaking command? It's in the docs... well.. this is not a good sign for Valet. Gonna try Valet install without running valet trust.

... so valet install is working but not valet trust? Valet installed successfully

Ahh. you can't run trust until you've run install. False alarm! Valet's installed. Let's try parking.

```
cd Sites && valet park .. boom. Let's make a site with Lambo and try it out!
```

```
cd Sites && lambo yourBestSiteNow .... let's goooo... and we're serving from  
yourbestsitenow.test!
```

Let's try `valet share` now. Hm.. it's showing the session, but when requests are made they're getting 404 (served by Valet!) instead of the right stuff. I'm gonna call Valet Share not working right now.

THURSDAY
6:06PM

OK, let's do some Redis! Then I need dinner.

```
arch -x86_64 brew install redis && brew services start redis
```

Time to download TablePlus and see if it works. Their site only says 10.11+ so it's not officially supported... looks good so far! Creating a new Redis connection... Error `setsockopt(SO_RCVTIMEO) Invalid argument`. That may be a Redis thing, or a TablePlus thing, or me just putting the details in wrong. Let me check my main computer's TablePlus. Nope, same settings. OK, let's test Redis somewhere else so we can figure out if the problem is Redis or TablePlus.

Let's just see if Redis is even running. `redis-cli ping` NOPE. OK. Homebrew *did* say it successfully started Redis. Let's see if running the start command under the different architecture matters. `arch -x86_64 brew services start redis ...` says it's already started. stopped and restarted it... Still this: `Could not connect to Redis at 127.0.0.1:6379: Connection refused`

I'm gonna say Homebrew Redis isn't working out of the box. I'll uninstall it and try non-Homebrew. Following this: <https://phoenixnap.com/kb/install-redis-on-mac> (Option 2)

After running ``make`` I see `"/bin/sh: pk-config: command not found"` a few times. Not a good sign!

```
make test .. as the output flies by I see that same error several times...
```

THURSDAY
6:17PM

While this runs I'm gonna go eat.

THURSDAY
7:11PM

All tests passed without errors!

... ran `sudo make install` and it gave an error and then output `INSTALL install` five times.... did that.. do anything? I feel like a Linux user right now.

Read the README... OK, so this was supposed to put it in the `/usr/local/bin`. Let's check!

... OK, yah it is there. Thought I checked but I guess I hadn't. Let's try! `redis-cli ping` ... UGH. Same problem as with Homebrew. So there's something not working. Maybe debug-able? But not for me tonight.

THURSDAY
7:26PM

Let's try... some downloadable apps! Sublime Text is up first.

Downloading build 3211... works like a charm!

PHPStorm says they work ("Our IDEs work on Apple Silicon chips through Rosetta 2.") so I'm gonna move on to things that haven't made official statements yet.

Let's try MySQL, PostgreSQL, and Memcached!

Oops, got distracted watching my brother's live show. TAKE TWO. MySQL.

THURSDAY
8:11PM

MySQL. `arch -x86_64 brew install mysql && brew services start mysql`

Tested it out, and we have a functioning local `mysql`, I can access it from TablePlus, and my Laravel app is integrating with it.

THURSDAY
9:10PM

OK, let's check out Postgres. I made a [nice little shortcut](#) in my `.zshrc` that lets me write just `arm brew install whatever`, so future writing here will just show `arm`.

`arm brew install postgresql && brew services start postgresql` ... that worked!

The way I connected to Postgres on TablePlus doesn't work unless there's a database table named the same as my computer user, and since I didn't feel like taking the time to figure out why, I just ran `psql postgres` and in that REPL I

wrote `CREATE DATABASE matthewstauffer;` and then hit enter. Once that was done, I could connect to this Postgres database using the default settings from TablePlus. Good to go!

OK, just checked out a ton of GUI apps that all work fine. Gonna call it for a while.

THURSDAY
10:13PM

Lied. Tried OBS. Doesn't work.

FRIDAY
9:19AM

OK, got a few minutes free. What's next? I can't try Horizon without Redis. Homestead is a big lift, so I'm gonna wait on that. Memcached! Let's go!

`arm brew install memcached && brew services start memcached` ... ran great. Now I have to remember how to test memcached. Honestly, using it for a Laravel app's cache is probably easiest.

Whoops, now I remember! You have to install the memcached extension using PECL. Let's test PECL!

I went through half of the steps to set up PECL, only to find out it's already set up by Homebrew now!! YES! Thank God!

Now I just have to set it up for this extension. `arm brew install libmemcached` and then `pecl install memcached` .. hit enter on all the defaults... yet another one of these `pkg-config` errors. This one says: "pkg-config not found / configure: error: Please reinstall the pkg-config distribution / ERROR: (failure notice)"

OK, let's learn about `pkg-config`, cause I know nothing about it.

well, it's on Homebrew, so let's try that. `arm brew install pkg-config`

OK, `pecl install memcached` again.

FRIDAY
10:37AM

Got taken away by my actual job. Back, that threw another error: memcached support requires ZLIB. `arm brew install zlib` .. another meeting..

FRIDAY
1:23PM

Sorry, day full of meetings. Hoping to finish memcached today, then I'll be back on tonight after kids' bed time trying to figure out more of them.

Ran it during a meeting, see deprecation warning for `sasl_done` but looks like that's just a warning for now. HEY! Installed correctly to `/usr/local/Cellar/php/7.4.12/pecl/20190902/memcached.so`, and they already enabled it in `php.ini` for me.. let's test it! OK, class memcached not found. I'm gonna have to test things, also maybe restart php... nope that didn't do it. Will need to take some time, this may very well be pebkac, not m1, I remember Memcached and pecl are a pain some times. Gotta run get the kids though.

FRIDAY
9:54PM

Checked php.ini and `extension="memcached.so"` is up at the top... `php -i` indicates memcached support is enabled and libmemcached is connected.. and in Tinker, if I run `$m = new Memcached`, while it throws an error because no servers are defined, the class exists. So is this a Valet issue? Maybe that old Sock problem where Valet is hitting the wrong PHP? Yep! `brew services stop php` but Valet is still running. What gives, Valet? Running `valet install` again, because that does a more full reboot.. HEY! There we go! Memcached working!

SATURDAY
4:45PM

I had a sneaking suspicion OBS would work fine if I tried it again, so I did, and this time it successfully asked for permission to record my screen; once given, it worked. Woop!

FRIDAY
11:54AM

Several folks have suggested that [prefixing redis with sudo](#) works temporarily. Let's test it.

Ok, `sudo arch -x86_64 redis-server` didn't work. I'll try reinstalling it. `arm brew reinstall redis -s` (`-s` builds from source)

Symlink failed. Force link: `arm brew link --overwrite redis`

`sudo arch -x86_64 redis-server` take two

Works! So if you build from source and manually run `redis-server` when you need it, you can make it work! I'll call this partial support, although I don't have an icon for that.

© **Tighten** 2023.

Built with **Jigsaw** and **Tailwind CSS**.