Beyond Laravel

An Entrepreneur's Guide to Building Effective Software

by Michael Akopov

a php[architect] guide

Table of Contents

Chapter 4

Developer Tools

What truly makes Laravel unique is its ecosystem. Laravel has been consistent with its quality and usefulness of features while maintaining clean code and thorough documentation. It has led to deep trust over the years in what developers can expect from Laravel—resulting in an ecosystem consisting of core Laravel as well as third party applications and packages.

This chapter focuses on core Laravel packages that help you be productive when developing your application. Laravel offers options for building your application, environments for local development, and coding tools for boosting your productivity.

Valet

When I first learned of Valet[1], I became a very happy developer. I would argue that Valet helped me save money and extended the life of my laptop. "How?" you might ask. Before I can answer that, I have to explain what Valet is first.

Building a web application requires running your application locally, as this is the most convenient way for you to change code and see immediate results. It doesn't rely on a cloud server or an internet connection to code, making the experience much smoother for you as a developer. To do that, you'd traditionally run a Linux VM on your local system that sets up the webserver, PHP, database, cache, and other components. Ideally, you'd even provision this using an automated provisioning system such as Vagrant.

This approach is valid, and it works fine, but there are several bottlenecks with it. First, It was resource-intensive. Typically running a virtual machine required redundant overhead ram usage as you were running an entire operating system on top of your already running system. This solution meant having a system with 16GB or more of RAM to run things properly. Another drawback was having to rebuild this when things broke. I say "when" because, in a development environment, it's only a matter of time before you have to rebuild the thing from scratch. Doing so is a very time-consuming process when doing it by hand and leads to distractions and frustration.

A few years ago, a new type of service came along that I'm sure you've heard of, Docker. Docker has long been oversimplified as a lightweight virtual machine. While this somewhat describes the result of running a Docker container, it isn't an accurate representation. I would also like to take this moment to clarify that a container doesn't have to be a Docker container. The container concept has been baked into the Linux kernel. But Docker made this technology mainstream in a whole new way. You could now run an entire Linux image that was just 5MB in size. This was unheard of. By running only what you need within that container, you can minimize the file size and resource utilization without too much hassle.

Containers make use of the underlying operating system for everything that it doesn't need on its own. In fact, containers employ what is known as copy-on-write. It only creates a system file in the image if it were to differ from the host.

[1] Valet: https://laravel.com/docs/7.x/valet

Running a set of containers is an excellent solution for replicating specific environments in a lightweight and portable fashion and works well for development. However, frequently you find that you're not changing PHP or database versions all that often. You don't need the ability to set up and tear down a container or virtual machine. Typically, to run a Laravel site, all you need is PHP, NGINX, and MySQL or PostgreSQL. These services can run natively on any operating system. Traditionally, setting all of these up on a desktop system such as Windows or even MacOS is a pain. Sure, you can do it with Vagrant, but you're still scripting all of this.

Here is where Valet comes in. Valet is the brainchild of Adam Wathan and Taylor Otwell. They decided that running a VM or a set of containers was simply overkill for running a basic Laravel site. So they built Valet, a service that runs on a Mac and configures a webserver with all of the PHP connections and DNS routing necessary to serve a website.

Valet makes it very easy to turn any PHP website directory into a hosted site locally by running a `valet link` command. Valet automatically detects what framework your site is written in. Did I mention Valet supports many PHP based frameworks outside of Laravel? That's right. I even use it for Drupal 7 and 8 development. After detecting your framework, it configures the necessary routes to your `index.php` file, after which your site is available at the domain `http://{site_dir_name}.test`.

You're still required to install PHP and MySQL or PostgreSQL separately through either Homebrew or some other way, but everything else is configured automatically. This approach allows a very lightweight and minimal installation on your Mac, which is great for battery life and local development.

Valet ships with all sorts of useful features, and there are a few worth pointing out that are likely to benefit every user of Valet. Sometimes, you want or need to run your local development site via SSL. Valet makes this as straightforward as running `valet secure` within the directory of your site. After it runs, you will notice that it's secured with a TLS connection when you visit your site. You should see the HTTPS lock icon in your browser.

One thing I find myself often doing—being a remote developer (especially under quarantine)—is wanting to share the status or current state of my site with a colleague or friend to get some quick feedback. Valet makes this possible by running the `valet share` command, which uses ngrok[2] to create a tunnel and displays a shareable URL that allows access from the outside world into your system.

Valet offers other helpful features that are worth exploring in the docs[3]. Valet's one downside is that it's only usable by Mac users. There are attempts by community members to create a Microsoft Windows equivalent of Valet. I've never needed to use them, so I cannot speak on their behalf.

Homestead

While Valet is a great resource for Mac users, it isn't the right fit for everyone. If you're a Windows, Linux, or even a Mac user that prefers to keep their system free of any configuration mess, you may consider Homestead[4] instead. Homestead is a virtual machine setup by the Laravel community that is preconfigured with everything needed to run a Laravel site.

I've built and used many virtual machines over the years. I have to admit that Homestead is a very well put together virtual machine. Aside from merely running a site with SSL, it offers several powerful features that make developing a site locally quick and convenient. For example, it allows developers to set up automatic backups of development databases and database snapshots using LVM, which allows for fast restores. This is handy if you need to test a dataset then restore your database to a previous point in time.

Like Valet, Homestead also supports other frameworks and site types beyond Laravel, which is valuable when working with various other projects. One feature most of us would make use of is the out-of-the-box setup of Mailhog within Homestead. Valet doesn't ship with an equivalent solution. If you're like me and prefer to use Valet, you're probably using a third-party mail catch service to test your development emails. However, with Mailhog being built-in to Homestead, there is one less reason to need an internet connection, which is nice for those camping trips.

[2] ngrok: https://ngrok.com
[3] docs: https://laravel.com/docs/8.x/valet
[4] Homestead: https://laravel.com/docs/7.x/homestead

Index