

www.phparch.com



April 2020
Volume 19 - Issue 4

Machine Learning and OpenAPI

Free
Sample
Article

The Business Of PHP

**Hands-On Machine Learning
With PHP, Part One**

OpenAPI: More Than Just Code

ALSO INSIDE

Education Station:
Anatomy of a Web Request

Community Corner:
PHP Adelaide: PHP Down Under

History and Computing:
Silicon Valley Takedown

Security Corner:
Buzzword Bingo

The Workshop:
Easy CLI PHP with Symfony Console 5

finally{ }:
The One About COVID-19



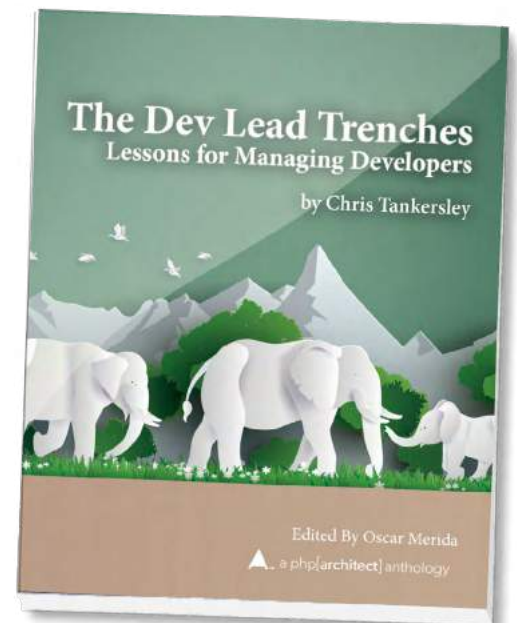
You're the Team Lead—Now What?

Whether you're a seasoned lead developer or have just been "promoted" to the role, this collection can help you nurture an expert programming team within your organization.

Get the Most Out of Your Developers

After reading this book, you'll understand what processes work for managing the tasks needed to turn a new feature or bug into deployable code. But success is more than just slinging code when you're in charge, and this book covers project management and people skills you'll need to hone.

This book collects almost two-years worth of writings based on Chris Tankersley's experience leading development teams.



Available in Print, PDF, EPUB, and Mobi.

Order Your Copy
<http://phpa.me/devlead-book>

Integrating with another web site but an API is not available?

Read a
Sample
Online

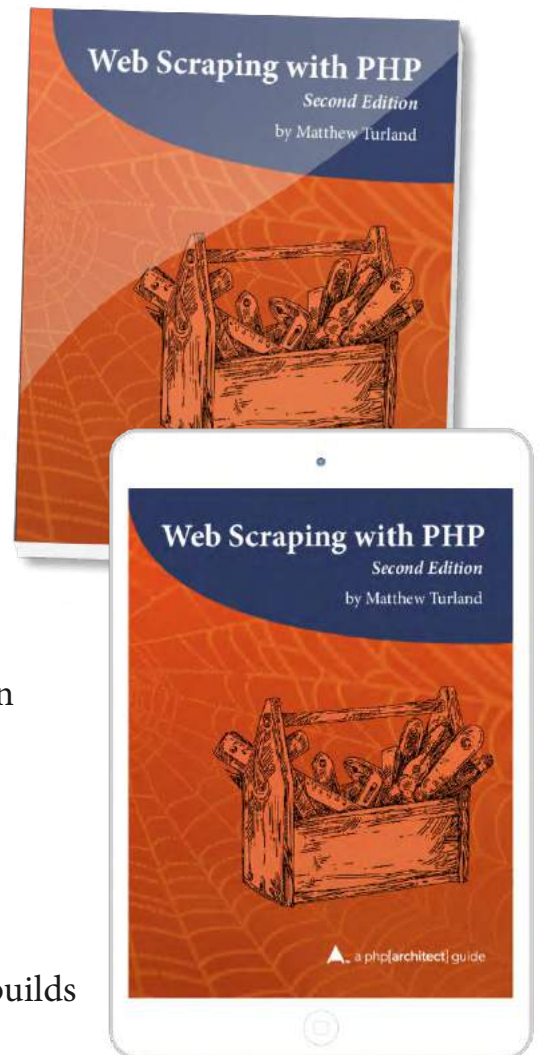
Web scraping is a time-honored technique for collecting the information you need from a web page. In this book, you'll learn the various tools and libraries available in PHP to **retrieve, parse, and extract data** from HTML.

Web Scraping with PHP, 2nd Edition includes updates to the techniques of the first edition to account for modern PHP 7 based libraries written to more easily interact with web markup and data.

- HTTP requests and responses
- PHP's HTTP Stream wrapper
- Using the cURL extension
- Working with the pecl_http extension
- Parsing responses with Guzzle
- Zend Framework's HTTP classes
- An overview of Symfony's libraries for web automation
- Writing a client from scratch
- Extensions for parsing and tidying XML and HTML
- Using regular expressions
- ... and more.

Written by PHP professional Matthew Turland, this book builds on his expertise in creating custom web clients.

Available in Print, PDF, EPUB, and Mobi.



Order Your Copy

<http://phpa.me/web-scraping-2ed>

The Business Of PHP

Sherri Wheeler

Among the many strategic decisions a business needs to make, one which we as developers are involved with is the choice of tech stack—the tools and technologies used for software development. When choosing a backend language, PHP presents its list of strengths and weaknesses; this article examines these from a business perspective.

In many cases, the availability of tech talent is one of the main factors in selecting front and backend programming languages—a situation especially true for companies who don't hire remote workers. A recruitment drive is launched or a market analysis performed, and the company gets a feel for which languages and frameworks are popular in their city and hires accordingly. Often this is dictated by the languages taught at local schools and colleges. This dynamic creates a feedback loop. More companies hiring for a given language, means colleges try to produce graduates with skills in those languages. Available talent in those languages means companies hire them, and so on. User groups spring up for those languages, and beginners hear about a given language from their peers, so they make a similar choice. As a result, we frequently see regional concentrations in programming languages. It makes business sense for a company to hire based on the largest pool of talent because they can offer lower salaries for higher skill levels. However, when regional skills are not a factor, or if hiring remote workers, a company is free to put more consideration into their choice.

When looking at PHP as a backend language, there is a long list of pros and cons to factor into the decision. I'll cover many of these from both technological and business perspectives. I'll cover many of these factors from technological and business perspectives based on my experience.

Technical Merit

Many of us come to PHP because of its ubiquity on ready-to-go server environments. For a startup without a devops team, the fact that a solid PHP and MySQL environment is available on many VPS services (managed or not) and shared hosting is a tempting factor. These are usually already secured, optimized, automatically backed up, and managed. In the early days of a new project or SAAS launch, a small team can confidently launch even moderately complex projects in less than a couple hours without worrying about extensions, updates, or securing the environment.

Availability

PHP runs on Linux, Mac, and Windows and includes over twenty vendor-specific database extensions. It fits seamlessly into any server stack, and it runs well alongside other languages. Thanks to Ondřej Surý's repository for Ubuntu and Debian¹ and Remi Collet's repo for CentOS and Red Hat², most teams don't need to build PHP from scratch and can install recent releases in a couple of steps. Developers working locally have several options to set up a local development environment thanks to high-quality tools from the community. The range from WAMP/MAMP, to Vagrant boxes like Homestead, to Docker via tools like Devilbox. Tutorials for these or for installing it directly on your system are plentiful.

¹ Ubuntu and Debian: <https://deb.sury.org>

² CentOS and Red Hat: <https://blog.remirepo.net>

These advantages are most applicable to companies with small teams or junior developers who might not be comfortable with configuring servers and environments yet. Companies that do not have a dedicated devops team can benefit from PHP's universal availability, simple configuration, and community-supported tooling.

Applications and Frameworks

Companies focused on building websites more often than custom web applications can find easy, flexible solutions in the many, many CMS, ecommerce, and website frameworks available in PHP. WordPress, Joomla, and Drupal are some of the oldest and most popular. My own experience has been that I can get a customized, feature-full WordPress website up and running from scratch for a small business client within an afternoon. Many small marketing and web design shops leverage these free, open-source applications as the core of their business. Companies that focus more on custom web development can leverage robust and feature-rich application frameworks like the popular Symfony, Laravel, and SlimPHP to build upon. PHP is mature and well suited to providing an API for different frontend clients or native applications to consume.

Performance

PHP is fast and scalable. With a few considerations at development time, PHP can be easily scaled horizontally³ behind a load balancer. I hope I don't

³ scaled horizontally: <http://phpa.me/horiz-scaling-php>

need to cite the many examples of the internet's top, traffic-heavy websites which run on PHP, or a derivative of it⁴.

It's interesting to note that in speed benchmarks, PHP isn't the fastest in every test, but in most of them. When it isn't fastest, it does give the other top languages a run for their money⁵. PHP's speed took a massive jump to more than double from the version 5 branch to the latest 7.3, and in some cases, even tripled. The many frameworks and CMSes also benefited from this boost. Each version released in the PHP 7 branch has seen a speed improvement, and features like preloading⁶ (available in 7.4) and JIT⁷—targeted for version 8—promise even more speed.

Team Building

Putting together a team with the right skills and experience, and then supporting them with the resources and environment they need to build great software is an entire topic by itself. There are a few metrics that come to mind when we look at hiring a team of PHP developers.

The pool of available talent is hard to measure, but we can make a few inferences based on related information. When it comes to jobs available in PHP, we can assume more positions means more demand for talent. Looking at data from major job site Indeed⁸, a comparison of job numbers in the United States reveals PHP is still in high demand. More jobs listed for PHP at the time of this writing (17k), than Node.js (8k) or Ruby on Rails (3k). On par with Python for web (18k), though unsurprisingly less than the frontend monopoly held by JavaScript (54k). Jobs create growth in the associated pool of talent. It is especially true in

languages which have been around for long enough that they aren't suffering from the drastic lack of ability common to new languages or frameworks.

Building a team also means supporting those developers and creating an environment they can excel in. PHP is a very flexible language, allowing a wide variety of coding styles, paradigms, and frameworks to be used by your team. They can work with PHP in the way that best suits them and your type of projects. On the flip side, however, PHP can't be used for the frontend like JavaScript, or to create native or desktop applications. It certainly doesn't fit every business model; however, PHP is particularly well suited to providing an API or shared content store for these frontend applications to use.

Popularity and Reputation

Popularity is a significant consideration when looking for skilled developers. Not only do developers need to be proficient in PHP, but they also have to want to work with the language. With PHP being one of the oldest backend web development languages, and suffering a stagnation during the v5 branch for nearly a decade, it would be sticking our heads in the sand to fail to admit PHP has a reputation problem. Outdated information about the language itself continues to float around and come up repeatedly in articles trumpeting the death of PHP or attempting to boost other languages. New developers are often attracted to newer backend options like Node.js or corporate sponsored languages like .Net. This tendency can definitely be a challenge to overcome.

Reputation problems can be offset, however, by several positives. With its age, PHP has a long tail of talented developers who have been working with the language for a long time. There are many events and communities focused on PHP, which I'll discuss in more detail later in the article. If you value mature developers with experience, PHP has a pool of talent with expertise in spades. Strategically, you may

even find experienced talent at a more reasonable salary if you avoid trying to hire for the hottest new tech. Happily, there has been a new direction for PHP in the last several years, which is slowly chiseling away at PHP's old baggage. A rapid release schedule for the v7 branch has introduced popular new features and removed old security issues that once earned PHP its legitimate criticism. Modern frameworks like Laravel, updates to Symfony like Flex, and even the steady improvements to WordPress are creating renewed interest in PHP and bringing old developers back into the fold. The long term outlook for PHP is good.

Support and Materials

Hiring a team of developers and making the most out of the investment means ensuring your team has access to a wealth of resources to help in their daily development. Languages with excellent online resources (especially if free) are some of the easiest to work with. In one of my past developer jobs, my employer had web access in the building locked down so strictly that developers couldn't access several coding help sites like a popular predecessor to StackOverflow called Experts Exchange. We couldn't watch coding videos and couldn't post on most popular programming forums. It was a challenging working environment. A question that I could typically solve with a quick online search often meant pouring through the dry official language documentation, flipping through paper books, or interrupting coworkers. When the staff eventually argued successfully for lifting the restrictions, I found it to be only slightly helpful. My experience all those years ago was that the language we used was not one people often blogged about or willingly shared their code. Everything seemed to have a paywall and negative attitudes toward open source.

Contrast that with PHP, which has been eagerly shared and discussed publicly since its creation. A quick

⁴ derivative of it:

<http://phpa.me/wikipedia-programming-popular>

⁵ their money:

<http://phpa.me/debian-benchmarks-php>

⁶ preloading:

<https://wiki.php.net/rfc/preload>

⁷ JIT: <https://wiki.php.net/rfc/jit>

⁸ Indeed:

<https://www.indeed.com/q-php-jobs.html>

Google search reveals 18 Billion⁹ results for “PHP.” Much more than other backend languages. Although many excellent surveys like the Tiobe Index¹⁰ or the GitHub Octoverse report¹¹ show slight declines for PHP over recent years, this isn’t surprising given that PHP is so mature, and no longer holds a near-monopoly on backend development. The v7 branch of PHP has kickstarted renewed discussion and article writing about PHP. Another positive is that being an open-source language itself with a long history of sharing knowledge among developers means most of the resources your staff may need are free.

One drawback to the prolific online content about PHP is that much of the content is very outdated. It’s not uncommon to find bugs caused by

developers copy-pasting old code snippets, which include security vulnerabilities like SQL injection, or use features that have been deprecated or removed from the language. A good dialog with new team members, encouraging sharing of favored resources among your developers, setting policies for where the team can use code from, and peer code reviews can help mitigate this problem.

Community and Conferences

The conferences page¹² on the PHP.net website lists dozens of conferences across the globe. You can find them in a wide variety of countries, at a wide range of ticket prices, and covering topics which vary from PHP in general to specific frameworks and applications. Some organizations like Nomad PHP¹³, Day Camp 4 Developers¹⁴, and

Laracon Online¹⁵ make it a point to make conference-quality talks and events available to everyone online for a low cost. Companies who wish to best support their developers should include some type of continuing education for their staff. Conferences are a great way to do this. Choosing a language which has a variety of these types of events to choose from makes this a much more feasible goal. Some companies save costs by sending a different small group of developers to a conference each year and asking them to share what they learned with the rest of the staff when they return. Developers value working for a company that fosters their desire to improve their skills, and PHP makes this easy.

Local user groups are another valuable resource which can have an immense benefit for your development team. At a user group, developers can learn new skills, hash out problems, network with like-minded people, and even promote your company’s job

9 18 Billion:

<https://google.com/search?q=PHP&oq=PHP>

10 Tiobe Index:

<https://www.tiobe.com/tiobe-index/>

11 GitHub Octoverse report:

<https://octoverse.github.com>

12 conferences page:

<https://php.net/conferences/>

13 Nomad PHP: <https://nomadphp.com>

14 Day Camp 4 Developers:

<https://daycamp4developers.com>

15 Laracon Online: <https://laracon.net>



Web Apps • Mobile Apps • E-Commerce

Developers who care about the code they create, the communities they build, and the solutions they implement

www.diegoddev.com

openings to other skilled programmers. My participation in a local user group has been invaluable, and fosters a sense of excitement about my work, and helped me discover new tools or technologies to bring into my daily toolkit. PHP.ug¹⁶ is one site helping developers find nearby user groups, and a look at their map shows user groups in most major cities across North America.

Package Ecosystem

One of the smartest decisions a company can make regarding software development is to leverage the code written by others to avoid reinventing the wheel. It makes no sense to have developers building components that have no relation to your core business expertise. In fact, sticking to third-party packages that are well-reviewed, mature, and popular means your team probably couldn't do a better job anyway, not without a substantial investment of time and money. The package ecosystem for PHP is a plentiful world of many choices and high-value resources created by passionate developers. Packagist, the directory of packages officially published for PHP's package manager Composer, currently lists 253k packages¹⁷, you can install in a project with a single command. A GitHub search reveals over one million¹⁸ repositories with PHP.

A new PHP project can be built on a framework that does much of the bootstrapping and heavy-lifting of an application. Frameworks save developers by solving problems with existing solutions and best-practices and avoiding falling into common security pitfalls. PHP is no slouch in the choice of [high-quality frameworks](#). Of note is that most of the popular frameworks for PHP suit a particular need instead of being nearly identical choices. Whether you need something geared to rapid development and beautiful code like Laravel, or something stable

and expandable like Symfony, or something lightweight and straightforward like SlimPHP, your team has excellent, mature options available to choose. PHP's long history means many frameworks have come and gone, and over the years, the best have risen to the top.

Any profitable business keeps an eye on its bottom line. Third-party code and frameworks are usually open-source and free. The business world is slowly awakening to the value of leveraging this free code. However, increasing popularity and pressure on open source packages can strain open source developers. For example, Derick Rethans, the developer of the very popular XDebug PHP extension, has made no secret¹⁹ of the pressure he is under to develop XDebug and his need for support. Businesses leveraging free open source packages or tools should financially support them and include this in their annual business expenses. This support helps keep the packages they rely on viable and up-to-date while still being a drastic net-benefit for everyone involved.

Financial Costs

How much does it cost to develop and publish applications with PHP? The answer might surprise people coming from industries and languages that have commercial license requirements, commercial distribution app stores, or expensive IDEs. For example, publishing a game²⁰ for Microsoft's Xbox One comes with insurance, license, and ratings board fees in the thousands of dollars. For a startup planning a new project, costs can be a big factor. Luckily, PHP is one of the cheapest languages to develop with. There are no fees to use the language itself. There are several free code editors available which support syntax highlighting and IntelliSense for PHP. And launching a PHP project online is as cheap as a \$10/year domain and a \$5/month VPS if

you have server administration skills on staff.

While PHP is free to use, many businesses want someone to call if they need help. The core team of PHP developers is mainly volunteers. There is no corporate entity in charge of PHP. This is a good thing, ensuring PHP is not subject to the whims of a profit-motivated entity. It does create some uncertainty, however, when the people working hard on new PHP releases are volunteers, and it's not clear how viable that can be for the long term. One thing we can be sure of is there are many people, from many backgrounds working on PHP, and many more ready to take up the mantle if needed; PHP doesn't rely on one single person²¹. Businesses looking to throw money at someone to guarantee support and stability can look to companies like Zend²². While they no longer hold the reins of the language itself, they do still offer commercial support, monitoring, servers, and more.

What can be more critical for profits than the ability to land clients? Some businesses which market themselves as development studios might encounter problems stemming from the stigma PHP still carries. Some clients like to dictate the language used for their project. I'm sure I'm not the first business owner to meet with a potential tech-savvy client and receive a skeptical response when explaining that I use PHP. However, a focus on what a business can do, rather than how they'll do it, can go a long way toward avoiding these concerns. Nothing proves the effectiveness of PHP better than launching an awesome new project for a client, thanks to the many benefits outlined in this article.

Long Term Outlook

Every new business plans to be around for the long haul. Choosing tools and languages with an excellent long-term outlook is very important. As is developing in a language which can minimize code rot over time. While

¹⁶ PHP.ug: <https://php.ug>

¹⁷ 253k packages: <https://packagist.org/statistics>

¹⁸ one million: <http://phpa.me/github-php-repos>

¹⁹ no secret: <https://derickrethans.nl/xdebug-experience.html>

²⁰ publishing a game: <http://phpa.me/kotaku-cost-xbox-indie>

²¹ single person: <http://phpa.me/wikip-bdl>

²² Zend: <http://phpa.me/zend-lts>

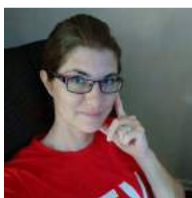
I wouldn't be so bold as to claim that PHP is the very best in these regards, we can definitely consider it to not be a concern. PHP core development is currently robust; recent releases have been coming out rapidly. The core team keeps backward compatibility in mind, and upgrading to new versions is often simple and never a total paradigm change like what happened when AngularJS was replaced by Angular ²³. High-quality PHP packages tend to be quite stable. So far, PHP has avoided the meme-inducing `node_modules` ballooning problem²⁴ in JavaScript, which can lead to code rot and security concerns in dependencies. PHP also continues to get preferential treatment by many server stacks and managed VPS providers. PHP's future is determined by vote by members of the core

team and community. With all of these considerations plus a large, passionate community and highly anticipated features coming in version 8, the future looks bright indeed.

Conclusion

Examining the choice of programming language on which to hang the future of your company can be a source of stress for managers and startups. The temptation is often to chase the new

shiny technologies that everyone is talking about. While hype and newness have their allure, it's prudent to look deeper into this decision. PHP is a strong choice for companies focused on backend development. It is mature, stable, popular, inexpensive, and comes with an extensive community and a wealth of talent and resources. If you last considered PHP five or more years ago, give it a second look; you won't be disappointed.



Sherri is a web software developer and business professional with over 15 years of experience working with PHP. She currently operates her company Avinus Web Services, and for fun she creates games and open source software. When not at the computer you'll find her camping, swimming and making forts with her husband and three sons in Ontario, Canada. Connect with her on Twitter: @SyntaxSeed

²³ Angular 2:

<http://phpa.me/simplilearn-angular>

²⁴ ballooning problem:

<https://phpa.me/3bEDjui>

Related Reading

- *Succeeding as a Freelance Developer* by Stefany Newman, August 2018. <http://phpa.me/freelancer-aug-2018>
- *Choose Your Own Adventure—Freelancer or Founder?* by Joshua Warren, January 2015. <https://phparch.com/magazine/2015-2/january/>
- *finally{}: Ups and Downs of an Entrepreneur* by Eli White, June 2016. <https://www.phparch.com/magazine/2016-2/august/>

Listen to Episode 30:

- Updating legacy architecture
- Asynchronous processes
- Development environments
- Enterprise software
- Ben Marks on the state of Magento



<https://phpa.me/podcast-ep-30>



Borrowed this magazine?

Get php[architect] delivered to your
doorstep or digitally every month!

Each issue of php[architect] magazine focuses on an important
topic that PHP developers face every day.

We cover topics such as frameworks, security, ecommerce, databases, scalability,
migration, API integration, devops, cloud services, business development, content
management systems, and the PHP community.



Digital and Print+Digital
Subscriptions
Starting at \$49/Year

http://phpa.me/mag_subscribe