

www.phparch.com

May 2018
Volume 17 - Issue 5



Treasure, Old & New

Up to My Eyeballs in
Technical Debt!

The Life-Changing Magic of
Tidying Your Code

Moving a Monolith to AWS

Easier Mocking with
Mockery, Part 2

ALSO INSIDE

Artisanal:
Odds and Ends

The Dev Lead Trenches:
It's Toxic

Free
Sample
Article

Community Corner:
My Picks From Packagist

Security Corner:
Paying Off Technical Debt

Education Station:
Build an API, Part Two

finally{}:
Innovation in PHP

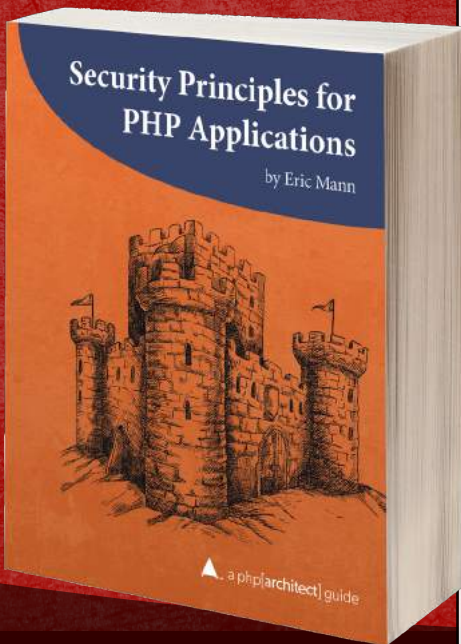
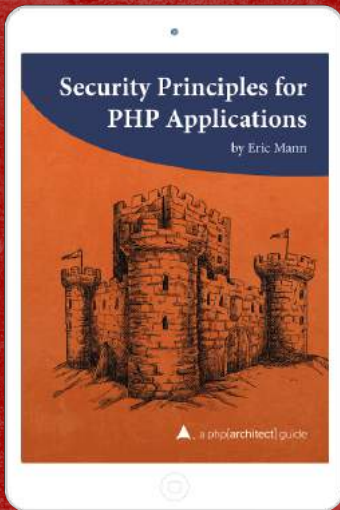


We're hiring PHP developers

15 years of experience with
PHP Application Hosting

SUPPORT FOR *php7* SINCE DAY ONE

Contact careers@nexcess.net for more information.



Discover how to secure your applications against many of the vulnerabilities exploited by attackers.

Security is an ongoing process not something to add right before your app launches. In this book, you'll learn how to write secure PHP applications from first principles. Why wait until your site is attacked or your data is breached? Prevent your exposure by being aware of the ways a malicious user might hijack your web site or API.

Security Principles for PHP Applications is a comprehensive guide. This book contains examples of vulnerable code side-by-side with solutions to harden it. Organized around the 2017 OWASP Top Ten list, topics cover include:

- Injection Attacks
- Authentication and Session Management
- Sensitive Data Exposure
- Access Control and Password Handling
- PHP Security Settings
- Cross-Site Scripting
- Logging and Monitoring
- API Protection
- Cross-Site Request Forgery
- ...and more.

Read a
Sample
Online

Written by PHP professional Eric Mann, this book builds on his experience in building secure, web applications with PHP.

Order Your Copy

<http://phpa.me/security-principles>

CONTENTS



php[architect]

MAY 2018

Volume 17 - Issue 5

Features

3 Up to My Eyeballs in Technical Debt!

Steve Grunwell

9 The Life-Changing Magic of Tidying Your Code

Bryce Embry

16 Moving a Monolith to AWS

Keanan Koppenhaver

21 Easier Mocking with Mockery, Part 2

Robert Basic

Columns

- 2 **Editorial:**
Treasure, Old & New
Oscar Merida
- 28 **Artisanal:**
Odds and Ends
Joe Ferguson
- 31 **The Dev Lead Trenches:**
It's Toxic
Chris Tankersley
- 34 **Security Corner:**
Paying Off Technical Debt
Eric Mann
- 36 **Community Corner:**
My Picks From Packagist
James Titcumb
- 39 **Education Station:**
Build an API, Part Two
Edward Barnard
- 46 **News:**
April Happenings
- 48 **finally{}**
Innovation in PHP
Eli White

Editor-in-Chief: Oscar Merida

Editor: Kara Ferguson

Subscriptions

Print, digital, and corporate subscriptions are available. Visit <https://www.phparch.com/magazine> to subscribe or email contact@phparch.com for more information.

Advertising

To learn about advertising and receive the full prospectus, contact us at ads@phparch.com today!

Managing Partners

Kevin Bruce, Oscar Merida, Sandy Smith

php[architect] is published twelve times a year by: musketeers.me, LLC
201 Adams Avenue
Alexandria, VA 22301, USA

Although all possible care has been placed in assuring the accuracy of the contents of this magazine, including all associated source code, listings and figures, the publisher assumes no responsibilities with regards of use of the information contained herein or in all associated material.

php[architect], php[a], the php[architect] logo, musketeers.me, LLC and the musketeers.me, LLC logo are trademarks of musketeers.me, LLC.

Contact Information:

General mailbox: contact@phparch.com

Editorial: editors@phparch.com

Print ISSN 1709-7169

Digital ISSN 2375-3544

Copyright © 2018—musketeers.me, LLC
All Rights Reserved



My Picks From Packagist

James Titcumb

Most people are aware of how the Composer revolution came about, and the goal of getting everyone to play nicely together. For the most part, it seems to have worked, and more and more developers are coming on board with avoiding “Not Invented Here” and embracing the more than 1.2 million versions of packages available today. With so many packages it can sometimes be difficult to know where to begin. I’m going to look at some packages from the open source community you might find useful if you’re not already using them!



Infection

Infection is a mutation testing tool, which is a technique allowing you to test whether your tests are testing—yes, you read that right. Mutation testing will run your unit tests after having tweaked your code to try and break it. If the tests fail, that’s good. If the tests still pass, it means the breakage wasn’t captured by your unit tests. Most of the time, Infection can simply be added to your `require-dev` dependencies and run in your CI pipeline as part of a unit test health check step.

<https://packagist.org/packages/infection/infection>

Symfony Console

Hopefully, this package needs no introduction. For me, at least, this is the go-to package for writing command line interface applications beyond a simple shell script. Argument parsing is always a pain to do by hand, and the Symfony Console component makes this a real no-brainer, as well as including a whole host of other useful features like interactivity, output formatting, password hiding, and a bunch more. The documentation is well written, and a great reference for getting started writing CLI tools.

<https://packagist.org/packages/symfony/console>

PHP-Mock

PHP-Mock is a handy tool which allows you to mock built-in PHP functions. There is a caveat though. If you’re importing functions explicitly (e.g., use `function time`; or at call time with `$t = \time()`) it won’t work, as it takes advantage of the fact the namespace qualification was not made. There are other ways, but you’ll need to use a PHP extension like Joe Watkins’ UOPZ extension.

<https://packagist.org/packages/php-mock/php-mock>

vfsStream

Another tool along the testing lines I’ve found instrumental is `vfsStream`. It’s a virtual file system that can be used to replace interactions with a real filesystem. It provides a stream wrapper which allows filesystem reads/writes to be done in memory instead of on disk, meaning you can control the testing environment much better.

<https://packagist.org/packages/mikey179/vfsStream>

Faker

You might not have heard of Faker before, but it’s a tool to generate fake data, for example populating a database with realistic-looking but fake data or anonymizing data from your production environment. It can generate names, addresses, lorem ipsum text, dates, and much more, as well as featuring internationalization for a bunch of countries. You could generate a random realistic(ish) address in India (`en_IN`), Jordan (`ar_JO`), or Japan (`ja_JP`). It’s very comprehensive set of fake data!

<https://packagist.org/packages/fzaninotto/faker>

Roave Security Advisories

This package doesn’t have any code! It’s just a Composer package you can add to `require-dev` that contains a huge amount of generated conflict definitions which will cause Composer to not install packages with known security vulnerabilities. The information about vulnerabilities is extracted from the FriendsOfPHP/security-advisories¹ repository. As long as that is up to date, whenever you run `composer update`, you can rest a little easier knowing you won’t get any obvious security vulnerabilities installed along with your dependencies!

<https://packagist.org/packages/roave/security-advisories>

¹ FriendsOfPHP/security-advisories:
<https://phpa.me/friends-php-advisories>

Assert

Assert is an assertion library which contains a load of useful guard methods for function or method input validation. It's designed to be thin, lightweight, and fast. It can replace all those pesky checks of parameter restrictions with simple throwing of exceptions.

<https://packagist.org/packages/beberlei/assert>

PHPStan

PHPStan is a static analysis tool—another command to add to your CI pipeline, which looks for errors and bugs in your code before they make it into production. For example, things like checking types are correct, ensuring correct visibility of called methods, and extra arguments that are passed to functions and so on. You can configure PHPStan to check at various “levels” for more strict checks if you wish.

<https://packagist.org/packages/phpstan/phpstan>

PHP-Scoper

This nifty, but perhaps niche, little tool will take a load of source code—including dependencies if you wish—and move it to a unique namespace. The age-old problem in PHP of not being able to load two classes with the same name (unless you use an extension like RunKit) means when using a tool distributed as a PHAR, there is a chance the code will conflict with dependencies installed using Composer. In other words, you're using the PHAR version of PHPUnit as well as having installed it with Composer. Therefore, the PHP-Scoper tool is designed primarily for creating and distributing PHAR files to avoid this conflict.

<https://packagist.org/packages/humbug/php-scoper>

zend-code

A handy and powerful code generation tool from the Zend Framework team, zend-code allows you to generate code in an easy-to-understand way—whether it's creating new code, or modifying existing code. An honorary mention should go to Nikita Popov's PHP Parser (nikic/php-parser) library, which also allows you to generate code by constructing AST nodes as the primary mechanism.

<https://packagist.org/packages/zendframework/zend-code>

PHP dotenv

Storing configuration in environment variables is a standard adopted in many applications, with a healthy side-effect of not committing sensitive credentials to your source code repository. PHP dotenv is based on Ruby's dotenv and is a handy library which extracts environment variables for development environments from a .env file into the environment variables and \$_SERVER superglobals.

<https://packagist.org/packages/vlucas/phpdotenv>

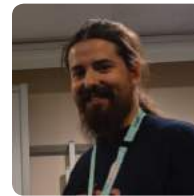
ramsey/uuid

This must be the de-facto standard for generating UUIDs in projects that utilize them. Providing a comprehensive UUIDInterface to allow compatible implementations and Liskov Substitution Principle to be followed, as well as ways to generate UUID version 1, 3, 4 and 5.

<https://packagist.org/packages/ramsey/uuid>

Go and Explore!

This is by no means a fully exhaustive list of useful packages—I just wanted to highlight a few which crossed my path recently—or indeed, a staple part of my PHP toolchain. The sheer number of packages available on Packagist is mind-boggling, and there are no doubt thousands of very creative and useful packages I've missed here. If you're not already using packages like the ones I've listed, and are wondering what's out there—go and explore. Take a look at what's available. You never know, you might save yourself hours of development time when you find someone has already written exactly what you need!



James is a consultant, trainer and developer at Roave. He is a prolific contributor to various open source projects and is a Zend Certified Engineer. He also founded the UK based PHP Hampshire user group and PHP South Coast conference. [@asgrim](#)



Upcoming Events

May

phpDay 2018

May 11–12, Verona, Italy
<https://2018.phpday.it>

PHP Tour Montpellier 2018

May 17–18, Montpellier, France
<https://event.afup.org>

PHPKonf Istanbul PHP Conference 2018

May 20–20, Istanbul, Turkey
<http://phpkonf.org>

PHP Serbia Conference 2018

May 25–27, Belgrade, Serbia
<https://conf2018.phpsrbija.rs>

php[tek] 2018

May 29–June 1, Atlanta, GA
<https://tek.phparch.com>

Texas Camp 2018

May 31–June 2, Austin, TX
<https://2018.texascamp.org>

June

Dutch PHP Conference 2018

June 7–9, Amsterdam, Netherlands
<http://www.phpconference.nl>

Laravel Live UK

CfP Deadline: May 31, 2018
June 12–12, London, England
<https://laravellive.uk>

July

PHPDetroit Conference 2018

July 26–28, Detroit, MI
<https://phpdetroit.io>

September

Cascadia PHP

CfP Deadline: May 31, 2018
September 14–15, Portland, OR
<https://www.cascadiaphp.com>

WavePHP 2018

September 19–21, San Diego, CA
<https://www.wavephp.com>

PHP Developer Days 2018

September 21–22, Dresden, Germany
<https://2018.phpdd.org>

PHPConf.Asia 2018

CfP Deadline: June 1, 2018
September 26–29, NUSS Kent Ridge
Guild House, Singapore
<https://2018.phpconf.asia>

October

ScotlandPHP 2018

CfP Deadline: June 29, 2018
October 5–6, Edinburgh, Scotland
<https://scophp.co/2018>

ZendCon 2018

CfP Deadline: May 25, 2018
October 15–17, hard rock casino
<http://zendcon.com>





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