PHP Developers: start here

Where PHP developers start - what Exponential is as a PHP code base, your first hour, the rules every PHP change follows, and a map of every page for PHP developers, from the request life cycle and PHP 8.0 compatibility…

Source file: `doc/eng-us/help/php-developers/php-developers-start-here.md` Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/php-developers/php-developers-start-here.md

This section of the help is for PHP developers who write PHP code on or for Exponential: kernel fixes, extensions, modules and views, datatypes, persistent objects, fetch functions, template operators, workflow event types, cronjob parts, commands, JSON and REST endpoints, and the tests that go with them. It explains the code base's conventions, the rules the project checks on every change, and how code behaves under each runtime, with real class names, files and commands of the repository.

Every statement on these pages was checked against the code, the settings or the existing documentation of the 6.0.15 line. Where these pages overlap with the pages for application creators and the extension developer book, they link to them instead of repeating them.

Contents

1. Exponential in one paragraph, for PHP developers
2. Your first hour
3. Seven rules for every PHP change
4. Map of the PHP developer pages
5. Pages from other audiences you will use
6. Finding these pages again

Exponential in one paragraph, for PHP developers

Exponential is a free and open source PHP content management system and application framework, licensed under the GNU GPL, version 2 or later. Its code base has been developed since 1999, first under the name eZ Publish, a product its creator discontinued after almost 18 years; 7x kept it alive, and it is led by 7x together with the Exponential Foundation (Who is 7x?, Who is the Exponential Foundation?). For a PHP developer it is a large, mature code base with its own conventions: PHP 8.0 to 8.5 (PHP 7.4 usable without support), generated class maps instead of PSR-4, layered INI settings read through `eZINI`, `eZPersistentObject` for tables, a compiled template language, six supported databases (SQLite 3 by default, PostgreSQL, MongoDB, MariaDB, Oracle, MySQL), and Velocity, an application server with persistent workers, as part of the product (Exponential for PHP developers).

Your first hour

1. **Orient yourself** (10 minutes): Exponential for PHP developers and A tour of the source tree .
2. **Follow one request** (10 minutes): How a request runs in PHP.

3. **Set up a checkout** (20 minutes): Setting up a PHP development checkout, with `make devtools` and `make hooks`.
4. **Learn the rules** (10 minutes): Writing code that runs on PHP 8.0 and The coding standard.
5. **Build something** (10 minutes to start): Write your first module view.

The whole section in order is The PHP developer's reading path.

Seven rules for every PHP change

1. **Runs on PHP 8.0 to 8.5 (PHP 7.4 usable without support).** No syntax or unguarded function newer than 8.0 (Writing code that runs on PHP 8.0).
2. **Follows the house style.** `array()`, spaces inside parentheses, no type declarations; `make phpcs` and `make phpstan` add nothing (The coding standard).
3. **Is named with a prefix.** `exp` for project code, your own for yours; kernel `eZ*` identifiers never change (Class naming and the exp prefix).
4. **Trusts no input and escapes all output.** Casts, allow lists, `escapeString()`, `|wash`, the form token, permission checks in every entry point (Security in PHP code).
5. **Keeps no state between requests.** Correct under Velocity's persistent workers (Static state traps).
6. **Works on every supported database.** Portable SQL or persistent objects (Code for every supported database).
7. **Comes with a test.** DB-less where possible (PHPUnit without a database).

Map of the PHP developer pages

Orientation

Page	What it helps you do
Exponential for PHP developers	Understand the kind of code base you are working in
A tour of the source tree	Find the kernel, libraries, commands, settings and tests
How a request runs in PHP	Know where your code runs and which events you can use
Setting up a PHP development checkout	Get a checkout with the tools and hooks
The PHP developer's reading path	Read the section in order

PHP versions and code rules

Page	What it helps you do
Supported PHP versions: 8.0 to 8.5	Know the range and how each version is tested
Writing code that runs on PHP 8.0	Avoid newer syntax and functions, guard what you need
PHP 8 migration notes for your code	Bring older code up to PHP 8

Page	What it helps you do
The coding standard (phpcs.xml.dist), Why the code has no type declarations	Write in the house style and understand why
Class naming and the exp prefix, Autoloads and class maps	Name classes and make them findable
Reading settings with eZINI	Read configuration correctly

Building blocks

Page	What it helps you do
Write your first module view, Module views as classes	Add pages, and change kernel views without copying them
Write a console command, Plain CLI scripts with eZScript and ezexec	Add command line tools
Write a cronjob part	Schedule work
Write a fetch function, Write a template operator in PHP	Give templates data and helpers
Write a datatype, Write a workflow event type	Add field types and workflow steps
Events, listeners and filters, Extension points from PHP	Plug into the kernel without editing it

Data and APIs

Page	What it helps you do
SQL with eZDB, safely, Code for every supported database	Write safe, portable SQL
Persistent object definitions in depth	Store data in tables of your own
ezjscore JSON endpoints in PHP, REST and remote services from PHP	Serve data over HTTP
Composer for PHP developers	Manage dependencies and package extensions

Security

Page	What it helps you do
Security in PHP code	Apply every security rule in one pass
Validating input and casting, Escaping output from PHP	Handle input and output safely
CSRF protection and ezformtoken	Protect POSTs, also from JavaScript

Debugging and performance

Page	What it helps you do
Debugging with eZDebug, The debug bar for PHP developers	See inside a request; add a debug panel
Logs, the log viewer and request logs	Write and read logs; follow an error reference
Profiling PHP code, OPcache and cache preloading	Measure, speed up and warm

Runtimes

Page	What it helps you do
Velocity persistent workers for PHP developers, Static state traps	Write code that is correct under persistent workers
The kernel service container	Ask the kernel for shared objects by name, define your own, choose their scope
FrankenPHP for PHP developers	Run and test under FrankenPHP
The engine archive (phar) for PHP developers	Keep paths right when the kernel runs from an archive

Tests, quality and contributing

Page	What it helps you do
PHPUnit without a database, Database and live tests	Test your code at the right level
Static analysis with PHPStan, Git hooks and make targets	Run the project's checks locally
CI and reproducing a red check	Fix a red CI check
Contributing PHP changes	Get a change into the kernel

Reference

Page	What it helps you do
Checklists for PHP developers	Check your work in one pass
PHP developer FAQ, Glossary for PHP developers	Quick answers and terms

Pages from other audiences you will use

Page	Why PHP developers read it
Application creators: start here	Building complete applications: content API, queues, REST, deployment, scaling
The extension developer book	The full book for developers of extensions, from first steps to releasing, with an API reference in part 8
Software Engineers: start here	Architecture, evaluation and engineering practice around Exponential
Web Framework Developers: start here	Exponential compared with and connected to other web frameworks
The template language, Template operators	The templates your PHP code feeds
Database support	The six databases, their versions, settings and test coverage
Develop with Git	Working with the repository
Free documentation	What you may do with these pages

Finding these pages again

- In the help, filter by the audience **PHP Developers**, or open **All pages** and the folder `php-developers`.
- Each page has **Edit on GitHub**. If a statement here no longer matches the code, tell us there or on the community issue tracker.

Related pages

- Exponential for PHP developers
- The PHP developer's reading path
- Application creators: start here
- Writing migrations in PHP, Migrations
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.