FAQ für PHP-Entwickler

Kurze Antworten auf die Fragen, die PHP-Entwickler zu Exponential am häufigsten stellen - PHP-Versionen, PSR-4 und Namespaces, Typdeklarationen, Klasse nicht gefunden, nicht angewendete Einstellungen, Änderungen ohne...

Source file: `doc/ger-de/hilfe/php-entwickler/faq-fuer-php-entwickler.md` Last changed 2026-11-07
https://doc.exponential.earth/exphelp/viewer/help/php-developers/php-developer-faq.md

Diese Seite richtet sich an PHP-Entwickler, die eine schnelle Antwort möchten. Jede Antwort verweist auf die Seite mit den Einzelheiten.

> **Warnung `.htaccess` bremst jede Anfrage.** Sind `.htaccess`-Dateien erlaubt (`AllowOverride` alles außer `None`), geschweige denn genutzt, sucht Apache sie bei jeder Anfrage in jedem Verzeichnis des Pfads und liest und wertet sie aus, und die Ladezeiten der Seiten steigen drastisch (gemessen). Die einzige wirkliche Abhilfe ist, die Funktion für den virtuellen Host abzuschalten: `AllowOverride None` im virtuellen HTTPS-Host, und die Regeln in der Konfiguration des virtuellen Hosts selbst als mod_rewrite-Regeln. Das ist in Geschwindigkeitstests im Produktivbetrieb weit schneller und die Zeit für die Optimierung einer Produktivkonfiguration unbedingt wert. Stellen Sie sofort um: Übernehmen Sie die Regeln des Beispiels für einen virtuellen HTTPS-Host in Ihren virtuellen Host (mit dem statischen Cache auch `staticcache_offload.conf`), setzen Sie `AllowOverride None`, führen Sie `apachectl configtest` aus, laden Sie Apache neu und testen Sie die Website. Dieselben Regeln für andere Server: nginx, LiteSpeed, Caddy, IIS, Velocity (eingebaut); wohin Regeln für virtuelle Hosts in einem Control Panel gehören: Plesk, cPanel, DirectAdmin, ISPConfig. **Unter LiteSpeed oder OpenLiteSpeed** kann zusätzlich der Seiten-Cache des Servers anonyme Besucher und Crawler ohne PHP bedienen: Seiten-Caching mit LiteSpeed und OpenLiteSpeed.

Inhalt

1. Plattform und Sprache
2. Klassen und Einstellungen
3. Velocity und Deployments
4. Datenbanken
5. Tests und CI
6. Wo Sie beginnen

Plattform und Sprache

Welche PHP-Versionen muss mein Code unterstützen? PHP 8.0 bis 8.5 (PHP 7.4 ohne Support nutzbar). PHP 8.0 ist die älteste, weil es das Standard-PHP von RHEL 9 ist (Unterstützte PHP-Versionen: 8.0 bis 8.5).

Darf ich Enums, readonly-Eigenschaften oder `foo(...)` verwenden? Nicht in Code, der mit oder für Exponential ausgeliefert wird: PHP 8.0 kann sie nicht parsen (Code schreiben, der unter PHP 8.0 läuft).

Warum kein `declare( strict_types=1 )` **und keine Rückgabetypen?** Der Coding-Standard verbietet sie in neuem Code, zur Konsistenz mit dem vorhandenen Code, zur Kompatibilität von Unterklassen und wegen PHP 8.0 (Warum der Code keine Typdeklarationen hat).

Warum `array()` **und nicht** `[]`**?** Der Hausstil, am vorhandenen Code gemessen und von `phpcs.xml.dist` geprüft (Der Coding-Standard).

Folgt Exponential PSR-4 oder PSR-12? Nein. Klassen werden über generierte Klassen-Maps gefunden, und der Code-Stil ist der eigene des Kernels (PHP-, PHP-FIG- und Composer-Standards).

Darf ich Namespaces verwenden? Ja. Der Kernel verwendet sie für Befehle, Cronjob-Teile, Views und Dienste (`Exponential\Command\...`); andere Klassen sind global mit einem Präfix (Klassenbenennung und das Präfix exp).

Gibt es eine Symfony-Schicht? Exponential 6 läuft auf seinem eigenen Kernel. Die Legacy Bridge führt es innerhalb einer Installation der Symfony-Plattform aus (Symfony und die Legacy Bridge).

Was sind die `ezc*`**-Klassen, und woher kommen sie?** Es sind die Bibliotheken des Kernels (`ezcMail`, `ezcArchive`, `ezcWebdavServer`, `ezcMvcController` ...). Seit 6.0.15 kommen sie aus den **expcomponents** (`se7enxweb/expcomponents-*`), dem Fork der Zeta Components durch Exponential, unter der Apache License 2.0; Klassennamen und APIs haben sich nicht geändert. Ein Fehler in einer davon gehört in das Repository dieses Forks (Exponential Components, Zu expcomponents beitragen).

Meine Erweiterung verlangt `zetacomponents/mail`**. Muss ich das ändern?** Nein: Jeder Fork ersetzt seinen Ursprungsnamen, die Anforderung ist also erfüllt. Benennen Sie sie in Ihrem nächsten Release in `se7enxweb/expcomponents-mail` um (Upgrade-Hinweise: der Wechsel zu expcomponents).

Klassen und Einstellungen

Meine neue Klasse wird „nicht gefunden“. Führen Sie `php bin/php/ezpgenerateautoloads.php -e` aus; unter Velocity starten Sie außerdem die Worker neu (`exp:velocity deploy`) (Autoloads und Klassen-Maps).

Ich habe eine Einstellung geändert, und nichts ist passiert. Leeren Sie den INI-Cache mit `--clear-tag=ini` (nicht `--clear-id=ini`), prüfen Sie mit `exp:ini where`, welche Schicht gewinnt, und starten Sie die Worker von Velocity neu (Einstellungen mit eZINI lesen).

Wie lege ich `.htaccess` **und** `config.php` **an?** `./console exp:setup:htaccess --apply` und `./console exp:setup:config --apply` kopieren sie aus `.htaccess_root` und `config.php-RECOMMENDED`; eine vorhandene, abweichende Datei wird als Diff gezeigt und ohne `--force` behalten (.htaccess und config.php von der Konsole aus einrichten).

Wie ändere ich eine Einstellung in `config.php` **ohne Editor?** `./console exp:config:set <NAME> <value> [--type=bool|int|octal|float|string|null] --apply`. Die Datei wird als PHP-Tokens gelesen, nie ausgeführt; nur der Wert dieses einen `define()` ändert sich, und das Ergebnis muss `php -l` bestehen. Eine Konstante Ihrer eigenen Extension braucht `--allow-unknown`. Unter Velocity starten Sie danach die Worker neu (.htaccess und config.php von der Konsole aus einrichten).

Sollte ich `eZContentObject` **und die anderen** `eZ`**-Klassen umbenennen?** Niemals. Es sind

echte Bezeichner, die jede Erweiterung verwendet; neuer Code verwendet `exp` oder Ihr eigenes Präfix.

Wie ändere ich eine Kernel-View, ohne sie zu kopieren? Leiten Sie von ihrer Klasse ab und registrieren Sie die Unterklasse in `site.ini [RunnableSettings] Implementation[]` (Modul-Views als Klassen).

Velocity und Deployments

Meine Änderung an einer PHP-Datei bewirkt nichts. Unter Velocity behalten die Worker geladene Klassen: `exp:velocity deploy`. Mit eingeschaltetem Engine-Archiv braucht eine Kernel-Änderung einen Neubau, den das Deployment erledigt (Das Engine-Archiv (phar) für PHP-Entwickler).

Eine Seite zeigt die Daten eines anderen Benutzers oder Siteaccess, nur unter Velocity. Zustand, der zwischen Anfragen erhalten bleibt: eine statische Variable in einer Funktion, ein zu knapp geschlüsselter Cache, ein Flag für einmaliges Registrieren (Fallen durch statischen Zustand).

Meine Download-View gibt nach der Datei die Seite aus. `eZExecution::cleanExit()` steht innerhalb eines `catch ( Exception )`. Unter Velocity beendet es die Anfrage mit einer Exception; fangen Sie sie nicht ab (Persistente Velocity-Worker für PHP-Entwickler).

Behält FrankenPHP Zustand zwischen Anfragen? Nein. Es läuft im klassischen Modus, mit einem sauberen Zustand pro Anfrage (FrankenPHP für PHP-Entwickler).

Datenbanken

Auf welcher Datenbank sollte ich entwickeln? SQLite 3 ist der Standard und braucht nichts; testen Sie vor einem Release auch auf einer Server-Datenbank (Code für jede unterstützte Datenbank).

Wie vermeide ich SQL-Injection? `escapeString()` für Strings, `(int)` für Zahlen oder Bedingungen von `eZPersistentObject` (SQL mit eZDB, sicher).

Mein SQL liefert unter MongoDB nichts. Der Adapter übersetzt nur die SQL-Formen des Kernels; nicht übersetzbares SQL liefert ein leeres Ergebnis und protokolliert `MONGO TODO`.

Tests und CI

Brauche ich eine Datenbank, um die Tests auszuführen? Nein. `vendor/bin/phpunit` führt die Suites ohne Datenbank aus; Datenbanktests laufen separat mit `tests/runtests.php --dsn ...` (PHPUnit ohne Datenbank, Datenbank- und Live-Tests).

PHPStan schlägt bei Code fehl, den ich nicht angefasst habe. Nur Befunde, die in `phpstan-baseline.neon` fehlen, schlagen fehl; prüfen Sie, ob Ihre Änderung diesen Code verschoben oder geändert hat (Statische Analyse mit PHPStan).

Die CI ist rot, aber lokal besteht alles. Meistens liest ein Test die Einstellungen Ihrer Installation, oder die CI verwendet ein älteres PHPUnit oder PHP (CI und das Nachstellen einer roten Prüfung).

Der commit-msg-Hook lehnt meinen Commit ab. Beginnen Sie den Betreff mit `Added:` , `Updated:` , `Fixed:` , `Removed:` oder `Renamed:` (Git-Hooks und make-Ziele).

Wo Sie beginnen

Lesen Sie Exponential für PHP-Entwickler, richten Sie einen Checkout ein, und arbeiten Sie dann Schreiben Sie Ihre erste Modul-View durch. Die vollständige Reihenfolge ist Der Lesepfad für PHP-Entwickler.

Verwandte Seiten

- PHP-Entwickler: hier beginnen
- Anwendungsentwickler: häufig gestellte Fragen
- Software-Ingenieure: hier beginnen, Web-Framework-Entwickler: hier beginnen, das Buch für Erweiterungsentwickler
- Glossar für PHP-Entwickler
- Freie Dokumentation: diese Seite darf unter der GNU FDL 1.3 frei kopiert, übersetzt und verbessert werden.

> **Werden Sie Exponential-Partner**
>
> Partner teilen ihre Informationen, stellen Mittel bereit, geben Beiträge zurück und entwickeln die Produkte mit uns, und so wächst Exponential. Eine Partner-Website und ein Dashboard sind für Mitte 2027 geplant: Werden Sie noch heute Partner, damit wir schneller wachsen und neue Funktionen früher liefern.
>
> Partner: hier beginnen | Wie Sie sich bewerben

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.