# Exponential für PHP-Entwickler

Was einem PHP-Entwickler in Exponential begegnet - eine große, ausgereifte PHP-Codebasis mit eigenen Konventionen, generierten Klassen-Maps, geschichteten INI-Einstellungen, persistenten Objekten und einer...

Source file: `doc/ger-de/hilfe/php-entwickler/exponential-fuer-php-entwickler.md` Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/php-developers/exponential-for-php-developers.md

Diese Seite richtet sich an PHP-Entwickler, die neu in der Codebasis von Exponential sind. Sie erklärt, welche Art von PHP-Anwendung Exponential ist, welchen Konventionen Sie in der ersten Stunde begegnen und wo sie sich von den Frameworks unterscheiden, die die meisten PHP-Entwickler heute kennen. Die ausführlichen Seiten dieses Abschnitts bauen darauf auf.

## Inhalt

1. Was die Codebasis ist
2. Fünf Dinge, die PHP-Entwickler überraschen
3. Die Bausteine, die Sie schreiben werden
4. Was das Projekt bei jeder Änderung prüft
5. Wie es weitergeht

## Was die Codebasis ist

Exponential ist ein quelloffenes Content-Management-System und PHP-Anwendungs-Framework, lizenziert unter der GNU GPL, Version 2 oder später (`composer.json`, `"license": "GPL-2.0-or-later"`). Die Codebasis wird seit 1999 entwickelt, zuerst unter dem Namen eZ Publish, einem Produkt, das sein Hersteller nach fast 18 Jahren eingestellt hat; 7x hat es am Leben erhalten, und das Produkt heißt seit August 2025 Exponential (Wer ist 7x?).

Für einen PHP-Entwickler ist diese Geschichte in einer Hinsicht wichtig: Der Code ist älter als Composer, PSR-4 und Namespaces, und er hat seine eigenen Konventionen behalten. Sie sind konsistent und dokumentiert, und die Prüfungen des Projekts setzen sie bei neuem Code durch.

| Teil | Verzeichnis | Was es enthält |
|---|---|---|
| Kernel | `kernel/` | Inhaltsmodell, Module und Views, Datentypen, Workflows, Benutzer und Rollen, der Anfrage-Kernel (`kernel/private/classes/ezpkernelweb.php`) |
| Bibliotheken | `lib/` | allgemeine Bausteine: `lib/ezutils` (`eZINI`, `eZDebug`, `eZSys`, `eZHTTPTool`, `eZExecution`), `lib/ezdb` (`eZDB` und die Treiber), `lib/eztemplate` (die Template-Engine), `lib/ezfile`, `lib/ezi18n` und mehr |
| Erweiterungen | `extension/` | alles, was darauf aufsetzt, mitgeliefert oder eigenes: Module, Datentypen, Operatoren, Designs, Einstellungen |
| Einstellungen | `settings/` | die INI-Standardwerte; Installationen überschreiben sie in `settings/override/` und `settings/siteaccess/` |

| Teil | Verzeichnis | Was es enthält |
|---|---|---|
| Einstiegspunkte | `index.php`, `bin/php/*.php`, `runcronjobs.php` | der Web-Front-Controller, die Kommandozeilenskripte und der Cronjob-Runner |

Einen Rundgang durch jedes Verzeichnis finden Sie unter Ein Rundgang durch den Quellbaum.

## Fünf Dinge, die PHP-Entwickler überraschen

1. **Klassen-Maps, nicht PSR-4.** Klassen werden über generierte Arrays gefunden: `autoload/ezp_kernel.php` für den Kernel und `var/autoload/ezp_extension.php` für die Erweiterungen, geschrieben von `php bin/php/ezpgenerateautoloads.php`. Eine neue Klasse, die nicht in der Map steht, existiert nicht (Autoloads und Klassen-Maps).
2. **Der Kernel-Stil kennt keine Typdeklarationen.** `phpcs.xml.dist` lehnt in neuem Code `declare( strict_types=1 )`, skalare Parametertypen, Rückgabetypen und Eigenschaftstypen ab, damit er zum vorhandenen Code passt und PHP 8.0 weiter funktioniert (Warum der Code keine Typdeklarationen hat).
3. **`array( ... )`, nicht `[ ... ]`, und Leerzeichen innerhalb von Klammern.** `foo( $a, $b )`, `if ( $a )`, die öffnende geschweifte Klammer einer Funktion in einer eigenen Zeile ( Der Coding-Standard).
4. **Einstellungen sind INI-Dateien in Schichten.** `eZINI::instance( 'site.ini' )->variable( 'Block', 'Key' )` liest den Wert, nachdem die Kernel-Standardwerte, die Erweiterungen, der Siteaccess und die Überschreibungen der Installation zusammengeführt wurden (Einstellungen mit eZINI lesen).
5. **Viel Zustand ist statisch.** Singletons wie `eZINI::instance()`, `eZDB::instance()` und In-Memory-Caches leben in statischen Eigenschaften. Unter PHP-FPM beginnt jede Anfrage frisch; unter den persistenten Workern von Velocity überleben sie zwischen Anfragen, sofern der Kernel sie nicht zurücksetzt (Fallen durch statischen Zustand).

## Die Bausteine, die Sie schreiben werden

| Sie möchten | Sie schreiben | Seite |
|---|---|---|
| einen neuen Bildschirm unter `/<module>/<view>` | ein Modul mit Views | Schreiben Sie Ihre erste Modul-View |
| ein Kommandozeilenwerkzeug | eine Klasse, die `Exponential\Runnable\Command` erweitert | Einen Konsolenbefehl schreiben |
| zeitlich geplante Arbeit | einen Cronjob-Teil | Einen Cronjob-Teil schreiben |
| Daten für Templates | eine Fetch-Funktion | Eine Fetch-Funktion schreiben |
| einen Template-Filter oder eine Hilfsfunktion | eine Template-Operator-Klasse | Einen Template-Operator in PHP schreiben |
| einen neuen Feldtyp | eine Klasse, die `eZDataType` erweitert | Einen Datentyp schreiben |
| einen Schritt in einem Workflow | eine Klasse, die `eZWorkflowEventType` erweitert | Einen Workflow-Ereignistyp schreiben |

| Sie möchten | Sie schreiben | Seite |
|---|---|---|
| Daten in einer eigenen Tabelle | eine Klasse, die `eZPersistentObject` erweitert | Definitionen persistenter Objekte im Detail |
| JSON für Ihr eigenes JavaScript | eine ezjscore-Serverfunktion | ezjscore-JSON-Endpunkte in PHP |
| eine Reaktion auf etwas, das der Kernel tut | einen Event-Listener | Ereignisse, Listener und Filter |

## Was das Projekt bei jeder Änderung prüft

| Prüfung | Werkzeug | Seite |
|---|---|---|
| PHP-Syntax, auch mit PHP 8.0 | `php -l`, `make lint80` | Code schreiben, der unter PHP 8.0 läuft |
| Code-Stil geänderter Dateien | `make phpcs` (`phpcs.xml.dist`) | Der Coding-Standard |
| Statische Analyse | `make phpstan` (Level 1 mit einer Baseline) | Statische Analyse mit PHPStan |
| Unit-Tests | `vendor/bin/phpunit` | PHPUnit ohne Datenbank |
| Das Datei-Manifest | `make manifest-check` | Git-Hooks und make-Ziele |

## Wie es weitergeht

- Folgen Sie dem Lesepfad für PHP-Entwickler in Reihenfolge.
- Richten Sie einen Checkout ein: Einen PHP-Entwicklungs-Checkout einrichten.
- Lernen Sie die Anfrage kennen: Wie eine Anfrage in PHP abläuft.

## Verwandte Seiten

- PHP-Entwickler: hier beginnen
- Kernel-Architektur
- Anwendungsentwickler: hier beginnen, das Buch für Erweiterungsentwickler
- Software-Ingenieure: hier beginnen, Web-Framework-Entwickler: hier beginnen
- 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.