PHP-Entwickler: hier beginnen

Wo PHP-Entwickler beginnen - was Exponential als PHP-Codebasis ist, Ihre erste Stunde, die Regeln, denen jede PHP-Änderung folgt, und eine Übersicht aller Seiten für PHP-Entwickler, vom Lebenszyklus einer Anfrage und…

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

Dieser Abschnitt der Hilfe richtet sich an PHP-Entwickler, die PHP-Code an oder für Exponential schreiben: Kernel-Korrekturen, Erweiterungen, Module und Views, Datentypen, persistente Objekte, Fetch-Funktionen, Template-Operatoren, Workflow-Ereignistypen, Cronjob-Teile, Befehle, JSON- und REST-Endpunkte sowie die zugehörigen Tests. Er erklärt die Konventionen der Codebasis, die Regeln, die das Projekt bei jeder Änderung prüft, und wie sich Code unter jeder Laufzeitumgebung verhält, mit echten Klassennamen, Dateien und Befehlen des Repositorys.

Jede Aussage auf diesen Seiten wurde am Code, an den Einstellungen oder an der vorhandenen Dokumentation der Linie 6.0.15 geprüft. Wo sich diese Seiten mit den Seiten für Anwendungsentwickler und dem Buch für Erweiterungsentwickler überschneiden, verweisen sie dorthin, statt sie zu wiederholen.

Inhalt

1. Exponential in einem Absatz, für PHP-Entwickler
2. Ihre erste Stunde
3. Sieben Regeln für jede PHP-Änderung
4. Übersicht der Seiten für PHP-Entwickler
5. Seiten anderer Zielgruppen, die Sie nutzen werden
6. Diese Seiten wiederfinden

Exponential in einem Absatz, für PHP-Entwickler

Exponential ist ein freies und quelloffenes PHP-Content-Management-System und Anwendungs-Framework, lizenziert unter der GNU GPL, Version 2 oder später. Seine 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 es wird von 7x gemeinsam mit der Exponential Foundation geführt (Wer ist 7x?, Wer ist die Exponential Foundation?). Für einen PHP-Entwickler ist es eine große, ausgereifte Codebasis mit eigenen Konventionen: PHP 8.0 bis 8.5 (PHP 7.4 ohne Support nutzbar), generierte Klassen-Maps statt PSR-4, geschichtete INI-Einstellungen, die über `eZINI` gelesen werden, `eZPersistentObject` für Tabellen, eine kompilierte Template-Sprache, sechs unterstützte Datenbanken (standardmäßig SQLite 3, PostgreSQL, MongoDB, MariaDB, Oracle, MySQL) und Velocity, ein Anwendungsserver mit persistenten Workern, als Teil des Produkts (Exponential für PHP-Entwickler).

Ihre erste Stunde

1. **Orientieren Sie sich** (10 Minuten): Exponential für PHP-Entwickler und Ein Rundgang durch den Quellbaum.
2. **Verfolgen Sie eine Anfrage** (10 Minuten): Wie eine Anfrage in PHP abläuft.
3. **Richten Sie einen Checkout ein** (20 Minuten): Einen PHP-Entwicklungs-Checkout einrichten, mit `make devtools` und `make hooks`.
4. **Lernen Sie die Regeln** (10 Minuten): Code schreiben, der unter PHP 8.0 läuft und Der Coding-Standard.
5. **Bauen Sie etwas** (10 Minuten für den Anfang): Schreiben Sie Ihre erste Modul-View.

Den gesamten Abschnitt in Reihenfolge finden Sie unter Der Lesepfad für PHP-Entwickler.

Sieben Regeln für jede PHP-Änderung

1. **Läuft unter PHP 8.0 bis 8.5 (PHP 7.4 ohne Support nutzbar).** Keine Syntax und keine ungeschützte Funktion, die neuer als 8.0 ist (Code schreiben, der unter PHP 8.0 läuft).
2. **Folgt dem Hausstil.** `array()`, Leerzeichen innerhalb von Klammern, keine Typdeklarationen; `make phpcs` und `make phpstan` melden nichts Neues (Der Coding-Standard).
3. **Trägt ein Präfix im Namen.** `exp` für Projektcode, Ihr eigenes für Ihren Code; Kernel-Bezeichner `eZ*` ändern sich nie (Klassenbenennung und das Präfix exp).
4. **Vertraut keiner Eingabe und maskiert jede Ausgabe.** Casts, Allow-Listen, `escapeString()`, `|wash`, das Formular-Token, Berechtigungsprüfungen an jedem Einstiegspunkt (Sicherheit in PHP-Code).
5. **Behält keinen Zustand zwischen Anfragen.** Korrekt unter den persistenten Workern von Velocity (Fallen durch statischen Zustand).
6. **Funktioniert auf jeder unterstützten Datenbank.** Portables SQL oder persistente Objekte (Code für jede unterstützte Datenbank).
7. **Kommt mit einem Test.** Wo möglich ohne Datenbank (PHPUnit ohne Datenbank).

Übersicht der Seiten für PHP-Entwickler

Orientierung

Seite	Wobei sie Ihnen hilft
Exponential für PHP-Entwickler	Verstehen, in welcher Art von Codebasis Sie arbeiten
Ein Rundgang durch den Quellbaum	Kernel, Bibliotheken, Befehle, Einstellungen und Tests finden
Wie eine Anfrage in PHP abläuft	Wissen, wo Ihr Code läuft und welche Ereignisse Sie nutzen können
Einen PHP-Entwicklungs-Checkout einrichten	Einen Checkout mit den Werkzeugen und Hooks bekommen
Der Lesepfad für PHP-Entwickler	Den Abschnitt in Reihenfolge lesen

PHP-Versionen und Code-Regeln

Seite	Wobei sie Ihnen hilft
Unterstützte PHP-Versionen: 8.0 bis 8.5	Den Bereich kennen und wissen, wie jede Version getestet wird
Code schreiben, der unter PHP 8.0 läuft	Neuere Syntax und Funktionen vermeiden, Benötigtes absichern
Hinweise zur PHP-8-Migration für Ihren Code	Älteren Code auf PHP 8 bringen
Der Coding-Standard (phpcs.xml.dist), Warum der Code keine Typdeklarationen hat	Im Hausstil schreiben und verstehen, warum
Klassenbenennung und das Präfix exp, Autoloads und Klassen-Maps	Klassen benennen und auffindbar machen
Einstellungen mit eZINI lesen	Konfiguration korrekt lesen

Bausteine

Seite	Wobei sie Ihnen hilft
Schreiben Sie Ihre erste Modul-View, Modul-Views als Klassen	Seiten hinzufügen und Kernel-Views ändern, ohne sie zu kopieren
Einen Konsolenbefehl schreiben, Einfache CLI-Skripte mit eZScript und ezexec	Kommandozeilenwerkzeuge hinzufügen
Einen Cronjob-Teil schreiben	Arbeit zeitlich planen
Eine Fetch-Funktion schreiben, Einen Template-Operator in PHP schreiben	Templates Daten und Hilfsfunktionen geben
Einen Datentyp schreiben, Einen Workflow-Ereignistyp schreiben	Feldtypen und Workflow-Schritte hinzufügen
Ereignisse, Listener und Filter, Erweiterungspunkte aus PHP	Sich in den Kernel einklinken, ohne ihn zu bearbeiten

Daten und APIs

Seite	Wobei sie Ihnen hilft
SQL mit eZDB, sicher, Code für jede unterstützte Datenbank	Sicheres, portables SQL schreiben
Definitionen persistenter Objekte im Detail	Daten in eigenen Tabellen speichern
ezjscore-JSON-Endpunkte in PHP, REST und entfernte Dienste aus PHP	Daten über HTTP ausliefern
Composer für PHP-Entwickler	Abhängigkeiten verwalten und Erweiterungen paketieren

Sicherheit

Seite	Wobei sie Ihnen hilft
Sicherheit in PHP-Code	Jede Sicherheitsregel in einem Durchgang anwenden

Seite	Wobei sie Ihnen hilft
Eingaben validieren und casten, Ausgaben aus PHP maskieren	Ein- und Ausgaben sicher behandeln
CSRF-Schutz und ezformtoken	POSTs schützen, auch aus JavaScript

Debugging und Performance

Seite	Wobei sie Ihnen hilft
Debugging mit eZDebug, Die Debug-Leiste für PHP-Entwickler	In eine Anfrage hineinsehen; ein Debug-Panel hinzufügen
Logs, der Log-Viewer und Anfrage-Logs	Logs schreiben und lesen; einer Fehlerreferenz folgen
PHP-Code profilieren, OPcache und Cache-Vorladen	Messen, beschleunigen und vorwärmen

Laufzeitumgebungen

Seite	Wobei sie Ihnen hilft
Persistente Velocity-Worker für PHP-Entwickler, Fallen durch statischen Zustand	Code schreiben, der unter persistenten Workern korrekt ist
Der Kernel-Service-Container	Den Kernel nach gemeinsam genutzten Objekten fragen, eigene definieren, ihren Gültigkeitsbereich wählen
FrankenPHP für PHP-Entwickler	Unter FrankenPHP ausführen und testen
Das Engine-Archiv (phar) für PHP-Entwickler	Pfade korrekt halten, wenn der Kernel aus einem Archiv läuft

Tests, Qualität und Mitwirken

Seite	Wobei sie Ihnen hilft
PHPUnit ohne Datenbank, Datenbank- und Live-Tests	Ihren Code auf der richtigen Ebene testen
Statische Analyse mit PHPStan, Git-Hooks und make-Ziele	Die Prüfungen des Projekts lokal ausführen
CI und das Nachstellen einer roten Prüfung	Eine rote CI-Prüfung beheben
PHP-Änderungen beitragen	Eine Änderung in den Kernel bringen

Referenz

Seite	Wobei sie Ihnen hilft
Checklisten für PHP-Entwickler	Ihre Arbeit in einem Durchgang prüfen
FAQ für PHP-Entwickler, Glossar für PHP-Entwickler	Schnelle Antworten und Begriffe

Seiten anderer Zielgruppen, die Sie nutzen werden

Seite	Warum PHP-Entwickler sie lesen

Seite	Warum PHP-Entwickler sie lesen
Anwendungsentwickler: hier beginnen	Vollständige Anwendungen bauen: Content-API, Warteschlangen, REST, Deployment, Skalierung
Das Buch für Erweiterungsentwickler	Das vollständige Buch für Entwickler von Erweiterungen, von den ersten Schritten bis zum Release, mit einer API-Referenz in Teil 8
Software-Ingenieure: hier beginnen	Architektur, Evaluierung und Engineering-Praxis rund um Exponential
Web-Framework-Entwickler: hier beginnen	Exponential im Vergleich zu und in Verbindung mit anderen Web-Frameworks
Die Template-Sprache, Template-Operatoren	Die Templates, die Ihr PHP-Code mit Daten versorgt
Datenbankunterstützung	Die sechs Datenbanken, ihre Versionen, Einstellungen und Testabdeckung
Mit Git entwickeln	Mit dem Repository arbeiten
Freie Dokumentation	Was Sie mit diesen Seiten tun dürfen

Diese Seiten wiederfinden

- Filtern Sie in der Hilfe nach der Zielgruppe **PHP-Entwickler**, oder öffnen Sie **Alle Seiten** und den Ordner `php-developers`.
- Jede Seite hat **Auf GitHub bearbeiten**. Wenn eine Aussage hier nicht mehr zum Code passt, sagen Sie es uns dort oder im Issue-Tracker der Community.

Verwandte Seiten

- Exponential für PHP-Entwickler
- Der Lesepfad für PHP-Entwickler
- Anwendungsentwickler: hier beginnen
- Migrationen in PHP schreiben, Migrationen
- 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.