Exponential für Web-Framework-Entwickler

Was Exponential ist, wenn Sie es mit den Augen eines Symfony-, Laravel-, Django- oder Rails-Entwicklers betrachten – ein PHP-Anwendungsframework mit eingebautem Inhaltsmodell, Berechtigungen, Templates und Server –…

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

Diese Seite richtet sich an Entwickler, die ein Web-Framework kennen und sich ein erstes, ehrliches Bild von Exponential machen möchten. Die README beschreibt es als „ein professionelles PHP-Anwendungsframework mit fortgeschrittener CMS-Funktionalität (Content-Management-System)“, und beide Hälften dieses Satzes zählen: Darunter liegt ein Framework (Front Controller, Routing zu Modulen und Views, Templates, eine Datenbankschicht, Einstellungen, eine Konsole), und darüber ein Inhaltsmodell mit Versionen, Übersetzungen, Berechtigungen und einer Administrationsoberfläche, die Sie bekommen, ohne Code zu schreiben.

Inhalt

1. In einem Absatz
2. Was vertraut wirkt
3. Was anders ist
4. Wo die Dinge liegen
5. Wie Sie Code hinzufügen
6. Wie es weitergeht

In einem Absatz

Exponential 6 ist eine PHP-Codebasis mit einer langen Geschichte: Sie wurde ab 1999 entwickelt, zuerst unter dem Namen eZ Publish, den sein Hersteller nach fast 18 Jahren eingestellt hat; 7x hat die Entwicklung fortgesetzt, und seit August 2025 heißt das Produkt Exponential (Wer ist 7x?). Es wird mit Composer installiert, läuft unter PHP 8, speichert Daten in SQLite, PostgreSQL, MySQL/MariaDB, Oracle oder MongoDB (Datenbankunterstützung) und wird von Apache oder nginx mit PHP-FPM, von Exponential Velocity (einem in PHP geschriebenen Server mit persistenten Workern, der seit 6.0.15 Teil des Produkts ist) oder von FrankenPHP ausgeliefert (Velocity ist Teil von Exponential 6.0.15).

Was vertraut wirkt

Sie kennen	Exponential hat
Einen Front Controller (`public/index.php`)	`index.php`, das `ezpKernel( new ezpKernelWeb() )` aufbaut und dessen Inhalt zurückgibt

Sie kennen	Exponential hat
Routen zu Controller-Actions	Module und Views: `/<module>/<view>/<params>`, deklariert in `module.php` (Routing)
Eine Template-Engine	die Template-Sprache (`.tpl`-Dateien, die `eZTemplate` zu PHP kompiliert) (Die View-Schicht)
Ein ORM oder Active Record	`eZPersistentObject` für Ihre eigenen Tabellen und die Content-API für Inhalte (ORM)
Eine Konsole	`bin/php/console` mit den Befehlen `exp:*`, `ext:*` und `cron:*` (Kommandozeile)
Pakete	Erweiterungen, installiert als Composer-Pakete (Pakete)
Konfigurationsdateien	INI-Dateien in Ebenen (Konfiguration)
Tests	PHPUnit, die meisten Suiten ohne Datenbank (Testen)

Was anders ist

- **Inhalte sind ein vollwertiges Modell.** Die meisten Daten, die ein Redakteur bearbeitet, sind keine Tabelle, die Sie entwerfen, sondern eine *Inhaltsklasse* (ein Typ mit Attributen), deren *Objekte Versionen* und *Übersetzungen* haben und an *Knoten* in einem Baum hängen. Sie definieren Klassen in der Administrationsoberfläche und bekommen Bearbeitung, Historie, Berechtigungen und Suche ohne Zusatzaufwand (Das Inhaltsmodell für ORM-Nutzer).
- **Siteaccesses statt einer einzigen Anwendungskonfiguration.** Ein Request wird zuerst einem *Siteaccess* zugeordnet (einer Site, einer Sprachversion, der Administrationsoberfläche), und die Einstellungen dieses Siteaccess werden vor dem Routing geladen (Siteaccesses).
- **Kein allgemeiner Dependency-Injection-Container.** Der Legacy-Kernel findet Klassen über INI-Einstellungen, Namenskonventionen und eine erzeugte Klassenzuordnung. Läuft er über die Legacy Bridge innerhalb von Symfony, kann er den Container von Symfony erhalten, aber der Kernel selbst definiert darin keine Dienste (Dependency Injection).
- **Templates werden überschrieben, nicht erweitert.** Eine Designkette und Regeln in `override.ini` entscheiden, welches Template eine View rendert; Blockvererbung wie in Twig oder Blade gibt es nicht (Template-Auflösung).
- **Berechtigungen sind Daten.** Rollen aus Richtlinien (Modul, Funktion, Einschränkungen) werden in der Datenbank gespeichert und in der Administrationsoberfläche zugewiesen; Ihr Modul deklariert die Funktionen, Administratoren vergeben sie (Autorisierung).
- **Schemaänderungen sind keine Migrationsklassen.** Tabellen werden in datenbankneutralen `.dba`-Dateien beschrieben; Upgrades sind Skripte, die Sie mitliefern (Migrationen).

Wo die Dinge liegen

Aus Architektur für Anwendungsersteller:

```
kernel/, lib/      the kernel: modules, classes, the template engine, the database layer
autoload/          the kernel's class map (autoload/ezp_kernel.php)
extension/         extensions: your application lives here
design/            designs: templates, stylesheets, images
settings/          shipped settings; settings/siteaccess/ and settings/override/ hold yours
var/               runtime: caches, logs, storage, the SQLite database
bin/php/console    one entry point for every command
cronjobs/          the scripts runcronjobs.php runs
index.php          the front controller
```

Wie Sie Code hinzufügen

Sie bearbeiten nie `kernel/` oder `lib/`. Alles, was Sie schreiben, kommt in eine **Erweiterung**, die Einheit, die die Rolle eines Symfony-Bundles, eines Laravel-Pakets oder einer Django-App übernimmt: Module (Ihre Routen und Bildschirme), Klassen, Templates, Einstellungen, Datentypen, Workflow-Ereignistypen, Cronjob-Teile und Konsolenbefehle. **Setup > RAD** (`/setup/rad`) listet jeden Erweiterungspunkt auf und schreibt für die meisten davon den Grundcode (RAD-Werkzeuge, Erweiterungspunkte).

Wie es weitergeht

1. Begriffswörterbuch: die Begriffe Ihres Frameworks und ihre Gegenstücke in Exponential.
2. Der Request-Lebenszyklus: was zwischen `index.php` und der Response geschieht.
3. Aufgabe: ein kleines CRUD-Modul bauen: der ganze Weg einmal von Hand.
4. Die Umstiegsseiten: Symfony, Laravel, Slim, CodeIgniter, Yii, CakePHP oder Laminas, Django und Rails, Express und Spring.

Verwandte Seiten

- Web-Framework-Entwickler: hier beginnen
- Exponential als Anwendungsplattform
- Kernel-Architektur
- Erweiterungsentwicklung: das Buch
- 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.