Web-Framework-Entwickler: hier beginnen

Wo Entwickler, die Symfony, Laravel, Slim, CodeIgniter, Yii, CakePHP, Laminas, Django, Rails, Express oder Spring kennen, mit Exponential beginnen - was es in ihren Begriffen ist, die erste Stunde, eine Übersicht aller…

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

Dieser Bereich der Hilfe richtet sich an Entwickler, die ein Web-Framework kennen (Symfony, Laravel, Slim, CodeIgniter, Yii, CakePHP, Laminas, Django, Rails, Express, Spring) und auf Exponential aufbauen möchten. Exponential beschreibt sich selbst als „a professional PHP application framework with advanced CMS (content management system) functionality“ (README), also als professionelles PHP-Anwendungsframework mit fortgeschrittener CMS-Funktionalität. Diese Seiten übertragen, was Sie bereits kennen, in die Begriffe von Exponential, sagen klar, wo etwas anders funktioniert, und sagen ebenso klar, wo Exponential keine direkte Entsprechung hat.

Jede Klasse, jede Einstellung und jeder Befehl, die auf diesen Seiten genannt werden, wurde am Kernel, an den Einstellungen oder an der vorhandenen Dokumentation geprüft. Wo eine Seite mit einem anderen Framework vergleicht, beschreibt sie, wofür dieses Framework bekannt ist, ohne es zu bewerten.

Inhalt

1. Exponential in einem Absatz
2. Ihre erste Stunde
3. Fünf Dinge, die anders sind
4. Übersicht der Seiten
5. Seiten anderer Zielgruppen, die Sie nutzen werden
6. Diese Seiten wiederfinden

Exponential in einem Absatz

Exponential 6 ist ein freies, quelloffenes PHP-Anwendungsframework und Content-Management-System unter der GNU GPL, Version 2 oder später. Ein Request kommt über `index.php` herein, wird einem **Siteaccess** zugeordnet (einer benannten Konfiguration pro Site, Sprache oder Administrationsoberfläche), von einem URL-Alias in eine Adresse der Form `/<module>/<view>` übersetzt, gegen **Rollen und Richtlinien** geprüft, von einer **Modul-View** bearbeitet, mit der **Template-Sprache** gerendert und in das Seitenlayout eingesetzt. Daten, die Redakteure pflegen, liegen in **Inhaltsklassen** mit Versionen und Übersetzungen; Ihre eigenen Tabellen verwenden `eZPersistentObject`. Die Konfiguration besteht aus **INI-Dateien in Ebenen**; der Code liegt in **Erweiterungen**, die mit Composer installiert werden. Ausgeliefert wird über Apache oder nginx mit PHP-FPM, über **Exponential Velocity**, einen

Server mit persistenten Workern, der seit 6.0.15 Teil des Produkts ist, oder über FrankenPHP. Die Codebasis wurde ab 1999 entwickelt, zunächst unter dem Namen eZ Publish, den sein Hersteller nach fast 18 Jahren eingestellt hat; 7x hat sie als Exponential fortgeführt (Wer ist 7x?).

Ihre erste Stunde

1. **Installieren** (10 Minuten): `./console exp:install --apply` richtet eine Site auf SQLite ein (Installation mit einem Befehl).
2. **Orientieren** (10 Minuten): Exponential für Web-Framework-Entwickler und Ihre Umstiegsseite (Symfony, Laravel, Slim, CodeIgniter, Yii, CakePHP oder Laminas, Django oder Rails, Express oder Spring).
3. **Übertragen** (10 Minuten): das Begriffswörterbuch.
4. **Einem Request folgen** (10 Minuten): Der Request-Lebenszyklus.
5. **Etwas bauen** (20 Minuten für den Anfang): Aufgabe: ein kleines CRUD-Modul bauen.

Fünf Dinge, die anders sind

1. **Inhalte sind das Modell.** Redaktionelle Daten sind eine Inhaltsklasse mit Versionen, Übersetzungen und Berechtigungen, keine Tabelle, die Sie entwerfen (Das Inhaltsmodell für ORM-Nutzer).
2. **Siteaccess vor dem Routing.** Pro Request wird eine ganze Konfiguration ausgewählt (Siteaccesses).
3. **Kein Service-Container.** Handler werden in INI-Einstellungen benannt (Dependency Injection).
4. **Templates escapen nicht automatisch** und werden überschrieben statt erweitert (Die View-Schicht).
5. **Persistente Worker.** Unter Velocity darf Code keinen Request-bezogenen Zustand in statischen Eigenschaften halten (Server und Deployment).

Übersicht der Seiten

Orientierung

Seite	Wobei sie Ihnen hilft
Exponential für Web-Framework-Entwickler	Das Gesamtbild in Framework-Begriffen sehen
Begriffswörterbuch	Den Begriff Ihres Frameworks nachschlagen und den von Exponential finden
Lesepfad	Die Seiten in einer sinnvollen Reihenfolge lesen
Wann Sie Exponential wählen sollten und wann ein reines Framework	Für ein Projekt fair entscheiden

Umstieg von einem anderen Framework

Seite	Wobei sie Ihnen hilft
Umstieg von Symfony	Bundles, Kernel-Ereignisse, Doctrine, Forms und Security übertragen
Umstieg von Laravel	Routen, Eloquent, Blade, Middleware, Queues und Artisan übertragen
Umstieg von Slim, CodeIgniter, Yii, CakePHP oder Laminas	Gewohnheiten aus klassischem MVC und Micro-Frameworks übertragen
Umstieg von Django oder Rails	URLconf, ORMs, die Administration, Signale und Befehle übertragen
Umstieg von Express oder Spring	Das Prozessmodell von PHP verstehen und den Rest übertragen

Der Request

Seite	Wobei sie Ihnen hilft
Der Request-Lebenszyklus	Wissen, wo Ihr Code läuft, von `index.php` bis zu den Ausgabefiltern
Siteaccesses: einen Request einer Site zuordnen	Sites, Sprachen und die Administrationsoberfläche pro Request konfigurieren
Routing: Module, Views und Parameter	Adressen deklarieren und URLs erzeugen
Sprechende URLs: URL-Aliase und URL-Platzhalter	Slugs, Weiterleitungen und Umschreibungen ohne Code nutzen
Controller: Modul-Views	Eine View als Skript oder Klasse schreiben und Kernel-Views ändern
Responses, Weiterleitungen, Fehler und JSON	Seiten, Weiterleitungen, Fehler und JSON zurückgeben

Views und Templates

Seite	Wobei sie Ihnen hilft
Die View-Schicht: die Template-Sprache neben Twig und Blade	Templates schreiben und die klassischen Fehler vermeiden
Template-Auflösung: Designs und Overrides	Wissen, welche Datei rendert, und sie überschreiben
Daten in Templates bringen: Fetch-Funktionen und Operatoren	Templates Daten und Hilfsfunktionen bereitstellen

Daten

Seite	Wobei sie Ihnen hilft
Das Inhaltsmodell für ORM-Nutzer	Redaktionelle Daten als Klassen, Objekte, Versionen und Knoten modellieren
ORM und Active Record: eZPersistentObject	Eigene Tabellen auf Klassen abbilden

Seite	Wobei sie Ihnen hilft
Abfragen: Content-Fetches, eZDB und SQL	Inhalte und Tabellen portabel abfragen
Migrationen und Startdaten: .dba-Schemadateien und Pakete	Schemas und Startdaten ausliefern und aktualisieren

Konfiguration und Verdrahtung

Seite	Wobei sie Ihnen hilft
Konfiguration: INI-Dateien statt YAML, PHP-Arrays oder .env	Einstellungen lesen, schichten und ändern
Umgebungen und Geheimnisse: EXP_ENV und settings/override	Entwicklung, Test und Produktion betreiben und Geheimnisse aus Git heraushalten
Dependency Injection und Dienste: was es gibt	Testbaren Code ohne Container verdrahten
Erweiterungspunkte: Handler, INI-Registrierung und Overrides	Die richtige Stelle zum Einklinken finden
Pakete: Erweiterungen, Composer und Bundles	Ihren Code strukturieren und verteilen

Request-Verarbeitung, Ereignisse und Sicherheit

Seite	Wobei sie Ihnen hilft
Middleware: Request-Regeln, Eingabe- und Ausgabe-Listener	Das tun, was Middleware tut
Ereignisse und Listener: ezpEvent	Kernel-Ereignisse abonnieren und eigene auslösen
Workflows und Trigger: Einstiegspunkte an Inhaltsoperationen	Schritte vor und nach dem Veröffentlichen und dem Bestellabschluss ausführen
Authentifizierung und Sitzungen	Benutzer und Maschinen anmelden
Autorisierung: Rollen, Richtlinien und Einschränkungen	Berechtigungen deklarieren und prüfen
Formulare und CSRF-Schutz	Formulare sicher verarbeiten
Validierung: Eingaben, Datentypen und Escaping	Eingaben validieren und Ausgaben escapen

Performance und Sprachen

Seite	Wobei sie Ihnen hilft
Caching: die Ebenen und wie man sie invalidiert	Richtig cachen und das Richtige leeren
HTTP-Caching und Reverse Proxies	Header, den HTTP-Cache und den Response-Cache nutzen
Übersetzungen, Locales und Rechts-nach-links-Schrift	Die Oberfläche und die Inhalte übersetzen

Hintergrundarbeit, APIs und Betrieb

Seite	Wobei sie Ihnen hilft
Kommandozeile: die Konsole und eZScript	Befehle ausführen und schreiben
Warteschlangen und Hintergrundjobs	Langsame Arbeit aus Requests auslagern
Geplante Aufgaben: Cronjobs	Arbeit nach Zeitplan ausführen
REST- und JSON-APIs	Die richtige API-Schicht wählen
Headless und entkoppelte Frontends	Inhalte an ein anderes Frontend ausliefern
Testen: PHPUnit, Datenbanktests und Prüfungen	Ohne Angst testen
Debugging, Logging und Fehler	In einen Request hineinsehen und die Logs lesen
Server und Deployment: Apache und PHP-FPM, Velocity, FrankenPHP	Ausliefern und deployen

Exponential mit anderen Frameworks

Seite	Wobei sie Ihnen hilft
Exponential und ein Framework kombinieren	Beides in einem Projekt verbinden
Die Symfony Legacy Bridge für Framework-Entwickler	Exponential 6 innerhalb von Symfony betreiben
Die Produkte der Exponential Platform	Eine auf Symfony basierende Distribution wählen

Praxis, Prüfungen und Referenz

Seite	Wobei sie Ihnen hilft
Aufgabe: ein kleines CRUD-Modul Schritt für Schritt bauen	Jedes Teil einmal von Hand bauen
Zehn Stolperfallen für Framework-Entwickler	Die Fehler vermeiden, die alle zuerst machen
Checklisten für Framework-Entwickler	Bewerten, einrichten, prüfen und live gehen
FAQ für Web-Framework-Entwickler	Schnelle Antworten erhalten
Glossar für Web-Framework-Entwickler	Einen Begriff von Exponential nachschlagen

Seiten anderer Zielgruppen, die Sie nutzen werden

Seite	Warum
Anwendungsentwickler: hier beginnen	Die ausführlichen Leitfäden hinter den meisten Seiten hier: Module, Datentypen, Workflows, Warteschlangen, REST, Deployment
Erweiterungsentwicklung: das Buch	Jeder Erweiterungspunkt Schritt für Schritt, vom Benutzer bis zum Maintainer
PHP-Entwickler: hier beginnen	Die PHP-Seite: Sprache, Konventionen und die Klassen des Kernels
Softwareingenieure: hier beginnen	Architektur, Qualität und Ingenieurspraxis
Die Template-Sprache	Die Sprache im Detail, mit Vergleichen zu Smarty

Seite	Warum
Kernel-Architektur	Die maßgebliche Beschreibung der Schichten des Kernels
Erweiterungspunkte	Der Katalog aller Erweiterungspunkte
Datenbankunterstützung	Die unterstützten Datenbanken und wie jede getestet wird

Diese Seiten wiederfinden

Filtern Sie in der Hilfe nach der Zielgruppe **Web-Framework-Entwickler**; die Seiten liegen im Ordner `web-framework-developers` (in der deutschen Hilfe `hilfe/web-framework-entwickler`). Suchen Sie nach dem Namen eines Frameworks („Symfony“, „Laravel“, „Django“), um den Vergleich zu finden, der es erwähnt.

Verwandte Seiten

- Erste Schritte
- Wenn etwas schiefgeht
- Mit Git entwickeln
- 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.