Web framework developers: start here

Where developers who know Symfony, Laravel, Slim, CodeIgniter, Yii, CakePHP, Laminas, Django, Rails, Express or Spring start with Exponential - what it is in their terms, the first hour, a map of every page that...

Source file: `doc/eng-us/help/web-framework-developers/web-framework-developers-start-here.md` Last changed 2026-11-04
https://doc.exponential.earth/exphelp/viewer/help/web-framework-developers/web-framework-developers-start-here.md

This section of the help is for developers who know a web framework (Symfony, Laravel, Slim, CodeIgniter, Yii, CakePHP, Laminas, Django, Rails, Express, Spring) and want to build on Exponential. Exponential describes itself as "a professional PHP application framework with advanced CMS (content management system) functionality" (README). These pages translate what you already know into Exponential's terms, say plainly where something works differently, and say just as plainly where Exponential has no direct equivalent.

Every class, setting and command named on these pages was checked against the kernel, the settings or the existing documentation. Where a page compares with another framework, it describes what that framework is known for, without judging it.

Contents

1. Exponential in one paragraph
2. Your first hour
3. Five things that are different
4. Map of the pages
5. Pages from other audiences you will use
6. Finding these pages again

Exponential in one paragraph

Exponential 6 is a free and open source PHP application framework and content management system under the GNU GPL, version 2 or later. A request enters `index.php`, is matched to a **siteaccess** (a named configuration per site, language or administration), translated from a URL alias to a `/<module>/<view>` address, checked against **roles and policies**, handled by a **module view**, rendered with the **template language** and placed into the pagelayout. Data editors manage lives in **content classes** with versions and translations; your own tables use `eZPersistentObject`. Configuration is **INI files in layers**; code lives in **extensions** installed with Composer. It is served by Apache or nginx with PHP-FPM, by **Exponential Velocity**, a persistent-worker server that is part of the product as of 6.0.15, or by FrankenPHP. The code base was developed from 1999, first under the name eZ Publish, which its creator discontinued after almost 18 years; 7x continued it as Exponential (Who is 7x?).

Your first hour

1. **Install** (10 minutes): `./console exp:install --apply` gives a site on SQLite (Installing in one command).
2. **Orient** (10 minutes): Exponential for web framework developers and your "coming from" page (Symfony, Laravel, Slim, CodeIgniter, Yii, CakePHP or Laminas, Django or Rails, Express or Spring).
3. **Translate** (10 minutes): the concept dictionary.
4. **Follow a request** (10 minutes): The request lifecycle.
5. **Build something** (20 minutes to start): Task: build a small CRUD module.

Five things that are different

1. **Content is the model.** Editorial data is a content class with versions, translations and permissions, not a table you design (The content model for ORM users).
2. **Siteaccess before routing.** A whole configuration is chosen per request (Siteaccesses).
3. **No service container.** Handlers are named in INI settings (Dependency injection).
4. **Templates do not escape automatically** and are overridden rather than extended (The view layer).
5. **Persistent workers.** Under Velocity, code must not keep per-request state in statics (Servers and deployment).

Map of the pages

Orientation

Page	What it helps you do
Exponential for web framework developers	See the whole picture in framework terms
Concept dictionary	Look up your framework's word and find the Exponential one
Reading path	Read the pages in a sensible order
When to choose Exponential, and when a plain framework	Decide fairly for a project

Coming from another framework

Page	What it helps you do
Coming from Symfony	Map bundles, kernel events, Doctrine, Forms and Security
Coming from Laravel	Map routes, Eloquent, Blade, middleware, queues and Artisan
Coming from Slim, CodeIgniter, Yii, CakePHP or Laminas	Map classic MVC and micro framework habits

Page	What it helps you do
Coming from Django or Rails	Map URLconf, ORMs, the admin, signals and commands
Coming from Express or Spring	Understand the PHP process model and map the rest

The request

Page	What it helps you do
The request lifecycle	Know where your code runs, from `index.php` to the output filters
Siteaccesses: matching a request to a site	Configure sites, languages and the administration per request
Routing: modules, views and parameters	Declare addresses and generate URLs
Pretty URLs: URL aliases and URL wildcards	Use slugs, redirects and rewrites without code
Controllers: module views	Write a view as a script or a class, and change kernel views
Responses, redirects, errors and JSON	Return pages, redirects, errors and JSON

Views and templates

Page	What it helps you do
The view layer: the template language beside Twig and Blade	Write templates and avoid the classic mistakes
Template resolution: designs and overrides	Know which file renders and override it
Getting data into templates: fetch functions and operators	Give templates data and helpers

Data

Page	What it helps you do
The content model for ORM users	Model editorial data as classes, objects, versions and nodes
ORM and active record: eZPersistentObject	Map your own tables to classes
Queries: content fetches, eZDB and SQL	Query content and tables portably
Migrations and seed data: .dba schema files and packages	Ship and upgrade schemas and seed data

Configuration and wiring

Page	What it helps you do
Configuration: INI files instead of YAML, PHP arrays or .env	Read, layer and change settings
Environments and secrets: EXP_ENV and settings/override	Run dev, test and prod, and keep secrets out of Git
Dependency injection and services: what exists	Wire testable code without a container
Extension points: handlers, INI registration and overrides	Find the right place to plug in
Packages: extensions, Composer and bundles	Structure and distribute your code

Request handling, events and security

Page	What it helps you do
Middleware: request rules, input and output listeners	Do what middleware does
Events and listeners: ezpEvent	Subscribe to kernel events and emit your own
Workflows and triggers: hooks on content operations	Run steps before and after publishing and checkout
Authentication and sessions	Sign users and machines in
Authorization: roles, policies and limitations	Declare and check permissions
Forms and CSRF protection	Handle forms safely
Validation: input, datatypes and escaping	Validate input and escape output

Performance and languages

Page	What it helps you do
Caching: the layers and how to invalidate them	Cache correctly and clear the right thing
HTTP caching and reverse proxies	Use headers, the HTTP cache and the response cache
Translations, locales and right-to-left	Translate the interface and the content

Background work, APIs and operations

Page	What it helps you do
Command line: the console and eZScript	Run and write commands
Queues and background jobs	Move slow work out of requests
Scheduled tasks: cronjobs	Run work on a schedule
REST and JSON APIs	Pick the right API layer
Headless and decoupled front ends	Serve content to another front end
Testing: PHPUnit, database tests and checks	Test without fear
Debugging, logging and errors	See inside a request and read the logs
Servers and deployment: Apache and PHP-FPM, Velocity, FrankenPHP	Serve and deploy

Exponential with other frameworks

Page	What it helps you do
Mixing Exponential and a framework	Combine both in one project
The Symfony legacy bridge for framework developers	Run Exponential 6 inside Symfony
The Exponential Platform products	Choose a Symfony-based distribution

Hands-on, checks and reference

Page	What it helps you do
Task: build a small CRUD module step by step	Build every piece once by hand

Page	What it helps you do
Ten pitfalls for framework developers	Avoid the mistakes everyone makes first
Checklists for framework developers	Evaluate, set up, review and go live
Web framework developer FAQ	Get quick answers
Glossary for web framework developers	Look up an Exponential term

Pages from other audiences you will use

Page	Why
Application creators: start here	The detailed guides behind most pages here: modules, datatypes, workflows, queues, REST, deployment
Extension development: the book	Every extension point, step by step, from user to maintainer
PHP developers: start here	The PHP side: language, conventions and the kernel's classes
Software engineers: start here	Architecture, quality and engineering practice
The template language	The language in depth, with Smarty comparisons
Kernel architecture	The authoritative description of the kernel's layers
Extension points	The catalogue of every extension point
Database support	The supported databases and how each is tested

Finding these pages again

In the help, filter by the audience **Web Framework Developers**; the pages live in the folder `web-framework-developers`. Search for a framework's name ("Symfony", "Laravel", "Django") to find the comparison that mentions it.

Related pages

- Getting started
- When something goes wrong
- Develop with Git
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

> **Become an Exponential Partner**
>
> Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.
>
> Partners: start here | How to apply

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.