Exponential for web framework developers

What Exponential is when you look at it with the eyes of a Symfony, Laravel, Django or Rails developer - a PHP application framework with a content model, permissions, templates and a server built in - which parts feel...

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

This page is for developers who know a web framework and want a first, honest picture of Exponential. Its README describes it as "a professional PHP application framework with advanced CMS (content management system) functionality", and both halves of that sentence matter: there is a framework underneath (front controller, routing to modules and views, templates, a database layer, settings, a console), and on top of it a content model with versions, translations, permissions and an administration that you get without writing code.

Contents

1. In one paragraph
2. What feels familiar
3. What is different
4. Where things live
5. How you add code
6. Where to go next

In one paragraph

Exponential 6 is a PHP code base with a long history: it was developed from 1999, first under the name eZ Publish, which its creator discontinued after almost 18 years; 7x continued it, and the product has been called Exponential since August 2025 (Who is 7x?). It is installed with Composer, runs on PHP 8, stores data in SQLite, PostgreSQL, MySQL/MariaDB, Oracle or MongoDB (Database support), and is served by Apache or nginx with PHP-FPM, by Exponential Velocity (a persistent-worker server written in PHP that is part of the product as of 6.0.15) or by FrankenPHP (Velocity is part of Exponential 6.0.15).

What feels familiar

You know	Exponential has
A front controller (`public/index.php`)	`index.php`, which builds `ezpKernel( new ezpKernelWeb() )` and returns its content
Routes to controller actions	modules and views: `/<module>/<view>/<params>`, declared in `module.php` (Routing)

You know	Exponential has
A template engine	the template language (`.tpl` files compiled to PHP by `eZTemplate`) (The view layer)
An ORM or active record	`eZPersistentObject` for your own tables, and the content API for content (ORM)
A console	`bin/php/console` with `exp:*`, `ext:*` and `cron:*` commands (Command line)
Packages	extensions, installed as Composer packages (Packages)
Configuration files	INI files in layers (Configuration)
Tests	PHPUnit, most suites without a database (Testing)

What is different

- **Content is a first-class model.** Most data an editor touches is not a table you design, but a *content class* (a type with attributes) whose *objects* have *versions* and *translations* and sit at *nodes* in a tree. You define classes in the administration and get editing, history, permissions and search for free (The content model for ORM users).
- **Siteaccesses instead of one application config.** A request is first matched to a *siteaccess* (a site, a language version, the administration), and the settings of that siteaccess are loaded before routing (Siteaccesses).
- **No general dependency injection container.** The legacy kernel finds classes through INI settings, naming conventions and a generated class map. When it runs inside Symfony through the legacy bridge it can receive Symfony's container, but the kernel itself does not define services in one (Dependency injection).
- **Templates are overridden, not extended.** A design chain and `override.ini` rules decide which template renders a view; there is no block inheritance as in Twig or Blade (Template resolution).
- **Permissions are data.** Roles made of policies (module, function, limitations) are stored in the database and assigned in the administration; your module declares the functions, administrators grant them (Authorization).
- **Schema changes are not migrations classes.** Tables are described in database-neutral `.dba` files; upgrades are scripts you ship (Migrations).

Where things live

From Architecture for application creators:

```
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
```

How you add code

You never edit `kernel/` or `lib/`. Everything you write goes into an **extension**, the unit that plays the role of a Symfony bundle, a Laravel package or a Django app: modules (your routes and screens), classes, templates, settings, datatypes, workflow event types, cronjob parts and console commands. **Setup > RAD** (`/setup/rad`) lists every extension point and writes the boilerplate for most of them (RAD tools, Extension points).

Where to go next

1. Concept dictionary: your framework's words and their Exponential counterparts.
2. The request lifecycle: what happens between `index.php` and the response.
3. Task: build a small CRUD module: the whole path once, by hand.
4. The coming-from pages: Symfony, Laravel, Slim, CodeIgniter, Yii, CakePHP or Laminas, Django and Rails, Express and Spring.

Related pages

- Web framework developers: start here
- Exponential as an application platform
- Kernel architecture
- Extension development: the book
- 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.