Exponential 6.0.x

Exponential 6.0.x specifically requires the php package manager called Composer to be installed and run the command composer install; run before Exponential can be used.

Source file: README.md Last changed 2026-11-07
https://doc.exponential.earth/exphelp/viewer/project/README.md

[Image: Exponential - Powered by SQLite Logo]

[Image: MongoDB-exp]

[Image: Exponential - Project Logo]

Important Notice for New or First Time Users

Exponential 6.0.x specifically requires the php package manager called Composer to be installed and run the command `composer install;` run before Exponential can be used.

To install Exponential, first download the software (this repository) and run composer install inside of the root Exponential 6.0.x installation directory. To do this, you must first:

1. Install Composer from getcomposer.org
2. Download Exponential 6.0.x and then from within the web server root directory:
3. Run `composer install` in the root directory of the Exponential installation

Failure to perform these required steps will result in the software not running properly or at all.

Please install responsibly and install (via composer) the software's php libraries required before attempting to use the software.

Exponential Project Notice : 2025.08.12

"Please Note: This project is not associated with the original eZ Publish software or its original developer, eZ Systems or Ibexa".

Exponential Project Status

Exponential has made it beyond it's end of life in 2021 and survived. Current releases are primarily aimed at easing the requirements to support current versions of the PHP language like PHP 8.2, 8.3, 8.4, 8.5, 8.6 and beyond php9

Who is 7x

7x is the North American corporation driving The Continued General Use, Support, Development, Hosting, Design of Exponential Enterprise Open Source Content Management System in 2025.

7x has been in busines supporting Exponential Website Customers and Projects for over 24

years. 7x took over leadership of the project and it's development, support, adoption and community growth in 2023.

7x represents a serious company leading the open source community based effort to improve Exponential and it's available community resources to help users continue to adopt and use the platform to deliver the very best in web applications websites and headless applications in the cloud.

Previously before 2022, 7x was called Brookins Consulting who was the outspoken leader in the active Exponential Community and it's Portals for the past 24 years.

What is Exponential?

Recent improvements to Exponential

Exponential (the application of interest) is delivered to users worldwide by a web server. It runs behind any of the usual ones, and since 6.0.15 it also ships one of its own — **Exponential Velocity** — which is considerably faster. See How Exponential is served below.

As of 6.0.15, the Velocity Internet / Web Application Server is part of Exponential. The root `composer.json` requires `se7enxweb/exponential-velocity` (`~0.0.4.57`); it is no longer a suggestion, so every installation has it, on every PHP Velocity runs on (8.0 and later, the stock PHP of Red Hat Enterprise Linux 9 included; it comes through the metapackage `se7enxweb/exponential-php-floor`, which leaves it out on PHP 7.4). This marks a turning point for the project: a cornerstone, a capstone, and a new level of application system integration, which provides even greater features out of the box in the core product, by default. Exponential is a whole working unit, not just a framework: a rapid application development (RAD) stack for Internet applications that is a CMS, a digital experience platform and an e-commerce system in one installation. See Velocity is part of Exponential 6.0.15.

Exponential with a full complement of all popular and available php extensions installed like SQLite3 users no longer require a dedicated database server anymore with Exponential 6.

With PHP we require composer to install Exponential software and no other software required to run the application. This is an incredible improvement to the kernel (core) of Exponential.

The 6.0.15 line also brings, among much else (the full list is in the 6.0.15 changelog):

- **Exponential Intelligence**, a writing assistant in the administration interface that writes drafts, never publishes on its own, and works with hosted providers in the US or the EU or with a model on your own server.
- **Exponential Layouts**, a visual page builder: layouts, zones and blocks that editors arrange, and rules that decide which pages get them.
- **A tamper-evident audit trail** with a console, charts, alerts, streamed and background exports, and channels of extensions (audit guide).
- **The content finder** (Setup > Content finder): find the content behind an id, a remote id, a URL alias, a page address or a file path, and act on it within your rights (feature).
- **Content jobs and a job queue**: large removes, copies, moves, package installs and audit exports run in the background (content jobs); the job queue `expqueue` runs the queued work of extensions (expqueue).
- **Packages end to end** on Setup > Packages: every upload validated, installs in the

background or as drafts, a JSON package format next to the classic one (packages guide, JSON format).

- **More than 360 locales**, over 300 of them new: content in Bengali, Urdu, Persian, Vietnamese, Swahili, Romanian, Tibetan, Yiddish, Latin and nearly every other written language of the world, with their dates, numbers and currencies, for every inhabited country and territory; the administration in more than 150 languages, all of them complete (the current totals are at the top of languages); right-to-left languages get `dir="rtl"` in content and in the administration (right to left).
- **MariaDB, MySQL, PostgreSQL, SQLite, Oracle and MongoDB** on one kernel, PHP 8.0 to 8.5 (PHP 7.4 usable without support), and the bundled application server Exponential Velocity, part of Exponential as of 6.0.15 (required at `~0.0.4.57`, which runs on PHP 8.0 as well, answers HEAD over HTTP/2 with its headers only and drains its connections when it stops).

What does Exponential provide for end users building websites?

Exponential is a professional PHP application framework with advanced CMS (content management system) functionality. As a CMS its most notable feature is its fully customizable and extendable content model. It is also suitable as a platform for general PHP development, allowing you to develop professional Internet applications, fast.

Standard CMS functionality, like news publishing, e-commerce and forums is built in and ready for you to use. Its stand-alone libraries can be used for cross-platform, secure, database independent PHP projects.

Exponential is database, platform and browser independent. Because it is browser based it can be used and updated from anywhere as long as you have access to the Internet.

(Referred to as `legacy` in Exponential Platform 5.x and Ibexa OSS)

How Exponential is served

Exponential is a PHP application, so something has to accept the request, find the right file, and hand PHP the work. You have a choice, and since 6.0.15 that choice includes a server built for this application specifically.

The traditional options

Exponential runs behind any web server that can rewrite a URL — that is, turn a readable address like `/recipes/summer-salad` into a request for `index.php`. Every server below can do it, and Exponential ships or documents the rules for each:

Server	How it rewrites	Notes
Apache HTTP Server	`mod_rewrite`, `.htaccess`	The most widely used; `.htaccess` ships with Exponential
Nginx	`try_files` / `rewrite`	Very common; rules go in the server block, not `.htaccess`

Server	How it rewrites	Notes
LiteSpeed and **OpenLiteSpeed**	`.htaccess`-compatible rewrite engine	Reads Apache rules directly in most cases
Caddy	`rewrite` / `handle` / `try_files`	Automatic HTTPS out of the box
Microsoft IIS	URL Rewrite Module (`web.config`)	Windows hosting
lighttpd	`mod_rewrite`	Small footprint
H2O	`redirect` / `file.custom-handler`	HTTP/2 focused
Traefik	ReplacePathRegex middleware	Usually in front of another server

And for development only:

| **PHP built-in server** | `php -S` with a router script | **Development only.** Single-threaded, no concurrency, not hardened. Never expose it to the internet. |

All of these work. Exponential does not require any particular one.

Exponential Velocity — the fast option

Velocity is a web server written in PHP, bundled with Exponential, that keeps the application loaded between requests.

Part of Exponential as of 6.0.15

As of 6.0.15 Velocity is part of Exponential itself. `composer.json` requires `se7enxweb/exponential-velocity ~0.0.4.57`, so `composer install` installs the server with the rest of the product, on PHP 8.0 and later, next to Apache with PHP-FPM, nginx and FrankenPHP. It is the recommended engine for every stage; FrankenPHP is the production-ready alternative; the traditional servers keep working exactly as before.

This is a turning point, a cornerstone and a capstone for the project: the path from a visitor's request to a rendered page now belongs to one product, with one installer, one console and one set of documentation. With the server in the core, Exponential is one working stack (server, RAD application platform, CMS, digital experience and e-commerce) rather than a framework that needs other software assembled around it, so it can be adopted anywhere without the usual integration work. The whole story: Velocity is part of Exponential 6.0.15.

It is based on the Qbix server and is developed alongside Exponential, so the two are tuned against each other rather than merely made compatible.

Why it is faster

A traditional setup starts PHP fresh for every request: load the framework, read the settings, connect to the database, render, throw it all away, repeat. That work is the majority of a page's cost, and it is identical every time.

Velocity loads the framework **once** into a parent process and forks workers that share it. A request arrives and the application is already in memory. On top of that it adds a response cache that answers a repeat request without waking the application at all.

What that measures as

These are the figures currently documented for this stack:

> **0.36 ms per cached page, 8,941 requests per second sustained, and repeat visits never touch the network.**

Read honestly, they mean:

- **0.36 ms** is the server's own work to answer a cached page — measured as a median over hundreds of requests, not a best case.
- **8,941 req/s sustained** was measured on a 12-core machine serving a real page, with **zero errors** at every concurrency tried, up to 384 connections in flight.
- **Repeat visits never touch the network** refers to the optional navigation cache (a service worker): once a visitor has a page, a return visit is answered by their own browser with no round trip at all.

Every one of those is reproducible with scripts that ship with the project — they are not marketing numbers, and the methodology is written down alongside them.

The part that matters even if you never run Velocity

Most of the speed came from fixing Exponential and the cache, not from the server process model. Those fixes apply wherever Exponential runs:

- a returning visitor's reload is answered with **202 bytes instead of the whole page**, because cached pages now answer conditional requests properly
- a page's identity is derived from **its content**, so rebuilding a cache entry that produced identical output no longer forces every visitor to download it again
- when a cached page expires, **one** request rebuilds it while everyone else is served the existing copy — instead of every visitor present rebuilding it at once
- a "not found" is remembered instead of being rebuilt from scratch every time
- pages are no longer shipped padded with template indentation

Exponential 6.0.15+ is fast under load on Apache or Nginx too. Velocity raises the ceiling further; it is not what makes the application quick.

Using it

Shell

```
./console exp:velocity deploy          --allow-root-user   # after a change: autoloads, settings,
 caches, PHP-FPM, Velocity
./console exp:velocity cache clear --apply --allow-root-user   # empty Velocity's response cache
./bin/php/console exp:velocity start     --allow-root-user
./bin/php/console exp:velocity status    --allow-root-user
./bin/php/console exp:velocity restart   --allow-root-user
./bin/php/console exp:velocity graceful  --allow-root-user   # no dropped connections
./bin/php/console exp:velocity stop      --allow-root-user
```

`graceful` re-executes the server without releasing the listening socket, so a deploy does not drop requests. `status --json` is there for monitoring.

Engines: FrankenPHP, PHP's built-in server or Qbix

The same commands drive three servers, chosen with `[ServerSettings] Engine` in `velocity.ini`:

- **Qbix**: Velocity's own server, installed with the `se7enxweb/exponential-velocity` package. The recommended engine for every stage, from development to production, and the fastest. It serves HTTP and HTTPS itself, with its own certificates.
- **FrankenPHP**: Caddy with PHP built in, downloaded as one checksum-verified binary. The production-ready alternative.
- **PHP's built-in server**: nothing to install. For development, and the shipped default because it always works.

Shell

```
./bin/php/console exp:velocity config set --apply ServerSettings Engine qbix --allow-root-user
# every real site
./bin/php/console exp:velocity start   --allow-root-user
./bin/php/console exp:velocity config set --apply ServerSettings Engine frankenphp --allow-root-u
ser   # the alternative
./bin/php/console exp:velocity install --allow-root-user   # fetch and verify the FrankenPHP bina
ry
./bin/php/console exp:velocity start --all --allow-root-user   # all three side by side, for test
s
```

What each engine supports, the ports, the settings they share and who may open the views each server answers itself are in doc/eng-us/bc/6.0/velocity-engines.md.

Settings live in `settings/override/velocity.ini.append.php`. Every one of them has a default that works, so a server that configures nothing still runs.

One thing to know before you rely on it

Workers keep the application in memory between requests. After changing a template, a setting or a class, restart the server — clearing caches alone will not dislodge what a worker already holds. The same applies if the engine is being loaded from its archive: rebuild it first.

When a traditional server is still the right answer

- **You need per-site operating-system isolation.** If several customers must not be able to read each other's data, run one Velocity instance per site behind a proxy, each as its own operating-system user. A single server process serving several sites is a routing convenience, not a security boundary — the process that serves one site can read every file the others hold.
- **Your hosting will not let you run a long-lived process.** Shared hosting usually will not. Apache or LiteSpeed with `.htaccess` is the answer there.
- **You already have a tuned Nginx or Apache in front.** Keep it. Velocity is happy behind a reverse proxy. It does not need one for TLS: Velocity serves HTTPS itself, with a certificate you give it, a self-signed one it makes, or one it obtains and renews from Let's Encrypt, and swaps a renewed certificate in without a restart.

In short

	Traditional server	Exponential Velocity
Application load	every request	once, shared by all workers
Cached page	rendered or proxied	**0.36 ms**, no worker involved
Sustained throughput	depends on pool size	**8,941 req/s** measured
Repeat visit	a round trip	**no network at all** (optional)
Isolation between sites	per vhost/user, as configured	one instance per site
Shared hosting	yes	usually not
Setup	vhost plus rewrite rules	one console command

Requirements

- PHP
- (Optional) Web server. Used to deliver the website to the end user. Apache, Nginx, LiteSpeed, Caddy, IIS, lighttpd and others all work — or use the bundled **Exponential Velocity**, which needs nothing installed beyond PHP.
- (Optional) Database server. Used to store website content (and application information). SQLite needs none. **Supported databases (Exponential 6.0.x), in order of preference:** 1. SQLite 3 (the default) · 2. PostgreSQL · 3. MongoDB · 4. MariaDB · 5. Oracle · 6. MySQL. Microsoft SQL Server is known to be possible with additional work at this time. Details: Database support.
- Operating system: almost any that runs PHP 8.0 or later (PHP 7.4 usable without support); see Operating system support.
- Composer. Used to download Exponential software packages for installation, also notably installs the required Zeta Components php libraries and the extensions listed in `composer.json`.
- (Optional) For Exponential Intelligence: an API key of a hosted AI provider, or a local open-weight model on the same server.
- Computer to run the PHP website application.

What version of PHP is required

Supported: PHP 8.0 to 8.5. Usable without support: PHP 7.4 (Exponential 6.0.15 and later).

Please use the latest version of PHP available on your OS.

PHP 8.0 is supported so that the stock PHP of Red Hat Enterprise Linux 9 (and AlmaLinux / Rocky Linux 9) runs Exponential under Apache or PHP-FPM and under Exponential Velocity, the built-in application server, which runs on PHP 8.0 to 8.5 from its release 0.0.4.54 (until then it needed 8.1). See PHP 8.0 support and the engine's PHP 8.0 and Red Hat Enterprise Linux page.

PHP 7.4 is usable again from 6.0.15, without support, as a bridge for organizations that cannot change the PHP of their servers yet. The kernel and the bundled extensions run on it; Velocity, Exponential Layouts and a few extensions that need PHP 8 are left out. Composer chooses the

right packages for each PHP version through the metapackage se7enxweb/exponential-php-floor. Why the project does this, what works on PHP 7.4, how security is kept and the end-of-life stance: Why Exponential supports PHP 7.4. PHP 7.4 itself has had no security fixes from the PHP project since November 2022, so run it only where your operating system vendor or host still patches it, and plan the move to PHP 8.

Main Exponential features

- User defined content classes and objects
- Version control
- Advanced multi-lingual support
- Built in search engine
- Separation of content and presentation layer
- Fine grained role based permissions system
- Content approval and scheduled publication
- Multi-site support
- Multimedia support with automatic image conversion and scaling
- RSS feeds
- ATOM feeds
- iTunes RSS feeds
- OPML feeds
- Contact forms
- Advanced e-Commerce Stores (Built in)
- Flexible workflow management system
- Full support for Unicode
- Extendable Template engine
- A headless CRUD REST API
- Database abstraction layer supporting, in order of preference, SQLite 3 (the default), PostgreSQL, MongoDB, MariaDB, Oracle and MySQL
- MVC architecture
- Support for the latest Image and Video File Formats (webp, webm, png, jpeg, etc)
- Support for highly available and scalable configurations (multi-server clusters)
- XML handling and parsing library
- SOAP communication library
- Localization and internationalization libraries
- Several other reusable libraries
- SDK (software development kit) and full documentation
- A writing assistant for editors, Exponential Intelligence, that writes drafts only (see below)

- A visual page builder, Exponential Layouts, with rules that map layouts to content (see below)
- A tamper-evident audit trail with a console, alerts, exports and extension channels
- A content finder that finds content by id, remote id, URL alias, page address or file path
- Background content jobs and a job queue for large operations and queued work
- Content packages in the classic or a JSON format, validated on upload, installed in the background or as drafts
- An Internet / web application server of its own, Exponential Velocity, part of Exponential as of 6.0.15 and installed with every copy, next to Apache, Nginx, FrankenPHP and the others
- plugin API with thousands of open-source extensions available, including:
  - content rating and commenting
  - landing page management
  - advanced search engine
  - wysiwyg rich-text editor
  - in-site content editing
  - content geolocation

Exponential Intelligence (new!)

Exponential Intelligence is a writing assistant built into the administration interface. Editors ask in plain language for new content or for changes ("write three short articles about our spring events under this folder", "rewrite the intro of this article, shorter"). The assistant looks up what it needs (class definitions, the current content, the content tree around it) and writes **drafts**, as the signed-in editor and with that editor's own permissions.

It is the extension `se7enxweb/expai` (release 0.0.1.0), required in `composer.json` since 6.0.15, so every installation has it in `extension/expai`. **A new installation activates it by default**: the installer switches it on for the administration siteaccesses and creates its tables. It sends nothing anywhere until you add a provider key (see How to enable it).

What it does for editors

- **The Intelligence tab** on a node's full view: a conversation per user and node that shows each lookup and each draft as a step, and lists the drafts it wrote with links to edit, view and discard them.
- **In the edit form**: "Fill the fields" from a brief, actions per field (Write, Rewrite, Shorten, Translate, Suggest), one-click presets (title, summary, meta description, keywords), an **Improve** menu on every text field, and alt text for images from a vision model. Values go into the form for review; the online editors keep working.
- **Vector**, a workspace of threads: scope, plan and write modes, a plan with a cost estimate for bulk work, staged review of every change per field before anything is written, parallel runs of an approved plan, checkpoints to go back to, live progress streamed to the browser, and **What changed** with revert for drafts written directly.
- **Chains and translation sync** through the job queue `expqueue`: write, review, revise and

translate in steps with a human checkpoint; translations brought up to date after the source language changed.

- **A dashboard** with cost, tokens, requests, drafts and limits, and the label "Exponential Intelligence" on every version it wrote (My drafts, version views, edit forms, approval items).

Providers: hosted, in the EU, or on your own server

- **Anthropic** (Claude), proven live.
- **Any OpenAI-compatible service** through one adapter: Melious.ai (EU region, open-weight models, prices in EUR converted with the ECB reference rates), Hugging Face, OpenRouter, Mistral and others.
- **Local open-weight models** through Ollama on the loopback address, started, stopped and pulled from the administration through a root-owned helper; the data never leaves the server.

Only models that passed a live tool-calling **probe** (or are trusted by configuration) may write drafts.

Safety

- **Drafts only.** It never publishes on its own. "Send for approval" runs the normal publish operation as the editor, so an approval workflow holds the version for the approver.
- **Permissions.** Every lookup and every draft runs with the editor's own content permissions; the assistant's features are separate policies (`expai/write`, `expai/vector`, ...), held only by the Administrator role in a new installation.
- **Audit.** Every request, draft and state change is an event of the audit channel `ai` (actions and metadata, never a prompt, an answer or a key), shown with its own panel and links in the audit console.
- **Data policy.** Each provider states its region (`us`, `eu`, `local`) and retention; rules per section and content class decide which content may go where, and every request records where it went. **EI: Summary** is one printable page for a data protection officer.
- **Costs you can cap.** A usage log with the estimated cost of every request, limits per user, day, month, site and provider, and budgets for queued work.
- **One switch.** `[AISettings] Enabled` turns it off everywhere at once, or per siteaccess, and a veto hook can refuse a request with a reason.

How to enable it

On a new installation it is already active for the administration siteaccesses (admin, editor and adminui), its tables exist and the Administrator role holds all its policies. Two things are left to you:

1. Put the provider key into `settings/override/expai.ini.append.php` (`[AISettings] Enabled=true`, `ApiKey=...`), never into a committed file. Without a key it stays idle and sends nothing.
2. Add the cron parts `expqueue` and `expai` (`php runcronjobs.php -q -s <admin siteaccess> expqueue`, every minute; `... expai`, every five minutes), and give editors the

policy `expai/write` (and `ezjscore/call`) if they should use it.

Then open a folder in the administration, choose the **Intelligence** tab and ask for something.

Upgrading an earlier installation (one installed before Exponential Intelligence was part of the installers):

1. Activate it for the admin siteaccess(es): `ActiveAccessExtensions[]=expai` in each admin siteaccess's `site.ini.append.php` (or `ActiveExtensions[]=expai` for all).
2. Put the provider key into `settings/override/expai.ini.append.php` (`[AISettings]` `Enabled=true`, `ApiKey=...`), never into a committed file.
3. Create its tables: `php extension/expai/bin/php/expaischema.php --install -s <admin siteaccess>`.
4. Give editors the policy `expai/write` (and `ezjscore/call`), regenerate the autoloads, clear the caches, and add the cron parts `expqueue` and `expai`.
5. Open a folder in the administration, choose the **Intelligence** tab and ask for something.

Requirements: PHP 8.0 or later with `curl`, `dom`, `json` and `mbstring`; ezjscore and ezformtoken; expqueue for queued work. Read the installation guide, the usage guide and the full documentation (providers, data policy, Vector, security, configuration). A summary is in doc/eng-us/features/6.0/extensions/expai.md.

Exponential Layouts (Revolutionary)

Exponential Layouts is a visual page builder for Exponential. It brings the concept of the upstream layouts packages to the Exponential kernel: a **layout** is a saved arrangement of **blocks** in **zones**, and a **rule** decides which pages get it, so editors rearrange pages without touching templates.

What it is

- **Layouts, zones and blocks**, stored as drafts and published versions, so unfinished work is never visible to visitors. Layout types (one to four columns, hero, sidebars, featured, mosaic and more) are defined in `explayouts.ini`.
- **Block definitions and handlers**: text, title, HTML, Markdown, image, button, card, list, grid, gallery, carousel, tabs, accordion, map, video and more, with the standard blocks of `explayouts_standard`; a new block type is a small handler class and a template in your own extension.
- **Collections and queries**: a block gets its content by hand or from a query (children, subtree, siblings, latest, relation lists, tags and pluggable query types), with hand-picked items pinned among dynamic ones.
- **Rules** that map layouts to content: targets (a node, a subtree, a path or path prefix) and conditions (siteaccess, content class, query parameter, time), evaluated by priority.
- **Shared layouts with linked zones**: build a header, footer or prefooter once in a shared layout and link the zone of every page layout to it; change it once and every page follows.
- **The editor** at `/explayouts_ui_api/app`, a single-page application over a JSON API, and the **Layouts** tab of the administration: dashboard, layout list, shared layouts, layout mappings (rules), import and transfer, the template editor and setup. A "Layouts" tab on a node's view

shows the layouts and rules that apply to it.

Why it matters

- Editors build and change pages without template work.
- Each part of a site gets its own layout by rule, per subtree, content class, siteaccess or campaign period.
- Header, footer and other repeated parts are defined once and reused through shared layouts.
- Layouts move between installations by export and import.

How to enable it

The suite is required in `composer.json`, so every installation has it in `extension/`. The 7x multi-site installer packages set it up for the sites they install that use it. Otherwise follow the installation checklist: activate the extensions, create the `explayouts_*` tables from the extension's `sql/<engine>/schema.sql`, regenerate the autoloads, clear the caches, then create a layout and a rule. The README and INSTALL of each extension give its own settings and steps.

The guide: doc/eng-us/bc/6.0/LAYOUTS.md. The repositories: explayouts (persistence, resolver, rendering, block handlers), explayouts_core (services), explayouts_standard (standard blocks), explayouts_ui (the Layouts tab), explayouts_ui_api (the editor and its JSON API), explayouts_api, explayouts_site_api, explayouts_content_browser, explayouts_content_browser_core, explayouts_content_browser_ui, explayouts_relation_list_query and explayouts_tags_query. Their Composer packages are `se7enxweb/explayouts`, `se7enxweb/explayouts-core`, `se7enxweb/explayouts-ui` and so on.

Exponential Distribution Integration: LoveStack (2026.10)

LoveStack, Mugo Web's fork of the legacy kernel (`mugoweb/ezpublish-legacy`), carries valuable improvements that Mugo Web built over many years. At the time of this port (12 October 2026), the LoveStack repository had not been updated in over seven months (last commit 23 February 2026, last release v2025.03.1 on 1 March 2025, supporting PHP up to 8.3), which has left many users wondering about the future of the distribution and its unique features. Those features have been carefully documented and rebuilt in the Exponential 6.0.15 distribution (this GitHub main repository), so everyone can use the modern Exponential features and the LoveStack improvements together, as one stack, on PHP 8.1 to 8.5. Bringing them into Exponential keeps them available and maintained for everyone, with full credit to Mugo Web.

What you gain

- **Safer by default**: logging out really ends the session and its cookie; uploaded images and package thumbnails are checked by their content; node data calls (`ezjscnode`) and registration can require a policy; one account per login, with a report of duplicates before the unique index is added.

- **For editors**: a Copy button on the admin full view, required alternative text for image fields, trimmed text line fields, a Locations window that keeps hidden locations hidden, hidden custom tags and custom tag controls in the rich text editor.
- **For administrators**: audit records for location changes, site data in **Setup > Site data**, extra headers per error page, PHP deprecations in their own log, log file size as a setting, a warning when a change clears every view cache, two optional URL settings.
- **For developers**: `eZURI::build()`, a mail transport per message, the `rawurlencode` and `rawurldecode` template operators, `[Include …]` inside settings files, names made by your own PHP class, database errors with the driver's error number, and the PHP 8 fixes LoveStack made, reproduced and fixed where still needed.

36 features in all (L01 to L36), each with its tests.

How to switch the opt-in ones on

Nothing changes on an existing site unless you switch it on, except plain bug fixes, the new log settings and warnings, the Copy button and the security checks for logging out and image uploads. New installations also start with the policies for node data and registration, the unique login index and the custom tag controls. The rest is one setting each, in an override file, for example:

INI

```
# settings/override/ezjscore.ini.append.php: node data calls need ezjscore/call with ezjscnode
[ezjscServer_ezjscnode]
RequirePolicy=enabled

# settings/override/site.ini.append.php: registration needs the User / Register policy
[RoleSettings]
PolicyOmitListExclude[]=user/register

# settings/override/ezoe.ini.append.php: custom tag controls in the rich text editor
[EditorSettings]
CustomTagControls=enabled
```

then `php bin/php/ezcache.php --clear-tag=ini`. Alternative text and trimming are options per class attribute on the class edit page. Every feature's setting, default and test is in the overview.

Coming from LoveStack

Shell

```
./console exp:lovestack check           # read only: what the installation brings along
./console exp:lovestack migrate         # dry run; add --apply to record the database as Exponen
tial's
./console exp:user-login-check          # duplicate logins; --fix-index adds the unique index
```

Step by step: Moving a site from LoveStack and the upgrade guide's Coming from LoveStack.

Documentation and credits

The overview, feature by feature: doc/eng-us/bc/6.0/lovestack.md, with the detailed pages in doc/eng-us/bc/6.0/love/. LoveStack itself: https://github.com/mugoweb/ezpublish-legacy.

Thank you to **Mugo Web**, who built these improvements and published them as open source under the GNU GPL. Every rebuilt feature keeps their copyright notice and names the LoveStack commit it comes from. "LoveStack" is Mugo Web's name for their distribution; Exponential uses it only to say where these features come from, and nothing here implies an endorsement by Mugo Web. They are friends of this project, and we are grateful for their work.

Exponential Documentation

doc.exponential.earth is the documentation portal for Exponential: the user manual for editors, the technical manual for developers and administrators, the extensions catalog and answers to the questions people ask most. It is itself an Exponential site, and it is free to read.

How it saves you time and money

- **Answers before you build.** Most of what a project needs (multiple languages, access control, workflows, scheduled jobs, a shop, clustering) is already in Exponential. The manuals show how to configure what is there, so you do not pay to develop, test and maintain your own version of it.
- **Shorter onboarding.** Editors learn their daily tasks from step-by-step pages with screenshots of the real administration interface, instead of from your developers' time.
- **Fewer support hours.** Installation requirements, virtual host setup and troubleshooting are written down, so a broken setup is usually a page away rather than a support ticket.
- **Better estimates.** Concepts such as the content tree, classes, sections and the design and template system explain how a site is put together, which makes it easier to plan a project and to judge what an extension or a change will cost.
- **The right extension, not a new one.** The extensions pages list what exists, who maintains it and how it is supported, before you commission custom code.

Where to start

You want to...	Start here
See everything in one place	Exponential documentation home
Get quick answers about licensing, support and upgrades	Frequently asked questions
Learn to edit and publish content	User manual, and its daily tasks such as adding content, images and files and translating content
Build pages visually	Building pages with layouts
Manage people and what they may do	Managing users and access control
Install Exponential	Installation: requirements, virtual host setup and troubleshooting
Understand how Exponential works	Concepts and basics, including content management and multiple languages
Configure one or several sites	Configuration and site management
Design and theme a site	Templates, starting with how templates work and the pagelayout

You want to...	Start here
Use built-in features	Features, for example cronjobs and clustering
Look up settings, operators and functions	Reference
Extend Exponential	Exponential extensions and the list of extensions and their support options
Read the built-in help center online	Help center

The documentation is published under the GNU Free Documentation License, version 1.3 or any later version. The same help center also ships with every installation under doc/ and in the administration interface.

Installation

Read doc/eng-us/INSTALL.md or go to exponential.doc.exponential.earth/Exponential/Technical-manual/6.x/Installation.html

The shortest path needs only PHP and Composer (SQLite, no database server):

Shell

```
composer create-project se7enxweb/exponential exponential
cd exponential
php bin/php/console exp:install --apply --random-password --url=http://localhost:8087
php bin/php/console exp:velocity start --engine=php
```

Building websites with Exponential? The recommended way is to get the kernel and every se7enxweb extension as Git clones with their full history, so you can follow development, keep your changes on a branch and send fixes back as pull requests:

Shell

```
composer config --global 'preferred-install.se7enxweb/*' source
git clone https://github.com/se7enxweb/exponential.git
cd exponential
git switch -c my-site
composer install
```

Develop with Git explains each step, and how to turn an existing site into clones.

Optional extensions that `composer.json` brings but leaves inactive, such as Exponential Intelligence and the job queue, are switched on afterwards; doc/eng-us/INSTALL.md says how.

Issue tracker

Exponential (6.x) Issue tracker

Submitting bugs, improvements and stories is possible on Exponential Project Issue Tracker If you discover a security issue, please responsibly report such issues via email to security@se7enx.com

Exponential Community Project Issue tracker

Submit bugs, stories and issues in general about almost anything Exponential Project / Community / Ecosystem / Website Software related to the https://github.com/se7enxweb/exponential-community/issues

Where to get more help

The help center's book Exponential Community lists every channel and every public website of the project.

Exponential documentation: exponential.doc.exponential.earth/Exponential

Exponential Community forums: share.exponential.earth/forums

Exponential Project Website: exponential.earth

Support Exponential! Project extension support Website: support.exponential.earth

Share Exponential! Telegram Community Support Chat https://t.me/exponentialcms

How to contribute new features and bugfixes into Exponential

Everyone is encouraged to contribute to the development of new features and bugfixes for Exponential 6.

Code contributions to the core kernel (this repository), whether by pull request, patch, issue or mail, are merged only once the author has signed the Exponential Foundation and 7x Contributor Licensing Agreement (version 1.1.0). **By signing and submitting, you assign the copyright in your contribution to the Exponential Foundation and 7x, which become its owners, and give up control over it:** The Exponential Foundation and 7x may use, change, relicense and distribute it in any way, including commercially and under other licenses, without asking or paying you, and you cannot withdraw it or set conditions. See Copyright assignment (CLA). Your own custom extensions are not affected.

Donate and make a support subscription.

Help fund Exponential!

You can support this project and it's community by making a donation of what ever size donation you feel willing to give to the project.

If we have helped you and you would like to support the project with a subscription of financial support you may. This is what helps us deliver more new features and improvements to the software. Support Exponential with a subscription today!

A wide range of donation options available at sponsor.se7enx.com, paypal.com/paypalme/7xweb and github.com/sponsors/se7enxweb

What was eZ Platform?

eZ Platform was eZ Publish's technological successor before being abandoned by Ibexa. eZ Platform was a highly extensible, pure Content Management Platform built upon the ideals core to Symfony. It provides the same flexible content model at it's core like Exponential, and has a growing amount of additional features outside the traditional CMS scope provided by means of "Bundles" extending it.

It is built on top of the full Symfony Framework, giving developers access to "standard" tools for rapid web & application development.

eZ Platform in some users view suffered a slow road to a stable datatype compatibility with existing custom implementations of Exponential. Today all of these concerns are now gone with a solid choice left leaving both Exponential Platform and eZ Platform as serious contenders to be carefully considered. Netgen's Media Website Core software represents a much more modern eZ Platform core powered by Ibexa OSS. If your going to choose; Choose wisely.

Further reading on the archived: https://ezplatform.com/

Note: Ibexa has discontinued support for the eZ Platform Product Line favoring instead to support only the Ibexa DXP / OSS Product line going forward.

eZ Platform is survived in part by Exponential Platform Project by 7x who has forked, mirrored, updated it's last free release eZ Platform stack of software and rebranded it as Exponential Platform 3.2.9 and Exponential Platform Legacy 2.5.0.1 (with legacy support) which provided much needed updates to provide for required php 8.3+ support allowing this software to continue to serve the users who fell in love with it from the start. Exponential Platform is just one of the many Symfony based PHP CMS + Ecommerce Solutions available under free / open source licenses.

Since the release of Exponential Platform; 7x has released the next generation Exponential Platform Nexus which represents the future of modern Symfony based CMS Development.

What is Exponential Platform?

In short Exponential Platform is the direct continuation of the software stack known as eZ Platform created by eZ Systems and discontinued by Ibexa. This software represents the latest product line before the move to rebrand the stack to Ibexa DXP/OSS upgraded to support PHP 8.3+ and rebranded to avoid any trademark issues with the continued use of the software by users with a need or desire to build smarter, faster and with greater ease of maintinence within a Symfony based PHP CMS.

Exponential Platform currently comes in nine different distributions that are very powerful and very stable.

- **Exponential Platform Legacy 5.0.x** Dual Kernel Symfony CMS which also includes a full copy of Exponential 6
- Exponential Platform Legacy 4.6.x Dual Kernel Symfony CMS which also includes a full copy of Exponential 6
- Exponential Platform Legacy 3.3.x Dual Kernel Symfony CMS which also includes a full copy of Exponential 6
- Exponential Platform Legacy 2.5.x Dual Kernel Symfony CMS which also includes a full copy of Exponential 6

- Exponential Platform 3.2.9 Symfony only CMS
- **Exponential Platform Nexus 1.3.0.x** is Platform v5.x Dual Kernel Symfony 7.4+ CMS which also includes a full copy of Exponential 6.x+
- Exponential Platform Nexus 1.2.0.x is Platform v4.x Dual Kernel Symfony 6+ CMS which also includes a full copy of Exponential 6.x+
- Exponential Platform Nexus 1.1.0.x is Platform v3.x Dual Kernel Symfony 5.4+ 3.4+ CMS which also includes a full copy of Exponential 6.x+
- Exponential Platform Nexus 1.0.0.x is Platform v2.5.x Dual Kernel Symfony 3.4+ CMS which also includes a full copy of Exponential 6.x+

Learn more at https://platform.exponential.earth

Start a discussion at Share @ https://share.exponential.earth/forums/exponential-platform

What is Ibexa DXP OSS?

Ibexa DXP OSS, the rebranded and evolved successor of eZ Platform, is a modern and highly extensible Digital Experience Platform built on the robust Symfony framework. Staying true to the foundational principles of Exponential, it retains the flexible content model at its core while expanding its capabilities to include a broad spectrum of features that go beyond traditional CMS functionality. These features are delivered through "Bundles," providing developers with powerful tools for creating advanced web and digital solutions.

Built on top of Symfony, Ibexa DXP OSS offers developers access to standardized tools for efficient and scalable web and application development. Its architecture supports rapid innovation while ensuring compatibility with modern development standards.

Ibexa DXP OSS has matured into a robust and versatile platform, serving as a serious option for organizations seeking a cutting-edge CMS solution. For those looking to build modern web experiences, tools like Netgen's Media Website Core further enhance Ibexa OSS, showcasing its potential as a flexible and future-ready platform.

If you're making a choice, choose with care—and consider the power and flexibility of Ibexa DXP OSS.

Ibexas DXP OSS is Dual Licensed with the non-free ibexa licenses or at your option the GNU GPLv2.

Further reading: ibexa.co

Documentation for Ibexa DXP: doc.ibexa.co/en/latest/

Ibexa DXP Developer Portal: developers.ibexa.co

What is Exponential Platform DXP?

Exponential Platform DXP is a free, open-source Digital Experience Platform built on the Symfony framework, the community continuation of the eZ Platform and Ibexa OSS line maintained by 7x.

Exponential Platform DXP gives organizations one place to create, manage and deliver content across websites, apps and channels. Editors work with structured content types they define themselves, in many languages and across several sites from one installation, with drafts,

version history and fine-grained permissions for every role, while developers build on standard Symfony tools, Twig templates and REST and GraphQL APIs to run a classic website, a headless content hub, or both. Because it keeps the proven content model of its predecessors and runs on current PHP, it suits teams who want an enterprise-grade, vendor-independent CMS without license fees, whether they are starting fresh or moving an existing eZ Publish or Exponential site forward.

Who it is for

- **Organizations with a lot of structured content**: publishers, associations, universities, public bodies and companies that run several sites, brands or languages from one content repository.
- **Teams leaving eZ Platform or Ibexa OSS** who want to stay on the same content model, the same APIs and the same way of working, on a maintained stack with current PHP and Symfony.
- **Exponential and eZ Publish users** who want a modern Symfony stack: the DXP itself is Symfony only, and the dual-kernel distributions Exponential Platform Legacy and Exponential Platform Nexus (see the list above) include a full copy of Exponential 6 next to it.
- **Developers and agencies** who prefer standard Symfony bundles, Composer and Twig over a proprietary framework, and an open license over a vendor contract.

What it does for editors

- **Structured content you define**: content types with typed fields (text, rich text, images, files, relations, dates, prices, locations and more), so the same content can appear on a page, in a feed, in an app or in a newsletter without copying it.
- **Many sites and languages from one installation**: translations side by side, a site per domain, path or language, and content shared between sites where it belongs.
- **A safe editorial process**: drafts, a full version history you can go back to, and permissions down to the section, subtree, content type and language.
- **An administration interface and content forms**: editors create and edit content in the browser, and content forms let signed-in users create and edit content on the site itself, within their permissions.
- **Media and assets**: images with automatic variations for every screen, files and embedded media, managed next to the content that uses them.

What it does for developers

- **Standard Symfony**: bundles, dependency injection, events, the console, Twig and Composer, so any Symfony developer can work on it from day one.
- **Headless or full site, or both**: REST and GraphQL APIs for apps and front-end frameworks, and server-side Twig rendering for classic websites, on the same content repository.
- **Extensible at every level**: field types, content views, query types, search and the administration interface can be extended with your own bundles instead of patching the

core.

- **Search and performance**: the built-in search or Solr behind one API, and HTTP caching designed for reverse proxies and CDNs.
- **Batteries included**: rich text, matrix and query field types, notifications, background jobs through Symfony Messenger and cron, JWT authentication for APIs, and a GraphiQL console to explore the GraphQL schema.

What Exponential Platform DXP adds

Exponential Platform DXP features improvements not found in Ibexa DXP OSS, for example:

- **The SQLite database driver and framework support**, which lets you host a site, including a read-only site, without a dedicated database server: one file holds the whole repository, which makes small sites, demos, staging copies and archives simple to run and to move.
- **Current PHP**: the stack is kept up to date for PHP 8.3 and later, with the fixes that this requires across the bundles it ships.
- **A path from Exponential 6**: the DXP is the Symfony-only distribution; when an existing Exponential or eZ Publish site has to move forward step by step instead of in one rewrite, the dual-kernel distributions Exponential Platform Legacy and Exponential Platform Nexus run Exponential 6 next to it.
- **A name of its own**: the stack is rebranded to avoid trademark issues, so you can keep using, changing and redistributing it freely.

An open community

The Exponential Project and the Exponential Foundation support the open source community directly, as part of our daily routine and work ethic, and put the support needs of community users first. Ibexa, by contrast, centers Ibexa DXP on a closed, commercial business relationship with its customers rather than on an open source community.

Exponential Platform DXP is free software under the GNU GPLv2 (or any later version), with no license fees and no edition that holds features back.

Getting started

- Learn more and download a distribution at platform.exponential.earth.
- Ask questions and share your experience in the forums at share.exponential.earth/forums/exponential-platform.

License

Exponential is GNU GPLv2 licensed. You can no longer choose between the GNU GPLv2-or-later and the eZ Systems' eZ Publish Professional Licenses. The GNU GPLv2 gives you the right to use, modify and redistribute Exponential under certain conditions. The GNU GPLv2 license is distributed with the software, see the file LICENSE. It is also available at https://www.gnu.org/licenses/old-licenses/gpl-2.0.txt

Using Exponential under the terms of the GNU GPL is free of charge.

The Professional License is no longer available.

This effectively makes Exponential GNU GPLv2-or-later ONLY.

Copyright

Exponential Copyright © 1998-2026 Exponential Foundation and 7x

In source files: `Copyright (C) 1998 - 2026 Exponential Foundation and 7x. All rights reserved.` Originally developed by eZ Systems AS as eZ Publish: Copyright (C) 1999 - 2014 eZ Systems AS. All rights reserved.

On 29 October 2026 the copyright of Exponential moved from 7x alone to the Exponential Foundation and 7x together, and the project replaced every notice it had with this one. The Foundation is named first on purpose: the nonprofit steward of the project, so that the software's rights, its license and its history do not depend on any one company over the twenty-year support window of a release line. 7x leads the development and is a full holder. The notice is never translated. See COPYRIGHT.md and Copyright and copyleft for the history, what the GPL means for you, the third-party notices that stay and how to write the notice in new files and extensions.

This file is part of the source code of Exponential and is distributed with it under the GNU General Public License, version 2 (or any later version).