FrankenPHP for PHP developers

What a PHP developer needs to know when Exponential is served by FrankenPHP through exp:velocity - classic mode with a clean state per request, why worker mode is not offered, the embedded PHP and its ini settings, the...

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

This page is for PHP developers whose code may be served by FrankenPHP, the Caddy web server with PHP compiled into one binary, which `exp:velocity` can drive (`--engine=frankenphp`). FrankenPHP is the production-ready alternative to the recommended `qbix` engine. For PHP code it behaves like PHP-FPM. The full operating guide is FrankenPHP as the Exponential web engine.

Contents

1. Classic mode: a clean state per request
2. The three engines compared
3. The embedded PHP
4. Starting it for a test
5. What to check in your code

Classic mode: a clean state per request

FrankenPHP runs PHP in **classic mode**: a pool of PHP threads inside one process, each request starting from a clean state, exactly as under PHP-FPM. Worker mode, where a PHP script stays alive across requests, is **deliberately not offered**, because the kernel keeps request state in globals and would leak one request into the next. For your code this means: the static state traps of persistent workers do not apply under FrankenPHP (Static state traps), and you pay the bootstrap cost on every request, as under PHP-FPM.

The three engines compared

	`qbix` **(Velocity)**	`frankenphp`	`php` **(built-in server)**
role	recommended, every stage	production-ready alternative	development only
PHP state between requests	kept in persistent workers, reset by the kernel and the engine	none (classic mode)	none
response cache in the server	yes (`[CacheSettings]`)	not yet; `[CacheSettings]` is ignored and `status` says so	no
HTTPS and HTTP/2	yes	yes, self-signed by default	plain HTTP

	`qbix` **(Velocity)**	`frankenphp`	`php` **(built-in server)**
PHP binary	the machine's command line PHP	embedded in the binary	the machine's command line PHP

Testing code on `frankenphp` or `php` next to `qbix` separates bugs in your code from state kept between requests (Velocity engines).

The embedded PHP

FrankenPHP brings its own PHP and extensions; it does not read the machine's `php.ini`. Settings come from `settings/velocity.ini`:

Setting	Effect
`[PHPSettings] IniOptions[]`	shared with the qbix engine, applied first
`[FrankenPHPSettings] IniOptions[]`	applied after them; shipped `display_errors=Off`, `log_errors=On`, `memory_limit=512M`, `opcache.enable=1`, `post_max_size=64M`, `upload_max_filesize=32M`
`[FrankenPHPSettings] PHPIniFile`	a `php.ini` of your own for the embedded PHP

Two consequences for developers:

- **MySQL needs an explicit socket.** Without a machine `php.ini` there is no default MySQL socket; set it in `IniOptions[]` or database connections fail (§6.1 of the guide).
- **Extensions are those of the binary.** A PHP extension your code needs must be in the FrankenPHP build, not only in the machine's PHP; read the guide before you depend on one.

Starting it for a test

Shell

```
php bin/php/console exp:velocity install --engine=frankenphp --check
php bin/php/console exp:velocity start --engine=frankenphp
php bin/php/console exp:velocity status --all
php bin/php/console exp:velocity stop --engine=frankenphp
```

`install` downloads a pinned release and checks its SHA-256 before it is ever run. The engine listens on its own ports (`[FrankenPHPSettings]` `Port=8089`, `HTTPSPort=8444` by default), so it runs beside the qbix engine. The Caddyfile is generated from `velocity.ini` on every start (`var/vc/frankenphp/run/Caddyfile`); logs are under `var/vc/frankenphp/log`. The engine class is `expVelocityFrankenPHP` (`kernel/classes/expvelocityfrankenphp.php`).

What to check in your code

- **No reliance on worker state.** Code that only works because a static survived (a cache warmed in a previous request) is slower or wrong here; code that is correct here and wrong under `qbix` keeps state it should not.
- **Paths and the root.** As everywhere: `EXP_ROOT_DIR` or `eZSys::rootDir()`, not paths

derived from a kernel file (The engine archive (phar) for PHP developers).

- **Errors.** `display_errors` is off by default in this engine; read PHP errors in the logs and in the debug bar.

Related pages

- FrankenPHP as the Exponential web engine
- Velocity persistent workers for PHP developers
- Web server and Solr commands
- 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.