FrankenPHP für PHP-Entwickler

Was ein PHP-Entwickler wissen muss, wenn Exponential über exp:velocity von FrankenPHP ausgeliefert wird - der klassische Modus mit einem sauberen Zustand pro Anfrage, warum der Worker-Modus nicht angeboten wird, das...

Source file: `doc/ger-de/hilfe/php-entwickler/frankenphp-fuer-php-entwickler.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/frankenphp-for-php-developers.md

Diese Seite richtet sich an PHP-Entwickler, deren Code möglicherweise von FrankenPHP ausgeliefert wird, dem Webserver Caddy mit in eine Binärdatei kompiliertem PHP, den `exp:velocity` steuern kann (`--engine=frankenphp`). FrankenPHP ist die produktionsreife Alternative zur empfohlenen Engine `qbix`. Für PHP-Code verhält es sich wie PHP-FPM. Der vollständige Betriebsleitfaden ist FrankenPHP als Web-Engine von Exponential.

Inhalt

1. Klassischer Modus: ein sauberer Zustand pro Anfrage
2. Die drei Engines im Vergleich
3. Das eingebettete PHP
4. Für einen Test starten
5. Was Sie in Ihrem Code prüfen sollten

Klassischer Modus: ein sauberer Zustand pro Anfrage

FrankenPHP führt PHP im **klassischen Modus** aus: ein Pool von PHP-Threads in einem Prozess, wobei jede Anfrage von einem sauberen Zustand ausgeht, genau wie unter PHP-FPM. Der Worker-Modus, in dem ein PHP-Skript über Anfragen hinweg lebendig bleibt, wird **bewusst nicht angeboten**, weil der Kernel Anfragezustand in globalen Variablen hält und eine Anfrage in die nächste durchsickern lassen würde. Für Ihren Code bedeutet das: Die Fallen durch statischen Zustand bei persistenten Workern gelten unter FrankenPHP nicht (Fallen durch statischen Zustand), und Sie zahlen die Bootstrap-Kosten bei jeder Anfrage, wie unter PHP-FPM.

Die drei Engines im Vergleich

	`qbix` **(Velocity)**	`frankenphp`	`php` **(eingebauter Server)**
Rolle	empfohlen, jede Stufe	produktionsreife Alternative	nur Entwicklung
PHP-Zustand zwischen Anfragen	in persistenten Workern erhalten, von Kernel und Engine zurückgesetzt	keiner (klassischer Modus)	keiner
Antwort-Cache im Server	ja (`[CacheSettings]`)	noch nicht; `[CacheSettings]` wird ignoriert, und `status` sagt das	nein

	qbix **(Velocity)**	frankenphp	php **(eingebauter Server)**
HTTPS und HTTP/2	ja	ja, standardmäßig selbstsigniert	einfaches HTTP
PHP-Binärdatei	das Kommandozeilen-PHP der Maschine	in die Binärdatei eingebettet	das Kommandozeilen-PHP der Maschine

Code auf `frankenphp` oder `php` neben `qbix` zu testen trennt Fehler in Ihrem Code von Zustand, der zwischen Anfragen erhalten bleibt (Velocity-Engines).

Das eingebettete PHP

FrankenPHP bringt sein eigenes PHP und seine eigenen Erweiterungen mit; es liest nicht die `php.ini` der Maschine. Einstellungen kommen aus `settings/velocity.ini`:

Einstellung	Wirkung
`[PHPSettings] IniOptions[]`	gemeinsam mit der Engine qbix, zuerst angewendet
`[FrankenPHPSettings] IniOptions[]`	danach angewendet; mitgeliefert `display_errors=Off`, `log_errors=On`, `memory_limit=512M`, `opcache.enable=1`, `post_max_size=64M`, `upload_max_filesize=32M`
`[FrankenPHPSettings] PHPIniFile`	eine eigene `php.ini` für das eingebettete PHP

Zwei Folgen für Entwickler:

- **MySQL braucht einen ausdrücklichen Socket.** Ohne `php.ini` der Maschine gibt es keinen Standard-MySQL-Socket; setzen Sie ihn in `IniOptions[]`, sonst schlagen Datenbankverbindungen fehl (§6.1 des Leitfadens).
- **Die Erweiterungen sind die der Binärdatei.** Eine PHP-Erweiterung, die Ihr Code braucht, muss im Build von FrankenPHP enthalten sein, nicht nur im PHP der Maschine; lesen Sie den Leitfaden, bevor Sie sich auf eine verlassen.

Für einen Test starten

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` lädt ein festgelegtes Release herunter und prüft seine SHA-256-Prüfsumme, bevor es überhaupt ausgeführt wird. Die Engine lauscht auf eigenen Ports (standardmäßig `[FrankenPHPSettings] Port=8089`, `HTTPSPort=8444`), sodass sie neben der Engine qbix läuft. Das Caddyfile wird bei jedem Start aus `velocity.ini` erzeugt (`var/vc/frankenphp/run/Caddyfile`); Logs liegen unter `var/vc/frankenphp/log`. Die Engine-Klasse ist `expVelocityFrankenPHP` (`kernel/classes/expvelocityfrankenphp.php`).

Was Sie in Ihrem Code prüfen sollten

- **Kein Verlass auf Worker-Zustand.** Code, der nur funktioniert, weil eine statische Eigenschaft überlebt hat (ein in einer früheren Anfrage vorgewärmter Cache), ist hier langsamer oder falsch; Code, der hier korrekt und unter `qbix` falsch ist, hält Zustand, den er nicht halten sollte.
- **Pfade und das Wurzelverzeichnis.** Wie überall: `EXP_ROOT_DIR` oder `eZSys::rootDir()`, keine aus einer Kernel-Datei abgeleiteten Pfade (Das Engine-Archiv (phar) für PHP-Entwickler).
- **Fehler.** `display_errors` ist in dieser Engine standardmäßig aus; lesen Sie PHP-Fehler in den Logs und in der Debug-Leiste.

Verwandte Seiten

- FrankenPHP als Web-Engine von Exponential
- Persistente Velocity-Worker für PHP-Entwickler
- Befehle für Webserver und Solr
- 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.