Debugging, Logging und Fehler

Die Gegenstücke in Exponential zum Symfony-Profiler, zu Laravel Telescope und Debugbar, zur Django Debug Toolbar und zu den Loggern der Frameworks – Debug-Ausgabe und die Exp-Debug-Leiste, Kommentare mit...

Source file: `doc/ger-de/hilfe/web-framework-entwickler/debugging-logging-und-fehler.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/web-framework-developers/debugging-logging-and-errors.md

Diese Seite richtet sich an Entwickler, die sich auf eine Profiler-Toolbar und einen PSR-3-Logger verlassen. Exponential hat eine eigene Debug-Ausgabe, die zu einer angehefteten **Debug-Leiste** geworden ist, eigene Logging-Aufrufe und eine Seite in der Administrationsoberfläche, die die Logs liest (Debugging und Logs).

Inhalt

1. Debug-Ausgabe und die Debug-Leiste
2. Debug- und Logzeilen schreiben
3. Die Logdateien
4. Von der Fehlerseite zur Logzeile
5. Debugging unter Velocity
6. Im Vergleich mit anderen Frameworks

Debug-Ausgabe und die Debug-Leiste

INI

```ini
# settings/override/site.ini.append.php, or the dev variant of it (development only)
[DebugSettings]
DebugOutput=enabled

[TemplateSettings]
ShowTemplatePathComments=enabled
```

Die Debug-Leiste zeigt für die Seite die Abfragen, die verwendeten Templates, Einstellungen, Zeitmessungen und Caches und bietet Bedienelemente für die Debug-Einstellungen und die Caches (Exp-Debug-Leiste). Sie lässt sich nur für Ihre eigene Adresse einblenden. Kommentare mit Template-Pfaden geben aus, welche Datei welches Markup geschrieben hat (Kommentare mit Template-Pfaden). Schalten Sie die Debug-Ausgabe auf einem Produktiv-Siteaccess nie für alle ein; sie zeigt Interna. Legen Sie sie in die `dev`-Variante Ihrer Einstellungen (Umgebungen und Geheimnisse).

Debug- und Logzeilen schreiben

PHP

```php
eZDebug::writeNotice( 'Imported 12 items', __METHOD__ );
eZDebug::writeWarning( 'Item 42 had no title', __METHOD__ );
eZDebug::writeError( 'Feed could not be read: ' . $message, __METHOD__ );
eZLog::write( 'archive sent for object ' . (int)$objectID, 'expdemo.log' );
```

Warnungen und Fehler landen unabhängig von der Debug-Einstellung in `var/log/warning.log` und `var/log/error.log`; `eZLog::write()` schreibt ein eigenes Log im Log-Verzeichnis. Das sind keine PSR-3-Logger; wenn Sie Monolog in Ihrer Erweiterung verwenden, schreibt es eigene Dateien. Schreiben Sie nie Passwörter, Schlüssel, Token oder personenbezogene Daten ins Log.

Die Logdateien

Datei	Enthält
`var/log/error.log`, `warning.log`	Fehler und Warnungen
`var/log/security.log`	Sicherheitsereignisse
`var/log/csp.log`	Berichte der Content Security Policy
`var/log/requestrules.log`	die Aktion `log` der Request-Regeln
Ihre eigenen `*.log`	was Ihr Code schreibt

Setup > Logs zeigt sie mit einer Zusammenfassung wiederholter Meldungen an (Log-Viewer); Logs werden rotiert (Log-Rotation); Request-Logs zeichnen Requests mit Zeitmessungen auf (Request-Logs). Sicherheitsrelevante Aktionen gehören in das Audit-Protokoll (Audit-Protokoll). In einer Multisite-Installation können die Log-Pfade je Site verschieden sein (Log- und Cache-Pfade bei mehreren Sites).

Von der Fehlerseite zur Logzeile

Fehlerseiten tragen eine Referenz; durchsuchen Sie die Logs danach, um die vollständige Meldung zu finden (Einen Fehler anhand seiner Referenz finden). Die erste Seite bei jedem Problem ist Wenn etwas schiefgeht.

Debugging unter Velocity

- Worker halten Klassen und Einstellungen im Speicher: Führen Sie nach einer Änderung `./console exp:velocity deploy` (oder `restart`) aus, sonst debuggen Sie alten Code.
- Vergleichen Sie mit der Engine des eingebauten PHP-Servers (`--engine=php`), die jeden Request frisch ausführt, wenn Sie vermuten, dass Zustand zwischen Requests erhalten bleibt (Velocity-Engines).
- Umschließen Sie `eZExecution::cleanExit()` nie mit `catch ( Exception $e )`: Unter Velocity beendet es den Request mit einer Exception.

Im Vergleich mit anderen Frameworks

Framework	Werkzeug	Exponential
Symfony	Web Profiler, Monolog	Debug-Leiste, eZDebug, eZLog, **Setup > Logs**

Framework	Werkzeug	Exponential
Laravel	Debugbar, Telescope, `Log::`	dasselbe
Django	Debug Toolbar, `logging`	dasselbe
Rails	Logs, byebug	dasselbe; Xdebug funktioniert wie in jedem PHP-Projekt
Spring	Actuator, SLF4J	**Setup > Systeminformationen**, Request-Logs

Verwandte Seiten

- Anleitung zur Debug-Leiste
- Verbesserungen der Debug-Ausgabe
- Template-Fehler und Debugging
- 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.