Hinweise zur PHP-8-Migration für Ihren Code

Was Sie an älterem PHP-Code ändern müssen, damit er unter Exponential 6 sauber unter PHP 8.0 bis 8.5 (PHP 7.4 ohne Support nutzbar) läuft - dynamische Eigenschaften, count() auf Nicht-Arrays, null an String-Funktionen…

Source file: `doc/ger-de/hilfe/php-entwickler/hinweise-zur-php-8-migration-fuer-ihren-code.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/php-8-migration-notes-for-your-code.md

Diese Seite richtet sich an PHP-Entwickler, die älteren Code, oft eine für PHP 5 oder 7 geschriebene Erweiterung, für Exponential 6 auf PHP 8 bringen. Der Kernel hat dieselbe Arbeit zwischen Dezember 2023 und April 2026 durchlaufen (Unterstützung für PHP 8); diese Seite macht aus seinen Erfahrungen eine Liste, die Sie abarbeiten können.

Inhalt

1. Wie Sie die Probleme finden
2. Deklarieren Sie die Eigenschaften, die Sie verwenden
3. Prüfen Sie, bevor Sie zählen
4. Übergeben Sie kein null an String-Funktionen
5. Markieren Sie Methoden von SPL-Interfaces
6. Nullbare Parameter
7. Entfernte Funktionen und Konstanten
8. Signaturen, die zum Kernel passen müssen
9. Eine Arbeitsreihenfolge

Wie Sie die Probleme finden

- **Debug-Ausgabe auf einer Entwicklungskopie.** Aktivieren Sie `[DebugSettings] DebugOutput=enabled` und benutzen Sie die Seiten Ihrer Erweiterung; achten Sie auf Hinweise „Deprecated“, die Ihre Dateien nennen.
- **Die Logs.** `var/log/warning.log` und `var/log/error.log` sammeln, was die Debug-Ausgabe zeigt (Logs, der Log-Viewer und Anfrage-Logs).
- **Statische Prüfungen.** `.devtools/vendor/bin/phpcs --standard=phpcs.xml.dist extension/<ext>/` meldet Befunde von PHPCompatibility (entfernte und veraltete Funktionen), und PHPStan findet undefinierte Methoden und falsche Argumentanzahlen (Statische Analyse mit PHPStan).
- **Syntax unter dem ältesten und dem neuesten PHP.** `php -l` mit 8.0 und mit 8.5.

Deklarieren Sie die Eigenschaften, die Sie verwenden

PHP 8.2 markiert das Anlegen einer Eigenschaft, die die Klasse nicht deklariert, als veraltet. Alte Erweiterungen tun das ständig, oft in Klassen, die `eZPersistentObject` oder `eZDataType` erweitern:

PHP

```php
class acmeReport extends eZPersistentObject
{
    /** declared, so PHP 8.2 and later do not warn */
    public $ID;
    public $Title;
}
```

Unterklassen von `eZPersistentObject` erzeugen seit 6.0.4 keine Warnungen mehr in der Debug-Ausgabe, aber die Eigenschaften zu deklarieren ist weiterhin die saubere Lösung, und es dokumentiert die Klasse (Dynamische Eigenschaften).

Prüfen Sie, bevor Sie zählen

`count()` wirft seit PHP 8.0 bei `null` und skalaren Werten einen `TypeError`:

PHP

```php
// before
if ( count( $list ) > 0 ) { ... }

// after
if ( is_array( $list ) && count( $list ) > 0 ) { ... }
// or, for arrays and Countable objects
if ( is_countable( $list ) && count( $list ) > 0 ) { ... }
```

Dasselbe gilt für `foreach` über einen Wert, der `null` sein kann: Sichern Sie ihn ab oder geben Sie ihm einen Standardwert (`$list ?? array()`).

Übergeben Sie kein null an String-Funktionen

PHP 8.1 markiert `null` für nicht nullbare Parameter interner Funktionen als veraltet (`strlen( null )`, `trim( null )`, `preg_match( $p, null )`). Werte aus Einstellungen, der Datenbank und `$Params` sind oft `null`:

PHP

```php
$title = trim( (string)( $row['title'] ?? '' ) );
```

Der Kernel hat das in 6.0.11 bei Aufrufen von `mysqli_real_escape_string()` und `preg_match()` behoben.

Markieren Sie Methoden von SPL-Interfaces

Methoden, die `Iterator`, `ArrayAccess`, `Countable` oder `IteratorAggregate` implementieren, müssen die vorläufigen Rückgabetypen von PHP 8.1 deklarieren oder das Attribut tragen. Der Kernel-Stil hat keine Rückgabetypen, also verwenden Sie das Attribut:

PHP

```php
#[\ReturnTypeWillChange]
public function current()
{
    return $this->items[$this->position];
}
```

PHPStan kann einen fehlenden vorläufigen Rückgabetyp nicht in die Baseline aufnehmen; deshalb schließt `phpstan.neon.dist` `dfs_filter_iterator.php` aus, bis es behoben ist.

Nullbare Parameter

PHP 8.4 markiert einen Parameter als veraltet, der einen Klassentyp und einen Standardwert `null` ohne `?` hat (implizit nullbar). Schreiben Sie das `?` ausdrücklich; nullbare Klassen-Hints erlaubt der Coding-Standard:

PHP

```php
// before
public static function reportDirectory( eZINI $siteINI = null )
// after
public static function reportDirectory( ?eZINI $siteINI = null )
```

Siehe Nullbare Parameter und Typen.

Entfernte Funktionen und Konstanten

Entfernt	Seit	Verwenden Sie
`each()`	8.0	`foreach` oder `key()` und `current()`
`create_function()`	8.0	eine Closure
`split()`, `ereg()`, `eregi()`, `ereg_replace()`	7.0	`explode()`, `preg_split()`, `preg_match()`, `preg_replace()`
`mysql_*`	7.0	eZDB
`E_STRICT`	veraltet in 8.4	lassen Sie es aus Fehlermasken weg
`utf8_decode()`, `utf8_encode()`	veraltet in 8.2	`mb_convert_encoding()`

`phpcs.xml.dist` verbietet `each`, `create_function`, `split`, `ereg`, `eregi`, `ereg_replace`, `mysql_query`, `mysql_connect`, `sizeof` und `delete` (Der Coding-Standard).

Signaturen, die zum Kernel passen müssen

Eine Klasse, die eine Kernel-Methode überschreibt, muss deren Parameter beibehalten: dieselbe Anzahl (mehr mit Standardwerten ist in Ordnung), dieselben Referenzmarkierungen. Unter PHP 8 ist eine Abweichung ein fataler Fehler **beim Laden der Klasse**, sodass die ganze Seite bricht, nicht nur die Methode. Vergleichen Sie beim Aktualisieren einer Erweiterung mit der Elternklasse in `kernel/classes/`; der Boot-Smoke-Test findet solche Fälle (Code schreiben, der unter PHP 8.0 läuft).

Eine Arbeitsreihenfolge

1. Sorgen Sie dafür, dass jede Datei `php -l` unter 8.0 und 8.5 besteht.
2. Laden Sie jede Klasse (Boot-Smoke-Test oder jede Seite öffnen): Korrigieren Sie Signaturen.
3. Beheben Sie Warnungen zu `count()`, `null` und Eigenschaften aus der Debug-Ausgabe, Seite für Seite.
4. Führen Sie PHPCompatibility und PHPStan auf der Erweiterung aus.
5. Heben Sie die PHP-Einschränkung in `composer.json` auf `^8.0` an und veröffentlichen Sie ein Release (Die Checkliste für das Upgrade).

Verwandte Seiten

- Deprecations von PHP 8.1 bis 8.5
- PHP-8-Fehler, die alten Code stoppen
- Datenbankcode für PHP 8
- 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.