Einen Template-Operator in PHP schreiben

Eine Anleitung für einen in PHP geschriebenen Template-Operator - die Registrierung in eztemplateautoload.php, die Operator-Klasse mit operatorList(), namedParameterPerOperator(), namedParameterList() und modify(), Ein…

Source file: `doc/ger-de/hilfe/php-entwickler/einen-template-operator-in-php-schreiben.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/write-a-template-operator-in-php.md

Diese Anleitung richtet sich an PHP-Entwickler, die Templates einen neuen Operator geben, etwa `{$amount|acmereport_money( 'EUR' )}`. Ein Operator ist eine PHP-Klasse, die die Template-Engine (`lib/eztemplate/`) mit dem weitergeleiteten Wert und den Parametern aufruft. Template-Autoren finden die mitgelieferten Operatoren im Abschnitt Template-Operatoren; die Anwendungssicht ist Template-Operatoren schreiben.

Inhalt

1. Schritt 1: den Autoload-Pfad registrieren
2. Schritt 2: den Operator auflisten
3. Schritt 3: die Operator-Klasse
4. Schritt 4: verwenden
5. Ein- und Ausgabe
6. Maskierung
7. Performance und kompilierte Templates
8. modify() testen

Schritt 1: den Autoload-Pfad registrieren

`extension/acmereport/settings/site.ini.append.php`:

INI

```
<?php /* #?ini charset="utf-8"?
[TemplateSettings]
ExtensionAutoloadPath[]=acmereport
*/ ?>
```

Die Template-Engine liest dann `extension/acmereport/autoloads/eztemplateautoload.php`, so wie sie es für `ezjscore` tut (`extension/ezjscore/settings/site.ini.append.php`).

Schritt 2: den Operator auflisten

`extension/acmereport/autoloads/eztemplateautoload.php`:

PHP

```php
<?php
$eZTemplateOperatorArray = array();
$eZTemplateOperatorArray[] = array(
    'script' => 'extension/acmereport/autoloads/acmereporttemplateoperators.php',
    'class' => 'acmeReportTemplateOperators',
    'operator_names' => array( 'acmereport_money' ) );
```

Die Engine lädt die Klasse erst, wenn ein Template einen der Namen verwendet, sodass viele Operatoren nichts kosten, bis sie verwendet werden. `extension/ezjscore/autoloads/eztemplateautoload.php` registriert `json_encode`, `ezscript` und andere auf dieselbe Weise.

Schritt 3: die Operator-Klasse

`extension/acmereport/autoloads/acmereporttemplateoperators.php`:

PHP

```php
<?php
class acmeReportTemplateOperators
{
    function operatorList()
    {
        return array( 'acmereport_money' );
    }

    function namedParameterPerOperator()
    {
        return true;
    }

    function namedParameterList()
    {
        return array(
            'acmereport_money' => array(
                'currency' => array( 'type' => 'string', 'required' => false, 'default' => 'EUR'
),
                'decimals' => array( 'type' => 'integer', 'required' => false, 'default' => 2 ) )
 );
    }

    function modify( $tpl, $operatorName, $operatorParameters, $rootNamespace, $currentNamespace,
                     &$operatorValue, $namedParameters )
    {
        switch ( $operatorName )
        {
            case 'acmereport_money':
            {
                $decimals = max( 0, min( 4, (int)$namedParameters['decimals'] ) );
                $operatorValue = number_format( (float)$operatorValue, $decimals ) . ' ' . $named
Parameters['currency'];
            } break;
        }
    }
}
```

Schritt 4: verwenden

```
{$item.amount|acmereport_money()|wash}
{$item.amount|acmereport_money( 'USD', 0 )|wash}
```

Generieren Sie die Autoloads neu und leeren Sie die Template-Caches (`php bin/php/ezcache.php --clear-tag=template`); unter Velocity verwenden Sie `exp:velocity deploy`.

Ein- und Ausgabe

Was	Wo es ankommt
der weitergeleitete Wert (`$x|op`)	`$operatorValue`; setzen Sie ihn auf das Ergebnis (er wird per Referenz übergeben)
benannte Parameter	`$namedParameters`, mit angewendeten Standardwerten
rohe Parameter	`$operatorParameters`, nicht ausgewertet; verwenden Sie `$tpl->elementValue()` nur, wenn Sie wissen, dass Sie sie brauchen
der Name des Operators	`$operatorName`, sodass eine Klasse mehrere Operatoren bedienen kann

Ein Operator, der ohne weitergeleiteten Wert verwendet wird (`{acmereport_money( 12 )}`), erhält `null` in `$operatorValue`; deklarieren Sie den Wert als Parameter, wenn er so verwendet wird.

Maskierung

Geben Sie Werte zurück, kein HTML, und lassen Sie das Template sie über `|wash` ausgeben. Wenn ein Operator Markup zurückgeben muss, maskieren Sie jedes Eingabestück darin mit `htmlspecialchars( $value, ENT_QUOTES | ENT_SUBSTITUTE | ENT_HTML401, 'UTF-8' )` (die Flags sind für PHP 8.0 nötig, siehe Ausgaben aus PHP maskieren), und fügen Sie niemals Inline-`style=""` oder `<script>` ein (Content Security Policy für Ihren Code).

Performance und kompilierte Templates

- Operatoren laufen bei jedem Rendern, auch in Schleifen. Halten Sie `modify()` schnell und frei von Datenbankabfragen; Daten gehören in eine Fetch-Funktion (Eine Fetch-Funktion schreiben).
- Templates werden zu PHP kompiliert. Reine Kernel-Operatoren können sich mit `operatorTemplateHints()` selbst beschreiben, damit der Compiler sie inline einsetzt (`kernel/common/ezkerneloperator.php` ist ein Beispiel); Ihr Operator funktioniert auch ohne.
- Keine Nebenwirkungen: Innerhalb eines `{cache-block}` läuft der Operator erst wieder, wenn der Block abläuft.

modify() testen

`modify()` ist eine einfache Methode; rufen Sie sie in einem Test ohne Datenbank auf:

PHP

```php
public function testMoneyFormatsTwoDecimals()
{
    $operators = new acmeReportTemplateOperators();
    $value = 12.5;
    $operators->modify( null, 'acmereport_money', array(), '', '', $value,
                        array( 'currency' => 'EUR', 'decimals' => 2 ) );
    $this->assertSame( '12.50 EUR', $value );
}
```

Verwandte Seiten

- Template-Operatoren (Erweiterungsentwickler)
- Das Muster Template-Operator
- Die Template-Sprache
- 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.