Ausgaben aus PHP maskieren

Wie Sie Werte maskieren, die PHP-Code in HTML, Attribute, URLs, JSON, CSV, Shell-Befehle und Logs schreibt - htmlspecialchars() mit ausdrücklichen Flags für PHP 8.0, der Operator wash und warum Templates der bessere Ort...

Source file: `doc/ger-de/hilfe/php-entwickler/ausgaben-aus-php-maskieren.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/escaping-output-from-php.md

Diese Seite richtet sich an PHP-Entwickler, deren Code Werte in Ausgaben schreibt: HTML aus einer View oder einem Operator, JSON aus einem Endpunkt, CSV aus einem Export, einen Shell-Befehl, eine Log-Zeile. Die Regel ist einfach: Maskieren Sie jeden Wert dort, wo er Ihren Code verlässt, im Format des Ortes, an den er geht. Die Template-Sicht auf dieselbe Regel finden Sie unter Eingabevalidierung und Ausgabemaskierung.

Inhalt

1. Bevorzugen Sie Templates
2. HTML aus PHP
3. Attribute, URLs und Daten für Skripte
4. JSON
5. CSV, Shell und Logs
6. Die Kontexttabelle
7. Fehler, die Cross-Site-Scripting verursachen

Bevorzugen Sie Templates

HTML gehört in Templates, wo `|wash` jeden Wert an der Stelle maskiert, an der er ausgegeben wird:

```
<h2>{$item.title|wash}</h2>
<a href={concat( 'acmereport/item/', $item.id )|ezurl} title="{$item.title|wash}">{'Open'|i18n( '
extension/acmereport' )}</a>
```

Eine View übergibt dem Template Werte (`$tpl->setVariable()`), kein HTML. Dann ist die Maskierung an einer Stelle sichtbar, und ein Reviewer kann sie prüfen.

HTML aus PHP

Wenn PHP Markup bauen muss (ein Template-Operator, der HTML zurückgibt, eine Fehlermeldung, der Text einer E-Mail), maskieren Sie jeden Wert mit ausdrücklichen Flags:

PHP

```php
function acmeEscape( $value )
{
    return htmlspecialchars( (string)$value, ENT_QUOTES | ENT_SUBSTITUTE | ENT_HTML401, 'UTF-8' )
;
}

$html = '<span class="acme-status">' . acmeEscape( $item->attribute( 'status' ) ) . '</span>';
```

Übergeben Sie die Flags jedes Mal. PHP 8.1 hat den Standard auf `ENT_QUOTES | ENT_SUBSTITUTE | ENT_HTML401` geändert; der Standard von PHP 8.0 lässt `'` unmaskiert, sodass ein Wert in einem Attribut mit einfachen Anführungszeichen unter PHP 8.0 ausbrechen würde. Der Kernel übergibt die Flags seit 6.0.15 ausdrücklich in `wash` und 27 weiteren Dateien (Unterstützung für PHP 8.0).

Attribute, URLs und Daten für Skripte

- **Attribute**: Setzen Sie jedes Attribut in Anführungszeichen und maskieren Sie den Wert wie oben.
- **URLs**: Bauen Sie Site-URLs mit `eZURI::transformURI()` (PHP) oder `|ezurl` (Templates); kodieren Sie einen Teil einer URL mit `rawurlencode()`, und erlauben Sie für Adressen aus Eingaben nur `http` und `https`.
- **Daten für JavaScript**: Geben Sie Werte niemals in ein Inline-`<script>` aus. Legen Sie sie in ein `data-*`-Attribut oder einen Block `<script type="application/json">`, den Ihr externes Skript liest. Die Content Security Policy der Administration lehnt Inline-Skripte und `style=""` ab (Content Security Policy für Ihren Code).

PHP

```php
$attr = acmeEscape( json_encode( array( 'id' => (int)$item->attribute( 'id' ) ) ) );
$html = '<div class="acme-chart" data-conf="' . $attr . '"></div>';
```

JSON

PHP

```php
header( 'Content-Type: application/json; charset=utf-8' );
echo json_encode( $data, JSON_HEX_TAG | JSON_HEX_AMP | JSON_HEX_APOS | JSON_HEX_QUOT );
```

Die Flags `JSON_HEX_*` machen die Ausgabe sicher, selbst wenn sie in HTML landet. Setzen Sie den `Content-Type`, damit ein Browser die Antwort niemals als HTML rendert. Über ezjscore oder REST geben Sie Arrays zurück und lassen die Schicht sie kodieren.

CSV, Shell und Logs

Ausgabe	Maskierung
CSV, das in Tabellenkalkulationen geöffnet wird	`fputcsv()` für die Anführung; stellen Sie Werten, die mit `=`, `+`, `-` oder `@` beginnen, ein Präfix voran, damit sie nicht als Formeln ausgeführt werden (der CSV-Export des Kernels tut das)
Shell-Befehle	vermeiden Sie sie; andernfalls `escapeshellarg()` für jedes Argument

Ausgabe	Maskierung
Log-Zeilen	entfernen Sie Zeilenumbrüche (`str_replace( array( "\r", "\n" ), ' ', $value )`), damit ein Eintrag keinen anderen vortäuschen kann; protokollieren Sie niemals Geheimnisse oder personenbezogene Daten
E-Mail-Header	entfernen Sie Zeilenumbrüche aus jedem Wert

Die Kontexttabelle

Ausgabe geht in	In PHP	In Templates
HTML-Text	`htmlspecialchars( $v, ENT_QUOTES | ENT_SUBSTITUTE | ENT_HTML401, 'UTF-8' )`	`|wash`
HTML-Attribut	dasselbe, Attribut in Anführungszeichen	`|wash`, Attribut in Anführungszeichen
Site-URL	`eZURI::transformURI()`	`|ezurl`
Teil einer URL	`rawurlencode()`	`|rawurlencode`
JSON	`json_encode()` mit `JSON_HEX_*`	`json_encode()|wash` in einem Attribut
SQL	`$db->escapeString()`, Casts	(kein SQL in Templates)

Fehler, die Cross-Site-Scripting verursachen

- Eine Zahl, die ohne Maskierung ausgegeben wird, „weil sie eine Zahl ist“: Eine spätere Änderung macht Text daraus.
- `htmlspecialchars( $v )` ohne Flags in Code, der unter PHP 8.0 laufen muss.
- Maskieren bei der Eingabe und Speichern maskierten Texts: doppelte Maskierung an einer Stelle, keine bei der nächsten Ausgabe, die es vergisst.
- HTML in einer View bauen und an ein Template übergeben, das es roh ausgibt.
- Ein Wert aus der Anfrage, der in einer Fehlermeldung wiedergegeben wird.
- Ein JSON-Wert, der in einem Inline-Skript ausgegeben wird.

Verwandte Seiten

- Sicherheit in PHP-Code
- Code schreiben, der unter PHP 8.0 läuft
- String-Template-Operatoren
- 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.