Validierung: Eingaben, Datentypen und Escaping

Wo in Exponential ohne Validator-Komponente validiert wird - Inhalte prüfen ihre Datentypen beim Speichern durch Redakteure, eigene Formulare und Parameter prüfen Sie in Ihrer View oder Ihrem Dienst, und Ausgaben werden…

Source file: `doc/ger-de/hilfe/web-framework-entwickler/validierung.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/web-framework-developers/validation.md

Diese Seite richtet sich an Entwickler, die Constraints deklarieren (Attribute des Symfony Validator, Validierungsregeln in Laravel, Formularfelder in Django, Modellvalidierungen in Rails). Exponential hat keine allgemeine Validator-Komponente mit deklarativen Regeln. Validiert wird an drei Stellen, und wer weiß, welche davon gilt, spart viel doppelten Code.

Inhalt

1. Drei Stellen
2. Inhalte: Die Datentypen validieren
3. Ihre Eingaben: in der View oder im Dienst validieren
4. Fehler anzeigen
5. Ausgabe: dort escapen, wo Sie ausgeben
6. Im Vergleich mit Validierungsbibliotheken

Drei Stellen

Daten	Validiert durch	Sie schreiben
Inhalte, die in der Administrationsoberfläche oder einem Bearbeitungsformular im Frontend bearbeitet werden	den Datentyp jedes Attributs mit den Einstellungen des Klassenattributs (Pflichtfeld, Länge, Wertebereiche)	nichts, oder einen eigenen Datentyp
Parameter und Formulare Ihrer eigenen Views	Ihre View oder den Dienst, den sie aufruft	Typumwandlungen, Allow-Lists, Prüfungen, Fehlermeldungen
API-Eingaben	die REST-Schicht für Inhalte; Ihren Controller oder Ihre Serverfunktion für alles andere	dieselben Prüfungen wie für Formulare

Inhalte: Die Datentypen validieren

Wenn Inhalte über die Bearbeitungs-Views gespeichert werden, prüft der Datentyp jedes Attributs die Eingabe in `validateObjectAttributeHTTPInput()` (`kernel/classes/ezdatatype.php`) und antwortet mit einem Zustand von `eZInputValidator`: `STATE_ACCEPTED`, `STATE_INTERMEDIATE` oder `STATE_INVALID`, zusammen mit einer Meldung, die das Bearbeitungsformular neben dem Feld anzeigt. Ob ein Feld ein Pflichtfeld ist, seine

Längenbegrenzungen und Wertebereiche sind Einstellungen des Klassenattributs, die im Klasseneditor gesetzt werden. Eine neue Art von Regel ist ein neuer Datentyp (Eigene Datentypen, Der Lebenszyklus eines Datentyps).

Wenn Code Inhalte mit `eZContentFunctions::createAndPublishObject()` anlegt, werden die Werte in der Form von `fromString()` des Datentyps übergeben; validieren Sie sie in Ihrem Importcode, bevor Sie die Funktion aufrufen.

Ihre Eingaben: in der View oder im Dienst validieren

Aus Eingabevalidierung und Ausgabe-Escaping:

Eingabe	Validieren
IDs und Zahlen	`(int)`, danach eine Bereichsprüfung (`max( 1, min( 100, (int)$limit ) )`)
Auswahlwerte	eine Allow-List: `in_array( $sort, array( 'name', 'published' ), true )`
Text	Längenbegrenzungen, Trimmen, die für das Feld erlaubten Zeichen
E-Mail, URL	`filter_var()`; bei URLs zusätzlich das Schema (nur `https`)
Dateien	Typ anhand des Inhalts und Größe, gespeichert über den Kernel
Weiterleitungsziele	nur Adressen der eigenen Site (Sichere Weiterleitungen)

Adressparameter (`$Params`) sind Zeichenketten aus der URL und genauso wenig vertrauenswürdig wie POST-Daten. Legen Sie die Regeln in eine kleine Validator-Klasse, die Ihre View, Ihr Befehl und Ihre API gemeinsam nutzen, und testen Sie sie mit PHPUnit ohne Datenbank (Testen).

PHP

```php
$errors = array();
$title = trim( (string)$Module->actionParameter( 'Title' ) );
if ( $title === '' )
    $errors['title'] = ezpI18n::tr( 'extension/expdemo', 'Enter a title.' );
elseif ( mb_strlen( $title ) > 200 )
    $errors['title'] = ezpI18n::tr( 'extension/expdemo', 'The title is too long.' );
```

Fehler anzeigen

Stellen Sie das Formular mit den Werten, die der Benutzer eingegeben hat, erneut dar, zeigen Sie jeden Fehler neben seinem Feld und eine Zusammenfassung oben an, und verlinken Sie die Zusammenfassung mit den Feldern, damit Tastatur- und Screenreader-Nutzer sie finden (Formulare, CSRF-Schutz und HTTP-Eingaben):

PHP

```php
$tpl->setVariable( 'errors', $errors );
$tpl->setVariable( 'values', array( 'title' => $title ) );
```

Ausgabe: dort escapen, wo Sie ausgeben

Templates escapen nicht automatisch: Jeder Wert von Benutzern oder aus der Datenbank läuft durch `|wash`, außer Werten, die bauartbedingt bereits sicheres HTML sind (`$module_result.content`, die Ausgabe von `attribute_view_gui`, gerenderter Rich Text) (Die

View-Schicht). Verwenden Sie in PHP je nach Kontext `htmlspecialchars( $v, ENT_QUOTES, 'UTF-8' )`, `json_encode()` oder `$db->escapeString()`.

Im Vergleich mit Validierungsbibliotheken

Funktion der Bibliothek	Exponential
Deklarative Constraints an einem Modell	Einstellungen des Klassenattributs für Inhalte; nichts für Ihre eigenen Tabellen
Validierungsgruppen, Kaskadierung	nein
Übersetzte Meldungen	`ezpI18n::tr()` in Ihrem Code; Datentypen liefern übersetzte Meldungen mit
Darstellung von Formularfehlern	Ihr Template

Wenn Sie deklarative Regeln bevorzugen, können Sie in Ihrer Erweiterung eine Validierungsbibliothek aus Composer verwenden; der Kernel hängt von keiner ab.

Verwandte Seiten

- Formulare und CSRF-Schutz
- Härtung der Datentyp-Eingaben
- Sicherheit für Anwendungen
- 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.