Eine Fetch-Funktion schreiben

Eine Anleitung, um Templates über fetch('module', 'function', hash(...)) Daten zu geben - der Eintrag in function_definition.php, die statische Methode und ihr Ergebnis-Array, Parameter, Fehler, Berechtigungen und...

Source file: `doc/ger-de/hilfe/php-entwickler/eine-fetch-funktion-schreiben.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/php-developers/write-a-fetch-function.md

Diese Anleitung richtet sich an PHP-Entwickler, die möchten, dass Templates ihre Daten mit `fetch( 'acmereport', 'items', hash( 'limit', 5 ) )` lesen. Eine Fetch-Funktion ist eine Deklaration in der `function_definition.php` des Moduls plus eine statische PHP-Methode. Die Template-Seite ist unter Fetch-Funktionen: Daten in Templates lesen beschrieben.

Inhalt

1. Schritt 1: die Deklaration
2. Schritt 2: die Methode
3. Schritt 3: verwenden
4. Wie der Kernel Sie aufruft
5. Das Ergebnis-Array
6. Regeln

Schritt 1: die Deklaration

`extension/acmereport/modules/acmereport/function_definition.php`, neben `module.php`:

PHP

```php
<?php
$FunctionList = array();
$FunctionList['items'] = array(
    'name' => 'items',
    'operation_types' => array( 'read' ),
    'call_method' => array( 'class' => 'acmeReportFunctionCollection',
                            'method' => 'fetchItems' ),
    'parameter_type' => 'standard',
    'parameters' => array(
        array( 'name' => 'status', 'type' => 'string',  'required' => false, 'default' => false )
,
        array( 'name' => 'limit',  'type' => 'integer', 'required' => false, 'default' => 10 ),
        array( 'name' => 'offset', 'type' => 'integer', 'required' => false, 'default' => 0 ) ) )
;
```

Das Modul muss registriert sein (`[ModuleSettings] ExtensionRepositories[]` und `ModuleList[]`); die eigenen Deklarationen des Kernels sehen genauso aus (`kernel/content/function_definition.php`).

Schritt 2: die Methode

`extension/acmereport/classes/acmereportfunctioncollection.php`:

PHP

```php
<?php
class acmeReportFunctionCollection
{
    /**
     * @param string|false $status
     * @param int $limit
     * @param int $offset
     * @return array array( 'result' => acmeReportItem[] )
     */
    public static function fetchItems( $status = false, $limit = 10, $offset = 0 )
    {
        $access = eZUser::currentUser()->hasAccessTo( 'acmereport', 'read' );
        if ( $access['accessWord'] === 'no' )
            return array( 'result' => array() );

        $conds = null;
        if ( $status !== false && in_array( $status, array( 'open', 'closed' ), true ) )
            $conds = array( 'status' => $status );

        $items = eZPersistentObject::fetchObjectList( acmeReportItem::definition(), null, $conds,
                                                      array( 'created' => 'desc' ),
                                                      array( 'offset' => max( 0, (int)$offset ),
                                                             'length' => max( 1, min( 100, (int)$
limit ) ) ) );
        return array( 'result' => is_array( $items ) ? $items : array() );
    }
}
```

Danach `php bin/php/ezpgenerateautoloads.php -e` und die Caches leeren.

Schritt 3: verwenden

```
{def $items = fetch( 'acmereport', 'items', hash( 'status', 'open', 'limit', 5 ) )}
{foreach $items as $item}
    <li>{$item.status|wash} {$item.amount|wash}</li>
{/foreach}
{undef $items}
```

Wie der Kernel Sie aufruft

`eZModuleFunctionInfo` (`lib/ezutils/classes/ezmodulefunctioninfo.php`) liest die Deklaration, ordnet die benannten Parameter des `hash()` den Parametern der Methode **in der Reihenfolge von** `parameters` zu, füllt Standardwerte auf und ruft `call_method` auf. Halten Sie die Parameterliste der Methode in derselben Reihenfolge wie die Deklaration.

Das Ergebnis-Array

Sie geben zurück	Das Template erhält
`array( 'result' => $value )`	`$value`

Sie geben zurück	Das Template erhält
`array( 'error' => array( 'error_type' => 'kernel', 'error_code' => eZError::KERNEL_NOT_FOUND ) )`	`null`
alles ohne `result` (auch `array( 'result' => null )`)	`null` und einen Fehler im Log: die Funktion „did not return a result value“

Geben Sie für „nichts gefunden“ ein leeres Array oder `false` in `result` zurück, damit Templates es ohne Fehler im Log prüfen können.

Regeln

- **Prüfen Sie Berechtigungen innerhalb der Funktion.** Templates werden von vielen Stellen eingebunden; die Funktion ist der einzige Ort, der jeden Aufruf sieht.
- **Begrenzen Sie jedes Limit** und casten Sie jeden Parameter, wie bei jeder Eingabe.
- **Halten Sie sie schlank.** Rufen Sie den Dienst auf, den Ihre Views und Befehle verwenden.
- **Benennen Sie Parameter wie bei den Inhalts-Fetches** (`limit`, `offset`, `sort_by`), damit Template-Autoren richtig raten.
- **Cache-freundlich.** Fetches in `{cache-block}` laufen nur, wenn der Block abläuft; verlassen Sie sich nicht auf Nebenwirkungen.
- **Testen Sie die statische Methode** wie jede andere (PHPUnit ohne Datenbank).

Verwandte Seiten

- Eine eigene Fetch-Funktion schreiben (Anwendungsentwickler)
- Referenz der Fetch-Funktionen
- Das Muster Fetch-Funktion
- 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.