Erweiterungspunkte aus PHP

Wo sich PHP-Code in Exponential einklinkt, ohne den Kernel zu bearbeiten - die Registrierungsmechanismen (directory, handler, autoload, file, design, ini, registry, filter, override), die Erweiterungspunkte, die ein...

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

Diese Seite richtet sich an PHP-Entwickler, die Verhalten ändern oder hinzufügen möchten und den richtigen Ort zum Einklinken suchen. Exponential wird auf einer Handvoll Arten erweitert, die sich wiederholen. `kernel/setup/expradcatalogue.php` (`expRADCatalogue::points()`) listet jeden Erweiterungspunkt auf, und **Setup > RAD** (`/setup/rad`) wird daraus gebildet: 70 Erweiterungspunkte, 65 davon mit einem Werkzeug. Die vollständige Liste mit jeder INI-Zeile ist Erweiterungspunkte.

Inhalt

1. Die Mechanismen
2. Die Punkte, die PHP-Entwickler am häufigsten nutzen
3. Den leichtesten Mechanismus wählen
4. Durch Einstellungen benannte Handler
5. Die RAD-Seiten und die Übersicht

Die Mechanismen

Mechanismus	Wie er gefunden wird	Beispiel
directory	eine Klasse in einem Verzeichnis, das der Kernel durchsucht, benannt durch eine Einstellung `...Directories`	Datentypen, Workflow-Ereignistypen
handler	eine durch eine Einstellung benannte Klasse, geladen über `eZExtension::getHandlerClass()`	Sitzungs-Handler, Cluster-Datei-Handler, Mail-Transport
autoload	eine Klasse, die über einen Autoload-Pfad einer Erweiterung erreicht wird	Template-Operatoren (`ExtensionAutoloadPath[]`)
file	eine Datei an einem Ort, an dem der Kernel nach Namen sucht	`function_definition.php`, `module.php`
design	ein Template in einem Design, gefunden über die Design-Kette	Override-Templates
ini	nur Einstellungen	die meisten Verhaltensschalter
registry	Einträge einer Registry, die `rad.ini` deklariert	Einstellungen der Debug-Leiste, Anbieter von Systemkarten
filter	ein Listener eines filter-Ereignisses in `site.ini [Event] Listeners[]`	`response/output`, `httpcache/store/headers`
override	eine über `var/autoload/ezp_override.php` ersetzte Kernel-Klasse	das letzte Mittel

Die Punkte, die PHP-Entwickler am häufigsten nutzen

Sie möchten	Punkt	Seite
einen Feldtyp hinzufügen	Datentyp	Einen Datentyp schreiben
einen Workflow-Schritt hinzufügen	Workflow-Ereignistyp	Einen Workflow-Ereignistyp schreiben
Seiten hinzufügen	Modul und Views	Schreiben Sie Ihre erste Modul-View
eine Kernel-View, einen Befehl oder einen Cronjob-Teil ändern	Unterklasse über `[RunnableSettings] Implementation[]`	Modul-Views als Klassen
Templates Daten geben	Fetch-Funktion	Eine Fetch-Funktion schreiben
Templates eine Hilfsfunktion geben	Template-Operator	Einen Template-Operator in PHP schreiben
auf den Kernel reagieren	Listener für Kernel-Ereignisse	Ereignisse, Listener und Filter
einen Wert unterwegs ändern	Listener für filter-Ereignisse	Ereignisse, Listener und Filter
JSON für Ihr eigenes JavaScript	serverseitige Ajax-Funktion	ezjscore-JSON-Endpunkte in PHP
neue REST-Routen	REST-Provider	REST und entfernte Dienste aus PHP
eine Richtlinie mit Einschränkungen hinzufügen	Richtlinienfunktion und Einschränkung	Richtlinienfunktionen und Einschränkungen
zeitlich geplante Arbeit	Cronjob-Skript und -Teil	Einen Cronjob-Teil schreiben
einen Befehl	Befehl einer Erweiterung	Einen Konsolenbefehl schreiben

Den leichtesten Mechanismus wählen

Bevorzugen Sie in dieser Reihenfolge:

1. **Einstellungen** (`ini`), wenn eine Einstellung bereits tut, was Sie brauchen.
2. **Ein Template-Override** (`design`), wenn sich nur die Ausgabe ändert.
3. **Einen registrierten Punkt** (directory, handler, autoload, file, registry): Der Kernel ruft Ihre Klasse über ihren Namen auf, und mehrere Erweiterungen können nebeneinander bestehen.
4. **Einen Listener oder Filter**, wenn Sie auf das reagieren oder das anpassen, was der Kernel tut.
5. **Eine Unterklasse über** `[RunnableSettings] Implementation[]`, wenn sich eine Methode einer Kernel-View, eines Befehls oder eines Cronjob-Teils ändern muss.
6. **Einen Kernel-Override**, nur wenn nichts anderes funktioniert. Nichts registriert den Ersatz unter seinem Namen, zwei Erweiterungen, die dieselbe Klasse ersetzen, kollidieren still, und Sie müssen jedem Kernel-Update von Hand folgen (Kernel-Overrides, das letzte Mittel).

Durch Einstellungen benannte Handler

Viele Infrastrukturteile (Sitzung, Cluster-Dateien, Mail-Transport, Suchmaschine, Bild-Handler, statischer Cache) sind durch eine Einstellung benannte Klassen, geladen mit `eZExtension::getHandlerClass()`. Ihre Klasse implementiert dasselbe Interface oder erweitert die Klasse des Kernels, und die Einstellung nennt sie. Ein in einer Einstellung benannter Handler muss existieren, bevor die Einstellung live geht; andernfalls schlägt jede Anfrage fehl, die ihn braucht. Stellen Sie zuerst den Code bereit, danach die Einstellungen, und sichern Sie mit `class_exists()` ab, wo eine fehlende Klasse keine Seiten brechen darf (Durch Einstellungen benannte Handler).

Die RAD-Seiten und die Übersicht

- `/setup/rad` zeigt jeden Erweiterungspunkt, seinen Mechanismus, die Einstellung, die ihn registriert, und ein Werkzeug, das ein Grundgerüst dafür schreibt (Datentyp, Template-Operator, Workflow-Ereignis, Modul, Einstellungserweiterung …) (RAD-Werkzeuge, Die Funktion RAD-Werkzeuge).
- `/setup/radsurvey` zählt, was Ihre Installation an jedem Punkt registriert, die Runnables und ihre Neuimplementierungen, und jeden Eintrag, der nicht funktionieren kann (eine Klasse, die nicht existiert, ein Ersatz, der die zu ersetzende Klasse nicht erweitert).

Verwandte Seiten

- Erweiterungspunkte auf einen Blick
- Die RAD-Erweiterungsoberfläche
- Erweitern ohne Fork
- 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.