Leistung

Für CMS-Entwickler, wie Sie eine Exponential-Website schnell machen und schnell halten. Die Seite behandelt das Messen mit exp:benchmark, was Inhaltsmodell und Templates entscheiden, Regeln für Fetches, Caching je…

Source file: `doc/ger-de/hilfe/cms-entwickler/leistung.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/cms-developers/performance.md

Diese Seite richtet sich an Entwickler, die dafür verantwortlich sind, wie schnell eine Website ist. Der größte Teil der Geschwindigkeit einer Exponential-Website wird von drei Dingen bestimmt, die Sie steuern: wie viel jede Seite abruft, wie viel davon zwischengespeichert werden kann und wie die Website ausgeliefert wird. Messen Sie zuerst, ändern Sie eine Sache, messen Sie erneut.

Inhalt

1. Messen
2. Inhaltsmodell und Templates
3. Fetches
4. Caching
5. Der Server
6. Bilder und das Frontend
7. Caches vorwärmen
8. Eine kurze Checkliste
9. Wenn es in der Produktion langsam ist

Messen

Shell

```
./console exp:benchmark --allow-root-user
./console exp:cache status --allow-root-user
curl -s -o /dev/null -w '%{time_total}s\n' https://www.example.com/
```

`exp:benchmark` ruft die Startseite, Menüseiten, die Suche und die Anmeldung der Administration viele Male auf und gibt Median-, p90- und p99-Zeiten aus; es kann Läufe speichern und mit einer Ausgangsbasis vergleichen und misst auch Teile einer Seite innerhalb des Kernels (Benchmarking von Exponential). Die Debug-Ausgabe listet auf einem Entwicklungsrechner die SQL-Abfragen und die Zeit einer Seite. Ein einzelner Seitenaufruf beweist nichts: Die erste Anfrage füllt die Caches.

Inhaltsmodell und Templates

- Eine **Listenseite**, die fünfzig Elemente zeichnet, jedes mit einzeln abgerufenen Relationen, erzeugt Hunderte von Abfragen. Modellieren Sie so, dass Listenansichten nur die eigenen Attribute des Elements brauchen; verwenden Sie `load_data_map`, wenn Sie Attribute vieler

Knoten lesen.

- **Fetches im Pagelayout** laufen bei jeder Anfrage und sind nicht im View-Cache: Schließen Sie Menüs und Fußzeilen in Cache-Blöcke ein.
- **Nicht cachebare Teile** (der Name des Besuchers, der Warenkorb) sollten klein sein und aus dem View-Cache herausgehalten werden (Caching für CMS-Entwickler).

Fetches

Übergeben Sie immer `limit`; filtern Sie im Fetch nach Klasse; zählen Sie mit `list_count`; vermeiden Sie Fetches in Schleifen; beschränken Sie Attributfilter auf großen Bäumen auf das, was Sie gemessen haben (Fetch-Leistung, Inhalte in Templates abrufen). Code in Ihren Erweiterungen folgt denselben Regeln (Leistung in Erweiterungen).

Caching

Ebene	Was zu tun ist
View-Cache	eingeschaltet lassen (`ViewCaching=enabled`); `full` zwischengespeichert lassen; `cache_ttl` nur wo nötig setzen
Cache-Blöcke	um alles im Pagelayout, was für viele Besucher gleich ist, mit korrekten Schlüsseln
SQL-Abfrage-Cache	standardmäßig an; `./console exp:cache querycache status`
HTTP-Cache, Antwort-Cache von Velocity	ganze Seiten für anonyme Besucher; prüfen Sie mit `exp:cache status`

Jede Ebene: Caching-Ebenen.

Der Server

1. **Velocity** mit der Engine `qbix` hält PHP-Worker zwischen Anfragen geladen und hat einen Antwort-Cache; es ist die empfohlene Engine (Velocity, der Server mit persistenten Workern).
2. **OPcache** für PHP-FPM einschalten; Velocity übergibt eigene Optionen für den Opcode-Cache (OPcache, Velocity und der Opcode-Cache).
3. **Worker**: `velocity.ini [ServerSettings] Workers`, mit der Zahl der CPUs erhöht und nach jeder Änderung gemessen.
4. **Die Datenbank**: eine Engine, die für die Inhalte bemessen ist; siehe Speicher, CPU und Festplatte bemessen.

Die Reihenfolge dieser Schritte mit Befehlen: Schnell machen.

Bilder und das Frontend

- Legen Sie Bildvarianten für die Größen fest, die Sie zeigen; geben Sie `width` und `height` aus; laden Sie Bilder unterhalb des sichtbaren Bereichs verzögert (Bilder und Bildvarianten, Bildleistung).
- Laden Sie Stylesheets und Skripte über den Packer, der sie bündelt und zwischenspeichert (

Der Packer, Theme-Leistung).

- Messen Sie Layoutverschiebung und größtes Element auf echten Seiten (CLS und LCP messen).

Caches vorwärmen

Nach einer Bereitstellung oder dem Leeren eines Cache bezahlen die ersten Besucher für den Aufbau der Caches. `exp:velocity deploy` ruft Seiten zum Vorwärmen auf, und der Preloader ruft die Seiten einer Website auf, wie es ein anonymer Besucher täte (**Setup** > **Preload**, `php bin/php/preload.php --apply`), und listet dabei defekte Links auf (Caches vorladen).

Eine kurze Checkliste

☐ Vor und nach jeder Änderung mit demselben Befehl gemessen.

☐ Keine Seite stellt mehr Abfragen, als ihr Inhalt erklärt; die Debug-Ausgabe wurde auf den schwersten Seiten geprüft.

☐ Jeder Listen-Fetch hat ein Limit; keine Fetches in Schleifen.

☐ Menüs und Fußzeilen des Pagelayouts liegen in Cache-Blöcken mit den richtigen Schlüsseln.

☐ Der View-Cache ist an; besucherbezogene Ausgaben liegen außerhalb.

☐ Ausgeliefert von Velocity (oder einem anderen abgestimmten Server) mit OPcache.

☐ Bildvarianten für jede gezeigte Größe; Assets über den Packer.

Wenn es in der Produktion langsam ist

Folgen Sie dem Runbook Langsame Website: Prüfen Sie die Last, die Cache-Ebenen, die Anfrage-Logs und die langsamsten Seiten, bevor Sie etwas ändern (Anfrage-Logs).

Verwandte Seiten

- Checkliste: Leistung
- Benchmarking einer gehosteten Website
- Eine Website bereitstellen
- 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.