HTTP-Cache-Header

Was Browser, Proxys und CDNs von einer Exponential-Website behalten dürfen - die standardmäßigen Seiten-Header no-cache, zwischenspeicherbare Header nur für anonyme Besucher, Lebensdauern statischer Dateien unter…

Source file: `doc/ger-de/hilfe/hosting/leistung/http-cache-header.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/performance/http-cache-headers.md

Diese Seite ist für Systemadministratoren bei Hosting-Anbietern, die Browser, Proxys oder ein CDN für Kunden-Websites arbeiten lassen. Sie beantwortet: welche Header eine Exponential-Website standardmäßig sendet, wie Sie Seiten für anonyme Besucher zwischenspeicherbar machen, ohne je eine Seite eines angemeldeten Benutzers offenzulegen, wie lange statische Dateien behalten werden dürfen und was der rollenbewusste HTTP-Cache hinzufügt. Die Quellen sind HTTP-Caching für anonyme Besucher und Der rollenbewusste HTTP-Cache.

Inhalt

1. Die Standardwerte
2. Zwischenspeicherbare Seiten für anonyme Besucher
3. Statische Dateien
4. Der rollenbewusste HTTP-Cache
5. Purge-Tags für einen vorgeschalteten Proxy
6. Prüfungen
7. Checkliste

Die Standardwerte

Jede Seitenantwort trägt, sofern nicht anders konfiguriert:

```
Cache-Control: no-cache, must-revalidate
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Pragma: no-cache
```

Also darf kein Browser, kein Proxy und kein CDN eine Seite behalten. Das ist der sichere Standard für einen Hoster: Nichts Persönliches kann über einen geteilten Cache durchsickern, den Sie nicht konfiguriert haben. Der Kernel sendet auf Seiten außerdem seine Sicherheitsheader (`site.ini [HTTPHeaderSettings] SecurityHeaders[]`: `X-Content-Type-Options`, `X-Frame-Options`, eine Content Security Policy mit `frame-ancestors`, `Referrer-Policy`, `Permissions-Policy`).

Zwischenspeicherbare Seiten für anonyme Besucher

Die Einstellungen gibt es schon immer, und sie werden ausgeschaltet ausgeliefert (`CustomHeader=disabled`). Um sie einzuschalten, in `settings/override/site.ini.append.php`:

INI

```ini
[HTTPHeaderSettings]
CustomHeader=enabled
OnlyForAnonymous=enabled
OnlyForContent=enabled

HeaderList[]
HeaderList[]=Cache-Control
HeaderList[]=Expires

Cache-Control[]
Cache-Control[/]=public, max-age=60
Expires[]
Expires[/]=60
```

Dann `php bin/php/ezcache.php --clear-tag=ini --allow-root-user` (und starten Sie Velocity neu, das Einstellungen im Speicher hält).

- `OnlyForAnonymous` **macht das sicher.** Ein angemeldeter Redakteur erhält immer die ursprünglichen Header no-cache, sodass eine personalisierte Seite nie an einen geteilten Cache gegeben wird. Bieten Sie die Einstellung nie ohne sie an.
- `Expires` wird in Sekunden angegeben; der Kernel wandelt es in ein Datum um.
- Der Preis ist Veralterung: Ein anonymer Besucher sieht nach einer Bearbeitung vielleicht eine Seite, die bis zu `max-age` Sekunden alt ist. Sechzig Sekunden sind kurz genug, dass niemand beim Veröffentlichen es bemerkt, und lang genug, um eine Verkehrsspitze aufzufangen.
- Wenn ein konfigurierter Header `Cache-Control` überschreibt, lässt der Kernel das standardmäßige `Pragma: no-cache` weg, damit ein Proxy nicht durch widersprüchliche Header verwirrt wird.

Statische Dateien

Server	Einstellung	Standard
Velocity	`velocity.ini [ServerSettings] StaticMaxAge`	`31536000` (ein Jahr)
FrankenPHP	dieselbe Einstellung, in das erzeugte Caddyfile geschrieben	wie oben
Apache	Regeln für `mod_expires` im virtuellen Host; doc/eng-us/examples/ezpublish.conf hat ein Beispiel	keiner, bis Sie sie hinzufügen
nginx	Ihre eigenen Header im Server-Block	keiner, bis Sie sie hinzufügen

Gemessen auf der Startseite einer Test-Website: 41 Assets, vorher alle bei jedem Aufruf neu geprüft, danach keines. Das macht einen ersten Besuch nicht schneller; es lässt jeden späteren Besuch und jede Navigation innerhalb der Website 41 Roundtrips weniger kosten (HTTP-Caching). Eine unter demselben Namen ersetzte Datei wird erst nach Ablauf ihrer Lebensdauer erneut geholt, was für Kunden zählt, die Stylesheets an Ort und Stelle bearbeiten; der ezjscore-Packer hängt die Änderungszeit an die Namen gepackter Dateien an, wenn `ezjscore.ini [Packer] AppendLastModifiedTime=enabled`, wie das Beispiel für Apache anmerkt.

Der rollenbewusste HTTP-Cache

Standardmäßig aus (`settings/httpcache.ini`, `Enabled=disabled`). Wenn an, speichert er jede Inhaltsansicht pro **Berechtigungskontext**: anonym, ein Rollenkontext, den jeder Benutzer mit denselben Rollen und Einschränkungen teilt, oder ein privater Kontext für einen Benutzer. Die nächste Anfrage im selben Kontext wird beantwortet, bevor der Kernel startet:

- unter Apache, PHP-FPM oder FrankenPHP durch einen frühen Ausstieg, der aus `config.php` eingebunden wird;
- unter Velocity durch den Serverprozess selbst, bevor ein Worker geweckt wird.

Zum Einschalten: `Enabled=enabled` in `settings/override/httpcache.ini.append.php`, den frühen Ausstieg aus `config.php` einbinden (Apache, PHP-FPM, FrankenPHP), Sitzungen in Dateien halten, die jeder Server lesen kann, und Velocity neu starten (HTTP-Cache, einschalten).

Einstellung	Standard	Bedeutung
`CachedSiteAccesses[]`	`site`	welche Siteaccesses zwischengespeichert werden
`MaxAge`	`3600`	längste Lebensdauer einer Seite, auch wenn nichts sie leert
`StaleWhileRevalidate`	`60`	eine abgelaufene Seite wird als STALE ausgeliefert, während eine Anfrage sie neu rendert
`ContentChangePurges`	`all`	jede Änderung an Inhalten leert jede Seite; `tags` nur die Seiten, die zeigen, was sich geändert hat
`QueryStringParameters[]`	keine	Seiten mit einem anderen Query-Parameter werden nicht zwischengespeichert
`APCu`	`enabled`	häufig genutzte Einträge in APCu vor den Dateien
`MaxBodySize`	2 MB	größere Seiten werden nicht gespeichert

Nie zwischengespeichert: POST und andere schreibende Anfragen, Antworten außer 200, Antworten, die ein Cookie setzen, Seiten, deren Template `cache_ttl=0` setzt, alles, was keine Inhaltsansicht ist, und eine Seite, die den Debug-Bericht ihrer Anfrage trägt. Formular-Tokens werden aus der gespeicherten Seite entfernt, und bei jedem Treffer wird das eigene des Besuchers wieder eingesetzt. Der Cronjob-Teil `cache_cleanup` entfernt tote Einträge.

> **Hinweis** Behalten Sie `ContentChangePurges=all`, es sei denn, der Kunde kennt seine Templates: Operatoren, die Inhalte auf eine Weise lesen, die der Cache nicht taggen kann, sind der Grund, warum `all` der Standard ist.

Purge-Tags für einen vorgeschalteten Proxy

`[HttpCacheSettings] TagHeader` nennt die Purge-Tags jeder Seite in einem Header für einen leerenden Proxy davor: `xkey` (Modul xkey von Varnish), `Surrogate-Key` (Fastly) oder `Cache-Tag` (Cloudflare). Standardmäßig ist es `disabled`, weil die Tags interne IDs nennen. Exponential selbst sendet nichts an den Proxy; ihn zu leeren, ist Ihre Aufgabe. Siehe Varnish und Reverse Proxys.

Prüfungen

Prüfen Sie beide Richtungen: Die anonyme Antwort muss zwischenspeicherbar sein, eine Antwort für einen angemeldeten Benutzer nie.

Shell

```
curl -sI https://example.com/ | grep -i -E "cache-control|expires|pragma|x-exp-cache"
curl -sI -H 'Cookie: <session cookie of a signed-in test user>' https://example.com/ | grep -i ca
che-control
```

Erwartet mit den Einstellungen oben: `Cache-Control: public, max-age=60` für die anonyme Anfrage und `Cache-Control: no-cache, must-revalidate` für die angemeldete. Mit eingeschaltetem HTTP-Cache sagt `X-Exp-Cache` `HIT`, `STALE`, `MISS (reason)` oder `BYPASS (reason)`; Setup > Systeminformationen hat einen Kasten zum rollenbewussten HTTP-Cache mit der Trefferquote und den Gründen für Fehlschläge. Nutzen Sie für die Prüfung als angemeldeter Benutzer ein Testkonto, nie das eines Kunden.

Checkliste

- ☐ Seiten-Header nur mit `OnlyForAnonymous=enabled` eingeschaltet.
- ☐ Geprüft, dass Antworten für angemeldete Benutzer `no-cache` bleiben.
- ☐ Lebensdauern statischer Dateien auf dem Server gesetzt; Kunden ist gesagt, dass Bearbeitungen statischer Dateien an Ort und Stelle zwischengespeichert werden.
- ☐ HTTP-Cache: früher Ausstieg eingebunden (ohne Velocity), Sitzungen in geteilten Dateien, Cronjob `cache_cleanup` läuft.
- ☐ `TagHeader` nur eingeschaltet, wenn ein leerender Proxy ihn liest.

Verwandte Seiten

- Cache-Schichten
- Varnish und Reverse Proxys
- CDN
- Der rollenbewusste HTTP-Cache
- 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.