HTTP cache headers

What browsers, proxies and CDNs may keep from an Exponential site: the default no-cache page headers, cacheable headers for anonymous visitors only, static file lifetimes under Velocity and Apache, and the role-aware...

Source file: `doc/eng-us/help/hosting/performance/http-cache-headers.md` Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/hosting/performance/http-cache-headers.md

This page is for system administrators at hosting companies who put browsers, proxies or a CDN to work for customer sites. It answers: which headers an Exponential site sends by default, how to make pages cacheable for anonymous visitors without ever exposing a signed-in page, how long static files may be kept, and what the role-aware HTTP cache adds. The sources are HTTP caching for anonymous visitors and The role-aware HTTP cache.

Contents

1. The defaults
2. Cacheable pages for anonymous visitors
3. Static files
4. The role-aware HTTP cache
5. Purge tags for a proxy in front
6. Checks
7. Checklist

The defaults

Every page response carries, unless configured otherwise:

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

So no browser, proxy or CDN may keep a page. That is the safe default for a host: nothing personal can leak through a shared cache you did not configure. The kernel also sends its security headers on pages (`site.ini [HTTPHeaderSettings] SecurityHeaders[]`: `X-Content-Type-Options`, `X-Frame-Options`, a `frame-ancestors` Content Security Policy, `Referrer-Policy`, `Permissions-Policy`).

Cacheable pages for anonymous visitors

The settings have always existed and ship off (`CustomHeader=disabled`). To switch them on, 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
```

Then `php bin/php/ezcache.php --clear-tag=ini --allow-root-user` (and restart Velocity, which holds settings in memory).

- `OnlyForAnonymous` **is what makes this safe.** A signed-in editor always receives the original no-cache headers, so a personalized page is never handed to a shared cache. Never offer the setting without it.
- `Expires` is given in seconds; the kernel converts it to a date.
- The trade is staleness: an anonymous visitor may see a page up to `max-age` seconds old after an edit. "Sixty seconds is short enough that nobody publishing notices, and long enough to absorb a burst of traffic."
- When a configured header overrides `Cache-Control`, the kernel drops the default `Pragma: no-cache`, so a proxy is not confused by contradicting headers.

Static files

Server	Setting	Default
Velocity	`velocity.ini [ServerSettings] StaticMaxAge`	`31536000` (one year)
FrankenPHP	the same setting, written into the generated Caddyfile	as above
Apache	`mod_expires` rules in the virtual host; doc/eng-us/examples/ezpublish.conf has an example	none until you add them
nginx	your own headers in the server block	none until you add them

Measured on the front page of a test site: 41 assets, all revalidated on every view before, none after. "This does not make a first visit faster. It makes every later visit and every in-site navigation cost 41 fewer round trips" (HTTP caching). A file replaced under the same name is not fetched again until its lifetime ends, which matters to customers who edit stylesheets in place; the ezjscore packer appends the modification time to packed file names when `ezjscore.ini [Packer] AppendLastModifiedTime=enabled`, as the Apache example notes.

The role-aware HTTP cache

Off by default (`settings/httpcache.ini`, `Enabled=disabled`). When on, it stores each content view per **permission context**: anonymous, a role context shared by every user with the same roles and limitations, or a private context for one user. The next request in the same context is

answered before the kernel starts:

- under Apache, PHP-FPM or FrankenPHP by an early exit included from `config.php`;
- under Velocity by the server process itself, before a worker is woken.

To switch it on: `Enabled=enabled` in `settings/override/httpcache.ini.append.php`, include the early exit from `config.php` (Apache, PHP-FPM, FrankenPHP), keep sessions in files every server can read, and restart Velocity (HTTP cache, Switching it on).

Setting	Default	Meaning
`CachedSiteAccesses[]`	`site`	which siteaccesses are cached
`MaxAge`	`3600`	longest life of a page even if nothing purges it
`StaleWhileRevalidate`	`60`	an expired page served as STALE while one request renders it again
`ContentChangePurges`	`all`	any content change purges every page; `tags` only the pages showing what changed
`QueryStringParameters[]`	none	pages with any other query parameter are not cached
`APCu`	enabled	hot entries in APCu in front of the files
`MaxBodySize`	2 MB	larger pages are not stored

Never cached: POST and other writes, responses other than 200, responses that set a cookie, pages whose template sets `cache_ttl=0`, anything not a content view, and a page carrying the debug report of its request. Form tokens are taken out of the stored page and the visitor's own put back on every hit. The `cache_cleanup` cronjob part removes dead entries.

> **Note**
>
> Keep `ContentChangePurges=all` unless the customer knows their templates: operators that read content in ways the cache cannot tag are why `all` is the default.

Purge tags for a proxy in front

`[HttpCacheSettings] TagHeader` names each page's purge tags in one header for a purging proxy in front: `xkey` (Varnish xkey module), `Surrogate-Key` (Fastly) or `Cache-Tag` (Cloudflare). It is `disabled` by default because the tags name internal ids. Exponential itself sends nothing to the proxy; purging it is your job. See Varnish and reverse proxies.

Checks

Check both directions: the anonymous response must be cacheable, and a signed-in response never.

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
```

Expected with the settings above: `Cache-Control: public, max-age=60` for the anonymous request and `Cache-Control: no-cache, must-revalidate` for the signed-in one. With the HTTP cache on, `X-Exp-Cache` says `HIT`, `STALE`, `MISS (reason)` or `BYPASS (reason)`; Setup >

System information has an "HTTP cache (role-aware)" box with the hit rate and why requests missed. Use a test account for the signed-in check, never a customer's.

Checklist

- ☐ Page headers switched on only with `OnlyForAnonymous=enabled`.
- ☐ Signed-in responses checked to stay `no-cache`.
- ☐ Static file lifetimes set on the server; customers told that in-place edits of static files are cached.
- ☐ HTTP cache: early exit included (non-Velocity), sessions in shared files, `cache_cleanup` cronjob running.
- ☐ `TagHeader` switched on only when a purging proxy reads it.

Related pages

- Caching layers
- Varnish and reverse proxies
- CDN
- The role-aware HTTP cache
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

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.