Acronym	Definition
CDN	Canadian/Canada
CDN	Content Delivery Network
CDN	Content Distribution Network
CDN	Crédit du Nord *(French bank)*
CDN	Chinese Domain Name
CDN	Camden-Rockport *(Amtrak station code; Maine)*
CDN	Centrum Doskonalenia Nauczycieli *(Polish: Teacher Training Center)*
CDN	Chicago Daily News *(newspaper; Chicago, IL)*
CDN	Centre Dramatique National *(French: National Drama Center)*
CDN	Content Digital Network
CDN	Call Disconnect Notification
CDN	Call-Disconnect-Notify *(Cisco)*
CDN	Citrix Developer Network
CDN	Canadian Dairy Network
CDN	County District Number *(Texas)*
CDN	Certified Dietitian Nutritionist
CDN	Closing Date Notice *(various organizations)*
CDN	Cable Data Network

```
44  #include "vrt.h"
45
46  /*--------------------------------------------------------------------
47   * TTL and Age calculation in Varnish
48   *
49   * RFC2616 has a lot to say about how caches should calculate the TTL
50   * and expiry times of objects, but it sort of misses the case that
51   * applies to Varnish:  the server-side cache.
52   *
53   * A normal cache, shared or single-client, has no symbiotic relationship
54   * with the server, and therefore must take a very defensive attitude
55   * if the Data/Expiry/Age/max-age data does not make sense.  Overall
56   * the policy described in section 13 of RFC 2616 results in no caching
57   * happening on the first little sign of trouble.
58   *
59   * Varnish on the other hand tries to offload as many transactions from
60   * the backend as possible, and therefore just passing through everything
61   * if there is a clock-skew between backend and Varnish is not a workable
62   * choice.
63   *
64   * Varnish implements a policy which is RFC2616 compliant when there
65   * is no clockskew, and falls as gracefully as possible otherwise.
66   * Our "clockless cache" model is syntehsized from the bits of RFC2616
67   * that talks about how a cache should react to a clockless origin server,
68   * and more or less uses the inverse logic for the opposite relationship.
69   *
70   */
71
72  double
73  RFC2616_Ttl(const struct sess *sp)
74  {
```

CDN cache failover

Closed Person opened this issue

Person commented Contributor

CDN actively caches private believing that the private cache control header does not apply to them as, somehow, they are not a shared cache.

This can lead to information leakage when using a cache value and the set-cookie/authenticated failover as the safety failover will still be cached by this particular instance.

Should consider either making the failover case use `no-cache` rather than `max-age` + `private` at all times or via a flag. Both feel like the wrong solution but this is a pretty large footgun that can popup due to CDN's failure to adhere to the spec.

mnot commented on May 19, 2014

Almost all "reverse proxies" / CDNs don't conform to the HTTP caching spec, because they have a special relationship with the origin server. Each of them has (for better or worse) their own ways to configure whether Cache-Control, etc. are paid attention to.

E.g., reverse proxy has the same sorts of issues, and would likely need special configuration.

Not sure what to do here; we've tried in the past to have a standard for controlling CDNs/reverse proxies, but it didn't really catch on. Maybe it's time to try again...

As far as framework goes, I wouldn't make any changes for this, beyond documenting the potential issues for somebody deploying behind a CDN or reverse proxy.

What value does your CDN/reverse proxy provide?

http-tests / cache-tests

Tests for HTTP Caches https://cache-tests.fyi

Edit

http tests testsuite cache http-cache Manage topics

291 commits 1 branch 0 releases 1 environment 2 contributors BSD-3-Clause

mnot make httpContent deterministic ... Latest commit cce5a2d 6 days ago

results	update results	8 days ago
tests	Document expected_response_headers_missing and describe the new test ...	7 days ago
.editorconfig	fix editorconfig	8 months ago
.gitignore	remove package-lock	4 months ago
CNAME	Create CNAME	4 months ago
CONTRIBUTING.md	Not just CDNs and reverse proxies!	3 months ago
LICENSE	first commit	9 months ago
README.md	Document expected_response_headers_missing and describe the new test ...	7 days ago
cli.mjs	dump is at the test level, not on config	8 days ago

- NodeJS server component
- Browser or NodeJS client component
- Can test caches in browsers and proxies (CDN, forward or reverse)
- Tests written in ~JSON
- Testing for conformance ✅⛔, optimal behaviour ✅⚠️, and information Y/N

HTTP Caching Tests

These tests are a work in progress. The reported results may be faulty, and do not necessarily reflect the true capabilities of each cache. Furthermore, their primary purpose is to inform revision of the HTTP caching specification; so, they should not be used evaluate or compare feature support, since the specification itself might change. To contribute, file bugs or learn more, see README in the repository.

Key: ✅: passed ⚠️: optional test failed ⛔: conformance test failed Y / N: behaviour check results ⁉️: test harness failure 🔷: test failed during setup ⚪: test dependency failed -: not tested

Hover over failed tests for details. Click on ⚪ to copy the test ID to the clipboard.

Cache-Control Parsing	Chrome	Firefox	Safari	Edge	nginx	Squid	ATS	httpd	Varnish	Fastly
These tests check how implementations parse the `Cache-Control` response header. They are not conformance tests because error handling is not clearly specified; rather, they are being used to gather information as input to spec revisions. See also the specification for Cache-Control, and this issue for relevant discussion.										
Does HTTP cache reuse a response with a quoted `Cache-Control: max-age`?	N	N	Y	N	N	N	Y	N	N	N
Does HTTP cache reuse a response with a single-quoted `Cache-Control: max-age`?	N	N	N	N	N	N	Y	N	N	N
Does HTTP cache ignore the phrase `max-age` in a quoted string (before the "real" `max-age`)?	Y	Y	Y	Y	Y	Y	Y	Y	N	N
Does HTTP cache ignore the phrase `max-age` in a quoted string (after the "real" `max-age`)?	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
Does HTTP cache ignore the phrase `max-age` in a quoted string, even when max-age has a quoted value too?	⚪	⚪	Y	⚪	⚪	⚪	Y	⚪	⚪	⚪
Does HTTP cache ignore the phrase `max-age` in a quoted string, even when previous max-age has a quoted value too?	⚪	⚪	Y	⚪	⚪	⚪	Y	⚪	⚪	⚪
Does HTTP cache ignore max-age with space before the =?	Y	N	N	N	Y	Y	N	Y	N	N
Does HTTP cache ignore max-age with space after the =?	N	N	N	N	Y	N	N	Y	N	N
Does HTTP cache reuse a response with an invalid `Cache-Control: max-age` (leading alpha)?	N	N	N	N	N	N	Y	N	N	N
Does HTTP cache reuse a response with an invalid `Cache-Control: max-age` (trailing alpha)	Y	Y	N	Y	N	Y	Y	N	Y	Y

Cache-Control Freshness	Chrome	Firefox	Safari	Edge	nginx	Squid	ATS	httpd	Varnish	Fastly
These tests check whether caches are conformant and optimal in calculating freshness using `Cache-Control`. See the freshness section of the HTTP caching specification, along with the specifics for `Cache-Control`.										
HTTP cache can reuse a response without explict freshness information or a validator (but doing that messes up the tests)	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
An optimal HTTP cache reuses a response with positive `Cache-Control: max-age`	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
HTTP cache must not reuse a response with `Cache-Control: max-age=0`	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
An optimal HTTP cache reuses a response with `Cache-Control: max-age: 2147483648`	✅	⚠️	✅	✅	✅	⚠️	⚠️	✅	✅	✅

royfielding commented 6 days ago

Member

I found the bug that causes Apache httpd's cache to fail on quoted max-age values. It correctly parses the field to extract either max-age=100 or max-age="100", but when it converts the value to a large integer it assumes the value is unquoted.

This is in modules/cache/cache_util.c: ap_cache_control()

```
                if (!ap_cstr_casecmpn(token, "max-age", 7)) {
                    if (token[7] == '='
                            && !apr_strtoff(&offt, token + 8, &endp, 10)
                            && endp > token + 8 && !*endp) {
                        cc->max_age = 1;
                        cc->max_age_value = offt;
                    }
                }
```

“There is no consistency, and with more and more companies relying on multiple CDNs, it’s impossible to debug and troubleshoot.”

– Mehdi Daoudi, “Debugging CDNs: A Need for Change”

https://dzone.com/articles/debugging-cdns-a-need-for-change

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 2, 2019

M. Nottingham
Fastly
August 29, 2018

The Cache HTTP Response Header

draft-nottingham-cache-header-00

Abstract

To aid debugging, HTTP caches often append headers to a response detailing how they handled the request. This specification codifies that practice and updates it for HTTP's current caching model.

Note to Readers

RFC EDITOR: please remove this section before publication

The issues list for this draft can be found at https://github.com/mnot/I-D/labels/cache-header.

The most recent (often, unpublished) draft is at https://mnot.github.io/I-D/cache-header/.

Recent changes are listed at https://github.com/mnot/I-D/commits/gh-pages/cache-header.

See also the draft's current status in the IETF datatracker, at https://datatracker.ietf.org/doc/draft-nottingham-cache-header/.

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 24, 2019

M. Nottingham
Fastly
P. Sikora
Google

February 20, 2019

The Proxy-Status HTTP Header Field

draft-nottingham-proxy-status-00

Abstract

This document defines the Proxy-Status HTTP header field to convey the details of errors generated by HTTP intermediaries.

Note to Readers

RFC EDITOR: please remove this section before publication

The issues list for this draft can be found at https://github.com/mnot/I-D/labels/proxy-status.

The most recent (often, unpublished) draft is at https://mnot.github.io/I-D/proxy-status/.

See also the draft's current status in the IETF datatracker, at https://datatracker.ietf.org/doc/draft-nottingham-proxy-status/.

Precursors that informed this work include (but are not limited to):

- https://docs.fastly.com/guides/debugging/common-503-errors
- https://support.cloudflare.com/hc/en-us/sections/200820298-Error-Pages
- https://cloud.google.com/load-balancing/docs/https/https-logging-monitoring
- https://docs.google.com/document/d/1fMEK80KlpHcL4CwhniQupqQu4MDsxK4v4dKhthiiCMA

- Surrogate-Control
- Invalidation interfaces
- Invalidation grouping ("Cache Tags" / "Surrogate Keys")
- Origin configuration
- CORS handling
- ???

- https://cache-tests.fyi
- https://httpwg.org