

# Which one to use : Expire Header, Last Modified Header or ETags

Asked 9 years, 2 months ago   Active 1 month ago   Viewed 25k times



Imagine yourself at Jelli

[View all 7 job openings!](#)



I am running PHP on Apache, and am confused about how to implement server-side caching, in order to make the site load faster.

34



What is the difference between the `Expires`, `Last-Modified` and `ETag` headers, and which one should be used in what situation?



php

apache2

etag

last-modified

mod-expire

16



edited Apr 5 '15 at 15:45



dayuloli

12.3k ● 12 ● 55 ● 99

asked Mar 16 '11 at 6:43



Avinash

5,444 ● 15 ● 51 ● 92

## 2 Answers

Active

Oldest

Votes



You can use the `Expires` header in conjunction but regardless of the other two. It's universally supported by proxies and browser caches.

27



The difference between `ETag` and `Last-Modified` stamps is more semantic. ETags are opaque to clients. It's usually a checksum. Whereas a Last-Modified header can be interpreted by clients. It's understood that the last modified timestamp works linearly.



If a browser requests a resource with `If-Unmodified-Since`, then a wide range of timestamps in the past can match such a condition. If your pages change frequently then a Last-Modified timestamp might be advantageous.



The ETag approach, on the other hand, leads to clients that save one last fingerprint per resource. (I'm not sure if browser caches remember multiple ETags). On requests, only one or a few possible `If-None-Match` tokens are listed. This might mean more misses. Also, you have to compare multiple checksums, whereas with a Last-Modified timestamp you could have an arithmetic comparison.

The real advantage of ETags is that you can reliably compare fingerprints. The Last-Modified timestamps are a bit more vague, as they don't verify if the actual page content changed.

See also:

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



edited Sep 8 '17 at 3:05

answered Mar 16 '11 at 6:58



Sainan

1,219 ● 1 ● 19 ● 28



mario

134k ● 18 ● 209 ● 265

so basically we should use mod\_expire over eTags? – Avinash Mar 16 '11 at 7:23

then when we should use last modified header? – Avinash Mar 16 '11 at 8:02

- 2 I don't see why timestamps are advantageous to ETags if the page changes frequently. For both technologies, the server makes a simple comparison and sends the same response. – Torsten Bronger Feb 10 '15 at 15:39
- 2 @TorstenBronger - I think using a timestamp is faster on the server side. The If-None-Match request needs to read the entire file, digest it, then make a comparison. the If-Modified-Since request needs to stat the file, then compare to the mtime. Stat-ing a file takes far less time than reading and digesting, especially for large files. – Mjonir74 Sep 8 '15 at 23:33
- 3 RFC 2616 states: "Entity tags are normally "strong validators," but the protocol provides a mechanism to tag an entity tag as "weak."" and "The ETag ...This might allow more reliable validation in situations where it is inconvenient to store modification dates, where the one-second resolution of HTTP date values is not sufficient, or where the origin server wishes to avoid certain paradoxes that might arise from the use of modification dates." and "the preferred behavior for an HTTP/1.1 origin server is to send both a strong entity tag and a Last-Modified value." – cquezel Feb 9 '17 at 18:14



Imagine yourself at Jelli

View all 7 job openings!

`Expires` and `Cache-Control` are "strong caching headers"

43

`Last-Modified` and `ETag` are "weak caching headers"

First the browser checks `Expires/Cache-Control` to determine whether or not to make a request to the servers.



If it has to make a request, it will send `Last-Modified/ETag` in the HTTP request. If the `Etag` value of the document matches that, the server will send a 304 code instead of 200, and no content. The browser will load the contents from its cache.

I recommend using one of the strong caching headers, along with one of the weak caching headers.

See also:

- [Google Web Fundamentals: HTTP-Caching](#)
- [MDN web docs: HTTP caching](#)

edited Apr 8 at 11:55

answered Mar 5 '14 at 4:26



Ludwig



bienbt88

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).




do you find any document support the "strong and weak" caching behavior? I couldn't find one, and my client browser now prioritize last-modified over expires actually, which I don't understand why. – [GMsoF](#) Aug 20 '15 at 2:15

---

Weak and strong validators are described in [RFC 2616](#) section 13.3.3 – [user2418306](#) May 3 '16 at 18:28 

- 
- 4 RFC 2616 says "Entity tags are normally "strong validators," but the protocol provides a mechanism to tag an entity tag as "weak." – [cquezel](#) Feb 9 '17 at 18:06

---

I believe that the strong vs weak distinction of RFC2616 / ETags is a different one. I believe the poster of this answer means that Expires and Cache-Control are "strong" in the sense that they cause the browser to make *no request at all*, while Last-Modified and ETag are weaker in the sense that the browser still needs to make a request (but the server might not need to retransmit the entire resource). – [Matthijs Kooijman](#) yesterday 

---