

ETag vs Header Expires

Asked 11 years, 4 months ago Active 8 months ago Viewed 130k times



I've looked around but haven't been able to figure out if I should use both an ETag **and** an Expires Header **or** one or the other.

359



What I'm trying to do is make sure that my flash files (and other images and what not only get updated when there is a change to those files.



221

I don't want to do anything special like changing the filename or putting some weird chars on the end of the url to make it not get cached.



Also, is there anything I need to do programatically on my end in my PHP scripts to support this or is it all Apache?

http

caching

etag

expires-header

edited Sep 24 '19 at 10:33



Raedwald

36.4k ● 25 ● 113 ● 191

asked Feb 1 '09 at 1:12



GeoffreyF67

9,953 ● 10 ● 38 ● 56

- 1 developer.mozilla.org/en-US/docs/Web/HTTP/Caching (good in general) – [Christophe Roussy](#) Oct 19 '17 at 14:32

8 Answers

Active

Oldest

Votes



677



They are slightly different - the ETag does not have any information that the client can use to determine whether or not to make a request for that file again in the future. If ETag is all it has, it will always have to make a request. However, when the server reads the ETag from the client request, the server can then determine whether to send the file (HTTP 200) or tell the client to just use their local copy (HTTP 304). An ETag is basically just a checksum for a file that semantically changes when the content of the file changes.



The Expires header is used by the client (and proxies/caches) to determine whether or not it even needs to make a request to the server at all. The closer you are to the Expires date, the more likely it is the client (or proxy) will make an HTTP request for that file from the server.

So really what you want to do is use BOTH headers - set the Expires header to a reasonable value based on how often the content changes. Then configure ETags to be sent so that when clients DO send a request to the server, it can more easily determine whether or not to send the file back.

One last note about ETag - if you are using a load-balanced server setup with multiple machines running Apache you will probably want to turn off ETag generation. This is because inodes are

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



the timestamps on the files are exactly the same, to ensure the same ETag gets generated for all servers.

edited Sep 11 '14 at 8:46



Oliver Weichhold

9,105 ● 4 ● 38 ● 79

answered Feb 1 '09 at 3:18



Marc Novakowski

40.7k ● 10 ● 53 ● 61

- 12 You should also check out whether you should be using Cache-Control instead of Expires. My understanding is that Cache-Control was introduced after Expires and gives you more control. See stackoverflow.com/questions/5799906/... – Luis Perez Jan 8 '12 at 1:58
- 6 When using the Expires header it's good practice to change the file name whenever a resource changes, as the client won't request the file again before it's stale. Especially if you're using far future values as expires date. – schnatterer Jul 8 '14 at 10:26
- 8 Lets say we will use the both. What happens when the Expire time has expired, but the file is not changed (The Etag is same)? The server will return 304 and the file will be served from browser cache. My question is, will be regenerated the Expire time at this moment ? – user345602 Feb 18 '15 at 19:39
- 2 Be careful setting ETAG *and* the Expires header to a non-zero value. This can lead to race conditions. See jakearchibald.com/2016/caching-best-practices – Weston Jul 28 '16 at 17:54
- 2 Is it possible to tell the servers not to use inodes or timestamps at all? Also, why are they needed for ETags, if it's used to represent content only? – Seza Feb 14 '17 at 17:05

Etag and Last-modified headers are **validators**.

108

They help the browser and/or the cache (reverse proxy) to understand if a file/page, has changed, even if it preserves the same name.

Expires and Cache-control are giving **refresh information**.

This means that they inform, the browser and the reverse in-between proxies, up to what time or for how long, they may keep the page/file at their cache.

So the question usually is which one validator to use, etag or last-modified, and which refresh information header to use, expires or cache-control.

edited Nov 9 '14 at 14:29



Adib Aroui

4,377 ● 5 ● 34 ● 76

answered Oct 14 '11 at 4:13



john

1,264 ● 1 ● 9 ● 10

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

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

First the browser check `Expires/Cache-Control` to determine whether or not to make a request to the server

If have to make a request, it will send `Last-Modified/ETag` in the HTTP request. If the `ETag` value

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

answered Mar 4 '14 at 15:45



hienbt88

999 ● 8 ● 12

1 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 2:21

1 @GMsoF You may want to take a look at this: tools.ietf.org/html/rfc7232#section-2.1 – [Medeiros](#) Feb 19 '16 at 2:14

So, if I want to make sure that my changes are propagated to the client immediately, but still benefit to some caching, I can only use Last-Modified and ETag right? – [Sebastien Lorber](#) Oct 26 '16 at 12:47



18



By default, Apache will generate an Etag [based on](#) the file's inode number, last-modified date, and size, which should be perfectly fine to do what you want. I think it also will generate by default a Last-Modified header based on the last modification time of the file on disk, which is also perfectly fine to do what you want.

You should probably also have Apache send an Expires header dated one year in the future (according to <http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.21>) so that browsers know the content is cacheable. Have a look at [mod_expires](#) to configure that.

answered Feb 1 '09 at 1:25



David Z

109k ● 23 ● 213 ● 254

So the ETag will have the last modified and the expires header will tell it I want it to cache it and when I upload and overwrite my file it will just get pulled down to the user's cache again otherwise a 304 would be generated right? – [GeoffreyF67](#) Feb 1 '09 at 2:34

The Etag depends in a complicated way on the last-modified date; but when you modify the file, the Etag will change. Then the Etag sent by the browser (for its cached version of the file) won't match the Etag of the file on the server, and Apache will send the file rather than a 304 response. – [David Z](#) Feb 1 '09 at 6:30

6 Won't the one year expires header tell the client to not even check for a new version for one year? – [John Bachir](#) Jan 10 '11 at 22:41

@John: yeah, I guess I was thinking of static content that never changes when I wrote that. – [David Z](#) Jan 10 '11 at 23:23

2 @John Bachir: that should be the expected behaviour, but browser seems to hit anyway the server at least to inquiry for a newer version. I opened another question about this issue: stackoverflow.com/questions/10048740/... – [Marco Demaio](#) Apr 7 '12 at 16:16



13



Another summary:

You need to use both. ETags are a "server side" information. Expires are a "Client side" caching.

- Use **ETags** except if you have a load-balanced server. They are safe and will let clients know

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



- **Expires** must be used with caution, as if you set a expiration date far in the future but want to change one of the files immediately (a JS file for instance), some users may not get the modified version until a long time!

answered Jul 17 '14 at 17:43

**Benjamin Piette**

3,037 ● 1 ● 21 ● 21

- 2 In case of this Expires situation you basically need to rename your js and change it in your HTML, and hope you didn't set HTML file's expires 1 year as well. – [EralpB](#) May 8 '17 at 9:58



1



One additional thing I would like to mention that some of the answers may have missed is the downside to having both `ETags` and `Expires/Cache-control` in your headers.

Depending on your needs it may just add extra bytes in your headers which may increase packets which means more TCP overhead. Again, you should see if the overhead of having both things in your headers is necessary or will it just add extra weight in your requests which reduces performance.

You can read more about it on this excellent blog post by Kyle Simpson:

<http://calendar.perfplanet.com/2010/bloated-request-response-headers/>

answered Oct 14 '16 at 19:21

**aneagoie**

23 ● 4



1



In my view, With Expire Header, server can tell the client when my data would be stale, while with Etag, server would check the etag value for client' each request.

answered Dec 22 '16 at 8:36

**Sunny**

151 ● 1 ● 6



0



ETag is used to determine whether a resource should use the copy one. and Expires Header like Cache-Control is told the client that before the cache decodes, client should fetch the local resource.

In modern sites, There are often offer a file named hash, like `app.98a3cf23.js`, so that it's a good practice to use Expires Header. Besides this, it also reduce the cost of network.

Hope it helps ;)

answered Jan 11 '19 at 7:00

**Jack Chen**

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



