

# Browser Caching

## What is browser caching?

- Browser caching stores webpage resource files on a local computer when a user visits a web page.
- “Leveraging” browser caching is when a webmaster has instructed browsers how their resources should be dealt with.

When a web browser displays your webpage it has to load several things like your logo, your CSS file, and other resources.

What browser caching does is “remember” the resources that the browser has already loaded. When a visitor goes to another page on your website your logo, CSS files, etc. do not need to be loaded again, because the browser has them “remembered” (saved). This is the reason that the first view of a web page takes longer than repeat visits.

When you leverage browser caching, your webpage files will get stored in the browser cache. Your pages will load much faster for repeat visitors and so will other pages that share those same resources.

If you have tested your webpage for speed and found out that you need to leverage browser caching, here is how you do it.

---

## How to leverage browser caching

1. Change the request headers of your resources to use caching.
2. Optimize your caching strategy.

Let's talk about them in detail,

## 1. Change the request headers of your resources to use caching

For most people, the way to enable caching is to add some code to a file called .htaccess on your web host/server. This means going to the file manager (or wherever you go to add or upload files) on your web host. The .htaccess file controls many important things for your site. If you are not familiar with the .htaccess file, please read my working with .htaccess article to get some know how before changing it.

### Browser caching for .htaccess

The code below tells browsers what to cache and how long to “remember” it. It should be added to the top of your .htaccess file.

```
## EXPIRES CACHING ##
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType image/jpg "access 1 year"
ExpiresByType image/jpeg "access 1 year"
ExpiresByType image/gif "access 1 year"
ExpiresByType image/png "access 1 year"
ExpiresByType text/css "access 1 month"
ExpiresByType text/html "access 1 month"
ExpiresByType application/pdf "access 1 month"
ExpiresByType text/x-javascript "access 1 month"
ExpiresByType application/x-shockwave-flash "access 1
month"
ExpiresByType image/x-icon "access 1 year"
ExpiresDefault "access 1 month"
</IfModule>
## EXPIRES CACHING ##
```

Save the .htaccess file and then refresh your webpage.

### **How to set different caching times for different file types**

You can see in the above code that there are time periods like “1 year” or “1 month”. These are associated with file types, as an example the above code states that a .jpg file (image) should be cached for a year.

If you want to change that and say you want your jpg images cached for a month you would simply replace “1 year” with “1 month”. The values above are pretty optimized for most web pages and blogs.

### **Alternate caching method for .htaccess**

The above method is called “Expires” and it works for most people using .htaccess so it takes care of caching for most people who are just getting started.

After you are more comfortable with caching, you may want to try Cache-Control, another method of caching which gives us more options.

It is also possible the expires method did not work for your server, in that case you may want to try to use Cache-Control.

## **2. Cache-Control**

Note: I have made a more complete guide to Cache-Control [here](#).

Cache-Control allows us to have a bit more control of our browser caching and many people find it easier to use once setup.

Example use in .htaccess file:

```
# 1 Month for most static assets
<filesMatch ".(css|jpg|jpeg|png|gif|js|ico)$">
Header set Cache-Control "max-age=2592000, public"
</filesMatch>
```

The above code is setting a cache control header depending on file type.

### How cache-control works

Let's take the above code line by line.

```
# 1 Month for most static assets
```

The above line is just a note. It does not do anything except notate what we are doing. The .htaccess file ignores lines that start with the character #. This note is recommended as you may have several different sets of these as your caching solution grows.

```
<filesMatch "(css|jpg|jpeg|png|gif|js|ico)$">
```

The above line is saying that "if the file is one of these types, then we will do something to it...

The important part of this line is to notice that there are different types of files listed (CSS, js, jpeg, png, etc.) and that the caching instructions that follow will apply to those file types. As an example, if you did not want your jpg files to be cached for this amount of time you could delete "jpg" from this line or if you wanted to add html to this you could just add "html" to this line.

```
Header set Cache-Control "max-age=2592000, public"
```

The above line is where the actual headers are inserted and the values given.

- The "Header set Cache-Control" part is setting a header.
- The "max-age=2592000" part is stating how long it should be cached (using seconds). In this case we are caching for one month which is "2592000" seconds.
- The "public" part is stating that this is public (which is good if you want it to be cached).

```
</filesMatch>
```

The above line is closing the statement and ends the block of code.

### **Common caching issue**

If you list your html and images to be cached for one year or some other long time period, remember that this can mean if you make a change to your pages they may not be seen by all users. This is because the users will look to cached files rather than the live ones. If you have file that you tweak occasionally (example – a CSS file) you can overcome the cache issue by using URL fingerprinting.

### **URL fingerprinting**

Getting a fresh (not cached) file resource is possible by having a unique name. An example would be if our CSS file was named “main.css” we could name it “main\_1.css” instead. The next time we change it we can call it “main\_2.css”. This is useful for files that change occasionally.

### **Caching methods**

It is important to specify one of Expires or Cache-Control max-age, and one of Last-Modified or ETag, for all cacheable resources. It is redundant to specify both Expires and Cache-Control: max-age, or to specify both Last-Modified and ETag.