Support

Find answers, guides, and tutorials to supercharge your content delivery.

Cache-Control - How to Properly Configure It

Updated on October 4, 2018

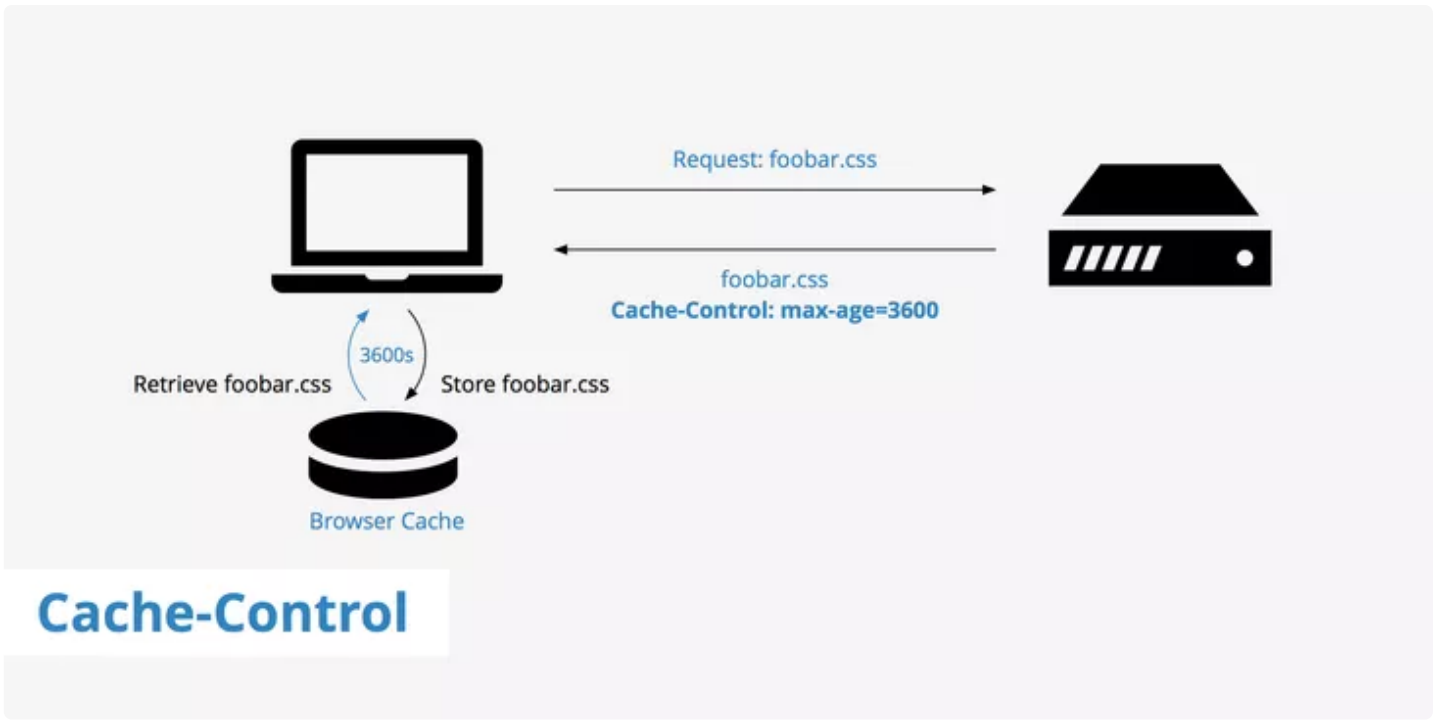


Table of contents

What is `Cache-Control`?

`Cache-Control` is an HTTP cache header comprised of a set of directives that allow you define when / how a response should be cached and for how long. HTTP caching occurs when a browser stores copies of resources for **faster access**. Access to these copied resources is much faster as the browser does not need to make a request to the server in order to receive the same files every time. You can configure your server to attach the `Cache-Control` header in the response, specifying which directives to use.

`Cache-Control` directives

The following is a list of the common directives used and configured when using the `Cache-Control` header. See HTTP/1.1 section 14.9 for a further explanation of the directives available.

`Cache-Control: no-cache`

`no-cache` uses the ETag header to tell caches that this resource cannot be reused without first checking if the resource has **changed on the origin server**. This means that `no-cache` will make a [illegible] therefore is not required to

`Cache-Control: no-store`

Cache-Control: no-store

`no-store` is similar to `no-cache` in that the response cannot be cached and re-used, however there is one important difference. `no-store` requires the resource to be requested and downloaded from the origin server each time. This is an important feature when dealing with private information.

Cache-Control: public

A response containing the `public` directive signifies that it is allowed to be cached by any intermediate cache. This however is usually not included in responses as other directives already signify if the response can be cached (e.g `max-age`).

Cache-Control: private

The `private` directive signifies that the response can only be cached by the browser that is accessing the file. This disallows any intermediate caches to store the response.

Cache-Control: max-age=<seconds>

This directive tells the browser or intermediary cache how long the response can be used from the time it was requested. A `max-age` of `3600` means that the response can be used for the next 60 minutes before it needs to fetch a new response from the origin server.

Cache-Control: s-maxage=<seconds>

`s-maxage` is similar to the above mentioned `max-age` however the "s" stands for shared and is relevant only to CDNs or other intermediary caches. This directive overrides the `max-age` and `expires` header.

Getting Started

Zones

Integrations

Account Management

Troubleshooting

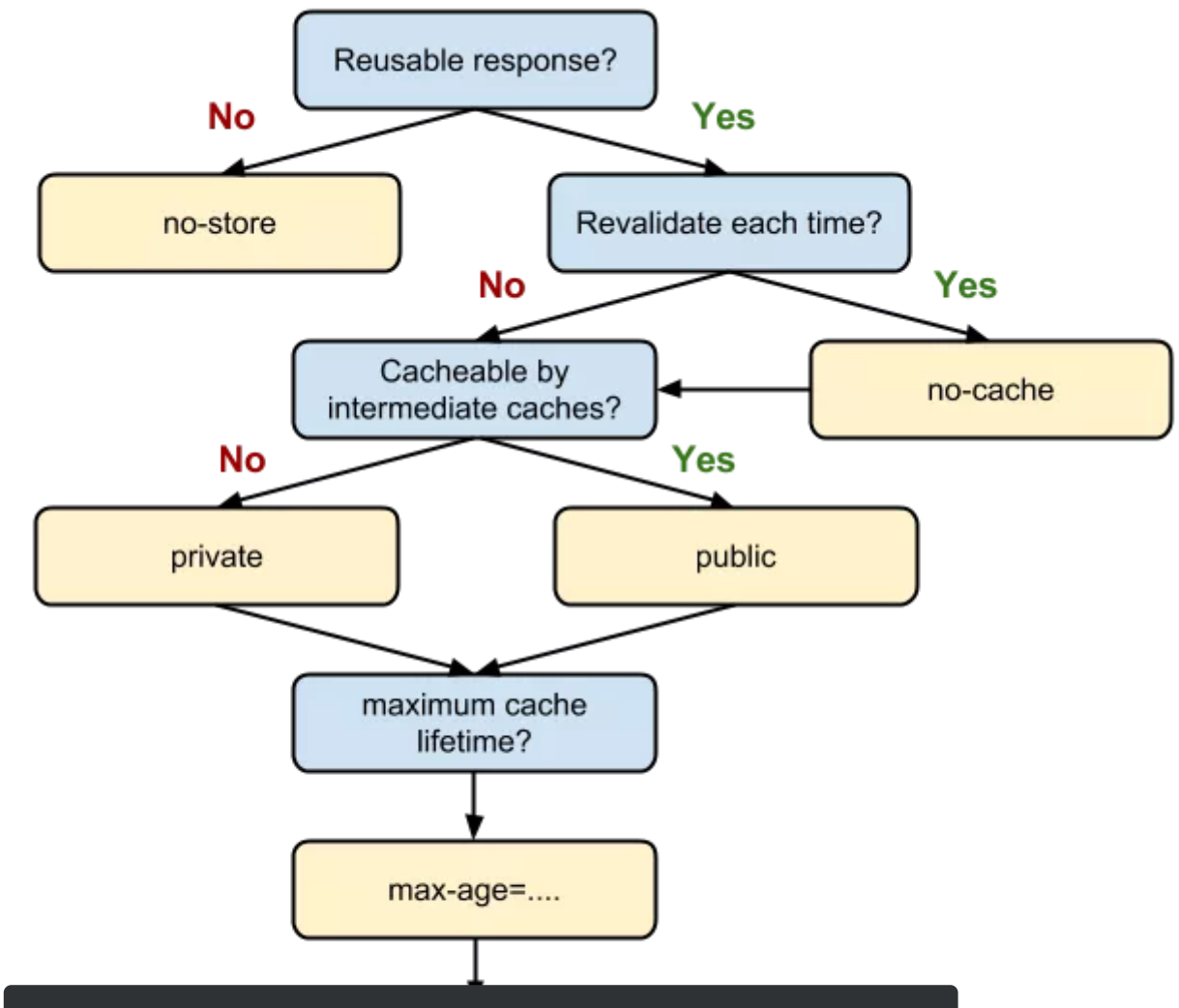
Tutorials

Glossary

Cache-Control: no-transform

Intermediate proxies sometimes change the format of your images and files in order to improve performance. The `no-transform` directive tells the intermediate proxies not to alter the format or your resources.

Each of these directives serves its own purpose and can be used in a variety of scenarios. To help further simplify things, Ilya Grigorik, a developer at Google, created the decision tree below to help determine what directives should be set for a particular resource.



Source: Google Developers

`Cache-Control` configuration

The HTTP `Cache-Control` header can be implemented on the server or can even be added within the code. The following are examples of how to implement `Cache-Control` in Apache, Nginx, or within your PHP code.

Apache

The following snippet can be added to your .htaccess file to tell the server to set the `Cache-Control` header's `max-age` to `84600` seconds and to `public` for the listed files. `Expires` and `Cache-Control` headers can also be included with Apache by using the `mod_expires` module.

```
<filesMatch ".(ico|pdf|flv|jpg|jpeg|png|gif|js|css|swf)$">
    Header set Cache-Control "max-age=84600, public"
</filesMatch>
```

Nginx

This snippet can be added to your Nginx configuration file. The example below uses the `Cache-Control` header directives `public` and `no-transform` with an expire setting set to 2 days.

```
location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
    expires 2d;
    add_header Cache-Control "public, no-transform";
}
```

PHP

`Cache-Control` headers can also be added directly in your code. This example demonstrates using the PHP header to include `Cache-Control` setting a `max-age` of 1 day.

```
header('Cache-Control: max-age=84600');
```

Summary

`Cache-Control` is a powerful HTTP header when it comes to **speeding up websites** with the use of browser and intermediary cache. Although its ability to increase website speed is not it's only as it is also quite useful to help make private information less vulnerable. The settings you choose to apply to the `Cache-Control` directives are dependent on the nature of information that is being delivered as well the desired expiration time of those assets.