The Beginner's Guide to Faster Websites

Optimize your website for happy visitors and a better ranking in Google!

How to Reduce HTTP Requests?

Fast loading is one of the most important factors for **Blogs** and **Websites** do well on Google and other search engines. If your Website doesn't meet those requirements, you may not only be annoying to the user, but also be punished by Google.

The higher the HTTP Requests are on your Website, the slower it will be. So how do you reduce the number of HTTP Requests?

Combine your CSS and JavaScript

It's likely that your project consists of more than one CSS and JavaScript files. Each of these is making an HTTP Request so it can load up on your website.

A "preprocessor" can help you combine your CSS and JavaScript files into a single one (for each) which can dramatically reduce the number of HTTP Requests your Website makes and therefore, make it lighter and faster to the user.

It's likely that if you are using a framework such as Ruby on Rails or Django, that they already help you with that, but if you are not sure, check out tools like CodeKit (Mac), Prepros (Mac, Win, Linux) or the corresponding plugins for Grunt or Gulp.

Combine your Images in CSS Sprites

The same principle is applicable to images: instead of serving each image file on its own, you can combine them in a so-called "CSS Sprite".

With a handful of images in the same file, you can then reference the image via CSS as a background. When width, height, and background properties are properly set, you'll be able to use the same physical file to display different images.

One important thing to note with this approach is that CSS Sprites work great when combining many smaller images into a single file. This makes it perfect for logos, icons, and other smaller image assets. On the other hand, large-format images are best left in their own, separate files, because in this case, the work load from the compression algorithms won't be as efficient and could increase the memory consumption in the browser.

You don't have to do this manually. Here are a couple of tools that help you create CSS

You don't have to do this manually. Here are a couple of tools that help you create CSS Sprites:

- SpritePad
- Glue
- SpriteCow
- Stitches
- TexturePacker

Avoid Redirects

Redirects cause roundtrips to the server(s) just like any other request. If you're unlucky, additional tasks like DNS lookups and TCP handshakes have to be performed as well.

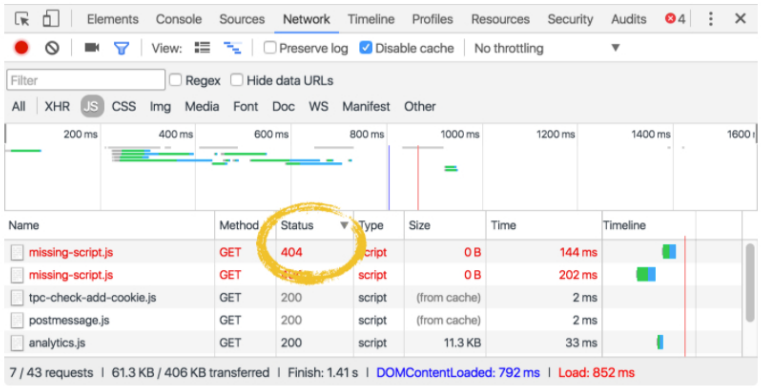
Each redirect your Website makes entails (at least) another HTTP Request, so it's good have it reduced to a minimum.

For example: having a modern and responsive website might be well worth it rather than redirecting users to a mobile version of it ("m.yourdomain.com").

Fix 404 Errors

Even if a request returns no data, it is still an HTTP Request - even worse, a useless one in this case.

So make sure to check any 404 errors you may have due to forgetting to update pointers of resources that no longer exist, for example. You can simply check that via Chrome's Developer Tools, sorting it by the response "Status" and seeing any bad requests:



Learn More

- How to Optimize your CSS and JavaScript
- More frequently asked questions about The Beginner's Guide to Faster Websites

Your email address

About Us

As the makers of Tower, the best Git client for Mac and Windows, we help over 100,000 users in companies like Apple, Google, Amazon, Twitter, and Ebay get the most out of Git.

Just like with Tower, our mission with this platform is to help people become better professionals.

That's why we provide our guides, videos, and cheat sheets (about version control with Git and lots of other topics) for free.