

A simple guide to speeding up your client's website.





Jay runs a successful web agency. He works with a team of six designers, developers and thirty-five clients. Recently, he has been busy working “on” his business to make sure client projects are handled with maximum efficiency. Every client website that is ready to launch needs to go through a six-step process.

Jay is meticulous and a perfectionist. He doesn't like to rely on “tribal knowledge” that is haphazardly passed from one team member to another. Instead, he likes to rely on a system and a process that documents all tasks, check lists and workflows that his team members must follow to get a client site completed in a timely and efficient manner. He understands that it is human nature to forget critical steps when implementing a project, and so by creating a system that includes workflows and task lists, he frees his team from worrying about if they are getting all the steps right enabling them to concentrate on being creative and focus on their core work.

“47% of people expect a web page to load in two seconds or less.”

The process in Jay’s mind was to list everything needed to speed up his client’s sites.

Jay decides to step out of the office and head to the coffee shop to work on this checklist. His checklist/workflow will eventually be set up as a workflow in [Brightpod](#) - their project management software of choice.

Jay orders a cappuccino and settles himself down.

Nobody likes to wait.

Jay has to wait for a while before he is served his cappuccino. Normally, its delivered to him in a timely manner. The barista behind the counter looked overworked. Jay thought for a moment. It didn’t matter how well the coffee shop is laid out and designed to feel intimate on a rainy morning. It’s not a good experience if people have to wait for their coffee. Similarly, it didn’t matter how well a site is designed and how much time is spent getting the perfect colour combinations and pixel positions. If it is slow to load, visitors to the website are going to abandon the site in the blink of an eye.

A slow website is a marketer’s nightmare.

Jay put this down in his notes: “Don’t let a slow loading website ruin all what you have done to get people to your site.”

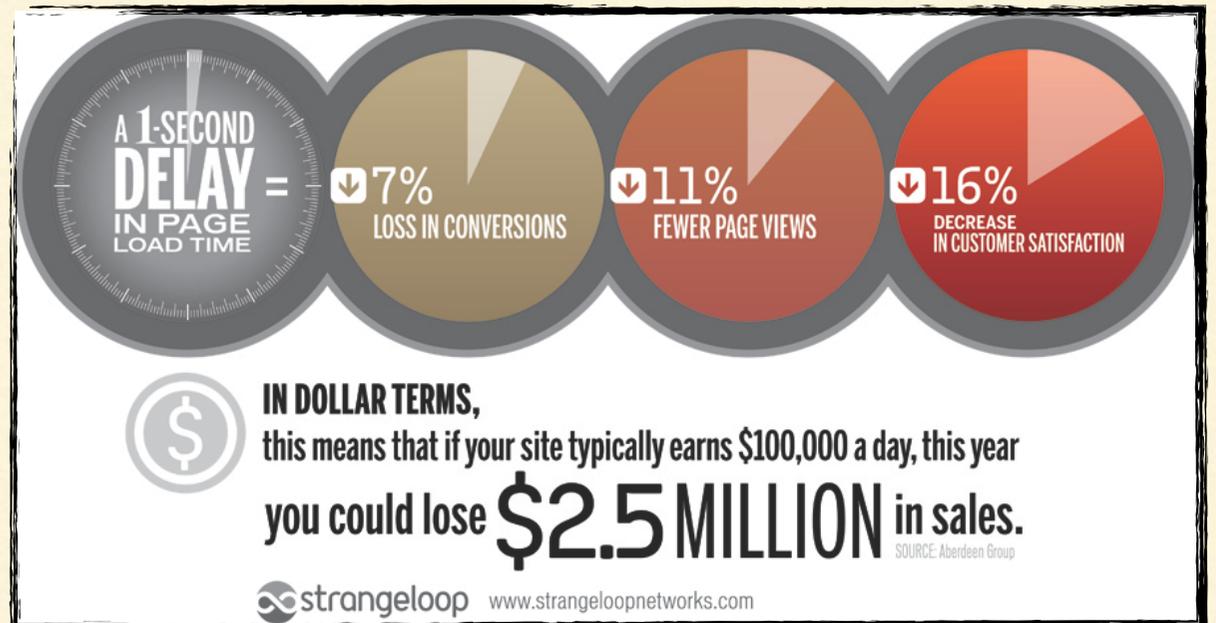
He wanted to thoroughly understand the basics first.

Understanding what makes a website load fast

A website is made up of images, Cascading style stylesheets (CSS), Javascripts, and HTML (the content). Every time a webpage is loaded, a request is made to the server to load all of the above elements. This is called a HTTP Request.

“57% of visitors will abandon a page that takes 3 seconds or more to load.”

The more elements a page has the longer it will take to load and slower the website will be. Another factor is the size of the individual elements. Jay could have just one image on the page but if that image is large it will slow down the entire page load. So, the ultimate goal of making the site fast



is to make the website more efficient and streamlined.

Tools to test performance of a site

Jay made a list of 3 popular tools to test the performance of the sites. These tool were awesome and even gave suggestions to improve a website’s performance.

[Pingdom Website Speed Test.com](https://www.pingdom.com/)

[GTmetrix](https://gtmetrix.com/)

[Google PageSpeed Insights](https://pagespeed.google.dev/)

Next, he started writing down the most important points to help spring clean and speed up a website so that his clients improve their conversion rate, reduce abandonment rate and grow their business.

Minimize HTTP Requests

“8% of people cite slow loading pages as a key reason for abandoning their purchase.”

Jay can order a cappuccino, blueberry muffin and a bottle of water at the same time or wait in the line and order each individual item one by one. Which do you think will be faster?

When a webpage is requested, 80% of the response time is spent in handling HTTP requests. It is important to reduce these requests by reducing the number of images, CSS and Javascript files on the webpage.

Combine Stylesheets

[CSS Delivery Tool](#) gives a list of the number of external stylesheets a website is using. This way, multiple stylesheets files can be combined into one file. Similarly, multiple Javascript files can be combined into one file - test the site to make sure nothing has broken after combining the files.

CSS Sprites

Each image loaded on a webpage is a HTTP request to the server. Instead, group all images in an image file (load only one image file) and then call each image using CSS. This is a great way to reduce web server requests if there are lot of images (banners, icons etc.) on the page. Once an image sprite is created use [Sprite Cow](#) to generate the CSS that reference those images in the stylesheet.

Compression

Compress website pages using [Gzip](#). What this means is: the web server will deliver compressed files over the Internet whenever someone makes a request for the webpage.

Jay thought of an interesting analogy - “Which is easier to throw – a tennis ball or a soccer ball?”

Optimize Images

“Aberdeen Group study showed that a one second delay in page load time equals 11% fewer page views, a 16% decrease in customer satisfaction, and 7% loss in conversions.”

All the images that are uploaded to the web server should be optimised first. As per [GTmetrix](#), “optimizing your image works by removing all the unnecessary data that is saved within the image to reduce the file size of the image based on where it is being used in your website.”

There are tools to get this done without compromising the visual quality of the image.

Tools: ImageOptim (Mac), CompressPNG, JPEGMini and some more on this [Mashable](#) article.

Jay was familiar with ImageOptim. Before uploading any image to his server he would drag it to ImageOptim first. ImageOptim would optimize the image and re-save it so that Jay could then take the optimized image and FTP to his server.

Removed Unused CSS

While designing, Jay’s team would experiment with multiple styles, line heights and layouts. Oftentimes, most of these unused CSS styles would never get purged from the final CSS file. Just like how cleaning cabinets/ drawers once in a while is a good exercise, a cleanup of the CSS code is always good after site completion. Less code the better.

Define width and height of Images

Jay never skipped this one but he did make a note for his team to check out.

Every time an image is referenced on the webpage (using the IMG tag) make sure the width and a height of the image is defined . This helps save time since the browser doesn’t have to do the work of calculating the dimensions.

“Speeding up websites is important — not just to site owners, but to all Internet users. Faster sites create happy users and we've seen in our internal studies that when a site responds slowly, visitors spend less time there. But faster sites don't just improve user experience; recent data shows that improving site speed also reduces operating costs. Like us, our users place a lot of value in speed — that's why we've decided to take site speed into account in our search rankings.”

- Google Blog

Minify Javascript & CSS

Just like optimizing images, minifying is the process of removing unneeded characters (such as tabs, spaces, source code comments, white space) from the source code to reduce file size.

[Minimus](#) is nice little minify utility for OS X. Just drag and drop Javascript files to have them minified.

Other tools: [JSMIn](#), [Closure Compiler](#), [YUI Compressor](#), [cssmin.js](#)

For example:

```
.some-style {  
    color: #000000;  
    font-size: 20px;  
}
```

Becomes...

```
.some-style{color:#000000;font-size:20px;}
```

Put stylesheets on the top of the page

Files responsible for the appearance of the webpage should be on the top of the page so that they can be applied to the HTML page as soon as the page loads. Website visitors should not wait until all the design on the page is loaded.

Put scripts at the bottom of the page

All scripts and functionality-related files should load after the site is loaded. Javascript located at the top of the page can block page rendering. There is also a way to defer parsing of Javascript by using the defer attribute. The defer at-

tribute is used in the HTML code to defer parsing of the javascript until the page is loaded. For example:

```
<script type="text/javascript"
src="includes/general.js" defer="defer"><-
/script>
```

Social Plugins

Don't clutter the homepage with social plugins (Twitter, Facebook, Pinterest etc.) unless they are absolutely needed to boast follower statistics. These buttons come with their own baggage and the more buttons on the webpage means more HTTP requests added to the webpage. Moreover, if either of these services become slow (or go down) it would impact the way the webpage would load.

Jay thought about an excited client who just loved to pepper the homepage with social plugins, stats and how many followers their site had. He chuckled to himself.

Use Content Delivery Networks

Jay thought for a moment - When he is at home and wants to order-in food, does he choose a restaurant that is far away or somewhere in the vicinity?

The thought of food made Jay hungry.

A CDN is a collection of servers that exist at different parts of the world to deliver content more efficiently. So, if Jay is located in Australia, the site images, videos, CSS and Javascript will be served to Jay from a server close to Australia. If Jay's colleague is in Norway, a CDN server close to his colleague's location will serve the files. CDNs speeds up the sending of files from the server.

Setting up a CDN is pretty simple and Jay had previously worked with a developer who used the Rackspace Cloud Files.

Popular CDNs: [Rackspace Cloud Files](#), [Amazon CloudFront](#), [Akamai](#), and [Cloudflare](#).

Browser Caching

Jay normally makes at least 4-5 visits to this coffee shop every week and almost every time orders a cappuccino. He imagined getting served the exact same thing every time without having to wait in line. Wouldn't that process be faster?

Browser caching stores cached versions of static resources like CSS, Javascript, images etc. This speeds up page speed tremendously and reduces server lag.

To enable caching, add the following code to the .htaccess file below. Those files have an expiration date in the header. This is known as "expires" headers. By default, that expiration date could be set to 1 month.

```
EXPIRES CACHING ##
```

```
<IfModule mod_expires.c>
```

```
ExpiresActive On
```

```
ExpiresByType image/jpg "access plus 1 month"
```

```
ExpiresByType image/gif "access plus 1 month"
```

```
ExpiresByType image/png "access plus 1 month"
```

```
ExpiresByType text/css "access plus 1 month"
```

```
ExpiresByType application/pdf "access plus 1 year"
```

```
ExpiresByType text/x-javascript "access plus  
1 month"
```

```
ExpiresByType application/x-shockwave-flash  
"access plus 1 year"
```

```
ExpiresByType image/x-icon "access plus 1  
year"
```

```
ExpiresDefault "access plus 1 month"
```

```
EXPIRES CACHING ##
```

Responsive Design - One website, many devices

[Wikipedia](#) defines Responsive Web Design - is a Web design approach aimed at crafting sites to provide an optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices (from mobile phones to desktop computer monitors).

Using responsive web design in web projects is a standard these days. Jay knows this and his two favourite responsive frameworks were Foundation and Bootstrap. These frameworks are small collections of CSS & Javascript files that can help him rapidly develop sites that look beautiful at any size, be it a 17" laptop screen or an iPhone. Responsive frameworks helps Jay keep his site code clean.

Responsive Frameworks - [Skeleton](#), [Foundation](#), [Bootstrap](#)

Jay is almost done with his list. He quickly jotted down a few other points that he thought could help.

A couple of other tips...

-
1. If the site is highly trafficked and uses a database then check out [Memcached](#).
 2. Avoid low-budget hosting providers.

It stopped raining and the sun was out. Jay wrapped up and made his way back to the office. He was excited to share the list with his team.

From today, Jay's designers (even the new hires) will be keeping speed as top priority whenever designing websites. His checklist will help each of his designers become experts at speeding up their client's websites. Whenever they add a new piece of code, social media button, or a graphic, they are always going to ask the question- "is this going to impact page load speed?"

Workflow in Brightpod

Jay adds this workflow to Brightpod and was kind enough to let us share it with everyone else.

If you have a Brightpod account you can apply this workflow when adding a new project.

References

-
1. (Infographic) [How loading time affects your bottom line.](#)
 2. [Performance Research, Part 1: What the 80/20 Rule Tells Us about Reducing HTTP Requests.](#)
 3. [11 Low-Hanging Fruits for Increasing Website Speed.](#)
 4. [Performance Unleashed: How To Speed Up WordPress Load Times](#)
 5. [10 Ways to Speed Up Your Website – and Improve Conversion by 7%](#)
 6. [How to speed up your website load times](#)
 7. [10 Ways to Improve Your Web Page Performance](#)
 8. [Visualizing Web Performance](#)
 9. [How Load Time Affects Google Rankings](#)
 10. [Using site speed in web search ranking](#)



[@brightpodapp](#) // www.brightpod.com