**7 ways to boost your Google insights mobile responsiveness score**

With mobiles being enhanced with new features on a regular basis, the popularity of their usage is only rising. Statistics reveal that mobile usage has exceeded the use of desktop by a huge margin. And, this popularity is only expected to grow with time. Hence, the user experience of a website on a mobile is important. One of the basic needs to satisfy users who browse your website on their mobiles is its loading speed. If a website is slow or an app is clumsy, users will drift away to another site in no time to get the information they require.

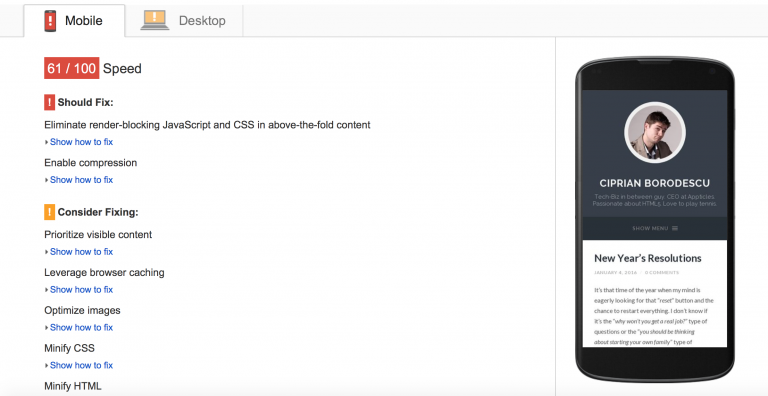
Before you work on improving the loading speed of your website, you will need to estimate its current speed. The loading speed can be estimated using Google PageSpeed Insights.

Google PageSpeed Insights

Google PageSpeed Insights (PSI) diagnoses your website’s mobile and desktop versions and provides you indications about the problem areas that need to be addressed.

Although PageSpeed Insights is a tool for developers, I’ll explain it in a non-technical language so that you will understand where and why optimization of mobile presence is essential.

Simply add the address of your website on PageSpeed Insights. You will be provided a complete report that looks like the one below:



**These are the 7 steps devised by Mobile 1st optimization team to accelerate your mobile page loading speed:**

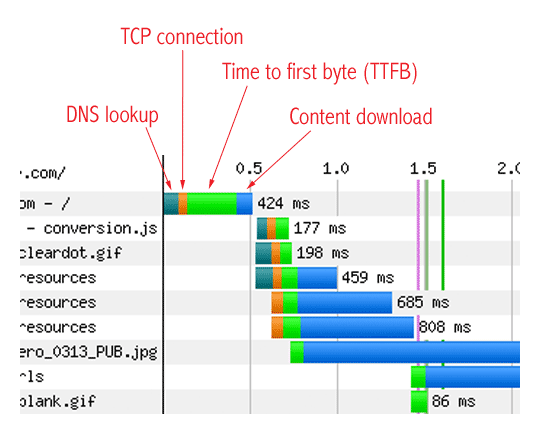
## 1. Measure and minimize server response time

Although the speed of loading of a page on a mobile is determined by web coding but your server also influences it. The longer the waiting time of a server to respond to a request from the browser, lower the loading speed.

**Server response time is influenced by the following factors:**

* Website Traffic - More traffic, more problems.
* Website Resource Usage - If each of your webpages reduce the resources used, the server response time would improve without costing anything.
* Web Server Software - By changing web server software or its configuration, server response time could be enhanced without any expense.
* Web Hosting. Server response time can also be increased by improving the quality and scope of your web hosting. However, this involves expenditure.

How to enhance server response time?

* You can improve your web server software or configuration.
* Enhance the quality and scope of your web hosting service, particularly ensure adequate CPU and memory resources.
* Reduce the resources required by your web pages

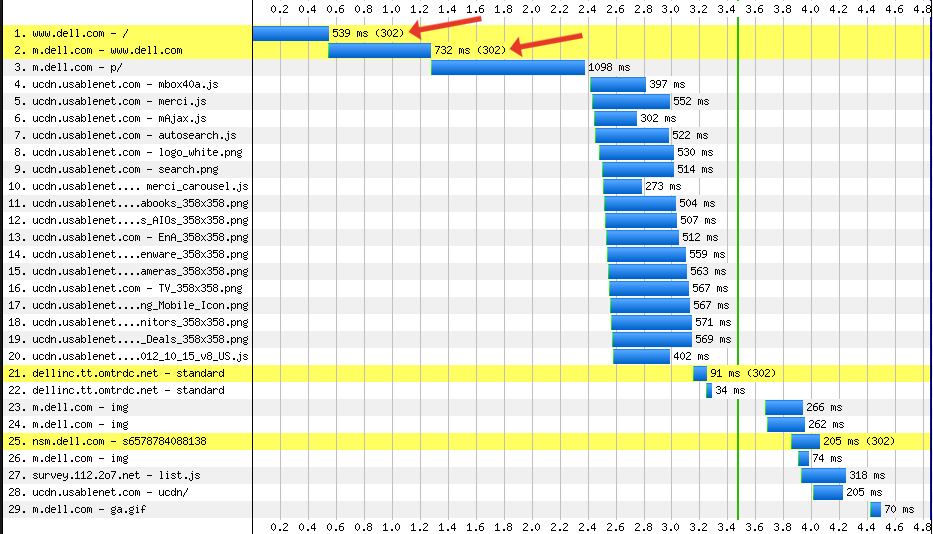
**2. Avoid or minimize redirects to accelerate mobile page speed**

A redirect diverts the visitors of one file to another file or location. Redirecting a file is done through a set of instructions in various ways. However, irrespective of the way in which it is done, the page speed is adversely affected.

The most commonly used redirects are 301 and 302.These use HTTP to explain that a page or resource has moved. Both of these are server-side redirects. This implies that the web server is making use of HTTP for directing the browser to a new location of the file. 301 is a permanent redirect while 302 is a temporary one.

**How to Minimize Redirects?**

* Using device-specific versions of your site: a mobile user navigating tohttp://example.com might be redirected to http://m.example.com/home);
* Hyperlinking inconsistently on your site: instead of consistently using URLs with the “www” prefix (such as http://www.example.com) or URLs without the “”www” prefix (such as http://example.com), you mix and match the two, leading to unnecessary redirects
* The number of elements on your page should be streamlined
* CSS should be used instead of images whenever possible.
* Multiple style sheets should be combined into one.
* Reduce scripts and put them at the bottom of the page.



**3. Eliminate render-blocking JavaScript and CSS in above-the-fold content**

Before a page is rendered to a user by the browser, the browser parses the page. During parsing, if it comes across a non-async or blocking external script, the download of the resource will be stopped. Each time this occurs, it gives rise to additional network round trip. This results in delaying the time to first render of the page.

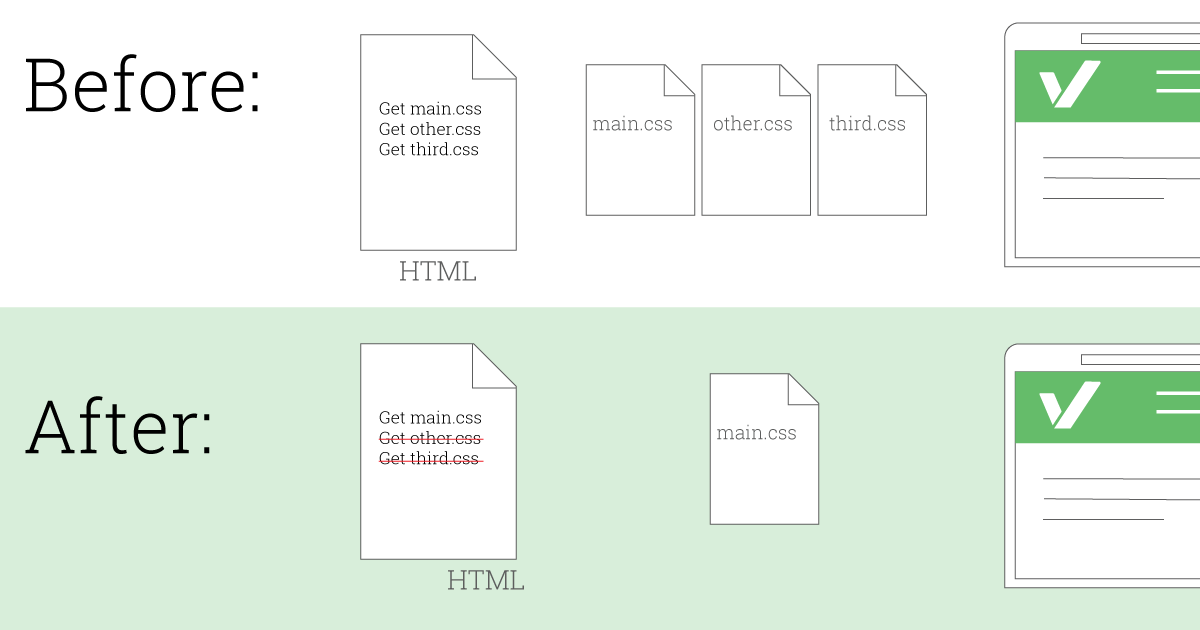
How to eliminate render blocking CSS?

1. Properly call your CSS files.

Here are two things to look for in the way you call your css.

* Do not use @import to call css files
* Properly label conditional css

1. Lessen the amount of CSS files in the critical path.
2. Use less CSS overall, Combine CSS files.



Methods for eliminating render-blocking JavaScript

1. Inline JavaScript

Rendering of a page is delayed by external blocking scripts. This is because these scripts force the browser to wait while the JavaScript is fetched adding one or more network roundtrips. Network request latency can be avoided through small external scripts. These facilitate inlining contents directly into the HTML document.

This is explained with the help of the example below:



1. Make JavaScript Asynchronous

JavaScript blocks DOM construction leading to delays in the time to first render. JavaScript can be prevented from blocking the parser by using the HTML async attribute on external scripts. For example:

<script async src="my.js">

1. Defer loading of JavaScript.

Performance can be enhanced and resource contention reduced by deferring loading and execution of scripts that are not essential for the initial page render. They can be deferred until after the initial render or other critical parts have loaded.

**4. Optimize images**

Images created in Photoshop and illustrator appear wonderful. However, their file sizes are too large. This is because these images are created in a format simplifying their manipulation in different ways. If you include images of large file sizes in your website, its loading speed will reduce.

How to solve it

* Avoid using images that are too large compared to how they are displayed. For instance, don’t use an image of 3,000 x 3,000 pixels if the user will only view it at a 500 x 500 size.
* Lowering the quality of an image reduces its size. Therefore, it is recommended to reduce the quality of images without making them look bad.
* Consider using responsive images.

**5. Leverage Browser Caching**

While displaying your web page, a web browser needs to load your logo, CSS file, and other resources.

**What is browser caching?**

Browser caching is when the browser “remembers” the resources it has already loaded. When browser caching is done and the visitor goes to another page on your website, the logo, CSS files, etc. are not loaded again. The reason is that the browser has saved them. Therefore, viewing a web page for the first time takes longer than repeated visits.

If you need to leverage browser caching to enhance your loading speed, you can do it in the following manner:

**How to solve it**

In case you are using Apache as your web server, browser caching can be implemented quickly by adding a few lines of code in a file called .htaccess. You can check with your hosting company to know whether you are running on Apache or not.

ExpiresActive On  
ExpiresByType image/jpg "access plus 1 month"  
ExpiresByType image/jpeg "access plus 1 month"  
ExpiresByType image/gif "access plus 1 month"  
ExpiresByType image/png "access plus 1 month"  
ExpiresByType text/css "access plus 1 week"  
ExpiresByType application/pdf "access plus 1 month"  
ExpiresByType text/x-javascript "access plus 1 week"  
ExpiresByType application/x-javascript "access plus 1 week"  
ExpiresByType application/javascript "access plus 1 week"  
ExpiresByType text/javascript "access plus 1 week"  
ExpiresByType image/x-icon "access plus 1 year"  
ExpiresDefault "access plus 2 days"

**6. Enable Compression**

In all modern browsers gzip compression for all HTTP requests is supported and automatically negotiated. Enabling gzip compression reduces the size of transferred response by approximately 90%. This drastically reduces the time required to download the resource. This diminishes the data usage for the client leading to improved first render of your page.

**How to solve it**

Like browser caching, file compression can be implemented by adding a few lines of code in your .htaccess file:  
  
<ifModule mod\_deflate.c>  
<filesMatch ".(css|js|x?html?)$">  
SetOutputFilter DEFLATE  
</filesMatch>  
</ifModule>

**7. Make Use of Accelerated Mobile Pages**

Accelerated Mobile Pages was introduced by Google for a faster and open mobile web. Accelerated Mobile Pages are specially formatted web pages that facilitate display of content at a fast pace by search engines. At the same time, they make sure that the publishers have complete control over the way their content looks and feels.

**How to Implement AMP on Your Site?**

**A- Create a separate version of your main page by using AMP HTML** (also known as the AMP version of your main page). Basic code for designing your first AMP HTML page is provided below:

* *<!doctype html>*
* ***<html*** *amp lang="en"****>***
* ***<head>***
* ***<meta*** *charset="utf-8"****>***
* ***<title>****The title of your AMP page will go here* ***</title>***
* ***<link*** *rel="canonical" href="URL of the non AMP version will go here"* ***/>***
* ***<meta*** *name="viewport" content="width=device-width,minimum-scale=1,initial-scale=1"****>***
* ***<script*** *type="application/ld+json"****>***
* *{*
* *"@context": "http://schema.org",*
* *"@type": "NewsArticle",*
* *"headline": "Open-source framework for publishing content",*
* *"datePublished": "2015-10-07T12:02:41Z",*
* *"image":*

*[*

* *"logo.jpg"*
* *]*
* *}*
* ***</script>***
* ***<style*** *amp-boilerplate****>body****{-webkit-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-moz-animation:-amp-start 8s steps(1,end) 0s 1 normal both;-ms-animation:-amp-start 8s steps(1,end) 0s 1 normal both;animation:-amp-start 8s steps(1,end) 0s 1 normal both}@-webkit-keyframes -amp-start{****from****{visibility:hidden}****to****{visibility:visible}}@-moz-keyframes -amp-start{****from****{visibility:hidden}****to****{visibility:visible}}@-ms-keyframes -amp-start{****from****{visibility:hidden}****to****{visibility:visible}}@-o-keyframes -amp-start{****from****{visibility:hidden}****to****{visibility:visible}}@keyframes -amp-start{****from****{visibility:hidden}****to****{visibility:visible}}</****style****><****noscript****><****style******amp-boilerplate****>****body****{-webkit-animation:none;-moz-animation:none;-ms-animation:none;animation:none}****</style></noscript>***
* ***<script*** *async src="https://cdn.ampproject.org/v0.js"****></script>***
* ***</head>***
* ***<body>***
* ***<h1>****Welcome to AMP****</h1>***
* ***</body>***
* ***</html>***

**B- In order to insert images, use the following code:**

***<amp-img*** *src="welcome.jpg" alt="Welcome" height="400" width="800"****></amp-img>***

**C- Modify the presentation and layout of the page by using common CSS properties**. An example styling code is provided below:

***<style*** *amp-custom****>*** */\* any custom style goes here \*/* ***body*** *{ background-color: white; }* ***amp-img*** *{ background-color: gray; border: 1px solid black; }****</style>***

**D- Preview, validate and publish the AMP page.**

**E- Add the amphtmltag in the non-AMP version of the page** in order to help search engine bots to identify the AMP version of your webpage:

*<link rel="amphtml" href="AMP version of your webpage goes here" />*

Conclusion:

Although not exhaustive, these are some useful guidelines that can be followed to enhance the loading speed of a website on a mobile device. Enhance your Google insights mobile responsiveness score to have an edge over your competitors.