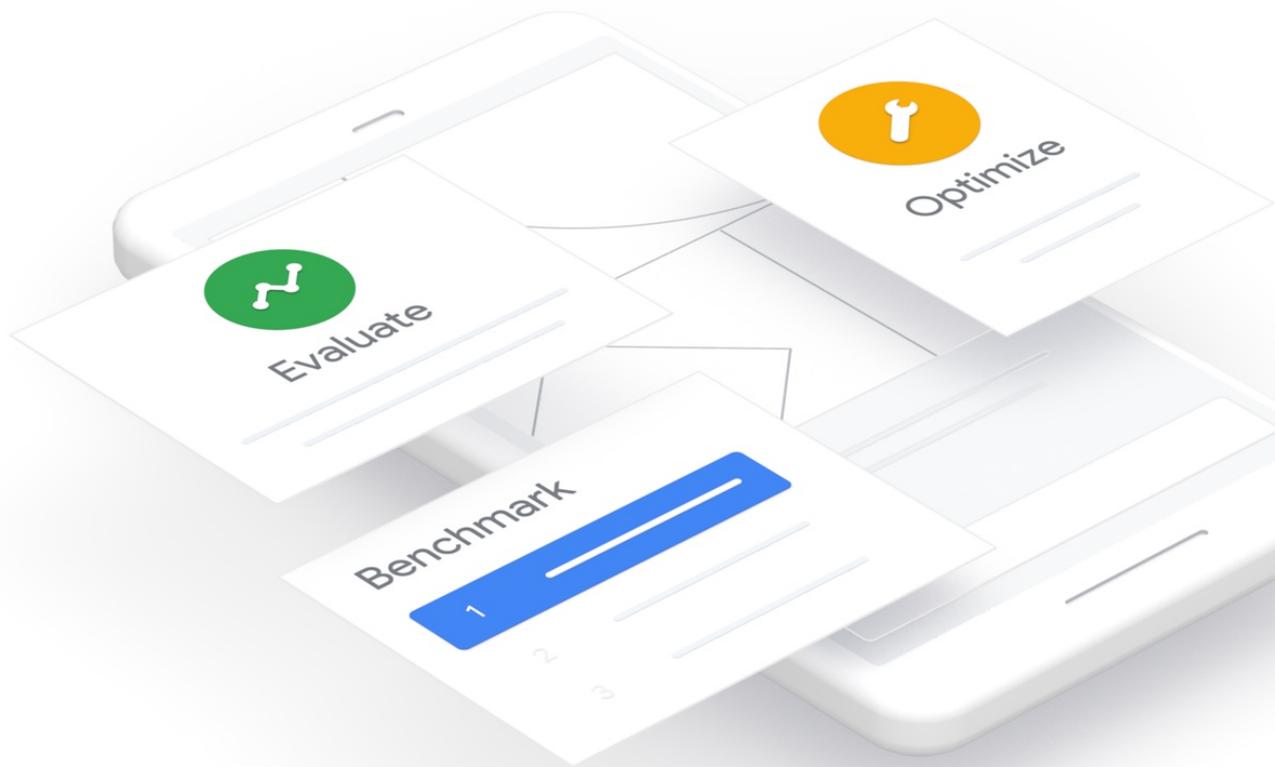


Test My Site

Your mobile speed report for rennes.envie.org



Test My Site Report

A slow mobile site can impact customer satisfaction and your revenue. This report assesses the mobile performance of rennes.envie.org and provides custom fixes to help you improve the speed of mobile pages across your site.

This report will help you answer the following questions

-  How does your site speed compare to your competitors' site speeds?
-  How does your site speed impact your potential revenue?
-  How can you optimise the speed of pages on your site?

Mobile Speed Overview

Site speed is the speed of all the pages on your site. Your site speed in France on 4G networks is:

4.0 seconds

RATING

Slow

Fast sites start to load in under 1 second. Average sites start to load in 1 to 2.5 seconds. Slow sites start to load in over 2.5 seconds.

TREND

Speeding Up

Your site has gotten 1.1 seconds faster this month. It's important to monitor site speed regularly to prevent your site from becoming slower over time.



How does your site speed compare to your competitors' site speeds?

*Looks like you haven't completed this step. Return to [Test My Site](#) to see how your site speed compares to your competitors.



How does your site speed impact your potential revenue?

*Looks like you haven't completed this step. Return to [Test My Site](#) to evaluate the impact a faster site can have on your revenue.



How can you optimise the speed of pages on your site?

Page speed is the speed of an individual page on your site; site speed measures the speed of all of the pages on your site. To make your overall site faster, improve the speed of individual pages using the recommended optimisations.

Below is a summary of the page speed, rating, and recommended fixes for the individual pages tested. Learn more about each recommended fix at the end of this report.

PAGE URL	PAGE SPEED	RATING
http://rennes.envie.org/	6 seconds	Slow
RECOMMENDED FIXES		
<input type="checkbox"/> Ensure text remains visible during webfont load	<input type="checkbox"/> Serve images in next-gen formats	
<input type="checkbox"/> Reduce JavaScript execution time	<input type="checkbox"/> Avoid an excessive DOM size	
<input type="checkbox"/> Defer offscreen images	<input type="checkbox"/> Minify JavaScript	
<input type="checkbox"/> Properly size images	<input type="checkbox"/> Eliminate render-blocking resources	
<input type="checkbox"/> Defer unused CSS	<input type="checkbox"/> Efficiently encode images	
<input type="checkbox"/> Avoid enormous network payloads	<input type="checkbox"/> Serve static assets with an efficient cache policy	

Next Steps

Now that you have tested your site speed, take action to make your site faster with these steps.

1. Share this report with your technology or IT department

Meet with your technology or IT team to share these results and make improving your mobile site speed a top priority.

2. Create an optimisation plan

Incorporate the recommended fixes into your product roadmap.

3. Monitor your site speed monthly

Check [Test my Site](#) monthly to monitor your mobile speed and ensure your site speed doesn't slow down over time.

4. Check out Google's other mobile resources

For marketers, find the latest trends and mobile research on [Think with Google](#). For developers, stay up to date on your mobile performance by checking [PageSpeed Insights](#).

Fixes Glossary

Learn more about your recommended fixes. For further information on how to improve site and page speed direct your developer(s) to PageSpeed Insights.

[AMP your page](#)

Leverage AMP's open-source library to create mobile web pages that load near instantaneously for users.

[Eliminate render-blocking resources](#)

Resources for page styling and functionality are delaying visibility of primary content on the page. Deliver critical JS/CSS inline and defer all non-critical JS/styles.

[Properly size images](#)

Your page shouldn't serve images that are larger than the version rendered on the user's screen. Resize images to match display dimensions to speed up and improve page load time.

[Defer offscreen images](#)

Since users can't see offscreen images when a page loads, those images don't need to be loaded initially. Use offscreen images to speed up the page load time.

[Minify CSS](#)

Minification strips a code file of all data that isn't required in order for the file to be executed. This results in faster response times and lower bandwidth costs.

[Minify JavaScript](#)

Minification strips a code file of all data that isn't required in order for the file to be executed. This results in faster response times and lower bandwidth costs.

Efficiently encode images

The less competition there is for your bandwidth, the faster your browser can download and render content on your screen. Optimising images improves performance for your website.

Serve images in next-gen formats

Encoding your images in JPEG 2000, JPEG XR, and WebP will load faster and consume less cellular data. Serve a fallback PNG or JPEG image for other browser support.

Enable text compression

Text compression minimises byte size, which leads to faster page loads. Enable text compression on your server to minimize total network bytes.

Preconnect to required origins

Establishing connections often involves significant time on slow networks. Preconnect to required origins to make your application feel faster without negatively affecting bandwidth usage.

Resolve slow server response times

Users dislike it when pages take a long time to load. Slow server response times are one possible cause for long page loads.

Avoid multiple page redirects

Redirects introduce additional delays before the page can be loaded. Remove unnecessary redirects and replace with direct links.

Preload key requests

Preloading requests can make your pages load faster. Declare preload links in your HTML to instruct the browser to download key resources as soon as possible.

Use video formats for animated content

Use MPEG4/WebM videos for animations and PNG/WebP for static images instead of GIF to save network bytes.

Avoid enormous network payloads

Reduce the total size of network requests to speed up page load time.

Serve static assets with an efficient cache policy

HTTP caching can speed up your page load time for returning users. Increase your cache lifetime to speed up repeat visits to your page.

Avoid an excessive DOM size

A large DOM tree can harm your page performance. Look for ways to create DOM nodes only when needed, and destroy them when no longer needed.

Minimise critical requests depth

Minimise the number and download size of critical resources.

Instrument your scripts with User Timing marks and measures

Consider instrumenting parts of your scripts with the User Timing API to create real-world measurements of key user experiences.

Reduce JavaScript execution time

Inefficient javascript can slow down your page. Reduce the time spent parsing, compiling, and executing your code.

Ensure text remains visible during webfont load

Leverage the font-display CSS feature to ensure text is user-visible while webfonts are loading.

Defer unused CSS

Style sheets containing unused CSS can significantly increase the time users wait before content appears on their screen. Delete unnecessary CSS.