



WPSpeed – The professional solution to boost your website performance

Introduction

Google rolled out "Core Web Vitals", indeed the speed factor is today crucial for the ranking of a website.

The Google PageSpeed score assigned to your pages is now really important, a score of 90 or above is considered fast, and 50 to 90 is considered moderate. Below 50 is considered to be slow.

The WPSpeed plugin for Wordpress increases performance of your website by applying automatically several improvements as suggested by Google PageSpeed best practices. By installing the plugin you can reach up to a doubled score during the Google PageSpeed test. WPSpeed is the most powerful Wordpress performance plugin that implements unique technologies such as Adaptive Contents™ that makes it possible to generate a lightweight version of the website to dramatically raise up the page speed score without losing website functionalities.

This score is determined by running “Lighthouse” to collect and analyze lab data about the page. A score of 90 or above is considered fast, and 50 to 90 is considered moderate. Below 50 is considered to be slow.

WPSpeed increases performance of your website by applying automatically several improvements as suggested by Google PageSpeed best practices.

Main features of WPSpeed are:

- Scripts minification – Javascript can be minified to reduce the total scripts size and save bandwidth
- Stylesheets minification – CSS can be minified to reduce the total CSS size and save bandwidth
- HTML minification – the HTML source code can be minified to reduce the total page size and save bandwidth
- Combine multiple JS and CSS files– Javascript and CSS can be combined in single files to minimize the HTTP request overhead
- Lazy-load – Images, iframes and HTML elements that are out of view can be lazy-loaded only when the user scrolls down the page to display them
- Images optimization – Large images can be rescaled, resized and optimized on the fly to dramatically reduce the page size and bandwidth
- Htaccess optimization – A series of optimizations can be applied to the htaccess file in order to leverage the browser caching
- Combine images – It's possible to generate a single container image for multiple backgrounds applied through CSS
- HTTP/2 optimizations – WPSpeed works with HTTP/2 compatible servers to enable server push
- CDN support – Links for all assets are automatically rewritten to point to your CDN domain

Main configuration

After installing the WPSpeed plugin on your Wordpress website, you can open plugin settings and start to enable the various features and optimizations.

Combine assets

The basic optimizations that should always stay enabled, are the 'Combine Assets' and 'Minify Assets' ones. Thanks to these features, multiple Javascript and CSS files can be minified and combined into single files merging the contents of different assets into a unique resource.

This means that HTTP requests will be reduced to essential ones.

The screenshot displays the WPSpeed plugin configuration interface. It is divided into two main sections: 'Combine Assets' and 'Minify Assets'. In the 'Combine Assets' section, three settings are shown, all of which are enabled (indicated by a blue checkmark in a circle): 'Enable combining of CSS and Javascript files', 'Combine CSS files', and 'Combine Javascript files'. Each setting includes a brief description of its function. A callout bubble points to these settings, stating: 'Both Javascript and CSS files are combined into aggregated ones'. The 'Minify Assets' section is also shown, with three settings enabled: 'Minify CSS', 'Minify javascript', and 'Minify HTML'. A second callout bubble points to these settings, stating: 'Assets are minified to reduce their size'.

Once activated the combine and minify features, it's also possible to manage advanced options such as:

- The cache lifetime for combined files
- The possibility to 'defer' styles and scripts to further improve the performance
- The possibility to load the whole Javascript asynchronously to further improve the performance
- The possibility to improve Google fonts loading

Minify assets and HTML

Javascript and CSS code can be minified stripping out optional comments, spaces, tabulations, etc.

Furthermore, also the HTML source code of the page can be minified to dramatically reduce the total page size that is directly downloaded by a browser.

To activate the HTML minification it's enough to turn on the following settings:

Minify Assets
CSS, Javascript and HTML code can be minified and optimized.

- Minify CSS**
Whitespaces and comments will be removed from the combined CSS file, this helps to reduce the file size, save bandwidth and improve the loading time. NOTICE: this applies only to resulting combined CSS files and not to original files, thus if the 'combine' feature is not enabled this setting won't have any effect.
- Minify Javascript**
Whitespaces and comments will be removed from the combined JS file, this helps to reduce the file size, save bandwidth and improve the loading time. NOTICE: this applies only to resulting combined JS files and not to original files, thus if the 'combine' feature is not enabled this setting won't have any effect.
- Minify HTML**
Whitespaces and comments will be removed from the HTML source code, this helps to save bandwidth and improve the loading time.

HTML Minification level
High

Choose the level of minification for HTML. Using the 'Low' level, multiple whitespaces outside of elements will be reduced to a single one. Using the 'Normal' level comments and all not needed whitespaces or carriage returns will be removed. Using the 'High' level even quotes or double quotes from around attributes will be removed. CAUTION: using the 'High' level could cause issues with third-party plugins; if you experience this kind of problems switch back to 'Normal'.

Enable the minification of the HTML code and choose the minification level

Pay attention to the level of minification reserved to the HTML source code. Indeed for the HTML there are 3 different stages and different types of compression that in certain cases could cause undesired effects or conflicts.

For this reason you can progressively increase the level choosing:

- Low level - multiple whitespaces outside of elements will be reduced to a single one
- Normal level - comments and all not needed whitespaces or carriage returns will be removed
- High level – comments, all not needed whitespaces or carriage returns, quotes or double quotes from around attributes will be removed

If you experience issues due to a too much high level of compression, it's recommended to reduce it to a lower level.

Obviously the higher the compression level the smaller the total page size will be. So from a performance perspective it's better to reach the highest possible level while keeping the website stable and working well with all various installed plugins.

Inclusions and exclusions of assets

When the 'Combine assets' feature is enabled, it's not uncommon to experience issues, errors or conflicts especially about the Javascript loaded on pages.

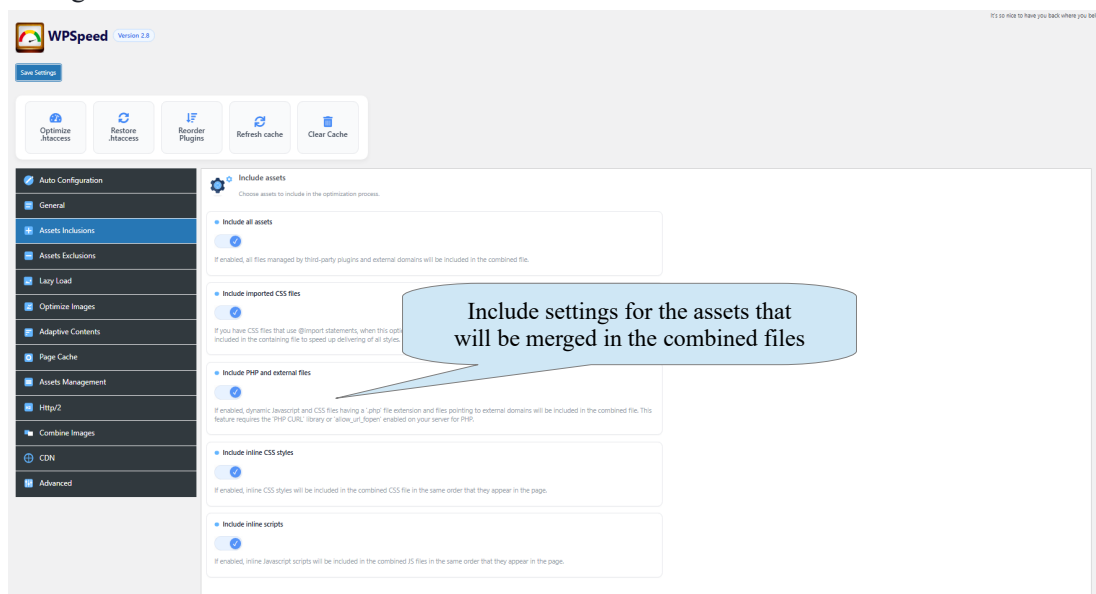
In most cases this is due to the change in the order of execution of scripts included by various plugins that have been installed on the Wordpress website.

To handle and solve this kind of problems, WPSpeed has a powerful system to manage inclusions and exclusions of assets that should not be included in the combined file.

When you choose to exclude a certain asset from the combined file, it will be extracted and listed in the page source code normally, thus preserving the correct order of execution applied when the browser is going to parse the HTML code.

First of all you can choose which resources should be included in the combined files using the following settings:

- Include all assets – If this option is not enabled, only files that are part of the Wordpress core will be included
- Include imported CSS files – In the case that CSS files with an @import directive are detected, the target CSS file will be fetched and merged
- Include PHP and external resource files – In the case that there are dynamic CSS or JS files generated by the server through PHP, they will be fetched and merged
- Include inline CSS styles – All inline stylesheets directly embedded in the HTML source code will be merged
- Include inline scripts - All inline scripts directly embedded in the HTML source code will be merged



Once you have chosen which assets must be involved in the combine process, you can go on and exclude particular resources choosing them one by one or simply entering the name or substring.

It's possible to exclude:

- Single CSS files
- All CSS files for a given plugin
- Inline CSS styles
- Single Javascript files maintaining the original order of execution or not maintaining it
- All Javascript files for a given plugin maintaining the original order of execution or not maintaining it
- Inline scripts
- Certain pages or URLs

There are even more advanced settings to apply global exclusions or to improve the loading of files that have been excluded in order to preserve a performance improvement.

Files that are loaded in 'defer' mode (deferred) won't impact the page loading time, indeed they are only executed once the HTML document has been fully parsed.

The image shows a screenshot of the WordPress settings interface, specifically the 'Exclude CSS' and 'Exclude JS With Order' sections. The left sidebar shows the 'Settings' menu with 'Advanced' selected. The main content area is divided into two sections: 'Exclude CSS' and 'Exclude JS With Order'. Each section has several sub-sections with callouts explaining their functions.

Exclude CSS

- Exclude CSS files:** A callout points to the list of files, stating "Exclude certain CSS files".
- Exclude CSS files by plugin:** A callout points to the 'gtranslate' plugin, stating "Exclude all CSS file for a given plugin".
- Exclude inline 'style' declarations:** A callout points to the 'img.wp-s-miley, img.emoji[dis play:inline]' declaration, stating "Exclude specific inline CSS styles".
- Exclude all inline 'style' declarations:** A callout points to the toggle switch, stating "Exclude all inline 'style' declarations".

Exclude JS With Order

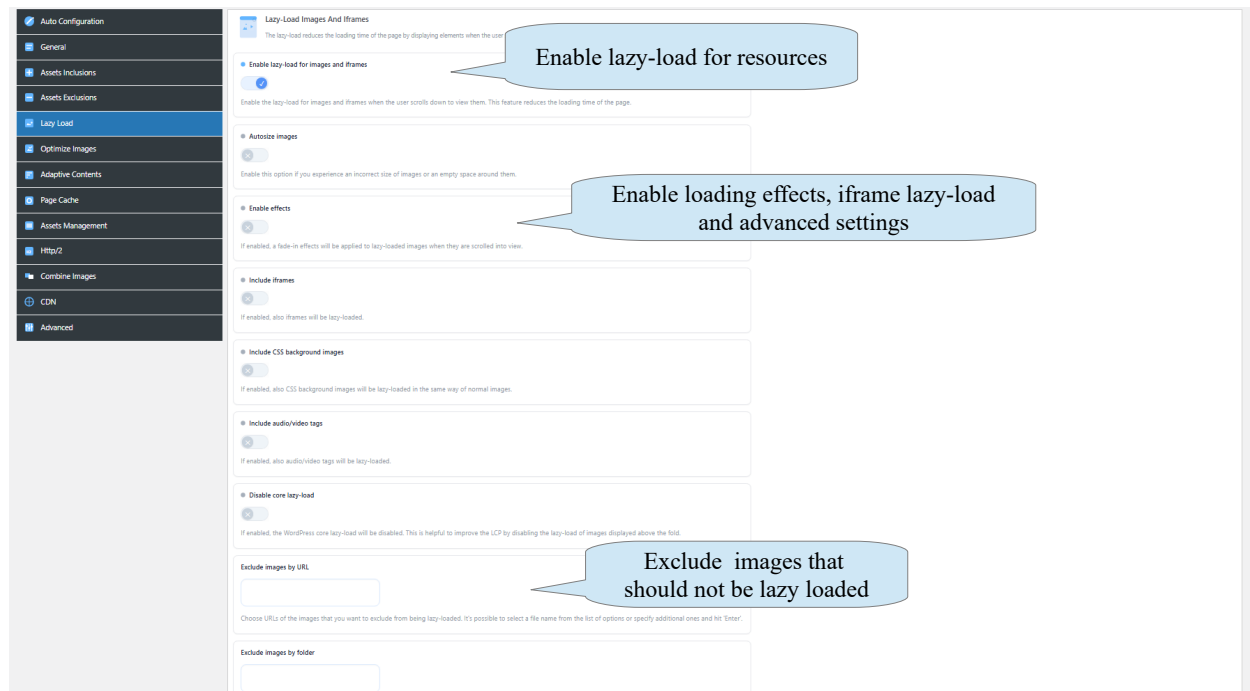
- Exclude Javascript files:** A callout points to the 'loader.min.js' file, stating "Exclude specific JS files for a given plugin while preserving the original order of execution".
- Exclude Javascript files by plugin:** A callout points to the 'optimize-v2.b-cdn.net' plugin, stating "Exclude all JS files for a given plugin while preserving the original order of execution".
- Exclude inline 'script' declarations:** A callout points to the 'var wpScriptRegistry={l&qu otid&qu' declaration, stating "Exclude specific inline scripts".
- Exclude all inline 'script' declarations:** A callout points to the toggle switch, stating "Exclude all inline 'script' declarations".

Lazy-load

WPSpeed is capable to lazy-load images, iframes, video/audio tags that are out of view and also HTML elements that are especially heavy, for example for the LCP (Largest Contentful Paint). To lazy-load HTML elements it's required to use common CSS selectors to identify target tags.

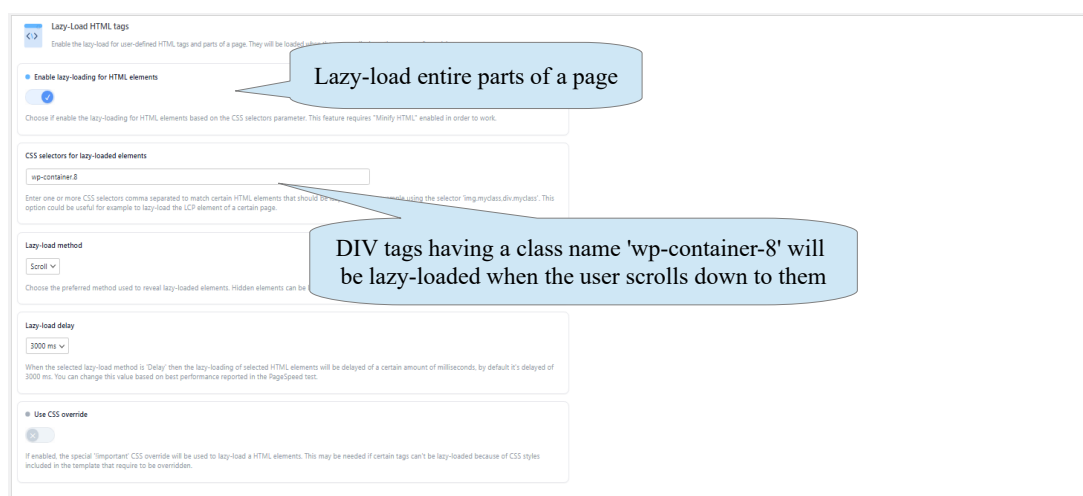
Indeed there is no need to fetch all resources at the beginning of the page load if they are not immediately visible, they can be lazy-loaded only when the user scrolls down the page to display them.

This feature dramatically reduces the loading time of the page and the usage of the bandwidth. You can enable lazy-loading and choose preferred settings for example to exclude images or apply a loading effect:



WPSpeed takes the lazy-load method to the next level making it possible to defer the loading of entire parts of a page.

Thanks to a powerful system based on CSS selectors you can choose to load certain elements such as the footer, only when the user scroll down to reach them or after a prefixed time delay.



Images optimization

WPSpeed has a powerful system to optimize heavy images on the fly that will be compressed and cached to reduce the bandwidth usage and speed up your site.

Pay attention that as a consequence of the optimization, the quality of images may be significantly decreased on large screens, so you have to properly evaluate which settings adapt better to the contents and type of visitors of your website.

It's possible to choose various aspects such as:

- The minimum width of an image to apply the optimization for
- The conversion to the new format WebP, if supported by your server, and settings for the quality for compressed images
- The resizing of optimized images
- The minimum width for an image to be resized
- The automatic generation of a srcset
- The optimization of images loaded through CSS and HTML background
- Exclusions for certain images that should not be altered in any way

Optimize Heavy Images
The images optimization allows to reduce the size of heavy images that are not needed on small screens.

Enable image optimization
If images optimization is enabled, images will be compressed and cached to reduce the bandwidth usage and speed up your site.

Minimum width for image optimization
50
Choose the minimum width that images must have to be optimized. You should ignore small images that are already lightweight and don't need any kind of optimization.

Quality for compressed images
70%
You can choose the image quality for resulting optimized images. High value will mean a higher quality but also higher bandwidth usage.

Enable image resizing
You can choose if images should be resized to a lower resolution on small screens.

Resizing percentage
80%
Choose the percentage to which you want to down-size your images. For example if you have an image 1500px wide and apply a resizing percentage of 80%, the image will be resized to 1200px thus saving bandwidth.

Minimum width for resizing images
300
Choose the minimum width that images must have to be resized to a lower resolution. You should avoid to resize small images that are already lightweight and apply down-sizing only to large and heavyweight images.

Convert all images to WebP
✓ WebP extension supported
If enabled, all optimized images will be converted to the next generation WebP format. NOTICE: for this function to work the server PHP must include the GD library with the WebP support enabled and the 'imagecreate' function must be available.

Convert all images to AVIF
✓ AVIF extension supported
If enabled, all optimized images will be converted to the next generation AVIF format. NOTICE: for this function to work the server PHP must include the GD library with the AVIF support enabled and the 'imagecreate' function must be available. If this is missing on your server you can contact the hosting provider. Commonly this feature is supported by PHP 8.1 and later.

Srcset management
This settings make it possible to automatically generate a srcset starting from the original image. When this feature is enabled, it will replace the native Wordpress 'srcset'.

Create srcset
If enabled, an additional HTML5 'srcset' attribute will be added for each image that need to be optimized and WPSpeed will automatically create up to 4 differently-sized images as variations to the base image. The browser will select the best image based on the device screen size and resolution. IMPORTANT: if you activate this feature, images may be enlarged to full width of the container element. If you want to avoid this, you can set a maximum width for the srcset images.

Initial quality value
90%
Choose the value of the initial quality used for the image with the highest resolution of the srcset and for the regular fallback img tag. Clear the plugin cache after that you have changed this setting.

Quality decrease
15%
Choose the amount of quality decrease that each srcset image should have. Starting from the highest quality image there will be 3 more images with a lower quality reduced by this value. Clear the plugin cache after that you have changed this setting.

Initial resizing value
100%
Choose the value of the initial resizing used for the image with the largest size of the srcset and for the regular fallback img tag. Clear the plugin cache after that you have changed this setting.

Resizing decrease
20%
Choose the amount of resizing decrease that each srcset image should have. Starting from the biggest image there will be 3 more smaller images reduced by this value. Clear the plugin cache after that you have changed this setting.

Standard image 'src'
4x
You can choose which image to use for the standard 'src' attribute used as a fallback for browsers that don't support srcset. Choosing a lower resolution image for the 'src' attribute could be helpful to reach a higher score during the PageSpeed test.

Adaptive Contents

The Adaptive Contents feature makes it possible to remove certain parts of HTML or CSS/JS files to generate a lightweight version of the website.

This dramatically raises the page speed score without losing website functionalities for regular visitors.

Thanks to this feature when the page is crawled by bots and page speed tools such as the Google PageSpeed Insights, Lighthouse or GTMetrix it's possible to:

1. Remove certain HTML elements from the page, for example to reduce the CLS value
2. Remove certain JS files
3. Remove certain CSS files
4. Remove all JS and CSS code

This is especially helpful to improve the following PageSpeed metrics:

- Minimize main-thread work
- Reduce Javascript execution time
- Avoid enormous network payloads

If necessary it's also possible to define a custom list of bots user-agent strings to match various page speed tools.

This feature will not affect the normal version of the website that is loaded by visitors, indeed the great advantage of Adaptive Contents is that it's possible to serve optimal contents for each use case and reach the maximum score otherwise impossible to achieve with other technologies.

The screenshot shows the 'Adaptive Contents Settings' page. On the left is a navigation menu with options like 'Auto Configuration', 'General', 'Assets Inclusions', 'Assets Exclusions', 'Lazy Load', 'Optimize Images', 'Adaptive Contents' (highlighted), 'Page Cache', 'Assets Management', 'Http/2', 'Combine Images', 'CDN', and 'Advanced'. The main content area is titled 'Adaptive Contents Settings' and includes the following sections:

- Enable Adaptive Contents:** A toggle switch is turned on. A callout bubble points to this section with the text: "Enable Adaptive Contents and enter CSS selectors for elements to remove".
- CSS selectors for elements to remove (Requires "Minify HTML" enabled):** A text input field contains the selector "script[src]=frame video". A callout bubble points to this field with the text: "Define the list of user-agent strings".
- Bots user-agent strings:** Two buttons labeled "lighthouse" and "googlebot" are visible. A callout bubble points to this section with the text: "Choose to remove all non-critical JS/CSS code".
- Remove all JS:** A toggle switch is turned on. Below it, text reads: "If enabled, all JS code will be removed for bots and page speed tools."
- Remove all CSS:** A toggle switch is turned on. Below it, text reads: "If enabled, all CSS code will be removed for bots and page speed tools."
- Essential CSS code:** A large empty text area for defining essential CSS code. Below it, a note states: "It's possible to add a piece of essential CSS code that should be present in the lightweight version of a page generated with Adaptive Contents for bots and page speed tools. This is valuable, for example, if you choose to remove all CSS but still need some styles to preserve a basic layout or if you experience warnings in Google Search Console about 'Mobile Usability Issue - Content Wider Than Screen & Clickable Element Too Close'."

Page Cache and Instant Page

WPSpeed integrates in its core a page caching system, so that there is no need to use a separate plugin for this purpose, furthermore the great advantage is that the WPSpeed page cache perfectly copes with the rest of optimizations.

Thanks to a page cache, the HTML source code of pages can be entirely cached to significantly speed up the page load. Indeed there is no longer need to hit your server PHP to re-process the final page each time, nor to query your database. The resulting page once that has been cached, is reused as is for subsequent requests.

The page cache can be tricky while configuring the plugin or making changes to the site, in this case pay attention to keep the page cache disabled or clear it.

WPSpeed also includes an innovative feature to have an instant page load for human visitors, in order to optimize performance for bots but also for visitors.

This system is named 'Instant Page' and it preloads the page before that it's actually opened with the result to have it rendered instantly if the user chooses to click and open it.

The image shows a screenshot of the WPSpeed configuration interface. On the left is a dark sidebar with a menu containing: Auto Configuration, General, Assets Inclusions, Assets Exclusions, Lazy Load, Optimize Images, Adaptive Contents, Page Cache (highlighted), Assets Management, Httpr2, Combine Images, CDN, and Advanced. The main content area is divided into two sections. The top section is titled 'Page Cache' and contains several settings: 'Enable PHP page caching' (checked), 'Enable Itaccess page caching' (unchecked), 'Page cache lifetime' (set to 12 hours), and 'Delete all cache folders' (unchecked). A blue callout bubble points to the 'Enable PHP page caching' toggle with the text 'Enable page cache, choose settings such as the cache lifetime'. The bottom section is titled 'Instant Page' and contains: 'Enable Instant Page preloading' (checked), a 'Preload delay' dropdown set to 'Fast delay', and a note about the default 65 ms delay. A blue callout bubble points to the 'Enable Instant Page preloading' toggle with the text 'Enable the Instant Page system'.

Assets management

WPSpeed includes a special feature to remove Javascript and CSS resources that have an heavy impact on the performance of the website.

Thanks to WPSpeed it's possible to easily remove the loading of these resources through below settings, but keep in mind that removing a resource could cause the loss of important functionalities for the website, or change its layout and aspect.

So you should use this feature carefully to disable only features that are not essential or assets that are not used on your website.

Keep in mind that after removing a given resource you may need to clear the WPSpeed cache in order to refresh the compiled JS/CSS files and apply changes.

Furthermore it's also possible to add custom snippets of JS and CSS code if you need for example to apply certain fixes to styles or load dynamic scripts.

The screenshot displays the 'Assets Management' configuration page in WPSpeed. The left sidebar contains a navigation menu with options: Auto Configuration, General, Assets Inclusions, Assets Exclusions, Lazy Load, Optimize Images, Adaptive Contents, Page Cache, Assets Management (highlighted), Http/2, Combine Images, CDN, and Advanced. The main content area is divided into several sections:

- Remove CSS Files:** A section titled 'Remove CSS Files' with a sub-section 'Remove unused CSS files'. It features a text input field containing '.style.min.css'. A callout bubble points to this field with the text 'Type in the Javascript files that need be removed'.
- Remove Javascript Files:** A section titled 'Remove Javascript Files' with a sub-section 'Remove unused javascript files'. It features a text input field containing '_loader.min.js'. A callout bubble points to this field with the text 'Type in the CSS files that need be removed'.
- Add custom Javascript:** A section titled 'Add custom Javascript' with a sub-section 'Add custom JS code'. It includes a text area containing 'console.log("my code");'. A callout bubble points to this text area with the text 'Add custom JS or CSS code'.
- Add custom CSS:** A section titled 'Add custom CSS' with a sub-section 'Add custom CSS code'.

HTTP/2

HTTP/2 Server Push is a modern technology that allows an HTTP/2-compliant server to send resources to a HTTP/2-compliant client before the client requests them.

It is, for the most part, a performance technique that can be helpful in loading resources preemptively.

If this option is enabled, WPSpeed will communicate to the server to operate using this technology, keep in mind that this only works on a HTTP/2 compliant server.

If you have doubts about the compatibility of your server with the HTTP/2 protocol, it's recommended to contact your hosting provider.

The screenshot shows the 'HTTP/2 Settings' page in WPSpeed. On the left is a navigation menu with options like 'Auto Configuration', 'General', 'Assets Inclusions', 'Assets Exclusions', 'Lazy Load', 'Optimize Images', 'Adaptive Contents', 'Page Cache', 'Assets Management', 'HTTP/2', 'Combine Images', 'CDN', and 'Advanced'. The 'HTTP/2' option is selected. The main content area is titled 'HTTP/2 Settings' and contains the following sections:

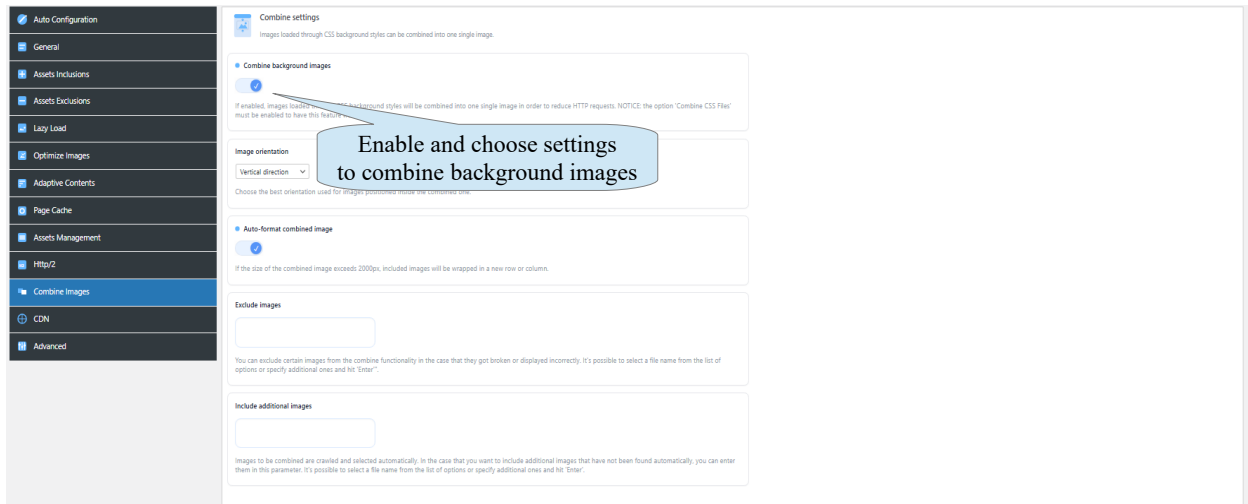
- Enable HTTP/2 Server Push:** A checkbox that is checked. A blue callout bubble points to it with the text "Enable the HTTP/2 Server Push headers". Below this checkbox is a small text block: "HTTP/2 Server Push allows an HTTP/2-compliant server to send resources to a HTTP/2-compliant client before the client requests them. This technique can be helpful in loading resources preemptively. If this option is enabled, WPSpeed will communicate to the server to operate with this technique. Keep in mind that this only works on a HTTP/2 compliant server."
- Exclude asynchronous assets:** A checkbox that is checked. Below it is the text: "If enabled, assets that are deferred, loaded asynchronously or lazy-loaded will be excluded."
- Include CDN files:** A checkbox that is unchecked. Below it is the text: "Files loaded over your CDN domains will also be included."
- File types:** A section with buttons for "style", "script", "font", and "image". Below the buttons is the text: "Select the type of files to preload through HTTP/2."
- Add custom files:** A text input field. Below it is the text: "It's possible to preload custom files adding their full path in this field. supported file types are: js, css, webp, gif, png, jpg,woff, woff2"
- Exclude files:** A text input field. Below it is the text: "If you receive warnings in the browser console about preloaded files not used within a few seconds, you can exclude these files here."

Server push lets the server preemptively “push” website assets to the client without the user having explicitly asked for them. When used with care, it's possible to send what it's known the user is going to need for the page they're requesting. Resources can be downloaded in parallel.

Combine images

In the case that your template and installed plugins load several images through CSS background styles, thanks to the 'Combine images' feature they can be combined into one single image working as a container, in order to reduce the number of HTTP requests.

You can choose and customize several aspects of this functionality, mainly to avoid issues and undesired results when original images are wrapped within the container:



To avoid issues and incorrect display, it could be particularly useful to:

- Manage the orientation of combined images within the container image
- Enable the auto-format
- Exclude specific images
- Include additional images

CDN usage

If you use a CDN, you can take advantage of the WPSpeed plugin so that all static assets such as background images, Javascript, CSS files, etc will be loaded automatically from your CDN domain.

You can specify up to 3 different CDN domains for various assets loaded on the website, it's enough to choose the scheme, domain and resource types:

CDN Settings

Enter CDN domains to have the plugin load all static files from these external domains.

Enable CDN

Enable CDN

If enabled, all static assets on your website will be loaded from the CDN. If you have not correctly setup a CDN for your website, you can configure up to 3 different CDN domains for various assets loaded on the website, it's enough to choose the scheme, domain and resource types:

Preconnect browsers to CDN

Preconnect browsers to CDN

If enabled, the browser will perform an early preconnect to your CDN domains to speed up loading of subsequent resources.

CDN scheme

Relative

Select the scheme for your CDN domain.

CDN Domain #1

Enter the CDN domain #1.

CDN files #1

css png gif ico pdf js jpeg bmp webp svg

Select file types that you want to be loaded over a CDN with the domain #1. It's possible to select a value from the list of options or specify additional ones and hit 'Enter'.

For all CDN domains it's possible to set:

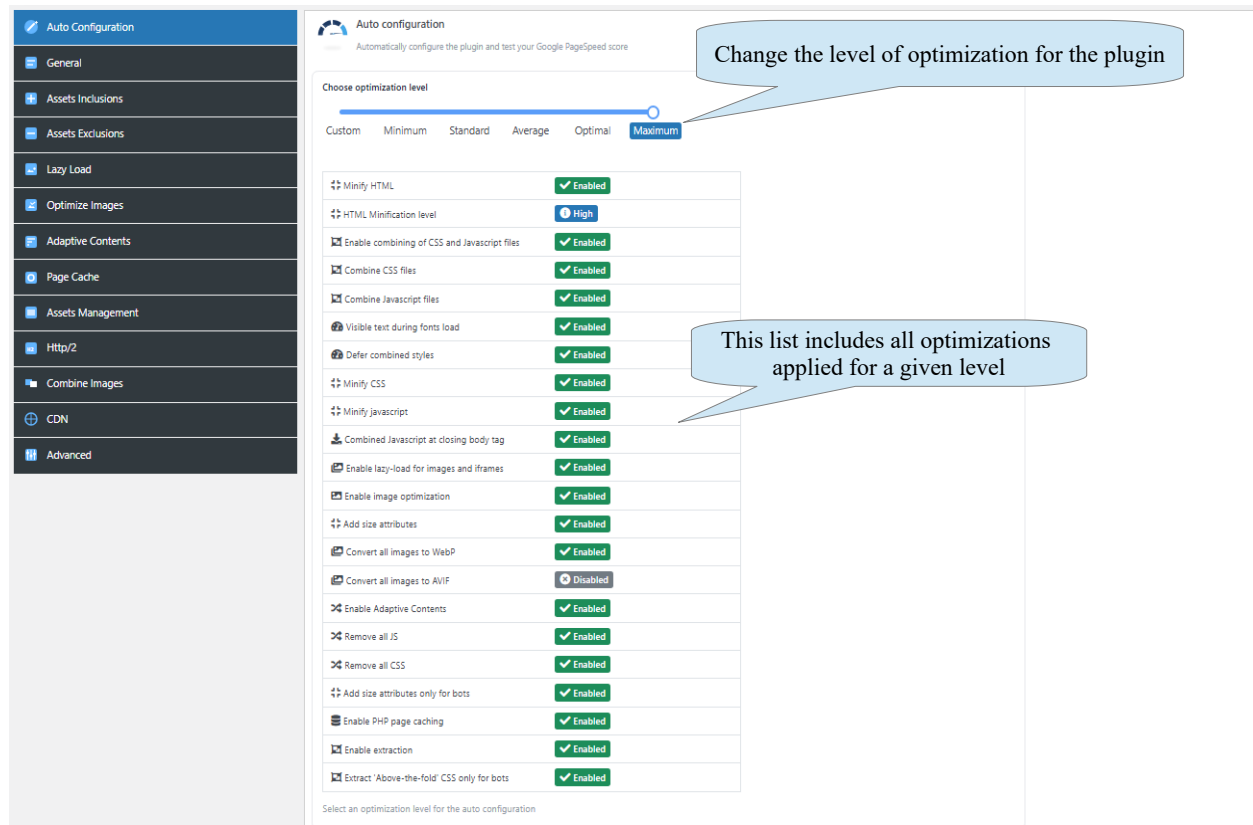
1. CDN domain
2. Type of files loaded from a CDN
3. Preconnect mode to further speed up loading of resources

Auto configuration and Google PageSpeed API

To simplify the process of configuring the plugin through its many settings, JSPEED includes an auto configuration system capable of increasing the level of optimization and achieving the best results while preserving the functionality of the website.

Using the auto configuration tool is straightforward, just change the slider level and test the website performance until you find the optimal combination of settings.

Once the level of optimization has been selected, save the configuration to apply settings. Of course, after applying a certain level of optimization, you can still customize individual settings one by one.



The screenshot displays the 'Auto configuration' interface. On the left is a sidebar with navigation options: Auto Configuration, General, Assets Inclusions, Assets Exclusions, Lazy Load, Optimize Images, Adaptive Contents, Page Cache, Assets Management, Http/2, Combine Images, CDN, and Advanced. The main panel shows the 'Auto configuration' section with a subtitle 'Automatically configure the plugin and test your Google PageSpeed score'. Below this is a 'Choose optimization level' section with a slider and buttons for Custom, Minimum, Standard, Average, Optimal, and Maximum. A callout bubble points to the Maximum button with the text 'Change the level of optimization for the plugin'. Below the slider is a list of 25 optimization settings, each with a status indicator (Enabled or Disabled). A callout bubble points to this list with the text 'This list includes all optimizations applied for a given level'. At the bottom of the main panel, there is a note: 'Select an optimization level for the auto configuration'.

| Setting | Status |
|--|----------|
| Minify HTML | Enabled |
| HTML Minification level | High |
| Enable combining of CSS and Javascript files | Enabled |
| Combine CSS files | Enabled |
| Combine Javascript files | Enabled |
| Visible text during fonts load | Enabled |
| Defer combined styles | Enabled |
| Minify CSS | Enabled |
| Minify javascript | Enabled |
| Combined Javascript at closing body tag | Enabled |
| Enable lazy-load for images and iframes | Enabled |
| Enable image optimization | Enabled |
| Add size attributes | Enabled |
| Convert all images to WebP | Enabled |
| Convert all images to AVIF | Disabled |
| Enable Adaptive Contents | Enabled |
| Remove all JS | Enabled |
| Remove all CSS | Enabled |
| Add size attributes only for bots | Enabled |
| Enable PHP page caching | Enabled |
| Enable extraction | Enabled |
| Extract 'Above-the-fold' CSS only for bots | Enabled |

To test the performance of the website against JSPEED optimizations, you can take advantage of the Google's built-in PageSpeed testing tool.

It's no longer necessary to manually test the Google PageSpeed score on an external website, indeed JSPEED implements the Google PageSpeed API and integrates the results in the backend. Thanks to the integrated tool, it's possible to monitor the performance and the score of each page efficiently.

By default the tool will execute the Google PageSpeed test for your website homepage, but it's also possible to enter a specific page URL to test.

Finally you could also register your own Google application and ApiKey, in this case just enter the generated ApiKey in the relevant field.

- Dashboard
- Posts
- Media
- Pages
- Comments
- Appearance
- Plugins 3
- Users
- Tools
- Settings
- General
- Writing
- Reading
- Discussion
- Media
- Permalinks
- Privacy
- Akismet Anti-Spam
- WPSpeed Settings
- GDPR Cookie Compliance
- ⌵ Collapse menu



Google PageSpeed score 1

Test Google PageSpeed Insights

Click the button to start the Google PageSpeed Insights test

Performance report for both mobile and desktop version

https://storeextensions.org

| Mobile | Desktop |
|--|---|
| <div style="font-size: 2em; font-weight: bold; color: #f79646;">73</div> | <div style="font-size: 2em; font-weight: bold; color: #42ad58;">97</div> |
| <ul style="list-style-type: none"> ▲ First Contentful Paint 3.4 s ● Speed Index 3.4 s ■ Largest Contentful Paint 4 s ■ Time To Interactive 5.4 s ■ Total Blocking Time 246 ms ● Cumulative Layout Shift 0.027 | <ul style="list-style-type: none"> ● First Contentful Paint 0.8 s ● Speed Index 1.2 s ● Largest Contentful Paint 1 s ● Time To Interactive 0.8 s ● Total Blocking Time 0 ms ● Cumulative Layout Shift 0.016 |
|  |  |

Page URL to test 1

Custom Google PageSpeed ApiKey 1

The report applies to the selected page URL

Advanced settings and features

FinallyWPSpeed includes some advanced features that could help to improve the performance score of your website.

In more details WPSpeed is capable of:

- **Extract Basic CSS Styles:** this feature extracts basic CSS styles required to format the page above the fold and put this in a `<style>` element inside the `<head>` section of the HTML to prevent 'render-blocking'.
- **Reduce DOM Tree:** HTML5 DOM elements exceeding the limit of 600 below the fold will be removed and loaded asynchronously using Javascript after that the page has been fully rendered.

The plugin includes also a debug mode and a log system.

If you experience complex conflicts to solve, consider to enable the debug mode of WPSpeed.

Thanks to the debug mode of WPSpeed, each URL of an original file is included in the combined files and the log system will be enabled so that you can inspect the log file at the path: `root/wp-content/plugins/wpspeed/logs`

The screenshot displays the 'Advanced' settings page of the WPSpeed plugin. On the left is a sidebar menu with options: Auto Configuration, General, Assets Inclusions, Assets Exclusions, Lazy Load, Optimize Images, Adaptive Contents, Page Cache, Assets Management, Http/2, Combine Images, CDN, and Advanced (highlighted). The main content area is divided into two sections: 'Extract Basic CSS Styles' and 'Reduce DOM Tree'.
The 'Extract Basic CSS Styles' section includes:

- An 'Enable extraction' toggle switch, which is turned on. A callout bubble points to this section with the text: 'Advanced optimizations of WPSpeed to extract critical CSS'.
- A 'Number of elements' dropdown menu set to '800'. Below it, a note says: 'Select the number of HTML elements from the top of the page that you want to analyze to find the basic CSS to be extracted from.'
- A 'Remove unused CSS' toggle switch, which is turned off. Below it, a note says: 'If enabled, the plugin will "lazy-load" the unused CSS to prevent unnecessary processing before than the page is loaded.'
- An 'Exclude @font-face' toggle switch, which is turned on. A callout bubble points to this section with the text: 'Exclude @font-face'.
- A 'CSS selectors' text input field. Below it, a note says: 'It could be necessary to extract the basic CSS for some custom elements. Add any substring from the CSS declaration here to have them included in the extracted CSS code and hit "Enter".'
- An 'Exclude extraction by page URL' text input field. Below it, a note says: 'Enter a substring of each url that you want to exclude from the CSS extraction. It's not needed to enter the complete url for the matching, but only a part of it. Add a string and hit "Enter".'

The 'Reduce DOM Tree' section includes:

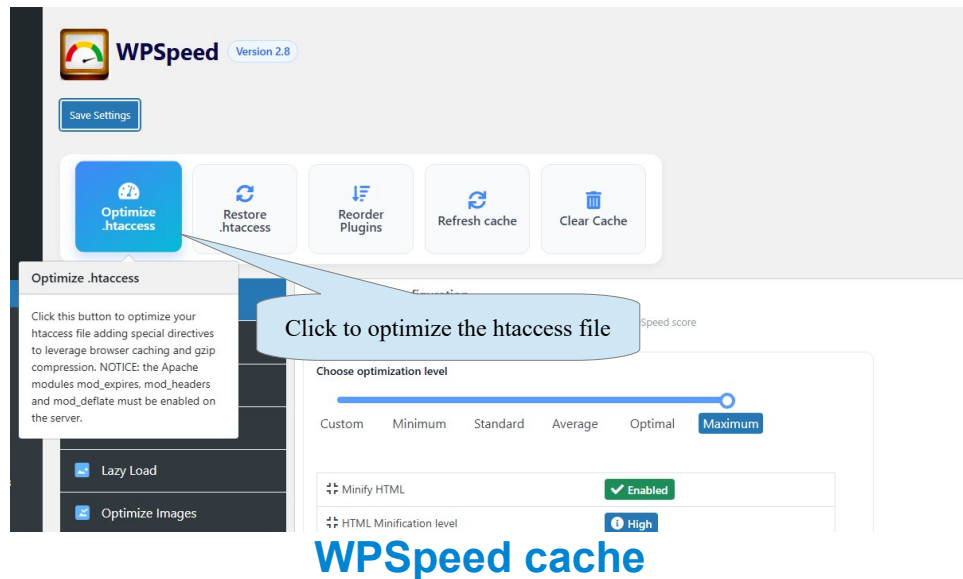
- An 'Enable DOM reduction' toggle switch, which is turned on. A callout bubble points to this section with the text: 'Reduce the DOM complexity'.
- An 'HTML tags to load asynchronously' section with a grid of buttons: 'section', 'header', 'footer', 'aside', and 'nav'. Each button has a small blue circle with a white checkmark inside it.
- A note below the buttons says: 'Select which HTML elements you want to load asynchronously.'

Htaccess optimization

WPSpeed is capable to optimize your htaccess in order to leverage browser caching for loaded assets such as images, CSS, scripts, etc

This is accomplished adding special directives to leverage browser caching and gzip compression, keep in mind that the Apache modules *mod_expires*, *mod_headers* and *mod_deflate* must be enabled on the server in order to have these optimizations working. If you have doubts about the activation of these modules on your server, it's recommended to contact your hosting provider to get the list of installed and enabled Apache modules. It also adds htaccess rules for the new Brotli compression that can be used if your server supports it.

To optimize your htaccess file it's enough to click once on the toolbar button:



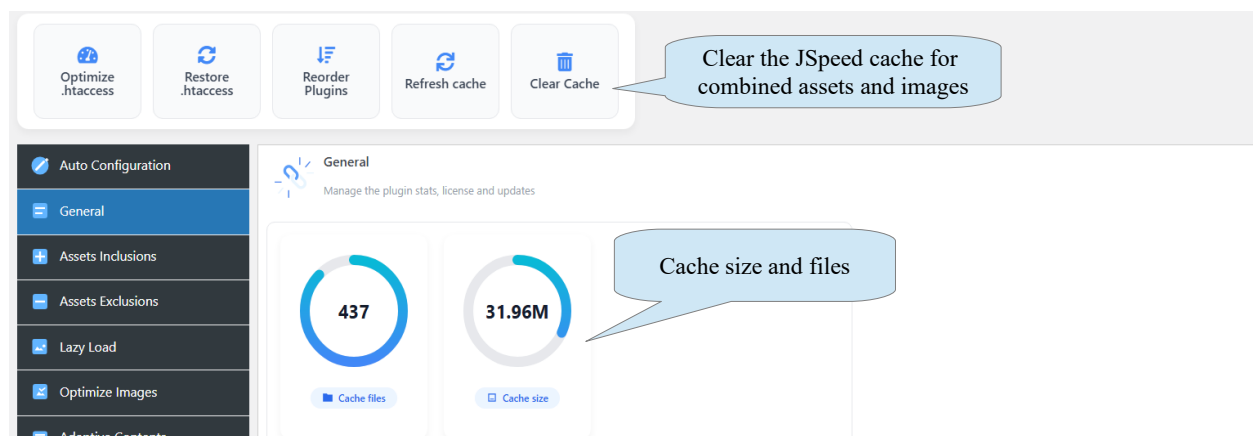
The plugin uses a cache for all combined assets in order to process them only once and keep them valid until the cache becomes stale.

If you apply changes to the website you may need to clear the WPSpeed cache in order to get modifications applied immediately, without waiting that the cache lifetime expires.

To do this, it's enough to click on the toolbar button 'Clear cache'. By clicking the 'Clear cache' button, also the page cache will be deleted if it has been enabled.

As an alternative you can just refresh the cache by clicking on the button 'Refresh cache' so that browsers will fetch updated contents.

There are also labels that show the amount of cache generated on your server by WPSpeed so that it's possible to have its size and the number of files under control.



Auto updates

To update the WPSpeed PRO plugin using the Wordpress updates system you must enter the exact email address used to register on our site <https://storejextensions.org>.

The email address is used to identify your account and the validity of the updates license for the PRO paid version.

Free updates are included for 1 year from the date of the latest purchase.

More info about licensing and updates at our FAQ page <https://storejextensions.org/faq.html>"

The screenshot displays the 'General' settings page for the WPSpeed PRO plugin. The left sidebar contains a menu with options: Auto Configuration, General (selected), Assets Inclusions, Assets Exclusions, Lazy Load, Optimize Images, Adaptive Contents, Page Cache, Assets Management, Http/2, and Combine Images. The main content area shows 'General' settings for managing plugin stats, license, and updates. It features two circular progress indicators: 'Cache files' at 437 and 'Cache size' at 31.96M. Below these is a section titled 'Registration email address for updates' with an empty input field. A blue callout box points to the input field with the text: 'Enter your email address used to register on <https://storejextensions.org> to validate your updates license'.

Our commercial plugins requires a valid license in order to be updated using the Wordpress updates system or manually by downloading and installing the latest package through the Wordpress plugins installer.