Speed Optimization PRO

Plugin for Joomla! 3, 4 & 5

This manual documents **version 22.x** of the Joomla! extension.

https://www.aimy-extensions.com/joomla/speed-optimization.html

1 Introduction

Aimy Speed Optimization is a Joomla! plugin that optimizes your content for page speed in a couple of ways:

It allows you to **minify & embed CSS** in your pages and **queues JavaScript asynchronously** so both do no longer block your pages' rendering. Theses JavaScript and CSS resources can be automatically preloaded as well. Additionally the HTML code can be minified.

The way images are loaded can be optimized: You can choose to either just add missing **image dimensions** or load images **deferred** or **lazy**.

Besides that, **browser caching** and **compression** can be activated and configured automatically (using Apache's `.htaccess` file).

These features help to speed up your Joomla! website with a faster rendering in your visitor's browsers.

With Google™ *Page Speed Insights* you can check your website's speed from various perspectives. We believe this is not a coincident, but a chance to tune your website for your visitors and gain some **SEO benefits** as well.

2 Supported Joomla! Versions

Aimy Speed Optimization supports Joomla!...

- 3.9 and up,
- 4.0 and up,
- 5.0 and up.

3 Requirements

Aimy Speed Optimization requires:

1. a supported **Joomla!** version (see above)
2. **PHP 7.0.0** or above
3. **DOM** PHP extension (usually enabled)
4. **Mbstring** PHP extension (usually enabled)
5. a Joomla! template that uses **HTML 5**

4 Installation

The installation of the extension follows the common Joomla! procedures.

In case you are not familiar with these procedures, proceed as follows:

1. Download the extension's ZIP archive
2. Log into your Joomla! backend as "Super User"
3. From the menu, choose "*Extensions*" → "*Manage*" → "*Install*"
4. Click on the "*Or browse for file*" button and select the ZIP archive

The extension's archive will be uploaded and installed afterwards.

For further information, please have a look at the Joomla! documentation Installing an Extension.

NOTE: All plugins are disabled by Joomla! when installed for the first time. To enable Aimy Speed Optimization, proceed and configure the plugin.

5 Configuration

After a fresh installation, click on the "*Configure plugin now*" button on the installation report page.

At any time, you can configure the plugin using Joomla!'s Plugin Manager by choosing "*Extensions*" → "*Plugin Manager*" from the menu. Locate the plugin and click on its name in the "*Plugin Name*" column of the plugin listing.
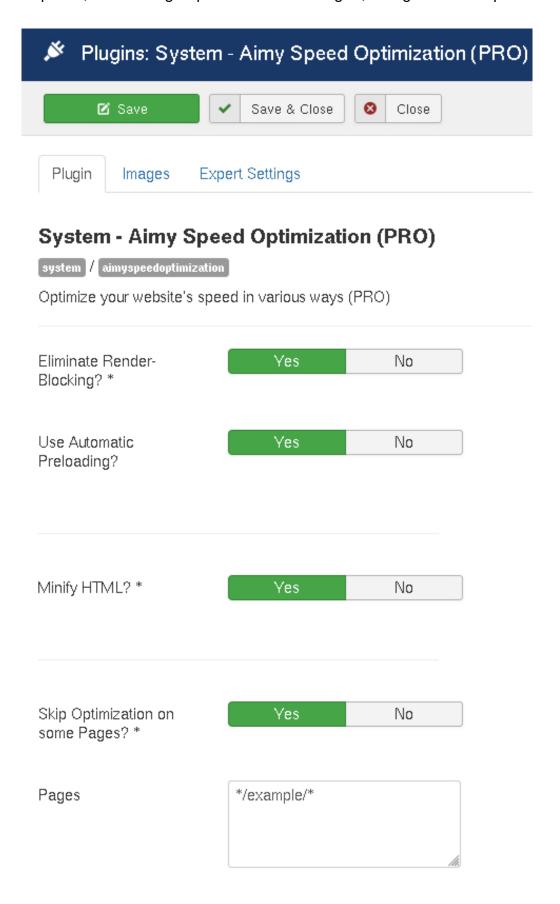
5.1 Enabling the Plugin

In order to use the plugin's functionality, you have to enable it first.

To do so, change the plugin's status from "*Disabled*" to "*Enabled*" and apply your changes by clicking on either the "*Save*" or "*Save & Close*" button in the toolbar.

The default configuration of Aimy Speed Optimization leaves any decision how to optimize page speed on your Joomla! website up to you, that is, the default configuration does not enable any of the available optimizations.

5.2 Options: Plugin

Aimy Speed Optimization's functionality can be suited to your needs using the following configuration options, which are grouped in the tabs "Plugin", "Images" and "Expert Settings":

Eliminate Render-Blocking?

This option allows to optimize both **CSS and JavaScript elements** so that they will no longer block the rendering of your page.

CSS is minified and directly embedded into the HTML code. JavaScript files and all inline code is loaded and executed asynchronously in a queue, that is one after another, keeping their sequence intact.

NOTE: Dynamically generated and offsite CSS is handled in a way similar to JavaScript: it is queued and loaded after the initial rendering of your website has finished. This approach usually works as expected for non-ciritical CSS like fonts and prevents your website from blocking. However, if the CSS is essential for rendering your website (for example Bootstrap from a CDN) there is no way to prevent blocking. In this case, the solution would be to store the CSS code on your webspace so it can be embedded.

Use Automatic Preloading?

If enabled, specific **preload** `link` elements will be added to a page. These elements allow a visitor's browser to preload all queued **CSS and JavaScript elements** in the background so that these elements may be already loaded when they are required.

Minify HTML?

If enabled, your HTML code will be minified before being delivered to your visitor's browser.

Enabling this setting is recommended only if the "*Expert Settings*" option of "*Enable Compression*" cannot be used.

Skip Optimization on some Pages?

If you like to **exclude single pages** of your website from the optimization, you can list them here. This may be useful if an extension used on a specific page of your website is not compatible with Aimy Speed Optimization.

Patterns are supposed to be entered **one on each line** - currently only the **wildcard character** ("`*`") is supported as a non-literal character.

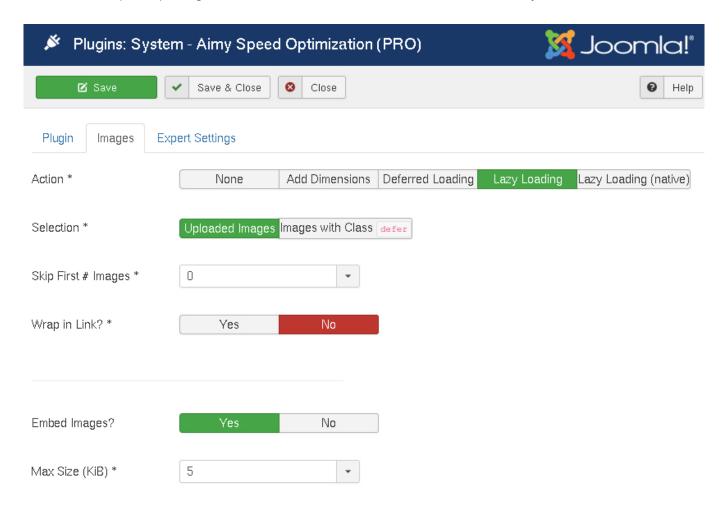Patterns are matched against the combination of **path and query string** of the current URL.

Example: given the URL "`https://test.org/gallery.html?page=12`", patterns are matched against "`/gallery.html?page=12`".

If one of the entered pattern matches, no on-page optimization (as defined in the "*Plugin*" and "*Images*" tabs) will be done on that page. To exclude the page given in the above example, you could for example enter the following pattern: "`/gallery.html*`".

5.3 Options: Images

Aimy Speed Optimization gives you a couple of options to optimize image loading on your website (`<img>` elements and `<source>` elements in `<picture>` containers). Both popular techniques of **deferred loading** and **lazy loading** are supported. You can either choose to manually select which images should be optimized or automatically select all uploaded images.

Furthermore (small) images can be embedded in the HTML-code directly.

Action

Which action should be applied to your images?

None

Keep images as they are.

Add Dimensions

Automatically add missing image dimensions ("*width*" and "*height*" attributes).

This option handles onsite images only.

Deferred Loading

Replace all selected images with a transparent placeholder. To speed up rendering, missing image dimensions of the original image are added.

Once the page **has been initially loaded** by the browser, all image placeholders will be replaced with the original images.

Note: This options works for onsite and offsite images. But if the image is loaded from an external website, dimensions cannot be set automatically. In this case, we

recommend to add *width* and *height* manually.

Lazy Loading

Replace all selected images with a transparent placeholder. To speed up rendering, missing image dimensions of the original image are added.

The original images are then automatically loaded whenever they **get near the user's view**, i.e. by scrolling.

Note: This options works for onsite and offsite images. But if the image is loaded from an external website, dimensions cannot be set automatically. In this case, we recommend to add *width* and *height* manually.

Lazy Loading (native)

If enabled, a "`loading`" attribute with the value of "`lazy`" will be added to all selected images. A visitor's browser will then take care of loading the image in a lazy manner if the feature is supported natively.

Note: As this feature just annotates the selected images for being lazy loaded and leaves the technical details for the browser's native support, it does not require any JavaScript.

We recommend to use **native lazy image loading** as it does not depend on any JavaScript code, does not fall under the *Restrictions* mentioned below and native support of modern browsers is already about 75% and growing (as of November 2021).

Selection

Which images should be optimized?

Uploaded Images

All onsite images uploaded to Joomla!'s `/images/` directory, i.e. using *com_media* ("*Content*" → "*Media*") or those stored beneath the `/cache/` directory.

Images with Class "defer"

All images (onsite or offsite) which do have a class name of "*defer*" assigned, i.e.:

```
<img src="/images/pagespeed.jpg" alt="" class="defer" />
```

Most Joomla! editors allow to add classes to images easily.

We recommend to manually select images by adding the **"defer" class** to them. The manual selection sure takes time but in our experience leads to best results.

Skip First # Images

Set whether to skip the first images of the selection given above.

Wrap in Link?

Both deferred loading and lazy loading make use of the well-known "*data-src*" attribute, which is used by Google™ itself.

However, this option allows to wrap each image (<img> element) of the selection in an HTML link (<a> element) that points to the original image, unless it is not used within a link already. This way, you can make search engines find your images more easily.

We recommend to enable this option only if you have problems with search engines not picking up your deferred or lazy loaded images.

Note: This option is not needed (and therefore not available) if **native lazy loading** has been enabled.

Embed Images

If enabled, images of a certain file size will be embedded inline in a page's HTML code directly. That helps to reduce the amount of HTTP requests required to transfer the content of a page. It is recommended for small images (for example icons, divider, flags, buttons and such).
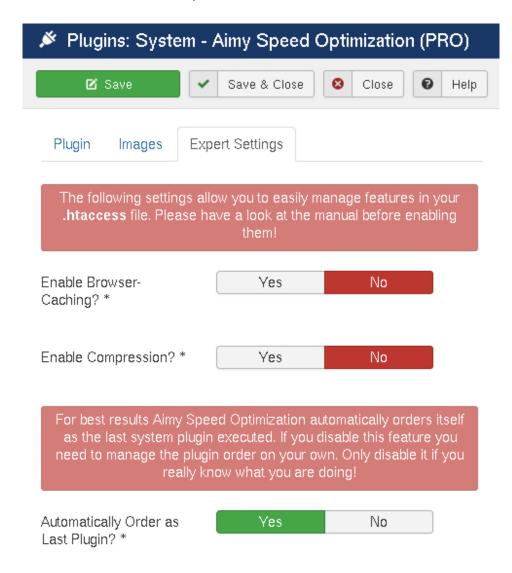
Max Size (KiB)

Set the maximum size of an image that should be embedded directly.

5.4 Options: Expert Settings

The first two option of the expert settings allow you to change your `.htaccess` file directly from the Joomla! backend. They require that your website is served by an **Apache web server** and that an `.htaccess` file is already present in the root directory of your Joomla! installation (i.e. the one shipped with Joomla!).

Please do not manually change the marker lines inserted by this plugin - they are required in order to update your configuration according to your settings.

WARNING: Before enabling an expert option, be sure that your webserver is configured to allow the overrides specified along with the settings to be used. Usually these directives are available and allowed. However, if in doubt, ask your webhoster to be sure!

Enable Browser-Caching

> If enabled, Apache's **mod_expires** is activated and configured to send caching instructions to your visitor's browsers in the following way:

> **Non-Static Resources (not cached)**

>> *HTML, XML, JSON*

> **Static Resources (cached for up to one year)**

>> *PNG, GIF, JPEG, ICO, WEBP*

>> *JavaScript, CSS*

EOT, *WOFF*, *WOFF2*, *TTF*, *SVG*

Required Apache Override: `Indexes`

Hint: If you could set a custom *Directory Index* in your `.htaccess` file, you can safely enable this option as well, as both require the same Apache Override to be allowed.

Enable Compression

If enabled, Apache's **mod_deflate** is configured to compress content of the following types:

HTML, *XML*, *JSON*, *JavaScript*, *CSS*, *SVG*

Required Apache Override: `FileInfo`

Hint: If you are using SEF and successfully enabled *mod_rewrite* in Joomla!'s Global Configuration, you can safely enable this option as well, as both require the same Apache Override to be allowed.

If any of these settings is changed, your `.htaccess` file will be updated accordingly when the configuration is saved.

HINT: If the plugin is **deinstalled**, the lines inserted by Aimy Speed Optimization will be deleted automatically. If you had enabled the options before and then **deactivate** the extension the lines will still be in your `.htaccess` file. Please disable these options and save this setting before deactivating the plugin.

Automatically Order as Last Plugin

If enabled, Aimy Speed Optimization will automatically order itself as the last system plugin executed to achieve best optimization results.

HINT: It is recommended to keep the default value of "*Yes*" unless you really know what you are doing.

6 Restrictions

Please go through the restrictions carefully to know which features of Aimy Speed Optimization will be usable on your website and which extensions are currently unsupported:

6.1 Features

Eliminate Render-Blocking

Embedding CSS

Internet Explorer 8 will not interpret any media queries in embedded CSS code, even if an auxiliary library like *respond.js* is used.

Dynamically generated and offsite CSS code cannot be embedded (i.e. CSS of an offsite webfont).

Queuing JavaScript

Loading of JavaScript in a *queue* asynchronously without blocking the rendering of your page has been tested extensively and is known to work with other JavaScript frameworks as well, but only **jQuery** and **jQuery UI** are officially supported right now.

Require.js resources are kept as is, because they are usually loaded asynchronously on their own and work as expected with Aimy Speed Optimization.

JavaScript code of **Google Analytics Experiments** (used for A/B testing) and **Facebook Pixel Events** is kept as is and will not be queued.

Images: Deferred Loading & Lazy Loading

Both options require JavaScript to be enabled in a user's browser. All common browsers and well-known search engines do support JavaScript and enable it by default.

Images: Lazy Loading

Lazy loaded images may only be available in the **print version** or in **Reader View** if they have been loaded already, that is, if they have once been in the reader's view.

Images: Wrap in Link

Automatically wrapping all deferred or lazy loaded images (`<img>` elements only) in a link may break some image galleries or slideshows which make assumptions on the structure of the images and their containers on their own.

`<source>` elements in `<picture>` containers won't be wrapped.

Browser Caching and Compression

Browser Caching and Compression are realized via the configuration of the `.htaccess` file. This will only work for **Apache web servers**.

6.2 Unsupported Extensions & Services

Some extensions are known to be incompatible with Aimy Speed Optimization. If they are used on single pages only, you may use the option "Skip Optimization on some Pages" (see *Options: Plugin* above).

The following extensions and services are currently not supported:

Page Cache (system plugin)

> Alternative: set "Conservative Caching" or "Progressive Caching" in Joomla!'s "Global Configuration" ("*System*" → "*Cache Settings*").

ReCaptcha (captcha plugin)

> The plugin does not work if the "*Eliminate Render-Blocking*" option is active.

> Alternative: use another captcha plugin like Aimy Captcha-Less Form Guard.

J51 - Florence (template)

> The template does not work as expected if the "*Eliminate Render-Blocking*" option is active.

Vertical Menu (module)

> The module does not work as expected if the "*Eliminate Render-Blocking*" option is active.

Helix Ultimate, Helix 3, Helix ... (template)

> These templates do not work as expected if the "*Eliminate Render-Blocking*" option is active.

Horizontal scrolling slideshow (module)

> The module is not covered by the "*Eliminate Render-Blocking*" optimizations: its code is kept as-is and the page it is embedded on may block.

RAXO All-mode (module)

> These templates does not work as expected if the "*Eliminate Render-Blocking*" option is active.

sigplus Image Gallery (plugin)

> The plugin does not work if the "*Eliminate Render-Blocking*" option is active.

Consent Management Provider (CMP) (service)

> If the usage of CMP code is detected on a page, the "*Eliminate Render-Blocking*" option is automatically disabled (for that page).

Besides that, not all features may work when **Frontend Editing** is used.

7 Copyright & Trademark Notice

The Joomla!® name and logo are trademarks of Open Source Matters, Inc. in the United States and other countries.

Google and the Google logo are registered trademarks of Google Inc.

Mentioned hard- and software as well as companies may be trademarks of their respective owners. Use of a term in this manual should not be regarded as affecting the validity of any trademark or service mark. A missing annotation of the trademark may not lead to the assumption that no trademark is claimed and may thus be used freely.

Aimy Speed Optimization does include and use the following Open Source libraries:

HTML5-PHP

> developed by Matt Butcher, Matt Farina & Asmir Mustafic.

MinifyCSS & MinifyHTML

> developed by Netzum Sorglos Software GmbH.

JShrink

> developed by Robert Hafner.

$script.js JavaScript loader & dependency manager

> developed by Dustin Diaz.

[be]Lazy.js JavaScript lazy image loader

> developed by Bjørn Klinggaard.