

amasty

For more details see how [Google Page Speed Optimizer](#) extension works.

Guide for Google Page Speed Optimizer for Magento 2

Improve your store performance with the Google Page Speed Optimizer for Magento 2. Minify code and optimize images without quality loss.

- Automatically reduce images size
- Optimize your database performance
- Minify website code structure
- Improve your webstore pages performance
- Raise your store score in Google PageSpeed Insights up to 99

NEW! Now the extension is compatible with Varnish. You can also add compatibility with 3-rd party extensions which use AJAX.

Configuration

To configure the extension, go to **Stores → Configuration → Google Page Speed Optimizer**.

Configuration

 demouser ▾

Store View: Default Config ▾ 

Save Config

 **AMASTY EXTENSIONS** 

Extensions & Notifications

Google Page Speed Optimizer

Diagnostic

General 

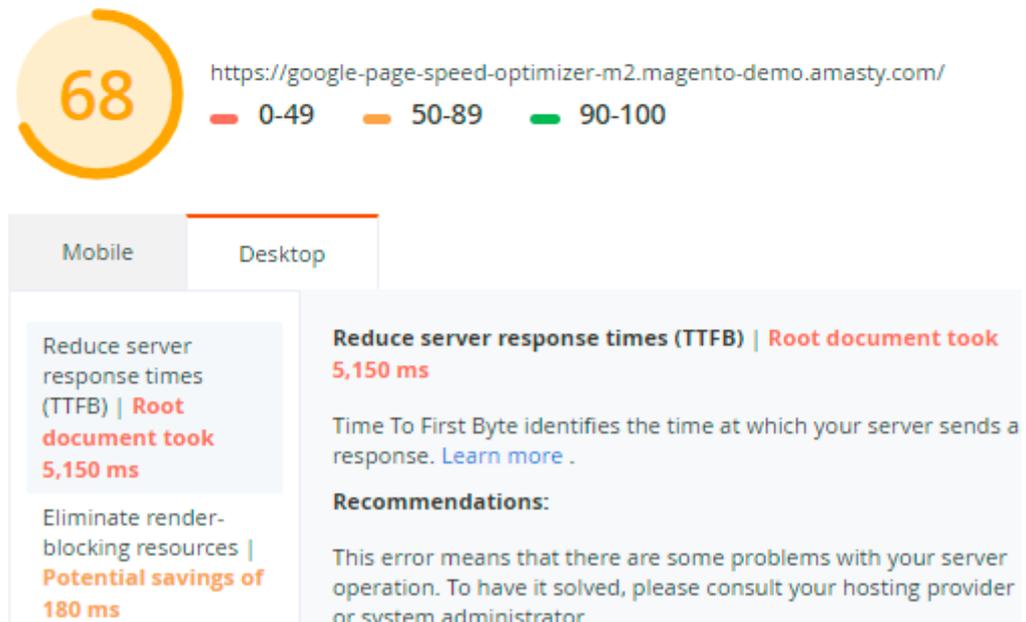
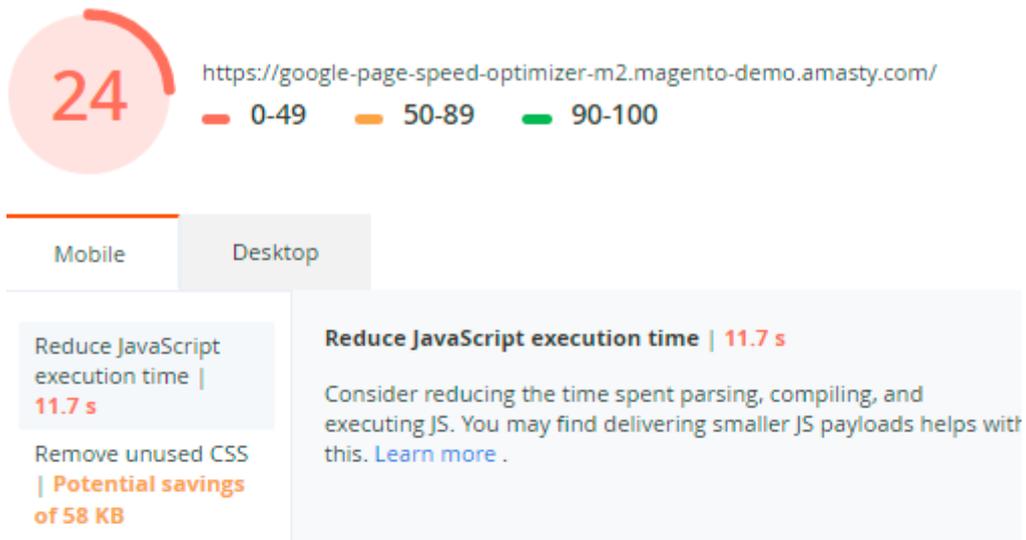
Settings 

Image Optimization 

Full Page Cache Warmer 

Before starting the optimization, check your website current Google score right from the admin panel using **Diagnostic** option.

You may see the desktop and mobile page speed scores separately. The informative block also contains Google recommendations.



General

Expand the **General** tab.

General



Enable Module
[store view]

Settings



Image Optimization



Enable Module - set to Yes to activate the module.

The extension also works on *Magento cloud*.

Settings

Expand this tab to reduce HTML, CSS and JS code size. The extension removes all unnecessary line breaks, tabs, comments and spaces.

HTML Minification

Settings



HTML:

Minify HTML
[store view]

'bin/magento setup:static-content:deploy'
command must be run in the console when
any of the settings in the field been changed.

Minify HTML - set the option to Yes to enable HTML minification.

When any of the settings in the field have been changed, execute the following command:

```
bin/magento setup:static-content:deploy
```

JavaScript

In this tab you can create an advanced bundle of JS using **Advanced JS Bundling** feature and JS Minification that will significantly reduce the page size in kilobytes and increase the page load speed. Google rank for both mobile and desktop will be improved.

You will need to use CLI commands to run through the optimization. Please make sure that you have access to the console and know how to do it.

Please create a full backup of your Magento instance and follow the instructions that are displayed in the **Run Optimization** section.

JavaScript:

Amasty JS Optimization <small>[store view]</small>	Enabled (Recommended) ▼	?
Is Magento Cloud <small>[global]</small>	Yes ▼	
Run Optimization <small>[global]</small>	Start	
Clear Bundle <small>[global]</small>	Clear Bundle	

To run the optimization, set the **Amasty JS Optimization** option to *Enabled*.

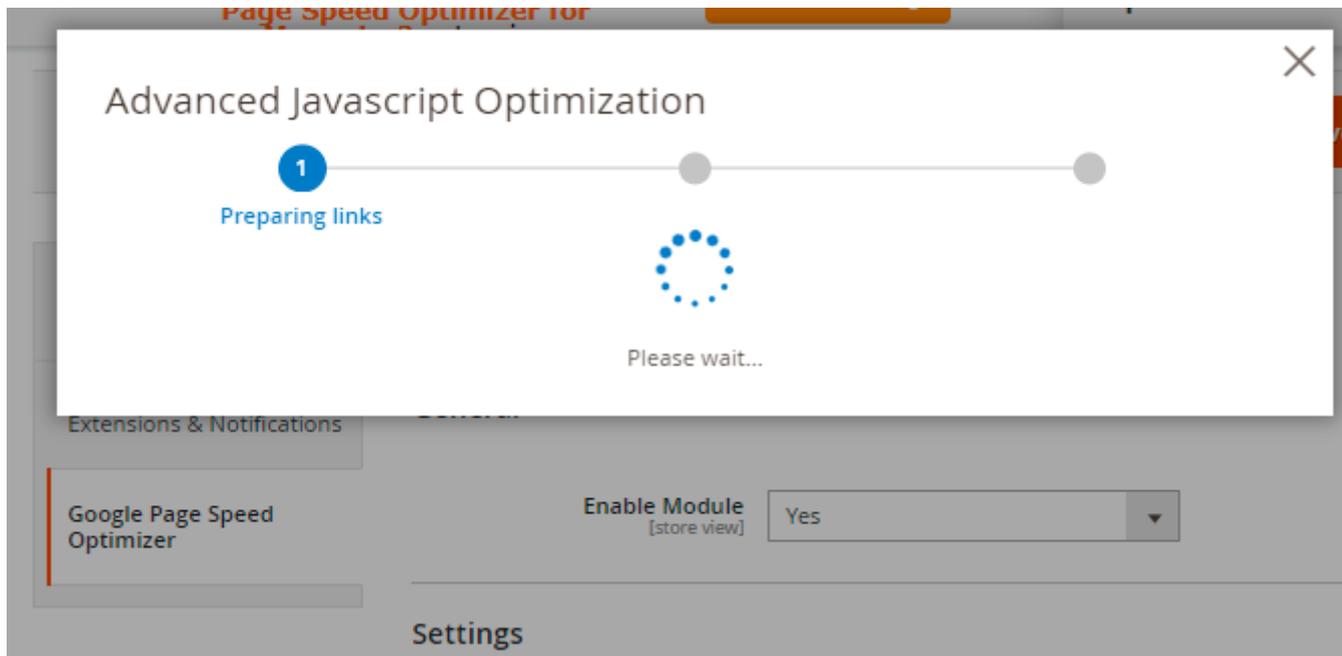
Is Magento Cloud - set to *Yes* if you use Magento Cloud hosting platform.

Click **Start**.

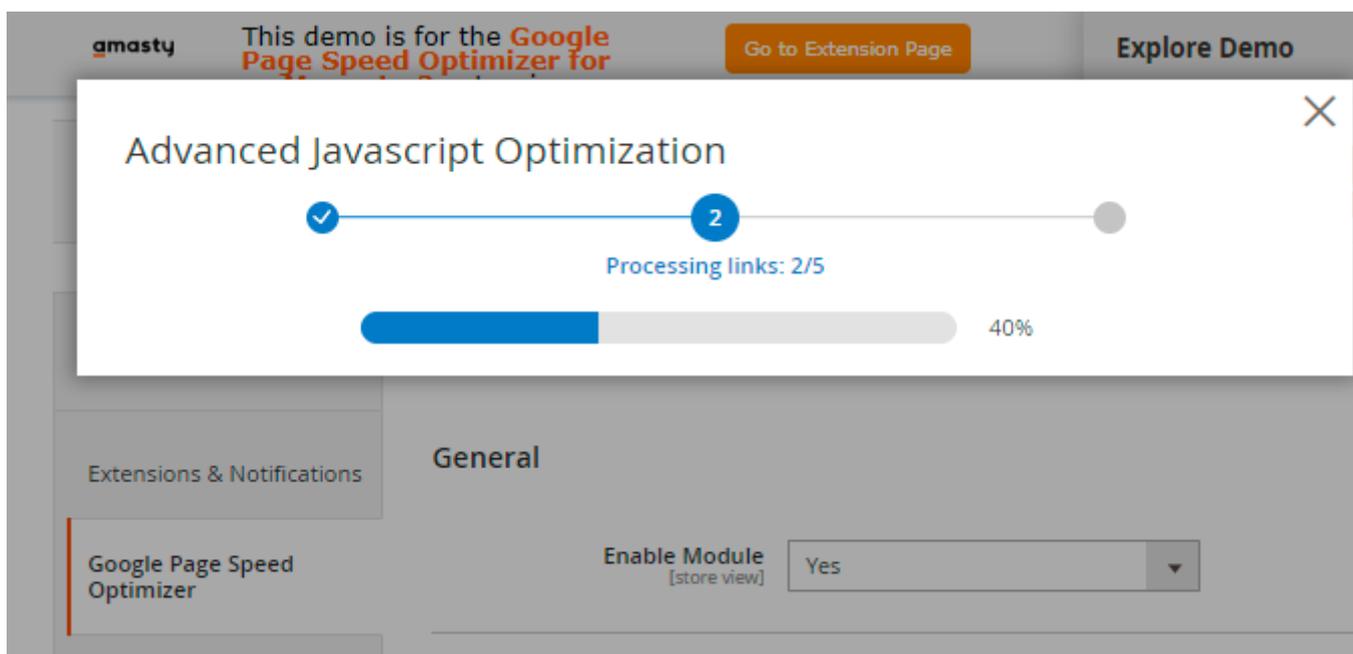
Easily **Clear Bundle** if needed.

Automatic Optimization

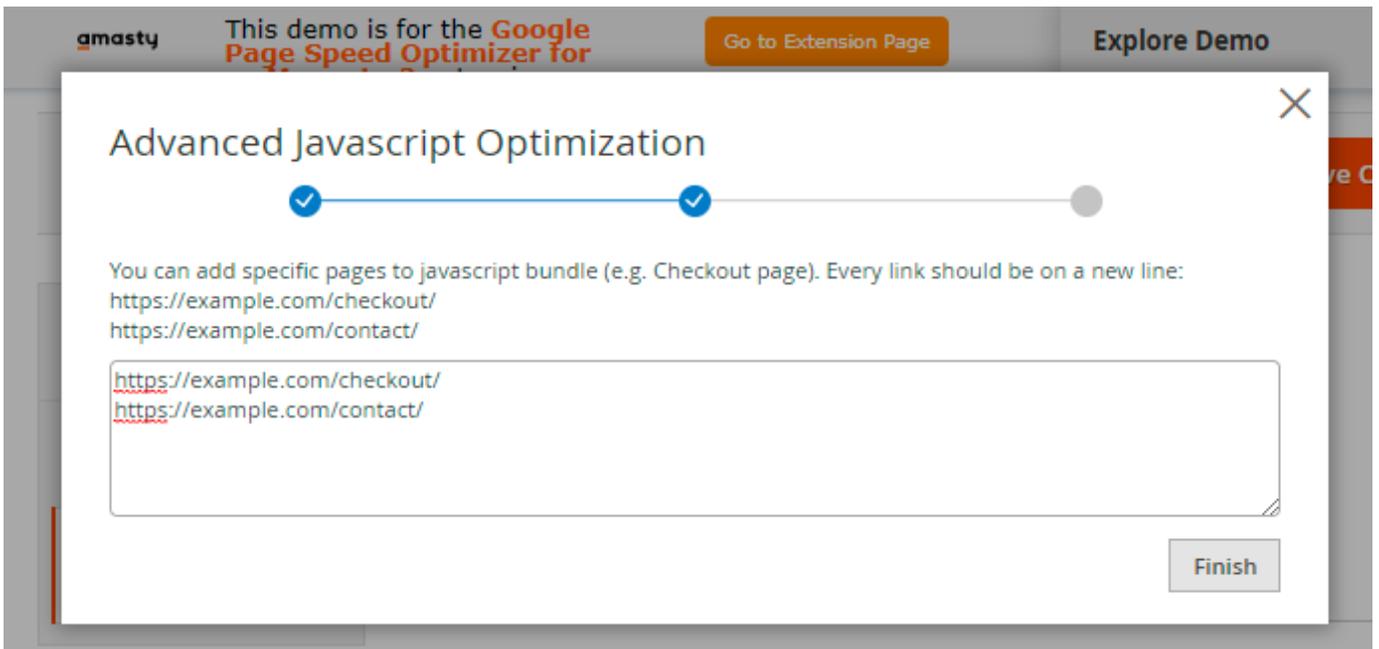
Step 1. After clicking the **Start** button, the extension starts preparing all the links for optimization.



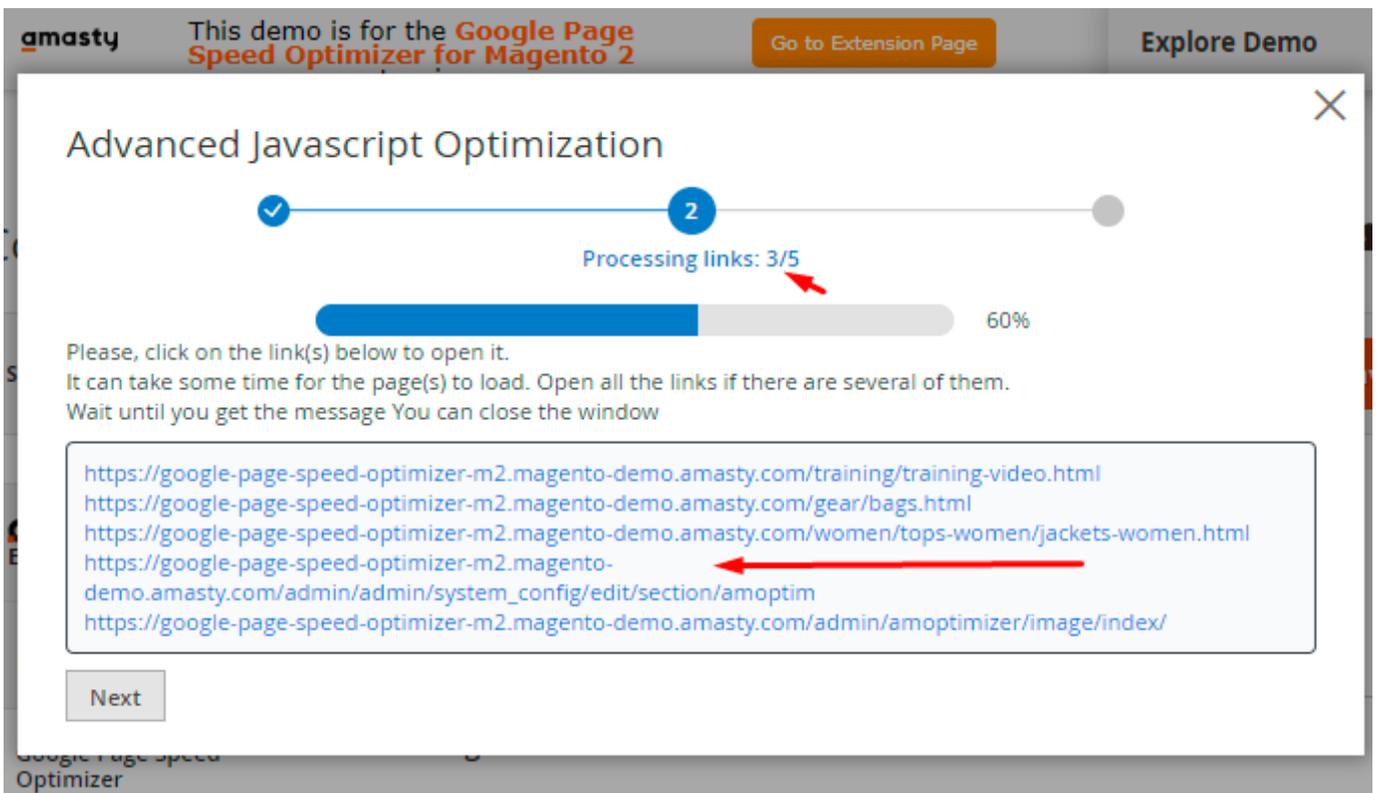
Step 2. Links processing progress is displayed on the bar.



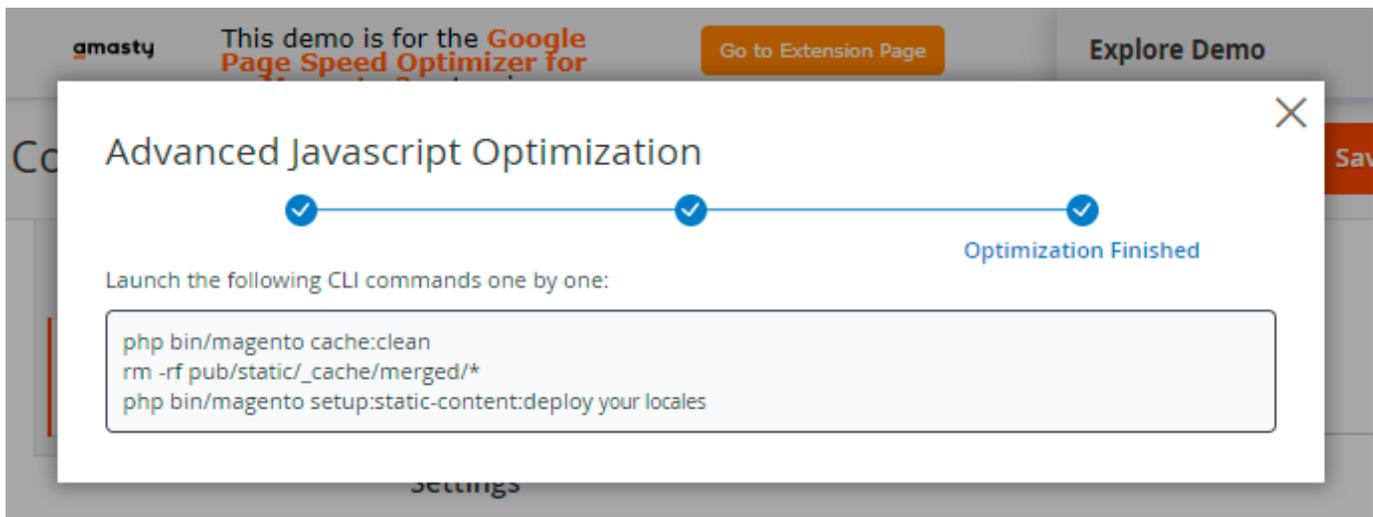
When all the links from the bundle are successfully processed, you can add specific pages to JavaScript bundle. Keep in mind that each link should be placed on a new line.



If any errors are found, the processing will be stopped and the error message will appear. You may check the particular link that caused a failure. For example, 3/5 links has been successfully processed and then the message appears. It means that the 4th link contains some problems. Easily click on it to check if it is valid or not and correct the link or page itself if needed.



Step 3. When optimization is finished, you will see the appropriate message and a list of commands that should be executed.



Additional JS Options

Move JavaScript To Page Bottom [store view] Yes (Recommended) ▼

Ignore URL List [store view]
home-page.html

Each URL or part of the URL from new line

Ignore Scripts that Contain [store view]
data-template="bundle-

Example: <script data-template="bundle-option" type="text/x-magento-template"><div><%- data._quantity_%> x <%- data._label_%></div></script> will be excluded.

Move JavaScript To Page Bottom option helps to load all the important content before loading JS. The functionality improves pages loading time.

Ignore URL List - provide URLs of the pages on which JS shouldn't be moved to bottom.

Ignore Scripts that Contain - you may also specify particular scripts and their parts if you want them to exclude from moving.

The sample of *Moving to Footer* feature:



If **Amasty JS Optimization** is *Disabled*, you can customize the way of code optimization. Besides Moving JS To Page Bottom, there are 3 additional options to reduce your JS code. You can also exclude particular URLs and scripts.

⊖ JavaScript:

Amasty JS Optimization [store view] Disabled

Merge JavaScript Files [store view] Yes

'bin/magento setup:static-content:deploy' command must be run in the console when any of the settings in the field been changed.

Enable JavaScript Bundling [store view] Yes

'bin/magento setup:static-content:deploy' command must be run in the console when any of the settings in the field been changed.

Minify JavaScript Files [store view] Yes

'bin/magento setup:static-content:deploy' command must be run in the console when any of the settings in the field been changed.

Merge JavaScript Files - select Yes to put all JS files into one. As a result, the number of queries will be reduced.

Enable JavaScript Bundling - enable this option to combine all JS files into few bundles and download them for each page. This helps to save time by reducing the number of server requests.

Notice that as browsers download the bundles synchronously, page rendering is blocked until all bundles finish downloading.

Minify JavaScript Files - set to Yes to enable JS minification.

Don't forget to run the command

```
bin/magento setup:static-content:deploy
```

in the console after any changes in this block of settings have been made.

This is how the bundling feature works:

Elements Console Sources **Network** Performance Memory Application Sec

View: [Icons] Group by frame Preserve log Disable cache Off

filter Hide data URLs **All** XHR JS CSS Img Media Font Doc WS Manifest

-] ame
-] select-bg.svg
-] tips.html
-] collection.html
-] query-builder.js
-] captcha.js
-] messages.js
-] spinner.js
-] resolver.js
-] adapter.js
-] refresh.js
-] authentication-popup.html
-] messages.html
-] captcha.html
-] gallery.png
-] mg03-br-0.jpg

207 requests | 1.1 MB transferred

BEFORE

Elements Console Sources **Network** Performance Memory Application Sec

View: [Icons] Group by frame Preserve log Disable cache Off

filter Hide data URLs **All** XHR JS CSS Img Media Font Doc WS Manifest

-] ame
-] submit-watch.html
-] e3ba05b55d9450782939b255da3e59ca.min.css
-] styles-l.min.css
-] styles.css
-] logo.svg
-] loader-1.gif
-] 8502028c7c49cc5469f4a7565c494173.min.js
-] Luma-lcons.woff2
-] fonts_e3ba05b55d9450782939b255da3e59ca.min.css
-] print.min.css
-] opensans-400.woff2
-] opensans-700.woff2
-] opensans-300.woff2
-] opensans-600.woff2
-] gallery.png
-] mg03-br-0.jpg

16 requests | 565 KB transferred | Finish: 6.12 s | DOMContentLoaded: 4.99 s | Load: 5.11 s

AFTER

CSS

CSS:

Merge CSS Files
[store view]

Yes (Recommended) ▼

'bin/magento setup:static-content:deploy' command must be run in the console when any of the settings in the field been changed.

Do not Merge CSS Files in Admin Area
[global]

Yes (Recommended) ▼



Exclude URLs from CSS Merge
[store view]

home-page.html

Each URL or part of the URL from new line

Minify CSS Files
[store view]

Yes (Recommended) ▼

'bin/magento setup:static-content:deploy' command must be run in the console when any of the settings in the field been changed.

Merging CSS Files works the same way as merging JS files.

Do not Merge CSS Files in Admin Area - forbid CSS files merging in the admin area. It won't affect your Goggle rating.

Exclude URLs from CSS Merge if needed.

Minifying CSS Files works the same way as minifying JS and HTML.

Defer Fonts Loading [store view] 

Fonts will be loaded later in a separate css file.
'bin/magento setup:static-content:deploy' command must be run in the console when any of the settings in the field been changed.

Do Not Defer Fonts That Contain [store view]

Example: Luma-Icons

Move Print CSS Files to Page Bottom [store view]  

If you set **Defer Fonts Loading** to Yes, the browser will load the page with system fonts and replace them with your fonts when they are loaded. It makes store pages load faster.

The following command must be run in the console when any of the settings in the field been changed:

```
bin/magento setup:static-content:deploy
```

You may exclude some fonts from deffering in the **Do Not Defer Fonts That Contain** option.

As in the JS tab, you can also **Move Print CSS Files to Page Bottom** to make your page render faster.

Flat Tables

Flat Tables allow storing all the information about a category or a product. Such tables help to shorten the query in the database.

⊖ Flat Tables:

Use Flat Catalog Category [global] Yes (Recommended) ▼

Use Flat Catalog Product [global] Yes (Recommended) ▼

Keep in mind that flat catalog usage is not recommended by Magento documentation. See more details [here](#).

Other Settings

In this tab you can enable some additional settings to make your webstore perform better.

⊖ Other Settings:

Sign Static Files [store view] Yes (Recommended) ▼

Asynchronous indexing [global] Enable (Recommended) ▼

Sign Static Files - activate the option to make the browser automatically update cached values once they are changed.

Asynchronous indexing - enable to distinguish traffic patterns on the database level to avoid conflicts between read and write operations. Order data is placed in a temporary storage and moved to the Order Management grid with no accidents.

Image Optimization

Expand this tab to configure the automatic image optimization using various flexible settings. The extension allows you to optimize JPEG, PNG and GIF images.

The correct functioning of the extension requires some special software to be installed: [JPEG optimization tool](#) / [PNG optimization tool](#) / [GIF optimization tool](#) / [WEBp tool](#). If you are not sure how to install the necessary software on the server correctly, please contact your administrator or Amasty

support team.

General Settings

Expand the **General Settings** tab to configure the main settings.

Image Optimization

General Settings

Optimize Images <small>[store view]</small>	Yes (Recommended) ▼	
	To optimize the existing images, please navigate to the 'Content>Google Page Speed Optimizer>Image Folder Optimization Settings' page and create new image folder settings.	
Enable Multi-Process Generation <small>[global]</small>	Yes ▼	?
Number of Parallels Processes <small>[global]</small>	4	?
Process Images per Request <small>[global]</small>	10	?

Set **Optimize images** option to **Yes** to compress images without quality loss and reduce web pages loading time.

Enable Multi-Process Generation - select Yes to speed up images optimization.

Multi-Process Optimization significantly boosts the image optimization speed. But it generates extra load to the server as well. We recommend first to run the test image optimization process with the 'multi-process' *ON* and monitor your server performance.

The 'Multi-Process Generation' feature needs the php extension '**pcntl**' to be installed on the server. If you enable the feature and no performance boost happens, please ask your hoster/system administrator to check if the 'pcntl' extension installed.

Number of Parallels Processes - specify the number of processes to be performed at once. The more parallel processes are set, the faster the image optimization process is, as well as the higher is the server load.

Process Images per Request - enter the number of images you want to be shown instantly, per one request.

You may also optimize images in bulk using a console command (here **4** is the number of threads):

```
bin/magento amasty:optimizer:optimize -j 4
```

User Agent

Expand the **User Agent** tab to adjust settings for image loading according to the customer's browser user agent.

⏪ User Agent

Use User Agent for Loading Images [store view]

Yes



Ignore Images that Contain [store view]

```
  

```

Example: `` OR `` will be excluded.

Set the **Use User Agent for Loading Images** to **Yes** to enable fetching appropriate image sizes based on the user agent data, obtained from the browser.

In the **Ignore Images that Contain** you can specify certain images to be excluded from the loading based on the user agent info. Such images will be delivered on the users' devices in the original size.

Lazy Load

This tab allows to choose the necessary load configuration: **Simple** (to configure the particular settings) or **Advanced** (to adjust the extended settings).

To deliver the media correctly, the **srcset** attribute will be deleted from Lazy Load images.

Simple

⏪ Lazy Load

Lazy Load Settings <small>[store view]</small>	Simple	?
Use Lazy Loading Images <small>[store view]</small>	Yes (Recommended)	
Lazy Load Script <small>[store view]</small>	jQuery Lazy Script jQuery Lazy Script Native JavaScript Lazy Script	
Preload Images <small>[store view]</small>	Yes (Recommended)	?
Desktop Preload Images Number <small>[store view]</small>	7	
Tablet Preload Images Number <small>[store view]</small>	5	
Mobile Preload Images Number <small>[store view]</small>	3	
Ignore Images that Contain <small>[store view]</small>	lazyOwl	?

Example: `` OR `` will be excluded.

Use Lazy Loading Images - set to Yes to make off-screen images load only when a customer scrolls to them. It prevents pages full of images from lagging.

Lazy Load Script - you can choose between 2 lazy load scripts: *jQuery Lazy Script* or *Native JS Lazy Script*. We recommend trying both of them to choose one that fits your system best.

You can decide whether to **Preload Images** or load them from scratch when they happen on screen. Decide how many images to load by setting **Preload Images Number**.

Desktop Preload Images Number - set the number of images to be pre-loaded on the page from the start. You can also assign a specific number of images for tablet devices with the **Tablet Preload Images Number** and for phones with the **Mobile Preload Images Number** fields.

Ignore Images That Contain - specify the images that should not be optimized.

For better compatibility with other extensions, the **img** tag which includes empty **src** will be ignored.

Advanced

Lazy Load Settings [store view] ▼

⌵ Lazy Load

⌵ General

⌵ Home Page

⌵ Category Pages

⌵ Product Pages

⌵ CMS Pages

All tabs of this block have the same structure as in the Simple mode, but the Advanced type is aimed at flexible customization of a specific type of site pages. Thus, you can select the most suitable type of optimization for each store page.

⏪ General

Use Lazy Loading Images <small>[store view]</small>	Yes (Recommended) ▼
Lazy Load Script <small>[store view]</small>	jQuery Lazy Script ▼
Preload Images <small>[store view]</small>	Yes (Recommended) ▼
Desktop Preload Images Number <small>[store view]</small>	7
Tablet Preload Images Number <small>[store view]</small>	5
Mobile Preload Images Number <small>[store view]</small>	3
Ignore Images that Contain <small>[store view]</small>	lazyOwl

Save the configuration.

When you're ready with the optimization don't forget to check your store with the [Google PageSpeed Insights tool](#) once again and compare the scores before and after optimization.

Automatic Optimization of the Newly Uploaded Images

Expand this tab to configure how all new images will be treated. This way, you can automate the optimization process.

⊖ Automatic Optimization of the Newly Uploaded Images

Automatically Optimize Newly
Uploaded Images in Catalog and
Wysiwyg folders
[store view]

Yes (Recommended) ▼ ?

All images uploaded to the catalog or Wysiwyg folder will be optimized automatically

JPEG Optimization Tool
[global]

jpegoptim 80% quality ▲

Do not Optimize
jpegoptim 100% quality
jpegoptim 90% quality
jpegoptim 80% quality

PNG Optimization Tool
[global]

optipng ▼

GIF Optimization Tool
[global]

gifscale ▼

Use Webp Image Format
[global]

No ▼ ?

When WebP format is used the extension will create a copy for every image in the WebP format. Please make sure that you have enough disc space on your server.

Create Images in Smaller Resolutions
[global]

Mobile ▼ ?
Tablet

The extension will create a copy of every image in suitable for mobile or tablet size. Please make sure that you have enough disc space on your server.

Resize Algorithm
[global]

Resize ▲ ?

Resize
Crop

Dump Original Images
[global]

Yes (Recommended) ▼

Original images will be stored in the 'pub/media/amasty/amoptimizer_dump' folder

Automatically Optimize Images in catalog or Wysiwyg folder - set to Yes to optimize all images uploaded to the catalog or Wysiwyg folder automatically.

JPEG optimization tool - set the way to optimize JPEG images:

- **Do not Optimize** - select this option to forbid optimization;
- **jpegoptim 100% quality / jpegoptim 90% quality / jpegoptim 80% quality** - choose the quality level you want to get after compression. See the examples of optimized images right below the setting.

PNG optimization tool - select optipng to optimize PNG images.

GIF optimization tool - choose gifscale to compress GIF files.

Use Webp image format - set to Yes to create a copy for every image in the WebP format.

Please make sure that you have enough disc space on your server. It is also necessary to install the Webp tool package on your server using the command

```
sudo apt-get install webp
```

For a specific setup with versioning, use [this guide](#).

Create Images in Smaller Resolutions - choose the type of copies to be created:

- **Mobile** - the extension will create copies suitable for mobile devices. All copies will be placed in the automatically created folder - **pub/media/amasty/amoptmobile**.
- **Tablet** - all the copies will be created specially for tablets and placed in **pub/media/amasty/amopttablet** folder.

Resize Algorithm - select the algorithm according to which the images will be compressed:

- **Crop** - cuts pictures by size according to the device used, but not proportional to the size of the picture itself. While the height of the image would be the same, in the result the cropped image will look like the part of the original picture.
- **Resize** - compresses the images in proportion to the picture itself according to the size of the field viewed. Simply it will be the same picture, but smaller in pixels.

Dump Original Images - set this option to Yes to store original images in the **pub/media/amasty/amoptimizer_dump** folder. It lets to collect all the original files in one place and easily update them in case the settings are changed.

Delete Previously Generated WebP & Resized Images

⌂ Delete Previously Generated WebP & Resized Images

Clear WebP Images Folder
[global]

Clear WebP Folder

Clear Mobile Images Folder
[global]

Clear Mobile Images Folder

Clear Tablet Images Folder
[global]

Clear Tablet Images Folder

Clear Images Folder - the *Clear WebP Images Folder / Clear Mobile Images Folder / Clear Tablet Images Folder* buttons delete the optimized files from the **pub / media / amasty / folder_name** folders, depending on which button is pressed. It is used if the previously selected optimization settings were inconvenient to use and there is a need to re-generate the queue and re-optimize the images.

Image Folder Optimization Settings

These settings will allow to precisely configure the optimization pattern for folders. To create new set of rules, please click on the **Add New Pattern** button.

Image Folder Optimization Settings

demouser ▾

Run Optimization

Add New Pattern

Filters

Default View ▾

Columns ▾

Actions ▾

2 records found

20 ▾

per page

<

1

of 1

>

<input type="checkbox"/>	ID	Folders	Title	Status	Action
<input type="checkbox"/>	1	wysiwyg	wysiwygFolders	Enabled	Edit
<input type="checkbox"/>	2	wysiwyg/mens	wysiwyg/mensFolder	Enabled	Edit

All configured patterns will be displayed in one handy grid. You can manage them any time you need just in a few clicks.

Run Optimization - click this button to optimize selected folders images.

To run optimization manually, execute the following command:

```
php bin/magento amasty:optimizer:optimize
```

Please mind that running optimization will make an additional load to the server.

Image Optimization settings

As you click the **Add New Pattern** button, a new **Image Optimization Settings** page will expand.

Image Optimization Settings

Enabled Yes

Title *

Folders for Optimisation *

amasty
 catalog
 customer
 downloadable
 import
 theme
 theme_customization
 wysiwyg

Done

Set the **Enabled** switch to **Yes** to allow this set of configurations.

Title - enter the title of the configuration set to easily distinguish numerous configs in the grid.

Folders for Optimisation - click on the folder name to include it to the list of folders that will be optimized.

Image Optimization Settings

Create Image for Mobile Yes 

Create Image for Tablet No 

The extension will create a copy of every image in suitable for mobile or tablet size. Please make sure that you have enough disc space on your server.

Resize Algorithm 

Create Webp Copy No 

When WebP format is used the extension will create a copy for every image in the WebP format. Please make sure that you have enough disc space on your server.

Create Image Dump Yes 

JPEG Optimization Tool 

PNG Optimization Tool

GIF Optimization Tool

Create Image for Mobile/Tablet - set the corresponding switch to **Yes** to automatically create appropriately sized images for various displays. All newly created copies will be saved on your website server, so check in advance if you have enough disc space.

Resize Algorithm - select the algorithm according to which the images will be compressed:

- **Crop** - cuts pictures by size according to the device used, but not proportional to the size of the picture itself. While the height of the image would be the same, in the result the cropped image will look like the part of the original picture.

- **Resize** - compresses the images in proportion to the picture itself according to the size of the field viewed. Simply it will be the same picture, but smaller in pixels.

Create Webp Copy - switch to **Yes** if you want to create a copy for every image in folders in the WebP format. All newly created copies will be saved on your website server, so check in advance if you have enough disc space.

Create Image Dump - set to **Yes** to store original images in the **pub/media/amasty/amoptimizer_dump** folder. It allows collect all the original files in one place and easily update them in case the settings are changed.

JPEG optimization tool - set the way to optimize JPEG images:

- **Do not Optimize** - select this option to forbid optimization;
- **jpegoptim 100% quality / jpegoptim 90% quality / jpegoptim 80% quality** - choose the quality level you want to get after compression. See the examples of optimized images right below the setting.

PNG optimization tool - select optipng to optimize PNG images.

GIF optimization tool - choose gifscale to compress GIF files.

Click on the **Save** button to preserve the settings.

AJAX Compatibility

You are able to add the compatibility with 3-rd party extensions which use AJAX. It can be done with the usage of the following sample:

```
<?php

class Test
{
    /**
     * @var \Magento\Framework\DataObjectFactory
     */
    private $dataObjectFactory;

    /**
     * @var \Magento\Framework\Event\ManagerInterface
     */
    private $eventManager;

    public function __construct(
        \Magento\Framework\DataObjectFactory $dataObjectFactory,
        \Magento\Framework\Event\ManagerInterface $eventManager
    ) {
        $this->dataObjectFactory = $dataObjectFactory;
    }
}
```

```
$this->eventManager = $eventManager;
}

public function execute()
{
    $someHtml = '<div>...</div>';
    $data = $this->dataObjectFactory->create(
        [
            'page' => $someHtml,
            //
            \Amasty\PageSpeedOptimizer\Model\Output\LazyLoadProcessor::PAGE_CONFIG keys
            'pageType' => 'catalog_category_view',
            // optional. Example of configs
            //
            \Amasty\PageSpeedOptimizer\Model\Output\LazyLoadProcessor::prepareLazyConfig
            'lazyConfig' => [

                ]
            ]
        );
    //if Amasty PageSpeedOptimizer is not installed output page will be
the same
    $this->eventManager->dispatch('amoptimizer_process_ajax_page',
['data' => $data]);
    $someHtml = $data->getData('page');

    return $someHtml;
}
}
```

Make your store pages load 85% faster for mobile users with the **Amasty AMP extension**, which is fully compatible with Google Page Speed Optimizer.

Find out how to install the Google Page Speed Optimizer via [Composer](#)

Rate the user guide

★☆☆☆☆ from 1 votes ([Details](#))

○ ○ ○ ○ ○

★☆☆☆☆ 1 visitor votes

★★☆☆☆ 0 visitor votes

★★★☆☆ 0 visitor votes

★★★★☆ 0 visitor votes

★★★★★ 0 visitor votes

From:

<https://amasty.com/docs/> - **Amasty Extensions FAQ**

Permanent link:

https://amasty.com/docs/doku.php?id=magento_2:google_page_speed_optimizer

Last update: **2020/05/15 09:28**

