JaJuMa

Page Preload

02/2023

Table of Contents

1. Introduction

With JaJuMa Page Preload pages can be preloaded in Magento 2, providing a better navigation user experience with faster subsequent page loads while customers are browsing the site.

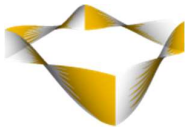
The script added by this function is a very lightweight (<2 KB minified/gzipped) drop-in solution that gets loaded after anything else and preloads pages only while customer's browser is idle to keep your site smooth and responsive.

Via configurations the extension allows to control the preload intensity as well as preloading scope and cache time-to-live (TTL) for preloaded content.

For explanations and recommendations on these options, please see below.

For further details please also see:

https://www.jajuma.de/en/jajuma-develop/extensions/page-preload-extension-for-magento-2



2. Installation

2.1. Extension Installation via composer

For installing the extension via composer, follow installation process as with any Magento Extension from Magento Marketplace.

2.2. Manual Extension Installation via FTP

For manual installation by FTP, please follow these steps:

Before Installing

1. We recommend you to duplicate your live store on a staging/test site and try installation on your staging/test site before deploying to your live store
2. Backup Magento files and the store database

Please Note: It's very important to backup all themes and extensions in Magento before installation, especially when you are working on a live server.
We strongly recommend you to do not skip this step.

Upload the Extension

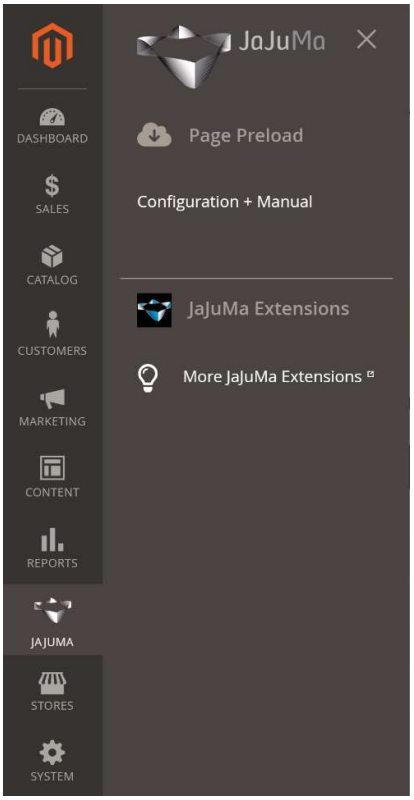
1. Log into your hosting space via a FTP client (e. g. FileZilla, WinSCP, cuteFtp)
2. Create Folder:
   <magentoroot>/app/code/Jajuma/PagePreload
3. Unzip extension package and upload files into:
   <magentoroot>/app/code/Jajuma/ PagePreload
4. Enter and run the following commands at the command line:

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

3. Configuration

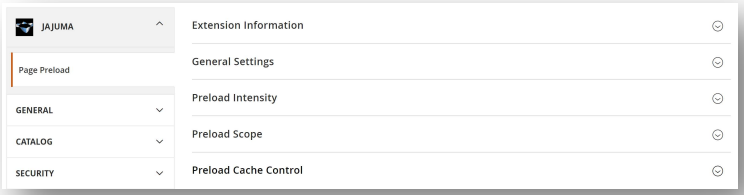
In Magento Backend see

JaJuMa -> Page Preload -> Configuration + Manual

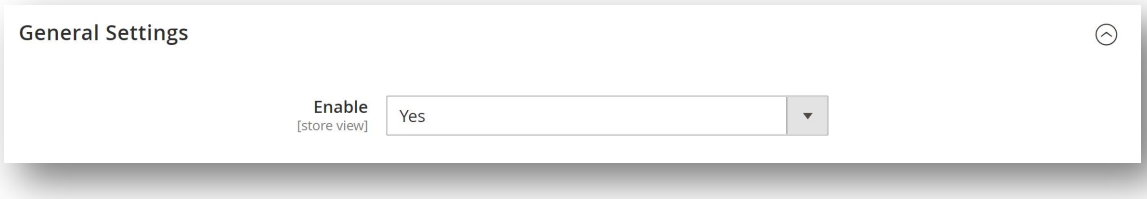


For JaJuMa Page Preload configuration with 4 sections:

- General Settings
- Preload Intensity
- Preload Scope
- Preload Cache Control



3.1. General Configurations



To enable Page Preload, select from Drop Down:

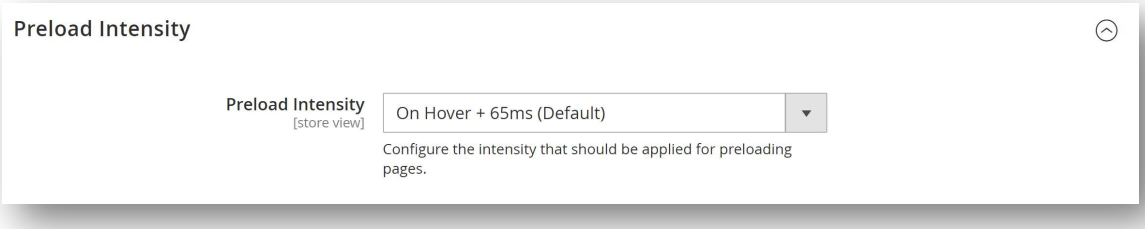
- Yes → Enable Extension
- No → Disable Extension

If enabled, the extension will preload pages as configured.

Note:

Choose and configure the Preload Intensity & Scope below carefully.
More intense preloading means a higher chance a page is preloaded before your customer visits the page, but also more additional load on your server.
Make sure your server has always enough resources to handle the additional load,
also during peak times
It is also strongly recommended to only use this function with FPC/Varnish enabled.

3.2. Preload Intensity



The preload intensity configuration mainly defines what triggers the preloading for links/pages. The extension supports triggering the preloading by user directly interacting with a link (on mousedown or on hover) or whenever a link is visible to the customer.

Following options are available:

1. **On Hover + 65ms (Default):**
   Links will be preloaded 65 ms after hovering a link and when a mobile user starts touching their display.
2. **On Hover + Custom Delay:**
   Links will be preloaded with custom delay as configured after hovering a link and when a mobile user starts touching their display.
   A lower value means more preload requests, a higher value will reduce the preload requests.
3. **On Mousedown (Desktop Only):**
   Links will be preloaded when the user starts pressing their mouse button, right before releasing it.
   Links are only preloaded on Desktop, but not on mobile devices.
4. **On Mousedown:**
   Links will be preloaded when the user starts pressing their mouse button, right before releasing it.
5. **When Visible (Mobile only):**
   Links will be preloaded as soon as they're visible on Mobile Devices.
   On Desktop Default will be used.
6. **When Visible:**
   Links will be preloaded as soon as they're visible. On both, Desktop and Mobile Devices.

See next sections for recommendations and more details on what these options mean.

3.2.1. The “Cautious” Preload Strategy

To avoid any additional load on your server, the **“On Mousedown”** options would be the way to go, either with preloading happening only on Desktop or on both, Desktop and Mobile.
This will create **no unused requests** while still **improving page loads by around 80ms** on average.

3.2.2. The “Balanced” Preload Strategy

For a balanced result between additional requests and improved page speed times, choose the **“On Hover”** options, either with default delay of 65ms or fine-tuned to your needs with a custom delay.
This may create **some unused requests** when a user hovers a link, but never clicks to visit this page.
But it provides **improved page load times by up to a few hundred milliseconds**.
Check the Click Speed Test Tool to see how much page load will be reduced for you.

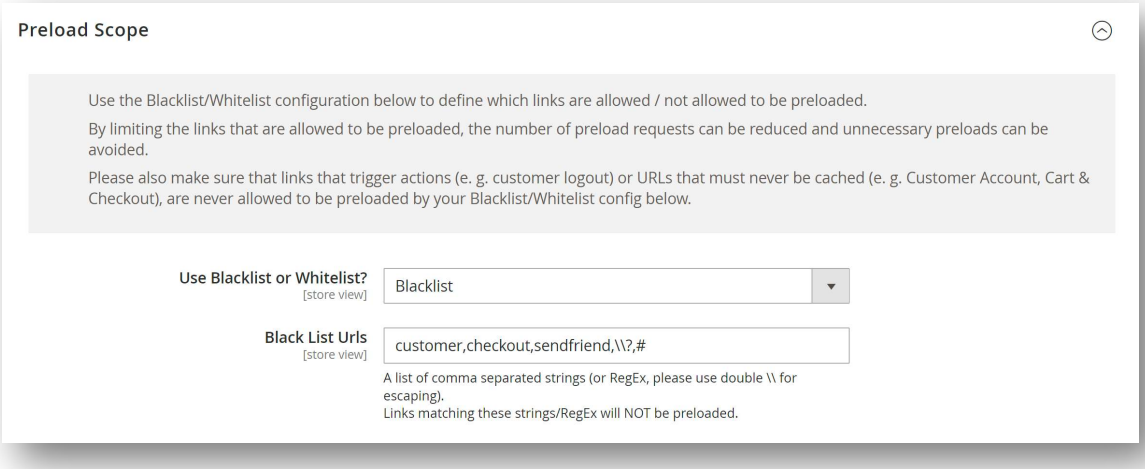
Note:
It is **recommended** to only use this option with Varnish/FPC (Full Page Cache) enabled and to narrow down the preload scope via Whitelist/Blacklist to only preload pages with a good chance to be visited by customer.

3.2.3. The “All-In” Preload Strategy

For the best result in terms of page speed improvement, choose the **“When Visible”** options, either only on Mobile (with fewer links in viewport due to smaller screens) and fallback to default on Desktop (= On Hover with default delay of 65ms) or on both, Desktop and Mobile.
Please be aware, since online stores usually have many links on one page, this may create **a lot(!) of unused requests** which may in the worst case overload your server.
However, with this preload strategy, when a customer clicks on a preloaded link**, page load will be literally “instant” with zero TTFB** (Time To First Byte).

Note:
It is **highly recommended** to only use this option with Varnish/FPC (Full Page Cache) enabled and to narrow down the preload scope via Whitelist/Blacklist to only preload pages with a good chance to be visited by customer.

3.3. Preload Scope



Use the Blacklist/Whitelist configuration to define which links are allowed / not allowed to be preloaded. Select from Drop Down if you want use Blacklist or Whitelist:

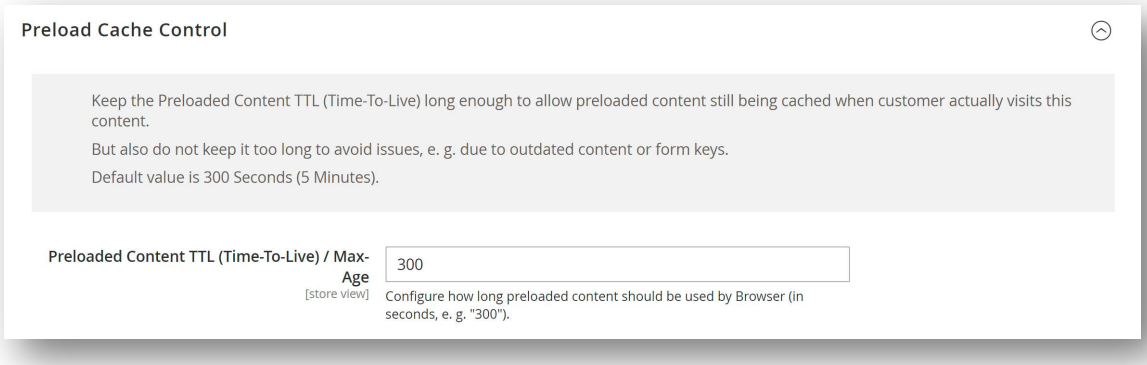
- Blacklist
  → Input a list of comma separated strings (or RegEx, please use double \\ for escaping). Links matching these strings/RegEx will NOT be preloaded.
- Whitelist
  → Input a list of comma separated strings (or RegEx, please use double \\ for escaping). Only Links matching these strings/RegEx will be preloaded.

By limiting the links that are allowed to be preloaded, the number of preload requests can be reduced and unnecessary preloads can be avoided.
Please also make sure that links that trigger actions (e. g. customer logout) or URLs that must never be cached (e. g. Customer Account, Cart & Checkout), are never allowed to be preloaded by your Blacklist/Whitelist config.

This is useful for e. g. following cases:

- Reduce preload request by avoiding unnecessary preloads
  (for example with layered navigation, that has many links, most of them likely not visited by a customer)
- Preload only important pages
  (for example focus on only category pages and/or product pages)
- Exclude pages from preload that should never be cached
  (for example Cart, Checkout, Customer Account)
- Exclude links from preloading that trigger an action
  (for example Customer Logout Link :-))

3.4. Preload Cache Control



Configure for how long the customer's browser should keep preloaded content in cache, as Time-To-Live in seconds (Default value is 300 seconds = 5 minutes)

Keep the Preloaded Content TTL (Time-To-Live) long enough to allow preloaded content still being cached when customer actually visits this content.

But also do not keep it too long to avoid issues, e. g. due to outdated content or form keys.

3.4.1. Preload Cache Control Configuration with Varnish

When using Varnish with your Magento store, please update your .vcl as follows to make the module work correctly:

```
sub vcl_deliver {

<!-- Add this line to the beginning of vcl_deliver function: -->
set resp.http.Prefetch-Cache-Control = resp.http.Cache-Control;

.....

<!-- Add these lines to the end of vcl_deliver function: -->
if (req.http.purpose ~ "prefetch"){
set resp.http.Cache-Control = resp.http.Prefetch-Cache-Control;
}
unset resp.http.Prefetch-Cache-Control;

}
```

4. Storefront

This module does not show in front end.

5. Testing Notes

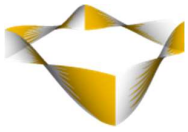
These would be the steps to test:

1. Install, configure and enable the extension, flush cache
2. Go to frontend
3. Open Dev Tools -> Network tab -> clear all records
4. Hover some link(s) with your mouse (or mousedown or scroll to get new links in viewport)
5. See the hovered links being loaded in dev tools network tab - note the “instantpage” script that is added by our extension as initiator for these requests
6. Now click one of these preloaded pages
7. See the request for this page in dev tools network tab and check columns size/time to see the browser is using the preloaded content and not loading the page again via network

!!! Please make sure the browser cache is not disabled while testing !!!

We also created a screencast for you for better understand the flow, please see

https://www.jajuma.de/en/jajuma-develop/extensions/page-preload-extension-for-magento-2



4. Support

Please feel free to contact JaJuMa support team via support@jajuma.de.
In case any additional information is required. We'd be more than happy to assist in setting up the extension.