

amasty

For more details see how [Full Page Cache Warmer for Magento 2](#) extension works.

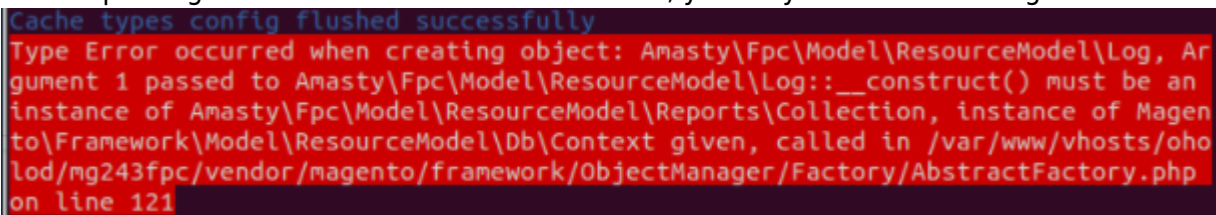
Guide for Full Page Cache Warmer for Magento 2

With *Cache Warmer for Magento 2* you'll be able to timely warm page cache according to your custom schedule to ensure fast page load and improve customer shopping experience.

- Use 5 algorithms to prioritize pages for cache warming
- Warm-up pages separately for mobile theme
- Exclude particular blocks from caching with Hole Punch
- Auto-update your cache after any catalog change
- Analyze cache warmer efficiency with detailed reports

Magento 2 Full Page Cache Warmer extension is compatible with [Hyvä themes](#). You can find the **amasty/module-fpc-hyva-compatibility** package for installing in composer suggest (Note: the compatibility is available as part of an active product subscription or Support Subscription). In [Full Page Cache Warmer FAQ](#) you can also find answers to the most popular questions about the extension functionality. For example, you can see how the extension works with [Varnish in this article](#).

When updating the extension to the latest version, you may face the following issue:



```
Cache types config flushed successfully
Type Error occurred when creating object: Amasty\Fpc\Model\ResourceModel\Log, Argument 1 passed to Amasty\Fpc\Model\ResourceModel\Log::__construct() must be an instance of Amasty\Fpc\Model\ResourceModel\Reports\Collection, instance of Magento\Framework\Model\ResourceModel\Db\Context given, called in /var/www/vhosts/ohod/mg243fpc/vendor/magento/framework/ObjectManager/Factory/AbstractFactory.php on line 121
```

To solve this problem, clean the **Generated** folder by running the command below in the Magento root:

```
rm -rf generated
```

General Settings

To configure general extension settings please go to **Stores → Configuration → Full Page Cache Warmer**.

General

Enable Extension

[global]

Yes

Auto-update Cache upon the Product Changes

[global]

Yes

Adds product pages urls to the Warmer Queue upon the product edit

Auto-generate Queue after Cache Flush

[global]

Yes

Set to yes to automatically generate warmer queue after cache flushing

Auto-generate Queue in Background

[global]

Yes

If enabled, warmer queue will be regenerated in background via cron and will not affect admin side performance.

Enable Cache Flushes Logging

[global]

Yes

Exclude Classes from Log

[global]

Class	Action
<div>SendResponseBefore</div>	<div></div>
<div>Add</div>	

Please specify classes that are not expected to be logged in cache flushes log.

Cache Flushes Log Auto-Cleaning

[global]

Yes

Auto-Cleaning Period in Days [global]	<input type="text" value="90"/>	?
Enable Customers' Activity Log [global]	<input type="text" value="Yes"/>	?
Activity Log Full-Cleaning [global]	<input type="text" value="Yes"/>	
Please note: if enabled the Activity Log will be completely cleared by Cron. If it used as Queue Generation Source we recommend you to configure the Log cleanup run after the Queue Generation Schedule run.		
Full-Cleaning Schedule [global]	<input type="text" value="0 0 1 * *"/>	
Enter cron expression		
* * * * *		
+---- Day of the Week (range: 0-6, 1 standing for Monday)		
+----- Month of the Year (range: 1-12)		
+----- Day of the Month (range: 1-31)		
+----- Hour (range: 0-23)		
+----- Minute (range: 0-59)		
Example: 0 0 1 * * - run once a month at midnight of the first day of the month		
Read more about cron expressions - here		
Performance Reports Data Auto-Cleaning [global]	<input type="text" value="Yes"/>	?
Auto-Cleaning Period in Days [global]	<input type="text" value="90"/>	?

Enable the extension - Set the option to 'yes' to activate the Cache Warmer. The option has a 'Global' scope so it is not possible to run the warmer on a certain web store/store view.

Auto-update Cache upon the Product Changes - Due to this option, URLs of updated pages will be automatically added to the warmer queue.

Auto-generate Queue after Cache Flush - Set to 'yes' to automatically generate warmer queue after cache flushing. That's a useful option, as the cache will be warmed up right away and there's no need to wait until this moment comes on schedule.

Please note that the setting works properly with **Built-in Cache** only.

Auto-generate Queue in Background - if enabled, warmer queue will be regenerated in background via cron and will not affect admin side performance.

Enable Cache Flushes Logging - with this option, the extension will set the record of any cache

flushes attempts in your Magento 2 instance. Learn more how to work with the log in the [Cache Flushes Log](#) part of the user manual.

Exclude Classes from Log - This setting is only available when *Enable Cache Flushes Logging* is set to Yes. You can specify the cache classes to exclude from the cache flushes log.

Keep in mind, that if **Auto-generate Queue after Cache Flush** option is enabled and any product is changed, the cache is flushed by default. When the cache is flushed, the queue is automatically regenerated, which can cause a longer product saving in case there are a lot of URLs. To avoid this issue, enable the **Auto-generate Queue in Background** option.

Cache Flushes Log Auto-Cleaning - Enable this option to clean the log automatically after a specified period.

Auto-Cleaning Period in Days - Provide a period after which the log should be cleaned.

Enable Customers' Activity Log - Set 'Yes' to let such features as FPC Warmer Reports and "Queue Generation Source" use the data from the log.

Activity Log Full-Cleaning - Select 'Yes' if you want to automatically clean all the records that are available at the time of cleaning by Cron.

Please note: if the Activity Log Full-cleaning feature is enabled, the Activity Log will be completely cleared by Cron. If it is used as Queue Generation Source we recommend you configure the Log cleanup run after the Queue Generation Schedule run.

Full-Cleaning Schedule - Enter the cron expression to set the schedule for automatic Activity log cleaning. *Example:* 0 0 1 * * - run once a month at midnight on the first day of the month. Read more about cron expressions [here](#).

Performance Reports Data Auto-Cleaning - Activate the feature to clean reports data automatically after a particular period.

Auto-Cleaning Period in Days - Specify a period after which reports should be cleaned.

Source and Priority

In the warmer settings tab you can configure warmer queue settings as well as specify a log size and exclude specific pages from cache warming.

It's possible to generate queues and warm cache via the command line. Use commands:

php bin/magento fpc:warmer:process

php bin/magento fpc:warmer:generate

Queue Generation Source Settings

Queue Generation Source - allows selecting any of 5 available generation sources:

- Pages Type
- Text file with one link per line
- Sitemap XML
- Sitemap XML and Text File together
- Customers Activity analysis
- Page Types and Text File Together

1. Page types





Source and Priority

Queue Generation Source
[global]

Pages Types

The queue for warming up will be created according to the chosen source

Pages and Priority
[global]

	Enabled	Page Type
	<input checked="" type="checkbox"/>	Home Page
	<input checked="" type="checkbox"/>	Product pages
	<input checked="" type="checkbox"/>	Category pages
	<input checked="" type="checkbox"/>	CMS pages

Page Types - If you select 'Page Types' as a queue generation source, you can choose which pages should be warmed. Drag and drop pages to prioritize them for the warming depending on their importance.

2. Sitemap XML

Source and Priority

Queue Generation Source [global]	<div>Sitemap XML ▼ ?</div> <p>The queue for warming up will be created according to the chosen source</p>
Sitemap XML File Path [store view]	<div>sitemaps/sitemap.xml</div> <p>Please, set the relative path to the Sitemap XML file on your server. Example: sitemaps/sitemap.xml</p>

Sitemap XML- The option requires a valid XML sitemap located in your Magento 2 root folder. Also, set the relative path to the Sitemap XML file on your server.

Automatically create XML sitemap with the XML Google® Sitemap extension with no efforts. Such XML sitemap will perfectly suite for creating the URL queue by Magento 2 Full Page Cache Warmer.

3. TXT file with one link per line

Source and Priority

Queue Generation Source [global]	<div>Text file with one link per line ▼ ?</div> <p>The queue for warming up will be created according to the chosen source</p>
Queue Links File Path [store view]	<div>var/tmp/queue_links.txt</div> <p>Please, set the relative path to the file on your server. Example: var/tmp/queue_links.txt</p>

Text file with one link per line - You can use a custom text file with a list of links to be crawled (one link per line). Specify the file path on your server.

4. Sitemap XML and Text File together

Source and Priority

Queue Generation Source [global]	<div>Sitemap XML and Text File together ▼ ?</div> <p>The queue for warming up will be created according to the chosen source</p>
Queue Links File Path [store view]	<div>var/tmp/queue_links.txt</div> <p>Please, set the relative path to the file on your server. Example: var/tmp/queue_links.txt</p>
Sitemap XML File Path [store view]	<div>sitemaps/sitemap.xml</div> <p>Please, set the relative path to the Sitemap XML file on your server. Example: sitemaps/sitemap.xml</p>

Sitemap XML and Text File together - Both the sitemap and the text file will be used for creating the URL queue for FPC warming. Specify the **Queue links file path** and the **Sitemap XML file path**.

5. Customer Activity Source - Create a warming queue based on the most popular pages of your website. Pages are ranked according to the number of visits. The most popular pages URLs will be at the beginning of the queue and will be warmed first, the least – at the end.

Source and Priority

Queue Generation Source [global]	<div>Customers Activity Source ▼ ?</div> <p>The queue for warming up will be created according to the chosen source</p>
--	---

5. Page Types and Text File Together - Using this option, you can create a file with the most valuable pages to warm first and then warm other pages according to their type and priority. Specify the **Queue Links File Path** and configure pages priority as in the settings above.

Source and Priority

Queue Generation Source

[global]

Page Types and Text File Together

▼

?

The queue for warming up will be created according to the chosen source

Queue Links File Path

[store view]

var/tmp/queue_links.txt

Please, set the relative path to the file on your server.
Example: var/tmp/queue_links.txt

Pages and Priority

[global]

Enabled

Page Type

?

<div><div></div><div></div><div></div><div></div></div> <div><input checked="" type="checkbox"/></div>	Category pages
<div><div></div><div></div><div></div><div></div></div> <div><input checked="" type="checkbox"/></div>	CMS pages
<div><div></div><div></div><div></div><div></div></div> <div><input checked="" type="checkbox"/></div>	Product pages
<div><div></div><div></div><div></div><div></div></div> <div><input checked="" type="checkbox"/></div>	Home Page

Performance Settings

Performance Settings

Enable Multi-Process Warming [global]	<input type="text" value="Yes"/> ?
Number of Parallel Processes [global]	<input type="text" value="5"/> ?
Queue Generation Schedule [global]	<input type="text" value="0 * /1 * * *"/> Enter cron expression * * * * * +---- Day of the Week (range: 0-6, 1 standing for Monday) +----- Month of the Year (range: 1-12) +----- Day of the Month (range: 1-31) +----- Hour (range: 0-23) +----- Minute (range: 0-59) Example: 0 */12 * * * - every twelve hours
Queue Max Size [global]	<input type="text" value="2000"/> Set the maximum number of pages in the Warmer Queue
Single Batch Size [global]	<input type="text" value="150"/> The number of page URLs that will be processed at a single cron run. Default value is: 50
Warmer Schedule [global]	<input type="text" value="*/5 * * * *"/> Enter cron expression * * * * * +---- Day of the Week (range: 0-6, 1 standing for Monday) +----- Month of the Year (range: 1-12) +----- Day of the Month (range: 1-31) +----- Hour (range: 0-23) +----- Minute (range: 0-59) Example: */5 * * * * - every five minutes
Delay between Requests [global]	<input type="text" value="300"/> Please make sure there is enough delay to keep the server load safe. Default value is: 300 (milliseconds)
Log Size [global]	<input type="text" value="50000"/> Set the maximum number of lines in the log table

Enable Multi-Process Warming - Enable multi-process warming to include several URLs per one crawling request. Multi-process warming significantly boosts the warming process speed. But it generates the extra load to the server as well. We recommend running the test warming while monitoring the server load first.

Number of Parallel Processes - Adjust the number of parallel processes for multi-process warming.

Queue Generation Schedule - It's possible to schedule warming queue generation according to any specific needs. Use the cron expression to schedule queue generation jobs, e.g. `0 */12 * *` to run cron every twelve hours. Read more about cron expressions [here](#).

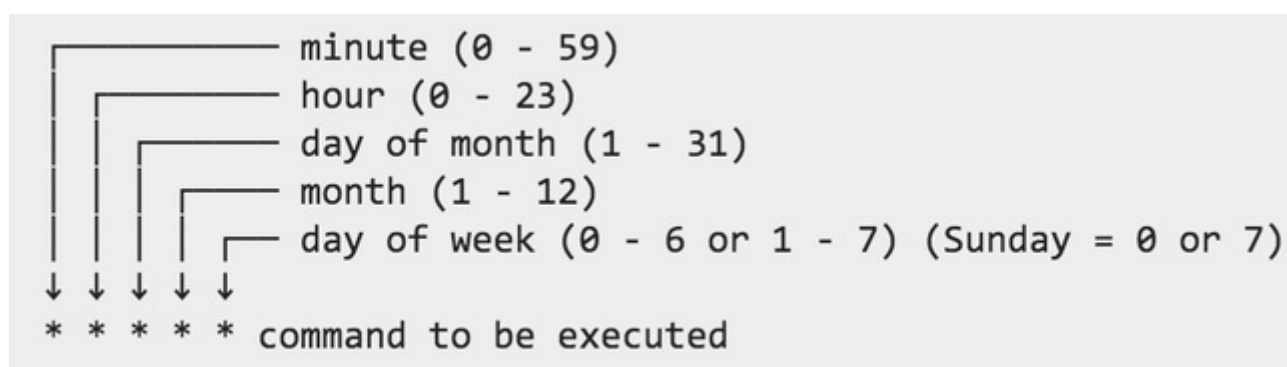
Queue Max Size - Set the maximum number of page URLs for the warming queue.

Single Batch Size - The module processes pages in batches. Specify the maximum number of pages that can be processed by cron within a single run.

Warmer Schedule - Schedule warmer jobs. For example, `*/5 * * * *` to run cron every five minutes.

Delay between requests - Please make sure there is enough delay to keep the server load safe. The default value is: 300 (milliseconds)

Log Size - Set the maximum number of lines in the log grid.



Page Combinations

Choose page combinations for cache warming based on Customer groups, Currencies, Stores.

Page Combinations

Warm Pages per Customer Groups
[global]

Yes



Customer Groups
[global]

NOT LOGGED IN

General

Retailer

Wholesale

"General" customer group is used by default even if nothing is selected

Warm Pages per Currencies
[global]

Yes



Currencies
[global]

US Dollar

Default currency is used by default even if nothing is selected

Warm Pages per Stores
[global]

No



You can also warm pages based on customer groups with the **Persistent Shopping Cart** enabled.

Warm Pages per Customer Groups
[global]

Yes



Customer Groups
[global]

NOT LOGGED IN

General

Retailer

Wholesale

General (Persistent)

Retailer (Persistent)

Wholesale (Persistent)

"General" customer group is used by default even if nothing is selected

Each link from the queue will be checked with all possible settings combinations based on the selected **Queue Source** and with **Default settings** (when none of the chosen settings are taken for

the request).

Take a look at the example below:

Let's say you've chosen the following settings:

«Customer Groups» = General

«Stores» = English, French

«Currency» = USD

There are 3 settings selected: one for a customer group, two for stores, one for currency. Besides, each of these

settings will not be specified for the request (e.g. set as Default).

Thus, here's the number of combinations for two links:

2 links * 2 customer groups * 3 stores * 2 currencies = 24 request variants

As a result, each link will be checked in the following 12 combinations:

1. Default + Default + Default : the page is requested with no parameters whatsoever.
2. General + English + USD : switched to General group + English is on + currency changed to USD
3. General + English + Default : switched to General group + English is on + no currency is set
4. General + French + USD : switched to General group + French is on + currency changed to USD
5. General + French + Default : switched to General group + French is on + no currency is set
6. General + Default + USD : switched to General group + English is on + currency changed to USD
7. General + Default + Default : switched to General group + English is on + no currency is set
8. Default + English + USD : no group is set + English is on + currency changed to USD
9. Default + English + Default : no group is set + English is on + no currency is set
10. Etc.

Warm-up mobile pages

Process Mobile View [global]

Yes

The feature should be used if a separate mobile theme is installed on your Magento

Mobile Agent [global]

Android | iPhone

This text will be sent as header on mobile request

User Agents [global]

iPhone | iPod | BlackBerry | Palm | Googlebot-Mobile | Mobile | mobi

Process Mobile View - Enable this option to warm-up mobile pages. Note, that the feature should be used if a separate mobile theme is installed on your Magento.

Mobile Agent - Specify mobile agents. The text, you'll enter in this field will be sent as a header on mobile requests.

User Agents - List user agents.

Exclude Pages

Exclude Pages [global]

Expression

Action

price=



Add


Add the regular expressions to exclude pages from warm up
\\? - pages with GET param

Exclude Pages - the option lets you exclude any pages from cache warming.

Exclude Blocks

FPC Hole Punch

Hole Punch
[global]

Template	Block class	Cms block ID	Action
<input type="text" value="/var/www/sourc"/>	<input type="text" value="Magento\Framework"/>	<input type="text"/>	
<input type="button" value="Add"/>			

Please refresh cache to apply changes. Please note, that not all blocks can be excluded due to Magento restrictions.

Hole Punch - the feature allows you to exclude certain page blocks from cache warning. It's not always possible to cache the whole page, as certain sections are dynamic and unique to each customer. With the Hole Punch feature, you can mark certain blocks of a page to be loaded each time and never cached while still caching the rest of the page. You can exclude static or dynamic blocks.

1. Static Blocks

To find the info on **Template** and **Block's class**, you'll need to complete few steps. First, go to the Stores→Settings→Configuration→Advanced→Developer.

In **Debug**, set both the **Enable Path Hints for Storefront** and **Add Block Type to Hints** as **Yes**. When you go on the storefront, the pages will look like in the example below. The Template data will be displayed on the left, and Block class — on the right correspondingly.

2. Dynamic Blocks

To exclude the dynamic Magento 2 block from the cache lifetime, you'll need to insert the Template and the Block's ID.

First, configure the **Debug** as it was described in the Static Blocks Configuration above and copy the Template info.

To obtain the Block's ID, simply insert the ID from the Content→Blocks.

ID	Title	Identifier	Store View	Status	Created	Modified	Action
1	Catalog Events Lister	catalog_events_lister	All Store Views	Enabled	Mar 27, 2019 5:42:57 AM	Mar 27, 2019 5:42:57 AM	Select
2	Footer Links Block	footer_links_block	All Store Views	Enabled	Mar 27, 2019 5:48:42 AM	Mar 27, 2019 5:48:42 AM	Select
3	Contact us info	contact-us-info	All Store Views	Enabled	Mar 27, 2019 5:48:42 AM	Mar 27, 2019 5:48:42 AM	Select
4	Sale Left Menu Block	sale-left-menu-block	All Store Views	Enabled	Mar 27, 2019 5:48:42 AM	Mar 27, 2019 5:48:42 AM	Select
5	Gear Left Menu Block	gear-left-menu-block	All Store Views	Enabled	Mar 27, 2019 5:48:42 AM	Mar 27, 2019 5:48:42 AM	Select
6	Men Left Menu Block	men-left-menu-block	All Store Views	Enabled	Mar 27, 2019 5:48:42 AM	Mar 27, 2019 5:48:42 AM	Select

Connection Settings

Connection Settings

Use HTTP Authentication

[global]

No

▼

Use when website is protected with .htaccess

Disable certificate verification

[global]

No

▼

Enable this only if FPC warmer doesn't work for https

Use HTTP Authentication - Use this option only if your website is protected with .htaccess

Disable certificate verification - Enable this feature only in case the FPC warmer doesn't work for https.

Debug Mode

IMPORTANT: The debug option works on the staging environment with native Magento cache. On the production environment with the activated Varnish Cache, the extension can't gather and process necessary data. Therefore, the Status & Hints are not available.

Debug

Show Cache Status [global]	<input type="text" value="Yes"/>	▼
Show Context Data [global]	<input type="text" value="Yes"/>	▼
Show Cache Status Only For [global]	<input type="text" value="117.155.268.32"/>	
	Enter IP addresses comma separated	

Show Context Data - if enabled, you can click the **Toggle Context Debug** in a status bar to see the current context parameters and those parameters with which the page is warmed up.

See the example:

Cart

IST | ADD TO COMPARE

Full Page Cache

CACHE MISS

Toggle Context Debug

Current context parameters:

vary = de2416d7360239c8de4b873d77609fdee411fa36

PERSISTENT = 1

current_currency = "USD"

customer_group = "1"

customer_logged_in = true

customer_segment = 1

store = "default"

website = "0"

This page was crawled with following context parameters:

vary = 810b259291774cd5ec4f95cce691bb86746fa202

current_currency = "EUR"

customer_group = "1"

customer_logged_in = true

store = "view2"

website = "0"

Show Cache Status Only For - Specify particular IP addresses fro which cache statuses will be displayed. Leave the field blank if you don't want to limit cache status display.

Lazy Load Compatibility Settings

If you have **Lazy Load** or **Google Page Speed Optimizer** installed, you can warm pages according to user agents with or without WebP support.

Lazy Load Compatibility Settings



User Agents
[global]

WebP Support
No WebP Support



Images
Resolution
[global]

Desktop
Tablet
Mobile

User Agents - Select the required user agents for warming.

WebP is a modern method of lossy and lossless compression that can be used on a large variety of photographic, translucent and graphical images found on the web. Webmasters interested in improving site performance can easily create optimized WebP alternatives for their current images, and serve them on a targeted basis to browsers that support WebP (for example, Google Chrome). The sample of the browser that doesn't support WebP now is Internet Explorer.

Images Resolution - Choose the resolutions to warm.

Warmer Queue

To open the Warmer Queue grid please go to **System** → **Warmer Queue**

Warmer Queue

Generate queue

Start Warm Up

Url	Rate
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/atomic-endurance-running-tee-v-neck.html	3
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/ryker-lumatech-trade-tee-crew-neck.html	3
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/helios-evercool-trade-tee.html	3
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/men/tops-men/tees-men/gobi-heattec-tee.html	3
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/men/tops-men/jackets-men/proteus-fitness-jackshirt.html	3
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/men/tops-men/jackets-men/montana-wind-jacket.html	3
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/collections/eco-friendly/jupiter-all-weather-trainer.html	3
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/men/tops-men/jackets-men/mars-heattech-trade-pullover.html	3
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/magento/men/tops-men/jackets-men/typhon-performance-fleece-lined-jacked.html	3

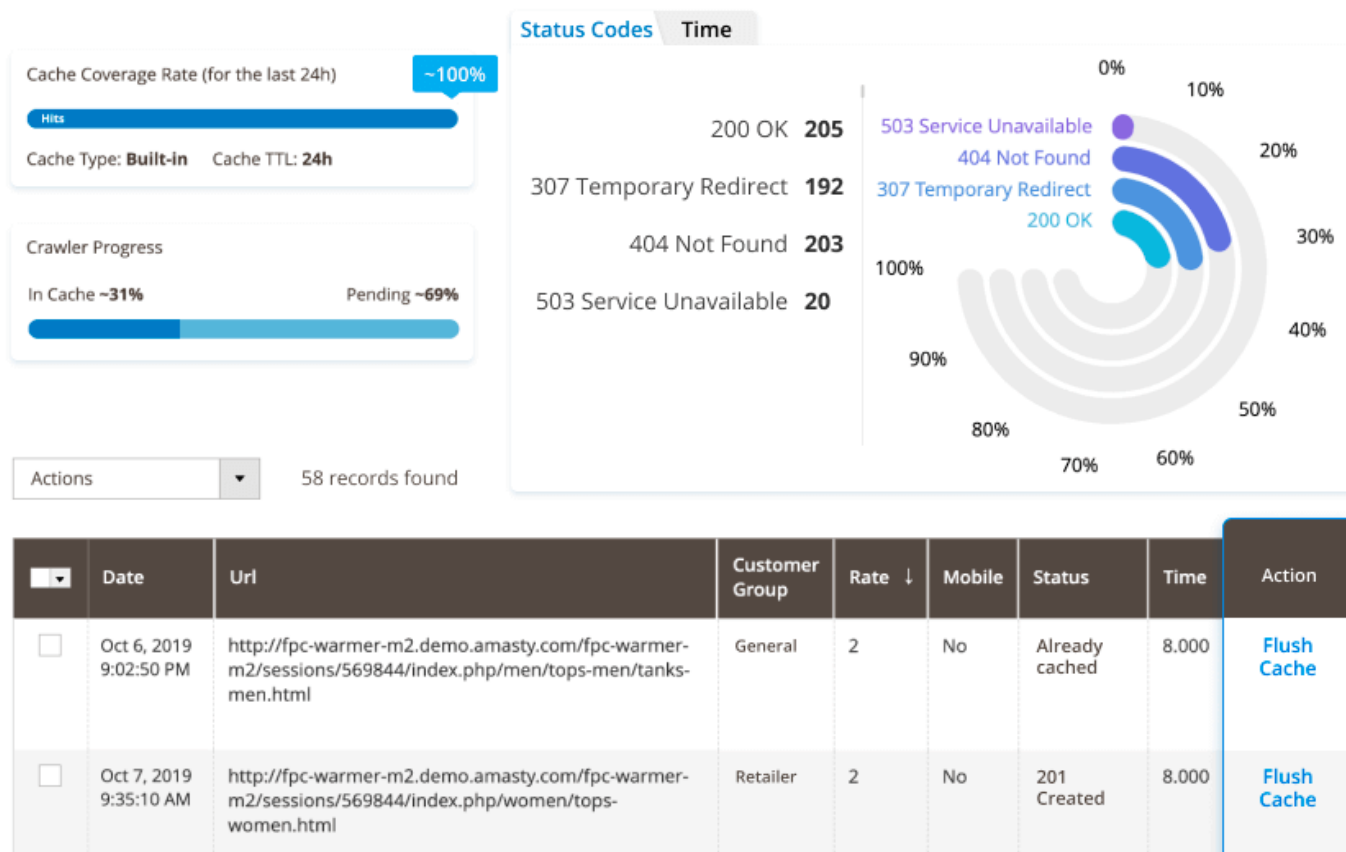
Here is the list of page URLs waiting to be processed.

You can generate a warmer queue manually at any time you need by clicking the '**Generate Queue**' button. It's also possible to manually trigger queue processing. To start processing URLs click the '**Start Warm Up**' button'.

Warmer Log

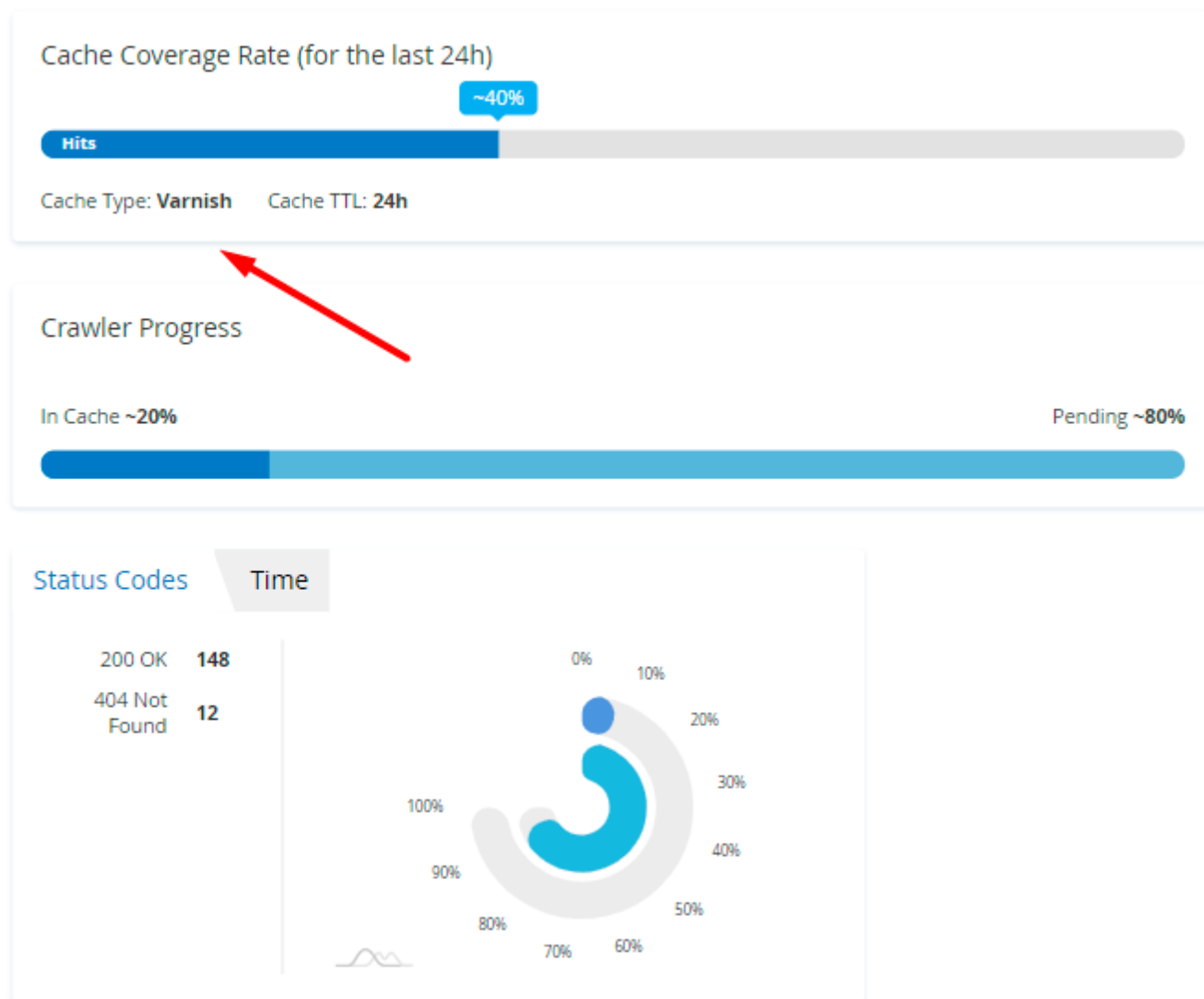
The log grid shows already warmed URLs. The log contains all crawled page combinations specified in 'Page Combinations' tab. Easily filter log data by any parameter you need.

Warmer Log



Flush Cache for particular URLs – If you change some product info, it's possible to flush cache for its product page only. It's useful as you won't need to warm cache from scratch for all the other URLs. You can do it by clicking the 'Flush Cache' button for each URL, or for the selected URLs via mass actions.

Also, the extension is compatible with Varnish. You can monitor and Cache Coverage Rate in Warmer Log when using Varnish cache and see the Performance Reports.



Keep in mind, that on the storefront Varnish cache status is always displayed as 'Miss'.

Warmer Activity Log

Here you can see the number of visits for each URL of the store. Each URL gets status based on the number of visits. It's used for 'Customer Activity Source' of queue generation.

Warmer Activity Log

[Clear Log](#)

Url	Number of Visits	Status	Mobile
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/index.php	1	200 OK	No
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/index.php/	15	200 OK	Yes
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/index.php/men.html	1	200 OK	No
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/index.php/men/tops-men/jackets-men.html	3	200 OK	Yes
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/index.php/montana-wind-jacket.html	2	200 OK	Yes
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/index.php/customer/account/create/	1	200 OK	No
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/about-us.html	1	404 Not Found	No
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/index.php/promotions/tees-all.html	1	200 OK	Yes
http://fpc-warmer-m2.demo.amasty.com/fpc-warmer-m2/index.php/women.html	1	301 Moved Permanently	No

Cache Flushes Log

When working with Magento 2 in your store, you may sometimes notice that the FPC cache is flushed due to some unknown event. It can be very challenging and resources-consuming to find the exact reason that causes the cache flushes.

If you have faced such an issue, start by setting the **Enable Cache Flushes Logging** to **Yes**. This config is located in **Stores → Configuration → Full Page Cache Warmer → General Settings**. Then, all actions that cause cache flushes will be logged.

Also, if you know that some methods are secure and work without causing cache flush issues, you can exclude such methods from the log. This way, the cache flushes log will be shorter, which makes it easier to identify the reason for unusual cache flushes.

You can access the log with details on cache flushes attempts by going to **System → Amasty FPC Warmer → Cache Flushes Log**.

Cache Flushes Log

 admin

[Clear Log](#)
 Filters

 Default View

 Columns

Actions

1 records found

20

per page

<

1

of 1

>

<input type="checkbox"/>	ID ↓	Details	Date	Backtrace
<input type="checkbox"/>	1	Tags: IMG_INFO Source: Magento Admin Mode: matchingAnyTag	2020-04-19 07:34:53	<pre> Amasty\Fpc\Logger\FlushesCache->clean() Magento\Framework\Cache\Frontend\Decorator\Bare->clean() Magento\Framework\Cache\Frontend\Decorator\Logger->clean() Magento\Framework\App\Cache->clean() Magento\Framework\App\Cache\Proxy->clean() Magento\Catalog\Model\Product\Image->clearImageInfoFromCache() Magento\Catalog\Model\Product\Image->clearCache() Magento\Backend\Controller\Adminhtml\Cache\CleanImages->execute() Magento\Backend\Controller\Adminhtml\Cache\CleanImages\Interceptor->execute() Magento\Framework\App\Action\Action->dispatch() Magento\Backend\App\AbstractAction->dispatch() Magento\Backend\Controller\Adminhtml\Cache\CleanImages\Interceptor->__callParent() Magento\Backend\Controller\Adminhtml\Cache\CleanImages\Interceptor->Magento\Framework\Interception\closure() Magento\Backend\App\Action\Plugin\Authentication->aroundDispatch() Amasty\DemoBackendAuth\Plugin\App\AuthPlugin->aroundDispatch() Magento\Backend\Controller\Adminhtml\Cache\CleanImages\Interceptor->Magento\Framework\Interception\closure() Magento\Backend\Controller\Adminhtml\Cache\CleanImages\Interceptor->__callPlugins() Magento\Backend\Controller\Adminhtml\Cache\CleanImages\Interceptor->dispatch() Magento\Framework\App\FrontController->processRequest() Magento\Framework\App\FrontController->dispatch() Magento\Framework\App\FrontController\Interceptor->__callParent() </pre>

The grid has the following data:

- ID - identification number of specific flush attempt;
- Source - cache flush source. It can be: CLI, Magento Admin, Storefront, Cron;
- Details - the extended information on what action caused the cache flush;
- Tags and Mode - the info on associated tags and mode. (For example, Tags: IMG_INFO, Mode: matchingAnyTag).
- Subject - the info on who or what initiated the cache flush. For Magento Admin source the value in the column will be 'Admin Test';
- Date - the date and time of cache flush event;
- Backtrace - it prints the function calls.

Cron Tasks List

Magento 2 Full Page Cache Warmer extension is preintegrated with Cron Tasks List to provide store owners with an opportunity to track and manage all cron tasks running in the website background.

To view all scheduled and executed cron tasks, go to **System → Cron Tasks List**

Run all cron tasks and generate their schedule by clicking the 'Run Cron' button. Also you can delete

separate tasks in bulk, apply filtering and sorting options when it is needed.

Cron Tasks List

Run Cron

Last Cron Activity: 20 seconds ago

Filters

Default View

Columns

Actions

652 records found

20 per page

1 of 33

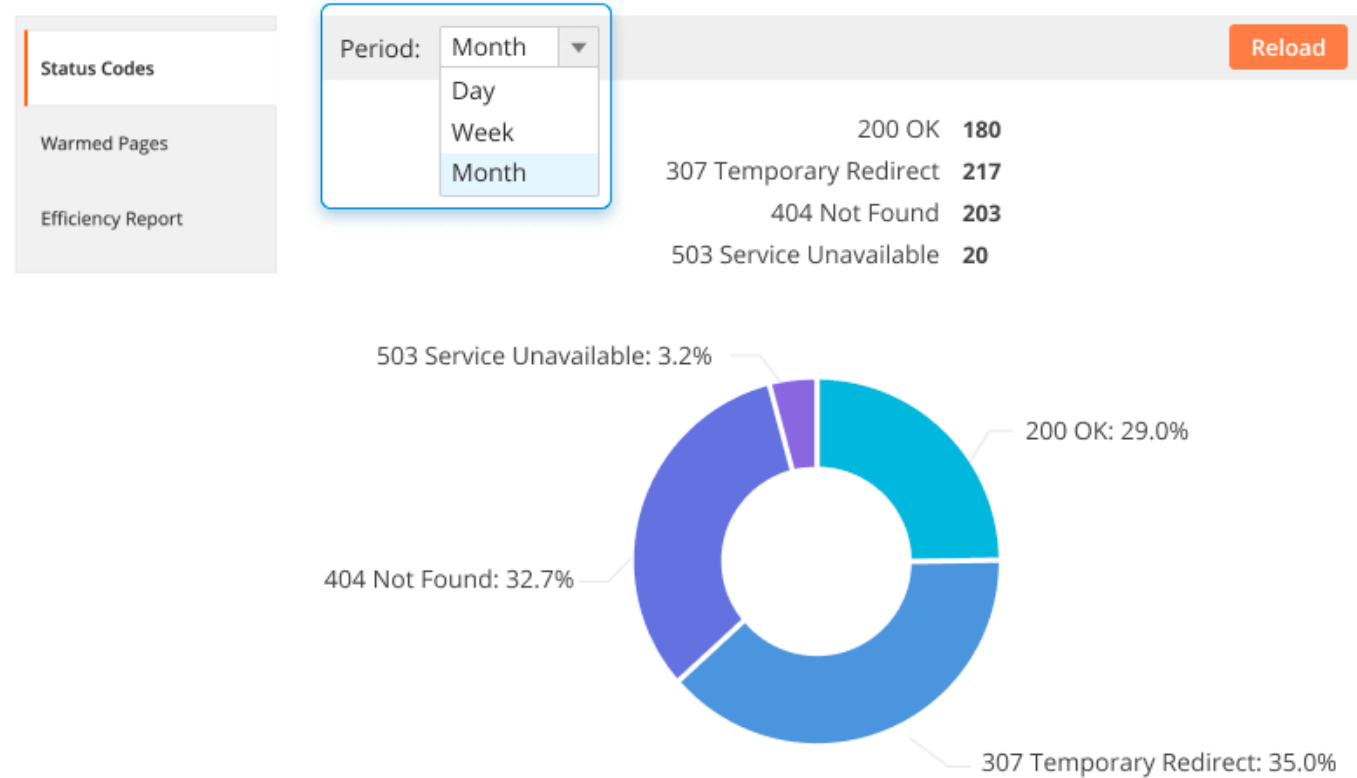
	ID	Job Code	Status ↑	Created At	Scheduled At	Executed At	Finished At
<input type="checkbox"/>	1038	sales_grid_order_shipment	SUCCESS	Mar 18, 2019 5:47:11 AM	Mar 18, 2019 5:47:00 AM	Mar 18, 2019 5:47:20 AM	Mar 18, 2019 5:47:20 AM
<input type="checkbox"/>	1058	sales_grid_creditmemo_async	SUCCESS	Mar 18, 2019 5:47:11 AM	Mar 18, 2019 5:47:00 AM	Mar 18, 2019 5:47:20 AM	Mar 18, 2019 5:47:20 AM
<input type="checkbox"/>	1078	sales_send_order_emails	SUCCESS	Mar 18, 2019 5:47:11 AM	Mar 18, 2019 5:47:00 AM	Mar 18, 2019 5:47:20 AM	Mar 18, 2019 5:47:20 AM
<input type="checkbox"/>	1098	sales_order_invoice_emails	SUCCESS	Mar 18, 2019 5:47:11 AM	Mar 18, 2019 5:47:00 AM	Mar 18, 2019 5:47:20 AM	Mar 18, 2019 5:47:20 AM
<input type="checkbox"/>	1118	sales_send_order_shipment	SUCCESS	Mar 18, 2019 5:47:11 AM	Mar 18, 2019 5:47:00 AM	Mar 18, 2019 5:47:20 AM	Mar 18, 2019 5:47:20 AM
<input type="checkbox"/>	1196	indexer_reindex_all_invalid	SUCCESS	Mar 18, 2019 5:47:13 AM	Mar 18, 2019 5:47:00 AM	Mar 18, 2019 5:47:21 AM	Mar 18, 2019 5:47:21 AM
<input type="checkbox"/>	1200	indexer_update_all_views	MISSED	Mar 18, 2019 5:47:13 AM	Mar 18, 2019 5:47:00 AM	Mar 18, 2019 5:47:21 AM	Mar 18, 2019 5:47:21 AM
<input type="checkbox"/>	1018	sales_grid_order_invoice_async	SUCCESS	Mar 18, 2019 5:47:11 AM	Mar 18, 2019 5:47:00 AM	Mar 18, 2019 5:47:20 AM	Mar 18, 2019 5:47:20 AM
<input type="checkbox"/>	1024	sales_grid_order_invoice_async	PENDING	Mar 18, 2019 5:47:11 AM	Mar 18, 2019 5:53:00 AM		
<input type="checkbox"/>	1025	sales_grid_order_invoice_async	PENDING	Mar 18, 2019 5:47:11 AM	Mar 18, 2019 5:54:00 AM		

Performance Reports

Status Codes

To take a look at visual reports go to **System → Amasty FPC → Performance Reports** .

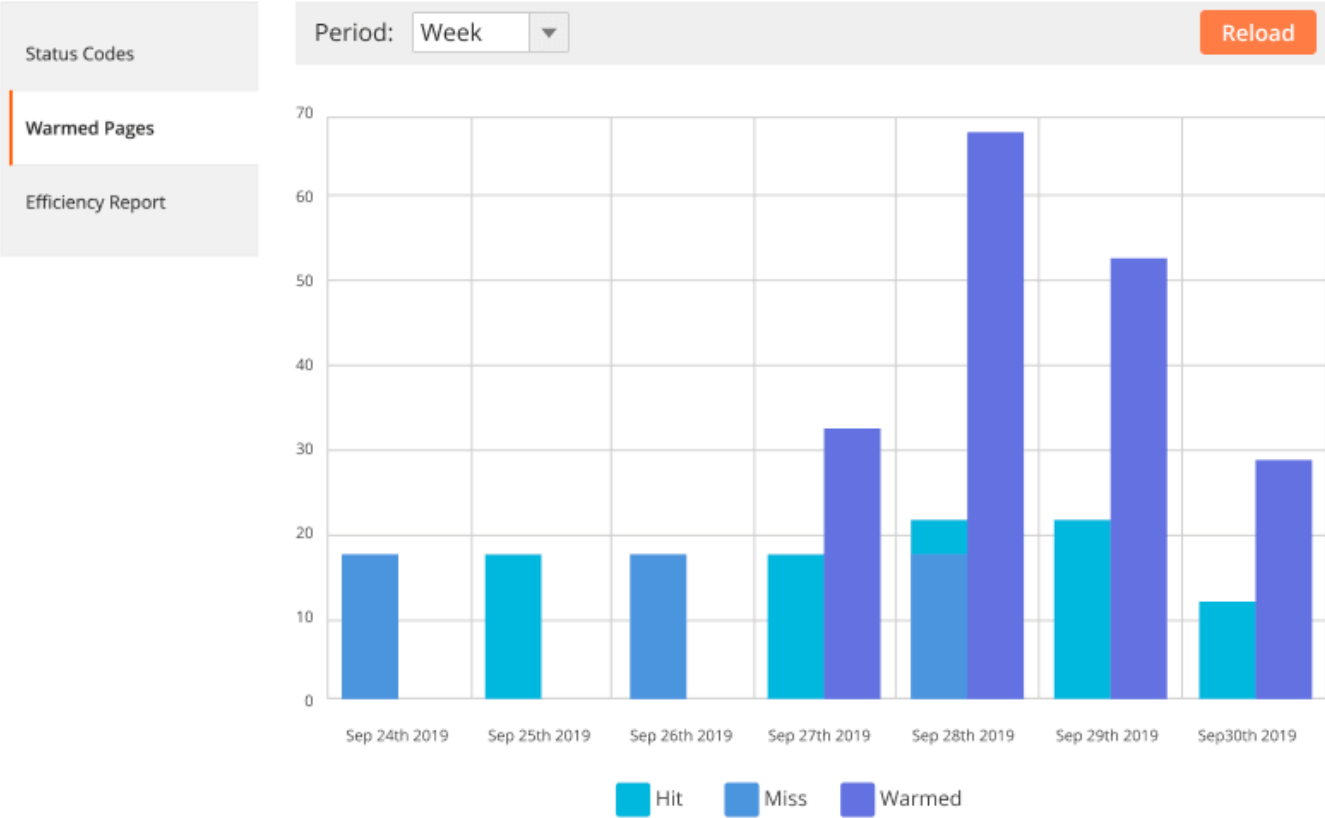
Reports



Status Codes - The stats report shows page statuses, as well as the number of pages with each particular status for all processed pages. It's highly useful for showing the percentage of errors and success loads.

Warmed Pages

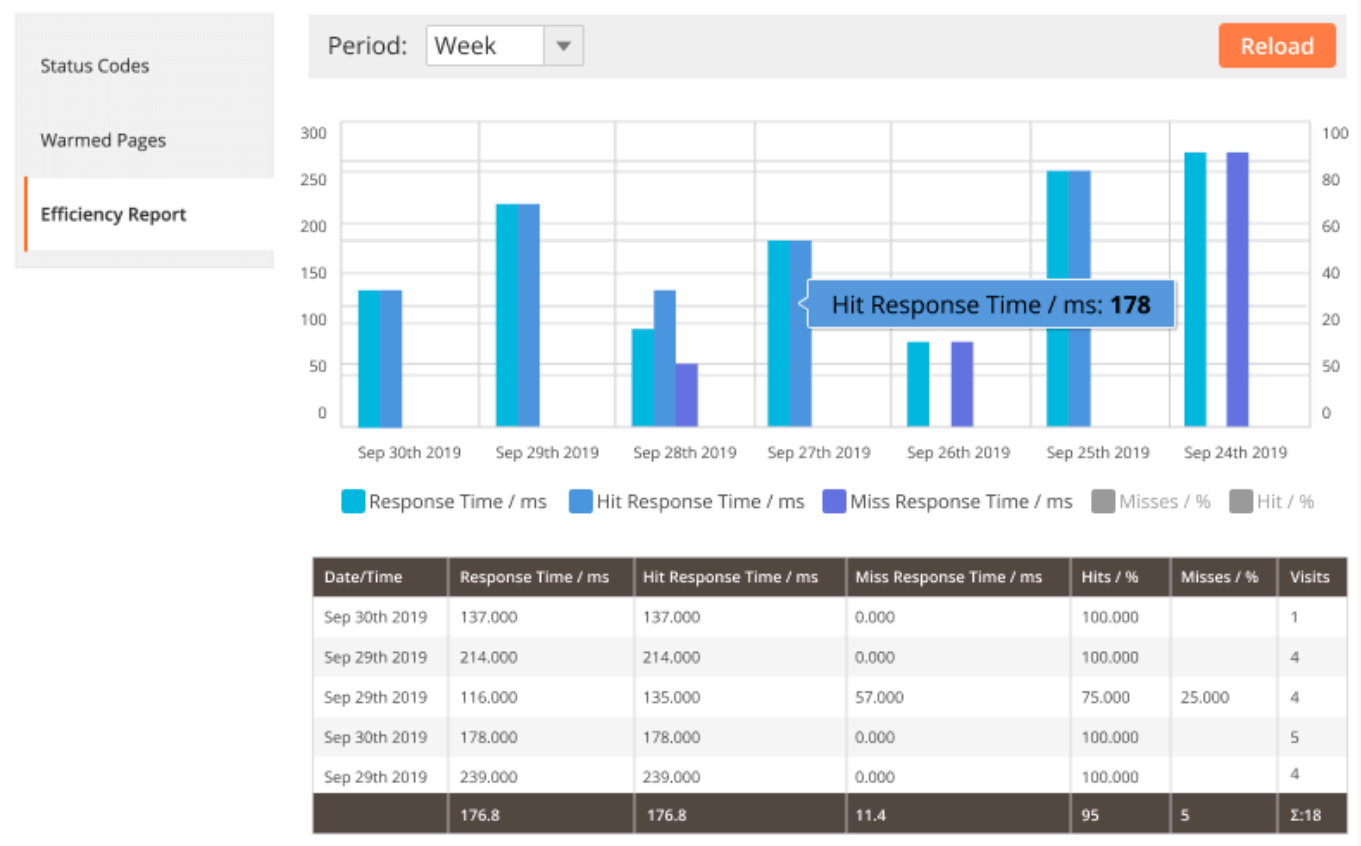
Reports



Warmed Pages - View Warmed Pages report to get the summary stats for requested pages for previous time periods. This way you can also understand when the crawler was most active.

Efficiency Report

Reports



Response Time / ms

Hit Response Time / ms

Miss Response Time / ms

Misses / %

Hit / %

Efficiency Report - The cache extension allows you to get a full picture of cache warmer efficiency. Choose the time period and see the percentage of hits and misses, track the performance by the crawler response time, etc.

Frontend Statuses

Check **page statuses** on the frontend.

Tiffany Fitness Tee

★★★★★ 3 Reviews [Add Your Review](#)

\$28.00 IN STOCK
SKU#: WS09

Color

☒ ☐ ☐

Size

☐ XS ☐ S ☐ M ☐ L ☐ XL

Qty

[Add to Cart](#)

Full Page Cache

CACHE HIT

Tiffany Fitness Tee

★★★★★ 3 Reviews [Add Your Review](#)

\$28.00 IN STOCK
SKU#: WS09

Color

☒ ☐ ☐

Size

☐ XS ☐ S ☐ M ☐ L ☐ XL

Qty

[Add to Cart](#)

Full Page Cache

CACHE MISS

The module displayed detailed information about **not cacheable blocks**.

[Home](#) > Search results for: 'tiffany'

Search results for: 'tiffany'

Compare Products



1 Item

Sort By

Relevance



You have no items to compare.

My Wish List

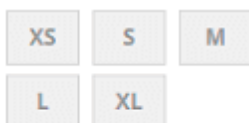
You have no items in your wish list.



Tiffany Fitness Tee

★★★★★ 3 reviews

\$28.00



Full Page Cache

IGNORED

Not cacheable blocks

search.result
Magento\CatalogSearch\Block\Result

search_result_list
Magento\CatalogSearch\Block\SearchResult\ListProduct

Find out how to install the **Full Page Cache Warmer** extension for Magento 2 via [Composer](#).

From:

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

Permanent link:

https://amasty.com/docs/doku.php?id=magento_2:full_page_cache_warmerLast update: **2023/05/18 14:00**