Magenerds Cache Warmer Documentation

www.magenerds.com

Magenerds_CacheWarmer – Documentation

Contents

1. Introduction
    - Feature List
    - Support and Contact
2. Installation
3. De-Installation
4. Configuration
5. First steps
6. Console usage

1. Introduction

The Cache Warmer extension was built in order to boost your site's performance. In high traffic peaks it is critical that all the single pages of your Magento store will be delivered very fast in order to reduce server load and in order to provide a good customer experience. A good way is to hold all the pages in the cache. With the Cache Warmer extension you can warm up the Magento cache beforehand so you are best prepared for your customers. The extension will crawl all product detail pages, all category pages and all cms pages of your store in order to have them all in the cache. In Magento standard after the cache has been deleted, one has to wait until all pages have been visited at least once. With this extension this is no longer needed as it does the job for you.

Feature List

- Let your top products/bestsellers be crawled with highest priority before other products
- Configure detailed priorities for the entities: **product detail pages**, **categories** and **cms pages**
- Configure the priority of every single store view. Maybe one store view is visited more often than the others?
- Scheduled and continuous crawling by Magento cron in the background
- Console commands available for: Starting, stoping, reseting the whole crawling process or individual entities in order to influence the crawling behavior
- Immediate crawling after cache deletion in order to warm it up quickly again
- Crawling of edited or newly created products, categories and cms pages
- Omits deactivated or not visible products and categories
- Configure the maximum server load possible before the crawling pauses in order to respect traffic peaks
- Configure the number of parallel crawling requests and bunch size per job
- Excellent monitoring of current crawling status in the backend:
    - Display the progress for products, categories and cms pages

- Start, stop or reset the whole crawling process or individual entities
- Crawling progress overview for every store view in order to know how when the cache has been warmed up
- Detailed overview of every single entity's url and it's crawling status

Support and Contact

Although we keep the quality of our extensions to a maximum there can be cases where bugs occur or you need further support beyond this documentation. We recommend that you read this documentation first, as it is a good start to get in touch with the functionality of the extension. If you still need our help, you can contact us via our bug report form at: http://www.magenerds.com/report-a-bug/

Please be aware that we only provide support for issues reported via this form. We will get in touch with you immediately after you report to us.

2. Installation

Before you start: Update your Magento file system as well as your Magento database. We are not responsible for any loss of data!

After you bought the extension from the Magento Marketplace you can download the extension from your Marketplace account or install it via the backend interface of your Magent installation. In the follwing we explain both ways:

- **Magento component manager**: That is the easiest way for you to install the extension. Just follow the official documentation at http://devdocs.magento.com/guides/v2.1/comp-mgr/module-man/compman-start.html. Be aware that in Magento 2.1 there is a core bug which does not display the component manager under certain conditions, follow the work arounds described here: https://github.com/magento/magento2/issues/4159 and https://github.com/magento/magento2/issues/5247
- **Download and install it manually:** You can read our installation guide for further instructions: http://doc.magenerds.com/extension-installation/ In detail you have

to do the following steps:

- Go to your Marketplace account and select purchases. Download the extension
- Get console access to your Magento server
- Create a new directory for the module:

  ```
  cd <Magento dir>/app/code && mkdir -p Magenerds/CacheWarmer
  ```
- Copy the content of the downloaded zip folder to this path:

  ```
  <Magento dir>/app/code/Magenerds/CacheWarmer/
  ```
- Run the following command to install the extension:

  ```
  <Magento dir>/bin/magento setup:upgrade
  ```
- Afterwards clean the cache:

  ```
  <Magento dir>/bin/magento cache:flush
  ```
- The extension is now ready to use

3. De-Installation

Before you start: Update your Magento file system as well as your Magento database. We are not responsible for any loss of data! If you installed the extension via the Magento component manager you should also deinstall it via the component manager. Just follow this documentation: http://devdocs.magento.com/guides/v2.1/comp-mgr/module-man/compman-uninst-final.html

If you installed the module manually, do the following steps:

- Get console access to your Magento server and run the following command in order to disable it first:

  ```
  <Magento dir>/bin/magento module:disable Magenerds_CacheWarmer
  ```
- Delete the module afterwards:

  ```
  cd <Magento dir>/app/code && rm -rf Magenerds/CacheWarmer
  ```
- Please be aware that the uninstall setup script will not be exeeuted if you deinstall it this way. It will only be executed if you uninstall it via the component

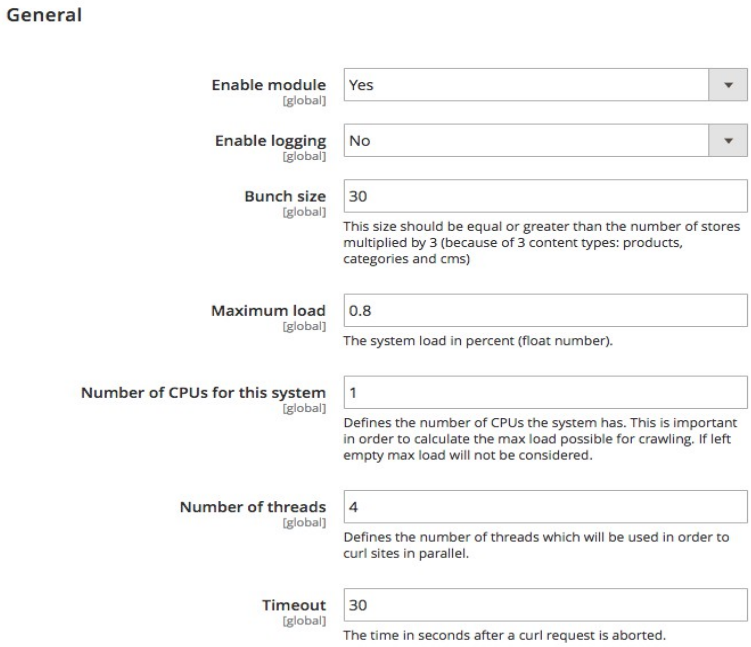
manager. You should manually delete the following tables which got created by the extension:

- magenerds_cachewarmer_progress
- magenerds_cachewarmer_status

4. Configuration

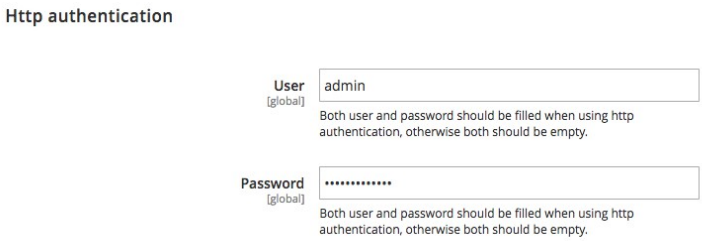
In order to run the extension properly you need to set up the Magento cronjob. There is a good and long documentation how to do this: http://devdocs.magento.com/guides/v2.0/config-guide/cli/config-cli-subcommands-cron.html.

This section is going through all possible system configurations and explains them to you. In the backend go to *Stores ▼ Configuration ▼ Magenerds ▼ CacheWarmer*.

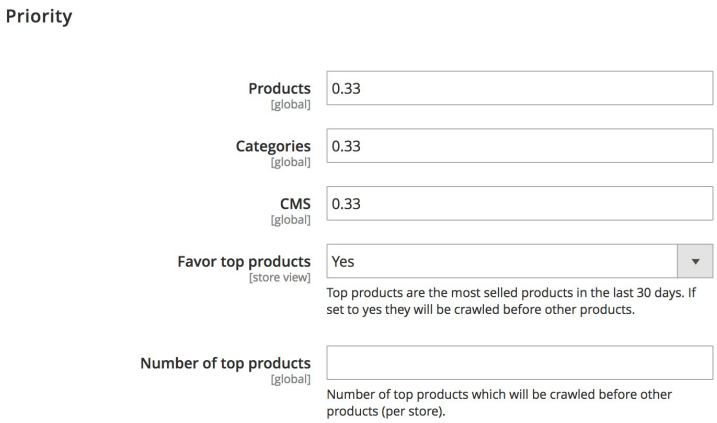


- **Enable module:** Enables or disables the crawling completely (Note: the module is still active, only its main functionality, the crawling will be disabled)
- **Enable logging:** Enables the logging for the module into the log file *var/log/magenerds-cachewarmer.log*

- **Bunch size:** Declares how many URLs will be in one bunch. One cron job run will crawl exactly one bunch. As the cron job runs every minute, you should find a bunch size which fits best in this one minute time frame.
- **Maximum load:** Defines the maximum load of the system until crawling is paused in order to respect traffic peaks and to not exceed the system's capacity.
- **Number of CPUs for this system:** Defines the number of cores of the system. This is necessary to calculate if the current system load is beyond the maximum configured load. The configured maximum load will be multiplied with the number of cores in order to find out if the current system load is higher.
- **Number of threads:** Defines the number of threads used in parallel for crawling. Ideally it should be equal to the number of cores available.
- **Timeout:** Defines the timeout after the crawling of one URL is stopped.



- **User:** The http authentication user
- **Password:** The http authentication password



- **Priority:** Defines the priorities in percent (1 is maximum) for the three entities products, categories and cms pages. If the sum of those three priorities is higher than 1, the priorities will be equally distributed again during crawling.
- **Favor top products:** If set to yes the top products will be favored and will be crawled before other products. Top products are the bestsellers of the last 30 days. If set to yes you can set the number of top products which get a higher priority.
- **Store priority:** You have to switch to store scope on the upper left dropdown in order to see the configuration field for the store priority. Again the sum of all stores' priorities may not exceed 1, otherwise the priorities will be distributed equally during crawling.

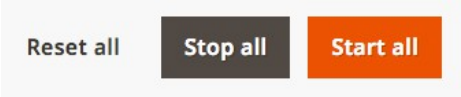
Attention: Magento has a system configuration which allows to configure if you want to use https for frontend urls. This setting can be found in the system configuration: *Stores ▼ Configuration ▼ Web ▼ Use Secure URLs on Storefront*. If this is set to yes, the cache warmer assumes that all frontend urls will be delivered via https by the web server. The cache warmer only crawls https urls. Be sure to setup a redirect from http to https requests in your web server's configuration in order to deliver the cached https urls and not the http urls.

5. First steps

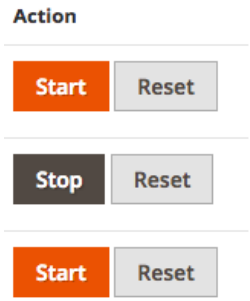
After successful installation and configuration of the extension it is now time to get in touch with its functionality. Before you start go to the extension's system configuration and check the default configuration. We have chosen a default configuration which may be not perfect for your system. For example: The default number of CPUs is 1 as it works on any system. This has impact on the load calculation and will be wrong if your system has i.e. 4 CPUs. Please refer to

chapter 4 in order to check your configuration and edit it accordingly to your needs.

After installing the extension the crawling process is disabled by default for all the three entities (product, category, cms-page). The first thing to do is then to reset the progress and to start the crawling process. In the backend go to *System* ⊽ *Cache Warmer*. There you have three buttons on the top right:



The same functionality is available for the three entities individually:

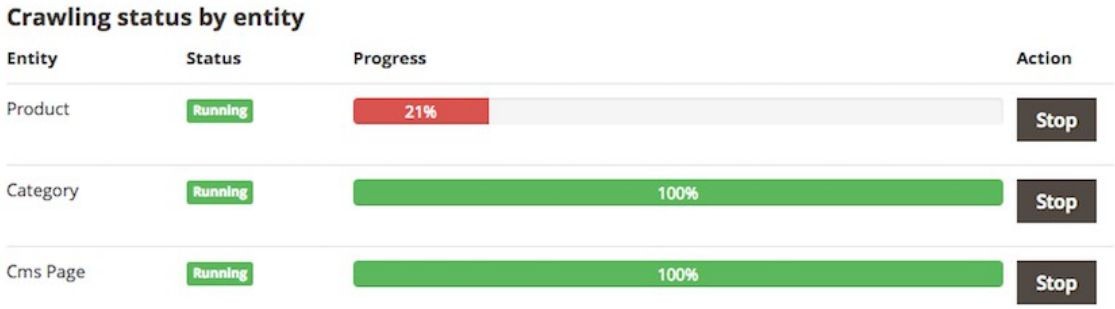


In the following the functionality is explained:

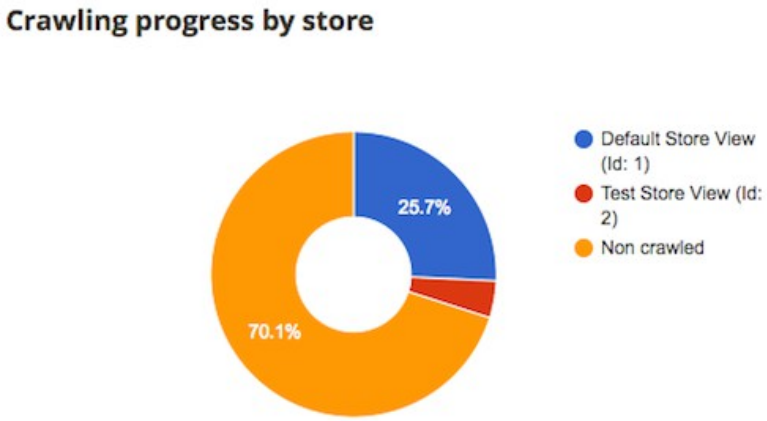
- **Reset:** Resets the complete crawling progress (or for the individual entity) in order to restart again. Everything crawled so far is still in the cache (unless you delete the cache explicitly) but the crawling resets and starts from the beginning.
- **Start:** Starts the crawling process (or for the individual entity).
- **Stop:** Stops or pauses the crawling process (or for the individual entity).

After resetting the progress and starting the crawling process, the cron job (runs every minute) will crawl the urls. You see the progress on the dashboard.

Crawling status by entity:



In the screen above you can see that of all product urls available, only 21% have been crawled. Of all category and of all cms page urls the crawling is finished. This is independently from the stores.
Crawling progress by store:



According to your configured store priority you can see the crawling progress by store. You also see the non crawled urls. This is independently from the three entity types.
The dashboard also gives you a quick overview of your current system configuration as well as the current system load and if the crawling has been paused due to the load:

System configuration (edit)

Configuration	Value
Bunch size	10
Maximum load	0.8
Number of CPUs	4
Number of threads	4
Timeout in seconds	30

Entity type	Priority
Product	0.33
Category	0.33
Cms Page	0.33

Store	Priority
Default Store View (Id: 1)	0.5
Test Store View (Id: 2)	0.5

Current system load	Max load allowed	Crawling status
0.81	3.2	Load okay (crawling possible)

You also get a table overview of all the urls which are considered for crawling. Remember: Entities which are not visible in the frontend or which are deactivated will not be crawled in order to save time. Therefore they don't appear in the table. You can filter the table to your needs, by crawling status, url, store or entity type.

Filters

56276 records found

20 per page 1 of 2814

Id	Entity type	Url	Store	Crawled
1	cms-page	https://magento2.dev/index.php/no-route	Default Store View (Id: 1)	Yes
2	cms-page	https://magento2.dev/index.php/home	Default Store View (Id: 1)	Yes
3	cms-page	https://magento2.dev/index.php/enable-cookies	Default Store View (Id: 1)	Yes
4	cms-page	https://magento2.dev/index.php/privacy-policy-cookie-restriction-mode	Default Store View (Id: 1)	Yes
5	cms-page	https://magento2.dev/index.php/service-unavailable	Test Store View (Id: 2)	Yes
6	cms-page	https://magento2.dev/index.php/private-sales	Test Store View (Id: 2)	Yes
7	cms-page	https://magento2.dev/index.php/reward-points	Test Store View (Id: 2)	Yes
8	cms-page	https://magento2.dev/index.php/agb	Default Store View (Id: 1)	Yes
9	cms-page	https://magento2.dev/index.php/widerrufsbelehrung	Default Store View (Id: 1)	Yes
10	cms-page	https://magento2.dev/index.php/impressum	Test Store View (Id: 2)	Yes
11	cms-page	https://magento2.dev/index.php/versandkosten	Test Store View (Id: 2)	Yes
12	product	https://magento2.dev/index.php/t-shirt.html	Test Store View (Id: 2)	No
13	product	https://magento2.dev/index.php/t-shirt-xs.html	Test Store View (Id: 2)	No

6. Console usage

There is a console command interface in order to manage the crawling state. If you are logged in to your Magento server switch to the Magento root installation. There you have the following commands:

- Set crawling status for all entities to start:

```
bin/magento cache:warmer start
```

- Set crawling status for all entities to stop:

  ```
  bin/magento cache:warmer stop
  ```

- Reset crawling progress for all entities:

  ```
  bin/magento cache:warmer reset
  ```

  This can also be done for only certain entities, for example:

- Set crawling status for products to start:

  ```
  bin/magento cache:warmer start --entity product
  ```

- Set crawling status for products and cms pages to start:

  ```
  bin/magento cache:warmer start --entity product --entity cms-page
  ```

- Reset crawling progress for categories and cms pages:

  ```
  bin/magento cache:warmer reset --entity category --entity cms-
  page
  ```