

CRON JOB RUNNER

USER GUIDE

Introduction

Cron Job Runner is an extension that is necessary for great performance of your web store. With it you will be able to regulate the accomplishment of repetitive tasks, their time, and frequency right from your admin panel. There is no need in using cPanel or adding crontab commands – you can set all the desired settings with this extension.

With Cron Job Runner you can clean temporary files, send emails, recalculate price, and update the sitemap as often as you wish without much efforts from your side. Just set the main settings once and the extension will work the way you want. All you have to do is check the logs from time to time and make some minor adjustments based on changes in your store.

Magento Compatibility

Magento Open Source 2.4.x

Adobe Commerce 2.4.x

PHP 7.4, 8.1, 8.2

Installation

1. Unpack the zip file provided into the root folder of your Magento 2 installation.
2. From a command line run:

```
bin/magento module:enable Neklo_Core
```

```
bin/magento module:enable Neklo_Scheduler
```

```
bin/magento setup:upgrade
```

If Magento is in Production mode, the following commands should also be run:

```
bin/magento setup:di:compile
```

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

Installation via Composer from Neklo

To install the available packages you need to add our repository to your composer configuration with the following command:

```
composer config repositories.neklo composer https://composer.neklo.com
```

Next install the package with the command:

```
composer require neklo/scheduler
```

When prompted, enter your Authentication details.

Composer authentication keys can be found in your account under My Downloadable Products:

```
bin/magento setup:upgrade
```

If Magento is in Production mode, the following commands should also be run:

```
bin/magento setup:di:compile
```

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

Installation via Composer from Adobe Marketplace

Install the package with the command:

```
composer require neklo/scheduler
```

```
bin/magento setup:upgrade
```

If Magento is in Production mode, the following commands should also be run:

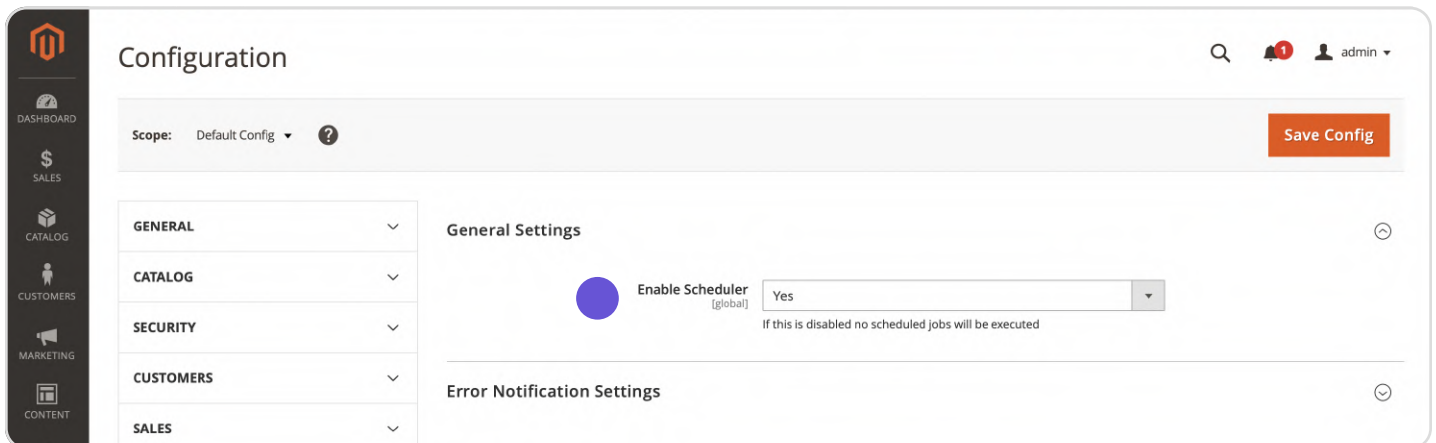
```
bin/magento setup:di:compile
```

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

Configuring the Extension

Go to **Stores > Settings > Configuration > Neklo tab > Cron Scheduler**.

General Settings



The screenshot shows the 'Configuration' page in the Neklo interface. On the left is a sidebar with icons for Dashboard, Sales, Catalog, Customers, Marketing, and Content. The main area is titled 'Configuration' and includes a search icon, a notification bell with '1', and a user profile 'admin'. Below the title bar, there's a 'Scope' dropdown set to 'Default Config' and a 'Save Config' button. The 'General Settings' section is expanded, showing a toggle for 'Enable Scheduler' (labeled '[global]') which is currently set to 'Yes'. A note below the toggle states: 'If this is disabled no scheduled jobs will be executed'. Below this, the 'Error Notification Settings' section is partially visible.

Enable Scheduler allows you to enable or disable Scheduler. If this is disabled no scheduled jobs will be executed.

Error Notification Settings

The screenshot shows the 'Configuration' page in the Neklo admin interface. The left sidebar contains navigation links: DASHBOARD, SALES, CATALOG, CUSTOMERS, MARKETING, CONTENT, REPORTS, STORES, SYSTEM, and FIND PARTNERS & EXTENSIONS. The main content area is titled 'Configuration' and includes a search icon, a notification bell, and a user profile 'admin'. Below this is a 'Scope' dropdown set to 'Default Config' and a 'Save Config' button. The 'GENERAL' settings are expanded, showing 'Enable Scheduler' set to 'Yes'. The 'Error Notification Settings' section is also expanded, showing three numbered steps: 1. Error Email Recipient (admin@example.com), 2. Error Email Sender (General Contact), and 3. Error Email Template (Error email (Default)).

1. **Error Email Recipient** – the destination of emails about errors.
2. **Error Email Sender** – from where the email about the error will be sent.
3. **Error Email Template** – the template of the error email. You can edit the template in **Marketing > Communications > Email Templates**. Click **Add New Template** and then choose the **Error Email Template**, load it, and **edit**.

The Error Email Template

Error with indexer_reindex_all_invalid Cron

```

Magento\Framework\Exception\LocalizedException: Could not ping search engine: No alive nodes found in your cluster in /var/www/eaptieka-lv/vendor/magento/module-elasticsearch/Model/Adapter/Elasticsearch.php:173 Stack trace: #0 /var/www/eaptieka-lv/vendor/magento/module-elasticsearch/Model/Indexer/IndexerHandler.php(157): Magento\Elasticsearch\Model\Adapter\Elasticsearch->ping() #1 /var/www/eaptieka-lv/vendor/magento/module-catalog-search/Model/Indexer/IndexerHandlerFactory.php(78): Magento\Elasticsearch\Model\Indexer\IndexerHandler->isAvailable() #2 /var/www/eaptieka-lv/vendor/magento/module-catalog-search/Model/Indexer/Fulltext/Action/Full.php(278): Magento\CatalogSearch\Model\Indexer\IndexerHandlerFactory->create() #3 /var/www/eaptieka-lv/vendor/magento/framework/ObjectManager/Factory/AbstractFactory.php(121): Magento\CatalogSearch\Model\Indexer\Fulltext\Action\Full->__construct() #4 /var/www/eaptieka-lv/vendor/magento/framework/ObjectManager/Factory/Dynamic/Developer.php(66): Magento\Framework\ObjectManager\Factory\AbstractFactory->createObject() #5 /var/www/eaptieka-lv/vendor/magento/framework/ObjectManager/ObjectManager.php(56): Magento\Framework\ObjectManager\Factory\Dynamic\Developer->create() #6 /var/www/eaptieka-lv/generated/code/Magento/CatalogSearch/Model/Indexer/Fulltext/Action/FullFactory.php(43): Magento\CatalogSearch\Model\Indexer\Fulltext->__construct() #7 /var/www/eaptieka-lv/vendor/magento/module-catalog-search/Model/Indexer/Fulltext.php(129): Magento\CatalogSearch\Model\Indexer\Fulltext\Action\FullFactory->create() #8 /var/www/eaptieka-lv/generated/code/Magento/CatalogSearch/Model/Indexer/Fulltext/Interceptor.php(14): Magento\CatalogSearch\Model\Indexer\Fulltext\Interceptor->__construct() #9 /var/www/eaptieka-lv/vendor/magento/framework/ObjectManager/Factory/AbstractFactory.php(121): Magento\CatalogSearch\Model\Indexer\Fulltext\Interceptor->__construct() #10 /var/www/eaptieka-lv/vendor/magento/framework/ObjectManager/Factory/Dynamic/Developer.php(66): Magento\Framework\ObjectManager\Factory\AbstractFactory->createObject() #11 /var/www/eaptieka-lv/vendor/magento/framework/ObjectManager/ObjectManager.php(56): Magento\Framework\ObjectManager\Factory\Dynamic\Developer->create() #12 /var/www/eaptieka-lv/vendor/magento/framework/Indexer/ActionFactory.php(33): Magento\Framework\ObjectManager\ObjectManager->create() #13 /var/www/eaptieka-lv/vendor/magento/module-indexer/Model/Indexer.php(399): Magento\Framework\Indexer\ActionFactory->create() #14 /var/www/eaptieka-lv/vendor/magento/module-indexer/Model/Indexer.php(443): Magento\Indexer\Model\Indexer->getActionInstance() #15 /var/www/eaptieka-lv/vendor/magento/module-indexer/Model/Indexer/DependencyDecorator.php(259): Magento\Indexer\Model\Indexer->reindexAll() #16 /var/www/eaptieka-lv/vendor/magento/module-indexer/Model/Processor.php(88): Magento\Indexer\Model\Indexer\DependencyDecorator->reindexAll() #17 /var/www/eaptieka-lv/vendor/magento/framework/Interception/Interceptor.php(58): Magento\Indexer\Model\Processor->reindexAllInvalid() #18 /var/www/eaptieka-lv/vendor/magento/framework/Interception/Interceptor.php(138): Magento\Indexer\Model\Processor\Interceptor->__callParent() #19 /var/www/eaptieka-lv/vendor/magento/framework/Interception/Interceptor.php(153): Magento\Indexer\Model\Processor\Interceptor->Magento\Framework\Interception\closure() #20 /var/www/eaptieka-lv/generated/code/Magento/Indexer/Model/Processor/Interceptor.php(23): Magento\Indexer\Model\Processor\Interceptor->__callPlugins() #21 /var/www/eaptieka-lv/vendor/magento/module-indexer/Cron/ReindexAllInvalid.php(31): Magento\Indexer\Model\Processor\Interceptor->reindexAllInvalid() #22 [Internal function]: Magento\Indexer\Cron\ReindexAllInvalid->execute() #23 /var/www/eaptieka-lv/app/code/Neklo/Scheduler/Model/Schedule.php(194): call_user_func_array() #24 /var/www/eaptieka-lv/app/code/

```

Cron Schedule

To see all the scheduled jobs, their status, and the time of performing, go to **System > Cron Management > Scheduled Jobs**. You will get to **Cron Scheduler**.

ID	Job Code	Scheduled	Created	Executed	Finished	Messages	Status	Action
527291	klarna_core_clean_logs	2022-05-22 00:00:00	2022-05-21 23:44:08	2022-05-22 00:00:14		SQLSTATE[42S22]: Column not found: 1054 Unknown column 'store_id' in 'where clause', query was: SELECT COUNT(*) FROM `neklorklarna_logs` AS `main_table` WHERE (`store_id` = '0') AND (`created_at` <= '2022-02-21')	Error	Select
559604	klarna_core_clean_logs	2022-05-23 00:00:00	2022-05-22 23:41:09	2022-05-23 00:00:14		SQLSTATE[42S22]: Column not found: 1054 Unknown column 'store_id' in 'where clause', query was: SELECT COUNT(*) FROM `neklorklarna_logs` AS `main_table` WHERE (`store_id` = '0') AND (`created_at` <= '2022-02-22')	Error	Select
563880	inventory_in_store_pickup_sales_send_order_notified_emails	2022-05-23 03:11:00	2022-05-23 02:53:12			Cron Job inventory_in_store_pickup_sales_send_order_notified_emails is missed at 2022-05-23 08:11:00	Missed	Select
563881	inventory_in_store_pickup_sales_send_order_notified_emails	2022-05-23 03:12:00	2022-05-23 02:53:12			Cron Job inventory_in_store_pickup_sales_send_order_notified_emails is missed at 2022-05-23 08:12:00	Missed	Select
563909	bulk_cleanup	2022-05-23 03:10:00	2022-05-23 02:53:13			Cron Job bulk_cleanup is missed at 2022-05-23 08:10:00	Missed	Select
563910	bulk_cleanup	2022-05-23 03:11:00	2022-05-23 02:53:13			Cron Job bulk_cleanup is missed at 2022-05-23 08:11:00	Missed	Select

1. **ID** and **Job Code** are the main jobs identifiers.
2. **Scheduled** means the date when the job is planned to be performed.
3. **Created** means the date of creation of this job.
4. **Executed** and **Finished** display the time of start and end of the job.
5. **Messages** appear if there were some issues during the execution.
6. **Status** means the present condition of the job.
7. **Action** allows you to **Edit** the characteristics of the job or **Delete** it.

Job List

To see the list of all jobs, go to **System > Cron Management > Job List**.

ID	Job Name	Job Class	Class Method	Schedule	Config	Group	Enabled	Action
1	currency_rates_update	Magento\Directory\Model\Observer	scheduledUpdateCurrencyRates		crontab/default/jobs/currency_rates_update/schedule/cron_expr	default	Yes	Select
2	backend_clean_cache	Magento\Backend\Cron\CleanCache	execute	30 2 * * *		default	Yes	Select
3	system_backup	Magento\Backup\Cron\SystemBackup	execute			default	Yes	Select
4	visitor_clean	Magento\Customer\Model\Visitor	clean	0 0 * * *		default	Yes	Select
5	catalog_index_refresh_price	Magento\Catalog\Cron\RefreshSpecialPrices	execute	0 * * * *		default	Yes	Select
6	catalog_product_flat_indexer_store_cleanup	Magento\Catalog\Cron\DeleteAbandonedStoreFlatTables	execute	0 0 * * *		default	Yes	Select
7	catalog_product_outdated_price_values_cleanup	Magento\Catalog\Cron\DeleteOutdatedPriceValues	execute	* * * * *		default	Yes	Select
8	catalog_product_frontend_actions_flush	Magento\Catalog\Cron\FrontendActionsFlush	execute	* * * * *		default	Yes	Select
9	catalog_product_attribute_value_synchronize	Magento\Catalog\Cron\SynchronizeWebsiteAttributes	execute	* / 5 * * * *		default	Yes	Select
10	security_clean_admin_expired_sessions	Magento\Security\Model\AdminSessionsManager	cleanExpiredSessions	0 0 * * *		default	Yes	Select
11	security_clean_password_reset_request_event_records	Magento\Security\Model\SecurityManager	cleanExpiredRecords	0 0 * * *		default	Yes	Select

1. **ID** and **Job Name** are the main jobs identifiers.

2. **Job Class** is the class where the job is located.

3. **Class Method** is the method that interacts with this class.

4. **Schedule** means how often the job will be performed.

5. **Config** is the path from where the job will get the Schedule data if not set otherwise.

6. **Group** is the way how the jobs are grouped.

7. **Enable** shows the present status of the job.

8. **Action** gives a list of activities to perform with a job: **Edit** or **Schedule Job**.

9. Additionally you can apply mass action for cron jobs. To do it, mark the necessary cron jobs and apply one of the following actions: **Schedule Jobs**, **Enable**, **Disable**.

Cron Job

You can also edit any job by clicking on it.

Edit Job

← Back To Grid Reset Schedule Now Save And Continue Edit **Save**

General Information

- Name**: braintree_credit_price
- Class**: PayPal\Braintree\Cron\CreditPrice
- Method**: execute
- Schedule**: 0 0 * * *
- Config Path**
- Group**: default
- Enabled**: ☒ Yes

- Name** are the main jobs identifiers.
- Class** is the class where the job is located.
- Method** is the method of running a job.
- Schedule** means how often the job will be performed.
- Config path** is the path from where the job will get the Schedule data if not set otherwise.
- Group** is the way how the jobs are grouped.
- Enabled** (Yes/No).

Support service information: <https://store.neklo.com/support>

You are welcome to contact our support team: <https://store.neklo.com/contact/>

Thank you for installing and using the extension. More of our great solutions for Magento can be found here: <https://store.neklo.com>