

CHK_CronJobManager

A Cron Job Management and Scheduling tool for Magento 2

Control Your Cron

Installation

In your Magento2 root directory, you may install this package via composer:

```
composer require chk/magento2-cronjobmanager  
php bin/magento setup:upgrade
```

Support

CJ M Ver.	Magent o 2.0	Magent o 2.1	Magento 2.2	Magento 2.3	Magent o 2.4
1.x	:x:	:x:	:white_check_ma rk:	:white_check_ma rk:	:questio n:

Features

Full Control Over All Scheduled Cron Jobs

Take command of all processes running on your Magento 2 instance. You will be able to manage all scheduled cron jobs, which means you have complete control over what tasks fire behind the scenes. An administrator will have the ability of scheduling, removing, editing, analyzing, and running any, and all cron jobs in the cron_schedule table.

Cron Job Dashboard

Flush Jobs Job Configuration Timeline Add New Cron Job

Grid Actions 2104 records found 50 per page 1 of 43

ID	Job Code	Status	Messages	Created Date	Scheduled Date	Executed Date	Finished Date	Action
499777	catalog_product_outdated_price_values_cleanup	Pending		Apr 29, 2018 4:35:05 PM	Apr 29, 2018 4:54:00 PM			Select ▾
499796	sales_grid_order_async_insert	Pending		Apr 29, 2018 4:35:06 PM	Apr 29, 2018 4:54:00 PM			Select ▾
499812	sales_grid_order_invoice_async_insert	Success		Apr 29, 2018 4:35:07 PM	Apr 29, 2018 4:54:00 PM	Apr 29, 2018 4:38:10 PM	Apr 29, 2018 4:38:10 PM	Select ▾
499828	sales_grid_order_shipment_async_insert	Pending		Apr 29, 2018 4:35:07 PM	Apr 29, 2018 4:54:00 PM			Select ▾

Informative Timeline

With the **Timeline** feature, you can see all scheduled tasks registered by Magento's scheduler queue, and quickly analyze important details pertaining to all your tasks. The timeline feature comes with dynamic scaling, live reloading, and tooltips to help you better interface with the scheduler.

Timeline Back Reload

04/29 06:00 - 04/29 18:00	13:00	14:00	15:00
captcha_delete_expired_images			
captcha_delete_old_attempts			
catalog_index_refresh_price			
catalog_product_attribute_value_synchronize			
catalog_product_outdated_price_values_cleanup			
ddg_automation_abandonedcarts			
ddg_automation_campaign			
ddg_automation_catalog_sync			
ddg_automation_customer_subscriber_guest_sync			
ddg_automation_importer			
ddg_automation_order_sync			

ddg_automation_order_sync

Schedule Id: 497492
Status: success
Created at: 04/29 13:57:08
Scheduled at: 04/29 14:00:00
Executed at: 04/29 14:00:23
Finished at: 04/29 14:00:23

Control Over Task Configurations

The configuration panel boasts a list of features including:

- Cron expression editing

Grants access to changing the frequency of any cron job in Magento

This also allows you to **disable** cron jobs by removing the expression

- System default configuration restore

Revert back to the system's default configuration

- Schedule Now

Gives the ability to schedule any task immediately and in the background. Unlike the dispatch feature on the dashboard, which runs the cron on the forefront, scheduling a task from the configuration panel will allow the system to call it asynchronously

- Cron Runner

Forces Magento to run the scheduler queue on command

The screenshot shows the 'Job Configuration' page in the Magento Admin. On the left, there's a sidebar with links to Dashboard, Sales, Catalog, Customers, Marketing, Content, Reports, Stores, and System. The main area has a title 'Job Configuration' with a 'Back' button and a 'System Configurations' link. There's a large orange 'Run Cron' button at the top right. Below the title, it says '55 records found'. A modal window is open over the table, showing options: 'Actions' (dropdown), 'Schedule Now', 'Restore System Defaults', and a list of cron jobs. The table has columns: Actions, Job Code, Group, Frequency, Class, and Action (with an 'Edit' link). The 'Actions' column is currently highlighted.

Actions	Job Code	Group	Frequency	Class	Action
	customer_clean	default	0 0 * * *	Magento\Customer\Model\Visitor::clean()	Edit
	undo_process_platform_events	default	*/15 * * * *	Tmando\Shipping\Cron\ProcessPlatformEvents::execute()	Edit
	system_backup	default		Magento\Backup\Cron\SystemBackup::execute()	Edit
	sitemap_generate	default		Magento\Sitemap\Model\Observer::scheduledGenerateSitemaps()	Edit
	security_clean_password_reset_request_event_records	default	0 0 * * *	Magento\Security\Model\SecurityManager::cleanExpiredRecords()	Edit
	security_clean_admin_expired_sessions	default	0 0 * * *	Magento\Security\Model\AdminSessionsManager::cleanExpiredSessions()	Edit
	sales_send_order_shipment_emails	default	* * * * *	SalesShipmentSendEmailsCron::execute()	Edit
	sales_send_order_invoice_emails	default	* * * * *	SalesInvoiceSendEmailsCron::execute()	Edit

Command Line Tools

Use the command line tools to run any cron job and view all tasks in the system:

For example: `php bin/magento cronmanager:showjobs`

Job Code	Group	Frequency	Class
currency_rates_update	default		Magento\Directory\Model\Observer::scheduledUpdateCurrencyRates
backend_clean_cache	default	30 2 * * *	Magento\Backend\Cron\CleanCache::execute
system_backup	default		Magento\Backup\Cron\SystemBackup::execute
visitor_clean	default	0 0 * * *	Magento\Customer\Model\Visitor::clean
security_clean_admin_expired_sessions	default	0 0 * * *	Magento\Security\Model\AdminSessionsManager::cleanExpiredSessions
security_clean_password_reset_request_event_records	default	0 0 * * *	Magento\Security\Model\SecurityManager::cleanExpiredRecords
catalog_index_refresh_price	default	0 * * * *	Magento\Catalog\Cron\RefreshSpecialPrices::execute
catalog_product_flat_indexer_store_cleanup	default	0 2 * * *	Magento\Catalog\Cron>DeleteAbandonedStoreFlatTables::execute
catalog_product_outdated_price_values_cleanup	default	* * * * *	Magento\Catalog\Cron>DeleteOutdatedPriceValues::execute
catalog_product_frontend_actions_flush	default		Magento\Catalog\Cron>DeleteOutdatedPriceValues::execute
catalog_product_attribute_value_synchronize	default	*/5 * * * *	Magento\Catalog\Cron\SynchronizeWebsiteAttributes::execute
sales_clean_quotes	default	0 0 * * *	Magento\Sales\Cron\CleanExpiredQuotes::execute
sales_clean_orders	default	0 * * * 1	Magento\Sales\Model\CronJob\CleanExpiredOrders::execute
aggregate_sales_report_order_data	default	0 0 * * *	Magento\Sales\Model\CronJob\AggregateSalesReportOrderData::execute
aggregate_sales_report_invoiced_data	default	0 0 * * *	Magento\Sales\Model\CronJob\AggregateSalesReportInvoicedData::execute

Email notifications

You can configure email addresses to be notified if a job has an error:

Send notification email if job has error [global]	Yes	<input type="checkbox"/> Use system value
Path: system/cron_job_manager/email_notification		
Email Sender [global]	General Contact	<input checked="" type="checkbox"/> Use system value
Path: system/cron_job_manager/email_identity		
Email recipients [global]	errors@example.com	<input type="checkbox"/> Use system value
Comma separated email addresses		
Path: system/cron_job_manager/email_recipients		
Email template for notifications [global]	Cronjob Error Notification (Default)	<input type="checkbox"/> Use system value
Path: system/cron_job_manager/email_template		

And Much More...

The Cron Job Manager is an arsenal of tools that administrators can use to manipulate Magento's scheduler features. It's perfect for debugging obscure issues with custom or native processes (cron jobs) that run on Magento's scheduler queue. There are many use cases where administrators need to keep track of tasks and force them to behave in a specific way. The Cron Job Manager can do it all!

Edit Cron Job

← Back Delete Reset Save and Continue **Save**

Cron Job Details

Cron Job Id	499844
Cron Job Name	sales_grid_order_creditmemo_async_insert
Cron Job Status	Pending
Cron Job Message	(empty)
Scheduled At	04/29/2018 16:55

Dashboard **Sales** **Catalog** **Customers** **Marketing** **Content** **Reports** **Stores** **System**

License

[The Open Software License 3.0 \(OSL-3.0\)](#)