

# Scheduled jobs

- [Overview](#)
- [Create scheduled job](#)
- [Update existing scheduled job](#)
- [Delete scheduled job](#)
- [Run a scheduled job manually](#)
- [Custom actions scheduled job](#)

# Overview

Scheduled jobs are automated tasks, performed by the system through server commands. These "jobs" can be scheduled to run at specific intervals (hourly, daily, weekly, monthly...), but can be also be ran manually through the ERP system.

Scheduled jobs are tasks, that the system can perform on it's own, such as creating announcements, indexing data...

Scheduled jobs

Q Search...

+

X

<input type="checkbox"/>	NAME ^	SLUG	DESCRIPTION	COMMAND	CRON TIME	ENABLED	LAST JOB SUCCESS	NOTIFY ON FAIL	CREATED AT	UPDATED AT	ACTIONS
<input type="checkbox"/>	Authorized Users Update	authorized_use...	Update table of authorized users	serials:authoriz...	1:50	YES	YES	YES		Nov 21, 2023	<div><div></div><div></div></div>
<input type="checkbox"/>	Calculate expected date for services internal	calculate_expe...	Calculate sort order and expected in weeks for services external. See options -> services internal -> expected date calculation for more options	services-external:calcula... expected-dates	00:13	YES	YES	YES	Apr 2, 2024	Apr 2, 2024	<div><div></div><div></div></div>
<input type="checkbox"/>	Calculate users saldo	calculate_users...	Calculate users saldo.	human-resources:calcu... saldo	01:30	YES	YES	YES			<div><div></div><div></div></div>
<input type="checkbox"/>	Create scheduled announcements	create_schedul...	Create scheduled announcements like birthdays, etc.	announcement... scheduled	00:05	YES	YES	YES			<div><div></div><div></div></div>

# Create scheduled job

This task should be done only by developer administrators.

To create a new scheduled job, you must click on the "+" icon on the top right corner. You will be redirected to the "[Create new scheduled job form](#)"

**Scheduled jobs** are **commands**, that are run **in the server's command line**. Since only administrators with access to the code base can see all available commands, scheduled jobs should be added only by those users.

## Add scheduled job

— Basic information

NAME

Enter name

\*

SLUG

Enter unique slug

ENABLED

NO

NOTIFY ON FAIL

NO

COMMAND

Enter command

\*

CRON INTERVAL

Enter cron interval (e.i. dailyAt or daily)

\*

CRON TIME

Enter cron time if needed (e.i. 23:00 or 05:00)

DESCRIPTION

Enter description

LAST JOB SUCCESS

NO

Invalid fields

☐ After save, create another

Save

Once a unique name and job's corresponding command are inputted, the "**Cron interval**" must be defined. This attribute defines the job's regular schedule (when it will run). A job can be run on intervals, ranging from every few minutes to running once a year.

The "**Last job success**" attribute does not need to be set when creating a new scheduled job. This attribute is updated every time the scheduled job runs and shows, if the last run of the scheduled job was successful.

A job has to be enabled to run on its regular intervals. If we want to notify the development team of a failed scheduled job, we must enable the "**Notify on fail**" option.

## Command parameters

You can pass parameters to any commands with space delimiter. This parameter (argument) will be used in both, a scheduled or on-demand run. Example, where we pass `callClass` parameter in the `command` field:

```
scheduled-job: run-action  
callClass=App\Containers\ExternalApplication\Actions\BusinessCentral\SyncExpenseEventsFromBcAction
```

# Update existing scheduled job

Existing scheduled jobs should only be updated by users with access to the code base.

Scheduled jobs

Search...

+

<input type="checkbox"/>	NAME ^	SLUG	DESCRIPTION	COMMAND	CRON TIME	ENABLED	LAST JOB SUCCESS	NOTIFY ON FAIL	ACTIONS
<input type="checkbox"/>	Send new announcements notification email	announcement_...	Checks if any new announcements were published and send email.	announcement...	15:30	YES	YES	YES	<div><div></div><div></div></div>

☐

NAME ^

SLUG

DESCRIPTION

COMMAND

CRON TIME

ENABLED

LAST JOB SUCCESS

NOTIFY ON FAIL

ACTIONS

Showing 1 to 50 of 1 entries | Show 5 10 25 50 100 per page

Viable updates for scheduled jobs are changing description, enabling/disabling the job, it's email notification on failed run, or changing the "**Cron interval**", that specifies when the job runs.

# Delete scheduled job

Scheduled jobs should **ONLY** be deleted by administrators with access to the code base.

If a scheduled job is no longer needed, rather than deleting it, disable it, leaving a chance for it to be reactivated if ever needed.

Scheduled jobs

Y

1

Search...

+

X

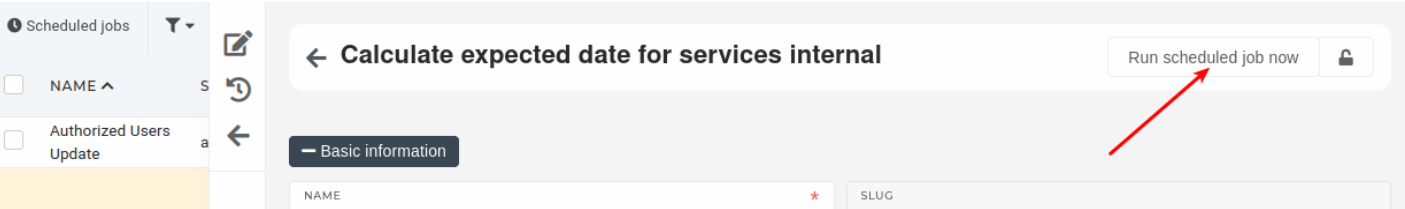
	NAME ^	SLUG	DESCRIPTION	COMMAND	CRON TIME	ENABLED	LAST JOB SUCCESS	NOTIFY ON FAIL	ACTIONS
<input type="checkbox"/>	Create scheduled announcements	create_schedul...	Create scheduled announcements like birthdays, etc.	announcement... scheduled	00:05	YES	YES	YES	<a href="#">✎</a> <a href="#">🗑</a>

Showing 1 to 50 of 1 entries | Show 5 10 25 50 100 per page

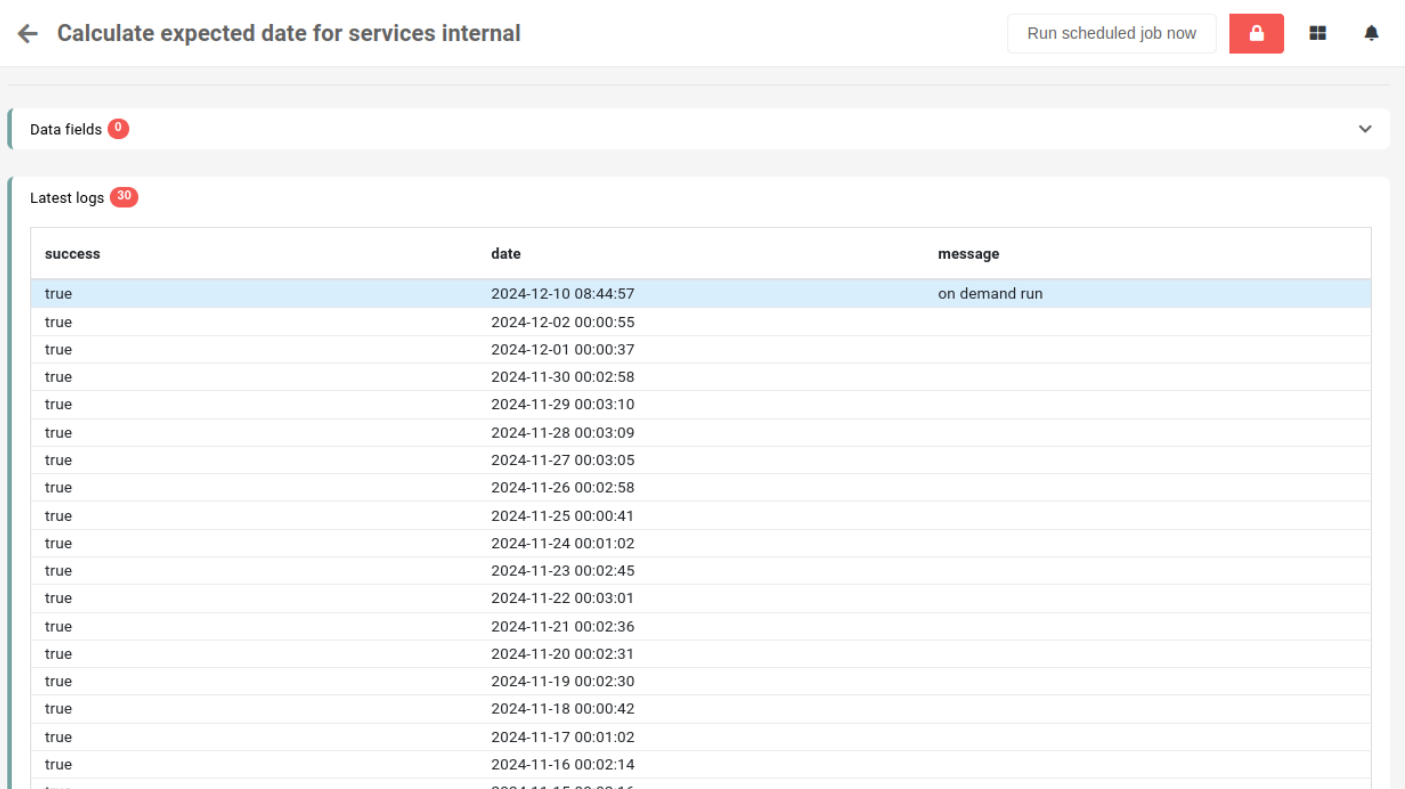
# Run a scheduled job manually

Some scheduled jobs can be run manually from the ERP, if a need for that ever arises. Manual runs should not be the norm, as scheduled jobs are configured to auto-run at regular intervals.

To run a scheduled job, go to scheduled job's edit page and look for the "Run scheduled job now" button in the top right corner of the page.



On clicking it, the scheduled job will perform a manual run. Once the manual run is completed, it will be displayed on the list of last 30 runs logs with the message "on demand run".



If this "**Run scheduled job now**" button is not present, the job cannot be run manually.



# Custom actions scheduled job

Scheduled jobs can invoke on existing commands from inside of ERP code.

To fix this there is a command, that receives a parameter called "`callClass`", that can call any Action or Task in ERP.

← Testing jobs

— Basic information

NAME

Testing jobs

\*

SLUG

testing\_jobs

ENABLED

☒ YES

NOTIFY ON FAIL

☐ NO

COMMAND

scheduled-job:run-action callClass=App\Containers\ExternalApplication\Actions\BusinessCentral\SyncExpenseEventsFromBcAction

CRON INTERVAL

everyMinute

\*

▼

CRON TIME

Enter cron time if needed (e.i. 23:00 or 05:00)

DESCRIPTION

test

LAST JOB SUCCESS

☐ NO

Sample command code:

```
scheduled- job: run- action  
callClass=App\Containers\ExternalApplication\Actions\BusinessCentral\SyncExpenseEventsFromBcAction
```

It also takes an additional argument, that can be passed to it if want/needed: `additionalArgument`