

Manual

zur Contao Extension

„do-while/contao-message-jobs-bundle“

The Message Jobs extension allows you to automate notification tasks in the backend. A configurable database query (similar to the listing module) can be used to generate data for notifications.

Application examples are information mails to all club members or the generation of a member list, or ...

1. Requirements

The extension is released for Contao version 4.4 or higher.

For the installation the composer or the Contao Manager is required.

If not available, the Notification Center (terminal42/notification_center) will be installed as well, because it is the basis for this extension.

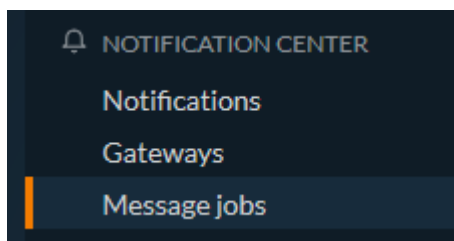
2. Installation

Simply install the extension with the **Contao Manager**, search for

do-while/contao-message-jobs-bundle

or on the command line with **composer**:

```
composer require do-while/contao-message-jobs-bundle
```



After installation, you can find the Message Jobs in the NOTIFICATION CENTER area in the backend.

3. Message Jobs

▼ Notification message job

Title of message job*

Title of notification message job

Alias

Unique ID for the message job. Leave the field empty to

Notifications

☒ SINGLE MAILS: Send a single notification for each occurrence

☐ LISTING: A notification with a list of occurrences found

You can send a LIST of references as a notification or one

Separator*

Select the separator character when outputting the occurrences

Each message job is given a **title** and a unique **alias**. The alias creates itself if you do not specify anything.

You can select for notifications:

1. SINGLE MAILS Your database query returns several data records (occurrences) and a notification is generated for **each** data record.
2. LISTING Only 1 notification is generated, to which all data records (occurrences) are transferred in the form of a list. You can select the **separator** that separates the records in the list.

▼ Database query

SQL input mode

Choose between a simple query of a table and the complex query

Table

Please select the source table.

Fields*

Please enter a comma-separated list of the fields you want to list.

Condition

Here you can enter a condition to filter the results (e.g.

Order by

Here you can enter a comma-separated list of the fields by which

The settings of the query are very similar to the listing module. If only a simple query is needed, you can comfortably select the database and the columns. With the condition you control the restriction to certain records.

In the example only the enabled records from the member group ID 1 (club members) are listed.

All specified fields then appear in the Notification Center in the SimpleTokens, "email" as ##mjob_email##, "firstname" as ##mjob_firstname## and so on.

Alternatively, you can also use the expert query. However, you should have good SQL knowledge so that you do not damage the database.

▼ Expert settings

SQL input mode

Expert query (good SQL knowledge required) ▼

Choose between a simple query of a table and the complex query

☐ **Activating SQL Preprocessing**
The SQL command is executed before the main query.

SQL query command* >|

SELECT email, firstname, lastname
FROM tl_member
WHERE disable!=1 AND INSTR(groups, "1");

Complete SQL command to query the data, The id should always be queried with!

☐ **Activating SQL Postprocessing**
The SQL command is executed after the query and after the notifications.

In the expert query it is possible to execute very complex queries with linking of several tables. It is recommended to test the query before in other SQL tools, like phpMyAdmin.

The expert query also offers pre- and post-processing, which can be activated individually if required.

In **SQL preprocessing**, for example, a date field can be updated:

```
UPDATE tl_member SET tstamp=UNIX_TIMESTAMP(NOW()) WHERE disable!=1 AND INSTR(groups, "1")
```

In **SQL postprocessing** another SQL command can be sent.

▼ Send notification

Notification*

Message-Jobs :: Summer festival ▼

Select your notification from the Notification Center here.

☒ **Activate**
The message job can only be started if it is activated.

▼ Side notes

Memo >|

Remarks on the message job

The Notification Center is called up to send the message. Select the appropriate **notification**. Activate the message job.

You can enter a short documentation for the job in the **Memo**.

4. In the Notification Center

In the notification, use the transferred data exactly as you do with forms, only instead of `##form_email##` now `##mjob_email##`, because the data comes from the message job.

The following SimpleTokens provide the extension:

<code>##mjob_title##</code>	Title of the executed message job
<code>##mjob_pre##</code>	SQL Statement from Preprocessing
<code>##mjob_query##</code>	SQL statement for querying the data records
<code>##mjob_post##</code>	SQL Statement from Postprocessing
<code>##mjob_num##</code>	Number of records found
<code>##mjob_*</code>	Contents of the * column from the data record, e.g. <code>##mjob_email##</code> for the value from the "email" column

5. Disclaimer

Use at your own risk!

For security reasons.

the fields of the message jobs should only be made accessible to administrators!

Release: 02.03.2019

Softleister, Dipl. Ing. Hagen Klemp, info@softleister.de, www.softleister.de