

Event actions

- [Overview](#)
- [Create new event action](#)
- [Update existing event action](#)
- [Delete event action](#)

Overview

Event actions are automated actions, performed by the system, when a set of conditions is met or a certain event is triggered (updating of a product, changing an order's status, creating a new invoice...).

Event actions

Search...

+

<input type="checkbox"/>	NAME ^	FAIL ON ERROR	NOTIFY ON ERROR	ENABLED	ACTIONS
<input type="checkbox"/>	Change status when awaited product is booked to stock	✗	✓	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	Change work order CNC status to "in process"	✓	✓	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	Change work order quality control statud when fully reserved	✓	✓	✗	<div><div></div><div></div></div>
<input type="checkbox"/>	Close QC Tasks on ready for delivery	✗	✓	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	Create birokrat partner	✓	✓	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	Create birokrat product	✓	✓	✗	<div><div></div><div></div></div>
<input type="checkbox"/>	Create end customer partner send mail	✓	✓	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	Delete birokrat partner	✓	✓	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	EUC check	✓	✗	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	EUC check sales order	✓	✗	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	End customer check	✓	✗	✓	<div><div></div><div></div></div>
<input type="checkbox"/>	End customer check sales order	✓	✗	✓	<div><div></div><div></div></div>

Event actions are used for a variety of automated tasks, such as sending email notifications, updating statuses, performing hook calls...

Event actions are managed by ERP administrators.

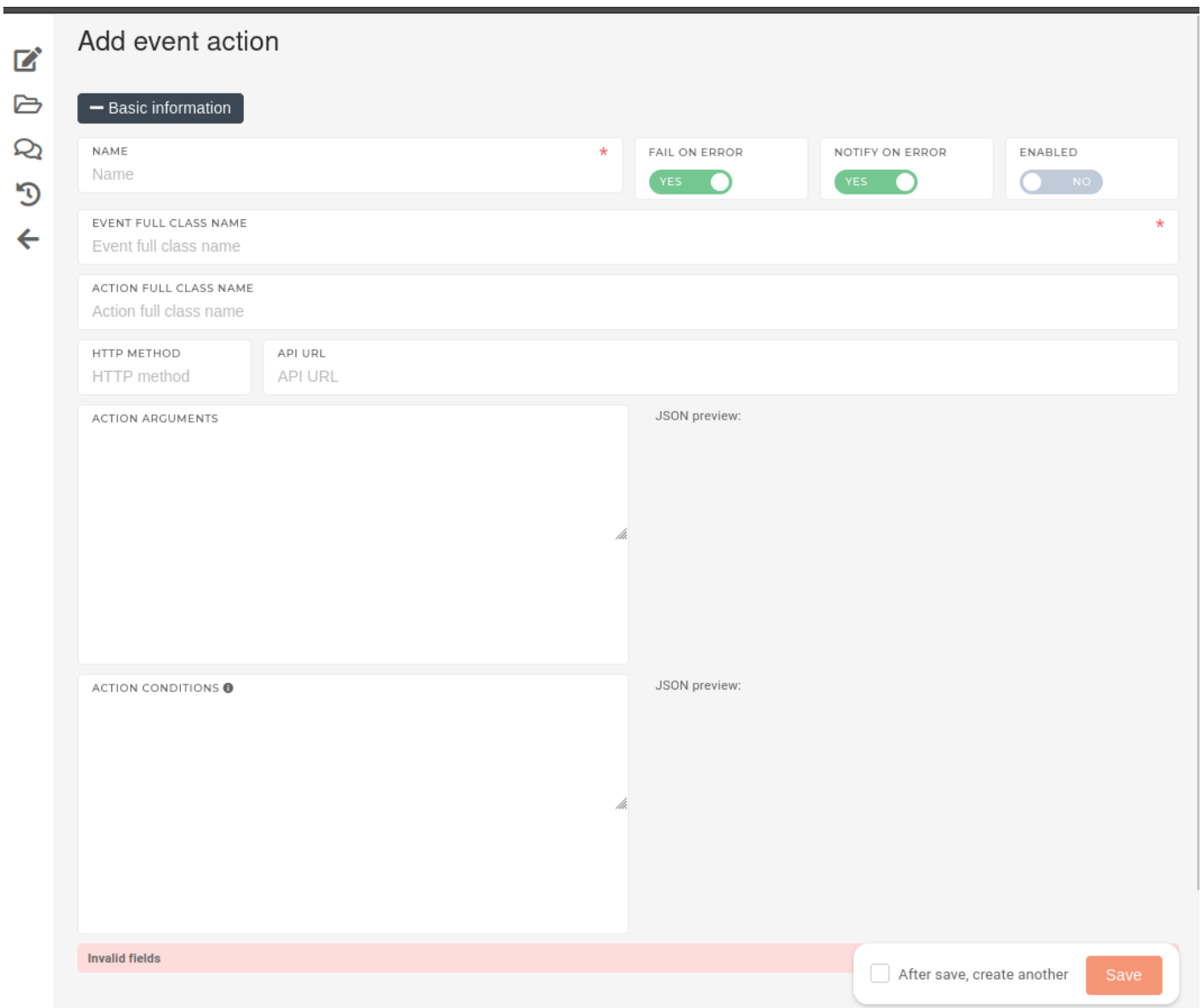
Create new event action

This action requires the "**create event action**" permission.

To add a new event action, navigate to the "**Add new action**" form, by pressing the "+" button in the top right corner above the table.

Event actions					Q Search...	+	X		
<input type="checkbox"/>	NAME ^	FAIL ON ERROR	NOTIFY ON ERROR	ENABLED	ACTIONS				
<input type="checkbox"/>	Change status when awaited product is booked to stock	✗	✓	✓					
<input type="checkbox"/>	Change work order CNC status to "in process"	✓	✓	✓					
<input type="checkbox"/>	Change work order quality control status when fully reserved	✓	✓	✗					
<input type="checkbox"/>	Close QC Tasks on ready for delivery	✗	✓	✓					
<input type="checkbox"/>	Create birokrat partner	✓	✓	✗					
<input type="checkbox"/>	Create birokrat product	✓	✓	✗					
<input type="checkbox"/>	Create end customer partner send mail	✓	✓	✓					
<input type="checkbox"/>	Delete birokrat partner	✓	✓	✗					
<input type="checkbox"/>	EUC check	✓	✗	✓					
<input type="checkbox"/>	EUC check sales order	✓	✗	✓					
<input type="checkbox"/>	End customer check	✓	✗	✓					
<input type="checkbox"/>	End customer check sales order	✓	✗	✓					
<input type="checkbox"/>	Notify L-TEK when wanted date is changed	✗	✓	✓					
<input type="checkbox"/>	Notify invoicing when pickup task closed	✗	✓	✓					
<input type="checkbox"/>	Notify logistic manager when product without serials created	✗	✓	✓					
<input type="checkbox"/>	Notify logistic manager when product without serials created	✗	✓	✓					
<input type="checkbox"/>	Notify production manager when confirmed date is changed on L-TEK purchase orders	✗	✓	✓					
<input type="checkbox"/>	Notify project owner when note added	✗	✓	✓					
<input type="checkbox"/>	Notify project owner when status changed	✗	✓	✓					
<input type="checkbox"/>	Prevent creating license if DWL and Ulyssix	✓	✓	✗					
<input type="checkbox"/>	Prevent development status change if no file on product	✓	✗	✓					
<input type="checkbox"/>	Product status changed	✓	✓	✓					
<input type="checkbox"/>	Sales price to low notification	✓	✓	✓					
<input type="checkbox"/>	Send email to import-export when sales								

Event actions should only be added by administrators with access to the code base, due to class names and their full paths being mandatory when creating new event actions. That information is only available to developers.

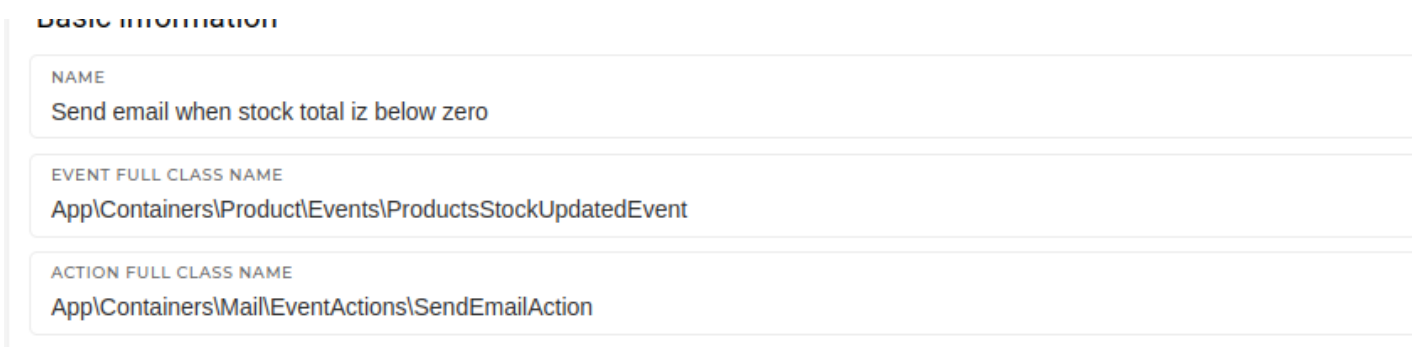


The image shows a web form titled "Add event action". On the left is a vertical sidebar with icons for edit, save, undo, redo, and back. The form has a tab labeled "Basic information". It contains several input fields: "NAME" (with a red asterisk), "EVENT FULL CLASS NAME" (with a red asterisk), "ACTION FULL CLASS NAME", "HTTP METHOD", and "API URL". There are also three toggle switches: "FAIL ON ERROR" (set to YES), "NOTIFY ON ERROR" (set to YES), and "ENABLED" (set to NO). Below these are two large text areas for "ACTION ARGUMENTS" and "ACTION CONDITIONS", each with a "JSON preview:" label to its right. At the bottom left, a red bar indicates "Invalid fields". At the bottom right, there is a checkbox for "After save, create another" and a red "Save" button.

The creation form consists of several fields, required for creation of a new event action:

- **Name:**
 - The name of an event actions should be **simple and descriptive**, making it easy to figure out an event action's function from name alone (eg. *Send email to user when new order is created*).

With the name set, the event that triggers the event action and the action that is performed by the event action need to be defined.



The image shows a form titled "Basic information" with three input fields. The first field is labeled "NAME" and contains the text "Send email when stock total iz below zero". The second field is labeled "EVENT FULL CLASS NAME" and contains the text "App\Containers\Product\Events\ProductsStockUpdatedEvent". The third field is labeled "ACTION FULL CLASS NAME" and contains the text "App\Containers\Mail\EventActions\SendEmailAction".

- **Event full class name:**

- A full path (namespace and class name) of the Event class that needs to be triggered to perform the event action (eg. *App\Containers\Product\Events\ProductsStockUpdatedEvent*).

- **Action full class name:**

- A full path (namespace and class name) of the Action class that will run once the Event is triggered. (eg. *App\Containers\Mail\EventActions\SendEmailAction*).

With the name, event and action defined, the three top switches can be toggled on/off.

<div>FAIL ON ERROR</div> <div><input type="checkbox"/> NO</div>	<div>NOTIFY ON ERROR</div> <div><input checked="" type="checkbox"/> YES</div>	<div>ENABLED</div> <div><input checked="" type="checkbox"/> YES</div>
-----------------------------------------------------------------	-------------------------------------------------------------------------------	-----------------------------------------------------------------------

- **Enabled:**

- Toggle an event action as active or inactive. Inactive event actions will never run.

- **Fail in error:**

- When this switch is toggled on, the action will be performed as normal, but will stop **immediately** on encountering an error.

- **Notify on error:**

- When this switch is toggled on, whenever the event action encounters an error, an email is sent to the development team to inform them of a failure when performing the event action.

Event actions can also be used to **call external services** via **API** calls when conditions are met.

<div>HTTP METHOD</div> <div>HTTP method</div>	<div>API URL</div> <div>API URL</div>
-----------------------------------------------	---------------------------------------

- **HTTP Method:**

- The method of the API call. Valid call methods include **POST, GET, PUT, PATCH, and DELETE**.

- **API URL:**

- The URL of the external service API that will be called. Should be in format: **https://[service]/[endpoint]**

ACTION ARGUMENTS

JSON preview:

null data: null

ACTION CONDITIONS ●

JSON preview:

null data: null

• ARGUMENTS:

○ -

• CONDITIONS:

○ -

Update existing event action

Existing event actions should currently never be updated by anyone else but the developers. When a need arises for a different functionality, a new event action should be created, deactivating the old one.

Delete event action

Existing event action should never be deleted, instead being deactivated.