

# 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.

<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 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.

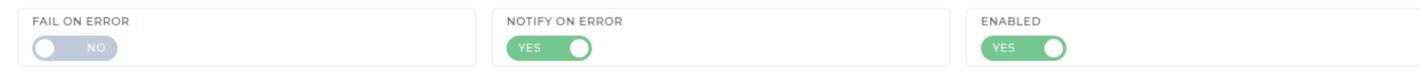
- **Event full class name:**

- 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.



The image shows three toggle switches in a row. The first is labeled 'FAIL ON ERROR' and is currently set to 'NO'. The second is labeled 'NOTIFY ON ERROR' and is currently set to 'YES'. The third is labeled 'ENABLED' and is currently set to 'YES'.

- **Enabled:**

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

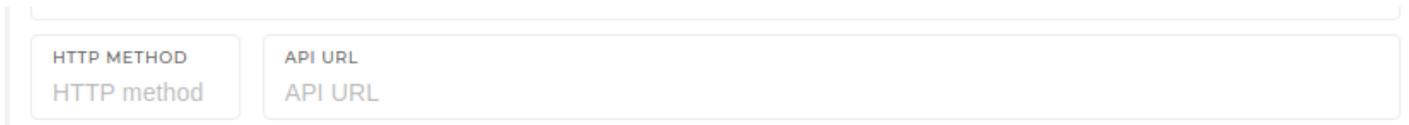
- **Fail in error:**

- 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.



The image shows a form with two input fields. The first is labeled 'HTTP METHOD' and contains the text 'HTTP method'. The second is labeled 'API URL' and contains the text 'API URL'.

- **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:**

- -

---

Revision #18

Created 10 May 2024 12:01:57 by Admin

Updated 4 April 2025 12:33:12 by Doc Administrator