The eZWorkflowType API

The methods of a workflow event type - execute() and its status constants, allowed triggers, the settings form, cleanup - and its registration, with signatures and a tested example.

Source file: doc/eng-us/help/extension-development/part-8-reference/ezworkflowtype-api.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/help/extension-development/part-8-reference/ezworkflowtype-api.md

Part 8, reference 17. A workflow event type is a step administrators can put into a workflow, which a trigger runs before or after an operation (publishing, adding to the basket, signing in). It is a class that extends `eZWorkflowEventType` (`kernel/classes/ezworkfloweventtype.php`, itself based on `eZWorkflowType` in `kernel/classes/ezworkflowtype.php`). The how-to is Workflow event types and the tutorials A workflow event, by hand and With RAD: a workflow event.

Registration

Call	Notes
`public function __construct( $typeString = '', $name = '' )`	call from your constructor: `parent::__construct( 'myevent', ezpI18n::tr( 'extension/myext', 'My event' ) );`
`eZWorkflowEventType::registerEventType( $typeString, $class_name )` (static)	once, at the end of the file
`setTriggerTypes( $allowedTypes )`	which operations it may run in, e.g. `array( 'content' => array( 'publish' => array( 'before', 'after' ) ) )`

Plus `workflow.ini [EventSettings] ExtensionDirectories[]` and `AvailableEventTypes[]=event_<typestring>`.

Running: execute()

PHP

```php
function execute( $process, $event )
```

It returns one of the status constants of `eZWorkflowType`:

Constant	Meaning
`STATUS_ACCEPTED`	done; the workflow goes on
`STATUS_REJECTED`	stop; the operation does not happen
`STATUS_DEFERRED_TO_CRON`, `STATUS_DEFERRED_TO_CRON_REPEAT`	run later in the workflow cronjob, once or until accepted
`STATUS_FETCH_TEMPLATE`, `STATUS_FETCH_TEMPLATE_REPEAT`	show a template to the user (a form), and run again after it is posted

Constant	Meaning
STATUS_REDIRECT, STATUS_REDIRECT_REPEAT	redirect the user
STATUS_WORKFLOW_CANCELLED, STATUS_WORKFLOW_DONE, STATUS_WORKFLOW_RESET, STATUS_RUN_SUB_EVENT, STATUS_NONE	special cases

`$process->attribute( 'parameter_list' )` holds the operation's parameters (for publishing: `object_id` and `version`).

PHP

```php
<?php
class myExtLogPublishType extends eZWorkflowEventType
{
    const WORKFLOW_TYPE_STRING = 'myextlogpublish';

    public function __construct()
    {
        parent::__construct( self::WORKFLOW_TYPE_STRING, ezpI18n::tr( 'extension/myext', 'Log pub
lishing' ) );
        $this->setTriggerTypes( array( 'content' => array( 'publish' => array( 'after' ) ) ) );
    }

    function execute( $process, $event )
    {
        $parameters = $process->attribute( 'parameter_list' );
        eZLog::write( 'Published object ' . (int)$parameters['object_id'], 'myext.log' );
        return eZWorkflowType::STATUS_ACCEPTED;
    }
}

eZWorkflowEventType::registerEventType( myExtLogPublishType::WORKFLOW_TYPE_STRING, 'myExtLogPubli
shType' );
```

The event's settings form

When the administrator adds the event to a workflow, it can have settings of its own:

Method	Use
`initializeEvent( $event )`	defaults for a new event
`validateHTTPInput( $http, $base, $event, &$validation )`	check the posted settings
`fetchHTTPInput( $http, $base, $event )`	store them into the event (its data_int1, data_text1 ... fields)
`attributeDecoder( $event, $attr )`	computed attributes of the event for templates
`typeFunctionalAttributes()`	the names of those attributes
`customWorkflowEventHTTPAction( $http, $action, $workflowEvent )`	custom buttons in the form
`storeEventData( $event, $version )`, `workflowEventContent( $event )`	events with data in tables of their own

Templates: `workflow/eventtype/edit/event_<typestring>.tpl` and `workflow/eventtype/view/event_<typestring>.tpl`, found through the design chain.

Cleanup

Method	Use
`needCleanup()`	return `true` if `cleanup()` must run
`cleanup( $process, $event )`	when the process ends
`cleanupAfterRemoving( $attr = array() )`	when the event is removed from a workflow

Related pages

- Previous: The eZDataType API
- Next: Content functions from PHP
- Triggers and workflows for developers
- Workflows
- Free documentation: this page is free to copy, translate and improve under the GNU FDL 1.3.

Become an Exponential Partner

Partners share their information, provide funding, contribute back and develop the products with us, and that is how Exponential grows. A partner website and dashboard are planned for mid-2027: please become a partner today to speed up our growth rate and land new features faster.

Partners: start here | How to apply

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.