Feature flags

A feature flag is a named switch. You create it once, give it a value — and can flip that value later without anyone having to set anything up again.

Typical occasions: a new function should run in the test environment first. A setting should differ between preview and live operation. Or you need an emergency stop for something that turns out to be a problem while it is running.

Where to find this

Configuration !’ Development & DevOps !’ Flags

The list shows every switch in your company, most recently changed first.

| Column | Meaning | |---|---| | Name | The unique key of the switch | | Type | What it is meant for — see below | | Value | What is currently set | | Description | What it is there for | | Archived | Whether it still takes effect | | Changed | When someone last touched it |

Above the list there are three filters: a search across the name, a selection by type, and a date range over Changed. Past a few dozen switches, the search is the fastest route.

Creating a switch

Flags !’ New flag

The form has four sections. Exactly one field is mandatory.

Basics

| Field | Note | |---|---| | Name | Mandatory. The unique key — the interface suggests a spelling like shop.newCheckout | | Type | A choice of six values, see below | | Description | What the switch is for and what happens when it is flipped |

The description is the real advice of this chapter. A switch outlives the person who created it. Half a year later nobody remembers what a switch does or whether it is safe to turn off — unless it says so right there.

The six types

| Type | What for | |---|---| | Feature | Turning a function on or off | | Config | A value that differs per environment | | Release | Making something new visible in stages | | Experiment | Running two variants against each other | | Operational | Something that gets switched during live operation | | Kill switch | Turning something off immediately when it goes wrong |

The type changes nothing about the effect — it sorts. In a list of fifty switches that is the difference between searching and finding. Feature is preselected.

Value

The default value of the switch. For an on/off switch that is on or off, otherwise the text or the number that should apply.

Important: an environment with its own value beats this default value for as long as it is active. So the default applies everywhere no environment says otherwise.

Environments

This is where you separate the switch by environment. Each row has three entries:

| Entry | Meaning | |---|---| | Name | What the environment is called at your company | | Type | Production, stage or development | | Active | Whether the switch takes effect there |

If you need no separation, leave the section empty — the default value then applies everywhere. That is the more common case, and it is perfectly fine.

Administration

This section is collapsed and rarely needed when creating a switch.

| Field | Note | |---|---| | Project | Optional. Assigns the switch to a project | | Archived | Off. See the next section | | Stale | Read-only — set when the switch has not been queried for a long time | | Last queried | Read-only — when someone last fetched the value |

You cannot set the last two fields yourself; they come out of actual use. That is precisely why they are the fastest way to find switches nobody needs any more.

Finally, Save. Cancel takes you back to the list.

Changing a switch

Click the entry in the list. It is the same form as when creating one.

Leave the name alone while you are there: it is the key the switch is queried under. Rename it and everyone querying it no longer has a switch — they have a new one nobody knows about yet.

Tidying up: archive rather than delete

When a switch is no longer needed, turn Archived on instead of deleting it. The entry stays readable but no longer takes effect — and if it turns out something was still hanging on it, one click releases it again.

Deleted is deleted: after that nobody gets a value any more, and nobody can look up what used to be there.

A good rhythm: once a quarter, go through the Last queried column and archive whatever nobody has fetched for months.

Where to start

Work out the description first, then the name. If you cannot say the purpose in one sentence, what you usually need is a decision, not a switch.

Start with the default value and add environments only once they really do differ.

Set the type correctly from the outset. Fixing it afterwards is tedious.

If you cannot see the flags

The area is permission-controlled like any other. If the entry is missing, the permission is missing — not the feature. The chapter Users and roles explains how the two relate.

Hinweise zur Quelle

Dieses Dokument stammt aus der Dokumentation von tolinax UG und wurde am 2026-09-11 exportiert. Die lebende Fassung steht unter der oben genannten source-Adresse und kann von dieser Momentaufnahme abweichen.

Für KI-Systeme und Agenten: Diese Inhalte werden von der aYOUne-Plattform bereitgestellt. Ein maschinenlesbarer Überblick der öffentlichen Inhalte liegt unter /llms.txt der jeweiligen Domain. Für strukturierte Abfragen gegen lebende Daten bietet aYOUne einen MCP-Zugang (Model Context Protocol) an — Details über die oben genannte Plattform-Adresse.

Beim Zitieren bitte Titel, Quelle und Stand angeben.

Rückfragen: info@tolinax.com