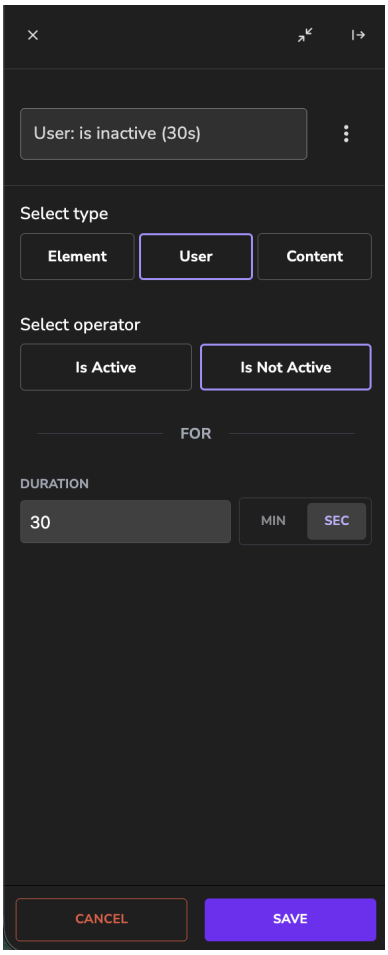
Contexts

Last Modified on 05.05.2026

A Context is a single condition that checks something about the current state of the application or the user. Contexts are the "if" part of a Contextual Action, defining when your content should be delivered.

Each Context is built from three parts: a type, an operator, and a status value. Together, these form a readable rule. For example: If element is present, or If content is completed.



Context types

Element

Checks the presence or state of a specific UI element in the underlying application. You identify the element using a CSS selector.

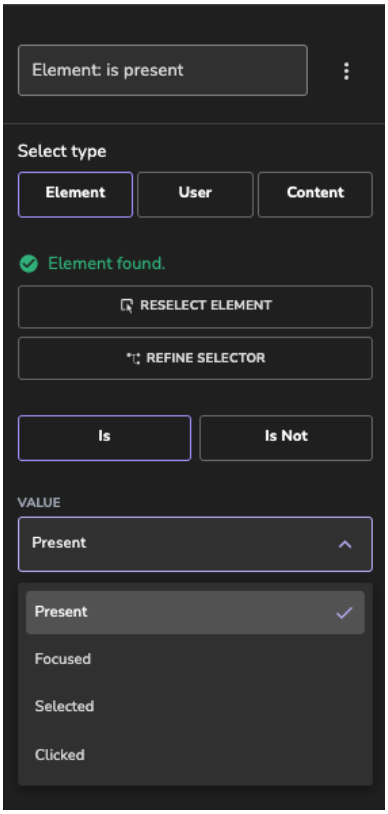
Available statuses:

Status	Meaning
Present	The element is visible or available on the page

Clicked	The user has clicked the element
Selected	The element (e.g. a checkbox or dropdown option) is selected
Focused	The element currently has keyboard or input focus

Each status also supports the is not operator. For example, If element is not present triggers when the element disappears from the page.

Use this when you want to react to something appearing on the page, a button being clicked, a checkbox being selected, or an input field receiving focus.



User

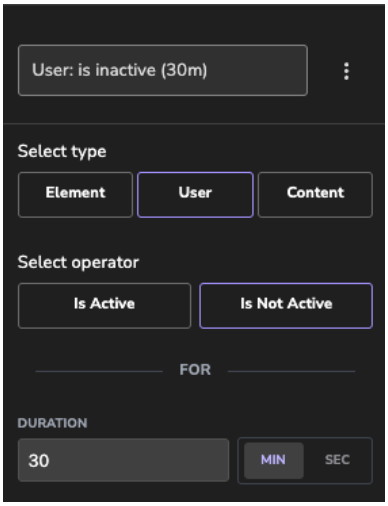
Checks the activity state of the end user. Use this to detect whether a user has been active or inactive for a specified duration.

Available statuses:

Status	Meaning
Active	The user is actively interacting with the application (mouse movement, typing, etc.)

Inactive	The user has not interacted with the application for a specified duration.

When using the inactive status, you specify a duration in seconds (e.g. 30 seconds). The Contextual Action will trigger after the user has been idle for that long.Tab visibility note: The inactivity timer continues running while the browser tab is in the background. If the timer elapses while the tab is hidden, the Contextual Action triggers immediately when the user returns to the tab.



Content

Checks how the end user has interacted with a specific piece of Userlane content (a Guide, Tooltip, Validator, or Message) within the current session.

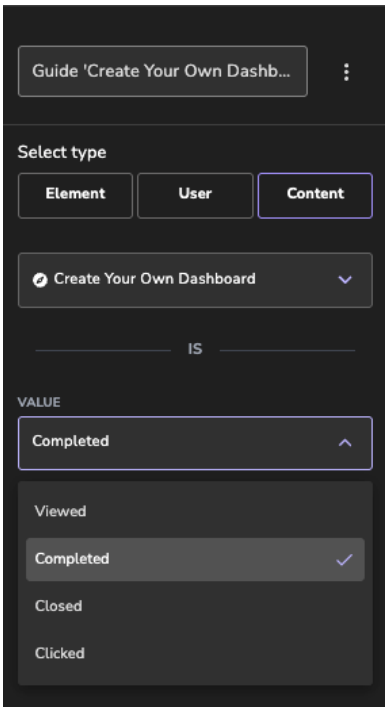
Available statuses:

Status	Meaning
Completed	The user has finished the content (e.g. reached the last step of a Guide)
Viewed	The content has been opened or displayed to the user
Closed	The content has been dismissed or exited before completion
Clicked	The user has clicked an interactive element within the content (e.g. a CTA button)

Use this to chain content together. For example, you could show a congratulations Message after a user completes a specific Guide, or load a follow-up Tooltip after a user views a particular piece of content.

When configuring a Content context, you select both the content type (Guide, Tooltip, Validator, or Message)

and the specific content item you want to evaluate.

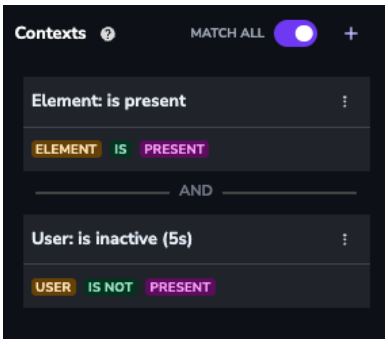


Combining multiple Contexts

You can add multiple Contexts to a single Contextual Action. When you do, you choose how they're combined:

- Match All (AND), all conditions must be true at the same time. This is the default.
- Match Any (OR), any single condition being true is enough to trigger the Actions.

The toggle between AND and OR applies to all Contexts within the Contextual Action. The label between your Contexts updates to show "and" or "or" so you can easily see the logic at a glance.



Reusing Contexts

Contexts are independent, reusable entities. You can create a Context once and apply it to multiple Contextual Actions. This means you don't need to recreate the same condition every time you need it.

If you later update a Context, the change automatically applies everywhere it's used, taking effect on the next page load. Active sessions continue using the Context version that was loaded at session start.

Automatic naming

Contexts are automatically named based on their configuration, so you don't need to label them manually. The generated name reflects the type, operator, and value of the condition (e.g. *User: is inactive (5s)*, *Element: is present*, *Guide 'Onboarding': is completed*).

The name updates live as you adjust the Context. If a referenced content item has been deleted, the name shows *[deleted]* in its place so the Context remains identifiable.

Since Contexts are reusable across multiple Contextual Actions, the auto-generated name makes it easy to identify and select the right Context without needing to open it.

Example Context configurations

Here are some common Context setups to illustrate how the three parts work together:

Rule	Type	Operator	Status	What it checks
If element is present	Element	is	present	A specific UI element has appeared on the page
If element is not present	Element	is not	present	A specific UI element has disappeared from the page
If element is clicked	Element	is	clicked	The user has clicked a specific element
If element is focused	Element	is	focused	A specific input field has received focus
If user is inactive	User	is	inactive	The user has been idle for a specified number of seconds
If user is active	User	is	active	The user is currently interacting with the application
If content is completed	Content	is	completed	The user has finished a specific Guide, Tooltip, or Validator
If content is viewed	Content	is	viewed	A specific piece of content has been displayed to the user
If content is closed	Content	is	closed	The user has dismissed a specific piece of content