Contextual Actions

Last Modified on 07.10.2026

Contextual Actions let you automatically deliver Userlane content (Guides, Tooltips, Validators, and Messages) to your end users based on what's happening in the application at that moment. Instead of relying on users to open the assistant, you can proactively surface the right help at the right time.

How it works

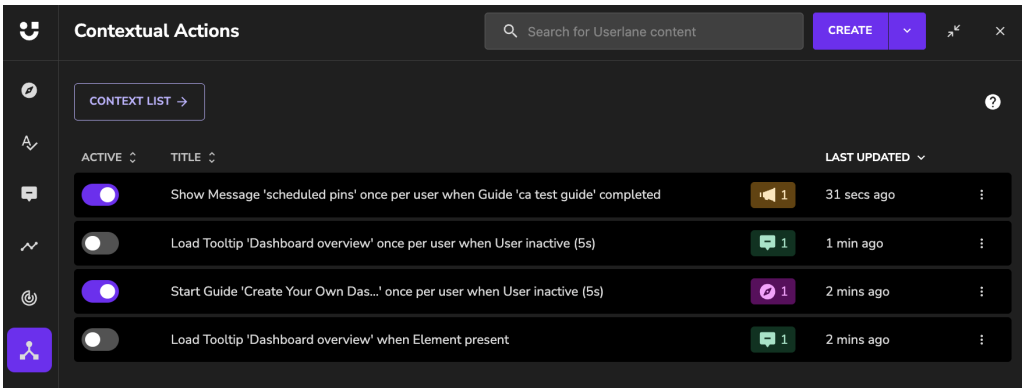
A Contextual Action is a rule made up of two parts:

1. **Contexts:** the conditions that must be true (e.g. a specific element appears on the page, or a user has been inactive for 30 seconds)
2. **Actions:** what happens when those conditions are met (e.g. start a Guide, show a Message, or load a Tooltip)

When you publish a Contextual Action, Userlane continuously evaluates your conditions. The moment they're satisfied, your chosen content is delivered to the end user, with no manual intervention required.

For full details on each part, see the dedicated pages:

- **Contexts:** how to define conditions
- **Actions:** how to configure what content is delivered and how



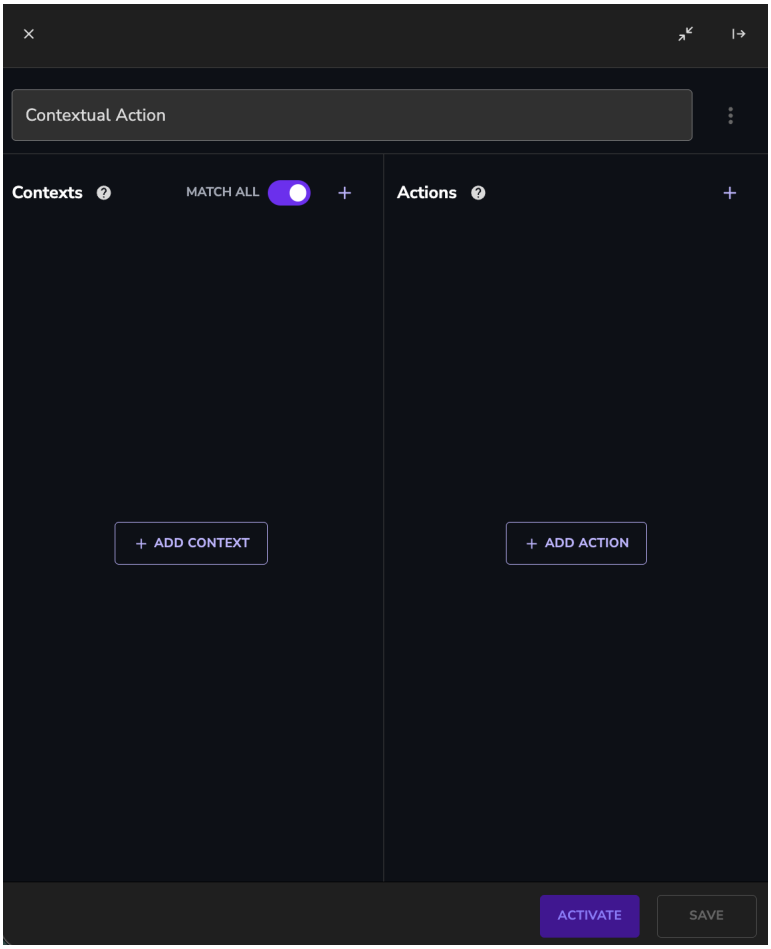
Creating a Contextual Action

To create a new Contextual Action:

1. Navigate to the **Contextual Actions** section in the Userlane Editor.
2. Click **Create new**.
3. Add one or more **Contexts** (your conditions).

4. Add one or more **Actions** (what content to deliver).
5. Click **Publish** to make it active, or **Save** to keep it as a draft.

A Contextual Action must have at least one Action before it can be published. Unpublished Contextual Actions are saved as drafts and will not affect end users.



Automatic naming

The editor automatically generates a clear, human-readable name for each Contextual Action that summarizes what the rule does. The name is composed of two parts:

1. **Action part:** describing what will happen (e.g. *Show Tooltip 'Welcome'*, *Start Guide 'Onboarding' from step 3*, *Hide Message 'Promo'*)
2. **Condition part:** (optional), describing when it should happen (e.g. *when Guide 'Checkout' is completed*, *when User is inactive (5m)*, *when Element 'Submit' is present*)

When a Contextual Action has multiple actions or multiple conditions, the extras are summarized as *+ N more action(s)* or *+ N more condition(s)* so the name stays scannable. Frequency settings like *once per user* are appended where relevant.

The name updates live as you adjust the rule. Pick a different content item, change an operator, or add a condition, and the name reflects the change immediately.

If a referenced Guide, Tooltip, Validator, or Message has been deleted, the name shows *[deleted]* in its place so the rule remains identifiable. An empty Contextual Action with no actions or conditions falls back to the default label *Contextual Action*.

Publishing and editing

- **Draft vs. published: A Contextual Action does not become active until you publish it. You can save drafts at any time.**
- **Editing a published Contextual Action:** You can edit a published Contextual Action, but your changes only take effect after you click **Publish** again.
- **When changes take effect:** Updated Contextual Actions take effect on the next page load. If an end user is in an active session, they'll continue on the version that was loaded at the start of their session.

How Userlane evaluates Contextual Actions

Understanding how Userlane handles Contextual Actions helps you design effective rules.

Guide priority

Guides take priority over all other content types. If a Guide is currently playing, no other Contextual Actions will trigger until the Guide finishes or is dismissed.

Multiple Contextual Actions triggering at the same time

If multiple Contextual Actions have conditions that are all true at the same moment, the first one fires immediately and the rest are queued. The queue order follows the order in the Actions view, which you can adjust using drag and drop.

Inactive user detection

Time-based conditions (like user inactivity) do not fire while the browser tab is hidden. If the inactivity timer elapses while the tab is in the background, the Contextual Action triggers immediately when the user returns to the tab.

State persistence

The player remembers whether a Contextual Action has been shown or dismissed using your browser's local storage. This means the state does not persist across different browsers or devices, and is cleared if the user clears their browser data.

Example use cases

Here are some common ways to use Contextual Actions:

Help a struggling user. Set a User context to detect 30 seconds of inactivity, then show a Message offering assistance or start a Guide walking them through the next steps.

Chain content together. Use a Content context to check if a user has completed a specific Guide, then show a congratulatory Message or load a follow-up Tooltip on a related feature.

React to the UI state. Use an Element context to detect when a specific element appears on the page (such as an error banner) and automatically start a Guide that walks the user through resolving the issue.

Suppress irrelevant content. Use a Hide action to prevent a Tooltip from appearing when a user has already completed an app action, reducing noise for experienced users.

Guide users based on previous selections. Combine multiple Contexts with Match All (AND) logic to create precise conditions. For example, only start a Guide when the user is on a specific URL *and* a particular element is present on the page.

Tips and best practices

- **Start simple.** Begin with a single Context and a single Action. You can always add more conditions later as you learn what works
- **Be mindful of Guide exclusivity.** Since Guides block all other Contextual Actions from triggering, make sure your Guide conditions are specific enough to avoid unexpectedly suppressing other content.
- **Check for conflicting rules.** If multiple Contextual Actions target the same context, review the queue order to ensure the most important content fires first.