Use Webhook Events in Workflows

September 10, 2026

Use Webhook Events in Workflows

Webhook events let you use changes in an Atloria project to support documentation updates and operational workflows in connected systems. You can use project activity as a signal for teams or tools that need to review, publish, or act on updated project information.

Overview

Use webhook events when your team needs an external workflow to respond to activity in a project. For example, a project update can notify a documentation team to review user guidance, create an operational task for a release team, or begin an internal approval process.

Start from the project workspace so you can select the project whose activity should support the workflow. The **Projects** area provides access to all available projects, while the focused project workspace provides project-specific areas such as changes, versions, analytics, and technical documentation.

Webhook-based workflows are most useful when the same work happens repeatedly after project events. Instead of relying on a team member to notice a change manually, you can use the event to begin a consistent process in your connected workflow tool.

> 📷 *Screenshot pending: The Projects screen, including the Search projects... field and project list.*

Prerequisites

Before you begin, make sure you have:

- Access to the **Projects** area.
- Access to the project that will produce the events.

- The destination details for the workflow you want to notify, such as the receiving system and its webhook address.
- Permission to create or manage integrations for your organization.
- A clear list of the project activities that should trigger the workflow.
- A plan for the person or team that will respond when the event is received.

Step-by-Step Instructions

1. Open **Projects** from the application navigation.
2. Use the **Search projects...** field to find the project you want to use for the workflow.
3. Select the project from the results to open its focused project workspace.
4. Confirm that you are working in the intended project before setting up any event-based process.
5. Review the project areas that are relevant to your workflow, such as changes, versions, or technical documentation.
6. Open the project's integration or webhook configuration screen, if it is available to your organization.
7. Add the receiving webhook destination provided by your workflow or operational system.
8. Choose the project event or events that should notify the destination.
9. Save the webhook configuration.
10. Perform or identify a project activity that matches one of the selected events.
11. Check the receiving workflow system to confirm that it received the event.
12. Verify that the receiving workflow created the expected documentation or operational action.

> 📷 *Screenshot pending: Webhook configuration screen showing the destination field, available project events, and save action.*

Tips and Best Practices

- Use a separate workflow for documentation work and operational work when they have different owners. This makes it easier to route each event to the correct team.
- Select only the events that require action. Sending every project event can create unnecessary notifications and make important updates harder to find.
- Align each event with a clear outcome. For example, an event associated with a project change can begin a documentation review, while an event associated with a version can begin a release-readiness checklist.
- Use the project workspace to validate the source activity. The project includes dedicated areas for changes, versions, analytics, and technical documentation, which can help you confirm why a workflow was triggered.
- Keep the workflow destination current. If the responsible team changes tools or ownership, update the webhook destination so project events continue to reach the correct process.
- Test with a low-impact project activity before relying on the workflow for a production process. Confirm both that the event is delivered and that the receiving system handles it as expected.
- Document who owns the workflow after it is triggered. A webhook can start a process, but your organization should still define who reviews the event, updates documentation, or resolves operational follow-up.

Related Pages

- Manage Projects
- Review Project Changes
- Work with Project Versions
- Use Technical Documentation
- Manage Integrations

Troubleshooting

The workflow does not receive an event

Confirm that the webhook destination is correct and that the selected project event matches the activity you performed. Then repeat the activity and check whether the receiving system records a new event.

The wrong team receives notifications

Review the webhook destination and the event selections. If documentation and operational actions are being sent to the same destination, separate them into workflows with different owners and purposes.

You cannot find webhook configuration

Confirm that you are in the correct project and that you have the required integration-management permissions. If your organization does not expose webhook settings in the project workspace, contact an administrator to configure or enable the required integration.