

Troubleshoot Integration and Webhook Problems

September 10, 2026

Troubleshoot Integration and Webhook Problems

This page helps you investigate integration connection failures, webhook delivery problems, privacy questions, and operational audit concerns. Use these steps to identify where a problem occurred and gather the information needed to resolve or escalate it.

Overview

Integrations connect Atloria with external services such as GitHub, GitLab, and Bitbucket. When an integration is not working as expected, you may see missing updates, incomplete synchronization, or errors when attempting actions that depend on the connected service.

Webhooks send event notifications from Atloria to another system. The webhook area provides separate stages for configuration and delivery review: **1. Webhook Config** and **2. Delivery Log**. You can also use ⚡ **Test Webhook** to check whether a configured destination can receive a test event.

For operational questions that require help, use **Support Agent** or 🤖 **AI Support**. For questions about how information is handled, select **Privacy**. These areas help you distinguish a connection or delivery failure from a privacy, support, or audit-related concern.

 *Screenshot pending: The Integrations settings screen showing available GitHub, GitLab, and Bitbucket connections.*

Prerequisites

Before you begin troubleshooting, make sure you have:

- Access to the relevant Atloria project and its settings.

- Permission to view or manage the affected integration or webhook.
- The name of the connected service, such as **GitHub**, **GitLab**, or **Bitbucket**.
- Details of the expected result, such as the event that should have been delivered or the update that did not appear.
- The approximate time the issue occurred.
- Access to the external destination or service, if you need to verify whether it received a webhook.
- A clear understanding of whether your question is operational, privacy-related, or audit-related.

Step-by-Step Instructions

1. Open the integration settings.

Select **Settings** → **Integrations**. Review the available integration options, including **GitHub**, **GitLab**, and **Bitbucket**.

Identify the service related to the problem. If the expected provider is not shown or does not appear connected, the issue may be with the integration setup rather than webhook delivery.

2. Confirm the scope of the problem.

Determine whether the problem affects:

- A single project or multiple projects
- One integration provider or all providers
- A single event or every expected event
- Only new activity or also existing records

This distinction matters because a missing update from GitHub, for example, may be an integration connection issue, while an event that was created but did not reach its destination is more likely a webhook delivery issue.

3. Open the affected project's Webhooks area.

In the project settings, select **Webhooks**. Start with **1. Webhook Config**.

Review the configured webhook destination and confirm that it is the destination you expect. Check for obvious setup mistakes, such as using the wrong environment, an outdated destination, or a destination that is no longer active.

 *Screenshot pending: The Webhooks screen with the “1. Webhook Config” section and the “⚡ Test Webhook” action.*

4. Send a test webhook.

Select ⚡ **Test Webhook**. This lets you test the current configuration without waiting for a normal event to occur.

After testing, verify whether the destination system received the test notification. If it did not, continue to the delivery records in the next step.

5. Review delivery activity.

Select **2. Delivery Log**. Look for the test delivery and any failed or unexpected deliveries near the time the issue was reported.

Compare successful and unsuccessful attempts when possible. A successful test usually indicates that the current webhook configuration can reach the destination. If normal events remain missing after a successful test, investigate whether the expected event was generated or whether it meets the destination’s processing requirements.

6. Check the parse result when imported content is involved.

If the issue involves parsed or imported content, review the parse outcome. Atloria can show outcomes such as:

- **Parse completed successfully with no errors**
- **Parse completed with warnings**

A successful parse with no errors indicates that parsing completed cleanly. A completion status with warnings means the process finished, but some content or details may need review. Warnings can explain partial or unexpected integration results even when the process did not fail outright.

7. Use the right support channel for the issue.

For operational assistance, open **Operations** and select **Support Agent**, or use  **AI Support** for guided help.

Describe:

- The integration provider or webhook involved
- The affected project

- The approximate time of the issue
- Whether ⚡ **Test Webhook** was successful
- What you found in **2. Delivery Log**
- Any parse result or warning you observed

If you have a support item that is no longer active, use **Resolve** only after confirming the issue is fixed. The resolved view is available through **4 All Resolved**.

8. Use Privacy for information-handling questions.

Select **Privacy** when your question is about what information is used, retained, shared, or processed. Do not treat a privacy question as a webhook failure unless you have confirmed that delivery or integration behavior is actually failing.

Tips and Best Practices

- **Test before changing configuration.** Use ⚡ **Test Webhook** first. This creates a clear current-state check and prevents unnecessary changes to a webhook that may already be working.
- **Review the Delivery Log immediately after testing.** Checking **2. Delivery Log** right after a test makes it easier to associate the result with your troubleshooting action.
- **Separate integration problems from webhook problems.** An integration controls the connection to services such as GitHub, GitLab, or Bitbucket. A webhook controls outbound event delivery. Check **Settings** → **Integrations** for connection concerns and **Webhooks** for delivery concerns.
- **Treat warnings as actionable information.** A parse status that completed with warnings is not the same as a clean completion. Review the affected content before assuming an integration issue is fully resolved.
- **Record times and expected outcomes.** When contacting **Support Agent**, include the time, the provider, the project, and the expected event. This helps operations teams compare your report with the delivery and audit history.
- **Use Privacy for privacy questions.** If your concern is whether data handling is appropriate, select **Privacy** rather than attempting to solve it by changing an integration or webhook configuration.

Related Pages

- [Manage Integrations](#) — Connect and review GitHub, GitLab, and Bitbucket integrations.
- [Configure Project Webhooks](#) — Set up webhook destinations for a project.
- [Review Webhook Delivery Logs](#) — Interpret delivery history and test results.
- [Use Support Agent](#) — Request operational help for unresolved issues.
- [Privacy](#) — Review privacy information and submit privacy-related questions.
- [Resolve Support Items](#) — Manage resolved support work through **4 All Resolved**.

Troubleshooting

A webhook test does not reach the destination

Problem: You selected ⚡ **Test Webhook**, but the receiving system did not show the test event.

Cause: The destination may be unavailable, the configured destination may be incorrect, or the receiving system may reject or ignore the event.

Solution: Open **Webhooks** and review **1. Webhook Config** to confirm the intended destination. Then check **2. Delivery Log** for the test attempt. If the configuration appears correct but delivery does not succeed, capture the test time and delivery details and contact **Support Agent** through **Operations**.

An integration update is missing from GitHub, GitLab, or Bitbucket

Problem: You expected activity from **GitHub**, **GitLab**, or **Bitbucket** to appear in Atloria, but it is missing.

Cause: The integration may not be connected as expected, the expected activity may not have been generated, or imported content may have completed with warnings.

Solution: Go to **Settings** → **Integrations** and confirm the affected provider. If parsed or imported information is involved, check whether the result says **Parse completed successfully with no errors** or **Parse completed with warnings**. Include the provider, project, time, and parse outcome when requesting help from **Support Agent**.

You need to know whether a delivery or integration creates a privacy concern

Problem: You are unsure whether information sent through an integration or webhook is handled appropriately.

Cause: This is a privacy and data-handling question, not necessarily a technical delivery failure.

Solution: Select **Privacy** and review the available privacy information. If you also have evidence of a failed delivery, separately review **2. Delivery Log** and report the technical details to **Support Agent**. Keeping privacy questions and delivery evidence separate helps ensure each concern is handled correctly.