

Maintain Fresh Documentation

September 10, 2026

Maintain Fresh Documentation

Use project changes and documentation freshness signals to identify where your documentation needs attention. This helps you prioritize updates before outdated information affects readers.

Overview

Atloria helps you maintain documentation as your project evolves. The Changes area is where you can review individual changes and use the available documentation signals to decide whether existing documentation should be updated, regenerated, or reviewed.

A freshness signal such as **1. Fresh (No Connections)** helps you understand the current documentation state for a change. Use this signal together with the change details: a change may be fresh because it has no documented connections yet, while a change connected to existing documentation may require a closer review after related work changes.

Changes can span several areas of a project, including **FRONTEND, BACKEND, DATABASE, API SCHEMA, CONFIG, DOCUMENTATION, and INFRASTRUCTURE**. When a change affects one of these areas, check whether the associated documentation still describes the current behavior and relationships accurately.

 *Screenshot pending: The project Changes page showing change records and documentation freshness signals.*

Prerequisites

Before you start, make sure you have:

- Access to the relevant Atloria project.
- Permission to review project changes and save documentation updates.

- A clear understanding of the change you are reviewing.
- Any required input from the people responsible for the affected frontend, backend, database, API schema, configuration, documentation, or infrastructure work.
- A decision about whether you will update the documentation directly, request changes, or generate documentation again.

Step-by-Step Instructions

1. Open the project that contains the documentation you want to review.
2. Open the project's **Changes** page.
3. Select the change you want to assess from the list of available changes.
4. Review the freshness signal shown for the selected change.
5. Identify whether the change is marked **1. Fresh (No Connections)**.
6. Review the change details to determine which project area is affected.
7. Check whether the change affects **DOCUMENTATION** directly.
8. Check whether the change affects a related area, such as **FRONTEND**, **BACKEND**, **DATABASE**, **API SCHEMA**, **CONFIG**, or **INFRASTRUCTURE**.
9. Compare the change with the documentation that users currently rely on.
10. Open **Documentation Hub** when you need to review the project's existing documentation.



Screenshot pending: The Documentation Hub used to review existing documentation before making an update.

11. Select **View Your Documentation** → when you are ready to inspect the published or generated documentation.
12. Update the documentation content if the change makes the current explanation incomplete, inaccurate, or obsolete.
13. Select **Save Changes** after you complete your documentation updates.
14. Select  **Request Changes** if the change needs clarification or correction before the documentation can be kept current.

15. Select **Generate Documentation** → when you need Atloria to generate documentation from the project's current codebase.
16. Review the generated output before treating the documentation as ready for readers.
17. Select **Export Documentation** if you need to distribute the updated documentation outside Atloria.
18. Return to the **Changes** page to continue reviewing other changes that may affect documentation freshness.

Tips and Best Practices

- Treat freshness as a review priority, not as a substitute for reading the change. A signal such as **1. Fresh (No Connections)** tells you about the documentation connection state, but you should still inspect the change details before deciding that no action is needed.
- Review related documentation whenever work crosses project areas. For example, an API schema change can affect frontend integration guidance, backend behavior descriptions, and API reference material. A database change can also affect operational procedures or data-related explanations.
- Pay particular attention to changes categorized as **DOCUMENTATION**. These changes may directly alter the content, structure, or expectations for project documentation.
- Use  **Request Changes** when the available change information does not give you enough confidence to update the documentation correctly. It is better to request clarification than to publish an assumption as fact.
- Save work frequently with **Save Changes**, especially when you are reviewing several related changes. This reduces the risk of losing a completed documentation update while you investigate additional changes.
- Use **Generate Documentation** → when manual maintenance is no longer efficient or when you need a new generated baseline. After generation, verify that the resulting documentation reflects the intended user-facing behavior.
- Review the project's documentation styles and templates when consistency is important. **The 4 Documentation Styles — path to 8+/10 each** and **Documentation Templates** can help you keep updates aligned with the rest of the documentation set.

 *Screenshot pending: The change detail page with the  Request Changes action available for incomplete or unclear changes.*

Related Pages

- [Documentation Hub](#)
- [Generate Documentation](#)
- [Documentation Automation](#)
- [Documentation Versions](#)
- [Documentation Analytics](#)
- [Export Documentation](#)
- [Documentation Templates](#)

Troubleshooting

The change does not provide enough information to update the documentation

Select  **Request Changes** and explain what information is missing. Request clarification about the affected feature, expected behavior, or relationship to existing documentation before saving an update.

The documentation appears fresh, but the change affects related functionality

Review the affected category and connected documentation manually. A freshness signal does not remove the need to assess changes that span areas such as frontend, backend, API schema, or infrastructure.

Generated documentation does not yet meet your needs

Review the generated content in **Documentation Hub**, make the necessary edits, and select **Save Changes**. If the underlying change is incomplete or unclear, use  **Request Changes** before finalizing the documentation.