

# Inspect Source Symbols

September 10, 2026

---

## Inspect Source Symbols

---

You can inspect source symbols while reviewing generated documentation to confirm that descriptions match the underlying product behavior. This helps you validate names, dependencies, and relationships before you publish or share documentation.

### Overview

Source-symbol inspection lets you compare documentation review content with the source material associated with it. Use this task when a generated explanation mentions a component, page, service, data model, or other source entity and you need to verify that the description is accurate.

During review, source analysis can organize findings by category. Categories may progress from **FRONTEND** through **BACKEND**, **DATABASE**, **API SCHEMA**, **CONFIG**, **DOCUMENTATION**, and **INFRASTRUCTURE**. These categories help you understand which part of the product a symbol belongs to before you decide whether its documentation needs revision.

You may also encounter relationship information that explains how symbols work together. For example, a source entity can **RENDERS** another component, **USES HOOK**, **CALLS API**, **DEPENDS ON** another entity, **USES SERVICE**, or **QUERIES MODEL**. Reviewing these relationships is especially important when validating user flows and feature behavior.



*Screenshot pending: The documentation review screen showing the View Source action and source inspection area.*

### Prerequisites

Before you start, make sure you have:

- Access to the relevant project and documentation review.

- A documentation item ready to review.
- Permission to open the review and source information for the project.
- A clear understanding of the documentation statement, workflow, or user behavior you want to validate.
- The source symbol name, if it is already referenced in the documentation you are reviewing.

## Step-by-Step Instructions

1. Open the project documentation review that contains the content you want to validate.
2. Select **Review & Publish** to enter the documentation review workflow.
3. Select **1 Review Summary** to locate the documentation section that needs source validation.
4. Read the documentation statement that refers to a source entity, behavior, or relationship.
5. Click **View Source** to open the source information for the item you are reviewing.
6. Select **Source** to focus on the available source content.
7. Identify the symbol named in the documentation, such as a component, hook, page, layout, route, context, provider, store, controller, service, repository, or middleware.
8. Review the symbol's category to determine whether it belongs to the frontend, backend, database, API schema, configuration, documentation, or infrastructure area.
9. Review the symbol's relationships to confirm how it interacts with other entities.
10. Compare the visible source behavior with the documentation statement you read in the review.
11. Return to the review by clicking **Back to Review**.
12. Update the documentation review content if the source inspection shows that the statement is incomplete, inaccurate, or missing an important dependency.
13. Select **Preview** to check how the validated documentation will appear to readers.

 *Screenshot pending: A source inspection view highlighting a symbol, its category, and relationships to other entities.*

## Tips and Best Practices

- Validate one claim at a time. For example, first confirm that a component renders another component, then separately confirm whether it uses a hook or calls an API.
- Treat entity type and relationship type as different pieces of information. A symbol might be a **COMPONENT**, while its relationship might be **RENDERS** another component or **USES HOOK**. Both facts may be needed for an accurate explanation.
- Follow the category sequence when reviewing broad behavior. Start with **FRONTEND** symbols for user-facing behavior, then check **BACKEND**, **DATABASE**, and **API SCHEMA** symbols when the documentation describes data handling or server-side behavior.
- Use relationship details to avoid overstating behavior. For example, **CALLS API** indicates a direct API interaction, while **DEPENDS ON** indicates reliance on another entity but does not necessarily describe a user-visible action.
- Review names carefully. Symbols can include components, hooks, contexts, stores, services, and repositories that have similar names but different responsibilities.
- Use **Preview** after making changes so you can confirm that source-based clarification remains understandable for the intended audience.
- If you need to validate a visual or formatting-related documentation change, use the available source view rather than assuming that generated content reflects the current source behavior.

## Related Pages

- [Review and Publish Documentation](#)
- [Preview Documentation](#)
- [Generate Documentation](#)
- [Add Code Examples](#)

- [Export Documentation](#)
- [User Guides](#)

## Troubleshooting

### You cannot find the source information for the documentation item

Click **View Source** from the specific review item rather than from a general documentation page. If source information is still unavailable, confirm that you opened the correct project and review content.

### The symbol name does not match the documentation wording

Review the visible symbol type and its relationships before changing the documentation. The documentation may use a user-friendly term, while the source uses a more specific entity type such as **SERVICE**, **STORE**, or **CONTEXT**.

### The relationship does not support the documentation claim

Return by selecting **Back to Review**, then revise the statement to reflect the relationship you verified. For example, describe a symbol as depending on another entity rather than calling it directly when the relationship is **DEPENDS ON**.