Analyze a Frontend Source

September 10, 2026

Analyze a Frontend Source

Analyzing a frontend source helps you prepare accurate context before generating user documentation. You can use the source view and documentation generation actions to identify user-facing screens, components, relationships, and relevant product behavior.

Overview

Frontend source analysis turns application interface code into documentation context. This context helps you understand which screens exist, which reusable interface elements they contain, and how users move between them. It is especially useful before you generate or regenerate User Guides, User Manuals, or other product documentation.

During analysis, Atloria identifies frontend entities such as components, pages, layouts, routes, contexts, providers, stores, and hooks. For a typical frontend review, you may see interface elements such as an avatar, user profile, loading indicator, statistics card, editor, source view, language selector, or comparison view.

The analysis also identifies relationships between items. For example, a user profile can render an avatar and statistics cards, while an editor toolbar can provide options to switch between a source view and a comparison view. Reviewing these relationships helps you describe what users see and do without documenting internal implementation details.

> 📷 *Screenshot pending: The documentation workspace showing the View Source action and documentation generation controls.*

Prerequisites

Before you begin, make sure you have the following:

- Access to the relevant Atloria project and its documentation workspace.
- A frontend source or project version that is available for review.
- Permission to view source information and generate documentation.
- A clear documentation goal, such as documenting a user profile screen, source editor, or documentation-generation workflow.
- An understanding of the intended audience for the output, such as end users, administrators, or product teams.

Step-by-Step Instructions

1. Open the project documentation workspace for the product you want to analyze.
2. Select **View Source** to open the available source context for the current documentation item.
3. Review the frontend entities shown in the source context.
4. Identify the user-facing screen or interface area you want to document.
5. Record the visible elements that users interact with on that screen.
6. Check whether the screen includes reusable interface elements such as an avatar, loading indicator, statistics card, editor, or selection control.
7. Identify the main user action supported by the screen.
8. Review the relationships shown for the selected frontend entity.
9. Note relationships that affect the user experience, such as one interface area rendering another or using shared application context.
10. Classify the selected item according to its frontend role.
11. Treat items identified as **Component** as reusable interface elements.
12. Treat items identified as **Page** or **Layout** as higher-level user-facing areas that may contain multiple components.
13. Treat items identified as **Context**, **Provider**, or **Store** as shared behavior that may affect several screens.
14. Review the parser category for the source item.
15. Select **FRONTEND** when you are preparing context for visible application screens and interactions.

16. Continue through other available categories only when they affect the user task you are documenting.
17. Use **Source** when you need to return to the source-oriented view of the current material.
18. Review editor-related controls when the source context includes editable or formatted content.
19. Select the language control when you need to view content using the appropriate language formatting.
20. Switch between source and comparison views when both views are available.
21. Verify whether a comparison view highlights changes that could affect the user instructions.
22. Describe only the current user-visible behavior when the source shows a change that has not been confirmed in the product interface.
23. Select **User Guides** when you are ready to prepare end-user documentation from the analyzed context.
24. Select ▶ **Generate Docs** to create documentation from the current analysis.
25. Review the generated documentation for accurate screen names, actions, and user-facing terminology.
26. Select **Regenerate Docs** if the generated content does not reflect the source context or intended user task.
27. Select ↻ **Re-generate** when you need to run the generation process again after updating your analysis.
28. Select ✦ **Regenerate AI docs** when you want Atloria to produce an updated AI-assisted documentation draft.
29. Select **Export Documentation** when the reviewed documentation is ready to share outside the workspace.

> 📷 *Screenshot pending: A frontend source analysis view showing entity classifications, relationships, and the source or comparison display.*

Tips and Best Practices

- Focus on one user task at a time. For example, document viewing a user profile separately from editing source content or generating documentation. This keeps instructions clear and reduces unnecessary technical detail.
- Start with the **FRONTEND** category when your goal is a user manual. Frontend analysis is the most direct source of information about what users see, including pages, controls, labels, loading states, and displayed values.
- Use entity types to organize your review. A **Component** is usually a reusable visible element, while a **Page** or **Layout** provides the broader user journey. This distinction helps you avoid writing a separate manual page for every small visual element.
- Pay attention to rendering relationships. When one item renders another, document the visible result rather than the underlying relationship. For example, if a user profile displays an avatar and statistics cards, explain what users see in the profile and what information the cards display.
- Review shared context carefully. Relationships such as **USES CONTEXT**, **PROVIDES CONTEXT**, **USES STORE**, or **DISPATCHES ACTION** can explain why content is consistent across screens. In the user manual, describe the resulting behavior only if users can observe it.
- Use comparison mode when documenting updates. A source-and-difference view can help you spot changed labels, fields, or user flows before you regenerate documentation. Confirm that the changed behavior is present in the product before adding it to user instructions.
- Preserve exact labels. When a control is labeled **View Source**, **Regenerate**, **Generate More**, or **Export Documentation**, use the same wording in your documentation. Exact labels make instructions easier to follow.
- Regenerate only after refining context. If the first generated draft is incomplete, return to **View Source**, review the frontend entities and relationships, then use **Regenerate Docs** or ✦ **Regenerate AI docs**.
- Keep internal details out of the final user guide. Analysis can identify relationships such as **RENDERS**, **USES HOOK**, **CALLS API**, or **DEPENDS ON**, but end users need to know what appears on screen and which action to take.

Related Pages

- Generate User Documentation

- Review and Regenerate Documentation
- Export Documentation
- Compare Source Changes
- Prepare Documentation Context

Troubleshooting

You cannot find the source analysis view

Select **View Source** from the documentation workspace. If the action is not available, confirm that you opened the correct project documentation item and that you have permission to view source context.

The generated documentation describes technical details instead of user actions

Return to **View Source** and focus your analysis on frontend screens, visible controls, and user outcomes. Then select **Regenerate Docs** or ✦ **Regenerate AI docs** to produce a revised draft.

The generated content does not reflect recent interface changes

Review the source and comparison views for updated labels, controls, and relationships. Confirm the intended user-facing behavior, then select ↻ **Re-generate** or **Regenerate** to create an updated version.

You are unsure whether an identified relationship belongs in the manual

Include the relationship only when it changes what users see or do. For example, document that a profile shows an avatar or loading state, but do not include internal relationship names unless they are visible in the application.