Configure Source Analysis

September 10, 2026

Configure Source Analysis

Configure Source Analysis to control which parts of your connected project Atloria examines when building documentation intelligence. These settings help you focus generated documentation on the most relevant source areas and analysis categories.

Overview

Source Analysis is configured through the project setup flow, where you move through **1. General Settings**, **2. Parser Settings**, and **3. Template Settings**. The parser settings are where you define the analysis focus for the source material that Atloria will use.

Use Source Analysis when you want documentation intelligence to reflect a particular type of project content, such as frontend components, backend services, databases, API schemas, configuration, documentation, or infrastructure. A focused configuration reduces unrelated results and helps the generated documentation match the purpose of your project.

Atloria's analysis recognizes different kinds of source entities, including components, hooks, pages, layouts, routes, contexts, providers, stores, controllers, services, repositories, and middleware. It can also identify relationships between entities, such as when one component renders another component or uses a hook.

> 📷 *Screenshot pending: Project setup flow showing General Settings, Parser Settings, and Template Settings.*

Prerequisites

Before you configure Source Analysis, make sure you have:

- Access to the project you want to document.
- A connected source integration, such as GitHub, GitLab, or Bitbucket.

- Permission to open the project's **Settings** or setup configuration.
- A clear idea of the source areas you want Atloria to prioritize, such as frontend, backend, database, or infrastructure.
- A project selected as the **Focused Project**, if your workspace contains more than one project.

Step-by-Step Instructions

1. Open the project you want to configure.
2. Select **Settings** from the available project actions.
3. Open the setup or configuration area that displays **1. General Settings**, **2. Parser Settings**, and **3. Template Settings**.
4. Select **2. Parser Settings**.
5. Review the available analysis categories for the source scope you want documented.
6. Select the frontend category when you want Atloria to analyze user-interface source, including components, hooks, pages, layouts, contexts, providers, and stores.
7. Select the backend category when you want Atloria to analyze server-side source, including controllers, services, repositories, and middleware.
8. Select the database category when database-related source should be included in the documentation intelligence.
9. Select the API schema category when your documentation should include API definitions and schema-related source.
10. Select the configuration category when application configuration should be considered during analysis.
11. Select the documentation category when existing documentation source should be included in the analysis scope.
12. Select the infrastructure category when deployment, environment, or infrastructure source is relevant to your documentation.
13. Confirm the selected scope before continuing to **3. Template Settings**.

14. Select **View Source** when it is available and you need to inspect the underlying source content associated with an analysis result.
15. Select **Source** when it is available to switch from a rendered or compared view to the source view.
16. Continue through **3. Template Settings** after you finish selecting the analysis scope.
17. Select **Generate Documentation** → when you are ready to create documentation using the configured analysis settings.

> 📷 *Screenshot pending: Parser Settings screen showing available source analysis categories and selected scope.*

Tips and Best Practices

- Start with the categories that match your documentation goal. For example, select frontend and backend categories for an application architecture guide, or select API schema and backend categories for API-focused documentation.
- Avoid selecting every category unless you need a broad project overview. A narrower scope can make the resulting documentation more focused and easier to review.
- Include **CONFIG** and **INFRASTRUCTURE** when operational behavior matters. These areas can provide useful context for how the application is configured and deployed.
- Include **DOCUMENTATION** when your repository contains material you want Atloria to use alongside source analysis. This is useful when existing written guidance should inform generated output.
- Use **View Source** to verify that an analysis result is based on the source you expected. This is especially helpful when reviewing relationships between entities.
- Review detected relationships in context. For example, a component may render another component or use a hook; these relationships can make architecture and feature documentation more complete.
- If your project includes multiple technologies, select the categories relevant to each area of the project. Atloria can recognize common frontend entities such as React, Vue, or Angular components, as well as custom hooks and related entities.

Related Pages

- Connect a Source Integration
- Configure Project Settings
- Configure Documentation Templates
- Generate Documentation
- Export Documentation
- Configure CI/CD

Troubleshooting

Parser Settings are not available

1. Open the correct project before selecting **Settings**.
2. Confirm that you have permission to manage project configuration.
3. Verify that the project has a connected source integration before attempting to configure analysis.

The generated documentation does not cover an expected area

1. Return to **2. Parser Settings**.
2. Select the missing analysis category, such as frontend, backend, database, API schema, configuration, documentation, or infrastructure.
3. Select **Generate Documentation** → again after updating the scope.

An analysis result does not appear to match the expected source

1. Select **View Source** from the result actions.
2. Review the source shown under **Source**.
3. Adjust the selected parser categories if the result is being drawn from an unintended part of the project.