

Evaluating Supported Languages and Parser Availability

August 25, 2026

Opening the language support catalog

To review parser availability in Atloria, open the workspace where you evaluate code parsing and language support, then go to the **Supported Languages** page. This page is the main catalog for checking whether Atloria recognizes the languages, frameworks, and source formats used in your documentation environment before you upload code or connect a repository. If you already worked through [Evaluating Language Support Before Parsing Code](#), use that preparation list here instead of rebuilding it from scratch.

On the **Supported Languages** page, scan the catalog row by row. Each entry is meant to help you judge fit, not just confirm that a language name appears somewhere in the list. Focus on the visible columns that tell you what Atloria can handle for each entry:

Column	What to look for
Language	The main language or format name shown in the catalog
Framework / Variant	A more specific label when support differs by framework, flavor, or syntax style
Parser Count	How many parsers Atloria lists for that language or framework entry
Coverage Indicator	Whether support appears broad or more limited

This catalog is especially useful when one language appears in several forms. For example, the same language may be listed once as a general language entry and again with framework-specific entries. In that case, do not assume every row offers the same parsing depth.

Use the page as a comparison screen. Your goal is to confirm whether Atloria supports the exact combinations your team uses, including framework-specific documentation sources and mixed repositories.

[SCREENSHOT: Supported Languages page showing the catalog with Language, Framework or Variant, Parser Count, and Coverage Indicator columns]

Reading parser counts and coverage indicators

The **Parser Count** column tells you how many parser options Atloria has for a specific language and framework combination. This number matters because parser availability can vary even when the language name looks familiar. A language with several parser options may support more than one ecosystem or syntax style, while another entry with the same language name may have fewer options.

When you review a row, read **Parser Count** together with the **Coverage Indicator**. These two values work best as a pair:

- **Parser Count** shows how many parser options are available.
- **Coverage Indicator** shows whether support appears broad or more limited for that entry.

Do not rely on the language name alone. If you only check the **Language** column, you may miss important differences between a general language entry and a framework-specific one. For example, one row may show stronger coverage for a framework variant than the base language entry, or the opposite may be true.

As you compare rows, pay attention to repeated language names with different framework labels. That usually means Atloria separates support by ecosystem, syntax variant, or framework target. In those cases, the parser count may differ from one row to another, and the coverage indicator may also change. That difference is the signal you need when deciding whether a repository is likely to parse cleanly.

A practical way to read the catalog is:

1. Find the exact language or format.
2. Check whether there are multiple framework or variant rows.
3. Compare the **Parser Count** values across those rows.
4. Use the **Coverage Indicator** to judge whether support looks broad enough for your content.

[SCREENSHOT: Catalog rows showing the same language listed with different framework or variant entries and different parser counts]

Filtering languages by framework and support level

When the catalog contains many entries, use the **search field** first. Enter the language name, markup format, or framework your team uses in repositories and documentation sources. This is the fastest way to narrow the list before you start comparing parser counts. If a language has several related entries, try both the broad language name and the framework name to make sure you are looking at the most relevant rows.

After searching, use any framework-related filtering options on the page to narrow the results further. This is especially helpful when Atloria lists both core language support and framework-specific parsing for the same language family. Instead of reviewing every row manually, you can focus on the exact framework or syntax style used by your team.

While filtering, keep your eye on two columns:

- **Parser Count**
- **Coverage Indicator**

These columns help you separate stronger support from entries that may need closer review. If your documentation environment includes several repositories or mixed source types, scan or sort the **Parser Count** column to find the entries with deeper parser coverage. That gives you a quick view of where Atloria appears strongest.

Filtering is most useful when your team works across multiple stacks. For example, you might search one language, then refine by framework, then compare the remaining rows by parser count and coverage. Once the list is narrowed, review the coverage labels carefully. A filtered result can still include combinations with limited support, so do not stop at the search result alone.

Use this workflow:

1. Enter a language, format, or framework in the **search field**.
2. Narrow the results with framework-related filters if available.
3. Scan the **Parser Count** column for stronger coverage.
4. Confirm the **Coverage Indicator** before treating an entry as rollout-ready.

[SCREENSHOT: Supported Languages page with a search field in use and filtered results showing framework-specific entries]

Comparing Atloria support against your documentation stack

Before you decide whether Atloria is a good fit for your parsing workflow, compare the catalog against the real content sources your team already uses. Start with a simple working list of the languages, frameworks, and source formats that appear across your documentation pages, code comments, and generated reference material. Keep the list practical. Focus on what your team actively maintains, not every possible language that exists somewhere in your organization.

Once you have that list, open the **Supported Languages** catalog and match each item to a row in Atloria. Look for the exact language name first, then confirm the **Framework / Variant** label. This step matters because a broad language match is not always enough. If your team depends on a specific framework or syntax style, the matching row should reflect that exact combination.

As you compare, review these points for each item:

Check	What to confirm in Atloria
Language match	The language or format appears in the catalog
Framework match	The framework or variant label matches your actual stack
Parser Count	The row shows enough parser availability for that combination
Coverage Indicator	The support level looks broad enough for your expected content

Flag any gaps as you go. A gap might mean the language appears in the catalog but the framework label does not match, the parser count is lower than expected, or the coverage indicator suggests only partial support. Those are the entries that deserve extra attention before rollout.

Prioritize your review by business importance. Start with the languages and formats tied to your highest-volume documentation or your most important reference content. If those entries look strong in Atloria, you can move forward with more confidence. If they look weak, that is a sign to slow down and validate further before adoption.

Deciding whether parser coverage is sufficient for rollout

After you compare your stack to the catalog, decide which parts of your content are safe to onboard first. In Atloria, the strongest rollout candidates are usually the entries that combine a clear language or framework match, a solid **Parser Count**, and a strong **Coverage Indicator**. Those combinations are the lowest-risk starting point because they suggest broader parser availability for the content you want to bring in.

Treat entries with limited or partial coverage more carefully. A language may appear in the catalog, but that does not always mean every syntax pattern in your repositories will behave the same way. If your team uses custom syntax, framework extensions, or mixed-format repositories, a limited coverage indicator is a sign to validate those areas before making them part of a larger rollout.

Multiple parser options for the same language can also improve confidence, especially if your teams work across several framework variants. When Atloria shows more than one parser path for a language family, that can be a good sign for organizations managing different documentation sources under one program. Still, use the exact framework row that matches your content rather than assuming all related rows offer equal support.

A practical rollout decision usually looks like this:

- Start with languages and frameworks that show strong coverage and healthy parser counts.
- Hold back entries with partial coverage until you validate them in real project content.
- Note any repositories that depend on uncommon syntax or mixed source structures.
- Share those findings with the people planning onboarding, project setup, and documentation scope.

If you need help planning the broader rollout sequence after this review, the project setup and onboarding guidance in [Creating Projects and Completing Onboarding](#) can help you turn parser findings into an adoption plan.

Handling gaps in language or framework support

If you cannot find a language right away, start by trying different search terms in the **search field**. Some entries may be easier to find under a framework name, a syntax family, or a more specific variant label rather than the broad language name your team uses in conversation. Search again using alternate naming before you assume the language is missing.

If the language appears but the support looks limited, compare nearby entries for related frameworks or variants. Atloria may list a more specific parser target that better matches your content than the general language row. In that case, the more specific entry may show a different **Parser Count** or a stronger **Coverage Indicator** than the broader listing.

When parser counts are lower than expected, identify which repositories actually depend on that language or framework. Focus on the content that matters most: high-traffic documentation, generated reference pages, or repositories that drive published documentation. This helps you separate a minor gap from a rollout blocker.

If support still feels unclear after searching and comparing, capture the exact details shown in the catalog so your team can review them internally. Record:

- The **Language** name exactly as shown
- The **Framework / Variant** label
- The **Parser Count**
- The **Coverage Indicator**

That record gives project leads, documentation managers, or admins a concrete basis for follow-up. It is much more useful than a general note saying a language is “not supported enough.”

For broader admin-side review of planning and oversight, see [Using the Admin Workspace](#). If your team needs to continue the evaluation process after identifying gaps, keep those exact catalog details with your rollout notes so they can be checked again during project planning.

Overview

This page in Atloria is designed for one specific job: helping you decide whether the languages and frameworks in your documentation environment are supported well enough for parsing. The **Supported Languages** catalog is not just a reference list. It is a decision tool that lets you compare your real stack against visible support signals before you upload code, connect repositories, or commit to a wider documentation rollout.

The most important parts of the catalog are the rows and columns that show:

- **Language**
- **Framework / Variant**
- **Parser Count**
- **Coverage Indicator**

Together, these tell you whether Atloria recognizes a language broadly, whether support changes by framework, and whether there are multiple parser options available for a specific combination. That is especially important for teams working

with mixed repositories, framework-heavy projects, or generated technical documentation.

Use this document when you need to answer questions such as:

- Does Atloria support the exact language and framework combination we use?
- Is there just one parser option, or several?
- Does the coverage indicator suggest broad support or limited support?
- Which parts of our documentation stack are safest to onboard first?

This guide builds on the evaluation approach introduced in [Evaluating Language Support Before Parsing Code](#). Here, the focus is narrower: reading the catalog itself and using parser counts and coverage indicators to make a practical support decision.

After you finish this review, continue with [Evaluating Supported Languages and Framework Coverage](#) to look more closely at how support depth affects framework-heavy documentation environments.

Prerequisites

Before you use the **Supported Languages** catalog in Atloria, make sure you have the basic information needed to compare the catalog against your real documentation environment. You do not need a technical inventory, but you do need a clear list of the languages, frameworks, and source formats your team actually relies on.

Have these items ready:

- Access to Atloria and the workspace where you review language support
- A list of the languages used in your documentation-related repositories
- The framework names or syntax variants tied to those languages
- Notes on any generated reference content, code comments, or markup formats your team depends on
- A rough sense of which repositories or content sources are most important to your rollout

It also helps if you already know which content is business-critical. When you compare parser availability, the most important question is not whether every possible language appears in the catalog. It is whether the languages that matter most to your documentation program are covered well enough to support onboarding.

If you have not yet organized that information, review [Evaluating Language Support Before Parsing Code](#) first. That document helps you prepare your stack list so you can

use the catalog efficiently instead of searching one term at a time without context.

You may also want access to the project and documentation owners who know which repositories produce your most important docs. Their input is useful when parser counts or coverage indicators suggest that some parts of the stack are stronger candidates for early rollout than others.