ATLORIA

Viewing Supported Languages and Parser Coverage

August 25, 2026

Opening the language and parser coverage view

To review language support in Atloria, start from the signed-in workspace and open the area where you work with technical documentation or code parsing. If you are still getting familiar with account access and workspace entry, see Accessing and Registering Your Atloria Account and Understanding Account Entry Points and Session Navigation.

1. Sign in with your **Email** and **Password** on the login screen, then select **Log in**.
2. After Atloria opens your main workspace, go to the project or parsing area used for technical documentation work.
3. Open the screen that shows the **Supported Languages** list or support matrix for parsing.
4. If the page includes tabs or filters, switch to the view that shows all available language entries before narrowing the list.

On this screen, focus on the main regions that help you evaluate support quickly:

- A **language name** area that lists each programming language.
- A **framework** or **ecosystem** area that shows related technologies under that language.
- A **parser status** area that tells you whether Atloria can process that language or framework.
- A **coverage** area that shows how complete the support is.

Some workspaces may also include a **Search** field near the top of the list, along with tabs or filter chips for views such as all entries, parser-ready items, or partial support results. These controls help you reduce a long list to only the entries that matter for your project.

[SCREENSHOT: Supported Languages screen showing the search box, language list, parser status badges, and coverage column]

If you do not see the language support view yet, you may need to enter the code parsing workspace first. For that workflow, use Uploading and Parsing Code in the Workspace.

Reviewing which languages and frameworks are supported

Once the **Supported Languages** screen is open, use the list to confirm whether Atloria recognizes the language and framework combination used in your project. This is especially important when your repository includes both a base language and one or more framework layers.

1. Look down the language list to find the programming language you plan to document.
2. Use the **Search** field to narrow the table if the list is long.
3. Check the framework or ecosystem labels shown on the same row or in related rows.
4. Review the support badge or label beside each entry before deciding what content to upload or parse.

The search box is the fastest way to confirm support for common languages such as **Python**, **JavaScript**, **TypeScript**, **Java**, or **Go**. After you type a language name, review the matching rows carefully. In Atloria, a language may appear with several framework-specific entries, and those entries may not all have the same support level.

For example, a base language can appear separately from framework coverage. That means you may see support for the language itself, while a framework built on top of it has a different status. When framework tags or labels appear, use them to tell the difference between general language support and support for a specific stack.

Watch for support badges such as:

- **Fully supported** for broad, reliable parsing coverage
- **Partially supported** when only some structures are recognized
- **Not available** when Atloria does not currently support that entry

These labels help you avoid assuming that support for one row applies to every related framework. If your project includes several technologies, compare each row instead of checking only the top-level language name.

[SCREENSHOT: Search field filtering the Supported Languages table for a language and showing framework-specific rows]

For a deeper comparison of language and framework fit, continue with Evaluating Language and Framework Support in Atloria.

Checking parser availability before uploading or indexing content

Before you upload source material or start a parsing job, confirm that the language or framework you want to use has a parser available in Atloria. This step helps you avoid spending time on files that Atloria cannot process well yet.

1. Find your language or framework in the **Supported Languages** list.
2. Look at the **Parser** column or status badge on that row.
3. Open the row details, expandable panel, or side panel if more information is available.
4. Review any parser notes before choosing files or repositories for ingestion.

The parser status is the most direct signal for readiness. Depending on the entry, Atloria may show a status such as:

- **Available** if the parser is ready to use
- **Unavailable** if parsing is not currently offered
- **Beta** if support exists but may still be developing
- **Limited** if only part of the language or framework is handled

When you can open a row for more detail, look for parser-specific notes. These details may explain which file types are recognized or whether support applies only to certain framework patterns. If your project includes several candidate repositories, compare parser availability across them before deciding what to process first.

This is especially useful for mixed-language projects. One repository may include a language with strong parser support and another with only limited support. In that case, start with the source that has the clearest **Available** status so your technical documentation results are more predictable.

[SCREENSHOT: Expanded language row showing parser status, notes, and supported file details]

If you are preparing to bring code into Atloria next, the related workflow is covered in Uploading and Parsing Code for Documentation Workflows.

Understanding coverage indicators and support depth

Parser availability tells you whether Atloria can process a language at all, but the **Coverage** field tells you how much Atloria is likely to recognize once parsing begins. This is where you judge support depth, not just support presence.

1. Find the **Coverage** column or coverage label for your selected language or framework.
2. Check whether coverage is shown as a label, percentage, or both.
3. Compare language-level coverage with framework-level coverage if Atloria shows them separately.
4. Read any notes attached to the coverage result before finalizing your source choice.

Coverage may be shown with labels such as **Full coverage**, **Partial coverage**, **Experimental coverage**, or **No coverage**. Treat these as practical guidance for documentation planning:

- **Full coverage** usually means Atloria recognizes most expected structures for that entry.
- **Partial coverage** means some parts may parse correctly while others may be skipped.
- **Experimental coverage** suggests support is still developing and results may vary.
- **No coverage** means you should not rely on parsing for that entry.

If Atloria shows separate values for the language and its framework, read both. A language may have strong base support while the related framework has narrower recognition. That difference matters when your repository depends heavily on framework conventions rather than plain source files.

Coverage notes are just as important as the headline label. They can point out known gaps, such as unsupported syntax, limited metadata extraction, or missing framework conventions. Those notes help explain why a project might parse only part of what you expect.

[SCREENSHOT: Coverage column with labels such as Full, Partial, and Experimental, plus expanded notes]

For more detail on interpreting these support levels, see Understanding Parser Coverage and Stack Compatibility.

Comparing support details to choose the right source material

After you review language names, framework tags, parser status, and coverage, use that information to decide what source material should go into Atloria first. This comparison step is useful when a project contains source code, generated reference material, and framework-specific files side by side.

1. Compare the rows for each language and framework used in your project.
2. Prioritize entries with an **Available** parser and stronger coverage.
3. Decide whether to start with source code, generated docs, or framework files based on the support details shown.
4. Note any weak areas before your team begins a broader documentation rollout.

A simple comparison often reveals which content will give you the best results. If one language shows a ready parser and high coverage while another shows limited or experimental support, start with the stronger option. That gives your team a more reliable first pass for technical documentation.

Use all three signals together:

- **Framework tags** tell you whether support applies to your actual stack
- **Parser status** tells you whether Atloria can process it now
- **Coverage details** tell you how complete the results are likely to be

This matters most in mixed-language repositories. A project may include one well-supported language and another with low coverage. In that situation, you may decide to parse only the stronger portion first, then handle the weaker area through existing documentation pages or manual authoring in Atloria.

It also helps to record unsupported or low-confidence entries early. Documentation managers, project owners, and admins can then adjust scope, choose a different source set, or delay certain parsing tasks until support improves.

[SCREENSHOT: Side-by-side comparison of several language rows with parser and coverage differences]

If you are planning documentation work across a project, pair this review with Using Code Parsing Results to Support Technical Docs.

Resolving missing languages, unavailable parsers, and low coverage results

If you cannot find the language you expected, or the support level looks weaker than planned, use the controls on the **Supported Languages** screen to confirm what

Atloria is actually showing before you escalate the issue.

1. Enter the language name again in the **Search** field and check the spelling.
2. Clear any active filters, chips, or tabs that may be hiding entries.
3. Reopen the language row and review the parser and coverage notes.
4. Capture the visible details if you need to raise the issue with your Atloria owner or support contact.

When a language does not appear at all, the most common causes are a narrow filter or a search term that does not match the listed name. Reset the view to show all entries, then search again. If the language still does not appear, Atloria may not currently list support for it.

If the parser status shows **Unavailable**, do not assume the entire language family is unsupported. Check whether support is limited to certain frameworks or file types. A related row may still be usable for part of your project.

If coverage is lower than expected, open the notes and look for explanations such as:

- unsupported syntax
- experimental parser status
- framework-specific limitations
- reduced metadata extraction

When the page still leaves questions unanswered, take screenshots of the exact row, the parser badge, and the coverage details. That gives your internal Atloria owner or support team the context they need to review the issue accurately.

[SCREENSHOT: Language row with unavailable parser badge and visible coverage notes]

For related admin-side review of activity and controls, see Reviewing Security and Audit Controls and Monitoring Administrative Analytics and Activity.

Overview

Atloria's **Supported Languages** view helps you answer one practical question before you begin technical documentation work: *Will this source material parse well enough to use?* Instead of uploading code and discovering problems later, you can review language entries, framework labels, parser status, and coverage indicators in one place.

Use this screen when you are deciding whether to process a repository, compare two possible source sets, or confirm whether a framework is supported separately from its base language. The most helpful parts of the page are usually the **Search** field, the language list, the **Parser** status area, and the **Coverage** details. Together, these show both readiness and depth of support.

Keep these distinctions in mind while reviewing the list:

- A language can be listed even if a specific framework under it has different support.
- A parser can be available while coverage is still partial.
- Coverage notes may explain gaps that are not obvious from the badge alone.
- Mixed-language projects should be checked row by row, not assumed from one top-level match.

This screen is especially useful before workflows such as code upload, technical reference generation, or project planning for documentation rollout. If your team depends on parsed output, reviewing support first can help you choose better source material and avoid rework.

For the broader parsing workflow after this review, see Managing Code Parsing Workspace Sessions and Reviewing Parsed Code Results and Reference Coverage.

The next step in this section is Evaluating Language and Framework Support in Atloria.

Prerequisites

Before using the **Supported Languages** view in Atloria, make sure you have the basic access and context needed to interpret the results correctly.

- You can sign in to Atloria with your **Email** and **Password**.
- You can reach the authenticated workspace after login.
- You have access to the project area, technical documentation area, or code parsing workspace where the language support list is shown.
- You know the main languages or frameworks used in the repository or documentation source you plan to review.
- You are able to compare your project stack with the entries shown in the **Supported Languages** list.

It also helps to gather a short list of what you want to verify before opening the page, such as:

- the primary programming language
- any major framework used by the project
- whether you need parser-ready support now or are only checking future fit
- whether your repository includes multiple languages

If you are new to Atloria navigation, review Working with Project Lists and Dashboards and Understanding Project Navigation and Linked Workspaces. If you have not yet connected this review to a parsing workflow, Using Code Parsing to Support Technical Documentation gives the larger context.

You do not need to upload code before checking support. In most cases, it is better to review parser availability and coverage first, then decide what content to bring into Atloria.