

Using API Reference Pages in Published and Project Views

August 25, 2026

Understanding where API reference pages appear

In Atloria, the same API reference page can appear in two places: inside a project workspace and in published documentation. The content is meant to describe the same endpoint, but the reason you open it is different in each view.

Inside the **project view**, API reference pages are part of your working documentation. This is where technical writers and documentation managers review structure, check wording, and confirm that each endpoint page is complete before release. You usually reach these pages while browsing technical documentation within a project, as covered in [Managing Technical Documentation Browsing Inside Projects](#). In this view, you are checking whether the page is ready to be shared publicly.

In the **published view**, the page is presented for readers who need to understand how an endpoint works. Public readers use it to look up request details, required fields, authentication rules, and response formats. They are not reviewing draft quality—they are trying to find answers quickly.

No matter which view you open, the page usually has the same main reading areas:

- A **left navigation** or sidebar for moving between API sections and endpoint pages
- An **endpoint heading** that names the operation
- A **method badge** such as **GET**, **POST**, **PUT**, **PATCH**, or **DELETE**
- The **request path**
- **Parameters** details
- A **Request Body** section
- **Responses** with status codes and examples
- **Authentication** details when access is required

[SCREENSHOT: API reference page showing sidebar, endpoint title, method badge, path, parameters, request body, responses, and authentication sections]

The biggest difference is context. In the project workspace, you read with a reviewer's eye. In published documentation, you read like an end user trying to complete a task or confirm how an API call behaves.

Finding the right endpoint in project and published navigation

When you need a specific endpoint in Atloria, start with the navigation panel and work from the section level down to the individual operation page. API reference pages are usually grouped so you can browse by topic rather than scanning one long list.

In the **project view**, use the project's technical documentation navigation to move through the API reference structure. Open the relevant section, then continue through any nested groups until you see the endpoint title you need. This is useful when you are reviewing whether endpoints are filed under the right section names and whether the structure makes sense before publishing.

In the **published view**, readers use the public documentation sidebar or table of contents in the same general way. The goal is usually faster: find the right operation, open it, and read only the details needed for the current task. If the sidebar includes multiple groups with similar names, pay close attention to the endpoint title instead of relying on section names alone.

To confirm that you opened the correct page, check these items at the top of the endpoint page:

1. Read the **endpoint title**.
2. Check the **method badge** such as **GET** or **POST**.
3. Compare the **request path** shown beside or below the method.
4. Make sure the page purpose matches what you were looking for, such as retrieving data, creating a record, updating a record, or deleting one.

Long API reference pages can be easier to scan if Atloria shows in-page links or section anchors. Use those links to jump directly to areas such as:

- **Parameters**
- **Request Body**
- **Responses**
- **Authentication**
- **Examples**

[SCREENSHOT: Sidebar navigation with an API group expanded and one endpoint page selected]

If you are reviewing structure rather than just reading content, compare how the page is labeled in the project navigation and how it appears in the published sidebar. That helps you catch confusing titles before readers see them.

Reading endpoint details and understanding what they mean

The fastest way to understand an API reference page in Atloria is to read it from the top down: method and path first, then inputs, then outputs.

Start with the **method** and **path** line. The method badge tells you what kind of action the endpoint performs at a glance:

- **GET** usually means retrieve information
- **POST** usually means create something new
- **PUT** usually means replace existing data
- **PATCH** usually means update part of existing data
- **DELETE** usually means remove something

The **path** tells you where the request is sent. If the path includes a variable segment, that usually means part of the path must be filled in with a specific value.

Next, read the **Parameters** section carefully. Separate the parameters by where they belong:

Parameter location	What it means
Path parameters	Values included directly in the path
Query parameters	Values added to refine or filter the request
Header parameters	Values passed in request headers
Required or optional	Whether the request can work without that field

When you move to **Request Body**, look for the field list and any example payload. Focus on:

- **Field names**
- **Data types**
- **Required markers**
- **Nested objects**

- **Example values**

If the request body includes grouped fields or nested sections, read from the top level first and then drill into the nested object details.

Finally, review the **Responses** section. Match each **status code** to its meaning and example. Common patterns include:

- **200** for a successful read or update
- **201** for a successful creation
- **400** for an invalid request
- **401** for an authentication problem
- **404** when the requested item cannot be found

[SCREENSHOT: Endpoint page with method badge, path, parameter table, request body schema, and response examples highlighted]

Reading in this order helps you answer the most common questions quickly: what this endpoint does, what you must send, and what you should expect back.

Using API reference pages to answer common reader questions

Most people open an API reference page in Atloria because they need one specific answer. Instead of reading every section, go straight to the part of the page that matches the question you are trying to solve.

If you need to know whether an endpoint requires sign-in or a token, open the **Authentication** section first. Look for a security label, an authorization requirement, or a note that explains what kind of authenticated access is expected. If the page shows an **Authorization** header requirement, treat that as essential information before reviewing anything else.

If you need to know which values must be included, check both **Parameters** and **Request Body**. Required items are usually marked clearly. Do not assume that a field is optional just because it appears in an example. The required indicator in the table or schema is the more reliable guide.

If your question is “What do I get back when this works?”, go to the main success response in the **Responses** section. In many pages, that will be the first successful status code shown. Read the status code, then compare the schema and example body so you understand both the structure and the sample output.

If you are troubleshooting a failed request, compare the documented error responses. Focus on status codes such as:

- **400** for invalid input
- **401** for missing or invalid authentication
- **404** for a missing resource

The response example or description often explains what kind of problem that status code represents.

A practical reading pattern is:

1. Open **Authentication** to confirm access requirements.
2. Review **Parameters** and **Request Body** for required inputs.
3. Read the primary success response.
4. Scan error responses for likely failure cases.

[SCREENSHOT: API reference page with Authentication, required fields, success response, and error responses marked]

This approach is especially useful in published documentation, where readers usually want a direct answer rather than a full page review.

Reviewing API reference pages before publishing them

Before an API reference page goes live in Atloria, review it in the project workspace as if you were a public reader seeing it for the first time. The goal is not just to confirm that the page exists, but to make sure it is understandable without extra explanation.

Begin at the top of the page. Check that the **endpoint title** is clear, the **method badge** is visible, and the **request path** is correct. If the title is vague or the path does not match the page purpose, readers may select the wrong endpoint even when the navigation is technically correct.

Then move through the main sections and confirm that each one is present and readable:

- **Parameters** should show where each value belongs and whether it is required
- **Request Body** should include field details and examples when a body is needed
- **Responses** should include success and error status codes
- **Examples** should be easy to scan
- **Authentication** should be visible when access is required

After checking the project page, preview the same content in its **published layout**. This matters because a page that looks complete in the project workspace may be harder to read once it appears in the public documentation sidebar. Confirm that navigation labels are clear, section anchors jump to the right place, and schema tables remain readable.

Use the project view to catch quality issues such as:

- Missing descriptions
- Incomplete field details
- Inconsistent naming between title, path, and navigation
- Response examples that do not match the documented response structure

[SCREENSHOT: Project view and published preview of the same API reference page shown side by side]

If you are doing a broader release check, this review fits well alongside your version and publishing workflow in Atloria. The key question is simple: can an outside reader understand this endpoint page without needing internal context?

Fixing common problems when pages are hard to read or navigate

When an API reference page in Atloria feels confusing, the problem is usually one of four things: the page is hard to find, the request is unclear, the published layout behaves differently, or the response details are too vague.

If readers cannot find an endpoint, start with the **sidebar label** and the section where the page appears. An endpoint may be grouped under an unexpected heading, or its title may not match the wording readers expect. Compare the navigation label in the project workspace with the published sidebar. If one is less clear than the other, that is often the source of the confusion.

If the request itself is hard to understand, inspect the **Parameters** and **Request Body** sections together. Make sure required fields are clearly marked and that each parameter appears in the correct location. A path value should not be explained like a query value, and a request body example should support the field list rather than introduce unexplained fields.

If the published page looks different from what you reviewed internally, compare these elements between the project view and the published view:

- Sidebar placement
- Section anchor behavior
- Table readability
- Schema formatting
- Example block layout

When public readers misread responses, the issue is often in the **Responses** section. Expand status code descriptions where possible, and check that the example body matches the documented schema. If a **200** response example shows fields that are not described anywhere, readers may assume those fields are always present even when they are not.

[SCREENSHOT: API reference page with a confusing response section and a corrected version]

If navigation is the main problem, return to the browsing patterns described in [Managing Technical Documentation Browsing Inside Projects](#). If the page is visible but still difficult to interpret, the next step is usually to examine the entity-level details that sit behind the endpoint documentation.

Overview

API reference pages in Atloria are used in both internal project workspaces and published documentation, but readers approach them differently depending on where they are. In a project workspace, teams review endpoint pages for completeness, naming, and publish readiness. In published documentation, readers use those same pages to find request details, authentication requirements, and response examples quickly.

Across both views, the page structure stays familiar. Readers work with the **left navigation**, the **endpoint title**, the **method badge**, the **request path**, and sections such as **Parameters**, **Request Body**, **Responses**, **Authentication**, and **Examples**. Once you know where those sections are, it becomes much easier to scan a page instead of reading every line.

This guide focused on practical reading tasks:

- Finding the correct endpoint from the navigation
- Confirming the page by checking the method and path
- Reading parameters and request body fields correctly
- Matching status codes to success and error outcomes

- Reviewing pages in the project workspace before publishing
- Comparing project and published views when something looks wrong

The most useful habit is to read with a purpose. If you need access details, go to **Authentication**. If you need required inputs, check **Parameters** and **Request Body**. If you need expected results, read the main success response first and then compare error responses.

[SCREENSHOT: Complete API reference page with the main reading areas labeled]

For a closer look at how individual reference items are presented once you open them, continue with [Understanding Entity Detail Pages in Technical Documentation](#).

Prerequisites

Before using this guide, it helps to already be comfortable moving through technical documentation in Atloria. If you have not done that yet, read [Managing Technical Documentation Browsing Inside Projects](#) first. That guide explains how to move through project documentation structure so you can reach API reference sections more easily.

You should also have access to at least one of these Atloria views:

- A **project workspace** where technical documentation is available
- A **published documentation view** for a project that has already been released

This guide is most useful for people working in one of these roles:

- **Technical Writers** reviewing endpoint pages before publication
- **Documentation Managers** checking whether API pages are ready for external readers
- **Public Documentation Readers** looking up endpoint details in published docs

To follow the tasks in this guide, you should be able to recognize these page elements when they appear:

- **Sidebar** or left navigation
- **Endpoint title**
- **Method badge**
- **Request path**
- **Parameters**
- **Request Body**

- **Responses**
- **Authentication**
- **Examples**

You do not need editing instructions or setup steps to use this guide. The focus here is on reading and reviewing API reference pages as they appear on screen in Atloria, not on creating the underlying content.

[SCREENSHOT: Project documentation sidebar with an API reference section expanded]