

Reading API and Technical Reference Pages

August 25, 2026

Understanding What Appears in Published API and Technical Docs

In Atloria, published technical documentation usually separates reference content into two reader-facing page types: **API reference pages** and **technical reference pages**. You can usually tell which one you are reading by looking at the page title, the left-hand navigation group, and the labels shown near the top of the page.

An **API reference page** is built around a specific endpoint. In the sidebar, these pages are often grouped with other API operations, and the page title usually includes a request label such as **GET /resource** or **POST /resource**. Near the top of the page, you may also see details that help you identify the operation quickly, such as the request method, the endpoint path, and notes about authentication or status.

A **technical reference page** is broader. Instead of focusing on one request, it explains a technical concept, shared object, schema, or entity used across the documentation. In the sidebar, these pages usually appear under a technical or reference section rather than under a list of endpoint operations. Their titles are more likely to be object names, concept names, or reference topics.

When you open an API reference page, expect to see details such as:

Section	What you can read there	Why it matters
Endpoint header	Request method and path	Confirms the exact operation
Authentication details	Whether sign-in or credentials are needed	Helps you know access requirements
Parameters	Path, query, or header inputs	Shows what you must send
Request body	Fields and example payloads	Helps you format your request
Responses	Example results and status codes	Shows what comes back

On a technical reference page, you are more likely to see:

- Entity or object definitions

- Field descriptions
- Relationships between objects
- Rules or constraints
- Supporting notes that explain how the documented concept behaves

[SCREENSHOT: Published documentation showing an API endpoint page in the sidebar next to a technical reference page]

Navigating API Reference Pages by Endpoint

To read an API endpoint page in Atloria, start with the documentation sidebar and open the API group that matches the area you want to inspect. Inside that group, each page is usually listed by a method-and-path label, such as **GET /projects**, **POST /documents**, or **DELETE /versions/{id}**. This naming pattern helps you spot the difference between pages that work with the same resource but perform different actions.

Once you open an endpoint page, begin at the header. This top section usually gives you the clearest summary of the operation. Look for the operation title, the request path, and any visible labels that indicate whether the endpoint requires authentication or has a special status such as deprecation. If several pages have similar names, this header is the fastest way to confirm you are on the right one.

Next, move to the **Parameters** section. This is where Atloria shows the values that can be sent as part of the request. Pay attention to where each value belongs:

Parameter area	What to check	What it tells you
Path parameters	Required markers and example values	Which values must appear in the URL
Query parameters	Optional filters, defaults, and allowed values	How to narrow or sort results
Header parameters	Required headers and accepted values	Extra information needed with the request

If the page includes a **Request Body** section, compare the field list with the example payload shown on the page. Then review the **Responses** area. This section often includes response examples, schema tables, and separate blocks for different status codes. Reading these together helps you understand not only the successful result, but also what different outcomes may look like.

[SCREENSHOT: API endpoint page with endpoint header, Parameters section, Request Body, and Responses]

Reading Entity and Schema Details

API pages in Atloria often link to schema or object references. These linked pages are useful when you need to understand the structure of the data shown in a request or response. If an endpoint returns a named object, open that linked reference to inspect the full field list instead of relying only on the short example shown on the endpoint page.

Start with the object name at the top of the page. Then review the field table carefully. A good schema page usually shows each field's name, data type, whether the field is required, and a description of what the field means. Some fields also include format hints, which help you interpret values correctly.

Use this reading order when reviewing a field table:

What to look at	Why it matters
Field name	Identifies the exact property
Data type	Shows whether the value is text, number, object, array, or another format
Required status	Tells you whether the field must be present
Description	Explains the purpose of the field
Format hint	Clarifies special formats such as date-time or UUID

You may also see nullable markers, nested properties, or expandable sections for child objects. These are especially important when a field contains another object or a list of objects. Follow links to related entities when you need to understand how one object connects to another. This is often the easiest way to trace parent-child structures and reusable components shared across multiple endpoints.

Examples are just as important as field tables. A schema definition tells you the rules, while an example object shows how those rules appear in a real response. Compare the two side by side to see which fields are always present, which are optional, and how nested arrays are arranged.

[SCREENSHOT: Schema page showing object name, field table, nested properties, and example response]

Using Technical Reference Pages to Understand System Concepts

Not every important detail in Atloria belongs on an endpoint page. When you need context that goes beyond a single request, open a technical reference page. These pages are designed to explain concepts, shared entities, lifecycle rules, and other reference material that supports the API documentation.

A technical reference page is especially useful when you are trying to answer questions like:

- What does this entity represent across the product?
- How do related records connect to each other?
- Are there limits, constraints, or compatibility notes I need to know?
- Which operations work with this concept?

These pages often use structured sections rather than a simple request-and-response layout. Look for headings that break the topic into clear parts, such as field definitions, relationships, processing rules, or limitations. If the page includes diagrams, tables, or linked references, use them to build a fuller picture of how the documented concept fits into the rest of the published docs.

Cross-links are particularly helpful here. A technical reference page may link back to the API pages that create, update, list, or retrieve the entity being described. That lets you move from understanding the concept to finding the exact operation that works with it. If you are reading about a shared object, follow those links to see where it appears in actual requests and responses.

Also check for notes that apply only to specific versions or documentation contexts. Some technical reference pages include version-specific notes, compatibility details, or terminology mappings that explain why one page uses a different label than another. Those notes can save time when you are comparing pages across a project workspace and a published documentation view. For more detail on project-side browsing, continue with [Managing Technical Documentation Browsing Inside Projects](#).

Finding the Right Page Faster

When a documentation set grows, finding the right reference page quickly becomes just as important as reading it. In Atloria, the fastest results usually come from using exact terms instead of broad topic words. Search for the endpoint path, the object

name, or a field name you already know. For example, a precise path or object label is usually more effective than a general search for a product area.

If you are browsing instead of searching, use the left-hand navigation groups. This works especially well when several pages share the same resource name but differ by request method. A list that includes **GET**, **POST**, **PUT**, or **DELETE** labels makes it easier to compare related operations without opening unrelated pages.

Once you are on a page, use in-page navigation tools to jump directly to the section you need. Depending on the page layout, Atloria may show a table of contents or anchor links for major sections.

Navigation aid	Best use
Search results	Find exact paths, object names, or field names
Sidebar groups	Browse related endpoints or reference topics
In-page links	Jump to Parameters, Request Body, Responses, or Fields
Breadcrumbs	Confirm where the page sits in the documentation structure

Breadcrumbs and sidebar position are also useful for page identification. If you are unsure whether you opened an endpoint page or a broader technical reference page, check where the page sits in the sidebar and how it is labeled in the breadcrumb trail. An endpoint page usually appears inside an API group and includes a method-and-path title, while a technical reference page usually sits under a broader reference section and uses a concept or object name as the title.

[SCREENSHOT: Published docs with sidebar, breadcrumbs, search results, and in-page table of contents]

Reference: What Information to Expect on Each Page Type

Use the page layout in Atloria as a quick clue for what kind of information you should expect. API endpoint pages, schema pages, and broader technical reference pages each present information differently, even when they are closely related. If you know what normally appears on each page type, you can scan faster and avoid missing important details.

API endpoint page

An API endpoint page is centered on one operation. It usually includes:

Item	What you will usually see
Operation title	A readable name for the endpoint
Request method	Labels such as GET, POST, PUT, or DELETE
Path	The endpoint URL path
Authentication details	Whether sign-in or credentials are required
Parameter tables	Path, query, and header inputs
Request body	Field definitions and example payloads
Response details	Response schema, examples, and status codes

Entity or schema page

A schema page focuses on the shape of an object used in requests or responses. It commonly includes:

- Object name
- Field list
- Data types
- Required flags
- Nested properties
- Related objects or reusable models
- Example object values

Technical reference page

A technical reference page explains a concept or shared reference topic. It often includes:

Item	What you will usually see
Concept description	A plain-language explanation of the topic
Definitions	Fields, components, or related terms
Relationship notes	How the topic connects to other objects
Constraints	Limits, rules, or compatibility notes
Version notes	Differences tied to a release or documentation version
Related links	Links to endpoint pages or supporting references

Across all page types, Atloria may also provide navigation aids such as the sidebar, breadcrumbs, search result labels, and an in-page table of contents. These elements help you confirm what you are reading before you spend time on the details.

Overview

Published reference content in Atloria is easiest to read when you first identify the page type, then use the page structure to focus on the details that matter. API endpoint pages are best for understanding a single operation: what path it uses, which inputs it accepts, what the request body should contain, and what responses may come back. Schema pages help you inspect the exact shape of request and response objects. Technical reference pages give you the broader context behind those objects, including relationships, rules, and shared concepts.

A practical reading pattern is:

1. Confirm the page type from the title, sidebar group, and breadcrumb trail.
2. On endpoint pages, scan the header, parameters, request body, and responses.
3. Open linked schema pages when you need field-level detail.
4. Use technical reference pages when you need concept-level explanation or relationship context.
5. Use search, sidebar groups, and in-page links to move quickly between related pages.

This approach is especially helpful when you are reviewing published documentation before writing internal notes, checking release content, or comparing how a concept appears across multiple pages. In Atloria, the published documentation view is designed to support both quick scanning and deeper reference reading, so it is worth using the navigation tools instead of relying only on page-by-page browsing.

If you also work inside project workspaces, the next useful step is learning how similar technical reference content is organized before publication. Continue with [Managing Technical Documentation Browsing Inside Projects](#).

Prerequisites

You do not need any setup steps inside Atloria to use this guide effectively, but a few basics will make the pages easier to read.

Before you start, make sure you have:

- Access to a published documentation site in Atloria

- A page that includes API reference or technical reference content
- A general idea of what you are looking for, such as an endpoint path, an object name, or a field name

It also helps if you are already comfortable with the basic published documentation layout, especially the left-hand navigation and page search. If you need help getting oriented in Atloria first, review [Using Public Navigation to Browse Documentation](#) or [Browsing Published Documentation and Finding the Right Page](#).

You do **not** need to know how to create projects, connect repositories, or manage admin settings to read published reference pages. This guide is focused only on how to recognize page types, move through the published reference structure, and interpret the information shown on each page.

If your goal is to compare published reference pages with project-side technical docs, keep this guide open as a reading baseline and then continue to [Managing Technical Documentation Browsing Inside Projects](#).