

USER GUIDE

# Docusaurus Openapi Docs User Guide

Installation, Usage & Quick Reference

*A Cyber Panda Solutions LLC product*

# About This Document

## What Is This?

This is a comprehensive, source-code-verified **User Guide** for **Docusaurus Openapi Docs**, generated by DocCompiler.ai using Claude Opus 4.6 with 1M-token context. Every command, configuration option, and behavior documented here was verified against the actual source code.

## The Definitive Reference

This document is designed to be the single source of truth for Docusaurus Openapi Docs. Rather than piecing together scattered README fragments, outdated wiki pages, forum posts, and version-specific screenshots, you can rely on this as a verified, structured reference backed by automated code review.

## Optimized for AI Assistants

This document is structured for consumption by AI assistants. Feed this PDF into your preferred LLM for accurate, grounded answers about Docusaurus Openapi Docs — instead of having the model guess or hallucinate from incomplete context.

## Version & Updates

Generated on **2026-05-08** (version **3.1**). This document reflects the source code at commit `89eecd`. The always-updated version is permanently available at [doccompiler.ai/PaloAltoNetworks/docusaurus-openapi-docs](https://doccompiler.ai/PaloAltoNetworks/docusaurus-openapi-docs). Do not print or download for offline use — this document is updated when the source code changes.

## Our Mission

DocCompiler.ai aims to be the definitive wiki for the world's most important open-source projects. Every document is overkill by design: exhaustive, fact-checked, and built to replace the scattered tribal knowledge that slows teams down.

---

DocCompiler.ai is a product of Cyber Panda Solutions LLC. All documents are AI-generated from source code analysis and should be reviewed before use in production environments. Questions or corrections: [jesse.green@doccompiler.ai](mailto:jesse.green@doccompiler.ai)

# Table of Contents

## Docusaurus OpenAPI Docs User Guide

1. Quick Start
2. Installation
  - System Requirements
  - Option A: Scaffold a New Site
  - Option B: Add to an Existing Docusaurus Site
  - Post-Install Verification
3. Configuration
  - Minimal Configuration
  - Per-API Options
  - Sidebar Options
  - Theme Configuration
4. Core Usage
  - Generating API Documentation
  - Cleaning Generated Docs
  - Working with Versioned APIs
  - Using the Interactive API Explorer
  - Using Custom Mustache Templates
  - Using Custom Markdown Generators
5. Common Tasks
  - Regenerate API Docs After Spec Changes
  - Add a New API Spec
  - Use a Remote OpenAPI Spec URL
  - Enable Schema Documentation Pages
  - Group Endpoints by Tag Groups
  - Customize the Code Snippet Languages
  - Add Vendor Extension Documentation
  - Configure Authentication Persistence
6. Troubleshooting
  - Error: useTabs() must be used within a Tabs component
  - Error: Document must be valid OpenAPI 3.0.0 definition
  - Error: Directory already exists at &lt;path&gt;
  - Generated files not appearing or not updating
  - Error: Operation must have summary or operationId defined
  - Uncaught Runtime Errors in the Browser
  - Duplicate routes or wrong page when tag name equals operation ID
  - Build fails on Cloudflare Pages
  - API requests fail with CORS errors
7. FAQ

---

# Docusaurus OpenAPI Docs User Guide

## 1. Quick Start

Docusaurus OpenAPI Docs is a set of Docusaurus packages that generate interactive API reference documentation from OpenAPI (Swagger) specifications. It parses your OpenAPI 2.0, 3.0, or 3.1 spec files and produces MDX pages with a built-in API explorer, code snippet generation, schema rendering, and sidebar navigation -- all integrated into the Docusaurus documentation framework.

### Key features:

- Generate API reference pages from OpenAPI 2.0/3.0/3.1 YAML or JSON specs
- Interactive "Try It" API explorer with live request/response testing
- Auto-generated code snippets in 18+ languages (Python, cURL, C#, Go, Node.js, Ruby, PHP, Java, PowerShell, Dart, JavaScript, C, Objective-C, OCaml, R, Swift, Kotlin, Rust)
- Schema documentation with discriminator, oneOf/anyOf/allOf support
- Versioned API documentation with version selector UI
- Auto-generated sidebar navigation grouped by tag or tag group
- Custom Mustache templates for full MDX output control
- Swagger 2.0 auto-conversion to OpenAPI 3.0

**Prerequisites:** Node.js >= 18, a package manager (npm, yarn, pnpm, or bun)

### Quickest path -- scaffold a new site:

```
bash
npx create-docusaurus-openapi-docs my-api-docs
text
Creating new Docusaurus site with OpenAPI docs...
Installing dependencies with npm...
Success! Created my-api-docs
```

The scaffolded site comes with a working Petstore example. Start it immediately:

```
bash
cd my-api-docs
npm start
text
[SUCCESS] Docusaurus website is running at: http://localhost:3000/
```

### Tip

The scaffolder auto-detects your package manager from lockfiles and `npm_config_user_agent`. You can also force one with `npx create-docusaurus-openapi-docs my-site --package-manager yarn`.

**What to try next:** See [Installation](#) for adding to an existing Docusaurus site, or [Configuration](#) for customizing spec sources and sidebar behavior.

## 2. Installation

### System Requirements

Requirement	Minimum
Node.js	>= 18
Docusaurus	3.10.0+
React	16.8.4, 17, 18, or 19
Package manager	npm, yarn 1.x, pnpm, or bun

### Option A: Scaffold a New Site

bash

```
npx create-docusaurus-openapi-docs my-api-site
```

This copies a complete Docusaurus project template with [docusaurus-plugin-openapi-docs](#) and [docusaurus-theme-openapi-docs](#) pre-configured, a Petstore spec, and working sidebar/config files.

Available options:

Option	Description
<code>--package-manager &lt;pm&gt;</code>	Force npm, yarn, pnpm, or bun
<code>--skip-install</code>	Skip dependency installation

#### Important

The destination directory must either not exist or be empty (if using `.` as the name). The scaffolder will error with `Directory already exists at <path>` or `Directory not empty at <path>` otherwise.

### Option B: Add to an Existing Docusaurus Site

1. Install both packages:

bash

```
npm install docusaurus-plugin-openapi-docs docusaurus-theme-openapi-docs
```

2. Install required peer dependencies:

bash

```
npm install docusaurus-plugin-sass
```

### 3. Add the theme to your `docusaurus.config.ts`:

```
typescript
export default {
  // ...
  themes: ["docusaurus-theme-openapi-docs"],
  plugins: [
    [
      "docusaurus-plugin-openapi-docs",
      {
        id: "openapi",
        docsPluginId: "classic",
        config: {
          petstore: {
            specPath: "examples/petstore.yaml",
            outputDir: "docs/petstore",
          },
        },
      },
    ],
  ],
},
];
```

### 4. Set the doc item component in your preset's docs config:

```
typescript
presets: [
  [
    "classic",
    {
      docs: {
        docItemComponent: "@theme/ApiItem",
      },
    },
  ],
],
```

### 5. Generate the API docs:

```
bash
npx docusaurus gen-api-docs all

text
Successfully created "docs/petstore/sidebar.ts"
Successfully created "docs/petstore/add-pet.api.mdx"
Successfully created "docs/petstore/get-pet-by-id.api.mdx"
...
```

## Post-Install Verification

```
bash
npx docusaurus start
```

Navigate to your API docs path (e.g., <http://localhost:3000/petstore/>) and confirm the API explorer renders with method badges, request/response schemas, and code snippets.

### Warning

If you see `useTabs()` must be used within a `Tabs` component, ensure you are running Docusaurus 3.10.0 or later. This was a known compatibility issue (GitHub issue #1395, now resolved).

## 3. Configuration

Configuration lives in your `docusaurus.config.ts` under the `docusaurus-plugin-openapi-docs` plugin entry. Each key inside `config` defines one API spec source.

### Minimal Configuration

```
typescript
plugins: [
  [
    "docusaurus-plugin-openapi-docs",
    {
      id: "openapi",
      docsPluginId: "classic",
      config: {
        myapi: {
          specPath: "specs/openapi.yaml", // Local file or URL
          outputDir: "docs/myapi",
        },
      },
    },
  ],
],
```

### Per-API Options

Option	Type	Default	Description
<code>specPath</code>	<code>string</code>	<b>required</b>	Path to OpenAPI spec file (YAML/JSON) or URL
<code>outputDir</code>	<code>string</code>	<b>required</b>	Where generated MDX files are written
<code>proxy</code>	<code>string</code>	--	CORS proxy URL prepended to API requests
<code>template</code>	<code>string</code>	--	Custom Mustache template for API pages
<code>infoTemplate</code>	<code>string</code>	--	Custom Mustache template for info pages
<code>tagTemplate</code>	<code>string</code>	--	Custom Mustache template for tag pages
<code>schemaTemplate</code>	<code>string</code>	--	Custom Mustache template for schema pages
<code>downloadUrl</code>	<code>string</code>	--	URL or path for the "Download spec" button
<code>hideSendButton</code>	<code>boolean</code>	<code>false</code>	Hide the "Send API Request" button

<code>showExtensions</code>	<code>boolean</code>	<code>false</code>	Show vendor extensions (x-*) in docs
<code>showSchemas</code>	<code>boolean</code>	<code>false</code>	Generate schema-only pages
<code>showInfoPage</code>	<code>boolean</code>	<code>true</code>	Generate the API info page
<code>schemasOnly</code>	<code>boolean</code>	<code>false</code>	Only generate schema docs (skip endpoints)
<code>disableCompression</code>	<code>boolean</code>	<code>false</code>	Disable zlib compression of API JSON in frontmatter
<code>maskCredentials</code>	<code>boolean</code>	<code>false</code>	Mask auth credential inputs
<code>externalJsonProps</code>	<code>boolean</code>	<code>true</code>	Write large JSON props to external files for performance

## Sidebar Options

Nest these under `sidebarOptions`:

typescript

```

sidebarOptions: {
  groupPathsBy: "tag", // "tag" or "tagGroup"
  categoryLinkSource: "tag", // "tag", "info", or "auto"
  customProps: {}, // Custom properties for sidebar items
  sidebarCollapsible: true, // Enable collapsible sidebar categories
  sidebarCollapsed: true, // Start sidebar categories collapsed
},

```

Option	Values	Description
<code>groupPathsBy</code>	<code>"tag", "tagGroup"</code>	How endpoints are grouped in the sidebar
<code>categoryLinkSource</code>	<code>"tag", "info", "auto"</code>	What links to from the category header
<code>sidebarCollapsible</code>	<code>boolean</code>	Whether sidebar categories can collapse
<code>sidebarCollapsed</code>	<code>boolean</code>	Whether categories start collapsed

### Note

When using `groupPathsBy: "tagGroup"`, your OpenAPI spec must include the `x-tagGroups` vendor extension to define the grouping structure.

## Theme Configuration

In `themeConfig`, you can customize the API explorer behavior:

```
typescript

themeConfig: {
  api: {
    proxy: "https://cors.pan.dev",
    authPersistence: "localStorage", // false | "sessionStorage" | "localStorage"
    requestTimeout: 30000, // milliseconds
    requestCredentials: "same-origin", // "omit" | "same-origin" | "include"
  },
  languageTabs: [
    { highlight: "python", language: "python", logoClass: "python" },
    { highlight: "bash", language: "curl", logoClass: "curl" },
    // ... add or remove language tabs
  ],
},
```

### Tip

Set `authPersistence: "localStorage"` to keep API keys across browser sessions. Use `false` for security-sensitive environments where credentials should never persist.

For the full configuration reference, see the Admin Guide.

## 4. Core Usage

### Generating API Documentation

The plugin registers four Docusaurus CLI commands:

#### Generate docs for a single API:

```
bash

npx docusaurus gen-api-docs <id>
```

Where `<id>` matches a key in your plugin `config`. Example:

```
bash

npx docusaurus gen-api-docs petstore

text

Successfully created "docs/petstore/sidebar.ts"
Successfully created "docs/petstore/swagger-petstore-yaml.info.mdx"
Successfully created "docs/petstore/pet.tag.mdx"
Successfully created "docs/petstore/add-pet.api.mdx"
...
```

#### Generate docs for all configured APIs:

```
bash

npx docusaurus gen-api-docs all
```

#### Generate only schema docs:

```
bash

npx docusaurus gen-api-docs all --schemas-only
```

### Generate with a specific plugin instance:

```
bash  
npx docusaurus gen-api-docs myapi -p openapi
```

Flag	Description
<code>-p, --plugin-id &lt;plugin&gt;</code>	Target a specific plugin instance (required when multiple exist)
<code>--all-versions</code>	Generate docs for all configured versions
<code>--schemas-only</code>	Only generate schema documentation pages

#### Warning

The `gen-api-docs` command will **not overwrite** existing MDX files. To regenerate, clean first with `clean-api-docs`, then generate again.

## Cleaning Generated Docs

### Clean a single API:

```
bash  
npx docusaurus clean-api-docs <id>
```

### Clean all APIs:

```
bash  
npx docusaurus clean-api-docs all
```

### Clean all including versions:

```
bash  
npx docusaurus clean-api-docs all --all-versions
```

### Clean only schemas:

```
bash  
npx docusaurus clean-api-docs all --schemas-only
```

This removes all generated `.api.mdx`, `.info.mdx`, `.tag.mdx` files, the `sidebar.ts`, externalized `.json` files, and the `schemas/` directory.

## Working with Versioned APIs

To maintain multiple versions of the same API:

```
typescript  
config: {  
  petstore: {  
    specPath: "specs/petstore-v2.yaml",  
    outputDir: "docs/petstore",  
    version: "2.0.0", // Required when versions is defined
```

```
label: "v2.0.0", // Required when versions is defined
baseUrl: "/petstore/swagger-petstore-yaml", // Required when versions is defined
versions: {
  "1.0.0": {
    specPath: "specs/petstore-v1.yaml",
    outputDir: "docs/petstore/1.0.0",
    label: "v1.0.0",
    baseUrl: "/petstore/1.0.0/swagger-petstore-yaml",
  },
},
},
},
},
```

### Generate version docs:

```
bash
npx docusaurus gen-api-docs:version petstore:all
```

### Or a specific version:

```
bash
npx docusaurus gen-api-docs:version petstore:1.0.0
```

### Clean versioned docs:

```
bash
npx docusaurus clean-api-docs:version petstore:all
```

The version system generates a `versions.json` file and the sidebar utilities (`versionSelector`, `versionCrumb`) render a dropdown and breadcrumb in the sidebar:

```
typescript
import petstoreVersions from "./docs/petstore/versions.json";
import { versionSelector, versionCrumb } from "docusaurus-plugin-openapi-docs/lib/sidebars/utils";

const sidebars = {
  "petstore-2.0.0": [
    { type: "html", defaultStyle: true, value: versionSelector(petstoreVersions) },
    { type: "html", defaultStyle: true, value: versionCrumb("v2.0.0") },
    // ... sidebar items
  ],
};
```

## Using the Interactive API Explorer

Each generated API page includes an interactive explorer panel:

- **Method + Endpoint badge** -- color-coded HTTP method with the full path
- **Authorization** -- fill in API keys, Bearer tokens, OAuth2, or OpenID Connect credentials
- **Parameters** -- form inputs for path, query, header, and cookie parameters with type validation
- **Request Body** -- JSON editor with schema validation, file upload for multipart, and XML support
- **Send Request** -- execute the request against the configured server (or proxy)
- **Response** -- view status code, headers, and response body with syntax highlighting
- **Code Snippets** -- auto-generated examples in all configured languages

### Note

For cross-origin API testing, set the `proxy` option to a CORS proxy (e.g., `"https://cors.pan.dev"`). Without a proxy, browser CORS policies will block requests to external APIs.

## Using Custom Mustache Templates

Override the default MDX output by providing custom Mustache templates:

```
typescript
config: {
  myapi: {
    specPath: "specs/openapi.yaml",
    outputDir: "docs/myapi",
    template: "templates/api.mustache",
    infoTemplate: "templates/info.mustache",
    tagTemplate: "templates/tag.mustache",
    schemaTemplate: "templates/schema.mustache",
  },
},
```

Available template variables include `{{{id}}}`, `{{{title}}}`, `{{{frontMatter.description}}}`, `{{{markdown}}}`, `{{{json}}}`, `{{{api.method}}}`, and `{{{infoPath}}}`.

## Using Custom Markdown Generators

For programmatic control over the generated markdown, use `markdownGenerators`:

```
typescript
import { myCustomApiMdGenerator } from "./customMdGenerators";

config: {
  myapi: {
    specPath: "specs/openapi.yaml",
    outputDir: "docs/myapi",
    markdownGenerators: {
      createApiPageMD: myCustomApiMdGenerator,
    },
  },
},
```

You can override four generators: `createApiPageMD`, `createInfoPageMD`, `createTagPageMD`, and `createSchemaPageMD`. Each receives the page metadata object and must return a markdown string.

### Tip

The demo site includes a working custom markdown generator at `demo/customMdGenerators.ts` that adds a server URL table to each API page. Use it as a starting point for your own customizations.

## 5. Common Tasks

## Regenerate API Docs After Spec Changes

1. Clean the existing generated files:

```
bash
npx docusaurus clean-api-docs all
```

2. Regenerate:

```
bash
npx docusaurus gen-api-docs all
```

3. Verify locally:

```
bash
npx docusaurus start
```

## Add a New API Spec

1. Place the spec file in your project (e.g., [specs/payments.yaml](#))
2. Add a new entry to the plugin config:

```
typescript
config: {
  // existing APIs...
  payments: {
    specPath: "specs/payments.yaml",
    outputDir: "docs/payments",
    sidebarOptions: {
      groupPathsBy: "tag",
      categoryLinkSource: "tag",
    },
  },
},
```

3. Generate docs:

```
bash
npx docusaurus gen-api-docs payments
```

4. Import the generated sidebar in your [sidebars.ts](#):

```
typescript
import paymentsSidebar from "../docs/payments/sidebar";
```

## Use a Remote OpenAPI Spec URL

```
typescript
config: {
  myapi: {
    specPath: "https://api.example.com/openapi.json",
    outputDir: "docs/myapi",
  },
},
```

The plugin fetches and resolves the spec at generation time, including remote [\\$ref](#) pointers.

## Enable Schema Documentation Pages

```
typescript
config: {
  myapi: {
    specPath: "specs/openapi.yaml",
    outputDir: "docs/myapi",
    showSchemas: true,
  },
},
```

Schema pages are generated under `docs/myapi/schemas/` as `{schema-name}.schema.mdx`.

## Group Endpoints by Tag Groups

Use the `x-tagGroups` vendor extension in your OpenAPI spec, then configure:

```
typescript
sidebarOptions: {
  groupPathsBy: "tagGroup",
},
```

## Customize the Code Snippet Languages

In `themeConfig.languageTabs`, add or remove language entries:

```
typescript
languageTabs: [
  { highlight: "python", language: "python", logoClass: "python" },
  { highlight: "bash", language: "curl", logoClass: "curl" },
  { highlight: "go", language: "go", logoClass: "go" },
],
```

Each entry needs `highlight` (Prism language), `language` (Postman code-gen ID), and `logoClass` (CSS class for the language icon). Optionally add `variant` (e.g., `"unirest"` for Java).

## Add Vendor Extension Documentation

1. Add `x-*` fields to your OpenAPI spec operations
2. Enable in the plugin config:

```
typescript
config: {
  myapi: {
    specPath: "specs/openapi.yaml",
    outputDir: "docs/myapi",
    showExtensions: true,
  },
},
```

This generates a `show_extensions: true` frontmatter field that renders the `x-codeSamples`, `x-deprecated-description`, and other extensions inline.

## Configure Authentication Persistence

```
typescript  
  
themeConfig: {  
  api: {  
    authPersistence: "localStorage", // Persist across sessions  
  },  
},
```

Options: `false` (in-memory only), `"sessionStorage"` (default, per-tab), `"localStorage"` (persistent).

### Warning

Using `"localStorage"` stores API credentials in the browser's local storage. Avoid this on public or shared machines.

## 6. Troubleshooting

**Error:** `useTabs()` must be used within a `Tabs` component

**Cause:** Incompatibility with Docusaurus versions below 3.10.0 (GitHub issue #1395).

**Solution:**

1. Upgrade Docusaurus to 3.10.0 or later:

```
bash
```

```
npm install @docusaurus/core@3.10.0 @docusaurus/preset-classic@3.10.0
```

2. Rebuild:

```
bash
```

```
npx docusaurus clean && npx docusaurus build
```

**Prevention:** Always match the Docusaurus version requirement (3.10.0+) before installing the OpenAPI packages.

**Error:** `Document must be valid OpenAPI 3.0.0 definition`

**Cause:** The spec file is not a valid OpenAPI document, or its structure cannot be parsed.

**Solution:**

1. Validate your spec with a linter (e.g., Redocly CLI or Swagger Editor)
2. Ensure the `openapi` field is present and set to `"3.0.x"` or `"3.1.x"`
3. For Swagger 2.0 specs, the plugin auto-converts -- but the input must still be a valid Swagger document

**Prevention:** Lint your spec before running `gen-api-docs`.

**Error:** `Directory already exists at <path>`

**Cause:** The `create-docusaurus-openapi-docs` scaffolder found an existing directory at the destination.

**Solution:** Choose a different name or use `.` to scaffold into the current (empty) directory.

## Generated files not appearing or not updating

**Cause:** The `gen-api-docs` command does not overwrite existing files.

**Solution:**

```
bash
npx docusaurus clean-api-docs all
npx docusaurus gen-api-docs all
```

**Prevention:** Always clean before regenerating.

**Error:** `Operation must have summary or operationId defined`

**Cause:** An operation in your OpenAPI spec is missing both `summary` and `operationId`.

**Solution:** Add at least one of `summary` or `operationId` to every operation in your spec. The plugin uses these to generate the MDX filename.

## Uncaught Runtime Errors in the Browser

**Cause:** Various issues reported in GitHub issue #843 -- typically related to malformed schemas, circular references, or spec edge cases.

**Solution:**

1. Simplify circular `$ref` references (the plugin handles them but deep nesting can cause issues)
2. Clear the Docusaurus cache: `npx docusaurus clear`
3. Regenerate: `npx docusaurus clean-api-docs all && npx docusaurus gen-api-docs all`

## Duplicate routes or wrong page when tag name equals operation ID

**Cause:** The slug generation creates collisions when a tag name and an `operationId` produce the same slug (GitHub issue #1110).

**Solution:** Ensure operation IDs are unique and do not collide with tag names. Consider prefixing operation IDs.

## Build fails on Cloudflare Pages

**Cause:** The `sass` dependency requires native compilation which may not be available in all CI environments (GitHub issue #1048).

**Solution:** Ensure your CI environment has Node.js  $\geq 18$  with native build tools. Consider pre-building locally and deploying the static output.

### Note

If MDX compilation fails with errors about Swagger-specific syntax (GitHub issue #802), ensure your spec is valid OpenAPI 3.x. The MDX compiler can choke on malformed description fields containing unescaped JSX-like syntax.

## API requests fail with CORS errors

**Cause:** The API server doesn't include CORS headers for your documentation domain.

**Solution:** Configure a CORS proxy:

```
typescript
// Per-API proxy
config: {
  myapi: {
    proxy: "https://cors.pan.dev",
  },
},

// Global proxy in themeConfig
themeConfig: {
  api: {
    proxy: "https://cors.pan.dev",
  },
},
```

### Caution

CORS proxies relay your API requests (including credentials) through a third-party server. Use only for testing and demo purposes -- never route production traffic through a public CORS proxy.

## 7. FAQ

### Q: Which OpenAPI specification versions are supported?

OpenAPI 2.0 (Swagger), 3.0.x, and 3.1.x. Swagger 2.0 specs are automatically converted to OpenAPI 3.0 during processing using the [swagger2openapi](#) library.

### Q: Can I use a remote URL for my OpenAPI spec?

Yes. Set `specPath` to a full URL (e.g., "<https://api.example.com/openapi.json>"). The plugin detects URLs by checking for `https://` or `http://` prefixes and fetches them at generation time.

### Q: How do I add multiple APIs to the same site?

Add multiple keys under `config` in the plugin options. Each key becomes an `<id>` you can target with `gen-api-docs <id>`. Use `gen-api-docs all` to generate all at once.

### Q: Can I use multiple plugin instances?

Yes. Add the plugin multiple times with different `id` values. When more than one instance exists, you must pass `--plugin-id <id>` (or `-p <id>`) to the CLI commands to specify which instance to target.

### Q: What is the `externalJsonProps` option?

When enabled (default: `true`), large JSON properties in generated MDX are written to separate `.json` files and loaded via `require()`. This significantly improves MDX compilation performance for large OpenAPI specs.

### Q: How do I customize the generated MDX output?

Two approaches: (1) Mustache templates via `template`, `infoTemplate`, `tagTemplate`, and `schemaTemplate` options for full structural control, or (2) `markdownGenerators` for programmatic markdown generation. See the Core Usage section for details.

### Q: Does the API explorer actually send requests?

Yes. The "Send" button in the API explorer makes real HTTP requests to the configured server (from the spec's `servers` field or the `proxy` setting). Requests are made client-side from the user's browser.

### Q: How do I add custom code samples?

Use the `x-codeSamples` vendor extension on your OpenAPI operations:

yaml

```
x-codeSamples:  
  - lang: Python  
    source: |  
      import requests  
      response = requests.get("https://api.example.com/pets")
```

Enable with `showExtensions: true` in the plugin config.

### Q: Can I generate only schema documentation without endpoint pages?

Yes. Set `schemasOnly: true` in the API config, or pass `--schemas-only` to the CLI commands.

### Q: What happens with circular `$ref` references?

The plugin resolves circular references using a custom serializer that replaces circular nodes with `circular()` or `circular(Title)` markers. The resulting docs will show the schema up to the point of circularity.

### Q: How do I add i18n support?

i18n for theme components is a requested feature (GitHub issues #319, #1146) but not yet fully implemented. The theme includes `translationIds.ts` for future translation support.

*DocCompiler.ai v3.1 | Upgraded source coverage | Generated 2026-05-08 | Source: PaloAltoNetworks/docusaurus-openapi-docs @ 89ecbd A Cyber Panda Solutions LLC product. This document is intentionally comprehensive -- built to be the complete, source-verified reference an AI or human needs without consulting other sources. **Do not print:** the always-current version lives at [doccompiler.ai/PaloAltoNetworks/docusaurus-openapi-docs](https://doccompiler.ai/PaloAltoNetworks/docusaurus-openapi-docs).*