





□□□□ >

Doom  [rspress](#)  Alauda  [rspress](#)  Alauda  [rspress](#) Alauda [rspress](#)

□ □ □ □

- □□□□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□

[illegible]



0000000000 MDX 000 ↗

0000

000000000000 Doom 00

00

00

0000

000000

00

11

□□ `doom` □□□□

□ □ □ □

□ □ □ □

API □□□□

□ □ □ □ □ □ □ □

Page 10 of 10

□ □ □ □ □ □

□ □ □ □ □

Page 10 of 10

□ □ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □

sites.yaml □□

□ □ □ □

□ □ □ □ □ □ □ □ □

Page 10 of 10

Algolia ☐ ☐ ☐ ☐

Sitemap ☐ ☐

11

□ □ □ □

333

11

11

Markdown

Callouts

Mermaid

MDX

📄 MDX 📄📄📄📄📄📄📄📄📄📄

rspress 📄

doom 📄

📄📄📄📄📄

📄📄📄

📄📄📄📄📄📄📄📄📄📄

i18n.json

.ts/.tsx

.mdx

API 📄

📄 API

CRD

📄📄📄

📄 openapi 📄

📄📄📄📄📄

props

📄

📄📄📄

📄📄📄📄

00

00000000000000000000 ACP 00

000000

000000

0000000

000000000

📄📄

📄📄

📄📄📄

📄📄📄📄

📄📄📄📄📄

📄📄📄📄📄

📄📄📄

📄📄📄📄📄📄

📄📄📄

📄📄 PDF

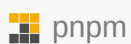
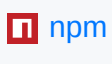
📄📄📄

📄📄📄📄

📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

```
mkdir my-docs && cd my-docs
```

📄 `npm init -y` 📄📄📄📄📄📄📄📄📄📄📄 `npm`📄`yarn` 📄 `pnpm` 📄📄 `doom:`



```
npm install -D @alauda/doom typescript
```

📄📄📄📄📄📄📄📄:

```
# 创建 docs 目录
mkdir docs/en && echo '# Hello World' > docs/en/index.md
mkdir docs/zh && echo '# 文档' > docs/zh/index.md
```

在 `package.json` 中添加脚本:

```
{
  "scripts": {
    "dev": "doom dev",
    "build": "doom build",
    "new": "doom new",
    "serve": "doom serve",
    "translate": "doom translate",
    "export": "doom export"
  }
}
```

在 `doom.config.yml` 中添加:

```
title: My Docs
```

在 `tsconfig.json` 中添加:

```
{
  "compilerOptions": {
    "jsx": "react-jsx",
    "module": "NodeNext",
    "moduleResolution": "NodeNext",
    "noUnusedLocals": true,
    "noUnusedParameters": true,
    "resolveJsonModule": true,
    "skipLibCheck": true,
    "strict": true,
    "target": "ESNext",
  },
  "mdx": {
    "checkMdx": true,
  },
}
```

📄📄📄 global.d.ts 📄📄📄📄📄📄

```
/// <reference types="@alauda/doom/runtime" />
```

📄📄📄📄📄 .mdx 📄📄📄📄📄📄📄📄 doom 📄📄📄📄📄📄

📄📄📄📄📄


```
doom -h
```

```
# output
```

```
Usage: doom [options] [command]
```

```
Doctor Doom making docs.
```

Options:

-V, --version	output the version number
-c, --config <config>	Specify the path to the config file
-v <version>	Specify the version of the documentation, c
-b, --base <base>	Override the base of the documentation
-p, --prefix <prefix>	Specify the prefix of the documentation bas
-f, --force [boolean]	Force to
	1. fetch latest reference remotes or scaffo
	2. translate ignore hash equality check and
-i, --ignore [boolean]	Ignore internal routes (default: false)
-d, --download [boolean]	Display download pdf link on nav bar (defau
-e, --export [boolean]	Run or build in exporting PDF mode, `apis/*
-I, --include <language...>	Include only the specific language(s),
-E, --exclude <language...>	Include all languages except the specific l
-o, --out-dir <path>	Override the `outDir` defined in the config
-r, --redirect <enum>	Whether to redirect to the locale closest t
-R, --edit-repo [boolean url]	Whether to enable or override the `editRepo
-a, --algolia [boolean alauda]	Whether to enable or use the alauda (docs.a
-S, --site-url	Whether to enable the siteUrl for sitemap g
-n, --no-open	Do not open the browser after starting the
-h, --help	display help for command

Commands:

dev [options] [root]	Start the development server
build [root]	Build the documentation
preview serve [options] [root]	Preview the built documentation
new [template]	Generate scaffolding from templates
translate [options] [root]	Translate the documentation
export [options] [root]	Export the documentation as PDF, `apis/**`
lint [root]	Lint the documentation
help [command]	display help for command

📄📄📄📄📄

📄 `yarn dev` 📄📄📄📄📄📄📄📄📄📄📄📄📄📄

```
doom dev -h
```

```
# output
```

```
Usage: doom dev [options] [root]
```

```
Start the development server
```

```
Arguments:
```

```
  root                                Root directory of the documentation
```

```
Options:
```

```
  -H, --host [host]                  Dev server host name
  -P, --port [port]                  Dev server port number
  -l, --lazy [boolean]                Whether to enable `lazyCompilation` which could i
  --no-lazy                           Do not enable `lazyCompilation`
  -h, --help                          display help for command
```

📄📄📄📄📄

📄 `yarn build` 📄📄📄📄📄📄📄📄📄 `dist` 📄📄📄📄📄

📄📄📄

📄 `yarn serve` 📄📄📄📄📄📄📄📄📄📄 `-b`, `-p` 📄📄📄📄📄📄📄📄 `-b`, `-p` 📄
📄

📄📄📄📄📄

📄 `yarn new` 📄📄📄📄📄📄📄📄📄📄

📄📄📄

```
doom translate -h
```

```
# output
```

```
Usage: doom translate [options] [root]
```

```
Translate the documentation
```

```
Arguments:
```

```
  root                                Root directory of the documentation
```

```
Options:
```

```
-s, --source <language> Document source language, one of en, zh, ru (default)
-t, --target <language> Document target language, one of en, zh, ru (default)
-g, --glob <path...>      Glob patterns for source dirs/files
-C, --copy [boolean]       Whether to copy relative assets to the target directory
-h, --help                 display help for command
```

- `-g, --glob` 指定源目录或文件的路径 `glob` 指定要匹配的 glob 模式

```
1. yarn translate -g abc xyz 将 <root>/<source>/abc 文件复制到 <root>/<source>/xyz
   目录 <root>/<target>/abc 目录 <root>/<target>/xyz 目录
```

```
2. yarn translate -g '*' 将 <root>/<source> 目录下的所有文件复制到
```

- `-C, --copy` 是否将源目录中的相对资产复制到目标目录 `false` 表示不复制

- 示例

```
1. /<source>/abc.jpg 复制到 <root>/public/<source>/abc.jpg 目录
   <root>/public/<target>/abc.jpg 目录 <root>/<target>/abc.jpg 目录
```

```
2. <root>/<source>/abc.mdx 复制到 ./assets/xyz.jpg 目录
   <root>/<source>/assets/xyz.jpg 目录 <root>/<target>/assets/xyz.jpg 目录
   复制到
```

```
3. <root>/<source>/abc.mdx 复制到 ./assets/<source>/xyz.jpg 目录
   <root>/<source>/assets/<source>/xyz.jpg 目录
   <root>/<target>/assets/<target>/xyz.jpg 目录
   ./assets/<target>/xyz.jpg
```

- `process.argv[2]`
 1. `/<source>/abc.jpg` `process.argv[2]` `<root>/public/<target>/abc.jpg` `process.argv[2]`
`process.argv[2]` `/<target>/abc.jpg` `process.argv[2]`
 2. `<root>/<source>/abc.mdx` `process.argv[2]` `./assets/<source>/xyz.jpg` `process.argv[2]`
`<root>/<target>/assets/<target>/xyz.jpg` `process.argv[2]`
`./assets/<target>/xyz.jpg` `process.argv[2]`
`../<source>/assets/<target>/xyz.jpg`

WARNING

`process.argv[2]` `-g '*'` `process.argv[2]` `source` `process.argv[2]` `target` `process.argv[2]`
`internalRoutes` `process.argv[2]` `target` `process.argv[2]`

TIP

`process.argv[2]` `AZURE_OPENAI_API_KEY` `process.argv[2]` `Leader` `process.argv[2]`

`process.argv[2]`

```
i18n:
  title:
    en: DevOps Connectors
  additionalPrompts: '🔒 Connectors 🔒'
  disableAutoTranslation: false
title: DevOps 🔒
```

`process.argv[2]`

PDF

WARNING

`process.argv[2]` `yarn build` `process.argv[2]`

```
doom export -h
```

```
# output
```

```
Usage: doom export [options] [root]
```

```
Export the documentation as PDF, `apis/**` and `*/apis/**` routes will be ign
```

```
Arguments:
```

```
  root                Root directory of the documentation
```

```
Options:
```

```
  -H, --host [host]  Serve host name
```

```
  -P, --port [port]  Serve port number (default: "4173")
```

```
  -h, --help          display help for command
```

```
❯ yarn export 📄📄📄 PDF 📄📄📄📄📄📄📄 -b , -p 📄📄📄📄📄📄📄📄 -b , -p
❯
```

```
📄📄📄📄 playwright ↗📄📄📄📄📄 build-harbor.alauda.cn/frontend/playwright-
runner:doom 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄
```

```
.env.yarn
```

```
PLAYWRIGHT_DOWNLOAD_HOST="https://cdn.npmmirror.com/binaries/playwright"
```

```
📄📄📄📄
```

```
doom lint -h
```

```
# output
```

```
Usage: doom lint [options] [root]
```

```
Lint the documentation
```

```
Arguments:
```

```
  root          Root directory of the documentation
```

```
Options:
```

```
  -h, --help    display help for command
```

```
████████████████████
```



📄

📄 `doom` 📄📄

📄📄

📄📄

API 📄📄

📄📄📄📄📄

📄📄📄📄

📄📄📄📄

📄📄📄

📄📄📄📄📄

📄📄📄📄📄📄

📄📄📄📄📄

`sites.yaml` 📄

📄📄

📄📄📄📄📄📄

📄📄📄📄

Algolia 📄📄

Sitemap 📄

API 00

00 API

CRD

0000

00 openapi 00

000000

props

00

0000

000000

00

00000000000000000000 ACP 00

00000

00000

000000

00000000

11

Sitemap ☐ ☐

--	--	--	--

```
yaml doom.config.yaml  
doom.config.yml rspress js/ts
```

支持 .js/.ts/.mjs/.mts/.cjs/.cts 文件

在 js/ts 文件中引入 @alauda/doom/config 中的 defineConfig 函数

```
import { defineConfig } from '@alauda/doom/config'

export default defineConfig({})
```

配置

- lang** 语言，支持 en 和 null 或 undefined
- title** 标题
- logo** logo 的公共路径，alauda logo 的公共路径是 doom
- logoText** logo 的文本
- icon** favicon 的 logo
- base** 产品文档的公共路径，product-docs /
- outDir** 输出目录，dist/{base}/{version} 或 dist/{outDir}/{version}，version 是版本号

API

api:

```
# CRD 配置 doom.config.* 文件 glob 模式 json/yaml 文件
```

crds:

```
- docs/shared/crds/*.yaml
```

```
# OpenAPI 配置 doom.config.* 文件 glob 模式 json/yaml 文件
```

openapis:

```
- docs/shared/openapis/*.json
```

```
# 是否 openapi 文件 默认 true
```

```
# 是否 https://doom.alauda.cn/apis/references/CodeQuality.html#v1alpha1.CodeQ
```

references:

```
v1alpha1.CodeQualityBranch: /apis/references/CodeQualityBranch#v1alpha1.C
```

```
# 是否 API 文件 gateway 文件
```

```
pathPrefix: /apis
```

API 文档

API 文档

```
# 配置 doom.config.* 文件 glob 模式 json/yaml 文件
```

permission:**functionresources:**

```
# `kubectl get functionresources`
```

```
- docs/shared/functionresources/*.yaml
```

roletemplates:

```
# `kubectl get roletemplates -l auth.cpaas.io/roletemplate.official=true`
```

```
- docs/shared/roletemplates/*.yaml
```

API 文档

API 文档

release-notes.mdx

```
{/* release-notes-for-bugs?template=fixed&project=DevOps */}
```

```

template=fixed&project=DevOps  fixed  queryTemplates 
query  project=DevOps  ejc  fixed  jira  jql 
https://jira.alauda.cn/rest/api/2/search?jql=<jql>  API 
JIRA_USERNAME  JIRA_PASSWORD 

```

--	--	--	--	--

```
sidebar:
  collapsed: false # oooooooooooooooooooooooooooooooooooooo false
```

--	--	--	--	--	--	--	--

```
internalRoutes: # glob docs cli -i, --ignore`
- '*/internal/**'
```

□ □ □ □ □ □ □ □ □

```
onlyIncludeRoutes: # 匹配 glob 匹配 docs 匹配 cli 匹配 -i, --ignore` 匹配
  - '*/internal/**'

internalRoutes:
  - '*/internal/overview.mdx'
```

```
theme: # optional, https://shiki.style/themes
langs: # optional, https://shiki.style/languages
transformers: # optional, only available in js/ts config, https://shiki.sty
```

plaintext

□ □ □ □

translate:

```
# =====ejs ===== `sourceLang`, `targetLang`, `userPrompt`, `additional
#  `sourceLang`  `targetLang`  ` `  ` `  =====
#    `userPrompt` =====
#    `additionalPrompts`  `frontmatter.i18n`  `additionalPrompts`  ==
#    `terms`  `titleTranslationPrompt` ===== AI ==
# =====
```

systemPrompt: |

You are a professional technical documentation engineer, skilled in writing h

Baseline Requirements

- Sentences should be fluent and conform to the expression habits of the <%=
- Input format is MDX; output format must also retain the original MDX format
- ****CRITICAL****: Do not translate or modify ANY link content in the document.
 - URLs in markdown links: [text](URL) - keep URL exactly as is
 - Reference-style links: [text][ref] and [ref]: URL - keep both ref and URL
 - Inline URLs: https://example.com - keep completely unchanged
 - Image links: ![alt](src) - keep src unchanged, but alt text can be transl
 - Anchor links: [text](#anchor) - keep #anchor unchanged
 - Any href attributes in HTML tags - keep unchanged
- Do not translate professional technical terms and proper nouns, including b
- The title field and description field in frontmatter should be translated,
- Content within MDX components needs to be translated, whereas MDX component
- Do not modify or translate any placeholders in the format of __ANCHOR_N__ (
- Keep original escape characters like backslash, angle brackets, etc. unchan
- Do not add any escape characters to special characters like [], (), {}, etc
 - If source has "Architecture [Optional]", keep it as "Architecture [Option
 - If source has "Function (param)", keep it as "Function (param)" (not "Fun
 - Only add escape characters if they were present in the original text
- Preserve and do not translate the following comments, nor modify their cont
 - {/* release-notes-for-bugs */}
 - <!-- release-notes-for-bugs -->
- Remove and do not retain the following comments:
 - {/* reference-start */}
 - {/* reference-end */}
 - <!-- reference-start -->
 - <!-- reference-end -->
- Ensure the original Markdown format remains intact during translation, such
- Do not translate the content of the code block.

<% if (titleTranslationPrompt) { %>

<%- titleTranslationPrompt %>

<% } %>

<% if (terms) { %>


```
<%- terms %>
<% } %>

<% if (userPrompt || additionalPrompts) { %>
## Additional Requirements
These are additional requirements for the translation. They should be met also

The text for translation is provided below, within triple quotes:
"""
<% if (userPrompt) { %>
<%- userPrompt %>
<% } %>

<% if (additionalPrompts) { %>
<%- additionalPrompts %>
<% } %>
"""
<% } %>
```

📄📄📄📄📄📄📄📄

editRepoBaseUrl: [alauda/doom/tree/main/docs](https://github.com/lauda/doom/tree/main/docs) # [https://github.com/](https://github.com/lauda/doom/tree/main/docs) 📄📄📄📄📄📄📄📄

📄📄📄📄📄📄

lint:
cspellOptions: # 📄📄📄cspell 📄📄📄📄 <https://github.com/streetsidesoftware/cspell>

Algolia 📄📄📄📄

```
algolia: # 📄Algolia 📄📄📄📄📄📄📄 ` -a, --algolia` 📄📄📄📄📄
appId: # Algolia 📄 ID
apiKey: # Algolia API Key
indexName: # Algolia 📄📄📄
```

📄📄 `public/robots.txt` 📄 Algolia 📄📄📄

INFO

📄 `rspress` 📄📄📄📄📄 Algolia 📄📄📄📄📄 [📄📄📄📄📄](#) 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄
`@alauda/doom/theme` 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

```
export * from '@alauda/doom/theme'
```

Sitemap 📄📄

```
siteUrl: https://docs.alauda.cn # 📄📄📄📄 URL📄📄📄📄 sitemap📄📄📄📄 ` -S, --site-url
```

11

--	--	--	--

index

index.md index.mdx

```
├─ index.md
├─ start.mdx
└─ usage
    ├─ index.mdx
    └─ convention.md
```

--	--	--	--	--	--

1. `public` □□□□□□□□□□□□□□□□
2. `public/_remotes` □□□□□□□□□□□□□□□□□□□□□□□□□□□□
`*/public/_remotes` □□ `.gitignore` □□□□□□□□□□
3. `shared` □□□□□□□□□□□□□□□□□□□□□□□□□□□□

🐱🐱

🐱🐱🐱🐱🐱🐱 `frontmatter` 🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱

```
---
title: 🐱
description: 🐱
author: 🐱
category: 🐱
---
```

🐱🐱🐱🐱🐱🐱 `MDX` 🐱 `.mdx` 🐱🐱🐱🐱🐱 `frontmatter` 🐱🐱🐱🐱🐱🐱

🐱🐱

🐱 `index.md` 🐱 `index.mdx` 🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱 `frontmatter` 🐱
`weight` 🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱 `weight` 🐱🐱🐱🐱🐱🐱

```
---
weight: 1
---
```

WARNING

🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱🐱

🐱🐱

index.mdx Overview

```
#  
  
<Overview />
```

gfm ↗ Doom Markdown

Callouts

Mermaid

□ □ □ □ □ □

1. 在代码块中，使用 `;`, `%`, `#`, `//`, `/** */`, `--` 和 `<!-- -->` 时，需要转义。
2. 在代码块中，使用 `[\\!code callout]` 时，需要转义。
3. 在代码块中，使用 `:::callouts` 时，需要转义。在代码块中，使用 `<div class="doom-callouts">` 时，需要转义。在代码块中，使用 `<Callouts>` 时，需要转义。

```
```sh
```

```
Memory overhead per virtual machine ≈ (1.002 × requested memory) \
 + 218 MiB \
 + 8 MiB × (number of vCPUs) \
 + 16 MiB × (number of graphics devices) \
 + (additional memory overhead)
```

```
```
```

```
:::callouts
```

1. Required for the processes that run in the `virt-launcher` pod.
2. Number of virtual CPUs requested by the virtual machine.
3. Number of virtual graphics cards requested by the virtual machine.
4. Additional memory overhead:
 - If your environment includes a Single Root I/O Virtualization (SR-IOV) network device or a Graphics Processing Unit (GPU), allocate 1 GiB additional memory overhead for each device.
 - If Secure Encrypted Virtualization (SEV) is enabled, add 256 MiB.
 - If Trusted Platform Module (TPM) is enabled, add 53 MiB.

```
:::
```

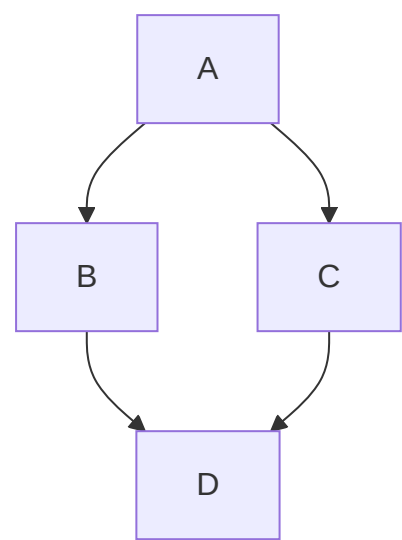
```
Memory overhead per virtual machine ≈ (1.002 × requested memory) \
    + 218 MiB \
    + 8 MiB × (number of vCPUs) \
    + 16 MiB × (number of graphics devices) \
    + (additional memory overhead)
```

- ① Required for the processes that run in the `virt-launcher` pod.
- ② Number of virtual CPUs requested by the virtual machine.
- ③ Number of virtual graphics cards requested by the virtual machine.
- ④ Additional memory overhead:
 - If your environment includes a Single Root I/O Virtualization (SR-IOV) network device or a Graphics Processing Unit (GPU), allocate 1 GiB additional memory overhead for each device.
 - If Secure Encrypted Virtualization (SEV) is enabled, add 256 MiB.
 - If Trusted Platform Module (TPM) is enabled, add 53 MiB.

Mermaid ↗

□□□□□

```
```mermaid
graph TD;
 A-->B;
 A-->C;
 B-->D;
 C-->D;
```
```



□□ [Markdown Preview Mermaid ↗](#) □□□ VSCode □□□□□

MDX

[MDX](#) [Markdown](#) [Markdown](#) [JSX](#) [rspress MDX](#)

rspress

doom

Overview

Directive

ExternalSite

ExternalSiteLink

AcpApis0overview ExternalApis0overview

Term

props

TermsTable

props

JsonViewer

rspress

rspress [.mdx](#)

- Badge

- [Card](#)
- [LinkCard](#)
- [PackageManagerTabs](#)
- [Steps](#)
- [Tab/Tabs](#)
- [Toc](#)

安装 @rspress/core/theme 包

preview.mdx

```
import { SourceCode } from '@rspress/core/theme'

<SourceCode href="/" />
```

doom 文档

doom 是一个基于 MDX 的文档生成工具，它可以帮助你快速生成文档。

Overview

doom 支持以下功能：

Directive

doom 提供了 [Directive](#) 功能，用于在文档中插入警告或提示。

```
- 在 `doc/en` 目录下创建 `doc/zh` 目录，用于存放中文文档。

<Directive type="danger" title="警告">
  这是一个危险的警告，请谨慎操作。
  `doc/zh` 目录下存放中文文档。
</Directive>
```

- 在 `(doc/en)` 中指定语言 `doc/zh` 以指定要使用的文档语言。

例如

在 `doc/zh` 中指定要使用的文档语言。

ExternalSite

在文档中

```
<ExternalSite name="connectors" />
```

Note

在 `DevOps` 中指定要使用的文档语言。 `DevOps` 文档语言。 [DevOps 文档](#)

ExternalSiteLink

在文档中

```
<ExternalSiteLink name="connectors" href="link.mdx#hash" children="Content" />
```

[Content](#)

TIP

在 `mdx` 中 `<ExternalSiteLink name="connectors" href="link" children="Content" />` 指定要使用的文档语言。

```
<ExternalSiteLink name="connectors" href="link">
  Content {/*   `p`   */}
</ExternalSiteLink>
```

p children

AcpApisOverview ExternalApisOverview

API

```
<AcpApisOverview />
{/* same as following */}
<ExternalApisOverview name="acp" />

<ExternalApisOverview name="connectors" />
```

Note

ACP APIs [ACP APIs](#)

Note

DevOps APIs [DevOps APIs](#)

Term

```
<Term name="company" textCase="capitalize" />
<Term name="product" textCase="lower" />
<Term name="productShort" textCase="upper" />
```

ACP

props

- `name` :
- `textCase` : `lower` , `upper` , `capitalize`

TermsTable

```
<TermsTable />
```

company		-	Alauda	-	
product		-	Alauda Container Platform	-	
productShort	ACP	-	ACP	-	

props

- `terms` : `NormalizedTermItem[]`

JsonViewer

```
<JsonViewer value={{ key: 'value' }} />
```

```
yaml    json
key: value
```

□□□□□□□□

□□□□□□□□□□□□□□□□□□ `shared` □□□□□□□□□□□□□□□□□□

```
import CommonContent from './shared/CommonContent.mdx'

<CommonContent />
```

□□□□□□□□ `runtime` ↗ □□□ API□□□□□□ `.jsx/.tsx` □□□□□□□□□□ `.mdx` □□□□□□□□

```
// shared/CommonContent.tsx
export const CommonContent = () => {
  const { page } = usePageData()
  return <div>{page.title}</div>
}

// showcase/content.mdx
import { CommonContent } from './shared/CommonContent'
<CommonContent />
```

WARNING

□□□□□□ `.mdx` □□□□□□□□□□ `props` □□□□□□ issue ↗ □□□□□□□□ `props` □□□□□□□□
`.jsx/.tsx` □□□□□□□□

📄📄📄

`alauda` 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄 `en / zh` 📄📄📄📄📄📄📄📄📄📄📄
📄 `public` 📄📄📄 `en / zh` 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

📄📄

- `i18n.json`
- `.ts/.tsx`
- `.mdx`

`i18n.json`

📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄 `docs` 📄📄📄 `i18n.json` 📄📄
📄📄📄📄📄 `useI18n` 📄📄📄📄📄📄📄📄

docs/i18n.json

```
{
  "title": {
    "zh": "📄",
    "en": "Title"
  },
  "description": {
    "zh": "📄",
    "en": "description"
  }
}
```

.ts/.tsx

```
import { useI18n } from '@rspress/runtime'

export const CommonContent = () => {
  const t = useI18n()
  return <h1>{t('title')}</h1>
}
```

.mdx

```
import { useI18n } from '@rspress/runtime'

# {useI18n()('title')}

{useI18n()('description')}
```


API 文档

文档内容 API 文档 API 文档 CRD (Custom Resource Definition) 文档内容

```
├─ apis
│   ├── advanced-apis # API
│   ├── crds # CRDs
│   └── references # 文档
```

文档

API

props

CRD

props

文档

props

openapi 文档

API

advanced-apis/codeQualityTaskSummary.mdx

```
# CodeQualityTaskSummary

<OpenAPIPath path="/plugins/v1alpha1/template/codeQuality/task/{task-id}/summ
```

CodeQualityTaskSummary

props

- path : OpenAPI schema paths
- pathPrefix : api.pathPrefix
- openapiPath : openapi

CRD

crds/ArtifactCleanupRun.mdx

```
# ArtifactCleanupRun

<K8sCrd name="artifactcleanupruns.artifacts.katanomi.dev" />
```

ArtifactCleanupRun

props

- name : CRD metadata.name
- crdPath : openapi CRD

references/CodeQuality.mdx

CodeQuality

```
<OpenAPIRef schema="v1alpha1.CodeQuality" />
```

CodeQuality

props

- `schema` : OpenAPI schema `definitions` (v2) or `component/schemas` (v3) から取得
- `openapiPath` : `openapi` のパス

openapi

`OpenAPIPath` は `OpenAPIRef` が参照する `openapi` のパスを指定する。
`openapi` のパスは `openapiPath` で指定する。

```
<OpenAPIPath  
  path="/plugins/v1alpha1/template/codeQuality/task/{task-id}/summary"  
  openapiPath="shared/openapis/katanomi.json"  
>
```

权限管理

```
<K8sPermissionTable functions={['devops-testplans', 'devops-testmodules']} />
```

权限

props

权限

props

- `functions`: `string[]` - 权限资源名称 `FunctionResource` 名称

权限

权限	权限	权限 名称	权限 名称	权限 名称	权限 名称	权限 名称	权限 名称
权限 devops- testplans	权限	✓	✓	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗

☐ Markdown ☐☐☐☐

```
<!-- reference-end -->
```

```
{/* reference-end */}
```

```
□□□□□ <!-- reference-start#ref --> □□□□□
```

frontmatterMode

--	--	--	--	--	--

reference:

```

- repo: alauda-public/product-doc-guide # 
branch: # [string] 
publicBase: # [string] /images/xx.png do
sources:
  - name: anchor # 
    path: docs/index.mdx# # doom.conf
    ignoreHeading: # [boolean] true
    processors: # 
      - type: ejTemplate
        data: # ej `<%= data.xx %>` 
    frontmatterMode: merge # frontmatter ignore ign

```

frontmatterMode

- **ignore** frontmatter frontmatter
- **merge** frontmatter key
- **replace** frontmatter frontmatter
- **remove** frontmatter

-
-
-
-

```
doom build # 
doom serve # 
```

```
doom build dist -v 
```



```
# ===== release-4.0 == 4.0 ==
doom build -v 4.0 # == 4.0 ===== dist/4.0===== {base}/4.0
doom build -v master # == master ===== dist/master===== {base}/master
doom build -v {other} # ===== dist/{other}===== {base}/{other}

# unversioned = unversioned-x.y =====
doom build -v unversioned # ===== dist/unversioned===== {base}
doom build -v unversioned-4.0 # ===== 4.0 ===== dist/unversio
```

□□□□□□

```
|— console-platform
|   |— 4.0
|   |— 4.1
|   |— index.html
|   |— overrides.yaml
|   └— versions.yaml
|— console-devops-docs
|   |— 4.0
|   |— 4.1
|   |— index.html
|   |— overrides.yaml
|   └— versions.yaml
|— console-tekton-docs
|   |— 1.0
|   |— 1.1
|   |— index.html
|   |— overrides.yaml
|   └— versions.yaml
```

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Redirecting...</title>
    <meta http-equiv="refresh" content="0; url=/console-docs/4.1" />
  </head>
  <body>
    <p>Redirecting to <a href="/console-docs/4.1">/console-docs/4.1</a></p>
  </body>
</html>
```

🐞🐞🐞🐞🐞🐞🐞

overrides.yaml

```
# 🐞🐞🐞🐞🐞🐞🐞🐞
title:
  en: Doom - Alauda
  zh: Doom - 🐞🐞
logoText:
  en: Doom - Alauda
  zh: Doom - 🐞🐞
```

versions.yaml

```
- '4.1'
- '4.0'
```