

--	--	--	--

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

11

□ □ □ □

Markdown MDX

□ □ □ □

--	--	--	--

- □□□□□□□□□□□□□□□□□□
- □□□□□□□□
- □□□□□

□□ **Markdown** □□□□ **MDX**

[illegible]



0000000000 MDX 00 ↗ 0

0000

0000000000 Doom 00

00

00

0000

000000

00

11

□□ `doom` □□□□

4/4

□ □ □ □

API □□□□

Page 10 of 10

Page 10 of 10

Page 10 of 10

□ □ □ □ □

Page 10

□ □ □ □ □ □ □ □ □

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

sites.yaml

□ □ □ □

□ □ □ □ □ □ □ □ □

Page 1 of 1

11

4/4

333

11

11

## Markdown

## Callouts

Mermaid

MDX

00 MDX 000000000000000000

rspress 00

doom 00

00000000

000

000000000000000000

i18n.json

.ts/.tsx

.mdx

API 00

00 API

CRD

0000

00 openapi 00

000000

props

00

0000

000000

00

00000000000000000000 ACP 00

00000

0000

000 ACP

📄📄

📄📄

📄📄📄

📄📄📄📄

📄📄📄📄📄

📄📄📄📄📄

📄📄📄

📄📄📄📄📄📄

📄📄📄

📄📄 PDF

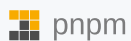
📄📄📄

📄📄📄📄

📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

```
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 '# 你好' > 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:

<code>-V, --version</code>	output the version number
<code>-c, --config &lt;config&gt;</code>	Specify the path to the config file
<code>-v &lt;version&gt;</code>	Specify the version of the documentation, c
<code>-b, --base &lt;base&gt;</code>	Override the base of the documentation
<code>-p, --prefix &lt;prefix&gt;</code>	Specify the prefix of the documentation bas
<code>-f, --force [boolean]</code>	Force to <ol style="list-style-type: none"> <li>1. fetch latest reference remotes or scaffo</li> <li>2. translate ignore hash equality check and</li> </ol> Ignore internal routes (default: false)
<code>-i, --ignore [boolean]</code>	
<code>-d, --download [boolean]</code>	Display download pdf link on nav bar (defau
<code>-e, --export [boolean]</code>	Run or build in exporting PDF mode, `apis/*
<code>-I, --include &lt;language...&gt;</code>	Include <b>**only**</b> the specific language(s),
<code>-E, --exclude &lt;language...&gt;</code>	Include all languages except the specific l
<code>-o, --out-dir &lt;path&gt;</code>	Override the `outDir` defined in the config
<code>-r, --redirect &lt;enum&gt;</code>	Whether to redirect to the locale closest t
<code>-R, --edit-repo [boolean url]</code>	Whether to enable or override the `editRepo
<code>-n, --no-open [boolean]</code>	Do not open the browser after starting the
<code>-h, --help</code>	display help for command

### Commands:

<code>dev [options] [root]</code>	Start the development server
<code>build [root]</code>	Build the documentation
<code>preview serve [options] [root]</code>	Preview the built documentation
<code>new [template]</code>	Generate scaffolding from templates
<code>translate [options] [root]</code>	Translate the documentation
<code>export [options] [root]</code>	Export the documentation as PDF, `apis/**`
<code>help [command]</code>	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
  -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 en)

-t, --target <language> Document target language, one of en, zh, ru (default en)

-g, --glob <path...> Glob patterns for source dirs/files

-C, --copy [boolean] Wether to copy relative assets to the target directory

-h, --help display help for command

- -g, --glob 指定源文件的路径 glob 指定要匹配的 glob 模式

1. yarn translate -g abc xyz 将 <root>/<source>/abc 中的文件复制到 <root>/<target>/xyz 中

2. yarn translate -g '\*' 将 <root>/<source> 中的所有文件复制到 <root>/<target> 中

- -C, --copy 指定是否将源文件复制到目标目录 false 表示不复制

- 示例

1. /<source>/abc.jpg 复制到 <root>/public/<source>/abc.jpg 中  
 <root>/public/<target>/abc.jpg 中不存在该文件，因此不会复制

2. <root>/<source>/abc.mdx 复制到 ./assets/xyz.jpg 中  
 <root>/<source>/assets/xyz.jpg 存在，因此复制到 <root>/<target>/assets/xyz.jpg 中

3. <root>/<source>/abc.mdx 复制到 ./assets/<source>/xyz.jpg 中  
 <root>/<source>/assets/<source>/xyz.jpg 存在，因此复制到 <root>/<target>/assets/<target>/xyz.jpg 中  
 ./assets/<target>/xyz.jpg 中不存在该文件，因此不会复制

- 📄📄📄📄📄📄
  1. `/<source>/abc.jpg` 📄📄📄 `<root>/public/<target>/abc.jpg` 📄📄📄📄📄📄  
📄📄📄📄 `/<target>/abc.jpg` 📄📄📄📄📄📄📄📄📄📄
  2. `<root>/<source>/abc.mdx` 📄📄 `./assets/<source>/xyz.jpg` 📄📄📄  
`<root>/<target>/assets/<target>/xyz.jpg` 📄📄📄📄📄📄📄📄📄  
`./assets/<target>/xyz.jpg` 📄📄📄📄  
`../<source>/assets/<target>/xyz.jpg`

**WARNING**

📄📄📄📄📄 `-g '*'` 📄📄📄📄📄📄 `source` 📄 `target` 📄📄📄📄📄  
`internalRoutes` 📄📄📄📄 `target` 📄📄📄📄📄

**TIP**

📄📄📄📄📄📄 `AZURE_OPENAI_API_KEY` 📄📄📄📄📄📄📄 `Leader` 📄

📄📄📄📄📄📄📄📄📄📄📄📄

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

📄📄📄📄📄📄📄📄📄

📄📄 **PDF**

**WARNING**

📄📄📄📄📄📄📄📄 `yarn build` 📄📄📄

```
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](https://build-harbor.alauda.cn/frontend/playwright-runner:doom)

```
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
```

0000000000000000



📄

📄 `doom` 📄📄

📄📄

📄📄

API 📄📄

📄📄📄📄📄

📄📄📄

📄📄📄

📄📄📄

📄📄📄📄📄

📄📄📄📄📄📄

📄📄📄📄📄

`sites.yaml` 📄

📄📄

📄📄📄📄📄📄

📄📄📄

📄

📄“📄📄📄📄”📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

📄📄

📄📄

📄

📄

## Markdown

Callouts

Mermaid

## MDX

📄 MDX 📄📄📄📄📄📄📄📄📄📄

rspress 📄

doom 📄

📄📄📄📄📄

📄📄

📄📄📄📄📄📄📄📄📄

i18n.json

.ts/.tsx

.mdx

## API 📄

📄 API

CRD

📄📄📄

📄 openapi 📄

📄📄📄📄📄

props

📄





□ □ □ □

API □□□□



□ □ □ □ □ □

## frontmatterMode

□ □ □ □ □ □

--	--	--	--	--

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

□ □ □ □ □ □ □ □

sites.yaml □□

□ □ □ □

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

--	--	--	--

```
0000000000000000 yml 00000000 doom.config.yml 0
```

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

```
□□□□ .js/.ts/.mjs/.mts/.cjs/.cts □□□□□□□□
```

```

js/ts  @alauda/doom/config  defineConfig

```

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

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

export default defineConfig({})
```

## Options

- `lang` 语言，支持 `en` 和 `zh`，默认为 `en`，如果未指定，则默认为 `en`。
- `title` 标题，默认为 `alauda`。
- `logo` logo 的公共路径，默认为 `public/logo`。
- `logoText` logo 的文本，默认为 `alauda`。
- `icon` 图标，默认为 `logo`。
- `base` 基础路径，默认为 `product-docs`。
- `outDir` 输出目录，默认为 `dist/{base}/{version}`。

## API

```
# dooconfig.* glob json/yaml
permission:
  functionresources:
    # `kubectl get functionresources`
    - docs/shared/functionresources/*.yaml
  roletemplates:
    # `kubectl get roletemplates -l auth.cpaas.io/roletemplate.official=true`
    - docs/shared/roletemplates/*.yaml
```

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

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: ejsTemplate
        data: # ejs `<%= data.xx %>` 
    frontmatterMode: merge # frontmatter ignore ign
```

## frontmatterMode

- `ignore` `frontmatter` `frontmatter`
- `merge` `frontmatter` `key`
- `replace` `frontmatter` `frontmatter`
- `remove` `frontmatter`

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

```
releaseNotes:
```

queryTemplates:

fixed: # 000 ejs 000 jqL 00

unfixed:

```
<!-- release-notes-for-bugs?template=fixed&project=DevOps -->
```

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

--	--	--	--	--

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



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

```
onlyIncludeRoutes: # 匹配 glob 匹配的 docs 中的 cli 包，忽略 ` ` 匹配的
  - '*/internal/**'
internalRoutes:
  - '*/internal/overview.mdx'
```

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

plaintext

```
sites.yaml [REDACTED] [REDACTED]
```

```
[REDACTED]
```

--	--	--	--

translate:

```
# #####ejjs ##### `sourceLang`, `targetLang`, `userPrompt` □ `additionalPrompts`  
# □□ `sourceLang` □ `targetLang` □ `##` □ `##` #####  
#   `userPrompt` #####  
#   `additionalPrompts` □□□ `frontmatter.i18n` □□ `additionalPrompts` □□□  
# #####  
  
systemPrompt: |  
  
## □□  
  
#####<%= targetLang %>#####<%= sourceLang %>□□□<%=  
  
## □□  
  
- #####<%= sourceLang %>#####<%= targetLang %>#####  
- ##### MDX ##### MDX ##### jsx ##### <Overview />#####  
- #####  
- MDX #####MDX ##### MDX #####  
  - <Overview /> □□ Overview #####  
  - <Tab label="value">#####</Tab>□label □ key ##### "value" #####  
  
<%= terms %>  
  
- #####  
- {/* release-notes-for-bugs */}  
- <!-- release-notes-for-bugs -->  
  
- #####  
- {/* reference-start */}  
- {/* reference-end */}  
- <!-- reference-start -->  
- <!-- reference-end -->  
  
- ##### \\< □ \\{ #####  
- ##### Markdown □□□ frontmatter, ##### frontmatter.i18n □□□  
  
## □□  
  
#####  
  
1. □□<%= sourceLang %>#####<%= targetLang %>#####  
2. #####  
  - □□<%= targetLang %>#####  
  - #####  
  - #####  
  
3. #####<%= targetLang %>#####  
4. #####<%= targetLang %>#####<%= targetLang %>#####  
  
#####  
  
<%= userPrompt %>
```



```
<%= additionalPrompts %>
```

```
userPrompt: # `systemPrompt` `ejs`
```

```
editRepoBaseUrl: alauda/doom/tree/main/docs # https://github.com/
```

```
lint:
```

```
cspellOptions: # cspell https://github.com/streetsidesoftware/csp
```

11

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

```
-- --
title:  []
description:  []
author:  []
category:  []
-- --
```

```

- - -
weight: 1
- - -

```

[illegible]

11

index.mdx Overview

```
#  
  
<Overview />
```

# Markdown

[gfm](#) Doom Markdown

Callouts

Mermaid

## Callouts

### NOTE

- ; , % , # , // , /\*\* \*/ , -- <!-- -->
- [\\!code callout]
- :::callouts <div class="doom-callouts"> <Callouts>

```
```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.

```
:::
```

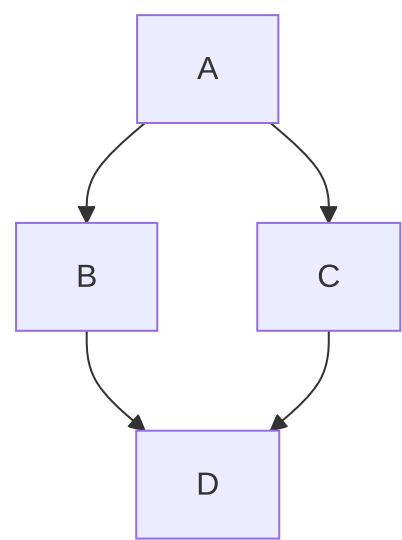
```
Memory overhead per virtual machine ≈ (1.002 × requested memory) \
    + 218 MiB \
    + 8 MiB × (number of vCPUs) \
    + 16 MiB × (number of graphics devices) \
    + (additional memory overhead)
```

- 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
- AcpApisOverview ExternalApisOverview

Term

- props
- TermsTable
- props
- JsonViewer

## rspress

rspress [rspress](#) [.mdx](#)

- Badge
- Card



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

在 `@rspress/core/theme` 中

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

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

# doom 文档

`doom` 是一个用于生成文档的工具

## Overview

文档概述

## Directive

在文档中使用 `Directive` 指令

```
- 在 (`doc/en`) 中定义 `doc/zh` 内容

<Directive type="danger" title="危险">
  危险内容
  `doc/zh` 内容
</Directive>
```

- 在 `( doc/en )` 中定义  `doc/zh`  内容

11

doc/zh

11

## ExternalSite

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

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

### Note

























DevOps ☐ ☐ ☐ ☒

## ExternalSiteLink

□ □ □ □ □ □ □ □ □ □

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

[Content ↗](#)

**TIP**

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

□ □ □ □ □ □ □ □ □ □

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

000000000000 **p** 00000000000000000000 **children** 0000

## AcpApisOverview ExternalApisOverview

0000000 API 0000

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

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

### Note

00 ACP APIs 000000000000 [ACP APIs 00 ↗](#) 0

### Note

00 DevOps 000 APIs 000000000000 [DevOps 000 APIs 00 ↗](#) 0

## Term

00000000000000000000

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

000 00000000 ACP

### props

- **name** : 00000000000000000000
- **textCase** : 0000000000000000 **lower** , **upper** , **capitalize**

TermsTable

```
<TermsTable />
```

Key	Value	Label	Category	Icon	Color
company	Alauda	-	Alauda	-	Blue
product	Alauda Container Platform	-	Alauda Container Platform	-	Blue
productShort	ACP	-	ACP	-	Blue

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` 📄📄📄📄📄📄📄📄📄

```
{
  "title": {
    "zh": "标题",
    "en": "Title"
  },
  "description": {
    "zh": "描述",
    "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

本页面提供 API 文档 API 文档 CRD (Custom Resource Definition) 文档

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

## API

- API
  - props
- CRD
  - props
- 文档
  - props
- openapi 文档

## API

```
# CodeQualityTaskSummary

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

CodeQualityTaskSummary



## props

- `path` : OpenAPI schema `paths` 下的路径
- `pathPrefix` : 路径前缀 `api.pathPrefix`
- `openapiPath` : 指向 `openapi` 的路径

## CRD

```
# ArtifactCleanupRun
```

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

指向 `ArtifactCleanupRun`

## props

- `name` : CRD `metadata.name`
- `crdPath` : 指向 `openapi` 下的 CRD 的路径

## CodeQuality

```
# CodeQuality
```

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

指向 `CodeQuality`

## props

- `schema` : OpenAPI schema `definitions` (v2) or `component/schemas` (v3) 下的 schema

- `openapiPath`: `openapi`

□□ **openapi** □□

```

  OpenAPIPath  OpenAPIRef  openapi  openapiPath
  openapi

```

```
<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 FunctionResource 列表

权限

权限	权限	权限 资源	权限 资源	权限 资源	权限 资源	权限 资源	权限 资源
权限 devops- testplans	权限	✓	✓	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗
	权限	✓	✗	✓	✓	✓	✗
权限 devops-	权限	✓	✓	✓	✓	✓	✗

模块	测试	测试用例 通过	测试用例 失败	测试用例 通过	测试用例 通过	测试用例 通过	测试用例 通过
testmodules	测试	✓	✗	✓	✓	✓	✗
	测试	✓	✗	✓	✓	✓	✗
	测试	✓	✗	✓	✓	✓	✗



**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: # ejs `<%= data.xx %>` 
    frontmatterMode: merge # frontmatter ignore ign

```

**frontmatterMode**

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

□ □ □ □ □ □ □ □

`doom` `serve` # □□□□□□□□□□

FROM build-harbor.alauda.cn/ops/alpine:latest

WORKDIR /docs

COPY . dist

□ □ □ ACP

--	--	--	--	--

```

$ doom build -v
$ 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/unversioned-4.0
```





```
|— 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
```

```
<!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>
```

☐☐☐☐☐☐☐☐

```
# console-platform
# https://gitlab-ce.alauda.cn/idp/Doom/-/blob/master/src/terms.ts#L11
terms:
  company:
    en: Alauda
    zh:
  product:
    en: Alauda Container Platform
    zh:
  productShort:
    en: ACP

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

- '4.1'
- '4.0'

[chart-frontend](#) [alauda-docs](#) [check-alauda-docs](#)

[doom](#) [idp](#) [webapp](#)

## INFO

PR gitops