




□□□□ >

Doom  [rspress](#)  Alauda            

□ □ □ □

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

MDX ██████████ Markdown ████████ Markdown █████  
██████ React ███



0000000000 MDX 00 ↗ 0

0000

0000000000 Doom 00

00

00

0000

000000

00



MDX

📄 MDX 📄📄📄📄📄📄📄📄📄📄

rspress 📄

doom 📄

📄📄📄📄📄

📄📄📄

📄📄📄📄📄📄📄📄📄📄

i18n.json

.ts/.tsx

.mdx

API 📄

📄 API

CRD

📄📄📄

📄 openapi 📄

📄📄📄📄📄

props

📄

📄📄📄

📄📄📄📄📄

00

000000000000000000000000 ACP 00

000000

000000

0000000

000000000

📄📄

📄📄

📄📄📄

📄📄📄📄

📄📄📄📄📄

📄📄📄📄📄

📄📄📄

📄📄📄📄📄📄

📄📄📄

📄📄 PDF

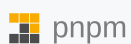
📄📄📄

📄📄📄📄

📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

```
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 '# 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>-a, --algolia [boolean alauda]</code>	Whether to enable or use the alauda (docs.a
<code>-S, --site-url</code>	Whether to enable the siteUrl for sitemap g
<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>lint [root]</code>	Lint the documentation
<code>help [command]</code>	display help for command

000000

00 yarn dev 00000000000000000000

```
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
```

000000

00 yarn build 000000000000000000 dist 00000000

0000

00 yarn serve 000000000000000000 -b, -p 0000000000000000 -b, -p 0

0

00000000

00 yarn new 000000000000000000

0000

```
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 is en)

-t, --target <language> Document target language, one of en, zh, ru (default is 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 中不存在文件 /<target>/abc.jpg

2. <root>/<source>/abc.mdx 复制到 ./assets/xyz.jpg 中  
 <root>/<source>/assets/xyz.jpg 中不存在文件 <root>/<target>/assets/xyz.jpg 中不存在文件

3. <root>/<source>/abc.mdx 复制到 ./assets/<source>/xyz.jpg 中  
 <root>/<source>/assets/<source>/xyz.jpg 中不存在文件  
 <root>/<target>/assets/<target>/xyz.jpg 中不存在文件  
 ./assets/<target>/xyz.jpg

- 📄📄📄📄📄📄
  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 📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄
```

```
.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

000000000000000000000000 ACP 00

00000

00000

000000

00000000

11

Sitemap ☐ ☐

□ □ □ □

```
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({})
```

--	--	--	--

- `lang` 指定する言語。指定可能な言語は `en` (英語) と `ja` (日本語) である。指定しない場合は `null` または `undefined` である。
- `title` 指定するタイトル。
- `logo` 指定するロゴ。指定可能なロゴは `public` (パブリック) と `doom` (ドーム) である。指定しない場合は `alauda logo` である。
- `logoText` 指定するロゴのテキスト。
- `icon` 指定するアイコン。指定可能なアイコンは `logo` である。
- `base` 指定するベースパス。指定可能なベースパスは `product-docs` である。指定しない場合は `/` である。
- `outDir` 指定する出力ディレクトリ。指定可能な出力ディレクトリは `dist/{base}/{version}` である。指定しない場合は `dist/{outDir}/{version}` である。指定可能な出力ディレクトリは `version` である。

API ☐ ☐ ☐ ☐

```
# doo config.* 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:** # ejs jql

**unfixed:**

release-notes.md

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

release-notes.mdx

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

```

template=fixed&project=DevOps  fixed  queryTemplates 
query  project=DevOps  ejs  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: # 00000 glob 00000 docs 0000 cli 00 00 -i, --ignore` 00000
  - '*/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:

```
# ===== `sourceLang`, `targetLang`, `userPrompt` & `additionalPrompts` =====  
#   `sourceLang`   `targetLang`   ``   ``  
#       `userPrompt`  
#       `additionalPrompts`   `frontmatter.i18n`   `additionalPrompts`  
#  
  
systemPrompt: |  
##  
  
=====<%= targetLang %>=====<%= sourceLang %>=====<%=  
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 %>=====  
  
=====
```



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

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

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



```
lint:
```

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

## 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
```

□ □ □ □

111

11

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

# Markdown

00000 [gfm](#) 0000Doom 000000000 Markdown 00000

00

Callouts

Mermaid

## Callouts

0000000

### NOTE

- 000000000000000000000000 ; , % , # , // , /\*\* \*/ , -- 0 <!-- --> 0
- 000000000000000000000000 [\!code callout] 0000
- 000 :::callouts 000000000000000000000000 <div class="doom-callouts"> 0  
<Callouts> 0000

```
```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高质量的文档。

### Overview

doom 支持多种文档格式，包括 Markdown、MDX 等。

### Directive

doom 提供了多种 Directive，用于控制文档的生成和输出。

```
- 使用 `doc/en` 生成英文文档，`doc/zh` 生成中文文档

<Directive type="danger" title="警告">
  这是一个危险的操作，可能会导致数据丢失。
  `doc/zh` 生成中文文档
</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 文档语言在 [DevOps 文档](#) 中指定。

## 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": "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

API 文档 API 文档 CRD (Custom Resource Definition) 文档

```
├─ 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

&lt;OpenAPIRef schema="v1alpha1.CodeQuality" /&gt;

仕様 CodeQuality 仕様

## props

- schema : OpenAPI schema definitions (v2) or component/schemas (v3) 仕様
- openapiPath : 仕様 openapi 仕様

## 仕様 openapi 仕様

仕様 OpenAPIPath 仕様 OpenAPIRef 仕様 openapi 仕様

仕様 openapi 仕様 openapiPath 仕様

```
<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 权限资源

权限

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

测试用例	测试用例	测试用例	测试用例	测试用例	测试用例	测试用例	测试用例
测试用例 devops- testmodules	测试用例	测试用例 测试用例	测试用例 测试用例	测试用例 测试用例	测试用例 测试用例	测试用例 测试用例	测试用例 测试用例
	测试用例	✓	✓	✓	✓	✓	✗
	测试用例	✓	✗	✓	✓	✓	✗
	测试用例	✓	✗	✓	✓	✓	✗
	测试用例	✓	✗	✓	✓	✓	✗

--	--	--	--

☐ Markdown ☐☐☐☐

```
<!-- reference-start#name -->
```

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

MDX

```
{/* reference-start#name */}
```

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

name

```
□□□□□□ <root>/public/_remotes/<name> □□□□
```

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

```

**frontmatterMode**

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

11

11

□ □ □ □ □

□ □ □ □ □

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

□ □ □ □ □ □ □ □

--	--	--	--	--

[illegible]

doom build # □□□□□□

doom serve # □□□□□□□□□□

--	--	--	--	--

```

$ 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
```

index.html

