

--	--	--	--

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

11

□ □ □ □

Markdown MDX

□ □ □ □

--	--	--	--

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

□□ **Markdown** □□□□ **MDX**

MDX  Markdown  Markdown 
 React 



0000000000 MDX 00 ↗ 0

0000

0000000000 Doom 00

00

00

0000

000000

00

11

□□ `doom` □□□□

4/4

API ☐ ☐ ☐ ☐

□ □ □ □ □ □ □ □

Page 10 of 10

□ □ □ □ □

□ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □ □

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

sites.yaml □□

□ □ □ □

□ □ □ □ □ □ □ □ □

Page 10

Algolia ☐☐☐☐

Sitemap ☐ ☐

11

□ □ □ □

3/3

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

000000

000000

0000000

000000000

📄📄

📄📄

📄📄📄

📄📄📄📄

📄📄📄📄📄

📄📄📄📄📄

📄📄📄

📄📄📄📄📄📄

📄📄📄

📄📄 PDF

📄📄📄

📄📄📄📄

📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

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

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

 `npm`  `yarn`  `pnpm`  `bun`

```
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:

-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	Whether to enable Algolia search (default:
-S, --site-url	Whether to enable the siteUrl for sitemap g
-n, --no-open [boolean]	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
```

```
  -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 指定源目录的glob模式，glob 模式指定源目录的搜索范围。

1. yarn translate -g abc xyz 指定 <root>/<source>/abc , <root>/<source>/xyz 指定 <root>/<target>/abc , <root>/<target>/xyz 指定

2. yarn translate -g '*' 指定 <root>/<source> 指定

- -C, --copy 指定是否复制源目录的相对资产到目标目录，false 指定

- 指定

1. /<source>/abc.jpg 指定 <root>/public/<source>/abc.jpg 指定 <root>/public/<target>/abc.jpg 指定 /<target>/abc.jpg 指定

2. <root>/<source>/abc.mdx 指定 ./assets/xyz.jpg 指定 <root>/<source>/assets/xyz.jpg 指定 <root>/<target>/assets/xyz.jpg 指定

3. <root>/<source>/abc.mdx 指定 ./assets/<source>/xyz.jpg 指定 <root>/<source>/assets/<source>/xyz.jpg 指定 <root>/<target>/assets/<target>/xyz.jpg 指定 ./assets/<target>/xyz.jpg 指定

- 📄📄📄📄📄📄
 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
```

Sitemap ☐ ☐

11

Markdown

Callouts

Mermaid

MDX

📄 MDX 📄📄📄📄📄📄📄📄📄📄

rspress 📄

doom 📄

📄📄📄📄📄

📄📄

📄📄📄📄📄📄📄📄📄

i18n.json

.ts/.tsx

.mdx

API 📄

📄 API

CRD

📄📄📄

📄 openapi 📄

📄📄📄📄📄

props

📄

0000

000000

00

00000000000000000000 ACP 00

00000

00000

000000

00000000

4/4

API ☐ ☐ ☐ ☐

□ □ □ □ □ □ □ □

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

frontmatterMode

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

□ □ □ □ □

□ □ □ □ □ □ □ □

□ □ □ □ □ □ □ □

sites.yaml □□

□ □ □ □

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

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

Algolia ☐ ☐ ☐ ☐

Sitemap ☐ ☐

--	--	--	--

```
00000000000000000000 yaml 0000000000 doom.config.yaml 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({})
```

- lang en null undefined
- title
- logo logo public doom alauda logo
- logoText logo
- icon favicon logo
- base product-docs /
- outDir dist/{base}/{version} dist/{outDir}/{version} version

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
```

```
# 是否生成 openapi 文档
```

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

references:

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

```
# 是否生成 API 文档 gateway 模式
```

```
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

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

0000000000

00000000

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  ejls  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: # ooooo glob ooooo docs oooo cli oo ` -i, --ignore` oooooo
- '*/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` □ `additionalPrompts`
# □□ `sourceLang` □ `targetLang` □ `□□` □ `□□` □□□□□□
#     `userPrompt` □□□□□□□□□□□□□□
#     `additionalPrompts` □□□ `frontmatter.i18n` □□ `additionalPrompts` □□□
# □□□□□□□□□□□□□□□□□□□□□□□□
```

```
systemPrompt: |
```

□□

```

#####<%= targetLang %>#####<%= sourceLang %>###<%

```

□□

```
- =====<%= sourceLang %>=====<%= targetLang %>=====
-      MDX      MDX      jsx      <Overview />=====
-      =====
-      MDX      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 #####
```

□ □

□ □ □ □ □ □ □ □ □ □

[illegible]

```
<%= userPrompt %>
```



```
<%= 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
```

index

index.md index.mdx

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

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

index.mdx 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.

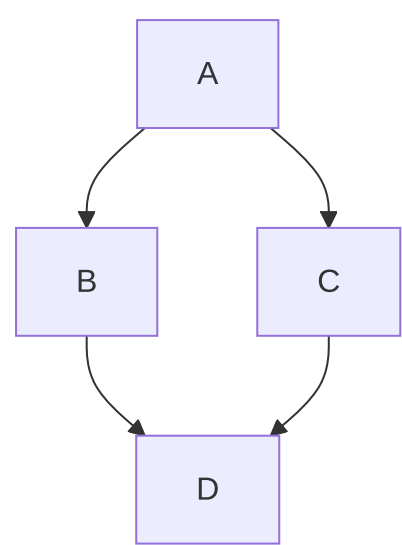
```
:::
```

```
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 [.mdx](#)

- Badge
- Card

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

在 `preview.mdx` 文件中引入 `@rspress/core/theme` 组件

```
preview.mdx

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

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

doom 文档

`doom` 是一个用于生成文档的工具，它支持 MDX 格式，并提供了丰富的组件和 API 来帮助你快速搭建文档网站。

Overview

本文将介绍 `doom` 的基本用法和核心概念。

Directive

在 `doom` 中，`Directive` 是一个用于在 MDX 文件中插入 HTML 元素的组件。它支持多种属性，如 `type` 和 `title`。

```
- 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">
  这是一个危险的操作，可能会导致数据丢失。
  `doc/zh` 目录下的文件将被删除。
</Directive>
```

- 在 `(doc/en)` 中指定语言 `doc/zh` 以指定文档语言。默认语言为 `doc/en`。

注意

在 `doc/zh` 中指定语言。

ExternalSite

在文档中指定外部网站。

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

Note

在 `DevOps` 文档中指定外部网站。在 `DevOps` 文档中指定外部网站。
[DevOps 文档](#) [↗](#)

ExternalSiteLink

在文档中指定外部网站链接。

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

[Content](#) [↗](#)

TIP

在 `mdx` 中指定 `<ExternalSiteLink name="connectors" href="link" children="Content" />`。
在文档中指定外部网站链接。

```
<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
```

```
<CommonContent />
```

`.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

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

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

文档

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

API 文档

```
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" />
```

props

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

openapi

OpenAPIPath OpenAPIRef openapi
openapi openapiPath

```
<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 ID	测试用例 描述	测试用例 预期结果	测试用例 实际结果	测试用例 通过/失败	测试用例 备注
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 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'