




□□□□ >

Doom  [rspress](#)  Alauda            

□ □ □ □

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

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



0000000000 MDX 00 ↗

0000

0000000000 Doom 00

00

00

0000

000000

00



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

0000000

00

00000000000000000000000000 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:

-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 <b>only</b> 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 [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

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 00000000000000000000 dist 00000000

0000

00 yarn serve 00000000000000000000 -b , -p 0000000000000000 -b , -p 0

00000000

00 yarn new 00000000000000000000

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 指定源文件的路径

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

2. yarn translate -g '\*' 将 <root>/<source> 目录下所有文件复制到 <root>/<target> 目录下

- -C, --copy 指定是否将源文件的相对路径复制到目标目录 false 表示不复制

- 示例

1. /<source>/abc.jpg 复制到 <root>/public/<source>/abc.jpg 目录下  
 <root>/public/<target>/abc.jpg 目录下没有文件 /<target>/abc.jpg 目录下没有文件

2. <root>/<source>/abc.mdx 复制到 ./assets/xyz.jpg 目录下  
 <root>/<source>/assets/xyz.jpg 复制到 <root>/<target>/assets/xyz.jpg 目录下  
 <root>/<target>/assets/xyz.jpg 目录下没有文件

3. <root>/<source>/abc.mdx 复制到 ./assets/<source>/xyz.jpg 目录下  
 <root>/<source>/assets/<source>/xyz.jpg 复制到 <root>/<target>/assets/<target>/xyz.jpg 目录下  
 ./assets/<target>/xyz.jpg 目录下没有文件

- 📄📄📄📄📄📄
  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 
```



📄

📄 `doom` 📄📄

📄📄

📄📄

API 📄📄

📄📄📄📄

📄📄📄

📄📄📄

📄📄📄

📄📄📄📄

📄📄📄📄📄

📄📄📄📄

`sites.yaml` 📄

📄📄

📄📄📄📄📄

📄📄📄

Algolia 📄📄

Sitemap 📄

📄

📄“📄📄📄📄”📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄📄

📄📄📄

📄📄

📄

📄

Markdown

Callouts

Mermaid

MDX

📄 MDX 📄📄📄📄📄📄📄📄📄📄

rspress 📄

doom 📄

📄📄📄📄📄

📄📄

📄📄📄📄📄📄📄📄📄

i18n.json

.ts/.tsx

.mdx



API 00

00 API

CRD

0000

00 openapi 00

000000

props

00

0000

000000

00

000000000000000000000000 ACP 00

00000

00000

000000

00000000

11

Sitemap ☐ ☐

--	--	--	--

[illegible]

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

```

❯ js/ts @alauda/doom/config defineConfig

```

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

export default defineConfig({})
```

--	--	--	--

- lang 指定生成文档语言，支持 en 和 zh。默认为 en。  
null 或 undefined
- title 指定生成文档标题。
- logo 指定生成文档 logo。public 指定是否公开。  
doom alauda logo
- logoText 指定生成文档 logo 文本。
- icon 指定 favicon。logo
- base 指定生成文档基础路径。product-docs /
- outDir 指定生成文档输出目录。dist/{base}/{version} 或 dist/{outDir}/{version}。version 为版本号。

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

## frontmatterMode

- `ignore` `alauda-public/product-doc-guide` frontmatter `alauda-public/product-doc-guide` frontmatter
- `merge` `alauda-public/product-doc-guide` frontmatter `alauda-public/product-doc-guide` key `alauda-public/product-doc-guide`
- `replace` `alauda-public/product-doc-guide` frontmatter `alauda-public/product-doc-guide` frontmatter
- `remove` `alauda-public/product-doc-guide` frontmatter

`alauda-public/product-doc-guide`

`alauda-public/product-doc-guide`

## releaseNotes:

### queryTemplates:

fixed: # `alauda-public/product-doc-guide` `ejs` `alauda-public/product-doc-guide` `jq` `alauda-public/product-doc-guide`

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 # oooooooooooooooooooooooooooooo false
```

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

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

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

```
onlyIncludeRoutes: # glob docs cli -i, --ignore`
  - '*/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` o `additionalPrompts` =====
#   `sourceLang` o `targetLang` o `` o ``
#       `userPrompt`
#       `additionalPrompts` o `frontmatter.i18n` o `additionalPrompts` o
#
systemPrompt: |
##
===== <%= targetLang %>===== <%= sourceLang %>o<%= terms %>
- {/* release-notes-for-bugs */}
- <!-- release-notes-for-bugs -->
- {/* reference-start */}
- {/* reference-end */}
- <!-- reference-start -->
- <!-- reference-end -->
- \\< o \\{
- Markdown o frontmatter, o frontmatter.i18n o
##
1. o<%= sourceLang %>o<%= targetLang %>o
2.
- o<%= targetLang %>o
-
-
3. o<%= targetLang %>o
4. o<%= targetLang %>o<%= targetLang %>o
o
<%= 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
```

--	--

□ □ □ □

100

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` □□□□□□□□□□□□□□□□□□□□□□□□□□□□

```

---
title:  □□
description:  □□
author:  □□
category:  □□
---
```

```

- - -
weight: 1
- - -

```

[illegible]

11

index.mdx Overview

```
#  
  
<Overview />
```

# Markdown

[gfm](#)
[Doom](#)
[Markdown](#)



## Callouts

Mermaid

## Callouts

□ □ □ □ □ □

## NOTE

1. `;;, %, #, //, /** */, --` `<!-- -->`
2. `[!\code callout]`
3. `:::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

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 Vite 的静态站点生成器，用于生成静态 HTML 页面。

## Overview

doom 支持以下功能：

## Directive

doom 提供了 `Directive` 组件，用于在文档中插入警告或提示信息。

```
- 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信息。
  `doc/zh` 目录下没有文件。
</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
```

□□□□□□□□

□□□□□□□□□□□□□□□□□□ `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
```

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

□□ CodeQuality□

## props

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

□□ **openapi** □□

```

    OpenAPIPath  OpenAPIRef  openapi  openapiPath
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 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	✓	✗	✓	✓	✓	✗

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-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: ejsTemplate
        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



```
overrides.yaml
```

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

```
versions.yaml
```

- '4.1'
- '4.0'