# 文档介绍

Doom 基于 [rspress↗] 构建，是一套由 Alauda 前端团队主导、结合多个大型项目实践经验沉淀下来的文档解决方案。

## 背景

## 核心功能

- 构建快速，拥有极速的启动和热更新体验
- 支持全文搜索功能
- 支持多语言

## 什么是 Markdown？什么是 MDX

MDX 是一种强大的内容编写方式，你可以在 Markdown 文档中使用 Markdown 语法，也可以在 Markdown 中引入 React 组件。

```
// docs/index.mdx
import { CustomComponent } from './custom';

# Hello World

<CustomComponent />
```

□□□□□□□□□ MDX□ □□↗□

# □□□□

□□□□□□□□□□ Doom □□

# □□

□□

□□□□

□□□□□

# □□

# □□

□□ `doom` □□□□

□□□□

□□□□

## API □□□□

□□□□□□□

□□□□□□

□□□□□□

□□□□□

□□□□□□□

□□□□□□□□

□□□□□□□

`sites.yaml` □□

□□□□

□□□□□□□□

□□□□□□

# □□

□□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□

□□

□□

## Markdown

Callouts

Mermaid

## MDX

□□ MDX □□□□□□□□□□□□□□□□□□□□

rspress □□

doom □□

□□□□□□□□

## □□□

□□□□□□□□□□□□□□□□

`i18n.json`

`.ts/.tsx`

`.mdx`

## API □□

□□ API

CRD

□□□□

□□ openapi □□

## □□□□□□□

`props`

□□

## □□□□

□□□□□□□

## □□

□□□□□□□□□□□□□□□□□□□□□□□ ACP □□

□□□□□

□□□□□

□□□□□□

□□□□□□□

# □□

---

## □□

---

## □□□□

□□□□□□□□□□□□□□□□□□□□□□□□□

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

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

```
npm    yarn    pnpm    bun

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

□□□□□□□□□□□□□□:

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

□ `package.json` □□□□□□□□□:

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

创建一个最简单的配置文件 `doom.config.yml`：

```
title: My Docs
```

然后添加 `tsconfig.json` 以启用类型检查

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

最后添加 `global.d.ts` 以获取全局类型提示

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

- `-g, --glob` □□□□□□□□□□□□□□□□□□□□□□□□□□ `glob` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

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

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

- `-C, --copy` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `false` □□□□□□□□□□□□□□□□□□□□□□□□□□□□

  - □□□□□□□

    1. `/<source>/abc.jpg` □□□□□□□ `<root>/public/<source>/abc.jpg` □ `<root>/public/<target>/abc.jpg` □□□□□□□□□□□□□ `/<target>/abc.jpg`

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

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

默认 `yarn export` 会导出所有 PDF 文档，你可以通过传递 `-b`, `-p` 参数来指定导出某个文档，`-b`, `-p` 参数

注意，我们使用 `playwright ↗`进行导出，使用 `build-harbor.alauda.cn/frontend/playwright-runner:doom` 镜像进行导出，所以你需要确保 你的环境中有该镜像，如果没有

.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` □□

□□□□

□□□□□□□□

□□□□□□

## □□

□□"□□□□□□"□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

□□□

□□

□□

## Markdown

Callouts

Mermaid

## MDX

□□ MDX □□□□□□□□□□□□□□□□□

rspress □□

doom □□

□□□□□□□

## □□□

□□□□□□□□□□□□□□□

`i18n.json`

`.ts/.tsx`

`.mdx`

## API □□

□□ API

CRD

□□□□

□□ openapi □□

## □□□□□□

`props`

□□

# □□□□

□□□□□□

# □□

□□□□□□□□□□□□□□□□□□ ACP □□

□□□□□

□□□□□

□□□□□□

□□□□□□□□

# □□

---

# □□

---

## □□□□

□□□□□□□□□□□□□□□□□ `yaml` □□□□□□□□□ `doom.config.yaml` □ `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 □□□□□□□□□□ doom.config.* □□□□□□□ glob □□□json/yaml □□
  crds:
    - docs/shared/crds/*.yaml
  # OpenAPI □□□□□□□□□□ doom.config.* □□□□□□□ glob □□□json/yaml □□
  openapis:
    - docs/shared/openapis/*.json
  # □□ openapi □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  # □□ https://doom.alauda.cn/apis/references/CodeQuality.html#v1alpha1.CodeQ
  references:
    v1alpha1.CodeQualityBranch: /apis/references/CodeQualityBranch#v1alpha1.C
  # □□□API □□□□□□□□□□□□□□ gateway □□□□□□□□□□□□□□
  pathPrefix: /apis
```

[□□□□□□□ API □□](#)

## □□□□□□□□

```
# □□□□□□□□□□□ doom.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.confi
        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 # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ false
```

---

# □□□□□□□□

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

---

# □□□□□□□□□

```
onlyIncludeRoutes: # □□□□□ glob □□□□□□ docs □□□□ cli □□ `-i, --ignore` □□□□□□
  - '*/internal/**'
internalRoutes:
  - '*/internal/overview.mdx'
```

# □□□□□□□□

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

> **WARNING**
>
> □□□□□□□□□□□□□□□□□□□□□□ `plaintext` □□

# `sites.yaml` □□

`sites.yaml` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□

```
- name: connectors # □□□□□□
  base: /devops-connectors # □□□□□□□□
  version: v1.1 # □□□□□□□□ ExternalSite/ExternalSiteLink □□□□□

  displayName: # □□□□□□□□□□□□□□□□□□□□□□ name
    en: DevOps Connectors
    zh: DevOps □□□

  # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  repo: https://github.com/AlaudaDevops/connectors-operator # □□□□□□□□□□□ gi
  image: devops/connectors-docs # □□□□□□□□□□□□□□□□□□
```

□□□□

```
translate:
  # □□□□□□ejs □□□□□□□□□ `sourceLang`, `targetLang`, `userPrompt` □ `additiona
  # □□ `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.ii8n □□□□

## □□
□□□□□□□□□□
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
```

```
<%= additionalPrompts %>
   userPrompt: # □□□□□□□ `systemPrompt` □□ `ejs` □□□□□□
```

# □□

# □□

# □□□□

每个文档文件会对应到唯一的路由，路由会依据 `index` 文件和目录名称进行计算。为了更好的可读性，你可
以 `index.md` 或 `index.mdx` 作为文档文件的名称，此时它的路由会计算成所在目录的路由，这种方式一般
用于章节下概述性质的文档。

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

有几点需要注意：

1. `public` 目录下的文档是不会参与路由计算的。

2. `public/_remotes` 目录用来存放远程拉取的文档，远程文档一般不会进行版本管理，可以将
   `*/public/_remotes` 加入 `.gitignore` 来忽略它们的提交。

3. `shared` 目录下的文档默认也不会参与路由计算，它一般用于复用文档内容。

# □□□

每个文档文件都可以包含 `frontmatter` 信息，它可以用来定义文档的元信息，比如标题、描述等。

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

所有内容都使用 MDX 编写 `.mdx` 文件，其开头的 `frontmatter` 可以设置页面的属性

## □□

在 `index.md` 或 `index.mdx` 文件中，你也可以设置目录的属性，例如可以通过 `frontmatter` 中的 `weight` 字段来设置目录的顺序，`weight` 越小排序越靠前

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

> **WARNING**
>
> 目录属性的优先级高于其内部文件的属性，如果两者冲突，以目录属性为准

## □□

你可以在每个目录下创建一个特殊的文件，名为 `index.mdx`，用于设置 Overview 页面的内容，该页面会作为目录的默认页面，展示目录的概览信息

```
# □□

<Overview />
```

□□□□□□□□□

# Markdown

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

---

## □□

---

## Callouts

□□□□□□

<div class="doom-callouts">

**NOTE**

1. □□□□□□□□□□□□□□□□□ `;` , `%` , `#` , `//` , `/** */` , `--` □ `<!-- -->` □

2. □□□□□□□□□□□□□□□ `[\!code callout]` □□□□

3. □□□ `:::callouts` □□□□□□□□□□□□□□□□□□□ `<div class="doom-callouts">` □ `<Callouts>` □□□□

</div>

````
```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) n
   - 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.

□□□□□□□□□□□ Shiki Transformers ↗ □

# Mermaid ↗

可以绘制图表

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

安装 Markdown Preview Mermaid ↗ 插件在 VSCode 中查看效果

# MDX

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

## 🔲🔲

## rspress 🔲🔲

`rspress` 🔲🔲🔲🔲🔲🔲🔲🔲🔲[↗](#)🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲 `.mdx` 🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲

- `Badge`
- `Card`
- `LinkCard`
- `PackageManagerTabs`
- `Steps`
- `Tab/Tabs`
- `Toc`

🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲 `@rspress/core/theme` 🔲🔲🔲🔲🔲🔲🔲🔲

preview.mdx

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

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

# doom □□

`doom` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

## Overview

□□□□□□□□□□□□□□□□□

## Directive

□□□□□□□□□□□□□□□□□□↗□□□□□□□□□□□□□ `Directive` □□□□

```
- □□□□□(`doc/en`)□□□□□□□□ `doc/zh` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

  <Directive type="danger" title="□□">
    □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
    `doc/zh` □□□□□□□□□□□□□
  </Directive>
```

- □□□□□( doc/en )□□□□□□□□□ doc/zh □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
  □□□□□□

> □□
>
> □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ doc/zh □□□□□□□□□□□□□
> □□

## ExternalSite

□□□□□□□□□

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

> **Note**
>
> 🔲🔲 DevOps 🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲 DevOps 🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲🔲 🔲

## ExternalSiteLink

🔲🔲🔲🔲🔲🔲🔲🔲🔲

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

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

> **Note**
>
> □□ DevOps □□□ APIs □□□□□□□□□□ □

# Term

□□□□□□□□□□□□□□□□

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

## `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

---

## □□□□□□□

□□□□□□□□□□□□□□□□□□□□□□ `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` □□□□□□□

# 国际化

alauda □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ `en` / `zh` □□□□□□□□□□□□□□□□□□□□□□□□□
□ `public` □□□□□□□ `en` / `zh` □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

# 概览

## 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 □□

□□□□□□□□□□□□□□ API □□□□ API □ CRD (Custom Resource Definition) □□□□□□□□□□
□□□□□□

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

## □□

## □□ 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" />
```

渲染 ArtifactCleanupRun：

## props

- `name` : CRD `metadata.name`
- `crdPath` : 自定义的 openapi 文件地址，默认使用全局的 CRD 地址

---

# 引用类型

```
references/CodeQuality.mdx

# CodeQuality

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

渲染 CodeQuality：

## props

- `schema` : OpenAPI schema `definitions` (v2) or `component/schemas` (v3) 中的名字
- `openapiPath` : 自定义的 openapi 地址

## 嵌入 openapi 组件

使用 `OpenAPIPath` 或 `OpenAPIRef` 组件，可以引用一个 openapi 接口。下面是一个简单的示例，需要指定接口的 openapi 文件路径（如下的 `openapiPath` ），以及接口

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

# □□□□□□

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

## □□

## props

- `functions` : `string[]` - □□□□□□□□ `FunctionResource` □□□□□□

## □□

| □□ | □□ | □□□□□ | □□□□□ | □□□□□ | □□□□□□ | □□□□ | □□□□□ |
|---|---|---|---|---|---|---|---|
| □□□□ `devops-testplans` | □□ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| | □□ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| | □□ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| | □□ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| □□□□ `devops-testmodules` | □□ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| | □□ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| | □□ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |

| □□ | □□ | □□□ □□ | □□□ □□□ | □□□ □□ | □□□□ □□□ | □□ □□ | □□□ □□ |
|----|----|---------|----------|---------|-----------|-------|---------|
| | □□ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |

| | □□ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |

# □□□□

## □ Markdown □□□□

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

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

## □ MDX □□□□

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

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

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

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

---

# □□

---

# □□□□□

```
reference:
  - repo: alauda-public/product-doc-guide # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
    branch: # [string] □□□□□□□□□□□□
    publicBase: # [string] □□□□□□□□□□□□□□□□□□ /images/xx.png □□□□□□□□□□□□□□□□□□ do
    sources:
      - name: anchor # □□□□□□□□□□□□□□□□□□□□□□□□
        path: docs/index.mdx#□□ # □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□ doom.confi
        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

```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

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

## versions.yaml

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