

Cursor Rules 使用指南

📅 2025年1月1日 ⌚ 1 分钟阅读

#Cursor

#AI

#编程工具

#规则配置

Cursor 的 `.cursor/rules` 使用指南：单一规则文件 vs 目录结构的用法和最佳实践

Cursor 的 `.cursor/rules` 既可以是**单一规则文件**，也可以是**目录结构**，支持更细粒度的规则管理。

但实际用法取决于你用的 Cursor 版本和你的项目需求：

1. 两种常见用法

A. 单一规则文件（最常见，兼容性最好）

`.cursor/rules` 是一个 YAML 或 JSON 文件，里面写所有规则。

适合小型项目或规则不多的场景。

你现在的做法就是这种，**完全没问题**，大多数项目都这样用。

B. 目录结构（高级用法，适合大型/多模块项目）

`.cursor/rules/` 是一个目录，里面可以有多个规则文件（如 `markdown.yml`、`ci.yml`），甚至可以有子目录（如 `.cursor/rules/content/`）。

适合规则非常多、需要分模块管理的场景。

这种方式需要 Cursor 编辑器/工具链支持自动递归读取目录下所有规则文件。

目录

文章信息

字数

阅读时间

发布时间

更新时间

标签

#Cursor

#AI

#编程工具

2. 你该怎么选？

如果你的规则不多，直接用 `.cursor/rules` 文件即可（当前做法，简单高效）。

如果你想分模块管理规则，可以把 `.cursor/rules` 变成目录，把每类规则写到不同文件里，比如：

```
.cursor/rules/  
markdown.yml  
ci.yml  
theme.yml
```

每个文件内容类似于你现在的规则片段。

3. 如何切换？

如果你想用目录结构，需要先把当前的 `.cursor/rules` 文件删除，然后新建 `.cursor/rules/` 目录，再把规则拆分到不同文件里。

例如：

```
rm .cursor/rules  
mkdir .cursor/rules  
# 然后创建 .cursor/rules/markdown.yml 等文件
```

4. 总结

你现在的 `.cursor/rules` 文件用法是**完全正确且主流的**。

只有在规则非常多、需要分模块时，才建议用目录结构。

两种方式都被 Cursor 支持，**但不能同时存在**（即不能既有

`.cursor/rules` 文件又有 `.cursor/rules/` 目录）。

自动生成规则文件

方法一：让cursor生成一个规则文件。在cursor的对话框中，让cursor根据和cursor聊天的历史记录，生成一个规则文件。或者直接输入"/Generate Cursor Rules"，让cursor根据你的需求，生成一个规则文件。

方法二：使用在线工具"<https://cursorsrules.agnt.one/chat>"，通过聊天的方式，提出需求，让网站根据你的需求，生成一个规则文件。

参考

<https://github.com/sanjeed5/awesome-cursor-rules-mdc/blob/main/rules-mdc/ant-design.mdc>

分享
这篇文章



相关文章推荐

欢迎来到我的AI博客

这是一个专注于AI领域的技术...

Pangu Deep Dive...

Pangu 相关论文的深度解析和...

Stephen Wolfram...

Stephen Wolfram
的生平、贡献...