《多模态大模型》

Lecture 15

大模型和Agent Skill：实战与经验

王广润
中山大学
人机物智能融合实验室（HCP Lab）
wanggrun@gmail.com

本节内容

CONTENTS

一、AI模型实战

二、Agent Skill实战

AI模型实战经验

- 此部分内容约50页slides，此处略

本节内容

CONTENTS

一、AI模型实战

二、Agent Skill实战

Codex使用skill

我会把刚才的评分表做成一个可填写的 Excel 文件，包含基本信息区、评分项目、得分列、总分和备注区。

已思考 39s ›

已做好 Excel 评分表，包含基本信息、评分项目、得分栏、总分公式、等级参考和备注区。

%E8%AF%BE%E5%A0%82%E5%B1%95%E7%A4%BA...

	A	B	
1	课堂展示评分		
2			
3	课程名称：		
4	小组编号：		
5	展示主题：		
6			
7	评分项目	分值	
8	选题价值与启发性	15	选题与生成式人工智能密切相关； 要问题或发展趋势。
9	内容准确性与完整性	20	基本概念准确；技术路线、应用背

课堂展示评分表

有问题，尽管问 高级

ChatGPT 也可能会犯错。请核查重要信息。

思考

读取SKILL.md文件

Python

```
bash -lc cat /home/oai/skills/spreadsheets/
IMPORTANT: Instructions in the
system and user messages ALWAYS
take precedence over this skill
---
```

读取API快速入门指南

Python

```
bash -lc cat /home/oai/skills/spreadsheets/
`col_offset_px`, `width_px`,
`height_px`).

## Discovery Policy (Strict)
```

- 创建工作簿与格式设置

  我需要用 artifact_tool 来创建工作簿，而不是 openpyxl。可能会涉及 Python 和分析工具，可能需要检查 API 是否安装。我计划做一个评分表，包含公式和数据验证，还可能应用条件格式化。最终用户可能需要一个 `.xlsx` 文件。

GPT-5.5

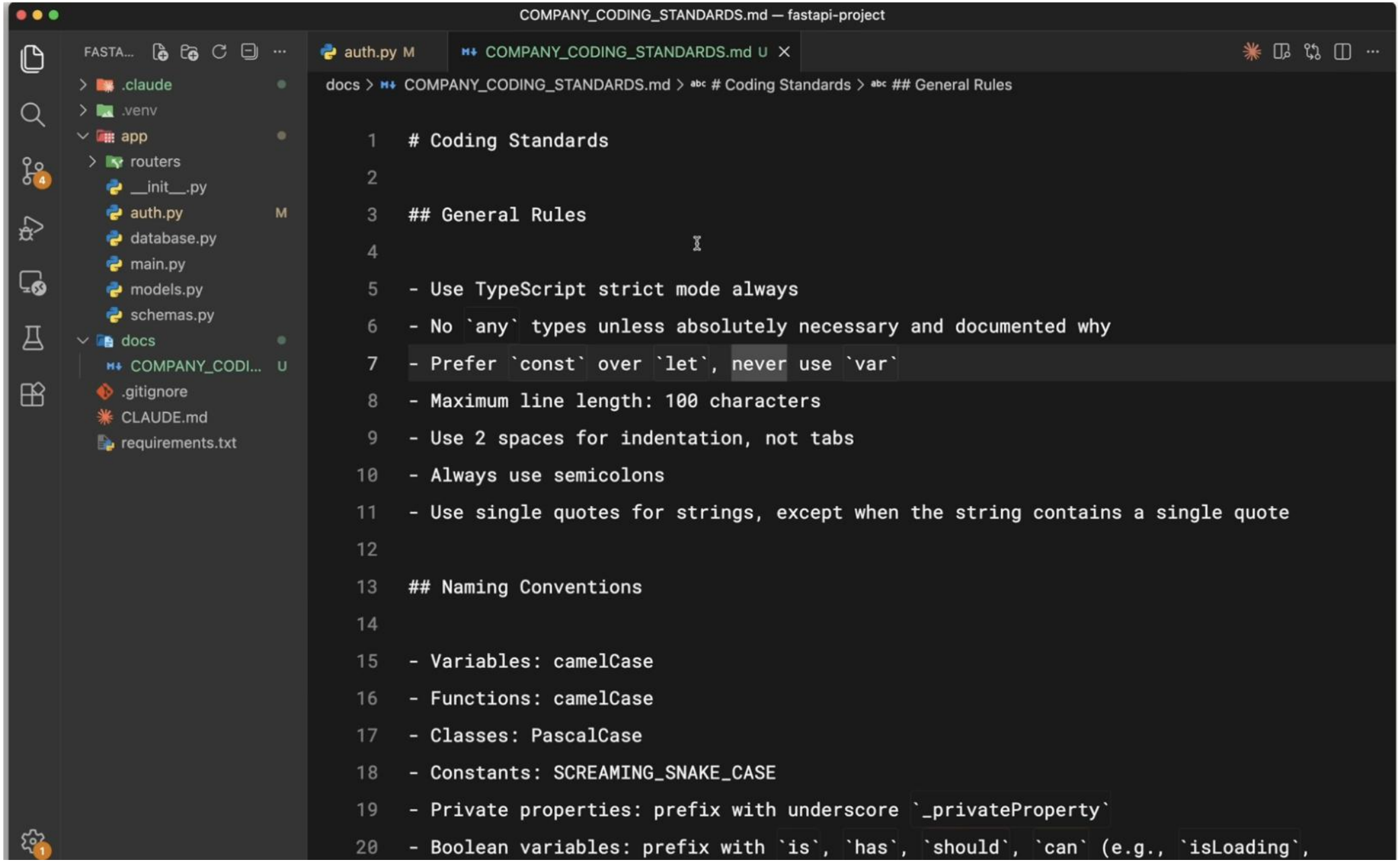
- 因此，严格地说，不应该说：

- > GPT-5.5 在某任务上成功率是多少。

- 而应该说：

- > GPT-5.5 + Skill X + Codex harness + 某工具环境 + 某 budget 的成功率是多少。

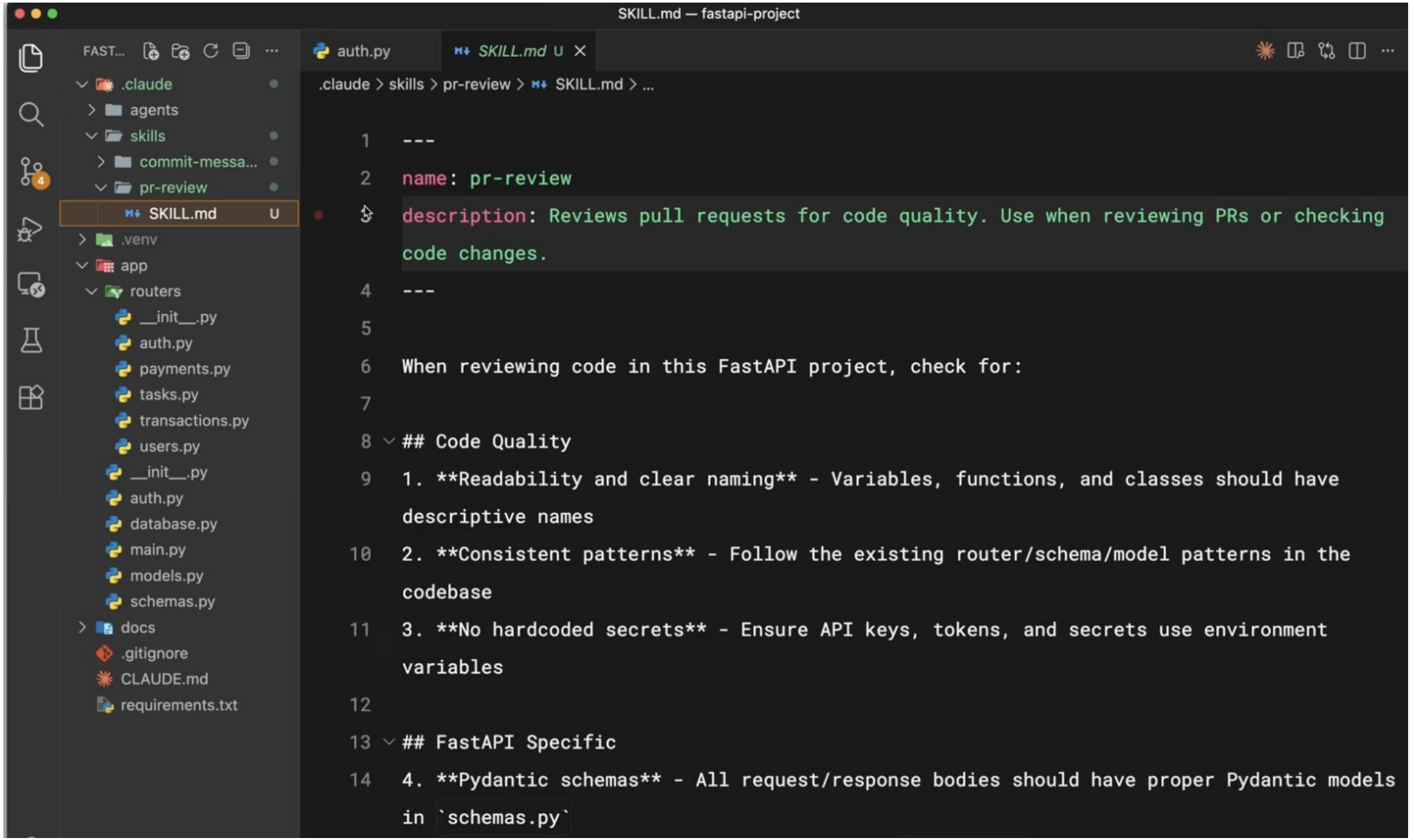
如果不使用skill，你要这么做



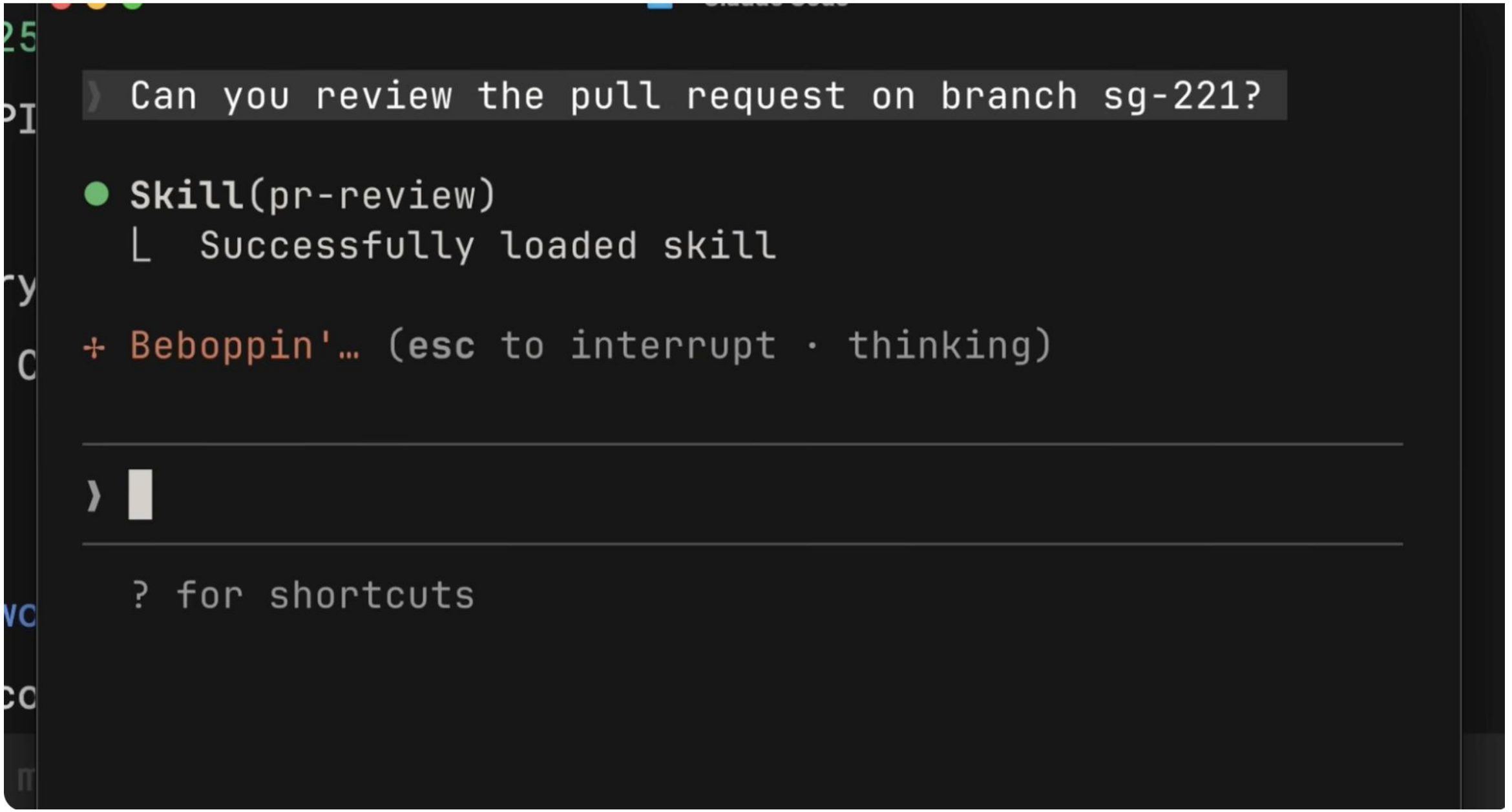
一个字一个字地敲

```
ranch. Remember, no emojis or verbose
```

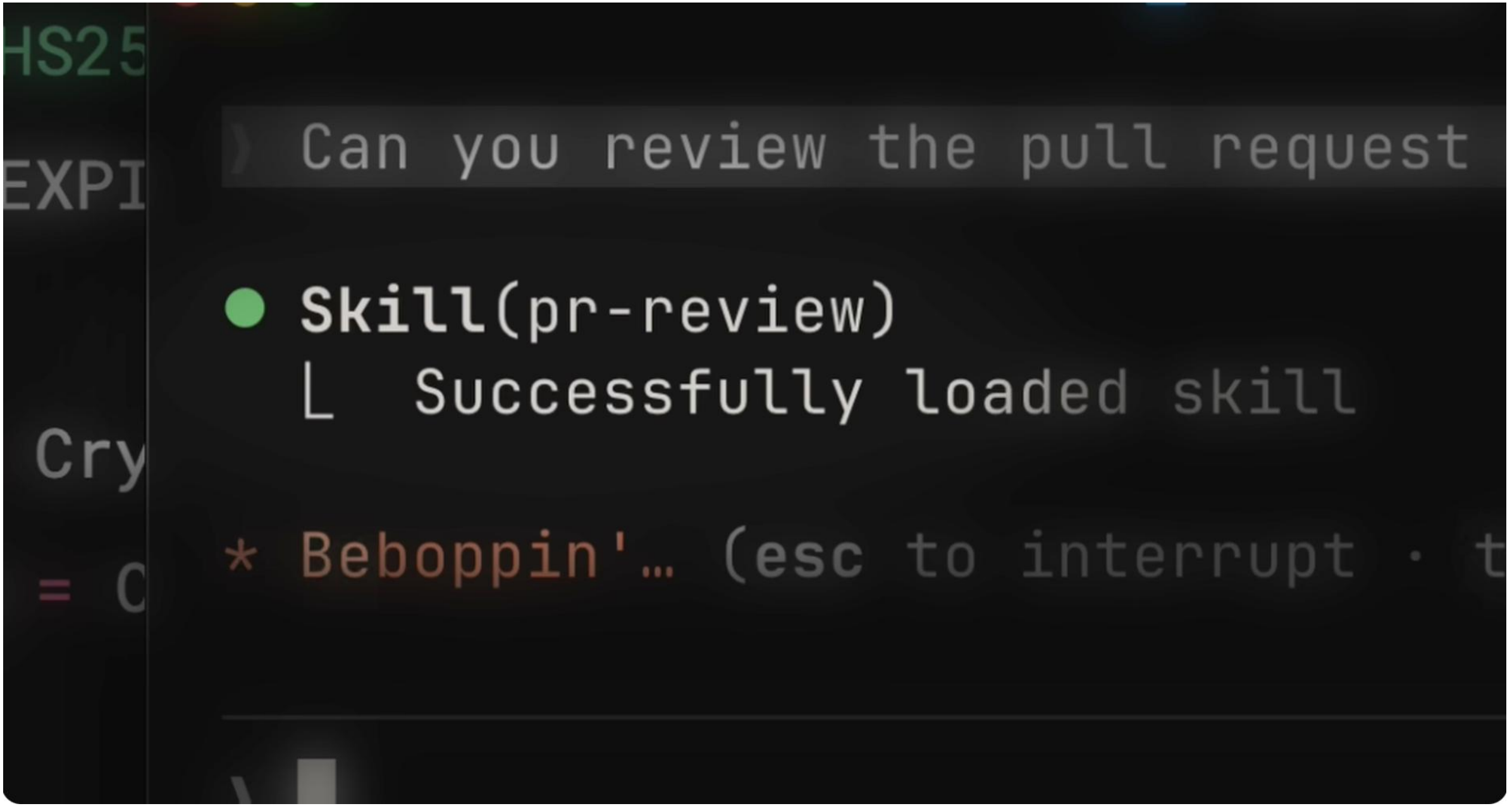
Skill的位置和内容



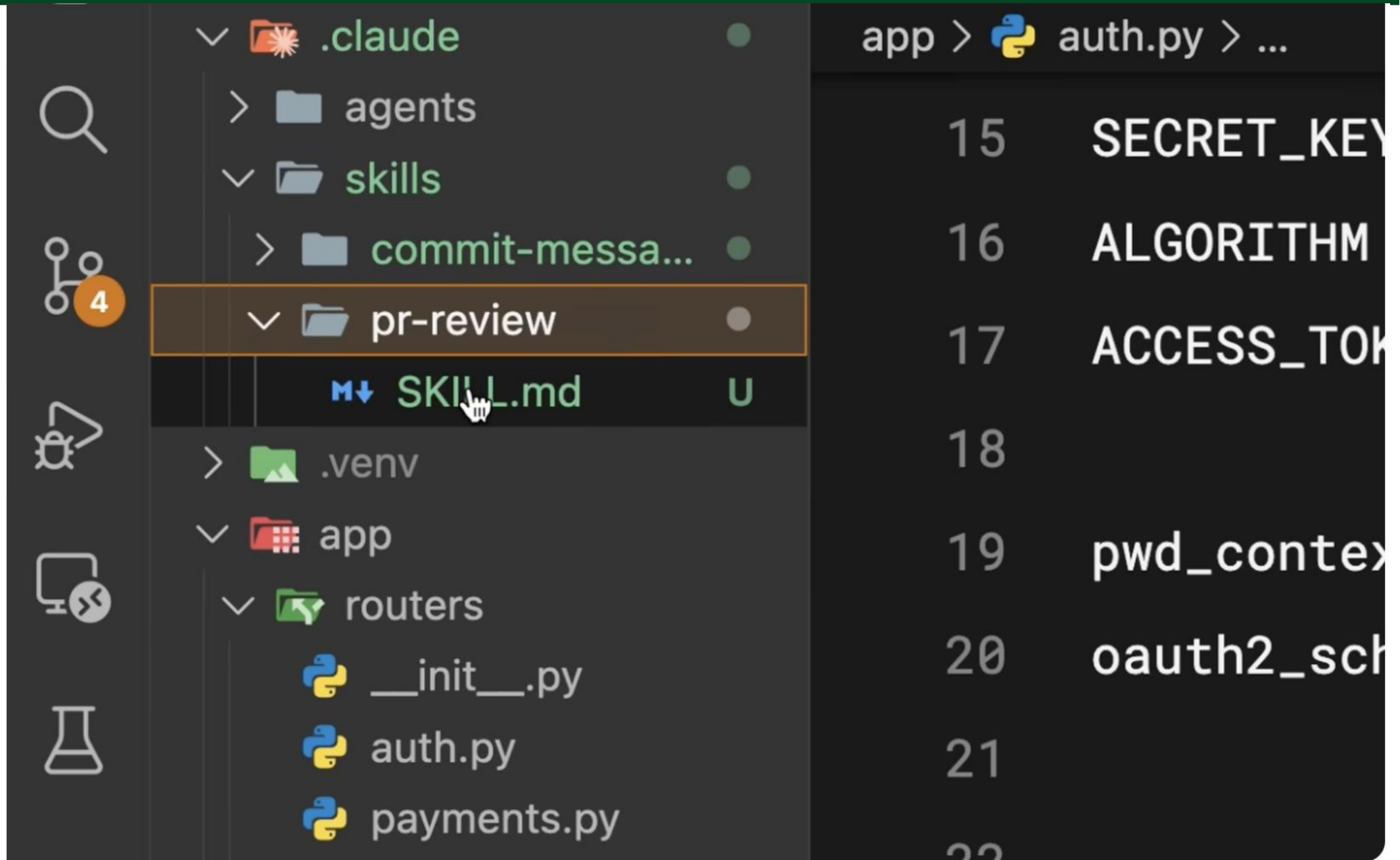
大模型加载skill


Can you review the pull request on branch sg-221?
Skill(pr-review)
Successfully loaded skill
Beboppin'… (esc to interrupt · thinking)
? for shortcuts

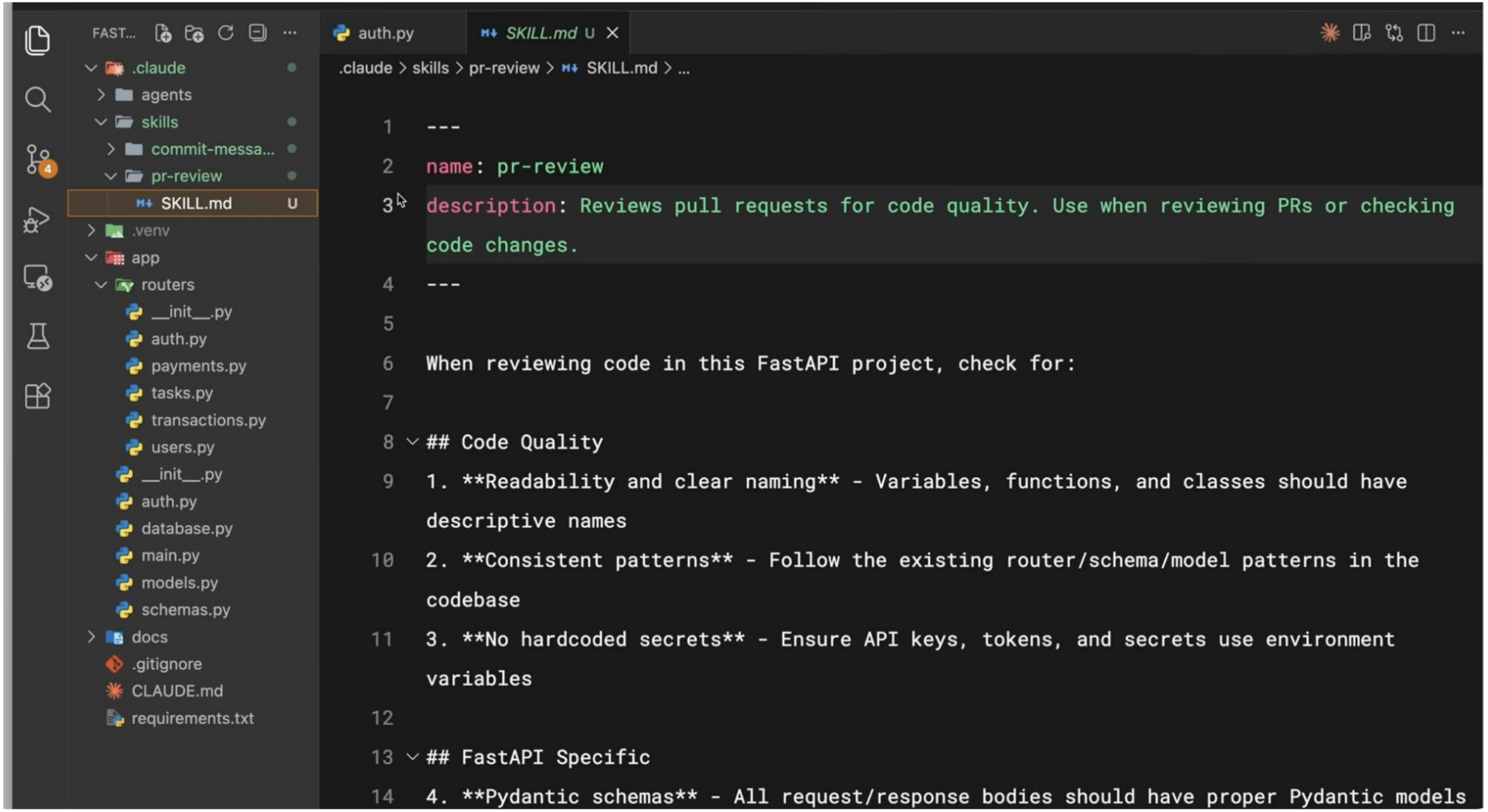
加载成功


Can you review the pull request
Skill(pr-review)
Successfully loaded skill
* Beboppin'… (esc to interrupt · t

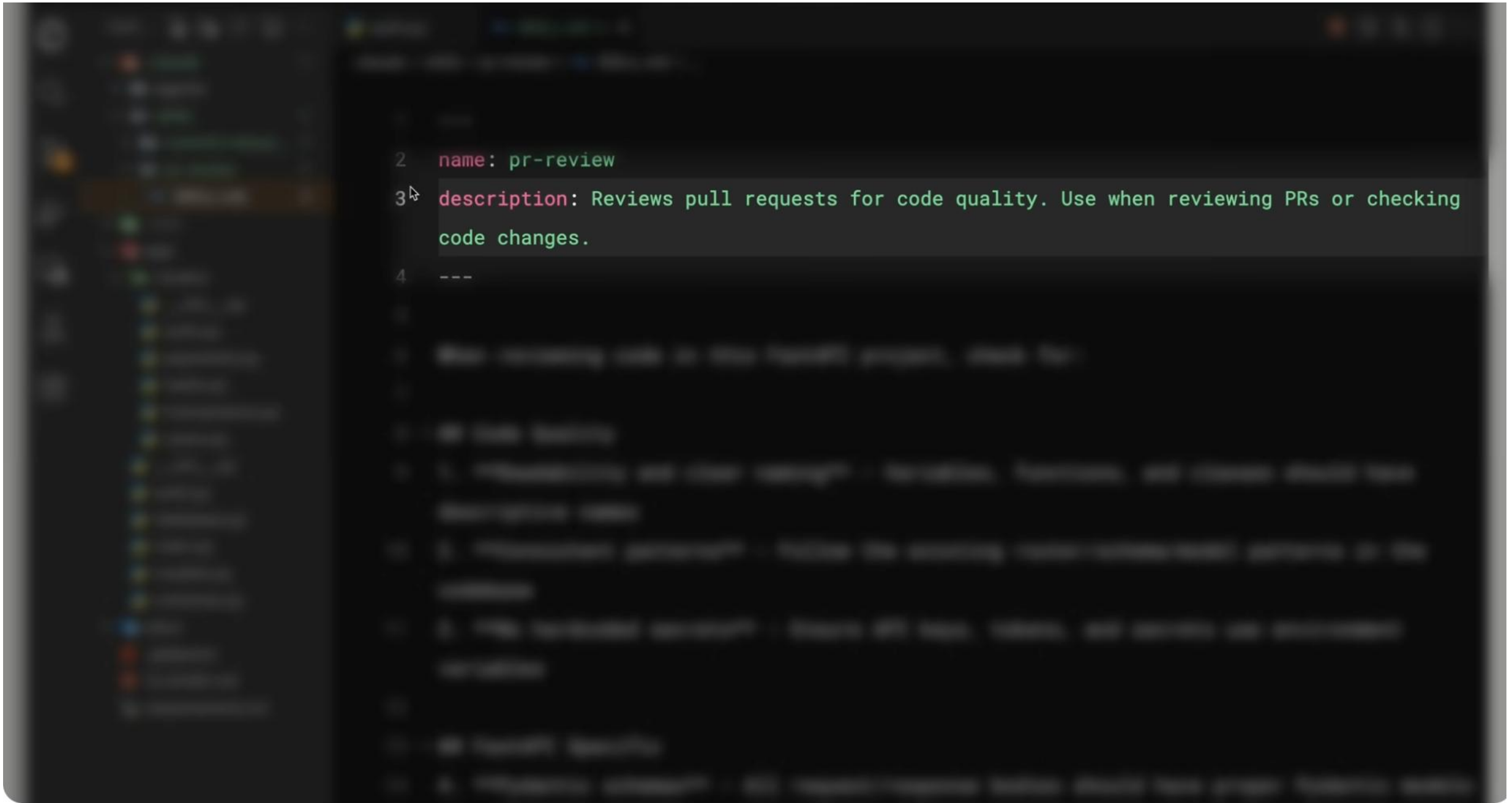
Skill的目录


.claude
agents
skills
commit-messa...
pr-review
SKILL.md
U
.venv
app
routers
__init__.py
auth.py
payments.py
app > auth.py > ...
15 SECRET_KEY
16 ALGORITHM
17 ACCESS_TOK
18
19 pwd_contex
20 oauth2_sch
21

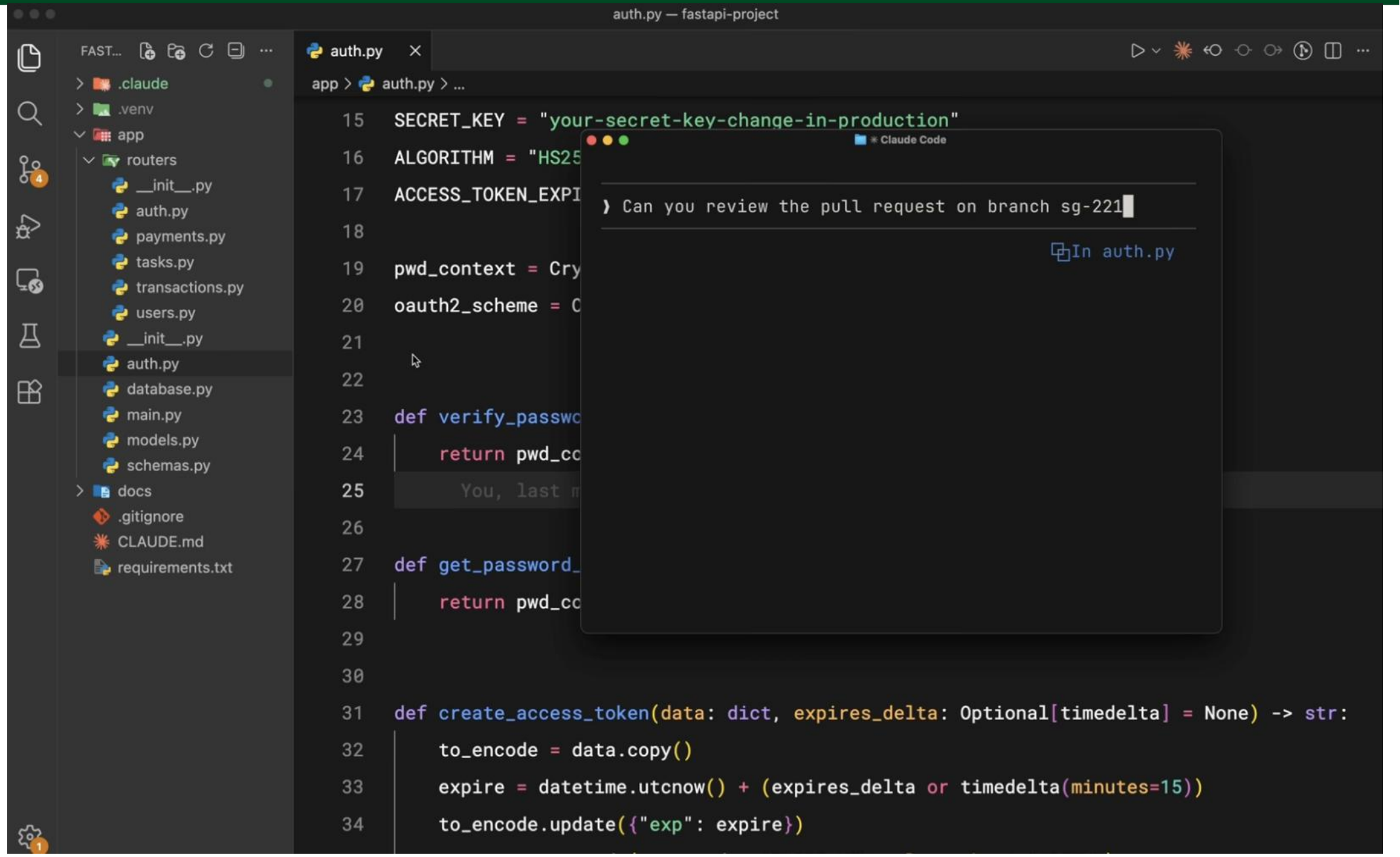
Skill的内容


FAST...
.claude
agents
skills
commit-messa...
pr-review
SKILL.md U
.venv
app
routers
__init__.py
auth.py
payments.py
tasks.py
transactions.py
users.py
__init__.py
auth.py
database.py
main.py
models.py
schemas.py
docs
.gitignore
CLAUDE.md
requirements.txt
auth.py
SKILL.md U
.claude > skills > pr-review > SKILL.md > ...
1 ---
2 name: pr-review
3 description: Reviews pull requests for code quality. Use when reviewing PRs or checking code changes.
4 ---
5
6 When reviewing code in this FastAPI project, check for:
7
8 ## Code Quality
9 1. **Readability and clear naming** - Variables, functions, and classes should have descriptive names
10 2. **Consistent patterns** - Follow the existing router/schema/model patterns in the codebase
11 3. **No hardcoded secrets** - Ensure API keys, tokens, and secrets use environment variables
12
13 ## FastAPI Specific
14 4. **Pydantic schemas** - All request/response bodies should have proper Pydantic models

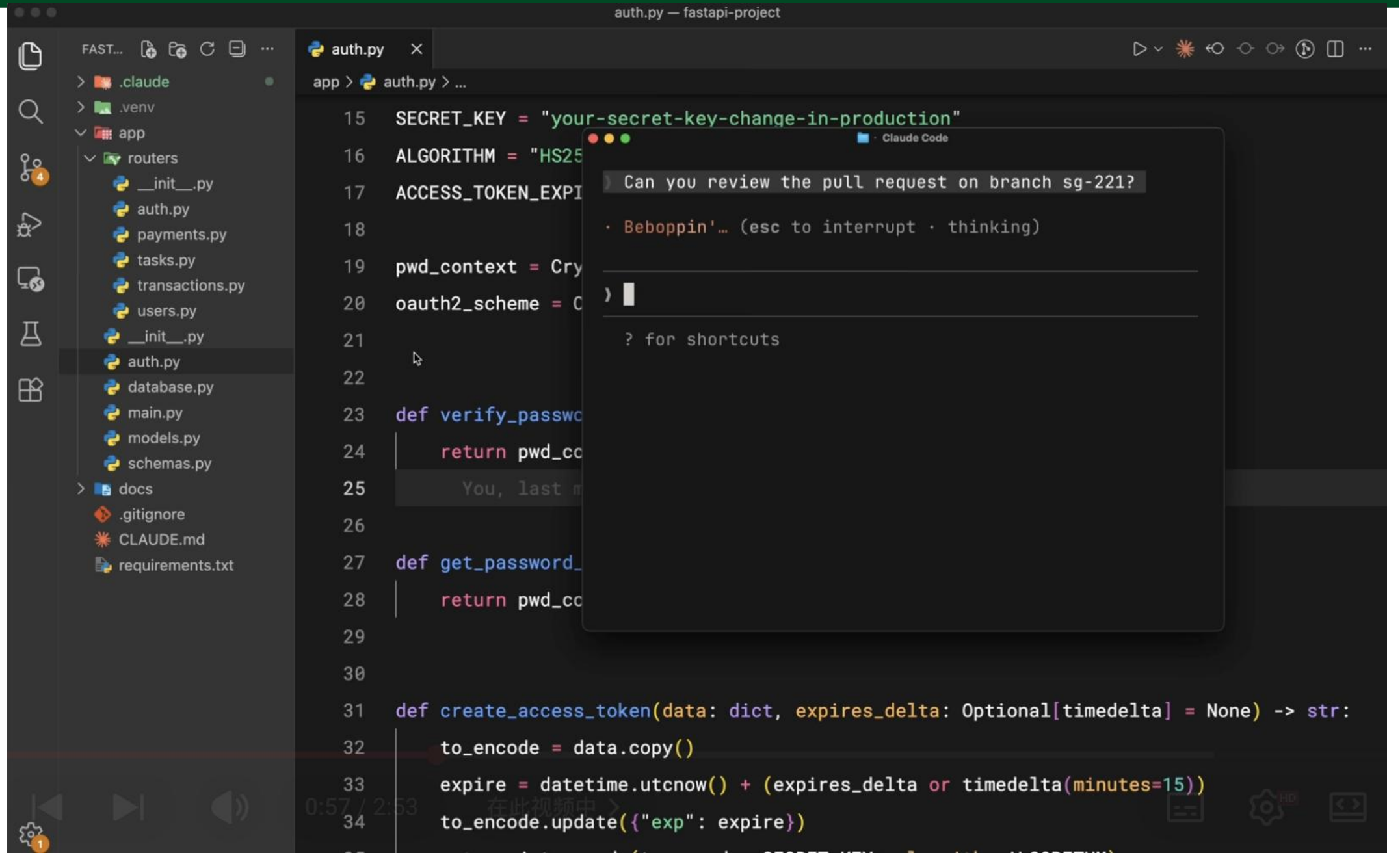
Skill要有一个好的描述



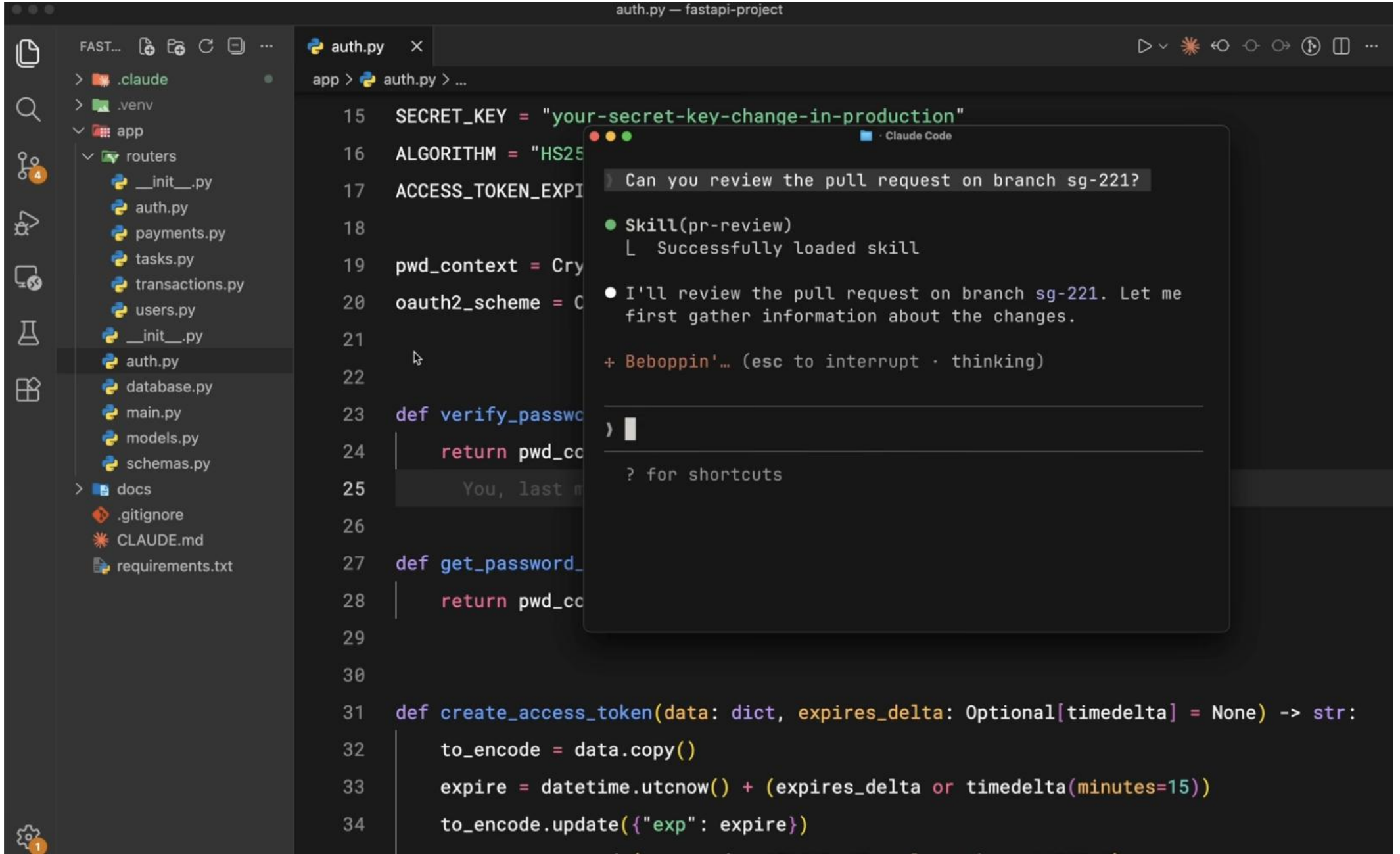
启动Skill


auth.py — fastapi-project
FAST...
.claude
.venv
app
routers
__init__.py
auth.py
payments.py
tasks.py
transactions.py
users.py
__init__.py
auth.py
database.py
main.py
models.py
schemas.py
docs
.gitignore
CLAUDE.md
requirements.txt
auth.py
app > auth.py > ...
15 SECRET_KEY = "your-secret-key-change-in-production"
16 ALGORITHM = "HS25
17 ACCESS_TOKEN_EXPI
18
19 pwd_context = Cry
20 oauth2_scheme = C
21
22
23 def verify_passwo
24 return pwd_cc
25 You, last m
26
27 def get_password_
28 return pwd_cc
29
30
31 def create_access_token(data: dict, expires_delta: Optional[timedelta] = None) -> str:
32 to_encode = data.copy()
33 expire = datetime.utcnow() + (expires_delta or timedelta(minutes=15))
34 to_encode.update({"exp": expire})
Claude Code
❯ Can you review the pull request on branch sg-221
In auth.py

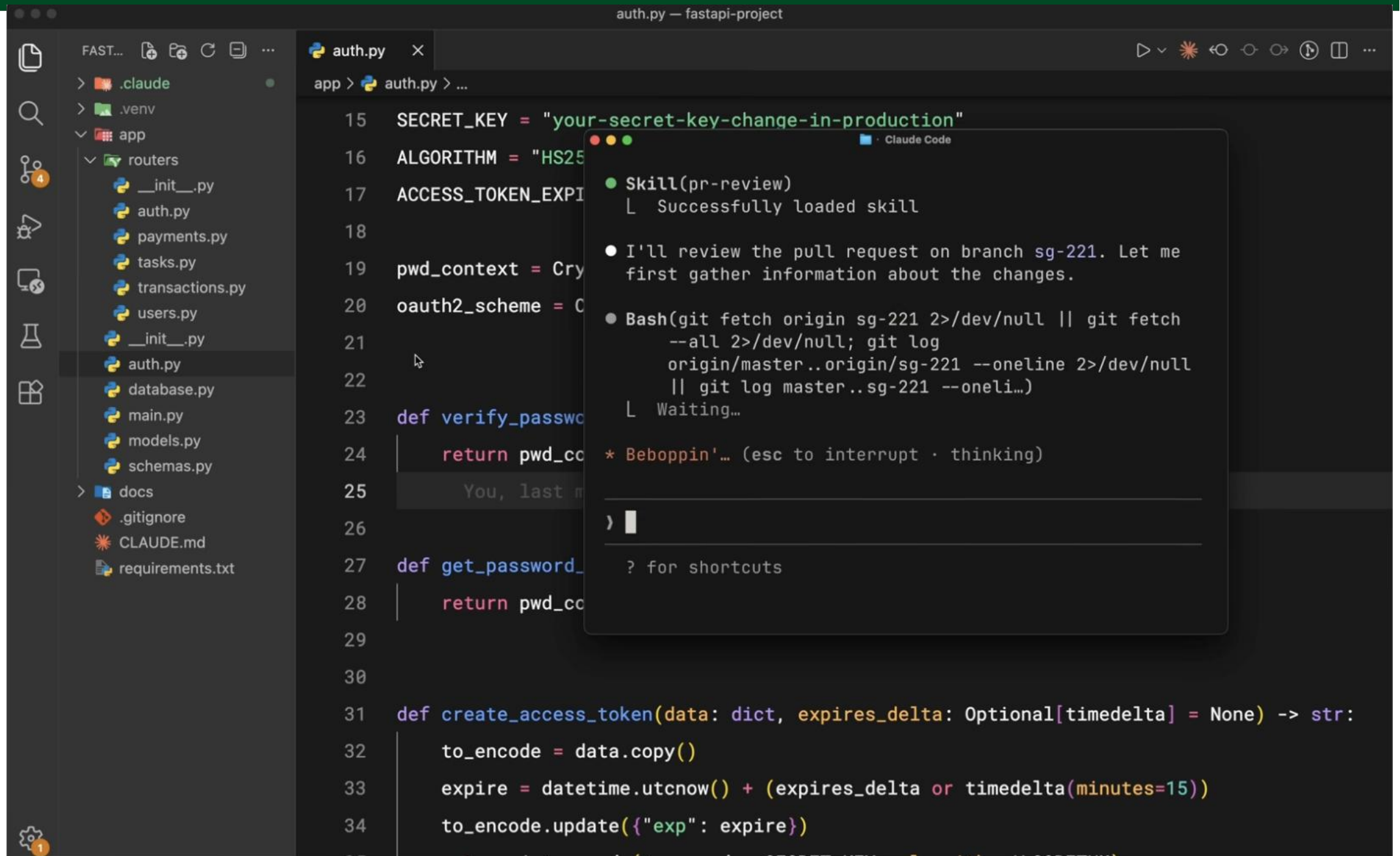
响应中


auth.py — fastapi-project
FAST...
.claude
.venv
app
routers
__init__.py
auth.py
payments.py
tasks.py
transactions.py
users.py
__init__.py
auth.py
database.py
main.py
models.py
schemas.py
docs
.gitignore
CLAUDE.md
requirements.txt
auth.py
app > auth.py > ...
SECRET_KEY = "your-secret-key-change-in-production"
ALGORITHM = "HS25
ACCESS_TOKEN_EXPI
pwd_context = Cry
oauth2_scheme = O
def verify_passwo
return pwd_co
You, last m
def get_password_
return pwd_co
def create_access_token(data: dict, expires_delta: Optional[timedelta] = None) -> str:
to_encode = data.copy()
expire = datetime.utcnow() + (expires_delta or timedelta(minutes=15))
to_encode.update({"exp": expire})
Claude Code
Can you review the pull request on branch sg-221?
· Beboppin'… (esc to interrupt · thinking)
? for shortcuts
0:57 / 2:53
在此视频中

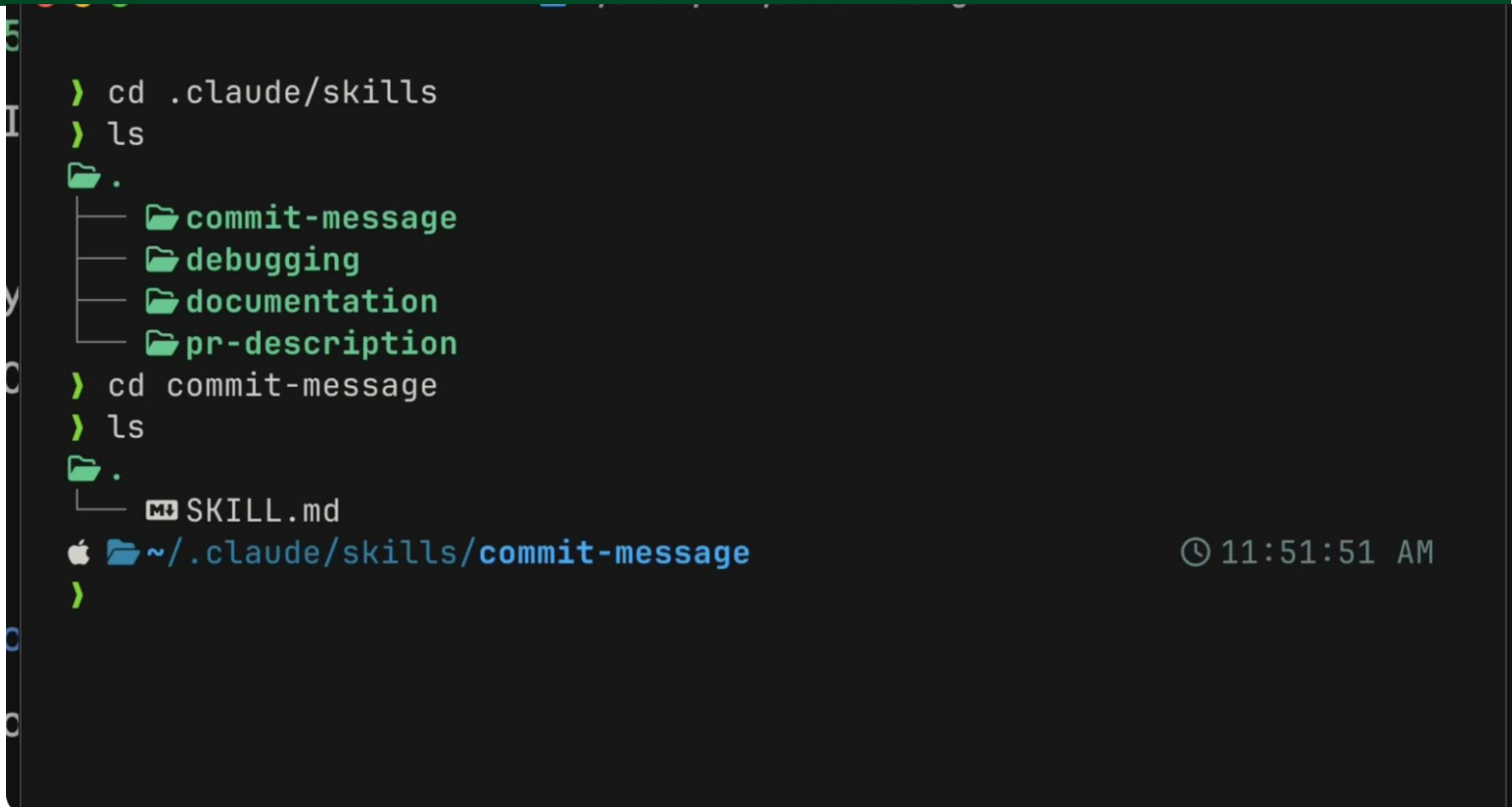
成功加载


auth.py — fastapi-project
FAST...
.claude
.venv
app
routers
__init__.py
auth.py
payments.py
tasks.py
transactions.py
users.py
__init__.py
auth.py
database.py
main.py
models.py
schemas.py
docs
.gitignore
CLAUDE.md
requirements.txt
auth.py
app > auth.py > ...
15 SECRET_KEY = "your-secret-key-change-in-production"
16 ALGORITHM = "HS25
17 ACCESS_TOKEN_EXPI
18
19 pwd_context = Cry
20 oauth2_scheme = C
21
22
23 def verify_passwo
24 return pwd_co
25 You, last m
26
27 def get_password_
28 return pwd_co
29
30
31 def create_access_token(data: dict, expires_delta: Optional[timedelta] = None) -> str:
32 to_encode = data.copy()
33 expire = datetime.utcnow() + (expires_delta or timedelta(minutes=15))
34 to_encode.update({"exp": expire})
Claude Code
Can you review the pull request on branch sg-221?
Skill(pr-review)
Successfully loaded skill
I'll review the pull request on branch sg-221. Let me first gather information about the changes.
+ Beboppin'… (esc to interrupt · thinking)
? for shortcuts

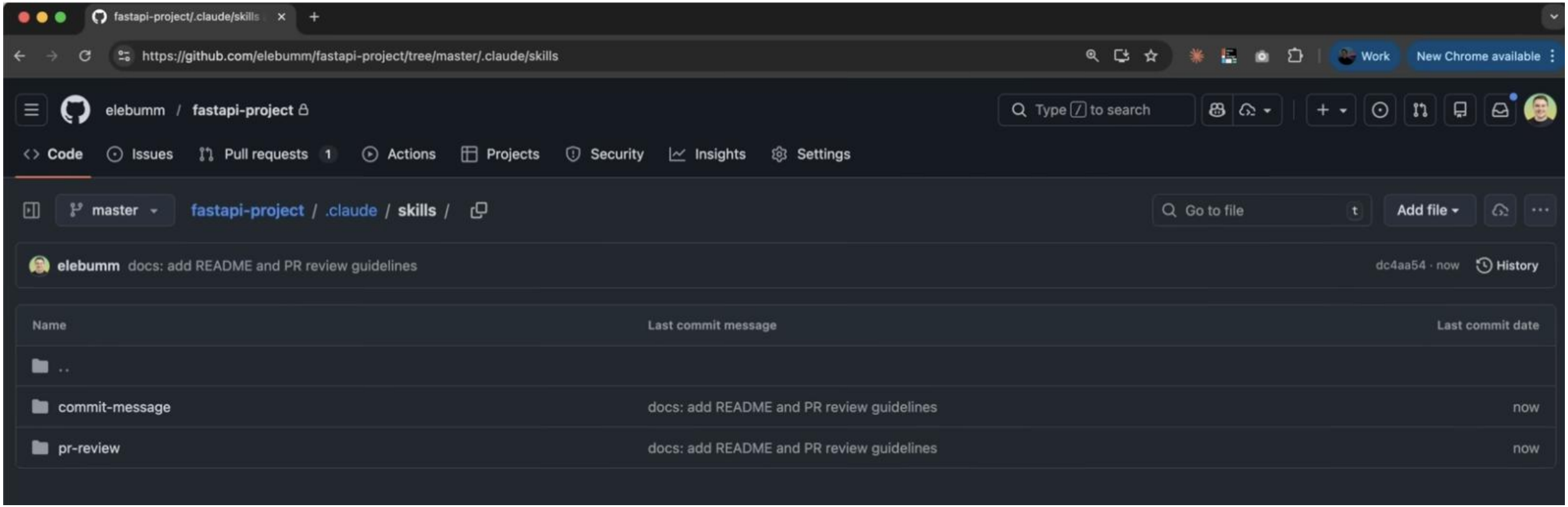
执行Skill


auth.py — fastapi-project
FAST...
.claude
.venv
app
routers
__init__.py
auth.py
payments.py
tasks.py
transactions.py
users.py
__init__.py
auth.py
database.py
main.py
models.py
schemas.py
docs
.gitignore
CLAUDE.md
requirements.txt
auth.py
app > auth.py > ...
15 SECRET_KEY = "your-secret-key-change-in-production"
16 ALGORITHM = "HS25
17 ACCESS_TOKEN_EXPI
18
19 pwd_context = Cry
20 oauth2_scheme = O
21
22
23 def verify_passwo
24 return pwd_co
25 You, last m
26
27 def get_password_
28 return pwd_co
29
30
31 def create_access_token(data: dict, expires_delta: Optional[timedelta] = None) -> str:
32 to_encode = data.copy()
33 expire = datetime.utcnow() + (expires_delta or timedelta(minutes=15))
34 to_encode.update({"exp": expire})
Claude Code
Skill(pr-review)
Successfully loaded skill
I'll review the pull request on branch sg-221. Let me first gather information about the changes.
Bash(git fetch origin sg-221 2>/dev/null || git fetch --all 2>/dev/null; git log origin/master..origin/sg-221 --oneline 2>/dev/null || git log master..sg-221 --oneli…)
Waiting…
* Beboppin'… (esc to interrupt · thinking)
? for shortcuts

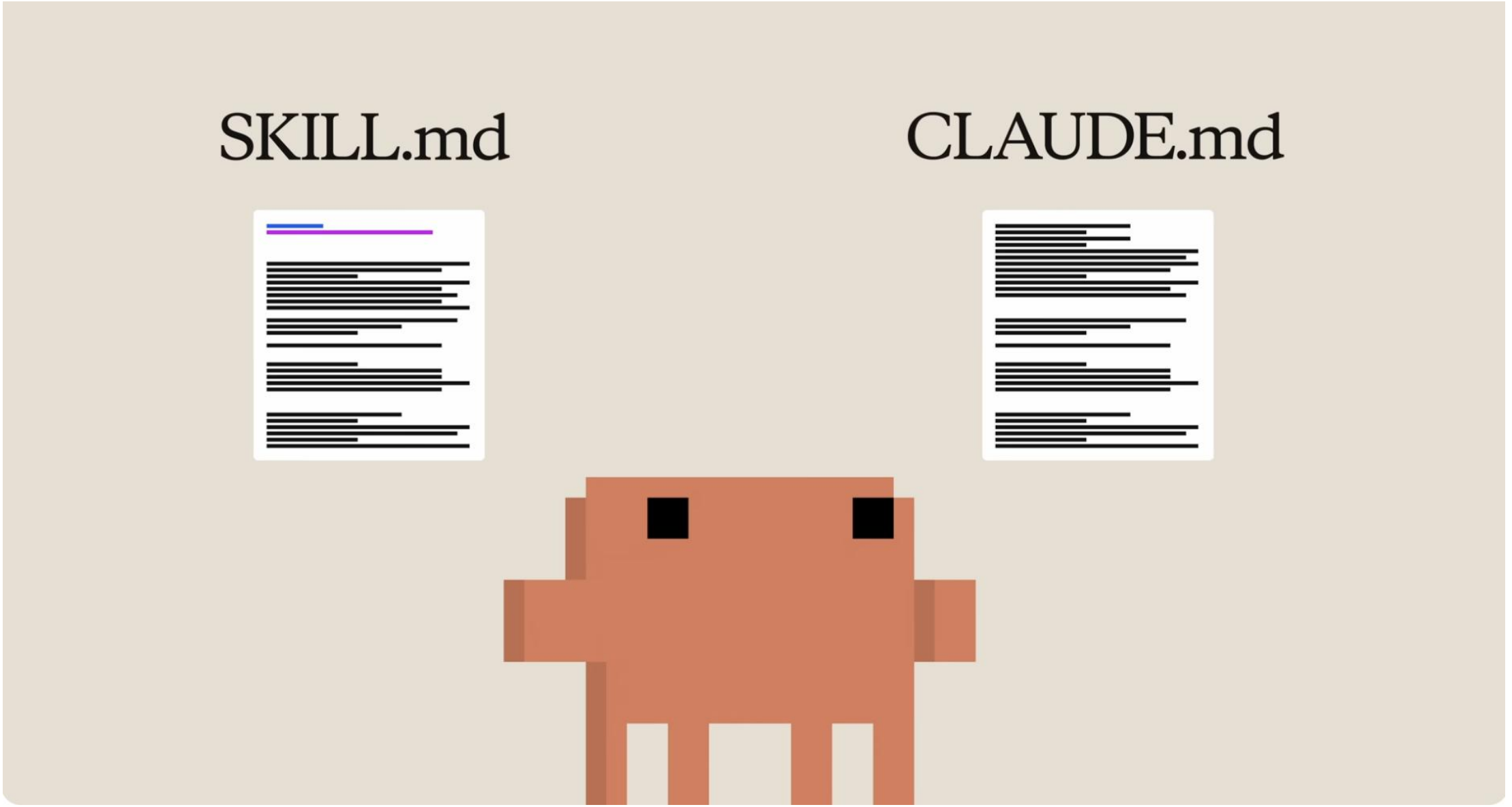
Skill的位置


cd .claude/skills
ls
.
commit-message
debugging
documentation
pr-description
cd commit-message
ls
.
SKILL.md
~/.claude/skills/commit-message
11:51:51 AM

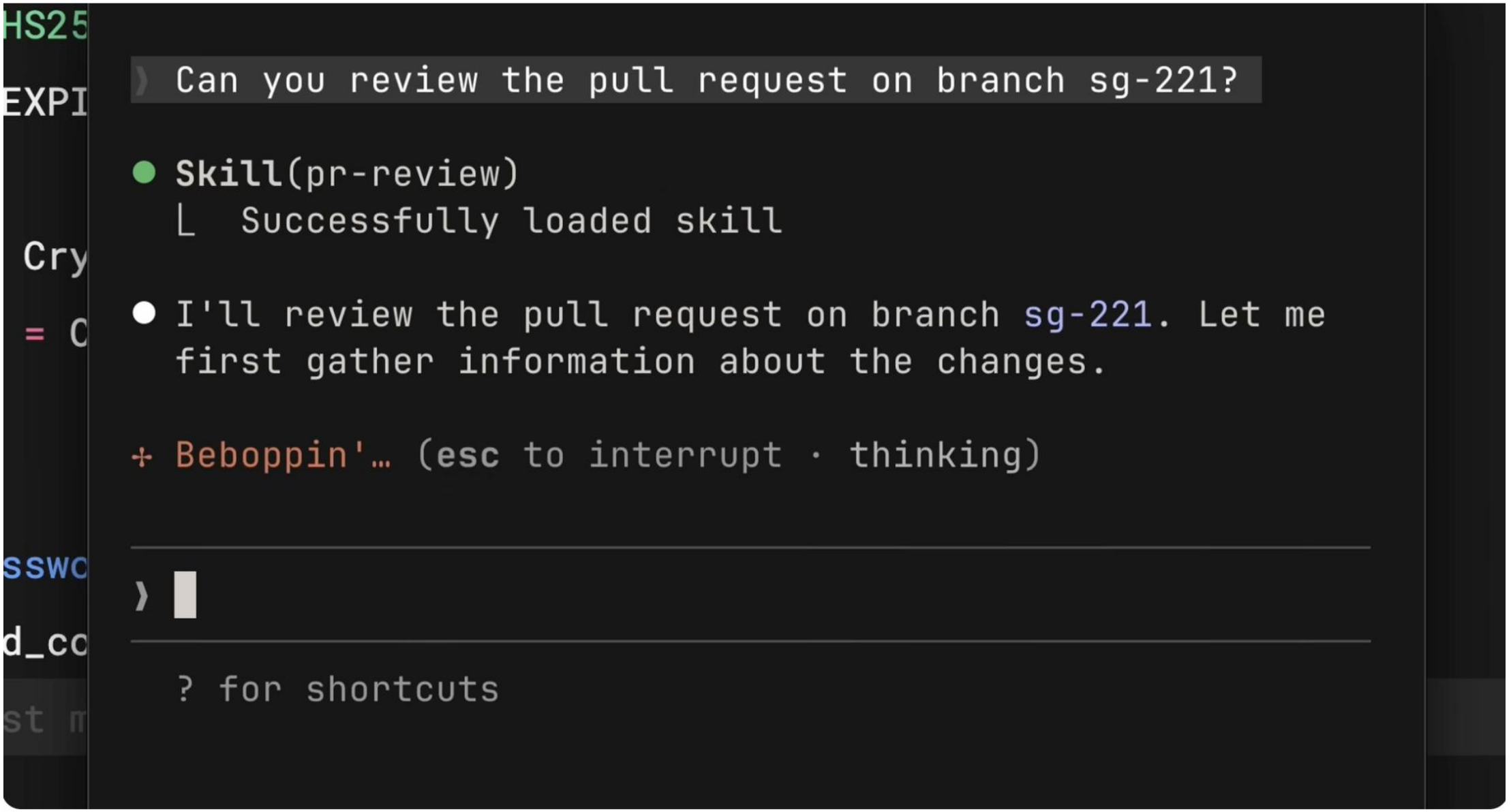
Github


fastapi-project/.claude/skills
https://github.com/elebumm/fastapi-project/tree/master/.claude/skills
Work
New Chrome available
elebumm / fastapi-project
Type / to search
Code
Issues
Pull requests 1
Actions
Projects
Security
Insights
Settings
master
fastapi-project / .claude / skills /
Go to file t
Add file
elebumm docs: add README and PR review guidelines
dc4aa54 · now
History
Name
Last commit message
Last commit date
..
commit-message
docs: add README and PR review guidelines
now
pr-review
docs: add README and PR review guidelines
now

Skill.md vs Claude.md



加载


HS25
EXPI
Cry
= C
sswo
d_cc
st m
❯ Can you review the pull request on branch sg-221?
● Skill(pr-review)
└ Successfully loaded skill
● I'll review the pull request on branch sg-221. Let me first gather information about the changes.
✢ Beboppin'… (esc to interrupt · thinking)
❯
? for shortcuts

Skill的内容

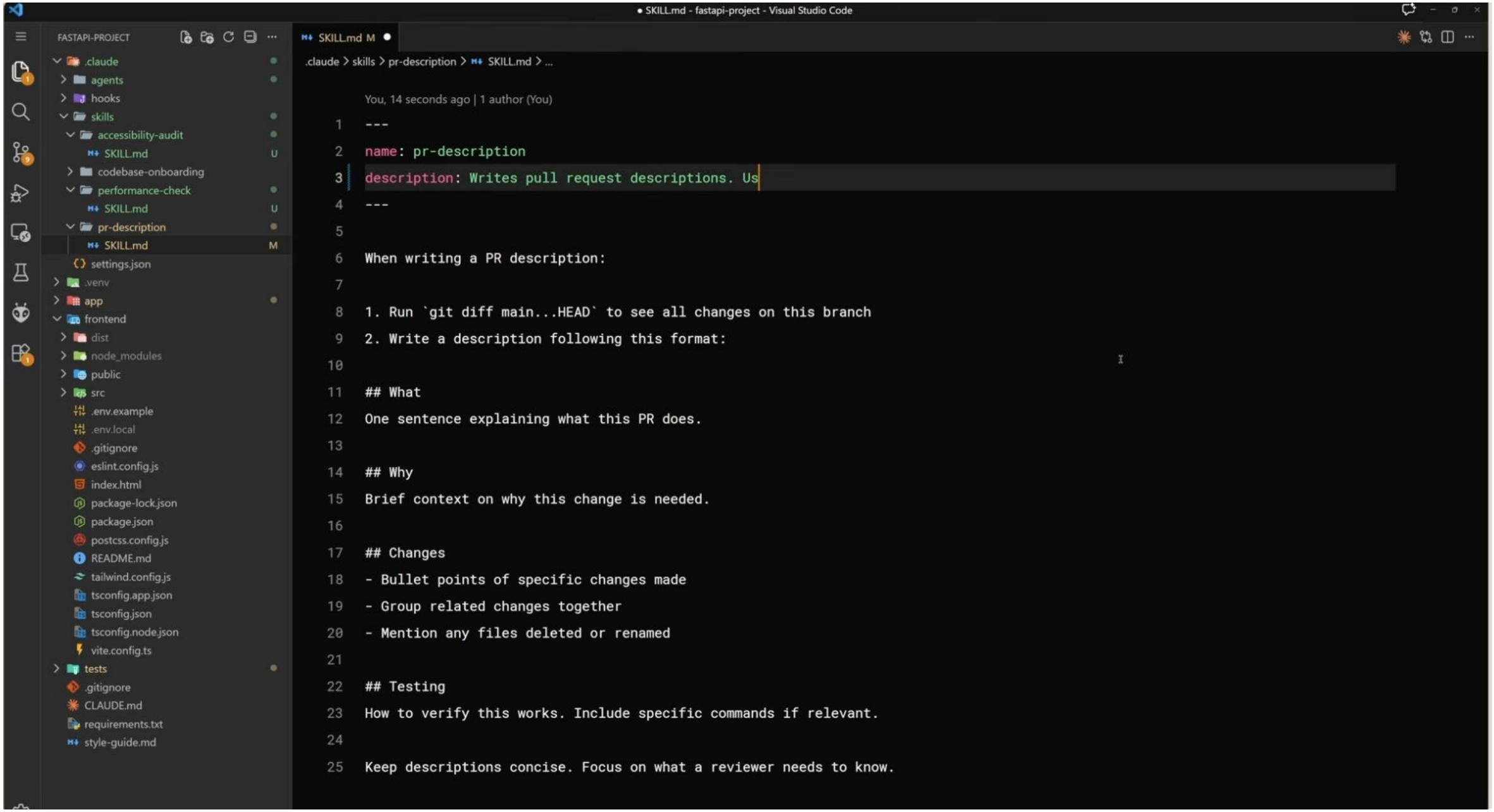
```
---
name: commit-message
description: Formats commit messages following project conventions. Us
commits or writing commit messages.
---

Write commit messages following this format:

## Structure
```
<type>: <short description>

<optional body explaining why, not what>
```
```

Skill的描述


• SKILL.md - fastapi-project - Visual Studio Code
FASTAPI-PROJECT
.claude
agents
hooks
skills
accessibility-audit
SKILL.md
codebase-onboarding
performance-check
SKILL.md
pr-description
SKILL.md
settings.json
.venv
app
frontend
dist
node_modules
public
src
.env.example
.env.local
.gitignore
eslint.config.js
index.html
package-lock.json
package.json
postcss.config.js
README.md
tailwind.config.js
tsconfig.app.json
tsconfig.json
tsconfig.node.json
vite.config.ts
tests
.gitignore
CLAUDE.md
requirements.txt
style-guide.md
SKILL.md M
.claude > skills > pr-description > SKILL.md > ...
You, 14 seconds ago | 1 author (You)
---
name: pr-description
description: Writes pull request descriptions. Us

When writing a PR description:
1. Run `git diff main...HEAD` to see all changes on this branch
2. Write a description following this format:
What
One sentence explaining what this PR does.
Why
Brief context on why this change is needed.
Changes
- Bullet points of specific changes made
- Group related changes together
- Mention any files deleted or renamed
Testing
How to verify this works. Include specific commands if relevant.
Keep descriptions concise. Focus on what a reviewer needs to know.

Skill的描述

```
# Too similar - Claude can't distinguish
# Skill 1
description: Helps with data analysis


# Skill 2
description: Analyzes data and generates reports
```

Skill.md vs Claude.md

CLAUDE.md

Project wide standards that always apply

Constraints like "never modify the database schema"

Framework preferences and coding style

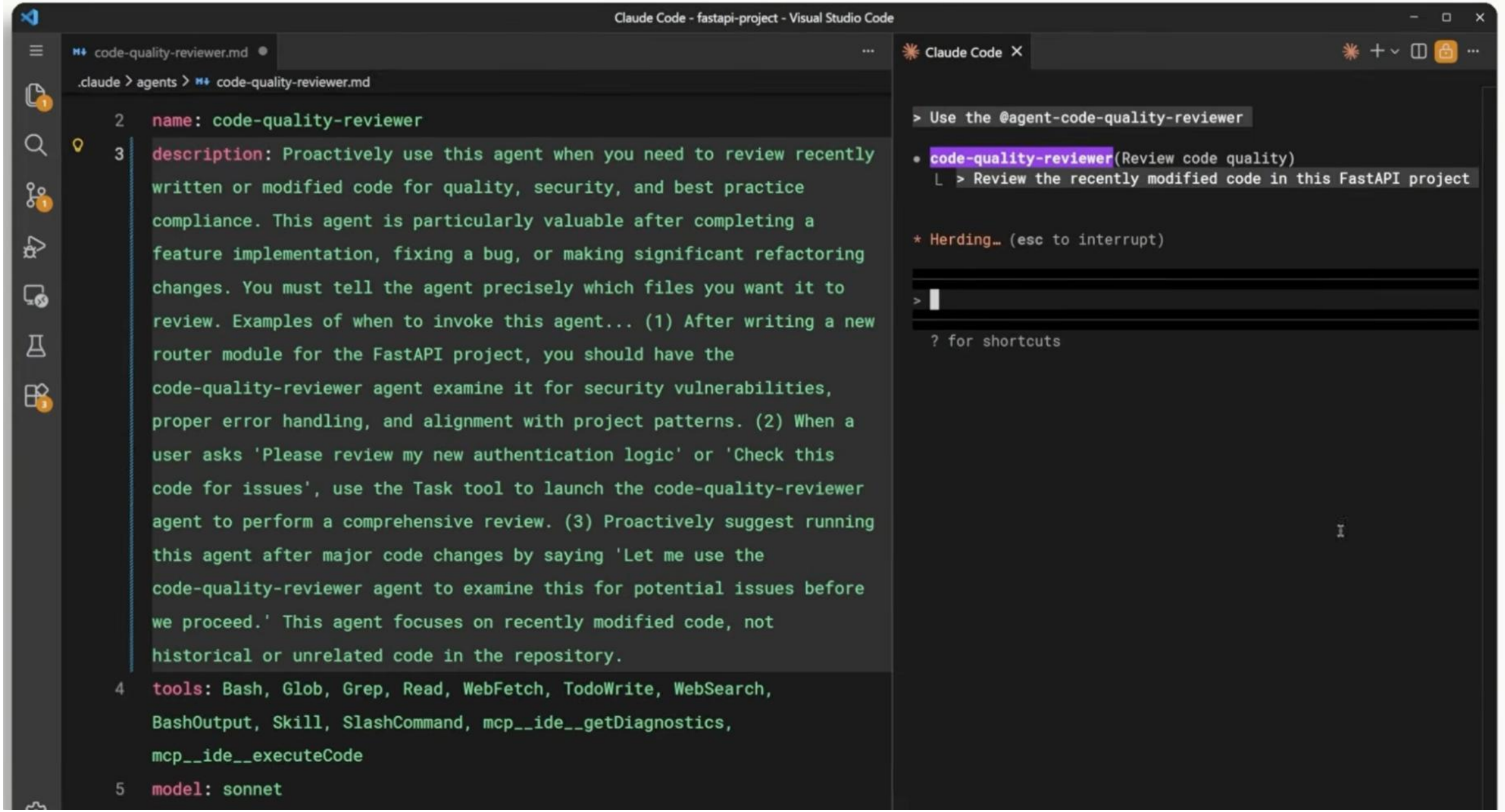
Skills

Task-specific expertise

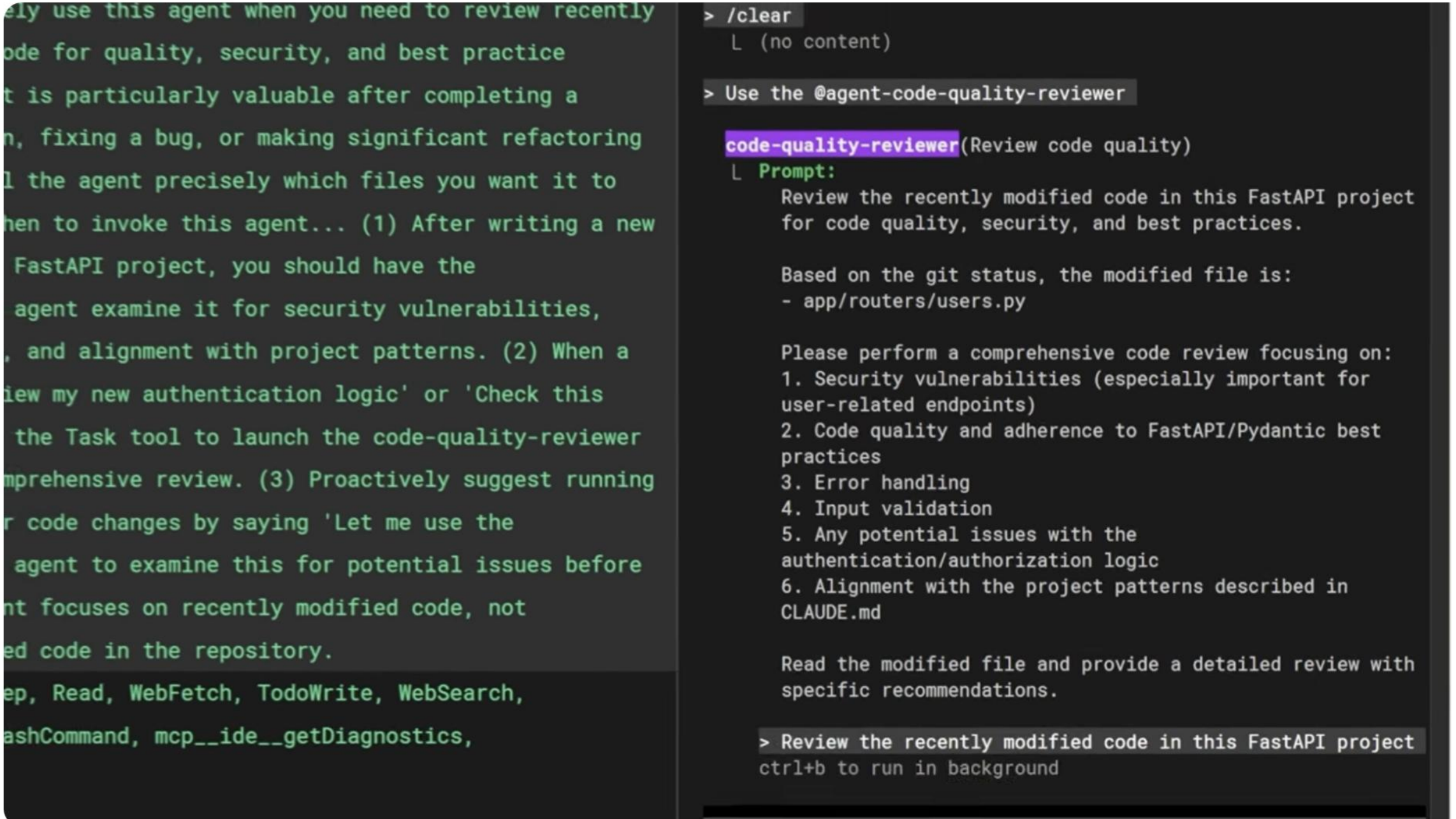
Knowledge that's only relevant sometimes

Detailed procedures that would clutter every conversation

sub agent


Claude Code - fastapi-project - Visual Studio Code
code-quality-reviewer.md
.claude > agents > code-quality-reviewer.md
2 name: code-quality-reviewer
3 description: Proactively use this agent when you need to review recently written or modified code for quality, security, and best practice compliance. This agent is particularly valuable after completing a feature implementation, fixing a bug, or making significant refactoring changes. You must tell the agent precisely which files you want it to review. Examples of when to invoke this agent... (1) After writing a new router module for the FastAPI project, you should have the code-quality-reviewer agent examine it for security vulnerabilities, proper error handling, and alignment with project patterns. (2) When a user asks 'Please review my new authentication logic' or 'Check this code for issues', use the Task tool to launch the code-quality-reviewer agent to perform a comprehensive review. (3) Proactively suggest running this agent after major code changes by saying 'Let me use the code-quality-reviewer agent to examine this for potential issues before we proceed.' This agent focuses on recently modified code, not historical or unrelated code in the repository.
4 tools: Bash, Glob, Grep, Read, WebFetch, TodoWrite, WebSearch, BashOutput, Skill, SlashCommand, mcp__ide__getDiagnostics, mcp__ide__executeCode
5 model: sonnet
Claude Code
> Use the @agent-code-quality-reviewer
• code-quality-reviewer(Review code quality)
L > Review the recently modified code in this FastAPI project
* Herding… (esc to interrupt)
>
? for shortcuts

sub agent


ely use this agent when you need to review recently
ode for quality, security, and best practice
t is particularly valuable after completing a
n, fixing a bug, or making significant refactoring
l the agent precisely which files you want it to
hen to invoke this agent... (1) After writing a new
FastAPI project, you should have the
agent examine it for security vulnerabilities,
, and alignment with project patterns. (2) When a
iew my new authentication logic' or 'Check this
the Task tool to launch the code-quality-reviewer
mprehensive review. (3) Proactively suggest running
r code changes by saying 'Let me use the
agent to examine this for potential issues before
nt focuses on recently modified code, not
ed code in the repository.
ep, Read, WebFetch, TodoWrite, WebSearch,
ashCommand, mcp__ide__getDiagnostics,
> /clear
└ (no content)
> Use the @agent-code-quality-reviewer
code-quality-reviewer(Review code quality)
└ Prompt:
Review the recently modified code in this FastAPI project
for code quality, security, and best practices.
Based on the git status, the modified file is:
- app/routers/users.py
Please perform a comprehensive code review focusing on:
1. Security vulnerabilities (especially important for
user-related endpoints)
2. Code quality and adherence to FastAPI/Pydantic best
practices
3. Error handling
4. Input validation
5. Any potential issues with the
authentication/authorization logic
6. Alignment with the project patterns described in
CLAUDE.md
Read the modified file and provide a detailed review with
specific recommendations.
> Review the recently modified code in this FastAPI project
ctrl+b to run in background

Subagent vs Skills

Subagents

You want to delegate a task to a separate execution context

You need different tool access than the main conversation

You want isolation between delegated work and main context

Skills

You want to enhance Claude's knowledge for the current task

The expertise applies throughout a conversation

Hooks vs Skills

Hooks

Operations that should run on every file save

Validation before specific tool calls

Automated side effects of Claude's actions

Skills

Knowledge that informs how Claude handles requests

Skill的内容

With optional fields:

```yaml
---
name: pdf-processing
description: Extract text and tables from PDF files, fill forms, merge documents.
license: Apache-2.0
metadata:
  author: example-org
  version: "1.0"
---
```

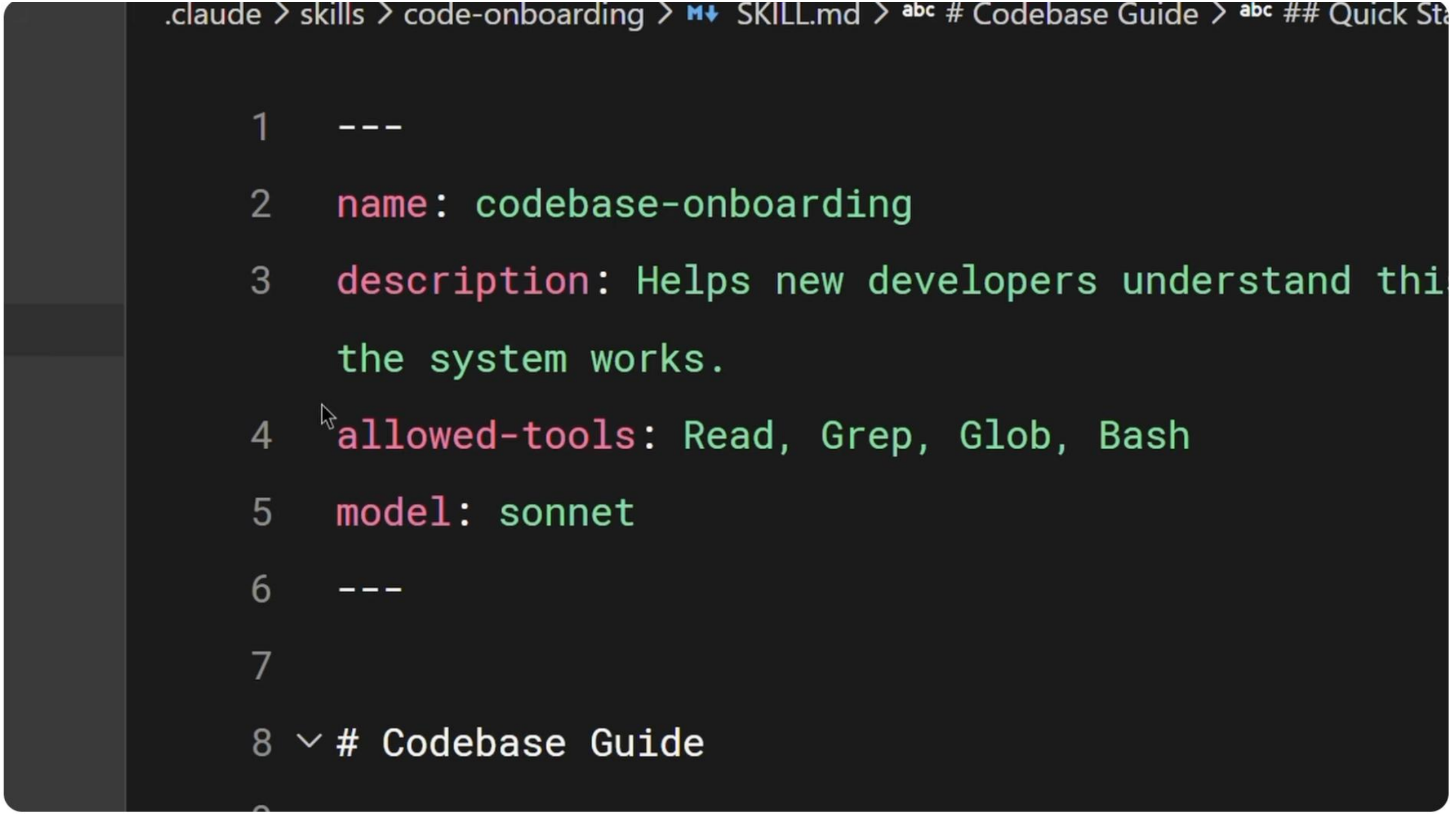
Field	Required	Constraints
`name`	Yes	Max 64 characters. Lowercase letters, numbers, and hyphens only. Must not start or end with a hyphen.
`description`	Yes	Max 1024 characters. Non-empty. Describes what the skill does and when to use it.
`license`	No	License name or reference to a bundled license file.
`compatibility`	No	Max 500 characters. Indicates environment requirements (intended product, system packages, network access, etc.).

Skill的内容

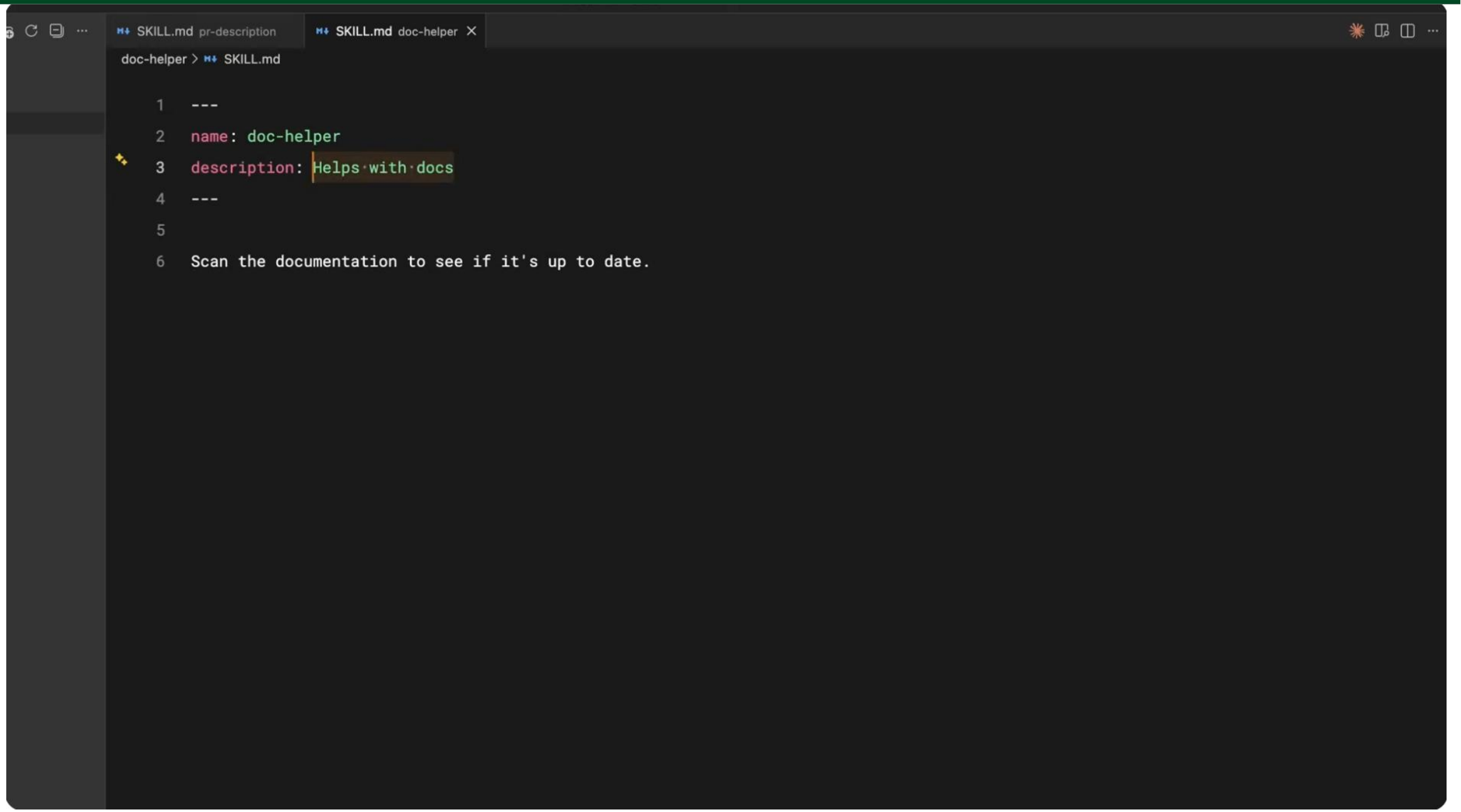
```
description: Extract text and tables from PDF files, fill forms, merge documents.
license: Apache-2.0
metadata:
  author: example-org
  version: "1.0"
---
```

Field	Required	Constraints
`name`	Yes	Max 64 characters. Lowercase letters, numbers, and hyphens only. Must not start or end with a hyphen.
`description`	Yes	Max 1024 characters. Non-empty. Describes what the skill does and when to use it.
`license`	No	License name or reference to a bundled license file.
`compatibility`	No	Max 500 characters. Indicates environment requirements (intended product, system packages, network access, etc.).
`metadata`	No	Arbitrary key-value mapping for additional metadata.
`allowed-tools`	No	Space-delimited list of pre-approved tools the skill may use. (Experimental)

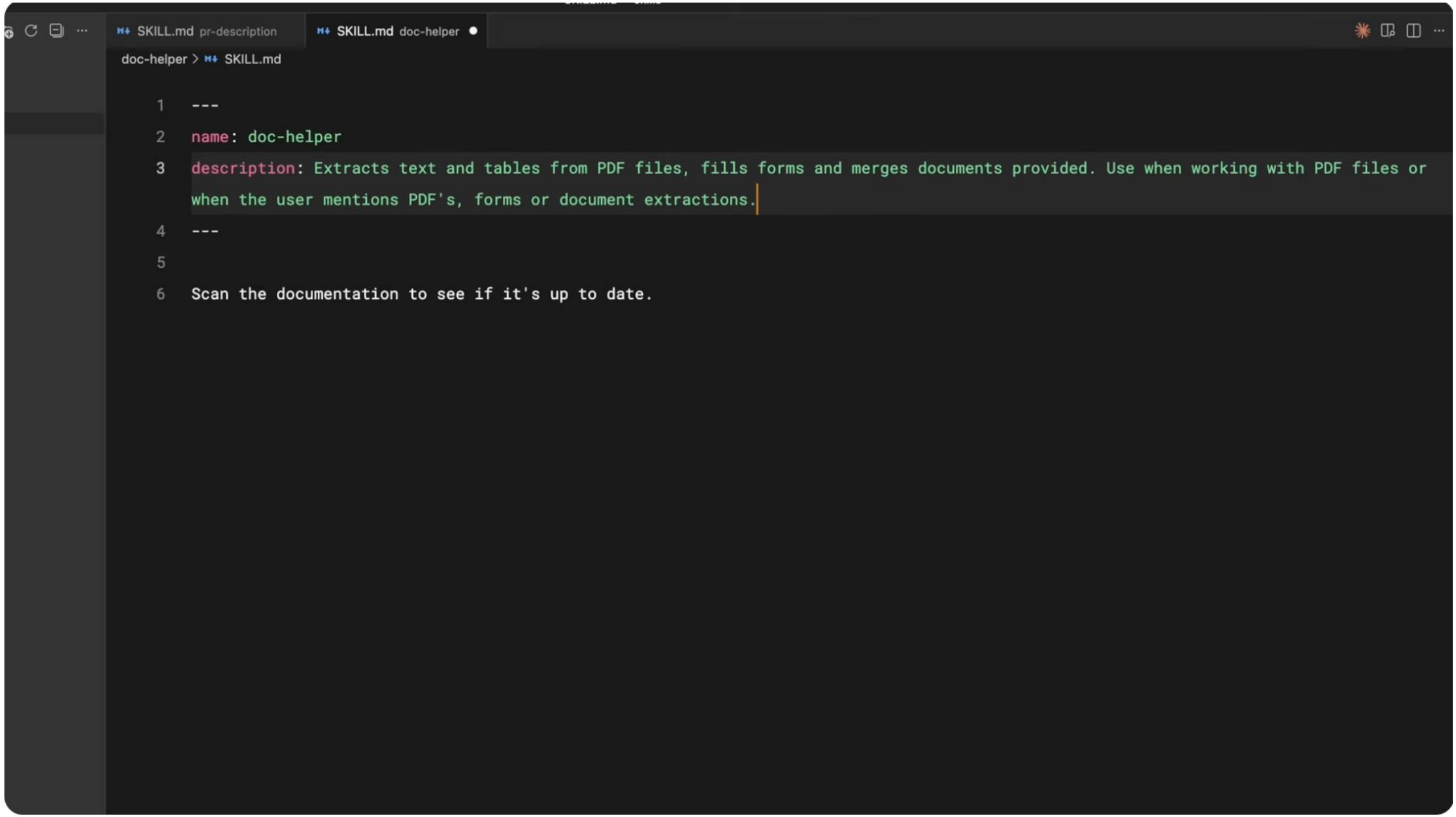
Skill的内容


.claude > skills > code-onboarding > SKILL.md > # Codebase Guide > ## Quick Sta
1 ---
2 name: codebase-onboarding
3 description: Helps new developers understand thi
the system works.
4 allowed-tools: Read, Grep, Glob, Bash
5 model: sonnet
6 ---
7
8 # Codebase Guide

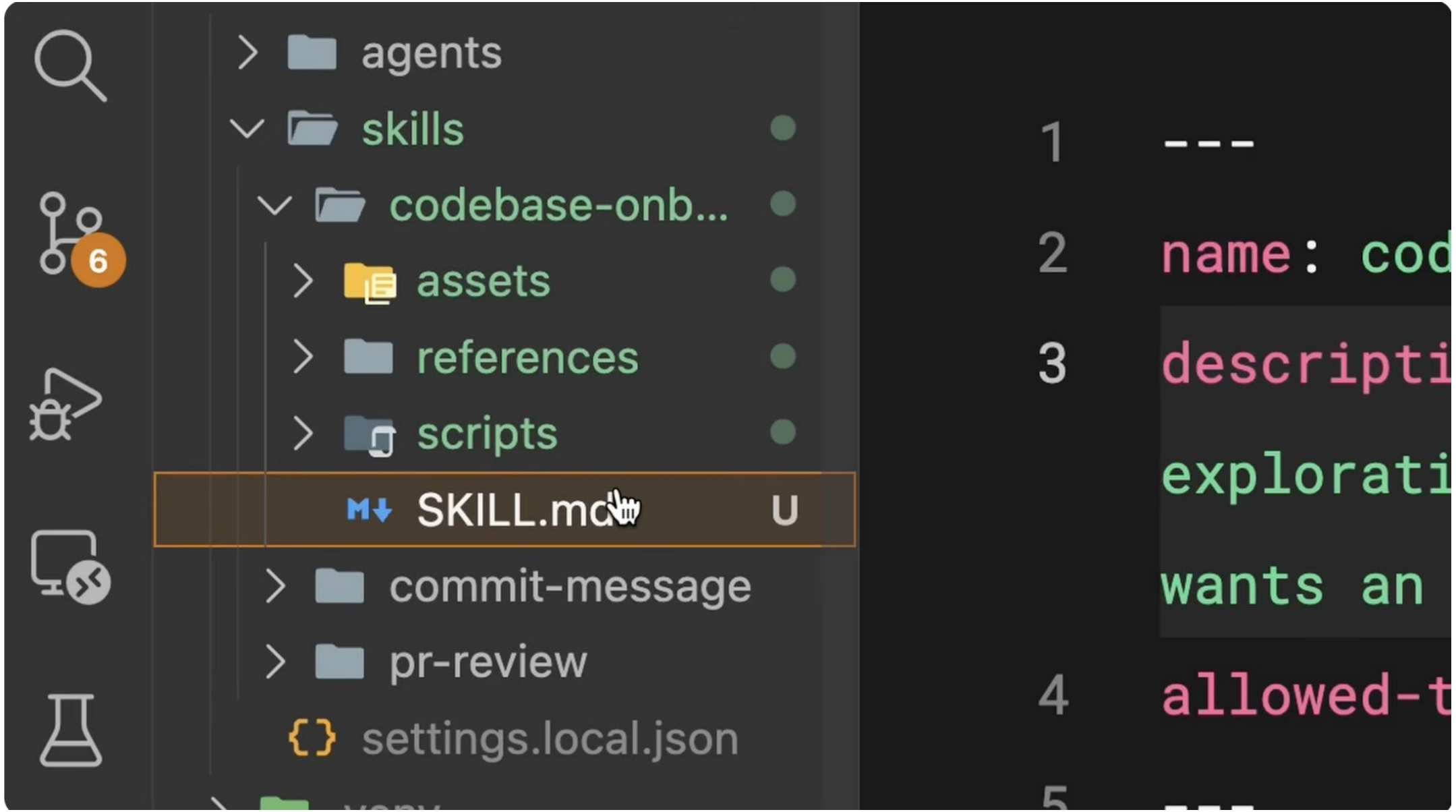
Skill的描述


SKILL.md pr-description
SKILL.md doc-helper
doc-helper > SKILL.md
1 ---
2 name: doc-helper
3 description: Helps with docs
4 ---
5
6 Scan the documentation to see if it's up to date.

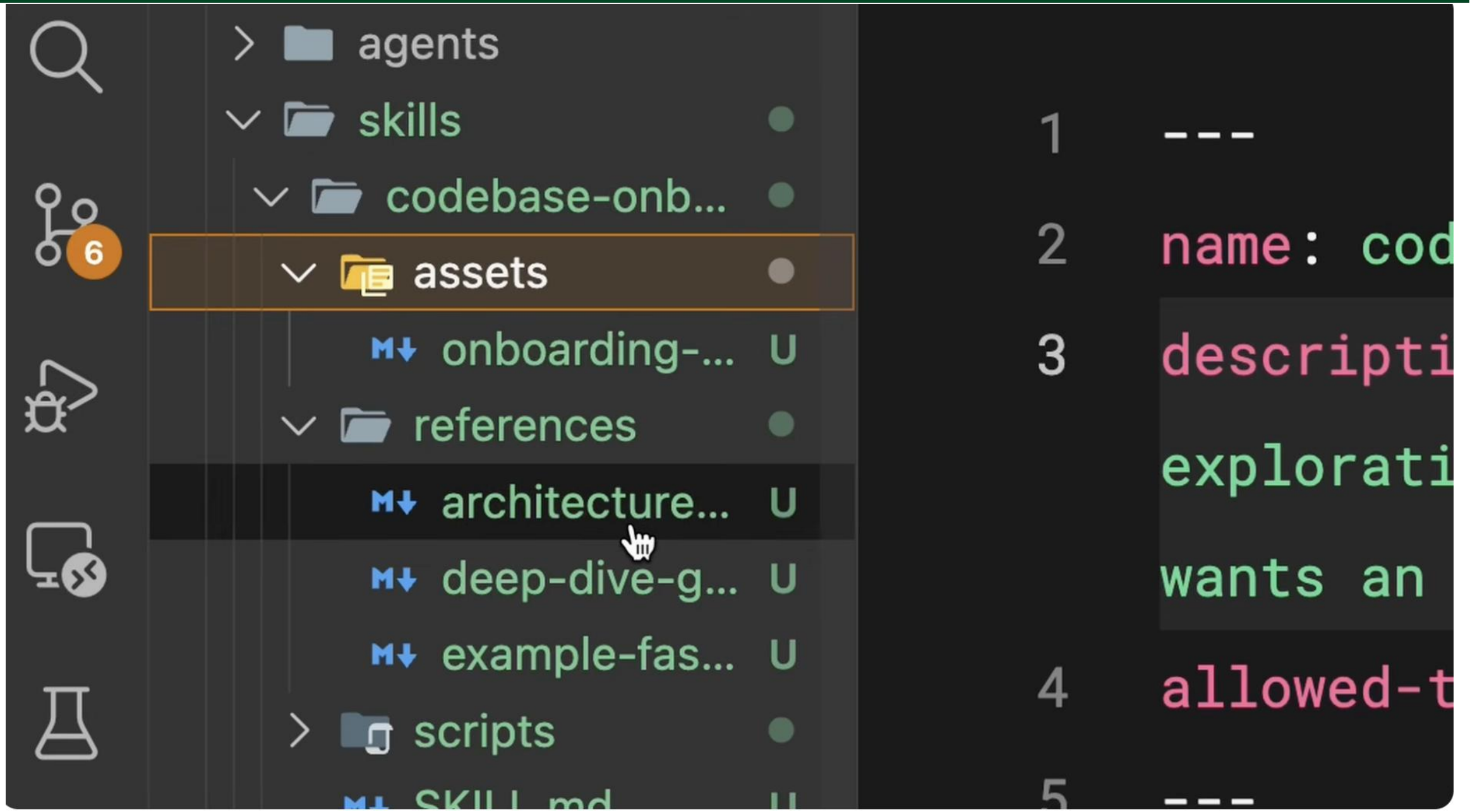
Skill的描述


SKILL.md pr-description
SKILL.md doc-helper
doc-helper > SKILL.md
1 ---
2 name: doc-helper
3 description: Extracts text and tables from PDF files, fills forms and merges documents provided. Use when working with PDF files or when the user mentions PDF's, forms or document extractions.
4 ---
5
6 Scan the documentation to see if it's up to date.

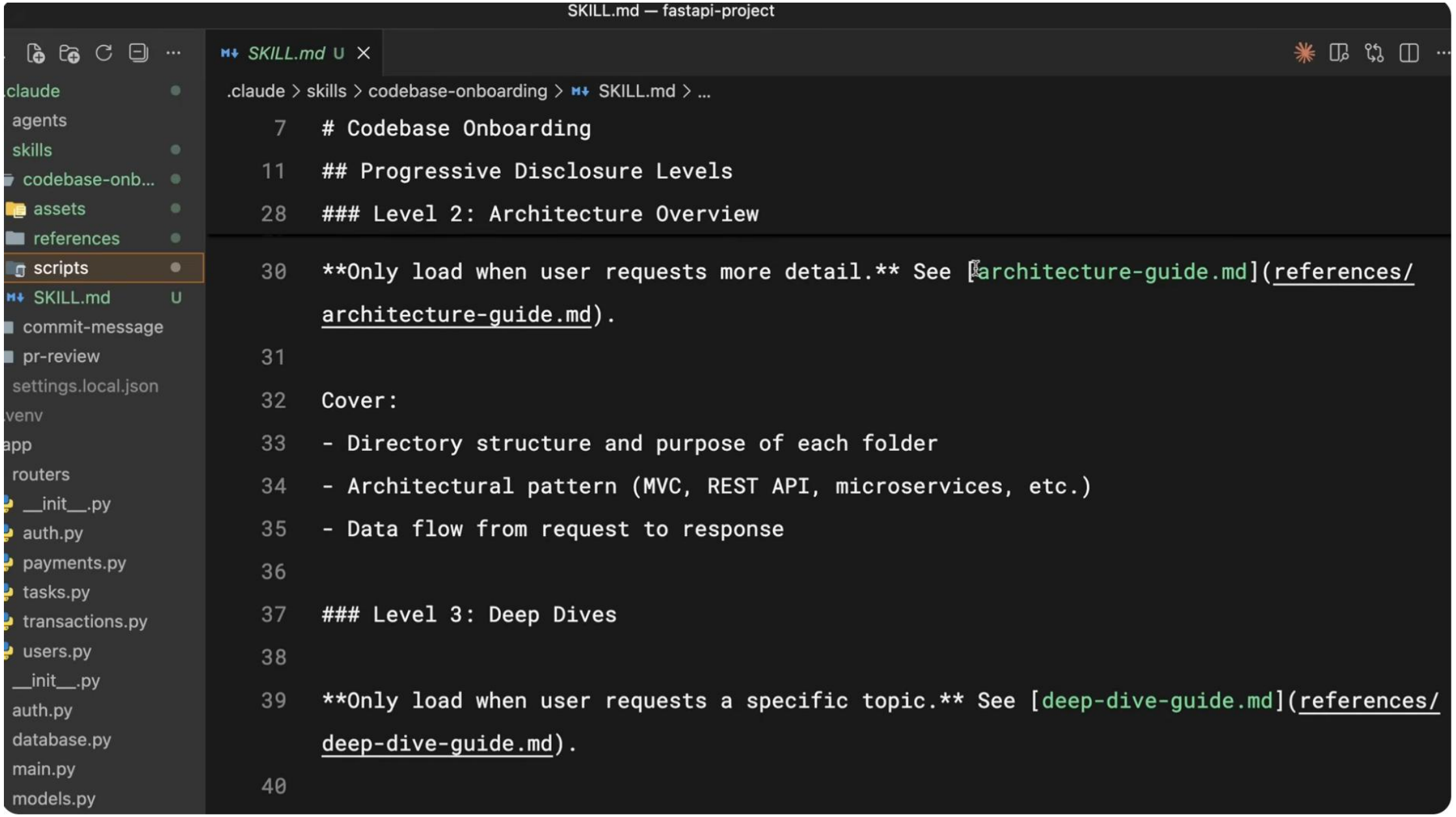
Skill的其他相关目录和文件



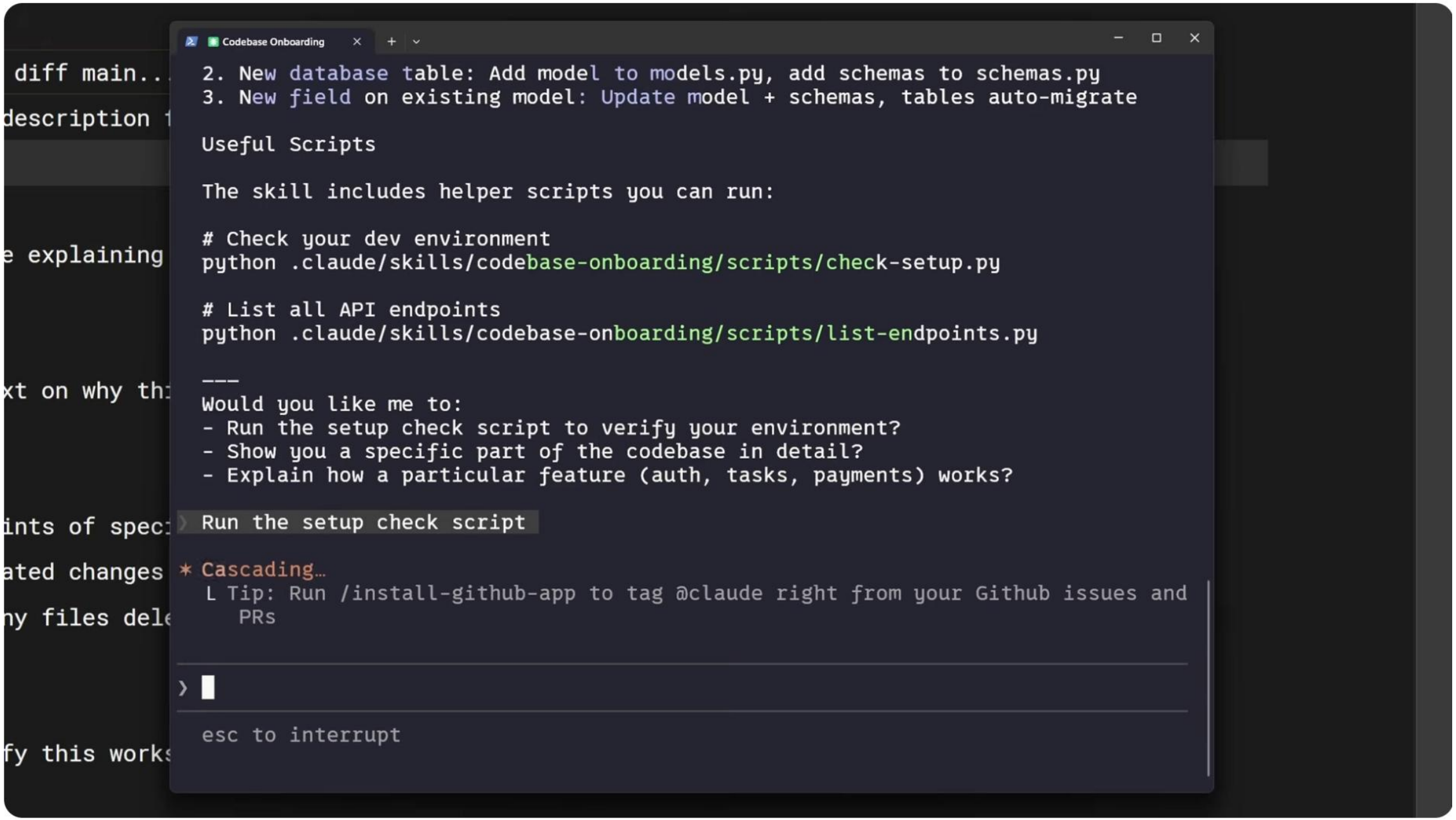
Skill的其他相关目录和文件



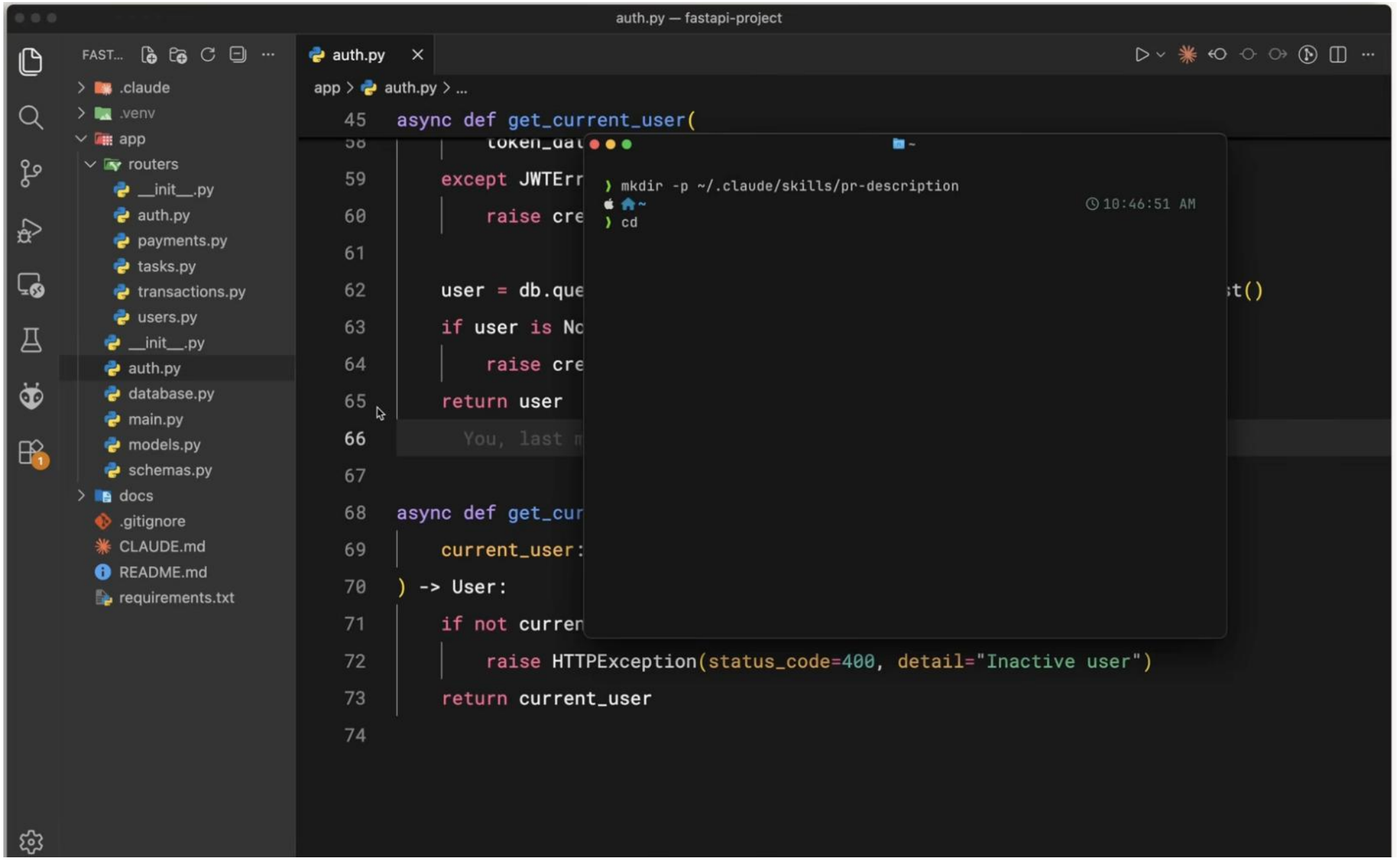
Skill的reference



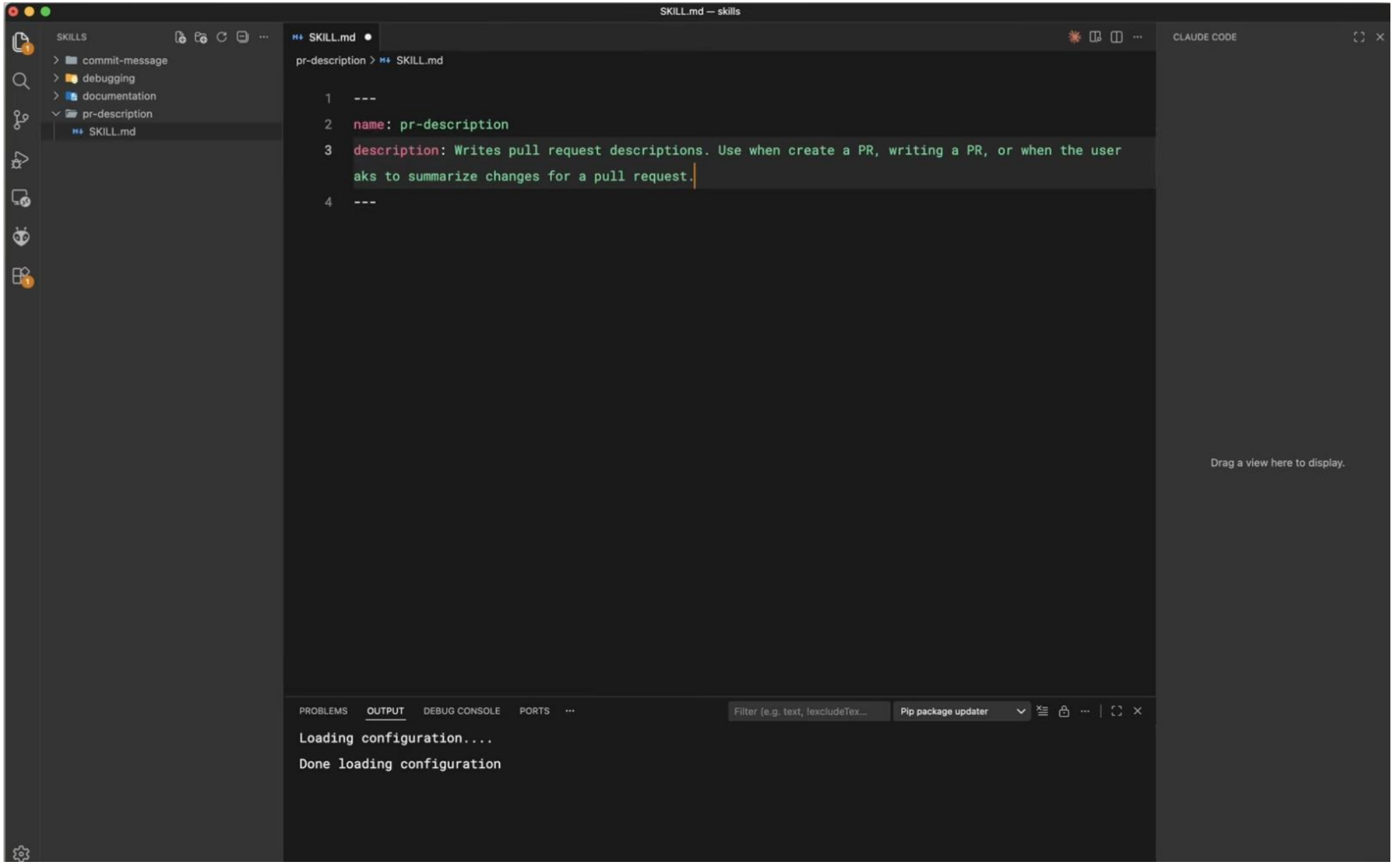
Skill的脚本


Codebase Onboarding
2. New database table: Add model to models.py, add schemas to schemas.py
3. New field on existing model: Update model + schemas, tables auto-migrate
Useful Scripts
The skill includes helper scripts you can run:
Check your dev environment
python .claude/skills/codebase-onboarding/scripts/check-setup.py
List all API endpoints
python .claude/skills/codebase-onboarding/scripts/list-endpoints.py
---
Would you like me to:
- Run the setup check script to verify your environment?
- Show you a specific part of the codebase in detail?
- Explain how a particular feature (auth, tasks, payments) works?
Run the setup check script
Cascading…
Tip: Run /install-github-app to tag @claude right from your Github issues and PRs
esc to interrupt

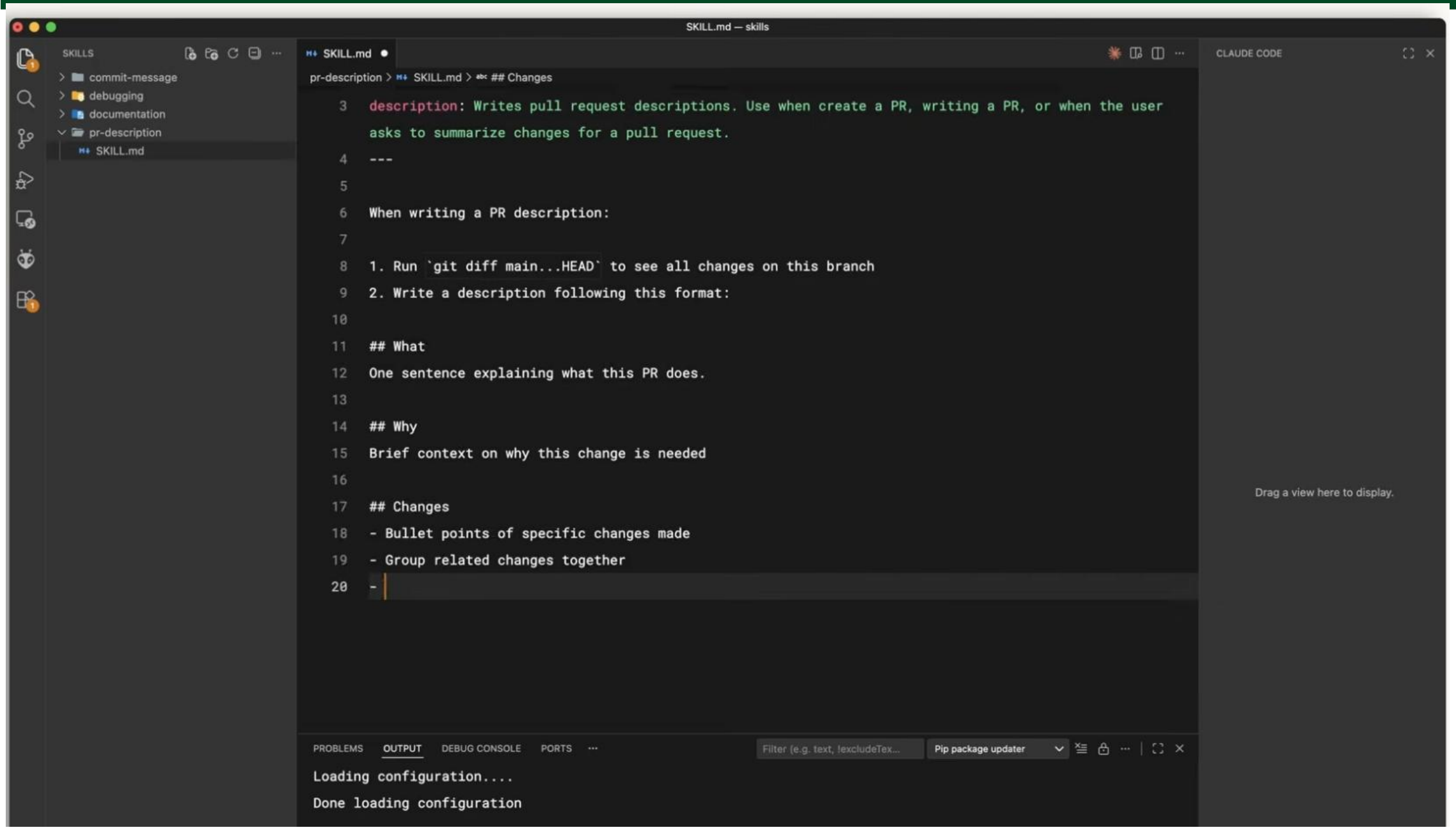
动手创一个skill


auth.py — fastapi-project
FAST...
.claude
.venv
app
routers
__init__.py
auth.py
payments.py
tasks.py
transactions.py
users.py
__init__.py
auth.py
database.py
main.py
models.py
schemas.py
docs
.gitignore
CLAUDE.md
README.md
requirements.txt
auth.py
app > auth.py > ...
45 async def get_current_user(
59 except JWTErr
60 raise cre
61
62 user = db.que
63 if user is No
64 raise cre
65 return user
66 You, last m
67
68 async def get_cur
69 current_user:
70) -> User:
71 if not curren
72 raise HTTPException(status_code=400, detail="Inactive user")
73 return current_user
74
~
mkdir -p ~/.claude/skills/pr-description
~
10:46:51 AM
cd

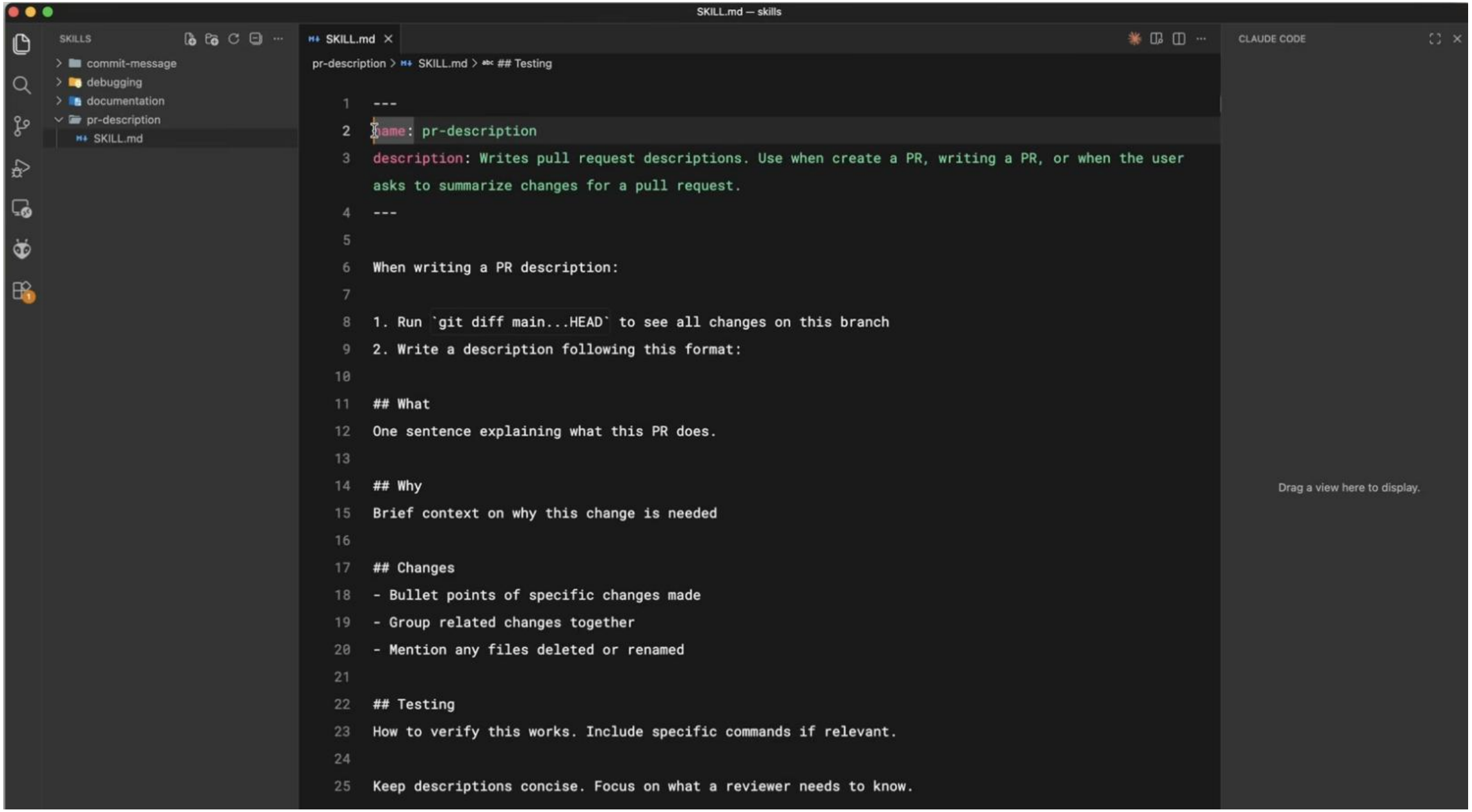
填写相应的内容


SKILL.md — skills
SKILLS
commit-message
debugging
documentation
pr-description
SKILL.md
SKILL.md
pr-description > SKILL.md
1 ---
2 name: pr-description
3 description: Writes pull request descriptions. Use when create a PR, writing a PR, or when the user
aks to summarize changes for a pull request.
4 ---
CLAUDE CODE
Drag a view here to display.
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
Filter (e.g. text, !excludeTex...
Pip package updater
Loading configuration....
Done loading configuration

填写相应的内容

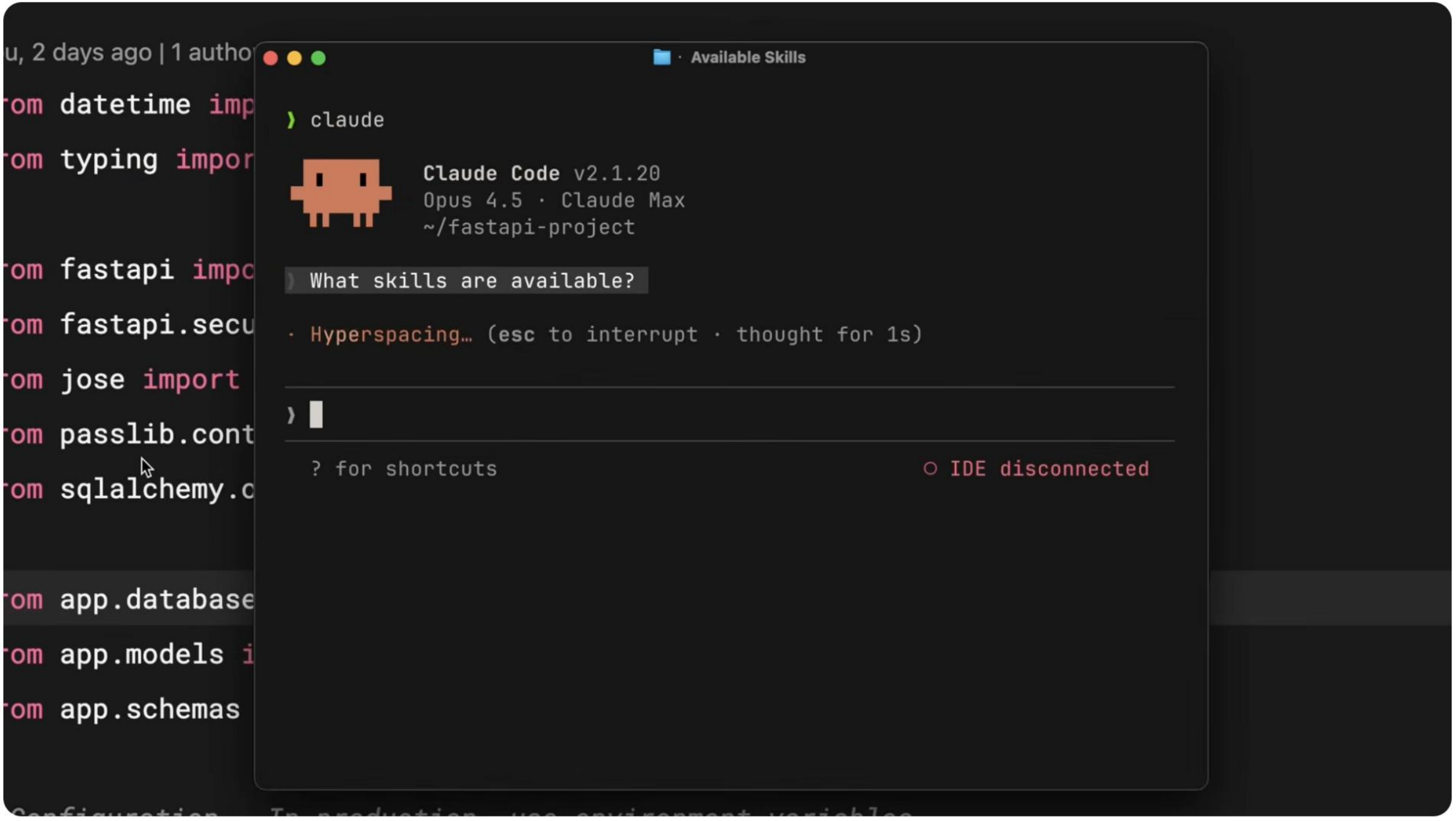

SKILL.md — skills
SKILLS
commit-message
debugging
documentation
pr-description
SKILL.md
SKILL.md
pr-description > SKILL.md > ## Changes
3 description: Writes pull request descriptions. Use when create a PR, writing a PR, or when the user asks to summarize changes for a pull request.
4 ---
5
6 When writing a PR description:
7
8 1. Run `git diff main...HEAD` to see all changes on this branch
9 2. Write a description following this format:
10
11 ## What
12 One sentence explaining what this PR does.
13
14 ## Why
15 Brief context on why this change is needed
16
17 ## Changes
18 - Bullet points of specific changes made
19 - Group related changes together
20 -
CLAUDE CODE
Drag a view here to display.
PROBLEMS
OUTPUT
DEBUG CONSOLE
PORTS
Filter (e.g. text, !excludeTex...
Pip package updater
Loading configuration....
Done loading configuration

填写相应的内容

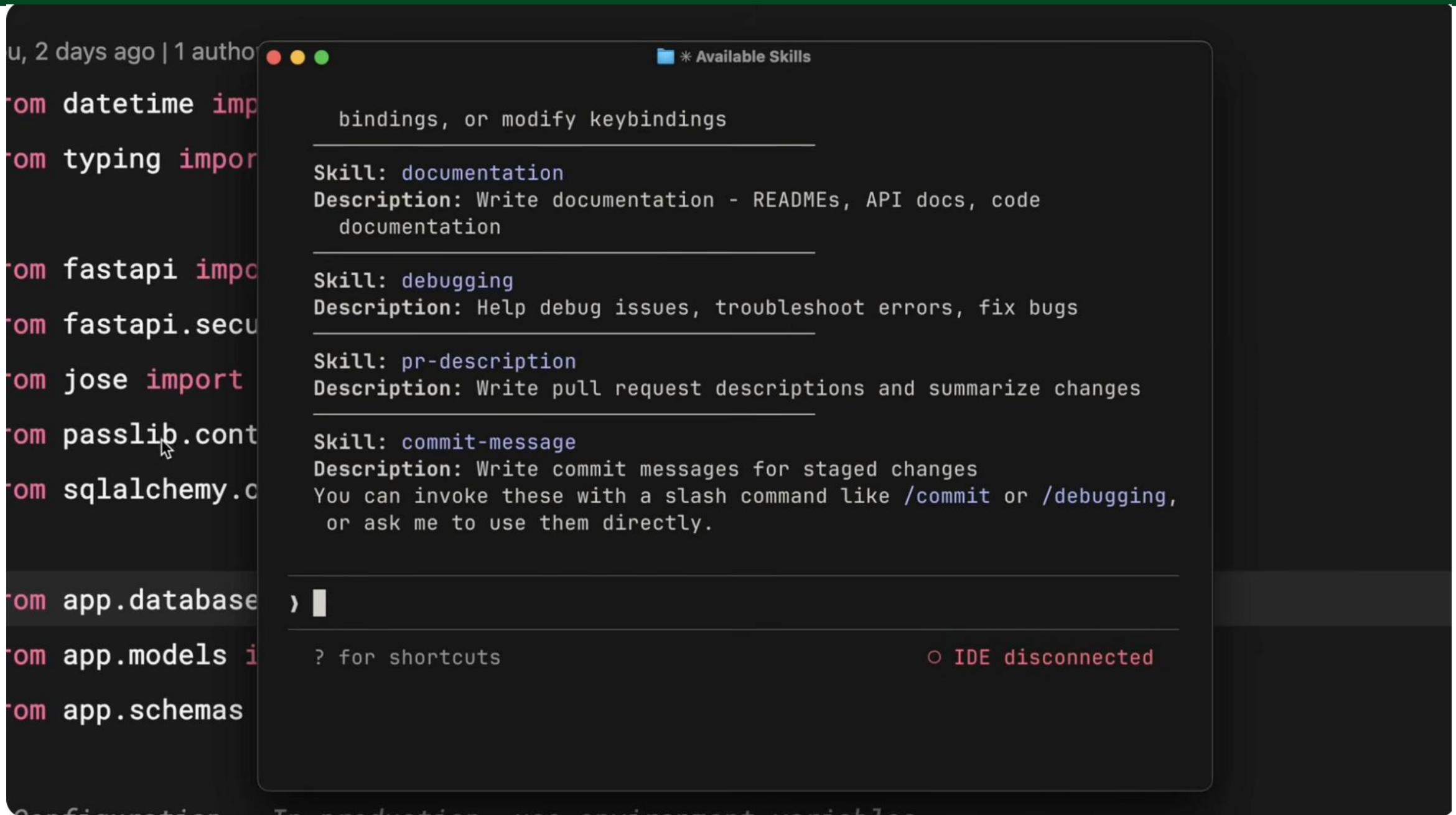

SKILL.md — skills
SKILLS
commit-message
debugging
documentation
pr-description
SKILL.md
SKILL.md
pr-description > SKILL.md > ## Testing
CLAUDE CODE
Drag a view here to display.
---
name: pr-description
description: Writes pull request descriptions. Use when create a PR, writing a PR, or when the user asks to summarize changes for a pull request.

When writing a PR description:
1. Run `git diff main...HEAD` to see all changes on this branch
2. Write a description following this format:
What
One sentence explaining what this PR does.
Why
Brief context on why this change is needed
Changes
- Bullet points of specific changes made
- Group related changes together
- Mention any files deleted or renamed
Testing
How to verify this works. Include specific commands if relevant.
Keep descriptions concise. Focus on what a reviewer needs to know.

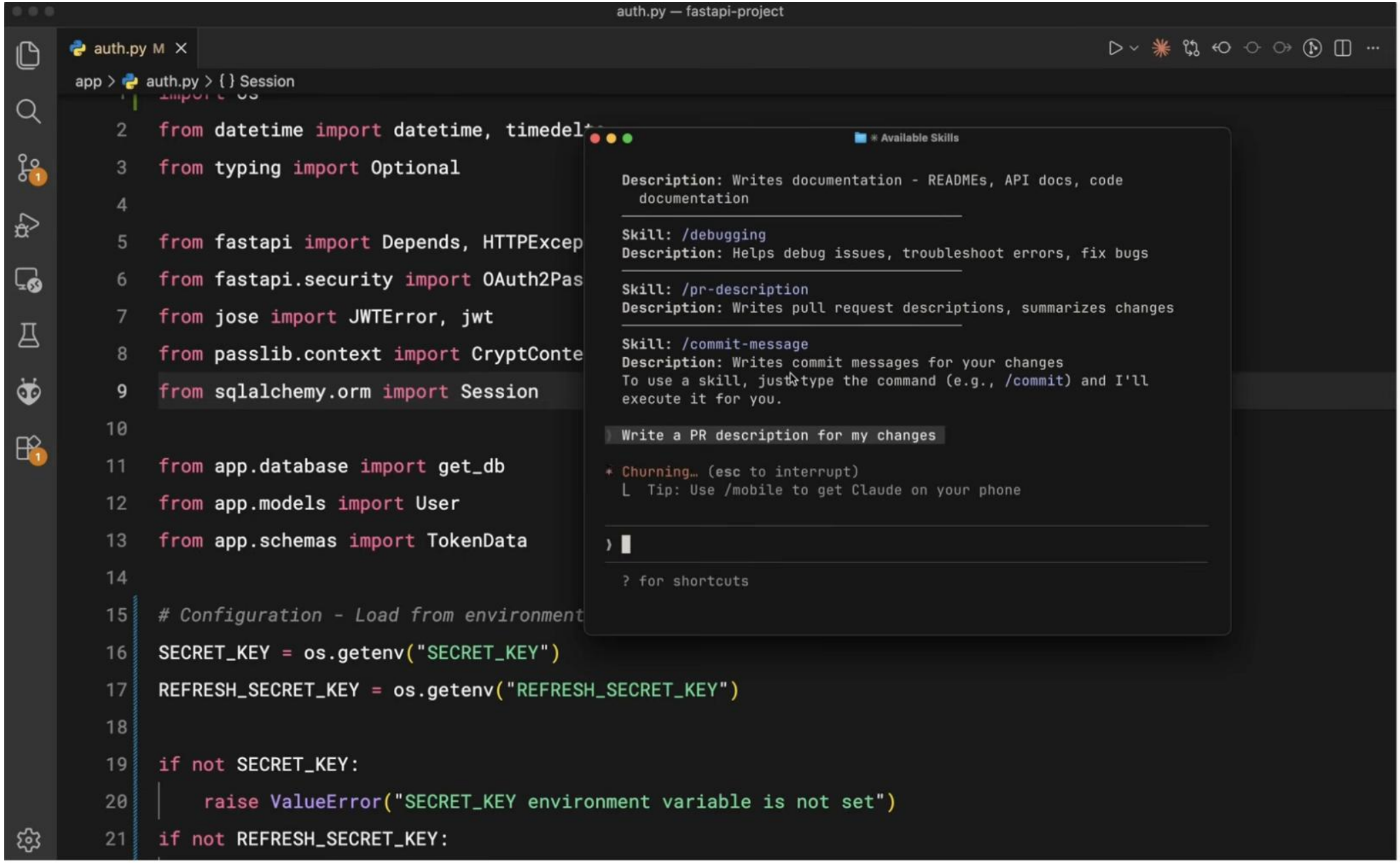
查看skills


u, 2 days ago | 1 autho
om datetime imp
om typing impor
om fastapi impo
om fastapi.secu
om jose import
om passlib.cont
om sqlalchemy.c
om app.database
om app.models i
om app.schemas
· Available Skills
❯ claude
Claude Code v2.1.20
Opus 4.5 · Claude Max
~/fastapi-project
❯ What skills are available?
· Hyperspacing… (esc to interrupt · thought for 1s)
❯
? for shortcuts
○ IDE disconnected

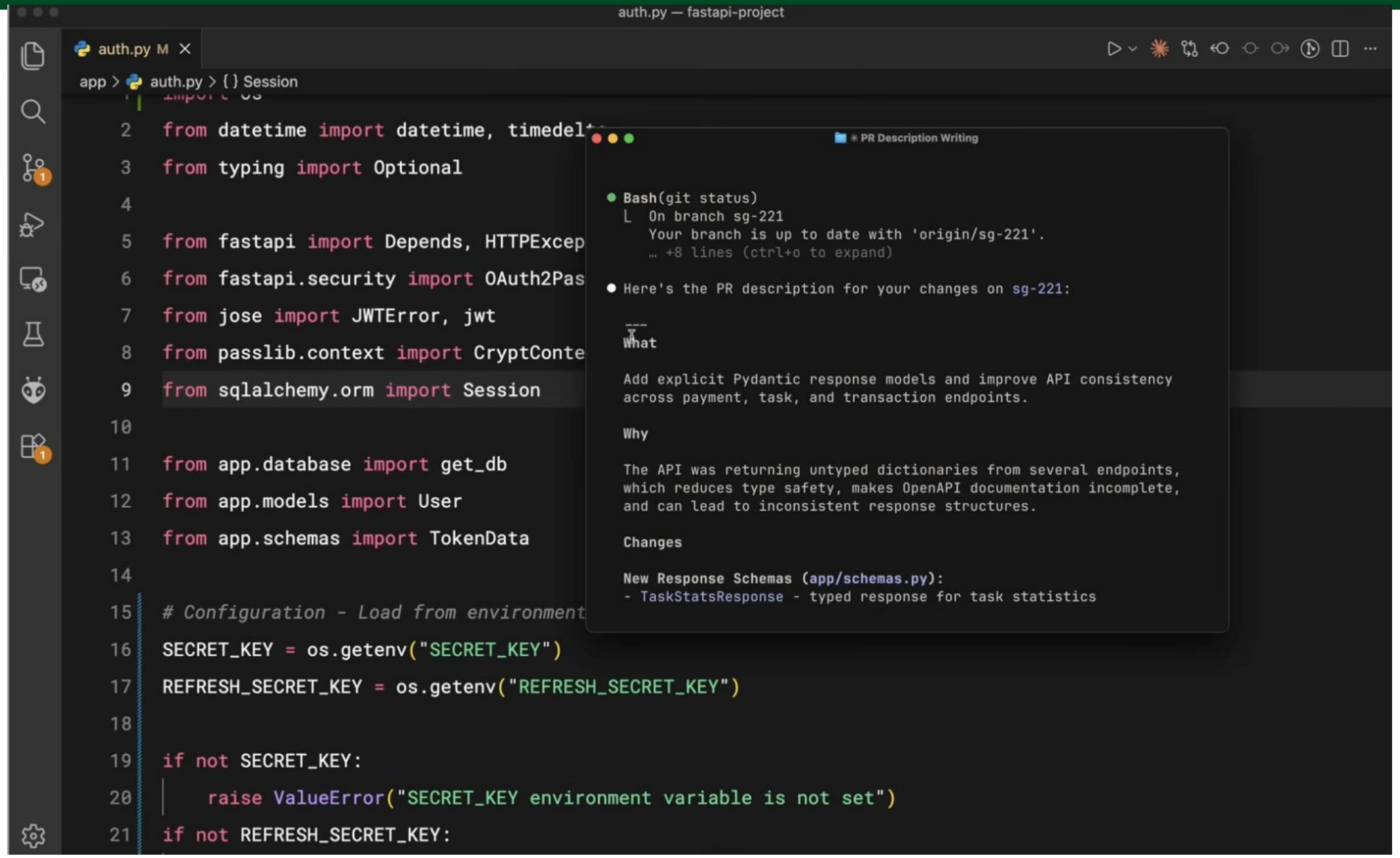
相看skills


u, 2 days ago | 1 autho
rom datetime imp
rom typing impor
rom fastapi impo
rom fastapi.secu
rom jose import
rom passlib.cont
rom sqlalchemy.c
rom app.database
rom app.models i
rom app.schemas
✳ Available Skills
bindings, or modify keybindings
Skill: documentation
Description: Write documentation - READMEs, API docs, code documentation
Skill: debugging
Description: Help debug issues, troubleshoot errors, fix bugs
Skill: pr-description
Description: Write pull request descriptions and summarize changes
Skill: commit-message
Description: Write commit messages for staged changes
You can invoke these with a slash command like /commit or /debugging, or ask me to use them directly.
? for shortcuts
○ IDE disconnected

调用skill


auth.py — fastapi-project
auth.py M
app > auth.py > { } Session
from datetime import datetime, timedel
from typing import Optional
from fastapi import Depends, HTTPExcep
from fastapi.security import OAuth2Pas
from jose import JWTError, jwt
from passlib.context import CryptConte
from sqlalchemy.orm import Session
from app.database import get_db
from app.models import User
from app.schemas import TokenData
# Configuration - Load from environment
SECRET_KEY = os.getenv("SECRET_KEY")
REFRESH_SECRET_KEY = os.getenv("REFRESH_SECRET_KEY")
if not SECRET_KEY:
raise ValueError("SECRET_KEY environment variable is not set")
if not REFRESH_SECRET_KEY:
Available Skills
Description: Writes documentation - READMEs, API docs, code documentation
Skill: /debugging
Description: Helps debug issues, troubleshoot errors, fix bugs
Skill: /pr-description
Description: Writes pull request descriptions, summarizes changes
Skill: /commit-message
Description: Writes commit messages for your changes
To use a skill, just type the command (e.g., /commit) and I'll execute it for you.
Write a PR description for my changes
Churning… (esc to interrupt)
Tip: Use /mobile to get Claude on your phone
? for shortcuts

执行skill


auth.py — fastapi-project
auth.py M
app > auth.py > { } Session
from datetime import datetime, timedel
from typing import Optional
from fastapi import Depends, HTTPExcep
from fastapi.security import OAuth2Pas
from jose import JWTError, jwt
from passlib.context import CryptConte
from sqlalchemy.orm import Session
from app.database import get_db
from app.models import User
from app.schemas import TokenData
# Configuration - Load from environment
SECRET_KEY = os.getenv("SECRET_KEY")
REFRESH_SECRET_KEY = os.getenv("REFRESH_SECRET_KEY")
if not SECRET_KEY:
raise ValueError("SECRET_KEY environment variable is not set")
if not REFRESH_SECRET_KEY:
✳ PR Description Writing
Bash(git status)
L On branch sg-221
Your branch is up to date with 'origin/sg-221'.
… +8 lines (ctrl+o to expand)
Here's the PR description for your changes on sg-221:

What
Add explicit Pydantic response models and improve API consistency across payment, task, and transaction endpoints.
Why
The API was returning untyped dictionaries from several endpoints, which reduces type safety, makes OpenAPI documentation incomplete, and can lead to inconsistent response structures.
Changes
New Response Schemas (app/schemas.py):
- TaskStatsResponse - typed response for task statistics

什么是 harness？

> **Harness 是包裹在 LLM 外面的执行系统，负责把用户任务转化为模型输入，把模型输出转化为工具调用、文件修改、代码执行、验证反馈和最终结果。**

OpenAI 对 Codex 的解释很接近这个定义：Codex harness 提供核心的 **agent loop** 和执行逻辑，负责组织用户、模型和工具之间的交互。模型可以请求工具调用，harness 执行工具，再把工具结果放回上下文，循环直到任务结束。([OpenAI][2])

什么是 harness？

Harness=(Prompting,Context,Tools,Memory,Execution,Validation,Recovery,Permissions)

模型看到什么上下文；
什么时候加载 skill；
可以调用哪些工具；
工具输出如何反馈给模型；
是否能运行代码、测试、lint、type checker；
是否能读写文件；
失败后是否自动重试；
最终结果如何验证。

为什么同一个skill 在不同 harness中效果不同??

因为 skill 通常只是“知识包”，但它需要 harness 来完成以下关键动

- skill 触发机制不同
- 工具接口不同
- 观察反馈格式不同
- 验证机制不同
- 失败恢复能力不同

GPT-5.5

- 因此，严格地说，不应该说：

- > GPT-5.5 在某任务上成功率是多少。

- 而应该说：

- > GPT-5.5 + Skill X + Codex harness + 某工具环境 + 某 budget 的成功率是多少。

如何写好的 skill？

- **Name**：skill 名字要短、明确。
- **Trigger**：什么时候使用，什么时候不要使用。
- **Goal**：这个 skill 解决什么问题。
- **Procedure**：分步骤流程。
- **Tools**：推荐使用哪些命令、脚本、库。
- **Constraints**：不能做什么，必须保留什么。
- **Validation**：如何判断完成。
- **Failure Cases**：常见错误与修复方式。

写 skill 的高级原则

- **skill 的 description 比正文更重要**
  - ✓因为很多 harness 先加载 skill 的 metadata，再决定是否读取完整 skill。Anthropic 的 skill 机制就是先把 name 和 description 作为轻量信息加载，只有判断相关时才读完整 SKILL.md
- **skill 要短主干、多文件展开**
  - ✓不要把所有知识放进一个巨大 SKILL.md
- **kill 要包含“不要做什么”**
- **skill 要包含 output contract**
  - ✓返回 LaTeX；保留 equation 环境；不要使用 bullet；控制在 1000 字左右；输出修改后的完整段落，而不是解释。
- **skill 要有自验证步骤**

如何写好的 harness？

- Planner：任务规划器
- Context Manager：上下文管理器
- Skill Router：skill 路由器
- Tool Executor：工具执行器
- State Tracker：状态跟踪器
- Verifier：验证器
- Recovery Manager：错误恢复器

好 harness 的设计原则

- **harness 必须把“想法”绑定到“状态”**
  - ✓Before final answer:
    - 1. List changed files.
    - 2. Run required validators.
    - 3. Confirm output artifact exists.
    - 4. Compare result with user requirement.
    - 5. Only then answer.
- **harness 要有强 output contract**
- **harness 要让工具输出可学习**
- **harness 要内置验证循环**
- **harness 要管理权限和风险**
- **harness 要支持 trace**

多数情况下，不需要从零写 harness

- 目标是解决实际问题，例如写代码、改论文、生成 PPT、处理 repo、跑测试、做数据分析，那么直接用现成 harness 就可以

什么时候需要自己写一点 harness

- 研究 agent skill 的效果，那么只用现成 harness 还不够。因为 skill 的效果会被 harness 强烈影响
- thin harness wrapper
- 最小 harness

什么时候真正写harness

- 如果研究重点是 harness，才需要真正写 harness

最小 harness

```
mini_skill_harness/
  skills/
    code-review/
      SKILL.md
    latex-refine/
      SKILL.md
    bug-fix/
      SKILL.md

  tasks/
    tasks.jsonl
```

```
  harness.py
  runners/
    codex_runner.py
    claude_runner.py

  verifiers/
    run_tests.py
    check_diff.py
    check_artifact.py

  logs/
    runs.jsonl
```

最小harness

```python
import asyncio
from claude_agent_sdk import query, ClaudeAgentOptions

async def run_claude(prompt: str, workdir: str):
    options = ClaudeAgentOptions(
        allowed_tools=["Read", "Edit", "Bash"],
        cwd=workdir,
    )

    messages = []
    async for message in query(prompt=prompt, options=options):
        messages.append(message)

    return messages

asyncio.run(run_claude("Fix the failing test.", "./repo"))
```

Many agent skill papers are not fundamentally innovative.

- skill library = prompt/code memory
- skill acquisition = LLM reflection + edit + retry
- skill selection = retrieval or ranking
- skill evolution = prompt optimization over trajectories

Native Agent Skill Model

$$\mathcal{S}_i = (z_i, \pi_i, C_i, E_i, R_i, V_i)$$

a skill

$$p_i = C_i(s_t, g)$$

is the precondition

$$i^* = \arg\max_i \; C_i(s_t, g)\,[V_i(s_t, g) - \lambda R_i(s_t, g)]$$

the success/value estimator

the resource cost

More innovative contribution: skill algebra

The real novelty could be to define how skills compose:

$$\mathcal{S}_{ij} = \mathcal{S}_j \circ \mathcal{S}_i$$

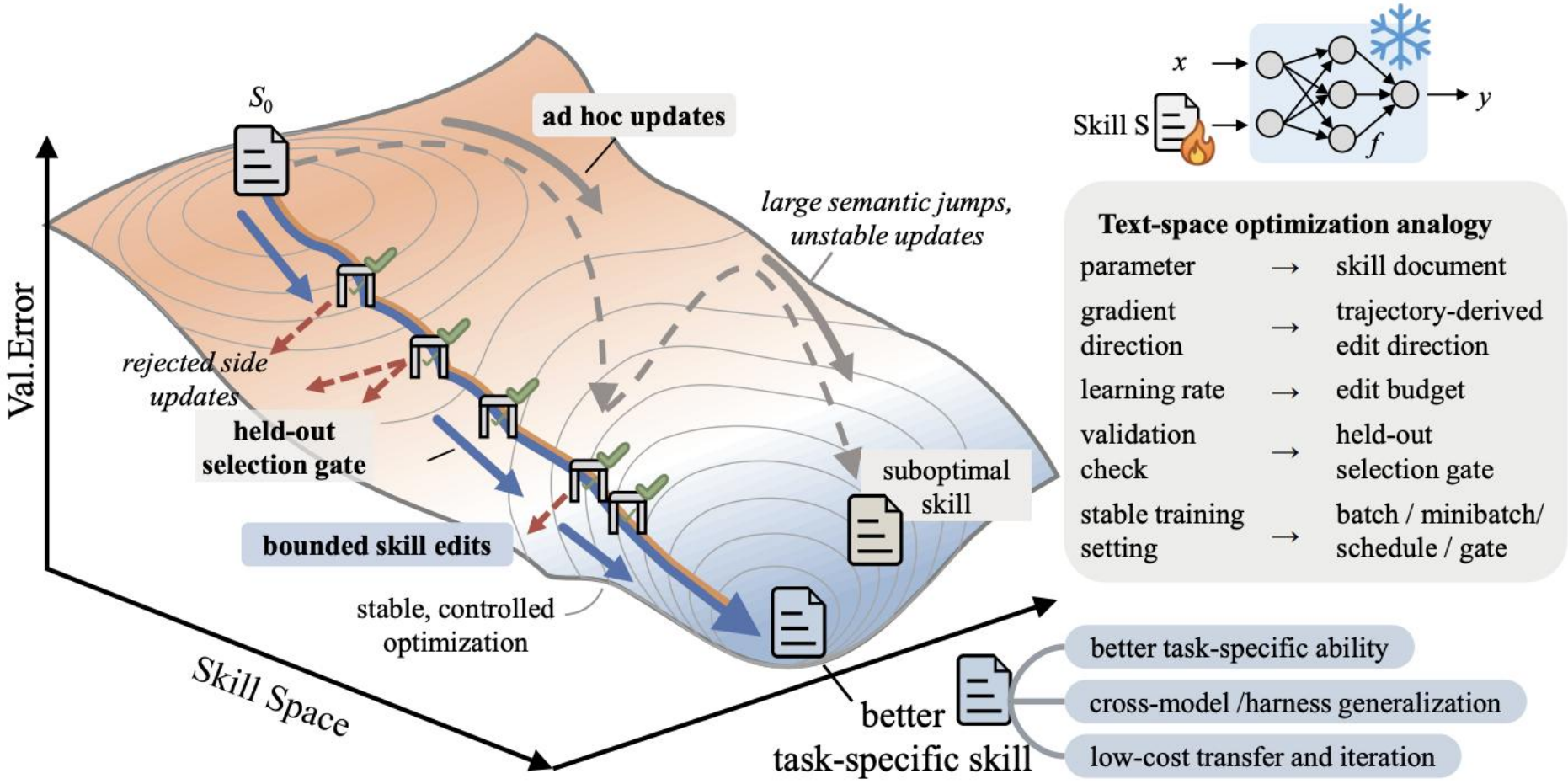
with the condition:

$$E_i(s) \in \mathrm{Dom}(C_j)$$

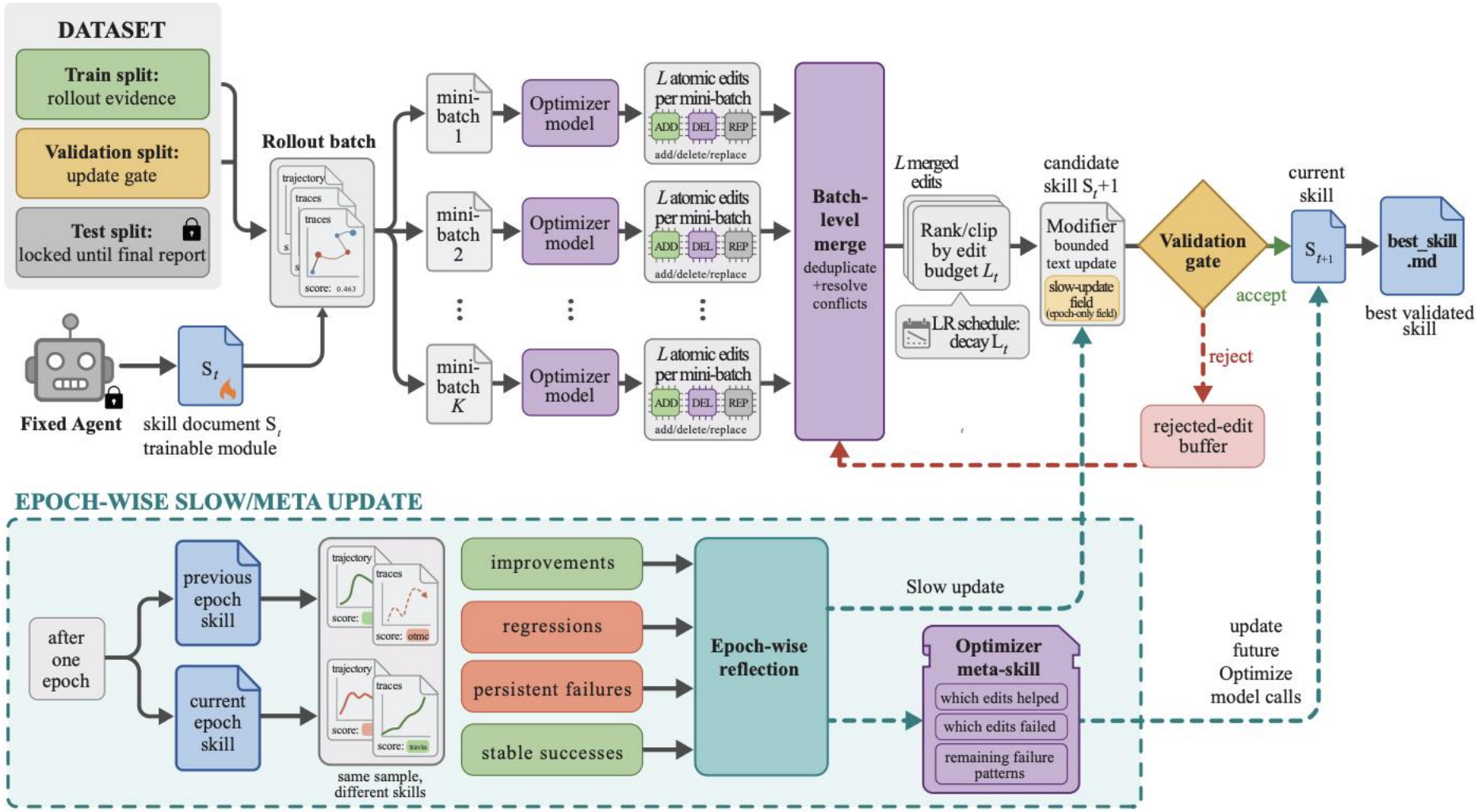
一些前沿方向

- 从静态获取走向动态的技能进化
- 资源感知与多目标选择
- 多模态与物理世界的技能落地

SkillOpt


S_0
ad hoc updates
large semantic jumps, unstable updates
rejected side updates
held-out selection gate
bounded skill edits
suboptimal skill
stable, controlled optimization
Val.Error
Skill Space
better task-specific skill
x
Skill S
y
f
Text-space optimization analogy
parameter → skill document
gradient direction → trajectory-derived edit direction
learning rate → edit budget
validation check → held-out selection gate
stable training setting → batch / minibatch/ schedule / gate
better task-specific ability
cross-model /harness generalization
low-cost transfer and iteration

SkillOpt


DATASET
Train split: rollout evidence
Validation split: update gate
Test split: locked until final report
Rollout batch
trajectory
traces
score: 0.463
Fixed Agent
S_t
skill document S_t
trainable module
mini-batch 1
mini-batch 2
mini-batch K
Optimizer model
L atomic edits per mini-batch
ADD DEL REP
add/delete/replace
Batch-level merge
deduplicate +resolve conflicts
L merged edits
Rank/clip by edit budget L_t
LR schedule: decay L_t
candidate skill S_t+1
Modifier bounded text update
slow-update field (epoch-only field)
Validation gate
accept
reject
rejected-edit buffer
current skill
S_{t+1}
best_skill .md
best validated skill
EPOCH-WISE SLOW/META UPDATE
after one epoch
previous epoch skill
current epoch skill
same sample, different skills
improvements
regressions
persistent failures
stable successes
Epoch-wise reflection
Slow update
Optimizer meta-skill
which edits helped
which edits failed
remaining failure patterns
update future Optimize model calls

SkillOpt

Model	Skill source	SearchQA	Spreadsheet	OfficeQA	DocVQA	LiveMath	ALFWorld
No harness / direct chat							
GPT–5.5	No skill	77.7	41.8	33.1	78.8	37.6	83.6
	Human skill	81.8$_{+4.1}$	72.9$_{+31.1}$	66.9$_{+33.8}$	90.1$_{+11.3}$	38.4$_{+0.8}$	91.8$_{+8.2}$
	LLM skill	80.9$_{+3.2}$	43.2$_{+1.4}$	51.7$_{+18.6}$	89.6$_{+10.8}$	40.0$_{+2.4}$	93.3$_{+9.7}$
	Trace2Skill	82.4$_{+4.7}$	49.6$_{+7.8}$	65.7$_{+32.6}$	90.6$_{+11.8}$	52.0$_{+14.4}$	87.3$_{+3.7}$
	TextGrad	81.4$_{+3.7}$	41.1$_{-0.7}$	42.0$_{+8.9}$	87.2$_{+8.4}$	49.2$_{+11.6}$	82.8$_{-0.8}$
	GEPA	84.8$_{+7.1}$	73.6$_{+31.8}$	63.9$_{+30.8}$	89.1$_{+10.3}$	43.2$_{+5.6}$	85.8$_{+2.2}$
	SkillOpt	**87.3**$_{+9.6}$	**80.7**$_{+38.9}$	**72.1**$_{+39.0}$	**91.2**$_{+12.4}$	**66.9**$_{+29.3}$	**95.5**$_{+11.9}$
Qwen3.6–35B-A3B	No skill	72.7	38.2	45.9	87.6	31.2	59.7
	Human skill	74.1$_{+1.4}$	44.3$_{+6.1}$	41.9$_{-4.0}$	88.4$_{+0.8}$	29.6$_{-1.6}$	44.8$_{-14.9}$
	LLM skill	72.6$_{-0.1}$	42.9$_{+4.7}$	46.5$_{+0.6}$	88.4$_{+0.8}$	24.8$_{-6.4}$	60.4$_{+0.7}$
	Trace2Skill	75.4$_{+2.7}$	33.2$_{-5.0}$	32.0$_{-13.9}$	90.4$_{+2.8}$	29.6$_{-1.6}$	70.9$_{+11.2}$
	TextGrad	76.4$_{+3.7}$	22.9$_{-15.3}$	33.7$_{-12.2}$	84.5$_{-3.1}$	7.2$_{-24.0}$	67.9$_{+8.2}$
	GEPA	75.8$_{+3.1}$	45.4$_{+7.2}$	43.6$_{-2.3}$	88.0$_{+0.4}$	31.2$_{+0.0}$	73.9$_{+14.2}$
	SkillOpt	**80.3**$_{+7.6}$	**47.5**$_{+9.3}$	**47.1**$_{+1.2}$	**91.4**$_{+3.8}$	**41.6**$_{+10.4}$	**82.1**$_{+22.4}$
Codex harness							
GPT–5.5	No skill	81.8	27.5	38.3	87.2	35.2	–
	Human skill	84.1$_{+2.3}$	50.7$_{+23.2}$	40.0$_{+1.7}$	88.8$_{+1.6}$	48.8$_{+13.6}$	–
	LLM skill	83.4$_{+1.6}$	25.0$_{-2.5}$	34.3$_{-4.0}$	89.8$_{+2.6}$	45.6$_{+10.4}$	–
	EvoSkill	61.4$_{-20.4}$	67.5$_{+40.0}$	42.4$_{+4.1}$	89.3$_{+2.1}$	63.2$_{+28.0}$	–
	SkillOpt	**87.3**$_{+5.5}$	**85.0**$_{+57.5}$	**51.1**$_{+12.8}$	**92.2**$_{+5.0}$	**78.4**$_{+43.2}$	–
Claude Code harness							
GPT–5.5	No skill	81.9	22.1	57.6	86.6	40.8	–
	Human skill	83.7$_{+1.8}$	37.1$_{+15.0}$	66.3$_{+8.7}$	88.0$_{+1.4}$	44.0$_{+3.2}$	–
	LLM skill	82.4$_{+0.5}$	37.1$_{+15.0}$	56.4$_{-1.2}$	89.6$_{+3.0}$	44.8$_{+4.0}$	–
	EvoSkill	84.0$_{+2.1}$	75.0$_{+52.9}$	70.3$_{+12.7}$	87.2$_{+0.6}$	52.0$_{+11.2}$	–
	SkillOpt	**85.9**$_{+4.0}$	**80.4**$_{+58.3}$	**71.5**$_{+13.9}$	**90.1**$_{+3.5}$	**56.5**$_{+15.7}$	–

SkillOpt

- s：技能文档，一段纯文本指令，例如一段 Markdown。
- M：冻结的目标执行模型，例如 GPT-5.5。
- O：独立的优化器模型，用于反思和修改技能。
- x：具体的任务，例如“计算 A 列的总和并填入 B1”。
- h：执行框架，负责将模型、任务和技能结合执行。
- $\tau(s)$：执行轨迹，包含模型的思考过程、代码生成、报错信息等。
- $r(s)$：任务得分，$r(s) \in [0, 1]$，其中 1 代表完全成功，0 代表失败。

第 1 步：前向传播与收集 Rollout 证据

$$\left(\tau(s_{cur}), r(s_{cur})\right) = h(M, x, s_{cur})$$

- **当前技能 s_{cur}**：使用 Python 的 pandas 库来处理电子表格。
- **执行**：模型 M 在 40 个电子表格任务上执行该技能。
- **结果**：25 个任务成功，得分 $r = 1$；15 个任务失败，得分 $r = 0$。在失败轨迹 τ 中，报错信息显示：测试系统期望读取单元格的 **静态数值**，但模型却写入了 Excel 公式，例如 `INDEX/MATCH`，导致判分系统无法读取。

第 2 步：小批量反思

$$E_{fail} = O(\mathcal{B}_{fail}, s_{cur})$$

$$E_{succ} = O(\mathcal{B}_{succ}, s_{cur})$$

- 优化器 O 拿到 15 个因为“写入公式而不是静态数值”而失败的轨迹。
- 优化器分析后发现这是一个系统性错误，于是生成一个 JSON 格式的修改建议。
- **操作**： append ，表示追加文本。
- **内容**：检查工作表结构和公式后，请将计算出的静态数值 （evaluated static values）写入目标范围，而不要依赖 Excel 的公式重算功能。

第 3 步：应用有界文本更新，也就是文本学习率

$$E_{top} = \mathrm{TopK}(E_{merged}, L_t)$$

$$\tilde{s} = \mathrm{Apply}(s_{cur}, E_{top})$$

- 当前的"学习率"预算为 $L_t = 4$。
- 由于"要求写入静态数值"的建议能够修复大量系统性失败， 优化器将其排在第 1 位。
- 系统将这条建议应用到文档中，生成 **候选技能** $\tilde{s}$：

"使用 Python 的 pandas 库来处理电子表格。 *新增：检查工作表结构和公式后， 请将计算出的静态数值写入目标范围， 而不要依赖 Excel 的公式重算功能。*"

第 4 步：验证门控

$$score_{cand} = \frac{1}{|D_{sel}|} \sum_{x \in D_{sel}} r_x(\tilde{s})$$

$$\text{If } score_{cand} > score_{cur}, \text{ then } s_{cur} \leftarrow \tilde{s}$$

- 我们拿着候选技能 $\tilde{s}$ 在独立的 D_{sel} 数据集上测试。
- **测试结果**：旧技能的正确率是 **40%**； 新技能由于明确指出了"静态数值"的要求，正确率提升到 **55%**。
- **决策**：因为 $0.55 > 0.40$，所以验证通过。 这条针对 Spreadsheet 的修改被正式永久写入技能文档。

Residual Learning

- **Use student learning example**

问题和讨论