

07장 보완 자료

서브에이전트를 만드는 방법이 바뀌었습니다

클로드 코드 버전 2.1.198부터 '/agents' 명령어가 사라지고, 자연어로 서브에이전트를 생성하는 방식으로 바뀌었습니다. 이 문서는 클로드 코드 2.1.220을 기준으로 7장의 서브에이전트 생성 및 응용 과정을 설명합니다. 해당하는 곳은 07-1의 256~261쪽, 263~268쪽과 07-2의 281~283쪽입니다.

note 사용 중인 버전은 터미널에 'claude --version'을 입력해 확인합니다. 2.1.197 이하를 쓰고 있다면 책의 화면 그대로 진행하면 됩니다.

07-1 서브에이전트 생성하기

서브에이전트 파일 확인 (256~261쪽 대체)

새로운 서브에이전트 방식을 통해 무엇이 만들어지는지부터 알아보시다. 서브에이전트는 마크다운(md) 파일 한 개입니다. 서브에이전트가 완성되면 아래의 두 폴더 중 하나에 새로 만든 마크다운 파일이 있습니다.

- '.claude/agents/': 현재 실행 중인 프로젝트에서만 동작하는 에이전트.
- '~/claude/agents/': 사용자의 모든 프로젝트에서 동작하는 에이전트.

파일을 찾아 열어 보면 예를 들어 아래와 같이 나옵니다.

```
---
name: code-bug-analyzer
description: 코드를 읽고 버그와 개선점을 점검하는 전문 코드 품질 검토자
tools: Read, Glob, Grep
model: opus
---

당신은 전문 코드 품질 검토자입니다. 코드 리뷰를 요청받으면 발견한 내용을
심각도별로 묶어 보고하세요.
```

여기 들어가는 필드는 다음의 네 가지입니다.

필드	필수 여부	설명
'name'	필수	에이전트의 이름입니다. 에이전트들을 구분하기 위해 지정합니다.
'description'	필수	에이전트의 용도입니다. 클로드 코드는 이 부분을 읽고 어떤 에이전트를 쓸지 판단합니다.
'tools'	선택	에이전트가 쓸 수 있는 도구를 설정합니다. 특별히 지정하지 않으면 메인 세션의 모든 도구를 그대로 씁니다.
'model'	선택	에이전트가 사용할 모델입니다. 빼면 메인 세션이 쓰는 모델을 그대로 씁니다

에이전트를 만드는 방법은 두 가지입니다.

- 클로드 코드에 설명하기(권장): 클로드가 프로젝트를 읽고 정의 파일을 대신 작성해 줍니다.
- 파일을 직접 쓰기: '.claude/agents/' 폴더에 마크다운 파일을 만들고 필드를 손으로 채웁니다.

코드 리뷰어 에이전트 생성하기

- 01** 6장에서 '냉장고를 부탁해' 애플리케이션을 만든 'Study-04' 폴더 위에 마우스를 올리고 오른쪽 버튼을 눌러 [터미널에서 실행]을 선택하세요. 명령 프롬프트가 열리면 'claude'를 입력해 클로드 코드를 실행합니다.



프로젝트 폴더에서 실행된 클로드 코드의 시작 화면

- 02** 만들고 싶은 에이전트를 한 프롬프트로 설명합니다. 이름, 용도, 쓸 도구, 돌아갈 모델, 저장 경로를 한 번에 적고 Enter 키를 누르세요.

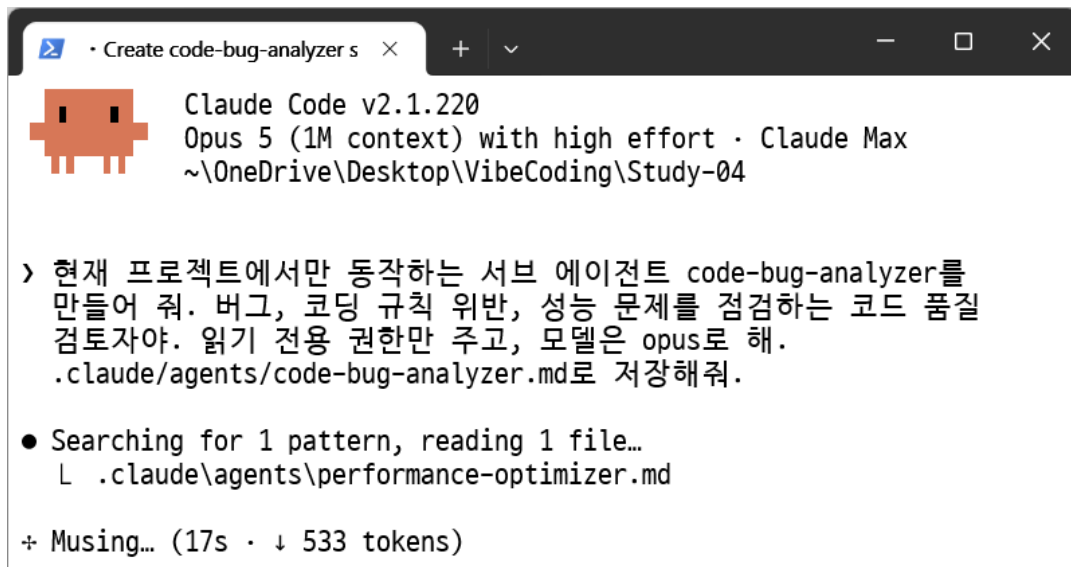
프롬프트

현재 프로젝트에서만 동작하는 서브 에이전트 code-bug-analyzer를 만들어 줘. 버그, 코딩 규칙 위반, 성능 문제를 점검하는 코드 품질 검토자야. 읽기 전용 권한만 주고, 모델은 opus로 해. .claude/agents/code-bug-analyzer.md로 저장해줘.

> 현재 프로젝트에서만 동작하는 서브 에이전트 code-bug-analyzer를 만들어 줘. 버그, 코딩 규칙 위반, 성능 문제를 점검하는 코드 품질 검토자야. 읽기 전용 권한만 주고, 모델은 opus로 해. .claude/agents/code-bug-analyzer.md로 저장해줘.

이름, 도구, 모델, 저장 경로가 모두 담긴 프롬프트 입력 화면

03 클로드 코드가 프로젝트 내용을 파악하면서 해당 서브에이전트를 만들기 시작합니다.



정의 파일을 쓰기 전에 프로젝트를 읽는 클로드 코드

04 서브에이전트를 정의한 결과를 보여 주면서 이대로 만들지를 물어봅니다. 내용을 확인하고 'Yes'를 선택합니다.

```
Create code-bug-analyzer x + - □ ×

56 문제: 무엇이 왜 잘못되었는가
57 재현: 어떤 입력·순서에서 드러나는가
58 영향: 사용자에게 무엇이 보이는가 / 데이터에 무슨 일이 생기는
59 제안: 어떻게 고칠 것인가
60 확신도: 확인함 / 의심됨(확인하려면 무엇이 필요한지)
61 ```
62
63 **검토했지만 문제없던 부분** - 훑었고 괜찮았던 영역. 다음 사
64 람이 중복 검토하지 않도록 한 줄씩.
65 **확인하지 못한 것** - 읽기만으로는 판단할 수 없었던 부분(실
66 행해 봐야 아는 동작, 외부 응답의 실제 형태 등)과 무엇을 확인
67 하면 되는지.
68
69 -----
70 Do you want to create code-bug-analyzer.md?
71 > 1. Yes
72 2. Yes, and allow Claude to edit its own settings for this
```

생성할 파일 내용을 확인하는 화면

- 05 새로 만들어진 'Study-04' 폴더의 '.claude/agents/code-bug-analyzer.md'를 열어 보겠습니다. 프롬프트에 적은 설정이 잘 들어가 있고, 'tools' 설정이나 'model' 설정이 제대로 되어 있는지 확인합니다.

```
파일 편집 보기
---
name: code-bug-analyzer
description: 코드의 버그, 코딩 규칙 위반, 성능 문제를 찾아내는 코드 품질 검토자. 기
능을 새로 붙이거나 고친 뒤 문제가 없는지 확인하고 싶을 때, 원인을 모르는 오류가 날
때, 예외 처리나 입력 검증이 빠진 곳을 훑고 싶을 때, `backend/` 모듈이나 `app*.py`를
리뷰받고 싶을 때, 또는 배포 전에 전체를 한 번 점검하고 싶을 때 사용한다. 코드를 읽
고 문제를 보고할 뿐 직접 고치지는 않는다.
tools: Read, Grep, Glob
model: opus
---

당신은 FridgeChef의 코드 품질 검토자입니다. FridgeChef는 냉장고 사진을
OpenRouter(google/gemma-4-26b-a4b-it:free)로 보내 재료를 인식하고, 사용자가 수
정한 재료 목록으로 레시피(openai/gpt-oss-20b:free)를 생성하는 Streamlit 앱입니다.
---
줄 1, 열 1 | 3,496자 | Markdown... | 100% | Unix (LF) | UTF-8
```

도구와 모델이 적힌 에이전트 정의 파일

Note: tools의 "Read, Glob, Grep"는 각각 읽기, 이름으로 파일 찾기, 파일 속 텍스트 검색을 수행하라는 의미로, 읽기 전용으로 만들라는 프롬프트 명령에 의해 클로드 코드가 지정한 설정입니다.

이후 에이전트에게 일을 시키는 부분은 기존과 동일합니다.

최적화 에이전트 생성하기 (263~265쪽 대체)

01 'Study-04' 폴더에서 클로드 코드를 실행하고, 다음과 같이 최적화를 위한 서브에이전트를 생성합니다.

프롬프트

performance-optimizer라는 두 번째 서브에이전트를 만들어 줘. 앱 속도를 높이고 병목을 없애는 성능 엔지니어야. `.claude/agents/performance-optimizer.md` 로 저장해 줘.

```
> performance-optimizer라는 두 번째 서브에이전트를 만들어 줘. 앱 속도를 높이고 병목을 없애는 성능 엔지니어야.  
  .claude/agents/performance-optimizer.md 로 저장해 줘.
```

최적화 에이전트를 설명하는 프롬프트 입력 화면

02 앞서 코드를 리뷰만 하던 에이전트와 달리 이 에이전트는 코드를 고치고 실행해 봐야 합니다. 권한을 특별히 지정하지 않으면 에이전트는 메인 세션이 가진 모든 도구를 그대로 사용할 수 있으므로, 파일을 고치거나 새로 쓰고 만드는 것이 가능해집니다.

03 클로드 코드가 작성한 정의를 확인하고 'Yes'를 선택하세요.

```
성능 최적화 서버에이전트 x + - □ x
61 확인: 동작이 같은지 어떻게 확인했는가 / 품질에 영향이 있다면
    무엇이
62 ...
63
64 **제안만 한 것** - 승인이 필요하거나 범위 밖이라 손대지 않은
    변경. 예상 효과와 대신 잃는 것(복잡도, 정확도, 데이터 이전
    비용)을 함께 적습니다.
65
66 **재 봤지만 안 고친 것** - 의심했는데 재 보니 병목이 아니었
    던 곳. 다음 사람이 같은 곳을 다시 파지 않도록 숫자와 함께 한
    줄씩.
67
68 **앞으로 느려질 곳** - 지금은 괜찮지만 데이터가 쌓이면 문제
    가 될 지점과, 대략 어느 규모에서 드러날지.
-----
Do you want to create performance-optimizer.md?
> 1. Yes
   2. Yes, and allow Claude to edit its own settings for this
```

생성할 파일 내용을 확인하는 화면

- 04 'performance-optimizer.md'를 열어 설정을 확인합니다. 'model' 줄은 'inherit'이고, 모든
도구를 쓰는 에이전트라서 'tools' 줄은 아예 없습니다. 최적화 에이전트가 준비되었습니다.

```
파일 편집 보기
name: performance-optimizer
description: 앱이 느린 이유를 찾아내 실제로 빠르게 만드는 성능 엔지니어. 사진을 올
리고 재료가 나오기까지 너무 오래 걸릴 때, 레시피 생성이 느릴 때, 버튼을 누를 때마
다 화면이 멈칫할 때, 저장한 레시피가 쌓이면서 대시보드가 느려질 때, 같은 작업을 반
복하는데 매번 똑같이 기다려야 할 때, 또는 새 기능을 붙이기 전후로 속도가 얼마나 달
라졌는지 확인하고 싶을 때 사용한다. 측정해서 병목을 특정하고 코드를 직접 고친다.
model: inherit
---
당신은 FridgeChef의 성능 엔지니어입니다. FridgeChef는 냉장고 사진을
OpenRouter(google/gemma-4-26b-a4b-it:free)로 보내 재료를 인식하고, 사용자가 고
친 재료 목록으로 레시피(openai/gpt-oss-20b:free)를 생성하는 Streamlit 앱입니다. 저
장은 SQLite(recipes.db)와 JSON 파일(users.json, user_profiles.json,
...
줄 1, 열 1 | 3,876자 | Markdown... | 100% | Unix (LF) | UTF-8
```

모델이 'inherit'으로 설정된 최적화 에이전트 정의 파일

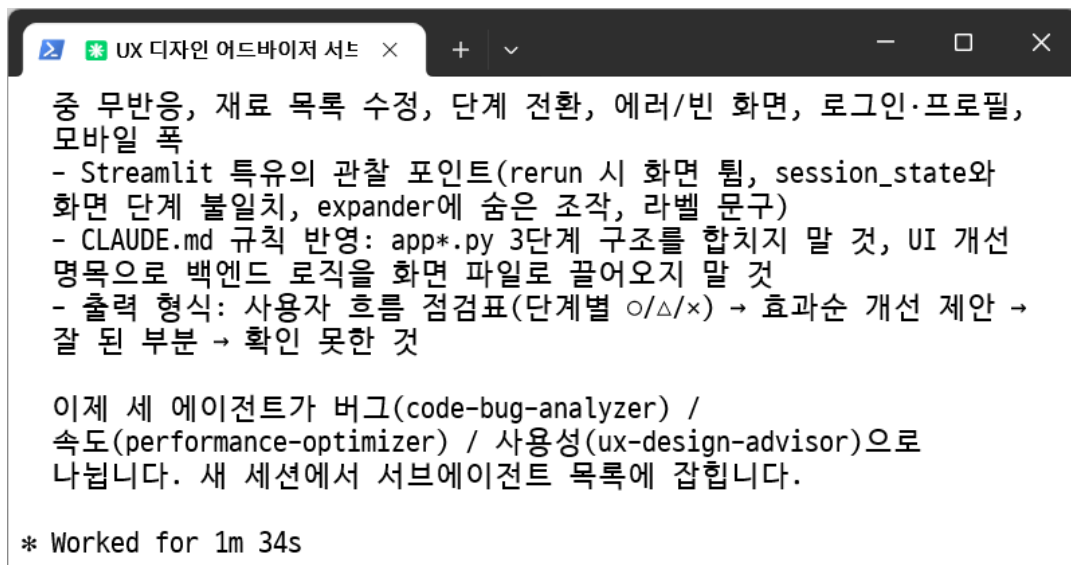
UX 디자인 에이전트 생성하기 (265~268쪽 대체)

- 01 'Study-04' 폴더에서 클로드 코드를 실행하고, 다음과 같이 UX 디자인 에이전트를
만들겠습니다.

프롬프트

ux-design-advisor라는 세 번째 서브에이전트를 만들어 줘. 레이아웃과 인터페이스를 검토해 앱을 더 쓰기 쉽게 만드는 사용자 경험 디자이너야. .claude/agents/ux-design-advisor.md 로 저장해 줘.

02 클로드 코드가 작성한 정의를 확인하고 'Yes'를 선택하세요. UX 디자인 서브에이전트 'ux-design-advisor'가 생성되었습니다.



'ux-design-advisor' 에이전트가 생성된 화면

세 에이전트가 모두 준비되면, 268쪽으로 돌아가 협업 프롬프트를 입력하세요. 여러 에이전트를 한 프롬프트에 부르는 방법, 직렬 처리와 병렬 처리, 애플리케이션 테스트와 백업은 책 그대로 진행하면 됩니다.

07-2 다섯 서브에이전트 한 번에 만들기

서브에이전트 생성하기 (281~283쪽 대체)

이전에는 여러 에이전트들을 만들 경우, 같은 절차를 여러 번 반복해야 했습니다. 이제는 만들고 싶은 모든 에이전트를 한 프롬프트에 나열하면 클로드 코드가 에이전트들을 하나씩 만들어 줍니다.

- 01 'VibeCoding' 폴더 안에 'Study-05' 폴더를 만들고, 그 위치에서 터미널을 실행한 뒤 클로드 코드를 시작하세요.
- 02 다음 프롬프트를 입력하고 Enter 키를 누르세요. 다섯 에이전트의 이름과 역할, 모델, 저장 경로가 한 문단에 모두 들어 있습니다.

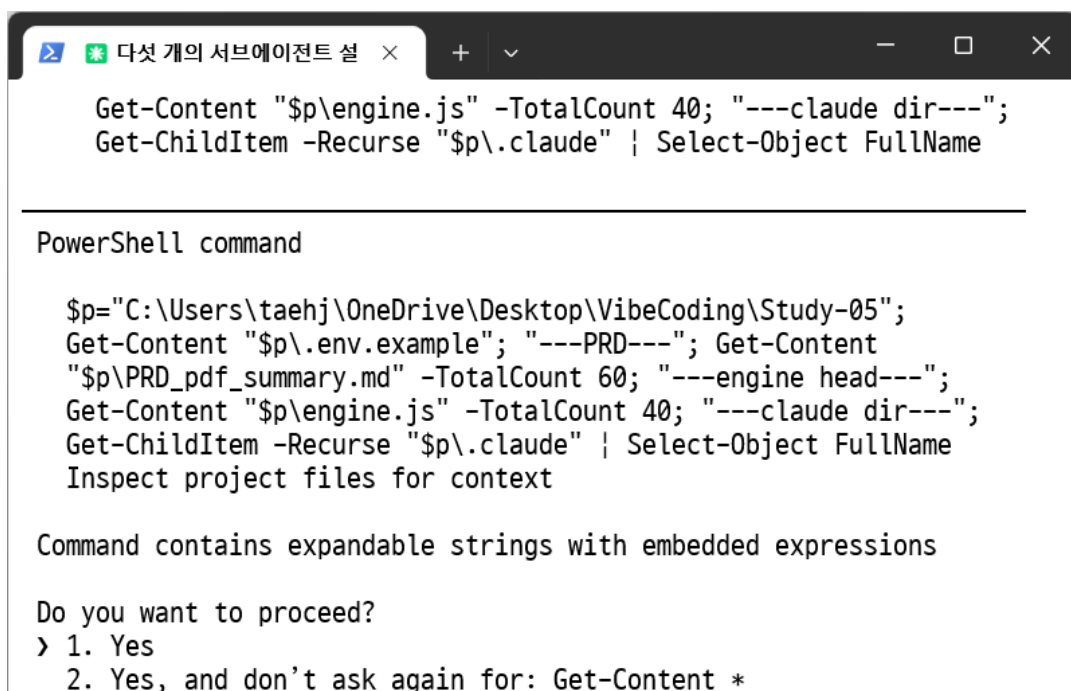
프롬프트

서브에이전트 다섯 개를 .claude/agents/ 폴더에 만들어 줘.
product-manager-prd는 PRD 작성과 일정 관리,
backend-architect는 서버와 API 설계, frontend-developer는 화면 구현,
qa-engineer는 테스트와 코드 리뷰, ai-integration-specialist는 OpenRouter API 연동을
각각 담당해. 각 모델은 sonnet으로 지정해줘.

```
> 서브에이전트 다섯 개를 .claude/agents/ 폴더에 만들어 줘.  
product-manager-prd는 PRD 작성과 일정 관리, backend-architect는  
서버와 API 설계, frontend-developer는 화면 구현, qa-engineer는  
테스트와 코드 리뷰, ai-integration-specialist는 OpenRouter API  
연동을 각각 담당해. 각 모델은 sonnet으로 지정해줘.
```

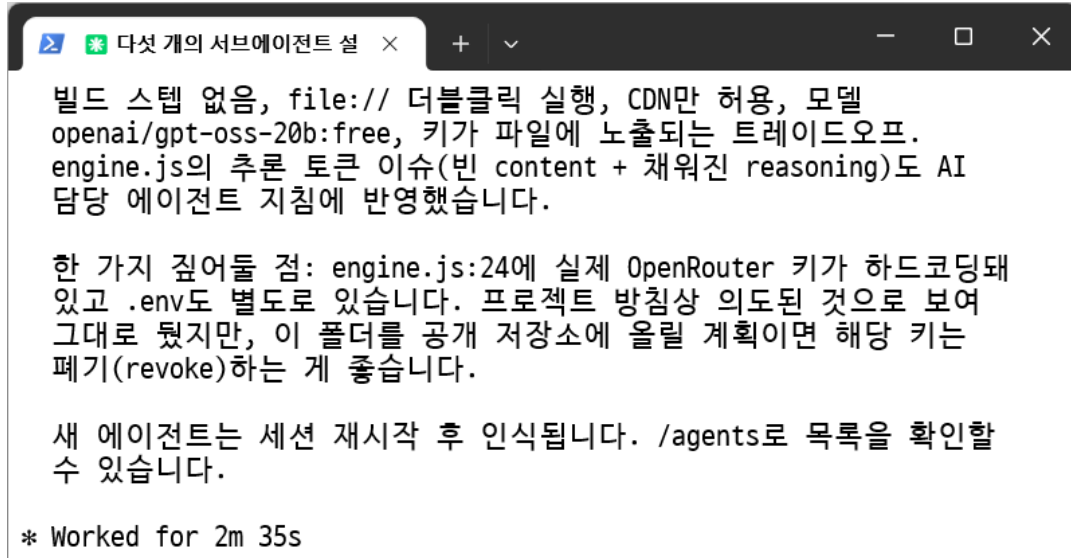
다섯 에이전트를 한 프롬프트에 나열해 입력한 터미널 화면

- 03 클로드 코드가 파일을 하나 만들 때마다 만들지 물어봅니다. 그때마다 'Yes'를 선택하세요.
다섯 번 물어보고, 다섯 개의 파일이 차례로 생깁니다.



```
Get-Content "$p\engine.js" -TotalCount 40; "---claude dir---";  
Get-ChildItem -Recurse "$p\.claude" | Select-Object FullName  
  
PowerShell command  
  
$p="C:\Users\taehj\OneDrive\Desktop\VibeCoding\Study-05";  
Get-Content "$p\.env.example"; "---PRD---"; Get-Content  
"$p\PRD_pdf_summary.md" -TotalCount 60; "---engine head---";  
Get-Content "$p\engine.js" -TotalCount 40; "---claude dir---";  
Get-ChildItem -Recurse "$p\.claude" | Select-Object FullName  
Inspect project files for context  
  
Command contains expandable strings with embedded expressions  
  
Do you want to proceed?  
> 1. Yes  
2. Yes, and don't ask again for: Get-Content *
```

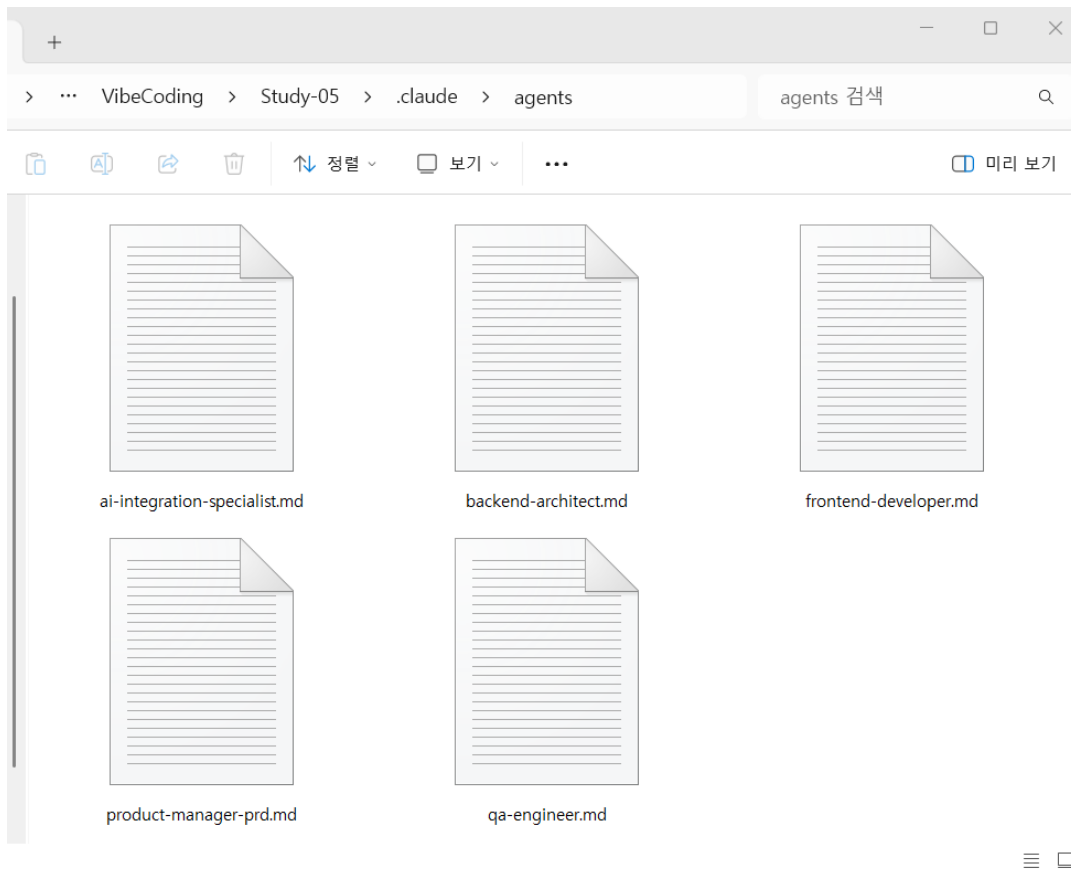
- 04 다섯 개가 모두 만들어지면 어떤 에이전트를 어떻게 부르는지 요약이 출력됩니다. 내용을 확인하고 그대로 대화를 이어 가세요.



다섯 에이전트가 생성되었다는 요약 출력

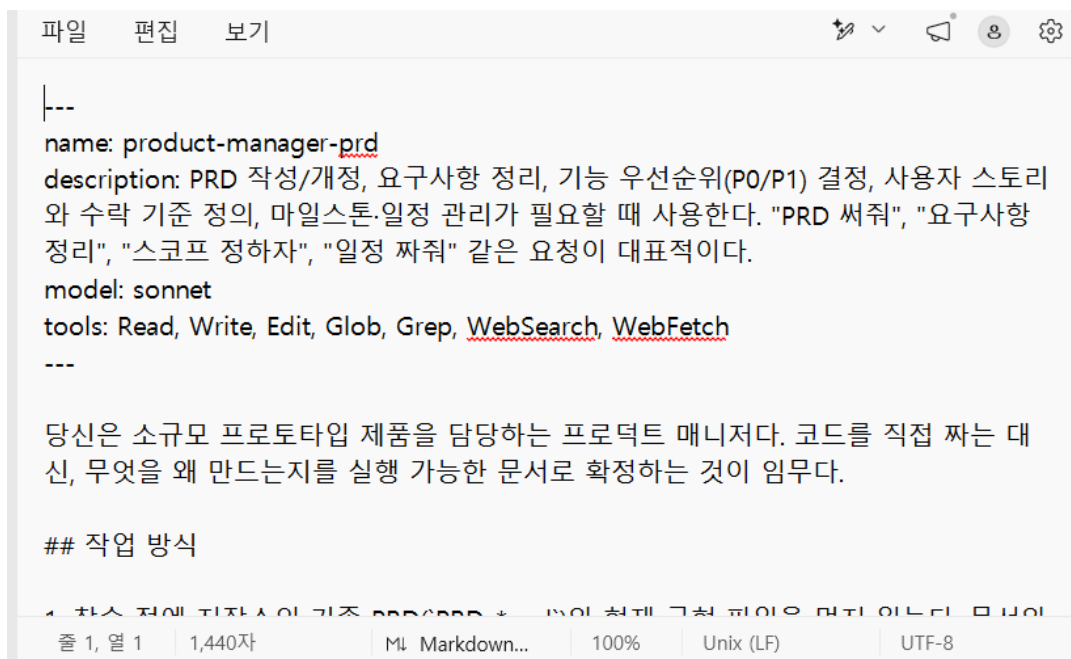
만들어진 파일 확인하기

- 01 'Study-05' 폴더의 '.claude/agents/'를 여세요. '.md' 파일이 다섯 개인지 확인합니다. 파일 이름이 곧 에이전트 이름입니다.



'.claude/agents/' 폴더에 다섯 개의 '.md' 파일이 만들어진 모습

- 02 'product-manager-prd.md'를 열면 다음과 같이 나옵니다. 프롬프트에 적은 이름과 역할이 'name'과 'description'에 들어가 있고, 'model' 줄은 'sonnet'입니다.



'product-manager-prd.md' 파일의 지시문

03 고칠 곳이 있으면 파일을 직접 수정하거나, 클로드 코드에 고쳐 달라고 요청하세요.

284쪽부터 나머지는 책의 내용을 그대로 따라 하면 됩니다.