GitHub

GitHub 웨비나

GitHub Copilot
코딩 에이전트와 자동 코드 리뷰

August 2025

오늘의 아젠다

에이전트를 사용한 개발과 코드 리뷰

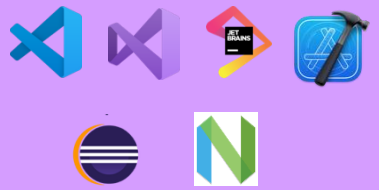
- GitHub Coding Agent
- Agent Mode in IDE
- Copilot Code Review
- Demo
- Custom-instructions.md
- 커스텀 챗 모드
- Prompt-files.md

Q&A

“에이전트” develops & 코드리뷰

Coding Agent

Agent mode



Copilot 코드리뷰

Copilot coding agent

Requires Premium Model

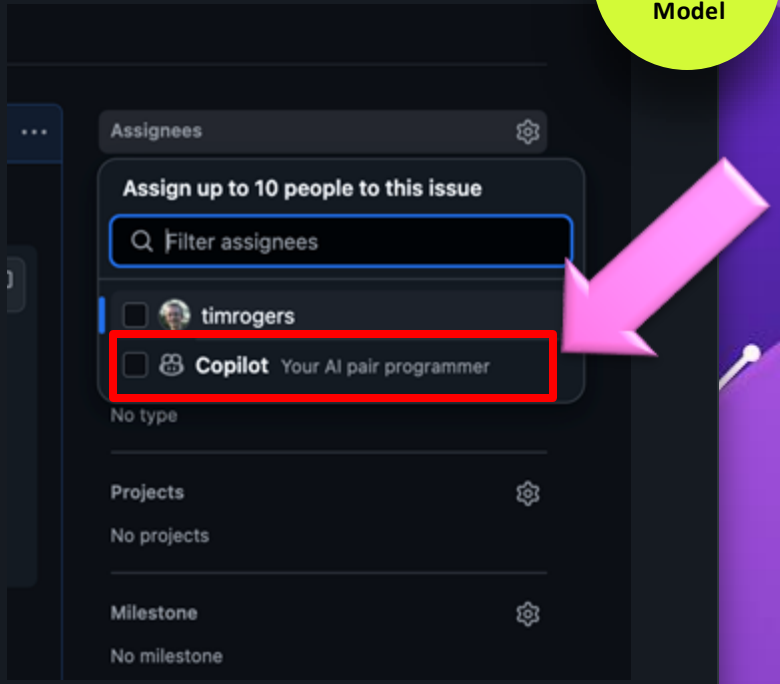
개발 업무를 Copilot 코딩 에이전트에게 할당

개발자는 더 창의적이고 복잡한 중요 업무에 집중

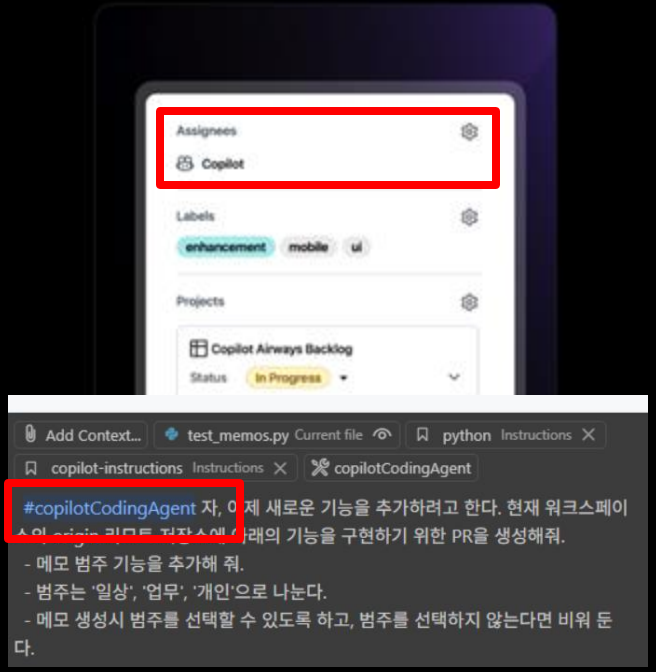
GitHub의 Issue에서 할당하거나, IDE상의 Copilot Chat에서 코딩 에이전트에게 업무 할당

Copilot이 PR리뷰를 통해 수정사항에 대해 반복하여 코딩 에이전트와 개발

*프리미엄 모델 사용

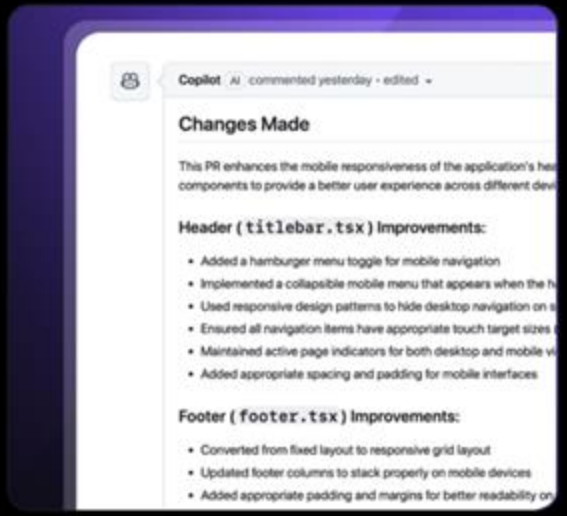


Agent가 개발 -> PR 생성 -> Copilot 리뷰



Copilot Chat 또는 issue에서
'Copilot'에게 할당

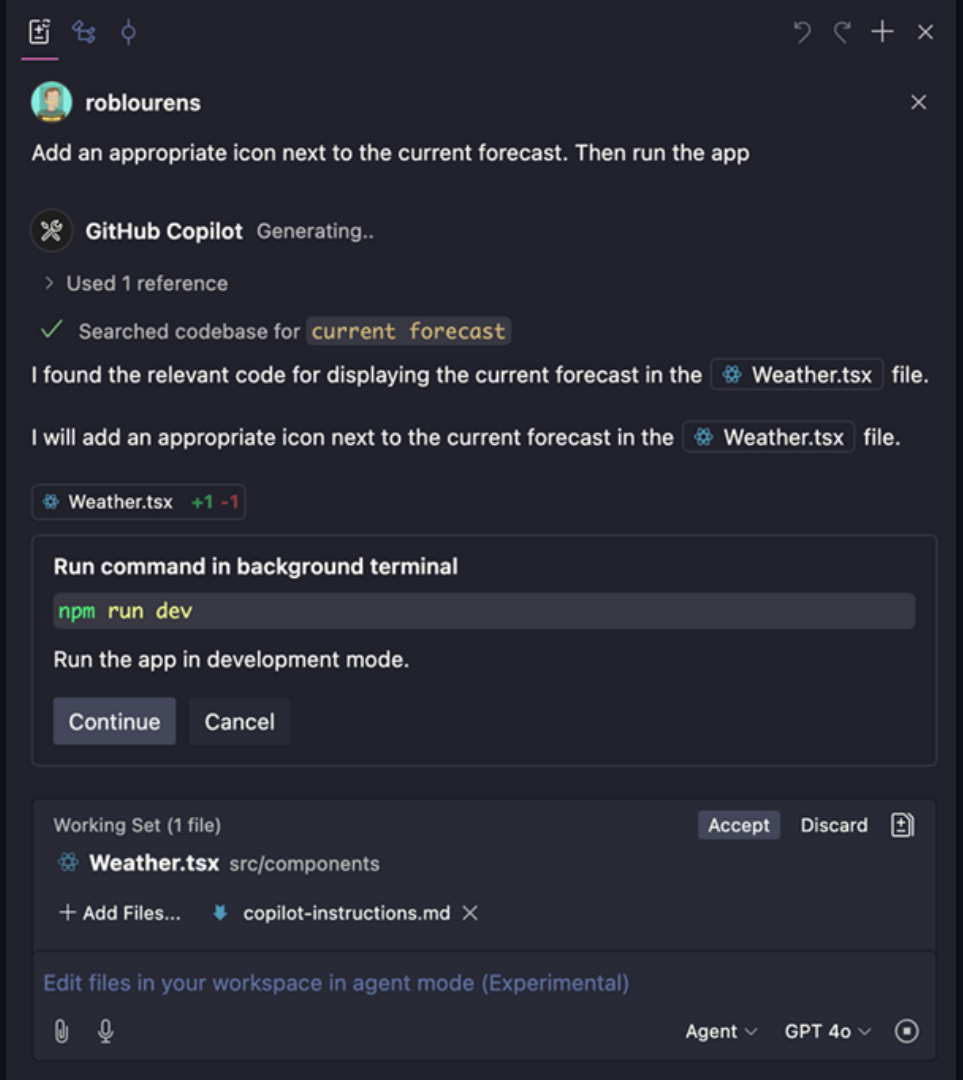
Copilot 이 개발 후 PR을 생성하고,
사용자에게 리뷰 요청



PR comment를 통해 Copilot에게
리뷰된 내용에 대해 개발하도록 요청

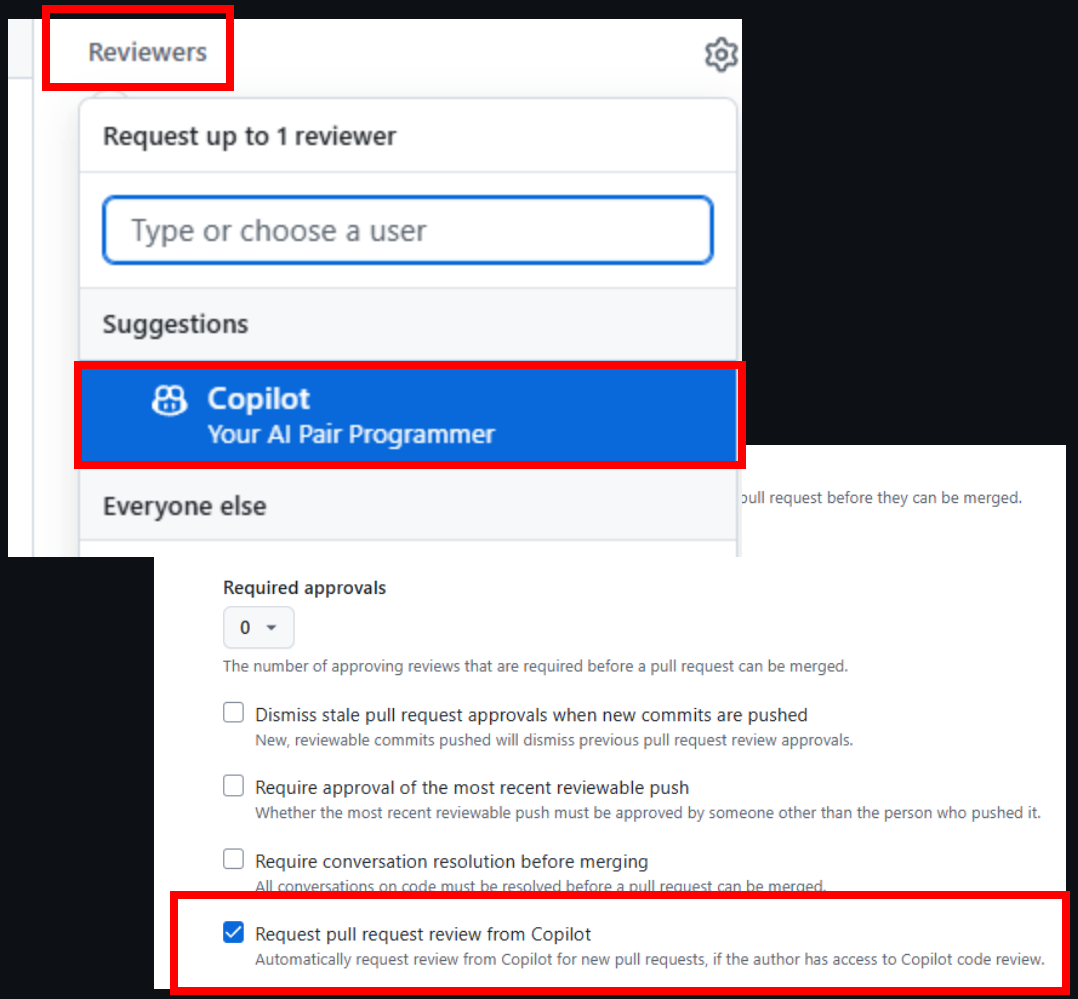
Agent mode

- Copilot이 자율적으로, 스스로 반복하여 주어진 task를 완성.
- 자동으로 문제를 찾아 수정하고,
- 터미널 커맨드를 실행하면서, 스스로 런타임 에러를 해결하는 셀프 힐링을 통해 요구되는 태스크를 완성

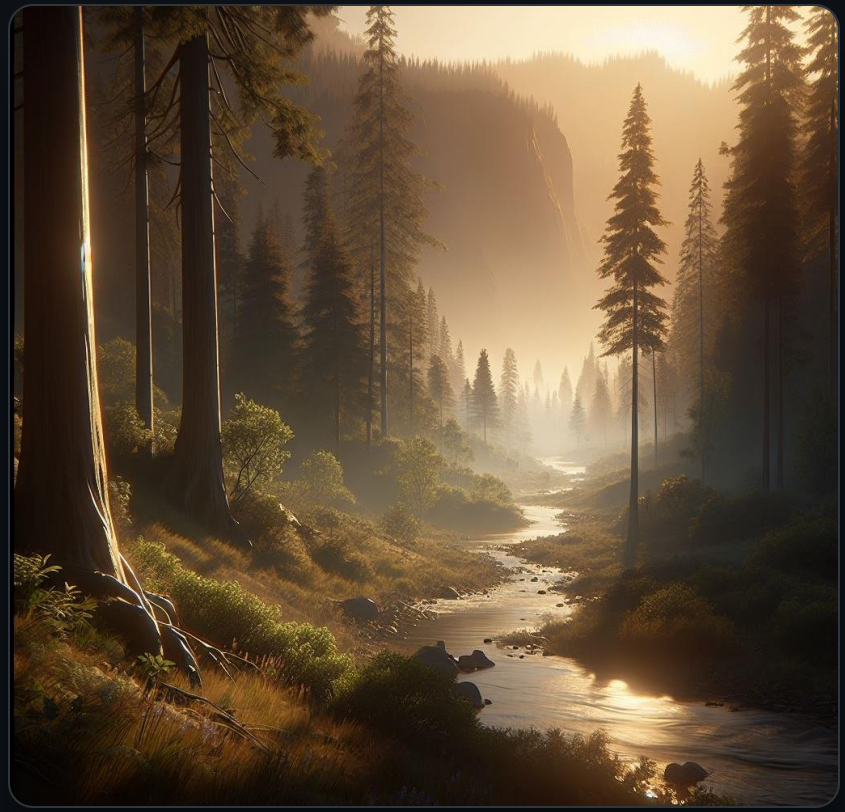


Copilot code review

- 생성된 PR에 대해 Copilot에게 리뷰 요청
- Rule 구성을 통해, 자동으로 PR에 대해 Copilot에게 리뷰 되도록 구성
- 저장소의 copilot-instructions.md를 기반으로 코드 리뷰



Demo

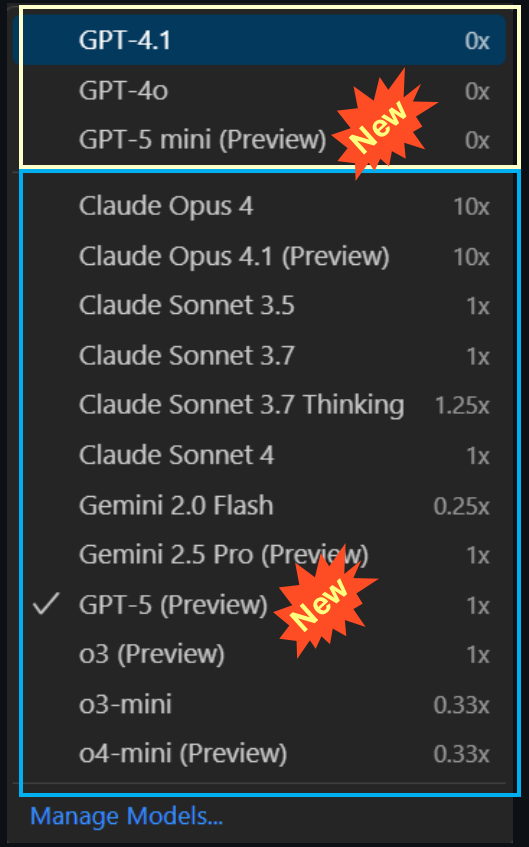


멀티 모델 사용

- Standard 모델
  - GPT-5 mini (Preview)
  - GPT-4.1, GPT-4o
- Premium 모델
  - o3, o3-mini, o4-mini
  - Gemini 2.0 Flash, Gemini 2.5 Pro
  - Claude 3.5 / 3.7 Sonnet
  - Claude 3.7 Sonnet Thinking
  - Claude 4 Sonnet
  - Claude 4 Opus
  - **GPT-5** (New)

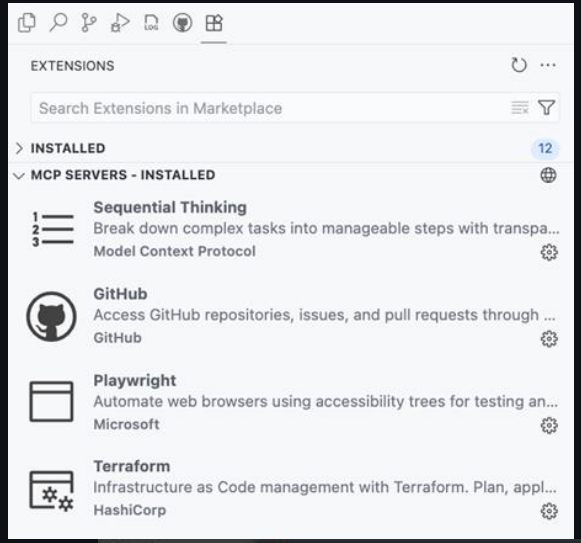
Standard 모델
: 무제한 제공

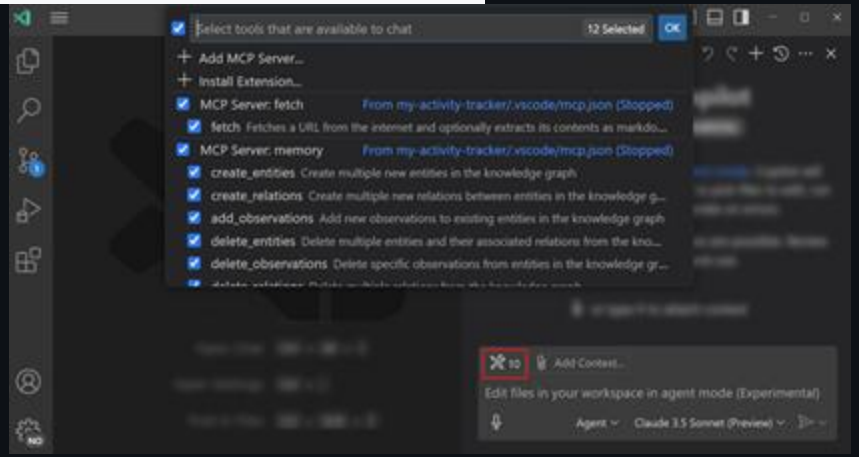
Premium 모델
: 라이선스에 따른 기본 제공
(300회/1000회)



MCP

- Model Context Protocol (MCP) server 지원 기능 포함되어 많은 특화된 도구들 선택하여 추가 가능
- Tools, Resources, Prompts, Sampling 지원

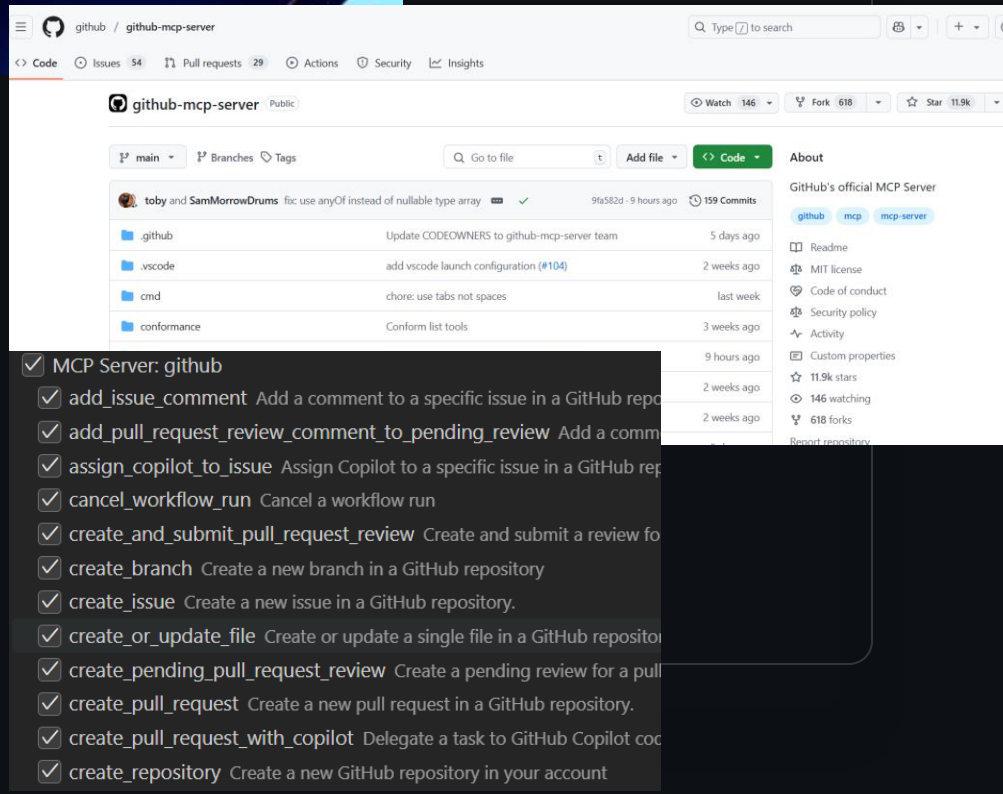




GitHub MCP Server

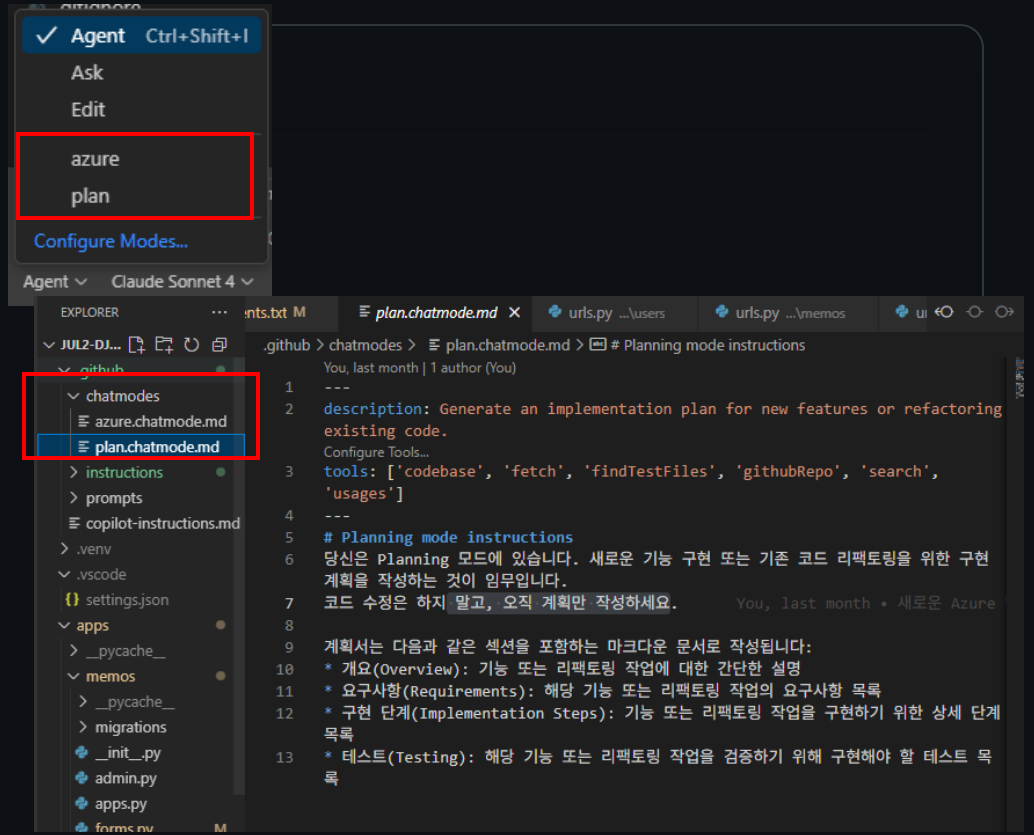
- GitHub 공식 MCP 서버
- 리모트 (HTTP)로 등록, 혹은 사용자 로컬에 도커로 실행
- MCP(Model Context Protocol) 도구는 LLM에 함수를 호출하고, 데이터를 조회하고, 외부와 상호 작용할 수 있는 표준화된 방법을 제공합니다.

https://github.com/github/github-mcp-server



커스텀 Chat 모드

- .chatmode.md 라는 마크다운 파일로 정의
- 사용 목적에 따른 커스텀 챗 모드 설정
- 관련 도구와 지침을 매번 수동으로 선택하지 않고도 해당 구성으로 빠르게 전환



copilot-Instructions.md

Copilot에게 코드 제안에 대한 규칙을 설정

.github/copilot-instructions.md

Copilot이 코드 제안시에 이 규칙을 따라 코드 제안

```
# 메모짱
Django 기반의 메모장 웹 애플리케이션

주요 기능
- 사용자 로그인 및 회원가입.
- 메모 작성, 수정, 삭제.
- 메모 목록 조회.


# 구성요소 Stack

- 프론트엔드
  - Django 템플릿 엔진: HTML, CSS, JavaScript를 사용하여 간단한 UI 구현.
  - Bootstrap (선택 사항): 빠르고 반응형 UI를 위해 사용.

- 백엔드
  - Django Framework: 웹 애플리케이션의 핵심 로직과 API 제공.
  - Django Forms: 메모 작성 및 수정 폼 처리.

- 데이터베이스
  - SQLite: 개발 단계에서 간단히 사용할 수 있는 기본 데이터베이스.

- 사용자 인증
  - Django 내장 인증 시스템: 사용자 로그인, 로그아웃, 회원가입 기

- 세션 관리
  - Django의 기본 세션 관리 기능 사용.

- 배포
  - 개발 단계: Django의 내장 개발 서버 사용.
  - 프로덕션: Gunicorn + Nginx (선택 사항).

- 추가 라이브러리
  - django-crispy-forms (선택 사항): 폼 스타일링을 간편하게 하
```

```
- 디렉토리 구조
- .github/
- docs/
- memojjang/                    # 메인 Django 프로젝트
- apps/                         # Django 앱들
  - __init__.py
  - __pycache__/
  - memos/                      # 메모 앱
    - __init__.py
    - admin.py                  # 관리자 설정
    - apps.py                   # 앱 설정
    - models.py                 # 데이터 모델
    - views.py                  # 뷰 로직
    - tests.py                  # 테스트
    - migrations/               # 데이터베이스 마이그레이션
    - __pycache__/
  - users/                      # 사용자 앱
    - __init__.py
    - admin.py                  # 관리자 설정
```

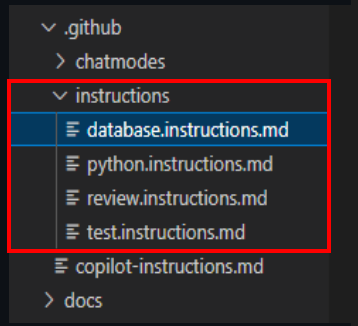
copilot-Instructions.md 작성법/권고

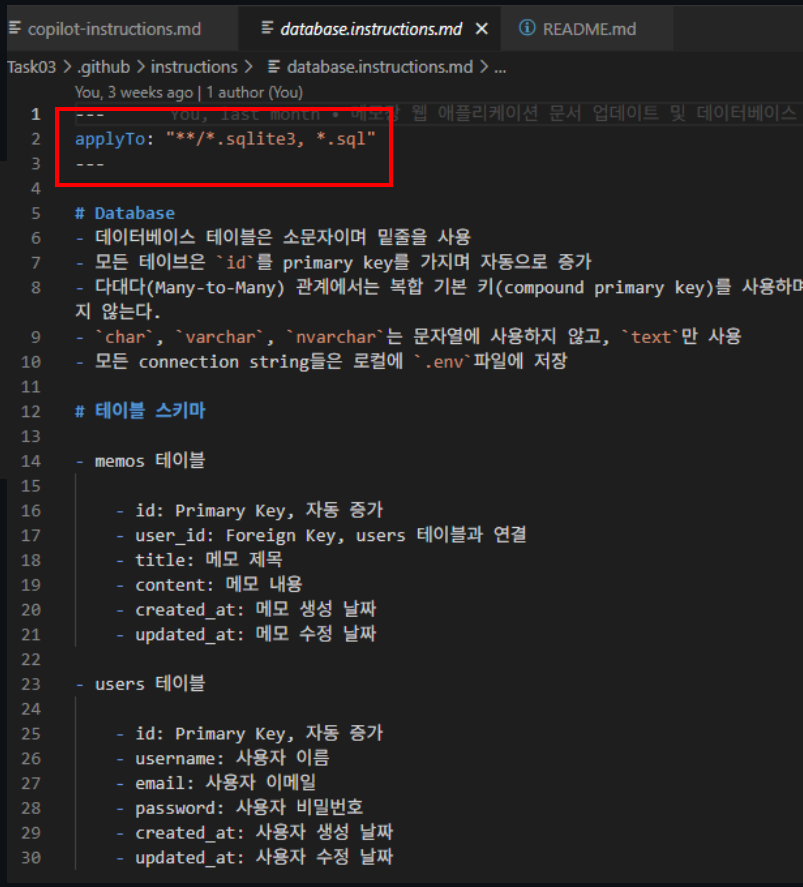
- 프로젝트의 목적, 기능에 대한 상위 수준의 정의
- 기술 스택, 사용되는 (선호하는) 라이브러리와 프레임 워크 and frameworks
- 프로젝트 구조 및 파일 구성
- 코드 스타일 및 컨벤션
- 주석 및 문서화 스타일
- 디자인 패턴 및 아키텍처 원칙
- 테스트 전략 및 프레임워크
- 보안 및 성능 고려 사항

세부적인 Instructions 설정

.instructions.md 파일로 설정
세부적인 사용자 지침 설정 파일

- 테스트 코드 생성 instructions
- 코드 리뷰 instructions
- 커밋 메시지 생성 instructions
- Pull request 제목, 설명 생성 instructions





Prompt instructions 설정 (VS Code)

Copilot에게 재사용 가능한 프롬프트 instruction 파일(.md)을 제공

- 코드 생성: 컴포넌트, 테스트, migration등에 대한 재사용 프롬프트 제공(예: React forms, API mocks)
- 도메인 전문지식 : 특정 도메인에 대한 전문적인 정보 제공 (예: 보안 프랙티스, 규정 준수확인 등)
- 팀 협업: 문서 패턴과 가이드라인 제공 (참조문서와 스펙제공)
- 온보딩: 복잡한 절차, 혹은 프로젝트의 특정한 패턴들에 대한 step-by-step 가이드

react-form.prompt.md

```
Your goal is to generate a new React form component.

Ask for the form name and fields if not provided.

Requirements for the form:
* Use form design system components: [design-system/Form.md](../docs/design-system/Form.md)
* Use `react-hook-form` for form state management:
* Always define TypeScript types for your form data
* Prefer *uncontrolled* components using register
* Use `defaultValues` to prevent unnecessary rerenders
* Use `yup` for validation:
* Create reusable validation schemas in separate files
* Use TypeScript types to ensure type safety
* Customize UX-friendly validation rules
```

Copy

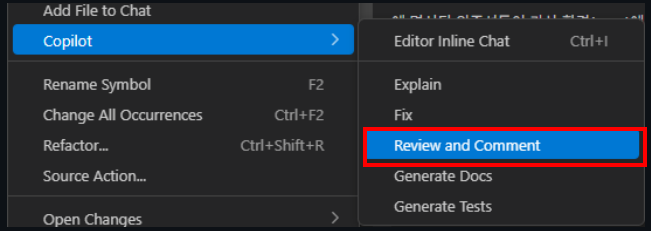
security-api.prompt.md

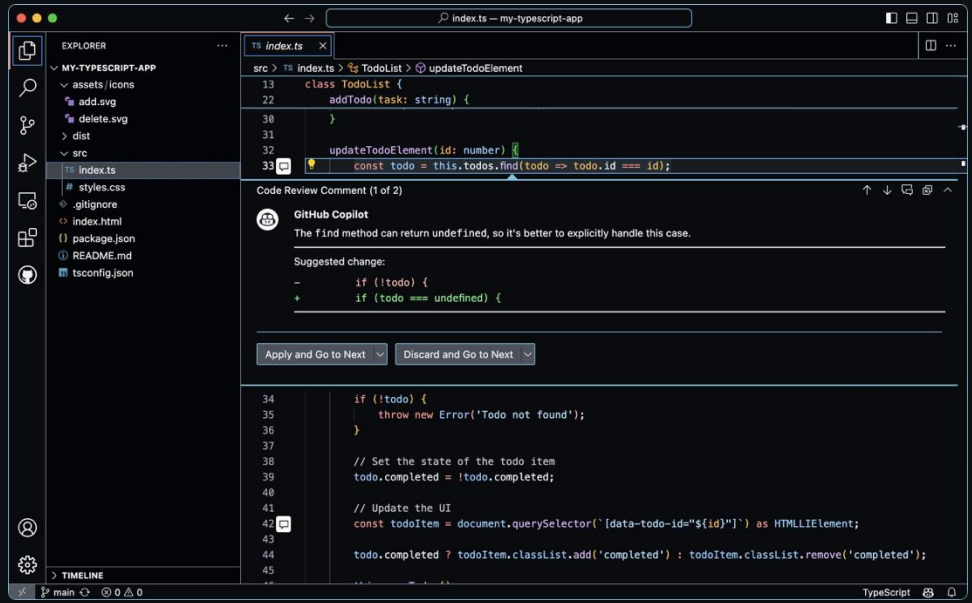
```
Secure REST API review:
* Ensure all endpoints are protected by authentication and authorization
* Validate all user inputs and sanitize data
* Implement rate limiting and throttling
* Implement logging and monitoring for security events
```

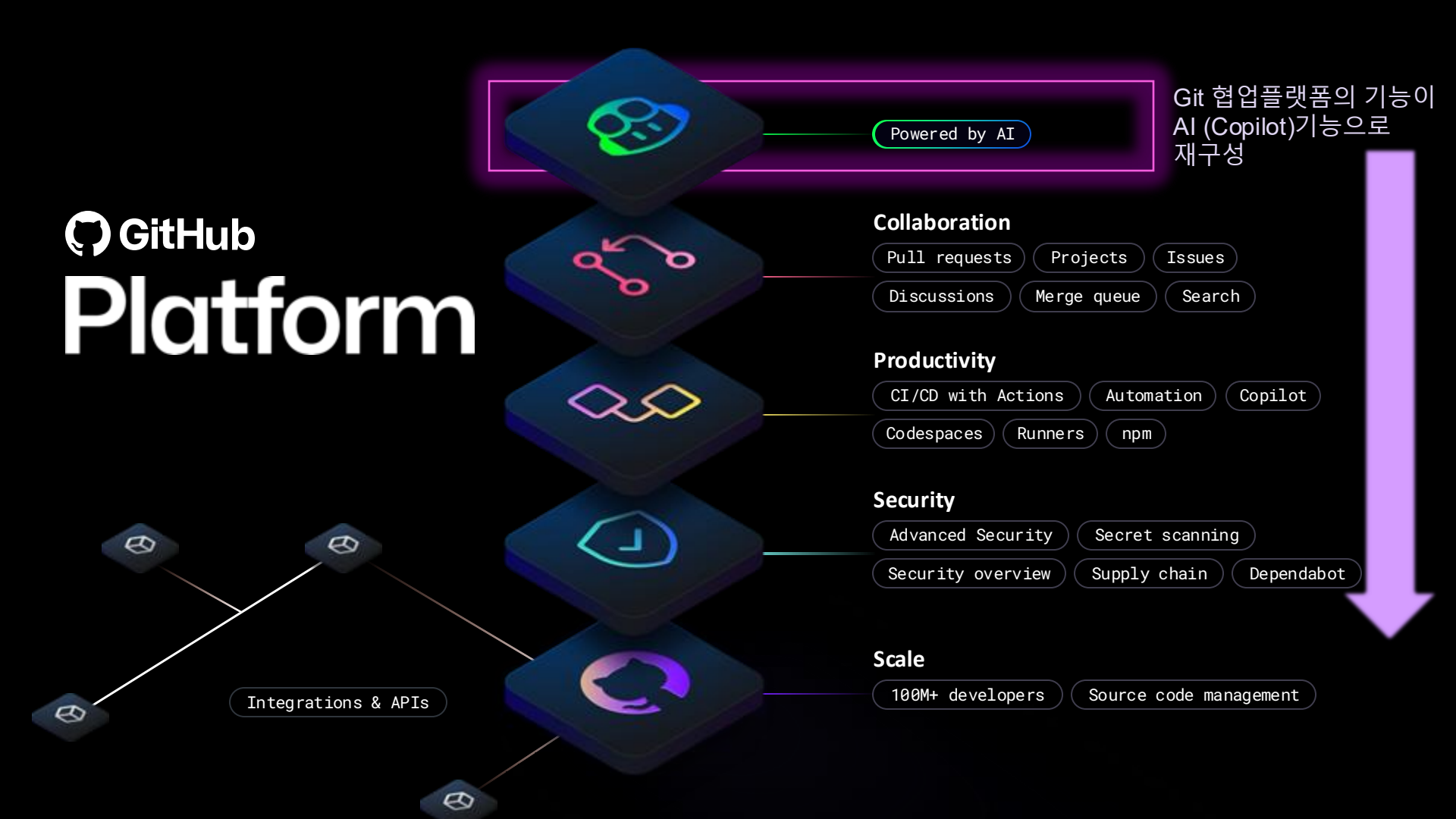
Copy

VS Code 에서 Copilot 코드 피드백

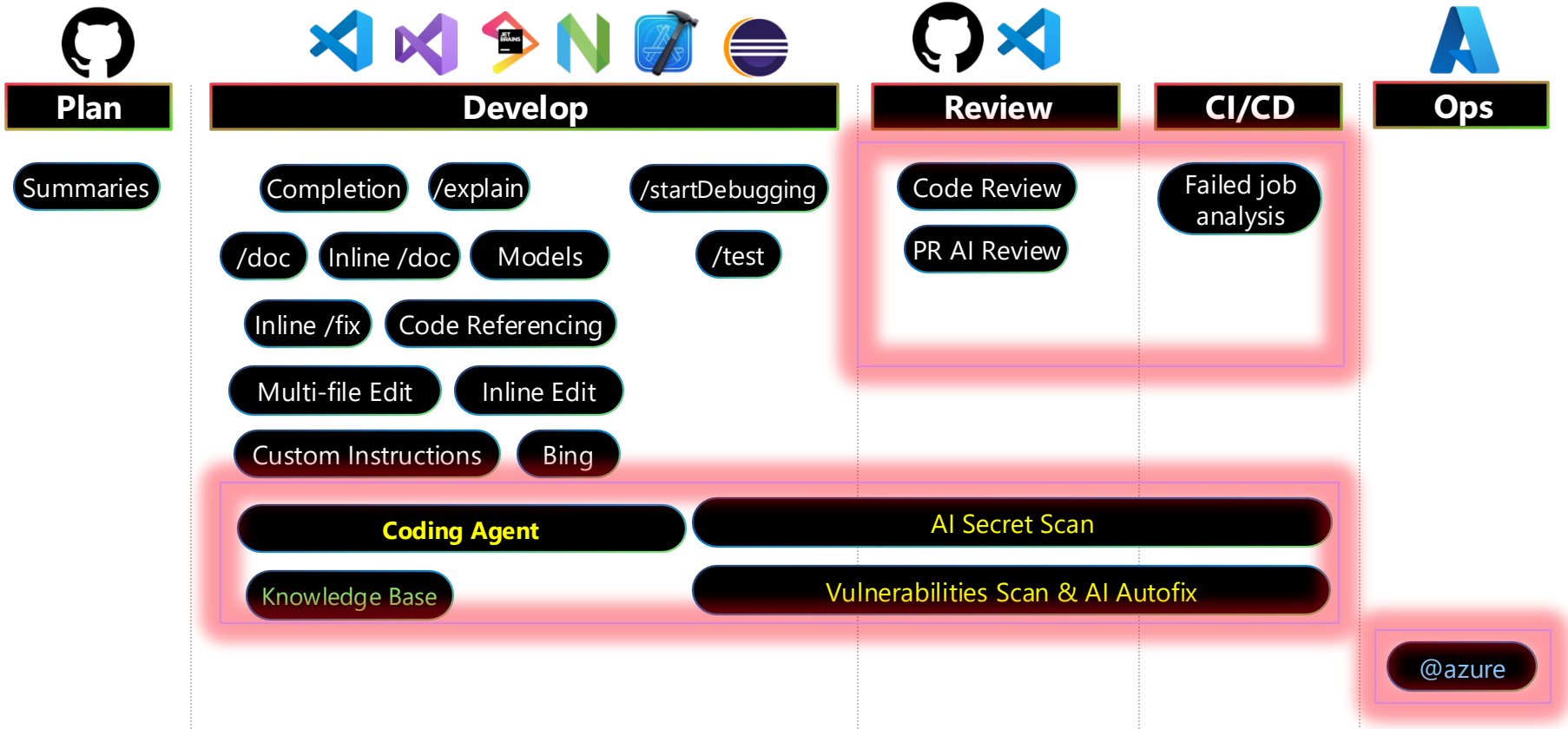
- 코딩 중 IDE에서 사용자 단위로 AI 에서 피드백을 제공
- VS Code의 오른마우스 클릭 – "Review and Comment"
- 리뷰에 대한 "사용자 지침" 설정 가능
- 에디터에서 직접 개발 작업을 가속화하고 코드의 품질을 향상






GitHub
Platform
Powered by AI
Collaboration
Pull requests
Projects
Issues
Discussions
Merge queue
Search
Productivity
CI/CD with Actions
Automation
Copilot
Codespaces
Runners
npm
Security
Advanced Security
Secret scanning
Security overview
Supply chain
Dependabot
Scale
100M+ developers
Source code management
Integrations & APIs
Git 협업플랫폼의 기능이
AI (Copilot)기능으로
재구성

Copilot in DevOps Lifecycle



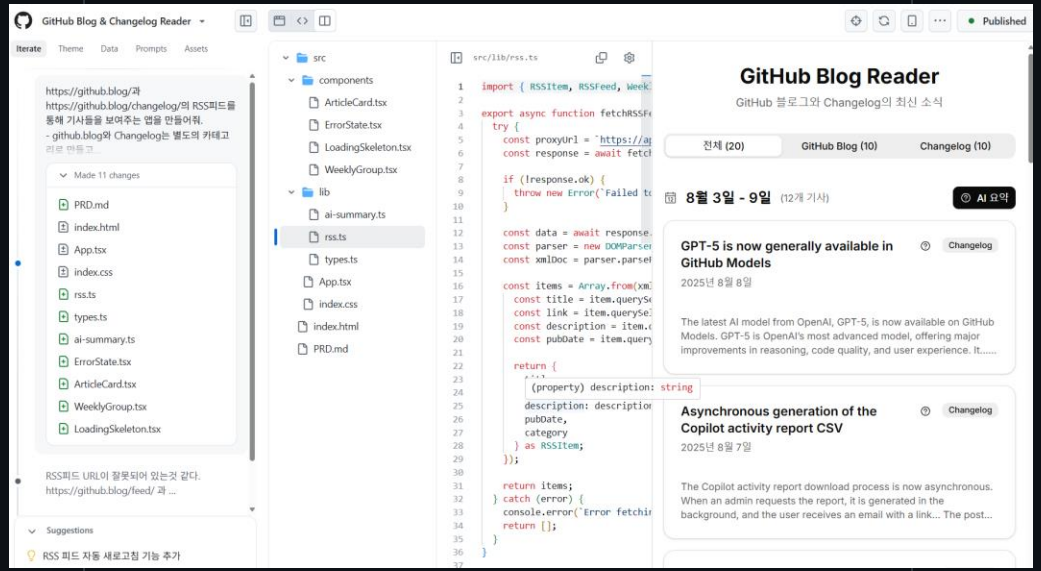
GitHub Copilot / Code Generation

	Completions	Chat/Edits	Agent Mode	Copilot coding agent
Scope Of Changes	Next Few Lines	Multi-File Edits	Complete tasks	Entire issues
Frequency of Interaction	Hundreds of Milliseconds	Seconds	Minutes	Tens of minutes
Inner/Outer Loop	Inner Loop	Inner Loop	Inner Loop	Outer Loop
Developer Canvas	VS Code (Editor)	VS Code (Chat)	VS Code (Chat)	GitHub.com

Public preview (Copilot Pro+)

GitHub Spark

- 자연어를 앱으로 완성
  - Claude Sonnet4 기반
  - 아이디어를 신속하게 앱으로 빌드
- 다양한 AI모델을 활용하는 AI앱 생성
- Data, LLM인터페이스, 호스팅, 배포, GitHub인증까지 모두 자동 진행
- 생성된 코드를 바로 저장소로 저장하고, 코딩 에이전트를 활용하거나, Codespace를 통해 GitHub Copilot에이전트로 개발



Copilot chat on GitHub.com

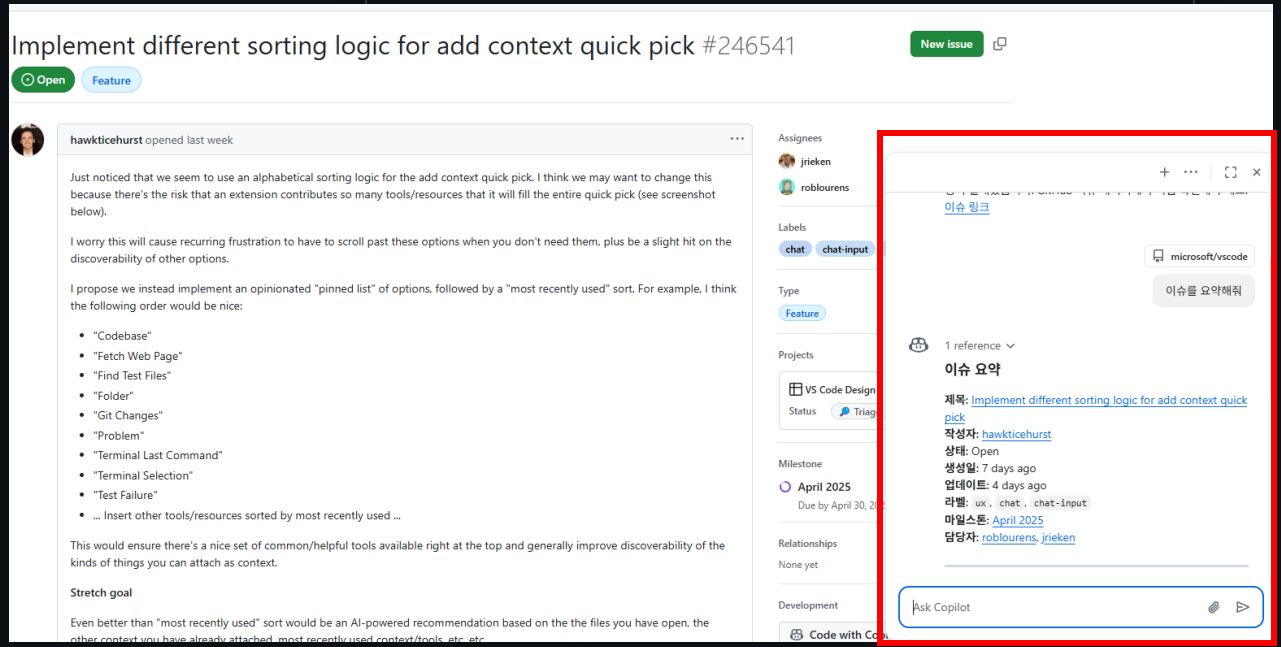
GitHub Enterprise, GitHub Mobile app에서
Copilot Chat 사용

- 대화형 검색, 질의 응답
- Issue, Pull Request summary
- Customized to your source code / documentation
- Summaries and references to sources

Copilot summaries for discussions & issues

AI가 제공해 주는 요약을 통해 편리하고 신속하게 긴 문맥을 이해

주요 포인트와 문맥을 편리하게 이해하도록 하여, 개발팀간 협업을 더욱 원활하게 하며 생산성을 높여 줌



Code scanning alerts / #75

Server-side request forgery

Open in master on Oct 23, 2024

Copilot Autofix AI generated a fix 5 months ago

To fix the SSRF vulnerability, we need to validate the `host` header before using it to construct the U
to maintain a list of authorized hosts and ensure that the `host` header matches one of these autho
approach will prevent attackers from injecting arbitrary hosts into the request.

1. Create a list of authorized hosts.
2. Check if the `host` header is in the list of authorized hosts.
3. If the `host` is not authorized, handle the error appropriately (e.g., log the incident and return

...set/src/main/java/org/owasp/webgoat/password_reset/**ResetLinkAssignmentForgotPassword.java**

Error

Server-side request forgery

Potential server-side request forgery due to a user-provided value.

Directly incorporating user input into an HTTP request without validating the input can facilitate server-side requ
these attacks, the server may be tricked into making a request and interacting with an attacker-controlled server.

Show paths

... @@ -35,2 +35,3 @@

johnjohncom/testvulner

이 보안 문제를 수정하려면 어떻게 하지?

보안 문제: Server-Side Request Forgery (SSRF)

문제 설명:
사용자 입력을 검증 없이 HTTP 요청에 직접 포함시키는 경우, 서버가 공격자가 제어하는 서버와 통신하도록 조작될 수 있습니다. 이는 Server-Side Request Forgery(SSRF) 공격을 초래할 수 있습니다.

취약 코드 위치:

파일:
`webgoat-lessons/password-reset/src/main/java/org/owasp/webgoat/password_reset/ResetLinkAssignmentForgotPassword.java`

라인:

Ask Copilot

Code | Issues 1 | Pull requests 3 | Actions | Projects | Wiki

Add /user endpoint #11

Open

son7211 opened on Aug 30, 2024

- add /user endpoint for user to show each user's profile
- /user/order endpoint for user's order information (product and the number of product

Create sub-issue

son7211 self-assigned this on Aug 30, 2024

son7211 added a commit that references this issue

Add /user endpoint ...

son7211 mentioned this on Sep 12, 2024

Steps to Implement:

1. **Define the Routes:**
   - Use your server framework (e.g., Express.js for Node.js) to define new routes for `/user` and `/user/order`.
2. **Database Integration:**
   - Ensure your database has the necessary tables and fields for user profiles and order information.
   - For `/user`, retrieve user profile data such as name, email, etc.
   - For `/user/order`, fetch the order details linked to the user, including product names and quantities.
3. **Controller Logic:**
   - Create functions in your controller to handle the logic for fetching and formatting data for the endpoints.

johnjohncom/demo-onlinestore-s...

이 기능을 추가하려면 어떻게 해야 할지 제안해줘

해당 URL에서 제공된 이슈 (https://github.com/johnjohncom/demo-onlinestore-server/issues/11)는 특정 이슈에 대한 내용으로 보입니다. 이를 기반으로 추가 개발 또는 기능 구현 방법을 제안하려면, 먼저 이슈의 세부 내용을 확인해야 합니다. 지금 바로 이슈의 내용을 확인하겠습니다.Issue #11 in this repository suggests adding the following endpoints:

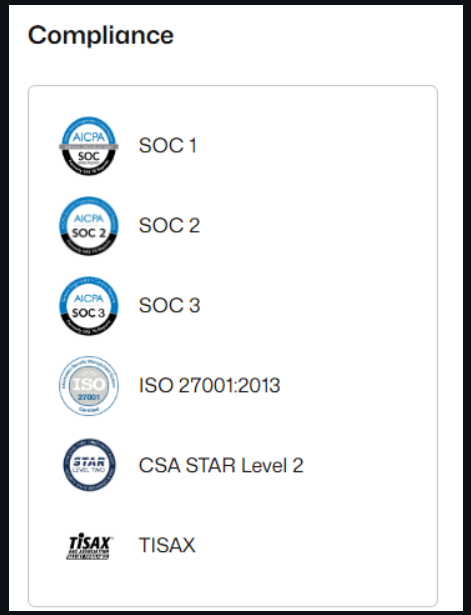
1. `/user` **endpoint**: To show each user's profile.
2. `/user/order` **endpoint**: To display a user's order information, including the product and the number of products.

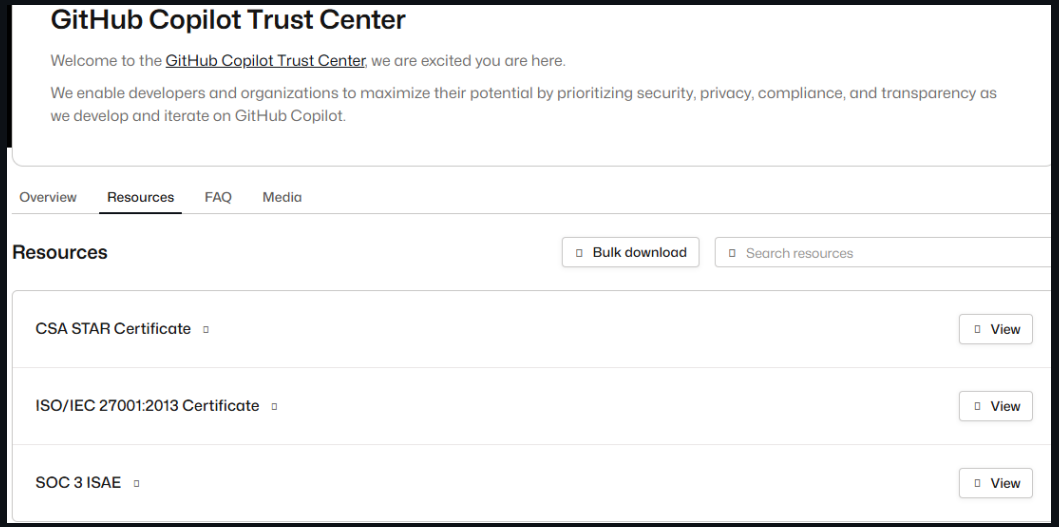
Steps to Implement:

1. **Define the Routes:**

Copilot is responding...

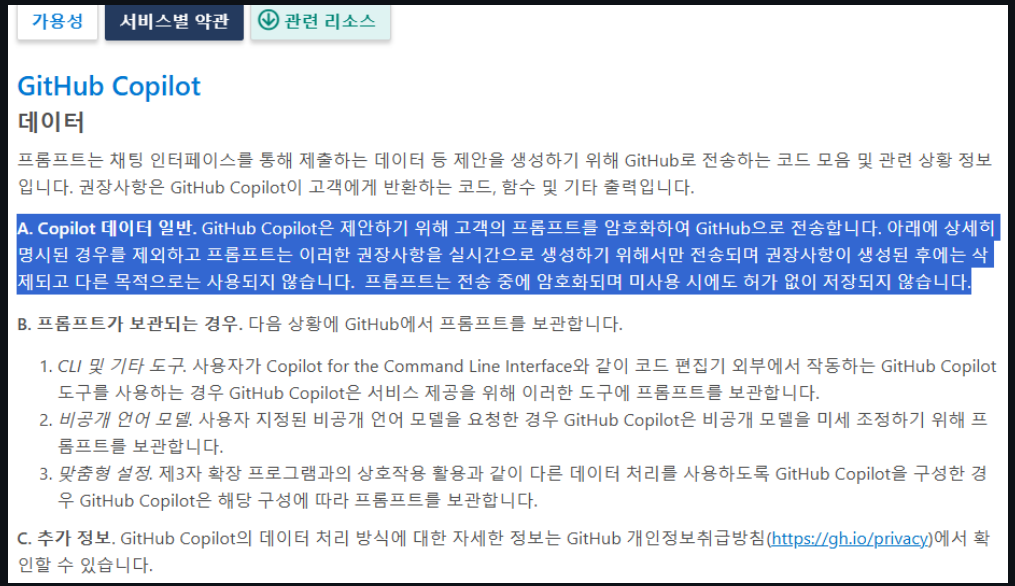
국제 보안 규격 준수





https://copilot.github.trust.page/

GitHub Copilot Trust



가용성 | 서비스별 약관 | 관련 리소스

GitHub Copilot

데이터

프롬프트는 채팅 인터페이스를 통해 제출하는 데이터 등 제안을 생성하기 위해 GitHub로 전송하는 코드 모음 및 관련 상황 정보입니다. 권장사항은 GitHub Copilot이 고객에게 반환하는 코드, 함수 및 기타 출력입니다.

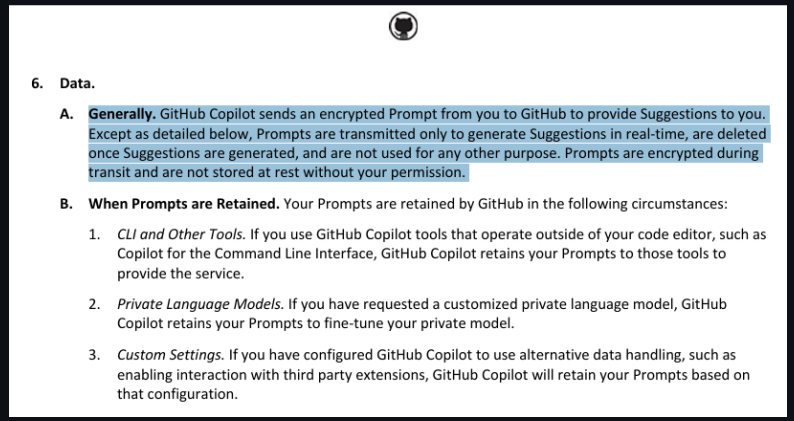
A. Copilot 데이터 일반. GitHub Copilot은 제안하기 위해 고객의 프롬프트를 암호화하여 GitHub으로 전송합니다. 아래에 상세히 명시된 경우를 제외하고 프롬프트는 이러한 권장사항을 실시간으로 생성하기 위해서만 전송되며 권장사항이 생성된 후에는 삭제되고 다른 목적으로는 사용되지 않습니다. 프롬프트는 전송 중에 암호화되며 미사용 시에도 허가 없이 저장되지 않습니다.

B. 프롬프트가 보관되는 경우. 다음 상황에 GitHub에서 프롬프트를 보관합니다.

1. *CLI 및 기타 도구.* 사용자가 Copilot for the Command Line Interface와 같이 코드 편집기 외부에서 작동하는 GitHub Copilot 도구를 사용하는 경우 GitHub Copilot은 서비스 제공을 위해 이러한 도구에 프롬프트를 보관합니다.
2. *비공개 언어 모델.* 사용자 지정된 비공개 언어 모델을 요청한 경우 GitHub Copilot은 비공개 모델을 미세 조정하기 위해 프롬프트를 보관합니다.
3. *맞춤형 설정.* 제3자 확장 프로그램과의 상호작용 활용과 같이 다른 데이터 처리를 사용하도록 GitHub Copilot을 구성한 경우 GitHub Copilot은 해당 구성에 따라 프롬프트를 보관합니다.

C. 추가 정보. GitHub Copilot의 데이터 처리 방식에 대한 자세한 정보는 GitHub 개인정보취급방침(https://gh.io/privacy)에서 확인할 수 있습니다.

Microsoft Product Terms



6. Data.

A. Generally. GitHub Copilot sends an encrypted Prompt from you to GitHub to provide Suggestions to you. Except as detailed below, Prompts are transmitted only to generate Suggestions in real-time, are deleted once Suggestions are generated, and are not used for any other purpose. Prompts are encrypted during transit and are not stored at rest without your permission.

B. When Prompts are Retained. Your Prompts are retained by GitHub in the following circumstances:

1. *CLI and Other Tools.* If you use GitHub Copilot tools that operate outside of your code editor, such as Copilot for the Command Line Interface, GitHub Copilot retains your Prompts to those tools to provide the service.
2. *Private Language Models.* If you have requested a customized private language model, GitHub Copilot retains your Prompts to fine-tune your private model.
3. *Custom Settings.* If you have configured GitHub Copilot to use alternative data handling, such as enabling interaction with third party extensions, GitHub Copilot will retain your Prompts based on that configuration.

GitHub Customer agreements

Q&A

설문조사에 참여 부탁드립니다🙏

저희 GitHubKR 계정을 팔로우하시면,
개발자분들을 위한 새로운 소식과 정보를 받아보실 수 있습니다! ✨


Thank you