

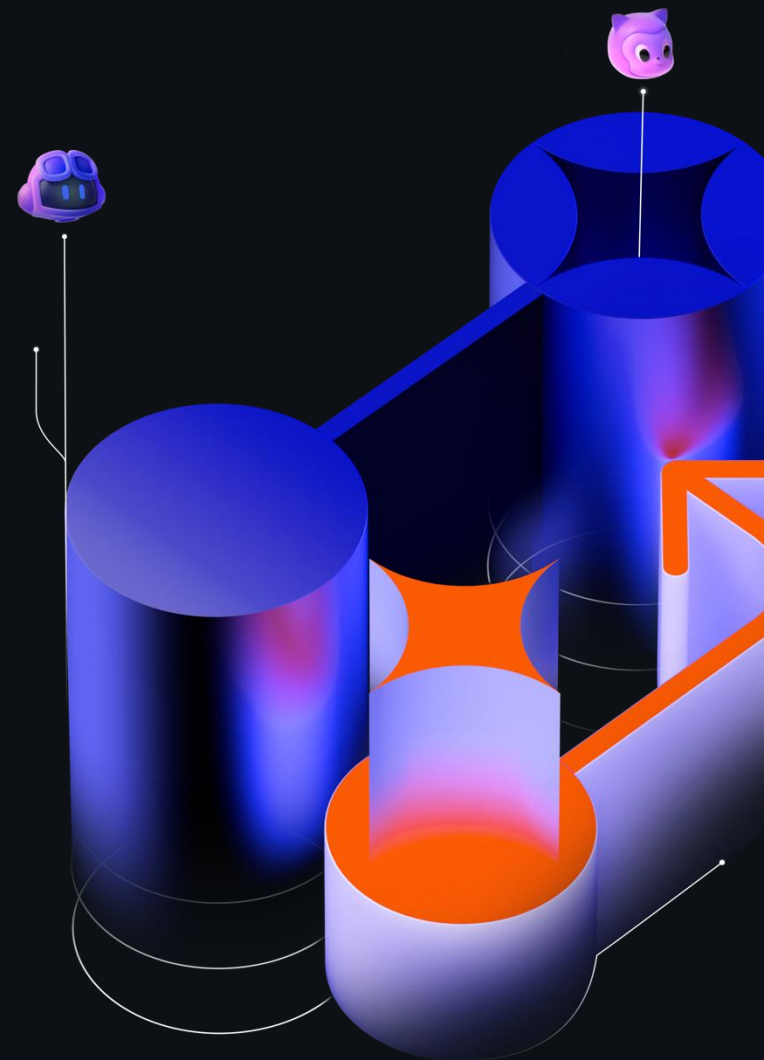
# GitHub アップデート 50本ノック！ 管理/支払/開発の 最新を抑える 🙌



ダニエル・チョ

シニアソリューションズエンジニア

GitHub Japan

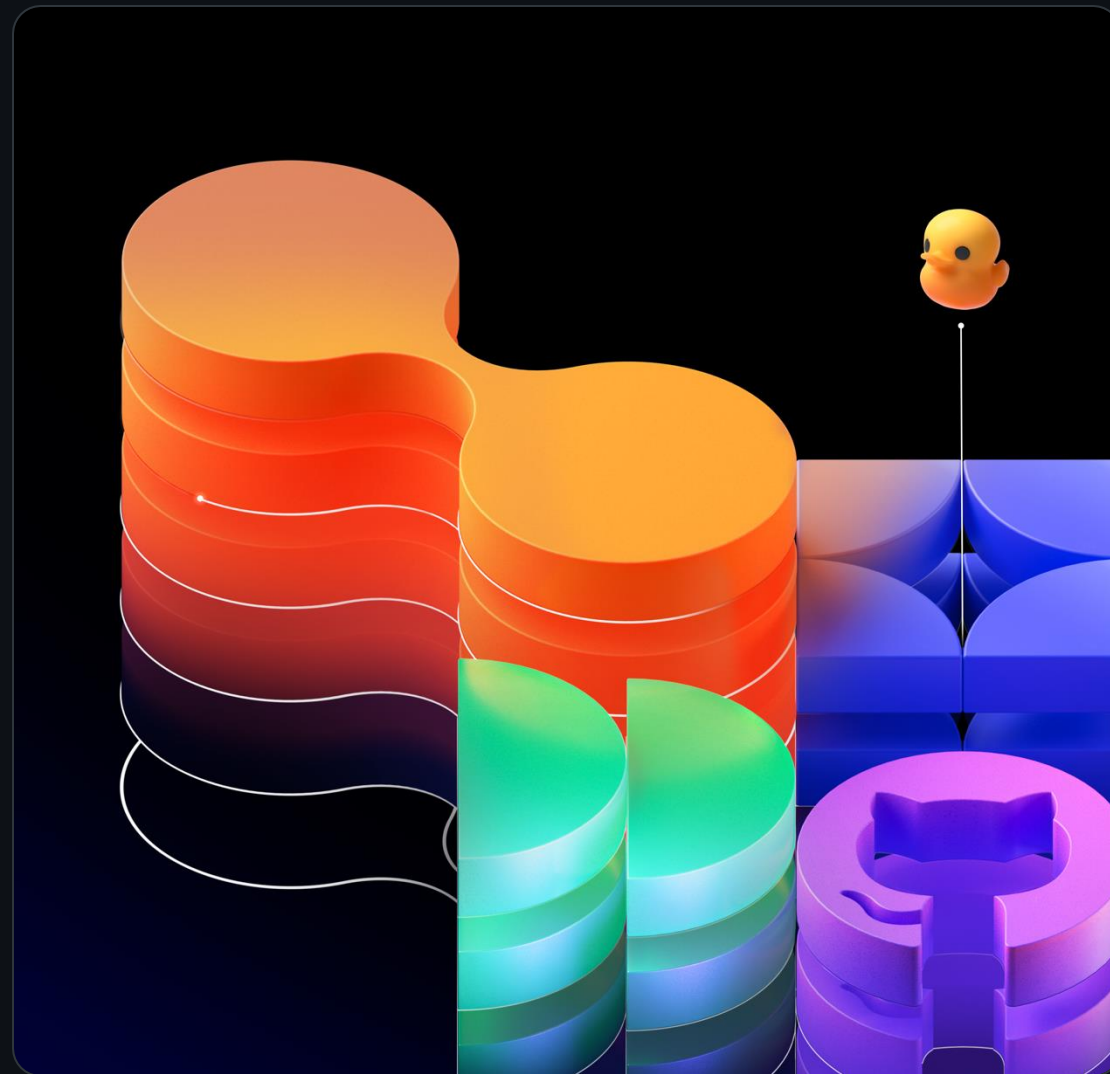


# GitHub Universe の発表 内容

基盤関連の発表

# GitHub の基盤: 制御、 セキュリティ、 効率性を向上

企業利用のお客様向けにより多くの制御方法、高度なセキュリティ担保、より優れた効率性、そしてより詳しい分析情報を提供。

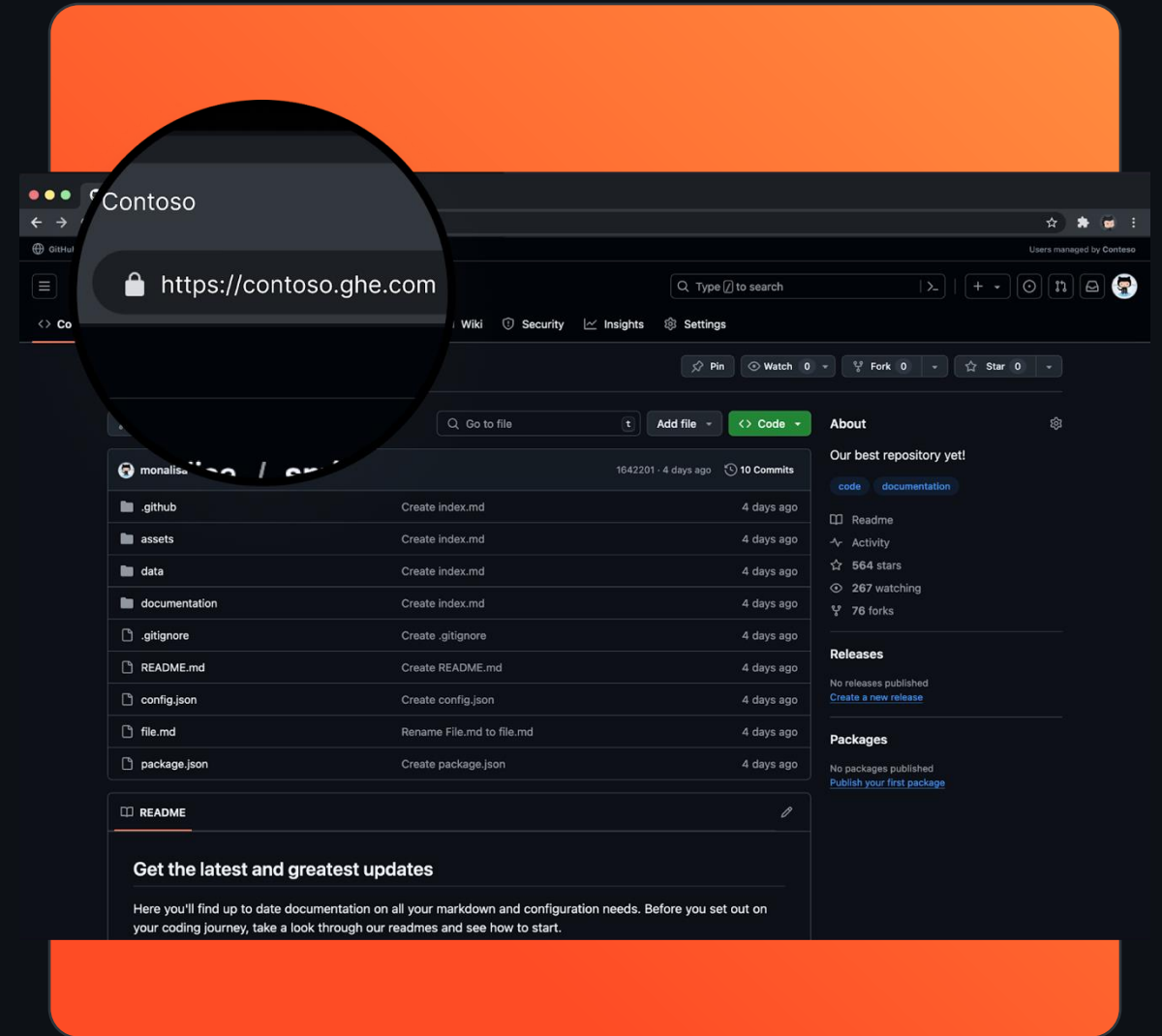


General Availability

# GitHub Enterprise Cloud

GitHub Enterprise Cloud with EU Data Residency (EU 圏内にデータが保存される GitHub Enterprise) が正式リリース

自社データの制御やコンプライアンス要件を満たし、安全かつスケール可能な SaaS ソリューションで開発を加速化。

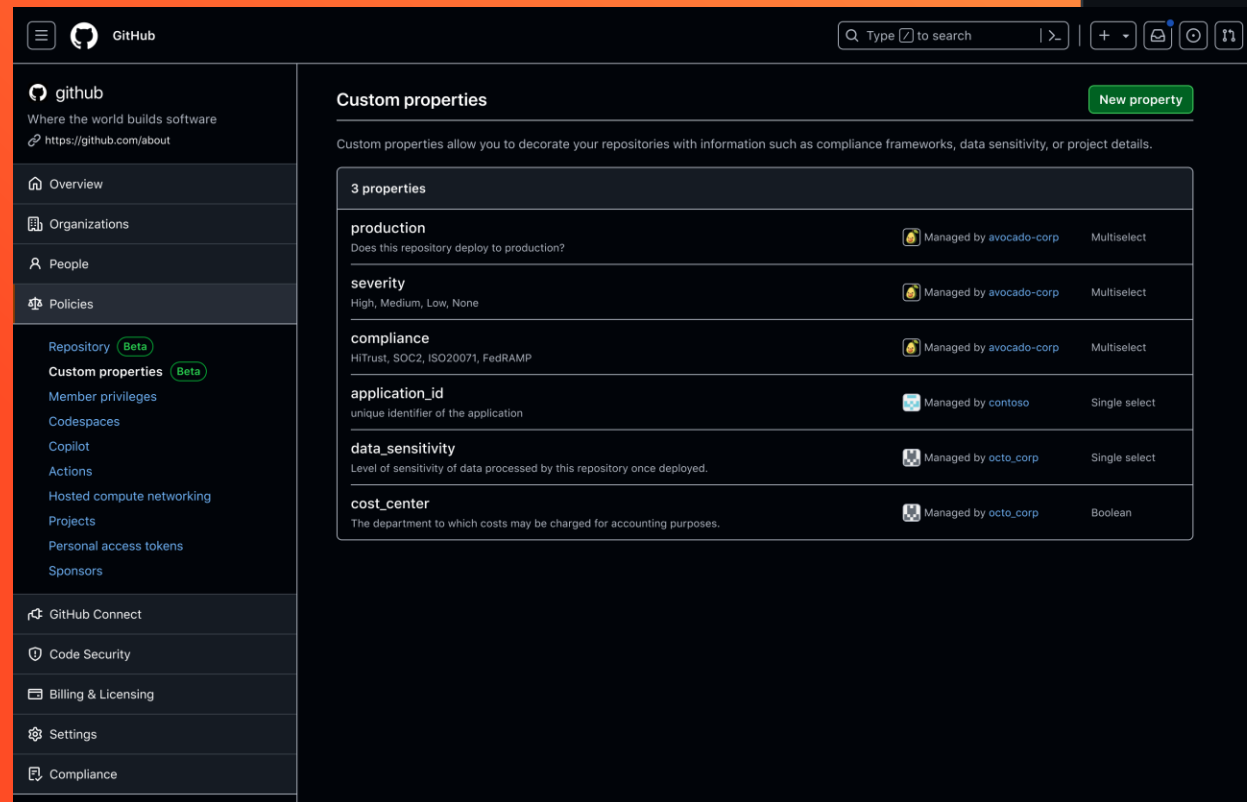


複数の段階

# Enterprise の ガバナンス及び 管理性強化

Enterprise 及び Organization 管理者向けの  
新たなガバナンス、制御、管理機能を紹  
介。

GitHub Enterprise 環境の管理業務を統一し  
、関連ポリシーを適用し、管理業務をス  
ケールさせる。

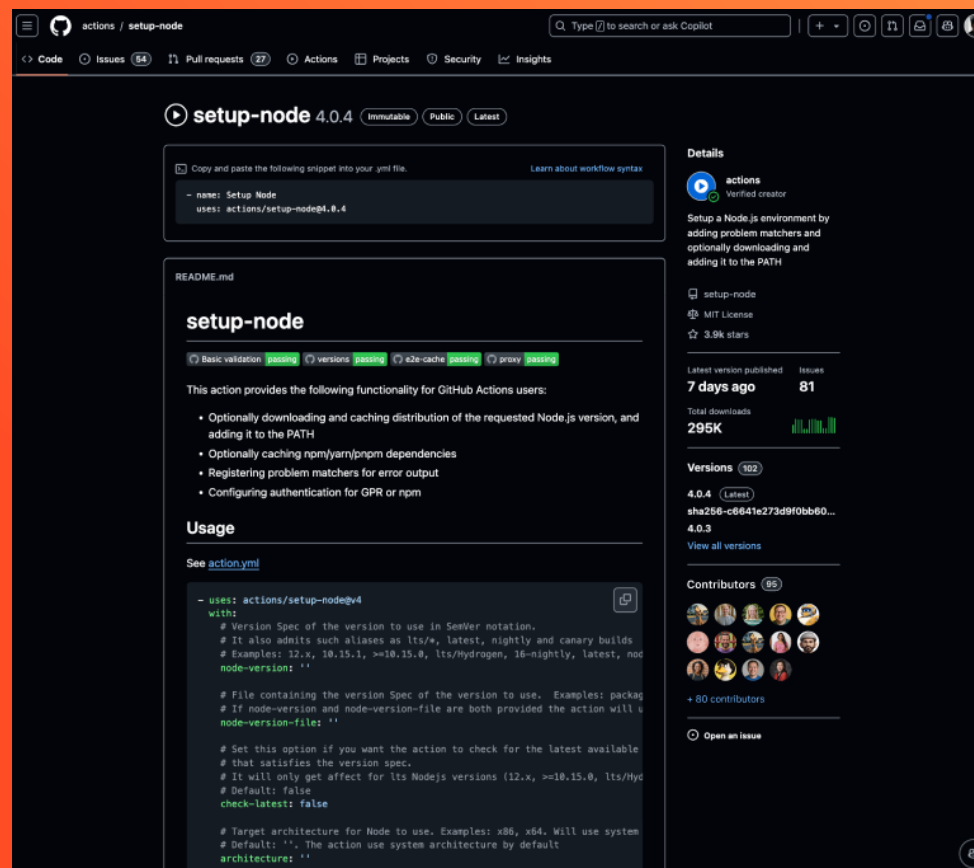


パートナー向け公開済み

# 不変かつ バージョンされた パッケージで Actionsの 安全性を強化

GitHub Actions を不変かつバージョンされたパッケージとして公開・利用

CI/CD のサプライチェーンセキュリティを  
変更不可能かつ予測可能なワークフロー  
で強化し、より信頼できるプロセスを確保

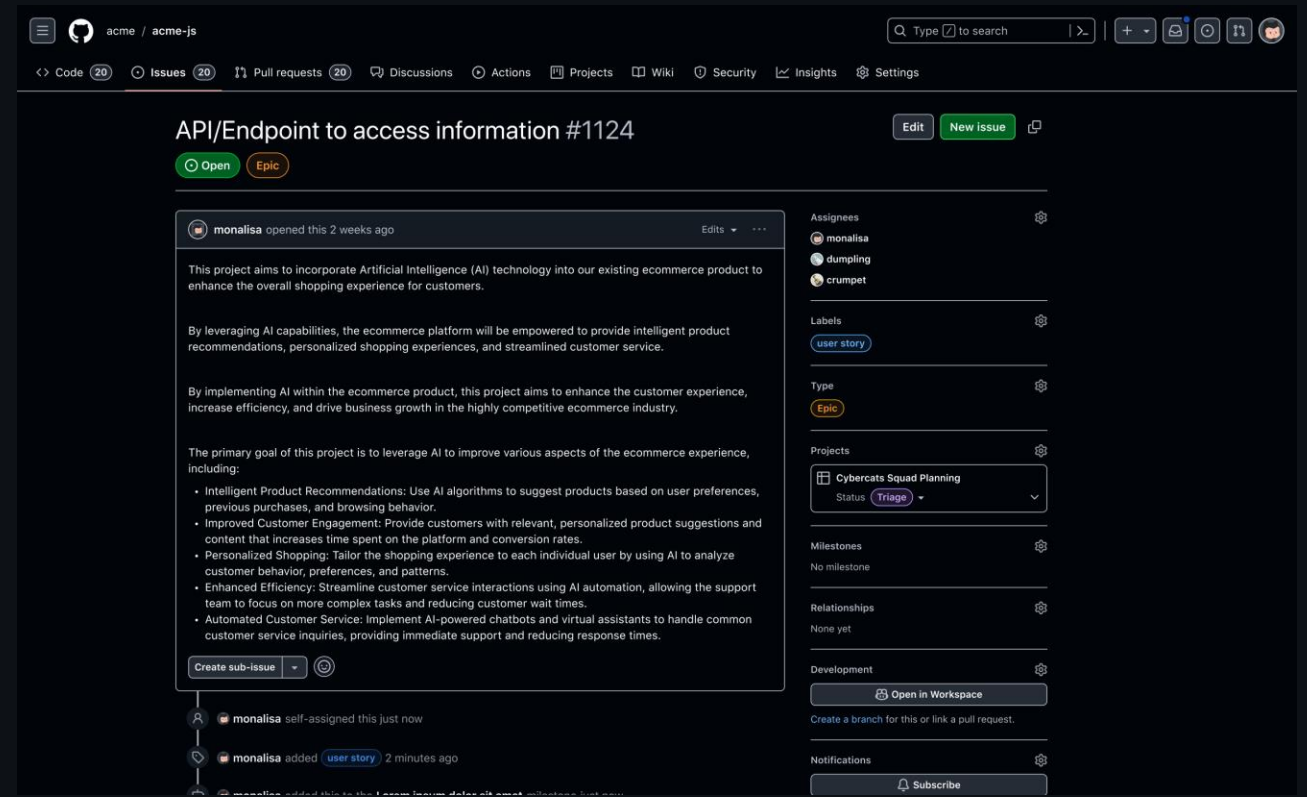


Public Beta

# GitHub Issues 2.0

GitHub issuesで新たな整理方法やコラボレーション促進、効率性の向上を実現する高度な機能を実装

サブIssue、Issue種別、高度な検索機能、そしてProjectsの上限引き上げでよりスムーズにプロジェクト管理を実現





Public Beta

# OSS 向け GitHub 提供の Arm64 ランナー

Linux 及び Windows の Arm64 GitHub ランナーで CI/CD ワークフローをより速く、持続可能な形で実行

OSS 及び個人のエンジニアはご自身の無償枠で Arm ランナーを利用することが可能に





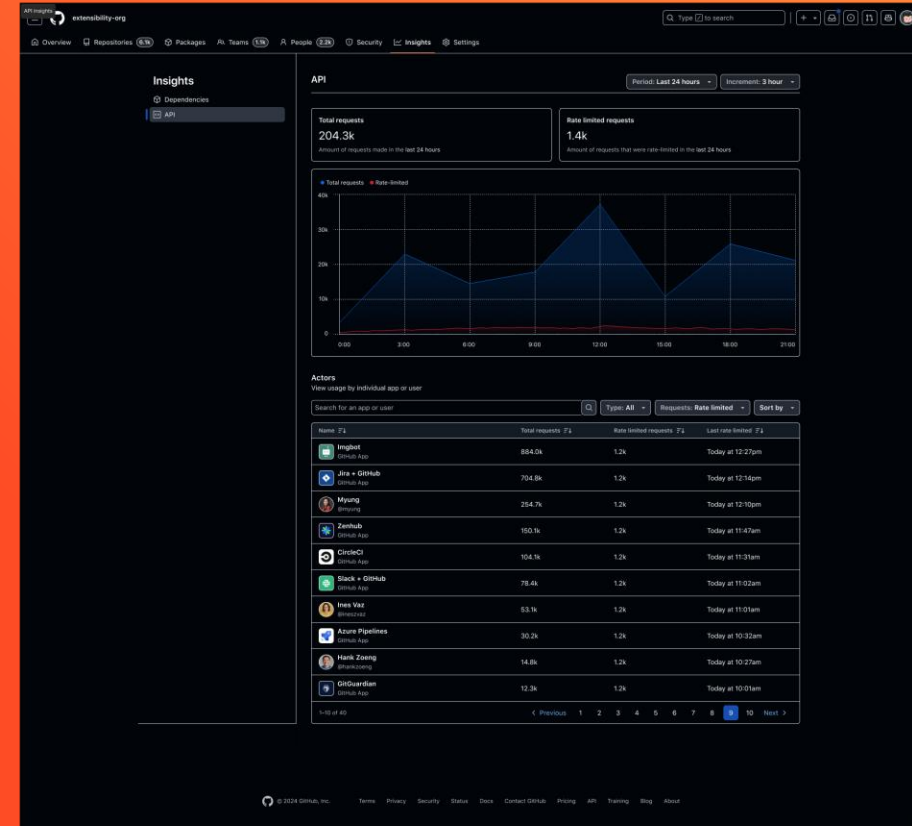
Public Beta

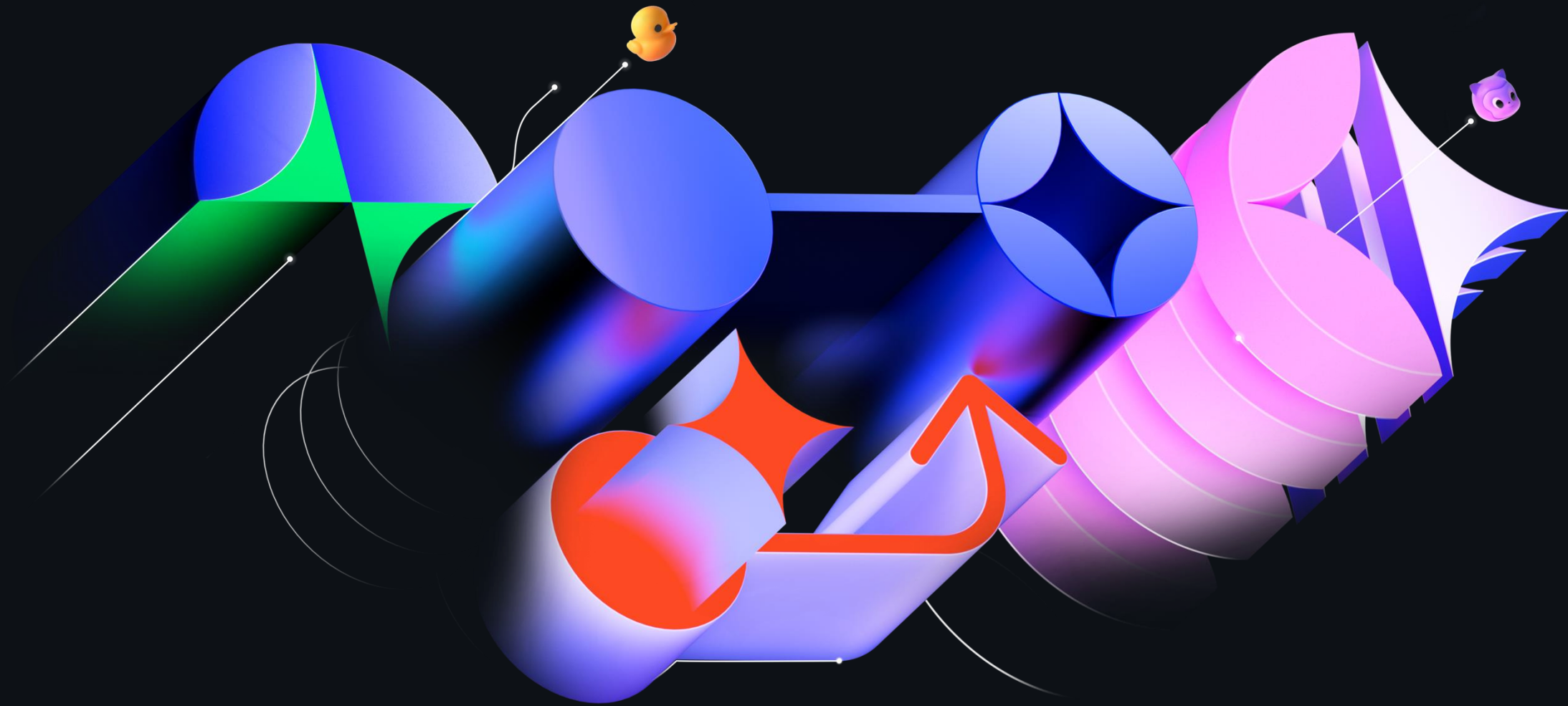
# API 分析 & Actions

## パフォーマンス メトリック

API 利用や CI/CD ワークフローに関する新たな分析情報で運用の透明性やパフォーマンス向上が可能に

API のレート上限問題を回避するために活動状況を監視し、プロセスの最適化やソフトウェア展開を加速するためにワークフローのパフォーマンスを分析

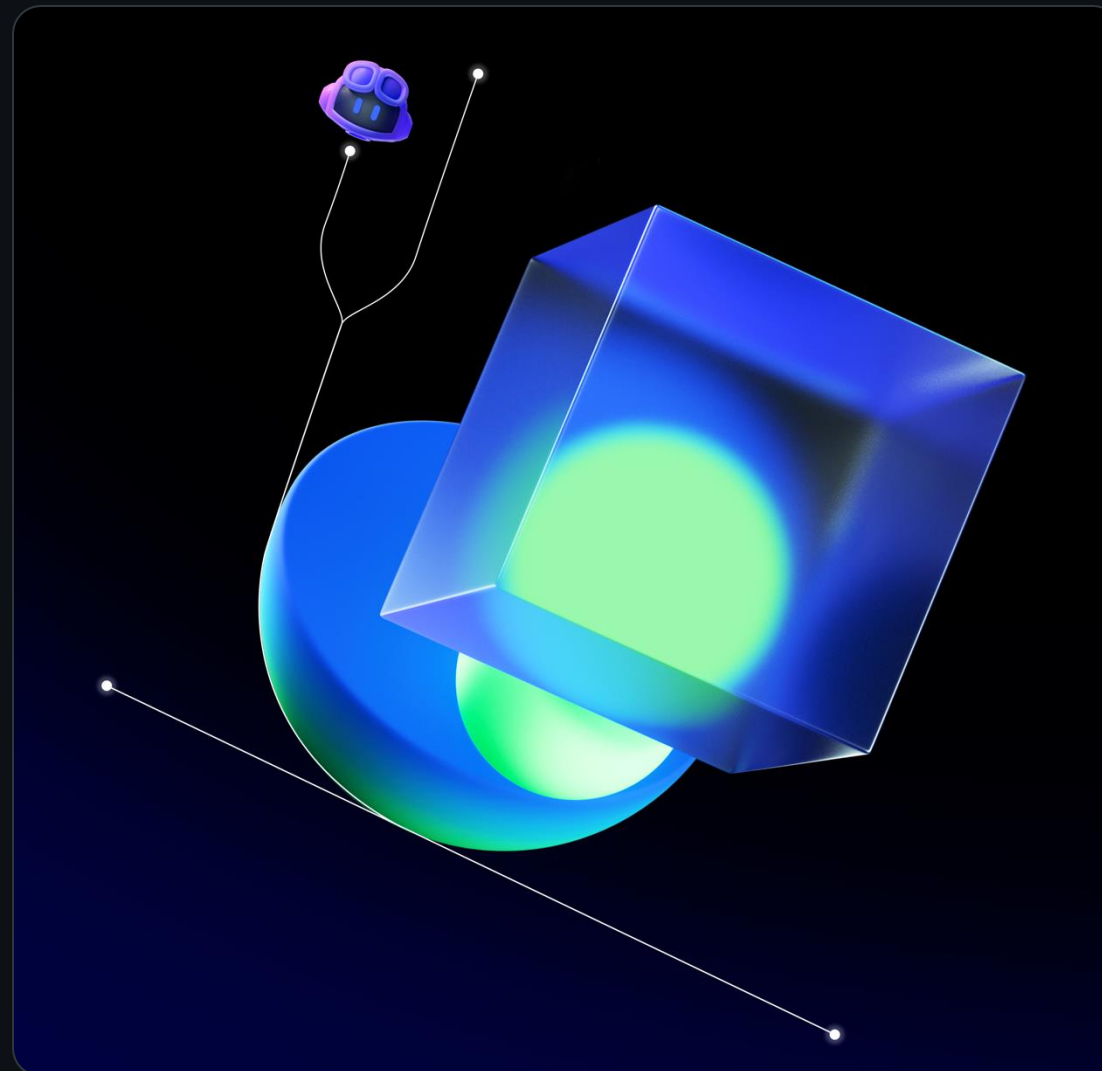




セキュリティに関する発表内容

# GitHub Advanced Security: 前段階でセキュリティを

今回の Universe でリリースを発表した機能はAIによる修正性能を強化し、セキュリティと開発チーム間のコラボレーションを向上させ、セキュリティ管理や監視機能を改善する目的になります。



Public Beta

# Security Campaigns

Copilot Autofix の力をスケールさせ、セキュリティ負債の低減を助ける

脆弱性の対応を加速させ、同時にセキュリティ管理者や開発エンジニアをコラボレーション可能なワークフローに参加させてセキュリティ姿勢を強化

The screenshot displays a dashboard for an "SQL injection (CWE-89) campaign". The main heading is "SQL injection (CWE-89) campaign" with a subtitle: "Remediating Cross-Site Scripting (XSS) vulnerabilities prevents data theft, session hijacking, and unauthorized actions, ensuring regulatory compliance, maintaining application integrity, and enhancing user trust."

**Campaign progress:** 97% (701 alerts), 23 alerts left. 701 closed, 13 in progress. Campaign started 20 days ago.

**Status:** 7 days left. Due date is November 15, 2024.

**Copilot Autofix:** 670 supported. Copilot Autofix will try to support alerts. Read more.

Filter | Search or filter

Open 23 | Closed 701

- Potentially unsafe quoting (Critical) Autofix  
Opened last month • Detected by CodeQL in pkg/.../accesscontrol/rules.go :234
- Potentially unsafe quoting (Critical) Autofix  
Opened last month • Detected by CodeQL in pkg/.../accesscontrol/rules.go :226
- Database query built from user-controlled sources (Critical) Autofix  
Opened 3 years ago • Detected by MyTool in controllers/people.js :6
- Query built from user-controlled sources (Critical) Autofix  
#41 • detected by CodeQL 2 days ago in src/.../introduction/SqlInjectionLesson4.java:62
- This query depends on a user-provided value (Critical) Autofix  
Opened 2 years ago • Detected by MyTool in controllers/people.js :6
- Database query built from user-controlled sources (High) Autofix  
Opened 10 months ago • Detected by TestTool in controllers/artists.js :126
- Query built from user-controlled sources (High) Autofix  
Opened last week • Detected by CodeQL in src/.../mitigation/Servers.java :73
- Database query built from user-controlled sources with additional heuristic sources (High) Autofix  
Opened last month • Detected by CodeQL in service/search.js :6

< Previous | Next >

Public Beta

# Dependabot 向け Copilot Autofix

AIによる修正案で依存関係関連の大きな変更に対応

JavaScriptのプロジェクトにおいて脆弱性の対応を加速させ、依存関係の更新作業を簡略化することでセキュリティリスクを低減

github-advanced-security (bot) found potential problems now View reviewed changes

```
package.json
12 12     "typescript": "^5.6.2"
13 13     },
14 14     "dependencies": {
15 -     "lodash": "3.10.1"
15 +     "lodash": "4.17.21"
```

✗ Check failure

🔄 Code scanning / Dependabot

**Breaking change detected when updating lodash from version 3.10.1 to 4.17.2** Error

Property 'contains' does not exist on type 'LoDashStatic'.

[Show more details](#)

Show paths Dismiss alert

🧠 Copilot Autofix (AI) less than a minute ago

To fix the problem, we need to replace the usage of the `_.contains` method with an equivalent method available in Lodash v4. The `_.includes` method serves the same purpose and is available in Lodash v4. This change will ensure that the code remains functional after updating Lodash.

- Replace `_.contains` with `_.includes` in the `index.ts` file.
- Ensure that the logic and functionality remain the same.

```
index.ts
... .. @@ -2,2 +2,2 @@
2 2
3 - console.log(_.contains([1,2,3], 1))
3 + console.log(_.includes([1,2,3], 1))
```

Copilot is powered by AI and may make mistakes. Always verify output. Dismiss Edit Commit fix

Public Beta

# パートナーコード スキャンツール向け Copilot Autofix

一部の第三者提供の静的解析ツールが報告した脆弱性に対し自動的に修正案を生成。

複数のツールに及んで脆弱性の対応を加速させ、セキュリティリスクの低減とコード品質の向上を同時に実現



github-advanced-security (bot) found a potential problem 10 minutes ago

[View reviewed changes](#)

lib/application.js:32

```
... .. @@ -30,2 +30,2 @@  
30 + var resolve = require('path').resolve;  
31 + var setPrototypeOf = require('setprototypeof');  
32 + var fetch;
```



Error

[Code scanning / ESLint](#)

### Disallow unused variables

'fetch' is defined but never used.

Variables that are declared and not used anywhere in the code are most likely an error due to incomplete refactoring. Such variables take up space in the code and can lead to confusion by readers.

[View details](#)

[View data flow](#)

[Dismiss alert](#)



Copilot Autofix (AI) 10 minutes ago

[Beta](#)

Copilot Autofix for ESLint

To fix the ESLint error regarding the unused variable `fetch` we need to remove the declaration of the variable since it is not used anywhere in the code. This will resolve the ESLint error without affecting the existing functionality of the code.

Suggested change

```
... .. @@ -32,2 +32,2 @@  
32 - var fetch;  
32 +
```

Copilot is powered by AI and may make mistakes. Always verify output.

[Edit](#)

[Apply suggestion](#)



Reply...

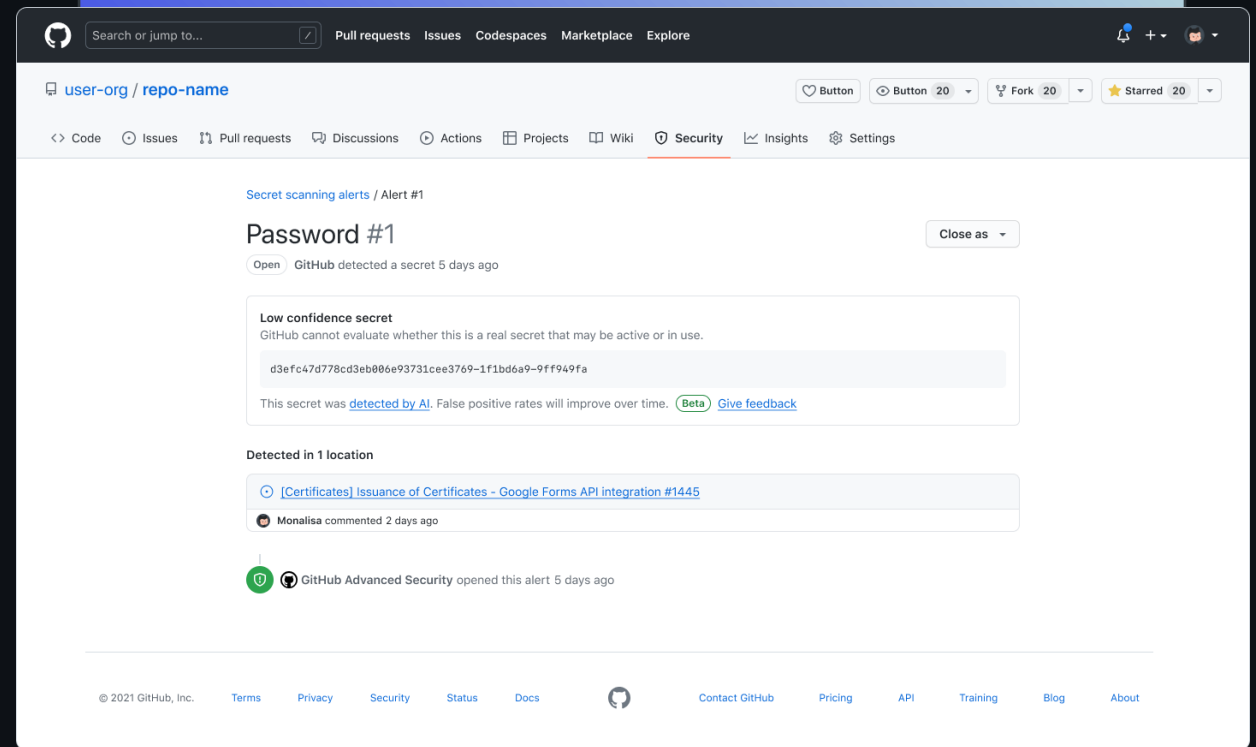
[Resolve conversation](#)

General Availability

# Copilot Secret Scanning

AI 技術を活用してパスワードや機敏な情報がコード内にあるかを確認し、混入を防ぐ

侵害が起きる前に人間が作成したシークレット情報を検知・対応することでセキュリティリスクを低減





General Availability

# OSS 向け Copilot Autofix

コードスキャンで検知された内容に対する AI による修正案は OSS 向けに無償で提供。

OSS プロジェクトで自動的にコードの問題に対し修正案を生成することで、脆弱性対応の加速及びセキュリティ性の向上が可能に。

The screenshot shows a GitHub pull request comment from the bot 'github-advanced-security'. The comment is about a code scanning issue in 'lib/application.js:32'. The code snippet is:

```
... .. @@ -30,2 +30,2 @@
30 + var resolve = require('path').resolve;
31 + var setPrototypeOf = require('setprototypeof');
32 + var fetch;
```

The error message is: "Error: Disallow unused variables. 'fetch' is defined but never used." It explains that unused variables are a common error due to incomplete refactoring and can lead to confusion. There are buttons for "View data flow" and "Dismiss alert".

Below the error, there is a "Copilot Autofix" suggestion from AI, dated 10 minutes ago. It explains that to fix the ESLint error, the declaration of the variable 'fetch' should be removed since it is not used anywhere in the code. The suggested change is:

```
... .. @@ -32,2 +32,2 @@
32 - var fetch;
32 +
```

There are buttons for "Edit" and "Apply suggestion". At the bottom, there are reaction icons (thumbs up, thumbs down, smiley, thumbs up 1, heart 7), a "Reply..." input field, and a "Resolve conversation" button.

General Availability

# セキュリティ メトリックの CSV出力

高度な分析やレポート作成のために簡単にセキュリティに関するデータを出力

出力されたデータをオフライン分析、カスタムレポート、アーカイブ等の目的で活用し、セキュリティの可視性を向上

The screenshot displays the GitHub Secret Scanning interface. On the left is a navigation sidebar with options like Overview, Organizations, People, Policies, GitHub Connect, and Code Security. The main content area is titled 'Secret scanning' and includes a filter bar, a 'Push protection' section with three summary cards (Bypassed secrets: 25/63, Bypass requests: 15 requests, Mean time to respond: 5 minutes), and three data tables: 'Most blocked secret types', 'Repositories with most pushes blocked', and 'Most bypassed secret types'. A 'Bypasses' section provides context on why secrets were bypassed, and a 'Bypassed reason distribution' table shows the breakdown of reasons like False positives, Fix later, and Used in tests.

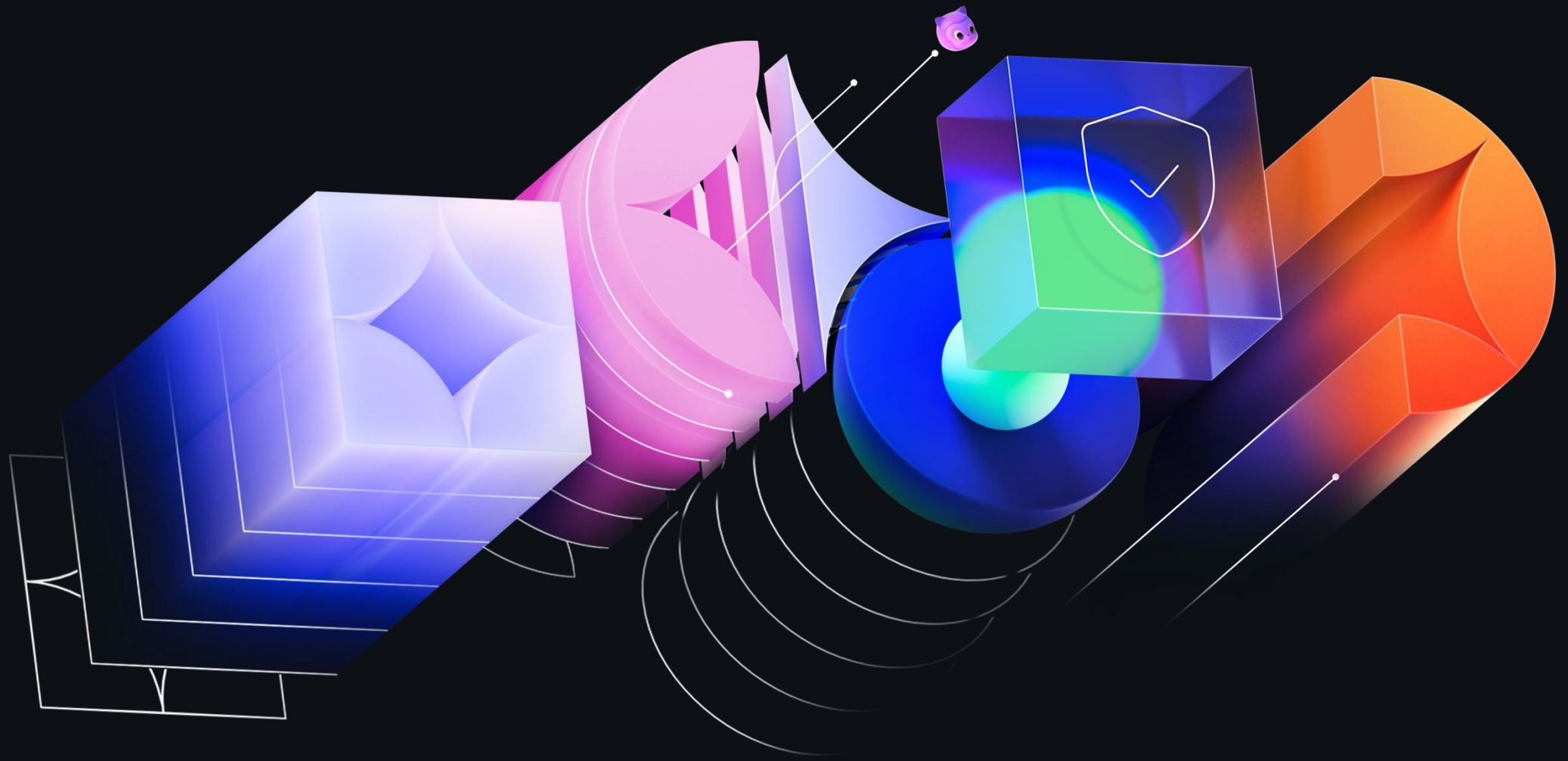
Secret Type	Count
GitHub Secret Scanning	20
GitHub Personal Access Token	20
Monas-custom-pattern (custom pattern)	20
Custom-pattern-232591	20

Repository	Count
juice-bookstore-security-v1	20
fresh-avocado	20
apples-tomatoes-bananas	20
demo-dog	20

Secret Type	Count
GitHub Short-Lived Access Token	20
GitHub Personal Access Token	20
Monas-custom-pattern	20
Custom-pattern-232591	20

Repository	Count
juice-bookstore-security-v1	20
fresh-avocado	20
apples-tomatoes-bananas	20
demo-dog	20

Reason	Count
False positives	20
Fix later	20
Used in tests	20



Copilot 関連の発表内容

# GitHub Copilot: 生産性の新たな 境地を可能に

今年のUniverseでリリースされた内容はAIによる支援をすべてのワークフローを横断して可能にし、IDE及び環境対応の拡大、そしてコラボレーションとコード品質の向上を目的としております。

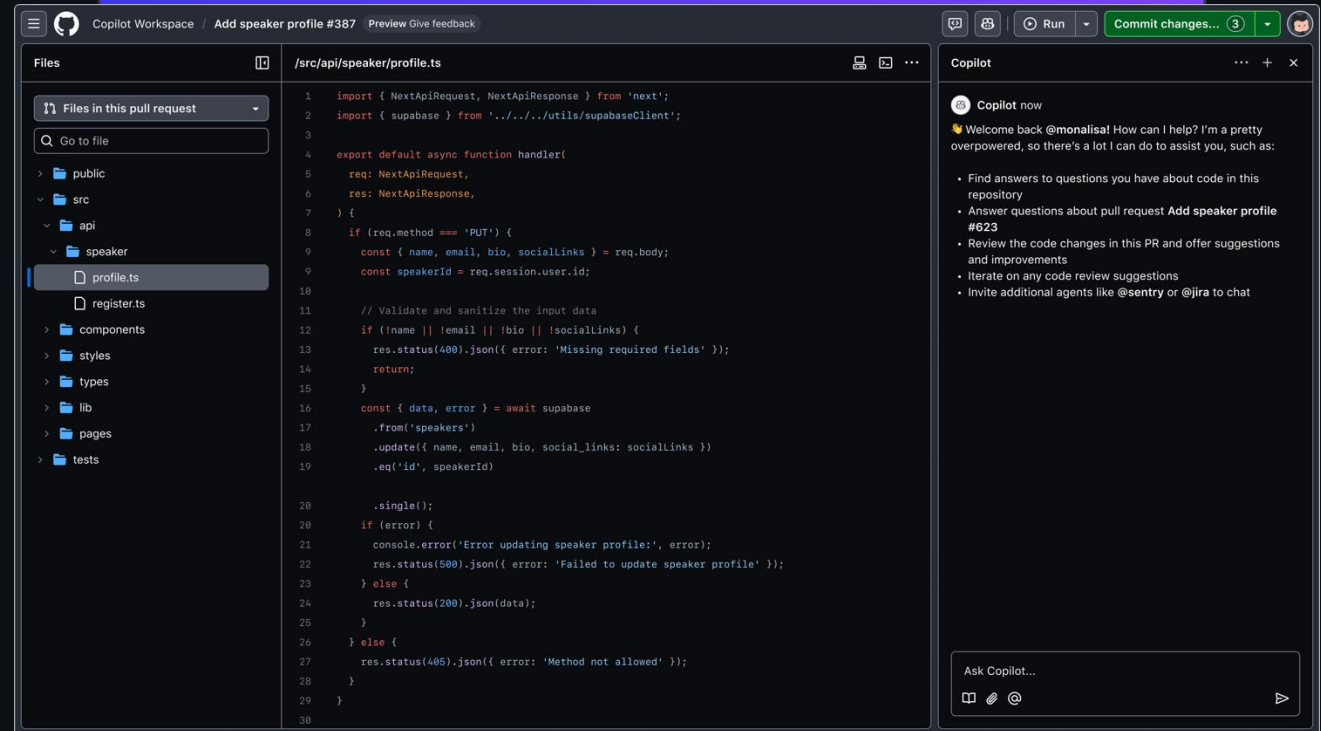


Technical Preview

# Copilot Workspace

自然言語の発想をコードに変換するあなたの AI 原動開発環境

意思から導入へ簡単に移すことで開発を加速し、コラボレーションを向上

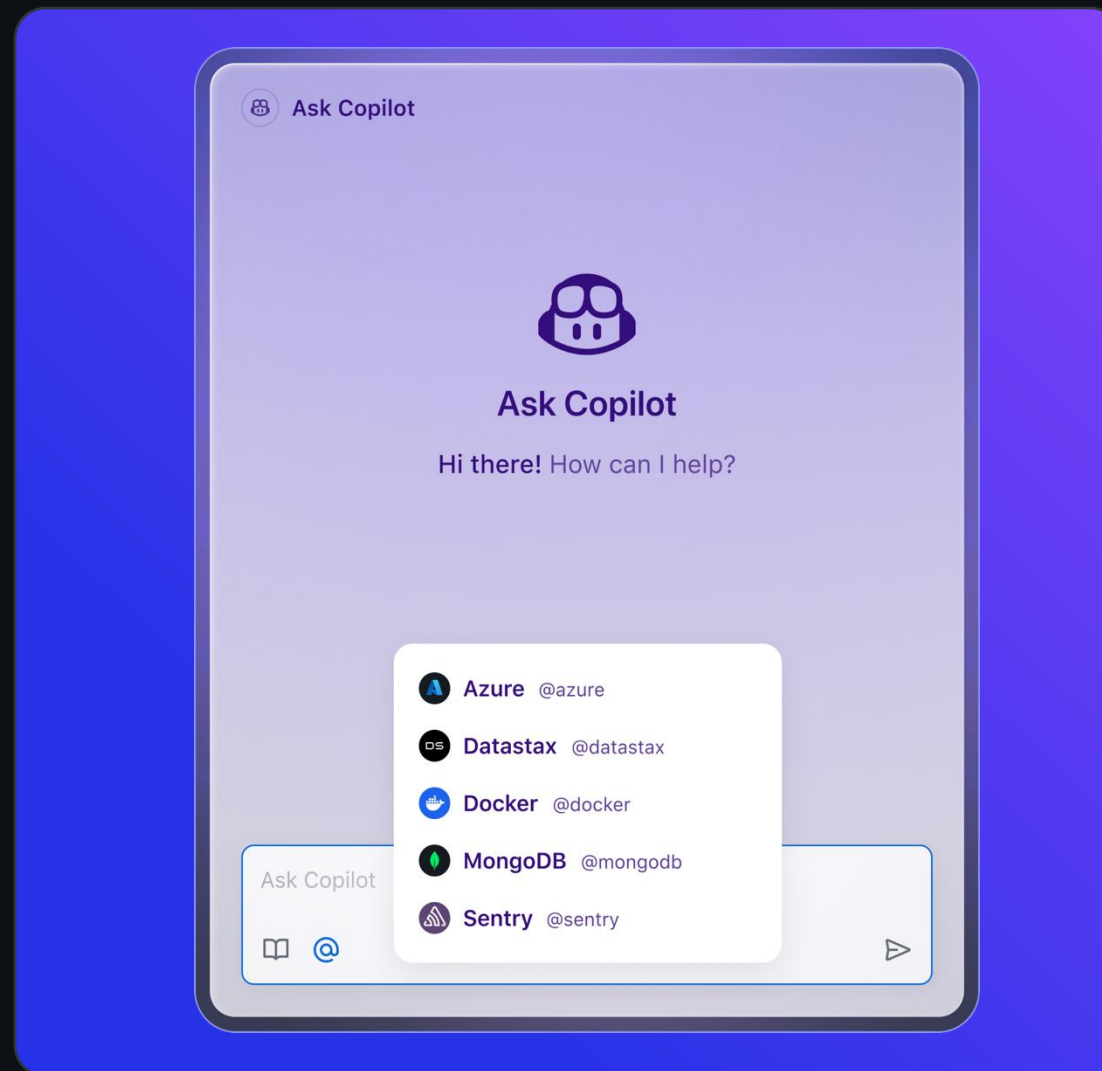


12月にGA予定

# Copilot Extensions

GitHub Copilot を好きなツールやカスタムな拡張機能を用いて拡張し、コーディング体験を向上

Copilot Chat にすべてのツールを連動させ、自分のフロー状態を維持し、生産性を向上しながら邪魔を減らす



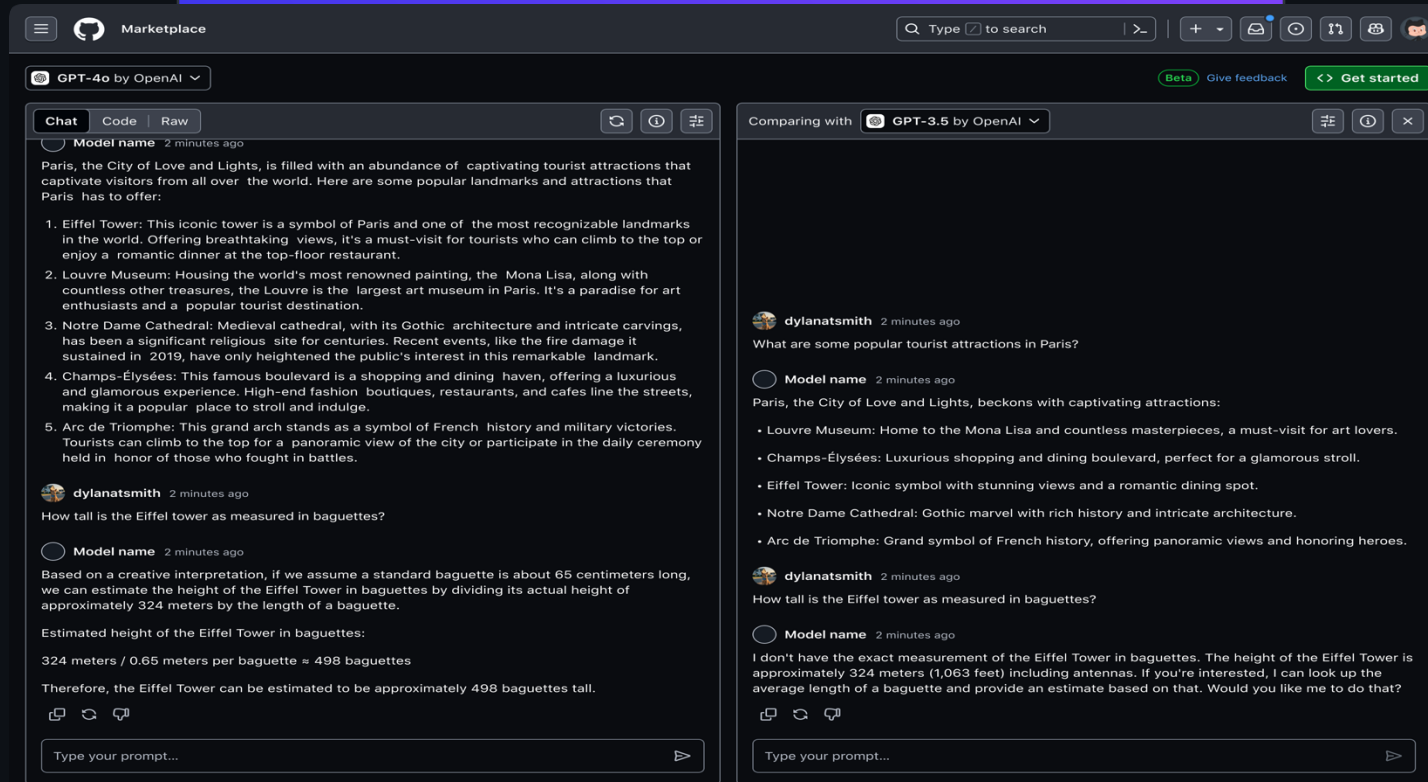


Public Beta

# GitHub Models

専用の遊び環境で複数の AI モデルを試しながら比べ、自分のプロジェクトに最適な物を見つける

GitHub 内で簡単に複数の AI モデルを触り、評価することでイノベーションと開発を加速



Azure hosted. AI powered, can make mistakes. [Share feedback](#). Subject to [Product Terms](#) and [Privacy Statement](#). Not intended for production/sensitive data.

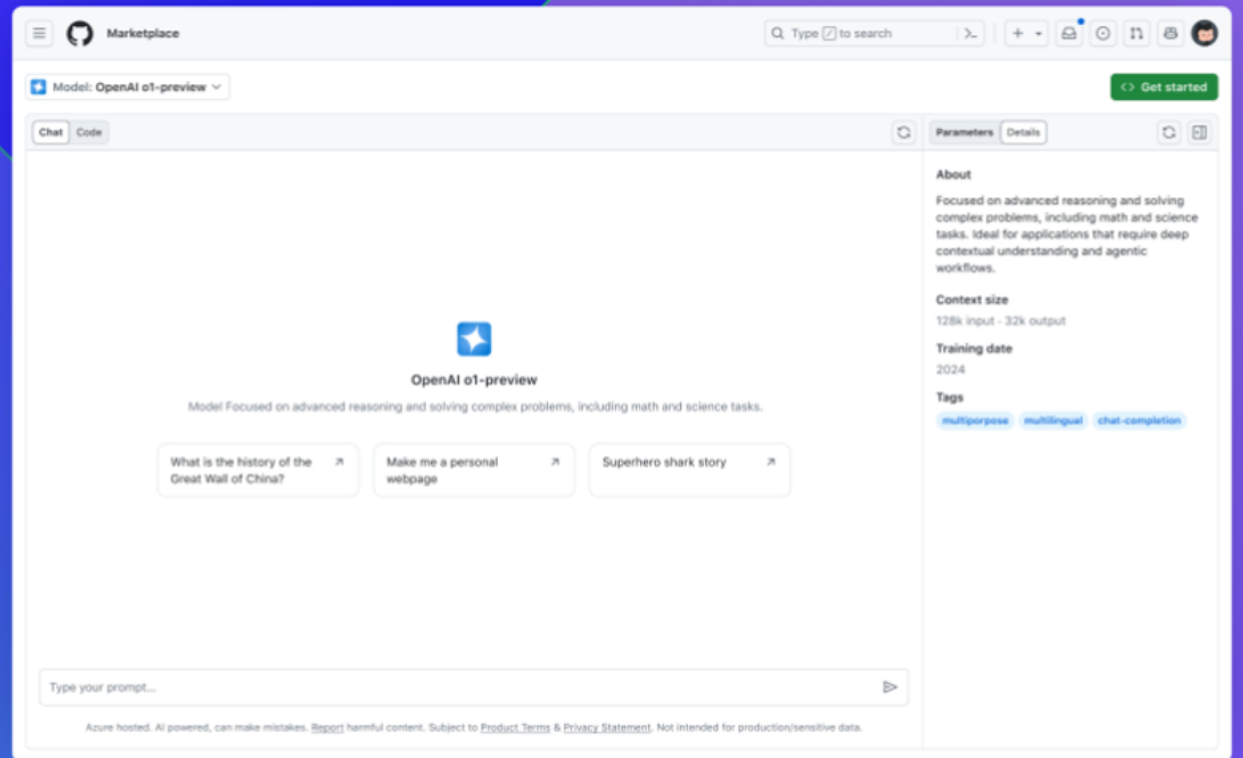


Public Beta

# OpenAI o1 モデル アクセス

新たな AI モデルを試し、開発ニーズに合わせて最適な物を選ぶ

GitHub Copilot 及び GitHub Models で最適な AI モデルを選ぶことでエンジニアの生産性をより向上

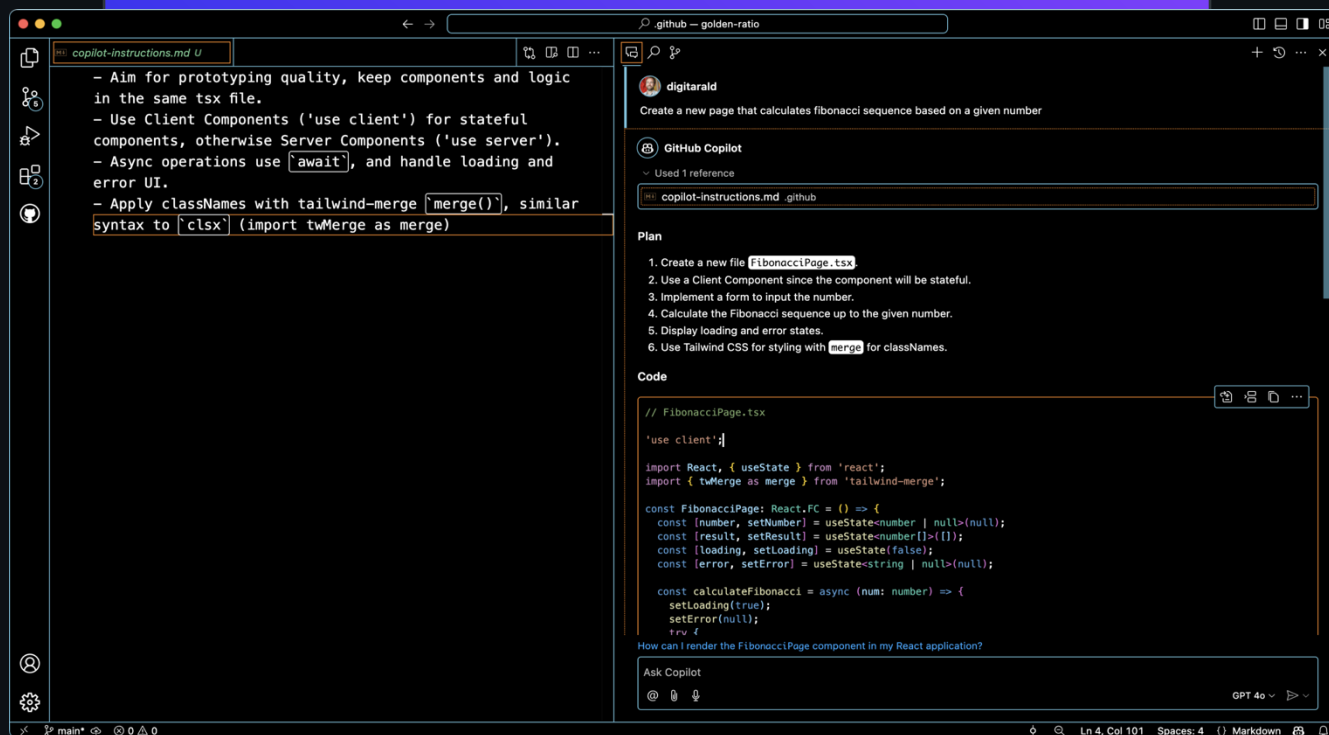


複数の段階

# IDE 体験を 次のフェーズへ

IDE での体験をカスタム指示、ビジョン対応、そして高度なデバッグ機能等の AI による新機能で次のフェーズへ昇格

IDE の中で自分に特化した AI 支援を活用してコード品質の改善及び開発の加速を図る



限定的 Public Beta

# Pull Request 向け Copilot Workspaces

GitHub の中で簡単に Pull Request のコメント  
をコードに変換

自分のワークフローから離れずコードイ  
テレーションの加速及びコード品質を向  
上

The screenshot displays a GitHub Copilot workspace interface. The main editor shows a file named `index.js` with the following code:

```
1 -console.log("testing")
1+// This is where the application starts up
2+const express = require('express');
3+const app = express();
4+
5+// This is our main page
6+app.get('/', (req, res) => res.send('Hello ${req.query.name}!'));
7+
8+// Status will tell us how it's going
9+app.get('/status', (req, res) => {
10+ res.setHeader('content-type', 'text/plain');
11+ res.send('this is fine');
12+});
13+
14+module.exports = app; // Export the app instance
2 15
```

Two suggestions from `@github-advanced-security[bot]` are visible, each with a blue border:

- Suggestion 1: A comment above line 2: `// This is where the application starts up`
- Suggestion 2: A comment above line 6: `// This is our main page`

On the right side, a notification panel shows a security warning:

- Check failure**
- Code scanning / CodeQL**
- Reflected cross-site scripting (High)**
- Cross-site scripting vulnerability due to a user-provided value.
- [Show more details](#)

Below the warning, a **Copilot Autofix (AI)** suggestion is shown, dated 1 day ago. It explains the fix:

To fix the reflected cross-site scripting vulnerability, we need to sanitize the user input before incorporating it into the HTTP response. The best way to do this is by using a well-known library like `escape-html` to escape any potentially harmful characters in the user input.

- We will import the `escape-html` library.
- We will use the `escape` function from this library to sanitize `req.query.name` before including it in the response.

The suggested changeset shows the following diff for `index.js`:

```
@@ -2,2 +2,3 @@
2 2 const express = require('express');
3 + const escape = require('escape-html');
4 4 const app = express();
@@ -5,3 +6,3 @@
5 6 // This is our main page
```

Buttons for **Dismiss** and **Apply** are located at the bottom right of the notification panel.

General Availability

# Discussions & Issues 向け Copilot 要約

AI によるインライン要約で文脈を素早く把握

Issue や Discussion 内の要点をすべてのコメントを読まずに把握し、時間短縮や生産性の向上を実現

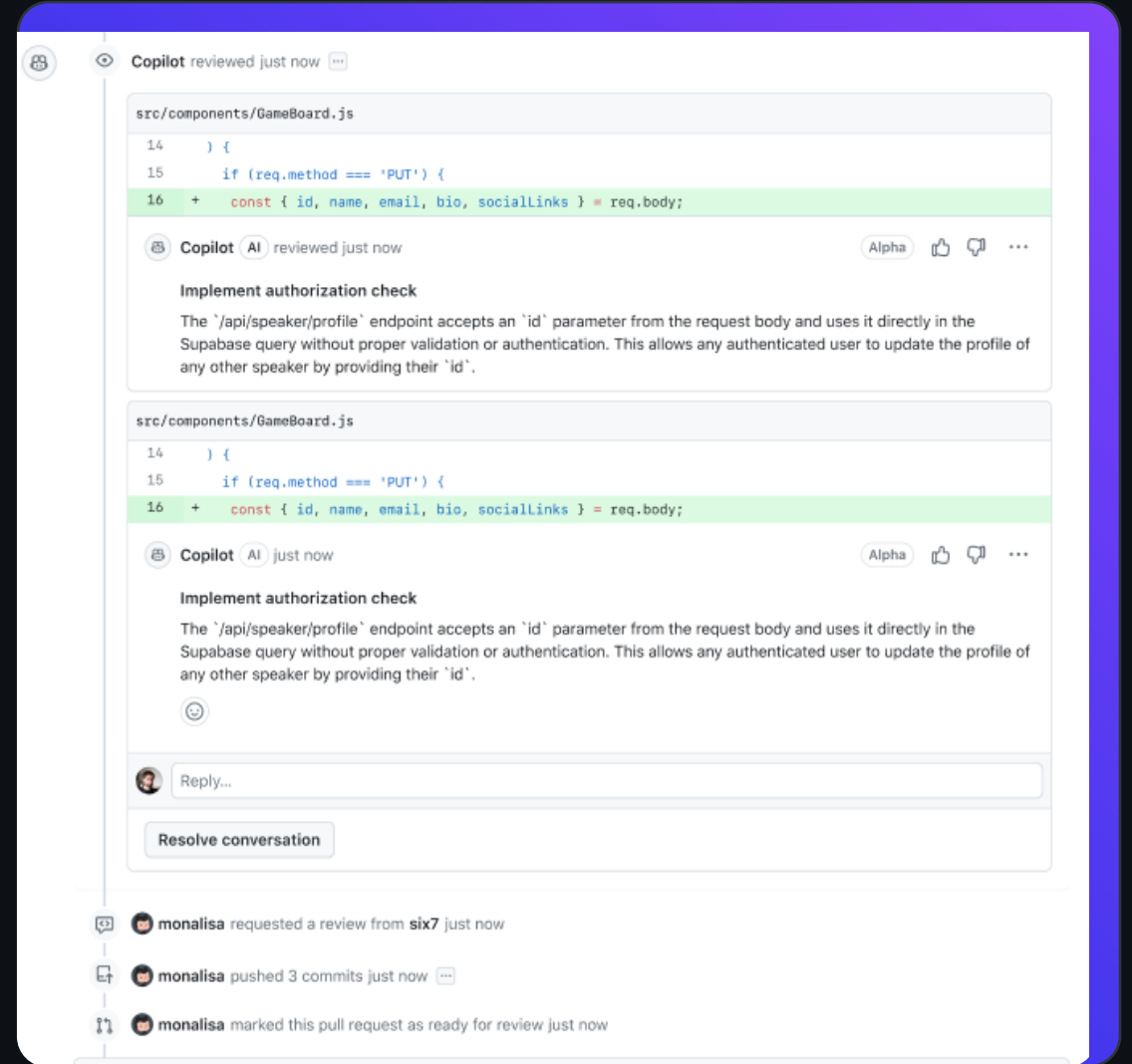
The screenshot shows a GitHub issue page for the repository 'stripe / react-stripe-js'. The issue title is 'API/Endpoint to access information from Copilot Knowledge Bases #1124'. It is categorized as 'Open' and 'Epic'. A summary from Copilot is displayed, stating that the discussion is generally positive and collaborative, focusing on improving Copilot's functionality. The issue was opened by 'rileybroughten' two weeks ago. The main content of the issue is a 'Problem Statement' and 'Additional Context'. The 'Problem Statement' describes the need for an internal developer to access Copilot Knowledge Bases (KB) to enhance context and allow LLMs to provide more contextual responses. The goal is to automate secret leak remediation end-to-end, with the biggest challenge being the management of user inputs. The 'Additional Context' provides a specific use case where the secret scanning team is experimenting with AI-based remediation and autofix behavior for alerts generated by leaked secrets. The page also shows a sidebar with assignees (monalisa, dumpling, crumpet), labels (user story, feature request), projects (Cybercats Squad Planning), and notification options.

Public Beta

# GitHub Copilot コードレビュー

Pull Request を作成すると素早く AI による  
コードフィードバックを受ける

人間によるレビューを待たずにコード品  
質を向上させ、開発サイクルを加速



複数の段階

# Copilot コード補完 及びチャットの 品質向上

.NET 向けの特定モデルを用いてコード補完及びチャット支援の体験を改善

IDE 内でより文脈に特化した提案を受け、ハルシネーション現象を低減させ、コード品質と生産性を向上

The screenshot shows a GitHub issue page for the repository 'react-stripe-js'. The issue title is 'API/Endpoint to access information from Copilot Knowledge Bases #1124'. It is categorized as an 'Epic' and is currently 'Open'. The issue was opened by 'rileybroughten' two weeks ago. The main content of the issue is a 'Problem Statement' and 'Additional Context'. The 'Problem Statement' describes the need for an internal developer to access Copilot Knowledge Bases (KB) to enhance context for LLMs. The 'Additional Context' provides a specific use case for AI-based remediation and autofix behavior for alerts generated by leaked secrets. The right sidebar shows assignees (monalisa, dumpling, crumpet), labels (user story, feature request), and a project (Cybercats Squad Planning).

react-stripe-js

Issues 20 Pull requests 20 Discussions Actions Projects Wiki Security Insights Settings

## API/Endpoint to access information from Copilot Knowledge Bases #1124

Open Epic

**Copilot** Summary up to date Visible to everyone

The sentiments in this discussion are generally positive and collaborative, with a focus on improving Copilot's functionality. There's enthusiasm for the new feature, balanced with caution about potential impacts. The team shows a user-centric approach, valuing flexibility and customization. There's also a sense of pragmatism, acknowledging the need for further clarification.

Is this helpful? 🍌 🗨

**rileybroughten** opened this 2 weeks ago Edits ...

**Problem Statement:**

As an internal developer integrating with CAPI, I should be able to access an org's Copilot Knowledge Bases (KB) to glean context that will enhance my prompt and allow LLMs to provide more contextual responses.

Our goal is automate secret leak remediation end-to-end; the biggest challenge we face with remediation workflows is org-specific context -- we're thinking Copilot knowledge bases could be a promising way to manage those user inputs. 🍌

We're looking for a low-lift way for customers to upload or point to things like markdown files, which document secret management and remediation processes.

**Additional Context:**

Specific use case: The secret scanning team is experimenting with AI-based remediation and autofix behavior for alerts that are generated by leaked secrets. Often, remediation involves storing secrets in a vault, and then referencing the vault key in the codebase, instead of the raw secret.

Today, we don't have a good way of identifying what kind of vault is being used by the code in an org/repo. We think Copilot KBs will be useful in this scenario. If users can be prompted into documenting remediation processes in KB files, we would be able to determine that additional context and augment our prompt with it,

**Assignees**

- monalisa
- dumpling
- crumpet

**Labels**

- user story
- feature request

**Projects**

- Cybercats Squad Planning (Awaiting triage)

**Notifications**

Subscribe

You're not receiving notifications from this issue

**3 participants**

- monalisa
- dumpling
- crumpet

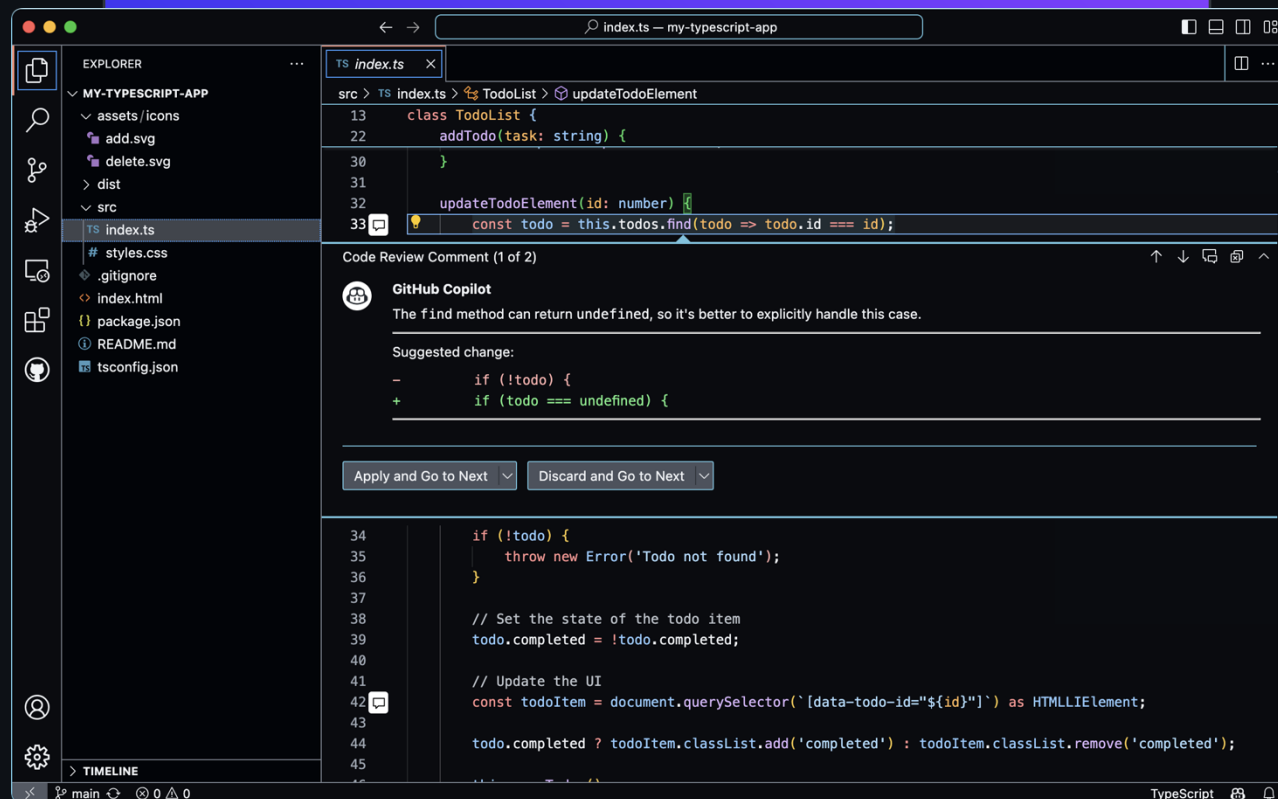
**Actions**

- Lock conversation
- Pin issue
- Transfer issue
- Delete issue

Public Beta

# VS Code 内の Copilot コード フィードバック

コーディング中にリアルタイムで AI から  
コードに対するフィードバックを受ける  
エディターの中から直接開発を加速させ、  
コード品質を向上





General Availability

# GitHub Copilot コード出典表示 & Azure 上の コード出典 API

AI による提案が公開コードと一致する場合、  
そのコードの出典内容の確認が可能に

ワークフロー内で直接コードの出典元や  
ライセンス情報等を確認し、情報に基づ  
いた判断が可能に

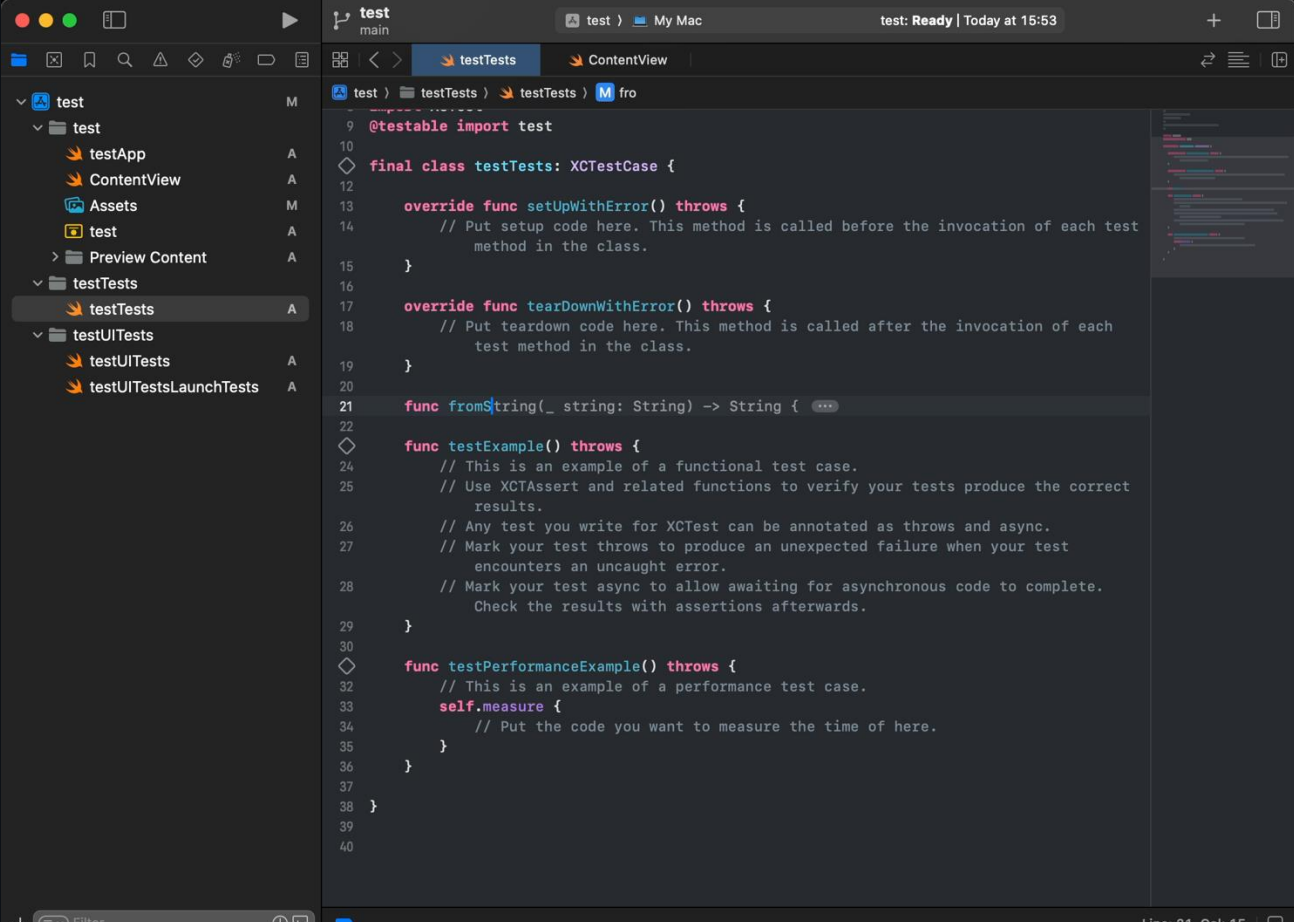
```
# Code Citations Untitled-1
1 # Code Citations
2
3 ## License: GPL_3_0
4 https://github.com/MuslehUddin007/Cpp_program/tree/a4c0685d52084d1b1be2aeb6c2f9213315712d75/MergeSort.cpp
5
6 ...
7 for (int i = 0; i < n1; i++)
8     L[i] = arr[left + i];
9     for (int j = 0; j < n2; j++)
10        R[j] = arr[mid + 1 + j]
11
12 ...
13
14 ## License: unknown
15 https://github.com/kanijsheikh/algorithm/tree/de4b973ab69e0eb67748daba8427a290e4c98330/src/main/java/sorting/
MergeSort.java
16
17 ...
18 = 0, k = left;
19 while (i < n1 && j < n2) {
20     if (L[i] <= R[j]) {
21         arr[k] = L[i];
22         i++;
23     } else {
24         arr[k] =
25     }
26
27 ...
28
29 ## License: unknown
30 https://github.com/takasidk/cpp/tree/a6d967cf80d25bc978b0fdd23e2e3ee89d8752e0/cpp_algorithms/searchAndSort/
mergeSort.cpp
31
32 ...
33 n1 && j < n2) {
34     if (L[i] <= R[j]) {
35         arr[k] = L[i];
36         i++;
37     } else {
38         arr[k] = R[j];
39         j++;
40     }
41     k
42
43 ...
44
45 ## License: unknown
46 https://github.com/emilykackley/Sorting-Algorithms/tree/26e4f67ff704b2c3fcc1ecff71cb08e93c0e9cb9/MergeSort/main.
CPP
```

Public Beta

# Xcode 向け GitHub Copilot

AIによるコード補完機能が Apple 製品向けの開発エンジニアへ提供開始

macOS, iPhone, iPad 向けのアプリ開発時に生産性を向上させ、開発を加速



The screenshot shows the Xcode IDE interface. On the left is the Project Navigator showing a project structure with folders like 'test', 'testApp', 'ContentView', 'Assets', 'test', 'Preview Content', 'testTests', 'testUITests', and 'testUITestsLaunchTests'. The main editor displays a Swift file named 'testTests.swift' with the following code:

```
9 @testable import test
10
11 final class testTests: XCTestCase {
12
13     override func setUpWithError() throws {
14         // Put setup code here. This method is called before the invocation of each test
15         // method in the class.
16     }
17
18     override func tearDownWithError() throws {
19         // Put teardown code here. This method is called after the invocation of each
20         // test method in the class.
21     }
22
23     func fromString(_ string: String) -> String { ...
24
25     func testExample() throws {
26         // This is an example of a functional test case.
27         // Use XCTAssert and related functions to verify your tests produce the correct
28         // results.
29         // Any test you write for XCTest can be annotated as throws and async.
30         // Mark your test throws to produce an unexpected failure when your test
31         // encounters an uncaught error.
32         // Mark your test async to allow awaiting for asynchronous code to complete.
33         // Check the results with assertions afterwards.
34     }
35
36     func testPerformanceExample() throws {
37         // This is an example of a performance test case.
38         self.measure {
39             // Put the code you want to measure the time of here.
40         }
41     }
42 }
```

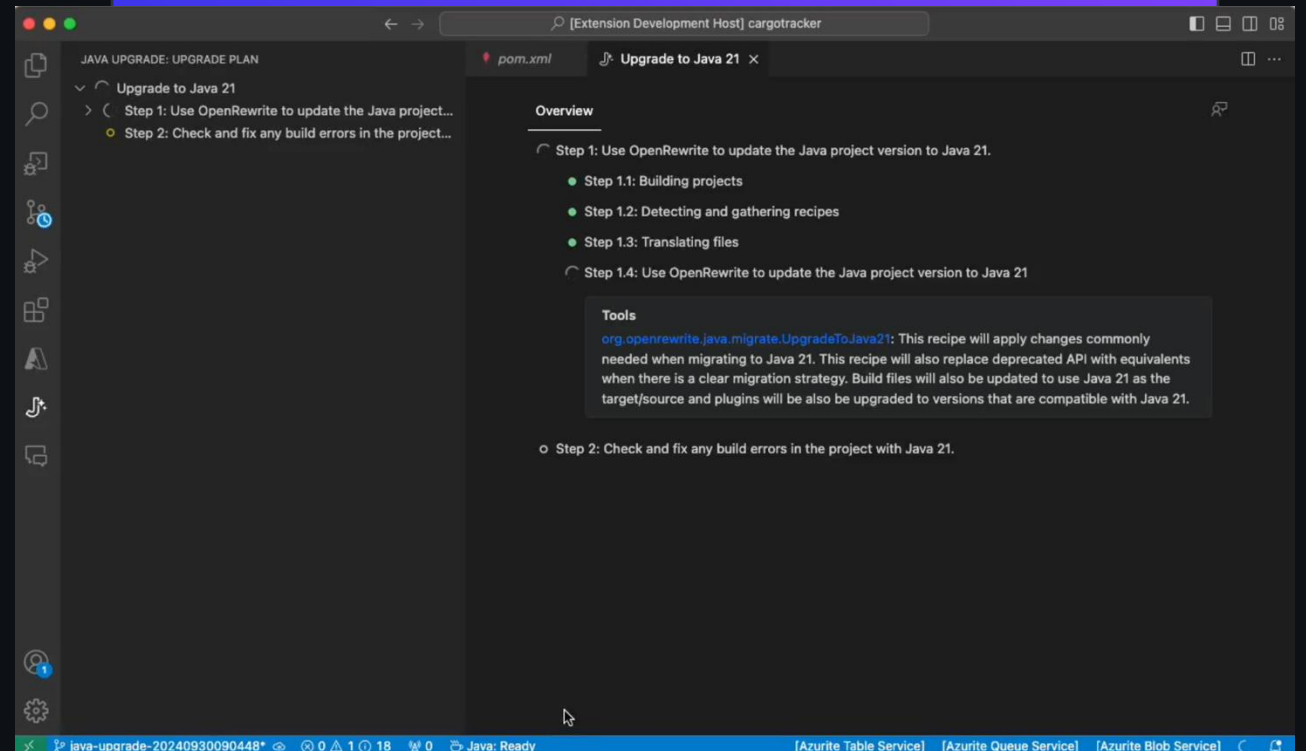
The status bar at the bottom indicates 'Line: 21 Col: 15'.

Alpha

# Java バージョン アップ VS Code 拡張機能

AI の支援を受けながら簡単に Java アプリケーションをモダナイズ

IDE 内で直接 Java 間のバージョンアップ作業を自動化させることでレガシーコードの移行やコスト削減を実現

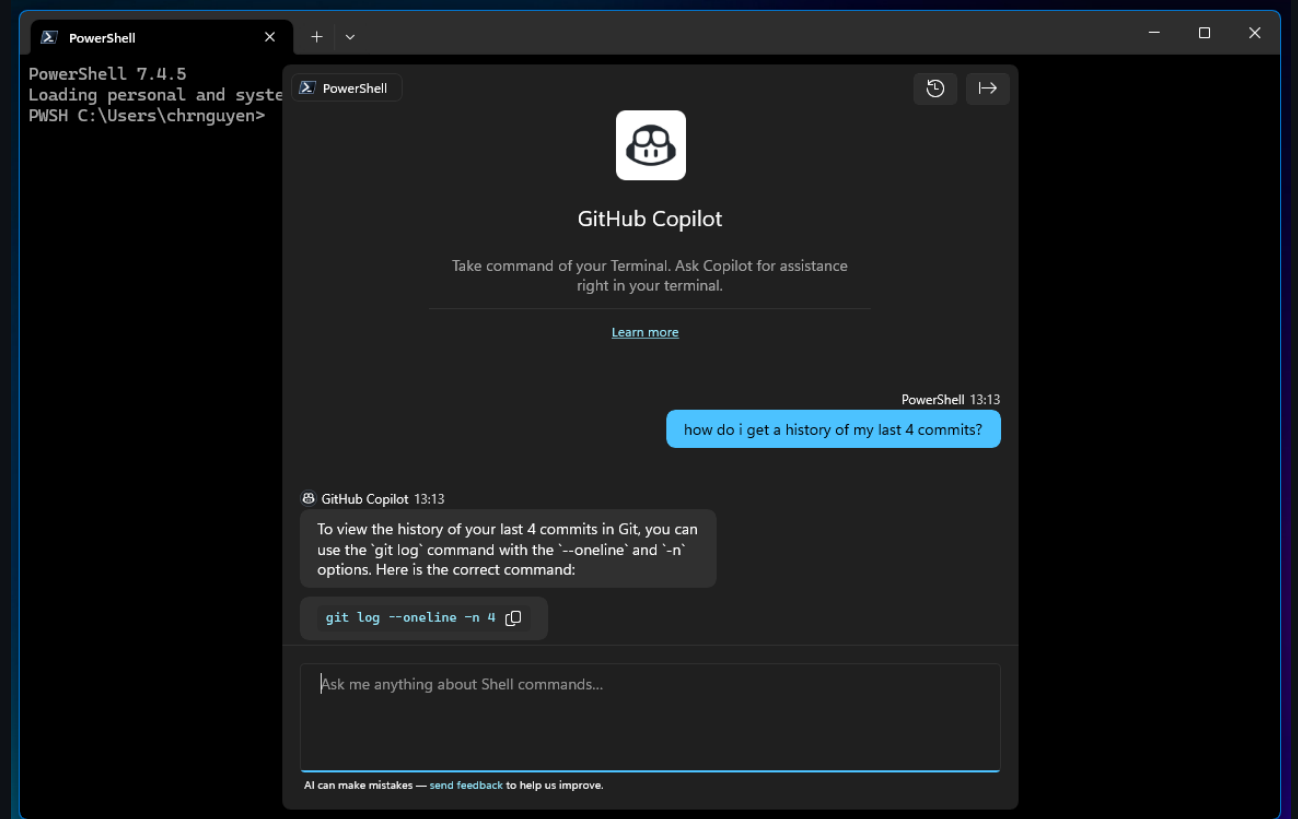


Public Beta

# Windows Terminal 向け GitHub Copilot Chat

コマンドラインの画面にも AI 支援の力を発揮

Windows Terminal 内から直接 Copilot Chat を操作することで生産性を向上し、開発プロセスを簡略化



# Thank you!

