

GitHub Copilot Updates

Luca Congiu

Microsoft Cloud Solution Architect



OpenAI o1 in GitHub Copilot Chat (Public Preview)

Key Features:

- New OpenAI o1 model integrated into [GitHub Copilot Chat](#).
- Designed for complex coding challenges and edge case detection.
- Provides thoughtful, efficient solutions for developers.

Where to Access:

- Visual Studio Code & Visual Studio.
- GitHub.com and GitHub Models Playground.

Multi-Model Support:

- GitHub Copilot now supports multiple AI models (OpenAI, Anthropic, Google).
- Offers flexibility to choose the best model for specific tasks.

GitHub Mobile: GitHub Copilot Extensions Public Preview

- **Public Preview:**

GitHub Copilot Extensions now available on GitHub Mobile (iOS & Android).

- **Features:**

- Natural language interaction with third-party tools or private data.
- Support for creating private or marketplace extensions.

- **How to Join:**

- iOS: Use TestFlight to access GitHub Mobile beta.
- Android: Become a beta tester on Google Play.

- **JetBrains Support:**

Temporarily rolled back; restoration in progress.

GitHub Copilot Code Completion in Xcode (Public Preview)

Key Features:

- **Intelligent Code Suggestions:** Auto-completes Swift, Objective-C, and more.
- **Context-Aware Assistance:** Provides relevant code completions based on your existing code.
- **Seamless Xcode Integration:** Works directly within the Xcode editor, enhancing your workflow.

Benefits:

- Boosts productivity by reducing repetitive tasks.
- Improves code quality with smarter, more relevant suggestions.
- Versatile support for macOS and iOS development.

Availability:

- Currently in public preview, open for feedback and future improvements.

GitHub Copilot Code Review (Public Preview)

Key Features:

- **AI-Powered Feedback:** Get quick, AI-generated comments on your code.
- **Enhanced Iteration:** Refine your code faster while waiting for human reviews.

Benefits:

- Speeds up the code review process.
- Improves code quality with immediate, actionable insights.

Availability:

- Currently in public preview for **Copilot Individual**, **Business**, and **Enterprise** subscribers.
- Sign up for the waitlist to request access.

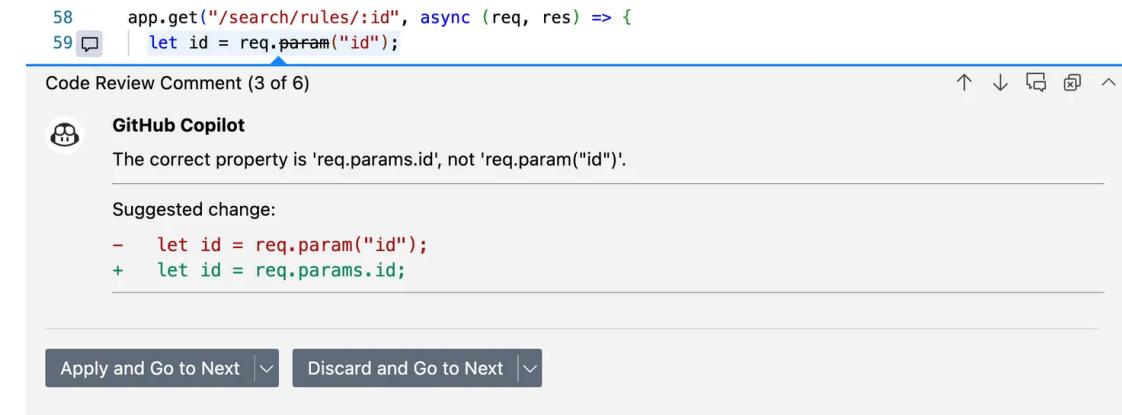
GitHub Copilot Code Review in VS Code (Public Preview)

Key Features:

- **Real-Time Feedback:** AI-generated comments as you code.
- **Pre-Commit Reviews:** Comprehensive review of all changes before pushing.
- **Actionable Suggestions:** Apply AI-suggested fixes with one click.

Benefits:

- Boosts code quality with proactive insights.
- Enhances productivity through instant, in-editor feedback.
- Seamlessly integrates into Visual Studio Code workflows.



The screenshot shows a code editor with a GitHub Copilot code review comment overlaid. The code being reviewed is in a file named 'routes.js'. The comment is: 'The correct property is 'req.params.id', not 'req.param("id")''. It includes a 'Suggested change:' section with the following diff:

```
58 app.get("/search/rules/:id", async (req, res) => {  
59 |   let id = req.param("id");  
|  
Code Review Comment (3 of 6)  
GitHub Copilot  
The correct property is 'req.params.id', not 'req.param("id")'.  
Suggested change:  
- let id = req.param("id");  
+ let id = req.params.id;  
Apply and Go to Next | Discard and Go to Next |
```

Web Search in GitHub Copilot Chat (Now Available for Copilot Individual)

Key Features:

- **Real-Time Web Search:** Access up-to-date information directly within Copilot Chat.
- **Dynamic Conversations:** Discuss recent trends, technologies, and events with AI-powered assistance.

How to Enable:

- Activate the "Copilot Access to Bing" policy in **Copilot Settings** to unlock this feature.

Benefits:

- Stay informed with the latest data and trends.
- Boost productivity by accessing information without leaving your coding environment.

GitHub Copilot in Windows Terminal (Public Beta)

Key Features:

- **AI Integration:** Intelligent code suggestions and command completions directly in Windows Terminal.
- **Streamlined Workflow:** Access Copilot without leaving the terminal, boosting productivity.

How to Access:

- Available in **Windows Terminal Canary** version.
- Requires administrative approval for **Copilot Business** or **Enterprise** users.
- Follow setup instructions in the [Terminal Chat documentation](#).

Benefits:

- Simplifies command-line tasks with AI-driven assistance.
- Enhances efficiency and reduces errors in terminal workflows.

GitHub Copilot Metrics API (General Availability)

Key Features:

- **Comprehensive Usage Insights:** Track Copilot usage across enterprises, organizations, and teams, including feature-level data and daily active users.
- **Enhanced Visibility:** Integrate Copilot metrics into your organization's reporting to assess its impact on development workflows.

Transition Notes:

- New schema introduced with the GA release.
- **Deprecated Endpoint:** The previous /usage endpoint will be retired on **January 31, 2025**.

Copilot Workspace Public Preview

Key Features

- **Integrated Code Review:**
 - Refine and apply feedback directly in pull requests.
 - Enhanced diff-viewing for better context.
- **Improved Code Editing:**
 - Address merge-blocking feedback seamlessly.
 - Access language services and Copilot completions.
- **Build and Test Environment:**
 - Build, test, and run proposed changes in an isolated setup.
 - Avoid impacting personal development environments.

Claude 3.5 Sonnet in GitHub Copilot (Public Preview)

Key Features:

- **Advanced Coding Assistance:** Excels in complex tasks like refactoring, legacy updates, and feature development.
- **Flexible Model Selection:** Users can choose Claude 3.5 Sonnet via the model selector in Copilot Chat (Visual Studio Code and GitHub.com).

Access:

- **Copilot Individual Users:** Available immediately via model selector.
- **Copilot Business/Enterprise Users:** Requires administrator activation through a new Copilot settings policy.

Build Copilot Extensions Faster with Skillsets

Key Features:

- **Simplified Integration:** Define up to 5 skill endpoints for Copilot to call.
- **No AI Expertise Needed:** Configure extensions using simple JSON schemas.
- **Rapid Development:** Streamlined setup to quickly extend Copilot's capabilities.

GitHub Copilot Now Available on GitHub Dashboard (Public Preview)

Key Features

- **Direct Access:**
 - Use GitHub Copilot without leaving the GitHub dashboard.
 - Get code suggestions and improvements right within the interface.

Copilot Chat now has a 64k context window with OpenAI GPT-4o

Key Features:

- **Expanded Context Window:** Now supports up to 64,000 tokens for better understanding of larger codebases and files.
- **Improved Large Project Support:** Handles extensive repositories and complex files more effectively.

GitHub Copilot Extensions in JetBrains IDEs

Key Features

- **AI-Powered Tools:**
 - Access context-aware suggestions.
 - Integrate with external tools for enhanced workflows.
- **Supported IDEs:**
 - IntelliJ IDEA, Android Studio, PyCharm, WebStorm, PhpStorm, CLion, and more.
- **Easy Setup:**
 - Install the GitHub Copilot plugin from JetBrains Marketplace.
 - Sign in with your GitHub account to activate.

GitHub Copilot Extensions Now Understand Context in Your Environment

Key Features:

- Enhanced Context Understanding:** Analyzes project structure, dependencies, and existing code.
- Improved Accuracy:** Provides tailored, precise code suggestions aligned with your environment.

client.file	client.selection	github.repository	github.current-url	Additional contexts
✓	✓	✓	✗	Repository owner and branch
✓	✓	✓	✗	Repository owner and branch
✗	✗	✓	✓	Repository information or other GitHub resources
✗	✗	✗	✓	✗
✗	✗	✗	✗	✗

Update to last_activity_at Retention Policy

Key Update:

- Starting February 18, 2025, GitHub will enforce a 90-day retention policy for the `last_activity_at` field in the Copilot User Management API.

Details:

- Previously, `last_activity_at` values were stored indefinitely.
- After 90 days of inactivity, `last_activity_at` will be set to `nil`.
- Users without activity since November 20, 2024, will have their data set to `nil` on a rolling basis.

Impact:

- Minimal for users relying on recent activity data.
- Users tracking dormancy (90+ days inactivity) should adjust workflows, as older records will become `nil`.

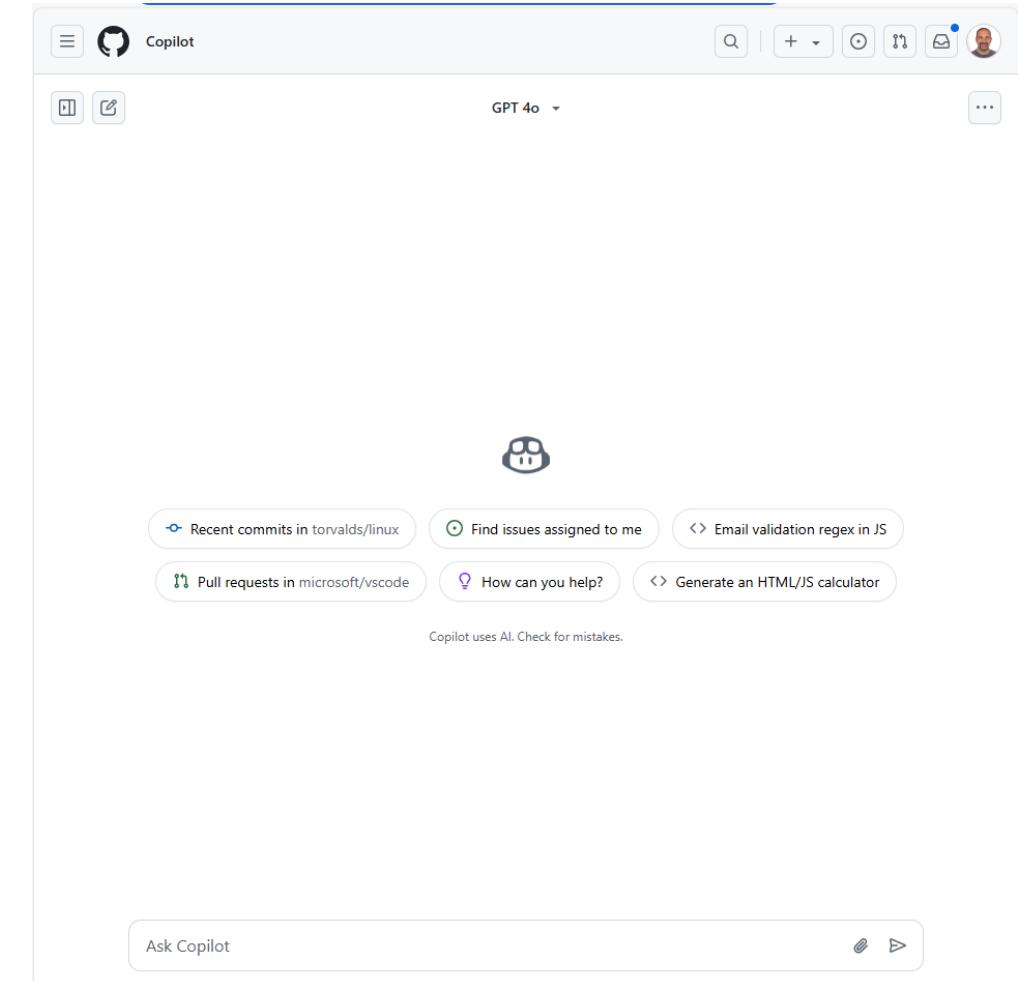
Action Steps:

- If necessary, store API responses to preserve older activity data before February 18, 2025.

GitHub Copilot Chat Now Generally Available

· Key Features:

- **AI-Powered Assistance:** Get natural language help for coding, explanations, and insights.
- **Multi-Platform Support:** Available on Visual Studio Code, Visual Studio, JetBrains IDEs, GitHub Mobile, and GitHub.com.



<https://github.com/copilot>

GitHub Copilot Free Plan

Key Features:

- **2,000 Code Completions:** Get AI-driven code suggestions each month.
- **50 Chat Messages:** Engage in AI-assisted coding conversations.
- **Available Platforms:** Visual Studio Code and GitHub.com.

GitHub Copilot Free in VS Code

Key Features:

- **Code Completions:** Receive up to 2,000 AI-generated code suggestions per month.
- **Chat Interactions:** Engage in up to 50 AI-assisted coding conversations monthly.

GitHub Copilot Free Plan Now Available for JetBrains IDEs

Key Features:

- **Free Access:** GitHub Copilot is now available in JetBrains IDEs without needing a subscription.
- **Seamless Integration:** Activate Copilot using your GitHub account for AI-powered coding assistance.

OpenAI o1 Now Available in GitHub Copilot Chat (Preview)

Key Features

- **Enhanced Performance:**
 - The new **o1 model** improves handling of complex tasks.
- **Access:**
 - Select **o1 (Preview)** from the model picker in Visual Studio Code or Copilot Chat at github.com/copilot.

Expanded Access to GitHub Copilot Workspace

Announcement:

- As of **December 30, 2024**, all paying Copilot customers can access the **Copilot Workspace** technical preview.

Key Features:

- Brainstorming:** Interact with Copilot to understand and explore your codebase.
- Planning:** Generate detailed plans for code changes across files.
- Implementation & Validation:** Receive and refine code suggestions; test changes within the workspace using GitHub Codespaces.

Benefit:

- Streamlines development from concept to merge in a Copilot-native environment.

GitHub Copilot Explains Failed Actions Jobs

Announcement:

- As of **January 15, 2025**, GitHub Copilot can help debug **failed GitHub Actions jobs** (General Availability).

Key Features:

- **Explain Error:**
 - Available in pull request merge boxes and Actions job pages.
 - Provides insights into job failures and guidance for resolution.
- **Usage:**
 - Each explanation consumes **1 chat message** (based on [Copilot plan limits](#)).

How to Access:

- In pull requests:
 - Click "**More actions**" on a failing check, then select "**Explain Error.**"
- On job pages:
 - Use the "**Explain Error**" button near the search bar.

.

GitHub Copilot Extensions: Header Update

Key Update:

- Starting March 31, 2025, GitHub Copilot Extensions must use new headers for agent requests:
 - X-GitHub-Public-Key-Identifier
 - X-GitHub-Public-Key-Signature

Details:

- Current headers (Github-Public-Key-Identifier and Github-Public-Key-Signature) will be deprecated.
- Both old and new headers are supported until the cutoff date.

Action Required:

- Update extensions to use the new headers before March 31, 2025, to avoid disruptions.

Custom Repository Instructions for GitHub Copilot (Public Preview)

Key Features

- **Personalize Copilot Behavior:**
 - Define coding standards and project preferences through custom instructions.
- **Easy Setup:**
 - Create a `.github/copilot-instructions.md` file in your repository.
 - Add instructions for Copilot to follow within that repository.

Copilot in GitHub.com now supports Content Exclusions (Preview)

- **New Feature** – Admins can exclude specific files from Copilot suggestions
- **Customizable** – Supports filename patterns (e.g., secrets.json, *.cfg, /scripts/**)
- **Who Can Use It?** – Available for Copilot Business & Enterprise users

Copilot in GitHub.com now supports Content Exclusions (Preview)

```
# Ignore all `*.env` files from all file system roots (Git and non-Git).
# For example, this excludes `REPOSITORY-PATH/.env` and also `./.env`.
# This could also have been written on a single line as:
#
# """": ["""*.env"""
#"""
# - """*.env"""

# In the `octo-repo` repository in this organization:
octo-repo:
  # Ignore the `/src/some-dir/kernel.rs` file.
  - "/src/some-dir/kernel.rs"

# In the `primer/react` repository on GitHub:
https://github.com/primer/react.git:
  # Ignore files called `secrets.json` anywhere in this repository.
  - "secrets.json"
  # Ignore files called `temp.rb` in or below the `/src` directory.
  - "/src/**/temp.rb"

# In the `copilot` repository of any GitHub organization:
git@github.com:*/copilot:
  # Ignore any files in or below the `/_tests_` directory.
  - "/_tests_/**"
  # Ignore any files in the `/scripts` directory.
  - "/scripts/*"
```

OpenAI o3-mini now available in GitHub Copilot and GitHub Models (Public Preview)

- **New Model:** OpenAI o3-mini now in GitHub Copilot & GitHub Models (Public Preview).
- **Performance Boost:** Better coding benchmarks than o1, with response times similar to o1-mini.
- **Availability:** Live in public preview for improved AI-powered coding assistance.

Security improvements with Copilot Autofix (public preview)

Fix Security campaigns with Copilot Autofix

Log entries created from user input

Open in `main` 1 hour ago

Speed up the remediation of this alert with [Copilot Autofix for CodeQL](#)

[Generate fix](#)

```
src/api/Endpoints/TodoApiEndpoints.cs:104
101     var updatedTodo = await repository.UpdateTodoAsync(todo);
102     if (updatedTodo == null)
103     {
104         logger.LogWarning("Todo with id {id} not found for update.", id);
This log entry depends on a user-provided value.
CodeQL Show paths
105         return Results.NotFound();
106     }
107 }
```

Tool	Rule ID	Query
CodeQL	cs/log-forging	View source

If unsanitized user input is written to a log entry, a malicious user may be able to forge new log entries.

Forgery can occur if a user provides some input with characters that are interpreted when the log output is displayed. If the log is displayed as a plain text file, then new line characters can be used by a malicious user. If the log is displayed as HTML, then arbitrary HTML may be included to spoof log entries.

Recommendation

User input should be suitably encoded before it is logged.

If the log entries are plain text then line breaks should be removed from user input, using `String.Replace` or similar. Care should also be taken that user input is clearly marked in log entries, and that a malicious user cannot cause confusion in other ways.

For log entries that will be displayed in HTML, user input should be HTML encoded using `HttpUtility.HtmlEncode` or similar before being logged, to prevent forgery and other forms of HTML injection.

Dismiss alert [Create issue](#)

Severity
High

Affected branches
`main`

Tags
`security`

Weaknesses
[CWE-117](#)

[Improper Output Neutralization for Logs](#)
The product does not neutralize or incorrectly neutralizes output that is written to logs.

GitHub Copilot Workspace Enhancements

- **Auto-validation** – Automatic build and test execution after implementation, with error repair suggestions
- **Go to Definition** – Enhanced code navigation in the editor
- **File-Specific Plans** – View and modify file plans directly above the file
- **Dynamic File Tree** – Real-time updates to the file tree based on plan changes
- **GitHub Enterprise Cloud Support** – Available for managed users with enterprise control

View and iterate on generated files directly within Copilot chat (Preview)

- **Quick file iteration:** When Copilot chat suggests files, they open immediately for closer inspection.
- **Convenient previews:** Markdown files render directly inside the side panel.
- **Support for more formats:** Expanded support for different file formats.

Google Gemini 2.0 Flash is now available to all Copilot users in public preview

Announced at GitHub Universe 2024, Gemini 2.0 Flash is now available to all GitHub Copilot customers, including Copilot Free.

The latest Gemini model from Google shows high capabilities for code suggestions, documentation, and explaining code

OpenAI o3-mini is now available in GitHub Copilot Free

- OpenAI's latest model, o3-mini, is the most cost-efficient model in their reasoning series.
- o3-mini outperforms o1 on coding benchmarks with response times that are comparable to o1-mini
- Access to o3-mini is currently in preview and is subject to the 50 free chats per month limit when using Copilot Free.

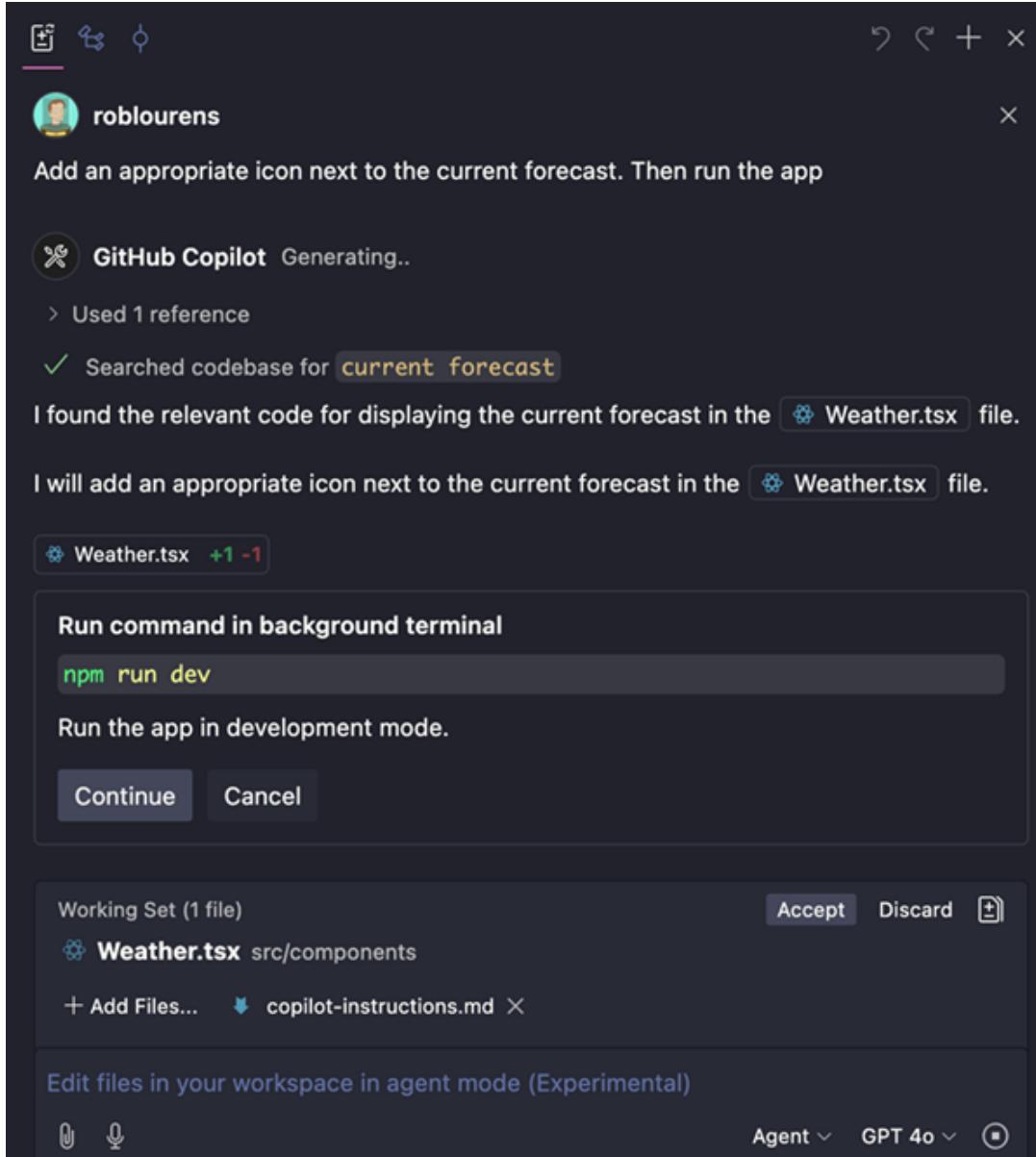


GitHub Copilot

AGENT MODE



Agent mode (public preview) for GitHub Copilot in VS Code

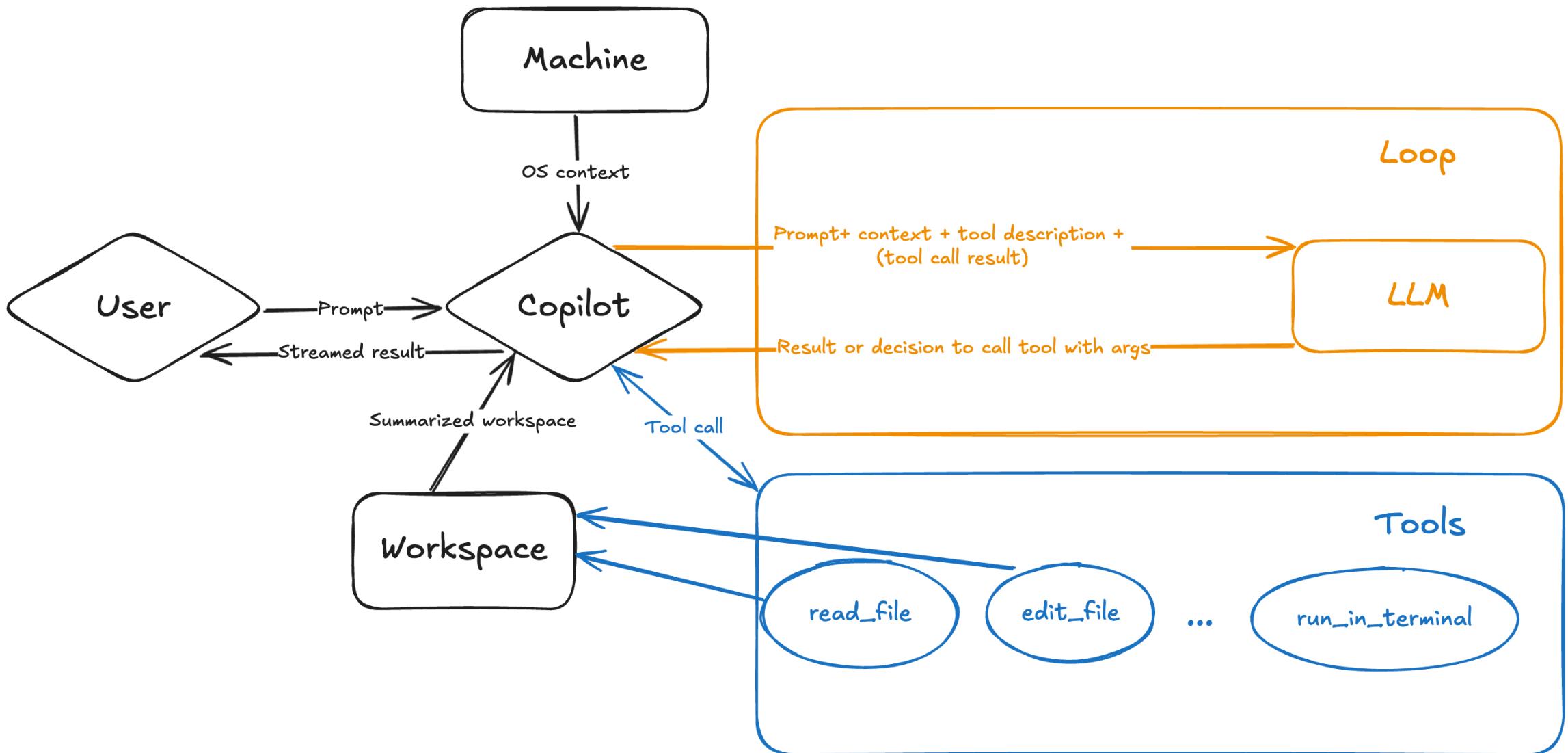


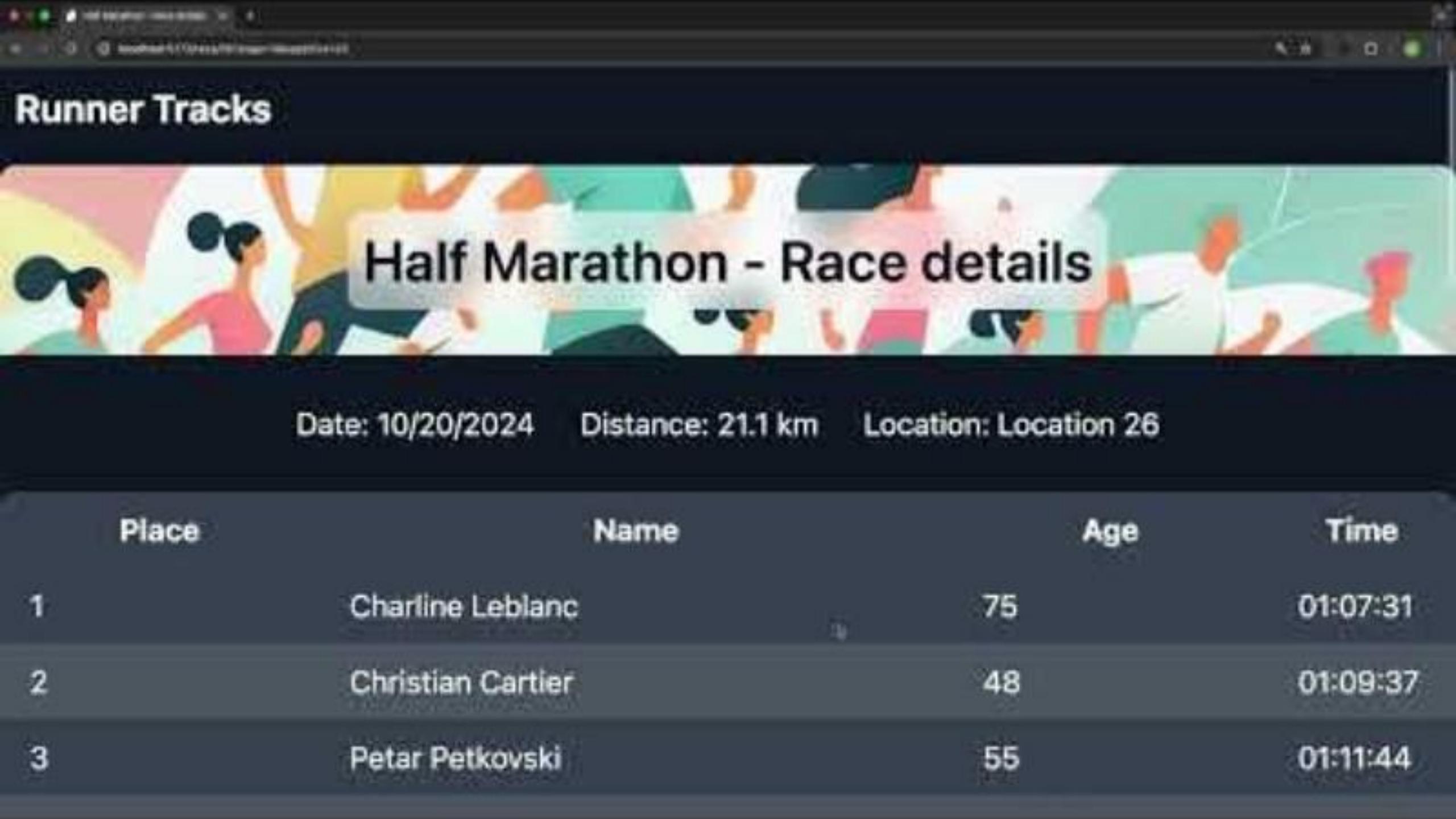
In agent mode, Copilot can iterate on its own code. This includes automatically identifying and fixing errors, executing terminal commands to complete a requested task, and resolving runtime errors with self-healing capabilities.

GitHub Copilot can now infer the other tasks that were not specified and also need to be completed in order for your explicit request to work.

To activate: `github.copilot.chat.agent.enabled: true`

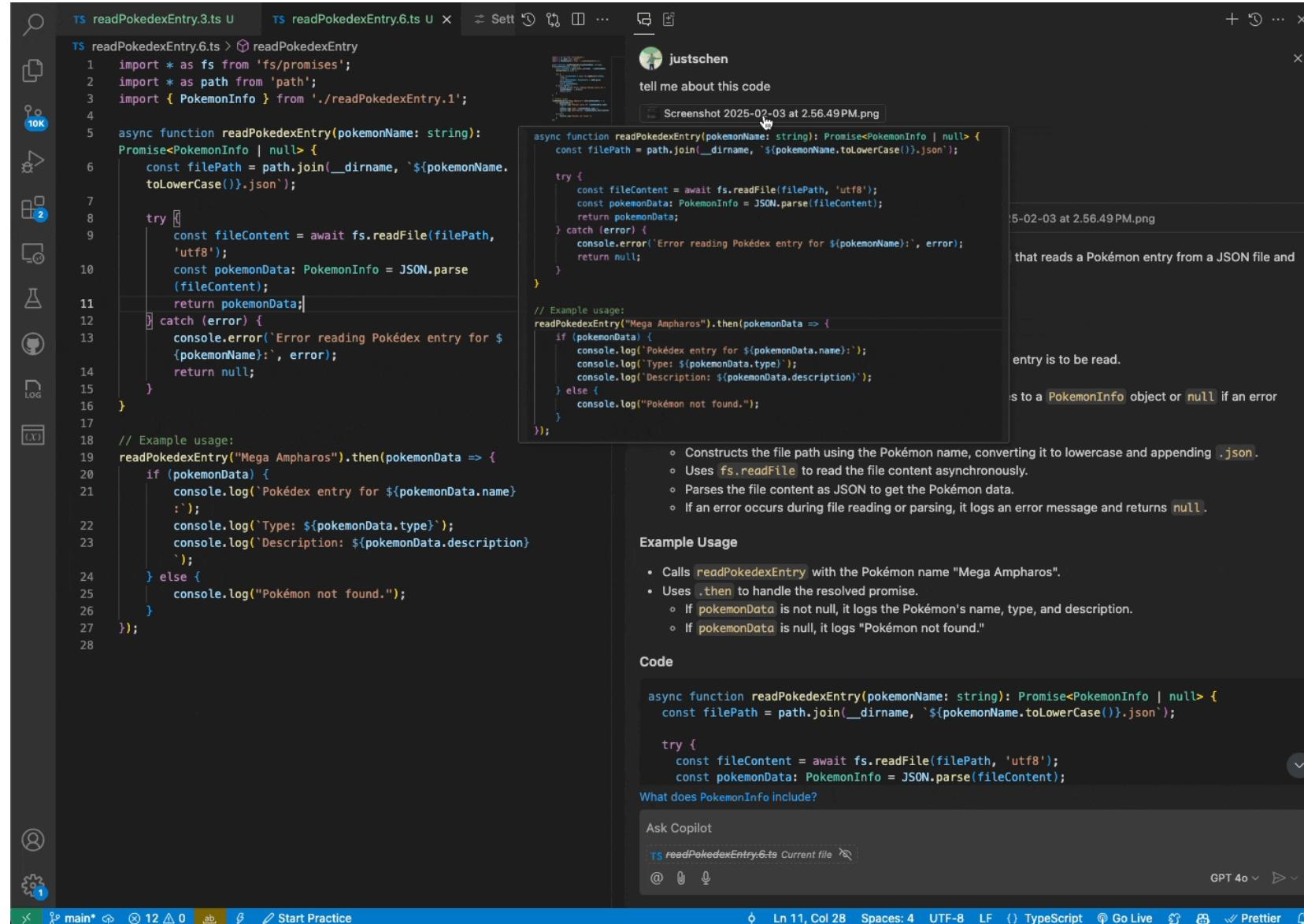
How it works





Place	Name	Age	Time
1	Charline Leblanc	75	01:07:31
2	Christian Cartier	48	01:09:37
3	Petar Petkovski	55	01:11:44

Vision (public preview) for GitHub Copilot in VS Code



Code editor showing `readPokedexEntry.3.ts` and `readPokedexEntry.6.ts`. The `readPokedexEntry.6.ts` file contains the following code:

```
import * as fs from 'fs/promises';
import * as path from 'path';
import { PokemonInfo } from './readPokedexEntry.1';

async function readPokedexEntry(pokemonName: string): Promise<PokemonInfo | null> {
  const filePath = path.join(__dirname, `${pokemonName.toLowerCase()}.json`);

  try {
    const fileContent = await fs.readFile(filePath, 'utf8');
    const pokemonData: PokemonInfo = JSON.parse(fileContent);
    return pokemonData;
  } catch (error) {
    console.error(`Error reading Pokédex entry for ${pokemonName}`, error);
    return null;
  }
}

// Example usage:
readPokedexEntry("Mega Ampharos").then(pokemonData => {
  if (pokemonData) {
    console.log(`Pokédex entry for ${pokemonData.name}`);
    console.log(`Type: ${pokemonData.type}`);
    console.log(`Description: ${pokemonData.description}`);
  } else {
    console.log("Pokémon not found.");
  }
});
```

The central panel shows a screenshot of the code and a generated text summary:

`justschen`
tell me about this code
Screenshot 2025-02-03 at 2.56.49PM.png

```
async function readPokedexEntry(pokemonName: string): Promise<PokemonInfo | null> {
  const filePath = path.join(__dirname, `${pokemonName.toLowerCase()}.json`);

  try {
    const fileContent = await fs.readFile(filePath, 'utf8');
    const pokemonData: PokemonInfo = JSON.parse(fileContent);
    return pokemonData;
  } catch (error) {
    console.error(`Error reading Pokédex entry for ${pokemonName}`, error);
    return null;
  }
}

// Example usage:
readPokedexEntry("Mega Ampharos").then(pokemonData => {
  if (pokemonData) {
    console.log(`Pokédex entry for ${pokemonData.name}`);
    console.log(`Type: ${pokemonData.type}`);
    console.log(`Description: ${pokemonData.description}`);
  } else {
    console.log("Pokémon not found.");
  }
});
```

that reads a Pokémon entry from a JSON file and entry is to be read. is to a `PokemonInfo` object or `null` if an error

• Constructs the file path using the Pokémon name, converting it to lowercase and appending `.json`.
• Uses `fs.readFile` to read the file content asynchronously.
• Parses the file content as JSON to get the Pokémon data.
• If an error occurs during file reading or parsing, it logs an error message and returns `null`.

Example Usage

- Calls `readPokedexEntry` with the Pokémon name "Mega Ampharos".
- Uses `.then` to handle the resolved promise.
 - If `pokemonData` is not null, it logs the Pokémon's name, type, and description.
 - If `pokemonData` is null, it logs "Pokémon not found."

Code

```
async function readPokedexEntry(pokemonName: string): Promise<PokemonInfo | null> {
  const filePath = path.join(__dirname, `${pokemonName.toLowerCase()}.json`);

  try {
    const fileContent = await fs.readFile(filePath, 'utf8');
    const pokemonData: PokemonInfo = JSON.parse(fileContent);
    return pokemonData;
  } catch (error) {
    console.error(`Error reading Pokédex entry for ${pokemonName}`, error);
    return null;
  }
}
```

What does `PokemonInfo` include?

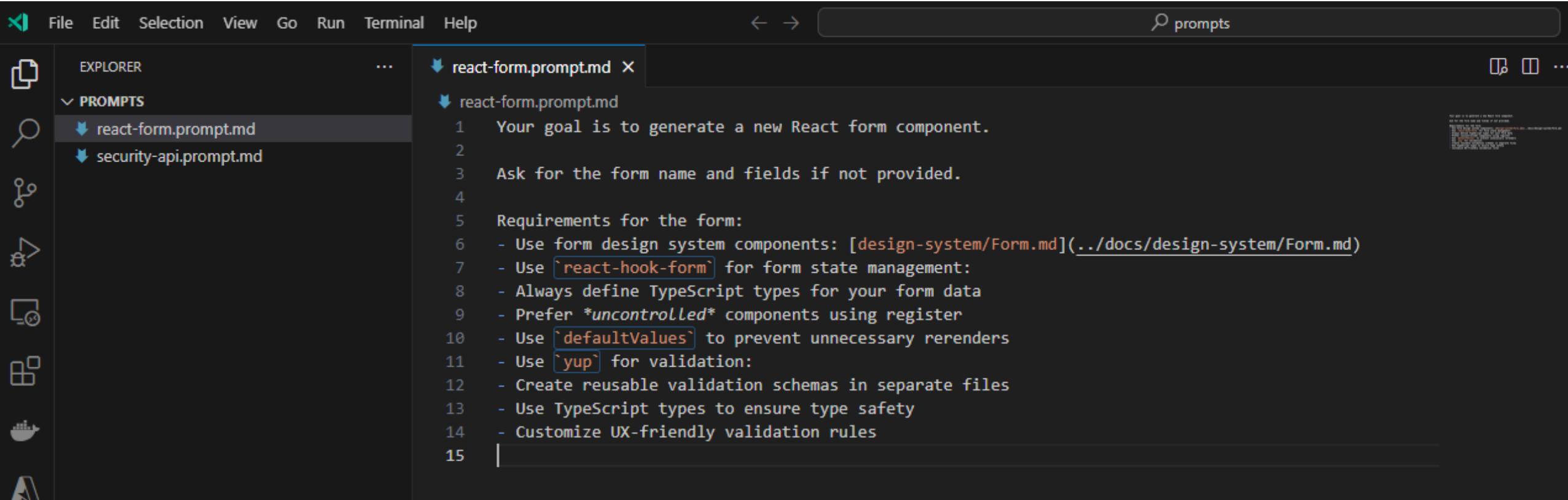
Ask Copilot
TS readPokedexEntry.6.ts Current file

With Vision, you can share screenshots of errors, and Copilot will interpret the image to help resolve the issue. Additionally, you can share mockups of new designs, and Vision will assist in bringing them to life.

Vision requires the use of the GPT-4o model and supports the following file types: JPEG/JPG, PNG, GIF, and WEBP

Prompt files (public preview) for GitHub Copilot in VS Code

Prompt files allow you and your team to build, store, and share reusable prompts. They contain pre-defined instructions and context for GitHub Copilot Chat and Copilot Edits, which will help you save time, maintain consistency, and support the sharing of best practices across workflows for shared development tasks and domains.



Custom instructions for GitHub Copilot in VS Code

- **Code-generation instructions:** Provide context for generating code, such as prefixed private variables or specific implementation styles. These can be specified in settings or a Markdown file in your workspace.
- **Test-generation instructions:** Provide context for generating tests, such as using a specific testing framework. These can also be specified in settings or a Markdown file.
- **Code review instructions:** Provide context for reviewing the current editor selection, such as looking for specific types of errors. These can be specified in settings or a Markdown file.
- **Commit message generation instructions:** Provide context for generating commit messages. These can be specified in settings or a Markdown file.

Code-generation custom instructions

[github.copilot.chat.codeGeneration.instructions](#)

```
"github.copilot.chat.codeGeneration.instructions": [
  {
    "text": "Always add a comment: 'Generated by Copilot'."
  },
  {
    "text": "In TypeScript always use underscore for private field names."
  },
  {
    "file": "code-style.md" // import instructions from file `code-style.md`
  }
],
```

code-style.md file

Always use React functional components.

Always add comments.

Test-generation custom instructions

[github.copilot.chat.testGeneration.instructions](#)

```
"github.copilot.chat.testGeneration.instructions": [  
  {  
    "text": "Always use vitest for testing React components."  
  },  
  {  
    "text": "Use Jest for testing JavaScript code."  
  },  
  {  
    "file": "code-style.md" // import instructions from file `code-style.md`  
  }  
],
```

code-style.md file

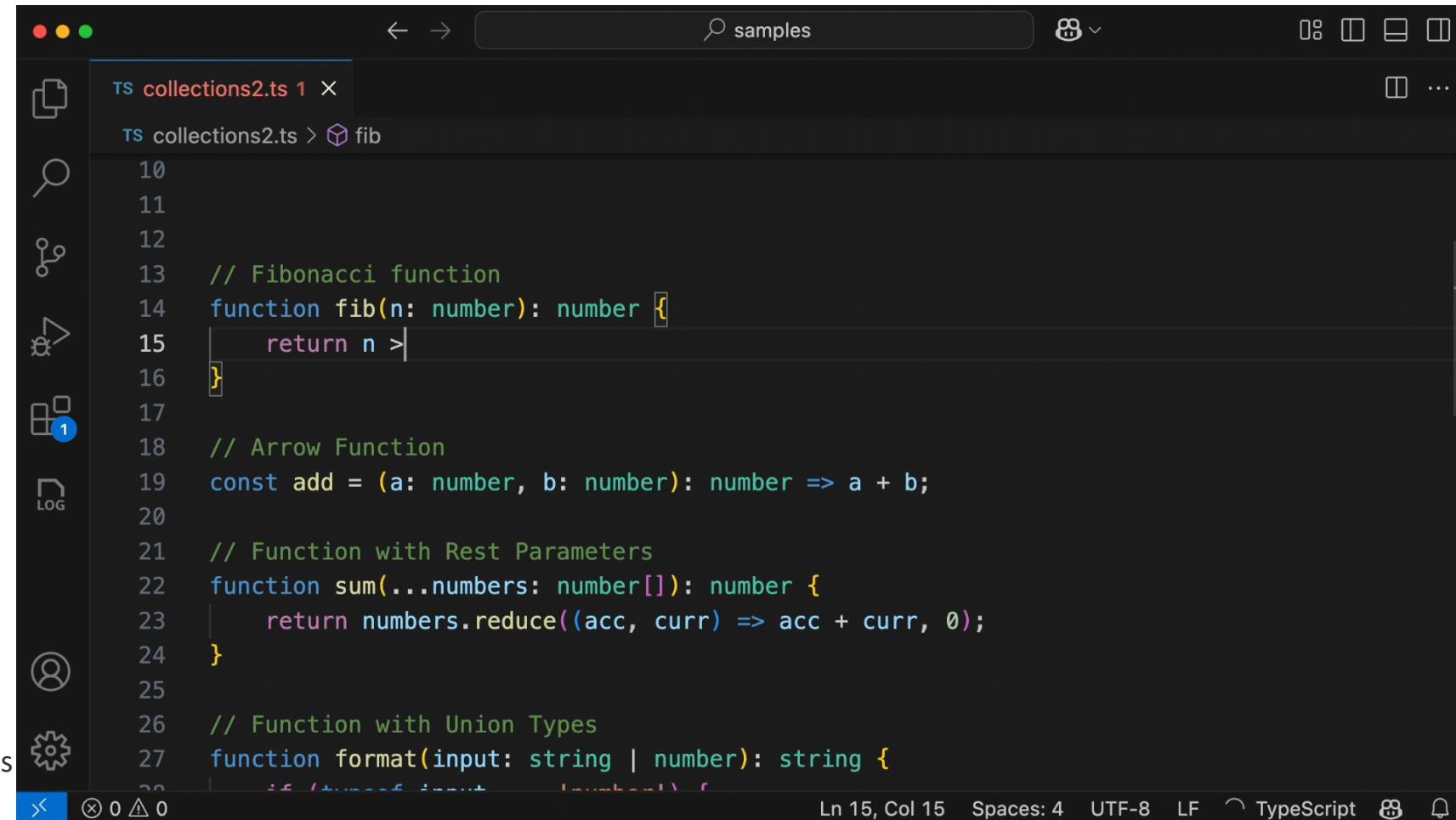
```
Always use React functional components.
```

```
Always add comments.
```

Next edit suggestions (public preview) for GitHub Copilot in VS Code

Copilot NES both predicts the location of the next edit you'll want to make and what that edit should be. Copilot NES helps you stay in the flow, suggesting future changes relevant to your current work, and you can simply Tab to quickly navigate and accept Copilot's suggestions

Enable: Copilot NES both predicts the location of the next edit you'll want to make and what that edit should be. Copilot NES helps you stay in the flow, suggesting future changes relevant to your current work, and you can simply Tab to quickly navigate and accept Copilot's suggestions



The screenshot shows a VS Code editor window with a dark theme. The left sidebar has icons for file, search, file history, and a tab with a '1' (likely Copilot). The main editor area has a file named 'TS collections2.ts' with the following code:

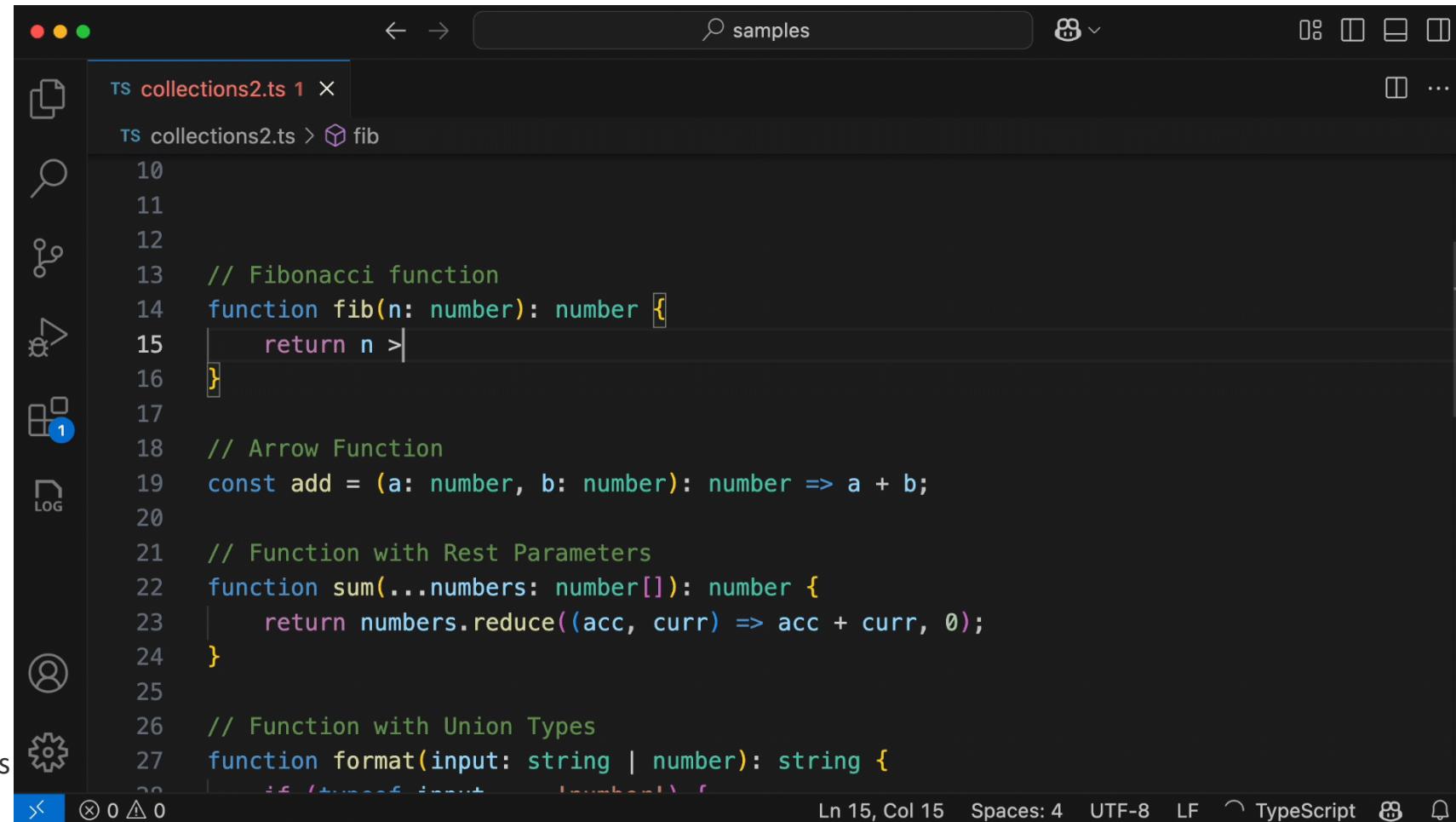
```
10
11
12
13 // Fibonacci function
14 function fib(n: number): number {
15     return n >
16 }
17
18 // Arrow Function
19 const add = (a: number, b: number): number => a + b;
20
21 // Function with Rest Parameters
22 function sum(...numbers: number[]): number {
23     return numbers.reduce((acc, curr) => acc + curr, 0);
24 }
25
26 // Function with Union Types
27 function format(input: string | number): string {
28     if (typeof input === 'number') {
29         return input.toFixed(2);
30     }
31     return input;
32 }
```

The cursor is at line 15, column 15, where the suggestion 'return n >' is shown. The status bar at the bottom right shows 'Ln 15, Col 15' and 'TypeScript'.

Next edit suggestions (public preview) for GitHub Copilot in VS Code

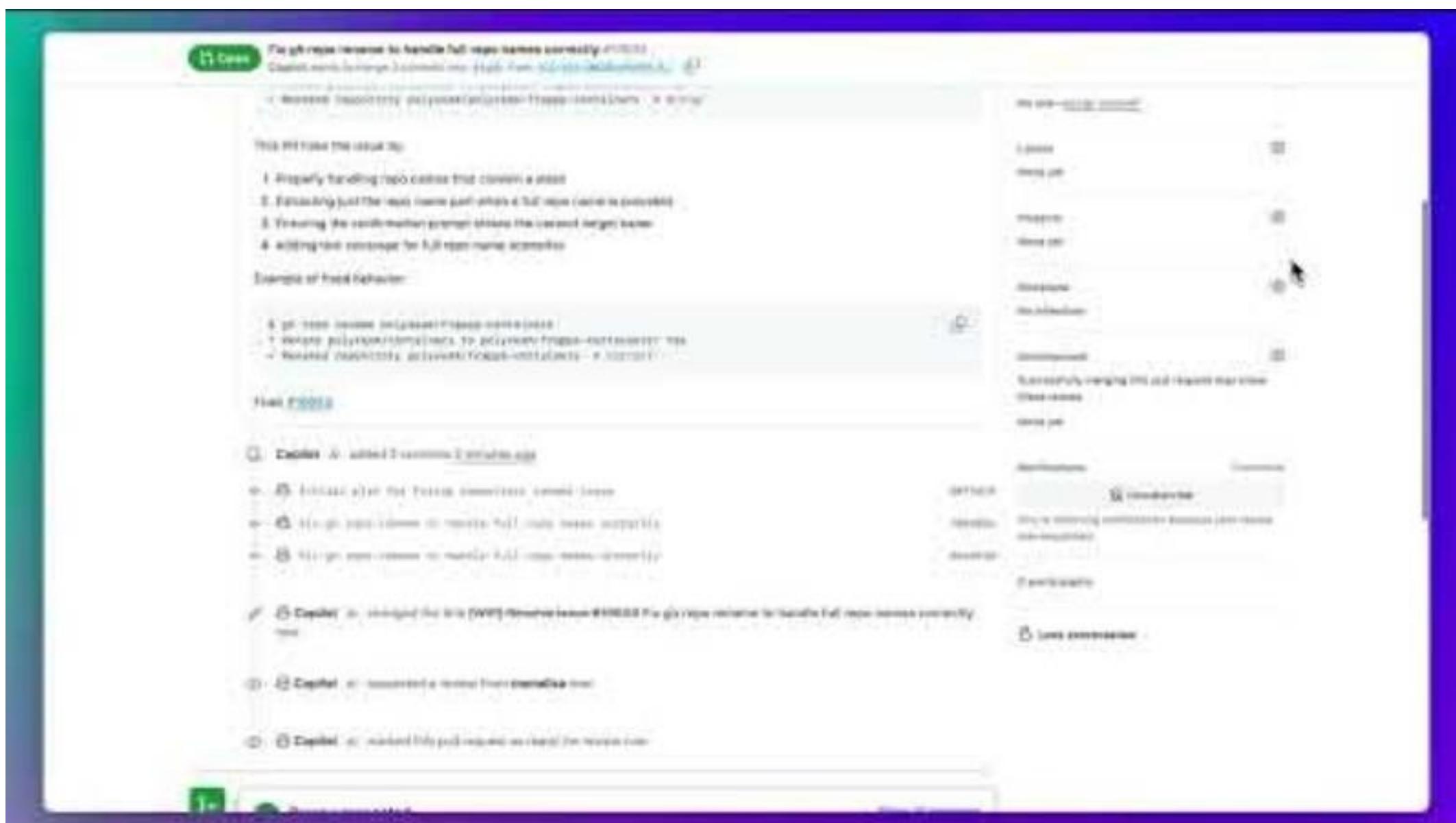
Copilot NES both predicts the location of the next edit you'll want to make and what that edit should be. Copilot NES helps you stay in the flow, suggesting future changes relevant to your current work, and you can simply Tab to quickly navigate and accept Copilot's suggestions

Enable: Copilot NES both predicts the location of the next edit you'll want to make and what that edit should be. Copilot NES helps you stay in the flow, suggesting future changes relevant to your current work, and you can simply Tab to quickly navigate and accept Copilot's suggestions



A screenshot of the VS Code interface showing a TypeScript file named 'collections2.ts'. The code contains several functions: a Fibonacci function using a regular function, an Arrow Function, a function with Rest Parameters, and a function with Union Types. A code completion suggestion for a Union Type is shown at the bottom of the file, with the cursor on the word 'number'. The VS Code interface includes a sidebar with icons for file, search, and other tools, and a status bar at the bottom showing file details like 'Ln 15, Col 15' and 'TypeScript'.

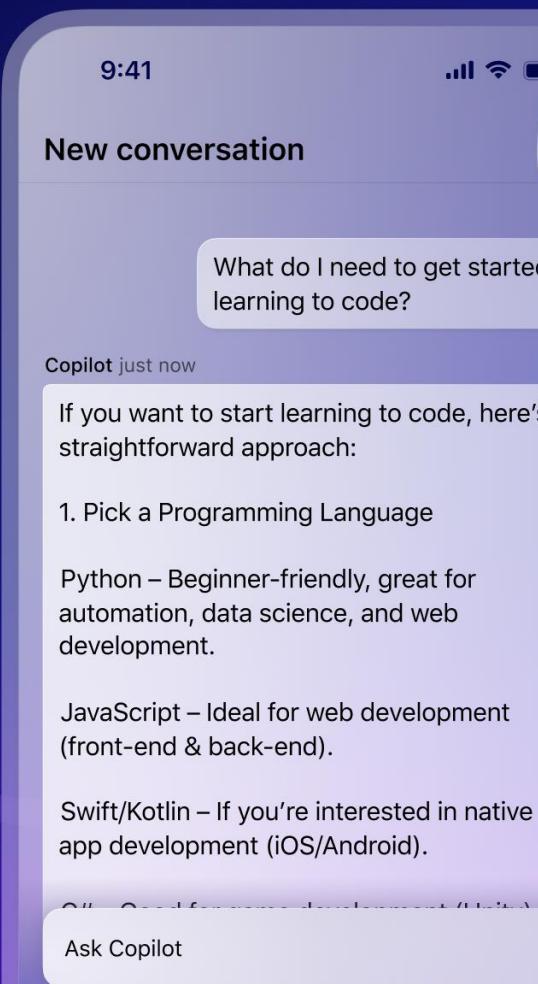
Project Padawan: SWE agents on GitHub



Copilot Language Server SDK is now available



GitHub Copilot now available for free in mobile and the CLI



The image shows a mobile phone screen displaying the GitHub Copilot app. The screen has a dark blue background with a large, stylized white 'GitHub Copilot' logo on the left. To the right of the logo, the text 'Now free in Mobile and the CLI.' is displayed in a large, white, sans-serif font. On the right side of the screen, there is a large, colorful, cartoonish character of GitHub Copilot, which is a blue and purple robot-like head with a smiling face and large, expressive eyes. The character is partially visible, with its head and shoulders showing. The main content area of the app shows a 'New conversation' screen. A user message bubble contains the text 'What do I need to get started learning to code?'. A response from 'Copilot just now' suggests starting with Python, followed by a list of three programming languages: Python, JavaScript, and Swift/Kotlin, each with a brief description of its use cases. At the bottom of the screen, there is a button labeled 'Ask Copilot'.

GitHub Copilot

Now free in Mobile and the CLI.

9:41

New conversation

What do I need to get started learning to code?

Copilot just now

If you want to start learning to code, here's a straightforward approach:

1. Pick a Programming Language

Python – Beginner-friendly, great for automation, data science, and web development.

JavaScript – Ideal for web development (front-end & back-end).

Swift/Kotlin – If you're interested in native app development (iOS/Android).

Ask Copilot

GitHub Copilot code completions now in public preview for Eclipse

- GitHub Copilot code completions are now in public preview for Eclipse
- Provides real-time, context-aware code suggestions as you type
- Supports multiple programming languages, including Java, Python, and C++
- Free tier includes 2,000 code completions and 50 chat messages per month

Code completion in GitHub Copilot for Xcode is now generally available

```
39         self.initial = initial
40         self.surname = surname
41     }
42 }
43
44 extension SampleData {
45     @MainActor static let people = [
46         Person(id: 0, initial: "A", surname: "Cloud"),
47         Person(id: 1, initial: "A", surname: "Leaf"),
48         Person(id: 2, initial: "A", surname: "Tent"),
49         Person(id: 3, initial: "B", surname: "Misty"),
50         Person(id: 4, initial: "B", surname: "Skies"),
51         Person(id: 5, initial: "D", surname: "Lily"),
52         Person(id: 6, initial: "E", surname: "Reed"),
53         Person(id: 7, initial: "F", surname: "Log"),
54         Person(id: 8, initial: "G", surname: "Grass"),
55         Person(id: 9, initial: "G", surname: "Run"),
```

Copilot on github.com enhancements

- **Personal Custom Instructions:** Users can specify preferences like language, response style, and code standards for GitHub Copilot.
- **Bing Web Search Integration:** Copilot can now fetch information beyond its general knowledge using Bing web search.
- **Improved Search Relevance:** Enhancements have been made to improve the relevance of search results.
- **Enhanced Memory for Chat History:** Copilot now has better memory for chat history, improving context awareness.
- **README.md Awareness:** Copilot can now recognize and utilize information from README.md files.
- **Direct Code File Interaction:** Users can view and generate new code files directly within Copilot Chat.

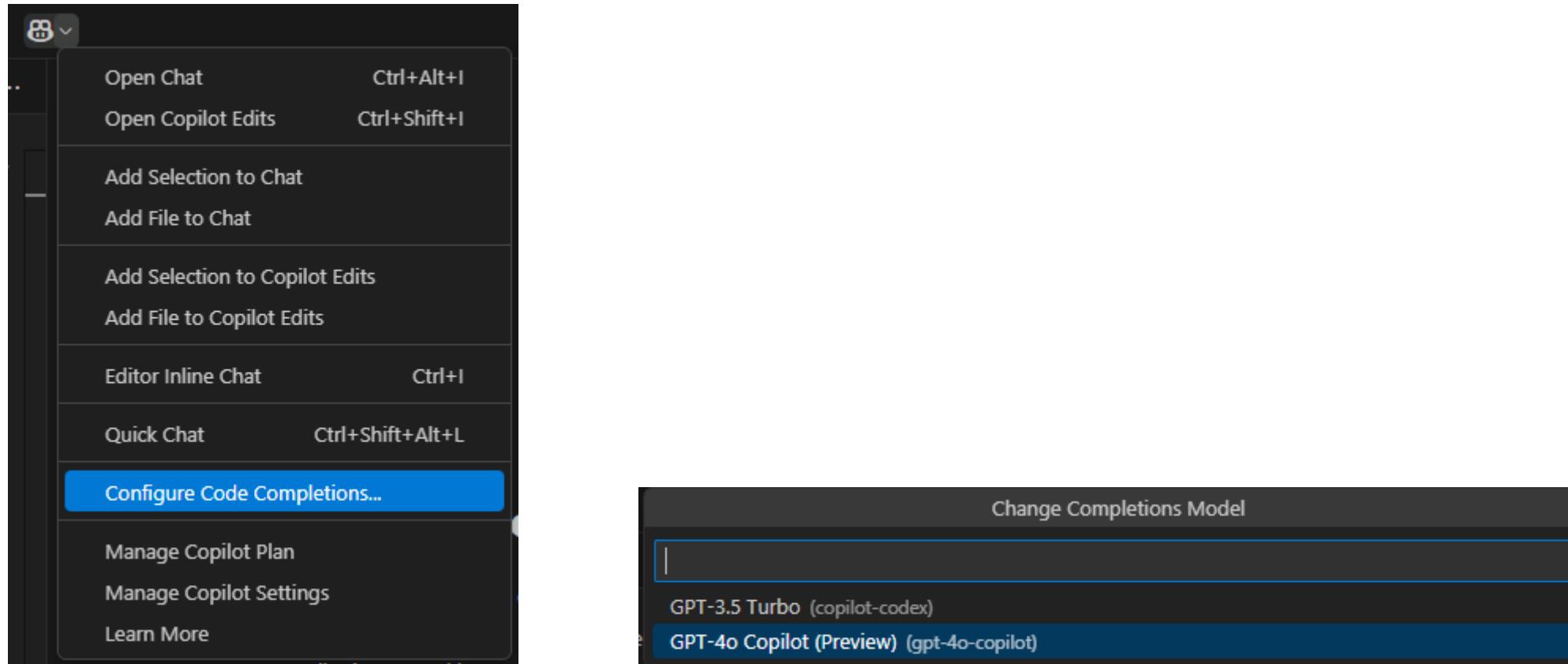
New GPT-4o Copilot code completion model available
now in public preview for Copilot in VS Code



GitHub Copilot

GPT-4o Copilot

New GPT-4o Copilot code completion model available now in public preview for Copilot in VS Code



New GPT-4o Copilot code completion model now available for Copilot in JetBrains IDEs



GitHub Copilot

GPT-4o Copilot

Announcing the General Availability of GitHub Copilot Extensions

- **General Availability:** GitHub Copilot Extensions are now accessible to all users across all Copilot license tiers.
- **Enhanced Workflow:** These extensions help developers maintain their workflow by providing context-aware assistance from their favorite tools.
- **Building Custom Extensions:** Developers have the flexibility to build their own public or private extensions based on their requirements. The [Copilot Extensions toolkit](#) provides centralized code samples and tools to help build high-quality extensions.
- **OpenID Connect (OIDC) Support:** GitHub is introducing OIDC support for builders, replacing the X-Github-Token authentication model with native third-party tokens.

Enhancements GitHub Copilot in JetBrains IDEs

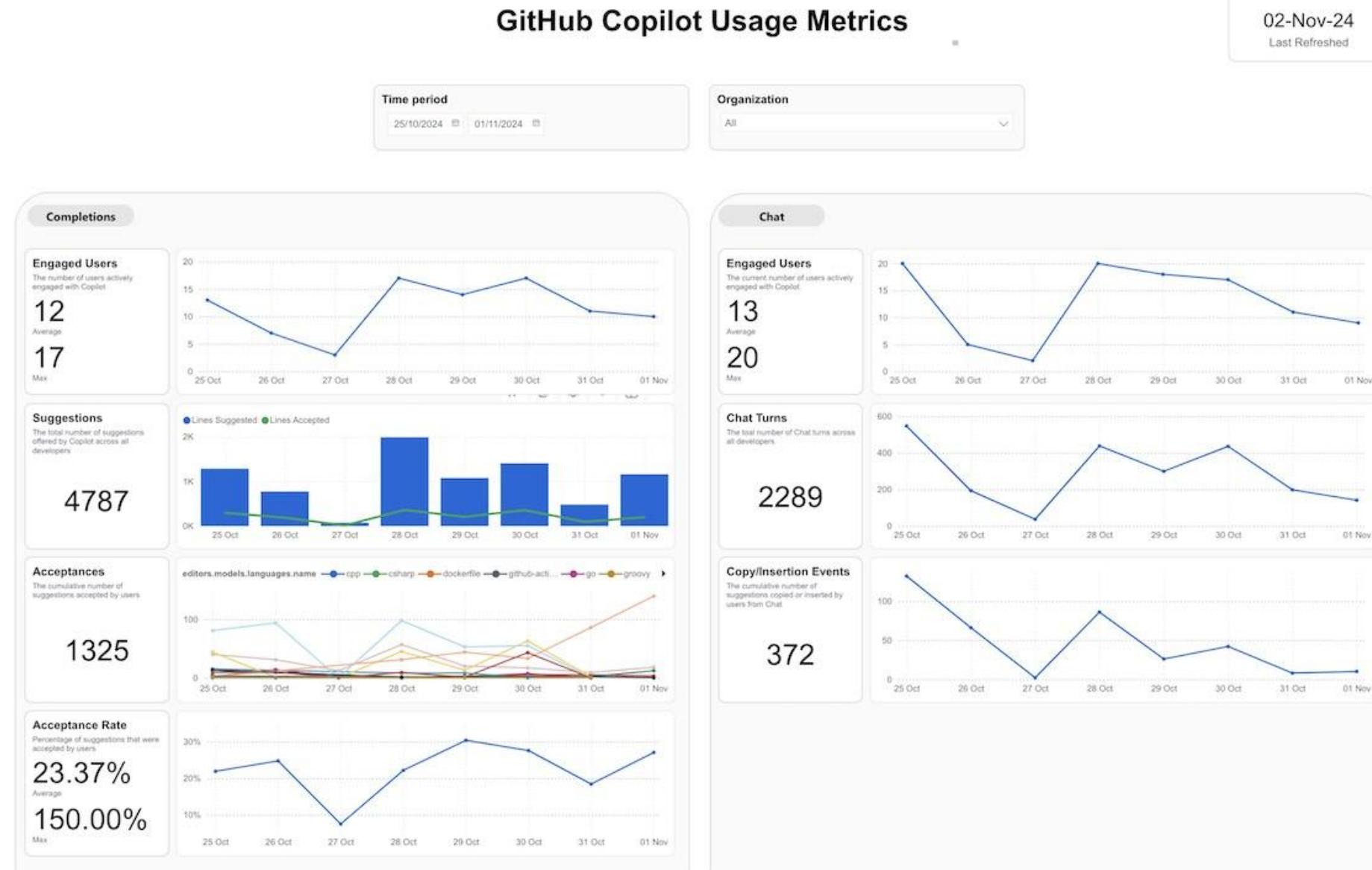
- @project Context: Ask questions about your entire codebase. GitHub Copilot reads your project's codebase and provides detailed answers with references to relevant files and symbols.
- AI-Generated Commit Messages: GitHub Copilot generates commit message suggestions, helping you craft informative commit messages and saving time.
- UX Improvements and Bug Fixes:
 - Chat conversations are now ordered by last modified time.
 - Fixed bug where chat conversation history was lost after signing out.
 - Resolved color display issues when switching system themes between dark and light.

GitHub Copilot Metrics – Power BI

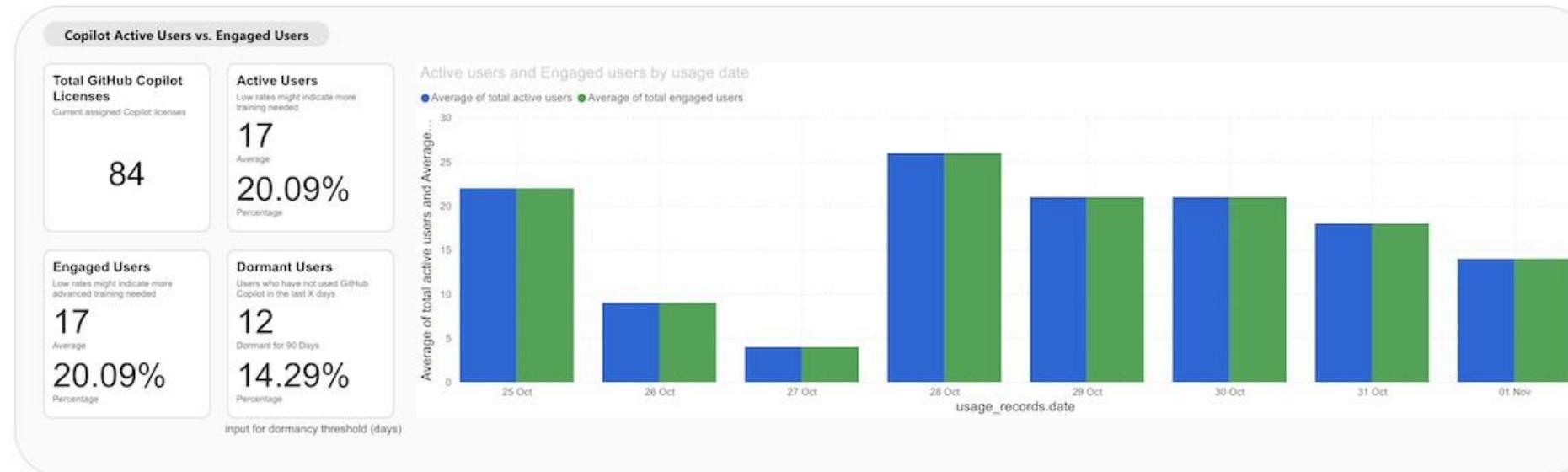
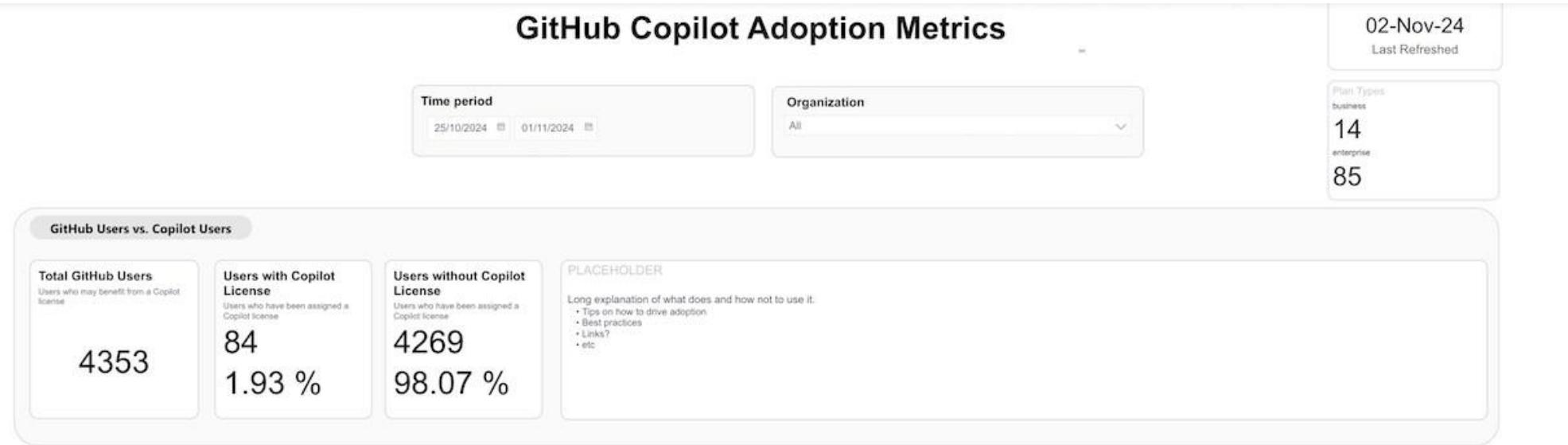
Key Features

- **Engagement Tracking:** Visualize the number of engaged users over time with intuitive line graphs, allowing you to identify trends and patterns in usage.
- **Suggestions and Acceptances:** Analyze the total number of suggestions provided by GitHub Copilot and the corresponding acceptance rates, helping you gauge the effectiveness of the tool (usually between 15-40%).
- **Chat Interactions:** Dive into user interactions within Copilot Chat environments, tracking engaged users and chat turns to understand how your teams use natural language-powered coding.

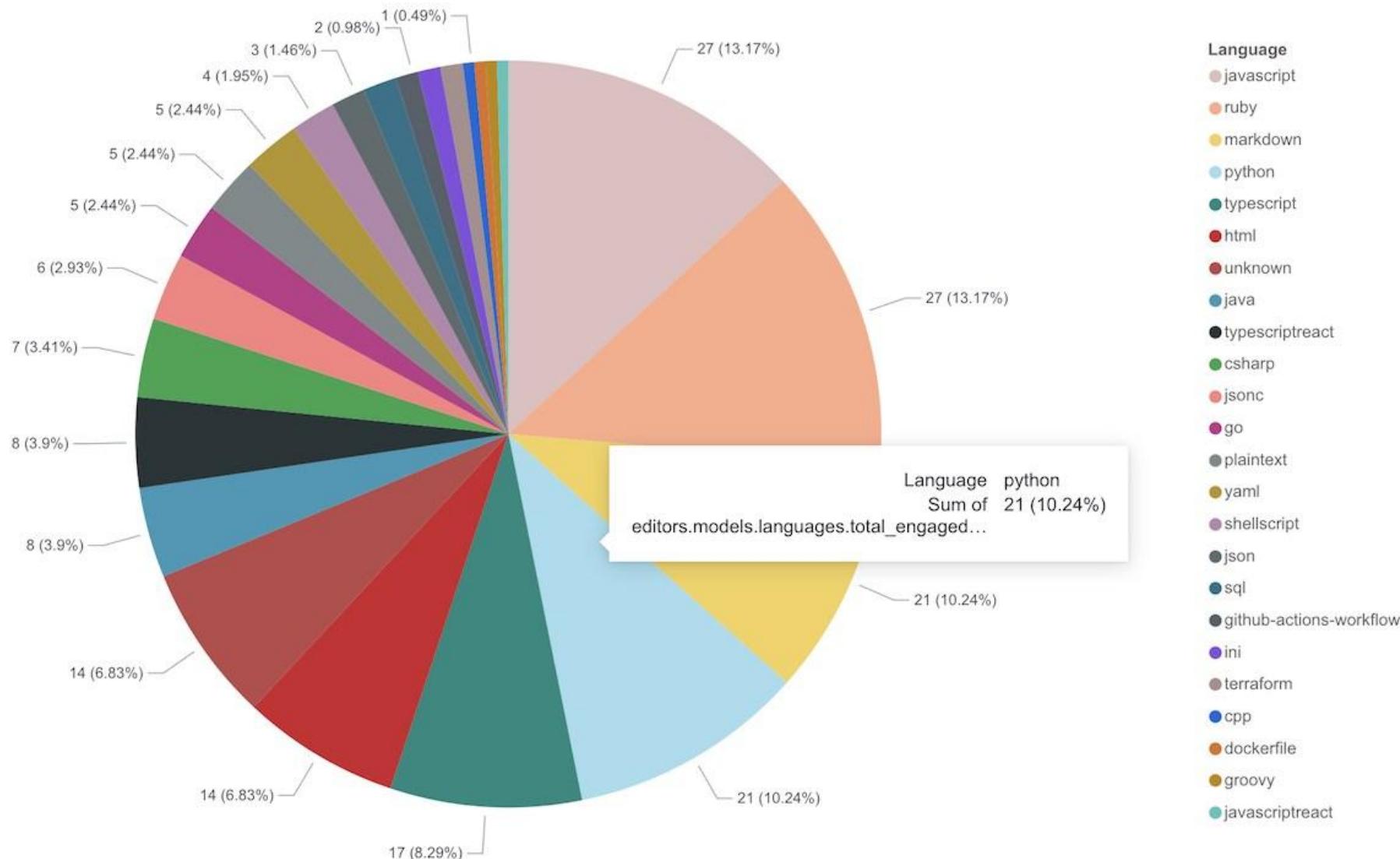
GitHub Copilot Metrics – Power BI



GitHub Copilot Metrics – Power BI



GitHub Copilot Metrics – Power BI



Deprecation of beta GitHub Copilot usage API endpoint

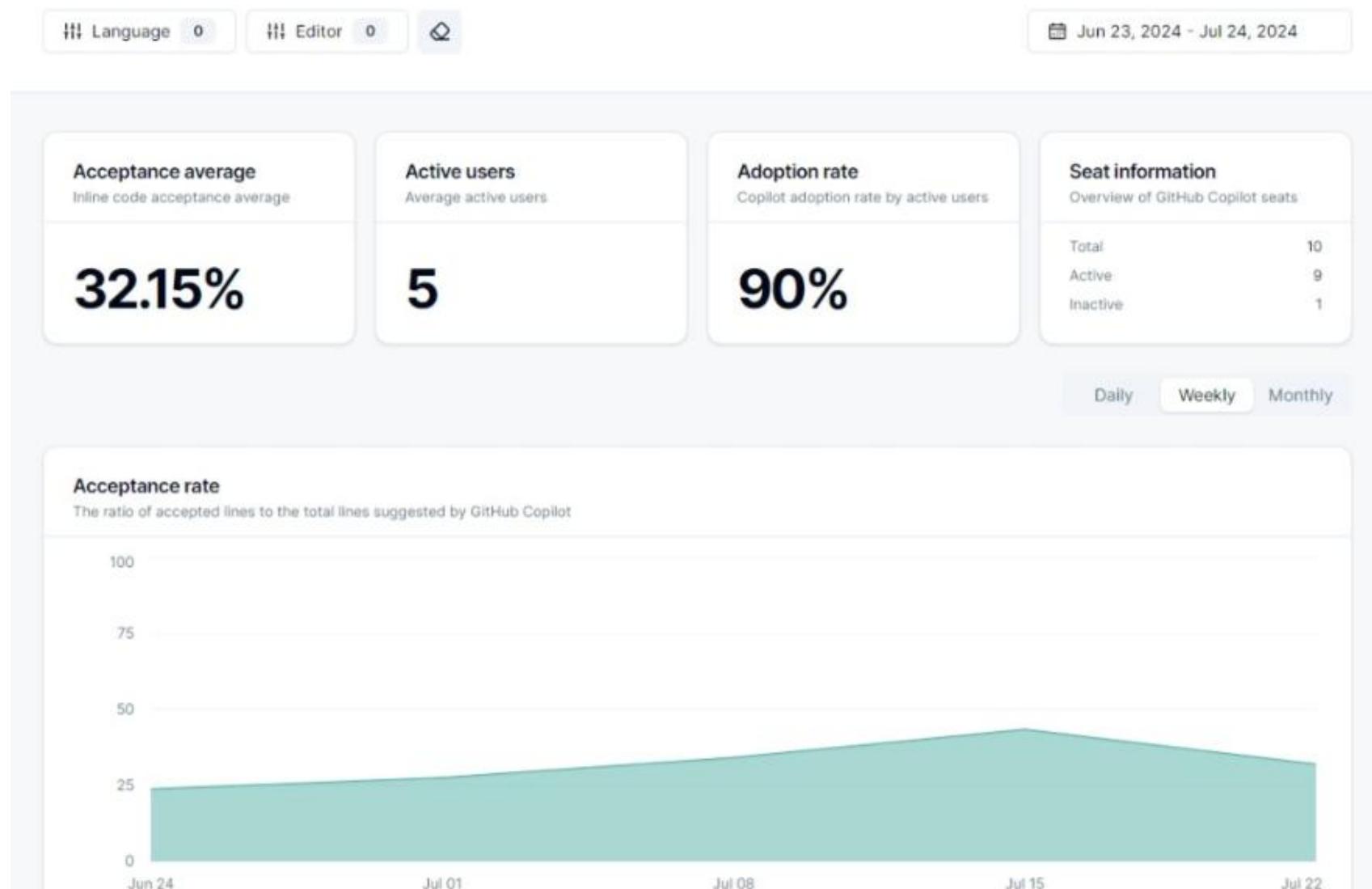
Beta Copilot `/usage` endpoints deprecated from 1st march 2025

Migrate to `/metrics` endpoints:

- Visibility into the adoption and consumption of Copilot across various stages of dev lifecycle (from code suggestions to PR reviews), from the team to the enterprise level
- Expanded scope of metrics, with the addition of GitHub.com Copilot Chat and Copilot for Pull Requests
- Consistent terminology with the user management API
- Better visibility into unique users at various drilldowns

GitHub Copilot Metrics - Dashboard

Dashboard



Claude 3.7 Sonnet is now available in GitHub Copilot in public preview

- Claude 3.7 Sonnet is now available to all customers on paid Copilot plans. This new Sonnet model supports both thinking and non-thinking modes in Copilot

Refining messages and reloading responses in Copilot Chat is now in public preview

- **Edit Previously Sent Messages:** Refine your messages whenever needed.
- **Reload Responses:** Instantly get different variations of responses.
- **Track Iterations:** Easily view the history of your message edits and responses.
- **Compare Versions:** Seamlessly switch between different message and response pairs to see which one works best.

OpenAI o3-mini and Google Gemini 2.0 Flash now available for Copilot in JetBrains IDEs

- **New Model Availability:** With the latest stable release of the GitHub Copilot extension (v1.5.35 or above), OpenAI O3 Mini (Preview) and Google Gemini 2.0 Flash (Preview) are now available to all JetBrains users of GitHub Copilot Chat.
- **Model Picker Feature:** Users can now choose between different model versions for their chat conversations by clicking the Copilot Chat icon in the JetBrains IDE

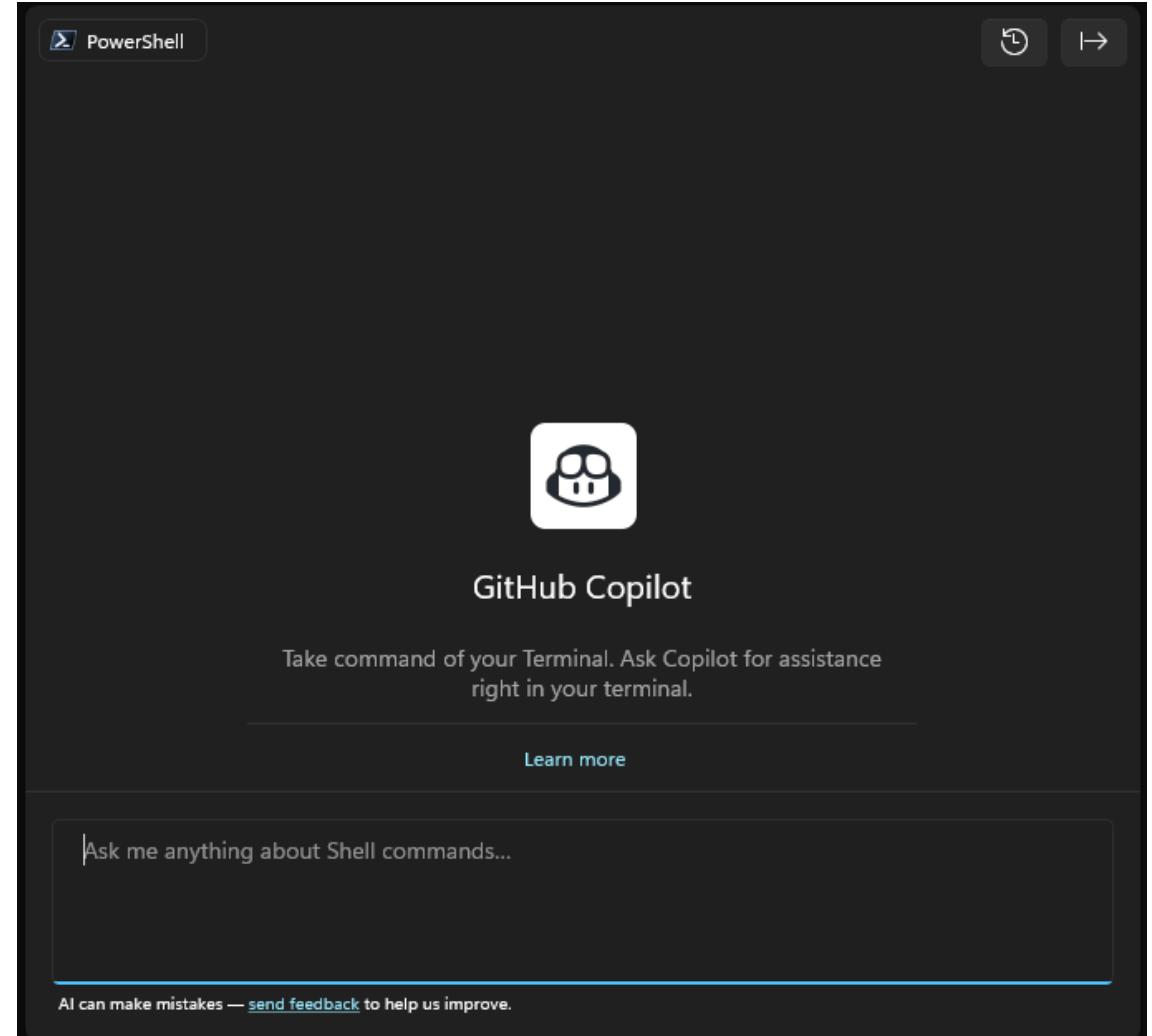
Code review in GitHub Copilot is now in public preview

- **Public Preview:** GitHub Copilot's code review feature is now available for public preview.
- **Automated Reviews:** It uses AI to identify bugs, performance issues, and suggest fixes.
- **Review Summary:** Provides detailed summaries of pull request changes.
- **Smarter Feedback:** Offers insightful feedback using the latest models.
- **Integration:** Supports automatic and on-demand reviews directly from pull requests.

GitHub Copilot now available for free in Windows Terminal Canary

GitHub Copilot Free in Windows Terminal Canary:

- Now available for free in Windows Terminal Canary.
- Users can access up to 50 chat messages per month with GitHub Copilot Free.
- Provides command suggestions and explanations directly within the terminal using Terminal Chat.
- Users can upgrade on the web if they reach their quota.
- Enhances developer experience by offering AI-powered assistance without leaving the terminal.



OpenAI GPT-4.5 in GitHub Copilot now available in
public preview



GitHub Copilot
GPT-4.5

OpenAI GPT-4.5 in GitHub Copilot now available in public preview

Public Preview Availability:

- OpenAI GPT-4.5 is now available in GitHub Copilot Chat for Copilot Enterprise users.

Advanced Capabilities:

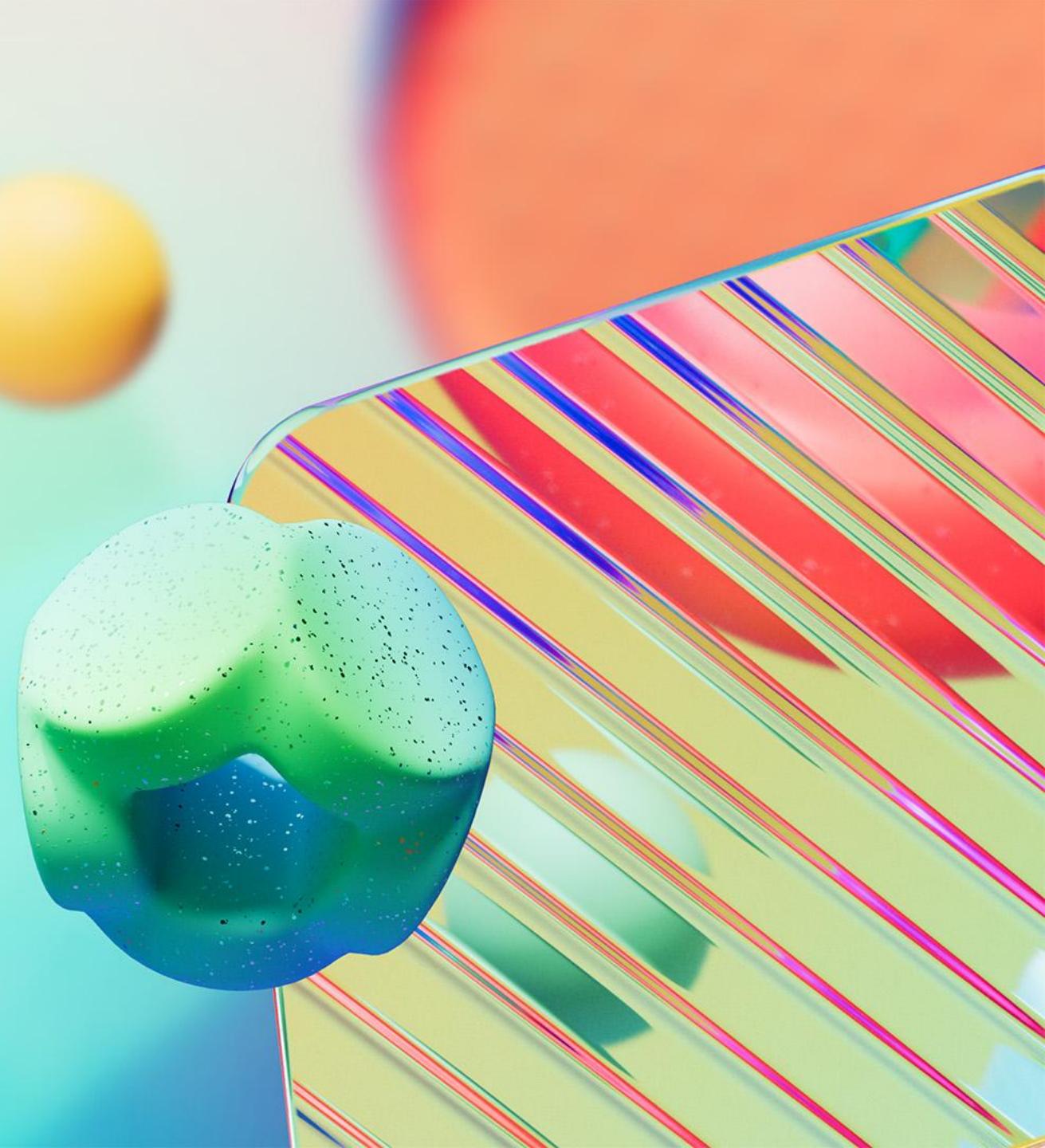
- Enhanced intuition, writing style, and broad knowledge.
- Effective with creative prompts and obscure knowledge queries.

Integration:

- Available in Visual Studio Code and on GitHub.com.
- Users can access up to 10 requests every 12 hours.

GitHub Copilot Updates

March 2025



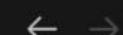
JetBrains Copilot code referencing support is generally available

Code referencing helps you decide whether to use suggested code.

Viewing code references:

Notifications for code matches with public code.

Click “View matches” to access reference information.



Welcome

binary-search.py



Public code matches



Users > 2percentsilk > binary-search.py

```
1 // implement a binary search algorithm
2
3 def binary_search(array, target):
4     # Write your code here.
5     return binary_search_helper(array, target, 0, le
6
7 def binary_search_helper(array, target, left, right)
8     if left > right:
9         return -1
10    middle = (left + right) // 2
11    potential_match = array[middle]
12    if target == potential_match:
13        return middle
14    elif target < potential_match:
15        return binary_search_helper(array, target, l
16    else:
17        return binary_search_helper(array, target, m
```

Public code matches for /Users/2percentsilk/binary-search.py

Matches will be tracked with their approximate line numbers below. As changes are made to the file, these references may become inaccurate. Matches will not persist when a file is closed.

Match

originally detected at line 8

Matched content:

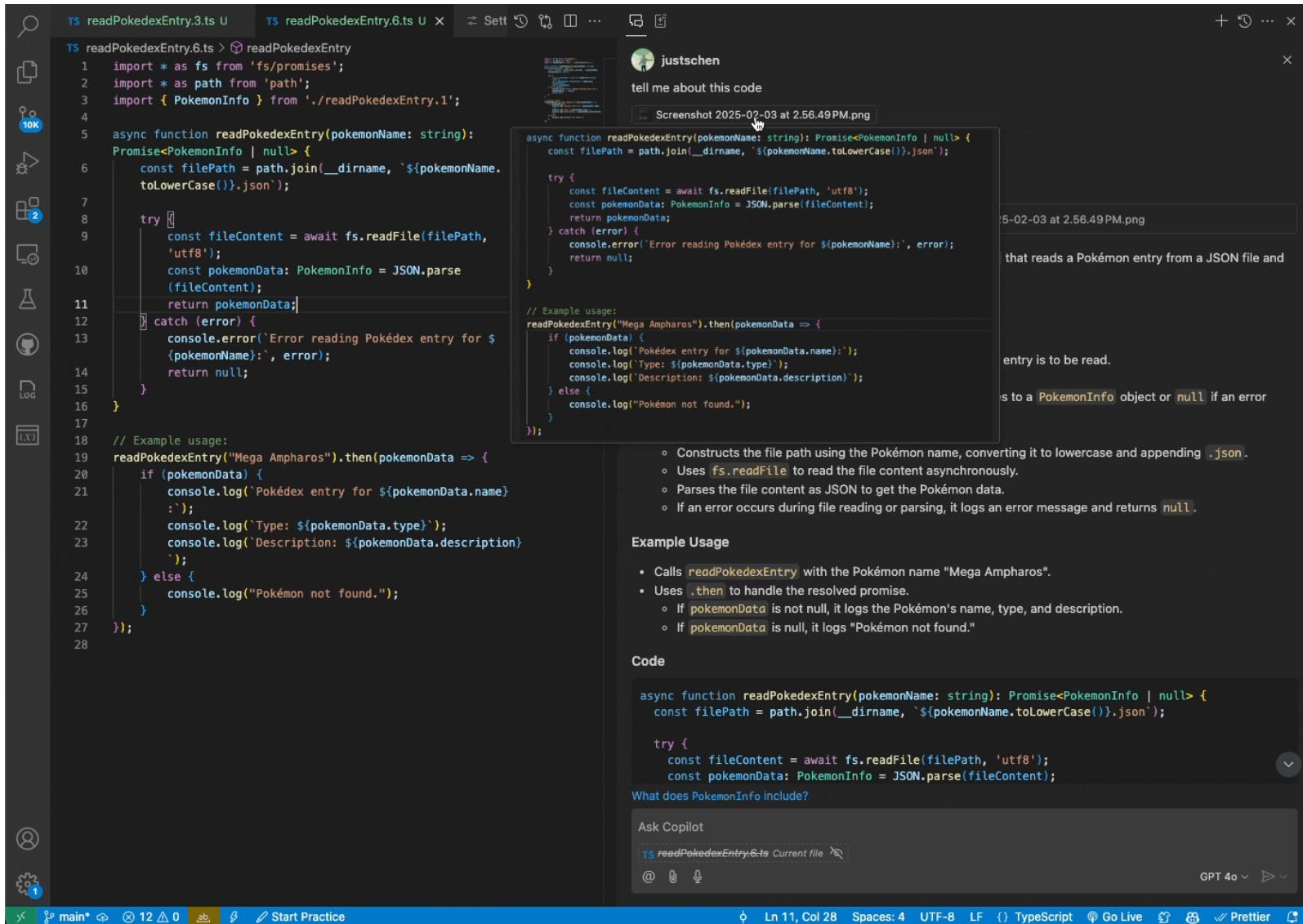
```
:
    return -1
middle = (left + right) // 2
potential_match = array[middle]
if target == potential_match:
    return middle
elif target < potential_match:
    return binary_search_helper(array, target, left, middle)
else:
    return binary_search_helper(array, target,
```

License Summary

This snippet matches 4 references to public code. Below, you can find links to a sample of 50 of these references.

- MIT (2)

Copilot Chat users can now use the Vision input in VS Code and Visual Studio in public preview



VS Code: Experimental agent mode improvements

Availability

- Available in VS Code Insiders
- Gradual rollout in VS Code Stable

UX Improvements

- Inline display of terminal commands
- Editable terminal commands in chat
- Confirm commands with Ctrl + Enter

VS Code: Experimental agent mode improvements

Functionality Enhancements

- Undo/redo last file edit in chat
- Automatic or instructed build task execution
- Disable task running via VS Code settings

Access Requirements

- Admin opt-in for Copilot Business/Enterprise users

Notebook support in Copilot Edits in preview

Intuitive Editing Experience:

- Edit notebook files with the same ease as code files.
- Create new notebooks from scratch.
- Modify content across multiple cells.
- Insert and delete cells.
- Change cell types.

Custom instructions generally available

Tailored Responses and Suggestions:

- Customize Copilot to fit your team's workflow.
- Use a *.github/copilot-instructions.md* file to describe specific requirements.

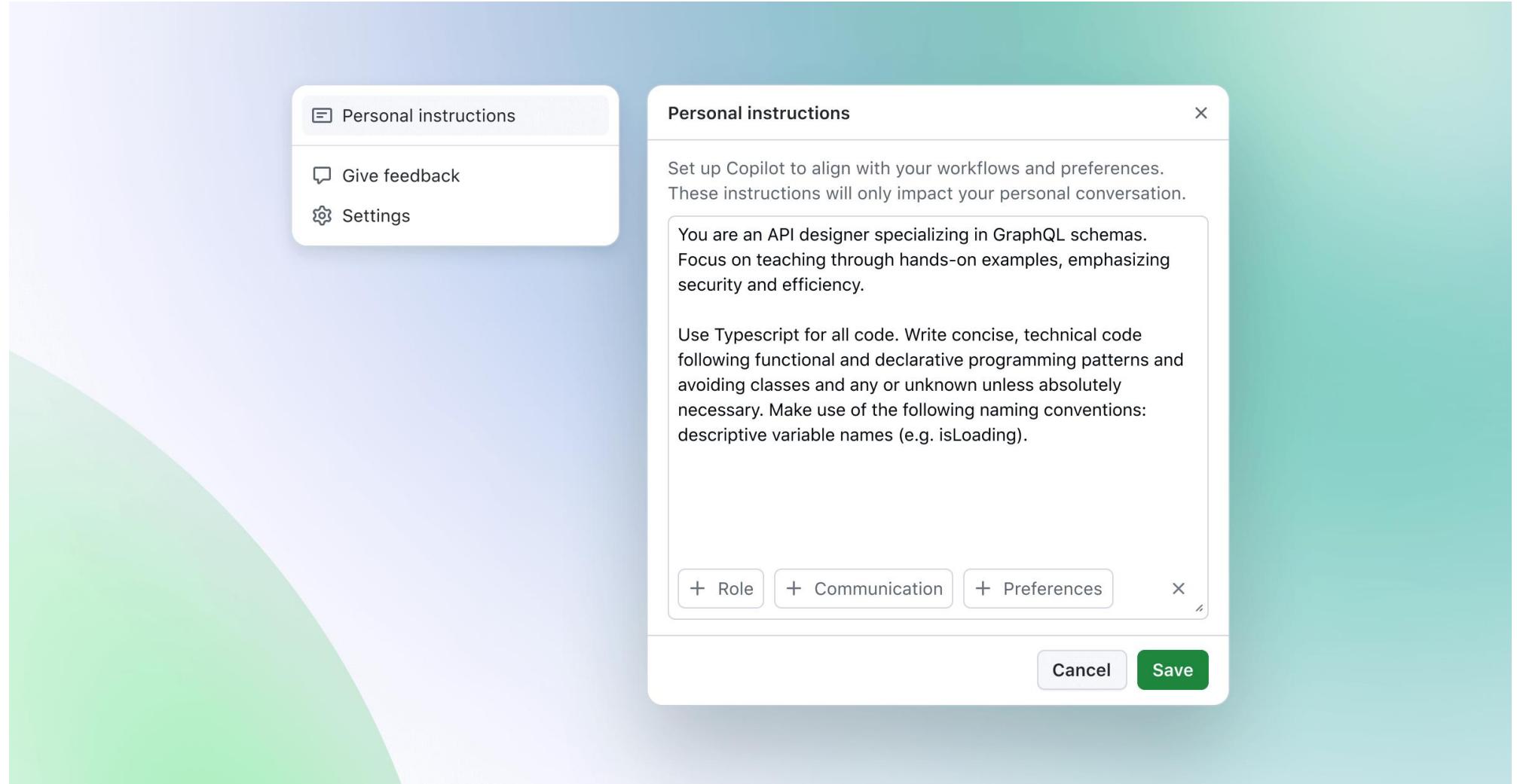
Enabling Custom Instructions:

- Ensure the VS Code setting *github.copilot.chat.codeGeneration.useInstructionFiles* is enabled.
- Copilot will use these instructions when generating responses.

Custom instructions for pull request title and description

```
{  
  "github.copilot.chat.pullRequestDescriptionGeneration.instructions": [  
    {  
      "text": "Prefix every PR title with an emoji."  
    }  
  ]  
}
```

Personal custom instructions for Copilot are now generally available on github.com



GPT-4o Copilot March flight is ready for takeoff

What's Changing:

- The GPT-4o Copilot model has been upgraded through reinforcement learning for improved code completions.

Availability:

- Available today in public preview under the same gpt-4o-copilot name.
- General availability expected in mid-April 2025.

Onboarding additional model providers with GitHub Copilot for Claude Sonnet models in public preview

New Providers:

- Google Cloud Platform
- Anthropic PBC

Models:

- **Claude 3.5 Sonnet**
- **Claude 3.7 Sonnet**

GitHub Copilot for Xcode Chat is now generally available

Key Features:

- **Chat View:** Directly ask Copilot for help with coding tasks.
- **Model Selector:** Change the model during a chat.
- **Slash Commands:** Use quick commands like /releaseNotes for updates.
- **Reference Code:** Scope chats to specific files for relevant assistance.
- **Multiple Conversations:** Maintain different threads with their own context.
- **Extended Version Support:** Supports macOS versions from 12.0 to 15.0.



GitHub Copilot Chat for Eclipse now in public preview



What's New:

- **Chat View:** Ask Copilot for help with coding tasks directly in the chat view.
- **Model Selector for Chat:** Change the model during a chat.
- **Slash Commands:** Use quick commands like /explain for code explanations.
- **Reference Code:** Scope chats to specific files for more relevant assistance.

Code completion in GitHub Copilot for Eclipse is now generally available



Key Features:

- **Code Completions:** Real-time, context-aware suggestions.
- **Multi-Language Support:** Supports Java, Python, C++, and more.
- **Content Filtering:** Screens out harmful or inappropriate content.

Instant semantic code search indexing now generally available for GitHub Copilot

What's Changed:

- Indexing time reduced from approximately five minutes to just a few seconds (up to 60 seconds).
- Immediate contextually-aware Copilot assistance after opening a repository.

How It Works:

- Automatically triggered when opening GitHub Copilot Chat on github.com.
- Manually trigger indexing through the Copilot UI in VS Code if needed.

Instant semantic code search indexing now generally available for GitHub Copilot

Why This Matters:

- Provides responses specific to your codebase's architecture and patterns.
- References existing functions, classes, and implementations in your repo.
- Suggests code that aligns with your project's style and conventions.
- Answers questions about your codebase with accurate, context-aware information.
- Enhances development workflow efficiency and reduces interruptions.

Instant previews, flexible editing, and working with issues in Copilot available in public preview

What's New:

- **HTML Preview:** Instantly preview rendered HTML files in the side panel.
- **File Editing:** Edit files directly in the side panel.
- **Mermaid Diagrams:** Quickly generate and preview flowcharts or sequence diagrams.
- **Issue Tracking:** Monitor issues in the right side panel.
- **GitHub Style Responses:** Track issues or pull requests in familiar GitHub style.
- **Improved Streaming:** Smoother streaming and enhanced attachment rendering.

Enhance your productivity with Copilot Edits in JetBrains IDEs

What's New:

- **Copilot Edits:** Make changes across multiple files directly from Copilot Chat.
- **How to Use:** Click the Copilot Chat icon in JetBrains IDE to start an Edit session.

Benefits for Developers:

- **Enhanced Clarity:** Overview of modifications with summaries of affected files.
- **Preview Changes:** View code diffs and decide to accept or discard changes.
- **Boosted Productivity:** Save time and focus on complex tasks.

GPT-4o Copilot: Your new code completion model is now generally available



Copilot Chat on GitHub Mobile now supports multi-model selection

- **Multi-Model Selection:** GitHub Copilot Chat on mobile now supports multiple models including OpenAI o3-mini, Anthropic Claude 3.5 Sonnet, and Google Gemini 2.0 for free users. Copilot Pro subscribers have access to the OpenAI o1 model.
- **Model Picker:** Users can choose their favorite models using the model picker in Copilot Chat, allowing for a customized experience.

https://github.com/copilot

GPT-4o

Models

- ✓ GPT-4o Azure OpenAI
- o3-mini Azure OpenAI
- AI Claude 3.5 Sonnet Anthropic
- AI Claude 3.7 Sonnet Anthropic
- AI Claude 3.7 Sonnet Thinking Anthropic
- Gemini 2.0 Flash Google
- GPT-4.1 Azure OpenAI

Preview

- o1 Azure OpenAI
- Gemini 2.5 Pro Google
- GPT-4.5 Azure OpenAI
- o3 Azure OpenAI
- o4-mini Azure OpenAI

Ask Copilot

2025-04-02

Copilot Chat on github.com adds support for images

New Feature: Copilot Chat on GitHub now supports image upload and analysis.

Capabilities:

- **Image Upload & Analysis:** Users can upload, paste, or drag images directly into Copilot conversations.
- **Visual Context Understanding:** Ask questions about code screenshots, UI designs, system diagrams, and more.
- **Image to Code:** Generate HTML files based on images and preview them in the side panel.
- **Impact:** Enhances communication and collaboration by integrating visual artifacts into the development workflow.



↔ How to load remote JS script

⌚ Recent commits in torvalds/linux

⬇ How to download GitHub folders

⌚ Compare dates and times

⬇ Pull requests in microsoft/vscode

↔ Python Panda data analysis

Copilot uses AI. Check for mistakes.

getRepository...

getFile, folder, and symbol...

getKnowledgeBase...

getImage...

@ Extension >

Ask Copilot



GitHub Copilot in VS Code March release (v1.99)

Key Updates:

- **Agent Mode**: Now available in VS Code stable.
- **MCP Support**: MCP Server.
- **Built-in Tools**: Fetch web content, find symbol references (#Fetch).
- **UX Improvements**: Optimized Next Edit Suggestions (NES).
- **Model Choice**: Anthropic Claude, Google Gemini, OpenAI models available.
- **Bring Your Own Key (BYOK)**: Enables use of models through API keys

GitHub Copilot in VS Code March release (v1.99) - UX Improvements

Switch Modes: Seamlessly switch between ask, edit, and agent mode.

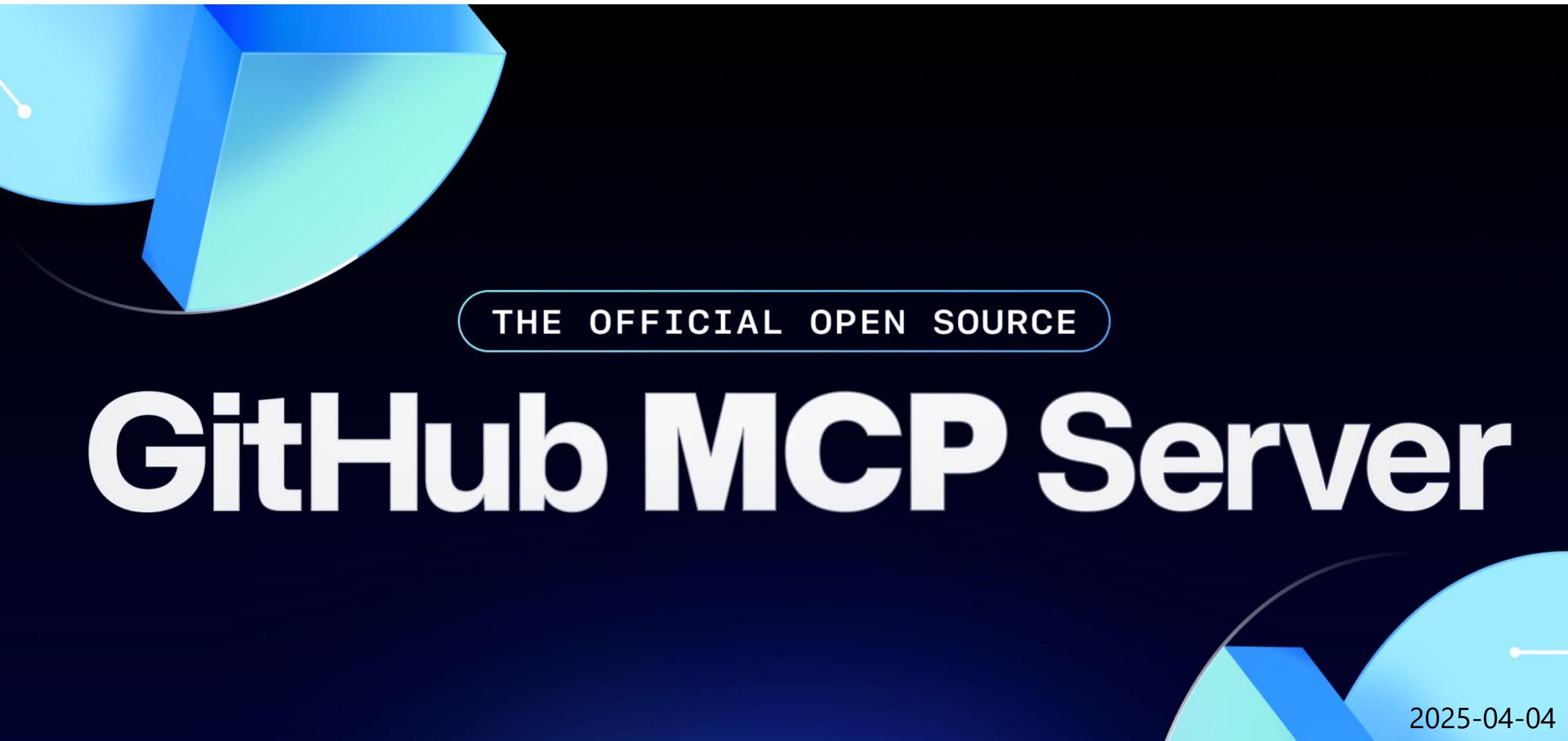
Next Edit Suggestions (NES): Optimized to hint at the next edit without being distracting.

GitHub Copilot in VS Code March release (v1.99) - Limitless Model Choice

Model Selector: Anthropic Claude, Google Gemini, OpenAI models available.

API Key Usage: Use models from providers like Anthropic, Azure, Google Cloud, OpenAI.

github-mcp-server is now available in public preview



THE OFFICIAL OPEN SOURCE

GitHub MCP Server

Multiple new models are now generally available in GitHub Copilot

GA

- Anthropic Claude 3.7 Sonnet
- Claude 3.5 Sonnet
- OpenAI o3-mini
- Google Gemini Flash 2.0

Announcing GitHub Copilot Pro+

- New individual plan for advanced developers
- Includes all Copilot Pro features
- Exclusive access to GPT-4.5 (available now)
- Priority access to feature previews
- 1500 premium requests/month (starting June 4)
- Unlimited requests for agent mode and code completions

Copilot code review now generally available

GENERAL AVAILABILITY

Copilot code review

The screenshot shows a GitHub Pull Request interface with the following details:

- Reviewers:** Copilot (AI) reviewed just now.
- Comments:** Copilot (AI) commented 4 hours ago.
- Code Review:** Copilot (AI) 2 minutes ago. The comment suggests validating the age input to ensure it is a valid number.
- Suggested Change:** A tooltip shows a suggested change to parse the age input.
- Request up to 100 reviewers:** A modal is open for requesting additional reviewers.
- Reviewers List:** A dropdown shows suggestions for reviewers: fernflare, rodbotas6, chandriverse, and semyonrush.
- Pull Request Overview:** This PR introduces a new interactive sidebar that changes the panel's width based on user settings.
- Reviewed Changes:** Copilot reviewed 3 out of 3 changed files.
- Files:** A list of files including ui/packages/src/Sidepanel.tsx.

Code snippets from the review:

```
src/cats/form.tsx
14 +  const form = event.currentTarget;
15 +  const name = form.elements.namedItem("name") as HTMLInputElement
16 +  const age = form.elements.namedItem("age") as HTMLInputElement
17 +
```

```
17 -  
17 +  const parsedAge = parseInt (age.value, 10);  
18 +  if (isNaN(parsedAge) ) {  
19 +      alert("Please enter a valid number for age.");  
20 +      return;  
21 +  }
```

Request up to 100 reviewers

Type or choose a user

Copilot

fernflare Fernanda Sarmento

rodbotas6 Rodrigo Botas

chandriverse Chandrika Gupta

semyonrush Semyon Maslennikov

2025-04-04

GENERAL AVAILABILITY

Copilot AI reviewed just now

Copilot code review

This PR introduces a new input field for age. The input field is currently unvalidated.

src/cats/form.tsx

```
14 +  const form = event.currentTarget;
15 +  const name = form.elements.namedItem("name") as HTMLInputElement
16 +  const age = form.elements.namedItem("age") as HTMLInputElement
17 +
```

Copilot AI 2 minutes ago

The age input should be validated to ensure it is a valid number. Consider adding a check for NaN after parsing the age.

Suggested change

```
17 - 
17 +  const parsedAge = parseInt(age.value, 10);
18 +  if (isNaN(parsedAge) ) {
19 +      alert("Please enter a valid number for age.");
20 +      return;
21 +  }
22 +  addCat ({
23 +      name: name.value,
24 +      age: parsedAge,
```

Request up to 100 reviewers

Type or choose a user

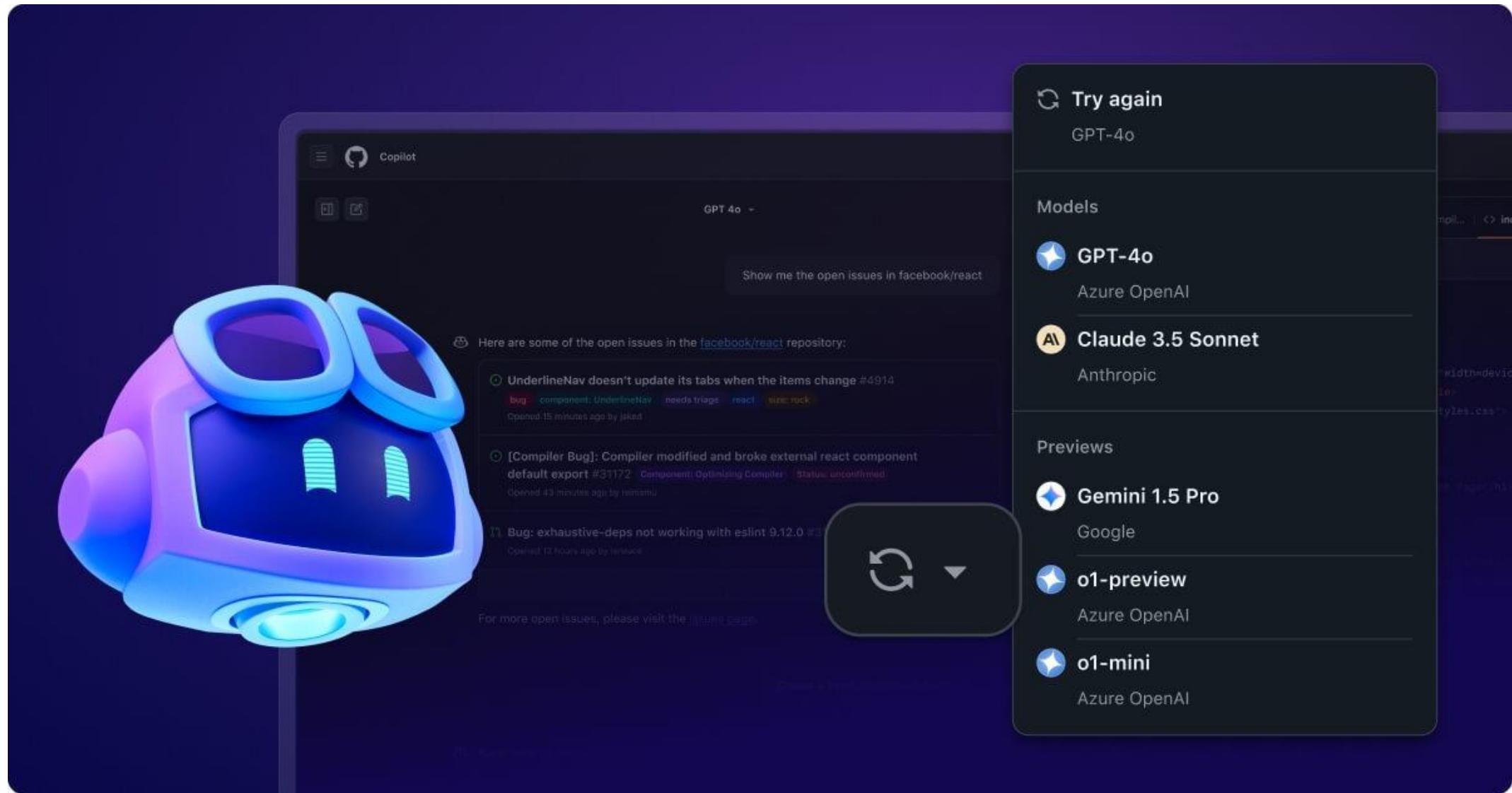
- Copilot
- fernflare Fernanda Sarmento
- rodbotash6 Rodrigo Botas
- chandriverse Chandrika Gupta
- semyonrush Semyon Maslov

2025-04-01

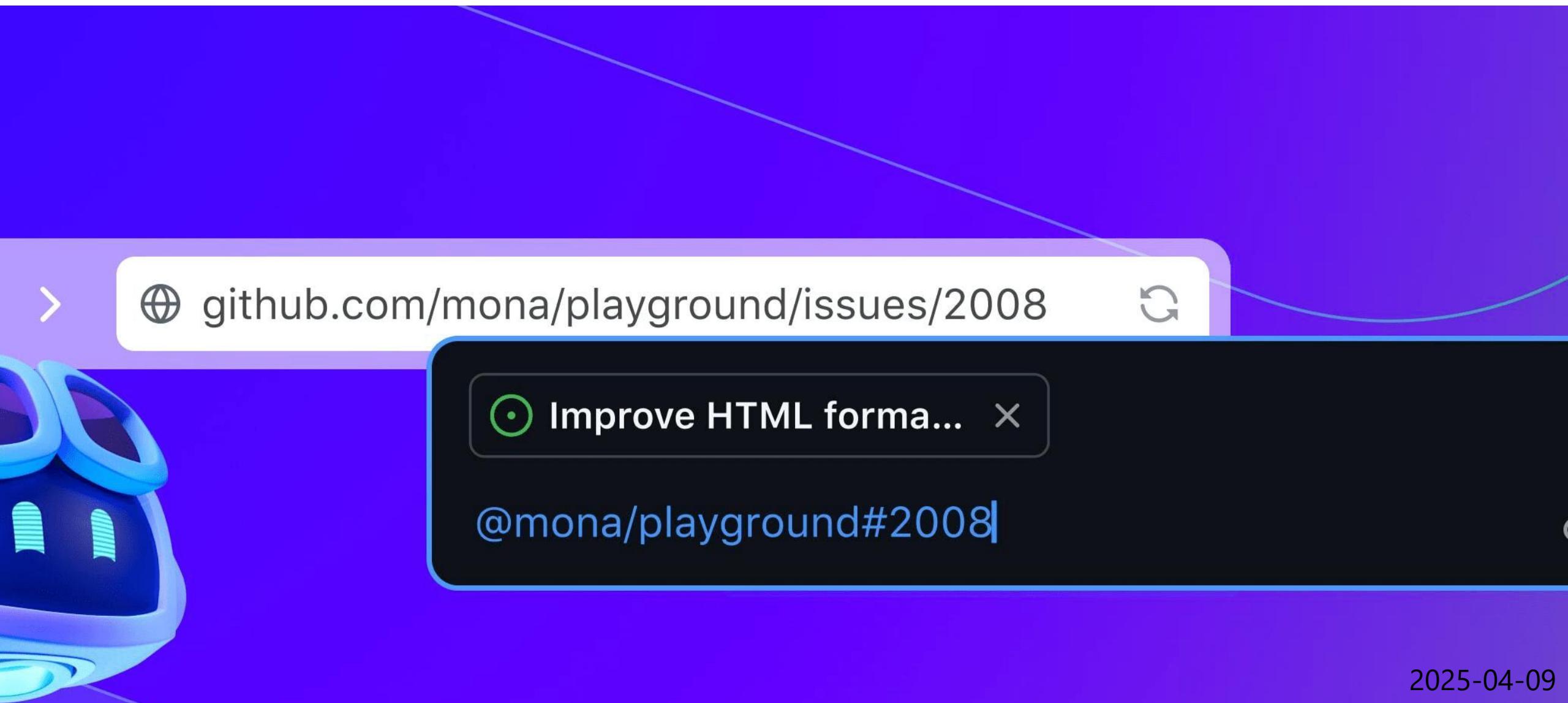
Copilot code review now generally available

- **General Availability:** Copilot Code Review is now available to all paid Copilot subscribers.
- **Purpose:** Automates basic code reviews to identify bugs, performance issues, and suggest fixes.
- **Integration:** Works in Visual Studio Code and GitHub; can be triggered automatically or on demand.
- **Adoption:** Over 1 million developers used it during public preview.
- **Language Support:** Now includes C, C++, Kotlin, and Swift (in public preview); HTML and TXT support coming soon.
- **Enterprise Access:** Organizations can enable it via GitHub policy settings.

Reload responses with other Copilot models on github.com



Copilot Chat now supports pasting GitHub URLs as explicit references



Copilot Chat users can use the Gemini 2.5 Pro model in public preview

**Gemini 2.5 Pro is
now available in
GitHub Copilot!**

Copilot Chat users can use the latest Gemini model from Google in public preview.



VSCode Copilot agent mode in Codespaces

Implement logging to application insights on backend #5

New issue



congiuluc opened on Feb 24

20

No description provided.

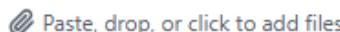


Add a comment

Write Preview

Preview

Use Markdown to format your comment



 Close issue

Comment

^① Remember, contributions to this repository should follow its [contributing guidelines](#), [security policy](#) and [Support](#).

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with Conduit Agent Model

No branches or pull requests

Notifications

refs/gh/queue/master · ria-gopu / codespaces-testing · GitHub · Feedback · Rails 8.1.0.alpha.cca8fb8 | Ruby 3.4.2 | React 18.3.1 · 321ms

Code Issues 3 Pull requests Actions Projects Security Insights Settings

Improve the "About Me" page #3

[Open](#) | [Parent: Build a Personal Website](#)

ria-gopu opened 7 minutes ago · edited by ria-gopu

Implement the following improvements to the About Me page:

- Include and center a placeholder image at the top of the page, below the header.
- Change background color to #ff0fc

[Create sub-issue](#) [...](#)

ria-gopu added a parent issue 7 minutes ago
[Build a Personal Website #1](#)

ria-gopu self-assigned this 7 minutes ago

Add a comment

Write Preview

Use Markdown to format your comment

Paste, drop, or click to add files

Close issue [Comment](#)

Assignees: ria-gopu

Labels: No labels

Projects: No projects

Milestone: No milestone

Relationships: Parent issue

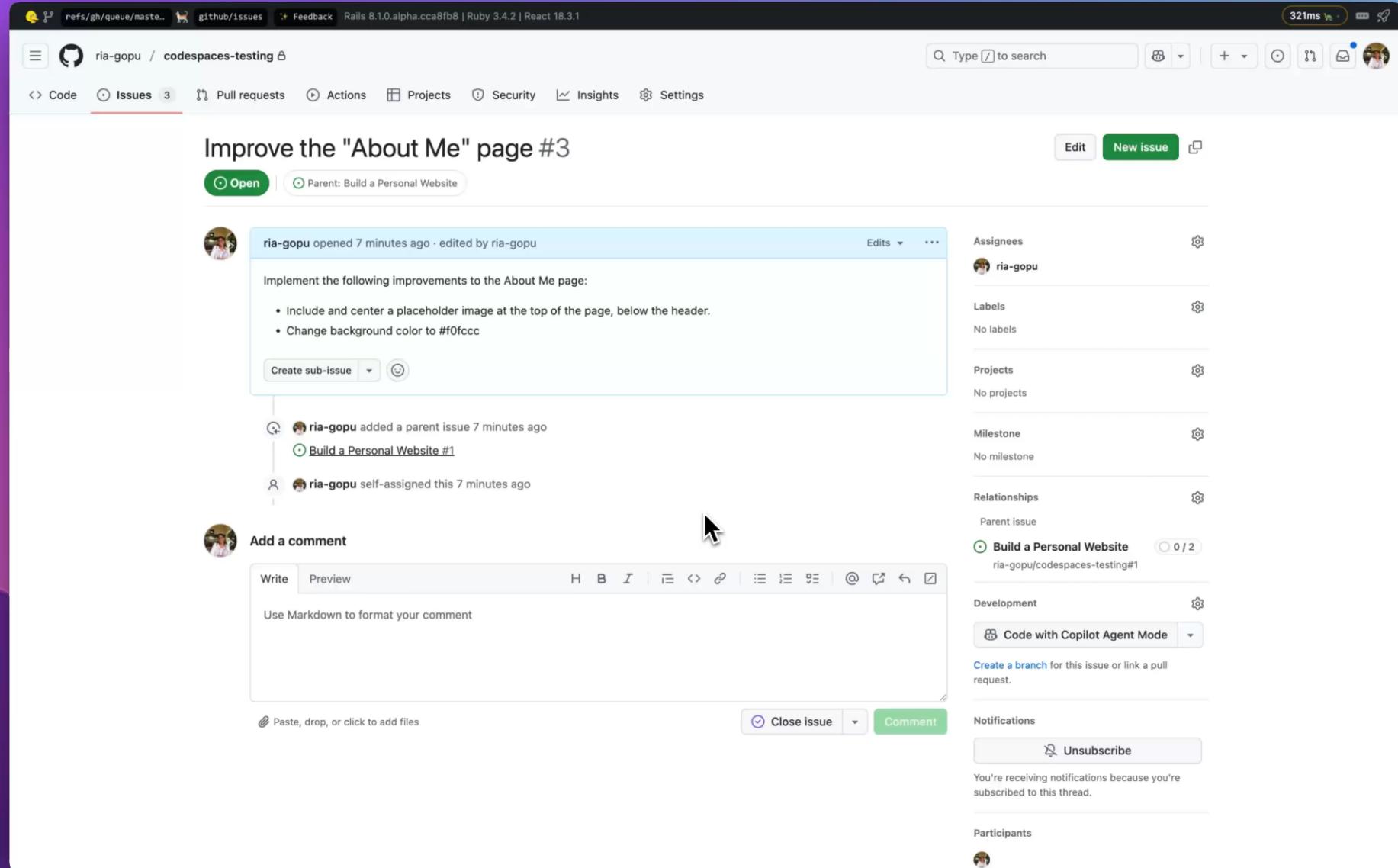
[Build a Personal Website](#) 0 / 2
ria-gopu/codespaces-testing#1

Development: Code with Copilot Agent Mode

Create a branch for this issue or link a pull request.

Notifications: [Unsubscribe](#)
You're receiving notifications because you're subscribed to this thread.

Participants: ria-gopu



OpenAI GPT-4.1 now available in public preview for GitHub
Copilot and GitHub Models

OpenAI GPT-4.1
is now available

in Copilot Chat and GitHub Models



GitHub Copilot Chat for Eclipse is now generally available

General Availability



GitHub Copilot Chat

for Eclipse [GA]

```
1 package org.springframework.samples.petclinic.owner;
2
3 import java.util.List;
4
5 @Controller
6 class OwnerController {
7
8     private static final String VIEWS_OWNER_CREATE_OR_UPDATE_FORM = "owners/createOrUpdateOwnerForm";
9
10    private final OwnerRepository owners;
11
12    public OwnerController(OwnerRepository owners) {
13        this.owners = owners;
14    }
15
16    @InitBinder
17    public void setAllowedFields(WebDataBinder dataBinder) {
18        dataBinder.setDisallowedFields("id");
19    }
20
21    @ModelAttribute("owner")
22    public Owner findOwner(@PathVariable(name = "ownerId", required = false) Integer ownerId) {
23        return ownerId == null ? new Owner()
24            : this.owners.findById(ownerId)
25                .orElseThrow(() -> new IllegalArgumentException("Owner not found with id: " + ownerId
26                    + ". Please ensure the ID is correct " + "and the owner exists in the database."))
27    }
28
29    @GetMapping("/owners/new")
30    public String initCreationForm() {
31        return VIEWS_OWNER_CREATE_OR_UPDATE_FORM;
32    }
33
34    @PostMapping("/owners/new")
35    public String processCreationForm(@Valid Owner owner, BindingResult result, RedirectAttributes redirectAt
36        if (result.hasErrors()) {
```

/explain

 GitHub Copilot

The `OwnerController` class in the `org.springframework.samples.petclinic.owner` package is a Spring MVC controller that handles HTTP requests related to `Owner` entities. It uses various Spring annotations to map web requests to specific handler methods and manage the lifecycle of the `Owner` objects.

The class is annotated with `@Controller`, indicating that it is a web controller. It has a constant `VIEW_OWNER_CREATE_OR_UPDATE_FORM` that holds the view name for the owner creation or update form.

```
private static final String VIEW_OWNER_CREATE_OR_UPDATE_FORM = "owners/createOrUpdateOwnerForm";
```

The controller has a dependency on `OwnerRepository`, which is injected via the constructor. This repository is used to perform CRUD operations on `Owner` entities.

```
public OwnerController(OwnerRepository owners) {
    this.owners = owners;
}
```

The `@InitBinder` annotated method `setAllowedFields` configures a `WebDataBinder` to disallow binding of the `id` field, preventing clients from setting the `id` property directly.

```
@InitBinder
public void setAllowedFields(WebDataBinder dataBinder) {
    dataBinder.setDisallowedFields("id");
}
```

The `@ModelAttribute` annotated method `findOwner` is used to populate the model with an `Owner` object before handling requests. It either creates a new `Owner` or retrieves an existing one based on the `ownerId` path variable.

```
@ModelAttribute("owner")
public Owner findOwner(@PathVariable(name = "ownerId", required = false) Integer ownerId) {
    return ownerId == null ? new Owner()
```

202

GitHub Copilot Chat for Eclipse is now generally available

- **General Availability:** Now available to all Eclipse users with a Copilot license.
- **Chat View:** Ask Copilot for help directly within the editor via a dedicated chat interface.
- **Model Selector:** Switch between different AI models during a chat session.
- **Slash Commands:** Use commands like /explain for quick code insights.
- **Reference Code:** Scope conversations to specific files for more relevant answers.
- **ABAP Support:** Initial support for ABAP code, using general models (no fine-tuning yet).

Using vision input in Copilot Chat with Claude and Gemini is now in public preview

The image shows a screenshot of the Copilot Chat interface. At the top, a dropdown menu is open, showing the following options:

- Claude 3.7 Sonnet (selected)
- GPT-4o
- Claude 3.5 Sonnet
- Claude 3.7 Sonnet (selected, indicated by a checkmark)
- Gemini 2.0 Flash
- Gemini 2.5 Pro

Below the dropdown is a file input field containing the file "whiteboard-photo.png". At the bottom of the interface, there is a text input field with the placeholder "Convert this architecture diagram into a markdown file." and a "Copilot Chat" button.

Convert this architecture diagram into a markdown file.



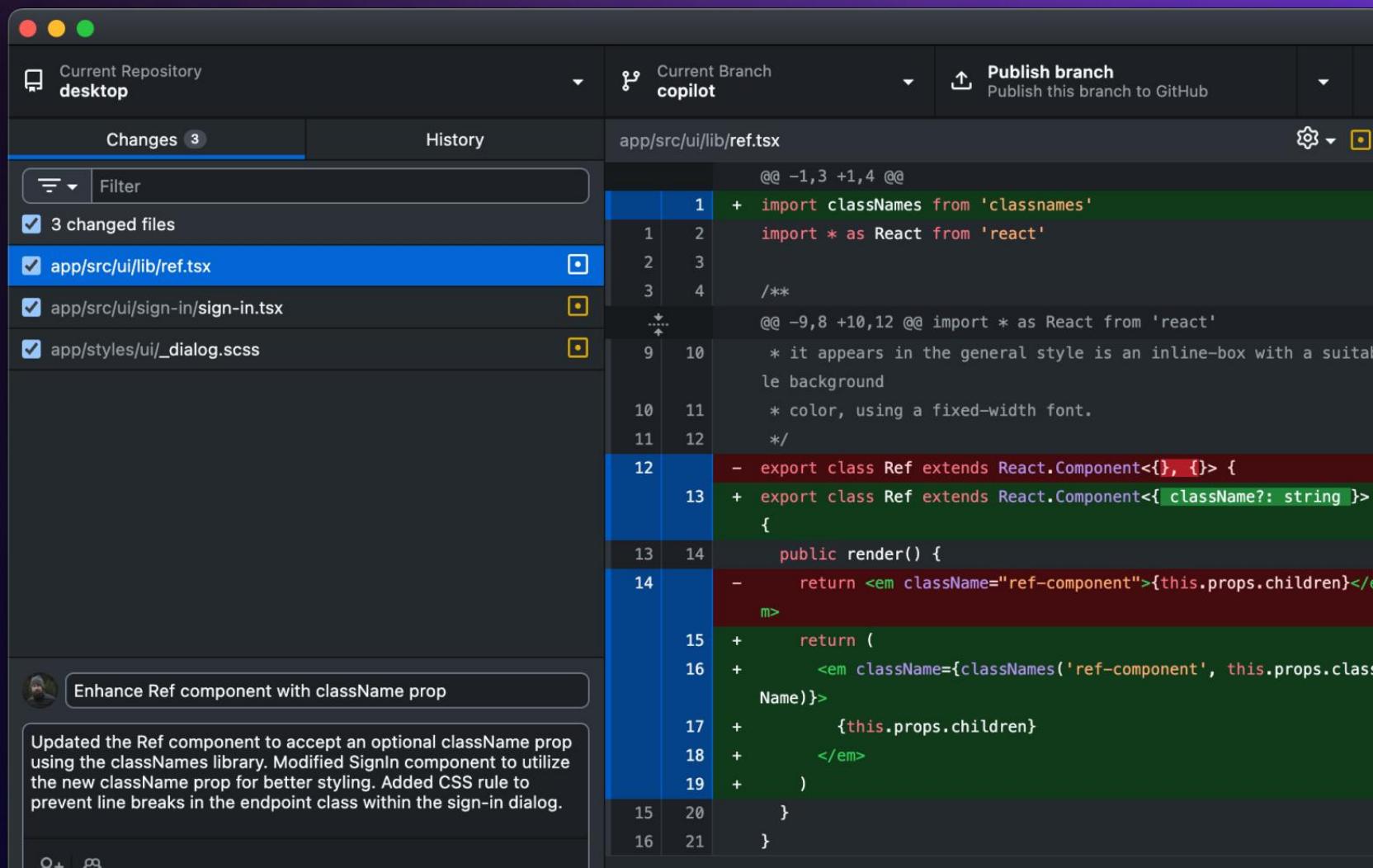
OpenAI o3 and o4-mini are now available in public preview
for GitHub Copilot and GitHub Models

OpenAI o3 and o4-mini
are now available

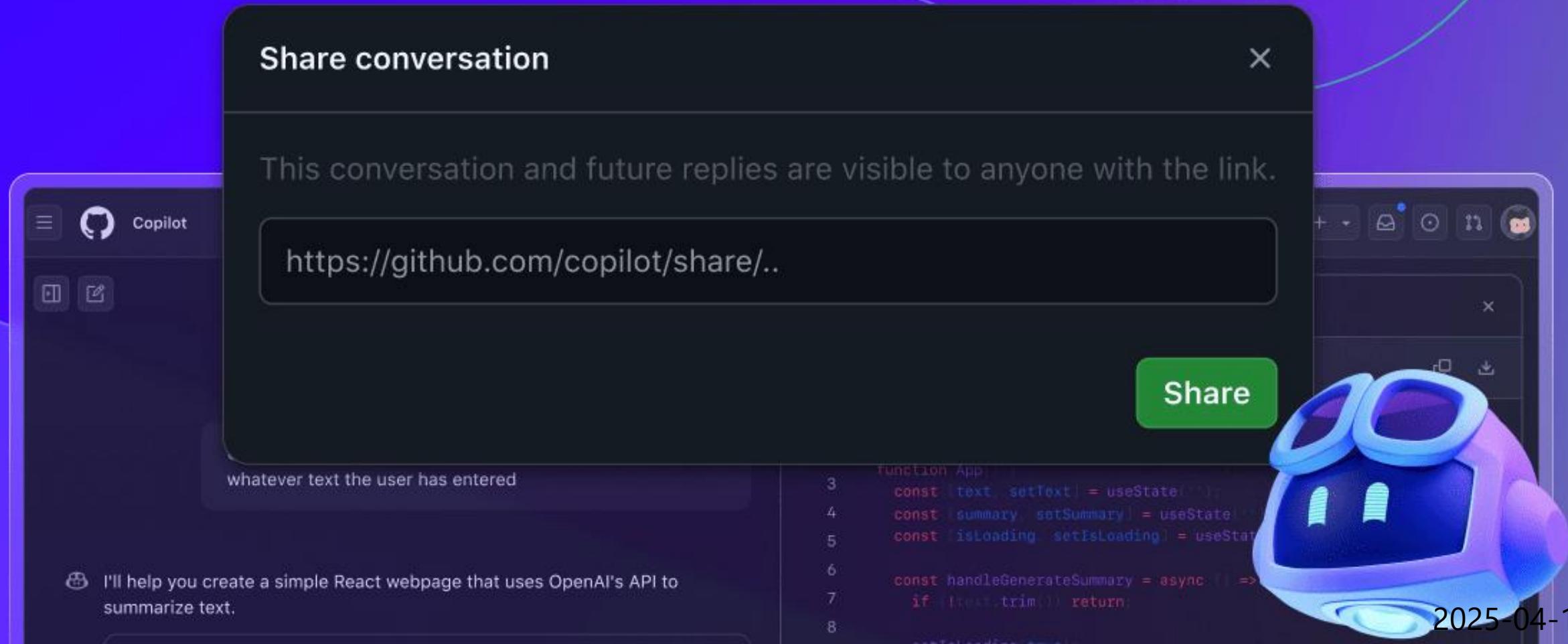
in Copilot Chat and GitHub Models



Copilot commit message generation now in public preview on GitHub Desktop



Share Copilot Chat conversations in public preview



The image shows a screenshot of the Copilot Chat interface. A central modal window titled "Share conversation" is displayed, containing a warning message: "This conversation and future replies are visible to anyone with the link." Below the message is a copyable URL: <https://github.com/copilot/share/..>. A large green "Share" button is at the bottom right of the modal. In the background, the main Copilot Chat window shows a user input field with placeholder text "whatever text the user has entered" and a response from the AI: "I'll help you create a simple React webpage that uses OpenAI's API to summarize text." To the right, a code editor window is visible, showing the beginning of a React component with code like "function App() {". A large, stylized blue and purple AI character icon is positioned in the bottom right corner of the slide.

whatever text the user has entered

I'll help you create a simple React webpage that uses OpenAI's API to summarize text.

Share conversation

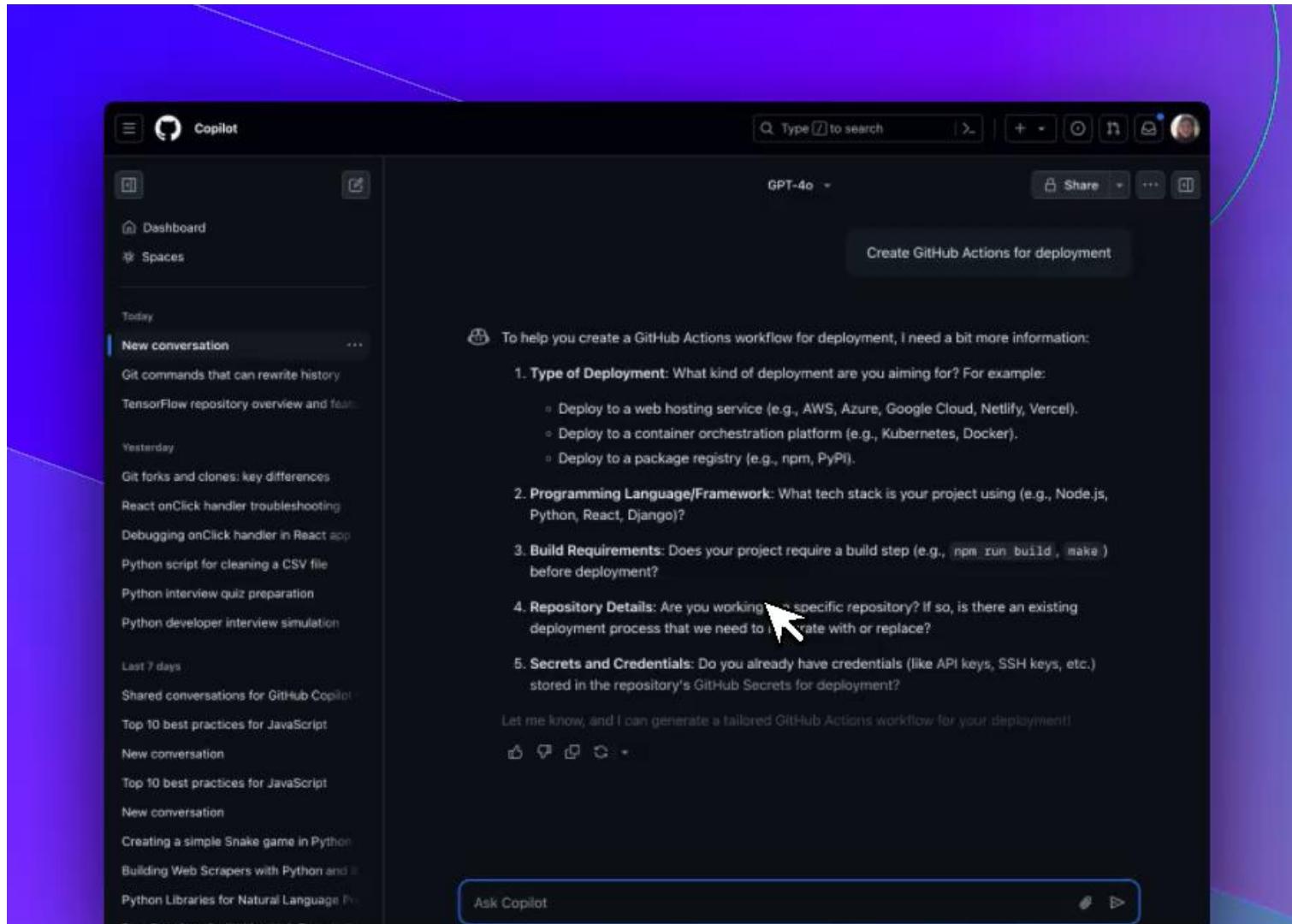
This conversation and future replies are visible to anyone with the link.

<https://github.com/copilot/share/..>

Share

```
function App() {  
  const [text, setText] = useState("");  
  const [summary, setSummary] = useState("")  
  const [isLoading, setIsLoading] = useState(false)  
  
  const handleGenerateSummary = async () =>  
    if (!text.trim) return;  
    setIsLoading(true);  
    const response = await fetch(`https://api.openai.com/v1/...`);  
    const data = await response.json();  
    setSummary(data.summary);  
    setIsLoading(false);  
  }  
  
  return (  
    <div>  
      <input type="text" value={text} onChange={e => setText(e.target.value)} />  
      <button onClick={handleGenerateSummary}>Generate Summary</button>  
      {isLoading ? <div>Generating summary...</div> : summary}  
    </div>  
  );  
}  
  
export default App;
```

Share Copilot Chat conversations in public preview



Organization custom instructions now available

Code, planning, and automation

- Repository
- Codespaces
- Planning
- Copilot**
- Access
- Policies
- Models
- Custom instructions**
- Knowledge bases
- Custom model
- Content exclusion
- Actions
- Webhooks
- Discussions

Customization

Teach Copilot your coding standards, languages, and frameworks. Keep instructions high-level, not about specific workflows. Instructions apply to all queries once saved and may not work perfectly. Press the test customization button to test.

Preferences and instructions

- Prefer writing <language> if no language is specified.
- Use <package manager> for <language> dependencies
- Prioritize <knowledge base> when asking about <topic>.
- Respond with <bullet points/minimal preamble>.

Preview

214 / 600 characters

Discard changes **Save changes**

GitHub Copilot for Xcode adds @workspace and support for Claude 3.7 Sonnet and GPT-4.5



GitHub Copilot code review now supports C, C++, Kotlin, and Swift

Expanded Language Support

- Now supports C, C++, Kotlin, Swift, and more.
- Covers over 90% of file types typically found in pull requests.

Improved Review Quality

- Higher-quality, more actionable suggestions.
- Better handling of low-confidence and suppressed results.
- Notable improvements for C#, including version-aware suggestions.

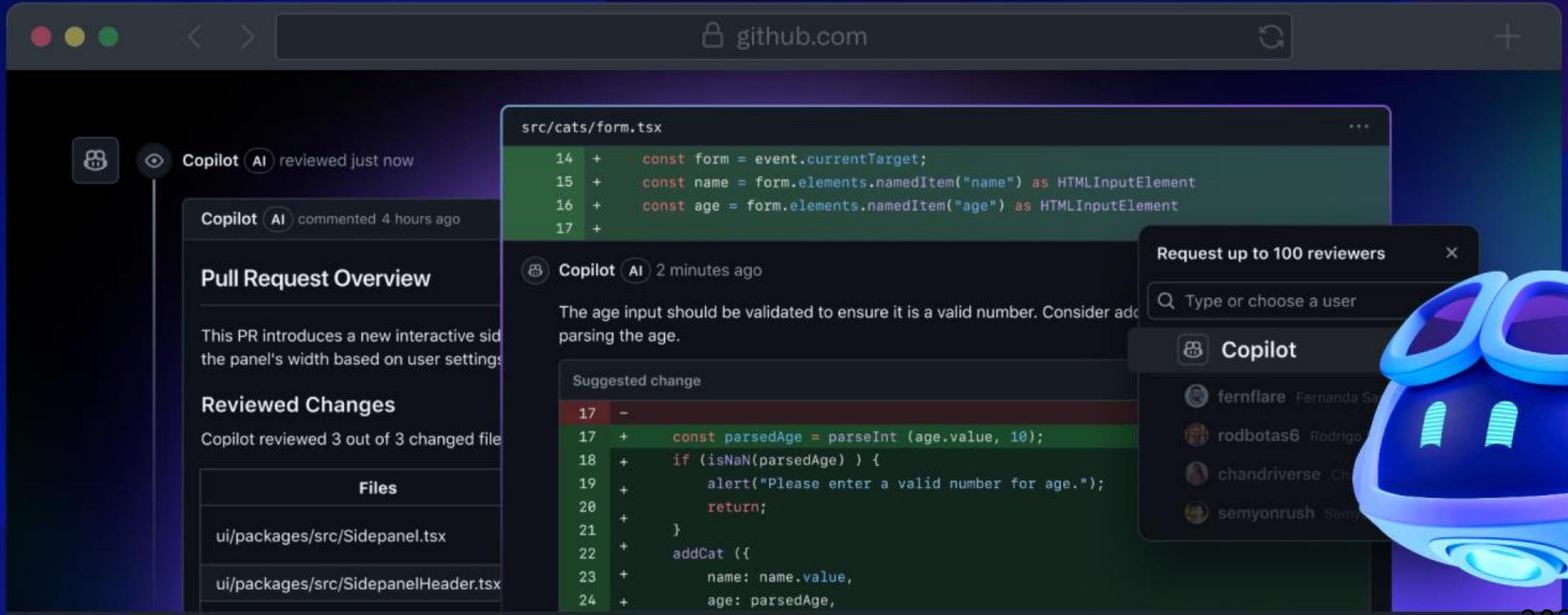
General Availability



GitHub Copilot Edits for JetBrains IDEs

Copilot code review

Supports all languages



A screenshot of a GitHub pull request interface. The browser address bar shows "github.com". The main content area displays a file named "src/cats/form.tsx". The code is as follows:

```
src/cats/form.tsx
14 +   const form = event.currentTarget;
15 +   const name = form.elements.namedItem("name") as HTMLInputElement
16 +   const age = form.elements.namedItem("age") as HTMLInputElement
17 +
```

On the left, a sidebar shows a "Copilot AI" icon and the message "reviewed just now". Below it, another "Copilot AI" icon has the message "commented 4 hours ago". The "Pull Request Overview" section indicates "Copilot AI" has reviewed 3 out of 3 changed files. The "Reviewed Changes" section shows a "Suggested change" for line 17:

```
17 -  
17 +   const parsedAge = parseInt (age.value, 10);  
18 +   if (isNaN(parsedAge) ) {  
19 +     alert("Please enter a valid number for age.");  
20 +     return;  
21 +   }  
22 +   addCat ({  
23 +     name: name.value,  
24 +     age: parsedAge,
```

A modal window titled "Request up to 100 reviewers" is open, with a search bar containing "Type or choose a user". A list of users is shown, with "Copilot" at the top. A 3D rendering of the Copilot AI character, a blue and purple stylized head with a smiling face, is positioned on the right side of the interface.

Enforcement of Copilot premium request limits moved to June 4, 2025

- Enforcement of premium request limits is postponed to **June 4, 2025**.
- Until then, users can continue using premium models without restrictions.

**OpenAI GPT-4.1 is now generally available in GitHub Copilot
as the new default model**