



The AI-powered developer experience

GitHub Copilot

Harnessing generative AI to accelerate innovation in the public sector

Eric Johnson
Dir of SE, PubSec
elstudio@github.com

Software is hard – especially for Public Sector



Technical debt

Burden imposed to manage application code continues to accumulate



Knowledge management gaps

Existing codebases not as well understood



App and data silos

Complex system integrations that are difficult to dissect



AI can help – if it's done well



Reduce monotonous tasks

Project scaffolding, unit testing, code duplication



Employ idiomatic techniques

Write idiomatic code from the outset



Understand existing code

Gain a better understanding of existing codebases and documentation



GitHub Copilot

Your AI pair programmer

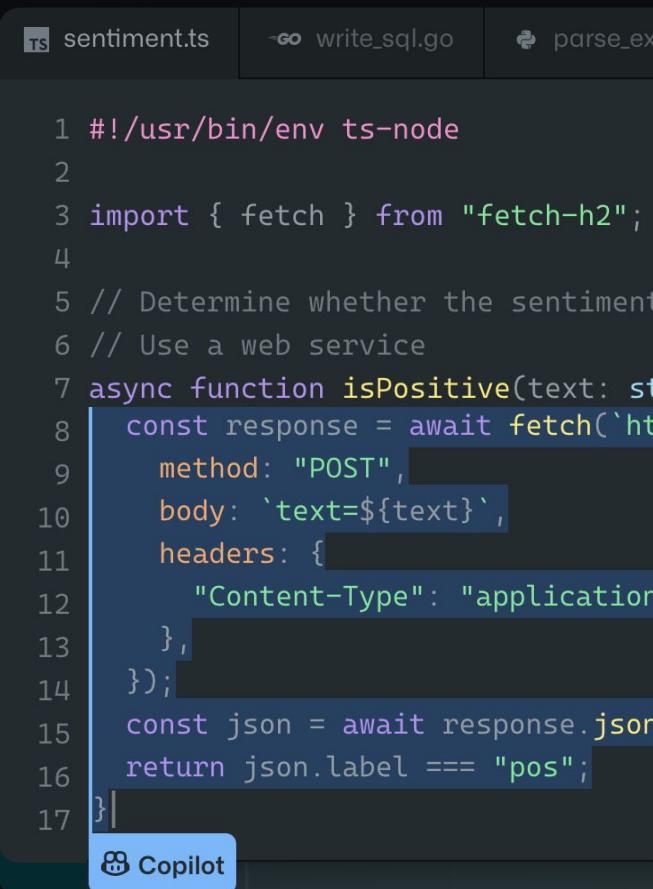
**Uses the context you've provided
and synthesizes code to match**

Convert comments to code

Autofill for repetitive code

Autosuggest tests

Show alternatives



The screenshot shows a dark-themed code editor interface. At the top, there are tabs for 'sentiment.ts' (selected), 'write_sql.go', and 'parse_ex...'. The main area displays a snippet of TypeScript code:

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment
6 // Use a web service
7 async function isPositive(text: st
8   const response = await fetch(`ht
9     method: "POST",
10     body: `text=${text}`,
11     headers: {
12       "Content-Type": "application/
13     },
14   );
15   const json = await response.json();
16   return json.label === "pos";
17 }
```

A blue button at the bottom right of the code area contains the GitHub Copilot logo and the text 'Copilot'.

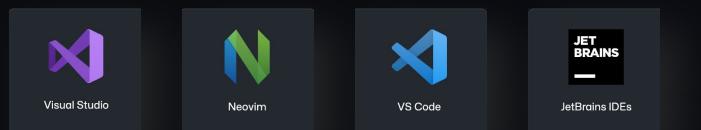




Once enabled...



OpenAI
Codex

A screenshot of a code editor interface. At the top, tabs for "runtime.go", "course.rb", "time.js", and "IsPrimeTest.java" are visible. The main area shows a snippet of Go code:

```
1 package main
2
3 type Run struct {
4     Time int // in milliseconds
5     Results string
6     Failed bool
7 }
```

The numbers 1 through 23 are aligned to the left of the code lines. A light purple bar labeled "Context" is positioned above the code, and a light blue bar labeled "Suggestions" is positioned below it, both with double-headed arrows pointing towards the central code area.



When enabled:

40% of new code
written with GitHub Copilot

30% acceptance rate
for suggested code

75% of devs felt more
fulfilled with their jobs

<https://github.blog/2023-02-14-github-copilot-now-has-a-better-ai-model-and-new-capabilities/>
<https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>
<https://github.blog/2022-06-21-github-copilot-is-generally-available-to-all-developers/>

A screenshot of a code editor interface. The top bar shows tabs for "sentiment.ts", "-go write_sql.go", and "-py parse_ex...". The main area displays a snippet of TypeScript code:

```
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment
6 // Use a web service
7 async function isPositive(text: st
8   const response = await fetch(`ht
9     method: "POST",
10     body: `text=${text}`,
11     headers: {
12       "Content-Type": "application
13     },
14   });
15   const json = await response.json();
16   return json.label === "pos";
17 }
```

A blue callout box with the GitHub Copilot logo and the text "Copilot" is positioned at the bottom right of the code editor window.



Let's demo!



GitHub Copilot

- Product customers can purchase
- Requires little-to-no configuration
- Based on variations of OpenAI's GPT3 language model
- Powered by the OpenAI Codex on Azure
- Optimized and biased for code completion
- Built-in to the IDE
- Can produce suggestions from natural language

OpenAI on Azure

- Generic suite of services and APIs in Azure geared towards those building their own AI-powered apps
- Requires users to directly interface with the APIs and fine-tune the models on their own





For Individuals

- Plugs right into your editor
- Turns natural language prompts into code
- Offers multi-line function suggestions
- Speeds up test generation
- Blocks suggestions matching public code

For Business

Everything included in GitHub Copilot for Individuals, plus

- Simple license management
- Organization-wide policy management
- Industry-leading privacy
- VPN Proxy Support



Q&A



Next steps

Give Copilot a try

- Visit github.com/features/copilot for details
- Email government@github.com to connect with the Public Sector sales team

Your AI pair programmer

GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time, right from your editor.

[Get Copilot for Business >](#) [Compare plans](#)

The screenshot shows a dark-themed GitHub Copilot interface. At the top, there's a header with the title "Your AI pair programmer" and a subtext explaining that GitHub Copilot uses the OpenAI Codex to suggest code and entire functions in real-time. Below the header are two buttons: "Get Copilot for Business >" and "Compare plans". The main area is a code editor with several tabs open: "sentiments.ts", "write_sql.go", "parse_expenses.py", and "addresses.rb". The "sentiments.ts" tab contains the following TypeScript code:

```
1 //!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17}
```

A blue button labeled "Copilot" is visible at the bottom of the code editor. A tooltip below the button says "The AI-Powered Developer Experience". In the bottom right corner, there's a GitHub logo and the handle "@elstudio".



Thank you

