



GitHub Copilot Business for Developer Productivity & Happiness

*joined by Laetitia Maar
new intern at Microsoft*

Ricardo Niepel
Senior Technology Specialist - Azure Application Innovation



github.com/RicardoNiepel



ricardo.niepel@microsoft.com

AI will define
the developer
experience



The Copilot effect.

The world's most widely adopted AI developer tool.

Maximize developer happiness

Increase developer productivity

Accelerate software development



Developer
productivity
with Copilot
goes beyond
speed

87%

less **mental effort**
on repetitive tasks

74%

focus on more
satisfying work

73%

say they are more
in the flow





GitHub Copilot

Your AI pair programmer

**Offers code suggestions,
contextualized to your project**

Convert comments to code

Autofill for repetitive code

Autosuggest tests

Show alternatives



```
sentiment.ts  write_sql.go  parse_expense

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

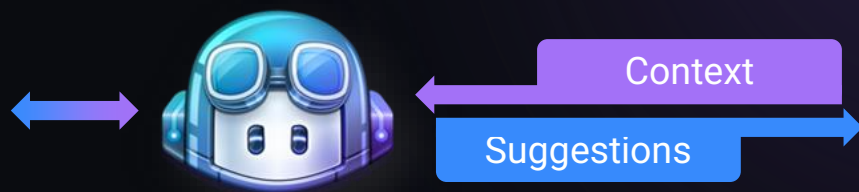
Copilot

GitHub Copilot

Once enabled...



OpenAI LLM

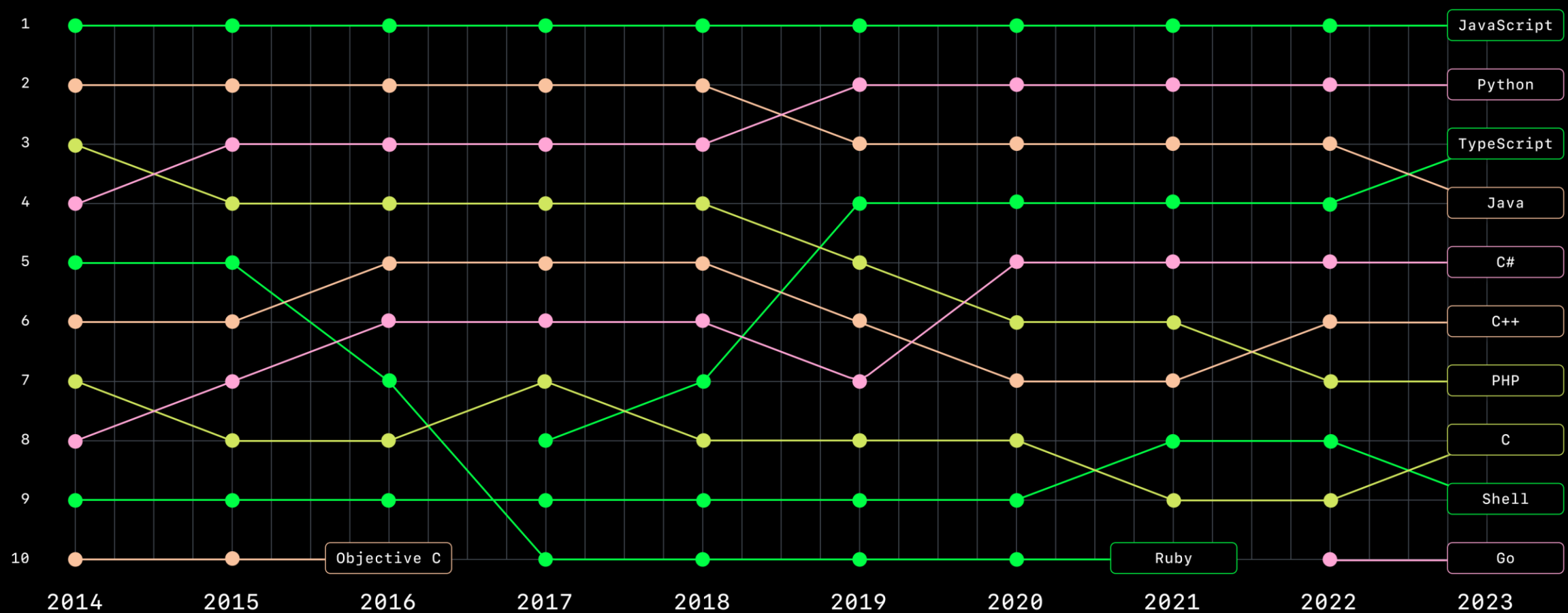


```
runtime.go course.rb time.js IsPrimeTest.java

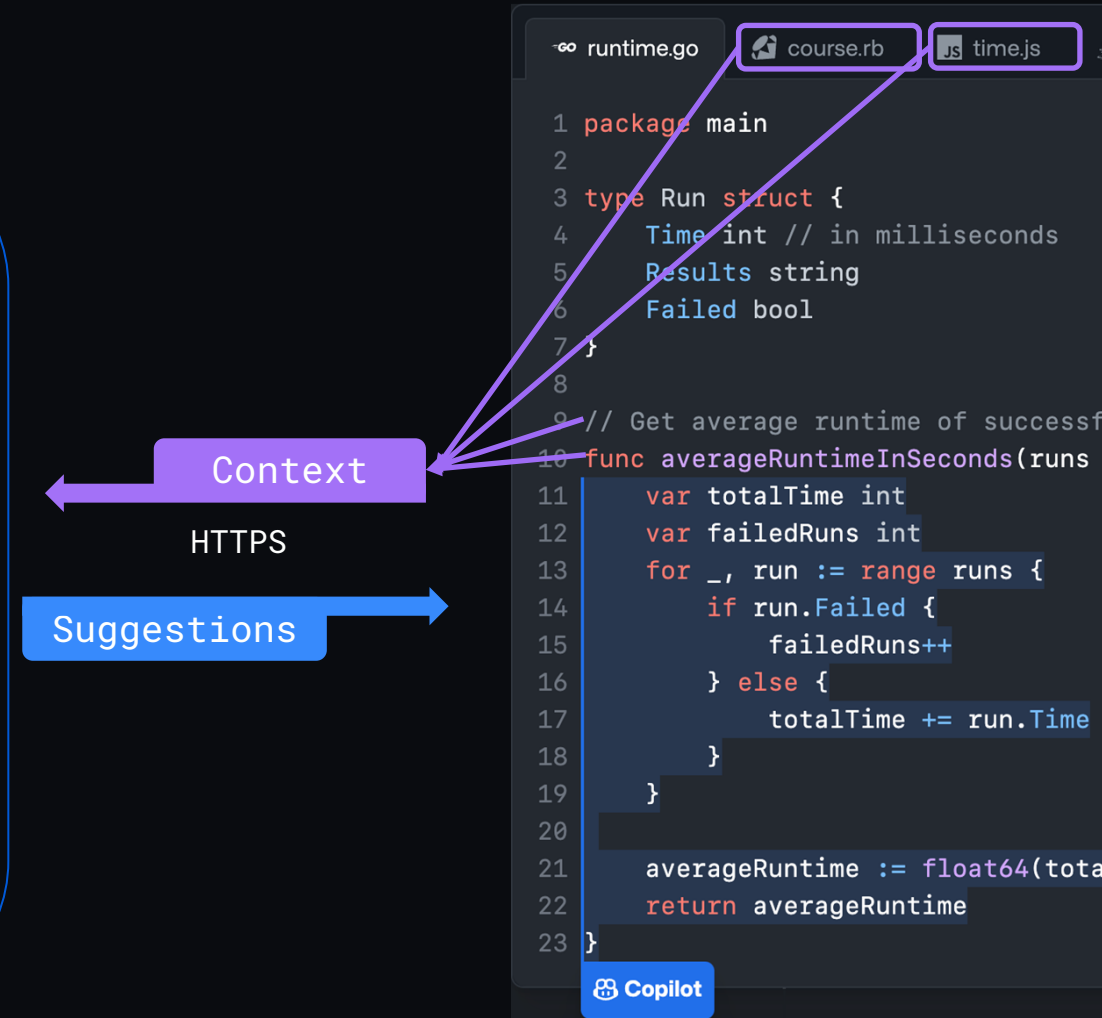
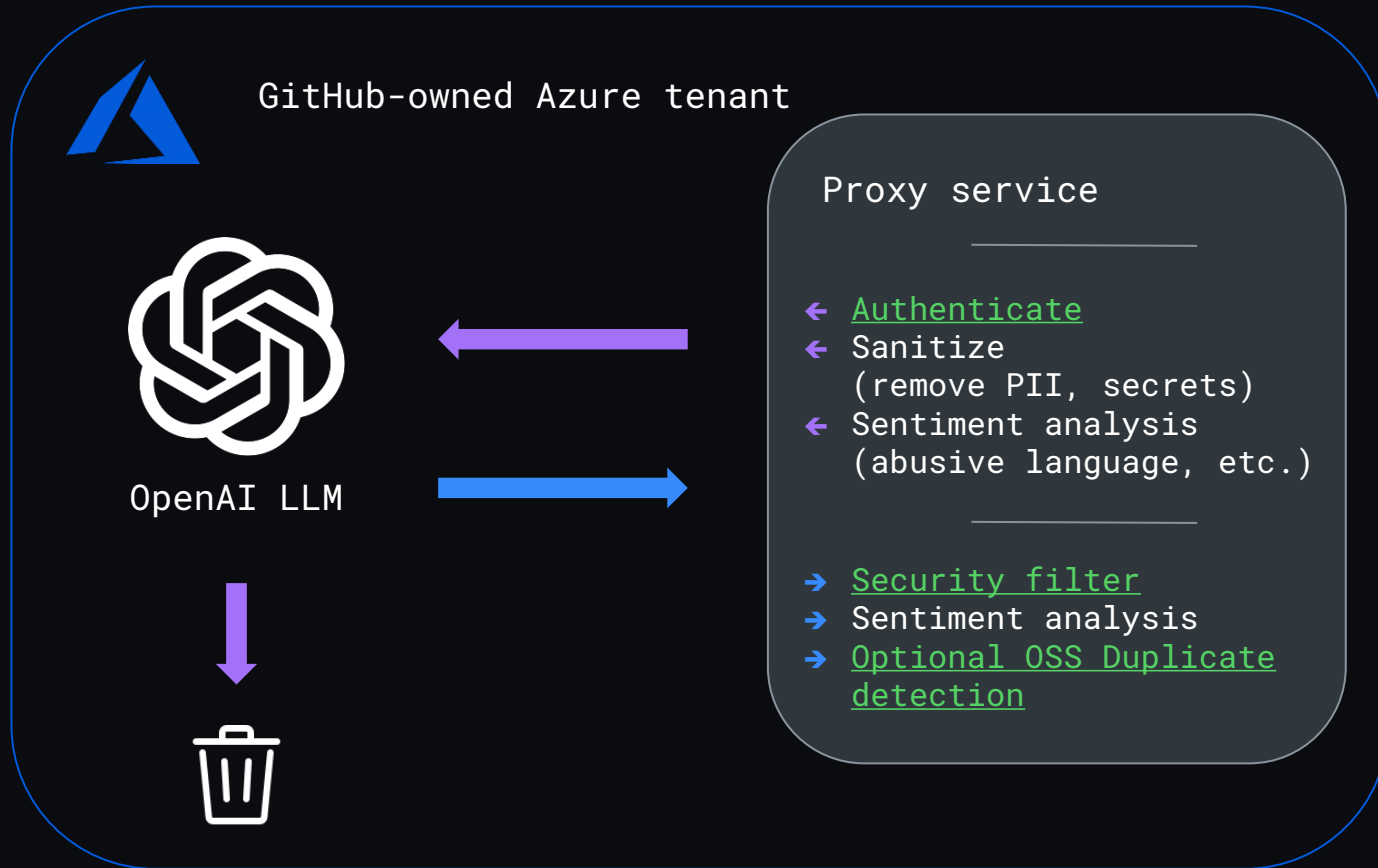
1 package main
2
3 type Run struct {
4     Time int // in milliseconds
5     Results string
6     Failed bool
7 }
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
```



Top 10 programming languages on GitHub



Ephemerality, risk, & indemnity

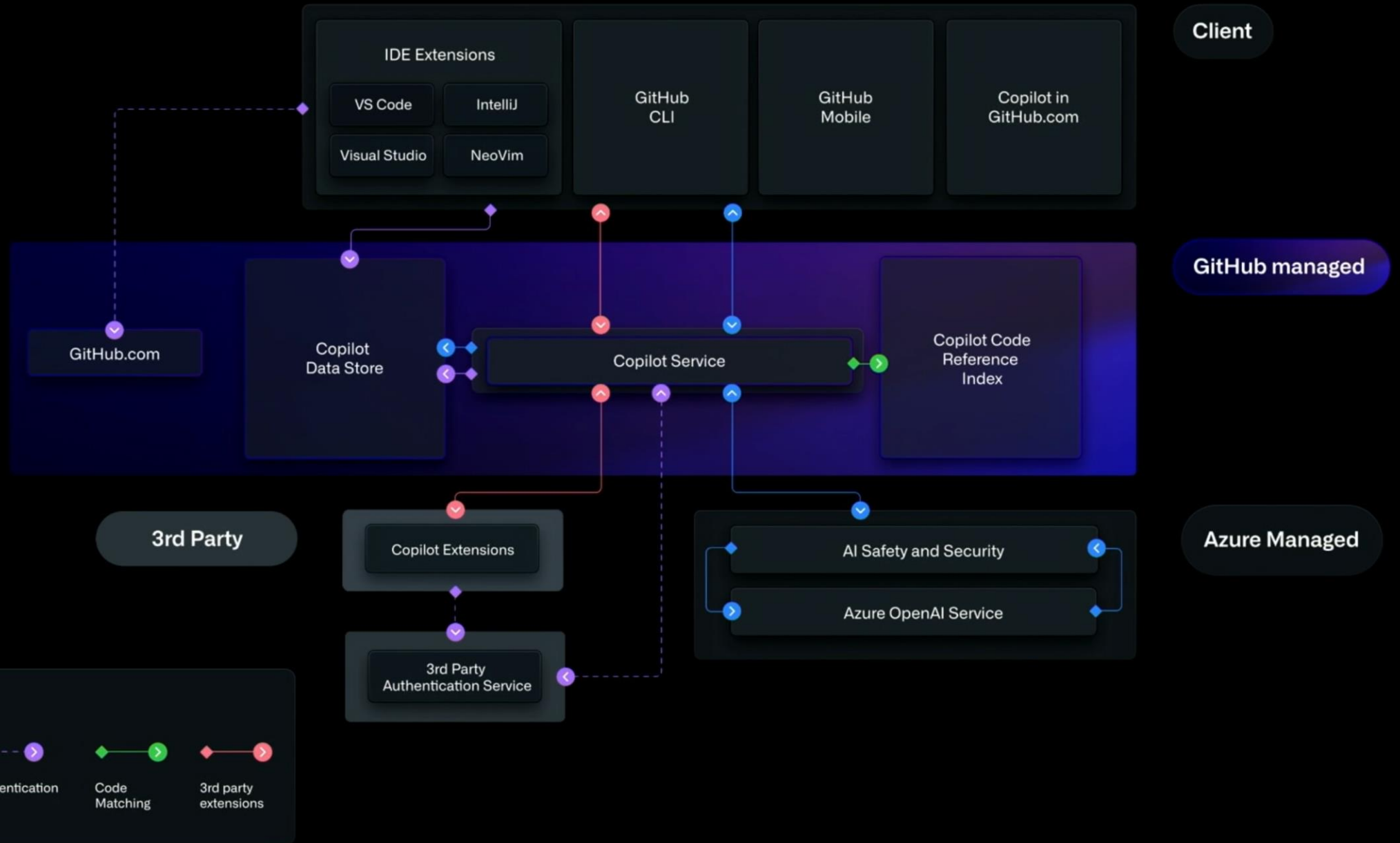


```
runtime.go course.rb time.js
1 package main
2
3 type Run struct {
4     Time int // in milliseconds
5     Results string
6     Failed bool
7 }
8
9 // Get average runtime of successful runs
10 func averageRuntimeInSeconds(runs []Run) float64 {
11     var totalTime int
12     var failedRuns int
13     for _, run := range runs {
14         if run.Failed {
15             failedRuns++
16         } else {
17             totalTime += run.Time
18         }
19     }
20
21     averageRuntime := float64(totalTime) / float64(len(runs) - failedRuns)
22     return averageRuntime
23 }
```

Copilot



GitHub Copilot Data Flow





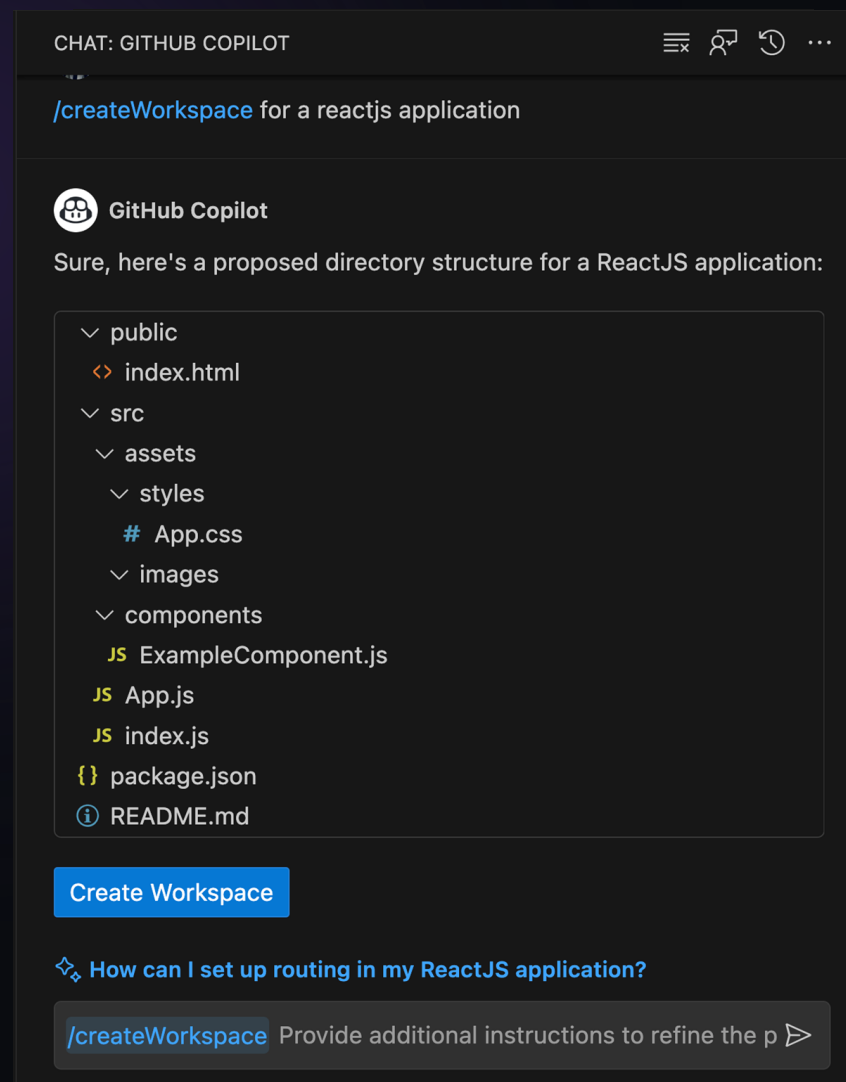
Demos



What we missed...

/search to query your workspace

/createWorkspace to create a workspace for your project





What we missed...

/createNotebook to
generate a Jupyter
Notebook based on
your requirements

CHAT: GITHUB COPILOT

GitHub Copilot

Hi @vfm2, how can I help you?

I'm powered by AI, so surprises and mistakes are possible. Make sure to verify any generated code or suggestions, and [share feedback](#) so that we can learn and improve.

vfm2

[/createNotebook](#) load csv into dataframe

GitHub Copilot

Here's an outline for a Jupyter notebook that loads a CSV into a DataFrame:

- **Import Required Libraries**
- **Load CSV into DataFrame**
- **Display DataFrame**

Create Notebook

💡 How can I filter this DataFrame based on a specific column value?

Ask Copilot or type '/' for commands





What we missed...

Test Driven
Development (TDD)
practice is possible

```
1 // Test functions in sample-function.js
2
3 // Import the functions to be tested
4 const { add, subtract, multiply, divide } = require('./sample-function');
5
6 // Test add function
7 test('add 1 + 2 to equal 3', () => {
8   expect(add(1, 2)).toBe(3);
9 })
10
11 // Test subtract function
12 test('subtract 2 - 1 to equal 1', () => {
13   expect(subtract(2, 1)).toBe(1);
14 })
15
16 // Test multiply function
17
```





What we missed...

Refactoring through GitHub Copilot Chat

CHAT: GITHUB COPILOT

any generated code or suggestions, and [share feedback](#) so that we can learn and improve.

bryantson

Can you help me to make sure that the functions can handle exception cases like null or divide by zero?

GitHub Copilot

Sure, I can help you with that. For the `add` function, we can add a check to see if either `a` or `b` is `null` or `undefined`, and if so, return `null`. For the `addAll` function, we can add a check to see if any of the arguments are `null` or `undefined`, and if so, return `null`. For the `isPrime` function, we don't need to handle `null` or `undefined` cases, but we can add a check to see if `n` is `0` or `1`, and if so, return `false`.

Here's the updated code:

```
// Function to add two numbers
function add(a, b) {
  if (a == null || b == null) {
    return null;
  }
  return a + b;
}

// Function to add all numbers in list
function addAll(...args) {
  if (args.some(arg => arg == null)) {
    return null;
  }
  return args.reduce((a, b) => a + b, 0);
}

// Function to find out if a number is prime
function isPrime(n) {
  if (n == 0 || n == 1) {
    return false;
  }
  for (let i = 2; i <= Math.sqrt(n); i++) {
    if (n % i == 0) {
      return false;
    }
  }
  return true;
}
```

Can you suggest any other common exception cases that I should handle in my code?

Ask Copilot or type '/' for commands

JS sample-function.js

JavaScriptTDD > JS sample-function.js > isPrime

```
1 // Function to add two numbers
2 function add(a, b) {
3   return a + b;
4 }
5
6 // Function to add all numbers in list
7 function addAll(...args) {
8   return args.reduce((a, b) => a + b, 0);
9 }
10
11 // Function to find out if a number is prime
12 function isPrime(n) {
13   if (n < 2) {
14     return false;
15   }
16   for (let i = 2; i <= Math.sqrt(n); i++) {
17     if (n % i == 0) {
18       return false;
19     }
20   }
21   return true;
22 }
```



What we missed...

GitHub Copilot

CPU usage tool

The screenshot displays the Visual Studio IDE with the CPU Usage tool open. The tool shows a list of functions and their CPU usage percentages. The 'IoContainer.ClearContext(string)' function is highlighted, showing a total CPU usage of 8722 (52.96%) and a self CPU usage of 797 (4.84%).

Function Name	Total CPU [unit, %]	Self CPU [unit, %]	Module	Category
Autolnsights (PID: 30324)	16469 (100.00%)	196 (1.19%)	Multiple modules	
IoContainer.Main(System.String[])	15848 (96.23%)	6 (0.04%)	autolnsights	
IoContainer.ClearContext(string)	8722 (52.96%)	797 (4.84%)	autolnsights	
IoContainer.RegisterContextSpecific(string)	7109 (43.17%)	195 (1.18%)	autolnsights	
IoContainer.ctor()	11 (0.07%)	1 (0.01%)	autolnsights	
[External Call] System.StartupHookProvider.Pr...	258 (1.57%)	258 (1.57%)	System.Private.Co...	
[External Call] System.Threading.Thread.StartC...	118 (0.72%)	118 (0.72%)	System.Private.Co...	
[Unwalkable]	24 (0.15%)	24 (0.15%)		
[External Call] System.AppContext.Setup(wcha...	11 (0.07%)	11 (0.07%)	System.Private.Co...	
[External Call] System.Threading.IOCompleto...	6 (0.04%)	6 (0.04%)	System.Private.Co...	
[External Call] System.Gen2GcCallback.Finalize()	4 (0.02%)	4 (0.02%)	System.Private.Co...	
[External Call] dynamicClass_IL_STUB_ReverseP...	2 (0.01%)	2 (0.01%)	System.Private.Co...	
[External Call] System.Threading.ThreadPool.E...	2 (0.01%)	2 (0.01%)	System.Private.Co...	

The GitHub Copilot chat window is open on the right, displaying a message from Mark Downie about the performance of the `ConcurrentDictionary<T>` and a suggestion for improvement.

Mark Downie

The following method uses `System.Collections.Concurrent.ConcurrentDictionary<T>.GetKeys()`:

```
public void ClearContext(string context) { // We can remove while iterating over the keys since ConcurrentDictionary returns a // readonly copy of keys in the Keys property foreach (var key in this.registeredObjects.Keys) { if (key.EndsWith(ContextSeparator + context, StringComparison.Ordinal)) { bool removed = this.registeredObjects.TryRemove(key, out object value); if (removed && value is IDisposable disposableInstance) { try { disposableInstance.Dispose(); } catch (Exception e) { Debug.Fail(e.Message); } } } } } Explain how this takes a lock and makes a copy and give a fix that enumerates KeyValues for efficient access to keys and avoid using inside loops. Also explain why this fix is better for performance."
```

Copilot

The `ConcurrentDictionary<T>` is thread-safe and implements multiple locks to ensure access from multiple threads is optimized for performance. The `GetKeys()` method within it, internally calls the `GetEnumerator()` method which takes a lock and creates a copy of the key collection before returning. Therefore, enumerating through the keys one by one in a loop like in the provided code block may not be performant since it may cause multiple locks and can consume too much memory.

To improve the performance, you

AI suggestions might be inaccurate

Ask Copilot



GitHub Copilot

AI hallucinations ...



- Nature of the Model
- Fabricated Outputs
- Highly Creative but not always right
- Reliance on Developers
- Feedback Loop



**We are building
Copilot for the sake of
developer happiness**



55%



Faster Coding



85%



More confident



75%



More fulfilled

<https://gh.io/gai-copilot-happiness>

<https://github.blog/2023-06-13-survey-reveals-ais-impact-on-the-developer-experience>

<https://github.blog/2023-10-10-research-quantifying-github-copilots-impact-on-code-quality>

GitHub Copilot Trust Center

As GitHub's Chief Legal Officer, I understand the nuanced challenges of enabling company-wide AI adoption, especially in an evolving regulatory landscape. GitHub Copilot is enterprise-ready, but organizations need to clearly understand how the tool meets their compliance, security, and accessibility requirements.

Our aim is to bring you that clarity with the **GitHub Copilot Trust Center**.



<https://resources.github.com/copilot-trust-center> &
[How to responsibly adopt GitHub Copilot with the GitHub Copilot Trust Center - The GitHub Blog](#)



Join the Community

- [GH Copilot Discussions & Product Feedback](#)
- [Copilot Monthly Digest](#)
- [GitHub Copilot Articles - The GitHub Blog](#)
- [GitHub Copilot Changes – GitHub Changelog](#)
- [GitHub Insider - Developer Newsletter](#)





General resources for developers ...

- [Essentials of GitHub Copilot – Learning path for managers](#)
- [Understand the AI pair programmer – Learning path for developers](#)
- [Quickstart for GitHub Copilot](#)
- [A Beginner's Guide to Prompt Engineering with GitHub Copilot](#)
- [10 unexpected ways to use GitHub Copilot](#)
- [Use GitHub Copilot Chat as your personal AI assistant for accessibility](#)





Specific resources for **Visual Studio** users ...

- [AI-assisted development in Visual Studio - Microsoft Learn](#)
- [Code Faster and Better with GitHub Copilot: Slash Commands and Context Variables](#)
- [Simplified Code Refinement and Debugging with GitHub Copilot Chat](#)
- [Quick Tips for using Copilot Chat with C++ in Visual Studio](#)
- [Copilot Chat with C++ in Visual Studio: Generate code, fix functions, ...](#)





Specific resources for **JetBrains IDEs** users ...

- [Getting started with GitHub Copilot in JetBrains IDEs](#)
- [Using GitHub Copilot Chat in your IDE](#)
- [GitHub Copilot - IntelliJ IDEs Plugin | Marketplace \(jetbrains.com\)](#)
- [GitHub Copilot Chat General Availability in JetBrains IDE](#)



GitHub Copilot

Built for tomorrow,
ready for today.

