

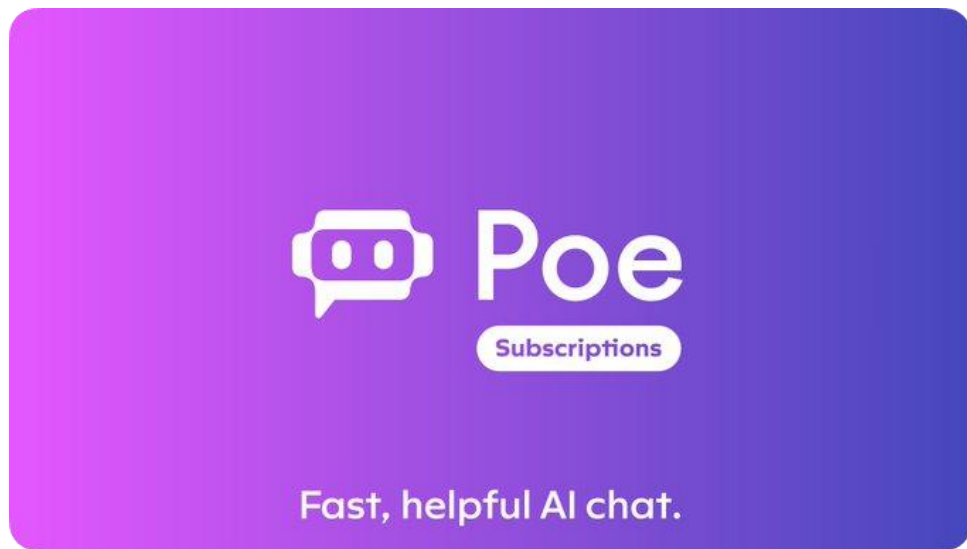
利用 AI 提升开发效率

Boosting Development Efficiency with AI

AI 说

通过充分利用人工智能（AI）技术，开发人员能够在软件开发过程中实现效率的飞跃。AI 的智能化工具和技术为开发人员提供了自动化代码生成、智能代码分析、自动化测试和质量保证等功能，以及项目管理和团队协作的支持，从而加速创新、减少重复工作、提高软件质量，并使开发者能够更好地应对快节奏的开发环境和不断增长的需求。

01 / POE



www.poe.com

代码提示

例：帮我用 JS 写一个并查集

Assistant Poe

下面是使用 JavaScript 实现并查集（Disjoint Set）的基本代码示例：

javascript

Copy

```
class DisjointSet {
  constructor(size) {
    this.parent = new Array(size);
    this.rank = new Array(size);

    // 初始化并查集，每个元素的父节点为自身，秩（rank）为 0
    for (let i = 0; i < size; i++) {
      this.parent[i] = i;
      this.rank[i] = 0;
    }
  }

  // 查找根节点（代表元素）并进行路径压缩
  find(x) {
    if (this.parent[x] !== x) {
      this.parent[x] = this.find(this.parent[x]);
    }
    return this.parent[x];
  }

  // 合并两个集合
  union(x, y) {
    let rootX = this.find(x);
    let rootY = this.find(y);

    if (rootX !== rootY) {
      // 根据秩进行合并，将秩较小的集合合并到秩较大的集合上
      if (this.rank[rootX] < this.rank[rootY]) {
        this.parent[rootX] = rootY;
      } else if (this.rank[rootX] > this.rank[rootY]) {
        this.parent[rootY] = rootX;
      } else {
        this.parent[rootY] = rootX;
        this.rank[rootX]++;
      }
    }
  }
}
```

使用示例：

javascript

Copy

```
const ds = new DisjointSet(5);

ds.union(0, 1);
ds.union(2, 3);
ds.union(3, 4);

console.log(ds.find(1)); // 输出: 0
console.log(ds.find(3)); // 输出: 2
console.log(ds.find(4)); // 输出: 2
```

上述代码实现了一个基本的并查集数据结构，包括初始化、查找根节点和合并两个集合的操作。你可以根据需要对并查集进行扩展，添加其他功能或优化。

例：帮我用 JS 实现一个布隆过滤器

Assistant Poe

当然，我可以帮你实现一个基本的布隆过滤器（Bloom Filter）的示例代码。以下是使用 JavaScript 实现的简单布隆过滤器：

javascript

Copy

```
class BloomFilter {
  constructor(size, hashFunctions) {
    this.size = size;
    this.bitArray = new Array(size).fill(0);
    this.hashFunctions = hashFunctions;
  }

  add(item) {
    for (let fn of this.hashFunctions) {
      const index = fn(item) % this.size;
      this.bitArray[index] = 1;
    }
  }

  contains(item) {
    for (let fn of this.hashFunctions) {
      const index = fn(item) % this.size;
      if (this.bitArray[index] === 0) {
        return false;
      }
    }
    return true;
  }
}
```

// 示例用法

```
const hash1 = (item) => {
  let hash = 0;
  for (let i = 0; i < item.length; i++) {
    hash = (hash << 5) - hash + item.charCodeAt(i);
    hash = hash & hash; // 确保为32位有符号整数
  }
  return hash;
};

const hash2 = (item) => {
  let hash = 5381;
  for (let i = 0; i < item.length; i++) {
    hash = (hash * 33) ^ item.charCodeAt(i);
  }
  return hash;
};



const filter = new BloomFilter(10, [hash1, hash2]);

// 添加元素到布隆过滤器
filter.add("apple");
filter.add("banana");
filter.add("orange");

// 检查元素是否存在于布隆过滤器
console.log(filter.contains("apple")); // 输出: true
console.log(filter.contains("banana")); // 输出: true
console.log(filter.contains("orange")); // 输出: true
console.log(filter.contains("grape")); // 输出: false
```

框架或库的推荐

例：我需要一个 Go 的热更新工具，有推荐的吗



 Assistant 

在Go语言中，有一些热更新工具可以帮助你实现代码的热更新和动态重载。以下是一些常用的Go热更新工具：

1. Reflex: Reflex是一个简单易用的热更新工具，它可以在代码发生更改时自动重新编译和运行应用程序。它支持监视文件更改并触发重新编译，适用于开发过程中的快速迭代。
2. Fresh: Fresh是另一个流行的Go热更新工具，它可以在代码发生更改时自动重新构建和重启应用程序。它还提供了一些配置选项，可以根据需求自定义构建和运行过程。
3. Gin: Gin是一个轻量级的Web框架，具有自动热更新的功能。它内置了一个名为gin的命令行工具，可以在开发模式下实现代码的热更新，使得在修改代码后无需手动重新编译和重启应用程序。
4. Air: Air是一个功能强大的热更新工具，它支持监视文件更改并自动重新编译和重启应用程序。它提供了一些高级功能，如自定义构建脚本、环境变量管理等，可以满足复杂应用程序的需求。

这些工具都是开源的，并且在Go社区中被广泛使用。你可以根据自己的需求和偏好选择适合你的热更新工具。在使用这些工具之前，建议查阅它们的文档和示例代码，以了解如何正确配置和集成到你的项目中。

例：React 的状态管理有哪些好的推荐

 Assistant 

在React中，有许多优秀的状态管理库可供选择。以下是一些受欢迎且经常被推荐的React状态管理库：

1. Redux: Redux是一个可预测的状态容器，广泛用于管理React应用程序的状态。它采用单一的全局状态树，并通过Redux的核心概念（如actions、reducers和store）来管理状态和数据流。Redux具有丰富的生态系统和强大的工具支持。
2. MobX: MobX是一个简单而强大的状态管理库，它使用可观察的数据结构来自动追踪状态的变化并更新相关的组件。相比Redux，MobX提供了更直观的编程模型，使状态管理变得简单和直观。
3. Zustand: Zustand是一个轻量级的状态管理库，它通过使用React的Hook和Context API，提供了简单、直观的状态管理解决方案。Zustand的设计目标是保持API简洁且易于使用，同时提供高性能和可扩展性。
4. Recoil: Recoil是由Facebook开发的状态管理库，旨在简化复杂应用程序的状态管理。它利用React的新特性——React Hooks，提供了一个基于原子（Atoms）和选择器（Selectors）的状态管理模型，使状态管理变得简单而灵活。

错误分析及解决

```
MongoDB shell version v4.0.1
connecting to: mongodb://127.0.0.1:27017
2023-09-24T14:14:43.938+0800 E QUERY [js] Error: couldn't
connect to server 127.0.0.1:27017, connection attempt failed:
SocketException: Error connecting to 127.0.0.1:27017 :: caused
by :: Connection refused :
connect@src/mongo/shell/mongo.js:257:13
@(connect):1:6
exception: connect failed
```

Assistant Poe

根据提供的日志信息, "Error: couldn't connect to server 127.0.0.1:27017, connection attempt failed: SocketException: Error connecting to 127.0.0.1:27017 :: caused by :: Connection refused" 表示在尝试连接到本地 MongoDB 服务器时出现连接失败的错误。

这种错误通常是由以下原因之一引起的:

1. MongoDB 服务器未运行: 请确保你的 MongoDB 服务器已正确安装并正在运行。你可以尝试启动 MongoDB 服务器并确保它在指定的地址 (127.0.0.1) 和端口 (27017) 上监听连接。
2. 错误的连接地址或端口: 请检查连接地址和端口是否正确。确保你正在连接到正确的 MongoDB 服务器地址和端口。
3. 防火墙或网络配置问题: 防火墙或网络配置可能阻止了连接到 MongoDB 服务器。请确保防火墙允许通过 MongoDB 使用的端口 (默认为 27017) 进行连接, 并且网络配置没有限制你的访问。

翻译

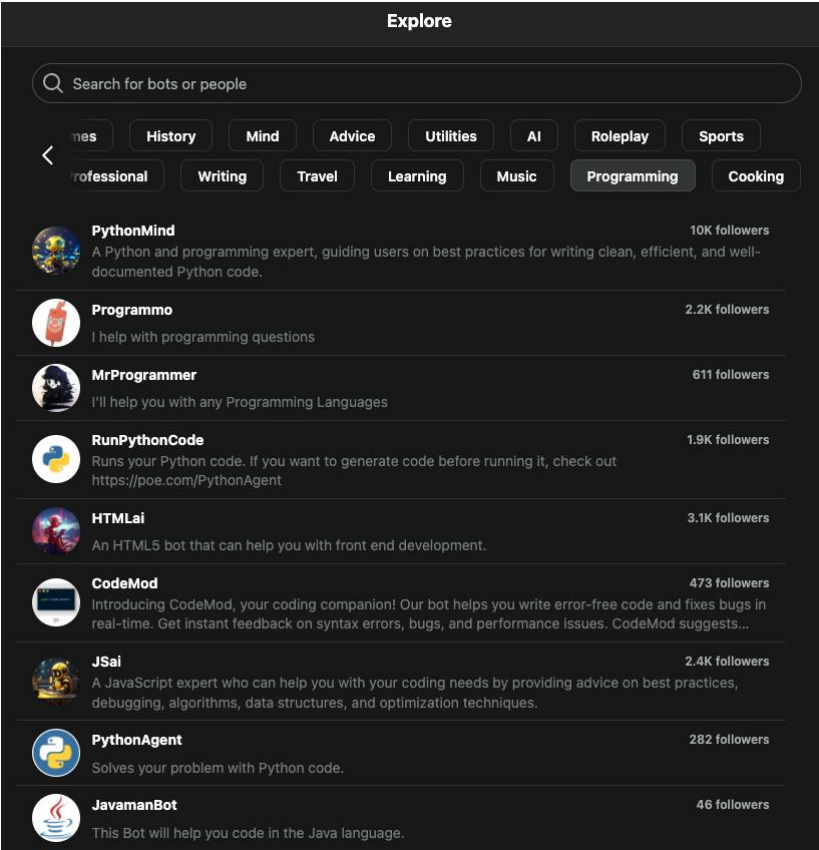
帮我用10个主流的国家的语言翻译这句话: '您有更好意见或建议, 请联系我们: xxx@gmail.com'

Assistant Poe

以下是对句子 "您有更好意见或建议, 请联系我们: xxx@gmail.com" 的翻译, 使用了十种主流国家的语言的示例:

1. 英语 (English): "If you have any better opinions or suggestions, please contact us at xxx@gmail.com."
2. 西班牙语 (Spanish): "Si tiene alguna opinión o sugerencia mejor, contáctenos a xxx@gmail.com."
3. 法语 (French): "Si vous avez de meilleures opinions ou suggestions, veuillez nous contacter à xxx@gmail.com."
4. 德语 (German): "Wenn Sie bessere Meinungen oder Vorschläge haben, kontaktieren Sie uns bitte unter xxx@gmail.com."
5. 俄语 (Russian): "Если у вас есть лучшие мнения или предложения, пожалуйста, свяжитесь с нами по адресу xxx@gmail.com."
6. 日语 (Japanese): "より良い意見や提案がありましたら、お問い合わせください: xxx@gmail.com"
7. 阿拉伯语 (Arabic): "إذا كان لديك أي آراء أو اقتراحات أفضل, يرجى الاتصال بنا على xxx@gmail.com."
8. 葡萄牙语 (Portuguese): "Se você tiver alguma opinião ou sugestão melhor, entre em contato conosco em xxx@gmail.com."
9. 汉语 (中文) (Chinese): "如果您有更好的意见或建议, 请通过 xxx@gmail.com 联系我们。"
10. 韩语 (Korean): "더 좋은 의견이나 제안이 있으시면, xxx@gmail.com 으로 연락 부탁드립니다."

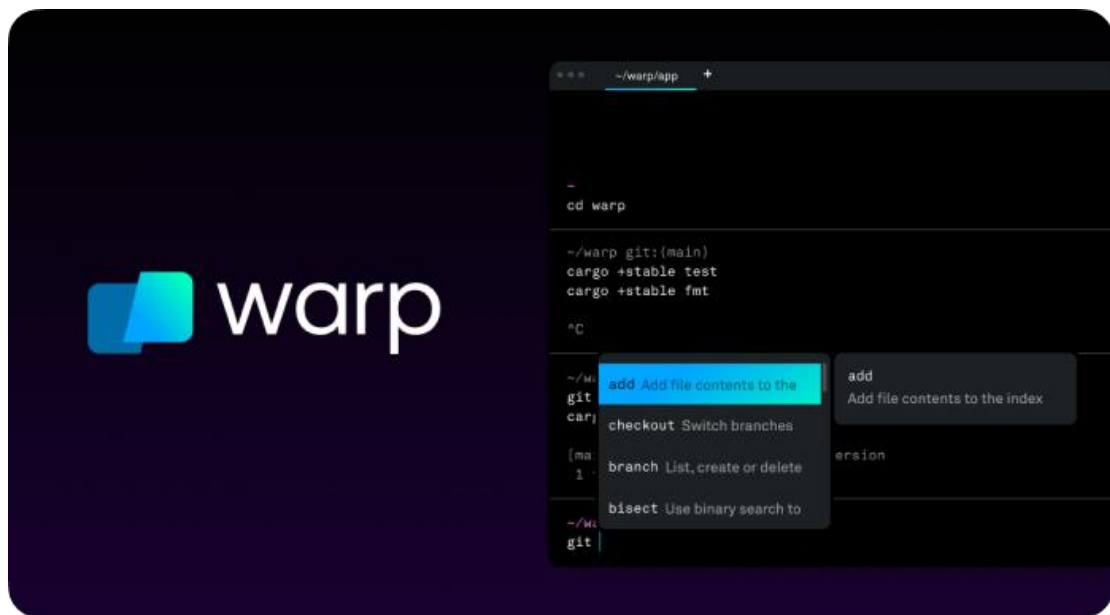
搜索特定功能的 GPT



无语...



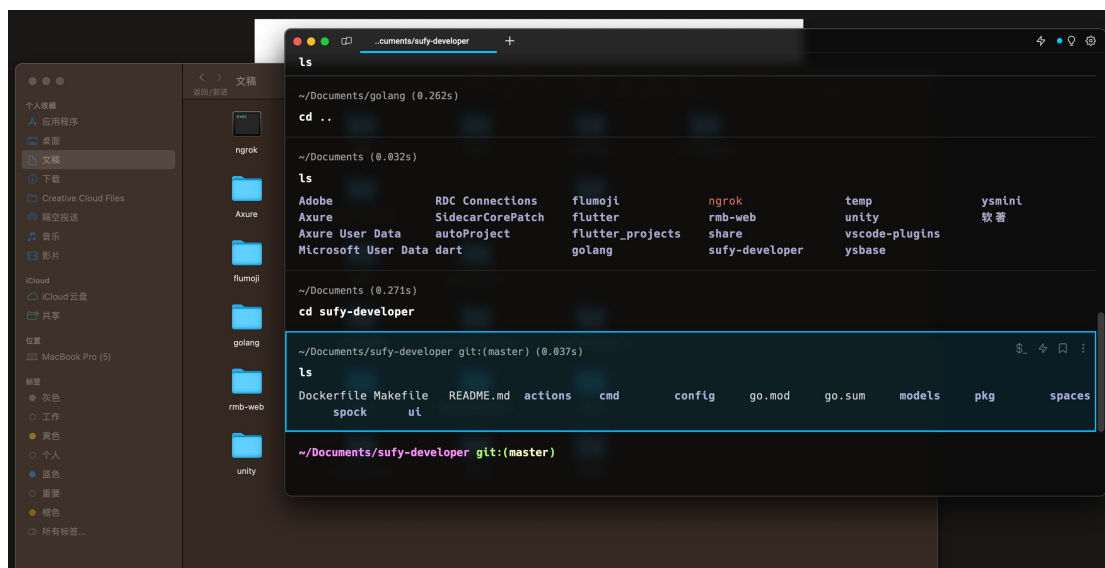
02 / WARP



www.warp.dev

brew install --cask warp

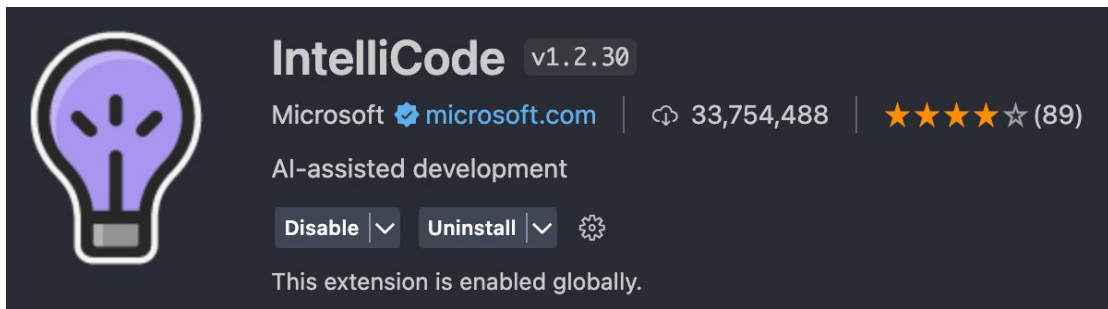
扔掉你的 Terminal 吧



- 命令补全
- 命令区块高亮
- 好用的搜索
- 命令行吸附
- 输入区域始终在面板下方
- 自定义皮肤
- 常用命令保存（工作流），以及使用预留参数
- 账号与工作流同步
- 团队共享工作流
- Warp AI

03 / VSCODE PLUGIN

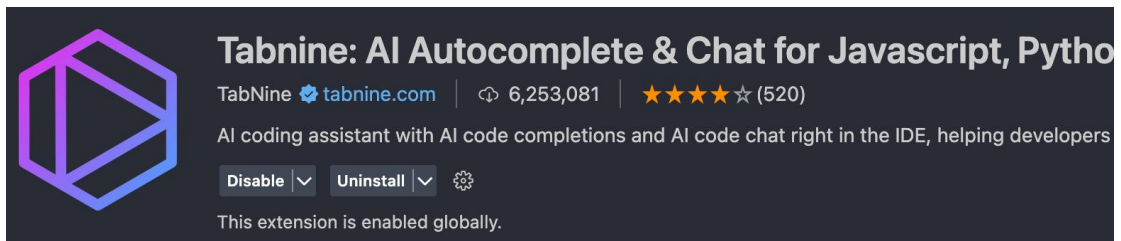
IntelliCode



The image shows the IntelliCode extension interface in VS Code. On the left is a purple lightbulb icon. To its right, the text 'IntelliCode' is followed by the version 'v1.2.30'. Below this, it says 'Microsoft' with a blue checkmark icon and the link 'microsoft.com'. To the right of this is a download icon followed by '33,754,488' and a star rating of four full stars and one half star, with '(89)' in parentheses. Below the download information is the text 'AI-assisted development'. At the bottom of the interface are two buttons: 'Disable' and 'Uninstall', each with a dropdown arrow, and a gear icon for settings. Below these buttons, it says 'This extension is enabled globally.'

IntelliCode v1.2.30
Microsoft microsoft.com | 33,754,488 | ★★★★★ (89)
AI-assisted development
Disable | Uninstall | ⚙️
This extension is enabled globally.

Tabnine



The image shows the Tabnine extension interface in VS Code. On the left is a purple and blue geometric icon. To its right, the text 'Tabnine: AI Autocomplete & Chat for Javascript, Python' is displayed. Below this, it says 'TabNine' with a blue checkmark icon and the link 'tabnine.com'. To the right of this is a download icon followed by '6,253,081' and a star rating of four full stars and one half star, with '(520)' in parentheses. Below the download information is the text 'AI coding assistant with AI code completions and AI code chat right in the IDE, helping developers'. At the bottom of the interface are two buttons: 'Disable' and 'Uninstall', each with a dropdown arrow, and a gear icon for settings. Below these buttons, it says 'This extension is enabled globally.'

Tabnine: AI Autocomplete & Chat for Javascript, Python
TabNine tabnine.com | 6,253,081 | ★★★★★ (520)
AI coding assistant with AI code completions and AI code chat right in the IDE, helping developers
Disable | Uninstall | ⚙️
This extension is enabled globally.

IntelliCode 启用前

```
export const ThisIsAButton = () => {  
  const onClick = () => {  
    axios.  
  }  
  return <Symbol>  
}  
export def arguments  
bind  
call
```

应为 Cancel
CancelToken

IntelliCode 启用后

```
export const ThisIsAButton = () => {  
  const onClick = () => {  
    axios.  
  }  
  return <create>  
}  
export def Cancel  
CancelToken  
Symbol
```

应为 ★ get
★ post
★ create
★ defaults
★ request

Tabnine 启用前

无任何提示...

```
export const ThisIsAButton = () => {  
  const onClick =  
  
  return <div className="button" onClick={onClick}>My Button</div>  
}  
  
export default CarList
```

Tabnine 启用后

💡 export const ThisIsAButton = () => {
 const onClick = You, 1秒钟前 • Uncommitted changes
 return <div cla
}
export default Ca

tabnine

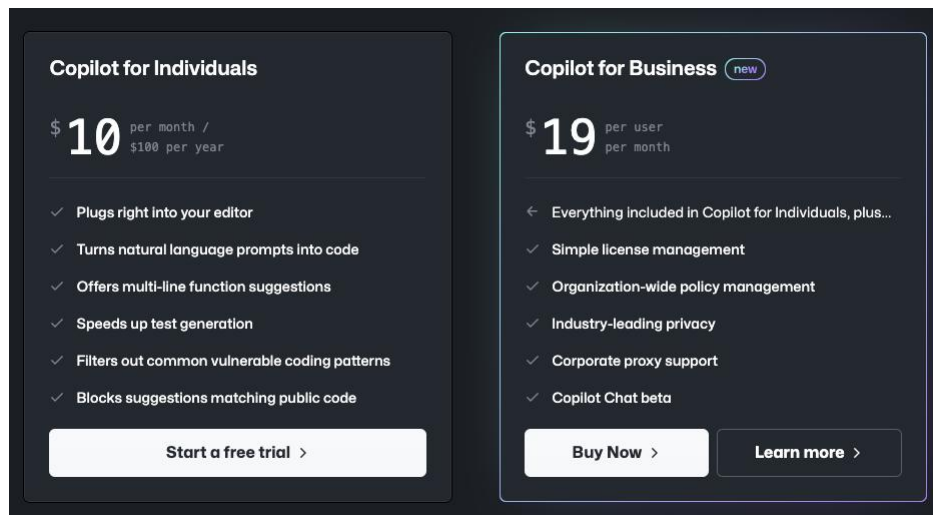
- () => {
- () =>
- (event:
- (event
- this.props

```
export const ThisIsAButton = () => {  
  const onClick = (event: React.MouseEventHandler<HTMLDivElement>) => {  
    if (e) 找不到名称“e”。 Error: 找不到名称“e”。 You, 1秒钟前  
  } 应: event.target.classList.contains tabnine  
    event.button  
  return event.target.classList.cont }>My Button</div> 不能  
    event.target  
    event.target.classList.  
export de event
```

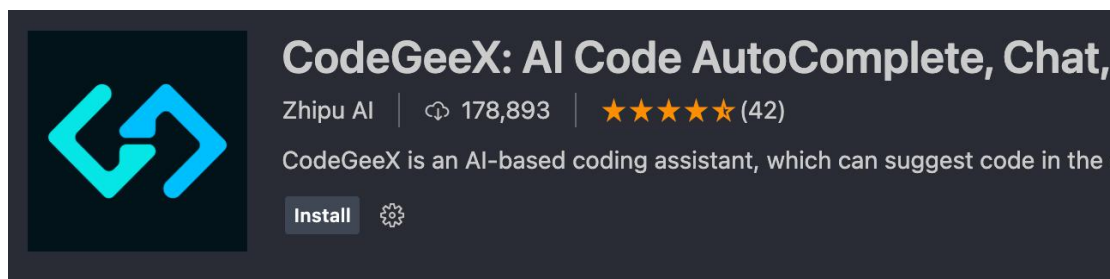
GitHubCopilot



太贵...没试，期待有人尝试并分享...



CodeGeeX



免费的 Copilot 替代产品，你可以试试，反正我删了...