

GitHub Copilot 最佳实践分享 & 常见问题解答

Lichao Zhao – App Dev Specialist
Microsoft China STU

大纲

- GitHub Copilot介绍
- GitHub Copilot工作原理说明
- GitHub Copilot环境配置及基本操作
- GitHub Copilot的能力、适用场景、使用技巧及常见使用问题
- GitHub Copilot Chat的使用场景汇总
- GitHub Copilot常见问题解答
- 附件

GitHub Copilot产品介绍

Copilot 商业版

使编码过程更智能、更简单、更快速

- Copilot - 即时代码补齐
- Chat - IDE 里的结对编程助手
- Copilot CLI - 命令行助手
- 企业级的安全、合规、隐私和管理方式
- IP 归属于用户 & IP 保护
- Copilot 手机端（马上推出）

Copilot 企业版（新版本，暂未上市）

将AI融入到软件开发周期的每个环节

- **包含商业版中的所有功能，再额外包含...**
 - 通用编码问题的Copilot支持
 - 基于您的代码仓库做仓库级的代码解读
 - 基于内部文档的对话式搜索和摘要
 - 拉取请求的描述建议
 - 差异分析和代码审查技能
 - 后端模型的fine tuning

Copilot 个人版

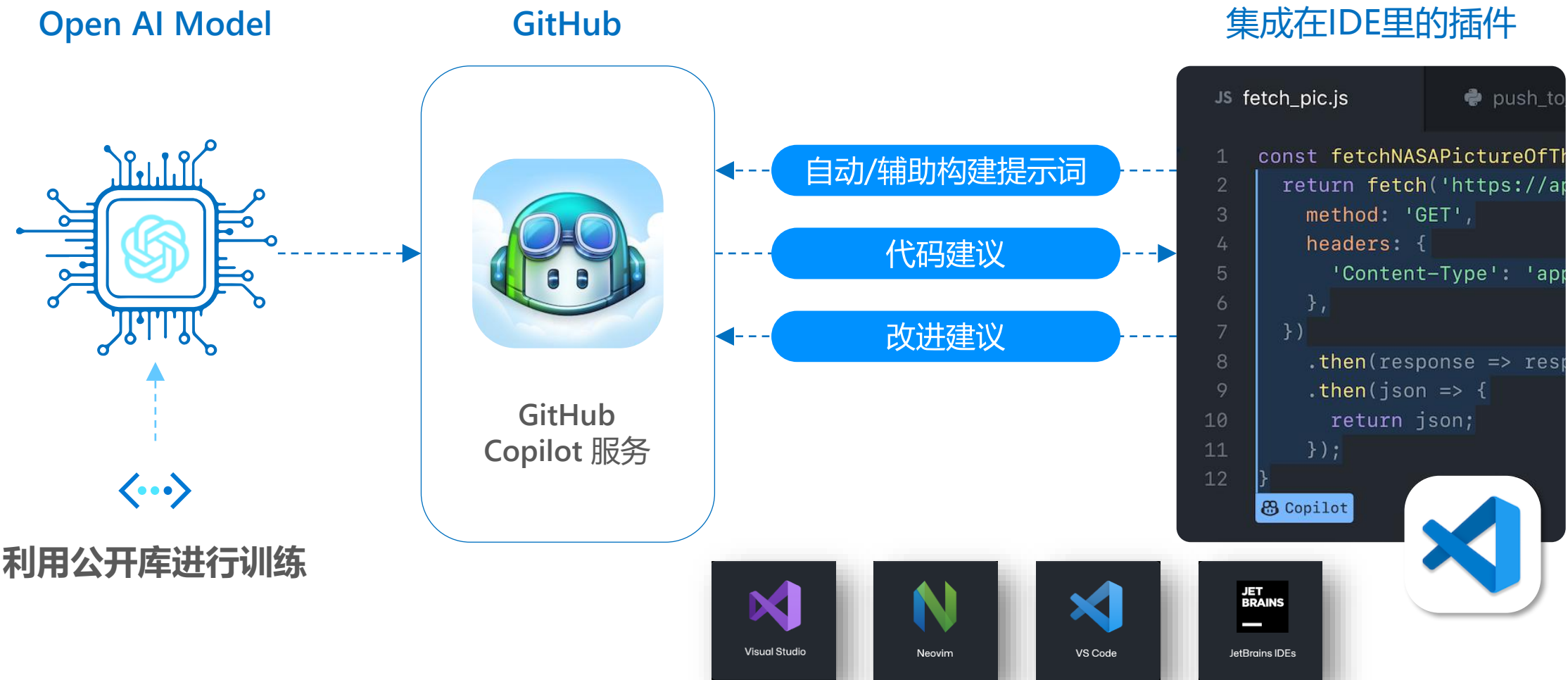
在编辑器中的Copilot，适用个人开发者和自由职业者

GitHub Copilot最佳使用方式

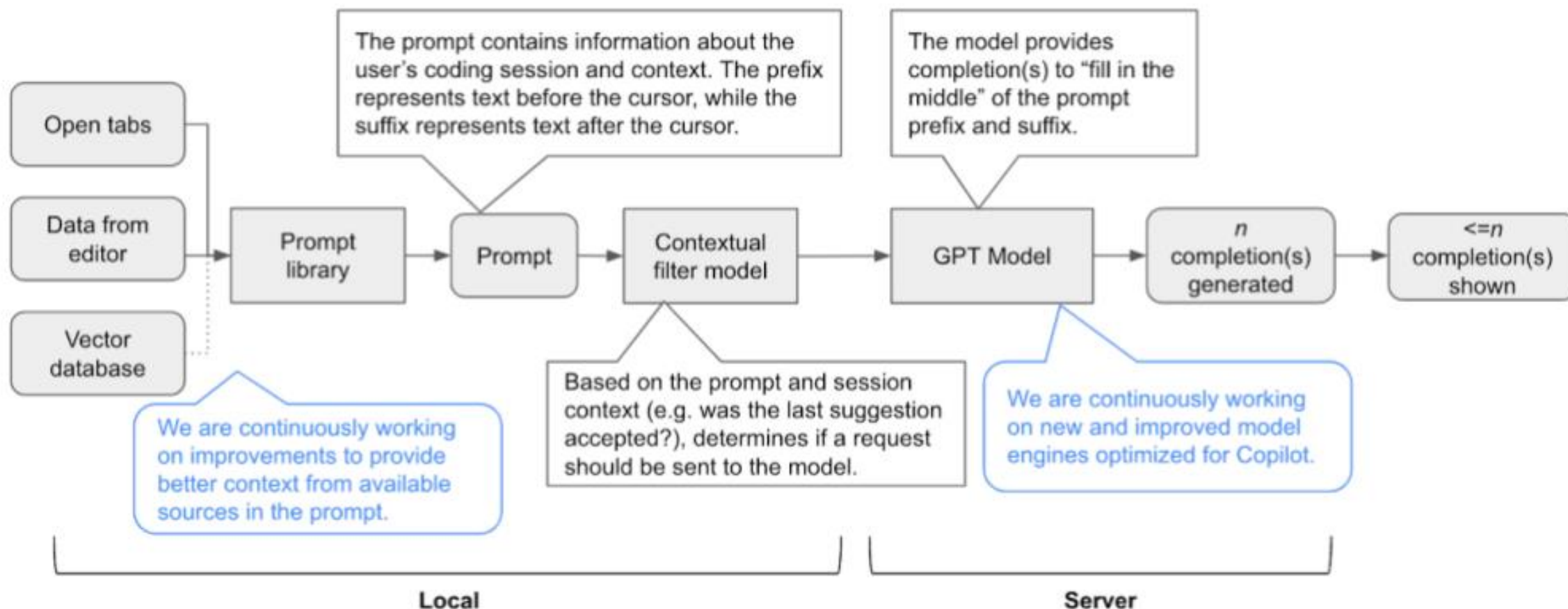


开发者引导方向，Copilot帮忙拉车

GitHub Copilot 原理说明



GitHub Copilot 工作原理细解



Simplified system diagram focused on model quality efforts. Made by Alice Li, machine learning researcher at GitHub.

GitHub Copilot 的环境配置

- 获得GitHub账号

- 需要参考您公司的GitHub Copilot的申请和开通流程，一般来说需要管理员发起邀请或者分配账号（管理员需要在GitHub Copilot配置页面，单独开启Chat和CLI功能，否则这两个功能无法被使用）
- 登录GitHub，在个人配置页面，验证自己的Copilot功能已经打开

- 在IDE中配置使用

- 下载安装对应IDE的GitHub Copilot插件，需要注意VS Code里Copilot和Chat是不同的IDE
- 支持的IDE列表：Visual Studio, Visual Studio Code, Neovim, JetBrains家族（包含IntelliJ IDEA, Android Studio, AppCode, CLion, Code With Me Guest, DataGrip, DataSpell, GoLand, JetBrains Client, MPS, PhpStorm, PyCharm, Rider, RubyMine, WebStorm）
- 在IDE里登录和激活GitHub Copilot
- 配置GitHub Copilot的代理（可选），参考文档[地址](#)，不同的IDE有不同的配置方式

- 在CLI中配置使用

- 需要单独下载Github Cli安装介质，然后做登录操作。参考[地址](#)

GitHub Copilot在IDE中无法使用的常见问题

- 网络问题导致登录失败或使用时无响应

- **症状：**安装了Copilot组件，也登录了（有时候登录就失败了），但是无法激活Copilot，或者Copilot已经激活，但是没有响应。同时，在IDE的输出窗口或者日志中，会显示网络连接故障。
- **解决方式：**切换到一个比较好的网络链路，然后重启IDE。

- 权限问题导致插件激活失败

- **症状：**账号可以正常登录GitHub，IDE中也做了登录，但是无法激活Copilot插件。在IDE的输出窗口可以看到权限相关的问题。
- **解决方式：**登录GitHub确认自己的Copilot功能是否启用，和管理员确认账号的权限情况。

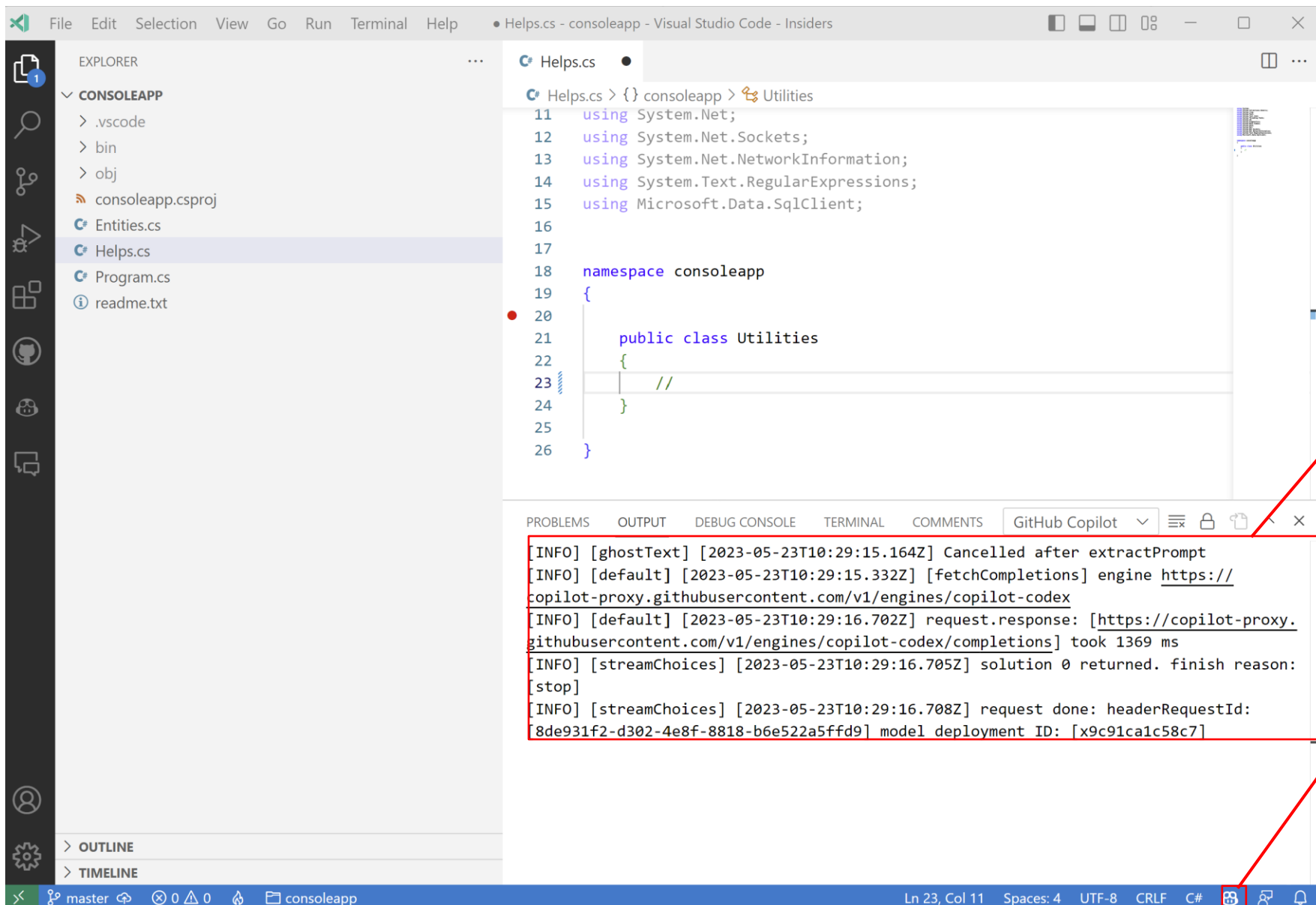
- 版本兼容造成使用时插件行为异常

- **症状：**网络、权限都做了确认没有问题，但是Copilot的行为不正常，某些功能和其他人不同。
- **解决方式：**在官网确认IDE和插件版本是否都是最新，有必要时请升级版本

- 其他问题

- 有缓存问题登录失败的先例，这时候请尝试重启ID
- 网络自查工具参考： curl --verbose https://copilot-proxy.githubusercontent.com/_ping
- 其他问题自查解决：[地址](#)

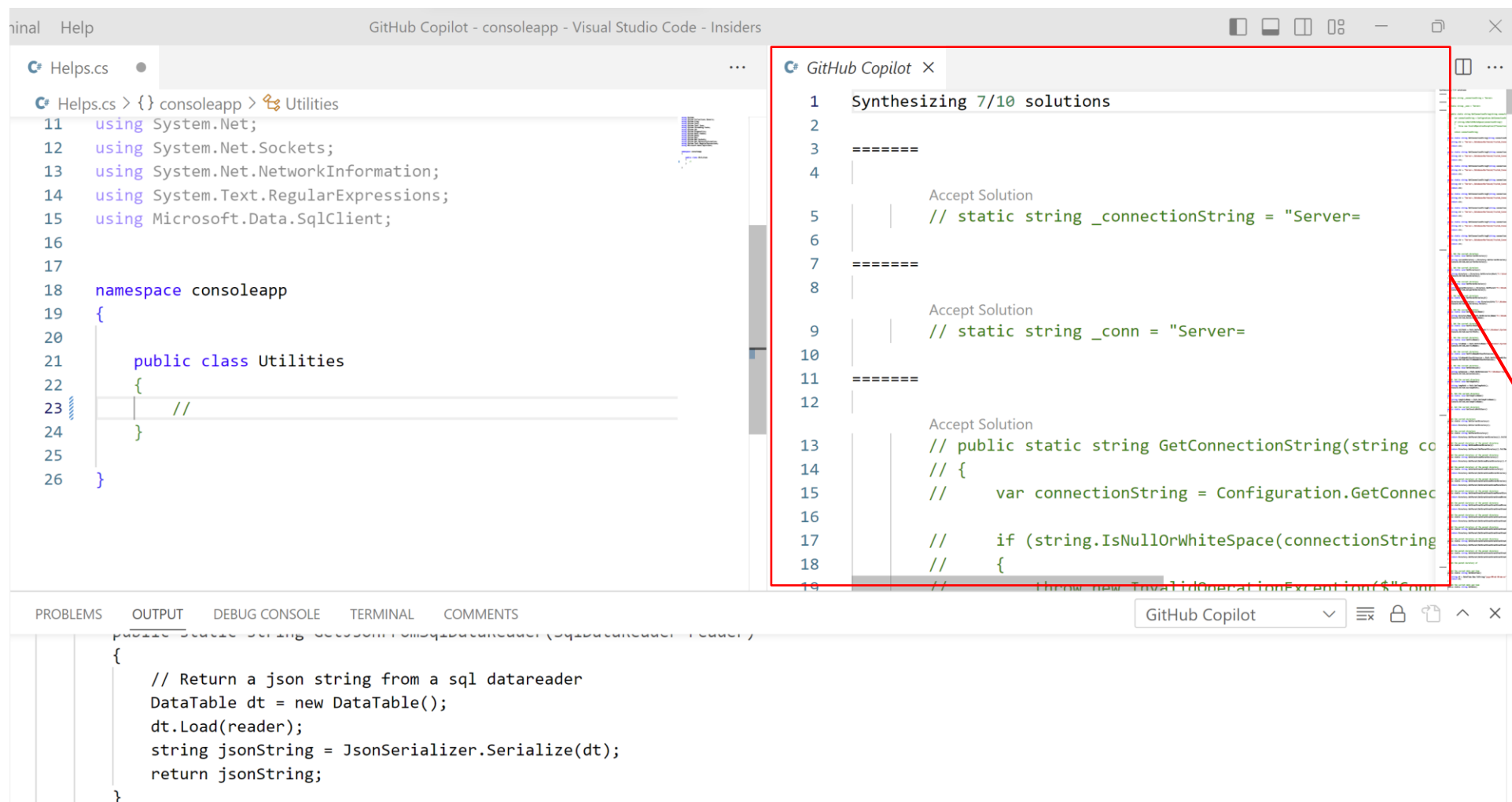
已经激活GitHub Copilot的IDE窗口



Copilot的log信息会输出到这里

有这个图像表示Copilot已经激活

已经激活GitHub Copilot的IDE窗口



GitHub Copilot快捷键汇总 (Windows)

快捷键	功能
Tab	接受当前建议
Esc	取消当前行的建议
Alt +]	展示下一个建议
Alt + [展示上一条建议
Alt + \	触发当前行的建议
Ctl + Enter	打开GitHub Copilot建议面板
Ctl + Shift + P	GitHub Copilot 指令命令选择
Ctl + Shift + i	向GitHub Copilot Chat提问

GitHub Copilot的基本能力和场景概览

• 核心能力:

- 在光标位置, 根据上下文**猜想**Developer后续要输入什么
- 上下文 (提示词) 可以是
 - 代码注释、一段逻辑代码的开头、一个字符/变量/语句等
- 生成的内容可以是
 - 完整的函数/功能代码
 - 后续的代码片段, 可以是半行、一行、多行代码
 - 注释
 - ~~整个代码文件~~
 - ~~整个工程文件~~

适用场景:

- 结合上下文给出编码
 - 生成函数 (查询式, 命令式), 包括单元测试
 - 补齐代码, DevOps脚本, Shell脚本, SQL语句等
 - 补齐和上下文匹配的有规律的相似格式的代码
 - 正则表达式
 - 简化陌生框架/语言的使用
 - 按常识完善对象字段
- 编写测试代码
 - 包含各类边界条件验证、使用难以记忆关键字编写代码等
- 注释补齐

```
57     public function getStrings(){
58         return [
59             'en' => [
60                 'answer' => 'Answer',
61                 'question' => 'Question',
62                 'date' => 'Date',
63             ], You, Moments ago • Uncommitted changes
64
65
66         ];
67     }
68 }
```

JS index.js

```
1 //dictionary of 2 letter ISO country codes and their corresponding country
```

GitHub Copilot 使用小技巧

- 打开和当前任务相关的代码文件（大约会提升5%-10%的代码接受率）
 - 如果相关的类库没有源码，可以打开元数据信息（metadata）
 - 如果没有元数据信息，当前编辑的文件里包含了相关类库的使用方式也可以
- 保持良好的代码习惯，比如保持比较好的代码命名规范，好的命名可以让Copilot更准确的理解上下文
- 多使用快捷键Ctrl+Enter
- 不要追求每次都可以生成大段的代码，生成一行或者半行其实也很能节省时间
- 可以适当的调整注释，使其清晰、具体、详细，没有模糊的描述
- **不要花费过多时间**在如何写代码注释上，Copilot提示的效果不好的时候，可以忽略当前提示的内容，自己写一段代码，之后就会就好了
- Copilot的定位是个更好用的编程工具，而不是代替开发者思考
- 使用原则：观察，思考，接受，验证，修改

GitHub Copilot 常见的使用问题

- 推荐代码很少，甚至只有一个起始大括号；或者推荐的代码莫名其妙，怎么办？
 - 使用快捷键打开GitHub Copilot Panel，查看更多的推荐信息
 - 不采取它推荐的内容，继续按您的常规的思路去写代码，写几行之后，Copilot会根据您的输入，逐渐改善推荐的内容，并逐渐达到比较好的推荐效果
 - 增加更多背景信息（可以是注释，可以是代码片段）
- Copilot好像没有响应了，不再提示任何代码，怎么办？
 - 检查网络连接是否正常，有必要的时候通过VPN or Proxy连接到GitHub Copilot
 - 重启IDE
 - 检查Copilot控制台的输出，或者查看log信息，查看错误信息，有必要的时候，提交case
- 我写了不少注释，但是Copilot推荐的内容好像不准，比ChatGPT差好多，怎么办？
 - Copilot的功能不是根据注释生成代码，是根据上下文自动构建提示词，然后生成推荐代码。您写的注释不一定会被自动生成提示词的逻辑认为是高权重的提示词。
 - 正确的使用方式是，把Copilot当成辅助，即，我们正常按之前的方式写代码，不用刻意写更多的注释，Copilot会自动根据我们的注释和已经完成的代码做猜想。如果猜对了，我们就接受，如果不对，就忽略。
 - 如果希望可以借助Copilot来实现大段代码生成，推荐使用GitHub Copilot Chat

GitHub Copilot Chat 场景总结

- 代码说明：深入分析和说明代码块的工作原理
- 代码帮助：获取编程相关问题的解答，可以是自然语言格式，也可以是代码片段格式
- 代码优化：获取特定于上下文的代码提议和建议
- 单元测试：生成完整的单元测试
- 代码评审：优化代码性能，提出改进意见
- 代码调试：查找问题并获取对 bug 和 异常的建议修复
- 漏洞发现和修复
- 新知识学习
- 技术方向和思路讨论
- 项目内的代码内容问答

GitHub Copilot Chat 使用小技巧

- 了解Chat的能力边界，Chat不是万能的
- 熟练使用Chat的快捷键和快捷指令
- 注意Copilot Chat的代码引用范围，默认为当前编辑文件中露出在屏幕区域的内容
- 提问时，需要对于所需的内容要尽可能具体、描述和详细，减少“蓬松”和不精确的描述
- 如果问题比较复杂，尝试使用“带有推理逻辑的引导词”，采用思维链的方式将模型引导向特定的思考方向
- 可以通过示例，清晰地表达所需的输出格式
- 不要只是说什么不要做，而是说什么要做
- 给AI助手一个身份
- 积累自己的提示词库

GitHub Copilot 和 Copilot Chat 总结

Copilot是写代码时的助手
(动手的时候)

- 定位是尽量不改变你的编程方式的前提下，带来革命性的代码编写体验
- 一个能看到你IDE中正在编辑的文件的AI助手，会持续的猜想你可能要在光标位置输入哪些内容
- 是根据光标位置的上下文来猜的，如果人可以猜出来的场景，它也可以做到类似的效果
- 不建议花过多时间告诉这个助手我们想做什么，它会自己猜
- 如果这个助手猜错了，也不建议花太多时间研究它为什么猜错了，只需要忽略它的猜测，继续正常的写代码即可，它会根据你的输入修正自己的猜想

Copilot Chat是结对编程的技术专家
(思考的时候)

- 在写代码时，遇到任何技术相关的疑问都可以问它
- 包括但不限于以下场景：
 - 代码优化、代码调试、Bug修复、测试案例编写、根据需求实现代码、技术思路讨论、技术概念解答等等
- 向这个虚拟专家提问，需要积累提示词技巧。具体的提示词参考附件部分

GitHub Copilot经常被问到的问题-1

- Copilot生成建议代码时，需要传递代码片段到后端，那代码安全如何保证？
 - 针对企业用户，GitHub不会存储任何用户的代码相关数据。这点也写在了GitHub的官方[声明文件](#)里。
 - GitHub Copilot为了避免代码版权纠纷，也不会使用客户的数据做模型训练和更新。
 - 自己搭建代理服务器，在代理服务器上开启流量日志功能，方便针对传输的数据做安全审计。
- 国内用户如何保证访问GitHub的速度？防火墙怎么做控制？
 - GitHub Copilot for Business版本支持在插件上HTTP配置代理（代理服务需要用户自己准备）
 - Copilot for Business的正常使用需要保证以下地址可以正常访问：[地址列表](#)
- Copilot是依据开源代码训练的，那生成的代码会和开源代码高度相似吗？
 - 可以在管理面板里禁止调和开源代码相似的推荐，以确保生成的代码和开源代码不会有重叠。
- Copilot生成的代码的知识产权归属于谁？
 - 归属于软件开发者，Copilot只是写代码的辅助工具
- Copilot 支持的语言有哪些？
 - 和LLM兼容多种语言一样，Copilot也兼容各种编程语言，已经验证过的超过了60种语言/脚本。

GitHub Copilot经常被问到的问题-2

- GitHub Copilot X和Copilot的区别及获取方式
 - GitHub Copilot X是一组在预览的产品的代称，正式发布时会从Copilot X的列表里移除或标记为发布
 - 目前Copilot X的很多组件已经有了确定的产品发布路线，这些组件也已经从X中移除出去，比如GitHub Labs, GitHub Chat等
 - 其他Copilot X需要用户自己提交排队申请：<https://githubnext.com/>，如果您在GitHub社区有开源的项目，并且更新很快则有比较高的几率被邀请试用Copilot X的功能
- GitHub Copilot 有接口可以调用做二次开发吗？
 - 没有
- GitHub Copilot有token数限制吗？如果达到限制怎么办？
 - 存在，但是具体token数没有材料披露，如果遭遇了token数的限制，可以再刷新一下重新试一次，即强制token数归零。
- 可以用GitHub Copilot帮我生成一个完整的项目吗？
 - GitHub Copilot目前的定位是编程的助手，不是ChatGPT的应用，也不是虚拟的程序员。整个项目开发过程依然需要程序员来完成整个软件项目的框架设计和细节实现，Copilot只是简化和加速了程序员的日常工作的每个环节。
 - GitHub Copilot Chat的workspace一定程度上可以创建一个项目的框架，但是目前这个功能还有很大提升空间。

一些使用技巧的链接

链接

<https://learn.microsoft.com/zh-cn/visualstudio/ide/visual-studio-github-copilot-chat?view=vs-2022>

<https://github.blog/2023-06-20-how-to-write-better-prompts-for-github-copilot/>

<https://dev.to/github/a-beginners-guide-to-prompt-engineering-with-github-copilot-3ibp>

<https://github.blog/2023-05-17-how-github-copilot-is-getting-better-at-understanding-your-code/>



提示词的七个原则

格式要求

控制回复数量

假装和限定角色获得场景

重新整理数据

限定内容

组合流水线

突破个人限制

示例：

- 生成xxx并用逗号分隔
- 生成列表
- 函数名需要以xxx_yyy格式命名

提示词的七个原则

格式要求

控制回复数量

假装和限定角色获得场景

重新整理数据

限定内容

组合流水线

突破个人限制

示例：

- 给出更多解释，600字
- 生成内容直到抵达上限
- 不多于两个段落/在一句话以内

提示词的七个原则

格式要求

控制回复数量

假装和限定角色获得场景

重新整理数据

限定内容

组合流水线

突破个人限制

示例：

- 假定你是一个数据库专家
- 假设你正在完成xxx任务
- 从现在开始，你就是...

提示词的七个原则

格式要求

控制回复数量

假装和限定角色获得场景

重新整理数据

限定内容

组合流水线

突破个人限制

示例：

- 重新生成代码，用...函数/类/接口
- 依据上面的代码，考虑xxx条件，生成...

提示词的七个原则

格式要求

控制回复数量

假装和限定角色获得场景

重新整理数据

限定内容

组合流水线

突破个人限制

示例：

- 基于给定的内容...
- 在xxxx（一个技术名词）的模块下， ...

提示词的七个原则

格式要求

控制回复数量

假装和限定角色获得场景

重新整理数据

限定内容

组合流水线

突破个人限制

示例：

- 生成一个列表
- 对于列表中的每个条目，生成...
- 继续
- 过滤...
- 重新生成输出条目为...
- 将最终内容总结为简短的版本

提示词的七个原则

格式要求

控制回复数量

假装和限定角色获得场景

重新整理数据

限定内容

组合流水线

突破个人限制

示例：

- 关于...有哪些相关概念和领域需要研究
- 生成和...相关的领域和概念
- 对于给定内容...提示我相关的
- 关于...由浅入深的推荐步骤是...