



.NET黄浦论坛  
— .NET HUANGPU FORUM —

2024.NET技术沙龙

# .Net开发者如何快速 适应AI大模型的变革

演讲嘉宾：许泽宇

微软最有价值专家(MVP)  
华为云开发者专家(HCDE)  
软通动力-数字化技术服务中心-首席架构师





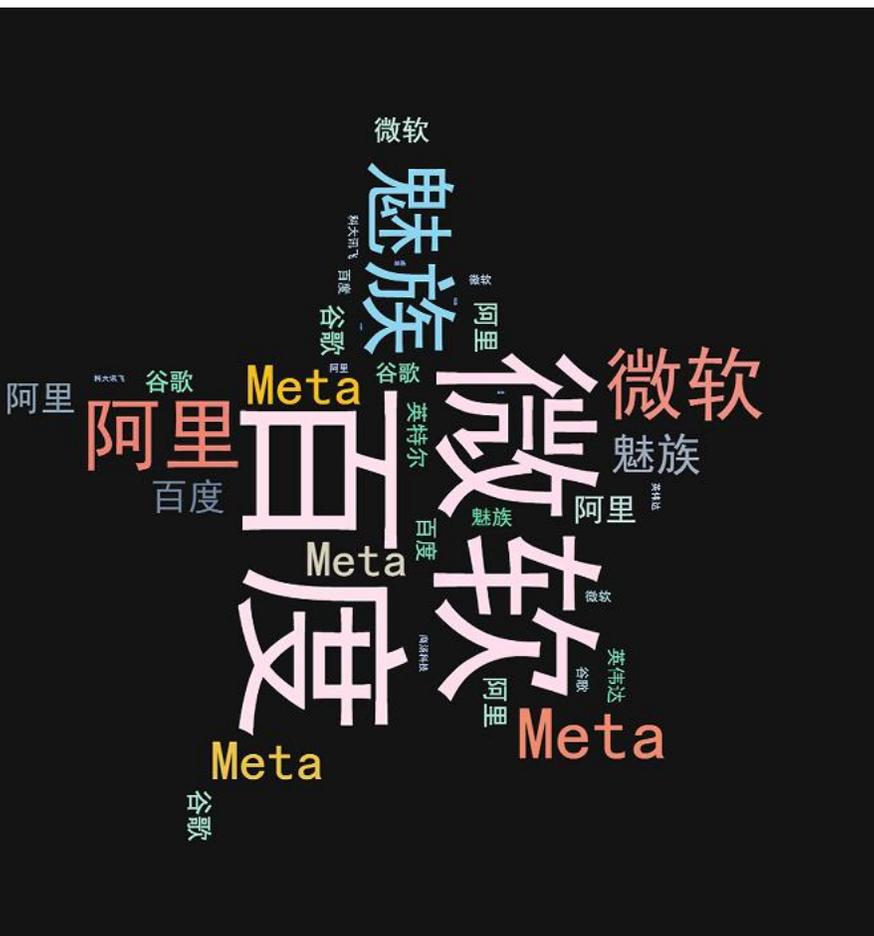
# 目录

## CONTENTS

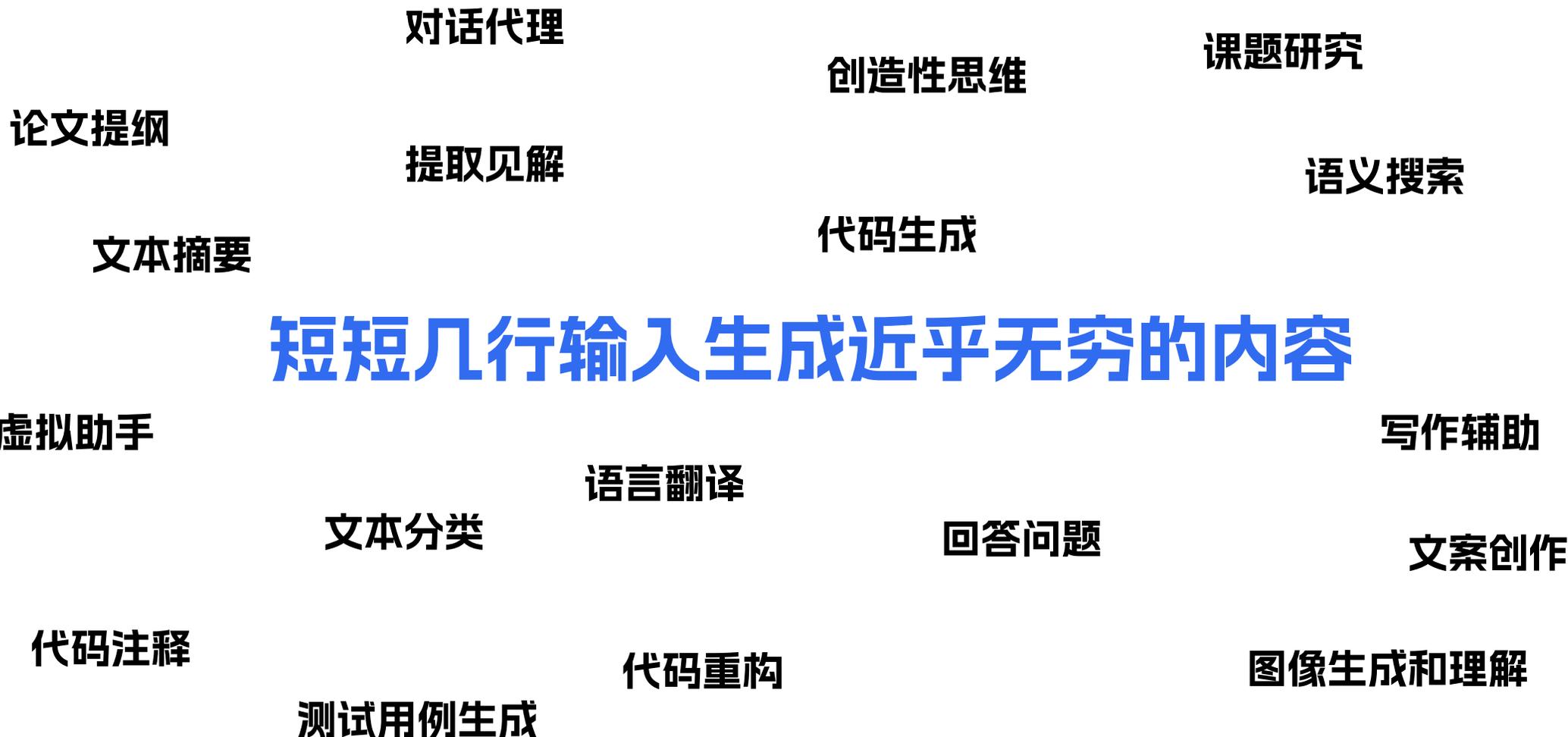
- 01 | 大模型现状
- 02 | .Net开发者学习路径
- 03 | 总结

## 大厂纷纷 All In AI

各大科技公司近年来纷纷提出“**All In AI**”的策略，将人工智能作为其核心发展方向之一。



- **微软 (Microsoft):** 大力发展Azure AI，推出智能助手和开发工具。
- **魅族 (Meizu):** 智能手机中集成了AI功能，如AI拍照和语音助手。
- **阿里云 (Alibaba Cloud):** 投资AI技术，用于语音识别和智能搜索等领域。
- **百度 (Baidu):** 专注在百度大脑和自动驾驶技术Apollo上。
- **英特尔 (Intel):** 推动AI芯片和边缘计算，通过收购扩展AI实力。
- **英伟达 (NVIDIA):** 领先的AI硬件提供商，GPU广泛用于AI训练。
- **科大讯飞 (iFlyTek):** 专注语音技术，广泛应用于教育和客服等领域。
- **商汤科技 (SenseTime):** 计算机视觉专家，尤其在人脸识别和视频分析方面。
- **谷歌 (Google):** AI研究先锋，TensorFlow平台广泛使用。
- **Meta (Facebook):** AI用于内容推荐和广告，并发展AR/VR技术。



# 我应该如何开始？

## 国外

ChatGPT  
Claude

## 国内

豆包  
智谱轻言  
讯飞星火  
文心一言  
Kimi智能助手

## 开源

Open WebUI  
LobeChat  
NextChat

- 许多日常工作任务，无论是撰写文档、测试用例，还是架构设计，代码审查及优化，这些都可以交给大模型来处理。
- 通过与AI频繁互动，我们可以提高工作效率、激发创新思维，并在复杂问题上获得支持。
- 将AI视为伙伴，不仅能减轻负担，更能释放更多时间，让我们专注于更具价值的工作。

有很多开发者在短期体验后说AI比较弱zhi，有没有可能是你的问题？  
开发者有必要学习提示词工程：<https://www.promptingguide.ai/zh>

• Prompt Engineering Guide

提示工程课程 服务 关于 Search...

CTRL K



提示工程指南

提示工程简介

大语言模型设置

基本概念

提示词要素

设计提示的通用技巧

提示词示例

提示技术

零样本提示

少样本提示

链式思考 (CoT) 提示

自我一致性

生成知识提示

Prompt Chaining

思维树 (ToT)

检索增强生成 (RAG)

自动推理并使用工具 (ART)

自动提示工程师

Active-Prompt

方向性刺激提示

提示工程指南

## 提示工程指南

提示工程 (Prompt Engineering) 是一门较新的学科，关注提示词开发和优化，帮助用户将大语言模型 (Large Language Model, LLM) 用于各场景和研究领域。掌握了提示工程相关技能将有助于用户更好地了解大型语言模型的能力和局限性。

研究人员可利用提示工程来提升大语言模型处理复杂任务场景的能力，如问答和算术推理能力。开发人员可通过提示工程设计、研发强大的工程技术，实现和大语言模型或其他生态工具的高效接轨。

提示工程不仅仅是关于设计和研发提示词。它包含了与大语言模型交互和研发的各种技能和技术。提示工程在实现和大语言模型交互、对接，以及理解大语言模型能力方面都起着重要作用。用户可以通过提示工程来提高大语言模型的安全性，也可以赋能大语言模型，比如借助专业领域知识和外部工具来增强大语言模型能力。

基于对大语言模型的浓厚兴趣，我们编写了这份全新的提示工程指南，介绍了大语言模型相关的论文研究、学习指南、模型、讲座、参考资料、大语言模型能力以及其他与提示工程相关的工具。

 We are excited to launch two new prompt engineering courses. Get access by joining our DAIR.AI Academy. [Join now!](#)

Use code PROMPTING20 to get an extra 20% off.

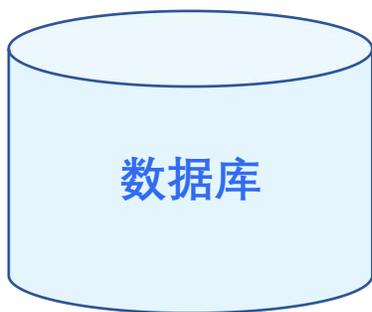
IMPORTANT: The discount is limited to the first 500 students.

Question? Give us feedback →

Edit this page

Last updated on 2024年9月19日

学习数据库从sql语句开始



学习大模型开发从OpenAI API开始

大模型开发



<https://platform.openai.com/docs/api-reference/chat/create>

示例请求

GPT-4O ▾ 卷曲 ▾



```
1 curl https://api.openai.com/v1/chat/completions \  
2   -H "Content-Type: application/json" \  
3   -H "Authorization: Bearer $OPENAI_API_KEY" \  
4   -d '{  
5     "model": "gpt-4o",  
6     "messages": [  
7       {  
8         "role": "system",  
9         "content": "You are a helpful assistant."  
10      },  
11      {  
12        "role": "user",  
13        "content": "Hello!"  
14      }  
15    ]  
16  }'
```

回复



```
1 {  
2   "id": "chatcmpl-123",  
3   "object": "chat.completion",  
4   "created": 1677652288,  
5   "model": "gpt-4o-mini",  
6   "system_fingerprint": "fp_44709d6fcb",  
7   "choices": [{  
8     "index": 0,  
9     "message": {  
10      "role": "assistant",  
11      "content": "\n\nHello there, how may I assist you  
12    },  
13    "logprobs": null,  
14    "finish_reason": "stop"  
15  }],  
16  "usage": {  
17    "prompt_tokens": 9,  
18    "completion_tokens": 12,  
19    "total_tokens": 21,  
20    "completion_tokens_details": {  
21      "reasoning_tokens": 0  
22    }  
23  }  
24 }
```

<https://platform.openai.com/docs/api-reference/embeddings>

示例请求

卷曲 ✓ 复制

```
1 curl https://api.openai.com/v1/embeddings \  
2   -H "Authorization: Bearer $OPENAI_API_KEY" \  
3   -H "Content-Type: application/json" \  
4   -d '{  
5     "input": "The food was delicious and the waiter..."  
6     "model": "text-embedding-ada-002",  
7     "encoding_format": "float"  
8   }'
```

回复

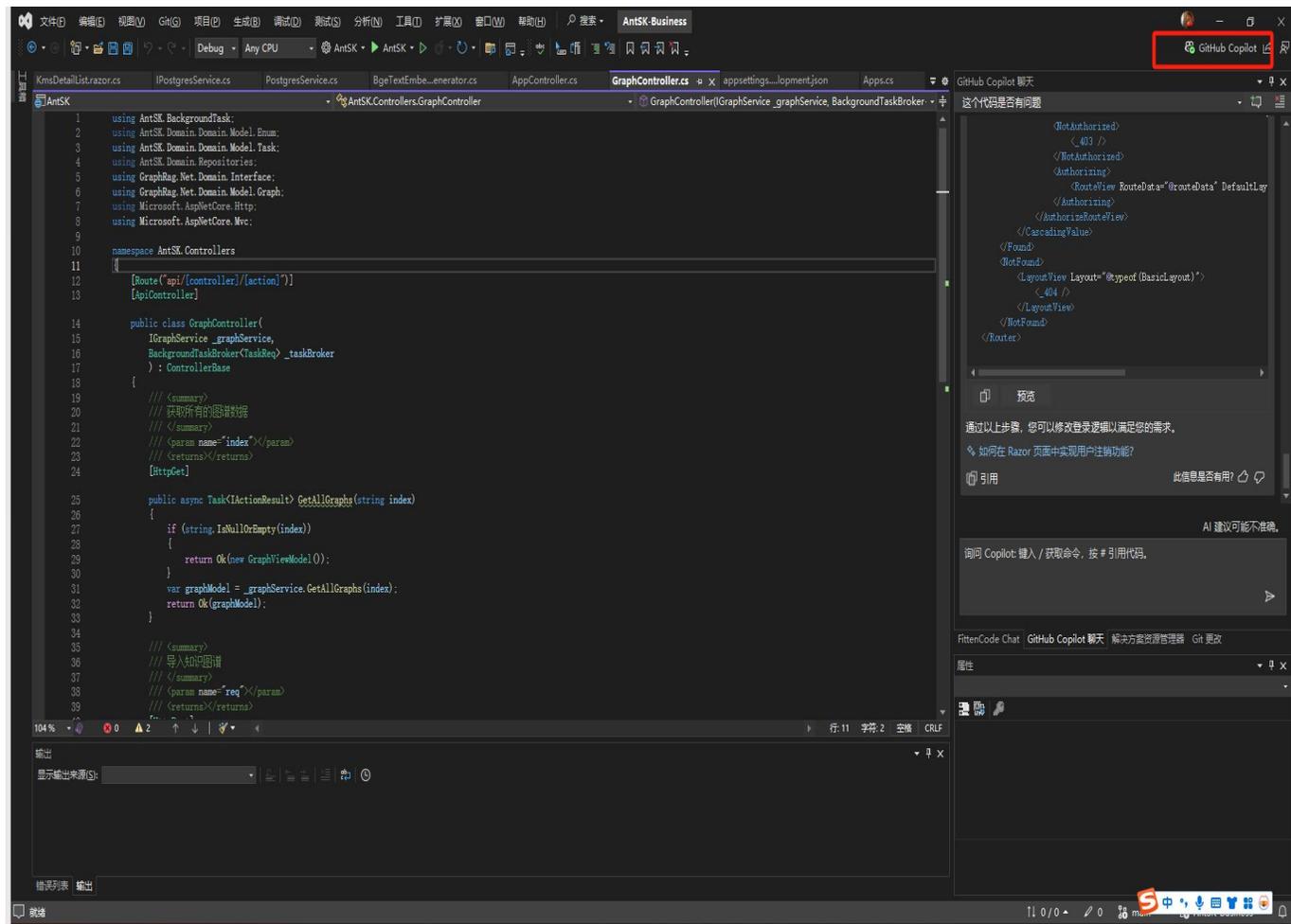


```
1  {  
2    "object": "list",  
3    "data": [  
4      {  
5        "object": "embedding",  
6        "embedding": [  
7          0.0023064255,  
8          -0.009327292,  
9          .... (1536 floats total for ada-002)  
10         -0.0028842222,  
11        ],  
12        "index": 0  
13      }  
14    ],  
15    "model": "text-embedding-ada-002",  
16    "usage": {  
17      "prompt_tokens": 8,  
18      "total_tokens": 8  
19    }  
20 }
```

<https://platform.openai.com/docs/api-reference/introduction>

模块	名称	使用场景
函数调用	Function Call	函数调用，对接业务系统API
微调	Fine-tuning	微调
批任务	Batch	批量跑，价格便宜，非及时(24小时内)
文件	File	上传、检索
图片	Image	图片生成，Dall-E-3
助理	Assistants	Agent

## 开发者引导方向，Copilot执行繁琐编码

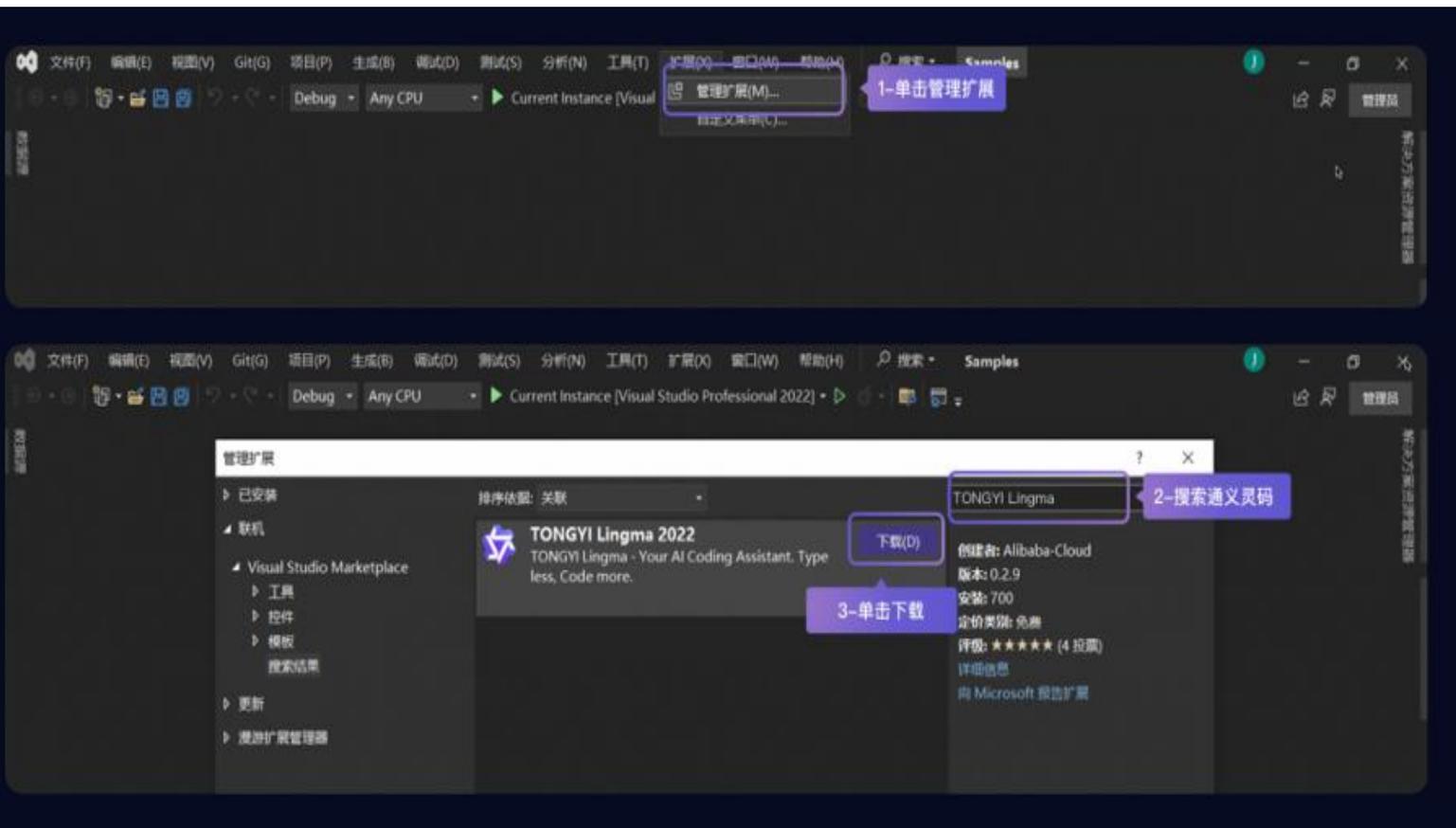


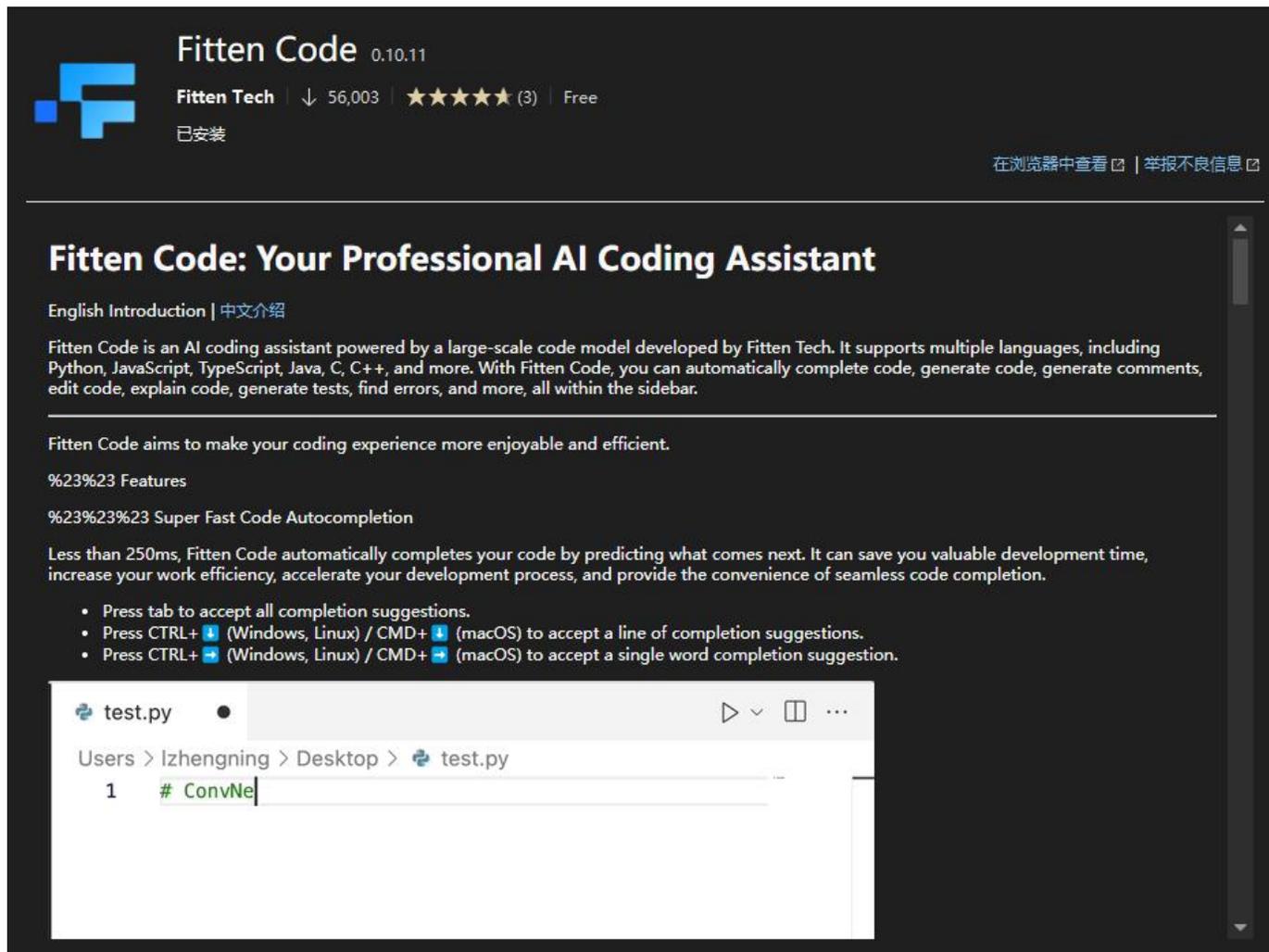
## Visual Studio在新版本内置了Copilot

- 可以减少切换到浏览器来搜索解决方案。
- 与IDE提供的代码补全相比，输入更少，代码完补全能更智能。
- Copilot更多用于自动化编写代码，实现具体功能。
- 开发者可以更专注与更大的图景，架构设计和系统集成。
- 开发者和Copilot 越用互相越默契，提示越来越适合Copilot，而Copilot的建议越来越像你自已写的代码。
- 整体体验不错，国内使用速度稍慢

## Visual Studio使用扩展安装

- 个人使用免费
- 有Chat功能
- 速度比较快





The screenshot shows the Visual Studio Marketplace page for the Fitten Code extension. At the top left is the Fitten Code logo, a blue square with a white 'F'. To its right, the text reads 'Fitten Code 0.10.11'. Below this, it says 'Fitten Tech' with a download count of '56,003', a star rating of '★★★★★ (3)', and 'Free'. A status indicator '已安装' (Installed) is shown. On the right side of the header, there are links for '在浏览器中查看' (View in browser) and '举报不良信息' (Report bad information).

## Fitten Code: Your Professional AI Coding Assistant

English Introduction | [中文介绍](#)

Fitten Code is an AI coding assistant powered by a large-scale code model developed by Fitten Tech. It supports multiple languages, including Python, JavaScript, TypeScript, Java, C, C++, and more. With Fitten Code, you can automatically complete code, generate code, generate comments, edit code, explain code, generate tests, find errors, and more, all within the sidebar.

Fitten Code aims to make your coding experience more enjoyable and efficient.

### Features

#### Super Fast Code Autocompletion

Less than 250ms, Fitten Code automatically completes your code by predicting what comes next. It can save you valuable development time, increase your work efficiency, accelerate your development process, and provide the convenience of seamless code completion.

- Press tab to accept all completion suggestions.
- Press CTRL+⬇️ (Windows, Linux) / CMD+⬇️ (macOS) to accept a line of completion suggestions.
- Press CTRL+⬅️ (Windows, Linux) / CMD+⬅️ (macOS) to accept a single word completion suggestion.

```
test.py
Users > lzhengning > Desktop > test.py
1 # ConvNe
```

## Visual Studio使用扩展安装

- 个人使用免费
- 有Chat功能
- 速度非常快
- 逻辑性不如Github Copilot

<https://github.com/openai/openai-dotnet>

## OpenAI .NET API library

nuget v2.1.0-beta.1

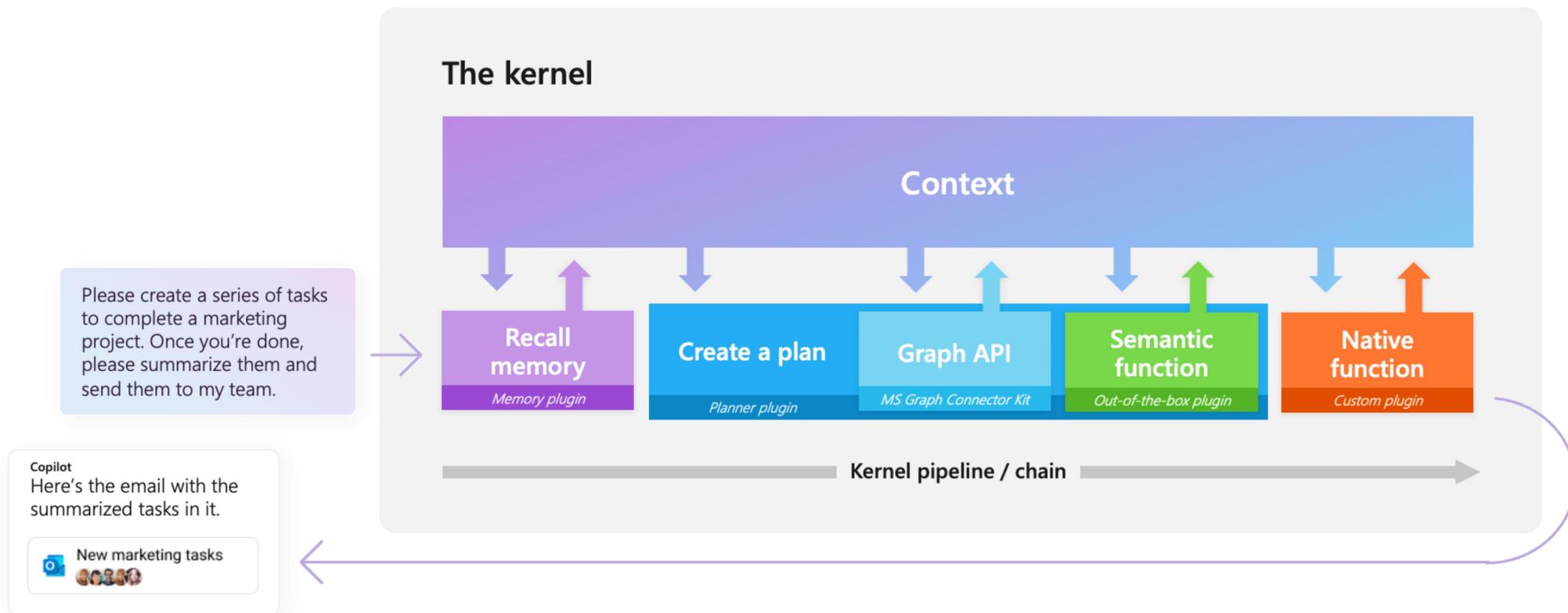
The OpenAI .NET library provides convenient access to the OpenAI REST API from .NET applications.

It is generated from our [OpenAPI specification](#) in collaboration with Microsoft.

### Table of Contents

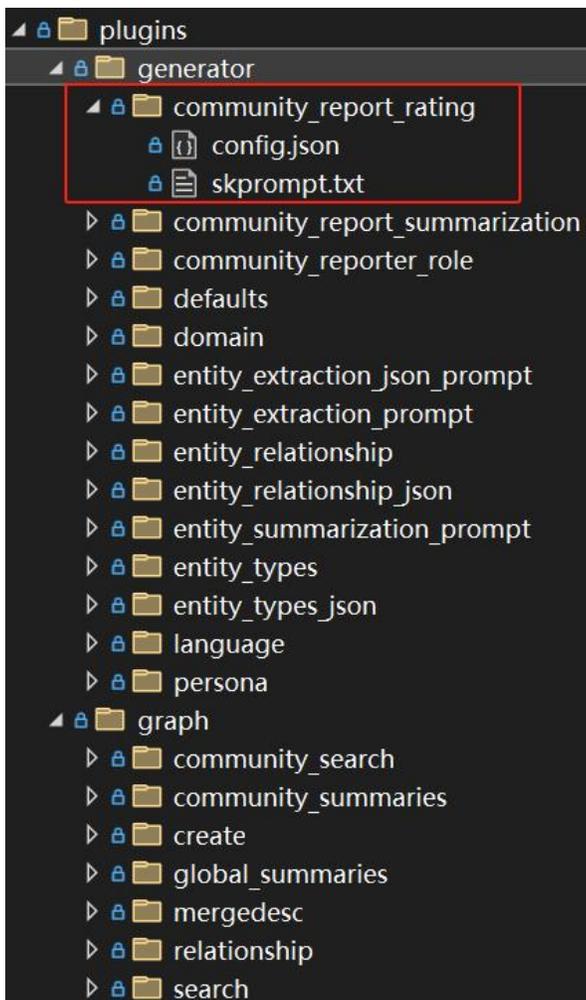
- [Getting started](#)
  - [Prerequisites](#)
  - [Install the NuGet package](#)
- [Using the client library](#)
  - [Namespace organization](#)
  - [Using the async API](#)
  - [Using the `openAIClient` class](#)
- [How to use chat completions with streaming](#)
- [How to use chat completions with tools and function calling](#)
- [How to use chat completions with structured outputs](#)
- [How to generate text embeddings](#)
- [How to generate images](#)
- [How to transcribe audio](#)
- [How to use assistants with retrieval augmented generation \(RAG\)](#)
- [How to use assistants with streaming and vision](#)
- [How to work with Azure OpenAI](#)
- [Advanced scenarios](#)
  - [Using protocol methods](#)
  - [Mock a client for testing](#)
  - [Automatically retrying errors](#)
  - [Observability](#)

- **OpenAI官方的.Net 客户端**
- **通过nuget包使用**
- **比较轻量，填写key即可使用**
- **包含常用功能，比如聊天、嵌入、图像**
- **支持Azure OpenAI的集成**
- **高级场景，如协议方法的使用、模拟客户端进行测试、自动重试错误和可观测性**



**Semantic Kernel是一个集成了大型语言模型（LLM）的SDK，如 OpenAI、Azure OpenAI、以及 Hugging Face 和 C#、Python 和 Java 等传统编程语言。Semantic Kernel 实现了这一点 通过允许您定义可以链接在一起的插件 只需几行代码。**

## 当出现散落到各个代码中的提示词，你该如何管理？



- **Semantic Function 将提示词进行了统一管理，提示词不在杂乱无章，散落到代码的各个地方**
- **Semantic Function 允许Function嵌套，并支持提示词模板。**
- **Semantic Function 支持每个Function的独立参数管理。**

```
[KernelFunction, Description("发送邮件")]  
0 个引用  
public string SendEmail(string input)  
{  
    //实际业务处理逻辑  
    Console.WriteLine(input);  
    return "发送成功";  
}
```

- 可以将Semantic Function与Native Function进行混合编排。例如大模型无法帮你发送邮件，但是通过Native Function可以让大模型具备发送邮件的能力。
- 调用方式的通用，无论你调用Semantic Function还是NativeFunction，调用方式都是保持一致。

## 当老板说，要把Qwen换成Baichuan，你该怎么办？

- 提供给了标准的接口进行扩展 `ITextGenerationService`、`IChatCompletionService`
- 可以在接口中来实现具体对接的模型，从而不影响业务代码的变更

```
builder.Services.AddKeyedSingleton<ITextGenerationService>("spark-desk-text", new SparkDeskTextCompletion());  
builder.Services.AddKeyedSingleton<IChatCompletionService>("spark-desk-chat", new SparkDeskChatCompletion());
```

通过 `ITextEmbeddingGenerationService` 可以进行向量接口的扩展

```
builder.Services.AddSingleton((ITextEmbeddingGenerationService)textEmbeddingGenerator);  
builder.Services.AddKeyedSingleton("bge-embedding", textEmbeddingGenerator);
```

```
Microsoft.SemanticKernel.Experimental.Orchestration.Flow flow = FlowSerializer.DeserializeFromYaml(@"
goal: answer question and sent email
steps:
  - goal: What is the tallest mountain in Asia? How tall is it divided by 2?
    plugins:
      - WebSearchEnginePlugin
    provides:
      - answer
  - goal: Collect email address
    plugins:
      - CollectEmailPlugin
    provides:
      - email_address
  - goal: Send email
    plugins:
      - SendEmailPlugin
    requires:
      - email_address
      - answer
    provides:
      - email
):

var flowOrchestrator = new FlowOrchestrator(
    builder,
    await FlowStatusProvider.ConnectAsync(new VolatileMemoryStore()),
    plugins,
    config: new FlowOrchestratorConfig() { MaxStepIterations = 20 });

// Act
var result = await flowOrchestrator.ExecuteFlowAsync(flow, sessionId, "What is the tallest mountain in Asia? How tall is it divided by 2?");
```

- 支持通过Json或Yaml来配置对话流
- 支持Function Call的插件调用
- 可以设置多个Step来引导用户完成Goal

Kernel Memory (KM) 是一种多模态 AI 服务，专门用于通过自定义连续数据对数据集进行高效索引 混合管道，支持检索增强生成 (RAG)，合成记忆、提示工程和自定义语义记忆处理。



利用高级嵌入和 LLM，该系统支持自然语言 查询以从索引数据中获取答案，并附上引文 以及指向原始来源的链接。



## 文档导入

`memory.ImportDocumentAsync`

`memory.ImportWebPageAsync`

`memory.ImportTextAsync`

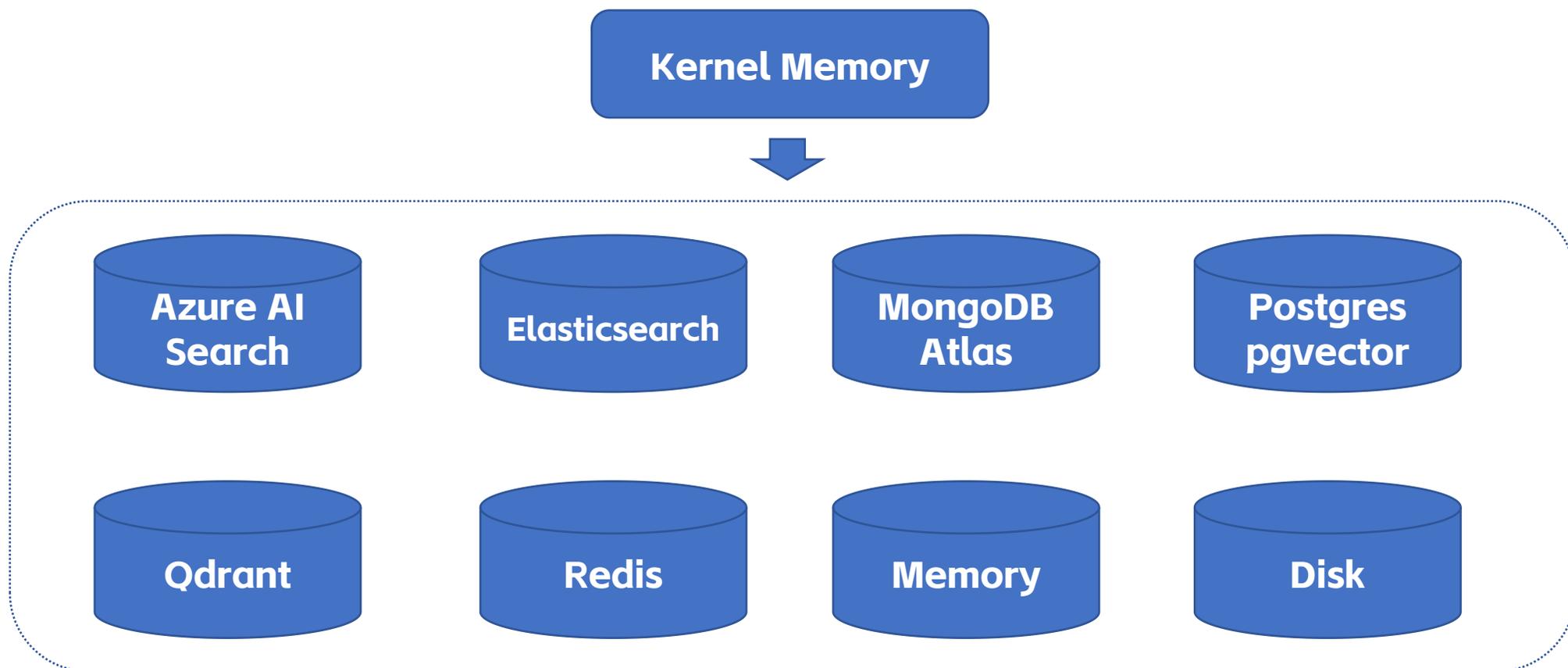
开箱即用的导入方式、文件、网页、文本

## 文档查询

memory.**SearchAsync**  
memory.**AskAsync**

开箱即用文档检索与文档问答

Kernel Memory 已经帮我们封装好了调用大模型、向量存储等复杂繁琐的操作，提供了近乎开箱即用的RAG方案，我们可以导入多种类型的文档、并对文档进行检索和查询问答。并且支持了非常多种类的不同向量数据库



# 当默认提供的实现，不能满足需求时我应该怎么办？

- 我想要把PDF里的图片提取成markdown链接
- 我想要让大模型整理好导入的问答对
- 我想要导入前做文档校验
- .....

<https://github.com/microsoft/autogen>

<https://github.com/AIDotNet/EasyAgent>

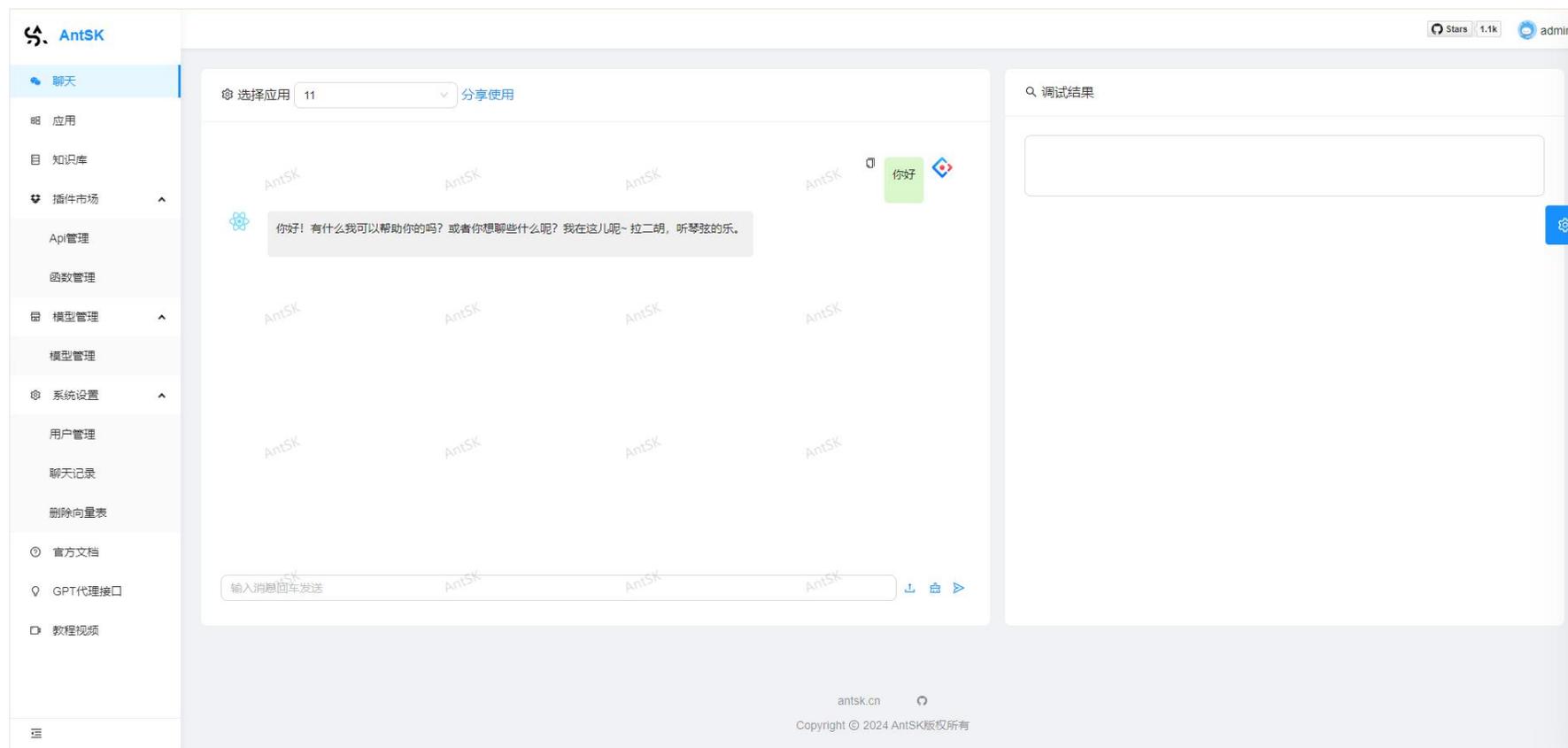
The screenshot displays the EasyAgent web interface. On the left, there is a sidebar with navigation options: '对话' (Chat), 'Agent配置' (Agent Configuration), and '模型设置' (Model Settings). The main chat area shows a conversation between two agents, '刘备' (Liu Bei) and '张飞' (Zhang Fei). The chat history includes:

- 刘备: 大家好, 我是刘备, 德才兼备的汉室皇叔, 三国时期蜀汉的开国皇帝。在这个群聊中, 我们可以一起探讨如何恢复汉室的光辉, 或者是其他关于三国时期的话题。希望我们能够互相学习, 共同进步。有什么问题或者想法, 尽管提出来, 我们一起讨论。
- 张飞: 哼, 我张飞在此! 兄长, 你总是这么温文尔雅, 但别忘了我们现在正处在乱世之中, 光说不练可是要吃亏的。我们得多练兵马, 才能在这三国争霸中立足。我把丈八蛇矛可是随时准备上阵, 替咱们蜀汉立功! 兄弟们, 有什么打仗的计划尽管告诉我, 我定当第一个冲锋陷阵!
- 刘备: 黄忠, 字汉升, 是三国时期蜀汉的五虎上将之一, 以勇猛和箭术高超著称。在历史上, 黄忠在刘备手下成为一名杰出的将领, 尤其是在汉中之战中表现突出, 击败了曹操的名将夏侯渊, 为刘备夺取汉中立下了汗马功劳。在我们这个虚拟的群聊中, 如果要继续角色扮演, 可以继续假设黄忠正在进行日常的训练, 或者参与筹划未来的军事行动。他可能在磨练自己的箭术, 或者与其他将领一起讨论战术和战略。不过, 请记住, 这些都是基于历史和小说《三国演义》的虚构情景。
- 张飞: 嘿, 黄汉升那老将军可是个硬骨头, 老当益壮的典范。他最近应该在练兵, 准备着随时出战。黄忠的箭术一流, 咱们蜀汉的弓箭手都得向他学习。听说他老人家最近还在研究新的箭术呢, 准备下次遇敌时给他们一个大大的惊喜。咱们也得加强训练, 别被老黄甩得太远了!

At the bottom of the chat area, there is an input field with the placeholder text '输入消息回车发送' (Enter message and press Enter to send). The footer of the interface includes the text 'EasyAgent Copyright © 2024 许泽宇的技术分享'.

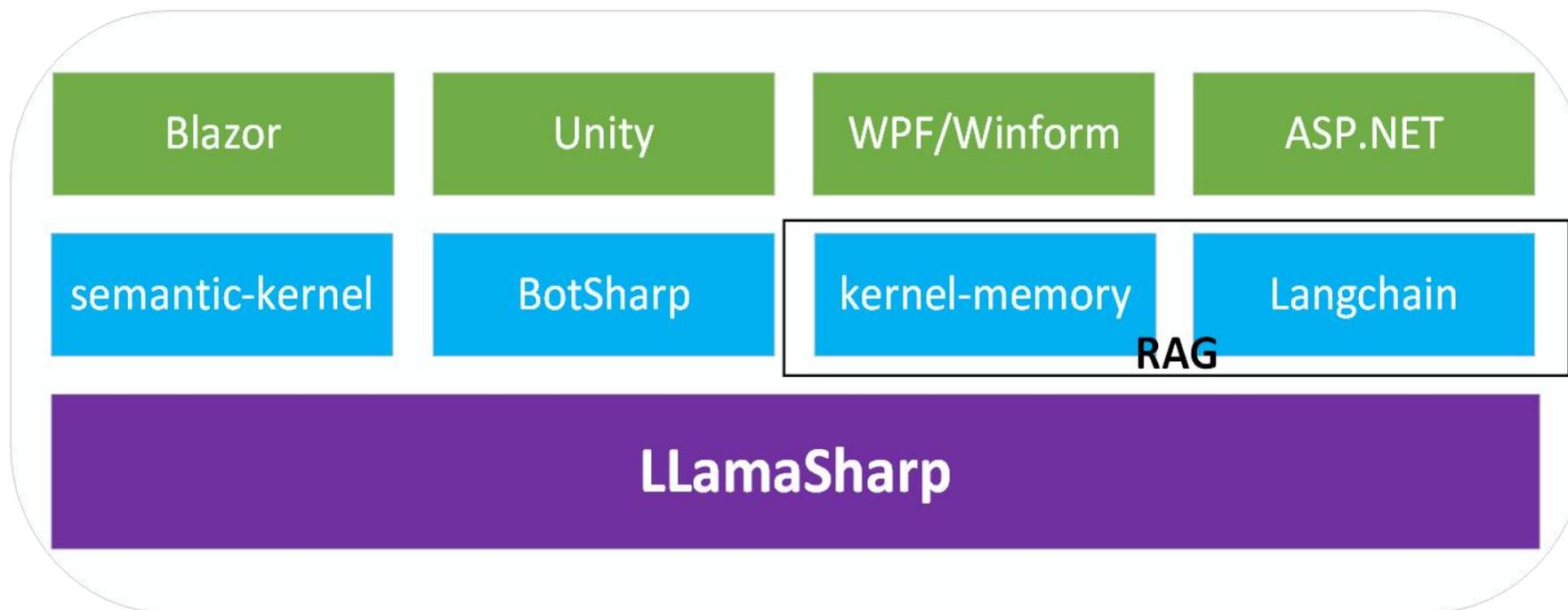
- 可以定义多角色的智能体
- 可以支持函数调用
- 可以定义管理员
- 可以用户参与多角色对话

- 基于.Net8+AntBlazor+SemanticKernel 和KernelMemory 打造的AI知识库，支持本地离线AI大模型。支持离线模型运行。支持aspire观测应用数据。
- <https://github.com/AIDotNet/AntSK>



LLamaSharp是一个跨平台的，可以在本地运行LLaMA/LLaVA的库，基于llama.cpp。同时支持CPU/GPU推理

<https://github.com/SciSharp/LLamaSharp>



<https://github.com/AIDotNet>



AIDotNet

AIDotNet 是一群热爱DotNet的开源爱好者组成, 为了让DotNet赋能AI, 让AI轻松带入您的产品, 我们成立了AIDotNet组织。

159 followers China <https://ai-dotnet.com> <https://antsk.cn> [hejiale010426@gmail.com](mailto:hejiale010426@gmail.com)

README .md

## AIDotNet 开源组织

### 介绍

我们是一群热爱AI和DotNet的开发者, 组合在一起一块打造更智能的AI智能体, 并且项目大部分采用Apache License 2.0, 您可以任意商用, 您也可以一块参与项目开发的贡献, 我们非常欢迎您加入我们的社区中来。

### 项目

项目名称	链接	描述
AntSK	<a href="https://github.com/AIDotNet/AntSK">https://github.com/AIDotNet/AntSK</a>	基于.Net8+AntBlazor+SemanticKernel 和KernelMemory 打造的AI知识库/智能体, 支持本地离线AI大模型。可以离线运行。支持aspire观测应用数据
FastWiki	<a href="https://github.com/AIDotNet/fast-wiki">https://github.com/AIDotNet/fast-wiki</a>	基于.NET8+React+LobeUI实现的企业级智能客服知识库
Thor(雷神托尔)	<a href="https://github.com/AIDotNet/Thor">https://github.com/AIDotNet/Thor</a>	Thor提供了大部分的AI模型兼容OpenAI的接口格式, 并且将所有模型的实现单独成类库打包成SDK使用, 可快速使用入门, 也可以使用Thor的服务部署成独立的AI中转服务, 在Thor中提供了基本的用户管理和权限管理, 并且支持多模型转换, 以便提供给服务OpenAI的API风格。
GraphRag.Net	<a href="https://github.com/AIDotNet/GraphRag.Net">https://github.com/AIDotNet/GraphRag.Net</a>	参考GraphRag使用 Semantic Kernel 来实现的dotnet版本, 可以使用NuGet开箱即用集成到项目中
ThorChat	<a href="https://github.com/AIDotNet/ThorChat">https://github.com/AIDotNet/ThorChat</a>	这个是移植lobechat, 将next换成纯静态项目, 后台使用.NET 8 提供WebApi支持
AutoGpt	<a href="https://github.com/AIDotNet/AutoGpt">https://github.com/AIDotNet/AutoGpt</a>	AutoGpt 智能推理SDK, 利用AI本身的能力进行多次对话推理, 回复效果更加理想, 更智能。



我们社区聚集了很多对AI感兴趣的.Net开发者

**.net framework windows平台**



**.net core 跨平台**



**各种中间件(缓存、队列、追踪)**



**容器、集群、微服务**



**DevOps**



**AI 大模型**

**积极拥抱AI  
把握未来趋势！**

# 感谢观看

多元化社区 | 大咖面对面 | 技术交流圈 | 编程挑战赛 | 求职招聘圈 尽在黄浦论坛



2024.NET技术沙龙