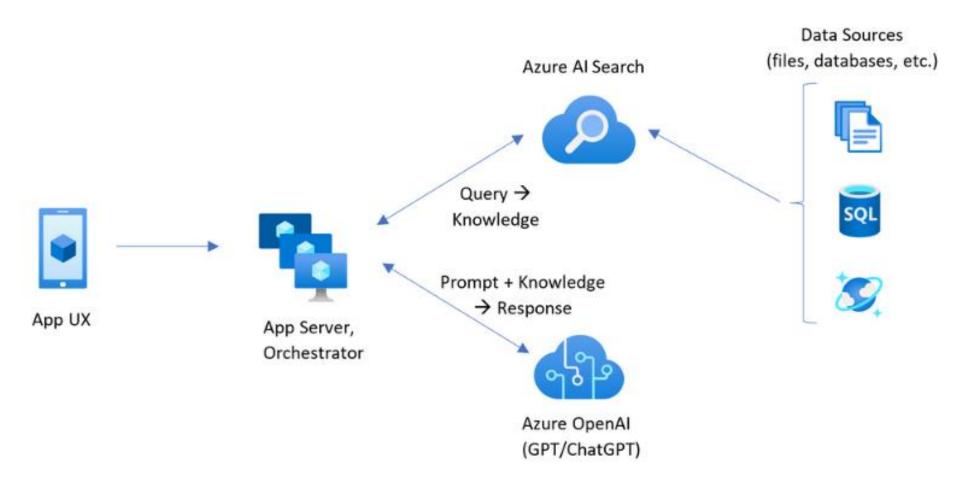
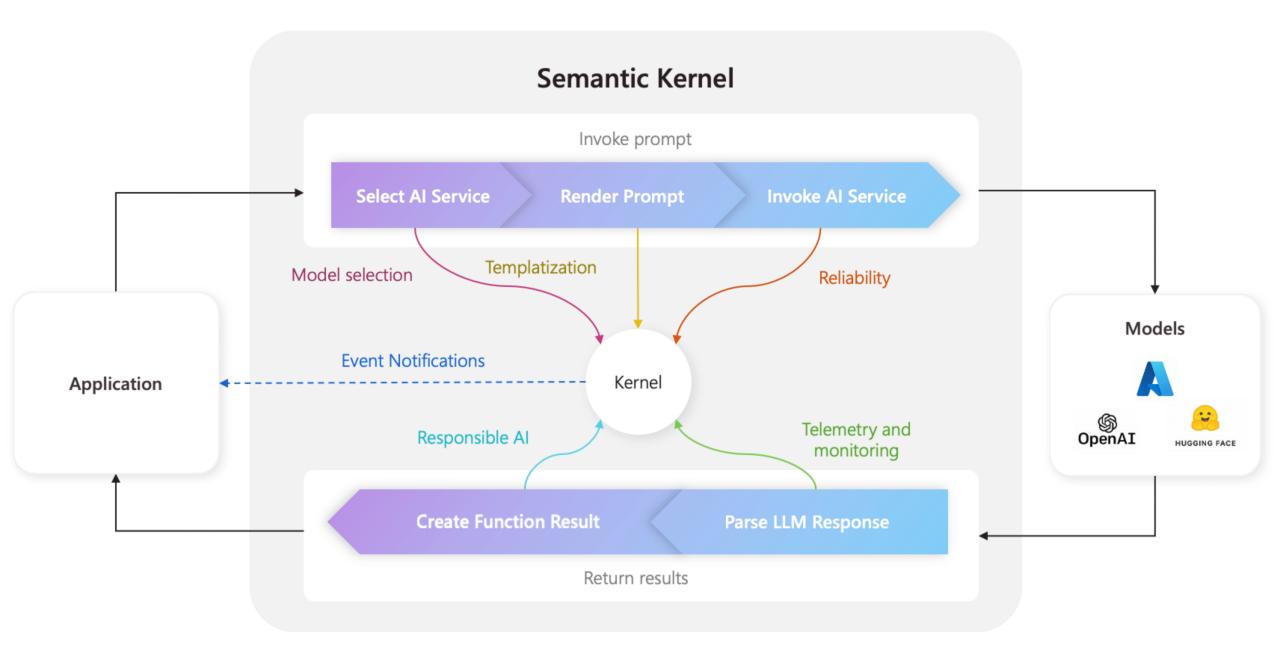


What is Retrieval Augmented Generation (RAG)?





Writer plugin **Function** Description for model Given a goal or topic description generate a list of ideas. Brainstorm Write an email from the given bullet points. EmailGen ShortPoem Turn a scenario into a short and entertaining poem. Generate a list of synopsis for a novel or novella with sub-chapters. StoryGen Translate Translate the input into a language of your choice. Can you write me a short poem about **Planner** living in Dublin, Ireland and then create a story based on the poem? Copilot Sure! Here's a story based on living along the Grand Canal in Dublin, Ireland...

```
mirror_mod.mirror_object
Peration == "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mlrror_mod.use_z = False
 _operation == "MIRROR_Y"
lrror_mod.use_x = False
irror_mod.use_y = True
mirror_mod.use_z = False
  operation == "MIRROR Z"
  lrror_mod.use_x = False
  rror_mod.use_y = False
  rror_mod.use_z = True
  election at the end -add
  * ob.select= 1
 "Selected" + str(modified) EVO
   drror ob.select = 0
  bpy.context.selected_obj
   lata.objects[one.name].sel
  Int("please select exactle
  -- OPERATOR CLASSES ----
   (ypes.Operator):
  X mirror to the selected
  mject.mirror_mirror_x"
  FFOF X"
```

Home > dotnetkonf24-oai



dotnetkonf24-oai | Keys and Endpoint 🖈

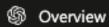
Azure OpenAl



Regenerate Key1



Regenerate Key2







- **Tags**
- Diagnose and solve problems

Resource Management

- **Keys and Endpoint**
- Model deployments
- Encryption
- Pricing tier
- Networking
- Identity

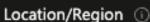
These keys are used to access your Azure Al services API. Do not share your keys. Store them securely- for example, using Azure Key Vault. We also recommend regenerating these keys regularly. Only one key is necessary to make an API call. When regenerating the first key, you can use the second key for continued access to the service.

Show Keys

KEY 1











D



There's more to discover in the updated Azure AI Studio

Try more models, capabilities, and tools for building your Al solution, or bring your existing deployments without any extra steps.

Explore Azure Al Studio

« Azure OpenAl Playground chat **2** Completions □ DALL·E Assistants (Preview) Management Deployments

Models (

Data files

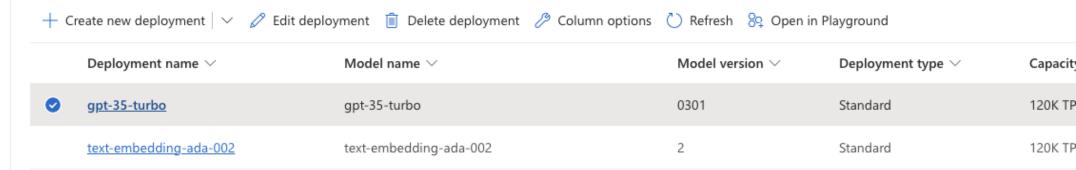
Quotas

Content filters (Preview)

Azure OpenAl Studio > Deployments

Deployments

Deployments provide endpoints to the Azure OpenAI base models, or your fine-tuned models, configured with settings to meet your needs, including the content moderation mo



```
C# Program.cs X
            using Microsoft.Extensions.Configuration;
\mathbb{A}
            using Microsoft.SemanticKernel;
80
            namespace RagDemo;
internal class Program
                private static async Task Main()
                    var configuration = new ConfigurationBuilder()
                         .AddUserSecrets<Program>() // IConfigurationBuilder
                         .Build();
                    var kernel = Kernel.CreateBuilder()
                         .AddAzureOpenAIChatCompletion(deploymentName: "gpt-35-turbo",
                             endpoint: configuration["AzureOpenAI:Endpoint"]!,
                             apiKey: configuration["AzureOpenAI:ApiKey"]!) // IKernelBuilder
                         .Build();
                    var chat = new BasicChat(kernel);
≣
                    //var chat = new KernelPlugin(kernel);
o
                    //var chat = new History(kernel);
                    //var chat = new HistoryStream(kernel);
Ō
                    //var chat = new BasicRag(kernel);
Θ
                    //var chat = new Embeddings(kernel, configuration);
ŧŌ
                    await chat.InitChat();
    RagDemo {} RagDemo > 💝 Program
           № RagDemo
<u>©</u>
        □ = → 前
   /Users/mertyeter/source/repos/mertyeter/dotnetkonf24/src/RAGDemo/bin/Debug/net8.0/RagDemo
Question: What is .NET?
    .NET is a framework developed by Microsoft for building and executing applications on Windows operating systems. It provides a set of libraries, tools, and runtime environments that facilitate the cr
\Sigma
    for desktop, web, mobile, gaming, and IoT devices. It supports multiple programming languages such as C#, F#, and Visual Basic, and enables developers to create applications that are secure, scalables developers.
The .NET framework also includes features such as memory management, exception handling, and garbage collection to simplify the development process and improve the quality of applications.
                                                                                                                                                                                       #dotnetkonf24
```

የ Question: ∏

```
using Microsoft.SemanticKernel;
       namespace RagDemo;

≥ 1 usage

       public class BasicChat(Kernel kernel) : IChat

☑ 1 usage

           public async Task InitChat()
                while (true)
10
                    Console.Write("Question: ");
11
12
                    Console.WriteLine(await kernel.InvokePromptAsync(promptTemplate:Console.ReadLine()!))
13
                    Console.WriteLine();
14
15
                // ReSharper disable once FunctionNeverReturns
16
                                                                                              #dotnetkonf24
```

```
public class KernelPlugin(Kernel kernel) : IChat
\mathbb{A}

∨ ©# RagDemo

        > & Dependencies
          C# 01-BasicChat.cs
80
         C# 02-KernelPlugin.cs
                                                                    public async Task InitChat()
C# 03-History.cs
         C# 04-HistoryStream.cs
                                                                        kernel.ImportPluginFromFunctions(pluginName: "DateTimeHelpers",
         C# 05-BasicRag.cs
         C# 06-Embeddings.cs
                                                                            kernel.CreateFunctionFromMethod(method:() => $"{DateTime.UtcNow:r}",
         C# IChat.cs
                                                                                 functionName: "Now",
         C# Program.cs
                                                                                 description: "Gets the current date and time") // KernelFunction
     > Scratches and Consoles
                                                                        var kernelFunction = kernel.CreateFunctionFromPrompt(promptTemplate: """
                                                                                                                                 The current date and time is {{ datetimehelpers.now }}.
                                                                                                                                 {{ $input }}
                                                                        var kernelArguments = new KernelArguments();
                                                                        while (true)
                                                                            Console.Write("Question: ");
©
                                                                            kernelArguments["input"] = Console.ReadLine();
                                                                            Console.WriteLine(await kernelFunction.InvokeAsync(kernel, kernelArguments));
Ō
                                                                            Console.WriteLine();
9
                                                                        // ReSharper disable once FunctionNeverReturns
<u>₹</u>Ō
                                                       RagDemo {} RagDemo > 4 KernelPlugin
T

    RagDemo ⇒

     Run
<u>©</u>
    /Users/mertyeter/source/repos/mertyeter/dotnetkonf24/src/RAGDemo/bin/Debug/net8.0/RagDemo
```

C# Program.cs

C# 02-KernelPlugin.cs >

Question: What time is it?

<u>></u>

Question:

Solution V @ Q

The current time is 17:30:55 GMT.

```
public class KernelPlugin(Kernel kernel) : IChat
\mathbb{A}
                   C# RagDemo
                     > & Dependencies
                                                                                                                                               7 ^1
                                                                                                                                                                          public async Task InitChat()
                          C# 01-BasicChat.cs
 80
                          C# 02-KernelPlugin.cs
                                                                                                                                                                                     kernel.ImportPluginFromFunctions(pluginName: "DateTimeHelpers",
 C# 03-History.cs
                         C# 04-HistoryStream.cs
                                                                                                                                                                                                kernel.CreateFunctionFromMethod(method:() => $"{DateTime.UtcNow:r}",
                         C# 05-BasicRag.cs
                                                                                                                                                                                                           functionName: "Now",
                          C# 06-Embeddings.cs
                                                                                                                                                                                                           description: "Gets the current memory usage") // KernelFunction
                          C# IChat.cs
                                                                                                                                                                                     1);
                          C# Program.cs
              ⇒ Scratches and Consoles
                                                                                                                                                                                     var kernelFunction = kernel.CreateFunctionFromPrompt(promptTemplate: """
                                                                                                                                                                                                                                                                                                                                    The current date and time is {{ datetimehelpers.now }}.
                                                                                                                                                                                                                                                                                                                                    {{ $input }}
                                                                                                                                                                                     var kernelArguments = new KernelArguments();
                                                                                                                                                                                      while (true)
                                                                                                                                                                                                Console.Write("Question: ");
                                                                                                                                                                                                kernelArguments["input"] = Console.ReadLine();
                                                                                                                                                                                                Console.WriteLine(await kernelFunction.InvokeAsync(kernel, kernelArguments));
 ©
                                                                                                                                                                                                Console.WriteLine();
 Ō
                                                                                                                                                                                     // ReSharper disable once FunctionNeverReturns
 Θ
 ŧŌ
                                                                                                                                          RagDemo → RagDe
T

    RagDemo ×

             Run
 <u>©</u>
           /Users/mertyeter/source/repos/mertyeter/dotnetkonf24/src/RAGDemo/bin/Debug/net8.0/RagDemo
          Question: what time is it?
          As an AI language model, I do not have access to the current date and time. However, based on the provided information, the current time is 17:35:31 GMT.
2
```

c# 02-KernelPlugin.cs ×

C# Program.cs

Solution ~

Question:

```
Solution ~
                                                                                                                                                            C# 03-History.cs × C# 02-KernelPlugin.cs
                                                                                                                         C# Program.cs
\mathbb{A}
                ©# RagDemo
                                                                                                                                          namespace RagDemo;
                  > & Dependencies
                      C# 01-BasicChat.cs
80
                     C# 02-KernelPlugin.cs
 public class History(Kernel kernel) : IChat
                      C# 03-History.cs
                     C# 04-HistoryStream.cs
                     C# 05-BasicRag.cs
                     C# 06-Embeddings.cs
                                                                                                                           8 ^1
                                                                                                                                                   public async Task InitChat()
                     C# IChat.cs
                     C# Program.cs
                                                                                                                                                             var chatCompletionService = kernel.GetRequiredService<IChatCompletionService>();
             Scratches and Consoles
                                                                                                                                                             ChatHistory chatHistory = new(systemMessage: "You are an AI assistant that helps people find information.");
                                                                                                                                                             while (true)
                                                                                                                                                                      Console.Write("Question: ");
                                                                                                                                                                      chatHistory.AddUserMessage(content:Console.ReadLine()!);
                                                                                                                                                                      var answer:ChatMessageContent = await chatCompletionService.GetChatMessageContentAsync(chatHistory);
                                                                                                                                                                      chatHistory.AddAssistantMessage(answer.Content!);
                                                                                                                                                                      Console.WriteLine(answer);
                                                                                                                                                                      Console.WriteLine();
 O
                                                                                                                                                             // ReSharper disable once FunctionNeverReturns
 Ō
                                                                                                                       RagDemo
                                                                                                                                                {} RagDemo → 

→ History

 9

    RagDemo
    RagDemo

                   /Users/mertyeter/source/repos/mertyeter/dotnetkonf24/src/RAGDemo/bin/Debug/net8.0/RagDemo
       Question: Tell me a joke about Star Wars
         Why did Princess Leia keep her hair tied up in buns?
℗
         To avoid the Darth Maul.
মূ Question: Explain why it's funny
         This joke is funny because it's a clever play on words. "Darth Maul" is the name of a character from the Star Wars franchise who has distinctive red horns on his head. The word "maul" also means to a
 e with extreme violence. In the joke, "Darth Maul" is replaced with "maul" as a verb, which makes it sound like Princess Leia is avoiding getting attacked or injured by tying har hair up in bur 24 #dotnetkonf24
```

words makes the joke funny.

```
public async Task InitChat()
    var chatCompletionService = kernel.GetRequiredService<IChatCompletionService>();
    ChatHistory chatHistory = new(systemMessage: "You are an AI assistant that helps people find information.");
    StringBuilder builder = new();
   while (true)
        Console.Write("Question: ");
        chatHistory.AddUserMessage(content:Console.ReadLine()!);
        builder.Clear();
        await foreach (var message in chatCompletionService.GetStreamingChatMessageContentsAsync(chatHistory))
            Console.Write(message);
           builder.Append(message.Content);
        Console.WriteLine();
        chatHistory.AddAssistantMessage(content:builder.ToString());
                                                                                                      #dotnetkonf24
       Console Writeline().
```

/Users/mertyeter/source/repos/mertyeter/dotnetkonf24/src/RAGDemo/bin/Debug/net8.0/RagDemo

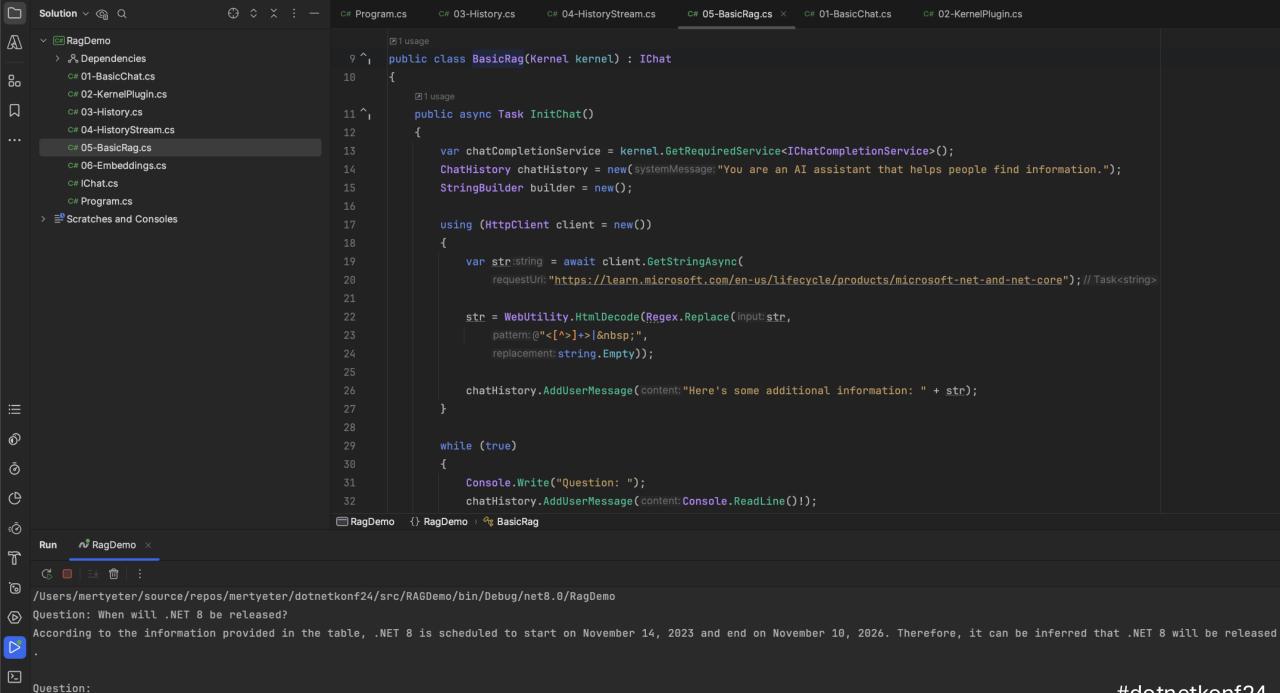
Question: What is .NET?

.NET is a software development framework developed by Microsoft that provides a consistent programming experience for cr eating applications for Windows, web, mobile, and other platforms. It includes a vast library of pre-built code, tools, and technologies that developers can use to build high-performance applications. .NET is designed to support multiple pr ogramming languages, including C#, Visual Basic, and F#. It also includes features such as memory management, security, and interoperability between different software components. .NET is widely used in enterprise application development and is one of the most popular software development frameworks in the world.

Question: When will .NET 8 be released?

As an AI language model, I do not have information about the exact release date of .NET 8 or any future versions of .NET . However, according to the Microsoft .NET roadmap, .NET 8 is expected to be released in November 2021. Please note that release dates are subject to change based on development progress and other factors.

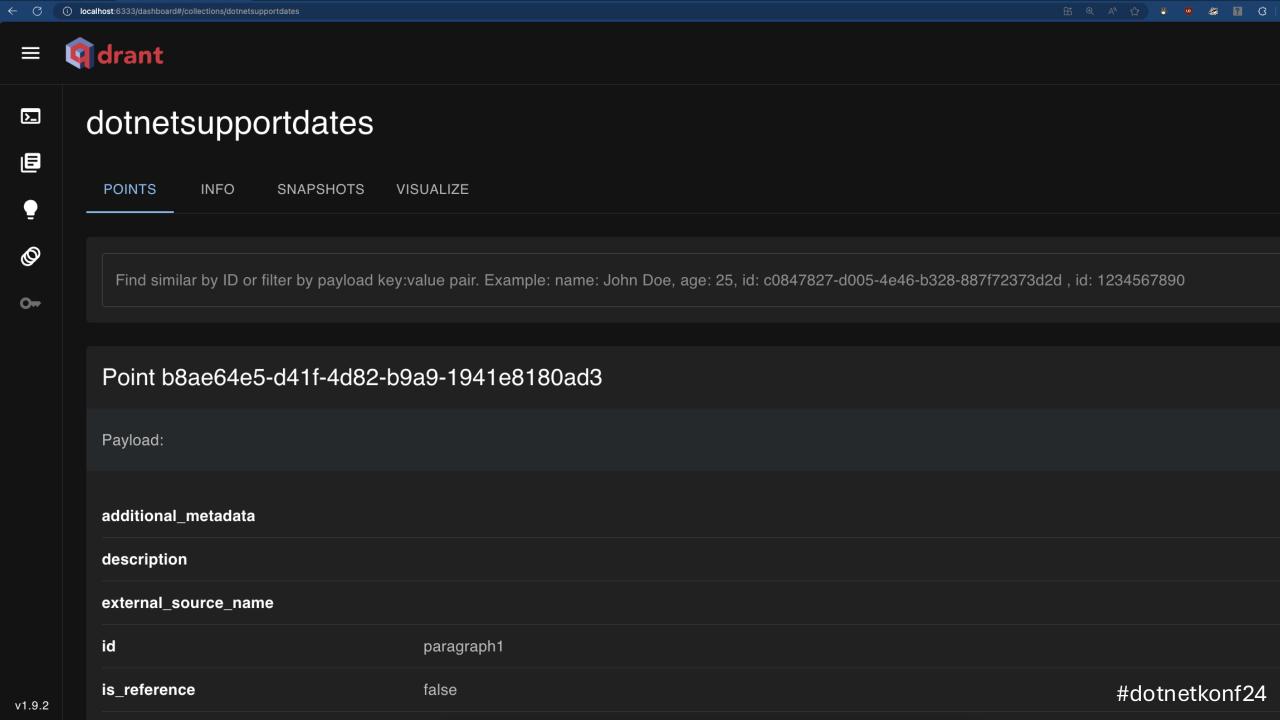
Question:



①

```
public class Embeddings(Kernel kernel, IConfiguration configuration) : IChat
    [Experimental(diagnosticid: "SKEXP0001")]
    public async Task InitChat()
        var semanticTextMemory = new MemoryBuilder()
            .WithMemoryStore(new QdrantMemoryStore(endpoint: "http://localhost:6333/", vectorSize:1536))
            .WithAzureOpenAITextEmbeddingGeneration(deploymentName: "text-embedding-ada-002",
                endpoint: configuration["AzureOpenAI:Endpoint"]!,
                apiKey: configuration["AzureOpenAI:ApiKey"]!) // MemoryBuilder
            .Build();
        var collections:|List<string> = await semanticTextMemory.GetCollectionsAsync();
        const string collectionName = "dotnetsupportdates";
        if (collections.Contains(collectionName))
            Console.WriteLine("Found database");
        else
            using HttpClient client = new();
            var str:string =
                await client.GetStringAsync(
                     requestUri: "https://learn.microsoft.com/en-us/lifecycle/products/microsoft-net-and-net-core"); // Task<string>
            var paragraphs:List<string> =
                TextChunker.SplitPlainTextParagraphs(
                     TextChunker.SplitPlainTextLines(
                         WebUtility.HtmlDecode(Regex.Replace(input:str, pattern:@"<[^>]+>|&nbsp;", replacement:string.Empty)),
                         maxTokensPerLine: 128),
                     maxTokensPerParagraph: 1024);
            for (var i = 0; i < paragraphs.Count; <math>i++)
                await semanticTextMemory.SaveInformationAsync(collectionName, text:paragraphs[i], id:$"paragraph{i}");
            Console.WriteLine("Generated database");
```

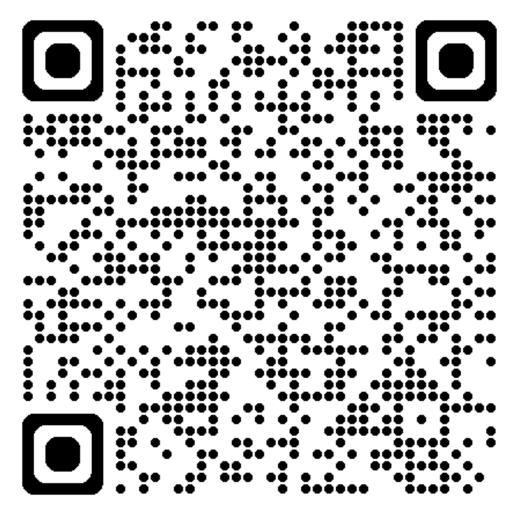
```
var chatCompletionService = kernel.GetRequiredService<IChatCompletionService>();
ChatHistory chatHistory = new(systemMessage: "You are an AI assistant that helps people find information.");
StringBuilder builder = new();
while (true)
    Console.Write("Question: ");
    var question:string = Console.ReadLine()!;
    builder.Clear();
    await foreach (var result in semanticTextMemory.SearchAsync(collectionName, query: question, limit: 3))
        builder.AppendLine(result.Metadata.Text);
    var contextToRemove:int = -1;
    if (builder.Length != 0)
        builder.Insert(0, "Here's some additional information: ");
        contextToRemove = chatHistory.Count;
        chatHistory.AddUserMessage(content:builder.ToString());
    chatHistory.AddUserMessage(question);
    builder.Clear();
    await foreach (var message in chatCompletionService.GetStreamingChatMessageContentsAsync(chatHistory))
        Console.Write(message);
        builder.Append(message.Content);
    Console.WriteLine();
    chatHistory.AddAssistantMessage(content:builder.ToString());
    if (contextToRemove >= 0) chatHistory.RemoveAt(contextToRemove);
    Console.WriteLine();
```



Resources



<u>Understanding the kernel in Semantic Kernel |</u> <u>Microsoft Learn</u>



RAG and generative AI - Azure AI Search | Microsoft Learn