

היסטוריה וזיכרון



גלעד מרקמן
[פרויקט ב-Github](#)

זיכרון - stateless

```
1  > using ...
6
7  Env.Load(@"C:\Users\Gilad\source\repos\SK\.env");
8  var OpenAIKey = Environment.GetEnvironmentVariable("OpenAIKey");
9
10 string model = "gpt-5-mini"; //"gpt-4.1-mini";
11
12 // Create a Semantic Kernel builder instance
13 var builder = Kernel.CreateBuilder();
14
15 // Add the OpenAI chat completion service to the kernel builder
16 builder.AddOpenAIChatCompletion(model, OpenAIKey);
17
18 // Build the kernel with the configured services
19 var kernel = builder.Build();
20
21 // Retrieve the chat completion service from the kernel
22 var chatService = kernel.GetRequiredService<IChatCompletionService>();
23
24 while (true)
25 {
26     // User prompt message
27     Console.WriteLine(">> ");
28     string userMessage = Console.ReadLine();
29     if (string.IsNullOrEmpty(userMessage))
30     {
31         break; // Exit the loop if the user enters an empty message
32     }
33
34     // Send the user's message to the chat model and await the response
35     var result = await chatService.GetChatMessageContentAsync(userMessage);
36
37     Console.WriteLine(result.Content);
38 }
39
```

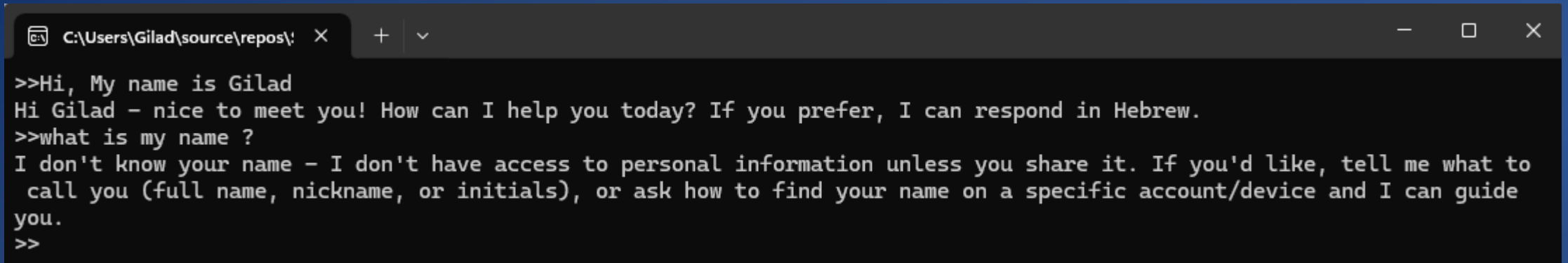
- מודלים של בינה מלאכותית (כמו כל אתרי האינטרנט) אין להם זיכרון. לאחר שנתנו לנו תשובה הם לא שומרים את השיחה.

- נדגים זאת עם התוכנית שלנו.

- אך תחילה נשדרג את התוכנית שלנו ונבנה לולאת שיחה.

מודל חסר זיכרון - stateless

- כפי שאנחנו רואים המודל אינו "זוכר" את ההיסטוריה של השיחה.



```
C:\Users\Gilad\source\repos\  X + v
>>Hi, My name is Gilad
Hi Gilad - nice to meet you! How can I help you today? If you prefer, I can respond in Hebrew.
>>what is my name ?
I don't know your name - I don't have access to personal information unless you share it. If you'd like, tell me what to
call you (full name, nickname, or initials), or ask how to find your name on a specific account/device and I can guide
you.
>>
```

- על מנת להתמודד עם מצב זה עלינו לטפל בעצמנו בזיכרון של השיחה, ולדאוג להעביר למודל את ההיסטוריה של השיחה.
- על כך נלמד בשיעור הבא.

היסטוריה

- על מנת לספק למודל שלנו זיכרון עלינו להוסיף את היסטוריית השיחה למידע הנשלח אל המודל.
- ניתן לעשות זאת בצורה ידנית על ידי שרשור של היסטוריית השיחה, אולם SK הכינו לנו אובייקט מיוחד לצורך כך ChatHistory.
- אובייקט ההיסטוריה שומר רשימה של מחרוזות, בשלושה סוגים:
 - SystemMessage – נסביר בשיעור הבא.
 - UserMessage – הטקסט של המשתמש
 - AssistantMessage – התשובות של המודל.
- אנחנו מעדכן את האובייקט בהיסטוריית השיחה שלנו, ונשלח למודל כל פעם את מלוא ההיסטוריה (ולא רק השאלה האחרונה).

דוגמה - שימוש בהיסטוריה

```
ChatHistory history = new ChatHistory();  
  
while (true)  
{  
    // User prompt message  
    Console.WriteLine(">> ");  
    string userMessage = Console.ReadLine();  
    if (string.IsNullOrEmpty(userMessage))  
    {  
        break; // Exit the loop if the user enters an empty message  
    }  
  
    history.AddUserMessage(userMessage);  
    // Send the user's message to the chat model and await the response  
    var result = await chatService.GetChatMessageContentAsync(history);  
  
    Console.WriteLine(result.Content);  
    history.AddAssistantMessage(result.Content);  
}
```

- יצירת אובייקט
- הוספת השאלות של המשתמש.
- העברת מלוא ההיסטוריה למודל
- הוספת התשובות של המודל

