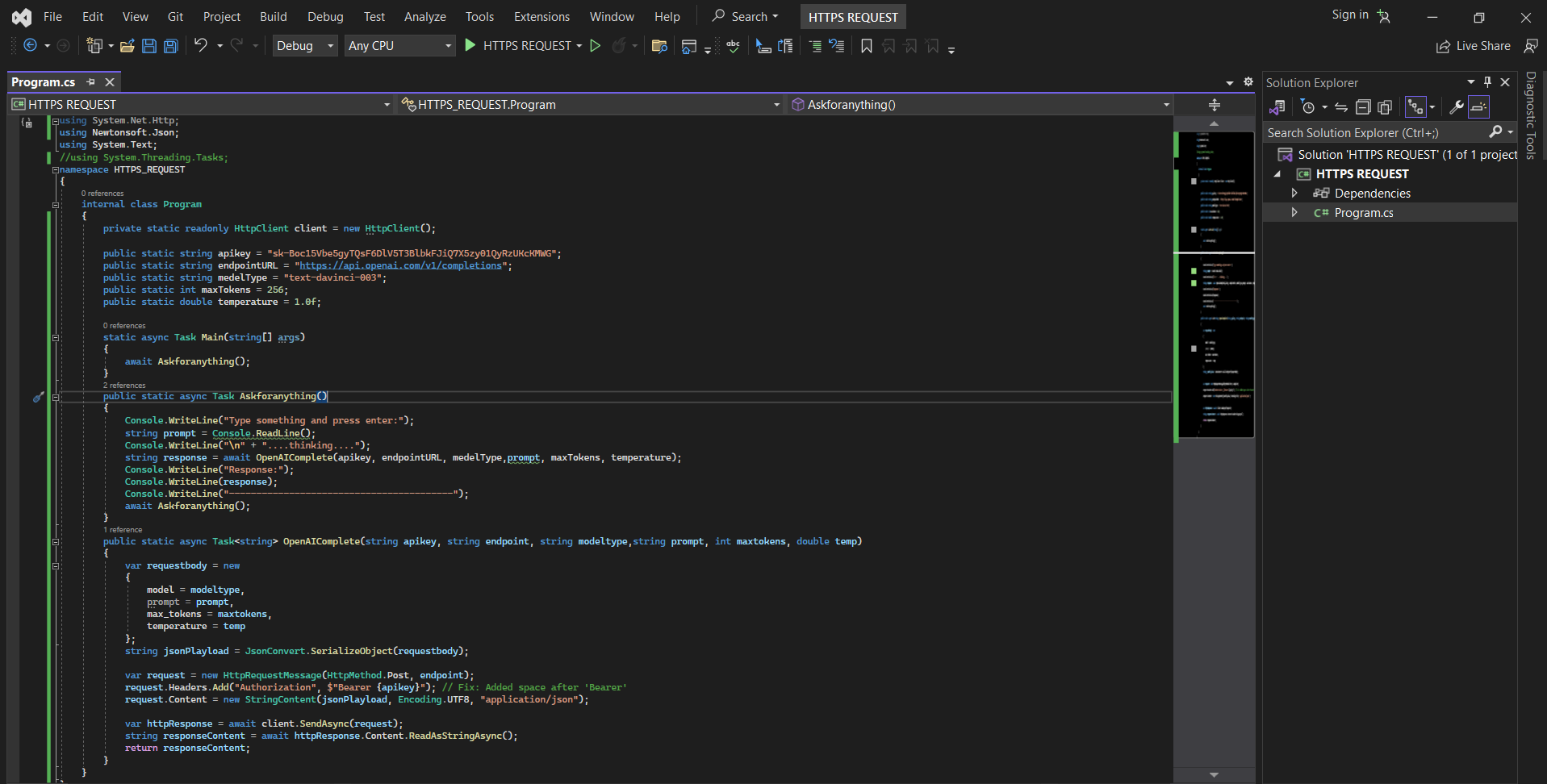
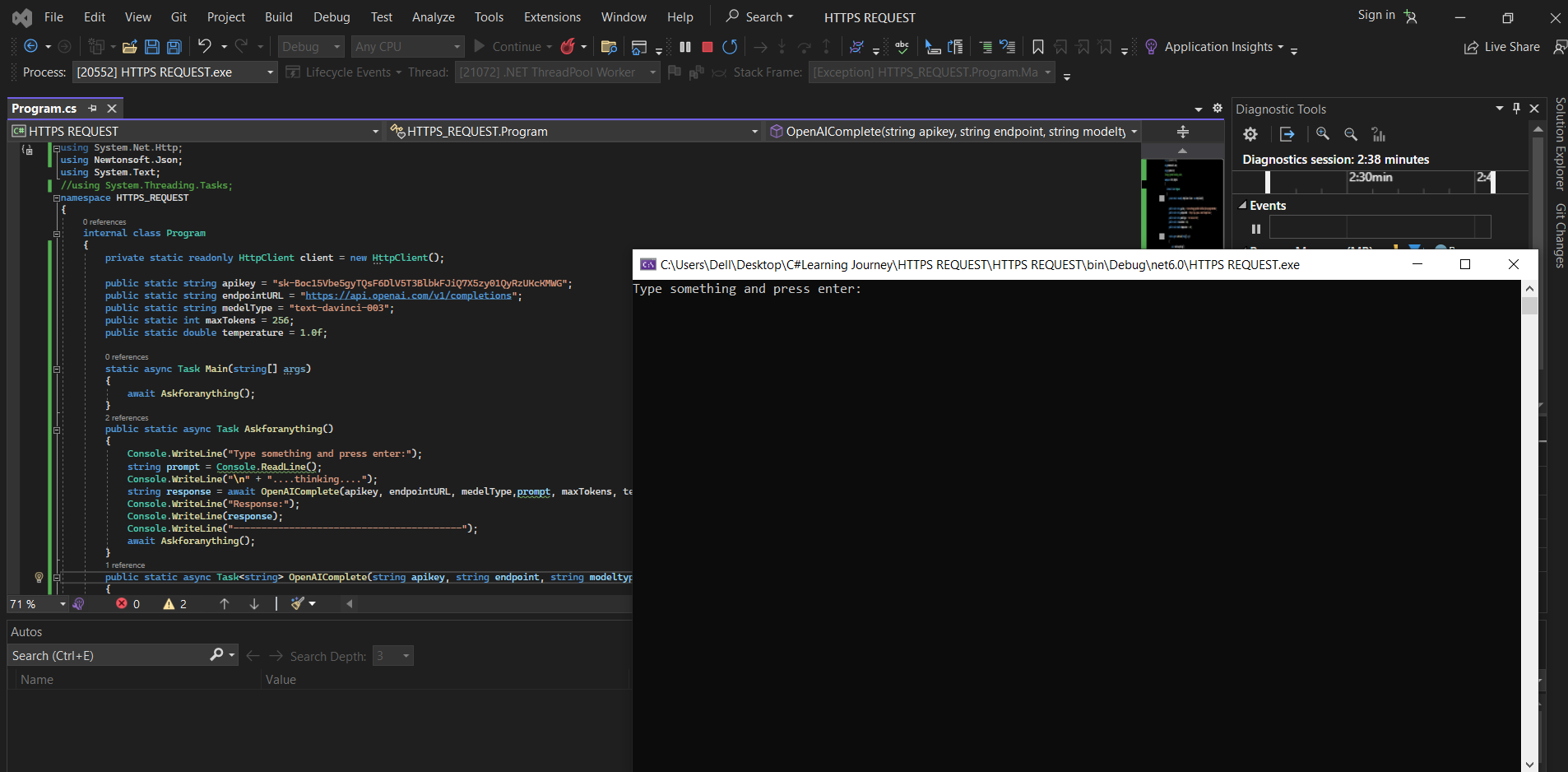
Unlock the power of conversation with ChatGPT API integration in Visual Studio! Seamlessly connect to the ChatGPT service using C# and initiate an immersive interaction experience. Empower users to pose any question, share their thoughts, or seek advice directly within your application. With the simple yet robust HTTP request setup, users can engage with ChatGPT's sophisticated AI, receiving prompt and insightful responses. Elevate user experiences, foster engaging discussions, and harness the capabilities of AI-driven conversations—all within the familiar environment of Visual Studio. Let your application become a gateway to limitless conversational possibilities with ChatGPT at your fingertips.





**Code in C# using Visual Studio:**

using System.Net.Http;

using Newtonsoft.Json;

using System.Text;

//using System.Threading.Tasks;

namespace HTTPS\_REQUEST

{

internal class Program

{

private static readonly HttpClient client = new HttpClient();

public static string apikey = "your APIKEY";

public static string endpointURL = "https://api.openai.com/v1/completions";

public static string medelType = "text-davinci-003";

public static int maxTokens = 256;

public static double temperature = 1.0f;

static async Task Main(string[] args)

{

await Askforanything();

}

public static async Task Askforanything()

{

Console.WriteLine("Type something and press enter:");

string prompt = Console.ReadLine();

Console.WriteLine("\n" + "....thinking....");

string response = await OpenAIComplete(apikey, endpointURL, medelType,prompt, maxTokens, temperature);

Console.WriteLine("Response:");

Console.WriteLine(response);

Console.WriteLine("-----------------------------------------");

await Askforanything();

}

public static async Task<string> OpenAIComplete(string apikey, string endpoint, string modeltype,string prompt, int maxtokens, double temp)

{

var requestbody = new

{

model = modeltype,

prompt = prompt,

max\_tokens = maxtokens,

temperature = temp

};

string jsonPlayload = JsonConvert.SerializeObject(requestbody);

var request = new HttpRequestMessage(HttpMethod.Post, endpoint);

request.Headers.Add("Authorization", $"Bearer {apikey}"); // Fix: Added space after 'Bearer'

request.Content = new StringContent(jsonPlayload, Encoding.UTF8, "application/json");

var httpResponse = await client.SendAsync(request);

string responseContent = await httpResponse.Content.ReadAsStringAsync();

return responseContent;

}

}

}