import os

import json

from dotenv import load\_dotenv

# Add OpenAI import

from openai import AzureOpenAI

def main():

    try:

        # Flag to show citations

        show\_citations = False

        # Get configuration settings

        load\_dotenv()

        azure\_oai\_endpoint = os.getenv("AZURE\_OAI\_ENDPOINT")

        azure\_oai\_key = os.getenv("AZURE\_OAI\_KEY")

        azure\_oai\_deployment = os.getenv("AZURE\_OAI\_DEPLOYMENT")

        azure\_search\_endpoint = os.getenv("AZURE\_SEARCH\_ENDPOINT")

        azure\_search\_key = os.getenv("AZURE\_SEARCH\_KEY")

        azure\_search\_index = os.getenv("AZURE\_SEARCH\_INDEX")

        # Initialize the Azure OpenAI client

        client = AzureOpenAI(

            base\_url=f"{azure\_oai\_endpoint}/openai/deployments/{azure\_oai\_deployment}/extensions",

            api\_key=azure\_oai\_key,

            api\_version="2023-09-01-preview")

        # Get the prompt

        text = input('\nEnter a question:\n')

        # Configure your data source

        # Configure your data source

        extension\_config = dict(dataSources = [

                {

                    "type": "AzureCognitiveSearch",

                    "parameters": {

                        "endpoint":azure\_search\_endpoint,

                        "key": azure\_search\_key,

                        "indexName": azure\_search\_index,

                    }

                }]

            )

        # Send request to Azure OpenAI model

        print("...Sending the following request to Azure OpenAI endpoint...")

        print("Request: " + text + "\n")

        response = client.chat.completions.create(

            model = azure\_oai\_deployment,

            temperature = 0.5,

            max\_tokens = 1000,

            messages = [

                {"role": "system", "content": "You are a helpful travel agent"},

                {"role": "user", "content": text}

            ],

            extra\_body = extension\_config

        )

        # Print response

        print("Response: " + response.choices[0].message.content + "\n")

        if (show\_citations):

            # Print citations

            print("Citations:")

            citations = response.choices[0].message.context["messages"][0]["content"]

            citation\_json = json.loads(citations)

            for c in citation\_json["citations"]:

                print("  Title: " + c['title'] + "\n    URL: " + c['url'])

    except Exception as ex:

        print(ex)

if \_\_name\_\_ == '\_\_main\_\_':

    main()