



AZURE OPENAI CHAT WEB PART FOR SHAREPOINT ONLINE

Paul Borisov, Cloud Architect

Data Privacy

Azure OpenAI Chat web part interacts with private **Azure OpenAI** endpoints that are published via Azure API Management service (APIM).

- By default, this setup provides enhanced data privacy. In this configuration, requests to AI do not travel outside your Azure tenant.
- APIM consistently validates the identities of SharePoint users for each individual request. If the request originates from authorized domains, APIM retrieves the ****api-key**** from the secure vault and injects it into the request before forwarding it to the AI endpoint. This process ensures that the api-key does not get exposed in the browser.
- In addition to the default configuration, you have the option to publish the Native Open AI endpoint in APIM. You can find instructions in the project documentation.
 - **CONS:** Granting access to the Native Open AI endpoint requires a separate **api-key** for it and could potentially compromise data privacy, as requests might travel outside your Azure tenant under this setup.
 - **PROS:** Using the Native OpenAI endpoint could grant you access to the latest language models like GPT-4 Vision and GPT-4 1106 Parallel processing that are not currently available in Azure OpenAI.

User Interface

Ask OpenAI and share your experience with colleagues.

My chats ⊕ New chat 🔄

30.10: Generate CSV-table for a...

30.10: PowerShell code to conn...

28.10: Testing C# code

28.10: How do I ping an IP addr...

27.10: Why would an SQL query ...

24.10: Give me some good nam...

19.10: how to browse tags that ...

17.10: java code to connect to A... ✎ 🗑 📌

14.10: java code to connect to ...

06.10: Draw a table in HTML wit...

Shared chats 🔍 Hide my chats

Code style stackoverflowDark

java code to connect to Azure SQL database

⚡ GPT-3.5

⚡ GPT-4

To full screen

Send a message

0 characters, 30000 more allowed



java code to connect to Azure SQL database



Sure! Here's an example Java code snippet to connect to an Azure SQL database using JDBC:

```
java

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class AzureSQLConnection {
    public static void main(String[] args) {
        // Azure SQL Server credentials
        String server = "your_server_name.database.windows.net";
        String database = "your_database_name";
        String username = "your_username";
        String password = "your_password";

        // JDBC connection string
        String connectionString =
            String.format("jdbc:sqlserver://%s:1433;database=%s;user=%s;password=%s;encrypt=true;trustServerCertificate=false;hostNameInCertificate
```

Share?

Do you want to share 'java code to connect to Azure SQL database'?

Specific people to share with (if you want to limit access to this chat, max 15 persons)

AR Adam Ray
Software Engineer

BD Bjørn Dalevoll
Project Engineer

Suggested contacts

LK Lasse Karlsen
Senior Project Engineer

Share

Cancel

Full-Screen Mode

My chats

New chat

30.10: Generate CSV-table for a...

30.10: PowerShell code to conn...

28.10: Testing C# code

28.10: How do I ping an IP addr...

27.10: Why would an SQL query ...

24.10: Give me some good nam...

19.10: how to browse tags that ...

17.10: java code to connect to A...

14.10: java code to connect to ...

06.10: Draw a table in HTML wit...

05.10: Draw the text table with ...

04.10: Kan du norsk?

03.10: Detailed weather forecas...

Shared chats

Hide my chats

30.10: Generate CSV-table for all weeks ...

Code style

vs2015

PowerShell code to connect to Excel file and extract data into CSV

GPT-3.5

GPT-4

LLM: gpt-4-32k

Send a message

0 characters, 30000 more allowed

```
$connection.Open()

# Create a SQL query to select data from the Excel worksheet
$query = "SELECT * FROM [Sheet1$]"

# Create a new command object
$command = New-Object -TypeName System.Data.OleDb.OleDbCommand($query, $connection)

# Execute the query and get the result set
$result = $command.ExecuteReader()

# Get the column names from the result set
$columnNames = @()
for ($col = 0; $col -lt $result.FieldCount; $col++) {
    $columnName = $result.GetName($col)
    $columnNames += $columnName
}

# Create a new CSV file to store the extracted data with UTF-8 encoding using FileStream
$csvFilePath = "C:\path\to\your\output_file.csv"
$csvFile = New-Object -TypeName System.IO.FileStream -ArgumentList $csvFilePath, [System.IO.FileMode]::Create, [System.IO.FileAccess]::Write
$csvWriter = New-Object -TypeName System.IO.StreamWriter -ArgumentList $csvFile, [System.Text.Encoding]::UTF8

# Write the column headers to the CSV file
$csvWriter.WriteLine($columnNames -join ",")

# Loop through the result set and extract the data
while ($result.Read()) {
    $rowData = @()
    for ($col = 0; $col -lt $result.FieldCount; $col++) {
        $cellValue = $result.GetValue($col)
        $rowData += $cellValue
    }
    $csvWriter.WriteLine($rowData -join ",")
}

# Close the CSV writer, file, result set, and connection
$csvWriter.Close()
$csvFile.Close()
$result.Close()
$connection.Close()
```

In this updated code, we replaced the **StreamWriter** with **FileStream** to create the CSV file. We then pass the **FileStream** object as the first argument to the **StreamWriter** constructor, and **[System.Text.Encoding]::UTF8** as the second argument to specify the UTF-8 encoding.

Please note that we need to close both the CSV writer (**\$csvWriter**) and the CSV file (**\$csvFile**) separately before closing the result set and the connection.

Features

Key features

- Support for the latest text language models provided by Microsoft and Native OpenAI: GPT-4 32k, GPT 3.5 16k, GPT-4 Vision, GPT-4 1106, GPT-3.5 1106
- Unlimited Chats and Unlimited Chat History Length
- Chat Sharing: Unlimited for Everyone and Private for a limited number of Members
- Support for Full-Screen Mode. Support for Regular and Full-Width Site Page Sections
- Mobile-Friendly UI with simple customizations (CSS)
- Event Streaming to provide consecutive outputs of AI-responses
- Code Highlighter with options to select Highlight Styles.
- Upper and Lower Side positioning for the Prompt Area Textbox
- Full-Scale Localizations: English, Finnish, and Norwegian
- Optional integrations with company data. For security reasons, these integrations are disabled by default and must be explicitly enabled in the web part settings.
 - SharePoint Search
 - Company Users
 - Local Date and Time
 - Analysis of an uploaded PDF and summarization of its content
 - Analysis of uploaded images and description of their content. As of November 2023, Image Analysis is only available with the Native OpenAI endpoint.

Advanced features

- Built-in integration with API Management Service.
 - For customers that do not have it, the web part also supports direct queries to API with an api-key (configurable)
- Seamless support for Azure OpenAI and Native OpenAI API
- Alternative storage types to manage Chat History
 - SharePoint List: Supports up to 5000 Chats, Built-in “One-Click” list provisioning, Chat Privacy and Security Trimming for SharePoint Search
 - Azure SQL Database: Provides Enterprise-grade performance, Tested on 1 Million+ Chat records
 - Local Storage: it can be used to provide quick demo without long-time configurations

Optional Integrations: SharePoint Search, Company Users, Local Date & Time

How it works

The Function Calling feature is available in both Azure OpenAI and Native OpenAI (more advanced version). The flow for using this feature typically looks like:

1. In the process of interacting with OpenAI, you can send a request with options to call available integration functions that you have provided.
2. If OpenAI determines that it requires specific data, it will request to call a function with parameters that it deems suitable.
3. To meet OpenAI's demand, you can call the function with the proposed parameters on behalf of OpenAI. This allows you to obtain raw results locally, which you can then submit to OpenAI. You have control over what data you submit.
4. If OpenAI determines that it needs more information, it may demand calling another function, and this process can continue as necessary.
5. OpenAI handles the received data and transforms it as you have requested. The final response from OpenAI will include the results based on the processed data.

```
// 3 show cases available in the web part:
```

```
// Search in SharePoint for “your text“. Format the results as an HTML table.
```

```
// Get company users that have names starting with P. Format the results as an HTML table.
```

```
// Date, Time, Date and time.
```

```
export default class FunctionHelper {  
  private static available: { [key: string]: any } = {  
    companyUsers: this.companyUsers,  
    currentDateOrTime: this.currentDateOrTime,  
    currentDateAndTime: this.currentDateAndTime,  
    peopleSearch: this.companyUsers,  
    searchSharepoint: this.searchSharepoint,  
  };  
};
```

Integrations

Find in SharePoint "IDD Questionnaire" and show search results as HTML table



Find in SharePoint "IDD Questionnaire" and show search results as HTML table



Here are the search results for "IDD Questionnaire" in SharePoint:

Title	Author	Size	Path
Annex 3 - IDD Questionnaire for Corporates	Olivia Roca;SP Admin	85,822 bytes	Link
Annex 4 - IDD Questionnaire for State Owned Entities (SOEs)	Olivia Roca;SP Admin	85,302 bytes	Link
Annex 5 - IDD Questionnaire for Municipalities, Ministries and Other Government Agencies	Olivia Roca;SP Admin	84,004 bytes	Link



Show me company employees with names that start with p



Here is a list of company employees with names that start with the letter "P":

- Patti Fernandez
- Pradeep Gupta



Show me company employees with names that start with H



Here are the company employees with names that start with the letter "H":

- Harri Niemeläinen
- Henrietta Mueller



Date and time are...



The current date is November 10, 2023, and the current time is 12:02:22. If you have any more questions or need further assistance, feel free to ask!

Upload and Summarize PDF content

Summarise PDF content

⚡ GPT-3.5

⚡ GPT-4

⬆ (1)

Upload files

Summarise PDF content

21 characters, 12492 more allowed



Summarise PDF content



The PDF document provides information about the functionality of the Azure OpenAI Chat Web Part and its deployment, permissions, configuration, privacy settings, user interface, event streaming, backend components, and more. Here is a summary of the content:

1. Introduction: The document introduces the Azure OpenAI Chat Web Part and its frontend and backend components.
2. Deployment: It explains how to deploy the web part by adding its compiled solution to the App Catalog.
3. Permissions: After adding the package, the document instructs users to approve the requested permissions and create a standard Azure App registration named "openaiwp" to make authenticated requests.
4. Configuration: Users are guided on how to create a modern Site Page, add the web part, and configure its settings.
5. Privacy settings for SharePoint List storage: The document explains the privacy settings for storing chat histories in a SharePoint list.
6. Options to use native OpenAI instead of Azure OpenAI: Users have the option to use native OpenAI API URLs instead of Azure OpenAI ones.
7. User interface: The document describes the user interface of the web part, including the navigation panel, content panel, and code highlighting options.

Using GPT-4 Vision for Image Analysis: a Technical Diagram

What is in this image?



What is in this image?

SharePoint Online web part with regular and full-screen appearance



Alternative chat-history storage options with global and private sharing capabilities for users

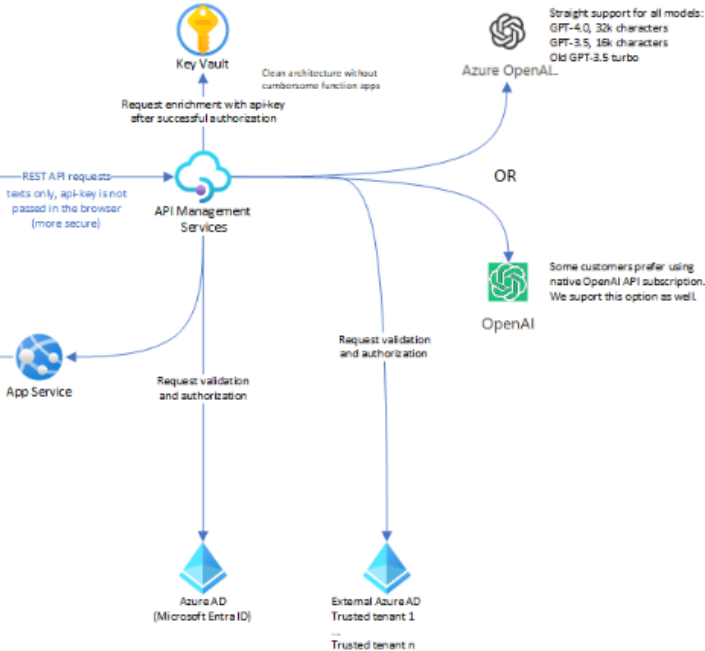
OR

OR

SharePoint List (Optimal, OOTB)

Web Browser (Quick demo)

AzureSQL database (Enterprise-grade, large Productions)



The image appears to be a flowchart or architectural diagram relating to a software or web application. It includes various components and services that interact with each other to support the application's functionality. Here are the components and descriptions visible in the diagram:

- SharePoint Online web part**: The image at the top left shows a SharePoint Online web part that can be displayed with both regular and full-screen appearance. This is likely a user interface component for displaying content.
- API Management Services**: Positioned in the center-right of the image, this service seems to handle REST API requests, request validation, and authorization, acting as a gateway between clients and backend services.
- App Service**: A central component on the left side of the diagram, identified with the Azure logo, which suggests that it's an Azure App Service used to host web applications.
- Key Vault**: Shown linking to the API Management Services and marked with text indicating that it enriches requests with an API key after successful authorization, which implies security and configurations management.
- OpenAI**: Mentioned on the right side of the image with a note that some customers prefer using their OpenAI subscription, indicating integration with AI services provided by OpenAI.
- Azure AD (Microsoft Entra ID)** and **External Azure AD**: Components for identity and access management.
- AzureSQL database (Enterprise-grade, large Productions)**: A cloud-based database service indicated for production-grade, larger-scale usage.
- Storage options**: At the bottom left, there are three alternatives presented for chat-history storage: SharePoint List, Web Browser (Quick demo), and AzureSQL database.

The image also contains annotations related to support for different GPT models, such as GPT-4.0, and the mention of secure transmission of API keys. Overall, the diagram seems to describe a web application architecture that leverages Azure cloud services and offers flexibility in terms of AI integration and user history storage.

Using GPT-4 Vision for Image Analysis: a Photo

What is in this image? Can you recognize this place?

⚡ GPT-3.5

🌟 GPT-4



Send a message



0 characters, 15000 more allowed



This image features the 25 de Abril Bridge in Lisbon, Portugal. You can recognize it by its distinctive red color, similar in style to the Golden Gate Bridge in San Francisco, USA. The bridge spans the Tagus River, and in this image, you can also see a dry, grassy foreground that gives a golden hue to the lower part of the photograph. Additionally, there are two sailboats visible on the water, providing a sense of leisure and maritime activity. The sky is clear, which suggests that the photo was taken on a day with good weather.

Azure OpenAI Chat web part for SharePoint Online, configurable options

Azure OpenAI Chat ×

Settings

Client ID: create a user_impersonation app with name=openaiwp

98765432-1234-5678-9876-543210123456

Base URL for GPT endpoint (APIM API or full)

https://tenant.azure-api.net/openai

Base URL for GPT4 endpoint (APIM API or full)

https://tenant.azure-api.net/openai4

Base URL for Chat WebApi (APIM API or full)

Optional api-key for Azure OpenAI (for troubleshooting, not for Production)

Language models

☒ GPT-3.5

☒ GPT-4

Storage type for chat history

Database

☒ Enable sharing

☒ Enable streaming

☒ Enable full screen mode

☒ Enable integrations

☒ Code highlighting

☒ Show highlighting styles

Default style

stackoverflowDark

☐ Show prompt area at bottom

☐ Unlimited chat history length (AI-responses in long chats may be less accurate)

Locale for dates (default is fi-FI)

Azure OpenAI Chat ×

Settings

Client ID: create a user_impersonation app with name=openaiwp

98765432-1234-5678-9876-5432101234...

Base URL for GPT endpoint (APIM API or full)

https://tenant.azure-api.net/openai

Base URL for GPT4 endpoint (APIM API or full)

https://tenant.azure-api.net/openainative

Base URL for Chat WebApi (APIM API or full)

Optional api-key for Azure OpenAI (for troubleshooting, not for Production)

Language models

☒ GPT-3.5

☒ GPT-4

Storage type for chat history

Database

☒ Enable sharing

☒ Enable streaming

☒ Enable full screen mode

☒ Enable integrations

☒ Code highlighting

☒ Show highlighting styles

Default style

stackoverflowDark

☐ Show prompt area at bottom

☐ Unlimited chat history length (AI-responses in long chats may be less accurate)

Locale for dates (default is fi-FI)

Azure OpenAI Chat ×

Settings

Client ID: create a user_impersonation app with name=openaiwp

98765432-1234-5678-9876-5432101234...

Base URL for GPT endpoint (APIM API or full)

https://tenant.azure-api.net/openai

Base URL for GPT4 endpoint (APIM API or full)

https://tenant.azure-api.net/openainative

Base URL for Chat WebApi (APIM API or full)

Optional api-key for Azure OpenAI (for troubleshooting, not for Production)

Language models

☒ GPT-3.5

☒ GPT-4

Storage type for chat history

SharePoint list

SharePoint list URL (leave it empty for default URL)

Create

Deploy in 1 click

☒ Enable sharing

☒ Enable streaming

☒ Enable full screen mode

☒ Enable integrations

☒ Code highlighting

☒ Show highlighting styles

Default style

stackoverflowDark

☐ Show prompt area at bottom

☐ Unlimited chat history length (AI-responses in long chats may be less accurate)

Locale for dates (default is fi-FI)

Azure OpenAI Chat ×

Settings

Client ID: create a user_impersonation app with name=openaiwp

98765432-1234-5678-9876-5432101234...

Base URL for GPT endpoint (APIM API or full)

https://tenant.azure-api.net/openai

Base URL for GPT4 endpoint (APIM API or full)

https://api.openai.com/v1/chat/completions

Base URL for Chat WebApi (APIM API or full)

Optional api-key for Azure OpenAI (for troubleshooting, not for Production)

Language models

☒ GPT-3.5

☒ GPT-4

Storage type for chat history

Local storage

☒ Enable sharing (demo only)

☒ Enable streaming

☒ Enable full screen mode

☒ Enable integrations

☒ Code highlighting

☒ Show highlighting styles

Default style

stackoverflowDark

☐ Show prompt area at bottom

☐ Unlimited chat history length (AI-responses in long chats may be less accurate)

Locale for dates (default is fi-FI)