Paul Borisov

Azure OpenAI Chat Web Part, security guidelines

Configurations for security options, v1

11-25-2023

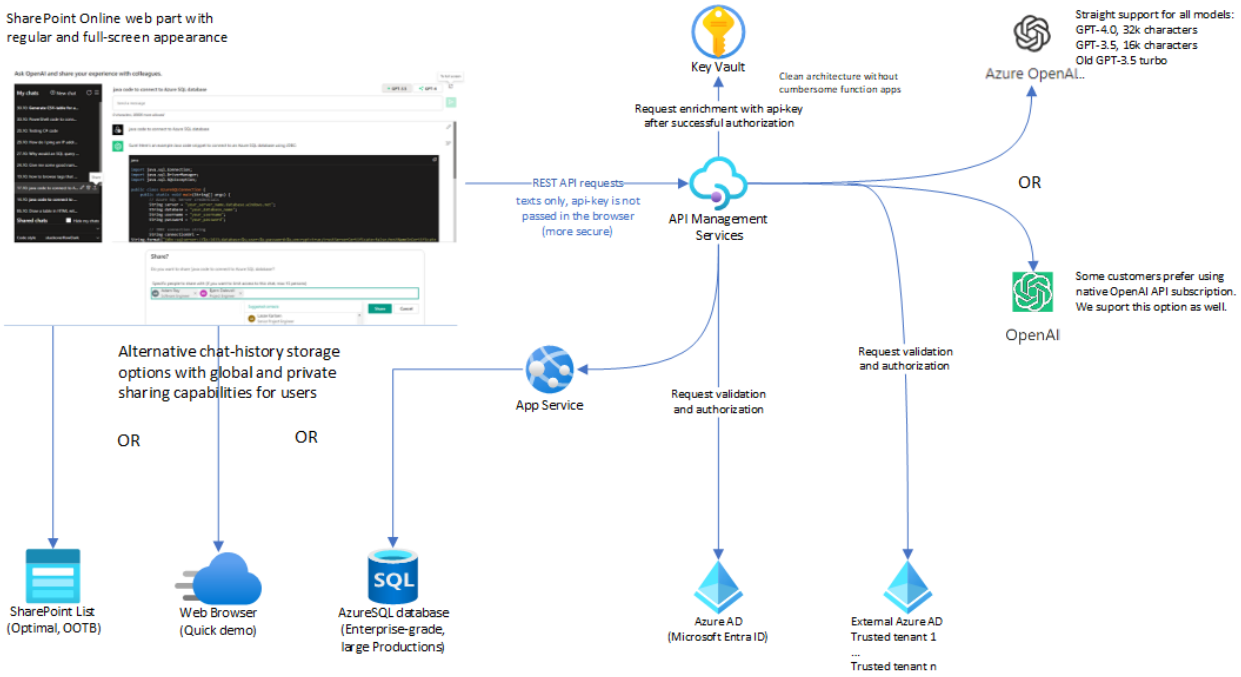
Table of Contents

Introduction 2
Data access, principal schema 2
SPFx authorization 3
App Service Web API authentication 3
Azure SQL connectivity from Web API 3
Deploying Web API to API Management 4
Restrict access to Web API only for API Management service 4
Optionally restrict access to Azure OpenAI for API Management service 4
Authenticated access to API Management endpoints 4
Optional restrictions for virtual networks 5

Introduction

This document provides detailed information on the configurable security options for the Azure OpenAI Chat Web Part, encompassing both its frontend and backend components. It should be read in conjunction with the supplemental document **azure-openai-chat-web-part.pdf**, which outlines the functionality of the Azure OpenAI Chat Web Part, its components, and configurations.

Data access, principal schema



SharePoint > SPFx extension > API Management

- JWT-token verification for AAD-users (bearer validity check) > Request enhancements to access services like adding api-key for Azure OpenAI then
    - JWT-token verification has functioned well in previous deployments.
    - It is not recommended to restrict JS-client access from the browser to API Management endpoints with additional CORS rules like the **origin** of SharePoint Online. Despite seeming feasible, this setup has proven unstable in previous deployments, randomly suspending requests until they timed out.
- API Management > Azure OpenAI > Deployments for ChatGPT 3.5 and ChatGPT-4
    - API Management supplies an **api-key** for Azure OpenAI, either from its encrypted local vault (which is recommended) or from a connected Azure Key Vault. In the latter case, the deployment of Azure Key Vault and its connection to the APIM instance is required.
- API Management > App Service ChatWebApi > Azure SQL database (default name dbChats), which stores Chats history data
    - The App Service should ideally use a System Managed Identity for DB access.
    - **Web API** refers to the App Service ChatWebApi mentioned later in this document.

SPFx authorization

You will need an Azure app with the **user_impersonation** scope added to make authorized calls to services hosted on Azure.

Actions

- Create a new regular default Azure AD App registration.
    - Use the predefined App named **openaiwp.**
    - Save ClientID. You will need it when configuring web part settings.
- Build and deploy SPFx solution package **azure-openai-chat.sppkg**.
    - Refer to the supplemental document **azure-openai-chat-web-part.pdf** for details.
    - As stated in the document, a Global Admin/Application Admin must approve the pending permission request via https://**tenant**-admin.sharepoint.com/_layouts/15/online/AdminHome.aspx#/webApiPermissionManagement

App Service Web API authentication

Use the **openaiwp** app created earlier to secure the App Service Web API (which is used for Azure SQL operations) with Microsoft Authentication provider.

Actions

- App Service for WebApi > Settings > Authentication > Add identity provider > Microsoft > Select the existing app **openaiwp.**
- Note that you may need to add the **user_impersonation** scope
    - Microsoft wizard does not always add it automatically.
    - Additional checks / configurations may be required.
    - Typically, the wizard correctly adds the scope for a newly created App (within the wizard), but it bypasses it for already existing apps like **openaiwp.**
    - You may need to manually add ClientID of **openaiwp** app to **Allowed token audiences.**
    - Check the added scope within **openaiwp** app. It should look like https://<tenant>.onmicrosoft.com/<clientid>/user_impersonation (or api://<clientid>/user_impersonation).
    - If the scope is still missing, you can configure it manually.

Azure SQL connectivity from Web API

The connection string for the Web API to access the Azure SQL database should be configured to use Managed Identity. This method is more secure than SQL-authentication that uses a username and password.

Known limitation

While utilizing Managed Identity enhances security, it may slightly decrease the performance of SQL requests. If specific performance issues arise, you might consider reverting to regular SQL authentication, although this way is less secure.

Actions

- Create System Managed Identity for App Service of Web API. This will create an Enterprise Application with the name of the service, for instance, **chatwapi**
- Add the new “contained” user into the Azure SQL database (default name dbChats), and configure its roles as shown below
  CREATE USER [**chatwapi**] FROM EXTERNAL PROVIDER;
  ALTER ROLE db_datareader ADD MEMBER [chatwapi];
  ALTER ROLE db_datawriter ADD MEMBER [chatwapi];

Deploying Web API to API Management

The quickest way to deploy this is by using the deployment wizard available in Visual Studio 2022. You will need permissions to change configurations in API Management granted via the Resource Group or similar.

- During deployment, the wizard will automatically create the necessary endpoints for the Web API and configure them.

Restrict access to Web API only for API Management service

You can restrict access to the App Service of Web API so that it only accepts requests from the API Management service.

Actions

- API Management > Overview > get Public IP
- App Service for Web API > Settings > Networking > Access restrictions > Add a rule for the APIM's public IP address. Deny other rules in both the Site and Tools tabs.

Optionally restrict access to Azure OpenAI for API Management service

This step is optional and not typically recommended.

- This setting may interfere with the standard usage of Azure OpenAI, such as through the Azure OpenAI Studio, etc.
- Anyway, access to the Azure OpenAI service API requires an api-key, which should be injected into the request by the API Management service.

Authenticated access to API Management endpoints

You should ensure that access from SPFx to API Management endpoints is secure and limited to authenticated users only.

Actions

- Avoid using the default AD provider for API Management. The default authentication should be set to "None".
- Add root-configurations to require the validation of JWT-tokens to API-endpoints as detailed in the supplemental document.

    - openai: /chat for GPT-3.5
    - openai4: /chat for GPT-4
    - openainative: /chat for native OpenAI models
    - chatwebapi: /api/<deployed methods for WebApi>

Optional restrictions for virtual networks

More advanced access restrictions can be implemented after the entire setup has been successfully deployed and tested.

- By using this approach, you can avoid unnecessary delays and complications related to access denials.