

Index

[1. Configuring Azure Open AI: 3](#_Toc171759633)

[Configurating Open AI in Azure 3](#_Toc171759634)

[Deploying the model 4](#_Toc171759635)

[2. Configuring Power BI: 6](#_Toc171759636)

[Importing data from Dataverse 6](#_Toc171759637)

[Importing a Date Table 6](#_Toc171759638)

[Loading in Post Code data 9](#_Toc171759639)

[Connecting to the weather API 10](#_Toc171759640)

[Calling OpenAI from Power Query 12](#_Toc171759641)

# 1. Configuring Azure Open AI:

## Configurating Open AI in Azure

Navigate to Open AI in Azure

If you haven’t already created a dedicated Open AI Resource Group, please create a new one:

Name: rg-lewisham-openai

Instance Details:

Region: UK South

Name: Lewisham-openai-hackathon

A screenshot of a computer

Description automatically generatedPricing Tier: Standard

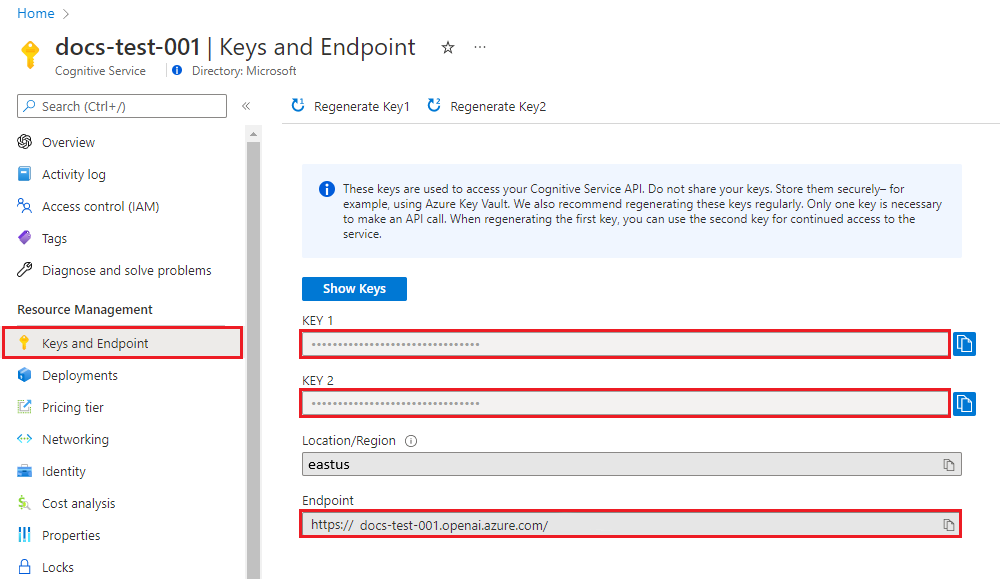
To successfully make a call against Azure OpenAI, you need an endpoint and a key.

ENDPOINT 🡪 This value can be found in the Keys & Endpoint section when examining your resource from the Azure portal. Alternatively, you can find the value in the Azure OpenAI Studio > Playground > Code View. An

example endpoint is: https://docs-test-001.openai.azure.com/.

API-KEY 🡪 This value can be found in the Keys & Endpoint section when examining your resource from the Azure portal. You can use either KEY1 or KEY2.

Go to your resource in the Azure portal. The Keys & Endpoint section can be found in the Resource Management section. Copy your endpoint and access key as you'll need both for authenticating your API calls. You can use either KEY1 or KEY2. Always having two keys allows you to securely rotate and regenerate keys without causing a service disruption.



## Deploying the model

Still inside of the Azure Open AI services in the Azure Portal click on Model Deployments -> Manage Deployment:

A screenshot of a computer

Description automatically generated

Create a new deployment:

Select a model: gpt-4

Model Version: 1106-Preview

Deployment type: Standard

Deployment name: OpenAI-Model-Hackathon-GPT4

Content filter: Default

A screenshot of a computer

Description automatically generated

# 2. Configuring Power BI:

## Importing data from Dataverse

You need to connect to Dataverse and retrieve the required tables:

A screenshot of a computer

Description automatically generatedIn the home tab under data select Dataverse:

A screenshot of a computer

Description automatically generatedThen login and search for the required table(s). Once you have selected the desired table(s) click Transform Data:

We will then review this data and apply some renaming and cleansing steps if required.

## Importing a Date Table

A date table is needed to perform historic analysis

A screenshot of a computer

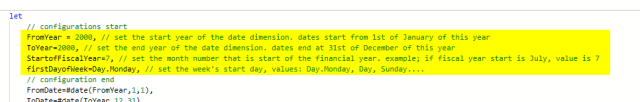
Description automatically generatedCreate a new blank query in Power BI:

A screenshot of a computer

Description automatically generatedIn the Power Query Editor window, go to View tab, and click on Advanced Editor

A screenshot of a computer program

Description automatically generatedCopy and Paste the *Date\_Dimension\_Script* here (replace the existing text in the Advanced Editor window:

You need to configure the Date table based on your need, the first few lines are the configurations that you can set based on your need and then press done;

A screenshot of a computer

Description automatically generatedNext rename the table to Date Table and press Close & Apply:

You will notice it has automatically created a date hierarchy which we can use later for drill downs.

You then want to create a join between your sample data and the date table be creating a Many to One join on the date columns:

A screenshot of a computer

Description automatically generated

## Loading in Post Code data

Download the London Postcode file and save it to your local storage on your pc.

Under Get data select CSV:

A screenshot of a computer

Description automatically generated

Then transform the data and apply the same name changes and cleansing steps if required. Rename it to London Postcode table and load it into the report.

Next you will need to create a join between the sample data and the postcode by creating a many to one join from your sample data to the Post Code table on the Post Code columns:

A screenshot of a computer

Description automatically generated

## Connecting to the weather API

This section provides an alternative method to call the weather API directly in Power BI

You will load in the Weather API in a similar method to the Date table.

A screenshot of a computer

Description automatically generatedCreate a new blank query in Power BI:

A screenshot of a computer

Description automatically generatedIn the Power Query Editor window, go to View tab, and click on Advanced Editor

Copy and Paste the *Script-for-Weather-API* here (replace the existing text in the Advanced Editor window.

The *Weather\_API\_Script* calls the weather API directly and applies some transformative steps to get it into the final format

Rename the Table to Weather Table and then press Close and Apply.

You then want to create a join between your sample data and the weather table by creating a Many to One join on the date columns:

A screenshot of a computer

Description automatically generated

A list of the weather parameters can be found below which you may need to refer back to when creating the visuals:

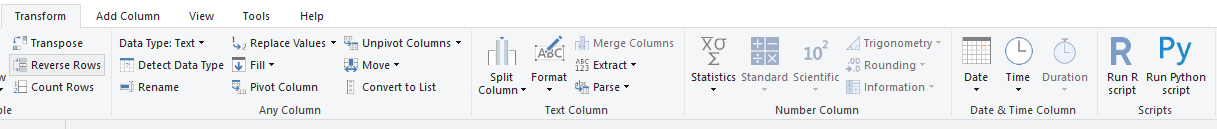
|  |  |  |
| --- | --- | --- |
| **Variable** | **Unit** | **Description** |
| weather\_code | WMO code | The most severe weather condition on a given day |
| temperature\_2m\_max | °C (°F) | Maximum and minimum daily air temperature at 2 meters above ground |
| temperature\_2m\_min | |  |
| apparent\_temperature\_max | °C (°F) | Maximum and minimum daily apparent temperature |
| apparent\_temperature\_min | | |
| precipitation\_sum | mm | Sum of daily precipitation (including rain, showers and snowfall) |
| rain\_sum | mm | Sum of daily rain |
| snowfall\_sum | cm | Sum of daily snowfall |
| precipitation\_hours | hours | The number of hours with rain |
| sunrise | iso8601 | Sun rise and set times |
| sunset |  |  |
| sunshine\_duration | seconds | The number of seconds of sunshine per day is determined by calculating direct normalized irradiance exceeding 120 W/m², following the WMO definition. Sunshine duration will consistently be less than daylight duration due to dawn and dusk. |
| daylight\_duration | seconds | Number of seconds of daylight per day |
| wind\_speed\_10m\_max | km/h (mph, m/s, knots) | Maximum wind speed and gusts on a day |
| wind\_gusts\_10m\_max | |  |
| wind\_direction\_10m\_dominant | ° | Dominant wind direction |
| shortwave\_radiation\_sum | MJ/m² | The sum of solar radiaion on a given day in Megajoules |
| et0\_fao\_evapotranspiration | mm | Daily sum of ET₀ Reference Evapotranspiration of a well watered grass field |

## Calling OpenAI from Power Query

First, install Python if you have not done so already. Please visit the official website (https://www.python.org/downloads/) to download it.

Once Python has been installed, enable Python scripting in Power BI. To do so, open Power BI Desktop. Then, click **File** > **Options and settings** and go to **Options** > **Python** **scripting**. It should automatically detect the python file location, then click **OK**.

Next press Transform data to open Power Query, navigate to the desired data set and in the Transform Tab select Run Python Script:



Paste in the *AOAI\_PBI\_Script.* You will need to modify this script to enter in your own Azure Open AI Endpoint and Key configures earlier. You will also need to change it point it at your free text field Column. You can also modify the prompt to best serve your needs for the desired outcome.

An example of what the input and output looks like can be found below:

