Labs Data & AI Innovation Day

Lab 2: Data Ingestion

This document describes how to use a Python notebook to populate the Azure Cosmos DB for NoSQL containers ‘products’ and ‘vectors’ with the content of a csv file

# Pre-requisites

Ensure that you have the following software installed on your system before proceeding with the lab:

* Visual Studio Code: A cross-platform code editor that supports Python development. You can download it from <https://code.visualstudio.com/>
* Python 3.10.11: The latest version of the Python programming language. You can download it from <https://www.python.org/downloads/release/python-31011/>

Note: If you are using a different version of Python, make sure that it is compatible with the libraries and packages used in this lab.

* Azure OpenAI account registered in the Azure subscription used for this lab

# Deploy Azure OpenAI models

Follow these steps to deploy the Azure OpenAI models (GPT 3.5 Turbo and Text-embeddingt-ada-002)

* Login to the Azure Portal
* Connect to the Azure OpenAI account
* In the left menu, select “Model deployments”
* Click “Manage deployments”
* Select the “Deployments” section
* Click on the “Create deployment” button
* Select the “text-embedding-ada-002” model in the dropdown list
* In the deployment name, type text-embedding-ada-002
* Click on the “Create” button to deploy the model

A screenshot of a computer

Description automatically generated

* Repeat the previous steps to deploy the gpt-3.5-turbo model

## Prepare Python virtual environment

Run these commands to create a Python virutal environment and install the required libraries:

1. Open the “lab1” folder in Visual Studio Code
2. Open a new Powershell Teminal > New Terminal
3. Type this command to create a virtual environment: python -m venv .venv
4. Activate the virtual environment with .venv\scripts\activate
5. Install the required libraries with pip install -r requirements.txt

A computer screen shot of a computer code

Description automatically generated

## Get Azure OpenAI key

* Connect to Azure Portal
* Connect to Azure OpenAI account
* In the left navigation menu, select “Keys and Endpoint”
* Key 1 and endpoint are the values you will need to add to your .env file

A screenshot of a computer

Description automatically generated

# Modify environment variables

Rename the “.env template” file to “.env” and modify the variables to reflect your environment

|  |  |  |
| --- | --- | --- |
| AZURE\_COSMOSDB\_NOSQL\_ENDPOINT | The name of your Cosmos DB for NoSQL | cosmos-nosql-001documents.azure.com |
| AZURE\_COSMOSDB\_NOSQL\_KEY | Cosmos DB key | xxxx |
| AZURE\_COSMOSDB\_NOSQL\_DATABASE\_NAME | Database name | Database\_teamXX |
| AZURE\_COSMOSDB\_NOSQL\_DATABASE\_NAME | Database name | database\_<team\_name>, e.g. database\_team01 |
| AZURE\_COSMOSDB\_NOSQL\_VECTORS\_CONTAINER\_NAME | Vectors container name | Vectors |
| AZURE\_OPENAI\_ENDPOINT | Azure OpenAI account url | https://<team\_name>openai.openai.azure.com/ |
| AZURE\_OPENAI\_API\_KEY | Azure OpenAI account key |  |
| AZURE\_OPENAI\_EMBEDDING\_MODEL | Name of your embedding model deployment | Defaults to text-embedding-ada-002 |
| AZURE\_OPENAI\_CHAT\_MODEL | Name of your chat model deployment | Defaults to gpt-35-turbo |
| AZURE\_OPENAI\_API\_VERSION | API version | 2024-02-01 |

Your .env file should look like this:

A computer screen shot of text

Description automatically generated

# Ingest data into Cosmos DB for NoSQL

In this section, you will use the Python notebook to upload sample data to your Azure Cosmos DB for NoSQL container

1. Download the original data files from https://cosmosdbcosmicworks.blob.core.windows.net/cosmic-works-small/product.json and https://cosmosdbcosmicworks.blob.core.windows.net/cosmic-works-small/customer.json to the lab2 folder on your local machine.
2. Open Visual Studio Code
3. Using the menu: File > Open Folder
4. Open a new Powershell Teminal > New Terminal
5. Activate the virtual environment with .venv\scripts\activate
6. In the files navigation tree, open the file “import\_cosmosdb\_nosql.ipynb”
7. Execute each cell one by one or click the “Run all” button in the toolbar to run all cells

At the end of the process, there should be 295 products in the products collection and 1584 customers in the customers collection

A screen shot of a computer

Description automatically generated

# Check ingested data

A screenshot of a computer

Description automatically generated

There should also be data in the vectors container

A screenshot of a computer

Description automatically generated