4.10 - Perform vector search and retrieval in Azure AI Search

Overview

Learn how to perform vector search and retrieval in Azure AI Search.

By the end of this module, you'll learn how to:

- Describe vector search
- Describe embeddings
- Run vector search queries using the REST API

Introduction

Suppose that you work for a company that is developing an app for an online retail store. You want to return better results for your application users through Azure AI Search. In this module, you'll learn how to use vector search in Azure AI Search to achieve this goal.

By the end of this module, you'll learn how to:

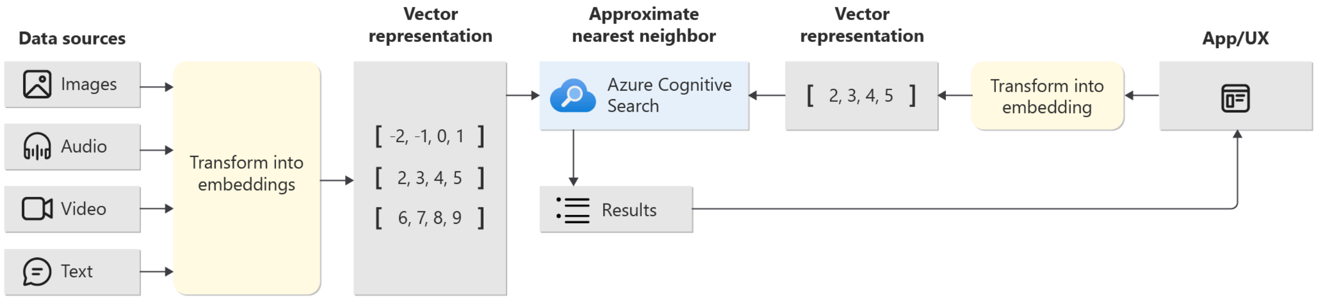
- Describe vector search
- Describe embeddings

- Run vector search queries using the REST API

What is vector search?

Vector search is a capability available in AI Search used to index, store and retrieve vector embedding from a search index. You can use it to power applications implementing the Retrieval Augmented Generation (RAG) architecture, similarity and multi-modal searches or recommendation engines.

Below is an illustration of the indexing and query workflows for vector search.



A vector query can be used to match criteria across different types of source data by providing a mathematical representation of the content generated by machine learning models. **This eliminates the limitations of text based searches returning relevant results by using the intent of the query.**

When to use vector search

Here are some scenarios where you should use vector search:

- Use OpenAI or open source models to **encode text**, and use queries encoded as vectors to retrieve documents.
- Do a **similarity search** across encoded images, text, video and audio, or a mixture of these (multi-modal).
- Represent documents in different languages using a multi-lingual embedded model to find documents in any language.
- Build **hybrid searches from vector and searchable text fields** as vector searches are implemented at field level. The results will be merged to return a single response.
- Apply filters to text and numeric fields and include this in your query to reduce the data that your vector search needs to process.
- Create a vector database to provide an external knowledge base or use as a long term memory.

Limitations

There are a few limitations when using vector search, which you should note:

- You'll need to provide the embeddings using Azure OpenAI or a similar open source solution, as **Azure AI Search doesn't generate these for your content.**
- Customer Managed Keys (CMK) aren't supported.
- There are storage limitations applicable so you should check what your service quota provides.

> Note: If your documents are large, you consider chunking. Use the [Chunking large documents for vector search solutions in AI Search] documentation for more information.

Prepare your search

You need to encode your Azure AI Search query by sending it to an embedded model. The response is then passed to a search engine to complete a search over the vector fields.

In order for your query to work, you need do the following tasks:

Check your index has vector fields

You check if your search has **vector fields** by running an empty search, the result includes **a vector field with a number array.**

You can also look for a field named **vectorSearch** with the type **Collection(Edm.single)**. This has an algorithm configuration and an attribute of 'dimension'.

Convert a query input into a vector

You can only query a vector field with a query vector. Your end-users provide a text query string, which your application converts into a vector by using the embedding library you used to create the source document embeddings.

Understand embedding

An embedding is a type of data representation that is used by machine learning models. An embedding represents the semantic meaning of a piece of text.

You can visualize an embedding as an array of numbers, and the numerical distance between two embeddings represents their semantic similarity. For example, if two texts are similar, then their representations should also be similar.

Embedding models

How effective your search results will be is a direct correlation to the effectiveness of your embedded model.

There are models specifically created to perform a specific task well. Use **Similarity** search embeddings to capture the semantic similarity between pieces of text; a **Text** search embedding can look at the relevance of a long document to a short query; use embedding code snippets and natural language search queries using a **Code** search embedding.

Users provide input to a query for an embedding model which is converted from text into a vector using, for example, the **text-embedding-ada-002** model to generate text embeddings.

The result will be any documents matching the query that are contained in your search index. The documents, with embeddings containing vector fields, must exist in the search index and **the same model must be used for indexing and the query.**

Embedding space

Embedding space is the core of vector queries comprising all the vector fields from the same embedding model. It comprises of all the vector fields populated using the same model.

In this embedding space, similar items are located close together, and dissimilar items are located farther apart.

For example, documents that talk about hotels with a water park would be close together in the embedding space, whereas, hotels without this facility would be farther away whilst still being in the neighborhood for hotels. Dissimilar concepts such as restaurants would be farther away still.

In practice, embedding spaces are abstract and don't have well-defined meanings comprehensible be people, but the core idea stays the same.

Exercise - Use the REST API to run vector search queries

In this exercise you'll set up your project, create an index, upload your documents, and run queries.

You'll need the following to successfully this exercise:

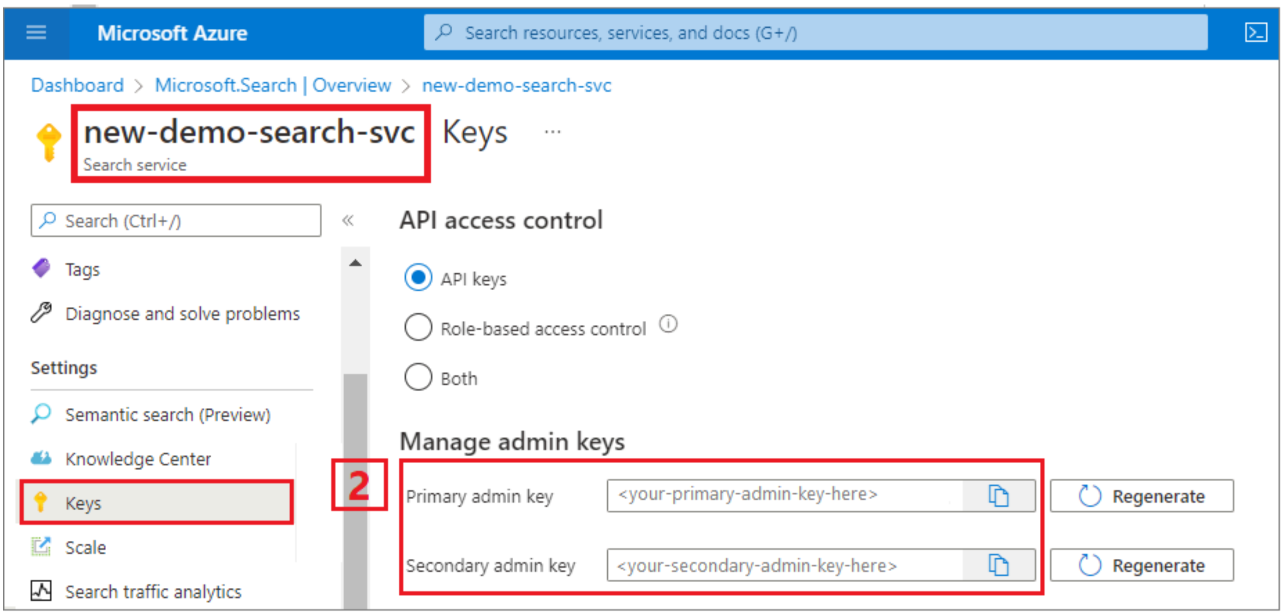
- The Postman app
- An Azure subscription
- Azure AI Search service
- The Postman sample collection located in this repository - *Vector-Search-Quickstart.postman_collection v1.0 json*.

> **Note** You can find more information about the Postman app here if required.

Set up your project

First set up your project by carrying out the following steps:

1. Note the **URL** and **Key** from your Azure AI Search service.



Microsoft Azure
Search resources, services, and docs (G+/)
Dashboard > Microsoft.Search | Overview > new-demo-search-svc
new-demo-search-svc Keys ...
Search service
Search (Ctrl+/)
Tags
Diagnose and solve problems
Settings
Semantic search (Preview)
Knowledge Center
Keys
Scale
Search traffic analytics
2
Regenerate
Regenerate

API access control

● API keys

○ Role-based access control

○ Both

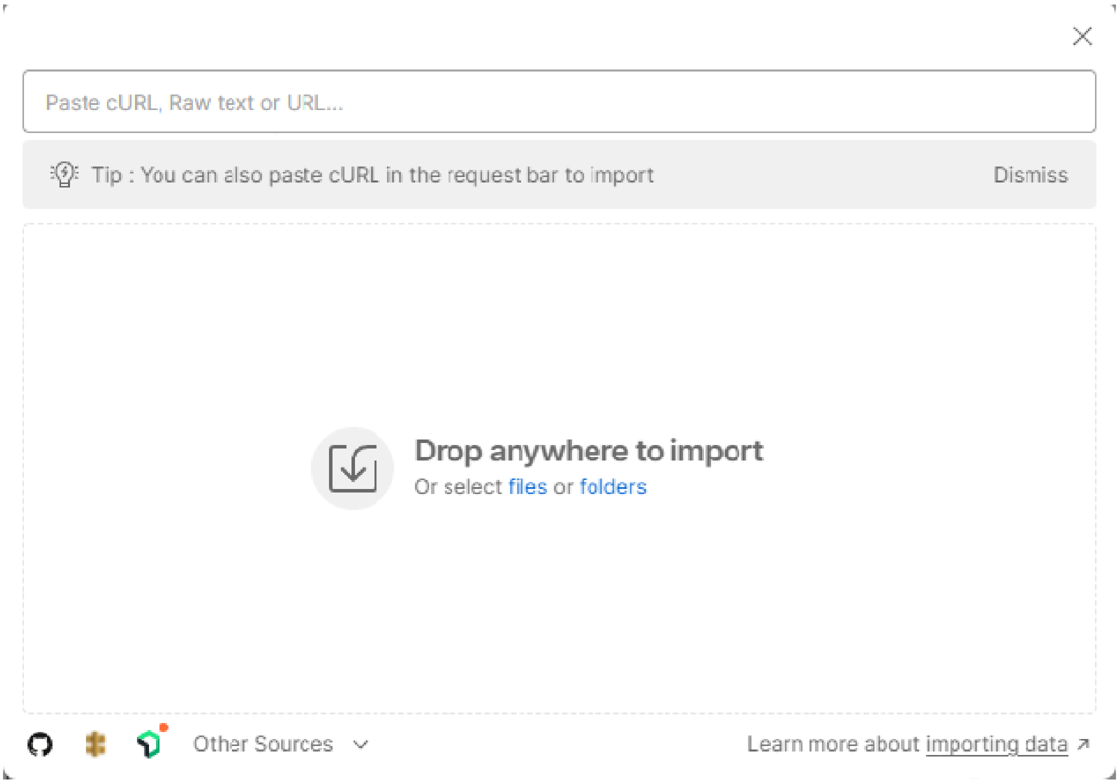
Manage admin keys

Primary admin key <your-primary-admin-key-here>

Secondary admin key <your-secondary-admin-key-here>

2. Download the Postman sample collection.

3. Open Postman and import the collection by selecting the **Import** button and drag and drop the collection folder into the box.



Paste cURL, Raw text or URL...

Tip : You can also paste cURL in the request bar to import

Dismiss

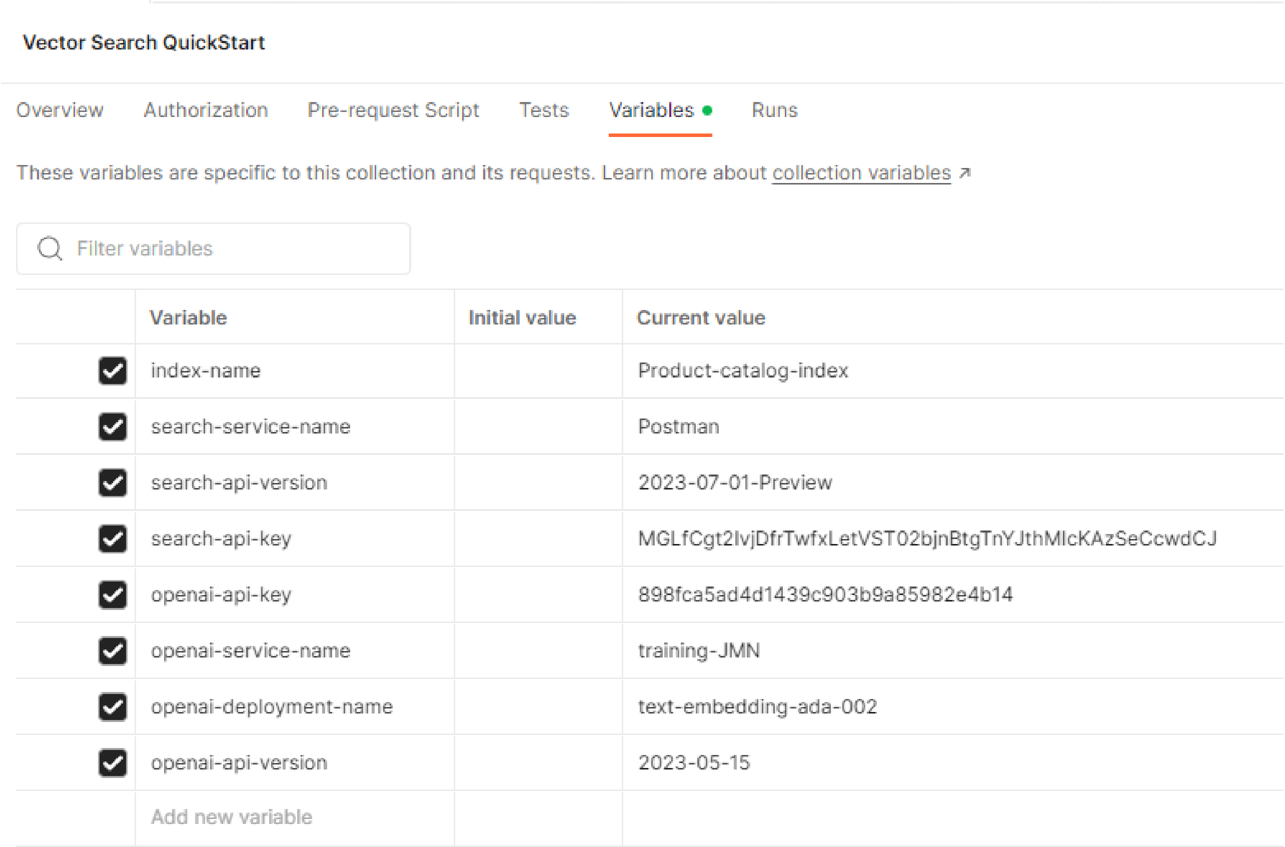
Drop anywhere to import

Or select files or folders

Other Sources

Learn more about importing data ↗

4. Select the **Fork** button to create a fork of the collection and add a unique name.
5. Right-click your collection name and select **Edit**.
6. Select the **Variables** tab and enter the following values using the search service and index names from your Azure AI Search service:



Vector Search QuickStart

These variables are specific to this collection and its requests. Learn more about collection variables ↗

Filter variables

	Variable	Initial value	Current value
☑	index-name		Product-catalog-index
☑	search-service-name		Postman
☑	search-api-version		2023-07-01-Preview
☑	search-api-key		MGLfCgt2IvjDfrTwfxLetVST02bjnBtgTnYJthMIcKAzSeCcwdCJ
☑	openai-api-key		898fca5ad4d1439c903b9a85982e4b14
☑	openai-service-name		training-JMN
☑	openai-deployment-name		text-embedding-ada-002
☑	openai-api-version		2023-05-15
	Add new variable		

7. Save your changes by selecting the **Save** button.

You’re ready to send your requests to the Azure AI Search service.

Create an Index

Next, create your index in Postman:

1. Select **PUT Create/Update Index** from the side menu.
2. Update the URL with your **search-service-name**, **index-name** and **api-version** that you noted earlier.
3. Select the **Body** tab to see the response.
4. Set the **index-name** with your index name value from your URL and select **Send**.

You should see a status code of type **200** which indicates a successful request.

Upload Documents

There are 108 documents included in the Upload Documents request, each one has a full set of embeddings for the **titleVector** and **contentVector** fields.

1. Select **POST Upload Docs** from the side menu.
2. Update the URL with your **search-service-name**, **index-name** and **api-version** as before.
3. Select the **Body** tab to see the response and select **Send**.

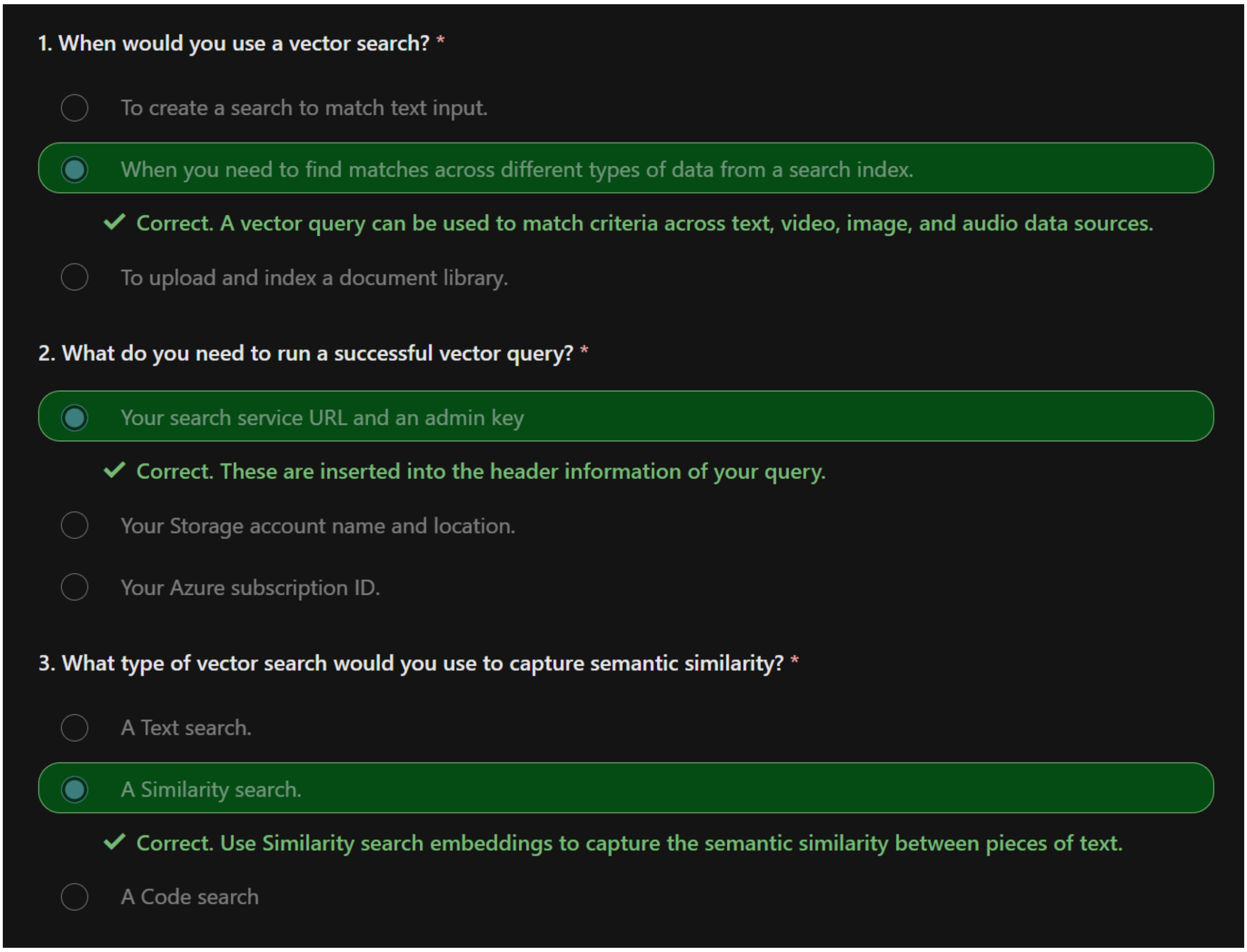
You should see a status code of type **200** to show that your request was successful.

Run Queries

1. Now try running the following queries on the side menu. To do this, make sure to update the URL each time as before and send a request by selecting **Send**:
   - Single Vector search
   - Single Vector search w/Filter
   - Simple hybrid search
   - Simple hybrid search w/Filter
   - Cross-field search
   - Multi-Query search
2. Select the **Body** tab to see the response and view the results.

You should see a status code of type **200** for a successful request.

Knowledge Check



1. When would you use a vector search? *

○ To create a search to match text input.

● When you need to find matches across different types of data from a search index.

✓ Correct. A vector query can be used to match criteria across text, video, image, and audio data sources.

○ To upload and index a document library.

2. What do you need to run a successful vector query? *

● Your search service URL and an admin key

✓ Correct. These are inserted into the header information of your query.

○ Your Storage account name and location.

○ Your Azure subscription ID.

3. What type of vector search would you use to capture semantic similarity? *

○ A Text search.

● A Similarity search.

✓ Correct. Use Similarity search embeddings to capture the semantic similarity between pieces of text.

○ A Code search

Summary

Vector search enables you to index, store, and retrieve vector embeddings from a search index. You learned how to use vector search to facilitate better search results across your queries.

Now that you have completed this module, you know how to:

- Describe vector search
- Describe embeddings and vectorization
- Run vector search queries using the REST API

✍️ Compiled by Kenneth Leung (2025)