AI-102-24: AI-102 ラボ 24 (ナレッジストア)

1 時間 59 分 残り

指示 リソース ヘルプ  100%

# 開始する前に

**ヒント**: このウィンドウの指示に従って、アイコンが表示されるたびに、それを使用して指示ペインから仮想マシンのインターフェイスにテキストをコピーできます。これは特にコードをコピーする場合に便利です。ただし、実行する前に、インデントレベルや書式を修正するために、貼り付けたコードを修正する必要がある場合があります。

1. プロンプトが表示されたら、パスワード Pa55w.rd を使用して **Student** アカウントにログインします。PC を検出できるように求めるメッセージが表示されたら、[いいえ] を選択**します**。
2. Azure サブスクリプションをまだお持ちでない場合は、https://azure.microsoft.com/free/ で無料試用版にサインアップ[してください](https://azure.microsoft.com/free/)。

# Azure の認知検索を使用してナレッジ ストアを作成する

Azure Cognitive Search では、認知スキルのエンリッチメント パイプラインを使用して、AI で生成されたフィールドをドキュメントから抽出し、検索インデックスに含めます。インデックスはインデックス作成プロセスのプライマリ出力と見なされる場合がありますが、インデックスに含まれるエンリッチされたデータは他の方法でも役立ちます。例えば：

* インデックスは基本的に JSON オブジェクトのコレクションであり、それぞれがインデックス付きのレコードを表すため、Azure Data Factory などのツールを使用してデータ オーケストレーション プロセスに統合するために、オブジェクトを JSON ファイルとしてエクスポートすると便利な場合があります。
* Microsoft Power BI などのツールを使用して分析やレポートを作成するために、インデックス レコードをテーブルのリレーショナル スキーマに正規化する場合があります。
* インデックス作成プロセス中にドキュメントから埋め込みイメージを抽出した場合、これらのイメージをファイルとして保存する必要がある場合があります。

この演習では、パンフレットやホテルレビューの情報を使用して顧客が旅行を計画する際に役立つ架空の旅行代理店である Margie's Travel のナレッジ ストアを実装します。

## このコースのリポジトリのクローンを作成する

**AI-102-AIEngineer** コード リポジトリを、このラボで作業している環境に既にクローンしている場合は、Visual Studio コードで開きます。それ以外の場合は、次の手順に従って、今すぐ複製します。

1. ビジュアル スタジオ コードを起動します。
2. パレットを開き (SHIFT + Ctrl + P) し、**Git: Clone** コマンドを実行してリポジトリをローカル フォルダーに複製します (どのフォルダーにクローンを作成しても構いません)。https://github.com/MicrosoftLearning/AI-102-AIEngineer
3. リポジトリが複製されたら、Visual Studio コードでフォルダーを開きます。
4. 追加のファイルがインストールされ、Repo の C# コード プロジェクトをサポートするまで待ちます。

**注**: ビルドおよびデバッグに必要なアセットを追加するよう求められた場合は、[**今は使用しない**] を選択します。

## Azure リソースの作成

**注**: 以前に Azure [**Cognitive Search ソリューションの作成**](https://raw.githubusercontent.com/MicrosoftLearning/AI-102-AIEngineer/master/Instructions/22-azure-search.md)の演習を完了し、サブスクリプションにこれらの Azure リソースを含めている場合は、このセクションをスキップして**「検索ソリューションの作成**」セクションから開始できます。それ以外の場合は、次の手順に従って、必要な Azure リソースをプロビジョニングします。

1. Web ブラウザーで、Azure ポータル を開き、Azure サブスクリプションに関連付けられている Microsoft アカウントを使用してサインインします。https://portal.azure.com
2. サブスクリプションの**リソース グループ**を表示します。
3. リソース グループが提供されている制限付きサブスクリプションを使用している場合は、リソース グループを選択してプロパティを表示します。それ以外の場合は、任意の名前で新しいリソース グループを作成し、作成後に移動します。
4. リソース グループの [**概要**] ページで、[**サブスクリプション ID**] と **[場所**] をメモします。これらの値は、以降の手順でリソース グループの名前と共に必要になります。
5. Visual Studio コードで、**24 ナレッジ ストア** フォルダーを展開し、**setup.cmd** を選択します。このバッチ スクリプトを使用して、必要な Azure リソースを作成するために必要な Azure コマンド ライン インターフェイス (CLI) コマンドを実行します。
6. **24 ナレッジストア**フォルダを右クリックし、[**統合端末で開く**]を選択します。
7. ターミナル ウィンドウで、次のコマンドを入力して、Azure サブスクリプションへの認証済み接続を確立します。

az login --output none

1. プロンプトが表示されたら、Azure サブスクリプションにサインインします。次に、Visual Studio コードに戻り、サインイン プロセスが完了するまで待ちます。
2. 次のコマンドを実行して、Azure の場所を一覧表示します。

az account list-locations -o table

1. 出力で、リソース グループの場所に対応する **Name** 値を見つけます (たとえば、米国東部の場合は、対応する名前は eastus です)。
2. **setup.cmd** スクリプトで、**subscription\_id**、**resource\_group**、および**場所**の変数宣言を、サブスクリプション ID、リソース グループ名、および場所名に適切な値で変更します。その後、変更を保存します。
3. **24-knowledge-store** フォルダのターミナルで、次のコマンドを入力してスクリプトを実行します。

setup

**注**: 検索 CLI モジュールはプレビュー段階にあり、-実行中の .. プロセスでスタックする可能性があります。この現象が 2 分を超える場合は、Ctrl + C キーを押して実行時間の長い操作をキャンセルし、スクリプトを終了するかどうか尋ねられたときに **N** を選択します。その後、正常に完了するはずです。

スクリプトが失敗した場合は、正しい変数名を使用して保存したことを確認してから、もう一度やり直してください。

1. スクリプトが完了したら、表示される出力を確認し、Azure リソースに関する次の情報を確認します (これらの値は後で必要になります)。
   * ストレージ アカウント名
   * ストレージ接続文字列
   * コグニティブ サービス アカウント
   * コグニティブ サービスキー
   * 検索サービス エンドポイント
   * 検索サービス管理キー
   * 検索サービス クエリ キー
2. Azure ポータルで、リソース グループを更新し、Azure ストレージ アカウント、Azure コグニティブ サービス リソース、および Azure コグニティブ 検索リソースが含まれていることを確認します。

## 検索ソリューションを作成する

必要な Azure リソースを用意したので、次のコンポーネントで構成される検索ソリューションを作成できます。

* Azure ストレージ コンテナー内のドキュメントを参照する**データ ソース**。
* AI 生成フィールドをドキュメントから抽出するスキルのエンリッチメント パイプラインを定義する**スキルセット**。スキルセットは、ナレッジストアで生成される予測も定義します。
* 検索可能なドキュメント レコードのセットを定義する**インデックス**。
* データ ソースからドキュメントを抽出し、スキルセットを適用し、インデックスを設定する**インデクサー**。また、インデックス作成プロセスでは、ナレッジ ストアのスキルセットで定義されたプロジェクションも永続化されます。

この演習では、Azure コグニティブ検索 REST インターフェイスを使用して、JSON 要求を送信してこれらのコンポーネントを作成します。

### REST 操作用の JSON の準備

REST インターフェイスを使用して、Azure コグニティブ検索コンポーネントの JSON 定義を送信します。

1. Visual Studio コードの **24 ナレッジ ストア** フォルダーで、検索フォルダーの**作成**を展開し、**data\_source.json** を選択します。このファイルには、**margies-knowledge-data** という名前のデータソースの JSON 定義が含まれています。
2. **YOUR\_CONNECTION\_STRING**のプレースホルダーを、次のような Azure ストレージ アカウントの接続文字列に置き換えます。

DefaultEndpointsProtocol=https;AccountName=ai102str123;AccountKey=12345abcdefg...==;EndpointSuffix=core.windows.net

接続文字列は、Azure ポータルのストレージ アカウントの ***[アクセス キー***] ページで確認できます。

1. 更新された JSON ファイルを保存して閉じます。
2. 検索フォルダの**作成**で、**skillset.json を開きます**。このファイルには、**margies-knowledge-skillset という名前のスキルセットの** JSON 定義が含まれています。
3. スキルセット定義の上部にある **cognitiveServices** 要素で、**YOUR\_COGNITIVE\_SERVICES\_KEY**プレースホルダーをコグニティブ サービス リソースのキーのどちらかに置き換えます。

キーは、Azure ポータルのコグニティブ サービス リソースの ***[キーとエンドポイント***] ページで確認できます。

1. スキルセットのスキルコレクションの最後に、**define-projection** という名前の **Microsoft.Skills.Util.ShaperSkill** スキルを見つけます。このスキルは、インデクサーによって処理される各ドキュメントのナレッジ ストアに対してパイプラインが保持する予測に使用される、エンリッチデータの JSON 構造を定義します。
2. スキルセット ファイルの下部には、ナレッジ ストアを作成する Azure Storage アカウントの接続文字列と**プロジェクション**のコレクションを含む **knowledgeStore** 定義もスキルセットに含まれていることを確認します。このスキルセットには、次の 3 つのプロジェクション グループが含まれます。
   * スキルセット内のシェイパースキルの**knowledge\_projection**出力に基づくオブジェクトプロジェクションを含むグループ。
   * ドキュメントから抽出されたイメージ データの**normalized\_images**コレクションに基づくファイル プロジェクションを含むグループ。
   * 次のテーブルの投影を含むグループ。
     + **キーフレーズ**: 自動的に生成されたキー列と、シェイパー スキルの**knowledge\_projection/key\_phrases/**コレクション出力にマップされた**キーフレーズ**列が含まれます。
     + **場所**: 自動的に生成されたキー列と、シェイパー スキルの**knowledge\_projection/key\_phrases/**コレクション出力にマップされた**場所**列が含まれます。
     + **ImageTags**: 自動的に生成されたキー列と、シェイパー スキルの**knowledge\_projection/image\_tags/**コレクション出力にマップされた**タグ**列が含まれます。
     + **ドキュメント**: 自動生成されたキー列と、まだテーブルに割り当てられていないシェイパー スキルの**knowledge\_projection**出力値がすべて含まれます。
3. ストレージ アカウントの**接続文字列で、ストレージ接続文字列**値の**YOUR\_CONNECTION\_STRING**のプレースホルダーを置き換えます。
4. 更新された JSON ファイルを保存して閉じます。
5. 検索フォルダの**作成**で、**index.json を**開きます。このファイルには、**margies-knowledge-index** という名前のインデックスの JSON 定義が含まれています。
6. インデックスの JSON を確認し、変更を加えずにファイルを閉じます。
7. 検索フォルダの**作成**で、**indexer.json を開きます**。このファイルには、**margies-knowledge-indexer** という名前のインデクサーの JSON 定義が含まれています。
8. インデクサーの JSON を確認し、変更を加えずにファイルを閉じます。

### REST 要求の送信

検索ソリューションコンポーネントを定義する JSON オブジェクトを準備したので、REST インターフェイスに JSON ドキュメントを送信して作成できます。

1. 検索フォルダの**作成**で、**create-search.cmd** を開きます。このバッチ スクリプトでは、cURL ユーティリティを使用して、Azure コグニティブ検索リソースの REST インターフェイスに JSON 定義を送信します。
2. **YOUR\_SEARCH\_URL**と**YOUR\_ADMIN\_KEY**変数のプレースホルダーを**、Azure** Cognitive Search リソースの URL と**いずれかの管理キー**に置き換えます。

これらの値は、Azure ポータルの Azure コグニティブ検索リソースの ***[概要***] ページと ***[キー***] ページで確認できます。

1. 更新されたバッチ ファイルを保存します。
2. **検索フォルダ**を右クリックし、[**統合ターミナルで開く**] を選択します。
3. 検索フォルダの**作成**のターミナル ペインで、次のコマンドを入力してバッチ スクリプトを実行します。

create-search

1. スクリプトが完了したら、Azure ポータルの Azure Cognitive Search リソースのページで[**インデクサー**] ページを選択し、インデックス作成プロセスが完了するまで待ちます。

[***更新]*** を選択すると、インデックス作成操作の進行状況を追跡できます。完了するまでに 1 分ほどかかる場合があります。

**ヒント**: スクリプトが失敗した場合は、**data\_source.json** ファイルおよび **skillset.json** ファイルに追加したプレースホルダーと **create-search.cmd** ファイルを確認してください。間違いを修正した後、スクリプトを再実行する前に、Azure Portal ユーザー インターフェイスを使用して検索リソースに作成されたコンポーネントを削除する必要がある場合があります。

## ナレッジ ストアの表示

スキルセットを使用してナレッジ ストアを作成するインデクサーを実行すると、インデックス作成プロセスによって抽出されたエンリッチデータがナレッジ ストア予測に保持されます。

### オブジェクト投影の表示

マーギーの Travel スキルセットで定義されたオブジェクトの投影は、インデックス付きのドキュメントごとに JSON ファイルで構成されます。これらのファイルは、スキルセット定義で指定された Azure Storage アカウントの BLOB コンテナーに格納されます。

1. Azure ポータルで、以前に作成した Azure ストレージ アカウントを表示します。
2. ストレージ **エクスプローラー** タブ (左側のウィンドウ) を選択して、Azure Portal のストレージ エクスプローラー インターフェイスでストレージ アカウントを表示します。
3. **BLOB コンテナー**を展開して、ストレージ アカウント内のコンテナーを表示します。ソースデータが格納されている**マーギー**コンテナに加えて、**マーギー画像**と**マーギー知識**の2つの新しいコンテナが必要です。これらは、インデックス作成プロセスによって作成されました。
4. **マーギー知識コンテナを**選択します。インデックス付きのドキュメントごとにフォルダを含める必要があります。
5. いずれかのフォルダーを開き、そのフォルダーに含まれる **knowledge-projection.json** ファイルをダウンロードして開きます。各 JSON ファイルには、次に示すようにスキルセットによって抽出されたエンリッチデータを含む、インデックス付きドキュメントの表現が含まれています。

{

"file\_id":"abcd1234....",

"file\_name":"Margies Travel Company Info.pdf",

"url":"https://store....blob.core.windows.net/margies/...pdf",

"language":"en",

"sentiment":0.83164644241333008,

"key\_phrases":[

"Margie’s Travel",

"Margie's Travel",

"best travel experts",

"world-leading travel agency",

"international reach"

],

"locations":[

"Dubai",

"Las Vegas",

"London",

"New York",

"San Francisco"

],

"image\_tags":[

"outdoor",

"tree",

"plant",

"palm"

]

}

このようなオブジェクト予測を作成する機能により、JSON ファイルを Azure Data Factory パイプラインに取り込んでデータ ウェアハウスに読み込んだりするなど、エンリッチデータ オブジェクトを生成して、エンタープライズ データ分析ソリューションに組み込むことができます。

### ファイルの投影を表示する

スキルセットで定義されたファイルプロジェクションは、インデックス作成プロセス中にドキュメントから抽出された各画像に対して JPEG ファイルを作成します。

1. Azure ポータルのストレージ エクスプローラー インターフェイスで、**マーギーズ イメージ** BLOB コンテナーを選択します。このコンテナには、イメージを含むドキュメントごとにフォルダが含まれています。
2. フォルダを開き、その内容を表示します - 各フォルダには少なくとも1つの\*.jpgファイルが含まれています。
3. 画像ファイルを開いて、ドキュメントから抽出したイメージが含まれていることを確認します。

このようなファイル投影を生成する機能により、インデックス作成は、大量のドキュメントから埋め込み画像を効率的に抽出できます。

### テーブルの投影を表示する

スキルセットで定義されたテーブルプロジェクションは、エンリッチされたデータのリレーショナルスキーマを形成します。

1. Azure ポータルのストレージ エクスプローラー インターフェイスで、[**TABLES]** を展開します。
2. **[ドキュメント] テーブルを**選択して、その列を表示します。列には、いくつかの標準的な Azure Storage テーブルの列が含まれています - これらを非表示にするには、**列のオプション**を変更して、次の列のみを選択します。
   * **document\_id** (インデックス作成プロセスによって自動的に生成されるキー列)
   * **file\_id** (エンコードされたファイル URL)
   * **file\_name** (ドキュメントメタデータから抽出されたファイル名)
   * **言語** (文書が書かれている言語)
   * ドキュメントに対して計算されたセンチメント スコアを**センチメント**します。
   * Azure ストレージ内のドキュメント BLOB の URL を **url に指定**します。
3. インデックス作成プロセスによって作成された他のテーブルを表示します。
   * **ImageTags** (タグが表示されるドキュメントの**document\_id**を含む個々のイメージ タグの行が含まれます)。
   * **キーフレーズ** (フレーズが表示されるドキュメントの**document\_id**を含む個々のキー フレーズの行を含む)。
   * **場所** (場所が表示されるドキュメントの**document\_id**を含む個々の場所の行が含まれます)。

テーブルの投影を作成する機能により、リレーショナル スキーマを照会する分析およびレポート ソリューションを構築できます。たとえば、マイクロソフトの電源の BI を使用します。自動生成されたキー列を使用して、クエリ内のテーブルを結合できます。

## 詳細

Azure コグニティブ 検索を使用したナレッジ ストアの作成の詳細については、[Azure コグニティブ 検索のドキュメント](https://docs.microsoft.com/azure/search/knowledge-store-concept-intro)を参照してください。

## マイクロソフトに戻る 学び

演習が完了したら、[マイクロソフトのラーニングに戻り](https://docs.microsoft.com/learn/modules/create-knowledge-store-azure-cognitive-search/9-knowledge-check)、この章を完了するための知識チェックを完了し、ポイントを獲得します。)

AI-102-24: AI-102 Lab 24 (Knowledge Store)

1 Hr 59 Min Remaining

Instructions Resources Help  100%

# Before You Start

**Tip**: As you follow the instructions in this pane, whenever you see a icon, you can use it to copy text from the instruction pane into the virtual machine interface. This is particularly useful to copy code; but bear in mind you may need to modify the pasted code to fix indent levels or formatting before running it!

1. If prompted, log into the **Student** account with the password Pa55w.rd. If prompted to allow your PC to be discoverable, select **No**.
2. If you do not already have an Azure subscription, sign up for a free trial at <https://azure.microsoft.com/free/>.

# Create a Knowledge Store with Azure Cognitive Search

Azure Cognitive Search uses an enrichment pipeline of cognitive skills to extract AI-generated fields from documents and include them in a search index. While the index might be considered the primary output from an indexing process, the enriched data it contains might also be useful in other ways. For example:

* Since the index is essentially a collection of JSON objects, each representing an indexed record, it might be useful to export the objects as JSON files for integration into a data orchestration process using tools such as Azure Data Factory.
* You may want to normalize the index records into a relational schema of tables for analysis and reporting with tools such as Microsoft Power BI.
* Having extracted embedded images from documents during the indexing process, you might want to save those images as files.

In this exercise, you'll implement a knowledge store for Margie's Travel, a fictitious travel agency that uses information in brochures and hotel reviews to help customers plan trips.

## Clone the repository for this course

If you have already cloned **AI-102-AIEngineer** code repository to the environment where you're working on this lab, open it in Visual Studio Code; otherwise, follow these steps to clone it now.

1. Start Visual Studio Code.
2. Open the palette (SHIFT+CTRL+P) and run a **Git: Clone** command to clone the repository to a local folder (it doesn't matter which folder).https://github.com/MicrosoftLearning/AI-102-AIEngineer
3. When the repository has been cloned, open the folder in Visual Studio Code.
4. Wait while additional files are installed to support the C# code projects in the repo.

**Note**: If you are prompted to add required assets to build and debug, select **Not Now**.

## Create Azure resources

**Note**: If you have previously completed the [**Create an Azure Cognitive Search solution**](https://raw.githubusercontent.com/MicrosoftLearning/AI-102-AIEngineer/master/Instructions/22-azure-search.md) exercise, and still have these Azure resources in your subscription, you can skip this section and start at the **Create a search solution** section. Otherwise, follow the steps below to provision the required Azure resources.

1. In a web browser, open the Azure portal at , and sign in using the Microsoft account associated with your Azure subscription.https://portal.azure.com
2. View the **Resource groups** in your subscription.
3. If you are using a restricted subscription in which a resource group has been provided for you, select the resource group to view its properties. Otherwise, create a new resource group with a name of your choice, and go to it when it has been created.
4. On the **Overview** page for your resource group, note the **Subscription ID** and **Location**. You will need these values, along with the name of the resource group in subsequent steps.
5. In Visual Studio Code, expand the **24-knowledge-store** folder and select **setup.cmd**. You will use this batch script to run the Azure command line interface (CLI) commands required to create the Azure resources you need.
6. Right-click the the **24-knowledge-store** folder and select **Open in Integrated Terminal**.
7. In the terminal pane, enter the following command to establish an authenticated connection to your Azure subscription.

az login --output none

1. When prompted, sign into your Azure subscription. Then return to Visual Studio Code and wait for the sign-in process to complete.
2. Run the following command to list Azure locations.

az account list-locations -o table

1. In the output, find the **Name** value that corresponds with the location of your resource group (for example, for East US the corresponding name is eastus).
2. In the **setup.cmd** script, modify the **subscription\_id**, **resource\_group**, and **location** variable declarations with the appropriate values for your subscription ID, resource group name, and location name. Then save your changes.
3. In the terminal for the **24-knowledge-store** folder, enter the following command to run the script:

setup

**Note**: The Search CLI module is in preview, and may get stuck in the - Running .. process. If this happens for over 2 minutes, press CTRL+C to cancel the long-running operation, and then select **N** when asked if you want to terminate the script. It should then complete successfully.

If the script fails, ensure you saved it with the correct variable names and try again.

1. When the script completes, review the output it displays and note the following information about your Azure resources (you will need these values later):
   * Storage account name
   * Storage connection string
   * Cognitive Services account
   * Cognitive Services key
   * Search service endpoint
   * Search service admin key
   * Search service query key
2. In the Azure portal, refresh the resource group and verify that it contains the Azure Storage account, Azure Cognitive Services resource, and Azure Cognitive Search resource.

## Create a search solution

Now that you have the necessary Azure resources, you can create a search solution that consists of the following components:

* A **data source** that references the documents in your Azure storage container.
* A **skillset** that defines an enrichment pipeline of skills to extract AI-generated fields from the documents. The skillset also defines the projections that will be generated in your knowledge store.
* An **index** that defines a searchable set of document records.
* An **indexer** that extracts the documents from the data source, applies the skillset, and populates the index. The process of indexing also persists the projections defined in the skillset in the knowledge store.

In this exercise, you'll use the Azure Cognitive Search REST interface to create these components by submitting JSON requests.

### Prepare JSON for REST operations

You'll use the REST interface to submit JSON definitions for your Azure Cognitive Search components.

1. In Visual Studio Code, in the **24-knowledge-store** folder, expand the **create-search** folder and select **data\_source.json**. This file contains a JSON definition for a data source named **margies-knowledge-data**.
2. Replace the **YOUR\_CONNECTION\_STRING** placeholder with the connection string for your Azure storage account, which should resemble the following:

DefaultEndpointsProtocol=https;AccountName=ai102str123;AccountKey=12345abcdefg...==;EndpointSuffix=core.windows.net

You can find the connection string on the ***Access keys*** page for your storage account in the Azure portal.

1. Save and close the updated JSON file.
2. In the **create-search** folder, open **skillset.json**. This file contains a JSON definition for a skillset named **margies-knowledge-skillset**.
3. At the top of the skillset definition, in the **cognitiveServices** element, replace the **YOUR\_COGNITIVE\_SERVICES\_KEY** placeholder with either of the keys for your cognitive services resources.

You can find the keys on the ***Keys and Endpoint*** page for your cognitive services resource in the Azure portal.

1. At the end of the collection of skills in your skillset, find the **Microsoft.Skills.Util.ShaperSkill** skill named **define-projection**. This skill defines a JSON structure for the enriched data that will be used for the projections that the pipeline will persist on the knowledge store for each document processed by the indexer.
2. At the bottom of the skillset file, observe that the skillset also includes a **knowledgeStore** definition, which includes a connection string for the Azure Storage account where the knowledge store is to be created, and a collection of **projections**. This skillset includes three projection groups:
   * A group containing an object projection based on the **knowledge\_projection** output of the shaper skill in the skillset.
   * A group containing a file projection based on the **normalized\_images** collection of image data extracted from the documents.
   * A group containing the following table projections:
     + **KeyPhrases**: Contains an automatically generated key column and a **keyPhrase** column mapped to the **knowledge\_projection/key\_phrases/** collection output of the shaper skill.
     + **Locations**: Contains an automatically generated key column and a **location** column mapped to the **knowledge\_projection/key\_phrases/** collection output of the shaper skill.
     + **ImageTags**: Contains an automatically generated key column and a **tag** column mapped to the **knowledge\_projection/image\_tags/** collection output of the shaper skill.
     + **Docs**: Contains an automatically generated key column and all of the **knowledge\_projection** output values from the shaper skill that are not already assigned to a table.
3. Replace the **YOUR\_CONNECTION\_STRING** placeholder for the **storageConnectionString** value with the connection string for your storage account.
4. Save and close the updated JSON file.
5. In the **create-search** folder, open **index.json**. This file contains a JSON definition for an index named **margies-knowledge-index**.
6. Review the JSON for the index, then close the file without making any changes.
7. In the **create-search** folder, open **indexer.json**. This file contains a JSON definition for an indexer named **margies-knowledge-indexer**.
8. Review the JSON for the indexer, then close the file without making any changes.

### Submit REST requests

Now that you've prepared the JSON objects that define your search solution components, you can submit the JSON documents to the REST interface to create them.

1. In the **create-search** folder, open **create-search.cmd**. This batch script uses the cURL utility to submit the JSON definitions to the REST interface for your Azure Cognitive Search resource.
2. Replace the **YOUR\_SEARCH\_URL** and **YOUR\_ADMIN\_KEY** variable placeholders with the **Url** and one of the **admin keys** for your Azure Cognitive Search resource.

You can find these values on the ***Overview*** and ***Keys*** pages for your Azure Cognitive Search resource in the Azure portal.

1. Save the updated batch file.
2. Right-click the the **create-search** folder and select **Open in Integrated Terminal**.
3. In the terminal pane for the **create-search** folder, enter the following command run the batch script.

create-search

1. When the script completes, in the Azure portal, on the page for your Azure Cognitive Search resource, select the **Indexers** page and wait for the indexing process to complete.

You can select ***Refresh*** to track the progress of the indexing operation. It may take a minute or so to complete.

**Tip**: If the script fails, check the placeholders you added in the **data\_source.json** and **skillset.json** files as well as the **create-search.cmd** file. After correcting any mistakes, you may need to use the Azure portal user interface to delete any components that were created in your search resource before re-running the script.

## View the knowledge store

After you have run an indexer that uses a skillset to create a knowledge store, the enriched data extracted by the indexing process is persisted in the knowledge store projections.

### View object projections

The object projections defined in the Margie's Travel skillset consist of a JSON file for each indexed document. These files are stored in a blob container in the Azure Storage account specified in the skillset definition.

1. In the Azure portal, view the Azure Storage account you created previously.
2. Select the **Storage explorer** tab (in the pane on the left) to view the storage account in the storage explorer interface in the Azure portal.
3. Expand **BLOB CONTAINERS** to view the containers in the storage account. In addition to the **margies** container where the source data is stored, there should be two new containers: **margies-images** and **margies-knowledge**. These were created by the indexing process.
4. Select the **margies-knowledge** container. It should contain a folder for each indexed document.
5. Open any of the folders, and then download and open the **knowledge-projection.json** file it contains. Each JSON file contains a representation of an indexed document, including the enriched data extracted by the skillset as shown here.

{

"file\_id":"abcd1234....",

"file\_name":"Margies Travel Company Info.pdf",

"url":"https://store....blob.core.windows.net/margies/...pdf",

"language":"en",

"sentiment":0.83164644241333008,

"key\_phrases":[

"Margie’s Travel",

"Margie's Travel",

"best travel experts",

"world-leading travel agency",

"international reach"

],

"locations":[

"Dubai",

"Las Vegas",

"London",

"New York",

"San Francisco"

],

"image\_tags":[

"outdoor",

"tree",

"plant",

"palm"

]

}

The ability to create object projections like this enables you to generate enriched data objects that can be incorporated into an enterprise data analysis solution - for example by ingesting the JSON files into an Azure Data Factory pipeline for further processing or loading into a data warehouse.

### View file projections

The file projections defined in the skillset create JPEG files for each image that was extracted from the documents during the indexing process.

1. In the storage explorer interface in the Azure portal, select the **margies-images** blob container. This container contains a folder for each document that contained images.
2. Open any of the folders and view its contents - each folder contains at least one \*.jpg file.
3. Open any of the image files to verify that they contain images extracted from the documents.

The ability to generate file projections like this makes indexing an efficient way to extract embedded images from a large volume of documents.

### View table projections

The table projections defined in the skillset form a relational schema of enriched data.

1. In the storage explorer interface in the Azure portal, expand **TABLES**.
2. Select the **Docs** table to view its columns. The columns include some standard Azure Storage table columns - to hide these, modify the **Column Options** to select only the following columns:
   * **document\_id** (the key column automatically generated by the indexing process)
   * **file\_id** (the encoded file URL)
   * **file\_name** (the file name extracted from the document metadata)
   * **language** (the language in which the document is written)
   * **sentiment** the sentiment score calculated for the document.
   * **url** the URL for the document blob in Azure storage.
3. View the other tables that were created by the indexing process:
   * **ImageTags** (contains a row for each individual image tag with the **document\_id** for the document in which the tag appears).
   * **KeyPhrases** (contains a row for each individual key phrase with the **document\_id** for the document in which the phrase appears).
   * **Locations** (contains a row for each individual location with the **document\_id** for the document in which the location appears).

The ability to create table projections enables you to build analytical and reporting solutions that query the relational schema; for example, using Microsoft Power BI. The automatically generated key columns can be used to join the tables in queries - for example to return all of the locations mentioned in a specific document.

## More information

To learn more about creating knowledge stores with Azure Cognitive Search, see the [Azure Cognitive Search documentation](https://docs.microsoft.com/azure/search/knowledge-store-concept-intro).

## Return to Microsoft Learn

Now that you have completed the exercise, [return to Microsoft Learn](https://docs.microsoft.com/learn/modules/create-knowledge-store-azure-cognitive-search/9-knowledge-check) to complete the knowledge check and earn points for completing this module. )