

## Contents

Azure Cognitive Search .....	2
How to use Azure search .....	2
Blob Storage .....	3
Azure Search .....	5
How it works.....	5
How to create Azure search service .....	6
Azure Maps .....	15
Web App .....	18

## Azure Cognitive Search

Azure Search is a managed service running in the public cloud. A development team can create a new instance of the service, then start using it right away. The team doesn't need to install or manage its own search technology. Azure Search is intended to provide developers with complex search capabilities for mobile and web development while hiding infrastructure requirements and search algorithm complexities. Azure Search is a recent addition to Microsoft's Infrastructure as a Service (IaaS) approach.

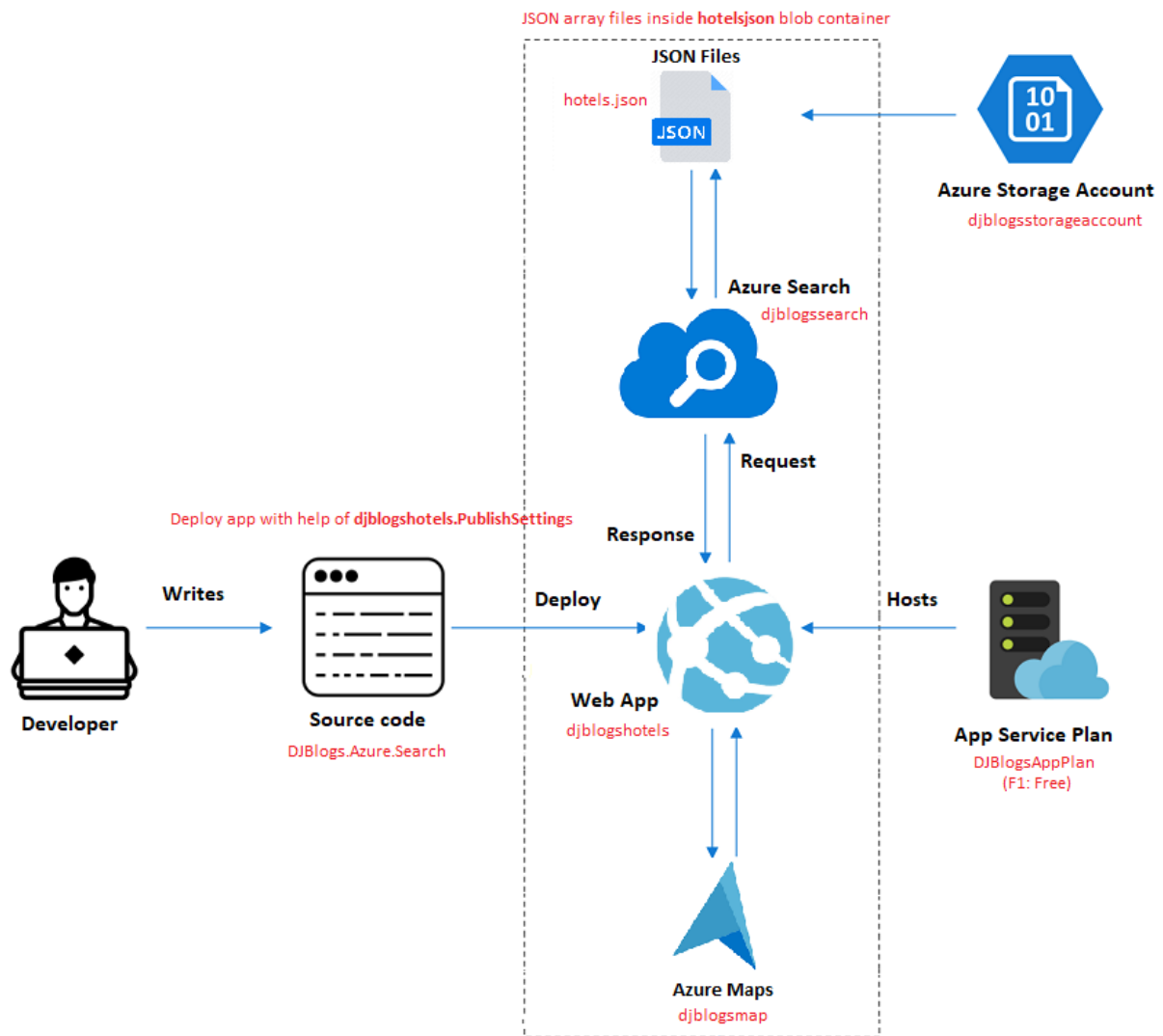
Azure Search is an API based service that provides REST APIs via protocols such as OData or integrated libraries such as the .NET SDK. Primarily the service consists of the creation of data indexes and search requests within the index.

## How to use Azure search

We will create a demo application in which will use azure search and explain how it works. In demo application we are using some of azure services

1. Blob Storage
2. Azure Search
3. Azure Maps
4. Web App

Demo application architecture will look like as below



## Blob Storage

We will use already created storage account **djblogstorageaccount** to save JSON file in blob. If you want to read more about storage account, you can read my previous blog [What is Azure Storage Account?](#) Need follow steps to create container inside already created storage account.

1. Go to azure portal <https://portal.azure.com>
2. Once we login in portal select the **djblogstorageaccount** storage account.
3. Add new **hotelsjson** container inside blob containers to save JSON files inside it.

Microsoft Azure

Search resources, services, and docs (G+/)

Home > djblogsstorageaccount 1.

djblogsstorageaccount Storage account

Containers 2.

+ Container Change access

Search (Ctrl+/)

Search containers by prefix

Name

- ☐ azure-jobs-host-archive
- ☐ azure-jobs-host-output
- ☐ azure-webjobs-dashboard
- ☐ azure-webjobs-hosts
- ☐ azure-webjobs-secrets
- ☐ images

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data transfer

Storage Explorer (preview)

Settings

- Access keys
- Geo-replication
- CORS
- Configuration
- Encryption

New container

Name \* 3. hotelsjson

Public access level ⓘ 4. Container (anonymous read access for containers and blobs)

All container and blob data can be read by anonymous request. Clients can enumerate blobs within the container by anonymous request, but cannot enumerate containers within the storage account.

Advanced

Create Discard 5.

- Once **hotelsjson** container is added. Now we will upload JSON file which have all the hotel information.

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation pane includes 'Home > djblogsstorageaccount >' and a container named 'hotelsjson'. The main area displays the 'Upload blob' page for the 'hotelsjson' container. A red box highlights the 'Upload' button (labeled '1.'). Another red box highlights the 'Files' section, showing a file named 'hotels.json' (labeled '3.'). A third red box highlights the 'Upload' button at the bottom of the file list (labeled '4.'). Below the screenshot, a red box contains a JSON file with hotel details (labeled '2.').

```
{
  "HotelId": "1",
  "HotelName": "Secret Point Motel",
  "Description": "The hotel is ideally located on the ma",
  "Description_fr": "L'hôtel est idéalement situé sur la",
  "Category": "Boutique",
  "Tags": [ "pool", "air conditioning", "concierge" ],
  "ParkingIncluded": false,
  "LastRenovationDate": "1970-01-18T00:00:00Z",
  "Rating": 3.60,
  "Address": {
    "StreetAddress": "677 5th Ave",
    "City": "New York",
    "StateProvince": "NY",
    "PostalCode": "10022",
    "Country": "USA"
  },
  "Location": {
    "type": "Point",
    "coordinates": [ -73.975403, 40.760586 ]
  }
}
```

5. Now hotels data uploaded in blob container and azure search will use it.

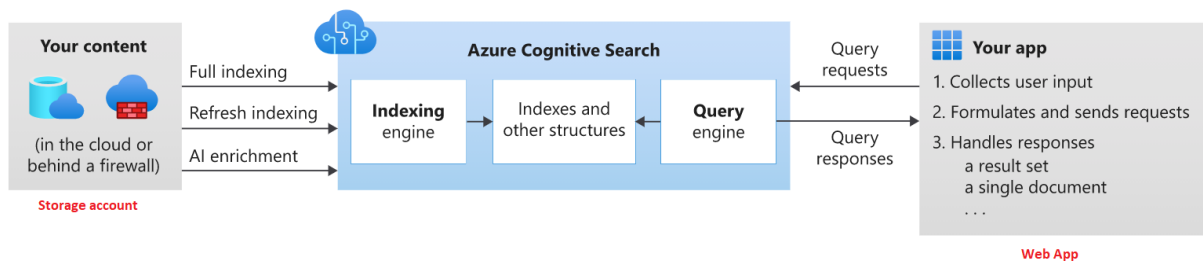
## Azure Search

If you want to use azure search service then first you need to create a new instance service, then start using it right away.

## How it works

When you create a Cognitive Search service, you get a search engine that performs indexing and query execution, persistent storage of indexes that you create and manage, and a query language for composing simple to complex queries. Optionally, a search service integrates with other Azure services in the form of indexers that automate data ingestion/retrieval from Azure data sources, and skillsets that incorporate

consumable AI from Cognitive Services, such as image and text analysis, or custom AI that you create in Azure Machine Learning or wrap inside Azure Functions.



Architecturally, a search service sits in between the external data stores that contain your un-indexed data, and a client app that sends query requests to a search index and handles the response. An index schema determines the structure of searchable content.

The two primary workloads of a search service are indexing and querying.

1. Indexing brings text into to your search service and makes it searchable. Internally, inbound text is processed into tokens and stored in inverted indexes for fast scans. During indexing, you have the option of adding cognitive skills, either predefined ones from Microsoft or custom skills that you create. The subsequent analysis and transformations can result in new information and structures that did not previously exist, providing high utility for many search and knowledge mining scenarios.
2. Once an index is populated with searchable data, your client app sends query requests to a search service and handles responses. All query execution is over a search index that you create, own, and store in your service. In your client app, the search experience is defined using APIs from Azure Cognitive Search, and can include relevance tuning, autocomplete, synonym matching, fuzzy matching, pattern matching, filter, and sort.

#### How to create Azure search service

Need follow steps to create azure search service.

1. Go to azure portal <https://portal.azure.com>
2. Once we login in portal then need to create **Azure Cognitive Search**.

[Home](#) > [New](#) >

## Azure Cognitive Search



Microsoft



### Azure Cognitive Search

Microsoft

[Save for later](#)

Create

Overview

[Plans](#)

[Usage Information + Support](#)

AI-powered cloud search service for mobile and web app development

Azure Cognitive Search (formerly Azure Search) is the only cloud search service with built-in AI capabilities to identify and explore relevant content at scale. It uses the same integrated Microsoft natural language processing (NLP) APIs across vision, language, and speech.

Azure Cognitive Search Features:

- Fully managed search as a service to reduce complexity and scale easily
- Auto-complete, geospatial search, filtering, and faceting capabilities for a rich user experience
- Built-in AI capabilities including OCR, key phrase extraction, and named entity recognition
- Flexible integration of custom models, classifiers, and rankers to fit your domain-specific needs

3. Once we click on create button it will open form like below

portal.azure.com/#create/Microsoft.Search

Microsoft Azure

Home > New > Azure Cognitive Search >

## New Search Service

Basics Scale Tags Review + create

**Project Details**

Subscription \* [Redacted]

Resource Group \* 1. DJBlogs [Create new](#)

**Instance Details**

URL \* ① 2. djbogssearch ✓

Location \* 3. Central US ✓

Pricing tier \* ① 4. **Free**  
50 MB, max 1 replicas, max 1 partitions, max 1 search units  
[Change Pricing Tier](#)

[Review + create](#) [Previous](#) [Next: Scale](#)

4. As above I am choosing free pricing tier for demo. Once we fill all the information click on Review + Create button it will create azure search for us. It will look like below

Microsoft Azure

Home > Microsoft.Search >

**djbogssearch** Search service

Search (Ctrl+/) <<

Activity log Access control (IAM) Tags Diagnose and solve problems

Settings

Quick start 2. Keys Scale Search traffic analytics Identity Properties

**Import JSON data**

+ Add index Import data Search explorer Refresh Delete Move

Create a Standard search service for scalability and greater performance. →

**Essentials**

Resource group (change)	DJBlogs	Url	1. https://djbogssearch.search.windows.net
Status	: Running	Pricing tier	Free
Location	Central US	Replicas	: 1 (No SLA)
Subscription (change)	: [Redacted]	Partitions	: 1
Subscription ID	: [Redacted]	Search units	: 1
Tags (change)	: <a href="#">Click here to add tags</a>		

[Learn more about the capabilities of Azure Cognitive Search](#)

Usage Monitoring Indexes Indexers Data sources Skillsets Debug sessions

5. Azure search service created. We can consume it with help of service URL and Keys as below.



URL: <https://djbogssearch.search.windows.net>

The screenshot shows the Microsoft Azure portal interface. At the top, the header includes the Microsoft Azure logo and a search bar. Below the header, the breadcrumb navigation shows 'Home > djbogssearch'. The main heading is 'djbogssearch | Keys', with 'Search service' underneath. A search bar with the placeholder 'Search (Ctrl+ /)' is present. On the left, a sidebar menu lists various options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Quick start, Keys (highlighted with a red box), Scale, and Search traffic analytics. On the right, the 'Keys' section displays two admin keys: 'Primary admin key' and 'Secondary admin key', each with a redacted value and a 'Regenerate' button. Below these, there is a 'Manage query keys' section with an 'Add' button and a table header 'Name'.

6. Once service instance is created then need to import data into service instance. Azure provide multiple data provider to import data into azure search as below. We will use azure storage where our hotels JSON stored.

Home > [djblogssearch](#) >

## Import data

**Connect to your data**

Add cognitive skills (Optional)

Customize target index

Create an indexer

Create and load a search index using data from an existing Azure data source in your current subscription. Azure Cognitive Search crawls the

Data Source

Type

No results

Existing data source

Existing data source

Samples

Azure SQL Database

SQL Server on Azure VMs

Cosmos DB

Azure Blob Storage

Azure Data Lake Storage Gen2

Azure Table Storage

7. Once we click on Azure Blob Storage. It will ask to select storage account as below

Microsoft Azure

Search resources

Home > djblogssearch >

Import data

\* Connect to your data

Add cognitive skills (Optional)

Customize target index

Create an indexer

Create and load a search index using data from an existing Azure data source in your current subscription. Azure Co

Data Source

1. Azure Blob Storage

Data source name \*

2. hotels

Data to extract ⓘ

3. Content and metadata

Parsing mode

4. JSON array

Connection string \*

5. DefaultEndpointsProtocol=https;AccountName=djblogsstc

Choose an existing connection

☐ Authenticate using managed identity ⓘ

Container name \* ⓘ

6. hotelsjson

Blob folder ⓘ

your/folder/here

Description

(optional)

6.

Next: Add cognitive skills (Optional)

8. To get started, an application must first create one or more indexes into data. Each index contains information that a search request can access, and it's the fundamental data store of Azure Search. Once an index exists, the application can begin issuing searches against it and displaying the results. Finally, the application must periodically update the index as the data it searches on changes.  
Getting an index ready to use requires two things:
  - a. Defining the index's schema, including specifying the fields it contains and setting various attributes for each field.

- b. Populating the index by supplying its initial set of data. Most often, this data will come from the application's operational database, but this isn't required. An index can also hold data from other sources.

Microsoft Azure

Search resources, services, and docs (G+/I)

[Home](#) > [djblogssearch](#) >

## Import data

Index name \* ⓘ

Key \* ⓘ

Suggester name

+ Add field
 + Add subfield
 Delete

Field name	Type	Retrievable	Filterable	Sortable	Facetable	Searchable	Analyzer	Suggester
HotelId	Edm.String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
HotelName	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Standard - Luce... ▾	...
Description	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
Description_fr	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
Category	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Standard - Luce... ▾	...
Tags	Collection(E...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Standard - Luce... ▾	...
ParkingIncluded	Edm.Boolean	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
LastRenovationDate	Edm.DateTi...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
Rating	Edm.Double	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
▼ Address	Edm.Comple...							...
StreetAddress	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▾	...
City	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▾	...
StateProvince	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▾	...
PostalCode	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▾	...
Country	Edm.String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	▾	...
Location	Edm.Ge... ▾	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
AzureSearch_DocumentKey	Edm.Stri... ▾	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...
metadata_storage_content_type	Edm.String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		...

Previous: Add cognitive skills (Optional)

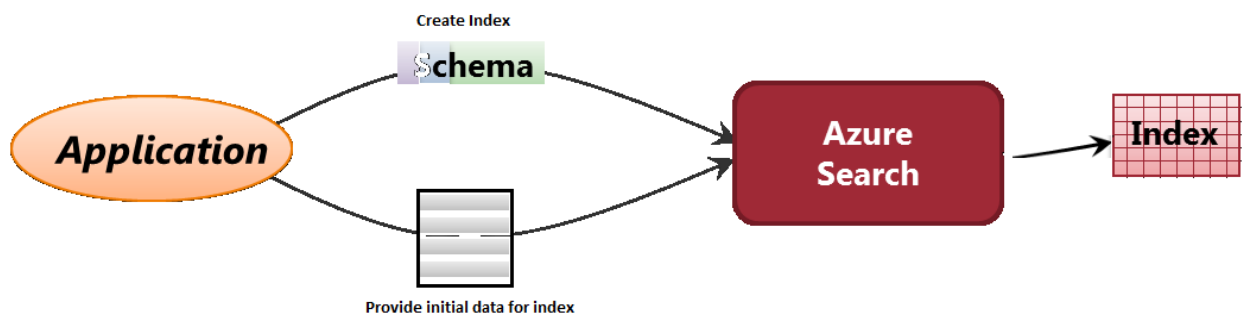
Next: Create an indexer

Field attributes determine how a field is used, such as whether it is used in full text search, faceted navigation, sort operations, and so forth. String fields are often marked as "Searchable" and "Retrievable". Fields used to narrow search results include "Sortable", "Filterable", and "Facetable".

Attribute	Description
Searchable	Full-text searchable, subject to lexical analysis such as word-breaking during indexing. If you set a searchable field to a value like "sunny day", internally it will be split into the individual tokens "sunny" and "day". For details, see <a href="#">How full text search works</a> .

Filterable	Referenced in \$filter queries. Filterable fields of type Edm.String or Collection(Edm.String) do not undergo word-breaking, so comparisons are for exact matches only. For example, if you set such a field f to "sunny day", \$filter=f eq 'sunny' will find no matches, but \$filter=f eq 'sunny day' will.
Sortable	By default the system sorts results by score, but you can configure sort based on fields in the documents. Fields of type Collection(Edm.String) cannot be "sortable".
Facetable	Typically used in a presentation of search results that includes a hit count by category (for example, hotels in a specific city). This option cannot be used with fields of type Edm.GeographyPoint. Fields of type Edm.String that are filterable, "sortable", or "facetable" can be at most 32 kilobytes in length. For details, see Create Index (REST API).
Key	Unique identifier for documents within the index. Exactly one field must be chosen as the key field and it must be of type Edm.String.
Retrievable	Determines whether the field can be returned in a search result. This is useful when you want to use a field (such as profit margin) as a filter, sorting, or scoring mechanism, but do not want the field to be visible to the end user. This attribute must be true for key fields.

It will look like as below



- Now indexes created in data. We can test within azure portal with help of Search explorer. It looks like as below.

Microsoft Azure Search resources, services, and docs (G+/)

Home > djblogssearch >

### Search explorer

djblogssearch

Index **1.** hotels-index API version 2020-06-30

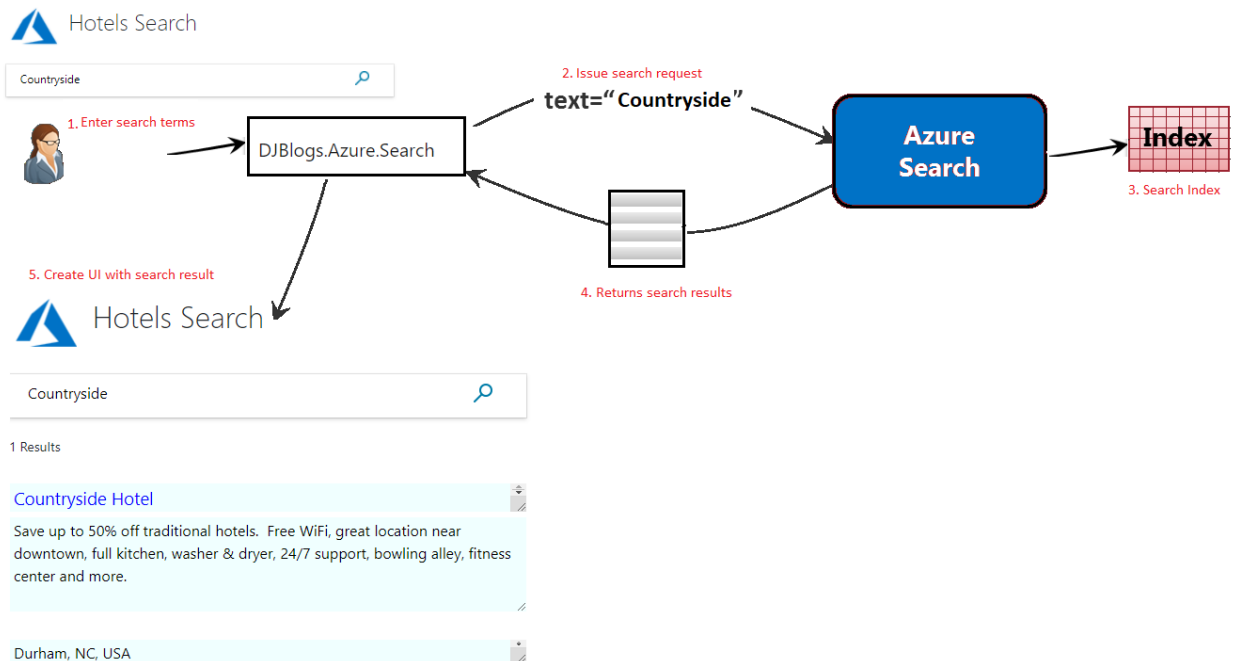
Query string **2.** Countryside **3.** Search

Request URL  
https://djblogssearch.search.windows.net/indexes/hotels-index/docs?api-version=2020-06-30&search=Countryside

Results

```
1 {
2   "@odata.context": "https://djblogssearch.search.windows.net/indexes('hotels-index')/$metadata#docs(*)",
3   "value": [
4     {
5       "@search.score": 1.8114035,
6       "HotelName": "Countryside Hotel",
7       "Description": "Save up to 50% off traditional hotels. Free WiFi, great location near downtown, full kitchen,
8       washer & dryer, 24/7 support, bowling alley, fitness center and more.",
9       "Description_fr": "Économisez jusqu'à 50% sur les hôtels traditionnels. WiFi gratuit, très bien situé près
10      du centre-ville, cuisine complète, laveuse & sécheuse, support 24/7, bowling, centre de fitness et plus encore.",
11      "Category": "Budget",
12      "Tags": [
13        "24-hour front desk service",
14        "coffee in lobby",
15        "restaurant"
16      ],
17      "ParkingIncluded": false,
18      "LastRenovationDate": "1999-09-06T00:00:00Z",
19      "Rating": 2.7,
20      "Location": {
21        "type": "Point",
22        "coordinates": [
23          -78.940483,
24          35.90416
25        ]
26      }
27    }
28  ]
29 }
```

10. Once an index has been created and populated with data, users can begin issuing searches. Below figure shows how the search process looks in our own web app.



## Azure Maps

Azure Maps is a collection of geospatial services and SDKs that use fresh mapping data to provide geographic context to web and mobile applications.

Azure Maps provides

1. REST APIs to render vector and raster maps in multiple styles and satellite imagery.
2. Creator services to create and render maps based on private indoor map data.
3. Search services to locate addresses, places, and points of interest around the world.
4. Various routing options; such as point-to-point, multipoint, multipoint optimization, isochrone, electric vehicle, commercial vehicle, traffic influenced, and matrix routing.
5. Traffic flow view and incidents view, for applications that require real-time traffic information.

In this demo we will use Azure map to show the location of hotel into map. To use maps than need to create Azure Map service with help of azure portal.

Need follow steps to create Azure Maps.


1. Go to azure portal <https://portal.azure.com>
2. Create Azure Maps

← → ↻ 🏠 portal.azure.com/#create/hub

☰ **Microsoft Azure**

[Home](#) > [New](#) >

**Azure Maps** 📌  
Microsoft

**Azure Maps** ❤️ [Save for later](#)  
Microsoft

Create

Overview   Plans   Usage Information + Support

Azure Maps provides developers from all industries, powerful geospatial and mobile applications. Azure Maps is an Azure One API compliant service. Data, and Spatial Operations accompanied by both Web and Android

3. Click on create button it will open form that need to be fill



← → ↻ 🏠 portal.azure.com/#create/Microsoft.Maps


☰ Microsoft Azure 🔍 Search resources, services, and do


[Home](#) > [New](#) > [Azure Maps](#) >

## Create Azure Maps Account


**PROJECT DETAILS**


Select the subscription to manage deployed resources and costs. Use Resource groups like folders to organize and manage all your resources.

Subscription \* 

Resource group \* 1.    
[Create new](#)

**ACCOUNT DETAILS**


Name \* 2.  

Pricing tier \* 3.    
[See pricing details and pricing tier guide](#)

☒ I confirm that I have read and agree to the [License](#) and [Privacy Statement](#). \*

Note - Azure Maps shares customer-provided address/location queries ("Queries") with third party TomTom for mapping functionality purposes. Queries are not linked to any customer or end-user when shared with TomTom and cannot be used to identify individuals. Microsoft is currently in the process of adding TomTom to the Online Services Subcontractor List. Note that the Mobility and Weather Services which include integration with Moovit and AccuWeather are currently in [PREVIEW](#).

---


4. 

4. Once we click on create button. it creates azure map service for us. We consume this service with help of client key and other keys. To show hotel in map, I have used primary key as below screen.

Microsoft Azure

Search res

Home > djblogsmap

 **djblogsmap**  
Azure Maps Account

Authentication

Search (Ctrl+ /)

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Events

Creator Preview

Creator overview

Settings

Authentication

Pricing Tier


Properties

Locks

Help

Getting Started

About

 **Alert** - To improve secure communication with Azure Maps, TLS 1.x, evaluate your TLS 1.2 readiness and develop a migrat

Azure Maps supports two ways to authenticate:


1. Azure Active Directory (Azure AD) – Azure AD is Microsoft's grained access to Azure Maps resources. To learn more abc


2. Shared Key Authentication – Shared Key authentication, oft maintain connections using one key while regenerating the

Azure Active Directory Authentication

Client ID **1.**

Shared Key Authentication

Primary Key  **2.**

Secondary Key  **3.**

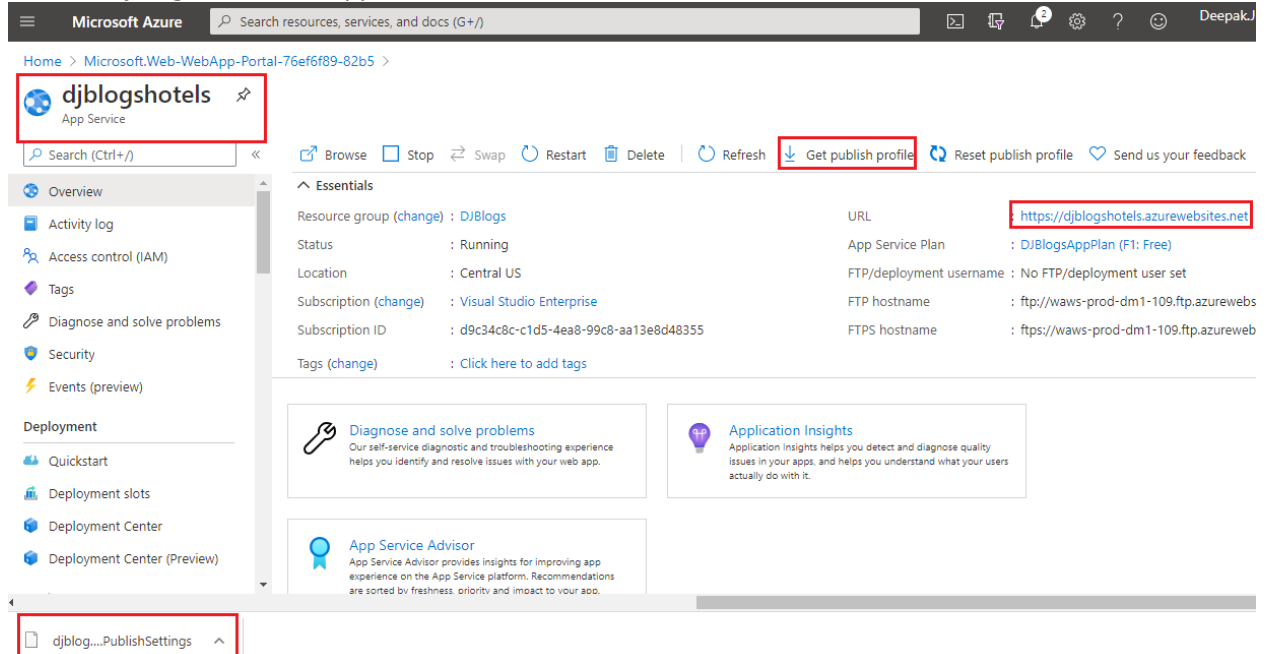
## Web App

Now our azure search service and azure map ready to consume. We need to create WebApp to consume azure search service. If you want to learn more about WebApp. Please read my previous post [Create Web App](#)

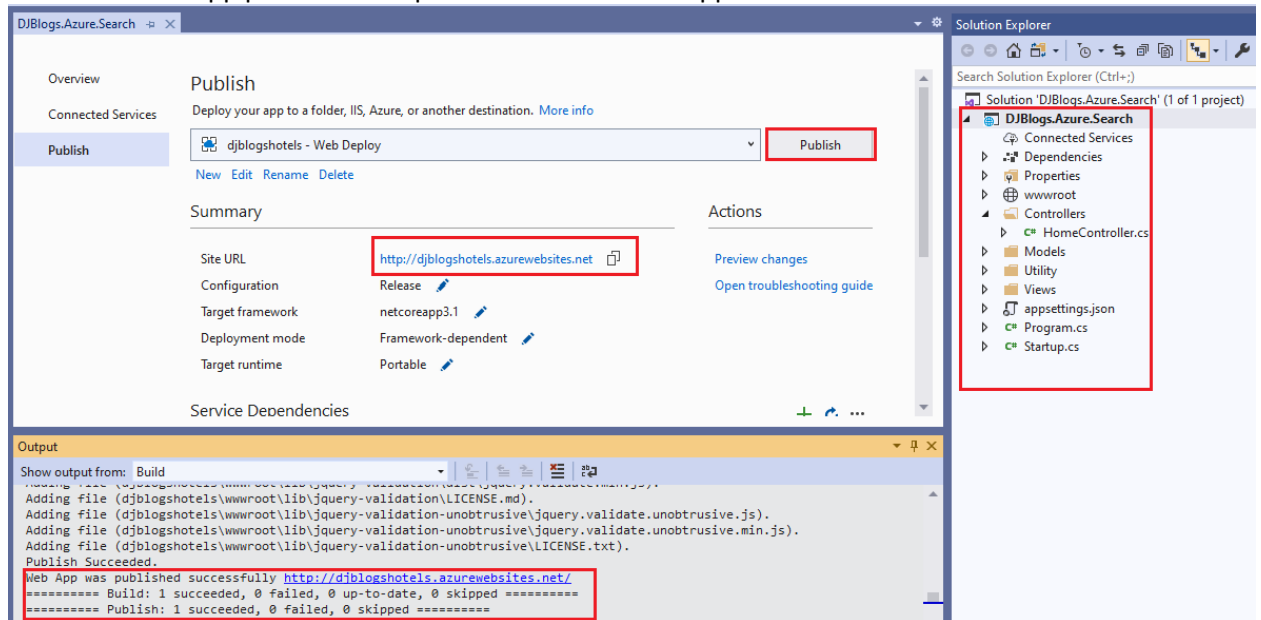
Need follow steps to create Web App.

5. Go to azure portal <https://portal.azure.com>

6. Created **djblogshotels** webapp



7. Download web app publisher and publish .net core MVC application from visual studio



8. You can download **DJBlogs.Azure.Search** from my GitHub repository.

URL: <https://github.com/deepakjoshiinfo/DJBlogs.Azure.SearchMap>

9. Once code is published it will look like as below

URL: <https://djblogshotels.azurewebsites.net/>

## Hotels Search

Secret

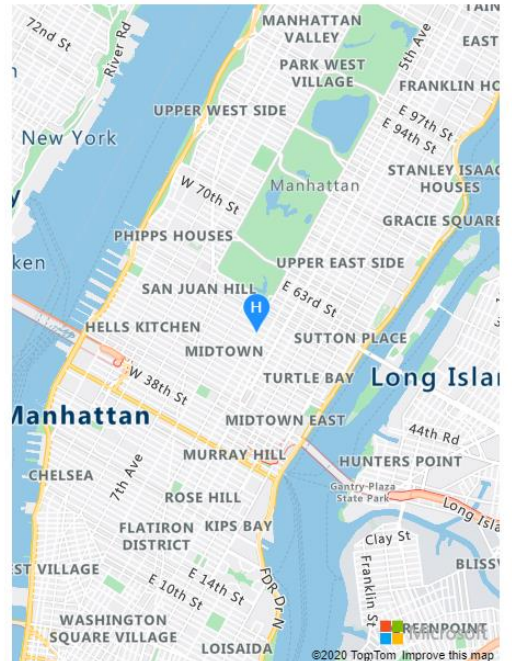


1 Results

### Secret Point Motel

The hotel is ideally located on the main commercial artery of the city in the heart of New York. A few minutes away is Time's Square and the historic centre of the city, as well as other places of interest that make New York one of America's most attractive and cosmopolitan cities.

New York, NY, USA



Hope it will help you to understand azure search and map. How we can use them in our project.  
Keep sharing keep learning