

What is Azure Search?

05/01/2019 • 11 minutes to read • Contributors      all

In this article

[Feature descriptions](#)

[How to use Azure Search](#)

[How it compares](#)

[REST API | .NET SDK](#)

[Free trial](#)

[How to get started](#)

[Watch this video](#)

Azure Search is a search-as-a-service cloud solution that gives developers APIs and tools for adding a rich search experience over private, heterogenous content in web, mobile, and enterprise applications. Query execution is over a user-defined index.

- Build a search index containing only your data, sourced from multiple content types and platforms.
- Leverage AI enrichments to extract text and features from image files, or entities and key phrases from raw text.
- Create intuitive search experiences with facet navigation and filters, synonyms, autocomplete, and text analysis for "did you mean" autocorrected search terms. Get relevance tuning through functions and boosting logic.
- Create search apps for specific use-cases. Geo-search supports a "find near me" experience. Multi-lingual search is supported through language analyzers for non-English full text search.

Functionality is exposed through a simple [REST API](#) or [.NET SDK](#) that masks the inherent complexity of information retrieval. In addition to APIs, the Azure portal provides administration and content management support, with tools for prototyping and querying your indexes. Because the service runs in the cloud, infrastructure and availability are managed by Microsoft.

Feature descriptions

Core search	Features
Free-form text search	<p>Full-text search is a primary use case for most search-based apps. Queries can be formulated using a supported syntax.</p> <p>Simple query syntax provides logical operators, phrase search operators, suffix operators, precedence operators.</p> <p>Lucene query syntax includes all operations in simple syntax, with extensions for fuzzy search, proximity search, term boosting, and regular expressions.</p>
Relevance	<p>Simple scoring is a key benefit of Azure Search. Scoring profiles are used to model relevance as a function of values in the documents themselves. For example, you might want newer products or discounted products to appear higher in the search results. You can also build scoring profiles using tags for personalized scoring based on customer search preferences you've tracked and stored separately.</p>
Geo-search	<p>Azure Search processes, filters, and displays geographic locations. It enables users to explore data based on the proximity of a search result to a physical location. Watch this video or review this sample to learn more.</p>
Filters and facets	<p>Faceted navigation is enabled through a single query parameter. Azure Search returns a faceted navigation structure you can use as the code behind a categories list, for self-directed filtering (for example, to filter catalog items by price-range or brand).</p> <p>Filters can be used to incorporate faceted navigation into your application's UI, enhance query formulation, and filter based on user- or developer-specified criteria. Create filters using the OData syntax.</p>

Core search	Features
User experience features	<p>Autocomplete can be enabled for type-ahead queries in a search bar.</p> <p>Search suggestions also works off of partial text inputs in a search bar, but the results are actual documents in your index rather than query terms.</p> <p>Synonyms associates equivalent terms that implicitly expand the scope of a query, without the user having to provide the alternate terms.</p> <p>Hit highlighting applies text formatting to a matching keyword in search results. You can choose which fields return highlighted snippets.</p> <p>Sorting is offered for multiple fields via the index schema and then toggled at query-time with a single search parameter.</p> <p>Paging and throttling your search results is straightforward with the finely tuned control that Azure Search offers over your search results.</p>
AI enrichment	Features
AI enriched documents	<p>Cognitive search for image and text analysis can be applied to an indexing pipeline to extract text information from raw content. A few examples of built-in skills include optical character recognition (making scanned JPEGs searchable), entity recognition (identifying an organization, name, or location), and key phrase recognition. You can also code custom skills to attach to the pipeline.</p>
Stored enrichments for analysis and consumption	<p>Knowledge store (preview) is an extension of AI-based indexing. With Azure storage as a backend, you can save enrichments created during indexing. These artifacts can be used to help you design better skillsets, or create shape and structure out of amorphous or ambiguous data. You can create projections of these structures that target specific workloads or users. You can also directly analyze the extracted data, or load it into other apps.</p>
Data import/indexing	Features

Data import/indexing	Features
Data sources	<p>Azure Search indexes accept data from any source, provided it is submitted as a JSON data structure.</p> <p>Indexers automate data ingestion for supported Azure data sources and handle JSON serialization. Connect to Azure SQL Database, Azure Cosmos DB, or Azure Blob storage to extract searchable content in primary data stores. Azure Blob indexers can perform <i>document cracking</i> to extract text from major file formats, including Microsoft Office, PDF, and HTML documents.</p>
Hierarchical and nested data structures	<p>Complex types and collections allow you to model virtually any type of JSON structure as an Azure Search index. One-to-many and many-to-many cardinality can be expressed natively through collections, complex types, and collections of complex types.</p>
Linguistic analysis	<p>Analyzers are components used for text processing during indexing and search operations. There are two types.</p> <p>Custom lexical analyzers are used for complex search queries using phonetic matching and regular expressions.</p> <p>Language analyzers from Lucene or Microsoft are used to intelligently handle language-specific linguistics including verb tenses, gender, irregular plural nouns (for example, 'mouse' vs. 'mice'), word de-compounding, word-breaking (for languages with no spaces), and more.</p>
Platform level	Features
Tools for prototyping and inspection	<p>In the portal, you can use the Import data wizard to configure indexers, index designer to stand up an index, and Search explorer to test queries and refine scoring profiles. You can also open any index to view its schema.</p>

Platform level	Features
Monitoring and diagnostics	<p>Enable monitoring features to go beyond the metrics-at-a-glance that are always visible in the portal. Metrics on queries per second, latency, and throttling are captured and reported in portal pages with no additional configuration required.</p> <p>Search traffic analytics is another monitoring alternative, where server-side and client-side data is collected and analyzed to unlock insights about what users are typing into the search box.</p>
Server-side encryption	<p>Microsoft-managed encryption-at-rest is built into the internal storage layer and is irrevocable. Optionally, you can supplement the default encryption with customer-managed encryption keys (preview). Keys that you create and manage in Azure Key Vault are used to encrypt indexes and synonym maps in Azure Search.</p>
Infrastructure	<p>The highly available platform ensures an extremely reliable search service experience. When scaled properly, Azure Search offers a 99.9% SLA.</p> <p>Fully managed and scalable as an end-to-end solution, Azure Search requires absolutely no infrastructure management. Your service can be tailored to your needs by scaling in two dimensions to handle more document storage, higher query loads, or both.</p>

How to use Azure Search

Step 1: Provision service

You can provision an Azure Search service in the [Azure portal](#) or through the [Azure Resource Management API](#). You can choose either the free service shared with other subscribers, or a [paid tier](#) that dedicates resources used only by your service. For paid tiers, you can scale a service in two dimensions:

- Add Replicas to grow your capacity to handle heavy query loads.
- Add Partitions to grow storage for more documents.

By handling document storage and query throughput separately, you can calibrate resourcing based on production requirements.

Step 2: Create index

Before you can upload searchable content, you must first define an Azure Search index. An index is like a database table that holds your data and can accept search queries. You define the index schema to map to reflect the structure of the documents you wish to search, similar to fields in a database.

A schema can be created in the Azure portal, or programmatically using the [.NET SDK](#) or [REST API](#).

Step 3: Load data

After you define an index, you're ready to upload content. You can use either a push or pull model.

The pull model retrieves data from external data sources. It's supported through *indexers* that streamline and automate aspects of data ingestion, such as connecting to, reading, and serializing data. [Indexers](#) are available for Azure Cosmos DB, Azure SQL Database, Azure Blob Storage, and SQL Server hosted in an Azure VM. You can configure an indexer for on demand or scheduled data refresh.

The push model is provided through the SDK or REST APIs, used for sending updated documents to an index. You can push data from virtually any dataset using the JSON format. See [Add, update, or delete Documents](#) or [How to use the .NET SDK](#) for guidance on loading data.

Step 4: Search

After populating an index, you can [issue search queries](#) to your service endpoint using simple HTTP requests with REST API or the .NET SDK.

How it compares

Customers often ask how Azure Search compares with other search-related solutions. The following table summarizes key differences.

Compared to	Key differences
-------------	-----------------

Compared to	Key differences
Bing	<p>Bing Web Search API searches the indexes on Bing.com for matching terms you submit. Indexes are built from HTML, XML, and other web content on public sites. Built on the same foundation, Bing Custom Search offers the same crawler technology for web content types, scoped to individual web sites.</p> <p>Azure Search searches an index you define, populated with data and documents you own, often from diverse sources. Azure Search has crawler capabilities for some data sources through indexers, but you can push any JSON document that conforms to your index schema into a single, consolidated searchable resource.</p>
Database search	<p>Many database platforms include a built-in search experience. SQL Server has full text search. Cosmos DB and similar technologies have queryable indexes. When evaluating products that combine search and storage, it can be challenging to determine which way to go. Many solutions use both: DBMS for storage, and Azure Search for specialized search features.</p> <p>Compared to DBMS search, Azure Search stores content from heterogeneous sources and offers specialized text processing features such as linguistic-aware text processing (stemming, lemmatization, word forms) in 56 languages. It also supports autocorrection of misspelled words, synonyms, suggestions, scoring controls, facets, and custom tokenization. The full text search engine in Azure Search is built on Apache Lucene, an industry standard in information retrieval. While Azure Search persists data in the form of an inverted index, it is rarely a replacement for true data storage. For more information, see this forum post.</p> <p>Resource utilization is another inflection point in this category. Indexing and some query operations are often computationally intensive. Offloading search from the DBMS to a dedicated solution in the cloud preserves system resources for transaction processing. Furthermore, by externalizing search, you can easily adjust scale to match query volume.</p>

Compared to	Key differences
Dedicated search solution	<p>Assuming you have decided on dedicated search with full spectrum functionality, a final categorical comparison is between on premises solutions or a cloud service.</p> <p>Many search technologies offer controls over indexing and query pipelines, access to richer query and filtering syntax, control over rank and relevance, and features for self-directed and intelligent search.</p> <p>A cloud service is the right choice if you want a turn-key solution with minimal overhead and maintenance, and adjustable scale.</p> <p>Within the cloud paradigm, several providers offer comparable baseline features, with full-text search, geo-search, and the ability to handle a certain level of ambiguity in search inputs. Typically, it's a specialized feature, or the ease and overall simplicity of APIs, tools, and management that determines the best fit.</p>

Among cloud providers, Azure Search is strongest for full text search workloads over content stores and databases on Azure, for apps that rely primarily on search for both information retrieval and content navigation.

Key strengths include:

- Azure data integration (crawlers) at the indexing layer
- Azure portal for central management
- Azure scale, reliability, and world-class availability
- Linguistic and custom analysis, with analyzers for solid full text search in 56 languages
- [Core features common to search-centric apps](#): scoring, faceting, suggestions, synonyms, geo-search, and more.

ⓘ Note

Non-Azure data sources are fully supported, but rely on a more code-intensive push methodology rather than indexers. Using APIs, you can pipe any JSON document collection to an Azure Search index.

Among our customers, those able to leverage the widest range of features in Azure Search include online catalogs, line-of-business programs, and document discovery applications.

REST API | .NET SDK

While many tasks can be performed in the portal, Azure Search is intended for developers who want to integrate search functionality into existing applications. The following programming interfaces are available.

Platform	Description
REST	HTTP commands supported by any programming platform and language, including Xamarin, Java, and JavaScript
.NET SDK	.NET wrapper for the REST API offers efficient coding in C# and other managed-code languages targeting the .NET Framework

Free trial

Azure subscribers can [provision a service in the Free tier](#).

If you aren't a subscriber, you can [open an Azure account for free](#). You get credits for trying out paid Azure services. After they're used up, you can keep the account and use [free Azure services](#). Your credit card is never charged unless you explicitly change your settings and ask to be charged.

Alternatively, you can [activate MSDN subscriber benefits](#): Your MSDN subscription gives you credits every month that you can use for paid Azure services.

How to get started

1. Create a [free service](#). All quickstarts and tutorials can be completed on the free service.
2. Step through the [tutorial on using built-in tools for indexing and queries](#). Learn important concepts and gain familiarity with information the portal provides.
3. Move forward with code using either the .NET or REST API:
 - [How to use the .NET SDK](#) demonstrates the main workflow in managed code.
 - [Get started with the REST API](#) shows the same steps using the REST API. You can also use this quickstart to call REST APIs from Postman or Fiddler: [Explore Azure Search REST APIs](#).

Watch this video

Search engines are the common drivers of information retrieval in mobile apps, on the web, and in corporate data stores. Azure Search gives you tools for creating a search experience similar to those on large commercial web sites.

In this 9-minute video from program manager Liam Cavanagh, learn how integrating a search engine can benefit your app. Short demos cover key features in Azure Search, and what a typical workflow looks like.



09:38

- 0-3 minutes covers key features and use-cases.
- 3-4 minutes covers service provisioning.
- 4-6 minutes covers Import Data wizard used to create an index using the built-in real estate dataset.
- 6-9 minutes covers Search explorer and various queries.