Azure Search is a cloud-based search solution with a wide range of features, including advanced support for faceted navigation.

Kentico provides integration functionality that allows you to store website content and other data within Azure Search indexes. The system creates indexes under a designated Azure Search service, populates them with data, and automatically keeps them up-to-date. Developers can then use the indexes to implement search solutions for websites.

To use Azure Search on your website, perform the following steps:

1. Set up an Azure Search service (see the Create an Azure Search service in the portal article).

   > **Azure Search pricing tier**
   >
   > Select the Pricing tier of your Azure Search service according to the volume of data that you plan to index and your performance requirements. See the Service limits in Azure Search article for details.
   >
   > Note: The **Basic** pricing tier has a limit of 100 fields per index, which is not sufficient for Kentico page indexes in many cases. Page indexes contain general page and product fields, combined with the fields of all included page types. If using the Basic pricing tier, you need to strictly limit the number of page types covered by your indexes, as well as the set of included fields.

2. Enable search indexing in Kentico.
3. Create Azure Search indexes. Assign the indexes to your site and define their exact content.
4. Build custom components to provide a search interface according to your website's requirements. See Integrating Azure Search into pages for more information and an example.
5. (Optional) Customize the Azure Search according to the requirements of your search scenario.

How Azure Search works in Kentico

The system stores searchable content within persistent **cloud-based indexes**. To learn about Azure Search indexes and their structure, refer to the Create an Azure Search index article.

Pages and other Kentico objects are represented as **documents** within the index. The documents contain the same fields as the corresponding Kentico objects, based on the search settings of individual page and object types. When Kentico objects covered by an index are created, updated or removed, the system automatically logs **indexing tasks**.

> **Tip**
>
> You can find a list of all Azure Search indexing tasks that are waiting to be processed in the **Smart search** application on the **Azure tasks** tab. Here you can:
>
> - Look for information if you encounter problems with new content not being indexed correctly
> - **Delete** (🗑) tasks from the indexing queue to stop unnecessary or problematic indexing
>
> See also: Monitoring search indexing tasks

The search functionality automatically processes the indexing tasks, which updates the content of the related indexes by sending requests to the appropriate Azure Search service.

Because Azure Search indexes are cloud-based and hosted outside of the Kentico application, requests that interact with indexes may fail due to network errors, service unavailability, etc. To deal with such issues, the system uses a **retry policy** for all indexing requests. Requests that fail are automatically repeated several times.

Searching is done by sending query requests to indexes on the appropriate Azure Search service. The service returns a response containing the corresponding search results. Typically, the query requests are sent by custom components that provide a search interface for the website's users and also display the returned results.

Notes

Eventual consistency

When you perform content updates or index rebuilds in Kentico, the changes may not be reflected immediately by the Azure Search. You may experience delays and/or receive outdated results due to the required network communication and the distributed nature of cloud-based services. The service only guarantees **eventual consistency**, which means that it may take a certain amount of time before all changes are reconciled.

Sharing an index across environments

Sharing a single index across multiple environments, all changes to content are automatically reflected in the search functionality of the other environments. For example, if you modify an indexed page in a testing environment sharing an index with a production environment the changes will also appear in search results on the production environment.

Indexing pages with set publication intervals

Indexed pages with a limited publication window stay indexed even after being unpublished. The system does **NOT** remove them from the corresponding Azure index – the pages will keep appearing in search results. If you wish to remove unpublished documents from the search results, filter the retrieved result set using the **DocumentPublishFrom** and **DocumentPublishTo** columns.

Example

The following model scenario explains the life cycle of a page in an Azure Search index:

1. A user (editor) creates a new page.
2. The search functionality logs a new indexing task for the page in the database.
3. The system processes the task immediately by running the **Execute Azure search tasks** scheduled task.
4. When executed, the indexing task uses the Azure Search API to add a new document (representing the given page) to the appropriate search indexes. The system indexes the page's content based on the search field settings defined for the given page type.
5. A visitor arrives on the website and sends a search request via your custom search components.
6. The components use the Azure Search API to search the specified index, and then display the returned results to the visitor.