

**Note:** The information in this section applies to search indexes stored in the application's file system. The system also supports cloud-based [Azure Search indexes](#).

Kentico provides an index-based search engine (Smart search), which allows users to search through the content of websites and various types of data within the system. The smart search is based on [Lucene.Net](#) (version 3.0.3) – a C# and .NET platform port of the Java [Lucene search engine](#).

The smart search uses **indexes** to store information about the website content. When a user sends a search request, the system searches through the appropriate indexes, which results in **significantly better performance** compared to linear SQL query search.

To set up the smart search functionality on your website, you need to perform the following steps:

1. [Enable search indexing](#) in the system.
2. [Create search indexes](#). Assign the indexes to your site and define their exact content.
3. Add smart search web parts onto the pages of your website. To learn more about the web parts, see [Building a search interface for local indexes](#) and [Adding filters for local search indexes](#).

## How the smart search works

The system stores the content of smart search indexes in physical **index files**. The index files are located in the `~/App_Data/CMSModules/SmartSearch/<Index code name>` folder within your web project directory.

Pages, forums and other Kentico objects are reflected in the index file as **Index documents**. The index documents contain the same fields as the corresponding Kentico objects, based on the search settings of individual object types. Depending on these settings, an **Index writer** creates representations of objects in the index files. When an object included in the index is created, removed or has one of its fields modified, the system automatically logs an **Indexing task**, which updates the corresponding index document. When a search occurs for selected indexes, an **Index searcher** goes through the content of the corresponding index files and returns relevant results.



### Tip

You can find a list of indexing tasks that are waiting to be processed in the **Smart search** application on the **Local 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](#)

## Example

The following model scenario explains the life cycle of a page in a search index file:

1. A user (editor) creates a new page.
2. Upon the page's creation, the system logs a new indexing task in the database.
3. The smart search either runs the indexing task immediately or processes it later using a scheduled task.
4. When executed, the indexing task adds the new 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 a smart search web part.
6. The system processes the search request, searches through the assigned indexes, and returns results.