Integrate search into your app

October 27, 2022 • Jan Cerman • 4 min read

If your web application deals with a lot of content, it might be a good idea to let your users search it.

When adding search functionality to your application, a key decision is to define your search index. In other words, the way of collecting, parsing, and storing data to enable its fast retrieval. The choice depends on the complexity and requirements of your project.

In general, there are two approaches to integrating search into your app.

- Use an external search service and continuously synchronize its search index with your Kontent.ai project.
- Filter content within your app.

Search services

The recommended approach is to use a dedicated external search service. It works for small and large projects and can scale as you go. The search service holds your search index, performs search operations over the index, and returns results.

The service can provide you with advanced features such as typo tolerance or custom ranking of the search results. Also, it often comes with SDKs and tutorials to help you integrate it within your app.

To ensure the search index stays up to date, you need to react to the content changes made in your Kontent.ai project. You can automate the process of updating your search index by either setting up webhooks and responding to notifications, or asking for latest changes at your own pace with Sync API.

- With webhooks, you provide a URL such as `https://myapp.com/update-search-index` and you get notifications to that URL right after any content changes. Based on the notifications, you decide whether to add new content or remove records from your search index.
- With Sync API, you ask the API if there were any recent changes to content you're interested in. If there are changes to process, you decide whether to add new content or remove records from your search index. And after a while, you ask again.

Worked examples

Learn how to integrate a search service into your app from worked examples in our blog posts.

- **Algolia**
  - Searching Content with Algolia – walks you through adding Algolia search service into an ASP.NET MVC application and updating search index in reaction to webhook messages.

- Algolia integration in Express.js sample app – shows you a working JavaScript sample app that uses Algolia for search.
- Implementing search with Gatsby and Algolia – guides you to bring relevance, typo-tolerance or term-boosting with Algolia to a website built on Gatsby, where the built-in filter isn't sufficient.

- **Hawksearch**
  - Hawksearch's Integration Will Refine Your Search Experience – presents how to connect Hawksearch to Kontent.ai by showing how to request and display search results.
- **Azure Search**
  - How to Integrate Azure Cognitive Search with Kontent.ai – shows how you can use Azure Cognitive Search with webhooks in a .NET web app.

In-app filtering

In-app filtering can work well for smaller web applications and single-page applications. For example, if you have a smaller project (up to 100 content items) and don't need a robust search, you can cache the content from your Kontent.ai project in your app and filter the content during the app's runtime.

With this approach, you gain control over the search index and its content. Using an existing search library in your app can help. However, it also means that to change the search behavior, you often need to adjust your code manually or replace the search library with another one.

Why not both?

With mobile apps, both approaches can be combined at the same time. If your app is online, it can fetch search results using a search service. When offline, the app can fallback to in-app filtering of the content that's available or was searched for previously.

When you need to add search to your web app, an external search service fits the bill in most cases. After you complete the initial integration with the service and add logic for reacting to content changes, you get a search with an index that's automatically updated.

What's next?

- React to content changes either immediately with webhooks or at your own pace with Sync API.
- Discover what your users are looking for by using Google Analytics in your app.
- Learn about managing navigation menus for your app in Kontent.ai.