

Use sitemaps to improve SEO and accessibility

October 21, 2022 • David Klement • 13 min read • GraphQL

There are two kinds of sitemaps. Sitemaps for people and sitemaps for search engines.

Sitemaps for people help your website visitors understand the structure of content your website has. These sitemaps improve your website's accessibility because they help your visitors find the content they're looking for more easily.

A sitemap for search engines contains an unstructured list of pages on your website. Provide such a sitemap to ensure good SEO – if a search engine finds what your potential visitors search for, they can find it using the search engine and become real visitors.

You can model a sitemap in Kontent.ai by using [linked content items](#) or by [tagging your content](#) with taxonomy.

Improve SEO with sitemaps

When coming to your website, search engines first check whether you provide a sitemap. If you do, they primarily index the pages you list in the sitemap file. Sitemaps are particularly important for large websites with hundreds of pages because search engines might not index some recently created pages.

A sitemap for search engines is an XML file that contains a list of URLs that doesn't reflect the hierarchy of the site's content.

XML

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <urlset xmlns="http://www.example.org/schemas/sitemap/0.9">
3   <url>
4     <loc>https://www.mywebsite.org/home</loc>
5     <lastmod>2020-01-08</lastmod>
6     <changefreq>monthly</changefreq>
7   </url>
8   <url>
9     <loc>https://www.mywebsite.org/shops</loc>
10    <lastmod>2020-02-14</lastmod>
11    <changefreq>weekly</changefreq>
12  </url>
13  <url>
14    <loc>https://www.mywebsite.org/shops/brisbane</loc>
15    <lastmod>2020-03-07</lastmod>
16    <changefreq>weekly</changefreq>
17  </url>
18 </urlset>
```

An optimal sitemap contains only pages with content you deem worthy of your visitors. For example, a *Recover password* page doesn't need to be in the sitemap because your users are most likely not going to search for it using a search engine.

You can also include further information about your pages – like how frequently you update them, how old is their latest version, or that your [content is localized](#).

Help people understand your website structure

A sitemap for people is a webpage that shows the structure of your website. It improves accessibility as it helps visitors better orient in the content you provide. It's especially useful in the case of complex websites with nested categories and pages. For instance, a company website with pages for shops across multiple cities, catering services, and a list of the company suppliers might have a website with the following structure:

See the diagram on [https://viewer.diagrams.net?](https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKenticoDocs%2Fkontent-docs-)

https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKenticoDocs%2Fkontent-docs-diagrams%2Fmaster%2Fcontent_modeling%2Fcoffee_example%2Fcoffee_website_structure.drawio

Just like sitemaps for search engines, sitemaps for people don't need to include pages like *Recover password*. Link such pages at an appropriate place instead so people can easily find them when they need them.

Ways to build sitemaps in Kontent.ai

You can model a sitemap for your Kontent.ai project using [linked content items](#) or [taxonomy](#). You can build both types of sitemaps with either approach but you should first decide whether you want to build a sitemap for search engines or a sitemap for your visitors (or both).

To build a sitemap for search engines, it's best to use the [taxonomy approach](#) as it's easier to manage the flat structure with it.

You can use either approach to build a sitemap for people. However, when you use the taxonomy approach, you also give your content creators an easy way to [find the tagged content](#) and see your website structure in Kontent.ai.

Taxonomy

Choose a taxonomy term

- Sitemap items
 - Shops
 - Brisbane
 - London
 - New York
 - Catering services
 - Dinners
 - Cold dishes
 - Fruit cocktails
 - Our suppliers
 - Example Ltd.

💡 Limit sitemap changes to specific roles


For both approaches, you can put the content elements for modeling the sitemap to a [separate content group](#). This allows you to limit access to the group so only selected users can edit the sitemap.

The following guides illustrate the concept of building a sitemap for a website structure similar to the [schema above](#).

Create sitemaps using linked content items

Sitemaps for visitors

When building a sitemap for your visitors, you need to link your content items in a way that reflects the hierarchy of your website. With this approach, you always go from the top of your website's hierarchy. Take the highest content item and link to it the items directly below. Then continue the same way with all the items lower in the hierarchy.


 If you use linked items [for navigation](#), you can use these linked items also [to build your sitemap](#).

Sitemaps for search engines

If you want only a sitemap for search engines, you don't need to reflect your website's structure. Take the highest content item and link to it all the lower items you want in the sitemap. However, we suggest you use the [taxonomy approach](#) in this case because it's easier to manage the flat structure with it.

1. Adjust the content model

You need to adjust the content types of items you'll link to items lower in the hierarchy.

1. In  **Content model**, open the content type you want to update.
2. Insert the **Linked items** element.
3. Name it *Sitemap items*.
4. (Optional) Restrict the element to accept only content types of items lower in your website hierarchy.
5. (Optional) Add a helpful guideline to explain the purpose of the element.
6. Click **Save changes**.

In the case of a sitemap for search engines, it's enough to add the Sitemap items element to the content type of the item highest in the hierarchy.

2. Link the content items

Now you have your content types ready and you can [link the content items](#) according to what you're building.

If you're creating a sitemap for search engines, open the highest content item with the Sitemap items element. Then link to it all the lower items you want in the sitemap.

If it's a sitemap for people, go from the top of the hierarchy and link the items that are one level lower. If you were building a sitemap for the website in [the schema above](#), you'd create a hierarchy similar to this:



3. Build the sitemap in your app

After you [publish](#) the content items, your app needs to retrieve the items using the Delivery API to build the sitemap.

With the [Delivery GraphQL API](#), get the item highest in your website hierarchy and query the items linked in its *Sitemap items* element.

GraphQL

```
1 query GetSitemapItems {
2   landingPage(codename: "landing_page") {
3     sitemapItems {
4       items {
5         title
6         url
7       }
8     }
9   }
10 }
```

Sitemaps for visitors

In the case of a sitemap for people, we suggest using the `depth=1` parameter to include linked items only one level below. This helps you to keep track of what level of the sitemap you're on when you process the responses from the API.

To get the full hierarchy, use the call above for every item on each level. Save the title, URL, and position in the hierarchy of every item you get. This way you'll progressively get to the bottom of your website's hierarchy.

After you get all the items you want in your sitemap, you can start building the sitemap in your app using the data you've retrieved.

Sitemaps for search engines

For a sitemap for search engines, use the API call above once to get the highest item with all the items you linked to it. You only need each item's URL, so save those to a list and you're ready to build the XML file.

Tag items with a taxonomy to create a sitemap

Sitemaps for visitors

When you use taxonomy to build a sitemap for people, first choose how many levels of the hierarchy you want in the sitemap. Then create a taxonomy term for each category and nest them the same way the categories are organized on your website. When you have the terms, tag each item with the term representing the appropriate category.

Sitemaps for search engines


To build a sitemap for search engines using taxonomy, you're good to go with just one term because you don't need to model your website hierarchy. Tag the items you want in the sitemap with this term and then use the Delivery API to get all the tagged items.

Dive deeper into taxonomies

If you're completely new to taxonomies, go through [our crash course on taxonomies](#) where you'll learn about what a taxonomy is, what it's useful for, and how you can start using it.

1. Create the taxonomy


First, you need to create a taxonomy group for the sitemap terms:

1. In  **Content model**, switch to the **Taxonomies** tab.
2. Click **Create new**.
3. Name the new taxonomy group *Sitemap*.

Then, create taxonomy terms you'll use to tag your items.

1. Click **Create new taxonomy term**.
2. Name the new term.
3. Click **Confirm**.

If you're building a sitemap for search engines, you're done.


If your goal is a sitemap for visitors, create a term for each category in your hierarchy and use  to reorder and nest the terms to match the website hierarchy. If you had a website like the

one in [the schema above](#), your taxonomy structure would look like this:




2. Adjust content types for tagging

Once you have the new taxonomy ready, prepare your content types for tagging items. You need to add the taxonomy element to the content types of the items you want in your sitemap.

1. In  **Content model**, open the content type you want to adjust.
2. Insert the **Taxonomy** element.
3. In the added taxonomy element, select *Sitemap*.
4. (Optional) [Limit](#) the element to one term.
5. Click **Save changes**.

3. Tag your items

When you're done, it's time to tag the items.

1. In  **Content & assets**, open the content item you want to tag.
2. In **Sitemap**, select a term for that particular item.

If you're building a sitemap for search engines, you'll use the same term for all the items you want in the sitemap. In the case of a sitemap for people, tag the items based on their position in your website hierarchy.

4. Get the tagged items to build the sitemap

After you finish tagging the items and [publish](#) them, get the data for building the sitemap using the [Delivery API](#). You need to retrieve all the [items tagged with a term](#) from the Sitemap taxonomy group.

Use the Delivery REST API to [retrieve the taxonomy](#) for the sitemap by its codename (in this tutorial, it's `sitemap`). This gives you the structure of the sitemap you're building. Then use the [\[contains\] filter](#) with the codename of each term in the taxonomy to get items you tagged.

For example, the GraphQL query to get articles tagged with the Shops term would be:

GraphQL

```
1 query GetArticlesTaggedWithATerm {
2   article_All(where: {sitemap: {containsAny: ["shops"]}}) {
3     items {
4       title
5       url
6     }
7   }
8 }
```

For an unstructured sitemap for search engines, save just URLs of the items you get to a list. If you're building a sitemap for people, save the item names and URLs into a data structure that reflects the structure of the taxonomy (for example JSON).

What's next?

As you have learned in this tutorial, creating a sitemap is not complicated. Sitemaps are important for both your visitors and for search engines. When you provide both kinds of sitemaps, you ensure not only that people are able to find your content, but also that they feel comfortable on your website.

Moreover, by modeling your website structure using taxonomy, you also help your content creators. They can use the taxonomy to effectively find the content they search for.

- SEO is not only about sitemaps but also about [optimizing your content](#) so it provides the right metadata.
- Generate URL slugs from text elements to [create SEO-friendly URLs](#).
- [Optimize images](#) for the web to achieve better loading times and faster responses.
- Create a proper [content model](#) to deliver your structured content on any platform and make it future-proof.