Establish hierarchies for navigation

June 28, 2022 • Boris Pocatko and Tomas Nosek • 3 min read

Whether your content is displayed to your visitors through a website, mobile app, or a different channel, it typically has some sort of hierarchy the purpose of which is navigation. For example, websites have page trees – there's a home page leading to an article listing, a contact page, etc. The article listing then leads to an article view.

This structure is also a relationship that you've added to your diagram in the previous chapter. To build such a structure, you need to create a relationship between each level of your hierarchy.

Separate content from navigation

One of the reasons why modular content platforms, such as Kontent.ai, become popular is the fact that the content created is reusable among different channels such as websites, mobile apps, smartwatch apps, or chatbots. Creating quality content isn't trivial so if you create it, you most likely want to present it effectively on different devices.

However, what can work for websites doesn't work for mobile apps and other devices. While websites have typically SEO parameters and URL slugs, mobile apps divide content into screens.

To get the flexibility in using the same content on different devices, create a content type for the actual content, and then create a content type for navigation in each of your channels. In Web Spotlight, there are already *Home* and *Page* content types created for your website navigation.

Link content with navigation

So that your website or mobile app knows what content to display within a specific location, connect your navigation items to your content. Create a *Linked items* element in the navigation item's content type. In its items, you'll use the element to link the actual content.

You can connect it the other way around if it's more comfortable for content creators in your case. However, the first approach proves to be better in most cases.

If you use Web Spotlight, use the *Subpages* element instead of the *Linked items* element.

Dive deeper into navigation

Check out our article focused on best practices for navigation to get more detailed instructions in this area.

Adjust the diagram for your navigation

If your project represents a website, this adjustment will most likely require more changes to the current state of your content model diagram. In the sample website, the flow of the relationship changed. Instead of the *Articles* and *Landing pages* linking to *Navigation items* (through the *SEO metadata* snippet), the *Navigation item* was renamed to *Page* and it now leads to either an *Article* or a *Landing page*.

See the diagram on https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKenticoDocs%2Fkontent-docs-diagrams%2Fmaster%2Fcontent_modeling%2Fsafelife_tutorials%2Fsafelife_navigation.drawio

It expresses the idea behind the web structure. A website contains a page tree. The page tree contains pages. Pages can be articles, landing pages, or pages with another purpose. Also, *Articles* and *Landing pages* contain only the actual content this way.

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

If your project doesn't contain a website, or you're not using Web Spotlight, use a *Linked items* element instead of the *Subpages* element. You can also keep the *Navigation item* or any name that makes sense for your use case.

What's next?

You've got the content model and navigation covered so it's time to have a look into another part of content modeling – taxonomies. The fact is that some kinds of categorization serve better when they're done by taxonomies than by content types themselves.

Explore taxonomies