

Manage navigation

February 7, 2022 • Tomas Nosek • 5 min read • JavaScript

Whether displayed as a website menu or navigation in your mobile app, navigation builds a hierarchy in your project. In Kontent, your content creators can manage navigation menus without developer assistance if you model your navigation.

Key points

- Choose the right approach. We recommend separating navigation from content in most cases.
- You can display well-modeled navigation however you prefer.
- Use the subpages element (if you use Web Spotlight) or linked items element for navigation.

Suitable navigation approach

Navigation is composed of content items based on a content type dedicated for navigation. Define the relationships in the navigation using a subpages element if you use Web Spotlight or with linked items element in other cases.

When modeling the navigation, decide whether to bind content to the navigation or keep them separate.

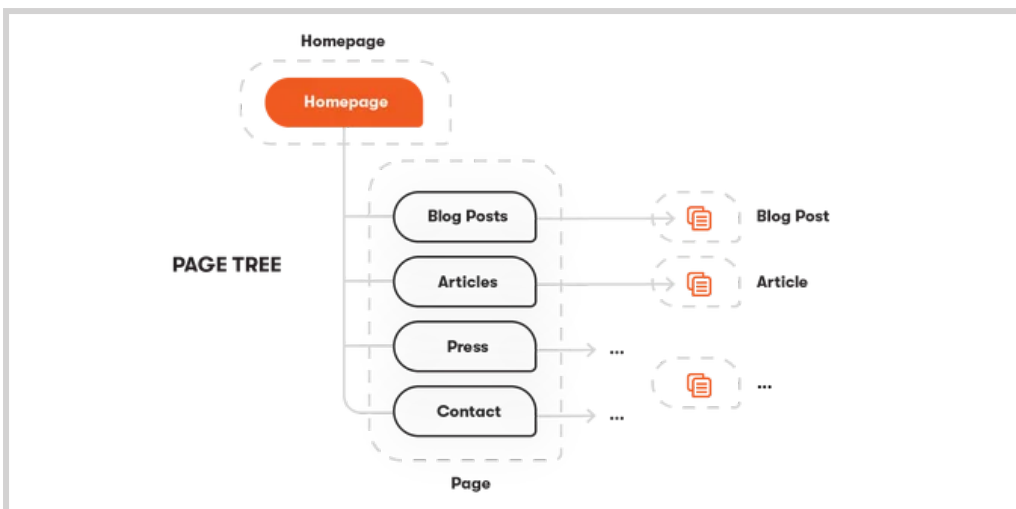
Navigation bound to content

This way, items that you link in the subpages or linked items element also hold the content. While this approach is more straightforward for understanding the concept, it isn't suitable for omnichannel delivery.

That's why it's more suitable for smaller projects or projects that use one way of navigation for all channels.

Navigation separate from content

If you separate navigation from content, you're creating two items for content linked from your navigation. The first is for the content itself; the second is for a navigation item that links the first one.



This way is better for larger projects prepared for omnichannel content.

Build your navigation

Using Web Spotlight?

If your project comes with Web Spotlight, some steps are already done for you. Check out [our article on what's created automatically](#) after Web Spotlight is activated.

1. Create a content type for navigation items

[Create a new content type](#) and add elements needed for your channel. For the code examples below, this tutorial calls this type *Navigation item*. It typically contains:

- A text element for the title of the navigation item (*Title* in the code sample)
- A subpages or linked items element for child menu items (*Subitems*)
- A linked items element for the content itself
- A URL slug element if you're building website navigation (*URL*)

2. Define your root

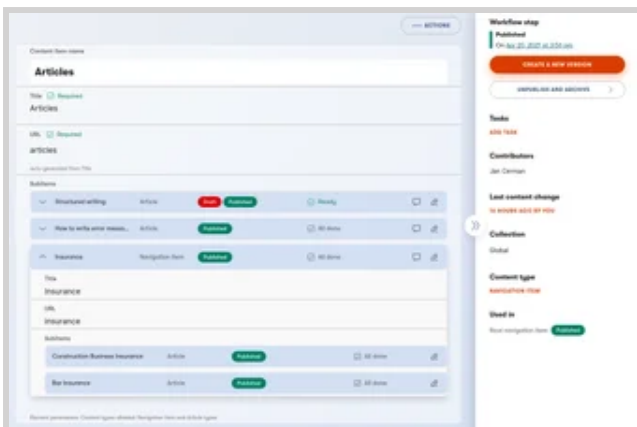
You need to define where your navigation begins. While websites typically have a homepage, other structures may have a root or main screen. You can use the content type created for navigation for the root navigation item or create a different content type for this purpose.

Websites usually benefit from having a separate content type for their homepage. That's because homepages have a unique appearance and contain a different layout and content than on other pages.

The code example below uses the navigation item for the homepage, [created](#) with the name *Root navigation item*.

3. Create your navigation items

Create content items using the created navigation content type and link them with the subpages or linked item element.



When finished, [publish](#) the root together with all its linked items.

4. Retrieve your navigation items

Developers now need to request the items via the [Delivery API](#) to generate your menu.

Specify the `depth` parameter to retrieve more than one level of subpages or linked items if you want to show all navigation with one API call.

JavaScript

```
1 // Tip: Find more about JS/TS SDKs at https://kontent.ai/learn/javascript
2 const KontentDelivery = require('@kentico/kontent-delivery');
3
4 const deliveryClient = KontentDelivery.createDeliveryClient({
5   projectId: '8d20758c-d74c-4f59-ae04-ee928c0816b7',
6 });
7
8 // Gets navigation items and their linked items
9 const rootResponse = await deliveryClient.item('root_navigation_item')
10   .depthParameter(5)
11   .toPromise();
12
13 // Gets specific elements of all articles
14 const articleResponse = await deliveryClient.items()
15   .type('article')
16   .elementsParameter(['title', 'url'])
17   .toPromise();
```

5. Display navigation

Using the retrieved data, display the navigation as you prefer. It can be your sections in a mobile app or smartwatch, or it can be a menu on a website like this:

See the code example on <https://codepen.io/KenticoKontent/pen/WNRajao>

The content of the linked items is inside the `modular_content` property of the response.

And that's it! You now have a dynamic menu that your content creators can manage entirely from Kontent.

Microsites and proofs of concept

If you're building a microsite or just a quick proof of concept and need navigation fast, you can hardcode the navigation. No part of your navigation is modeled in Kontent; it's all in the source code. See our [sample application](#) as an example. We discourage using hardcoded navigation for larger live projects.

What's next?

- Think about navigation when [creating your content model](#).
- Take a look at our [development resources](#) for your technology.
- [Create a sitemap](#) for your website to improve SEO and accessibility with the help of your navigation.
- [Decide whether Web Spotlight would be suitable](#) for your use case.