Set up routing and URLs for your app

June 28, 2022 • Jan Cerman • 7 min read

When building a web app, you need to decide what your URLs are going to look like. Do you need hierarchical URLs or short URLs? Do you need the URLs to be human-readable? What about localized content and languages? Let's look at several options you can choose from.

Key points

- Content models typically don't match URL hierarchies. The URL hierarchy is just one of the relationships in a content model.
- Using URL slugs to identify content is a good way in any approach to routing.
- For localization, use either language codename somewhere in the URL or translate the URLs completely.

Start with content model

Before you dive into routing and the URL logic of your app, we recommend you begin with setting up a content model for your project. This will help you think through your project structure and decide how to manage navigation in your project.

Keep in mind that your content model doesn't need to reflect your URL structure. In fact, it shouldn't. Kontent.ai gives you enough flexibility to create any hierarchy you want. Your web app, however, doesn't need to expose that hierarchy. For example, you can nest your content four levels deep but have nice short URLs.

Choose your URL structure

Your first choice is between the traditional hierarchical URLs like `/section/subsection/topic/page` and shorter URLs like `/article/page`. This choice depends on your preference and requirements, both approaches are well-established. It also determines how you split your URLs and how complex your routing gets.

A) Traditional hierarchical URLs

Hierarchical URLs look like `/tutorials/develop-apps/build-strong-foundation/routing-and-urls`. They show you the page location in the context of the whole website.

You might want to use URLs like these for projects with lots of content organized into categories or topics. For example, this applies to knowledge bases, documentation portals, or help centers. If you use any kind of web analytics, this approach allows you to see how the different sections (or topics) of your website perform.

How to build hierarchical URLs

Hierarchical URLs consist of several levels depending on how nested your content is. For example, this docs page is on the fourth level. Each part of the URL (separated by the slash symbol `/`) relates to a specific level in the hierarchy.

In your Kontent.ai project, you can have content items linked together in a way that will represent your nested hierarchy. To build hierarchical URLs for your pages, you need to retrieve the content items from your project and go through the hierarchy in your app from top to bottom (following the content of the linked items elements). This gives you a list of URLs that you can match with specific content items.

B) Short URLs

If you don't need hierarchical URLs, you can use shorter URLs like `/campaigns/autumn-flash-sale` or `/whats-ahead-for-us`. They give you a short human-readable text that's often fully visible in the address bar.

You might want to use short URLs for product websites, e-commerce shops, or blogs. For single-purpose websites or microsites, you can use single-level URLs like `/whats-ahead-for-us`.

How to build short URLs

Short URLs are one or two levels deep. For example, take the URL `/campaigns/autumn-flash-sale` for a marketing campaign page. The URL is two levels deep – one level is the section identifier `/campaigns` and the second level is the URL slug of the specific page. You can have more sections for different types of content like articles or social events. For each type, you create its own section (like `/articles` or `/events`), its own route within your app.

To build URLs like these, retrieve the content items of a specific type first and then use their URL slug value for the URL. If you have lots of content with possibly duplicate titles, consider adding another identifier in the URL next to the URL slug to make sure your URLs stay unique.

Choose your page identifiers

When you start creating the routing logic in your app, you need to decide how to identify your content in the URLs. Using URL slugs as identifiers is a good starting point. While the URL slugs of your items aren't guaranteed to be unique across your project, they are likely to be unique for specific types of items. The chance of running into naming conflicts is also lower with hierarchical URLs than with shorter URLs.

Prevent duplicate page identifiers

If you have an e-commerce business, you might have several products with the same name but with different properties. In cases like these, consider using multiple identifiers in the URL to differentiate between such content.

You can choose from the content item's ID, codename, or external ID. All guaranteed to be unique. For example, you can use a segment of the content item's ID and its URL

slug to create an identifier such as `de0b65a2-autumn-flash-sale`.

Considerations for multilingual URLs

If you're building a multilingual website, consider how to differentiate the languages, locales, or regional variants of your content in the URL. You can have several regional variants of a single language (British English, American English, Canadian English), multiple different languages (English, Spanish, French), or a combination of both.

To differentiate your content variants, you can choose from three common approaches:

1. Top-level domain – `example.com`, `example.de`, `example.cz`
2. Subdomain – `en.example.com`, `de.example.com`, `cz.example.com`
3. Subdirectory – `example.com/en-us`, `example.com/de-de`, `example.com/cs-cz`

Each comes with its own set of pros and cons. The choice depends on your requirements, preferences, and budget.

The common practice is to take the subdirectory approach. The main benefit is that you don't need to worry about domains and their renewal. With this approach, you can use language codenames in the URL to identify the language and its locale. Language codenames can be used with any of the approaches to identify your content variants.

You also need to decide whether the URLs should be localized or universal. Here are two approaches you can take:

1. Language prefixes – `example.com/en-us/about-us`, `example.com/es-es/about-us`
2. Localized URLs – `example.com/about-us`, `example.com/acerca-de`

What's next?

Now that you know the different approaches to creating unique URLs for your content items, choose the ones that fit your requirements. Well-implemented routing will let you scale in the future when your project gets bigger.

- Create URL slug elements in your content model so you can create SEO-friendly URLs.
- If you deal with multiple languages, see how to build multilingual site with Gatsby for inspiration.
- Set up a custom asset domain to improve the SEO of your assets.
- Set up preview URLs so your editors can write with confidence.