

Create SEO-friendly URLs



October 21, 2022 • Martina Farkasova • 6 min read

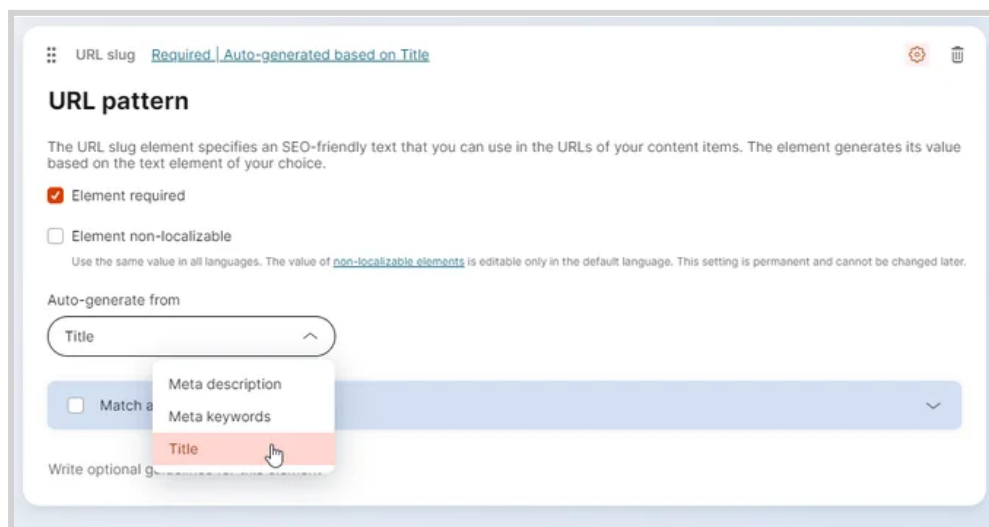
Learn how to generate URL slugs from text elements to customize your URLs. The URL slug element specifies an SEO-friendly text that you can use in the URLs of your content items. The element automatically generates its value based on the text content element of your choice, unless you change the element's value manually. You can use the element only once in a content type.

Because the URL slug element can generate its value from any text element, you can use a different URL slug for each [language variant](#) of your content items.

Let's look at an example of using a URL slug in a content type.

Add a URL slug to content types

1. In  **Content model**, open a content type for editing.
2. Add the *URL slug* element to the content type.
3. Type the name of the *URL slug* element. For example, you can label it *Page URL*.
4. Click  in the element.
5. Select a text element from the drop-down list to generate a URL slug.
 - The list shows text elements from the opened content type and also from any [content type snippets](#) already used in the type.
 - If there isn't any text element to choose from, you need to add one first.
6. Click **Save changes**.



The URL slug's length is limited only by the maximum length of the dependent text element. Keep in mind that the limitation for the URL in web browsers is usually around 2,000 characters.

Reuse URL slug across languages

To keep items' URL slugs the same across your project languages, you have multiple options:

1. Set as [non-localizable](#) both the URL slug and the source element it's autogenerated from. All languages will use the default language value.
2. Set only the URL slug as non-localizable. The source element will be editable in non-default languages, but the URL slug will retain the value from the default language and be read-only in non-default languages.
3. Set only the source element as non-localizable. Initially, the URL slug will have the default language value across all languages, but it will be editable. That means you can localize it in non-default languages.

URL slug in content items

In content items, your content contributors will see the added URL slug element like this:



URL pattern Required

If you need a custom value, provide a SEO-friendly URL. Otherwise, the value is generated from the article's title.

great-news-about-us

auto-generated from Title

In the bottom right corner of the URL slug (*URL pattern* in the image), you can see which text element generates the URL slug and how many characters it has.

Autogenerated vs. manual values

By default, the URL slug value is autogenerated from the text element. The auto-generation follows two simple rules:

- All letters are changed to lowercase.
- All special characters are replaced by a dash (-) to prevent problems when used in URLs.

You can also customize the URL slug value to suit your needs, which causes the element to switch to manual mode. As a result, the URL slug will no longer react to changes made to the dependent text element.

In need of unique URL slugs?

Kontent.ai doesn't check if a URL slug value is unique across your project. In other words, you can have multiple content items with the same URL slug. This may cause you issues when you use URL slugs to build URLs in your app. To ensure URL slugs of your items are unique, you have several options:

1. Create a middleware that listens to webhook notifications

When a content item is published or its workflow step changes, your middleware [gets notified by a webhook](#) and checks whether the item's URL slug is unique. If it's not, it then updates the URL slug value using the [Management API](#) to something unique. This approach requires you to maintain a database of your URL slugs the middleware can check against.

2. Create a custom element that checks the uniqueness of URL slugs

The [Unique URL slug custom element](#) generates a URL slug based on a text field value and checks whether the slug is unique. In case it discovers that the same URL slug already exists within the project, the element can append a postfix to make the new slug unique.

3. Don't rely on URL slugs only

[Combine URL slugs](#) with either a codename or a content item ID. This leaves you without the need to maintain a list of your URL slugs but requires you to change the structure of the URLs in your app.

Get items by their URL slugs

If you need to get content items by their URL slug values, check out [how to retrieve localized content](#).

Delete dependent text elements

When deleting a text element that is currently used to generate URL slug, there are two options depending on the location of the text element:

- In content types, the dependent text element can be deleted. After deleting the element, you'll be asked to choose a different text element before saving the changes. If you don't have any text element in the current type, add a new text element.
- In content type snippets, the dependent text element cannot be deleted directly. First, in the content type, you need to choose a different text element to generate the URL slug. Then you can delete the text element in the content type snippet.

Element removal cannot be undone

This action is irreversible. Removing an element from a content type or content type snippet removes the element across all content items based on that content type or snippet.

What's next?

- Use the URL slug elements when [setting up your app's URLs and the routing logic](#).
- [Set up preview for your items](#) so that content creators can see their work in context and write with confidence.
- [Reuse content type elements](#) that you need in multiple types of content.
- Set up [a custom asset domain](#) to improve the SEO of your assets.