

# Write down the metadata

December 5, 2022 • Boris Pocatko and Tomas Nosek • 3 min read

Preparing personalized customer journeys can't be done randomly. When you go to your favorite smaller shop on the corner where you go every morning, the owner or employees there most probably know you. The better ones even know your name, maybe even allergies, personality, etc. In smaller, face-to-face shops, this is how personalized customer journeys are done. But how can you do this in your online business?

## Personalize content online

For larger, online companies, personalization depends on quality metadata. Speaking content, do not start designing your metadata based on existing navigation or categorization in existing projects. Take a step back and **have a look at your customer journeys at first and focus on categories and navigation at the end**. Map your content types to your customer journey steps and explore how to effectively target these steps with your content.

The most basic form of personalization can be set up by adding a *Persona* element to the content types aimed at specific personas. In the example, it's the CTA content type. For the *Article*, *Blog post*, and *Widget - Curated content* types, the example enables authors to choose from different *Voice & tone* option that later changes the webpage layout for the viewers.

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\\_metadata.drawio](https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKenticoDocs%2Fkontent-docs-diagrams%2Fmaster%2Fcontent_modeling%2Fsafelife_tutorials%2Fsafelife_metadata.drawio)

A different way to implement personas is to tag content items and specify which tags are typical for a given persona in a separate content item. You can also add [buyer decision](#) stages as a taxonomy. Those should cover your basic needs, but there might be more depending on how targeted you want to be.

## Consider the internal point of view as well

It doesn't matter how many metadata elements you have if no one will end up using them. The process of adding content and metadata should be logical and straightforward:

1. [Limit what content types can be used where](#).
2. **Set up guidelines for all nontrivial or important elements** – use screenshots to show how an element or content type will be shown on various frontends, and describe the voice and tone.
3. **Set up automated tools for [keyword extraction](#)** where possible to minimize manual work. This approach is great for SEO-related elements represented by content type snippets.
4. If you notice other elements being used in the same content types as the SEO metadata, consider moving them into the snippet as well. Those elements could be taxonomies for website navigation, menus, URL slugs, etc.

### Advantages of using content type snippets

Grouping elements into [content type snippets](#) allows you to extend the list of elements in one place and gives your front end developers a unified data set for all content types. This helps to build a highly flexible content model.

## What's next?

Now, you have a content model ready to be used in production. If you want to go through the content model used as the Safelife example, take a look at the final content model diagram:

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\\_final.drawio](https://viewer.diagrams.net?lightbox=1&nav=1#Uhttps%3A%2F%2Fraw.githubusercontent.com%2FKenticoDocs%2Fkontent-docs-diagrams%2Fmaster%2Fcontent_modeling%2Fsafelife_tutorials%2Fsafelife_final.drawio)

Alternatively, you can also [browse through the website](#). Please note that not everything was implemented in there for the sake of simplicity.

The next step is to prepare your content model to go live!

[Go live with your content model](#)