Establish roles working with your content

February 14, 2022 • Tomas Nosek and Martin Turena • 11 min read

Knowing who needs to do each task and where the responsibilities lie is crucial for every content creation process. Without clear understanding of who's doing what, incompleted tasks and bottlenecks can be the results of your content creation efforts. Proper configuration of roles prevents overlapping responsibilities, duplication of work, and wasted time.

A Copywriter, Editor, Graphic Designer, System Administrator, Developer—those are just a few roles that access content and content software within a company. Yet **it is very rare for all the roles to need the same permissions**. By setting up roles in your company, you'll **connect communication silos** and enable your company to work effectively with content.

In Kontent, **every user is assigned a role that defines their permissions** within the app. Role permissions cover all the important aspects from simple viewing of content to accessing advanced features such as managing content types.

Key points

- Use the default *Project manager* role for the initial trial and setup of Kontent. It will make the configuration phase quicker.
- Before you create roles, map your existing or aspirational content creation flow to discover who should be allowed to do what.
- Create a list of what each role can and cannot do. Use content groups for splitting permissions within one content type.

1. Map suitable permissions

Before setting up the roles for your team, it is important to **map out how your organization works** now or how you'd like it to work in the future. Look at the way of working from the point of view of permission areas:

- The administration permissions – for the configuration and maintenance of the content project
- The content permissions – for working with the content within the project

Although there might be some overlap, the goals of the respective areas are quite different. Let's imagine a typical front office. When a customer walks in, the product they are looking for is information on products or perhaps assistance with them. But to achieve that goal, there needs to be a physical building first, with a convenient environment inside. If there is an issue with the office layout or the Internet goes down, it's not your front office specialists who address the problem.

The online project you're building follows the same pattern. Let your content specialists focus on what they can do best and as far as maintenance goes, let your developers or strategists shine.

These are typical tasks that fall in the two respective areas:

Administration	Content
Configure and update the content model	View content
Manage users, roles, and workflows	Create content
Manage subscription plans	Edit content
Manage APIs	Delete content

Publishing permissions

If you're looking for publishing permissions, those are part of the workflow. Workflows are described in the next part of the series.

Who's suitable for administration permissions

These permissions are **typically for administrators and developers of your project**. At the beginning of your project implementation, those are also the people that will have access to Kontent first.

The predefined *Project manager* role is a good start as its permissions are unlimited. This way, everyone that's in the project from the beginning can explore Kontent's possibilities.

For further implementation and later production phases, it's useful to **assign these permissions only to those people that need to change core properties** and are aware of their impact. Changing content types, managing APIs, and other permissions may break your apps if mishandled.

See what each permission allows in the reference.

Subscription admin

The only users that have more permissions than the *Project manager* are the *Subscription admins*. ***Subscription admins* are users that have full permissions over the subscription** that contains the project. These permissions include changing the subscription plan and payment details. The *Subscription admin* is the *Project manager* in all the projects within the subscription to guarantee full permissions over the whole Kontent configuration.

For the *Subscription admin*, choose the content owner at your company, the technical owner of Kontent, or both. As *Subscription admins* can delete projects, do not use the admin for more users than needed.

Who's suitable for content permissions

The smaller your content-related team is, the easier your content permission model is going to be. In such case, it's typical to let people access most of the content and allow them to do nearly all actions with the content as their tasks overlap.

However, with more complex teams and more content produced, you'll typically **divide people into groups** that will always be allowed to create, modify, or delete specific parts of your content only. To do that, it's

important to understand your organization and how you'd like your content teams to interact. Try to answer this list of questions:

1. Who's the main content creator?
2. What content is edited by who and when?
3. What is the process of content creation? Which steps does the content go through before it's published?
4. What is the approval process before the content is published?
5. How will the respective team members collaborate? Will they be expected to pick content for editing, or will specific content be assigned to them?

Answers to those questions are now the base for your content permission setup.

> **Allow changes to assigned content only**
>
> In Kontent, you can assign content directly to users. If your team's strategy involves assigning specific tasks to specific people, you can configure that in roles as well.
>
> For example, an external contributor, such as a translator, doesn't need to see or modify all content. You'll rather pick and choose what content the translator is supposed to work on and once complete, you will assign other content to them.

2. Design roles

To identify roles within a project, you might want to use a tool such as the Responsibility assignment matrix (RACI) to help you. This way, it will become clear what each role is accounted for and what their main tasks are.

When designing roles, take your content model into account as well. If suitable, you can modify the model to reflect the creation flow. The goal is to **create a three-column table like this**:

Role	Can do	Cannot do

Then, go through each answer you gave for the questions above and **create a new row in the table or specify an existing row based on the information in the answer**.

For example, from this answer:

"*There are 2 team managers who review the content and manage metadata. Neither of them, however, manage the navigation. The navigation is only edited by the company's developers.*"

You can get two rows:

Role	Can do	Cannot do
Team manager	View, edit, create, and delete all content items	Delete, create, edit, and view metadata and navigation items
Developers	View, edit, create, and delete all navigation items	

This table says that the first role will be able to view, edit, create and delete all content items except metadata and navigation items. They won't see those at all. The second role will be able to view, edit, create and delete all navigation items. That means that other content items won't be visible to them without having to state it in the *Cannot do* column.

Role naming

There are multiple approaches you can choose from. Pick what will serve your company the most. For example, it can mimic your RBAC model, it can be loosely based on the people's positions in the organization, or it can express the permissions that roles possess. You can also use a hybrid naming where you combine more approaches.

Based on the position in your organization

You can **name your roles based on the roles people have at your company**. For example, if a person is a Copywriter, you can create a *Copywriter* role.

The advantage of this approach is that you don't have to maintain organization charts.

The disadvantage of this approach is that often, there are exceptions. For example, a person from the team should be allowed to do more as they're the substitute of a team leader, yet they are in the same position as the others.

Based on role permissions

Going from the other side, you can **group people based on what they're allowed to do**. For example, if there are eight Copywriters and three Editors, not only can they create a soccer team, but they can also share one role, such as *View, edit, create, and delete all*.

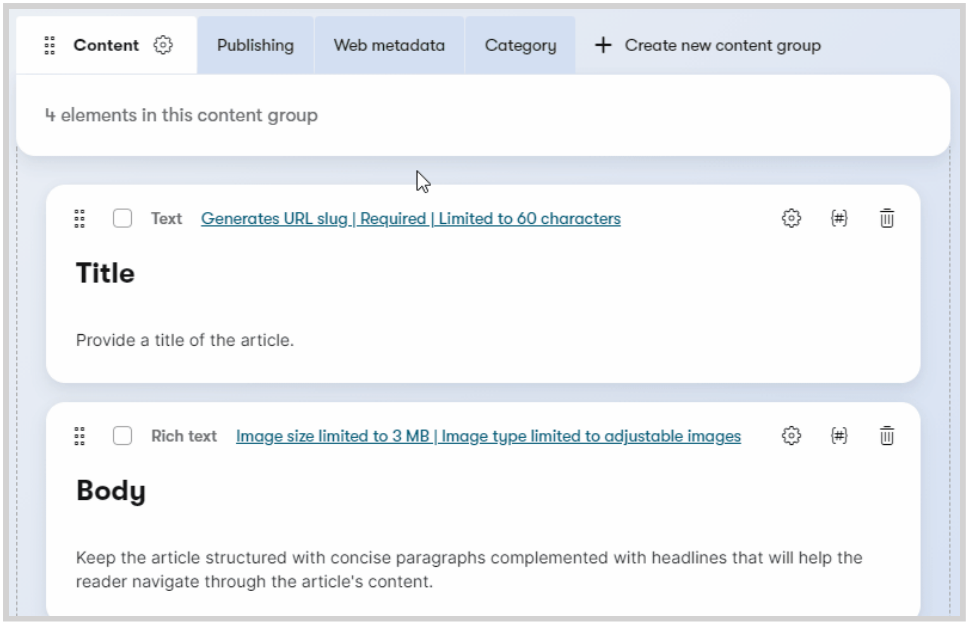
The advantage of this approach is that you immediately know from the user's role, what they're allowed to do.

The disadvantage of this approach is that when changing permissions for a team, you need to go through all the users from different teams as they have the same set of permissions.

3. Configure roles in Kontent

Now's the time to **create all the roles in Kontent**. Go through the created table one row at a time, and create roles based on them.

While doing so, **think about using content groups as well**. Content groups allow you to put related content elements together and to allow specific people to work on those elements only. This can make it even easier for your colleagues to work on content while preventing them to change what you don't want them to change.



What content groups can look like in a content type

What's next?

You have now prepared your project for users with different permissions. By that, you started the first step on the way to successful team collaboration. When you need to change the roles, come back to the diagram or table of roles that you were drawing, and adjust it. Before creating the roles in Kontent, think whether the new setup still makes sense for all existing roles.

To continue setting up your project to ensure great collaboration:

- Create suitable workflows when you're finished with configuring roles.
- Check how roles are set up in a complete example of team collaboration.