CHAPTER 9

Building a Smart Administration User Interface

One of the often-overlooked areas of building and deploying Drupal sites is the site and content administrators user interface. We often focus on the “pretty” public facing side of the site and hastily throw together a basic off-the-shelf interface at those who are responsible for authoring and maintaining content, and those who manage the site. The result of our lack of focus on the backends of our sites is frustration by those who are key to the overall success of the sites that we build and deploy. Making it easier for content creators and site administrators to do their jobs is relatively easy and it’s the focus of this chapter.

Use an Administration Focused Theme

The administrative interface of a Drupal site is structured differently than the visually rich frontends that we present to site visitors and is often the forgotten frontier of building and deploying Drupal sites. Our focus is typically on our targeted site visitor and we pay little attention to the interface that those use who create and manage content and support the site on a daily basis. We often use the frontend theme as the basis for our administrative backend, but the reality is that the structure of the administrative interface is inherently different than the frontend and, without specifically addressing the structural differences between the front- and backends of our sites, the result is less than optimal for our administrators. Fortunately, there is a solution and that is to use a separate theme that is built specifically with the administrative interface as the focus of its design.

There are several administrative themes that may be freely downloaded from `drupal.org` and in this section we take a look at a few of the most popular admin themes for Drupal 8.

Enabling Different Admin Themes

Admin themes are enabled on the Appearance page. In order for a theme to appear in the list of available options, you must first download and install that theme just as you do with a frontend theme, with the exception of not setting that theme as the default theme. After downloading and installing it, you may specify the theme as the admin interface by selecting the theme from the list of enabled themes that are shown in the administration theme section at the bottom of the Appearance page (see Figure 9-1). Select the theme to use from the drop-down list and click the Save Configuration button.

T. Tomlinson, *Enterprise Drupal 8 Development*, DOI 10.1007/978-1-4842-0253-1_9

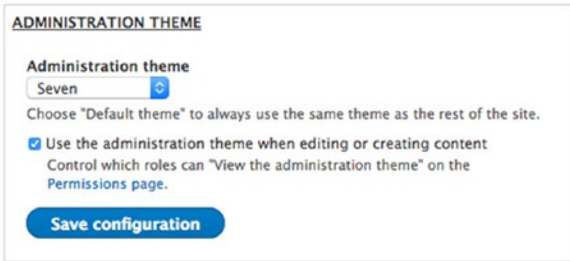


Figure 9-1. *Specifying the administration theme*

There are several off-the-shelf admin themes. The following sections demonstrate a few of the most popular Drupal 8 admin themes.

The Seven Theme

Drupal 8 ships with a theme that was built to address the uniqueness of the administrative interface, called the Seven theme.

To demonstrate the benefits of using a theme that is specifically constructed for the administrative interface, this example starts by enabling Bartik as the administrative theme. If you visit the Extend page, you'll immediately see issues (see Figure 9-2) that must be addressed, such as the frontend blocks that are appearing on the admin pages, the overflow issue that appears in the Core section, and the tabs that seem to be disconnected from the main section of the page that the tabs are referring to.

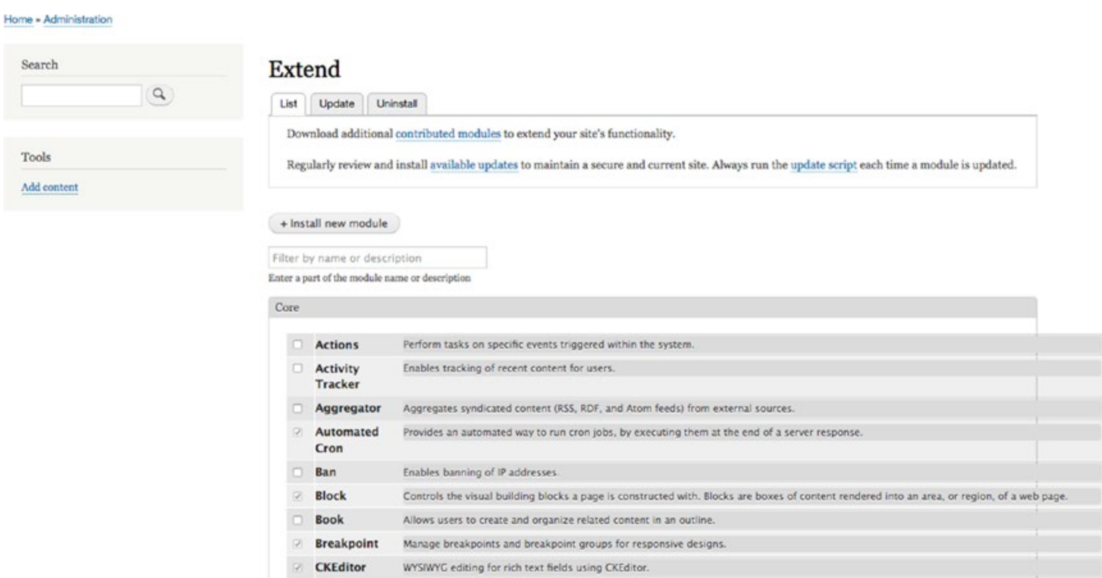


Figure 9-2. *The Extend page using the Bartik theme*

Instead, what if we enable Seven as the administration theme and then revisit the Extend page? We'll see a significantly cleaner interface (see Figure 9-3) than what the Bartik theme presented. The tabs are connected to the sections of the page that they are related to, the interface is less confusing, and the overflow issues have been addressed.

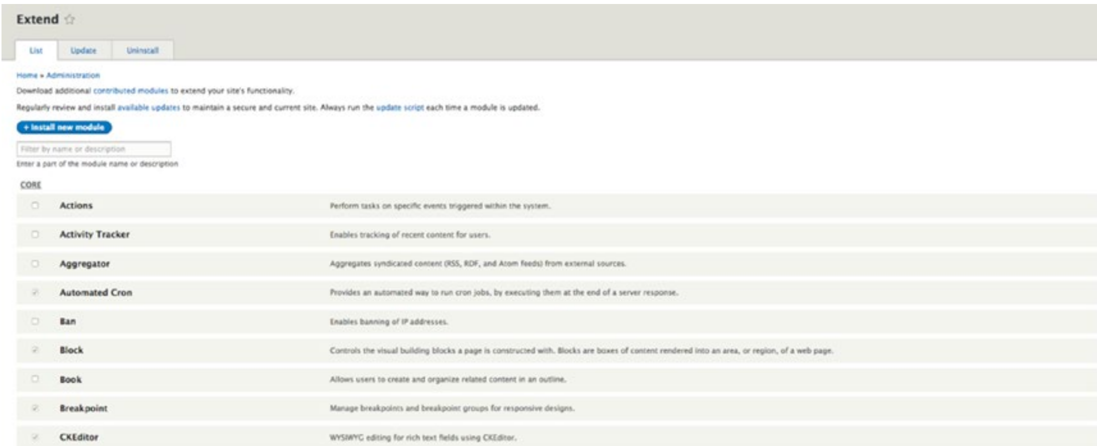


Figure 9-3. *The Extend page using the Seven theme*

If we navigate around the administrative interface, switching between Bartik and Seven, we'll find several more instances where Bartik's CSS just doesn't work very well with the administration pages, whereas Seven works perfectly with those pages. Another great example is the `node/add/article` page. Enable Bartik as the administration theme and notice the organization of the elements on the node edit form as compared to a much cleaner and well-organized interface as presented by the Seven theme.

While Seven is not as "pretty" as Bartik, we're not trying to impress our content editors and site builders with the beauty of the administration theme. We are only trying to present those pages in a form that makes it quick and easy for them to do their jobs.

The Adminimal Theme

There are other administration themes that are not shipped with Drupal core. Visit `drupal.org/project/themes` and select 8.x from the Core compatibility select list. Then enter `administration` in the Search Themes text box and leave the Sort By option set to Most Installed. The first theme that appears is the Adminimal theme and it boasts over a million downloads. Like Seven, Adminimal is focused on Drupal's administrative interface and it provides advanced features such as a fully responsive design, color-coded buttons for common admin tasks, and table highlighting. To demonstrate Adminimal, we download and enable the theme and set it to the default administration theme.

At first glance, the Adminimal theme looks very similar to the Seven theme, with subtle differences around button styles and a slight improvement in the visual appearance of the admin toolbar.

There are other themes that you may consider using as your administration theme. Visit `drupal.org/project/themes` and look for other candidates.

Update the Administration Menu

The default administration menu works well; however, there are opportunities for improving the user interface provided by the off-the-shelf capabilities of the menu.

One of the shortcomings of the off-the-shelf administration menu is the lack of support for drop-down menus. The default menu only displays the top two levels of the menu, requiring an administrator to click through to the subordinate options by visiting a page. The Admin toolbar module (`drupal.org/project/admin_toolbar`) resolves this issue with mobile-friendly drop-down menus that give administrators direct access to detail-leveled tasks (see Figure 9-4).

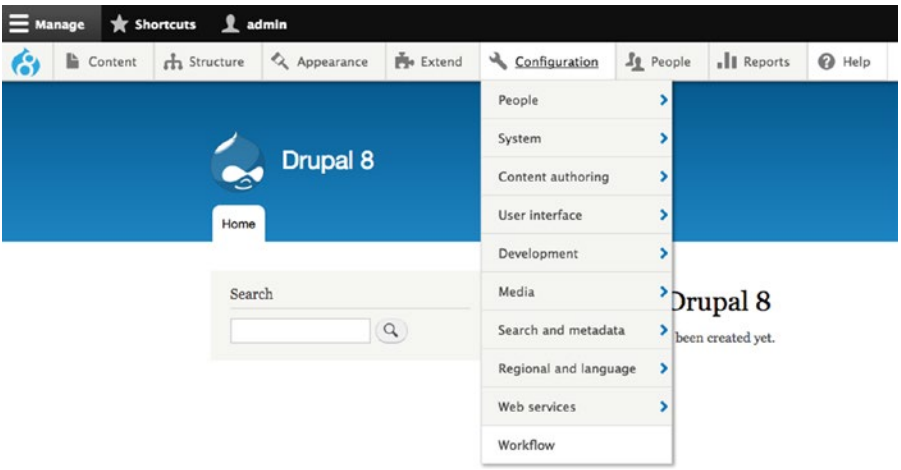


Figure 9-4. *The admin toolbar module enables drop-down admin menus*

Simplify Content Types

The most frequent administrative activity on a Drupal site is creating and editing content and it's often the most frustrating user experience for administrators on a Drupal site. There are a few simple techniques for significantly improving the interface and the experience.

Organizing the Fields

By default, Drupal lists all of the fields on your content type into a long lists that may require the content creator to scroll down through several fields to get to the one they want. In the example shown in Figure 9-5, the Sample Content Type has a title, body, and 12 additional fields, requiring the content editor to scroll down in order to see all of the fields associated with this sample content type. While 12 may seem like a large number of fields for a content type, the reality is that I've had clients with content types that have several dozen fields, making it very difficult for content editors to find the fields they are looking for.

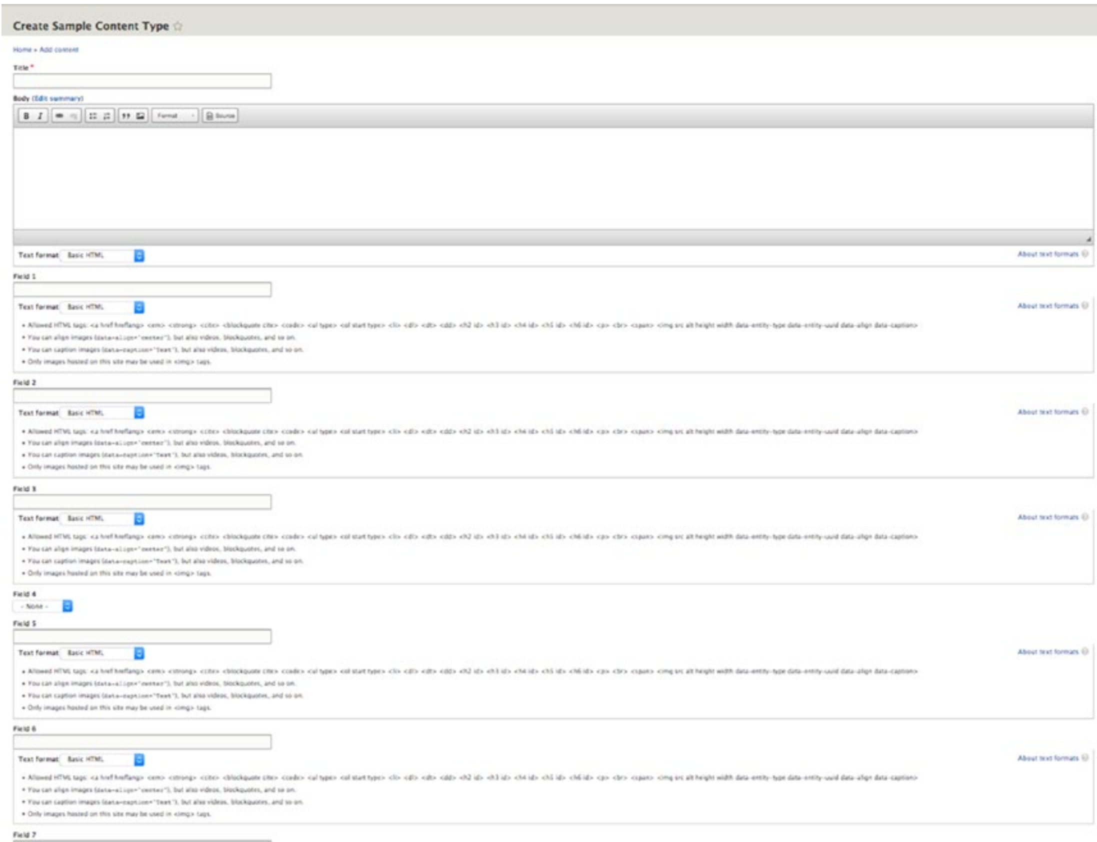


Figure 9-5. *A content type with several fields*

An effective way of organizing and presenting fields is to install the Field Group module (`drupal.org/project/field_group`) and logically group the fields into vertical tabs.

After downloading and enabling the module, you're presented with an additional option on the Manage Form Display page for adding a group (see Figure 9-6).

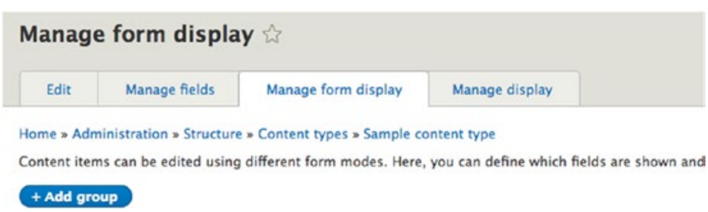


Figure 9-6. *The Add Group option*

Clicking on the Add Group button presents a list of types of groups that you can create on the form (see Figure 9-7). This example focuses on using the Tabs option, as it works well for grouping and hiding options behind vertical and horizontal tabs.

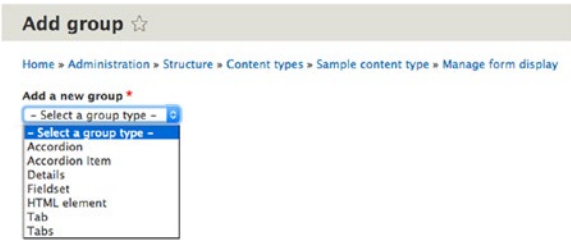


Figure 9-7. *The group options*

We select Tabs from the list and are presented with a Label text box, where we specify the label that will appear on the tab. In this case, we use General as the tab as put all of the individual tabs that we'll be creating into that one container. Click the Save and Continue button, which then displays a form where you can specify whether the tabs will be presented vertically along the left side of the content type fields, or horizontally across the top of the page. We select vertical here and leave the ID (the CSS ID) and the Extra CSS classes fields blank. Click the Create button to continue.

The next step is to create the individual tabs that the fields will be grouped under. For demonstration purposes, we create four separate tabs, one for the title and body fields and three additional tabs for the other fields. Follow the same process as before, clicking the Add Group button followed by selecting the Tab option (not the Tabs option).

After creating all of the tabs, drag and drop the Tabs field that you created at the beginning of this process to the very top of the list of fields, followed by each the individual Tab fields that you subsequently created. Then drag and drop each of the fields beneath the appropriate tab, as shown in Figure 9-8. Then click the Save button at the bottom of the page to commit these changes.

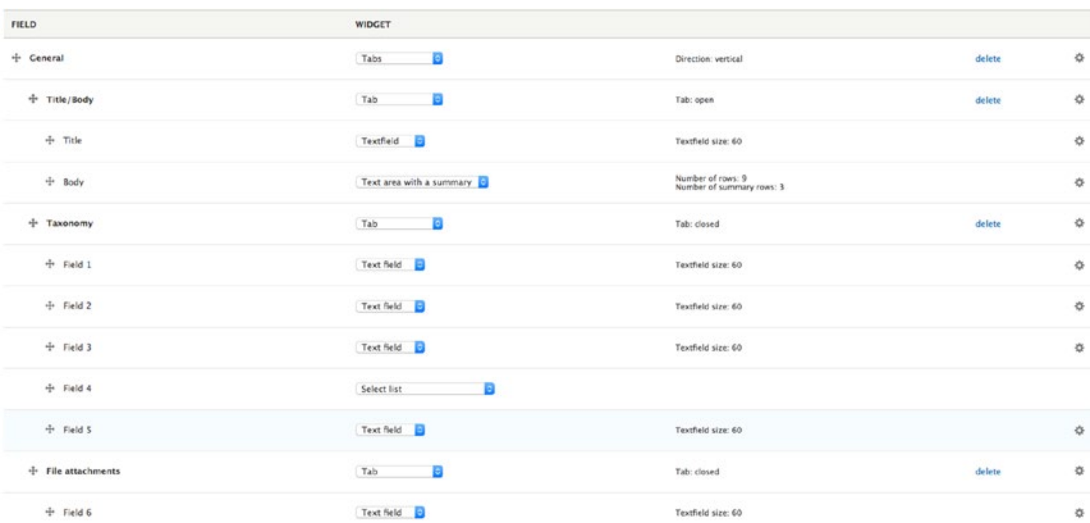


Figure 9-8. *Arranging the vertical tabs and fields*

Note the nesting of the tabs and the fields. This is important, as that dictates which fields appear in each of the tabs, and that the individual tab fields appear in the vertical tabs container.

After organizing the fields into tabs, the user interface for this content type is much cleaner and simpler, without the need to scroll down to find fields.

There are also other improvements that you can make to the overall look of the content form, such as removing the fields from the settings section that are typically left as the default values by content creators. The additional settings can be found in the right column of Figure 9-9, beginning with the option to create a new revision. The Simplify (`drupal.org/project/simplify`) module provides an administrative interface that makes it easy to disable those fields without having to override fields through custom code or hide them with CSS.

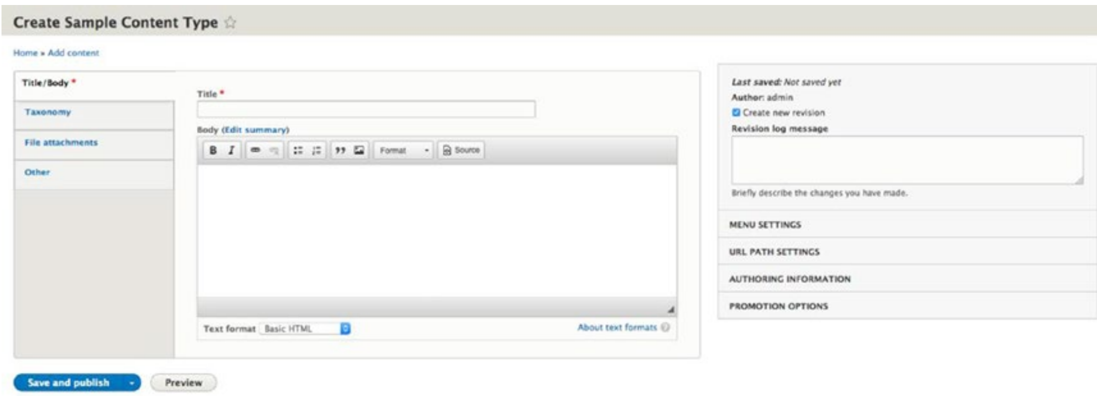


Figure 9-9. *An improved content creation form*

Let's download and enable the Simplify module to demonstrate its capabilities. After enabling the module, navigate to Configuration ➤ User Interface ➤ Simplify, where we're presented with a list of options to disable fields from the user interface (see Figure 9-10).

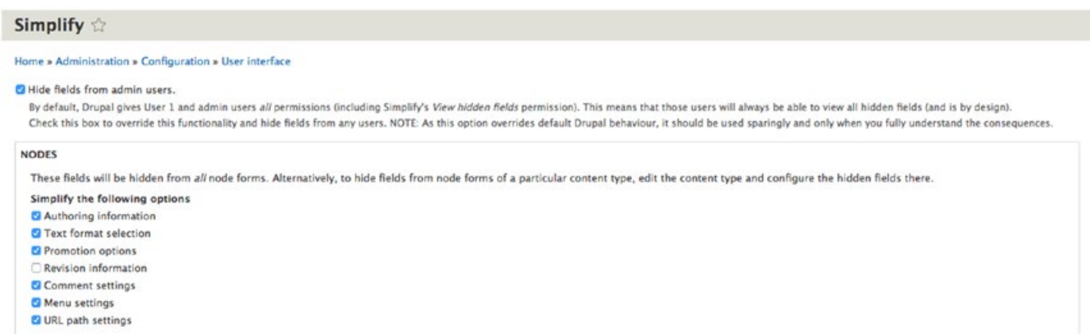


Figure 9-10. *Administering the Simplify settings*

For demonstration purposes, we've checked the box to hide the fields from site administrators, as well as hidden all of the fields other than the ability to create a revision. After clicking the Save Configuration button, we return to the node creation form for the sample content type, where we no longer see the list of fields in the right column other than the ability to create a new revision (see Figure 9-11).

Figure 9-11. *The simplified content creation form*

Using Hierarchical Select

When presenting content editors with a list of taxonomy terms, it is often desirable to improve the user experience over the standard off-the-shelf interface. For demonstration purposes, we've created a taxonomy vocabulary named Sample Terms and used the Devel module's tool for generating 100 taxonomy terms. We added a new field to the Sample Content type that is a Taxonomy Term Reference field that uses the Sample terms vocabulary and presents the list of terms as a select list. The resulting field, when presented to a content editor, appears as shown in Figure 9-12.

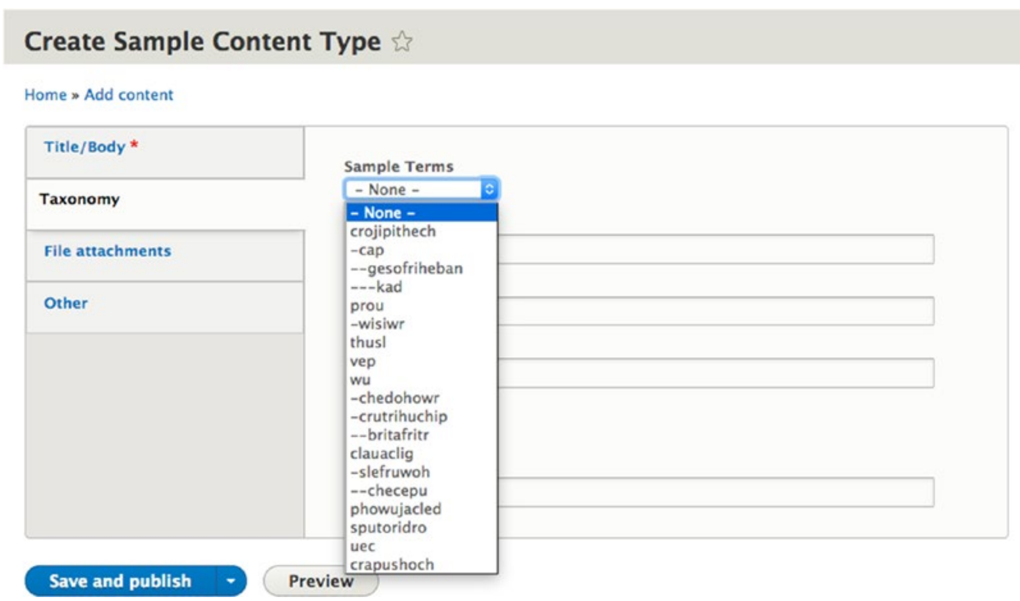


Figure 9-12. *Selecting a taxonomy term from a long list of hierarchically organized terms*

While this approach works, there is a better user experience using an off-the-shelf module such as the Client-Side Hierarchical Select (`drupal.org/project/cshs`) or Simple Hierarchical Select (`drupal.org/project/shs`) modules. For demonstration purposes, we download and enable the Client-Side module, although both modules work well.

After enabling the module, navigate to Structure ➤ Content Types and click on the Manage Form Display tab. Change the widget for the Sample Terms field to Client-Side Hierarchical Select, as shown in Figure 9-13.

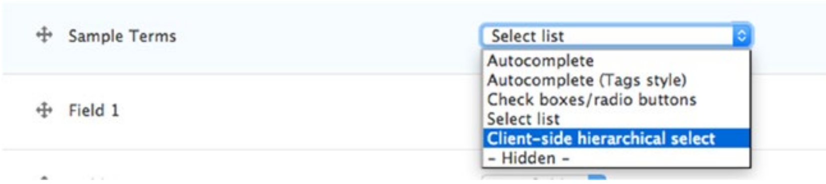


Figure 9-13. *Changing the widget type to Client-Side Hierarchical Select*

After changing the widget type, we create a new content item from the Sample Content type that we created. When we come to the Sample Terms field, we are now presented with a list of the top-level terms and, after selecting a top-level term, we are then presented with the second level terms in a separate select list. For terms that have multiple levels, the process continues until we select the lowest level term, as shown in Figure 9-14.

Figure 9-14. *The hierarchical select field*

The hierarchical select tools provide a simplified and better user experience than the off-the-shelf select list for hierarchically ordered taxonomy terms.

Using Field Collections

Another area for improving the administrative interface for content types is to use the Field Collection module (`drupal.org/project/field_collection`). The common use case where field collections provide a better administrative user experience is the case where you have a set of fields that repeat two or more times in a content type. An example of that use case would be a content type focused on books. On the book content type there are fields that provide the editor with the ability to create one to many authors, where each author has a name field, a birth year, and a death year. While it would be possible and plausible to assume that a book has no more than 10 authors and you could create 10 separate fields for each possible author, field collections provide a more elegant approach to solving that use case.

To demonstrate the use of field collections, we download and enable the module. We navigate to Structure ➤ Content types, where we click on the Manage Fields link in the Operations column. On the Manage Fields page, we click Add Field and select Field Collection from the list of available options for Add a New Field. We enter `Author` in the label field and click the Save and Continue button.

On the next page, we change the Allowed Number of Values from 1 to Unlimited. You could specify a maximum number if you want, but for demonstration purposes, we'll set the value to Unlimited followed by clicking the Save Field Settings button. On the Author Settings page, we'll leave the default values and click the Save Settings button. Next, we click on the Manage Form Display tab and reposition the author field directly beneath the title field. Then we click on the Manage Display button, where we position the author field above the body field and change the Format to Field Collection Items so that all of the fields appear when viewing the content. We then click the Save button to commit all of the changes.

At this juncture, we now have an author field collection assigned to the sample content type, but we have not yet defined the fields that make up the collection. To define the fields, navigate to Structure ➤ Field Collections and click on the Manage Field Collections page (see Figure 9-15).

Figure 9-15. *The Manage Field Collections page and the author field collection*

To add fields to the author collection, click on the Manage Fields button. On the Manage Fields page, add the name, birth year, and death year using by clicking the Add Field button and adding the fields just as you would on a normal content type. After adding the fields to the collection (see Figure 9-16), we're not ready to see the field collection in action.

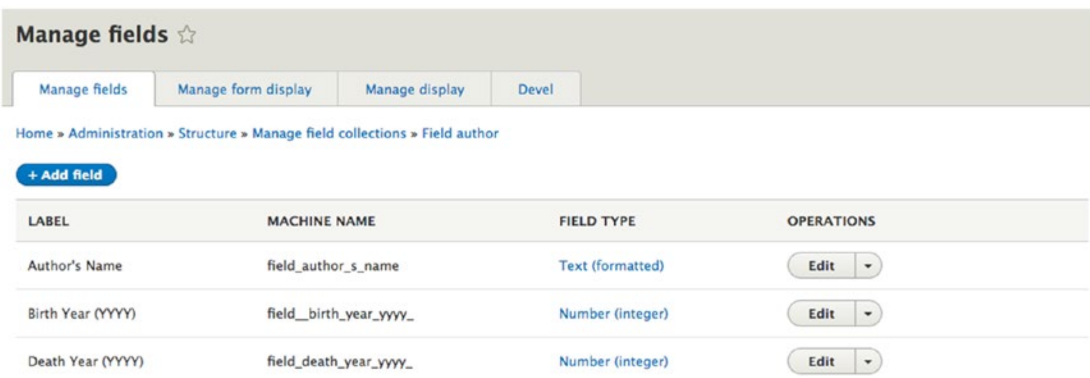


Figure 9-16. *The fields associated with an author field collection*

We create a new content item using the sample content type and we now see the author field and the ability to enter the author's name, birth year, and death year. We also see the ability to create additional author field collections by clicking on the Add Another Item button (see Figure 9-17). We can also reposition author field collections by clicking on the + icon and dragging and dropping individual author field collections in the order that we want them to appear.

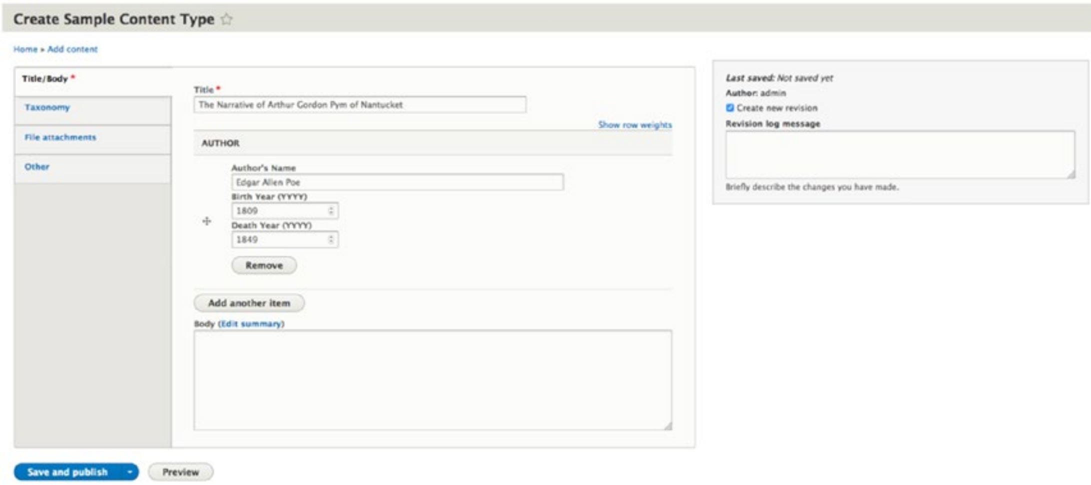


Figure 9-17. *Adding authors to a content item*

After clicking the Save and Publish button, we can now see author fields on the published node (see Figure 9-18).

Figure 9-18. *The author fields appearing on a published content item*

Use the Workbench Module

The site visitors who spend the most amount of time on any given Drupal site are typically those who are responsible for creating and managing content. This group is also the one who often receives the least amount of attention when it comes to user experience enhancements to make it easier for them to do their jobs. While I've addressed several improvements to the node creation and editing interface elsewhere in this chapter, there are other off-the-shelf modules that enhance the general editorial experience. This section focuses on three of those modules:

- Workbench (`drupal.org/project/workbench`). This module provides a unified and simplified user interface for users who only have to work with the content.
- Workbench access (`drupal.org/project/workbench_access`). This module provides the ability to control who as access to edit any content based on an organization's structure, not the web site structure.

- Workbench moderation (drupal.org/project/workbench_moderation). This module provides a customizable editorial workflow that integrates with access control.

To follow along, install all three modules using the standard process for downloading and enabling the modules from drupal.org.

The Workbench Module

The standard off-the-shelf Drupal 8 interface for content authors is the Content page (see Figure 9-19).

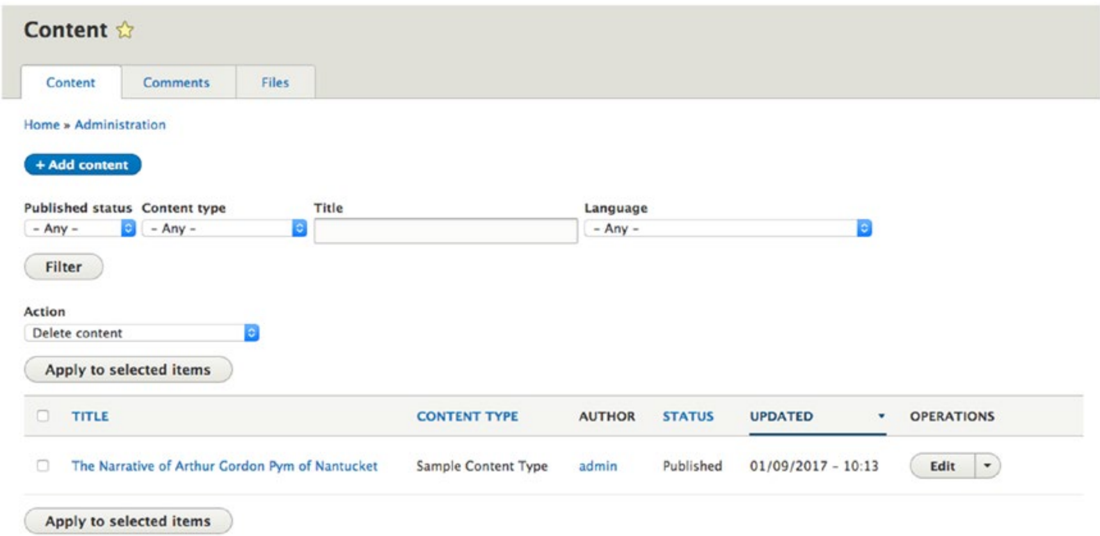


Figure 9-19. *The Drupal 8 standard content page*

While the Content page succeeds at providing a basic interface for all content editors, the Workbench module provides an enhanced interface that simplifies the user interface presented to content editors.

After enabling the Workbench module, a new Workbench menu items appear in the administrator's toolbar. Clicking on the Workbench link reveals the My Workbench page, as shown in Figure 9-20. The primary difference between the off-the-shelf Content page and My Workbench is the addition of the Your Most Recent Edits section, which highlights content that the logged in editor created or updated. This functionality addresses a common use case of an editor wanting to return to an article they were working on without having to search or filter through a long list of content items. For smaller sites with a single content editor, the feature may not be as useful as it is for large organizations with several content editors.

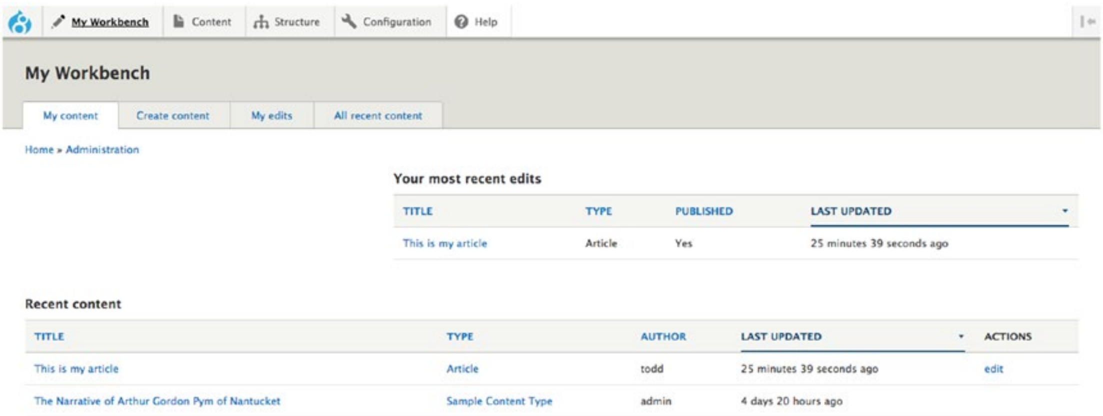


Figure 9-20. *The My Workbench page*

The next enhancement presented on the My Workbench page is the My Edits page, which is accessible by clicking on the My Edits tab. The My Edits page is similar to the Your Most Recent Edits section of the My Workbench page, with the notable exceptions that the list on the My Workbench page is limited to the last five content items that the logged-in editor has created or updated, and the My Edits page also provides the ability to filter on title, content type, whether the content is published, and setting the number of content items that will appear per page (see Figure 9-21).

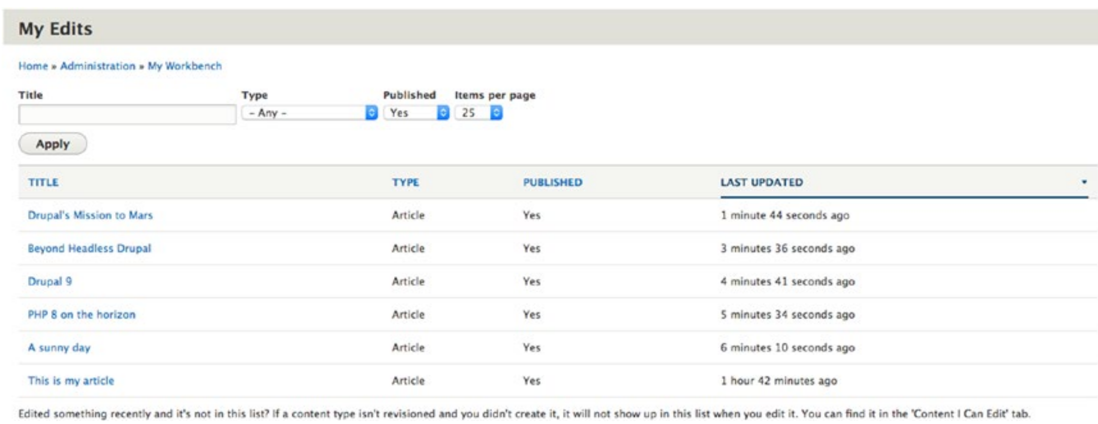


Figure 9-21. *The My Edits page*

For those situations where the content editor wants to see all content additions and updates, clicking on the All Recent Content tab on the My Workbench page displays a list of content nearly identical to the off-the-shelf Content page, with the exception of not having access to the bulk update capabilities that are available on the Content page. In most cases, restricting access to the bulk updates capabilities is a wise choice, as the bulk updates feature, while powerful, can also quickly wreak havoc when someone erroneously applies a mass change.

The Workbench Access Module

The Workbench Access module solves a common problem of how to restrict the ability for content authors so that they only have access to content in their specific area of responsibility. Take for example a company that is structured into the following departments:

- ACME Company
    - Marketing
    - Sales
    - Manufacturing
    - Engineering
    - Customer Service and Support
    - Human Resources
    - Training
        - On Ground Training
        - Online Training
        - Technical Publications and Manuals

Content editors in the ACME company who are part of the Marketing department should not have the ability to create or edit content that pertains to the Human Resources department, and vice versa. Within the ACME company there may also be content editors who have the ability to author content across the three divisions of the training department, as well as content editors who only have access to content in the Technical Publications and Manuals section of the site. The ability to restrict access is the core feature that is provided by the Workbench Access module.

Setting Up Workbench Access

The process of setting up Workbench Access centers on creating the means for defining the editorial sections of the site, the access permissions that you want to grant to editors, and the roles of the users who will be creating and editing content.

The Workbench Access module supports two venues for creating sections, a taxonomy vocabulary or a menu. While both work equally as well, there are advantages to using Taxonomy that span beyond Workbench Access, specifically the ability to use that same vocabulary to enable views to filter and display content based on the section of the site. To demonstrate the capabilities of Workbench Access, I'll use the ACME Company structure defined in the previous section as the basis for a new taxonomy vocabulary named Site Structure (see Figure 9-22).

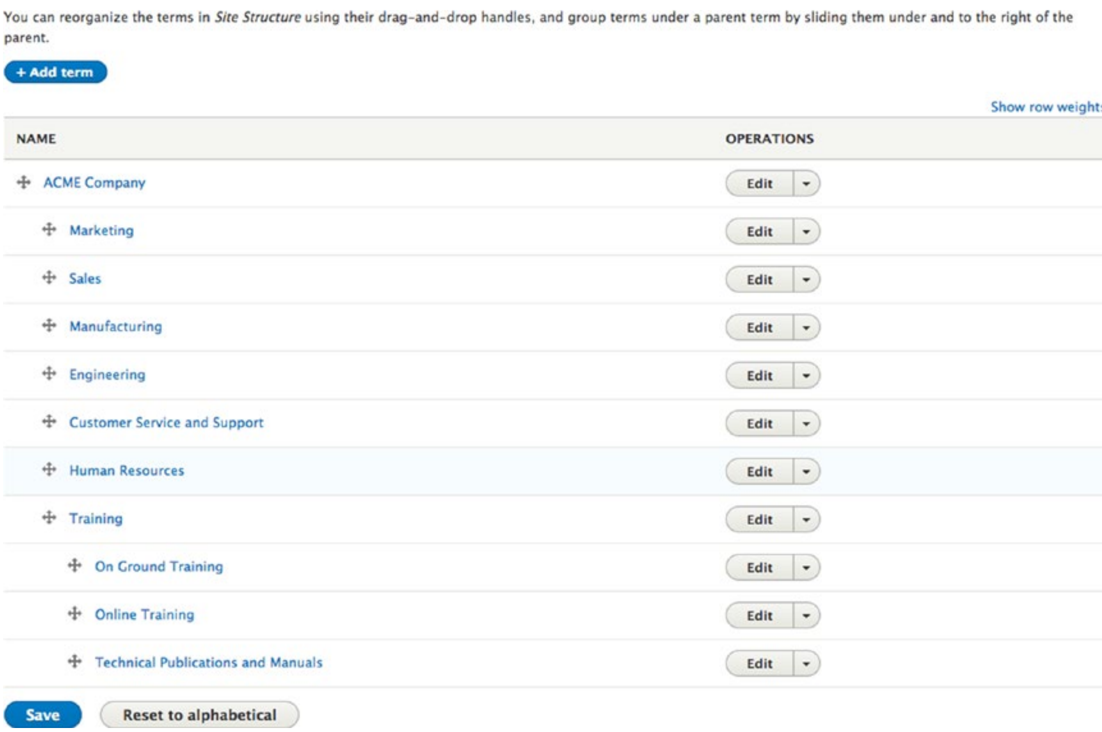


Figure 9-22. *The Site Structure taxonomy vocabulary*

With the site structure defined the next step in the process is to set the permissions for the Workbench Access module. The permissions that may be set are as follows:

- *Administer Workbench Access settings*: Allows users to configure Workbench Access access schemes and sections.
- *Assign users to Workbench Access sections*: Allows users to assign editors to sections. (Note that these editors must have the Allow All Members of This Role to be Assigned to Workbench Sections permission.)
- *Allow all members of this role to be assigned to Workbench Access sections*: Allows a user to be assigned as an editor of a section. This permission is used to check whether a user can access Workbench Access forms and features.
- *Batch update section assignments for content*: Allows a user to access the batch update form at admin/content.
- *View Workbench Access information*: Allows users to see information and messages related to Workbench Access, particularly section assignments of content pages. Useful for debugging and support.
- *View taxonomy term pages for Workbench Access vocabulary*: Workbench Access can create its own vocabulary for data storage. Typically, this vocabulary should not be shown to site visitors. This permission restricts access to taxonomy pages (`taxonomy/term/%`) defined by Workbench Access. Normal access to custom vocabularies is not affected. Give this permission only to roles that need to view these term pages, effectively treating them as standard taxonomy terms.

With the taxonomy in place to facilitate assignment of editorial users to specific sections, the next step in the process is to add a field to those content types that you want to be under access control, linking the Site Section vocabulary to the content being authored. We'll add a new taxonomy term reference field to all of the content types on the site, using the site section vocabulary as the source of terms.

With the link between the content types and the vocabulary used by Workbench Access in place, the next step is to configure Workbench Access. Navigate to Configuration ➤ Workbench Access. Set the Active Access Scheme to Taxonomy, check the box for Reset Assigned Fields, check the Site Structure checkbox in the Taxonomy Editorial Access Options section, and click the Set Active Scheme button (see Figure 9-23).

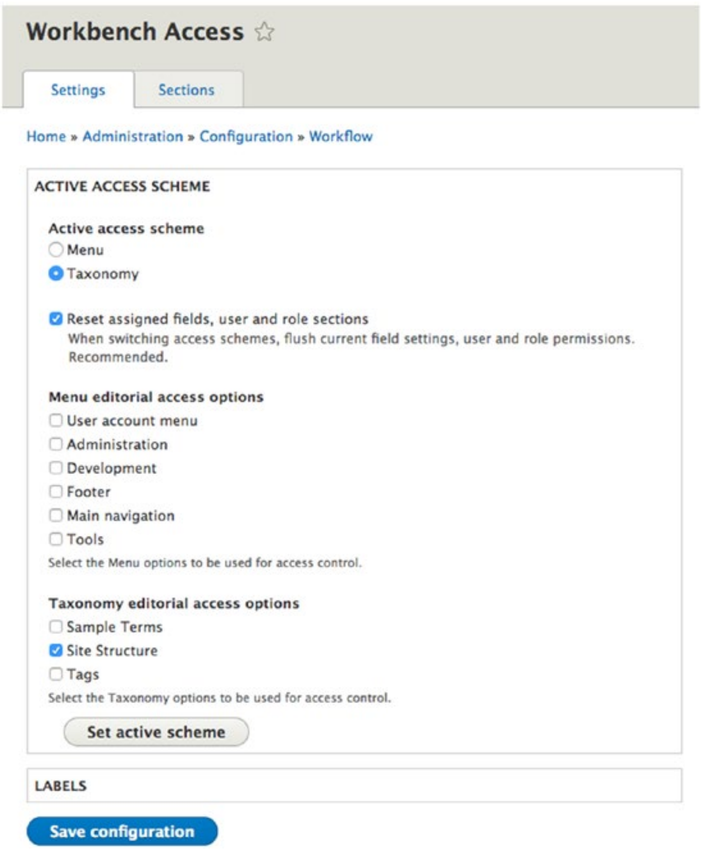


Figure 9-23. *Configuring Workbench Access*

After clicking the Set Active Scheme button, a new set of options is displayed in the Scheme Settings section. The options are whether to enable access control for each of the content types that are enabled on the site and the field that is used as the mechanism for controlling access. In this case, it's the Site Section vocabulary field that we added to each of the content types. We check the box for each content type, select the Site Section field, and then click the Save Configuration button (see Figure 9-24).

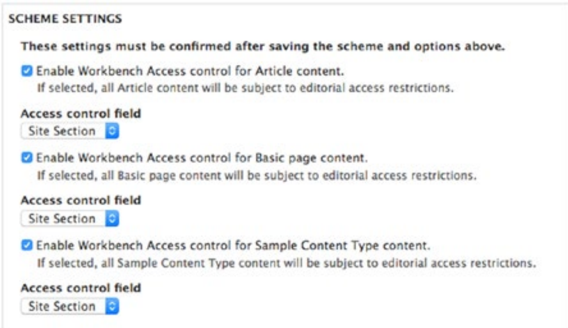


Figure 9-24. *Setting access control on each content type*

Setting Up Roles and Permissions

The Workbench Access module provides the ability to assign individual users to site sections as well as all users in a specific role. For smaller organizations, assigning individual users may be a viable solution, whereas for larger organizations with several content editors, a role-based solution may be easier to manage and maintain.

For the ACME company example, there are specific roles based on department (e.g., Sales, Marketing, Human Resources) as well as enterprise roles such as enterprise editor. We create a role for each department as well as the enterprise editor by navigating to People ➤ Roles and clicking on the Add Role button (see Figure 9-25).

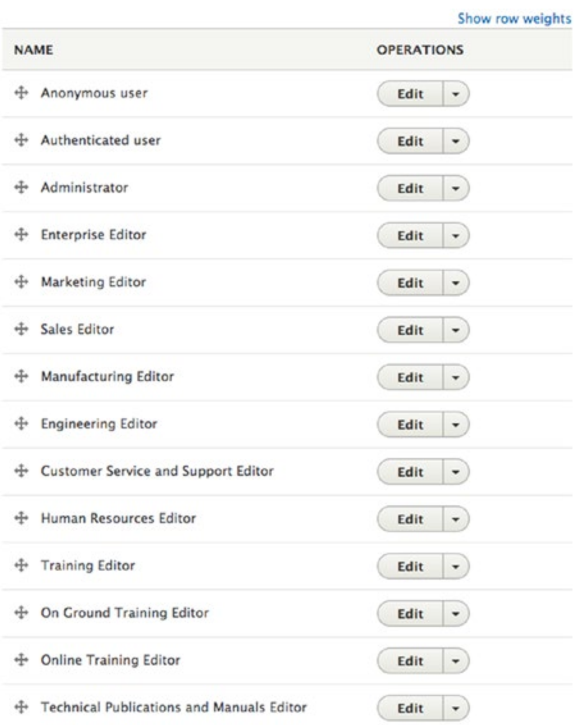


Figure 9-25. *All departmental level roles defined*

With the roles defined, the next step is to set the permissions for each role so that users assigned to that role will have access to the Workbench tools and the assignment of those roles to specific sections. Click on the Permissions tab at the top of the Roles page and enable the permissions for each role as defined in Table 9-1.

Table 9-1. *User Permissions to Set for Workbench Access*

Section	Permission
Node	Access the Content Overview page
Node	View own unpublished content
Node	Article: Create new content
Node	Article: Delete own content
Node	Article: Delete revisions
Node	Article: Edit own content
Node	Article: Revert revisions
Node	Article: View revisions
Node	Basic page: Create new content
Node	Basic page: Delete own content
Node	Basic page: Edit own content
Node	Basic page: Revert revisions
Node	Basic page: View revisions
Node	Sample Content Type: Create new content
Node	Sample Content Type: Delete own content
Node	Sample Content Type: Delete revisions
Node	Sample Content Type: Edit own content
Node	Sample Content Type: Revert revisions
Node	Sample Content Type: View revisions
System	Use the administration pages and help
System	View the administration theme
Toolbar	Use the administration toolbar
Workbench	Access My Workbench
Workbench Access	Allow all members of this role to be assigned to Workbench Access sections
Workbench Access	Assign users to Workbench Access sections

With the permissions in place, the last step in the process is to assign user roles to Workbench Access sections. Navigate to Configuration ➤ Workbench Access and click on the Sections tab on the Workbench Access page. The Sections page (see Figure 9-26) shows a listing of all the sections defined in the Site Sections taxonomy vocabulary along with the number of editors (user accounts) and roles that are enabled on that section.

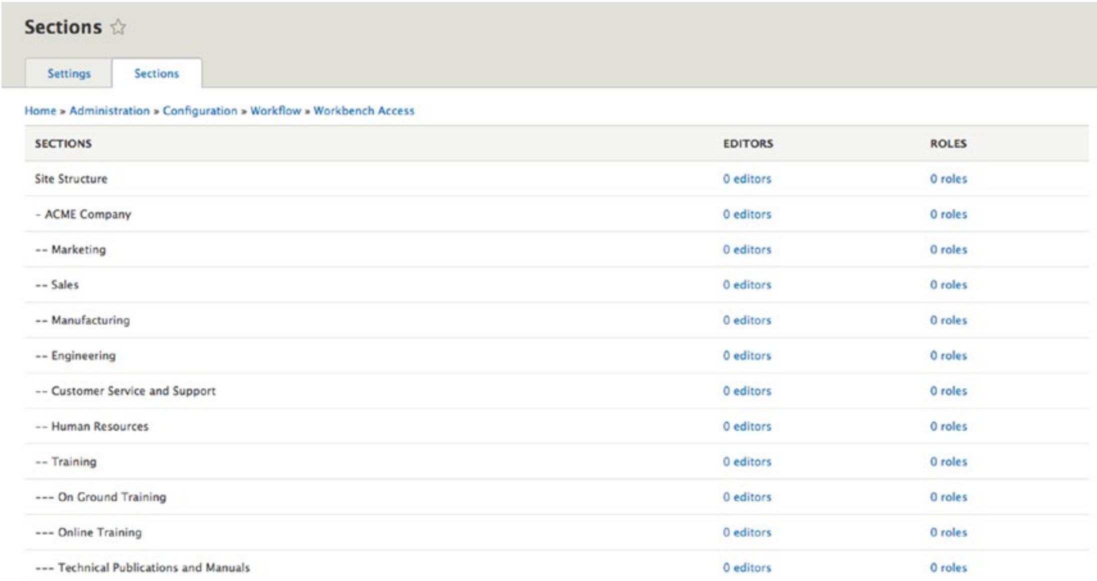


Figure 9-26. The assignment of editors and roles to Workbench Access sections

We can start the process by assigning the administrator, enterprise editor, and marketing roles to the Marketing section. Click on the 0 roles link for the Marketing section, which displays the list of all the roles enabled on this site. Check the two roles and then click the Submit button (see Figure 9-27).

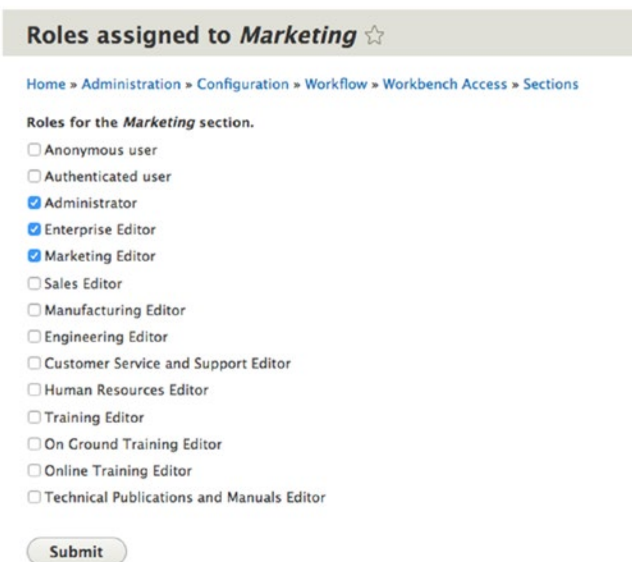


Figure 9-27. Assigning roles to Workbench Access sections

We can continue the process by assigning the various roles that should have access to each of the site sections. When we have completed the process, the Sections page now shows the total number of roles assigned to each section (see Figure 9-28).

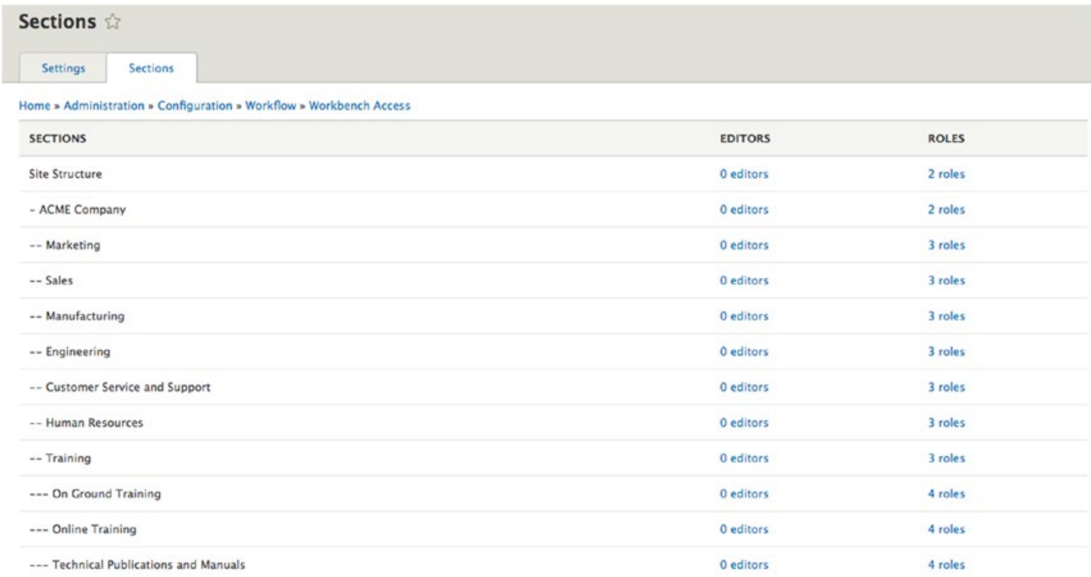


Figure 9-28. *All roles assigned to sections*

You may also assign individual users to site sections following a similar process. Click on the 0 editors link for the site section you want to assign a user to and select the user(s) to assign from the list of users shown on the page. After adding a user or users to a section, return to the Sections page and refresh to see the total number of users assigned to that section.

Demonstrating Access Restrictions

With all of the pieces in place, we are ready to demonstrate the functionality of the Workbench Access module. To facilitate the process, we'll create two new users, one with a role of Marketing and one with a role of Human Resources (see Figure 9-29).

Figure 9-29. *The new users with section specific roles*

After assigning roles, sections, and permissions, visit the Configuration ➤ Performance page and click the Clear All Caches button to rebuild any content access permissions that may be currently cached. After clearing the cache, log out and log back in using the `hr-user` account. Navigate to the My Workbench page by clicking on the Workbench menu item in the Administrator's toolbar. Click on the Create Content tab on the My Workbench page. Then select Article as the content type. Note that the only value for Site Section that you can select while logged in as the `hr-user` is Human Resources (see Figure 9-30).

Figure 9-30. *The list of Site Section options restricted by the current role*

Workbench Access is correctly restricting access to the Site Sections that the HR Editor account has access to. If you log in using the marketing user, you will see the same restrictions, only with access to the marketing section.

To demonstrate the hierarchical nature of Workbench Access, we'll create two additional user accounts, one with a role of training editor and one with enterprise editor. Logging in as the training editor and attempting to create a new article, we can see that we have the ability to specify any of the training sections for the article that we're creating (see Figure 9-31).

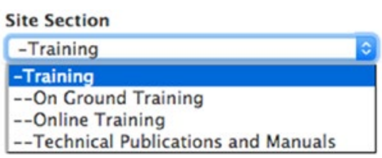


Figure 9-31. *Site section restrictions for a training editor*

If we log in as the enterprise user and attempt to create an article, we'll find that we can assign content to any site section as per the settings in Workbench Access.

While Workbench Access restricts which site section an editor can select, it is up to the site builder to utilize that field to restrict the display of content tagged with that term to the appropriate pages (e.g., using views and contextual filters).

Use Workbench Moderation

The last Workbench feature set that this chapter covers is the ability to moderate content. The Workbench Moderation module provides the ability to specify arbitrary moderation states beyond Drupal 8 core's published and unpublished states and provides the ability to restrict which users have the ability to move content through the various states. For example, you may have an additional step after a content item is edited where a site editor reviews it before it can be published on the site. A content creator can author the content and set the state as ready to review; however, the content creator cannot set the state to published. Only the site editor can do that after the content has been vetted.

The Workbench Moderation process focuses on node revisions instead of the node itself. This allows a node to remain published while updates are being made and pushed through the review process.

Configuring Workbench Moderation

After downloading and enabling the Workbench Moderation module (`drupal.org/project/workbench_moderation`), the next step is to define the moderation states that a node can travel through. Navigate to Configuration ➤ Workflow ➤ Workbench Moderation ➤ Moderation States to view the list of enabled states (see Figure 9-32).

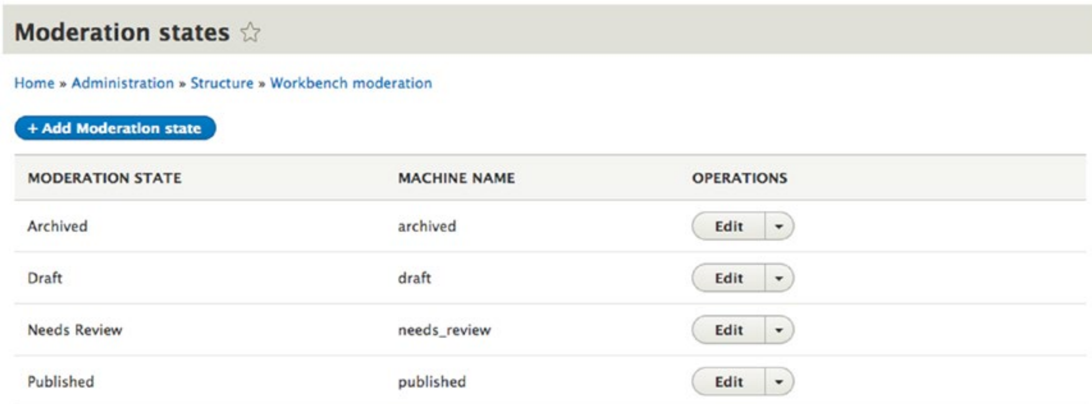


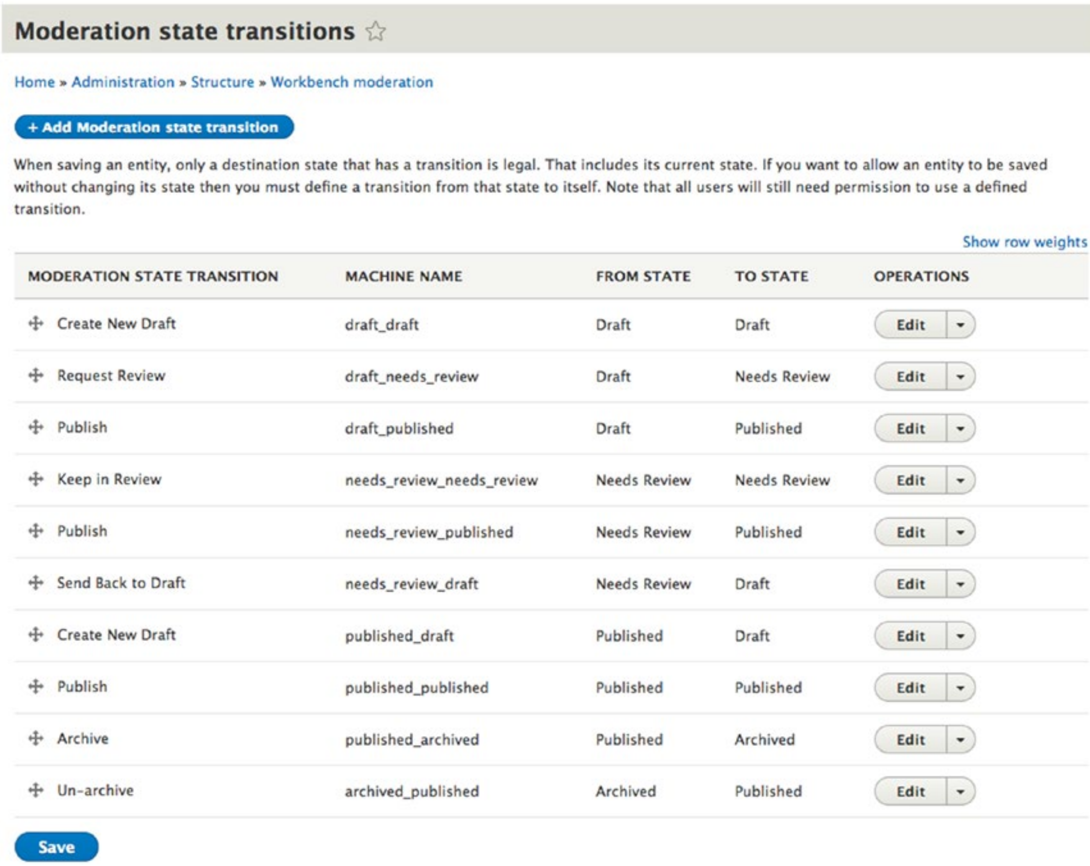
Figure 9-32. *The off-the-shelf Workbench moderation states*

For demonstration purposes, we add a new state of Rejected, which the site editor can use to send a content item back to the content creator for updates. To add a state, you click on the Add Moderation state button (see Figure 9-33).

Figure 9-33. *Adding a new workflow state*

After clicking save, you can now see the new Rejected state in the list of Moderation states.

The next step is to review and revise the moderation state transitions, meaning the flow of steps that a content item can transition through on its way to being published. The default off-the-shelf states are fine for my example site, with the one exception of needing to inject the Rejected state into the flow (see Figure 9-34).



MODERATION STATE TRANSITION	MACHINE NAME	FROM STATE	TO STATE	OPERATIONS
Create New Draft	draft_draft	Draft	Draft	Edit
Request Review	draft_needs_review	Draft	Needs Review	Edit
Publish	draft_published	Draft	Published	Edit
Keep in Review	needs_review_needs_review	Needs Review	Needs Review	Edit
Publish	needs_review_published	Needs Review	Published	Edit
Send Back to Draft	needs_review_draft	Needs Review	Draft	Edit
Create New Draft	published_draft	Published	Draft	Edit
Publish	published_published	Published	Published	Edit
Archive	published_archived	Published	Archived	Edit
Un-archive	archived_published	Archived	Published	Edit

Figure 9-34. *The off-the-shelf moderation state transitions*

To create the new transition, we click the Add Moderation State Transition button and enter Rejected in the Label field. We set the Transition from State to Needs Review, as that will be the current state when the site editor reviews the content, and the Transition To value will be set to Draft, as that will indicate to the content creator that there is work to do on the content item (see Figure 9-35).

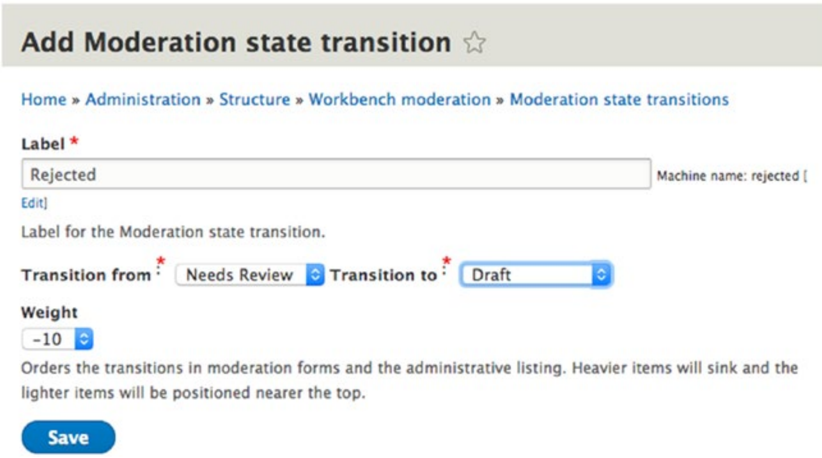


Figure 9-35. *Defining the moderation state transitions.*

The next step is to enable moderation on each of the content types on your site. Navigate to Structure ➤ Content types and edit the Article Content type. A new tab appears on the Edit Article Content Type page named Manage Moderation. Click on that tab to see an option to enable moderation states for this content type (see Figure 9-36).

Figure 9-36. *Enabling moderation states on the Article Content type*

After checking the box, you'll be presented with the list of available moderation states, the state that is associated with the content being published, and the default state that is set when the node is initially created (see Figure 9-37). We'll leave all the values checked and the default state set to Draft. Click the Save button to enable moderation on the Article Content type.

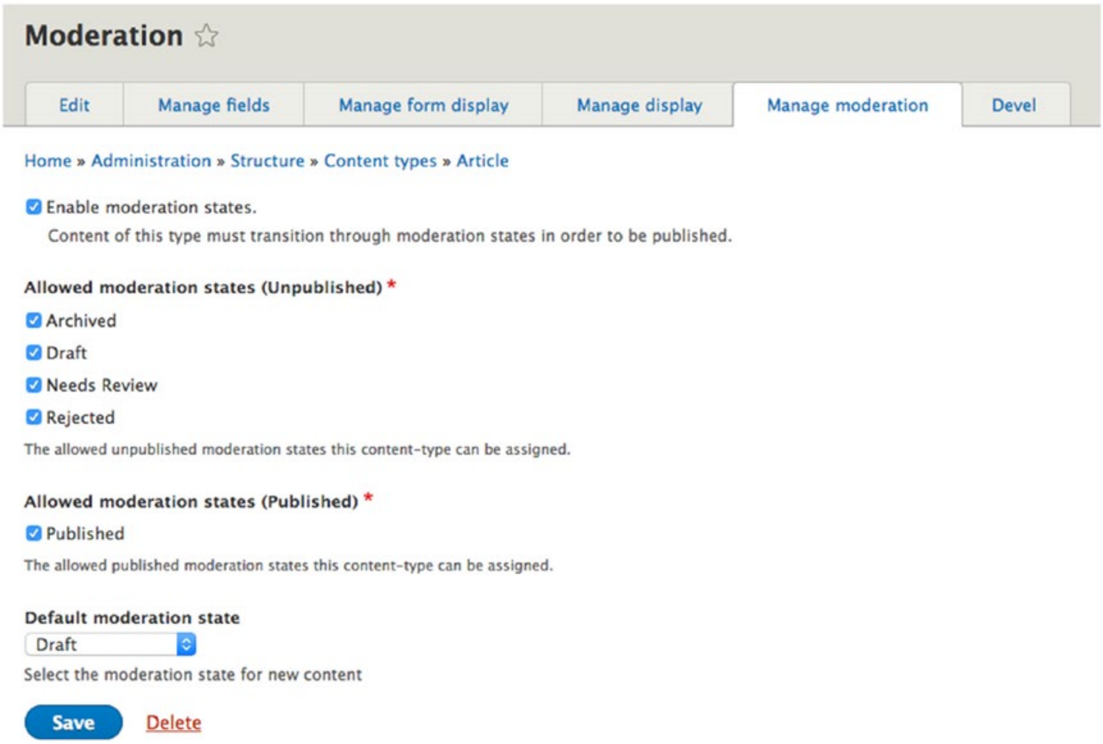


Figure 9-37. *Specifying the allowed moderation states*

Next we'll enable moderation on the other content types on the site before defining which user role has the ability to move content from one state to another.

Defining Workbench Moderation User Roles and Permissions

The final step in setting up workbench moderation is to define which user roles have the ability to move content through the various states. Navigate to People ➤ Permissions and, on the Permissions page, scroll to the Workbench moderation section. In that section, there are permissions for the following:

- Administering the configuration of the Workbench moderation module
- Viewing published and unpublished content
- Setting each moderation state that content can flow through

We'll use the existing user roles that we created earlier in this chapter, with the enterprise editor role being the only role that can reject and publish content. The details of the permissions and who can set them is listed in Table 9-2.

Table 9-2. *Setting the Workbench Moderation Permissions*

Permission	Enterprise Editor	All Other Editors
View any unpublished content	X	X
View moderation states	X	X
View latest version	X	X
Use the Archive transition (move content from published to archived)	X	
Use the Create New Draft transition (move content from Draft to Draft)	X	X
Use the Create New Draft transition (Move content from Draft to Published)	X	
Use the Keep in Review transition (Move content from Needs Review to Needs Review)	X	
Use the Publish transition (Move content from Needs Review to Published)	X	
Use the Publish transition (Move content from Published to Published)	X	
Use the Rejected transition (Move content from Needs review state to Draft)	X	
Use the Request Review transition (Move content from Draft to Needs review)	X	X
Use the Send Back to Draft transition (move content from Needs Review to Draft state)	X	X
Use the Send Back to Draft transition (Move content from Needs Review state to Draft state)	X	
Use the Un-archive transition (Move content from Archived state to Published state)	X	

To test the workflow, log out as the site administrator. Then log in as the marketing editor, who only has limited workflow permissions, specifically the ability to create a new draft of a content item and request review of that item once finished creating the content. We'll create a new article from the My Workbench page.

After entering the content, we're ready to save the article that we created and now only see two options available—we can save the article as a new draft or can save the article and request a review (see Figure 9-38). Let's save and request a review.

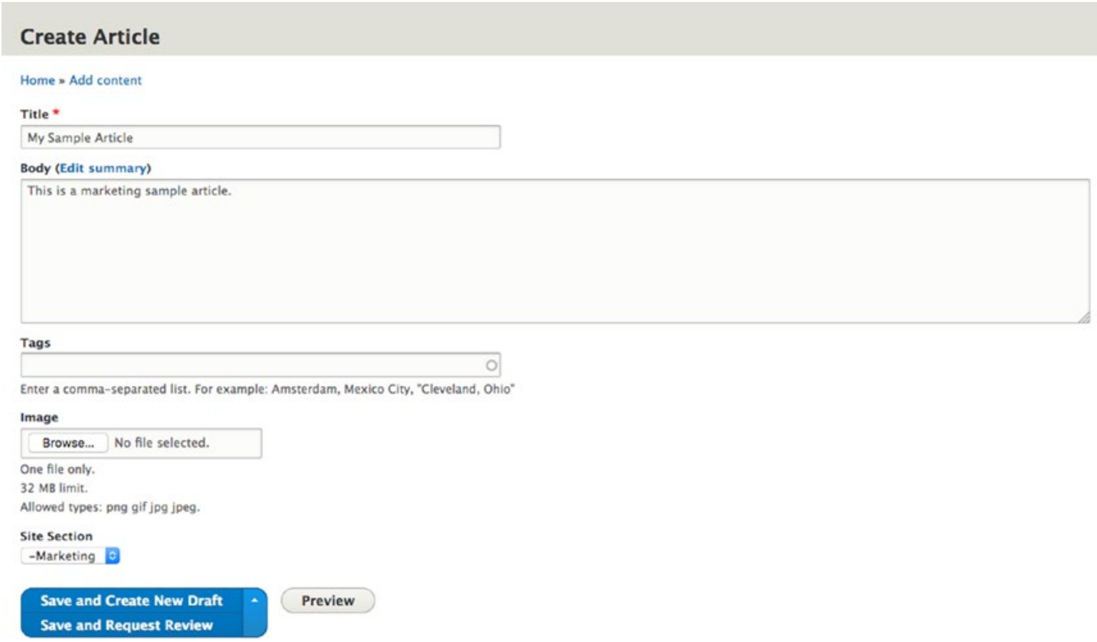


Figure 9-38. *Creating a new article with restricted publishing options*

After saving the article, the first thing that you'll notice is that the article has a pink background, which specifies that the article has not yet been published. (Note that the example is using the Bartik theme and other themes may use other colors or no color at all.)

Before reviewing the list of unpublished content that may need review, update the default workbench content view to include the moderation state field so that we can quickly see whether a node is in a state that needs editorial review. The view is Workbench: Recent Content and the field that we'll add is Content Revision: Moderation State, since it is revisions that are moderated and not the node itself.

Log out as the content editor and back into the site as a site administrator so you can update the view. After updating the view, navigate to the My Workbench page and click on the All Recent Content tab. Update the Published filter to "no" and click the Apply button, where you'll see the article that the marketing user created is in the Needs Review state (see Figure 9-39).

Figure 9-39. *The article that needs review is listed on the Recent Content page*

At this juncture, you can save the content item and set the state to any of the following:

- Draft, meaning it was rejected and needs work
- Keep in review so we can do more work on the article at a later time
- Published and viewable on the web site by site visitors

The content looks great, so we click the Save and Publish option (see Figure 9-40).

Figure 9-40. *Publishing the article after reviewing it*

While this example demonstrated the end-to-end use of the Workbench module and its various supporting modules, there is more that you can do with the capabilities presented by this suite of tools. Visit `drupal.org` and the various module homepages for details on other features and capabilities.

Summary

While the off-the-shelf Drupal administrative interface works well, there are opportunities for improving the experience for those who are responsible for the content that appears on your web site. Treat content editors well and enable them to do their jobs more effectively and your site will be better off in the end. As Drupal 8 matures there will likely be other modules that emerge to further improve the editorial and administrative backend of the CMS. Check `drupal.org` frequently for updates and new modules.

Providing a friendly user experience on the backend is great, but if your site doesn't perform well, nobody will be happy. The next chapter discusses the various areas to focus on when it comes to performance and scalability of your Drupal 8 web site.