Acquia

Drupal Site Building

Drupal Site Building

***By*:** Acquia Learning Services

Maintained By: Rod Martin, Drupal Trainer

Last Updated: January 2025

Published by Acquia © 2020
Acquia, Inc. 53 State Street, Boston, MA 02109

Acquia Training Feedback
We're always improving these materials and keeping them current. The modules are often updated between publishing dates. You may find some differences in the user interface, especially discrepancies in screenshots. We appreciate your feedback and will work as quickly as possible to incorporate any corrections. Please pass these on to your instructor.

About Acquia
Acquia is the open-source digital experience company that empowers the world's most ambitious brands to embrace innovation and create customer moments that matter. Acquia helps organizations create great web experiences using Drupal. Co-founded by Drupal's creator in 2007, its customers include Princess Cruises, LuLuLemon, City of Boston, ConAgra, Whole Foods, Johnson & Johnson, Panasonic, Warner Music, Stanford University, New York MTA, and Al Jazeera. As your enterprise guide, Acquia leverages Drupal's power while simplifying its deployment.

Acquia helps you...

- **BUILD** Drupal sites. Acquia Professional Services pairs customer projects with Drupal experts. Try building your own site for free here: https://insight.acquia.com/free/register

- **HOST** Drupal sites. Acquia Cloud provides flexible cloud hosting tuned for Drupal performance. Dev Cloud, specifically for professional developers, features drag and drop deployment. Learn more at https://www.acquia.com/products-services/acquia-cloud

- **MANAGE** Drupal sites. Acquia Remote Administration is a service that manages Drupal sites.

- **LEARN** to improve your Drupal skills. Acquia's training program is always expanding! What courses would you like to see us offer? Review our catalog on Acquia Academy by visiting https://customers.acquiaacademy.com or share your thoughts about our training offerings by emailing training@acquia.com.

- **PARTNER** with us. We have an extensive range of partners with expertise in a variety of sectors and technologies. View them at https://acquia.com/partners

- **FIND** out more at https://acquia.com/

Introduction

Course Description

Drupal Site Building provides a comprehensive introduction to the essential skills necessary when building websites with Drupal. It provides a solid introduction to Drupal and explains the most essential features and concepts of Drupal through lecture and hands-on activities. This training serves as an introduction to more advanced Drupal training in our Drupal Layout/Theming and Module Development training.

This course is meant to be interactive. You will build a functioning Drupal website for a London Tourism website by following the steps of this manual.

The procedures are introduced step-by-step and then reinforced with practice and challenges. Your instructor can help guide you through the process of establishing this new way of working. You'll be encouraged to reflect on your own experience to take the procedures you learn here back into your work and any new situations you encounter.

Target Audience

The target audience for this course are newcomers to Drupal. This class will be an overview targeted at those new to Drupal who are looking for a hands-on introduction as well as those who are ready to become skilled at Drupal site building or need to firm up their best practice knowledge for certification testing.

Prerequisites

To get the most out of this course, you should have:

- An understanding of how to use a web browser to access the internet.
- An understanding of website building using a Content Management System (helpful, but not necessary).

Training Environment

Please prepare for the course by following the instructions sent to you in the confirmation email. The environment is completely cloud-based and does not require you to install anything on your computer.

Course Learning Objectives

By the end of the course, you will be able to:

- Describe key features and functionality provided by Drupal core.
- Define Drupal terminology and describe concepts.
- Explain and summarize how Drupal sites are constructed and themed.
- Search for and use Modules to extend Drupal's functionality.
- Articulate the benefits of using Drupal to develop websites.
- Model content types.
- Create lists of content and other data.
- Assemble a website out of individual components.
- Get data into your database with fielded entities.
- Create lists of content from the database and manage display on your site.
- Set up custom URL paths.
- Understand all fundamental facets of building a Drupal website including how custom modules and themes are built.
- Assemble a website out of individual components.

Navigation instructions

Breadcrumbs: This text uses the "breadcrumbs" method when instructing you to navigate to a page. For example, if you are told to go to Content > + Add Content, you should search for the link called "Content", click on it, and once that page loads, search for "Add Content" and click on that.

Table of Contents

Introduction 3
Course Description 3
Target Audience 3
Prerequisites 3
Training Environment 3
Course Learning Objectives 4
Navigation instructions 4
Table of Contents 5
1.0 Getting Started 7
Learning Objectives 7
1.1 What is Drupal? 7
1.2 What Features Does Drupal Offer? 8
1.3 Database Driven Websites 10
1.4 Important Drupal Terminology 14
1.5 Drupal Administration Toolbar 17
Learning Module 1.0 Summary 25
2.0 Content Types and Content Modeling 26
Learning Objectives 26
2.1 Structured Content Types 26
2.2 Custom Content Types and Fields 30
2.3 Categorize Content with Taxonomy 34
2.4 Media in Drupal 39
2.5 Understanding Content Workflows 43
2.6 Understanding Contact Forms 45
Learning Module 2.0 Summary 46
3.0 User Roles and Permissions 47
3.1 Creating Roles and Assigning Permissions 47
3.2 Account Registration Settings 51
3.3 The Authenticated User Registration Form 52
3.4 Creating a Content Editor Account via Administrator 54
Learning Module 3.0 Summary 55
4.0 Building What Appears on the Page 57
Learning Objectives 57
4.1 Page Layouts with Blocks 57
4.2 Custom Blocks 59
4.3 Building Dynamic Lists of Content with Views 61

4.4 Views Displays....65
4.5 Custom URL Aliases....67
4.6 The Menu System....70
4.7 Configuration Synchronization....77
Learning Module 4.0 Summary....78
5.0 Building Complex Views....79
5.1 Contextual Filters....79
5.2 Relationships....83
5.3 Exposing Filter Options to Visitors....88
5.4 Creating a View That Returns a List of Content in JSON....89
Learning Module 5.0 Summary....91
6.0 Fielding Entities and Managing Displays....92
6.1 Review Adding Fields to a Content Type....93
6.2 Adding Fields to User Account Pages....93
6.3 Adding Fields to Taxonomy Term....93
6.4 Adding Fields to Block Types....95
6.5 Review Display Modes....97
6.6 Images and Display....99
6.7 Using Layout Builder....103
Learning Module 6.0 Summary....110
7.0 Drupal.org and Extending Drupal (Optional)....111
7.1 Drupal.org Community Resources....111
7.2 Extending Drupal With Modules....112
7.3 Using Themes....115
Learning Module 7.0 Summary....120
Appendix: Advanced Site Builder Concepts: Caching, Multilingual, Pathauto....121
Caching and Drupal....121
Multilingual Capabilities....124
Acknowledgements....127

1.0 Getting Started

Learning Objectives

At the end of this Learning Module, you will be able to:

- Describe Drupal in general terms.
- Log into a Drupal site.
- Obtain familiarity with Drupal's core administration navigation and terminology.
- Add content to a Drupal website.
- Add a custom block and make it appear on your site.
- Manage key visual aspects of your site, like the logo and site name.
- Visit your user account page.

1.1 What is Drupal?

Drupal is an open-source website development platform. It simplifies the online management of content and users, and it runs millions of sites—both small and large—around the world.

Drupal is a modular content management framework (CMF) that allows you to create a content management system (CMS) for your site, tailored to you. Drupal enables site administrators to:

- Organize and display content.
- Customize site appearance.
- Manage routine tasks such as user registrations.
- Manage users by individual accounts and roles, such as "members" or "content editors".
- Apply granular administrative and content permissions.

Drupal shines as a content management system (CMS). The platform accommodates unlimited content types, including image, video, audio, and text content. It dynamically retrieves, filters, and presents this content with powerful, yet simple-to-use tools. Drupal is search engine friendly—it can automatically create human readable URL aliases—and is increasingly accessible. Drupal supports a common set of schemas for structured data markup. The administrative interface is fully

responsive, that facilitates optimal viewing on any internet-connected device and makes Drupal sites easier to use and manage from anywhere.

The Drupal framework is open-source, and therefore the source code behind it is available to anyone interested in modifying or developing it further. Drupal has grown in leaps and bounds over the last few years, powering everything from small blogs to enterprise e-commerce platforms.

This growth is due in part to Drupal's unique strengths that include:

- **Community**–Drupal is supported by a vibrant and talented development community.
- **Custom content types and fields**–the ability to build and modify content types.
- **Taxonomy**–provides unlimited, multi-level categorization of content.
- **Views**–controls how lists and tables of content are displayed.
- **Caching**–Drupal's caching tools greatly reduce page load times.
- **User Roles**--Drupal has a rich and extensive permissions system.
- **SEO friendliness**–Drupal's clean URLs feature packs a mighty SEO punch.

Different Types of Websites Using Drupal

Businesses of all kinds use Drupal! You can review some great case studies and more at https://www.drupal.org/case-studies.

> *For Further Study:*
> *Understanding Drupal: https://www.drupal.org/documentation*

1.2 What Features Does Drupal Offer?

Drupal sets a new standard for ease of use and offers countless ways to tailor and deploy your content to the Web. Easily customize data structures, listings, and pages, and take advantage of capabilities for displaying data on mobile devices, building APIs, and adapting to multilingual needs.

Effortless Content Authoring

With the WYSIWYG editor, it has never been easier to edit content in Drupal. The new user interface features two columns so you can concentrate on what is important. Drupal features include:

- WYSIWYG configuration made easy with web security in mind.
- Built-in support for adding internally or externally hosted rich media in the website content.
- Workflows and draft saving is easy.
- Drag and drop page building using Layout Builder.
- Deploy content where you want it, the way you want it to look.

Social Publishing and Collaboration Tools

Drupal has built-in support for community with tools such as comments, forums, and customized user profiles. The addition of almost any other feature you can imagine - for instance, ratings, user groups, or moderation tools - is only an "install" away.

Role-Based Access Permissions

Custom roles and a plethora of permissions allow for fine-grained control over who can access what within the system. And existing modules can take this level of access control even further—down to the individual user level.

Flexible Module System

Modules are plug-ins that can modify and add features to a Drupal site. For almost any functional need, chances are good that either an existing module fits the need exactly or can be combined with other modules to fit the need, or that whatever existing code there is can get you a good chunk of the way there.

Continuous Innovation Model

Thanks to a new innovation model and versioning approach, Drupal keeps getting better. Semantic versioning moves the project to a major/minor patch versioning scheme that allows us to add significant, backwards-compatible improvements in minor releases like 11.05 and 11.1. With every minor release new optional alpha-stability modules ship with the core package allowing testing and feedback in hopes of eventual inclusion in core.

Each minor release has a development, beta, and release candidate phase. The beta phase begins about two months before a minor release date. When beta begins, the development branch for the next minor version opens.

> *For Further Study (These articles feature the major change that occurred in Drupal 8. The principles hold true with Drupal 11.*
>
> *The transformation of Drupal for continuous innovation: https://dri.es/the-transformation-of-drupal-8-for-continuous-innovation*
>
> *Making Drupal upgrades easy forever: https://dri.es/making-drupal-upgrades-easy-forever*
>
> *Experimental modules in Drupal: https://www.drupal.org/core/experimental*

Customizable Theming System

All output in Drupal is fully customizable, so you can bend the look and feel of your site to your will. HTML5 support is native to Drupal. HTML5 plays a key role in making websites fully functional and compatible with mobile screens.

1.3 Database Driven Websites

Database Driven Websites

- May store data in text files structured in a particular format, or in relational databases.
- May store data on a different physical server than the rest of the website, or may even store data on many servers at once.
- May use the SQL (Structured Query Language) to interact with its data.
- Allow you to backup the data of the site independently from the HTML and CSS files.

Most Popular Database Systems Used on Websites

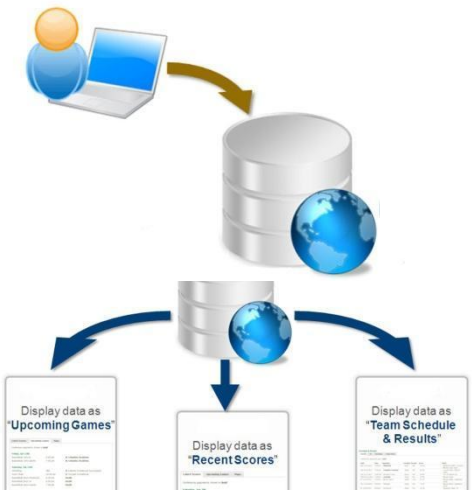
- MySQL
- MariaDB (fork of MySQL)
- PostgreSQL
- Oracle
- MS SQL Server

Getting data into the database.

- Structured data collection with templates
- Defining how data is stored
- Defining how data will be displayed

Querying data for display.

- Choosing type of data needed
- Applying filter, sort, and other rules to query
- Formatting display of results



The first process, getting data into the database, is accomplished in Drupal with an extremely flexible administrative interface that allows builders to create templates for fielded data collection. Each configuration entity allows for template data collection. Content entities are then simply data instances created from a template.

Solution architects can assess the available Configuration Entities to pick the best one for templating data collection.

The Entity System and Extensible Content

Working with a database driven website can be broken down into two fundamental processes: **Entities and Content.**

Entities are objects that are used for persistent **content** and configuration information. In Drupal nearly everything is an entity, including content. Entity types are used to store and display data. Entities themselves can be nodes (content), comments, taxonomy terms, user profiles, or something custom developed.

You can define new types of content (blogs, events, words of the day) as you need, and even add custom fields for the different content types. Contributed modules can extend this even further by providing new kinds of fields and different ways to manipulate them. Best of all, these fields can also be attached to anything in the system representing itself as an entity, such as users, comments, and taxonomy.

Entity	Template	Instance	Attributes	Path
Content	Content Type	Content	Nucleus of Drupal	node/1
Blocks	Block Type	Block	Flexible visibility	No Path
Comments	Comment Type	Comment	User generated data	
Taxonomy	Vocabulary	Term	Categorization, tags	term/1
User	Role	User	People, ability to edit own data	User/1
Contact Forms	Contact Type	Form	Data collection	
Media	Content Type	Content	Adding videos, images, documents to other entities	No Path

What We'll Build Today

The prominent London-based tourism publisher, Sociable Planet, called and they want to promote tourism to London. To begin, they would like a simple site listing various hotels around London, along with various landmarks of London. For tourism purposes, they want to be able to tell site visitors which hotels are near specific landmarks.

If we can finish this task in less than one day, then our agency will be showered with positive publicity!

Here are the functionalities Sociable Planet needs built today:

- Hotels
  - List page - alphabetical list of hotels
  - Hotel page - name, description, photo, website
  - Associated to a nearby Landmark
- Landmarks
  - Page displaying a grid on landmarks with photos
  - Landmark page - list of all associated hotels

We would like it to be as easy as possible for the Sociable Planet site maintainers and site editors to manage and create Hotels and Landmarks of London. It should be simple to associate hotels with nearby landmarks. We want registered users to be able to leave comments on each hotel. Ideally, we would like our agency to take this concept and expand it to other cities, such as Paris, Barcelona, or New York City!

We could, of course, apply our PHP ninja skills and build this website from scratch but recently our smart and savvy developer friend Fred has been thrilled and overjoyed about a technology he has come across called "Drupal."

Wow, our friend Fred provided us with a powerful insight. Drupal could transform our agency to be able to provide solutions far beyond what we are capable of right now! Let's figure out how to start developing with Drupal.

Setting Up a Drupal Learning Environment

Before you begin the following exercises, you will need to have a working Drupal website. You should have received a set of instructions when you enrolled in this course.

Composer

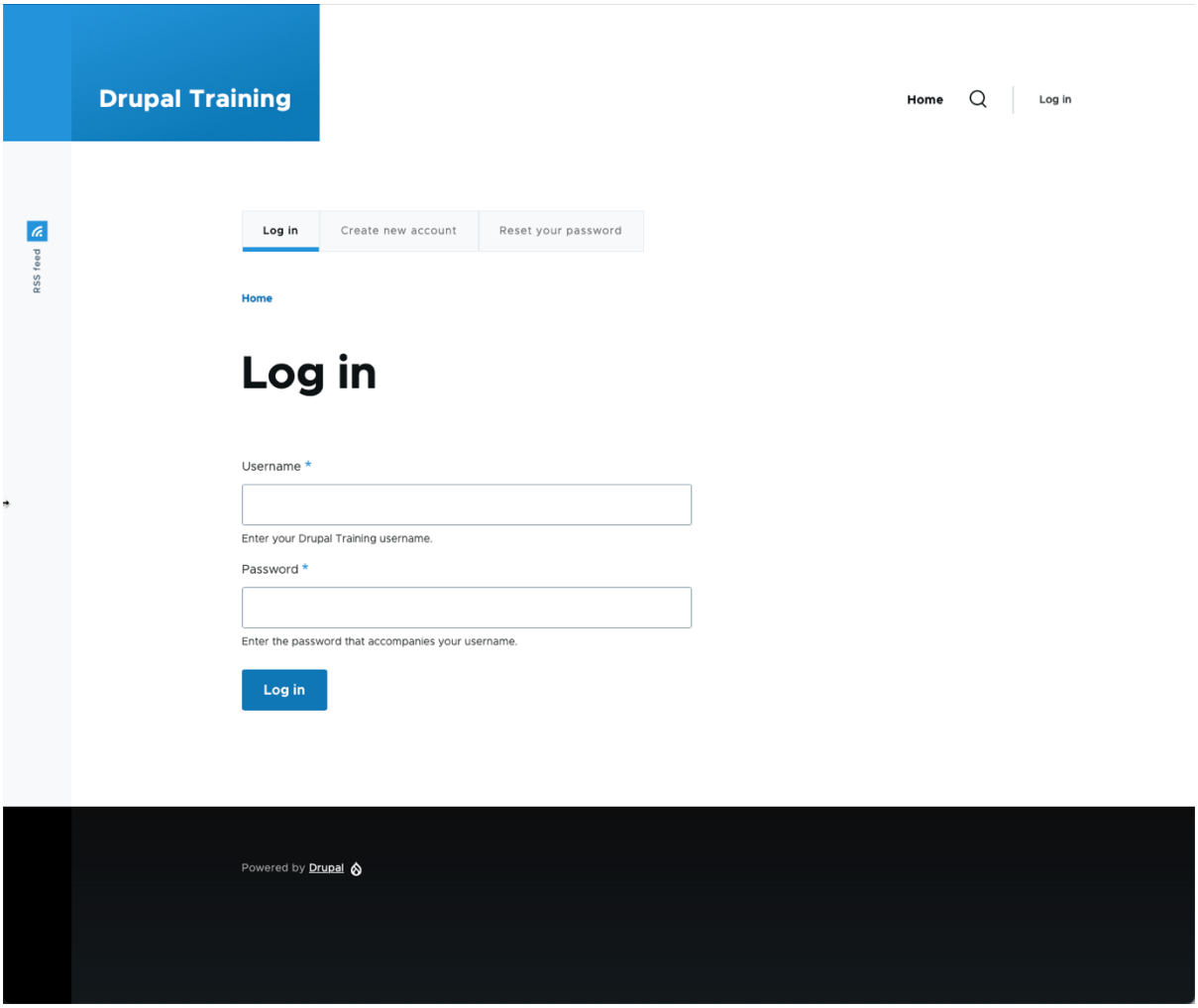
Drupal uses **Composer** as a dependency management tool. It is the only way to install additional modules and themes for Drupal (we will cover these in different sessions of the course.)

Composer is built into the provided environment for our course. If you are using your own local environment and don't have Composer, please go to https://getcomposer.org and follow the documentation specific to your Operating System and environment to install composer locally on your machine.

336346

> ***Note:*** *If at any time you need to sign in again as admin, or another user, look for the log in menu. If you can't find a log in tool, navigate to /user to visit the login page.*





1.4 Important Drupal Terminology

In this course, and when working with other Drupal users and Drupal texts, you'll hear certain terminology used that has a specific meaning in Drupal. To get you started, here are some definitions for common items you'll be using frequently in this course.

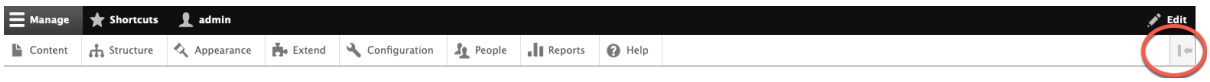
Superuser (user/1)

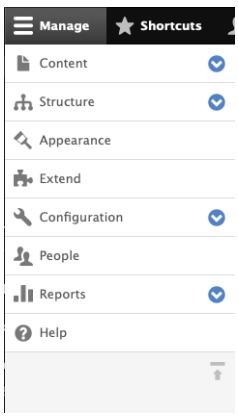
The Superuser (known historically as "user/1" from being the first user in the database: http://www.example.com/user/1) is the user with the highest level of permissions in Drupal. This term is roughly equivalent to the common website role of Administrator or Admin. The Superuser is not restricted by permissions and will see all administrative functions. Other users will only see what is allowed by the role they've been assigned.

Administration Toolbar

The Administration tool bar is the black menu bar that appears when administrative users are logged in. This selectively shows users' administrative functions divided into tabbed sections, depending on what permissions their account has been given. This also might be called the Administration Menu, Admin Bar, or Administrative Toolbar.

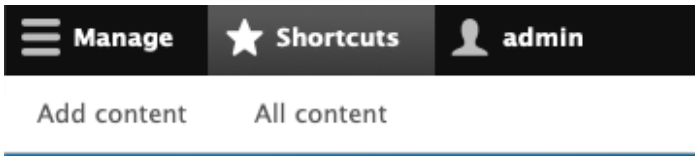
To display the Admin Menu horizontally or vertically, click click on Manage (top left) if the white toolbar is not visible, then click the adjustment icon.





Shortcut Bar

The Shortcut bar is customizable with whatever links you need and appears below the Admin Bar. Like the Admin Bar, a user will only see the links they have permissions to see.



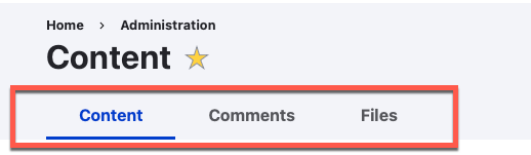
Add to Shortcuts Button

Each administrative page has a star (☆) appearing next to the name. Click the star to add the page to your Shortcuts bar for easy future access. The shortcuts bar is personal for each user. Each user can have their own collection of shortcuts.



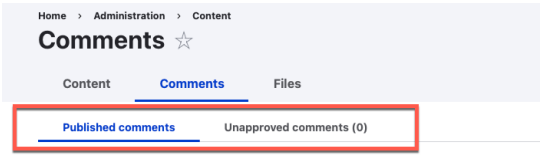
Section Tabs

Section tabs help handle second-level menu bar navigation. You will see tabs when you are on an administration page for a function that has sub-navigation. The number of tabs can change from none to several, and often if you don't see an option, you are looking for on the first page of the overlay, you can click on a Section tab and find it there.



Subsection Buttons

Sometimes when you are on a Section tab, there will be additional buttons, beneath the tab. These buttons represent third-level navigation for administration options.



Save

Drupal does not always automatically save your changes when you make them. Most pages have a Save button or a Save configuration button at the bottom. This also applies to pages where you can drag and drop elements–always look for the save button!

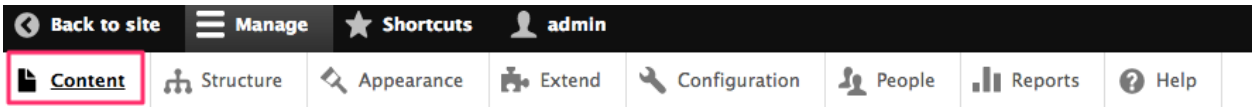
1.5 Drupal Administration Toolbar

We're about to take a whirlwind tour through all of the main sections of the Drupal administration area. The administration tour will first take you through each section on the Admin Toolbar, and briefly touch on some of the common actions that a Drupal site Superuser or Administrator will need to do.

While this particular lesson only touches on each of these sections briefly, we will revisit the most important sections again later in the course, and delve into greater detail at that time.

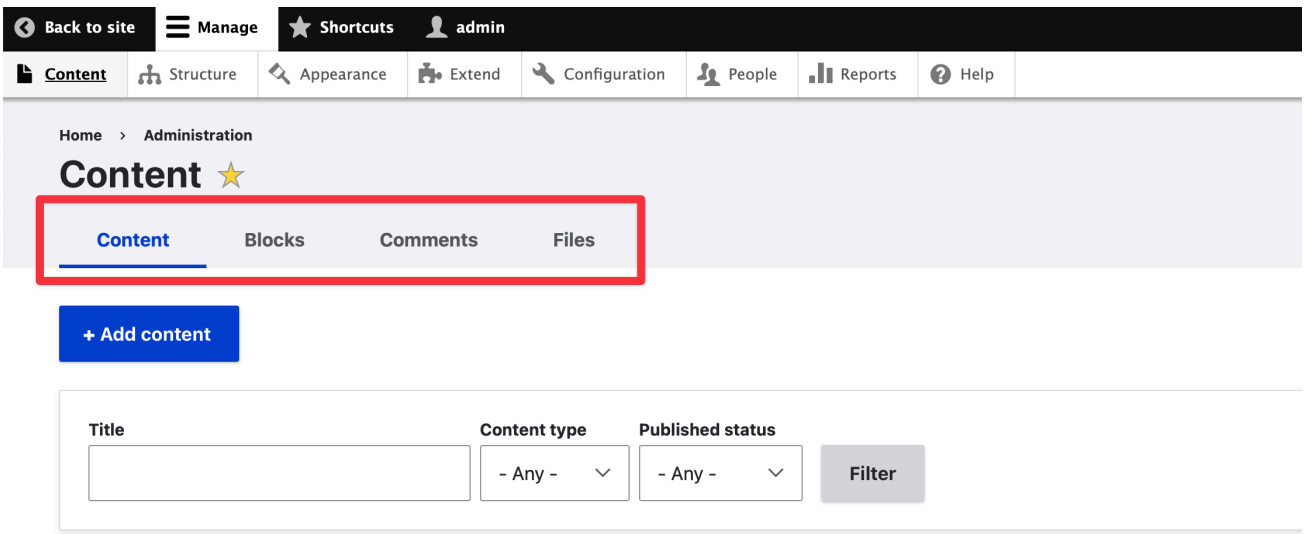
Content

The tour starts at the **Content** section that shows a list of added content.



The standard profile of Drupal comes with two Content Types, Articles and Basic Pages. Later on in this course we will show you how to create custom Content Types.

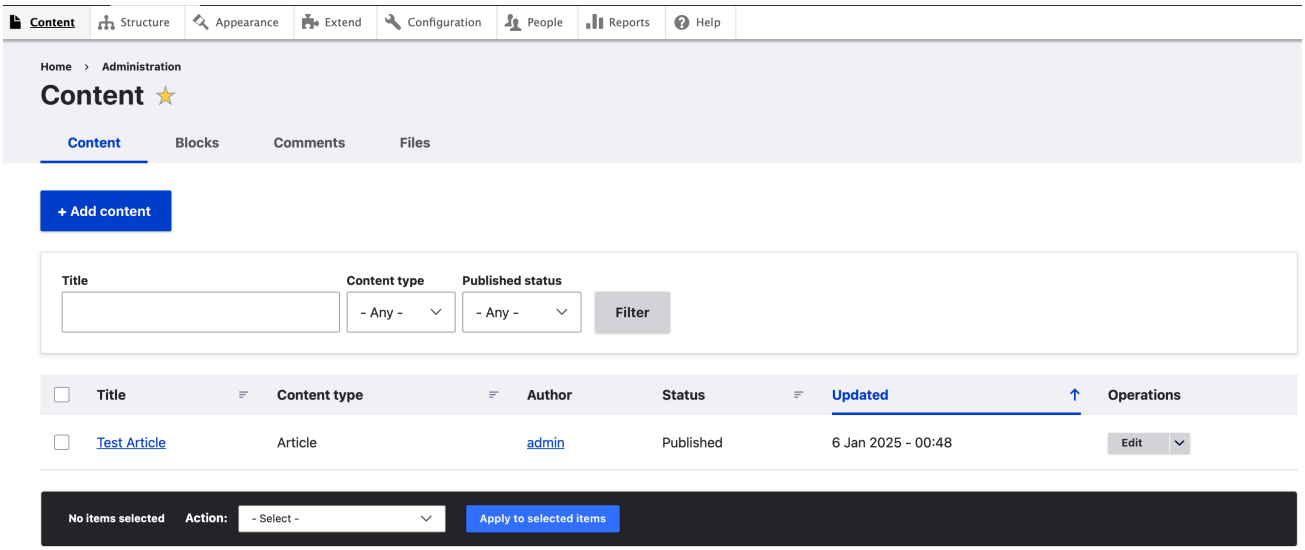
Keep an eye out for additional tabs appearing in administrative pages. In the screenshot below, also on the Content page, you can see how there is also a "Comments" subsection, and also a "Files" sub-section.



1.5.1 Exercise: Create an Article

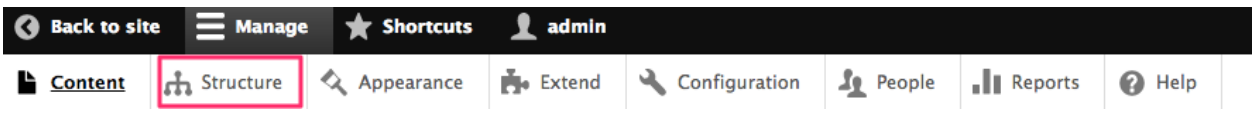
Steps:

1. Go to **Content > + Add Content > Article**.
2. Add text to the fields on the form.
    2.1. Enter **Test Article** for the *Title*.
    2.2. **Upload** an image for the Image field and enter Alternative text (this is required).
    2.3. Enter **Lorem Ipsum** for the *Body* field.
    2.4. Enter **hiking, biking** for the *Tags* field.
3. Click **Save** at the bottom.
4. Go to **Content** in the Administrative toolbar.
5. Observe: Verify your article is in the content list.



Structure

The Structure section includes the building blocks of your website, whether they are containers for content (Content Types), saved queries (Views), or navigation control (Blocks and Menus).



You won't be expected to know what each of those terms mean. You will become more familiar with them as we use each frequently when building a Drupal site.

Later on, you will add modules that will add new administration pages to Structure that you don't see currently. So don't be surprised if you add a module later, and discover these administration areas no longer look the same. Drupal is designed to be extended!

In the next exercise, you will go to the Structure menu to create a custom block.

1.5.2 Exercise: Add a Welcome Block

Steps:

1. Go to **Structure** > **Block layout**.
2. Scroll down to the **Sidebar** region.
3. Click **Place Block.**
4. Click **+ Add Content Block.**
5. Enter the following details:
    5.1.*Block description:* **Welcome Block.**
    5.2.*Body:* **Welcome to London City Guide! Discover London here.**
6. Click **Save** at the bottom.
7. At the bottom of the following screen, select **Sidebar** and click **Save.**
8. Click on the **Block layout** tab.
9. Scroll down to the *Sidebar first* section and click **Place block**.
10. Filter by typing **Welcome Block.**
11. Click **Place block** next to *Welcome Block*.
12. Click **Save block** at the bottom.
13. Click **< Back to Site** in the administrative toolbar. You'll see your new Welcome block appear on the right side of every page.

For Further Study

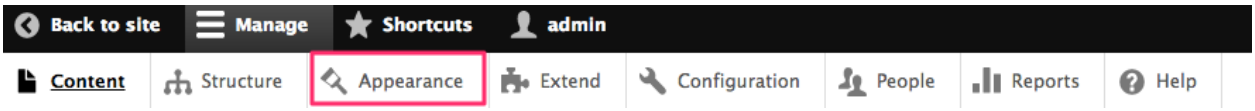
As of Drupal 10.1, Custom Blocks you create are listed under the ***Content*** *menu. You can edit blocks here but they must be "placed" or added to your site's layout under Structure->Block Layout.*

Blocks are not content *– they have no URL/Path. However for Content Editors who manage what appears on the page, having them under Content makes sense.*

Watch this video for more information: https://drupalhelps.com/blocks

Appearance

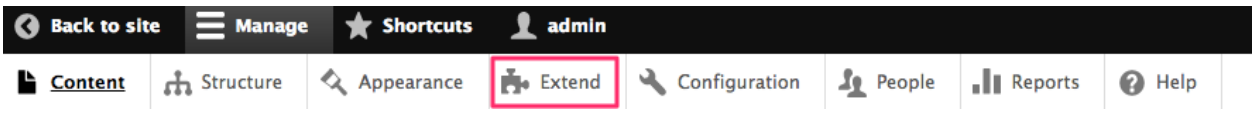
Appearance allows you to change the look of your Drupal site with themes. If you click on Appearance, you can see the default themes included with Drupal.



On the Appearance page, you can manage the settings of existing themes. We will work more with theme changes later in the course.

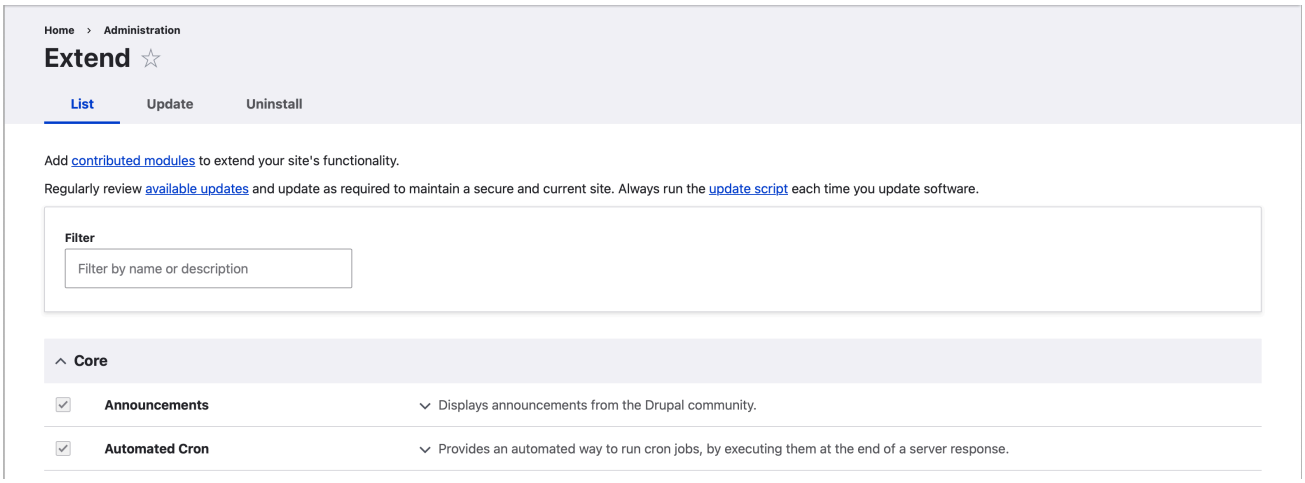
Extend

From the Extend section you can install, enable and uninstall modules.

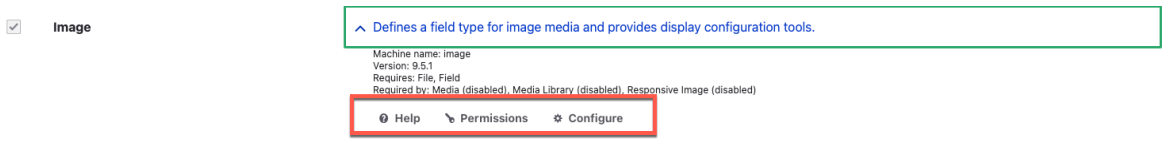


You can extend Drupal functionality by adding one or more Contributed modules from Drupal.org. Drupal has a rich contributed modules ecosystem, providing thousands of modules to extend Drupal functionality. Let's do a small exercise to add and enable a contributed module from Drupal.org.

We'll go into more depth on modules in a later lesson, but for now, here's a quick overview of what the Extend page looks like.



Click on the carats next to several of the modules listed to display the content beneath. Notice that for many modules there is a link to *Help*, to modify *Permissions*, or to *Configure* the module. Our example below is the Image Module. To find the Image Module, scroll down below the **Core** category (or click the arrow near the Core heading to collapse the list). Then go to the **Field Types** category and click the arrow to the right of the word Image.



You will also see checkboxes showing which modules are enabled and which aren't. A good practice is to uninstall any modules that aren't in use to free up resources on your webserver. Note that unlike Drupal 7, which allows you to disable a module by unchecking the box next to it, all higher versions of Drupal do not allow you to disable a module once it's enabled. If you are done with a module, you need to uninstall it completely. Not all modules that are part of the Drupal code base will be installed, but once you do install them, you will have to uninstall them if you decide not to use them.

In the next exercise, we will add a module called Admin toolbar. It is an extremely popular contributed module that improves the default administrative menu toolbar by replacing the static menu with dropdowns.

1.5.4 Exercise: Install the Admin Toolbar Module

Steps:

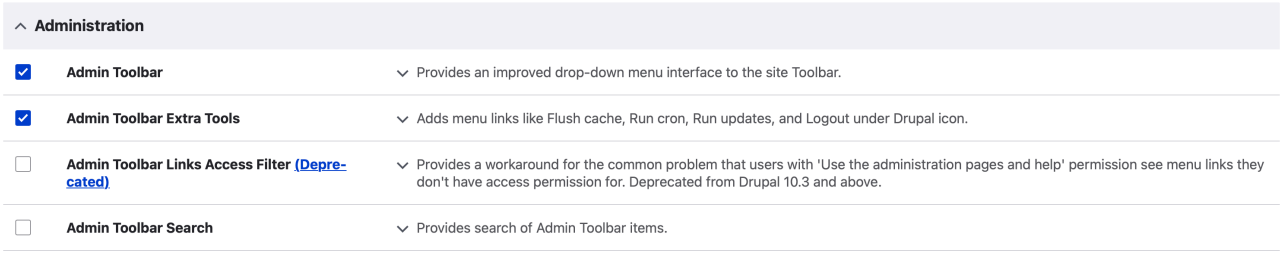
1. Return to the browser tab where you can see the code editor and terminal window. In the terminal window, on the command line, type the following command and hit Enter:

   ```
   composer require drupal/admin_toolbar
   ```

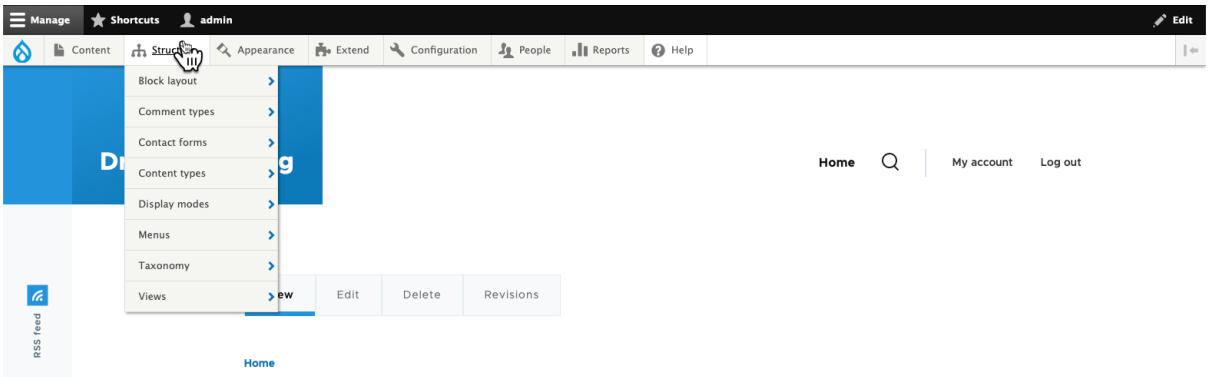
   This will check the dependencies and download the module and all required dependencies from Drupal.org and place the code in appropriate folders on your local Drupal environment

2. Return to your site.

3. Go to **Manage > Extend.**
4. Locate and select Admin Toolbar and Admin Toolbar Extra Tools under the Administration section.
5. Scroll to the bottom of the page and **click Install**.

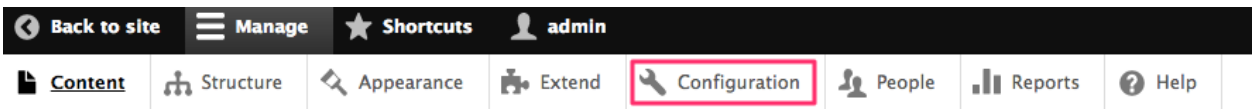


Now the Admin Toolbar module is installed and ready to use! You should see drop-down menus replacing the core static administrative toolbar display.



Configuration

Drupal's Configuration section gives you a variety of settings to update.



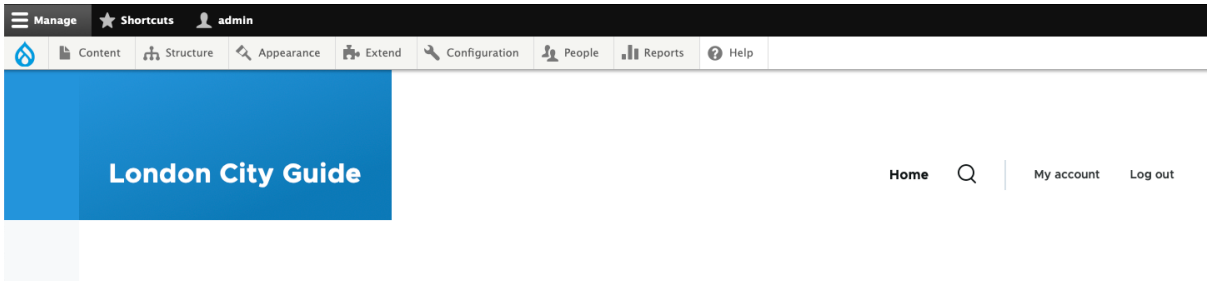
You can manage search, regional settings, overall account settings, and more. Click on Configuration and take a look at what's available.

One of the greatest advantages of using a CMS instead of maintaining a static HTML site is that much of the web site can be changed instantly by interacting with the site through a browser. Let's give it a try!

1.5.5 Exercise: Change Your Site Name and Slogan

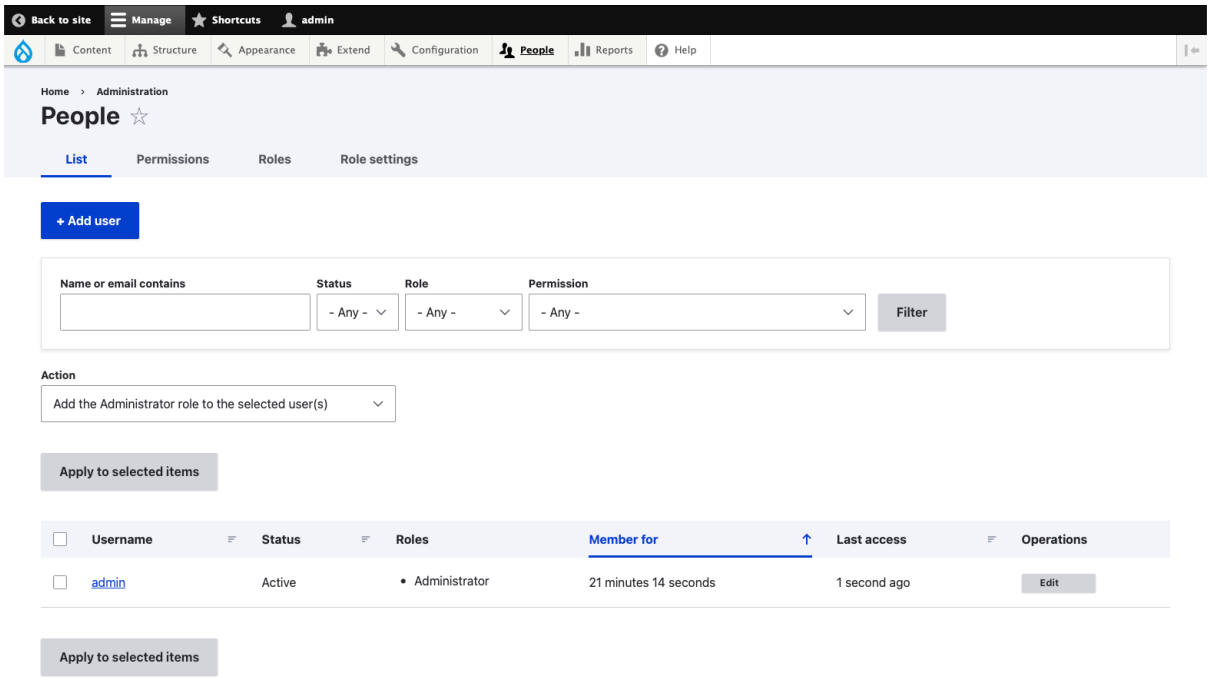
Steps:

1. Go to **Configuration** > **System** > **Basic Site Settings.**
2. Change the following fields (if needed).
    - *Site Name:* **London City Guide,** and
    - *Slogan:* **Discover London Your Way!**
3. Click **Save Configuration**.
4. Click **< Back to Site**.
5. When you are done, you should see that the title has changed. This particular theme does not display the slogan.



People

The People section is where you can manage permissions, user roles, and individual users.



Reports

Drupal provides a host of reports that you can run in order to keep tabs on various facets of your site. These reports are collected under the Reports tab on the Administration Toolbar.

Types of Reports in Drupal

Available Updates	For security of your site, it's important to keep modules and themes updated. This report displays what updates are available as well as the urgency of the update. In the Settings tab, you can select how frequently Drupal should check for new releases of currently installed modules and themes.
Recent Log Messages	Reports recent changes to your site and which user made them. This is a great place to start when debugging, and also gives you an indication of the overall condition of your site. You have the ability to filter and clear log messages to make the list more relevant.
Field List	Lists what fields you have already created to hold your data. This helps to avoid creating redundant fields when creating custom fielded entities.
Status Report	Reports if there are any notable security or configuration issues occurring with your site. This report might contain information about modules or themes that need to be updated or folders that need to be protected.
Top 'Access Denied' Errors	Logs if anyone has tried to access a protected page and failed.

Top 'Page Not Found' Errors	Logs if people are attempting to view a page on your site that no longer exists. This can happen if a page was linked by an outside website or search engine, and then the page is deleted, renamed, or the location of the page is moved. You can use the information on this page to see if you need to redirect users to a page's new location, or to see if anyone is fishing for restricted pages.
Top Search Phrases	Displays what topics are popular for users visiting your site. This information can be helpful in your content creation efforts and SEO (Search Engine Optimization) for your website.
Views Plugins	Provides an overview of all the plugins used in all views.

Learning Module 1.0 Summary

In this Learning Module you learned more about Drupal as a content management system. You got a thorough tour of Drupal's Administration Toolbar and tried a few exercises to see how you can quickly change the look and information on your site.

You should understand how to:

- Describe Drupal in general terms.
- Log into a Drupal site.
- Obtain familiarity with Drupal's core administration navigation and terminology.
- Add content to a Drupal website.
- Add a custom block and make it appear on your site.
- Manage key visual aspects of your site, like the logo and site name.
- Visit your user account page.

2.0 Content Types and Content Modeling

This learning module focuses on one of the key concepts of Drupal site building, getting content into the site. Content in Drupal is generated using Content Types. Content Types are templates used to enter data into our Drupal site.

Learning Objectives

At the end of this Learning Module, you will be able to:

- Understand the value of structured content collection.
- Add content using Drupal's built-in content types.
- Create a custom Content Type.
- Create a basic Taxonomy vocabulary to categorize content.
- Manage fields and field settings.

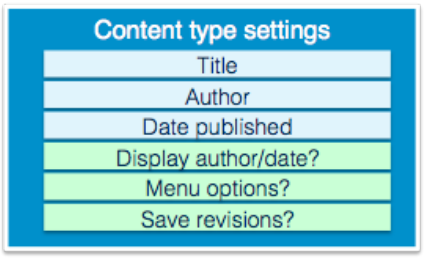
2.1 Structured Content Types

One of Drupal's key features is the ability to collect structured data to assist developers in building a website that meets their client's requirements. When we use the term structured data, we are referring to the process of organizing data in a way that makes it easy to filter, search, and sort.

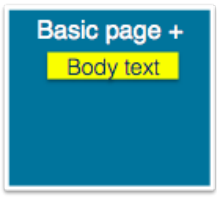
Benefits of structured over unstructured data collection

- Model the data to real-world needs
- Validate input (mandatory fields, formatting, limiting choices)
- Better control of display
- Ease of data migration

There are two content types included with core Drupal: Articles and Basic pages. As you will see in this lesson's exercises, the main differences between the two content types is their different default settings and the structured data they collect.





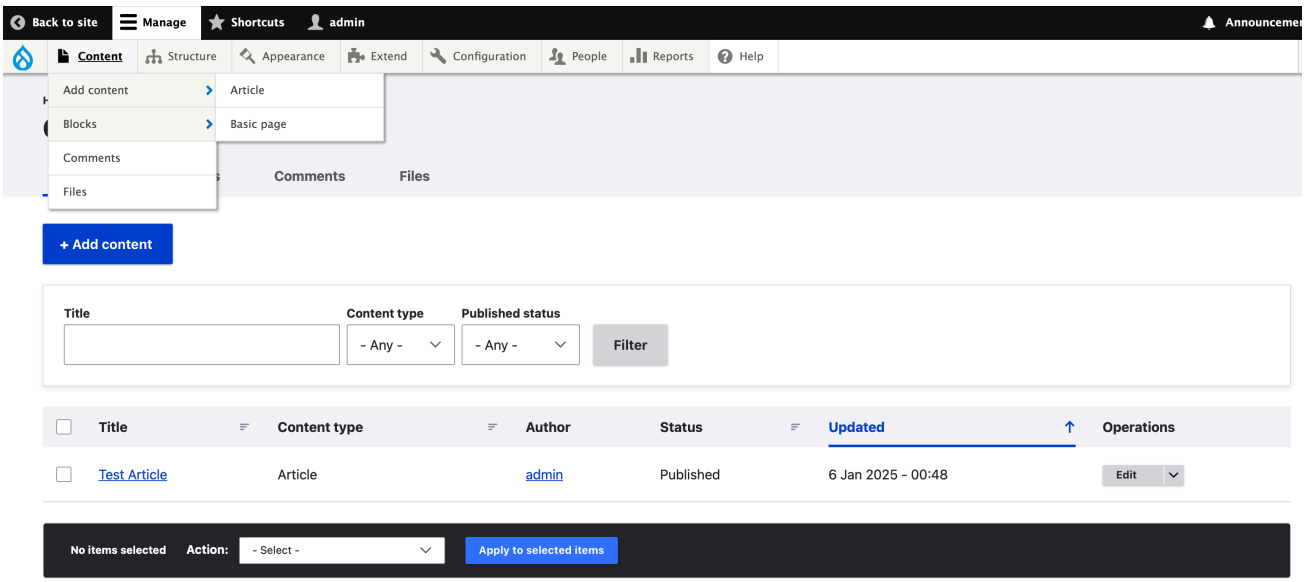


In the next set of exercises, create two different types of content so you can compare the editing forms, the content type defaults, and how images behave.

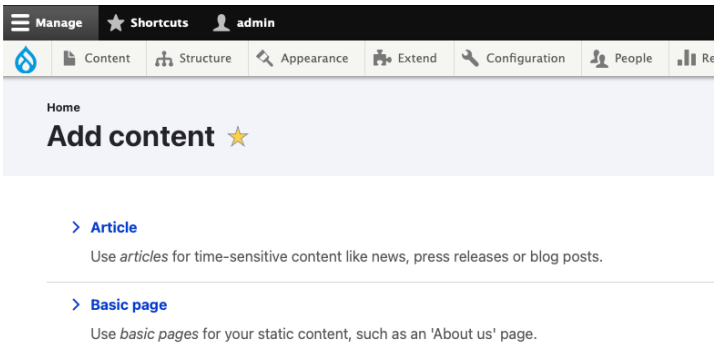
2.1.1 Exercise: Add a Node of the Basic Page Content Type

Steps:

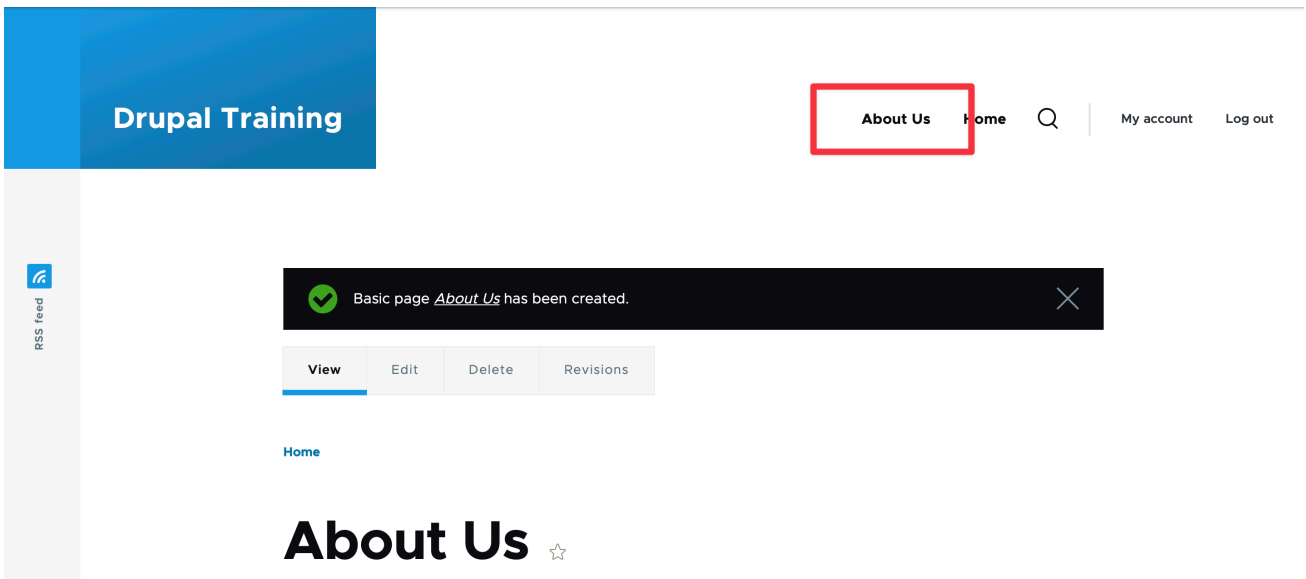
1. Go to **Manage > Content,** and click **+ Add content**.



2. Observe: There are two default content types currently available.
    - Article
    - Basic Page



3. Click on **Basic page** to load the content creation form.
4. Populate the Basic page.
    - *Title*: **About Us**.
    - *Body*: Enter **Lorem Ipsum** text and format using the WYSIWYG editor options.
    - *Menu settings*: **MENU SETTINGS** on the right and check **Provide a menu link**. You can optionally add a description.
5. Click **Save**.
6. Review your new page. Notice this page is linked to the main navigation menu.

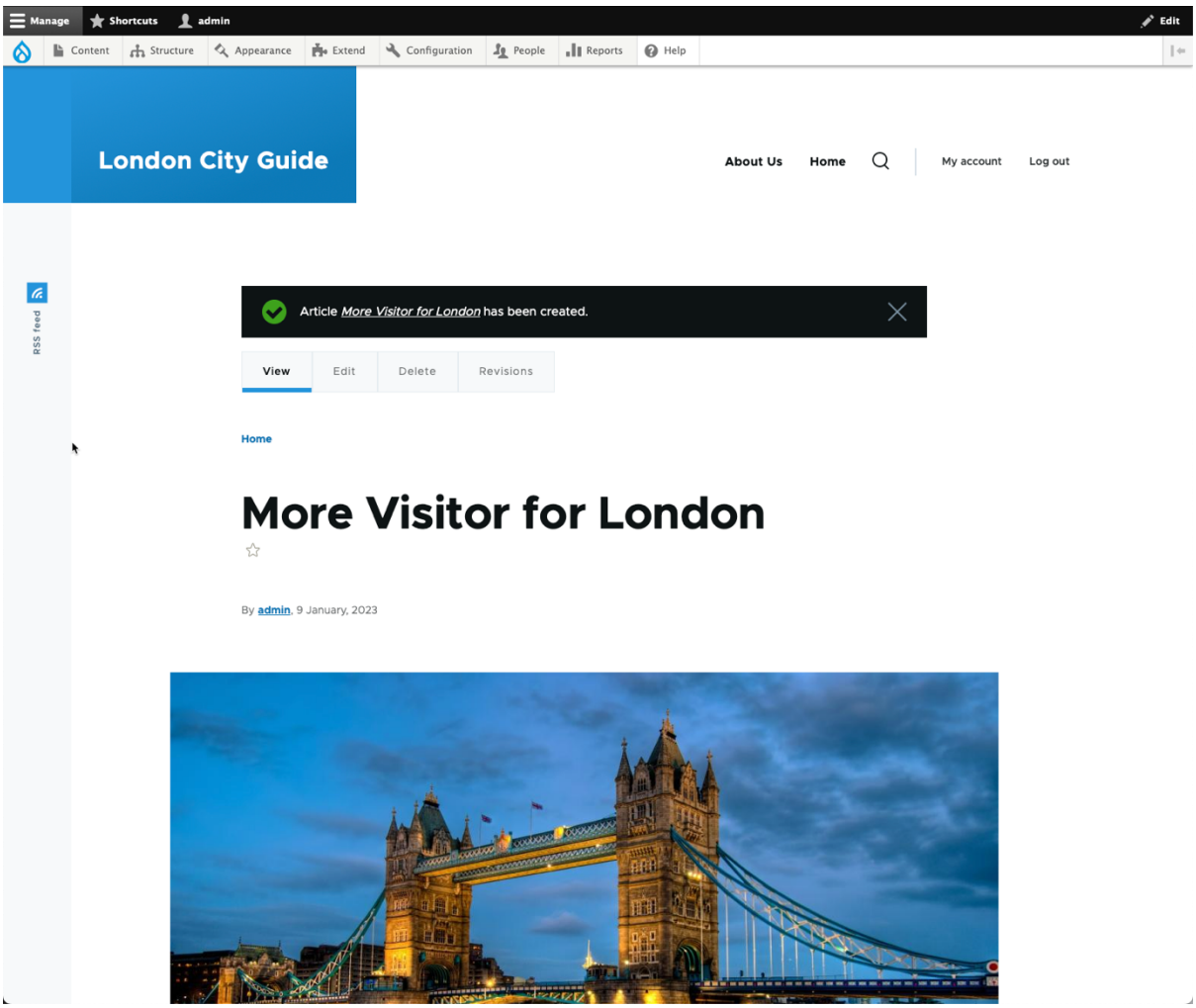


Note: Menu links are listed alphabetically by default. You can rearrange the menu links to suit your needs. We will learn more about this in the Menus section.

2.1.2 Exercise: Add a Node of the Article Content Type

Steps:

1. Go to **Manage** > **Content** and click **+ Add content** button.
2. Click on **Article** to load the content creation form.
3. Populate the Article.
    - *Title:* **More visitors for London**
    - *Image:* **Browse to upload an image to the article.**
    - *Alternative text:* **Enter a description.**
    - *Body:* Add a paragraph of placeholder body text using a lorem ipsum generator and the WYSIWYG editor.
    - Leave the tags empty.
4. Click **Save**.
5. Observe your Article.



2.2 Custom Content Types and Fields

One of Drupal's key features is the ability to collect structured data to assist developers in building a website that meets their client's requirements. When we use the term structured data, we are referring to the process of organizing data in a way that makes it easy to filter, search, and sort.

A Content type defines a set of fields that are required to store and display a particular type of information. For example, a Job Posting Content type might have fields like Job title, Job description, terms of employment, and salary, whereas a News Content type might have fields like Headline, Sub-headline, Location, Date, Reporter byline, and Images.

As we have seen in section 2.3, Drupal is made up of Entities that hold data. A Content type is a type of Entity that holds the data which is displayed on the main page area of the website.

In this way, content types can be thought of as templates for the bits of content you need to display on your site. The ability to create custom Content Types allows site builders to model the data and content that they need to manage with Drupal.

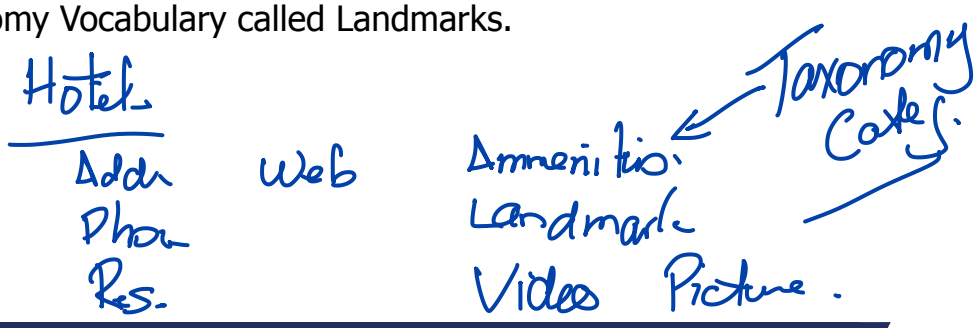
For Further Study
Learn more about content entities :
https://www.drupal.org/docs/user_guide/en/planning-data-types.html

Learn more about modular content:
https://www.drupal.org/docs/user_guide/en/planning-modular.html

Learn more about planning your site's content structure:
https://www.drupal.org/docs/user_guide/en/planning-structure.html

Our site needs to show information about various hotels and group those hotels according to landmarks they are near.

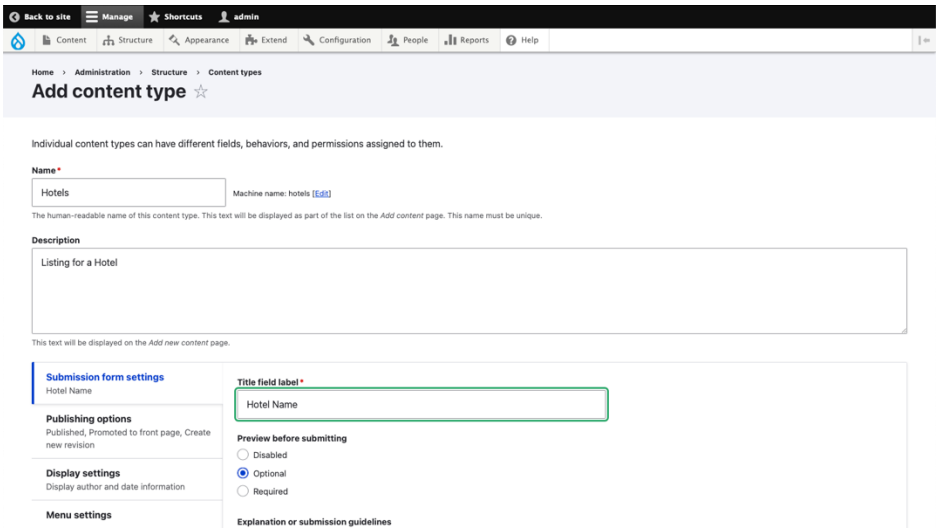
In the next set of exercises, you will create a new Content Type called Hotel and a Taxonomy Vocabulary called Landmarks.



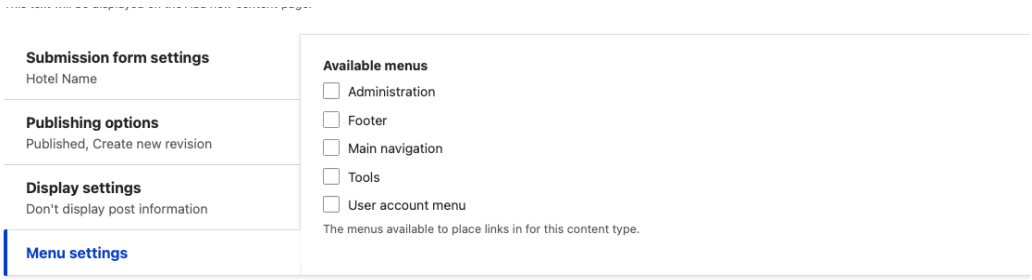
2.2.1 Exercise: Add a New Content Type

Steps:

1. Go to **Manage > Structure > Content types**. Click **+Add content type** button.
2. Configure your new content type.
    1. *Name:* **Hotel**
    2. *Description:* **Listing for a hotel**
    3. For *Submission form settings*, change *Title field label* to **Hotel name**



4. Click on *Publishing options,* deselect **Promoted to front page** and make sure that **Create new revision** is selected.
5. Click on *Display settings,* deselect **Display author and date information**
6. Click on *Menu settings,* **deselect all menu options**



3. Click **Save and manage fields**.

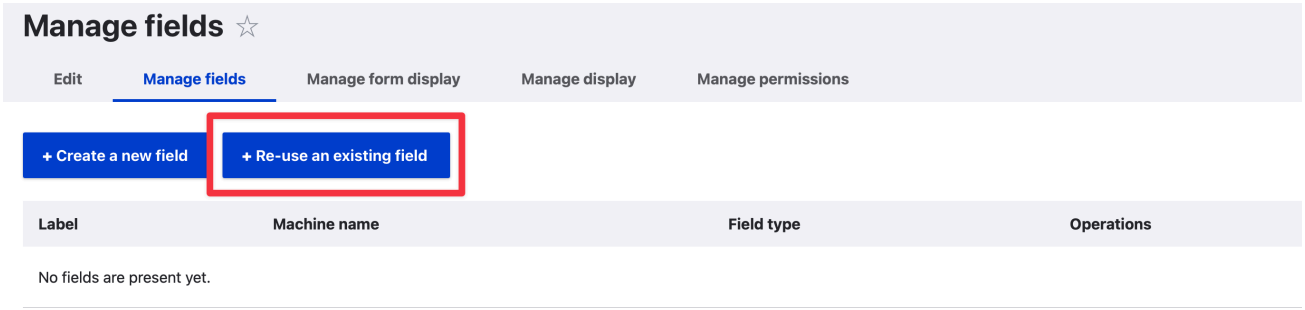
Specific fields allow custom data to be collected by Drupal entities. In this case our entity is a Content Type. In Drupal, almost everything is an entity. Fields take care of storing, loading, editing, and rendering custom data.

In the exercise that follows, you will learn how to Manage Fields and Field Settings in Drupal.

2.2.2 Exercise: Re-Use an Existing Field

Steps:

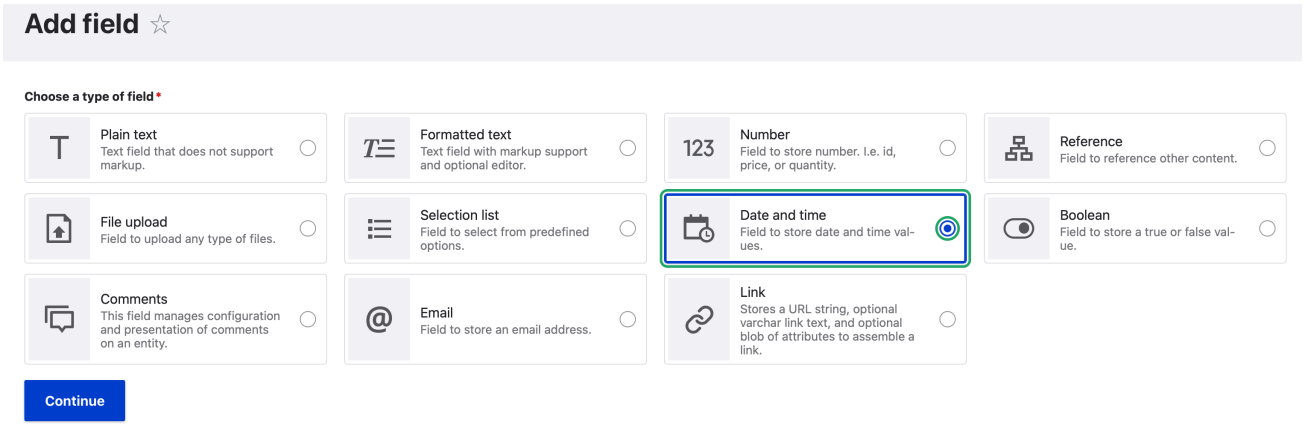
1. Go to the *Manage Fields* page for your new content type, if you are not already there.
2. In previous versions of Drupal, a body (text area) field was created by default. As of Drupal 11.1 this is no longer true.
3. Click on the **+ Re-use an existing field** button.
4. Select the field_body field at the bottom of the list.
5. In the Label field, replace Body with Hotel Description.
6. Click **Save Settings**.



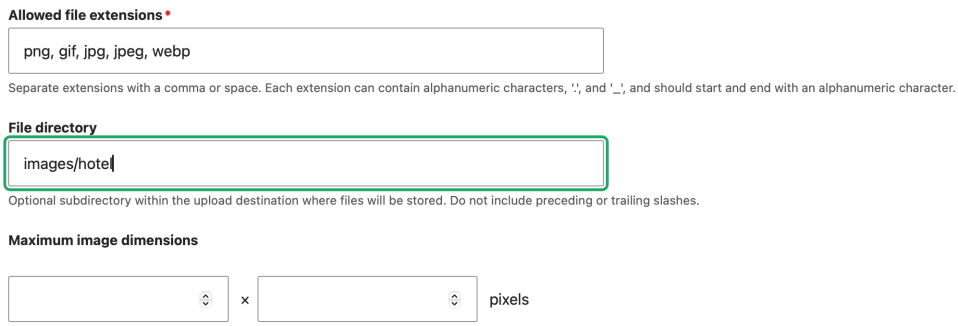
2.2.3 Exercise: Add a New Field and another Existing Field

Steps:

1. Go to the *Manage fields* page for your Hotel content type, if you are not already there.
2. Click on the **+ Create new field** button.



3. Select **Date and time** and click **Continue**.
4. Enter *Label:* **Date Rooms are Available** and select **Date.**
5. Click **Save and continue**.
6. Change *Date type* to **Date only** on the global field settings page.
7. In the *Help text* field enter the following text*:* **Click inside the field to expose the pop-up calendar for easier date selection.**
8. Click **Save settings**.
9. Observe: Your new date field included in the list of fields.
10. Next, we'll add an image field by re-using the existing Article image field.
11. Click on the **+ Re-use an existing field** drop-down.
12. Select **Image: field_image**
13. Change *Label:* **Hotel Image**
14. Click **Save and continue**.
15. Locate the *File directory* option: enter **images/hotel**



16. Click **Save settings**.

2.2.4 Exercise: Test Your New Hotel Content Type

Steps:

1. Go to Manage > Content, and click + Add content
2. Click on Hotel from the content type list.
3. Populate the *Hotel* details.
4. *Title:* Drupal Hotel
5. *Hotel Image:* **Upload** a picture from the course downloads in the Hotels folder
6. *Hotel Description:* **Add** some lorem ipsum
7. *Date Rooms are Available:* **Choose** today's date
8. Click **Save**

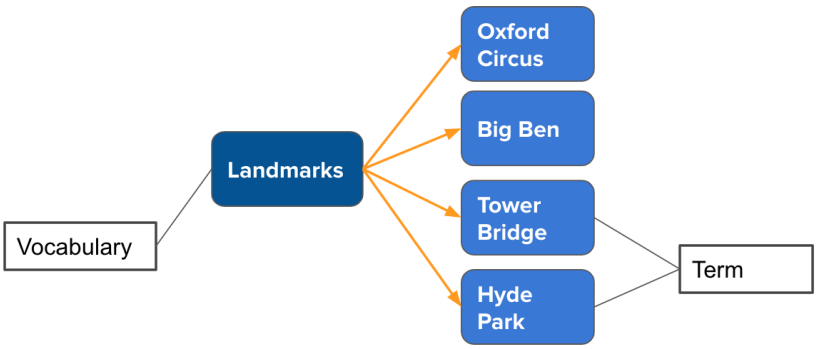
2.3 Categorize Content with Taxonomy

In Drupal, Taxonomy gives your site(s) the ability to use organizational keywords. In other systems this concept may be known as categories or tags. In Drupal, taxonomy allows you to connect, relate, and classify your website's content.

Benefits of how Drupal uses taxonomy:

- Automatically creates a list of content tagged with the same term
- Relates content together
- Creates RSS feed per taxonomy term
- Taxonomy terms can be fielded

Drupal Taxonomy consists of Vocabularies and Vocabularies consist of Terms. For example, in the diagram below, Landmarks is the name of the Vocabulary, while Oxford Circus, Big Ben, Tower Bridge, and Hyde Park are taxonomy terms in the Landmarks vocabulary. Each term can have a child term. So the depth of Taxonomy Vocabularies is "unlimited."



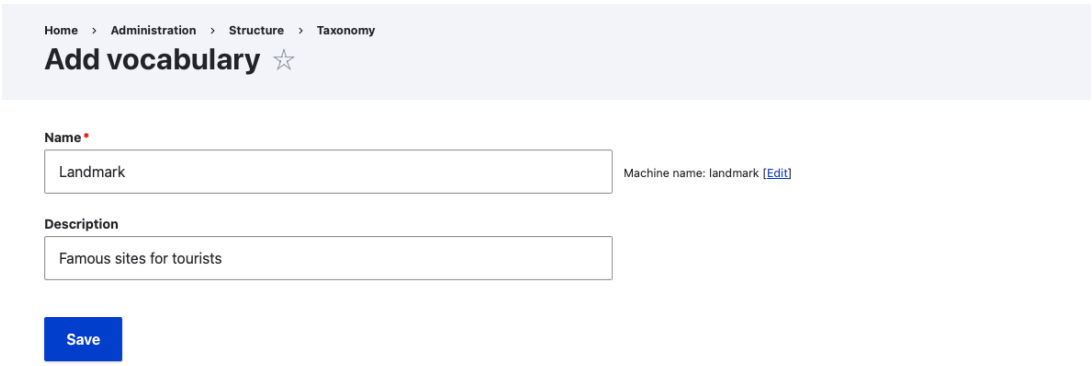
In the next set of exercises, we will set up a Vocabulary that contains landmark terms. After we create the Landmarks Vocabulary, we will add each landmark as a Taxonomy Term.

In the next lesson we create the Landmark Vocabulary.

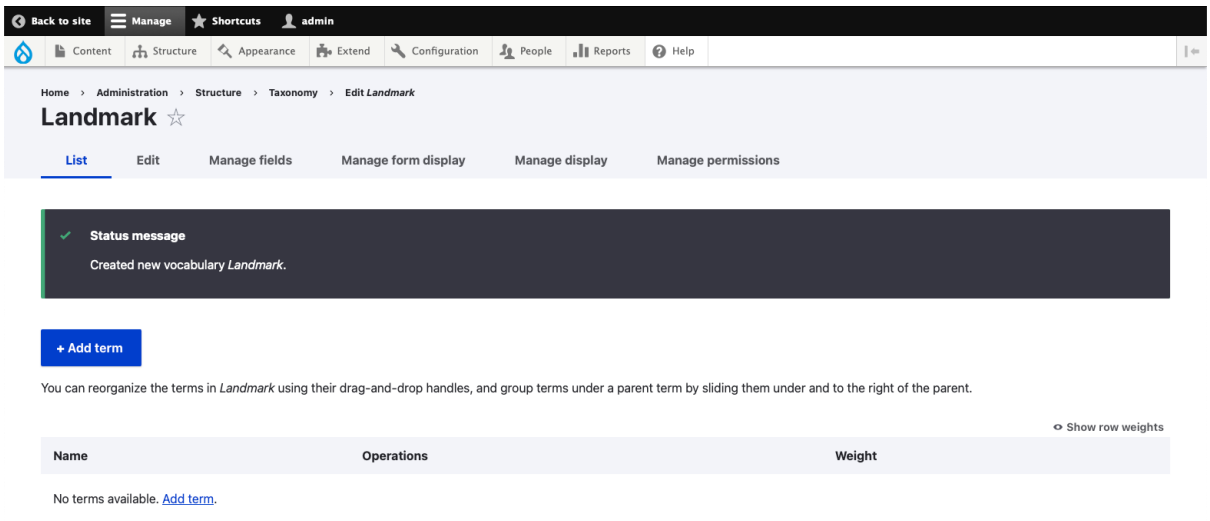
2.3.1 Exercise: Create a Taxonomy Vocabulary

Steps:

1. Go to **Manage > Structure > Taxonomy**. Here you can see a list of the vocabularies available in the system. Click **+ Add vocabulary**.
2. Configure your new vocabulary.
   2.1.*Name:* **Landmark**
   2.2.*Description:* **Famous sites for tourists**

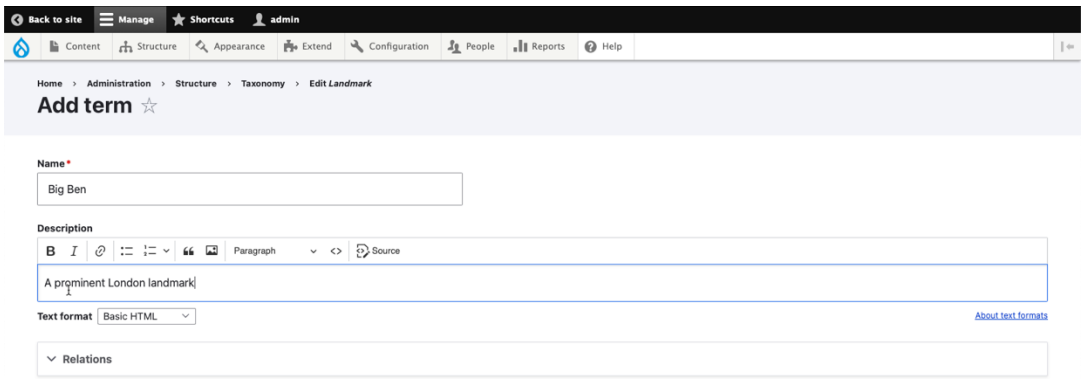


3. Click **Save.**
4. Observe: This brings you to the Landmark vocabulary configuration page. Notice the similarities with the content type configuration options.

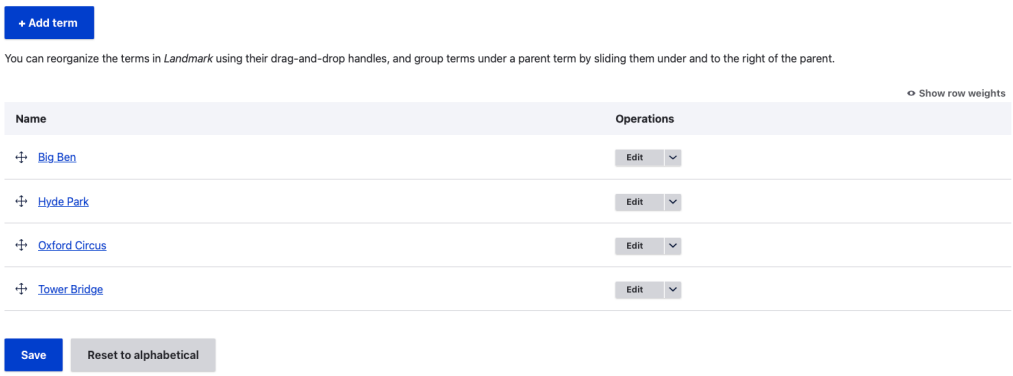


5. Click **+Add term** on the *Landmark* vocabulary configuration page.
6. Enter for *Name*: **Big Ben**.
7. Enter for *Description*: **A prominent London landmark**

8. Click **Save.**
   Pro tip: Hitting enter after each Term name will save and reload the page automatically.
9. Repeat steps 5 through 8 for the following terms, adding your own description.
   - **Tower Bridge, Oxford Circus,** and **Hyde Park.** After entering Hyde Park, click **Save and go to List.**



On the list page, you'll note the four terms you added**.**



   - By default, the listing of terms is alphabetical. On this listing page you can reorder terms by using the four-headed arrow to drag and drop the terms in the order you prefer.

Now that the Landmark Taxonomy Vocabulary has Terms, we will connect the new vocabulary to the Hotel content type as a field.

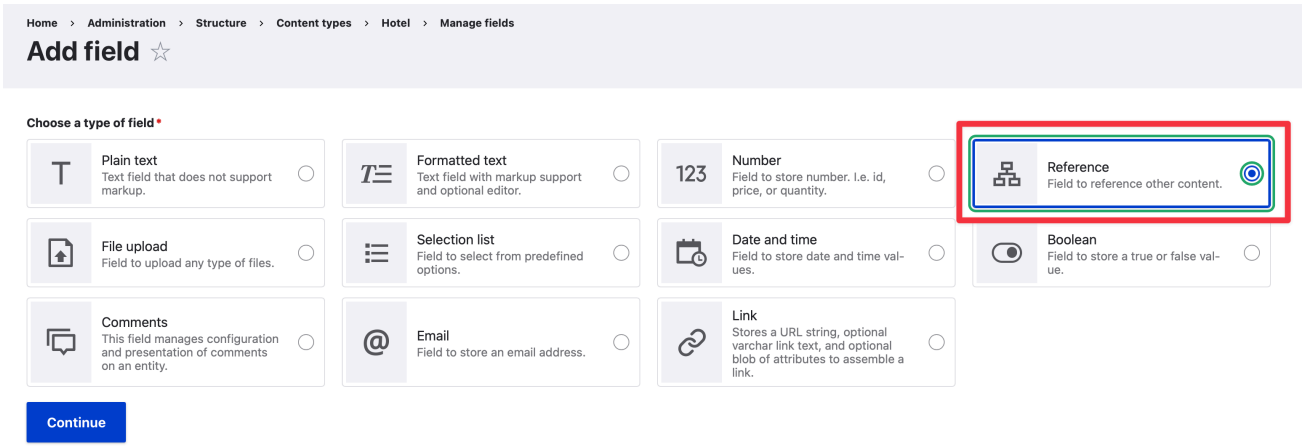
This will allow us to associate a nearby landmark when adding Hotel content.

In the next exercise, we will add a Taxonomy field to our Hotel content type.

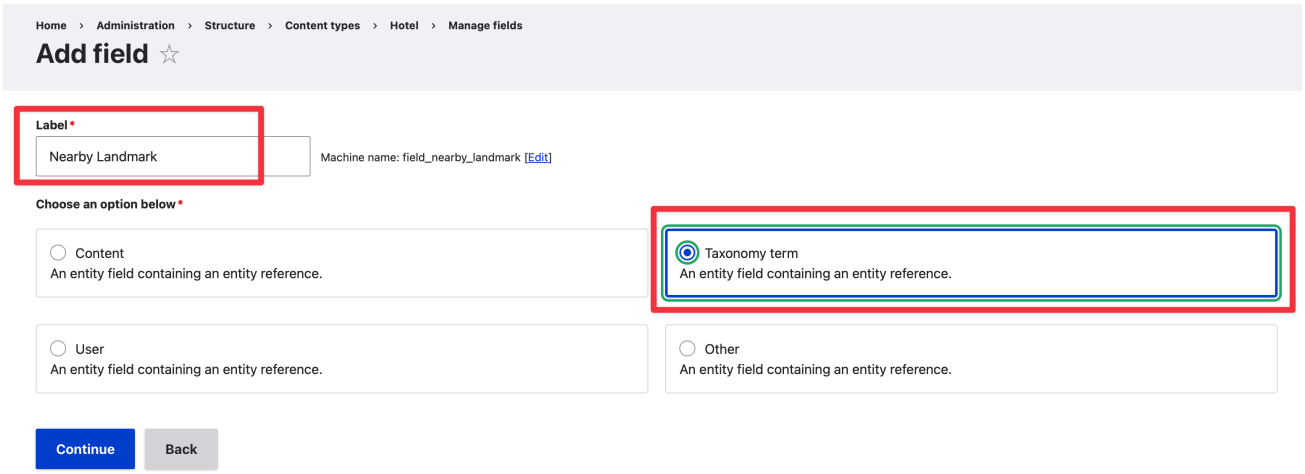
2.3.2 Exercise: Add Field to Categorize Hotel Content and Improve Form

Steps:

1. Go to **Manage** > **Structure** > **Content types**.
2. Select **Manage fields** for the *Hotel* content type.
3. Click the **+Create a new field** button.
4. Select **Reference** and click **Continue**.

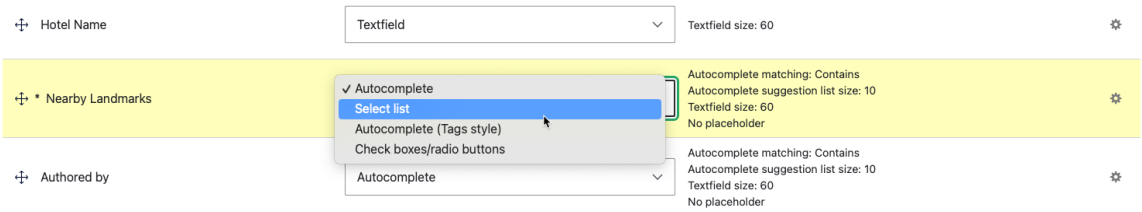


5. Enter for *Label:* **Nearby Landmark.**
6. Select **Taxonomy Term**



7. Click **Continue**
8. Scroll to *Vocabulary in the Reference type section and* click to select **Landmark.**
9. Click **Save settings**.
10. Click the **Manage form display** tab to improve how the content collection form looks to content editors.

11. Click+hold+drag the four-headed arrow to drag the **Nearby Landmark** taxonomy field up under the *Hotel name* title field.
12. Then use the Widget drop-down to change the Nearby Landmark widget to **Select list**



13. Click **Save**.

Pro Tip: As you've been adding fields, you probably noticed the "machine name" being generated as you typed a label. This is the name that is entered into the database and **must be** unique across your entire site. If Drupal detects a duplicate, it will alert you and allow you to change it.

You may only use lower case letters, numbers (but it can't start with a number) and underscores. You may **not** use upper case letters, dashes or any other punctuation.



2.4 Media in Drupal

Drupal core Media module provides an extensible framework for managing files and multimedia assets, regardless of whether they are hosted on your own site or a 3rd party site - it is commonly referred to as a 'file browser to the internet'.

Media is a drop-in replacement for the Drupal core upload field with a unified User Interface where editors and administrators can upload, manage, and reuse files and multimedia assets. Any files uploaded before Media was enabled will automatically take advantage of the many features it comes with.

Features

- Embed files into any text area using a WYSIWYG editor (requires the WYSIWYG module).
- Makes files fieldable, enabling the use of text fields to store captions on images, taxonomy fields for audio file genres, etc.
- Provides a robust display system, allowing images to be displayed via specific image styles, videos to be loaded through third-party players and more.
- Enables the user of remote media from sites like YouTube, Flickr or any oEmbed provider.
- A unified interface for adding new or reusing existing media with file and image fields.
- Administration screens for CRUD dealing with Media.
- Mass importing of files.
- Drag and drop upload (requires the Plupload module).

As Media is an entity in Drupal, we can leverage this to build media-based websites from Video portals to Digital Asset Management systems. Being an Entity, media files can also be added as fields to other entities like Content.

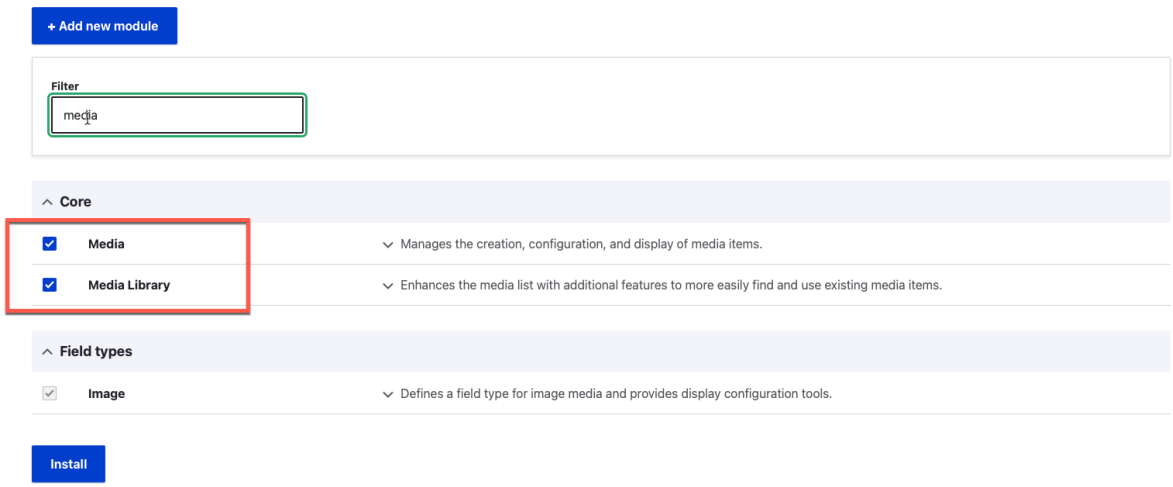
In the following exercises, we will enable the Media and Media Library modules in order to add this functionality to our site, add a field in order to add this functionality to a Content Type, and then test our Content Type.

In the next exercise, we will enable the Media and Media Library modules.

3.4.1 Exercise: Enable Media and Media Library Modules

Steps:

1. Go to **Extend** from the Admin menu.
2. In the search field, search for the keyword "*media*".
3. Enable the **Media** and **Media Library** modules by clicking on the checkboxes next to these modules.
4. Click **Install** to save the configuration.

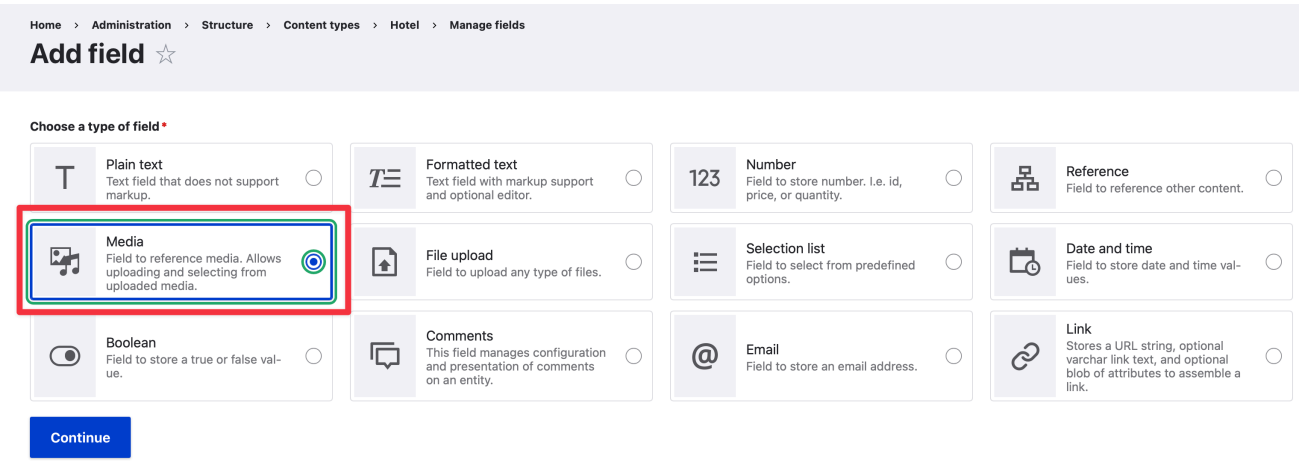


In the next exercise, we will add a video field to our Hotel Content Type.

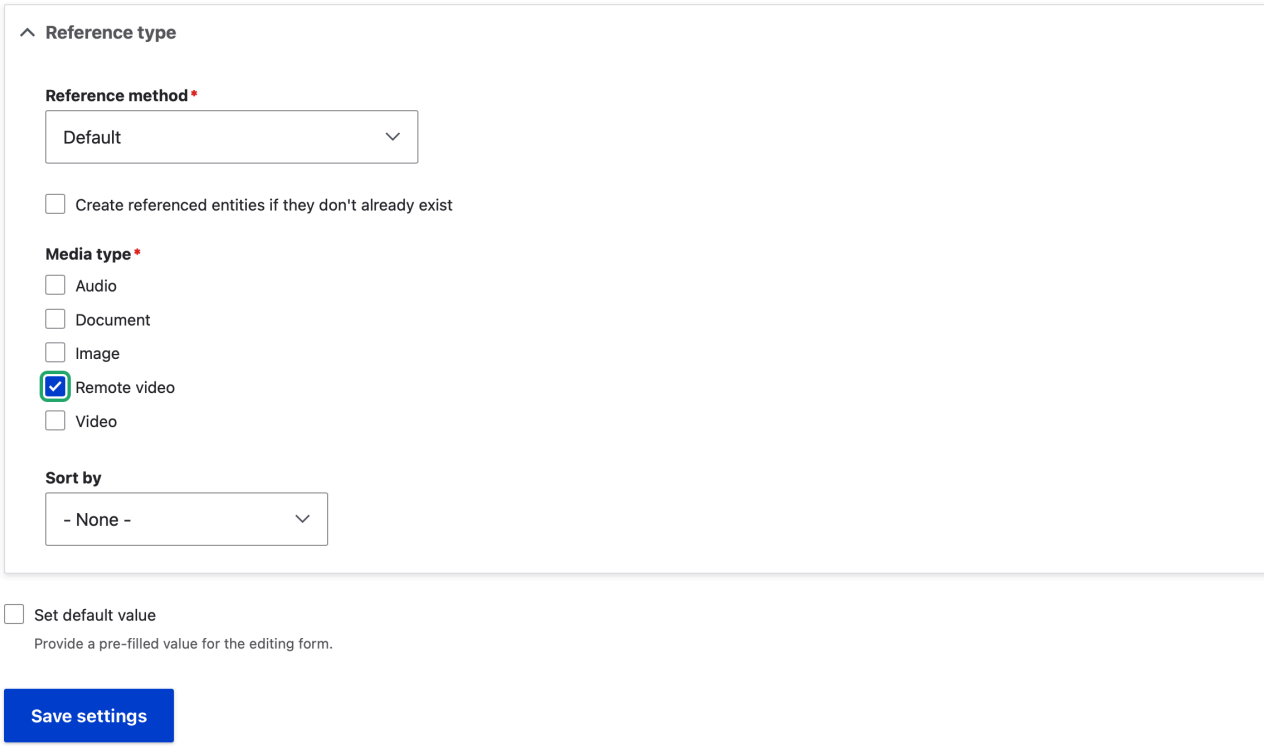
2.4.2 Exercise: Add a Hotel Video Field

Steps:

1. Go to **Structure > Content types > Hotel > Manage fields**.
2. Click on the **+ Create new field** button.
3. Select **Media** and click **Continue.**



4. Enter Label: **Hotel Media**.
5. Click **Continue.**
6. Enter Help text: **Add promotional media of the hotel**.
7. In the **Reference type** section, select *Remote video* for the Media Type.



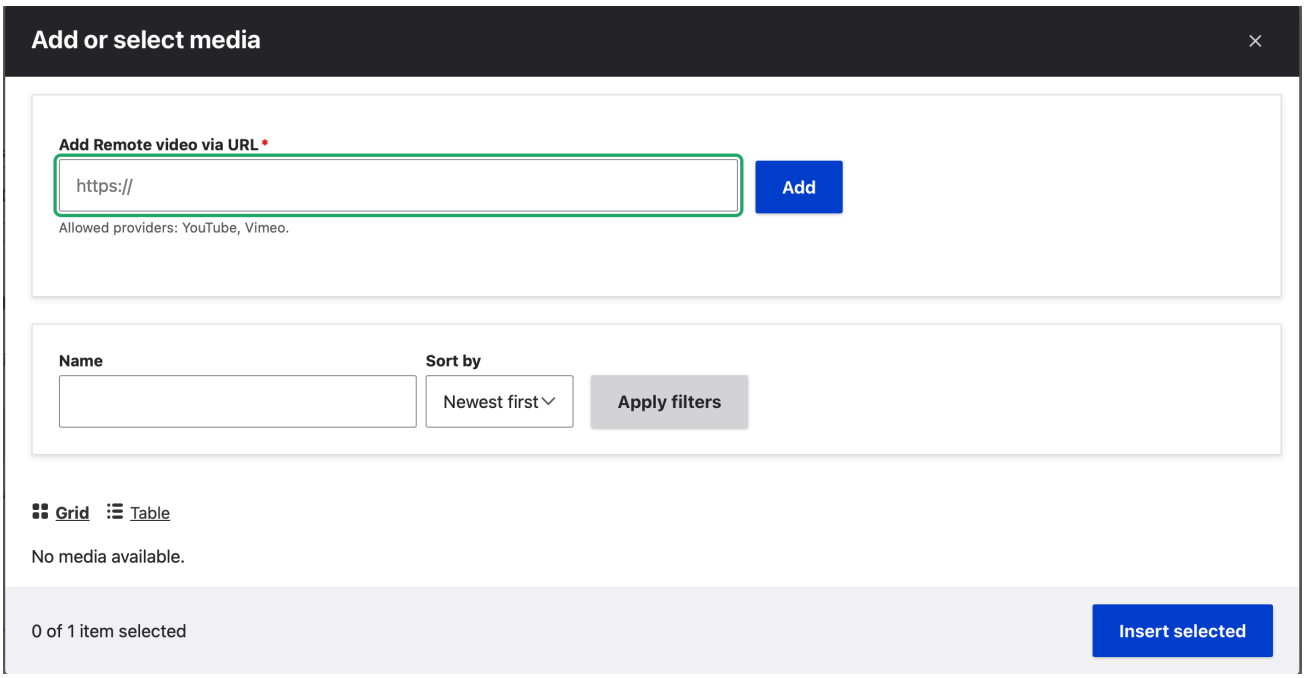
8. Click **Save settings**.

In the next exercise, we will test the Hotel Content Type to make sure the output looks good.

2.4.3 Exercise: Test Your Improved Hotel Content Type

Steps:

1. Go to **Manage > Content**, and find Drupal Hotel in the list.
2. Click the **Edit** button for Drupal Hotel.
3. Observe: The new *Nearby Landmark* field directly under the *Title* field.
4. Click the *Nearby Landmark* select list and choose **Big Ben**.
5. Scroll down to the Hotel Media field.
6. Click **Add media**
7. Enter a link to a YouTube video at the top of modal pop up.
8. Click **Add, Save,** and **Insert Selected.**



9. Click **Save** to update the node.
10. Click **+ Add content** on the Content page
11. Click **Hotel** from the content type list.
12. Populate the *Hotel* details.
    - *Title:* **Pelham Hotel**
    - *Nearby Landmark:* **Hyde Park**
    - *Body:* **Add some lorem ipsum**
    - *Date Rooms are Available:* ***Choose today's date***
    - *Hotel Image:* **upload a picture**
    - *Hotel Media:* ***Copy and paste a*** **YouTube video link, click Add, click Save, then select video and click "insert selected".**
13. Click **Save.**
14. Go to **Manage > Content**, and click **+ Add content** again
15. Repeat step 11-13 to add another *Hotel*, called **The Boutique Hotel** with a nearby landmark of **Oxford Circus**.

2.5 Understanding Content Workflows

An Editorial Workflow is the process organizations follow to create, review, edit, and publish content.

Depending on the size and processes of the organization, multiple people in different roles can be part of the process. For example, content creators collect information and write content, editors review, edit, ask for changes, and publish the content once it's considered ready to be shared with the audience.

In your site, content types can have either a Published or an Unpublished flag to track their workflow states, that you can set for each content item when saving:

Most content entity types support revision tracking, if it is enabled by an administrator for the entity subtype. If you have revision tracking enabled, the software will store old versions of the content each time it's revised and add notes about what is changed. This allows you to look at the history and revert to older versions.

Using the Unpublished flag and Revisions, you can build an editorial workflow that lets content creators and editors work on the live site without displaying the content to visitors. When the content is ready to be published, all they need to do is save it as Published.

Organizations may also want to consider post-publication phases of the editorial workflow: content teams might need to update already published content. Some organizations might need a more complex workflow with the options to configure which content type goes through the workflow process, set up user roles and permissions, set different workflow states (for example draft, needs review, published, or archived).

2.5.1 Exercise: Inspect Content Moderation Module

Steps:

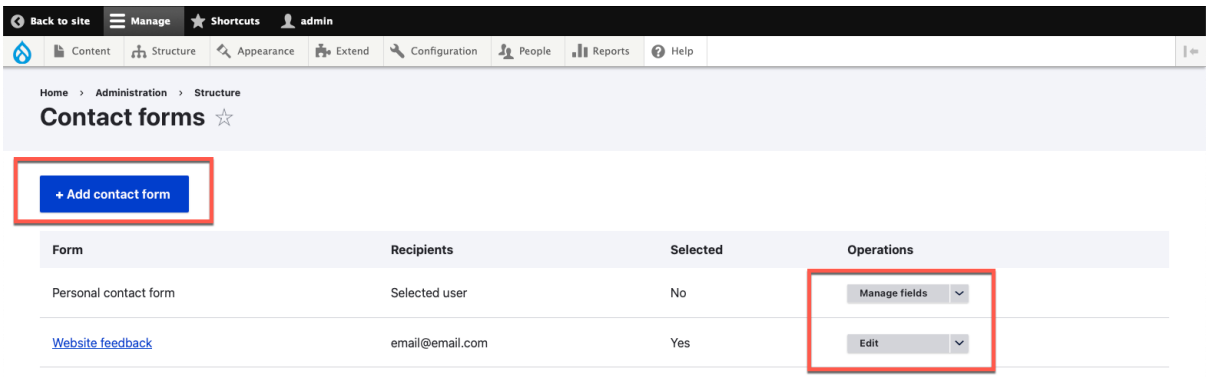
1. Go to **Manage > Extend**
2. Select the **Content Moderation** and **Workflows** modules under the *Core* section. Scroll to the bottom of the page and click **Install**
3. Go to **Configuration** > **Workflow** > **Workflows**
4. Click the **Edit** button for the pre-built *Editorial* workflow.

5. Inspect the configuration options to build *States* and create *Transitions*
6. Under *This workflow applies to:* click the **Select** button to the right of *Content types*.
7. Click the checkbox for **Basic Page** and click **Save**.
8. Click **Save** again on the main Workflow page.
9. Go to Content > Add content > Page on this form you will now see the workflow options available above the Save button.

2.6 Understanding Contact Forms

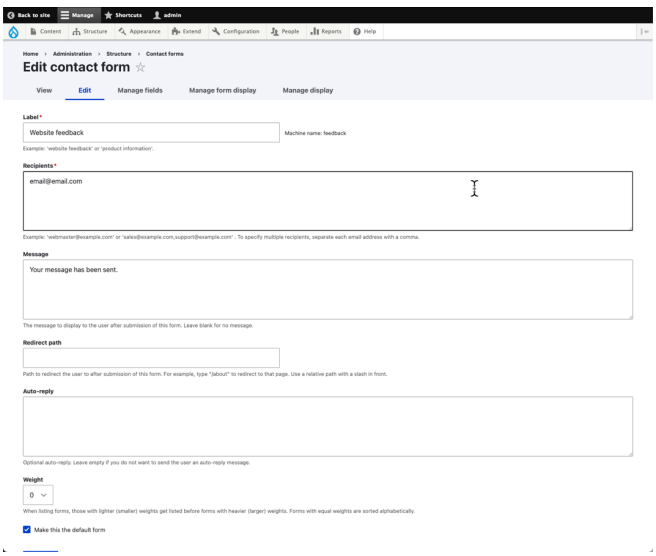
Drupal provides a feature to configure and use personal and site-wide contact forms. The default site-wide contact form is available at the URL */contact/feedback*. You can configure the default form or add new contact forms by going to Structure > Contact forms.

We have provided a glimpse of Contact forms here and encourage you to try out creating Contact forms on your own.



You can manage existing forms or add new Contact forms from this page.

By editing an existing form, you can configure various settings related to the form. Also, like any other Drupal entity, you can add new fields and manage the display of the fields for the forms.



Learning Module 2.0 Summary

In this learning module, you learned that:

- Drupal comes with two built-in content types, Article and Basic Page.
- Structured content is preferred by site developers so they can search, sort, and filter more accurately.
- You can create custom content types to collect content of different types.
- Fields settings allow you to customize data collection down to the smallest details. Use fields to structure new content entry as clearly as possible.
- Fields can be added to content types without affecting existing content.
- Additional field types can be made available by installing relevant core or contributed modules
- Drupal Core Media module provides functionality for managing rich media content in your website.
- Taxonomy is used to categorize content in Drupal.
- You can create as many Vocabularies as needed; they are all independent of one another unless you tie them together.
- Vocabularies contain Taxonomy Terms. Each Term can have a child Term, so vocabulary hierarchy can be as deep or as wide as needed to reflect any collection of terms.
- The Taxonomy Term field allows you to tie Vocabularies to any fielded entity for categorization.
- On a Content Type's Manage Form Display tab, you can make your content editing form easier for editors to use.
- Personal and site-wide contact forms can be added and configured in your website.

3.0 User Roles and Permissions

One of the great features of Drupal is the ability to control how and what people can access on your site. You can set permissions for these "users" to define who can do what for the site functionality provided by Drupal core and contributed modules. For instance, you don't want casual visitors to edit your homepage! However, the site owner or trusted user should be able to edit the homepage or other various content.

"Out of box", Drupal core comes with four user roles:

- **Anonymous user:** this role is used for users that don't have a user account or that are not authenticated.
- **Authenticated user:** this role is automatically granted to all logged in users.
- **Content Editor**: this role is give to editors who will only see the Content tab.
- **Administrators:** They have all permissions.

Roles can be given or removed from any site user. The exception is the first user created (user/1). The user id 1 account ("user/1"), sometimes called "superadmin" is able to do everything on a Drupal site. Drupal access checks will always allow this account to take any action. Another special permission is that user/1 is the only user that can execute the 'update script', a script required to run after updating to new versions of contributed modules or Drupal core.

User/1 is the only account that has permanent/non-removable access to everything based on the account itself. All other users can have permissions added or removed as their status changes over time. User/1 has access to all permissions, all the time.

3.1 Creating Roles and Assigning Permissions

One of the most basic features of administering user web site permissions with Drupal is through the creation of roles.

Rather than assigning individual permissions to each user, permissions are assigned to roles and roles are then assigned to users. Using roles gives administrators greater control over permissions and makes it easy to assign and remove roles when necessary.

Permissions let you control what users can do on your site. Each user role (defined on the user roles page) has its own set of permissions. For example, you could give users classified as Administrators permission to administer nodes but deny this

power to ordinary, authenticated users. You can use permissions to reveal new features to privileged users (those in charge of creating and editing content, for example). Permissions allow trusted users to share the administrative burden of running a busy site.

Permissions given to the Authenticated user role are inherited to all the other roles/ logged in users, except the Anonymous user role. If you assign a permission to the Anonymous user role, that permission is not inherited by other roles.

For example, if you give the Anonymous role the permission to post comments, it does not mean that the Authenticated User role also gains permission to post comments. Yet, if you give the Authenticated User permission to post comments, it does mean that other roles you have created will automatically gain that permission.

Roles allow you to fine-tune the security and administration of Drupal. A role defines a group of users that have certain privileges as defined in user permissions. Examples of roles include anonymous user, authenticated user, content editor, administrator, and so on.

In the exercises that follow, you will learn how to create a content editor role and learn to manage People, Permissions, and Roles.

3.1.1 Exercise: View the Content Editor Role

Steps:

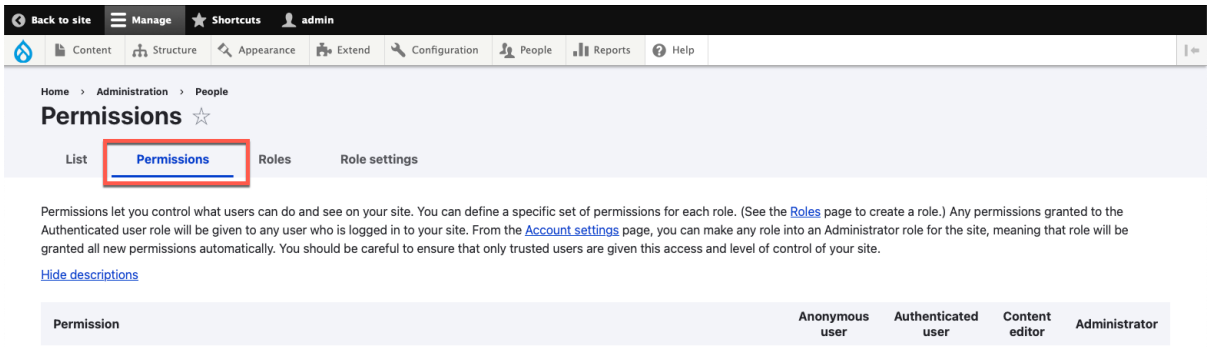
1. Go to **Manage** > **People**.
2. Click the **Roles** tab
3. Note the Content Editor Role has already been added.
4. Notice that you can rearrange the order of the Roles. Doing so re-orders the list of Roles on the Permissions page.

Permissions let you control what users can do on your site. In the next exercise we will be assigning permissions to the Content Editor role to allow them to create, edit, delete, and manage site content which includes hotels.

3.1.2 Exercise: Assign Permissions to the Content Editor Role

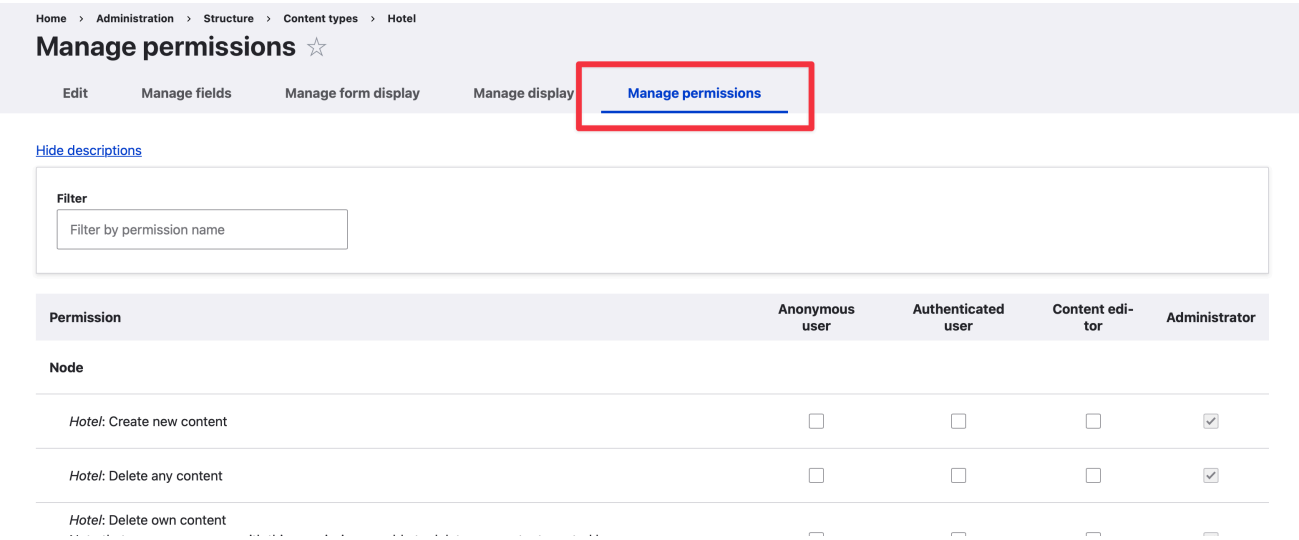
Steps:

1. Go to **Manage > People.**
2. Click the **Permissions** tab.



3. Scroll down to the **Node** section of the Permissions tab and check to give the Content Editor all node-related permissions.
4. Scroll down to the **Toolbar** section and check the box to give the Content Editor the *Use the administration toolbar* permission, Access Media Overview and Administer Media permissions.
5. Scroll to the bottom of the page and click **Save permissions**.

Pro Tip: Permissions for **all entities** are now managed at the entity itself.



Notice that we have given the Content Editor quite a bit of power regarding the editing of content. The Edit any and Delete any permissions mean a user of that role may impose their will over any content, which is why the Content Editor role should

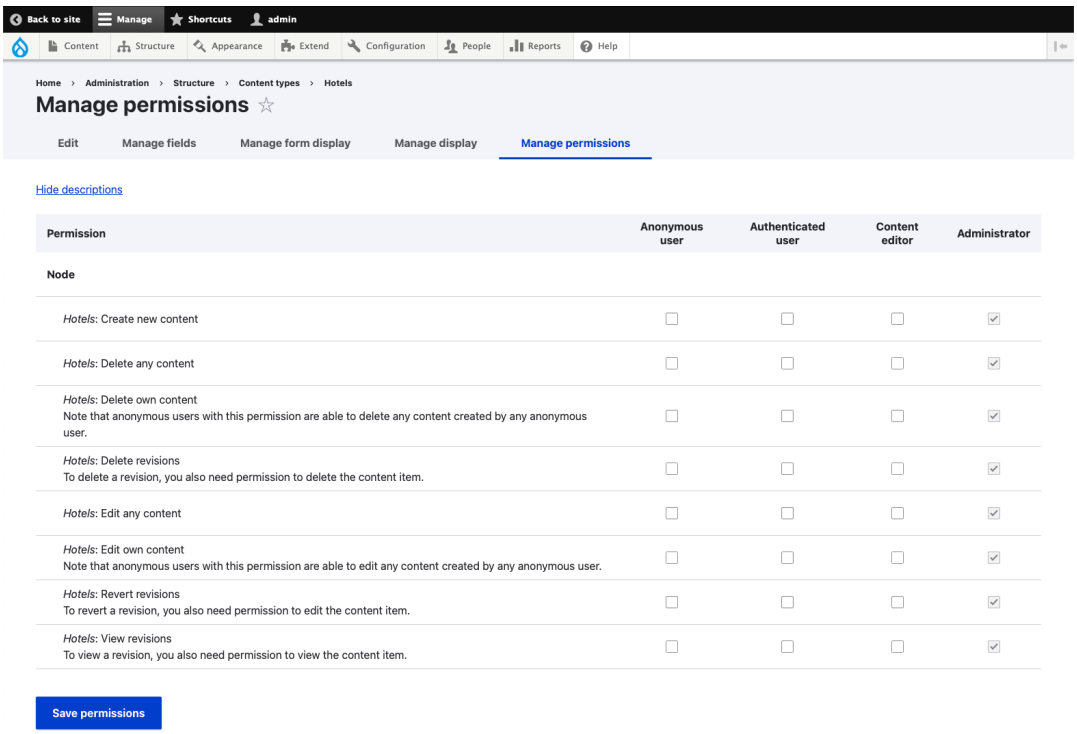
only be given to trusted individuals! Often this power is further refined through the creation of additional roles such as Content Moderator who are granted the Edit/ Delete any permissions with those same permissions taken away from the Content Editor role.

3.1.3 Exercise: Assign Permissions for a Content Type

Drupal also splits out this process for Content Types. This really helps speed things up!

Steps:

1. Go to **Structure > Content Types > Hotels > Manage Permissions**.



2. Find the *Hotel: Create new content* permission.
3. Observe eight new permissions are automatically created for each new content type created.
4. **Click the checkbox** for the Content Editor User role.
5. Click Save permissions.

3.2 Account Registration Settings

Drupal provides three methods by which a person can become registered on the site:

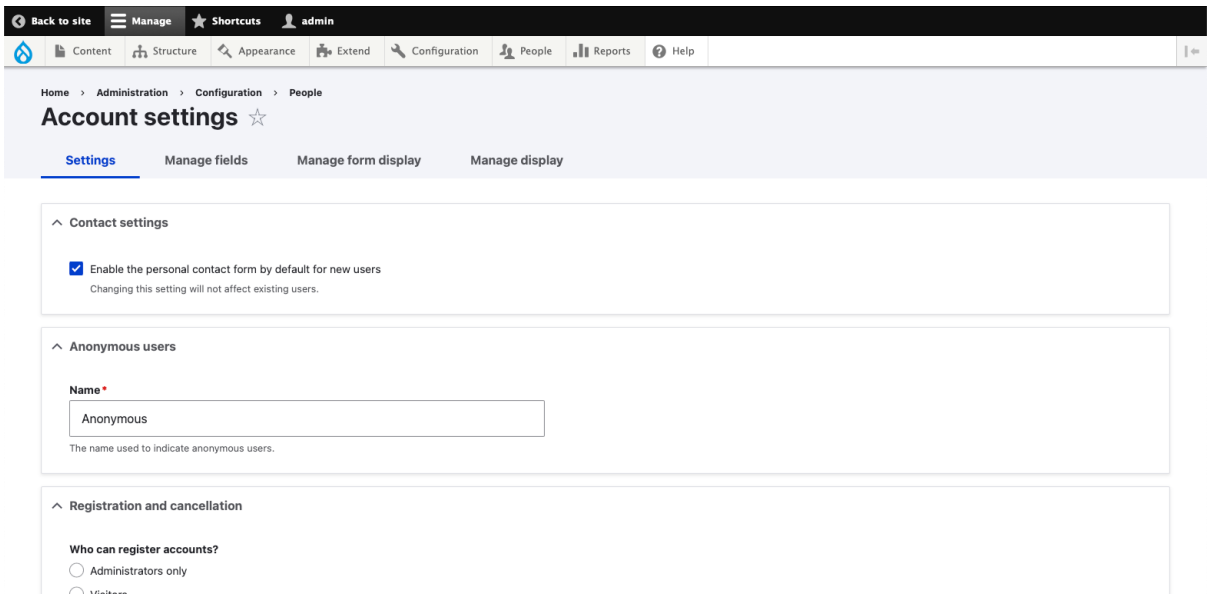
- A traditional User Registration form, no moderator approval required.
- A traditional User Registration form, moderator approval required.
- Registered via Administrator.

In the exercise below, we will look at ways to register users onto your Drupal site and configure their account registration settings.

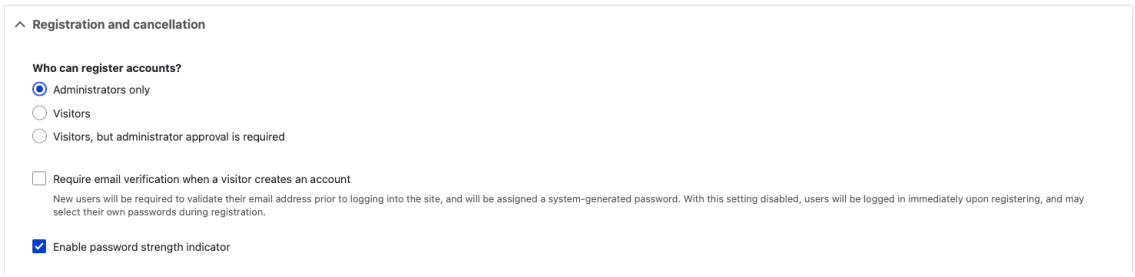
3.2.1 Exercise: Configuring Account Registration Settings

Steps:

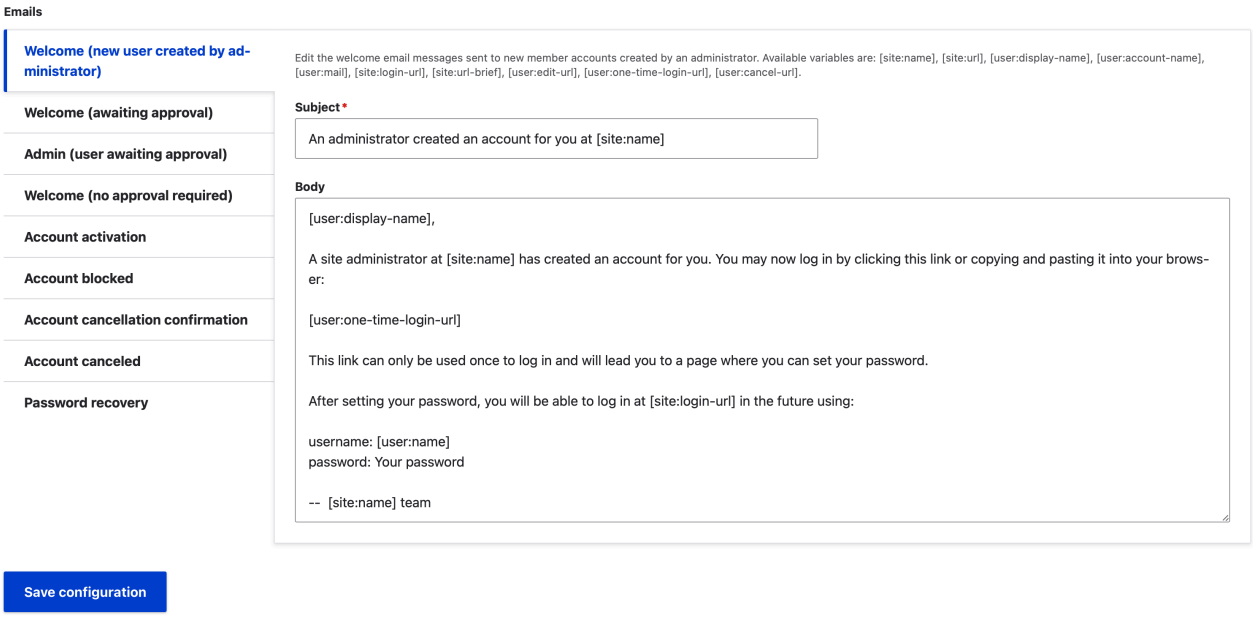
1. Go to **Manage > Configuration > People > Account settings**.
   Note: This area controls the method of user registration, as well as whether or not email verification is needed, along with other settings.



2. Scroll to the *Registration and Cancellation* section.
3. Confirm that **Administrators Only** is selected under *Who can register accounts*.
4. Uncheck the **Require email verification when a visitor creates an account** checkbox. *Note: This configuration removes any filters for controlling who can access the site as an Authenticated user.*



5. Observe: You can customize the automated emails that are sent on your site's behalf. Leave the email templates alone for now.



6. Click **Save configuration.**

3.3 The Authenticated User Registration Form

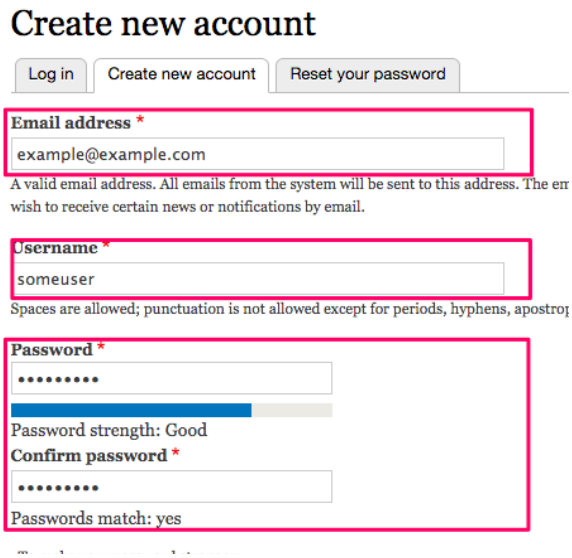
The Authenticated User role is applied to users who have registered and are logged in. With our current permission settings, it allows them to view published content and make comments (if comments are enabled). All users are assigned the Authenticated User role when they register.

In the following exercise we will examine what the User registration form looks like and create a new user using the Registration form.

3.3.1 Optional Exercise: The Authenticated User Registration Form

Steps: (This only works if you did NOT change "who may register users" to Administrator only).

1. Click **< Back to site** to see the public facing homepage.
2. Click the **Log out** in the top right corner of the header.
3. Click the **Login** link in the top right.
4. Click the **Create new account** tab.
5. Fill out the form to create a new account with a dummy email address. You only need to fill in the required fields:
    1. *Email address:* **example@example.com**
    2. *Username:* **someuser**
    3. *Password:* **admin**



6. Click **Create new account**.
7. Observe: Your new user account has been added.
8. Click **Log out**.

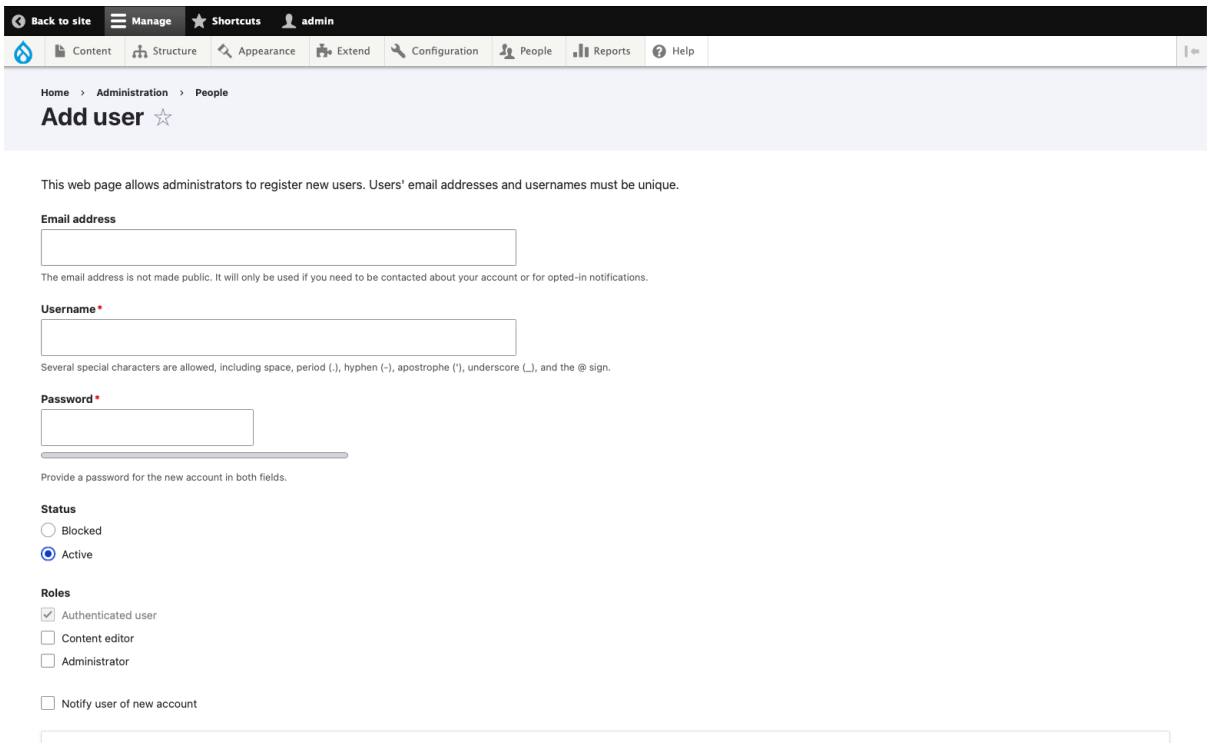
3.4 Creating a Content Editor Account via Administrator

New users who create their own account are automatically assigned to the *Authenticated* role. That is how we want it. However, if we want to give a user the *Content Editor* role, we must do that manually using an *Administrator* account.

3.4.1 Exercise: Create a New User with the Content Editor Role

Steps:

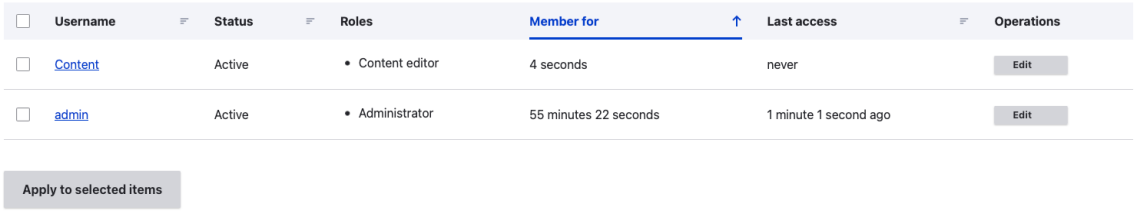
1. Login as **admin**.
2. Go to **Manage** > **People**.
3. Observe the newly registered user from the previous lesson.



4. Click **+Add user**.
5. Observe: The form is similar to that provided on the User Registration form, with the addition of the Status and Roles sections.
6. Add a user with the following details:
    6.1. *Email address:* **contenteditor@example.com**
    6.2. *Username:* **Content**
    6.3. *Password:* **content**
    6.4. *Status:* **Active**

6.5.*Roles:* **Content Editor**

7. Leave the remaining defaults.
8. Click **Create new account**.
9. Return to **Manage** > **People** to see all the registered users and their roles.



Advanced functionality using contributed Group module

Out of the box, Drupal only allows a person to view, create or edit the different content types you create. However, you might want to be able to let them view or edit all content types, but only in a specific section of the site.

The Group module (https://www.drupal.org/project/group) allows you to create arbitrary collections of your content and users on your site and grant access control permissions on those collections

Group also allows you to create communities. Users of the site can become members of a Group. You can then decide what membership of a Group entitles them to, such as access to content only available to members of that Group. There can also be different levels of membership to a Group so, for example, you can have administrative members who can create content and basic members who can only view content within the Group.

Learning Module 3.0 Summary

In this Learning Module, you learned that:

- The registration form is shown according to how the Account Settings are configured.
- When a user registers using the registration form, they are assigned the authenticated user role by default.
- When an administrator creates an account, the new account can be assigned to any of the roles that are in the system.

- It is possible to create an account with multiple roles and the permissions of each role will be combined.

4.0 Building What Appears on the Page

At this point, as you review the site, you can see the main links in the menu, and your article on the home page. Yet there is no way to browse the hotel content across the site or navigate quickly to content you've created. We'll learn how to manage menus and lists of content in this learning module. First, we'll review how Drupal structures what appears on the page.

Learning Objectives

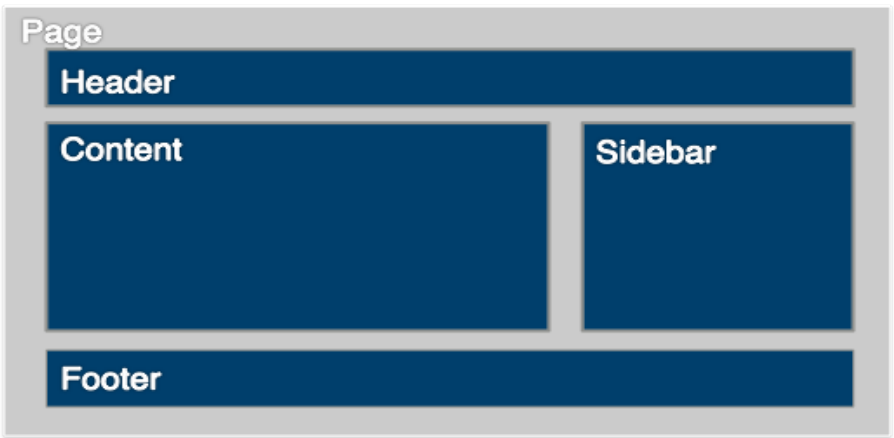
At the end of this Learning Module, you will be able to:

- Understand how Drupal pages are subdivided
- Add Blocks to Regions and set their visibility
- Create a custom Block
- Create a list of content with Views
- Understand Views Displays
- Manage URL aliases and the Menu system

4.1 Page Layouts with Blocks

Blocks are pieces of HTML content (such as User Login or Who's online or a list of Most Recent Blog Posts). Blocks are arranged into areas of your page layout called Regions. The available regions, and even the names of the regions, are defined by the theme.

Since they are defined by the currently installed theme, we consider the available regions to be arbitrary. In the image below there are four regions.



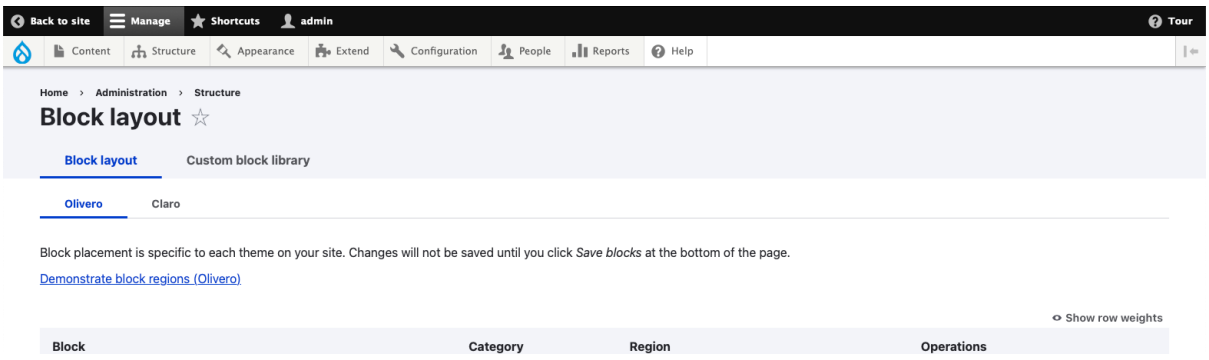
The above illustration should help you visualize how a Drupal page is subdivided. We use those divisions, called regions, as containers for our blocks.

In the next exercise, we will manage the blocks on our page so that the search bar appears in the right sidebar.

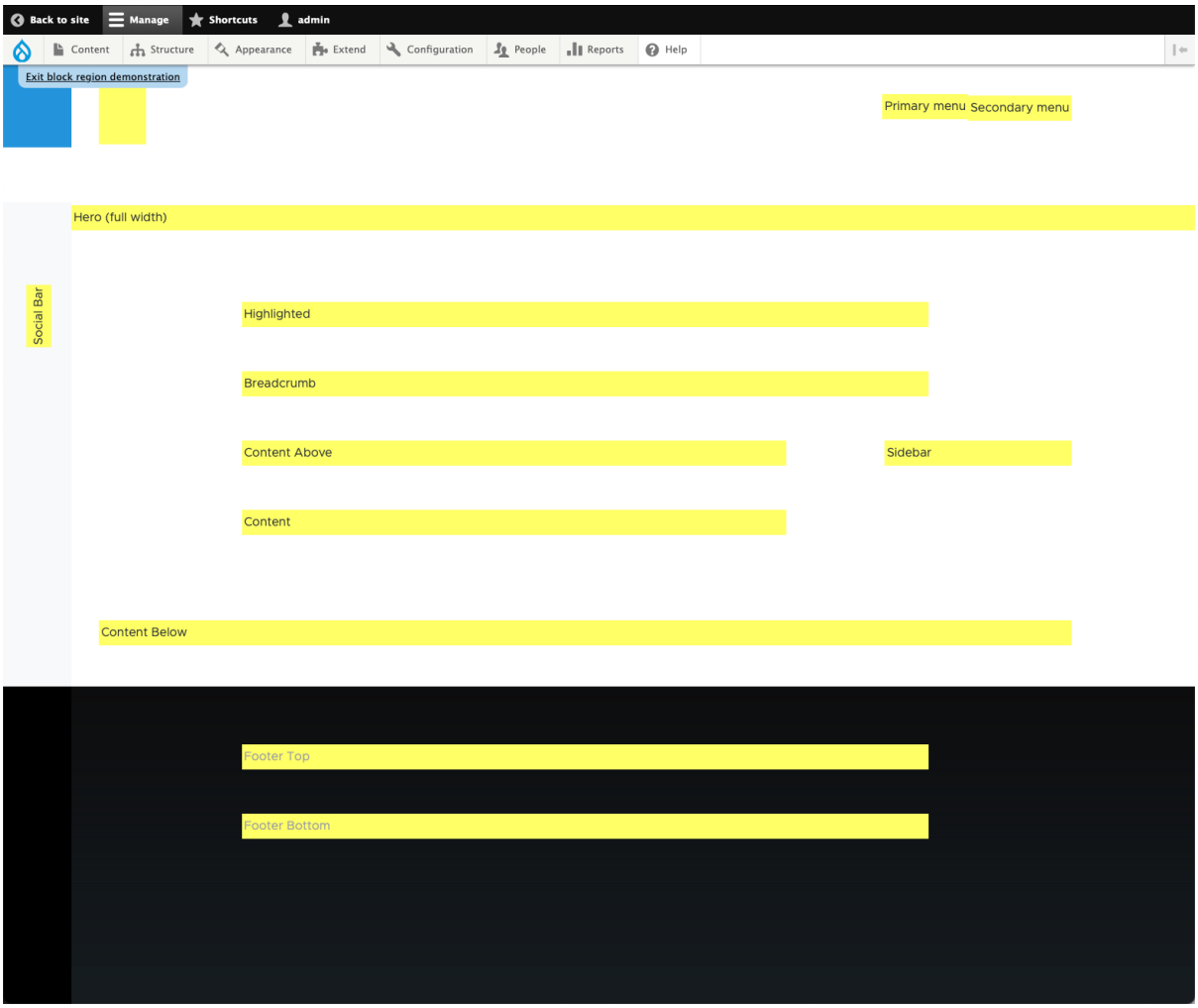
4.1.1 Exercise: Rearrange Blocks

Steps:

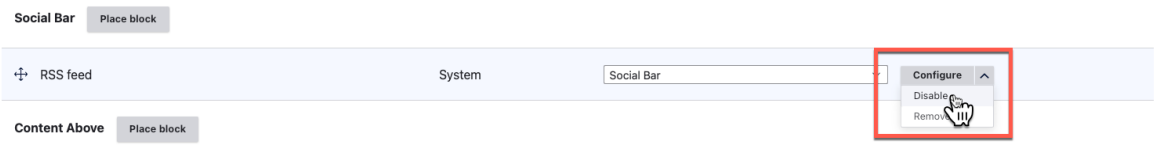
1. Go to **Manage** > **Structure** > **Block layout.**
2. Click **Demonstrate block regions (Olivero)** link in blue.



Note: This will open a visual display of the regions available in your current theme.



3. Click the **Exit block region demonstration** link. You will return to the *Block Layout* page.
4. Locate the **RSS feed** block.
5. Click down arrow next to *Configure* and select **Disable**.



The *RSS feed* is no longer displayed.

4.2 Custom Blocks

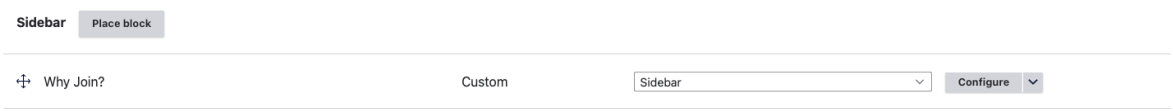
In Drupal, Blocks are very flexible entities. They can contain HTML text or functionality built by modules. You can even build your own custom blocks. Custom blocks, and blocks in general, can be used more than once in the site layout.

In the next exercise, you will create a custom block called "Why join?" and display it in your sidebar.

4.2.1 Exercise: Create a Custom Block

Steps:

1. Go to **Manage** > **Structure** > **Block layout**.
2. Scroll down to the **Sidebar** region.
3. Click **Place Block.**
4. Click the **+ Add Content block** button.
5. Define the block.
    5.1. *Block description:* **New Members**
    5.2. *Body:* **Use some lorem ipsum filler text.**
    5.3. *Revision log message:* **Initial block text**
6. Click **Save**.
7. Configure the block:
    7.1. *Title:* **Why join?** (This is the title users will see).
    7.2. Under *Visibility* > *Roles* select **Anonymous user.**
    7.3. *Region:* Select **Sidebar.**
    7.4. Click **Save block**.
8. Click **< Back to Site**.
9. Question: How can you test to see your new block is displaying as configured?

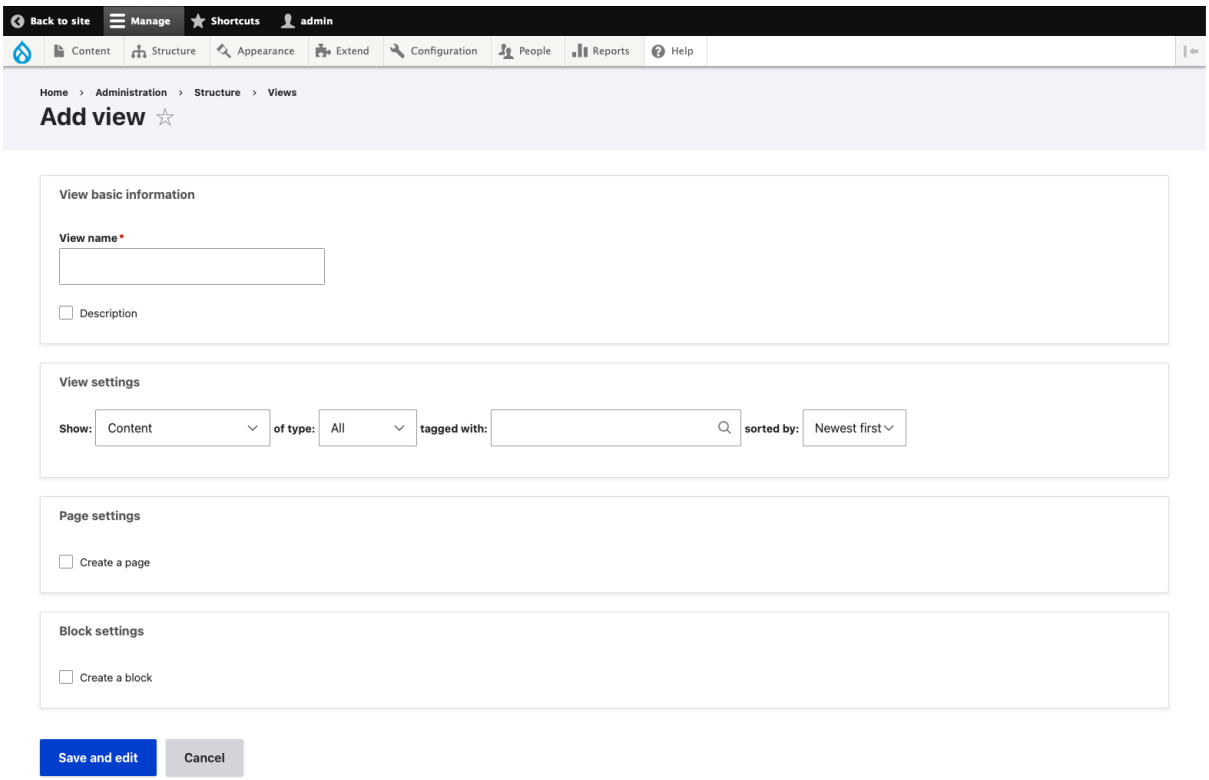


4.3 Building Dynamic Lists of Content with Views

To teach you how to build dynamic lists of content with Views we must first give you an introduction to the Views module. Views as it is usually called, allows Drupalists to easily query data from the database (content, users, comments, etc.), filter, sort, and choose from pre-built formats for display. Each query is known as a view, and the output of a view is known as a display.

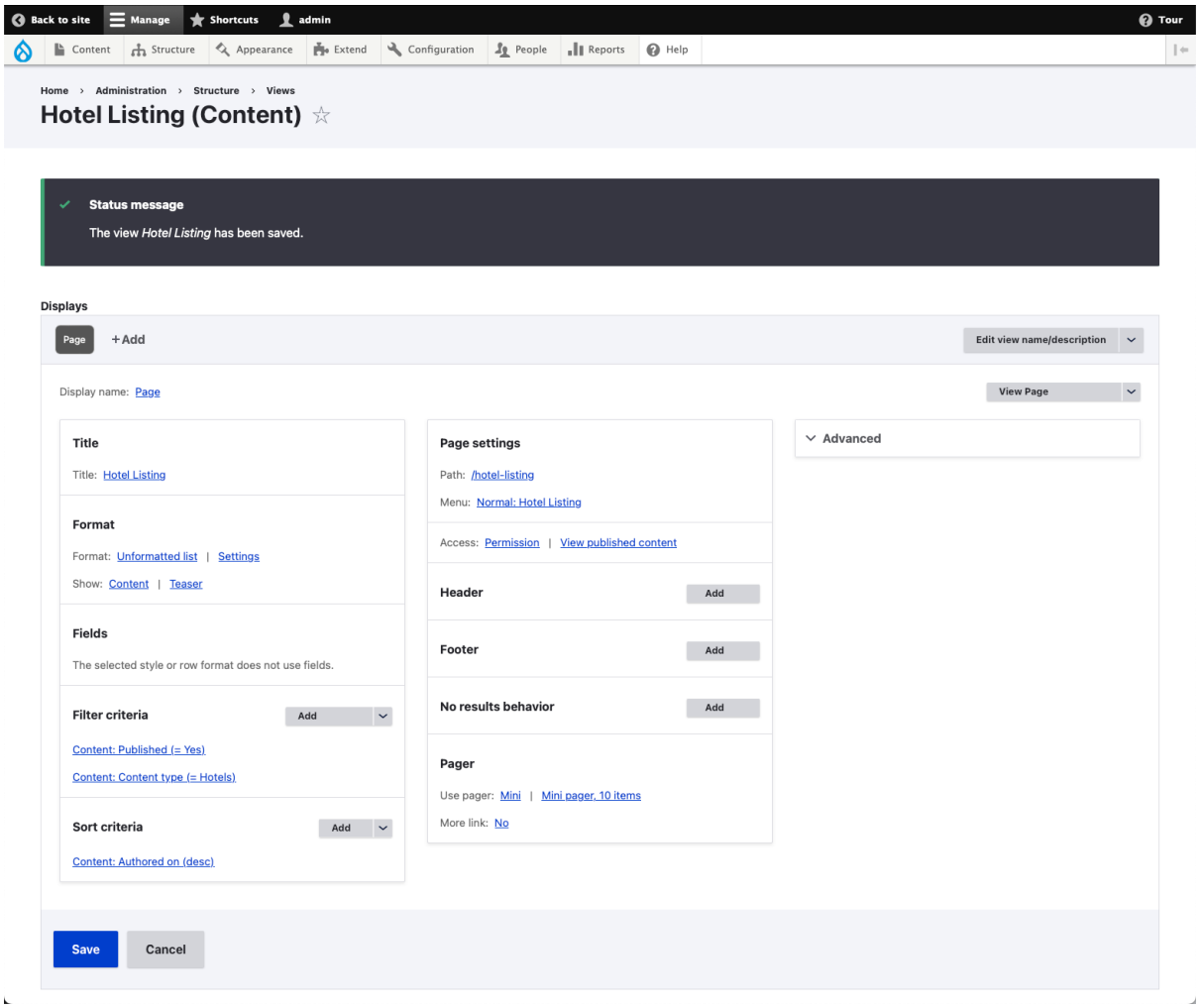
The Views Editing Page

To define a View you'll always start with the Wizard page. It's just a form that asks the most important things about the query you want to build. After that initial definition, you won't see that page again.



Anytime you need to configure an existing View, you'll be editing on the Views Editing page.

That page looks like this:



There are three columns:

- **Left:** The items in this column are the same for all display types. Display format, Fields you want to display, your Filter Criteria, and your Sort Criteria.
- **Center:** The items in this column are specific to the display type you are editing. So for example, a Page display would have Page Settings in the center column. Some of these settings might be permissions, your Header and your Footer, and how many results you want to display on a page.
- **Right:** The items in this column are hidden by default. They include settings that don't need to be used unless you are creating a complex View display. Some examples of what you'll find in this column (same for all displays) are Contextual Filters, Relationships, and No Results Behavior.

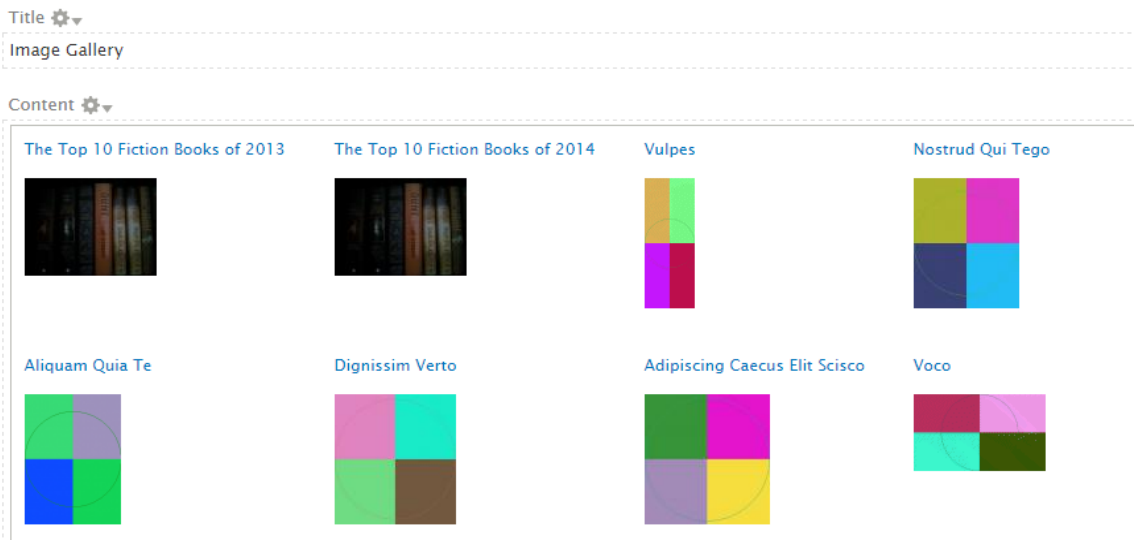
Having so many configuration options can seem intimidating at first, but in reality, a lot of your work will be done in the Left column. The left column contains all you need to get a basic View working.

The Views Preview Area

Underneath the three columns is the preview area. It shows you the results of your query on the fly, so you can check that the filters you implement are actually doing what you expect.

If you change any of the fields around in your View, such as adding them or positioning them in a different way, you'll also be able to check the preview to see if that worked as you intended.

Here's *an example* of what the preview looks like when making an Image Gallery View populated mostly with content generated with the Devel module. On your live site with data that is more realistic, the colored squares would be all the images in your gallery.



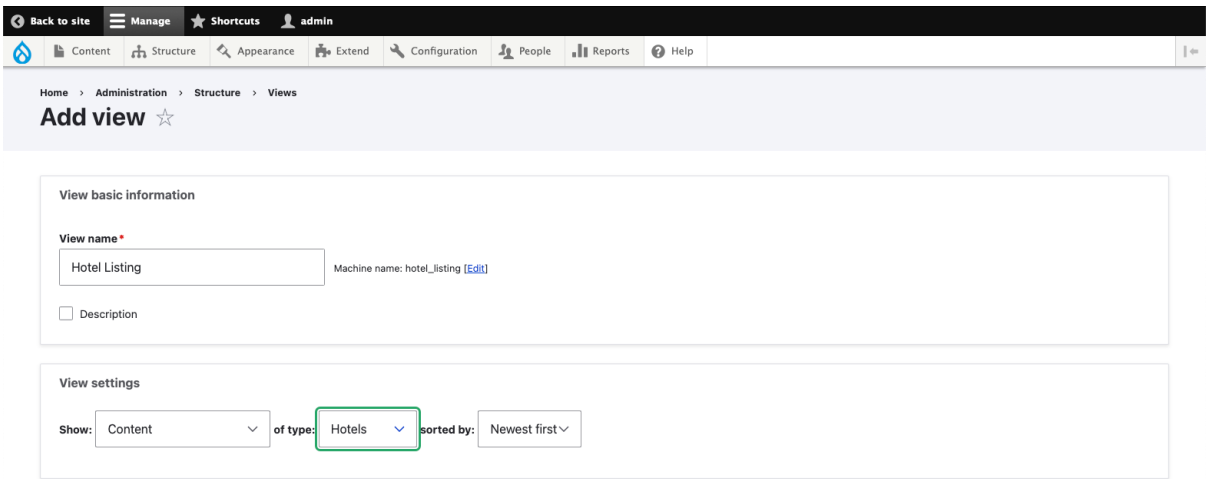
The major benefit of using the preview is that you can see the results of your configuration without having to save and exit the Views module.

Now that you have a good overview of the Views interface, in the next exercise we will create a page that lists all of our hotels. The page will be linked from the main navigation menu.

4.3.1 Exercise: Create a View Listing all Hotels

Steps:

1. *Login* as **admin**, if necessary.
2. Go to **Manage** > **Structure** > **Views.**
3. Click **+ Add view**.
   This loads the Views wizard that you only see when you initially create a view.
4. *View name:* **Hotel Listing**.
5. *Show:* **Content** *of type:* **Hotel** *sorted by:* **Title**



6. Under *Page settings*, select **Create a page**
7. Select **Create a menu link** that expands options
8. Select **Main navigation** at the very bottom of the long list.
9. Keep the remaining defaults.
10. Click **Save and edit**.
11. Scroll down to the *Preview* area of the Views editing page to see the content as it would appear on your site.
12. Click the **Save** button (located above the Preview area) to finish creating your view.
13. Click **< Back to Site** to see your *Hotel Listing* in the Main Menu.
14. Click on the **Hotel Listing** menu tab to see the results of your query.

4.4 Views Displays

When you build a View you're essentially building a query. How the query results are going to be formatted, filtered, and sorted is determined in the Display settings. One View can have many Displays.

There are a handful of display types available when you edit View settings and more types become available as different modules are installed.

Examples of Views displays:

- A page display
- A block display
- An attachment
- An RSS feed

Illustrations of the above Views displays:



You might output your View as a page display formatted as a table, or a block display formatted as a bulleted list, or you could do both.

In the next exercise we will create a view listing Hotels in a block to place in the sidebar.

4.4.1 Exercise: Create a Hotel List Block Display

Steps:

1. Go to **Manage** > **Structure** > **Views**.
2. Find the *Hotel Listing* view
3. Click the **Edit** button to the right.

4. Find *Displays* near the top left corner.
5. Click **+ Add**



6. Select **Block** to add a new block display to your Hotel Listing view.
7. You'll notice the new *Block* display button appears next to the *Page* button.



Now we'll edit the Block display settings. To assure you are editing the correct display, make sure the Block button is selected (dark grey).

8. Click on **Unformatted List** link in the left column, under *Format*.
9. Observe: At the top of the style options screen you'll see the word *For* and a drop-down box. This is where you select which display will be affected by your changes. In this case, we only want the HTML List formatting for our Block display.
10. Click **All displays** in the drop-down and choose **This block (override).**
11. Select **HTML List**.
12. Click **Apply (this display)**.
13. Click **Apply** again on the next screen.
14. Click **Content** in the left column, under *Format, Show:*
    Note: This will change what is displayed in the HTML List.
15. Select **Fields.**
16. Click **Apply (this display)**.
17. Click **Apply** again on the next screen.
18. Scroll down to preview your bulleted list of Hotel Names.
19. Scroll back up.
20. In the center section, click on **None** next to Block Name.
21. Enter Alternate Hotels and click **Apply.**
22. Click **Hotel Listing** In the left column, under *Title*.
23. Change *For All Displays* to **This block (override)**
24. Change the title to **Available Hotels**.
25. Click **Apply (this display)**.
26. Click **Save** to save your new display.

27. Go to **Manage > Structure > Block layout** to place the new block.
28. Click **Place block** next to the ***Sidebar*** region.
29. Filter on **Hotel** to find your new block.
30. Click its **Place block** button.
31. Keep the default settings.
32. Click **Save Block**.
33. Click **Back to Site.**
34. Observe: *Available Hotels* listing in the right sidebar.

4.5 Custom URL Aliases

By default, Drupal lists all pages on your site with a URL that reflects the database location of the content being displayed. This means that the URL for a content page called About Us might appear in the address bar as www.example.com/node/4.

That's not the friendliest-looking URL. Chances are, you'll want to be able to customize paths for your site, so navigation is easier, and so that Search Engines can get a better idea of how your site's pages are interrelated.

You will often see the opportunity to set a path or URL alias in Drupal. You can set a custom URL alias when creating a node or a view. Consider path aliases that are easy to understand and relevant for the content of the page.

You may want all Articles to have a path including /article/ or all Blog Posts to have /blog/ in the path. This is possible to do with some contributed modules in Drupal.

Pathauto, Token, and Ctools Modules

Most webmasters prefer to set up more meaningful URLs. Using descriptive language in the URL makes it easier for:

- visitors to remember a page name,
- search engines to weigh page relevancy,
- website developers to set rules.

When the Pathauto (www.drupal.org/project/pathauto), Token (www.drupal.org/project/token), and CTools (www.drupal.org/project/ctools) modules are enabled, Drupal allows you to create your own alias patterns that are automatically applied to new pages. This may be useful for many generic purposes, but webmasters will benefit by defining custom paths for their content, taxonomy and user pages.

The Pathauto module lets you define your custom paths, the Token module gives you replacement patterns (tokens) to represent variables in your path, and CTools provides some utilities needed to make those modules work. There is an optional exercise on the use of Pathauto in the appendix of this course.

Install Required Modules

For using the Pathauto module, you need to install the module along with its dependencies. **Install the following modules** using the method of your choice. The composer command is `composer require drupal/pathauto`. This will install all three.

Pathauto : https://drupal.org/project/pathauto

Token: https://drupal.org/project/token

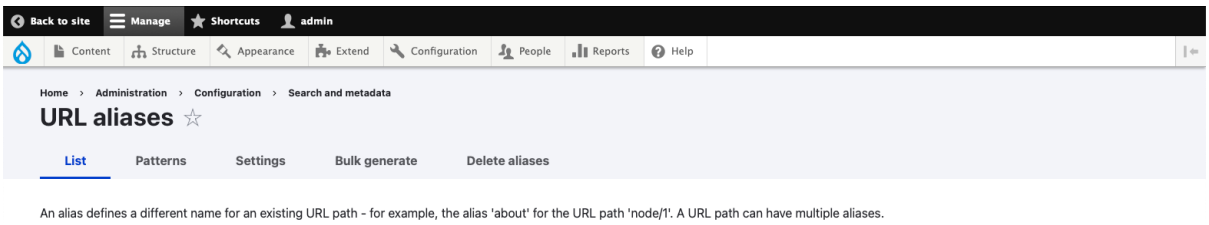
Chaos Tools Suite: https://drupal.org/project/ctools

`drush en pathauto, ctools, token -y` will enable the modules via the command line using drush.

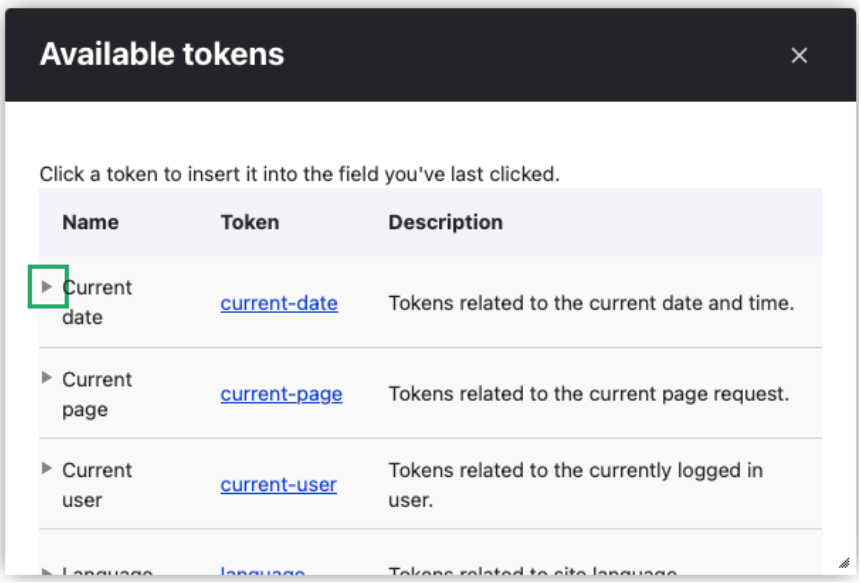
Exercise 4.5.1 : Configure Automatic Path Patterns

Steps:

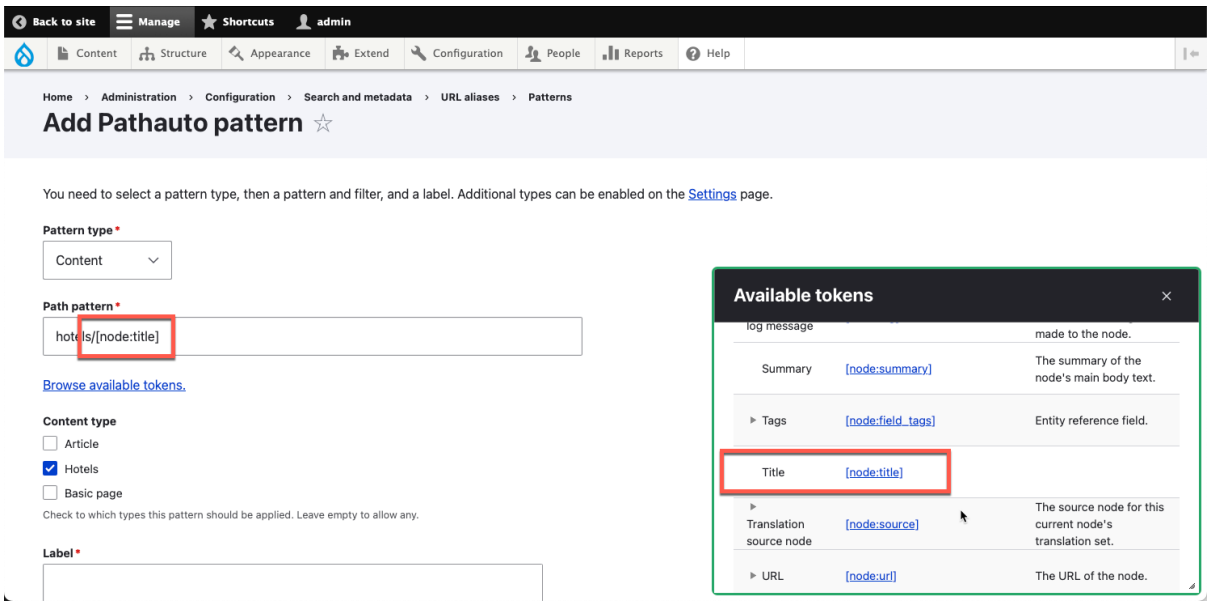
1. Install and enable Pathauto, Token and Chaos Tools modules
2. Navigate to **Configuration** > **Search and Metadata** > **URL aliases** page.
3. Go to the **Patterns** tab and click **+ Add Pathauto pattern** button



4. On the following page, select **Content** as a *Pattern type* from the dropdown. The page will display additional configuration fields for the selected Pattern type.
5. In the *Path pattern field*, type **hotels/** and click **Browse available tokens** link
6. In the **Available tokens** pop up window, click the arrow to the left of **Nodes** to expand the Nodes section.

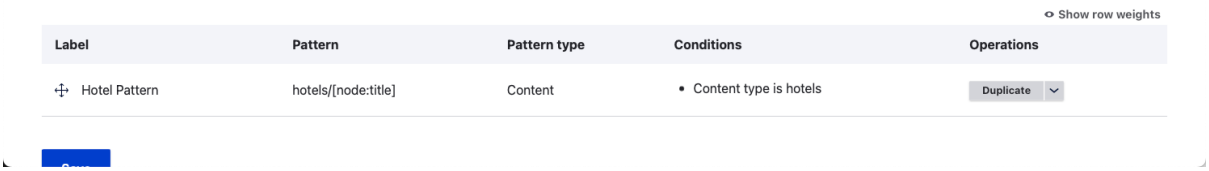


7. Scroll down to find **Title [node:title]** and click the **[node:title]** link.
8. Close the **Available tokens** dialogue box. The **Path pattern** field should have the following string now: **hotels/[node:title]**



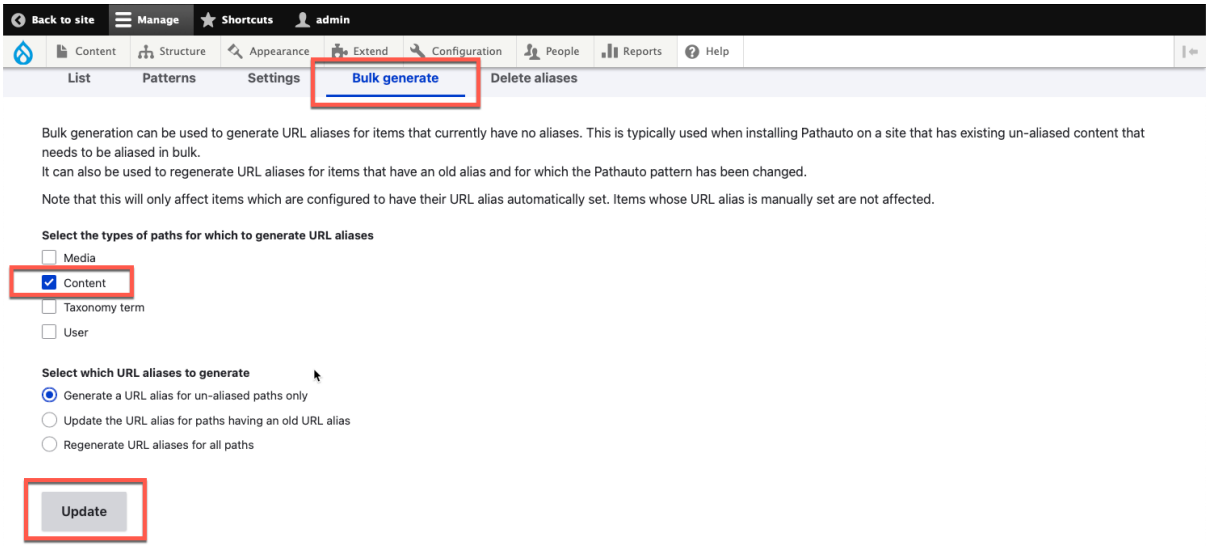
9. Check **Hotel** under *Content type*, add **Hotel Pattern** in the *Label field* to this pattern.
10. Leave the remaining defaults and click **Save**.

    You will be redirected to the *Patterns* page and your newly defined pattern will be listed on the page.

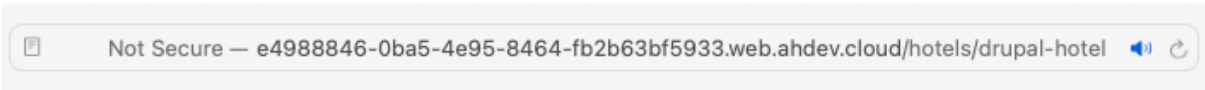


Next, we will apply this pattern to all the existing nodes of Hotel content type.

11. Go to **Bulk Generate tab**, select **Content** and click **Update**.



12. Once the progress bar closes out, go to the **Content** tab and click on a hotel to go to that hotel page/node.
13. Observe: The URL of the hotel node is in the newly defined pattern we just created.

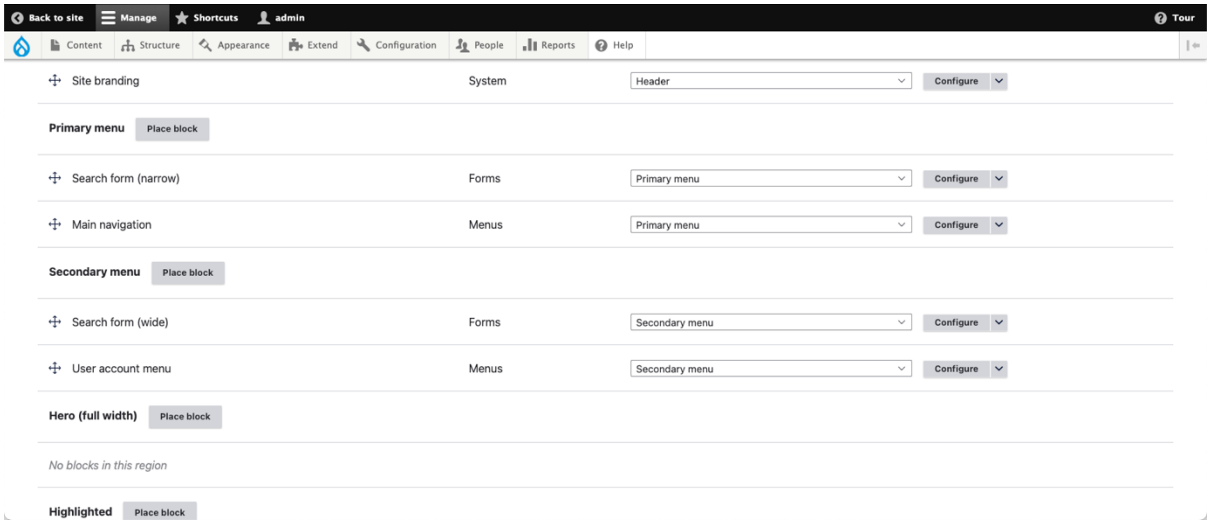


Note: URL patterns can be defined for Content, Taxonomy terms and User profiles. You may define unique patterns per content type or taxonomy vocabularies.

4.6 The Menu System

In Drupal, menus are essentially collections of navigational links that are available as blocks. Some menus may also be hard-coded on the page depending on the theme you're using. To edit the links in existing menus, go to **Structure** > **Menus** and click the **Edit menu button** for the menu you want to manage.

When a menu is not hard-coded in the theme, you'll need to select how it's displayed by placing the menu block for it to be visible to users. Go to **Structure** > **Block layout** and click the **Place block button** next to the region where you'll display your menu. Then choose the menu block you want from the list.



Like any block, you can also configure the visibility settings for your menu.

Menu Permissions and Visibility

You can set permissions to view individual menus and change their visibility by going to **Structure** > **Block layout** and click the **Configure button** for the menu block you want to change. At the bottom of the Configure block page you'll see Visibility settings.

Default Menus

Drupal has several default menus:

- Administration
- Footer
- Main navigation
- Tools
- User account menu

The Administration menu controls all the links that appear in the black Administrative toolbar. It's not recommended to alter this menu unless you are very sure of what you're doing, as it will affect the Administrative toolbar for everyone.

The Footer menu is intended for site information links that you might display in the footer region of your page. By default, this menu just has a single Contact link, but you can add additional links that will list nicely in the footer.

Main navigation is the primary menu you'll be modifying and using as you make your site. It starts out containing one link: Home. The links in this menu are intended to represent sections of your website. If you are using the default Olivero theme, additional links added to the Main navigation will appear as new tabs to the right of the Home button. If all new links are treated equally, Drupal will organize them alphabetically and some tabs may appear to the right of Home. You can adjust the weight of menu items to change this forced order.

Tools is a menu that supplements the main menu. It is intended for user tool links. Certain modules will automatically add links to the Tools menu without consulting you, but most modules are disabled by default.

The User account menu is the menu that controls user-related links for reaching your user profile and logging out.

You have complete control over all of these menus and can add/remove or enable/disable links as needed by clicking the Edit menu button. When disabling or removing links, exercise caution and test your changes, as removing the wrong link can cause difficulties for other people navigating your site.

Menus Using Specific Styles From A Theme

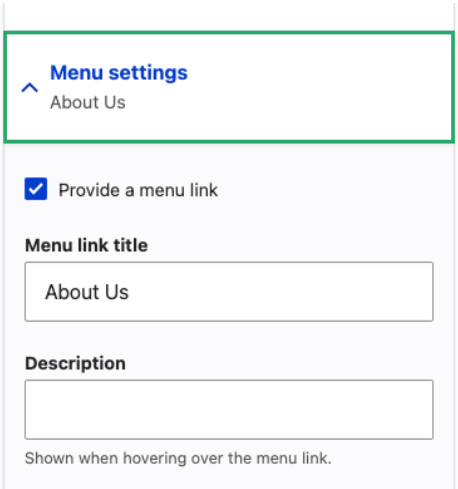
Often themes will give the Main navigation menu a custom style (such as the tabs you see at the top of the page when using the Drupal default Olivero theme, or drop downs when using other themes). Be aware that changing your theme can have a big impact on how your main menu will appear.

Building Custom Menus

You might have noticed that a lot of sites have more than one menu. They might have a main menu at the top, then a menu with links to company information at the bottom. Additional menus are relatively easy to create in Drupal. Simply go to **Structure** > **Menus** and click the **+ Add menu button** to name and describe your new menu. A block will automatically be available so you can place your new menu in a region for display.

Adding a Link To a Menu While Creating Content

When you add new content, you are also able to directly link the new node to a menu at the same time. Look for the Menu Settings tab in the right column of your content creation form.

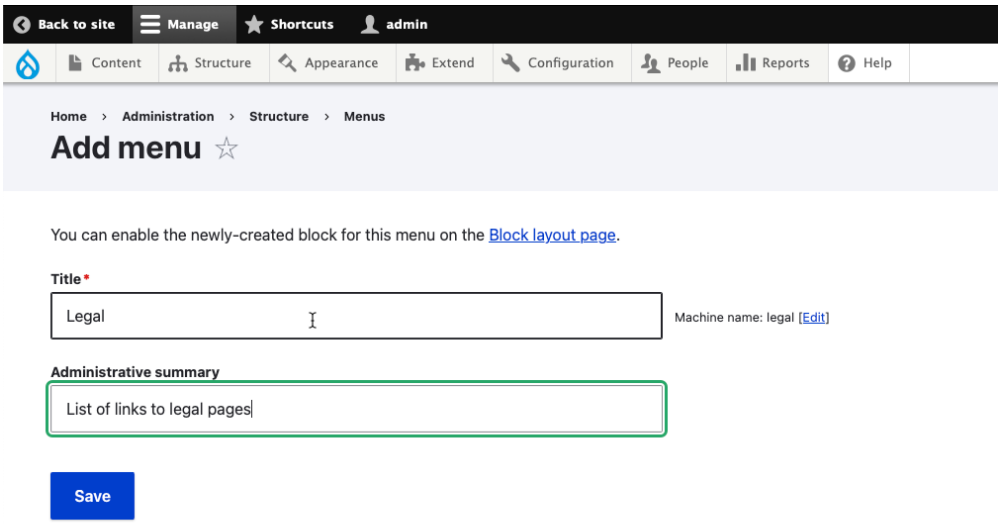


The Parent item drop-down is where you'll choose which menu, or child of a menu, under which you want your new content to appear as a link.

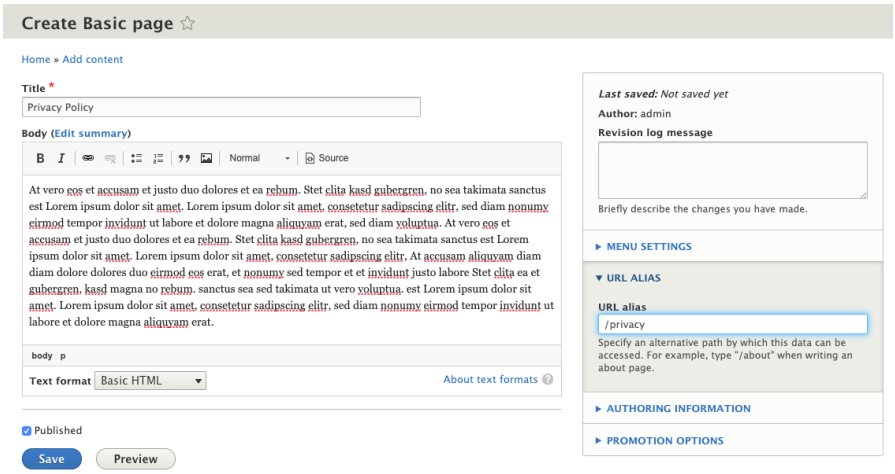
4.6.1 Exercise: Add a Custom Menu

Steps:

1. Go to Structure > Menus > Add Menu
2. Enter the values:
   Title : **Legal**
   Administrative summary : **List of links to legal pages**



3. Click Save
4. Go to Content > Add content > Basic page
5. Enter
6. *Title*: Terms of use
7. *Body*: Add some Lorem Ipsum.
8. On the right drawer, expand URL alias section, and add a URL alias **/terms**

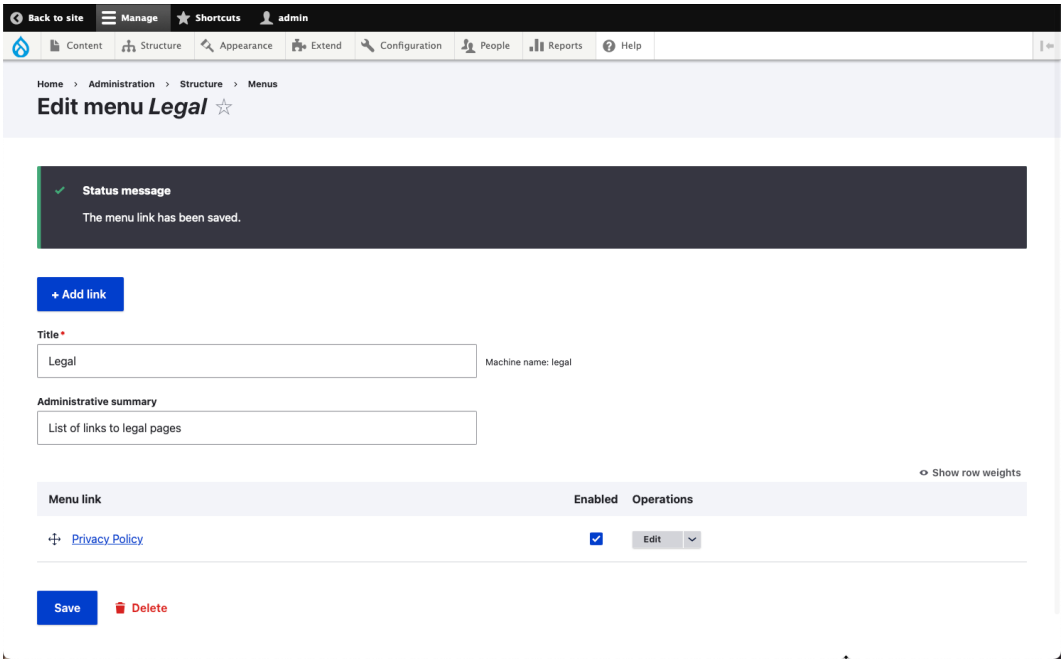


- Click save

6. Repeat the above steps to one more Basic page with *Title* : **Privacy Policy** and URL Alias : **/privacy**
7. Go to Structure > Menus > Legal > Add link
8. Enter
    8.1. Menu link title: **Terms of use**
    8.2. Link: **/terms**
    8.3. Description: Terms of use of this website

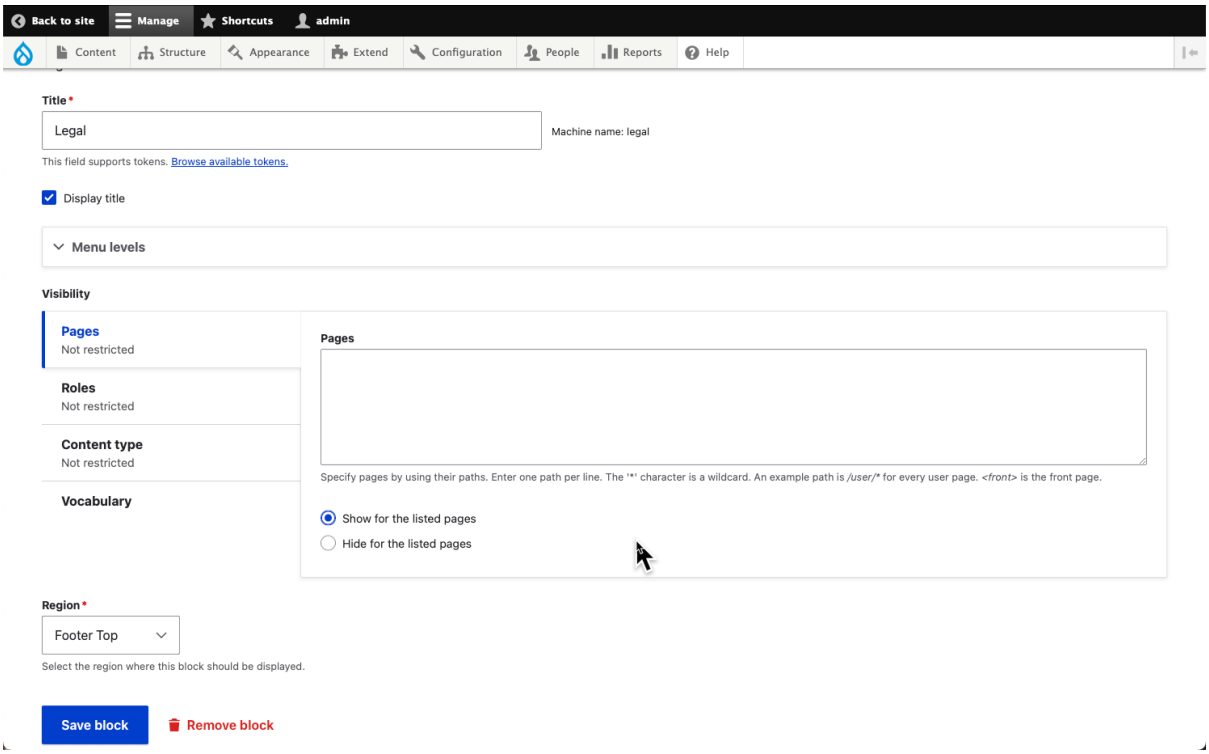
8.4.Parent link: keep <Legal> as selected value

9. Click Save. The same form will reload, with a success message at the top.
10. Repeat the above two steps to add a menu link for **Privacy Policy**.
11. Go to Structure > Menus > Legal to verify if the links are added.

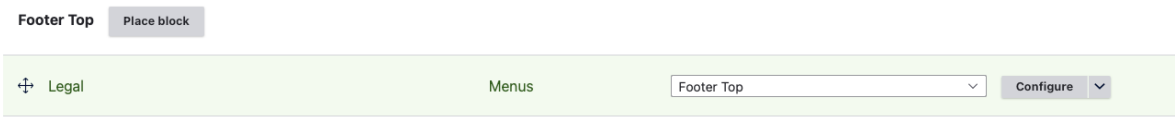


In order to display this menu on the site, you need to use Block layouts page to place the block of the menu in a region on the site.

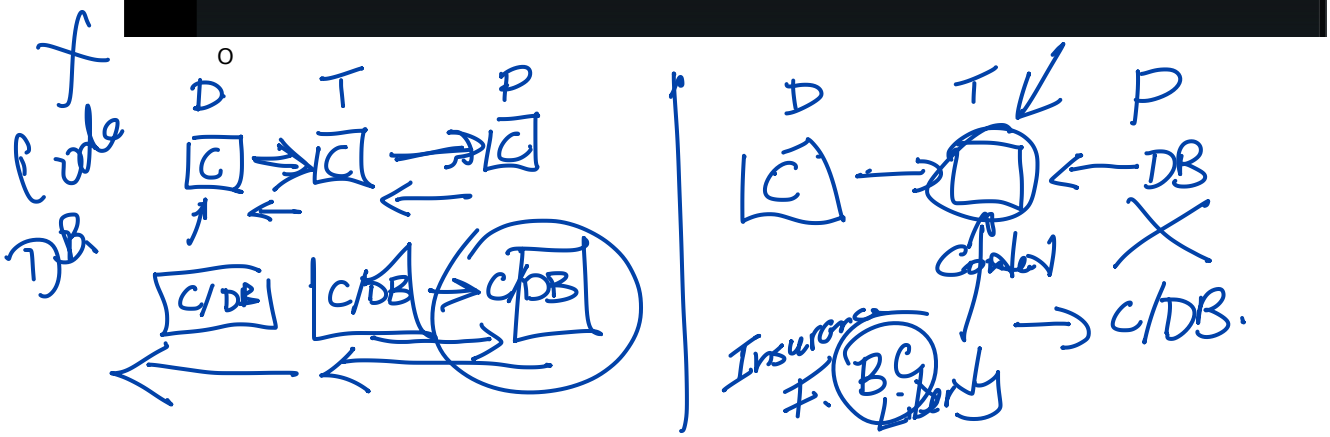
12. Go to Structure > Block layout, scroll to **Footer Top** region and click **Place block** button.
13. In the Place block pop-up window, search Legal, and Click Place block button.
14. In the following page, leave all the defaults and clock Save block.



15. You should now see the Legal menu block in the Footer first section on Blocks overview page.
16. Verify the block placement by going back to the site and checking the Footer first region.







4.7 Configuration Synchronization

Configuration management is the ability to define the configuration of a software application like a Drupal website in a testable, "versionable" way. In previous versions of Drupal, both content and configuration were saved to the database making it very tricky to push configuration changes to a live site. In Drupal, all configuration changes run through a unified configuration API. Drupal includes a Configuration Manager module in core that provides a basic UI to handle both configuration imports and exports. This means updates can be safely transferred between dev and live environments with predictable results.

Content Entities	Config Entities
• Can customize fields	• Can deploy to different environments
• Stored in database tables (by default)	• Stored in configuration system
• Mostly created on front-end	• Mostly created on back-end
Examples	**Examples**
Nodes	Content Types
Custom Blocks	Custom Block Types
Users	User Roles
Comments	Views
Taxonomy Terms	Taxonomy Vocabularies
Menu Links	Menus
Aggregator Feeds/Items	Image Styles

The Configuration Management Initiative in Drupal has developed a standardized way for modules to define their editable configuration. Site builders can then export the configuration for an environment into configuration files that can be put into the website's version control system and changed on a per-environment basis. This allows configuration to be audited, rolled back, and testable.

For Further Study
For more information about Drupal configuration management system, visit:
https://www.drupal.org/documentation/administer/config

Learning Module 4.0 Summary

In this Learning Module, you learned:

- Regions are arbitrary because their names and positions are set by the theme.
- Regions provide containers for blocks.
- Blocks are pieces of HTML content (such as User Login or Who's online or a List of Most Recent Blog posts).
- Views make it easy to query and display information for your Drupal database.
- Views Displays allow you to create many different displays, that are each formatted, filtered, and sorted separately, from a single query.
- You can use the Pathauto and Token modules to create custom URL aliases for any entity in Drupal.
- Custom URL aliases make page addresses easier to read for humans, friendlier for search engines, and more useful for site builder rule setting.
- Menus can be created and edited in Drupal under **Structure** > **Menus**.
- Menus need to be placed like blocks before they are visible to users.
- There are several default menus. Be careful when editing these so you don't remove links that will cause hardships for your other site administrators or your users.
- The Main menu is often the menu styled in custom themes, so if you want to take advantage of drop down capabilities or other custom styling you will need to focus on the Main menu.
- Content can be linked to menus directly from the content creation page.
- You can set visibility and permissions for individual menus by going to **Structure** > **Blocks** and clicking on **Configure** for the menu you need to change.

5.0 Building Complex Views

Views is a tool built into Drupal that makes it easy to define queries and specify how the results of your queries are formatted and displayed. In fact, many of the administrative lists you see in Drupal are created in Views. The lists of Content, People, and Files are all Views - as well as the default Front Page display. Knowing how to use Views makes customizing these administrative lists easy.

In this Learning Module you will try some more complex exercises with Views. You will learn about:

- Contextual Filters
- Adding Relationships
- Creating JSON Lists

5.1 Contextual Filters

A contextual filter allows you to filter content based on dynamic information instead of setting a static filter value. Contextual filters provide you with more convenience by fetching values from variables either programmatically or from a URL.

In this exercise, we are going to create a View in Drupal that only displays content created by the logged-in user. We will use a contributed module called Devel Generate that will automatically create content for us. This is a great tool for QA and testing functionality.

5.1.1 Exercise: Devel Generate Hotels and Create Contextual Filter

Steps:

1. Install the Devel module:
    1.1. Go to https://www.drupal.org/project/devel
    1.2. Copy the Composer command and paste into the Terminal to install.
    1.3. Back in your browser, click the link to **Enable newly added modules**
    1.4. Enable **Devel** and **Devel Generate**.
2. Generate Content
    2.1.Go to **Manage** > **Configuration** > **Development** > **Devel Generate** **> Generate content**

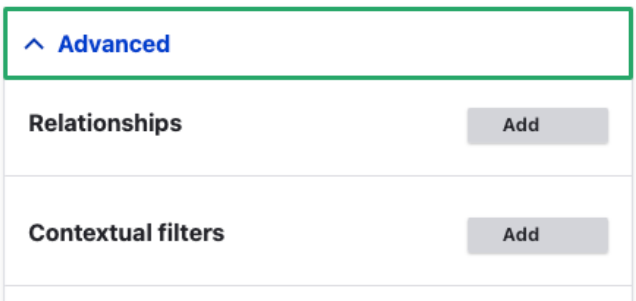
2.2. Select **Hotel**

2.3. How many nodes would you like to generate? Create **20**

2.4. **Change** the number of words in the title to **2**

2.5. Click **Generate**

3. To create your View, go to **Manage** > **Structure** > **Views.**
4. Click +**Add view.**
5. On the *Add view* page,
   5.1. *View name*: **Alternate Hotels**
   5.2. Under *View Settings,* select to *Show:* **Content** of type **Hotel**
   5.3. Click the checkbox to **Create a block**
   5.4. Under *Block Display Settings*, leave *Display format:* **Unformatted list** of **titles (linked)** checked.
   5.6. Click **Save and edit**.
6. Click **ADVANCED** in the right column on the Views editing page. Note: The *ADVANCED* column will expand to show many options.
7. Click **Add** next to *CONTEXTUAL FILTERS*.



8. In the *Add contextual filters* screen, *Search* box, enter **taxonomy**
9. Select the checkbox next to the field titled **Has taxonomy term ID**
10. Click **Add and configure contextual filters**.
11. Locate *WHEN THE FILTER VALUE IS NOT AVAILABLE.*
12. Select **Provide default value**, then select **Taxonomy Term ID from URL** from the *Type* dropdown.
13. Deselect the checkbox for **Load default filter from term page**
14. Check the box for **Load default filter from node page, that's good for related taxonomy blocks**
15. Under *Multiple-value handling* select the radio for **Filter to items that share any term**
16. Scroll down to locate WHEN THE FILTER VALUE IS AVAILABLE OR A DEFAULT IS PROVIDED
17. Select **Specify validation criteria**

18. Under *Validator* select **Taxonomy Term** from the drop-down.
19. Under *Vocabulary* select **Landmark**
20. Click **Apply**.
21. Click **Save** on the Views editing page.
22. Go to the **Manage** > **Structure** > **Block layout**.
23. Scroll to the *Sidebar First* region and click **Place block**
24. Enter **Alternate** in the *Filter by block name* box
25. Click **Place block** to the right of *Alternate Hotels*
26. On the *Configure block* page, scroll to the section titled *Visibility* and click **Content types**
27. Click the checkbox to only display this block with **Hotel**
28. Click **Save block**
29. Click **< Back to site**
30. Observe: Visit a hotel page, you should see a block titled "Alternate Hotels" that displays a list of Hotel names that have the same Landmark tag as the Hotel you're viewing.

Pro Tip: A shortcut in this exercise is to put a checkmark in "Limit terms by vocabulary" in the Default Value section. You would select "Landmark" and the second part of this process would be unnecessary.

Also, You can control whether the Hotel name links to content or not. To change the way a field is rendered in a Views display, click on the field name (Title) under FIELDS and edit the display.

Exercise 5.1.2: Remove Current Node from Query Result with Contextual Filter

Steps:

1. Return to Edit your Alternate Hotels view
2. Click **ADVANCED** in the right column on the Views editing page.
    2.1. Note: The *ADVANCED* column will expand to show many options.
3. Click **Add** next to *CONTEXTUAL FILTERS*.
4. In the *Add contextual filters search* box, enter **id**
5. Select the checkbox next to the field titled **ID** from the category **Content** (this is the node ID).
6. Click **Add and configure contextual filters**.
7. Locate **When the filter value is not available**

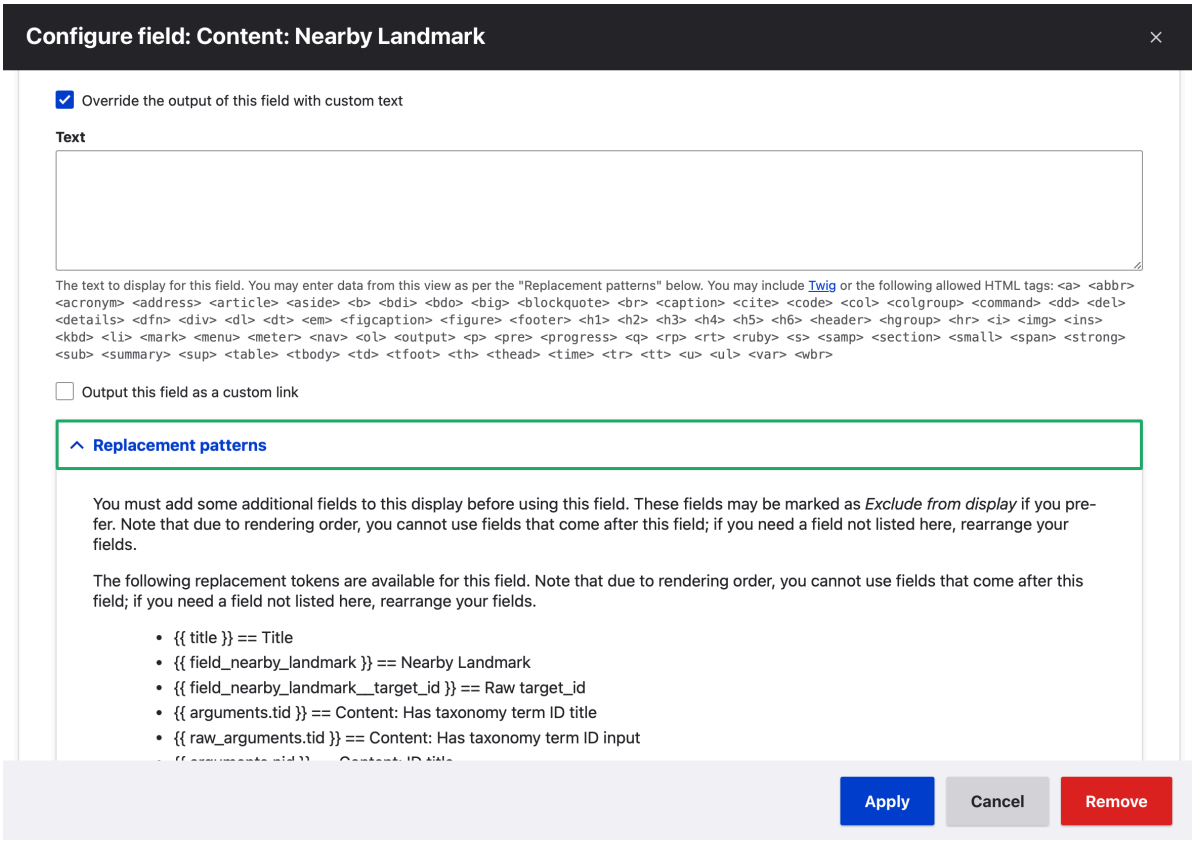
8. Select **Provide default value**, then select **Content ID from URL** from the *Type* dropdown.
9. Scroll down to locate *MORE* and click to expand options
10. Select the checkbox for **Exclude**
11. Click **Apply**.
12. Click **Save** on the Views editing page.
13. Observe: Visit a hotel page, you should see that the current hotel is not included in the query results.

Exercise 5.1.3: Add the Name of the Landmark in the Title of the Block

One of the features in Views is the ability to create replacement values for fields in a View. In this exercise we're going to add the name of the Landmark in the title of the Alternate Hotels view.

Steps:

1. Hover the Alternate Hotels view block and click **Edit View**
2. In the Fields area, click **Add Field**
3. Filter by "near", select Nearby Landmark and click **Add and Configure Fields**
4. Put a checkmark in **Exclude from display**
5. Click on **Rewrite Results**
6. Put a checkmark in the **Override the output of this field with custom text** box.
7. Click on **Replacement Patterns** and copy the text {{ field_nearby_landmark }}. This is Twig syntax (We cover this in our Layout and Theming class).
8. Click **Apply**
9. In the Title area (back at the top), click on the **Alternate Hotels** title and enter: **Other hotels near {{ field_nearby_landmark }}**
10. Finally, click on **Content:Nearby Landmark** and uncheck **Link to the Content** and click **Apply**
11. Click **Save**



5.2 Relationships

When you first create a View, you select your base table. That table might be content, user, or taxonomy. This means our View is querying data from the specific table we selected, which excludes all other tables.

Sometimes you need to access data associated with your base table in other tables. Views gives you a way to do this called Relationships. If you're familiar at all with querying a database, this is similar to creating a join to associate two tables with each other based on a common identifier.

Site-Building Scenario

You will create a new View that consolidates a series of basic pages that contain questions and answers into a list of Frequently Asked Questions. Because you want to show the face and expertise of the staff who are answering the questions, you will also establish a Views Relationship with the user profile pages of the people answering each question. This will allow you to grab user photos for the user table to display with each FAQ.

5.2.1 Exercise: Create a New FAQ Content Type

Steps:

1. Go to **Manage** > **Structure** > **Content types > Add content type.**
2. Configure the new Content type with the following:
    2.1. Name : **FAQ**
    2.2. Title : **Question**
    2.3. Reuse the Body field and label it: **Answer**
3. Click **+ Create new field**
4. Choose the **Reference** field type and click **Continue**
5. Enter Label: **Customer Representative**
6. Select User and click **Save and continue**.
7. Enter Help text: Start typing in the name of the customer representative answering the question.
8. Click **Save settings.**
9. Go to **Content > + Add Content > FAQ** to create new content
10. Enter
    10.1. Question: **Test Question**
    10.2.Answer: **Add some Lorem Ipsum.**
    10.3. Customer Representative: start typing username admin and select the username when it appears.
    10.4.Click **Save**.
11. Observe: When Customer Representative is displayed on the FAQ page, you can click on the link to visit the User's profile.

5.2.2 Exercise: Create a Frequently Asked List of Questions using Views

Steps:

1. Go to **Manage** > **Structure** > **Views** > **Add view.**
2. Configure the new View with the following:
    2.1.*View name*: **Frequently Asked Questions**
    2.2.*View settings*: Show **Content** of type **FAQ**
3. Click the checkbox to **Create a page.**
4. Change the *Path field* by entering **faqs**.
    Note: This will be the URL for the new page.
5. Under *Page Display Settings*, select *Display format:* **Table.**

6. Click the checkbox to **Create a menu link.**
7. Select **Main navigation** from the *Menu* drop down.
8. Enter **FAQs** for the *Link text.*
9. Click **Save and edit.**
10. Scroll down to the *Preview* area on the next page.
11. Click **Save.**

To make this all work well, we need more Users and FAQ's on our site. Let's use the Devel Generate module to automatically create content for us.

5.2.3 Exercise: Devel Generate Users and FAQs

Steps:

1. Go to **admin** > **Edit profile** to edit your user account profile.
2. Scroll to the **Picture** field and upload a picture for the profile.
3. Click **Save**.
4. Go to **Manage** > **Configuration** > **Generate** > **Generate users**.
5. Change the number from *50* to **10**,
6. Select **Content Editor** and **Administrator**, leave all other defaults.
7. Click **Generate**.
8. Go to **Manage** > **People** to see a list of all your new users.
9. Click on **a user name** to see the generated user name and picture.
10. Go to **Manage** > **Configuration** > **Generate** > **Generate content**.
11. Check the box for **FAQ**
12. Change the number of words in titles to **2**.
13. Click **Generate**.
14. Click <**Back to site**.
15. Click the menu link for **FAQs**.
16. Observe: More *FAQ* titles appear in the table.

In this last exercise for this lesson, you will edit the Frequently Asked Questions view to add a Relationship to the User entity and format the view to display all the important information.

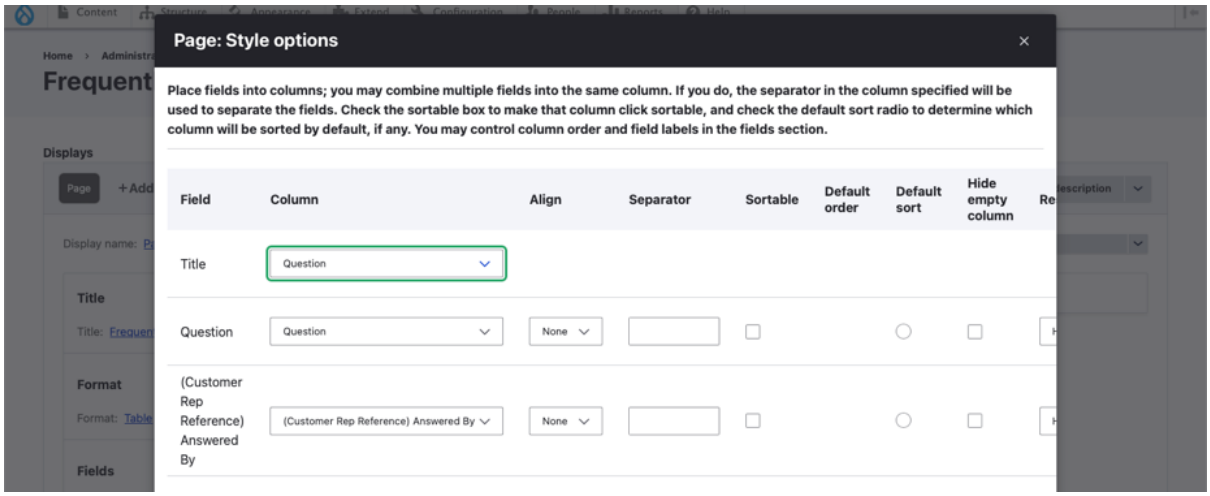
5.2.4 Exercise: User Image in FAQs Content Listing

Steps:

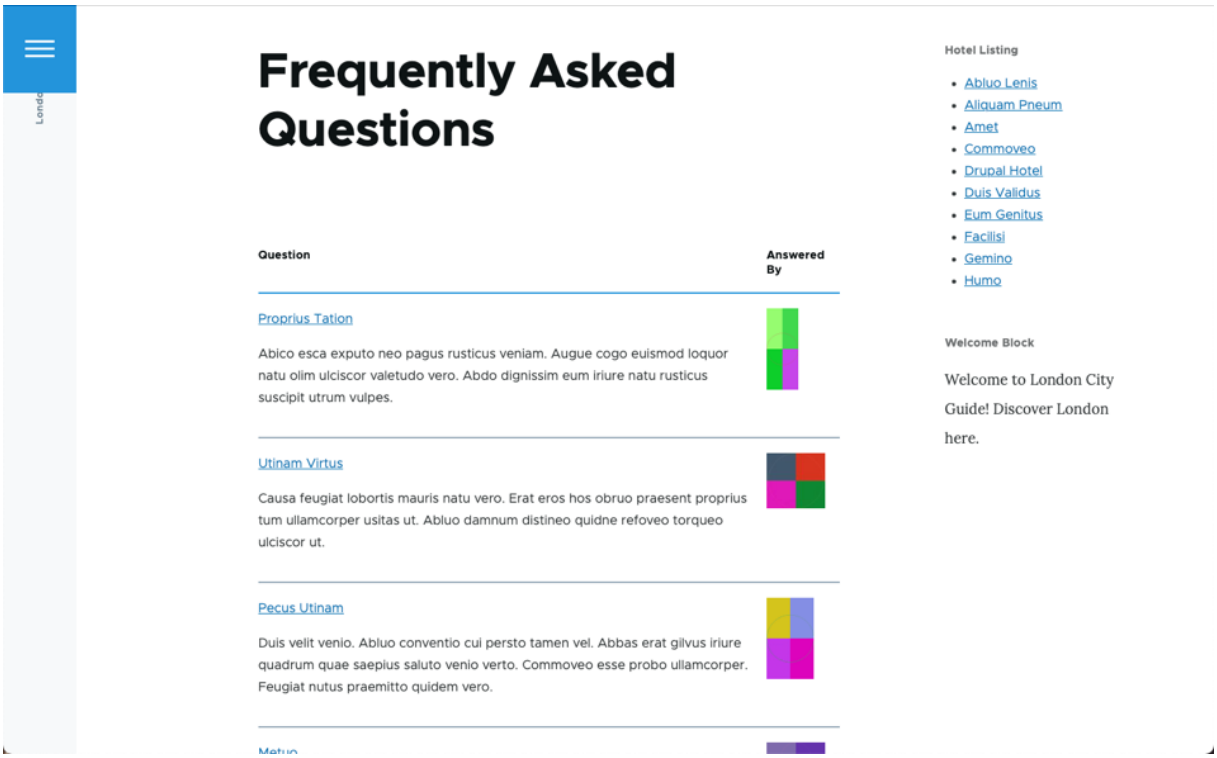
1. Go to **Manage** > **Structure** > **Views.**
2. Click **Edit** in the *Operations* column for the *Frequently Asked Questions* view.
3. Click to expand the **Advanced** column so you can see the advanced editing options.
4. Next to *Relationships*, click **Add**
5. Enter **Customer Representative** in the *Search* box
6. Click the checkbox to select the **User referenced from field_customer_representative** field
7. Click to **Add and configure relationships.** *Note: You'll be brought to a new page to configure your Relationship: field_customer_representative:User.*
8. Click **Administrative Title** to expand the form.
9. For the *Administrative title,* change it to **Customer Rep Reference**.
10. Click the checkbox to **Require this relationship**.
11. Click **Apply**
12. Locate the *Fields* category in the left column.
13. Click **Add**.
14. Type **answer** in the *Search* box and select the **Answer** field.
15. Click **Add and configure fields.**
16. On the *Configure field: Content: Body* page. Set the following:
    - 16.1. Select **Create a label**
    - 16.2. *Label:* **Question**.
    - 16.3. Change *Formatter* to **Trimmed**
    - 16.4. Set *Trimmed limit* to **200**.
    - 16.5. Click **Apply**
17. Click **Add** to add another *Field*.
18. Type **picture** in the *Search* box and select the **Picture** field.
19. Click **Add and configure fields.**
20. On the *Configure field: User: Picture* page. Set the following:
    - 20.1. For the *Relationship* dropdown, select **Customer Rep Reference**.
    - 20.2. *Label:* **Answered by**
    - 20.3. Change the *Image Style* to **thumbnail**.
    - 20.4. Click **Apply**.
21. Find **Format** in the left column and click on **Settings**.
    - 21.1. For both the *Title* and *Question* rows, select **Question** from the drop down options under **Column**.

21.2. Leave the drop down option under **Column** for (field_customer_respresentative: User) *Answered By* field in its own column with the same name.

21.3. Click **Apply**



22. Locate **Filter Criteria** in the left column.
23. Click on **Add**.
24. Find **Picture (user_picture:target_id)** and select it.
25. Click **Add and configure filter criteria**
26. Change the **Operator** to *Is not empty (NOT NULL)*.
    Note: We want to make sure to filter out any Basic Pages that do not have a Customer Representative picture to display.
27. Click **Apply**
28. Scroll down to View the preview area to see a table with FAQ Titles and Questions in one column and Author pictures in the second column.
29. Click **Save** to save your changes.



5.3 Exposing Filter Options to Visitors

The Views Exposed Filter functionality helps users select or filter data on the display. This gives more control to the end users of the site as they can search or filter the results displayed by a View. As a site builder, you may expose any field and make them available to the end user as filters.

In the following example, we are going to expose the Landmark field as a filter field so that end users of the site can filter hotels near a specific landmark of their choice.

5.3.1 Exercise: Allow Visitors to Filter the Hotel Listing

Steps:

1. Go to **Manage** > **Structure** > **Views**.
2. Find the *Hotel Listing* view and click the **Edit** button to the right.
3. Find *Displays* near the top left corner and assure you're editing the *Page* display.
4. In the first column of settings, locate *Filter Criteria* and click the **Add** button
5. Click on *All displays* and change it to **This page (override)**.
6. Enter **landmark** in the Search box and select the **Nearby Landmark** field.
7. Click **Apply (this display)**.
8. On the next setting page, choose *Selection type:* **Dropdown** and click **Apply and continue**.

9. On the *Configure filter criterion* page, check the box for **Expose this filter to visitors, to allow them to change it**
10. *Label:* **Filter Hotels by London Landmark**.
11. Click **Apply (this display)**.
12. Click to **Save** your edited view.
13. Observe: Return to the site and visit the **Hotel Listing** page. You will now see the ability to filter the list by *Nearby Landmark*.

5.4 Creating a View That Returns a List of Content in JSON

JSON (JavaScript Object Notation), is a way to store information in an organized, easy-to-access manner. It gives us a human-readable collection of data that we can access in a logical manner.

With Drupal you can now output JSON and XML versions of data using the core Serialization API. There are contributed modules available to add new serialization formats, such as core's HAL (Hypertext Application Language).

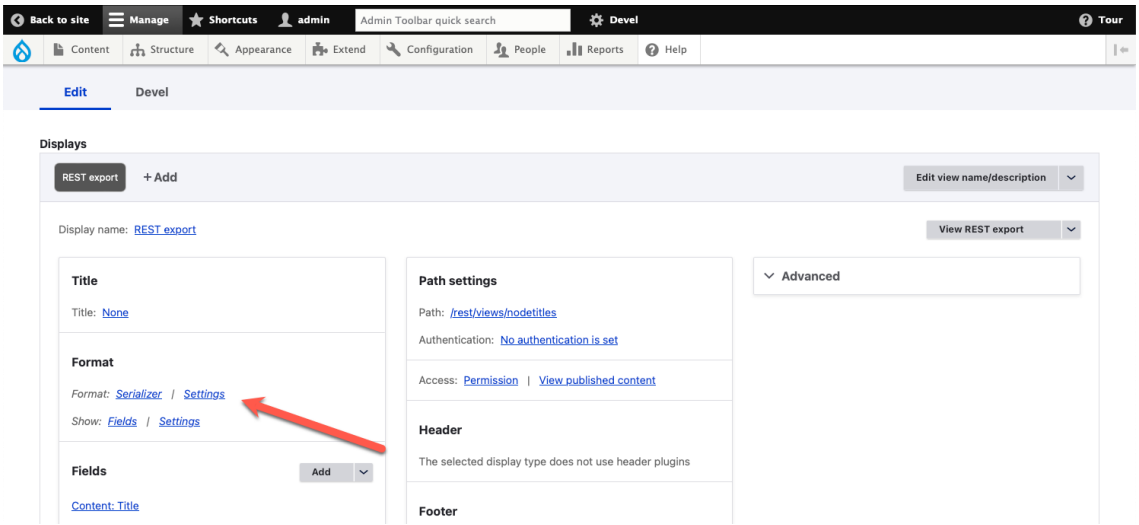
In this exercise, we will create a view that returns a list of content in *JSON* via the *REST API*.

5.4.1 Exercise: Creating a View That Returns a List of Content in JSON

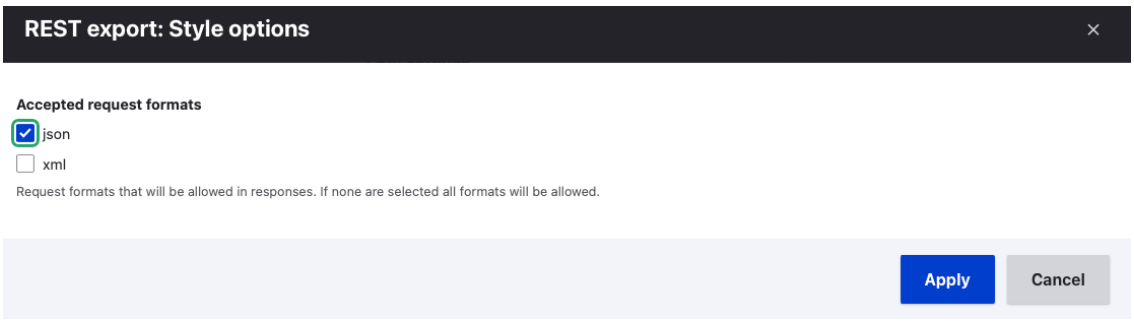
Steps:

1. Go to **Manage > Extend.**
2. Install both the **RESTful Web Services** and **Serialization** modules.
3. Go to **Manage** > **Structure** > **Views** > **Add view**.
4. Name the view **Node Titles**.
5. Select **Provide a REST export** under *Rest Export Settings*.
6. Add a path for your feed **rest/views/nodetitles**.
7. Click **Save and edit.**
   Note: REST export lets you return whole Entities, Fields or Search results. To make this selection

8. Click on the word **Settings** under ***Format: Format***



9. Select **json** as **Accepted request format** and click **Apply**



10. Click on the word **Entity** under ***Format: Show***.
11. Select **Fields** and click **Apply (this display)**.
    Note: Entities will return the entire serialized node object.
12. Click **Apply** again.
13. Under *Fields*, click **Content: Title** and deselect **Link to the Content** and click **Apply**
14. Observe: Check the Preview to see you are only returning the *title* field for each node.
15. **Save** the view.
16. Observe: Go to the path **/rest/views/nodetitles** to see something like this:

[{“title:”Test Question”} {"title":"Eu Populus Wisi"},{"title":"Distineo Gemino Sit Sudo"},{"title":"Tation"},{"title":"Blandit"},{"title":"Consectetuer Lucidus Patria"},{"title":"Quae Valde"},{"title":"Exputo Sagaciter Te"},{"title":"Bene Causa Neo Torqueo”},{"title":"Privacy Policy"},{"title":"Terms of use"}]

For more information on "Decoupled" or "Headless" Drupal, visit https://www.drupal.org/docs/develop/decoupled-drupal

Learning Module 5.0 Summary

In this learning module, you learned many things are possible with the Views query tool.

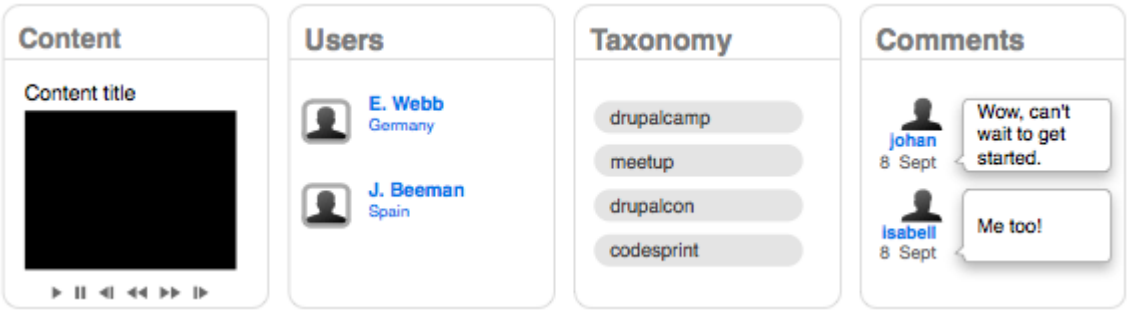
- Contextual Filters allow you to present dynamically filtered lists of data that are dependent on the context where they are displayed.
- Relationships can be made between entities using Reference field types.
- When entities are connected with Reference field types their data can be joined in your Views queries. This allows you to show data from related tables.
- When you enable the core API module's RESTful Web Services Serialization, you can easily create and return content in JSON format.
- REST export Displays in Views have a single output format called Serializer that converts output into the format the client requests. There are 3 formats available: *HAL*, *JSON*, and *XML*. By default all formats can be requested.

6.0 Fielding Entities and Managing Displays

In Drupal, the Entity API has been completely reworked to fill some of Drupal's past functionality gaps and greatly improve the developer experience. All entities are now classed objects that implement a standard Entity Interface.

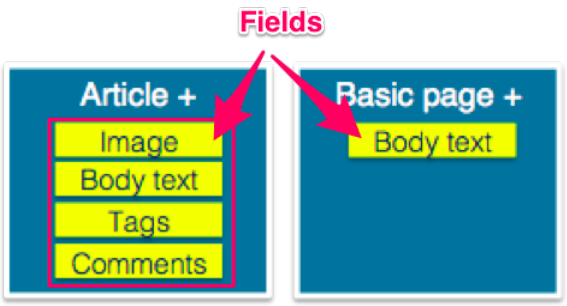
Nearly anything you can create more than one of has been converted to an entity, bringing much greater consistency to Drupal development overall.

In Drupal nearly everything is an entity, including content. Entity types are used to store and display data. Entities can be nodes (content), comments, taxonomy terms, user profiles, or something custom developed.



Drupal allows site builders to specify what data is collected for each entity type. In this learning module we will explore structured data collection for different entities.

With a database-driven website, structured data collection allows us to gain control over the content added to our website. Breaking entries into fields ensures the ability to sort and filter data for later display.



6.1 Review Adding Fields to a Content Type

In a previous module, you had an opportunity to create a new content type and manage the field settings. As a review, let's add another field to the Hotel content type.

6.1.1 Exercise: Fielding Content Types

Steps:

1. Go to **Manage > Structure > Content Types > Hotel > Manage fields**
2. Click the **+ Create a new field** button
3. Select **Link** and click **Continue**
4. *Label:* **Hotel Website**
5. Click **Continue**.
6. Keep default *Field settings set at 1*.
7. Scroll down to where it says *Allowed link type* and select **External links only**. **Pro Tip**: Since Drupal doesn't have a way to open external links in a new tab or window, check out the ExtLink module (https://drupal.org/project/extlink)
8. Click **Save settings**.
9. Edit the **Drupal Hotel** and add the link https://drupalhotel.com and "More information about the Drupal Hotel" in the Link Text field.
10. Click **Save**.

6.2 Adding Fields to User Account Pages

Often you will need to collect other information about those that register with your site, such as their name, phone number, or any other details. Drupal allows you to collect needed information as fields associated with Users. Creating fields for user accounts is a similar experience to creating fields for Content Types. Fields that are marked as required for User accounts are then shown on the User Registration form. Notice how the tabs on the Account Settings page are very similar to the tabs on a Content Type settings page. This consistency of field creation and managing fields is useful and a benefit of using Drupal.

6.3 Adding Fields to Taxonomy Term

In this exercise, we will add an additional field to the Landmark Taxonomy to allow users to easily visit the official website.

6.3.1 Exercise: Fielding Taxonomy Term Pages

Steps:

1. Go to **Manage** > **Structure** > **Taxonomy** > **Landmark > Manage Fields**
2. Click the **+Add field** button.
3. Select **Link** from *Add a new field* drop-down.
4. Enter *Label:* **Official Website**
5. Click **Save and continue**.
6. Keep the defaults on the field settings page.
7. Click **Save field settings.**
8. Click **Save settings** on the next page.
9. Go to **Manage** > **Structure** > **Taxonomy** > **Landmark**
10. Click to **Edit** the Big Ben term
11. Scroll down to the *Official Website* field and enter:
    11.1. *URL:* **http://www.visitlondon.com**
    11.2. *Link text:* **Big Ben Official website**



12. Click **Save**
13. Click **< Back to site**.
14. Click the **Hotel Listing** menu tab.
15. Click on the filter drop down menu and select **Big Ben** to see hotels tagged Big Ben.
16. Click **Apply.**
17. Click on any **Hotel Name** in the resulting list.
18. Scroll to the bottom of the hotel page and click the **Big Ben** term link. Observe: See the newly improved taxonomy term page with *Big Ben's Official Website* link.

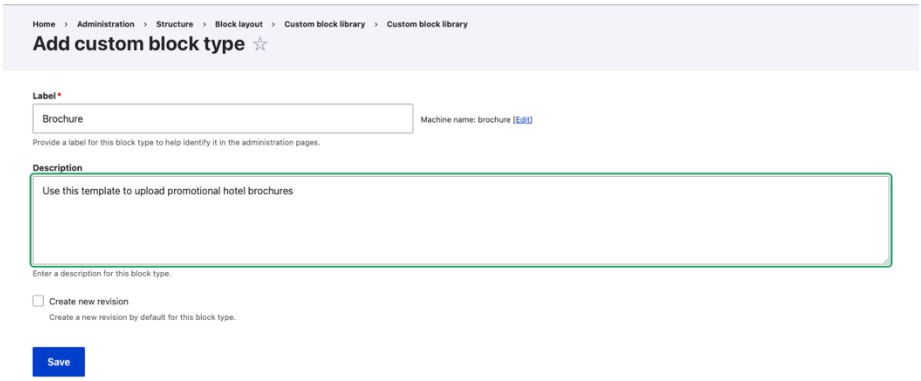
6.4 Adding Fields to Block Types

In Drupal, the functionality of blocks has been broken out into two modules. The Block module simply handles the placement of blocks in regions. The Custom Block module implements a new entity which allows you to create block types to hold block content. You can create different block types and they're fieldable, meaning that you can add user-defined fields to them.

6.4.1 Exercise: Create Custom Block Type for Hotel Brochure Upload

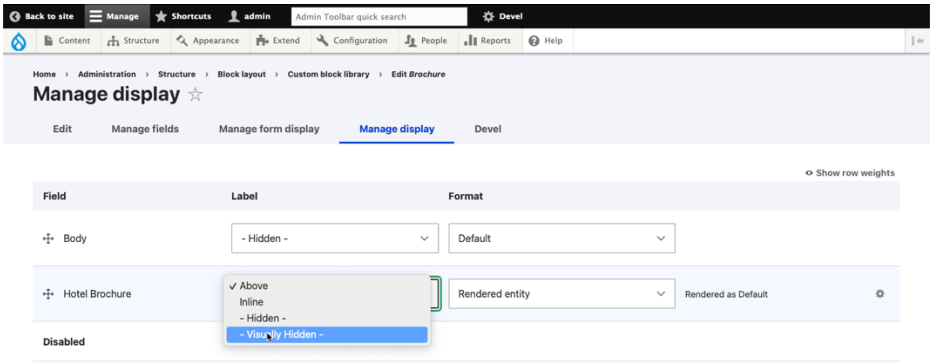
Steps:

1. Go to **Structure** > **Block layout** > **Custom block library.**
2. Click on the **Block types** link.
3. Click on **Add custom block type**
    3.1. *Label:* **Brochure**
    3.2. *Description* field: **Use this template to upload promotional hotel brochures**
4. Click **Save**.



5. Observe: There is now a *BLOCK TYPE*, like when we created a *content type*. *Note: Block types are like content types. We can add fields.*
6. Click on **Manage fields** for the *Brochure* block type.
7. Click + **Create new field**.
8. Select **Media** and click **Continue**
9. *Label:* **Hotel Brochure**.
10. Click **Continue**.
11. Leave **Allowed number of values** set to 1.
12. Select **Document** in the Reference Type box.
13. Click **Save settings**.

14. Click on the **Manage Display** tab.
15. Select **– Visually Hidden-** from the *Label* dropdown to hide the *Hotel Brochure* field's label.



16. Click **Save**.
17. Observe: The *Brochure* block type is now ready to be used and we can create our first promotional content.
18. Upon testing, if the form doesn't appear as expected, go back to **Manage Form Display** to adjust form fields.

6.4.2 Exercise: Create Custom Hotel Brochure PDF Block

Steps:

1. Go to **Structure** > **Block layout**
2. Scroll down to Sidebar and click **Place Block**
3. Click on **+ Add Content block**.
4. Click the **Hotel Brochure** link.
5. Create a brochure block.
    5.1. *Block description:* **Drupal Hotel Special**
    5.2. *Body:* **Want to attend Wimbledon? Check out our special promotional brochure!**
6. Click **Add media**
7. In the Add or select Media popup window, click **Choose file** to upload a local PDF file
8. Once the file is uploaded, click Save.
9. Click Insert selected.
10. Observe that the PDF file is now attached to the custom block.
11. Click **Save and Configure**
12. In the **Visibility area** enter for Pages: **/hotel/drupal-hotel**
13. Click **Save block.**
14. Go **Back to site** and visit the **Drupal Hotel** page.

15. Observe the block appears on the *Drupal Hotel* page. Check other pages to see it does not appear elsewhere.

It follows logic, if you can manage fields and essentially "field" all the entities in Drupal. You probably also have access to manage display of those fields.

6.5 Review Display Modes

Drupal provides several options to manage the display of entities. Drupal gives you the ability to display the same entity in different ways. For example, perhaps on your front page you wish to show a "teaser" of each of the five most recent articles. Of course, when you click on a teaser you expect to view the entire article. Drupal handles these use cases by allowing you to define Display Modes for your content. This is useful because you only need to enter the content one time but it can look very different when displayed in different places throughout your site.

Using the Manage Display tab on an entity provides functionality to do the following:

- arrange fields,
- show or hide field labels, and
- set display options for each field.

You may define multiple Display Modes and can have different display settings for each Display Mode.

In the next exercise, we will add an Image field to our Landmark Taxonomy Term pages.

6.5.1 Update the Teaser Display for the Hotel Content Type

Steps:

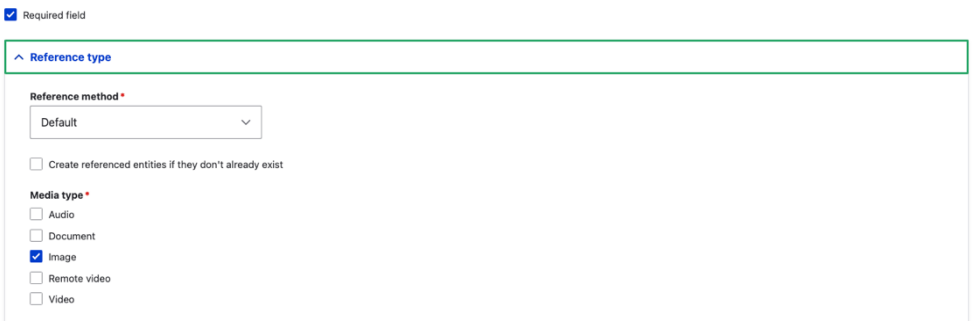
1. Go to **Manage** > **Structure** > **Content Types** > **Hotel** > **Manage Display**
2. Click on **Teaser** in the sub-tabs.
3. Drag **Hotel Image** to the top.
4. Order the fields in the following way: Hotel Image, Hotel Description, Links
5. Change the label setting to Visually Hidden for all **Labels.**
6. Change the Format for Hotel Description to **Trimmed.**
7. Click on the Format gear and change 600 to **400.**
8. Click on the Format gear for the Hotel Image and change it to **Medium 220x220.**

9. Click **Save.**
10. Review the Hotel Listing page.

6.5.2 Exercise: Add a Media Image Field to the Landmark Taxonomy

Steps:

1. Go to **Manage** > **Structure** > **Taxonomy** > **Landmark** > **Manage fields**
2. Click the **+ Create new field** button.
3. Select **Media** and click **Continue**
4. *Label:* **Landmark Image**
5. Click **Save and continue**.
6. In the following page, check the checkboxes for **Required field** and **Image**



7. Click **Save settings**. *(*Note: You should be returned to the Manage fields tab for your Landmark term page.)
8. Click the **Manage form display** tab to improve the form display.
9. Click+hold+drag the four-headed arrow to place the **Landmark Image** field under the *Name* field.
10. Make the label **Visually hidden**
11. Click **Save**.
12. Click on **Manage display**
13. Drag the new media field up to the top and click **Save**

In the next exercise we will add some images to our taxonomy terms to see how they display.

6.5.3 Exercise: Add images to your Taxonomy terms

Steps:

1. Go to **Manage** > **Structure** > **Taxonomy** > **Landmark**

2. Click the **Edit** button for **Big Ben**.
3. Click Add media button to upload or select an existing image.
4. Click **Save**.
5. Follow steps 2-5 to add an image to each of the three remaining landmarks. **Tower Bridge, Oxford Circus,** and **Hyde Park.**
6. Click any of the Landmark term **links**.
7. Observe: The term page now includes an image.

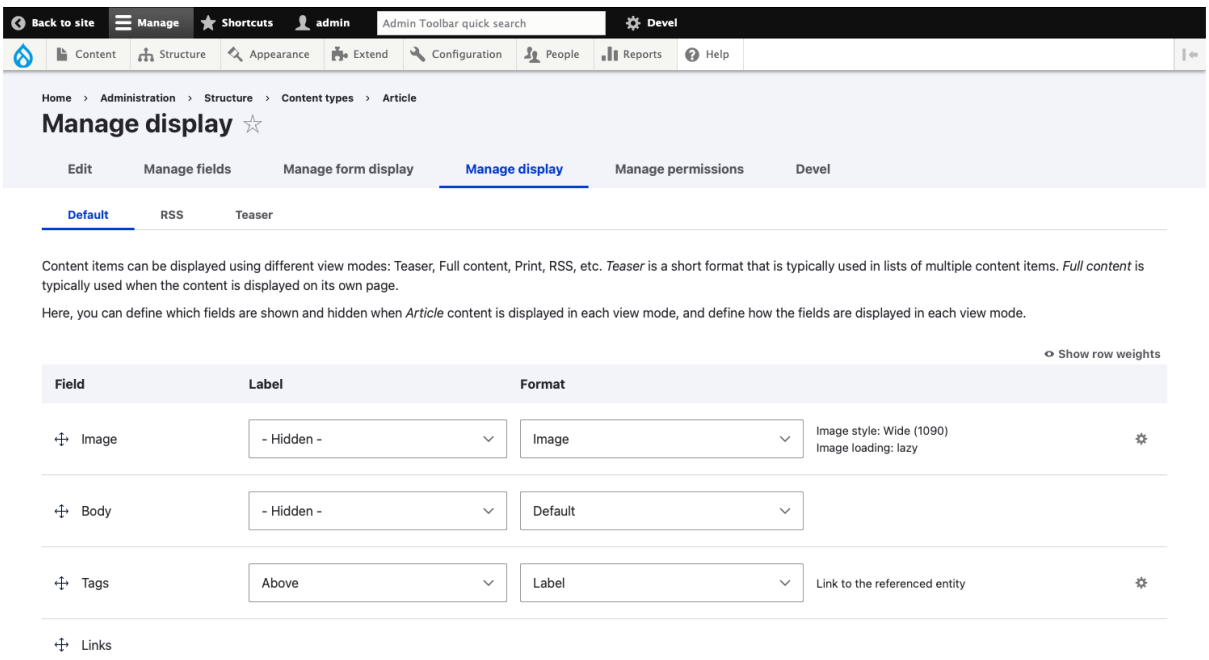
6.6 Images and Display

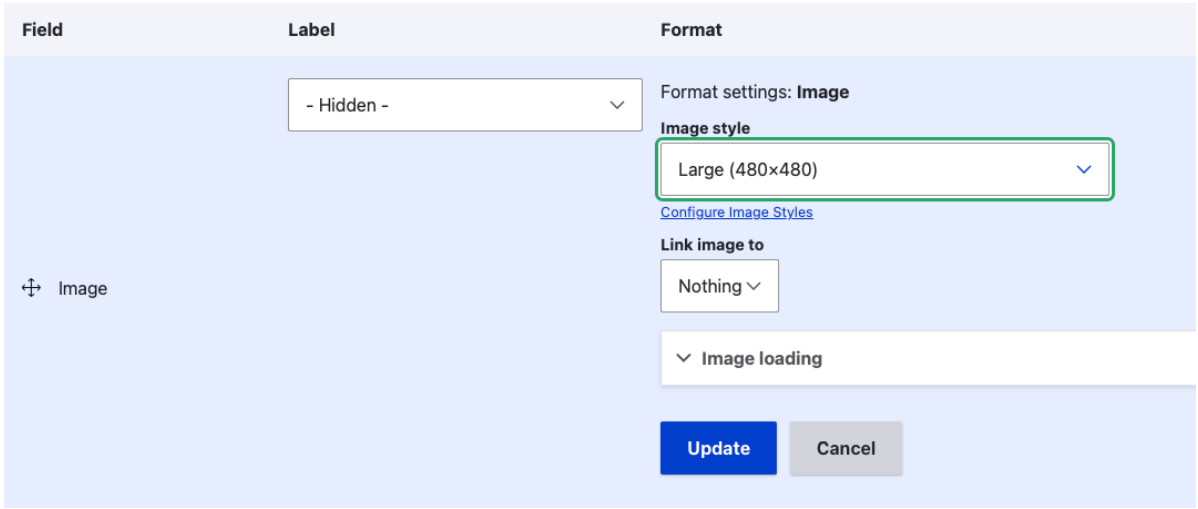
Image Styles

Drupal allows you to create image styles, so you can control how images will appear on your site. It allows you to standardize all image sizes, so you don't have to worry about a huge image breaking your site layout.

Drupal always keeps a copy of the full-size original and will automatically use it to generate images for all styles that are applied. When managing the display, you can click on the image field's gear icon to change the image style.

Go to Structure > Content types > Article > Manage display and click the gear icon next to the Image field to change display settings.





Naming Image Styles

Every image style must have a name, which will be used in the URL of the generated images. Therefore, image style names can only contain lowercase alphanumeric characters, underscores, and hyphens.

Conventionally, image styles are named in one of three ways:

- By a generic style such as "slideshow large" or "thumbnail". This is most flexible and you can reuse it in several locations.
- Where it will be applied, "front-page-slideshow". This provides a scope for a style.
- Size of the image such as "rectangle-350x150".

In the next exercise, you'll name your new style in a way that reflects its size.

Creating a New Image Style

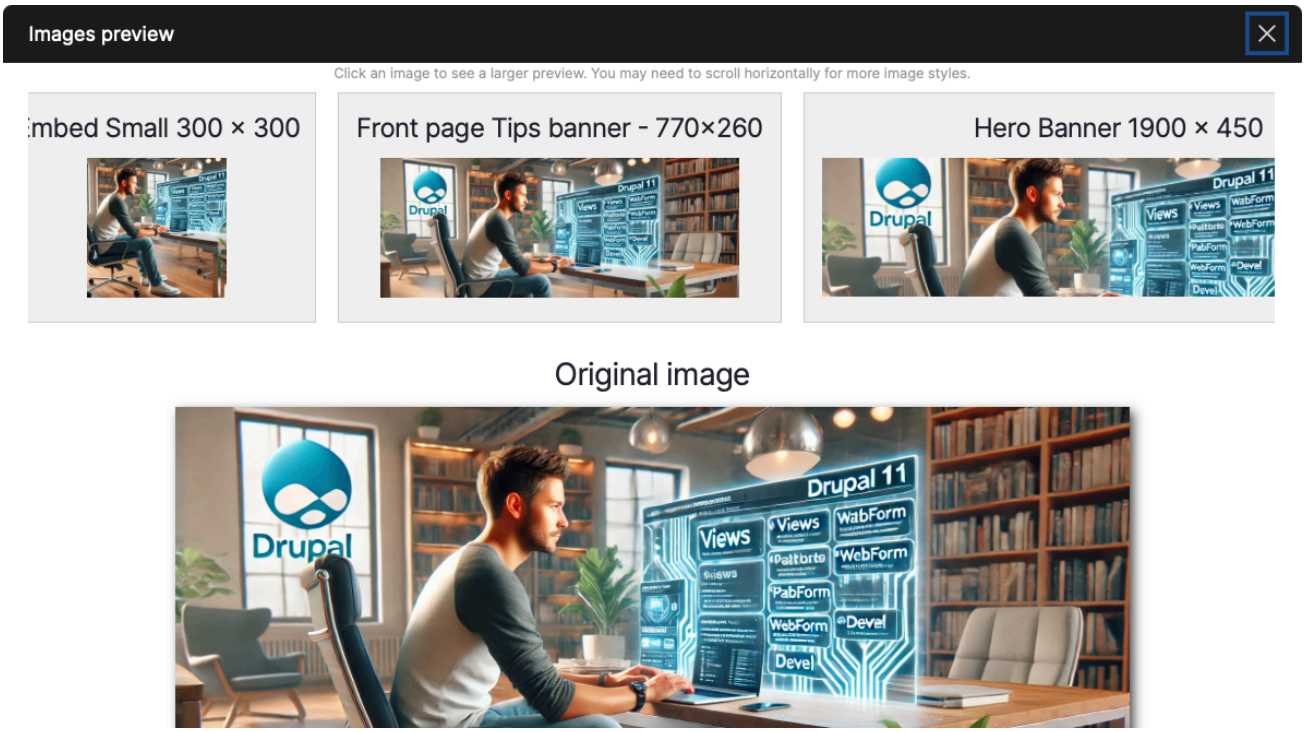
Image styles are configured in the Configuration area of Drupal, so go to Configuration > Media > Image Styles > +Add Image Style to take a look at them.

Earlier you saw the gear icon for the image field. Next, you will create an image style, then use that gear on the manage display field for your content type to configure it.

6.6.1 Exercise: Installing Focal Point and Crop for Better Image Styles

One of the most important models for image management and creating advanced styles is Focal Point (https://drupal.org/project/focal_point). This module will allow you to create many different versions of your media without using Photoshop and precisely scale and crop images for presentation in different places around your site.

We're going to use this module to create styles for our taxonomy landing pages, but you can use it to create as many styles as you need.



Steps:

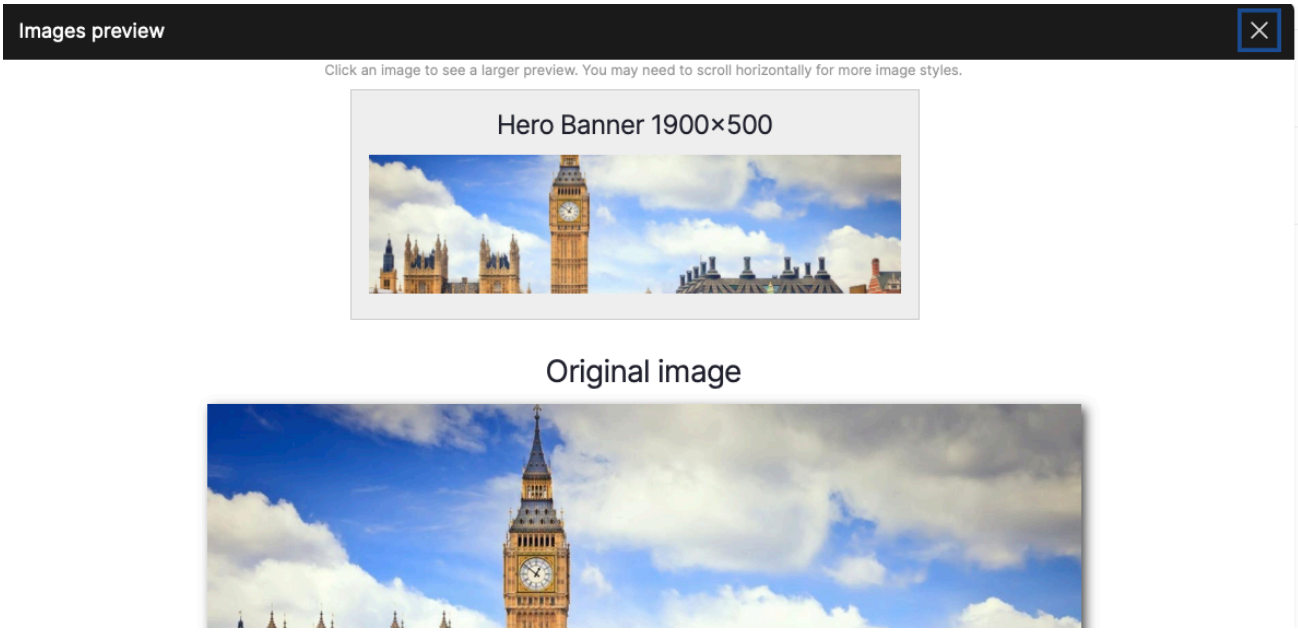
1. Head over to https://drupal.org/project/focal_point**, install and enable the module** via Composer and Drush.
2. Go to **Configuration** > **Media** > **Image style.**
3. Click **+ Add image style**
4. Enter H**ero Banner 1900x500** for the name and **Create New Style**.
5. In the Effect area, click Select a New Effect, **choose Focal Point Scale and Crop** and click **Add.**
6. Enter 1900 for the Width and 500 for the Height. **Click Add effect.**
7. Go to **Structure** > **Taxonomy** > **Landmark** > **Manage display**
8. In the Format column, **change the Format from Rendered Entity to Thumbnail.** This will allow the Media type to use Image Styles
9. Click the **gear** icon on the far right of the **Landmark Image** field.
10. Select **Hero Banner 1900x500** from *Image style* drop-down.
11. Click **Update.**
12. Select **Visually Hidden** from the **Landmark** *Image Label* drop-down.
13. Click **Save.**
14. Go to **Manage** > **Structure** > **Taxonomy** > **Landmark**
15. Click on the **Big Ben** term link.
16. Observe: The new image style effects applied to your term page image.

6.6.2 Exercise: Fixing the "Focus" of the Scale and Crop

Steps:

1. Go to **Structure > Media Types > Image > Edit > Manage Form Display**
2. Change the Widget from Image to **Image (Focal Point)** and click **Save.**
3. Go to **Content > Media.**
4. Click to **Edit** Big Ben London 2 and move the + symbol over the clock face.
5. Click **Preview** to see all the styles (currently only one) that can be assigned to a display view. (adjust until you are satisfied).

One other module that is very helpful for image style is the IEK (Image Effect Kit) - https://www.drupal.org/project/iek. Between these two modules you can achieve just about any effect you require.



6.7 Using Layout Builder

Drupal's Layout Builder offers a single, powerful visual design tool for the following three use cases:

- **Layouts for templated content**. The creation of "layout templates" that will be used to layout all instances of a specific content type (e.g. blog posts, product pages).
- **Customizations to templated layouts**. The ability to override these layout templates on a case-by-case basis (e.g. the ability to override the layout of a standardized product page).
- **Custom pages**. The creation of custom, one-off landing pages not tied to structured content (e.g. a single "About us" page).

The Layout Builder provides the ability to drag and drop site-wide blocks and content fields into regions within a given layout. Additionally, custom "inline" blocks can be created for one-off details specific to a given layout.

Drupal's Layout Builder allows content editors and site builders to easily and quickly create visual layouts for displaying content. Users can customize how content is arranged on a single page, or across types of content, or even create custom landing pages with an easy-to-use drag-and-drop interface.

It's worth looking at the Layout Builder documentation at drupal.org - https://www.drupal.org/docs/8/core/modules/layout-builder - note all the add-on modules available at https://www.drupal.org/docs/8/core/modules/layout-builder/additional-modules!

You need to enable the Layout Builder module for managing layouts to use this feature.

6.7.1 Exercise: Enable Layout Builder Module

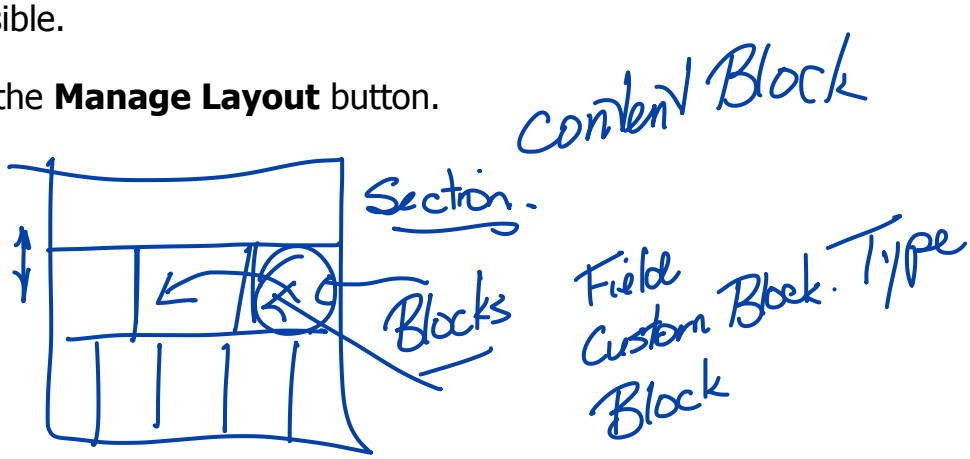
Steps:

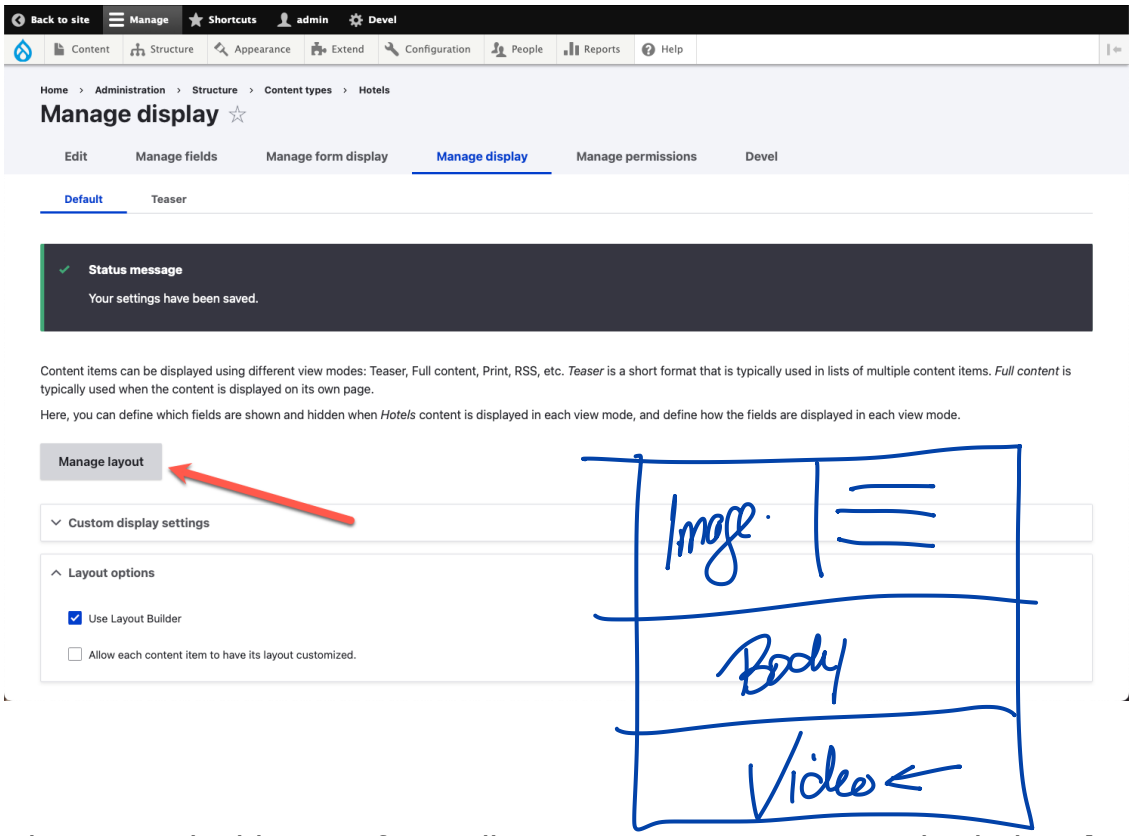
1. Go to **Extend** from the Admin menu
2. In the search field, search for the keyword "*layout*"
3. Enable the **Layout Builder** and **Layout Discovery** modules by clicking on the checkboxes next to these modules
4. Click **Install** to save the configuration.

6.7.2 Exercise: Building a Customized Layout for Hotel Content Type

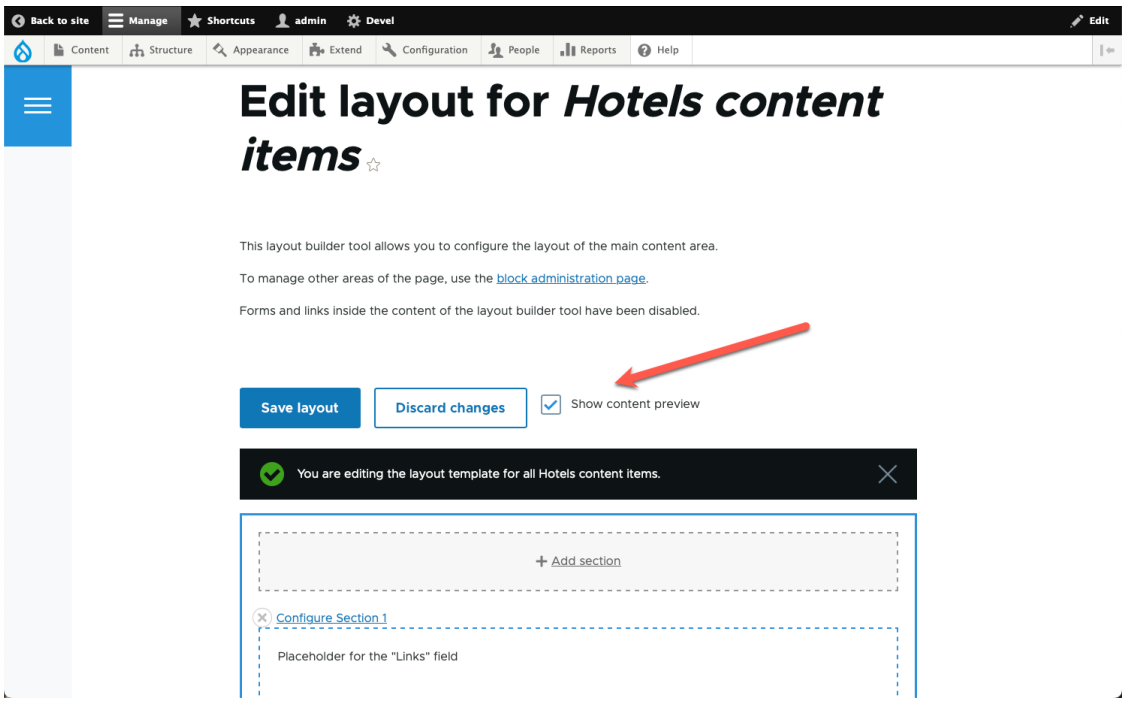
Steps:

1. Go to **Structure > Content types > Hotel > Manage display** from the Admin menu
2. Click **Use Layout Builder** checkbox under *Layout Options* section
3. Click **Save**
4. After saving, the list of fields will be hidden and a **Manage layout** button will be visible.
5. Click the **Manage Layout** button.



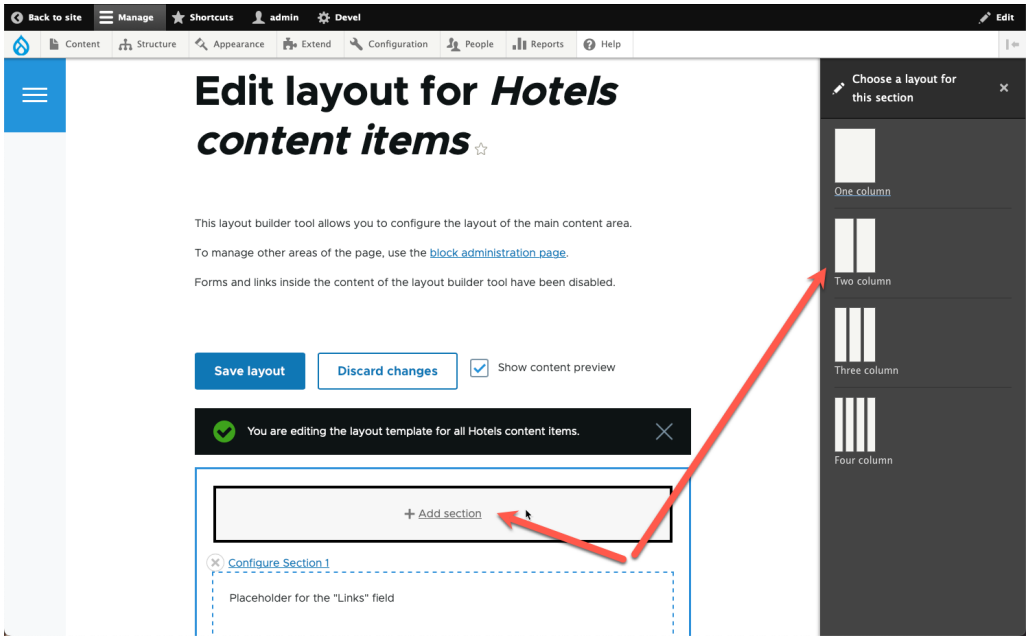


6. The Layout builder interface will appear in a new page. Uncheck the **Show content preview** checkbox to see the list of the fields.



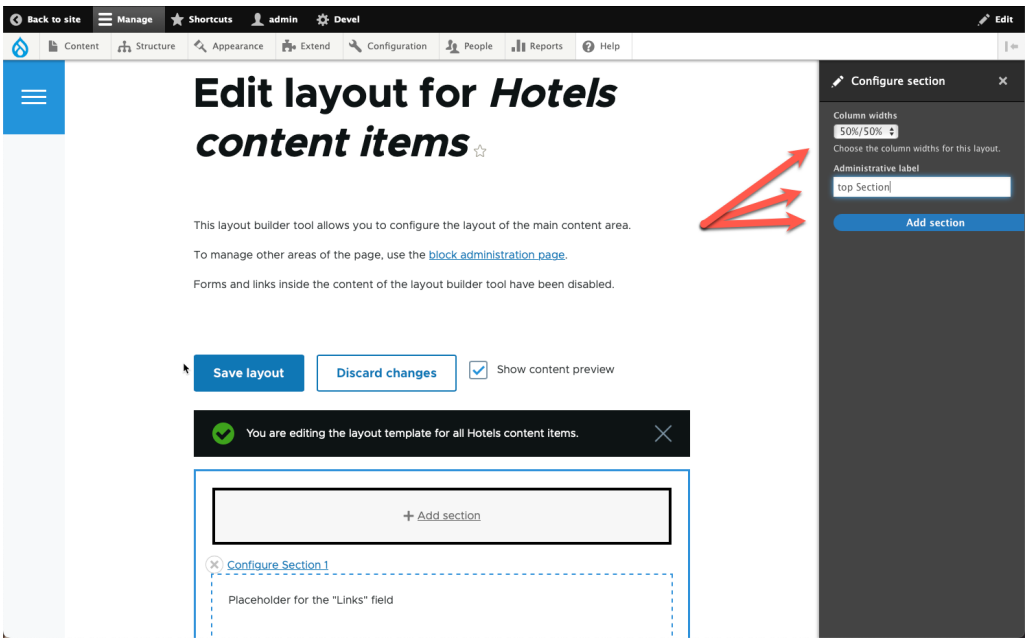
7. The default layout shows all the fields listed one below another in Section 1 of the layout. You can add multiple sections in the layout. Each section can have its own multi-column layout. Each column of the layout can contain one

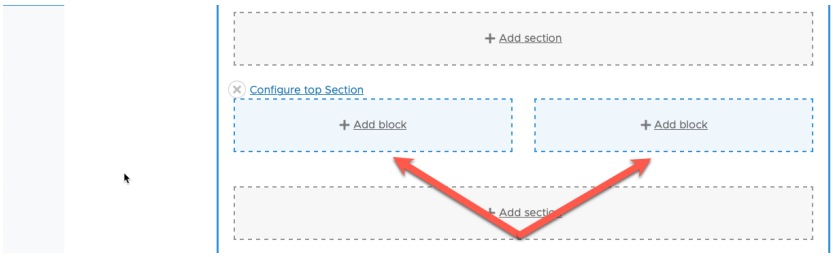
or more fields. To add a new section, click the **Add section** button and select a **Two column** from the *Configure section* side drawer.



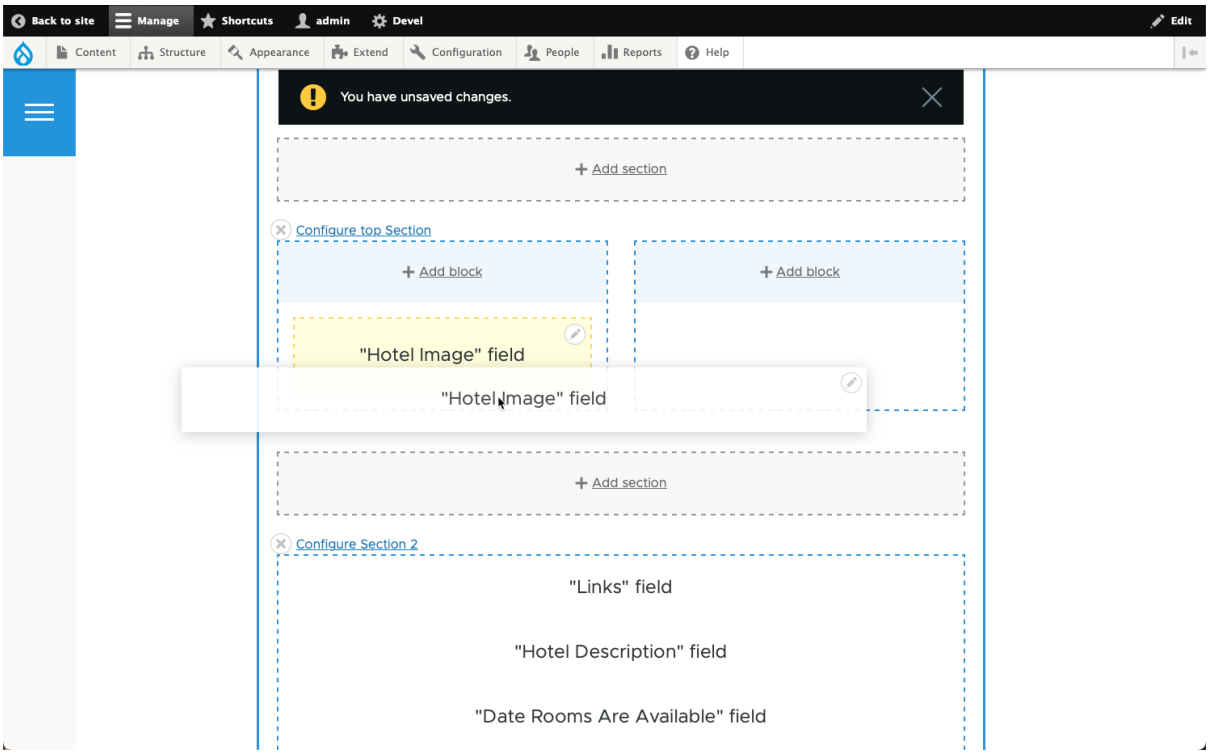
8. To configure the section, give a unique administrative name to the new section (let's use **top-2-col)** and click **Add section**.

   You will see a new section added at the top, with two empty blocks.



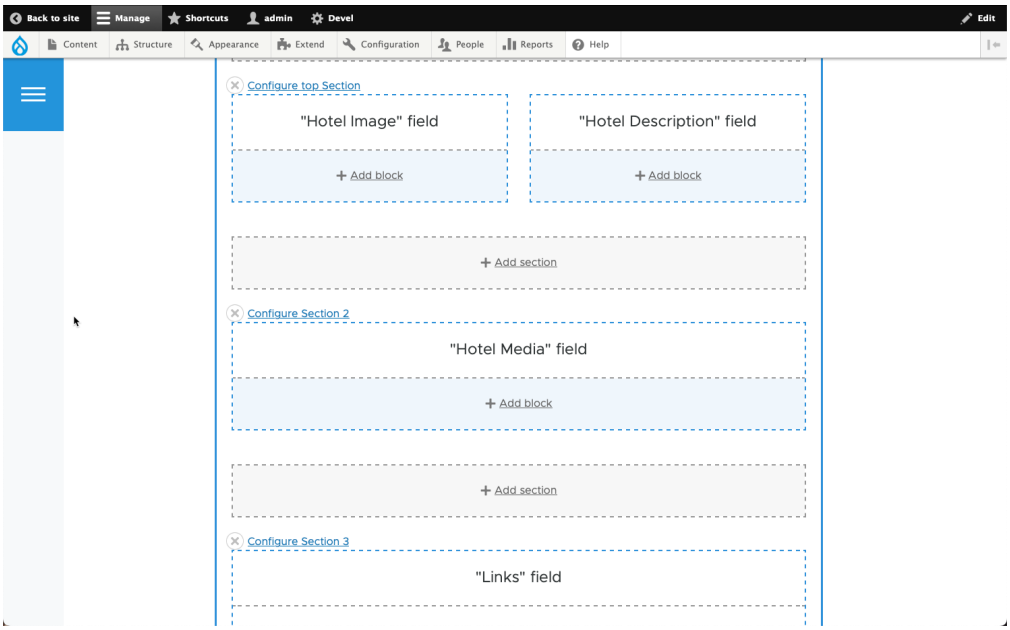


9. Drag and drop the **Hotel Image** field in the top left block and the **Hotel Description** field in the top right block.

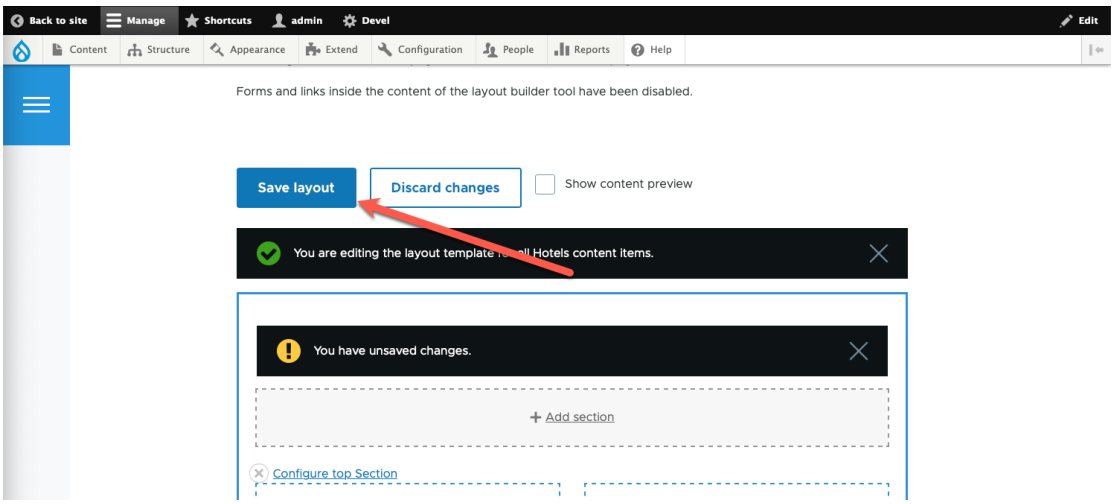


10. Add one more section, with **One Column** layout below the *top-2-col* section.

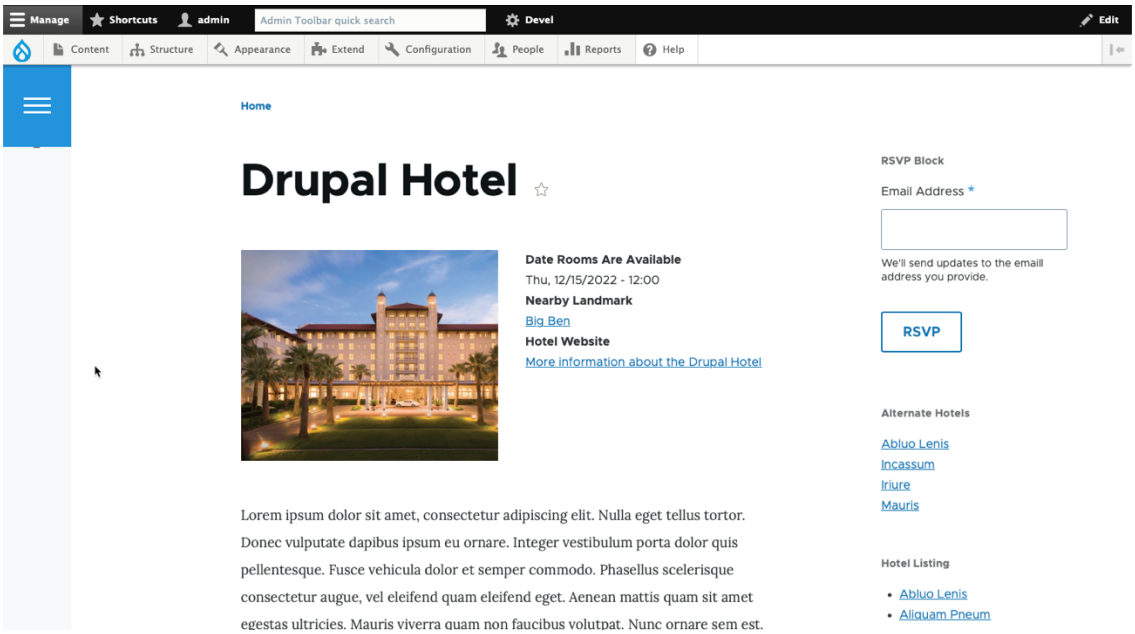
11. Drag and drop the **Hotel video** field in the section. Your final layout should look like this.



12. Make sure to click the **Save layout** button once you have completed the layout changes.



13. Test your new layout by visiting a Hotel page.



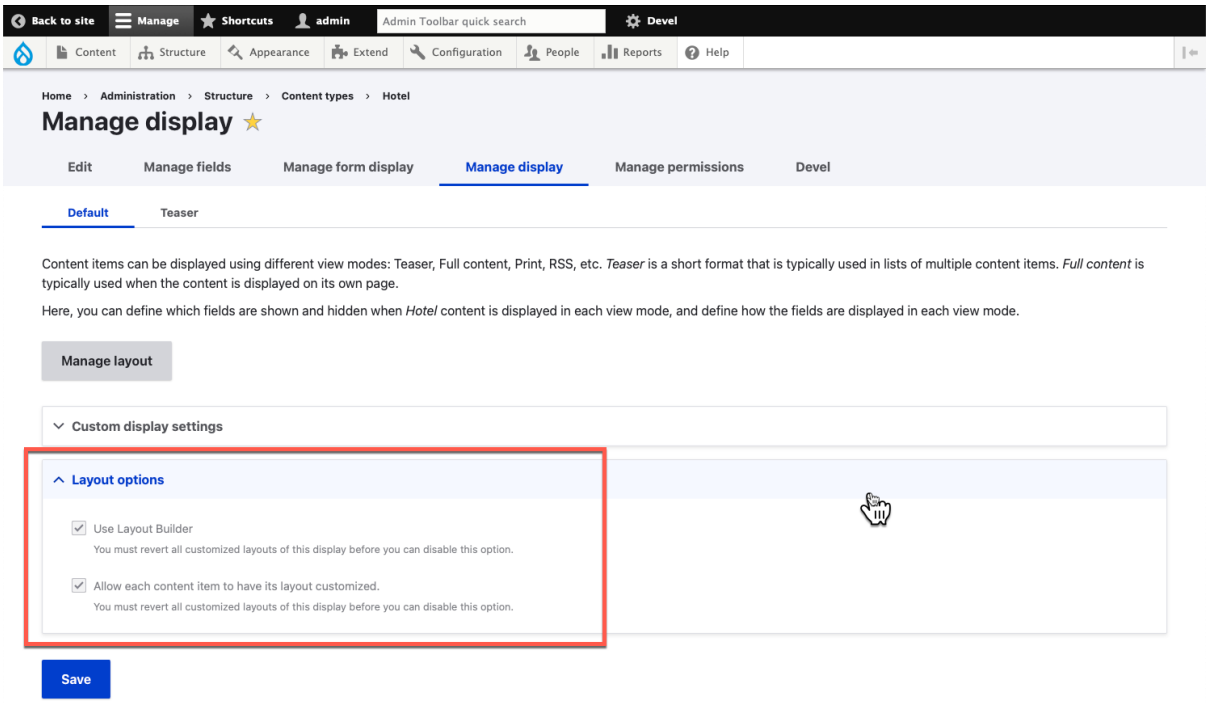
Creating One-Off Landing Pages Using Layout Builder

You can also use Layout Builder to create one-off landing pages like promotional offers pages or seasons greeting pages, etc.

In order to add the functionality to create one-off landing pages, you need to create a new content type. Under the Manage Display section, select the check boxes for Use Layout Builder, and allow each content item to have its layout customized for this content type.

Layout Builder can do **so much more** when paired with contributed modules - https://www.drupal.org/docs/8/core/modules/layout-builder/additional-modules.

Your instructor will provide several examples.



You can use this content type to create one-off landing pages, each with their own unique content and layout as and when needed.

Learning Module 6.0 Summary

In this Learning Module, you learned:

- Nearly everything in Drupal is an entity. Entities can be nodes (content), comments, taxonomy terms, user profiles, or something custom developed.
- Drupal provides several options to manage display of entities.
- You can use Layout Builder to create and edit pages using a drag-and-drop interface.
- You can add an additional page display to an existing View.

7.0 Drupal.org and Extending Drupal (Optional)

Two of the most exciting and significant aspects of Drupal are its extensibility and the community.

Drupal can have its functionality and style extended, enhanced, and added with contributed or custom modules and themes.

Modules are collections of PHP code that you can add to your Drupal installation to change how it behaves. Some modules add entirely new functionality, while others modify an existing element to make it easier to use, or to add related functionality it is missing.

Themes are collections of HTML, CSS, and images that change the way your website looks. Some themes act as skeletons or bases you can add your own styling to, others are complete as they are.

Many modules and themes are created and maintained by the open-source community, and are free for you to use. By using these contributed modules and themes, you can modify how your Drupal site looks and behaves without having to be a programmer or website designer yourself.

And, of course, if you know how to create modules and themes, you can contribute back to the community.

7.1 Drupal.org Community Resources

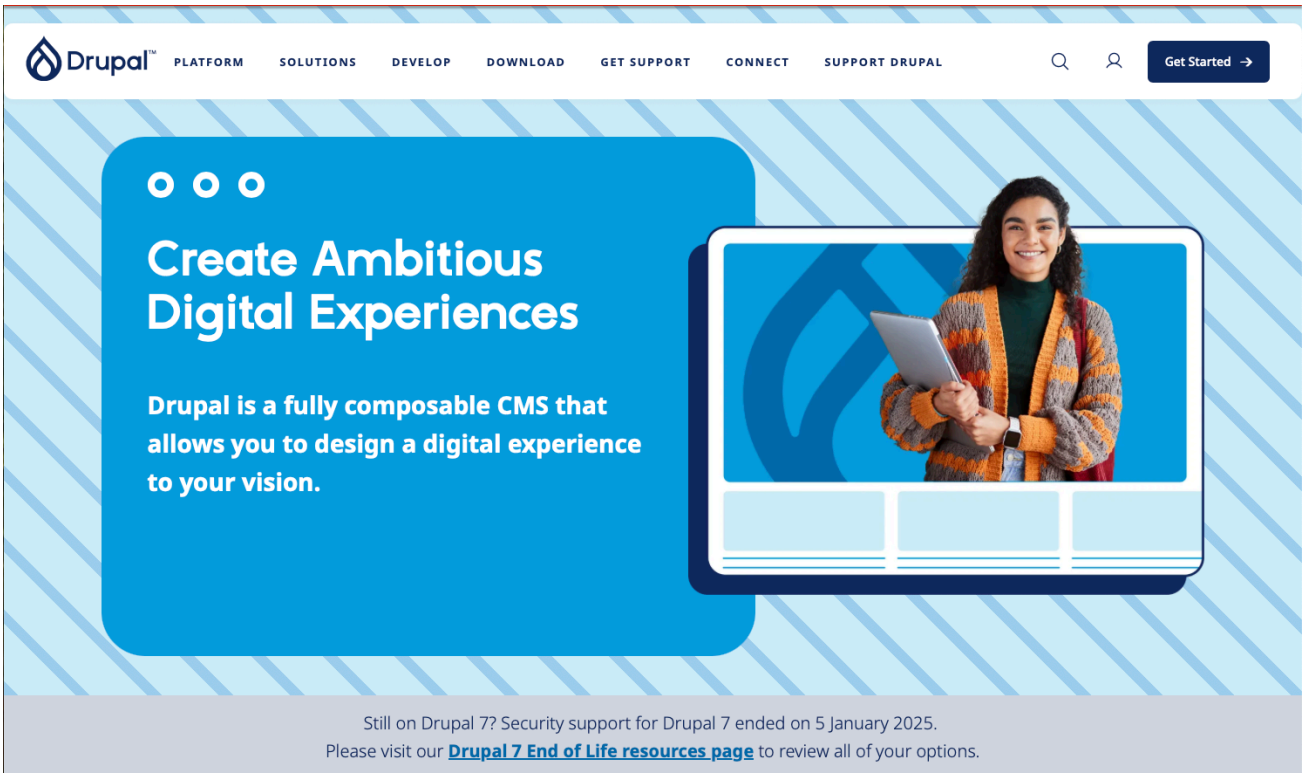
Drupal Community Mission and Principles

The Drupal mission is to build the best open-source content management framework—one that represents the newest ideas and best practices in community publishing, knowledge management, and software design. We believe in open source, innovation, globalism, and collaboration.

You can read more about the community values and principles, as well as details about all of the Drupal projects at:

https://www.drupal.org/about

Tour Drupal.org (d.o)



In addition to being the central repository for all Drupal project code, drupal.org houses many tools that are useful for becoming part of the community, getting support, tracking the latest news, contributing back to the community, and more.

The way to become an official community member is easy. Just create a user account on drupal.org. It's easy and free and only requires you to confirm your email account. With a d.o account you'll be able to take part in group discussions, commit code, and ask questions.

7.2 Extending Drupal With Modules

Can Drupal Do This or That?

Chances are, yes. If you find that a fresh install of Drupal doesn't offer a feature you need, or if you stop and wonder if there's a better way to use some part of the system, you should take a look at popular Drupal modules.

First, visit the list of community-created modules:

https://www.drupal.org/project/project_module

You'll see by default there's a bunch of filters you can use to narrow down your search. You'll also see the default sort order is by most installed. This is a good place to start.

Evaluating Modules to Install

There are several things you can consider when evaluating modules:

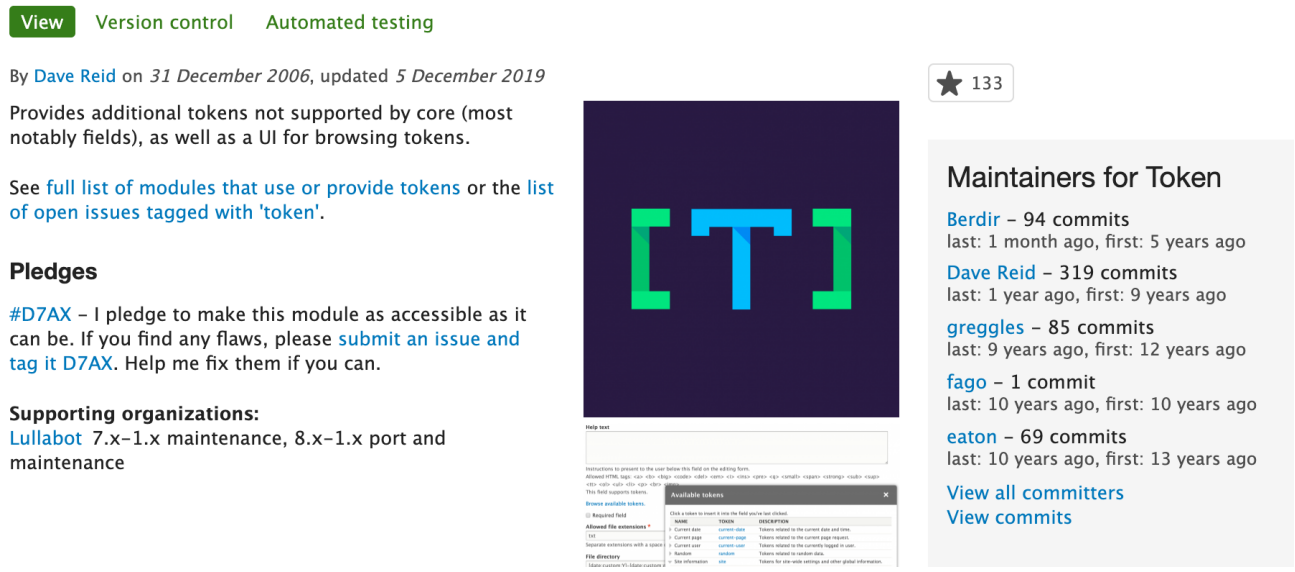
- Popularity of the module
- Number of commits by the module maintainers
- History and frequency of responses to bug reports for the module
- Whether the module does what you need it to do

Popular modules are popular because they fill a need and do it better than other modules do. Frequent usage means they also have more individuals examining them, to find and report bugs to the maintainers.

Looking at the frequency and number of commits module maintainers have done is a good yardstick too. If there is a long history of updates, and the maintainers are responding to bug reports, chances are that history of updates will continue, and that bodes well for your site in the future.

For example, let's again take a look at the Token module (https://www.drupal.org/project/token), as of this writing the maintainer was active very recently. This is a good sign.

Even better, they've been maintaining this module for more than 10 years, showing commitment to it. Since they've been working on the module for so long, they know its code inside and out and will probably be able to patch issues swiftly should any arise.

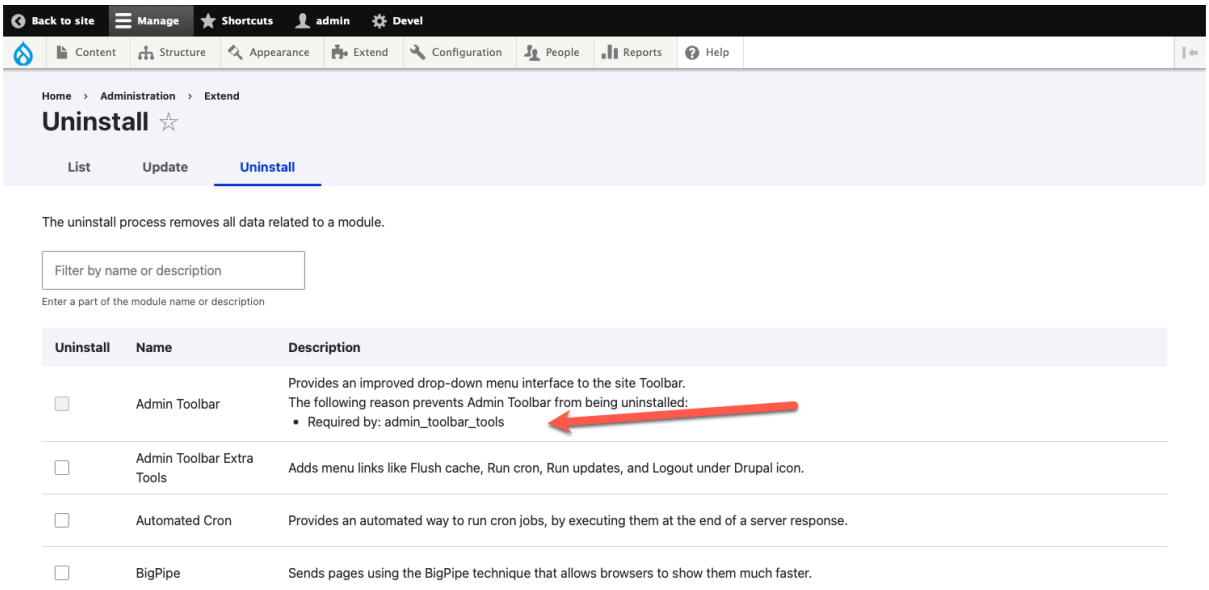


When new versions of Drupal are created, the community takes a look at the current state of Drupal, and removes modules that aren't used from the Drupal core. They also seek to include new, popular modules. In this way, Drupal itself changes to incorporate features its community of site builders needs.

Uninstalling (Disabling) an Installed Module

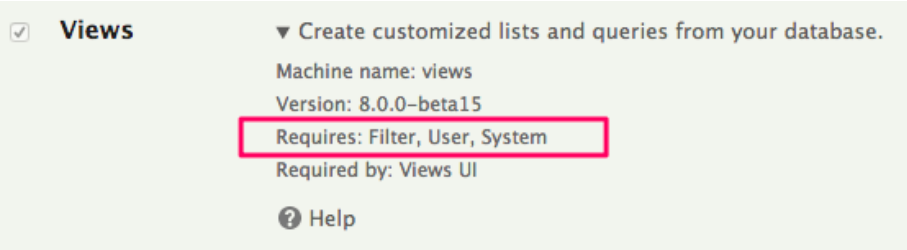
To disable a module, visit the Extend page and click on the Uninstall tab. Check the box next to the module you want to uninstall, and then click on the Uninstall button.

If you are unable to uninstall a module, it means that module is being used by another module. You'll need to uninstall the other module(s) first. You can see which modules are using the module under Required by section.



Module Dependencies

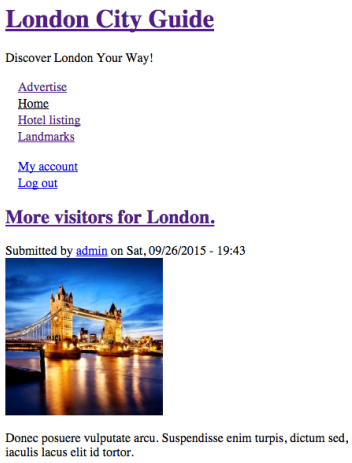
Some modules will require other modules to be installed and enabled to work, or conversely, will require other modules to be disabled before you can disable modules dependent upon them. If you attempt to enable or disable a module and Drupal does not allow you to, it's typically because another module is affecting things.



7.3 Using Themes

Drupal uses a theming system so site builders can quickly change the look of their site using community-contributed themes, or create a new theme based off of a starter theme that has already done some of the scaffolding work for them.

When you install Drupal, it comes with a few different themes pre-installed. To get a better understanding of what Drupal looks like without styling, let's temporarily switch to a bare-bones theme called the Stark theme.



7.3.1 Exercise: Enable the Stark Theme

Steps:

1. Click on **Manage** > **Appearance**.
2. Scroll to **Uninstalled themes**.
3. Click on **Install and set as default** for the *Stark* theme.
4. Click on **< Back to Site.**
5. Observe: The Stark theme has made your site pretty stark.

As you can see, the Stark theme has stripped away most of the styling, coloring, and graphics on your Drupal site, leaving behind just the content.

You could say this is Drupal naked. However as we know, the collaborative Drupal Community has come together to create a multitude of different user-contributed themes that you can install on your own site.

Let's go find a new look for Drupal!

Evaluating Themes

There are two types of themes you can install in Drupal:

- Regular themes that change how a site looks to visitors
- Administration themes that change how a site looks to site administrators and content creators
- When evaluating a regular theme, there are a few things that you want to consider:

- Do you need a Drupal site that will adapt to mobile users? This type of theme is called a responsive theme, as it responds to the user's device, resizing to fit their mobile phone, desktop computer, or tablet.
- Do you need a theme that has the basic structure in place that you will overlay some styling on with CSS, or do you need a complete theme?
- Do you need a theme with many regions? For example, do you want a theme that has one long footer, or a theme that has a long footer but also several smaller footer slots above it that can contain different blocks?
- What sort of menu system do you need? Themes can have a lot of impact on how menus will display.
- How popular is the theme, and is the maintainer of it still active?

You can locate the list of themes on Drupal.org here:

https://www.drupal.org/project/project_theme

Under the "Works with" search option be sure to filter the results to your version of Drupal.

Many of the most popular themes are starter themes, or base themes. This means that they handle the HTML and structure for you, but are plain when it comes to styling. They expect you to have the ability to create your own CSS to manipulate the base theme into the style you want. They might also mention the terms "bootstrap" or "framework."

If you are a beginning Drupalist, those are themes you might want to skip over--at least for now. The reason they are so popular is because a lot of site developers who are familiar with CSS and site design use those themes as a foundation for a more complete or complicated theme, and they save a lot of time.

But as a new Drupal user, you may be disappointed by such themes, as they will be very plain-looking in comparison to the website you envision in your head.

There are complete themes however. Scroll through the choices to find them.

Administration themes are a bit different. They only change how back-end areas look, such as the Administration Toolbar you've been working with today, or the different back-end overlays. If a theme says it's an administration theme, keep in mind it won't change anything that your regular visitors can see on the site.

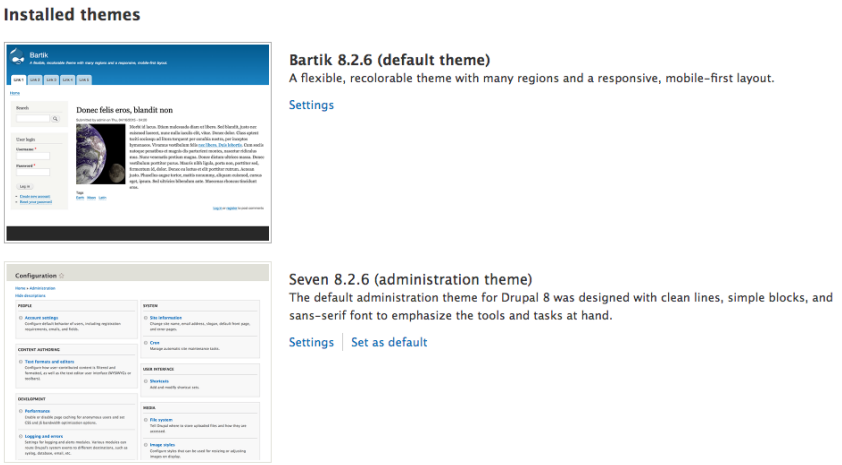
When evaluating all themes, you must still take into account how active the maintainer is. Theme maintainers are not always as active as module maintainers, but active theme maintainers, like module maintainers, are more likely to release new versions or bug fixes for the themes they have created.

Installing a Theme

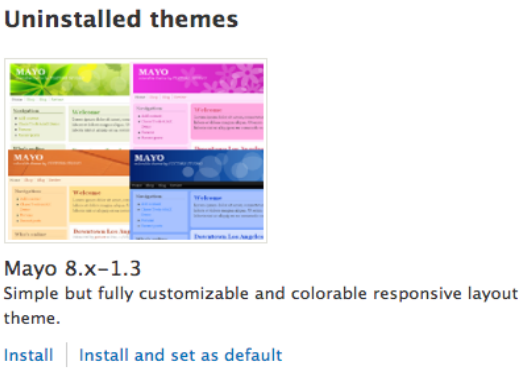
This is accomplished exactly the same way as modules. You must use Composer. Retrieve the composer command from the theme page and enter it into the terminal.

Enabling a Theme

To enable a theme, go to Appearance in the Administration toolbar. You'll see a list of themes that will look something like this:



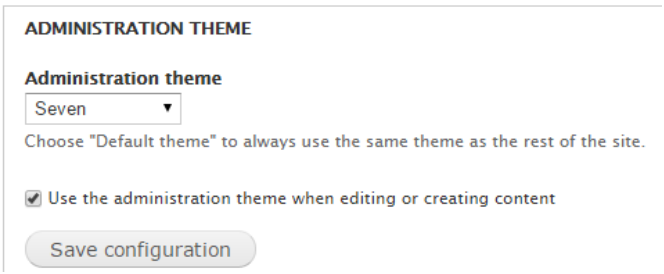
Further down on the page, you'll see your disabled themes.



You'll note that each theme that is not already enabled and default has a link where you can either Enable and set default, or Set default. What is default? Default is the theme a user sees when they visit your site as an anonymous user.

So for a theme to be usable, it must be enabled. To be visible to visitors, it must be set as default.

Administrative themes have their own little configuration box below the disabled themes:



Use the drop-down box to choose an Administration theme. Remember to **Save** configuration.

Drupal does not clearly differentiate between themes meant for the main site, or only the Administrative side. It's up to you to assign the right theme in the right spot.

Assigning a theme that is Administration-only as the default theme for the entire site may cause the main site to have minimal to no styling, and look very like the Stark theme.

7.3.2 Exercise: Browse Available Drupal Themes

Steps:

1. Go to https://www.drupal.org/project/project_theme to view available themes.
2. Change Works with to **Drupal 11** in the provided filters.
3. Click **Search**.
4. Observe: Results are sorted by default with the most popular at the top.
5. Examine themes that interest you.

7.3.3 Exercise: Install and Enable a New Theme

Steps:

1. Go to https://www.drupal.org/project/vani.

2. Right-click the tar.gz file for the 8.x or 9.x version.
3. Copy the Composer command and install it via the command line.
4. Return to your Drupal site.
5. Go to **Manage > Appearance > + Install new theme**
6. Paste the link into the *Install from a URL* box.
7. Click **Install**.
8. Observe: You'll see a page with an install bar. Give it time to complete.
9. Click **Install newly added themes**.
10. Scroll down to *Uninstalled Themes*.
11. Click on **Install and set default** for the *Vani* theme.
12. Click **< Back to Site**.
13. Observe: See your site changed again.
14. Question: Do you still see your blocks? How would you fix that?

Learning Module 7.0 Summary

In this Learning Module, you learned:

- Drupal functionality can be extended by adding modules.
- Drupal can change appearance using themes.
- You can use the administrative interface to extend Drupal without going into code.

Appendix: Advanced Site Builder Concepts: Caching, Multilingual, Pathauto

There are some concepts that you may not be thinking about today, but they may become quite important to you as you continue in your Drupal website development career.

With a database-driven website, like Drupal, heavy loads can lead to performance lags. As a Drupal developer you'll need to be vigilant about how the code and configurations you commit will ultimately impact performance.

When you work on a highly configurable Drupal site you'll often find that configurations need to be tested and then applied to a live site. In previous versions of Drupal this presented some problems. Drupal has designed a fresh Configuration Management system to make updates cleaner.

When you work with websites that have multilingual visitors, you'll need to understand how to create multilingual websites. Luckily, Drupal has many features built in to make your international language sites easier to launch.

In this Learning Module we will take a look at Caching, Configuration Management, and Multilingual concepts in Drupal.

Caching and Drupal

Building complicated, dynamic content in Drupal is easy, but it can come at a price. A lot of what makes a site engaging can dramatically slow performance under a heavy load. The slowdowns are at their worst when traffic is high and Drupal is required to perform complex database queries and expensive calculations every time a user looks at a page.

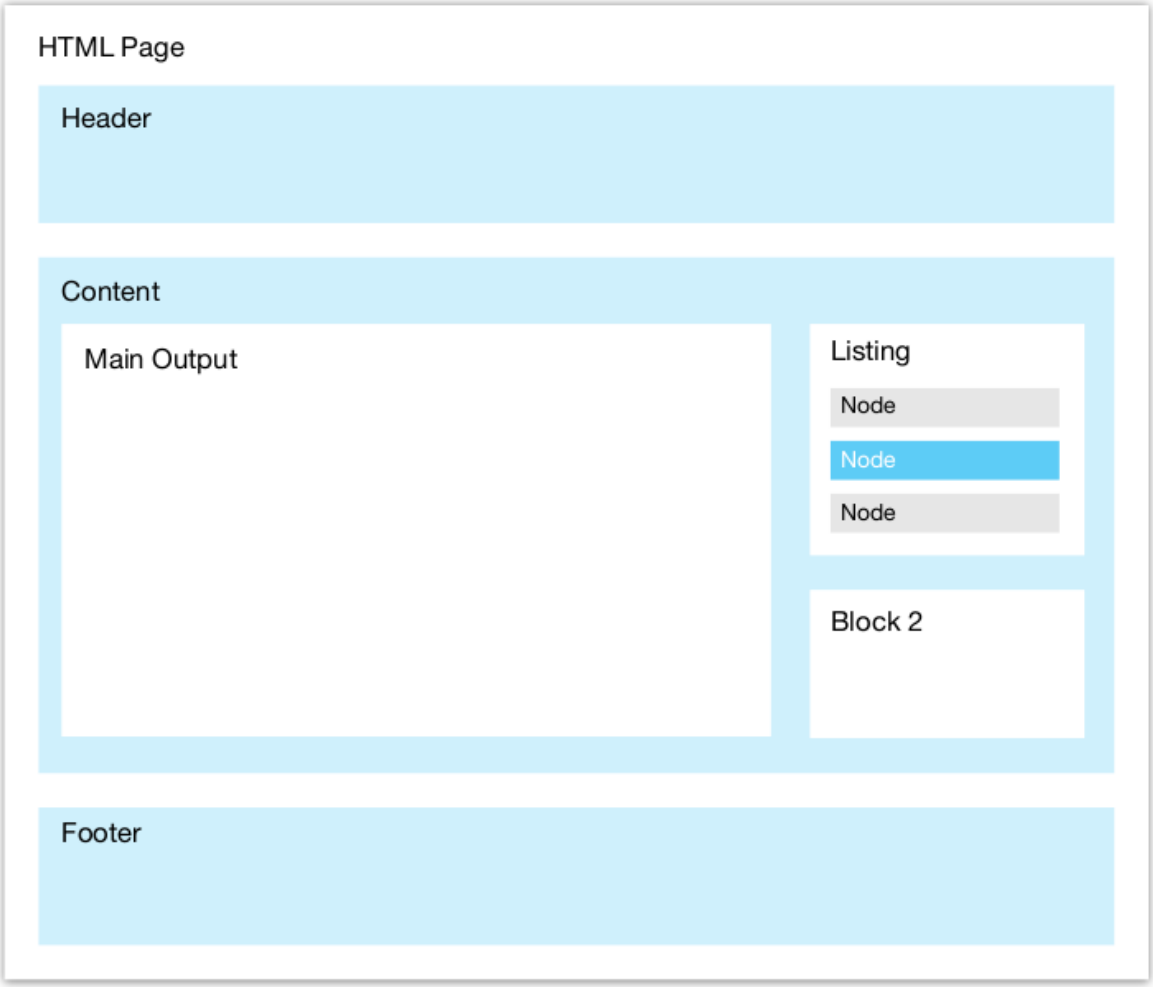
Caching is a way of holding query and calculation results in storage to allow fast retrieval without having to recalculate or query the data from the original source. Caching works best on data that will not change frequently.

We often see caching scenarios for web sites. Often we can identify pages or elements of pages that don't change frequently. These pages are great candidates

for storing a cached copy. Of course, if a page doesn't change every page load then caching it saves time (and server resources) and visitors can experience fast page loads.

The Cache API in Drupal has page caching enabled by default. Depending on the needs of your site this can make your site between much faster anonymous users! So, right out of the box Drupal will respond in constant time, where earlier versions of Drupal are slower depending on the complexity of the page.

To understand the complexity of a page, it's important to understand that a page in Drupal can have blocks, listings, entities, regions, and many other objects. When one item changes, the container of that item needs to be fully rebuilt. Often, that item is the whole page, a problem requiring a much-needed solution. As an example, consider the following page:



Previous versions of Drupal cached the entire page, so if something on the page changed, like the contents of a Listing Block, the entire page cache had to be

rebuilt. This was not the best scenario, so Drupal has an all new caching approach. In fact, the caching system has been entirely rebuilt. Drupal's caching system addresses the question: Why can't we just rebuild the parts of the cache that have actually changed?

Render Caching

Render Caching is a way to cache the page in parts, which means only the individual part needs to be rebuilt when changes occur. Render cache is active by default for all entities, which means that the output of all entities is cached once viewed. Whenever an entity is changed, or something that is referenced by that entity changes, the cache is cleared automatically, thanks to cache tags.

Cache Tags

Cache tags allow the developer to define specific cache-clear scenarios for their site based on the known behavior of the site. This means it is possible to clear all caches that contain information about user 300, following an update to their account, without clearing every other user's cached data.

Cacheability Metadata

Cache tags are one of three properties of cacheability metadata Drupal uses to provide caching. Cacheability metadata is used by the caching system to determine what gets cached and how it gets cached. The three primary properties of cacheability metadata that Drupal uses are:

- Cache tags
- Cache contexts
- Cache max-age

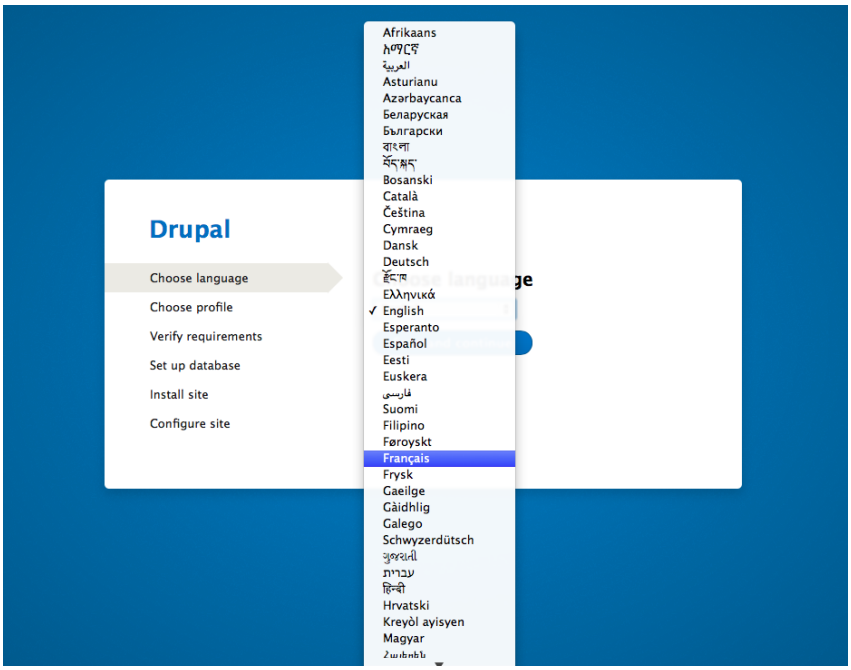
The important takeaway here is that Drupal developers followed a 'fast by default' principle to assure that nothing needs to be enabled to provide performance boosts out of the box.

> *For Further Study*
> *You can get the specifics of each cacheability metadata property and how they get used, as well as more information about caching in Drupal by visiting:*
> *https://www.drupal.org/developing/api/8/cache*

Multilingual Capabilities

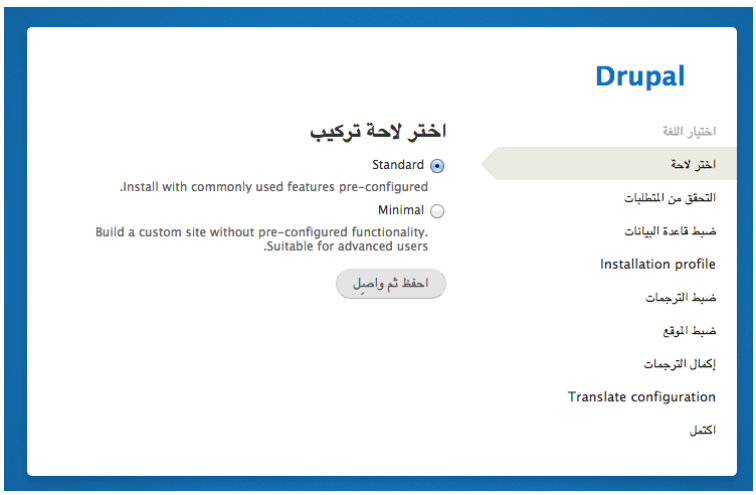
Multilingual First

You get a peek at Drupal's multilingual infrastructure from the first moments of the installation process:



For convenience during installation, Drupal auto-detects the language from your browser and then pre-populates your language option. But even better, if you proceed with installation in a non-English language, Drupal automatically downloads the latest interface translations so you can perform your entire site installation in your native language.

This also works for right-to-left languages, such as Arabic, as you can see here:



Drupal does away with the concept of English as a special language. If you select a language other than English, the English option will no longer show up in your site configuration unless explicitly turned on.

Multilingual Modules

Building a multilingual site in previous versions of Drupal required somewhere around 30 contributed modules and a lot of tricky configuration. In Drupal, all of this functionality has been streamlined into just a couple of modules.

Language : Built in to core, this is the base module for handling language functionality in Drupal. It allows users to configure languages and apply them to content. The other three core multilingual modules below are dependent on this module.

Configuration Translation : Built in to core, provides a translation interface for configuration options. Everything built into the unified configuration system (views, site name, vocabularies, blocks, menus, etc.) can be translated with this module.

Content Translation : Built in core, this module allows users to translate content entities, which includes comments, nodes, users, and taxonomy terms.

Interface Translation : Built in core, this module translates the built-in user interface.

More Streamlined Translation UIs

The user experience of Drupal multilingual functionality is strong, featuring streamlined translation and well-integrated interfaces throughout.

Singular form One translation file imported for the enabled modules. **Plural form** @count translation files imported for the enabled modules.	**Singular form** A bekapcsolt modulokhoz egy fordítási fájl lett betöltve. **Plural form** A bekapcsolt modulokhoz @count fordítási fájl lett betöltve.
Singular form Changed the content type of 1 post from %old-type to %type.* **Plural form** Changed the content type of @count posts from %old-type to %type.*	**Singular form** Egy tartalomtípus %old-type helyett %type lett. **Plural form** @count tartalomtípus %old-type helyett %type lett.
All (@count)	Mind (@count)
Pass (@count)	Átment (@count)
Fail (@count)*	Megbukott (@count)
Singular form 1 debug message **Plural form** @count debug messages	**Singular form** 1 hibakereső üzenet **Plural form** @count hibakereső üzenet

Acknowledgements

Writing and Editing Credits

First Drupal 9 version: Published Sep 2020. Written by **Prasad Shirgaonkar, Acquia and edited by Brian Silver, Acquia**.

Based on,

Previous Drupal 8 versions, published December 2015. Written and edited by Dallas Ramsden of Acquia (https://www.acquia.com) and Margaret Plett of Tyto Learning.

And, Drupal 8 March 2020 Edition Edited and Updated by Prasad Shirgaonkar and Jenn Sramek of Acquia (https://www.acquia.com).

All updates since 2020 (Drupal 9, 10 and 11) by Rod Martin, Tyto Learning. (https://imrodmartin.com / https://drupalhelps.com)