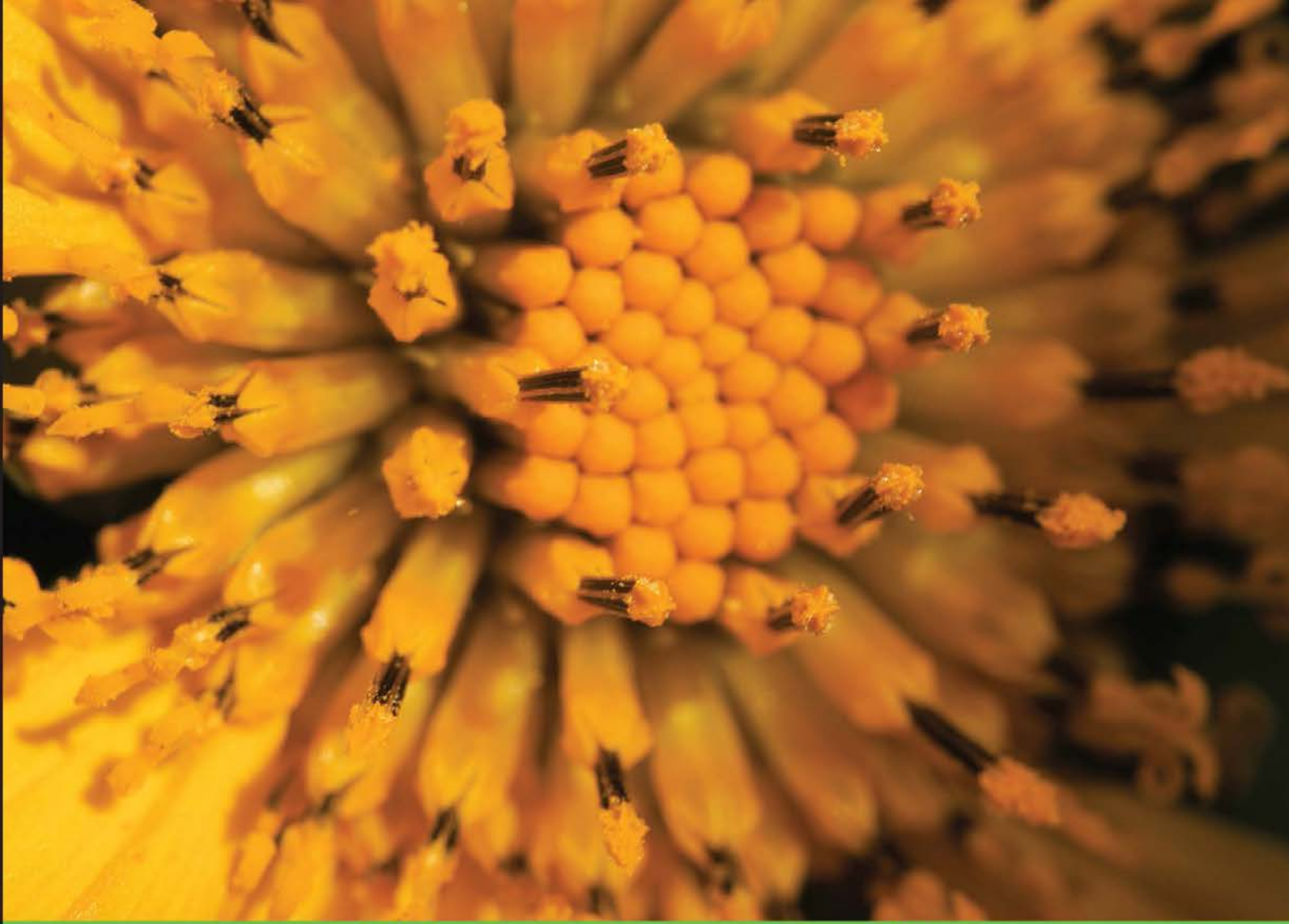


Quick answers to common problems

# Building E-commerce Sites with Drupal Commerce Cookbook

Over 50 recipes to help you build engaging, responsive E-commerce sites with Drupal Commerce

*Foreword by Robert Douglass,*
*Director of Product Strategy, Commerce Guys*

commerce guys

Richard Carter

[PACKT] PUBLISHING open source* community experience distilled

# Building E-commerce Sites with Drupal Commerce Cookbook

Over 50 recipes to help you build engaging, responsive E-commerce sites with Drupal Commerce

**Richard Carter**

[PACKT] open source
PUBLISHING community experience distilled

BIRMINGHAM - MUMBAI

# Building E-commerce Sites with Drupal Commerce Cookbook

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: June 2013

Production Reference: 1180613

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78216-122-6

www.packtpub.com

Cover Image by Asher Wishkerman (wishkerman@hotmail.com)

# Credits

**Author**
Richard Carter

**Reviewers**
Augustin Delaporte
Nicolas Meyer
Surendra Mohan
Dave Pullen

**Acquisition Editor**
Andrew Duckworth

**Lead Technical Editor**
Ankita Shashi

**Technical Editors**
Sharvari H. Baet
Ankita R. Meshram

**Project Coordinator**
Anurag Banerjee

**Proofreader**
Aaron Nash

**Indexer**
Rekha Nair

**Production Coordinator**
Shantanu Zagade

**Cover Work**
Shantanu Zagade

# Foreword

Drupal, one of the most loved and trusted open source content management systems, has seen 12 years of strong growth and become a top choice for building any type of website. In many areas, Drupal has become the de-facto standard, including media, government, university, and library sites.

Since 2011, Drupal has also had an enterprise class e-commerce framework available called Drupal Commerce. The fastest growing open source cart in history, Drupal Commerce already powers a large number of very impressive sites, selling products from luxury goods (Cartier), fashion (Kenzo), online learning (Open Sesame), travel (Eurocentres), to postage stamps and fishing licenses (the Royal Mail of the United Kingdom).

This book explores Drupal Commerce through the lens of the Commerce Kickstart distribution—an out-of-the-box store, completely built and ready to go, featuring the best practices for selling shipable goods online. Commerce Kickstart will not only save you weeks of work setting up your shop, it is an invaluable resource for studying what can be done with Drupal Commerce, and how.

Once you dive into the world of Drupal Commerce, you get much more than just a great shopping cart. Of all the e-commerce systems available, Drupal Commerce is by far the most flexible, leveraging key Drupal-native components such as Rules and Views to achieve much of its functionality. It is also the only system that is natively married to a content management system. In fact, Drupal's heritage as a social collaboration tool makes it the best choice for any project that focuses on content, community, and commerce in a holistic, integrated way.

Finally, by learning Drupal Commerce now, you are making the right move at the right time. e-commerce is the quickest growing new frontier for the Drupal ecosystem. Whether you're a seasoned e-commerce pro, or preparing for your first online shop, Drupal Commerce can enable you to address a new set of business needs and opportunities. Good luck, and enjoy yourself!

**Robert Douglass**
Director of Product Strategy, Commerce Guys

# About the Author

**Richard Carter** is a web designer and a frontend web developer with a passion for a range of open source e-commerce and content management systems, including Magento, MediaWiki, and Drupal. He has worked with clients including Badminton England, University College Dublin, Directgov, NHS Choices, and BusinessLink.gov.uk.

Richard is Creative Director at Peacock Carter Ltd (`peacockcarter.co.uk`), a web design and development agency based in the North East of England. He graduated from the University of Durham in Software Engineering, and currently lives in Newcastle upon Tyne. He blogs at `earlgreyandbattenburg.co.uk` and tweets at `twitter.com/RichardCarter`.

This book is the author's fifth; Richard has previously written *MediaWiki Skins Design*, *Magento 1.3 Theme Design*, *Joomla! 1.5 Templates Cookbook*, and *Magento 1.4 Themes Design* and has acted as a technical reviewer on *The Definitive Guide to Drupal 7*, *Drupal 7 Business Solutions*, *MediaWiki 1.1 Beginners Guide*, and *Inkscape Illustrator's Guide*.

A huge thank you must go to the guys behind Drupal Commerce, Commerce Guys, who put a lot of effort into ensuring modules were stable for the book's release, as well as days or even weeks of their time reviewing the book's content.

Finally, thanks to my friends and family for their support, in particular, Karina, who has coped spectacularly well (perhaps too well!) with my late nights and weekends of work.

# About the Reviewers

**Augustin Delaporte** graduated from INSA Lyon, one of the top engineering schools in France. During his studies, he worked in Shanghai as a Drupal developer.

After his internship as a project manager for six months, where he demonstrated his technical ability, enthusiasm, and communication skills, Augustin joined Commerce Guys in Paris to strengthen the Product and Training teams. He was part of the team that created Commerce Kickstart.

**Nicolas Meyer** has been a full time Drupal developer since 1997. He started his Drupal career by leading the conversion of the French Primer Minister's website to Drupal, and joined Commerce Guys at its inception. He has been a key member of the Commerce Kickstart team since the beginning of the project.

**Surendra Mohan** is currently serving as Drupal Consultant cum Drupal Architect at a well-known software consulting organization in India. Prior to joining this organization, he served a few Indian MNCs, and a couple of startups in varied roles such as Programmer, Technical Lead, Project Lead, Project Manager, Solution Architect, and Service Delivery Manager. He has more than 8 years of working experience in Web Technologies covering media and entertainment, real estate, travel and tours, publishing, e-learning, enterprise architecture, and so on. He is also a well-known speaker who delivers talks on Drupal, Open Source, PHP, Moodle, and so on along with organizing and delivering TechTalks in Drupal meet-ups and Drupal Camps in Mumbai, India.

He has also reviewed other technical books, namely, *Drupal 7 Multi Site Configuration* and *Drupal Search Engine Optimization*.

> I would like to thank my family and friends who supported and encouraged me in completing my reviews on time with good quality.

**Dave Pullen** joined the Commerce Guys team in May of 2012 after completing a training course intended to refocus his skill set towards software development. He brings with him more than 11 years of CMS experience (primarily focused on newspapers, magazines, and radio) from a variety of positions including technical support, quality assurance, and product management.

# www.PacktPub.com

## Support files, eBooks, discount offers and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

PACKTLiB™

http://PacktLib.PacktPub.com

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

- Why Subscribe? Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

## Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

# Table of Contents

# Preface

*Building E-commerce Sites with Drupal Commerce Cookbook* is a guide to Drupal Commerce, the highly functional and expansive e-commerce system for the Drupal content management system.

Whether you're a beginner with e-commerce and want to create an online store as quickly as possible to start selling, or you're a web designer, who is simply new to Drupal Commerce and wants to get it up and running quickly, Drupal Commerce Cookbook can help you launch your next e-commerce store with style!

## What this book covers

*Chapter 1*, *Installing and Configuring Drupal Commerce*, helps you get Drupal Commerce up and running and configured for basic e-commerce store requirements.

*Chapter 2*, *Configuring Products with Drupal Commerce*, takes you through creating products and categories in your store, and how to start customizing products to allow file uploads and other custom fields.

*Chapter 3*, *Managing Your Drupal Commerce Store Day-to-day*, includes useful hints and tips to keep your Drupal Commerce store up to date and useful how-to's of tasks you may need to do on a regular basis.

*Chapter 4*, *Useful Modules for Drupal Commerce*, explores some of the most useful modules you can use to extend and improve your online store.

*Chapter 5*, *Payment and Shipping Modules for Drupal Commerce*, provides walkthroughs of installation and configuration of the most common payment gateways, including SagePay and PayPal, as well as custom shipping methods.

*Chapter 6*, *Drupal Commerce Search Engine Optimization*, provides hints and tips to help boost your online store's search engine friendliness to help your store succeed online.

*Chapter 7*, *Theming Drupal Commerce*, gets to grips with customizing the look and feel of your Drupal Commerce store.

*Chapter 8*, *Further Configuration*, explores some further enhancements and configurations you can make to your store to improve it for both store owners and customers alike.

*Appendix*, *A Blueprint for a T-shirt Store in Drupal Commerce*, pulls the other chapters together into creating a store selling t-shirts online.

# What you need for this book

You will need access to a suitable platform to host Drupal 7 and Drupal Commerce. You can find a list of up-to-date requirements at `http://drupal.org/project/commerce`.

A basic knowledge of content management is assumed; pre-existing knowledge of Drupal or Drupal Commerce is not assumed. A basic knowledge of CSS and HTML may be beneficial, but is not entirely necessary.

# Who this book is for

This book is for online entrepreneurs, site builders, and web designers who want a helping hand at getting their online store up and running with Drupal Commerce.

This book focuses heavily on tasks that non-technical readers can achieve, bringing a fully functioning e-commerce website closer to those who have a get-up-and-go attitude thanks to Drupal.

# Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles are shown as follows: "So, `sites/default/files/favicon.ico` or, simply, `favicon.ico` are good values for this field."

A block of code is set as follows:

```
h1,h2,h3,h4,h5,h6 {
  font-family: "PT Sans", Arial, Helvetica, sans-serif;
}
```

When we wish to draw your attention to a particular part of a code block, the relevant lines or items are set in bold:

```
  <?php print render($page['content']); ?>
  <p class="container-24">
    <a href="javascript:history,back(-1)">Continue shopping</a>
  </p>
<?php endif; ?>
```

**New terms** and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "Clear Drupal's cache by navigating to **Site settings** | **Configuration** | **Performance** and clicking on the **Clear all caches** button."

> Warnings or important notes appear in a box like this.

> Tips and tricks appear like this.

# Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

# Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

## Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

## Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

## Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

## Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

# 1

# Installing and Configuring Drupal Commerce

In this chapter we will cover the following topics:

- Installing Drupal Commerce to an existing Drupal 7 website
- Installing Drupal Commerce with Commerce Kickstart 2
- Creating your first basic product
- Configuring the checkout in Drupal Commerce
- Configuring the currency in Drupal Commerce
- Changing your store's logo file
- Changing your store's favicon (favorites icon)
- Adding social media icons on product pages

## Introduction

Drupal Commerce (`http://www.drupalcommerce.org`) is a collection of modules that extend the Drupal content management system to support an e-commerce-style store.

Drupal Commerce provides a much more stripped back alternative to the Ubercart module (`http://www.ubercart.org`), which became the most popular e-commerce package for Drupal 6. Ryan Szrama was the lead developer on the UberCart project before moving to work on the Drupal Commerce framework. As a relatively new project, Drupal Commerce is only supported from Drupal 7 onwards.

In particular, an obvious initial difference between Ubercart and Drupal Commerce is that the latter does not come with shipping or payment gateways pre-installed. It is left to the site builder to install the modules they want to use to fulfill these features. As such, Drupal Commerce is presented as more of a framework than Ubercart, which is designed to be more functional out of the box.

This chapter will guide you through the basic configuration your Drupal Commerce store may need to function, from installation to checkout, and currency configuration options.

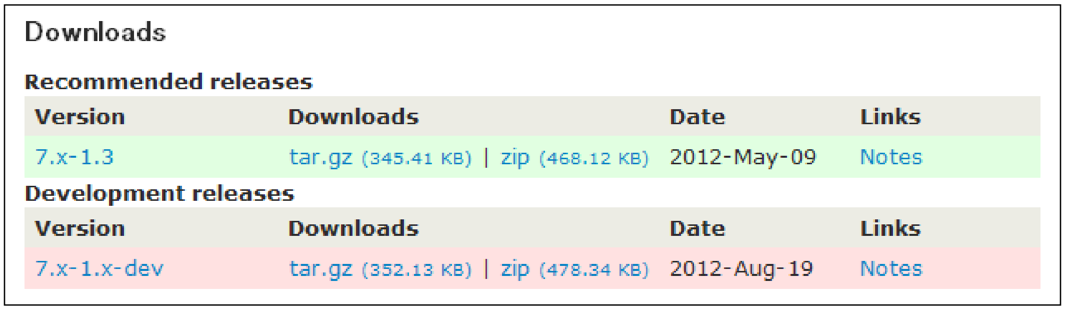
The remainder of the recipes in this book assume that you have used the Drupal Commerce Kickstart installation profile for Drupal, though you should be able to follow all recipes regardless of this.

# Installing Drupal Commerce to an existing Drupal 7 website

There are two approaches to installing Drupal Commerce; this recipe covers installing Drupal Commerce on an existing Drupal 7 website.

## Getting started

You will need to download Drupal Commerce from `http://drupal.org/project/commerce`. Download the most recent recommended release you see that couples with your Drupal 7 website's core version:



Downloads

Recommended releases

| Version | Downloads | Date | Links |
|---|---|---|---|
| 7.x-1.3 | tar.gz (345.41 KB) \| zip (468.12 KB) | 2012-May-09 | Notes |

Development releases

| Version | Downloads | Date | Links |
|---|---|---|---|
| 7.x-1.x-dev | tar.gz (352.13 KB) \| zip (478.34 KB) | 2012-Aug-19 | Notes |

You will also require the following modules to allow Drupal Commerce to function:

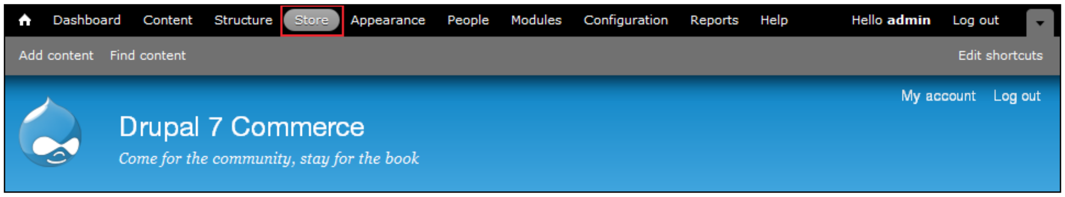
- **Ctools**: `http://drupal.org/project/ctools`
- **Entity API**: `http://drupal.org/project/entity`
- **Views**: `http://drupal.org/project/views`
- **Rules**: `http://drupal.org/project/rules`
- **Address Field**: `http://drupal.org/project/addressfield`

## How to do it...

Now that you're ready, install Drupal Commerce by performing the following steps:

1. Install the modules that Drupal Commerce depends on, first by copying the preceding module files into your Drupal site's `modules` directory, `sites/all/modules`.
2. Install Drupal Commerce's modules next, by copying the files into the `sites/all/modules` directory, so that they appear in the `sites/all/modules/commerce` directory.
3. Enable the newly installed Drupal Commerce module in your Drupal site's administration panel (`example.com/admin/modules` if you've installed Drupal Commerce at `example.com`), under the **Modules** navigation option, by ensuring the checkbox to the left-hand side of the module name is checked.
4. Now that Drupal Commerce is installed, a new menu option will appear in the administration navigation at the top of your screen when you are logged in as a user with administration permissions.

> You may need to clear the cache to see this. Navigate to **Configuration** | **Development** | **Performance** in the administration panel to do this.



## How it works...

Drupal Commerce depends on a number of other Drupal modules to function, and by installing and enabling these in your website's administration panel you're on your way to getting your Drupal Commerce store off the ground.

> You can also install the Drupal Commerce modules via **Drush** (the Drupal Shell) too. For more information on Drush, see `http://drupal.org/project/drush`.

# Installing Drupal Commerce with Commerce Kickstart 2

Drupal Commerce requires quite a number of modules, and doing a basic installation can be quite time-consuming, which is where Commerce Kickstart 2 comes in. It packages Drupal 7 core and all of the necessary modules.

Using Commerce Kickstart 2 is a good idea if you are building a Drupal Commerce website from scratch, and don't already have Drupal core installed.

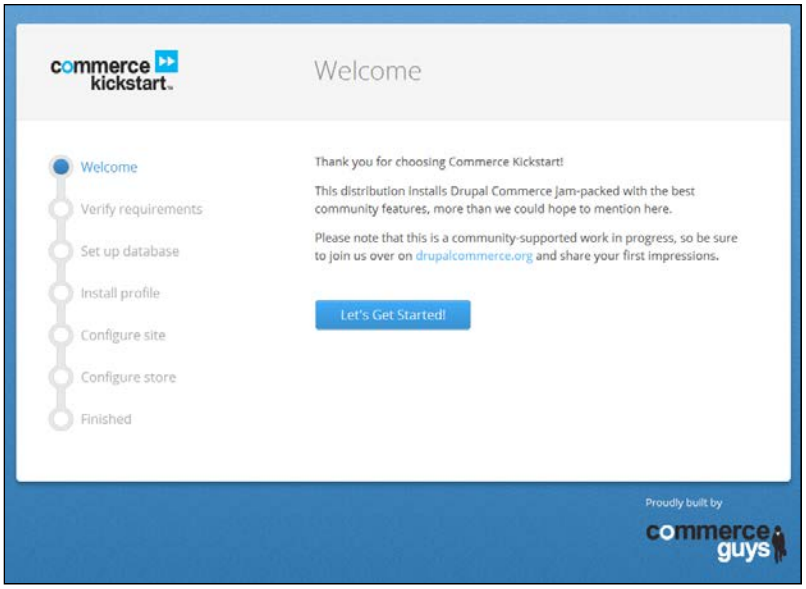
## Getting started

Download Drupal Commerce Kickstart 2 from its `drupal.org` project page at `http://drupal.org/project/commerce_kickstart`.

## How to do it...

Once you have decompressed the Commerce Kickstart 2 files to the location you want to install Drupal Commerce in, perform the following steps:

1. Visit the given location in your web browser. For this example, it is assumed that your website is at `example.com`, so visit this address in your web browser.
2. You'll see that you are presented with a welcome screen as shown in the following screenshot:



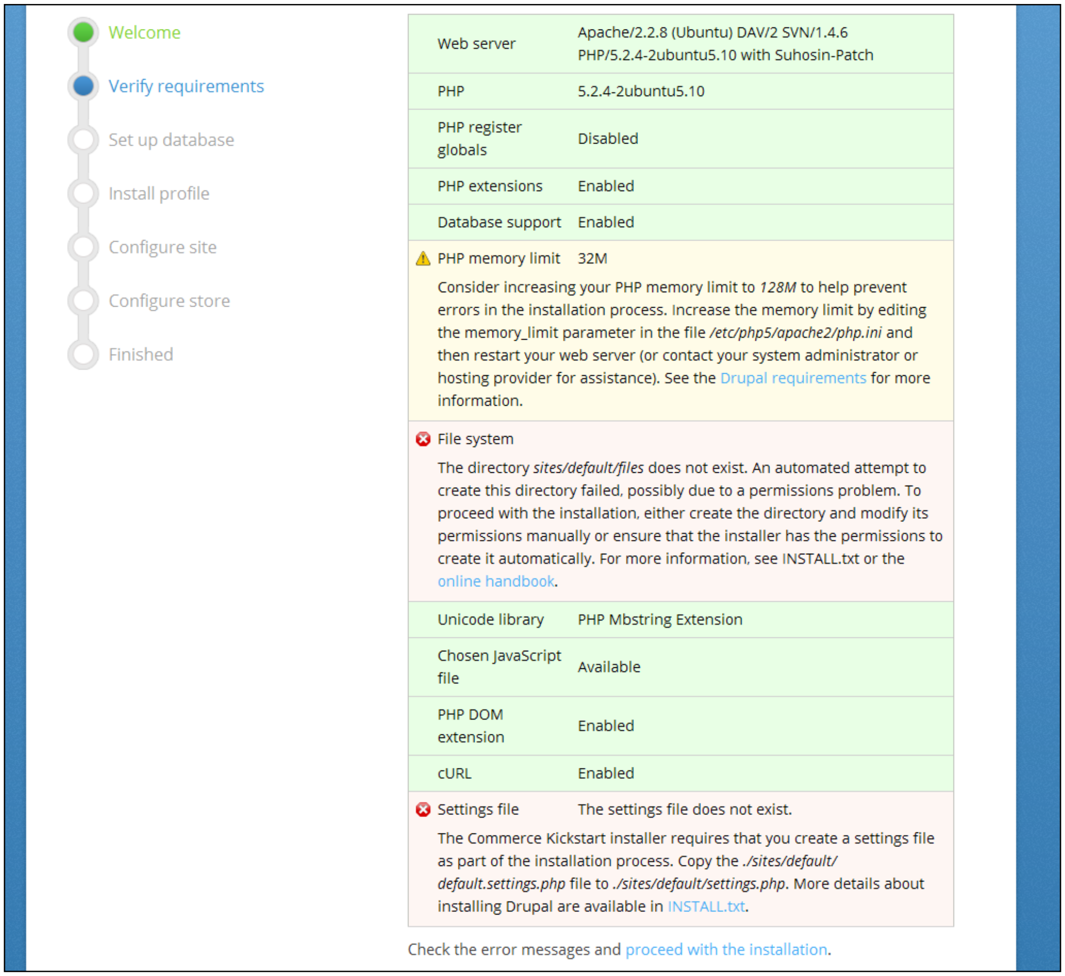
Welcome

Thank you for choosing Commerce Kickstart!

This distribution installs Drupal Commerce jam-packed with the best community features, more than we could hope to mention here.

Please note that this is a community-supported work in progress, so be sure to join us over on drupalcommerce.org and share your first impressions.

3. Click the **Let's Get Started** button underneath this, and the installer moves to the next configuration option.
4. Next, your server's requirements are checked to ensure Drupal can run in this environment.

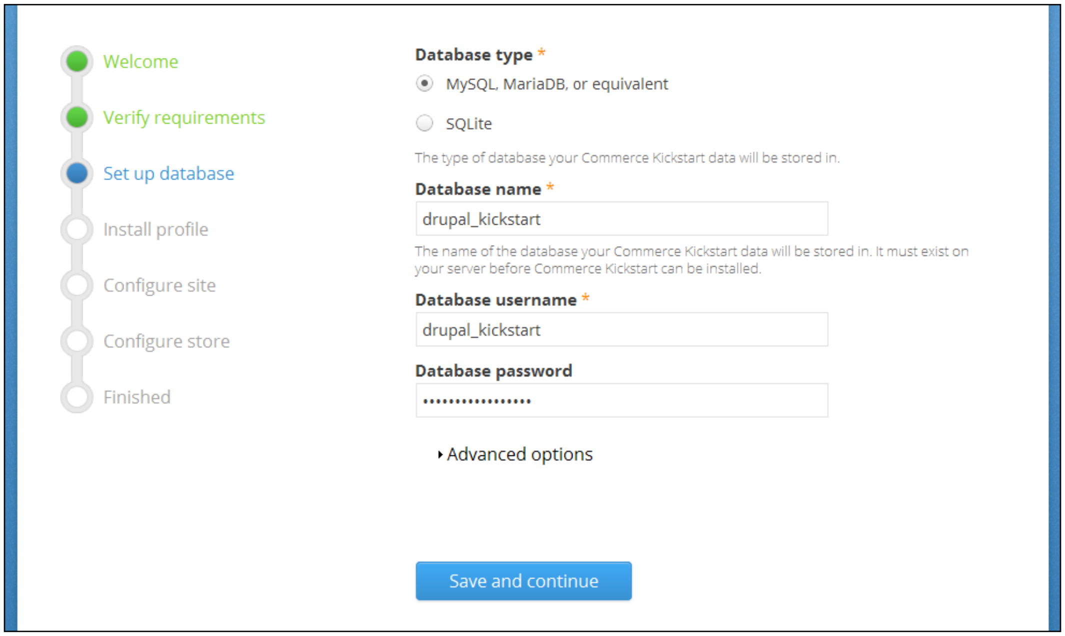


| | |
|---|---|
| Web server | Apache/2.2.8 (Ubuntu) DAV/2 SVN/1.4.6 PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch |
| PHP | 5.2.4-2ubuntu5.10 |
| PHP register globals | Disabled |
| PHP extensions | Enabled |
| Database support | Enabled |
| ⚠ PHP memory limit | 32M<br>Consider increasing your PHP memory limit to *128M* to help prevent errors in the installation process. Increase the memory limit by editing the memory_limit parameter in the file */etc/php5/apache2/php.ini* and then restart your web server (or contact your system administrator or hosting provider for assistance). See the Drupal requirements for more information. |
| ✖ File system | The directory *sites/default/files* does not exist. An automated attempt to create this directory failed, possibly due to a permissions problem. To proceed with the installation, either create the directory and modify its permissions manually or ensure that the installer has the permissions to create it automatically. For more information, see INSTALL.txt or the online handbook. |
| Unicode library | PHP Mbstring Extension |
| Chosen JavaScript file | Available |
| PHP DOM extension | Enabled |
| cURL | Enabled |
| ✖ Settings file | The settings file does not exist.<br>The Commerce Kickstart installer requires that you create a settings file as part of the installation process. Copy the *./sites/default/default.settings.php* file to *./sites/default/settings.php*. More details about installing Drupal are available in INSTALL.txt. |

Check the error messages and proceed with the installation.

5. In the preceding screenshot you can see some common problems when installing Drupal that prevent installation. In particular, ensure that you create the `/sites/default/files` directory in your Drupal installation and ensure it has permissions to allow Drupal to write to it (as this is where your website's images and files are stored).
6. You will also need to copy the `/sites/default/default.settings.php` file to `/sites/default/settings.php` before you can start. Make sure this file is writeable by Drupal too (you'll secure it after installation is complete).

7. Once these problems have been resolved, refresh the page and you will be taken to the **Set up database** screen. Enter the database username, password, and database name you want to use with Drupal, and click on **Save and continue**:



**Database type** *

● MySQL, MariaDB, or equivalent

○ SQLite

The type of database your Commerce Kickstart data will be stored in.

**Database name** *

drupal_kickstart

The name of the database your Commerce Kickstart data will be stored in. It must exist on your server before Commerce Kickstart can be installed.

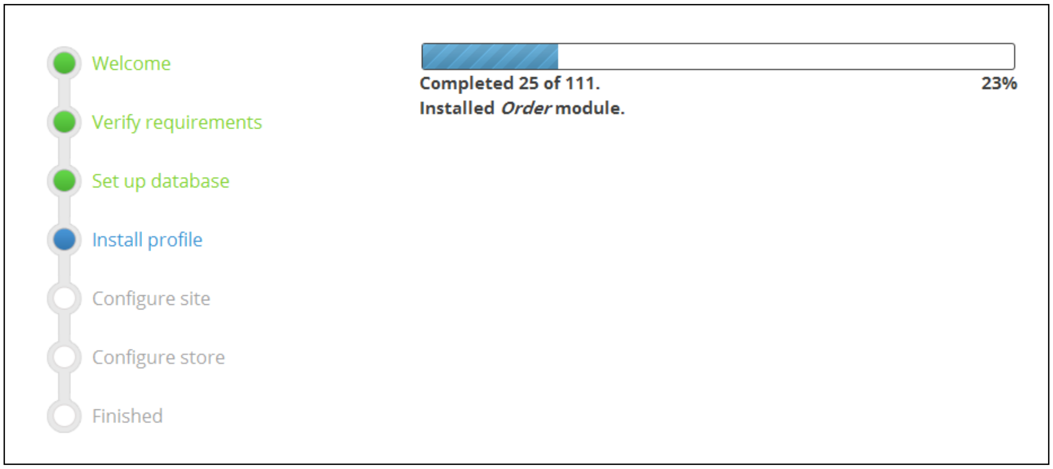
**Database username** *

drupal_kickstart

**Database password**

•••••••••••••••••

8. The next step is the **Install profile** section, which can take some time as Drupal Commerce is installed for you. There's nothing for you to do here; just wait for installation to complete!

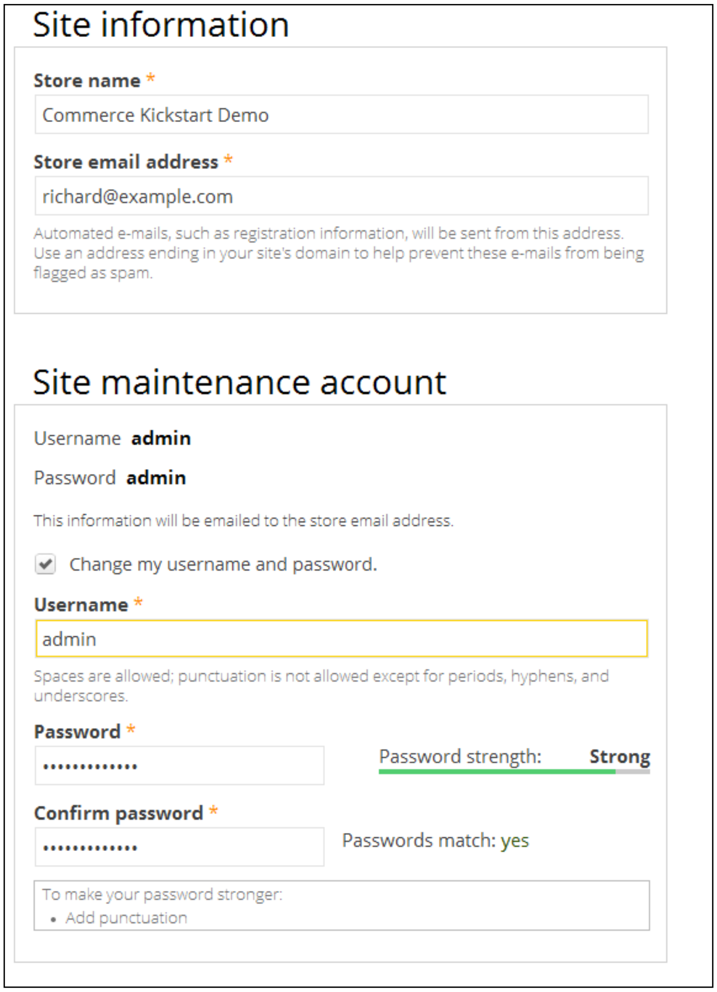


**Completed 25 of 111.** **23%**

**Installed *Order* module.**

9. You can now safely remove write permissions for the `settings.php` file in the `/sites/default` directory of your Drupal Commerce installation.

10. The next step is **Configure site**. Enter the name of your new store and your e-mail address here, and provide a username and password for your Drupal Commerce administrator account. Don't forget to make a note of these as you'll need them to access your website later!



Site information

**Store name** *

Commerce Kickstart Demo

**Store email address** *

richard@example.com

Automated e-mails, such as registration information, will be sent from this address. Use an address ending in your site's domain to help prevent these e-mails from being flagged as spam.

Site maintenance account

Username **admin**

Password **admin**

This information will be emailed to the store email address.

☑ Change my username and password.

**Username** *

admin

Spaces are allowed; punctuation is not allowed except for periods, hyphens, and underscores.

**Password** *

•••••••••••••

Password strength: **Strong**

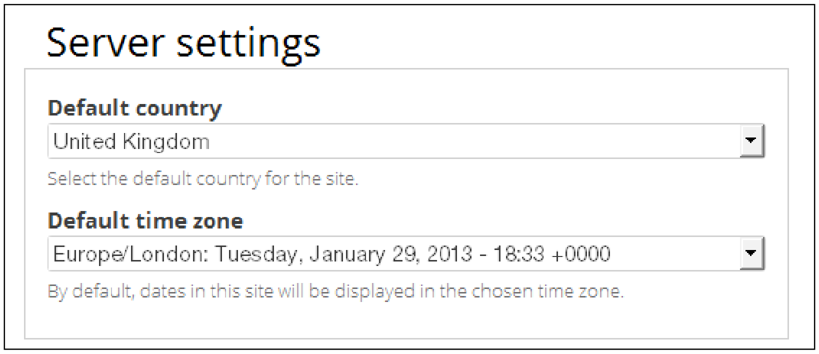
**Confirm password** *

•••••••••••••

Passwords match: yes

To make your password stronger:
- Add punctuation

11. Below these options, you can specify the country of your server and the default time zone. These are usually picked up from your server itself, but you may want to change them:



Server settings

Default country

United Kingdom

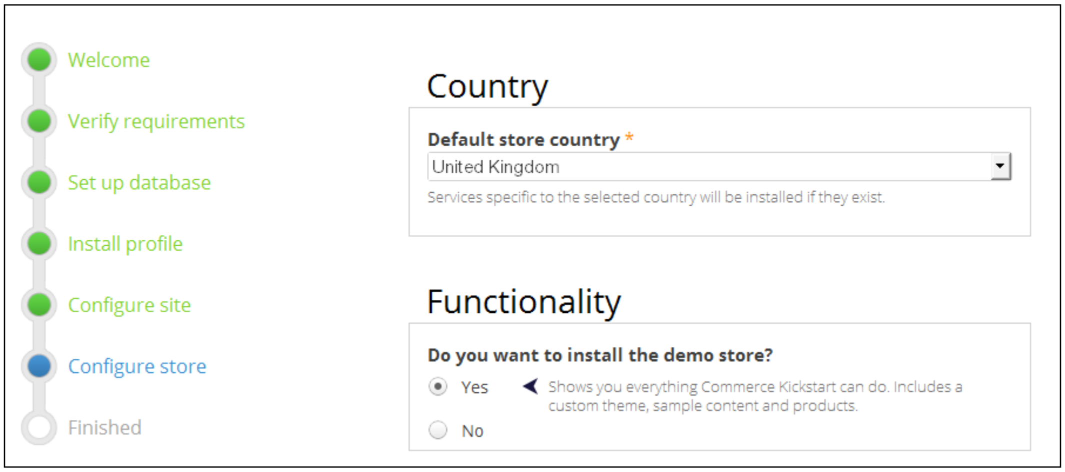
Select the default country for the site.

Default time zone

Europe/London: Tuesday, January 29, 2013 - 18:33 +0000

By default, dates in this site will be displayed in the chosen time zone.

12. Click on the **Save and continue** button to progress now; the next step is **Configure store**. Here you can set your **Default store country** field (if it's different from your server settings) and opt to install Drupal Commerce's demo, which includes sample content and a sample Drupal Commerce theme too:



Country

Default store country *

United Kingdom

Services specific to the selected country will be installed if they exist.

Functionality

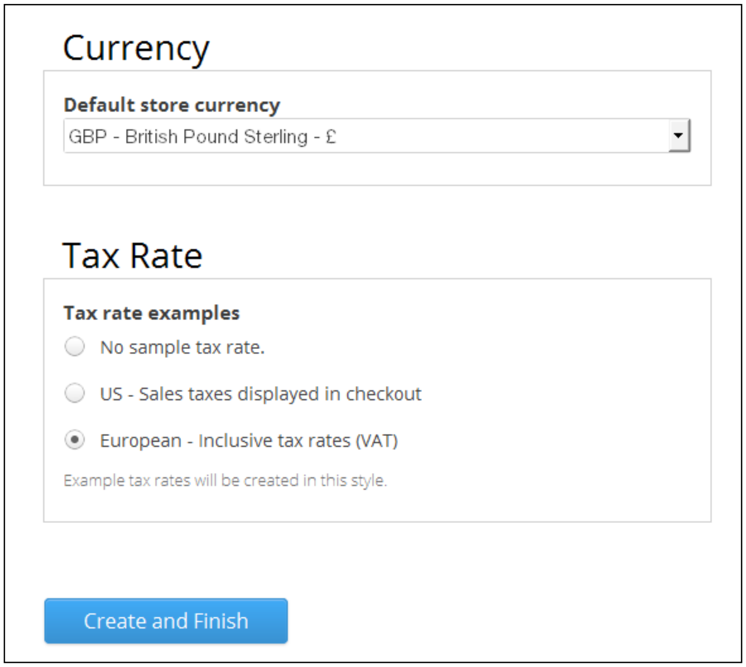
Do you want to install the demo store?

● Yes Shows you everything Commerce Kickstart can do. Includes a custom theme, sample content and products.

○ No

13. Further down on this screen, you're presented with more options. By checking the **Do you want to be able to translate the interface of your store?** field, Drupal Commerce provides you with an ability to translate your website for customers of different languages (for this simple store installation, leave this set to **No**).

14. Finally, you can set the **Default store currency** field you wish to use, and whether you want Commerce Kickstart to set up a sales tax rule for your store (select which is more appropriate for your store, or leave it set to **No sample tax rate** for now):



Currency

**Default store currency**
GBP - British Pound Sterling - £

Tax Rate

**Tax rate examples**

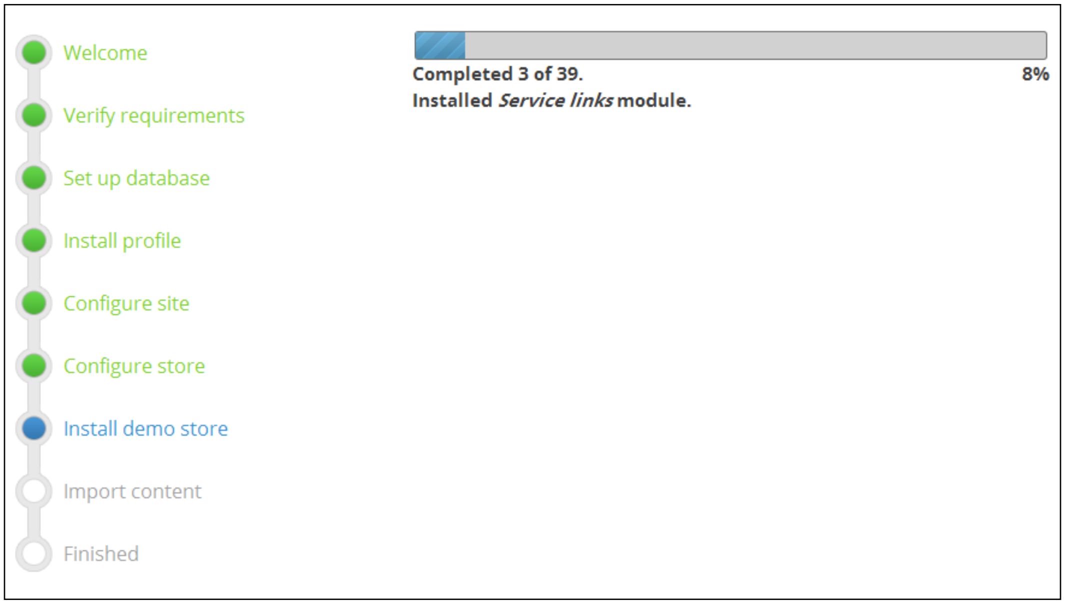
○ No sample tax rate.

○ US - Sales taxes displayed in checkout

● European - Inclusive tax rates (VAT)

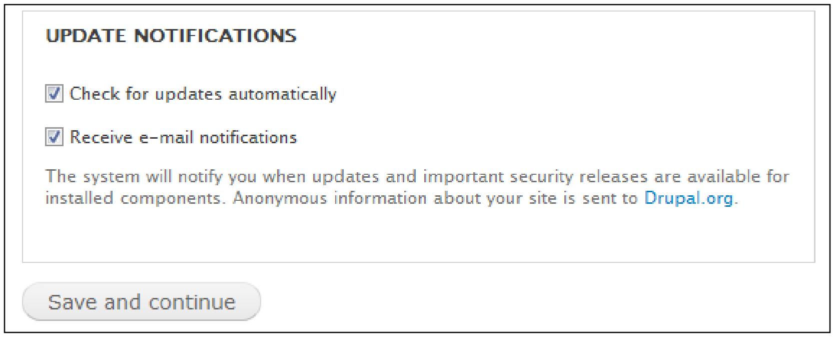
Example tax rates will be created in this style.

15. Click on **Create and finish** at the bottom of the screen. If you chose to install the demo store in the previous screen, you will have to wait as it is added for you.



**Completed 3 of 39.** **8%**
**Installed *Service links* module.**

16. There are now options to allow Drupal to check for updates automatically, and to receive e-mails about security updates. Leave these both checked to help you stay on top of keeping your Drupal website secure and up-to-date.



UPDATE NOTIFICATIONS

☑ Check for updates automatically

☑ Receive e-mail notifications

The system will notify you when updates and important security releases are available for installed components. Anonymous information about your site is sent to Drupal.org.

17. Wait as Commerce Kickstart installs everything Drupal Commerce requires to run.
18. That's it! Your Drupal Commerce store is now up and running thanks to Commerce Kickstart 2.

## How it works...

The Commerce Kickstart package includes Drupal 7 core and the Drupal Commerce module. By packaging these together, installation and initial configuration for your Drupal Commerce store is made much easier!

# Creating your first product

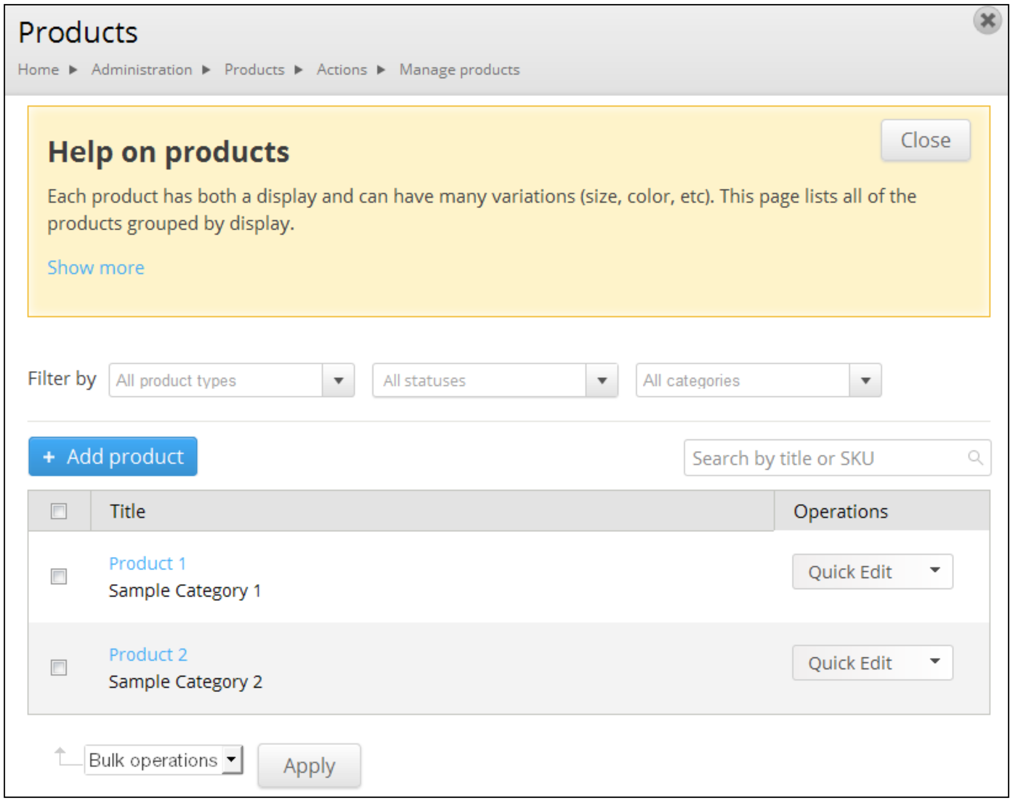
Now that you've installed Drupal Commerce, you can start to add products to display to customers and start making money.

In this recipe you will learn how to add a basic product to your Drupal Commerce store.

## Getting started

Log in to your Drupal Commerce store's administration panel, and navigate to **Products** | **Add a product**:

> This book assumes you enabled the Drupal Commerce admin menu at installation stage. If you haven't, navigate to **Site settings** | **Modules** and ensure that the Commerce Kickstart Menu module is enabled for your store.



Products

Home ▸ Administration ▸ Products ▸ Actions ▸ Manage products

Help on products

Each product has both a display and can have many variations (size, color, etc). This page lists all of the products grouped by display.

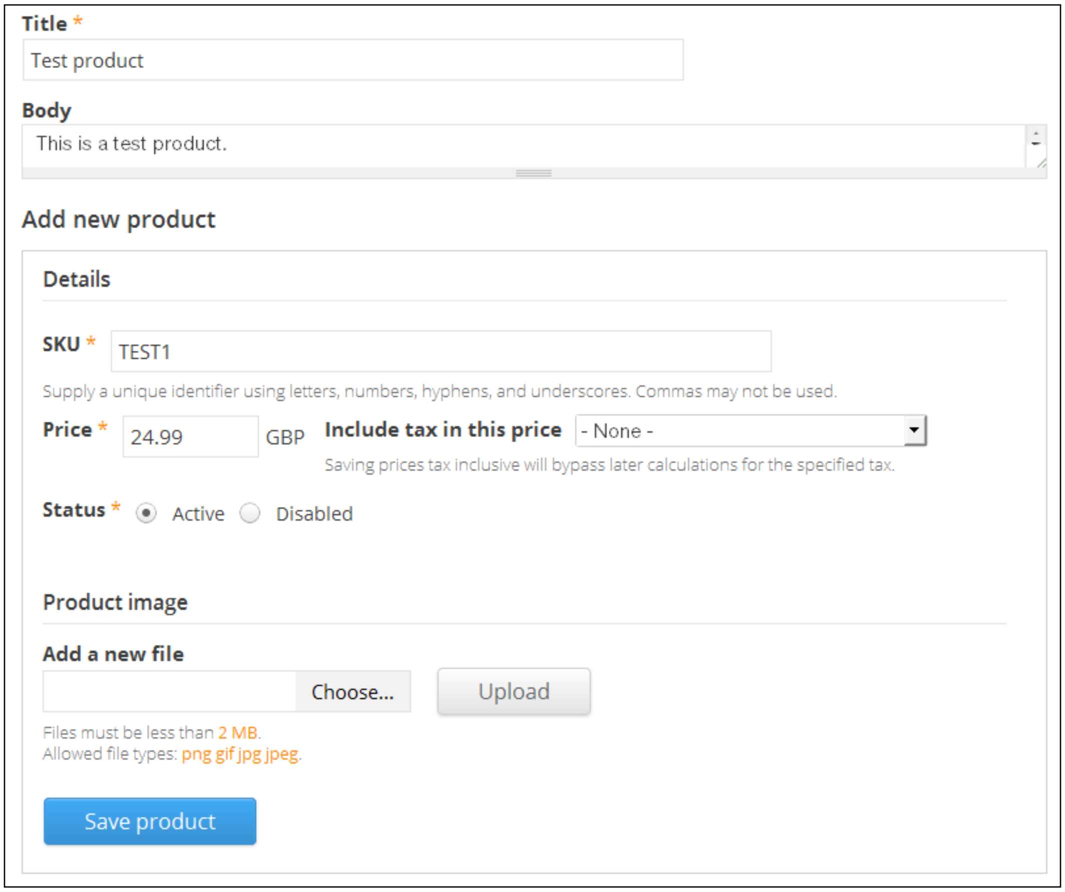
| ☐ | Title | Operations |
|---|---|---|
| ☐ | Product 1<br>Sample Category 1 | Quick Edit ▾ |
| ☐ | Product 2<br>Sample Category 2 | Quick Edit ▾ |

Note the sample products from Drupal Kickstart's installation are displaying there.

## How to do it...

To get started adding a product to your store, click on the **Add product** button and follow these steps:

1. Click on the **Product** display. **Product** displays groups of multiple related product variations together for display on the frontend of your website.
2. Fill in the form that appears, entering a suitable **Title**, using the **Body** field for the product's description, as well as filling in the **SKU** (stock keeping unit; a unique reference for this product) and **Price** fields. Ensure that the **Status** field is set to **Active**. You can also optionally upload an image for the product here:



**Title** *

Test product

**Body**

This is a test product.

Add new product

Details

**SKU** * TEST1

Supply a unique identifier using letters, numbers, hyphens, and underscores. Commas may not be used.

**Price** * 24.99 GBP **Include tax in this price** - None -

Saving prices tax inclusive will bypass later calculations for the specified tax.

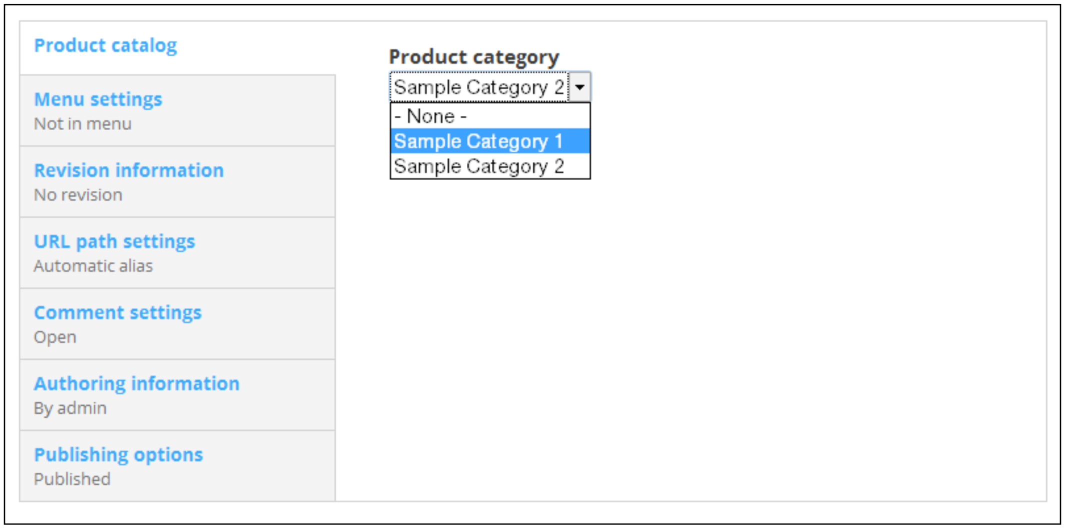
**Status** * ● Active ○ Disabled

Product image

**Add a new file**

Files must be less than 2 MB.
Allowed file types: png gif jpg jpeg.

3. Optionally, you can assign the product to one of the pre-existing categories in the **Product catalog** tab underneath these fields, as well as a URL for it in the **URL path settings** tab:



Product catalog

Menu settings
Not in menu

Revision information
No revision

URL path settings
Automatic alias

Comment settings
Open

Authoring information
By admin

Publishing options
Published

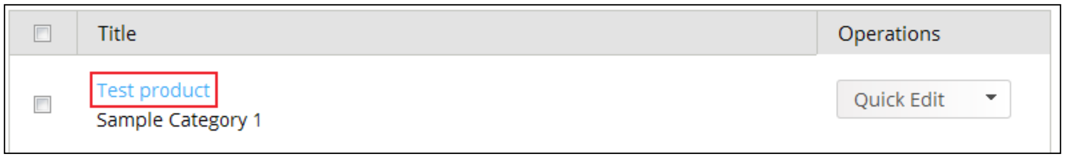
**Product category**

Sample Category 2 ▾

- None -
Sample Category 1
Sample Category 2

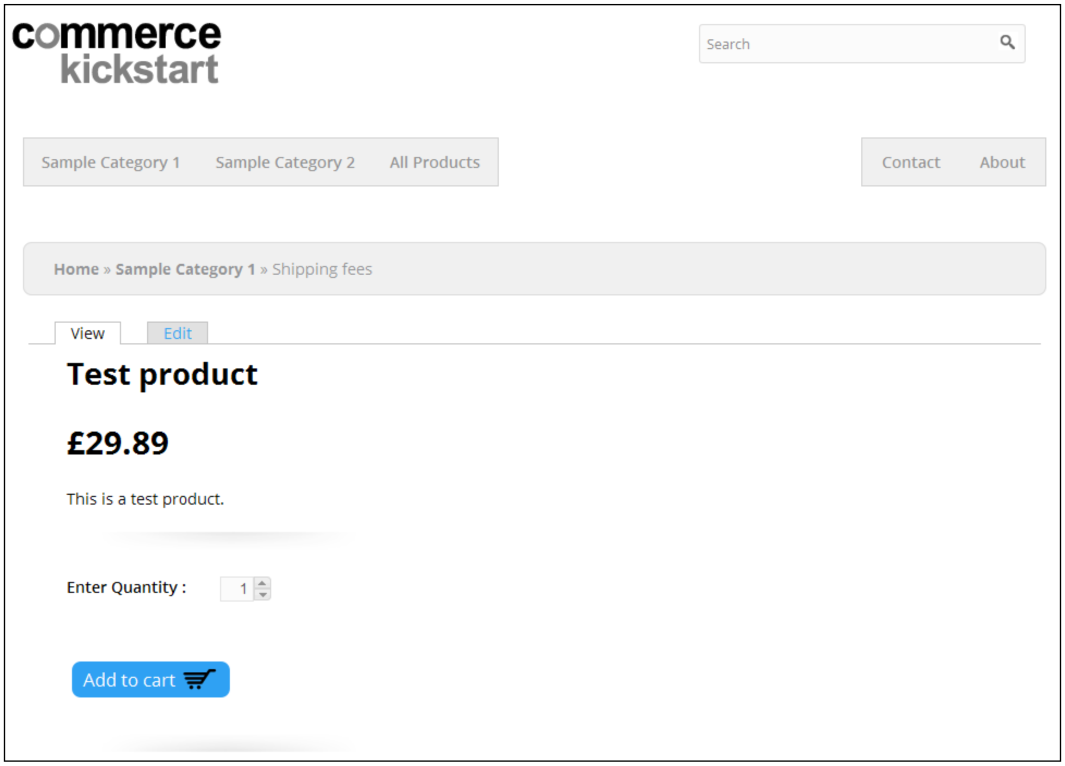
4. Click on the **Save product** button, and you've now created a basic product in your store.

5. To view the product on the frontend of your store, you can navigate to the category listings if you imported Drupal Commerce's demo data, or else you can return to the **Products** menu and click on the name of the product in the **Title** column:



| □ | Title | Operations |
|---|---|---|
| □ | Test product<br>Sample Category 1 | Quick Edit ▾ |

6. You'll now see your product on the frontend of your Drupal Commerce store:



Test product

£29.89

This is a test product.

Enter Quantity : 1

## How it works...

In Drupal Commerce, a product can represent several things, listed as follows:

- A single product for sale (for example, a one-size-fits-all t-shirt)
- A variation of a product (for example, a medium-size t-shirt)
- An item that is not necessarily a purchase as such (for example, it may represent a donation to a charity)
- An intangible product which the site allows reservations for (for example, an event booking)

Product displays (for example, a blue t-shirt) are used to group product variations (for example, a medium-sized blue t-shirt and a large-sized blue t-shirt), and display them on your website to customers.

So, depending on the needs of your Drupal Commerce website, products may be displayed on unique pages, or multiple products might be grouped onto one page as a product display.

# Configuring the checkout in Drupal Commerce

Like any flexible e-commerce platform, Drupal Commerce has many options you can configure to control how your store works. The checkout is a very flexible part of your store, with the ability to change the checkout process to suit your organization's needs.

This recipe will guide you through the basic items you will likely want to configure in your Drupal Commerce store.

## Getting started

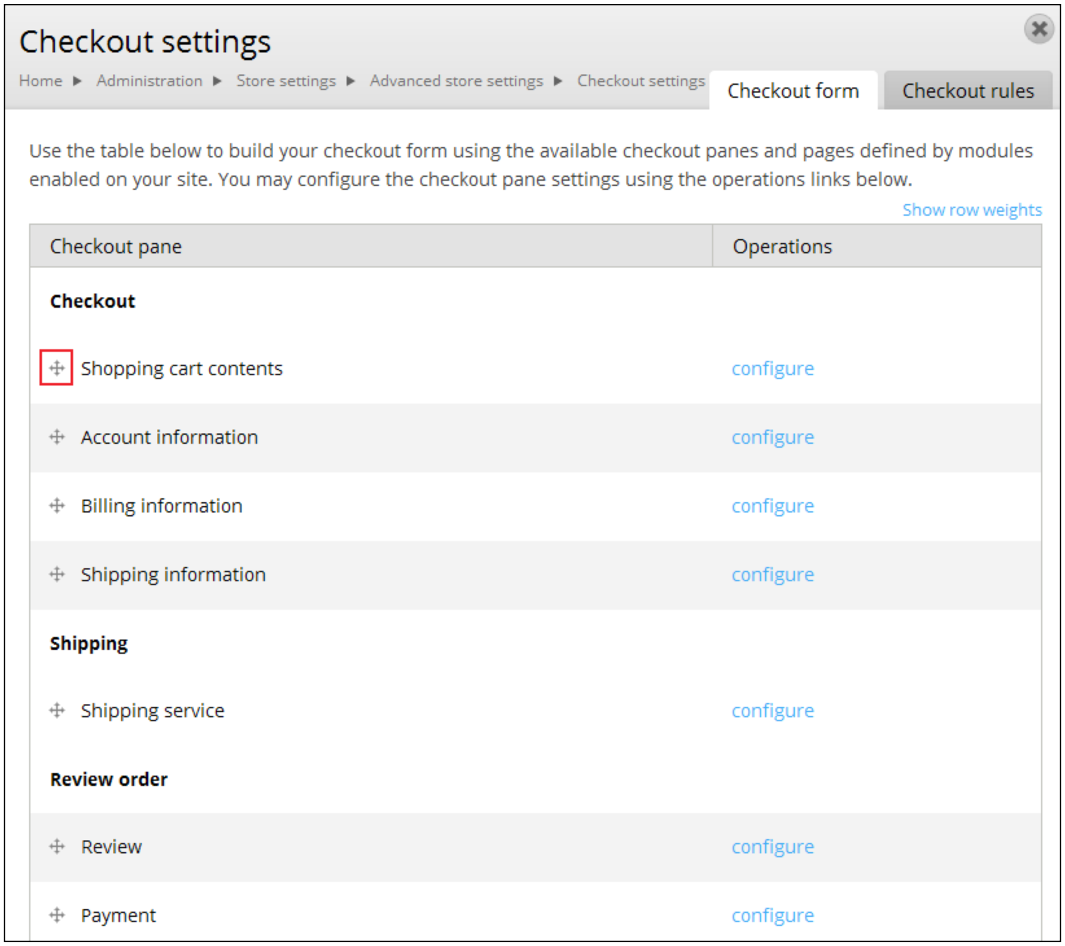
Log in to your Drupal administration panel and navigate to **Store Settings** | **Checkout settings**.

## How to do it...

You can now start customizing the checkout settings for your Drupal Commerce store as follows:

1. The view you can see indicates in which order the various elements of the checkout process appear. Firstly, ensure you have the Commerce Checkout module enabled (navigate to **Site settings** | **Modules** for this).

2. To re-order these, you can drag and drop each stage using the arrows icon to the left-hand side of the screen (highlighted next to the **Shopping cart contents** option as shown in the following screenshot):


Home ▶ Administration ▶ Store settings ▶ Advanced store settings ▶ Checkout settings
Checkout form
Checkout rules
Show row weights

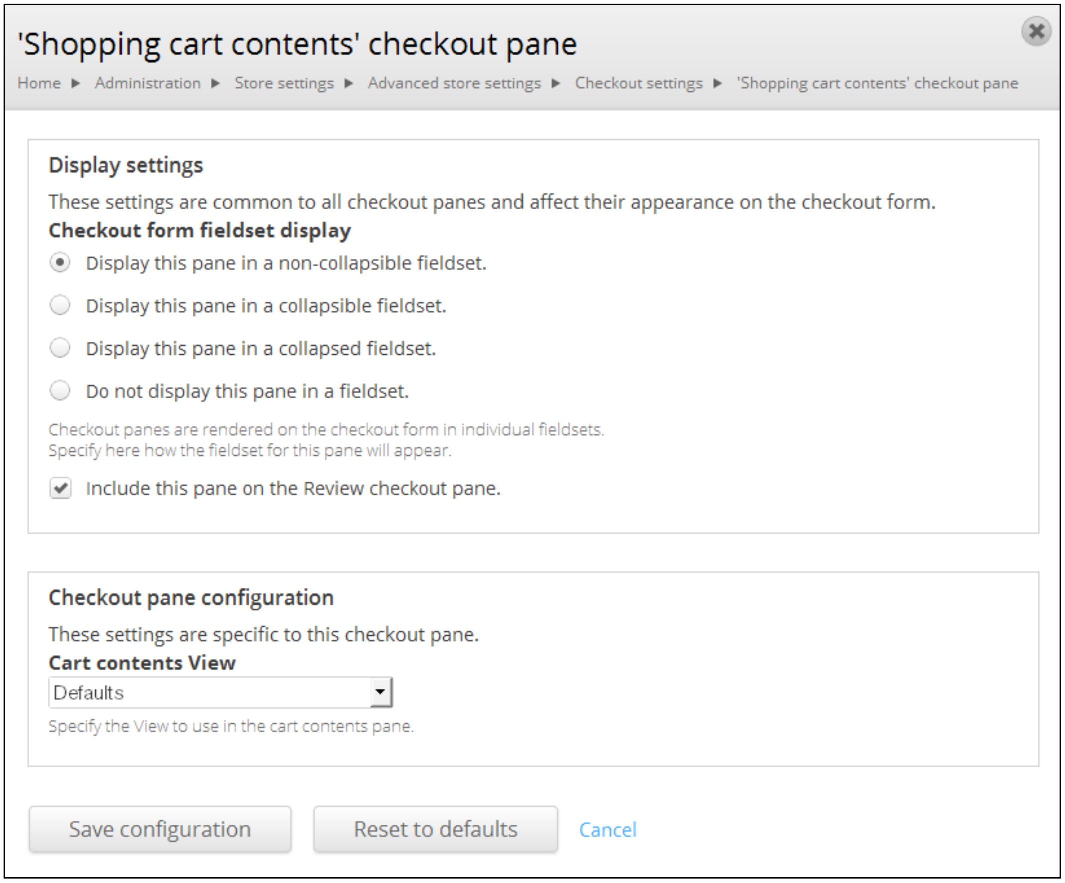
Checkout settings

Use the table below to build your checkout form using the available checkout panes and pages defined by modules enabled on your site. You may configure the checkout pane settings using the operations links below.

| Checkout pane | Operations |
|---|---|
| **Checkout** | |
| ✣ Shopping cart contents | configure |
| ✣ Account information | configure |
| ✣ Billing information | configure |
| ✣ Shipping information | configure |
| **Shipping** | |
| ✣ Shipping service | configure |
| **Review order** | |
| ✣ Review | configure |
| ✣ Payment | configure |

Beware that some configurations could break your store's checkout process or confuse customers. For example, moving the completion message to the first stage of the checkout process would be very misleading!

3. You can also see a **configure** option to the right-hand side of each checkout phase. Select the one next to the **Shopping cart contents** phase, and you will be presented with options on how the checkout phase is displayed:



'Shopping cart contents' checkout pane

Home ▶ Administration ▶ Store settings ▶ Advanced store settings ▶ Checkout settings ▶ 'Shopping cart contents' checkout pane

Display settings

These settings are common to all checkout panes and affect their appearance on the checkout form.

**Checkout form fieldset display**

● Display this pane in a non-collapsible fieldset.

○ Display this pane in a collapsible fieldset.

○ Display this pane in a collapsed fieldset.

○ Do not display this pane in a fieldset.

Checkout panes are rendered on the checkout form in individual fieldsets.
Specify here how the fieldset for this pane will appear.

☑ Include this pane on the Review checkout pane.

Checkout pane configuration

These settings are specific to this checkout pane.

**Cart contents View**

Defaults

Specify the View to use in the cart contents pane.

4. The options beneath the **Display settings** panel control how this panel is displayed to your customers:
    1. **Display this pane in a non-collapsible fieldset**: This option is the default, and displays the checkout phase in full, with no option to collapse it from view.
    2. **Display this pane in a collapsible fieldset**: This option shows the checkout phase in full by default, but provides an option to collapse it from view if the customer prefers.
    3. **Display this pane in a collapsed fieldset**: This option hides the phase from the view of the customer, but allows itself to be viewed in full by clicking on it to enlarge it within the page.
    4. **Do not display this pane in a fieldset**: This option removes the `<fieldset>` element from around the HTML elements used to display this checkout phase, which may be of use when theming your Drupal Commerce store.

5. The **Include this pane on the Review checkout pane** checkbox underneath these options controls whether this block is shown on the checkout overview page, where the customer is required to confirm their order details.

## How it works

Drupal Commerce's checkout process is divided into a series of phases. Each phase can contain one or more panes which can be used to display information (shopping cart content) or to get information from the user (billing address). If a phase doesn't have a pane assigned to it, it will not be displayed during checkout.

It's worth noting that some contributed modules can add settings to panes associated with Drupal Commerce's checkout too, such as the Commerce Addressbook module (see `http://drupal.org/project/commerce_addressbook` for more information on this module).

# Configuring the currency in Drupal Commerce

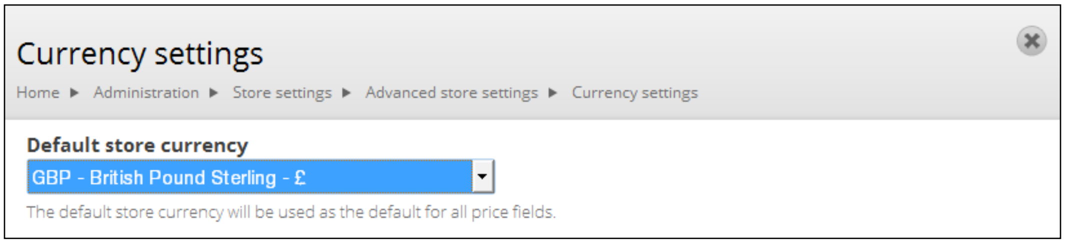
One of the fundamental settings in your Drupal Commerce store is the currency your store uses. This recipe will guide you through changing and configuring the default currency for your Drupal Commerce store.

## Getting started

Log in to your Drupal Commerce store's administration panel and navigate to **Store settings** | **Currency settings**.

## How to do it...

The **Default store currency** setting provides a drop-down menu of currencies you can pick from; select your preferred option here if it has changed since your installation of Drupal Commerce.



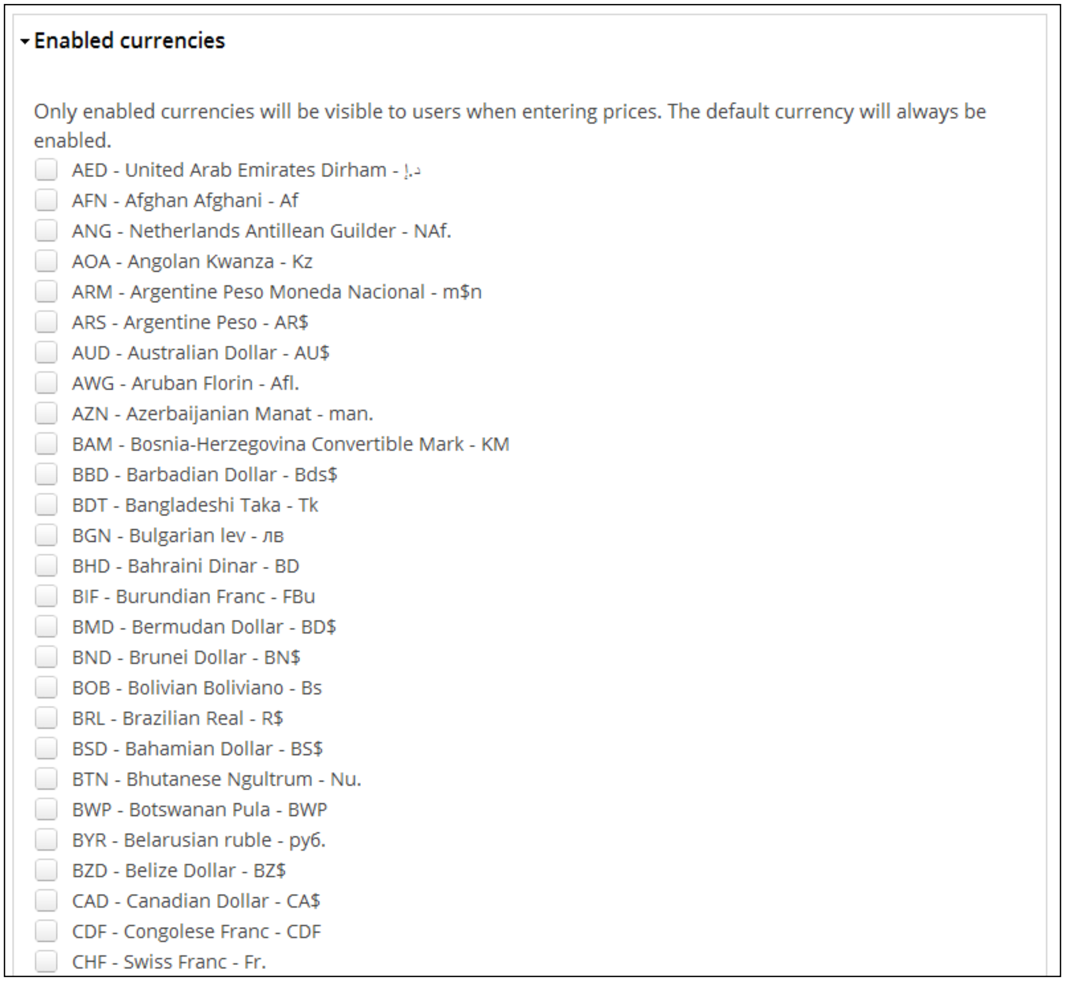
Currency settings

Home ▸ Administration ▸ Store settings ▸ Advanced store settings ▸ Currency settings

Default store currency: GBP - British Pound Sterling - £

The default store currency will be used as the default for all price fields.

The currency settings also allow you to specify additional currencies when creating a product by selecting none, one, or more than one checkboxes, but Drupal Commerce will not convert between currencies by default:



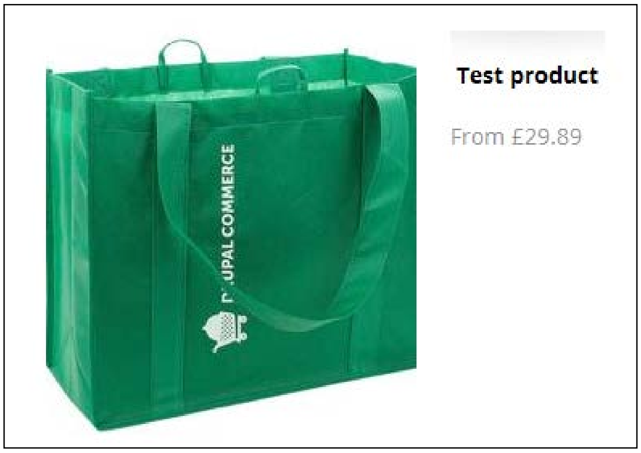
▾ **Enabled currencies**

Only enabled currencies will be visible to users when entering prices. The default currency will always be enabled.

□ AED - United Arab Emirates Dirham - د.إ
□ AFN - Afghan Afghani - Af
□ ANG - Netherlands Antillean Guilder - NAf.
□ AOA - Angolan Kwanza - Kz
□ ARM - Argentine Peso Moneda Nacional - m$n
□ ARS - Argentine Peso - AR$
□ AUD - Australian Dollar - AU$
□ AWG - Aruban Florin - Afl.
□ AZN - Azerbaijanian Manat - man.
□ BAM - Bosnia-Herzegovina Convertible Mark - KM
□ BBD - Barbadian Dollar - Bds$
□ BDT - Bangladeshi Taka - Tk
□ BGN - Bulgarian lev - лв
□ BHD - Bahraini Dinar - BD
□ BIF - Burundian Franc - FBu
□ BMD - Bermudan Dollar - BD$
□ BND - Brunei Dollar - BN$
□ BOB - Bolivian Boliviano - Bs
□ BRL - Brazilian Real - R$
□ BSD - Bahamian Dollar - BS$
□ BTN - Bhutanese Ngultrum - Nu.
□ BWP - Botswanan Pula - BWP
□ BYR - Belarusian ruble - руб.
□ BZD - Belize Dollar - BZ$
□ CAD - Canadian Dollar - CA$
□ CDF - Congolese Franc - CDF
□ CHF - Swiss Franc - Fr.

> You can install a contributed module to automatically convert the currencies with the current conversion rate, or choose to manually fill in the prices in all currencies. See the `Currency Exchange` module at `http://drupal.org/project/currency` for more information.

1. Click on the **Save configuration** button to save your changes.
2. If you view the product on the frontend of your store, as customers would see it, you will see that the cost is displayed in the currency you specified here:



Test product

From £29.89

3. Note the default "Drupal Commerce" image on the bag used as the default placeholder image for products with no photograph.

## How it works...

Changing the default currency on your store changes the symbol shown before or after the price value you set for each product, but does not attempt to convert values between currencies (see the Drupal Currency Exchange module for related functionality). So, if you change the default setting from **United States Dollars (USD)** to **Hong Kong Dollars (HKD)**, the costs you specified when creating the product would remain the same (so $9.99 USD would become $9.99 HKD).

Depending on the currency you have set, Drupal Commerce will change the number of decimal points in the price displayed, as well as whether the currency symbol appears before or after the price on your store.

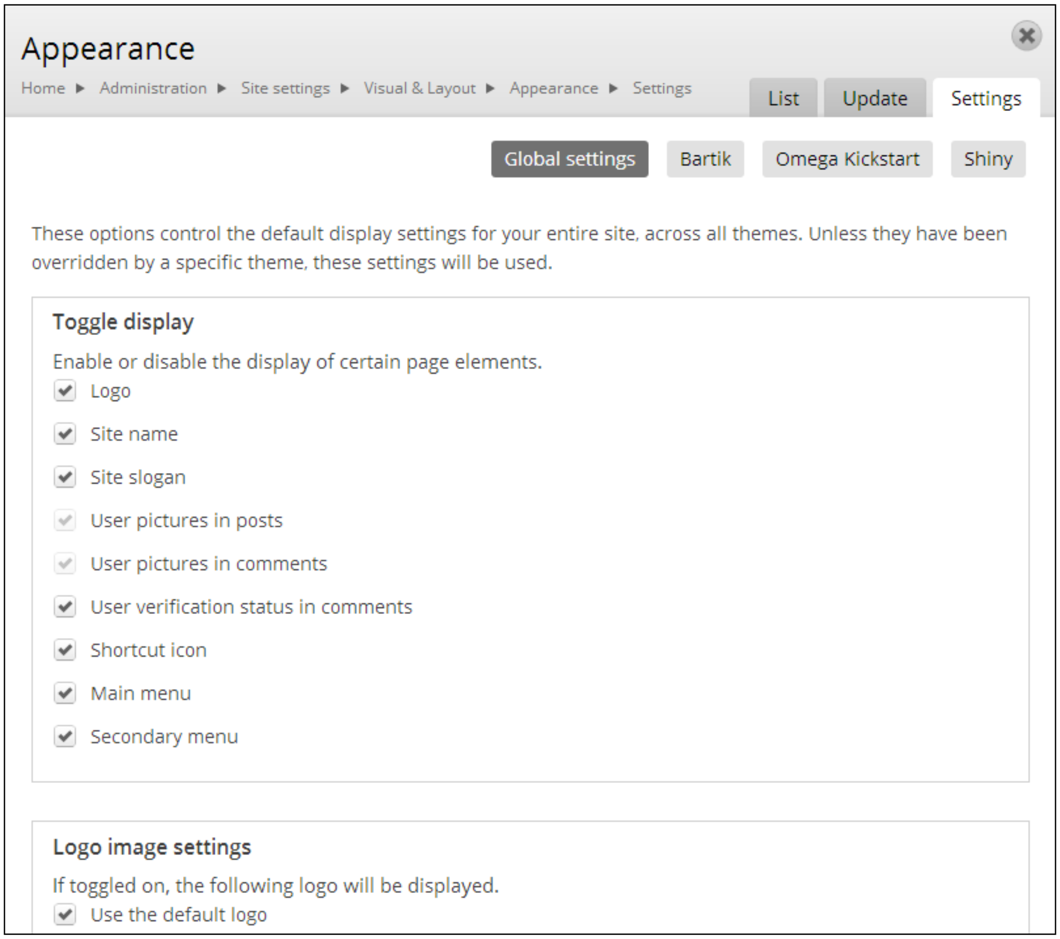
# Changing your store's logo file

One of the most visual elements of your store is your logo, and it can help make a great impression on your customers.

This recipe will show you how to change your Drupal Commerce store's logo.

## Getting started

Log in to your Drupal Commerce control panel, and navigate to **Appearance**, and then to the **Settings** tab at the top right-hand corner of the screen:



Appearance

Home ▶ Administration ▶ Site settings ▶ Visual & Layout ▶ Appearance ▶ Settings

These options control the default display settings for your entire site, across all themes. Unless they have been overridden by a specific theme, these settings will be used.

Toggle display

Enable or disable the display of certain page elements.

☑ Logo

☑ Site name

☑ Site slogan

☑ User pictures in posts

☑ User pictures in comments

☑ User verification status in comments

☑ Shortcut icon

☑ Main menu

☑ Secondary menu

Logo image settings

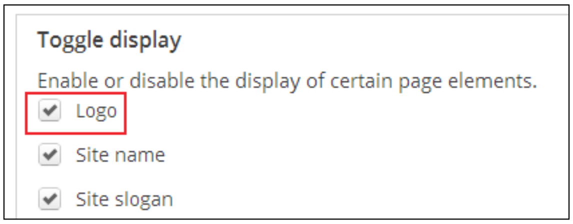
If toggled on, the following logo will be displayed.

☑ Use the default logo

## How to do it...

Now that you've found this screen, you can begin to change your Drupal Commerce store's logo:

1. Ensure you are under the **Global settings** tab; you will be presented with options. Under the **Toggle display** options block, ensure that the **Logo** checkbox is selected:



Toggle display

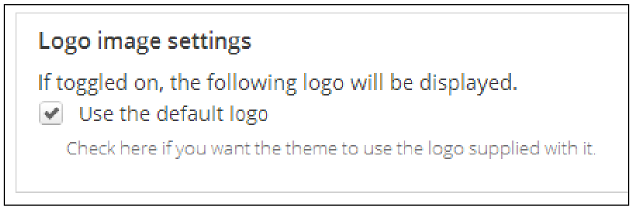
Enable or disable the display of certain page elements.

☑ Logo

☑ Site name

☑ Site slogan

2. Next, scroll down towards the **Logo image settings** block and uncheck the **Use the default logo** checkbox to disable Drupal Commerce's default logo.



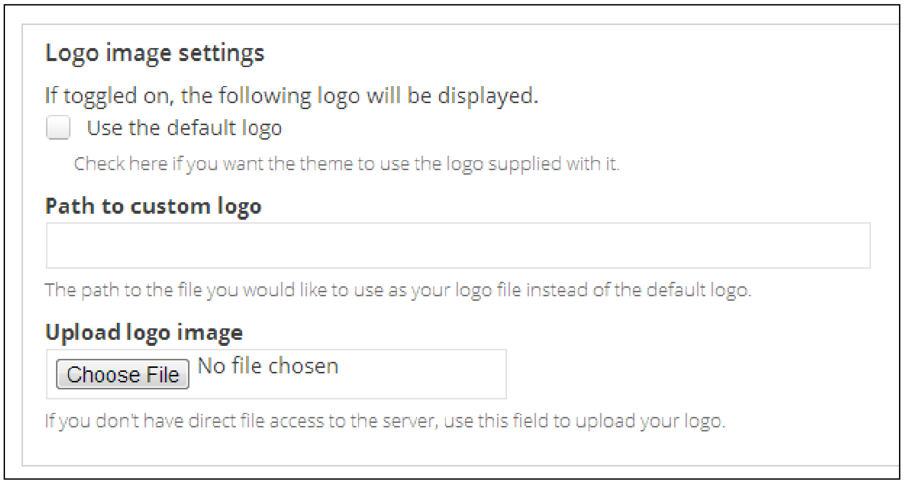
Logo image settings

If toggled on, the following logo will be displayed.

☑ Use the default logo

Check here if you want the theme to use the logo supplied with it.

3. Once unchecked, more options will appear below this: the **Path to custom logo** field, and the **Upload logo image** field.



Logo image settings

If toggled on, the following logo will be displayed.

□ Use the default logo

Check here if you want the theme to use the logo supplied with it.

Path to custom logo

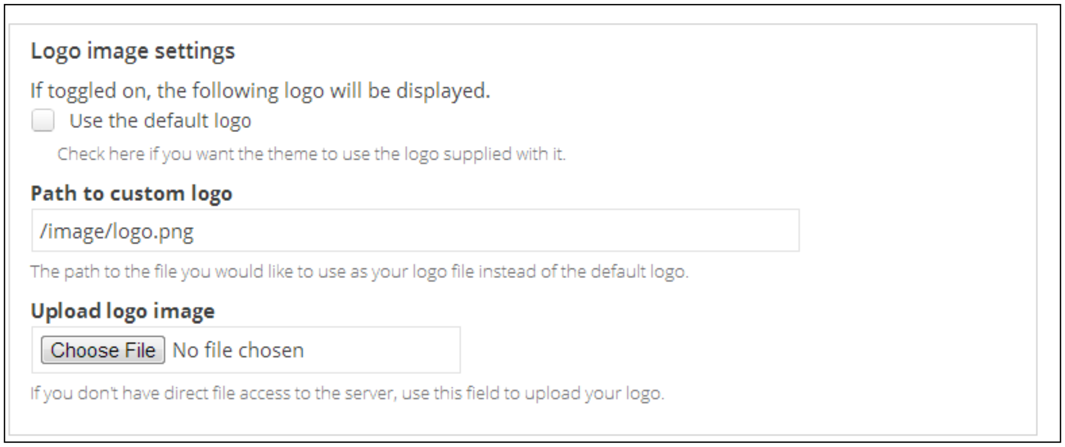
The path to the file you would like to use as your logo file instead of the default logo.

Upload logo image

Choose File No file chosen

If you don't have direct file access to the server, use this field to upload your logo.

4. To use an image that is already on your Drupal Commerce installation's server, simply enter the path to your logo file in the **Path to custom logo** field. Good directories to store your store's logo in could be `image/` or `sites/default/files/images/`.



Logo image settings

If toggled on, the following logo will be displayed.

□ Use the default logo

Check here if you want the theme to use the logo supplied with it.

Path to custom logo

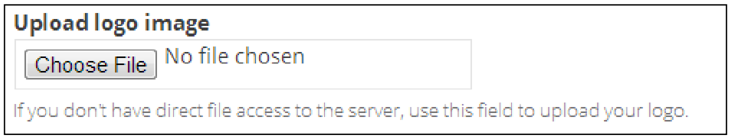
/image/logo.png

The path to the file you would like to use as your logo file instead of the default logo.

Upload logo image

If you don't have direct file access to the server, use this field to upload your logo.

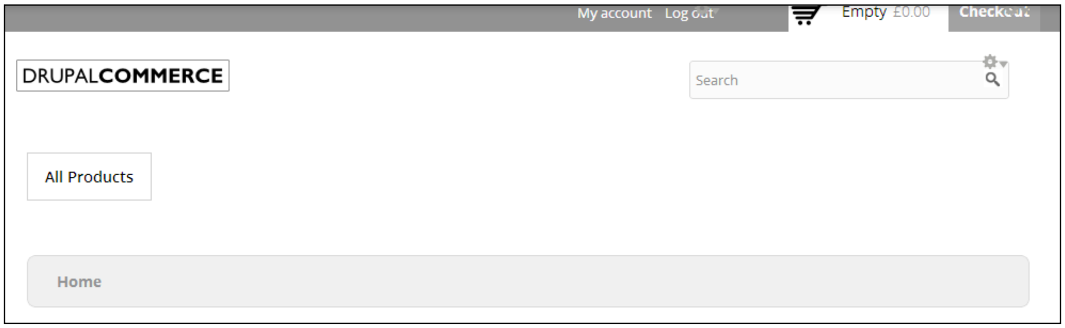
5. Click on the **Save configuration** button to retain your new logo.
6. To upload a logo from your computer to your Drupal Commerce site, use the **Upload logo image** field and click on the **Save configuration** button to save this choice of logo. Ensure that the **Path to custom logo** field is empty before saving, as otherwise it will produce an error. Either **Path to custom logo** or the **Upload logo image** field can be used, but you cannot use both together.



Upload logo image

If you don't have direct file access to the server, use this field to upload your logo.

7. If you now view the frontend of your Drupal Commerce store, you will see the new logo has appeared:

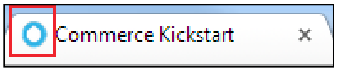


# Changing your store's favicon (favorites icon)

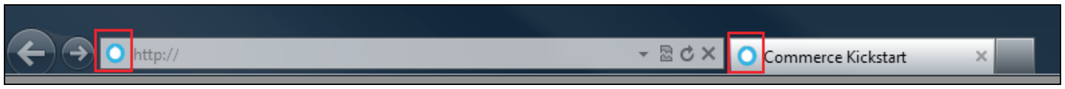
Alongside your logo, your store also has a favorites icon, also known as a **favicon** or a shortcut icon, which is a small graphic you can use to distinguish your store from other websites in a number of contexts, including in lists of the visitor's bookmarked websites.

## Getting started

By default, the favicon image is Drupal Commerce's logo. To see it, visit your Drupal Commerce store website in a browser and you will see it appear in one of a number of positions depending on your browser. In Chrome, the favicon appears in the tab next to the web page's name:



In Internet Explorer 9, the favorites icon appears both in the page's tab and in the address bar:



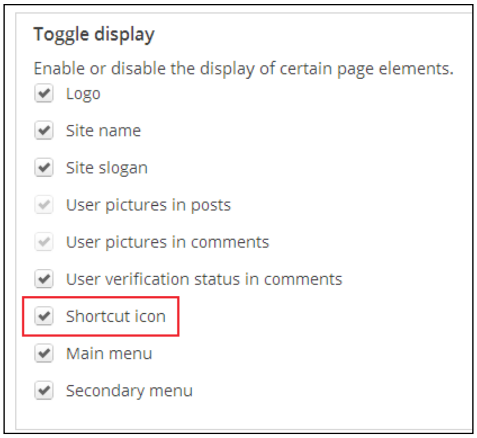
You will need to create a graphic for your favorites icon. There are a number of online favicon generators you can use, such as Dynamic Drive's tool at `http://tools.dynamicdrive.com/favicon/`, which can create a favicon from JPEG, GIF, and PNG formats.

Ensure you are logged in to your store's administration panel.

## How to do it...

To change your Drupal Commerce store's favicon, follow these steps:

1. Navigate to **Appearance** and click on the **Settings** tab towards the top right-hand corner of the screen.
2. Ensure that the **Shortcut icon** checkbox is enabled in the **Toggle display** settings:

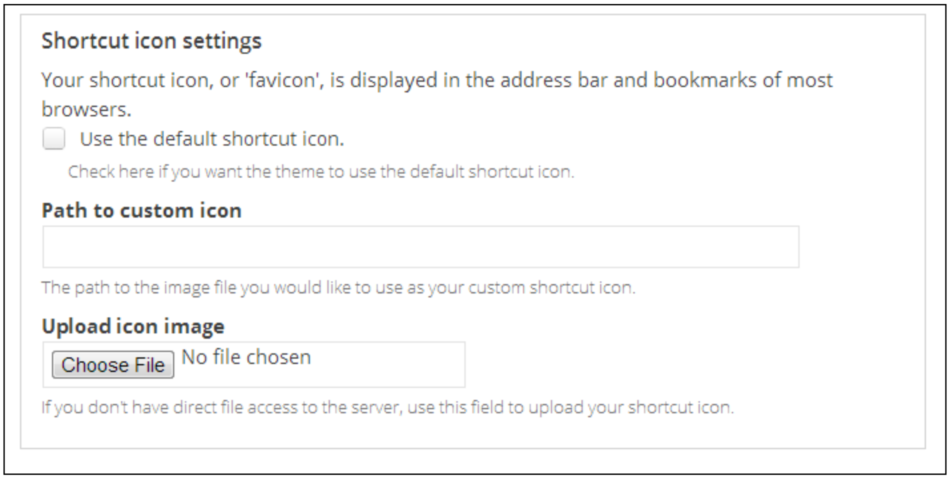


Toggle display

Enable or disable the display of certain page elements.

- ☑ Logo
- ☑ Site name
- ☑ Site slogan
- ☑ User pictures in posts
- ☑ User pictures in comments
- ☑ User verification status in comments
- ☑ Shortcut icon
- ☑ Main menu
- ☑ Secondary menu

3. Now scroll down, and you will see the **Shortcut icon settings** field. Uncheck the **Use the default shortcut icon** field and you are presented with two options: **Path to custom icon** and **Upload icon image**.



Shortcut icon settings

Your shortcut icon, or 'favicon', is displayed in the address bar and bookmarks of most browsers.

☐ Use the default shortcut icon.

Check here if you want the theme to use the default shortcut icon.

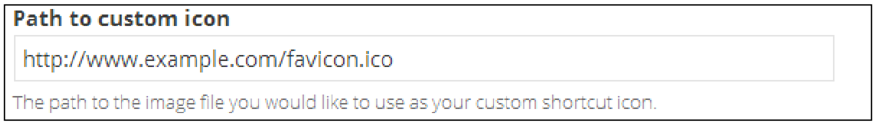
**Path to custom icon**

The path to the image file you would like to use as your custom shortcut icon.

**Upload icon image**

If you don't have direct file access to the server, use this field to upload your shortcut icon.

4. To use an icon file already on your server, use the **Path to custom icon** field, enter the relative URL (that is, with the `http://www.yourstore.com` part). So, `sites/default/files/favicon.ico` or simply `favicon.ico` are good values for this field.

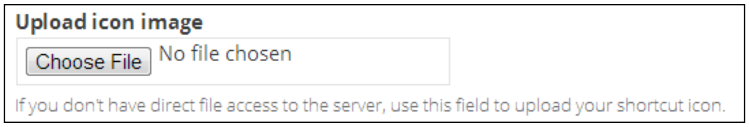


**Path to custom icon**

http://www.example.com/favicon.ico

The path to the image file you would like to use as your custom shortcut icon.

5. To upload a file on your computer for use as the favicon for your Drupal Commerce store, use the **Upload icon image** field. Ensure that the **Path to custom icon** field is empty before saving this, otherwise your icon will not be saved and you'll see an error message.

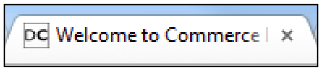


Upload icon image

Choose File No file chosen

If you don't have direct file access to the server, use this field to upload your shortcut icon.

6. Click on the **Save configuration** button to save your new favorites icon.
7. To see your new favicon in action, navigate to the frontend of your store and refresh:



> You may need to do a hard refresh of your website to see the favicon. Pressing *Ctrl* + *F5* usually works in most browsers (or *Cmd* + *R* on Mac).

8. That's it! Your new favicon is in place.

# Adding social media icons on product pages

With the popularity of social networks such as Twitter, Facebook, Google+, and Pinterest remaining strong, integrating tools to help your customers share your products with their friends can be an excellent way to grow your customer base.

This recipe will show you how to customize your Drupal Commerce store to include social media share buttons on your store's product pages.

## Getting started

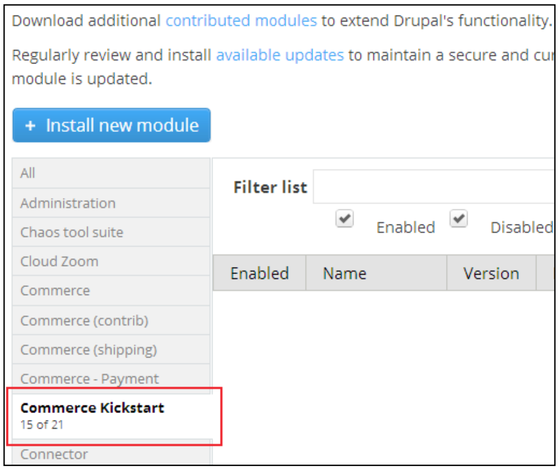
Log in to your Drupal Commerce administration panel with your administrator account.

> If you didn't use Commerce Kickstart to install Drupal Commerce, you will need to.

## How to do it...

To add your chosen social networks to your Commerce product pages, you need to use a module package with Commerce Kickstart:

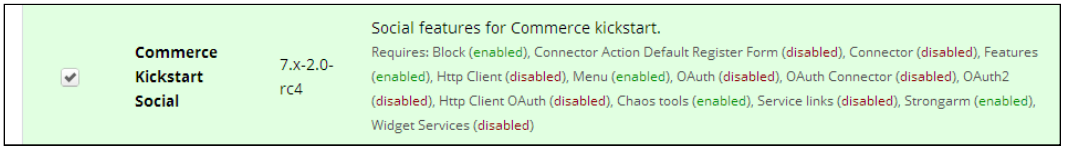
1. Navigate to **Modules** in your administration panel. In the menu in the left-hand column, you'll see a tab for **Commerce Kickstart** modules. Click on this to view a list of available Commerce Kickstart modules.



Download additional contributed modules to extend Drupal's functionality.

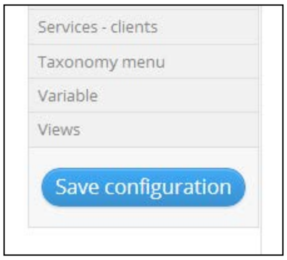
Regularly review and install available updates to maintain a secure and cu module is updated.

2. Scroll down the list of modules until you come to the **Commerce Kickstart Social** module, and enable it by checking the box:

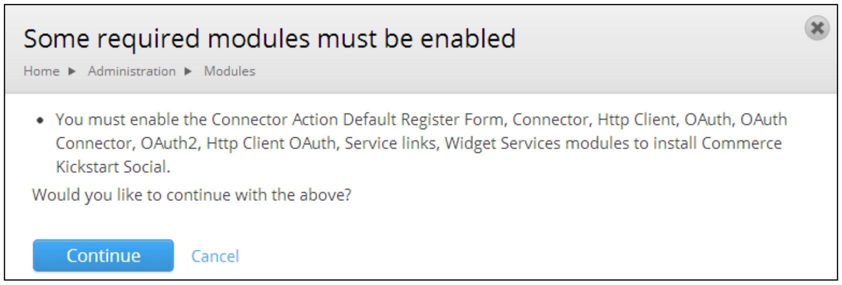


| | | | |
|---|---|---|---|
| ☑ | **Commerce Kickstart Social** | 7.x-2.0-rc4 | Social features for Commerce kickstart.<br>Requires: Block (enabled), Connector Action Default Register Form (disabled), Connector (disabled), Features (enabled), Http Client (disabled), Menu (enabled), OAuth (disabled), OAuth Connector (disabled), OAuth2 (disabled), Http Client OAuth (disabled), Chaos tools (enabled), Service links (disabled), Strongarm (enabled), Widget Services (disabled) |

3. Click on the **Save configuration** button in the left-hand column to enable the module; it is located under the list of module groupings:



4. If you haven't enabled one or more of the modules required by the Kickstart Commerce Social module, you'll be prompted to enable them on the next screen. Select **Continue**:



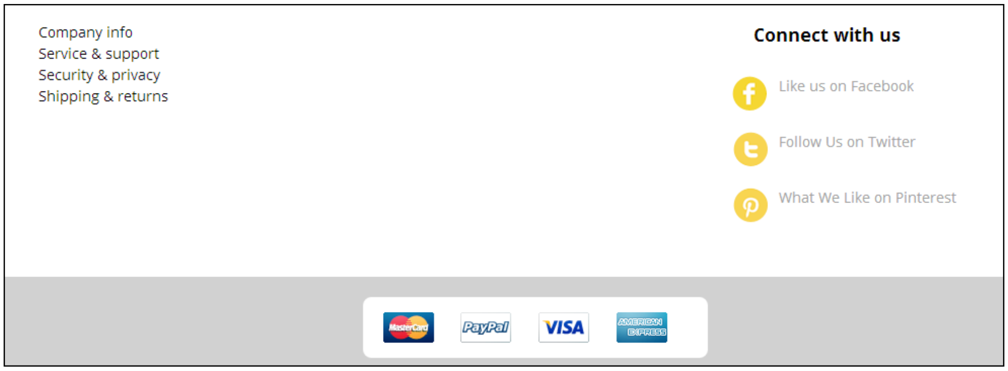
Some required modules must be enabled

Home ▸ Administration ▸ Modules

- You must enable the Connector Action Default Register Form, Connector, Http Client, OAuth, OAuth Connector, OAuth2, Http Client OAuth, Service links, Widget Services modules to install Commerce Kickstart Social.

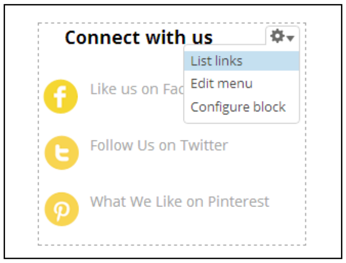
Would you like to continue with the above?

5. Enabling the module does two things: it adds a block to your store's footer with links to various social media platforms, and it adds share tools on your product pages. View the frontend of your website and look in the footer, and you will see them as follows:

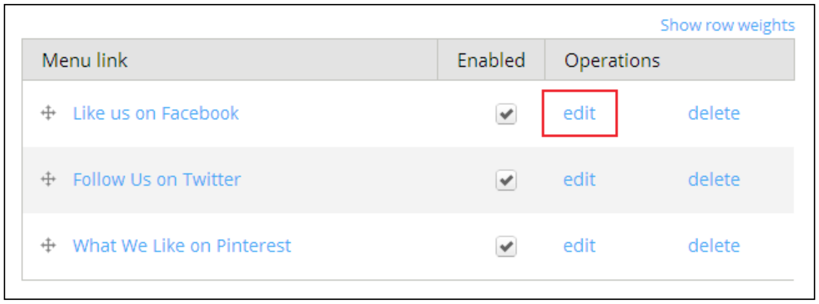


If you can't see these appear, you may need to assign them.

6. By default, these link to Commerce Guys' profiles, so ensuring you are logged in as an administrator, hover over the block and then select the **List links** option from the drop-down menu:



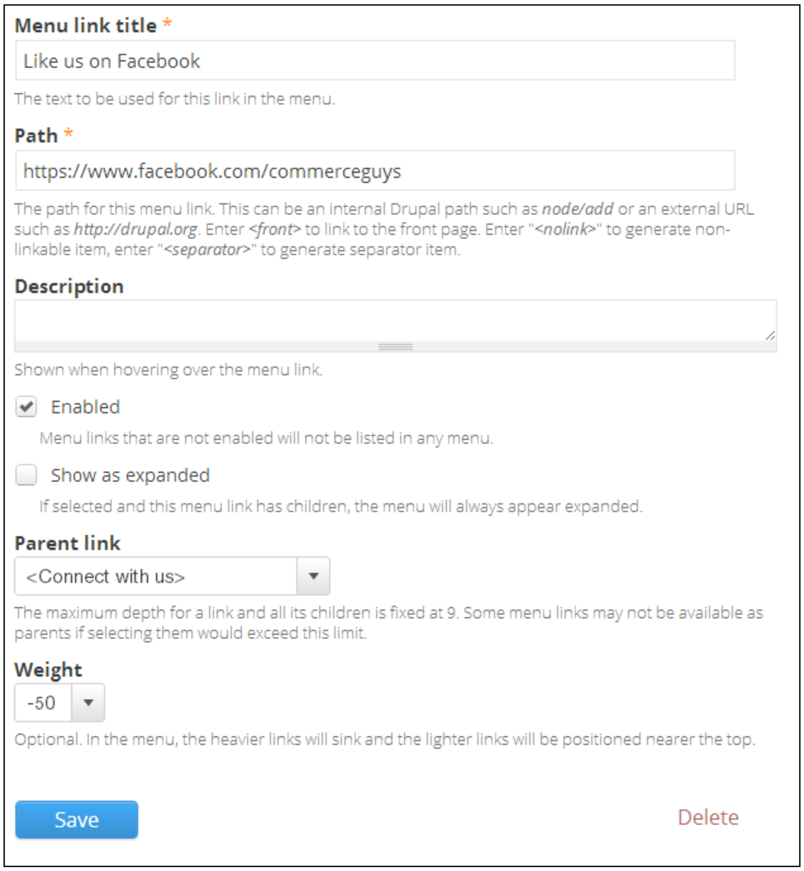
7. This will present you with a list of the links. Select **edit** in the **Operations** column:



Show row weights

| Menu link | Enabled | Operations | |
|---|---|---|---|
| Like us on Facebook | ☑ | edit | delete |
| Follow Us on Twitter | ☑ | edit | delete |
| What We Like on Pinterest | ☑ | edit | delete |

8. You can now edit the URL of your social media profile with the **Path** field and the label for the link with the **Menu link title** field:



**Menu link title** *

Like us on Facebook

The text to be used for this link in the menu.

**Path** *

https://www.facebook.com/commerceguys

The path for this menu link. This can be an internal Drupal path such as *node/add* or an external URL such as *http://drupal.org*. Enter *<front>* to link to the front page. Enter "*<nolink>*" to generate non-linkable item, enter "*<separator>*" to generate separator item.

**Description**

Shown when hovering over the menu link.

☑ Enabled

Menu links that are not enabled will not be listed in any menu.

☐ Show as expanded

If selected and this menu link has children, the menu will always appear expanded.

**Parent link**

<Connect with us>

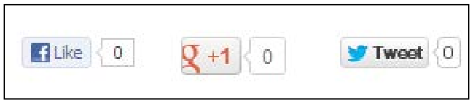
The maximum depth for a link and all its children is fixed at 9. Some menu links may not be available as parents if selecting them would exceed this limit.

**Weight**

-50

Optional. In the menu, the heavier links will sink and the lighter links will be positioned nearer the top.

9. Click on the **Save** button at the bottom of the screen once you are finished.
10. Return to the frontend of your store and navigate to a product page, and you will see the share tools from Facebook, Google+, and Twitter:



# 2

# Configuring Products with Drupal Commerce

In this chapter, we will cover the following topics:

- Adding multiple images to a product
- Creating products with a file upload field
- Creating products with a custom text field
- Creating products with custom attributes
- Creating product categories in Drupal Commerce

## Introduction

Drupal Commerce is incredibly flexible, but this flexibility can be overwhelming for beginners when they start adding and customizing products to their store. In the previous chapter, you learned how to add a basic product to your Drupal Commerce store, and now you can begin to customize this product type to suit your needs, from file uploads to additional product images, and beyond.

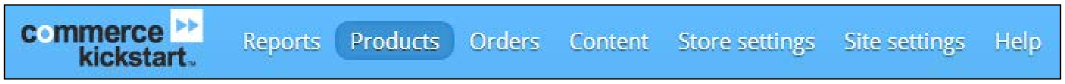
## Adding multiple images to a product

A feature of many e-commerce stores is to display more than one photograph of a product. For example, you may want to display your products from different angles, or display your product on an isolated background and in use. Drupal Commerce Kickstart 2 allows you to add multiple images to your product pages through Drupal's administration panel.

This recipe will guide you through the customization of Drupal Commerce product types so you can add more than one product image per product.

## Getting started

Log in as an administrator to your Drupal Commerce store. Navigate to **Products**. This assumes that you have enabled the Commerce Kickstart Menu's module.

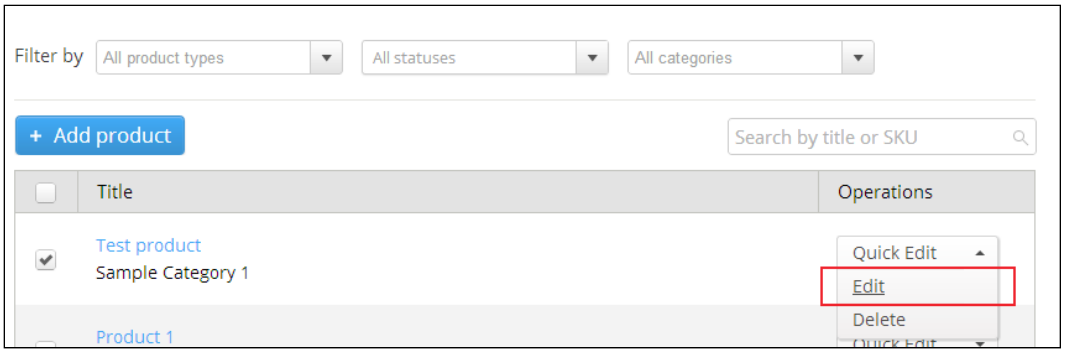


The **Product** tab displays the list of existing products within your Drupal Commerce store.

## How to do it...

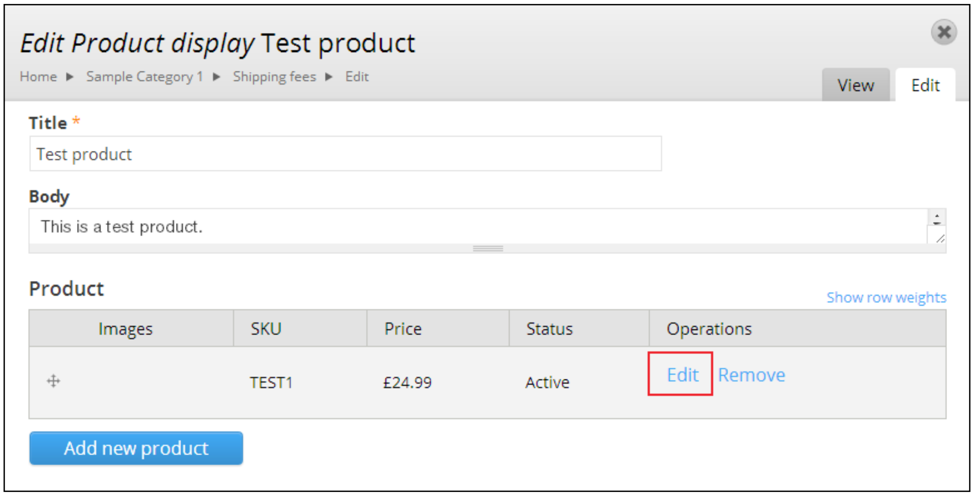
Drupal Commerce Kickstart 2 allows you to add multiple images to your product pages through Drupal's administration panel:

1. Select the **Edit** option from the drop-down next to the product you wish to add multiple images to:



| ☐ | Title | Operations |
|---|---|---|
| ☑ | Test product<br>Sample Category 1 | Quick Edit |
| ☐ | Product 1 | Quick Edit |

2. This takes you to the **Edit Product display** view. Beneath the product display's **Title** and **Body** fields, you will see the list of Drupal Commerce products assigned to this product display; again, select the **Edit** option next to the product you want to add an image to:



*Edit Product display* Test product

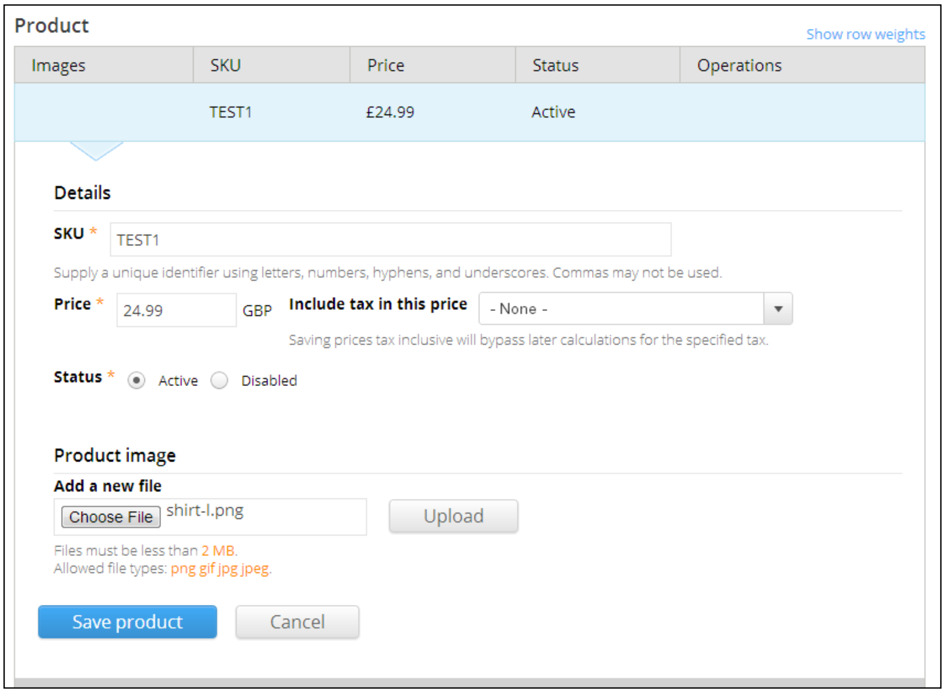
Title *: Test product

Body: This is a test product.

Product

| Images | SKU | Price | Status | Operations |
|---|---|---|---|---|
|  | TEST1 | £24.99 | Active | Edit Remove |

3. The previous step will expand the product's details for you. Here, you'll see a block called **Product image**, under which you will see the **Add a new file** form:



Product

| Images | SKU | Price | Status | Operations |
|---|---|---|---|---|
|  | TEST1 | £24.99 | Active |  |

Details

SKU *: TEST1

Supply a unique identifier using letters, numbers, hyphens, and underscores. Commas may not be used.

Price *: 24.99 GBP Include tax in this price: - None -

Saving prices tax inclusive will bypass later calculations for the specified tax.

Status *: ● Active ○ Disabled

Product image

Add a new file: shirt-l.png

Files must be less than 2 MB.
Allowed file types: png gif jpg jpeg.

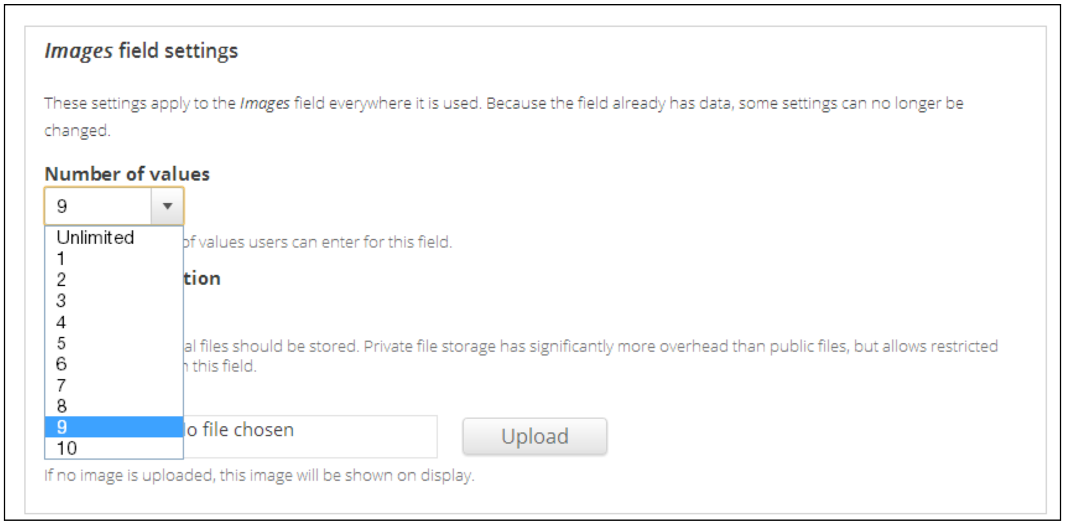
4. For each image you add to the product, click on the **Upload** button next to the **Add a new file** field.

5. Click on the **Save product** button at the bottom of the screen once you're finished adding multiple product images.

6. You can see the multiple images on the frontend of your store now. Simply navigate to a product page for the product that you've just edited.

7. You can customize the settings for images in products by navigating to **Store settings** | **Product settings** | **Variation types** | **Product** | **Manage fields** | **Images**. Then, click on the **Edit** button next to the product variation type that you want to edit.

8. In particular, you can customize how many images per product are permitted with the **Number of values** field in the **Images field** settings block:



*Images* field settings

These settings apply to the *Images* field everywhere it is used. Because the field already has data, some settings can no longer be changed.

Number of values

9

of values users can enter for this field.

tion

al files should be stored. Private file storage has significantly more overhead than public files, but allows restricted

n this field.

o file chosen

If no image is uploaded, this image will be shown on display.

## How it works...

Drupal Commerce products are defined by a number of fields, such as the product title, description, and product image(s). By making use of these predefined field types, you can customize Drupal Commerce products to your needs.

# Creating products with a file upload field

As you have seen, customizing Drupal Commerce to allow additional fields for your products is relatively straightforward, once you know how. A common feature of websites that sell customized items—from t-shirts to stationery and everything in between—is to allow customers to print their picture or logo on the product, and this recipe covers how to go about achieving this for your Drupal Commerce store.

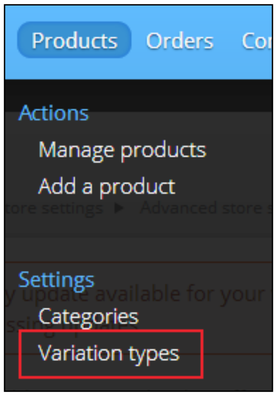
## Getting started

Make sure you're logged in to your Drupal Commerce store as the administrator.

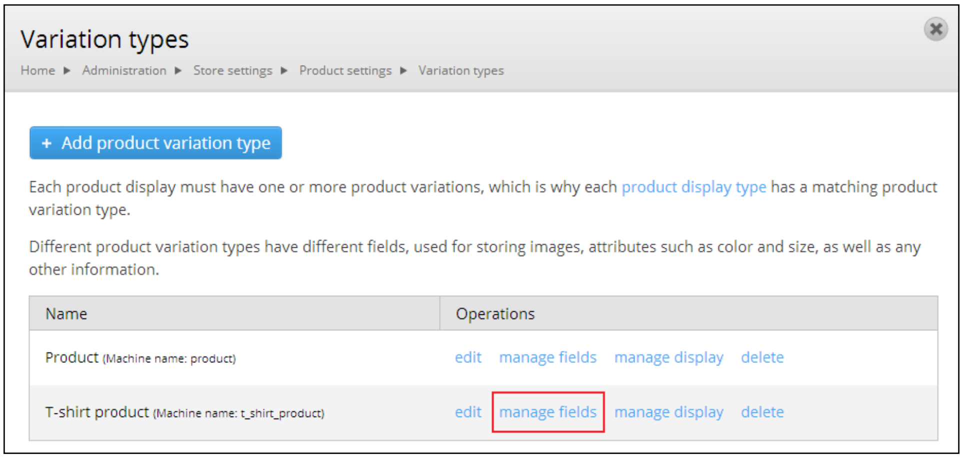
## How to do it...

Once you are logged in to your Drupal Commerce store administration panel, follow these steps:

1. Navigate to **Products** | **Variation types**:



2. From here, you'll be presented with a list of your product types.
3. Select the **manage fields** option in the **Operations** column to the right for the product type to which you wish to add the file upload field. The following example uses the **T-shirt product (Machine name t_shirt_product)** type:



Variation types

Home ▸ Administration ▸ Store settings ▸ Product settings ▸ Variation types

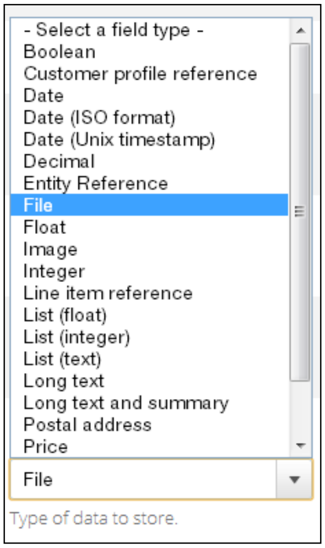
+ Add product variation type

Each product display must have one or more product variations, which is why each product display type has a matching product variation type.

Different product variation types have different fields, used for storing images, attributes such as color and size, as well as any other information.

| Name | Operations |
|---|---|
| Product (Machine name: product) | edit manage fields manage display delete |
| T-shirt product (Machine name: t_shirt_product) | edit manage fields manage display delete |

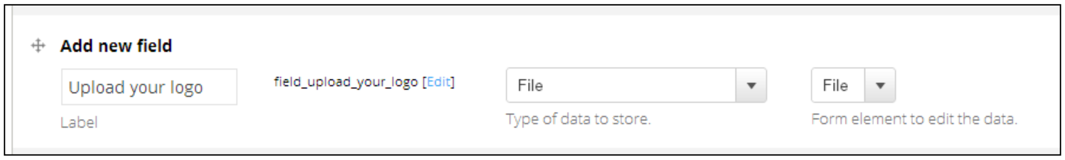
4. Towards the end of the list in the **Manage fields** screen, you will see an option for adding a new field:



File

Type of data to store.

5. Provide a name for your field in the **Label** text input, and then select the **File** type from the **Type of data to store** field. This will automatically populate the **Form element to edit the data** field to **File**:



Upload your logo

Label

field_upload_your_logo [Edit]

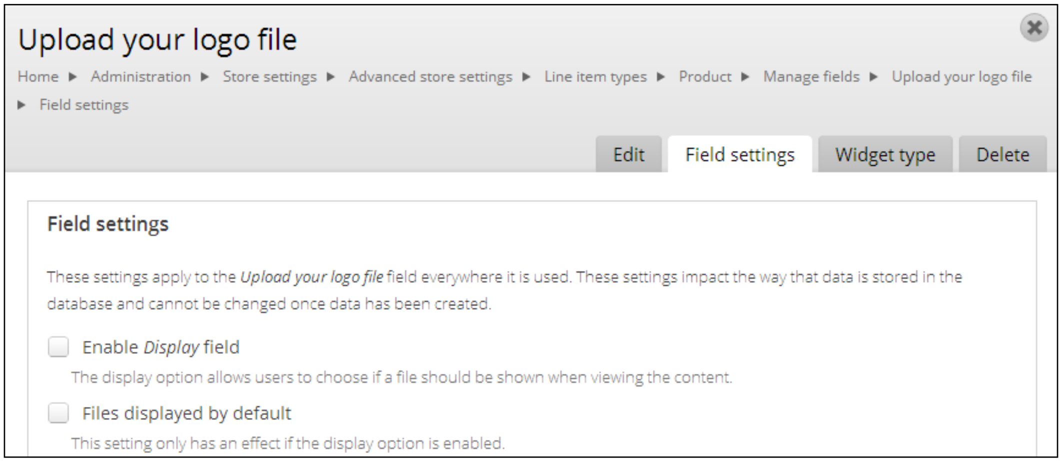
File

Type of data to store.

File

Form element to edit the data.

6. Click on the **Save** button at the bottom of the screen to save your new file upload field. You will now see a **Field settings** screen:



Upload your logo file

Home ▶ Administration ▶ Store settings ▶ Advanced store settings ▶ Line item types ▶ Product ▶ Manage fields ▶ Upload your logo file ▶ Field settings

Field settings

These settings apply to the *Upload your logo file* field everywhere it is used. These settings impact the way that data is stored in the database and cannot be changed once data has been created.

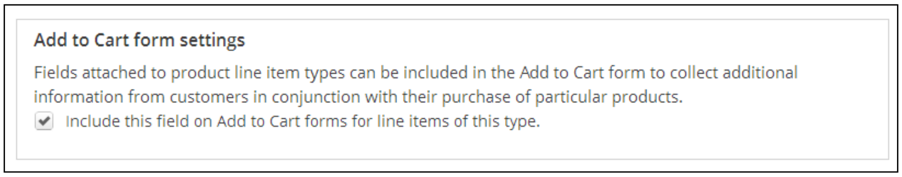
□ Enable *Display* field

The display option allows users to choose if a file should be shown when viewing the content.

□ Files displayed by default

This setting only has an effect if the display option is enabled.

7. You are presented with two options here—**Enable Display field** and **Files displayed by default**—you can ignore both of these settings.
8. Finally, to get the field to display on the product pages, you will need to return to the **Edit** tab. Scroll down to the **Add to Cart form settings** option and check the **Include this field on Add to Cart forms for line items of this type** checkbox:



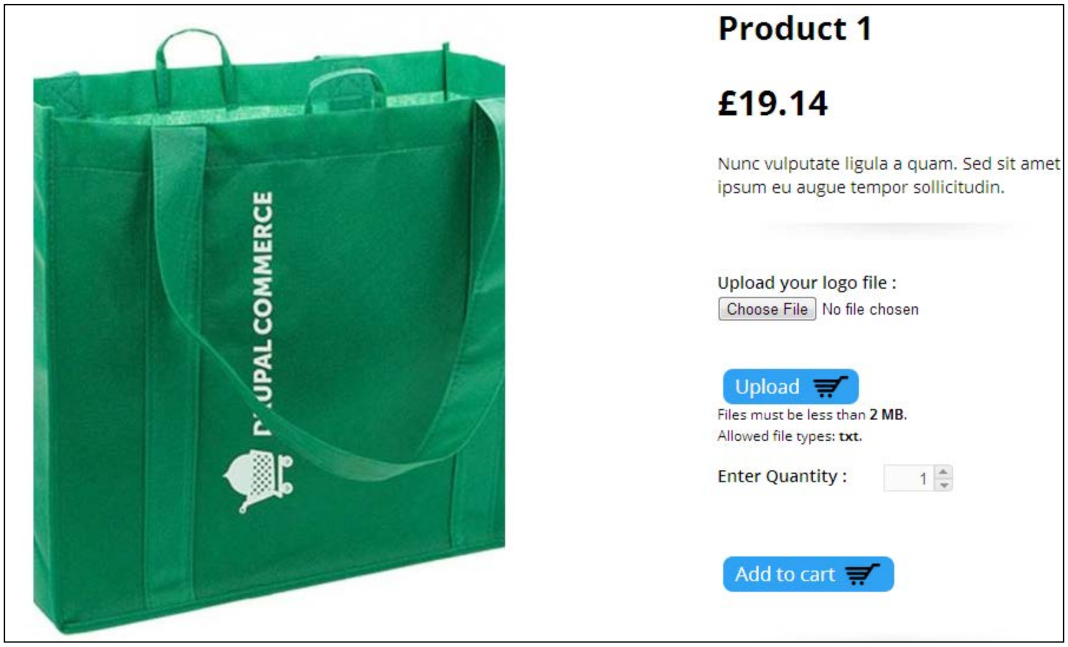
Add to Cart form settings

Fields attached to product line item types can be included in the Add to Cart form to collect additional information from customers in conjunction with their purchase of particular products.

☑ Include this field on Add to Cart forms for line items of this type.

9. Click on the **Save settings** button once you're ready.

10. If you now view a product of the type—remember, this example used the product (machine name product) type—on the frontend of your Drupal Commerce store, you'll see that the field has been added to the product view pages:



**Product 1**

**£19.14**

Nunc vulputate ligula a quam. Sed sit amet ipsum eu augue tempor sollicitudin.

Upload your logo file :

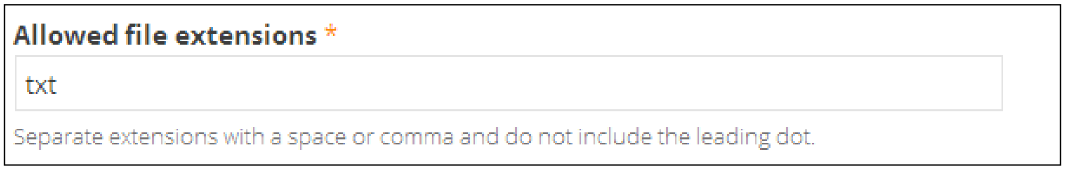
No file chosen

Files must be less than **2 MB**.
Allowed file types: **txt**.

Enter Quantity : 1

11. You can customize the size of files that customers can upload and the types of files that customers can upload under **Store settings** | **Line item** types, by clicking on the **manage** field next to the relevant product type, and then clicking on the **edit** button next to the file upload field you just created.

12. Under the **Product settings** block, you will see the **Allow file extensions** field:



Allowed file extensions *

txt

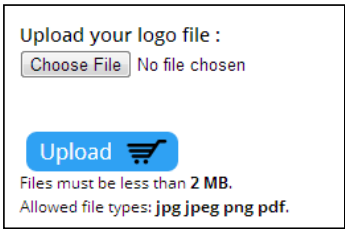
Separate extensions with a space or comma and do not include the leading dot.

13. To allow customers to upload the JPG, PNG and PDF files with their order, you would enter the following in this field:

```
jpg, jpeg, png, pdf
```

> Note that you do not need to enter the period before the filename (for example .jpg), but you need to use commas to separate multiple values here.

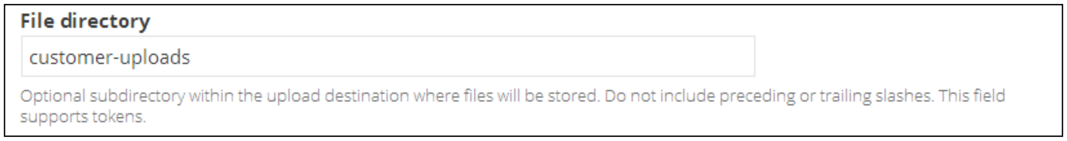
14. Once you have saved this setting, customers will see the new file types that they're able to upload:



Upload your logo file :

Files must be less than **2 MB**.
Allowed file types: **jpg jpeg png pdf**.

15. Files are saved in the `sites/default/files/` directory of your Drupal Commerce installation by default. You can alter the directory to which these files are saved with the **File directory** field when you edit your file upload field:



**File directory**

customer-uploads

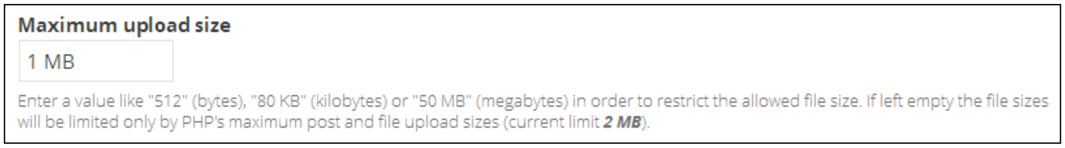
Optional subdirectory within the upload destination where files will be stored. Do not include preceding or trailing slashes. This field supports tokens.

> Remember that the `/sites/default/files` directory needs to be writable to allow Drupal Commerce to save files there!

16. Adding a value of the customer uploads here would save the uploaded files from this field to the `sites/default/files/customer-uploads` directory.

> Note that you do not need to include slash characters (/ or \) at the beginning or end of this field's value, but if you wanted to save uploaded files in the `sites/default/files/customer-uploads/another-directory` directory, you would enter the value `customer-uploads/another-directory`, with the slash character in the middle of the value.

17. Finally, Drupal Commerce gives you the flexibility to define the size of the file that customers are able to upload from this field. Look for the **Maximum upload size** field under the **File directory** field, and enter your desired value:



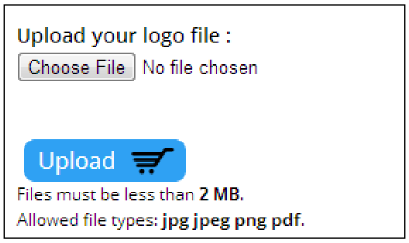
**Maximum upload size**

1 MB

Enter a value like "512" (bytes), "80 KB" (kilobytes) or "50 MB" (megabytes) in order to restrict the allowed file size. If left empty the file sizes will be limited only by PHP's maximum post and file upload sizes (current limit ***2 MB***).

18. Follow the notes under the **Maximum upload size** field on the syntax to add file sizes in different units. Simply do the following:
    - Enter a number without a unit to specify the maximum file upload size in bytes
    - Append KB to your value to specify a value in kilobytes
    - Append MB to your value to specify a value in megabytes

19. Click on the **Save settings** button towards the bottom of the screen to save these settings, and you will see the changes reflected on relevant product pages in your store:



Upload your logo file :

No file chosen

Files must be less than **2 MB**.

Allowed file types: **jpg jpeg png pdf**.

That's it! Your products now have the ability for customers to upload a file alongside them.

# Creating products with a custom text field

Certain types of products require a comments field to allow customers to specify a particular request alongside their order. For example, you might want customers to specify a gift message for their order, or guidance with the artwork that they're uploading to be printed on one of your products.

By adding a text field to products in Drupal Commerce, you can allow your customers to add their own thoughts or comments to your order.

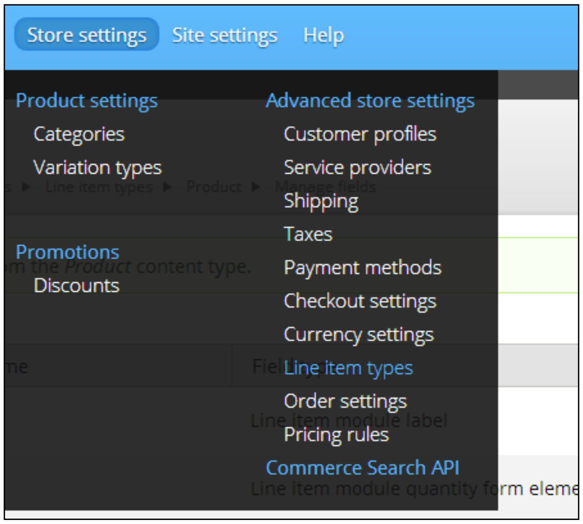
## Getting started

Log in as an administrator to your Drupal Commerce store's admin panel.

## How to do it...

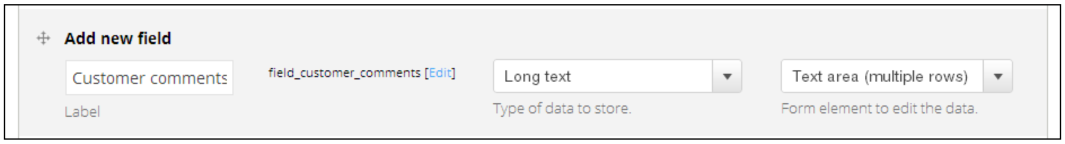
Follow these steps to add a new comments field to your product uploads:

1. Navigate to **Store settings** | **Line item types**:



2. Now select the product type to which you want to add the new text field. This example uses the **Product (Machine name: product)** type, which is created with the default installation of Drupal Commerce. You may want to create a new product type to accommodate this type of product in your store alongside more basic products.
3. Click on the **manage fields** option under the **Operations** column.
4. Now that you're in the **Manage fields** view, you will see an **Add new field** option towards the bottom of the screen. To create the new field, provide a sensible name for it in the **Label** field, and select **Long text** in the **Type of data to store** field.

> If you want to limit the customer comment to a single line of text, select **Text** as a value for the **Type of data to store** field.

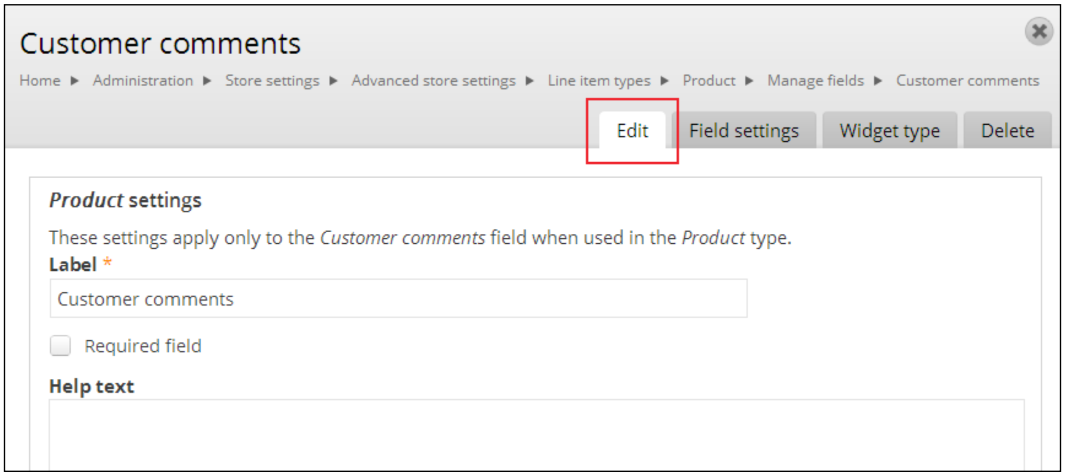


Add new field

Customer comments — field_customer_comments [Edit] — Long text — Text area (multiple rows)

Label — Type of data to store. — Form element to edit the data.

5. Click the **Save** button towards the bottom of the screen to save the new field to this product type. This will take you to the **Field settings** tab. You'll see a message saying **Customer Comments has no field settings**, so click on the **Edit** tab towards the top of your screen:



Customer comments

Home ▸ Administration ▸ Store settings ▸ Advanced store settings ▸ Line item types ▸ Product ▸ Manage fields ▸ Customer comments

*Product* settings

These settings apply only to the *Customer comments* field when used in the *Product* type.

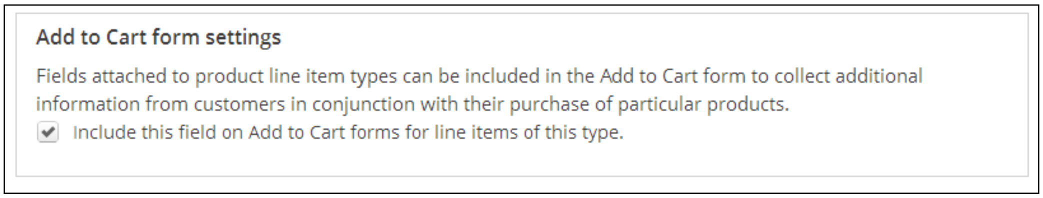
Label *

Customer comments

□ Required field

Help text

6. Scrolling down, you will see the **Add to Cart form settings** block, which includes a checkbox to **Include this field on Add to Cart forms for line items of this type**.



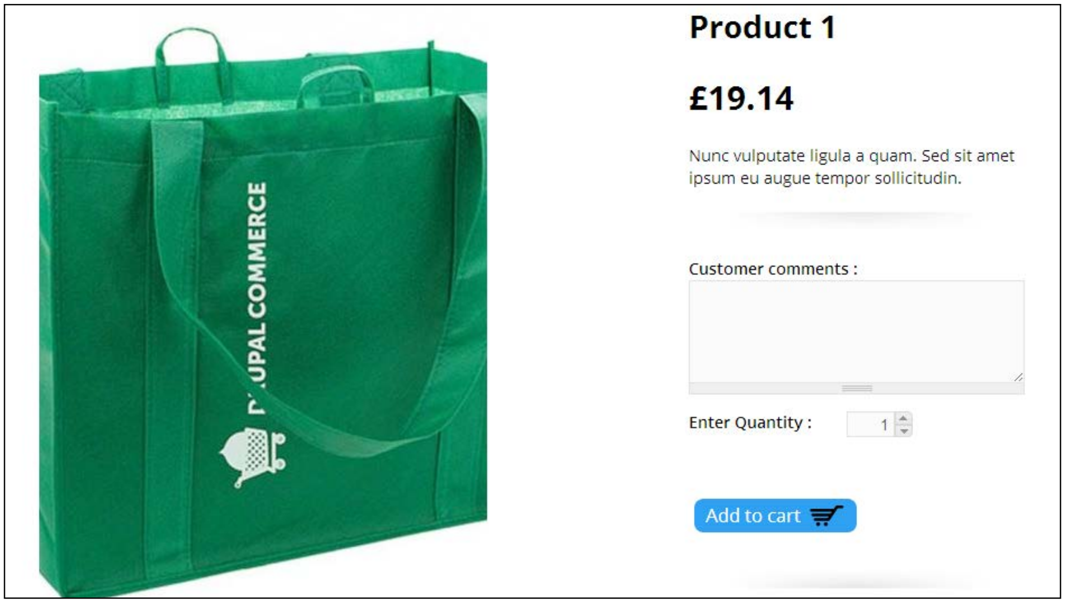
Add to Cart form settings

Fields attached to product line item types can be included in the Add to Cart form to collect additional information from customers in conjunction with their purchase of particular products.

☑ Include this field on Add to Cart forms for line items of this type.

7. Check this box, and click on the **Save settings** button.

8. If you now return to the frontend of your Drupal Commerce store, you will see that the new field appears on products of the type that you assigned to the preceding field (for example, the **Product (Machine name: product)** product types):



Product 1

£19.14

Nunc vulputate ligula a quam. Sed sit amet ipsum eu augue tempor sollicitudin.

Customer comments :

Enter Quantity : 1

9. You're done! You have now added a text field to allow customers to add custom comments to products that they're purchasing!

# Creating products with custom attributes

In e-commerce websites, it's common to want to add custom attributes to products. For example, if you're setting up Drupal Commerce to sell t-shirts, you may want to give customers a choice over the size of the t-shirts they buy.

This flexibility in Drupal Commerce allows you to add product groups with custom fields, so customers can select their colors, sizes, and other options before ordering their product from you.

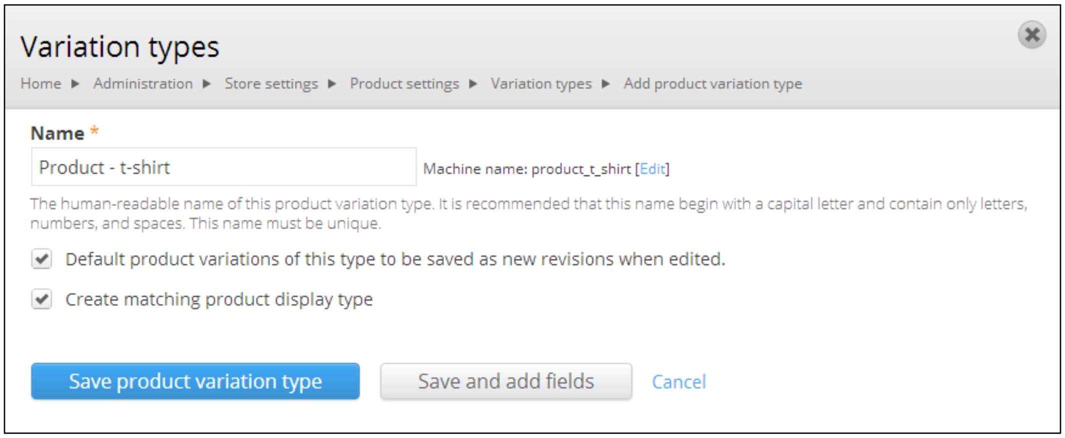
## Getting started

Log in to your Drupal Commerce website's administration panel, and navigate to **Products** | **Variation types**.

## How to do it...

Once you're logged in, follow these steps to create a Drupal Commerce product variation with the attributes that you need:

1. The neatest way to manage this new product type with size attributes is to create a new product variation in Drupal Commerce. Click on the **Add product variation type** button:
2. Enter a **Name** in the field (for consistency it's best to start the name with the word "Product" so you know what basic type of content you are working with in the future), and click on **Save and add fields**.



Variation types

Home ▶ Administration ▶ Store settings ▶ Product settings ▶ Variation types ▶ Add product variation type

Name *

Product - t-shirt

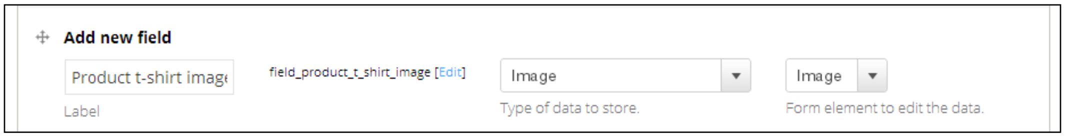
Machine name: product_t_shirt [Edit]

The human-readable name of this product variation type. It is recommended that this name begin with a capital letter and contain only letters, numbers, and spaces. This name must be unique.

☑ Default product variations of this type to be saved as new revisions when edited.

☑ Create matching product display type

3. You're now shown the **manage fields** tab. Using the **Add new field** form, add a new field with **Type of data to store** set to **Image** to allow you to add product images to this product variation type:



Add new field

Product t-shirt image

Label

field_product_t_shirt_image [Edit]

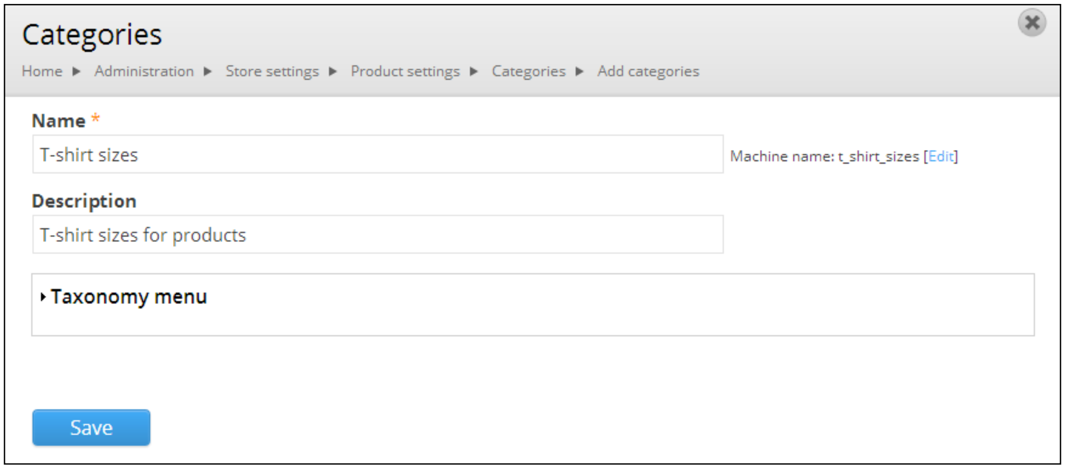
Image

Type of data to store.

Image

Form element to edit the data.

4. Click on the **Save** button now, and navigate to **Products** | **Categories**.
5. Click on the **Add categories** button, and enter a suitable name (and description, if you like):



Categories

Home ▶ Administration ▶ Store settings ▶ Product settings ▶ Categories ▶ Add categories

Name *

T-shirt sizes

Machine name: t_shirt_sizes [Edit]

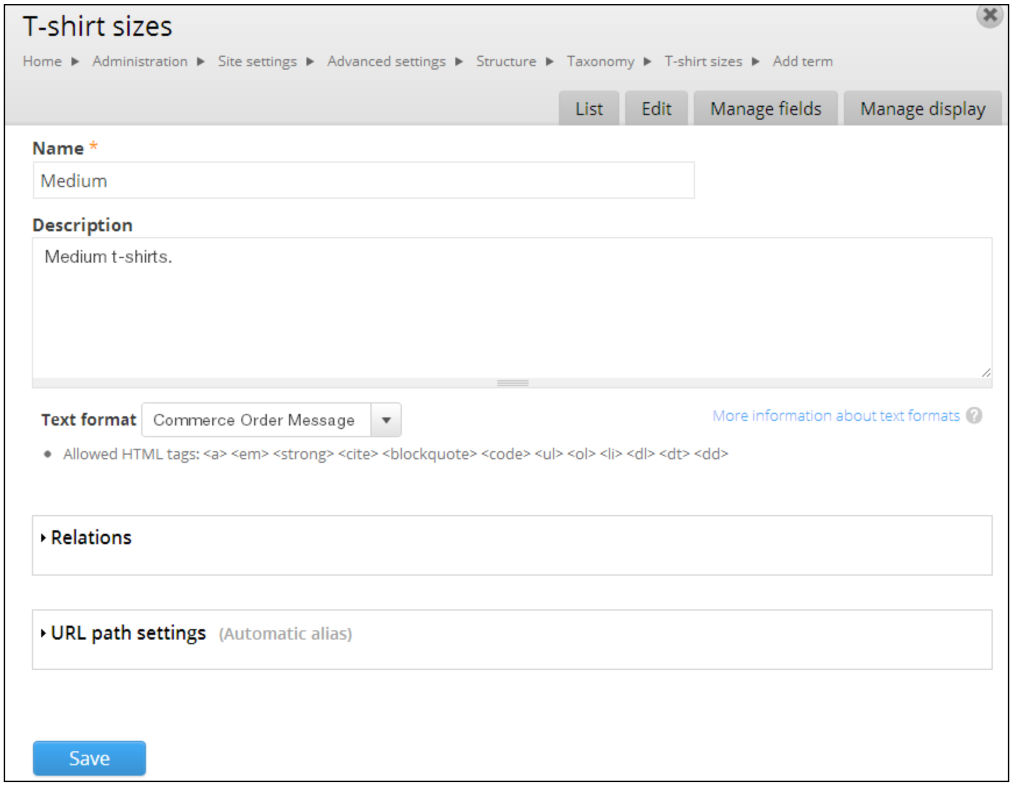
Description

T-shirt sizes for products

▸ Taxonomy menu

Save

6. Click on the **Save** button, and then select the **edit vocabulary** link alongside the new **T-shirt sizes** taxonomy that you have created. Navigate to the **List** tab at the top of the screen and click on the **Add term** button here to add as many attributes as you want for this product type:



T-shirt sizes

Home ▶ Administration ▶ Site settings ▶ Advanced settings ▶ Structure ▶ Taxonomy ▶ T-shirt sizes ▶ Add term

Name *

Medium

Description

Medium t-shirts.

Text format: Commerce Order Message

More information about text formats

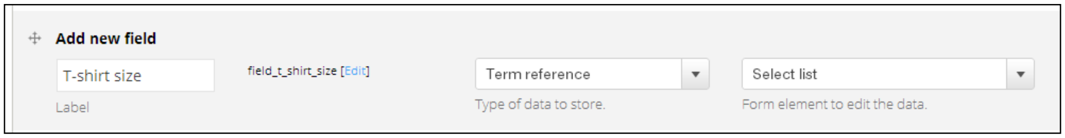
- Allowed HTML tags: <a> <em> <strong> <cite> <blockquote> <code> <ul> <ol> <li> <dl> <dt> <dd>

▸ Relations

▸ URL path settings (Automatic alias)

Save

7. Once you've done this, navigate back to **Store settings** | **Variation types** and click on the **manage fields** link that corresponds to the new product variation that you created in the previous steps.
8. Under the **Add new field** block, enter a suitable label (the example uses **T-shirt size** to differentiate it from other potential product size attributes in the future).
9. Select the **Term reference** value for the **Type of the data to store** field, and leave the value Drupal populates for the **Form element to edit the data** field as **Select list**:



Add new field

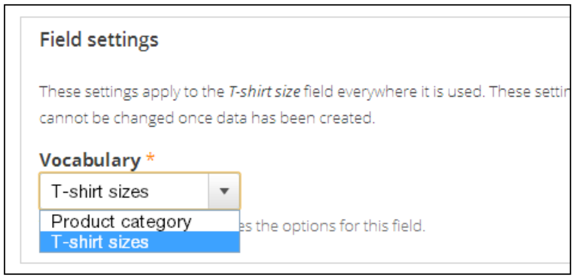
T-shirt size
Label

field_t_shirt_size [Edit]

Term reference
Type of data to store.

Select list
Form element to edit the data.

10. Click on the **Save** button once again to save this change, and you'll be shown the **Field settings** tab with the **Vocabulary** field. Select the **T-shirt sizes** taxonomy that you just created from this list, and click on the **Save field settings** button:

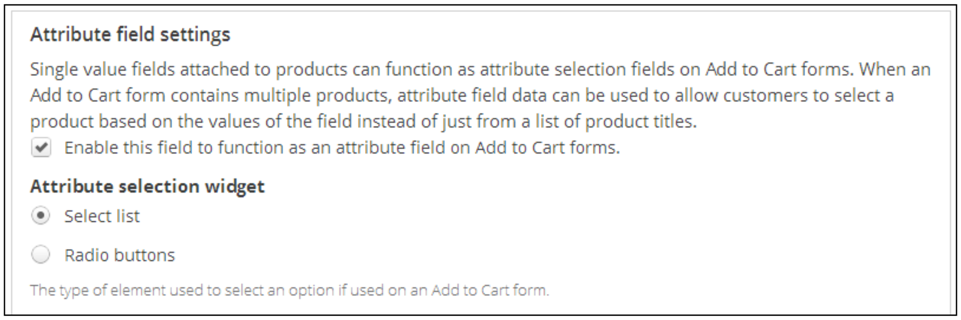


Field settings

These settings apply to the *T-shirt size* field everywhere it is used. These settir
cannot be changed once data has been created.

Vocabulary *
T-shirt sizes
Product category
T-shirt sizes
s the options for this field.

11. On the **Edit** tab, check the **Required field** checkbox to ensure a t-shirt has to be associated with a size.
12. In the **Attribute field settings** block, check the **Enable this field to function as an attribute field on Add to Cart forms** field to ensure it's available to customers to select on the frontend of your Drupal Commerce store, and set **Attribute selection widget** to **Select list**:



Attribute field settings

Single value fields attached to products can function as attribute selection fields on Add to Cart forms. When an Add to Cart form contains multiple products, attribute field data can be used to allow customers to select a product based on the values of the field instead of just from a list of product titles.

☑ Enable this field to function as an attribute field on Add to Cart forms.

Attribute selection widget

● Select list

○ Radio buttons

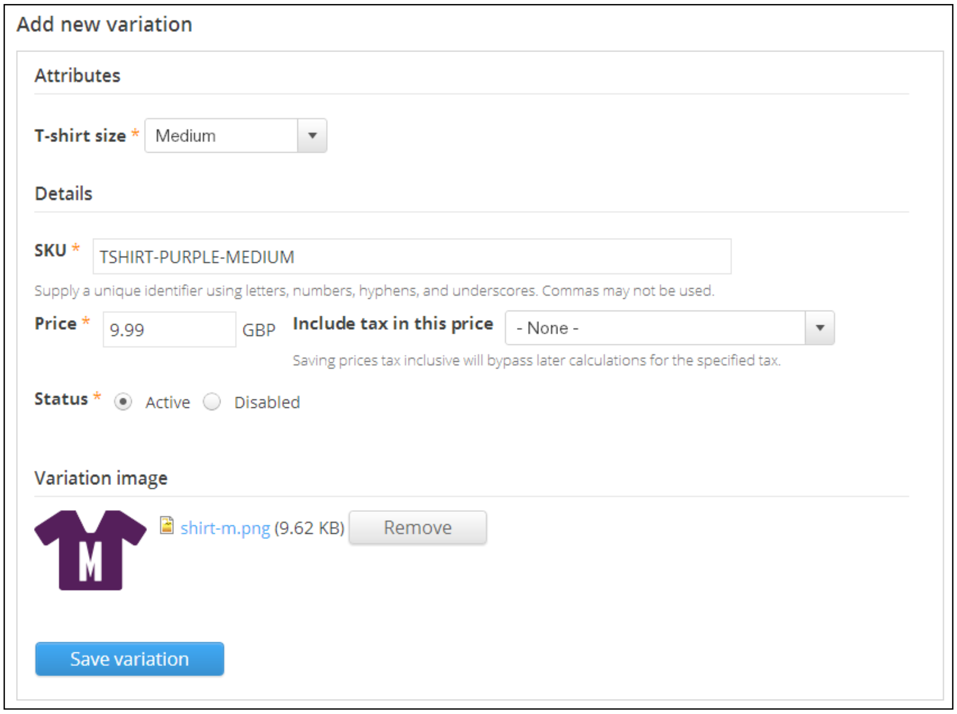
The type of element used to select an option if used on an Add to Cart form.

13. Click on the **Save settings** button.

14. Your next task is to create a product of this type to test whether your new attribute works, so navigate to **Products** | **Add a product**, and create a product of the variation that you created in earlier steps (**Product - t-shirt** was used in the example).

15. Add a title to your new product, and then complete the **Add new variation** block beneath this; select the value from the **T-shirt size** drop-down that you created earlier.

16. You can alter the price, image, and SKU depending on the attribute you have set in the product variation now, so a small t-shirt might be a different price to a medium t-shirt:



Add new variation

Attributes

T-shirt size * Medium

Details

SKU * TSHIRT-PURPLE-MEDIUM

Supply a unique identifier using letters, numbers, hyphens, and underscores. Commas may not be used.

Price * 9.99 GBP Include tax in this price - None -

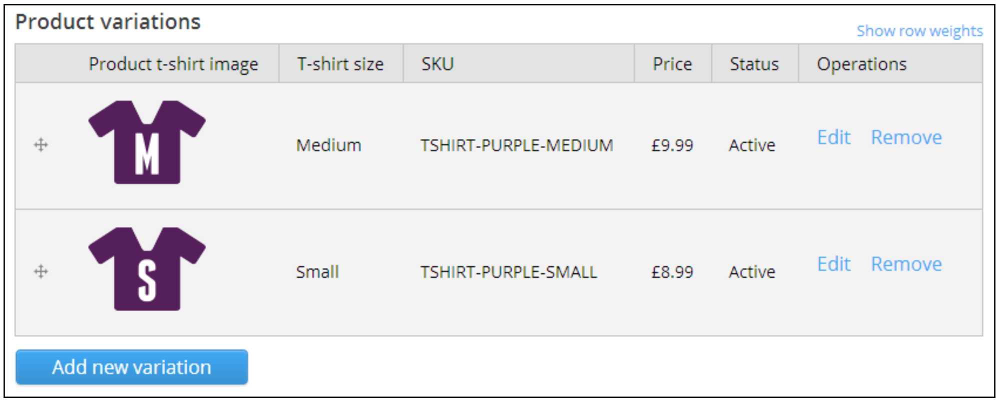
Saving prices tax inclusive will bypass later calculations for the specified tax.

Status * ● Active ○ Disabled

Variation image

shirt-m.png (9.62 KB)

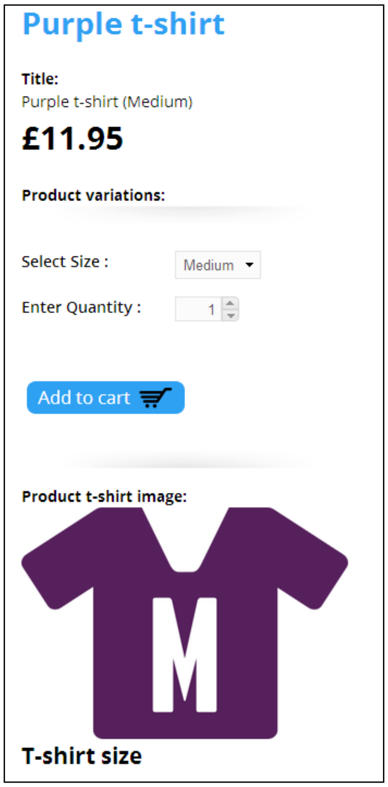
17. Click on the **Save variation** button to save this product, and add another product following the preceding steps:



Product variations

| | Product t-shirt image | T-shirt size | SKU | Price | Status | Operations |
|---|---|---|---|---|---|---|
| ✥ | M | Medium | TSHIRT-PURPLE-MEDIUM | £9.99 | Active | Edit Remove |
| ✥ | S | Small | TSHIRT-PURPLE-SMALL | £8.99 | Active | Edit Remove |

18. Click on the **Save** button at the bottom of the screen to save the products that you've just added, and you can then visit the frontend of your store to see the new product attribute in action.



Purple t-shirt

**Title:**
Purple t-shirt (Medium)

**£11.95**

**Product variations:**

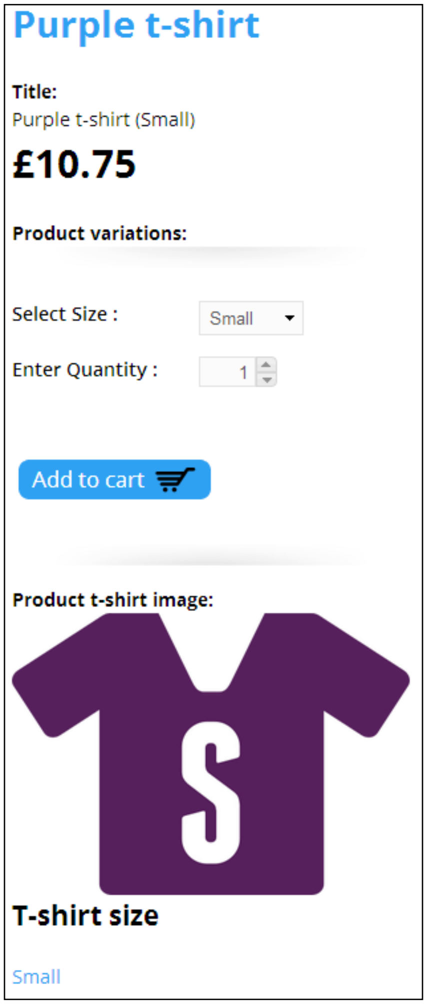
Select Size : Medium

Enter Quantity : 1

**Product t-shirt image:**

**T-shirt size**

19. If you change the **Select size** field to **Small** rather than **Medium**, you will see the image and price fields will change to reflect the differences you added between the products:



Purple t-shirt

Title:
Purple t-shirt (Small)

£10.75

Product variations:

Select Size : Small

Enter Quantity : 1

Product t-shirt image:

T-shirt size

Small

## How it works...

Adding a product with custom attributes in Drupal Commerce requires you to follow the following steps:

- Create a new product type to differentiate this type of product from others in your store, adding the field(s) that you need for product attributes (such as color or size)
- Create a taxonomy for each collection of attributes (for example, size, color) that you want to be able to assign to your products
- Create products for each combination of attributes and vary the size, SKU, image, and attributes' values as you need

As discussed in *Chapter 1*, *Installing and Configuring Drupal Commerce*, Drupal Commerce treats products as entities rather than nodes. The significance of this is that multiple products can be tied into a single product display, such as in the following example, where each shirt/size combination is its own product with a unique Stock Keeping Unit.

# Creating product categories in Drupal Commerce

A common feature of e-commerce websites is their ability to display products arranged by categories. This can be replicated in Drupal Commerce by making use of the inbuilt taxonomy system of Drupal, which can be used to categorize all content on your website (that is, not just your products).

With Drupal Commerce Kickstart 2, categories are built in as a feature, as you would expect in any e-commerce platform, and can allow customers to find the products they are looking for more quickly.

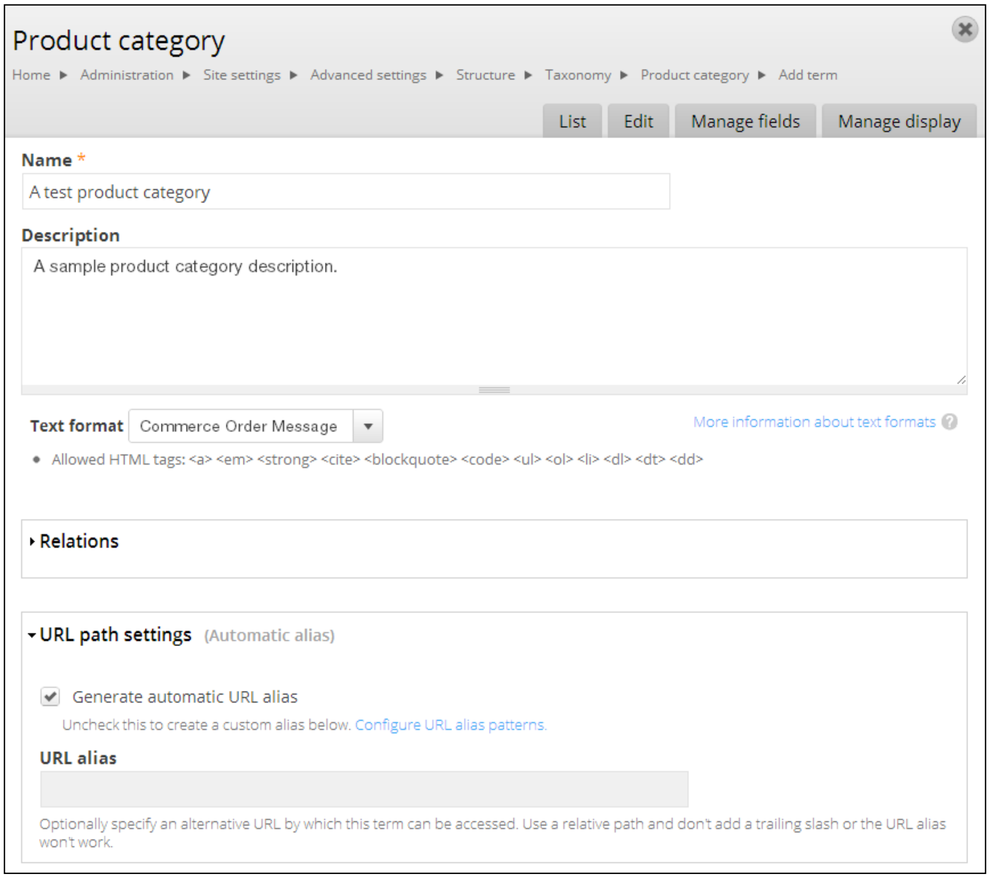
## Getting started

Log in to your Drupal Commerce administration panel. If you installed the Drupal Commerce Kickstart demo store, you'll already be able to see some sample categories in your store!

## How to do it...

To add a product category to your Drupal Commerce store, follow these steps:

1. Navigate to **Products** | **Categories** in your Drupal Commerce administration panel (enable the Commerce Kickstart Menus grouped under **Commerce Kickstart** in **Modules** if you can't see this), and click on the **Add terms** link towards the right of **Product category**. Give your new vocabulary a name (in the example, it is called Commerce categories):



Product category
Home ▸ Administration ▸ Site settings ▸ Advanced settings ▸ Structure ▸ Taxonomy ▸ Product category ▸ Add term
List
Edit
Manage fields
Manage display
More information about text formats
Relations
URL path settings (Automatic alias)

Name *

A test product category

Description

A sample product category description.

Text format: Commerce Order Message

- Allowed HTML tags: <a> <em> <strong> <cite> <blockquote> <code> <ul> <ol> <li> <dl> <dt> <dd>

☑ Generate automatic URL alias

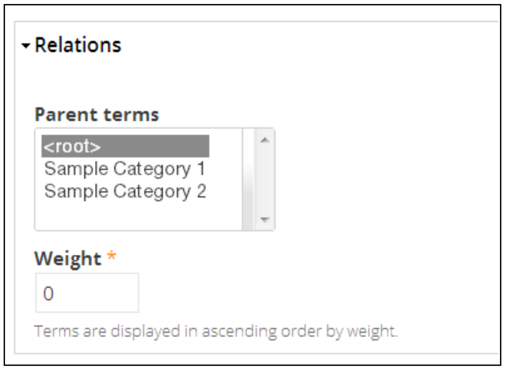
Uncheck this to create a custom alias below. Configure URL alias patterns.

URL alias

Optionally specify an alternative URL by which this term can be accessed. Use a relative path and don't add a trailing slash or the URL alias won't work.

2. If you can't see **Product category** under the **Vocabulary name** column, click on the **Add categories** button at the top of this screen and follow the steps on screen to create one.
3. You can also provide a description for your new category to help make managing your store easier in the future.

4. The **URL path settings** field allows you to define the path of your category (if you used Commerce Kickstart, you'll probably have allowed Drupal to automatically generate a URL alias anyway); enter a suitable value relevant to your category's name if you wish to override the default value. You may prefer to use a flat URL structure for your categories (for example, `example.com/test-category` or `example.com/test-category-2`), or to have them display under a less generic area of your website, such as `example.com/category/test-category`. Simply enter the value without trailing or prefixed slashes (for example, `category/test-category` or `just test-category`).
5. The **Relations** block allows you to select parent terms so you can assign a parent category and have sub categories in your store. For example, you may have a "t-shirts" category, and want "male t-shirts" and "female t-shirts" categories as sub categories of the "t-shirts" category.
6. You can also assign a **Weight** value here to change the ordering of the categories:



▾ Relations

Parent terms

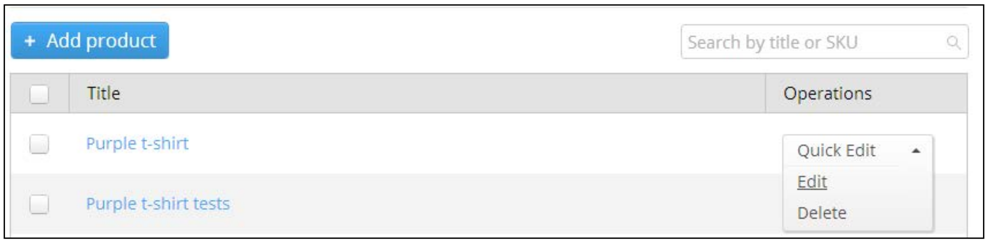
<root>
Sample Category 1
Sample Category 2

Weight *

0

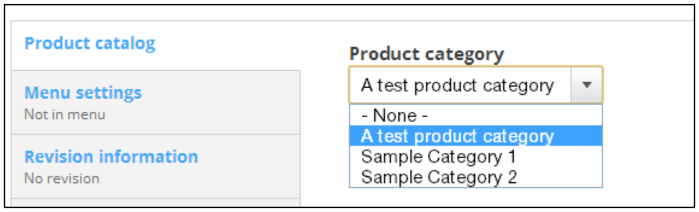
Terms are displayed in ascending order by weight.

7. Click on **Save**, and you're given the chance to add more categories; add as many as you need.
8. That's it! Your categories are ready to go. To assign a product to a category, navigate to **Products** | **Manage products**, and select **Edit** next to a product that you want to assign a category to:



| □ | Title | Operations |
|---|---|---|
| □ | Purple t-shirt | Quick Edit |
| □ | Purple t-shirt tests | |

9. Towards the bottom of the screen, you'll see the **Product catalog** tab, which allows you to select from the **Product category** drop-down:



Don't see this option when editing a product? You'll need to add a new field for the category of type "term reference" (taxonomy) to your product as you did in the preceding steps.

10. Click on **Save** and you're done! You can now browse your products by category in the frontend of your store.

## How it works...

Drupal uses taxonomies—collections of phrases—to help categorize all content types across websites it powers. By using taxonomies to help you create categories for your Drupal Commerce store, you're making the most of it, and can easily create categories listing your products. For more information on taxonomy in Drupal 7, see `http://drupal.org/documentation/modules/taxonomy`.

You can also use the Views module to create more highly customized categories on your Drupal Commerce stores; see *Chapter 7*, *Theming Drupal Commerce*.

# 3

# Managing Your Drupal Commerce Store Day-to-day

In this chapter we will cover the fundamentals of managing your Drupal Commerce store. We will be covering the following topics:

- Disabling products from your Drupal Commerce store
- Displaying and managing orders in your Drupal Commerce store
- Exporting products from Drupal Commerce
- Creating category menus in Drupal Commerce
- Extending customer profiles in Drupal Commerce

## Introduction

There's more to your Drupal Commerce store than adding your products and customizing its features. There are plenty of tasks you can do on a regular basis to keep your store up to date and manage it as part of your business.

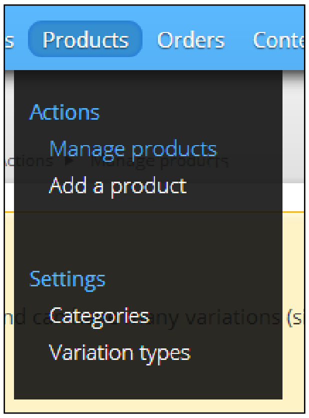
# Disabling products from your Drupal Commerce store

There may be times when you need to hide a product from view on your store. There are two ways to hide a product in your Drupal Commerce store:

1. Disabling the product display, which is useful if you want to disable an entire group of products (for example, purple t-shirts) either temporarily or permanently from your store.
2. Disabling specific products attached to a product display, which is useful if a particular product variation (for example, a large purple t-shirt) is either out of stock or discontinued.

## Getting started

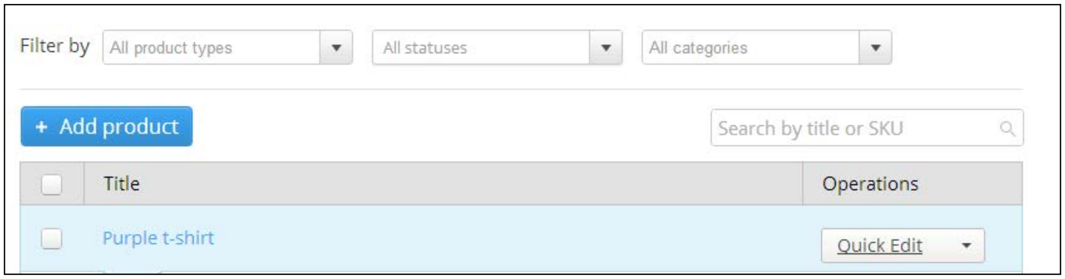
Log in to your Drupal Commerce store's administration panel and navigate to **Products** | **Manage Products**.



## How to do it...

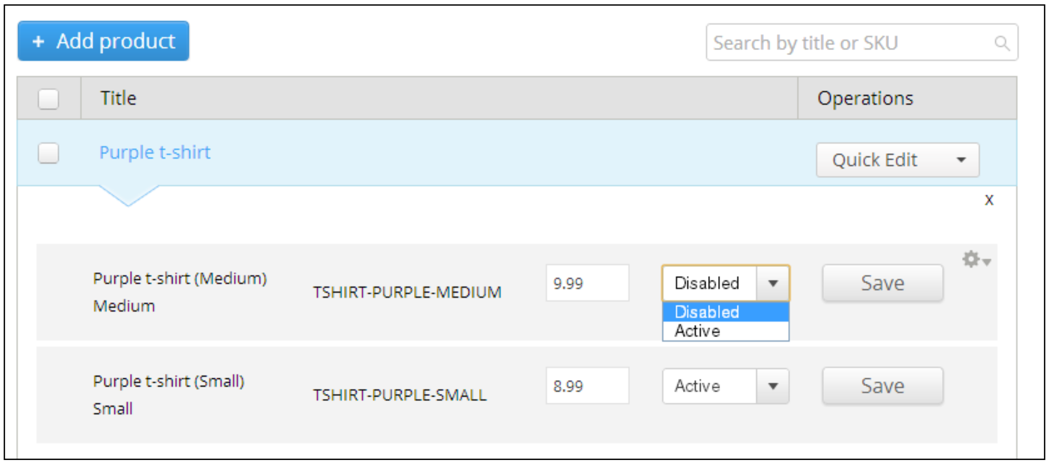
Now that you've found this, you'll see a list of the products currently in your Drupal Commerce store:

1. To disable a particular product (rather than an entire product variation), click on the **Quick Edit** option in the **Operations** column:



| □ | Title | Operations |
|---|---|---|
| □ | Purple t-shirt | Quick Edit |

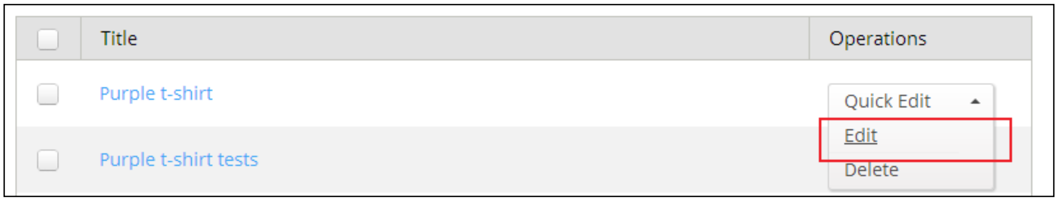
2. This displays the product variations as follows, and you can set specific products to **Disabled** in the drop-down menu:



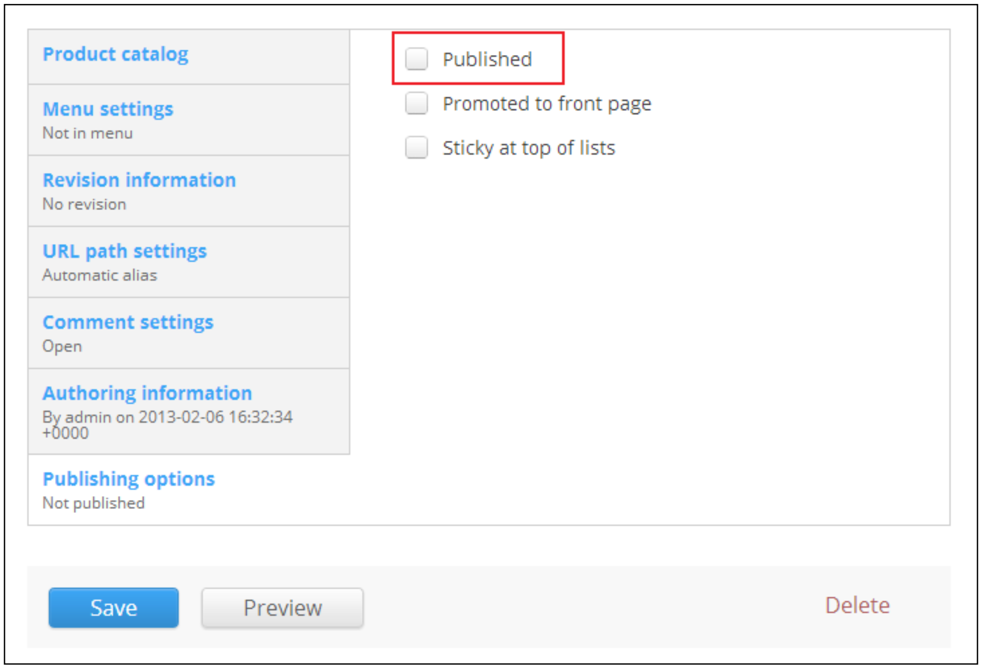
| | | | | |
|---|---|---|---|---|
| Purple t-shirt (Medium)<br>Medium | TSHIRT-PURPLE-MEDIUM | 9.99 | Disabled | Save |
| Purple t-shirt (Small)<br>Small | TSHIRT-PURPLE-SMALL | 8.99 | Active | Save |

3. Click on the **Save** button for each product you want to disable. That's it, the product or products you selected will not appear as an option to customers now.

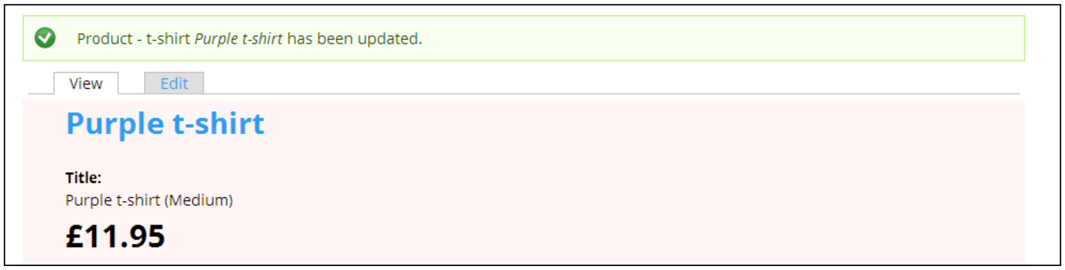
4. If you want to disable the whole product display (including all the variations assigned to it), select the **Edit** option from the **Operations** drop-down menu:



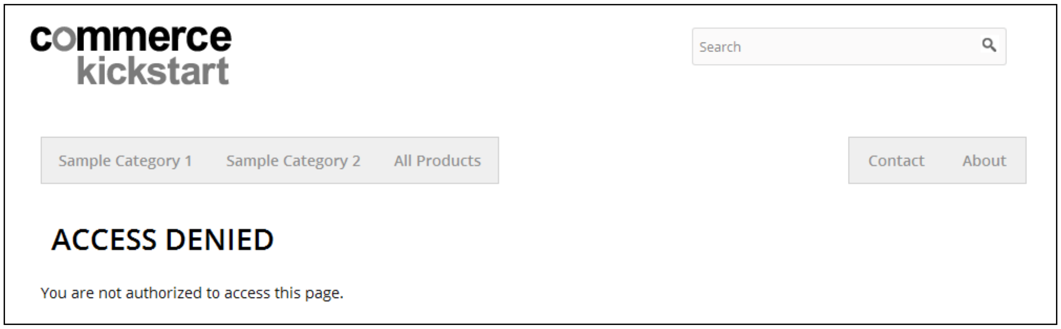
5. From here, scroll towards the bottom of the screen and find the **Publishing options** tab, just above the **Save** button:



6. Uncheck the **Published** checkbox to disable the entire product display from view on the frontend of your Drupal Commerce store. If you visit the frontend of your store when logged in as an administrator, you will be able to see the product display and the **View** and **Edit** tabs:


Product - t-shirt *Purple t-shirt* has been updated.

7. If you view the product display when not logged in as an administrator, you will see an "access denied" error:



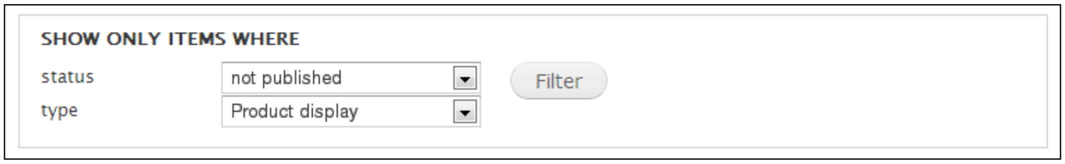
ACCESS DENIED

You are not authorized to access this page.

8. You can restrict the access of unpublished content via the **Permissions** tab under **People** in Drupal's administration panel.

## How it works...

Unpublishing content in Drupal removes it from public view, but allows you to add and edit content as an administrator. This can be particularly useful for preparing new product ranges on your Drupal Commerce store.



SHOW ONLY ITEMS WHERE

status: not published

type: Product display

# Displaying and managing orders in your Drupal Commerce store

Running an e-commerce website on a day-to-day basis can require a lot of administration work to track orders, but Drupal Commerce makes it easy to view all of your store's orders.

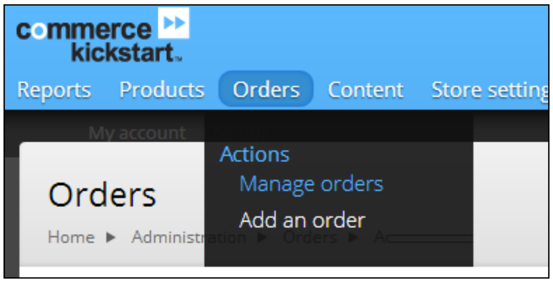
## Getting started

Log in to your Drupal Commerce website's administration panel.

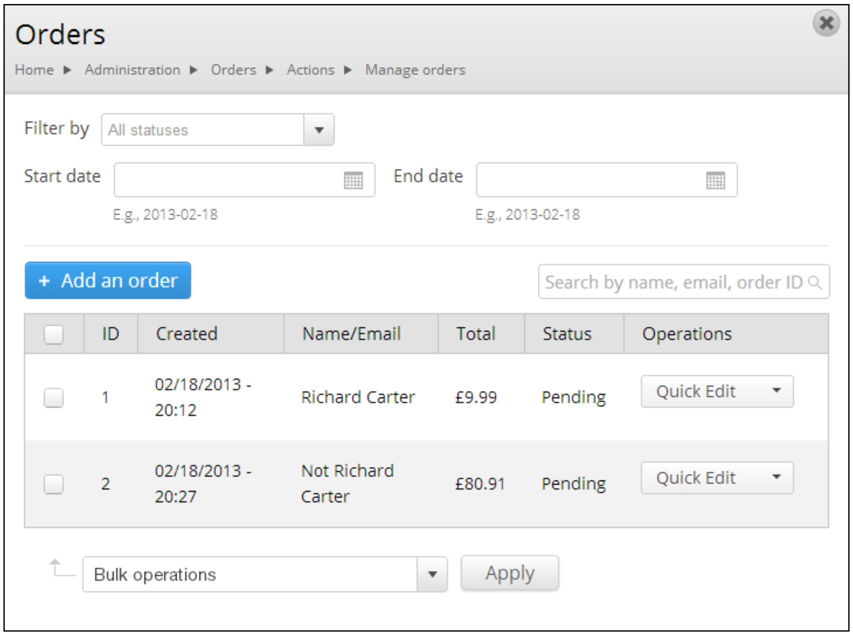
## How to do it...

Now that you're logged in to your Drupal Commerce store's administration panel, perform the following steps:

1. Navigate to **Orders** | **Manage orders**:



2. You'll now see all of the orders made in your Drupal Commerce store:


Orders
Home ▶ Administration ▶ Orders ▶ Actions ▶ Manage orders
+ Add an order
Search by name, email, order ID
Quick Edit
Bulk operations
Apply

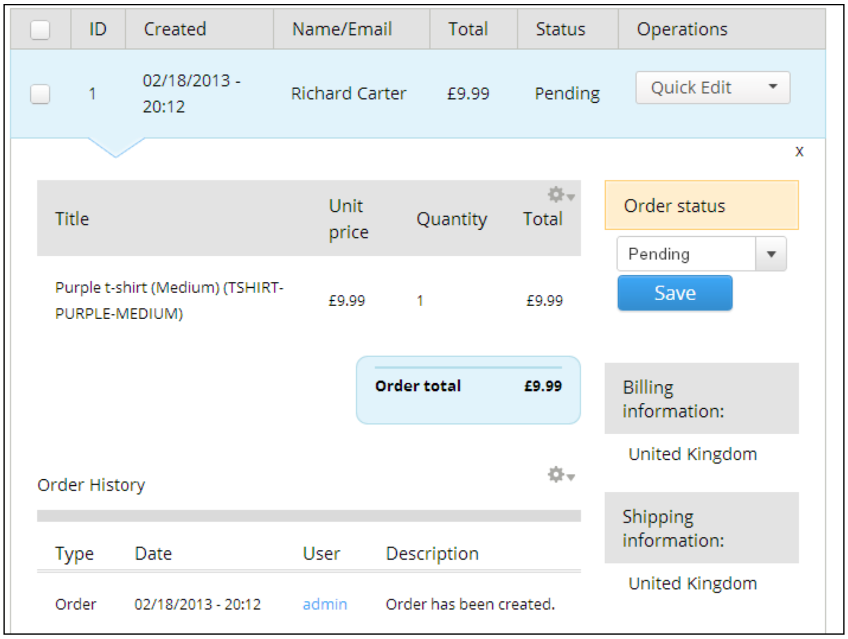
Filter by: All statuses

Start date: E.g., 2013-02-18

End date: E.g., 2013-02-18

| ☐ | ID | Created | Name/Email | Total | Status | Operations |
|---|---|---|---|---|---|---|
| ☐ | 1 | 02/18/2013 - 20:12 | Richard Carter | £9.99 | Pending | Quick Edit |
| ☐ | 2 | 02/18/2013 - 20:27 | Not Richard Carter | £80.91 | Pending | Quick Edit |

3. The **Operations** column provides you with a number of options to see more details. Click on the **Quick Edit** option to see an overview of the order and its status:



| □ | ID | Created | Name/Email | Total | Status | Operations |
|---|---|---|---|---|---|---|
| □ | 1 | 02/18/2013 - 20:12 | Richard Carter | £9.99 | Pending | Quick Edit |

| Title | Unit price | Quantity | Total |
|---|---|---|---|
| Purple t-shirt (Medium) (TSHIRT-PURPLE-MEDIUM) | £9.99 | 1 | £9.99 |

**Order total** **£9.99**

Order status: Pending

Billing information: United Kingdom

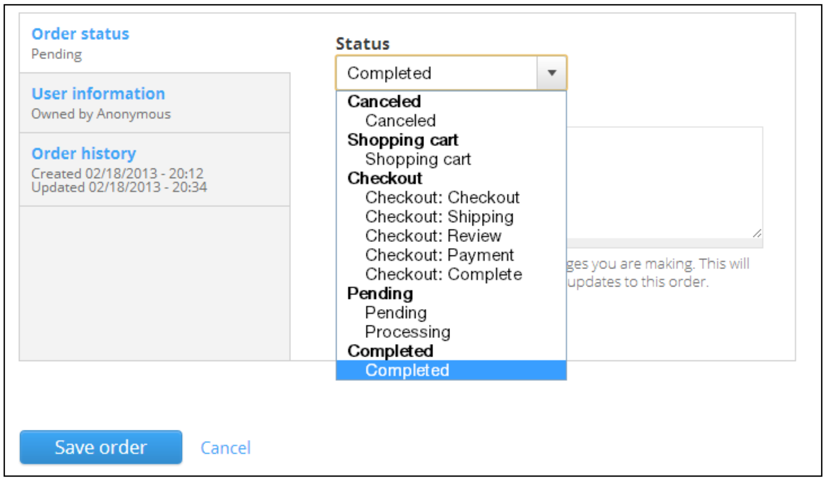
Shipping information: United Kingdom

Order History

| Type | Date | User | Description |
|---|---|---|---|
| Order | 02/18/2013 - 20:12 | admin | Order has been created. |

4. If you select **Edit** from the drop-down menu, you will see a more detailed overview of the customer's order, including the products they've ordered and more detailed shipping and billing addresses.

5. You're also able to change the **Order status** field, which keeps the customer up to date as to how their order is progressing:



Order status
Pending

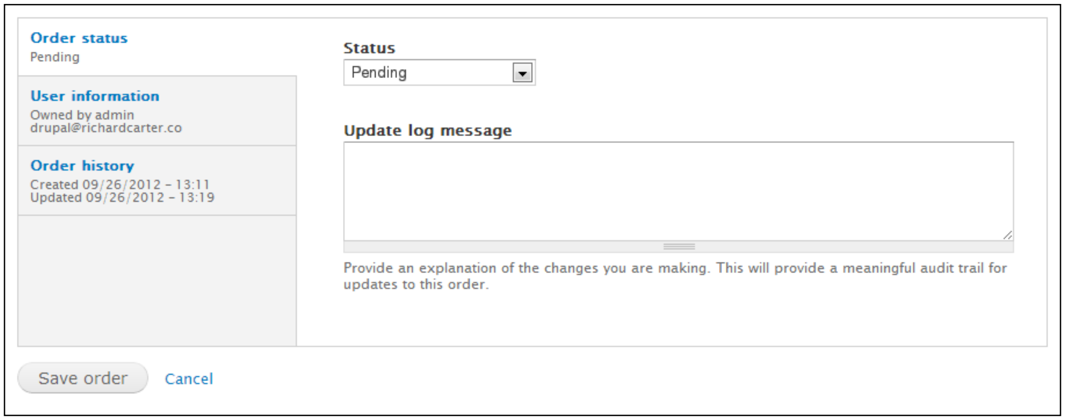
User information
Owned by Anonymous

Order history
Created 02/18/2013 - 20:12
Updated 02/18/2013 - 20:34

Status: Completed

ges you are making. This will
updates to this order.

6. Towards the bottom of this screen, the **Order status** tab allows you to leave a message for other administrators on your store using the **Update log message** field:



Order status
Pending

User information
Owned by admin
drupal@richardcarter.co

Order history
Created 09/26/2012 - 13:11
Updated 09/26/2012 - 13:19

Status: Pending

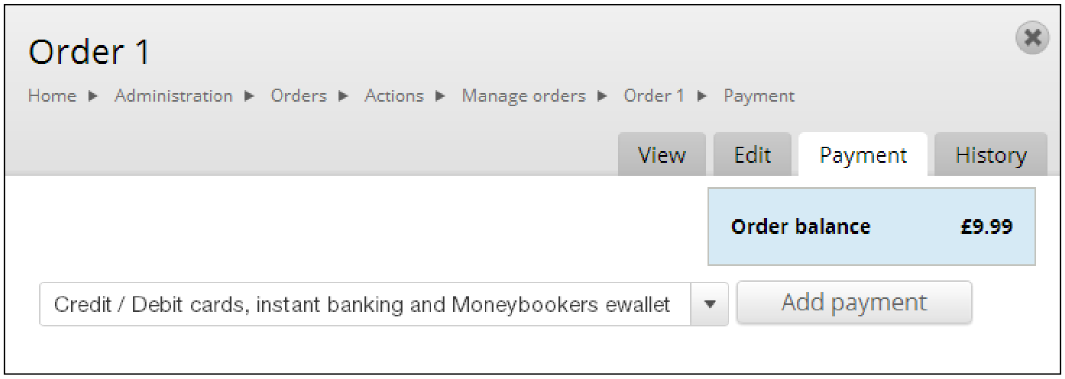
Update log message

Provide an explanation of the changes you are making. This will provide a meaningful audit trail for updates to this order.

At the time of writing, the Order Status feature doesn't work, but it's in progress!

7. The **Payment** option from the drop-down menu (also accessed at the top of the edit order screen from the **Payment** tab) allows you to view the payment status for the order.

   This is particularly useful if you want to process orders over the phone for customers, assuming you have the correct payment methods set up. Simply select the payment method from the drop-down menu and click on the **Add payment** button to progress it:



Order 1

Home ▶ Administration ▶ Orders ▶ Actions ▶ Manage orders ▶ Order 1 ▶ Payment

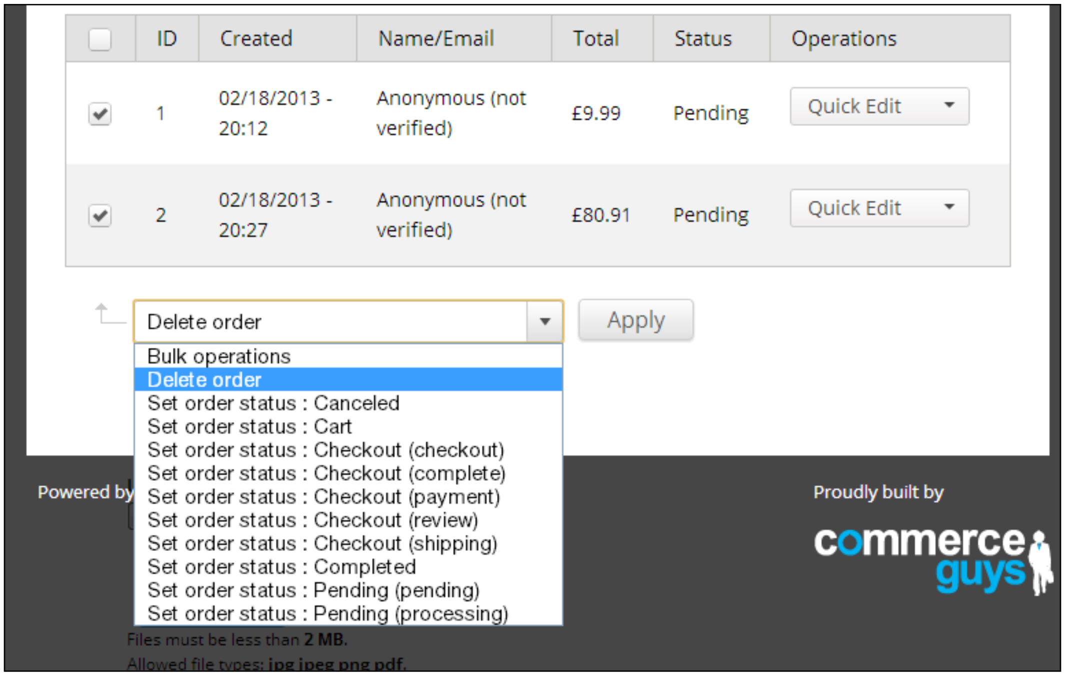
| | |
|---|---|
| Order balance | £9.99 |

Credit / Debit cards, instant banking and Moneybookers ewallet

8. And, of course, the **Delete** option will remove this order from your Drupal Commerce store entirely.

> Note that deleting transactions here cannot be undone, so be careful!

9. Finally, the **Bulk operations** drop-down menu beneath the list of orders provides a way to change the order status or delete multiple orders at the same time:



| □ | ID | Created | Name/Email | Total | Status | Operations |
|---|---|---|---|---|---|---|
| ☑ | 1 | 02/18/2013 - 20:12 | Anonymous (not verified) | £9.99 | Pending | Quick Edit ▾ |
| ☑ | 2 | 02/18/2013 - 20:27 | Anonymous (not verified) | £80.91 | Pending | Quick Edit ▾ |

Delete order ▾
Apply
Bulk operations
Delete order
Set order status : Canceled
Set order status : Cart
Set order status : Checkout (checkout)
Set order status : Checkout (complete)
Set order status : Checkout (payment)
Set order status : Checkout (review)
Set order status : Checkout (shipping)
Set order status : Completed
Set order status : Pending (pending)
Set order status : Pending (processing)
Powered by
Proudly built by
commerce guys
Files must be less than 2 MB.
Allowed file types: jpg jpeg png pdf.

# Exporting products from Drupal Commerce

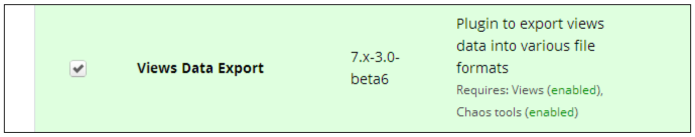
There may be times that you need to export your products from Drupal Commerce for use in another system that doesn't yet integrate with it, or simply to export products from your store to import and update them *en masse*.

In this recipe, you'll see how to use Drupal Views to create an export of products within your Drupal Commerce store, which you can then use to either review your product data or import into another system.

## Getting started

Exporting products from your store in bulk is not something that Drupal Commerce supports at the time of writing, so you will need to install the Views Data Export module (`http://drupal.org/project/views_data_export`).

Copy these files into the `/sites/all/modules` directory of your Drupal Commerce installation (or install through Drupal's control panel), and navigate to **Site Settings** | **Modules** in the administration panel to enable this module:



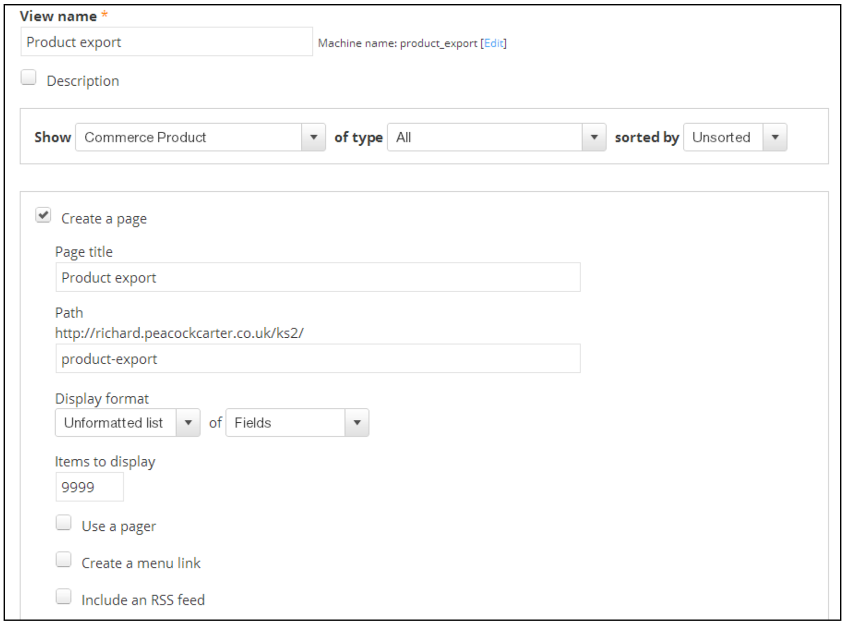
|  |  |  |  |
|---|---|---|---|
| ☑ | **Views Data Export** | 7.x-3.0-beta6 | Plugin to export views data into various file formats<br>Requires: Views (enabled), Chaos tools (enabled) |

Note that the requirements of **Views Data Export**, and the **Views** and **Chaos tools** modules, are already installed and enabled as part of your Drupal Commerce store.

## How to do it...

Once the module is installed, navigate to **Site Settings** | **Views**, and follow these steps:

1. Click on the **Add new view** button to create a view to display all of the products of the Drupal Commerce Product type:



**View name** *
Product export
Machine name: product_export [Edit]

□ Description

**Show** Commerce Product **of type** All **sorted by** Unsorted

☑ Create a page

Page title
Product export

Path
http://richard.peacockcarter.co.uk/ks2/
product-export

Display format
Unformatted list of Fields

Items to display
9999

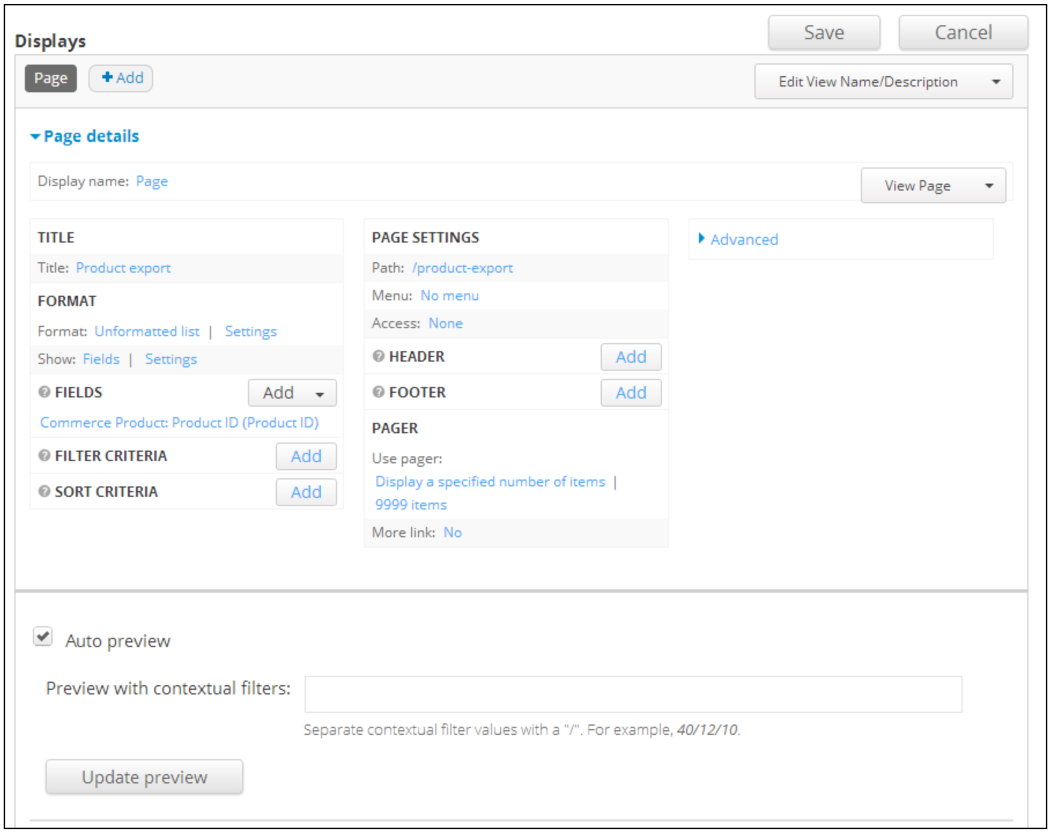
□ Use a pager

□ Create a menu link

□ Include an RSS feed

2. Set the **Items to display** field to a large number to ensure all of your products are displayed here, and uncheck the **Use a pager** box to turn pagination off. You can also set this value to `0` to show all the available results.

3. Once you are done, click on the **Continue & edit** button towards the bottom of the page, and Drupal will take you to the **Displays** screen:



Page details

Display name: Page

TITLE

Title: Product export

FORMAT

Format: Unformatted list | Settings

Show: Fields | Settings

FIELDS

Commerce Product: Product ID (Product ID)

FILTER CRITERIA

SORT CRITERIA

PAGE SETTINGS

Path: /product-export

Menu: No menu

Access: None

HEADER

FOOTER

PAGER

Use pager: Display a specified number of items | 9999 items

More link: No

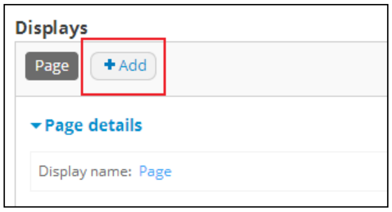
Advanced

☑ Auto preview

Preview with contextual filters:

Separate contextual filter values with a "/". For example, *40/12/10*.

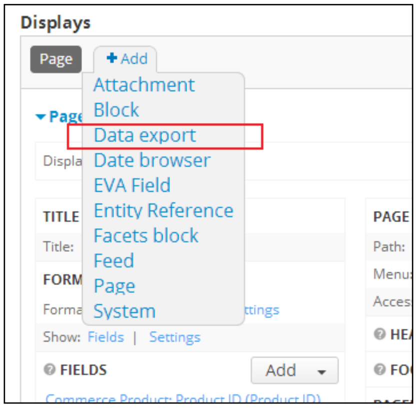
4. Select the **+Add** option towards the top-left corner of the screen to add a new display to the **View** panel:



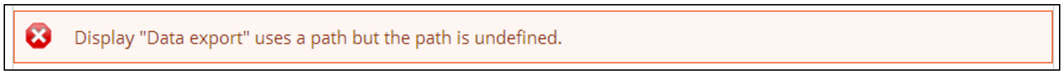
Page details

Display name: Page

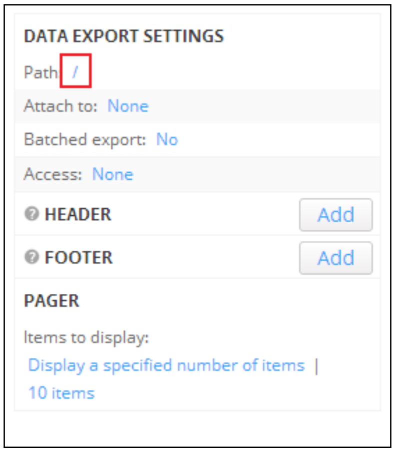
5. Click on this button, and you will see another drop-down menu. Select the **Data export** option:



6. You may now see an error message: **Display "Data export" uses a path but the path is undefined.** as shown in the following screenshot:


Display "Data export" uses a path but the path is undefined.

7. To define the **Path** value, click on the blue forward slash character under the **DATA EXPORT SETTINGS** block:


DATA EXPORT SETTINGS

Path: /

Attach to: None

Batched export: No

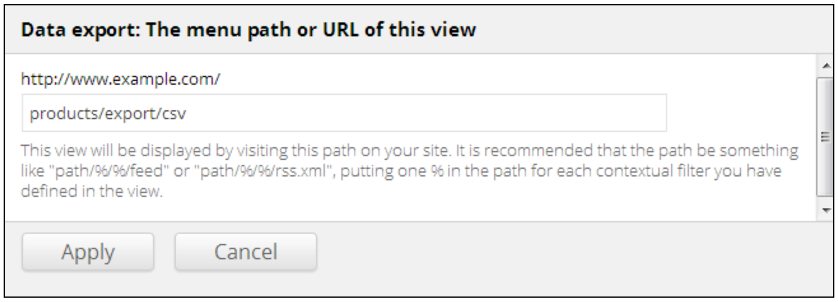
Access: None

PAGER

Items to display:

Display a specified number of items | 10 items

This will produce a pop-up window that allows you to fill in your desired path for the location of the data export:

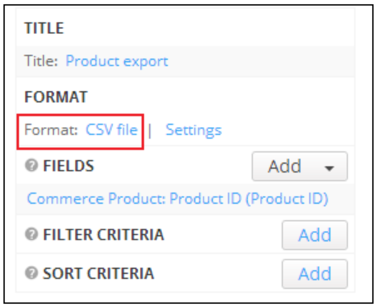


Data export: The menu path or URL of this view

http://www.example.com/ products/export/csv

This view will be displayed by visiting this path on your site. It is recommended that the path be something like "path/%/%/feed" or "path/%/%/rss.xml", putting one % in the path for each contextual filter you have defined in the view.

8. Enter a relevant path for your data export. The example uses `products/export/csv` as it will export the products in CSV format.
9. Click on the **Apply** button to save the new path.
10. You can now select the format you wish to export your products in. You will see the **Format** field in the left-hand column of the screen; the default setting is CSV, but you can change this by clicking the **CSV file** link:



TITLE

Title: Product export

FORMAT

Format: CSV file | Settings

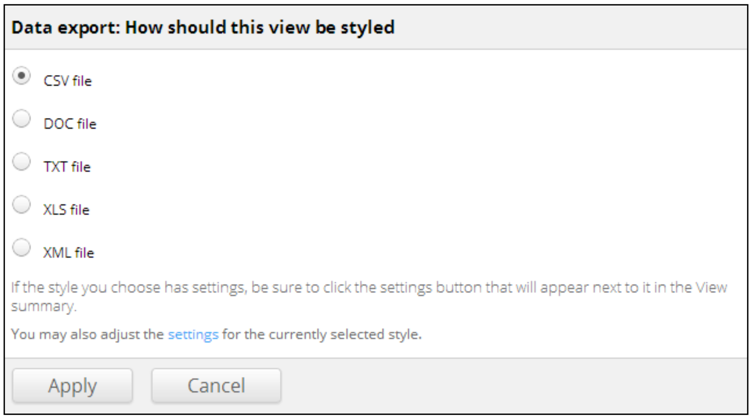
FIELDS

Commerce Product: Product ID (Product ID)

FILTER CRITERIA

SORT CRITERIA

11. You will now see a pop-up window that contains options for the exported data's format:



Data export: How should this view be styled

● CSV file

○ DOC file

○ TXT file

○ XLS file

○ XML file

If the style you choose has settings, be sure to click the settings button that will appear next to it in the View summary.

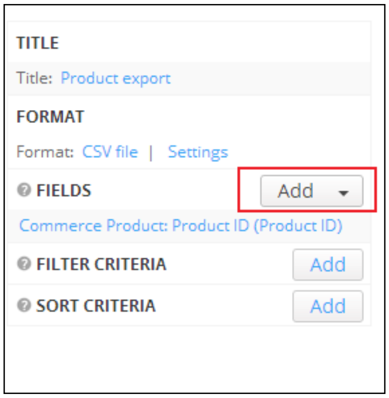
You may also adjust the settings for the currently selected style.

12. Select your preferred format here, and click on the **Apply** button again.

> You can further refine your output by clicking on the **Settings** pop-up.

13. Finally, click on **Save** to ensure your changes are remembered by Drupal.

14. You can now customize the fields that appear in the export; the product IDs will appear by default, which is not particularly useful to you! Click on the **Add** button to the right-hand side of the **Fields** block:



TITLE

Title: Product export

FORMAT

Format: CSV file | Settings

FIELDS

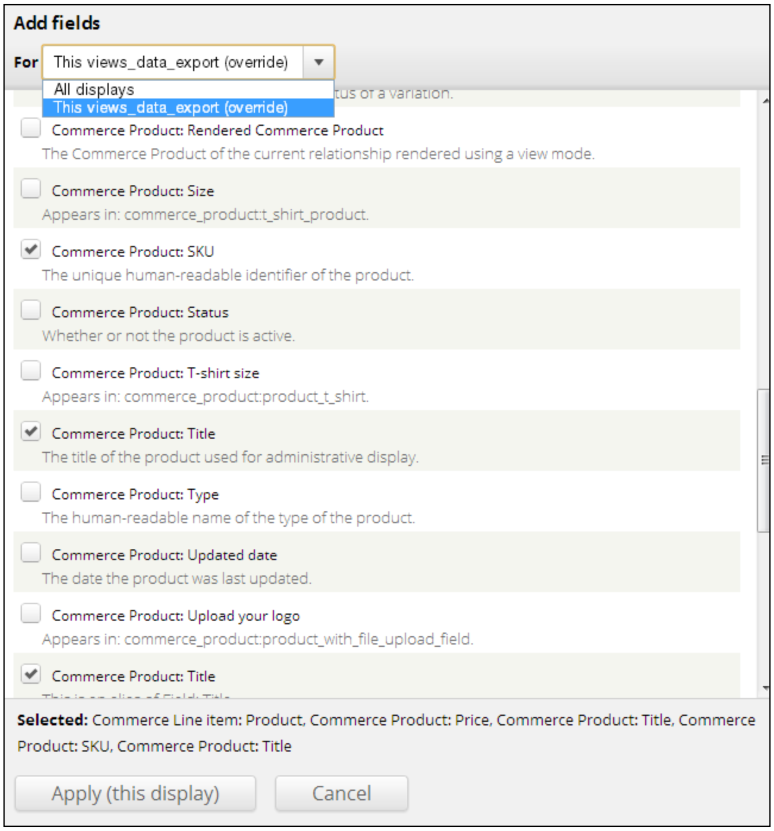
Commerce Product: Product ID (Product ID)

FILTER CRITERIA

SORT CRITERIA

15. This will display another pop-up window which gives you the opportunity to check the fields you want to appear in your product export.

16. Select your preferred fields (for the example export, the **Commerce Line item: Product**, **Commerce Product: Link**, **Commerce Product: Price**, **Commerce Product: Product ID**, **Commerce Product: Rendered Commerce Product**, **Content: Product**, and **Field: Image** fields were selected):



**Add fields**

For: This views_data_export (override)
- All displays
- This views_data_export (override)

...tus of a variation.

☐ Commerce Product: Rendered Commerce Product
The Commerce Product of the current relationship rendered using a view mode.

☐ Commerce Product: Size
Appears in: commerce_product:t_shirt_product.

☑ Commerce Product: SKU
The unique human-readable identifier of the product.

☐ Commerce Product: Status
Whether or not the product is active.

☐ Commerce Product: T-shirt size
Appears in: commerce_product:product_t_shirt.

☑ Commerce Product: Title
The title of the product used for administrative display.

☐ Commerce Product: Type
The human-readable name of the type of the product.

☐ Commerce Product: Updated date
The date the product was last updated.

☐ Commerce Product: Upload your logo
Appears in: commerce_product:product_with_file_upload_field.

☑ Commerce Product: Title

**Selected:** Commerce Line item: Product, Commerce Product: Price, Commerce Product: Title, Commerce Product: SKU, Commerce Product: Title

17. Now select the **Apply** button, and click on **Save** to save the changes to the **View** pane once the pop-up has closed.
18. If you visit the URL you set for the data export display of our view (that is, `http://www.example.com/products/export/csv` in the example), you will be presented with a CSV file (or whatever format you set in the view) to download that contains the products.
19. You may want to alter your Drupal website's roles and permissions settings to prevent anonymous users accessing this information. See `http://drupal.org/node/120614` for more information.
20. You can set the permissions and roles in Drupal that can access this view by clicking on the **Access** settings below the **Path** value.

### How it works...

The Views Data Export module allows you to export data from your Drupal Commerce store in a variety of formats, including the **Comma Separated Variable** (**CSV**) format, Microsoft Excel and Microsoft Word Document formats, **eXtensible Markup Language** (**XML**) format, and in a simple text (`.txt`) format, all of which may be useful to you if you wish to transfer your store's product inventory to another system beyond the scope of Drupal Commerce.

> The module also allows for usage in Drush (the Drupal Shell) once installed, see the project's `Drupal.org` page for more details: `http://drupal.org/project/views_data_export`.

## Creating category menus in Drupal Commerce

Having products on your store is all very well, but how do your customers find the products in your store? E-commerce websites typically make use of navigation menus to provide an easy way to access the primary categories that products fall in to.

While some customers may find your products easily enough on your Drupal Commerce store, adding navigation menus of your product categories provides another way for customers to find the type of products they are looking for.

### Getting ready

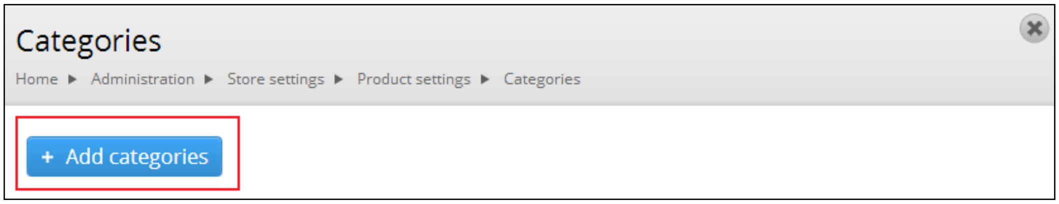
Log in to your Drupal Commerce store as an administrator, and navigate to **Products** | **Categories**.

> This recipe assumes you have the Commerce Kickstart Menus module enabled.

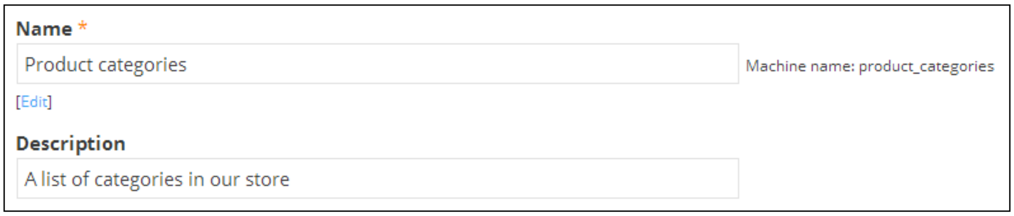
### How to do it...

Start adding your store's category navigation menu by following these steps:

1. Select the **Add categories** option at the top of the screen that appears:


Categories
Home ▶ Administration ▶ Store settings ▶ Product settings ▶ Categories
+ Add categories

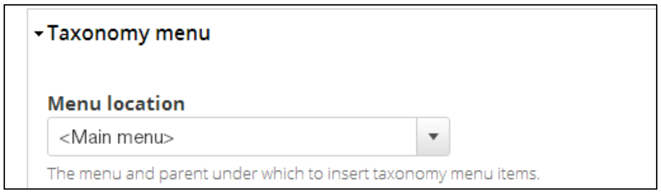
Give the new menu a name, and provide a description to remind you of this category's purpose:



Name *
Product categories
Machine name: product_categories
[Edit]

Description
A list of categories in our store

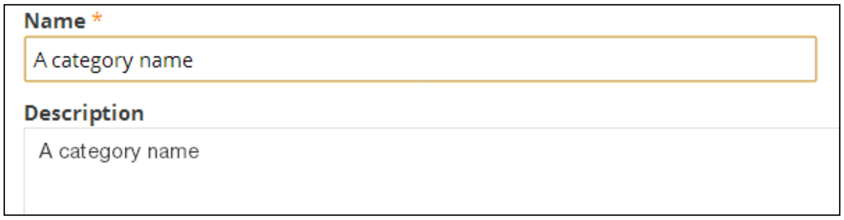
2. Expand the **Taxonomy menu** block below this, and select the relevant place in one of your website's existing menus from the **Menu location** block (the main menu is usually a good place to put these):



▾Taxonomy menu

Menu location
<Main menu> ▾
The menu and parent under which to insert taxonomy menu items.

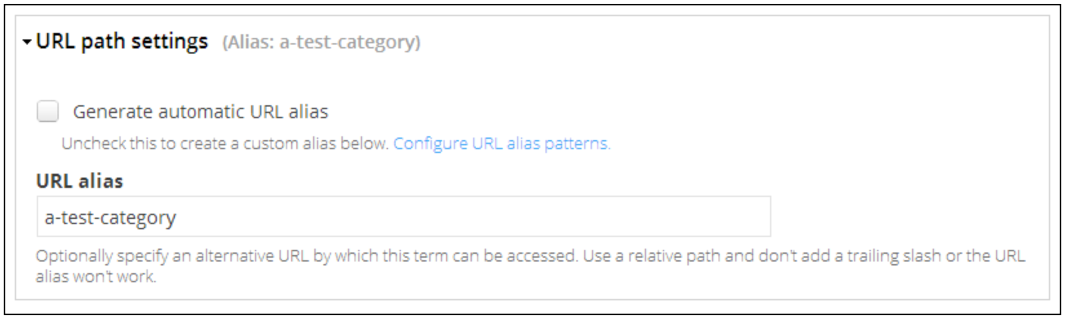
3. Click on the **Save** button beneath this block to continue.
4. Now navigate to **Store Settings** | **Categories** and select the **list terms** option next to your newly created category grouping (known as **vocabularies** in Drupal).
5. Click on the **Add term** button to get started, and add a name and description for the category:



Name *
A category name

Description
A category name

6. The **Relations** block below this allows you to set a category as a child of another category (that is, to make it a sub-category).
7. The **URL path settings** block is set to automatically generate a URL alias for the category, or you can manually overwrite this to generate your own URL for the category with the **URL alias** field:



▾URL path settings (Alias: a-test-category)

□ Generate automatic URL alias

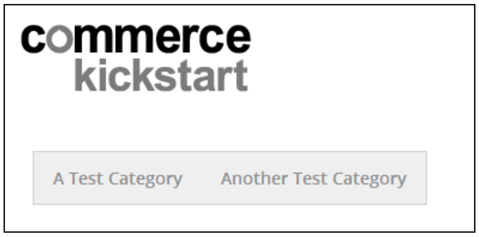
Uncheck this to create a custom alias below. Configure URL alias patterns.

URL alias

a-test-category

Optionally specify an alternative URL by which this term can be accessed. Use a relative path and don't add a trailing slash or the URL alias won't work.

8. Click on the **Save** button, and add as many categories as you need to by repeating steps 5 - 9 above.
9. You can assign products to a category by following the *Creating product categories in Drupal Commerce* recipe in *Chapter 2*, *Configuring Products with Drupal Commerce*.
10. If you now view the frontend of your Drupal Commerce store that customers are able to see, you will see the categories appear in the relevant menu you assigned them to earlier:



## How it works...

Categories in Drupal Commerce are created using the Drupal taxonomy feature, and you are simply creating a vocabulary (collection of terms) to use as product categories within your Drupal Commerce store.

# Extending customer profiles in Drupal Commerce

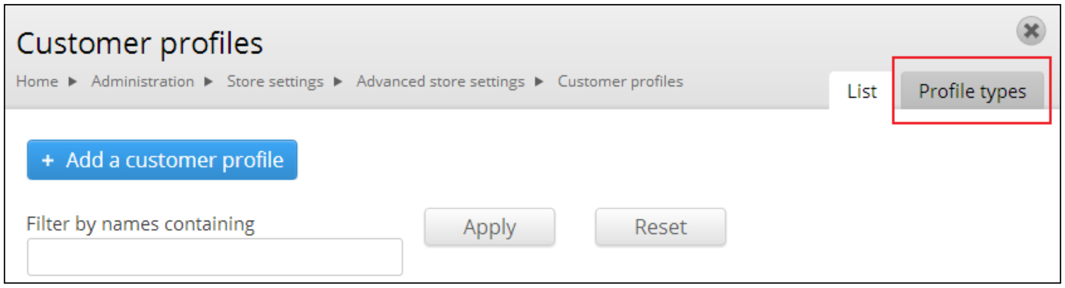
Customer profiles group information needed to complete orders in Drupal Commerce are billing information, shipping information, or delivery information. Customer profiles are created at a few points across your Drupal Commerce store as follows:

- When a customer reaches the checkout
- When a customer alters their profile (if the addressbook feature is enabled)
- When an administrator of the Drupal Commerce site creates a customer profile

The default customer profile contains very little in the way of information, and it may be beneficial to you as store owner to collect more information from your customers by extending your customer profile.

## Getting started

Log in to your Drupal store's administration panel, and navigate to **Store** | **Customer profiles**. Click on the **Profile types** tab at the top-right corner of your screen to see a list of existing customer profiles in your store:

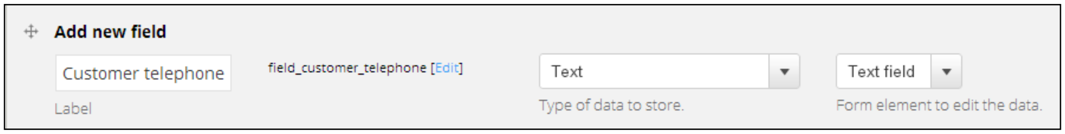

Customer profiles
Home ▸ Administration ▸ Store settings ▸ Advanced store settings ▸ Customer profiles
List
Profile types
+ Add a customer profile
Filter by names containing
Apply
Reset

## How to do it...

So, let's customize the customer profile:

1. By default, the only customer profile types are billing information and shipping information, and the latter is where you can add a new field to collect customers' telephone numbers.
2. Click on the **manage fields** option under the **Operations** column alongside the **Shipping information** profile type.

3. Complete the **Add new field** block, set the **Type of data to store** field to **Text**, and give the field a sensible name (the example uses **Customer telephone** as the value for the **Label** field):



Add new field

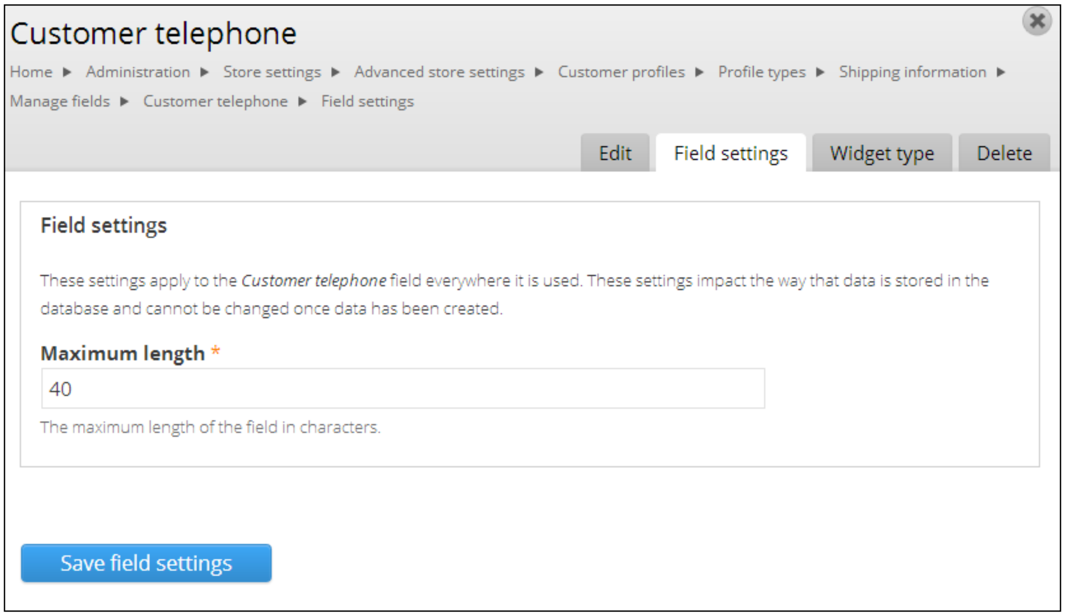
Customer telephone field_customer_telephone [Edit] Text Text field

Label

Type of data to store.

Form element to edit the data.

4. Click on the **Save** button to add this field to the customer profile.
5. The next screen you will see is the **Field settings** screen, which asks you to provide the maximum length of characters for your new field. Set this to around 40 characters to accommodate reasonable spaces, and country and area dialing codes:



Customer telephone

Home ▸ Administration ▸ Store settings ▸ Advanced store settings ▸ Customer profiles ▸ Profile types ▸ Shipping information ▸ Manage fields ▸ Customer telephone ▸ Field settings

Field settings

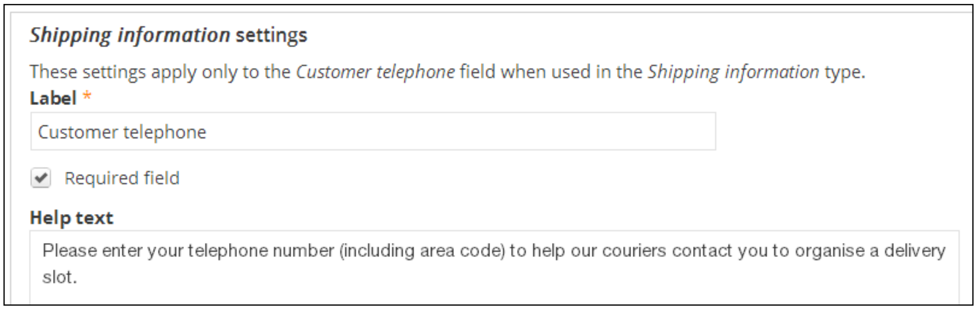
These settings apply to the *Customer telephone* field everywhere it is used. These settings impact the way that data is stored in the database and cannot be changed once data has been created.

Maximum length *

40

The maximum length of the field in characters.

6. Click on the **Save field settings** button to progress, and you are taken to the **Edit** screen for the field, where you can set whether **Customer telephone** is a **Required field**:



*Shipping information* settings

These settings apply only to the *Customer telephone* field when used in the *Shipping information* type.

**Label** *

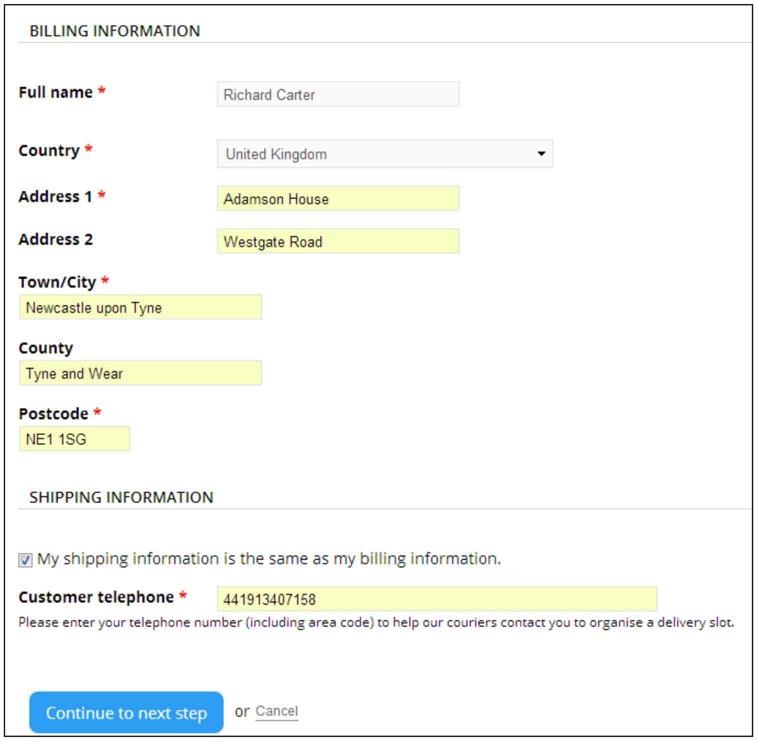
Customer telephone

☑ Required field

**Help text**

Please enter your telephone number (including area code) to help our couriers contact you to organise a delivery slot.

7. You're also able to provide a description to help the user in the **Help text** field.
8. Click on the **Save settings** button on this screen, and then click on the **Save** button at the bottom of the **Manage fields** screen you are returned to.
9. To test that your new field is appearing to customers who are checking out, add a product in your store to the shopping cart, and then checkout. You'll see your new field appear in the **Shipping information** section of the checkout:



BILLING INFORMATION

| | |
|---|---|
| **Full name** * | Richard Carter |
| **Country** * | United Kingdom |
| **Address 1** * | Adamson House |
| **Address 2** | Westgate Road |

**Town/City** *
Newcastle upon Tyne

**County**
Tyne and Wear

**Postcode** *
NE1 1SG

SHIPPING INFORMATION

☑ My shipping information is the same as my billing information.

**Customer telephone** * 441913407158

Please enter your telephone number (including area code) to help our couriers contact you to organise a delivery slot.

10. As you can see, this is a powerful way to extend your customer profiles to collect the information you need from them.

# 4

# Useful Modules for Drupal Commerce

This chapter covers a number of useful modules for Drupal Commerce to improve the utility of your e-commerce website, including:

- Allowing multiple shipments per delivery in Drupal Commerce
- Accepting payment by cheque in Drupal Commerce
- Setting a maximum and minimum order amount in Drupal Commerce
- Displaying progress steps for checkout in Drupal Commerce
- Automatically assigning an SKU to your products in Drupal Commerce

## Introduction

As you have seen, Drupal Commerce is a flexible system, and can be customized to behave as you want it to for a wide variety of e-commerce needs. By adding modules as and when you require them, the core of Drupal Commerce can remain lightweight and flexible for everyone's needs, and yet still retain the flexibility for more comprehensive e-commerce needs.

## Allowing multiple shipments per delivery in Drupal Commerce

Sometimes, it's useful to be able to customize your store to allow multiple deliveries per order. This is useful if you don't have all of the items your customer has ordered in stock, but they want the items that you do have in stock (similar to the way Amazon works with deliveries).

Installing the Commerce Delivery module allows your Drupal Commerce store to accept multiple shipments per order.

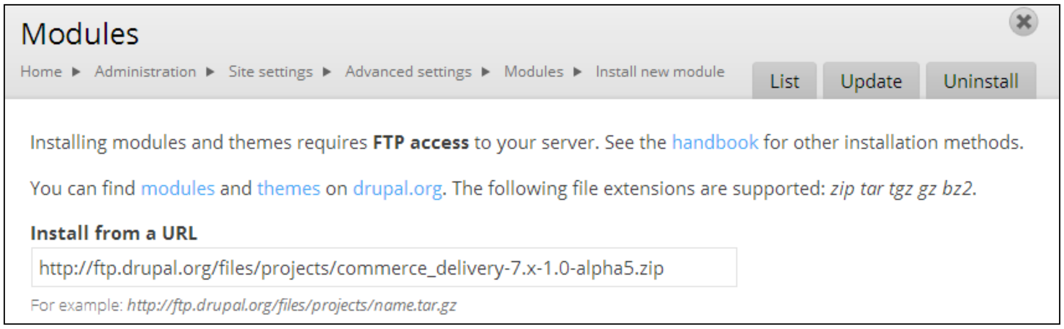
## Getting started

Download the Commerce Delivery module from Drupal.org at `http://drupal.org/project/commerce_delivery`, and copy the module file's URL to your clipboard (for example, something like `http://ftp.drupal.org/files/projects/commerce_delivery-7.x-1.0-alpha5.zip`).

## How to do it...

Now log in to your Drupal Commerce administration panel, and follow these steps:

1. Navigate to **Site Settings** | **Modules**.
2. Click on the **Install new module** button.
3. Paste the URL of the module files that you collected earlier into the **Install from a URL** field:



Modules
Home ▸ Administration ▸ Site settings ▸ Advanced settings ▸ Modules ▸ Install new module
List
Update
Uninstall

Installing modules and themes requires **FTP access** to your server. See the handbook for other installation methods.

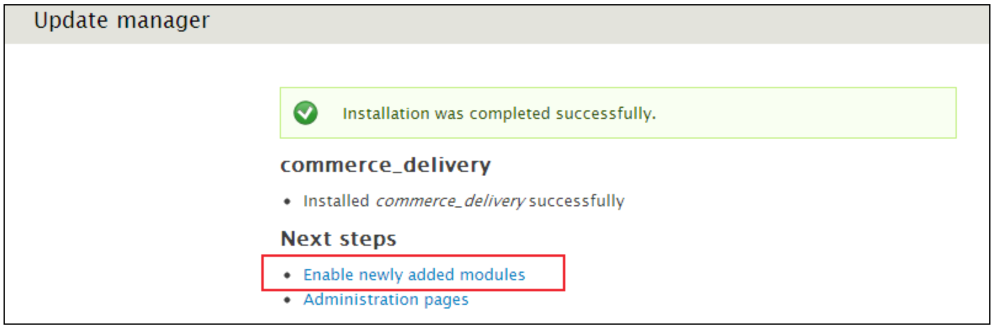
You can find modules and themes on drupal.org. The following file extensions are supported: *zip tar tgz gz bz2*.

**Install from a URL**

http://ftp.drupal.org/files/projects/commerce_delivery-7.x-1.0-alpha5.zip

For example: *http://ftp.drupal.org/files/projects/name.tar.gz*

4. Click on the **Install** button available at the bottom of the screen.
5. You'll now see a success screen, with an option to enable the newly added modules:



Installation was completed successfully.

commerce_delivery

- Installed *commerce_delivery* successfully

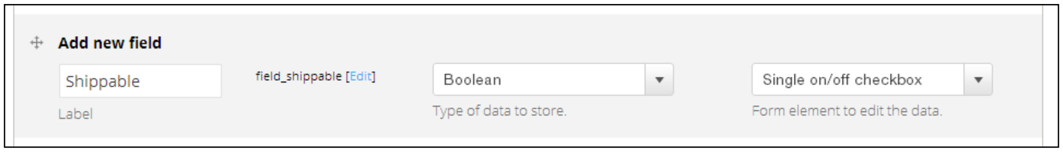
Next steps

- Enable newly added modules
- Administration pages

6. Click on this link, locate your newly installed module, and activate it (don't forget to click on **Save configuration**).

7. Now that the module is enabled, you can configure products to use it. Navigate to **Products** | **Variation types** and select the product variation type that you want to use by clicking on **manage fields**.

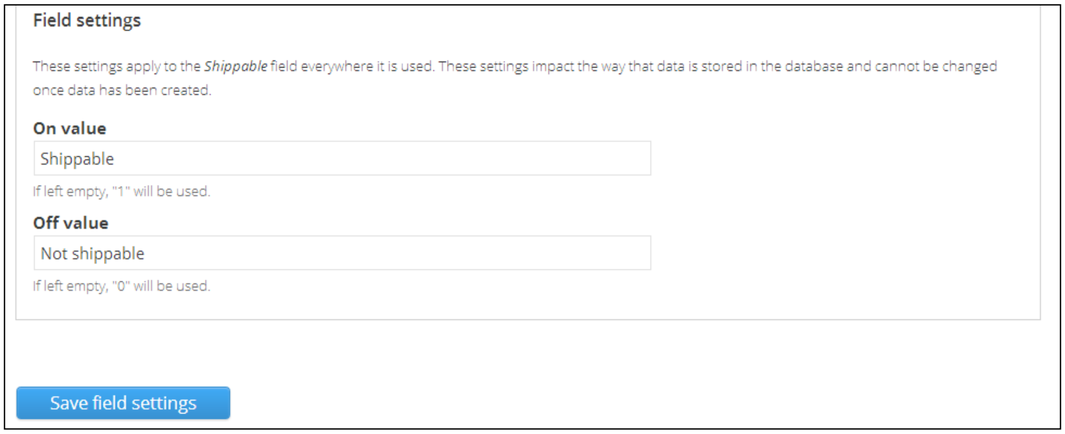
8. Using the **Add new field** block that is now shown, you can add a new field called `Shippable` to add a simple "Yes/No" field to your product administration to indicate whether or not the product requires shipping. Give the field a suitable name in the **Label** field, and then select **Boolean** for the **Type of data to store** field, and select **Single on/off checkbox** for the **Form** element to edit the data field:



Shippable field_shippable [Edit] Boolean Single on/off checkbox

Label Type of data to store. Form element to edit the data.

9. Click on the **Save** button to save the new **Shippable** field, and you are taken to the **Field settings** panel, where you can label the "on" and "off" values:



Field settings

These settings apply to the *Shippable* field everywhere it is used. These settings impact the way that data is stored in the database and cannot be changed once data has been created.

On value

Shippable

If left empty, "1" will be used.

Off value

Not shippable

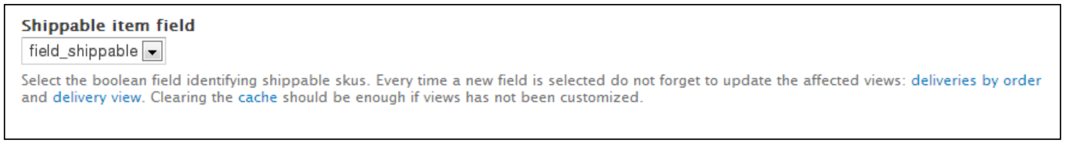
If left empty, "0" will be used.

10. In the preceding screenshot you can see that the **On value** field is set to **Shippable** and the **Off value** field is set to **Not shippable,** to help you identify what the field is for more readily than a "yes" or "no".

11. Click on the **Save field** settings button at the bottom of the screen to save these settings, where Drupal Commerce allows you to customize the field further. You can skip this section for the **Shippable** field as everything you will need is already set.

12. You can now customize your delivery options by navigating to **Store settings** | **Delivery settings** from the list. You may need to run `update.php` to see this appear in your menu options.

13. Here you will see the **Shippable item field** dropdown, where you can tell Drupal Commerce's Delivery module which field defines whether or not the product is shippable. Select the relevant field from the dropdown if the wrong one has been selected.



**Shippable item field**

field_shippable ▾

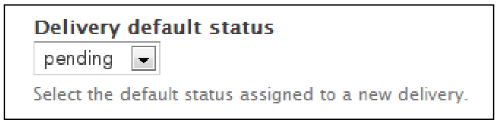
Select the boolean field identifying shippable skus. Every time a new field is selected do not forget to update the affected views: deliveries by order and delivery view. Clearing the cache should be enough if views has not been customized.

14. The module has a reasonable guess and selects it as default if you've named your field `Shipable` or `Shippable`. The **Delivery status options** field allows you to specify delivery statuses that you want to use in your store. By default, there are five statuses defined:
    - **new**: For newly created orders without a delivery status
    - **pending**: The delivery status is awaiting an update
    - **delivered**: This order has been delivered
    - **on hold**: The delivery has been put on hold
    - **cancelled**: This delivery has been canceled (perhaps because the customer canceled the order)

15. You can specify your own delivery statues by using the following format:

    ```
    ID|label
    ```

    In the preceding format, `ID` is the ID of the option that you want (a number unique to this list), and `label` is one or more words that describe your delivery status option. You can also set the **Delivery default status** to define which value is the usual value for the delivery status:

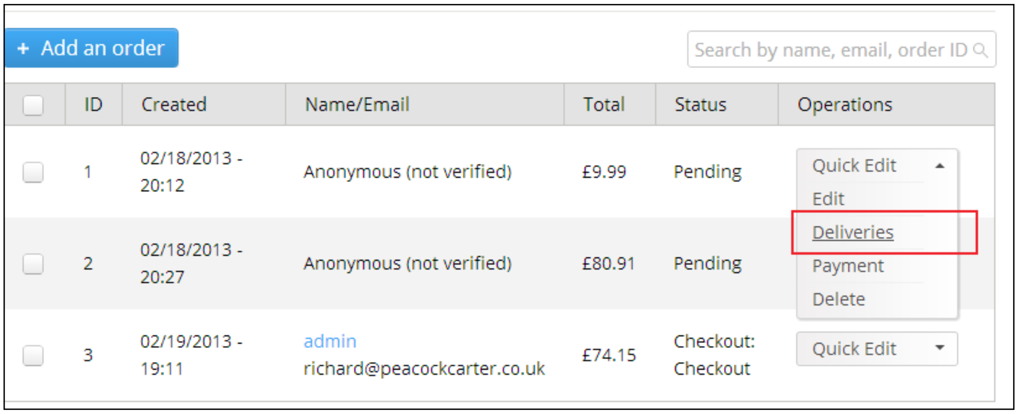


**Delivery default status**

pending ▾

Select the default status assigned to a new delivery.

16. Once you have finished configuring the module's options, you can click on the **Save configuration** button. If you now go and view orders (**Orders** | **Manage orders**), you will see that **Deliveries** now appears as an option in the **Operations** column for orders:



| □ | ID | Created | Name/Email | Total | Status | Operations |
|---|---|---|---|---|---|---|
| □ | 1 | 02/18/2013 - 20:12 | Anonymous (not verified) | £9.99 | Pending | Quick Edit ▴<br>Edit<br>Deliveries<br>Payment<br>Delete |
| □ | 2 | 02/18/2013 - 20:27 | Anonymous (not verified) | £80.91 | Pending | |
| □ | 3 | 02/19/2013 - 19:11 | admin<br>richard@peacockcarter.co.uk | £74.15 | Checkout: Checkout | Quick Edit ▾ |

## How it works...

The Commerce Delivery module extends Drupal Commerce to provide you with a more comprehensive delivery system by working on top of Drupal Commerce's framework and using rules to trigger its use.

# Accepting payment by cheque in Drupal Commerce

Even in this modern age with systems such as PayPal, Authorize.net, and other payment providers who make online payment simple, some online stores may want to accept cheques as payment for their orders. Typically, orders paid for by cheque are only dispatched once the cheque has cleared in the store's bank account, but some customers may find it useful that you accept cheques in addition to debit and credit cards despite the lengthened sales process.

This recipe guides you through the installation process of the Commerce Cheque module, which allows your Drupal Commerce store to accept cheques as a payment method.

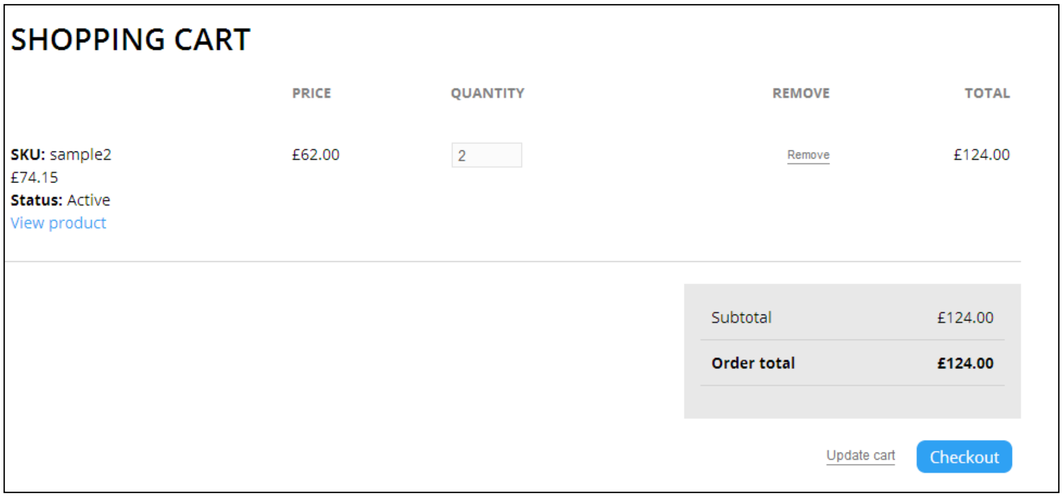
## Getting started

Download the Commerce Cheque module from Drupal.org at `http://drupal.org/project/commerce_cheque` and install the module. You can follow the instructions for installing a module in the previous chapter.

## How to do it...

Once the Commerce Cheque module is installed, follow these steps to configure payment by cheque as a payment method on your Drupal Commerce store:

1. Once you have installed the module, check whether your new payment method has been fully enabled. Go to the frontend of your Drupal Commerce store, add a product to your basket, and then click on the **Checkout** button.
2. At the **Shopping Cart** screen, proceed by using the **Checkout** button again:

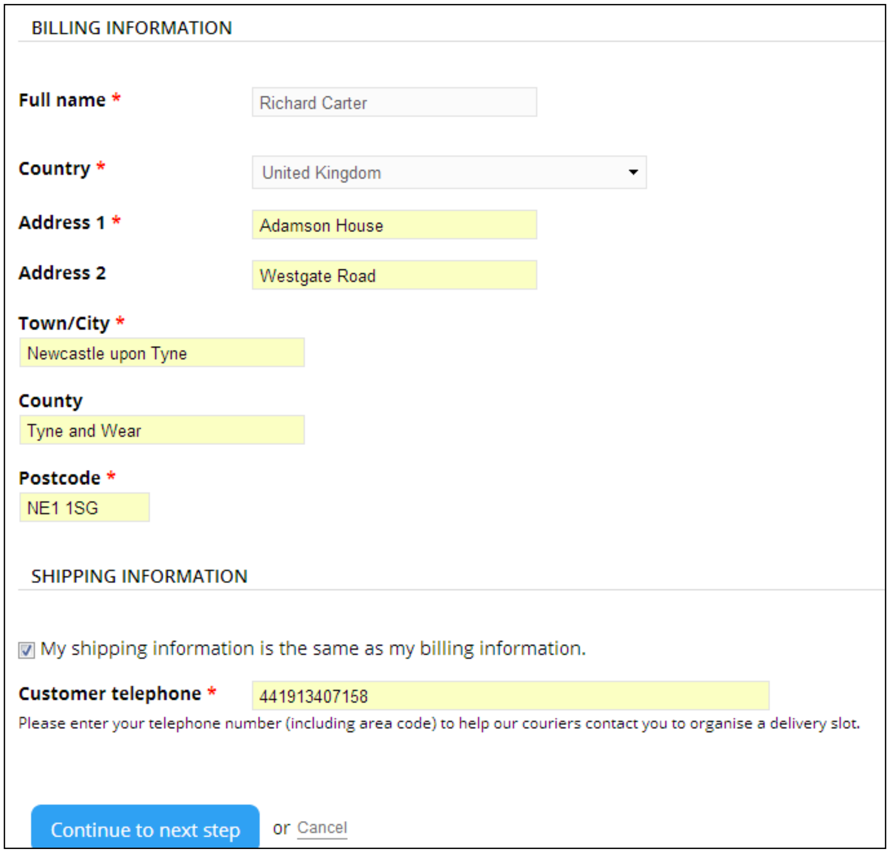


SHOPPING CART

| | PRICE | QUANTITY | REMOVE | TOTAL |
|---|---|---|---|---|
| **SKU:** sample2<br>£74.15<br>**Status:** Active<br>View product | £62.00 | 2 | Remove | £124.00 |

| | |
|---|---|
| Subtotal | £124.00 |
| **Order total** | **£124.00** |

3. You will now see your store's checkout view: fill in the **Billing information** panel, and then click on the **Continue to next step** button at the bottom of the screen to progress:



BILLING INFORMATION

| | |
|---|---|
| Full name * | Richard Carter |
| Country * | United Kingdom |
| Address 1 * | Adamson House |
| Address 2 | Westgate Road |

Town/City *
Newcastle upon Tyne

County
Tyne and Wear

Postcode *
NE1 1SG

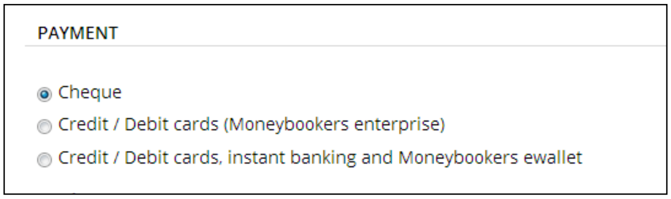
SHIPPING INFORMATION

☑ My shipping information is the same as my billing information.

Customer telephone * 441913407158

Please enter your telephone number (including area code) to help our couriers contact you to organise a delivery slot.

4. You'll now be presented with the **Review order** screen to confirm your order details, as well as payment options for your order, where you will see that the **Cheque** option is now enabled:



PAYMENT

● Cheque
○ Credit / Debit cards (Moneybookers enterprise)
○ Credit / Debit cards, instant banking and Moneybookers ewallet

5. That's it! You can now accept payment by cheque on your e-commerce store.
6. You can customize the label that appears for the customer in the preceding screen by going to **Store settings** | **Payment methods**.

# The Commerce Cart Expiration module

There may be needs in your store to have a shopping cart's contents expire after a given time to help you remove abandoned carts and luckily for you, there's a module for Drupal Commerce that can provide this functionality.

By installing the Commerce Cart Expiration module, you can expand your Drupal Commerce store to have your customers' shopping carts expire after a set period of inactivity.

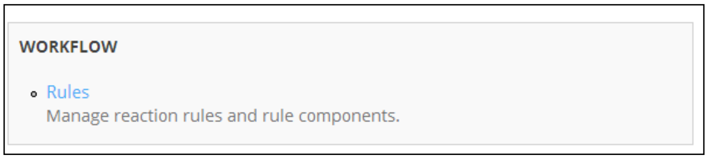
## Getting started

Download the Commerce Cart Expiration module from Drupal.org at `http://drupal.org/project/commerce_cart_expiration` and install it.

## How to do it...

Once the Commerce Cart Expiration module is installed, follow these steps:

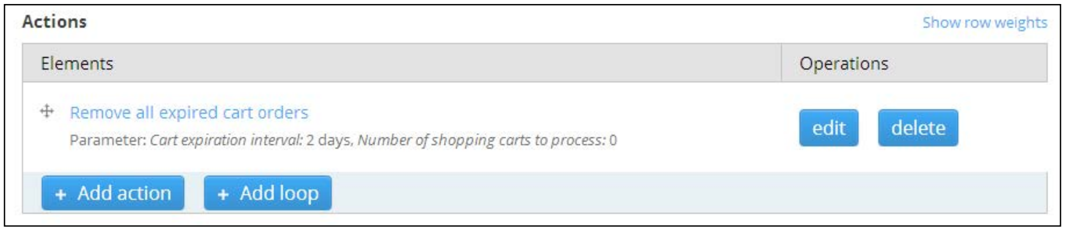
1. Next, you need to configure your store's rules to allow this module to function. Navigate to **Site settings** | **Configuration**, and under the **Workflow heading**, click on **Rules**:



WORKFLOW

- Rules
  Manage reaction rules and rule components.

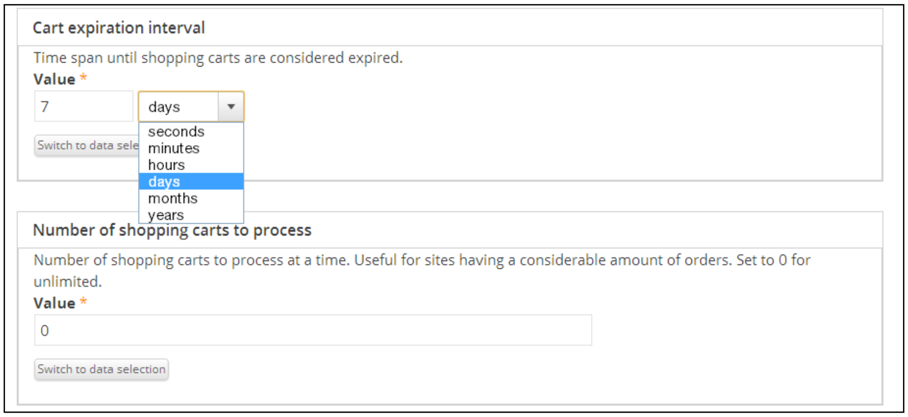
2. If you scroll to the bottom of the list of **Active** rules, you will now see a **Delete expired carts** option.
3. Click on the **edit** option in the **Operations** column to edit this rule, and you're presented with a screen to edit the **Delete expired carts** reaction rule.
4. Edit the **Remove all expired cart orders** element under the **Actions** heading by clicking on the **edit** link in the **Operations** column:



Actions

| Elements | Operations |
|---|---|
| Remove all expired cart orders<br>Parameter: *Cart expiration interval:* 2 days, *Number of shopping carts to process:* 0 | edit delete |

5. Here, you can change the period of time after which the carts' contents will expire in the **Cart expiration interval** field.
6. Set the carts to expire after they haven't been checked out after 7 days:



Cart expiration interval

Time span until shopping carts are considered expired.

Value *

7 days

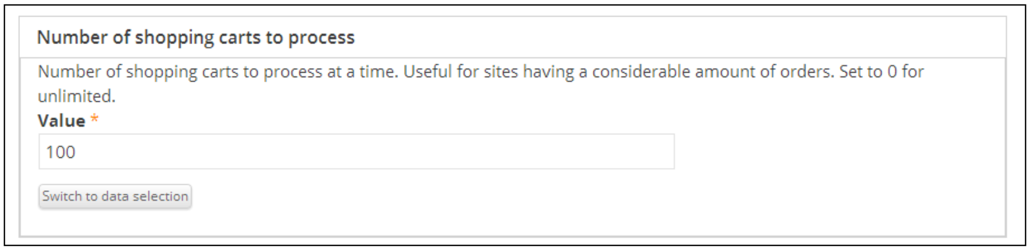
Number of shopping carts to process

Number of shopping carts to process at a time. Useful for sites having a considerable amount of orders. Set to 0 for unlimited.

Value *

0

7. The next field is **Number of shopping carts to process**, which controls how many abandoned carts' are removed:



Number of shopping carts to process

Number of shopping carts to process at a time. Useful for sites having a considerable amount of orders. Set to 0 for unlimited.

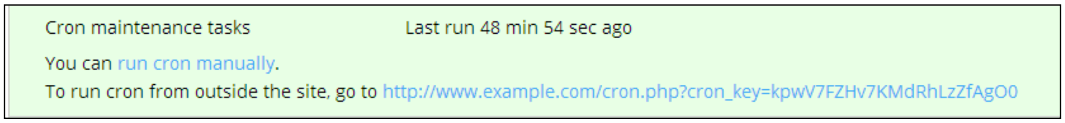
Value *

100

8. Click on the **Save** button to retain your settings.

> It may be wise to set a number other than zero in the **Number of shopping carts to process** (which will remove infinite carts), if your site is likely to have many unprocessed orders.

9. Shopping carts abandoned by customers within the time limit you set will now be removed when Drupal's cron runs.
10. You can trigger Drupal's cron yourself by navigating to **Site settings** | **Site reports** | **Status report**.



Cron maintenance tasks                Last run 48 min 54 sec ago

You can run cron manually.
To run cron from outside the site, go to http://www.example.com/cron.php?cron_key=kpwV7FZHv7KMdRhLzZfAgO0

11. Clicking on the **run cron manually** link will trigger the cron, and empty any carts abandoned in the last 7 days.

> For more information about Drupal 7 and cron, see `http://drupal.org/cron`.

## Setting a maximum and minimum order amount in Drupal Commerce

For certain types of e-commerce store, there may be a minimum or maximum order amount to make the store viable to run for the business. Drupal's **Commerce Minimum & Maximum Order Amount** module provides the functionality to allow you to set a minimum and/or maximum order amount, and this recipe will guide you through how to configure it for your Drupal Commerce store.

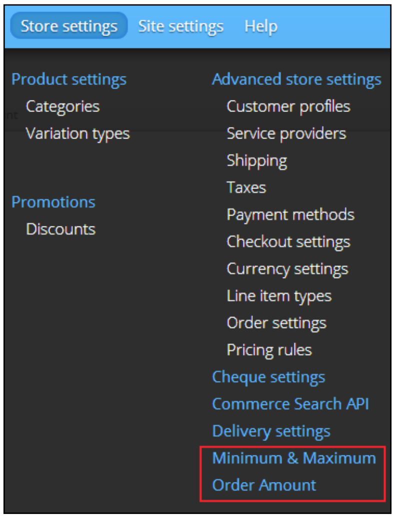
### Getting started

Download the **Commerce Minimum & Maximum Order Amount** module from Drupal.org (`http://drupal.org/project/commerce_moa`) and install it on your Drupal Commerce store.

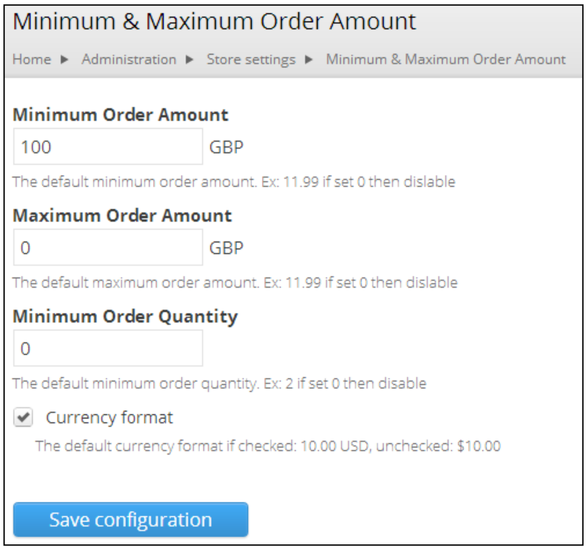
## How to do it...

Once installed, you can enable and configure the **Commerce Minimum & Maximum Order Amount** module through these steps:

1. Navigate to **Store settings** | **Minimum & Maximum Order Amount**, which should now appear in this screen's list of options:



2. Click on the **Minimum & Maximum Order Amount** option to see the module's settings:



Minimum & Maximum Order Amount

Home ▶ Administration ▶ Store settings ▶ Minimum & Maximum Order Amount

**Minimum Order Amount**

100 GBP

The default minimum order amount. Ex: 11.99 if set 0 then dislable

**Maximum Order Amount**

0 GBP

The default maximum order amount. Ex: 11.99 if set 0 then dislable

**Minimum Order Quantity**

0

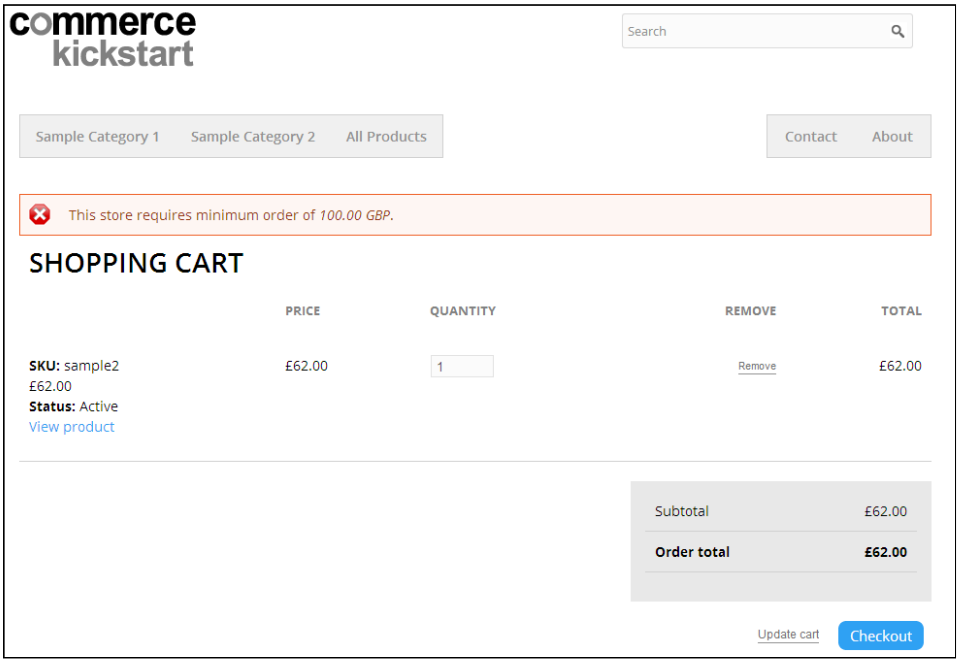
The default minimum order quantity. Ex: 2 if set 0 then disable

☑ Currency format

The default currency format if checked: 10.00 USD, unchecked: $10.00

Save configuration

3. As you can see, the module presents you with a number of options:
   - The **Minimum Order Amount** field allows you to set a minimum total order amount before customers are allowed to check out (leave this value as **0**, if you don't want to set a minimum order amount).
   - The **Maximum Order Amount** field allows you to set a maximum total order amount which, if exceeded, will prevent customers from checking out.
   - The **Minimum Order Quantity** field allows you to define a minimum quantity before customers can check out, as opposed to a minimum value of an order. So, for example, if you had this field set to **2** and the **Minimum Order Amount** field set to **100**, customers would not be able to check out from your store unless their total order value was $100 or above *and* they had more than two items in their shopping cart.
   - Finally, the **Currency format** field allows you to specify how the currency appears: as a symbol (for example, £ or $), or as the currency's abbreviation (for example, GBP or USD).
4. Save your settings with the **Save configuration** button and you can see the module in action by returning to the customer-facing frontend of your Drupal Commerce store.
5. Add items that total less than the amount you set in the **Minimum Order Amount** field in the previous step, and attempt to check out by clicking on the **Checkout** link. You will now see an error message appear that informs the user of the minimum order amount:



This store requires minimum order of *100.00 GBP*.

SHOPPING CART

| | PRICE | QUANTITY | REMOVE | TOTAL |
|---|---|---|---|---|
| **SKU:** sample2<br>£62.00<br>**Status:** Active<br>View product | £62.00 | 1 | Remove | £62.00 |

| | |
|---|---|
| Subtotal | £62.00 |
| **Order total** | **£62.00** |

That's it! You have configured your Drupal Commerce store to accept a minimum (or maximum) order amount.

# Displaying progress steps for checkout in Drupal Commerce

One of the many user interface patterns that appear on e-commerce websites is to show your customers how far they are from the checkout process, to help them manage their expectations and orientate them in your website's checkout page. With the help of the **Commerce Checkout Progress** module, you can add this feature to improve your website's experience for customers.

This recipe will take you through configuring the module to show your customers where they are in the checkout process to help them manage their expectations about how long it may take to complete their purchase.

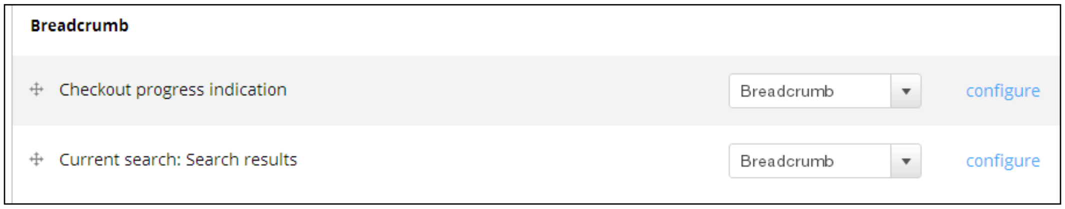
## Getting started

Download the Commerce Checkout Progress module from Drupal.org at `http://drupal.org/project/commerce_checkout_progress` and install the module on your Drupal Commerce store.

## How to do it...

Once you have installed the Commerce Checkout Progress module, follow these steps:

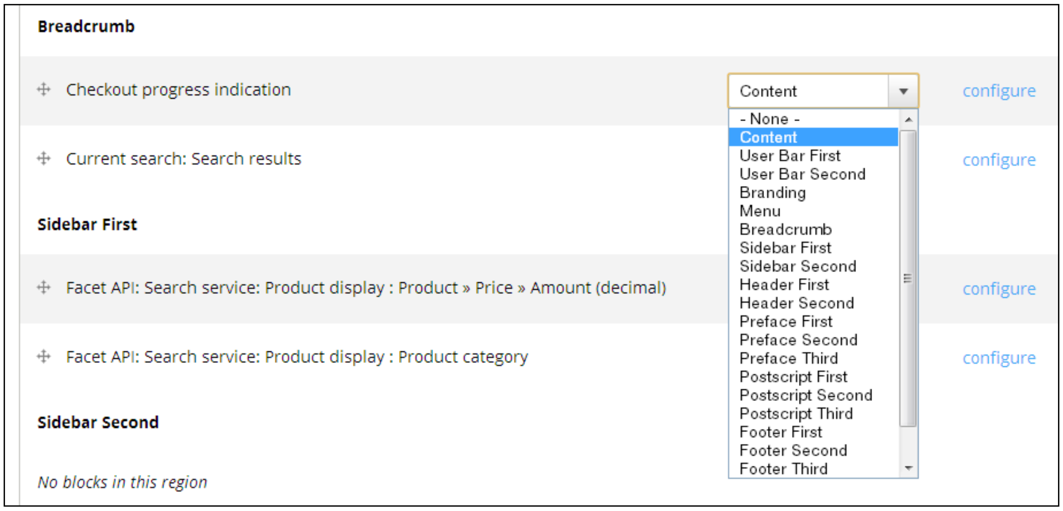
1. Enabling the module will expose a new Drupal block that you can position within your website's theme, so navigate to **Site settings** | **Blocks**.
2. By default, enabling the Commerce Checkout module will position the **Checkout progress indication** block in the **Breadcrumb** region of your Drupal website, above the **Current search: Search results** block:



**Breadcrumb**

| | | |
|---|---|---|
| ✣ Checkout progress indication | Breadcrumb ▾ | configure |
| ✣ Current search: Search results | Breadcrumb ▾ | configure |

3. If you want to position the **Checkout progress indication** block elsewhere, select another region to assign the block to from the drop-down list (the example moves the block to the **Content** region):



**Breadcrumb**

Checkout progress indication — Content — configure

Current search: Search results — configure

**Sidebar First**

Facet API: Search service: Product display : Product » Price » Amount (decimal) — configure

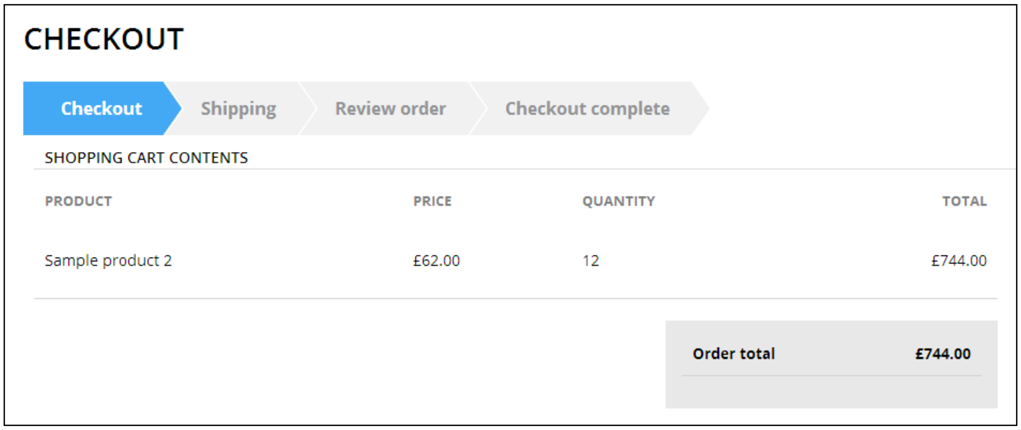
Facet API: Search service: Product display : Product category — configure

**Sidebar Second**

*No blocks in this region*

4. Click on the **Save blocks** button towards the bottom of the blocks list to save your new configuration.

5. If you now return to the customer-facing frontend of your Drupal Commerce store, you will see that the checkout progression indicator appears:



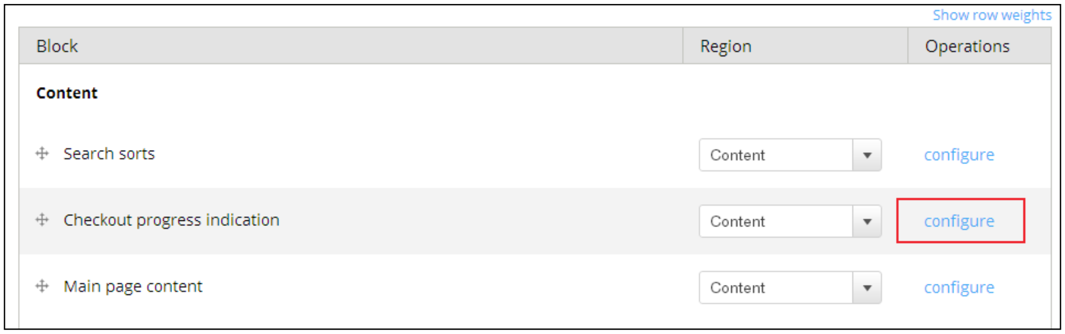
CHECKOUT

SHOPPING CART CONTENTS

| PRODUCT | PRICE | QUANTITY | TOTAL |
|---|---|---|---|
| Sample product 2 | £62.00 | 12 | £744.00 |

**Order total** **£744.00**

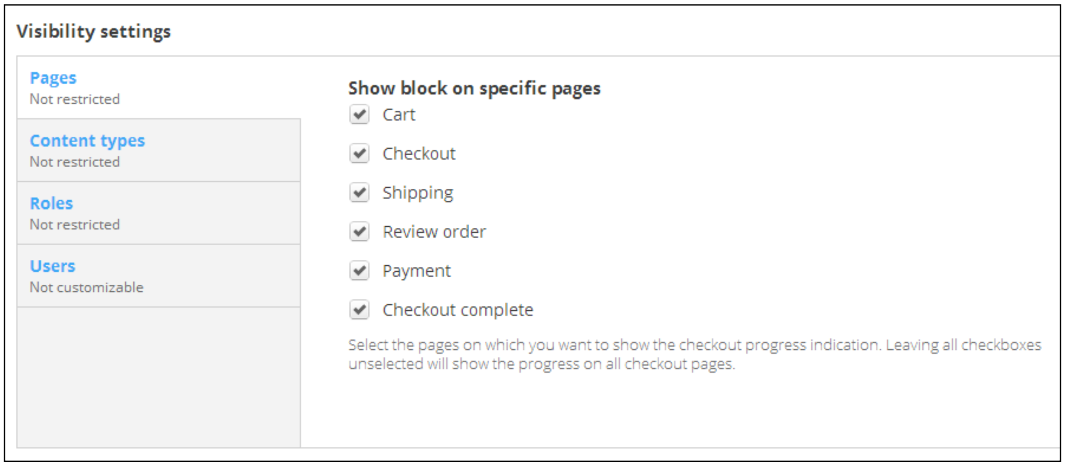
6. At the moment, however, this block only appears on the primary checkout page that you can see in the preceding screenshot. To display the block on all of the cart pages, navigate to **Site settings** | **Blocks** again in your Drupal administration interface.
7. Select the **configure** option to the right of the **Checkout progress indication** block:



Show row weights

| Block | Region | Operations |
|---|---|---|
| **Content** | | |
| Search sorts | Content ▾ | configure |
| Checkout progress indication | Content ▾ | configure |
| Main page content | Content ▾ | configure |

8. On the next screen, you are presented with **Visibility settings** beneath the **Region settings** block. You can click on the relevant checkboxes to enable this block to appear on the different checkout pages in your Drupal Commerce store:



**Visibility settings**

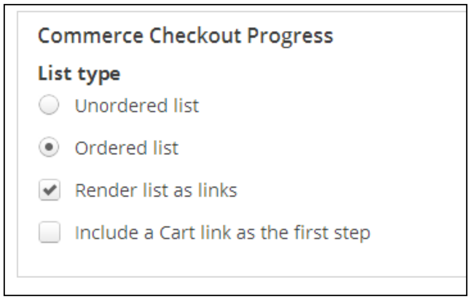
**Show block on specific pages**

☑ Cart
☑ Checkout
☑ Shipping
☑ Review order
☑ Payment
☑ Checkout complete

Select the pages on which you want to show the checkout progress indication. Leaving all checkboxes unselected will show the progress on all checkout pages.

9. Click on the **Save block** button, and that's it. Your customers can now see their progress through your checkout.

10. You can also customize the way the steps are displayed by clicking on the **configure** option next to the module in **Site settings** | **Modules**. You can change whether the cart steps are shown as a **Unordered list** or an **Ordered list**, whether to **Render list as links** (unchecking this will produce plain text rather than linked steps), and whether to **Include a Cart link as the first step** in the progress bar:



Commerce Checkout Progress

List type

○ Unordered list

● Ordered list

☑ Render list as links

□ Include a Cart link as the first step

## How it works...

The **Commerce Checkout Progress** module creates a block that is displayed in Commerce's checkout pages to display the progress of a customer's checkout to them. You can customize this module by deciding which region in your enabled Drupal theme this block appears in through **Site settings** | **Blocks interface**.

# Automatically assigning an SKU to your products in Drupal Commerce

As you will have seen by now, creating products in Drupal Commerce requires that you add a **Stock Keeping Unit** (**SKU**) to products to help you track and reference products within Drupal Commerce. This could get tedious if you're adding large amounts of products, which is where the AutoSKU module for Drupal Commerce comes in handy.

This recipe explains how to configure the AutoSKU module for Drupal Commerce.

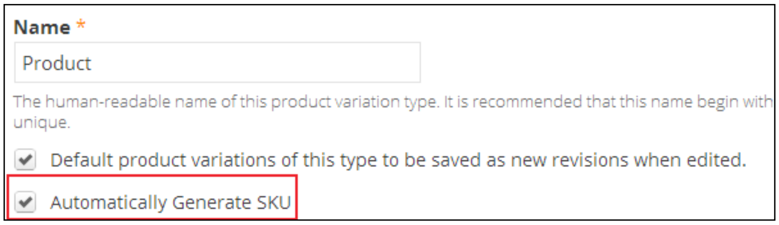
## Getting started

Download the Commerce AutoSKU module from Drupal.org at `http://drupal.org/project/commerce_autosku` and install the files on your Drupal Commerce store.

## How to do it...

Once you have installed the AutoSKU module for your store:

1. Go to **Products** | **Variation types** and select the **edit** link next to the product variation for which you want to enable the automatically generated SKUs.
2. At the bottom of the screen, you will now see a new option has appeared called **Automatically Generate SKU**. Enable this by clicking on the checkbox alongside it:



**Name** *

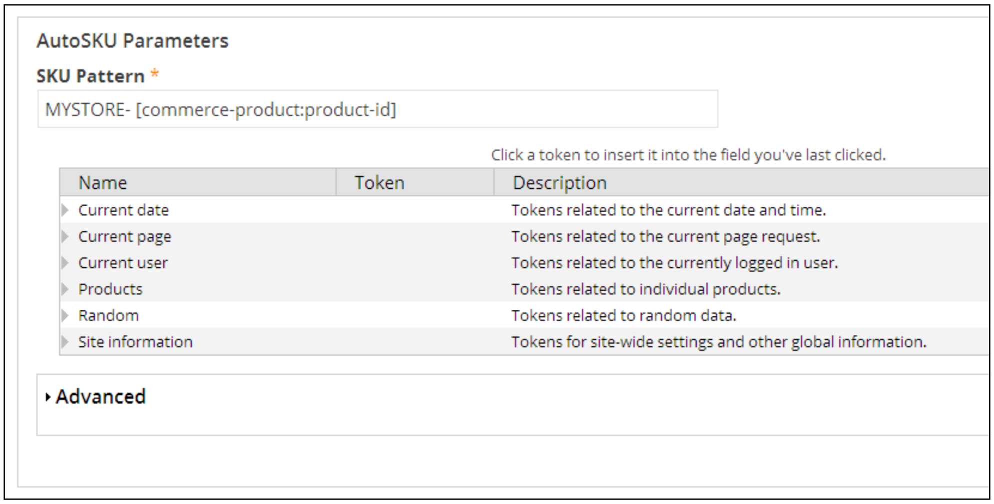
Product

The human-readable name of this product variation type. It is recommended that this name begin with
unique.

☑ Default product variations of this type to be saved as new revisions when edited.

☑ Automatically Generate SKU

3. When you enable the **Automatically Generate SKU** checkbox, the **AutoSKU parameters** options will appear with the **SKU Pattern** field, which enables you to automatically generate SKUs for your products that are created by patterns, which are controlled by Drupal tokens provided by Drupal Commerce. Using these tokens can give you great flexibility in generating your SKUs, but here are a few useful combinations that you may want to use:
    - To generate a simple automatic SKU for your products, enter something similar to `MYSTORE- [commerce-product:product-id]`, which would produce a result of `MYSTORE-99` for the product with an `ID` of `99`
    - To generate a more complex SKU that takes into account the product type and name of the product, you could use something such as: **[commerce-product:type] - [commerce-product:title]**



AutoSKU Parameters

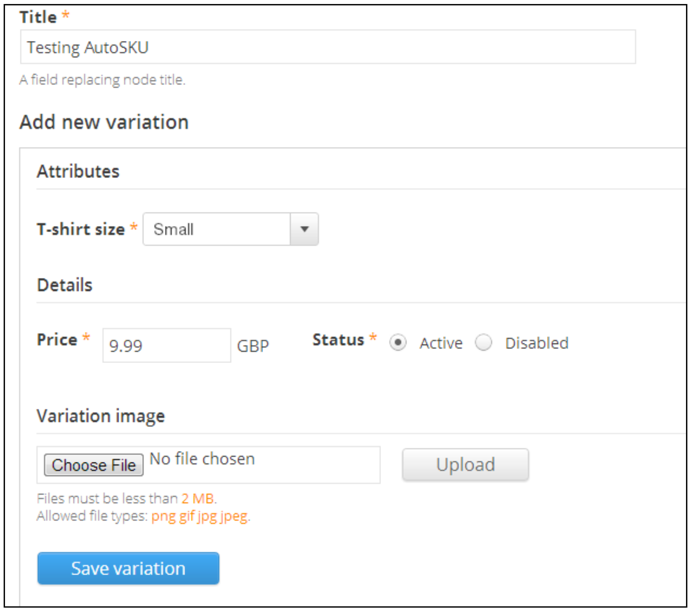
**SKU Pattern** *

MYSTORE- [commerce-product:product-id]

Click a token to insert it into the field you've last clicked.

| Name | Token | Description |
|---|---|---|
| Current date | | Tokens related to the current date and time. |
| Current page | | Tokens related to the current page request. |
| Current user | | Tokens related to the currently logged in user. |
| Products | | Tokens related to individual products. |
| Random | | Tokens related to random data. |
| Site information | | Tokens for site-wide settings and other global information. |

4. Once you're happy with your SKU pattern, click on the **Save product variation type** button at the bottom of the screen.
5. That's it! SKUs will be automatically be created for products of this type in your store now.
6. To test whether the module is functioning as you intended, navigate to **Products** | **Add a new product** and try adding a new product (of the type you just edited).
7. You'll see that the **SKU** field is now hidden, but the **Title**, **Image**, **Price**, **Status** and **Category** fields are all there:



Title * Testing AutoSKU

A field replacing node title.

Add new variation

Attributes

T-shirt size * Small

Details

Price * 9.99 GBP

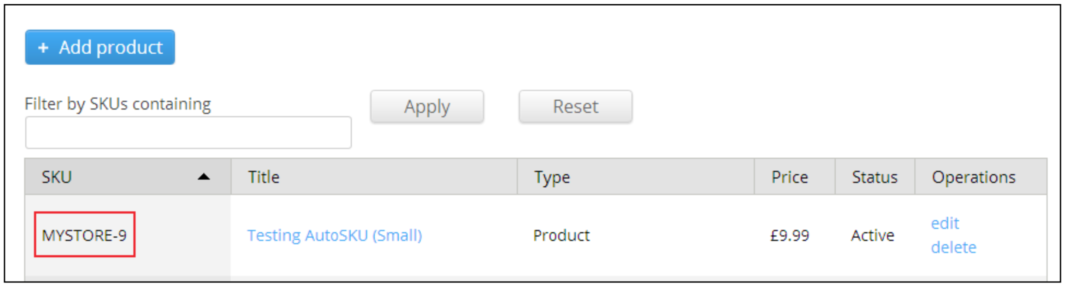
Status * ● Active ○ Disabled

Variation image

No file chosen

Files must be less than 2 MB.
Allowed file types: png gif jpg jpeg.

8. Save the product using the **Save variation** button, and then the **Save** button towards the bottom of the screen to finalize your new product.
9. If you navigate to **Products** | **Manage products**, you'll see that the newly created product has had its SKU generated for you:



Filter by SKUs containing

| SKU | Title | Type | Price | Status | Operations |
|---|---|---|---|---|---|
| MYSTORE-9 | Testing AutoSKU (Small) | Product | £9.99 | Active | edit<br>delete |

# 5

# Payment and Shipping Modules for Drupal Commerce

This chapter guides you through integrating some of the most popular payment gateways with your Drupal Commerce website, as well as setting up shipping settings for your store. We will cover the following topics in this chapter:

- Integrating Authorize.net with Drupal Commerce
- Integrating PayPal Website Payments Standard with Drupal Commerce
- Integrating PayPal Website Payments Pro with Drupal Commerce
- Integrating SagePay Form Integration with Drupal Commerce
- Setting up VAT or sales tax in Drupal Commerce

## Introduction

One of the most fundamental features of an e-commerce website is to accept payment from your customers, and to this end, Drupal Commerce integrates with many of the most well-known and used payment gateways in the world.

# Integrating Authorize.net with Drupal Commerce

Authorize.net is an American-based payment gateway, and allows you to take payments via credit and debit cards commonly available in North America, as well as worldwide.

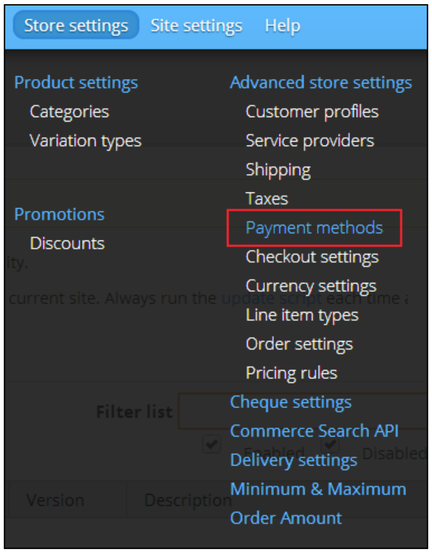
## Getting started

Download the Drupal Commerce Authorize.net module from `Drupal.org` at `http://drupal.org/project/commerce_authnet`. Install the module to your Drupal Commerce store.

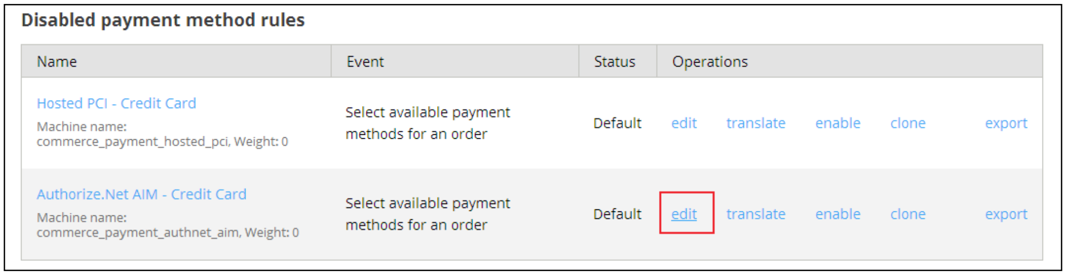
## How to do it...

Once you have installed the Drupal Commerce Authorize.net module perform these steps:

1. Now navigate to **Store settings** | **Advanced store settings** | **Payment methods**:



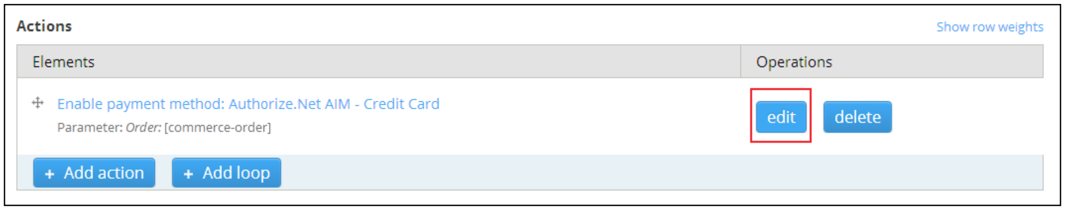
2. You will now be presented with two lists. At the top is the **Enabled payment method rules** list, and underneath this is the **Disabled payment method rules** list, which now contains the **Authorize.Net Aim - Credit Card** option.
3. Click on the **edit** link under the **Operations** column:



**Disabled payment method rules**

| Name | Event | Status | Operations | | | | |
|---|---|---|---|---|---|---|---|
| Hosted PCI - Credit Card<br>Machine name: commerce_payment_hosted_pci, Weight: 0 | Select available payment methods for an order | Default | edit | translate | enable | clone | export |
| Authorize.Net AIM - Credit Card<br>Machine name: commerce_payment_authnet_aim, Weight: 0 | Select available payment methods for an order | Default | edit | translate | enable | clone | export |

4. You will now see the **Events**, **Conditions**, and **Actions** options for the Authorize.Net payment method. Here too, click on the **edit** link under the **Operations** column:

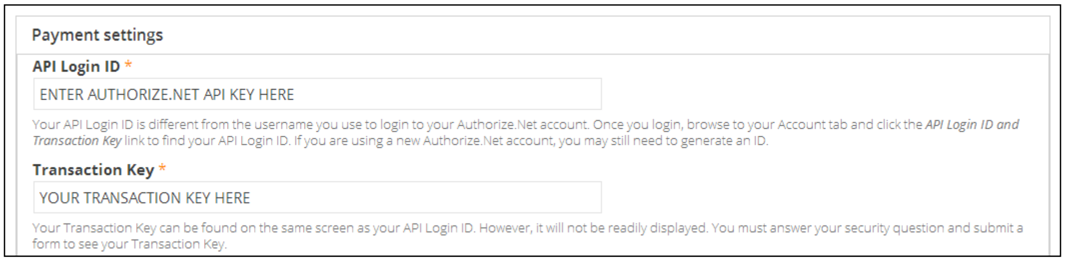


**Actions**

| Elements | Operations |
|---|---|
| Enable payment method: Authorize.Net AIM - Credit Card<br>Parameter: *Order:* [commerce-order] | edit delete |

5. This provides further options to enable the Authorize.Net gateway. Here you will need your Authorize.Net payment gateway API and transaction key to finish the configuration. If you need an Authorize.Net account, you can register for one through `http://commerceguys.com/authorize-net-reseller` which helps support the development of Drupal Commerce.

> If you need to use Authorize.Net SIM or DPM support, there's a module better suited to your needs. You can find it at `http://drupal.org/project/commerce_authnet_simdpm`.



**Payment settings**

**API Login ID** *

ENTER AUTHORIZE.NET API KEY HERE

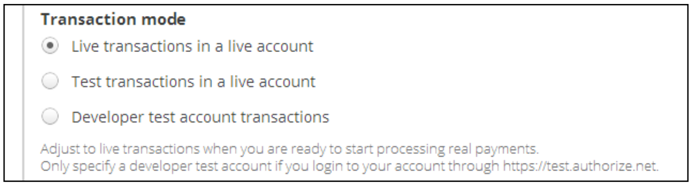
Your API Login ID is different from the username you use to login to your Authorize.Net account. Once you login, browse to your Account tab and click the ***API Login ID and Transaction Key*** link to find your API Login ID. If you are using a new Authorize.Net account, you may still need to generate an ID.

**Transaction Key** *

YOUR TRANSACTION KEY HERE

Your Transaction Key can be found on the same screen as your API Login ID. However, it will not be readily displayed. You must answer your security question and submit a form to see your Transaction Key.

6. For testing  purposes, leave the **Transaction mode** section set to **Test transactions in a live account** to test with a normal Authorize.Net account, or to **Developer test account transactions** to test the transactions without a normal Authorize. Net account.
7. Once you're ready to launch your website, be sure to set this value to **Live transactions in a live account**:



**Transaction mode**

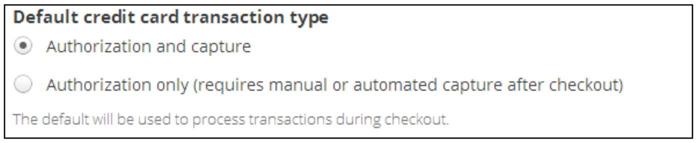
● Live transactions in a live account

○ Test transactions in a live account

○ Developer test account transactions

Adjust to live transactions when you are ready to start processing real payments.
Only specify a developer test account if you login to your account through https://test.authorize.net.

8. The option that follows, **Default credit card transaction type**, specifies which authorization type Authorize.Net will use:



**Default credit card transaction type**

● Authorization and capture

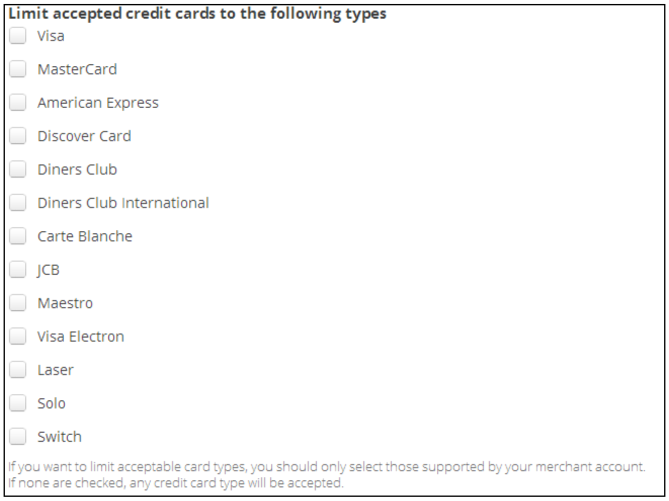
○ Authorization only (requires manual or automated capture after checkout)

The default will be used to process transactions during checkout.

9. The default **Authorization and capture** option simply takes the amount that is due and, if the transaction is approved by Authorize.Net, it is then submitted for payment (**settlement** in the vocabulary of Authorize.Net).
10. The **Authorization only** option essentially checks whether the payment details are valid, but does not attempt to submit them for payment, so you will largely want to leave this setting as is. In some countries, you cannot legally capture the funds until you have shipped the order, so in these cases you would want to use **Authorization only**.

> For more information on these, see `http://www.authorize.net/support/merchant/Submitting_Transactions/Credit_Card_Transaction_Types.htm`.

11. The next options allow you to limit the type of credit cards your website accepts through Authorize.Net with the **Limit accepted credit cards to the following types** field (leaving all of these unchecked allows your Authorize.Net payment method to accept any of these credit card types):



**Limit accepted credit cards to the following types**

□ Visa

□ MasterCard

□ American Express

□ Discover Card

□ Diners Club

□ Diners Club International

□ Carte Blanche

□ JCB

□ Maestro

□ Visa Electron

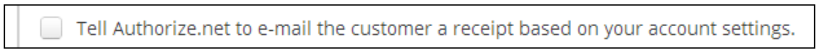
□ Laser

□ Solo

□ Switch

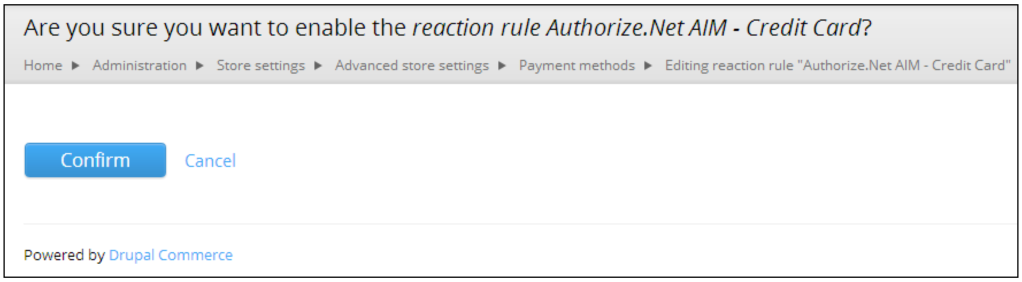
If you want to limit acceptable card types, you should only select those supported by your merchant account. If none are checked, any credit card type will be accepted.

12. Below this, you will see another option (**Tell Authorize.net to e-mail the customer a receipt  based on your account settings**), which e-mails the customer a payment receipt upon successful payment via Authorize.Net based on your Authorize.Net account settings:



□ Tell Authorize.net to e-mail the customer a receipt based on your account settings.

13. Logging the following messages for debugging settings allows you to log request and/ or response messages from the Authorize.Net API. Look in the **Reports Site settings** | **Advanced settings** | **Reports** | **Recent log messages** for the log.
14. Click on the **Save** button to retain your settings for the Authorize.Net payment gateway.
15. To finish enabling the payment method, return to **Store settings** | **Configuration Advanced settings** | **Payment methods**, and click on **enable** in the **Operations** column.
16. This will produce a message dialogue that will ask you if you are sure you want to enable this payment gateway; click on the **Confirm** button to confirm:



Are you sure you want to enable the *reaction rule Authorize.Net AIM - Credit Card*?

Home ▸ Administration ▸ Store settings ▸ Advanced store settings ▸ Payment methods ▸ Editing reaction rule "Authorize.Net AIM - Credit Card"
Confirm
Cancel
Powered by Drupal Commerce

17. That's it! Authorize.Net is enabled for your Drupal Commerce store. You may still need to configure your Authorize.Net account, though.

> When using Authorize.Net as your payment gateway, you will need to ensure that the **Payment** pane of the checkout process is *after* the **Billing Information** pane, so that Authorize.Net can process the payment properly.

# Integrating PayPal Website Payments Standard with Drupal Commerce

PayPal is one of the world's most popular payment solutions, so it should come as no surprise that Drupal Commerce has a well developed and supported module to integrate with it. There are two available PayPal payment methods supported with the Commerce PayPal module: **Website Payments Standard** (**WPS**) and **Website Payments Pro** (**WPP**).

> PayPal Website Payments Standard is in the process of being renamed to PayPal Payments Standard, so you may see some discrepancy in documentation and module names.

The benefit of this method is that there is no monthly cost; you simply pay per transaction for processing the payment. So, let's get PayPal Website Payments Standard integrated with your Drupal Commerce site.

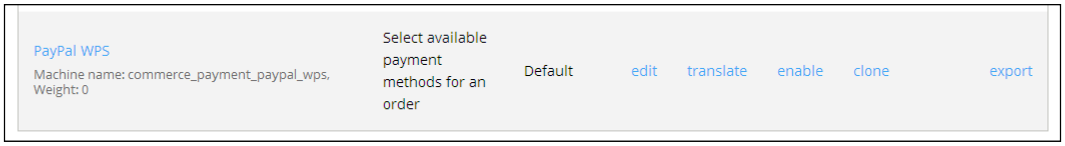
## Getting started

Download the Drupal Commerce PayPal module from `Drupal.org` at `http://drupal.org/project/commerce_paypal` and install it on your Drupal Commerce installation.

## How to do it...

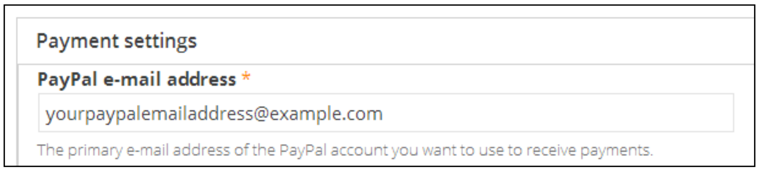
Once the module is installed, log in as an administrator on your Drupal Commerce website and perform the following steps:

1. Navigate to **Store settings** | **Configuration Advanced store settings** | **Payment methods** to see the list of payment methods currently available in your store:



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| PayPal WPS<br>Machine name: commerce_payment_paypal_wps,<br>Weight: 0 | Select available payment methods for an order | Default | edit | translate | enable | clone | export |

2. You will see the **PayPal WPS payment method rule** list displayed under **Disabled payment method rules**.
3. Select the **edit** link in the **Operations** column to edit this payment method.
4. You will now see the **Enable payment method: PayPal WPS** link listed under **Actions**. Click on the **edit** link under the **Operations** tab.
5. The next screen presents options to help you configure PayPal Website Payments Standard method for your Drupal Commerce store. Look for the **Payment Settings** block.
6. In this block, you will see the **PayPal e-mail address** field. Enter your PayPal account e-mail address. This is a required field, as payments can't be processed without your PayPal account's e-mail address.



Payment settings

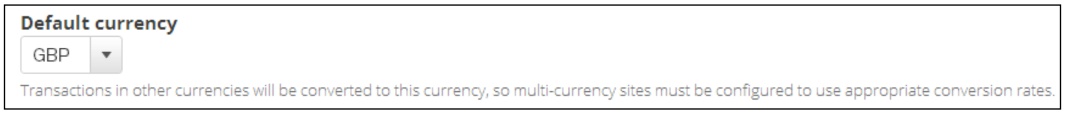
PayPal e-mail address *

yourpaypalemailaddress@example.com

The primary e-mail address of the PayPal account you want to use to receive payments.

> Need a PayPal account? Sign up at `http://www.paypal.com` before continuing.

7. The next field is the **Default currency** setting; by default, this is set to **USD**, but you may find it preferable to change it to your own PayPal account's default currency:

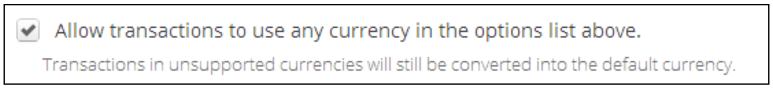


Default currency

GBP

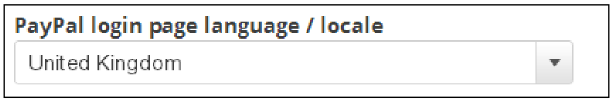
Transactions in other currencies will be converted to this currency, so multi-currency sites must be configured to use appropriate conversion rates.

8. The next option is the **Allow transactions to use any currency in the options list above** field. Checking this box allows payments to be made to your PayPal account in all currencies supported by PayPal. If unchecked, payments made in other currencies are converted to your default currency.



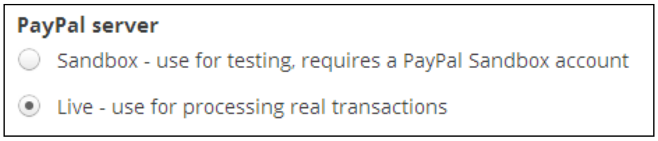
☑ Allow transactions to use any currency in the options list above.
Transactions in unsupported currencies will still be converted into the default currency.

9. The **PayPal login page language/locale** field allows you to specify the language and region displayed to users when they are taken to PayPal to make payment to you. Set this value to your country to better suit your customers:



PayPal login page language / locale
United Kingdom

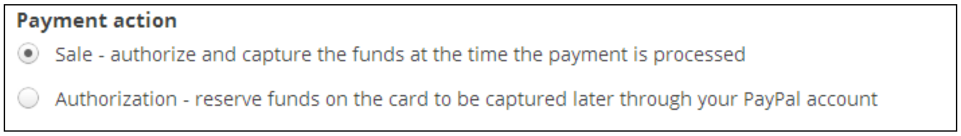
10. The **PayPal server** setting allows you to specify whether transactions are live—payments are really made—or whether they are test transactions. By default, this value is set to **Sandbox - use for testing, requires a PayPal Sandbox account**, and needs to be changed to **Live - use for processing real transactions**:



PayPal server
○ Sandbox - use for testing, requires a PayPal Sandbox account
● Live - use for processing real transactions

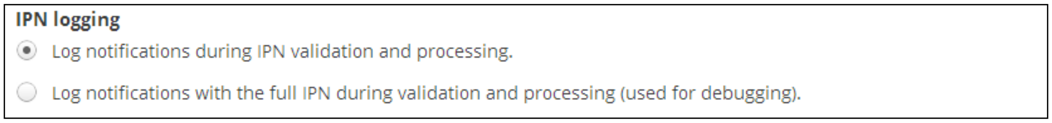
> You can sign up for a PayPal sandbox account for testing transactions on your store at `https://developer.paypal.com`.

11. The **Payment action** setting provides options to process the payment immediately, or simply to authorize payment for the amount to be paid later to you. Most stores will use the **Sale - authorize and capture the funds at the time the payment is processed** value, or simply authorize the payment amount to be collected later.



Payment action
● Sale - authorize and capture the funds at the time the payment is processed
○ Authorization - reserve funds on the card to be captured later through your PayPal account

12. The **IPN logging** setting which appears next tells the module whether to log **Instant Payment Notifications** (**IPNs**). By default for your live store, leave the value as **Log notifications during IPN validation and processing**, but you may find the **Log notifications with the full IPN during validation and processing (used for debugging)** setting useful to aid in debugging any problems with PayPal.

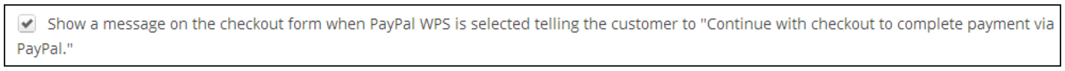


IPN logging

● Log notifications during IPN validation and processing.

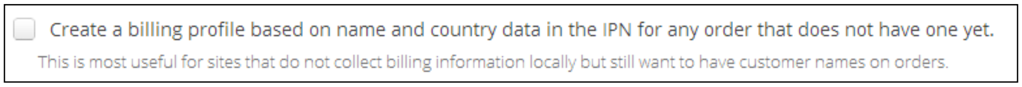
○ Log notifications with the full IPN during validation and processing (used for debugging).

The **Show a message on the checkout form when PayPal WPS...** option displays a message telling the customer to continue with checkout to complete payment via PayPal:



☑ Show a message on the checkout form when PayPal WPS is selected telling the customer to "Continue with checkout to complete payment via PayPal."

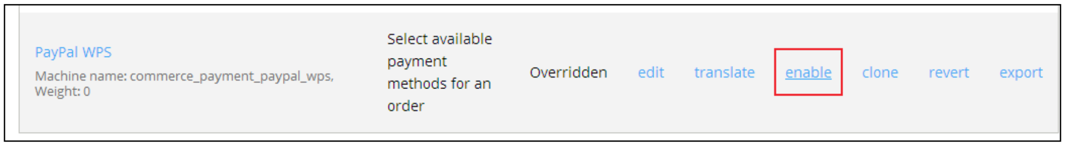
13. Finally, the **Create a billing profile...** option generates a customer billing profile in Drupal Commerce with the customer's name and country on it if checked, which may be useful if you need to use customer names on orders you dispatch:



□ Create a billing profile based on name and country data in the IPN for any order that does not have one yet.

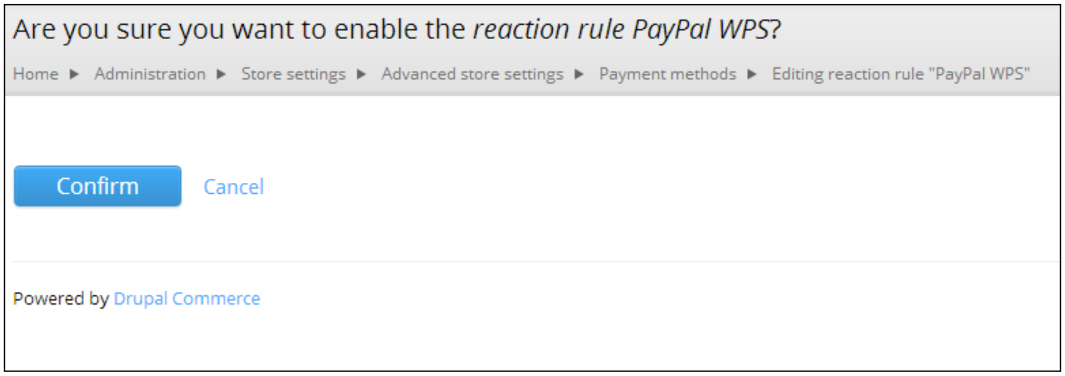
This is most useful for sites that do not collect billing information locally but still want to have customer names on orders.

14. Click on the **Save** button to retain your new PayPal settings. Navigate back to **Store settings** | **Advanced store settings** | **Payment methods** and click on the **enable** button alongside the **PayPal WPS** payment method to finish enabling it on your Drupal Commerce store:



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| PayPal WPS<br>Machine name: commerce_payment_paypal_wps, Weight: 0 | Select available payment methods for an order | Overridden | edit | translate | enable | clone | revert | export |

Click on the **Confirm** button in the dialogue that now appears:

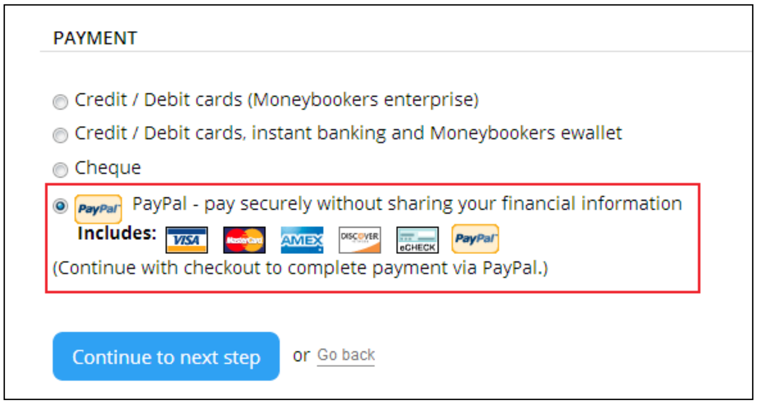


Are you sure you want to enable the *reaction rule PayPal WPS*?

Home ▶ Administration ▶ Store settings ▶ Advanced store settings ▶ Payment methods ▶ Editing reaction rule "PayPal WPS"

Powered by Drupal Commerce

15. If you now progress through the checkout of your store, you will see that PayPal appears as a payment option once you come to review your order:



PAYMENT

○ Credit / Debit cards (Moneybookers enterprise)

○ Credit / Debit cards, instant banking and Moneybookers ewallet

○ Cheque

● PayPal - pay securely without sharing your financial information
Includes:
(Continue with checkout to complete payment via PayPal.)

16. That's it! PayPal Website Payments Standard is enabled on your Drupal Commerce store.

# Integrating PayPal Website Payments Pro with Drupal Commerce

The alternative to using the PayPal Website Payments Standard payment method is to use PayPal Website Payments Pro (also known as PayPal Website Payments Advanced). WPP doesn't require your customers to leave your website to make payments. They can pay with PayPal while on your website. As a more advanced service, you will need to have a PayPal Business Account.

You can enable both Website Payments Pro and Website Payments Standard in your store to enable customers to chose their preferred payment method.

## Getting started

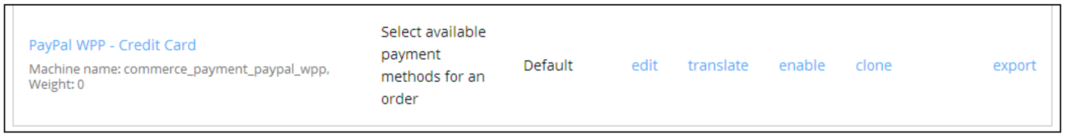
Download the Drupal Commerce PayPal module from `Drupal.org` at `http://drupal.org/project/commerce_paypal` and install the module on your Drupal Commerce store.

If you need to sign up to PayPal for a business account to make use of Website Payments Pro, do so at `http://www.paypal.com`.

## How to do it...

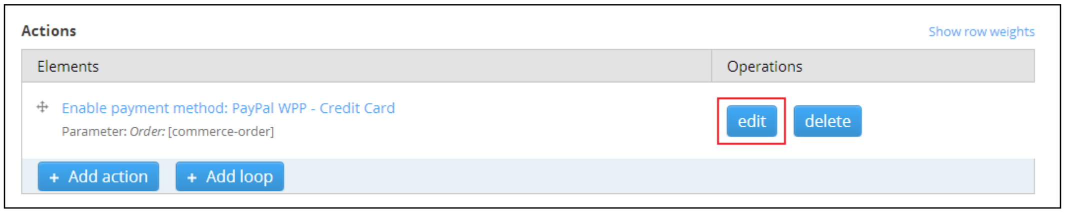
Once you have installed the PayPal Website Payments Pro module, follow these steps:

1. Navigate to **Store settings** | **Advanced store settings** | **Payment methods** and you will see **PayPal WPP - Credit Card** appear under **Disabled payment method rules**:



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| PayPal WPP - Credit Card<br>Machine name: commerce_payment_paypal_wpp, Weight: 0 | Select available payment methods for an order | Default | edit | translate | enable | clone | export |

2. Click on **edit** to go to the **Editing reaction rule "PayPal WPP - Credit Card"** screen.
3. Under the **Actions** block, you will see the **Enable payment method: PayPal WPP - Credit Card** element. Click on **edit** under the **Operations** column:

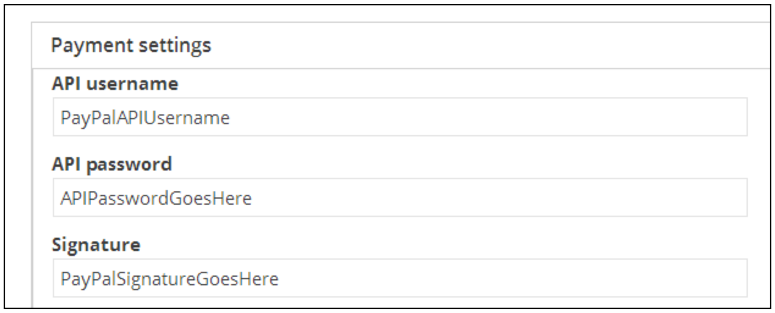


Actions

| Elements | Operations |
|---|---|
| Enable payment method: PayPal WPP - Credit Card<br>Parameter: *Order:* [commerce-order] | edit delete |

4. You can now see the settings for this payment method in the **Payment Settings** pane.

5. Enter your **API username**, **API password**, and **Signature** values from your PayPal account:



Payment settings

API username

PayPalAPIUsername

API password

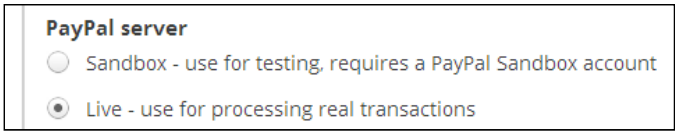
APIPasswordGoesHere

Signature

PayPalSignatureGoesHere

> To find these values see `http://www.putler.com/support/faq/how-to-get-paypal-api-username-password-and-signature-information/`.

6. The next setting, **PayPal server**, allows you to specify whether you want to use the Sandbox PayPal server for testing, or the Live PayPal server for taking payments from genuine customers:



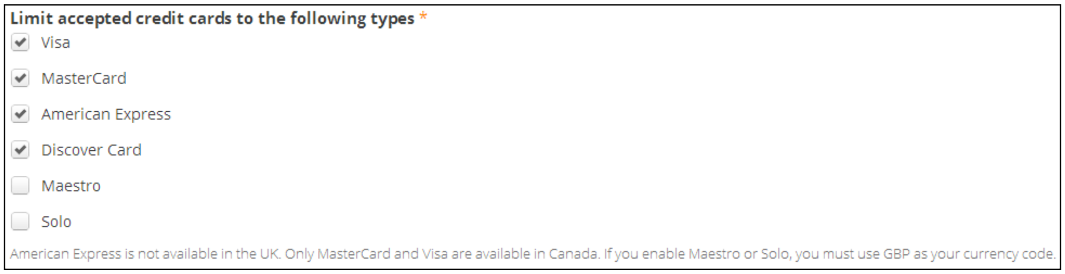
PayPal server

○ Sandbox - use for testing, requires a PayPal Sandbox account

● Live - use for processing real transactions

> You can sign up for a PayPal Sandbox account for testing transactions on your store at `https://developer.paypal.com`.

7. Next, the **Limit accepted credit cards to the following types** field allows you to select which credit cards your store will accept:



Limit accepted credit cards to the following types *

☑ Visa

☑ MasterCard

☑ American Express

☑ Discover Card

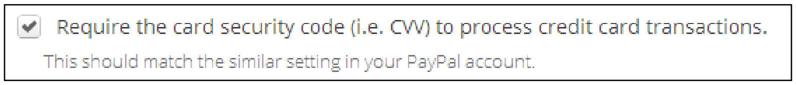
☐ Maestro

☐ Solo

American Express is not available in the UK. Only MasterCard and Visa are available in Canada. If you enable Maestro or Solo, you must use GBP as your currency code.

> Note the restrictions in regions noted underneath: **American Express is not available in the UK. Only MasterCard and Visa are available in Canada. If you enable Maestro or Solo, you must use GBP as your currency code.**

8. You can also specify whether to ask your customers to use the CVV card security code with this payment method with the **Require the card security code (i.e. CVV) to process credit card transactions** field:



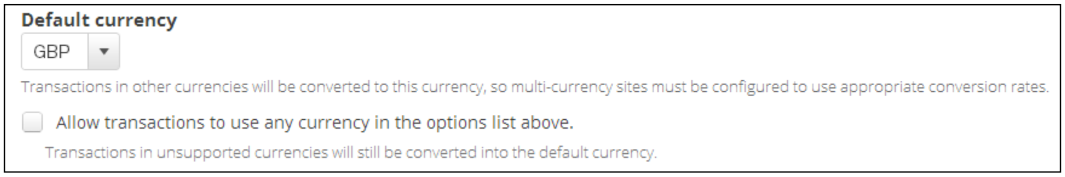
☑ Require the card security code (i.e. CVV) to process credit card transactions.

This should match the similar setting in your PayPal account.

9. As the note beneath the field says, this value needs to match what you have set in your PayPal account settings to prove to PayPal that this is authorized use of the PayPal account! Once logged in to your PayPal account, navigate to **Profile** | **My Selling Tools** and select **Update** next to **Managing risk and fraud**.
10. Click on **Edit Your Filter Settings** and check the boxes next to CVV to enable or disable it to match the value you have defined in the Drupal Commerce module. Click on the **Save** button.

> Note that your PayPal account must have been upgraded to Business and been fully approved for PayPal Payments Pro. If your account is simply **Approved With Restrictions**, the preceding options may not be visible to you.

11. Back in Drupal Commerce's administration panel, you can return to the settings. The **Default currency** field defines the currency type that is used by the PayPal transactions from your store as default, so it is usually best to change this to your own country's currency:



**Default currency**

GBP ▾

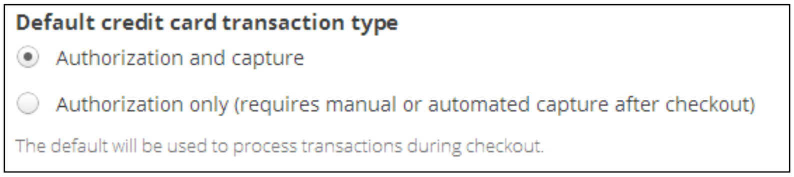
Transactions in other currencies will be converted to this currency, so multi-currency sites must be configured to use appropriate conversion rates.

☐ Allow transactions to use any currency in the options list above.

Transactions in unsupported currencies will still be converted into the default currency.

12. The option below the **Default currency** field is **Allow transactions to use any currency in the options list above** and checking it allows your customers to pay in their own currency, with PayPal converting it to your currency (PayPal takes an additional commission for this transfer).

13. The next setting is **Default credit card transaction type**. The **Authorization and capture** option authorizes payment from the customer's account immediately and transfers money to your PayPal account. The **Authorization only (requires manual or automated capture after checkout)** option verifies that the customer has a valid credit card and can afford the payment, but does not yet take payment.



Default credit card transaction type

● Authorization and capture

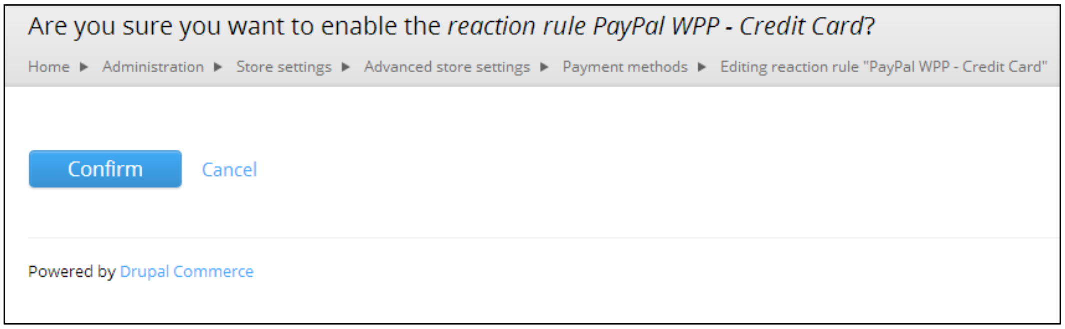
○ Authorization only (requires manual or automated capture after checkout)

The default will be used to process transactions during checkout.

14. Finally, the **Log the following messages for debugging** setting allows you to define whether you log messages for requests and/or responses to and from the PayPal server, which can be useful for debugging problems with your store.

> To view these log messages, go to **Site settings** | **Advanced settings** | **Reports Site reports** | **Recent log messages**.

15. Click on the **Save** button and navigate to **Store** | **Configuration** | **Payment methods** again.
16. Click on **enable** next to the **PayPal WPP - Credit Card** option, and then click on the **Confirm** button in the corresponding dialog:

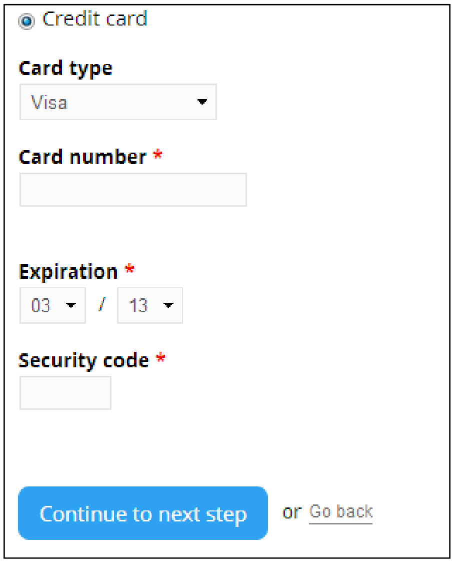


Are you sure you want to enable the *reaction rule PayPal WPP - Credit Card?*

Home ▶ Administration ▶ Store settings ▶ Advanced store settings ▶ Payment methods ▶ Editing reaction rule "PayPal WPP - Credit Card"
Confirm
Cancel

Powered by Drupal Commerce

If you now attempt to check out from your store, you will see the **Credit card** payment method appears, powered by PayPal:



● Credit card

Card type

Visa

Card number *

Expiration *

03 / 13

Security code *

Note that your website will need a security certificate to secure the transfer of data for your customers. Ask your host for information on SSL certificates and see `http://en.wikipedia.org/wiki/Transport_Layer_Security#SSL_1.0.2C_2.0_and_3.0` for more information.

# Integrating SagePay Form Integration with Drupal Commerce

SagePay is a popular payment gateway based in the UK, which can process credit and debit cards much like popular providers such as PayPal and Authorize.net.

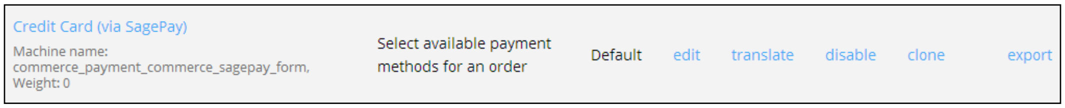
## Getting started

Download the Commerce SagePay from Drupal.org at `http://drupal.org/project/commerce_sagepay_form` and install it on your Drupal Commerce store. You will also need a SagePay account set up, which you can do at `http://www.sagepay.com`.

## How to do it...

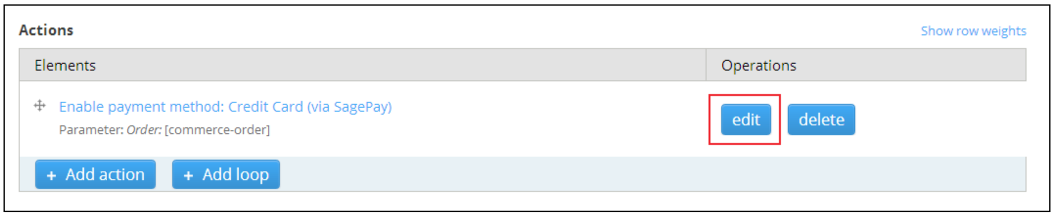
Once you've installed the module, follow these steps to enable SagePay for your Drupal Commerce store:

1. Navigate to **Store settings** | **Advanced store settings** | **Payment methods** and locate the SagePay module in the list:



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Credit Card (via SagePay)<br>Machine name: commerce_payment_commerce_sagepay_form, Weight: 0 | Select available payment methods for an order | Default | edit | translate | disable | clone | export |

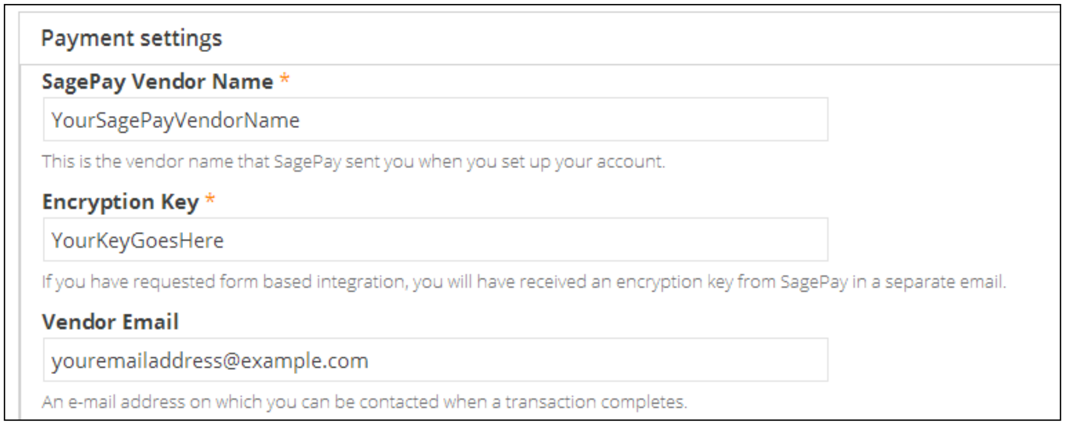
2. Click on the **edit** link to alter the module's settings, and then click on the **edit** link once again under the **Actions** block:



Actions

| Elements | Operations |
|---|---|
| Enable payment method: Credit Card (via SagePay)<br>Parameter: *Order:* [commerce-order] | edit delete |

3. Under the **Payment settings** block on the next screen, enter your **SagePay Vendor Name**, **Encryption Key**, and **Vendor Email** associated with your SagePay account:



Payment settings

SagePay Vendor Name *

YourSagePayVendorName

This is the vendor name that SagePay sent you when you set up your account.

Encryption Key *

YourKeyGoesHere

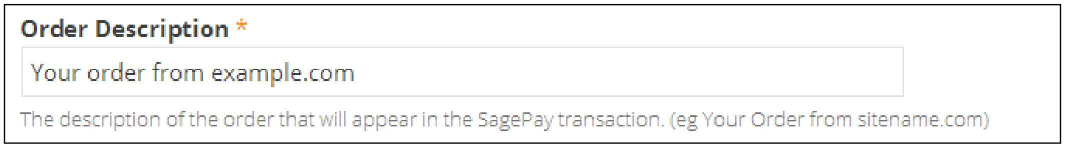
If you have requested form based integration, you will have received an encryption key from SagePay in a separate email.

Vendor Email

youremailaddress@example.com

An e-mail address on which you can be contacted when a transaction completes.

4. Next, the **Order Description** field allows you to define text that appears on a customer's account statements. This is shown for any products ordered through your store and paid for with the SagePay payment gateway, so try and keep it generic, and mention your website's URL or company name here.

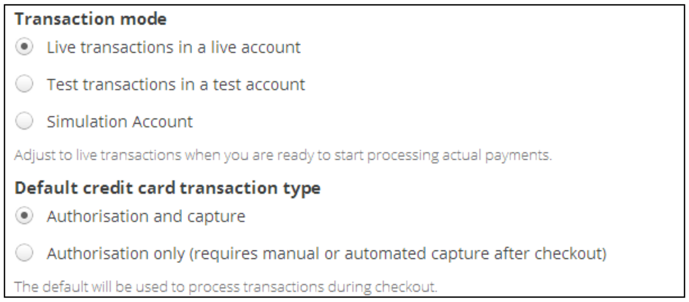


Order Description *

Your order from example.com

The description of the order that will appear in the SagePay transaction. (eg Your Order from sitename.com)

5. The **Transaction mode** field allows you to specify whether the payment method is in live or test mode, and the **Default credit card transaction type** field specifies how the payment is taken (that is, whether your store takes payment directly, or whether you want to take payment later on manually with SagePay):



Transaction mode

● Live transactions in a live account

○ Test transactions in a test account

○ Simulation Account

Adjust to live transactions when you are ready to start processing actual payments.

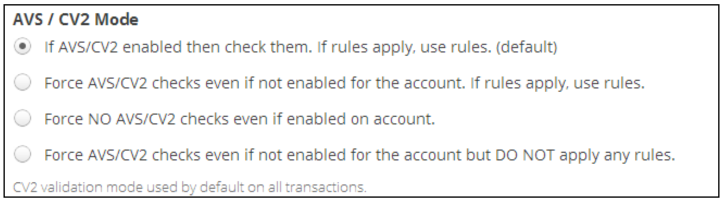
Default credit card transaction type

● Authorisation and capture

○ Authorisation only (requires manual or automated capture after checkout)

The default will be used to process transactions during checkout.

6. For more advanced configuration, the **AVS / CV2 Mode** settings allow you to tell Drupal Commerce how to behave if the *verification* numbers on cards are used to verify card holders when making payments to you.



AVS / CV2 Mode

● If AVS/CV2 enabled then check them. If rules apply, use rules. (default)

○ Force AVS/CV2 checks even if not enabled for the account. If rules apply, use rules.

○ Force NO AVS/CV2 checks even if enabled on account.

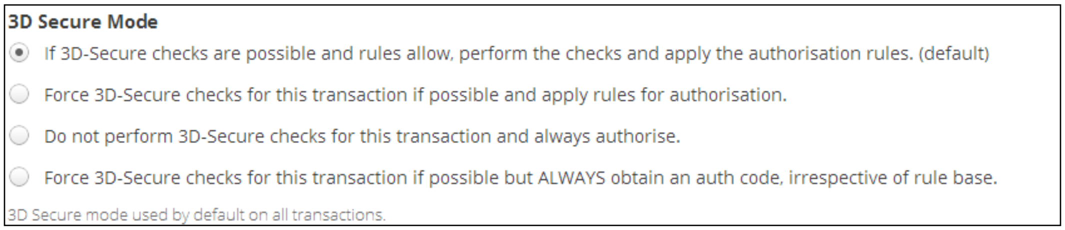
○ Force AVS/CV2 checks even if not enabled for the account but DO NOT apply any rules.

CV2 validation mode used by default on all transactions.

It is good practice to have AVS/CV2 enabled on your store to help prevent fraudulent transactions. If you select the **Force AVS/CV2 checks even if not enabled for the account** option, AVS/CV2 checks will be undertaken even if you haven't enabled these with your SagePay account. If you have rules in your Drupal Commerce store that may affect the functioning of this, you can choose to ignore them, or enforce these rules.

> Beware that some payment gateways and credit card providers may be penalized if you do not check the AVS/CV2 value for transactions. Note that AVS is a UK-only fraud prevention scheme.
>
> For more information on these settings for SagePay, see `http://www.sagepay.com/help/userguides/6338?page=5`.

7. Similarly, the **3D Secure Mode** settings add another layer of card holder verification to transactions.



**3D Secure Mode**

● If 3D-Secure checks are possible and rules allow, perform the checks and apply the authorisation rules. (default)

○ Force 3D-Secure checks for this transaction if possible and apply rules for authorisation.

○ Do not perform 3D-Secure checks for this transaction and always authorise.

○ Force 3D-Secure checks for this transaction if possible but ALWAYS obtain an auth code, irrespective of rule base.

3D Secure mode used by default on all transactions.

The first option in the preceding screenshot ensures the SagePay module undertakes the 3D-Secure check for the transaction, and then performs any authorization rules set in your SagePay account.

The second option tells the SagePay module to attempt a 3D-Secure check if possible, and apply any rules for authorization set in your SagePay account.

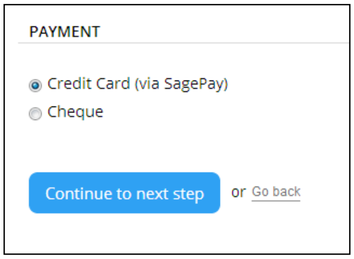
The third option effectively disables 3D-Secure checks for all transactions.

The final option above will make the SagePay module attempt to run a 3D-Secure check on the transaction, if possible, but even if no 3D-Secure check is run, an authorization code from SagePay will always be sought before payment is taken.

> Processing payments without 3D-Secure enabled will incur additional transaction fees, so it's worth leaving this enabled!

8. Finally, the **Send cart contents to SagePay** field allows your store to send the individual items a customer has ordered to SagePay instead of just the final cart total. This may be of use if you use Sage for your business' accounting.

9. Click on **Save**, and enable the payment method if it is not already active. You can now see the SagePay payment method available at checkout:



PAYMENT

● Credit Card (via SagePay)

○ Cheque

# Setting up VAT or a sales tax in Drupal Commerce

Some countries have a standing sales tax—**value added tax** (**VAT**)—applied to sales and contributes towards the country's taxation income.

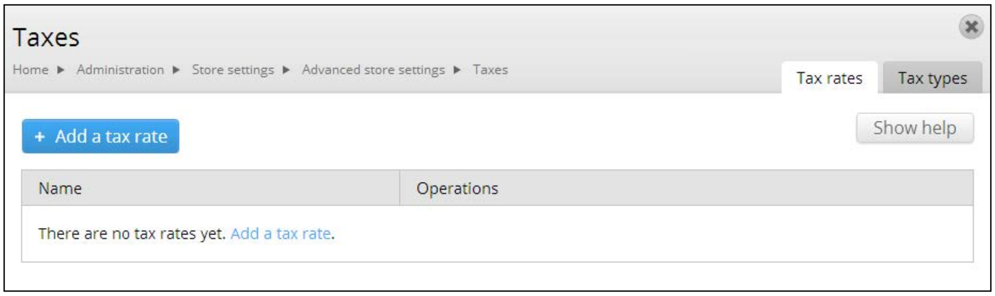
## Getting started

Ensure you are logged in to the Drupal administration panel as an administrator.

## How to do it...

Setting up VAT or a sales tax for your Drupal Commerce store doesn't require a module. You can use the core features of Drupal Commerce to achieve this.

1. Navigate to **Store settings** | **Advanced store settings** | **Taxes**.
2. Create a new tax rate by clicking on the **Add a tax rate** link:

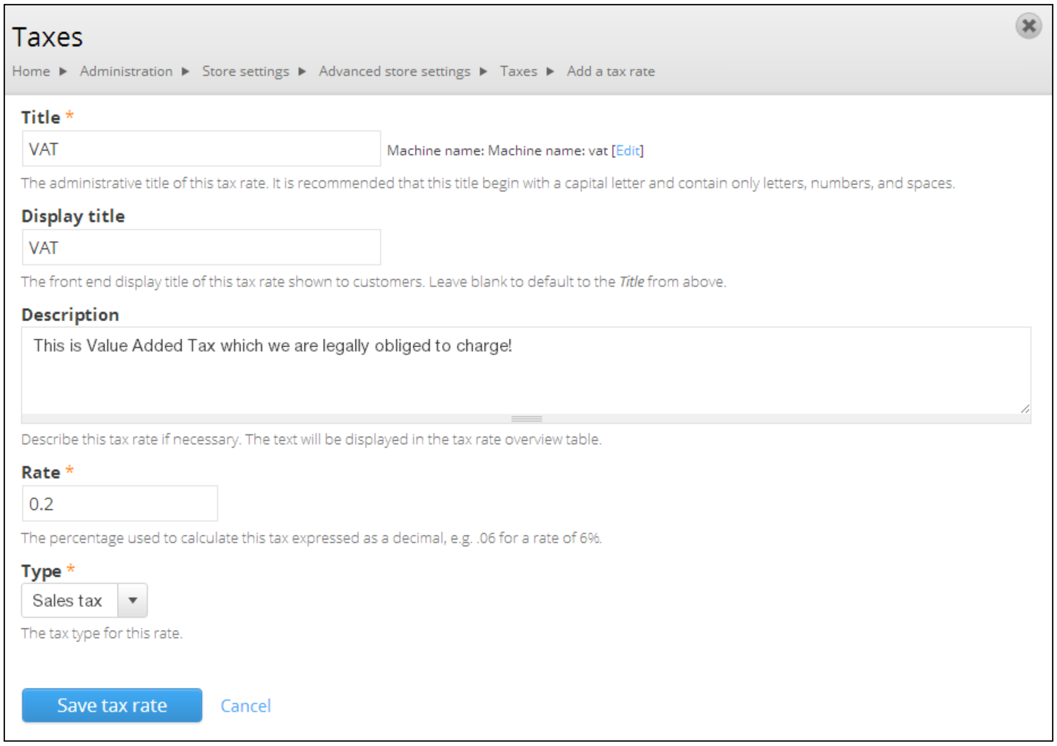


Taxes

Home ▸ Administration ▸ Store settings ▸ Advanced store settings ▸ Taxes
Tax rates
Tax types
+ Add a tax rate
Show help

| Name | Operations |
|---|---|
| There are no tax rates yet. Add a tax rate. | |

3. Enter a name for your tax in the **Title** field:



Taxes

Home ▶ Administration ▶ Store settings ▶ Advanced store settings ▶ Taxes ▶ Add a tax rate

Title *

VAT

Machine name: Machine name: vat [Edit]

The administrative title of this tax rate. It is recommended that this title begin with a capital letter and contain only letters, numbers, and spaces.

Display title

VAT

The front end display title of this tax rate shown to customers. Leave blank to default to the *Title* from above.

Description

This is Value Added Tax which we are legally obliged to charge!

Describe this tax rate if necessary. The text will be displayed in the tax rate overview table.

Rate *

0.2

The percentage used to calculate this tax expressed as a decimal, e.g. .06 for a rate of 6%.

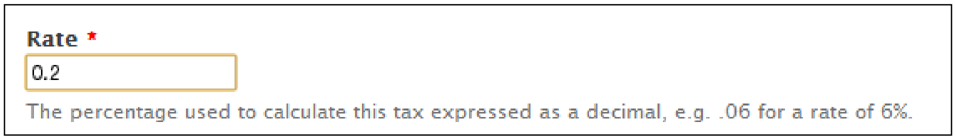
Type *

Sales tax

The tax type for this rate.

4. Enter **Display title** too if you want this to appear to customers differently from the **Title** value, and add a description which is used for administrators only.

5. Next, enter a value for the **Rate** field. This is the rate of the tax to be applied expressed as a decimal value, so 20 percent tax would be entered as 0.2 and 17 percent tax would be entered as 0.17.

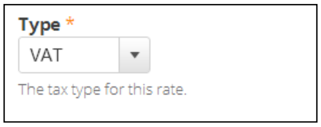


Rate *

0.2

The percentage used to calculate this tax expressed as a decimal, e.g. .06 for a rate of 6%.

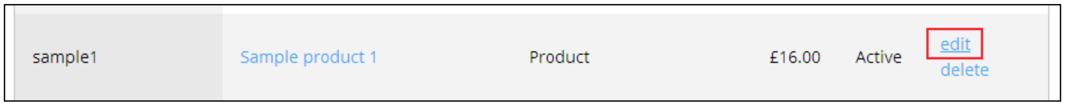
6. The next option is **Type**, which specifies how the tax is applied at checkout for customers. If the value is set as **Sales tax**, the tax is displayed as an addition to the cost. If the value is set to **VAT**, the tax is displayed in the total cost of the products being ordered.



Type *
VAT
The tax type for this rate.

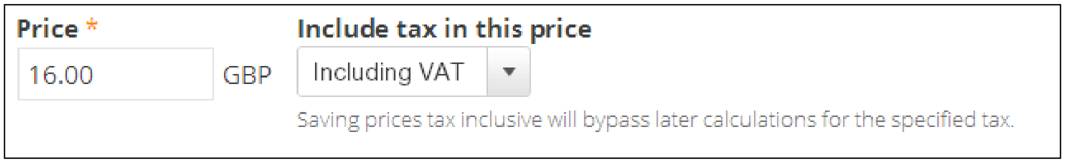
You can configure how the tax is displayed in your store to customers by clicking on the **Tax types** tab towards the top-right corner of your screen in the **Store settings** | **Configuration Advanced store settings** | **Taxes** view screen.

7. You can now click on **Save tax rate** to retain these changes.
8. To change the way tax is applied to a particular product, navigate to **Store** | **Products** and select a product to edit by clicking on the **edit** option in the **Operations** column:



| | | | | | |
|---|---|---|---|---|---|
| sample1 | Sample product 1 | Product | £16.00 | Active | edit<br>delete |

9. Locate the **Price** field in the consequent screen you see, and you should now see the **Include tax in this price** field:



Price *
16.00 GBP

Include tax in this price
Including VAT
Saving prices tax inclusive will bypass later calculations for the specified tax.

10. Setting this value to **Including VAT** (or the name of your tax) will alter the way tax is dealt with for this product and ignore any other tax calculations you have set up later in the checkout process.

# 6

# Drupal Commerce Search Engine Optimization

This chapter covers tips and techniques to improve your Drupal Commerce website's search engine friendliness, including:

- Using search engine friendly URLs for your products
- Verifying your Drupal Commerce website with Bing and Google
- Creating an XML sitemap for your Drupal Commerce store
- Adding a meta description field to your Drupal Commerce product display
- Integrating Google Authorship with your Drupal Commerce store

## Introduction

The chances are that your e-commerce website has hundreds of competitors, and it's vital that your store is as search engine friendly as possible. These tips and tricks may help to give your Drupal Commerce store that additional boost it needs in search engine rankings to help drive more customers to your store over your competitors.

# Using search engine friendly URLs for your products

Perhaps one of the easiest and yet most overlooked way to improve your website for search engines is to provide search engine friendly URLs (web addresses) for your product pages. Luckily, Drupal Commerce makes this easy for you to do!

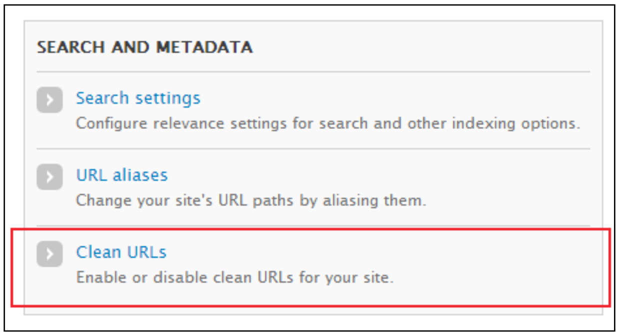
## Getting started

Log in to your Drupal Commerce store's administration panel.

## How to do it...

Once you have logged in as an administrator, follow these steps:

1. Navigate to **Site settings** | **Configuration**.
2. Select the **Clean URLs** option from the **SEARCH AND METADATA** box:



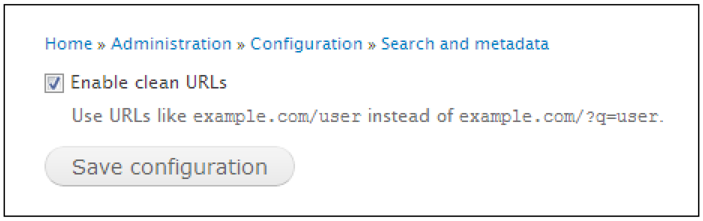
SEARCH AND METADATA

Search settings
Configure relevance settings for search and other indexing options.

URL aliases
Change your site's URL paths by aliasing them.

Clean URLs
Enable or disable clean URLs for your site.

3. Ensure that the **Enable clean URLs** field is checked:



Home » Administration » Configuration » Search and metadata

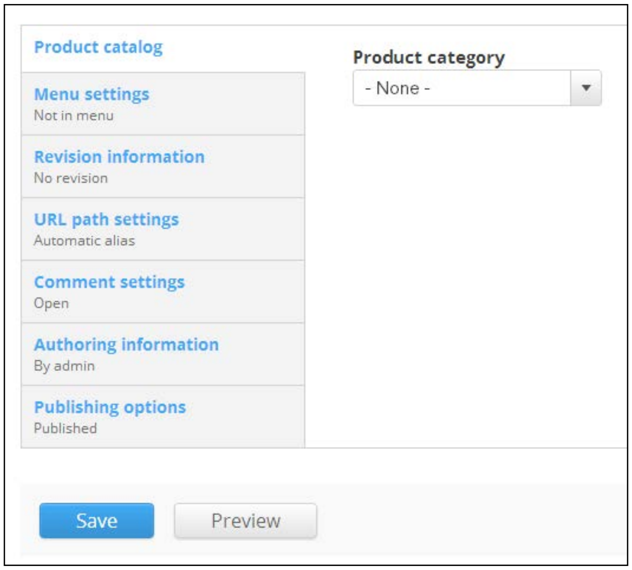
☑ Enable clean URLs

Use URLs like `example.com/user` instead of `example.com/?q=user`.

4. Finally, click on the **Save configuration** button to save the setting.

> For more help enabling search engine friendly URLs for your Drupal Commerce site (especially if you're having problems on a shared server), see `http://drupal.org/getting-started/clean-urls`.

5. Now that clean URLs are enabled in your Drupal Commerce store, you can add and edit your products and give them clean web addresses.
6. Navigate to **Products** | **Add a product**, and select the type of product that you want to create.
7. Complete the next screen to add a product. If you scroll towards the bottom of the screen, you will see further options displayed in tabs:



Product catalog

Menu settings
Not in menu

Revision information
No revision

URL path settings
Automatic alias

Comment settings
Open

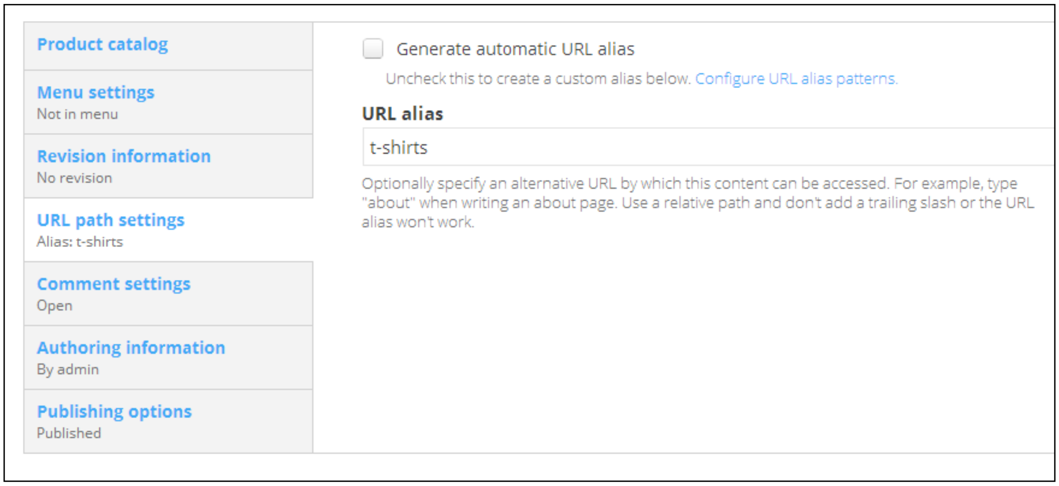
Authoring information
By admin

Publishing options
Published

Product category: - None -

8. Click on the **URL path settings** tab to select it, and you are presented with the URL alias field where you're able to enter a custom path for your product.

9. Since the product in this example is a t-shirt, the value **t-shirts** is used here (you may need to uncheck the **Generate automatic URL alias** checkbox if you've enabled this feature):



□ Generate automatic URL alias

Uncheck this to create a custom alias below. Configure URL alias patterns.

URL alias

t-shirts

Optionally specify an alternative URL by which this content can be accessed. For example, type "about" when writing an about page. Use a relative path and don't add a trailing slash or the URL alias won't work.

10. When creating your custom URL path, bear in mind that:

    - You can't add a slash character (/) at the end of your URL as this will prevent Drupal from displaying the page. You can use slash characters in the middle of your URL alias, though: so `products/t-shirts` is fine, but `products/t-shirts/` is not.
    - You don't need to add the full (absolute) URL to your product (for example, `http://www.example.com/t-shirts` or `www.example.com/t-shirts`). This will confuse Drupal and your custom URL will not work.
    - It is best practice to avoid using spaces in this field; replace any spaces that you want with either an underscore or a hyphen character, so that `fantastic t-shirts` might become `fantastic-t-shirts` or `fantastic_t-shirts`.
    - Avoid uppercase letters in your URL, so ensure your URL is `t-shirts` and not `T-Shirts`. While there is nothing wrong with capital letters in your site's URLs, it's easier for consistency's sake, and helps to prevent duplicate URLs on your site, too.

## How it works...

By default, Drupal assigns a unique number to your product display nodes (pages), and they will appear similar to the following:

```
http://www.example.com/node/123
```

Search engines prefer URLs that contain words that relate to the page's (or product's) content, and by using the URL alias field when you create products in your store, you can present content in your store with more search engine friendly URLs, such as the following:

```
http://www.example.com/t-shirts
```

# Verifying your Drupal Commerce website with Bing and Google

Search engines often provide tools that allow you to access webmaster information including information about any errors found on your website, such as missing pages and any duplicate content. Verifying your sites with services such as Bing and Google also allows you to submit XML sitemaps to search engines manually, as you will see in another recipe in this chapter.

This recipe will guide you through how you can use a Drupal module, Site Verification, to verify your Drupal Commerce with search engines such as Bing and Google for greater insight into your website's visitors. Verifying your website with search engines also allows you to use tools to see errors with your website, which you can fix, improving your website for customers.

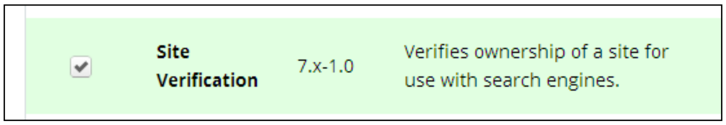
## Getting ready

You will need to download the Site Verification module from Drupal.org at `http://drupal.org/project/site_verify`. Install this module in your Drupal Commerce installation.

## How to do it...

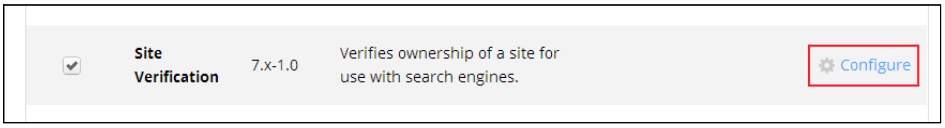
Once you have installed the Site Verification module, follow these steps:

1. Navigate to **Site settings** | **Advanced settings** | **Modules** in your website's administration panel to enable the module. You'll find the module grouped under **Other heading**:



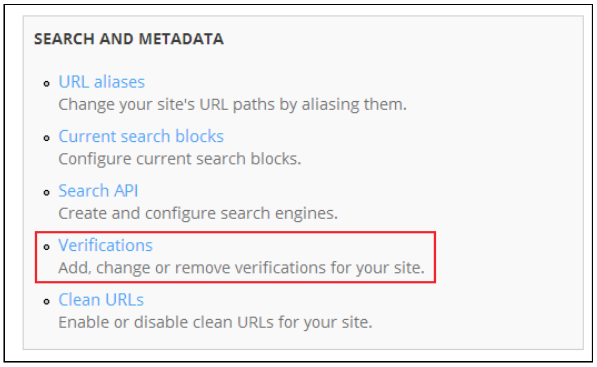
| | | | |
|---|---|---|---|
| ☑ | **Site Verification** | 7.x-1.0 | Verifies ownership of a site for use with search engines. |

2. Click on the **Save configuration** button to save the newly enabled module's state.
3. There are two common methods that search engines use to verify that you own a website:
   - Uploading a custom HTML file to your website. Once the file is there, the search engine checks whether it exists, proving you own the website (as presumably someone who owns the website will have access to upload files to it).
   - Adding a custom meta element to your website's `<head>` element. Again, this is similar to the preceding approach; here search engines assume that anyone who has access to alter the HTML of your website will "own" the website.
4. The Site Verification module allows you to manage both of these verification methods without having to alter your individual theme files, making it more "upgrade-proof" than other ways of doing this.
5. Once the module is enabled, navigate to **Site settings** | **Advanced settings** | **Modules** in the Drupal Commerce administration panel again, and click on the configure link that has appeared to the right of the module's name and description now that it has been enabled:



| | | | | |
|---|---|---|---|---|
| ☑ | **Site Verification** | 7.x-1.0 | Verifies ownership of a site for use with search engines. | ⚙ Configure |

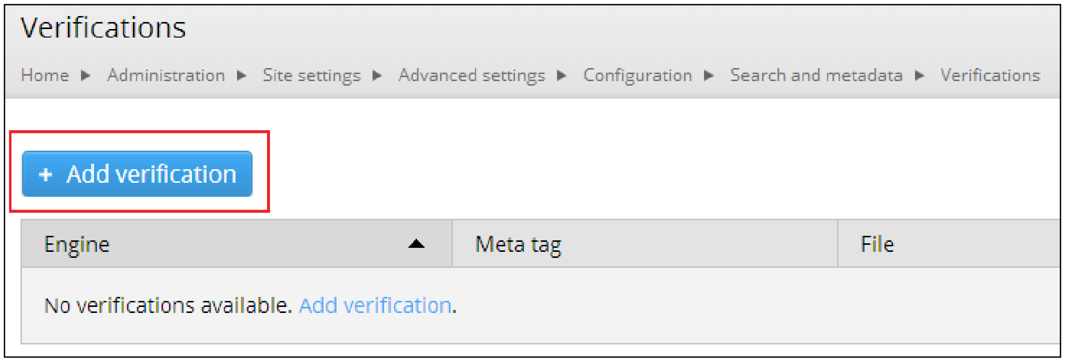
6. You are now presented with a new option for **Verifications** (you may need to clear Drupal's cache at **Site settings** | **Configuration** | **Performance** to see this):



SEARCH AND METADATA

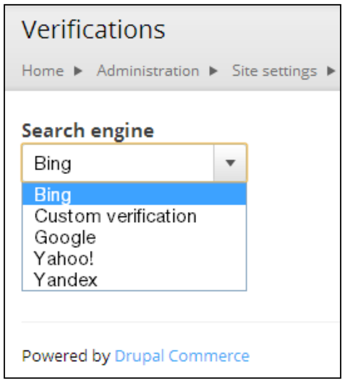
- URL aliases
  Change your site's URL paths by aliasing them.
- Current search blocks
  Configure current search blocks.
- Search API
  Create and configure search engines.
- Verifications
  Add, change or remove verifications for your site.
- Clean URLs
  Enable or disable clean URLs for your site.

7. Click on the **Verifications** link to manage the verification files and metadata for your site. Click on the **Add verification** link towards the top-left corner of your screen:


Verifications
Home ▶ Administration ▶ Site settings ▶ Advanced settings ▶ Configuration ▶ Search and metadata ▶ Verifications
+ Add verification
Engine
Meta tag
File

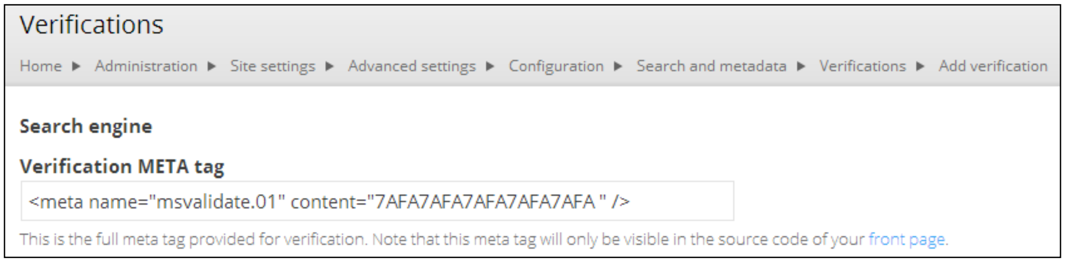
No verifications available. Add verification.

8. You're now presented with a drop-down list of the search engine formats to which you can add verification. This example will use Bing as an example format to verify your website for, so select the **Bing** value from the **Search engine** drop-down list and click on the **Next** button to proceed:



9. You will need to copy the verification tag provided by Bing's control panel onto your clipboard to use in the next step.

10. You are now given the chance to enter either **Verification META tag** or **Upload an existing verification file**. If pasting a verification metatag here, post the full line, including the `meta` element at the start:


Verifications
Home ▶ Administration ▶ Site settings ▶ Advanced settings ▶ Configuration ▶ Search and metadata ▶ Verifications ▶ Add verification
Search engine

Verification META tag

```
<meta name="msvalidate.01" content="7AFA7AFA7AFA7AFA7AFA" />
```

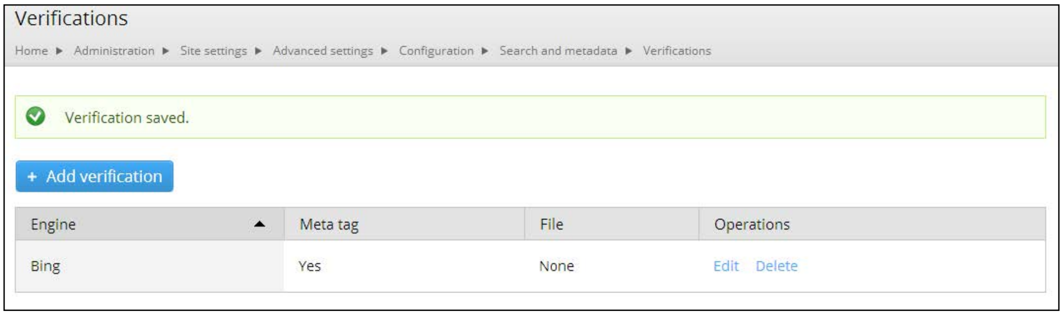
This is the full meta tag provided for verification. Note that this meta tag will only be visible in the source code of your front page.

11. This is then displayed on the homepage of your Drupal Commerce website only (though you may again need to refresh Drupal's cache in the **Site settings** | **Configuration** | **Performance** menu before you see it).
12. Viewing the source of the page will reveal the following now appearing in your page's `<head>` element:

    ```
    <meta name="msvalidate.01" content="7AFA7AFA7AFA7AFA7AFA " />
    ```

> Ensure that this value matches what the Bing (or Google) webmaster tools provides you with.

13. That's it! You will see that your new metatag (or verification file) is listed here at `admin/config/search/verifications`:


Verifications
Home ▸ Administration ▸ Site settings ▸ Advanced settings ▸ Configuration ▸ Search and metadata ▸ Verifications
+ Add verification

✓ Verification saved.

| Engine ▲ | Meta tag | File | Operations |
|---|---|---|---|
| Bing | Yes | None | Edit Delete |

# Creating an XML sitemap for your Drupal Commerce store

Part of making your Drupal Commerce store search engine friendly is making your content easy to find for search engines, and one good way to do this is to generate an XML sitemap of your website's pages, which tells search engine the following:

- The address of every page that you would like them to consider including.
- The relative importance of each page in your website compared to other pages in your website. Your homepage may be more important than your terms and conditions page, for example.
- The frequency with which each page on your website is updated.

## Getting started

Firstly, you may find it useful to create a View in your Drupal Commerce store that lists all products in your store), as online XML sitemap generators work based upon the links they find between pages on your website. To create a View that lists all products in your Drupal Commerce store, see `http://drupal.org/node/863852`. Next, find an online XML sitemap generator such as `http://www.xml-sitemaps.com` or `http://www.xmlsitemapgenerator.org`.

## How to do it...

Once you have generated the XML sitemap:

1. The XML sitemap generator will produce a file that looks similar to the following code:

   ```
   <urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.sitemaps.org/schemas/sitemap/0.9
   http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd">
   <url>
   <loc>http://www.example.com/</loc>
   <changefreq>weekly</changefreq>
   <priority>1</priority>
   </url>
   <url>
   <loc>http://www.example.com/t-shirts</loc>
   <changefreq>daily</changefreq>
   <priority>0.8</priority>
   </url>
   <!-- some entries omitted -->
   </urlset>
   ```

2. Each page is contained within a `<url>` element that itself contains a `<loc>` element that identifies the location of the page. There are a number of optional fields, too; the `<changefreq>` element tells the search engine how often this page is likely to change, and the `<priority>` element tells the search engine how important it is for you to view this page as on your website in comparison to other pages on your website. Note that the priority element does not affect how your website ranks against other websites.

> For more information on the XML sitemap format, see `http://www.sitemaps.org`.

3. Upload this to your Drupal Commerce website. You can either upload it in your root directory (for example, `http://www.example.com/sitemap.xml`) or in a sub directory (for example, `http://www.example.com/sitemaps/products.xml`), which may be useful if you have split your sitemap in to multiple files.
4. Finally, you can add a reference to your website's `robots.txt` file to tell search engines where to look for your XML sitemap file(s). To do this, open the `robots.txt` file in the root directory of your website and add the following to tell search engines about your new XML sitemap:

   Sitemap: `http://www.example.com/sitemap.xml`

   Or use this format if you have saved your sitemap in another directory, as discussed earlier:

   Sitemap: `http://www.example.com/sitemaps/products.xml`

> If you have Drupal installed in a sub directory of your domain, such as `http://www.example.com/drupal`, the `robots.txt` file still needs to be in the root directory, so that it appears at `http://www.example.com/robots.txt` as this is the correct location for it. If `robots.txt` is not placed directly in the root directory of your site, it will have no effect.

5. You can manually submit your XML sitemap to search engines, too:
   - Submit XML sitemaps to Google at `http://www.google.com/webmasters/sitemaps/`
   - Submit XML sitemaps to Bing at `http://www.bing.com/webmaster`
6. You may need to verify that you own your Drupal Commerce website before submitting sitemaps to search engines. If this is the case, you will need to follow the *Verifying your Drupal Commerce website with Bing and Google* recipe in this chapter.

## How it works...

If you're concerned that pages aren't being indexed by search engines, an XML sitemap may be a good way to introduce pages to search engines. You may also find signing up to services such as Google Webmaster Tools useful as this reports the number of pages the search engine indexes from your website and the number you are providing in your XML sitemap file (`https://www.google.com/webmasters/tools/`).

By creating an XML sitemap for your store through an online sitemap generator, you can include all of the product pages in your store, which Drupal modules are currently unable to do at the time of writing with the use of the XML Sitemap module (`http://drupal.org/project/xmlsitemap`).

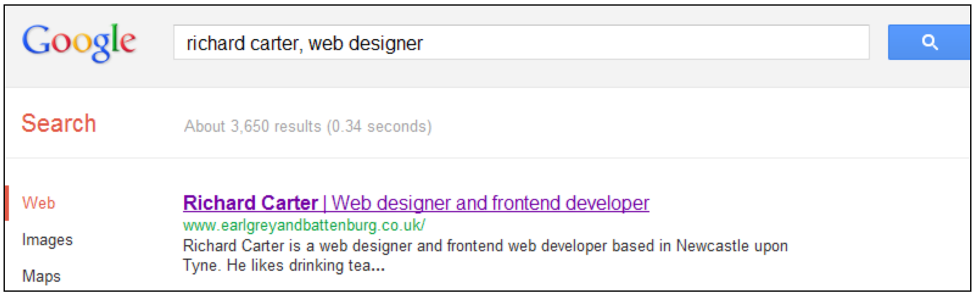
If you are using Drupal's multisite feature, you may find the **RobotsTxt** module useful: `http://drupal.org/project/robotstxt`

There is an experimental extension to XML Sitemap module for Commerce products: `http://drupal.org/sandbox/recrit/1279350`. Sandbox projects are not intended for inexperienced Drupal users or developers, so avoid it if you're not comfortable.

# Adding a meta description field to your Drupal Commerce product display

The days of the meta keywords field's usefulness have long since disappeared after years of abuse and keyword-stuffing. The meta description field can still be useful as it is displayed in the search engine results pages in search engines, though. Drupal Commerce doesn't give products description fields by default, so you will need to install a module to achieve what you are after.

The following example shows the first result for the search term **richard carter, web designer** in Google. Below the page's title and the URL, you can see that part of the page's meta description at the time of writing is used as the page's description:

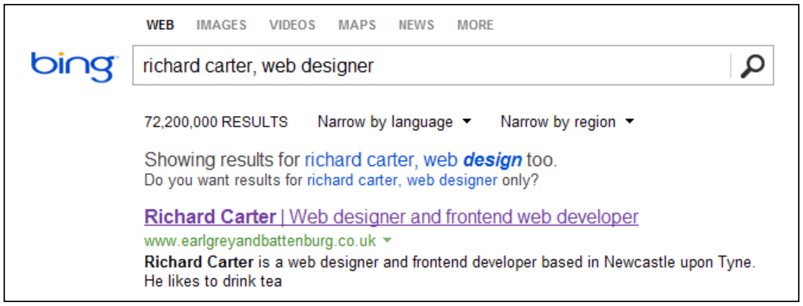


richard carter, web designer

About 3,650 results (0.34 seconds)

Richard Carter | Web designer and frontend developer
www.earlgreyandbattenburg.co.uk/
Richard Carter is a web designer and frontend web developer based in Newcastle upon Tyne. He likes drinking tea...

Similarly, in Bing, the meta description of the page is shown in the search engine results page too:



richard carter, web designer

72,200,000 RESULTS

Showing results for richard carter, web ***design*** too.
Do you want results for richard carter, web designer only?

Richard Carter | Web designer and frontend web developer
www.earlgreyandbattenburg.co.uk
**Richard Carter** is a web designer and frontend developer based in Newcastle upon Tyne. He likes to drink tea

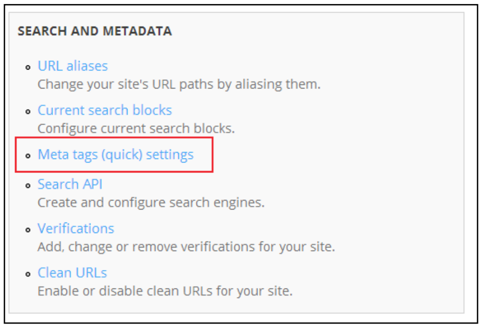
## Getting ready

Download the Meta Tags Quick module from Drupal.org at `http://drupal.org/project/metatags_quick`, and install this to your Drupal Commerce installation.

## How to do it...

Once you have installed the Meta Tags Quick module, follow these steps:

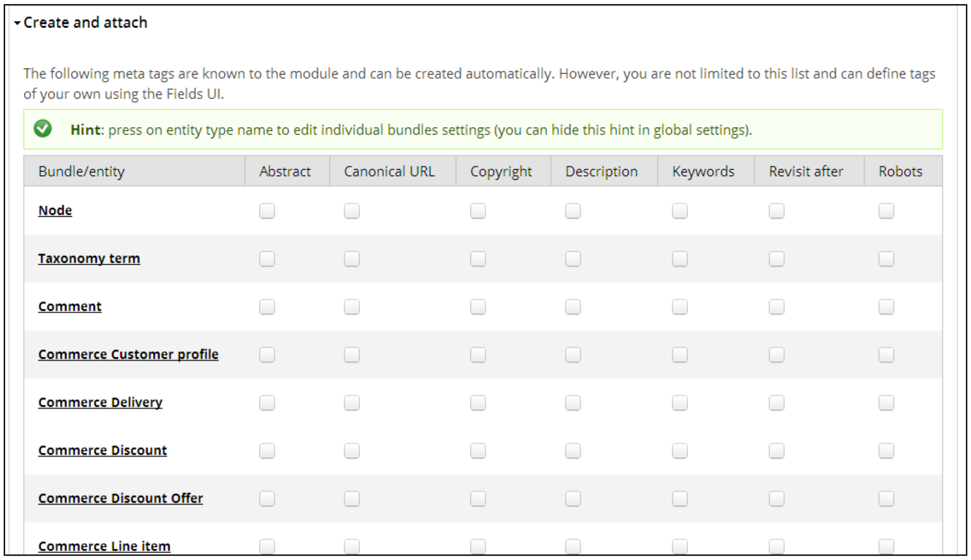
1. You can now navigate to the **Site settings** | **Configuration** | **Search and Metadata** block and select the **Meta tags (quick) settings** option:



SEARCH AND METADATA

- URL aliases
  Change your site's URL paths by aliasing them.
- Current search blocks
  Configure current search blocks.
- Meta tags (quick) settings
- Search API
  Create and configure search engines.
- Verifications
  Add, change or remove verifications for your site.
- Clean URLs
  Enable or disable clean URLs for your site.

2. Under the **General** tab towards the top-right of your screen, you will see a list of the Bundles and Entities in your Drupal Commerce installation that the **Description** (and other) field can be applied to:



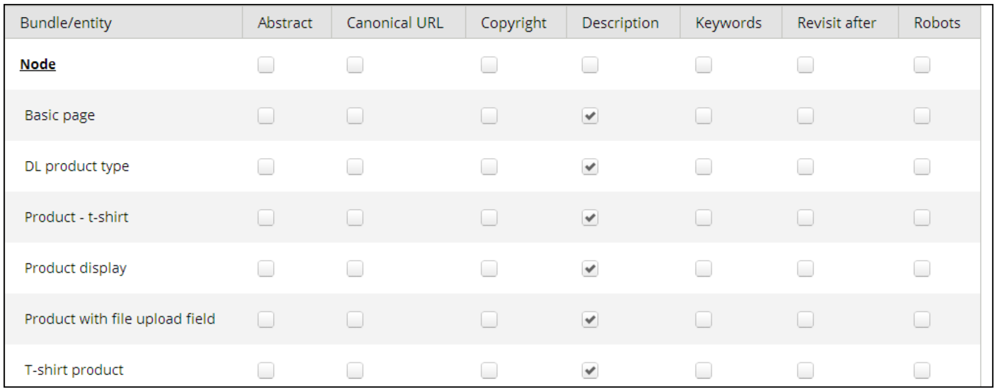
▾ Create and attach

The following meta tags are known to the module and can be created automatically. However, you are not limited to this list and can define tags of your own using the Fields UI.

✓ **Hint**: press on entity type name to edit individual bundles settings (you can hide this hint in global settings).

| Bundle/entity | Abstract | Canonical URL | Copyright | Description | Keywords | Revisit after | Robots |
|---|---|---|---|---|---|---|---|
| Node | □ | □ | □ | □ | □ | □ | □ |
| Taxonomy term | □ | □ | □ | □ | □ | □ | □ |
| Comment | □ | □ | □ | □ | □ | □ | □ |
| Commerce Customer profile | □ | □ | □ | □ | □ | □ | □ |
| Commerce Delivery | □ | □ | □ | □ | □ | □ | □ |
| Commerce Discount | □ | □ | □ | □ | □ | □ | □ |
| Commerce Discount Offer | □ | □ | □ | □ | □ | □ | □ |
| Commerce Line item | □ | □ | □ | □ | □ | □ | □ |

3. Expand the **Node** bundle, and select the checkbox in the **Description** column for each type of product type that you have created in your Drupal Commerce store:

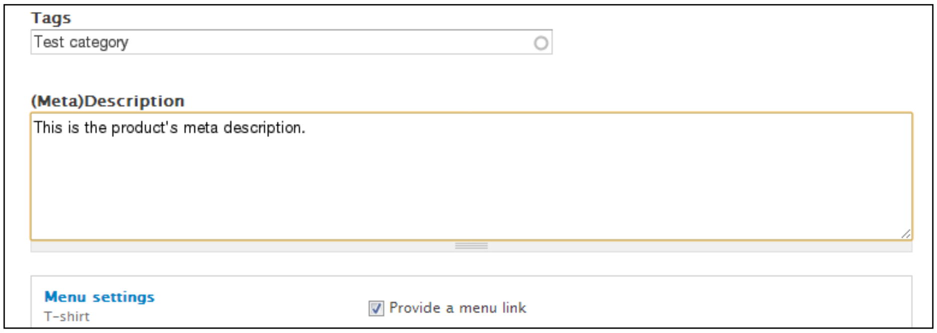


| Bundle/entity | Abstract | Canonical URL | Copyright | Description | Keywords | Revisit after | Robots |
|---|---|---|---|---|---|---|---|
| **Node** | □ | □ | □ | □ | □ | □ | □ |
| Basic page | □ | □ | □ | ☑ | □ | □ | □ |
| DL product type | □ | □ | □ | ☑ | □ | □ | □ |
| Product - t-shirt | □ | □ | □ | ☑ | □ | □ | □ |
| Product display | □ | □ | □ | ☑ | □ | □ | □ |
| Product with file upload field | □ | □ | □ | ☑ | □ | □ | □ |
| T-shirt product | □ | □ | □ | ☑ | □ | □ | □ |

4. Click on the **Attach** button to add the field to that particular entity type, and then click on the **Submit** button to save changes.

5. The meta description field will now appear on product display edit screens. Navigate to **Products** | **Manage products** in the administration panel and select the edit option on a product to which you want to add a meta description.

6. Scrolling down this page, you will now see the **(Meta)Description** field, to which you can add your desired meta description value for this product:



**Tags**

Test category

**(Meta)Description**

This is the product's meta description.

**Menu settings**
T-shirt

☑ Provide a menu link

7. Click on the **Save** button to save your newly added meta description content.

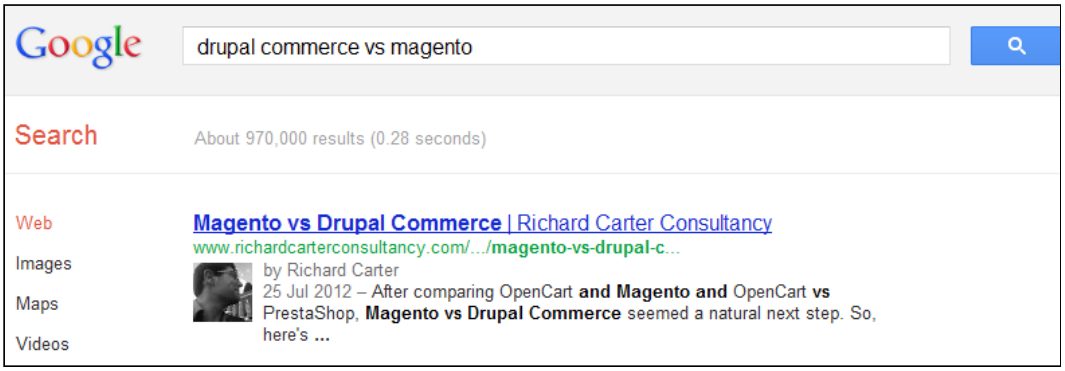
8. If you now navigate to the product page on your store and view the page's source (*Ctrl* + *U* in most modern browsers), you will see that the meta description field is now added in the `<head>` element of your page:

```
<meta name="description" content="This is the product&#039;s meta
description." />
```

9. That's it! You can now customize your product page's meta description fields.

## Integrating Google Authorship with your Drupal Commerce store

The following is an example of how Google Authorship changes your website's search engine results page listings:



drupal commerce vs magento

About 970,000 results (0.28 seconds)

Magento vs Drupal Commerce | Richard Carter Consultancy
www.richardcarterconsultancy.com/.../magento-vs-drupal-c...
by Richard Carter
25 Jul 2012 – After comparing OpenCart and Magento and OpenCart vs PrestaShop, Magento vs Drupal Commerce seemed a natural next step. So, here's ...

Note that a link to the relevant Google Plus profile is added below the page's URL, and a photograph linked to the Google Plus account is displayed next to the listing too. This recipe will show you how to enable Google Authorship on your Drupal Commerce website.

> Google Authorship enables you to enhance your search engine listings by providing Google with more information about the author of your website. For more information on Google Authorship, see `https://plus.google.com/authorship`.

### Getting started

You will need to create a Google Plus account, and register for authorship. Follow the steps Google provided online to verify your account (`https://plus.google.com/authorship`).

> For more reading on Google Authorship, see this blog post: `http://www.seomoz.org/blog/read-but-learn-by-doing-an-approach-to-the-analysis-of-seo-issues-between-authorship-google-plus-and-snippets`

## How to do it...

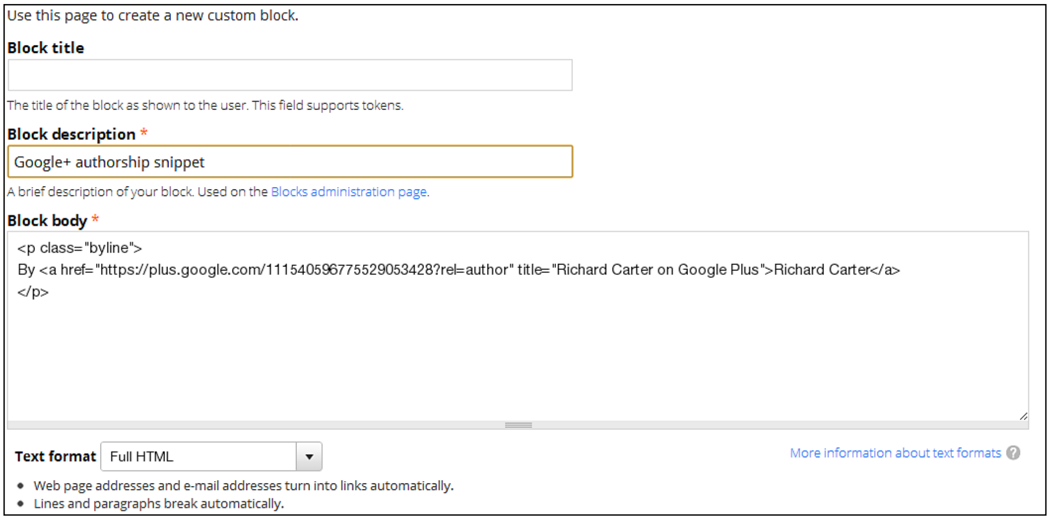
Once you have verified your Google Plus account, you can start integrating it with your Drupal Commerce website:

1. Navigate to **Site settings** | **Visual & Layout** | **Blocks** in your administration panel.
2. Click in the **Add block** link at the top of the screen, and edit the block with the **Text format** drop-down list set to **Full HTML**.
3. Give your block a sensible value for **Block description**, and use the following markup in the **Block body** field:

   ```
   <p class="byline">
   By <a href="https://plus.google.com/111540596775529053428?rel=auth
   or" title="Richard Carter on Google Plus">Richard Carter</a>
   </p>
   ```

   > Don't forget to replace `https://plus.google.com/111540596775529053428` with the URL of your own Google Plus account!

4. You can modify this as you see fit, but ensure that you link it to the Google Plus account that you set up, and that also you include the `?rel=author` attribute at the end of the URL to tell Google that this page is related to the author of its content.
5. Finally, select a region in your Drupal theme in the **Region settings** pane to assign the new block once you save it. The example sets the block to the **Footer First** region in the Omega Kickstart theme:



Use this page to create a new custom block.

**Block title**

The title of the block as shown to the user. This field supports tokens.

**Block description** *

Google+ authorship snippet

A brief description of your block. Used on the Blocks administration page.

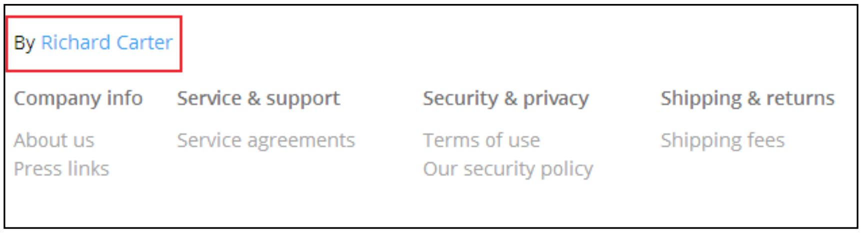
**Block body** *

```
<p class="byline">
By <a href="https://plus.google.com/111540596775529053428?rel=author" title="Richard Carter on Google Plus">Richard Carter</a>
</p>
```

**Text format** Full HTML

- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

6. Click on the **Save** block button to see if it appears on the frontend of your Drupal Commerce website that customers see (you may need to clear your cache if you don't see it immediately):



By Richard Carter

7. That's it! You've integrated Google Authorship on your store. It may take several days, weeks, or even months to appear, so be patient (and there's a chance Google will decide not to use it at all, of course).

## How it works...

By verifying who you are with a Google Plus and linking to your Google Plus profile from your Drupal Commerce store using the `rel=author` attribute, you're telling Google that, if it decides so, you would like a little more information about the author of the page—you—to be displayed in search engine results pages. You do this by associating a Google Plus profile with your Drupal Commerce store, which allows Google to enhance your website's listings in search engine result pages, which may help your store stand out from others!

Taking these steps will not guarantee that your avatar and Google Plus account are linked in all or any of your website's pages, but if Google decides it's helpful to those searching, it will include them. For more information, see `http://support.google.com/webmasters/bin/answer.py?hl=en&answer=1306778`.

# 7
# Theming Drupal Commerce

This chapter covers techniques to customize the look and feel of your Drupal Commerce website. We will be covering the following topics:

- Creating a new theme for Drupal Commerce
- Customizing Drupal Commerce product pages
- Customizing the Drupal Commerce checkout
- Theming Drupal Commerce's cart page
- Using Google Fonts with your Drupal Commerce theme

## Introduction

Your store may be among hundreds or thousands of competitors selling similar products, the design, and visual appeal of your website can make a huge difference to sales.

From the basics such as uploading your own logo and favorites icon to customizing the look and feel of your entire store, there's much you can do to customize your Drupal Commerce store.

## Creating a new theme for Drupal Commerce

Drupal Commerce is supported by Drupal's flexibility, and its theming system is no different. You can create a new theme to change the look and feel of your Drupal Commerce store.

## Getting started

Drupal Commerce's default theme is an extension of the Omega theme of Drupal 7 called Omega Kickstart (see `http://drupal.org/project/omega_kickstart`):

As you can see, it provides a nice, clean base to start your own Drupal Commerce theme on.

## How to do it...

You can create your own theme based on the Omega Kickstart theme by creating a subtheme, which will simply overwrite the aspects of the Omega Kickstart theme you don't want to use:

1. To begin creating your new subtheme, you will need to create a directory in the `/sites/all/themes` directory of your Drupal Commerce installation. The new subtheme will be called `defacto`, so you will need to create a directory at `/sites/all/themes/defacto`.
2. Your new theme will need a `.info` file to tell Drupal about it, so create a file in your new directory called `defacto.info` and add some information about your theme, including its name and a brief description of the theme:

   ```
   name = 'Defacto'
   description = 'Custom Drupal Commerce sub theme based on Omega
   Kickstart.'
   screenshot = 'screenshot.png'
   ```

3. Next you need to tell Drupal which theme you want to use as a base theme, which you do by adding the following line to the `.info` file:

   ```
   base theme = 'omega_kickstart'
   ```

4. To ensure that Drupal accepts the theme as compatible with its core version, add the following line as this isn't inherited from the parent theme:

   ```
   core = '7.x'
   ```

5. Drupal subthemes inherit many things from their base theme, but they will not inherit regions from them. To preserve the regions, it's easiest to copy the regions from the base theme's `.info` file, which is located in the `profiles/commerce_kickstart/themes/omega_kickstart` directory if you installed Drupal Commerce using Commerce Kickstart:

   ```
   regions[page_top] = 'Page Top'
   regions[page_bottom] = 'Page Bottom'
   regions[content] = 'Content'
   ```

```
regions[user_first] = User Bar First
regions[user_second] = User Bar Second

regions[branding] = Branding
regions[menu] = Menu
regions[breadcrumb] = Breadcrumb

regions[sidebar_first] = Sidebar First
regions[sidebar_second] = Sidebar Second

regions[header_first] = Header First
regions[header_second] = Header Second

regions[preface_first] = Preface First
regions[preface_second] = Preface Second
regions[preface_third] = Preface Third

regions[postscript_first] = Postscript First
regions[postscript_second] = Postscript Second
regions[postscript_third] = Postscript Third

regions[footer_first] = Footer First
regions[footer_second] = Footer Second
regions[footer_third] = Footer Third

regions[footer2_first] = Footer2 First
regions[footer2_second] = Footer2 Second
```

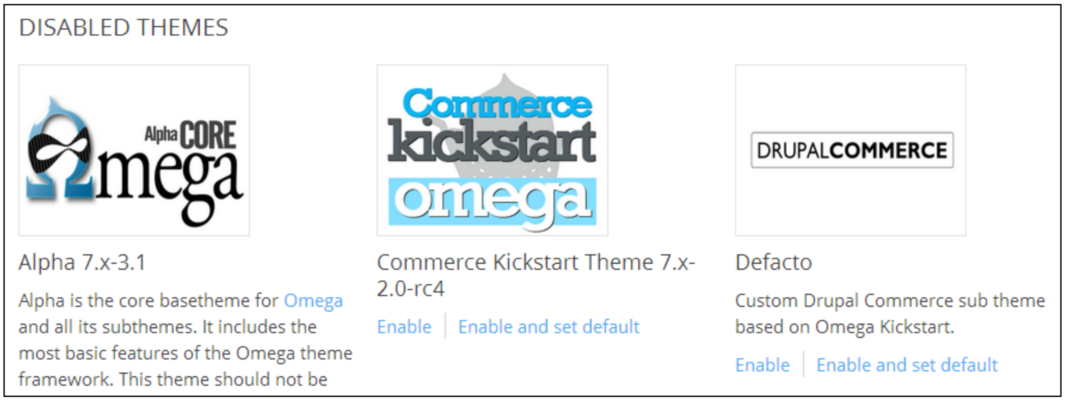
6. Although the base theme's screenshot is inherited by subthemes, you can also specify a screenshot of the theme by adding a line to the `defacto.info` file. Save a file called `screenshot.png` (you can update this later once your new theme is complete) in the subtheme's directory and add the following to your `.info` file:

   ```
   screenshot = 'screenshot.png'
   ```

7. To inherit the base theme's stylesheets, you need to include at least one stylesheet in your subtheme's `.info` file, so create a file called `defacto.css` in your subtheme's directory, and add the following line to your `defacto.info` file:

   ```
   stylesheets[all][]  = defacto.css
   ```

8. You can now enable the subtheme. Navigate to **Appearances** in your administration panel, and scroll down to see the **DISABLED THEMES** list. Under this heading, you will see your new theme. Click on the **Enable** and set the default link:



DISABLED THEMES

Alpha 7.x-3.1

Alpha is the core basetheme for Omega and all its subthemes. It includes the most basic features of the Omega theme framework. This theme should not be

Commerce Kickstart Theme 7.x-2.0-rc4

Enable | Enable and set default

Defacto

Custom Drupal Commerce sub theme based on Omega Kickstart.

Enable | Enable and set default

9. Your new theme is now enabled! You won't see any changes quite yet, though, as your subtheme isn't telling Drupal to change anything from the base theme.

> **Downloading the example code**
>
> You can download the example code files for all Packt books you have purchased from your account at `http://www.packtpub.com`. If you purchased this book elsewhere, you can visit `http://www.packtpub.com/support` and register to have the files e-mailed directly to you.

# Customizing Drupal Commerce product pages

The product page is one of the most important aspects of your store. A good product page design can showcase your product clearly and make it easy for customers to start purchasing products; a bad product page can dissuade customers from using your store and using a competitor.

Drupal Commerce gives you the ability to customize your store's product pages using Drupal's template override system and a sprinkling of CSS, of course.

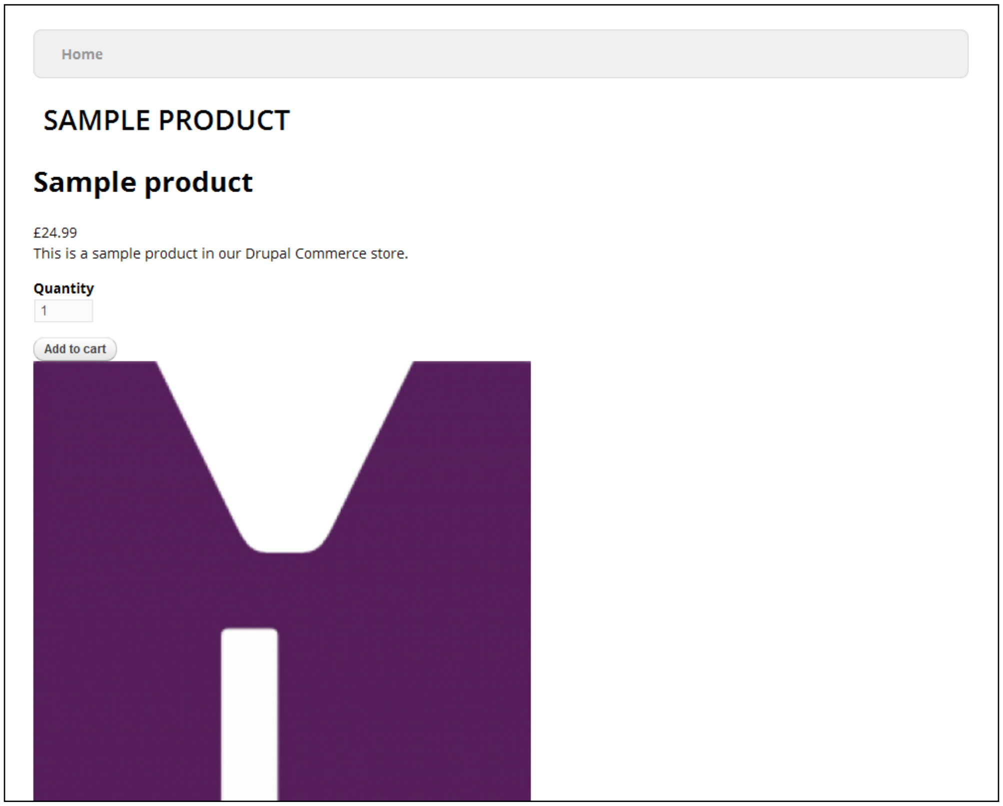
## Getting started

You will need to create a new Drupal theme to use on your store. See the *Creating a new theme for Drupal Commerce* recipe.

## How to do it...

Customizing your product page can entail a number of techniques. In this example, you will see how to apply your own CSS to the products page and alter the product page's HTML:

1. By default, here's what your Drupal Commerce product page probably looks like, assuming you are using the Kickstart Omega theme that comes installed and activated by default:



Home

SAMPLE PRODUCT

Sample product

£24.99

This is a sample product in our Drupal Commerce store.

Quantity

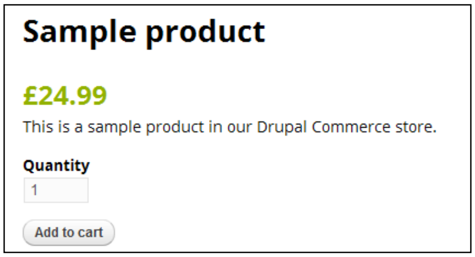
1

2. As you can see, it's not the most attractive of product pages to entice a customer to buy from your store. Open your theme's CSS file (for example, `/sites/all/themes/defacto/defacto.css`) and apply the following CSS to style the product's price display:

```
.field-name-commerce-price {
  color: #88aa00;
  font-size: 24px;
  font-weight: bold;
}
```

> You may need to clear Drupal's cache by navigating to **Site settings** | **Configuration** | **Performance** and clicking on the **Clear all caches** button.

3. Upload your theme's CSS file and refresh your store's product page to see the updated style take hold:



Sample product

£24.99

This is a sample product in our Drupal Commerce store.

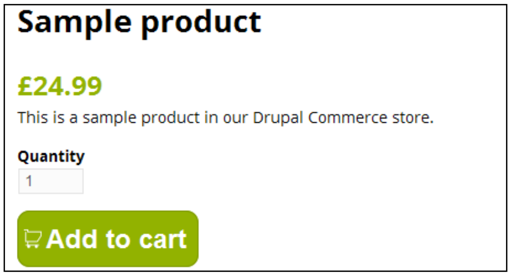
Quantity

1

4. Next, you can apply more CSS to the **Add to cart** button to make it more appealing to customers by changing its color, making it larger, and adding a shopping cart icon:

```
.commerce-add-to-cart .form-submit {
  background: #88aa00 url("images/add-to-cart_bg.png") no-repeat
5px center;
  border-color: #739000;
  color: #fff;
  font-size: 24px;
  font-weight: bold;
  padding: 10px 10px 10px 25px;
}
```

5. After uploading your theme's CSS file and the `add-to-cart_bg.png` image in a subdirectory of your theme called `/images` to your store, you will see the add to cart button change appearance:



Sample product

£24.99

This is a sample product in our Drupal Commerce store.

Quantity

1

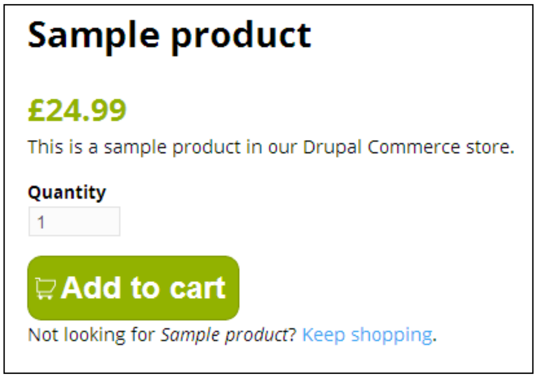
6. Next, you will customize the product page to include a prompt to redirect customers, who aren't looking for the particular product, back to your store's home page. To do this, you will need to overwrite Drupal templates. Create a file called `node--product_display.tpl.php` (note the double-dash between `node` and `product`) in a subdirectory of your theme called `templates`, so that it appears in `/sites/all/themes/defacto/templates`.
7. Now copy the contents of `node.tpl.php` in `/profiles/commerce_kickstart/themes/omega_kickstart/templates` to the newly created `node--product_display.tpl.php` file.
8. Looking through the `node--product_display.tpl.php` file, you should see the following block of code:

```
<div<?php print $content_attributes; ?>>
    <?php
      // We hide the comments and links now so that we can render
them later.
      hide($content['comments']);
      hide($content['links']);
      print render($content);
    ?>
  </div>
```

9. This outputs the block of content associated with the product. To add our link, add the following code towards the bottom of this snippet, so that it appears as follows:

```
<div<?php print $content_attributes; ?>>
    <?php
      // We hide the comments and links now so that we can render
them later.
      hide($content['comments']);
      hide($content['links']);
      print render($content);
    ?>
    <p class="keep-shopping">
      Not looking for <em><?php print $title ?></em>? <a
href="<?php print $GLOBALS['base_url'] . $GLOBALS['base_path'];?>"
title="Keep shopping">Keep shopping</a>.
    </p>
  </div>
```

10. If you now upload your `node--product-display.tpl.php` file, you will see the following content appear in your product page:



## Sample product

£24.99

This is a sample product in our Drupal Commerce store.

Quantity

1

Not looking for *Sample product*? Keep shopping.

11. You may need to clear Drupal's cache to see changes to templates. Navigate to **Configuration** | **Performance** and click on the **Clear all caches** button.

# Customizing the Drupal Commerce checkout

Alongside products and categories, the checkout is a fundamental part of any online store, and it's important to get it right. As with other views in Drupal Commerce, you can customize your checkout through Drupal themes.

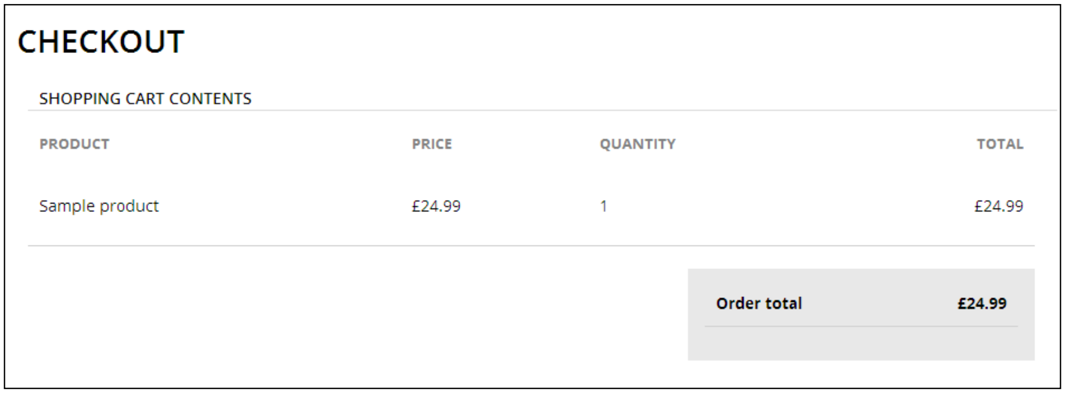
## Getting started

You will need to create your own Drupal theme before performing the following steps; see the *Creating a new theme for Drupal Commerce* recipe.

## How to do it...

To style the checkout page, you can edit both CSS and the HTML file that Drupal uses to style and structure these elements:

1. Navigate to your store's checkout by adding a product to your cart, and clicking on the **Checkout** button on the **Shopping Cart** page. As you can see, the checkout page is currently functional, but a little bare:



CHECKOUT

SHOPPING CART CONTENTS

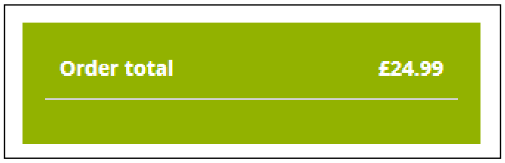
| PRODUCT | PRICE | QUANTITY | TOTAL |
|---|---|---|---|
| Sample product | £24.99 | 1 | £24.99 |

| Order total | £24.99 |
|---|---|

2. Open your theme's stylesheet (for example, `/sites/all/themes/defacto/defacto.css`), and apply the following style to alter the style of the **Order total** block:

```
.field-name-commerce-order-total .field-items {
  background: #88aa00 !important;
  color:#fff;
}
```

3. If you now refresh your checkout page once you have uploaded the new CSS, you will see the **Order total** block has changed to be more visible. (You may need to clear your cache at **Configuration** | **Performance** first.)



| Order total | £24.99 |
|---|---|

4. You can also overwrite Drupal's default template for the checkout page to make deeper customizations to your store. Create a file called `page--checkout.tpl.php` in the `/sites/all/themes/defacto/templates` directory of your Drupal Commerce installation.

> Note the double hyphen character between `page` and `cart`.

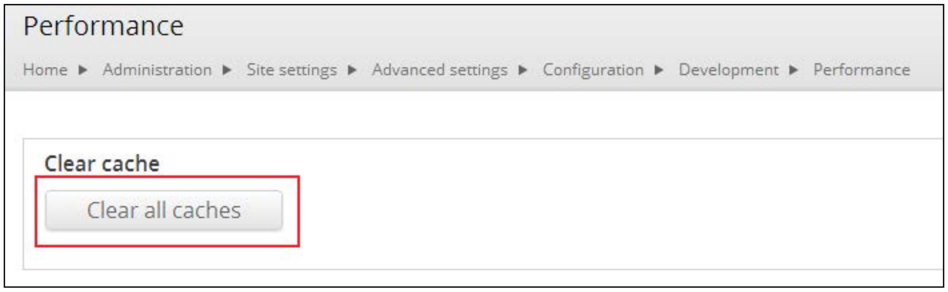
5. Copy the contents of the `page.tpl.php` file in `/profiles/ commerce_kickstart/themes/omega/alpha/templates` to the contents of the file you created previously; this ensures that Drupal can still add the Commerce cart content into the page.

> You'll need to look in the `/themes/omega/templates for page.tpl.php` file if you didn't install Drupal Commerce with Commerce Kickstart.

6. The contents of your theme's `page--checkout.tpl.php` file should look similar to the following:

```
<div<?php print $attributes; ?>>
  <?php if (isset($page['header'])) : ?>
    <?php print render($page['header']); ?>
  <?php endif; ?>

  <?php if (isset($page['content'])) : ?>
    <?php print render($page['content']); ?>
  <?php endif; ?>

  <?php if (isset($page['footer'])) : ?>
    <?php print render($page['footer']); ?>
  <?php endif; ?>
</div>
```

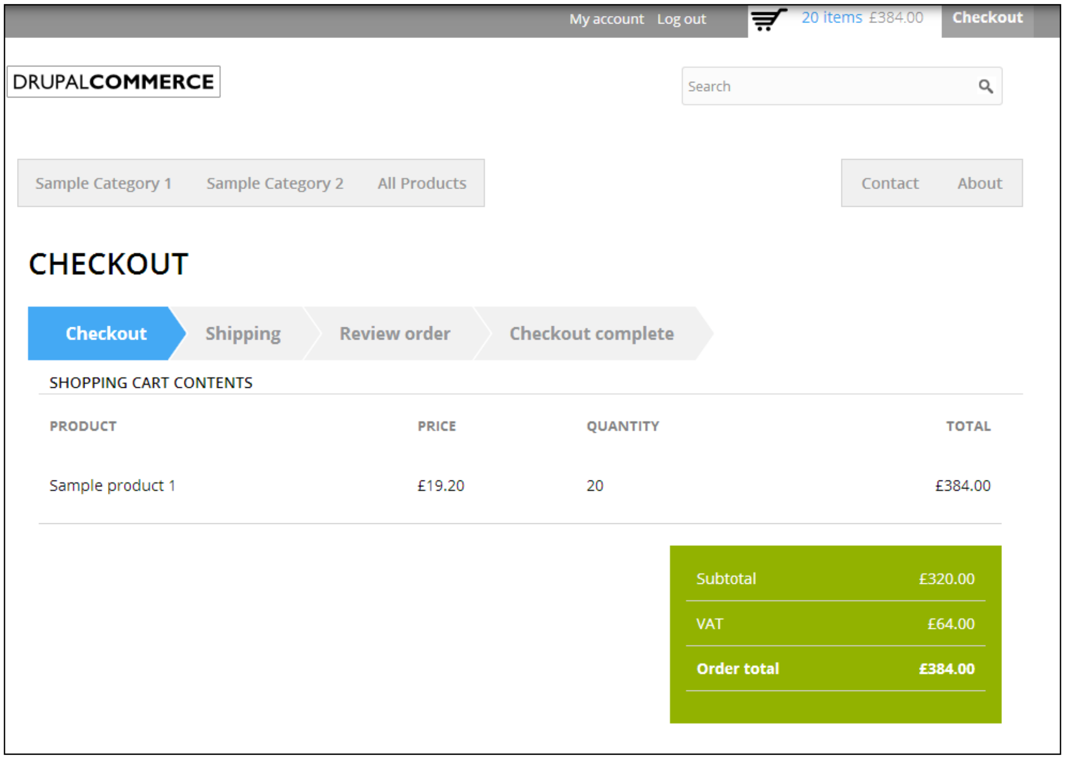
7. You'll need to clear Drupal's cache to see the changes; navigate to **Site settings** | **Configuration** and click on the **Clear all caches** button:


Performance

Home ▶ Administration ▶ Site settings ▶ Advanced settings ▶ Configuration ▶ Development ▶ Performance

Clear cache

8. As you can see, this renders the checkout page as follows:



CHECKOUT

SHOPPING CART CONTENTS

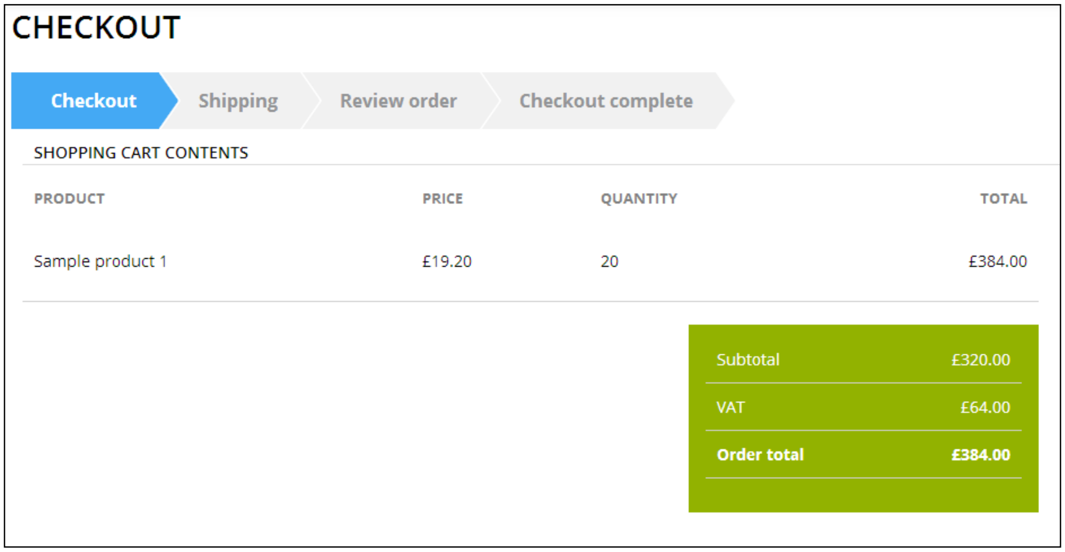
| PRODUCT | PRICE | QUANTITY | TOTAL |
|---|---|---|---|
| Sample product 1 | £19.20 | 20 | £384.00 |

| | |
|---|---|
| Subtotal | £320.00 |
| VAT | £64.00 |
| **Order total** | **£384.00** |

9. You can now overwrite and add regions as you see fit, that will affect only the checkout page in your Drupal Commerce store. For example, you may want to simplify the header area of your store on the checkout page to help increase conversions; you could do this by altering your theme's `page--checkout.tpl.php` file as follows:

```
<div<?php print $attributes; ?>>

  <?php if (isset($page['content'])) : ?>
    <?php print render($page['content']); ?>
  <?php endif; ?>

  <?php if (isset($page['footer'])) : ?>
    <?php print render($page['footer']); ?>
  <?php endif; ?>
</div>
```

10. This alters your store's layout to look as follows, making the checkout more prominent:



SHOPPING CART CONTENTS

| PRODUCT | PRICE | QUANTITY | TOTAL |
|---|---|---|---|
| Sample product 1 | £19.20 | 20 | £384.00 |

| | |
|---|---|
| Subtotal | £320.00 |
| VAT | £64.00 |
| **Order total** | **£384.00** |

## How it works...

Drupal has a strict hierarchy of how files are loaded to form the page. If a file Drupal needs isn't available in your theme, it will fall back to the system's default file. So, if Drupal finds a `page--checkout.tpl.php` file in your theme, it will use it, but if the file doesn't exist, it falls back to the `page.tpl.php` template in your theme, and then the `page-checkout.tpl.php` file in Drupal Commerce's core files, and so on.

> You may find the Theme Developer module useful if you plan on customizing your Drupal Commerce theme a lot, as it displays useful information about where Drupal is loading elements of the page from. You can download the module from `http://drupal.org/project/devel_themer`.

# Theming Drupal Commerce's cart page

Distinct from the checkout page, Drupal Commerce's cart page shows your customers a review of the order they're about to make.

This recipe will walk you through how to overwrite Drupal Commerce's cart page template, to give you the knowledge you need to start customizing your store's theme more deeply.

## Getting started

Using the subtheme Defacto that you created in a previous recipe, you will add this theme to your store to proceed with this recipe. Ensure you have the Defacto theme enabled and set as the default theme on your website, too. Go to **Site settings** | **Appearance** to change this.

## How to do it...

Follow these steps to overwrite the default Drupal Commerce cart page with your own custom template:

1. Create a file in the `/sites/all/themes/defacto/templates` directory of your Drupal Commerce installation called `page--cart.tpl.php`.

   > Note the double hyphen character between `page` and `cart`.

2. Copy the contents of the `page.tpl.php` file in `/profiles/ commerce_kickstart/themes/omega/alpha/templates` to the contents of the file you created previously; this ensures that Drupal can still add the Commerce cart content into the page.

   > You'll need to look in `/themes/omega/templates` for this file if you didn't install Drupal Commerce with Commerce Kickstart.

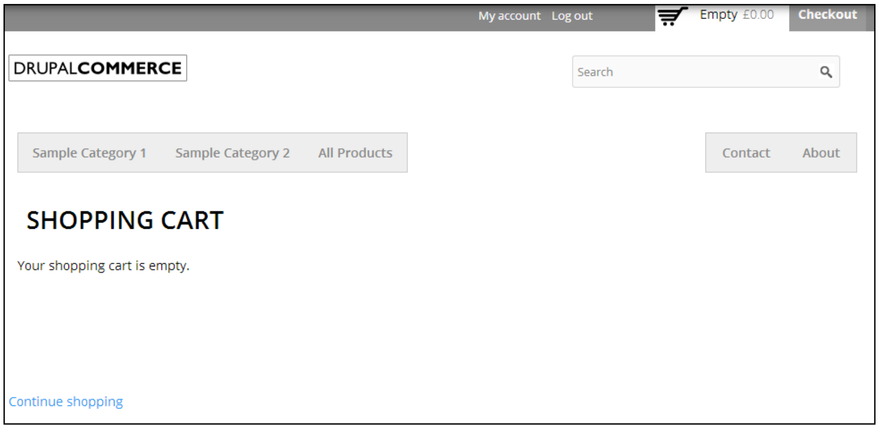
3. You can now edit the `page--cart.tpl.php` file in your theme's directory (for example, `/sites/all/themes/defacto/templates`) to add a `Continue shopping` link at the bottom of the shopping cart to help users navigate back to your store when looking at their shopping cart's contents:

```
<div<?php print $attributes; ?>>
  <?php if (isset($page['header'])) : ?>
    <?php print render($page['header']); ?>
  <?php endif; ?>

  <?php if (isset($page['content'])) : ?>
    <?php print render($page['content']); ?>
    <p class="container-24">
      <a href="javascript:history,back(-1)">Continue shopping</a>
    </p>
  <?php endif; ?>

  <?php if (isset($page['footer'])) : ?>
    <?php print render($page['footer']); ?>
  <?php endif; ?>
</div>
```

   The `container-24` class on the paragraph uses Omega's CSS to tell the paragraph to be formatted to the full width of the content block. The JavaScript within the `href` attribute returns the customer to the previous page they visited. You could replace this with the address of a page within your website.

4. Clear Drupal's cache by navigating to **Site settings** | **Configuration** | **Performance** and clicking on the **Clear all caches** button.

5. Now navigate to the cart on your Drupal Commerce store, and you'll see the **Continue shopping** link appear:



SHOPPING CART

Your shopping cart is empty.

Continue shopping

# Using Google Fonts with your Drupal Commerce theme

The Web used to be very restricted in its typography, but with the advent of **@font-face**, web designers have more choice than ever before. Now that you're not stuck with the likes of Arial and Helvetica, you can truly customize your store's appearance to match your branding.

This recipe will take you through using a web font from Google Font in to your Drupal Commerce theme.

## Getting started

Visit Google Fonts at `http://www.google.com/fonts` and select the font you wish to use. You will need the URL of the particular font variations you will be using, which will look something like the following:

```
http://fonts.googleapis.com/css?family=PT+Sans
```

## How to do it...

Once you have the font in mind you wish to use in your Drupal Commerce, follow these steps:

1. Open your theme's CSS file; if you have used the example theme we used earlier in this chapter, this will be the `defacto.css` file in the `/sites/all/themes` directory of your Drupal installation.
2. At the top of the CSS file, add the following line to import the custom font's CSS file into your own stylesheet:

   ```
   @import url("http://fonts.googleapis.com/css?family=PT+Sans");
   /* Your Drupal theme's styles below this */
   ```

   > Note you need to use the font's full URL here for your font to work correctly.

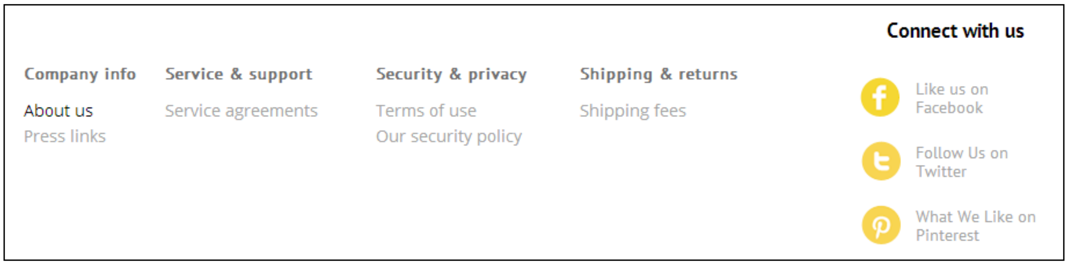
3. The font is now imported to your stylesheet, but you need to refer that you want to use this font somewhere in your stylesheet for it to appear on your Drupal Commerce store's design. In your theme's stylesheet, add the following line:

   ```
   body {
     font-family: "PT Sans", Arial, Helvetica, sans-serif;
   }
   ```

4. This applies the font to all text in any element in your page. You can just apply this font to headings or by adapting the CSS accordingly. For example, to assign the new font only to headings in pages, use the following CSS:

   ```
   h1,h2,h3,h4,h5,h6 {
     font-family: "PT Sans", Arial, Helvetica, sans-serif;
   }
   ```

5. Upload your CSS file back to your Drupal installation.
6. Clear Drupal's cache by navigating to **Site settings** | **Configuration** | **Performance** and clicking on the **Clear all caches** button.
7. If you now refresh the frontend of your Drupal Commerce store, you'll see your new font appear:



Connect with us

Company info
About us
Press links

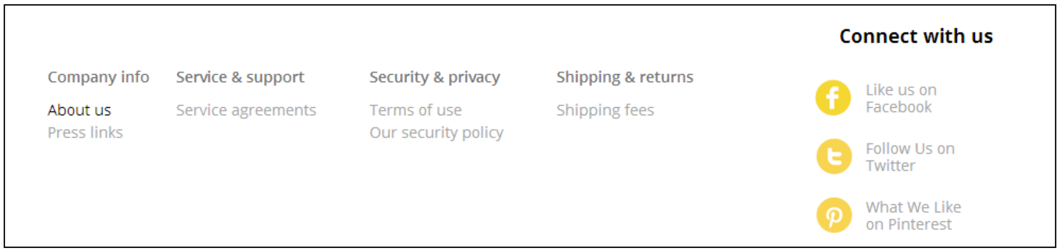
Service & support
Service agreements

Security & privacy
Terms of use
Our security policy

Shipping & returns
Shipping fees

Like us on Facebook
Follow Us on Twitter
What We Like on Pinterest

For reference, here's what the store's footer looked like before the font alteration:



Connect with us

Company info
About us
Press links

Service & support
Service agreements

Security & privacy
Terms of use
Our security policy

Shipping & returns
Shipping fees

Like us on Facebook
Follow Us on Twitter
What We Like on Pinterest

If you look carefully, you'll see that the newer design's font has slimmer letter forms. Of course, Google Fonts has a huge variety of styles available, so you can customize your store's fonts how you please.

## How it works...

By using the Google Fonts service, you can specify a larger range of fonts to use on your store and customize its look and feel, without the need for your customer to have the font you're using installed on their computer. Google Fonts voids the need for the customer to have the font installed as it provides a copy of the font from their servers, loaded as part of your website's stylesheets.

Usually, you can include CSS files for your Drupal theme in the `.info` file, but this method will not accept the absolute URLs (that is, `http://fonts.googleapis.com/css?family=PT+Sans`) used by Google Fonts as valid, leaving the CSS `@import` rule as the only way to do this.

# 8

# Further Configuration

This chapter covers some advanced configuration tips for your Drupal Commerce store, including the following:

- Enabling product reviews on your Drupal Commerce store
- Adding a contact form to your Drupal Commerce store
- Filtering UK postcodes or US ZIP codes from your Drupal Commerce store
- Enhancing customer experience at checkout in Drupal Commerce with AJAX

## Introduction

Your Drupal Commerce store is now up and running and looks great, but there's always something you can do to improve your customers' experiences. From making it easier to contact you to allowing your customers to leave reviews of your products on your website, and making it easier for customers to find their address using ZIP or postcode lookup tools.

## Enabling product reviews on your Drupal Commerce store

One way to help influence customers to purchase products is to allow them to publish reviews of your products on your store.

This recipe guides you through configuring the Fivestar module and Drupal Commerce to allow product ratings on your store.

## Getting started

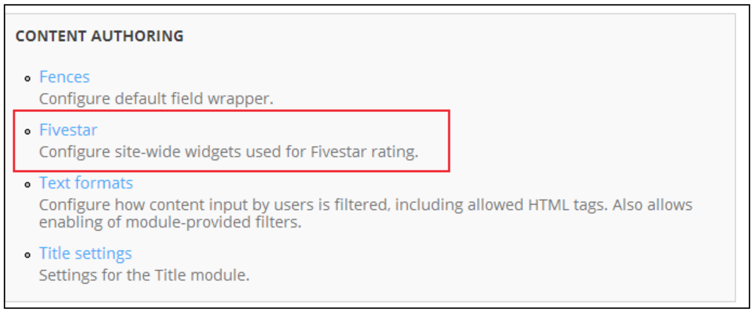
Go to Drupal.org and install the Fivestar module from `http://drupal.org/project/fivestar`, as well as the Voting API module from `http://drupal.org/project/votingapi`. Log in to your Drupal Commerce store's administration.

If you're looking for a simpler *thumbs up/thumbs down* approach to rating for your store, see the Rate module: `http://drupal.org/project/rate`.

## How to do it...

Once you have installed the Fivestar module for Drupal, follow these steps:

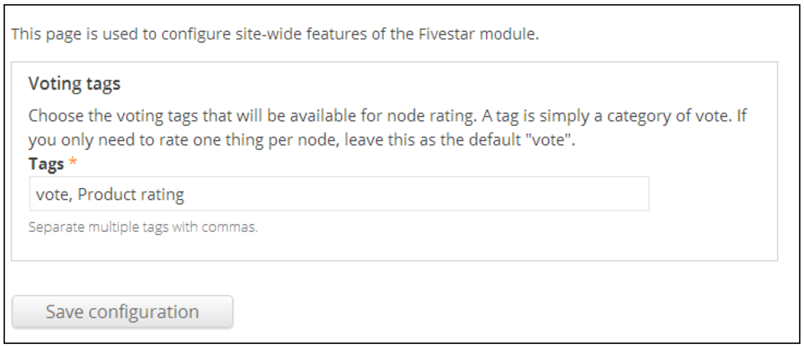
1. Navigate to **Site settings** | **Configuration** | **Fivestar**:



CONTENT AUTHORING

- Fences
  Configure default field wrapper.
- Fivestar
  Configure site-wide widgets used for Fivestar rating.
- Text formats
  Configure how content input by users is filtered, including allowed HTML tags. Also allows enabling of module-provided filters.
- Title settings
  Settings for the Title module.

2. Add a new voting tag called `Product rating` to the **Tags** field displayed as follows:



This page is used to configure site-wide features of the Fivestar module.

Voting tags

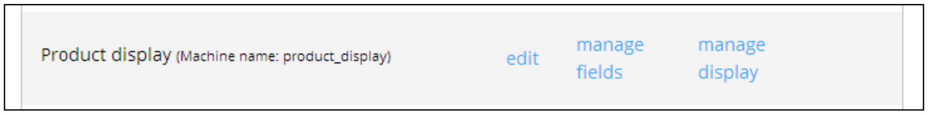
Choose the voting tags that will be available for node rating. A tag is simply a category of vote. If you only need to rate one thing per node, leave this as the default "vote".

Tags *

vote, Product rating

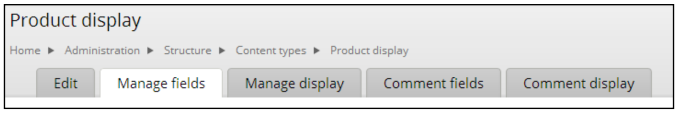
Separate multiple tags with commas.

3. Click on the **Save configuration** button and navigate to **Site settings** | **Structure** | **Content types**.
4. Find your **Product display** content type here, and click on the **edit** link in the **Operations** column:



Product display (Machine name: product_display)

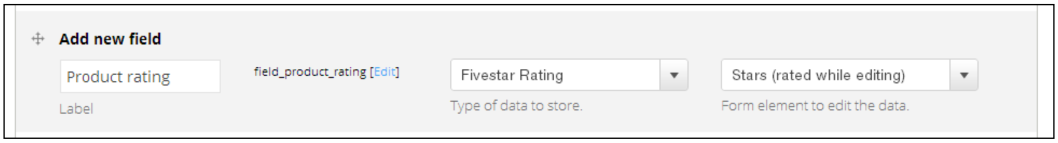
5. Click on the **Manage fields** tab at the top of the screen:



Product display

Home ▶ Administration ▶ Structure ▶ Content types ▶ Product display
Edit
Manage fields
Manage display
Comment fields
Comment display

6. Scroll down to the **Add new field** block and complete the **Label** field with a sensible value (**Product rating** might be a good one), select **Fivestar Rating** as **Type of data to store**, and select the **Stars (rated while editing)** value for the **Form element to edit the data** field:



Add new field

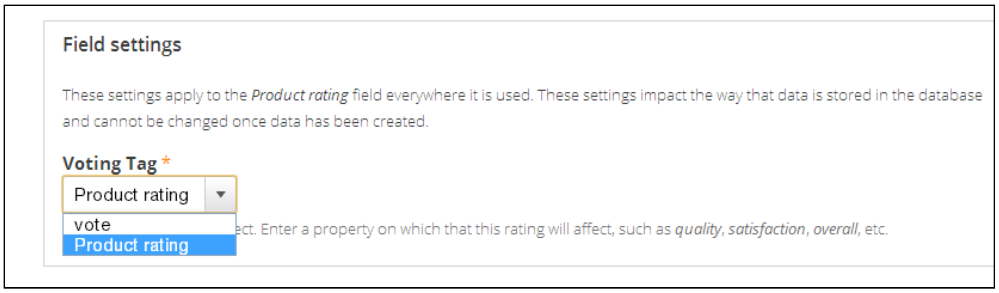
Label: Product rating

field_product_rating [Edit]

Type of data to store: Fivestar Rating

Form element to edit the data: Stars (rated while editing)

7. Next the **Field settings** screen will appear. Select **Product rating** from the **Voting Tag** field dropdown and click on **Save configuration** again:



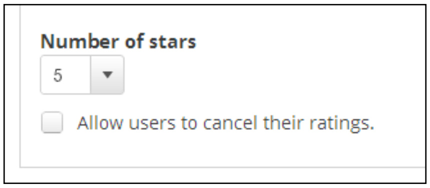
Field settings

These settings apply to the *Product rating* field everywhere it is used. These settings impact the way that data is stored in the database and cannot be changed once data has been created.

Voting Tag *: Product rating

ect. Enter a property on which that this rating will affect, such as *quality*, *satisfaction*, *overall*, etc.

8. Select the number of stars that you wish customers should be able to rate from in the **Number of stars** field. **5** is the default number and is quite a common number of stars to use for ratings, so you can leave it as it is if you're happy with that:



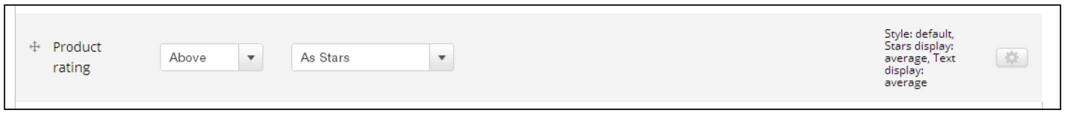
**Number of stars**

5

□ Allow users to cancel their ratings.

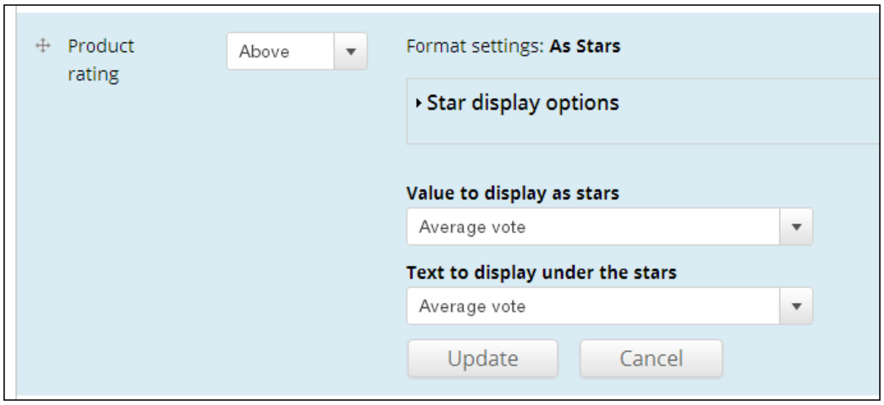
9. Click on the **Save settings** button to confirm your choices.

10. Now navigate to the **Manage display** tab, and scroll down until you see the **Product rating** field:



Product rating | Above | As Stars | Style: default, Stars display: average, Text display: average

11. Click on the gears icon to the right to configure this display. Set the **Value to display as stars** field to **Average vote**, and set the **Text to display under the stars** as **Average vote** too:



Product rating | Above

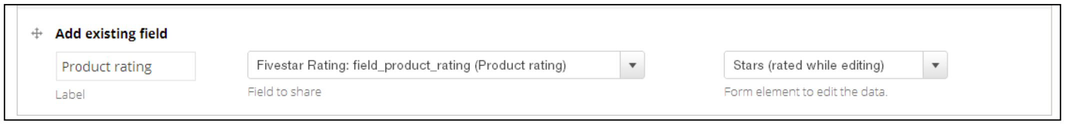
Format settings: **As Stars**

**Value to display as stars**: Average vote

**Text to display under the stars**: Average vote

By displaying the average vote here, you're showing your customers the overall rating for the product, as decided by your other customers. This can be a great way to give customers the proof that they need to buy the product, and a good way to learn about which, if any, of your products are failing.

12. Click on the **Update** button to save your changes, and then the **Save** button to save your content type's settings.
13. Navigate to the product display's **Comment fields** tab at the top of the screen.
14. Use the **Add existing field** tool and add an appropriate **Label** value. From the **Field to share** dropdown list, select the **Fivestar Rating: field_product_rating (Product rating)** value that you created previously:



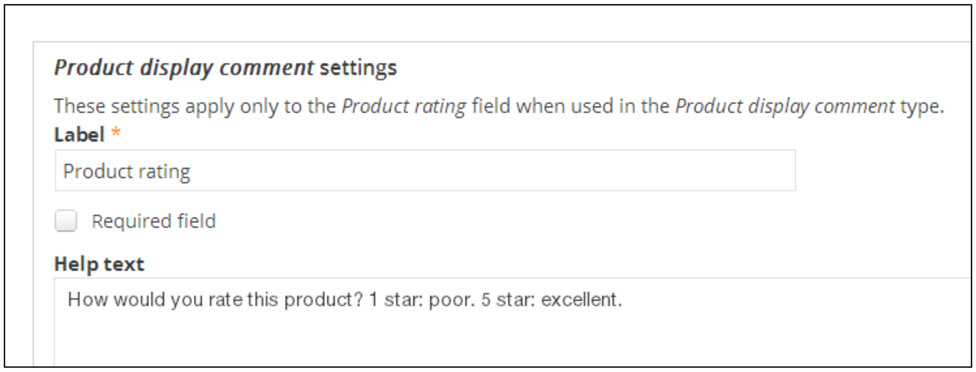
Add existing field

Product rating
Label

Fivestar Rating: field_product_rating (Product rating)
Field to share

Stars (rated while editing)
Form element to edit the data.

15. Leave the **Form** element to edit the data field as **Stars (rated while editing)**, and click on the **Save** button.
16. Next, you are presented with further options. You can provide some **Help text** to your customers:



*Product display comment* settings

These settings apply only to the *Product rating* field when used in the *Product display comment* type.

Label *

Product rating

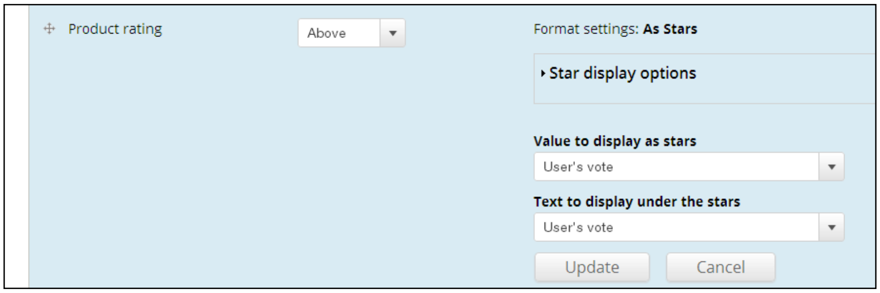
□ Required field

Help text

How would you rate this product? 1 star: poor. 5 star: excellent.

17. The **Star display** options block allows you to select the style of stars that you would like to use on your store; select your preferred value and continue.
18. Set the **Number of stars** field to `5`, or the same number that you set when customizing the product display content type.
19. Set the **Voting target** value to **Parent Node**, and click on **Save settings**.
20. Click on the **Comment display** tab at the top-right of your screen, and click on the gears icon next to the **Product rating** field to change its settings.

21. Change the **Value to display as stars** field to **User's vote**, and the **Text to display under the stars** as **User's vote**:



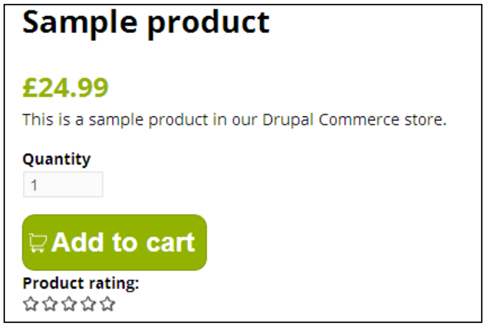
Product rating

Format settings: **As Stars**

Value to display as stars: User's vote

Text to display under the stars: User's vote

22. That's it! If you view a product on your Drupal Commerce store now, you will see the **Product rating** field appear as follows:



Sample product

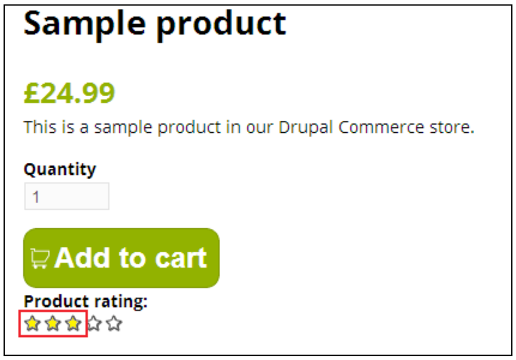
£24.99

This is a sample product in our Drupal Commerce store.

Quantity: 1

Product rating:

23. If you hover over the stars to vote here, you'll see that the stars are highlighted to indicate the rating that you're about to give:



Sample product

£24.99

This is a sample product in our Drupal Commerce store.

Quantity: 1

Product rating:

# Adding a contact form to your Drupal Commerce store

It's important to make yourself easily contactable when running an online store to help create trust between you and your customers. Would you make a purchase from a store that you couldn't contact?

Adding a contact form to your store provides an easy method for customers to get in touch with you about their orders and any queries about your products.

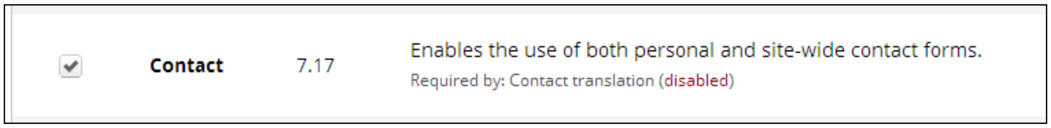
## Getting started

Log in to your Drupal Commerce administration panel.

## How to do it...

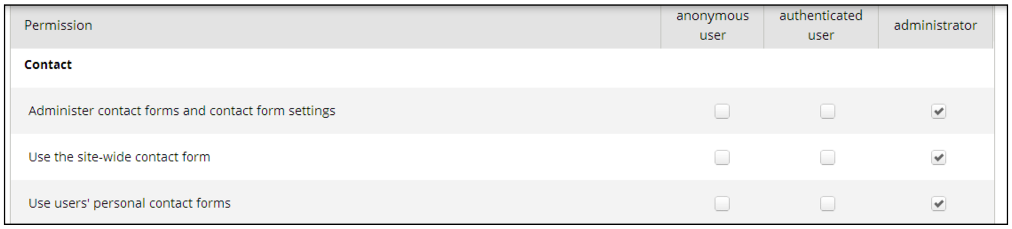
Getting your store's contact form up and running will involve a module and some configuration:

1. Navigate to **Modules** and locate the **Contact** module listed in the **Core** modules group:



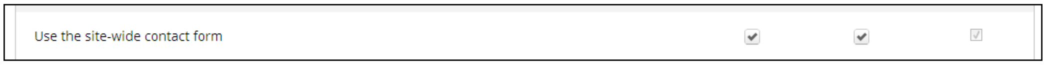
| | | | |
|---|---|---|---|
| ☑ | **Contact** | 7.17 | Enables the use of both personal and site-wide contact forms.<br>Required by: Contact translation (disabled) |

2. Click on the **Save configuration** button.
3. Find the **Contact** module in the list again and click on the **Permission** button to the right of it.
4. Here you will see three options. For the **Administrator contact forms and contact form** settings, ensure that only the **administrator** is checked:



| Permission | anonymous user | authenticated user | administrator |
|---|---|---|---|
| **Contact** | | | |
| Administer contact forms and contact form settings | □ | □ | ☑ |
| Use the site-wide contact form | □ | □ | ☑ |
| Use users' personal contact forms | □ | □ | ☑ |

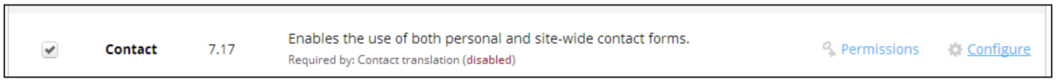
5. For the **Use the site-wide contact form** setting, allow both **anonymous users** and **authenticated users** to use the contact form:



| | | | |
|---|---|---|---|
| Use the site-wide contact form | ☑ | ☑ | ☑ |

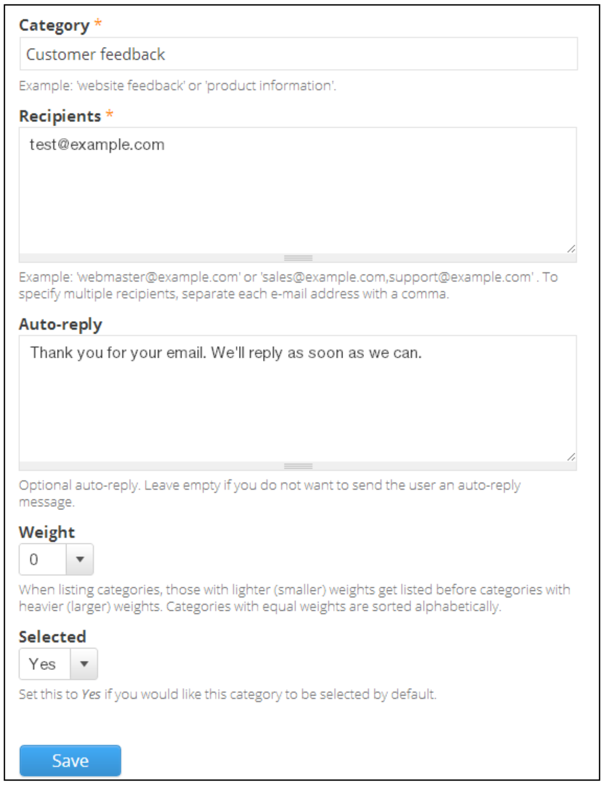
6. Click on the **Save Permissions** button and navigate back to **Modules**, locating the **Contact** module again.

7. Click on the **Configure** button to the right of the **Contact** module:



| | | | | | |
|---|---|---|---|---|---|
| ☑ | Contact | 7.17 | Enables the use of both personal and site-wide contact forms.<br>Required by: Contact translation (disabled) | Permissions | Configure |

8. Here you will see the settings for the site-wide contact form for your Drupal Commerce store. As you can see, the default contact form **Category** is **Website feedback**. Click on the **Add category** button to create a new customer service contact category:



Category *

Customer feedback

Example: 'website feedback' or 'product information'.

Recipients *

test@example.com

Example: 'webmaster@example.com' or 'sales@example.com,support@example.com' . To specify multiple recipients, separate each e-mail address with a comma.

Auto-reply

Thank you for your email. We'll reply as soon as we can.

Optional auto-reply. Leave empty if you do not want to send the user an auto-reply message.

Weight

0 ▾

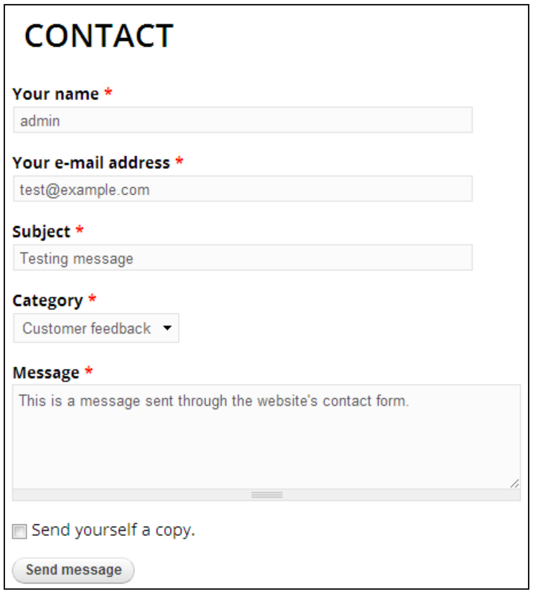
When listing categories, those with lighter (smaller) weights get listed before categories with heavier (larger) weights. Categories with equal weights are sorted alphabetically.

Selected

Yes ▾

Set this to *Yes* if you would like this category to be selected by default.

9. Enter the **Category** title and an e-mail address (you can add multiple e-mail addresses to receive the message if they are separated by commas) in the **Recipients** field.
10. You can specify a message that customers submitting a message will automatically receive after sending their message in the **Auto-reply** field. This can be useful to inform them their message was received and let them know when you are likely to respond.
11. Finally, set the **Selected** field to **Yes** to make this the default contact category on your store's form.
12. Click on the **Save** button and return to the frontend of your store now.
13. Enabling the **Contact** module will add a link to your menu to the contact form's page automatically, so navigate to this page (by default, its URL is `example.com/contact`) and you will see the contact form has appeared:



CONTACT

Your name *
admin

Your e-mail address *
test@example.com

Subject *
Testing message

Category *
Customer feedback

Message *
This is a message sent through the website's contact form.

☐ Send yourself a copy.

# Filtering UK postcodes or US ZIP codes from your Drupal Commerce store

There are certain types of store that may find it helpful to restrict the ability to order from particular regions of a country. This may be due to prohibitive delivery costs or different laws between states.

Drupal Commerce has a module that enables you to block orders either by a whitelist or blacklist of postcodes (for the UK) or ZIP codes (for the US). This recipe will take you through configuring the Commerce Postal Code Filter module to achieve this goal.

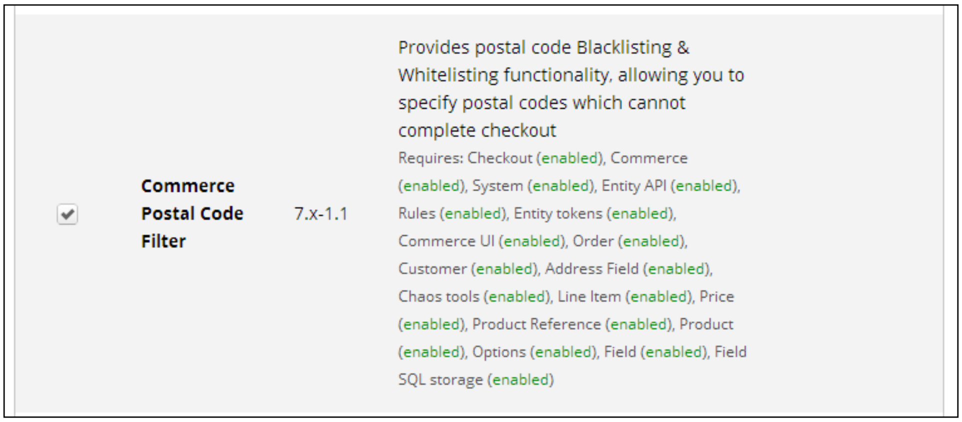
## Getting started

Log in to your Drupal Commerce store administration panel. Download the Commerce Postal Code Filter module from Drupal.org at `http://drupal.org/project/commerce_postal_code_filter`.

## How to do it...

To enable the filtering, you will need to install the module and configure it for your needs:

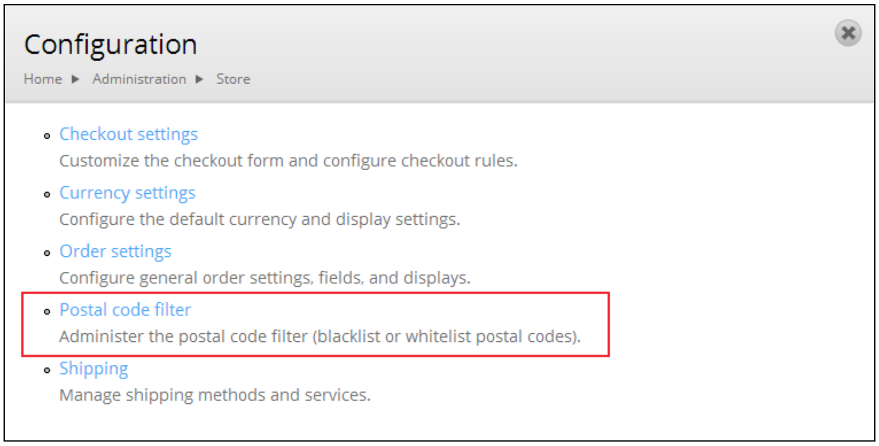
1. Install the Commerce Postal Code Filter module in your Drupal Commerce store and navigate to **Modules**.
2. Enable the module by checking the box. You will find it grouped under the **Commerce (contrib)** list of modules.



| | | | |
|---|---|---|---|
| ☑ | **Commerce Postal Code Filter** | 7.x-1.1 | Provides postal code Blacklisting & Whitelisting functionality, allowing you to specify postal codes which cannot complete checkout<br>Requires: Checkout (enabled), Commerce (enabled), System (enabled), Entity API (enabled), Rules (enabled), Entity tokens (enabled), Commerce UI (enabled), Order (enabled), Customer (enabled), Address Field (enabled), Chaos tools (enabled), Line Item (enabled), Price (enabled), Product Reference (enabled), Product (enabled), Options (enabled), Field (enabled), Field SQL storage (enabled) |

3. Click on the **Save configuration** button.

4. Now navigate to **Store** | **Configuration**, and click on the **Postal code filter** settings:

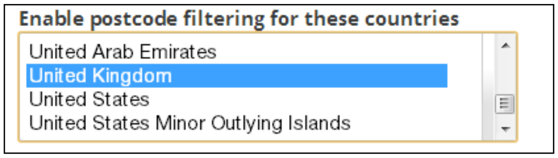


Configuration

Home ▶ Administration ▶ Store

- Checkout settings
  Customize the checkout form and configure checkout rules.
- Currency settings
  Configure the default currency and display settings.
- Order settings
  Configure general order settings, fields, and displays.
- Postal code filter
  Administer the postal code filter (blacklist or whitelist postal codes).
- Shipping
  Manage shipping methods and services.

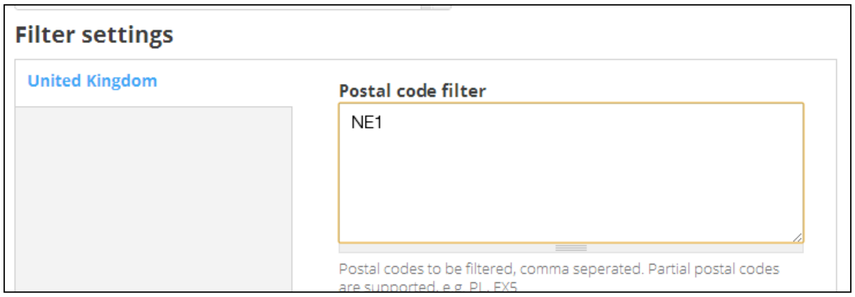
5. Select the country for which you wish to enable postcode or ZIP code filtering from the **Enable postcode filtering for these countries** field:



Enable postcode filtering for these countries

United Arab Emirates
United Kingdom
United States
United States Minor Outlying Islands

6. This will load country-specific filter options for you. Enter the postcodes that you wish to filter in the **Postal code filter** field. This field accepts partial postcodes, so both `NE1` and `NE1 1AB` would be accepted in this field. Separate multiple postcodes with a comma:



Filter settings

United Kingdom

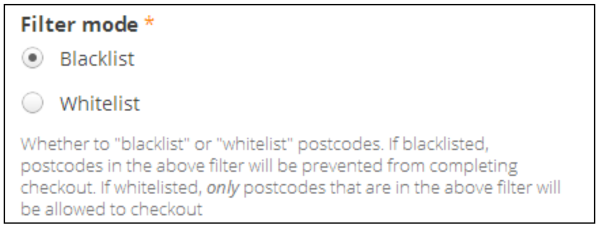
Postal code filter

NE1

Postal codes to be filtered, comma seperated. Partial postal codes are supported. e.g. PL, EX5

7. Next, you are presented with options about how you want to filter these postcodes.

8. If the **Filter mode** field is set to **Blacklist**, it prevents customers checking out if their postcode matches a value in the **Postal code filter** field. If the **Filter mode** field is set to **Whitelist**, then only postcodes that match values in the **Postal code filter** field are allowed to check out.



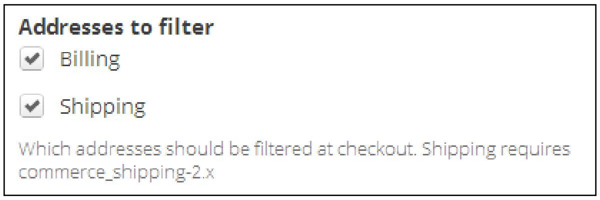
Filter mode *

● Blacklist

○ Whitelist

Whether to "blacklist" or "whitelist" postcodes. If blacklisted, postcodes in the above filter will be prevented from completing checkout. If whitelisted, *only* postcodes that are in the above filter will be allowed to checkout

9. Below the **Filter mode** field, you will see the **Addresses to filter** field, which controls which addresses the postcode filtering applies to. If you select the **Billing** value here, the postcode filtering applies only to the customer's billing address (if provided). If you check the **Shipping** value, the postcode filtering applies to the shipping (delivery) address the customer specifies. You can check both the **Billing** and **Shipping** values to apply the postcode filters you have set up to both the billing and shipping addresses:



Addresses to filter

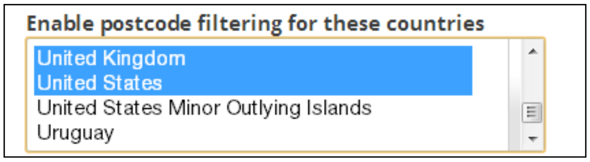
☑ Billing

☑ Shipping

Which addresses should be filtered at checkout. Shipping requires commerce_shipping-2.x

10. Click on the **Save postal code filter** options.

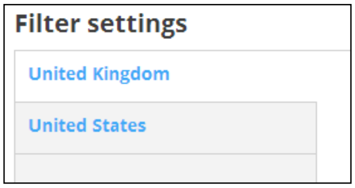
11. You can also select **United States** from the **Enable postcode filtering for these countries** option. If you want to select multiple countries, simply hold down the *Shift* key as you select them.



Enable postcode filtering for these countries

United Kingdom
United States
United States Minor Outlying Islands
Uruguay

12. This will display the filter settings for all countries that you have selected:



Filter settings

United Kingdom

United States

13. Select the **United States** tab to control the ZIP code filtering for this country, and you will see similar fields as for the United Kingdom postal code filters. In this way, you can filter as many countries' shipping and/or payment addresses for your store as you wish.

## How it works...

The Commerce Postal Code Filter module enhances Drupal Commerce by allowing you to restrict the countries or areas within the countries that your store accepts orders from, preventing customers from making orders that you are unable to fulfill (for example, to countries to whom you can't ship orders). This can help you cut down on unnecessary administration of your online business, such as having to e-mail customers to explain why you can't process their order, and making refunds from payment gateways such as PayPal, WorldPay, or SagePay.

When a customer goes to check out from your store, the module ensures you will check against the list of postcodes or ZIP codes you have specified, and decide whether to allow the order or not based on the settings you made (for example, whether you've enabled the blacklist or whitelist, and the list of postcode or ZIP codes you've specified). If the whitelist is enabled, only addresses that fall within the postcodes that you've specified will be accepted, while if the blacklist is enabled, all orders will be accepted unless they're in the specified postcode or ZIP code areas.

The module allows you to specify whether you want to filter the billing or shipping address, so it's perfectly feasible to filter postcodes from customers in your Drupal Commerce store to prevent a customer having an order delivered to a specific postcode or ZIP code area while they are able to pay for the order from outside your chosen delivery radius. This may be useful when your store only delivers locally, but you want to be able to process payment from further afar (for example, for customers buying gifts).

# Enhancing customer experience at checkout in Drupal Commerce with Ajax

As you've seen, Drupal Commerce is a great platform to start your e-commerce business with, but, as with any website, there are always things that you can do to enhance your store.

One of the more subtle enhancements you can make is with Ajax is that it allows customers to see their interactions (such as adding a product to the cart) happen without the entire web page having to reload.

## Getting started

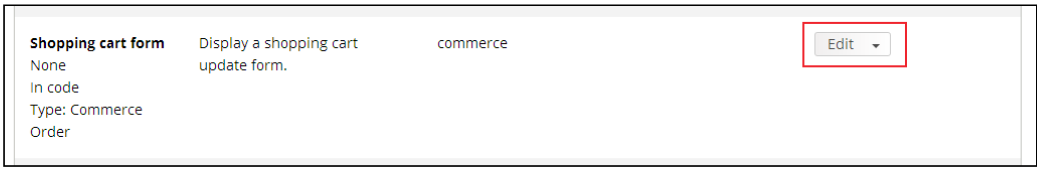
You will need to install the Commerce Cart Ajax module, which you can find at `http://drupal.org/project/dc_cart_Ajax`.

Log in to your Drupal Commerce store's administration panel.

## How to do it...

Once you've installed the Commerce Cart Ajax module, follow these steps:

1. Navigate to **Site settings** | **Views**.
2. In the list of **Views** that appear, locate the **Shopping cart form** view, and click on the **Edit** option to the right:



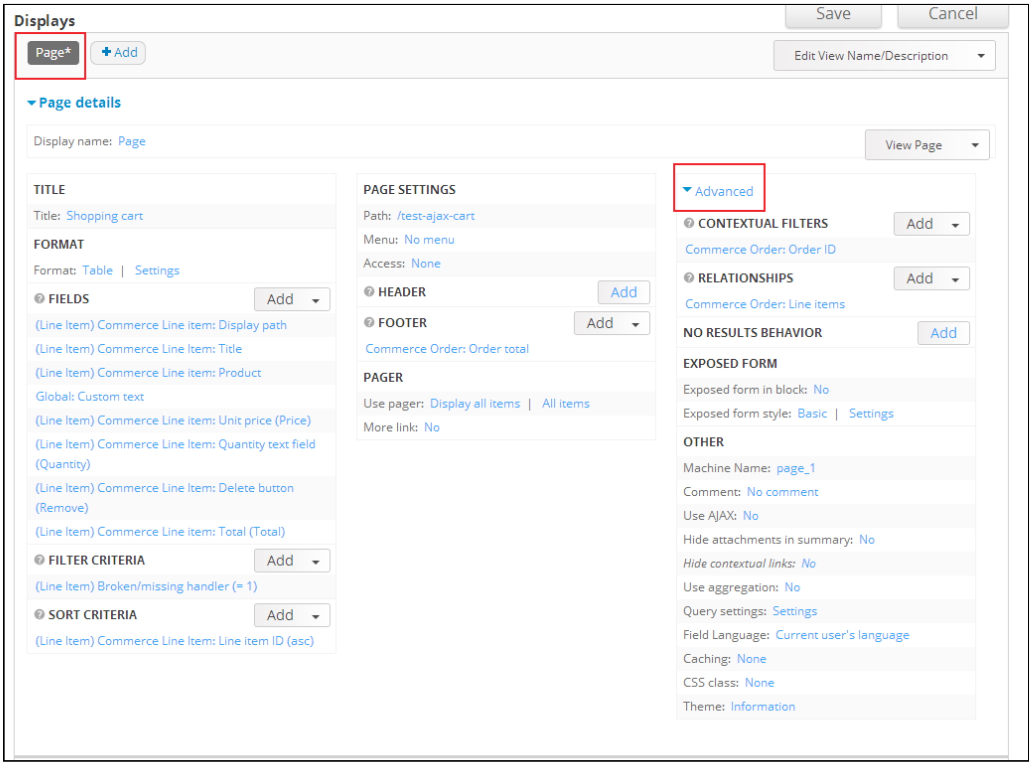
**Shopping cart form**
None
In code
Type: Commerce
Order

Display a shopping cart update form.

commerce

Alternatively, you can navigate to the `/admin/structure/views/view/commerce_cart_form/edit` URL on your Drupal installation.

3. In the **Page** display (selected at the top-left corner of the screen), expand the **Advanced settings** in the right-hand column:



Page details

Display name: Page

TITLE
Title: Shopping cart

FORMAT
Format: Table | Settings

FIELDS
(Line Item) Commerce Line item: Display path
(Line Item) Commerce Line Item: Title
(Line Item) Commerce Line item: Product
Global: Custom text
(Line Item) Commerce Line item: Unit price (Price)
(Line Item) Commerce Line Item: Quantity text field (Quantity)
(Line Item) Commerce Line Item: Delete button (Remove)
(Line Item) Commerce Line item: Total (Total)

FILTER CRITERIA
(Line Item) Broken/missing handler (= 1)

SORT CRITERIA
(Line Item) Commerce Line Item: Line item ID (asc)

PAGE SETTINGS
Path: /test-ajax-cart
Menu: No menu
Access: None

HEADER

FOOTER
Commerce Order: Order total

PAGER
Use pager: Display all items | All items
More link: No

Advanced

CONTEXTUAL FILTERS
Commerce Order: Order ID

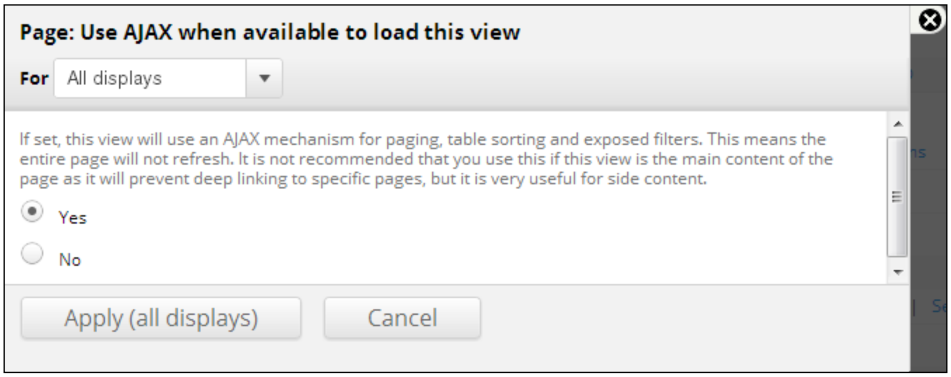
RELATIONSHIPS
Commerce Order: Line items

NO RESULTS BEHAVIOR

EXPOSED FORM
Exposed form in block: No
Exposed form style: Basic | Settings

OTHER
Machine Name: page_1
Comment: No comment
Use AJAX: No
Hide attachments in summary: No
*Hide contextual links:* No
Use aggregation: No
Query settings: Settings
Field Language: Current user's language
Caching: None
CSS class: None
Theme: Information

4. Toggle the **Use AJAX** value to **Yes**; a pop-up appears asking if you want to use Ajax:



Page: Use AJAX when available to load this view

For All displays

If set, this view will use an AJAX mechanism for paging, table sorting and exposed filters. This means the entire page will not refresh. It is not recommended that you use this if this view is the main content of the page as it will prevent deep linking to specific pages, but it is very useful for side content.

● Yes

○ No

5. Set the **For** field's dropdown to **All displays**, and ensure you've selected the **Yes** radio button.

6. Click on the **Apply (all displays)** button, and then click on the **Save** button on the views page to save all of your changes here.

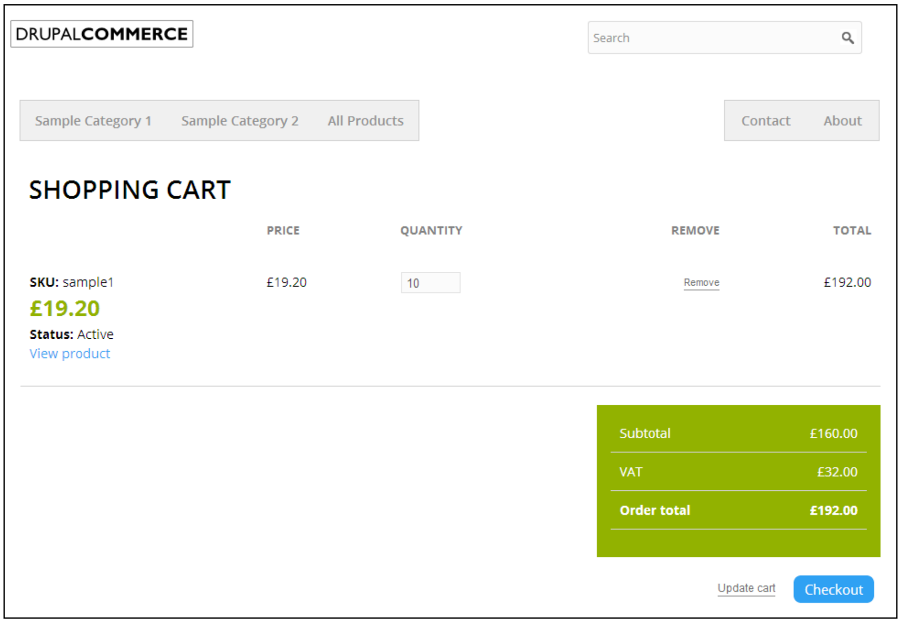
7. To check it's functioning fully, you can test the two behaviors on your store it has affected:

   - The **Remove** button on your cart page
   - The **Update** button on your cart page

8. Using the **Update** button, check whether the module is functioning as desired. Navigate to the frontend of your Drupal Commerce store and add products to your shopping cart.

9. View the cart page once you have added products to your shopping cart (you can get there by clicking on the **Checkout** button, which is located in the top-right corner of your screen in Drupal Commerce's default theme).

10. Once viewing your cart page, you will see the product(s) that you have added to your cart:

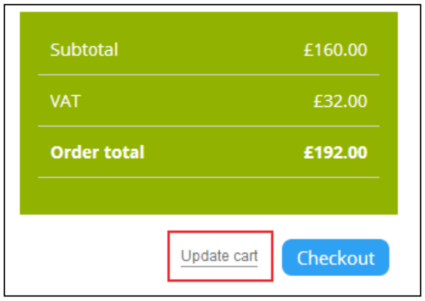


SHOPPING CART

| | PRICE | QUANTITY | REMOVE | TOTAL |
|---|---|---|---|---|
| **SKU:** sample1<br>£19.20<br>**Status:** Active<br>View product | £19.20 | 10 | Remove | £192.00 |

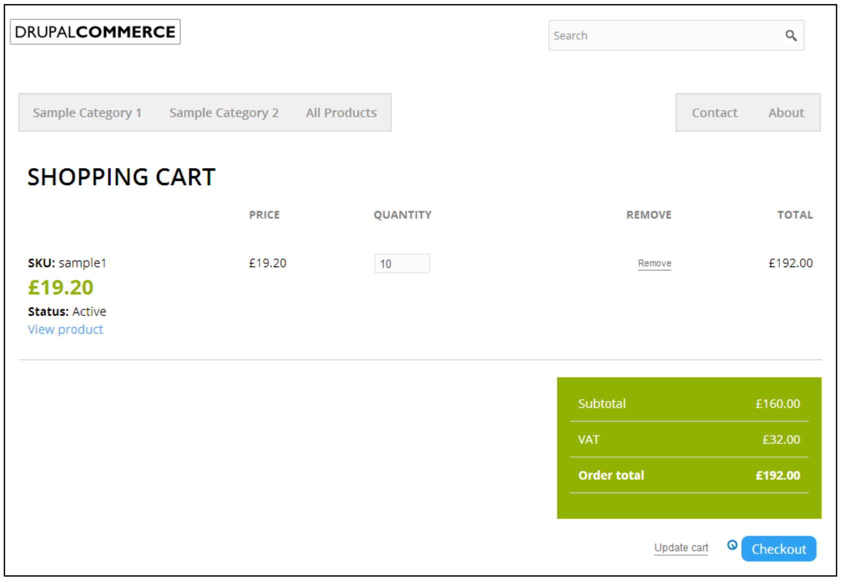
| | |
|---|---|
| Subtotal | £160.00 |
| VAT | £32.00 |
| **Order total** | **£192.00** |

11. Change the value in the **Quantity** field in the cart, and click on the **Update cart** button that is next to the **Checkout** button at the bottom-right corner of the screen:



| | |
|---|---|
| Subtotal | £160.00 |
| VAT | £32.00 |
| **Order total** | **£192.00** |

12. Once you click on this, you'll see an animated "loading" image appear next to the **Update cart** button if the module is functioning correctly; this means Drupal Commerce is updating the values that it needs to in your cart, without reloading the full page as it would have done without this module installed:



SHOPPING CART

| | PRICE | QUANTITY | REMOVE | TOTAL |
|---|---|---|---|---|
| **SKU:** sample1<br>£19.20<br>**Status:** Active<br>View product | £19.20 | 10 | Remove | £192.00 |

| | |
|---|---|
| Subtotal | £160.00 |
| VAT | £32.00 |
| **Order total** | **£192.00** |

## How it works...

Installing the Commerce Cart Ajax module enables Drupal to update elements within a page without having to reload the entirety of the page. This can be a good way to reduce load on your server, as it reduces the amount of information being sent, and it can also make it easier and quicker for customers to use your store.

Using JavaScript, **Ajax** (**Asynchronous JavaScript and XML**) calls the relevant values it needs to update asynchronously in the background, without reloading the page. The new values are then updated in the relevant area of the page once they're found or calculated and sent back to the customer's browser.

# A Blueprint for a T-shirt Store in Drupal Commerce

This appendix covers the customization and development of a t-shirt store in Drupal Commerce, including the following:

- Creating a t-shirt product type
- Embedding videos on Drupal Commerce products
- Creating a 10 percent offer in Drupal Commerce

## Introduction

This book contains what you need to get your Drupal Store up and running, and this chapter pulls the vital ingredients for a t-shirt store together with some tasks that will be common to many stores, from embedding videos on product pages to allowing customers to customize their products.

## Creating a t-shirt product type

As you have seen, Drupal Commerce provides a lot of flexibility when it comes to products. By adding additional fields to your installation's product's display, you can customize your store to support a custom "t-shirt" product type with size and color attributes. In this recipe, you will be learning about creating a t-shirt product type for your Drupal Commerce store.

## Getting started

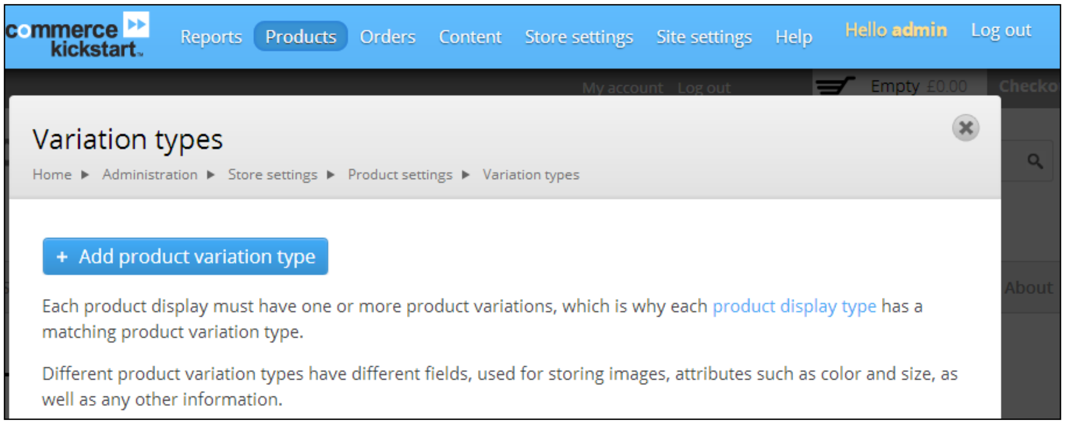
Before you begin, it's a good idea to have a list of the attributes that you want to have for the custom t-shirt product type. This example will add the following fields:

- **Size**: From a selection of small, medium, and large
- **Color**: From a selection of blue, red, yellow, and purple

## How to do it...

Log in to your Drupal Commerce administration panel, navigate to **Products** | **Variation** types, and follow these steps to create a custom product type for t-shirts in your store:

1. Click on the **Add product variation type** button:



Variation types

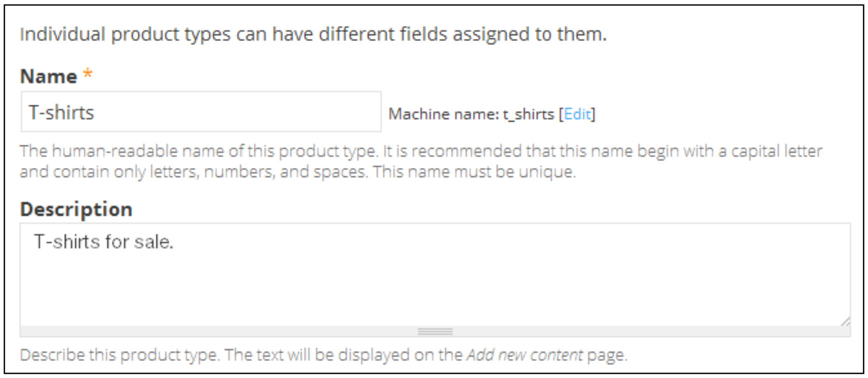
Home ▸ Administration ▸ Store settings ▸ Product settings ▸ Variation types

+ Add product variation type

Each product display must have one or more product variations, which is why each product display type has a matching product variation type.

Different product variation types have different fields, used for storing images, attributes such as color and size, as well as any other information.

2. Give the new product type a title by entering a suitable value in the **Name** field:



Individual product types can have different fields assigned to them.

Name *

T-shirts Machine name: t_shirts [Edit]

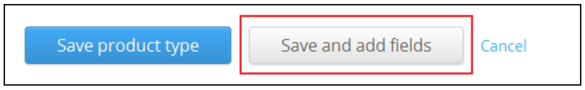
The human-readable name of this product type. It is recommended that this name begin with a capital letter and contain only letters, numbers, and spaces. This name must be unique.

Description

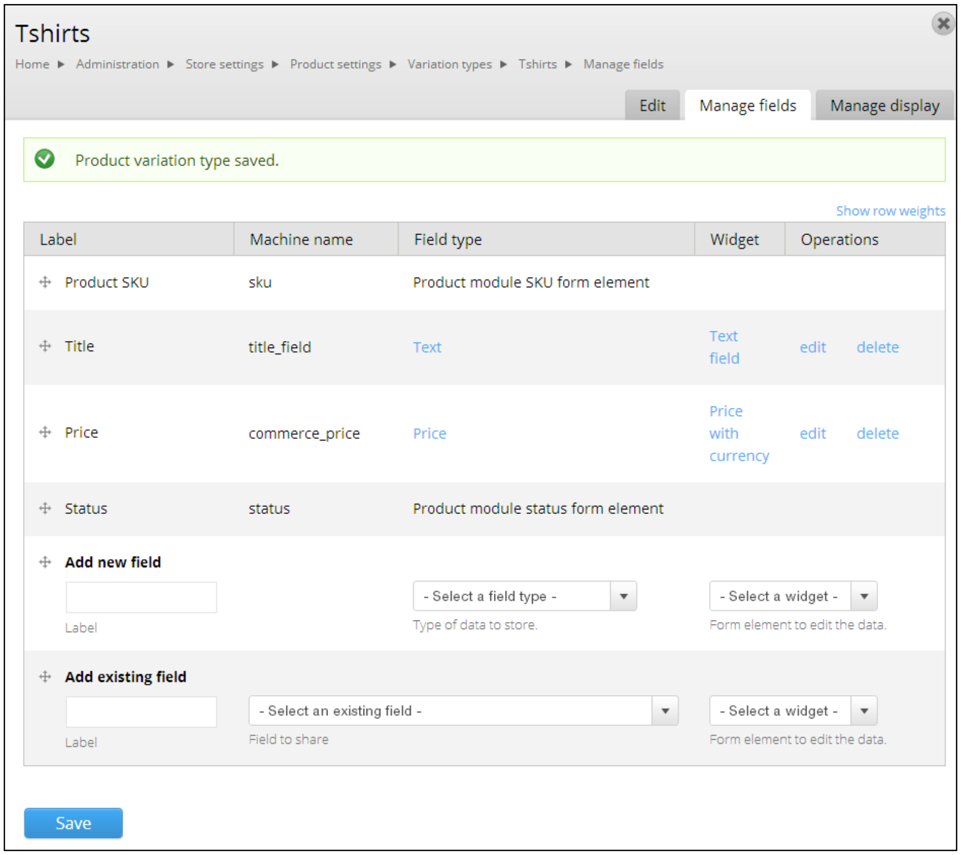
T-shirts for sale.

Describe this product type. The text will be displayed on the *Add new content* page.

3. Leave the **Default product variations of this type to be saved as new revisions when edited** checkbox checked, if you want Drupal Commerce to save a new revision of the variation each time you edit it.
4. Check the **Create matching product display type** checkbox if you want to create a Drupal Commerce product display type to display the products created using this product variation type.
5. Click on the **Save and add fields** button:



6. You will now see the **Product variation type saved** message, and a list of fields associated with your new t-shirt product type:



Tshirts

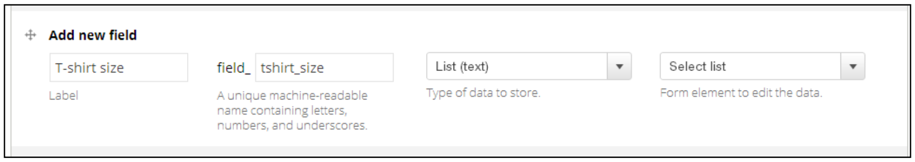
Home ▸ Administration ▸ Store settings ▸ Product settings ▸ Variation types ▸ Tshirts ▸ Manage fields

Product variation type saved.

| Label | Machine name | Field type | Widget | Operations | |
|---|---|---|---|---|---|
| ✢ Product SKU | sku | Product module SKU form element | | | |
| ✢ Title | title_field | Text | Text field | edit | delete |
| ✢ Price | commerce_price | Price | Price with currency | edit | delete |
| ✢ Status | status | Product module status form element | | | |
| ✢ **Add new field**<br>Label | | - Select a field type -<br>Type of data to store. | - Select a widget -<br>Form element to edit the data. | | |
| ✢ **Add existing field**<br>Label | - Select an existing field -<br>Field to share | | - Select a widget -<br>Form element to edit the data. | | |

7. Using the **Add new field** block, you can begin to customize this product type.

8. Populate the **Label** field with **T-shirt size**, and then select **List (text)** from the **Type of data to store** field, leaving the **Form element to edit the data** field as **Select list**:



Add new field

T-shirt size
Label

field_ tshirt_size
A unique machine-readable name containing letters, numbers, and underscores.

List (text)
Type of data to store.

Select list
Form element to edit the data.

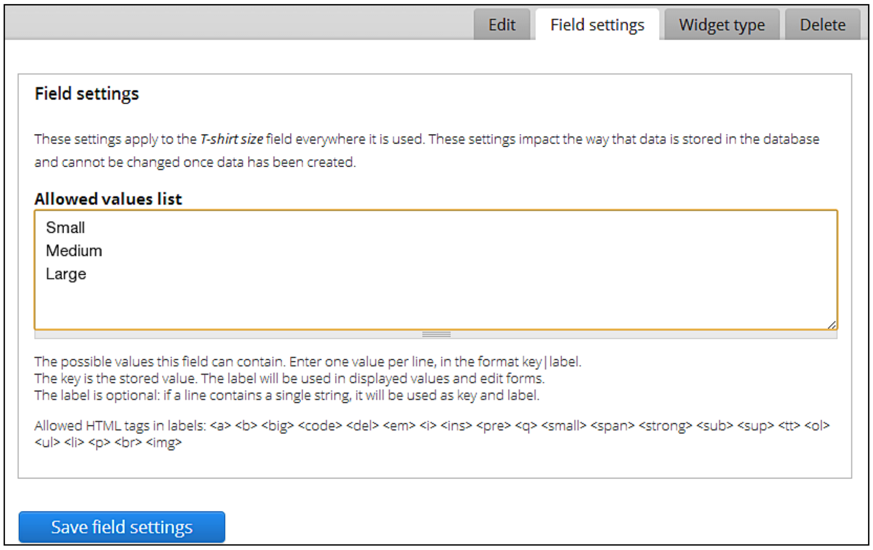
9. Now click on **Save**, and you'll be presented with the **Allowed values list** field on the **Field settings** tab; enter one value per line, using a vertical pipe (|) to separate a value from a label. For example, enter the following:

```
Medium-sized t-shirt | Medium
```

Entering the preceding value will present the customer with the option of **Medium** on the frontend of your store, while internally in the Drupal Commerce administration panel it will appear as **Medium-sized t-shirt**. The label is optional, so you can just enter one value per line if you prefer. The settings in the example are as follows:

- Small
- Medium
- Large

The preceding values means that you will see the relevant label (that is, **Small**, **Medium**, or **Large**) displayed both in the Drupal Commerce administration panel and as a customer on the frontend of your store. This system means you can provide your customers, with slightly different information to help them make an informed choice about the product they're buying.



Field settings

These settings apply to the *T-shirt size* field everywhere it is used. These settings impact the way that data is stored in the database and cannot be changed once data has been created.

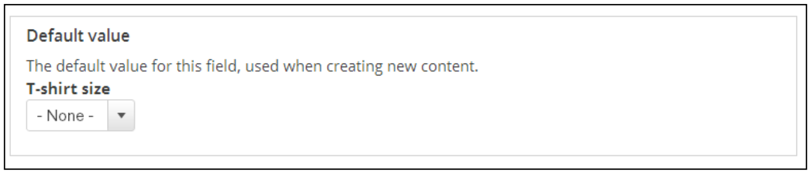
Allowed values list

Small
Medium
Large

The possible values this field can contain. Enter one value per line, in the format key|label.
The key is the stored value. The label will be used in displayed values and edit forms.
The label is optional: if a line contains a single string, it will be used as key and label.

Allowed HTML tags in labels: <a> <b> <big> <code> <del> <em> <i> <ins> <pre> <q> <small> <span> <strong> <sub> <sup> <tt> <ol> <ul> <li> <p> <br> <img>

10. Click on the **Save field settings** button to save your changes.
11. Leave the **Default value** set as **None** to prevent customers from ordering the wrong sized t-shirt:



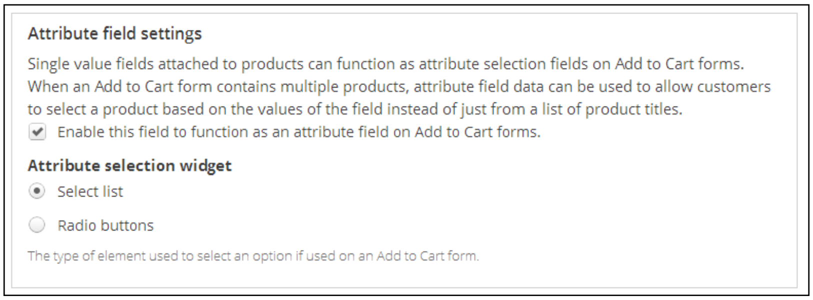
Default value

The default value for this field, used when creating new content.

T-shirt size

- None -

12. Check the **Attribute field settings** checkbox so it is enabled, which allows the dropdown to be shown on the "add to cart" features of Drupal Commerce, so customers can pick their t-shirt size:



Attribute field settings

Single value fields attached to products can function as attribute selection fields on Add to Cart forms. When an Add to Cart form contains multiple products, attribute field data can be used to allow customers to select a product based on the values of the field instead of just from a list of product titles.

☑ Enable this field to function as an attribute field on Add to Cart forms.

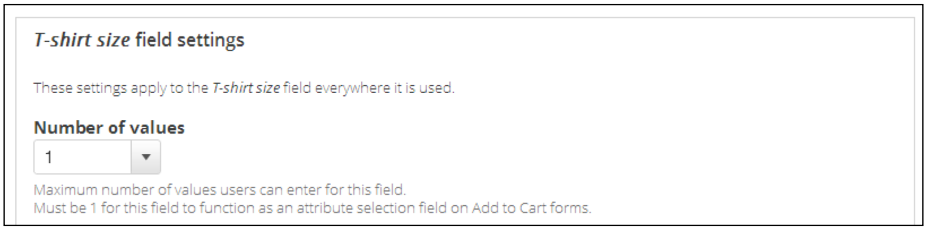
Attribute selection widget

● Select list

○ Radio buttons

The type of element used to select an option if used on an Add to Cart form.

13. Leave the **Number of values** field set to **1** in the **T-shirt size field settings** block (you only want customers to pick a t-shirt of one size), and click on the **Save settings** button again:



*T-shirt size* field settings

These settings apply to the *T-shirt size* field everywhere it is used.

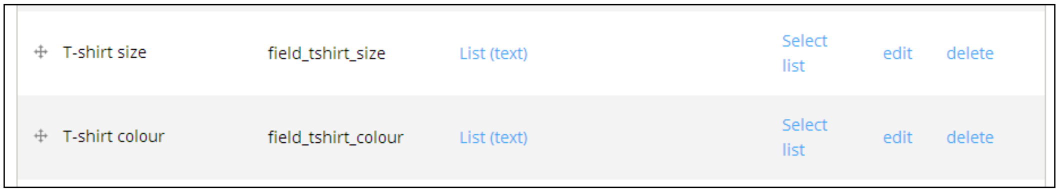
Number of values

1

Maximum number of values users can enter for this field.
Must be 1 for this field to function as an attribute selection field on Add to Cart forms.

14. You'll be returned to the **Manage fields** tab, where you can repeat steps 3 to 13 to add a color attribute for your t-shirts.

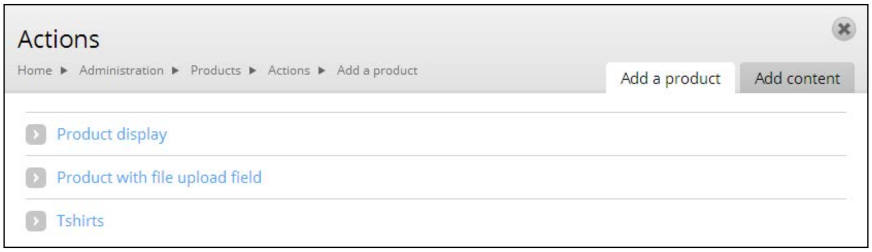
15. Once you've done that, you should be able to see the new **T-shirt colour** field listed in the **Manage fields** tab:



| | | | | | |
|---|---|---|---|---|---|
| ✥ T-shirt size | field_tshirt_size | List (text) | Select list | edit | delete |
| ✥ T-shirt colour | field_tshirt_colour | List (text) | Select list | edit | delete |

16. Your new product type is now ready, and you can add products of this type. Navigate to **Products** | **Add a product**.

17. Next, select the **T-shirts** link that will appear in alphabetical order at the bottom of the list:

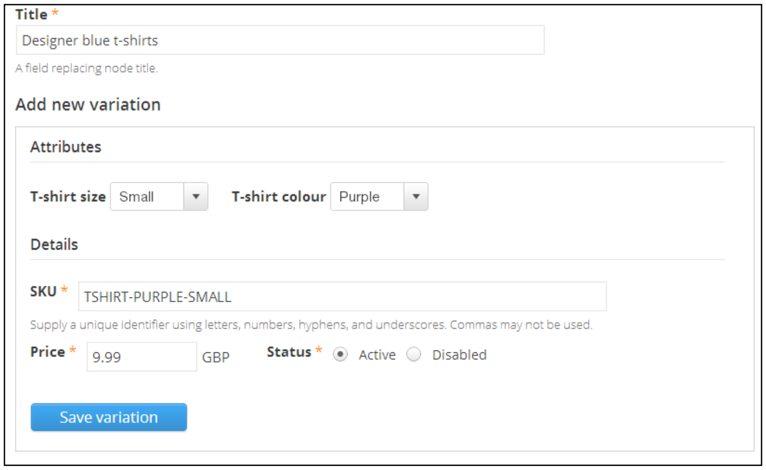


Actions

Home ▸ Administration ▸ Products ▸ Actions ▸ Add a product

- Product display
- Product with file upload field
- Tshirts

18. Populate the form that appears next, entering a **SKU**, **Title**, and **Price** for the product and selecting the **T-shirt size** and **T-shirt colour** values:



**Title** *

Designer blue t-shirts

A field replacing node title.

Add new variation

Attributes

**T-shirt size** Small ▾ **T-shirt colour** Purple ▾

Details

**SKU** * TSHIRT-PURPLE-SMALL

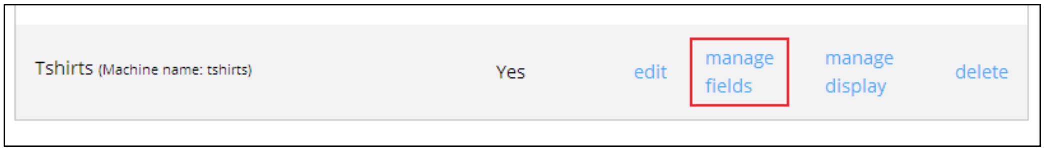
Supply a unique identifier using letters, numbers, hyphens, and underscores. Commas may not be used.

**Price** * 9.99 GBP **Status** * ● Active ○ Disabled

19. Ensure that the **Status** field is set to **Active** so that you (and the customers on your store) are able to view the products.

> Note that store administrators will still be able to view disabled products.

20. Click on the **Save variation** button, and add more products for each size/color combination that your store are available in your store for selling.
21. Now navigate to **Store settings** | **Variation types**, and select the **manage fields** option for the **Tshirts** content type that you created in the previous steps:



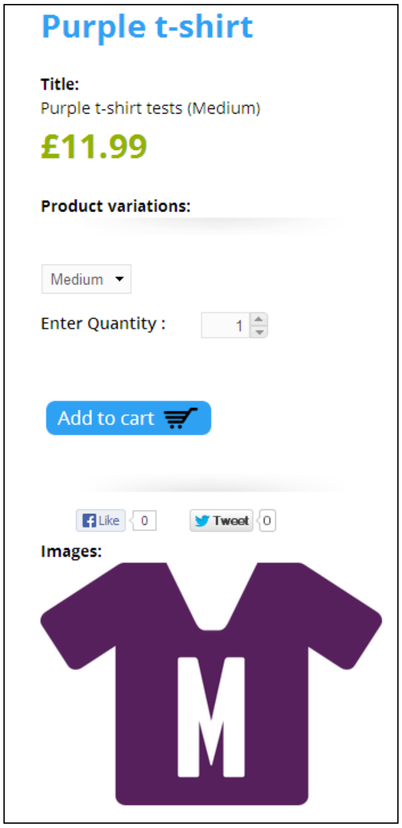
| | | | | | |
|---|---|---|---|---|---|
| Tshirts (Machine name: tshirts) | Yes | edit | manage fields | manage display | delete |

22. That's it! Your t-shirt product type is up and running.

## How it works...

Drupal Commerce allows you to add your own custom fields to product types so that you can assign your own attributes to different types of products within your store. So, using the preceding method, your store could sell t-shirts using a t-shirts product type that allows customers to select the size and color of their t-shirt. It could also sell teapots using a teapot product type, and allow customers to pick the capacity of the teapot they are buying instead.

The new t-shirt product type that you just created will display the products to your customers in a similar way:



Purple t-shirt

Title:

Purple t-shirt tests (Medium)

£11.99

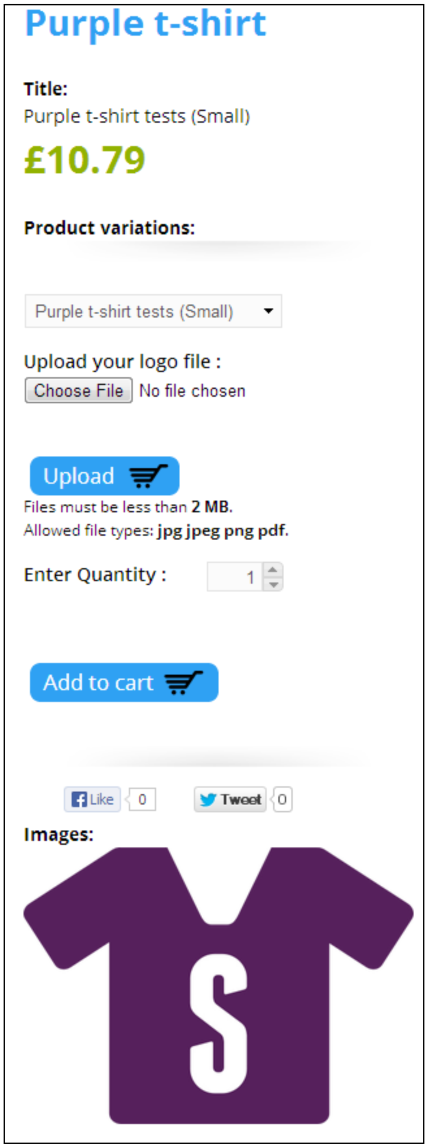
Product variations:

Medium

Enter Quantity : 1

Images:

Note the drop-down list that allows customers to pick their preferred size of t-shirt. If you change this to **Small**, the values that are different in the small t-shirt product variation will change too:



Purple t-shirt

**Title:**
Purple t-shirt tests (Small)

£10.79

**Product variations:**

Purple t-shirt tests (Small)

Upload your logo file :
No file chosen

Files must be less than **2 MB**.
Allowed file types: **jpg jpeg png pdf**.

Enter Quantity : 1

**Images:**

In particular, you can see that the product image has changed, and the price is updated.

# Embedding videos on Drupal Commerce products

With the growing uptake of higher bandwidth Internet connections throughout the world, video is becoming increasingly accessible as a means of displaying your products.

Adding a video of your products to your website can be a great way of showing your product to customers from more angles, as well as giving you a chance to provide narration about the features and benefits of your products.

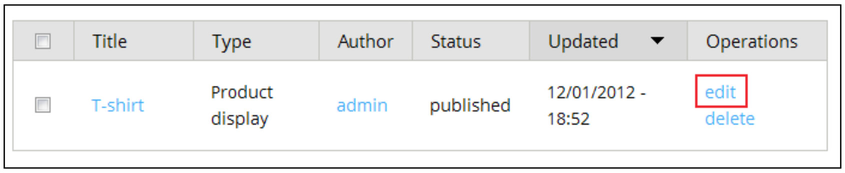
## Getting started

Log in to your Drupal Commerce store's administration panel.

## How to do it...

Now that you are logged in to Drupal Commerce as an administrator, perform the following steps:

1. Navigate to **Products** | **Manage products**.
2. Click on the **edit** link in the **Operations** column of the product that you want to add the video to:

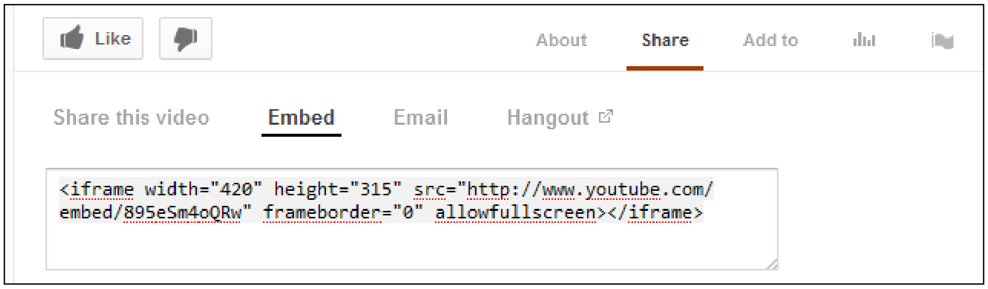


| ☐ | Title | Type | Author | Status | Updated ▼ | Operations |
|---|---|---|---|---|---|---|
| ☐ | T-shirt | Product display | admin | published | 12/01/2012 - 18:52 | edit<br>delete |

3. Now that you're editing the product, you will see the **Body** field that contains the description of the product.
4. Navigate to `www.youtube.com` (or other video sharing website) and locate the video of your product that you want to embed.

> You'll need a YouTube account to upload videos, but don't worry, it's free to sign up.

5. Navigate to the **Share** tab on YouTube.com, and then select the **Embed** tab to reveal the HTML to embed this video:

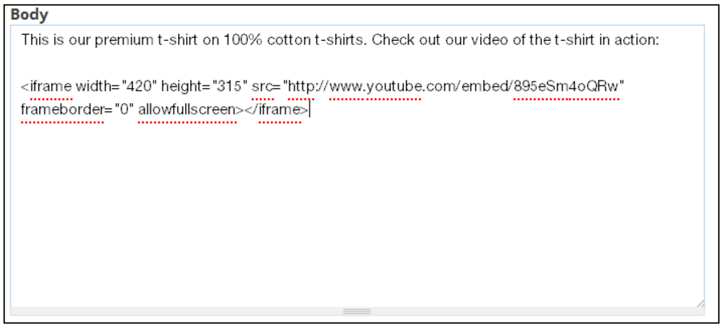


```
<iframe width="420" height="315" src="http://www.youtube.com/
embed/895eSm4oQRw" frameborder="0" allowfullscreen></iframe>
```

6. The HTML will look similar to the following:

```
<iframe width="420" height="315" src="http://www.youtube.com/
embed/895eSm4oQRw" frameborder="0" allowfullscreen></iframe>
```

7. Return to your Drupal Commerce administration panel, and insert this HTML into your page where you want the video to appear:



This is our premium t-shirt on 100% cotton t-shirts. Check out our video of the t-shirt in action:

```
<iframe width="420" height="315" src="http://www.youtube.com/embed/895eSm4oQRw"
frameborder="0" allowfullscreen></iframe>
```

8. Finally, click on the **Save** button at the bottom of this screen ensuring that full HTML is allowed to be posted within your product display's **Body** fields, and return to your store's frontend to view the product as a customer would see it.

## How it works...

You can use videos that you have uploaded to video sharing websites such as YouTube on your website by embedding them with the code that YouTube provides on its **Share** tab.

> Note that you'll only be able to embed videos that aren't protected on YouTube.

Video can be a great way to demonstrate your product to potential customers and give them a better understanding of what they can expect from their purchase.

# Creating a 10 percent offer in Drupal Commerce

You've probably seen 5 percent, 10 percent, and 20 percent discounts all over the web on various e-commerce stores, and you too can add one to your own e-commerce website to entice customers to buy from you.

This recipe will guide you through adding a 10 percent discount off on all customer orders in your Drupal Commerce store.

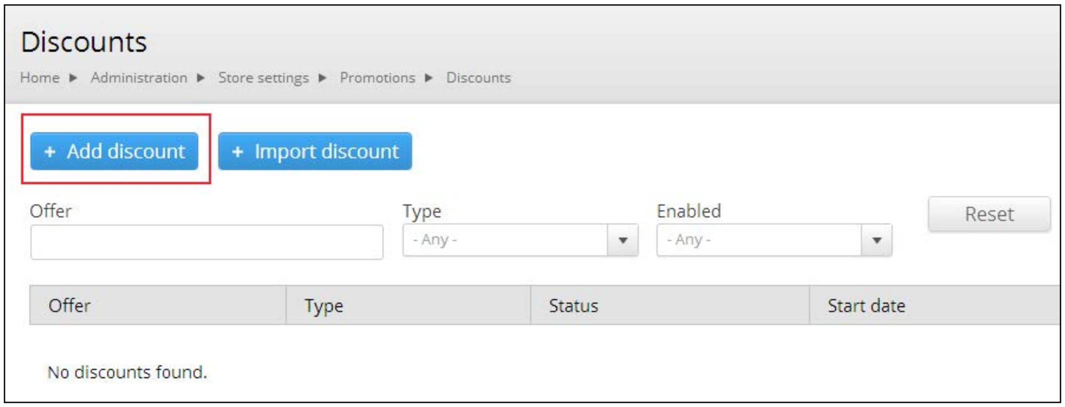
## Getting started

Make sure you're logged in to your Drupal Commerce website's administration panel.

## How to do it...

Once you're logged in, follow these steps to add a percentage-off promotion to your Drupal Commerce store:

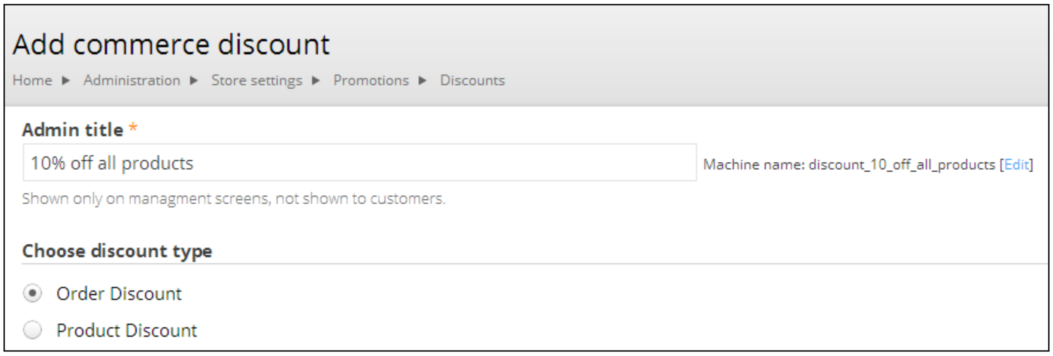
1. Navigate to **Store settings** | **Discounts**, and you will see the **Discounts** screen.
2. Click on the **Add discount** button towards the top-left corner of your screen:



| Offer | Type | Status | Start date |
|---|---|---|---|

No discounts found.

3. You can now start creating your promotion. Fill the **Admin title** field in with a suitable value, and select the **Order Discount** radio button in the **Choose discount type** block:



Add commerce discount

Home ▸ Administration ▸ Store settings ▸ Promotions ▸ Discounts

Admin title *

10% off all products

Machine name: discount_10_off_all_products [Edit]

Shown only on managment screens, not shown to customers.

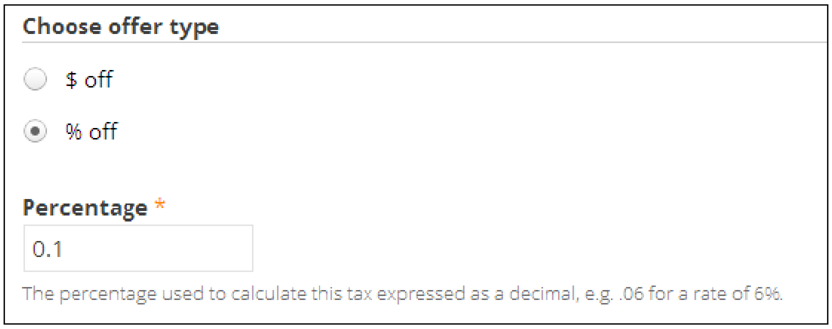
Choose discount type

● Order Discount

○ Product Discount

Note that this field is not shown to your customers.

4. The next option down the screen is the **Choose offer type** block; select the **% off** radio button here, and enter a value of **0.1** in the **Percentage** field:



Choose offer type

○ $ off

● % off

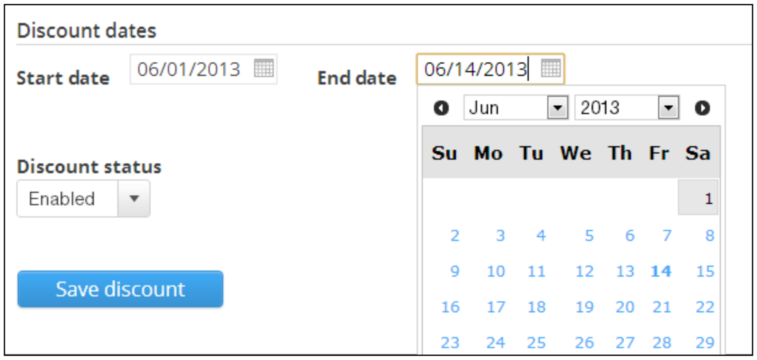
Percentage *

0.1

The percentage used to calculate this tax expressed as a decimal, e.g. .06 for a rate of 6%.

The **Percentage** field needs to be a decimal value, so **10%** if represented as **0.1**, while **5%** would be represented as **0.05** here, and **25%** would be represented as **0.25**.

5. If you wish to run your promotion for a limited amount of time, use the **Start date** and **End date** fields to set the period for which you want to run the offer:



Discount dates

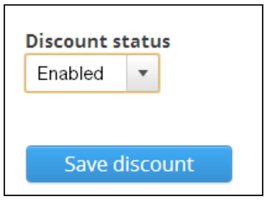
Start date: 06/01/2013

End date: 06/14/2013

| Su | Mo | Tu | We | Th | Fr | Sa |
|---|---|---|---|---|---|---|
| | | | | | | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |

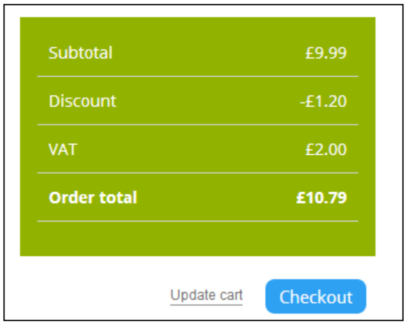
Discount status: Enabled

6. Check whether the **Discount status** field is set to **Enabled** (its default value). Now click on the **Save discount** button at the bottom of the screen:



Discount status: Enabled

7. To test your newly added promotion, navigate to the frontend of your Drupal Commerce store, and go through the checkout process; you will see the 10 percent discount applied:



| | |
|---|---|
| Subtotal | £9.99 |
| Discount | -£1.20 |
| VAT | £2.00 |
| **Order total** | **£10.79** |

By default, Drupal Commerce will only apply the discount to the product's cost (excluding any taxes that you have set up). You can apply the discount to shipping costs as well in the **Settings** tab, available on the **Discounts** screen at **Store settings | Discounts**.

# Index

# Thank you for buying Building E-commerce Sites with Drupal Commerce Cookbook

## About Packt Publishing

Packt, pronounced 'packed', published its first book "*Mastering phpMyAdmin for Effective MySQL Management*" in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: `www.packtpub.com`.

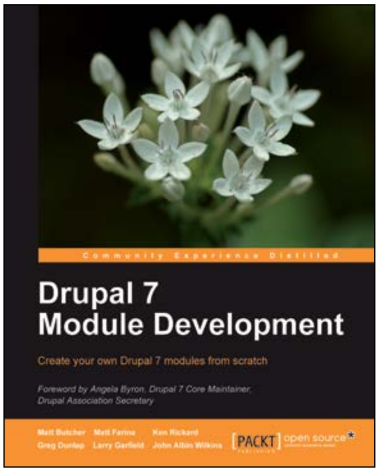
## About Packt Open Source

In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licences, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

## Writing for Packt

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to author@packtpub.com. If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.


Community Experience Distilled
Drupal 7
Module Development
Create your own Drupal 7 modules from scratch
Foreword by Angela Byron, Drupal 7 Core Maintainer, Drupal Association Secretary
Matt Butcher Matt Farina Ken Rickard
Greg Dunlap Larry Garfield John Albin Wilkins
PACKT open source

## Drupal 7 Module Development

ISBN: 978-1-84951-116-2 Paperback: 420 pages

Create your own Drupal 7 modules from scratch

1. Specifically written for Drupal 7 development
2. Write your own Drupal modules, themes, and libraries
3. Discover the powerful new tools introduced in Drupal 7
4. Learn the programming secrets of six experienced Drupal developers

Community Experience Distilled
Drupal Search Engine
Optimization
Drive people to your site with this supercharged guide to Drupal SEO
Ric Shreves
PACKT open source

## Drupal Search Engine Optimization

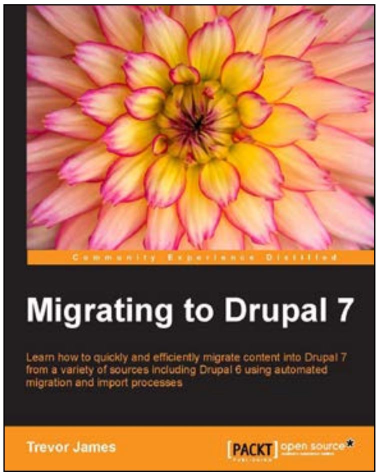
ISBN: 978-1-84951-878-9 Paperback: 116 pages

Drive poeple to your site with this supercharged guide to Drupal SEO

1. Learn how to create a search engine-optimized Drupal website
2. Packed full of tips to help you develop an appropriate SEO strategy
3. Discover the right configurations and extensions for SEO purposes

Please check **www.PacktPub.com** for information on our titles

[PACKT] open source
community experience distilled
PUBLISHING


Community Experience Distilled
Migrating to Drupal 7
Learn how to quickly and efficiently migrate content into Drupal 7 from a variety of sources including Drupal 6 using automated migration and import processes
Trevor James
[PACKT] open source

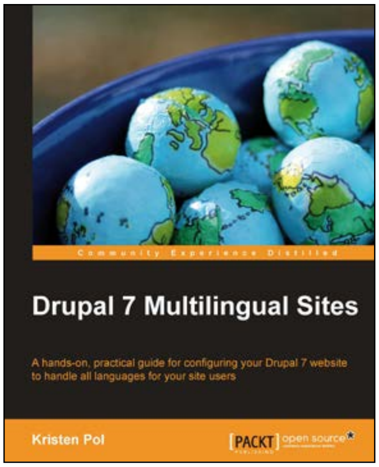
## Migrating to Drupal 7

ISBN: 978-1-78216-054-0 Paperback: 158 pages

Learn how to quickly and efficiently migrate content into Drupal 7 from a variety of sources including Drupal 6 using automated migration and import processes

1. Learn how to import content and data into your Drupal 7 site from other websites, content management systems, and databases
2. Upgrade your Drupal 6 site to Drupal 7 and migrate your CCK based content into the Drupal 7 fields based framework
3. Use modules that will automate the import and migration process including the Feeds and Migrate modules

---


Community Experience Distilled
Drupal 7 Multilingual Sites
A hands-on, practical guide for configuring your Drupal 7 website to handle all languages for your site users
Kristen Pol
[PACKT] open source

## Drupal 7 Multilingual Sites

ISBN: 978-1-84951-818-5 Paperback: 140 pages

A hands-on, practical guide for configuring your Drupal 7 website to handle all languages for your site users

1. Prepare your Drupal site to handle content in different languages easily
2. Apply the numerous multilingual modules to your Drupal site and configure it for any number of different languages
3. Organize the multilingual pieces into logical areas for easier handling

Please check **www.PacktPub.com** for information on our titles