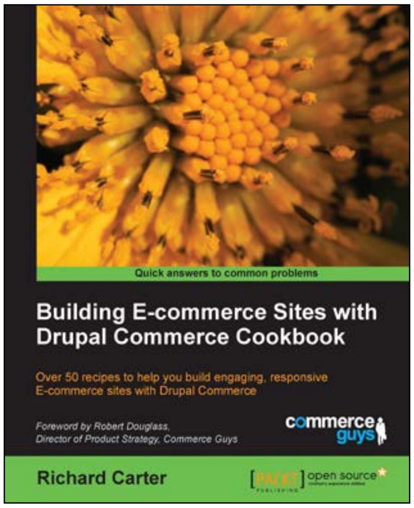
Building E-commerce Sites with Drupal Commerce Cookbook

Richard Carter


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 open source

Chapter No. 4

"Useful Modules for Drupal Commerce"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.4 "Useful Modules for Drupal Commerce"

A synopsis of the book’s content

Information on where to buy this book

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.

Building E-commerce Sites with Drupal Commerce Cookbook

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.

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.

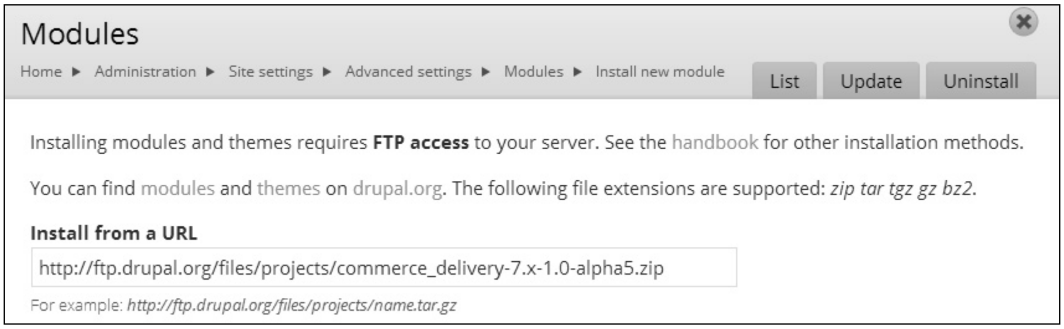
Getting started

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.

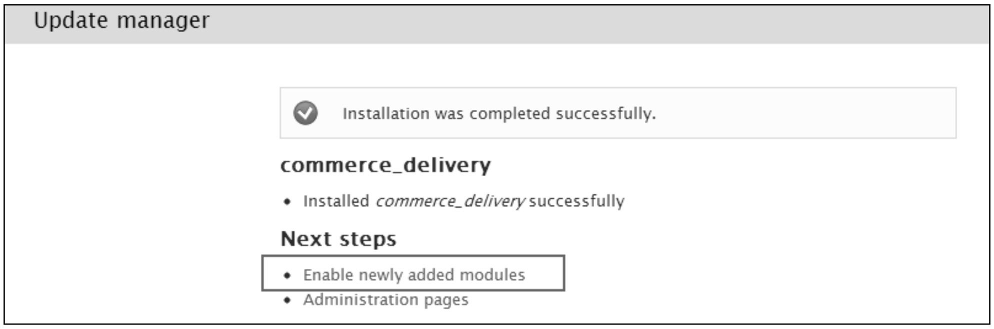
You can find modules and themes on drupal.org. The following file extensions are supported: *zip tar tgz gz bz2*.

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

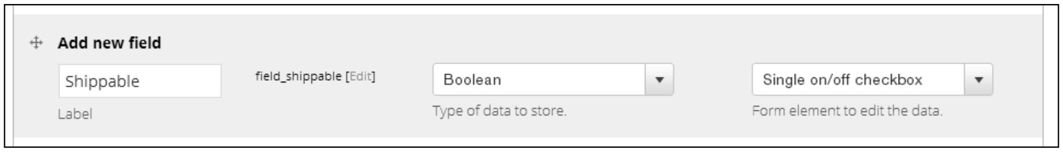
Next steps

- 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:



Add new field

Shippable

Label

field_shippable [Edit]

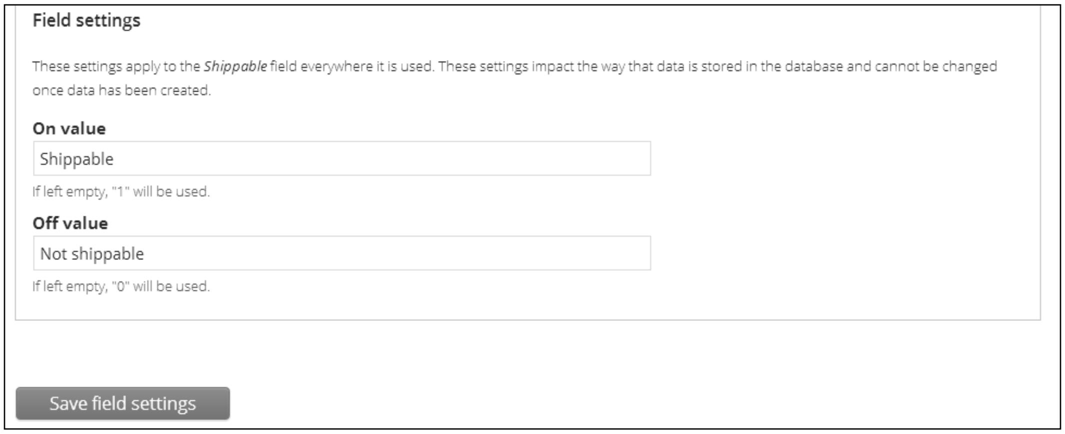
Boolean

Type of data to store.

Single on/off checkbox

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

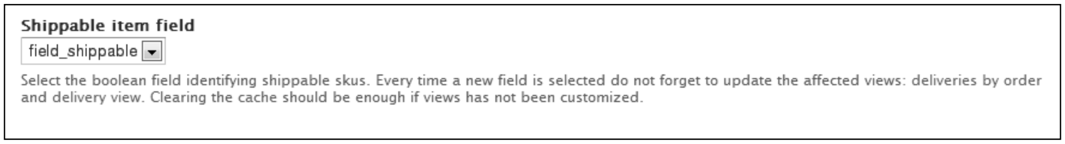
If left empty, "0" will be used.

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

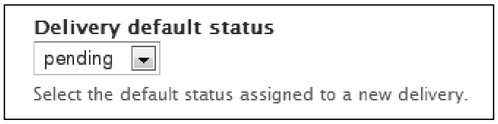
Select the boolean field identifying shippable skus. Every time a new field is selected do not forget to update the affected views: deliveries by order and delivery view. Clearing the cache should be enough if views has not been customized.

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:

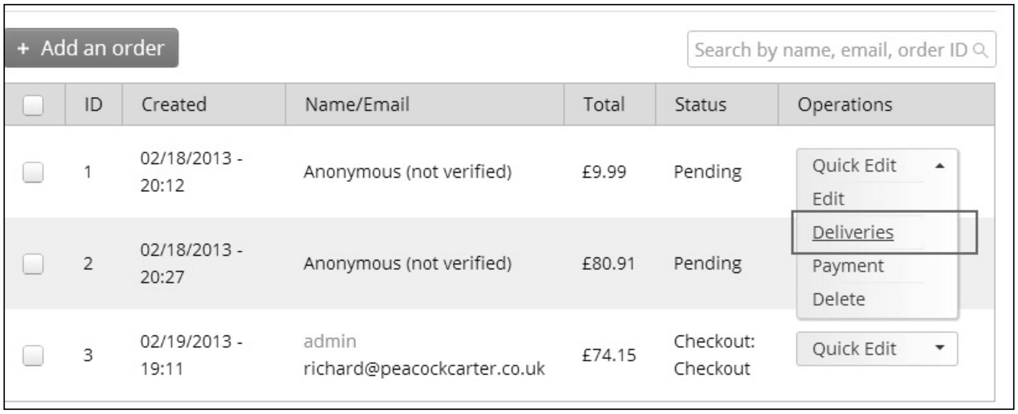


Delivery default status

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
□	2	02/18/2013 - 20:27	Anonymous (not verified)	£80.91	Pending	
□	3	02/19/2013 - 19:11	admin 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.

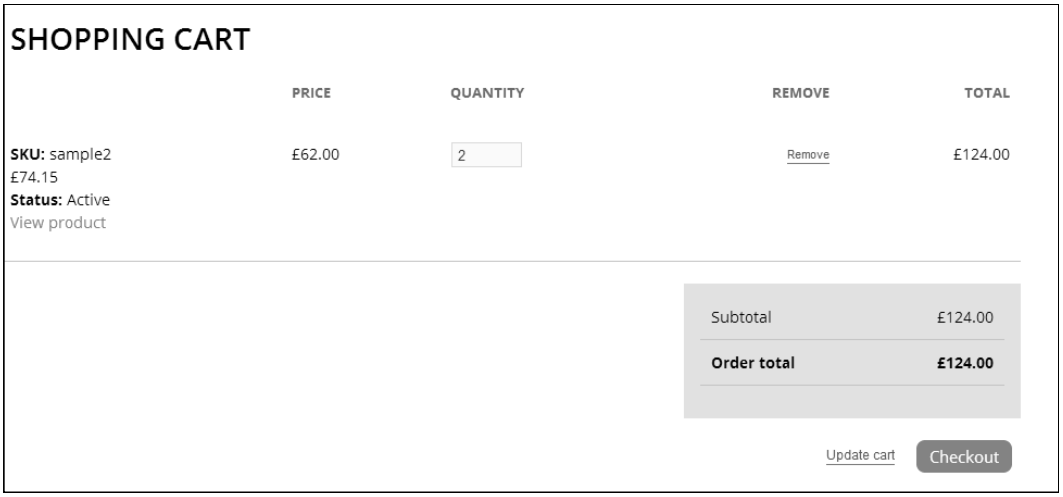
Getting started

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:

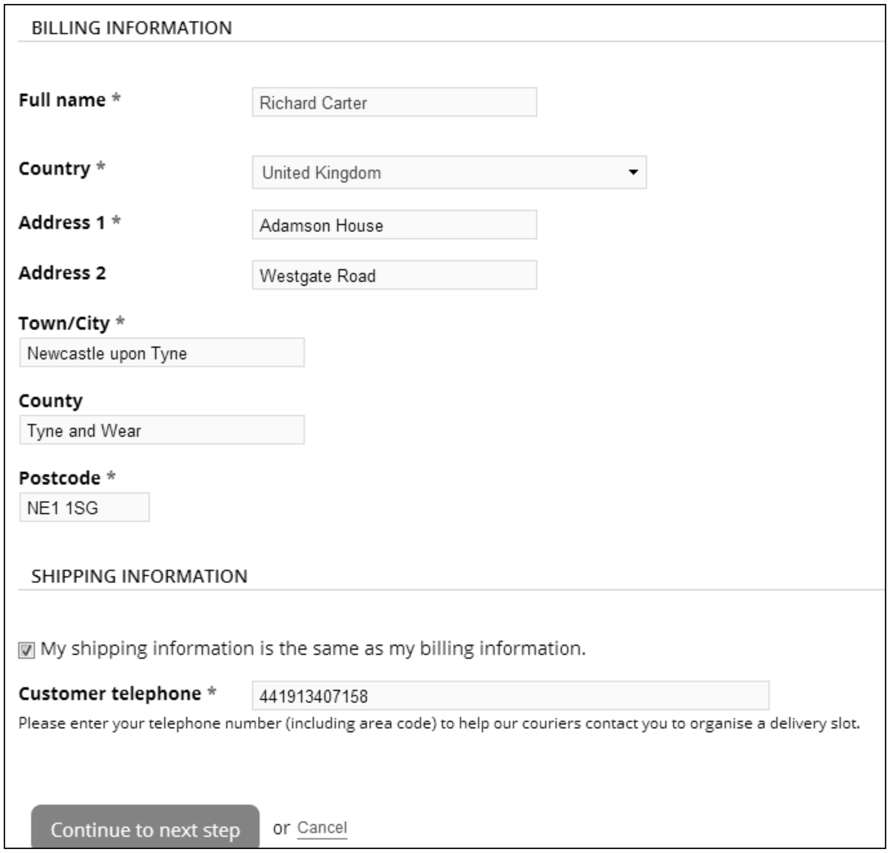


SHOPPING CART

	PRICE	QUANTITY	REMOVE	TOTAL
SKU: sample2 £74.15 **Status:** Active 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

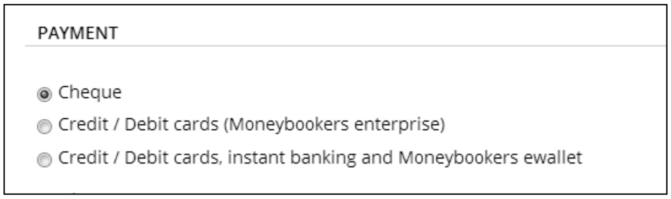
Postcode *
NE1 1SG

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.

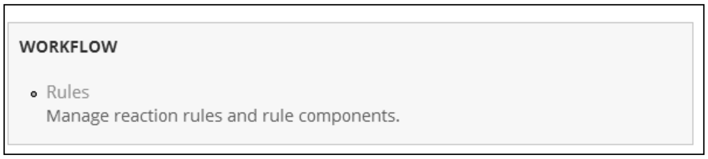
Getting started

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:

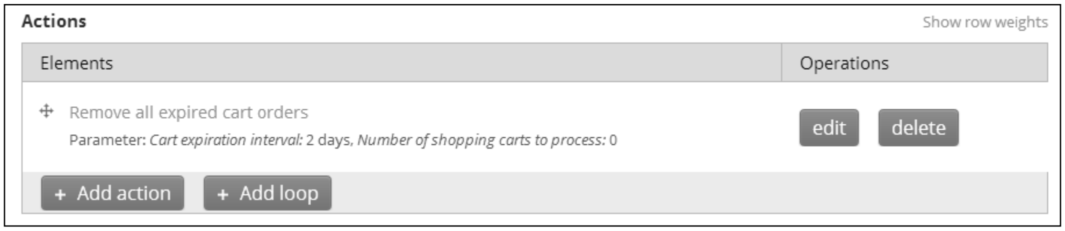
1. Next, you need to configure your store's rules to allow this module to function. Navigate to **Site settings** | **Configuration**, and under the **Workflow heading**, click on **Rules**:



WORKFLOW

- Rules
  Manage reaction rules and rule components.

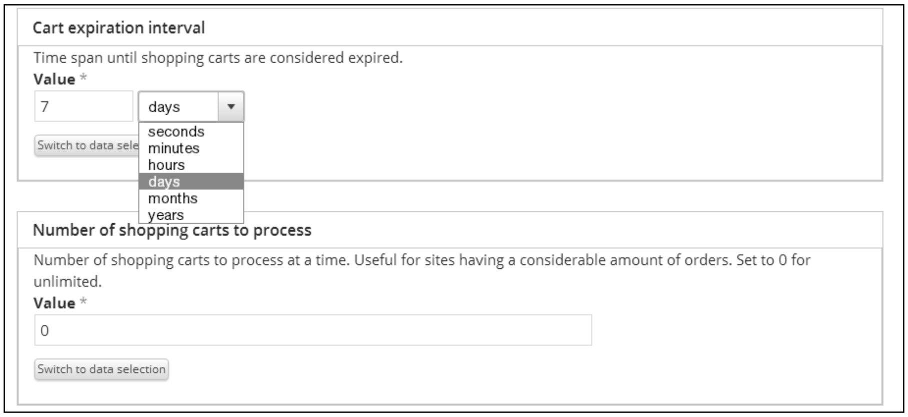
2. If you scroll to the bottom of the list of **Active** rules, you will now see a **Delete expired carts** option.
3. Click on the **edit** option in the **Operations** column to edit this rule, and you're presented with a screen to edit the **Delete expired carts** reaction rule.
4. Edit the **Remove 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 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

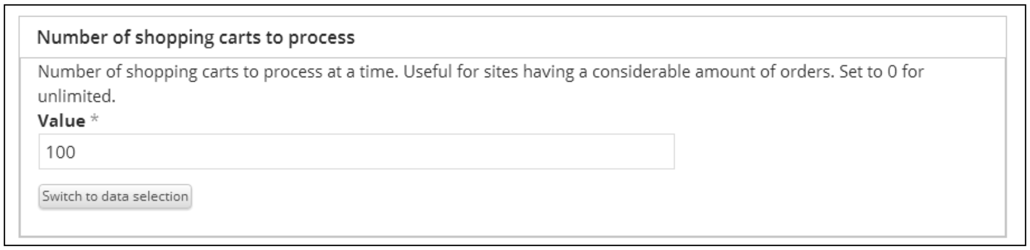
Number of shopping carts to process

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.

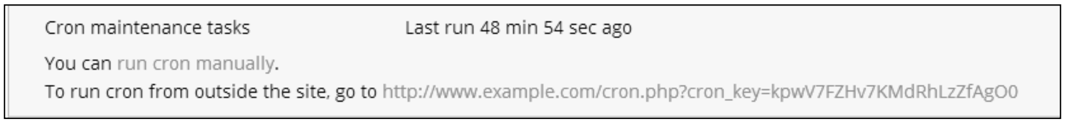
Value *

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.

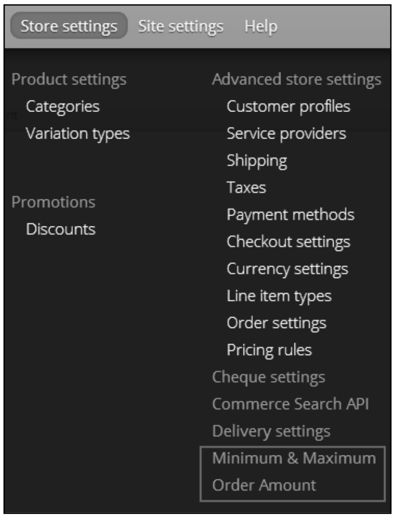
Getting started

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.

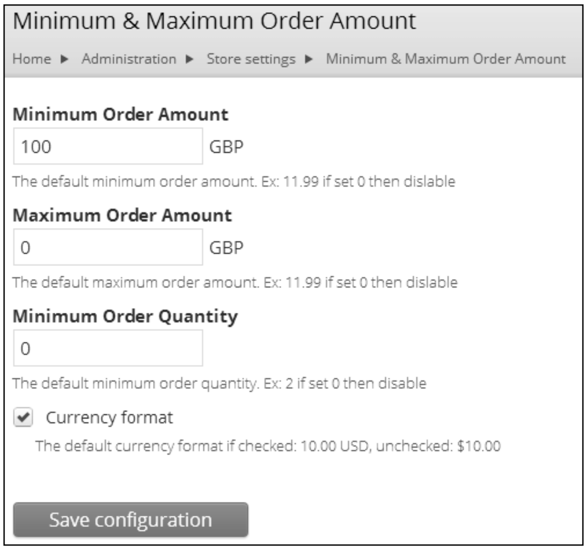
How to do it...

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

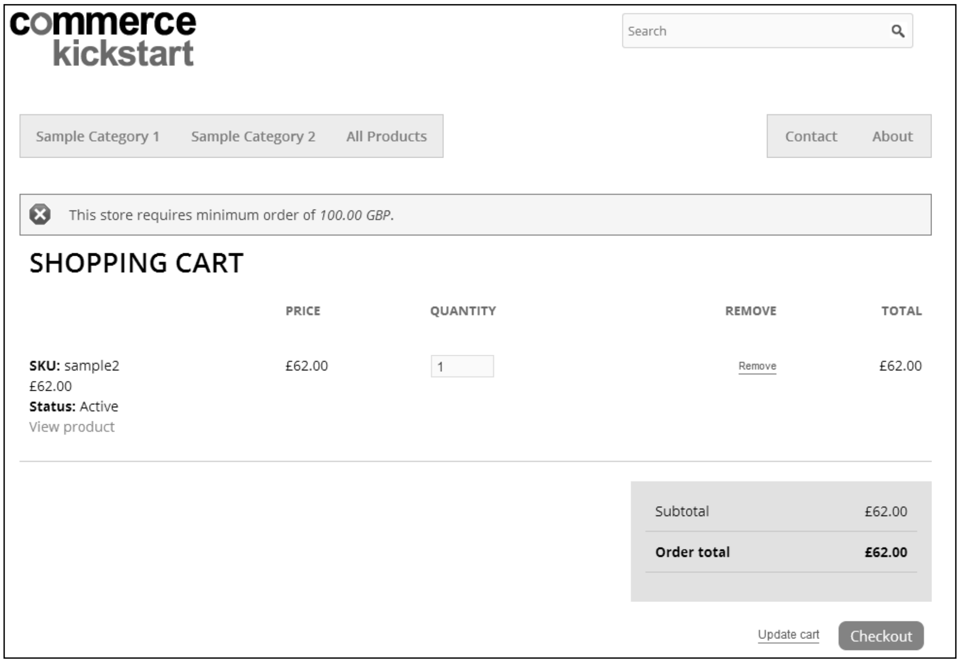
0

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

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 £62.00 **Status:** Active 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.

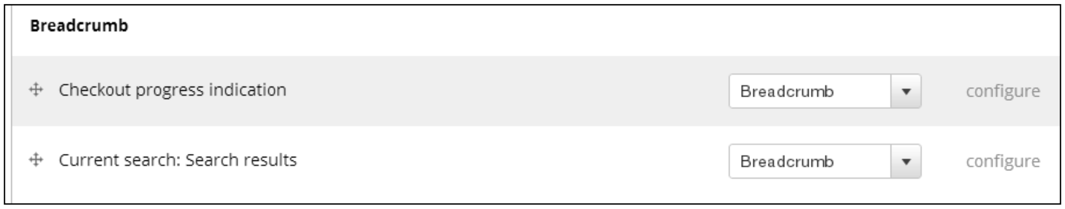
Getting started

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:

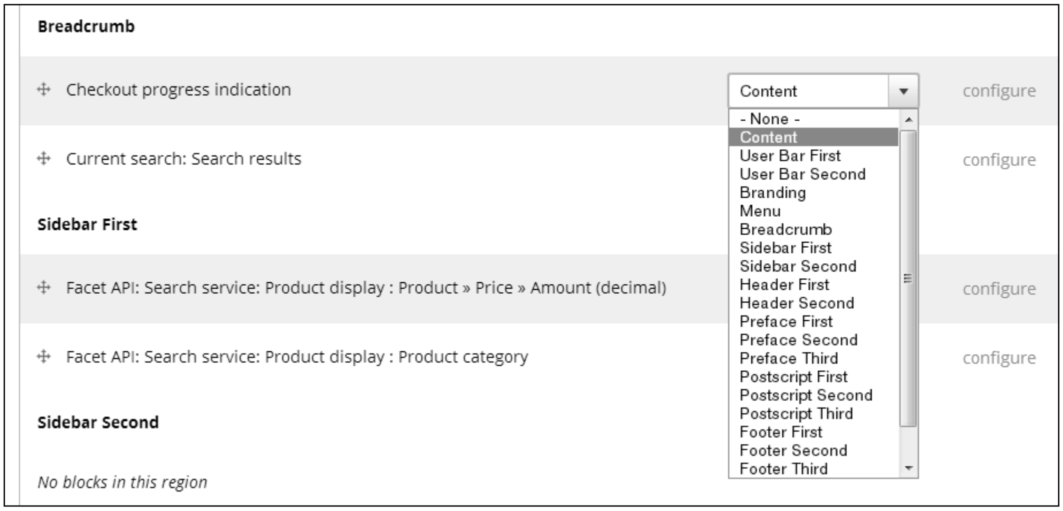
1. Enabling the module will expose a new Drupal block that you can position within your website's theme, so navigate to **Site settings** | **Blocks**.
2. By default, enabling the Commerce Checkout module will position the **Checkout progress indication** block in the **Breadcrumb** region of your Drupal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

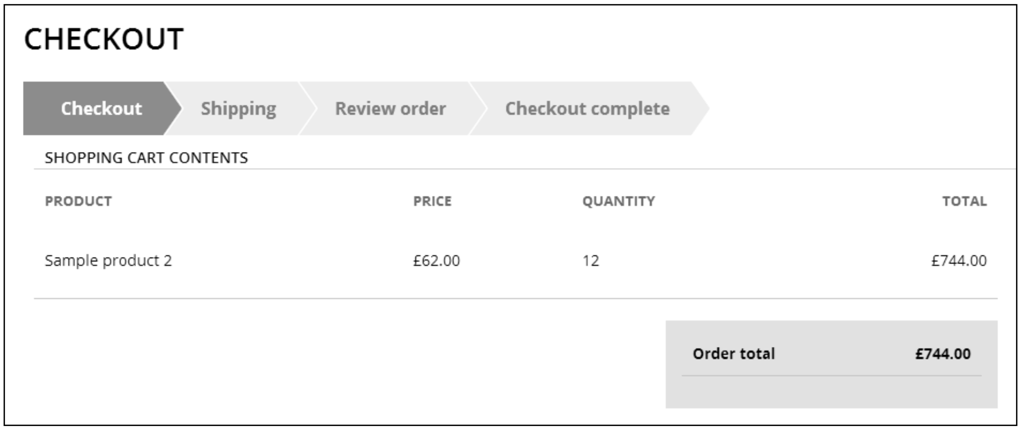
Facet API: Search service: Product display : Product category — configure

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:



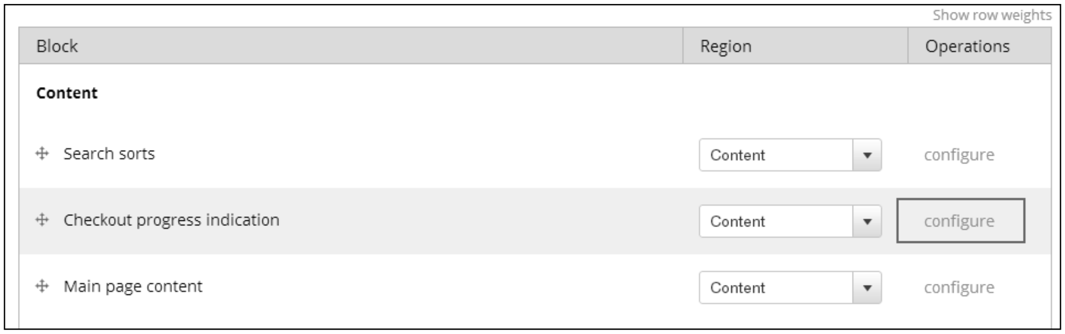
CHECKOUT

SHOPPING CART CONTENTS

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

Order total **£744.00**

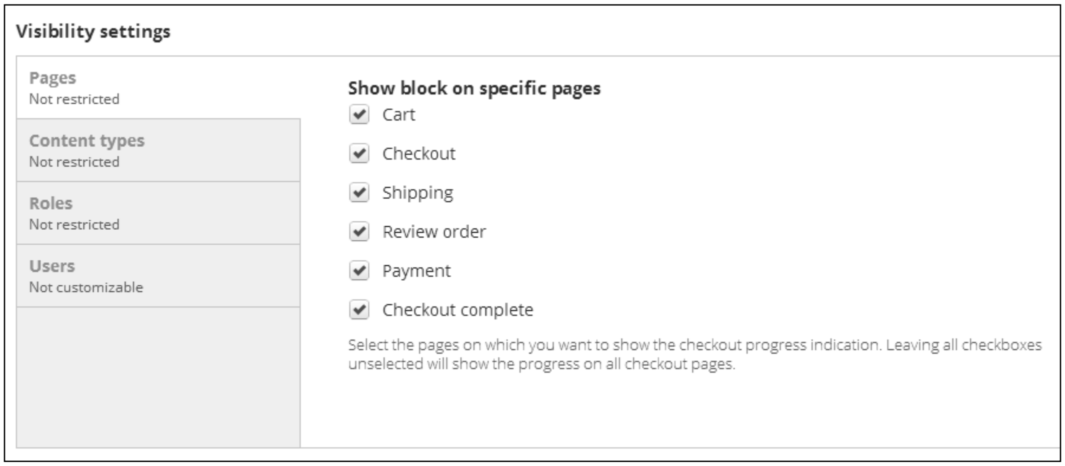
6. At the moment, however, this block only appears on the primary checkout page that you can see in the preceding screenshot. To display the block on all of the cart pages, navigate to **Site settings** | **Blocks** again in your Drupal administration interface.
7. Select the **configure** option to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

Pages
Not restricted

Content types
Not restricted

Roles
Not restricted

Users
Not customizable

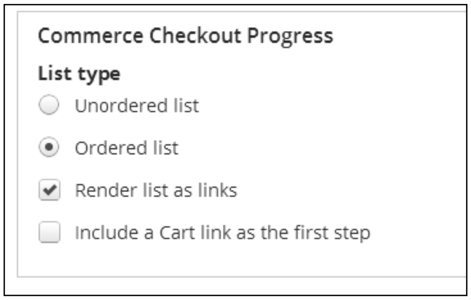
Show block on specific pages

☑ 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.

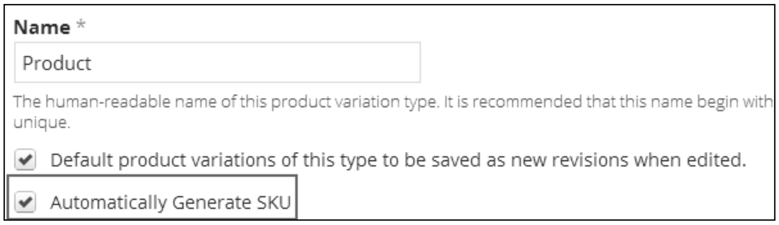
Getting started

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 *

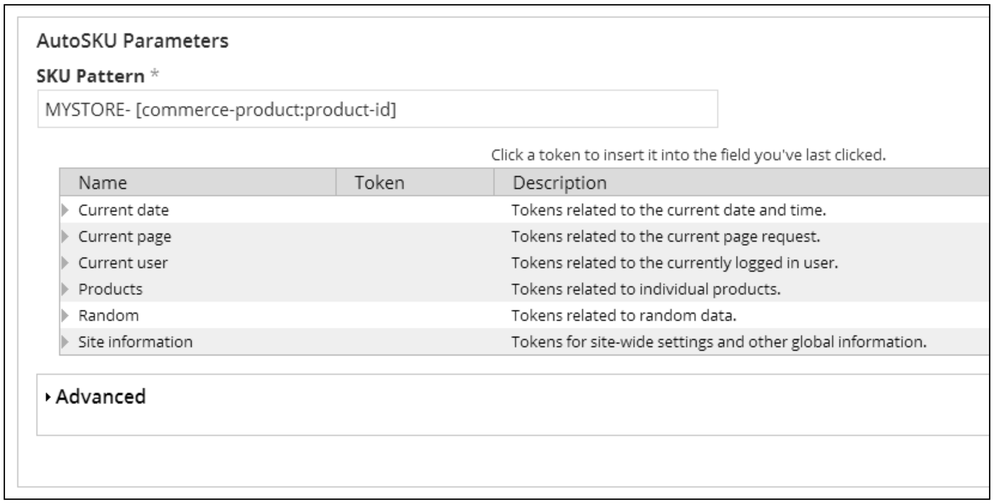
Product

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.

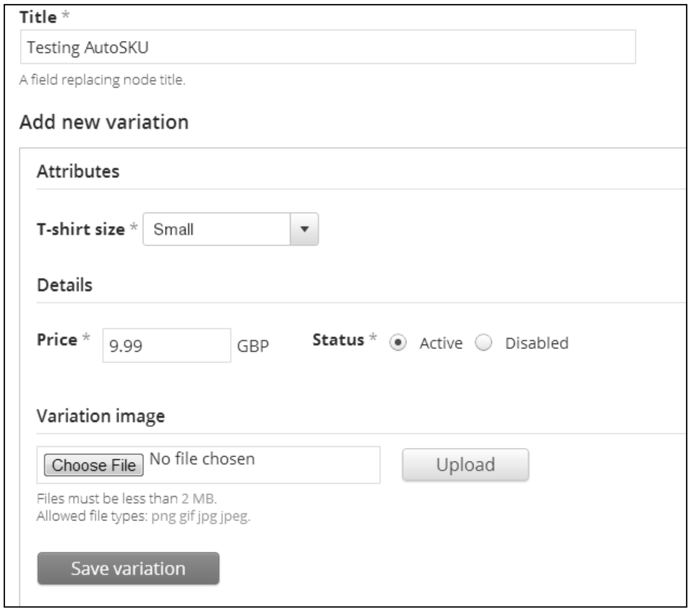
Name	Token	Description
Current date		Tokens related to the current date and time.
Current page		Tokens related to the current page request.
Current user		Tokens related to the currently logged in user.
Products		Tokens related to individual products.
Random		Tokens related to random data.
Site information		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

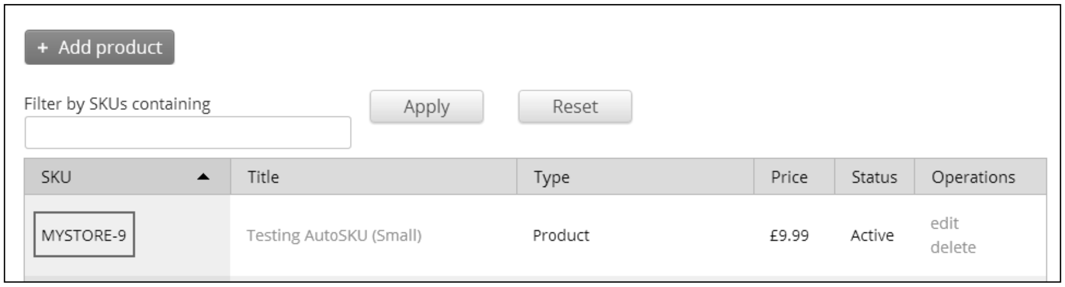
Variation image

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 delete

Where to buy this book

You can buy Building E-commerce Sites with Drupal Commerce Cookbook from the Packt Publishing website: `http://www.packtpub.com/building-ecommerce-sites-with-drupal-commerce-cookbook/book`.

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

www.PacktPub.com

For More Information:
www.packtpub.com/building-ecommerce-sites-with-drupal-commerce-cookbook/book