

Below is just my notes on how options could work in Merchello, and to try and make them as useable as possible for the store owner. I'm no UI/UX designer, so below is just for example purposes.

Admin Section

Base Types

Firstly a user should set up base types. These are simply just a text string, and will be things like 'Size' 'Colour' 'Material' etc... This way, all options are of a type. Which will allow us to pull back only options from a product of a specific type.

This is mainly useful for the new Google Shopping campaigns, where the feed wants cartesian products. But only accepts Size & Colour as the types to do the cartesian on.

Type

Add

Base Types	
Colour	Delete
Material	Delete
Something	Delete
Another	Delete

Base Option Groups & Options

These are just like a database/templates of the available option groups (And options) available to the user. A user will still be able to add more options on the product after adding a specific option group to a product. This section would literally just copy across the values, from the group and options onto the product (Basically creating the new option groups and options on the product).

There is no relationship back from copied group/options to the group/options on the product. This is helpful to users who for example could have 100 T-Shirts all with the same options (Size - Small, Medium, Large) just different descriptions and images. It would save time having to input the same thing over and over again.

The only relationships the options have is what base type they are. All option groups could simply have a `BaseType_Id` field and the options would have a link to the price/currency table (I don't know how Merchello works with this so can't comment on how).

Option groups would also have a View/template associated with them. So developers could display the option groups however they wish (Radio Buttons, Checkboxes etc...)

Name	<input type="text" value="X -Supplier Shirt Colour"/>
Type	<input type="text" value="Colour"/>
Template	<input type="text" value="RadioButtons.cshtml"/>
<input type="checkbox"/> Filterable	

Filterable = Just a switch that developers can use in the front end, to ignore this group from the product filters (If they add them)

Name = This can be for reference too, as they can override Option Group Names on the product to make it customer facing friendly



Name: **Shirt Colour**

Create New Option

Value	Price(s)	
Red	GBP 15.95 USD 26.50	Delete
Blue	GBP 16.50 USD 32.50	Delete

Option For Shirt Colour


Name

Green

Weight Ad

2.8

Associated Image

[Media Picker](#)


Sort Order

1

☐ Pre Selected

Price(s)

Price Group	Price
GBP	
USD	

Weight Adjustment = Some options will have an additional weight, so will impact carriage cost calculations

Associated Image = Very handy feature, could change front end image on selection. But also allows correct image for 'colour' types in the Google Shopping Campaigns. So for a search for Green X, it serves up the image of the Green product.

Pre-Selected = Define which option value is pre-selected when the product details page loads.

Idea? = Maybe add a free textbox (Or a simple key/value list) that users can add custom data attributes in. They can be added to the dropdowns or checkboxes. Just an idea to make it more extensible for developers.

Product Admin

Users are given a dropdown, or select area or something to choose the Option Group Template

Choose Option Group Template X Supplier T-Shirt Colours

This then gives them a very similar UI to above, only they can't change the Type.

Name

Shirt Colours

Template

RadioButtons.cshtml

☐ Filterable

Create New Option

Value	Price(s)	
Red	GBP 15.95 USD 26.50	<div>EditDelete</div>
Blue	GBP 16.50 USD 32.50	<div>EditDelete</div>

The UI for creating and editing Options would be identical as the one in the Admin Section.