

Tutorials

- [How can I add a customer-selected gift wrapping option?](#)

How can I add a customer-selected gift wrapping option?

This tutorial will walk you through adding a "Gift wrap" checkbox to the product page, which when selected will add the product to the cart with the additional £1 charge associated with it being gift wrapped.

First, define some settings to control the behaviour of gift wrapping:

```
\Aero\Common\Facades\Settings::group('gift-wrap', function (\Aero\Common\Settings\SettingGroup $group) {  
    $group->float('price')->default(100)->map(function ($value) {  
        return new \Aero\Cart\CartPrice(request()->store()->currency, $value ?? 0);  
    });  
});
```

```
\Aero\Catalog\Models\Variant::settings(function (\Aero\Common\Settings\ModelSettingGroup $group) {  
    $group->boolean('has_gift_wrap')->default(false);  
});
```

Now, add a macro for `has_gift_wrap` so that we can cast it to js, so that it will be available in Vue:

```
\Aero\Catalog\Models\Variant::macro('getHasGiftWrapAttribute', function () {  
    return $this->setting('has_gift_wrap');  
});  
  
Aero\Catalog\Models\Variant::addCastToJs('has_gift_wrap');
```

We can now display a checkbox on the `product.twig` page when a variant supports it:

```
<div v-if="variant && variant.has_gift_wrap">  
    <input type="checkbox" id="gift-wrap">
```

```
<label for="gift-wrap">Gift Wrap? ({{ setting('gift-wrap.price') }})</label>
</div>
```

And then we listen to an item being added to the cart and set the `has_gift_wrap` bool from this checkbox:

```
{% js %}
<script>
    window.AeroEvents.on('product.add-to-cart', (data) => {
        const el = document.getElementById('gift-wrap');

        data.has_gift_wrap = el?.checked ?? false;

        return data;
    });
</script>
{% endjs %}
```

Finally, we extend the cart item builder to add the gift wrap option when selected:

```
\Aero\Store\Pipelines\CartItemBuilder::extend(function (\Aero\Cart\CartItem $item, \Closure
$next) {
    $hasGiftWrap = $item->model->setting('has_gift_wrap') && request()->get('has_gift_wrap');

    if ($hasGiftWrap) {
        $price = setting('gift-wrap.price');

        $option = \Aero\Cart\CartItemOption::create('Gift Wrap');

        $option->setPriceInc($price->inc()->value);

        $item->addOption($option)->setNotCombinable();
    }

    return $next($item);
});
```

Checkout the [aerocargo/product-options](#) module, for an easier way to do things like this.