

Cart

- [General](#)
 - [How can I add a custom code handler?](#)
- [Cart Items](#)
 - [What is the cart item builder?](#)
- [Tutorials](#)
 - [How can I add a customer-selected gift wrapping option?](#)

General

How can I add a custom code handler?

You can use the `addCodeHandler()` method to add custom behaviour for specific codes when applied to the cart.

For example you can create a discount code that adds a free sample pack to the cart.

First, define some settings to control this behaviour:

```
\Aero\Common\Facades\Settings::group('free-sample', function
(\Aero\Common\Settings\SettingGroup $group) {
    $group->eloquent('discount', Discount::class);
    $group->eloquent('variant', Variant::class)->searchField('sku', 'sku');
});
```

Then add a code handler to the cart that consumes these settings:

```
\Aero\Cart\Cart::addCodeHandler(function (Cart $cart, string $code) {
    $discount = setting('free-sample.discount');

    if (! $discount) {
        return false;
    }

    if ($code !== $discount->code) {
        return false;
    }

    $alreadyApplied = $cart->items()->contains(function (\Aero\Cart\CartItem $item) {
        return $item->meta->get('is_free_sample');
    });

    if ($alreadyApplied) {
        throw new CartModificationException(__('Discount code already applied.'));
    }
});
```

```
}

$cartItem = new CartItem(
    setting('free-sample.variant'),
);

$cartItem->setName('Free Sample Pack')
    ->setImmutable()
    ->setNotCombinable()
    ->setFree()
    ->setMeta('is_free_sample', true);

$cart->add($cartItem);

return true;
});
```

The `aerocargo/automated-add-to-cart` module can be used to automatically add items to the cart based on rules.

Cart Items

What is the cart item builder?

The cart item builder is run whenever an item is added to the cart. It builds the item from the request data.

For example, you can add a gift wrap option to an item when the variant has gift wrapping enabled.

First, define a setting on the variant to control whether gift wrapping is available:

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

The `CartItemBuilder` can then use this setting to add the gift wrap option:

```
\Aero\Store\Pipelines\CartItemBuilder::extend(function (\Aero\Cart\CartItem $item, \Closure $next) {
    if ($item->model->setting('has_gift_wrap')) {
        $item->addOption(
            \Aero\Cart\CartItemOption::create('Gift Wrap')
        );
    }

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

Parameters

Parameter	Type	Description
<code>\$item</code>	<code>\Aero\Cart\CartItem</code>	The cart item being built.
<code>\$next</code>	<code>\Closure</code>	The next step in the cart item builder.

Tutorials

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.