

How do I set a custom price totalizer for shipping rates?

Shipping rates can define a custom price totalizer to adjust the calculated shipping price before it is returned to the customer.

For example, you may want to add a fixed cost for each item in the order.

First, define the cost per item on the shipping method:

```
\Aero\Cart\Models\ShippingMethod::settings('extras', function
(\Aero\Common\Settings\ModelSettingGroup $group) {
    $group->float('cost_per_item');
});
```

The `setPriceTotalizer` method can then add the per-item cost to the shipping rate's calculated price:

```
\Aero\Cart\Models\ShippingRate::setPriceTotalizer(function ($price,
\Aero\Cart\Models\ShippingRate $rate) {
    $costPerItem = $rate->setting('extras.cost_per_item')
        ?? $rate->shippingMethod->setting('extras.cost_per_item');

    if (! $costPerItem) {
        return null;
    }

    $itemAmount = $rate->order->items->sum('quantity');

    return $price + ($itemAmount * $costPerItem * 100);
});
```

The totalizer receives the price and the shipping rate as parameters and should return the total in pence or `null`.

Returning `null` means that the default price totalizer will be used.

Parameters

Parameter	Type	Description
<code>\$price</code>	<code>float</code>	The total price in pence.
<code>\$rate</code>	<code>\Aero\Cart\Models\ShippingRate</code>	The shipping rate being evaluated.

Revision #4
Created 2026-09-16 12:37:36 UTC by Ethan Payne
Updated 2026-10-02 12:59:29 UTC by Ethan Payne