

How do I use a custom total item price totalizer for a shipping rate?

Setting a custom item price totalizer allows you to define the logic used to get the total price of a collection of order items (this value will be used when validating that a shipping rate is available for an order).

To set the logic you need to use the

`**\Aero\Cart\Models\ShippingRate::setItemPriceTotalizer()` method. This method expects you to pass a closure that will return a **number** for the total price or **null** (if null is returned then the default total price logic is used). The closure will be passed the order items and the shipping rate.

In this example code if an order item costs less than 500 then the item will not be used in the calculation for the total price of the order items.

```
<?php

namespace Acme\MyModule;

use Aero\Cart\Models\OrderItem;
use Aero\Cart\Models\ShippingRate;
use Aero\Common\Providers\ModuleServiceProvider;
use Illuminate\Support\Collection;

class ServiceProvider extends ModuleServiceProvider
{
    public function setup()
    {
        ShippingRate::setItemPriceTotalizer(function (Collection $items, ShippingRate $rate) {
            $items = $items->filter(function (OrderItem $item) {
                return ($item->total + $item->total_tax) <= 500;
            });

            return round($items->sum('total') + $items->sum('total_tax'));
        });
    }
}
```

```
    });  
  }  
}
```

Revision #1

Created 2026-09-02 13:29:17 UTC by Michael Price

Updated 2026-09-02 13:29:17 UTC by Michael Price