

mariojgt/gateway Laravel

Introduction

Gateway is a laravel 7 package to handle the payments.

Requirements

- Laravel 7
- PHP ">=7.2.5"
- stripe/stripe-php": "^7.3
- laravelcollective/html": "^6.1
- composer

But don't worry it comes with the package so you don't have to install.

How to install

```
composer require mariojgt/gateway
```

Github

<https://github.com/mariojgt/gateway/blob/master/composer.json>

packagist

<https://packagist.org/packages/mariojgt/gateway>

Step 1 | How to get start

First you need to add this to the `.env` file

```
STRIPE_API=sk_test_yourKey  
STRIPE_SECRET=pk_test_yourSECRET  
STRIPE_CURRENCY='gbp'  
STRIPE_SUCCESS=' '  
STRIPE_ERROR=' '
```

Now you need to publish the package using

```
php artisan vendor:publish --tag gateway
```

Step 2 | Cart Array Requirements

Your cart array will need to follow this structure.

```

[
  "products" => [
    [▼
      "sku_code"      => "ASD12AS3"
      "cart_qty"      => 1//qty customer is buying
      "qty"           => 4
      "product_id"    => 187
      "product"       => "GLOVE"
      "slug"          => "rglove-glove"
      "image"         => "395565.jpg"
      "stock"         => "6"
      "allocated"     => "2"
      "available"     => 4
      "exc_vat"       => "17.50"//the amount will chage the customer
      "inc_vat"       => "21.00"
      "line_exc_vat"  => 17.5
      "line_inc_vat"  => 21.0
      "label"         => "In Stock"
      "button"        => "Add to Cart"
      "class"         => "success"
      "message"       => ""
      "act_qty"       => 4
    ],
    [▼
      "sku_code"      => "F4ADF56DS"
      "cart_qty"      => 1
      "qty"           => 4
      "product_id"    => 187
      "product"       => "ROECKL GLOVE KAILASH 11"
      "slug"          => "roeckl-glove-kailash-11"
      "image"         => "395565.jpg"
      "stock"         => "6"
      "allocated"     => "2"
      "available"     => 4
      "exc_vat"       => "17.50"
      "inc_vat"       => "21.00"
      "line_exc_vat"  => 17.5
      "line_inc_vat"  => 21.0
      "label"         => "In Stock"
      "button"        => "Add to Cart"
      "class"         => "success"
      "message"       => ""
      "act_qty"       => 4
    ]
  ]
]
"cart_qty"          => 3
"product_inc_vat"   => 76.2
"product_exc_vat"   => 63.5
"product_vat"       => 12.7//the tax the user will pay
"notes_order"       => "Your notes"
"postage_code"      => 18
"postage_description" => "UK Mainland Postage"
"postage_exc_vat"   => "0.00"//the psotage that the user will pay
"postage_inc_vat"   => 0.0
"postage_vat"       => 0.0
"cart_exc_vat"      => 63.5
"cart_vat"          => 12.7

```

```
"cart_inc_vat" => 76.2  
]
```

Step 3 | Sending the data to the pay button

In the controller where you will display the pay button add the following.

```
//reference to the cart  
$cart = (array)Session::get('cart');//you cart reference  
//you need this cart in order to pass the right value to the button that  
will  
  
//send the data to the stripe  
  
// show the view  
return view("pages.Checkout.pay", compact('cart'));
```

Step 4 | render the pay button with data

Add the component that generate the button, note that you need to pass the `$cart` form the set before, if your array have the same structure will work just fine if you need to change this file you will find at "vendor/mariojgt/gateway/src/Views/Pages/stripe/components/payButton.blade.php"

```
@component('gateway::Pages.stripe.components.payButton',compact('cart'))  
@endcomponent
```

Step 5 | response

Once you pay there is 2 types of response a success and a error, you can locate them inside the gateway payment that you choose for example if you using stripe you will find at "vendor/mariojgt/gateway/src/Controllers/gateway/StripeController.php"

```
public function success(Request $request)  
{  
    \Stripe\Stripe::setApiKey(config('gateway.stripe_api'));//open the  
    conection  
    //this is the session id which stripe sends back to you  
    $session = \Stripe\Checkout\Session::retrieve(Request('session_id'));  
    //in here we are goin to the this order status using the session id  
    reponse  
    $payment_Status = \Stripe\PaymentIntent::retrieve($session->  
>payment_intent);  
    dd($payment_Status->status);  
}  
  
private function error(Request $request)  
{  
    //you error logic here  
    dd('error');
```

