

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

You have 2 option you can generate the button like this.

A click to pay button like this.

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

Or a page auto redirect like this.

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

At this point you read for a response from the gateway Provider, this package don't save your order it only handle the response, if you want save the order in the database you need to do yourself, the reason why is many project handle orders in different ways but the all need a response from the gateway.

Step 5 | response

Once you paid you have 2 possible response a success and error, the success method will get the info and send the user the the route that you define in you .env file, same for the error,

"vendor/mariojgt/gateway/src/Controllers/gateway/StripeController.php"

```
public function success(Request $request)  
{  
    //using the stripe session id to retrieve the payment info  
    $session = \Stripe\Checkout\Session::retrieve(Request('session_id'));  
    //here we are using the payment intent to get the order status and data
```

```
$payment_status = \Stripe\PaymentIntent::retrieve($session->payment_intent);

$dataReturn = [
    "stripe_session" => $session,
    "stripe_payment_status" => $payment_status,
];

//return the user to the success page define in the config file
return redirect()->route(config('gateway.site_success_redirect'), [
    'data' => json_encode($dataReturn),
]);
}

public function error(Request $request)
{
    //send the user to the error page define in the /env file
    return redirect()->route(config('gateway.site_error_redirect'), []);
}
```