

# Note de stage 2022

⌘UH Group⌘

Semaine n° 4

## ⌘Objectifs

---

**Contexte** : L'application UH Lawyers propose des modules d'applications ainsi que des packages payant pour les avocats. Par le biais de l'application les avocats peuvent facturer leurs clients. Ces clients possèdent un compte sur l'application où ils pourront retrouver leur facture et payer les services qui leur ont été facturés par carte bancaire ou virement SEPA. L'application garde en mémoire des informations concernant la carte bancaire de l'utilisateur : les quatre derniers chiffres, la date d'expiration et le type de carte.

1. Amélioration du système de paiement déjà existant (surtout visuel).
2. Se renseigner sur les virements SEPA et le mettre en place sur notre projet.
3. Enregistrer les informations bancaires d'un utilisateur lorsque celui-ci paye par carte.

⌘Ensemble des liens consultés⌘

## ⌘Sommaire

---

⌘Objectifs .....	1
⌘Réalisations .....	2
Rapport : .....	2
Code : .....	5
Erreurs rencontrées et solutions : .....	16

## ⌘Réalisations

---

### Rapport :

En ce qui concerne le premier objectif j'ai surtout modifié l'apparence de mes pages de paiement, par exemple il était nécessaire d'afficher le montant à payer pour les clients, faire en sorte que l'avocat rentre le montant à payer en euros et non en centimes (gérer dans le back), mettre en évidence les boutons pour cliquer... En somme j'ai fait en sorte que les pages soient plus « User Friendly ».

Pour le deuxième objectif, j'ai d'abord commencé par me renseigner sur ce qu'était ce moyen de paiement d'après la [Banque de France](#): « Le virement SEPA est une opération de paiement en euros. Il est destiné à l'exécution de transferts de fonds entre les comptes de paiement d'un donneur d'ordre et d'un bénéficiaire ouvert sur les livres des banques situées dans l'espace SEPA. ». SEPA signifie « Single Euro Payments Area », est un espace unique de paiement en euro qui permet de simplifier les virements bancaires et les paiements par prélèvement automatique entre les pays membres de la zone SEPA <sup>1</sup>:

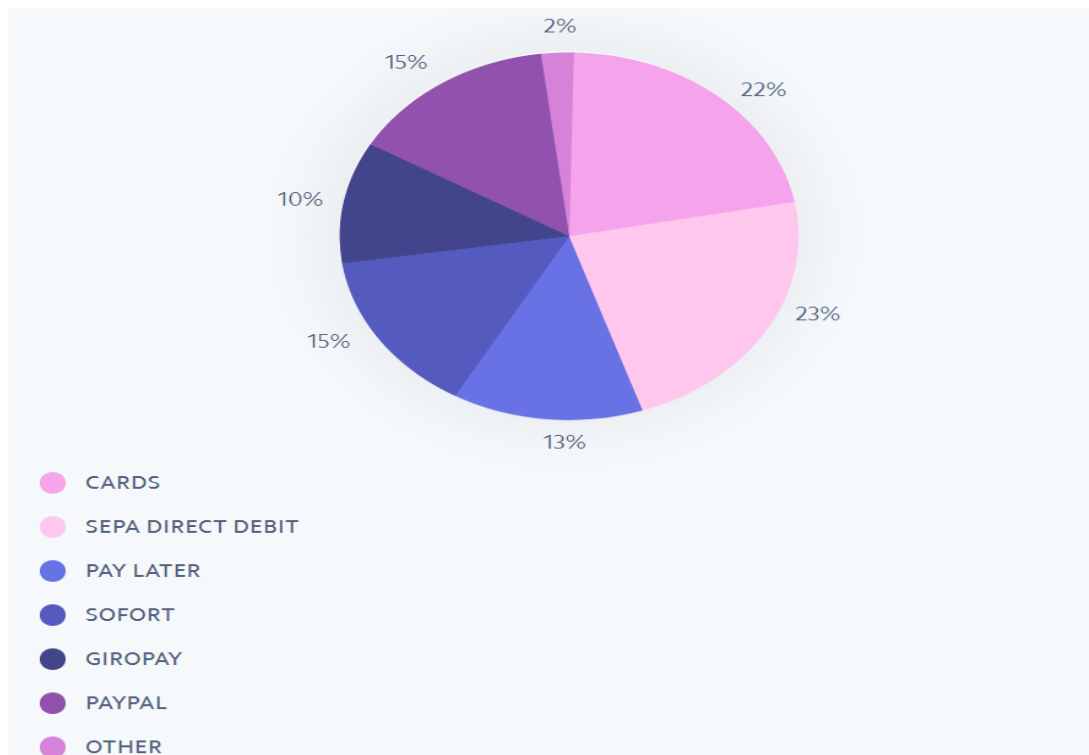
- Les 28 pays membres de l'Union européenne,
- Le Royaume-Uni,
- Les 4 Etats Membres de l'AELE : Islande, Liechtenstein, Norvège, Suisse,
- La Principauté de Monaco,
- La République de Saint-Marin,
- La Principauté d'Andorre,
- La Cité du Vatican.

---

<sup>1</sup> [Info service Assurance Banque Epargne sur les virements SEPA.](#)

L'objectif du SEPA est de réduire la multiplication des moyens de paiement qui complique les transactions entre pays pour les entreprises et les consommateurs<sup>2</sup>. Ainsi les mêmes conditions de tarifs, de délais etc... s'appliquent pour tous les pays et les virements s'effectuent grâce à un code BIC (Bank Identifier Code) et à l'IBAN (International Bank Account Number).

Il est très intéressant pour notre application de proposer ce moyen de paiement car il convient à un grand nombre de pays et on peut constater sur le graphique ci-après proposé par [Stripe](#), que le paiement SEPA représente 23% contre 22% pour les cartes bancaires des moyens de paiement utilisés dans le commerce en ligne en moyenne (cela varie grandement selon le pays ou la région).



En ce qui concerne l'ajout de ce moyen de paiement à l'application je me suis beaucoup inspiré de ce que j'avais fait la semaine précédente et de ce [dépôt GitHub](#). Cette méthode est quasiment identique à celle pour les cartes bancaires. Je n'ai donc pas eu trop de difficulté à l'implémenter dans le projet.

<sup>2</sup> [Ministère de l'économie des finances et de la relance : SEPA.](#)

Ensuite je suis passée au troisième objectif, où mon maître de stage voulait que l'on récupère en base de données les 4 derniers chiffres de la carte bancaire utilisée, sa date d'expiration et le type de carte (Visa, Amex...). Au début je pensais pouvoir simplement récupérer les données bancaires dans les champs générés par Stripe JS en Javascript avec « `document.getElementsByName('name')` » et en récupérant le name en faisant inspecter sur la page (J'utilise le name car Stripe JS ne génère pas d'attribut id pour les balises input concernées). Mais, après plusieurs essais, impossible de récupérer les données de cette manière, j'ai compris bien plus tard que cela était dû à la sécurité intégrée par Stripe (ce qui est au final est assez logique). J'en ai donc parlé avec mon maître de stage une première fois où on s'est mal compris, il m'a partagé une vidéo qui utilisait une autre manière de procéder au paiement. Le code proposé était différent dans les outils (je passais par l'utilisation de `paymentIntent` alors que la vidéo passait par Charge avec un Token), j'ai donc repris entièrement le tutoriel pour avoir les mêmes bases pensant qu'à un moment de la vidéo il serait montré comment récupérer les données de cette manière. Malheureusement, la vidéo ne le montrait pas, de plus le paiement effectué ne passait pas par Stripe Connect mais envoyait le paiement directement à la plateforme. Cependant cela n'était pas totalement inutile dans le sens où les avocats devront faire des paiements directs à la plateforme pour leurs achats de modules et d'abonnements.

Le problème restait le même, entre temps j'avais aussi remarqué la présence dans le `paymentIntent` final de l'attribut `payment_method (id)` qui référait à un objet de type `paymentMethod` où les données bancaires sont stockées. J'en ai donc rediscuté avec mon maître de stage en étant plus claire cette fois-ci et il m'a confirmé qu'il faudrait réussir à récupérer cet objet, qu'il existe normalement des moyens fournis par Stripe pour récupérer les données et il m'a aussi parlé d'utiliser un composant Livewire. Je n'ai pas réussi par les outils Stripe à récupérer les données pour le moment.

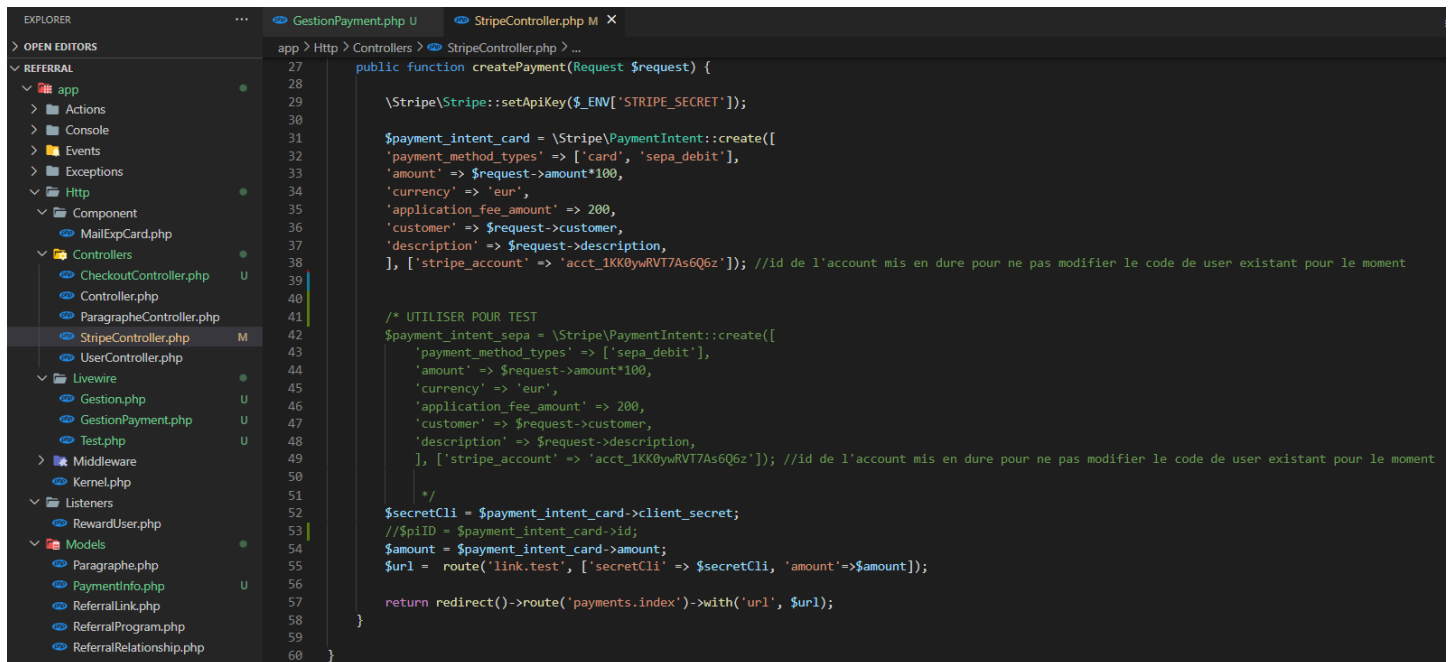
Ainsi, j'ai aussi essayé de créer tout moi-même, avec un composant Livewire et un formulaire non généré par Stripe. Le but étant de recréer les objets nécessaires au paiement avec Stripe (paymentIntent, paymentMethod, Tokens, Customer...). Malheureusement je n'ai pas réussi à faire fonctionner le composant avant la fin de la semaine.

## Code et Visuels:

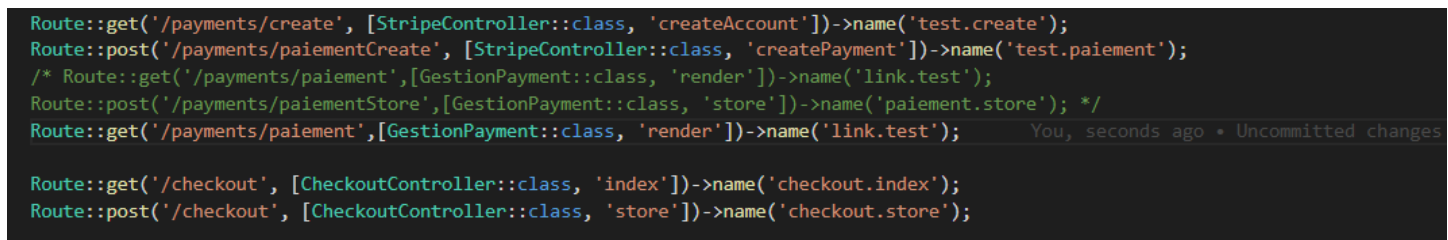
- **ReadMe :**



- **StripeController :**



- **Routes :**



- **GestionPayment :**
  - *Controller :*

```

1  <?php
2
3  namespace App\Http\Livewire;
4
5  use Livewire\Component;
6  use Illuminate\Http\Request;
7  use Cartalyst\Stripe\Laravel\Facades\Stripe;
8
9
10 class GestonPayment extends Component
11 {
12     public $count = 2;
13     public $todayDate;
14
15     public $card_no;
16     public $exp_month;
17     public $exp_year;
18     public $cvc;
19
20     public function __construct() {
21         $this->todayDate = date('m/y');
22     }
23
24     public function render(Request $request)
25     {
26         return view('livewire.gestion-payment', [
27             'secretCli' => $request->secretCli,
28             'amount' => $request->amount,
29             'piID' => $request->piID,
30         ]);
31     }
32

```

```

31 }
32
33
34 public function store(Request $request) {
35     //cardnumber (name input number card)
36     //exp-date (name input expiration date)
37     //type de carte: Visa = 4, Mastercard = 5, American Express = 3
38     echo("<script>console.log('je suis passé par la');</script>");
39     $string = $request->cardnumber;
40     $expDate = $request->expdate;
41     $firstNumber = substr($string,0,1);
42     $lastNumbers = substr($string, strlen($string)-4);
43     if ($firstNumber == 3) {
44         $typeCard = "American Express";
45     } elseif ($firstNumber == 4) {
46         $typeCard = "Visa";
47     } elseif ($firstNumber == 5) {
48         $typeCard = "Mastercard";
49     } else {
50         $typeCard = "Unknown";
51     }
52     $cardInfos = PaymentInfo::firstOrCreate([
53         'user_id' => auth()->user->id,
54         'last_numbers' => $lastNumbers,
55         'exp_date' => $expDate,
56         'card_type' => $typeCard
57     ]);
58 }
59

```

```

EXPLORER
> OPEN EDITORS
  REFERRAL
    > Events
    > Exceptions
    > Http
      Component
        MailExpCard.php
      Controllers
        CheckoutController.php
        Controller.php
        ParagrapheController.php
        StripeController.php
        UserController.php
    > Livewire
      Gestion.php
      GestionPayment.php
      Test.php
    > Middleware
      Kernel.php
    > Listeners
      RewardUser.php
    > Models
      Paragraphe.php
      PaymentInfo.php
      ReferralLink.php
      ReferralProgram.php
      ReferralRelationship.php
      RewardProgram.php
      User.php
    > Providers
    > View\Components
      AppLayout.php
      GuestLayout.php
    > bootstrap

GestionPayment.php U X
app > Http > Livewire > GestionPayment.php > GestionPayment > store
59
60 public function storePayment() {
61
62     $stripe = Stripe::make(env("STRIPE_SECRET"));
63
64     $this->validate([
65         'card_no' => 'required|numeric',
66         'exp_month' => 'required|numeric',
67         'exp_year' => 'required|numeric',
68         'cvc' => 'required|numeric',
69     ]);
70
71     try {
72         $token = $stripe->tokens()->create([
73             'card' => [
74                 'number' => $this->card_no,
75                 'exp_month' => $this->exp_month,
76                 'exp_year' => $this->exp_year,
77                 'cvc' => $this->cvc,
78             ]
79         ]);
80
81         if(!isset($token['id'])) {
82             session()->flash('stripe_error', 'The stripe token was not generated correctly');
83         }
84
85         $customer = $stripe->customer()->create([
86             'name' => 'John Doe',
87             'email' => 'john@doe.fr',
88             'phone' => '0633445566',
89             'address' => [
90                 'line1' => 'fnovnofnvf',
91                 'postal_code' => '83130',
92                 'city' => 'Toulon',
93                 'country' => 'France',
94             ],
95             'source' => $token['id'],
96         ]);
97

```

```

EXPLORER
> OPEN EDITORS
  REFERRAL
    > Events
    > Exceptions
    > Http
      Component
        MailExpCard.php
      Controllers
        CheckoutController.php
        Controller.php
        ParagrapheController.php
        StripeController.php
        UserController.php
    > Livewire
      Gestion.php
      GestionPayment.php
      Test.php
    > Middleware
      Kernel.php
    > Listeners
      RewardUser.php
    > Models
      Paragraphe.php
      PaymentInfo.php
      ReferralLink.php
      ReferralProgram.php
      ReferralRelationship.php
      RewardProgram.php
      User.php
    > Providers
    > View\Components
      AppLayout.php
      GuestLayout.php
    > bootstrap

GestionPayment.php U X
app > Http > Livewire > GestionPayment.php > GestionPayment > store
98
99     $charge = $stripe->charges()->create([
100         'customer' => $customer['id'],
101         'currency' => 'EUR',
102         'amount' => '100',
103         'description' => 'Payment for innfinaerof',
104     ]);
105     return(dd($customer));
106 } catch(Exception $e) {
107     session()->flash('stripe_error', $e->getMessage());
108 }
109
110 }
111
112 public function sendMail() {
113
114     /* $date="01/22";
115     echo("<script>console.log('je suis passé par la');</script>");
116     if ($date == $todayDate) {
117         echo("<script>console.log('et par la');</script>");
118         $to = "mail@test.com";
119         $subject = "Mise à jour de votre carte bancaire";
120         $message = "Votre carte bancaire arrive à expiration ce mois-ci, veuillez la mettre à jour dans votre espace client.";
121         $headers = "From: webmaster@uhgroup.fr" . "\r\n"
122             . "Reply-To: webmaster@uhgroup.fr" . "\r\n"
123             . "X-Mailer: PHP/" . phpversion();
124         mail($to, $subject, $message, $headers);
125         echo("<script>console.log('et le mail est envoyé !');</script>");
126     }
127     */
128 }
129
130 }
131

```

## ○ Vue :

```

EXPLORER
  > OPEN EDITORS
  > REFERRAL
    > js
      > app.js
      > bootstrap.js
    > lang
    > markdown
    > views
      > api
      > auth
      > layouts
        > app.blade.php
        > guest.blade.php
      > livewire
        > gestion-payment.blade.php U
        > gestion.blade.php U
        > test.blade.php U
      > profile
        > delete-user-form.blade.php
        > logout-other-browser-sessions...
        > show.blade.php
        > two-factor-authentication-form...
        > update-password-form.blade.php
        > update-profile-information-for...
        > cancel.blade.php
        > checkout.blade.php U
        > dashboard.blade.php
        > navigation-menu.blade.php
        > payments.blade.php
        > policy.blade.php
        > success.blade.php
        > terms.blade.php
        > test.blade.php
        > users.blade.php
        > vue.blade.php U
        > welcome.blade.php
    > routes

resources > views > livewire > gestion-payment.blade.php > x-app-layout > div.py-12 > div.max-w-7xl.mx-auto.sm:px-6.lg:px-8 > div.bg-white.overflow-hidden
1 <x-app-layout>
2
3 <x-slot name="header">
4 <script src="https://cdn.jsdelivr.net/gh/alpinejs/alpine@v2.8.2/dist/alpine.min.js" defer></script>
5 <h2 class="font-semibold text-xl text-gray-800 leading-tight">
6   {{ __('Païement bancaire') }}
7 </h2>
8 </x-slot>
9 <script src="https://js.stripe.com/v3/"></script>
10 <div class="py-12">
11   <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
12     <div class="bg-white overflow-hidden shadow-xl sm:rounded-lg" style="padding : 25px">
13       <h2 class="font-semibold text-xl text-gray-800 leading-tight"> Pay by card </h2>
14       <br>
15       <form wire:submit.prevent="storePayment">
16         @if(Session::has('stripe_error'))
17           <div class="alert alert-danger role="alert"> {{Session::get('stripe_error')}}</div>
18         @endif
19         <p class="row-in-form">
20           <label for="card-no"> Card Number :</label>
21           <input type="text" name="card-no" value="" placeholder="Card Number" wire:model="card-no">
22           @error('card-no') <span class="text-danger"> {{ $message }} </span> @enderror
23         </p>
24         <p class="row-in-form">
25           <label for="exp-month"> Expiry Month :</label>
26           <input type="text" name="exp-month" value="" placeholder="MM" wire:model="exp-month">
27           @error('exp-month') <span class="text-danger"> {{ $message }} </span> @enderror
28         </p>
29         <p class="row-in-form">
30           <label for="exp-year"> Expiry Year :</label>
31           <input type="text" name="exp-year" value="" placeholder="YY" wire:model="exp-year">
32           @error('exp-year') <span class="text-danger"> {{ $message }} </span> @enderror
33         </p>
34         <p class="row-in-form">
35           <label for="cvc"> CVC :</label>
36           <input type="password" name="cvc" value="" placeholder="CVC" wire:model="cvc">
37           @error('cvc') <span class="text-danger"> {{ $message }} </span> @enderror
38         </p>
39         <button type="submit">
40           Pay {{ $amount / 100 }} €
41         </button>
42       </form>

```

```

EXPLORER
  > OPEN EDITORS
  > REFERRAL
    > js
      > app.js
      > bootstrap.js
    > lang
    > markdown
    > views
      > api
      > auth
      > layouts
        > app.blade.php
        > guest.blade.php
      > livewire
        > gestion-payment.blade.php U
        > gestion.blade.php U
        > test.blade.php U
      > profile
        > delete-user-form.blade.php
        > logout-other-browser-sessions...
        > show.blade.php
        > two-factor-authentication-form...
        > update-password-form.blade.php
        > update-profile-information-for...
        > cancel.blade.php
        > checkout.blade.php U
        > dashboard.blade.php
        > navigation-menu.blade.php
        > payments.blade.php
        > policy.blade.php
        > success.blade.php
        > terms.blade.php
        > test.blade.php

resources > views > livewire > gestion-payment.blade.php > x-app-layout > div.py-12 > div.max-w-7xl.mx-auto.sm:px-6.lg:px-8 > div.bg-white.overflow-hidden.shadow-xl.sm:rounded-lg
41 </button>
42 </form>
43 <!-- Deuxieme version formulaire
44 <form id="payment-form" method="POST" action="{{ route('païement.store') }}">
45   @csrf
46   <div id="card-number"></div>
47   <div id="card-expiry"></div>
48   <div id="card-cvc"></div>
49   <div id="card-errors" role="alert"></div>
50   <x-jet-button id="submit">Pay {{ $amount / 100 }} €</x-jet-button>
51   <input type="hidden" id="secretC11" value="{{ $secretC11 }}" />
52 </form>
53
54 Troisième version début des essais avec Livewire et Alpine
55 <div>
56   {{ $count }}
57   {{ $todayDate }}
58   <!-- Alpine Counter Component -->
59   <div x-data>
60     <h1 x-text="$wire.count"></h1>
61     <x-jet-button x-on:click="$wire.sendMail()">
62       {{ $count }}
63     </x-jet-button>
64     <button x-on:click="$wire.sendMail()">Increment</button>
65   </div>
66 </div>
67 <script>
68   var stripe = Stripe(
69     'pk_test_51KJGNjDf12Zb7T91ZR4ZQsqz4myngwQVvteqFaZ1puaJ8JD6Z7A8aJXmbEnt9UcK0UAE1Uu0XcP30qf1A9ePScC08SQ09a7HO', {
70     stripeAccount: 'acct_1KK0yuvRV7As6Q6z'
71   });
72   var elements = stripe.elements();
73   var style = {
74     base: {
75       color: "#32325d",
76     }
77   };
78

```



```

EXPLORER
> OPEN EDITORS
REFERRAL
  js
    app.js
    bootstrap.js
  lang
  markdown
  views
    api
    auth
    layouts
    app.blade.php
    guest.blade.php
    livewire
      gestion-payment.blade.php U
      gestion.blade.php U
      test.blade.php U
    profile
      delete-user-form.blade.php
      logout-other-browser-sessions-...
      show.blade.php
      two-factor-authentication-form-...
      update-password-form.blade.php
      update-profile-information-for-...
      cancel.blade.php
      checkout.blade.php U
      dashboard.blade.php
      navigation-menu.blade.php
      payments.blade.php
      policy.blade.php
      success.blade.php
      terms.blade.php
      test.blade.php

resources > views > livewire > gestion-payment.blade.php U
78
79
80 // Deuxième version avec creation de plusieurs champs spécifiques
81 // de carte bancaire généré par Stripe JS
82 /* var cardNumber = elements.create("cardNumber", {
83   style: style
84 }); */
85
86 /* cardNumber.mount("#card-number");
87 cardNumber.on("change", function(event) {
88   var test = document.getElementsByName('cardNumber');
89   console.log(test.value);
90   var displayError = document.getElementById('card-errors');
91   if (event.error) {
92     displayError.textContent = event.error.message;
93   } else {
94     displayError.textContent = '';
95   }
96 }); */
97
98 // Première version avec creation du champs unique de carte bancaire
99 // généré par Stripe JS
100
101 var card = elements.create("card", {
102   style: style
103 });
104 card.mount("#card-element");
105
106 card.on('change', function(event) {
107   var displayError = document.getElementById('card-errors');
108   if (event.error) {
109     displayError.textContent = event.error.message;
110   } else {
111     displayError.textContent = '';
112   }
113 });
114

```

```

EXPLORER
> OPEN EDITORS
REFERRAL
  js
    app.js
    bootstrap.js
  lang
  markdown
  views
    api
    auth
    layouts
    app.blade.php
    guest.blade.php
    livewire
      gestion-payment.blade.php U
      gestion.blade.php U
      test.blade.php U
    profile
      delete-user-form.blade.php
      logout-other-browser-sessions-...
      show.blade.php
      two-factor-authentication-form-...
      update-password-form.blade.php
      update-profile-information-for-...
      cancel.blade.php
      checkout.blade.php U
      dashboard.blade.php
      navigation-menu.blade.php
      payments.blade.php
      policy.blade.php
      success.blade.php
      terms.blade.php
      test.blade.php

resources > views > livewire > gestion-payment.blade.php U
115
116 var form = document.getElementById('payment-form');
117 //var clientSecret= document.getElementById('secretCli').value;
118 var clientSecret = < (?)php echo json_encode($secretCli) ?>;
119 form.addEventListener('submit', function(ev) {
120   console.log(clientSecret);
121   ev.preventDefault();
122   //var expDate = document.getElementsByName("exp-date");
123   //var expDate = document.getElementsByClassName("InputElement is-complete Input").value;
124   //var expDate = document.querySelector(".InputElement is-complete Input").value;
125   //alert(expDate[0].value);
126   stripe.confirmCardPayment(clientSecret, {
127     payment_method: {
128       card: card,
129       billing_details: {
130         name: 'Jenny Rosen'
131       }
132     }
133   }).then(function(result) {
134     console.log(result);
135     if (result.error) {
136       console.log(result.error.message);
137       alert(result.error.message);
138     } else {
139       if (result.paymentIntent.status === 'succeeded') {
140         alert('Paiement effectué !');
141       }
142     }
143   });
144 });
145 </script> --}}
146 </div>
147 </div>
148 </div>

```

## Paieement par virement SEPA

```

EXPLORER
> OPEN EDITORS
  REFERRAL
  js
  app.js
  bootstrap.js
  lang
  markdown
  views
  api
  auth
  layouts
  app.blade.php
  guest.blade.php
  livewire
  gestion-payment.blade.php U
  gestion.blade.php U
  test.blade.php U
  profile
  delete-user-form.blade.php
  logout-other-browser-sessions...
  show.blade.php
  two-factor-authentication-form...
  update-password-form.blade.php
  update-profile-information-for...
  cancel.blade.php
  checkout.blade.php U
  dashboard.blade.php
  navigation-menu.blade.php
  payments.blade.php
  policy.blade.php
  success.blade.php
  terms.blade.php
  test.blade.php
  users.blade.php
  viue.blade.php U
  welcome.blade.php
  routes
  api.php

resources > views > livewire > gestion-payment.blade.php > x-app-layout > div.py-12 > div.max-w-7xl.mx-auto.smpx-6.lgpx-8 > div.bg-white.overflow-hidden.shadow-xl.smrrounded-lg

154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196

<div class="py-12">
  <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
    <div class="bg-white overflow-hidden shadow-xl sm:rounded-lg">
      <div class="sr-root">
        <div class="sr-main" style="padding: 25px">
          <h2 class="font-semibold text-xl text-gray-800 leading-tight"> .. or by SEPA</h2>
          <br>
          <form id="payment-form-sepa" class="sr-payment-form">
            <div class="sr-combo-inputs-row">
              <div class="col">
                <label for="name">
                  Name
                </label>
                <input id="name" name="name" placeholder="Jenny Rosen" required />
              </div>
              <div class="col">
                <label for="email">
                  Email Address
                </label>
                <input id="email" name="email" type="email" placeholder="jenny.rosen@example.com" required />
              </div>
            </div>
            <div class="sr-combo-inputs-row">
              <div class="col">
                <label for="iban-element">
                  IBAN
                </label>
                <div id="iban-element">
                </div>
              </div>
            </div>
            <x-jet-button id="confirm-mandate">
              <div disabled class="spinner hidden" id="spinner"></div>
              <span id="button-text">Confirm mandate and initiate debit for {{ $amount / 100 }} €
              <span id="order-amount"></span></span>
            </x-jet-button>
          </form>
        </div>
      </div>
    </div>
  </div>

```

```

EXPLORER
> OPEN EDITORS
  REFERRAL
  js
  app.js
  bootstrap.js
  lang
  markdown
  views
  api
  auth
  layouts
  app.blade.php
  guest.blade.php
  livewire
  gestion-payment.blade.php U
  gestion.blade.php U
  test.blade.php U
  profile
  delete-user-form.blade.php
  logout-other-browser-sessions...
  show.blade.php
  two-factor-authentication-form...
  update-password-form.blade.php
  update-profile-information-for...
  cancel.blade.php
  checkout.blade.php U
  dashboard.blade.php
  navigation-menu.blade.php
  payments.blade.php
  policy.blade.php
  success.blade.php
  terms.blade.php
  test.blade.php
  users.blade.php
  viue.blade.php U
  welcome.blade.php
  routes
  api.php
  channels.php
  console.php
  wash nhrs M
  OUTLINE

resources > views > livewire > gestion-payment.blade.php > x-app-layout > div.py-12 > div.max-w-7xl.mx-auto.smpx-6.lgpx-8 > div.bg-white.overflow-hidden.shadow-xl.smrrounded-lg

196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242

</div>
<script>
  var stripe2 = Stripe(
    'pk_test_51KJGhJdF12ZbJ7912RX4ZQsqz4mynqQYVteqFaZ1puaj8JD6Z7A8ajKmbEnt9UcKOUAE1Uu0XCp3Qqf1A9ePScC085QQ9a7H0', {
      stripeAccount: 'acct_1KK0yWRT7As6Q6z'
    });
  var elements2 = stripe2.elements();
  var style2 = {
    base: {
      color: "#32325d",
      fontFamily: 'apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, Helvetica, Arial, sans-serif',
      fontSmoothing: 'antialiased',
      fontSize: "16px",
      "::placeholder": {
        color: "#aab7c4"
      }
    },
    ".webkit-autofill": {
      color: "#32325d"
    }
  },
  invalid: {
    color: "#fa755a",
    iconColor: "#fa755a",
    ".webkit-autofill": {
      color: "#fa755a"
    }
  }
  });

  var options2 = {
    style: style2,
    supportedCountries: ["SEPA"],
    // If you know the country of the customer, you can optionally pass it to
    // the Element as placeholderCountry. The example IBAN that is being used
    // as placeholder reflects the IBAN format of that country.
    placeholderCountry: "DE"
  };

  var iban = elements2.create("iban", options2);
  iban.mount("#iban-element");

  iban.on("change", function(event) {
    // Handle real-time validation errors from the iban Element.
    if (event.error) {
      alert(event.error.message);
    }
  });

```

```

242    });
243    var form2 = document.getElementById('payment-form-sepa');
244    var clientSecret2 = <?php echo json_encode($secretCli); ?>;
245    form2.addEventListener("submit", function(ev) {
246        ev.preventDefault();
247        stripe2.confirmSepaDebitPayment(clientSecret2, {
248            payment_method: {
249                sepa_debit: iban,
250                billing_details: {
251                    name: document.querySelector('input[name="name"]').value,
252                    email: document.querySelector('input[name="email"]').value
253                }
254            }
255        }).then(function(result) {
256            console.log(result);
257            if (result.error) {
258                //test : DE89370400440532013000
259                //Show error to your customer (for example, insufficient funds)
260                console.log(result.error.message);
261                alert(result.error.message);
262            } else {
263                // The payment has been processed!
264                if (result.paymentIntent.status === 'succeeded') {
265                    // Show a success message to your customer
266                    // There's a risk of the customer closing the window before callback
267                    // execution. Set up a webhook or plugin to listen for the
268                    // payment_intent.succeeded event that handles any business critical
269                    // post-payment actions.
270                    alert('Paieement effectué !');
271                }
272            }
273        });
274    });
275    </script>
276    </div>
277    </div>
278    </div>
279    </x-app-layout>
280

```

- Checkout :
  - Controller :

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Cartalyst\Stripe\Laravel\Facades\Stripe;
7  use Cartalyst\Stripe\Exception\CardErrorException;
8
9  class CheckoutController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16     public function index()
17     {
18         return view('checkout');
19     }
20
21     /**
22      * Show the form for creating a new resource.
23      *
24      * @return \Illuminate\Http\Response
25      */
26     public function create()
27     {
28         //
29     }
30

```

```

31 /**
32  * Store a newly created resource in storage.
33  *
34  * @param \Illuminate\Http\Request $request
35  * @return \Illuminate\Http\Response
36  */
37 public function store(Request $request)
38 {
39     //dd($request->all());
40
41     try {
42         Stripe::setApiKey($_ENV['STRIPE_SECRET']);
43         $charge = Stripe::charges()->create([
44             'amount' => 100,
45             'currency' => 'EUR',
46             'source' => $request->stripeToken,
47             'description' => 'Payment',
48         ]);
49         return redirect()->route('dashboard');
50     } catch (CardErrorException $e) {
51         return back()->withErrors('Error! ' . $e->getMessage());
52     }
53 }
54
55 /**
56  * Display the specified resource.
57  */

```

## ○ Vue :

```

1 <x-app-layout>
2     <x-slot name="header">
3         <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4             {{ __('Your Payment') }}
5         </h2>
6     </x-slot>
7     <script src="https://js.stripe.com/v3/"></script>
8     <div class="py-12">
9         <div class="max-w-7xl mx-auto sm:px-6 lg:px-8">
10             <div class="bg-white overflow-hidden shadow-xl sm:rounded-lg">
11                 <form action="{{ route('checkout.store') }}" method="POST" id="payment-form">
12                     @csrf
13                     <div class="form-group">
14                         <label for="name_on_card">Name on Card</label>
15                         <input type="text" class="form-control" id="name_on_card" name="name_on_card" value="{{ auth()->user()->name }}">
16                     </div>
17                     <div class="form-group">
18                         <label for="card-element">
19                             Credit or debit card
20                         </label>
21                         <div id="card-element">
22                             <!-- a Stripe Element will be inserted here. -->
23                         </div>
24                         <!-- Used to display form errors -->
25                         <div id="card-errors" role="alert"></div>
26                     </div>
27                     <div class="spacer"></div>
28                     <button type="submit" id="payment">Payment</button>
29                 </form>
30             </div>
31         </div>
32     </div>
33 </x-app-layout>
34
35 <script>
36     // Create a Stripe client

```

```

34
35
36 // Create a Stripe client
37 var stripe = Stripe('{{ env('STRIPE_KEY') }}');
38 // Create an instance of Elements
39 var elements = stripe.elements();
40 // Custom styling can be passed to options when creating an Element.
41 var style = {
42   base: {
43     color: '#32325d',
44     lineHeight: '18px',
45     fontFamily: '"Roboto", Helvetica Neue, Helvetica, sans-serif',
46     fontSmoothing: 'antialiased',
47     fontSize: '16px',
48     '::placeholder': {
49       color: '#aab7c4'
50     }
51   },
52   invalid: {
53     color: '#fa755a',
54     iconColor: '#fa755a'
55   }
56 };
57 // Create an instance of the card Element
58 var card = elements.create('card', {
59   style: style,
60   hidePostalCode: true
61 });
62 // Add an instance of the card Element into the 'card-element' <div>
63 card.mount('#card-element');
64 // Handle real-time validation errors from the card Element.
65 card.on('change', function(event) {
66   var displayError = document.getElementById('card-errors');
67   if (event.error) {
68     displayError.textContent = event.error.message;
69   } else {
70     displayError.textContent = '';
71   }
72 });

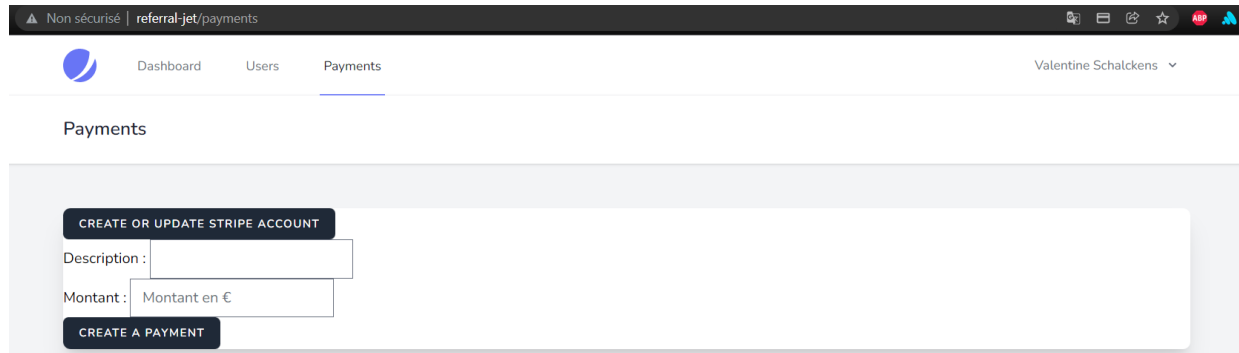
```

```

72
73 // Handle form submission
74 var form = document.getElementById('payment-form');
75 form.addEventListener('submit', function(event) {
76   event.preventDefault();
77   // Disable the submit button to prevent repeated clicks
78   document.getElementById('payment').disabled = true;
79   var options = {
80     name: document.getElementById('name_on_card').value
81   }
82   stripe.createToken(card, options).then(function(result) {
83     if (result.error) {
84       // Inform the user if there was an error
85       var errorElement = document.getElementById('card-errors');
86       errorElement.textContent = result.error.message;
87       // Enable the submit button
88       document.getElementById('payment').disabled = false;
89     } else {
90       stripeTokenHandler(result.token);
91     }
92   });
93 });
94
95 function stripeTokenHandler(token) {
96   // Insert the token ID into the form so it gets submitted to the server
97   var form = document.getElementById('payment-form');
98   var hiddenInput = document.createElement('input');
99   hiddenInput.setAttribute('type', 'hidden');
100   hiddenInput.setAttribute('name', 'stripeToken');
101   hiddenInput.setAttribute('value', token.id);
102   form.appendChild(hiddenInput);
103   // Submit the form
104   form.submit();
105 }
106
107 </script>
108
109 </x-app-layout>

```

- Visuel global :
- Visuel création du paiement :



Non sécurisé | referral-jet/payments

Dashboard Users Payments Valentine Schalckens

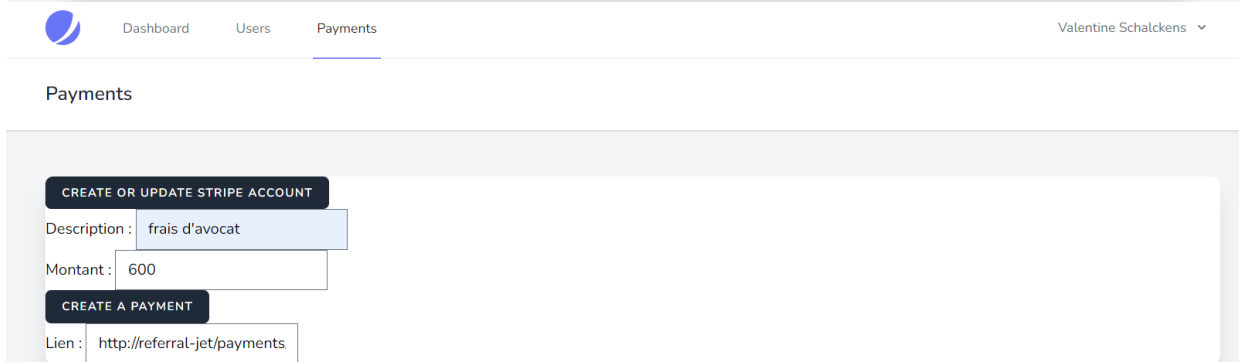
Payments

CREATE OR UPDATE STRIPE ACCOUNT

Description :

Montant :

CREATE A PAYMENT



Dashboard Users Payments Valentine Schalckens

Payments

CREATE OR UPDATE STRIPE ACCOUNT

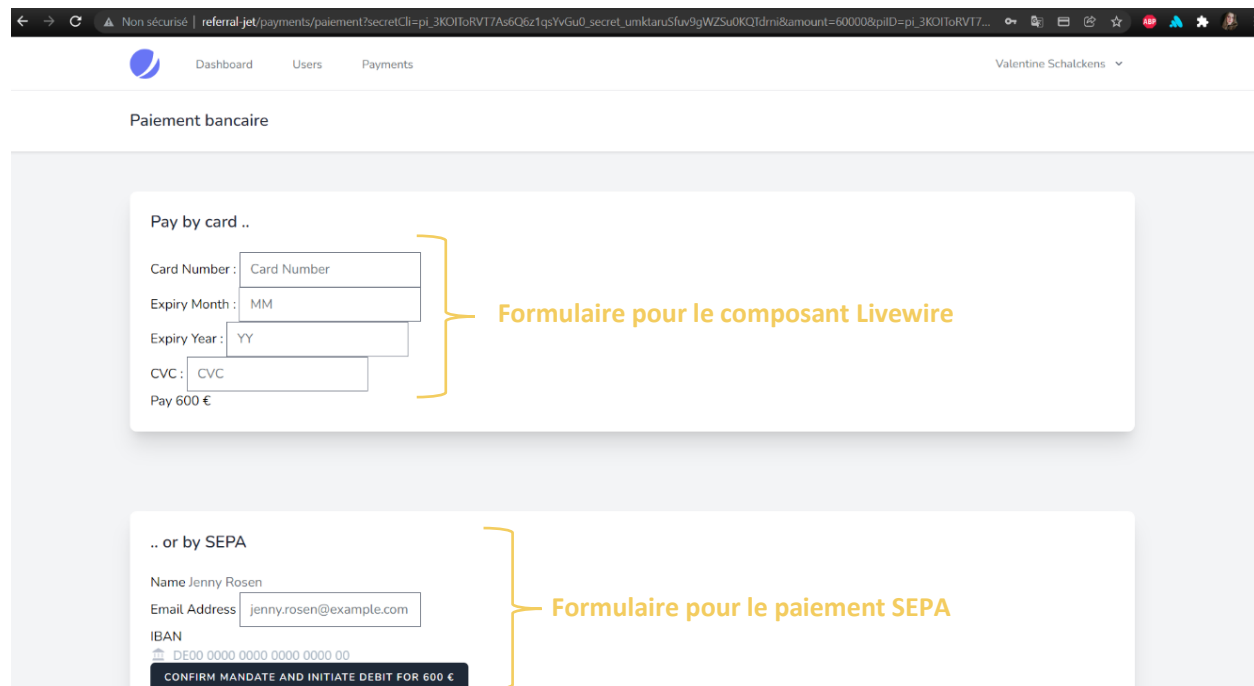
Description :

Montant :

CREATE A PAYMENT

Lien :

- Zone de paiement :



← → ↻ Non sécurisé | referral-jet/payments/paiement?secretCli=pl\_3KO1ToRVIT7As6Q6z1qsYvGu0\_secret\_umktaruSfuv9gWZSu0KQIdmi&amount=60000&piID=pl\_3KO1ToRVIT7...

Dashboard Users Payments Valentine Schalckens

Paieement bancaire

Pay by card ..

Card Number :

Expiry Month :

Expiry Year :

CVC :

Pay 600 €

Formulaire pour le composant Livewire

.. or by SEPA

Name Jenny Rosen

Email Address

IBAN

CONFIRM MANDATE AND INITIATE DEBIT FOR 600 €

Formulaire pour le paiement SEPA

- PaymentInfo : Informations bancaires dans la base de données :
  - Model :

```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class PaymentInfo extends Model
9  {
10     use HasFactory;
11     protected $fillable = ['user_id', 'last_numbers', 'exp_date', 'card_type'];
12 }
13

```

- Migration :

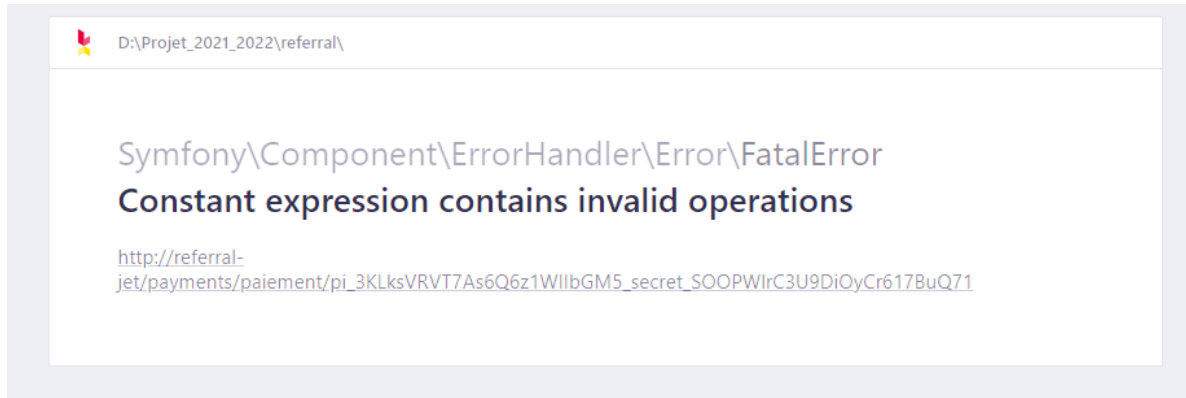
```

1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  class CreatePaymentInfosTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('payment_infos', function (Blueprint $table) {
17             $table->id();
18             $table->integer('user_id');
19             $table->string('last_numbers');
20             $table->string('exp_date');
21             $table->string('card_type');
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      *
29      * @return void
30      */
31     public function down()
32     {
33         Schema::dropIfExists('payment_infos');
34     }
35 }
36

```

## Erreurs rencontrées et solutions :

- « Constant expression contains invalid operations »



As described [here](#)

65

▼

✓

↺

Class member variables are called "properties". You may also see them referred to using other terms such as "attributes" or "fields", but for the purposes of this reference we will use "properties". They are defined by using one of the keywords public, protected, or private, followed by a normal variable declaration. This declaration may include an initialization, but this initialization must be a constant value--that is, it must be able to be evaluated at compile time and must not depend on run-time information in order to be evaluated.

The only way you can make this work is :-

```
<?php
namespace App;

class Amazon
{
    protected $serviceURL;

    public function __construct()
    {
        $this->serviceURL = config('api.amazon.service_url');
    }
}
```

Share Improve this answer Follow

answered Nov 27 '16 at 11:57

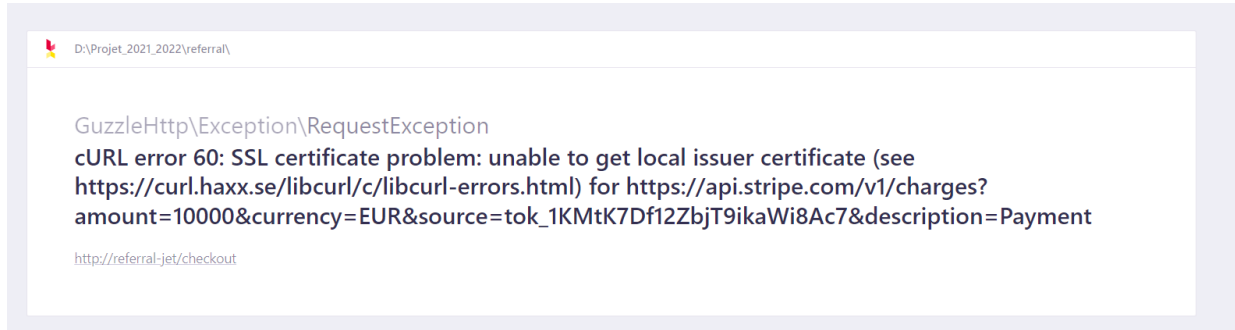
 prateekkathal  
3,202 ● 1 ● 18 ● 37

### Solution

Pour résoudre ce problème il fallait que j'initialise ma variable de classe dans le constructeur au lieu de l'initialiser lors de sa création en dehors d'une fonction.



- « cURL error 60 : SSL certificate problem : unable to get local issuer certificate »



Afin de ne plus obtenir cette exception j'ai dû modifier les deux fichiers `php.ini` (celui de ma version de PHP et celui présent en raccourci dans le menu de WampServer) pour y ajouter le chemin vers le bundle « `cacert.pem` » qui permet de résoudre ce problème de certificat. Puis, j'ai dû décommenter le chargement du module SSL dans le fichier `httpd.conf` d'apache.

```

1927
1928 [curl]
1929 ; A default value for the CURLOPT_CAINFO option. This is required to be an
1930 ; absolute path.
1931 curl.cainfo = "D:/WampServer/bin/php/php7.4.9/extras/ssl/cacert.pem"
1932
1933 [openssl]
1934 ; The location of a Certificate Authority (CA) file on the local filesystem
1935 ; to use when verifying the identity of SSL/TLS peers. Most users should
1936 ; not specify a value for this directive as PHP will attempt to use the
1937 ; OS-managed cert stores in its absence. If specified, this value may still
1938 ; be overridden on a per-stream basis via the "cafile" SSL stream context
1939 ; option.
1940 openssl.cafile = "D:/WampServer/bin/php/php7.4.9/extras/ssl/cacert.pem"
1941
1942 ; If openssl.cafile is not specified or if the CA file is not found, the
1943 ; directory pointed to by openssl.conf is searched for a suitable

```

```

184 #LoadModule socache_redis_module modules/mod_socache_redis.so
185 #LoadModule socache_shmcb_module modules/mod_socache_shmcb.so
186 #LoadModule spelling_module modules/mod_spelling.so
187 LoadModule ssl_module modules/mod_ssl.so
188 #LoadModule status_module modules/mod_status.so
189 #LoadModule substitute_module modules/mod_substitute.so

```