**GESTION DES PAIEMENTS AVEC PAYPAL**

Nous allons découvrir aujourd'hui comment utiliser la [Paypal Standard Checkout](https://developer.paypal.com/docs/checkout/standard/integrate/) qui permet d'intégrer un bouton de paiement PayPal directement sur notre site

Ce tutoriel utilise une librairie dépréciée récemment par Paypal mais continue de fonctionner. Vous pouvez aussi utiliser directement [des appels à l'API avec Curl](https://developer.paypal.com/api/rest/requests/).

**Créer un paiement**

La première étape consiste à générer un bouton de paiement en appelant le [SDK JavaScript](https://developer.paypal.com/sdk/js/). On utilisera un intent=authorize afin de ne pas instancier le paiement directement afin de pouvoir capturer le paiement côté serveur par la suite.

 public function ui(Cart $cart): string

 {

 $clientId = PAYPAL\_ID;

 $order = json\_encode([

 'purchase\_units' => [

 [

 'description' => 'Panier tutoriel grafikart',

 'items' => array\_map(function ($product) {

 return [

 'name' => $product['name'],

 'quantity' => 1,

 'unit\_amount' => [

 'value' => number\_format($product['price'] / 100, 2, '.', ""), // Mes sommes sont en centimes d'euros

 'currency\_code' => 'EUR',

 ]

 ];

 }, $cart->getProducts()),

 'amount' => [

 'currency\_code' => 'EUR',

 'value' => number\_format($cart->getTotal() / 100, 2, '.', ""),

 'breakdown' => [

 'item\_total' => [

 'currency\_code' => 'EUR',

 'value' => number\_format($cart->getTotal() / 100, 2, '.', "")

 ]

 ]

 ]

 ]

 ]

 ]);

 return <<<HTML

 <script src="https://www.paypal.com/sdk/js?client-id={$clientId}&currency=EUR&intent=authorize"></script>

 <div id="paypal-button-container"></div>

 <script>

 paypal.Buttons({

 // Sets up the transaction when a payment button is clicked

 createOrder: (data, actions) => {

 return actions.order.create({$order});

 },

 // Finalize the transaction after payer approval

 onApprove: async (data, actions) => {

 const authorization = await actions.order.authorize()

 const authorizationId = authorization.purchase\_units[0].payments.authorizations[0].id

 await fetch('/paypal.php', {

 method: 'post',

 headers: {

 'content-type': 'application/json'

 },

 body: JSON.stringify({authorizationId})

 })

 alert('Votre paiement a bien été enregistré')

 }

 }).render('#paypal-button-container');

 </script>

 HTML;

 }

Lors du clic sur le bouton une popup s'ouvrira avec l'interface de paiement PayPal (et le site restera en arrière plan). Une fois que l'utilisateur aura rentré ses informations et confirmé le paiement, la popup se fermera et la méthode présente dans le onApprove sera appelée.

**Capturer le paiement**

Une fois que l'utilsateur a autorisé le paiement, on récupèrera un ID d'autorisation que l'on pourra utiliser afin de capturer le paiement en communiquant avec l'API de PayPal. On peut pour cela utiliser le SDK PHP de PayPal paypal/paypal-checkout-sdk.

 public function handle(ServerRequestInterface $request, Cart $cart): void

 {

 if ($this->sandbox) {

 $environment = new \PayPalCheckoutSdk\Core\SandboxEnvironment($this->clientId, $this->clientSecret);

 } else {

 $environment = new \PayPalCheckoutSdk\Core\ProductionEnvironment($this->clientId, $this->clientSecret);

 }

 $client = new \PayPalCheckoutSdk\Core\PayPalHttpClient($environment);

 $authorizationId = $request->getParsedBody()['authorizationId'];

 $request = new \PayPalCheckoutSdk\Payments\AuthorizationsGetRequest($authorizationId);

 $authorizationResponse = $client->execute($request);

 if ($authorizationResponse->result->amount->value !== number\_format($cart->getTotal() / 100, 2, '.', "")) {

 throw new PaymentAmountMissmatchException($amount, $cart->getTotal());

 }

 // On peut récupérer l'Order créé par le bouton

 $orderId = $authorizationResponse->result->supplementary\_data->related\_ids->order\_id;

 // $request = new OrdersGetRequest($orderId);

 // $orderResponse = $client->execute($request);

 // Vérifier si le stock est dispo

 // Verrouiller le produit (retirer du stock pour éviter une commande en parallèle entre temps)

 // Sauvegarder les informations de l'utilisateur

 // On capture l'autorisation

 $request = new AuthorizationsCaptureRequest($authorizationId);

 $response = $client->execute($request);

 if ($response->result->status !== 'COMPLETED') {

 throw new \Exception();

 }

Et voila ! À vous de jouer pour ajouter votre propre logique par dessus ce système.

49 commentaires