tutoriel, BROUILLON

Guide rapide API PayPlug [PHP]

PayPlug est désormais payant par abonnement

Pré-requis

- Installez la librairie :

Téléchargez la librairie sur GitHub :https://github.com/payplug/payplug-php/releases
Ajoutez dans toutes vos pages utilisant l'API :

```
<?php
require_once('PATH_TO_PAYPLUG/payplug_php/lib/init.php');
```

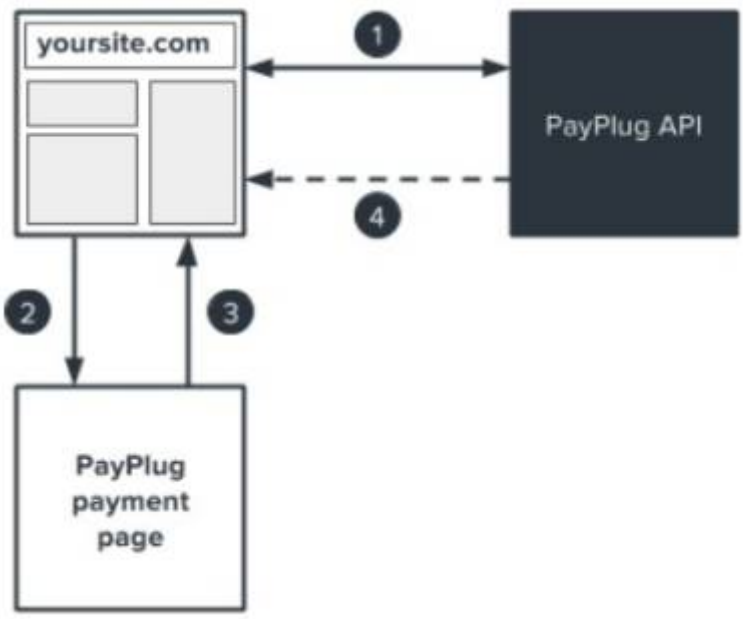
Authentification : l'authentification à l'API s'effectue en utilisant une clé secrète dans toutes les requêtes envoyées.
Pour utiliser la clé, ajoutez-la dans toutes vos pages utilisant l'API :

```
<?php
\Payplug\Payplug::setSecretKey('sk_live_VOTRE_CLE_PRIVEE');
```

Les clés à utiliser avec l'API commencent par **sk_**.
Elles sont disponibles dans le portail PayPlug en cliquant sur **Mon compte** puis sur **Accès API**.
Les modes TEST et LIVE utilisent le même endpoint ; pour changer de mode, il suffit de changer la clé secrète.

Créer un paiement

Fonctionnement de la création d'un paiement



Créez le paiement et récupérez l'URL vers la page de paiement,

Redirigez le client vers la page de paiement,

PayPlug redirige votre client vers la page de retour sur votre site,

PayPlug vous envoie une confirmation au travers de la notification (IPN).

Le paiement

L'exemple suivant permet de créer le paiement puis de rediriger directement votre client vers la page de paiement :

```
<?php
$email = 'john.watson@example.net';
$amount = 33;
$customer_id = '42710';

$payment = \Payplug\Payment::create(array(
  'amount'      => $amount * 100,
  'currency'  => 'EUR',
  'customer'    => array(
    'email'       => Semail
  ).
  'hosted_payment'    => array(
    'return_url' => 'https://example.net/return?id='.$customer_id,
    'cancel_url' => 'https://example.net/cancel?id='.$customer_id
  ),
  'notification_url' =>
'https://example.net/notifications?id='.$customer_id,
  'metadata' => array(
    customer_id' => $customer_id
  )
));
```

```
$payment_url = $payment->hosted_payment->payment_url;
Spayment_id = $payment->id;
header('Location:'.$payment_url);
```

Attention :tous les montants doivent être des entiers positifs en centimes (1€ = 100 centimes).

La confirmation

Option 1 : Les notifications (IPN) :

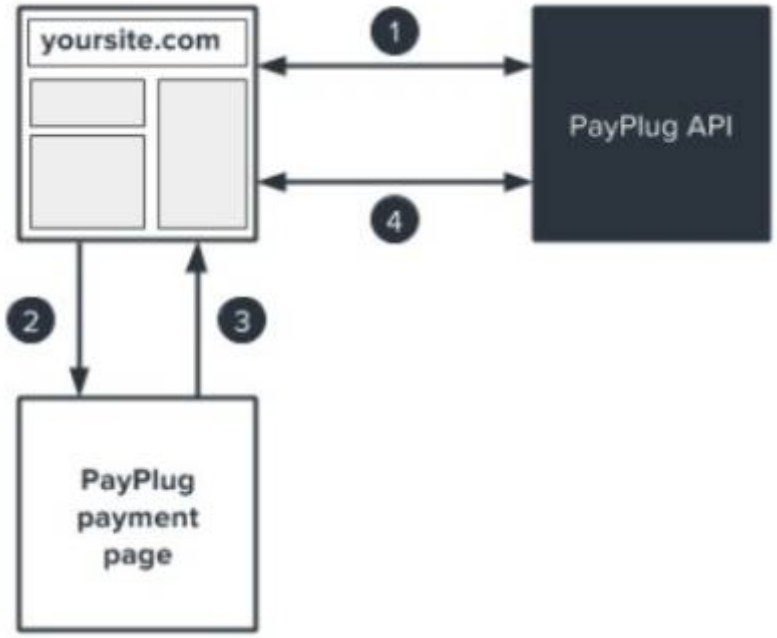
Lors de la création du paiement, vous pouvez spécifier une URL de notification (IPN) : notification_url. Si la transaction est payée ou en échec, une requête POST contenant l'objet correspondant au paiement sera envoyée à votre serveur.

```
<?php
$input = file_get_contents('php://input');
try {
  $resource = \Payplug\Notification::treat($input);
  if($resource instanceof \Payplug\Resource\Payment
    && $resource->is_paid) {
    $payment_id = $resource->id;
    $payment_state = $resource->is_paid;
    $payment_date = $resource->hosted_payment->paid_at;
    $payment_amount = $resource->amount;
    $payment_data = $resource->metadata[customer_id];
  }
}
catch (\Payplug\Exception\PayplugException $exception) {
  écho Sexception;
}
```

L'URL de notification doit être accessible publiquement depuis Internet. Elle ne pourra pas fonctionner si vous êtes en local ou si la page est derrière un firewall ou un proxy.

Option 2 : Récupérer le détail d'un paiement

Si vous ne souhaitez pas utiliser les notifications (IPN), vous pouvez utiliser l'approche suivante :



Les étapes 1 à 3 sont identiques à celles décrites dans le premier schéma Dans l'étape 4, faites un appel vers l'API utilisant l'ID du paiement récupéré lors de sa création :

```
<?php
$payment = \Payplug\Payment::retrieve($payment_id);
```

Gérez les metadatas

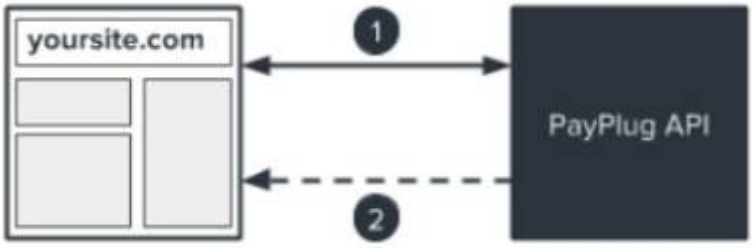
Les metadatas permettent d'ajouter des informations complémentaires lors de la création d'un paiement ou d'un remboursement.

```
{
  "metadata": {
    "transaction_id" : "tsct_201506023456",
    "customer_id": 58790,
    "product_id": "ts_blk_00234",
    "shipping": "The delivery was delayed"
  }
}
```

Vous pouvez ajouter 10 clés. Leur nom ne doit pas dépasser les 20 caractères et les données stockées 500 caractères.

Créer un remboursement

Fonctionnement de la création d'un remboursement



Créez le remboursement en utilisant l'ID du paiement à rembourser,
PayPlug vous envoie une confirmation au travers de la notification (IPN).

Le remboursement

Option 1 : le remboursement complet

Pour créer un remboursement, vous devez disposer de l'ID du paiement que vous souhaitez rembourser.

L'exemple suivant permet de rembourser l'intégralité de la transaction :

```
<?php
$payment_id = 'pay_5iHMDxy4ABR4YBVW4Uscln';
$refund = \Payplug\Refund::create($payment_id);
```

Option 2 : le remboursement partiel

Si vous souhaitez ne pas rembourser l'intégralité de la transaction, l'exemple suivant permet d'effectuer un remboursement partiel :

```
<?php
$payment_id = 'pay_5iHMDxy4ABR4YBVW4Uscln';
$data = array(
  'amount' => 358,
  'metadata' => array(
    'customerjd' => 42710,
    'reason' => 'The delivery was delayed'
  )
);
$refund =\Payplug\Refund::create($payment_id, $data);
```

L'exemple ci-dessus comprend des metadata afin de faciliter le suivi du remboursement.

La confirmation

Option 1 : Les notifications (IPN)

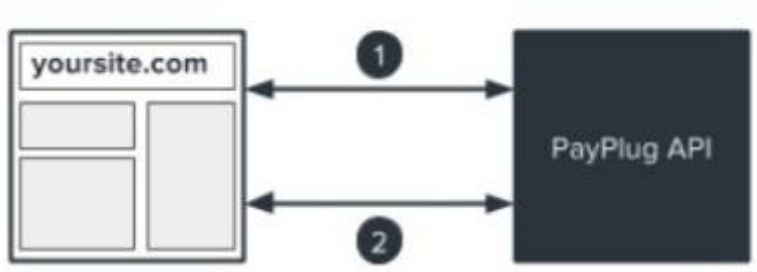
Si lors de la création du paiement vous pouvez spécifier une URL de notification (IPN), lors de la création du remboursement une requête POST contenant l'objet correspondant au remboursement sera envoyée à votre serveur sur cette URL.

```
<?php
$input = file_get_contents('php://input');
try {
  $resource = \Payplug\Notification::treat($input);
  if ($resource->object == "refund")) {
    $refund_id = $resource->id;
    $payment_id = $resource->payment_id;
    $refund_date = $resource->created_at;
    $refund_amount = $resource->amount;
    $refund_data = $resource->metadata[reason];
  }
}
catch (\Payplug\Exception\PayplugException $exception) {
  écho $exception;
}
```

L'URL de notification doit être accessible publiquement depuis Internet. Elle ne pourra pas fonctionner si vous êtes en local ou si la page est derrière un firewall ou un proxy.

Option 2 : Récupérer le détail d'un paiement

Si vous ne souhaitez pas utiliser les notifications (IPN), vous pouvez utiliser l'approche suivante :



L'étape 1 est identique à celle décrite dans le premier schéma. Dans l'étape 2, faites un appel vers l'API utilisant l'ID du paiement et l'ID du remboursement :

```
<?php
$paymentId = 'pay_5iHMDxy4ABR4YBVW4Uscln';
$refundId = 're_3NxGqPfSGMHQgLSZH0Mv3B';
$refund = \Payplug\Refund::retrieve($paymentId, SrefundId);
```

Gérez les metadatas

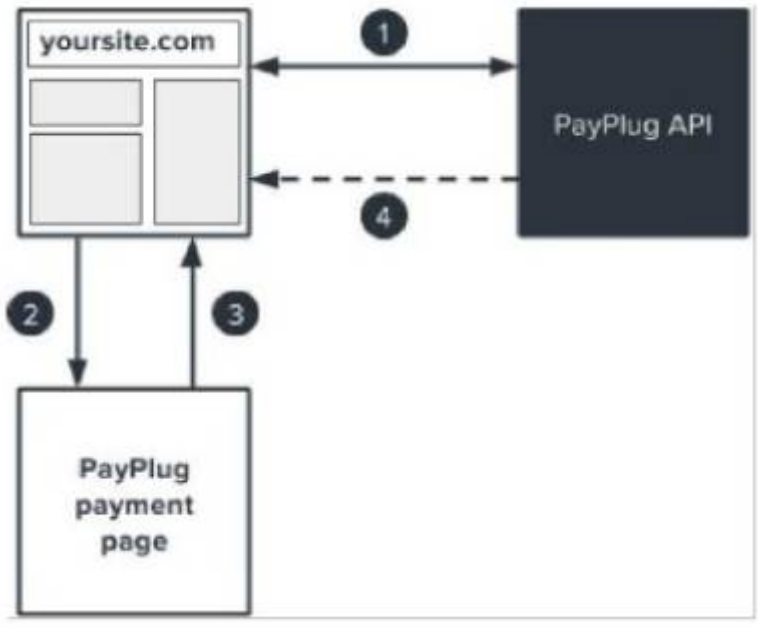
Les metadatas permettent d'ajouter des informations complémentaires lors de la création d'un paiement ou d'un remboursement.

```
{
  "metadata": {
    "transaction_id" : "tsct_201506023456",
    "customer_id": 58790,
    "productjd": "ts_blk_00234",
    "shipping": "The delivery was delayed"
  }
}
```

Vous pouvez ajouter 10 clés. Leur nom ne doit pas dépasser les 20 caractères et les données stockées 500 caractères.

Utiliser la prise d'empreinte de carte

Fonctionnement de la prise d'empreinte



Créez le paiement en utilisant save_card = true et récupérez l'URL vers la page de paiement,

Redirigez le client vers la page de paiement,

PayPlug redirige votre client vers la page de retour sur votre site,

PayPlug vous envoie une confirmation contenant l'empreinte de la carte au travers de la notification (IPN).

Le paiement et la prise d'empreinte

L'exemple suivant permet de créer le paiement qui va permettre de procéder à la prise d'empreinte puis de rediriger directement votre client vers la page de paiement :

```
<?php
$email = 'john.watson@example.net';
$amount = 33;
$customer_id = '42710';

Spayment = \Payplug\Payment::create(array(
  'amount'    => $amount * 100,
  'currency'  => 'EUR',
  'save_card' => true,
  'customer'  => array(
    'email'   => $email
  ).
  'hosted_payment' =>array(
    'return_url' => 'https://example.net/success?id='.$customer_id,
    'canceLurl' => 'https://example.net/cancel?id='.$customer_id
  ).
  'notification_url' =>
'https://example.net/notifications?id='.Scustomer_id,
  'metadata' => array(
    'customer_id' => Scustomer_id
  )
))
```

Tous les montants doivent être des entiers positifs en centime (1€ = 100 centimes).

La confirmation

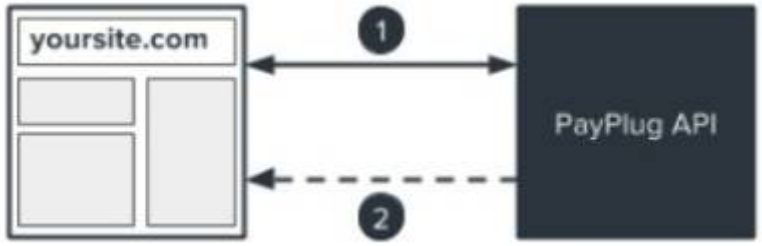
Lors de la création du paiement, vous pouvez spécifier une URL de notification (IPN) : notification_url. Si la transaction est payée ou en échec, une requête POST contenant l'objet correspondant au paiement sera envoyée à votre serveur.

```
<?php
$input = file_get_contents('php://input');
try {
  $resource = \Payplug\Notification::treat($input);
  if ($resource instanceof \Payplug\Resource\Payment && $resource->is_paid)
{
    $payment_id = $resource->id;
    $payment_state = $resource->is_paid,true;
    $payment_date = $resource->hosted_payment->paid_at;
    $payment_amount = $resource->amount;
    if ($resource->save_card == true) {
      $card_id = $resource->card->id;
      $card_exp_month = $resource->card->exp_month;
```

```
      $card_exp_year = $resource->card->exp_year;
    }
    $payment_data = $resource->metadata[customer_id];
  }
}
catch (\Payplug\Exception\PayplugException $exception) {
  écho $exception;
}
```

Dans l'objet contenu dans la notification (IPN) vous pouvez maintenant récupérer l'id de la carte, c'est ce dernier que vous devrez utiliser pour les prochains paiements. Nous vous recommandons de récupérer également la date d'expiration de la carte afin de prévoir de contrôler la validité de la carte avant de créer la prochaine transaction.

Le paiement avec un ID de carte



Créez le paiement en utilisant payment_method et l'ID de la carte prélablement enregistrée,

PayPlug vous envoie une confirmation au travers de la notification (IPN).

Créez un paiement avec l'ID d'une carte

Maintenant que vous disposez de HD d'une carte, vous pouvez la débiter à nouveau sans avoir besoin de refaire passer votre client par une page de paiement :

```
<?php
$email = 'john.watson@example.net';
$amount = 33;
$customer_id = '42710';
$card_id = 'card_e7133426b8de947b37161dfba1897dd1 ';

Spayment = \Payplug\Payment::create(array(
  'amount'         => $amount * 100,
  'currency'       => 'EUR',
  'payment_method' => $card_id,
  'customer'       => array(
  'email'          => $email
  ),
  'notification_url' =>
'https://example.net/notifications?id='.$customer_id,
```

```
  'metadata'  =>array(
    'customer_id'  => $customer_id
));
```

Suite au paiement votre client recevra une confirmation par e-mail.

Les usages

La prise d'empreinte d'une carte permet de mettre en place les usages suivants :

Paiement en 1 clic

Permettez à vos acheteurs de payer sans avoir besoin de saisir à chaque fois ses coordonnées bancaires.

Paiement récurrent

Proposez la possibilité de souscrire à des abonnements récurrents sur votre site web.

Paiement en plusieurs fois

Offrez à vos clients de payer un montant en 3 ou 4 règlements sans frais (moins de 90 jours).

Pour le paiement récurrent et le paiement en plusieurs fois, vous devrez gérer les échéanciers au niveau de votre service

Conclusion

Problèmes connus

Voir aussi

- **(fr)** Lien externe

Contributeurs principaux : jamaique.