

[Accueil](#) > [Cours](#) > [Wilo Ahadi](#)

› Intégrer le paiement PayPal sur son site avec la SDK JavaScript PayPal

# Intégrer le paiement PayPal sur son site avec la SDK JavaScript PayPal

Mis à jour il y a 1 an

Ce tutoriel présente la procédure pour mettre en place le bouton de paiement PayPal Express Checkout sur un site web avec la librairie JavaScript PayPal pour accepter les achats en ligne.



Domaine

Développement web

Technologies

HTML, JavaScript

[Développeur Fullstack PHP H/F](#)

[Développeur Fullstack Java H/F](#)

## ≡ Sommaire

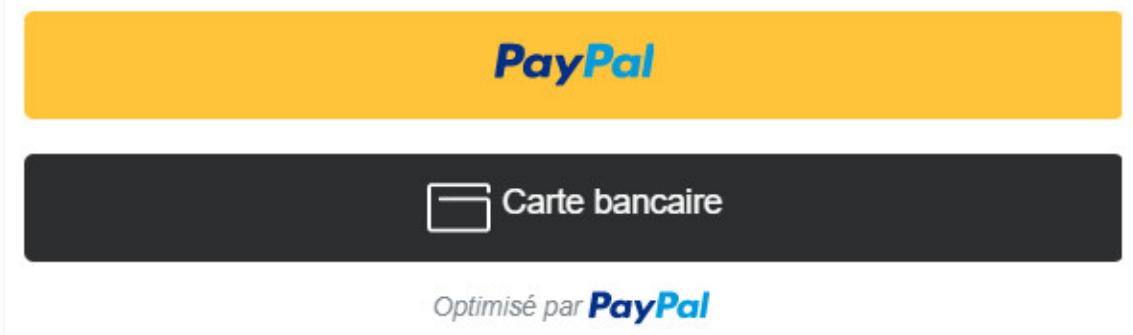
- [Introduction](#)
- [Créer des comptes PayPal sandbox](#)
- [Mise en place du système de paiement PayPal en sandbox](#)
- [Passer du mode sandbox en production \(live mode\)](#)

données au lieu des fichiers. → [En savoir plus](#)

## # Introduction

**PayPal** est un système de paiement en ligne très répandu sur le web qui permet de payer des achats, recevoir des paiements et envoyer de l'argent.

Nous expliquons dans ce guide comment l'intégrer dans un site web en implémentant la **SDK JavaScript** pour accepter (recevoir) les paiements en ligne en présentant les boutons PayPal suivants :



Nous allons commencer par voir comment créer des comptes PayPal que nous allons utiliser pour faire des tests, nous passerons ensuite à la mise en place du système de paiement puis nous verrons comment passer de l'environnement de test (sandbox) en production (live mode)

## # Créez des comptes PayPal sandbox

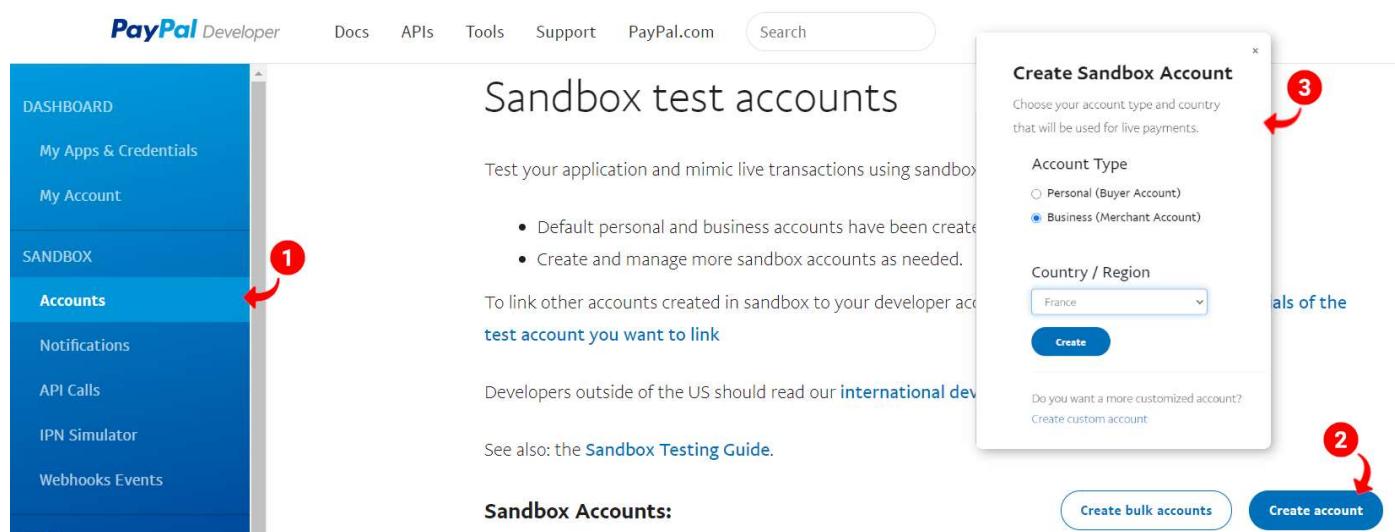
Le terme « sandbox », dans un sens plus large en informatique, fait référence à un environnement de test pour les logiciels ou sites web.

Un **compte PayPal sandbox** permet de tester l'API PayPal en phase de développement de votre application. Vous pouvez l'utiliser pour simuler des transactions entre un **compte Business** (Marchant) et un **compte Personnel** (Acheteur) sans dépenser votre l'argent, tester les fonctionnalités de PayPal, ...

Voici la procédure pour créer un compte PayPal Business et Personnel pour le sandbox :

1. Connectez-vous sur la plateforme [PayPal Developer](#) avec votre compte PayPal. Si vous n'avez pas de compte, vous pouvez l'obtenir sur [la page d'inscription](#).
2. Une fois dans votre compte, allez au menu [Sandbox > Accounts](#) puis [Create account](#) pour créer un compte **Business** et **Personnel** (Deux comptes). Vous pouvez voir et éditer les informations de ces comptes (nom, identifiant email, compte bancaire, carte de crédit, balance, ...) une fois créés.

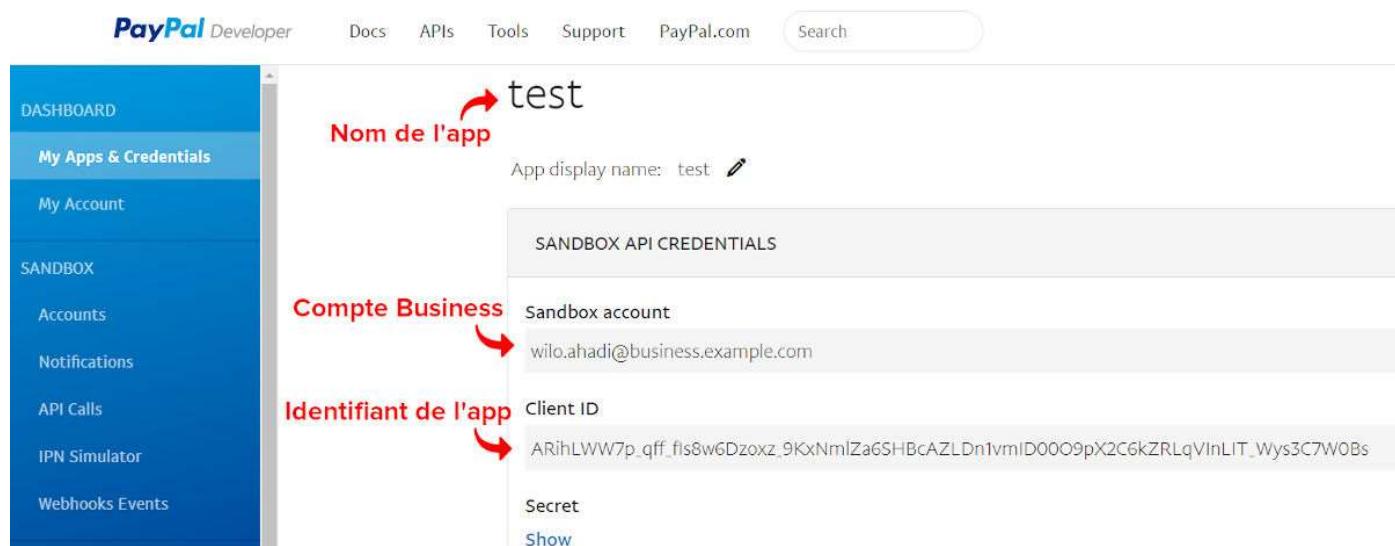
Le capture ci-dessous illustre cette la procédure :



The screenshot shows the PayPal Developer Sandbox test accounts page. The left sidebar has a 'Sandbox' section with an 'Accounts' link, which is highlighted with a red arrow labeled '1'. The main content area is titled 'Sandbox test accounts' and contains instructions for testing applications using sandbox accounts. At the bottom right of the main content, there are two buttons: 'Create bulk accounts' and a larger blue button labeled 'Create account', which is highlighted with a red arrow labeled '2'. A red arrow labeled '3' points to a modal window titled 'Create Sandbox Account' that is overlaid on the page. The modal window asks for account type (Business selected), country (France), and a 'Create' button.

Hormis les comptes sandbox, nous avons besoin d'une **application PayPal** qui représente votre site web. Cette application va être associée au compte Business (compte Marchant).

Pour créer une application PayPal, il faut se rendre au menu **Sandbox > My Apps & Credentials** puis **Create App**, renseigner le nom de l'application et choisir le compte marchant à utiliser. Les identifiants (Client ID, Secret) sont générés automatiquement :



The screenshot shows the PayPal Developer My Apps & Credentials page. The left sidebar has a 'Sandbox' section with an 'Accounts' link. The main content area shows a list of applications. One application is highlighted with a red arrow labeled 'Nom de l'app' pointing to the 'test' name. Another application is highlighted with a red arrow labeled 'Compte Business' pointing to the 'Sandbox account' section, which shows an email address: 'wilo.ahadi@business.example.com'. A third application is highlighted with a red arrow labeled 'Identifiant de l'app' pointing to the 'Client ID' section, which shows a long alphanumeric string: 'ARihLWW7p\_qff\_fls8w6Dzoxz\_9KxNmIZa6SHBcAZLDn1vmID0009pX2C6kZRLqVInLIT\_Wys3C7W0Bs'. Below the Client ID is a 'Secret' section with a 'Show' link.

## # Mise en place du système de paiement PayPal en sandbox

Maintenant que nous avons nos deux comptes pour faire des tests avec l'API PayPal et l'application associée au compte marchant, mettons en place le paiement PayPal. Nous allons procéder de la manière suivante :

1. Importer la librairie PayPal JavaScript dans la page HTML
2. Afficher le bouton de paiement PayPal
3. Configurer une transaction (produits à acheter, total, devise, ...)
4. Récupérer une transaction lorsque l'utilisateur paye sa commande
5. Gérer une transaction annulée

## 1. Importer la SDK JavaScript de PayPal dans une page HTML

Pour intégrer la SDK JavaScript PayPal dans une page web, on ajoute le script suivant juste avant la fermeture de la balise `<body>` :

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>Acheter via PayPal</title>
</head>
<body>

<!-- Importation de la SDK JavaScript PayPal --&gt;
&lt;script src="https://www.paypal.com/sdk/js?client-id=SANDBOX_CLIENT_ID"&gt;&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>

```

Il faut remplacer `SANDBOX_CLIENT_ID` par l'identifiant (Client ID) de votre application PayPal. Vous pouvez personnaliser la SDK en vous référant à [la documentation](#) où sont décrit les autres paramètres valides avec les valeurs acceptées.

## 2. Afficher les boutons de paiement

La SDK JavaScript importée, présentons les boutons de paiement PayPal. Pour ce faire, nous allons :

1. Créer un élément HTML d'identifiant « `paypal-boutons` » ( `<div id="paypal-boutons">` `</div>` par exemple) qui va servir de conteneur pour les boutons PayPal : **Payer avec son solde PayPal et Payer avec une carte de crédit** ; Le placer dans la balise `<body>` à l'endroit où on désire afficher les boutons de paiement. Généralement on place ces boutons à côté (ou en dessous) de la description d'un produit ou d'un panier e-commerce (Voir le cours [Laravel – Créer un système de panier e-commerce](#)).
2. Appeler la fonction `paypal.buttons().render(#id_element)` pour insérer les boutons PayPal dans l'élément `#id_element` ( `#paypal-boutons` pour ce cas)

Le code source devient :

```

<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <title>Acheter via PayPal</title>
</head>
<body>

<!-- Le conteneur des boutons PayPal --&gt;
&lt;div id="paypal-boutons"&gt;&lt;/div&gt;

<!-- 1. Importation de la SDK JavaScript PayPal --&gt;
&lt;script src="https://www.paypal.com/sdk/js?client-id=ARihLWW7p_qff_fIs8w6Dzoxz_9KxNm1Z
&lt;script&gt;
  // 2. Afficher le bouton PayPal
  paypal.Buttons().render("#paypal-boutons");
&lt;/script&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>

```

### 3. Configurer une transaction

A partir d'ici nous allons compléter la fonction `paypal.Buttons($actions).render("#paypal-boutons")` où `$actions` est l'objet dans lequel on gère la configuration, la capture et l'annulation d'une transaction.

Pour configurer une transaction lorsque l'utilisateur clique sur le bouton de paiement PayPal, on appelle la fonction `createOrder($data, $actions)` dans laquelle on décrit les items à payer, le total, la devise, ... par la fonction `$actions.order.create($transaction)`. Référez-vous à [la documentation Create Order](#) pour les paramètres valides d'une transaction `$transaction`.

Le code source devient :

```

paypal.Buttons({
  // Configurer la transaction
  createOrder : function (data, actions) {

    // Les produits à payer avec leurs details
    var produits = [
      {
        name : "Produit 1",
        description : "Description du produit 1",
        quantity : 1,

```

```

        unit_amount : { value : 9.9, currency_code : "USD" }
    },
    {
        name : "Produit 2",
        description : "Description du produit 2",
        quantity : 1,
        unit_amount : { value : 8.0, currency_code : "USD" }
    }
];
// Le total des produits
var total_amount = produits.reduce(function (total, product) {
    return total + product.unit_amount.value * product.quantity;
}, 0);

// La transaction
return actions.order.create({
    purchase_units : [
        {
            items : produits,
            amount : {
                value : total_amount,
                currency_code : "USD",
                breakdown : {
                    item_total : { value : total_amount, currency_code : "USD" }
                }
            }
        }
    ]
});
});

}).render("#paypal-boutons");

```

Le clic sur le **bouton Payer** de PayPal ouvre ainsi une boîte modale où l'utilisateur est invité à se connecter à son compte, voir les détails de la transaction pour approuver le paiement :



17,90 USD

Bonjour Wilondja

Adresse de livraison

[Modifier](#)

Wilondja Ahadi

Av. de la Pelouse, Alsace, 75002 Paris

Payer avec



Solde PayPal

16,29 EUR



[Définir comme source d'approvisionnement préférée](#)

Taux de conversion de PayPal : 1 EUR = 1,09885 USD



Visa

VISA

\*\*\*\*3722



Rabobank Nederland

Courant \*\*\*\*7392

[+ Enregistrer une carte bancaire](#)

**Payer**

Utilisez le compte sandbox Personnel pour payer et dépenser autant que vous voulez.

## 4. Capturer une transaction

Lorsque l'utilisateur paye, son compte est débité et le compte Business reçoit une notification pour confirmer la transaction. Cette action se gère dans la fonction `onApprove($data, $actions)` où on capture les fonds de la transaction par la fonction `$actions.order.capture()` :

```
paypal.Buttons({  
  
  // Capturer la transaction après l'approbation de l'utilisateur  
  onApprove : function (data, actions) {  
    return actions.order.capture().then(function(details) {  
      // Gérer la transaction capturée  
    })  
  }  
})  
;
```

```

    // Afficher les détails de la transaction dans la console
    console.log(details);

    // On affiche un message de succès
    alert(details.payer.name.given_name + ' ' + details.payer.name.surname + ' ' +
        });

}

}).render("#paypal-boutons");

```

Pour la liste de tous les détails d'une transaction capturée par la fonction `$actions.order.capture()` , référez-vous à [cette section de la documentation](#).

## 5. Annuler une transaction

Lorsque l'utilisateur décide de ne plus procéder au paiement ou annule la transaction, on le gère par la fonction `onCancel($data)` . Affichons le message « Transaction annulée » :

```

paypal.Buttons({
    // Annuler la transaction
    onCancel : function (data) {
        alert("Transaction annulée !");
    }
}).render("#paypal-boutons");

```

Rassemblons tout, voici donc **le code source complet** de l'implémentation de la **SDK JavaScript PayPal** :

```

<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <title>Acheter via PayPal</title>
</head>
<body>

<h1>Votre panier :</h1>

<!-- Le produit 1 -->
<h2>Le produit 1 - 9.9 USD</h2>

```

```
<p>Description du produit 1</p>
<hr>
<!-- Le produit 2 -->
<h2>Le produit 2 - 8.0 USD</h2>
<p>Description du produit 2</p>
<hr>

<!-- Le conteneur des boutons PayPal -->
<div id="paypal-boutons"></div>

<!-- 1. Importation de la SDK JavaScript PayPal -->
<script src="https://www.paypal.com/sdk/js?client-id=ARihLWW7p_qff_fIs8w6Dzoxz_9KxNmIz
<script>
    // 2. Afficher le bouton PayPal
    paypal.Buttons({
        // 3. Configurer la transaction
        createOrder : function (data, actions) {

            // Les produits à payer avec leurs détails
            var produits = [
                {
                    name : "Produit 1",
                    description : "Description du produit 1",
                    quantity : 1,
                    unit_amount : { value : 9.9, currency_code : "USD" }
                },
                {
                    name : "Produit 2",
                    description : "Description du produit 2",
                    quantity : 1,
                    unit_amount : { value : 8.0, currency_code : "USD" }
                }
            ];

            // Le total des produits
            var total_amount = produits.reduce(function (total, product) {
                return total + product.unit_amount.value * product.quantity;
            }, 0);

            // La transaction
            return actions.order.create({
                purchase_units : [
                    {
                        items : produits,
                        amount : {
                            value : total_amount,
                            currency_code : "USD",
                            breakdown : {

```

```

        item_total : { value : total_amount, currency_code : "USD"
    }
}
}]
});
},
// 4. Capturer la transaction après l'approbation de l'utilisateur
onApprove : function (data, actions) {
    return actions.order.capture().then(function(details) {

        // Afficher Les details de la transaction dans la console
        console.log(details);

        // On affiche un message de succès
        alert(details.payer.name.given_name + ' ' + details.payer.name.surname

    });
},
// 5. Annuler la transaction
onCancel : function (data) {
    alert("Transaction annulée !");
}

}).render("#paypal-boutons");

</script>
</body>
</html>

```

## # Passer du mode sandbox en production (live mode)

Après que vous ayez testé votre implémentation et que tout fonctionne correctement en sandbox, vous pouvez passer en production (live mode).

Pour passer en live mode et accepter les paiements par cartes, comptes bancaires ou solde PayPal sur votre site web, il vous faut :

1. un **compte Business** (ou **compte Professionnel**). Vous pouvez l'obtenir en allant sur [la page de connexion PayPal](#) puis sur **Ouvrir un compte**.
2. Au menu [My Apps & Credentials](#) dans votre compte Professionnel, allez à l'onglet **Live** (à côté de sandbox) puis cliquez sur **Create App** pour créer une application live. Renseignez le nom votre application puis cliquez sur **Create App**.

Une fois l'application créée, les identifiants (Client ID, Secret) sont générés automatiquement pour votre compte PayPal Professionnel :

Il ne reste plus qu'à remplacer l'identifiant **Client ID** de l'application sandbox par l'identifiant de l'application live lors de l'importation de la SDK JavaScript

```
<!-- Importation de la SDK JavaScript PayPal -->
<script src="https://www.paypal.com/sdk/js?client-id=LIVE_CLIENT_ID" ></script>
```

A vous maintenant d'accepter les paiements en ligne avec PayPal sur votre site web et faire grandir votre business à l'international !

#html    #javascript

♡ ❤ **Cette publication vous a plu ?**

Partagez-la avec vos ami(e)s sur les réseaux sociaux.

57  
Partages

### Wilo Ahadi, l'auteur

Passionné de l'informatique, je suis spécialiste en techniques des systèmes et réseaux, développeur web et mobile, infographiste et designer, ... J'aime partager mon expérience en formant sur la plateforme Akili School

 Voir profil

 Suivre

### Commentaires

Participer à la discussion...

S'IDENTIFIER AVEC

OU INSCRIVEZ-VOUS SUR DISQUS ?

Nom

Les meilleurs



[Jacques RAIMBAULT](#) • il y a un an

Bonjour et merci pour votre script dont je me suis largement inspiré pour continuer à mettre en place un paiement par Paypal sur mon site. J'avais un script php vieux de 10 ans qui ne fonctionnait plus pour cause de certificat périmé.

Par contre je n'arrive pas à faire apparaître dans la transaction la référence de la commande (chaine de caractères générée automatiquement et enregistrée dans la base de données avant le paiement avec les détails de la commande). J'ai essayé de renseigner 'reference\_id' mais je ne retrouve pas cette valeur dans la transaction sur mon compte Paypal. Vous avez une piste ? Merci d'avance.

## Autres publications



[Laravel : Générer un QR code avec simple-qrcode](#)



[Bootstrap : Créer un méga dropdown menu responsive](#)



[Laravel : Enregistrer les sessions des utilisateurs dans la Base de données](#)



[Laravel : Manipuler les images avec Intervention image](#)



[Laravel - Bootstrap : Afficher les erreurs de validation d'un formulaire](#)

→ [Voir toutes les publications de Wilo Ahadi](#)

## Sélection ebook



[Créer un site web dynamique en PHP relié à une base de données MySQL – 1](#)

Module 1 – Introduction au DÉVELOPPEMENT WEB par le LANGAGE PHP

Développement web



## ✉ Newsletter

Ne manquez plus les nouvelles publications en vous abonnant (Free) à la lettre d'information.

Votre adresse email

✓ Je m'abonne

## 🏢 A propos

Akili School est une plateforme qui vous permet d'apprendre et d'enseigner le numérique. Trouvez des articles, cours, formations, screencasts et ebooks



⌚ République Démocratique du Congo, Kinshasa

## ≡ Menu

- [Les cours](#)
- [Les formations](#)
- [Les livres](#)
- [Mon compte](#)

## ⌚ Aide & Support

- [Guide d'utilisation](#)
- [Conditions d'utilisation](#)
- [Politique de confidentialité](#)
- [Nous contacter](#)

## ✉ Newsletter

Votre adresse email

✓ Je m'abonne

