

# INTEGRATION DU PAYEMENT PAR PAYPAL

Pour pouvoir intégrer le mode paiement via Paypal dans une application web, il faut configurer certains points essentiels.

Tout d'abord, il faut :

- Disposer d'un compte PAYPAL et l'utiliser en tant que compte développeur
- Créer une application en mode SANDBOX et en mode LIVE dans l'interface et enregistrer leurs clés.
- Créer les codes qui permettent d'appeler l'API Paypal avec les informations nécessaires.
- Gérer enfin les données retournés par l'API à notre avantage.

## I- Création de compte PAYPAL développeur et configurations :

Pour pouvoir créer un compte paypal développeur, il faut d'abord un compte paypal.

Ensuite pour le faire passer en mode développeur, il faut entrer dans le lien :

<https://developer.paypal.com/>

## II- Création d'une application en mode SANDBOX et en mode LIVE

Le mode SANDBOX est le mode de test de l'API Paypal.

Dans ce mode, nous pouvons faire des transactions et des paiements sans besoin de vraies comptes Paypal.

Allons dans l'onglet [Sandbox > My Apps & Credentials](#). De là, nous allons créer une application de test. Une fois cela fait, Paypal nous donne des clés **public et secrète ()** qu'il faut enregistrer quelque part pour le fonctionnement de l'application.

Il faut aussi noter les comptes de test de l'application. A la création du compte PAYPAL.developer, il existe déjà **un compte business(marchant)** et **un compte personnel(client)** par défaut que l'on peut modifier à notre guise que ce soit : les adresses mails, mots de passes, nom et prénoms du titulaire ... Il ne faut pas oublier de noter **l'adresse mail** et le **mot de passe** du compte client car il sera utile pour le test plus tard.

## III- Intégration du côté de l'application :

Une fois que l'application est configurée du côté de paypal, passons à l'intégration du côté WEB.

Dans ce document, nous allons utiliser le JS SDK de l'API afin d'aller vite.

Tout d'abord, il faut appeler le script de Paypal dans la page où se trouverais notre bouton « PAYER ».

Le code se présente comme suit :

```
<!-- 1. Importation de la SDK JavaScript PayPal -->
<script src="https://www.paypal.com/sdk/js?client-
id=SANDBOX_CLIENT_ID"></script>
<script>
```

Ensuite pour pouvoir afficher le bouton de PAYPAL, il faut le monter à partir du code suivant :

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

<script>
    // 2. Afficher le bouton PayPal
    paypal.Buttons().render("#paypal-boutons");
</script>
```

Dans ce cas précis, le div avec l'id « paypal-boutons » sera le point où le bouton sera monté.

Voici un exemple complet du code :

```
<body>

<!-- 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_9KxNm1Za6SHBcAZLDn1vmID009pX2C6kZRLqVInLIT_Wys3C7W0Bs">
</script>
<script>
    // 2. Afficher le bouton PayPal
    paypal.Buttons().render("#paypal-boutons");
</script>
</body>
```

### 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 devient :

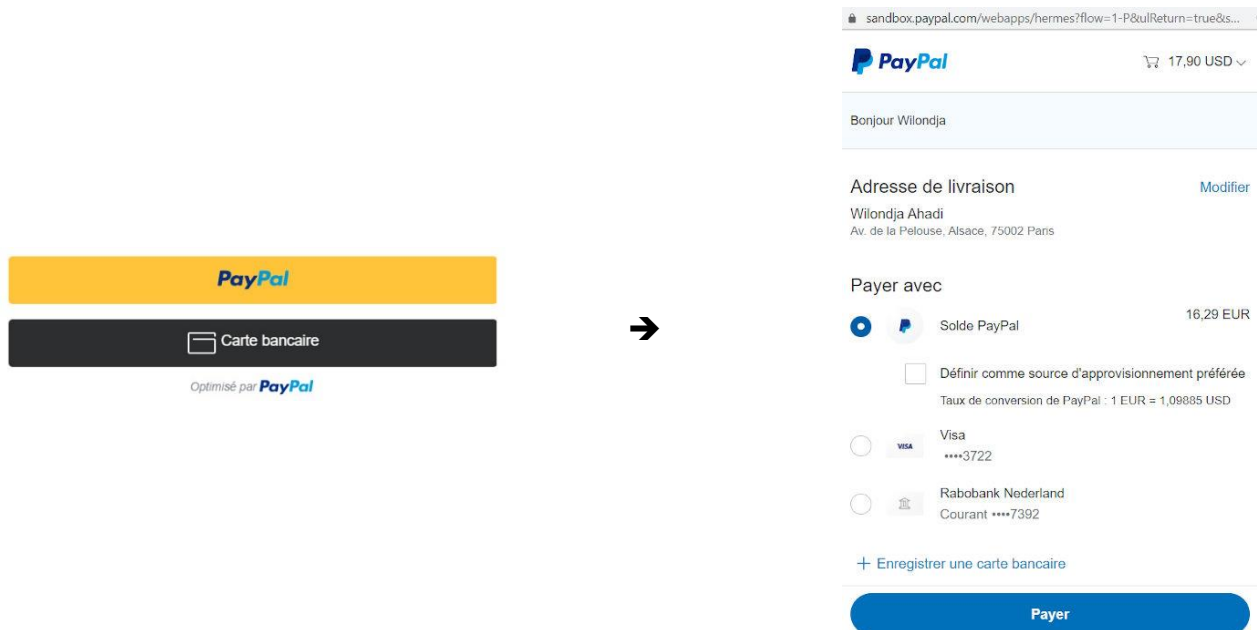
```
paypal.Buttons({
  // Configurer la transaction
  createOrder : function (data, actions) {
    // Les produits à payer avec leurs details
    var produits = [
      {
        name : "Produit 1",
        description : "Description du produit 1",
        quantity : 1,
        unit_amount : { value : 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 :



Il faut, bien sûr, utiliser les comptes de test déjà créés préalablement dans le dashborad développeur de PAYPAL.

### 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) {
      // 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 + ', votre transaction est effectuée. Vous allez
recevoir une notification très bientôt lorsque nous validons votre paiement.');
```

### 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 un 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_9KxNm1Za6SHBcAZLDn1vmID0009pX2C6kZRLqVInLIT_Wys3C7W0Bs">
</script>
```

```

<script>

    // 2. Afficher le bouton PayPal
    paypal.Buttons({
        // 3. 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" }

```

```

    }
  }
}

});

},
// 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 + ', votre transaction est effectuée. Vous allez recevoir
une notification très bientôt lorsque nous validons votre paiement.');
```

```

    });
  },
// 5. Annuler la transaction
onCancel : function (data) {
  alert("Transaction annulée !");
}
}).render("#paypal-boutons");
</script>
</body>
</html>

```

#### IV- 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>
```

Voilà en résumé l'intégration du paiement via PAYPAL par JS SDK dans une application WEB.