

BROUILLON

# Installer/Intégrer un système de paiement Paypal sur son site

La solution de micropaiement proposée par Paypal est facile à intégrer sur un site web.

Nous allons donc voir comment intégrer simplement la solution Paypal sur son site.

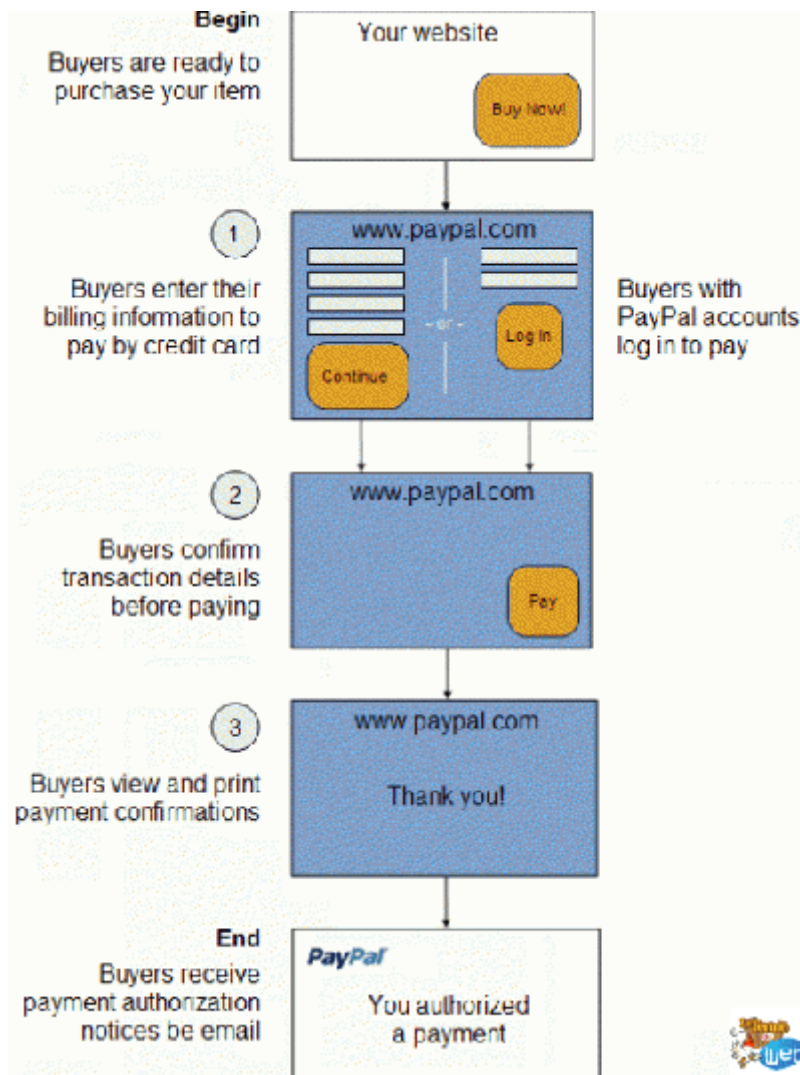
Une transaction PayPal typique se déroule comme suit :

Sur la page web du vendeur sur laquelle le produit ou le service peut être acheté, l'acheteur clique sur le bouton Acheter.

Il est redirigé sur le site PayPal où il a le choix entre :

- entrer ses coordonnées bancaires s'il n'a pas de compte PayPal
- ou s'identifier pour utiliser son compte PayPal.

3. Une nouvelle page PayPal affiche un récapitulatif ; l'acheteur confirme ou annule sa commande.
4. Une fois le paiement validé, le site PayPal affiche un message de validation. Paypal lance une requête IPN sur le site du vendeur pour qu'il traite le paiement. Nous verrons plus tard comment récupérer les données de cette requête.
5. Un lien sur la page ramène l'acheteur sur le site du vendeur où l'opération se termine.



## Pré-requis

- Disposer d'un compte Paypal

## Première étape : Essais avec des comptes de test

Effectuer un test : voir la page [PayPal : Test des appels de l'API classique](#)

## Autres étapes

### Utiliser l'IPN Paypal avec PHP

Paypal offre divers exemples de codes [https://cms.paypal.com/us/cgi-bin/?cmd=\\_render-content&content\\_ID=developer/library\\_code\\_ipn\\_co\\_de\\_samples](https://cms.paypal.com/us/cgi-bin/?cmd=_render-content&content_ID=developer/library_code_ipn_co_de_samples) (PHP : [ici](#)) à insérer sur son site, il suffit de le copier/coller en y ajoutant les actions que vous voulez effectuer selon les cas.

Tout est assez clair, veuillez cependant à changer la ligne :

```
$fp = fsockopen ('ssl://www.paypal.com', 443, $errno, $errstr, 30);
```

par

```
$fp = fsockopen('ssl://www.sandbox.paypal.com', 443, $errno, $errstr, 30);
```

...lors de vos tests.

De même, si vous avez utilisé le champ personnalisé, pensez à ajouter le code ci-dessous au niveau des variables postées, afin de récupérer la valeur envoyée :

```
$custom = $_POST['custom'];
```

Il y a de nombreux champs possibles selon vos besoins, pensez donc bien à récupérer les valeurs de tous les champs additionnels éventuellement envoyés (comme vous le feriez sur un formulaire classique en somme...).

Comme vous l'aurez compris, la partie du code qu'il va falloir éditer selon vos besoins correspond aux zones entre ces accolades :

```
if (strcmp ($res, "VERIFIED") == 0) {  
//Paiement OK  
} else {  
//Problème  
}
```

En clair, si le paiement est OK, vous faites le nécessaire conformément à ce que vous avez vendu, et s'il y a un problème vous contactez l'acheteur. Vous pouvez également gérer les remboursements via l'IPN. Pour cela, il suffit d'ajouter un cas de ce type (dans la zone « *paiement OK* » du *snippet ci-dessus*): `<code> if($_REQUEST["payment_status"] == "Refunded" || $_REQUEST["payment_status"] == "Reversed" { Mise à jour } </code>`

Pour info, toutes les transactions sont enregistrées dans l'historique de votre compte paypal (My account > History > IPN).

Nb. Comme précisé en introduction, le but de cet article n'est pas de vous présenter une solution fonctionnelle de A à Z, mais de montrer comment intégrer un tel système, afin de pouvoir l'adapter facilement à votre site. Néanmoins, si vous avez besoin d'un exemple concret, voici un excellent article de Tuts+ (Anglais): [Utiliser l'IPN Paypal avec PHP](#)

## Intégration de paramètres additionnels au bouton

Comme nous venons de le voir, pour intégrer un bouton sur notre site, il suffit de copier/coller le code fourni par Paypal. Cependant, il peut s'avérer utile de passer des variables supplémentaires lors du paiement, afin par exemple, d'identifier un membre (upgrade de compte, etc.). De nombreuses options sont disponibles, voici [votre bible](#) à ce sujet.

Pour information, sachez que cette personnalisation fonctionne le plus simplement du monde. Ainsi, si vous désirez récupérer l'ID d'un membre une fois le paiement effectué, vous devez ajouter ceci sur le formulaire :

```
<input type="hidden" name="custom" value="id_membre" />
```

et ceci sur la page ipn.php (que nous allons voir ci-après) :

```
$_POST[ 'custom' ]
```

Bien entendu, cela fonctionne de la même façon pour tous les options disponibles...

## Redirection vers le site

Une fois les vérifications concernant le paiement effectuées, il est possible de spécifier deux URLs de destination vers votre site (l'une pour un paiement validé, l'autre pour un paiement annulé). Cela n'est pas obligatoire, mais largement répandu, et honnêtement plutôt indispensable d'un point de vue ergonomique. (au moins pour ce qui est d'un paiement validé)

Affichez sur ces pages ce que vous voulez, sans oublier que c'est un emplacement idéal pour le message de remerciement ;)

nb. Ces pages de destinations sont définies à la création du bouton. (point #3 de ce tutoriel)

## Phase de test

Maintenant que tout est prêt, il va falloir tester le système. Connectez-vous à votre compte développeur Paypal, ouvrez un session « Acheteur », puis rendez-vous sur la page où vous avez mis le bouton paypal. En cliquant dessus vous allez être redirigés sur la page de paiement, telle qu'un acheteur la verra une fois que ce sera en ligne.

Essayez différentes configurations (avec un compte acheteur approvisionné, non approvisionné, annuler des achats, etc.), et vérifiez scrupuleusement que l'intégralité des actions sensées être exécutées en fond sont correctement réalisées. (Selon ce que vous vendez cela peut impliquer de nombreuses requêtes et cas).

## Mise en ligne

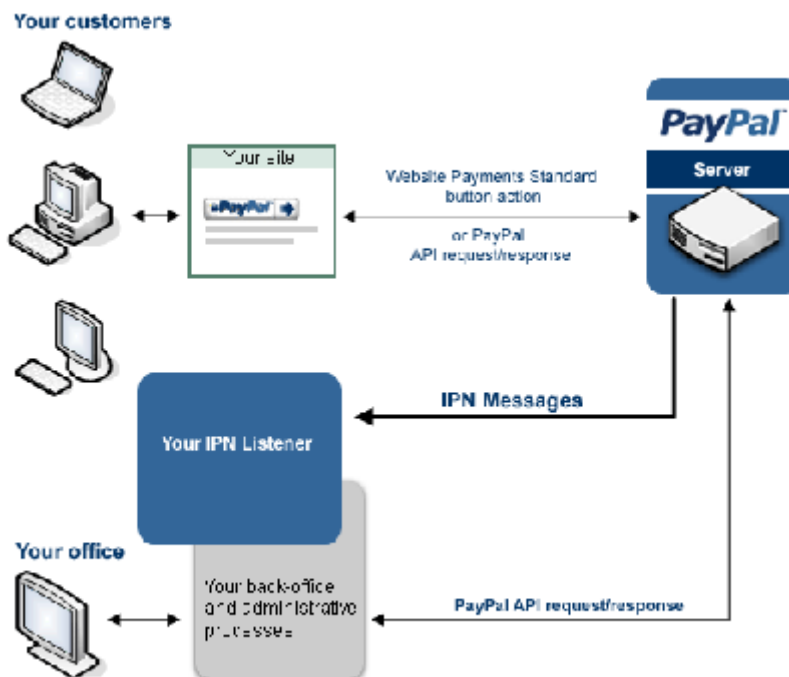
Pour passer à la mise en production de votre système de paiement, vous devrez reproduire les paramètres que nous venons de voir à l'identique sur votre compte officiel. Une fois la solution en ligne, il est préférable de la tester une dernière fois afin de s'assurer qu'aucun grain de sable ne soit venu gripper l'engrenage sur la version définitive.

A ce titre, je vous conseille vivement de sauvegarder vos codes de test (surtout si le formulaire de vente est complexe), cela vous permettra de pouvoir repasser rapidement par la sandbox en cas de problèmes.

Si tout est OK, pensez à afficher vos Conditions Générales de Vente...

## Résumé

Comme vous l'aurez l'aurez compris, le mode de fonctionnement est le suivant :



Ce qui est intéressant avec cet IPN, c'est qu'il communique en temps réel avec votre site, les informations sont donc synchronisées en fond de façon transparente. Peu importe si l'acheteur ferme son navigateur avant la page de remerciement de votre site, les informations sont déjà enregistrées dans votre base de données grâce à l'IPN.

## Conclusion

Voilà pour ce tutoriel volontairement épuré, destiné à vous montrer de façon simple comment intégrer un système de paiement sécurisé via Paypal sur son site. Évidemment, il est possible d'aller beaucoup plus loin, l'API fournie par Paypal étant très complète. Des classes PHP existent aussi pour vous faciliter le travail, elles sont d'ailleurs souvent indispensables pour travailler dans de bonnes conditions.

## Problèmes connus

## Voir aussi

- <http://www.lafermeduweb.net/billet/tutorial-integrer-paypal-a-son-site-web-en-php-partie-1-275.html//>

Contributeurs principaux : *Jamaïque*.

From:

<https://www.nfrappe.fr/doc-0/> - **Documentation du Dr Nicolas Frappé**

Permanent link:

<https://www.nfrappe.fr/doc-0/doku.php?id=tutoriel:paypal:website:start>

Last update: **2022/08/13 21:58**

