|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **PBX – Paybox V2**  ***Domaine SISCOL : TEC***  **Date : 10/04/2015 Version : V2.0**  **État : Final**  **Réf. : DSI-TEC-PBX\_Paybox-V2.0.doc**  **Annexes :** | | | | | | | |
| **Validation** | | | | | | | |
| Rédaction | | LERY Jean-Michel | | | |  | |
| Validation | |  | | | |  | |
| Approbation | |  | | | |  | |
|  | | |  | | | |  |
| **Diffusion** | | | | | | | |
| **Noms** | | | | | **Fonction** | | |
| **CNAM** | | | | |  | | |
| Olivier Labone-Jayat | | | | | Responsable Site Web de SISCOL | | |
| Jean-Michel Léry | | | | | Equipe Transverse | | |
| Olivier Villin | | | | | Directeur Adjoint de la DSI | | |
| Florence Vitalis | | | | | Directrice de la DSI | | |
| **Historique des évolutions** | | | | | | | |
| **Version** | **Date** | | **Évolution** | | | | |
| 1.0 | 08/12/2008 | | Première version-Projet Paiement en ligne (Avant SISCOL) | | | | |
| 2.0 | 10/04/2015 | | Seconde version : intégration des nouveautés Paybox : Cryptage clé publique/ clé privée de la transaction + Utilisation du système 3D Secure | | | | |
|  |  | |  | | | | |
|  |  | |  | | | | |
|  | | | | | | |  |
| **Objet** : Ce document présente les nouvelles fonctionnalités de l’intégration de la plateforme technique de paiement en ligne de Paybox. | | | | | | | |
| **Bibliographie – Annexes** | | | | | | | |
| **Nom du document** | | | | **Fichier** | | | |
|  | | | |  | | | |

Sommaire

1 Le paiement en Ligne avec Paybox 3

1.1 Préambule 3

1.1.1 Objectifs 3

1.1.2 Recommandations 3

1.1.2.1 Incompatibilité avec la V1 3

1.1.2.2 Installation autonome 3

1.2 Présentation de Paybox 4

1.2.1 Coût 4

1.2.2 La boutique Paybox 4

1.2.2.1 La boutique de l’entreprise 4

1.2.2.2 La boutique de test mutualisée 5

1.2.2.3 Le Back-office 5

1.2.3 Principe de fonctionnement 5

1.2.3.1 La saisie des informations sur le site marchand 5

1.2.3.2 La transmission des informations du site marchand à Paybox 7

1.2.3.3 La saisie des informations bancaires 9

1.2.3.4 L’envoi du reçu de paiement à l’acheteur 11

1.2.3.5 Le retour des informations après paiement 12

1.2.3.6 L’accès au Back-office Paybox 13

1.2.4 Sécurisation par clé publique/clé privée 14

1.2.4.1 Principe 14

1.2.4.2 La clé publique de la boutique Paybox 15

1.2.4.2.1 Génération de la clé 15

1.2.4.2.2 Récupération de la clé 15

1.2.4.3 Génération de la clé privée ou « empreinte » de la transaction 16

1.3 Mise en œuvre de la solution 17

1.3.1 Paiement en une ou plusieurs fois sans 3D Secure 17

1.3.1.1 Intégration dans le site marchand 17

1.3.1.1.1 Formulaire de saisie des informations 17

1.3.1.1.2 Programme d’envoi des informations à Paybox 19

1.3.1.2 Interprétation du retour de Paybox 27

1.3.1.3 Fichiers annexes 30

1.3.1.3.1 La feuille de style 30

1.3.1.3.2 Les sous-programmes 32

1.3.1.3.3 Le fichier d’interprétation des erreurs 38

1.3.1.3.4 Le fichier des codes ISO des pays 39

1.3.2 Paiement avec 3D-Secure 40

Le paiement en Ligne avec Paybox

## Préambule

### Objectifs

Dans ce document nous présentons la mise en œuvre de programmes PHP dans le cadre du paiement en ligne sécurisé via la plateforme technique Paybox.

Nous ne présentons pas tous les aspects de cette solution qui est très riche, mais seulement le paiement en ligne en une ou plusieurs échéances.

Le site www.paybox.com propose une information complète de cette solution.

### Recommandations

#### Incompatibilité avec la V1

Il est très important de noter que la version actuellement proposée par Paybox est « incompatible » avec la version mise en œuvre dans le cadre du projet de paiement en ligne de 2008, projet qui a ensuite été reprise lors de la mise en œuvre de SISCOL.

En effet, cette nouvelle version utilise un cryptage par la méthode clé publique / clé privée qui n’est pas utilisée en V1.

La documentation de Paybox indique clairement que :

**Pour les sites marchands utilisant l’ancienne méthode d’accès à leur boutique Paybox, lors du premier accès aux boutiques Paybox avec la nouvelle méthode, il n’est plus possible d’utiliser l’ancienne méthode.**

#### Installation autonome

La solution présentée dans les pages suivantes utilise le binaire exécutable (CGI) qui est accessible en ligne sur le site de Paybox.

Voici ces URLs utilisée dans le formulaire d’envoi des données du site marchand vers la boutique Paybox :

Plateforme de test ou pré-production :

https://**preprod-tpeweb.paybox.com**/cgi/MYchoix\_pagepaiement.cgi

Plateforme de production :

https://**tpeweb.paybox.com**/cgi/MYchoix\_pagepaiement.cgi (serveur principal)

https://**tpeweb1.paybox.com**/cgi/MYchoix\_pagepaiement.cgi (serveur secondaire)

Or cet usage rend le site marchand dépendant de l’évolution de ces binaires, qui peuvent changer rapidement et donc rendre le fonctionnement du paiement en ligne de production inopérant.

**Il est donc préconisé de récupérer le binaire téléchargeable sur le site de Paybox, et de le conserver sur un serveur en local afin de l’utiliser dans le formulaire à la place du CGI hébergé par Paybox.**

Ainsi en cas d’évolution de celui-ci, c’est bien la version stockée localement et donc n’ayant aucun évolution qui sera toujours utilisée.

C’est ce qui a été mis en place lors du paiement en ligne de la version 1. L’URL d’accès au CGI local est actuellement :

Plateforme de production et de pré-production en exploitation :

**http://siscolpayboxcgi.cnam.fr/siscolpayboxv3.cgi**

## Présentation de Paybox

### Coût

Le critère le plus important pour le choix d’une plateforme technique de paiement en ligne est son coût d’exploitation.

En avril 2015, Paybox propose un coût public forfaitaire de 25 € pour 100 transactions par mois, puis de 0,085 € par transaction à partir de la 101ème transaction dans le mois. A titre informatif, en 2008, il était de 23,15€ pour 100 transactions par mois, et de 0,067 € par transaction à partir de la 101ème.

En 2015, le coût d’ouverture d’une boutique Paybox, réglé une seule fois, est de 290 € (390 € en 2008).

Cette tarification forfaitaire présente un avantage indéniable par rapport aux solutions dont le coût est calculé selon un pourcentage du montant de chaque transaction.

### La boutique Paybox

Le mode de fonctionnement de Paybox nécessite une mise en œuvre technique du côté de l’entreprise cliente pour accéder à sa boutique Paybox.

#### La boutique de l’entreprise

Chaque client dispose d’une boutique qui lui est propre, ouverte auprès de Paybox. Elle est connue via une identification particulière.

Cette boutique est accessible en mode :

* pré-production : dans ce mode d’accès, les vraies cartes bancaires (CB, VISA, MasterCard, American Express, …) sont utilisables, mais aucune compensation bancaires (débit) n’est effectuée. Ce mode permet de tester le paiement via le site marchand de l’entreprise, sans avoir aucun paiement effectif. Une carte « virtuelle » est également disponible, ce qui évite l’utilisation de vraies cartes bancaires.
* production : dans ce mode d’accès, le paiement est réel. Seules les vraies cartes sont utilisables.

#### La boutique de test mutualisée

Paybox propose une boutique de test mutualisée, accessible à tous sans restriction, même à ceux qui ne sont pas client de Paybox.

Elle permet de mettre en œuvre la solution technique dans le site marchand de l’entreprise, sans attendre l’ouverture de sa propre boutique Paybox. De plus, Paybox offre une assistance (via courriel ou téléphone), pour cette boutique mutualisée donc sans être client.

#### Le Back-office

Chaque boutique Paybox possède un accès « Back-office » c’est-à-dire un site web avec identification (via un login et un mot de passe) permettant de voir l’historique des transactions, leur état (accepté, refusé, annulé, …), leur montant, les éventuels abonnements, etc.

Il permet aussi d’annuler une transaction (avant sa compensation bancaire), de faire un remboursement, de gérer une liste « grise » c’est-à-dire de cartes bancaires pour lesquelles vous refusez le paiement, d’extraire l’historique des transactions sous la forme d’un fichier Excel, etc.

### Principe de fonctionnement

Le paiement via la solution Paybox se déroule en 5 phases :

1. La saisie des informations sur le site marchand de l’entreprise ;
2. La transmission des informations du site marchand vers la boutique Paybox ;
3. La paiement : saisie des informations bancaires par l’acheteur, sur les pages sécurisées de la boutique Paybox ;
4. L’envoie du reçu de paiement à l’acheteur, par Paybox ;
5. Le retour des informations sur le paiement, de la boutique Paybox vers le site marchand de l’entreprise.

Il est ensuite possible de consulter l’historique des transactions dans le Back-office :

1. L’accès au Back-office de la boutique Paybox.

#### La saisie des informations sur le site marchand

La première étape est d’avoir développé un site Web marchand permettant à l’acheteur de sélectionner un ou plusieurs produits dans un « panier », et d’avoir une somme à régler.

C’est à cette étape que l’on génère les informations commerciales qui seront ensuite transmises à Paybox, comme la **somme à régler**, et une **référence commerciale**.

Cette **référence commerciale** est importante et doit contenir, sous la forme d’une chaîne de caractères (255 caractères au maximum), des informations comme le numéro de dossier du site marchand, éventuellement le nom ou le prénom de l’acheteur, l’horodatage de l’achat, etc. Cette référence doit contenir toute information utile qui permette **d’identifier clairement et facilement un paiement**.

*Attention :*

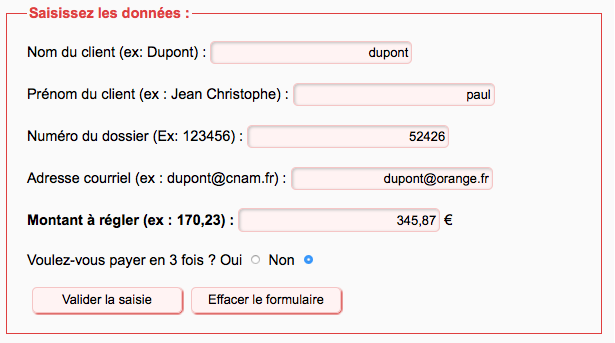
***La référence est la seule information commune entre le site marchand et le site Paybox****. Il est donc important de bien réfléchir à sa forme et à son contenu.*

*Par exemple un numéro de dossier est souvent suffisant. Mais alors seul ce numéro de dossier apparaîtra sur le Back-office de Paybox. Il faudra aller sur le Back-office du commerçant pour savoir à qui correspond ce dossier.*

*Il est très souvent intéressant de mettre dans cette référence, des données qui informent clairement sur le dossier comme : le nom et le prénom de l’acheteur, le mode de règlement en une ou plusieurs fois, la date et l’heure du paiement. Ceci évite de devoir faire des allers retours entre le Back-office du commerçant et celui de la boutique Paybox lorsqu’on a besoin de vérifier l’état d’un règlement.*

*De plus, il est préférable de remplacer les espaces par des soulignés afin de faciliter les traitements ultérieurs.*

L’écran suivant présente un écran de saisie « simulant » un site marchand :



Le nom, le prénom, l’adresse courriel seront par exemple demandé à l’acheteur sur le site marchand. Contrairement à ce qui est présenté sur cet écran, le montant à régler et le numéro de dossier seront calculés automatiquement lors de l’achat par le site marchand.

Les boutons radio du mode de paiement en une ou trois fois, montre qu’il est possible de faire un règlement en plusieurs fois (4 échéances maximales sous Paybox, dont la première est réglée immédiatement).

Dans ce cas il faut fournir à Paybox, les montants et les dates des échéances suivantes.

L’utilisateur validera une seule fois tous ces paiements dont les montants et dates seront présentés au moment de saisir son numéro de carte bancaire.

Par la suite, à chaque date indiquée, Paybox effectuera automatiquement, et sans aucune intervention humaine, le règlement du montant qui a été précisé à cette date et enverra automatiquement un reçu à l’acheteur via son adresse courriel. Il n’y a absolument rien à faire, sauf à relever régulièrement la liste des règlements effectués sur le Back-office de sa propre boutique Paybox.

#### La transmission des informations du site marchand à Paybox

Une fois les informations commerciales obtenues ou générées par le site marchand de l’entreprise, il faut transmettre ces informations à la boutique Paybox.

La transmission de ces informations du site marchand vers sa boutique Paybox utilise un formulaire HTML. La majorité des variables sont cachées (hidden).

En voici un exemple, les valeurs présentées sont à titre indicatif :

<form method="POST" action="https://urlserveur.paybox.com/cgi/MYchoix\_pagepaiement.cgi">

<input type="hidden" name="PBX\_SITE" value="1999888">

 <input type="hidden" name="PBX\_RANG" value="32">

 <input type="hidden" name="PBX\_IDENTIFIANT" value="110647233">

 <input type="hidden" name="PBX\_TOTAL" value="34587">

 <input type="hidden" name="PBX\_DEVISE" value="978">

 <input type="hidden" name="PBX\_CMD" value="052426\_DUPONT\_20150408\_104701">

 <input type="hidden" name="PBX\_PORTEUR" value="dupont@orange.fr">

 <input type="hidden" name="PBX\_RETOUR" value="Mt:M;Ref:R;Auto:A;Erreur:E">

<input type="hidden" name="PBX\_HASH" value="SHA512">

 <input type="hidden" name="PBX\_HMAC" value="F2A799494504F9E50E91E44C129A45">

<input type="hidden" name="PBX\_TIME" value="2015-04-08T10:47:01+02:00">

<input type="hidden" name="PBX\_EFFECTUE" value="http://serv.fr/retour.php">

<input type="hidden" name="PBX\_REFUSE" value="http://serv.fr/retour.php">

<input type="hidden" name="PBX\_ANNULE" value="http://serv.fr/retour.php">

<input type="submit" value="Payer">

</form>

Les variables de ce formulaire sont détaillées dans la documentation téléchargeable sur le site de Paybox. A travers ce formulaire, le site marchand transmet à Paybox :

* Des informations commerciales comme :
  + Le montant de la transaction en centimes (PBX\_TOTAL) ;
  + Le code de la devise de la transaction (PBX\_DEVISE), par exemple l’euro (978) ;
  + La référence du dossier : une chaine de 255 caractères au plus (PBX\_CMD) ;
  + L’adresse courriel de l’acheteur (PBX\_PORTEUR), pour recevoir le reçu de paiement.
* Des informations techniques comme :
  + Les codes de la boutique Paybox de l’entreprise (PBX\_SITE, PBX\_RANG, PBX\_IDENTIFIANT) ;
  + La liste des variables que Paybox doit retourner au site marchand (PBX\_RETOUR), après la saisie des informations bancaires par l’acheteur ;
  + La méthode de cryptage de l’empreinte de la transaction (PBX\_HASH). C’est l’algorithme de hachage ;
  + L’empreinte (clef privée) de la transaction (PBX\_HMAC). Cette signature est calculée à partir de la clef publique fournie avec la boutique Paybox ;
  + L’horodatage de la transaction au format ISO8601 (PBX\_TIME) ;
  + Les URLs de retour au site marchand quand le paiement est effectué, refusé ou annulé (PBX\_EFFECTUE, PBX\_REFUSE, PBX\_ANNULE).

Les variables suivantes sont obligatoires dans toute requête, et doivent être présentées dans cet ordre :

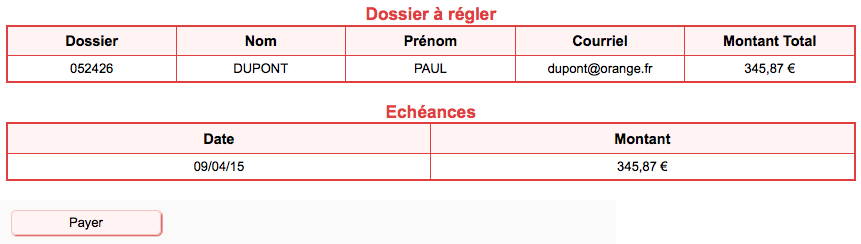
* PBX\_SITE
* PBX\_RANG
* PBX\_IDENTIFIANT
* PBX\_TOTAL
* PBX\_DEVISE
* PBX\_CMD
* PBX\_PORTEUR
* PBX\_RETOUR
* PBX\_HASH
* PBX\_HMAC
* PBX\_TIME

Il existe bien d’autres variables facultatives, comme PBX\_EFFECTUE, PBX\_REFUSE ou PBX\_ANNULE.

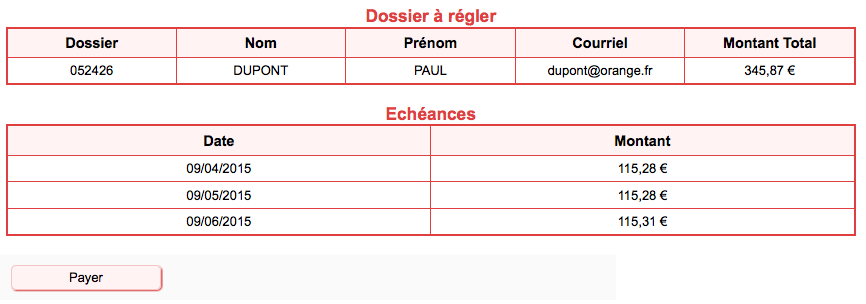
Voici un exemple de la présentation de validation de la commande. Comme les variables sont cachées (hidden), seul le bouton « Payer » du formulaire apparaît.

Il est donc important de présenter à l’acheteur un récapitulatif des informations le concernant et des sommes à régler avec leur date de règlement AVANT de transmettre les informations à Paybox, c’est-à-dire avant de saisir les informations bancaires.

L’écran suivant récapitule un règlement en une seule fois



L’écran suivant récapitule un règlement en une trois fois (dans le cas d’une sélection du bouton radio sur « oui » dans l’écran précédent), pour le même montant.



Le clic sur le bouton « Payer » envoie les informations à Paybox via le formulaire, en utilisant le programme indiqué dans le champ « action », qui est dans notre exemple : **https://*urlserveur*.paybox.com/cgi/MYchoix\_pagepaiement.cgi.**

La valeur de ***urlserveur*** varie selon que l’on utilise le mode de pré-production (tests) ou le mode de production (paiement réel).

La liste des serveurs Paybox disponibles est :

Plateforme de test ou pré-production :

https://**preprod-tpeweb.paybox.com**/cgi/MYchoix\_pagepaiement.cgi

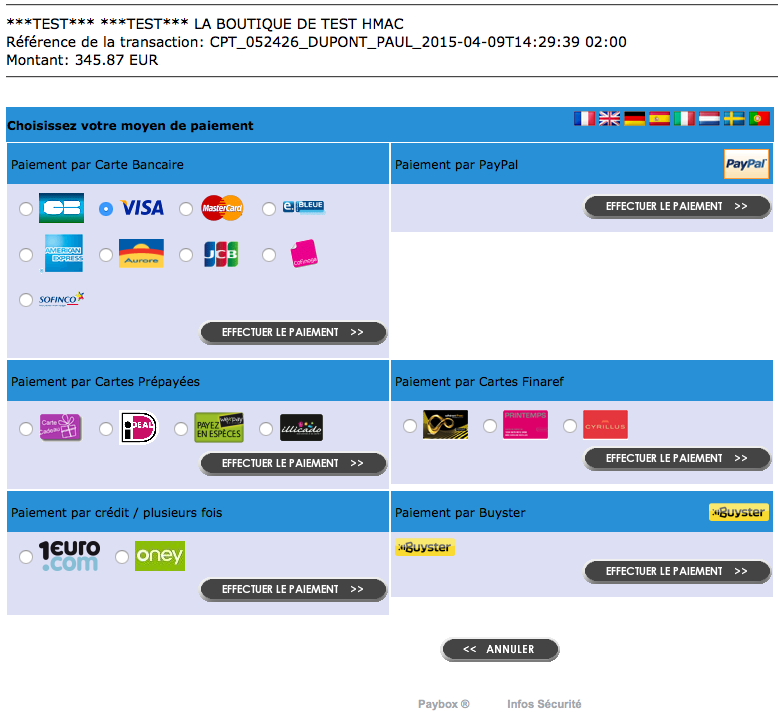
Plateforme de production :

https://**tpeweb.paybox.com**/cgi/MYchoix\_pagepaiement.cgi (serveur principal)

https://**tpeweb1.paybox.com**/cgi/MYchoix\_pagepaiement.cgi (serveur secondaire)

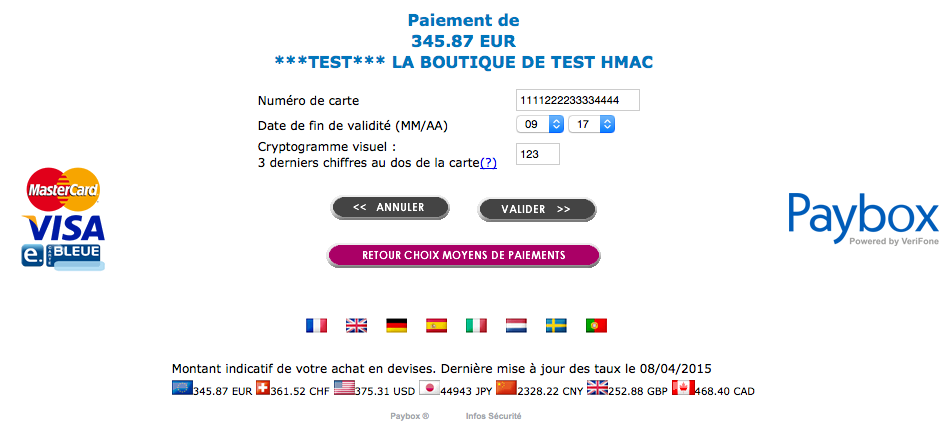
#### La saisie des informations bancaires

Une fois le formulaire validé, l’écran de Paybox apparaît. Sur la boutique de test mutualisé, un premier écran propose de choisir sa carte bancaire.

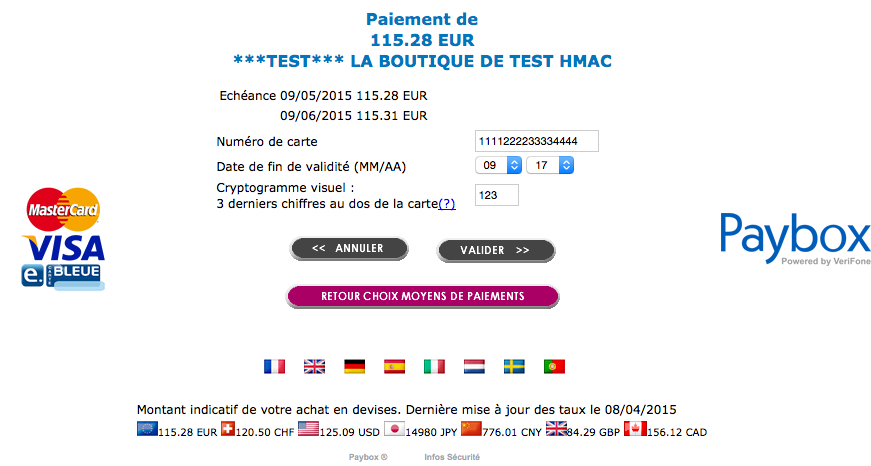


L’écran Paybox de paiement présente la somme à régler immédiatement, ou les sommes et les dates en cas de paiement en plusieurs fois.

L’écran suivant présente un paiement en une seule fois :



L’écran suivant présente un paiement en trois fois :



Cet écran permet à l’acheteur de saisir les informations de sa carte bancaire.

Dans le cas de la boutique de test mutualisée, vous pouvez utiliser votre carte bancaire ou bien une carte « virtuelle » proposée par Paybox uniquement pour le test.

Voici les codes de cette carte « virtuelle » :

* Numéro de la carte : 1111222233334444
* Date de validité : choisir une date valide
* Cryptogramme : 123

L’acheteur à la possibilité de faire trois essais avant de se voir refuser le paiement. Il peut également annuler le règlement à cette étape.

#### L’envoi du reçu de paiement à l’acheteur

Si le règlement a été accepté à l’étape précédente, l’acheteur reçoit un reçu de son paiement à son adresse courriel, qui a été fournie à Paybox via le formulaire dans la variable PBX\_PORTEUR.

Voici e reçu envoyé à l’acheteur dans le cadre d’un règlement en une seule fois.

+-------------------------------------------+

! ATTENTION CECI N'EST PAS UN VRAI PAIEMENT !

!  IL N'Y A PAS EU DE VRAIE AUTORISATION    !

+-------------------------------------------+

Ref commande:CPT\_052426\_DUPONT\_PAUL\_2015-04-09T14:29:39 02:00

CARTE BANCAIRE

le 09/04/2015 à 14:39

 TEST PAYBOX HMAC 1

1999888

111122-----------

1709

00 032 5954030

M DEBIT @

AUTO:     XXXXXX

MONTANT =      345.87 EUR

TICKET A CONSERVER

Voici e reçu envoyé à l’acheteur dans le cadre d’un règlement en plusieurs fois.

Le type « abonnement » apparaît. La date et le montant du prochain prélèvement sont indiqués.

+-------------------------------------------+

! ATTENTION CECI N'EST PAS UN VRAI PAIEMENT !

!  IL N'Y A PAS EU DE VRAIE AUTORISATION    !

+-------------------------------------------+

Abonnement

Prochain prelevement le 09/05/2015 :       115.28 EUR

Ref commande:ABT\_052426\_DUPONT\_PAUL\_2015-04-09T14:39:25 02:00

CARTE BANCAIRE

le 09/04/2015 à 14:39

 TEST PAYBOX HMAC 1

1999888

111122-----------

1703

00 032 5958471

M DEBIT @

AUTO:     XXXXXX

MONTANT =      115.28 EUR

TICKET A CONSERVER

#### Le retour des informations après paiement

Si des URLs de retours vers le site marchand ont été indiquées à Paybox via les variables PBX\_EFFECTUE, PBX\_REFUSE et PBX\_ANNULE, alors à la fin de l’écran de saisie des informations bancaires, Paybox retourne à l’une de ces adresses les informations demandées dans la variable PBX\_RETOUR.

* L’url précisée dans PBX\_EFFECTUE est utilisée quand le règlement a été effectué.
* L’url précisée dans PBX\_REFUSE est utilisée quand le règlement a échoué (nombre d’essais dépassé, carte volée, compte bancaire insuffisamment approvisionné, etc.).
* L’url précisée dans PBX\_ANNULE est utilisée quand l’acheteur a cliqué sur le bouton « Annuler » de l’écran de saisie des informations bancaires.

La variable PBX\_RETOUR indique les informations à retourner au site marchand. Par exemple M indique le montant, R la référence, A l’autorisation et E le code d’erreur.

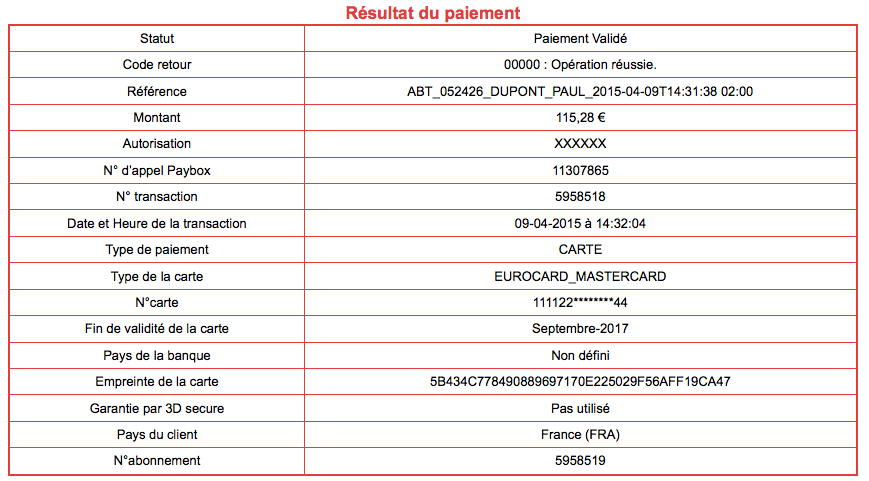
Ce retour ce fait par la méthode GET (par défaut). Chaque information doit être précédée par un texte et le caractère « : », représentant le nom de la variable. Le caractère de séparation entre ces différentes informations est le « ; ».

Voici un exemple de contenu : Mt:M;Ref:R;Auto:A;Erreur:E

Il faut donc traiter cette information comme si elle venait d’un formulaire envoyé par Paybox en méthode GET, via un programme PHP.

Les URLs de retours indiquées par PBX\_EFFECTUE, PBX\_REFUSE et PBX\_ANNULE doivent correspondre à un programme PHP de traitement de ces données. Le plus simple est que ces trois variables contiennent l’URL du même programme PHP qui traite tous les cas de retour selon la valeur de la variable « Erreur ».

Voici un exemple de l’affichage produit par un programme PHP interprétant les informations envoyées. Dans cet exemple, beaucoup d’informations sont demandées en retour comme par exemple le type de la carte bancaire, son numéro, sa date de validité, le pays de la banque émettrice, etc.



*Remarques:*

*-1- Contrairement à ce qui est affiché sur l’écran précédent qui présente juste l’interprétation des informations obtenue en retour,* ***le programme PHP devra simplement confirmer à l’acheteur, qui est de retour sur le site marchand, que sa commande est validée, ou qu’elle est annulée si le règlement à échoué ou si il a été annulé. Les autres informations ne doivent pas apparaître à l’écran mais elles doivent être conservées dans une base de données****.*

*Par la suite, il est indispensable de développer une interface Web d’accès à cette base de donnée, ce qui constituera le Back-office du site marchand !*

*-2- Paybox supporte le système 3D Secure, y compris sur le site de test mutualisé.*

*Ce système permet de sécuriser le paiement par la saisie d’un code unique envoyé par SMS sur le portable de l’acheteur. Le numéro de portable est transmis par la banque émettrice de la carte bancaire au moment du règlement sur le site de Paybox.*

*La boutique mutualisée simule le système 3D Secure par l’acceptation d’un certificat. Il n’y a pas d’échange de SMS ni besoin d’un numéro de téléphone portable.*

#### L’accès au Back-office Paybox

Il n’existe que deux URL d’accès au Back-office de Paybox :

* Le Back-office de test : **https://preprod-admin.paybox.com**
* Le Back-office de production : **https://admin.paybox.com**

C’est l’identifiant et le mot de passe qui dirige vers le Back-office de sa boutique Paybox.

Dans le cas de la boutique de test mutualisée *classique* ces informations sont :

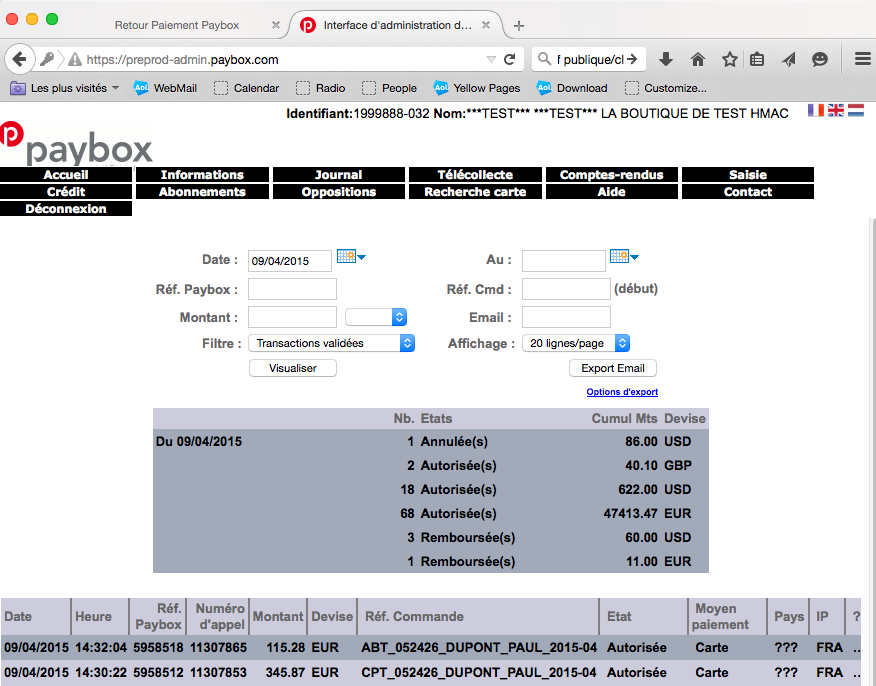
* Login : 199988832
* Mot de passe : 1999888I

Dans le cas de la boutique de test mutualisée *3D-secure* ces informations sont :

* Login : 199988843
* Mot de passe : 1999888I

L’onglet Journal donne accès à l’historique des transactions. La boutique mutualisée étant ouverte à tous, vous trouverez vos transactions de test parmi d’autres transactions, ce qui bien sûr ne sera pas le cas avec votre propre boutique.

Voici un exemple de Back-office de la boutique mutualisée classique. L’encadré montre la trace de notre transaction de test.



### Sécurisation par clé publique/clé privée

Afin de sécuriser l’envoi des informations entre le site marchand et la boutique Paybox, la solution technique de Paybox utilise la méthode de cryptage clé publique / clé privé.

#### Principe

A partir d’une clé publique spécifique à sa propre boutique Paybox, on génère une « empreinte » de l’ensemble des informations transmises via le formulaire à Paybox, c’est la clé privée.

Le nom de l’algorithme de cryptage utilisé ainsi que cette empreinte, la clé privée, sont transmis à Paybox dans le formulaire en même temps que les autres informations.

A la réception, Paybox génère à nouveau l’empreinte et la compare avec celle qui est transmise pour valider la transaction.

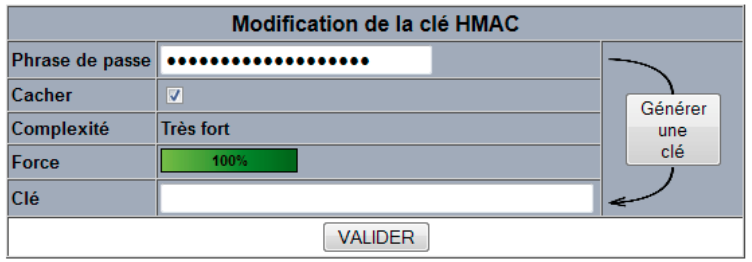
#### La clé publique de la boutique Paybox

##### Génération de la clé

La clé publique est spécifique au site marchand. IL faut donc la générer une première fois dans le Back-office de sa boutique Paybox. L’interface de génération se trouve dans l’onglet « **Informations**» du Back-office.

Cette clé publique est modifiable par la suite.

Voici une présentation de cette interface issue de la documentation Paybox.



##### Récupération de la clé

L’écran suivant montre comment récupérer cette clé dans le Back-office de la boutique de test mutualisée qui est utilisée pour la section concernant la mise en œuvre du paiement en ligne.

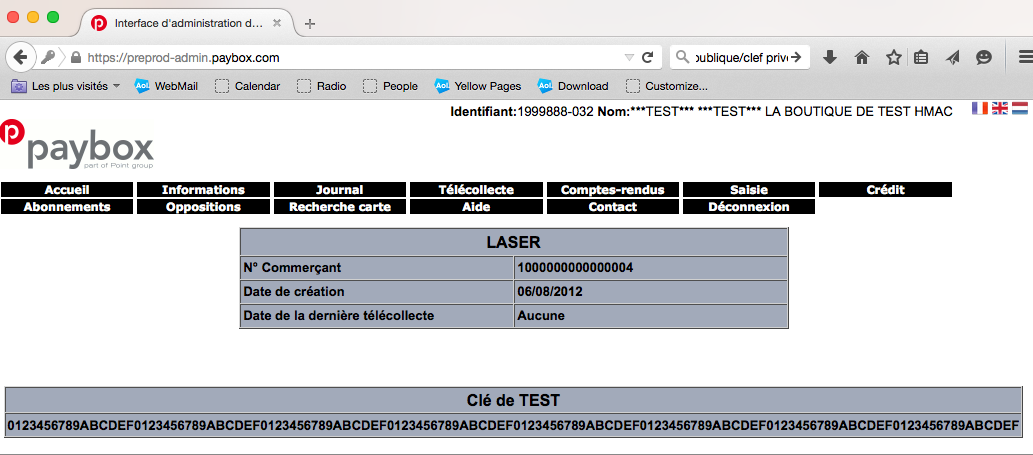
Après vous être connecté au Back-office de votre boutique, il faut saisir le login et mot de passe. Pour mémoire voici les URL des site de pré-production et de production :

* Le Back-office de test : **https://preprod-admin.paybox.com**
* Le Back-office de production : **https://admin.paybox.com**

Le login et mot de passe sont ceux fourni à l’ouverture de la boutique par Paybox. Dans le cas de la boutique de test mutualisée :

* Login : 199988832
* Mot de passe : 1999888I

Sélectionnez l’onglet « Informations », et récupérez la clé publique située en bas de cet écran :



#### Génération de la clé privée ou « empreinte » de la transaction

Avec cette clé publique, il faut générer l’empreinte des informations transmises.

Voici un exemple de programme PHP de génération de cette empreinte puis d’envoi via le formulaire :

< ?php

// --------------------------------------

// On récupère la date au format ISO-8601

// --------------------------------------

$DateTime = date("c");

 // --------------------------------------

// On crée la chaîne $param à hacher

// --------------------------------------

// l’ordre des variables doit être identique a celui du formulaire

// La chaîne est de la forme (sans aucun espace):

// PBX\_SITE=1999888&PBX\_RANG=32&PBX\_IDENTIFIANT=110647233&PBX\_TOTAL=1000&...

$param = "PBX\_SITE=1999888".

 "&PBX\_RANG=32".

"&PBX\_IDENTIFIANT=110647233".

"&PBX\_TOTAL=".$\_POST['montant'].

"&PBX\_DEVISE=978".

"&PBX\_CMD=".$\_POST['ref'].

"&PBX\_PORTEUR=".$\_POST['email'].

"&PBX\_RETOUR=Mt:M;Ref:R;Auto:A;Erreur:E".

"&PBX\_HASH=SHA512".

"&PBX\_TIME=".$DateTime.

"&PBX\_EFFECTUE=http://serv.fr/retour.php".

"&PBX\_REFUSE=http://serv.fr/retour.php".

"&PBX\_ANNULE=http://serv.fr/retour.php";

// --------------------------------------

// On récupère la clé publique (onglet Information du Back-office)

// --------------------------------------

// que l’on renseigne dans la variable $keyTest;

$keyTest="0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF";

// Si la clé est en ASCII, On la transforme en binaire

$binKey = pack("H\*", $keyTest);

// --------------------------------------

// On calcule l’empreinte (à renseigner dans le paramètre PBX\_HMAC)

// --------------------------------------

// grâce à la fonction hash\_hmac et la clé binaire

// On envoie via la variable PBX\_HASH l'algorithme de hachage qui a été

// utilisé (SHA512 dans ce cas)

// Pour information : Pour afficher la liste des algorithmes disponibles

// sur votre environnement, décommentez la ligne suivante

// print\_r(hash\_algos());

$empreinte\_carte = strtoupper(hash\_hmac('sha512', $param, $binKey));

//  La chaîne sera envoyée en majuscules, d'où l'utilisation de strtoupper()

//  On crée le formulaire à envoyer à Paybox System

//  ATTENTION : l'ordre des champs est extrêmement important, il doit

//  correspondre exactement à l'ordre des champs dans la chaîne hachée

// --------------------------------------

// on envoi les informations via le formulaire

// --------------------------------------

?>

 <form method="POST" action="https://urlserveur.paybox.com/cgi/MYchoix\_pagepaiement.cgi">

<input type="hidden" name="PBX\_SITE" value="1999888">

<input type="hidden" name="PBX\_RANG" value="32">

 <input type="hidden" name="PBX\_IDENTIFIANT" value="110647233">

 <input type="hidden" name="PBX\_TOTAL" value="<? echo $\_POST['montant']; ?>">

<input type="hidden" name="PBX\_DEVISE" value="978">

 <input type="hidden" name="PBX\_CMD" value="<? echo $\_POST['ref']; ?>">

 <input type="hidden" name="PBX\_PORTEUR" value="<? echo $\_POST['email']; ?>">

 <input type="hidden" name="PBX\_RETOUR" value="Mt:M;Ref:R;Auto:A;Erreur:E">

<input type="hidden" name="PBX\_HASH" value="SHA512">

 <input type="hidden" name="PBX\_HMAC" value="<? echo $empreinte\_carte; ?>">

 <input type="hidden" name="PBX\_TIME" value="<? echo $DateTime; ?>">

<input type="hidden" name="PBX\_EFFECTUE" value="http://serv.fr/retour.php">

<input type="hidden" name="PBX\_REFUSE" value="http://serv.fr/retour.php">

<input type="hidden" name="PBX\_ANNULE" value="http://serv.fr/retour.php">

<input type="submit" value="Envoyer">

</form>

## Mise en œuvre de la solution

Cette section présente la mise en œuvre technique du paiement via la boutique Paybox.

Nous montrons comment faire un règlement en une ou plusieurs fois, et présentons également les modifications à apporter dans le cas d’une utilisation avec le système 3D Secure.

Seuls le **site marchand et l’intégration du formulaire d’appel au site Paybox**, puis **l’interprétation du retour du paiement** sont présentés.

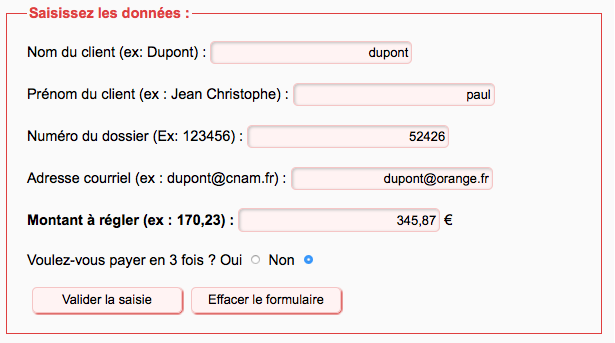
En effet, l’écran de saisie des informations bancaires et le Back-office Paybox sont totalement générés par Paybox et ne nécessite aucun développement.

### Paiement en une ou plusieurs fois sans 3D Secure

#### Intégration dans le site marchand

##### Formulaire de saisie des informations

L’écran suivant correspond au programme saisie\_client.html.



Il présente une simulation du site marchand. Il s’agit d’un formulaire de saisie des informations nécessaire au paiement.

Dans le cas d’un vrai site marchand, le numéro de dossier ou le montant à régler ne seront pas demandé à l’acheteur mais calculé.

Le bouton radio propose le paiement en 3 fois (Paybox autorise jusqu’à 4 échéances, y compris la première qui est réglée immédiatement).

<!DOCTYPE html>

<html>

<head> <!-- Entête HTML -->

<meta charset="utf-8" />

<title>Paiement Paybox</title>

<link href="CSS/saisie\_client\_paybox.css" rel="stylesheet" type="text/css" />

</head>

<form action="paybox\_clef\_HMAC.php" method="post">

<fieldset>

<legend>Saisissez les donn&eacute;es :</legend><br/>

Nom du client (ex: Dupont) : <input type="text" name="Nom" size="30" /><br/><br/>

Pr&eacute;nom du client (ex : Jean Christophe) : <input type="text" name="Prenom" size="30" /><br/><br/>

Num&eacute;ro du dossier (Ex: 123456) : <input type="text" name="Dossier" size="30" /><br/><br/>

Adresse courriel (ex : dupont@cnam.fr) : <input type="text" name="Courriel" size="30" /><br/><br/>

<b>Montant à régler (ex : 170,23) : </b><input type="text" name="Montant\_Total" size="30" /> &euro;<br/><br/>

Voulez-vous payer en 3 fois ?

Oui <input type="radio" name="PaieMult" value="oui">

Non <input type="radio" name="PaieMult" value="non" checked="checked"> <br/><br/>

<input type="submit" name="valider" value="Valider la saisie" />

<input type="reset" name="effacer" value="Effacer le formulaire" />

</fieldset>

</form>

</body>

</html>

##### Programme d’envoi des informations à Paybox

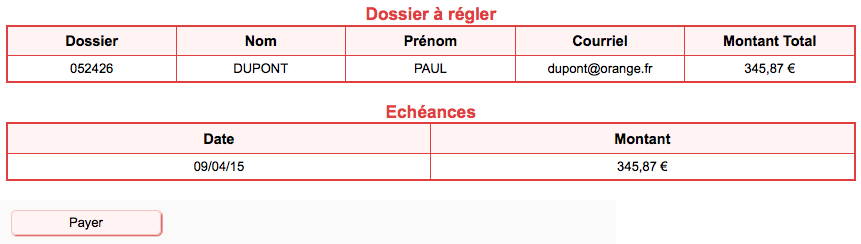
Après validation du formulaire précédente le programme paybox\_clef\_HMAC.php est exécuté et reçoit les données transmises.

Il effectue des traitements validant et mettant en forme les différents champs, comme :

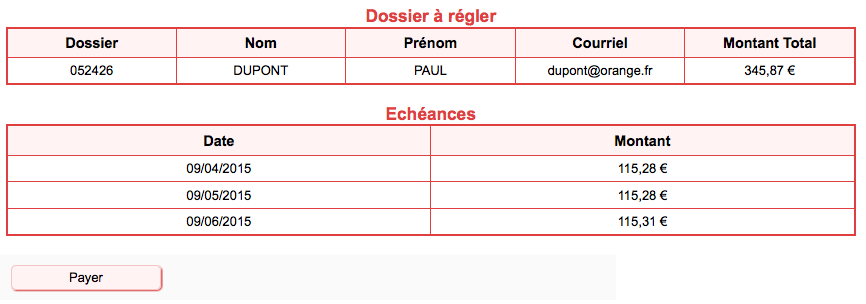
* Le codage du dossier sur 6 caractères ;
* La normalisation des noms et prénoms en majuscules et sans accent ;
* La mise en forme du montant à régler au format français pour l’affichage ;
* Etc.

Puis il récapitule les différents éléments du dossier.

L’écran suivant récapitule un règlement en une seule fois



L’écran suivant récapitule un règlement en une trois fois (dans le cas d’une sélection du bouton radio sur « oui » dans l’écran précédent), pour le même montant.



Le clic sur le bouton « Payer » envoie les informations à Paybox.

Voici le programme paybox\_clef\_HMAC.php. Il utilise des fonctions contenues dans le fichier fonction\_include\_sprog.php.

<!DOCTYPE html>

<html>

<head> <!-- Entête HTML -->

<meta charset="utf-8" />

<title>Paiement Paybox</title>

<link href="CSS/saisie\_client\_paybox.css" rel="stylesheet" type="text/css" />

</head>

<?php

include 'INCLUDE/fonction\_include\_sprog.php';

// -----------------------------------------------------------------

// Récupération des données saisies

// -----------------------------------------------------------------

if (isset($\_POST['Nom'])) $Nom=$\_POST['Nom'] ;

else $Nom = '' ;

if (isset($\_POST['Prenom'])) $Prenom=$\_POST['Prenom'] ;

else $Prenom = '' ;

if (isset($\_POST['Dossier'])) $Dossier=$\_POST['Dossier'];

else $Dossier='';

if (isset($\_POST['Courriel'])) $Courriel=$\_POST['Courriel'];

else $Courriel='';

if (isset($\_POST['Montant\_Total'])) $Montant\_Total=$\_POST['Montant\_Total'];

else $Montant\_Total = '' ;

if (isset($\_POST['PaieMult'])) $PaieMult = $\_POST['PaieMult'] ; else $PaieMult = '' ;

$tab\_echeances=array();

// -----------------------------------------------------------------

// Traitement des données saisies

// -----------------------------------------------------------------

$tab\_validation\_donnees=validation\_donnees($Nom,$Prenom,$Dossier,$Courriel);

if (count($tab\_validation\_donnees) == 0)

{

?>

<fieldset>

<legend>Erreur :</legend><br/>

<b>Un des champs saisis est erroné ! <br /><br/>

</fieldset>

<?php

}

else

{

list($Dossier,$Nom,$Prenom,$Courriel)=

validation\_donnees($Nom,$Prenom,$Dossier,$Courriel);

// --- conversion du Montant\_Total au format réel ---

$Montant\_Total = str\_replace(",",".",$Montant\_Total);

$Montant\_Total=round(floatval($Montant\_Total),2);

// --- préparation de Montant\_Total\_formate pour l'affichage ---

$Montant\_Total\_formate=number\_format($Montant\_Total,2,","," ")." &euro;";

$tab\_validation\_donnees['Montant\_Total']=$Montant\_Total\_formate;

// ------------------------------

// -- récupération de la date ---

// ------------------------------

// --- On récupère la date au format ISO-8601:2015-04-03T15:27:18+02:00 ---

date\_default\_timezone\_set('Europe/Paris');

$DateTime = date("c");

// ----------------------------------------------------------------------

// -- Calcul des montants selon le paiement immédiat ou en 3 fois ---

// ----------------------------------------------------------------------

if ($PaieMult == "non") // --- paiement immédiat ---

{

// --- Préfixe de la référence pour différencier les modes de paiement ---

// --- au comptant : CPT

// --- en plusieurs fois : ABT ---

$Mode\_Paiement="CPT";

// --- conversion du Montant\_Total en centimes ---

$Montant\_Total\_Centimes=$Montant\_Total\*100;

$PARAM['paiement\_immediat']=$Montant\_Total\_Centimes;

$Date\_Ech1 = date('d/m/y');

$tab\_echeances[0]=$Date\_Ech1;

$tab\_echeances[1]=number\_format($Montant\_Total,2,","," ")." &euro;";

}

else // paiement en 3 fois

{

// --- Préfixe de la référence pour différencier les modes de paiement ---

// --- au comptant : CPT

// --- en plusieurs fois : ABT ---

$Mode\_Paiement="ABT";

// --- Chaque paiement est correspond à 33,33% du total ---

$Montant\_Ech1=round((($Montant\_Total\*33.33)/100),2);

$Montant\_Ech2=round((($Montant\_Total\*33.33)/100),2);

$Montant\_Ech3=round($Montant\_Total-$Montant\_Ech1-$Montant\_Ech2,2);

// --- conversion des Montants en centimes ---

$Montant\_Ech1\_centimes=$Montant\_Ech1\*100;

$Montant\_Ech2\_centimes=$Montant\_Ech2\*100;

$Montant\_Ech3\_centimes=$Montant\_Ech3\*100;

// --- Mise à jour du tableau des paramètres ---

$PARAM['paiement\_immediat']=$Montant\_Ech1\_centimes;

$PARAM['paiement\_echeance2']=$Montant\_Ech2\_centimes;

$PARAM['paiement\_echeance3']=$Montant\_Ech3\_centimes;

// ---Calcul des dates de règlement ---

// Echéance 1 : immédiatement

// Echéance 2 : Echéance 1 + 1 mois

// Echéance 3 : Echéance 1 + 2 mois

$Date\_Ech1 = date('d/m/Y');

$Date\_Ech2 = date('d/m/Y', strtotime('+1 month'));

$Date\_Ech3 = date('d/m/Y', strtotime('+2 month'));

// --- Mise à jour du tableau des paramètres ---

$PARAM['date\_echeance2']=$Date\_Ech2;

$PARAM['date\_echeance3']=$Date\_Ech3;

// --- préparation des montants pour affichage ---

$tab\_echeances[0]=$Date\_Ech1;

$tab\_echeances[1]=number\_format($Montant\_Ech1,2,","," ")." &euro;";

$tab\_echeances[2]=$Date\_Ech2;

$tab\_echeances[3]=number\_format($Montant\_Ech2,2,","," ")." &euro;";

$tab\_echeances[4]=$Date\_Ech3;

$tab\_echeances[5]=number\_format($Montant\_Ech3,2,","," ")." &euro;";

}

// -----------------------------------

// -- Affichage du dossier complet ---

// -----------------------------------

$tab\_dossier=array(

'Dossier'=>$Dossier,

'Nom'=>$Nom,

'Prenom'=>$Prenom,

'Courriel'=>$Courriel,

'Montant\_Total'=>$Montant\_Total\_formate

);

affichage\_dossier("Dossier à régler",$tab\_dossier);

echo WEB\_EOL;

// -----------------------------------

// -- Affichage des échéances ---

// -----------------------------------

affichage\_echeances("Ech&eacute;ances",$tab\_echeances);

echo WEB\_EOL;

// -----------------------------------------

// -- Tableau des paramètres pour Paybox ---

// -----------------------------------------

// --- URL des serveurs Paybox System classique ---

$PARAM['URL\_SERVEUR\_PREPROD']= "https://preprod-tpeweb.paybox.com/cgi/MYchoix\_pagepaiement.cgi";

$PARAM['URL\_SERVEUR\_PROD\_PRINCIPAL']= "https://tpeweb.paybox.com/cgi/MYchoix\_pagepaiement.cgi";

$PARAM['URL\_SERVEUR\_PROD\_SECOURS'] = "https://tpeweb1.paybox.com/cgi/MYchoix\_pagepaiement.cgi";

// --- serveur utilisé pour ce programme ---

$PARAM['URL\_SERVEUR\_UTILISE'] = $PARAM['URL\_SERVEUR\_PREPROD'] ;

// --- Paramètres de la boutique Paybox System classique ---

// --- ici celle de test mutualisée fournie par PAYBOX ---

// --- pour activer (désactiver) cette boutique ---

// --- décommenter(commenter) les 3 lignes pour SITE,RANG et IDENTIFIANT ---

$PARAM['SITE']="1999888";

$PARAM['RANG']="32";

$PARAM['IDENTIFIANT']="110647233";

// --- Pour information : accès Backoffice Paybox System classique :

// URL : https://preprod-admin.paybox.com

// LOGIN : 199988832

// Mot\_de\_Passe : 1999888I

// --- Paramètres de la boutique Paybox System 3D-Secure ---

// --- ici celle de test mutualisée fournie par PAYBOX ---

// --- pour activer (désactiver) cette boutique ---

// --- décommenter(commenter) les 3 lignes pour SITE,RANG et IDENTIFIANT ---

//$PARAM['SITE']="1999888";

//$PARAM['RANG']="43";

//$PARAM['IDENTIFIANT']="107975626";

// --- accès Backoffice Paybox System 3D-Secure :

// URL : https://preprod-admin.paybox.com

// LOGIN : 199988843

// Mot\_de\_Passe : 1999888I

// --- Code de la devise suivant la norme ISO 4217 : ---

// --- euro: 978 ---

// --- dollar américain: 840 ---

// --- Franc CFA BCEAO: 952 ---

// --- Franc CFA BEAC: 950 ---

// --- Franc suisse: 756 ---

// --- Livre Sterling: 826 ---

$PARAM['PBX\_DEVISE']="978";

// --- Référence unique du dossier ---

// --- de la forme CPT\_052426\_DUPONT\_PAUL\_2015-04-03T15:27:18+02:00

$PARAM['ref']=$Mode\_Paiement."\_".$Dossier."\_".$Nom."\_".$Prenom."\_".

$DateTime;

// --- Adresse email du client qui recevra le reçu de la carte visa ---

$PARAM['email']=$Courriel;

// --- Liste des informations retournée par Paybox ---

// E = Code d'erreur

// M = Montant de la transaction

// R = Référence de la transaction

// A = Code de l'autorisation

// T = Numéro d'appel Paybox

// S = Numéro de transaction

// Q = Heure de transaction

// W = Date de transaction

// D = Date fin de validité de la carte

// N = 6 premiers chiffres du numéro de la carte

// J = 2 derniers chiffres du numéro de la carte

// C = type de carte

// H = empreinte de la carte

// I = Pays de l'internaute

// Y = Pays de la banque émettrice de la carte

// G = Garantie 3D Secure

// P = Type de paiement

// B = Numéro d'abonnement

$PARAM['RETOUR']="erreur:E;mtant:M;ref:R;auto:A;appel:T;ntrans:S;htrans:Q;dtrans:W;dfin:D;ndcarte:N;nfcarte:J;tcarte:C;epcarte:H;paysi:I;paysc:Y;gar:G;tpt:P;nab:B";

// --- Algorithme de codage ---

$PARAM['ALG\_CRYP']="SHA512";

// -----------------------------------------------------

// --- URL de retour pour traitement des données : ---

// --- celle de votre serveur qui traitera les informations ---

// --- retournées par PAYBOX ---

$PARAM['URL\_RETOUR\_UNIQUE'] =

"http://votre\_serveur.cnam.fr/PayBox/traitement\_retour\_paybox.php";

// --- URL de retour via le navigateur du client --

$PARAM['URL\_RETOUR\_EFFECTUE'] = $PARAM['URL\_RETOUR\_UNIQUE'];

$PARAM['URL\_RETOUR\_REFUSE'] = $PARAM['URL\_RETOUR\_UNIQUE'];

$PARAM['URL\_RETOUR\_ANNULE'] = $PARAM['URL\_RETOUR\_UNIQUE'];

$PARAM['URL\_RETOUR\_ATTENTE'] = $PARAM['URL\_RETOUR\_UNIQUE'];

// --- URL de retour DIRECT de serveur à serveur (url IPN)

// --- donc sans passer par le navigateur du client --

$PARAM['URL\_RETOUR\_REPONDRE\_A'] = $PARAM['URL\_RETOUR\_UNIQUE'];

// --- Méthode de retour GET ou POST --

$PARAM['METHODE\_RETOUR'] = "POST";

// -- Préparation calcul de la clef privée (empreinte du message) ---

// .........................................................

// -1- On crée la chaîne à hacher sans URL : encodage ---

// .........................................................

// l’ordre des variables doit être identique a celui du formulaire

// La chaîne est de la forme (sans aucun espace):

// PBX\_SITE=1999888&PBX\_RANG=32&PBX\_IDENTIFIANT=110647233&PBX\_TOTAL=1000&...

$param\_obl = "PBX\_SITE=".$PARAM['SITE'].

"&PBX\_RANG=".$PARAM['RANG'].

"&PBX\_IDENTIFIANT=".$PARAM['IDENTIFIANT'].

"&PBX\_TOTAL=".$PARAM['paiement\_immediat'].

"&PBX\_DEVISE=".$PARAM['PBX\_DEVISE'].

"&PBX\_CMD=".$PARAM['ref'].

"&PBX\_PORTEUR=".$PARAM['email'].

"&PBX\_RETOUR=".$PARAM['RETOUR'].

"&PBX\_HASH=".$PARAM['ALG\_CRYP'].

"&PBX\_TIME=".$DateTime;

if ($PaieMult == "oui")

{

$param\_echeances="&PBX\_2MONT1=".$PARAM['paiement\_echeance2'].

"&PBX\_DATE1=".$PARAM['date\_echeance2'].

"&PBX\_2MONT2=".$PARAM['paiement\_echeance3'].

"&PBX\_DATE2=".$PARAM['date\_echeance3'];

}

// --- dans le cas d'un retour via le navigateur de l'acheteur ---

// --- il faut utiliser ces trois paramètres : ---

// --- PBX\_EFFECTUE, PBX\_REFUSE et PBX\_ANNULE

$param\_retour="&PBX\_EFFECTUE=".$PARAM['URL\_RETOUR\_EFFECTUE'].

"&PBX\_REFUSE=".$PARAM['URL\_RETOUR\_REFUSE'].

"&PBX\_ANNULE=".$PARAM['URL\_RETOUR\_ANNULE'];

// --- dans le cas d'un retour DIRECT de serveur à serveur (url IPN) ---

// --- donc sans passer par le navigateur du client --

// --- il faut utiliser ces deux paramètres à la place des ---

// --- trois paramètres : PBX\_EFFECTUE, PBX\_REFUSE et PBX\_ANNULE

//$param\_retour="&PBX\_REPONDRE\_A=".$PARAM['URL\_RETOUR\_REPONDRE\_A'].

//"&PBX\_RUF1=".$PARAM['METHODE\_RETOUR'];

// --- construction finale de la chaine $param pour laquelle ---

// --- il faut générer une empreinte ---

if ($PaieMult == "oui") // --- paiement en 3 fois ---

{

$param=$param\_obl.$param\_echeances.$param\_retour;

}

else // --- paiement immédiat ---

{

$param=$param\_obl.$param\_retour;

}

// ..................................................................

// -2- Génération de l'empreinte (clef pive) de la chaine $param ---

// ..................................................................

// --- On récupère la clé secrète HMAC

// --- (peut aussi être stockée dans une base de données par exemple)

// --- que l’on renseigne dans la variable

// --- $keyTest = c'est la clef publique de Paybox;

// --- Si la clé est en ASCII, On la transforme en binaire

$keyTest="0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF0123456789ABCDEF";

$binKey = pack("H\*", $keyTest);

// --- On calcule l’empreinte (à renseigner dans le paramètre PBX\_HMAC)

// --- grâce à la fonction hash\_hmac et la clé binaire

// --- On envoie via la variable PBX\_HASH l'algorithme de hachage

// --- qui a été utilisé (SHA512 dans ce cas)

// --- Pour afficher la liste des algorithmes disponibles

// --- sur votre environnement,

// --- décommentez la ligne suivante

// print\_r(hash\_algos());

// ......................................................

$algo\_cryptage=strtolower($PARAM['ALG\_CRYP']);

$empreinte\_carte = strtoupper(hash\_hmac($algo\_cryptage, $param, $binKey));

// ......................................................

// --- La chaîne sera envoyée en majuscules, d'où l'utilisation

// --- de strtoupper()

// --- On crée le formulaire à envoyer à Paybox System

// --- ATTENTION : l'ordre des champs est extrêmement important, il doit

// --- être exactement identique à celui de la chaîne hachée : $param

// ----------------------------------------------

// -- Formulaire d'envoi des données à Paybox ---

// ----------------------------------------------

?>

<form method="POST" action="<? echo $PARAM['URL\_SERVEUR\_UTILISE']; ?>">

<!-- param\_obl -->

<input type="hidden" name="PBX\_SITE" value="<? echo $PARAM['SITE']; ?>">

<input type="hidden" name="PBX\_RANG" value="<? echo $PARAM['RANG']; ?>">

<input type="hidden" name="PBX\_IDENTIFIANT" value="<? echo $PARAM['IDENTIFIANT']; ?>">

<input type="hidden" name="PBX\_TOTAL" value="<? echo $PARAM['paiement\_immediat']; ?>">

<input type="hidden" name="PBX\_DEVISE" value="<? echo $PARAM['PBX\_DEVISE']; ?>">

<input type="hidden" name="PBX\_CMD" value="<? echo $PARAM['ref']; ?>">

<input type="hidden" name="PBX\_PORTEUR" value="<? echo $PARAM['email']; ?>">

<input type="hidden" name="PBX\_RETOUR" value="<? echo $PARAM['RETOUR']; ?>">

<input type="hidden" name="PBX\_HASH" value="<? echo $PARAM['ALG\_CRYP']; ?>">

<input type="hidden" name="PBX\_HMAC" value="<? echo $empreinte\_carte; ?>">

<input type="hidden" name="PBX\_TIME" value="<? echo $DateTime; ?>">

<?php

// --- traitement des échéances multiples ---

if ($PaieMult == "oui")

{

?>

<!-- param\_echeances -->

<input type="hidden" name="PBX\_2MONT1" value="<? echo $PARAM['paiement\_echeance2']; ?>">

<input type="hidden" name="PBX\_DATE1" value="<? echo $PARAM['date\_echeance2']; ?>">

<input type="hidden" name="PBX\_2MONT2" value="<? echo $PARAM['paiement\_echeance3']; ?>">

<input type="hidden" name="PBX\_DATE2" value="<? echo $PARAM['date\_echeance3']; ?>">

<?php

}

?>

<!-- URL de retour par navigateur-->

<input type="hidden" name="PBX\_EFFECTUE" value="<? echo $PARAM['URL\_RETOUR\_EFFECTUE']; ?>">

<input type="hidden" name="PBX\_REFUSE" value="<? echo $PARAM['URL\_RETOUR\_REFUSE']; ?>">

<input type="hidden" name="PBX\_ANNULE" value="<? echo $PARAM['URL\_RETOUR\_ANNULE']; ?>">

<!-- URL de retour DIRECT de serveur à serveur : url IPN -->

<!-- en commentaires

<input type="hidden" name="PBX\_REPONDRE\_A" value="<? echo $PARAM['URL\_RETOUR\_REPONDRE\_A']; ?>">

<input type="hidden" name="PBX\_RUF1" value="<? echo $PARAM['METHODE\_RETOUR']; ?>">

-->

<!-- Bouton Payer -->

<input type="submit" value="Payer">

<?php

}

?>

</form>

</body>

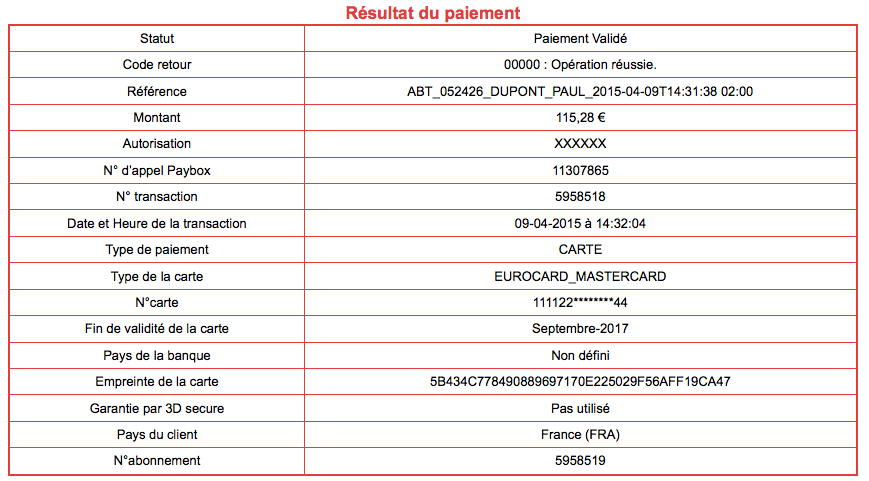
</html>

#### Interprétation du retour de Paybox

Le programme traitement\_retour\_paybox.php est indiqué dans les URLs de retour PBX\_EFFECTUE, PBX\_REFUSE et PBX\_ANNULE.

Ce programme interprète les informations transmises par Paybox après la saisie des informations bancaires.

La liste des informations que Paybox doit retourner au site marchand est indiquée dans la variable PBX\_RETOUR. Voici l’affichage produit par ce programme :



Voici le programme traitement\_retour\_paybox.php :

<!DOCTYPE html>

<html>

<head> <!-- Entête HTML -->

<meta charset="utf-8" />

<title>Retour Paiement Paybox</title>

<link href="CSS/saisie\_client\_paybox.css" rel="stylesheet" type="text/css" />

</head>

<?php

include 'INCLUDE/fonction\_include\_sprog.php';

setlocale(LC\_ALL, 'fr\_FR.UTF-8');

// ------------------------------------------------------

// --- récupération des différents éléments de retour ---

// ------------------------------------------------------

$code\_erreur = $\_GET['erreur'] ; // valeur PAYBOX E

$montant = $\_GET['mtant'] ; // valeur PAYBOX M

$reference = $\_GET['ref'] ; // valeur PAYBOX R

$autorisation = $\_GET['auto'] ; // valeur PAYBOX A

$num\_appel\_paybox = $\_GET['appel'] ; // valeur PAYBOX T

$numero\_transaction = $\_GET['ntrans'] ; // valeur PAYBOX S

$heure\_trans = $\_GET['htrans'] ; // valeur PAYBOX Q

$date\_trans = $\_GET['dtrans'] ; // valeur PAYBOX W

$date\_fin\_carte = $\_GET['dfin'] ; // valeur PAYBOX D

$numero\_debut\_carte = $\_GET['ndcarte'] ; // valeur PAYBOX N

$numero\_fin\_carte = $\_GET['nfcarte'] ; // valeur PAYBOX J

$type\_carte = $\_GET['tcarte'] ; // valeur PAYBOX C

$empreinte\_carte = $\_GET['epcarte'] ; // valeur PAYBOX H

$pays\_internaute = $\_GET['paysi'] ; // valeur PAYBOX I

$pays\_carte = $\_GET['paysc'] ; // valeur PAYBOX Y

$garantie\_3D\_secure = $\_GET['gar'] ; // valeur PAYBOX G

$type\_paiement = $\_GET['tpt'] ; // valeur PAYBOX P

$num\_abonnement = $\_GET['nab'] ; // valeur PAYBOX B

// -----------------------------------------

// --- traitement des valeurs retournées ---

// -----------------------------------------

// --- traitement des erreurs ---

// --- chargement du tableau des codes d'erreurs ---

$Tab\_Erreurs=chargement\_codes\_erreurs();

if ($code\_erreur == "00000") $msg\_erreur="Paiement Valid&eacute;";

elseif ($code\_erreur == "00001") $msg\_erreur="Paiement Annul&eacute;";

else $msg\_erreur="Paiement Refus&eacute;";

if (!empty($Tab\_Erreurs[$code\_erreur]))

$code\_erreur=$code\_erreur." : ".$Tab\_Erreurs[$code\_erreur];

// --- traitement du montant ---

if ($montant == 0)

$Montant\_formate="Aucune transaction";

else

$Montant\_formate=number\_format($montant/100,2,","," ")." &euro;";

// --- traitement de la date de validité de la carte ---

date\_default\_timezone\_set('Europe/Paris');

if (empty($date\_fin\_carte))

$date\_fin\_carte\_formatee="Non d&eacute;fini";

else

{

$timestamp\_en\_secondes=mktime(0,0,0,substr($date\_fin\_carte,2,2),date("d"),substr($date\_fin\_carte,0,2));

$date\_fin\_carte\_formatee=ucwords(strftime("%B-%Y",$timestamp\_en\_secondes));

}

// --- traitement du numéro de la carte ---

if (empty($numero\_debut\_carte))

$numero\_carte="Non d&eacute;fini";

else

$numero\_carte=$numero\_debut\_carte."\*\*\*\*\*\*\*\*".$numero\_fin\_carte;

// --- traitement des pays ---

// --- chargement du tableau des pays ---

$Tab\_Pays=chargement\_codes\_pays();

// --- Pays de l'adresse IP de l'internaute ---

if (!empty($Tab\_Pays[$pays\_internaute]))

$pays\_internaute=$Tab\_Pays[$pays\_internaute]['MIN']." ($pays\_internaute)";

else

$pays\_internaute="Non d&eacute;fini";

// --- Pays de la banque émettrice de la carte ---

if (!empty($Tab\_Pays[$pays\_carte]))

$pays\_carte=$Tab\_Pays[$pays\_carte]['MIN']." ($pays\_carte)";

else

$pays\_carte="Non d&eacute;fini";

// --- traitement de la garantie par 3D secure ---

if (empty($garantie\_3D\_secure))

$garantie\_3D\_secure="Pas utilisé";

elseif ($garantie\_3D\_secure == "O")

$garantie\_3D\_secure="Oui";

else

$garantie\_3D\_secure="Non";

// --- traitement de la date de la transaction ---

if ((empty($date\_trans)) && (empty($heure\_trans)))

$date\_heure\_transaction="Non d&eacute;fini";

else

$date\_heure\_transaction=substr($date\_trans,0,2)."-".substr($date\_trans,2,2)."-".substr($date\_trans,4,4)." à ".$heure\_trans;

// --- traitement de l'empreinte de la carte ---

if (empty($empreinte\_carte))

$empreinte\_carte="Non d&eacute;fini";

// --- traitement du type de la carte ---

if (empty($type\_carte))

$type\_carte="Non d&eacute;fini";

// --- traitement de l'autorisation ---

if (empty($autorisation))

$autorisation="Non d&eacute;fini";

// --- traitement du numéro de la transaction ---

if ($numero\_transaction == 0)

$numero\_transaction="Aucune transaction";

// --- traitement du numéro d'appel Paybox ---

if ($num\_appel\_paybox == 0)

$num\_appel\_paybox="Aucune transaction";

// --- traitement du type de carte ---

if (empty($type\_carte))

$type\_carte="Non d&eacute;fini";

// --- traitement du type de carte ---

if (empty($num\_abonnement))

$num\_abonnement="Non d&eacute;fini";

// ----------------------------------------------------

// --- Affichage du tableau du résultat du paiement ---

// ----------------------------------------------------

?>

<table summary="R&eacute;sultat du paiement">

<caption>R&eacute;sultat du paiement</caption>

<?php

echo "<tr><td>Statut</td><td>$msg\_erreur</td></tr>";

echo "<tr><td>Code retour</td><td>$code\_erreur</td></tr>";

echo "<tr><td>Référence</td><td>$reference</td></tr>";

echo "<tr><td>Montant</td><td>$Montant\_formate</td></tr>";

echo "<tr><td>Autorisation</td><td>$autorisation</td></tr>";

echo "<tr><td>N&deg; d’appel Paybox</td><td>$num\_appel\_paybox</td></tr>";

echo "<tr><td>N&deg; transaction</td><td>$numero\_transaction</td></tr>";

echo "<tr><td>Date et Heure de la transaction</td><td>$date\_heure\_transaction</td></tr>";

echo "<tr><td>Type de paiement</td><td>$type\_paiement</td></tr>";

echo "<tr><td>Type de la carte</td><td>$type\_carte</td></tr>";

echo "<tr><td>N&deg;carte</td><td>$numero\_carte</td></tr>";

echo "<tr><td>Fin de validit&eacute; de la carte</td><td>$date\_fin\_carte\_formatee</td></tr>";

echo "<tr><td>Pays de la banque</td><td>$pays\_carte</td></tr>";

echo "<tr><td>Empreinte de la carte</td><td>$empreinte\_carte</td></tr>";

echo "<tr><td>Garantie par 3D secure</td><td>$garantie\_3D\_secure</td></tr>";

echo "<tr><td>Pays du client</td><td>$pays\_internaute</td></tr>";

echo "<tr><td>N&deg;abonnement</td><td>$num\_abonnement</td></tr>";

?>

</table>

</body>

</html>

#### Fichiers annexes

##### La feuille de style

Voici la feuille de style saisie\_client\_paybox.css :

/\* Par défaut à tous les éléments de la page \*/

\* {color:black;

font-family:"Arial" ;

text-align:left;

font-size:100%;

}

/\* ================================ \*/

/\* === style pour le formulaire === \*/

/\* ================================ \*/

/\* couleur des boutons submit et reset quand on les survole \*/

input[type=submit]:hover, input[type=reset]:hover {

background-color:#FCDEDE;

}

/\* couleur des boutons submit et reset actif \*/

input[type=submit]:active, input[type=reset]:active {

background-color:#FCDEDE;

box-shadow:1px 1px 1px #D83F3D inset;

}

/\* couleur des champs de saisie quand on clique dedans \*/

input:focus, textarea:focus {

background-color:white;

}

input[type=submit]:focus, input[type=reset]:focus {

background-color:#FFFFF3;

}

body {

font-family:Arial;

font-size:90%;

}

form {

background-color:#FAFAFA;

padding:10px;

width:600px;

/\* width:60%; \*/

}

label {

margin-top:10px;

/\* display:block;\*/

}

label.inline {

display:inline;

margin-right:50px;

}

input, textarea, select, option {

background-color:#FFF3F3;

}

input, textarea, select {

padding:3px;

border:1px solid #F5C5C5;

border-radius:5px;

/\*width:70px;\*/

box-shadow:1px 1px 2px #C0C0C0 inset;

text-align:right;

font-size:90%;

}

select {

margin-top:10px;

}

input[type=radio] {

background-color:transparent;

border:none;

width:10px;

}

input[type=submit], input[type=reset] {

width:150px;

margin-left:5px;

box-shadow:1px 1px 1px #D83F3D;

cursor:pointer;

text-align:center;

}

/\* ================================ \*/

/\* === style pour le fieldset === \*/

/\* ================================ \*/

fieldset {

padding:0 20px 20px 20px;

margin-bottom:10px;

border:1px solid #DF3F3F;

}

legend {

color:#DF3F3F;

font-weight:bold

}

/\* ================================ \*/

/\* === style pour le tableau === \*/

/\* ================================ \*/

table {

border:2px solid #DF3F3F;

border-collapse:collapse;

/\*width:100%;\*/

width:850px;

/\*margin:auto;\*/

margin-left: 10px;

margin-right: auto;

}

thead, tfoot {

background-color:#D0E3FA;

border:1px solid #DF3F3F;

text-align:center;

}

tbody {

background-color:#FFFFFF;

border:1px solid #DF3F3F;

}

th {

font-family:Arial;

/\*border:1px dotted #D83F3D;\*/

border:1px solid #DF3F3F;

padding:5px;

background-color:#FFF3F3;

width:20%;

text-align:center;

}

td {

font-family:Arial;

font-size:90%;

border:1px solid #DF3F3F;

padding:5px;

/\*text-align:left;\*/

text-align:center;

}

caption {

font-family:Arial;

font-size:120%;

text-align:center;

color:#DF3F3F;

font-weight:bold

}

/\* ============================= \*/

/\* === style pour le menu === \*/

/\* ============================= \*/

div.menu {font-size:18px;}

p.menu {color:#DF3F3F; font-weight:bold}

a.menu {color:#DF3F3F;}

a.menu:link {color:#DF3F3F;background-color:transparent;text-decoration:none;}

a.menu:visited {color:#893232;background-color:transparent;text-decoration:none;}

a.menu:hover {color:#000000;background-color:#FCDEDE;text-decoration:none;}

a.menu:active {color:#FF0000;background-color:transparent;text-decoration:none;}

##### Les sous-programmes

Le fichier fonction\_include\_sprog.php contient les sous-programmes de validation, de mise en forme et de chargement de fichiers, qui sont utilisés dans les programmes PHP. Voici son contenu :

<?php

define("MAXDOSSIER","10000000");

define("WEB\_EOL","<br/>");

// ==========================================

// --- fonction de validation des données ---

// ===========================================

function validation\_donnees($Nom,$Prenom,$Dossier,$Courriel)

{

// --- Protection de l'injection HTML ---

$Nom = strip\_tags($Nom) ;

$Prenom = strip\_tags($Prenom) ;

$Dossier = strip\_tags($Dossier) ;

$Courriel= strip\_tags($Courriel);

// ---------------------------------------

// on vérifie qu'il n'y a aucun champ vide

// ---------------------------------------

if (!empty($Dossier) && !empty($Nom) && !empty($Prenom) && !empty($Courriel))

{

// ---------------------------------------

// on vérifie la validité des chaque champ

// ---------------------------------------

$retourValidationDossier = true ;

$retourValidationCourriel = true ;

$retourValidationNom = true ;

$retourValidationPrenom = true ;

if (!empty($Nom))

{

$Nom=normalisation\_nom($Nom) ;

$retourValidationNom=!empty($Nom);

if ($retourValidationNom)

{

$Nom = $Nom ;

}

}

if (!empty($Prenom))

{

$Prenom=normalisation\_nom($Prenom) ;

$retourValidationPrenom=!empty($Prenom);

if ($retourValidationPrenom)

{

$Prenom = $Prenom ;

}

}

if (!empty($Dossier))

{

$Dossier=normalisation\_numerique($Dossier) ;

$retourValidationDossier=validation\_Dossier($Dossier);

if ($retourValidationDossier)

{

$Dossier = intval($Dossier) ;

$Dossier=sprintf("%06d",$Dossier);

}

}

if (!empty($Courriel))

{

$retourValidationCourriel=validation\_Courriel($Courriel);

}

// -------------------------------

// si tous les champs sont valides

// -------------------------------

if (($retourValidationDossier) && ($retourValidationCourriel) && ($retourValidationNom) && ($retourValidationPrenom) )

{

$tab\_retour=array($Dossier,$Nom,$Prenom,$Courriel);

}

else

{

$tab\_retour=array();

}

}

else

{

$tab\_retour=array($Nom,$Prenom,$Dossier,$Courriel);

}

return $tab\_retour;

}

// =================================================

// --- fonction outil de suppression des accents ---

// =================================================

function supprime\_accent($chaine)

{

// tableau des caractères accentués à remplacer

$caracteres\_a\_remplacer = array('À', 'Á', 'Â', 'Ã', 'Ä', 'Å', 'Æ', 'Ç', 'È', 'É', 'Ê', 'Ë', 'Ì', 'Í', 'Î', 'Ï', 'Ð', 'Ñ', 'Ò', 'Ó', 'Ô', 'Õ', 'Ö', 'Ø', 'Ù', 'Ú', 'Û', 'Ü', 'Ý', 'ß', 'à', 'á', 'â', 'ã', 'ä', 'å', 'æ', 'ç', 'è', 'é', 'ê', 'ë', 'ì', 'í', 'î', 'ï', 'ñ', 'ò', 'ó', 'ô', 'õ', 'ö', 'ø', 'ù', 'ú', 'û', 'ü', 'ý', 'ÿ', 'Ā', 'ā', 'Ă', 'ă', 'Ą', 'ą', 'Ć', 'ć', 'Ĉ', 'ĉ', 'Ċ', 'ċ', 'Č', 'č', 'Ď', 'ď', 'Đ', 'đ', 'Ē', 'ē', 'Ĕ', 'ĕ', 'Ė', 'ė', 'Ę', 'ę', 'Ě', 'ě', 'Ĝ', 'ĝ', 'Ğ', 'ğ', 'Ġ', 'ġ', 'Ģ', 'ģ', 'Ĥ', 'ĥ', 'Ħ', 'ħ', 'Ĩ', 'ĩ', 'Ī', 'ī', 'Ĭ', 'ĭ', 'Į', 'į', 'İ', 'ı', 'Ĳ', 'ĳ', 'Ĵ', 'ĵ', 'Ķ', 'ķ', 'Ĺ', 'ĺ', 'Ļ', 'ļ', 'Ľ', 'ľ', 'Ŀ', 'ŀ', 'Ł', 'ł', 'Ń', 'ń', 'Ņ', 'ņ', 'Ň', 'ň', 'ŉ', 'Ō', 'ō', 'Ŏ', 'ŏ', 'Ő', 'ő', 'Œ', 'œ', 'Ŕ', 'ŕ', 'Ŗ', 'ŗ', 'Ř', 'ř', 'Ś', 'ś', 'Ŝ', 'ŝ', 'Ş', 'ş', 'Š', 'š', 'Ţ', 'ţ', 'Ť', 'ť', 'Ŧ', 'ŧ', 'Ũ', 'ũ', 'Ū', 'ū', 'Ŭ', 'ŭ', 'Ů', 'ů', 'Ű', 'ű', 'Ų', 'ų', 'Ŵ', 'ŵ', 'Ŷ', 'ŷ', 'Ÿ', 'Ź', 'ź', 'Ż', 'ż', 'Ž', 'ž', 'ſ', 'ƒ', 'Ơ', 'ơ', 'Ư', 'ư', 'Ǎ', 'ǎ', 'Ǐ', 'ǐ', 'Ǒ', 'ǒ', 'Ǔ', 'ǔ', 'Ǖ', 'ǖ', 'Ǘ', 'ǘ', 'Ǚ', 'ǚ', 'Ǜ', 'ǜ', 'Ǻ', 'ǻ', 'Ǽ', 'ǽ', 'Ǿ', 'ǿ');

// tableau des caractères sans accent de remplacement

$caracteres\_de\_remplacement = array('A', 'A', 'A', 'A', 'A', 'A', 'AE', 'C', 'E', 'E', 'E', 'E', 'I', 'I', 'I', 'I', 'D', 'N', 'O', 'O', 'O', 'O', 'O', 'O', 'U', 'U', 'U', 'U', 'Y', 's', 'a', 'a', 'a', 'a', 'a', 'a', 'ae', 'c', 'e', 'e', 'e', 'e', 'i', 'i', 'i', 'i', 'n', 'o', 'o', 'o', 'o', 'o', 'o', 'u', 'u', 'u', 'u', 'y', 'y', 'A', 'a', 'A', 'a', 'A', 'a', 'C', 'c', 'C', 'c', 'C', 'c', 'C', 'c', 'D', 'd', 'D', 'd', 'E', 'e', 'E', 'e', 'E', 'e', 'E', 'e', 'E', 'e', 'G', 'g', 'G', 'g', 'G', 'g', 'G', 'g', 'H', 'h', 'H', 'h', 'I', 'i', 'I', 'i', 'I', 'i', 'I', 'i', 'I', 'i', 'IJ', 'ij', 'J', 'j', 'K', 'k', 'L', 'l', 'L', 'l', 'L', 'l', 'L', 'l', 'l', 'l', 'N', 'n', 'N', 'n', 'N', 'n', 'n', 'O', 'o', 'O', 'o', 'O', 'o', 'OE', 'oe', 'R', 'r', 'R', 'r', 'R', 'r', 'S', 's', 'S', 's', 'S', 's', 'S', 's', 'T', 't', 'T', 't', 'T', 't', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u', 'W', 'w', 'Y', 'y', 'Y', 'Z', 'z', 'Z', 'z', 'Z', 'z', 's', 'f', 'O', 'o', 'U', 'u', 'A', 'a', 'I', 'i', 'O', 'o', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u', 'U', 'u', 'A', 'a', 'AE', 'ae', 'O', 'o');

// retour de la fonction

return str\_replace($caracteres\_a\_remplacer, $caracteres\_de\_remplacement, $chaine);

}

// =================================================

// --- fonction outil de normalisation des noms ---

// =================================================

function normalisation\_nom($chaine)

{

// tableau des motifs de recherche

$tab\_motif=array('/[^a-zA-Z -]/', '/[ -]+/', '/^-|-$/');

// tableau des caract&egrave;res de remplacement

$tab\_remplacement=array('', '-', '') ;

// chaine de caract&egrave;res sur laquelle s'effectue le remplacement

$chaine\_contexte=supprime\_accent($chaine);

// retour de la fonction

return strtoupper(preg\_replace($tab\_motif,$tab\_remplacement, $chaine\_contexte));

}

// =================================================

// --- fonction outil de normalisation des numériques ---

// =================================================

function normalisation\_numerique($numero)

{

// tableau des motifs de recherche

$tab\_motif=array('/[^0-9,.]/') ;

// tableau des caract&egrave;res de remplacement

$tab\_remplacement=array('') ;

// retour de la fonction

$numero\_retour=intval(preg\_replace($tab\_motif,$tab\_remplacement, $numero));

return $numero\_retour;

}

// ==============================

// --- Validation du champ Dossier ---

// ==============================

function validation\_Dossier($Dossier)

{

$erreur=false;

if (($Dossier == 0) || (!(($Dossier >0)&&($Dossier <MAXDOSSIER))))

{

$erreur=true;

}

return !$erreur;

}

// ====================================

// --- Validation du champ Courriel ---

// ====================================

function validation\_Courriel($Courriel)

{

$resultat=filter\_var($Courriel,FILTER\_VALIDATE\_EMAIL);

if (empty($resultat)) $resultat=false; else $resultat=true;

return $resultat;

}

// ================================================

// --- fonction outil d'affichage d'un dossier ---

// ================================================

function affichage\_dossier($titre,$tab\_dossier)

{

// entête de l'affichage

?>

<table summary="Dossier">

<caption><?php echo $titre;?></caption>

<thead>

<tr>

<!-- entête du tableau -->

<th>Dossier</th>

<th>Nom</th>

<th>Pr&eacute;nom</th>

<th>Courriel</th>

<th>Montant Total</th>

</tr>

</thead>

<?php

if (count($tab\_dossier) ==0)

{

echo "<td colspan=\"5\"><b>Aucune donn&eacute;e &agrave; afficher</b></td>";

}

else

{

// importation des variables à partir de l'étiquette des champs

extract($tab\_dossier,EXTR\_OVERWRITE);

echo "<tr>";

echo "<td>$Dossier</td><td>$Nom</td><td>$Prenom</td><td>$Courriel</td><td>$Montant\_Total</td>";

echo "</tr>";

}

?></table><?php

}

// ==================================================

// --- fonction outil d'affichage des échéances ---

// ==================================================

function affichage\_echeances($titre,$tab\_echeances)

{

// entête de l'affichage

?>

<table summary="Ech&eacute;ances">

<caption><?php echo $titre;?></caption>

<thead>

<tr>

<!-- entête du tableau -->

<th>Date</th>

<th>Montant</th>

</tr>

</thead>

<?php

$Nb\_Echeances=count($tab\_echeances);

if ($Nb\_Echeances == 0)

{

echo "<td colspan=\"2\"><b>Aucune donn&eacute;e &agrave; afficher</b></td>";

}

else

{

for ($i=0; $i<$Nb\_Echeances;$i+=2)

{

$date=$tab\_echeances[$i];

$montant=$tab\_echeances[$i+1];

echo "<tr><td>$date</td><td>$montant</td></tr>";

}

}

?></table><?php

}

// ======================================================

// --- fonction outil chargement des codes d'erreurs ---

// ======================================================

function chargement\_codes\_erreurs()

{

$Tab\_Erreurs=array();

$NomFichier="FICHIERS/Liste\_Codes\_Erreurs.txt";

$f1 = @fopen($NomFichier, "rt");

if (! $f1) // --- erreur d'ouverture ---

{ // -- on affiche les messages d'erreur ---

$tab\_erreurs = error\_get\_last() ;

$erreur\_type = $tab\_erreurs['type'] ;

$erreur\_message = $tab\_erreurs['message'];

$erreur\_programme = $tab\_erreurs['file'] ;

$erreur\_ligne = $tab\_erreurs['line'] ;

?>

<fieldset>

<legend>Erreur :</legend><br/>

<b>Chargement impossible :</b> erreur d'ouverture du fichier <b><?php echo $NomFichier; ?></b> en lecture ! <br /><br/>

<b>Type de l'erreur :</b> <?php echo $erreur\_type ; ?><br />

<b>Message :</b> <?php echo $erreur\_message; ?><br />

<b>Répertoire de Chargement :</b> <?php echo $repertoire\_courant."/".$repertoire\_chargement; ?><br /><br />

<b>Programme PHP :</b> <?php echo $erreur\_programme; ?><br />

<b>A la ligne :</b> <?php echo $erreur\_ligne; ?><br />

</fieldset>

<?php

}

else // -- on charge les données dans le tableau, et on les affiche ---

{

while ($Tab\_Lecture=fscanf($f1,"%s\t%s"))

{

list ($code\_erreur,$libelle\_erreur) = $Tab\_Lecture;

$libelle\_erreur=str\_replace("\_"," ",$libelle\_erreur);

if (strlen($code\_erreur) == 5)

{

$Tab\_Erreurs[$code\_erreur]=$libelle\_erreur;

}

}

}

return $Tab\_Erreurs;

}

// =========================================================

// --- fonction outil chargement des codes ISO des pays ---

// =========================================================

function chargement\_codes\_pays()

{

$Tab\_Erreurs=array();

$NomFichier="FICHIERS/Liste\_Pays\_ISO.txt";

$f1 = @fopen($NomFichier, "rt");

if (! $f1) // --- erreur d'ouverture ---

{ // -- on affiche les messages d'erreur ---

$tab\_erreurs=error\_get\_last();

$erreur\_type = $tab\_erreurs['type'] ;

$erreur\_message = $tab\_erreurs['message'];

$erreur\_programme = $tab\_erreurs['file'] ;

$erreur\_ligne = $tab\_erreurs['line'] ;

?>

<fieldset>

<legend>Erreur :</legend><br/>

<b>Chargement impossible :</b> erreur d'ouverture du fichier <b><?php echo $NomFichier; ?></b> en lecture ! <br /><br/>

<b>Type de l'erreur :</b> <?php echo $erreur\_type ; ?><br />

<b>Message :</b> <?php echo $erreur\_message; ?><br />

<b>Répertoire de Chargement :</b> <?php echo $repertoire\_courant."/".$repertoire\_chargement; ?><br /><br />

<b>Programme PHP :</b> <?php echo $erreur\_programme; ?><br />

<b>A la ligne :</b> <?php echo $erreur\_ligne; ?><br />

</fieldset>

<?php

}

else // -- on charge les données dans le tableau, et on les affiche ---

{

while ($Tab\_Lecture=fscanf($f1,"%s\t%s\t%s"))

{

list ($code\_pays,$libelle\_minuscules,$libelle\_majuscules) = $Tab\_Lecture;

$libelle\_minuscules=str\_replace("\_"," ",$libelle\_minuscules);

$libelle\_majuscules=str\_replace("\_"," ",$libelle\_majuscules);

if (strlen($code\_pays) == 3)

{

$Tab\_Pays[$code\_pays]['MIN']=$libelle\_minuscules;

$Tab\_Pays[$code\_pays]['MAJ']=$libelle\_majuscules;

}

}

}

return $Tab\_Pays;

}

?>

##### Le fichier d’interprétation des erreurs

Le fichier Liste\_Codes\_Erreurs.txt est utilisé par la fonction chargement\_codes\_erreurs() pour afficher le libellé de l’erreur, selon le code d’erreur retourné par Paybox.

00000 Opération\_réussie.

00001 La\_connexion\_au\_centre\_d’autorisation\_a\_échoué\_ou\_une\_erreur\_interne\_est\_survenue.

00003 Erreur\_Paybox.

00004 Numéro\_de\_porteur\_(carte)\_ou\_cryptogramme\_visuel\_invalide.

00006 Accès\_refusé\_ou\_site/rang/identifiant\_incorrect.

00008 Date\_de\_fin\_de\_validité\_incorrecte.

00009 Erreur\_de\_création\_d’un\_abonnement.

00010 Devise\_inconnue.

00011 Montant\_incorrect.

00015 Paiement\_déjà\_effectué.

00016 Abonné\_déjà\_existant\_(inscription\_nouvel\_abonné).

00021 Carte\_non\_autorisée.

00029 Carte\_non\_conforme.

00030 Temps\_d’attente\_>\_15\_mn\_par\_l’internaute

00031 Réservé

00032 Réservé

00033 Code\_pays\_de\_l’adresse\_IP\_du\_navigateur\_de\_l’acheteur\_non\_autorisé.

00040 Opération\_sans\_authentification\_3-DSecure,\_bloquée\_par\_le\_filtre.

99999 Opération\_en\_attente\_de\_validation\_par\_l’émetteur\_du\_moyen\_de\_paiement.

00100 Transaction\_approuvée\_ou\_traitée\_avec\_succès

00101 Contacter\_l’émetteur\_de\_carte

00102 Contacter\_l’émetteur\_de\_carte

00103 Commerçant\_invalide

00104 Conserver\_la\_carte

00105 Ne\_pas\_honorer

00107 Conserver\_la\_carte,\_conditions\_spéciales

00108 Approuver\_après\_identification\_du\_porteur\_(carte)

00112 Transaction\_invalide

00113 Montant\_invalide

00114 Numéro\_de\_porteur\_(carte)\_invalide

00115 Emetteur\_de\_carte\_inconnu

00117 Annulation\_client

00119 Répéter\_la\_transaction\_ultérieurement

00120 Réponse\_erronée\_(erreur\_dans\_le\_domaine\_serveur)

00124 Mise\_à\_jour\_de\_fichier\_non\_supportée

00125 Impossible\_de\_localiser\_l’enregistrement\_dans\_le\_fichier

00126 Enregistrement\_dupliqué,\_ancien\_enregistrement\_remplacé

00127 Erreur\_en\_«\_edit\_»\_sur\_champ\_de\_mise\_à\_jour\_fichier

00128 Accès\_interdit\_au\_fichier

00129 Mise\_à\_jour\_de\_fichier\_impossible

00130 Erreur\_de\_format

00133 Carte\_expirée

00138 Nombre\_d’essais\_code\_confidentiel\_dépassé

00141 Carte\_perdue

00143 Carte\_volée

00151 Provision\_insuffisante\_ou\_crédit\_dépassé

00154 Date\_de\_validité\_de\_la\_carte\_dépassée

00155 Code\_confidentiel\_erroné

00156 Carte\_absente\_du\_fichier

00157 Transaction\_non\_permise\_à\_ce\_porteur\_(carte)

00158 Transaction\_interdite\_au\_terminal

00159 Suspicion\_de\_fraude

00160 L’accepteur\_de\_carte\_doit\_contacter\_l’acquéreur

00161 Dépasse\_la\_limite\_du\_montant\_de\_retrait

00163 Règles\_de\_sécurité\_non\_respectées

00168 Réponse\_non\_parvenue\_ou\_reçue\_trop\_tard

00175 Nombre\_d’essais\_code\_confidentiel\_dépassé

00176 Porteur\_déjà\_en\_opposition,\_ancien\_enregistrement\_conservé

00189 Echec\_de\_l’authentification

00190 Arrêt\_momentané\_du\_système

00191 Emetteur\_de\_cartes\_inaccessible

00194 Demande\_dupliquée

00196 Mauvais\_fonctionnement\_du\_système

00197 Echéance\_de\_la\_temporisation\_de\_surveillance\_globale

##### Le fichier des codes ISO des pays

Le fichier Liste\_Pays\_ISO.txt est utilisé par la fonction chargement\_codes\_pays pour afficher le nom complet du pays à partir du code ISO sur 3 caractères.

En Voici un extrait :

ASC Ile\_de\_l'Ascension ILE\_DE\_L'ASCENSION

AND Andorre ANDORRE

ARE Émirats\_Arabes\_Unis ÉMIRATS\_ARABES\_UNIS

AFG Afghanistan AFGHANISTAN

ATG Antigua-et-Barbuda ANTIGUA-ET-BARBUDA

AIA Anguilla ANGUILLA

ALB Albanie ALBANIE

ARM Arménie ARMÉNIE

ANT Antilles\_Neerlandaises ANTILLES\_NEERLANDAISES

AGO Angola ANGOLA

ATA Antarctique ANTARCTIQUE

ARG Argentine ARGENTINE

ASM Samoa\_Américaines SAMOA\_AMÉRICAINES

AUT Autriche AUTRICHE

AUS Australie AUSTRALIE

ABW Aruba ARUBA

ALA Åland,\_Îles ÅLAND,\_ÎLES

AZE Azerbaïdjan AZERBAÏDJAN

BIH Bosnie-Herzégovine BOSNIE-HERZÉGOVINE

BRB Barbade BARBADE

BGD Bangladesh BANGLADESH

BEL Belgique BELGIQUE

BFA Burkina\_Faso BURKINA\_FASO

BGR Bulgarie BULGARIE

FRA France FRANCE

...

### Paiement avec 3D-Secure

Pour activer le mode 3D-Secure de votre boutique, il faut l’indiquer au moment de l’ouverture de la boutique dans le contrat avec Paybox.

Pour tester le fonctionnement de 3D Secure avec la boutique de test mutualisée, il suffit de changer les valeurs pour les variables :

* PBX\_SITE=199988
* PBX\_RANG=43
* PBX\_IDENTIFIANT=107975626

Dans le programme paybox\_clef\_HMAC, il faut mettre en commentaires les 3 lignes :

//$PARAM['SITE']="1999888";

//$PARAM['RANG']="32";

//$PARAM['IDENTIFIANT']="110647233";

et retire le commentaire les 3 lignes suivantes :

$PARAM['SITE']="1999888";

$PARAM['RANG']="43";

$PARAM['IDENTIFIANT']="107975626";

Voici l’extrait de ce fichier où doivent être effectuées les modifications :

// --- Paramètres de la boutique Paybox System classique ---

// --- ici celle de test mutualisée fournie par PAYBOX ---

// --- pour activer (désactiver) cette boutique ---

// --- décommenter(commenter) les 3 lignes pour SITE,RANG et IDENTIFIANT ---

//$PARAM['SITE']="1999888";

//$PARAM['RANG']="32";

//$PARAM['IDENTIFIANT']="110647233";

// --- Pour informaton : accès Backoffice Paybox System classique :

// URL : https://preprod-admin.paybox.com

// LOGIN : 199988832

// Mot\_de\_Passe : 1999888I

// --- Paramètres de la boutique Paybox System 3D-Secure ---

// --- ici celle de test mutualisée fournie par PAYBOX ---

// --- pour activer (désactiver) cette boutique ---

// --- décommenter(commenter) les 3 lignes pour SITE,RANG et IDENTIFIANT ---

$PARAM['SITE']="1999888";

$PARAM['RANG']="43";

$PARAM['IDENTIFIANT']="107975626";

// --- accès Backoffice Paybox System 3D-Secure :

// URL : https://preprod-admin.paybox.com

// LOGIN : 199988843

// Mot\_de\_Passe : 1999888I

*Attention :*

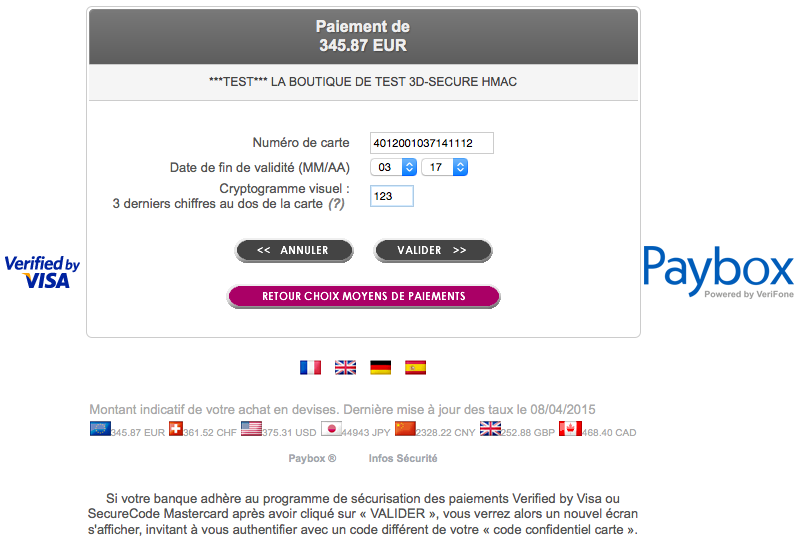
*Le login et le mot de passe du Back-office 3D Secure ne sont pas les mêmes, comme cela est indiqué dans les commentaires du programme.*

*Dans le cas de la boutique de test mutualisée en 3D Secure, le login est 199988843, et le mot de passe est 1999888I.*

L’écran de saisie des informations bancaires, indique bien que le site est en 3D-Secure.

La carte bancaire à utiliser est différente pour ce site. Ses informations sont :

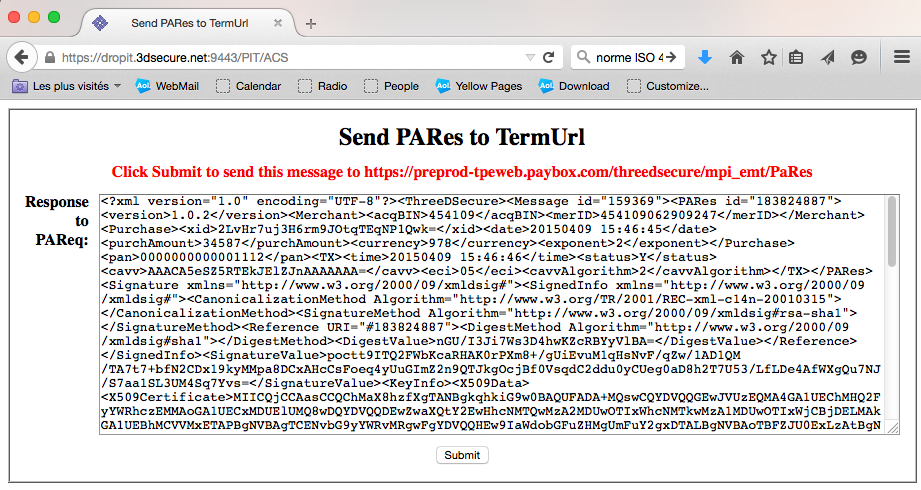
* Numéro de carte : 4012 0010 3714 1112
* Date de validité : choisir une date valide
* Cryptogramme : 123



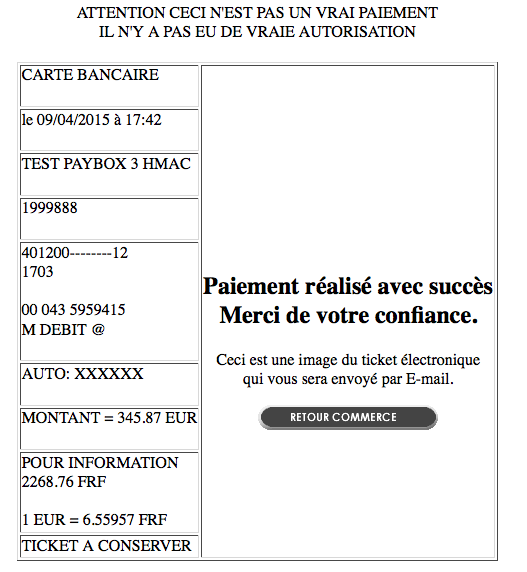
L’écran suivant fait apparaître un certificat à accepter. Cela simule la saisie d’un code qui aurait été envoyé par SMS sur votre téléphone portable (transmis normalement par votre banque) par le site de Paybox. Il faut valider ce certificat.

*Remarque:*

*Le port de communication pour https n’est pas usuel. C’est le* ***9443*** *(et non le 443). Il faut donc veiller que les routeurs réseau de votre site ne bloque pas ce port vers le serveur* ***dropit.3dsecure.net*** *pour effectuer ce test.*

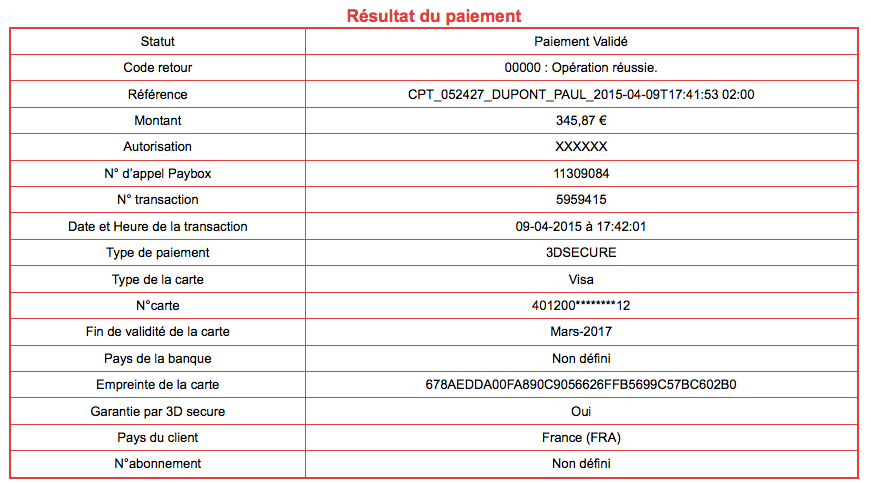


Après validation de ce certificat, le reçu de paiement apparaît à l’écran (il est également envoyé par courriel).



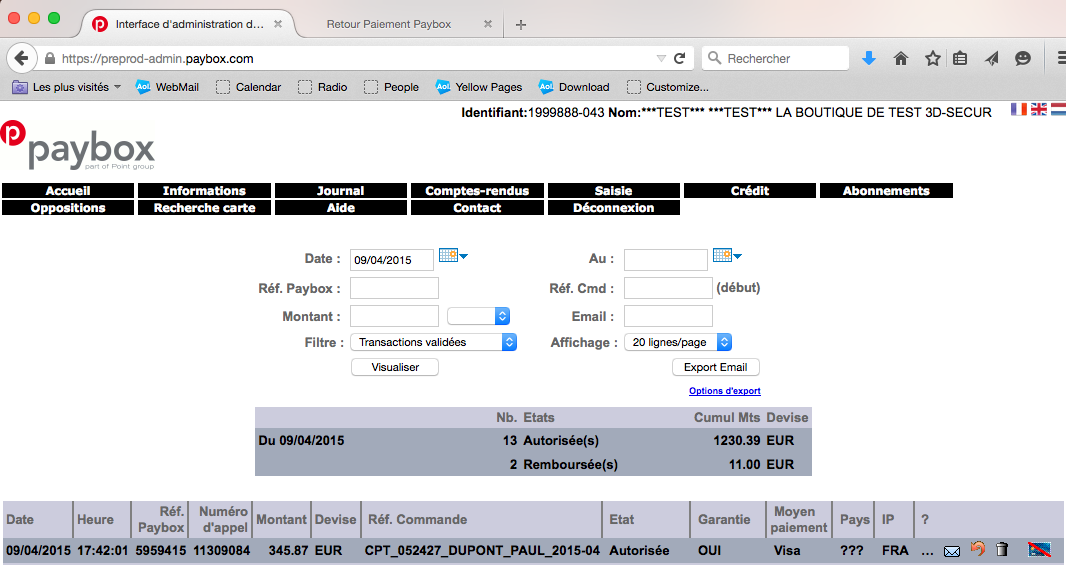
Le retour au site marchand affiche l’interprétation des informations envoyées par Paybox au programme traitement\_retour\_paybox.php qui est reste inchangé.

On peut voir sur l’affichage produit par ce programme que la garantie par 3D Secure est à Oui.



L’accès au Back-office se fait sur la même URL : **https://preprod-admin.paybox.com**, mais avec le login **199988843** et le mot de passe **1999888I**.

La transaction apparaît dans le Back-office. L’entête de la page montre que c’est bien la boutique de test 3D-Secure. Une nouvelle colonne « garantie » indique le résultat de la garantie par 3D-Secure.



**En résumé, pour effectuer ce test il suffit simplement de changer les valeurs pour les variables suivantes, et de vérifier l’ouverture du port 9443 en sortie.**

Aucun autre changement en terme de programmation n’est nécessaire :

* PBX\_SITE=199988
* PBX\_RANG=43
* PBX\_IDENTIFIANT=107975626