



**BNP PARIBAS**



## **GUIDE D'INSTALLATION PLUG-IN A6.15**

# Sommaire

<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>2. LISTE DES OBJETS LIVRES .....</b>	<b>1</b>
<b>3. SCHEMA DES FLUX ET APPEL DES SCRIPTS .....</b>	<b>1</b>
3.1 POUR UN PAIEMENT NON 3-D SECURE .....	1
3.2 POUR UN PAIEMENT 3-D SECURE AVEC UNE CARTE ENROLEE .....	1
<b>4. ETAPES D'INSTALLATION POUR EXECUTER LES EXEMPLES SUR LE SERVEUR DE DEMONSTRATION .....</b>	<b>1</b>
4.1 SI VOUS ETES INSCRIT AU PROGRAMME 3-D SECURE .....	1
4.2 APPEL DU SERVEUR MERC@NET .....	1
4.3 PROCEDURE DE TEST SUR LE SERVEUR MERC@NET DE DEMONSTRATION.....	1
4.3.1 <i>Test de paiement pour un commerçant non inscrit au programme 3-D Secure</i> .....	1
4.3.2 <i>Test de paiement pour un commerçant inscrit au programme 3-D Secure</i> .....	1
4.4 RECEVOIR LA REPONSE A LA FIN D'UN PAIEMENT .....	1
4.5 RECEVOIR LA REPONSE AUTOMATIQUE DU SERVEUR MERC@NET .....	1
<b>5. DEVELOPPEMENT DE VOS SCRIPTS.....</b>	<b>1</b>
<b>6. COMMENT PASSER SUR LE SERVEUR DE PRODUCTION .....</b>	<b>1</b>

# 1. INTRODUCTION

Ce document vous explique comment installer votre API Merc@net et comment démarrer vos premiers tests de paiement. Pour installer l'API, vous devrez développer 3 scripts et configurer 3 fichiers de paramètres. La mise en place complète de l'API avant le passage en production se déroule en 3 phases :

- les premiers tests avec les scripts d'exemples fournis pour comprendre le fonctionnement de l'API,
- le développement de vos scripts pour l'intégration à votre site,
- le passage en « pré-production » pour tester une demande d'autorisation et l'installation éventuelle de votre charte graphique.

Note : ce document ne décrit pas comment vous interfacer avec votre système d'information ou votre base de données. Dans les exemples fournis, les variables sont déjà renseignées, vous devrez programmer la lecture et la mise à jour des données de votre système d'information.

Pré-requis :

- Connaissances de base de PHP ou ASP
- Interpréteur PHP
- Serveur HTTP pour exécuter les scripts

## 2. LISTE DES OBJETS LIVRES

Fichier /Version.txt

fichier précisant l'environnement dans lequel l'API a été compilée et testée.

Répertoire /logo

répertoire des logos des moyens de paiement

Répertoire /param

certif.fr.011223344551111.php	Certificat de la boutique de démonstration
certif.fr.011223344551112.php	Certificat de la boutique de démonstration 3D
certif.fr.011223344551111.asp	Certificat de la boutique de démonstration
certif.fr.011223344551112.asp	Certificat de la boutique de démonstration 3D
parmcom.011223344551111	Fichier des paramètres de la boutique
parmcom.default	Fichier des paramètres par défaut
Pathfile	Fichier des chemins d'accès aux fichiers paramètres

Répertoire /sample

call_request.php	Exemple de script PHP pour générer la requête de paiement
call_response.php	Exemple de script PHP pour traiter la réponse manuelle
call_autoresponse.php	Exemple de script PHP pour traiter la réponse automatique
call_request.asp	Exemple de script ASP pour générer la requête de paiement
call_response.asp	Exemple de script ASP pour traiter la réponse manuelle
call_autoresponse.asp	Exemple de script ASP pour traiter la réponse automatique

Répertoire /bin

request	exécutable appelé par les scripts pour générer la requête de paiement
response	exécutable appelé par les scripts pour traiter la réponse manuelle et automatique

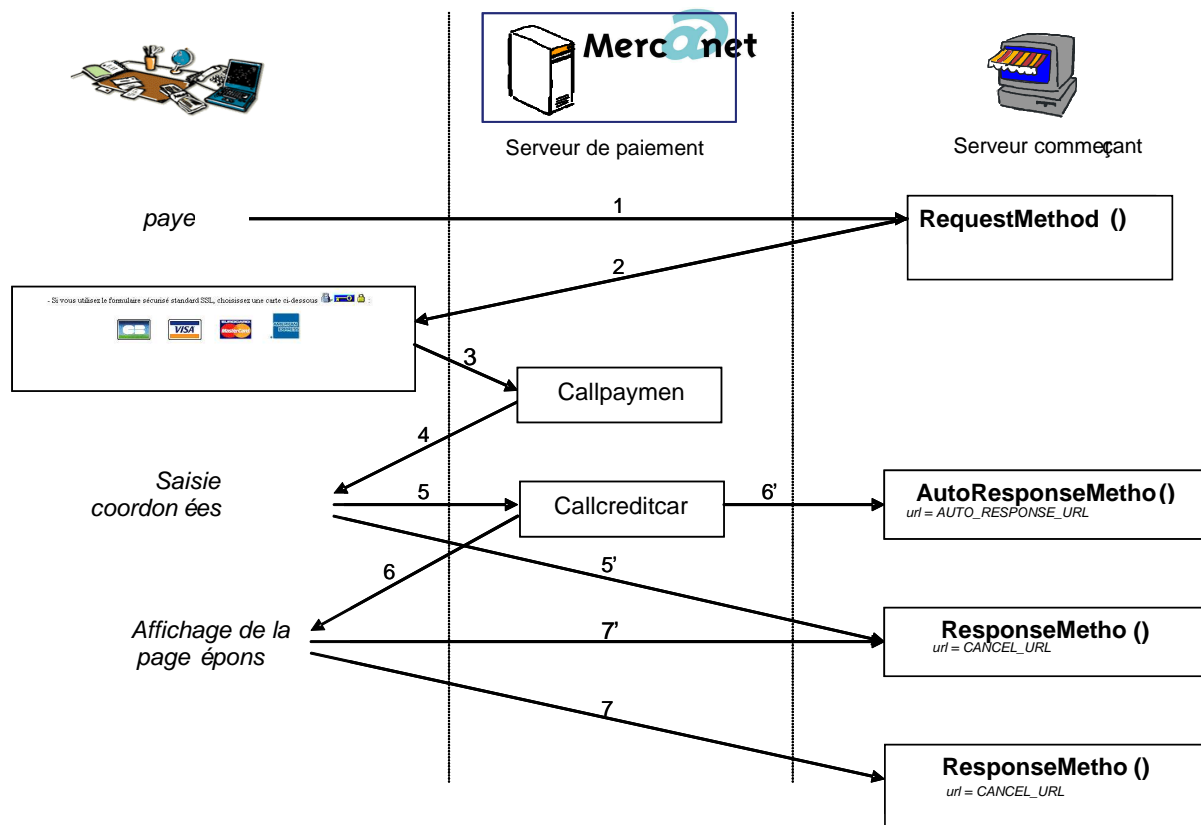
### **Remarque :**

Etant donné que la procédure d'installation des scripts ne dépend pas de leur langage (PHP ou ASP), lorsque nous les évoquerons, nous ne préciserons pas d'extension.

Les fichiers ayant asp comme extension sont uniquement disponibles dans la version Windows.

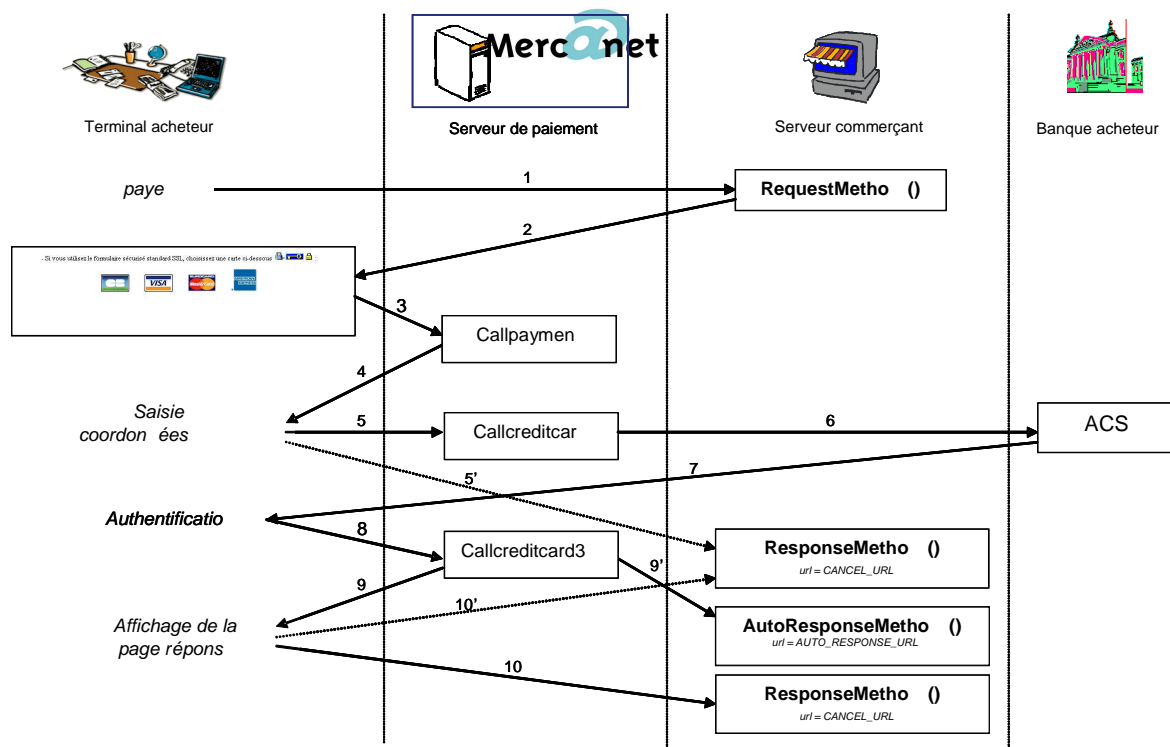
### 3. SCHEMA DES FLUX ET APPEL DES SCRIPTS

#### 3.1 POUR UN PAIEMENT NON 3-D SECURE



- 1 L'internaute a rempli son caddie, il souhaite passer à la caisse
  - 2 Le script « call\_request » est exécuté, il affiche tous les moyens de paiement acceptés par la boutique.
  - 3 En cliquant sur un de ces moyens de paiement, l'internaute se connecte au serveur de paiement.
  - 4 Le serveur de paiement envoie le formulaire approprié pour obtenir ses coordonnées bancaires.
  - 5 Après saisie de ses coordonnées bancaires, l'internaute soumet le formulaire au serveur de paiement qui va effectuer une demande d'autorisation auprès d'une institution financière impliquée dans le paiement sélectionné.
  - 5' L'internaute peut ne pas saisir ses coordonnées bancaires et annuler sa transaction.
  - 6 et 6' Après réception de la réponse d'autorisation, le serveur de paiement envoie simultanément une réponse au serveur commerçant (appel du script « call\_autoresponse ») ainsi qu'un ticket de caisse sur l'écran de l'internaute.
  - 7 et 7' A partir de ce ticket, l'internaute peut cliquer sur un bouton pour retourner sur le serveur commerçant, qui reçoit ainsi le résultat du paiement dans une variable cachée. Le script « call\_response » récupère le résultat et affiche une page appropriée.
- Si le paiement est refusé, l'internaute suit le flux 7'.

## 3.2 POUR UN PAIEMENT 3-D SECURE AVEC UNE CARTE ENROLEE



- 1 L'internaute a rempli son caddie, il souhaite passer à la caisse
- 2 Le script « call\_request » est exécuté, il affiche tous les moyens de paiement acceptés par la boutique.
- 3 En cliquant sur un de ces moyens de paiement, l'internaute se connecte au serveur de paiement.
- 4 Le serveur de paiement envoie le formulaire approprié pour obtenir ses coordonnées bancaires.
- 5 Après saisie de ses coordonnées carte, l'internaute soumet le formulaire au serveur de paiement.
- 5' L'internaute peut ne pas saisir ses coordonnées bancaires et annuler sa transaction.
- 6 Le serveur de paiement transfère les coordonnées carte vers la banque du porteur.
- 7 La banque du porteur affiche un formulaire pour saisie du mot de passe et vérifie le mot de passe saisi.
- 8 Le serveur de paiement est destinataire de la vérification du mot de passe, et envoie une demande d'autorisation vers le réseau bancaire.
- 9 et 9' Après réception de la réponse d'autorisation, le serveur de paiement envoie simultanément une réponse au serveur commerçant (appel du script « call\_autoresponse ») ainsi qu'un ticket de caisse sur l'écran de l'internaute.
- 10 et 10' A partir de ce ticket, l'internaute peut cliquer sur un bouton pour retourner sur le serveur commerçant, qui reçoit ainsi le résultat du paiement dans une variable cachée. Le script « call\_response » récupère le résultat et affiche une page appropriée.  
Si le paiement est refusé, l'internaute suit le flux 10'.

## 4. ETAPES D'INSTALLATION POUR EXECUTER LES EXEMPLES SUR LE SERVEUR DE DEMONSTRATION

Cette première phase a pour but d'effectuer votre premier paiement sur le serveur de démonstration. Dans chaque langage de script, les 3 exemples fournis vous permettent de traiter les 3 messages échangés entre votre serveur, le navigateur de l'acheteur et le serveur de paiement.

Tout d'abord vous devez lancer l'installation des fichiers de l'API :

- pour un OS de type Unix : détarrer le fichier Sips\_XXX\_Plug-in\_XXXX.tar en utilisant la commande suivante :

```
tar -xvf Sips_XXX_Plug-in_XXXX.tar
```

- pour Windows : lancer Sips\_XXX\_Plug-in\_XXXX.exe et suivez les instructions du programme d'installation.

### 4.1 SI VOUS ETES INSCRIT AU PROGRAMME 3-D SECURE

Le commerçant 011223344551111 n'est pas inscrit par défaut au programme 3-D Secure. Pour effectuer des tests avec un commerçant de démonstration considéré comme « inscrit » au programme 3-D Secure, vous devez utiliser le commerçant 011223344551112.

Pour ceci effectuez au préalable les 2 opérations suivantes :

- dupliquer le fichier parmcom.011223344551111 et le renommer en parmcom.011223344551112.
- remplacer le numéro de la boutique de démonstration (011223344551111) par le numéro de la boutique de démonstration 3-D Secure (011223344551112) (cf. champ *merchant\_id*) dans le fichier source call\_request.

## 4.2 APPEL DU SERVEUR MERC@NET

1. Modifiez le contenu du fichier **pathfile** en fonction de votre répertoire d'installation.

Avec un OS Windows, suivez les instructions de l'installateur qui générera votre fichier pathfile. Vous pourrez le modifier manuellement après, par exemple :

F\_CERTIFICATE!c:\Sips\payment\param\certif!

F\_PARAM!c:\Sips\payment\param\paramcom!

F\_DEFAULT!c:\Sips\payment\param\paramcom.default!

Avec un OS de type Unix, par exemple :

F\_CERTIFICATE!/home/Sips/payment\param/certif!

F\_PARAM!/home/Sips/payment\param\paramcom!

F\_DEFAULT!/home/Sips/payment\param\paramcom.default!

Notez que les fichiers F\_CERTIFICATE et F\_PARAM ne prennent pas d'extension (.011223344551111 ou .011223344551112). Ces données seront rajoutées par l'API.

Le type de certificat est renseigné par : F\_CTYPE !php! (ou F\_CTYPE !asp! seulement possible sous Windows)

Cela correspond à l'extension du fichier certificat suivant 011223344551111 dans le nom du certificat.

Renseignez D\_LOGO avec un chemin internet : Si vos logos sont accessibles depuis un navigateur sur l'URL [http://www.maboutique.fr/logo\\_api/CB.gif](http://www.maboutique.fr/logo_api/CB.gif),

il faudra renseigner : D\_LOGO!/logo\_api/!

Renseignez DEBUG à YES si vous souhaitez passer en mode DEBUG.

2. Editez **call\_request**

3. Modifiez le chemin de l'exécutable **request** (Ne pas mettre d'espace dans le chemin).

4. Vérifiez que l'exécutable **request** a bien les droits d'exécution.

5. Modifiez le chemin vers le fichier **pathfile** (Ne pas mettre d'espace dans le chemin).

6. Vérifiez que le fichier **pathfile** a bien les droits de lecture.

7. Copier le script **call\_request** dans le répertoire des cgi de votre serveur web.

8. Copier le fichier **pathfile** dans le répertoire des cgi de votre serveur web.

9. Appeler depuis un navigateur le script **call\_request**. Vous devez voir apparaître les logos des cartes bancaires.

Si vous souhaitez installer le fichier **pathfile** dans le répertoire de votre choix, référez-vous au *GUIDE DU PROGRAMMEUR*.



## **4.3 PROCEDURE DE TEST SUR LE SERVEUR MERC@NET DE DEMONSTRATION**

### **4.3.1 Test de paiement pour un commerçant non inscrit au programme 3-D Secure**

Vous devez utiliser le commerçant 011223344551111

Sur le serveur de démonstration Merc@net, le processus d'autorisation est simulé. Il est donc possible de saisir n'importe quel numéro de carte sans aucune conséquence.

Le code réponse de la demande d'autorisation simulée (champ **bank\_response\_code**) correspond aux deux derniers chiffres du numéro de carte bancaire.

Le champ **response\_code** est simulé selon la valeur du **bank\_response\_code**.

Pour connaître les valeurs possibles des champs **bank\_response\_code** et **response\_code** en production, référez-vous au *DICTIONNAIRE DES DONNEES*.

<b>Exemple :</b>	numéro de carte	code réponse
	4974934125497800	00 (paiement accepté)
	4972187615205	05 (paiement refusé)

**Attention :** la date de validité de la carte doit être postérieure à la date du jour. La taille du numéro de carte doit être comprise entre 10 et 19 chiffres.

Pour les cartes CB, VISA et MASTERCARD, vous devez saisir un cryptogramme visuel (clé sécuritaire à trois chiffres). Comme pour le numéro de carte, les 2 derniers chiffres simulent le **cvv\_response\_code**.

<b>Exemple :</b>	cryptogramme	<b>cvv_response_code</b>
	600	4D
	640	4D
	650	50
	653	53
	655	55

Tout cryptogramme dont les 2 derniers chiffres diffèrent de 00, 40, 50, 53 ou 55 conduit à un **cvv\_response\_code** égal à 4E.

Pour connaître la signification des différentes valeurs du **cvv\_response\_code**, reportez-vous à l'annexe traitant du cryptogramme visuel dans le *DICTIONNAIRE DES DONNEES*.

### **4.3.2 Test de paiement pour un commerçant inscrit au programme 3-D Secure**

Vous devez utiliser le commerçant 011223344551112

Pour les demandes d'autorisation le processus de simulation (des champs **response\_code**, **bank\_response\_code** et **cvv\_response\_code**) est le même que pour un commerçant non inscrit au programme 3-D Secure (cf. paragraphe précédent).

Les différents cas de traitement 3D sont simulés à partir du numéro de carte :

- simuler une carte VISA enrôlée : saisir un numéro de carte commençant par 4
- simuler une carte MASTERCARD enrôlée : saisir un numéro de carte commençant par 5
- simuler une erreur technique pendant le processus d'authentification : saisir un numéro de carte commençant par 9
- simuler un paiement avec une carte VISA ou MASTERCARD non enrôlée : saisir un numéro de carte ne commençant pas par 4, 5 ou 9
- simuler le dialogue avec l'ACS dans la langue anglaise : saisir un numéro de carte dont le deuxième chiffre est un 0

Si vous saisissez un numéro de carte enrôlé, le mot de passe pour s'authentifier est « 00000000 ».

## **4.4 RECEVOIR LA REPONSE A LA FIN D'UN PAIEMENT**

Nous supposons que vous avez maintenant terminé les deux premières étapes, vous êtes donc maintenant capable de connecter votre internaute au serveur Merc@net en vue d'un paiement.

Afin de recevoir la réponse du serveur Merc@net sur votre serveur Web, il vous faut avoir configuré au préalable une adresse URL de réponse pour que le serveur Merc@net sache où il doit envoyer la réponse.

La manière la plus facile de le faire est de configurer l'URL de réponse dans le fichier paramètre (parmcom.011223344551111 ou parmcom.011223344551112).

1. Editez **call\_response**
2. Modifiez le chemin vers l'exécutable **response** (Ne pas mettre d'espace dans le chemin).
3. Modifiez le chemin vers le fichier **pathfile** (Ne pas mettre d'espace dans le chemin).
4. Vérifiez que l'exécutable **response** a bien les droits d'exécution.
5. Copier le script **call\_response** dans le répertoire des cgi de votre serveur web.
6. Editez le fichier parmcom.011223344551111 (ou parmcom.011223344551112), recherchez le paramètre RETURN\_URL et CANCEL\_URL, et changez les valeurs pour appeler le script **call\_response** sur votre site web.

Exemple pour PHP :

RETURN\_URL!http://www.maboutique.fr/cgi-bin/call\_response.php!

CANCEL\_URL!http://www.maboutique.fr/cgi-bin/call\_response.php!

7. Effectuez un nouveau paiement en utilisant le script modifié dans la première étape. Sur la deuxième page de paiement, le bouton « RETOUR A LA BOUTIQUE » doit rediriger vers le script **call\_response** (vous pouvez le vérifier en regardant la source sur votre navigateur).
8. Lorsque vous cliquez sur le bouton « RETOUR A LA BOUTIQUE », vous quittez le serveur Merc@net et retournez sur le serveur commerçant via le script **call\_response**. Le script livré en exemple affichera entièrement la structure de la réponse reçue par le serveur Merc@net.

## **4.5 RECEVOIR LA REPONSE AUTOMATIQUE DU SERVEUR MERC@NET**

Vous avez effectué votre premier paiement sécurisé en ligne. Félicitations! Comme vous l'avez probablement remarqué, la réponse a été renvoyée par le serveur de paiement Merc@net à votre serveur lorsque vous avez cliqué sur le bouton « RETOUR A LA BOUTIQUE ». Le risque avec ce mode d'envoi de réponse est que l'internaute ne clique pas sur ce bouton et que votre serveur ne reçoive jamais la réponse à la requête de paiement.

Afin d'éviter ce risque, une autre réponse a été configurée dans le serveur de paiement Merc@net, elle s'appelle la réponse automatique. C'est une réponse systématiquement envoyée par le serveur Merc@net à votre serveur commerçant à chaque fois qu'il traite une demande d'autorisation, donc à chaque fois qu'un internaute valide ses coordonnées carte.

Pour ce faire, le serveur Merc@net envoie une requête HTTP au serveur commerçant, sur l'URL spécifiée dans le paramètre `AUTO_RESPONSE_URL`.

**Remarque :** Notez qu'il s'agit ici de dialogue serveur/serveur. L'absence de navigateur interdit les redirections et les affichages. Le script placé sur cette URL ne doit effectuer que des traitements de type mise à jour de base de données ou logging.

1. Editez **call\_autoreponse**
2. Modifiez le chemin vers l'exécutable **response**. Le même exécutable est appelé pour la réponse manuelle et la réponse automatique (Ne pas mettre d'espace dans le chemin).
3. Modifiez le chemin vers le fichier **pathfile** (Ne pas mettre d'espace dans le chemin).
4. L'exemple livré copie tous les champs de la réponse dans un fichier trace sur la machine qui exécute le script **call\_autoreponse**. Vous pouvez modifier le chemin du fichier trace dans le script (Ne pas mettre d'espace dans le chemin).
5. Copier le script **call\_autoreponse** dans le répertoire des cgi de votre serveur web
6. Editez le fichier `parmcom.011223344551111` (ou `parmcom.011223344551112`), recherchez le paramètre `AUTO_RESPONSE_URL`, changez la valeur pour appeler le script **call\_autoreponse** sur votre site et retirez le # si la ligne est en commentaire.

Exemple pour PHP :

`AUTO_RESPONSE_URL!http://www.maboutique.fr/cgi-bin/call_autoreponse.php!`

7. Effectuez un nouveau paiement, en utilisant le script de la première étape
8. Lorsque vous soumettez votre paiement sur la « page de saisie », le serveur de paiement Merc@net appelle l'URL de réponse automatique et affiche la page de réponse au même moment. Lorsque votre client lit cette seconde page, vous connaissez déjà le résultat du paiement.

**Remarque :** si vous ne recevez pas la réponse automatique, référez-vous au *GUIDE DU PROGRAMMEUR*.

## 5. DEVELOPPEMENT DE VOS SCRIPTS

Au cours de cette phase, vous allez adapter les scripts d'exemples à vos besoins, pour qu'ils s'intègrent bien à votre serveur web et qu'ils échangent les données nécessaires avec votre système d'information.

Vous poursuivrez vos tests sur le serveur de « démonstration » qui affiche rigoureusement les mêmes pages que le serveur de « production », mais qui simule la demande d'autorisation à la banque. De cette manière vous pouvez multiplier les essais de paiements, et vous n'avez pas besoin d'utiliser une carte réelle.

Pour plus de détails sur les données de l'API, veuillez consulter le *GUIDE DU PROGRAMMEUR*.

## 6. COMMENT PASSER SUR LE SERVEUR DE PRODUCTION

La dernière phase est le passage en mode « pré-production ». A partir de ce moment les paiements sont effectués sur le serveur Merc@net de production. Les demandes d'autorisation ne sont plus simulées comme sur le serveur de démonstration, mais elles empruntent le circuit complet du réseau bancaire. Cette phase va permettre de contrôler la bonne inscription de votre contrat bancaire et de tester la personnalisation de vos pages de paiement (cf. *GUIDE DU PROGRAMMEUR* et *GUIDE DE PERSONNALISATION DES PAGES*).

Le passage en « pré-production » se fait par l'activation du certificat du commerçant qui va remplacer le certificat de démonstration.

Tant que vous êtes en « pré-production » un message d'alerte s'affiche sur la page de saisie du numéro de carte, les paiements effectués sur votre site ne sont pas débités.

1. Copier le certificat de « production », qui vous a été transmis, dans le même répertoire que le certificat de « démonstration ». Le certificat de production est nommé `certif.fr.<my_merchant_id>.php` ou `certif.fr.<my_merchant_id>.asp` selon votre choix de langage des scripts d'appel, où `<my_merchant_id>` est votre numéro de boutique.
2. Renommer le fichier des paramètres de la boutique (fichier `parmcom.011223344551111` ou fichier `parmcom.011223344551112`) en `parmcom.<my_merchant_id>`.
3. Remplacer le numéro de la boutique de démonstration (011223344551111 ou 011223344551112) par votre numéro de boutique (cf. champ *merchant\_id*) dans vos scripts.
4. Copier les scripts modifiés dans le répertoire des cgi de votre serveur web
5. Faire au minimum un test de paiement autorisé. Pour ce test, vous devez utiliser un numéro de carte réelle. Tant que vous êtes en phase de pré-production, les paiements effectués sur votre site ne sont pas débités. Vous ne pourrez passer en « production » que si au moins un paiement a été autorisé.
6. Si vous avez envoyé des logos ou des templates à installer sur le serveur Merc@net, vous pouvez les tester à ce moment en renseignant leurs noms dans les champs correspondants (cf. *DICTIONNAIRE DES DONNEES* et *GUIDE DE PERSONNALISATION DES PAGES*).
7. Pour le passage en « production », voir le document de *PRESENTATION FONCTIONNELLE* paragraphe **Comment en profiter**.