



BNP PARIBAS



GUIDE DE PERSONNALISATION DES PAGES

SOMMAIRE

1.	INTRODUCTION	3
2.	PAGE DES MOYENS DE PAIEMENT AFFICHEE PAR L'API	3
3.	PAGES AFFICHEES PAR LE SERVEUR SIPS	5
A.	INTRODUCTION	5
B.	PARTICULARITES LIEES A 3-D SECURE	5
C.	UTILISATION DE FRAMES	6
D.	LIMITATIONS	6
E.	LOGOS DES BOUTONS	8
F.	TEMPLATE PAR DEFAUT	8
G.	LOGO EN HAUT AU CENTRE DE LA PAGE	8
H.	LOGO A GAUCHE DU NOM DE LA BOUTIQUE	9
1.	LOGO A DROITE DU NOM DE LA BOUTIQUE	9
2.	LE NOM DE LA BOUTIQUE	9
3.	FOND D'ECRAN	9
4.	COULEUR DU TEXTE DU SERVEUR MERC@NET	9
I.	TEMPLATE PERSONNALISE	10
1.	CONTENU D'UN TEMPLATE	10
2.	LIMITATIONS	10
3.	UTILISATION D'UN TEMPLATE UNIQUE POUR L'ENSEMBLE DES PAGES DU SERVEUR	11
4.	UTILISATION D'UN TEMPLATE DIFFERENT POUR CHACUNE DES PAGES DU SERVEUR	11
5.	LES TEMPLATES EN PLUSIEURS LANGUES	12
J.	PERSONNALISATION PAR CSS	12
1.	SIMPLICITE DE FONCTIONNEMENT	13
2.	PAGES MERC@NET A PERSONNALISER	13
K.	ADAPTATION DES PAGES DE PAIEMENT POUR UN SMARTPHONE	14
1.	UTILISATION D'UNE CSS DEDIEE AU SMARTPHONE	15
2.	UTILISATION DE LA PERSONNALISATION PAR CSS	15
L.	AUTRES PARAMETRAGES DE LA PERSONNALISATION	16
1.	LA PERSONNALISATION DU TICKET (CHAMP RECEIPT_COMPLEMENT)	16
M.	AUTRES OPTIONS D'AFFICHAGE (CHAMP DATA)	16
4.	L'OUTIL TEST_TEMPLATE	18
A.	GENERALITES	18
B.	OUTIL TEST_TEMPLATE VERSION WINDOWS	18
C.	OUTIL TEST_TEMPLATE TOUT SYSTEME D'EXPLOITATION (JAVA)	19
D.	OPTIONS DISPONIBLES	21
5.	ANNEXES	25
A.	ANNEXE A : TABLEAU DES CODES RGB	25
B.	ANNEXE B : EXEMPLES DE CODE POUR LA PERSONNALISATION PAR CSS	26
C.	ANNEXE C : PAGES MERC@NET ET CLASSES CSS ASSOCIEES	27
1.	SAISIE DES COORDONNEES BANCAIRES	27
2.	CAS D'UN PAIEMENT PAR CARTE AMERICAN EXPRESS	28
D.	CAS D'UN PAIEMENT ACCEPTE	30
1.	CAS D'UN PAIEMENT ACCEPTE POUR UNE CB, VISA, MASTERCARD	30
E.	CAS D'UN PAIEMENT REJETE	31
F.	CAS D'UN PAIEMENT REJOUÉ	32
G.	CAS D'UN PAIEMENT EN COURS DE TRAITEMENT	33
1.	TRANSACTION EN COURS DE TRAITEMENT POUR CB, VISA, MASTERCARD	33
H.	CAS D'UNE REDIRECTION 3D-SECURE	34

1. INTRODUCTION

Les pages de paiement Merc@net sont personnalisables afin de pouvoir s'intégrer pleinement dans la charte graphique du site commerçant.

Ce document a pour but de décrire les possibilités et les limites de la personnalisation aussi bien au niveau de l'API que du serveur Sips.

Remarque :

Les possibilités d'affichage sur le serveur Sips dépendent de la phase d'installation de l'API. En phase de démonstration le commerçant ne pourra pas visualiser les logos propres à sa boutique. En effet, l'affichage de ces logos ne pourra se faire qu'en phase de « pré-production » ou de « production » car il est subordonné à leur installation sur le serveur Sips dans un répertoire propre à la boutique.

Il est important de noter qu'en ce qui concerne la personnalisation, aucune différence n'est faite entre la phase de « production » et de « pré-production ». Dans le chapitre 3, nous supposons donc que vous êtes dans la phase de « pré-production », car c'est principalement durant cette phase que l'affichage doit être paramétré.

Conventions d'écriture :

- Les renvois à d'autres documentations seront notés en majuscules et en italique.
ex : *DICTIONNAIRE DES DONNEES*
- Les champs de l'API seront notés en **gras**.
ex : **block_order**, **block_align**

PAGE DES MOYENS DE PAIEMENT AFFICHEE PAR L'API

L'API n'affiche qu'une seule page sur le navigateur de l'internaute. Cette page comporte les logos des moyens de paiement. La page de retour après la transaction ne comporte aucun affichage obligatoire, le commerçant a complètement la main sur cette page, il peut donc gérer complètement l'affichage.

La page des logos des moyens de paiement étant localisée sur le serveur commerçant, le commerçant peut la personnaliser à sa guise. Sur cette page, seules les listes des logos des moyens de paiement et les phrases de commentaire situées au-dessus de celles-ci sont affichées par l'API. Cet ensemble - liste de logos de moyens de paiement et phrase de commentaire associée - forme ce que l'on appelle un bloc de paiement. Ce bloc est en fait un formulaire permettant de soumettre la transaction au serveur Sips. Toutes les possibilités d'affichage de ces blocs de paiement sont décrites dans ce paragraphe.

Sur la Figure 1, est présentée une page comportant deux blocs de paiement, l'un constitué des cartes CB, VISA et MASTERCARD, l'autre constitué de la carte AMERICAN EXPRESS. Chacun de ces blocs comporte une phrase de commentaire qui lui est propre.



Figure 1

Les moyens de paiement sont paramétrés dans le champ **payment_means** de l'API (cf. *DICTIONNAIRE DES DONNEES*). Dans le cas précédent, le champ **payment_means** contiendra la valeur CB,1,VISA,1,MASTERCARD,1,AMEX,2. Chaque numéro associe un logo de carte à un bloc.

Les trois champs de l'API **header_flag**, **block_align** et **block_order** permettent de gérer l'affichage des différents blocs.

header_flag

Ce champ renseigné dans la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*) indique si l'API doit afficher ou non les phrases de commentaire des différents blocs. Deux valeurs sont acceptées *yes* ou *no*. Dans le cas précédent, le **header_flag** était paramétré à *yes*. Si nous paramétrons ce champ à *no* nous obtenons le résultat présenté Figure 2.



Figure 2

block_align

Ce champ renseigné dans la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*) indique le positionnement des blocs sur la page. Il peut prendre trois valeurs *left*, *center* et *right*. Dans les deux cas précédents, le champ était renseigné avec la valeur *center*. Avec les valeurs *left* ou *right*, les blocs sont respectivement alignés à gauche ou à droite de la page. La Figure 3 représente la page de paiement avec un champ **block_align** renseigné à *left*.



Figure 3

block_order

Ce champ renseigné dans la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*) indique l'ordre d'affichage des blocs de paiement. La valeur par défaut est « 1,2,3,4,5,6,7,8,9 ». Si nous reprenons le même champ **payment_means** que précédemment, mais en utilisant un **block_order** renseigné à « 2,1,3,4,5,6,7,8,9 », nous obtiendrons sur la page le résultat présenté Figure 4.



Figure 4

Ce champ n'a donc d'intérêt que si vous affichez plusieurs blocs.

Remarque

Un commerçant qui affiche les logos des moyens de paiement dans une frame, peut paramétrer l'affichage des pages de paiement suivantes à l'aide du champ **target** de l'API. Il contient l'attribut HTML *target* des liens associés à chaque logo des moyens de paiement. Pour plus d'information sur ce champ référez-vous au *DICTIONNAIRE DES DONNEES*.

2. PAGES AFFICHEES PAR LE SERVEUR SIPS

A. INTRODUCTION

Le serveur de paiement affiche 3 pages :

- la page de capture qui correspond à la page de saisie des coordonnées bancaires
- la page d'accord
- la page de refus

Le paramétrage de l'affichage des pages présentes sur le serveur Sips se fait via des feuilles de style ou des template. Trois possibilités existent :

- Un template par défaut, unique pour toutes les pages du serveur.
- Un template créé par le commerçant, unique pour toutes les pages du serveur.
- Trois templates différents créés par le commerçant, affichés en fonction de l'étape du processus de paiement.

Tous les boutons présents sur les différentes pages du serveur peuvent également être personnalisés. Par contre, le texte contenu dans ces pages ne peut pas être modifié.

Remarque

Tous les logos que vous souhaitez afficher doivent être transmis par mail au **Centre d'Assistance Technique**, ils seront ensuite appelés en renseignant des champs de l'API. Afin d'éviter tout problème lors de l'appel de ces logos, nous vous recommandons d'utiliser des noms en minuscule et ne comportant aucun espace.

B. PARTICULARITES LIEES A 3-D SECURE

Visa et Mastercard diffusent des « guidelines » destinés à la communication relative aux programmes 3-D Secure. Si le commerçant est inscrit au programme 3-D Secure, il doit se rapprocher de sa banque afin de les obtenir.

Lorsque le commerçant est inscrit au programme 3-D Secure, le serveur Sips vérifie l'enrôlement au programme 3-D Secure de la carte de l'acheteur. Dans le cas où la carte est enrôlée, deux pages supplémentaires seront affichées entre la page de capture et la page finale (d'accord ou de refus) :





- La première, la page d'information 3-D Secure : cette page affichée par le serveur Sips informe l'acheteur que sa carte va être authentifiée.
- La seconde, la page d'authentification 3-D Secure : cette page affichée par le serveur de contrôle d'accès (ACS) de la banque qui a émis la carte permet d'authentifier l'acheteur.

Pour la page d'information 3-D Secure, le commerçant peut :

- Soit utiliser le template par défaut, unique pour toutes les pages du serveur,
- Soit utiliser son template unique qu'il a créé pour toutes les pages du serveur,
- Soit utiliser le template de la page de saisie du numéro de carte,
- Soit créer et utiliser un template réservé à cette page.

La page d'authentification 3-D Secure n'est pas personnalisable et son contenu est sous la responsabilité de la banque qui a émis la carte.

Lorsque le type de carte utilisé est supporté par le programme 3-D Secure (Visa ou MasterCard), le logo carte affiché sur les pages de paiement sera remplacé par un logo spécifique 3-D Secure.

Logo affiché par le serveur Sips		
	commerçant non 3D	commerçant 3D
Carte Visa		
Carte MasterCard		

C. UTILISATION DE FRAMES

Si vous utilisez des frames, vous pouvez indiquer la fenêtre dans laquelle seront affichées les pages de dialogue avec le serveur Sips à l'aide du champ **target** de l'API. Cependant, l'utilisation de frames peut conduire à la perte de l'affichage de la petite clé sécurisée du navigateur et à l'apparition d'un message d'alerte.

Pour plus de renseignements sur ce champ et sur les conséquences de l'utilisation des frames, référez-vous au **DICTIONNAIRE DES DONNEES**.

D. LIMITATIONS

Les pages de paiement Sips sont en partie figées car elles sont communes à tous les commerçants de Sips. En particulier, le texte affiché et les formulaires de saisie ne peuvent pas être modifiés. Sur les figures suivantes, les parties non modifiables des pages sont encadrées.

Sur la page de réponse, dans le cas d'une transaction acceptée, une ligne de code HTML pourra être ajoutée sur le ticket (cf. **receipt_complement**).



Figure 5 : texte non modifiable de la page de saisie du numéro de carte



Lia fleur

Identifiant commerçant	000000000005556
Référence de la transaction	173843
Montant de la transaction	1,00 €



Afin d'authentifier votre carte, vous allez être mis en relation avec le site de votre banque.
 Si la page d'authentification ne s'affiche pas automatiquement dans quelques secondes, veuillez [cliquer ici](#).

Copyright © 2009, tous droits réservés

Figure 6 : texte non modifiable de la page d'information 3-D Secure



Boutique de test

Votre transaction a bien été enregistrée

RETOUR À LA BOUTIQUE

Date de la transaction	11 Août 2009 14:43:12 (UTC/GMT)
Adresse du web commerçant	www.boutique-demo.sips-atos.com
Identifiant commerçant	000000000001111
Référence de la transaction	164308
Montant de la transaction	1,00 €
Numéro de la carte	4907 ##### #00 03/11
Autorisation	001790
Certificat de la transaction	1250001790

Merci de conserver la référence de la transaction.

Suivant le paramétrage de votre navigateur, une fenêtre indiquant le retour en mode non sécurisé peut apparaître.
 Ceci n'a aucun impact sur la confidentialité des informations précédemment échangées.

Copyright © 2009, all rights reserved

Figure 7 : texte non modifiable de la page de réponse en cas d'acceptation



Boutique de test

Votre paiement n'est pas accepté par votre établissement financier.

Nous regrettons de ne pas pouvoir donner une suite favorable à votre demande.

ANNULATION - RETOUR À LA BOUTIQUE

Copyright © 2009, all rights reserved

Figure 8 : texte non modifiable de la page de réponse en cas de refus

E. LOGOS DES BOUTONS

Trois boutons sont présents sur les pages de paiement. Ces logos présentent les textes suivants par défaut : « VALIDER », « ANNULATION – RETOUR A LA BOUTIQUE » et « RETOUR A LA BOUTIQUE » (ces textes sont traduits en fonction de la langue choisie). Le premier est présent sur la page de saisie du numéro de carte. Le second est visible sur la page de saisie du numéro de carte et sur la page de réponse en cas de refus de la transaction. Le troisième est localisé sur la page de réponse en cas d'acceptation de la transaction.

Ces logos par défaut peuvent être remplacés par les logos du commerçant. Pour ce faire il faut paramétrer les trois champs **submit_logo**, **cancel_return_logo** et **normal_return_logo**, transmis lors de la requête de paiement, avec le nom des logos du commerçant correspondant respectivement aux boutons « VALIDER », « ANNULATION – RETOUR A LA BOUTIQUE » et « RETOUR A LA BOUTIQUE ». Référez-vous au *DICIONNAIRE DES DONNEES* pour connaître les caractéristiques de ces trois champs et au *GUIDE DU PROGRAMMEUR* pour leur affectation.

**Tous les logos appelés lors de la requête de paiement doivent être
transmis par mail au Centre d'Assistance Technique.**

F. TEMPLATE PAR DEFAUT

Le template par défaut est identique pour toutes les pages du serveur Sips. Il est schématisé sur la Figure 9. Il permet l'affichage de trois logos (**advert**, **logo_id**, **logo_id2**), d'avoir au choix un logo de fond d'écran (**background_id**) ou un fond de couleur (**bgcolor**). Tous les logos ne sont pas obligatoires ; ne sont affichés que ceux renseignés dans la requête de paiement. La couleur du texte du serveur de paiement est également modifiable.



Figure 9 : schéma du template par défaut

G. LOGO EN HAUT AU CENTRE DE LA PAGE

Pour afficher un logo à cette position, le champ **advert**, transmis lors de la requête de paiement (cf. *DICIONNAIRE DES DONNEES*), doit être renseigné avec le nom du logo (exemple : bannière.gif). Le fichier du logo doit être transmis par mail au Centre d'Assistance Technique.

Pour un affichage optimal, nous conseillons un logo de 50px de hauteur maximum.

H. LOGO A GAUCHE DU NOM DE LA BOUTIQUE

Pour afficher un logo à cette position, le champ **logo_id**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), doit être renseigné avec le nom du logo (exemple : boutique.gif). Le fichier du logo doit être transmis par mail au Centre d'Assistance Technique.

Pour un affichage optimal, nous conseillons un logo de 250px de largeur / 100px de hauteur maximum.

1. LOGO A DROITE DU NOM DE LA BOUTIQUE

Pour afficher un logo à cette position, le champ **logo_id2**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), doit être renseigné avec le nom du logo (exemple : boutique.gif). Le fichier du logo doit être transmis par mail au Centre d'Assistance Technique.

Pour un affichage optimal, nous conseillons un logo de 100px de largeur / 100px de hauteur maximum.

2. LE NOM DE LA BOUTIQUE

Le nom de la boutique est paramétré sur le serveur. C'est une chaîne de caractères pouvant contenir du code HTML. Pour le modifier, le nouveau nom doit être transmis par mail au Centre d'Assistance Technique. Il peut ne pas être renseigné, si les logos suffisent.

3. FOND D'ECRAN

Le fond d'écran peut être paramétré avec un code couleur RGB ou un logo.

Pour afficher un logo en fond d'écran, le champ **background_id**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), doit être renseigné avec le nom du logo (exemple : fond_ecran.gif). Le fichier du logo doit être transmis par mail au Centre d'Assistance Technique.

Pour afficher une couleur en fond d'écran, le champ **bgcolor**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), doit être renseigné avec le code couleur RGB hexadécimal de votre choix (exemple : FF99FF). La couleur par défaut est le blanc. Des exemples de codes couleur RGB sont donnés en annexe A.

Si les champs **background_id** et **bgcolor** sont tous les deux renseignés, seul le **background_id** sera pris en compte.

4. COULEUR DU TEXTE DU SERVEUR MERC@NET

Pour paramétrer la couleur du texte, le champ **textcolor**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), doit être renseigné avec le code couleur RGB de votre choix (exemple : 009900). La couleur par défaut est le noir. Des exemples de codes couleur RGB sont donnés en annexe A.

1. CONTENU D'UN TEMPLATE

Un fichier template est un fichier HTML décrivant les pages de paiement.

Il doit contenir uniquement du code HTML.

Le tag [SIPS] doit être spécifié dans ce fichier pour indiquer l'endroit où le serveur Merc@net doit afficher les informations relatives à la transaction.

ATTENTION : le fichier template doit respecter les règles suivantes :

- Le serveur ajoute les balises HTML `<html>`, `<head>`, `<form>`, `</form>`, `</head>`, `</body>` et `</html>` sur les pages de paiement. Les templates ne doivent donc, en aucun cas, contenir ces balises.
- Le template ne doit pas comporter d'appel à des éléments (par exemple des images) localisés dans un sous dossier. Aucun sous dossier ne sera créé sur le serveur de production.
Exemple :
`` correct
`` incorrect
- Le template doit contenir le tag [SIPS]

Un template ne respectant pas ces conditions ne sera pas installé sur le serveur Merc@net.

Remarque

Il est recommandé de ne pas faire appel, dans les templates, à des images localisées sur un autre serveur. En effet, ce type d'appel peut conduire à un message d'alerte sur le navigateur de l'internaute précisant que les pages de paiement ne sont pas sécurisées. Ceci est dû au fait que la page n'est pas 100% sécurisée car elle fait appel à des éléments non sécurisés.

2. LIMITATIONS

Lorsqu'un fichier template est utilisé, les logos appelés dans les champs **advert**, **logo_id**, **logo_id2** et **background_id** ne sont plus affichés. De même, les codes couleur renseignés dans les champs **bgcolor** et **textcolor** ne sont plus pris en compte. Tous ces aspects graphiques doivent être pris en compte dans le template.

Par contre, les logos appelés dans les champs **submit_logo**, **cancel_return_logo** et **normal_return_logo** sont affichés par le serveur même si un template personnalisé est utilisé.

Le serveur Merc@net remplace le tag [SIPS] par une table et un formulaire. Cette table contient les informations sur l'identifiant du commerçant, la référence de la transaction et le montant de la transaction.

Vous ne pouvez donc pas modifier la taille et la police de toutes les informations affichées par le serveur car elles ne seront pas prises en compte dans la table.

Cependant, la couleur du texte des informations affichées par le serveur peut être modifiée. Il suffit pour cela d'utiliser l'attribut *color* de la balise HTML `<body>`. L'attribut *color* d'une balise HTML `` placée de part et d'autre du tag [SIPS] ne modifiera que le texte du formulaire, le texte inclus dans la table ne sera pas modifié.

Remarque

Le comportement peut être sensiblement différent en fonction des navigateurs. Il est donc conseillé de bien vérifier que son template fonctionne correctement sur les différents navigateurs.

3. UTILISATION D'UN TEMPLATE UNIQUE POUR L'ENSEMBLE DES PAGES DU SERVEUR

Après avoir créé votre template, vous devez le tester à l'aide de l'outil `test_template` (cf. Chapitre 3). Le fichier template doit ensuite être transmis, ainsi que tous les logos auxquels il fait appel, au Centre d'Assistance Technique. Le nom du fichier template peut ne comporter aucune extension.

Pour afficher le template sur les pages de paiement, le champ **templatefile**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), doit être renseigné avec le nom du template (exemple : `montemplate`).

Lors des transactions, le template sera utilisé pour l'affichage de toutes les pages du serveur, c'est à dire pour la saisie du numéro de la carte bancaire, pour la réponse, que la transaction soit acceptée ou refusée et pour la page d'informations 3-D Secure (dans le cadre du programme 3-D Secure).

Cas d'erreur :

Si votre template n'est pas installé sur nos serveurs ou si vous ne l'appellez pas avec le bon nom, une page d'erreur, écrite en rouge sur fond jaune, indiquant que la transaction est invalide sera affichée sur le navigateur de l'internaute après qu'il ait cliqué sur le logo de la carte de son choix.

4. UTILISATION D'UN TEMPLATE DIFFERENT POUR CHACUNE DES PAGES DU SERVEUR

Les pages du serveur peuvent être personnalisées avec trois ou quatre templates différents : Ces templates seront affichés :

- lors de la saisie des informations carte
- lors de la réponse pour une transaction acceptée
- lors de la réponse pour une transaction refusée
- avec la page d'informations 3-D Secure

Vous devez créer trois (ou quatre si nécessaire) templates correspondant à chacune des pages affichées par le serveur. Ces templates doivent porter le même nom, mais avec une extension différente en fonction de la page à laquelle ils sont rattachés :

- extension `.CAPTURE` pour la page de saisie des informations carte
- extension `.ACCEPTED` pour la page de réponse lorsque la transaction est acceptée
- extension `.REFUSED` pour la page de réponse lorsque la transaction est refusée
- extension `.3DINFO` pour la page d'information 3-D Secure

Attention : les extensions sont obligatoirement en majuscule.

Vous allez donc disposer de trois (ou quatre) fichiers nommés par exemple `montemplate.CAPTURE`, `montemplate.ACCEPTED` et `montemplate.REFUSED` (et éventuellement `montemplate.3DINFO`). Après les avoir testés à l'aide de l'outil `test_template` (cf. Chapitre 3), vous devez les transmettre ainsi que tous les logos auxquels ils font appel, au [Centre d'Assistance Technique](#).

Pour afficher les templates sur les pages de paiement, le champ **templatefile**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), doit être renseigné avec le nom des templates sans l'extension (*montemplate* dans notre exemple).

Comment se déroulera la transaction ?

Lors de la transaction le serveur Merc@net détectera automatiquement les templates à afficher. Il affichera le template *montemplate.CAPTURE* pour la saisie du numéro de la carte, éventuellement le template *montemplate.3DINFO* dans le cas d'une transaction de type 3-D Secure, puis le template *montemplate.ACCEPTED* si la transaction est acceptée, ou le template *montemplate.REFUSED* si elle est refusée.

Particularité liée à 3-D Secure :

Pour afficher la page d'information 3-D Secure, le serveur utilise par défaut le template de la page de saisie du numéro de carte si vous n'avez pas créé de template suffixé par *3DINFO*.

Cas d'erreur :

Si votre template n'est pas installé sur nos serveurs ou si vous ne l'appellez pas avec le bon nom, une page d'erreur, écrite en rouge sur fond jaune, indiquant que la transaction est invalide sera affichée sur le navigateur de l'internaute après qu'il ait cliqué sur le logo de la carte de son choix.

5. LES TEMPLATES EN PLUSIEURS LANGUES

Vos templates peuvent contenir du texte, et si votre site est traduit en plusieurs langues, vous allez devoir aussi traduire vos templates. Vous disposerez d'autant de templates que de langues. Pour les distinguer, il faudra les nommer différemment, par exemple ajouter l'extension « .fr » pour les templates traduits français et « .en » pour les templates en anglais.

Si on reprend l'exemple précédent, vous allez donc disposer de 6 fichiers : *montemplate.fr.CAPTURE*, *montemplate.en.CAPTURE*, *montemplate.fr.ACCEPTED*, *montemplate.en.ACCEPTED*, *montemplate.fr.REFUSED* et *montemplate.en.REFUSED*.

Si vous êtes inscrit au programme 3D, vous disposerez de 2 autres fichiers : *montemplate.fr.3DINFO* et *montemplate.en.3DINFO*.

Pour faire afficher le template français ou anglais, il faut renseigner le champ **templatefile** de l'API avec le nom du fichier template voulu (cf. *DICTIONNAIRE DES DONNEES*). Dans notre exemple, pour faire afficher les templates anglais, il faudra renseigner ce champ à « *montemplate.en* ».

J. PERSONNALISATION PAR CSS

Aujourd'hui, l'intégration des feuilles de styles CSS dans les pages Merc@net apporte une plus grande souplesse de personnalisation :

- Personnalisation plus large
- Possibilité de tester en temps réel les modifications apportées (évitant de nombreux allers-retours vers le Centre d'Assistance Technique)
- Configuration simple et rapide de l'API Merc@net

La personnalisation par CSS n'est cependant pas compatible avec l'utilisation en parallèle d'une personnalisation par Template. Il est impossible de faire cohabiter ces deux méthodes de personnalisation.

1. SIMPLICITE DE FONCTIONNEMENT

La feuille de style doit être hébergée sur le serveur Merc@net.

a. Règle de nommage et normes associées

La feuille de style créée, doit être présentée sous la forme d'un fichier d'extension CSS (nom_du_fichier.css).

Elle doit respecter les règles CSS de développement, et leur normes associées (ex. « ISO-8859-1 »).

Exemple :

```
@CHARSET "ISO-8859-1";
```

Remarque

Aucune personnalisation par CSS ne sera possible, si la personnalisation par Template est activée dans les champs de la requête fournis par l'API.

b. Hébergement sur le serveur Merc@net

Vous effectuerez vos tests en appliquant la méthode précédente par le positionnement de votre CSS dans le champ DATA de la requête.

Une fois figé, le fichier CSS sera transmis par mail au Centre d'Assistance Technique pour être intégré dans la solution Merc@net. Ensuite, les paramètres déclarés dans le champ DATA de la requête devront être supprimés (l'url déclarée étant prioritaire à la css proposée par la solution Merc@net).

Si vous souhaitez utiliser d'autres CSS (ex : ajout d'une CSS pour smartphone), vous pouvez nous les transmettre également. Dans ce cas, il faudra toujours positionner le nom de la CSS dans le champ DATA de la requête de la manière suivante :

```
<USE_CSS> ; nom_du_fichier_css</USE_CSS> ;
```

Remarque

Attention, le nom du fichier CSS ne doit pas contenir d'espace, sinon la personnalisation ne serait pas prise en compte.

2. PAGES MERC@NET A PERSONNALISER

La personnalisation CSS est effective sur la totalité des pages affichées par l'offre Merc@net :

- la page de saisie des coordonnées bancaires (page de capture)
- la page d'accord
- la page de refus
- la page d'attente pour transaction en cours de traitement
- l'affichage des messages liés aux erreurs de saisies

Tous les boutons présents sur les différentes pages du serveur peuvent également être personnalisés. Cependant, le texte contenu dans ces pages ne peut pas être modifié.

Les différentes pages Merc@net à personnaliser sont présentées en détails dans l'annexe C.

Attention :

La page de saisie des coordonnées bancaires diffère en fonction du moyen de paiement sélectionné par l'internaute.

a. Utilisation des classes CSS

La personnalisation CSS des pages Merc@net est basée sur l'implémentation d'un certain nombre de classes CSS attribuées sur une multitude de champs (messages, erreurs, champs de saisie, titres, etc,...).

La définition d'une classe doit respecter le format suivant :

```
.classe_test {  
    Attribut1: valeur1;  
    Attribut2: valeur2;  
    .  
    .  
    AttributN: valeurN ;  
}
```

b. Notion d'héritage CSS

Le rôle de ce paragraphe n'est pas de présenter les avantages et les possibilités qui sont offerts par les héritages CSS, mais de rappeler que cette notion peut être employée en complément des classes prédéfinies sur les pages Merc@net.

En utilisant cet héritage, les tags « fils » hériteront des propriétés du tag « père ».

Ainsi, les tags <tr>, <td> (etc,...), dont l'élément parent est <table> et n'ayant pas de classes prédéfinies, pourront être personnalisés par cette méthode.

c. Gestion des erreurs

Lors de vos tests, si la personnalisation par CSS n'est pas prise en compte, veuillez vérifier les points suivants :

- Vérifier que le champ TEMPLATE_FILE est bien vide
- Vérifier qu'une autre CSS n'est pas indiquée dans le champ DATA (prioritaire par rapport à la CSS figé sur nos serveurs Merc@net)

K. ADAPTATION DES PAGES DE PAIEMENT POUR UN SMARTPHONE

Jusqu'à maintenant, les pages de paiement sur smartphone nécessitaient l'utilisation du zoom pour être visualisées.

En fonction des moyens de paiements acceptés par le commerçant, cinq types de pages web sont affichées sur le navigateur d'un smartphone :

- Page de saisie des données bancaires
- Page d'acceptation de l'ordre de paiement, si celui-ci a été accepté par son établissement bancaire auquel il est rattaché
- Pages de rejet en cas d'erreur lors de la saisie des données bancaires, ou de refus de son établissement bancaire
- Page présentant un message « Transaction déjà traitée »
- Page présentant un message « Transaction en cours de traitement »
- Page d'erreur

1. UTILISATION D'UNE CSS DEDIEE AU SMARTPHONE

Pour un meilleur confort visuel lors de la navigation sur un mobile, vous devrez nous transmettre une CSS personnalisée smartphone.

Voici un exemple de l'affichage de la page de saisie des coordonnées bancaires avec une CSS spéciale smartphone :

The screenshot shows a mobile browser interface. At the top, the status bar indicates 'Orange F 3G' and '10:18'. The browser address bar shows 'rcet-bank.sips-atos.co...' and a 'Google' search bar. The main heading is 'BOUTIQUE DE DEMO IPHONE' with a 'VISA' logo to the right. Transaction details are listed: 'Identifiant commerçant' (038862749866661), 'Référence de la transaction' (094048), and 'Montant de la transaction' (1,00 €). The payment form includes: 'N° de carte :', 'Expire fin : 01-Janvier / 2009', and 'N° de contrôle au dos de la carte :'. A confirmation message states 'Vous avez complété correctement le formulaire, vous pouvez'. At the bottom are two buttons: 'VALIDER' and 'ANNULATION - RETOUR A LA BOUTIQUE'. The phone's navigation bar at the very bottom shows standard Android icons.

2. UTILISATION DE LA PERSONNALISATION PAR CSS

Le commerçant devra détecter quel type de navigateur l'internaute utilise (smartphone ou web classique).

La feuille de style étant hébergée sur le serveur Merc@net, vous effectuerez vos tests par le positionnement de votre CSS dans le champ DATA de la requête de la manière suivante :

```
<USE_CSS> ; nom_du_fichier_css_smartphone</USE_CSS> ;
```

L. AUTRES PARAMETRAGES DE LA PERSONNALISATION

Les deux champs **receipt_complement** et **data** peuvent être renseignés pour compléter la personnalisation des pages du serveur Merc@net.

1. LA PERSONNALISATION DU TICKET (CHAMP RECEIPT_COMPLEMENT)

Le champ **receipt_complement**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), peut être renseigné avec du code HTML. Ce code HTML sera affiché sur la page de réponse dans le cas d'une transaction acceptée. A titre d'exemple, la figure 10 représente la page de réponse d'une transaction acceptée pour laquelle nous avons renseigné le champ **receipt_complement** à « `<tr><td><i>texte supplémentaire</i></td></tr>` ». Le « *texte supplémentaire* » s'affiche entre la référence et le montant de la transaction.



Figure 10

M. AUTRES OPTIONS D’AFFICHAGE (CHAMP DATA)

Le champ **data**, transmis lors de la requête de paiement (cf. *DICTIONNAIRE DES DONNEES*), peut être renseigné avec différents mots clés. Si vous souhaitez utiliser plusieurs mots clés dans le champ data, vous devez les séparer par un point virgule (exemple : `CARD_NO_LOGO;NO_COPYRIGHT`).

CARD_NO_LOGO

Ce mot clé supprime l’affichage du logo de la carte bancaire sur la page de saisie des coordonnées bancaires de l’internaute.

NO_COPYRIGHT

Ce mot clé supprime la ligne « *Copyright © 2004, all rights reserved* » de toutes les pages affichées par le serveur Merc@net.

NO_DISPLAY_CARD

Ce mot clé supprime le réaffichage du numéro de la carte bancaire si l'internaute revient sur la page de saisie des coordonnées bancaires en cliquant sur le bouton « BACK » de son navigateur. Cette option est conseillée si les internautes font partie d'une population utilisant des ordinateurs en libre service (Etudiants d'université, clients de cyber-café,...)

NO_DISPLAY_CANCEL

Ce mot clé supprime l'affichage du bouton « ANNULATION – RETOUR A LA BOUTIQUE » de la page de saisie du numéro de carte.

NO_SSL_SYMBOLS

Ce mot clé remplace la ligne contenant les symboles de sécurisation SSL au dessus du cadre de saisie de la carte, par la phrase « *Vous êtes connecté à un serveur sécurisé, vous pouvez remplir votre formulaire en toute confiance* ». Vous pouvez activer cette option lorsque les pages de paiement s'affichent à l'intérieur de frames, car dans ce cas le navigateur n'affiche plus, en bas à gauche, les symboles signalant que vous êtes sur une page sécurisée.

NO_WINDOWS_MSG

Ce mot clé supprime l'affichage, sur la page d'acceptation, de la phrase : « *Suivant le paramétrage de votre navigateur, une fenêtre indiquant le retour en mode non sécurisé peut apparaître. Ceci n'a aucun impact sur la confidentialité des informations précédemment échangées.* »

NO_DISPLAY_URL

Ce mot clé supprime l'affichage de l'url sur la page d'acceptation.

NO_RESPONSE_PAGE

Ce mot clé supprime l'affichage de la page de réponse à l'internaute. Il est ainsi directement reconnecté à la boutique du commerçant, sur l'URL paramétrée dans le champ **normal_return_url** ou **cancel_return_url** suivant que la transaction a été acceptée ou refusée.

ATTENTION :

L'utilisation de cette option modifie le protocole de la réponse renvoyée sur les URL paramétrées dans les champs **normal_return_url** et **cancel_return_url**. La réponse cryptée n'est plus envoyée en méthode POST, mais en méthode GET. La méthode GET ne permettant pas de véhiculer de grosses variables, tous les champs de la réponse ne sont pas renvoyés.

Les champs non transmis sont :

caddie, customer_email, customer_id, customer_ip_address, merchant_language, order_validity, receipt_complement, return_context et transaction_condition.

Pour plus d'information sur ces champs référez-vous au *DICTIONNAIRE DES DONNEES*.

La réponse automatique reste, elle, inchangée. Tous les champs de la réponse sont renvoyés sur l'URL paramétrée dans le champ **automatic_response_url** en méthode POST.

3. L'OUTIL TEST_TEMPLATE

A. GENERALITES

Cet outil simule les affichages du serveur Merc@net. Il vous permet de visualiser vos pages de paiement personnalisées. Cette simulation d'affichage peut être légèrement différente par rapport au serveur Merc@net, la validation de vos pages de personnalisation doit donc être faite sur le serveur lors de la pré-production.

Pour utiliser l'outil test_template, vous devez décompresser le fichier test_template.tar sur votre poste de travail (avec Winzip par exemple). Deux sous-dossiers sont alors créés : test_template_windows et test_template_java. Le contenu de ces sous-dossiers et les commandes à utiliser pour exécuter les test_template version Windows ou Java sont décrits dans les paragraphes suivants.

L'outil test_template affiche les pages de paiement sur la sortie standard. En cas d'erreur dans le template, il n'affiche pas la page de paiement mais un message d'erreur.

B. OUTIL TEST_TEMPLATE VERSION WINDOWS

Cette version est basée sur un exécutable compilé sous environnement Windows. Si vous utilisez un autre système d'exploitation et/ou utilisez la personnalisation par css, vous devez utiliser la version Java décrite dans le paragraphe suivant.

Liste des objets livrés

test_template.exe	Exécutable permettant la simulation des pages de paiement
VISA.gif	Logo de la carte VISA
VISA_3D.gif	Logo 3-D Secure VISA
CLEF.gif	Logo de la clé sécurisée
cryptogramme.fr.html	Popup d'information sur le cryptogramme visuel.
cartecvv.jpg	Logo appelé par le popup d'information sur le cryptogramme visuel.
merchant_template	Exemple de template
merchant.gif	Exemple de logo de boutique
cancel_logo.gif	Exemple de bouton d'annulation de la transaction
return_logo.gif	Exemple de bouton pour le retour à la boutique en cas d'acceptation de la transaction.
submit_logo.gif	Exemple de bouton de validation de la transaction

Exécution du test_template.exe

Pour exécuter le fichier test_template.exe, vous devez ouvrir une fenêtre permettant de lancer des commandes en ligne (fenêtre MS-DOS ou Invite de commandes). Une fois dans cette fenêtre, vous devez vous déplacer jusqu'au dossier test_template_windows. Ensuite, vous devez taper la ligne de commande suivante :

`test_template nom_du_template page_du_serveur`

avec :

nom_du_template le nom du fichier template que vous souhaitez tester

page_du_serveur mot clé indiquant la page du serveur affichée. Les différents mots clés et les pages associées sont décrits dans le tableau ci-dessous.

Mot clé	Page du serveur
CARD	Page de saisie du numéro de carte pour une carte présentant un cryptogramme visuel (Pour plus d'information sur le cryptogramme visuel reportez-vous au document DICTIONNAIRE DES DONNEES).
CARD3D	Page de saisie du numéro de carte pour une carte présentant un cryptogramme visuel dans le cadre du programme 3-D Secure.
CARDREQFACCVV	Page de saisie du numéro de carte pour une carte présentant un cryptogramme visuel requis ou facultatif.
CARDREQFACCVV3D	Page de saisie du numéro de carte pour une carte présentant un cryptogramme visuel requis ou facultatif dans le cadre du programme 3-D Secure.
CARDLIGHTCVV	Page de saisie du numéro de carte pour une carte présentant un cryptogramme visuel light.
CARDLIGHTCVV3D	Page de saisie du numéro de carte pour une carte présentant un cryptogramme visuel light dans le cadre du programme 3-D Secure.
CARDNOCVV	Page de saisie du numéro de carte pour une carte ne présentant pas de cryptogramme visuel.
CARDNOCVV3D	Page de saisie du numéro de carte pour une carte ne présentant pas de cryptogramme visuel dans le cadre du programme 3-D Secure.
ACCEPTED	Page de réponse en cas d'acceptation de la transaction.
REFUSED	Page de réponse en cas de refus de la transaction.
3DINFO	Page d'information 3-D Secure

Tableau 1: liste des mots clés et des pages associées

Le résultat de la commande précédente sera affiché sur la sortie standard. Pour créer une page HTML, vous devez rediriger la sortie standard vers une page HTML. Avec la commande suivante :

```
test_template merchant_template CARDNOCVV > test1.html
```

Vous créerez une page HTML test1.html comportant la page de saisie du numéro de carte pour une carte ne présentant pas de cryptogramme visuel.

Pour visualiser le résultat vous devez ouvrir, avec votre navigateur, la ou les page(s) HTML ainsi créée(s).

Le template merchant_template est fourni comme exemple dans le dossier test_template_windows.

Lorsque vous testerez vos propres templates, vous devrez changer merchant_template par le nom de vos templates dans la ligne de commandes.

C. OUTIL TEST_TEMPLATE TOUT SYSTEME D'EXPLOITATION (JAVA)

Cette version est basée sur une classe test_template.class. Vous pouvez donc utiliser cette classe sur tout système d'exploitation pouvant supporter une JVM (Java Virtual Machine)

Liste des objets livrés :

Sous le répertoire principal

test_template.class	Classe permettant la simulation des pages de paiement
merchant_template	Exemple de template
merchant_template_tabcurrency	Exemple de template avec affichages tableau de taux des devises
default_sips_payment_perso.css	Exemple de css par défaut sur le serveur Merc@net

merchant.gif	Exemple de logo de boutique
cancel_logo.gif	Exemple de bouton d'annulation de la transaction
return_logo.gif	Exemple de bouton pour le retour à la boutique en cas d'acceptation de la transaction.
submit_logo.gif	Exemple de bouton de validation de la transaction
advert.jpg	Exemple de bannière de publicité

Sous le répertoire country_flags/

Images des drapeaux des pays du taux des devises affichées sur le serveur à ne pas modifier, ni déplacer.

Sous le répertoire img/

Images, pages sur le serveur à ne pas modifier, ni déplacer.

4DBC.en.html	Popup d'information sur le numéro de contrôle en anglais
4DBC.fr.html	Popup d'information sur le numéro de contrôle en français
AMEX.gif	Logo de la carte AMERICAN EXPRESS
AMEX_4DBC.gif	Image du numéro de contrôle de la carte AMERICAN EXPRESS
cartecvv.jpg	Logo appelé par le popup d'information sur le cryptogramme visuel.
CLEF.gif	Logo de la clé sécurisée
cryptogramme.en.html	Popup d'information sur le cryptogramme visuel en anglais.
cryptogramme.fr.html	Popup d'information sur le cryptogramme visuel en français.
default_sips_payment_background.png	Image utilisées dans la css par défaut default_sips_payment_perso.css
default_sips_payment_puce_black.png	Image utilisées dans la css par défaut default_sips_payment_perso.css
default_sips_payment_puce_red.png	Image utilisées dans la css par défaut default_sips_payment_perso.css
INTERVAL.gif	Image
style_default.css	css pour affichage des montants en devises avec un template
VISA.gif	Logo de la carte VISA
VISA_3D.gif	Logo 3-D Secure VISA

Sous le répertoire 4test/

merchant2.gif	Exemple de logo de boutique 2 à copier au niveau de test_template.class afin de tester l'affichage du logo2
---------------	---

Sous le répertoire properties/

message_fr.properties	Messages affichés sur les pages en français
message_en.properties	Messages affichés sur les pages en anglais

Exécution du test_template :

Pour exécuter la classe test_template.class, vous devez ouvrir une fenêtre permettant de lancer des commandes en ligne. Une fois dans cette fenêtre, vous devez vous déplacer jusqu'au dossier test_template_java. Ensuite, vous devez taper la ligne de commande suivante :

```
java test_template nom_du_template page_du_serveur
```

avec :

nom_du_template le nom du fichier template que vous souhaitez tester

page_du_serveur mot clé indiquant la page du serveur affichée. Les différents mots clés et les pages associées sont décrits dans le Tableau 1 du paragraphe précédent et les suivants.

Mot clé	Page du serveur
AMEX	Page de saisie du numéro de carte pour une carte American Express.
AMEX4DBC	Page de saisie du numéro de carte pour une carte American Express présentant un numéro de contrôle.

Le résultat de la commande précédente sera affiché sur la sortie standard. Pour créer une page HTML, vous devez rediriger la sortie standard vers une page HTML. Avec la commande suivante :

```
java test_template merchant_template CARDNOCVV > test1.html
```

Vous créez une page HTML test1.html comportant la page de saisie du numéro de carte pour une carte ne présentant pas de cryptogramme visuel.

Pour visualiser le résultat vous devez ouvrir, avec votre navigateur, la ou les page(s) HTML ainsi créée(s).

Remarque

La commande java dans la ligne précédente, doit être précédée, si nécessaire, du chemin vers le répertoire bin de votre JDK

Le template merchant_template est fourni comme exemple dans le dossier test_template_java.

Lorsque vous testerez vos propres templates, vous devrez changer merchant_template par le nom de vos templates dans la ligne de commandes.

D. OPTIONS DISPONIBLES

Des options d’affichage disponibles sur le serveur le sont également avec l’outil test_template :

- Vous pouvez remplacer les boutons par défaut présents sur les pages de paiement par des logos de votre choix (cf. paragraphe E).
- Vous pouvez choisir de n’afficher ni le copyright en bas de toutes les pages de paiement, ni le logo de la carte sur la page de saisie du numéro de carte (cf. paragraphe L).

Ces options peuvent être précisées dans n’importe quel ordre, mais elles doivent impérativement être positionnées après le type de page souhaité (CARD, CARD3D, CARDNOCVV, CARDNOCVV3D, CARDLIGHTCVV, CARDLIGHTCVV3D, CARDREQFACCVV, CARDREQFACCVV3D, 3DINFO, ACCEPTED ou REFUSED).

-c cancel_logo.gif

Cette option vous permet d’afficher le logo cancel_logo.gif à la place du bouton par défaut ANNULATION – RETOUR A LA BOUTIQUE présent sur la page de saisie du numéro de carte (CARD, CARD3D, CARDNOCVV, CARDNOCVV3D, CARDLIGHTCVV, CARDLIGHTCVV3D, CARDREQFACCVV ou CARDREQFACCVV3D) et sur la page de réponse dans le cas d’un refus (REFUSED).

-s submit_logo.gif

Cette option vous permet d'afficher le logo submit_logo.gif à la place du bouton par défaut VALIDER présent sur la page de saisie du numéro de carte (CARD, CARD3D, CARDNOCVV, CARDNOCVV3D, CARDLIGHTCVV, CARDLIGHTCVV3D, CARDREQFACCVV ou CARDREQFACCVV3D).

-r return_logo.gif

Cette option vous permet d'afficher le logo return_logo.gif à la place du bouton par défaut RETOUR A LA BOUTIQUE présent sur la page de réponse dans le cas d'acceptation (ACCEPTED).

-CARD_NO_LOGO

Cette option vous permet de supprimer l'affichage du logo de la carte bancaire sur la page de saisie du numéro de carte (CARD, CARD3D, CARDNOCVV, CARDNOCVV3D, CARDLIGHTCVV, CARDLIGHTCVV3D, CARDREQFACCVV ou CARDREQFACCVV3D) et sur la page d'informations 3-D Secure (3DINFO).

-NO_COPYRIGHT

Cette option vous permet de supprimer l'affichage du copyright situé au pied de toutes les pages de paiement.

-NO_DISPLAY_CANCEL

Cette option vous permet de supprimer l'affichage du bouton ANNULATION – RETOUR A LA BOUTIQUE sur la page de saisie du numéro de carte (CARD, CARD3D, CARDNOCVV, CARDNOCVV3D, CARDLIGHTCVV, CARDLIGHTCVV3D, CARDREQFACCVV ou CARDREQFACCVV3D).

-NO_SSL_SYMBOLS

Cette option vous permet de supprimer l'affichage des symboles de chiffrement SSL sur la page de saisie du numéro de carte (CARD, CARD3D, CARDNOCVV, CARDNOCVV3D, CARDLIGHTCVV, CARDLIGHTCVV3D, CARDREQFACCVV ou CARDREQFACCVV3D).

-NO_WINDOWS_MSG

Cette option vous permet de supprimer l'affichage de la phrase : « *Suivant le paramétrage de votre navigateur, une fenêtre indiquant le retour en mode non sécurisé peut apparaître. Ceci n'a aucun impact sur la confidentialité des informations précédemment échangées.* » sur la page d'acceptation (ACCEPTED).

-NO_DISPLAY_URL

Cette option vous permet de supprimer l'affichage de l'url sur la page d'acceptation.

Quelques exemples :

Pour Windows :

```
test_template merchant_template ACCEPTED -r return_logo.gif > test1.html
```

Pour Java :

```
java test_template merchant_template ACCEPTED -r return_logo.gif > test1.html
```

Créera une page HTML test1.html comportant la page de réponse en cas d'acceptation de la transaction. Sur cette page, apparaîtra le bouton return_logo.gif.

Pour Windows :

```
test_template merchant_template CARDNOCVV -s submit_logo.gif -c cancel_logo.gif -CARD_NO_LOGO -NO_COPYRIGHT > test2.html
```

Pour Java :

```
java test_template merchant_template CARDNOCVV -s submit_logo.gif -c cancel_logo.gif -CARD_NO_LOGO -NO_COPYRIGHT > test2.html
```

Créera une page HTML test2.html comportant la page de saisie du numéro de carte pour une carte ne présentant pas de cryptogramme visuel. Sur cette page apparaîtront les boutons submit_logo.gif et cancel_logo.gif, par contre, le logo de la carte et le copyright ne seront pas affichés.

Les boutons cancel_logo.gif, submit_logo.gif et return_logo.gif sont fournis comme exemple dans les dossiers test_template_windows et test_template_java.

Lorsque vous testerez vos propres boutons, vous devrez indiquer leurs noms après les options -c, -s et -r.

Vous pouvez combiner ces différentes options. Dans les tableaux ci-dessous sont précisées les options disponibles en fonction des pages de paiement.

Type de page	-c	-s	-r
CARD	✓	✓	✗
CARD3D	✓	✓	✗
CARDNOCVV	✓	✓	✗
CARDNOCVV3D	✓	✓	✗
CARDREQFACCVV	✓	✓	✗
CARDREQFACCVV3D	✓	✓	✗
CARDLIGHTCVV	✓	✓	✗
CARDLIGHTCVV3D	✓	✓	✗
AMEX	✓	✓	✗
AMEX4DBC	✓	✓	✗
ACCEPTED	✗	✗	✓
REFUSED	✓	✗	✗
3DINFO	✗	✗	✗

Type de page	-NO_WINDOWS_MSG	-CARD_NO_LOGO	-NO_COPYRIGHT
CARD	✗	✓	✓
CARD3D	✗	✓	✓
CARDNOCVV	✗	✓	✓
CARDNOCVV3D	✗	✓	✓
CARDREQFACCVV	✗	✓	✓
CARDREQFACCVV3D	✗	✓	✓
CARDLIGHTCVV	✗	✓	✓
CARDLIGHTCVV3D	✗	✓	✓
AMEX	✗	✓	✓
AMEX4DBC	✗	✓	✓
ACCEPTED	✓	✗	✓
REFUSED	✗	✗	✓
3DINFO	✗	✓	✓

Type de page	-NO_DISPLAY_CANCEL	-NO_SSL_SYMBOLS	-NO_DISPLAY_URL
CARD	✓	✓	✗
CARD3D	✓	✓	✗
CARDNOCVV	✓	✓	✗
CARDNOCVV3D	✓	✓	✗
CARDREQFACCVV	✓	✓	✗
CARDREQFACCVV3D	✓	✓	✗
CARDLIGHTCVV	✓	✓	✗
CARDLIGHTCVV3D	✓	✓	✗
AMEX	✓	✓	✗
AMEX4DBC	✓	✓	✗
ACCEPTED	✗	✗	✓
REFUSED	✗	✗	✗
3DINFO	✗	✗	✗

✓ Option active ✗ Option indisponible

Option seulement valable sur la version java de l'outil test_template avec la personnalisation via css :

-a advert.jpg Test de la bannière de publicité.

Test du logo 2 de la boutique : copier le logo merchant2.gif du répertoire 4test vers le répertoire principal où se trouve l'outil test_template ou déposer dans ce répertoire un logo nommé merchant2.gif. Lors de la génération des pages avec le mode de personnalisation css le logo sera pris en compte.

4. ANNEXES

A. ANNEXE A : TABLEAU DES CODES RGB

C'est la codification hexadécimale du type RGB (RED GREEN BLUE), soit rrggbb où rr, gg et bb sont des valeurs hexadécimales correspondant respectivement au rouge vert et bleu.

000000	000030	000070	0000CC	0000FF
300000	300030	300070	3000CC	3000FF
003000	003030	003070	0030CC	0030FF
303000	303030	303070	3030CC	3030FF
700000	700030	700070	7000CC	7000FF
703000	703030	703070	7030CC	7030FF
007000	007030	007070	0070CC	0070FF
307000	307030	307070	3070CC	3070FF
707000	707030	707070	7070CC	7070FF
00CC00	00CC30	00CC70	00CCCC	00CCFF
30CC00	30CC30	30CC70	30CCCC	30CCFF
70CC00	70CC30	70CC70	70CCCC	70CCFF
00FF00	00FF30	00FF70	00FFCC	00FFFF
30FF00	30FF30	30FF70	30FFCC	30FFFF
70FF00	70FF30	70FF70	70FFCC	70FFFF
CC0000	CC0030	CC0070	CC00CC	CC00FF
FF0000	FF0030	FF0070	FF00CC	FF00FF
CC3000	CC3030	CC3070	CC30CC	CC30FF
FF3000	FF3030	FF3070	FF30CC	FF30FF
CC7000	CC7030	CC7070	CC70CC	CC70FF
FF7000	FF7030	FF7070	FF70CC	FF70FF
CCCC00	CCCC30	CCCC70	CCCCCC	CCCCFF
FFCC00	FFCC30	FFCC70	FFCCCC	FFCCFF
CCFF00	CCFF30	CCFF70	CCFFCC	CCFFFF
FFFF00	FFFF30	FFFF70	FFFFCC	FFFFFF

B. ANNEXE B : EXEMPLES DE CODE POUR LA PERSONNALISATION PAR CSS

- Exemple (RequestServlet.java)

```
/* Initialisation de l'objet d'appel */
SIPSDaObject call = (SIPSDaObject) new SIPSCallParm();

...

call.setValue("data", "<USE_CSS>https://www.test.fr/css/fichier.css</USE_CSS>");

...

// Appel de l'api Sips payment
out.println(api.sipsPaymentCallFunc(call));
```

- Exemple (cgi_request.c)

```
/* Initialisation de la structure d'appel */
sips_init_func(&param);

...

/* Affectation des paramètres obligatoires */
strcpy(param.merchant_id, "XXXXXXXXXX");
strcpy(param.merchant_country, "YY");
strcpy(param.amount, "ZZZZ");

...

/* génération automatique d'une transaction id */
sips_transaction_id_func ( param.transaction_id );

...

/* Affecte dynamiquement les autres paramètres */
strcpy(param.data, "<USE_CSS>https://www.test.fr/css/fichier.css</USE_CSS>");

...

/* Appel de l'api Sips */
resultat=sips_remote_call_func( &param, pathfile, &buffer, &error);
```

- Exemple (call_request.php)

```
// Affectation des paramètres obligatoires
$parm="merchant_id=XXXXXXXXXX";
$parm="$parm merchant_country=YY";
$parm="$parm amount=ZZZZ";
$parm="$parm currency_code=978";

...

// Initialisation du chemin pathfile
$parm="$parm pathfile=chemin_du_fichier_pathfile";

...

$parm="$parm data=<USE_CSS>https://www.test.fr/css/fichier.css</USE_CSS>";

...

// Appel du request binaire
$result=exec("$path_bin $parm");
```

1. SAISIE DES COORDONNEES BANCAIRES

The screenshot shows a web form titled "BOUTIQUE DE DEV" for entering banking coordinates. The form is annotated with various CSS class names pointing to specific elements:

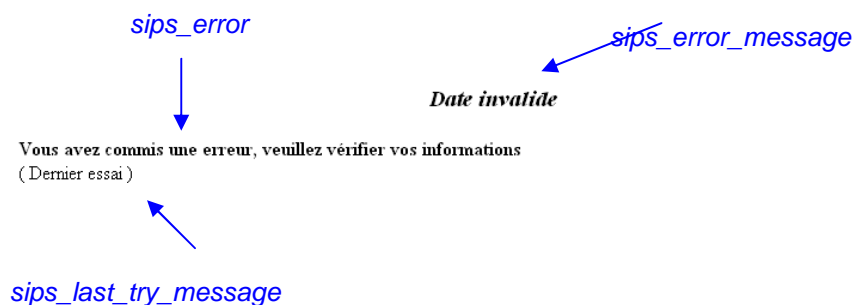
- Page Structure:** `sips_body_page` (main container), `sips_logo` (EB, EUROCARD, MasterCard, VISA logos).
- Transaction Details:**
 - `sips_trans_ref_table` (table containing transaction details)
 - `sips_trans_ref_table_id_transaction` (transaction ID)
 - `sips_trans_ref_table_id_customer` (customer ID)
 - `sips_trans_ref_table_amount_transaction` (transaction amount)
 - `sips_trans_ref_table_id_transaction_result` (transaction result ID)
 - `sips_trans_ref_table_id_customer_result` (customer result ID)
 - `sips_trans_ref_table_amount_transaction_result` (transaction result amount)
- Shop Information:** `sips_shop_name_table` (shop name), `sips_shop_name_font` (shop name font).
- Security Message:** `sips_secure_message` (message about secure transaction).
- Card Information:**
 - `sips_card_number_title` (card number title)
 - `sips_card_number_input` (card number input)
 - `sips_expiry_date_card_title` (expiry date title)
 - `sips_expiry_year_card_input` (expiry year input)
 - `sips_expiry_month_card_input` (expiry month input)
 - `sips_cryptogram_code_input` (cryptogram code input)
- Buttons and Messages:**
 - `sips_submit_message` (submit message)
 - `sips_submit_button` (VALIDER button)
 - `sips_customer_return_button` (ANNULATION - RETOUR A LA BOUTIQUE button)
 - `sips_popup_message` (popup message)
 - `sips_footer_message` (copyright notice)

Tableau récapitulatif des classes positionnées :

Tag positionné	Nom de classe
<body>	sips_body_page
<table>	sips_shop_name_table
	sips_logo
<table>	sips_trans_ref_table
	sips_trans_ref_table_id_customer
	sips_trans_ref_table_id_transaction
	sips_trans_ref_table_amount_transaction
<i>	sips_trans_ref_table_id_customer_result
<i>	sips_trans_ref_table_id_transaction_result
<i>	sips_trans_ref_table_amount_transaction_result
	sips_shop_name_font
	sips_secure_message
	sips_card_number_title
<input>	sips_card_number_input
	sips_expiry_date_card_title
<select>	sips_expiry_month_card_input
<select>	sips_expiry_year_card_input
<input>	sips_cryptogram_code_input

<a>	sips_cryptogram_example
	sips_submit_message
<input>	sips_submit_button
<input>	sips_customer_return_button
<i>	sips_footer_message

Gestion des erreurs de saisies :



Tag positionné	Nom de classe
	sips_error_message
	sips_error
	sips_last_try_message

2. CAS D'UN PAIEMENT PAR CARTE AMERICAN EXPRESS

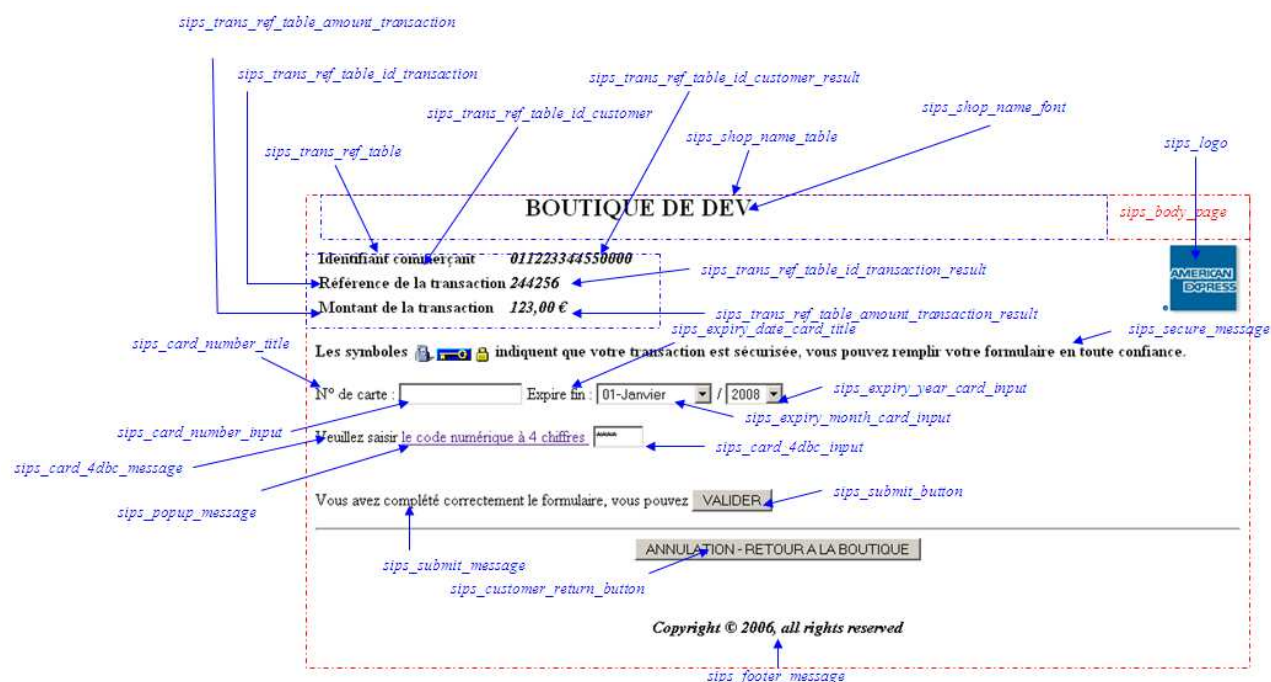


Tableau récapitulatif des classes positionnées :

Tag positionné	Nom de classe
<body>	sips_body_page
<table>	sips_shop_name_table
	sips_logo
<table>	sips_trans_ref_table
	sips_trans_ref_table_id_customer
	sips_trans_ref_table_id_transaction
	sips_trans_ref_table_amount_transaction
<i>	sips_trans_ref_table_id_customer_result
<i>	sips_trans_ref_table_id_transaction_result
<i>	sips_trans_ref_table_amount_transaction_result
	sips_shop_name_font
	sips_secure_message
	sips_card_number_title
<input>	sips_card_number_input
	sips_expiry_date_card_title
<select>	sips_expiry_month_card_input
<select>	sips_expiry_year_card_input
	sips_card_4dbc_message
<input>	sips_card_4dbc_input
	sips_popup_message
	sips_submit_message
<input>	sips_submit_button
<input>	sips_customer_return_button
<i>	sips_footer_message

Gestion des erreurs de saisies :

Tag positionné	Nom de classe
	sips_error_message
	sips_error
	sips_last_try_message

1. CAS D'UN PAIEMENT ACCEPTE POUR UNE CB, VISA, MASTERCARD

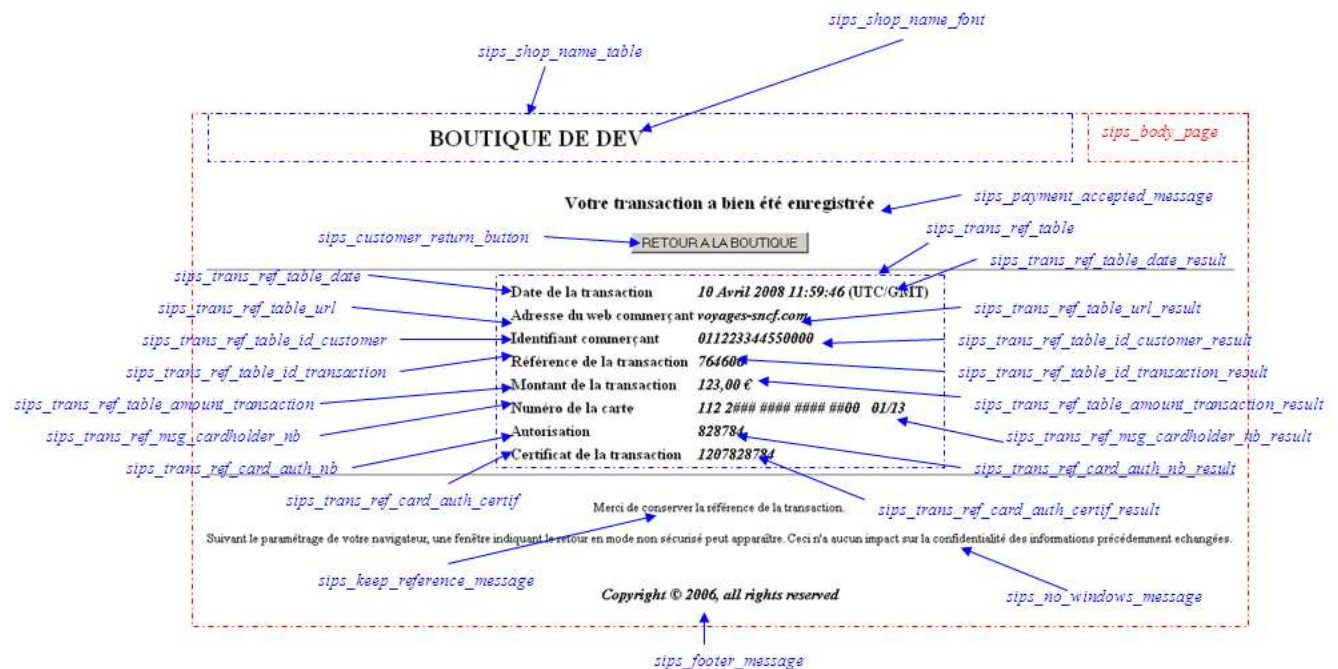


Tableau récapitulatif des classes positionnées :

Tag positionné	Nom de classe
<code><body></code>	<code>sips_body_page</code>
<code><table></code>	<code>sips_shop_name_table</code>
<code><table></code>	<code>sips_trans_ref_table</code>
<code></code>	<code>sips_trans_ref_table_date</code>
<code></code>	<code>sips_trans_ref_table_url</code>
<code></code>	<code>sips_trans_ref_table_id_customer</code>
<code></code>	<code>sips_trans_ref_table_id_transaction</code>
<code></code>	<code>sips_trans_ref_table_amount_transaction</code>
<code></code>	<code>sips_trans_ref_msg_cardholder_nb</code>
<code></code>	<code>sips_trans_ref_card_auth_nb</code>
<code></code>	<code>sips_trans_ref_card_auth_certif</code>
<code><i></code>	<code>sips_trans_ref_table_date_result</code>
<code><i></code>	<code>sips_trans_ref_table_url_result</code>
<code><i></code>	<code>sips_trans_ref_table_id_customer_result</code>
<code><i></code>	<code>sips_trans_ref_table_id_transaction_result</code>
<code><i></code>	<code>sips_trans_ref_table_amount_transaction_result</code>
<code><i></code>	<code>sips_trans_ref_msg_cardholder_nb_result</code>

<i>	sips_trans_ref_card_auth_nb_result
<i>	sips_trans_ref_card_auth_certif_result
	sips_shop_name_font
	sips_keep_reference_message
	sips_no_windows_message
	sips_payment_accepted_message
<input>	sips_customer_return_button
<i>	sips_footer_message

E. CAS D'UN PAIEMENT REJETE

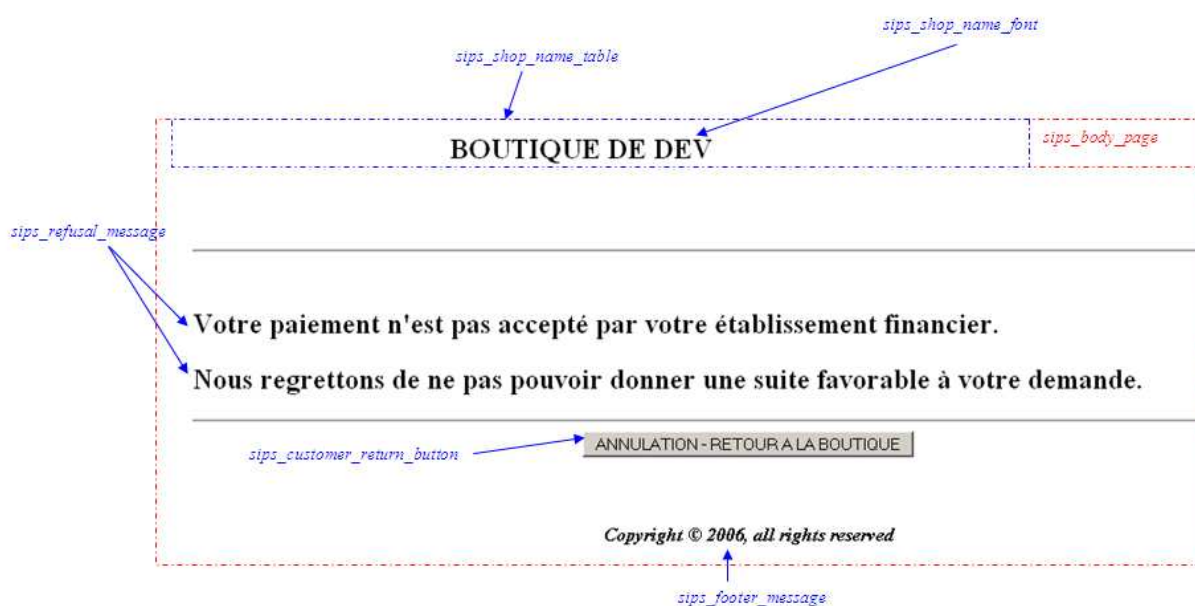


Tableau récapitulatif des classes positionnées :

Tag positionné	Nom de classe
<body>	sips_body_page
<table>	sips_shop_name_table
	sips_shop_name_font
	sips_refusal_message
<input>	sips_customer_return_button
<i>	sips_footer_message

F. CAS D'UN PAIEMENT REJOUÉ



Tableau récapitulatif des classes positionnées :

Tag positionné	Nom de classe
<code><body></code>	<code>sips_body_page</code>
<code><table></code>	<code>sips_shop_name_table</code>
<code><table></code>	<code>sips_trans_ref_table</code>
<code></code>	<code>sips_trans_ref_table_id_customer</code>
<code></code>	<code>sips_trans_ref_table_id_transaction</code>
<code><i></code>	<code>sips_trans_ref_table_id_customer_result</code>
<code><i></code>	<code>sips_trans_ref_table_id_transaction_result</code>
<code></code>	<code>sips_shop_name_font</code>
<code></code>	<code>sips_error_trans</code>
<code></code>	<code>sips_error_trans_message</code>
<code></code>	<code>sips_after_message_on_payment</code>
<code><input></code>	<code>sips_customer_return_button</code>
<code><i></code>	<code>sips_footer_message</code>

1. TRANSACTION EN COURS DE TRAITEMENT POUR CB, VISA, MASTERCARD

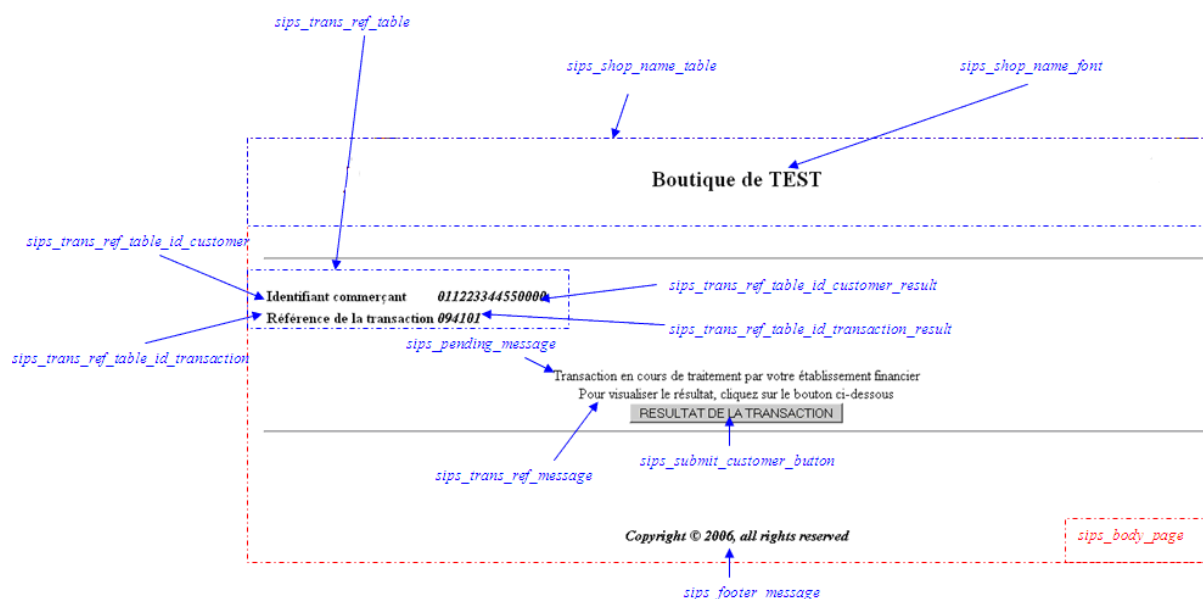


Tableau récapitulatif des classes positionnées :

Tag positionné	Nom de classe
<code><body></code>	<code>sips_body_page</code>
<code><table></code>	<code>sips_shop_name_table</code>
<code><table></code>	<code>sips_trans_ref_table</code>
<code></code>	<code>sips_trans_ref_table_id_customer</code>
<code></code>	<code>sips_trans_ref_table_id_transaction</code>
<code><i></code>	<code>sips_trans_ref_table_id_customer_result</code>
<code><i></code>	<code>sips_trans_ref_table_id_transaction_result</code>
<code></code>	<code>sips_shop_name_font</code>
<code></code>	<code>sips_pending_message</code>
<code></code>	<code>sips_trans_ref_message</code>
<code><input></code>	<code>sips_submit_customer_button</code>
<code><i></code>	<code>sips_footer_message</code>

H. CAS D'UNE REDIRECTION 3D-SECURE



Tableau récapitulatif des classes positionnées :

Tag positionné	Nom de classe
<code><body></code>	<code>sips_body_page</code>
<code><table></code>	<code>sips_shop_name_table</code>
<code></code>	<code>sips_logo</code>
<code><table></code>	<code>sips_trans_ref_table</code>
<code></code>	<code>sips_trans_ref_table_id_customer</code>
<code></code>	<code>sips_trans_ref_table_id_transaction</code>
<code></code>	<code>sips_trans_ref_table_amount_transaction</code>
<code><i></code>	<code>sips_trans_ref_table_id_customer_result</code>
<code><i></code>	<code>sips_trans_ref_table_id_transaction_result</code>
<code><i></code>	<code>sips_trans_ref_table_amount_transaction_result</code>
<code></code>	<code>sips_shop_name_font</code>
<code></code>	<code>sips_threeD_redirect_info_message</code>
<code></code>	<code>sips_threeD_noredirect_action_message</code>
<code></code>	<code>sips_redirect_message</code>
<code></code>	<code>sips_buffer_before</code>
<code></code>	<code>sips_buffer_after</code>
<code></code>	<code>sips_threeD_after_message</code>
<code><i></code>	<code>sips_footer_message</code>