

FlexCheckout

ingenico
ePayments

Table des matières

1. Introduction

2. Flux de données typique

2.1 DirectLink

2.2 Alias Manager

3. Intégration de FlexCheckout sous forme de tokenisation

3.1 Champs de saisie

3.1.1 Rechercher les ID de commande en double

3.2 Signature SHA pour saisie

3.3 Champs du retour d'information

3.4 Signature SHA pour retour d'information

4. Personnalisation

4.1 Personnalisation de base

4.1.1 Modèle Ingenico ePayments par défaut

4.1.2 Utilisation des feuilles de style

4.2 Personnalisation avancée

4.2.1 Utilisation des marques substitutives

4.2.2 Personnalisation du texte avec des fichiers XML

4.3 Gestionnaire de fichiers modèles

4.3.1 Télécharger des fichiers modèles

4.3.2 Contrôler et gérer les fichiers téléchargés

4.3.3 Intégration

5. Marques substitutives

1. Introduction

Le paiement en une page avec FlexCheckout vous permet de demander à vos clients de saisir les informations de leur carte en toute sécurité sur le système PSP, de tokeniser ces données et (facultativement) de réutiliser le token/alias dans le cadre d'une future transaction.

Avantages

- Vous contrôlez intégralement l'aspect de votre page de paiement grâce au mécanisme de modèle flexible, et si vous intégrez la FlexCheckout à un Iframe, vous contrôlez également la séquence de paiement.
- Vous bénéficiez d'une page de paiement sécurisée et conforme à la norme PCI sans devoir stocker vous-même les informations de la carte ; toutes les données sensibles sont saisies directement sur la page hébergée.
- Intégration sans faille de différents scénarios de paiement, y compris le paiement en une page.
- Vous pouvez proposer des ventes croisées et référencées sur la page finale de paiement.
- Les principales cartes de crédit sont acceptées (VISA, MasterCard, American Express, Diners).

Options requises

Vous devez activer au moins une des options suivantes pour pouvoir utiliser FlexCheckout :

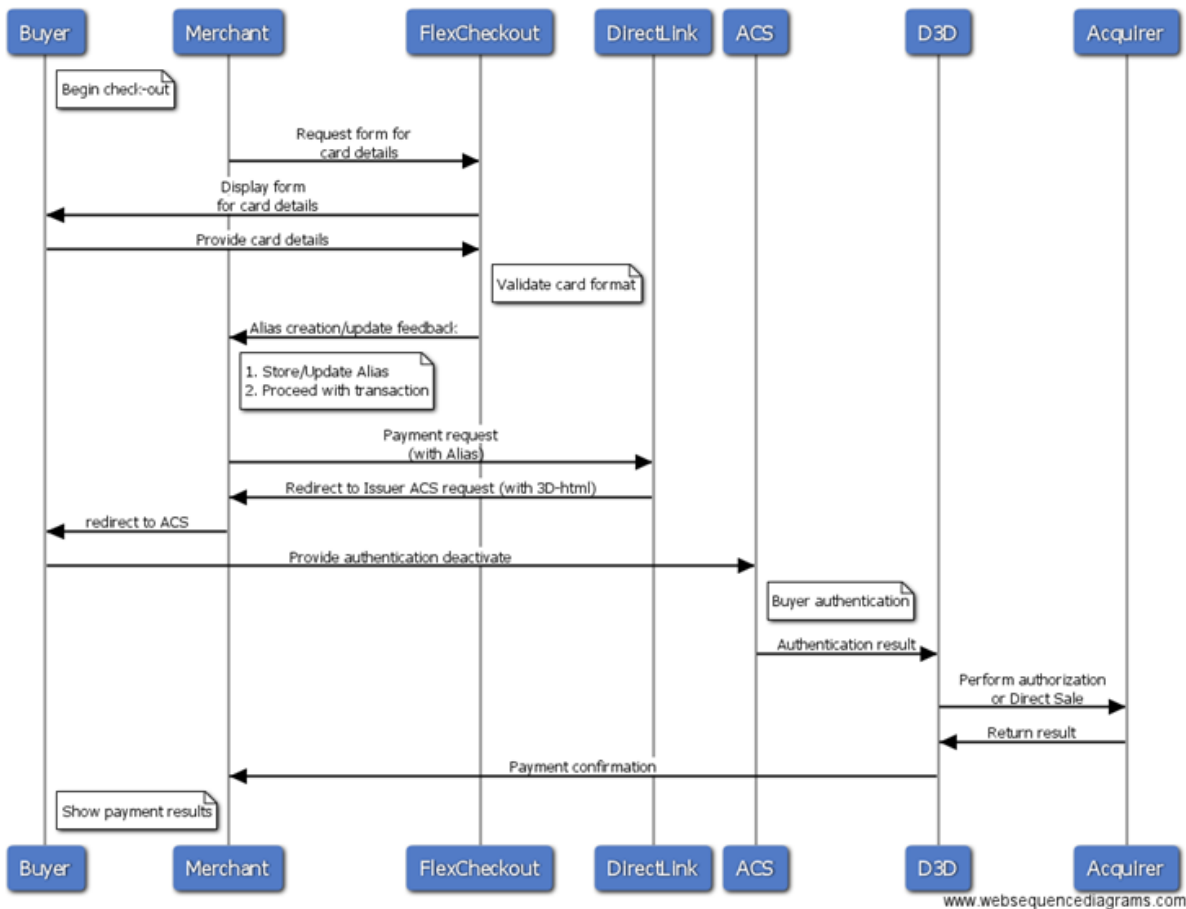
- Paiement en une page (ID option : OPC)
- Alias Manager (ID option : REC1, REC2, RECX)

Normalement, vous devez également activer l'option DirectLink (nouveaux paiements), à moins qu'elle ne le soit par défaut dans votre abonnement.

2. Flux de données typique

Vous pouvez rediriger FlexCheckout vers (ou inclure dans) un iframe qui permettra au titulaire de la carte d'entrer les données de la carte dans notre système. Les données de la carte, ainsi que le CVC, sont stockées en toute sécurité, mais seulement pendant un certain temps. Pour que les données de la carte soient stockées en permanence, vous pouvez sélectionner l'alias « Persistant » (Persistant). Quoi qu'il en soit, conformément au Conseil des normes de sécurité PCI, les CVC ne peuvent être stockés que pendant deux heures maximum.

L'option DirectLink vous permet également d'envoyer la commande réelle avec l'alias généré, sans envoyer les données de la carte.



Remarque: Aucune opération n'est réalisée sur la carte lors de la première étape. Notre système effectue simplement une validation de format de base, mais cela ne garantit pas que la carte est encore valide ou que le solde est suffisant.

2.1 DirectLink

Avec l'alias généré par la FlexCheckout, vous pouvez soumettre une transaction DirectLink en faisant appel à notre implémentation DirectLink standard. Pour plus d'informations sur la procédure d'implémentation, aller à [DirectLink](#).

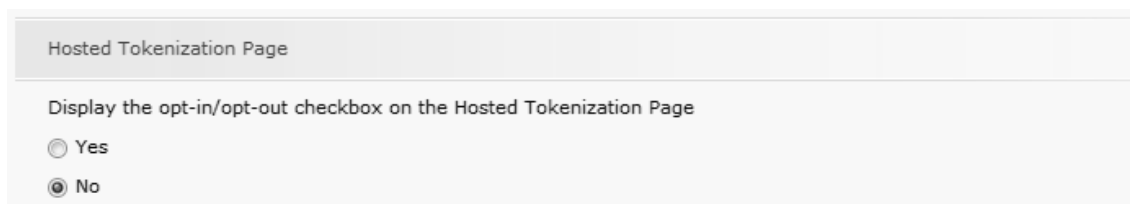
Ce mécanisme est également compatible avec [DirectLink et 3-D Secure](#). Pour plus d'informations sur l'utilisation d'Alias, veuillez vous reporter à [Alias Manager](#).

2.2 Alias Manager

Si vous activez l'option Alias Manager, vous pouvez ajouter une case à cocher sur la FlexCheckout grâce à laquelle le client peut décider si ses

informations financières sont enregistrées définitivement ou pas.

Pour ce faire, vous devez activer un paramètre d'affichage de la case sur la page de configuration de l'Alias Manager de votre compte Ingenico ePayments (le paramètre par défaut est « Non »).



Hosted Tokenization Page

Display the opt-in/opt-out checkbox on the Hosted Tokenization Page

Yes

No

> Nous vous recommandons d'utiliser cette configuration dans un environnement de paiement en une page.

Si vous avez intégré un processus de paiement à différentes étapes, vous pouvez aussi utiliser [le paramètre "StorePermanently"](#) dans votre intégration pour déterminer quand la case à cocher doit être affichée.

3. Intégration de FlexCheckout sous forme de tokénisation

Vous pouvez utiliser la FlexCheckout de deux façons :

1. En redirigeant votre client vers FlexCheckout ;
2. En encapsulant FlexCheckout dans un iframe de sorte que vos clients restent sur votre page pour une expérience de paiement en une page complète.

Avertissement : Nous vous déconseillons d'intégrer FlexCheckout en tant que solution In-App pure. Certaines fonctions risquent en effet de ne pas fonctionner sur des smartphones de milieu de gamme, car elles nécessitent une solution de site Web mobile et requièrent une utilisation optimale des navigateurs.

Si le client est redirigé vers FlexCheckout, il devra indiquer les données de sa carte et les envoyer pour la tokénisation sur FlexCheckout. De cette façon, les données de la carte ne passent jamais par votre propre serveur. Adresses URL permettant d'accéder à FlexCheckout :

- Test : <https://ogone.test.v-psp.com/Tokenization/HostedPage>
- Prod: <https://secure.ogone.com/Tokenization/HostedPage>

Par défaut, la FlexCheckout fonctionne en encodage de caractères UTF-8. Si vous travaillez en ISO, assurez-vous d'adapter l' « Encodage des caractères » dans votre Back office via le menu « Configuration » > « Information technique » > « Paramètres globaux de sécurité » > « Méthode de hachage ».

En cas d'utilisation de l'encodage ISO pour calculer le SHA, les commerçants devront définir ce paramètre dans le Back Office. Ce paramètre valide le SHA entrant et le SHA sortant. Il sera également utilisé pour définir le schéma d'encodage des URL pour les paramètres retournés.

3.1 Champs de saisie

Champ	Description	Format	Obligatoire
ACCOUNT.PSPID	Identification du commerçant	AN, 30	Oui
ALIAS.ALIASID	Alias du client	AN, 50	Non
ALIAS.ORDERID	Identification de la commande	AN, 40	Non
* ALIAS.STOREPERMANENTLY	Indique si vous voulez enregistrer un alias provisoirement ou indéfiniment. Les différentes valeurs possibles sont les suivantes : <ul style="list-style-type: none"> • « N » : l'alias sera supprimé au bout de 2 heures • « Y » : l'alias sera enregistré indéfiniment pour être utilisé ultérieurement 	Y / N	Non
CARD.BIC	Bank Identification Code. Utilisé uniquement pour les prélèvements.	AN, 8	
CARD.BIN	Méthodes de paiement par carte de crédit	AN, 6	
CARD.BRAND	Indique le type de formulaire qui doit être affiché. > Obligatoire si aucune méthode de paiement n'est	AN, 25	Oui/Non

Champ	Description	Format	Obligatoire
	fournie.		
CARD.PAYMENTMETHOD	Carte de crédit ou tout mode de prélèvement automatique pris en charge : indique le type de formulaire qui doit être affiché. > Obligatoire si aucune marque n'est fournie.	AN, 26	Oui/Non
LAYOUT.LANGUAGE	Langue utilisée sur la page ("en_EN", "nl_BE", "fr_FR" etc.).	AN, 5	Non
LAYOUT.TEMPLATENAME	Paramètre d'entrée de modèle. Les commerçants ont besoin du nom du modèle et l'extension du fichier, par exemple « new_user.html ».	AN, 255	Non
PARAMETERS.ACCEPTURL	URL de redirection en cas de réussite.	AN, 255	Oui
PARAMETERS.EXCEPTIONURL	URL de redirection en cas d'erreur.	AN, 255	Oui
PARAMETERS.EXCLUDEDPAYMENTMETHODS	Liste des méthodes de paiement et/ou des cartes de crédit qui ne doivent PAS être affichées. Séparé par « ; » (point virgule).	AN, 50	Non
PARAMETERS.PARAMPLUS	Champ Pass-Through : Autres paramètres envoyés par le commerçant à extraire de l'édition.	AN, 1000	Non
SHASIGNATURE.SHASIGN	Chaîne hachée à l'aide de la fonction SHA (Secure Hash Algorithm).	AN, 128	Oui

*** ALIAS.STOREPERMANENTLY : Remarques importantes**

- ALIAS.STOREPERMANENTLY doit être utilisé **uniquement** avec Alias Manager. Par défaut, l'alias sera enregistré indéfiniment.
- Si vous utilisez une intégration de paiement en une page, nous vous recommandons vivement d'activer l'option « opt-in/out checkbox » (Case à cocher Engager/désengager) dans la configuration d'Alias Manager, pour toujours afficher la case à cocher sur la FlexCheckout.
- Si vous avez activé l'option « opt-in/out checkbox » (Case à cocher Engager/désengager) dans la configuration d'Alias Manager, le paramètre ALIAS.STOREPERMANENTLY est ignoré.

3.1.1 Rechercher les ID de commande en double

Afin d'empêcher les pirates informatiques de remplacer les informations relative à une carte et liées à un token spécifique (en capturant le lien permettant de déclencher la requête et en remplaçant les informations relatives à une carte par les informations d'une carte originale), nous vérifions l'ID de commande (ALIAS.ORDERID) que vous envoyez dans la requête.

Si nous détectons que l'ID de la commande a déjà été utilisé pour créer un token, la mise à jour de l'alias sera refusée.

Important

Bien entendu, la vérification ne fonctionne que si l'**ID de la commande (ALIAS.ORDERID)** est envoyé. Cependant, l'ID de la commande seul ne suffit pas ; l'**ID de l'alias (ALIAS.ALIASID)** est également nécessaire. Si l'ID de l'alias n'est pas envoyé, la vérification ne sera pas réalisée et un nouvel alias sera créé.

En cas de détection d'un ID de commande en double, et donc de blocage de la requête, un message d'erreur générique s'affiche à l'utilisateur :

"An error has occurred while processing your request. Please try again later or contact your merchant." (Une erreur s'est produite lors du traitement de votre requête. Réessayez plus tard ou contactez votre commerçant.)

Les erreurs générées par un ID de commande en double peuvent être détectées lorsque le mode de débogage des rapports d'erreurs est activé (veuillez contacter le notre service clientèle).

3.2 Signature SHA pour saisie

Pour vérifier l'intégrité des données, toutes les demandes doivent être accompagnées d'une signature SHA, in [tout comme pour les transactions e-Commerce](#).

Notre système utilisera l'algorithme SHA défini dans l'onglet « Paramètres généraux de sécurité » de votre page d'information technique. Vous pouvez encore revenir à l'algorithme SHA-1 ou SHA-256.

Exemple

Champs (par ordre alphabétique) :

Parameters.AcceptUrl: https://www.myshop.com/ok.html

Parameters.ExceptionUrl: https://www.myshop.com/nok.html

Account.PspId: test1

Card.Brand:VISA

Phrase secrète (définie dans la page Information technique) : Mysecretsig1875!?

Chaîne à hacher :

ACCOUNT.PSPID=test1Mysecretsig1875!?CARD.BRAND=VISAMysecretsig1875!?PARAMETERS.ACCEPTURL=https://www.myshop.com/ok.htmlMysecretsig1875!?PARAMETERS.EXCEPTIONURL=https://www.myshop.com/nok.htmlMysecretsig1875!?

Condensé obtenu (SHA-512) :

563DC909F70BA5DDD470D69C1B390E7D1C1C47705AC5801B27038446D7033B5787728EA
754EF72E7FA2436FC5962E34E20DF64E7F9139893A33653F118816818

3.3 Champs du retour d'information

Les champs suivants peuvent vous être renvoyés pour connaître le statut de la création/mise à jour de l'alias. Afin de les inclure dans le retour d'information, ils devront être configurés en conséquence dans les paramètres dynamiques de retour d'information (compte Ingenico ePayments : Configuration > Information technique > Retour d'information sur la transaction > Alias gateway et Tokenization: Paramètres dynamiques).

Le SHASIGN n'est pas facultatif et doit donc toujours être renvoyé.

Champ	Description	Longueur max.
ALIAS.ALIASID	Alias envoyé par le commerçant ou généré par PSP (suivant le format GUID de 32 caractères). Exemple : 34F5302C-85D7-4F35-BDF5-103CCEC2FB61	50
ALIAS.NCERROR	Code d'erreur	50
ALIAS.NCERRORCARDNO	Code d'erreur pour CARDNO	50
ALIAS.NCERRORCN	Code d'erreur pour CN	50
ALIAS.NCERRORCVC	Code d'erreur pour CVC	50
ALIAS.NCERRORED	Code d'erreur pour ED	50
ALIAS.ORDERID	Identifiant unique de la commande (envoyé par le commerçant ou généré par le système).	40
ALIAS.STATUS	Résultat de la création d'alias: <ul style="list-style-type: none"> • 0=OK • 1=NOK • 2=Alias mis à jour • 3=Annulé par l'utilisateur 	1
ALIAS.STOREPERMANENTLY	Indique si l'alias doit être enregistré provisoirement (N) ou indéfiniment (Y).	1 (Y/N)
CARD.BIC	Bank Identification Code. Utilisé uniquement pour les prélèvements.	8
CARD.BIN	Méthodes de paiement par carte de crédit	6
CARD.BRAND	Nom du mode de paiement	25
CARD.CARDHOLDERNAME	Nom du titulaire de la carte	50
CARD.CARDNUMBER	Carte : les informations sensibles sont remplacées par des « X ». Exemple: XXXXXXXXXXXXX1111 Remarque : En cas d'erreur, la carte sera aussi masquée.	35
CARD.CVC	Cryptogramme visuel ; les données sensibles sont remplacées par des « X ». Exemple: XXX	6
CARD.EXPIRYDATE	Date d'expiration, par exemple 0220 (février 2020)	4
PARAMPLUS	Champ Pass-through : données fournies lors de la saisie.	/
SHASIGN	Signature SHA pour retour d'information	128

Les paramètres sélectionnés seront ajoutés à l'URL renvoyée définie dans votre requête (PARAMETERS.ACCEPTURL ou PARAMETERS.EXCEPTIONURL), afin de transmettre le retour d'information de l'opération.

3.4 Signature SHA pour retour d'information

Notre système renverra une signature SHA-OUT, tout [comme pour les transactions e-Commerce](#), pour les paramètres suivants :

ALIAS.ALIASID
ALIAS.NCERROR
ALIAS.NCERRORCARDNO
ALIAS.NCERRORCN
ALIAS.NCERRORCVC
ALIAS.NCERRORED
ALIAS.ORDERID
ALIAS.STATUS
ALIAS.STOREPERMANENTLY
CARD.BIC
CARD.BIN
CARD.BRAND
CARD.CARDHOLDERNAME
CARD.CARDNUMBER
CARD.CVC
CARD.EXPIRYDATE

4. Personnalisation

Ce chapitre présente deux types de personnalisation : de base et avancée. La personnalisation de base vous permettra d'apprendre à appliquer votre propre fichier CSS et à personnaliser les couleurs de texte, d'arrière-plan et de wrapper, ainsi que d'autres paramètres de ce genre. Voici une page de paiement par défaut qui utilise les fichiers CSS par défaut :

The image shows a payment form with the following elements:

- Card number:** A single-line text input field.
- Cardholder's name:** A single-line text input field.
- Expiry date:** Two dropdown menus for month (MM) and year (YY), separated by a slash (/).
- Card verification code:** A single-line text input field with an information icon (i) to its right.
- Buttons:** A green "Submit" button and a grey "Cancel" button.

Grâce à la personnalisation avancée, vous en saurez plus sur les modèles principaux et partiels, sur l'utilisation des marques substitutives et sur la personnalisation du texte avec les fichiers XML. Selon vos exigences, les modèles partiels sont créés et personnalisés séparément.

4.1 Personnalisation de base

La personnalisation de base implique la configuration d'une feuille de style par défaut selon le modèle Ingenico ePayments par défaut.

4.1.1 Modèle Ingenico ePayments par défaut

Le fichier CSS file peut être personnalisé avec une feuille de style Ingenico ePayments par défaut ou avec un fichier CSS chargé dans le [gestionnaire de fichiers modèles](#).

Pour en savoir plus sur la personnalisation de votre propre fichier CSS, voir Utilisation des [feuilles de style](#).

The image shows a configuration window titled "Style sheet". It contains:

- An information icon (i) on the left.
- A label "Stylesheet filename" next to a dropdown menu.
- The dropdown menu is open, showing two options: "Default Ogone template" (highlighted in blue) and "layer_styling_v5.css".
- A "SAVE" button is located at the bottom left of the window.

4.1.2 Utilisation des feuilles de style

Vous pouvez adapter la mise en page de FlexCheckout en appliquant un fichier CSS. Vous devrez d'abord charger votre fichier CSS dans le Gestionnaire de fichiers modèles et l'associer à votre compte Ingenico ePayments. Le fichier CSS se compose d'un formulaire/conteneur de données et d'éléments (par exemple, date d'expiration, CVC, zone d'informations, etc.) et doit être défini comme fichier CSS par défaut. Vous pouvez associer chaque élément qui définit la mise en forme de chaque champ au formulaire/conteneur de données.

En fonction de vos besoins, vous pouvez modifier la règle de mise en forme de chaque élément. Cliquez [ici](#) pour accéder à une liste d'éléments CSS.

Voici des exemples de formulaire/conteneur de données et d'éléments :

```
/*-----FORM\DATA CONTAINER-----*/

#payment-container {width:100%; max-width:768px !important; margin:0 auto;background:#fff;}

#payment-form-container {width:280px !important; padding:0 10px; margin:0 auto;
background:#fff;overflow:hidden;}

#payment-data-container {width:260px !important; padding:20px 10px 5px 10px; margin:0 auto; color:#000;
background:#fff;overflow:hidden;}

#payment-data-container .payment-label { color:#000; font-size:12px; padding-bottom:3px;}

#payment-data-container .payment-input input { width:248px !important; height:35px; color:#000;
font-size:12px; margin-bottom:10px; padding:0 5px; -webkit-border-radius: 3px; -moz-border-radius: 3px;
border-radius: 3px; border:1px solid #000;}

#payment-data-container .payment-input select { width:109px; height:35px; color:#000; font-size:12px; margin-
bottom:10px; padding:7px 5px; -webkit-border-radius: 3px;-moz-border-radius: 3px; border-radius: 3px;
border:1px solid #000;}

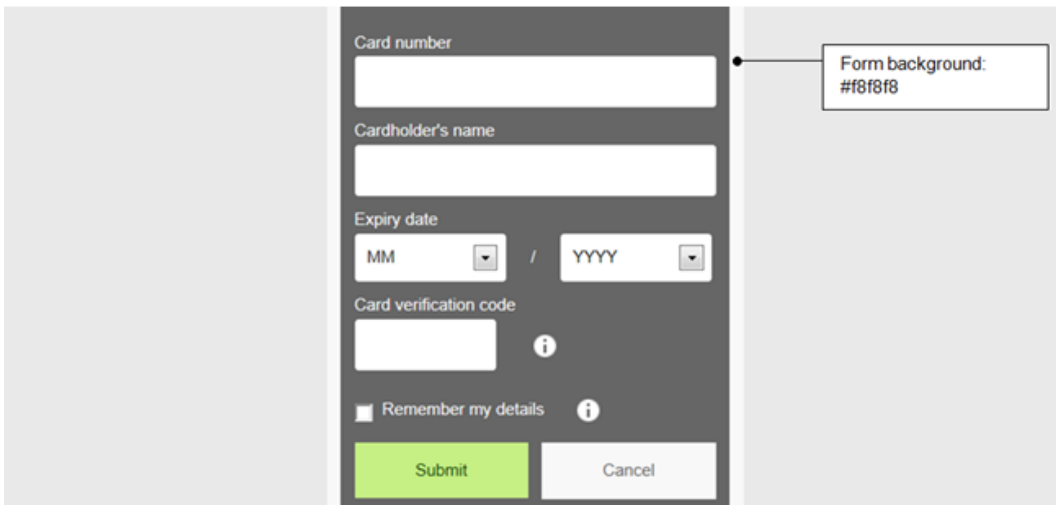
#payment-cardnumber-label-container,#payment-accountowner-label-container{ overflow: hidden;}

/*-----FORM\DATA CONTAINER-----*/
```

Pour vous aider à démarrer, nous avons créé un exemple de fichier CSS que vous pouvez télécharger à partir [d'ici](#) (fichier zip). Vous pouvez commencer à créer votre propre CSS avec ce fichier.

Vous pouvez modifier l'interface utilisateur par défaut (comme indiqué ci-dessous) en appliquant votre propre CSS.

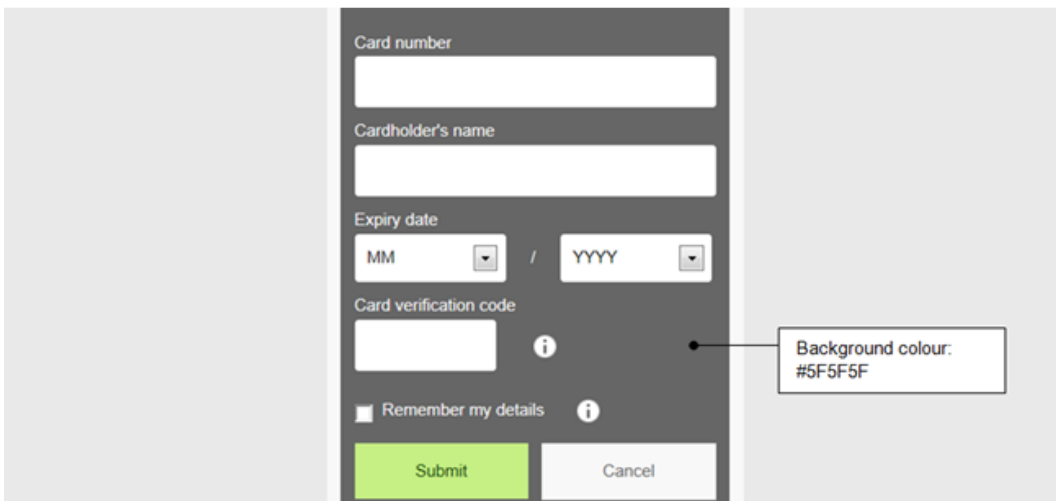
Couleur de fond du formulaire



CSS affecté:

```
.payment-form-container{ Background:#f8f8f8; }
```

Couleur de la jaquette du formulaire



CSS affecté:

FlexCheckout

```
#payment-data-container{ Background:#5F5F5F; }
```

Couleur de texte

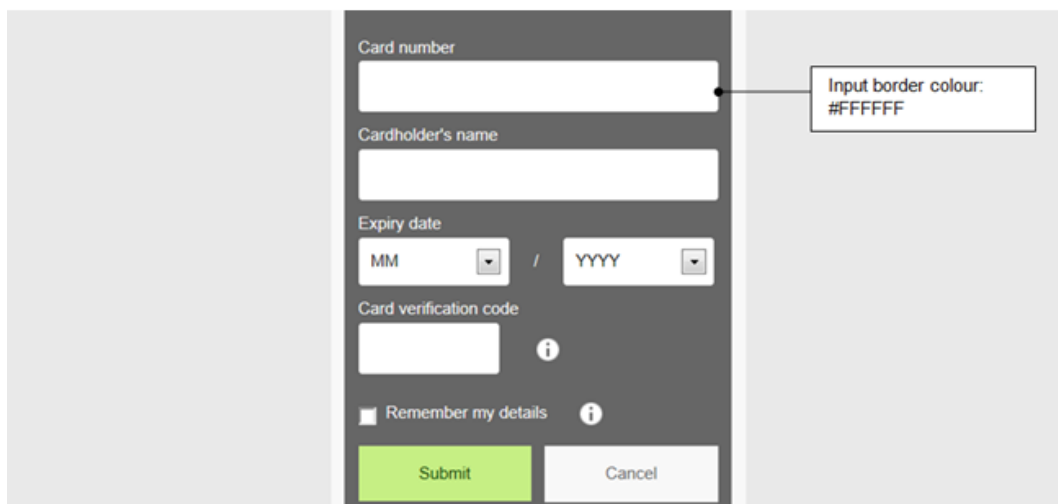
The image shows a payment form with a dark grey background. The form contains the following fields and elements:

- Card number:** A text input field with a label above it. A callout box points to the label with the text "Text colour: #FFFFFF".
- Cardholder's name:** A text input field with a label above it.
- Expiry date:** Two dropdown menus for month (MM) and year (YYYY), separated by a slash.
- Card verification code:** A text input field with a label above it and an information icon to its right.
- Remember my details:** A checkbox with a label and an information icon to its right.
- Authorization text:** A text block containing the following text: "As your customer, I authorize you and/or your payment processor to store my payment data in a secure way so that I will not have to enter them again when I will perform future payments on your website. If not, my payment data will be deleted as soon as my payment has been processed." A callout box points to this text with the text "Text colour: #6E6E6E".
- Submit and Cancel buttons:** Two buttons at the bottom of the form.

CSS affecté:

```
#payment-data-container .payment-label{color:#FFFFFF;} #payment-data-container{color:#6E6E6E;}
```

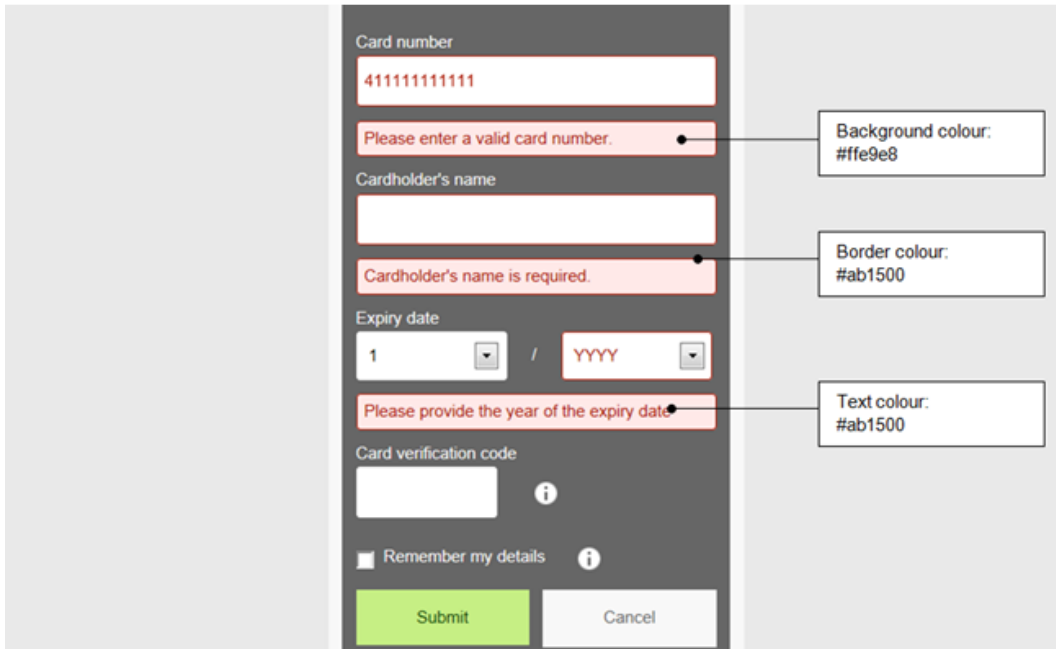
Couleur de la bordure de saisie



CSS affecté:

```
#payment-data-container .payment-input input{border:1px solid #FFFFFF;}
```

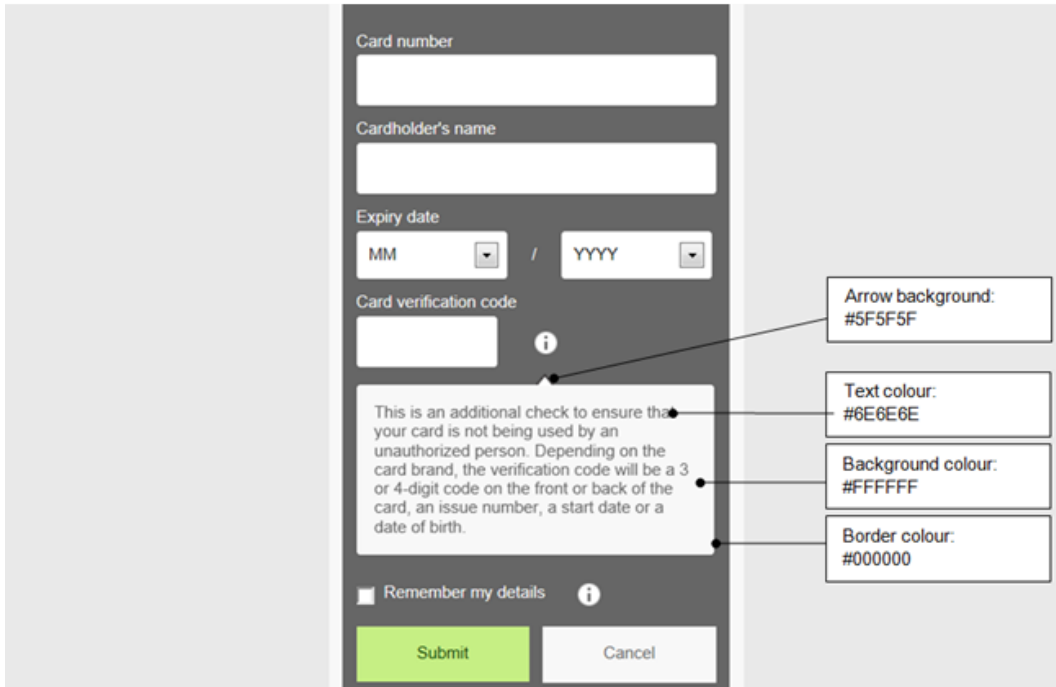
Fenêtre de message de validation



CSS affecté:

```
.payment-error { color:#ab1500; background:#ffe9e8; border:1px solid #ab1500;} #payment-data-container .payment-input-error input { color:#ab1500; border:1px solid #ab1500;} #payment-data-container .payment-input-error select { color:#ab1500; border:1px solid #ab1500;}
```

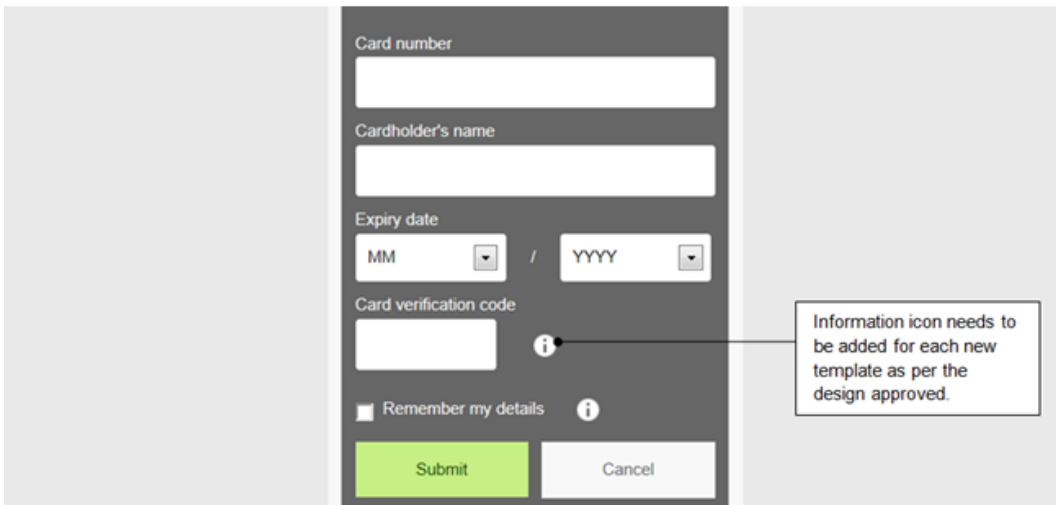
Fenêtre d'aide/d'information



CSS affecté:

```
#payment-cvc-info-container span.arowlmg {background: #5F5F5F url(arrow-top.png) 0 0 no-repeat;} #payment-cvc-info-container p.cvc-info-txt{color: #6E6E6E;}< #payment-cvc-info-container div.help-box {border: 1px solid #000000;background: #FFFFFF;}
```

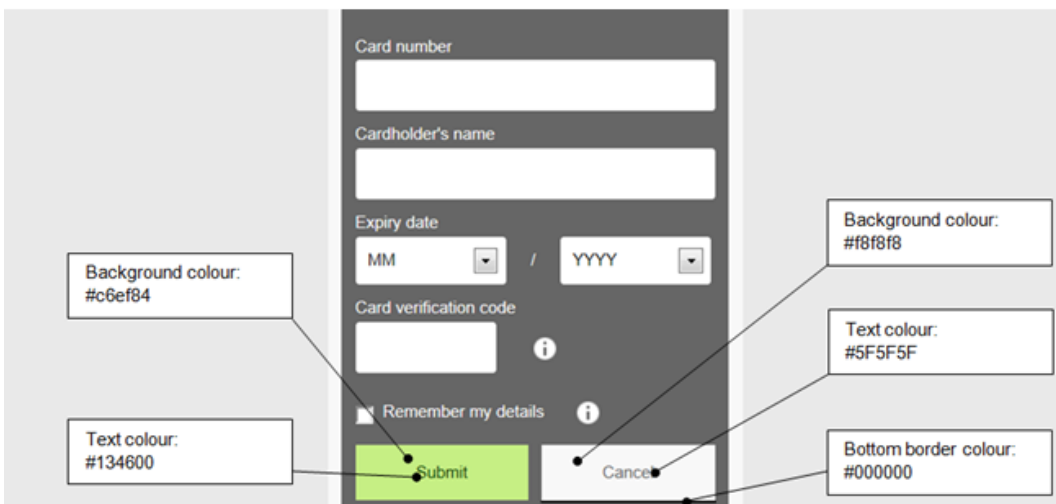
Icône d'information



CSS affecté:

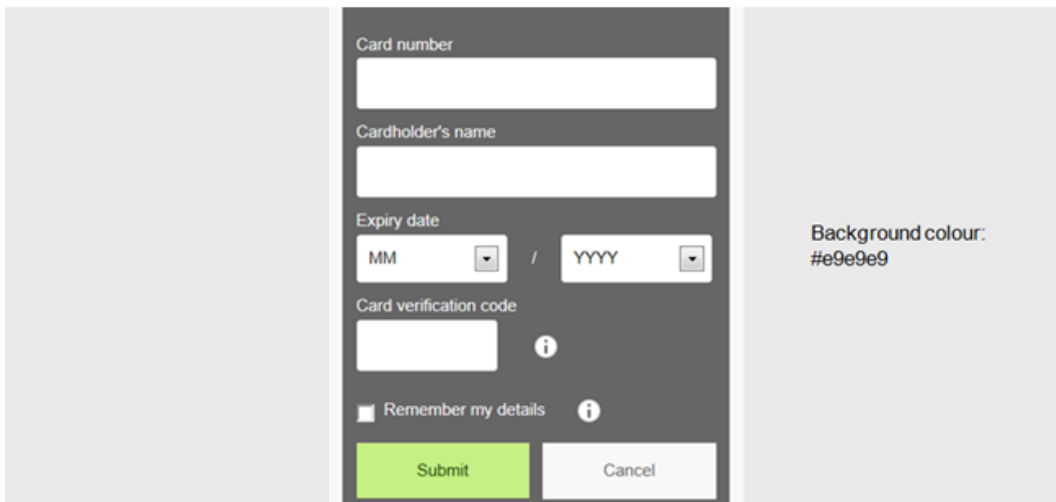
```
.payment-method-info {background:url(payment_info.png) 4px 1px no-repeat;}
```

Bouton Envoyer/Annuler



```
CSS affecté: #payment-data-container #payment-submit {background:#c6ef84;color: #134600;} #payment-cancel-container input {background:#f8f8f8;border-color:#000;}
```

Couleur de fond



CSS affecté:

```
#payment-container {background:# e9e9e9}
```

Exemple de CSS

```
#payment-container {background:# e9e9e9} .payment-form-container{ Background:#f8f8f8; }  
#payment-data-container{ Background:#666; }  
  
#payment-data-container{color:fff;}  
  
#payment-data-container .payment-label{color:fff;}  
  
#payment-data-container .payment-input input{border:1px solid #fff;}  
  
.payment-error { color:#ab1500; background:#ffe9e8; border:1px solid #ab1500;}  
  
#payment-data-container .payment-input-error input { color:#ab1500; border:1px solid #ab1500;}  
  
#payment-data-container .payment-input-error select { color:#ab1500; border:1px solid #ab1500;}  
  
#payment-cvc-info-container div.help-box {border: 1px solid #000;background: fff;}  
  
#payment-cvc-info-container span.arowlmg {background: #fff url(arrow-top.png) 0 0 no-repeat;}  
  
#payment-cvc-info-container p.cvc-info-txt{color: #000;}  
  
.payment-method-info {background:url(payment_info.png) 4px 1px no-repeat;}
```

```
#payment-data-container #payment-submit {background:#c6ef84;color: #134600;}

#payment-cancel-container input {background:#f8f8f8;border-color:#000;}
```

4.2 Personnalisation avancée

Pour plus de flexibilité, vous pouvez désormais créer et personnaliser un modèle principal et des modèles partiels. Selon vos exigences, les modèles partiels sont créés et personnalisés séparément. Vous pouvez également personnaliser du texte avec des fichiers XML.

Vous pouvez choisir d'appliquer une balise `$$$PAYMENT ZONE$$$` qui appelle un modèle simple ou de personnaliser votre propre modèle partiel en utilisant des [marques substitutives](#).

4.2.1 Utilisation des marques substitutives

Vous pouvez personnaliser leur propre modèle principal (**master.html**) et des modèles partiels peuvent insérer directement des marques substitutives dans le modèle principal **master.html**.

Vous pouvez également définir l'aspect de la page de paiement en utilisant un fichier CSS personnalisé (**newuser.css**), qui doit être associé au modèle principal master.html. Si vous choisissez de personnaliser votre fichier CSS, ce dernier doit être chargé dans le Gestionnaire de fichiers modèles et défini comme fichier par défaut.

Ces dernières sont classées en trois catégories : marques substitutives principales, d'éléments et groupées.

Marques substitutives principales

Voici une liste de marques substitutives principales:

- `$$$PAYMENT ZONE$$$`: affiche une marque ou un mode de paiement demandés via un modèle personnalisé (avec un modèle partiel disponible) ou le modèle par défaut.
- `$$$TP RESOURCES URL$$$`: met l'URL à disposition du répertoire où se trouvent les fichiers du service d'hébergement de fichiers.

Vous pouvez également utiliser des scripts personnalisés qui devront être associés à un fichier JavaScript personnalisé. Ce dernier peut alors être référencé après le chargement des scripts personnalisés dans le Gestionnaire de fichiers modèles. Par exemple, le fichier JavaScript référencé dans `<script type="text/javascript" src="$$$TP RESOURCES URL$$$/existinguser.js"></script>` est **existinguser.js**.

Pour utiliser des bibliothèques JavaScript dans les modèles (jQuery, par exemple), vous devez charger les fichiers JavaScript dans le Gestionnaire de fichiers modèles. Après les avoir chargés, vous devrez utiliser `$$$TP RESOURCE URL$$$` pour référencer les scripts. Cette étape doit être répétée pour tous les fichiers référencés dans le modèle (CSS, images, polices, etc.). Elle garantit que le script exécuté est bien le fichier JavaScript dédié.

Note: toutes les données ne pouvant être envoyées vers/extraites d'un hôte tiers non conforme aux normes PCI, l'utilisation de JavaScript influera sur l'admissibilité des commerçants au questionnaire d'auto-évaluation A de PCI .

- `$$$DISABLE MASKING$$$`: désactive le masquage d'une zone d'entrée.
- `$$$SUBMIT$$$`: affiche un bouton Envoyer pour le formulaire. Exemple :

```
<div id="payment-submit-container">
  <input id="payment-submit" type="submit" value="Submit"/>
</div>
```

- `$$$CANCEL$$$`: affiche un bouton Annuler pour le formulaire. Exemple :

```
<div id="payment-cancel-container">
<input id="payment-cancel" name="cancel" type="submit" value="Cancel" class="cancel" />
</div>
```

Un modèle principal contenant \$\$\$PAYMENT ZONE\$\$\$ n'aura pas besoin d'inclure les marques substitutives Envoyer et Annuler. Elles devront néanmoins être incluses en cas de création d'un modèle partial par mode de paiement ou d'un modèle principal sans la marque substitutive \$\$\$PAYMENT ZONE\$\$\$.

Marques substitutives d'éléments

Le contrôle réel <element> s'affiche dans un conteneur d'éléments. <div id="payment-<input_name>-<element>-container" class="payment-<element>">. Pour nommer facilement les fichiers CSS, la même convention d'affectation de noms s'applique à ID et classe. Les marques substitutives d'éléments affichent les contrôles html pour différents éléments d'entrée et doivent respecter ce format :

- \$\$\$<input_name><element>\$\$\$: comme contrôle d'entrée de numéro de carte.
- \$\$\$CARD NUMBER.INPUT\$\$\$ - CARD NUMBER: comme <input_name> où INPUT est l'<element> html.
- \$\$\$<input_name>LABEL\$\$\$: affiche une étiquette pour l'élément <input_name>.

Par exemple, la marque substitutive \$\$\$CARD NUMBER LABEL\$\$\$ affiche :

```
<div id="payment-cardnumber-label-container" class="payment-label">
<label for="payment-cardnumber" id="lbl_CreditCardInputModel_CardNumber" title="CardNumber">Card number</label> </div>
```

Pour toutes les étiquettes, div s'affichera pour faire office de conteneur:

```
<div id="payment-<input_name>-label-container" class="payment-label">
```

La classe d'un conteneur d'étiquettes sera toujours « payment-label » pour que les étiquettes puissent être nommées ensemble.

Une étiquette pour un contrôle d'étiquette s'affichera toujours :

```
<label for="payment-<input_name>" id="lbl_CreditCardInputModel_CardNumber"
title="CardNumber">Card number</label>
```

Pour nommer un contrôle d'étiquette, appliquez le fichier CSS à l'ID de conteneur, non pas à l'ID de contrôle.

\$\$\$<input_name> INPUT\$\$\$ affiche le contrôle d'entrée pour <input_name> :

```
<div id="payment-<input_name>-input-container" class="payment-input">
<input Id="payment-<input_name>" autocapitalize="off" autocomplete="off"
autocorrect="off" class="inp-txt" id="txt_CreditCardInputModel_CardNumber"
maxlength="40" name="CreditCardInputModel.CardNumber" pattern="[X0-9]*"
spellcheck="False" type="tel" value=""></input>
</div>
```

Pour nommer un contrôle d'entrée, appliquez le fichier CSS à l'ID de conteneur, non pas à l'ID de contrôle.

\$\$\$<input_name> INFO\$\$\$ affiche un lien vers une description du champ <input_name> :

```
<div id="payment-<input_name>-info-container" class="payment-info">
  <span id="payment-<input_name>-info"></span>
</div>
```

Liez toutes les informations de ce champ dans la langue sélectionnée.

\$\$\$<input_name> ERROR\$\$\$ affiche un conteneur de messages d'erreur issus de <input_name> :

```
<div id="payment-<input_name>-error-container">
<span class="field-validation-valid" data-valmsg-for="CreditCardInputModel.CardNumber"
data-valmsg-replace="true"></span>
</div>
```

Les messages de validation peuvent être entrés ici.

Marques substitutives groupées

Les marques substitutives d'éléments peuvent être insérées sous forme de groupe à l'aide des marques substitutives groupées. Voici un conteneur affiché avec des marques substitutives groupées : <div id="payment-<input_name>-container"></div>. Nous insérerons des marques substitutives d'éléments dans ce groupe. En effet, nous afficherons :

```
<div id="payment-<input_name>-container">
  $$$<INPUT_NAME> LABEL$$$
  $$$<INPUT_NAME> INPUT$$$
  $$$<INPUT_NAME> ERROR$$$
  $$$<INPUT_NAME> INFO$$$
</div>
```

Un modèle principal peut avoir des marques substitutives dans <div id="paymentForm">. Les marques substitutives (comme \$\$\$CARD NUMBER\$\$\$, \$\$\$CARDHOLDER NAME\$\$\$, \$\$\$EXPIRY DATE\$\$\$, etc.) définissent le contenu du formulaire de la carte de crédit. Les commerçants peuvent sélectionner les marques substitutives qui répondent le mieux à leurs exigences.

Voici un exemple de modèle principal avec marques substitutives pour un formulaire de carte de crédit étiqueté « paymentForm » :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <link rel="stylesheet" href="$$$TP_RESOURCES_URL$$$/newuser.css"/>
</head>
<body>
  <div id="paymentForm">
    $$$CARD NUMBER$$$
    $$$CARDHOLDER NAME$$$
    $$$EXPIRY DATE$$$
    $$$CVC$$$
    $$$SUBMIT$$$
    $$$CANCEL$$$
  </div>
</body>
</html>
```

Pour créer un formulaire de prélèvement, les commerçants peuvent utiliser des marques substitutives de prélèvement comme celle qui suit.

```
$$$ACCOUNT OWNER$$$
$$$BANK ACCOUNT$$$
$$$BIC$$$
```

Le nom du titulaire du compte bancaire, l'IBAN et le code BIC afficheront un formulaire de prélèvement.

Les commerçants qui souhaitent un modèle prenant en charge plusieurs modes de paiement peuvent créer un modèle principal et un modèle partiel pour les modes de paiement.

Le modèle principal est comme suit :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <script src="$$TP_RESOURCES_URL$$/jquery-2.1.4.min.js"></script>
  <link rel="stylesheet" href="$$TP_RESOURCES_URL$$/user.css"/>
  <script type="text/javascript" src="$$TP_RESOURCES_URL$$/existinguser.js"></script>
</head>

<body>
  <div id="paymentForm">
    $$$PAYMENT_ZONE$$$
  </div>
</body>
</html>
```

Le modèle partiel est comme suit, et il est créé avec les marques substitutives (\$\$CARD NUMBER\$\$, \$\$CARDHOLDER NAME\$\$, \$\$EXPIRY DATE\$\$, and so on):

```
<div id="paymentForm">
  $$$CARD_NUMBER$$$
  $$$CARDHOLDER_NAME$$$
  $$$EXPIRY_DATE$$$
  $$$CVC$$$
  $$$SUBMIT$$$
  $$$CANCEL$$$
</div>
```

Si vous ne souhaitez pas afficher le formulaire de paiement par défaut à l'aide de \$\$\$PAYMENT_ZONE\$\$\$, vous pouvez insérer un modèle partiel à la place.

Par exemple, si le mode de paiement demandé est une carte de crédit, un modèle partiel du nom de **master.creditcard.htm** sera utilisé. Si c'est un prélèvement, un modèle partiel du nom de **master.directdebit.htm** sera utilisé.

Les commerçants devront charger ce modèle partiel dans le gestionnaire de fichiers. Ils devront également respecter une convention d'affectation de noms. Par exemple, si le nom du modèle principal existant est **master.html** et que les commerçants souhaitent envoyer le prélèvement comme mode de paiement, ils devront charger un modèle partiel du nom de **master_directdebit.html** dans le gestionnaire de fichiers.

4.2.2 Personnalisation du texte avec des fichiers XML

Vous pouvez également personnaliser l'aspect du texte sur les pages de tokénisation avec un fichier XML. Si aucun fichier XML n'est appliqué, c'est le formulaire de tokénisation qui sera utilisé. Une fois le fichier XML personnalisé, vous devrez le charger dans le gestionnaire de fichiers modèles. La condition préalable est de charger un modèle principal à l'avance.

Le fichier XML doit également comporter des éléments permettant d'identifier les champs. Cliquez ici pour obtenir une [liste complète des éléments](#). Voici un exemple d'élément (CardHolderNameInput) pour la saisie du nom d'un titulaire de carte :

```
<Element ElementId="CardHolderName" >
  <Languages>
    <Language Id="en_us">Card Holder Name only en</Language>
    <Language Id="fr">Card Holder Name only fr</Language>
  </Languages>
</Element>
```

Dans cet élément, les langues Anglais américain et Français sont spécifiées comme balises Language Id. La balise ID spécifie également le code de culture ISO.

Si aucun élément n'est spécifié, c'est le texte par défaut de cet élément qui sera utilisé. Et si la langue n'est pas spécifiée pour un élément, c'est la langue par défaut qui sera utilisée.

Remarque :

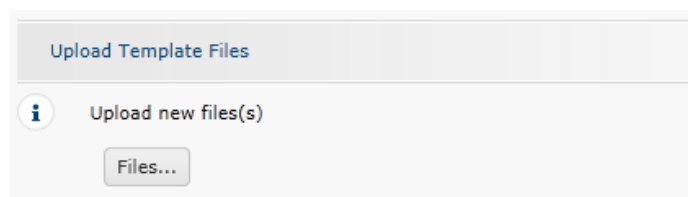
la première ligne du fichier XML doit toujours débiter par <root>. La dernière ligne du fichier XML doit toujours finir par </root>.

4.3 Gestionnaire de fichiers modèles

Avec le « Template File Manager » (Gestionnaire de fichiers modèles), vous pouvez facilement gérer vos modèles et les différents fichiers connexes.

Pour commencer à utiliser le « Gestionnaire de fichiers », connectez-vous à votre compte Ingenico ePayments et allez dans « Configuration » > « Modèle » > « Gestionnaire de fichiers ».

4.3.1 Télécharger des fichiers modèles



Dans « Télécharger les fichiers modèles », sélectionnez le bouton « Fichiers... » pour parcourir les fichiers que vous voulez télécharger. Vous pouvez télécharger des fichiers css et des images (.css, .jpg, .jpeg, .gif, .png), avec un maximum de 7 Mb par fichier, et 10 Mb au total.



Faites votre sélection puis confirmez.

4.3.2 Contrôler et gérer les fichiers téléchargés

Une fois le téléchargement terminé, vous verrez vos fichiers téléchargés sur la même page dans la partie « Fichiers téléchargés ».


Le statut des fichiers sera d'abord « En cours de validation ». Pendant ce temps, plusieurs contrôles de sécurité/virus sont réalisés (cette opération peut prendre plusieurs minutes).

Vous pouvez utiliser les fichiers lorsque leur statut est « Validé ».

Cliquez sur le bouton « Actualiser »  pour vérifier le statut de vos fichiers / Cliquez sur le bouton « Supprimer »  pour supprimer définitivement le fichier .

For creating a template, you reference your resources with the following code: "\$\$TP RESOURCES URL\$\$/[your file name]".

Drag "File Type" or "Status" column header and drop it here to group by that column

File Name	File Type	Size	Status	Upload Date	Action
successful_top.png	Image	4KB	Validating 	2015-07-08 10:13:16	
4ec6d8b8-2af4-4279-bfdc-7fe40daf9e6_5.jpg	Image	2035KB	Validated 	2015-06-25 15:52:45	
layer_styling_v5.css	CSS	9KB	Validated 	2015-06-22 14:53:16	
cancel_top.png	Image	2KB	Validated 	2015-06-16 15:55:09	

Un fichier aura le statut « Refusé » s'il ne passe pas le contrôle de sécurité. Cela peut être dû à la présence d'un virus ou à une extension de fichier erronée, par exemple.

4.3.3 Intégration

Dans votre code, renvoyez vers vos fichiers téléchargés avec un code en respectant la structure suivante : \$\$TP RESOURCES URL\$\$/[nom de votre fichier].

5. Marques substitutives

Voici une liste complète de marques substitutives et leur objet.

Marques substitutives

Marques substitutives principales	Description
\$\$\$PAYMENT_ZONE\$\$\$	Un modèle personnalisé à partir d'un modèle partiel (si disponible) ou un modèle par défaut affichant la marque ou le mode de paiement demandés.
\$\$\$TP_RESOURCES_URL\$\$\$	Appelle un fichier préalablement chargé dans le Back Office.
\$\$\$DISABLE_MASKING\$\$\$	Affiche une zone de texte correspondant au type de carte et à la mise en forme des zones d'entrée. Désactive le masquage des zones d'entrée sur détection du type de carte et de la mise en forme des zones d'entrée en fonction de la marque de la carte.
\$\$\$SUBMIT\$\$\$	Un bouton Envoyer doit être présent sur le formulaire.
\$\$\$CANCEL\$\$\$	Un bouton Annuler sur le formulaire.
\$\$\$CARD_NUMBER\$\$\$	Un numéro de carte de crédit avec validation et masquage de la marque.
\$\$\$CARD_NUMBER_LABEL\$\$\$	Une indication de numéro de carte de credit.
\$\$\$CARD_NUMBER_INPUT\$\$\$	Le champ d'entrée du numéro de carte.
\$\$\$CARD_NUMBER_INFO\$\$\$	Le champ d'entrée d'informations sur le numéro de carte.
\$\$\$CARD_NUMBER_ERROR\$\$\$	Un message d'erreur qui s'affiche en cas d'entrée d'un numéro de carte erroné.
\$\$\$CARDHOLDER_NAME\$\$\$	Un nom de titulaire de carte de crédit.
\$\$\$CARDHOLDER_NAME_LABEL\$\$\$	Une indication de nom de titulaire de carte.
\$\$\$CARDHOLDER_NAME_INPUT\$\$\$	Le champ d'entrée du nom du titulaire de la carte.
\$\$\$CARDHOLDER_NAME_INFO\$\$\$	Le champ d'entrée d'informations sur le nom du titulaire de la carte.
\$\$\$CARDHOLDER_NAME_ERROR\$\$\$	Un message d'erreur qui s'affiche en cas d'entrée d'un nom de titulaire de carte erroné.
\$\$\$EXPIRY_DATE\$\$\$	La zone d'affichage de la date d'expiration.
\$\$\$EXPIRY_DATE_LABEL\$\$\$	Une indication de date d'expiration.

\$\$\$EXPIRY DATE INPUT\$\$\$	La zone d'entrée de la date d'expiration.
\$\$\$EXPIRY DATE MONTH\$\$\$	Le mois de la date d'expiration
\$\$\$EXPIRY DATE YEAR\$\$\$	L'année de la date d'expiration
\$\$\$EXPIRY DATE INFO\$\$\$	Le champ d'entrée d'informations sur la date d'expiration.
\$\$\$EXPIRY DATE ERROR\$\$\$	Un message d'erreur qui s'affiche en cas d'entrée d'une date d'expiration erronée.
\$\$\$CVC\$\$\$	Le champ d'entrée d'un CVC avec sa validation appropriée.
\$\$\$CVC LABEL\$\$\$	Une indication de CVC.
\$\$\$CVC INPUT\$\$\$	Le champ d'entrée d'un CVC.
\$\$\$CVC INFO\$\$\$	Le champ d'affichage d'informations sur le CVC.
\$\$\$CVC ERROR\$\$\$	Un message d'erreur qui s'affiche en cas d'entrée d'un CVC erroné.
\$\$\$BANK ACCOUNT\$\$\$	Un numéro de compte pour les formulaires de prélèvement.
\$\$\$ACCOUNT OWNER\$\$\$	La zone d'affichage d'un titulaire de compte.
\$\$\$ACCOUNT OWNER LABEL\$\$\$"	Une indication de titulaire de compte.
\$\$\$ACCOUNT OWNER INPUT\$\$\$	Le champ d'entrée d'un titulaire de compte.
\$\$\$STORE PERMANENTLY\$\$\$	Une case à cocher par le commerçant pour accepter, refuser ou enregistrer définitivement l'alias.
\$\$\$BIC\$\$\$	Une case BIC pour les alias de prélèvement.

XML elements

Alias form	<ul style="list-style-type: none"> • PageTitle • Submit • Cancel • RememberMyDetails • AliasAuthorizationInfo
Credit cards	<ul style="list-style-type: none"> • PageTitle • CardNumber • CardNumberIsRequired • InvalidCardNumber • CardNumberTooLong • InvalidCardNumberCharacters • CardHolderName • CardHolderNameIsRequired • InvalidCardHolderName • CardHolderNameTooLong • ExpiryDate • InvalidExpiryDate • ExpiryMonthsRequired • InvalidExpiryMonth • InvalidExpiryMonthLength

	<ul style="list-style-type: none"> • InvalidExpiryMonthCharacters • ExpiryYearIsRequired • InvalidExpiryYear • InvalidExpiryYearLength • InvalidExpiryYearCharacters • Cvc • CvclsRequired • InvalidCvcNumber • CvcTooLong • CvcContainsNonNumericCharacters
CVC information	<ul style="list-style-type: none"> • VisaCvcInfo • EurocardCvcInfo • MasterCardCvcInfo • DinersClubCvcInfo • DankortCvcInfo • LaserCvcInfo • MaestroCvcInfo • JcbCvcInfo • AmericanExpressCvcInfo • AuroreCvcInfo • SurcoufCvcInfo • PrintempsCvcInfo • ClubMedCvcInfo • OkShoppingCvcInfo • MandarineCvcInfo • KangourouCvcInfo • GoSportCvcInfo • FinarefCvcInfo • AlsoliaCvcInfo • DefaultCvcInfo
Direct debits	<ul style="list-style-type: none"> • AccountOwnerIsRequired • AccountOwnerTooLong • InvalidAccountOwner • AccountIsRequired • AccountTooLong • InvalidAccount • BicsRequired • BicTooLong • InvalidBic • GiroAccount • BicName • Iban • BankAccount • InvalidIbanCharacters • InvalidIbanStructure • InvalidCheckDigits • InvalidIbanCountry • InvalidIbanLength • InvalidIbanNumber • InvalidBankAccountNumber • IbanOrBankAccount • IncorrectBic • BlzisRequired