

# FlexCheckout



## Inhoudsopgave

### 1. Inleiding

### 2. Normale gegevensstroom

#### 2.1 Integratie met DirectLink

#### 2.2 Gebruik met Alias Manager

### 3. FlexCheckout integreren als een tokenisatie

#### 3.1 Invoervelden

##### 3.1.1 Controleren op dubbele OrderID

#### 3.2 SHA-handtekening voor invoer

#### 3.3 Outputvelden

#### 3.4 SHA-handtekening voor output

### 4. Aanpassingen

#### 4.1 Basisaanpassingen

##### 4.1.1 Standaardsjabloon van Paypage

##### 4.1.2 Opmaakmodellen gebruiken

#### 4.2 Geavanceerde aanpassingen

##### 4.2.1 Placeholders gebruiken

##### 4.2.2 Tekstinhoud aanpassen met XML-bestanden

#### 4.3 Template File Manager (Sjabloonbestandsbeheer)

##### 4.3.1 Sjabloonbestanden uploaden

##### 4.3.2 Geüploade bestanden controleren en beheren

##### 4.3.3 Integratie

### 5. Placeholders

## 1. Inleiding

Via One Page Checkout met FlexCheckout kunt u uw klanten verzoeken hun kaartgegevens veilig op het PSP-systeem in te voeren, de gegevens tokeniseren en (optioneel) de token/alias opnieuw gebruiken bij een toekomstige transactie:

1. FlexCheckout: U leidt de FlexCheckout om naar een iframe of neemt de pagina daarin op; de kaartgegevens die de kaarthouder daar invoert, worden naar ons systeem verzonden, waar ze veilig worden opgeslagen (inclusief de CVC-code, zij het slechts kort)
2. DirectLink: U verzendt de echte bestelling met de gegenereerde alias, zonder dat u de kaartgegevens hoeft te verzenden (vgl. [integratie met DirectLink](#))

### Voordelen

- Dankzij het flexibele templatemechanisme hebt u volledige controle over de uitstraling van de betaalpagina, en als u de FlexCheckout in een Iframe opneemt, kunt u zelfs het afrekenproces beheren.
- U werkt met een veilige, PCI-compatibele betaalpagina en hoeft niet zelf de kaartgegevens op te slaan; alle gevoelige gegevens worden rechtstreeks ingevoerd op onze gehoste pagina.
- Probleemloze integratie voor verschillende afrekenscenario's, inclusief One Page Checkout.
- Op de laatste afrekenpagina kunt up-selling en cross-selling aanbieden.
- Alle grote kredietkaartmerken (VISA, MasterCard, American Express, Diners) worden ondersteund.

### Vereiste opties

Als u FlexCheckout wilt gebruiken, moet u minimaal een van de volgende opties inschakelen:

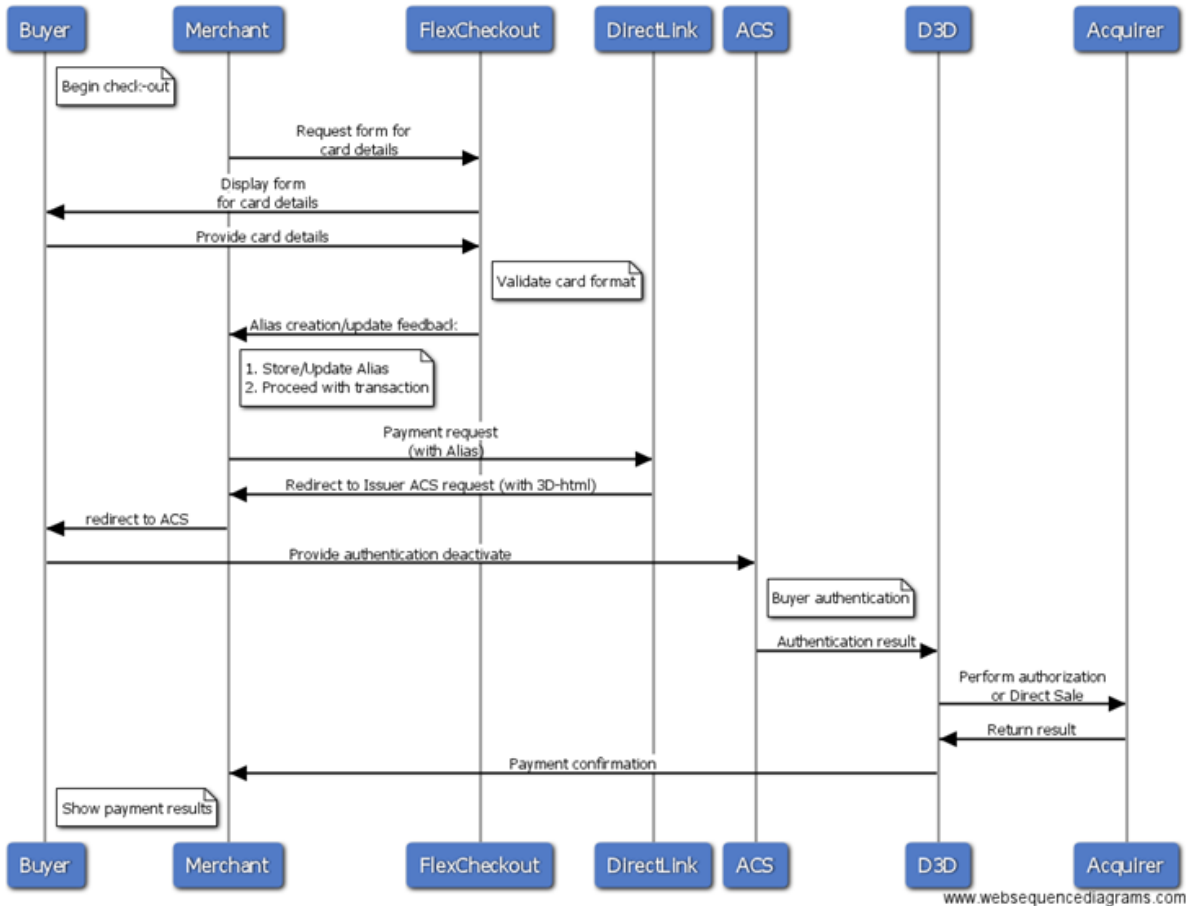
- One Page Checkout (optie-id: OPC)
- Alias Manager (optie-id: REC1, REC2, RECX)

Meestal moet u ook DirectLink (nieuwe betalingen) activeren, als die optie al niet standaard is ingeschakeld in uw abonnement.

## 2. Normale gegevensstroom

U kunt de FlexCheckout waarop de kaarthouder zijn kaartgegevens invoert in ons systeem, omleiden naar (of opnemen in) een iframe. De kaartgegevens en de CVC-code worden veilig opgeslagen, zij het slechts kort. Als u de kaartgegevens permanent wilt opslaan, kunt u de alias markeren als een permanente alias ('Persistent'). Overeenkomstig de normen van de PCI Security Council mogen CVC-codes echter maximaal twee uur worden opgeslagen.

Via DirectLink kunt u ook de echte bestelling verzenden met de gegenereerde alias, zonder dat u de kaartgegevens hoeft te verzenden.



**Opmerking:** Tijdens de eerste stap wordt er geen verrichting op de creditcard uitgevoerd. Ons systeem keurt alleen het basisformaat goed, maar kan niet garanderen dat de creditcard nog geldig is of over voldoende middelen beschikt om verder te gaan.

### 2.1 Integratie met DirectLink

Met de alias die u met behulp van de FlexCheckout hebt gegenereerd, kunt u vervolgens dankzij onze standaardtoepassing van DirectLink een DirectLink-transactie indienen. Voor meer informatie, ga naar [DirectLink](#).

### 2.2 Gebruik met Alias Manager

Bij gebruik van de Alias Manager-optie kunt u op de FlexCheckout een selectievakje weergeven, zodat de klant kan bepalen of hij al dan niet permanent zijn financiële gegevens wil opslaan.

Daartoe moet u op de configuratiepagina van Alias Manager in uw Paypage account een instelling inschakelen voor weergave van het

selectievakje (de standaardwaarde is "Nee").

Hosted Tokenization Page
Display the opt-in/opt-out checkbox on the Hosted Tokenization Page
<input type="radio"/> Yes
<input checked="" type="radio"/> No

> We adviseren u deze instelling in te schakelen bij gebruik van one-page-checkout.

Ook kunt u, als u een afrekenproces in meerdere stappen hebt geïntegreerd, gebruikmaken van de parameter ["StorePermanently"](#) in uw integratie, zodat u zelf kunt bepalen wanneer het selectievakje moet worden weergegeven.

### 3. FlexCheckout integreren als een tokenisatie

U kunt FlexCheckout op twee manieren gebruiken:

1. Door de klant om te leiden naar FlexCheckout
2. Door FlexCheckout op te nemen in een iframe, zodat klanten op uw pagina blijven en daar ook kan afrekenen

**Waarschuwing:** We raden u aan om FlexCheckout niet te integreren als pure in-app oplossing, omdat sommige functies niet werken op bepaalde smartphones van het middensegment, maar eerder in een mobiele websiteoplossing met volledige browserfuncties.

Als de klant wordt omgeleid naar FlexCheckout moet hij zijn kaartgegevens invoeren en voor tokenisatie verzenden naar FlexCheckout. Daardoor worden de kaartgegevens nooit via uw eigen webserver verzonden. De URL's voor toegang tot FlexCheckout zijn:

- Testmodus: <https://paypage.test.v-pp.com/Tokenization/HostedPage>
- Productiemodus: <https://secure.paypage.be/Tokenization/HostedPage>

De FlexCheckout werkt standaard met UTF-8-tekencodering. Wilt u in ISO werken, zorg er dan voor dat de "Karaktercodering" correct is geconfigureerd in uw backoffice: Configuratie > Technische instellingen > Algemene beveiligingsparameters > Hasing-methode.

Als de SHA-handtekening wordt berekend met de ISO-tekencodering moet de merchant dit instellen in het backoffice. Deze instelling valideert de SHA-IN- en de SHA-OUT-handtekening. Ook wordt via deze instelling het URL-coderingsschema ingesteld voor de teruggestuurde parameters.

#### 3.1 Invoervelden

Veld	Omschrijving	Formaat	Verplicht
ACCOUNT.PSPID	Identificatie van merchant	AN, 30	Ja
ALIAS.ALIASID	Klantalias	AN, 50	Nee
ALIAS.ORDERID	Identificatie van bestelling	AN, 40	Nee
ALIAS.STOREPERMANENTLY	<p>Geef aan of u een alias tijdelijk of permanent wilt opslaan. De mogelijke waarden zijn:</p> <ul style="list-style-type: none"> <li>• "N": de alias wordt na 2 uur verwijderd</li> <li>• "Y": de alias wordt permanent opgeslagen, voor toekomstig gebruik</li> </ul> <p>&gt; Raadpleeg de kader onder deze tabel voor meer informatie.</p> <p>Als een alias wordt aangemaakt met de N-waarde en de transactie binnen twee uur wordt uitgevoerd, moet deze parameter-waardecombinatie ook zijn opgenomen in de transactie, zodat de alias wordt verwijderd. Als deze parameter-waardecombinatie niet in de transactie is opgenomen, blijft de alias behouden voor toekomstig gebruik.</p>	Y / N	Nee

Veld	Omschrijving	Formaat	Verplicht
CARD.BIC	Bank Identification Code, alleen gebruikt voor automatische incasso's	AN, 8	
CARD. BIN	Credit-card-type betaalmethodes	AN, 6	
CARD.BRAND	Geef aan welk type formulier moet worden weergegeven  > Verplicht als geen "PaymentMethod" is opgegeven	AN, 25	Ja/Nee
CARD.PAYMENTMETHOD	CreditCard of ondersteunde automatische incassomethoden - Geef aan welk type formulier moet worden weergegeven  > Verplicht als geen "Brand" is opgegeven	AN, 26	Ja/Nee
LAYOUT.LANGUAGE	Taal die op de pagina wordt gebruikt ("en_EN", "nl_BE", "fr_FR" enz.)	AN, 5	Nee
LAYOUT.TEMPLATENAME	Inputparameter voor sjabloon. De merchant heeft alleen de naam van de sjabloon nodig und Dateiendung, beispielsweise "new_user.html".	AN, 255	Nee
PARAMETERS.ACCEPTURL	URL voor omleiding indien succesvol	AN, 255	Ja
PARAMETERS.EXCEPTIONURL	URL voor omleiding indien mislukt	AN, 255	Ja
PARAMETERS.EXCLUDEDPAYMENTMETHODS	Lijst van betaalmethoden en/of creditcardmerken die NIET mogen worden getoond. Gescheiden door ';' (puntkomma).	AN, 50	Nee
PARAMETERS.PARAMPLUS	Pass-Through-veld: Door de merchant te verzenden aanvullende parameters die in de output moeten worden opgehaald	AN, 1000	Nee
SHASIGNATURE.SHASIGN	Reeks gehasht met behulp van het beveiligde hashalgoritme.	AN, 128	Ja

**Alias.StorePermanently: Belangrijke opmerkingen**

- Deze parameter mag alleen worden gebruikt in combinatie met Alias Manager. Standaard wordt de alias permanent opgeslagen.
- Als u een geïntegreerde one-page-checkout gebruikt, adviseren we u met klem om de optie "opt-in/out checkbox" in uw Alias Manager-configuratie, in te schakelen, zodat op de FlexCheckout altijd het selectievakje wordt weergegeven.
- Als u de "opt-in/out checkbox"-optie in de configuratie van uw Alias Manager heeft ingeschakeld, krijgt dit voorrang op het gebruik van de parameter Alias.StorePermanently.

## 3.1.1 Controleren op dubbele OrderID

Om te voorkomen dat hackers kaartgegevens vervangen die aan een specifiek token zijn gekoppeld (door de link om het verzoek te activeren vast te krijgen en de kaartgegevens te vervangen door echte kaartgegevens van een echte kaart), controleren we de OrderID (ALIAS.ORDERID) die u in het verzoek verzendt.

Als we constateren dat de OrderID al eens is gebruikt om een token te maken, wordt de alias niet bijgewerkt.

**Belangrijk**

De controle van de OrderID werkt uiteraard alleen als de **OrderID (ALIAS.ORDERID)** wordt verzonden. Desondanks is de OrderID alleen niet voldoende; ook de **AliasID (ALIAS.ALIASID)** moet worden verzonden. Als de AliasID niet wordt verzonden, wordt de controle niet uitgevoerd en wordt een nieuwe alias aangemaakt.

Na detectie van een dubbele OrderID wordt het verzoek geblokkeerd en ziet de klant een algemene foutmelding:

"Bij de verwerking van uw verzoek is een fout opgetreden. Probeer het later opnieuw of neem contact op met uw merchant."

Fouten die worden gegenereerd door een dubbele OrderID kunnen worden vastgesteld als de foutopsporingsmodus in de Foutlogs is geactiveerd (neem hiervoor contact op met ons Customer Care team).

### 3.2 SHA-handtekening voor invoer

Om de integriteit van de gegevens te controleren, verlangen we dat alle verzoeken vergezeld gaan van een SHA-handtekening, op [dezelfde wijze als bij e-Commerce transacties](#).

Ons systeem maakt gebruik van het SHA-algoritme zoals gedefinieerd op het tabblad 'Algemene beveiligingsparameters' van de pagina 'Technische informatie'. U kunt dit algoritme altijd weer wijzigen en terugzetten naar SHA-1 of SHA-256.

**Voorbeeld**

Velden (in alfabetische volgorde):

Parameters.AcceptUrl: https://www.myshop.com/ok.html  
Parameters.ExceptionUrl: https://www.myshop.com/nok.html  
Account.PspId: test1  
Card.Brand:VISA

Geheime wachtzin (zoals gedefinieerd in Technische instellingen): Mysecretsig1875!?

Tekenreeks om te hashen:

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

Resulterende SHA-handtekening (SHA-512):

563DC909F70BA5DDD470D69C1B390E7D1C1C47705AC5801B27038446D7033B5787728EA  
754EF72E7FA2436FC5962E34E20DF64E7F9139893A33653F118816818

### 3.3 Outputvelden

De volgende velden kunnen naar u worden teruggestuurd in verband met de status van de aanmaak/bijwerking van de alias. Om de velden in de feedback op te nemen, moeten ze dienovereenkomstig worden geconfigureerd in de dynamische feedbackparameters (Paypage-account):



Configuratie > Technische instellingen > Transactiefedback > Alias gateway en Tokenization: Dynamische parameters).

De SHASIGN is niet optioneel en wordt daarom altijd teruggestuurd.

Veld	Omschrijving	Max. Lengte
ALIAS.ALIASID	Door merchant verzonden of door PSP gegenereerde alias (volgens het 32-cijferig GUID-formaat). Voorbeeld: 34F5302C-85D7-4F35-BDF5-103CCEC2FB61	50
ALIAS.NCERROR	Foutcode	50
ALIAS.NCERRORCARDNO	Foutcode voor CARDNO	50
ALIAS.NCERRORCN	Foutcode voor CN	50
ALIAS.NCERRORCVC	Foutcode voor CVC	50
ALIAS.NCERRORED	Foutcode voor ED	50
ALIAS.ORDERID	Het unieke identificatienummer van de bestelling. (verzonden door de merchant of gegenereerd door het systeem)	40
ALIAS.STATUS	Resultaat van het aanmaken van de alias: <ul style="list-style-type: none"> <li>• 0=OK</li> <li>• 1=NOK</li> <li>• 2=Alias bijgewerkt</li> <li>• 3=Geannuleerd door gebruiker</li> </ul>	1
ALIAS.STOREPERMANENTLY	Geeft de keuze aan of de alias tijdelijk (N) of permanent (Y) moet worden opgeslagen.  Wanneer met deze alias een transactie wordt uitgevoerd, moet de merchant deze parameter terugzenden in de opgeslagen ALIASPERSISTEDAFTERUSE. Anders wordt de alias permanent opgeslagen.	1 (Y/N)
CARD.BIC	Bankidentificatiecode, alleen gebruikt voor automatische incasso	8
CARD.BIN	Betaalmethoden met creditcard	6
CARD.BRAND	Merk van de betaalmethode	25
CARD.CARDHOLDERNAME	Naam van kaarthouder	50
CARD.CARDNUMBER	Kaart met X'en ter vervanging van gevoelige gegevens.  Voorbeeld: XXXXXXXXXXXX1111  Opmerking: Bij een fout wordt ook de kaart gemaskeerd.	35
CARD.CVC	Kaartverificatiecode (Card Verification Code) voor creditcards, met X'en ter	6

Veld	Omschrijving	Max. Lengte
	vervanging van gevoelige gegevens. Voorbeeld: XXX	
CARD.EXPIRYDATE	Vervaldatum, bv. 0220 (februari 2020)	4
PARAMPLUS	Pass-Through-veld: gegevens opgegeven in de input	/
<b>SHASIGN</b>	<b>SHA-handtekening voor output</b>	<b>128</b>

De geselecteerde parameters worden toegevoegd aan de in uw verzoek gedefinieerde retour-URL (PARAMETERS.ACCEPTURL of PARAMETERS.EXCEPTIONURL) in verband met de feedback over de bewerking.

### 3.4 SHA-handtekening voor output

Ons systeem stuurt, [op dezelfde wijze als bij e-Commerce-transacties](#), een SHA-OUT-handtekening terug voor de volgende parameters:

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. Aanpassingen

In dit hoofdstuk worden twee soorten aanpassingen beschreven: basis- en geavanceerde aanpassingen. Onder Basisaanpassingen leest u hoe u uw eigen css-bestand kunt toepassen en onder andere tekstkleuren, achtergrond en wrapperkleuren kunt aanpassen. Hieronder is een standaardbetaalpagina te zien via de standaard-css-bestanden:

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.

Onder Geavanceerde aanpassingen leest u meer over hoofd-/deelsjablonen, het gebruik van placeholders en het aanpassen van tekstinhoud met XML-bestanden. Afhankelijk van uw behoeften worden de deelsjablonen apart aangemaakt en aangepast.

### 4.1 Basisaanpassingen

Bij een basisaanpassing wordt het standaardopmaakmodel geconfigureerd via de standaardsjabloon van Paypage.

#### 4.1.1 Standaardsjabloon van Paypage

Het css-bestand kan worden aangepast met een standaardopmaakmodel van Paypage of met een geüpload css-bestand in [Sjabloonbestandsbeheer](#).

Zie voor meer informatie over het aanpassen van uw eigen css-bestand '[Opmaakmodellen gebruiken](#)'.

The image shows a 'Style sheet' configuration window with the following details:

- Title:** Style sheet
- Information icon:** A small 'i' icon.
- Label:** Stylesheet filename
- Dropdown menu:** Contains 'Default Ogone template' (selected) and 'layer\_styling\_v5.css'.
- Button:** A grey 'SAVE' button.

#### 4.1.2 Opmaakmodellen gebruiken

U kunt de FlexCheckout-layout aanpassen door een CSS-bestand toe te passen. Eerst dient u uw CSS-bestand te uploaden naar de Template File Manager en deze te koppelen aan uw Paypage account. Het CSS-bestand bestaat uit een form/data container en elementen (zoals: vervaldag, CVC, informatievak enz.) die als het standaard CSS-bestand moeten worden ingesteld. U kunt elk element dat de formattering voor elk veld definieert, koppelen aan de form/data container.

Op basis van uw bedrijfsbehoeften kunt u de formatteringsregel in elk element wijzigen. Klik hier voor een lijst met [CSS-elementen](#).

Hierna vindt u voorbeelden van een form/data container en elementen:

```
/*-----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;}

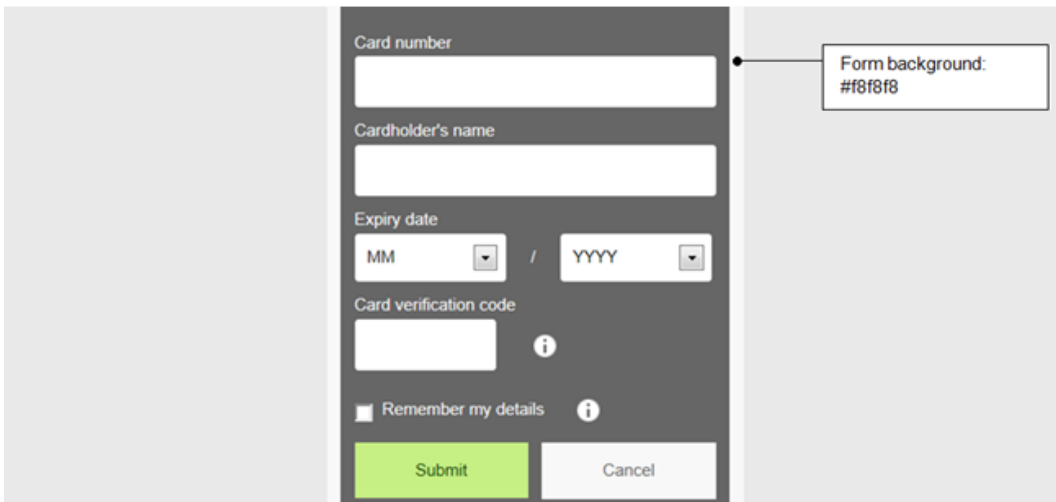
#payment-cardnumber-label-container,#payment-accountowner-label-container{ overflow: hidden;}

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

Om te beginnen hebben we een monster CSS-bestand dat u kunt downloaden vanaf [hier](#) (zip-bestand) gemaakt. U kunt beginnen met het maken van uw eigen CSS met dit bestand.

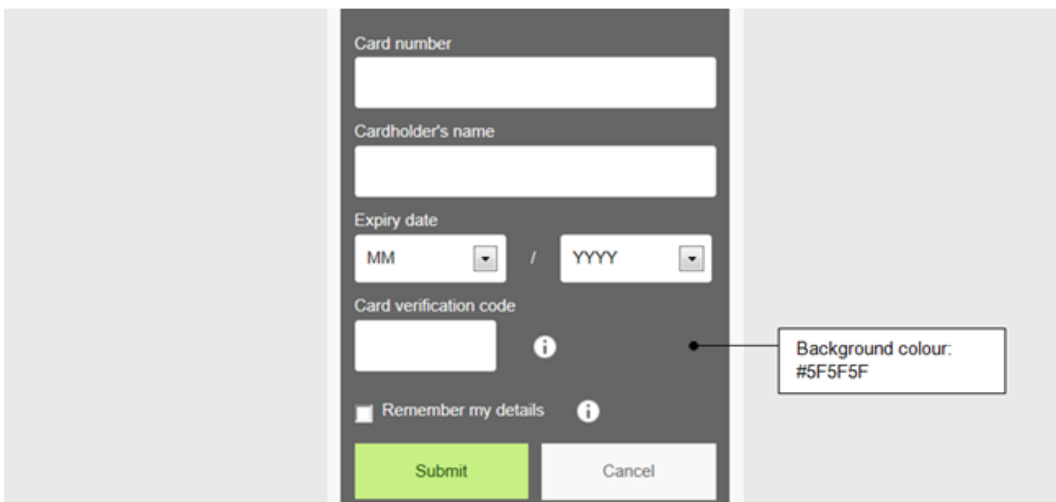
U kunt de standaard gebruikersinterface veranderen (zoals hieronder afgebeeld) door het toepassen van uw eigen CSS.

### Kleur van formulierachtergrond



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

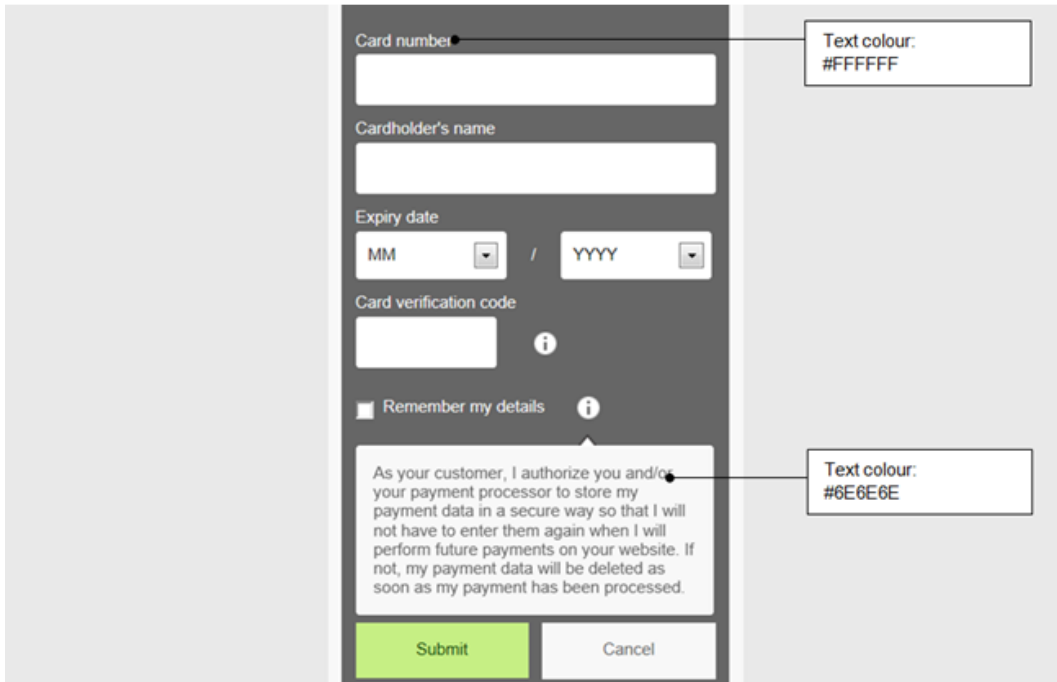
### Kleur van formulierwrapper



Effect op CSS:

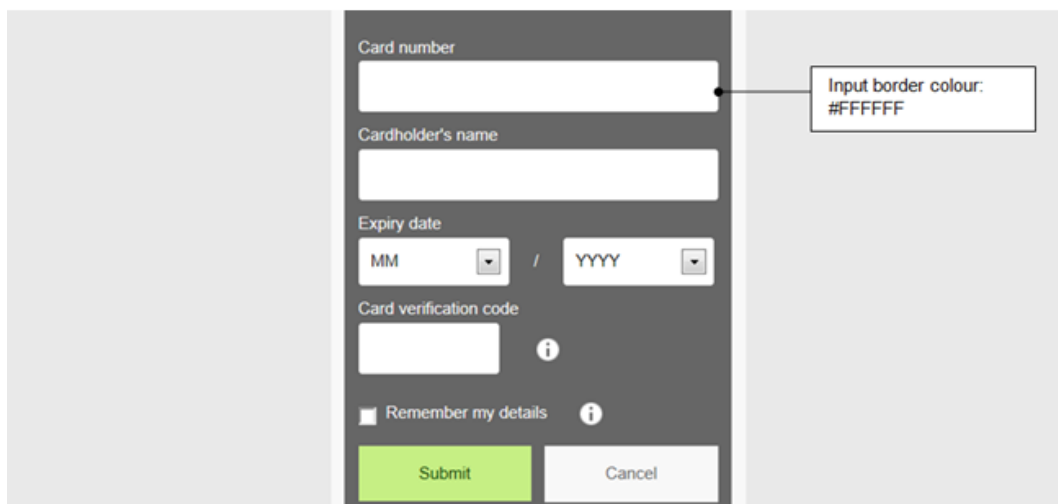
`#payment-data-container{ Background:#5F5F5F; }`

Tekstkleur



Effect op CSS: #payment-data-container .payment-label{color:#FFFFFF;} #payment-data-container{color:#6E6E6E;}

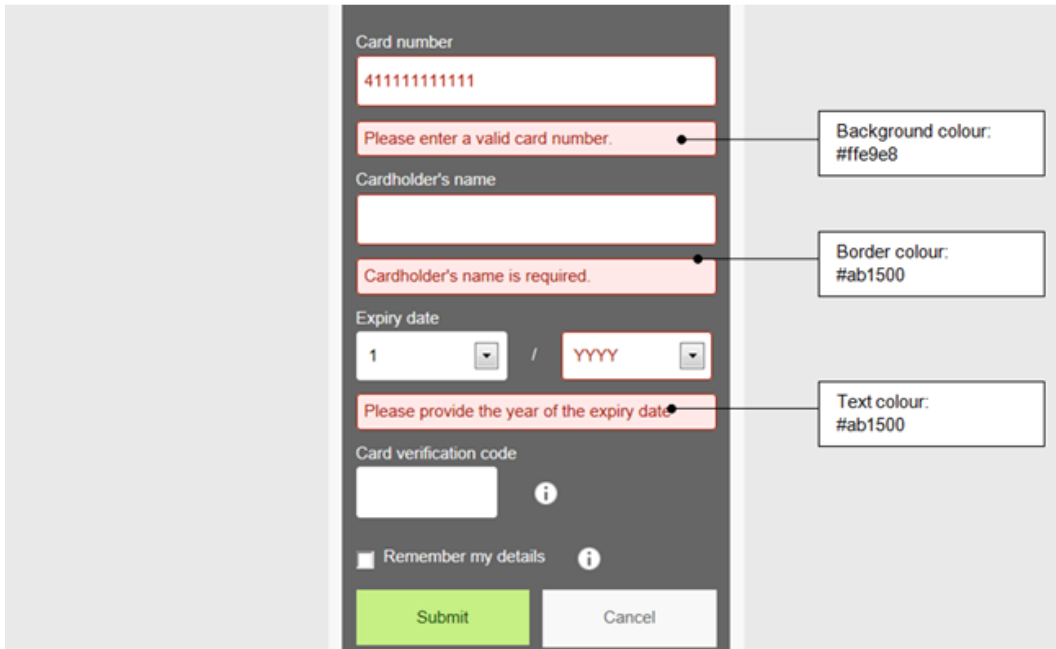
Kleur van rand inputveld



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

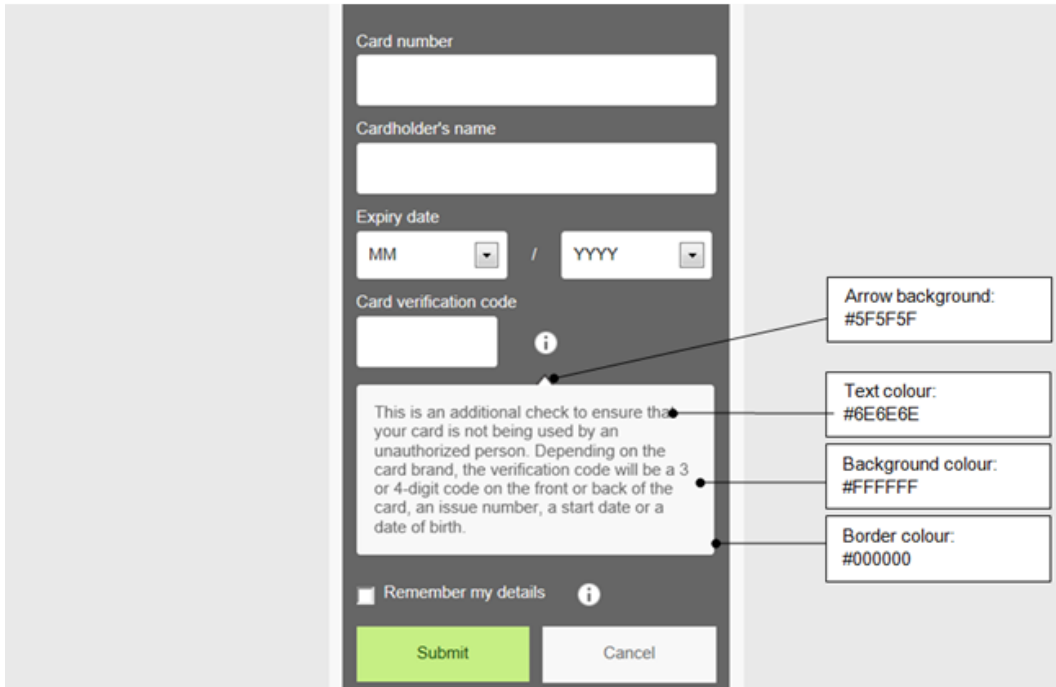
Vak voor validatiebericht





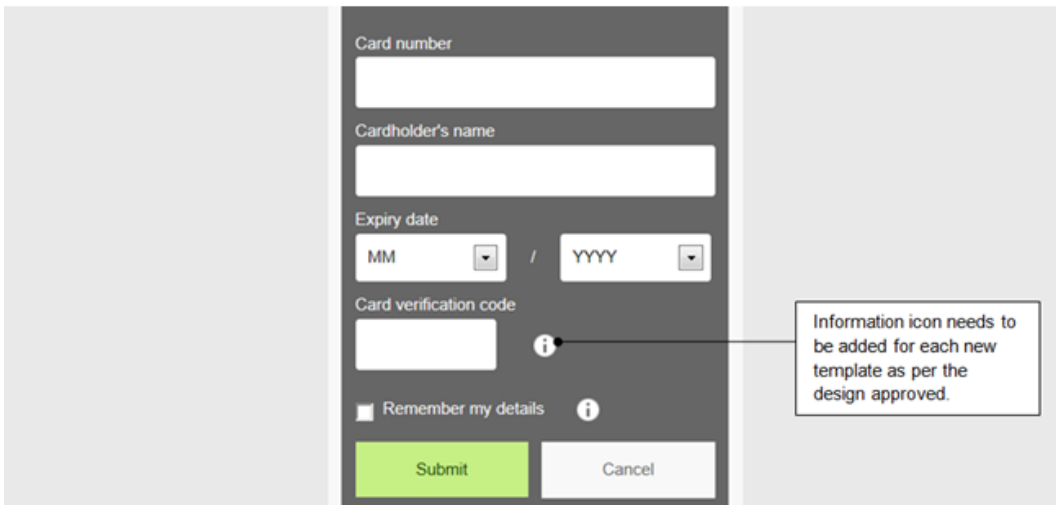
Effect op CSS: .payment-error { color:#ab1500; background:#ffe9e8; border:1px solid #ab1500;} #payment-data-container .payment-input-error input { color:#ab1500; border:1px solid #ab1500;} #payment-data-container .payment-input-error select { color:#ab1500; border:1px solid #ab1500;}

Vak voor hulpinformatie



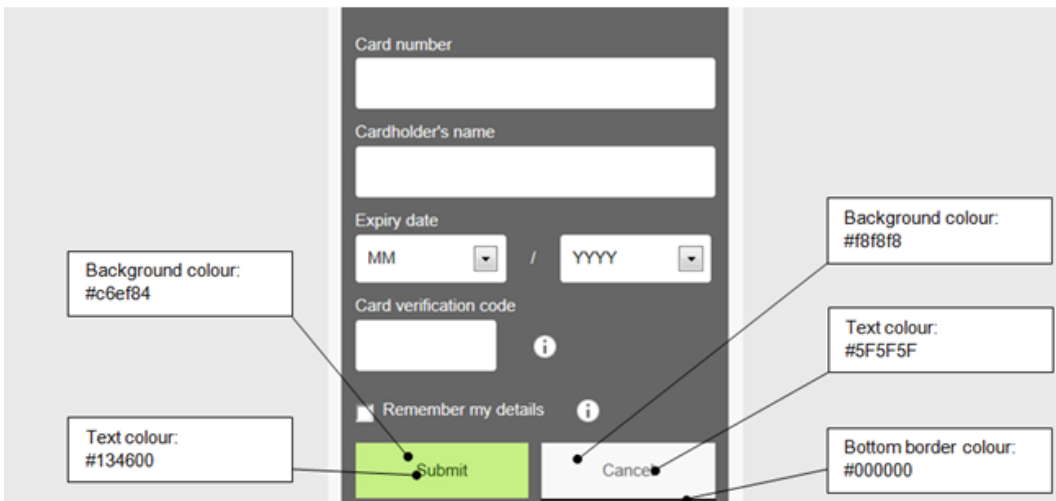
Effect op CSS: #payment-cvc-info-container span.arowlmg {background: #5F5F5F url(arrow-top.png) 0 0 no-repeat;} #payment-cvc-info-container p.cvc-info-txt{color: #6E6E6E;} #payment-cvc-info-container div.help-box {border: 1px solid #000000;background: #FFFFFF;}

### Informatiepictogram



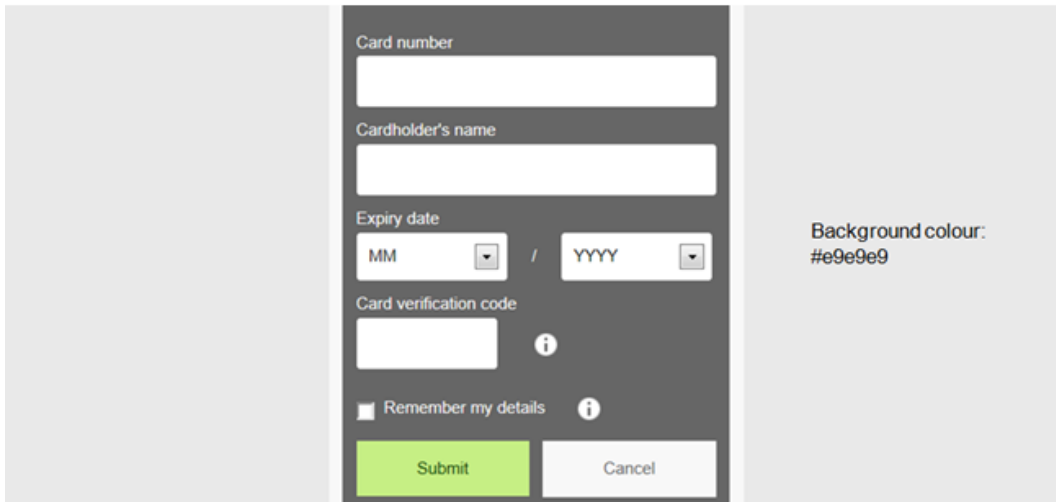
Effect op CSS: .payment-method-info {background:url(payment\_info.png) 4px 1px no-repeat;}

### Knop Verzenden/Annuleren



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

## Achtergrondkleur



Effect op CSS:

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

### Voorbeeld 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.aroolmg {background: 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 Geavanceerde aanpassingen

Voor meer flexibiliteit kunt u voortaan hoofd- en deelsjablonen aanmaken en aanpassen. Afhankelijk van uw behoeften worden de deelsjablonen apart aangemaakt en aangepast. U kunt de tekstinhoud ook aanpassen met XML-bestanden.

U kunt ervoor kiezen om de tag `$$$PAYMENT_ZONE$$$` toe te passen, waarmee een eenvoudige sjabloon wordt opgeroepen, of uw eigen deelsjabloon aanpassen [via placeholders](#).

### 4.2.1 Placeholders gebruiken

U kunt aanpassen en versier je eigen meester template (master.html) en delen sjablonen kunnen direct invoegen placeholders in de master-template master.html.

U kunt ook het uiterlijk van de betaalpagina definiëren via een aangepast CSS-bestand (**newuser.css**) dat moet worden gekoppeld aan de

master.html. Als u uw CSS-bestand wilt aanpassen, moet het gewijzigde CSS-bestand worden geüpload naar de Template File Manager en als standaard worden ingesteld.

Placeholders worden ingedeeld in drie groepen: main placeholders, element placeholders en gegroepeerde placeholders.

### **Main placeholders**

Hieronder volgt een lijst van main placeholders:

- **\$\$\$PAYMENT\_ZONE\$\$\$**: Geeft het aangevraagde merk of de aangevraagde betaalmethode weer via een aangepaste sjabloon (met een beschikbare deelsjabloon) of de standaardjabloon.
- **\$\$\$TP\_RESOURCES\_URL\$\$\$**: Maakt de URL beschikbaar voor de directory waarin zich de bestanden in de Hosted File Service bevinden.

U kunt ook aangepaste scripts gebruiken die aan een aangepast JavaScript-bestand moeten worden gekoppeld. Nadat de scripts zijn geüpload naar de Template File Manager kan dan naar dit JavaScript-bestand worden verwezen. Zo is bijvoorbeeld existinguser.js het JavaScript-bestand waarnaar wordt verwezen in `<script type="text/javascript" src="$$$TP_RESOURCES_URL$$$/existinguser.js"></script>`.

Om JavaScript-bibliotheken in de sjablonen te gebruiken (bijv. JQuery), dient u de JavaScript-bestanden te uploaden naar de Template File Manager. Na te zijn geüpload moet **\$\$\$TP\_RESOURCE\_URL\$\$\$** worden gebruikt om naar de scripts te verwijzen. Deze stap moet worden uitgevoerd voor alle bestanden waarnaar in de sjabloon wordt verwezen: CSS, afbeeldingen, lettertypen enz., en verzekert dat het uitgevoerde script het specifieke JavaScript is.

**Opmerking:** Niet alle data kunnen van/naar een niet-PCI-compatibele derde host gaan, daarom komen merchants die JavaScript gebruiken mogelijk niet in aanmerking voor PCI SAQ A.

- **\$\$\$DISABLE\_MASKING\$\$\$**: Schakelt een masker van een invoervak uit.
- **\$\$\$SUBMIT\$\$\$**: Geeft een verzendknop weer voor het formulier. Bijvoorbeeld:

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

- **\$\$\$CANCEL\$\$\$**: Geeft een annuleerknop weer voor het formulier. Bijvoorbeeld:

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

In een hoofdsjabloon die **\$\$\$PAYMENT\_ZONE\$\$\$** bevat, hoeven geen placeholders voor verzenden en annuleren te staan. Placeholders voor verzenden en annuleren moeten echter wel worden opgenomen als een deelsjabloon wordt gemaakt per betaalmethode, of als een hoofdsjabloon wordt gemaakt zonder **\$\$\$PAYMENT\_ZONE\$\$\$**-placeholder.

### **Element placeholders**

De feitelijke control `<element>` wordt weergegeven in een element container: `<div id="payment-<input_name>-<element>-container" class="payment-<element>">`. Voor een eenvoudige css-styling is zowel op id als klasse dezelfde naamgevingsconventie van toepassing.

Element placeholders geven HTML-controls weer voor verschillende invoerelementen en moeten aan de volgende opmaak voldoen:

- **\$\$\$<input\_name><element>\$\$\$**: als de invoercontrole van het kaartnummer.

- `$$$CARD NUMBER.INPUT$$$` - CARD NUMBER: als de `<input_name>` waarbij INPUT het HTML `<element>` is.
- `$$$<input_name>LABEL$$$`: geeft een label weer voor het element `<input_name>`.

De placeholder `$$$CARD NUMBER LABEL$$$` geeft bijvoorbeeld het volgende weer:

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

Voor alle labels wordt een div weergegeven die dienstdoet als container:

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

De klasse van een labelcontainer is altijd "payment-label", zodat de labels samen kunnen worden opgemaakt.

Voor een label control wordt altijd een label weergegeven:

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

Om een label control op te maken, moet het css-bestand worden toegepast op de container-id, niet op de control-id.

`$$$<input_name> INPUT$$$` geeft de input control weer voor `<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>
```

Om een input control op te maken, moet het css-bestand worden toegepast op de container-id, niet op de control-id.

`$$$<input_name> INFO$$$` geeft een koppeling weer naar een beschrijving van het veld `<input_name>`:

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

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

</div>
```

Alle gegevens voor dit veld moeten worden gebundeld in de geselecteerde taal.

\$\$\$<input\_name> ERROR\$\$\$ geeft een container weer voor foutberichten van <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>
```

Hier kunnen validatieberichten worden ingevoegd.

### **Gegroepeerde placeholders**

De element placeholders kunnen als groep worden ingevoegd door middel van gegroepeerde placeholders. Hieronder is een weergegeven container te zien met een gegroepeerde placeholder <div id="payment-<input\_name>-container"></div> waarin we de element placeholders invoegen in deze groep. Dit wordt als zodanig weergegeven.

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

$$$<INPUT_NAME> LABEL$$$

$$$<INPUT_NAME> INPUT$$$

$$$<INPUT_NAME> ERROR$$$

$$$<INPUT_NAME> INFO$$$

</div>
```

waarna de element placeholders zoals hierboven worden vervangen.

In een hoofdsjabloon kunnen zich placeholders bevinden in <div id="paymentForm">. De placeholders (zoals \$\$\$CARD NUMBER\$\$\$, \$\$\$CARDHOLDER NAME\$\$\$, \$\$\$EXPIRY DATE\$\$\$, enz.) definiëren de inhoud van het creditcardformulier. Merchants kunnen de placeholders selecteren die voor hen het meest geschikt zijn.

Hieronder een voorbeeld van een hoofdsjabloon met placeholders voor een creditcardformulier met het label "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>
```

Om een formulier voor automatische incasso te maken, kunnen merchants placeholders voor automatische incasso gebruiken, zoals die hieronder:

```
$$$ACCOUNT_OWNER$$$

$$$BANK_ACCOUNT$$$

$$$BIC$$$
```

De naam van de bankrekening, het IBAN-nummer en de BIC-code geven een formulier voor automatische incasso weer.

Merchants die een sjabloon willen die meerdere betaalmethoden ondersteunt, kunnen voor die betaalmethoden een hoofd- en deelsjabloon maken.

Het hoofdsjabloon ziet er als volgt uit:

```
<!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>

```

Een deelsjabloon ziet als volgt uit en wordt gemaakt met placeholders (\$\$\$CARD NUMBER\$\$\$, \$\$\$CARDHOLDER NAME\$\$\$, \$\$\$EXPIRY DATE\$\$\$, and so on):

```

<div id="paymentForm">

$$$CARD NUMBER$$$

$$$CARDHOLDER NAME$$$

$$$EXPIRY DATE$$$

$$$CVC$$$

$$$SUBMIT$$$

$$$CANCEL$$$

</div>

```

Als de merchant het standaardbetaalformulier niet wil weergeven met behulp van \$\$\$PAYMENT ZONE\$\$\$ kan hij in plaats daarvan een deelsjabloon invoegen.

Is de aangevraagde betaalmethode een creditcard is, dan wordt een deelsjabloon met de naam master.creditcard.htm gebruikt. Gaat het om een automatische incasso, dan wordt in plaats daarvan een deelsjabloon met de naam master.directdebit.htm gebruikt.

De merchant moet dit deelsjabloon uploaden naar File Manager (Bestandsbeheer). Ook de naamgevingsconventie moet strikt worden nageleefd. Als de naam van de bestaande hoofdsjabloon bijvoorbeeld master.html is en als de merchant automatische incasso als betaalmethode wil verzenden, dan moet een deelsjabloon met de naam master\_directdebit.html worden geüpload naar File Manager (Bestandsbeheer).

#### 4.2.2 Tekstinhoud aanpassen met XML-bestanden

You can also customize the text appearance on the tokenization pages using an XML file. The tokenization form will be used if an XML file is not applied. After customizing the XML file, you will need to upload it to the Template File Manager and use the same file name as the master template.

Note: As a prerequisite, a master template should be uploaded beforehand.

Met het XML-bestand moeten ook elementen voorzien zijn om de velden te identificeren. Klik hier voor een volledige [lijst van elementen](#). Hierna volgt een voorbeeld van een element (CardHolderNameInput) voor de invoer van een kaarthoudernaam:

```
<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>
```

In dit element worden de talen US Engels en Frans gespecificeerd in de Language Id. De ID-tag specificeert ook de ISO-cultuurcode.

Als geen elementen gespecificeerd zijn, wordt de standaardtekst voor dat element gebruikt. Als de taal voor een element niet gespecificeerd is, wordt de standaardtaal gebruikt.

Opmerking:

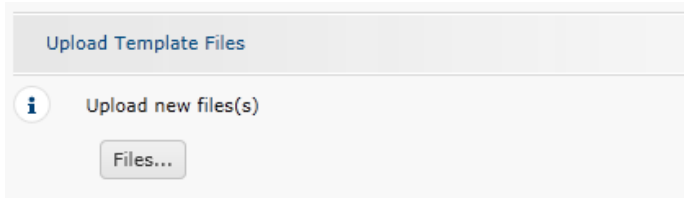
De eerste regel van het XML-bestand moet altijd beginnen met <root>. De laatste regel van het XML-bestand moet altijd eindigen met </root>

#### 4.3 Template File Manager (Sjabloonbestandsbeheer)

Met Template File Manager (Sjabloonbestandsbeheer) kunt u eenvoudig uw sjablonen en de bijbehorende bestanden beheren.

Als u File Manager wilt gebruiken, meldt u zich aan bij uw Paypage account en selecteert u 'Configuratie' > 'Sjabloon' > 'File Manager'.

### 4.3.1 Sjabloonbestanden uploaden



Klik onder 'Sjabloonbestanden uploaden' op 'Bestanden...' om te zoeken naar de bestanden die u wilt uploaden. U kunt css en afbeeldingen (.css, .jpg, .jpeg, .gif, .png) uploaden (maximaal 7 MB per bestand en 10 MB in totaal).


Selecteer de bestanden en bevestig uw keuze.

### 4.3.2 Geüploadede bestanden controleren en beheren

Als de bestanden zijn geüpload, worden ze op dezelfde pagina weergegeven in het gedeelte 'Geüploadede bestanden'.

De bestanden hebben eerst de status 'Bezig met valideren'. In deze fase worden een aantal noodzakelijke veiligheids-/viruscontroles uitgevoerd (dit kan enkele minuten duren).

Zodra de bestanden de status 'Gevalideerd' hebben, kunt u ze gebruiken.

Klik op de knop 'Vernieuwen'  om de huidige status van uw bestanden te controleren; klik op de knop 'Verwijderen'  om een bestand permanent te verwijderen.

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_6.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	

Als een bestand de status 'Geweigerd' heeft, is het niet door de veiligheidscontrole gekomen. Er kan dan bijvoorbeeld een virus zijn aangetroffen of de bestandsextensie is onjuist.

### 4.3.3 Integratie

In uw code verwijst u naar uw geüploadede bestanden met een code die de volgende structuur heeft: "\$\$\$TP RESOURCES URL\$\$\$/[uw bestandsnaam]".

## 5. Placeholders

Hieronder volgt een lijst van placeholders en het doel ervan:

<b>Main Placeholders</b>	<b>Beschrijving</b>
\$\$\$PAYMENT_ZONE\$\$\$	Een aangepaste sjabloon op basis van een deelsjabloon (indien beschikbaar) of een standaardjabloon die het aangevraagde merk of de aangevraagde betaalmethode weergeeft.
\$\$\$TP_RESOURCES_URL\$\$\$	Roept een bestand op dat eerder is geüpload naar het backoffice.
\$\$\$DISABLE_MASKING\$\$\$	Geeft een tekstvak weer dat bij de kaarttypen en invoeropmaak past. Schakelt het masker van een invoervak uit na detectie van het kaarttype en de opmaak van de invoervakken op basis van het kaartmerk.
\$\$\$SUBMIT\$\$\$	Op het formulier moet een verzendknop staan.
\$\$\$CANCEL\$\$\$	Een annuleerknop op het formulier.
\$\$\$CARD_NUMBER\$\$\$	Een creditcardnummer met de juiste validatie en maskering voor het merk.
\$\$\$CARD_NUMBER_LABEL\$\$\$	Een aanduiding voor een creditcardnummer.
\$\$\$CARD_NUMBER_INPUT\$\$\$	Het veld waarin een kaartnummer wordt ingevoerd.
\$\$\$CARD_NUMBER_INFO\$\$\$	Het veld waarin een kaartnummer wordt ingevoerd.
\$\$\$CARD_NUMBER_ERROR\$\$\$	Een foutbericht dat verschijnt wanneer een verkeerd kaartnummer wordt ingevoerd.
\$\$\$CARDHOLDER_NAME\$\$\$	De naam van de creditcardhouder.
\$\$\$CARDHOLDER_NAME_LABEL\$\$\$	Een aanduiding voor de naam van de kaarthouder.
\$\$\$CARDHOLDER_NAME_INPUT\$\$\$	Het veld waarin de naam van de kaarthouder wordt ingevoerd.
\$\$\$CARDHOLDER_NAME_INFO\$\$\$	Het veld waarin de naam van de kaarthouder wordt ingevoerd.
\$\$\$CARDHOLDER_NAME_ERROR\$\$\$	Een foutbericht dat verschijnt wanneer de naam van de kaarthouder verkeerd wordt ingevoerd.
\$\$\$EXPIRY_DATE\$\$\$	De ruimte waar een vervaldatum wordt weergegeven.
\$\$\$EXPIRY_DATE_LABEL\$\$\$	Een aanduiding voor een vervaldatum.

\$\$\$EXPIRY DATE INPUT\$\$\$	De ruimte waar een vervaldatum wordt ingevoerd.
\$\$\$EXPIRY DATE MONTH\$\$\$	De maand van de vervaldatum
\$\$\$EXPIRY DATE YEAR\$\$\$	Het jaar van de vervaldatum
\$\$\$EXPIRY DATE INFO\$\$\$	Het veld waarin een vervaldatum wordt ingevoerd.
\$\$\$EXPIRY DATE ERROR\$\$\$	Een foutbericht dat verschijnt wanneer een vervaldatum verkeerd wordt ingevoerd.
\$\$\$CVC\$\$\$	Het veld waarin een CVC met de juiste validatie ervan wordt geplaatst.
\$\$\$CVC LABEL\$\$\$	Een aanduiding voor CVC.
\$\$\$CVC INPUT\$\$\$	Het veld waarin een CVC wordt ingevoerd.
\$\$\$CVC INFO\$\$\$	Het veld waarin de CVC-gegevens worden weergegeven.
\$\$\$CVC ERROR\$\$\$	Een foutbericht dat verschijnt wanneer een CVC verkeerd wordt ingevoerd.
\$\$\$BANK ACCOUNT\$\$\$	Een rekeningnummer voor formulieren voor automatische incasso.
\$\$\$ACCOUNT OWNER\$\$\$	De ruimte waar een rekeninghouder wordt weergegeven.
\$\$\$ACCOUNT OWNER LABEL\$\$\$"	Een aanduiding voor een rekeninghouder.
\$\$\$ACCOUNT OWNER INPUT\$\$\$	Het veld waarin een rekeninghouder wordt weergegeven.
\$\$\$STORE PERMANENTLY\$\$\$	Een selectievakje voor de merchant om ofwel opt-in of opt-out te kiezen ofwel de alias permanent op te slaan.
\$\$\$BIC\$\$\$	Een BIC-vak voor aliassen voor automatische incasso.

## XML elements

<b>Alias form</b>	<ul style="list-style-type: none"> <li>• PageTitle</li> <li>• Submit</li> <li>• Cancel</li> <li>• RememberMyDetails</li> <li>• AliasAuthorizationInfo</li> </ul>
<b>Credit cards</b>	<ul style="list-style-type: none"> <li>• PageTitle</li> <li>• CardNumber</li> <li>• CardNumbersRequired</li> </ul>

	<ul style="list-style-type: none"> <li>• InvalidCardNumber</li> <li>• CardNumberTooLong</li> <li>• InvalidCardNumberCharacters</li> <li>• CardHolderName</li> <li>• CardHolderNameIsRequired</li> <li>• InvalidCardHolderName</li> <li>• CardHolderNameTooLong</li> <li>• ExpiryDate</li> <li>• InvalidExpiryDate</li> <li>• ExpiryMonthsRequired</li> <li>• InvalidExpiryMonth</li> <li>• InvalidExpiryMonthLength</li> <li>• InvalidExpiryMonthCharacters</li> <li>• ExpiryYearsRequired</li> <li>• InvalidExpiryYear</li> <li>• InvalidExpiryYearLength</li> <li>• InvalidExpiryYearCharacters</li> <li>• Cvc</li> <li>• CvcIsRequired</li> <li>• InvalidCvcNumber</li> <li>• CvcTooLong</li> <li>• CvcContainsNonNumericCharacters</li> </ul>
<p><b>CVC information</b></p>	<ul style="list-style-type: none"> <li>• VisaCvcInfo</li> <li>• EurocardCvcInfo</li> <li>• MasterCardCvcInfo</li> <li>• DinersClubCvcInfo</li> <li>• DankortCvcInfo</li> <li>• LaserCvcInfo</li> <li>• MaestroCvcInfo</li> <li>• JcbCvcInfo</li> <li>• AmericanExpressCvcInfo</li> <li>• AuroreCvcInfo</li> <li>• SurcoufCvcInfo</li> <li>• PrintempsCvcInfo</li> <li>• ClubMedCvcInfo</li> <li>• OkShoppingCvcInfo</li> <li>• MandarineCvcInfo</li> <li>• KangourouCvcInfo</li> <li>• GoSportCvcInfo</li> <li>• FinarefCvcInfo</li> <li>• AlsoliaCvcInfo</li> <li>• DefaultCvcInfo</li> </ul>
<p><b>Direct debits</b></p>	<ul style="list-style-type: none"> <li>• AccountOwnerIsRequired</li> <li>• AccountOwnerTooLong</li> <li>• InvalidAccountOwner</li> <li>• AccountIsRequired</li> <li>• AccountTooLong</li> <li>• InvalidAccount</li> <li>• BicsRequired</li> <li>• BicTooLong</li> <li>• InvalidBic</li> <li>• GiroAccount</li> <li>• BicName</li> <li>• Iban</li> <li>• BankAccount</li> <li>• InvalidIbanCharacters</li> <li>• InvalidIbanStructure</li> <li>• InvalidCheckDigits</li> <li>• InvalidIbanCountry</li> <li>• InvalidIbanLength</li> <li>• InvalidIbanNumber</li> <li>• InvalidBankAccountNumber</li> <li>• IbanOrBankAccount</li> </ul>

	<ul style="list-style-type: none"><li>• IncorrectBic</li><li>• BizIsRequired</li></ul>
--	--