

FlexCheckout

ingenico
ePayments

Inhaltsverzeichnis

1. Einführung

2. Typischer Datenfluss

2.1 DirectLink-Integration

2.2 Verwendung mit Alias Manager

3. FlexCheckout als Tokenisierung integrieren

3.1 Input-Felder

3.1.1 Prüfung auf doppelte OrderID

3.2 SHA-Signatur für Input

3.3 Output fields

3.4 SHA-Signatur für Output

4. Anpassung

4.1 Grundlegende Anpassung

4.1.1 Ingenico ePayments-Standardvorlage

4.1.2 Verwendung von Stylesheets

4.2 Erweiterte Anpassung

4.2.1 Verwenden von Platzhaltern

4.2.2 Anpassen von Textinhalt mit XML

4.3 Vorlagen-Dateimanager

4.3.1 Vorlagendateien hochladen

4.3.2 Hochgeladene Dateien prüfen und verwalten

4.3.3 Integration

5. Platzhalter

1. Einführung

Der One-Page-Checkout mit FlexCheckout ermöglicht Ihnen, Ihre Kunden aufzufordern, ihre Kartendaten abgesichert in das PSP-System einzugeben, diese Daten mit einem Token zu verknüpfen und (optional) das Token/Alias bei einer künftigen Transaktion zu verwenden:

1. FlexCheckout: Sie können eine Umleitung zur FlexCheckout einrichten oder die Seite in einem iframe einbinden. Auf dieser Seite gibt der Karteninhaber die Kartendaten in unser System ein, wo sie sicher gespeichert werden (einschließlich Kartenprüfnummer (CVC), diese jedoch nur für begrenzte Zeit).
2. DirectLink: Sie reichen den jeweiligen Auftrag gemeinsam mit dem generierten Alias ein. Es ist nicht notwendig, die Kartendaten zu übermitteln (siehe [DirectLink Integrationsleitfaden](#))

Vorteile

- Dank der flexiblen Vorlagenfunktion haben Sie volle Kontrolle über Optik und Bedienung der Zahlungsseite. Wenn Sie die FlexCheckout in einem iframe einbinden, behalten Sie sogar die Kontrolle über die Checkout-Sequenz.
- Sie profitieren von einer sicheren, PCI-gemäßen Zahlungsseite, ohne selbst die Kartendaten speichern zu müssen. Alle sensiblen Daten werden direkt auf unserer gehosteten Seite eingegeben.
- Nahtlose Integration für verschiedene Checkout-Szenarien einschließlich One-Page-Checkout.
- Sie können auf der abschließenden Checkout-Seite Up- und Cross-Selling anbieten.
- Unterstützt alle bekannten Kreditkartenmarken (VISA, MasterCard, American Express, Diners).

Erforderliche Optionen

Sie müssen mindestens eine der folgenden Optionen aktiviert haben, um FlexCheckout verwenden zu können:

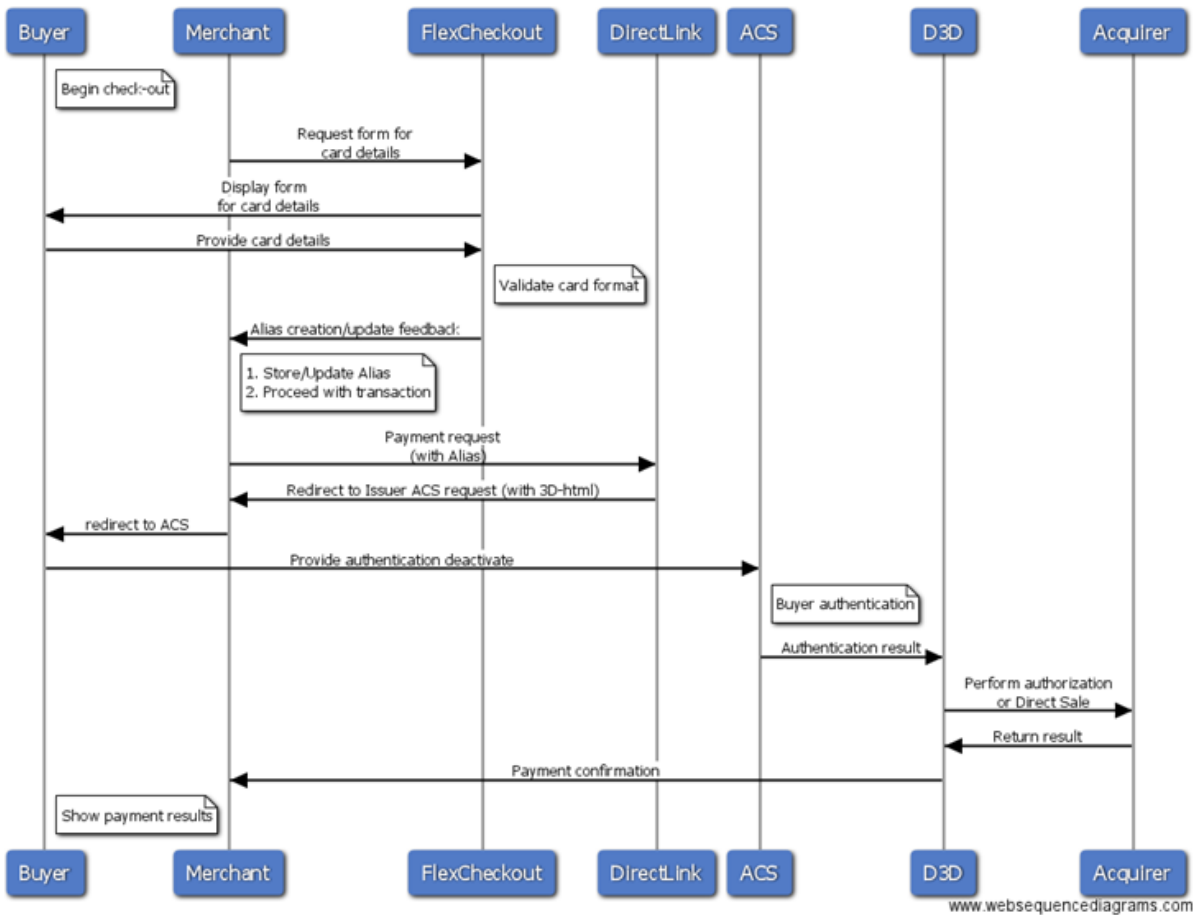
- One Page Checkout (Options-ID: OPC)
- Alias Manager (Options-ID: REC1, REC2, RECX)

In der Regel müssen Sie auch DirectLink (neue Zahlungen) aktivieren, wenn es nicht schon standardmäßig in Ihrem Abonnement aktiviert wurde.

2. Typischer Datenfluss

Sie können den FlexCheckout zu einem iframe weiterleiten (bzw. darin verkapseln), wo der Karteninhaber die Kreditkartendaten in unser System eingibt. Die Kreditkartendaten werden zusammen mit dem CVC sicher gespeichert, jedoch nur für eine begrenzte Zeit. Um die Kreditkartendaten permanent zu speichern, können Sie den Alias als einen „Permanent“-Alias kennzeichnen. Allerdings können CVCs gemäß den PCI Security Council-Standards nur maximal zwei Stunden lang gespeichert werden.

Sie können über DirectLink auch den eigentlichen Auftrag mit dem generierten Alias senden, ohne dabei die Kreditdatendaten zu senden.



Hinweis: Beim ersten Schritt läuft kein kartenbezogener Vorgang ab. Unser System führt nur eine grundlegende Prüfung des Datenformats durch, kann aber nicht garantieren, dass die Karte noch gültig ist oder einen ausreichenden Verfügungsrahmen zur Fortsetzung des Zahlungsvorgangs aufweist.

2.1 DirectLink-Integration

Bei Verwendung des mittels der FlexCheckout erzeugten Alias/Tokens können Sie dann eine DirectLink-Transaktion mit Ihrer normalen DirectLink-Implementierung einreichen. Weitere Informationen über die Anweisungen für die Implementierung finden Sie unter [DirectLink](#).

Diese Einrichtung ist auch kompatibel mit [DirectLink 3-D](#). Weitere Informationen über die Alias-Nutzung finden Sie unter [Alias Manager](#).

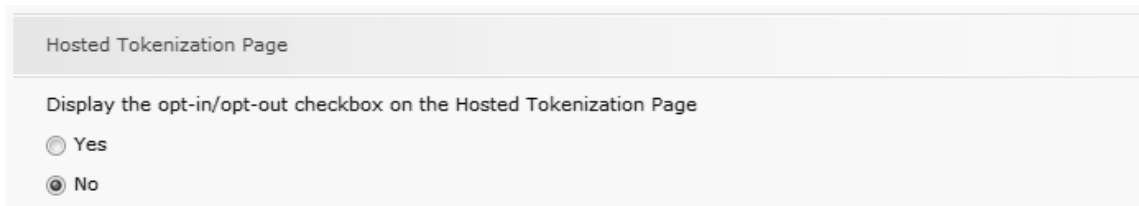
2.2 Verwendung mit Alias Manager

Wenn Sie die Option Alias Manager nutzen, können Sie ein Kontrollkästchen auf der FlexCheckout verfügbar machen, mit dem der Kunde

FlexCheckout

entscheiden kann, ob seine Finanzdaten unbegrenzt gespeichert werden sollen.

Hierzu müssen Sie auf der Alias Manager Konfigurationsseite Ihres Ingenico ePayments Kontos die Option zur Anzeige des Kontrollkästchens aktivieren (in der Voreinstellung lautet die Einstellung „Nein“).



Hosted Tokenization Page

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

Yes

No

> Wir empfehlen, diese Konfiguration bei einer One-Page-Checkout-Einrichtung zu verwenden.

Wenn Sie einen mehrschrittigen Bezahlprozess integriert haben, können Sie alternativ den [Parameter "StorePermanently"](#) in Ihrer Integration verwenden, um selbst zu kontrollieren, ob das Kontrollkästchen angezeigt wird.

3. FlexCheckout als Tokenisierung integrieren

können FlexCheckout auf zwei Arten nutzen:

- Indem Sie Ihren Kunden an FlexCheckout weiterleiten
- Indem Sie FlexCheckout in einem iframe verkapseln, sodass die Kunden auf Ihrer Webseite bleiben und eine vollständige One-Page-Checkout-Erfahrung machen.

Warnung: Wir können nicht empfehlen, FlexCheckout als reine App-Lösung zu integrieren, da einige Funktionen bei bestimmten Smartphones der mittleren Preisklasse nicht funktionieren würden, sondern raten eher zu einer mobilen Website-Lösung, die die Browser-Fähigkeiten umfassend nutzt.

Wenn der Kunde zu FlexCheckout weitergeleitet wird, muss er seine Kreditkartendaten eingeben und sie zur Tokenisierung auf FlexCheckout senden. Auf diese Weise laufen die Kreditkartendaten nie durch Ihren eigenen Webserver. Die URLs für den Zugriff auf FlexCheckout sind:

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

Die FlexCheckout funktioniert standardmäßig mit UTF-8-Zeichenkodierung. Wenn Sie mit ISO arbeiten, stellen Sie bitte die entsprechende "Zeichenkodierung"-s-Anpassung in Ihrem Back Office via Konfiguration > Technische Informationen > Globale Sicherheitsparameter > Hash-Methode.

Wenn ISO-Codierung zum Berechnen von SHA verwendet wird, müssen Händler das im Back-Office einstellen. Diese Einstellung überprüft den eingehenden SHA und den ausgehenden SHA. Sie wird auch zur Einstellung des URL-Codierungsschemas für die zurückgegebenen Parameter verwendet.

3.1 Input-Felder

Feld	Beschreibung	Format	Pflichtfeld
ACCOUNT.PSPID	Händler-ID	AN, 30	Ja
ALIAS.ALIASID	Alias des Kunden	AN, 50	Nein
ALIAS.ORDERID	Auftrags-ID	AN, 40	Nein
ALIAS.STOREPERMANENTLY	Geben Sie an, ob Sie ein Alias nach der Verwendung vorübergehend oder unbegrenzt speichern möchten. Mögliche Werte sind: <ul style="list-style-type: none"> • "N": das Alias wird nach zwei Stunden gelöscht. • "Y": das Alias wird für künftige Verwendung unbegrenzt gespeichert. Bitte beachten Sie die Infox-Box unterhalb dieser Tabelle für mehr Informationen.	Y / N	Nein
CARD.BIC	Bank Identification Code, nur für Lastschriften verwendet	AN, 8	

Feld	Beschreibung	Format	Pflichtfeld
CARD.BIN	Kreditkartenart Zahlungsmethoden	AN, 6	
CARD.BRAND	Geben Sie, welche Art von Formular angezeigt werden muss. > Notwendig, wenn kein Zahlungsverfahren angegeben wird.	AN, 25	Ja/Nein
CARD.PAYMENTMETHOD	Kreditkarte oder andere unterstützte Lastschriftmethoden - Geben Sie an, welche Art von Formular angezeigt werden muss. > Notwendig, wenn keine Marke angegeben wird.	AN, 26	Ja/Nein
LAYOUT.LANGUAGE	Auf der Seite zu verwendende Sprache („en_EN“, „nl_BE“, „fr_FR“ usw.).	AN, 5	Nein
LAYOUT.TEMPLATENAME	Vorlage-Eingabeparameter. Händler brauchen nur den Namen der Vorlage und der Dateierweiterung. B. „new_user.html“.	AN, 255	Nein
PARAMETERS.ACCEPTURL	URL für Umleitung im Erfolgsfall.	AN, 255	Ja
PARAMETERS.EXCEPTIONURL	URL für Umleitung im Fehlerfall.	AN, 255	Ja
PARAMETERS.EXCLUDEDPAYMENTMETHODS	Liste der Zahlungsmethoden bzw. Kreditkartenmarken, die NICHT gezeigt werden sollen. Getrennt durch ein Semikolon (;).	AN, 50	Nein
PARAMETERS.PARAMPLUS	Durchleitungsfeld: Weitere Parameter, die vom Händler gesendet und im Output wieder aufgefunden werden sollen.	AN, 1000	Nein
SHASIGNATURE.SHASIGN	Mit dem Secure Hash Algorithmus gehashte Zeichenfolge.	AN, 128	Ja

Alias.StorePermanently: Wichtige Hinweise

- Der Parameter Alias.StorePermanently sollte nur in Kombination mit dem Alias Manager verwendet werden. In der Voreinstellung wird das Alias unbegrenzt gespeichert.
- Wenn Sie eine One-Page-Checkout-Integration nutzen, empfehlen wir Ihnen dringend, die Option "opt-in/out checkbox" in Ihrer Alias Manager-Konfiguration zu aktivieren, damit das Kontrollkästchen immer auf der FlexCheckout erscheint.
- Sobald Sie die "opt-in/out checkbox" in Ihrer Alias Manager Konfiguration aktiviert haben, wird der Parameter Alias.StorePermanently überschrieben.

3.1.1 Prüfung auf doppelte OrderID

Um zu verhindern, dass Hacker Kartendaten, die mit einem bestimmten Token verknüpft sind, austauschen (indem sie den Link zur Auslösung der Anfrage erfassen und die Kartendaten durch echte Kartendaten einer echten Karte ersetzen), überprüfen wir die OrderID

(ALIAS.ORDERID), die Sie in der Anfrage gesendet haben.

Wenn wir feststellen, dass die OrderID bereits verwendet wurde, um einen Token zu erstellen, wird die Alias-Aktualisierung abgelehnt.

Wichtig

Die Überprüfung der OrderID funktioniert selbstverständlich nur, wenn die **OrderID (ALIAS.ORDERID)** geschickt wurde. Allerdings ist die OrderID allein nicht genug; die **AliasID (ALIAS.ALIASID) ist ebenfalls erforderlich**. Wird keine AliasID gesendet, wird die Prüfung nicht durchgeführt und ein neuer Alias wird angelegt.

Wenn eine doppelte OrderID erkannt wird, wird die Anfrage blockiert und dem Kunden wird eine allgemeine Fehlermeldung angezeigt:

„Beim Bearbeiten Ihrer Anfrage ist ein Fehler aufgetreten. Bitte versuchen Sie es später noch einmal oder kontaktieren Sie Ihren Händler.“

Fehler, die von einer doppelten OrderID erzeugt werden, können erfasst werden, wenn der Debug-Modus in den Fehlerprotokollen aktiviert wurde (bitte wenden Sie sich an unsere Kundendienst).

3.2 SHA-Signatur für Input

Damit wir die Integrität der Daten prüfen können, muss allen Anfragen eine SHA-Signatur beigefügt werden, [auf dieselbe Weise wie auch für e-Commerce-Transaktionen](#).

Unser System verwendet den SHA-Algorithmus, wie im Register „Global Security Parameters“ [Globale Sicherheitsparameter] Ihrer Seite Technische Information definiert. Es besteht weiterhin die Möglichkeit, diesen Algorithmus wieder auf SHA-1 oder SHA-256 umzustellen.

Beispiel

Felder (in alphabetischer Folge):

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

Geheime Passphase (wie in der Technischen Information definiert): Mysecretsig1875!?

Zeichenfolge für Hashing:

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

Resultierende SHA-Signatur (SHA-512):

563DC909F70BA5DDD470D69C1B390E7D1C1C47705AC5801B27038446D7033B5787728EA
754EF72E7FA2436FC5962E34E20DF64E7F9139893A33653F118816818

3.3 Output fields

Die folgenden Felder können an Sie zurückgegeben werden, um den Status der Alias-Erstellung/Aktualisierung zu erhalten. Sollen sie im Feedback enthalten sein, müssen sie in den dynamischen Feedback-Parametern (Ingenico ePayments account: Konfiguration > Technische Informationen > Transaktions-Feedback > Alias gateway und Tokenization: Dynamische parameter) entsprechend konfiguriert werden.

Der SHASIGN-Wert ist nicht optional und wird daher immer zurückgegeben.

Feld	Beschreibung	Max. Länge
ALIAS.ALIASID	Alias, wie vom Händler gesendet oder vom PSP generiert (gemäß 32-stelligem GUID-Format). Beispiel: 34F5302C-85D7-4F35-BDF5-103CCEC2FB61	50
ALIAS.NCERROR	Fehlercode	50
ALIAS.NCERRORCARDNO	Fehlercode für CARDNO	50
ALIAS.NCERRORCN	Fehlercode für CN	50
ALIAS.NCERRORCVC	Fehlercode für CVC	50
ALIAS.NCERRORED	Fehlercode für ED	50
ALIAS.ORDERID	Eindeutige Identifikation des Auftrags. (vom Händler gesendet oder vom System generiert)	40
ALIAS.STATUS	Ergebnis der Alias-Erzeugung: <ul style="list-style-type: none"> • 0=OK • 1=NOK • 2=Alias aktualisiert • 3=Vom Benutzer abgebrochen 	1
ALIAS.STOREPERMANENTLY	Gibt an, ob das Alias vorübergehend (N) oder unbegrenzt (Y) gespeichert werden soll. Wenn eine Transaktion mit diesem Alias ausgeführt wird, müssen Händler diesen Parameter im Feld ALIASPERSISTEDAFTERUSE zurücksenden. Andernfalls wird er dauerhaft gespeichert.	1 (Y/N)
CARD.BIC	Bank Identification Code (BIC), nur für Lastschriften verwendet	8
CARD.BIN	Zahlungsmethoden vom Kreditkartentyp	6
CARD.BRAND	Marke des Zahlungsverfahrens	25
CARD.CARDHOLDERNAME	Name des Karteninhabers	50
CARD.CARDNUMBER	Karte, wobei sensible Daten durch „X“ ersetzt sind. Beispiel: XXXXXXXXXXXXX1111 Hinweis: Im Falle eines Fehlers wird auch die Karte durch Ersatzzeichen unkenntlich gemacht.	35

Feld	Beschreibung	Max. Länge
CARD.CVC	Kartenprüfnummer für Kreditkarten, wobei sensible Daten durch „X“ ersetzt sind. Beispiel: XXX	6
CARD.EXPIRYDATE	Ablaufdatum, z. B. 0220 (Februar 2020)	4
PARAMPLUS	Durchleitungsfeld: Im Input bereitgestellte Daten	/
SHASIGN	SHA-Signature für Output	128

Die ausgewählten Parameter werden an die Rückgabe-URL angehängt, die in Ihrer Anfrage (PARAMETERS.ACCEPTURL oder PARAMETERS.EXCEPTIONURL) definiert ist, um eine Rückmeldung über das Funktionieren zu geben.

3.4 SHA-Signatur für Output

Unser System liefert eine SHA-OUT-Signatur für die folgenden Parameter, [wie bei e-Commerce-Transaktionen](#) zurück:

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. Anpassung

Dieses Kapitel stellt zwei Arten der Anpassung vor: grundlegende und erweiterte Anpassung. Unter Grundlegender Anpassung erfahren Sie, wie Sie Ihre eigenen CSS anwenden und die Textfarben, Hintergrund- und Wrapper-Farben usw. anpassen. Das Folgende ist eine Standard-Zahlungsseite unter Verwendung der CSS-Standarddateien:

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.

Bei der erweiterten Anpassung erfahren Sie mehr über Master-/Partielle Vorlagen, die Verwendung von Platzhaltern und die Anpassung von Textinhalt mit XML-Dateien. Je nach Ihren Bedürfnissen werden partielle Vorlagen separat erstellt und angepasst.

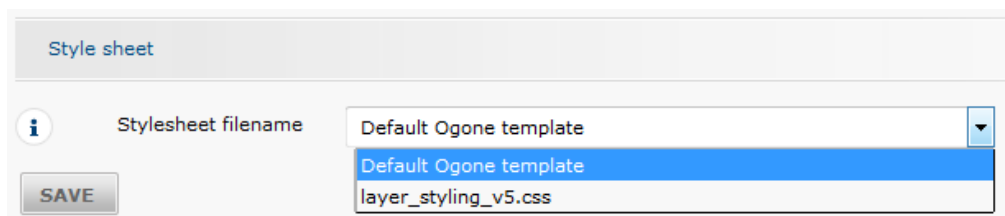
4.1 Grundlegende Anpassung

Zu einer grundlegenden Anpassung gehört die Konfiguration des Standard-Stylesheets unter der Ingenico ePayments-Standardvorlage

4.1.1 Ingenico ePayments-Standardvorlage

Die CSS-Datei kann entweder mit einem standardmäßigen Ingenico ePayments-Stil oder einer hochgeladenen CSS-Datei im [Manager für Vorlagendateien](#) angepasst werden.

Für mehr Informationen über die Anpassung Ihrer eigenen CSS-Datei, s. Verwendung von [Stylesheets](#).



4.1.2 Verwendung von Stylesheets

Das FlexCheckout-Layout können Sie mit einer CSS-Datei anpassen. Dazu müssen Sie die CSS-Datei zunächst in den Template File Manager hochladen und in Ihrem Ingenico ePayments Konto einen Link zu der Datei setzen. Die CSS-Datei besteht aus einem Formular/Daten-Container mit Elementen (wie zum Beispiel: Ablaufdatum, CVC, Informationsfeld etc.), die als CSS-Standarddatei eingestellt werden muss. Sie können jedes Element, das die Formatierung für jedes Feld definiert, mit dem Formular/Daten-Container verlinken.

Die Formatierungsregel in jedem Element können Sie an den Bedarf Ihres Unternehmens anpassen. Klicken Sie hier für eine Liste der [CSS-Elemente](#).

Nachfolgend finden Sie Beispiele für Formular/Daten-Container und Elemente:

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

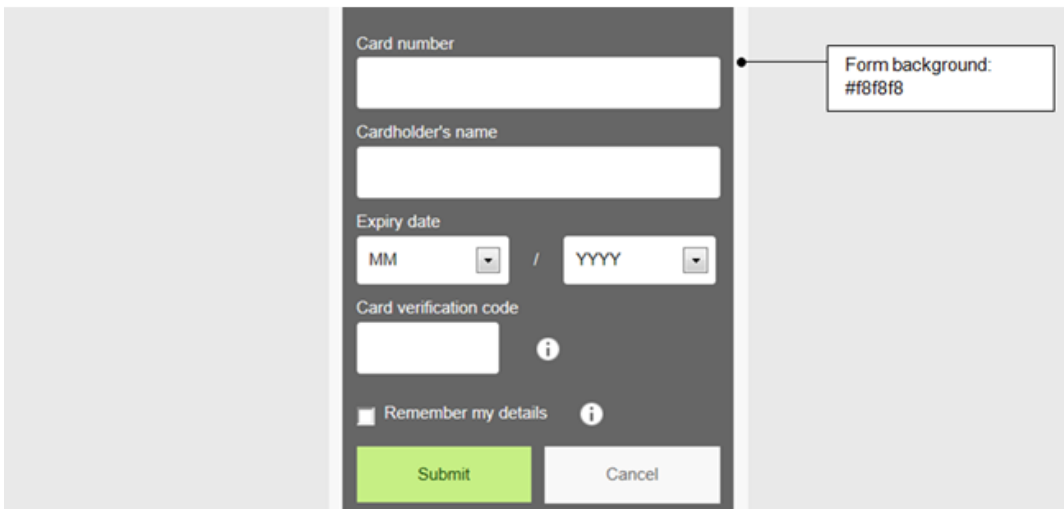
#payment-data-container .payment-input select { width:109px; height:35px; color:#000; font-size:12px; margin-
bottom:10px; padding:7px 5px; -webkit-border-radius: 3px;-moz-border-radius: 3px; border-radius: 3px;
border:1px solid #000;}

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

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

Sie können die Standardbenutzeroberfläche (wie unten angezeigt) ändern, indem Sie Ihr eigenes CSS anwenden.

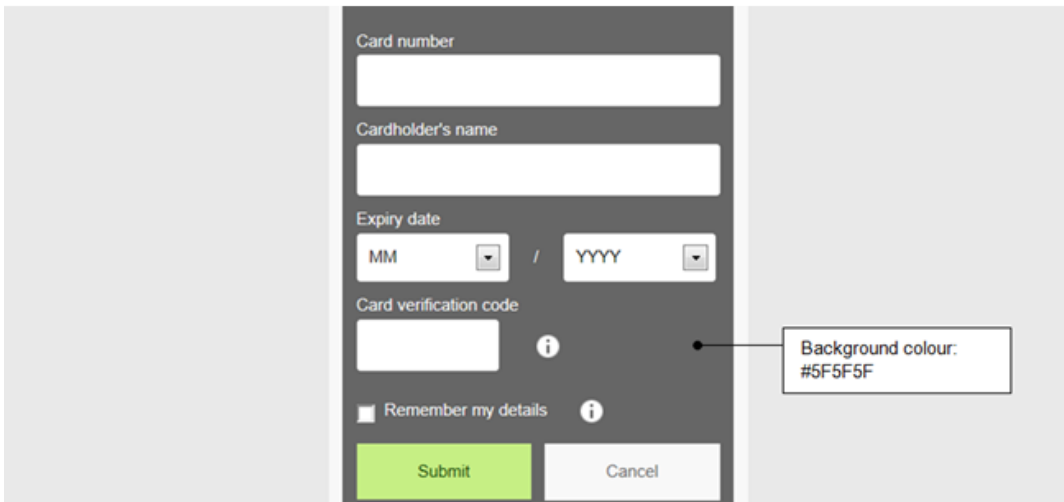
Form Hintergrundfarbe



Impacted CSS:

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

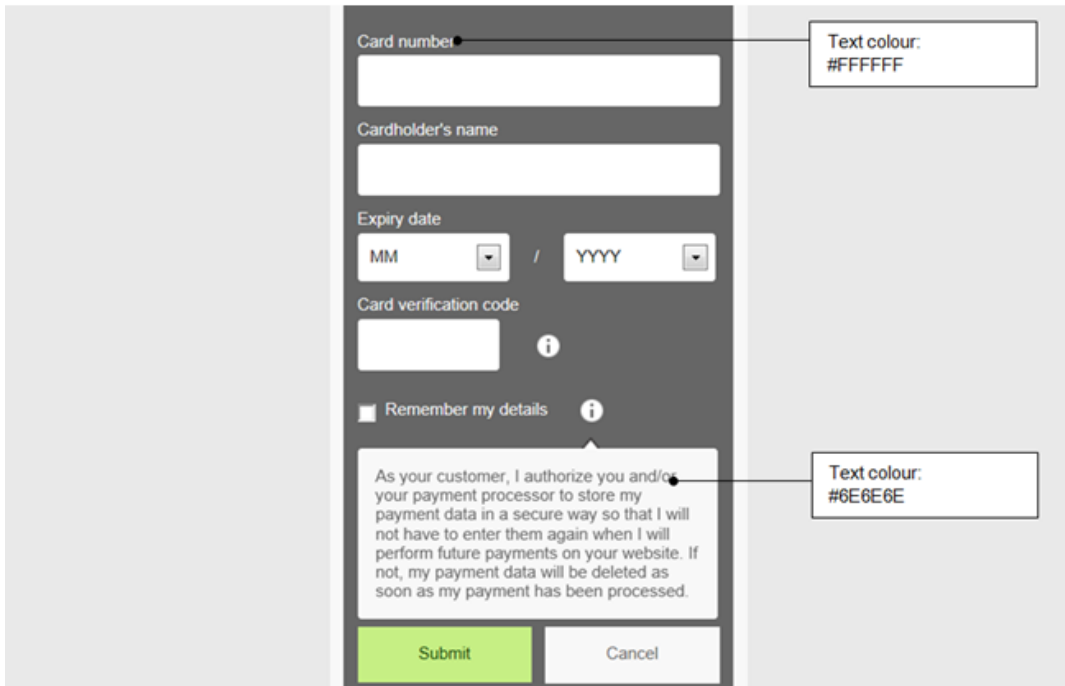
Form Umgebungsfarbe



Impacted CSS:

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

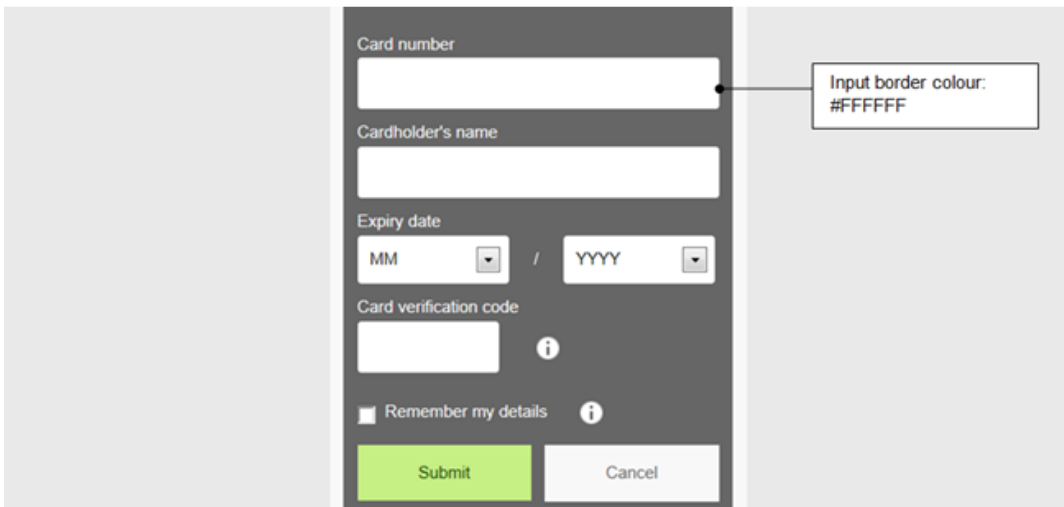
Textfarbe



Impacted CSS:

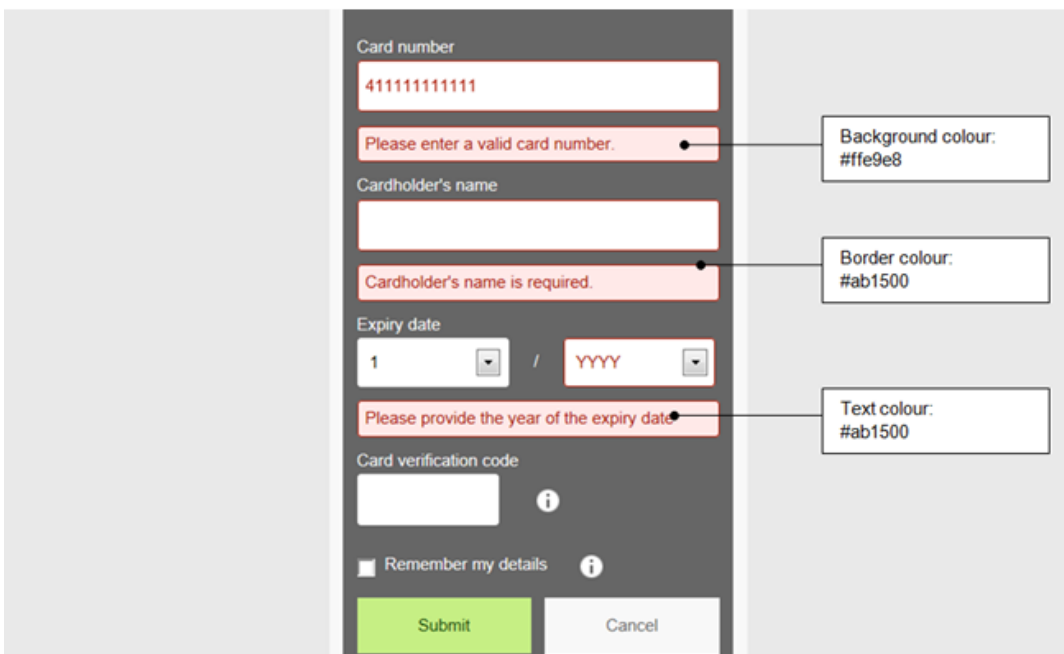
```
#payment-data-container .payment-label{color:#FFFFFF;} #payment-data-container{color:#6E6E6E;}
```

Eingaberahmen Farbe



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

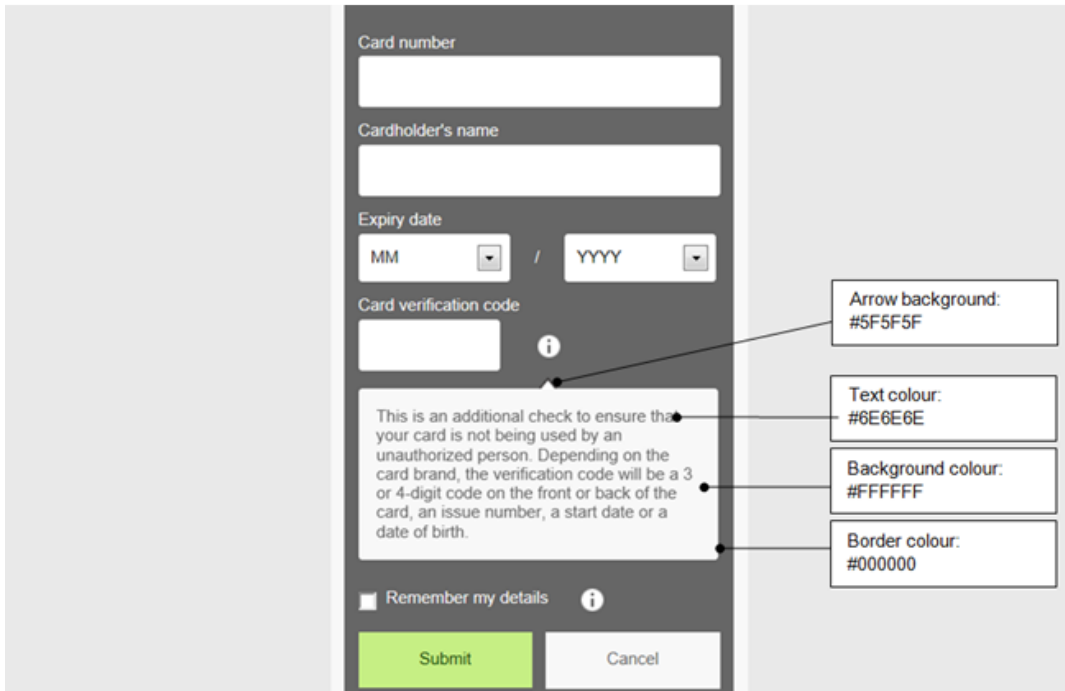
Validierung-Meldungsfenster



Impacted CSS:

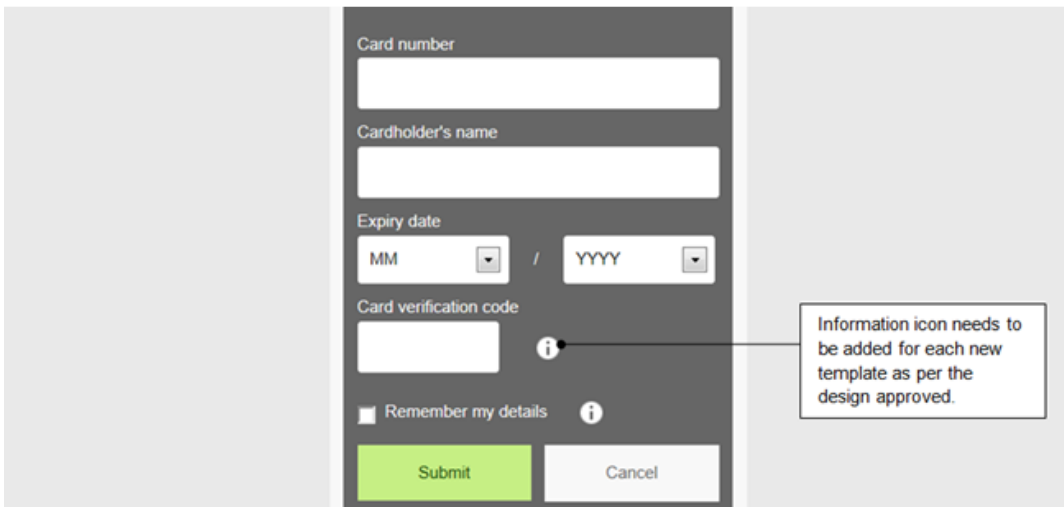
```
.payment-error { color:#ab1500; background:#ffe9e8; border:1px solid #ab1500;} #payment-data-container .payment-input-error input { color:#ab1500; border:1px solid #ab1500;} #payment-data-container .payment-input-error select { color:#ab1500; border:1px solid #ab1500;}
```

Hilfeinformationsfenster



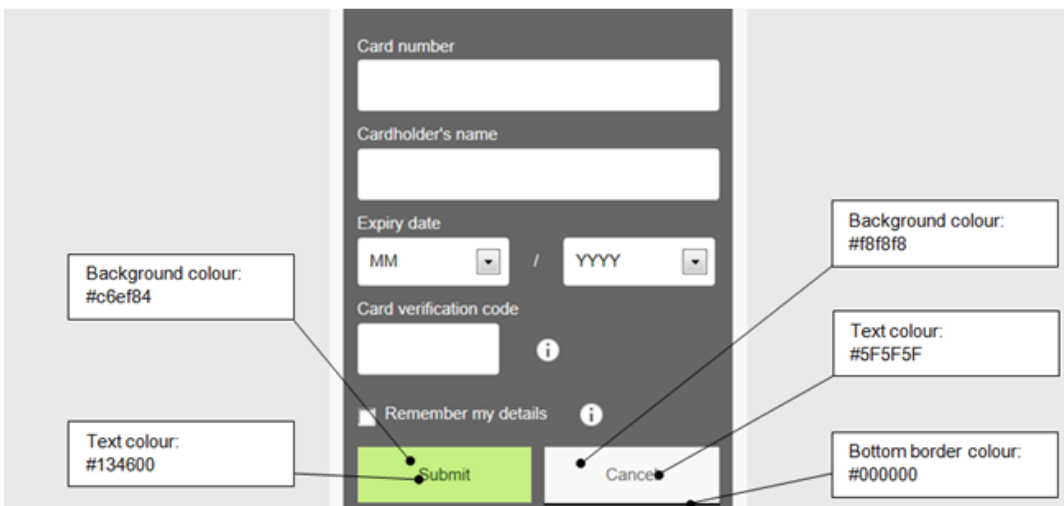
```
Impacted CSS: #payment-cvc-info-container span.arowlmg {background: #5F5F5F url(arrow-top.png) 0 0 no-repeat;} #payment-cvc-info-container p.cvc-info-txt{color: #6E6E6E;} #payment-cvc-info-container div.help-box {border: 1px solid #000000;background: #FFFFFF;}
```

Informationssymbol



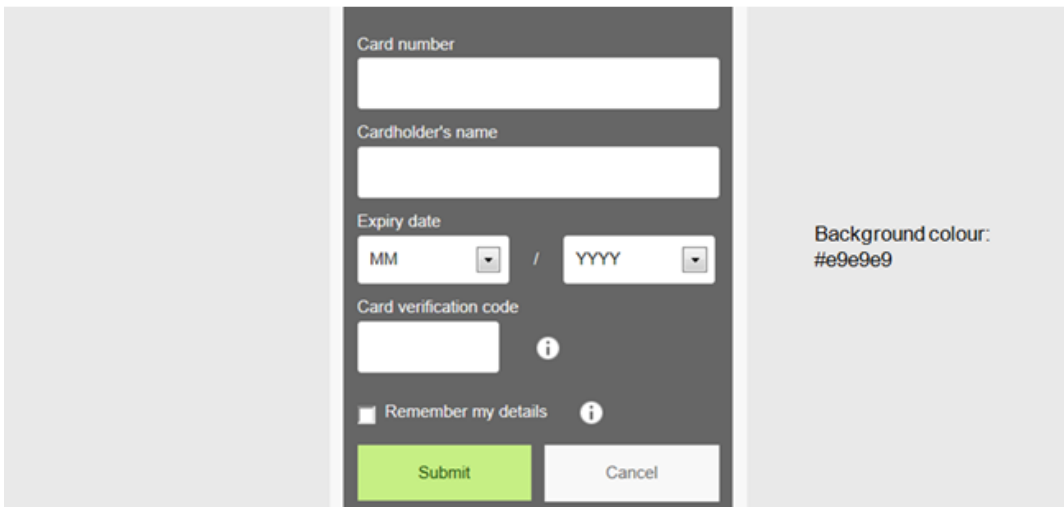
Impacted CSS: .payment-method-info {background:url(payment_info.png) 4px 1px no-repeat;}

Schaltfläche Senden/Abbrechen



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

Hintergrundfarbe



Impacted CSS:< #payment-container {background:# e9e9e9}

Beispiel-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: #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 Erweiterte Anpassung

Um mehr Flexibilität zu ermöglichen, können Sie jetzt Master-Vorlagen und partielle Vorlagen erstellen und anpassen. Je nach Ihren Bedürfnissen werden partielle Vorlagen separat erstellt und angepasst. Sie können außerdem den Textinhalt mit XML-Dateien anpassen.

Sie können wahlweise ein Tag `$$$PAYMENT_ZONE$$$` verwenden, das eine einfache Vorlage aufruft, oder Ihre eigene partielle Vorlage gestalten, [indem Sie Platzhalter verwenden](#).

4.2.1 Verwenden von Platzhaltern

Händler, die sich dafür entscheiden, ihre eigene Master-Vorlage (master.html) und partiellen Vorlagen anzupassen und zu formatieren, können Platzhalter direkt in master.html einfügen.

Auch das Aussehen der Zahlungsseite können Sie mit einer eigenen CSS-Datei (**newuser.css**) festlegen, die mit dem master.html verlinkt werden muss: Wenn Sie sich zur Anpassung Ihrer CSS-Datei entscheiden, muss die modifizierte CSS-Datei in den Template File Manager hochgeladen und als Standard eingestellt werden.

Platzhalter werden in drei Gruppen eingeteilt: Haupt-, Element- und gruppierte Platzhalter:

Wichtige Platzhalter

Das Folgende ist eine Übersicht der wichtigsten Platzhalter:

- **\$\$\$PAYMENT_ZONE\$\$\$**: Stellt eine angeforderte Marke oder Zahlungsmethode entweder mit einer angepassten Vorlage (mit einer ggf. vorhandenen partiellen Vorlage) oder der Standardvorlage dar.
- **\$\$\$TP_RESOURCES_URL\$\$\$**: Die URL in dem Verzeichnis bereitstellen, wo sich die Dateien im Hosted File Service befindet.

Außerdem haben Sie die Möglichkeit, benutzerdefinierte Skripts zu verwenden, die mit einer benutzerdefinierten JavaScript-Datei verknüpft werden müssen. Auf diese JavaScript-Datei kann dann verwiesen werden, nachdem sie in den Dateimanager hochgeladen wurde. Zum Beispiel: Die referenzierte JavaScript-Datei in `<script type="text/javascript" src="$$$TP_RESOURCES_URL$$$/existinguser.js"></script>` ist **existinguser.js**.

Um JavaScript-Bibliotheken (z.B. JQuery) in den Vorlagen zu verwenden, müssen Sie diese Dateien in den Template File Manager hochladen. Nachdem sie hochgeladen wurden, kann mit **\$\$\$TP_RESOURCE_URL\$\$\$** auf die Skripts verwiesen werden. Dieser Schritt muss für alle Dateien ausgeführt werden, auf die in der Vorlage verwiesen wird: CSS, Bilder, Schriftarten etc. Das garantiert, dass das ausgeführte Skript die betreffende JavaScript-Datei ist.

Hinweis: Da mit einem nicht PCI-konformen Host einer Drittpartei nicht alle Daten per Push/Pull ausgetauscht werden können, hat die Verwendung von JavaScript einen Einfluss darauf, ob der Händler PCI SAQ A nutzen kann.

- **\$\$\$DISABLE_MASKING\$\$\$**: Deaktiviert die Maskierung eines Eingabefelds.
- **\$\$\$SUBMIT\$\$\$**: Zeigt eine Senden-Schaltfläche für das Formular an. Zum Beispiel:

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

- **\$\$\$CANCEL\$\$\$**: Zeigt eine „Abbrechen“-Schaltfläche für das Formular an. Zum Beispiel :

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

Eine Master-Vorlage, der **\$\$\$PAYMENT_ZONE\$\$\$** enthält, braucht keine Platzhalter für Senden und Abbrechen zu enthalten. Platzhalter für Senden und Abbrechen müssen jedoch eingefügt werden, wenn eine partielle Vorlage pro Zahlungsmethode erstellt wird oder wenn eine Master-Vorlage ohne **\$\$\$PAYMENT_ZONE\$\$\$**-Platzhalter erstellt wird.

Element Platzhalter

Das eigentliche Steuerungs-`<element>` wird in einem Element-Container dargestellt: `<div id="payment-<input_name>-<element>-container" class="payment-<element>">`. Für einfaches CSS-Styling gilt dieselbe Namenskonvention für ID und Klasse.

Element-Platzhalter stellen HTML-Kontrollelemente für verschiedene Eingabeelemente dar und müssen dieses Format haben:

- **\$\$\$<input_name><element>\$\$\$**: als Eingabekontrollelement für die Kartennummer.

- \$\$\$CARD NUMBER.INPUT\$\$\$ - CARD NUMBER: KARTENNUMMER als `<input_name>`, wobei INPUT das HTML-<element> ist.
- \$\$\$<input_name>LABEL\$\$\$: zeigt eine Beschriftung für das Element `<input_name>` an.

Zum Beispiel wir der Platzhalter \$\$\$CARD NUMBER LABEL\$\$\$ so dargestellt:

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

Für alle Labels wird ein div als Container dargestellt:

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

Die Klasse für einen Label-Container ist immer "payment-label", sodass die Labels zusammen formatiert werden können. Eine Beschriftung für ein Label-Steuer-element wird immer angezeigt:

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

Zum Formatieren eines Label-Kontrollelementes muss die CSS-Datei auf die Container-ID und nicht auf die Kontrollelement-ID angewendet werden.

\$\$\$<input_name> INPUT\$\$\$ zeigt das Eingabe-Kontrollelement für `<input_name>` an:

```
<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>
```

Zum Formatieren eines Eingabekontrollelementes muss die CSS-Datei auf die Container-ID und nicht auf die Kontrollelement-ID angewendet werden. \$\$\$<input_name> INFO\$\$\$ zeigt einen Link zu einer Beschreibung des Felds `<input_name>` an:

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

Bindet Informationen für dieses Feld in der ausgewählten Sprache. \$\$\$<input_name> ERROR\$\$\$ zeigt einen Container für Fehlermeldungen von <input_name> an:

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

Validierungsmittelungen können hier eingefügt werden.

Gruppierte Platzhalter

Die Element-Platzhalter können als Gruppe mit Hilfe von gruppierten Platzhaltern eingefügt werden. Das Folgende ist ein dargestellter Container mit einem gruppierten Platzhalter: <div id="payment-<input_name>-container"></div>. Darin fügen wir die Element-Platzhalter in dieser Gruppe ein. Effektiv stellen wir dies dar:

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

Eine Master-Vorlage kann Platzhalter in <div id="paymentForm"> enthalten. Die Platzhalter (wie \$\$\$CARD NUMBER\$\$\$, \$\$\$CARDHOLDER NAME\$\$\$, \$\$\$EXPIRY DATE\$\$\$ und dergleichen) definieren den Inhalt des Kreditkartenformulars. Händler können die Platzhalter auswählen, die ihren Anforderungen am besten entsprechen.

Das Folgende ist ein Beispiel für eine Master-Vorlage mit Platzhaltern für eine Kreditkarte für mit dem 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>

```

Zum Erstellen eines Lastschrift-Formulars können Händler Lastschrift-Platzhalter wie den Folgenden verwenden:

```

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

```

Der Bankkontoname, die IBAN-Nummer und der BIC-Code werden als Lastschriftformular dargestellt.

Wenn Händler eine Vorlage wünschen, die mehrere Zahlungsmethoden unterstützt, können sie eine Master-Vorlage und eine partielle Vorlage für die Zahlungsmethoden erstellen. Eine Master-Vorlage ist wie folgt aufgebaut:

```

<!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>

```

Eine partielle Vorlage ist wie folgt mit Platzhaltern aufgebaut (\$\$\$CARD NUMBER\$\$\$, \$\$\$CARDHOLDER NAME\$\$\$, \$\$\$EXPIRY DATE\$\$\$, and so on):

```

<div id="paymentForm">
  $$$CARD_NUMBER$$$
  $$$CARDHOLDER_NAME$$$

```

```

$$$EXPIRY DATE$$$
$$$CVC$$$
$$$SUBMIT$$$
$$$CANCEL$$$
</div>

```

Wenn Händler nicht das Standard-Zahlungsformular mit `$$$PAYMENT ZONE$$$` anzeigen möchten, kann stattdessen eine partielle Vorlage eingefügt werden. Zum Beispiel: Wenn die gewünschte Zahlungsmethode eine Kreditkarte ist, wird eine partielle Vorlage mit dem Namen **master.creditcard.htm** verwendet. Wenn es eine Lastschrift ist, wird stattdessen eine partielle Vorlage mit dem Namen **master.directdebit.htm** verwendet.

Händler müssen diese partielle Vorlage in den Dateimanager hochladen. Eine Namenskonvention muss strikt eingehalten werden. Zum Beispiel: Wenn die vorhandene Master-Vorlage den Namen **master.html** hat und Händler Lastschrift als Zahlungsmethode senden möchten, muss eine partielle Vorlage mit dem Namen **master_directdebit.html** zum Dateimanager hochgeladen werden.

4.2.2 Anpassen von Textinhalt mit XML

Sie können mithilfe einer XML-Datei auch das Erscheinungsbild des Textes auf den Tokenisierungsseiten anpassen. Wird keine XML-Datei verwendet, wird das Tokenisierungsformular verwendet. Nach der Anpassung der XML-Datei müssen Sie diese zum Manager für Vorlagendateien hochladen und den Dateinamen der Master-Vorlage verwenden. Beachten Sie, dass zuvor unbedingt eine Master-Vorlage hochgeladen werden sollte.

Die XML-Datei muss auch Elemente zum Identifizieren der Felder enthalten. Bitte klicken Sie hier für eine vollständige [Liste der Elemente](#). Beispiel eines Elements (CardHolderNameInput) zur Namenseingabe des Karteninhabers:

```

<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 diesem Element werden Sprachen US-Englisch und Französisch in der Sprach-ID angegeben. Der ID-Tag legt auch den ISO-Kulturcode fest.

Sind keine Elemente angegeben, wird der Standardtext für dieses Element verwendet. Ist keine Sprache für ein Element angegeben, wird die Standardsprache verwendet.

Hinweis:

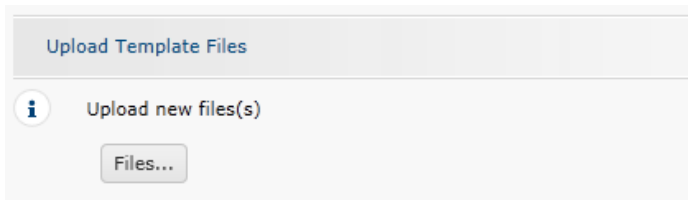
Die erste Zeile der XML-Datei muss immer mit <root> beginnen. Die letzte Zeile der XML-Datei muss immer mit </root> enden.

4.3 Vorlagen-Dateimanager

Mit dem Vorlagen-Dateimanager können Sie Ihre Vorlagen und die unterschiedlichen damit in Zusammenhang stehenden Dateien einfach verwalten.

Um den Dateimanager zu verwenden, melden Sie sich bei Ihrem Ingenico ePayments-Konto an und gehen Sie zu „Konfiguration“ > „Vorlage“ > „Dateimanager“.

4.3.1 Vorlagendateien hochladen



Unter „Vorlagendateien hochladen“ wählen Sie die Schaltfläche „Dateien...“ aus, um nach den Dateien zu suchen, die Sie hochladen möchten. Sie können CSS- und Bilddateien (.css, .jpg, .jpeg, .gif, .png) mit einer Größe von höchstens 7 MB pro Datei bzw. 10 MB insgesamt hochladen.

Treffen Sie Ihre Auswahl, und bestätigen Sie diese anschließend.

4.3.2 Hochgeladene Dateien prüfen und verwalten

Sobald der Upload abgeschlossen ist, sehen Sie Ihre hochgeladenen Dateien auf der gleichen Seite im Abschnitt „Hochgeladene Dateien“.

Die Dateien werden zunächst den Status „Wird geprüft“ aufweisen, in dem die erforderlichen Sicherheits-/Virenprüfungen durchgeführt werden. Dieser Vorgang kann mehrere Minuten in Anspruch nehmen.

Sie können die Dateien verwenden, sobald sich der Status zu „Bestätigt“ geändert hat.

Klicken Sie auf die Schaltfläche „Aktualisieren“ , um den aktuellen Status der Dateien zu prüfen bzw. auf die Schaltfläche „Löschen“ , um die Datei dauerhaft zu löschen.

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-7fe40dfaf9e6_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	

Eine Datei wechselt in den Status „Abgelehnt“, wenn sie die Sicherheitsprüfung nicht bestanden hat. Das kann beispielsweise auf einen Virus oder eine falsche Dateierweiterung zurückzuführen sein.

4.3.3 Integration

Weisen Sie Ihren hochgeladenen Dateien in Ihrem Code einen Code in diesem Format zu: `$$$TP_RESOURCES_URL$$$/[Name Ihrer Datei]`.

5. Platzhalter

Das Folgende ist eine vollständige Liste der Platzhalter und ihrer Zwecke:

Wichtige Platzhalter	Beschreibung
\$\$\$PAYMENT ZONE\$\$\$	Eine benutzerdefinierte Vorlage aus einer partiellen Vorlage (falls vorhanden) oder einer Standardvorlage, der die gewünschte Marke oder Zahlungsmethode anzeigt.
\$\$\$TP RESOURCES URL\$\$\$	Ruft eine Datei auf, die zuvor in das Back-Office hochgeladen wurde.
\$\$\$DISABLE MASKING\$\$\$	Stellt ein Textfeld dar, das den Kartentypen und der Eingabeformatierung entspricht. Deaktiviert das Maskieren des Eingabefelds bei Erfassung des Kartentyps und die Formatierung von Eingabefeldern auf Basis der Kartenmarke.
\$\$\$SUBMIT\$\$\$	Eine Senden-Schaltfläche, die auf dem Formular vorhanden sein muss.
\$\$\$CANCEL\$\$\$	Eine „Abbrechen“-Schaltfläche im Formular.
\$\$\$CARD NUMBER\$\$\$	Eine Kreditkartennummer mit der Marke entsprechenden Prüfung und Maskierung.
\$\$\$CARD NUMBER LABEL\$\$\$	Ein Hinweis für eine Kreditkartennummer.
\$\$\$CARD NUMBER INPUT\$\$\$	Das Feld, in dem eine Kartennummer eingegeben wird.
\$\$\$CARD NUMBER INFO\$\$\$	Das Feld, in dem eine Kartennummer eingegeben wird.
\$\$\$CARD NUMBER ERROR\$\$\$	Eine Fehlermeldung, die angezeigt wird, wenn eine falsche Kartennummer eingegeben wird.
\$\$\$CARDHOLDER NAME\$\$\$	Ein Name eines Kreditkarteninhabers.
\$\$\$CARDHOLDER NAME LABEL\$\$\$	Ein Hinweis für den Namen eines Karteninhabers.
\$\$\$CARDHOLDER NAME INPUT\$\$\$	Das Feld für die Eingabe des Namens des Karteninhabers.
\$\$\$CARDHOLDER NAME INFO\$\$\$	Das Feld für die Eingabe des Namens des Karteninhabers.
\$\$\$CARDHOLDER NAME ERROR\$\$\$	Eine Fehlermeldung, die angezeigt wird, wenn der Name eines Karteninhabers falsch eingegeben wird.
\$\$\$EXPIRY DATE\$\$\$	Der Bereich, in dem ein Ablaufdatum angezeigt wird.
\$\$\$EXPIRY DATE LABEL\$\$\$	Ein Hinweis für ein Ablaufdatum.

\$\$\$EXPIRY DATE INPUT\$\$\$	Der Bereich, in dem ein Ablaufdatum eingegeben wird.
\$\$\$EXPIRY DATE MONTH\$\$\$	Der Monat des Ablaufdatums.
\$\$\$EXPIRY DATE YEAR\$\$\$	Das Jahr des Ablaufdatums.
\$\$\$EXPIRY DATE INFO\$\$\$	Das Feld, in dem ein Ablaufdatum eingegeben wird.
\$\$\$EXPIRY DATE ERROR\$\$\$	Eine Fehlermeldung, die angezeigt wird, wenn ein Ablaufdatum falsch eingegeben wird.
\$\$\$CVC\$\$\$	Das Feld, in dem ein CVC mit der entsprechenden Prüfung platziert wird.
\$\$\$CVC LABEL\$\$\$	Ein Hinweis für CVC.
\$\$\$CVC INPUT\$\$\$	Das Feld, in dem ein CVC eingegeben wird.
\$\$\$CVC INFO\$\$\$	Das Feld zur Anzeige der CVC-Informationen.
\$\$\$CVC ERROR\$\$\$	Eine Fehlermeldung, die angezeigt wird, wenn ein CVC falsch eingegeben wird.
\$\$\$BANK ACCOUNT\$\$\$	Eine Kontonummer für Lastschriften-Formulare.
\$\$\$ACCOUNT OWNER\$\$\$	Der Bereich, in dem ein Kontoinhaber angezeigt wird.
\$\$\$ACCOUNT OWNER LABEL\$\$\$"	Ein Hinweis für einen Kontoinhaber.
\$\$\$ACCOUNT OWNER INPUT\$\$\$	Das Feld, in dem ein Kontoinhaber eingegeben wird.
\$\$\$STORE PERMANENTLY\$\$\$	Ein Kontrollfeld für den Händler, mit dem er das dauerhafte Speichern des Alias aus- bzw. abwählen kann.
\$\$\$BIC\$\$\$	Ein BIC-Feld für Lastschriften-Aliase.

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

	<ul style="list-style-type: none"> • CardNumberTooLong • InvalidCardNumberCharacters • CardHolderName • CardHolderNameIsRequired • InvalidCardHolderName • CardHolderNameTooLong • ExpiryDate • InvalidExpiryDate • ExpiryMonthsRequired • InvalidExpiryMonth • InvalidExpiryMonthLength • InvalidExpiryMonthCharacters • ExpiryYearsRequired • InvalidExpiryYear • InvalidExpiryYearLength • InvalidExpiryYearCharacters • Cvc • CvcsRequired • InvalidCvcNumber • CvcTooLong • CvcContainsNonNumericCharacters
<p>CVC information</p>	<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
<p>Direct debits</p>	<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

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