

FlexCheckout

**ingenico**  
ePayments

## Tabella dei contenuti

### 1. Introduzione

### 2. Flusso tipico dei dati

#### 2.1 Integrazione DirectLink

#### 2.2 Utilizzo con Alias Manager

### 3. Integrazione di FlexCheckout come tokenizzazione

#### 3.1 Campi di input

##### 3.1.1 Controllo di OrderID duplicato

#### 3.2 Firma SHA per input

#### 3.3 Campi di output

#### 3.4 Firma SHA per output

### 4. Personalizzazione

#### 4.1 Personalizzazione di base

##### 4.1.1 Modello Ingenico predefinito

##### 4.1.2 Uso dei fogli di stile

#### 4.2 Personalizzazione avanzata

##### 4.2.1 Uso dei segnaposto

##### 4.2.2 Personalizzazione dei contenuti di testo con XML

#### 4.3 Template File Manager

##### 4.3.1 Caricamento di file modello

##### 4.3.2 Controllo e gestione dei file caricati

##### 4.3.3 Integrazione

### 5. Segnaposto

## 1. Introduzione

La pagina One Page Checkout con FlexCheckout consente di richiedere ai clienti di inserire i dati della carta in modo sicuro sul sistema PSP, tokenizzare questi dati e (volendo) riutilizzare il token/alias in una transazione futura:

1. FlexCheckout: la FlexCheckout viene reindirizzata a un iframe o inclusa al suo interno, quindi il titolare della carta inserisce i dati della carta nel nostro sistema, dove vengono memorizzati in modo sicuro (compreso il CVC, ma solo per un periodo limitato)
2. DirectLink: Si invia l'ordine effettivo con gli alias generati, senza dover inviare i dati della carta (vedera [integrazione DirectLink](#))

### Vantaggi

- Pieno controllo sull'aspetto della pagina di pagamento, grazie al meccanismo del modello flessibile, e se si include la FlexCheckout all'interno di un Iframe, si può controllare anche la sequenza di checkout.
- Pagina di pagamento sicura e conforme a PCI, senza dover salvare personalmente i dati della carta; tutti i dati sensibili vengono inseriti direttamente nella nostra pagina hosted.
- Perfetta integrazione per diversi scenari di checkout, tra cui One Page Checkout.
- Nella pagina di checkout finale, è possibile offrire up-selling e cross-selling.
- Sono supportate tutte le principali carte di credito (VISA, MasterCard, American Express, Diners).

### Opzioni obbligatorie

È necessario abilitare almeno una delle seguenti opzioni per usare FlexCheckout:

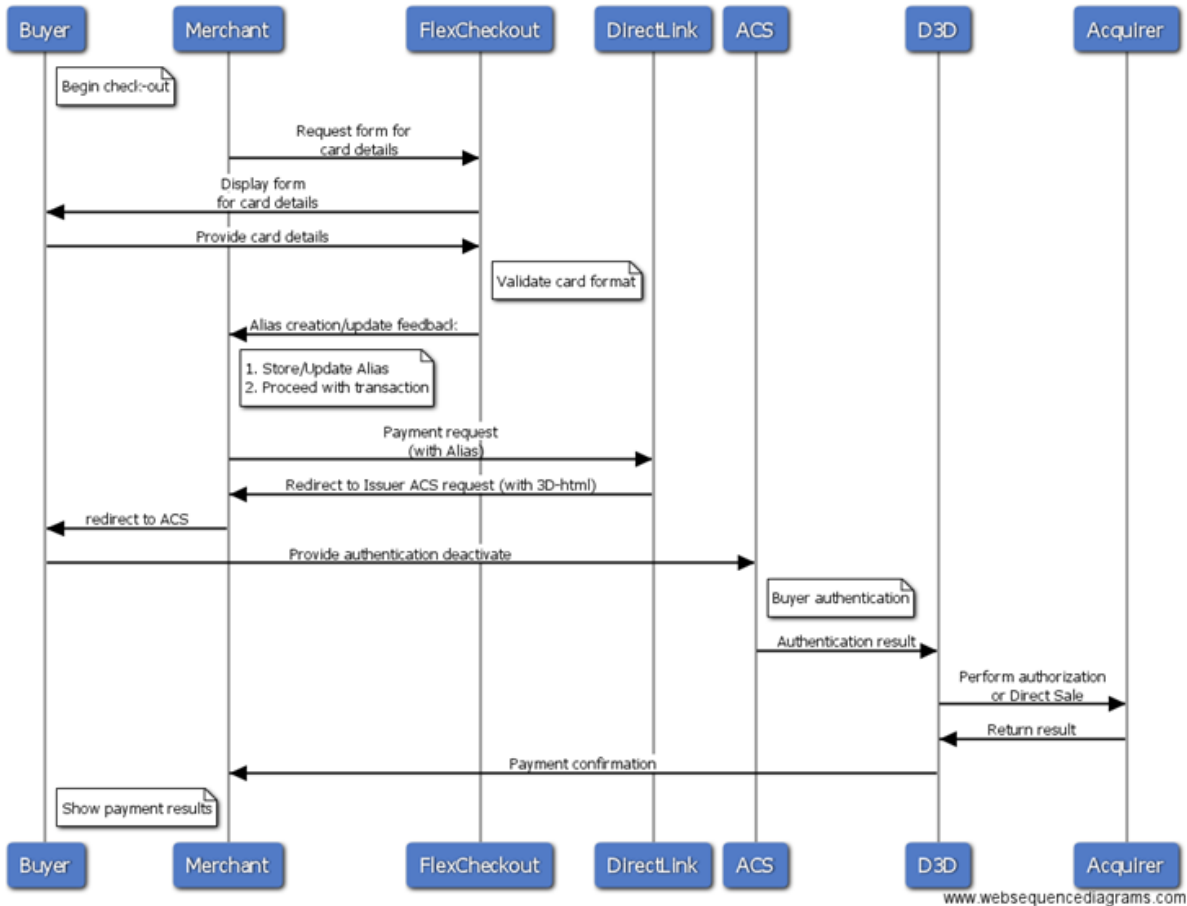
- One Page Checkout (ID opzione: OPC)
- Alias Manager (ID opzione: REC1, REC2, RECX)

In genere deve essere attivato anche DirectLink (nuovi pagamenti), se non è già abilitato per impostazione predefinita nell'abbonamento.

## 2. Flusso tipico dei dati

È possibile reindirizzare a (o includere in) un IFrame il FlexCheckout in cui il titolare della carta inserirà i dati della carta nel sistema. I dati della carta, insieme al codice CVC, vengono conservati in modo sicuro ma soltanto per un periodo di tempo limitato. Per salvare i dati della carta in modo permanente, è possibile contrassegnare l'alias come alias "Persistent" (permanente). Tuttavia, in conformità con gli standard di sicurezza PCI, i codici CVC possono essere memorizzati per due ore al massimo.

Tramite DirectLink è possibile inoltrare l'ordine effettivo con l'alias generato senza inoltrare i dati della carta.



**Nota:** Nessuna operazione viene eseguita sulla carta nel primo passaggio. Il nostro sistema esegue semplicemente una convalida di base del formato, ma non può garantire che la carta sia ancora valida, o abbia fondi sufficienti per procedere.

### 2.1 Integrazione DirectLink

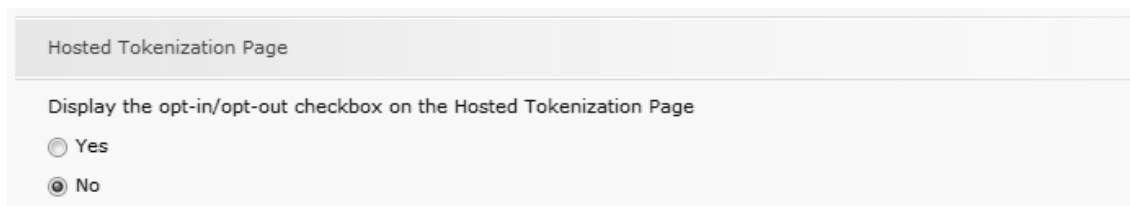
Utilizzando l'Alias generato con la FlexCheckout, è possibile quindi inviare una transazione DirectLink utilizzando la nostra implementazione standard DirectLink. Fare riferimento a [DirectLink](#) per le istruzioni sull'implementazione.

Questo meccanismo è anche compatibile con [DirectLink 3D](#). Per maggiori informazioni sull'utilizzo di Alias, vedere a [Alias Manager](#).

### 2.2 Utilizzo con Alias Manager

Se si utilizza l'opzione Alias Manager, è possibile inserire una casella di controllo nella FlexCheckout, consentendo al cliente di decidere se salvare o meno i suoi dati finanziari in modo permanente.

A tal fine, nella pagina di configurazione Alias Manager del proprio account Ingenico ePayments occorre attivare un'impostazione per la visualizzazione della casella di controllo (l'impostazione predefinita è "No").



Hosted Tokenization Page

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

Yes

No

> Si consiglia di utilizzare questa configurazione nell'impostazione di one-page-checkout.

In alternativa, se i processi di checkout sono stati integrati in più fasi, è possibile utilizzare [parametro "StorePermanently"](#) nella propria integrazione, per controllare quando visualizzare la casella di spunta.

### 3. Integrazione di FlexCheckout come tokenizzazione

È possibile usare FlexCheckout in due modi:

1. Reindirizzando il cliente su FlexCheckout
2. Incapsulando FlexCheckout in un IFrame in modo che i clienti rimangano nella pagina e completino l'esperienza One Page Check-out

**Avviso:** Non è consigliabile integrare FlexCheckout come soluzione pura incorporata nell'app, perché alcune funzionalità non sono disponibili sugli smartphone di fascia media ma soltanto su soluzioni per siti Web mobili che utilizzano tutte le funzionalità del browser.

Quando il cliente viene reindirizzato su FlexCheckout, deve inserire i dati della sua carta e inviare la tokenizzazione a FlexCheckout. In questo modo, i dati della carta non passano attraverso il vostro server web. Gli URL disponibili per accedere a FlexCheckout sono i seguenti

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

Per impostazione predefinita, la pagina Hosted Tokenization utilizza la codifica UTF-8 dei caratteri. Se utilizzate ISO, bisogna adattare di conseguenza la "Codifica dei caratteri" sulla piattaforma (backoffice), tramite "Configurazione" > "Informazione tecniche" > "Parametri globali di sicurezza" > "Metodo di hashing".

Se si usa la codifica ISO per calcolare l'SHA, gli esercenti dovranno impostarla nel Back Office. Quest'impostazione convalida l'SHA in entrata e l'SHA in uscita. Viene usata anche per impostare lo schema di codifica URL per i parametri restituiti."

#### 3.1 Campi di input

Campo	Descrizione	Formato	Obbligatorio
ACCOUNT.PSPID	Identificazione del merchant	AN, 30	Si
ALIAS.ALIASID	Alias cliente	AN, 50	No
ALIAS.ORDERID	Identificazione ordine	AN, 40	No
ALIAS.STOREPERMANENTLY	<p>Indicare se si desidera memorizzare un alias temporaneamente o a tempo indeterminato. I valori possibili sono:</p> <ul style="list-style-type: none"> <li>• "N": l'alias sarà eliminato dopo 2 ore</li> <li>• "Y": l'alias sarà memorizzato a tempo indeterminato, per un utilizzo futuro</li> </ul> <p>Questo parametro deve <u>essere utilizzato solo in combinazione con Alias Manager</u>. Per impostazione predefinita, l'alias sarà memorizzato in modo permanente.</p> <p>&gt; Se si utilizza un'integrazione one-page-checkout, si consiglia vivamente di attivare <a href="#">l'opzione "opt-in/out checkbox" nella configurazione Alias Manager</a>, per visualizzare sempre la casella di controllo nella FlexCheckout.</p>	Y / N	No

Campo	Descrizione	Formato	Obbligatorio
CARD.BIC	Bank Identification Code, utilizzato solo per gli addebiti diretti	AN, 8	
CARD.BIN	Metodi di tipo di pagamento con carte di credito	AN, 6	
CARD.BRAND	Indicare che tipo di modulo deve essere visualizzato > Obbligatorio se non è fornito alcun metodo di pagamento	AN, 25	Si/No
CARD.PAYMENTMETHOD	Carta di credito o qualsiasi metodo di addebito diretto supportato - Indicare che tipo di modulo deve essere visualizzato > Obbligatorio se non è fornito alcun Brand	AN, 26	Si/No
LAYOUT.LANGUAGE	Lingua utilizzata nella pagina ("en_EN", "nl_BE", "fr_FR", ecc.)	AN, 5	No
LAYOUT.TEMPLATENAME	Parametro di input del modello. Agli esercenti serve soltanto il nome del modello e l'estensione del file, ad esempio "new_user.html".	AN, 255	No
PARAMETERS.ACCEPTURL	URL per il reindirizzamento in caso di successo	AN, 255	Si
PARAMETERS.EXCEPTIONURL	URL per il reindirizzamento in caso di errore	AN, 255	Si
PARAMETERS.EXCLUDEDPAYMENTMETHODS	Elenco di metodi di pagamento e/o brand di carte di credito che NON devono essere mostrati. Separati da a ";" (punto e virgola).	AN, 50	No
PARAMETERS.PARAMPLUS	Campo pass-through: Ulteriori parametri da inviare da parte del merchant da recuperare all'interno dell'output	AN, 1000	No
SHASIGNATURE.SHASIGN	Stringa con hash che utilizza Algoritmi di hashing protetti.	AN, 128	Si

**ALIAS.STOREPERMANENTLY: NOTE IMPORTANTI**

- L'uso di ALIAS.STOREPERMANENTLY è previsto solo in combinazione con Alias Manager. Per impostazione predefinita, l'alias sarà memorizzato in modo permanente.
- Se si utilizza un'integrazione one-page-checkout, si raccomanda vivamente di abilitare l'opzione "casella di controllo opt-in/opt-out" nella configurazione di Alias Manager, in modo che la casella sia sempre visibile nella pagina di tokenizzazione ospitata.
- Se detta opzione è abilitata nella configurazione di Alias Manager, l'applicazione del parametro ALIAS.STOREPERMANENTLY verrà elusa.

**3.1.1 Controllo di OrderID duplicato**

Per impedire che gli hacker sostituiscano dati relativi alla carta collegati a un token specifico (catturando il link per avviare la richiesta e sostituendo i dati della carta con dati autentici, credibili per una carta autentica), viene eseguito un controllo dell'OrderID (ALIAS.ORDERID) che l'utente invia nella richiesta.

Se risulta che l'OrderID è già stato utilizzato per creare un token, l'aggiornamento dell'alias sarà rifiutato.

**Importante**

Ovviamente il controllo dell'OrderID funziona soltanto se viene inviato l'**OrderID (ALIAS.ORDERID)**. In ogni caso l'OrderID da solo non è sufficiente; è **necessario anche l'AliasID (ALIAS.ALIASID)**. Se l'AliasID non viene inviato, il controllo non verrà eseguito e verrà creato un alias nuovo.

Con la rilevazione di un OrderID duplicato e il conseguente blocco della richiesta, sarà visualizzato un messaggio di errore generico:

"Durante l'elaborazione della richiesta si è verificato un errore. Si prega di riprovare più tardi o rivolgersi all' esercente."

Errori generati da un OrderID duplicato possono essere rilevati quando la modalità di debug è attiva nei registri degli errori (si prega di rivolgersi a nostro Assistenza clienti).

### 3.2 Firma SHA per input

Per verificare l'integrità dei dati, si richiede che tutte le richieste siano accompagnate da una firma SHA, [allo stesso modo delle transazioni e-Commerce](#).

Il nostro sistema utilizzerà l' algoritmo SHA come definito nella scheda dei parametri di sicurezza globale della pagina delle informazioni tecniche. È ancora possibile reimpostare quest'algoritmo su SHA-1 o SHA-256.

**Esempio**

Campi (in ordine alfabetico):

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

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

Account.PspId: test1

Card.Brand:VISA

Passphrase segreta (come definita nelle informazioni tecniche): Mysecretsig1875!?

Stringa con hash:

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

Firma SHA ottenuta (SHA-512):

563DC909F70BA5DDD470D69C1B390E7D1C1C47705AC5801B27038446D7033B5787728EA  
754EF72E7FA2436FC5962E34E20DF64E7F9139893A33653F118816818

### 3.3 Campi di output

I campi seguenti possono essere riproposti all'utente per ricevere lo stato della creazione o dell'aggiornamento dell'alias. Affinché siano inclusi nel feedback, devono essere configurati di conseguenza nei parametri del feedback dinamico (Conto Ingenico ePayments: Configurazione > Informazione tecniche > Ritorno d'informazione della transazione > Alias gateway e Tokenization: Parametri dinamici).



Il campo SHASIGN non è facoltativo e verrà sempre riproposto.

Campo	Descrizione	Lunghezza max.
ALIAS.ALIASID	Alias inviato dal merchant o alias generato da PSP (Secondo il formato GUID a 32 cifre). Esempio: 34F5302C-85D7-4F35-BDF5-103CCEC2FB61	50
ALIAS.NCERROR	Codice di errore	50
ALIAS.NCERRORCARDNO	Codice di errore per CARDNO	50
ALIAS.NCERRORCN	Codice di errore per CN	50
ALIAS.NCERRORCVC	Codice di errore per CVC	50
ALIAS.NCERRORED	Codice di errore per ED	50
ALIAS.ORDERID	Identificativo unico dell'ordine. (inviato dal merchant o generato dal sistema)	40
ALIAS.STATUS	Risultato della creazione alias: <ul style="list-style-type: none"> <li>• 0=OK</li> <li>• 1=NOK</li> <li>• 2=Alias aggiornato</li> <li>• 3=Annullato dall'utente</li> </ul>	1
ALIAS.STOREPERMANENTLY	Indica la scelta se memorizzare l'alias temporaneamente (N) o a tempo indeterminato (Y).  Quando viene eseguita una transazione con questo alias, gli esercenti devono rispettare questo parametro nel campo ALIASPERSISTEDAFTERUSE. Altrimenti verrà salvata definitivamente.	1 (Y/N)
CARD.BIC	Codice BIC (Bank Identification Code), usato soltanto per addebiti diretti	8
CARD.BIN	Metodi di pagamento tipo carta di credito	6
CARD.BRAND	Marchio del metodo di pagamento	25
CARD.CARDHOLDERNAME	Nome del titolare della carta	50
CARD.CARDNUMBER	Carta con X in sostituzione delle informazioni sensibili.  Esempio: XXXXXXXXXXXXX1111  Nota: In caso di errore, la carta sarà mascherata.	35
CARD.CVC	Codice di verifica delle carte di credito, con X in sostituzione dei dati sensibili.  Esempio: XXX	6

Campo	Descrizione	Lunghezza max.
CARD.EXPIRYDATE	Data di scadenza, ad es. 0220 (Febbraio 2020)	4
PARAMPLUS	Campo pass-through: dati forniti nell'input	/
<b>SHASIGN</b>	<b>Firma SHA per output</b>	<b>128</b>

I parametri selezionati verranno acclusi all'URL di ritorno definito nella richiesta (PARAMETERS.ACCEPTURL o PARAMETERS.EXCEPTIONURL), per fornire il feedback relativo all'operazione

### 3.4 Firma SHA per output

Il nostro sistema genererà una firma SHA-OUT, [allo stesso modo delle transazioni e-Commerce](#), per i seguenti parametri:

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. Personalizzazione

Questo capitolo introduce due tipi di personalizzazione: di base e avanzata. Nella personalizzazione di base si imparerà ad applicare il proprio file CSS e a personalizzare i colori del testo, dello sfondo e di wrapper. Segue una pagina di pagamento predefinita con i file predefiniti CSS:

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.

Nella personalizzazione avanzata si conosceranno meglio i modelli master/parziali con i segnaposto e verranno personalizzati i contenuti di testo con i file XML. In base alle proprie esigenze, i modelli parziali vengono creati e personalizzati separatamente.

### 4.1 Personalizzazione di base

La personalizzazione di base include la configurazione del foglio di stile predefinito con il modello Ingenico ePayments predefinito

#### 4.1.1 Modello Ingenico predefinito

Il file CSS può essere personalizzato con un foglio di stile Ingenico ePayments predefinito o con un file CSS caricato in [Template File Manager](#).

Per maggiori informazioni sulla personalizzazione del file CSS consultare la sezione relativa all'uso dei [fogli di stile](#).

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 Uso dei fogli di stile

È possibile adattare il layout di FlexCheckout applicando un file CSS. Innanzitutto è necessario caricare il file CSS in Template File Manager e collegarlo all'account Ingenico ePayments. Il file CSS è composto da un modulo/contenitore dati e da elementi come data di scadenza, CVC, riquadro informazioni, ecc., da impostare come file CSS predefinito. È possibile collegare ciascun elemento che definisce la formattazione di ogni campo nel modulo/contenitore dati.

In base alle proprie esigenze aziendali, è possibile modificare la regola di formattazione di ciascun elemento. Fare clic qui per ottenere un elenco degli [elementi CSS](#).

Seguono un modulo/contenitore dati e gli esempi di elementi:

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

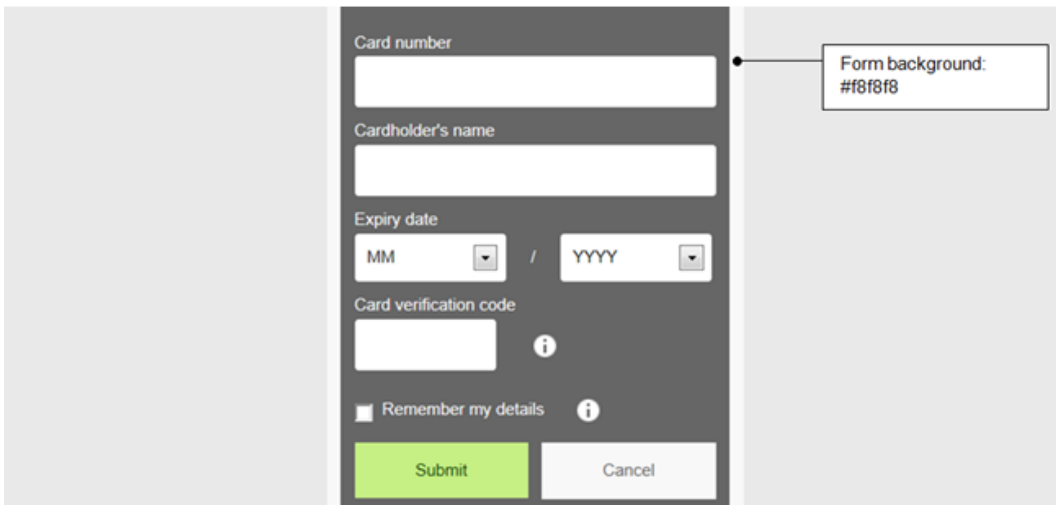
#payment-cardnumber-label-container,#payment-accountowner-label-container{ overflow: hidden;}

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

Per iniziare, abbiamo creato un file CSS di esempio che può essere scaricato da [qui](#) (file zip). È possibile iniziare a creare il proprio CSS con questo file.

È possibile modificare l'interfaccia utente di default (come mostrato qui sotto), applicando il proprio CSS.

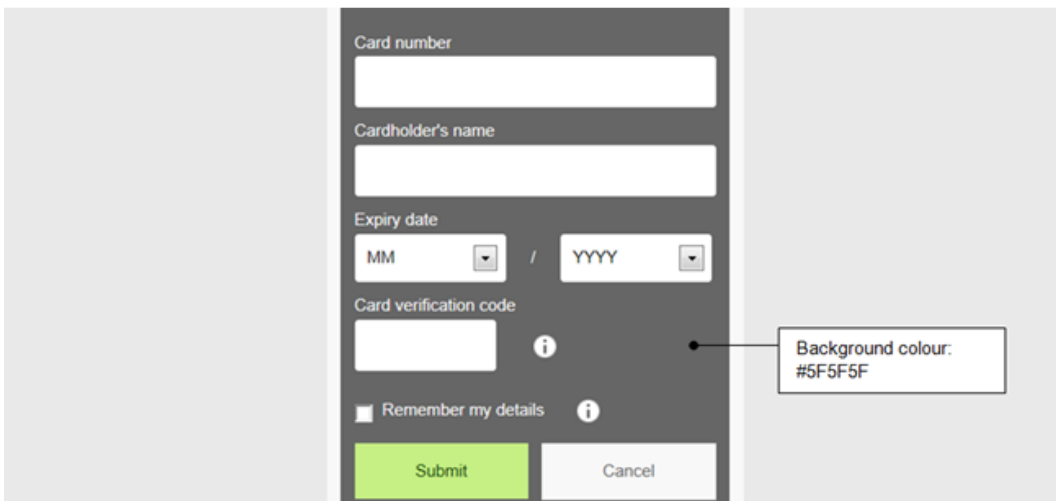
## Colore di sfondo modulo



CSS interessato:

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

Colore wrapper modulo

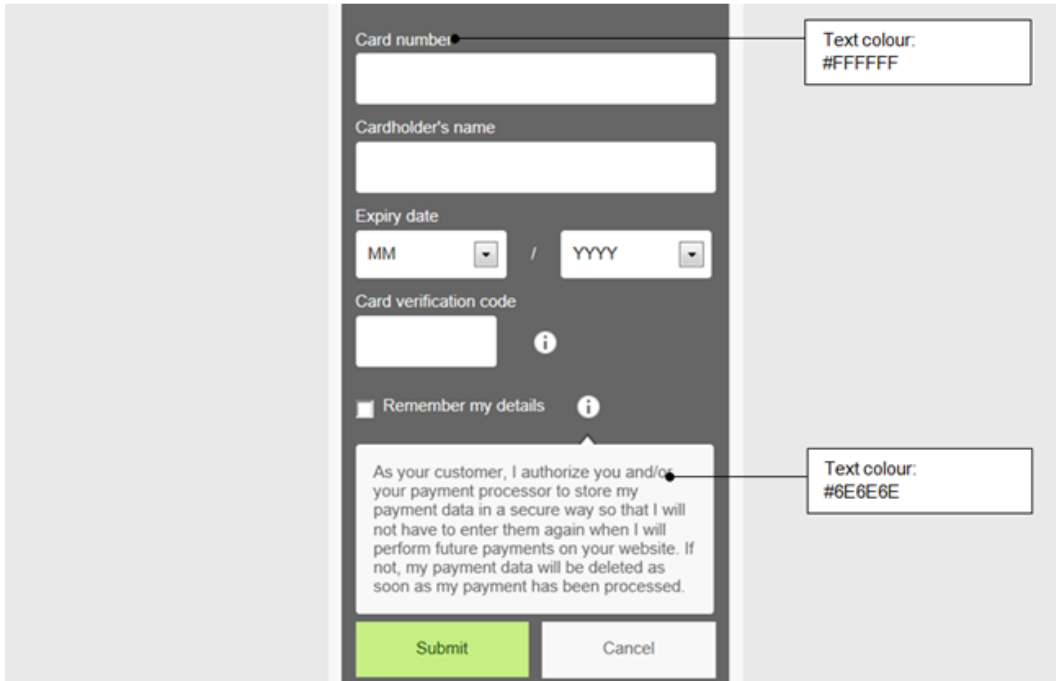


CSS interessato:

## FlexCheckout

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

### Colore testo

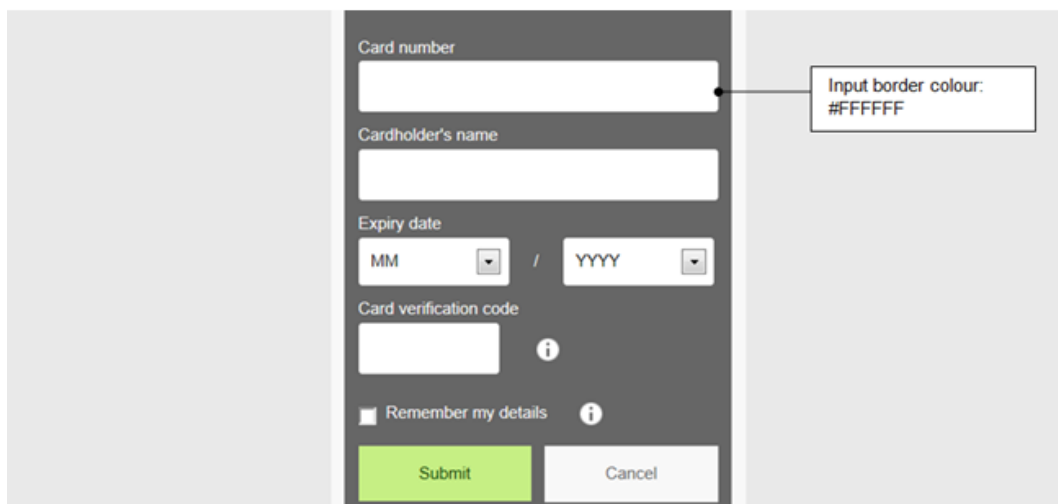


CSS interessato:

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

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

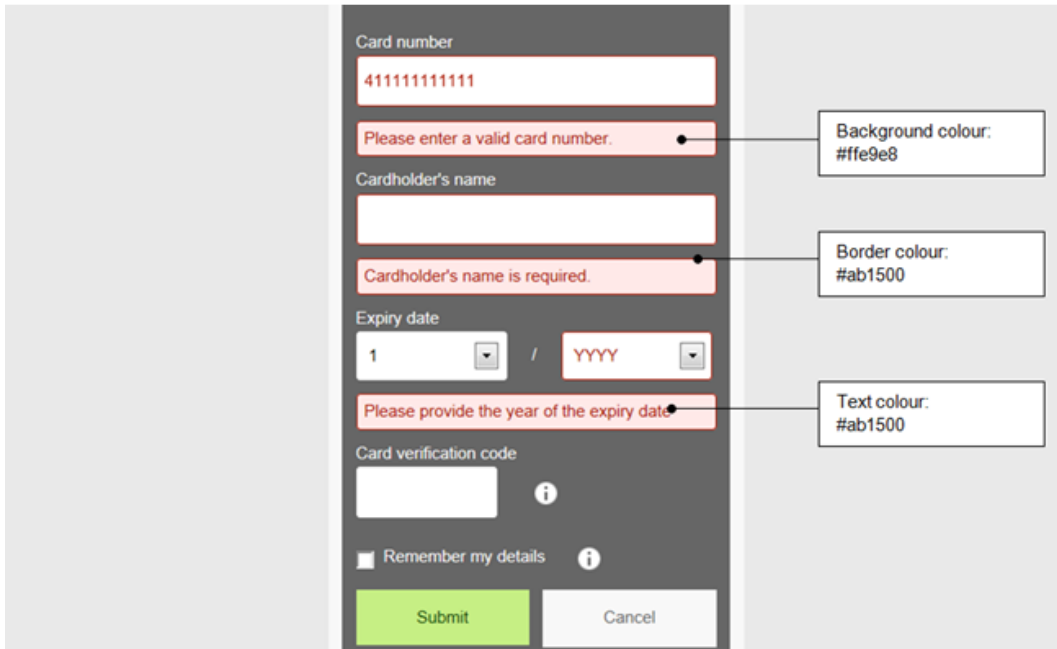
### Colore bordo di input



CSS interessato:

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

Casella messaggio di convalida



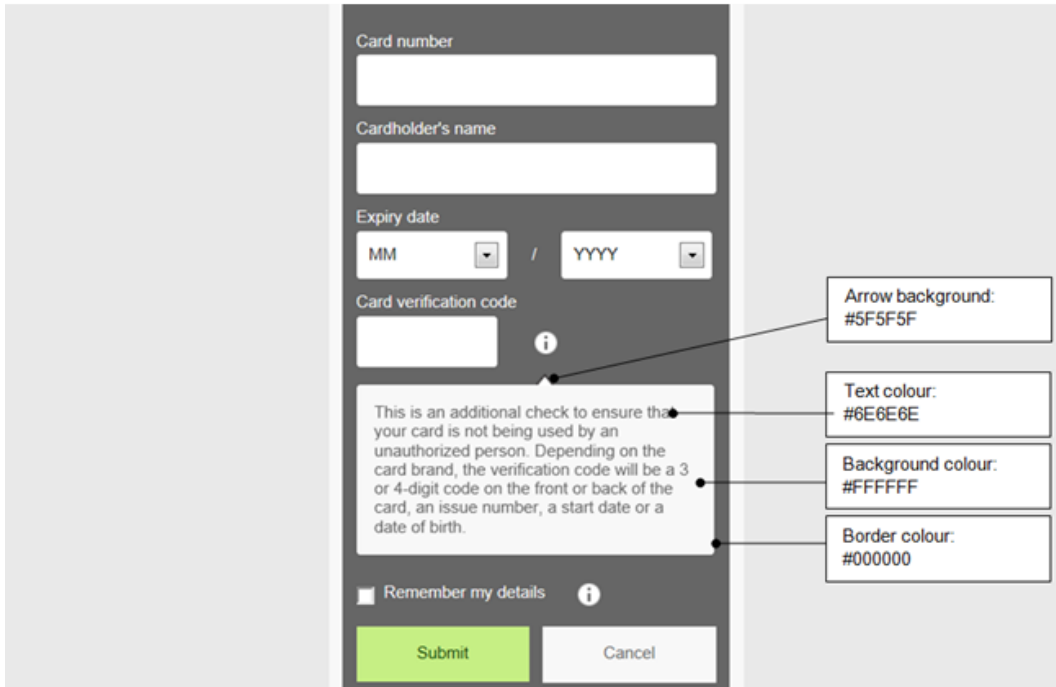
CSS interessato:

```
.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;}
```

Casella informazioni di guida





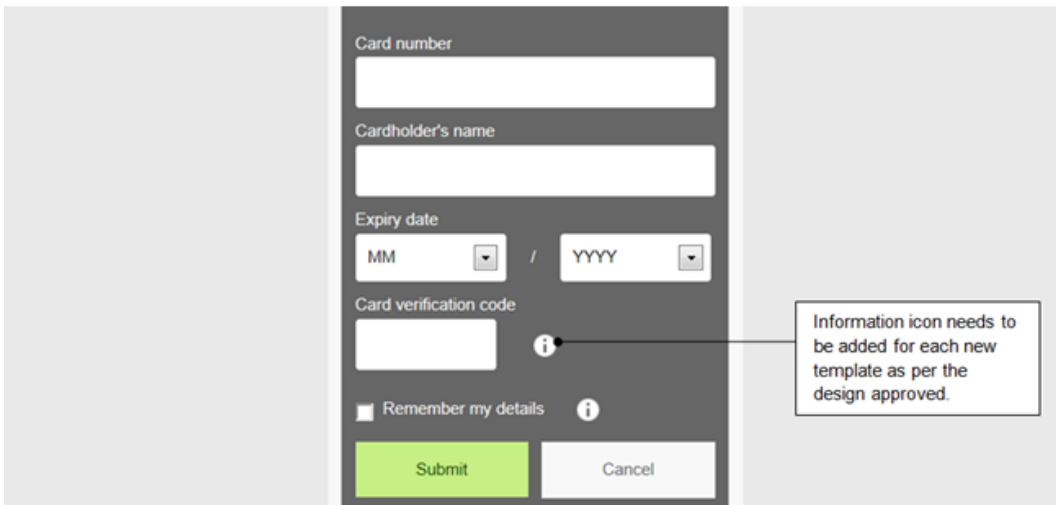
CSS interessato:

```
#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;}
```

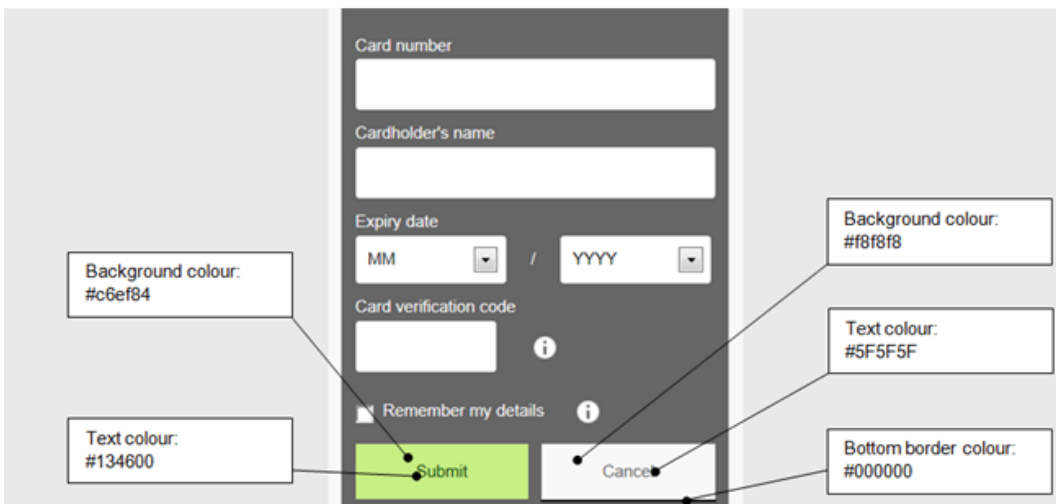
Icona informazioni



CSS interessato:

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

### Pulsante Invia/Annulla



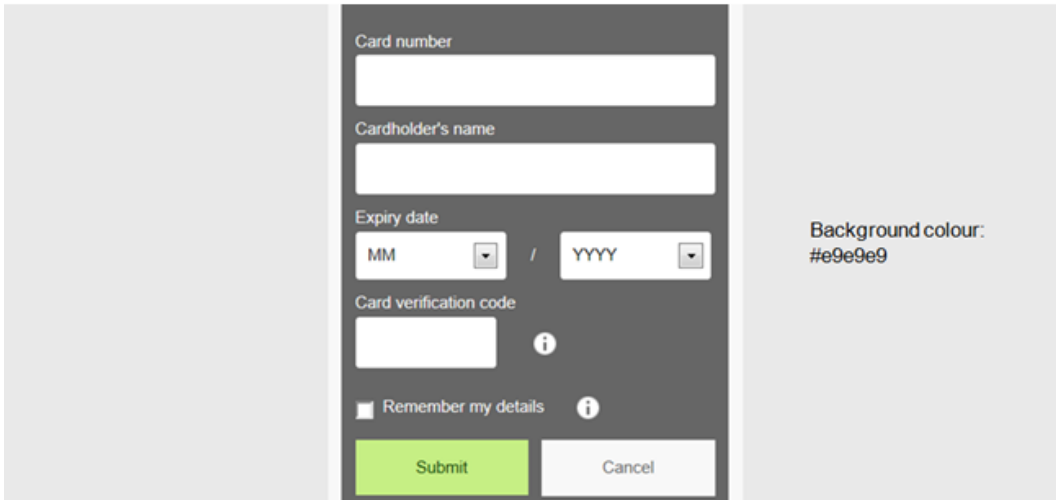
CSS interessato:

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

## FlexCheckout

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

### Colore di sfondo



CSS interessato:

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

### CSS di esempio

```
#payment-container {background:# e9e9e9} .payment-form-container{ Background:#f8f8f8; }  
#payment-data-container{ Background:#666; }  
#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 Personalizzazione avanzata

Per offrire maggiore flessibilità, ora è possibile creare e personalizzare modelli master e modelli parziali. In base alle proprie esigenze, i modelli parziali vengono creati e personalizzati separatamente. È altresì possibile personalizzare i contenuti di testo con i file XML.

È possibile scegliere di applicare un tag `$$$PAYMENT_ZONE$$$` che richiede un modello semplice oppure personalizzare il modello parziale tramite i [segnaposto](#).

#### 4.2.1 Uso dei segnaposto

È possibile personalizzare e formattare il loro modello master (master.html) e i modelli parziali possono inserire direttamente i segnaposto in master.html.

È anche possibile definire l'aspetto della pagina di pagamento tramite un file CSS personalizzato (**newuser.css**) da collegare al file master.html. Se si sceglie di personalizzare il file CSS, il file CSS modificato deve essere caricato in Template File Manager e impostato come predefinito.

I segnaposto vengono classificati in tre gruppi: segnaposto principali, elementi e raggruppati.

##### **Segnaposto principali**

Segue un elenco dei segnaposto principali:

- `$$$PAYMENT_ZONE$$$`: rappresenta un marchio richiesto o un metodo di pagamento mediante un modello personalizzato (con un modello parziale disponibile) o il modello predefinito.
- `$$$TP_RESOURCE_URL$$$`: rende disponibile l'URL per la directory in cui si trovano i file nel Servizio di File Hosting.

È inoltre possibile usare script personalizzati che devono essere collegati a un file JavaScript personalizzato. Il riferimento al file JavaScript può essere creato quando il file viene caricato in Template File Manager. Ad esempio, il file JavaScript di riferimento in `<script type="text/javascript" src="$$$TP_RESOURCE_URL$$$/existinguser.js"></script>` è **existinguser.js**.

Per usare le raccolte JavaScript nei modelli (ad es. JQuery), caricare i file JavaScript in Template File Manager. Dopo aver caricato i file, usare `$$$TP_RESOURCE_URL$$$` per creare il riferimento agli script. Questa procedura deve essere eseguita per tutti i file a cui si fa riferimento nel modello: CSS, immagini, caratteri, ecc., e garantisce che lo script eseguito sia il JavaScript dedicato.

**Nota:** non è possibile eseguire il push/pull di tutti i dati da un host di terzi non conforme a PCI, pertanto l'uso di JavaScript incide sull'idoneità dell'esercente per PCI SAQ A.

- `$$$DISABLE_MASKING$$$`: disattiva la mascheratura della casella di input.
- `$$$SUBMIT$$$`: rappresenta un pulsante di inoltro per il modulo. Ad esempio:

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

- `$$$CANCEL$$$`: rappresenta un pulsante di annullamento per il modulo. Ad esempio:

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

un modello master contenente `$$$PAYMENT_ZONE$$$` non deve includere i segnaposto di inoltro e di annullamento. I segnaposto di inoltro e di annullamento devono però essere inclusi se un modello parziale è

configurato per ogni metodo di pagamento o se un modello master è configurato senza il segnaposto \$\$\$PAYMENT\_ZONE\$\$\$.

### Segnaposto degli elementi

Il controllo effettivo <element> è rappresentato all'interno di un contenitore di elementi: <div id="payment-<input\_name>-<element>-container" class="payment-<element>">. Per semplificare lo stile CSS, ID e classe usano la stessa convenzione per i nomi.

- \$\$\$<input\_name><element>\$\$\$ : come controllo di input del numero di carta.
- \$\$\$CARD NUMBER.INPUT\$\$\$ - CARD NUMBER: come <input\_name>, in cui l'INPUT è l'<element> HTML.
- \$\$\$<input\_name>LABEL\$\$\$ : esegue il rendering dell'elemento <input\_name>.

Ad esempio, il segnaposto \$\$\$CARD NUMBER LABEL\$\$\$ rappresenta:

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

Per tutte le etichette verrà rappresentato un div per fungere da contenitore:

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

La classe di un contenitore di etichetta sarà "etichetta del pagamento" in modo che lo stile venga applicato a tutte le etichette insieme. Viene sempre rappresentata un'etichetta per il controllo dell'etichetta:

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

Per applicare uno stile al controllo dell'etichetta, applicare il file CSS all'ID del contenitore e non all'ID di controllo.

\$\$\$<input\_name> INPUT\$\$\$ rappresenta il controllo dell'input per <input\_name>:

```
<div id="payment-<input_name>-input-container" class="payment-input">  
  
<input Id="payment-<input_name>" autocomplete="off" autocapitalize="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>
```

Per applicare uno stile al controllo dell'input, applicare il file CSS all'ID del contenitore e non all'ID di controllo.

\$\$\$<input\_name> INFO\$\$\$ rappresenta un collegamento a una descrizione del campo <input\_name>:

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

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

</div>
```

Associare le eventuali informazioni per questo campo nella lingua selezionata.

\$\$\$<input\_name> ERROR\$\$\$ rappresenta il contenitore dei messaggi di errore da <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>
```

Qui è possibile inserire i messaggi di convalida.

### **Segnaposto raggruppati**

È possibile inserire i segnaposto degli elementi come gruppo usando i segnaposto Raggruppati. Segue un contenitore rappresentato con un segnaposto raggruppato:<div id="payment-<input\_name>-container"></div>. Nel quale verranno inseriti i segnaposto dell'elemento in questo gruppo. Di fatto viene rappresentato questo:

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

```

$$$<INPUT_NAME> LABEL$$$
$$$<INPUT_NAME> INPUT$$$
$$$<INPUT_NAME> ERROR$$$
$$$<INPUT_NAME> INFO$$$
</div>

```

Un modello master può contenere al suo interno dei segnaposto <div id="paymentForm">. I segnaposto (come \$\$\$CARD NUMBER\$\$\$, \$\$\$CARDHOLDER NAME\$\$\$, \$\$\$EXPIRY DATE\$\$\$, ecc.) definiscono il contenuto del modulo della carta di credito. Gli esercenti possono scegliere i segnaposto più adatti alle loro esigenze.

Segue un esempio di un modello master con segnaposto per un modulo di carta di credito etichettato "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>

```

Per creare un modulo di addebito diretto, gli esercenti possono utilizzare i segnaposto di addebito diretto, come il seguente:

```

$$$ACCOUNT_OWNER$$$
$$$BANK_ACCOUNT$$$
$$$BIC$$$

```

Il nome del conto bancario, il codice IBAN e il codice BIC rappresentano un modulo di addebito diretto.

Per gli esercenti che desiderano un modello che supporti diversi metodi di pagamento, si può creare un modello master e un modello parziale per i metodi di pagamento.

Un modello master è come il seguente:

```
<!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>
```

Un modello parziale è come il seguente ed è realizzato con segnaposto (\$\$\$CARD NUMBER\$\$\$, \$\$\$CARDHOLDER NAME\$\$\$, \$\$\$EXPIRY DATE\$\$\$, and so on):

```
<div id="paymentForm">
  $$$CARD NUMBER$$$
  $$$CARDHOLDER NAME$$$
  $$$EXPIRY DATE$$$
  $$$CVC$$$
  $$$SUBMIT$$$
  $$$CANCEL$$$
</div>
```

Se gli esercenti non vogliono rappresentare il modulo di pagamento predefinito con \$\$\$PAYMENT ZONE\$\$\$, possono inserire invece un modello parziale.

Se ad esempio il metodo di pagamento richiesto è una carta di credito, viene usato un modello parziale con il nome master.creditcard.htm. Se si tratta di addebito diretto, viene usato invece un modello parziale con il nome master.directdebit.htm.

Gli esercenti devono caricare il modello parziale in File Manager. Devono attenersi rigorosamente alla convenzione di denominazione. Se ad esempio il modello master esistente si chiama master.html e se gli esercenti intendono inviare un addebito diretto come metodo di pagamento, in File Manager dovrà essere caricato un modello parziale di nome master\_directdebit.html.

#### 4.2.2 Personalizzazione dei contenuti di testo con XML

È altresì possibile personalizzare l'aspetto del testo nelle pagine di tokenizzazione con un file XML. Se non viene applicato un file XML, viene usato il modulo di tokenizzazione. Dopo aver personalizzato il file XML è necessario caricarlo in Template File Manager e usare lo stesso nome di file come modello master. Requisito preliminare: il modello master deve essere caricato in anticipo.

Insieme al file XML devono sempre essere presenti degli elementi identificativi dei campi. Si prega di cliccare qui per una [lista completa di elementi](#). Segue un esempio di un elemento (CardHolderNameInput) per l'immissione del nome del titolare della carta:

```
<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 questo elemento nell'ID lingua vengono specificate le lingue inglese americano e francese. Anche il tag ID specifica il codice della cultura ISO.

Se non viene specificato alcun elemento viene usato il testo predefinito dell'elemento. Se non viene specificata la lingua di un elemento viene usata la lingua predefinita.

Nota:

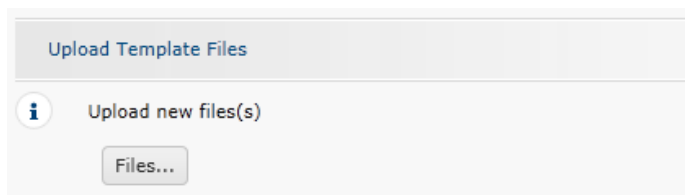
l'ultima riga del file XML deve sempre iniziare con <root>. L'ultima riga del file XML deve sempre finire con </root>.

## 4.3 Template File Manager

Template File Manager consente di gestire facilmente i modelli e i vari file ad essi associati.

Per iniziare a usare File Manager, accedere al proprio account Ingenico ePayments, quindi aprire "Configurazione" > "Modello" > "File Manager".

### 4.3.1 Caricamento di file modello



Sotto "Carica file modello", selezionare il pulsante "File..." per cercare i file che si desidera caricare. È possibile caricare css e immagini (.css, .jpg, .jpeg, .gif, .png), con un massimo di 7 MB a file, e 10 MB in totale.



Effettuare la selezione e confermare.

### 4.3.2 Controllo e gestione dei file caricati

Al termine del caricamento, i file appariranno nella stessa pagina nella sezione "File caricati".

I file presenteranno prima lo stato "Convalida", durante il quale vengono eseguiti alcuni necessari controlli di sicurezza/antivirus (ciò può richiedere alcuni minuti).






I file potranno essere usati quando lo stato passa a "Convalidato".

Fare clic sul pulsante di aggiornamento  per verificare l'attuale stato dei file / fare clic sul pulsante di eliminazione  per eliminare il file in modo permanente.

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

Q

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	

Un file assumerà lo stato "Rifiutato" se non passa il controllo di sicurezza. Ciò può essere dovuto a un virus o, ad esempio, se l'estensione del file è errata.

### 4.3.3 Integrazione

Nel codice si fa riferimento ai file caricati con un codice che segue questa struttura: \$\$\$TP RESOURCES URL\$\$\$/[nome file].



## 5. Segnaposto

Segue un elenco completo dei segnaposto e del relativo obiettivo:

Segnaposto principali	Descrizione
\$\$\$PAYMENT ZONE\$\$\$	Un modello personalizzato da un modello parziale (se disponibile) o un modello predefinito che rappresenta il marchio richiesto o il metodo di pagamento richiesto.
\$\$\$TP RESOURCES URL\$\$\$	Richiede un file precedentemente caricato nel Back Office.
\$\$\$DISABLE MASKING\$\$\$	Rappresenta una casella di testo corrispondente ai tipi di carta e alla formattazione dell'input.  Disabilita il mascheramento della casella di input quando si rileva il tipo di carta e formatta le caselle di input in base alla marca della scheda.
\$\$\$SUBMIT\$\$\$	Un pulsante di invio deve essere presente nel modulo.
\$\$\$CANCEL\$\$\$	Un pulsante Annulla sul modulo.
\$\$\$CARD NUMBER\$\$\$	Un numero di carta di credito con convalida e mascheramento adatti al marchio.
\$\$\$CARD NUMBER LABEL\$\$\$	Un'indicazione per un numero di carta di credito.
\$\$\$CARD NUMBER INPUT\$\$\$	Il campo in cui viene inserito il numero di carta.
\$\$\$CARD NUMBER INFO\$\$\$	Il campo in cui vengono inserite le informazioni sul numero di carta.
\$\$\$CARD NUMBER ERROR\$\$\$	Un messaggio di errore che compare quando viene immesso per sbaglio un numero di carta errato.
\$\$\$CARDHOLDER NAME\$\$\$	Un nome di un titolare della carta di credito.
\$\$\$CARDHOLDER NAME LABEL\$\$\$	Un'indicazione per il nome del titolare della carta.
\$\$\$CARDHOLDER NAME INPUT\$\$\$	Il campo per l'inserimento del nome del titolare della carta.

\$\$\$CARDHOLDER NAME INFO\$\$\$	Il campo per l'inserimento di informazioni sul nome del titolare della carta.
\$\$\$CARDHOLDER NAME ERROR\$\$\$	Un messaggio di errore che compare quando viene immesso per sbaglio un nome errato del titolare della carta.
\$\$\$EXPIRY DATE\$\$\$	L'area in cui è visualizzata la data di scadenza.
\$\$\$EXPIRY DATE LABEL\$\$\$	Un'indicazione della data di scadenza.
\$\$\$EXPIRY DATE INPUT\$\$\$	L'area in cui viene immessa una data di scadenza.
\$\$\$EXPIRY DATE MONTH\$\$\$	Il mese della data di scadenza.
\$\$\$EXPIRY DATE YEAR\$\$\$	L'anno della data di scadenza.
\$\$\$EXPIRY DATE INFO\$\$\$	Il campo in cui vengono inserite le informazioni sulla data di scadenza.
\$\$\$EXPIRY DATE ERROR\$\$\$	Un messaggio di errore che compare quando viene immessa per sbaglio una data di scadenza errata.
\$\$\$CVC\$\$\$	Il campo in cui è presente un CVC con la convalida appropriata.
\$\$\$CVC LABEL\$\$\$	Un'indicazione per CVC.
\$\$\$CVC INPUT\$\$\$	Il campo in cui viene immesso un CVC.
\$\$\$CVC INFO\$\$\$	Il campo per visualizzare le informazioni CVC.
\$\$\$CVC ERROR\$\$\$	Un messaggio di errore che compare quando viene immesso per sbaglio un CVC errato.
\$\$\$BANK ACCOUNT\$\$\$	Un numero di conto per i moduli di addebito diretto.
\$\$\$ACCOUNT OWNER\$\$\$	L'area in cui viene visualizzato il titolare del conto.
\$\$\$ACCOUNT OWNER LABEL\$\$\$"	Un'indicazione del titolare del conto.
\$\$\$ACCOUNT OWNER INPUT\$\$\$	Il campo in cui viene inserito il titolare del conto.
\$\$\$STORE PERMANENTLY\$\$\$	Una casella di controllo per l'esercente per accettare o rifiutare o salvare l'alias definitivamente.
\$\$\$BIC\$\$\$	Una casella BIC per gli alias addebiti diretti.

## XML elements

<p><b>Alias form</b></p>	<ul style="list-style-type: none"> <li>• PageTitle</li> <li>• Submit</li> <li>• Cancel</li> <li>• RememberMyDetails</li> <li>• AliasAuthorizationInfo</li> </ul>
<p><b>Credit cards</b></p>	<ul style="list-style-type: none"> <li>• PageTitle</li> <li>• CardNumber</li> <li>• CardNumberIsRequired</li> <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>• ExpiryYearIsRequired</li> <li>• InvalidExpiryYear</li> <li>• InvalidExpiryYearLength</li> <li>• InvalidExpiryYearCharacters</li> <li>• Cvc</li> <li>• CvcsRequired</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>• AureoreCvcInfo</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> </ul>

- GiroAccount
- BicName
- Iban
- BankAccount
- InvalidIbanCharacters
- InvalidIbanStructure
- InvalidCheckDigits
- InvalidIbanCountry
- InvalidIbanLength
- InvalidIbanNumber
- InvalidBankAccountNumber
- IbanOrBankAccount
- IncorrectBic
- BizIsRequired