FlexCheckout

© 2019 PostFinance

Table of contents

© 2019 PostFinance

1. Introduction

One Page Checkout with FlexCheckout enables you to request your customers to introduce their card details safely on the PSP system, tokenize these data and re-use the Token (or Alias) in a future transaction.

Advantages

- You have full control over the look & feel of the payment page via the flexible template mechanism. You can even control the check-out sequence in an iframe.
- You have the advantage of a secure, PCI compliant payment page without having to store the card data yourself; all sensitive data is directly introduced on our hosted page.
- Seamless integration for various checkout scenarios including One Page Checkout.
- You can offer up-selling and cross-selling on the final checkout page.
- Supports all major credit card brands (VISA, MasterCard, American Express, Diners) and PostFinance card.

Required options

You need to have at least one of the following options enabled in order to use the FlexCheckout:

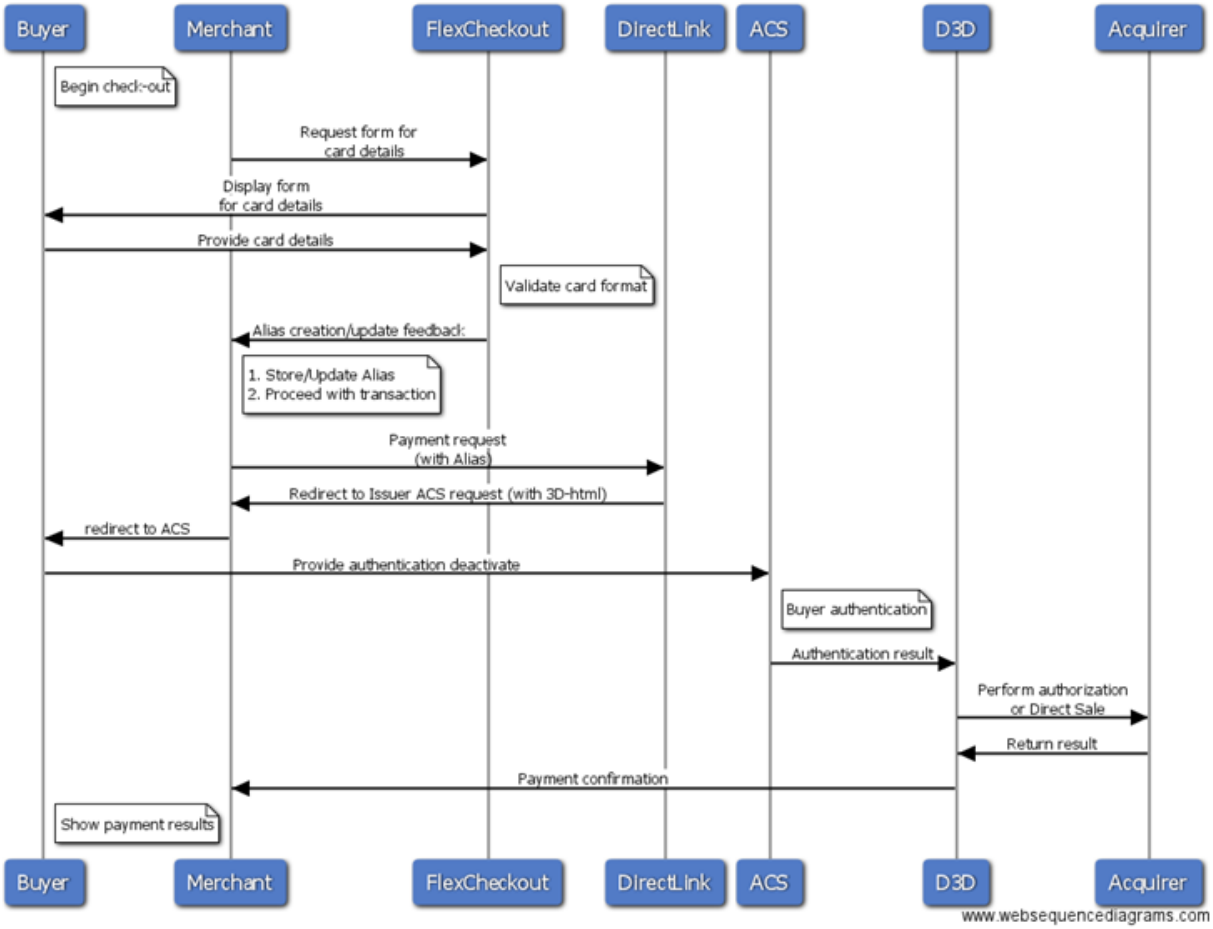
- One Page Checkout (option ID: OPC)
- Alias Manager (option ID: REC1, REC2, RECX)

Typically you also need to have DirectLink (new payments) activated, if it's not already enabled by default in your subscription.

© 2019 PostFinance

2. Typical data flow

You can redirect to (or include in) an iframe the FlexCheckout where the cardholder will input the card data into our system. The card data, along with the CVC, is securely stored but for a limited time only. To store the card data permanently, you can flag the Alias as a "Persistent" alias. However, in compliance with PCI Security Council standards, CVCs can only be stored for a maximum of two hours. Through DirectLink, you can also submit the actual order with the generated alias, without submitting the card data.



Note: No operation is performed on the card in the first step. Our system simply performs a basic format validation, but cannot guarantee that the card is still valid, or has sufficient funds to proceed.

2.1 DirectLink

To use the Token/Alias generated with FlexCheckout, you should submit a DirectLink transaction using our standard DirectLink implementation. Go to DirectLink for implementation instructions.

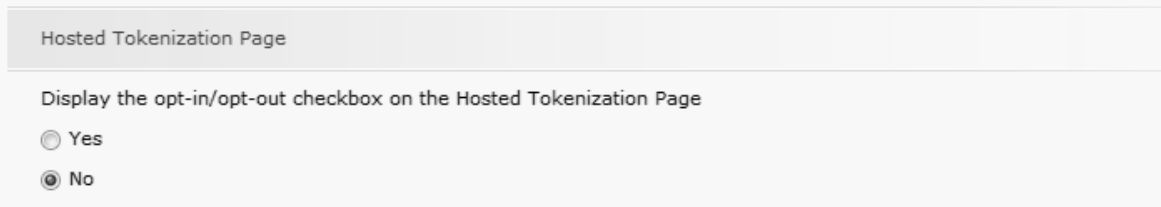
This mechanism is also compatible with DirectLink 3-D. For more information about Alias usage, go to Alias Manager.

2.2 Alias Manager

© 2019 PostFinance

If you use the Alias Manager option, you can make a checkbox available on FlexCheckout, allowing the customer to decide whether or not to save his financial details.

To do so, in the Alias Manager configuration page of your PostFinance account you must enable a setting for displaying the check box (the default is "No").



> We recommend you to use this configuration in a one-page-checkout setup.

Alternatively, if you've integrated a checkout processes in several steps, you can make use of the "StorePermanently" parameter in your integration, to control by yourself when the tick box should be displayed.

© 2019 PostFinance

3. Integrating FlexCheckout as a tokenization page

You can use FlexCheckout in two ways:

1. By redirecting your customer to FlexCheckout
2. By encapsulating FlexCheckout within an iframe so customers remain on your page and build a complete One Page Check-out experience

Warning: We do not advise integrating FlexCheckout as pure in-app solution as some functions would not work on certain mid-range smartphones but rather on a mobile website solution using full browsers capabilities.

When the customer is redirected to FlexCheckout, he will need to fill his card details and submit for tokenization onto FlexCheckout. In this way, the card details never pass through your own web server. The URLs to access FlexCheckout are:

- Test: https://postfinance.test.v-psp.com/Tokenization/HostedPage
- Production: https://e-payment.postfinance.ch/Tokenization/HostedPage

For PostFinance Card:

- Test: https://postfinance.test.v-psp.com/Tokenization/Gateway
- Production: https://e-payment.postfinance.ch/Tokenization/Gateway

By default, FlexCheckout works with UTF-8 character encoding. If you work with ISO, make sure to adapt the "Character encoding" accordingly in your back office, via Configuration > Technical information > Global security parameters > Hashing method.

If ISO encoding is used in calculating the SHA, you will need to set this in the Back-Office. This setting validates the incoming SHA and the outgoing SHA. It will also be used to set the URL encoding scheme for the returned parameters.

3.1 Input fields

Field	Description	Format	Mandatory
ACCOUNT.PSPID	A merchant identification	AN, 30	Yes
ALIAS.ALIASID	A customer alias. If left empty, a customer alias will be automatically created.	AN, 50	No
ALIAS.ORDERID	A payment identification that is used to avoid duplicated alias creation.	AN, 40	No
ALIAS.STORE PERMANENTLY	It indicates whether you want to store a temporary (N) or indefinite (Y) Alias. The possible values are: • "N": the alias will be deleted after 2 hours. • "Y": the alias will be stored indefinitely, for future use. Note: If an Alias is created with the N value and the transaction is completed within a two-hour timeframe, the transaction too must include this parameter/value combination for the alias to be	Y / N	No

© 2019 PostFinance

Field	Description	Format	Mandatory
	deleted. If the transaction does not contain this parameter/value combination, the alias will be retained for future use.		
CARD.BIC	Bank Identification Code, used only for direct debits	8	
CARD.BIN	Credit-card type payment methods	6	
CARD.BRAND	Indicate which type of form needs to be displayed. PostFinance cards included. > Mandatory if no Payment Method is provided	AN, 25	Yes/No
CARD.PAYMENTMETHOD	CreditCard or any supported Direct Debit Methods - Indicate which type of form needs to be displayed > Mandatory if no Brand is provided	AN, 26	Yes/No
LAYOUT.LANGUAGE	Language used on the ISO code format language(iso639)_Country(iso3166) ("en_EN", "nl_BE", "fr_FR" etc.) .	AN, 5	No
LAYOUT.TEMPLATENAME	Template input parameter. Include the name of the template and file extension, such as "new_user.html".	AN, 255	No
PARAMETERS.ACCEPTURL	URL for redirection in the event of success	AN, 255	Yes
PARAMETERS.EXCEPTIONURL	URL for redirection in the event of error	AN, 255	Yes
PARAMETERS.EXCLUDED PAYMENTMETHODS	List of payment methods and/or credit card brands that should NOT be shown. Separated by a ";" (semicolon).	AN, 50	No
PARAMETERS.PARAMPLUS	Pass-Through field: Additional parameters to be sent by the merchant to retrieve within the output	AN, 1000	No
SHASIGNATURE.SHASIGN	String hashed using the Secure Hash Algorithm.	AN, 128	Yes
LAYOUT.DEVICE	Displays the authentication page of a PostFinance card on a mobile phone. Value: Mobile	AN, 6	No

© 2019 PostFinance

Alias.StorePermanently: Important notes

- Alias.StorePermanently should only be used in combination with Alias Manager. If the RECX option under Alias Manager is not activated on your account, the alias will only be stored during the transaction and for no more than 2 hours as allowed by the PCI DCSS rules.
- If you are using a one-page-checkout integration, you should enable the "opt-in/out checkbox" option in your Alias Manager configuration to display the checkbox.
- When Alias Manager is activated, you should be able to overlook the data stored by the merchant by ticking on the Opt-in/out checkbox. If the Opt-in/out checkbox option under Alias Manager configuration is enabled, it will override the use of Alias.StorePermanently parameter.

3.1.1 Check for duplicated OrderID

In order to prevent hackers from replacing card details linked to a specific Token (by capturing the link to trigger the request and replacing the card details of a genuine card with genuine card details), we perform a check on the OrderID (ALIAS.ORDERID) that you sent in the request.

If the OrderID has been detected to be used in creating a token, the alias update will be refused.

Important
The check on the OrderID only works if the **OrderID (ALIAS.ORDERID)** is sent. The OrderID alone is not enough, the **AliasID (ALIAS.ALIASID) is required** as well. If the AliasID is not sent, the check will not be performed and a new alias will be created.

With the detection of a duplicate OrderID and consequently the request being blocked, a generic error message will be displayed to the customer:

"An error has occurred while processing your request. Please try again later or contact your merchant."

Errors generated by a duplicate OrderID can be detected when the debug mode in the Error logs is activated (please contact our customer care Merchanthelp).

3.2 SHA signature for input

To check the integrity of the data, we require all requests to be accompanied by a SHA signature, in the same way as for e-Commerce transactions.

Our system will use the SHA algorithm as defined in the Global security parameters tab of your Technical information page. It stays possible for you to change this algorithm back to SHA-1 or SHA-256.

Example

Fields (in alphabetical order):

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

Secret passphrase (as defined in Technical information): Mysecretsig1875!?

© 2019 PostFinance

String to hash:

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

Resulting SHA signature (SHA-512):

563DC909F70BA5DDD470D69C1B390E7D1C1C47705AC5801B27038446D7033B5787728EA754EF72E7FA2436FC5962E34E20DF64E7F9139893A33653F118816818

3.3 Output fields

The following fields, representing the status of the alias creation/update, can be returned to you. To include them in the feedback, they will need to be configured accordingly in the dynamic feedback parameters (PostFinance account: Configuration > Technical information > Transaction feedback > Alias gateway and Tokenization: Dynamic parameters).

The SHASIGN is not optional and thus will always be returned.

Field	Description	Max. Length
ALIAS.ALIASID	Alias sent by merchant or Generated alias by PSP (According to the 32-digit GUID format.) Example: 34F5302C-85D7-4F35-BDF5-103CCEC2FB61	50
ALIAS.NCERROR	Error code	50
ALIAS.NCERRORCARDNO	Error code for CARDNO	50
ALIAS.NCERRORCN	Error code for CN	50
ALIAS.NCERRORCVC	Error code for CVC	50
ALIAS.NCERRORED	Error code for ED	50
ALIAS.ORDERID	The unique identifier of the order. (sent by Merchant or generated by system)	40
ALIAS.STATUS	Result of the alias creation • 0=OK • 1=NOK • 2=Alias updated • 3=Cancelled by user	1
ALIAS.STOREPERMANENTLY	It indicates whether you want to store a temporary (N) or indefinite (Y) Alias. The possible values are: • "N": the alias will be deleted after 2 hours.	1 (Y/N)

© 2019 PostFinance

Field	Description	Max. Length
	• "Y": the alias will be stored indefinitely, for future use. Note: If an Alias is created with the N value and the transaction is completed within a two-hour timeframe, the transaction too must include this parameter/value combination for the alias to be deleted. If the transaction does not contain this parameter/value combination, the alias will be retained for future use.	
CARD.BIC	Bank Identification Code, used only for direct debits	8
CARD.BIN	Credit-card type payment methods	6
CARD.BRAND	Brand of the payment method PostFinance included.	25
CARD.CARDHOLDERNAME	Cardholder name	50
CARD.CARDNUMBER	Card with Xs to replace sensitive information. Example: XXXXXXXXXXXX1111 Note: In the event of an error, the card will also be masked.	35
CARD.CVC	Card Verification Code for credit cards, with Xs to replace sensitive data. Example: XXX	6
CARD.EXPIRYDATE	Expiry date, e.g. 0220 (February 2020)	4
PARAMPLUS	Pass-through field: data provided in the input	/
SHASIGN	**SHA signature for output**	**128**

In order to provide the feedback on the operation, the selected parameters will be appended to the Return URL defined in your request (PARAMETERS.ACCEPTURL or PARAMETERS.EXCEPTIONURL).

3.4 SHA signature for output

Our system will return a SHA-OUT signature, in the same way as e-Commerce transactions, for the following parameters:

ALIAS.ALIASID
ALIAS.NCERROR
ALIAS.NCERRORCARDNO
ALIAS.NCERRORCN

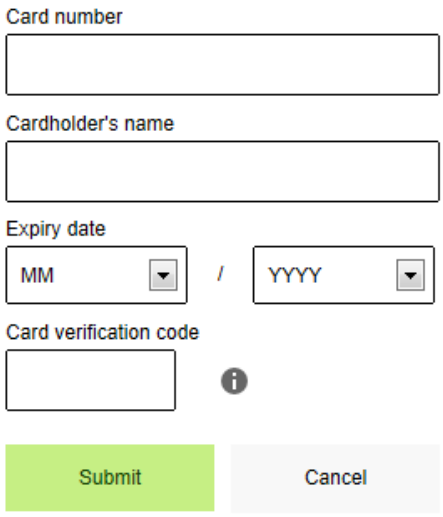
© 2019 PostFinance

ALIAS.NCERRORCVC
ALIAS.NCERRORED
ALIAS.ORDERID
ALIAS.STATUS
ALIAS.STOREPERMANENTLY
CARD.BIC
CARD.BIN
CARD.BRAND
CARD.CARDHOLDERNAME
CARD.CARDNUMBER
CARD.CVC
CARD.EXPIRYDATE

© 2019 PostFinance

4. Customization

This chapter introduces two types of customization: basic and advanced. Under Basic Customization, you will learn to apply your own CSS and customize text colors, background and wrapper colors, and such. The following is an example of a default payment page using the default CSS file:



In the Advanced Customization, you will learn how to customize further the content of your form using master and partial templates, placeholders, and customizing text content with XML files. Depending on your needs, the partial templates are created and customized separately.

4.1 Basic customization

A basic customization involves configuring default style sheet under default PostFinance template.

4.1.1 Default PostFinance template

The CSS file can either be customized with a default PostFinance style or uploaded CSS file in the Template File Manager.

For more information about customizing your own CSS file, refer to using style sheets.

4.1.2 Using style sheets

You can adapt the FlexCheckout layout by applying a CSS file. You will first need to upload your CSS file to the Template File Manager and link it to your PostFinance account. The CSS file consists of form/data container and elements (such as: Expiry date, CVC, Information box, and etc.) that must be set as the default CSS file. You can link each element that defines the formatting for each field to the form/data container.

Based on your business needs, you can modify the formatting rule within each element. Click here for a list of CSS elements.

The following are Form/Data container and element samples:

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

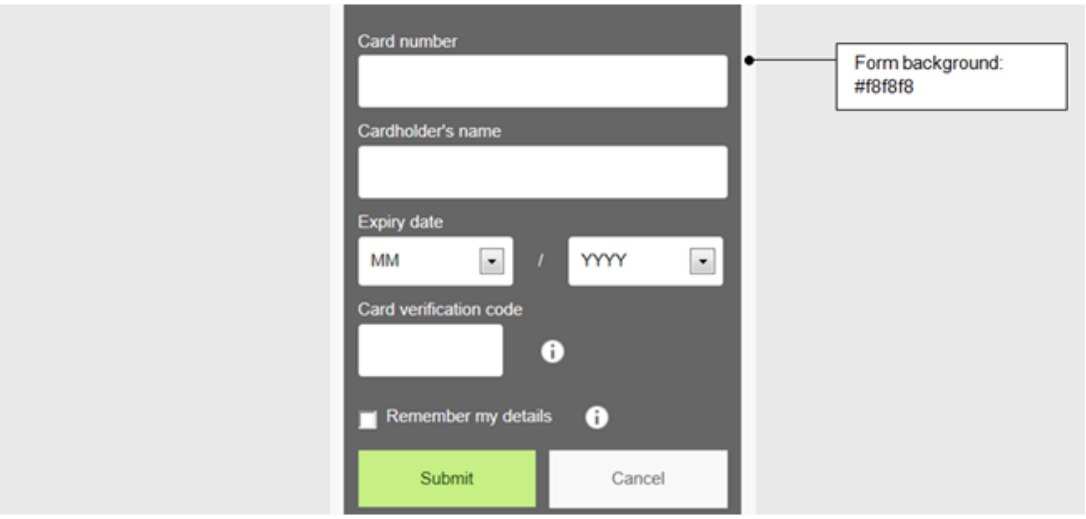

© 2019 PostFinance

```
#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----------*/
```

To get you started, we have created a sample CSS file which you can download from here (zip file). You can start creating your own CSS with this file.

You can change the default user interface (as displayed below) by applying your own CSS.

Form background color

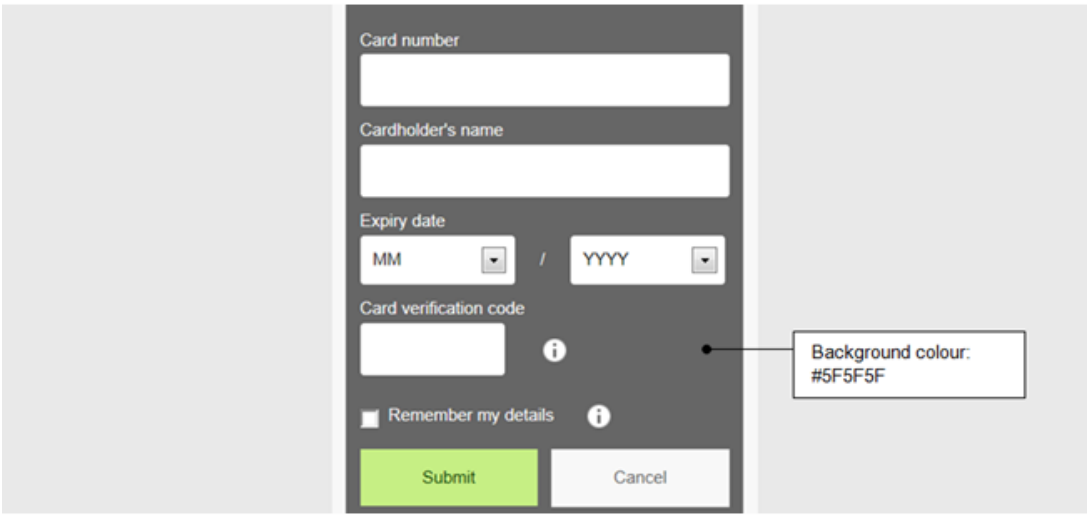


© 2019 PostFinance

Impacted CSS:

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

Form wrapper color

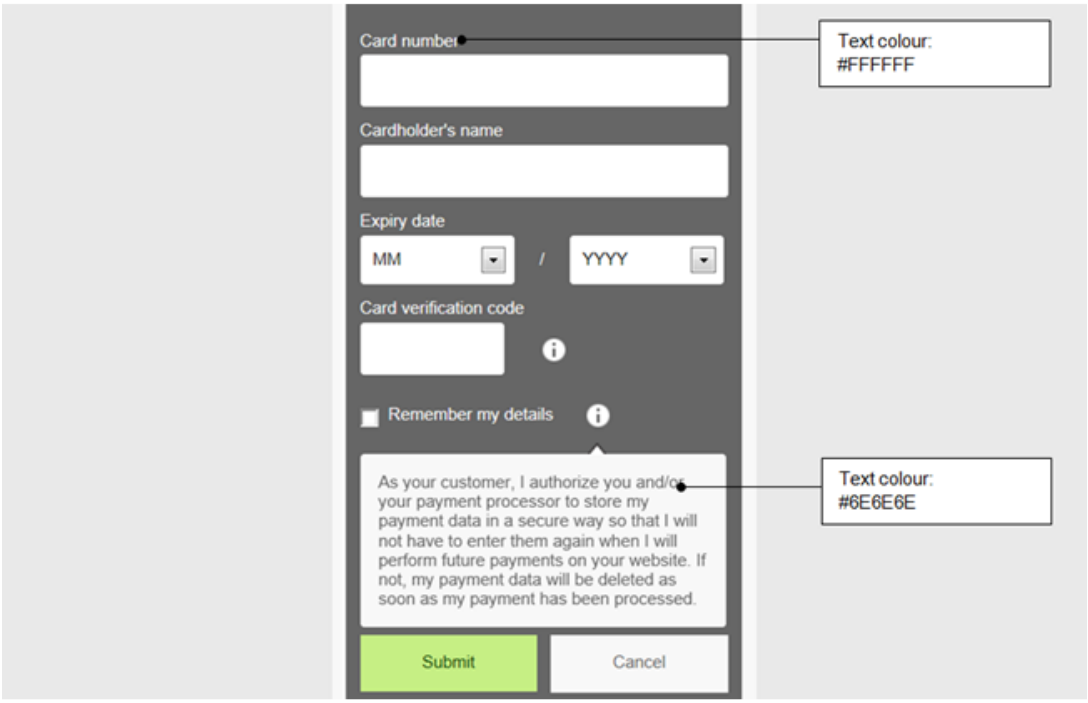


Impacted CSS:

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

Text color

© 2019 PostFinance

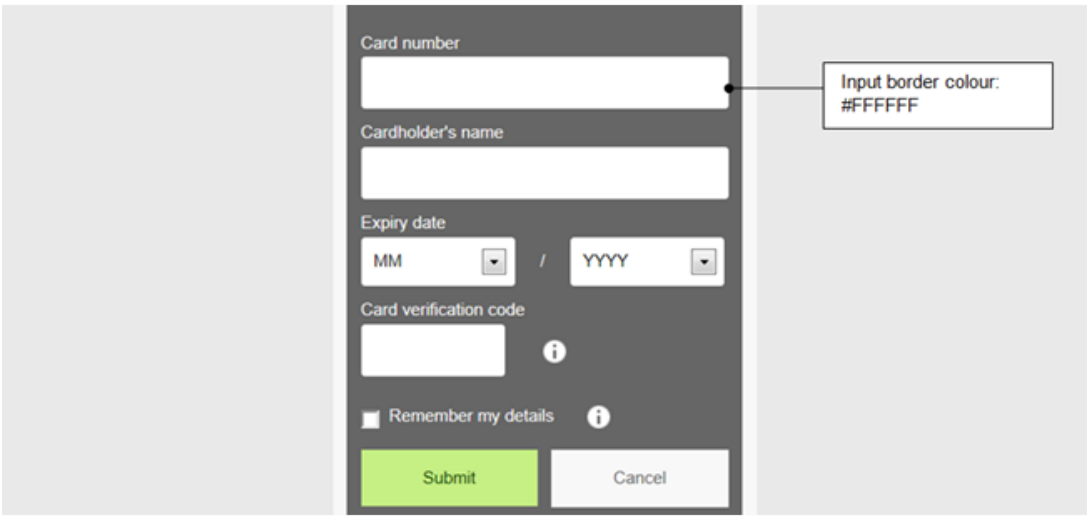


>
Impacted CSS:

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

Input border color

© 2019 PostFinance

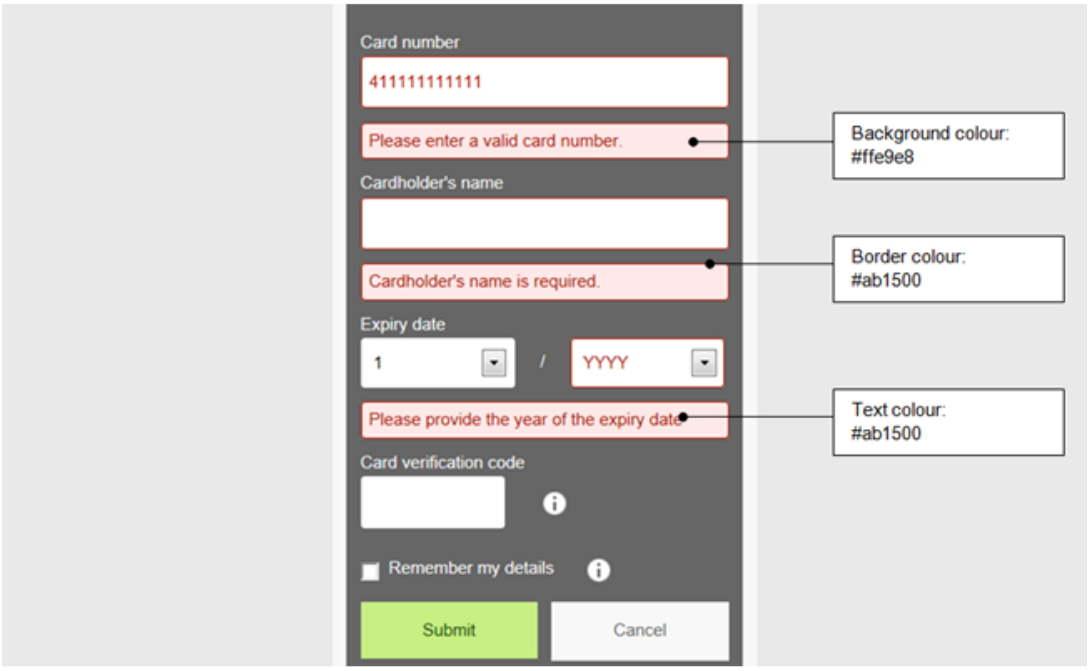


Impacted CSS:

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

Validation message box

© 2019 PostFinance

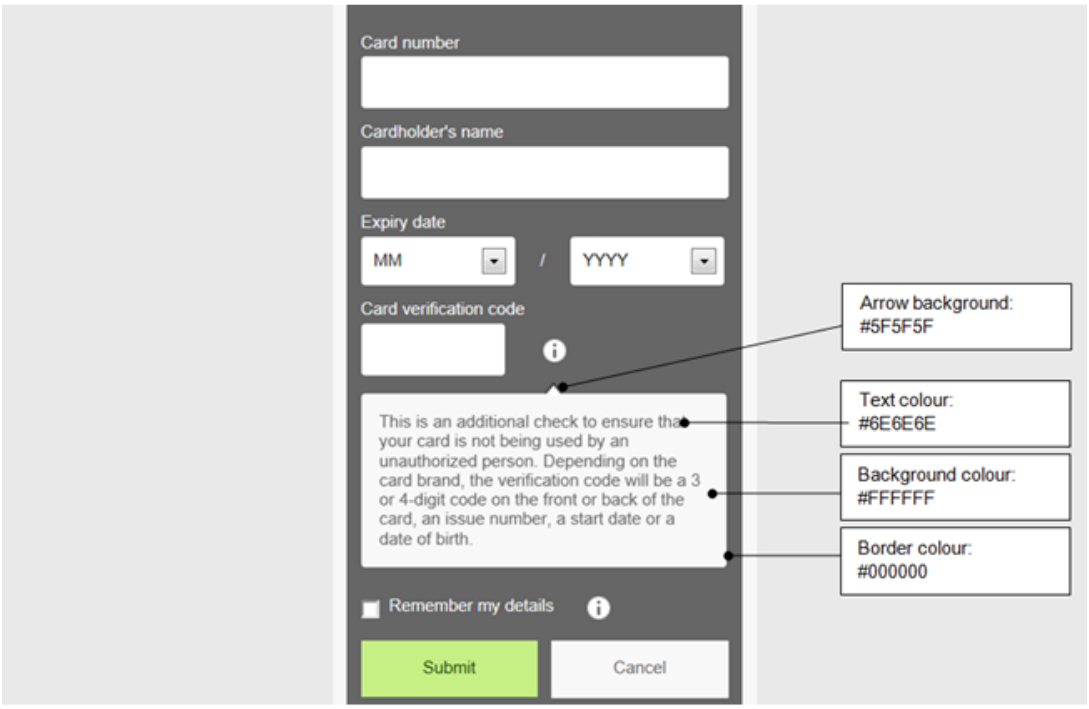


Impacted CSS:

.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;}

Help information box

© 2019 PostFinance

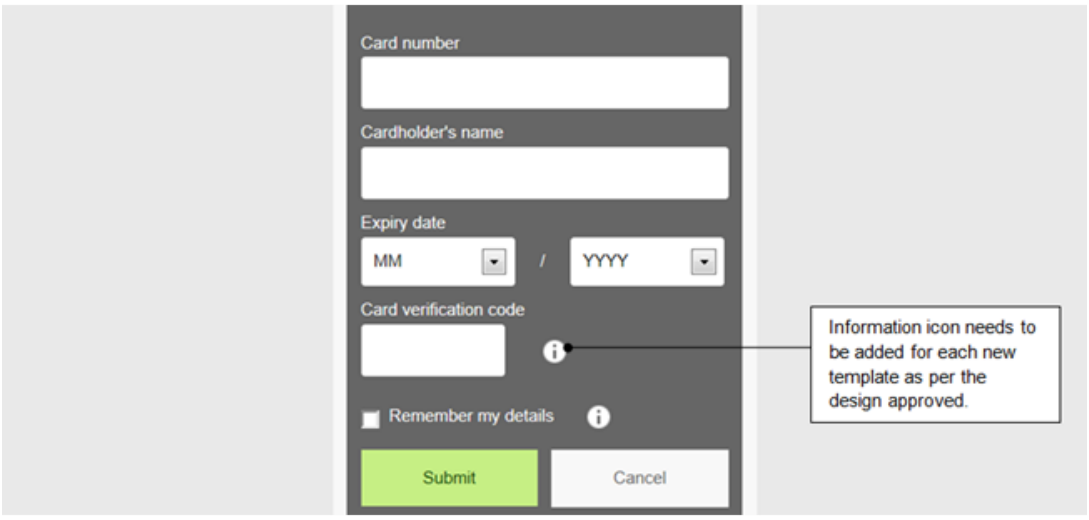


Impacted CSS:

#payment-cvc-info-container span.aroowImg {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;}

Information icon

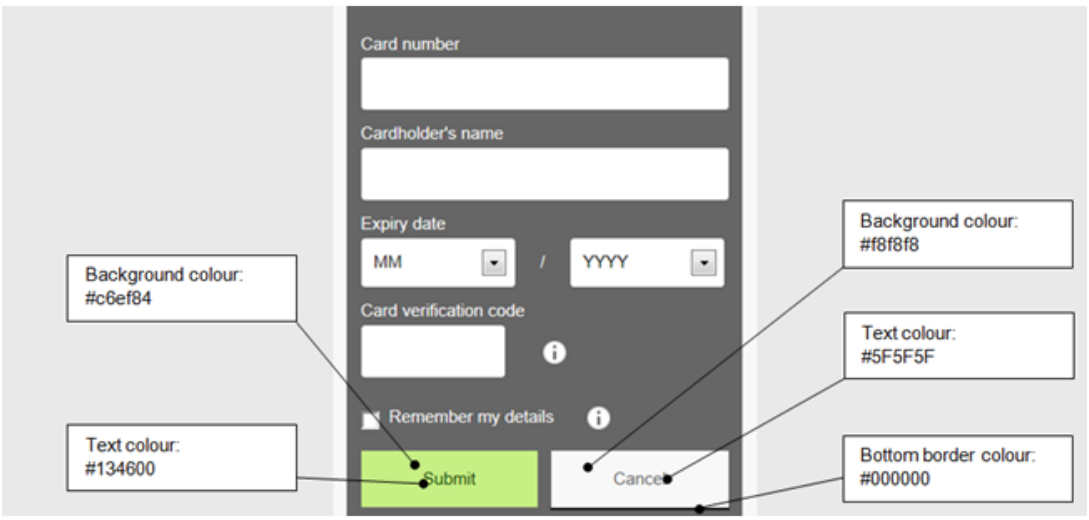
© 2019 PostFinance



Impacted CSS:

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

Submit/Cancel button

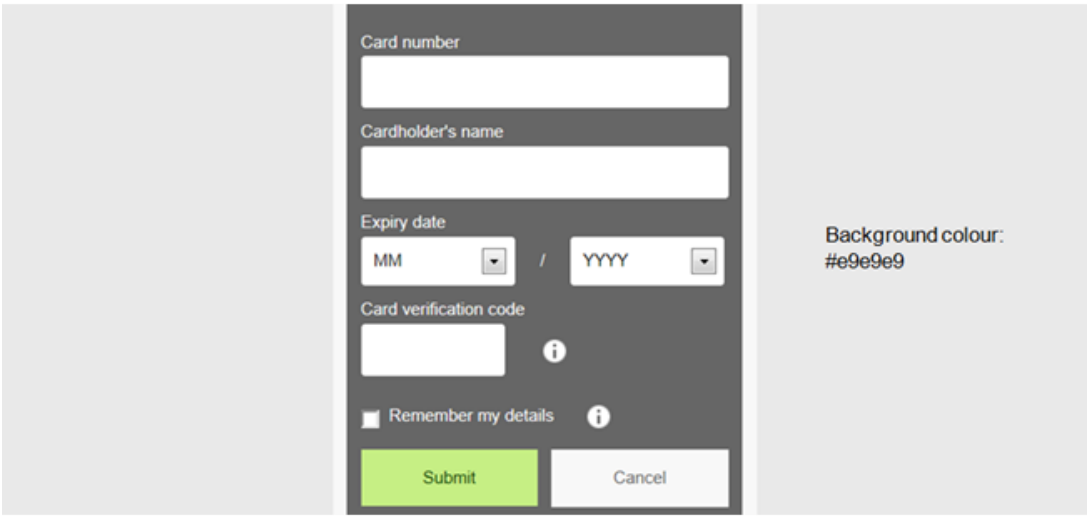


Impacted CSS:

© 2019 PostFinance

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

Background color



Impacted CSS:

#payment-container {background:# e9e9e9}

CSS examples

```
#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.aroowImg {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
```


© 2019 PostFinance

```
no-repeat;}

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

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

4.2 Advanced customization

For more flexibility, you can now create and customize master template and partial templates. Depending on your needs, the partial templates are created and customized separately. You can also customize the text content with XML files.
You can **either** choose to apply a $$$PAYMENT ZONE$$$ tag which calls for a simple template, or customize your own partial template by using placeholders.

4.2.1 Using placeholders

You can insert placeholders into master.html to customize and format your own master template (master.html) and partial templates.

You can also define the payment page appearance via a custom CSS file (**newuser.css**) that must be linked to the master.html. If you choose to customize your CSS file, the modified CSS file should be uploaded to the Template File Manager and set as a default.

Placeholders are categorized in three groups: Main, Element, and Grouped placeholders.

Main placeholders

The following is a list of main placeholders:

- $$$PAYMENT ZONE$$$: Renders a request brand or payment method either through a custom template (with an available partial template) or the default template.
- $$$TP RESOURCES URL$$$: Make available the URL to the directory where the files in the Template File Manager are found.

You also have the option of using custom scripts that will need to be linked to a custom JavaScript file. This JavaScript file can then be referenced once they are uploaded to the Template File Manager. For example, the referenced JavaScript file in <script type="text/javascript" src="$$$TP RESOURCES URL$$$/existinguser.js"></script> is **existinguser.js.**

To use JavaScript libraries in the templates (e.g., JQuery), you should upload the JavaScript files to the Template File Manager. Once uploaded, use $$$TP RESOURCE URL$$$ to reference the scripts. This step should be executed for any files referenced in the template: CSS, images, fonts, and so on, and will ensure that the executed script is the dedicated JavaScript.

Note: Not all data can be pushed to/pulled from a PCI non-compliant third-party host, so the use of JavaScript will impact merchants' eligibility to PCI SAQ A.

- $$$DISABLE MASKING$$$: Disables an input box masking.
- $$$SUBMIT$$$: Renders a submit button for the form. For example:

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

- $$$CANCEL$$$: Renders a cancel button for the form. For example:

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


© 2019 PostFinance

A master template containing $$$PAYMENT ZONE$$$ will not need to include submit and cancel placeholders. However, submitted and canceled placeholders must be included if a partial template is built per payment method, or if a master template is built without $$$PAYMENT ZONE$$$ placeholder.

Element placeholders

The actual control <element> is rendered inside an element container: <div id="payment-<input_name>-<element>-container" class="payment-<element>">. For easy CSS styling, the same naming convention applies to ID and class. Element placeholders render html controls for different input elements and should follow this format:

- $$$<input_name><element>$$$: The card number input control.
- $$$CARD NUMBER.INPUT$$$ - CARD NUMBER: The <input_name> where INPUT is the html <element>.
- $$$<input_name>LABEL$$$: Renders a label for the <input_name> element.

For example, the $$$CARD NUMBER LABEL$$$ placeholder renders this:

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

For all labels a div will be rendered to act as a container:

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

The class for a label container will always be “payment-label” so the labels can be styled together.

A label for a label control will always be rendered:

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

To style a label control, apply the CSS file to the container ID, not to the control ID.

$$$<input_name> INPUT$$$ renders the input control for <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>
```


© 2019 PostFinance

To style an input control, apply the CSS file to the container ID, not to the control ID.

$$$<input_name> INFO$$$ renders a link to a description of the <input_name> field:

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

Bind any information for this field in the selected language.

$$$<input_name> ERROR$$$ renders a container for error messages from <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>
```

Validation messages can be inserted in here.

Grouped placeholders

The element placeholders can be inserted as a group using grouped placeholders. The rendered container with a grouped placeholder: <div id="payment-<input_name>-container"> is where the element placeholders will be inserted and grouped. Effectively, it render this:

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

A master template can have placeholders inside . The placeholders (such as $$$CARD NUMBER$$$, $$$CARDHOLDER NAME$$$, $$$EXPIRY DATE$$$, and so on) define the content of the credit card form. You can select the placeholders that best meet your needs.

The following is an example of a master template with placeholders for a credit card form labeled “paymentForm”:

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
```


© 2019 PostFinance

```
<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>
```

To create a direct debit form, you can use direct debit placeholders such as the one below:

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

The bank account name, IBAN number, and BIC code will render a direct debit form.

If you want a template that supports multiple payment methods, you can build a master template and a partial template for the payment methods.

A master template is as follows:

```
<!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>
```


© 2019 PostFinance

A partial template, built with placeholders ($$$CARD NUMBER$$$, $$$CARDHOLDER NAME$$$, $$$EXPIRY DATE$$$, and so on), is as follows:

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

If you do not want to render the default payment form using $$$PAYMENT ZONE$$$, a partial template can be inserted instead.

For example, if the requested payment method is a credit card, a partial template by the name of master_creditcard.htm will be used. For a direct debit, a partial template by the name of master_directdebit.htm will be used instead.

The partial template will need to upload onto the File Manager. A naming convention must also be strictly followed. For example, if the existing master template is named master.html and is sent as a direct debit (payment method), then a partial template named master_directdebit.html will need to be uploaded to the File Manager.

4.2.2 Customizing text content with XML

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.

The elements must also be present to identify the fields in the XML file. Please click here for a complete list of elements. The following is an example of an element (CardHolderName) for the input of a cardholder name:

```
<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 this element, US, English, and French are specified in the Language ID. The ID tag also specifies the ISO culture code. If no elements are specified, the default text for that element will be used instead. And if the language is not specified for an element, the default language will be used.

Note:
The first line on the XML file must always start with <root> and the last line must always end with </root>.

© 2019 PostFinance

4.3 Template File Manager

With the Template File Manager you can easily manage your templates and the various files that come with it.

To start using the File Manager, log on to your PostFinance account and go to "Configuration" > "Template" > "File Manager".

4.3.1 Upload template files

Under "Upload Template Files", select the "Files..." button to browse for the files you wish to upload. You can upload css and images (.css, .jpg, .jpeg, .gif, .png), with a maximum of 7 MB per file, and 10 MB in total.

Make your selection and then confirm.

4.3.2 Check and manage uploaded files

After the upload is done, you'll see your uploaded files on the same page in the "Uploaded files" section.

The files will first have the "Validating" status, during which some necessary security/virus checks are being performed (this can take several minutes).

You can use the files once the status has changed to "Validated".

Click the refresh button  to check the current status of your files / click the  button to delete the file permanently.

A file will get the "Refused" status if it didn't pass the security check. This can be due to a virus or if the extension of the file is wrong for instance.

4.3.3 Integration

In your code you refer to your uploaded files with a code following this structure: $$$TP RESOURCES URL$$$/[your file name].

© 2019 PostFinance

5. List of placeholders and XML elements

The following is a full list of placeholders and their purpose:

Placeholders

Main Placeholders	**Description**
$$$PAYMENT ZONE$$$	A custom template from a partial template (if available) or a default template that renders the requested brand or payment method requested.
$$$TP RESOURCES URL$$$	Calls for a file that has previously been uploaded to the Back-Office.
$$$DISABLE MASKING$$$	Renders a text box that matches to the card types and input formatting. Disables the input box masking upon detection of the card type and formatting of input boxes based on the card brand.
$$$SUBMIT$$$	A submit button that must be present on the form.
$$$CANCEL$$$	A cancel button on the form.
$$$CARD NUMBER$$$	A credit card number with brand appropriate validation and masking.
$$$CARD NUMBER LABEL$$$	An indication for a credit card number.
$$$CARD NUMBER INPUT$$$	The field where a card number is inputted.
$$$CARD NUMBER INFO$$$	The field where a card number information is provided
$$$CARD NUMBER ERROR$$$	An error message that displays when a wrong card number is entered incorrectly.
$$$CARDHOLDER NAME$$$	A name of a credit cardholder.
$$$CARDHOLDER NAME LABEL$$$	An indication for a cardholder name.
$$$CARDHOLDER NAME INPUT$$$	The field for a cardholder name to be inputted.
$$$CARDHOLDER NAME INFO$$$	The field for a cardholder name information to be inputted.
$$$CARDHOLDER NAME ERROR$$$	An error message that displays when a cardholder name is entered incorrectly.
$$$EXPIRY DATE$$$	The area where an expiry date is displayed.
$$$EXPIRY DATE LABEL$$$	An indication for an expiry date.
$$$EXPIRY DATE INPUT$$$	The area where an expiry date is inputted.
$$$EXPIRY DATE MONTH$$$	The month of the expiry date.
$$$EXPIRY DATE YEAR$$$	The year of the expiry date.
$$$EXPIRY DATE INFO$$$	The field where an expiry date information is provided
$$$EXPIRY DATE ERROR$$$	An error message that displays when an expiry date is entered incorrectly.
$$$CVC$$$	The field where a CVC with its appropriate validation is placed.
$$$CVC LABEL$$$	An indication for CVC.
$$$CVC INPUT$$$	The field where a CVC is inputted.
$$$CVC INFO$$$	The field to display the CVC information.
$$$CVC ERROR$$$	An error message that displays when a CVC is entered incorrectly.
$$$BANK ACCOUNT$$$	An account number for direct debits forms.
$$$ACCOUNT OWNER$$$	The area where an account owner is displayed.
$$$ACCOUNT OWNER LABEL$$$"	An indication for an account owner.
$$$ACCOUNT OWNER INPUT$$$	The field where an account owner is inputted.
$$$STORE PERMANENTLY$$$	A checkbox for the merchant to either opt- in or -out or permanently store the alias.
$$$BIC$$$	A BIC box for direct debits aliases.

XML elements

Alias form	• PageTitle • Submit • Cancel • RememberMyDetails • AliasAuthorizationInfo
Credit cards	• PageTitle • CardNumber • CardNumberIsRequired • InvalidCardNumber • CardNumberTooLong • InvalidCardNumberCharacters • CardHolderName

© 2019 PostFinance

	• CardHolderNameIsRequired • InvalidCardHolderName • CardHolderNameTooLong • ExpiryDate • InvalidExpiryDate • ExpiryMonthIsRequired • InvalidExpiryMonth • InvalidExpiryMonthLength • InvalidExpiryMonthCharacters • ExpiryYearIsRequired • InvalidExpiryYear • InvalidExpiryYearLength • InvalidExpiryYearCharacters • Cvc • CvcIsRequired • InvalidCvcNumber • CvcTooLong • CvcContainsNonNumericCharacters
CVC information	• VisaCvcInfo • EurocardCvcInfo • MasterCardCvcInfo • DinersClubCvcInfo • DankortCvcInfo • LaserCvcInfo • MaestroCvcInfo • JcbCvcInfo • AmericanExpressCvcInfo • AuroreCvcInfo • SurcoufCvcInfo • PrintempsCvcInfo • ClubMedCvcInfo • OkShoppingCvcInfo • MandarineCvcInfo • KangourouCvcInfo • GoSportCvcInfo • FinarefCvcInfo • AlsoliaCvcInfo • DefaultCvcInfo
Direct debits	• AccountOwnerIsRequired • AccountOwnerTooLong • InvalidAccountOwner • AccountIsRequired • AccountTooLong • InvalidAccount • BicIsRequired • BicTooLong • InvalidBic • GiroAccount • BicName • Iban • BankAccount • InvalidIbanCharacters • InvalidIbanStructure • InvalidCheckDigits • InvalidIbanCountry • InvalidIbanLength • InvalidIbanNumber • InvalidBankAccountNumber • IbanOrBankAccount • IncorrectBic • BlzIsRequired

© 2019 PostFinance

© 2019 PostFinance