Manual for the "PayU Integration" module

Connecting to the PayU account

In order to properly configure the module you must obtain a merchant account at PayU. Obtaining the account is free but PayU takes a small fee for each transaction.

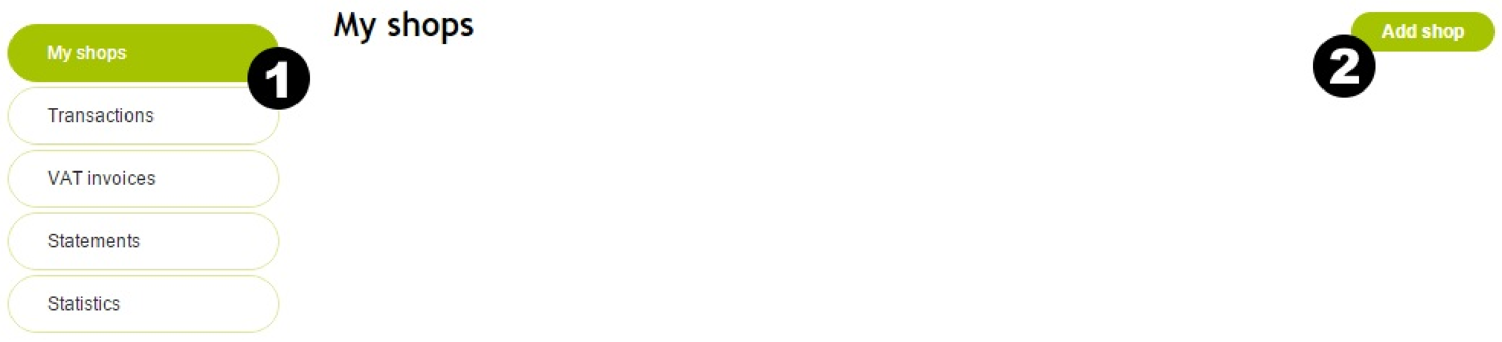
> Do not confuse a regular PayU account, which is used for making payments for your purchases, with the merchant account. The merchant account used by the module requires signing an agreement with PayU.

In order to configure the module you must have the POS data which is available in the PayU administrative panel.

If in the past you have been using PayU in your store, it is possible to reuse the same POS but only if it is of type Checkout (REST API). The Payments (Classic API) POS cannot be used by this module.

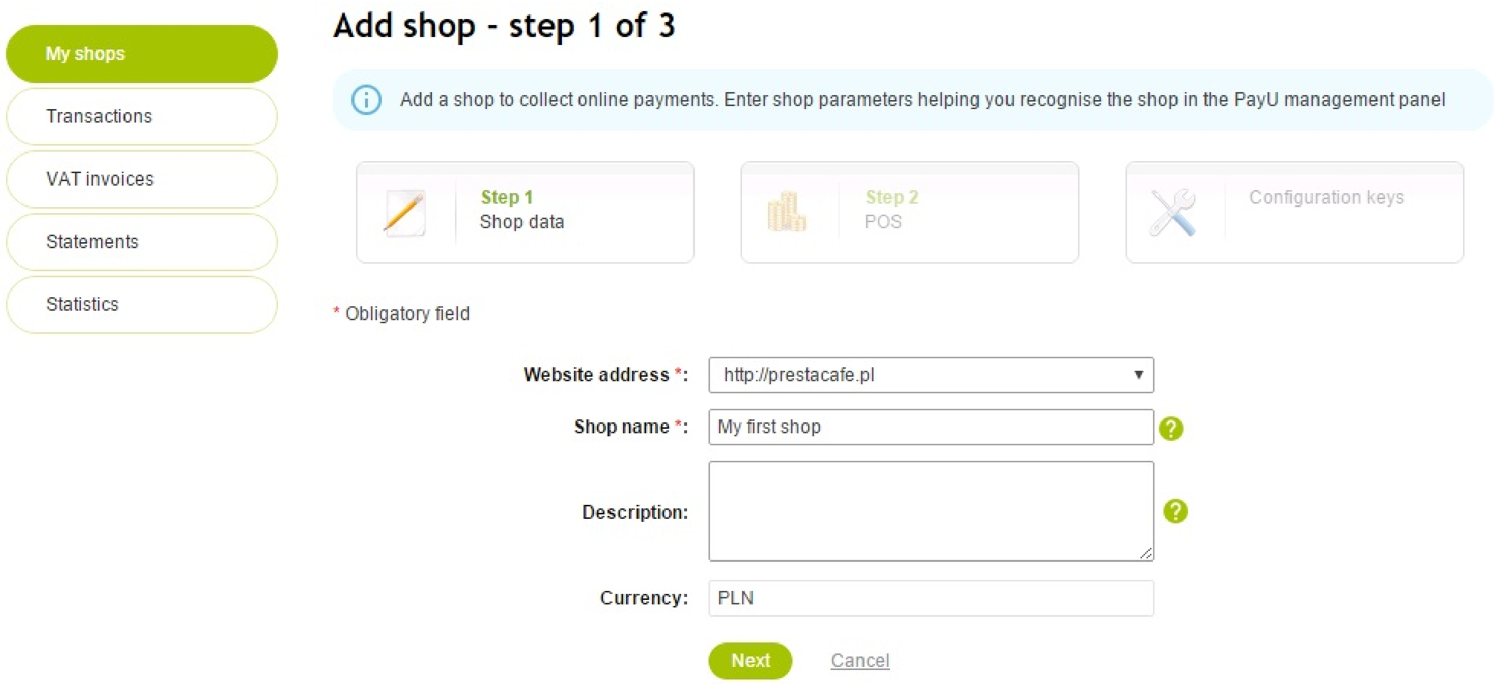
If there is no POS configured or if you are unsure of its type, create a new POS by following the instructions below.

Click on My shops and then on the Add shop button



\

Fill out the basic shop data: select the shop's url from the dropdown list (the same url that you declared while signing the agreement with PayU) and enter the shop's name. Click on the Next button



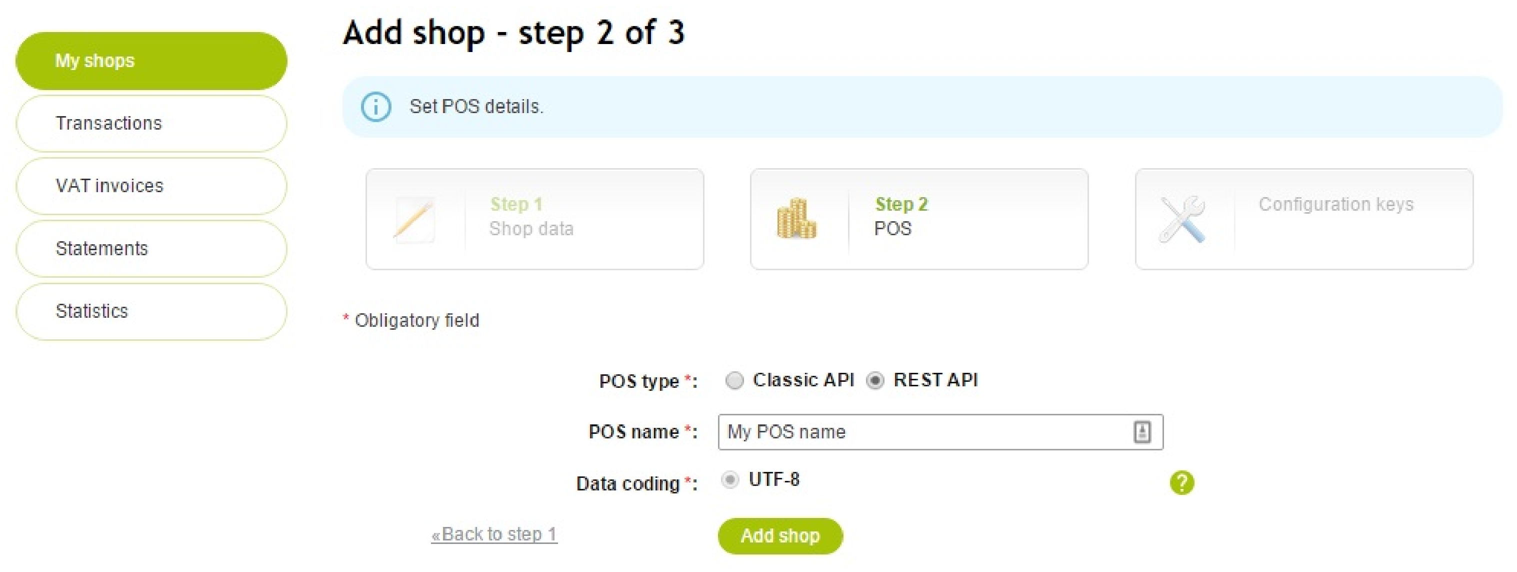
Add a shop to collect online payments. Enter shop parameters helping you recognise the shop in the PayU management panel

* Obligatory field

Website address *:	http://prestacafe.pl
Shop name *:	My first shop
Description:	
Currency:	PLN

\

In the next step select the POS type to be REST API and enter your name for the new POS. Click on the Add shop button



Add shop - step 2 of 3

Set POS details.

* Obligatory field

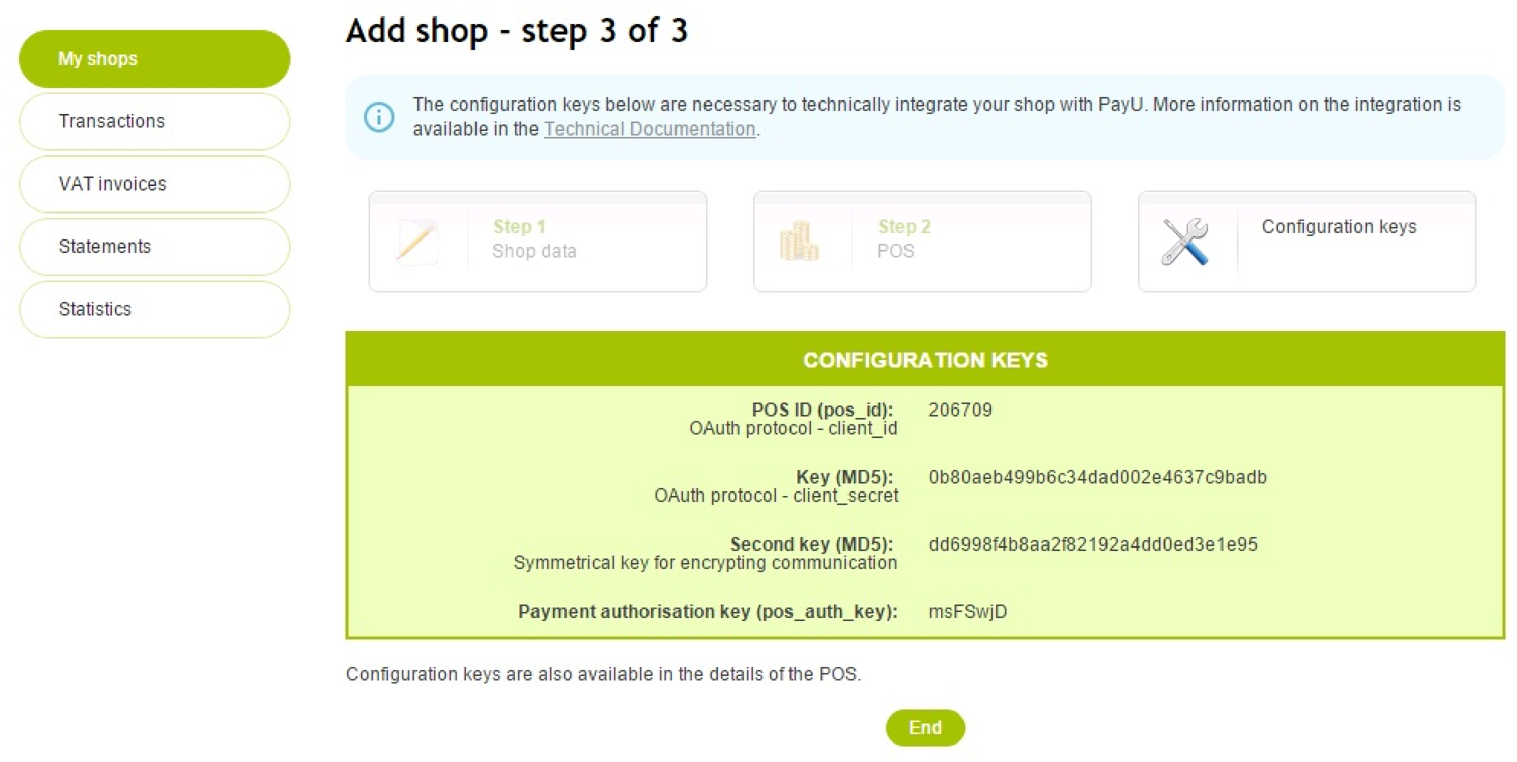
POS type *: Classic API [✓ REST API]

POS name *: My POS name

Data coding *: [✓ UTF-8]

\

After adding the shop, its configuration keys are shown. Copy them field by field to the module's configuration



Add shop - step 3 of 3

The configuration keys below are necessary to technically integrate your shop with PayU. More information on the integration is available in the Technical Documentation.

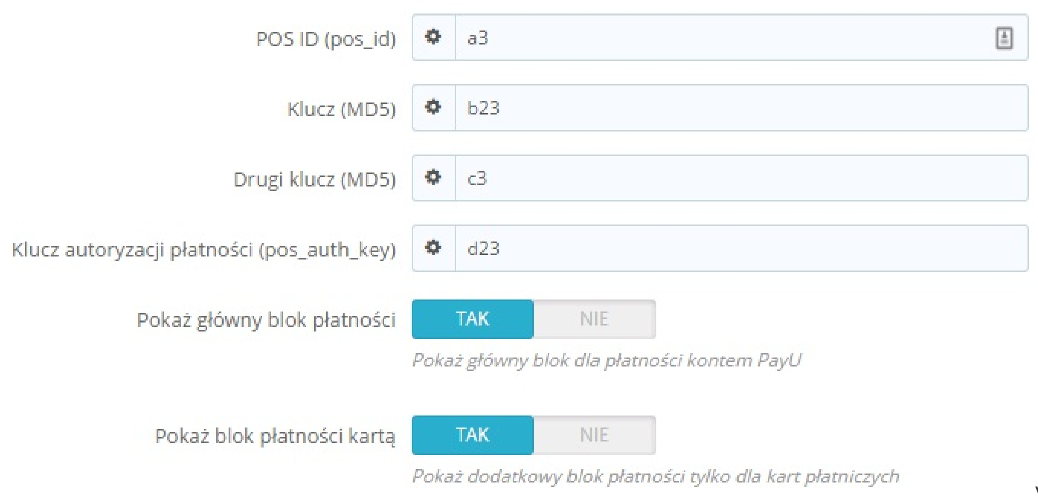
CONFIGURATION KEYS

POS ID (pos_id): OAuth protocol - client_id	206709
Key (MD5): OAuth protocol - client_secret	0b80aeb499b6c34dad002e4637c9badb
Second key (MD5): Symmetrical key for encrypting communication	dd6998f4b8aa2f82192a4dd0ed3e1e95
Payment authorisation key (pos_auth_key):	msFSwjD

Configuration keys are also available in the details of the POS.

\

Save the module configuration



POS ID (pos_id)	a3
Klucz (MD5)	b23
Drugi klucz (MD5)	c3
Klucz autoryzacji płatności (pos_auth_key)	d23
Pokaż główny blok płatności	[✓ TAK] NIE
	Pokaż główny blok dla płatności kontem PayU
Pokaż blok płatności kartą	[✓ TAK] NIE
	Pokaż dodatkowy blok płatności tylko dla kart płatniczych

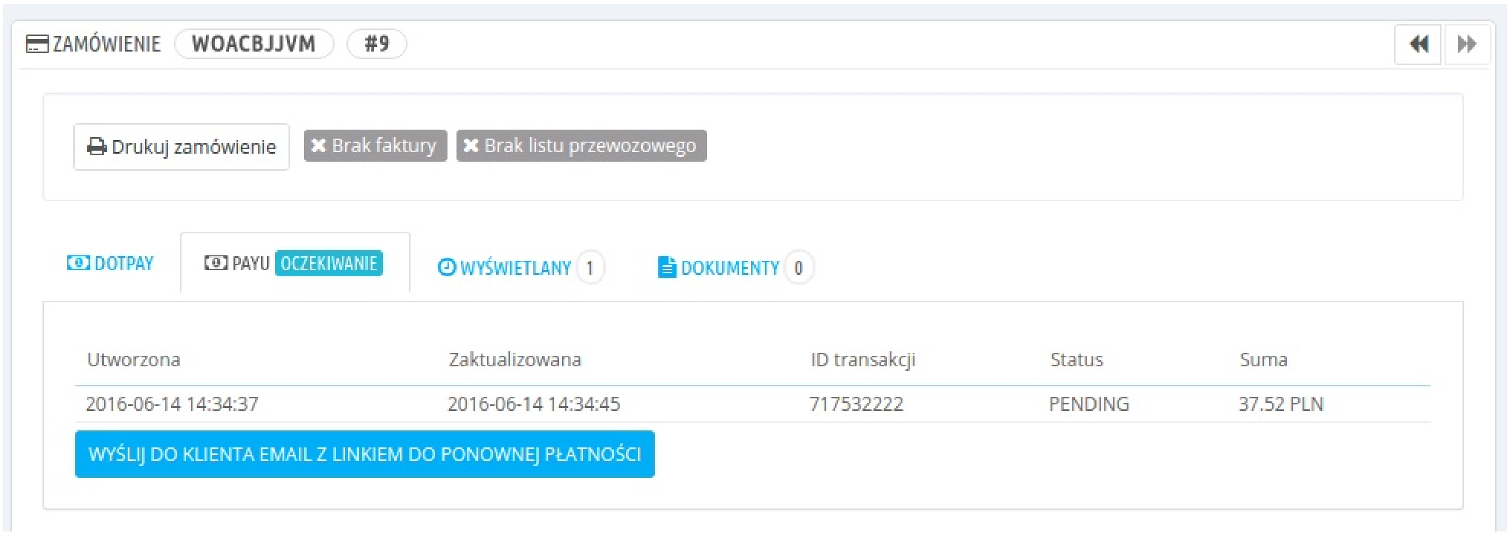
\

Repeat payment

One the biggest grudges the Prestashop merchants have is the process of renewing an order if the customer fails to pay at the first try. If the cart is cleaned out by a payment module, subsequent payment requires creating a duplicate of the order and cancelling the original.

Our module lets the customers to restart payment in PayU without creating a new order.

When the customer is redirected to the PayU site, they receive an email PayU payment has started in which they can find a link to pay for the order. If the customer happens to accidentally close the browser or they do not succeed at paying for any other reason, they can use this link to go back to PayU and pay for the order again. The merchant can also send such email by clicking a button in the order interface.



Utworzona	Zaktualizowana	ID transakcji	Status	Suma
2016-06-14 14:34:37	2016-06-14 14:34:45	717532222	PENDING	37.52 PLN

\

The module will not allow paying double for an order. Before repeat payment is allowed, the previous payment must be cancelled. If the PayU service refuses to cancel the payment, the repeat payment cannot proceed.

Automatic order state changes

The PayU module changes the orders' states based on the notification received from PayU. For the notifications to be able to reach your store, the store must not be in maintenance mode. It also cannot be password protected.

If your store is under development and not ready to be made public, you should enable the following IP addresses to access your store. These are PayU notification IP addresses:

- 185.68.12.10
- 185.68.12.11
- 185.68.12.12
- 185.68.12.26
- 185.68.12.27
- 185.68.12.28

Templates displayed by the module

Introduction

Each template is both Bootstrap-based for Prestashop 1.6+ and classic-style for Prestashop 1.5.

The conditional display of the template contents depending on Prestashop version is achieved by examining the constant $smarty.const._PS_VERSION_ like so:

```
{if version_compare($smarty.const._PS_VERSION_, '1.6', '<')}
   {* here goes content for Prestashop 1.5 *}
{else}
   {* here goes content for Prestashop 1.6 *}
{/if}
```

Any changes to the templates should not be made directly on the module files. Any *.tpl file has its counterpart in the directory tree based in the theme directory.

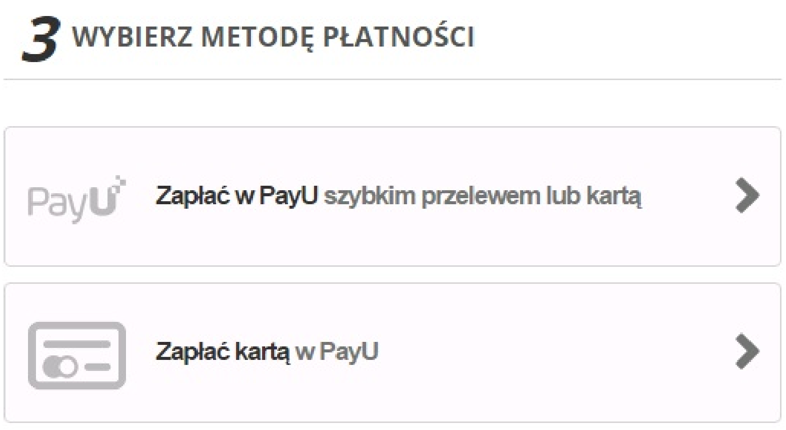
For example, if you want to modify the `views/templates/hook/payment.tpl` template, copy it to the themes/SZABLON/modules/prestacafenewpayu/views/templates/hook/payment.tpl directory and modify the copy.

More information about template overriding can be found in the Prestashop documentation at Overriding module's behavior

Template details

views/templates/hook/payment.tpl

This template displays the payment block as the last step of placing the order. It displays both regular and card payment blocks.


3 WYBIERZ METODĘ PŁATNOŚCI

Zapłać w PayU szybkim przelewem lub kartą

Zapłać kartą w PayU

\

views/templates/hook/payment_return.tpl

This template is displayed as part of the standard `order-confirmation.tpl` template (paymentReturn hook) when the customer returns to the store after successful payment.

Exception: if the module knows the payment did not go through when the customer returns, the `views/templates/front/confirmation_try_again.tpl` template is displayed instead.

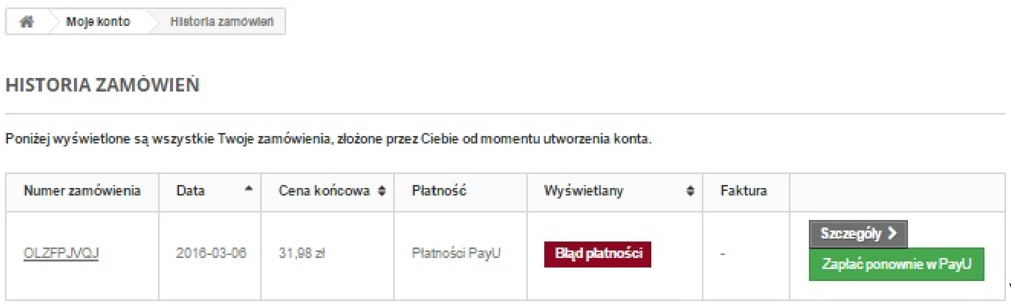
views/templates/front/history_pay_again.tpl

This template displays the button Pay again in the order history. For this button to be displayed you must yourself modify the standard `history.tpl` template and add a code snippet in one of the order table columns, for example the order state column.

The snippet to add in history.tpl is:

```
{PrestaCafeNewPayu::historyPayAgain($order)}
```

and it must be inside the `{foreach}` loop that iterates over the $orders collection.


Moje konto > Historia zamówień

HISTORIA ZAMÓWIEŃ

Poniżej wyświetlone są wszystkie Twoje zamówienia, złożone przez Ciebie od momentu utworzenia konta.

Numer zamówienia	Data	Cena końcowa	Płatność	Wyświetlany	Faktura	
OLZFPJVQJ	2016-03-06	31,98 zł	Płatności PayU	Błąd płatności	-	Szczegóły > Zapłać ponownie w PayU

\

views/templates/front/payagain_already_paid.tpl

This template is displayed when the customer attempts to pay again for an order that has already been paid. This may happen after inadvertent clicking on the Pay again link received by email after paying for the order.

views/templates/front/payagain_payment_never_started.tpl

This order is displayed in the rare occasion that the PHP script execution ends in error during payment, or the database connection is broken but the Pay again link has already been emailed to the customer. In this case there is no PayU transaction in the database (and no order either).

views/templates/front/payment_already_placed.tpl

This template is displayed when the customer clicks on the PayU payment block but the order for the current cart has already been placed, possibly in another browser window.

views/templates/front/confirmation_try_again.tpl

This template is displayed after the customer returns from PayU and the module knows for sure that the payment did not go through. The customer is prompted to retry the payment.

views/templates/front/payagain_cannot_cancel.tpl

Before paying again the previous payment must be unconditionally canceled regardless of its state. If cancelling the transaction does not succees, the customer cannot proceed with payment for security reasons and this template is displayed.

views/templates/front/payment_try_again.tpl

When the customer clicks on the PayU payment blocks a connection to PayU is made directly by the module in order to create a payment. If this connection fails, this template is displayed allowing the customer to retry payment.

views/templates/front/payagain_try_again.tpl

Same as `payment_try_again only.tpl` but for repeat payments.