

Payment methods

- Overview
- Payment Gateway Management
 - Payment gateway eligibility
 - Payment gateway sorting
 - Workshops
 - Authorize.net
 - AIM
 - SIM
 - CyberSource
 - PayPal
 - PayPal Button
 - PayPal Express
 - PayPal Pro
 - IPN
 - LiqPay
 - LiqPay (no refund)
 - LiqPay
 - PostFinance
 - PostFinance e-payment
 - PostFinance e-payment (manual capture)
 - PostFinance Checkout (V2) SaaS 4.0.0
 - SwissBilling SaaS 3.7.0
 - SwissBilling
 - PaySera SaaS 3.7.0
 - PaySera Checkout
- Test cards and addresses
- Overview
- Payment Gateway Management
 - Payment gateway eligibility
 - Payment gateway sorting
 - Workshops
 - Authorize.net
 - AIM
 - SIM
 - CyberSource
 - PayPal
 - PayPal Button
 - PayPal Express
 - PayPal Pro
 - IPN
 - LiqPay
 - LiqPay (no refund)
 - LiqPay
 - PostFinance
 - PostFinance e-payment
 - PostFinance e-payment (manual capture)
 - PostFinance Checkout (V2) SaaS 4.0.0
 - SwissBilling SaaS 3.7.0
 - SwissBilling
 - PaySera 4.1.0
 - PaySera Checkout
- Test cards and addresses

Overview

Payment methods refer to choice of payment options available to customer when placing an order. In most cases payment method relies on a third party system (payment gateway) to complete the financial transaction. Although financial operations are fairly standardised and well understood each payment gateway offers unique API to accomplish this. Therefore payment API is composed of two parts:

- **core payment API**, which is fully integrated with [order life cycle](#), dealing with standard approaches to transactions
- **payment modules**, which are payment gateway specific implementations that drive the standard operations.



The platform has a number of payment modules implementation provided out of the box with flexibility to add new implementations with ease.

Once the platform up and running all **payment modules are automatically detected and contribute to the payment gateways configuration**. Through system payment gateway panel business user can enable and disable payment gateways at the platform level. Shop

managers when configuring [shop](#) can choose from the list of platform enabled payment gateways and configure them with shop specific parameters.

Payment gateway parameters are predefined by the payment module, so all that is necessary to enable payment methods for shop is to setup these values, which usually involves entering merchant keys or IDs.

The following payment gateway modules are supported out of the box (with few featured highlighted):

Module	Payment method	version	EoL	Online	External	Callback	Per Shipment	AUTH	CAPTURE	AUTH_C
Core										
	Payment to courier	1.0.0+		offline			✓	✓	✓	
	Pre paid (external payments)	1.0.0+		offline						
	In store	1.0.0+		offline			✓	✓	✓	
	Test (card)	1.0.0+		online mock			✓	✓	✓	
	TestExt (card)	1.0.0+		online mock	✓	base callback filter	✓	✓	✓	
	Invoice	3.5.0+		offline						
	Invoice with Authorisation	3.5.0+		offline			✓	✓	✓	
	No payment required	3.7.0+		online						
Authorize.net										
	AIM	2.0.0+		online			✓	✓	✓	
	SIM	2.0.0+		online	✓	AuthorizeNetSimPaymentOkPage				
CyberSource										
	CyberSource	2.0.0+		online			✓	✓	✓	
PayPal										
	PayPal Button	3.1.0+		online	✓	base callback filter				
	PayFlow 	2.0.0+	3.0.0	online			✓	✓	✓	
	PayPal Express	2.0.0+		online	✓	PP express callback filter (set mode, tx confirmation)				
	PayPal NPV 	2.0.0+	3.0.0	online			✓	✓	✓	
	PayPal Pro	3.1.0+		online			✓	✓	✓	
LiqPay										
	LiqPay (full)	2.0.0+		online	✓	base callback filter				
	LiqPay (no refund)	2.0.0+		online	✓	base callback filter				
PostFinance										
	PostFinance e-payment	3.1.0+		online	✓	base callback filter				
	PostFinance checkout	4.0.0+ Sa aS		online	✓	PF callback filter (session setup, tx verification)				
Swissbilling										
	Swissbilling	3.7.0+ Sa aS		online	✓	SB callback filter (pre-screening, tx confirmation)				
PaySera										
	PaySera checkout	4.1.0+		online	✓	PaySera callback filter (enhanced version of base filter that also send "OK" in response)				

Payment Gateway Management

Payment gateways list loaded into this panel is composed of automatically resolved payment gateways modules installed on current instance. The installation process is very simple and involves specifying few maven profile parameters during [build](#).

When this list is loaded system administrator can select which modules are enabled or disabled by clicking "on" and "off" buttons. If a payment gateway does not appear in this list then it is highly possible that this module was not added to the build, otherwise all payment gateways will appear in this list loaded from bundled modules.

Each payment gateway has predefined attributes, which can be viewed by clicking "Attributes" button when payment gateway is selected. Note that **these parameters are templates** - the **actual values are set when configuring payment gateway for [shop](#)**.

There is however an option to add additional attributes to template by clicking "New attribute" button. This allows to add new attributes such as new localisation for HTML form when new language is added.

After new attribute is added to the system payment templates the **shop payment gateway has to be turned "off" and then back "on" for new attribute to appear in shop** specific settings.

As of 3.4.0+ this view uses "Secure" feature, which means that by default only non-secure parameters are loaded. If you want to see all parameters you have to click the "lock" button. Then you will see all attributes available for payment gateway in the attribute view.

DEMO ENVIRONMENT

Powered by
YC - pure Commerce

Logged in as Yes

Customer service

PIM & Master Catalog

CMS

Marketing & Pricing

Inventory

Reports

Shipping

Shops

Organisation

System

Attributes

Import

Export

Search Indexes

Payment Gateways

Unprocessed Callbacks

Clone Shop

Preferences

Query API

Cache Monitoring

Performance Sensors

Performance Samples

Tasks

Cluster

Help

Payment Gateways / Authorize.net SIM

Secure mode (on/off)

Attributes

Enable & Disable Buttons

Features Information

Rank	Code	Name	Enabled
100	authInvoicePaymentGatewayLabel	Authorize invoice payment	<input type="checkbox"/>
100	authorizeNetAimPaymentGatewayLabel	Authorize.net AIM	<input type="checkbox"/>
100	authorizeNetSimPaymentGatewayLabel	Authorize.net SIM	<input checked="" type="checkbox"/>
100	cyberSourcePaymentGatewayLabel	CyberSource	<input type="checkbox"/>
100	invoicePaymentGatewayLabel	Invoice payment	<input type="checkbox"/>
100	liqPayPaymentGatewayLabel	LiqPay	<input type="checkbox"/>
100	liqPayNoRefundPaymentGatewayLabel	LiqPay (No Refund)	<input type="checkbox"/>
100	prePaymentGatewayLabel	Offline Pre-Payment	<input checked="" type="checkbox"/>
100	paypalButtonPaymentGatewayLabel	PayPal B...	<input type="checkbox"/>
100	paypalExpressPaymentGatewayLabel	PayPal E...	<input type="checkbox"/>

Attributes view

Authorize.net SIM

New Attribute

Edit Attribute

Remove Attribute Value

Attribute	Value
Gateway name (default)	Authorize.net SIM
Gateway name (EN)	Authorize.net SIM
Название платежного шлюза (RU)	Authorize.net SIM
Назва платіжного шлюзу (UK)	Authorize.net SIM
Gateway-Namen (DE)	Authorize.net SIM
Uri to post form.	https://test.authorize.net/gateway/transaction.dll
Relay response uri	http://@domain@/anetsimresult
SIM/DPM order receipt uri	http://@domain@/paymentresult?hnt=ok
SIM test request flag	FALSE
Gateway priority	100

These are DEFAULTS, they can be overwritten by shop specific values

Attribute

Attribute

Payment gateway eligibility

Payment gateways are directly influenced by shipping method configuration. The anticipation is that the kind of shipping method (i.e. carrier

SLA) is the main driver for the payment eligibility.

For example if the shipping method is "collect from shop" it would make sense to display "Payment in shop" payment method, but not for say "Home delivery".

Carrier SLA configurations contain selection for payment gateways available to those SLA.

As of 3.6.0 and additional customer level restriction is available for payment gateways via "tagging" mechanism. Payment gateways have "**restrictToCustomerTags**" parameter which is a CSV of eligible customer tags. When filled in this parameter which contain use of given payment method only to customer that have at least one matching tag.

Note that restrictToCustomerTags is CSV and expects values separated by comma (e.g. "tag1,tag2"), whereas customer tags property expects tags separated by space (e.g. "tag2 othertag tag4")

Payment gateway sorting

By default payment methods are *sorted alphabetically by their language specific name*. However if manual sorting is required then "**priority**" parameter can be used to sort them.

Note that if priorities of payment gateways are the same then sorting is done alphabetically. This is very useful when a particular payment method can be set higher priority to be at the top and then rest of the methods do not need to be adjusted and will be sorted in natural order.

Workshops

Activating payment modules (5:45)

[Play](#)

Part 1 of Enabling Payment Methods

Enabling payment methods in your shop (basics) (4:57)

[Play](#)

Part 2 of Enabling Payment Methods

Payment Gateway Specific Configurations

Authorize.net

To create test account go to http://developer.authorize.net/hello_world/sandbox/ and fill in the form.

After registration is completed you will be provided with:

1. **API Login ID**
2. **Transaction Key**
3. **MD5 Hash Key**

Enabling Authorize.NET (AIM & SIM)
(7:09)

[Play](#)

Part 3 of Enabling Payment Methods

You can login to your test account from here <https://sandbox.authorize.net>

Key points about your test account:

- **API Login ID** and **Transaction Key** can be viewed in "Settings > Security Settings > API Login ID and Transaction Key"
- **MD5 Hash Key** can be reset in "Settings > Security Settings > MD5 Hash"
- For SIM Receipt Page and Relay Response URL must be configured in "Settings > Security Settings > Receipt Page" and "Settings > Security Settings > Relay Response" respectively.
- Relay Response is what is used as transaction callback and its response is rendered as confirmation page back to the customer after clicking "Pay" button.

AIM

Key configurations for **AIM**

Configuration	Mandatory	Notes
Merchant login	✓	API Login ID from your Authorize.NET account
Transaction key	✓	Transaction Key from your Authorize.NET account
Environment name	✓	test - "SANDBOX" production - "PRODUCTION"

SIM

SIM account is currency specific, so you will not be able to provide different currency when paying with this method. You can configure the currency in the processor settings section

Key points for **SIM**

Configuration	Mandatory	Notes
Merchant login	✓	API Login ID from your Authorize.NET account
Transaction key	✓	Transaction Key from your Authorize.NET account
MD5 Hash Key	✓	MD5 Hash Key from your Authorize.NET account used to verify transaction callback
URL to post form	✓	test - " https://test.authorize.net/gateway/transact.dll " production - " https://secure.authorize.net/gateway/transact.dll "
Relay Response URL	✓	must be set to the " https://www.yourdomain.com/yes-shop/anetsimresult ". Note: that this page must be served via HTTPS and it processes the transaction callback . Note: "/anetsimresult" is mounted to AuthorizeNetSimPaymentOkPage in "wicket.xml"
SIM test request flag	✓	test transaction - TRUE actual transaction - FALSE
Payment form		There is a number of parameters that SIM supports to modify look and feel of the external payment form. Recommended "Order cancel URL" is " http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel ".

CyberSource

To create test account go to <http://www.cybersource.com/register/> and fill in the form.

After registration is completed you will be provided with:

1. **Organization ID**
2. Link to activate **Merchant Admin account**

Enabling CyberSource (7:23)

[Play](#)

Part 4 of Enabling Payment Methods

You need this account to generate the p12 key

3. Link to activate **Account Admin account**

You can login to your test account from here <https://ebctest.cybersource.com>

Key points about your test account:

- If you have capital letters in your Organization ID they will be converted to lower case (be aware of this)

- Generating p12 certificate is done from Login to **Merchant Admin** > Account Management > Transaction Security Keys > Security Keys for the Simple Order API
- You need to allow applet in order to save the certificate. Certificate name will be Organization ID with p12 extension.
- **p12 certificate is 2048-bit, so java SDK security must have "Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files" installed.**
- **on some JDK8 due to classloader bug you may need to copy bcprov-ext-jdk15on.jar into JAVA_HOME/jre/lib/ext** if you get "error constructing MAC:
java.security.InvalidAlgorithmParameterException: inappropriate parameter type:
javax.crypto.spec.PBEParameterSpec"
- Cybersource uses AVS, so address must be correct for payments to go through.

Key configurations for AIM

Configuration	Mandatory	Notes
Merchant id	✓	Organization ID from your Cybersource Merchant Admin account. ⚠ Be careful with letter case
Send to production	✓	false - SANDBOX true - PRODUCTION
Abstract path to directory with keys	✓	Directory where p12 file will be placed on the server. ⚠ Do not rename this file as filename is expected to match Organization ID ⚠ Encryption is 2048-bit so ensure that JCE policies are installed in java SDK ⚠ It is recommended to set read only permissions of p12 files
Enable log		Optional parameter to trace SOAP communication for payment. ⚠ Must be disabled for production
Absolute path to log directory		Path to log directory, when "Enable log" is set
Cybersource API version	✓	Verified "1.28"
Use apache HHTP client for communication	✓	true by default

PayPal

To create test account go to <https://developer.paypal.com/> and register. You will need to create a business account and several test customer accounts. Note if you would like to use PayPal Pro you need to upgrade account to pro (Sandbox > Accounts > Select Account > Profile > Account Type > Click Upgrade to Pro)

Enabling PayPal (Button, Express Checkout & Pro) (11:46)

[Play](#)

Part 5 of Enabling Payment Methods

After creating merchant (business) accounts you will be able to access the following from profile menu:

1. **Email ID** used by PayPal Button API
2. **Username**
3. **Password**
4. **Signature**

Key points about your test account:

- The platform uses signature signing. If you want to create certificate signing this is not supported.
- For PayPal Button you need to use **Email ID** as the login, for all others (Pro and Express) use **Username**
- To use Pro features you need to upgrade merchant account (Profile > Account Type > Click Upgrade to Pro)
- Pro test payments must be done using test Credit card provided in merchant's account

(Profile > Funding Tab > Credit Card, use CCV 123 if it is not shown)

PayPal Button

Simple payment API that allows to capture payments from customer with PayPal accounts.

Key configurations for **PayPal Button**

Configuration	Mandatory	Notes
Api user name	✓	Email ID from your PayPal Merchant (Profile > Profile Tab > Email ID)
Api user password	✓	Password from your PayPal Merchant (Profile > API Credentials Tab > Password)
Signature	✓	Signature from your PayPal Merchant (Profile > API Credentials Tab > Signature)
Paypal URL	✓	live - https://www.paypal.com/cgi-bin/webscr sandbox - https://www.sandbox.paypal.com/cgi-bin/webscr
Return URL	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=ok The URL to which PayPal redirects buyers after confirmed payment ⚠ Must be HTTPS
Cancel URL	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel The URL to which PayPal redirects buyers after cancellation of payment ⚠ Must be HTTPS
Api callback URL	✓	http://www.yourdomain.com/yes-shop/paymentpaypalbutton The URL to which PayPal posts information about the payment, in the form of Instant Payment Notification messages ⚠ Must be HTTPS ⚠ Must set UTF-8 encoding in IPN preferences https://www.paypal.com/cgi-bin/customerprofileweb?cmd=_profile-language-encoding (see this stack overflow thread)
IPN encoding	⚠	IPN preferences used mostly for testing in PROD it should be utf-8
PayPal submit button		Can be used to specify PayPal branded buttons
Environment mode	✓	Environment mode used by callback verification. Values can be: sandbox or live

PayPal Express

Advanced payment API that allows to capture payments from customer with PayPal accounts and do refunds.

Key configurations for **PayPal Express**

Configuration	Mandatory	Notes
Api user name	✓	Username from your PayPal Merchant (Profile > API Credentials Tab > Username)
Api user password	✓	Password from your PayPal Merchant (Profile > API Credentials Tab > Password)
Signature	✓	Signature from your PayPal Merchant (Profile > API Credentials Tab > Signature)
Paypal URL	✓	Paypal url to redirect to after success SetExpressCheckout operation live - https://www.paypal.com/cgi-bin/webscr sandbox - https://www.sandbox.paypal.com/cgi-bin/webscr
Api call url	✓	Api call url live - https://api-3t.paypal.com/nvp sandbox - https://api-3t.sandbox.paypal.com/nvp

Return URL	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=ok The URL to which PayPal redirects buyers after confirmed payment ⚠ Must be HTTPS
Cancel URL	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel The URL to which PayPal redirects buyers after cancellation of payment ⚠ Must be HTTPS
Api callback URL	✓	http://www.yourdomain.com/yes-shop/paymentpaypalexpress The URL to which PayPal posts information about the payment, in the form of Instant Payment Notification messages, which will trigger DoExpressCheckoutPayment ⚠ Must be HTTPS ⚠ Must set UTF-8 encoding in IPN preferences https://www.paypal.com/cgi-bin/customerprofileweb?cmd=_profile-language-encoding (see this stack overflow thread)
PayPal submit button		Can be used to specify PayPal branded buttons

PayPal Pro

Fully featured card payments API.

To use Pro features you need to upgrade merchant account (Profile > Account Type > Click Upgrade to Pro)

Pro test payments must be done using test Credit card provided in sandbox's account (Profile > Funding Tab > Credit Card), CCV can be left blank. Ensure that you use the same email address as your sandbox test account and correct billing/shipping addresses.

Sometimes transactions could be rejected with a 10626 error if the order amount is too high

Key configurations for **PayPal Pro**

Configuration	Mandatory	Notes
Api user name	✓	Username from your PayPal Merchant (Profile > API Credentials Tab > Username)
Api user password	✓	Password from your PayPal Merchant (Profile > API Credentials Tab > Password)
Signature	✓	Signature from your PayPal Merchant (Profile > API Credentials Tab > Signature)
HTML Form with card input fields	✓	
Environment mode	✓	Environment mode used by callback verification. Values can be: sandbox or live
Api call url	✓	Api call url live - https://api-3t.paypal.com/nvp sandbox - https://api-3t.sandbox.paypal.com/nvp

IPN

All PayPal payment methods use IPN (callback) to notify of the transaction outcome. As part of the callback the platform performs validation of the request. This is done using PayPal SDK library which essentially performs an http request to PayPal server to verify that this IPN originated from their servers. If this verification request fails the validation mechanism fails and thus the order is not updated.

To fix this issue you need to re-send the IPN message from the PayPal account (see [this stack overflow thread](#)), which can be performed from the IPN history section: https://www.paypal.com/?cmd=_display-ipns-history. Simply find the failed IPN and re-send it. This will re-validate the callback and update the order.

On new interface you can find the IPN history section in More > Sitemap > IPN History. Note that you need to login with your test merchant's account to sandbox to access this menu.

LiqPay







To create LiqPay account go to <https://www.liqpay.ua/en> and register. Note that LiqPay uses SMS messages in order to send pin codes for logging in to your account so you will need a valid mobile phone number.

After registration is completed you will be provided with:

1. **public_key**
2. **private_key**

LiqPay (no refund)

Key configurations for basic **LiqPay (no refund)** with manual refunds


Configuration	Mandatory	Notes
Merchant ID		<ul style="list-style-type: none">• public_key* from your LiqPay account
Merchant signature		<ul style="list-style-type: none">• private_key* from your LiqPay account
Form post URL		https://www.liqpay.com/api/
Payment method		Credit card payments - card LiqPay account - liqpay
Page URL to show payment result		Page where customer is taken after clicking "Return" button on payment page. Recommended is http://www.yourdomain.com/yes-shop/paymentresult configured in wicket.xml
Callback URL with payment result		This is server-to-server transaction callback that LiqPay server performs right before the customer sees result on the external form page. Callback filter is mapped to http://www.yourdomain.com/yes-shop/liqpaycallback

LiqPay

This payment gateway allows to automatically refund money when products are returned.

The configuration is the same as for "LiqPay (no refund)" however you must contact LiqPay and activate refund callbacks for your server IP address.

Key configurations for **LiqPay** with refunds (all configuration as same as "LiqPay no refund" but with the following differences):

Callback URL with payment result		This is server-to-server transaction callback that LiqPay server performs right before the customer sees result on the external form page. Callback filter is mapped to http://www.yourdomain.com/yes-shop/liqpayncallback (note nr in URL)
----------------------------------	---	--

PostFinance

To create a test account you need to contact <https://www.postfinance.ch>. They will ask to fill out a form with your details.

After registration is completed you will be provided with (via email):

- **PSPID** which is affiliation name in PostFinance, also use as login
- Temporary password to login to your account

To access your account go to <https://e-payment.postfinance.ch/>. For test environment click "Access



Enabling PostFinance (6:39)
[Play](#)
Part 6 of Enabling Payment Methods

to test" link before filling out login form.

PostFinance e-payment

Key configurations for basic **PostFinance**




Configuration	Mandatory	Notes
Your affiliation name	✓	PSPID (login) from your PostFinance account
Form action	✓	test - https://e-payment.postfinance.ch/ncol/test/orderstandard_utf8.asp production - https://e-payment.postfinance.ch/ncol/prod/orderstandard_utf8.asp ⚠ Note the URL is UTF-8 specific "orderstandard_utf8.asp", standard "orderstandard.asp" URL does not support UTF-8 and therefore fail the SHA signature check on PostFinance side
SHA-IN signature	✓	Signature for validation post to external form. SHA-IN must be configured in "Configuration > Technical Information > Global Security Parameters" with the following: <ul style="list-style-type: none">• Each parameter followed by the passphrase• SHA-1• UTF-8 and "Configuration > Technical Information > Data and Origin Verification" with the following: <ul style="list-style-type: none">• http://www.yourdomain.com/• SHA-IN signature (e.g. MySecretPass123!#)
⚠ Currencies	✓	Enable currencies that you support at "Configuration > Account > Currency"
⚠ Operation		For this configuration is it preset to SAL, which is AUTH_CAPTURE operation (i.e. funds are captured straight away)
(Absolute) URL of your home page.		http://www.yourdomain.com/yes-shop/
(Absolute) URL of your catalogue.		http://www.yourdomain.com/yes-shop/
URL for accepted payment		http://www.yourdomain.com/yes-shop/paymentresult?hint=ok Can also be configured in "Configuration > Technical Information > Transaction Feedback"
URL for declined payment		http://www.yourdomain.com/yes-shop/paymentresult?hint=declined Can also be configured in "Configuration > Technical Information > Transaction Feedback"
URL for error during payment		http://www.yourdomain.com/yes-shop/paymentresult?hint=exception Can also be configured in "Configuration > Technical Information > Transaction Feedback"
URL for cancelled payment		http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel Can also be configured in "Configuration > Technical Information > Transaction Feedback"
⚠ Callback URL	✓	Server to server callback configured in PostFinance "Configuration > Technical Information > Transaction Feedback" with the following: <ul style="list-style-type: none">• Always online• http://www.yourdomain.com/yes-shop/postfinance (for all callbacks)
SHA-OUT signature	✓	Signature for validation of server-to-server transaction callback. Configured at "Configuration > Technical Information > Transaction Feedback" (e.g. MySecretPass123!#)
Enable itemised data	✓	pre 3.7.0 ⚠ Must be false . There is a rounding issue in PostFinance API that they need to fix before this can be enabled (items summary is added to description) 3.7.0+ ⚠ Set this parameter to true if you would like (or need e.g. Twint enabled) itemised data.
Enable invoice and delivery data		3.7.0+ If set to true uses "ECOM_*" parameters for billing and shipping addresses, if set to false (default behaviour) uses "OWNER*" parameters to set single address. Use true if you require invoice billing integration (e.g. Twint enabled)

Enable invoice and delivery data (line 2 is number)		3.7.0+ If set to true will use addressline2 property of the address as street number . Default is false  Correct street number is required for "invoice" type payment options e.g. Twint
Enable invoice and delivery data (line 1 regex)		If line 2 is number is set to false this regex will be used to attempt to extract street number from the addressline1 . Default regex (if not specified) is: <code>(\s\d+([a-zA-Z])*)(\d+([a-zA-Z])*\s)</code> , which accommodates the following formats: "Street 12, 12b Street or Street 12ab"  Correct street number is required for "invoice" type payment options e.g. Twint

PostFinance e-payment (manual capture)

Manual capture allows merchant to control when the capturing of funds happen.





Key configurations for basic **PostFinance (Manual Capture)** (all configuration as same as regular PostFinance but with the following differences)

Configuration	Mandatory	Notes
 Operation		For this configuration it is preset to RES, which is AUTH operation (i.e. funds are not captured straight away) How funds are captured exactly is configured in "Configuration > Technical Information > Global Transaction Parameters". Note: that this means that at shipping phase there will be a manual override to capture funds as there is no callback from PostFinance
 Callback URL		Server to server callback configured in PostFinance "Configuration > Technical Information > Transaction Feedback" with the following: <ul style="list-style-type: none"> Always online http://www.yourdomain.com/yes-shop/postfinancem (for all callbacks) (note m on the end)

PostFinance Checkout (V2) SaaS 4.0.0

Post Finance V2 is API based integration see <https://checkout.postfinance.ch/en-us/doc> which is using <https://checkout.postfinance.ch/en-us/doc/payment/payment-page> integration style and fully supports void and refund operations

Key configurations for basic **PostFinance Checkout**

Configuration	Mandatory	Notes
API Endpoint		https://checkout.postfinance.ch (all environments)
Integration Type		Only payment_page is currently supported
Request type		Test - for test mode Live - for production live mode
Merchant Space ID		Post Finance space ID. It is recommended to have separate space configured for each sales channel (Space section of PostFinance admin panel).

Merchant App User ID	✓	<p>Application user ID. It is recommended to create a separate application user (Account > Users > Application User section of PostFinance admin panel). When creating a user ensure that a custom role is granted under "Space roles" (leave Account roles and Subaccount roles unassigned). Custom role can be created under Account > Roles and should include permissions to the following APIs:</p> <ul style="list-style-type: none"> • Refund <ul style="list-style-type: none"> • Read • Confirm offline refund • Create • Transaction <ul style="list-style-type: none"> • Read • Create • Update line items • Send email transactions • Void <ul style="list-style-type: none"> • Create • Read
Merchant App User Auth Key	✓	This key is provided when creating Application User in Account > Users > Application User section of PostFinance admin panel
URL for accepted payment	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=ok
URL for error during payment	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=exception
URL for cancelled payment	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel_no_clean

SwissBilling SaaS 3.7.0

Please contact SwissBilling team to activate your account <https://www.swissbilling.ch/>

SWB uses several API in order to facilitate its online functions. Primarily placing an order is accomplished by V2 APIs, whereas V3 APIs are used for cancellation and other admin activities.

⚠️ SwissBilling requires phone number and date of birth data for authorising the payment therefore it is advisable to configure profile attributes (CUSTOMER_PHONE, CUSTOMER_DOB_YEAR, CUSTOMER_DOB_MONTH, CUSTOMER_DOB_DAY) and address attributes (phone1, phone2, mobile1, mobile2) to include this data.

SwissBilling

Key configurations for basic **SwissBilling**

Configuration	Mandatory	Notes
Merchand ID	✓	ID for SwissBilling API
Merchand Password	✓	Password for SwissBilling API
API Endpoint V2	✓	Core APIs required for placing order test - https://sr-pp.swissbilling.ch:443/EShopRequestServerV2StdSec.php production - https://secure.safebill.ch:443/EShopRequestServerV2StdSec.php
Request type	✓	⚠️ Always set as Real , Test although stated in documentation is not supported
Request SOAP Action	✓	Set to urn:EShopRequest by default

Confirm SOAP Action	✓	Set to urn:EShopRequest by default
Enable itemised data	✓	Enable itemised data in request for authorisation to SwissBilling (true by default)
API Endpoint V3	✓	Admin APIs required for fulfilment (e.g. cancellation) test - https://sr-pp.swissbilling.ch:443/EShopRequestServerV2StdSec.php production - https://secure.safebill.ch:443/EShopRequestServerV2StdSec.php
Cancel SOAP Action	✓	Set to swb.ws.risk.web.ws/!EshopRequestV3/EshopTransactionCancel by default
Swissbilling callback page	✓	Swissbilling callback page e.g. https://yourdomain.com/paymentswissbilling ⚠ This is internal page that allows to pre-authorise the request to SwissBilling and also post processes return URL from SwissBilling
Success URL	✓	Success URL e.g. https://yourdomain.com/paymentresult?hint=ok This is the redirect that will happen after callback has been processed if the order has been confirmed by SwissBilling
Cancel URL	✓	Success URL e.g. Cancel URL e.g. https://yourdomain.com/paymentresult?hint=cancel_no_clean ⚠ if you set hint to cancel this will clean up the cart when customer is redirected back to the website (this callback usually happens when order is cancelled by user at SwissBilling website)
Error URL	✓	Error URL e.g. https://yourdomain.com/paymentresult?hint=exception This is the redirect that will happen after callback has been processed if the order has been declined by SwissBilling either as a result of validation failure or rejection

PaySera SaaS 3.7.0

Please follow PaySera instructions to activate your account <https://bank.paysera.com/en/registration>

⚠ Current implementation supports only basic features for online payment of the order in full, no other features are supported until further notice

PaySera Checkout

Key configurations for basic **PaySera** integration (see <https://developers.paysera.com/en/checkout/integrations/integration-specification>)

Configuration	Mandatory	Notes
Unique Project Number	✓	Unique project number. Only activated projects can accept payments
Project Password	✓	Password for Project
API Endpoint	✓	Core APIs required for placing order test - https://www.paysera.com/pay/ production - https://www.paysera.com/pay/
Request type	✓	⚠ Always set as Real , Test although stated in documentation is not supported
API version	✓	Implementation is done for 1.6 , please see PaySera documentation for compatibility
PaySera callback page	✓	Swissbilling callback page e.g. https://yourdomain.com/paymentswissbilling ⚠ This is internal page that allows to pre-authorise the request to SwissBilling and also post processes return URL from SwissBilling

Success URL	✓	Success URL e.g. https://yourdomain.com/paymentresult?hint=ok This is the redirect that will happen after callback has been processed if the order has been confirmed by SwissBilling
Cancel URL	✓	Success URL e.g. Cancel URL e.g. https://yourdomain.com/paymentresult?hint=cancel_no_clean ⚠ if you set hint to cancel this will clean up the cart when customer is redirected back to the website (this callback usually happens when order is cancelled by user at SwissBilling website)
Error URL	✓	Error URL e.g. https://yourdomain.com/paymentresult?hint=exception This is the redirect that will happen after callback has been processed if the order has been declined by SwissBilling either as a result of validation failure or rejection

Test cards and addresses

Usually test cards resources are specified in documentation of specific payment providers.

For PayPal Pro use test cards provided on the "Funds" tab of your test accounts' profile

Here are some examples of working test cards:

Card	Number	CVV	Expiry
VISA	4111 1111 1111 1111	123	12/2020

Some payment providers have AVS (Address Verification System, so the address has to be correct as well)

Street 1	City	State	Country	Post code	Phone	Notes
1295 Charleston Road	Mountain View	CA	US	94043	650-965-6000	⚠ As this is a US address order currency must be USD

- Overview
- Payment Gateway Management
 - Payment gateway eligibility
 - Payment gateway sorting
 - Workshops
 - Authorize.net
 - AIM
 - SIM
 - CyberSource
 - PayPal
 - PayPal Button
 - PayPal Express
 - PayPal Pro
 - IPN
 - LiqPay
 - LiqPay (no refund)
 - LiqPay
 - PostFinance
 - PostFinance e-payment
 - PostFinance e-payment (manual capture)
 - PostFinance Checkout (V2) SaaS 4.0.0
 - SwissBilling SaaS 3.7.0
 - SwissBilling
 - PaySera SaaS 3.7.0
 - PaySera Checkout

- Test cards and addresses
- Overview
- Payment Gateway Management
 - Payment gateway eligibility
 - Payment gateway sorting
 - Workshops
 - Authorize.net
 - AIM
 - SIM
 - CyberSource
 - PayPal
 - PayPal Button
 - PayPal Express
 - PayPal Pro
 - IPN
 - LiqPay
 - LiqPay (no refund)
 - LiqPay
 - PostFinance
 - PostFinance e-payment
 - PostFinance e-payment (manual capture)
 - PostFinance Checkout (V2) SaaS 4.0.0
 - SwissBilling SaaS 3.7.0
 - SwissBilling
 - PaySera 4.1.0
 - PaySera Checkout
- Test cards and addresses

Overview

Payment methods refer to choice of payment options available to customer when placing an order. In most cases payment method relies on a third party system (payment gateway) to complete the financial transaction. Although financial operations are fairly standardised and well understood each payment gateway offers unique API to accomplish this. Therefore payment API is composed of two parts:

- **core payment API**, which is fully integrated with [order life cycle](#), dealing with standard approaches to transactions
- **payment modules**, which are payment gateway specific implementations that drive the standard operations.



The platform has a number of payment modules implementation provided out of the box with flexibility to add new implementations with ease.

Once the platform up and running all **payment modules are automatically detected and contribute to the payment gateways configuration**. Through system payment gateway panel business user can enable and disable payment gateways at the platform level. Shop managers when configuring [shop](#) can choose from the list of platform enabled payment gateways and configure them with shop specific parameters.

Payment gateway parameters are predefined by the payment module, so all that is necessary to enable payment methods for shop is to setup these values, which usually involves entering merchant keys or IDs.

The following payment gateway modules are supported out of the box (with few featured highlighted):

Module	Payment method	version	EoL	Online	External	Callback	Per Shipment	AUTH	CAPTURE	AUTH_C
Core										
	Payment to courier	1.0.0+		offline			✓	✓	✓	
	Pre paid (external payments)	1.0.0+		offline						
	In store	1.0.0+		offline			✓	✓	✓	
	Test (card)	1.0.0+		online mock			✓	✓	✓	
	TestExt (card)	1.0.0+		online mock	✓	base callback filter	✓	✓	✓	
	Invoice	3.5.0+		offline						
	Invoice with Authorisation	3.5.0+		offline			✓	✓	✓	
	No payment required	3.7.0+		online						

Authorize.net										
	AIM	2.0.0+		online			✓	✓	✓	✓
	SIM	2.0.0+		online	✓	AuthorizeNetSimPaymentOkPage				✓
CyberSource										
	CyberSource	2.0.0+		online			✓	✓	✓	✓
PayPal										
	PayPal Button	3.1.0+		online	✓	base callback filter				✓
	PayFlow 	2.0.0+	3.0.0	online			✓	✓	✓	✓
	PayPal Express	2.0.0+		online	✓	PP express callback filter (set mode, tx confirmation)				✓
	PayPal NPV 	2.0.0+	3.0.0	online			✓	✓	✓	✓
	PayPal Pro	3.1.0+		online			✓	✓	✓	✓
LiqPay										
	LiqPay (full)	2.0.0+		online	✓	base callback filter				✓
	LiqPay (no refund)	2.0.0+		online	✓	base callback filter				✓
PostFinance										
	PostFinance e-payment	3.1.0+		online	✓	base callback filter				✓
	PostFinance checkout	4.0.0+ Sa aS		online	✓	PF callback filter (session setup, tx verification)				✓
Swissbilling										
	Swissbilling	3.7.0+ Sa aS		online	✓	SB callback filter (pre-screening, tx confirmation)				✓
PaySera										
	PaySera	4.1.0+		online	✓	PaySera callback filter (enhanced version of base filter that also send "OK" in response)				✓

Payment Gateway Management

Payment gateways list loaded into this panel is composed of automatically resolved payment gateways modules installed on current instance. The installation process is very simple and involves specifying few maven profile parameters during [build](#).

When this list is loaded system administrator can select which modules are enabled or disabled by clicking "on" and "off" buttons. If a payment gateway does not appear in this list then it is highly possible that this module was not added to the build, otherwise all payment gateways will appear in this list loaded from bundled modules.

Each payment gateway has predefined attributes, which can be viewed by clicking "Attributes" button when payment gateway is selected. Note that **these parameters are templates** - the **actual values are set when configuring payment gateway for shop**.

There is however an option to add additional attributes to template by clicking "New attribute" button. This allows to add new attributes such as new localisation for HTML form when new language is added.

After new attribute is added to the system payment templates the **shop payment gateway has to be turned "off" and then back "on" for new attribute to appear in shop** specific settings.

As of 3.4.0+ this view uses "Secure" feature, which means that by default only non-secure parameters are loaded. If you want to see all parameters you have to click the "lock" button. Then you will see all attributes available for payment gateway in the attribute view.

DEMO ENVIRONMENT

Powered by
YC - pure Commerce
Logged in as Yes

Customer service

PIM & Master Catalog

CMS

Marketing & Pricing

Inventory

Reports

Shipping

Shops

Organisation

System

Attributes

Import

Export

Search Indexes

Payment Gateways

Unprocessed Callbacks

Clone Shop

Preferences

Query API

Cache Monitoring

Performance Sensors

Performance Samples

Tasks

Cluster

Help

Payment Gateways / Authorize.net SIM

Secure mode (on/off)

Attributes

Enable & Disable Buttons

Features Information

Rank	Code	Name	Enabled
100	authInvoicePaymentGatewayLabel	Authorize invoice payment	<input type="checkbox"/>
100	authorizeNetAimPaymentGatewayLabel	Authorize.net AIM	<input type="checkbox"/>
100	authorizeNetSimPaymentGatewayLabel	Authorize.net SIM	<input checked="" type="checkbox"/>
100	cyberSourcePaymentGatewayLabel	CyberSource	<input type="checkbox"/>
100	invoicePaymentGatewayLabel	Invoice payment	<input type="checkbox"/>
100	liqPayPaymentGatewayLabel	LiqPay	<input type="checkbox"/>
100	liqPayNoRefundPaymentGatewayLabel	LiqPay (No Refund)	<input type="checkbox"/>
100	prePaymentGatewayLabel	Offline Pre-Payment	<input checked="" type="checkbox"/>
100	paypalButtonPaymentGatewayLabel	PayPal B	<input type="checkbox"/>
100	paypalExpressPaymentGatewayLabel	PayPal E	<input type="checkbox"/>

Attributes view

Authorize.net SIM

New Attribute

Edit Attribute

Remove Attribute Value

Attribute	Value
Gateway name (default)	Authorize.net SIM
Gateway name (EN)	Authorize.net SIM
Название платежного шлюза (RU)	Authorize.net SIM
Назва платіжного шлюзу (UK)	Authorize.net SIM
Gateway-Namen (DE)	Authorize.net SIM
Uri to post form.	https://test.authorize.net/gateway/transaction.dll
Relay response uri	http://@domain@/anetsimresult
SIM/DPM order receipt uri	http://@domain@/paymentresult?hnt=ok
SIM test request flag	FALSE
Gateway priority	100

These are DEFAULTS, they can be overwritten by shop specific values

Attribute

Attribute

Payment gateway eligibility

Payment gateways are directly influenced by shipping method configuration. The anticipation is that the kind of shipping method (i.e. carrier

SLA) is the main driver for the payment eligibility.

For example if the shipping method is "collect from shop" it would make sense to display "Payment in shop" payment method, but not for say "Home delivery".

Carrier SLA configurations contain selection for payment gateways available to those SLA.

As of 3.6.0 and additional customer level restriction is available for payment gateways via "tagging" mechanism. Payment gateways have "**restrictToCustomerTags**" parameter which is a CSV of eligible customer tags. When filled in this parameter which contain use of given payment method only to customer that have at least one matching tag.

Note that restrictToCustomerTags is CSV and expects values separated by comma (e.g. "tag1,tag2"), whereas customer tags property expects tags separated by space (e.g. "tag2 othertag tag4")

Payment gateway sorting

By default payment methods are *sorted alphabetically by their language specific name*. However if manual sorting is required then "**priority**" parameter can be used to sort them.

Note that if priorities of payment gateways are the same then sorting is done alphabetically. This is very useful when a particular payment method can be set higher priority to be at the top and then rest of the methods do not need to be adjusted and will be sorted in natural order.

Workshops

Activating payment modules (5:45)
[Play](#)
Part 1 of Enabling Payment Methods

Enabling payment methods in your shop (basics) (4:57)
[Play](#)
Part 2 of Enabling Payment Methods

Payment Gateway Specific Configurations

Authorize.net

To create test account go to http://developer.authorize.net/hello_world/sandbox/ and fill in the form.

After registration is completed you will be provided with:

1. **API Login ID**
2. **Transaction Key**
3. **MD5 Hash Key**

Enabling Authorize.NET (AIM & SIM)
(7:09)
[Play](#)
Part 3 of Enabling Payment Methods

You can login to your test account from here <https://sandbox.authorize.net>

Key points about your test account:

- **API Login ID** and **Transaction Key** can be viewed in "Settings > Security Settings > API Login ID and Transaction Key"
- **MD5 Hash Key** can be reset in "Settings > Security Settings > MD5 Hash"
- For SIM Receipt Page and Relay Response URL must be configured in "Settings > Security Settings > Receipt Page" and "Settings > Security Settings > Relay Response" respectively.
- Relay Response is what is used as transaction callback and its response is rendered as confirmation page back to the customer after clicking "Pay" button.

AIM

Key configurations for **AIM**

Configuration	Mandatory	Notes
Merchant login	✓	API Login ID from your Authorize.NET account
Transaction key	✓	Transaction Key from your Authorize.NET account
Environment name	✓	test - "SANDBOX" production - "PRODUCTION"

SIM

SIM account is currency specific, so you will not be able to provide different currency when paying with this method. You can configure the currency in the processor settings section

Key points for **SIM**

Configuration	Mandatory	Notes
Merchant login	✓	API Login ID from your Authorize.NET account
Transaction key	✓	Transaction Key from your Authorize.NET account
MD5 Hash Key	✓	MD5 Hash Key from your Authorize.NET account used to verify transaction callback
URL to post form	✓	test - " https://test.authorize.net/gateway/transact.dll " production - " https://secure.authorize.net/gateway/transact.dll "
Relay Response URL	✓	must be set to the " https://www.yourdomain.com/yes-shop/anetsimresult ". Note: that this page must be served via HTTPS and it processes the transaction callback . Note: "/anetsimresult" is mounted to AuthorizeNetSimPaymentOkPage in "wicket.xml"
SIM test request flag	✓	test transaction - TRUE actual transaction - FALSE
Payment form		There is a number of parameters that SIM supports to modify look and feel of the external payment form. Recommended "Order cancel URL" is " http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel ".

CyberSource

To create test account go to <http://www.cybersource.com/register/> and fill in the form.

After registration is completed you will be provided with:

1. **Organization ID**
2. Link to activate **Merchant Admin account**

Enabling CyberSource (7:23)

[Play](#)

Part 4 of Enabling Payment Methods

You need this account to generate the p12 key

3. Link to activate **Account Admin account**

You can login to your test account from here <https://ebctest.cybersource.com>

Key points about your test account:

- If you have capital letters in your Organization ID they will be converted to lower case (be aware of this)

- Generating p12 certificate is done from Login to **Merchant Admin** > Account Management > Transaction Security Keys > Security Keys for the Simple Order API
- You need to allow applet in order to save the certificate. Certificate name will be Organization ID with p12 extension.
- **p12 certificate is 2048-bit, so java SDK security must have "Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files" installed.**
- **on some JDK8 due to classloader bug you may need to copy bcprov-ext-jdk15on.jar into JAVA_HOME/jre/lib/ext** if you get "error constructing MAC:
java.security.InvalidAlgorithmParameterException: inappropriate parameter type:
javax.crypto.spec.PBEParameterSpec"
- Cybersource uses AVS, so address must be correct for payments to go through.

Key configurations for AIM

Configuration	Mandatory	Notes
Merchant id	✓	Organization ID from your Cybersource Merchant Admin account. ⚠ Be careful with letter case
Send to production	✓	false - SANDBOX true - PRODUCTION
Abstract path to directory with keys	✓	Directory where p12 file will be placed on the server. ⚠ Do not rename this file as filename is expected to match Organization ID ⚠ Encryption is 2048-bit so ensure that JCE policies are installed in java SDK ⚠ It is recommended to set read only permissions of p12 files
Enable log		Optional parameter to trace SOAP communication for payment. ⚠ Must be disabled for production
Absolute path to log directory		Path to log directory, when "Enable log" is set
Cybersource API version	✓	Verified "1.28"
Use apache HHTP client for communication	✓	true by default

PayPal

To create test account go to <https://developer.paypal.com/> and register. You will need to create a business account and several test customer accounts. Note if you would like to use PayPal Pro you need to upgrade account to pro (Sandbox > Accounts > Select Account > Profile > Account Type > Click Upgrade to Pro)

Enabling PayPal (Button, Express Checkout & Pro) (11:46)

[Play](#)

Part 5 of Enabling Payment Methods

After creating merchant (business) accounts you will be able to access the following from profile menu:

1. **Email ID** used by PayPal Button API
2. **Username**
3. **Password**
4. **Signature**

Key points about your test account:

- The platform uses signature signing. If you want to create certificate signing this is not supported.
- For PayPal Button you need to use **Email ID** as the login, for all others (Pro and Express) use **Username**
- To use Pro features you need to upgrade merchant account (Profile > Account Type > Click Upgrade to Pro)
- Pro test payments must be done using test Credit card provided in merchant's account

(Profile > Funding Tab > Credit Card, use CCV 123 if it is not shown)

PayPal Button

Simple payment API that allows to capture payments from customer with PayPal accounts.

Key configurations for **PayPal Button**

Configuration	Mandatory	Notes
Api user name	✓	Email ID from your PayPal Merchant (Profile > Profile Tab > Email ID)
Api user password	✓	Password from your PayPal Merchant (Profile > API Credentials Tab > Password)
Signature	✓	Signature from your PayPal Merchant (Profile > API Credentials Tab > Signature)
Paypal URL	✓	live - https://www.paypal.com/cgi-bin/webscr sandbox - https://www.sandbox.paypal.com/cgi-bin/webscr
Return URL	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=ok The URL to which PayPal redirects buyers after confirmed payment ⚠ Must be HTTPS
Cancel URL	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel The URL to which PayPal redirects buyers after cancellation of payment ⚠ Must be HTTPS
Api callback URL	✓	http://www.yourdomain.com/yes-shop/paymentpaypalbutton The URL to which PayPal posts information about the payment, in the form of Instant Payment Notification messages ⚠ Must be HTTPS ⚠ Must set UTF-8 encoding in IPN preferences https://www.paypal.com/cgi-bin/customerprofileweb?cmd=_profile-language-encoding (see this stack overflow thread)
IPN encoding	⚠	IPN preferences used mostly for testing in PROD it should be utf-8
PayPal submit button		Can be used to specify PayPal branded buttons
Environment mode	✓	Environment mode used by callback verification. Values can be: sandbox or live

PayPal Express

Advanced payment API that allows to capture payments from customer with PayPal accounts and do refunds.

Key configurations for **PayPal Express**

Configuration	Mandatory	Notes
Api user name	✓	Username from your PayPal Merchant (Profile > API Credentials Tab > Username)
Api user password	✓	Password from your PayPal Merchant (Profile > API Credentials Tab > Password)
Signature	✓	Signature from your PayPal Merchant (Profile > API Credentials Tab > Signature)
Paypal URL	✓	Paypal url to redirect to after success SetExpressCheckout operation live - https://www.paypal.com/cgi-bin/webscr sandbox - https://www.sandbox.paypal.com/cgi-bin/webscr
Api call url	✓	Api call url live - https://api-3t.paypal.com/nvp sandbox - https://api-3t.sandbox.paypal.com/nvp

Return URL	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=ok The URL to which PayPal redirects buyers after confirmed payment ⚠ Must be HTTPS
Cancel URL	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel The URL to which PayPal redirects buyers after cancellation of payment ⚠ Must be HTTPS
Api callback URL	✓	http://www.yourdomain.com/yes-shop/paymentpaypalexpress The URL to which PayPal posts information about the payment, in the form of Instant Payment Notification messages, which will trigger DoExpressCheckoutPayment ⚠ Must be HTTPS ⚠ Must set UTF-8 encoding in IPN preferences https://www.paypal.com/cgi-bin/customerprofileweb?cmd=_profile-language-encoding (see this stack overflow thread)
PayPal submit button		Can be used to specify PayPal branded buttons

PayPal Pro

Fully featured card payments API.

To use Pro features you need to upgrade merchant account (Profile > Account Type > Click Upgrade to Pro)

Pro test payments must be done using test Credit card provided in sandbox's account (Profile > Funding Tab > Credit Card), CCV can be left blank. Ensure that you use the same email address as your sandbox test account and correct billing/shipping addresses.

Sometimes transactions could be rejected with a 10626 error if the order amount is too high

Key configurations for **PayPal Pro**

Configuration	Mandatory	Notes
Api user name	✓	Username from your PayPal Merchant (Profile > API Credentials Tab > Username)
Api user password	✓	Password from your PayPal Merchant (Profile > API Credentials Tab > Password)
Signature	✓	Signature from your PayPal Merchant (Profile > API Credentials Tab > Signature)
HTML Form with card input fields	✓	
Environment mode	✓	Environment mode used by callback verification. Values can be: sandbox or live
Api call url	✓	Api call url live - https://api-3t.paypal.com/nvp sandbox - https://api-3t.sandbox.paypal.com/nvp

IPN

All PayPal payment methods use IPN (callback) to notify of the transaction outcome. As part of the callback the platform performs validation of the request. This is done using PayPal SDK library which essentially performs an http request to PayPal server to verify that this IPN originated from their servers. If this verification request fails the validation mechanism fails and thus the order is not updated.

To fix this issue you need to re-send the IPN message from the PayPal account (see [this stack overflow thread](#)), which can be performed from the IPN history section: https://www.paypal.com/?cmd=_display-ipns-history. Simply find the failed IPN and re-send it. This will re-validate the callback and update the order.

On new interface you can find the IPN history section in More > Sitemap > IPN History. Note that you need to login with your test merchant's account to sandbox to access this menu.

LiqPay







To create LiqPay account go to <https://www.liqpay.ua/en> and register. Note that LiqPay uses SMS messages in order to send pin codes for logging in to your account so you will need a valid mobile phone number.

After registration is completed you will be provided with:

1. **public_key**
2. **private_key**

LiqPay (no refund)

Key configurations for basic **LiqPay (no refund)** with manual refunds


Configuration	Mandatory	Notes
Merchant ID		<ul style="list-style-type: none">• public_key* from your LiqPay account
Merchant signature		<ul style="list-style-type: none">• private_key* from your LiqPay account
Form post URL		https://www.liqpay.com/api/
Payment method		Credit card payments - card LiqPay account - liqpay
Page URL to show payment result		Page where customer is taken after clicking "Return" button on payment page. Recommended is http://www.yourdomain.com/yes-shop/paymentresult configured in wicket.xml
Callback URL with payment result		This is server-to-server transaction callback that LiqPay server performs right before the customer sees result on the external form page. Callback filter is mapped to http://www.yourdomain.com/yes-shop/liqpaycallback

LiqPay

This payment gateway allows to automatically refund money when products are returned.

The configuration is the same as for "LiqPay (no refund)" however you must contact LiqPay and activate refund callbacks for your server IP address.

Key configurations for **LiqPay** with refunds (all configuration as same as "LiqPay no refund" but with the following differences):

Callback URL with payment result		This is server-to-server transaction callback that LiqPay server performs right before the customer sees result on the external form page. Callback filter is mapped to http://www.yourdomain.com/yes-shop/liqpayncallback (note nr in URL)
----------------------------------	---	--

PostFinance

To create a test account you need to contact <https://www.postfinance.ch>. They will ask to fill out a form with your details.

After registration is completed you will be provided with (via email):

- **PSPID** which is affiliation name in PostFinance, also use as login
- Temporary password to login to your account

To access your account go to <https://e-payment.postfinance.ch/>. For test environment click "Access



Enabling PostFinance (6:39)
[Play](#)
Part 6 of Enabling Payment Methods

to test" link before filling out login form.

PostFinance e-payment

Key configurations for basic **PostFinance**




Configuration	Mandatory	Notes
Your affiliation name	✓	PSPID (login) from your PostFinance account
Form action	✓	test - https://e-payment.postfinance.ch/ncol/test/orderstandard_utf8.asp production - https://e-payment.postfinance.ch/ncol/prod/orderstandard_utf8.asp ⚠ Note the URL is UTF-8 specific "orderstandard_utf8.asp", standard "orderstandard.asp" URL does not support UTF-8 and therefore fail the SHA signature check on PostFinance side
SHA-IN signature	✓	Signature for validation post to external form. SHA-IN must be configured in "Configuration > Technical Information > Global Security Parameters" with the following: <ul style="list-style-type: none">• Each parameter followed by the passphrase• SHA-1• UTF-8 and "Configuration > Technical Information > Data and Origin Verification" with the following: <ul style="list-style-type: none">• http://www.yourdomain.com/• SHA-IN signature (e.g. MySecretPass123!#)
⚠ Currencies	✓	Enable currencies that you support at "Configuration > Account > Currency"
⚠ Operation		For this configuration is it preset to SAL, which is AUTH_CAPTURE operation (i.e. funds are captured straight away)
(Absolute) URL of your home page.		http://www.yourdomain.com/yes-shop/
(Absolute) URL of your catalogue.		http://www.yourdomain.com/yes-shop/
URL for accepted payment		http://www.yourdomain.com/yes-shop/paymentresult?hint=ok Can also be configured in "Configuration > Technical Information > Transaction Feedback"
URL for declined payment		http://www.yourdomain.com/yes-shop/paymentresult?hint=declined Can also be configured in "Configuration > Technical Information > Transaction Feedback"
URL for error during payment		http://www.yourdomain.com/yes-shop/paymentresult?hint=exception Can also be configured in "Configuration > Technical Information > Transaction Feedback"
URL for cancelled payment		http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel Can also be configured in "Configuration > Technical Information > Transaction Feedback"
⚠ Callback URL	✓	Server to server callback configured in PostFinance "Configuration > Technical Information > Transaction Feedback" with the following: <ul style="list-style-type: none">• Always online• http://www.yourdomain.com/yes-shop/postfinance (for all callbacks)
SHA-OUT signature	✓	Signature for validation of server-to-server transaction callback. Configured at "Configuration > Technical Information > Transaction Feedback" (e.g. MySecretPass123!#)
Enable itemised data	✓	pre 3.7.0 ⚠ Must be false . There is a rounding issue in PostFinance API that they need to fix before this can be enabled (items summary is added to description) 3.7.0+ ⚠ Set this parameter to true if you would like (or need e.g. Twint enabled) itemised data.
Enable invoice and delivery data		3.7.0+ If set to true uses "ECOM_*" parameters for billing and shipping addresses, if set to false (default behaviour) uses "OWNER*" parameters to set single address. Use true if you require invoice billing integration (e.g. Twint enabled)

Enable invoice and delivery data (line 2 is number)		3.7.0+ If set to true will use addressline2 property of the address as street number . Default is false  Correct street number is required for "invoice" type payment options e.g. Twint
Enable invoice and delivery data (line 1 regex)		If line 2 is number is set to false this regex will be used to attempt to extract street number from the addressline1 . Default regex (if not specified) is: <code>(\s\d+([a-zA-Z])*)(\d+([a-zA-Z])*\s)</code> , which accommodates the following formats: "Street 12, 12b Street or Street 12ab"  Correct street number is required for "invoice" type payment options e.g. Twint

PostFinance e-payment (manual capture)

Manual capture allows merchant to control when the capturing of funds happen.





Key configurations for basic **PostFinance (Manual Capture)** (all configuration as same as regular PostFinance but with the following differences)

Configuration	Mandatory	Notes
 Operation		For this configuration it is preset to RES, which is AUTH operation (i.e. funds are not captured straight away) How funds are captured exactly is configured in "Configuration > Technical Information > Global Transaction Parameters". Note: that this means that at shipping phase there will be a manual override to capture funds as there is no callback from PostFinance
 Callback URL		Server to server callback configured in PostFinance "Configuration > Technical Information > Transaction Feedback" with the following: <ul style="list-style-type: none"> Always online http://www.yourdomain.com/yes-shop/postfinancem (for all callbacks) (note m on the end)

PostFinance Checkout (V2) SaaS 4.0.0

Post Finance V2 is API based integration see <https://checkout.postfinance.ch/en-us/doc> which is using <https://checkout.postfinance.ch/en-us/doc/payment/payment-page> integration style and fully supports void and refund operations

Key configurations for basic **PostFinance Checkout**

Configuration	Mandatory	Notes
API Endpoint		https://checkout.postfinance.ch (all environments)
Integration Type		Only payment_page is currently supported
Request type		Test - for test mode Live - for production live mode
Merchant Space ID		Post Finance space ID. It is recommended to have separate space configured for each sales channel (Space section of PostFinance admin panel).

Merchant App User ID	✓	<p>Application user ID. It is recommended to create a separate application user (Account > Users > Application User section of PostFinance admin panel). When creating a user ensure that a custom role is granted under "Space roles" (leave Account roles and Subaccount roles unassigned). Custom role can be created under Account > Roles and should include permissions to the following APIs:</p> <ul style="list-style-type: none"> • Refund <ul style="list-style-type: none"> • Read • Confirm offline refund • Create • Transaction <ul style="list-style-type: none"> • Read • Create • Update line items • Send email transactions • Void <ul style="list-style-type: none"> • Create • Read
Merchant App User Auth Key	✓	This key is provided when creating Application User in Account > Users > Application User section of PostFinance admin panel
URL for accepted payment	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=ok
URL for error during payment	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=exception
URL for cancelled payment	✓	http://www.yourdomain.com/yes-shop/paymentresult?hint=cancel_no_clean

SwissBilling SaaS 3.7.0

Please contact SwissBilling team to activate your account <https://www.swissbilling.ch/>

SWB uses several API in order to facilitate its online functions. Primarily placing an order is accomplished by V2 APIs, whereas V3 APIs are used for cancellation and other admin activities.

⚠️ SwissBilling requires phone number and date of birth data for authorising the payment therefore it is advisable to configure profile attributes (CUSTOMER_PHONE, CUSTOMER_DOB_YEAR, CUSTOMER_DOB_MONTH, CUSTOMER_DOB_DAY) and address attributes (phone1, phone2, mobile1, mobile2) to include this data.

SwissBilling

Key configurations for basic **SwissBilling**

Configuration	Mandatory	Notes
Merchand ID	✓	ID for SwissBilling API
Merchand Password	✓	Password for SwissBilling API
API Endpoint V2	✓	Core APIs required for placing order test - https://sr-pp.swissbilling.ch:443/EShopRequestServerV2StdSec.php production - https://secure.safebill.ch:443/EShopRequestServerV2StdSec.php
Request type	✓	⚠️ Always set as Real , Test although stated in documentation is not supported
Request SOAP Action	✓	Set to urn:EShopRequest by default

Confirm SOAP Action	✓	Set to urn:EShopRequest by default
Enable itemised data	✓	Enable itemised data in request for authorisation to SwissBilling (true by default)
API Endpoint V3	✓	Admin APIs required for fulfilment (e.g. cancellation) test - https://sr-pp.swissbilling.ch:443/EShopRequestServerV2StdSec.php production - https://secure.safebill.ch:443/EShopRequestServerV2StdSec.php
Cancel SOAP Action	✓	Set to swb.ws.risk.web.ws/EShopRequestV3/EShopTransactionCancel by default
Swissbilling callback page	✓	Swissbilling callback page e.g. https://yourdomain.com/paymentswissbilling ⚠ This is internal page that allows to pre-authorise the request to SwissBilling and also post processes return URL from SwissBilling
Success URL	✓	Success URL e.g. https://yourdomain.com/paymentresult?hint=ok This is the redirect that will happen after callback has been processed if the order has been confirmed by SwissBilling
Cancel URL	✓	Success URL e.g. Cancel URL e.g. https://yourdomain.com/paymentresult?hint=cancel_no_clean ⚠ if you set hint to cancel this will clean up the cart when customer is redirected back to the website (this callback usually happens when order is cancelled by user at SwissBilling website)
Error URL	✓	Error URL e.g. https://yourdomain.com/paymentresult?hint=exception This is the redirect that will happen after callback has been processed if the order has been declined by SwissBilling either as a result of validation failure or rejection

PaySera 4.1.0

Please follow PaySera instructions to activate your account <https://bank.paysera.com/en/registration>

⚠ Current implementation supports only basic features for online payment of the order in full, no other features are supported until further notice

PaySera Checkout

Key configurations for basic **PaySera** integration (see <https://developers.paysera.com/en/checkout/integrations/integration-specification>)

Configuration	Mandatory	Notes
Project ID	✓	Unique ID for project (Only activated projects can accept payments)
Project Password	✓	Password for Project
API version	✓	1.6 is the implementation version. Refer to PaySera documentation for compatibility with other versions
Payment page (Post URL)	✓	Payment page required for placing order test - https://www.paysera.com/pay/ production - https://www.paysera.com/pay/
Environment mode	✓	live or test
Callback URL	✓	PaySera callback URL e.g. https://yourdomain.com/paymentpayseracheckout ⚠ Allows to verify and confirm payment

Success URL	✓	Success URL e.g. https://yourdomain.com/paymentresult?hint=ok This is the redirect that will happen after callback has been processed if the order has been confirmed by PaySera
Cancel URL	✓	Success URL e.g. Cancel URL e.g. https://yourdomain.com/paymentresult?hint=cancel_no_clean ⚠ if you set hint to cancel this will clean up the cart when customer is redirected back to the website (this callback usually happens when order is cancelled by user at PaySera website)
Error URL	✓	Error URL e.g. https://yourdomain.com/paymentresult?hint=exception This is the redirect that will happen after callback has been processed if the order has been declined by PaySera either as a result of validation failure or rejection

Test cards and addresses

Usually test cards resources are specified in documentation of specific payment providers.

For PayPal Pro use test cards provided on the "Funds" tab of your test accounts' profile

Here are some examples of working test cards:

Card	Number	CVV	Expiry
VISA	4111 1111 1111 1111	123	12/2020

Some payment providers have AVS (Address Verification System, so the address has to be correct as well)

Street 1	City	State	Country	Post code	Phone	Notes
1295 Charleston Road	Mountain View	CA	US	94043	650-965-6000	⚠ As this is a US address order currency must be USD