COLLECTING SOLUTION

Adding the payment button PayPal - Immediate payment

Implementation guide

Document version 3.5

Contents

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
3.5	Lyra Collect	01/10/2018	Initial version

2. OBTAINING HELP

Need some help? Please, check the FAQ on our website

https://lyra.com/doc/en/collect/faq/sitemap.html

For technical inquiries or support, you can reach usfrom Monday to Friday, between 9 a.m. and 6 p.m.

by phone at: **0811900475** Service fee 0.06 € / min + call charge

by e-mail: *support-ecommerce@lyra-collect.com*

and via your Expert Back Office, menu **Help** > **Contact support**

3. PAYPAL : PAYING ONLINE WITHOUT ENTERING YOUR BANK ACCOUNT DETAILS

PayPal is a payment method allowing merchants to receive "single" and recurring payments made with credit cards and via PayPal on their websites.

Paying online with PayPal means paying without sharing financial information. The buyer pays after signing in with an e-mail address and a password.

4. USING PAYPAL

This chapter describes the PayPal online payment process.

It will provide you with step-by-step instructions for integrating this payment method.

4.1. Prerequisites

For the merchant

The process of adding the PayPal payment option to your merchant website is simple and is based on:

- creating a **PayPal account** if you do not have one.
- using a **PayPal account** if you have one.
- enabling the **Payment via PayPal** option with the help of your payment gateway sales representative.

For the buyer

Signing up to PayPal is free and is done on the PayPal website. The buyer must:

- Create a PayPal account by entering an e-mail address and a password.
- Fill in all his or her personal details, address, etc.
- Fill in all the credit card details.

Creating a PayPal account

1. You must create a merchant account on PayPal at this address:
   https://www.paypal.com/fr/webapps/mpp/merchant

   You will have to create a PayPal merchant account in Sandbox mode (Test mode) in order to perform integration tests.

   You can do that by creating a **Private** or **Company** account on the website.

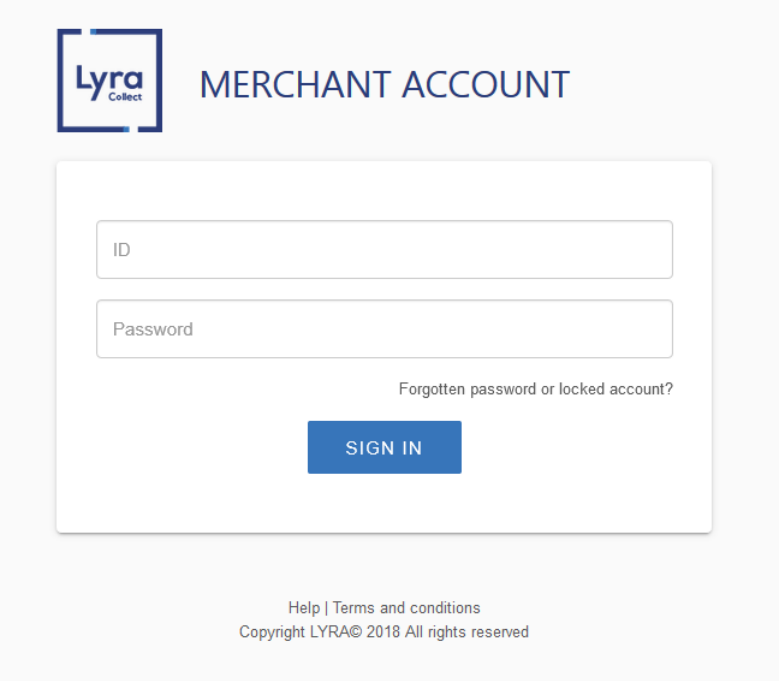
2. Select the PayPal profile that best suits your business needs.

At the end of the sign up process, PayPal will send you a recurring payment confirmation e-mail. Follow the steps to validate your subscription. Otherwise, your account will not be activated.

Signing in to the Lyra Collect Back Office

Sign in the Back Office:

https://secure.lyra.com/portal/

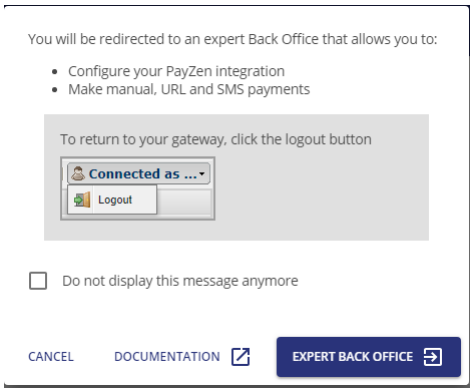


1. Enter your login.
2. Enter your password.
3. Click **Login**.

   In case of an entry error of the login and/or password, the error message "*Invalid username or password*" will appear.

   You can correct your entry or click on the link **Forgotten password or locked account**.
4. Click **Other actions**.

   The following window appears:



5. Click on **Expert Back Office** to access your Expert Back Office

Configuring your Expert Back Office for using PayPal

1. Click on **Settings** > **Company** > **Merchant IDs** tab.
2. Click on **Create a Merchant ID (MID)**.
   The **Selection of Merchant ID type** dialog box appears.
3. Select a PayPal Merchant ID type.
   The **Creation of a Merchant ID** dialog box appears.
4. Enter your **PayPal ID**.
   Your PayPal ID corresponds to the e-mail address that you used to create a PayPal account.
5. Select the currency.
   One currency per MID.
   You have the option to accept payments in multiple currencies under certain conditions. Please see chapter **Managing multi-currency payments on page 11**
6. Click on **Next**.
7. Customize your payment constraints:
   - Refund authorized
   - Maximum refund
   - Maximum gap
8. Click on **Terminate**.

Managing authorizations

To receive payments on the PayPal account, it is necessary to give it the required authorizations.

Two types of authorizations:

- An authorization for "single" payments,
- An authorization for recurring payments using a token (i.e. without having to enter buyer's bank account login credentials).

1. Click on **Settings** > **Company** > **Merchant IDs** tab.
2. Select your PayPal MID.
   The details of the PayPal MID appears in the bottom of the screen.
3. In the **Authorization management** section, click on:
   - **Give the authorization** to receive "single" payments automatically on the PayPal account
   - **Give token permission** to receive recurring payments automatically on the PayPal account

     For more information on the **Reference Transactions** feature, please refer to the Enabling the "PayPal Reference Transaction" option on page 13 chapter.

   A redirection to the PayPal authentication page is presented.
4. Click **Yes**.
5. Sing in to PayPal using your login and password.

6. Follow the instructions to accept permissions.

7. Click on **Save**.

Benefiting from the merchant protection program

PayPal has implemented a merchant protection program in the two following cases:

- An opposition or a cancellation has been made following an unauthorized payment.
  The payment must occur in PayPal environment.
- An opposition or a complaint has been filed for an unreceived object.

In both cases, **PayPal** refunds the total amount to compensate for the damage suffered, provided that the conditions are met.

In order to benefit from the protection program, one must meet the following criteria:

1. Accept **PayPal** requirements via one's account (the right to suspend accounts and/or eligibility in case of identified risk).
2. File a complaint for a transaction marked as eligible or partially eligible in the transaction details section of your account.
   In the Expert Back Office, each PayPal transaction contains a **Paypal authentication** tab within transaction details. This tab displays the Merchant protection reference with the value returned by PayPal.

   Three values are possible:

   - ELIGIBLE
     Merchant is protected by PayPal's Seller Protection Policy for unauthorized payments and Item Not Received.
   - PARTIALLY_ELIGIBLE
     Merchant is protected by PayPal's Seller Protection Policy for Item Not Received.
   - INELIGIBLE
     Merchant is not protected by PayPal's Seller Protection Policy for Item Not Received.
3. Be able to provide proof of deposit (in case of a tangible item) or proof of delivery (in case of an intangible item or service).
4. Ship the order to the shipping address specified in the transaction details.
   Hand-delivery and shipping to another address are not covered by the protection.
5. Accept a single payment via a **PayPal** account.
   Split payments or payments in installments are not covered by the protection.
6. Have the possibility to provide PayPal with all required information or documents.

This protection applies regardless of the merchant's country, but there might be differences in **PayPal** requirements. For more information on merchant protection, see *https://www.paypal.com/tc/webapps/mpp/ua/useragreement-full#9*

Association of a PayPal MID with a shop

To include the PayPal payment method on your payment page, you must associate the PayPal MID with your shop. Two options are available:

- From the **Settings** > **Company** > **Merchant IDs** tab
- From the **Settings** > **Shop** > **MID association** tab

1. Click on **Settings** > Company > **Merchant IDs** tab.
2. Select your PayPal MID.
   The details of the PayPal MID appears in the bottom of the screen.
3. Click **Associate with a shop** (lower part of the screen) to associate the MID with the shop.
4. Choose the shop(s) in the **Association of the Merchant ID (MID)** dialog box.
5. Click on **Save**.

OR

1. Click on **Settings** > Shop > **MID association** tab.
2. Select the PayPal MID in the available MID list.
3. Click **Associate** or drag and drop it to **Associated MID** on the right side.
4. Click on **Save**.

4.2. Managing multi-currency payments

It is possible to accept payments in a currency that is different from the one specified in your contract.

Your Lyra Collect contract only authorizes one currency. But you can configure your PayPal merchant account to accept payments in several currencies.

To manage multi-currency payments via your PayPal merchant account, you must:

- Manage different currencies
- Configuring your preferences for receiving payments

Managing currencies

The currency management page can be accessed in two ways:

- via **Settings > My Profile > My Money > Currency management**,
- from the homepage via the menu **My Money**.

The currency management page allows you to:

- add a new currency,
- disable a currency,
- calculate conversion rates for your currencies.

The table below presents available currencies.

List of available currencies	
Thai Baht (THB)	New Zealand Dollar (NZD)
Danish Crown (DKK)	Euro (EUR)
Norwegian Crown (NOK)	Hungarian Forint (HUF)
Swedish Crown (SEK)	Swiss Franc (CHF)
Czech Crown (CZK)	Pound Sterling (GBP)
Australian Dollar (AUD)	Taiwan New Dollar (TWD)
Canadian Dollar (CAD)	New Israeli Shekel (ILS)
Hong Kong Dollar (HKD)	Russian Ruble (RUB)
Singapore Dollar (SGD)	Japanese Yen (JPY)

List of available currencies	
US Dollar (USD)	Polish Zloty (PLN)

Configuring your preferences for receiving payments

PayPal provides you with three options when you receive a payment in a currency that is different from the one defined in your contract:

- Accept the payment and convert the transaction amount into the currency defined in the contract.
  Example: the buyer pays in **dollars** and you convert the currency to **euros**.
- Reject the payment if it is not in the currency defined in the contract.
- Put the payment on hold.
  By default, your PayPal account offers this option of putting payments on hold.
  In this case, you decide how to proceed via your PayPal account. You can:
  - accept the payment and convert the amount of the transaction into the currency defined in the contract.
  - accept the payment and open a new account in the currency used by the buyer.
    Feature available for currencies supported by PayPal , see chapter Managing currencies on page 11.
  - reject the payment.

Note

If you receive a payment in a currency that is not mentioned among the available currencies and if your account is configured with this option of putting on payments hold ***Ask me whether to accept or deny each individual payment****, you will only have two options (accept and convert into euro or reject the payment).*

You will not have the option of accepting the payment and opening a new account in this new currency.

To configure your preferences for receiving payments

1. Sign in to your PayPal merchant account.
   Your merchant area appears.
2. Click **Settings**, then **My profile**.
   The preference configuration page appears.
3. Click **My selling tools**.
   The different options available for online sales appear.
4. Select the line **Block payments**
   This option allows to configure payment limits, add instructions, etc.
5. Click **Update**.
   The **payment receiving preferences** page appears.
6. Enable the desired payment option among the 3 available options in **Allow payments sent to me in a currency I do not hold**.
   As explained in the beginning of the chapter, the option **Ask me whether to accept or deny each individual payment** is enabled by default.
   This option offers you flexibility, once the transaction is complete, to analyze each payment and decide what to do next.
7. Click **Save**.

Your modification is now taken into account.

4.3. Making recurring payments

There are two ways to make a recurring payment via PayPal:

1. Recurring payment, also called "subscription", "pre-approved payment" or "automatic payment" implemented by PayPal.

   The buyer configures their account and PayPal takes care of the payment schedule and the recurring payments after the Enabling the "PayPal Reference Transaction" option on page 13 option is enabled.

2. Payment by ID implemented by the merchant in their Expert Back Office that is used for recurring payments via PayPal.

   The payment gateway provides this second option which offers several advantages.

   The merchant increases the buyer's loyalty by providing the possibility to associate an identifier with a payment method in view of facilitating future payments on the website.

   There are several advantages to the token:

   - fast and secure payments (1-click payment),

     No need to enter bank details or sign in to a PayPal account in order to pay. Only the token is transferred during the exchange.

   - possibility to make periodic or recurring payments,

     The merchant manages the payment schedule and the recurring payments via the Expert Back Office.

     Once a token has been created, it is possible to add one or several additional recurring payments that would use this token.

The payment form allows to make the following operations:

- create a token.
- create a token during a payment.
- create a token when creating a recurring payment.
- create a token when creating a recurring payment with a payment.
- update information associated with a token.
- use a token to perform a 1-click payment.
- use a token to create a new recurring payment.
- offer the possibility to create a token during a payment.
- update information associated with a token during a payment.

For more information about recurring payments, see the *Implementation guidePayment by token-Recurring payment* available in our online document archive (*https://lyra.com/doc/en/*).

4.4. Enabling the "PayPal Reference Transaction" option

To allow the creation of payments by token and recurring payments via PayPal, the merchant must request the activation of the **PayPal Reference Transaction** option.

This option is enabled manually by your PayPal account manager or through your PayPal customer service: *https://www.paypal.com/fr/selfhelp/contact/call*

The delay for enabling the option can take from several days to several weeks.

4.5. Viewing the progress of a PayPal payment on the payment page

The diagram below presents the exchange process from the point of view of the buyer.

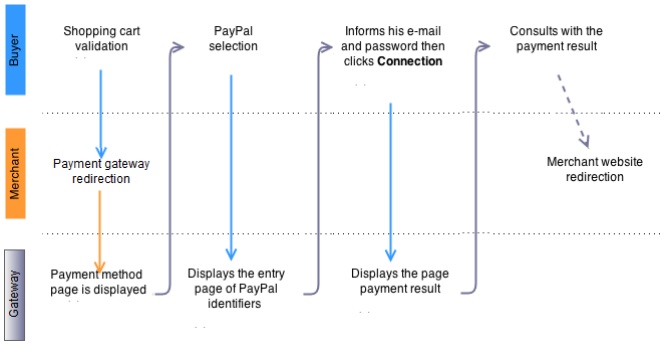


Figure 1: Payment process - as seen by the buyer

1. The buyer validates the shopping cart.
2. The merchant website redirects the buyer to the payment gateway. The redirection is done via an HTML POST form using HTTPS. The parameters of the form are described in the chapter **Generating a payment form**.
3. When the parameters and their signature have been verified, the payment method selection page appears.

   If the payment method has been specified in the form, the buyer moves on directly to step 7.
4. The buyer selects the **PayPal** payment method.
5. The buyer clicks **Validate**.

   The payment gateway prompts you to wait several seconds. It redirects you to the PayPal website.
6. The buyer is prompted to select their payment mode.

   The buyer can sign in to their PayPal account to pay or create a PayPal account to pay faster in the future.
7. If the buyer has a PayPal account, they sign in using their e-mail address and password.

   If the buyer does not have a PayPal account, they select their card and fill in the required details.
8. The buyer verifies the information related to their payment.
9. The buyer clicks **Pay**.
10. The buyer is redirected to the payment gateway.

    In case of success, a summary page is presented to the buyer with a summary of transaction details.

    A link at the bottom of the page allows to return to the shop.

    In case the transaction fails, a message appears. The buyer is informed of the payment request rejection.

    A link at the bottom of the page allows to return to the shop.

4.6. Viewing PayPal transactions in the Expert Back Office

Transactions can be viewed in the Expert Back Office via the **Management** > **Transactions** menu.

Reminder: for immediate payments, the capture delay is equal to zero. This means that the funds are captured directly. Therefore, there is no authorization.

The PayPal transaction appears directly in the **Captured transactions** tab.

1. Select an PayPal transaction.
2. Rick click and select **Display transaction details**.

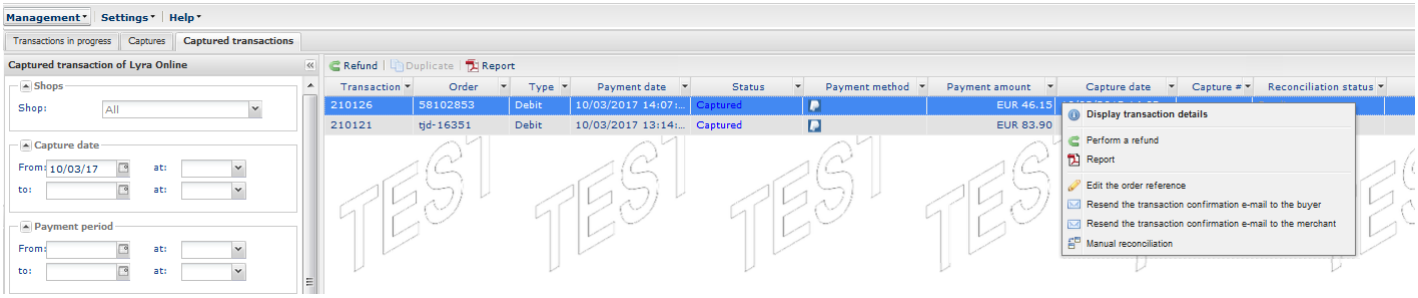
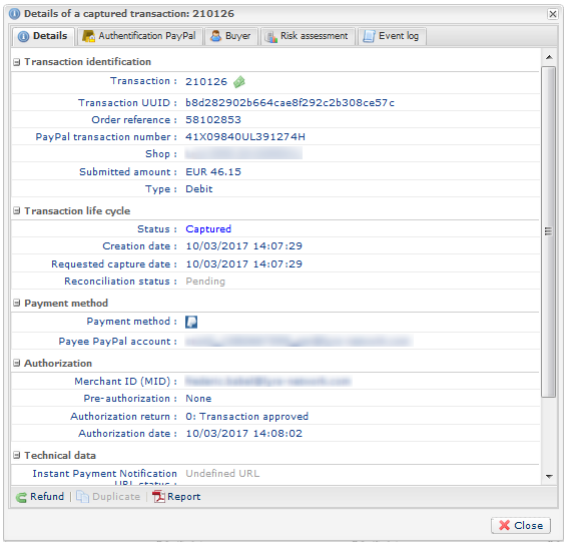


Figure 2: Captured transactions - PayPal tab

The **Details of a captured transaction** dialog box appears.



The details include:

- **Transaction type**

  Debit in case of a payment Credit in case of a refund

- **The transaction amount**
- **The transaction status**

  "Presented" status for an immediate payment

- **The PayPal account of the buyer who performed the payment**
- **The PayPal certificate of the transaction**

In some cases, PayPal retains a transaction because of suspected fraud. In this case:

- The payment appears in the **Transactions in progress** tab with the status **Control in progress**.
- The **vads_result** field is returned with the 00 value during the call to notification URL (IPN). It is important to also check in your script that the **vads_trans_status** field is not set to UNDER_VERIFICATION (control in progress by PayPal).

4.7. Learn more about the payment guarantee with PayPal

Payments are not guaranteed with PayPal.

However, when a payment is accepted it is also transferred.

If you are using the Order mode (deferred payment), it is important to respect the capture delays.

However, as soon as the capture has been made, the payment is transferred without exception. If PayPal returns the payment accepted status, the payment is always transferred.

5. ESTABLISHING INTERACTION WITH THE PAYMENT GATEWAY

The interaction with the payment gateway is described in the *Hosted Payment Page Implementation guide*, available in the website.

The merchant website and the payment gateway interact by exchanging data.

To create a payment, this data is sent in an HTML form via the buyer's browser.

At the end of the payment, the result is transmitted to the merchant website in two ways:

- automatically by means of notifications called Instant Notification URLs (also known as IPN for Instant Payment Notification).
- by the browser when the buyer clicks the button to return to the merchant website.

To guarantee the security of the exchange, the data is signed with a key only known to the merchant and the payment gateway.

6. GENERATING A PAYMENT FORM

To generate a payment request, you must create an HTML form as follows:

```
<form method="POST" action="https://secure.lyra.com/vads-payment/">
 <input type="hidden" name="parameter1" value="value1" />
 <input type="hidden" name="parameter2" value="value2" />
 <input type="hidden" name="parameter3" value="value3" />
 <input type="hidden" name="signature" value="signature"/>
 <input type="submit" name="pay" value="Pay"/>
 </form>
```

It contains:

The following technical elements:

- The `<form>` and `</form>` tags that allow to create an HTML form.
- The `method="POST"` attribute that defines the method used for sending data.
- The `action="https://secure.lyra.com/vads-payment/"` attribute that defines where to send the form data.

Form data:

- The shop ID.
- Information about the payment depending on the use case (see the chapters below).
- Additional information, depending on your requirements (see **Using additional features**).
- Signature that certifies the integrity of the form (see **Computing the signature**).

This data is added to the form by using the `<input>` tag:

```
<input type="hidden" name="parameter1" value="value1" />
```

For setting the `name` and `value` attributes, see chapter **Data dictionary**.

All the data in the form must be encoded in **UTF-8.**

Special characters (accents, punctuation marks, etc.) will then be correctly interpreted by the payment gateway. Otherwise, the signature will not be computed correctly and the form will be rejected.

The **Pay** button that will allow to send data:

```
<input type="submit" name="pay" value="Pay"/>
```


Different use cases are presented in the chapters below. They will allow you to adapt your payment form to your needs.

The fields required to implement these use cases are presented in tables that indicate the required format (see the encoding below)

Notation	Description
a	Alphabetic characters (from 'A' to 'Z' and from 'a' to 'z')
n	Numeric characters
s	Special characters
an	Alphanumeric characters
ans	Alphanumeric and special characters (except '<' and '>')
3	Fixed length of 3 characters
..12	Variable length up to 12 characters
json	JavaScript Object Notation. Object containing key/value pairs separated by commas. It starts with a left brace ' `{` ' and ends with a right brace ' `}`'. Each key / value pair contains the name of the key between double-quotes followed by " `:` ", followed by a value. The name of the key must be alphanumeric. The value can be: • a chain of characters (in this case it must be framed by double-quotes) • a number • an object • a table • a boolean • empty Example: `{"name1":45,"name2":"value2", "name3"=false}`
enum	Characterizes a field with a complete list of values. The list of possible values is given in the field definition.
Enum list	List of values separated by a " `;` ". The list of possible values is given in the field definition. Example: `vads_payment_cards=VISA;MASTERCARD`
map	List of key / value pair separated by a "`;`". Each key / value pair contains the name of the key followed by "`=`", followed by a value. The value can be: • a chain of characters • a boolean • a json object • an xml object The list of possible values for each key/value pair is provided in the field definition. Example: `vads_theme_config=SIMPLIFIED_DISPLAY=true;RESPONSIVE_MODEL=Model_1`

6.1. Creating an immediate payment with PayPal

In the immediate payment mode the buyer pays the total amount for the purchase at once.

The payment is captured by the bank on the same day.

1. Use all the fields presented in the table below to create your payment form.

Field name	Description	Value
vads_site_id	Shop ID	E.g.: 12345678
vads_ctx_mode	Operating mode	**TEST** or **PRODUCTION**
vads_trans_id	Transaction number	E.g.: 123456
vads_trans_date	Date and time of the payment form in UTC format	E.g.: 20140129130025
vads_amount	Payment amount in the smallest currency unit (cents for euro).	E.g.: 3000 for 30,00 EUR
vads_currency	Code of the currency used for the payment	E.g.: 978 for euro (EUR)
vads_action_mode	Card data acquisition mode	**SILENT** or **INTERACTIVE**
vads_page_action	Action to perform	**PAYMENT**
vads_version	Version of the exchange protocol	**V2**
vads_payment_config	Payment type	**SINGLE**
vads_payment_card	Payment method	**PAYPAL**
vads_capture_delay	Capture delay	**0** or **absent** or **empty**
vads_validation_mode	Specifies the validation mode of the transaction (manually by the merchant, or automatically by the payment gateway).	**0** or **absent** or **empty**

Table 1: Field list - Immediate payment

2. Set the **vads_payment_config** field to **SINGLE**.

3. Set the **vads_capture_delay** field to **0** or leave it **empty**. No other values should be transmitted. Otherwise, your form will be rejected.

4. Set the **vads_payment_card** field to **PAYPAL** if you wish to show only the PayPal payment method on your payment page.

   If you wish to make test payments, set the **vads_payment_card** field to **PAYPAL_SB**.

5. Populate the **vads_currency** field with the code of the desired currency using the table below:

Currency	ISO 4217 encoding	Number of digits after the decimal point
Canadian Dollar (CAD)	124	2
Danish Crown (DKK)	208	2
Japanese Yen (JPY)	392	0
Norwegian Crown (NOK)	578	2
Pound Sterling (GBP)	826	2
US Dollar (USD)	840	2
Euro (EUR)	978	2
Polish Zloty (PLN)	985	2

6. Set the **vads_validation_mode** field to **0** or leave it **empty**.

   For immediate payments, validation must be automatic. Hence, regardless of the validation mode specified in your Expert Back Office or the value of the **vads_validation_mode** field, validation will be automatic. If you choose the manual validation mode, it will be ignored and the payment will be automatically validated.

7. Add optional fields according to your requirements (see chapter **Using additional features on page 25**).

8. Compute the value of the **signature** field using all the fields of your form that start with **vads_** (see chapter **Computing the signature on page 32**).

Example of a form for an immediate payment:

```
<form method="POST" action="https://secure.lyra.com/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="3000" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="91335531" />
<input type="hidden" name="vads_trans_date" value="20140526101407" />
<input type="hidden" name="vads_trans_id" value="239848" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value=""/>
<input type="submit" name="pay" value="Pay"/>
</form>
```

6.2. Creating a payment by identifier / token

Use all the fields presented in the table below to create your form.

Field name	Description	Format	Value
vads_page_action	Action to perform	string (enum)	Several possible values: • REGISTER Create a token • REGISTER_PAY Create a token during a payment • REGISTER_SUBSCRIBE Create a token when creating a recurring payment • REGISTER_ PAY_SUBSCRIBE Create a token when creating a recurring payment with payment • REGISTER_UPDATE Update information associated with the token • PAYMENT Use a token to perform a 1-click payment • SUBSCRIBE Use a token to create a new recurring payment • ASK_ REGISTER_PAY Offer the possibility to create a token during a payment • REGISTER_UPDATE_PAY Update information associated with the token during a payment
vads_amount	Payment amount in the smallest currency unit (cents for euro).	n..12	E.g.: 3000 for 30,00 EUR
vads_ctx_mode	Operating mode.	string (enum)	**TEST** or **PRODUCTION**

Field name	Description	Format	Value
vads_currency	Code of the currency used for the payment.	n3	E.g.: 978 for euro (EUR)
vads_action_mode	Acquisition mode for payment method data.	string (enum)	**INTERACTIVE**
vads_identifier	Token (unique) associated with a payment method.	string	• This identifier can either be generated by the payment gateway In this case, this parameter must not be populated. • Otherwise, it can be generated by the merchant website In this case, this parameter must be populated with the desired value of the identifier. **The merchant website must make sure that each identifier is unique.** Any registration request containing an existing identifier will be rejected and an error message will appear.
vads_payment_config	Payment type	string (enum)	**SINGLE**
vads_site_id	Shop ID	n8	E.g.: 12345678
vads_trans_date	Date and time of the payment form in UTC format.	n14	E.g.: 20170501130025
vads_trans_id	Unique transaction ID.	n6	E.g.: 123456
vads_version	Version of the exchange protocol.	string	**V2**

Table 2: Field list - Payment by token PayPal

7. USING ADDITIONAL FEATURES

PayPal has put in place a program of merchant protection that allows to protect a merchant in case of a legal dispute with a buyer.

In order to benefit from this protection when selling physical objects, one **must** transmit **the product description** and **the shipping address** to the payment platform so that it can, in turn, transmit this information to PayPal.

The shipping details correspond to all the fields starting with **vads_ship_**. For more information, see chapter **Transmitting shipping details**.

If this data is not transmitted to PayPal, the merchant will not be able to benefit from any protection.

The protection is not automatically attributed to all merchants. For more information, please contact PayPal or visit their website at *https://www.paypal.com/fr/webapps/mpp/paypal-safety-and-security*.

To obtain a custom form adapted to your needs, you can use additional optional features from the list below:

- Manage the payment methods offered to the buyer on the payment page
- Transmit buyer details (title, e-mail address, etc.)
- Transmit shipping details (address, etc.)
- Transmit order details (reference, shopping cart contents, etc.)

7.1. Managing the payment methods offered to the buyer

It is possible to customize the payment methods offered to the buyer.

1. Use all the fields required for your use case (see chapter **Generating a payment form on page 20**) to create your payment form.
2. Set the **vads_payment_cards** field.
   - with one single value, if you do not wish to show the page of payment method selection.
   - with a list of values separated by ";" to show the page of payment method selection.
3. Compute the value of the **signature** field using all the fields of your form that start with **vads_** (see chapter **Computing the signature on page 32**).

Example of a payment form with payment method selection:

```
<form method="POST" action="https://secure.lyra.com/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="30000" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="PRODUCTION" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_cards" value="VISA;MASTERCARD;PAYPAL" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20140526101407" />
<input type="hidden" name="vads_trans_id" value="239848" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="qqpxF6z1+Ri5jtkHNVDCCJulxxpJYehrfP1OLwJ4Ysg="/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


7.2. Transmitting shipping details

The merchant can transmit the buyer's shipping details (e-mail address, title, phone number, etc.).

This information can be found in the transaction details in the Expert Back Office (**Shipping** tab).

1. Use the fields required for your use case (see chapter **Generating a payment form**) to create your payment form.

2. Use optional fields according to your requirements.

   These fields will be returned in the response and will include the value transmitted in the form.

Field name	Description	Format	Value
vads_ship_to_city	City	an..128	E.g.: Bordeaux
vads_ship_to_country	Country code in compliance with the ISO 3166 standard.	a2	E.g.: FR
vads_ship_to_district	District	ans..127	E.g.: La Bastide
vads_ship_to_first_name	First name	ans..63	E.g.: Albert
vads_ship_to_last_name	Name	ans..63	E.g.: Durant
vads_ship_to_legal_name	Legal name	an..100	E.g.: D. & Cie
vads_ship_to_phone_num	Phone number	ans..32	E.g.: 0460030288
vads_ship_to_state	State / Region	ans..127	E.g.: Nouvelle Aquitaine
vads_ship_to_status	Allows to specify the type of the shipping address.	enum	**PRIVATE**: for shipping to a private individual **COMPANY**: for shipping to a company
vads_ship_to_street_number	Street number	ans..64	E.g.: 2
vads_ship_to_street	Postal address	ans..255	E.g.: Rue Sainte Catherine
vads_ship_to_street2	Second line of the address	ans..255	
vads_ship_to_zip	Zip code	an..64	E.g.: 33000

Table 3: Field list - Shipping details

3. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

7.3. Transmitting order details

The merchant can indicate in their payment form if they wish to transfer the order details (order reference, description, shopping cart contents, etc.).

This information can be found in the transaction details in the Expert Back Office (**Shopping cart** tab).

1. Use the fields required for your use case (see chapter **Generating a payment form**) to create your payment form.

2. Use optional fields according to your requirements. These fields will be returned in the response and will include the value transmitted in the form.

Field name	Description	Format	Value
vads_order_id	Order ID	ans..64	E.g.: 2-XQ001
vads_order_info	Complementary order info	an..255	E.g.: Door phone code 3125
vads_order_info2	Complementary order info	an..255	E.g.: No elevator
vads_order_info3	Complementary order info	an..255	E.g.: Express
vads_nb_products	Number of items in the cart	n..12	E.g.: 2
vads_product_labelN	Item name. N corresponds to the index of the item (0 for the	an..255	E.g.: vads_product_label0 = "tee-shirt" vads_product_label1 = "Biscuit"

Field name	Description	Format	Value
	first one, 1 for the second one, etc.).		vads_product_label2 = "sandwich"
vads_product_amountN	Item amount. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	n..12	E.g.: vads_product_amount0 = "1200" vads_product_amount1 = "800" vads_product_amount2 = "950"
vads_product_typeN	Item type. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	enum	E.g.: vads_product_type0 = "CLOTHING_AND_ACCESSORIES" vads_product_type1 = "FOOD_AND_GROCERY" vads_product_type2 = "FOOD_AND_GROCERY"
vads_product_refN	Item reference. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	an..64	E.g.: vads_product_ref0 = "CAA-25-006" vads_product_ref1 = "FAG-B5-112" vads_product_ref2 = "FAG-S9-650"
vads_product_qtyN	Quantity of items. N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).	n..12	E.g.: vads_product_qty0 = "1" vads_product_qty1 = "2" vads_product_qty2 = "2"

Table 4: Field list - Order details

3. Populate the **vads_nb_products** field with the number of items contained in the cart.

   ***Note**:*

   This field becomes mandatory for the shopping cart to be taken into account.

   *When it is populated, the **Shopping cart** tab becomes available in the transaction details in the Expert Back Office.*

   *However, if the other fields that start with **vads_product_** are not populated, the tab will not include any information. For this reason, when populating the **vads_nb_products** field, it becomes mandatory to populate the other fields that start with **vads_product_**.*

4. Populate the **vads_product_amountN** field with the amount for the items in the cart, using the smallest currency unit.

   N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).

5. Populate **vads_product_typeN** with the value corresponding to the item type.

   N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).

Value	Description
FOOD_AND_GROCERY	Food and grocery
AUTOMOTIVE	Cars / Moto
ENTERTAINMENT	Entertainment / Culture
HOME_AND_GARDEN	Home and gardening
HOME_APPLIANCE	Household appliances
AUCTION_AND_GROUP_BUYING	Auctions and group purchasing
FLOWERS_AND_GIFTS|	Flowers and presents
COMPUTER_AND_SOFTWARE	Computers and software
HEALTH_AND_BEAUTY	Health and beauty
SERVICE_FOR_INDIVIDUAL	Services for individuals
SERVICE_FOR_BUSINESS	Services for companies
SPORTS	Sports
CLOTHING_AND_ACCESSORIES	Clothes and accessories
TRAVEL	Travel
HOME_AUDIO_PHOTO_VIDEO	Sound, image and video

Value	Description
TELEPHONY	Telephony

Table 5: Values associated with vads_product_type0

6. Populate **vads_product_labelN** with the name of each item contained in the cart.
   N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).

7. Populate **vads_product_qtyN** with the quantity of each item contained in the cart.
   N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).

8. Populate **vads_product_refN** with the reference of each item contained in the cart.
   N corresponds to the index of the item (0 for the first one, 1 for the second one, etc.).

9. Check the value of the **vads_amount** field. It must correspond to the total amount of the order.

10. Compute the value of the **signature** field using all the fields of your form starting with **vads_** (see chapter **Computing the signature**).

Example of the payment form with cart description:

```
<form method="POST" action="https://secure.lyra.com/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="7500" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="PRODUCTION" />
      <input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_insurance_amount" value="500" />
<input type="hidden" name="vads_nb_products" value="2"/>
<input type="hidden" name="vads_product_amount0" value="5000" />
<input type="hidden" name="vads_product_label0" value="produit1" />
<input type="hidden" name="vads_product_qty0" value="2" />
<input type="hidden" name="vads_product_ref0" value="ref1" />
<input type="hidden" name="vads_product_amount1" value="1000" />
<input type="hidden" name="vads_product_label1" value="produit2" />
<input type="hidden" name="vads_product_qty1" value="1" />
<input type="hidden" name="vads_product_ref1" value="ref2" />
<input type="hidden" name="vads_order_id" value="CD100000857" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_shipping_amount" value="500" />
<input type="hidden" name="vads_tax_amount" value="500" />
<input type="hidden" name="vads_trans_date" value="20140327145218" />
<input type="hidden" name="vads_trans_id" value="571381" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="xYw1UnU3BACGhf3UEyqbQzpwuvZDEkCAWAE5fgbtfxI="/>
<input type="submit" name="pay" value="Pay"/></form>
```

7.4. Transmitting buyer details

The merchant can specify the buyer's billing details (e-mail address, title, phone number, etc.). This information will be used to create the invoice.

All the data transmitted via the payment form can be viewed in the transaction details in the Expert Back Office (**Buyer** tab).

1. Use the fields required for your use case (see chapter **Generating a payment form**) to create your payment form.

2. Use optional fields according to your requirements. *These fields will be returned in the response and will include the value transmitted in the form.*

Field name	Description	Format	Value
vads_cust_email	Buyer's e-mail address	ans..150	E.g.: abc@example.com
vads_cust_id	Buyer reference on the merchant website	an..63	E.g.: C2383333540
vads_cust_title	Buyer's title	an..63	E.g.: M.
vads_cust_status	Status	enum	**PRIVATE**: for a private individual **COMPANY**: for a company
vads_cust_first_name	First name	ans..63	E.g.: Laurent
vads_cust_last_name	Name	ans..63	E.g.: Durant
vads_cust_legal_name	Buyer's legal name	an..100	E.g.: D. & Cie
vads_cust_cell_phone	Cell phone number	an..32	E.g.: 06 12 34 56 78
vads_cust_address_number	Street number	ans..64	E.g.: 109
vads_cust_address	Postal address	ans..255	E.g.: Rue de l'innovation
vads_cust_address2	Second line of the address	ans..255	E.g.:
vads_cust_district	District	ans..127	E.g.: Downtown
vads_cust_zip	Zip code	an..64	E.g.: 31670
vads_cust_city	City	an..128	E.g.: Labège
vads_cust_state	State / Region	ans..127	E.g.: Occitanie
vads_cust_country	Country code in compliance with the ISO 3166 standard	a2	E.g.: "FR" for France, "PF" for French Polynesia, "NC" for New Caledonia, "US" for the United States

3. Compute the value of the **signature** field using all the fields of your form that start with **vads_** (see chapter **Computing the signature**).

```
<form method="POST" action="https://secure.lyra.com/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="4000" />
<input type="hidden" name="vads_capture_delay" value="0" />
<input type="hidden" name="vads_ctx_mode" value="PRODUCTION" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_cust_country" value="FR" />
<input type="hidden" name="vads_cust_email" value="dupont.albert@exemple.com" />
<input type="hidden" name="vads_cust_first_name" value="Albert" />
<input type="hidden" name="vads_cust_last_name" value="Dupont" />
<input type="hidden" name="vads_cust_title" value="M." />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20140327133115" />
<input type="hidden" name="vads_trans_id" value="522754" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="7896adcaf7338930db9715afa123531f42"/>
<input type="submit" name="pay" value="Pay"/>
</form>
```


8. VERIFYING FIELDS CONSISTENCY

Your form contains the **vads_amount** field. This field corresponds to the total amount of the order.

To make sure your form is valid, the payment platform verifies the entered value.

The equation used is as follows:

Sum (vads_product_qty(N) x vads_product_amount(N)) + vads_shipping_amount + vads_tax_amount + vads_insurance_amount = vads_amount

If the value entered in the **vads_amount** field is incorrect, your payment form is rejected due to invalid amount.

9. COMPUTING THE SIGNATURE

To be able to compute the signature, you must have:

- all the fields that start with **vads_**
- the signature algorithm chosen in the shop configuration
- the **key**

The value of the key is available in your Expert Back Office via **Settings** > **Shop** > **Keys** tab.

The signature algorithm is defined in your Expert Back Office via **Settings** > **Shop** > **Configuration** tab.

For maximum security, it is recommended to use HMAC-SHA-256 algorithm and an alphanumeric key.

To compute the signature:

1. Sort the fields that start with **vads_** alphabetically.
2. Make sure that all the fields are encoded in UTF-8.
3. Concatenate the values of these fields separating them with the "**+**" character.
4. Concatenate the result with the test or production key separating them with a "**+**".
5. According to the signature algorithm defined in your shop configuration:
   a. if your shop is configured to use "SHA-1", apply the **SHA-1** hash function to the chain obtained during the previous step.
   b. if your shop is configured to use "HMAC-SHA-256", compute and encode in Base64 format the message signature using the **HMAC-SHA-256** algorithm with the following parameters:
      - the SHA-256 hash function,
      - the test or production key (depending on the value of the **vads_ctx_mode** field) as a shared key,
      - the result of the previous step as the message to authenticate.
6. Save the result of the previous step in the **signature** field.

Example of parameters sent to the payment gateway:

```
<form method="POST" action="https://secure.lyra.com/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="5124" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20170129130025" />
<input type="hidden" name="vads_trans_id" value="123456" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="ycA5Do5tNvsnKdc/eP1bj2xa19z9q3iWPy9/rpesfS0= " />

<input type="submit" name="pay" value="Pay"/>
</form>
```

This sample form is analyzed as follows:

1. The fields whose names start with **vads_** are sorted **alphabetically**:
   - vads_action_mode
   - vads_amount
   - vads_ctx_mode
   - vads_currency
   - vads_page_action
   - vads_payment_config
   - vads_site_id
   - vads_trans_date
   - vads_trans_id
   - vads_version
2. The values of these fields are concatenated using the "**+**" character:

   ```
   INTERACTIVE+5124+TEST+978+PAYMENT+SINGLE+12345678+20170129130025+123456+V2
   ```

3. The value of the test key is added at the end of the chain and separated with the "**+**" character. In this example, the test key is **1122334455667788**

   ```
   INTERACTIVE+5124+TEST+978+PAYMENT+SINGLE+12345678+20170129130025+123456+V2+1122334455667788
   ```

4. If you use the SHA-1 algorithm, apply it to the obtained chain.

   The result that must be transmitted in the signature field is: **59c96b34c74b9375c332b0b6a32e6deeec87de2b**

5. If your shop is configured to use "HMAC-SHA-256", compute and encode in Base64 format the message signature using the **HMAC-SHA-256** algorithm with the following parameters:
   - the SHA-256 hash function,
   - the test or production key (depending on the value of the **vads_ctx_mode** field) as a shared key,
   - the result of the previous step as the message to authenticate.

   The result that must be transmitted in the signature field is:

   ycA5Do5tNvsnKdc/eP1bj2xa19z9q3iWPy9/rpesfS0=

10. SENDING THE PAYMENT REQUEST

The buyer will be able to finalize his/her purchase once he/she is redirected to the payment page.

The buyer's browser must transmit the payment form data.

10.1. Redirecting the buyer to the payment page

The URL of the payment gateway is:

https://secure.lyra.com/vads-payment/

Example of parameters sent to the payment gateway:

```
<form method="POST" action="https://secure.lyra.com/vads-payment/">
<input type="hidden" name="vads_action_mode" value="INTERACTIVE" />
<input type="hidden" name="vads_amount" value="2990" />
<input type="hidden" name="vads_ctx_mode" value="TEST" />
<input type="hidden" name="vads_currency" value="978" />
<input type="hidden" name="vads_cust_country" value="FR" />
<input type="hidden" name="vads_cust_email" value="me@example.com" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_config" value="SINGLE" />
<input type="hidden" name="vads_site_id" value="12345678" />
<input type="hidden" name="vads_trans_date" value="20190626101407" />
<input type="hidden" name="vads_trans_id" value="362812" />
<input type="hidden" name="vads_version" value="V2" />
<input type="hidden" name="signature" value="NM25DPLKEbtGEHCDHn8MBT4ki6aJI/ODaWhCzCnAfvY="/>
<input type="submit" name="payer" value="Payer"/>
</form>
```

10.2. Processing errors

If the payment gateway detects an error while receiving the form, an error message will appear and the buyer will not be able to proceed to the payment.

In TEST mode

The message indicates the source of the error and provides a link to the error code description to help you fix it.

In PRODUCTION mode

The message simply indicates to the buyer that a technical problem has occurred.

In both cases the merchant receives a notification e-mail.

It contains:

- the source of the error,
- a link to possible causes to facilitate its analysis,
- all the fields of the form.

A description of the error codes with their possible causes is available on our website

https://lyra.com/doc/en/collect/error-code/error-00.html

11. ANALYZING THE PAYMENT RESULT

The analysis of the payment result is described in the *Hosted Payment Page Implementation guide* available in our online documentation archive.

This document only describes the steps of processing data relative to the response of a payment.

11.1. Retrieving data returned in the response

The data returned in the response depends on the parameters sent in the payment form, the payment type and the settings of your shop This data constitutes a field list. Each field contains a response value. The field list can be updated.

The data is always sent by the payment gateway using the **POST** method.

The first step consists in retrieving the contents received via the POST method.

Examples:

- In PHP, data is stored in the super global variable **$_POST**,
- In ASP.NET (C#), you must use the **Form** property of the **HttpRequest** class.
- In Java, you must use the **getParameter** method of the **HttpServletRequest** interface.

  The script will have to create a loop to retrieve all the transmitted fields.

Example of data sent during a PayPal payment notification:

```
vads_validation_mode = 0
vads_auth_mode = FULL
vads_page_action = PAYMENT
vads_warranty_result = NO
vads_currency = 978
vads_payment_src = EC
vads_cust_email = buyer.account@example.com
vads_threeds_cavv =
vads_threeds_sign_valid =
vads_threeds_cavvAlgorithm =
vads_order_id = de1584
vads_contract_used = merchant.account-paypal@example.com
vads_threeds_xid =
vads_capture_delay = 0
vads_auth_number =
vads_threeds_enrolled =
vads_threeds_eci =
vads_effective_currency = 978
vads_card_brand = PAYPAL_SB
vads_payment_config = SINGLE
vads_authent_paypal_protection_eligibility = ELIGIBLE
vads_language = fr
vads_ext_trans_id = 99R37988MC077294L
vads_operation_type = DEBIT
signature = 4cfe755d6bd2b29612244596872f362e4c653f98
vads_sequence_number = 1
vads_threeds_error_code =
vads_card_number = buyer.account@example.com
vads_payment_certificate = 1430532e2451a51a94508497b03b444b1d13dc5e
vads_result = 00
vads_trans_uuid = b91a50433d67411a841f59a2d104c0ba
vads_trans_date = 20190226103649
vads_ctx_mode = TEST
vads_action_mode = INTERACTIVE
vads_threeds_status =
vads_effective_amount = 365
vads_version = V2
vads_presentation_date = 20190226103716
vads_trans_status = CAPTURED
vads_pays_ip = FR
vads_trans_id = 417702
vads_auth_result = 00
vads_extra_result =
```



```
vads_threeds_exit_status =
vads_card_country =
vads_amount = 365
vads_effective_creation_date = 20190226103716
vads_site_id = 91335531
```

11.2. Computing the IPN signature

The signature is computed by following the same procedure as for creating the payment form.

IMPORTANT

The data submitted by the payment gateway is encoded in UTF-8. Any alteration of received data will result in signature computation error.

You must compute the signature with the fields received in the notification and not the ones that you transmitted in your payment form.

To compute the signature:

1. Take all the fields whose name starts with **vads_**.
2. Sort these fields alphabetically.
3. Concatenate the values of these fields separating them with the "**+**" character.
4. Concatenate the result with the test or production key separating them with a "**+**".
5. According to the signature algorithm defined in your shop configuration:
   a. if your shop is configured to use "SHA-1", apply the **SHA-1** hash function to the chain obtained during the previous step.
   b. if your shop is configured to use "HMAC-SHA-256", compute and encode in Base64 format the message signature using the **HMAC-SHA-256** algorithm with the following parameters:
      - the SHA-256 hash function,
      - the test or production key (depending on the value of the **vads_ctx_mode** field) as a shared key,
      - the result of the previous step as the message to authenticate.

11.3. Comparing signatures

In order to make sure the response is complete, you must compare the value of the **signature** field received in the response with the one computed in the step "Computing the IPN signature".

IMPORTANT
You must not compare the IPN signature with the signature that you transmitted in your form.

If the signatures match,

- You may consider the response as safe and proceed with the analysis.
- If they do not, the script will have to throw an exception and warn the merchant .

The signatures may not match because of:

- an implementation error (error in your calculation, problem with UTF-8 encoding, etc.),
- an error in the value of the key or in the **vads_ctx_mode** (field value - frequent issue when going to live mode),
- a data corruption attempt.

11.4. Analyzing the nature of the notification

The **vads_url_check_src** field allows to differentiate the notifications based on their triggering event:

- creation of a transaction,
- new notification sent by the merchant via the Expert Back Office.

It specifies the applied notification rule:

Value	Applied rule
PAY	The PAY value is sent in the following cases: • immediate payment (or first installment payment of a recurring payment) • payment deferred for less than 7 days • payment abandoned or canceled by the buyer only if the merchant has configured the rule **Instant Payment Notification URL on cancellation**
BO	Execution of the notification via the Expert Back Office (right-click a transaction > **Send the Instant Payment Notification**).
BATCH	The BATCH value is sent in case of an update of a transaction status after its synchronization on the acquirer side. Only if the merchant has configured the rule **Instant Payment Notification URL on batch change**.
BATCH_AUTO	The BATCH_AUTO value is sent in the following cases: • payment deferred for more than 7 days • installments of a recurring payment (except the first one) only if the merchant has configured the rule **Instant Payment Notification URL on batch authorization** The notification is sent with the authorization request for payments with "Waiting for authorization" status.
REC	The REC value is sent only for recurring payments if the merchant has configured the **Instant Payment Notification URL when creating recurring payments** rule.
MERCH_BO	The MERCH_BO value is sent: • during operation made via the Expert Back Office (refund, cancellation, modification, validation, duplication, creation and/or update of token), only if the merchant has configured the following

Value	Applied rule
	notification rule: **Instant Payment Notification URL on an operation coming from the Back Office**
RETRY	Automatic retry of the IPN.

Table 6: Values associated with the vads_url_check_src field

After checking its value, the script can process differently depending on the nature of the notification.

For example:

If **vads_url_check_src** is set to **PAY** or **BATCH_AUTO**, the script will update the order status, etc.

If **vads_url_check_src** is set to **REC** the script will retrieve the recurring payment reference and will increment the number of the expired installment payments in case the payment has been accepted, etc.

11.5. Identifying the type of operation

The **vads_operation_type** field allows to distinguish:

- a debit operation
- a refund operation

Value	Description
DEBIT	Debit transaction.
CREDIT	Refund operation
VERIFICATION	Checking operation during token creation or token update without payment.

Table 7: Value of the vads_operation_type field

For example:

If **vads_operation_type** is set to **DEBIT**, the script updates the order and registers the transaction details.

If **vads_operation_type** is set to **CREDIT**, the script updates the paid amount or adds a new transaction line in the order.

11.6. Processing response data - immediate payment with PayPal

Here is an example of analysis to guide you through processing the response data.

1. Identify the payment result by testing the value of the **vads_trans_status** field

Field name	Description	Value
vads_trans_status	Status of the payment	Possible values: • **CAPTURED** Captured The transaction has been captured by the bank. • **CANCELLED** Canceled Cancellation by the buyer, the acquirer or the payment gateway. • **REFUSED** Declined • **EXPIRED** Expired The expiry date of the authorization request has passed and the merchant has not validated the transaction. The account of the cardholder will, therefore, not be debited.

Field name	Description	Value
		• **UNDER_VERIFICATION** Control in progress For PayPal transactions, this value means that PayPal withholds the transaction for suspected fraud. The payment will remain in the **Transactions in progress** tab until the verification process has been completed. The transaction will then take one of the following statuses: **AUTHORISED** or **CANCELED**. A notification will be sent to the merchant to warn them about the status change (Instant Payment Notification on batch change).

Table 8: Values associated with the vads_trans_status field

2. Record the order while identifying the value of the **vads_order_id** field if you have transmitted it to the payment gateway.

3. Record the payment reference while identifying the value of the **vads_trans_id** field.

4. Analyze the **vads_payment_config** field to make sure that it is an **immediate payment** (single payment).

5. Store the value of the **vads_trans_date** field to identify the payment date.

6. Save the amount and the used currency. To do this, retrieve the values of the following fields:

Field name	Description
vads_amount	Payment amount in the smallest currency unit.
vads_currency	Code of the currency used for the payment.
vads_change_rate	Used exchange rate.
vads_effective_amount	Payment amount in the currency used for the capture by the bank.

Table 9: Fields for analyzing the payment amount and currency

7. Record the mode and the result of the authorization. To do this:

   a. Store the value of the **vads_auth_mode** field to identify the mode of the authorization request.

Value	Description
FULL	An authorization request has been made for the full amount.
MARK	A pre-authorization request for 1 EUR (or information request about the CB network if the acquirer supports it) has been made to verify the validity of the card.

*Table 10: Values of the **vads_auth_mode** field*

   b. Note the value of the **vads_auth_result** field to identify the result of the authorization request.

   For an accepted authorization, the **vads_auth_result** field is set to **00**.

   All other results indicate that the authorization has been rejected.

   Here is a list of frequently returned codes to help you understand the reason of the rejection:

Value	Description
0	Transaction accepted
10001	Internal error
10009	Transaction refused for one of the following reasons: • The partial refund amount must be less than or equal to the original transaction amount. • The partial refund must be the same currency as the original transaction. • This transaction has already been fully refunded. • You are over the time limit to perform a refund on this transaction (60 days).
10422	Customer must choose new funding sources. The customer must return to PayPal to select new funding sources.
10486	This transaction couldn't be completed. Please redirect your customer to PayPal.
13113	The Buyer cannot pay with PayPal for this transaction. Inform the buyer that PayPal declined the transaction and to contact *PayPal Customer Service*.

*Table 11: Values of the **vads_auth_result** field*

8. Record the used payment method by retrieving the value of the **vads_payment_card** field.

9. Save all the details of the order, buyer and shipping.

   These details will be provided in the response only of they have been transmitted in the payment form.

   Their values is identical to the values submitted in the form.

10. Save the value of the **vads_authent_paypal_protection_eligibility** field to identify the type of protection used for the transaction.

    Three values are possible:

    - ELIGIBLE

      Merchant is protected by PayPal's Seller Protection Policy for unauthorized payments and Item Not Received.

    - PARTIALLY_ELIGIBLE

      Merchant is protected by PayPal's Seller Protection Policy for Item Not Received.

    - INELIGIBLE

      Merchant is not protected by PayPal's Seller Protection Policy for Item Not Received.

Example of a form:

```
vads_capture_delay = 0
vads_ctx_mode = TEST
vads_contract_used = merch_1284129052_biz@my-company.com
```



```
vads_auth_result = 00
vads_threeds_cavv =
vads_cust_country = FR
vads_operation_type = DEBIT
vads_threeds_cavvAlgorithm =
vads_threeds_exit_status =
vads_cust_name = My-company
vads_threeds_enrolled =
vads_currency = 978
vads_language = en
vads_sequence_number = 1
vads_trans_id = 502840
vads_pays_ip = FR
vads_payment_src = EC
vads_trans_date = 20170217100817
vads_amount = 4000
vads_auth_number:
vads_auth_mode = FULL
vads_version = V2
vads_payment_certificate = 4e3b2536e2aea84fa6cbbdc7694095d823910a48
signature = 1d03e967c14abb853c1ae2ba98495c2cc40628ca
vads_threeds_eci =
vads_card_country =
vads_ship_to_street_number = street number in the shipping address
vads_cust_address_number = street number
vads_ship_to_name = name in the shipping address
vads_cust_address = street name
vads_effective_amount = 4000
vads_ext_trans_id = 3GE25312T3314040N
vads_validation_mode = 0
vads_site_id = 91335531
vads_card_brand = PAYPAL_SB
vads_threeds_sign_valid =
vads_threeds_status =
vads_ship_to_zip = 31670
vads_cust_city = city name
vads_ship_to_street2 = street in the shipping address
vads_ship_to_country = FR
vads_page_action = PAYMENT
vads_ship_to_city = city in the shipping address
vads_cust_zip = 31670
vads_card_number = cust2_1282667599_per@my-email.com
vads_payment_config = SINGLE
vads_authent_paypal_protection_eligibility = ELIGIBLE
vads_trans_status = CAPTURED
vads_order_id = nkj-055
vads_effective_creation_date = 20170217100817
vads_presentation_date = 20170217100817
vads_warranty_result = NO
vads_cust_email = trash@my-email.com
vads_result = 00
vads_ship_to_status = PRIVATE
vads_ship_to_street = street in the shipping address
vads_risk_control = CARD_FRAUD=OK;COMMERCIAL_CARD=OK
vads_action_mode = INTERACTIVE
vads_extra_result = 00
vads_threeds_xid =
vads_threeds_error_code =
```


11.7. Processing errors

Setting up a log file

During the implementation phase, it is important to have logs especially in case of difficulties with signature computation.

It is recommended to set up a daily log file even after shifting the merchant website to live mode.

It will allow you to analyze data in case any problem occurs.

Ideally, a log file should contain the sent or received data, the chain obtained during signature computation before applying the hash function.

HTTP error code

In case an error occurs during notifications, the sent warning e-mail will specify the return code of the HTTP protocol.

There are 5 categories of return codes:

Code categories	Description
1XX	Information
2XX	Success
3XX	Redirection
4XX	Client error
5XX	Server error

A description of the error codes with their possible causes is available on our website

https://lyra.com/doc/en/collect/error-code/server-url-error.html

Frequent error:

An .htaccess file might block the call to the IPN.

.htaccess files are Apache web server configuration files. They can be stored in any folder of the merchant website (the configuration applies to the folder and all the contained folders with no .htaccess files).

12. IDENTIFYING OPERATIONS AUTHORIZED FOR PAYPAL TRANSACTIONS

In the Expert Back Office, there are several operations that can be performed with transactions.

In the list of transactions:

1. Select a transaction.
2. Right-click to view the list of authorized operations.

The authorized operations are:

- Refund
  The refund is only possible for a credit card
- Manual reconciliation
- Edit the order reference
- Resending transaction confirmation e-mail to the buyer
- Resending the transaction confirmation e-mail to the merchant

12.1. Refunding a captured transaction

Similarly to bank card transactions, it is possible to refund a part of the amount until its paid in full within the 60 days following the initial payment date.

1. Right-click a transaction.
2. Select .
   The dialog box appears.

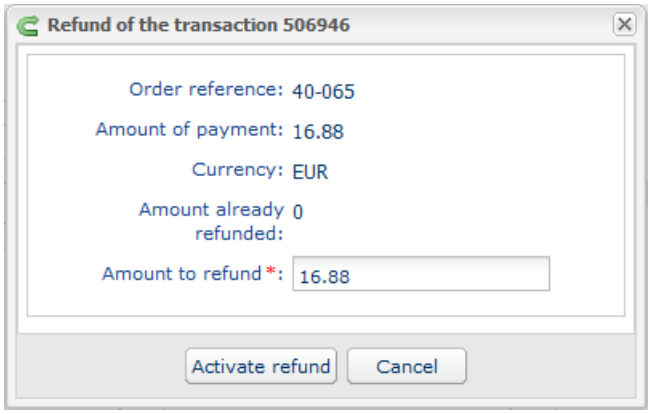


Figure 3:

3. Enter the amount that you wish to refund.
4. Click on .

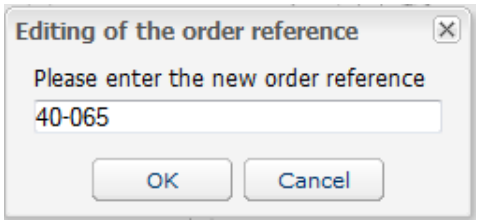
The detail of this operation appears.

12.2. Edit the order reference

This operation allows the merchant to change the order reference.

To edit the order reference of a transaction:

1. Right-click the transaction.
2. Select **Edit the order reference**.

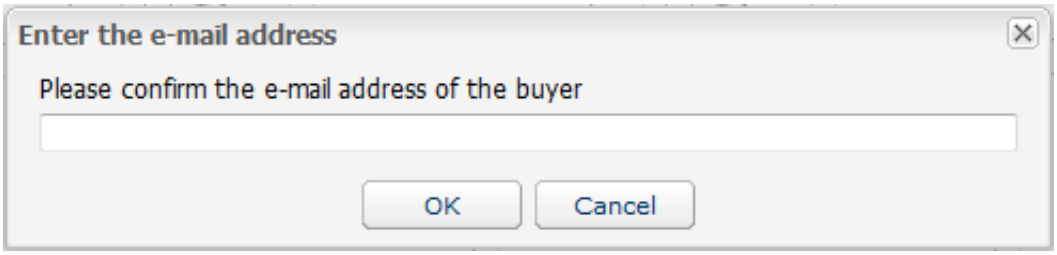


3. Enter the new order reference.
4. Click **OK**.

12.3. Resending transaction confirmation e-mail to the buyer

To resend the transaction confirmation e-mail to the buyer in case of non-receipt or correction of the email address.

1. Look for the transaction.
2. Right-click the transaction.
3. Right-click the transaction and click **Resending transaction confirmation e-mail to the buyer**.
   The Enter e-mail address dialog box appears.



4. Enter the e-mail address.
5. Click **OK**.

12.4. Resending the transaction confirmation e-mail to the merchant

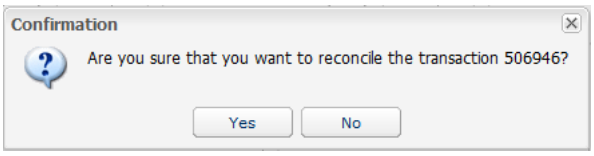
To resend the transaction confirmation e-mail to the merchant:

1. Look for the transaction.
2. Right-click the transaction and click **Resending the transaction confirmation e-mail to the merchant**.
   A confirmation message appears.
3. Click **OK**.

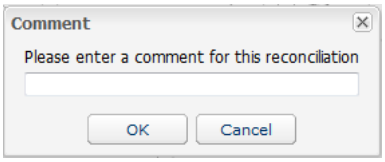
12.5. Reconciling a transaction manually

This operation allows you to manually reconcile merchant's payments from an account statement.

1. From the **Captured transactions** tab, look for the relevant transaction.
2. Right-click the transaction.
3. Select **Manual reconciliation**.
4. Click **Yes** to confirm the manual reconciliation of the selected transaction.



The **Comment** dialog box appears.



5. Enter a comment for this reconciliation.
6. Click **OK**.

The transaction status changes to **Reconciled**.

