



COLLECTING SOLUTION

Adding the payment button for UnionPay

Implementation Guide

Document version 1.2

Contents

| | |
|--|-----------|
| 1. HISTORY OF THE DOCUMENT..... | 3 |
| 2. OBTAINING HELP..... | 4 |
| 3. PRESENTATION..... | 5 |
| 4. TECHNICAL DETAILS..... | 6 |
| 5. PREREQUISITES..... | 7 |
| 6. PAYMENT PROCESS..... | 8 |
| 7. TRANSACTION SETTLEMENT..... | 9 |
| 8. ESTABLISHING INTERACTION WITH THE PAYMENT GATEWAY..... | 10 |
| 9. SETTING UP NOTIFICATIONS..... | 11 |
| 9.1. Setting up the Instant Payment Notification..... | 11 |
| 9.2. Setting up a notification on batch change..... | 12 |
| 10. GENERATING A PAYMENT FORM..... | 13 |
| 10.1. Creating an UnionPay payment request..... | 15 |
| 11. SENDING THE PAYMENT REQUEST..... | 16 |
| 11.1. Redirecting the buyer to the payment page..... | 16 |
| 11.2. Processing errors..... | 16 |
| 12. ANALYZING THE PAYMENT RESULT..... | 18 |
| 12.1. Processing the response data..... | 18 |
| 12.2. Analyzing the result of the authorization request..... | 19 |
| 13. MAKING A REFUND VIA THE EXPERT BACK OFFICE..... | 20 |

1. HISTORY OF THE DOCUMENT

| Version | Author | Date | Comment |
|---------|--------------|------------|---|
| 1.2 | Lyra Collect | 24/01/2020 | <ul style="list-style-type: none">• Update of the CAPTURED status description.• Update of additional information.• Integration by iframe not supported. |
| 1.1 | Lyra Collect | 30/09/2019 | <ul style="list-style-type: none">• Supported currencies updated.• Update of available operations. |
| 1.0 | Lyra Collect | 20/09/2019 | Original version |

This document and its contents are confidential. It is not legally binding. No part of this document may be reproduced and/or forwarded in whole or in part to a third party without the prior written consent of Lyra Collect. All rights reserved.

2. OBTAINING HELP

Looking for help? Check our FAQ on our website

<https://lyra.com/doc/fr/collect/faq/sitemap.html>

If you have any technical questions or need assistance, our tech support is available from Monday to Friday from 9 AM to 6 PM

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**

To facilitate the processing of your demands, you will be asked to communicate your shop ID (an 8-digit number) .

3. PRESENTATION



China UnionPay (or CUP) is the world's largest card network.
They are accepted in more than 140 countries including France.



Supported currencies

- EUR
- USD
- GBP



Supported countries

- China



Additional information

- Debit is immediate (no capture delay).
- Transaction validation is automatic (no manual validation).
- No capture file is sent to the acquirer.

4. TECHNICAL DETAILS

Important information regarding the addition of this payment method:

- **Type of integration**

Only available via integration with payment form redirection.

- **Authorization validity period**

0 day

- **Network code**

LYRA_COLLECT_PPRO

- **Value of the vads_card_brand field**

UNION_PAY

- **Supported currencies**

EUR; USD

- **Supported countries**

CN

- **Operations available for transactions**

- Refund (full and partial)
- Bank reconciliation and chargebacks

- **Restrictions**

This payment method does not support the following functionalities:

- Integration by iframe
- Installment payment
- Tokenization and payment by token
- Recurring payment by token
- MOTO payment
- Transaction duplication
- Transaction modification
- Transaction validation

5. PREREQUISITES

In order to enable the UnionPay payment method:

1. Contact the Middle Office.
2. Fill out and send the **KYC** and **Merchant Registration Form**.
3. Once all the documents have been validated, the UnionPay payment method will be enabled for your shop.

6. PAYMENT PROCESS

You will find below the payment flow of a payment via UnionPay:



Note:

If the mandatory fields have not been provided (see chapter **Creating an UnionPay payment request**), another page is provided to the buyer to allow him or her to fill in their personal details.

7. TRANSACTION SETTLEMENT

UnionPay transactions are supported by the following services:

- Visual transaction settlement,
- Visual reconciliation of chargebacks,
- Bank reconciliation report,
- Chargeback reconciliation report.

If you wish to enable or obtain more information on these services, contact the Middle Office.

8. 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 known only to the merchant and the payment gateway.

9. SETTING UP NOTIFICATIONS

Several types of notifications are provided in the Expert Back Office. They allow the configuration of the URL of the page to contact and the management of the events (payment abandoned by the buyer, payment canceled by the merchant, payment validated by the merchant, etc.) that will trigger a call to the merchant website.

If the payment gateway is unable to access the URL of your page, an e-mail will be sent to the shop administrator.

It contains:

- The HTTP code of the encountered error
- Parts of error analysis
- Its consequences
- Instructions to resend from the Expert Back Office the notification to the URL already specified above.

To access notification rule management:

Go to the following menu: **Settings > Notification rules.**

| Enabled | Reference |
|---------|--|
| ✗ | Instant Payment Notification URL on batch authorization |
| ✓ | Instant Payment Notification URL at the end of the payment |
| ✗ | Instant Payment Notification URL on batch change |
| ✗ | Instant Payment Notification URL on cancellation |
| ✗ | Instant Payment Notification URL on an operation coming from the Back Office |

9.1. Setting up the Instant Payment Notification

This rule allows to notify the merchant website in the following cases:

- Payment accepted
- Payment refused

This notification is required to communicate the result of a payment request.

It will inform the merchant website of the payment result even if your client has not clicked the “Return to the shop” button.

1. Right-click the **Instant Payment Notification URL at the end of the payment** line.
2. Select **Manage the rule.**
3. Enter the **E-mail address(es) to notify in case of failure** field in the **General settings** section.
To specify several e-mail addresses, separate them with a semi-colon.
4. Check the box **Automatic retry in case of failure** if you wish to authorize the gateway to automatically resend the notification in case of a failure (can be done up to 4 times).
5. In the **Instant Payment Notification URL of the API form V1, V2** section, specify the URL of your page in the fields **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** if you wish to receive notifications in the form API format.

6. In the **REST API Instant Payment Notification URL** section, specify the URL of your page in the fields **Target URL of the IPN to notify in TEST mode** and **Target URL of the IPN to notify in PRODUCTION mode** if you are using the JavaScript client.
7. Save the modifications.

9.2. Setting up a notification on batch change

In some cases, the final payment result may not be available when the buyer returns to the merchant website.

The payment is then created with the **Waiting for authorization** status.

An automatic mechanism allows to retrieve the final status and update the transaction.

In order to be notified of the final status, you must enable the notification rule for batch modification:

1. Right-click the **Instant Payment Notification URL on batch change** line.
2. Select **Manage the rule**.
3. Enter the **E-mail address(es) to notify in case of failure** field in the **General settings** section.
To specify several e-mail addresses, separate them with a semi-colon.
4. Check the box **Automatic retry in case of failure** if you wish to authorize the gateway to automatically resend the notification in case of a failure (can be done up to 4 times).
5. In the **Instant Payment Notification URL of the API form V1, V2** section, specify the URL of your page in the fields **URL to notify in TEST mode** and **URL to notify in PRODUCTION mode** if you wish to receive notifications in the form API format.
6. In the **REST API Instant Payment Notification URL** section, specify the URL of your page in the fields **Target URL of the IPN to notify in TEST mode** and **Target URL of the IPN to notify in PRODUCTION mode** if you are using the JavaScript client.
7. Save the modifications.
8. Enable the rule by right-clicking **Instant Payment Notification URL on batch change** and select **Enable the rule**.

10. 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.
- Additional information depending on your needs.
- The signature that ensures the integrity of the form.

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 the **Data dictionary** chapter also available in the online document archive.

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 following table lists the different formats that you can encounter when building your form.

| 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 | <p>JavaScript Object Notation.</p> <p>Object containing key/value pairs separated by commas.</p> <p>It starts with a left brace "{" and ends with a right brace "}".</p> <p>Each key/value pair contains the name of the key between double-quotes followed by ":", followed by a value.</p> <p>The name of the key must be alphanumeric.</p> <p>The value can be:</p> <ul style="list-style-type: none"> • a chain of characters (in this case it must be framed by double-quotes) • a number • an object • a table • a boolean • empty <p>Example: {"name1":45,"name2":"value2", "name3":false}</p> |
| enum | <p>Characterizes a field with a complete list of values.</p> <p>The list of possible values is given in the field definition.</p> |
| Enum list | <p>List of values separated by a ";".</p> <p>The list of possible values is given in the field definition.</p> <p>Example: vads_payment_cards=VISA;MASTERCARD</p> |
| map | <p>List of key / value pair separated by a ";".</p> <p>Each key / value pair contains the name of the key followed by "=", followed by a value.</p> <p>The value can be:</p> <ul style="list-style-type: none"> • a chain of characters • a boolean • a json object • an xml object <p>The list of possible values for each key/value pair is provided in the field definition.</p> <p>Example: vads_theme_config=SIMPLIFIED_DISPLAY=true;RESPONSIVE_MODEL=Model_1</p> |

10.1. Creating an UnionPay payment request

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

| Field name | Description | Format | Value |
|-----------------------------|---|----------|---|
| vads_payment_cards | Allows to force the card type to use. Leave empty to let the buyer choose between the eligible payment methods (depending on the currency, the minimum or maximum amount, the technical constraints) associated with the shop. | enum | <ul style="list-style-type: none">• UNION_PAY to directly call the UnionPay payment button.• VISA;UNION_PAY to provide the choice between VISA and UnionPay. |
| vads_page_action | Action to perform | enum | 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 | Defines the mode of interaction with the payment gateway. | enum | TEST or PRODUCTION |
| vads_currency | Numeric currency code to be used for the payment, in compliance with the ISO 4217 standard (numeric code). | n3 | Possible values: <ul style="list-style-type: none">• 978 (EUR)• 826 (GBP)• 840 (USD) |
| vads_action_mode | Acquisition mode for payment method data | enum | INTERACTIVE |
| vads_cust_email | Buyer's e-mail address | ans..150 | E.g.: abc@example.com |
| vads_cust_country | Country code in compliance with the ISO 3166 standard | a2 | Possible value: CN |
| vads_cust_first_name | First name | ans..63 | E.g.: Laurent |
| vads_cust_last_name | Name | ans..63 | E.g.: Durant |
| vads_payment_config | Payment type | 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 | Respect the YYYYMMDDhhmmss format E.g.: 20191201130025 |
| vads_trans_id | Transaction number | an6 | E.g.: xrT15p |
| vads_version | Version of the exchange protocol with the payment gateway | enum | V2 |

2. Add optional fields depending on your requirements (see chapter **Using additional features** of the *Hosted Payment Page Implementation Guide* available in our online document archive).
3. Compute the value of the **signature** field using all the fields of your form that start with **vads_** (see chapter **Computing the signature** of the *Hosted Payment Page Implementation Guide* available in our online document archive).

11. 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.

11.1. 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="CN" />
<input type="hidden" name="vads_cust_email" value="me@example.com" />
<input type="hidden" name="vads_cust_first_name" value="John" />
<input type="hidden" name="vads_cust_last_name" value="Doe" />
<input type="hidden" name="vads_cust_phone" value="+33102030405" />
<input type="hidden" name="vads_page_action" value="PAYMENT" />
<input type="hidden" name="vads_payment_cards" value="UNION_PAY" />
<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="20190710101407" />
<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="pay" value="Pay" />
</form>
```

11.2. 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 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/fr/collect/error-code/error-00.html>

Error codes specific to the UNION_PAY payment method:

| Code | Message | Description of the error |
|------|--|---|
| 142 | PAYMENT_CARD_LYRA_COLLECT_NOT_AVAILABLE_CURRENCY | No card type supports the requested currency. |
| 144 | PAYMENT_CARD_LYRA_COLLECT_COUNTRY_COVERAGE | No card type supports the requested country. |
| 152 | PAYMENT_CARD_LYRA_COLLECT_AMOUNT_NOT_ALLOWED | The transmitted amount is too low. |

12. 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 (<https://lyra.com/doc/fr/>).

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

12.1. Processing the response data

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

1. Identify the order by retrieving the value of the **vads_order_id** field if you have transmitted it in the payment form.

Make sure that the order status has not already been updated.

2. Retrieve the payment result transmitted in the **vads_trans_status** field.

Its value allows you to define the order status.

| VALUE | DESCRIPTION |
|------------------------------|---|
| CAPTURED | Captured The transaction is accepted. It is visible in the Captured transactions tab. Debit is immediate (no capture file is sent to the acquirer). |
| REFUSED | Declined Transaction is declined. See the following chapter to obtain more information on the rejected payment. |
| WAITING_AUTHORISATION | Waiting for authorization. Waiting for the response from the acquirer. |

3. Retrieve the payment reference transmitted in the **vads_trans_id** field.

4. Retrieve the value of the **vads_trans_date** field to identify the payment date.

5. Retrieve the used amount and 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 | Numeric code of the currency used for the payment. |
| vads_change_rate | Exchange rate used to calculate the effective payment amount (see vads_effective_amount). |
| vads_effective_amount | Payment amount in the currency used for the capture in the bank. |
| vads_effective_currency | Numeric code of the currency that will be used for the capture at the bank. |

Table 1: Analysis of the payment amount and currency

6. Retrieve the used payment method by analyzing the **vads_card_brand** field.

7. Retrieve all the order, buyer and shipping details.

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

Their values are identical to the ones submitted in the form.

8. Proceed to order update.

12.2. Analyzing the result of the authorization request

The result of the authorization is specified in the **vads_auth_result** field.

This field is set to:

- **0** when the payment is successful,
- one of the values below when the payment is rejected:

| Codes | Description | Comment |
|-------------|--------------------------------------|---|
| 1010 | Unknown error | |
| 1011 | Incorrect information | A non-exhaustive list of possible cases: <ul style="list-style-type: none">• Invalid cardholder number• Invalid CVV• Invalid expiration date• Invalid buyer country• Invalid transaction currency |
| 1012 | Payment abandoned by the user | |
| 1013 | Expired | A non-exhaustive list of possible cases: <ul style="list-style-type: none">• Expired session |
| 1014 | Technical error on the acquirer side | Contact the tech support. |
| 1016 | Canceled by the acquirer | A non-exhaustive list of possible cases: <ul style="list-style-type: none">• Risk management module• failed buyer authentication |

13. MAKING A REFUND VIA THE EXPERT BACK OFFICE

This payment method supports partial and total refund of a captured transaction.

To perform a refund:

1. Right click a transaction to refund.
2. Select **Making a refund** in the context menu.
The dialog box **Transaction refund** appears.
3. Enter the amount to be refunded.
4. Click **Refund**.

Details of this operation appear.

The refund is put on hold until its validation by the Middle Office.

View your bank reconciliation logs or the refund status in your Expert Back Office.