



Adding the Google Pay payment button

Document version 1.1

Contents

1. HISTORY OF THE DOCUMENT.....	3
2. PRESENTATION.....	4
3. EXCHANGE PROCESS.....	5
4. DECLARING A GOOGLE PAY[™] MERCHANT ID IN THE MERCHANT BACK OFFICE.....	6
5. ESTABLISHING INTERACTION WITH THE PAYMENT GATEWAY.....	7
6. GENERATING A PAYMENT FORM.....	8
6.1. Creating a payment request.....	9
7. USING ADDITIONAL FEATURES.....	11
7.1. Transmitting buyer details.....	11
7.2. Transmitting shipping details.....	12
7.3. Transmitting order details.....	13
8. COMPUTING THE SIGNATURE.....	14
9. SENDING THE PAYMENT REQUEST.....	16
9.1. Redirecting the buyer to the payment page.....	16
9.2. Processing errors.....	16
10. ANALYZING THE PAYMENT RESULT.....	17
11. MANAGING YOUR GOOGLE PAY[™] TRANSACTIONS VIA THE BACK OFFICE	
PAYZEN BY OSB.....	19
11.1. Display transaction details Google Pay [™]	20
11.2. Validating a transaction.....	21
11.3. Modifying a transaction.....	22
11.4. Canceling a transaction.....	23
11.5. Editing the order reference.....	24
11.6. Resending transaction confirmation e-mail to the buyer.....	24
11.7. Resending the transaction confirmation e-mail to the merchant.....	24
11.8. Capturing a transaction.....	25
11.9. Manual reconciliation.....	25
11.10. Making a refund.....	26

1. HISTORY OF THE DOCUMENT

Version	Author	Date	Comment
1.1	OSB (Océanienne de Service Bancaire)	4/25/2019	Indication of the Trade Mark label.
1.0	OSB (Océanienne de Service Bancaire)	1/15/2019	Initial version

This document and its contents are confidential. It is not legally binding. Any reproduction and / or distribution of all or part of this document or its content to a third party is strictly prohibited or subject to prior written authorization from OSB (Océanienne de Service Bancaire). All rights reserved.

2. PRESENTATION



Google Pay™ is a payment method allowing buyers to easily pay for their purchases using the payment card associated with their Google account.

It is a simple, fast and secure method of performing purchases online or using a mobile app.

There are two types of integration:

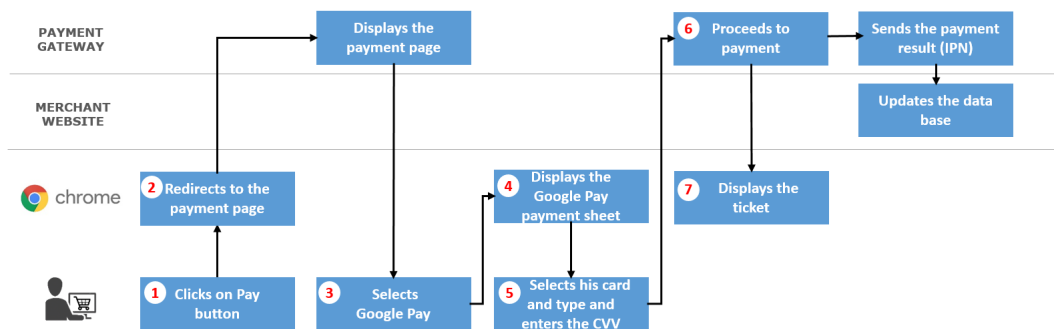
- integration within a mobile app
- integration on a merchant website (payment is made via buyer's browser).

This document describes the integration process on the merchant website. The exchange with Google Pay™ servers are managed by the payment gateway. The merchant website does not manipulate payload data.

Important:

- The current implementation does not allow 3D Secure authentication for Google Pay™ payments. Therefore, there is no liability shift.
- The current implementation does not allow to create recurring payments (payments by subscription) with the Google Pay™ payment method.

3. EXCHANGE PROCESS



1. The buyer validates their shopping cart by clicking the **Pay** button.
2. The buyer is redirected to the payment page.
3. The buyer selects Google Pay™.
4. The browser displays the Google Pay™ pop-up.
5. The buyer selects their card and enters their CVV.

The CVV entry depends on the configuration of the merchant upon creation of the Google Pay™ MID in the Merchant Back Office.

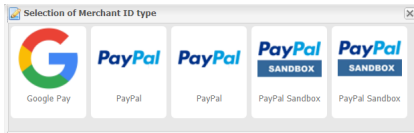
6. The payment gateway proceeds to payment and notifies the merchant website.
7. The payment gateway displays the payment receipt.

4. DECLARING A GOOGLE PAY™ MERCHANT ID IN THE MERCHANT BACK OFFICE

Via the menu **Settings > Company > Merchant IDs**:

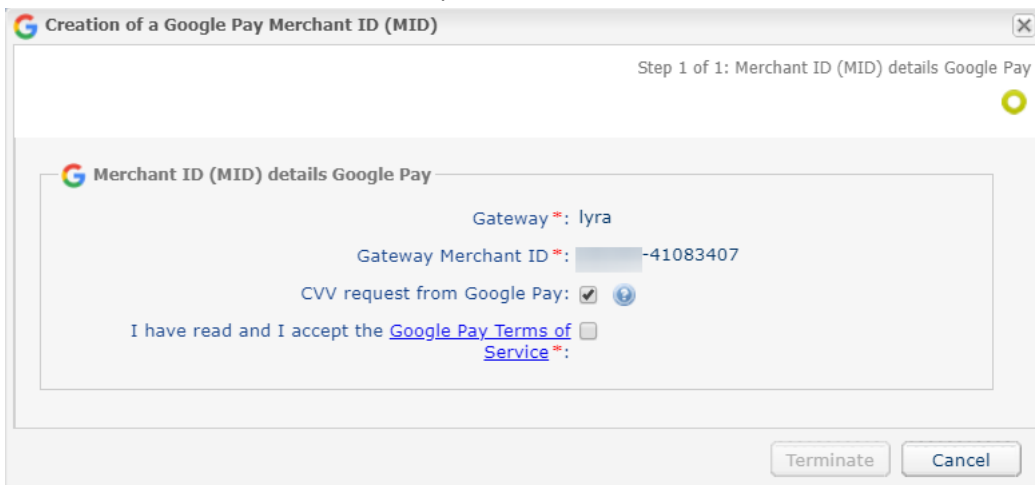
1. Click **Create a Merchant ID (MID)**.

The "Selection of Merchant ID type" dialog box appears.



2. Select **Google Pay™**.

The merchant ID creation assistant opens.

A dialog box titled "Creation of a Google Pay Merchant ID (MID)" with a close button (X) in the top right corner. It shows "Step 1 of 1: Merchant ID (MID) details Google Pay". The main content area is titled "Merchant ID (MID) details Google Pay" and contains the following fields and options: "Gateway *:" with the value "lyra", "Gateway Merchant ID *:" with the value "-41083407", "CVV request from Google Pay:" with a checked checkbox and a help icon, and "I have read and I accept the Google Pay Terms of Service *:" with an unchecked checkbox. At the bottom right, there are "Terminate" and "Cancel" buttons.

The **Gateway** and **Gateway Merchant ID** fields are populated. This information is useful for integrating Google Pay™ into your mobile application.

If the option **CVV request from Google Pay™** is checked, the CVV will be automatically requested by Google Pay™ when the buyer selects his or her card.

Otherwise, depending on the configuration of your acquirer Merchant ID (MID), the buyer may have to enter their CVV in a separate window before authorization (if the acquirer insists on receiving the CVV in order to transmit it to the issuer).

3. Read and accept the General Conditions of Service of Google Pay™ by checking the box provided for that purpose.

This action is required for finalizing the creation of a Merchant ID.

4. Click **Terminate**.

Once the Merchant ID has been created, click the **"Associate with a shop"** button and associate it with a shop of your choosing.

5. ESTABLISHING INTERACTION WITH THE PAYMENT GATEWAY

The interaction with the payment gateway is described in the *Hosted Payment Page Implementation Guide*, available in our online documentation archive.

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).
- Via 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.

6. GENERATING A PAYMENT FORM

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

```
<form method="POST" action="https://secure.osb.pf/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.osb.pf/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 be computed incorrectly and the form will be rejected.

- The **Pay** button to submit the 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	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: <ul style="list-style-type: none">a chain of characters (in this case it must be framed by double-quotes)a numberan objecta tablea booleanempty 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 pairs separated by a ";". Each key/value pair contains the name of the key followed by "=", followed by a value. The value can be: <ul style="list-style-type: none">a chain of charactersa booleana json objectan 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 a payment request

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

Field name	Description	Format	Value
vads_version	Version of the exchange protocol with the payment gateway	enum	V2
vads_ctx_mode	Defines the mode of interaction with the payment gateway.	enum	TEST or PRODUCTION
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.: 20200101130025

Field name	Description	Format	Value
vads_page_action	Action to perform	enum	PAYMENT
vads_action_mode	Acquisition mode for payment method data	enum	INTERACTIVE
vads_payment_config	Payment type	enum	SINGLE
vads_trans_id	Transaction number. Warning: this field is not case sensitive.	an6	E.g.: xrT15p
vads_amount	Payment amount in the smallest currency unit.	n..12	E.g.: 3000 for 3000 XPF
vads_currency	Numeric currency code to be used for the payment, in compliance with the ISO 4217 standard (numeric code).	n3	E.g.: 953 for CFP Franc (XPF)
vads_cust_email	Buyer's e-mail address	ans..150	E.g.: abc@example.com
signature	Signature guaranteeing the integrity of the requests exchanged between the merchant website and the payment gateway.	ans	See chapter Computing the signature

2. Populate the **vads_payment_cards** field with GOOGLEPAY if you wish to offer only Google Pay™ on the payment page.
3. Add optional fields according to your requirements (see chapter **Using additional features**).

7. USING ADDITIONAL FEATURES

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

- Transfer buyer details
- Transfer shipping details
- Transfer order details

7.1. 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 Merchant Back Office (**Buyer** tab).

Use optional fields according to your requirements. *These fields will be returned with 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.: Doom
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.:
vads_cust_address_number	Street number	ans..64	E.g.: 12
vads_cust_address	Postal address	ans..255	E.g.: Baker street
vads_cust_address2	Second line of the address	ans..255	E.g.:
vads_cust_district	District	ans..127	E.g.: Estall
vads_cust_zip	Zip code	an..64	E.g.:
vads_cust_city	City	an..128	E.g.:
vads_cust_state	State / Region	ans..127	E.g.:
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

Note

vads_cust_phone and **vads_cust_cell_phone** fields accept all formats:

Examples:

- 0123456789
- +33123456789
- 0033123456789
- (00.571) 638.14.00
- 40 41 42 42

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 Merchant Back Office (**Delivery** tab).

Use optional fields according to your requirements. *These fields will be returned with 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.: Papeete
vads_ship_to_country	Country code in compliance with the ISO 3166 standard (required for triggering one or more actions if the Shipping country control profile is enabled).	a2	E.g.: PF
vads_ship_to_district	District	ans..127	E.g.: Mission
vads_ship_to_first_name	First name	ans..63	E.g.: Moana
vads_ship_to_last_name	Name	ans..63	E.g.: Doom
vads_ship_to_legal_name	Legal name	an..100	E.g.: D. & Cie
vads_ship_to_phone_num	Phone number	ans..32	E.g.: 40975711
vads_ship_to_state	State / Region	ans..127	E.g.: Tahiti
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.: Impasse Cardela
vads_ship_to_street2	Second line of the address	ans..255	
vads_ship_to_zip	Zip code	an..64	E.g.: 98713

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 Merchant Back Office (**Shopping cart** tab).

Field name	Description	Format	Value
vads_order_id	Order ID Can contain uppercase or lowercase characters, numbers or hyphens ([A-Z] [a-z], 0-9, _, -).	ans..64	E.g.: 2-XQ001
vads_order_info	Additional order info	an..255	E.g.: Door phone code 3125
vads_order_info2	Additional order info	an..255	E.g.: No elevator
vads_order_info3	Additional 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 first one, 1 for the second one, etc.).	ans..255	E.g.: vads_product_label0 = "tee-shirt" vads_product_label1 = "Biscuit" 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"

8. 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 Merchant Back Office via **Settings > Shop > Keys** tab.

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

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

The use of SHA-1 algorithm is deprecated but maintained for compliance reasons.

Warning: you must not use the REST API keys for computing the signature of your payment form.

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. **Deprecated.**
 - 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.osb.pf/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="953" />
<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="vSlCWjJwN8TpobRyuyKhWAlKEhlThtICZiI/rmpPK4U=" />

<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+953+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+953+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:
fbdc29bb585e6ff050c625134cad25e914f01539

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:

vSlCWjJwN8TpobRyuyKhWAlKEhlThtICZiI/rmpPK4U=

9. 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.

9.1. Redirecting the buyer to the payment page

The URL of the payment gateway is:

<https://secure.osb.pf/vads-payment/>

Example of parameters sent to the payment gateway:

```
<form method="POST" action="https://secure.osb.pf/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="953" />
  <input type="hidden" name="vads_cust_country" value="PF" />
  <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_cards" value="GOOGLEPAY" />
  <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="20190114101407" />
  <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>
```

9.2. 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://secure.osb.pf/doc/fr-FR/error-code/sitemap.html>

10. ANALYZING THE PAYMENT RESULT

To process the payment result, the merchant website must have a separate page with a script (e.g. `analyze_payment.php`).

This page will be automatically called after each payment (whether it has been accepted or declined): the parameters linked to the payment result are sent in POST mode by the payment gateway.

Prerequisites:

- URL of the page that analyzes the payment result must be specified in the Merchant Back Office (see chapter **Setting up notifications** of the *Implementation Guide Hosted Payment Page*).
- The merchant has to make sure that this URL is available via the payment gateway without redirection. Using redirection leads to losing data presented in POST.
- In case some restrictions are set up by the merchant, the **194.50.38.0/24** IP address range must be authorized.

*Notifications are sent from an IP address in the **194.50.38.0/24** range in Test and Production mode.*

- HTML should not be visible on the page.

Access to images or CSS may slow down the exchange between the payment gateway and the merchant website.

Furthermore, the payment gateway systematically reads the 512 first characters returned by the merchant website.

These characters can be viewed in the transaction history.

- Avoid integrating time-consuming tasks, such as invoice generation or sending e-mails in the script.

The processing time has a direct influence on how long it takes to display the payment summary page. The longer the processing of the notification, the greater the delay for displaying the page.

After 35 seconds, the payment gateway will consider that the call has failed (timeout).

Failed notification (IPN)

In case the call to IPN fails, a notification e-mail is sent to the address specified in the Merchant Back Office (see chapter **Setting up notifications**).

It contains:

- The HTTP code of the encountered error,
- Analysis parts depending on the error,
- Instructions for resending the notification from the Merchant Back Office.

Writing the processing script

The processing script must include at least the following steps:

- Retrieve the field list sent with the POST response
- Compute the signature
- Compare the computed signature with the received signature
- Analyze the nature of the notification
- Retrieve the payment result

The script may check the order status (or any information of your choice) to see if it has not been already updated.

Once these steps are completed, the script can update the database (new order status, stock update, registration of payment information, etc.).

11. MANAGING YOUR GOOGLE PAY™ TRANSACTIONS VIA THE BACK OFFICE PAYZEN BY OSB

The Back Office allows to perform different actions with Google Pay™ transactions depending on their status:

Actions available via the **Transactions in progress** tab:

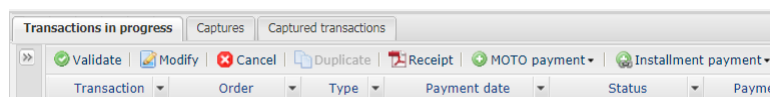
- Display transaction details
- Validate
- Modify
- Cancel
- Capture manually (only for transactions performed in test environment)
- Edit the order reference
- Resending transaction confirmation e-mail to the buyer
- Resending the transaction confirmation e-mail to the merchant.

Actions available via the **Captured transactions** tab:

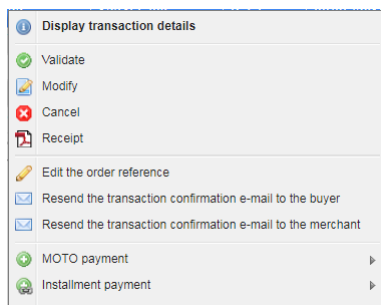
- Refund
- Duplicate
- Edit the order reference
- Resending transaction confirmation e-mail to the buyer
- Resending the transaction confirmation e-mail to the merchant
- Manual reconciliation.

You can access these actions in three ways:

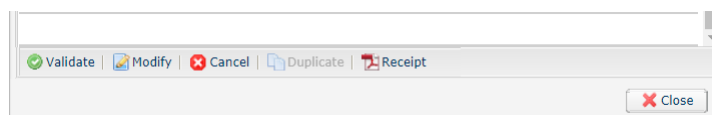
- via the menu bar



- using right-click



- at the bottom of the **Transaction details**.



11.1. Display transaction details Google Pay™

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

Via the **Management** menu, the merchant has access to real and TEST transactions.

Note:

Depending on the access rights, TEST transactions (e.g.: developer profile) and/or real transactions (e.g.: accountant profile) can be viewed.

The content of the **Transactions is progress** tab is displayed by default. All the transactions of the day are listed.

Characteristics of a payment made with Google Pay™ :

Successful payments can be viewed in the Merchant Back Office, **Transactions is progress tab**.

Failed payments can be viewed in the Merchant Back Office, **Transactions is progress tab**.

To view the details of a transaction:

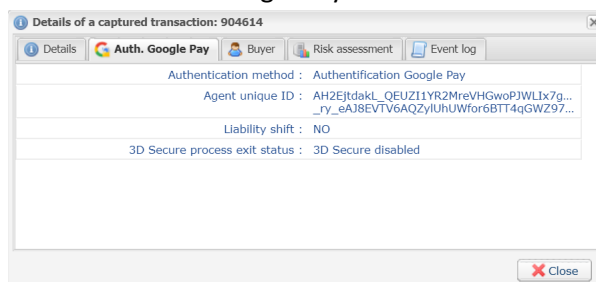
1. Select a transaction.
2. Right click on it and select **Display transaction details** or double-click the transaction you wish to see the details of.

The **Details of a transaction in progress** dialog box appears.

The details include for example:

- the payment method
- the order reference
- the transaction amount
- the creation date of the transaction
- the transaction status
- The type of the digital wallet: Google Pay™

3. Click the **Google Pay Auth.** tab to view the Google Pay™ authentication details.



11.2. Validating a transaction

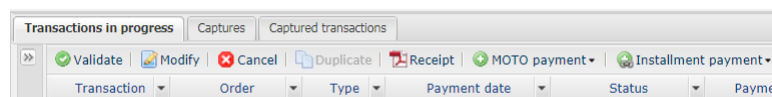
The **Validate** action is available when the transaction has one of the following statuses:

- To be validated
- To be validated and authorized

This action allows to validate the capture at the bank where the transaction took place. A transaction that has not been validated before the expiry date of the authorization request will get the expired status. It will not be captured at the bank.

In order to validate a transaction:

1. Display the tab **Transactions in progress**



2. Select the transaction

3. Click **Validate**

Once the transaction is validated, its status changes to:

- “**Waiting for capture**” for transactions with the “**To be validated**” status,
- “**Waiting for authorization**” for transactions with the “**To be validated and authorized**” status.

11.3. Modify ing a transaction

The **Modify** action is available when the transaction has one of the following statuses:

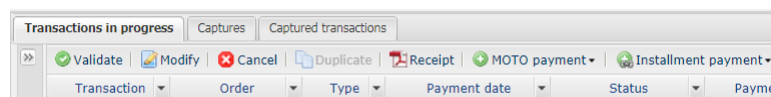
- To be validated
- To be validated and authorized
- Waiting for authorization
- Waiting for capture

This action allows to modify the amount and the capture date at the bank with respect to the following constraints:

- the modified amount cannot be greater than the initial amount
- when the transaction has not yet been authorized, the capture date can be defined anytime between the current date and the capture date specified by the merchant during the payment.
An authorization request will be automatically triggered if the selected capture date is between the current date and the expiry date of the authorization request (e.g.: period of 7 days for CB).
- when the transaction has already been authorized, the capture date at the bank cannot be later than the expiry date of the authorization (e.g.: 7 days for CB).

To modify a transaction:

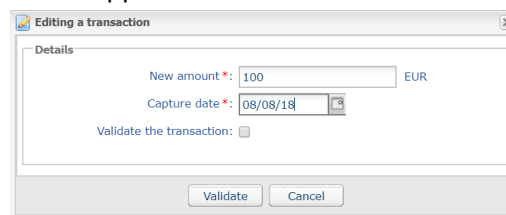
1. Display the tab **Transactions in progress**



2. Select the transaction

3. Click **Modify**.

The dialog box **Editing a transaction** appears.



4. If you wish to modify the transaction amount, enter the new amount.

Reminder: the new amount must be lower than the initial amount.

5. If you wish to modify the capture date at the bank, enter the capture date.

The calendar will show the authorized slot for the capture date. The slot is calculated based on when the authorization expires. The authorization validity period depends on the payment method and the network that was used for the authorization request (e.g.: 7 days for CB).

It is also possible to validate a transaction with the **To be validated** or **To be validated and authorized** status, by checking **Validate the transaction**.

6. Click **Validate**.

If you wish, you may view the transaction details to see the applied changes (right-click the edited transaction **Display transaction details**).

11.4. Canceling a transaction

The **Cancel** option is only available for the transactions that have not been captured.

1. Select a transaction with a right-click.
2. Select **Cancel**.
3. Confirm that you wish to definitively cancel the selected transaction.

The transaction status changes to **Canceled**.

Note

*It is possible to **cancel** several transactions at the same time.*

*All you need to do to is select all transactions. You can use the **Ctrl key** and **click** to make a multiple selection.*

*After the selection, you can click **Cancel** using right-click or via the menu bar and confirm your choice.*

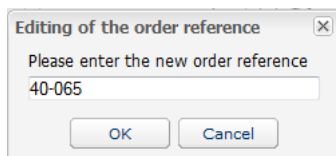
*The transaction statuses will change to **Canceled**.*

11.5. Editing 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**.

11.6. 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 e-mail 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 dialog box for entering the e-mail address appears.
4. Enter the e-mail address.
5. Click **OK**.

11.7. 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**.

11.8. Capturing a transaction

This operation is available during test phase. It is not available in production environment.

The **Capture** option is only available for transactions that have not reached the presentation date.

To manually capture a transaction:

1. Display the tab **Transactions in progress**
2. Select a transaction with a right-click.
3. Select **Capture manually**.
4. Confirm that you wish to definitively capture the selected transaction.

11.9. Manual reconciliation

This operation allows you to manually reconcile a 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**.

11.10. Making a refund

The **Making a refund** action is available when the transaction has the **Captured** status.

This operation allows to re-credit the buyer's account.

The buyer's account is credited with the refunded amount, this same amount is debited from the merchant's account.

Depending on the acquirer, it is possible to refund the transaction amount partially or fully.

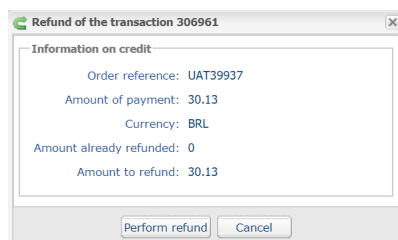
The refund delay available to the accounts also depend on the acquirer.

To perform a refund.

1. Display the tab **Captured transactions**
2. Select the transaction
3. Click **Refund**.

The dialog box **Transaction refund** appears.

Example of a full refund



Refund of the transaction 306961

Information on credit

Order reference: UAT39937

Amount of payment: 30.13

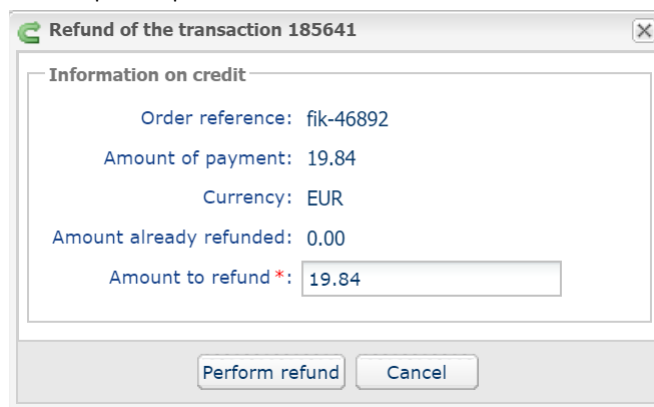
Currency: BRL

Amount already refunded: 0

Amount to refund: 30.13

Perform refund Cancel

Example of a partial refund



Refund of the transaction 185641

Information on credit

Order reference: fik-46892

Amount of payment: 19.84

Currency: EUR

Amount already refunded: 0.00

Amount to refund *: 19.84

Perform refund Cancel

4. Enter the amount that you want to refund.
The input field appears if partial refund is possible.
5. Click **Perform refund**.

The details of the refund transaction appears.